MITSUBISHI SEMICONDUCTORS 1991

SINGLE-CHIP 16-BIT MICROCOMPUTERS

Enlarged edition

DATA BOOK

MITSUBISHI ELECTRIC

MITSUBISHI SEMICONDUCTORS 1991

SINGLE-CHIP 16-BIT MICROCOMPUTERS

Enlarged edition

DATA BOOK

MITSUBISHI ELECTRIC

All values shown in this catalogue are subject to change for product improvement.

The information, diagrams and all other data included herein are believed to be correct and reliable. However, no responsibility is assumed by Mitsubishi Electric Corporation for their use, nor for any infringements of patents or other rights belonging to third parties which may result from their use.

GUIDANCE 1

MELPS 7700 16-BIT MICROCOMPUTERS 2

PROGRAMMABLE ROM MICROCOMPUTERS 3

APPENDICES 4

INDEX

1 GUIDANCE

PAGE
Index by Function ..... 1—3
Development Support Systems ..... 1—6
Ordering Information ..... 1—8
Package Outlines ..... 1—9
Letter Symbols for The Dynamic Parameters ..... 1—12
Symbology ..... 1—15
Quality Assurance and Reliability Testing ..... 1—18
Precaution in Handing MOS IC/LSIs ..... 1—24

2 MELPS 7700 16-BIT MICROCOMPUTERS

M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP, M37702S1FP, M37702S1AFP, M37702S1BFP
Single-Chip 16-Bit CMOS Microcomputer ..... 2—3
M37702M4-XXXFP, M37702M4AXXXFP, M37702M4BXXXFP, M37702S4FP, M37702S4AFP, M37702S4BFP
Single-Chip 16-Bit CMOS Microcomputer ..... 2—62
M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP, M37703S1SP, M37703S1ASP, M37703S1BSP
Single-Chip 16-Bit CMOS Microcomputer ..... 2—65
M37703M4-XXXSP, M37703M4AXXXSP, M37703M4BXXXSP, M37703S4SP, M37703S4ASP, M37703S4BSP
Single-Chip 16-Bit CMOS Microcomputer ..... 2—84
M37720S1FP, M37720S1AFP 16-Bit CMOS Microcomputer ..... 2—86
M37730S2FP, M37730S2AFP, M37730S2BFP, M37730S2SP, M37730S2ASP, M37730S2BSP
16-Bit CMOS Microcomputer ..... 2—179
M37732S4FP, M37732S4AFP, M37732S4BFP
16-Bit CMOS Microcomputer ..... 2—237
MELPS 7700 Addressing Modes ..... 2—298
MELPS 7700 Instruction Code Table ..... 2—341
MELPS 7700 Machine Instructions ..... 2—344

3 PROGRAMMABLE ROM MICROCOMPUTERS

M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP, M37702E2FS, M37702E2AFS, M37702E2BFS
PROM Version of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP ..... 3—3
M37702E4-XXXFP, M37702E4AXXXFP, M37702E4BXXXFP, M37702E4FS, M37702E4AFS, M37702E4BFS
PROM Version of M37702M4-XXXFP, M37702M4AXXXFP, M37702M4BXXXFP ..... 3—26
M37703E2-XXXSP, M37703E2AXXXSP, M37703E2BXXXSP
PROM Version of M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP ..... 3—28
M37703E4-XXXSP, M37703E4AXXXSP, M37703E4BXXXSP
PROM Version of M37703M4-XXXSP, M37703M4AXXXSP, M37703M4BXXXSP ..... 3—51

4 APPENDICES

MELPS 7700 Mask ROM Ordering Method ..... 4—3
MELPS 7700 PROM Ordering Method ..... 4—16
Mark Specification Form ..... 4—29

Contact Addresses for Further Information

GUIDANCE 1

INDEX BY FUNCTION

■MELPS 7700 16-BIT MICROCOMPUTERS

Type	Circuit function and organization	Structure	Supply voltage (V)	Electrical characteristics: Typ. power dissipation (mW)	Electrical characteristics: Min. cycle time (ns)	Electrical characteristics: Max. frequency (MHz)	Package	Page
M37702M2-XXXFP	16K-Byte Mask-Prog. ROM, 512-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	30	500	8	80P6N	2—3
M37702M2AXXXFP	16K-Byte Mask-Prog. ROM, 512-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	60	250	16	80P6N	
M37702M2BXXXFP	16K-Byte Mask-Prog. ROM, 512-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	95	160	25	80P6N	
M37702S1FP	External ROM, 512-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	30	500	8	80P6N	
M37702S1AFP	External ROM, 512-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	60	250	16	80P6N	
M37702S1BFP	External ROM, 512-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	95	160	25	80P6N	
M37702M4-XXXFP	32K-Byte Mask-Prog. ROM, 2048-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	30	500	8	80P6N	2—62
M37702M4AXXXFP	32K-Byte Mask-Prog. ROM, 2048-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	60	250	16	80P6N	
M37702M4BXXXFP	32K-Byte Mask-Prog. ROM, 2048-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	95	160	25	80P6N	
M37702S4FP	External ROM, 2048-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	30	500	8	80P6N	
M37702S4AFP	External ROM, 2048-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	60	250	16	80P6N	
M37702S4BFP	External ROM, 2048-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	95	160	25	80P6N	
M37703M2-XXXSP	16K-Byte Mask-Prog. ROM, 512-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	30	500	8	64P4B	2—65
M37703M2AXXXSP	16K-Byte Mask-Prog. ROM, 512-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	60	250	16	64P4B	
M37703M2BXXXSP	16K-Byte Mask-Prog. ROM, 512-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	95	160	25	64P4B	
M37703S1SP	External ROM, 512-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	30	500	8	64P4B	
M37703S1ASP	External ROM, 512-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	60	250	16	64P4B	
M37703S1BSP	External ROM, 512-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	95	160	25	64P4B	
M37703M4-XXXSP	32K-Byte Mask-Prog. ROM, 2048-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	30	500	8	64P4B	2—84
M37703M4AXXXSP	32K-Byte Mask-Prog. ROM, 2048-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	60	250	16	64P4B	
M37703M4BXXXSP	32K-Byte Mask-Prog. ROM, 2048-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	95	160	25	64P4B	
M37703S4SP	External ROM, 2048-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	30	500	8	64P4B	

■MELPS 7700 16-BIT MICROCOMPUTERS (Continue)

Type	Circuit function and organization	Structure	Supply voltage (V)	Electrical characteristics			Package	Page
				Typ. power dissipation (mW)	Min. cycle time (ns)	Max. frequency (MHz)		
M37703S4ASP	External ROM, 2048-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	60	250	16	64P4B	2—84
M37703S4BSP	External ROM, 2048-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	95	160	25	64P4B	
M37720S1FP	External ROM, 512-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O, DMA/DRAM Controller	C, Si	5±10%	30	500	8	100P6S	2—86
M37720S1AFP	External ROM, 512-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O, DMA/DRAM Controller	C, Si	5±10%	60	250	16	100P6S	
M37730S2FP	External ROM, 1024-Byte RAM 16-Bit Timer, Serial I/O	C, Si	5±10%	30	500	8	64P6N	2—179
M37730S2AFP	External ROM, 1024-Byte RAM 16-Bit Timer, Serial I/O	C, Si	5±10%	60	250	16	64P6N	
M37730S2BFP	External ROM, 1024-Byte RAM 16-Bit Timer, Serial I/O	C, Si	5±10%	95	160	25	64P6N	
M37730S2SP	External ROM, 1024-Byte RAM 16-Bit Timer, Serial I/O	C, Si	5±10%	30	500	8	64P4B	
M37730S2ASP	External ROM, 1024-Byte RAM 16-Bit Timer, Serial I/O	C, Si	5±10%	60	250	16	64P4B	
M37730S2BSP	External ROM, 1024-Byte RAM 16-Bit Timer, Serial I/O	C, Si	5±10%	95	160	25	64P4B	
M37732S4FP	External ROM, 2048-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	30	500	8	80P6N	2—237
M37732S4AFP	External ROM, 2048-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	60	250	16	80P6N	
M37732S4BFP	External ROM, 2048-Byte RAM 16-Bit Timer, 8-Bit A-D Converter Two Serial I/O	C, Si	5±10%	95	160	25	80P6N	

■PROGRAMMABLE ROM MICROCOMPUTERS

Type	Circuit function and organization	Structure	Supply voltage (V)	Electrical characteristics			Package	Page
				Typ. power dissipation (mW)	Min. cycle time (ns)	Max. frequency (MHz)		
M37702E2-XXXFP	One time PROM Version of M37702M2-XXXFP 16K-Byte PROM, 512-Byte RAM	C, Si	5±10%	30	500	8	80P6N	3—3
M37702E2AXXXFP	One time PROM Version of M37702M2AXXXFP 16K-Byte PROM, 512-Byte RAM	C, Si	5±10%	60	250	16	80P6N	
M37702E2BXXXFP	One time PROM Version of M37702M2BXXXFP 16K-Byte PROM, 512-Byte RAM	C, Si	5±10%	95	160	25	80P6N	
M37702E2FS	EPROM Version of M37702M2-XXXFP 16K-Byte EPROM, 512-Byte RAM	C, Si	5±10%	30	500	8	80D0	
M37702E2AFS	EPROM Version of M37702M2AXXXFP 16K-Byte EPROM, 512-Byte RAM	C, Si	5±10%	60	250	16	80D0	
M37702E2BFS	EPROM Version of M37702M2BXXXFP 16K-Byte EPROM, 512-Byte RAM	C, Si	5±10%	95	160	25	80D0	
M37702E4-XXXFP	One time PROM Version of M37702M4-XXXFP 32K-Byte PROM, 2048-Byte RAM	C, Si	5±10%	30	500	8	80P6N	3—26
M37702E4AXXXFP	One time PROM Version of M37702M4AXXXFP 32K-Byte PROM, 2048-Byte RAM	C, Si	5±10%	60	250	16	80P6N	
M37702E4BXXXFP	One time PROM Version of M37702M4BXXXFP 32K-Byte PROM, 2048-Byte RAM	C, Si	5±10%	95	160	25	80P6N	
M37702E4FS	EPROM Version of M37702M4-XXXFP 32K-Byte EPROM, 2048-Byte RAM	C, Si	5±10%	30	500	8	80D0	
M37702E4AFS	EPROM Version of M37702M4AXXXFP 32K-Byte EPROM, 2048-Byte RAM	C, Si	5±10%	60	250	16	80D0	
M37702E4BFS	EPROM Version of M37702M4BXXXFP 32K-Byte EPROM, 2048-Byte RAM	C, Si	5±10%	95	160	25	80D0	
M37703E2-XXXSP	One time PROM Version of M37703M2-XXXSP 16K-Byte PROM, 512-Byte RAM	C, Si	5±10%	30	500	8	64P4B	3—28
M37703E2AXXXSP	One time PROM Version of M37703M2AXXXSP 16K-Byte PROM, 512-Byte RAM	C, Si	5±10%	60	250	16	64P4B	
M37703E2BXXXSP	One time PROM Version of M37703M2BXXXSP 16K-Byte PROM, 512-Byte RAM	C, Si	5±10%	95	160	25	64P4B	
M37703E4-XXXSP	One time PROM Version of M37703M4-XXXSP 32K-Byte PROM, 2048-Byte RAM	C, Si	5±10%	30	500	8	64P4B	3—51
M37703E4AXXXSP	One time PROM Version of M37703M4AXXXSP 32K-Byte PROM, 2048-Byte RAM	C, Si	5±10%	60	250	16	64P4B	
M37703E4BXXXSP	One time PROM Version of M37703M4BXXXSP 32K-Byte PROM, 2048-Byte RAM	C, Si	5±10%	95	160	25	64P4B	

Development support systems for MELPS 7700

MELPS 7700 type name	Assembler	Emulation pod	For evaluate
M37702M2-XXXFP M37702M2AXXXFP M37702S1FP M37702S1AFP M37702E2-XXXFP M37702E2AXXXFP M37702E2FS M37702E2AFS	RASM77	M37702T-HPD	M37702E2FS M37702E2AFS M37702E2BFS M37702E4FS M37702E4AFS M37702E4BFS
M37702M4-XXXFP M37702M4AXXXFP M37702S4FP M37702S4AFP M37702E4-XXXFP M37702E4AXXXFP M37702E4FS M37702E4AFS		M37702T-HPD (Exchange MCU for M37702S4AFP)	
M37702M2BXXXFP M37702S1BFP M37702E2BXXXFP M37702E2BFS		M37702TB-HPD★★	
M37702M4BXXXFP M37702S4BFP M37702E4BXXXFP M37702E4BFS		M37702TB-HPD★★ (Exchange MCU for M37702S4BFP)	
M37703M2-XXXSP M37703M2AXXXSP M37703S1SP M37703S1ASP M37703E2-XXXSP M37703E2AXXXSP		M37702T-HPD	
M37703M4-XXXSP M37703M4AXXXSP M37703S4SP M37703S4ASP M37703E4-XXXSP M37703E4AXXXSP		M37702T-HPD (Exchange MCU for M37702S4AFP)	M37703E2-XXXSP M37703E2AXXXSP M37703E2BXXXSP M37703E4-XXXSP M37703E4AXXXSP M37703E4BXXXSP
M37703M2BXXXSP M37703S1BSP M37703E2BXXXSP		M37702TB-HPD★★	
M37703M4BXXXSP M37703S4BSP M37703E4BXXXSP		M37702TB-HPD★★ (Exchange MCU for M37702S4BFP)	
M37720S1FP M37720S1AFP		M37720T-HPD★★	—
M37730S2FP M37730S2AFP M37730S2SP M37730S2ASP		M37730T-HPD★★	—
M37730S2BFP M37730S2BSP		M37730TB-HPD★★	
M37732S4FP M37732S4AFP		M37732T-HPD★★	—
M37732S4BFP		M37732TB-HPD★★	

★★ Under development

MITSUBISHI ELECTRIC

Program writing adapter for built-in PROM type microcomputers of MELPS 7700

Built-in PROM type microcomputers type name	Program writing adapter
M37702E2-XXXFP M37702E2AXXXFP M37702E2BXXXFP	PCA4774
M37702E2FS M37702E2AFS M37702E2BFS	PCA4708
M37702E4-XXXFP M37702E4AXXXFP M37702E4BXXXFP	PCA4774
M37702E4FS M37702E4AFS M37702E4BFS	PCA4708
M37703E2-XXXSP M37703E2AXXXSP M37703E2BXXXSP	PCA4709
M37703E4-XXXSP M37703E4AXXXSP M37703E4BXXXSP	

ORDERING INFORMATION

FUNCTION CODE

Mitsubishi integrated circuit may be ordered using the following simplified alphanumeric type-codes which define the function of the IC/LSIs and the package style.

For Mitsubishi Original Products

Example : **M 3 77 02 E 4 - 001 FP**

- **M** : Mitsubishi integrated prefix
- **3** : Represent an original single-chip microcomputer
- Series designation using 2 digits
- Circuit function identification code using 2 digits
- Memory identification code using a digit
  - **E** : PROM
  - **P** : Piggyback
  - **M** : Mask ROM
  - **S** : External ROM
  - **N** : Mask ROM+EEPROM
- Memory size identification code using a digit
- Normally, using hyphen.
  When electrical characteristic, or division of quality identification code using alphanumeric character.
  T : For automobile, industrial equipment
- Mask ROM number
- Package style
  - **J** : PLCC, or SOJ package
  - **FP** : Molded plastic flat package
  - **FS** : Ceramic flat package
  - **SP** : Molded plastic shrink package
  - **SS** : Ceramic shrink package

PACKAGE CODE

Package style may be specified by using the following simplified alphanumeric code.

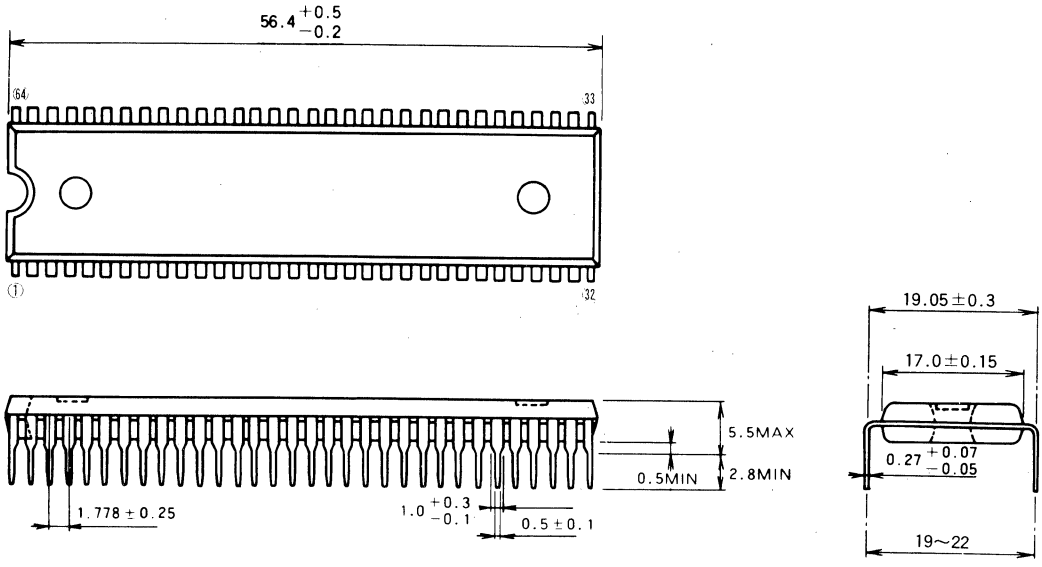
Example : **64 P 4 B**

- Number of pins
- Package structure
  - **K** : Glass-sealed ceramic
  - **P** : Molded plastic
  - **S** : Metal-sealed ceramic
- Package outline
  - **0** : PLCC, SOJ
  - **4** : DIP
  - **1** : DIP
  - **6** : QFP
  - **2** : SOP
- Secondary outline code
  Special-purpose secondary codes describing outline are included as necessary. For details, contact your sales representative.

PACKAGE OUTLINES

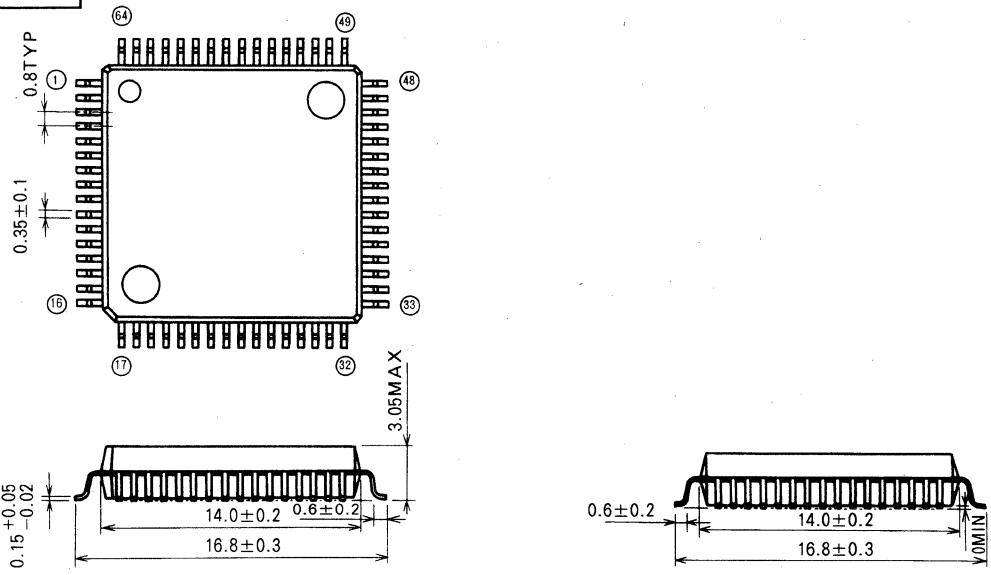
64P4B Plastic 64pin 750mil SDIP

Dimension in mm



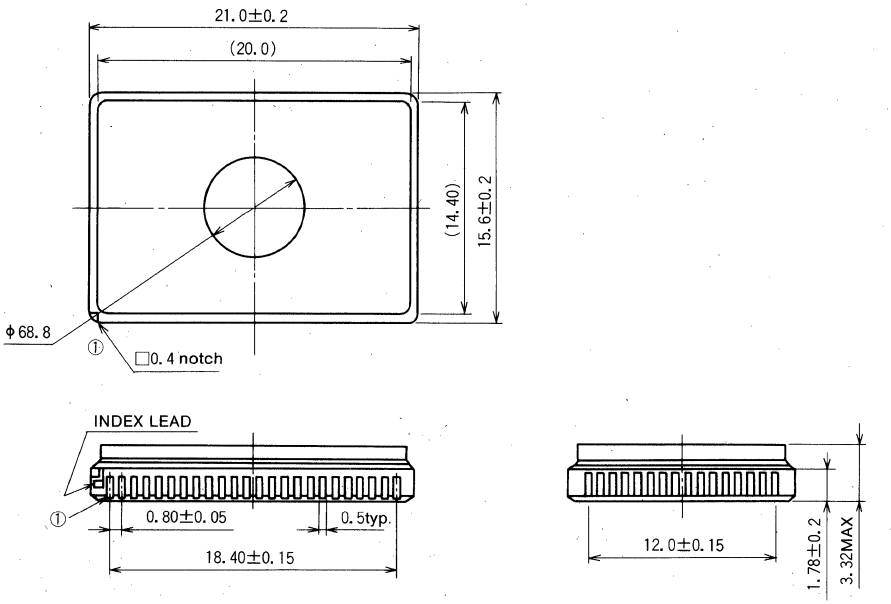
64P6N Plastic 64pin QFP

Dimension in mm



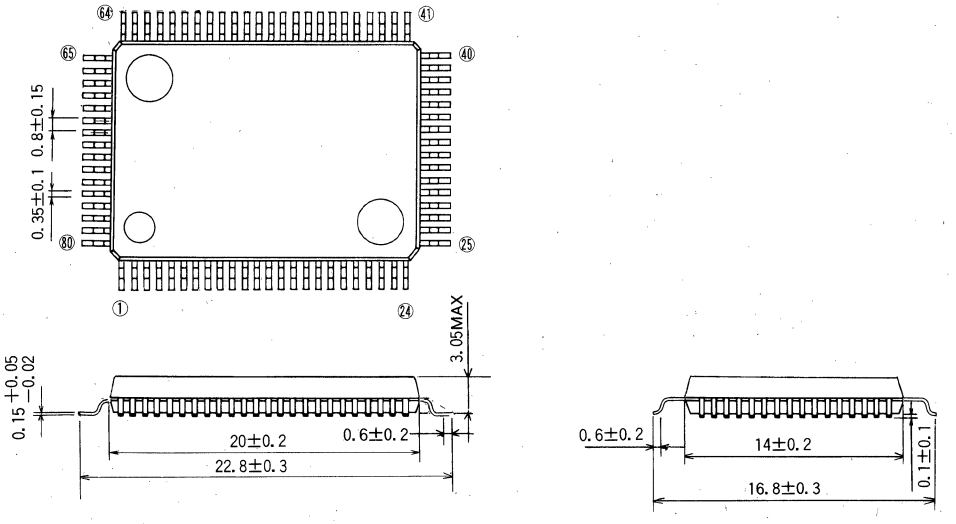
80D0 Ceramic 80pin LCC

Dimension in mm



80P6N Plastic 80pin QFP

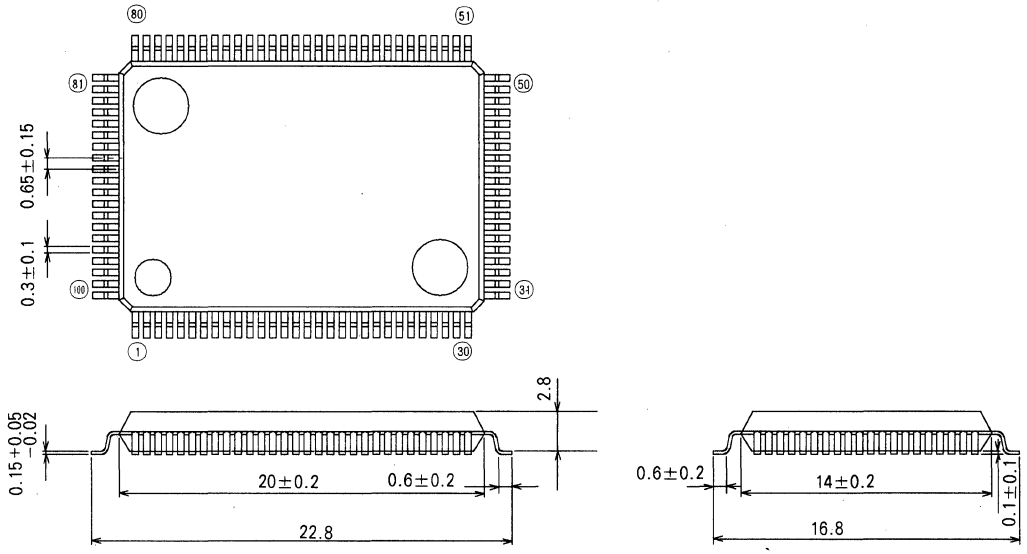
Dimension in mm



MITSUBISHI ELECTRIC

100P6S Plastic 100pin QFP

Dimension in mm



LETTER SYMBOLS FOR THE DYNAMIC PARAMETERS

1. INTRODUCTION

A system of letter symbols to be used to represent the dynamic parameters of intergrated circuit memories and other sequential circuits especially for single-chip microcomputers, microprocessors and LSIs for peripheral circuits has been discussed internationally in the TC47 of the International Electrotechnical Committee (IEC). Finally the IEC has decided on the meeting of TC47 in February 1980 that this system of letter symbols will be a Central Office document and circulated to all countries to vote which means this system of letter symbols will be a international standard.

The system is applied in this LSI data book for the new products only. Future editions of this data book will be applied this system. The IEC document which describes "Letter symbols for dynamic parameters of sequential integrated circuits, including memories" is introduced below. In this data book, the dynamic parameters in the IEC document are applied to timing requirements and switching characteristics.

2. LETTER SYMBOLS

The system of letter symbols outlined in this document enables symbols to be generated for the dynamic parameters of complex sequential circuits, including memories, and also allows these symbols to be abbreviated to simple mnemonic symbols when no ambiguity is likely to arise.

2.1. General Form

The dynamic parameters are represented by the general symbol of the form:-

$$t_{A(BC-DC)F} \quad \text{..... (1)}$$

where :

Subscript A indicates the type of dynamic parameter being represented, for example; cycle time, setup time, enable time, etc.

Subscript B indicates the name of the signal or terminal for which a change of state or level (or establishment of a state or level) constitutes a signal event assumed to occur first, that is, at the beginning of the time interval. If this event actually occurs last, that is, at the end of the time interval, the value of the time interval is negative.

Subscript C indicates the direction of the transition and/or the final state or level of the signal represented by B. When two letters are used, the initial state or level is also indicated.

Subscript D indicates the name of the signal or terminal for which a change of state or level (or establishment of a state or level) constitutes a signal event assumed to occur last, that is, at the end of the time interval. If this event actually occurs first, that is, at the beginning of the time interval, the value of the time interval is negative.

Subscript E indicates the direction of the transition and/or the final state or level of the signal represented by D. When two letters are used, the initial state or level is also indicated.

Subscript F indicates additional information such as mode of operation, test conditions, etc.

Note 1: Subscripts A to F may each consists of one or more letters.
2: Subscripts D and E are not used for transition times.
3: The "–" in the symbol (1) above is used to indicate "to", hence the symbol represents the time interval from signal event B occuring to signal event D occuring, and it is important to note that this convention is used for all dynamic parameters including hold times. Where no misunderstanding can occur the hyphen may be omitted.

2.2. Abbreviated Form

The general symbol given above may be abbreviated when no misunderstanding is likely to arise. For example to :

$t_{A(B-D)}$

or $t_{A(B)}$

or $t_{A(D)}$ – often used for hold times

or t_{AF} – no brackets are used in this case

or t_{A}

or t_{BC-DE} – often used for unclassified time intervals

2.3. Allocation of Subscripts

In allocating letter symbols for the subscripts, the most commonly used subscripts are given single letters where practicable and those less commonly used are designated by up to three letters. As far as possible, some form of mnemonic representation is used. Longer letter symbols may be used for specialised signals or terminals if this aids understanding.

3. SUBSCRIPT A (For Type of Dynamic Parameter)

The subscript A represents the type of dynamic parameter to be designated by the symbol and, for memories, the parameters may be divided into two classes :

a) those that are timing requirements for the memory and

LETTER SYMBOLS FOR THE DYNAMIC PARAMETERS

b) those that are characteristics of the memory.
The letter symbols so far proposed for memory circuits are listed in sub-clauses 3.1 and 3.2 below.
All subscripts A should be in lower-case.

3.1. Timing Requirements

The letter symbols for the timing requirements of semiconductor memories are as follows:

Term	Subscript
Cycle time	c
Time interval between two signal events	d
Fall time	f
Hold time	h
Precharging time	pc
Rise time	r
Recovery time	rec
Refresh time interval	rf
Setup time	su
Transition time	t
Pulse duration (width)	w

3.2. Characteristics

The letter symbols for the dynamic characteristics of semiconductor memories are as follows :

Characteristic	Subscript
Access time	a
Disable time	dis
Enable time	en
Propagation time	P
Recovery time	rec
Transition time	T
Valid time	v

Note: Recovery time for use as a characteristic is limited to sense recovery time.

4. SUBSCRIPTS B AND D (For Signal Name or Terminal Name)

The letter symbols for the signal name or the name of the terminal are as given below.
All subscripts B and D should be in upper-case.

Signal or terminal	Subscript
Address	A
Clock	C
Column address	CA
Column address strobe	CAS
Data input	D
Data input/output	DQ
Chip enable	E
Erasure	ER
Output enable	G
Program	PR
Data output	Q
Read	R
Row address	RA
Row address strobe	RAS
Refresh	RF
Read/Write	RW
Chip select	S
Write (write enable)	W

Note 1. In the letter symbols for time intervals, bars over the subscripts, for example CAS, should not be used.
2. It should be noted, when further letter symbols are chosen, that the subscript should not end with H, K, V, X, or Z. (See clause 5)
3. If the same terminal, or signal, can be used for two functions (for example Data input/output, Read/Write) the waveform should be labelled with the dual function, if appropriate, but the symbols for the dynamic parameters should include only that part of the subscript relevant to the parameter.

5. SUBSCRIPTS C AND E (For Transition of Signal)

The following symbols are used to represent the level or state of a signal :

Transition of signal	Subscript
High logic level	H
Low logic level	L
Valid steady-state level (either low or high)	V
Unknown, changing, or 'don't care' level	X
High-impedance state of three-state output	Z

The direction of transition is expressed by two letters, the direction being from the state represented by the first letter to that represented by the second letter, with the letters being as given above.
When no misunderstanding can occur, the first letter may be omitted to give an abbreviated symbol for subscripts C and E as indicated below.
All subscripts C and E should be in upper-case.

Examples	Subscript	
	Full	Abbreviated
Transition from high level to low level	HL	L
Transition from low level to high level	LH	H
Transition from unknown or changing state to valid state	XV	V
Transition from valid state to unknown or changing state	VX	X
Transition from high-impedance state to valid state	ZV	V

Note: Since subscripts C and E may be abbreviated, and since subscripts B and D may contain an indeterminate number of letters, it is necessary to put the restriction on the subscripts B and D that they should not end with H, L, V, X, or Z, so as to avoid possible confusion.

6. SUBSCRIPT F (For Additional Information)

If necessary, subscript F is used to represent any additional qualification of the parameter such as mode of operation, test conditions, etc. The letter symbols for subscript F are given below.
Subscript F should be in upper-case.

Modes of operation	Subscript
Power-down	PD
Page-mode read	PGR
Page-mode write	PGW
Read	R
Refresh	RF
Read-modify-write	RMW
Read-write	RW
Write	W

FOR DIGITAL INTEGRATED CIRCUITS

New symbol	Former symbol	Parameter—definition
C_I		Input capacitance
C_O		Output capacitance
$C_{I/O}$		Input/output terminal capacitance
$C_{I(\phi)}$		Input capacitance of clock input
f		Frequency
$f_{(\phi)}$		Clock frequency
I		Current—the current into an integrated circuit terminal is defined as a positive value and the current out of a terminal is defined as a negative value
I_{BB}		Supply current from V_{BB}
$I_{BB(AV)}$		Average supply current from V_{BB}
I_{CC}		Supply current from V_{CC}
$I_{CC(AV)}$		Avarage supply current from V_{CC}
$I_{CC(PD)}$		Power-down supply current from V_{CC}
I_{DD}		Supply current from V_{DD}
$I_{DD(AV)}$		Average supply current from V_{DD}
I_{GG}		Supply current from V_{GG}
$I_{GG(AV)}$		Average supply current from V_{GG}
I_I		Input current
I_{IH}		High-level input current—the value of the input current when V_{OH} is applied to the input considered
I_{IL}		Low-level input current—the value of the input current when V_{OL} is applied to the input considered
I_{OH}		High-level output current—the value of the output current when V_{OH} is applied to the output considered
I_{OL}		Low-level output current—the value of the output current when V_{OL} is applied to the output considered
I_{OZ}		Off-state(high-impedance state)output current—the current into an output having a three-state capability with input condition so applied that it will establish according to the product specification, the off(high-impedance)state at the output
I_{OZH}		Off-state(high-impedance state)output current, with high-level voltage applied to the output
I_{OZL}		Off-state(high-impedance state)output current, with low-level voltage applied to the output
I_{OS}		Short-circuit output current
I_{SS}		Supply current from V_{SS}
Pd		Power dissipation
N_{EW}		Number of erase/write cycles
N_{RA}		Number of read access unrefreshed
R_I		Input resistance
R_L		External load resistance
R_{OFF}		Off-state output resistance
R_{ON}		On-state output resistance
t_a		Access time—the time interval between the application of a specified input pulse during a read cycle and the availability of valid data signal at an output
$t_{a(A)}$	$t_{a(AD)}$	Address access time—the time interval between the application of an address input pulse and the availability of valid data signals at an output
$t_{a(CAS)}$		Column address strobe access time
$t_{a(E)}$	$t_{a(CE)}$	Chip enable access time
$t_{a(G)}$	$t_{a(OE)}$	Output enable access time
$t_{a(PR)}$		Data access time after program
$t_{a(RAS)}$		Row address strobe access time
$t_{a(S)}$	$t_{a(CS)}$	Chip select access time
t_C		Cycle time
t_{CR}	$t_{C(RD)}$	Read cycle time—the time interval between the start of a read cylce and the start of the next cycle
t_{CRF}	$t_{C(REF)}$	Refresh cycle time—the time interval between successive signals that are intended to restore the level in a dynamic memory cell to its original level
t_{CPG}	$t_{C(PG)}$	Page-mode cycle time
t_{CRMW}	$t_{C(RMW)}$	Read-modify-write cycle time—the time interval between the start of a cycle in which the memory is read and new data is entered, and the start of the next cycle
t_{CW}	$t_{C(WR)}$	Write cycle time—the time interval between the start of a write cycle and the start of the next cycle

New symbol	Former symbol	Parameter—definition
t_d		Delay time—the time between the specified reference points on two pulses
$t_{d(\phi)}$		Delay time between clock pulses—e.g., symbology, delay time, clock 1 to clock 2 or clock 2 to clock 1
$t_{d(CAS\text{-}RAS)}$		Delay time, column address strobe to row address strobe
$t_{d(CAS\text{-}W)}$	$t_{d(CAS\ WR)}$	Delay time, column address strobe to write
$t_{d(RAS\text{-}CAS)}$		Delay time, row address strobe to column address strobe
$t_{d(RAS\text{-}W)}$	$t_{d(RAS\text{-}WR)}$	Delay time, row address strobe to write
$t_{dis(R\text{-}O)}$	$t_{dis(R\text{-}DA)}$	Output disable time after read
$t_{dis(S)}$	$t_{PXZ(CS)}$	Output disable time after chip select
$t_{dis(W)}$	$t_{PXZ(WR)}$	Output disable time after write
t_{DHL}		High-level to low-level delay time } the time interval between specified reference points on the input and on the output pulses when the output is
t_{DLH}		Low-level to high-level delay time } going to the low (high)level and when the device is driven with a specified loading networks
$t_{en(A\text{-}Q)}$	$t_{PZV(A\text{-}DQ)}$	Output enable time after address
$t_{en(R\text{-}Q)}$	$t_{PZV(R\text{-}DQ)}$	Output enable time after read
$t_{en(S\text{-}Q)}$	$t_{PZX(CS\text{-}DQ)}$	Output enable time after chip select
t_f		Fall time
t_h		Hold time—the interval of time during which a signal at a specified input terminal appears after an active transition occurs at another specified input terminal
$t_{h(A)}$	$t_{h(AD)}$	Address hold time
$t_{h(A\text{-}E)}$	$t_{h(AD\text{-}CE)}$	Chip enable hold time after address
$t_{h(A\text{-}PR)}$	$t_{h(AD\text{-}PRO)}$	Program hold time after address
$t_{h(CAS\text{-}CA)}$		Column address hold time after column address strobe
$t_{h(CAS\text{-}D)}$	$t_{h(CAS\text{-}DA)}$	Data-in hold time after column address strobe
$t_{h(CAS\text{-}Q)}$	$t_{h(CAS\text{-}OUT)}$	Data-out hold time after column address strobe
$t_{h(CAS\text{-}RAS)}$		Row address strobe hold time after column address strobe
$t_{h(CAS\text{-}W)}$	$t_{h(CAS\text{-}WR)}$	Write hold time after column address strobe
$t_{h(D)}$	$t_{h(DA)}$	Data-in hold time
$t_{h(D\text{—}PR)}$	$t_{h(DA\text{-}PRO)}$	Program hold time after data-in
$t_{h(E)}$	$t_{h(CE)}$	Chip enable hold time
$t_{h(E\text{-}D)}$	$t_{h(CE\text{-}DA)}$	Data-in hold time after chip enable
$t_{h(E\text{-}G)}$	$t_{h(CE\text{-}OE)}$	Output enable hold time after chip enable
$t_{h(R)}$	$t_{h(RD)}$	Read hold time
$t_{h(RAS\text{-}CA)}$		Column address hold time after row address strobe
$t_{h(RAS\text{-}CAS)}$		Column address strobe hold time after row address strobe
$t_{h(RAS\text{-}D)}$	$t_{h(RAS\text{-}DA)}$	Data-in hold time after row address strobe
$t_{h(RAS\text{-}W)}$	$t_{h(RAS\text{-}WR)}$	Write hold time after row address strobe
$t_{h(S)}$	$t_{h(CS)}$	Chip select hold time
$t_{h(W)}$	$t_{h(WR)}$	Write hold time
$t_{h(W\text{-}CAS)}$	$t_{h(WR\text{-}CAS)}$	Column address strobe hold time after write
$t_{h(W\text{-}D)}$	$t_{h(WR\text{-}DA)}$	Data-in hold time after write
$t_{h(W\text{-}RAS)}$	$t_{h(WR\text{-}RAS)}$	Row address hold time after write
t_{PHL}		High-level to low-level propagation time } the time interval between specified reference points on the input and on the output pulses when the
t_{PLH}		Low-level to high-level propagation time } output is going to the low (high)level and when the device is driven and loaded by typical devices of stated type
t_r		Rise time
$t_{rec(W)}$	t_{wr}	Write recovery time—the time interval between the termination of a write pulse and the initiation of a new cycle
$t_{rec(PD)}$	$t_{R(PD)}$	Power-down recovery time
t_{su}		Setup time—the time interval between the application of a signal which is maintained at a speciifed input terminal and a consecutive active tarnsition at another specified input terminal
$t_{su(A)}$	$t_{su(AD)}$	Address setup time
$t_{su(A\text{-}E)}$	$t_{su(AD\text{-}CE)}$	Chip enable setup time before address
$t_{su(A\text{-}W)}$	$t_{su(AD\text{-}WR)}$	Write setup time before address
$t_{su(CA\text{-}RAS)}$		Row address strobe setup time before column address

New symbol	Former symbol	Parameter—definition
$t_{SU(D)}$	$t_{SU(DA)}$	Data-in setup time
$t_{SU(D-E)}$	$t_{SU(DA-CE)}$	Chip enable setup time before data-in
$t_{SU(D-W)}$	$t_{SU(DA-WR)}$	Write setup time before data-in
$t_{SU(E)}$	$t_{SU(CE)}$	Chip enable setup time
$t_{SU(E-P)}$	$t_{SU(CE-P)}$	Precharge setup time before chip enable
$t_{SU(G-E)}$	$t_{SU(OE-CE)}$	Chip enable setup time before output enable
$t_{SU(P-E)}$	$t_{SU(P-CE)}$	Chip enable setup time before precharge
$t_{SU(PD)}$		Power-down setup time
$t_{SU(R)}$	$t_{SU(RD)}$	Read setup time
$t_{SU(R-CAS)}$	$t_{SU(RA-CAS)}$	Column address strobe setup time before read
$t_{SU(RA-CAS)}$		Cloumn address strobe setup time before row address
$t_{SU(S)}$	$t_{SU(CS)}$	Chip select setup time
$t_{SU(S-W)}$	$t_{SU(CS-WR)}$	Write setup time before chip select
$t_{SU(W)}$	$t_{SU(WR)}$	Write setup time
t_{THL}		High-level to low-level transition time } the time interval between specified reference points on the edge of the output pulse when the output is going to the low(high)level and when a specified input signal is applied through a specified network and the output is loaded by another specified network
t_{TLH}		Low-level to high-level transition time }
$t_{V(A)}$	$t_{dV(AD)}$	Data valid time after address
$t_{V(E)}$	$t_{dV(CE)}$	Data valid time after chip enable
$t_{V(E)PR}$	$t_{V(CE)PR}$	Data valid time after chip enable in program mode
$t_{V(G)}$	$t_{V(OE)}$	Data valid time after output enable
$t_{V(PR)}$		Data valid time after program
$t_{V(S)}$	$t_{V(CS)}$	Data valid time after chip select
t_W		Pulse width (pulse duration)the time interval between specified reference points on the leading and training edges of the waveforms
$t_{W(E)}$	$t_{W(CE)}$	Chip enable pulse width
$t_{W(EH)}$	$t_{W(CEH)}$	Chip enable high pulse width
$t_{W(EL)}$	$t_{W(EL)}$	Chip enable low pulse width
$t_{W(PR)}$		Program pulse width
$t_{W(R)}$	$t_{W(RD)}$	Read pulse width
$t_{W(S)}$	$t_{W(CS)}$	Chip select pulse width
$t_{W(W)}$	$t_{W(WR)}$	Write pulse width
$t_{W(\phi)}$		Clock pulse width
T_a		Ambient temperature
T_{opr}		Operating temperature
T_{stg}		Storage temperature
V_{BB}		V_{BB} supply voltage
V_{CC}		V_{CC} supply voltage
V_{DD}		V_{DD} supply voltage
V_{GG}		V_{GG} supply voltage
V_I		Input voltage
V_{IH}		High-level input voltage—the value of the permitted high-state voltage at the input
V_{IL}		Low-level input voltage—the value of the permitted low-state voltage at the input
V_O		Output voltage
V_{OH}		High-level output voltage—the value of the guaranteed high-state voltage range at the output
V_{OL}		Low-level output voltage—the value of the guaranteed low-state voltage range at the output
V_{SS}		V_{SS} supply voltage

QUALITY ASSURANCE AND RELIABILITY TESTING

1 INTRODUCTION

IC & LSI have made rapid technical progress in electrical performances of high integration, high speed, and sophisticated functionality. And now they have got boundless wider applications in electronic systems and electrical appliances.

To meet the above trend of expanding utilization of IC & LSI, Mitsubishi considers that it is extremely important to supply stable quality and high reliable products to customers.

Mitsubishi Electric places great emphasis on quality as a basic policy "Quality First", and has striven always to improve quality and reliability.

Mitsubishi has already developed the Quality Assurance System covering design, manufacturing, inventory and delivery for IC & LSI, and has supplied highly reliable products to customers for many years. The following articles describe the Quality Assurance System and examples of reliability control for Mitsubishi Single-chip 16-bit Microcomputer.

2. QUALITY ASSURANCE SYSTEM

The Quality Assurance System places emphasis on built-in reliability in designing and built-in quality in manufacturing. The System from development to delivery is summarized in Figure 1.

2.1 Quality Assurance in Designing

The following steps are applied in the designing stage for a new product.

(1) Setting of perfomance, quality and reliability target for new product.
(2) Discussion of performance and quality for circuit design, device structure, process, material and package.
(3) Verification of design by CAD system to meet standardized design rule.
(4) Functional evaluation for bread-board device to confirm electrical performance.
(5) Reliability evaluation for TEG (Test Element Group) chip to detect basic failure mode and investigate failure mechanism.
(6) Reliability test (In-house qualification) for new product to confirm quality and reliability target.
(7) Decision of pre-production from the standpoint of performance, reliability, production flow/conditions, production capability, delivery etc.

2.2 Quality Assurance in Manufacturing

Quality assurance in manufacturing is performed as follows.

(1) Environment control such as temperature, humidity and dust as well as deionized water and utility gases.
(2) Maintenance and calibration control for automatized manufacturing equipments, automatic testing equipments, and measuring instruments.
(3) Material control such as silicon wafer, lead frame, packaging material, mask and chemicals.
(4) In-process inspections in wafer-fabrication, assembly and testing.
(5) 100% final inspection of electrical characteristics, visual inspection and burn-in, if necessary.
(6) Quality assurance test
-Electrical characteristics and visual inspection, lot by lot sampling
-Environment and endurance test, periodical sampling.
(7) Inventory and shipping control, such as storage environment, date code identification, handling and ESD (Electro Static Discharge) preventive procedure.

2.3 Reliability Test

To verify the reliability of a product as described in the Mitsubishi Quality Assurance System, reliability tests are performed at three different stages in new product development, pre-production and mass-production.

At the development of a new product the reliability test plan is fixed corresponding to the quality and reliability target of each product, respectively. The test plan includes in-house qualification test and TEG evaluation, if necessary. TEG chips are designed and prepared for new device structure, new process and new material.

After the proto-type product has passed the in-house qualification test, the product advances to the pre-production. In the pre-production stage, the specific reliability tests are programmed and performed again to verify the quality of pre-production product.

In the mass production, the reliability tests are performed periodically to confirm the quality of the mass production product according to the quality assurance test program.

Table 1 shows an example of reliability test program for plastic encapsulated IC & LSI.

Table 1 TYPICAL RELIABILITY TEST PROGRAM FOR PLASTIC ENCAPSULATED IC & LSI

Group	Test	Test condition
1	Solderability	230℃, 5sec. Rosin flux
2	Soldering heat	260℃, 10sec.
	Thermal shock	−55℃, 125℃, 15cycles
	Temperature cycling	−65℃, 150℃, 100cycles
3	Lead fatigue	250gr, 90°, 2arcs
4	Shock	1500G, 0.5msec.
	Vibration	20G, 100~2000Hz X, Y, Z direction 4min./cycle, 4cycles/direction
	Constant acceleration	20000G, Y direction, 1min.
5	Operation life	T_a=125℃, Vccmax 1000hours
6	High temperature storage life	T_a=150℃, 1000hours
7	High temperature and high humidity	85℃, 85%, 1000hours
	Pressure cooker	121℃, 100%, 100hours

QUALITY ASSURANCE AND RELIABILITY TESTING

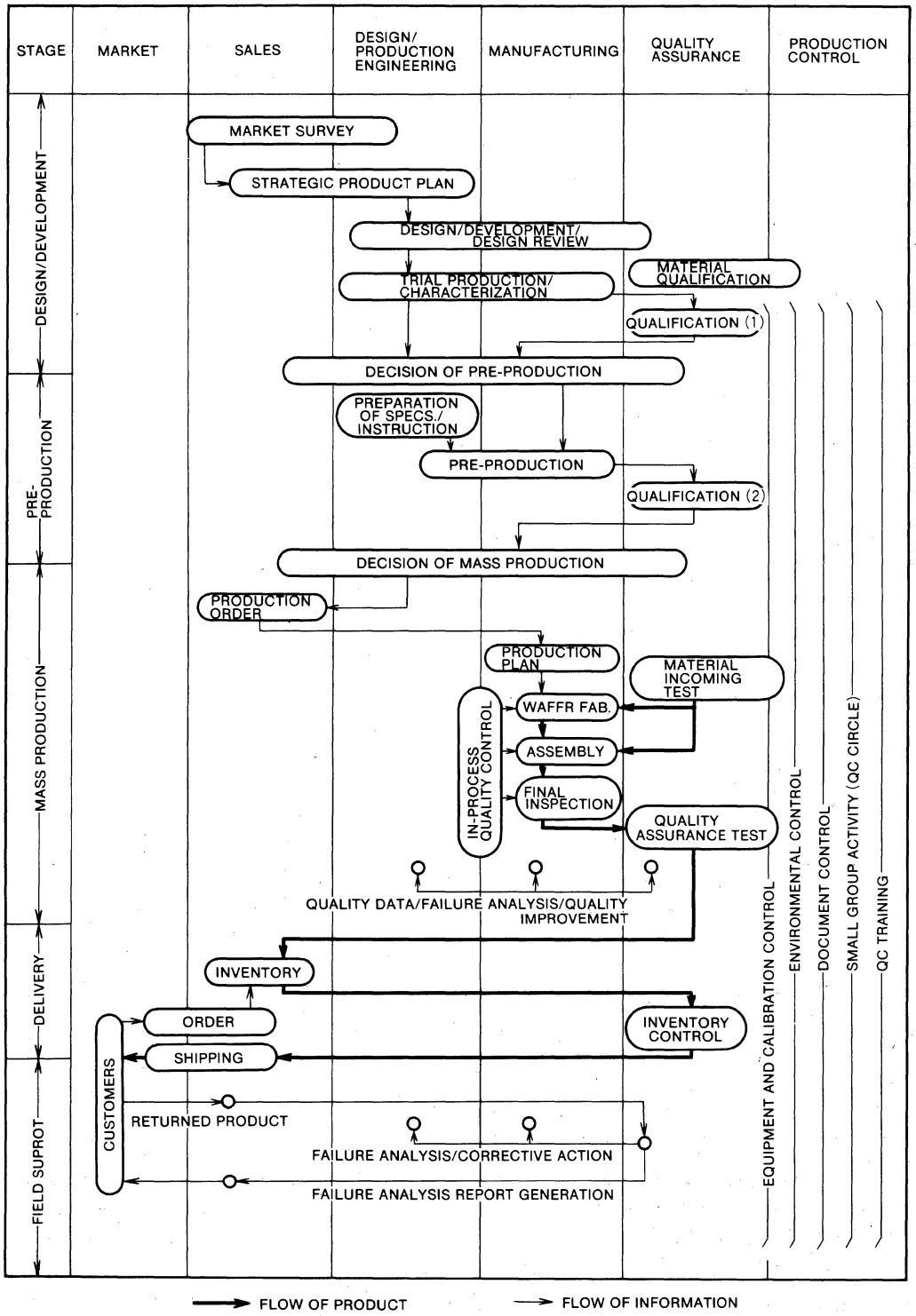


Fig.1 FLOW CHART OF QUALITY ASSURANCE SYSTEM

2.4 Returned Product Control

When failure analysis is requested by a customer, the failed devices are returned to Mitsubishi Electric via the sales office of Mitsubishi using the form of "Analysis Request of Returned Product"

Mitsubishi provides various failure analysis equipments to analyze the returned product. A failure analysis report is generated to the customer upon completion of the analysis. The failure analysis result enforces taking corrective action for the design, fabrication, assembly or testing of the product to improve reliability and realize lower failure rate.

Figure 2 shows the procedure of returned product control from customer.

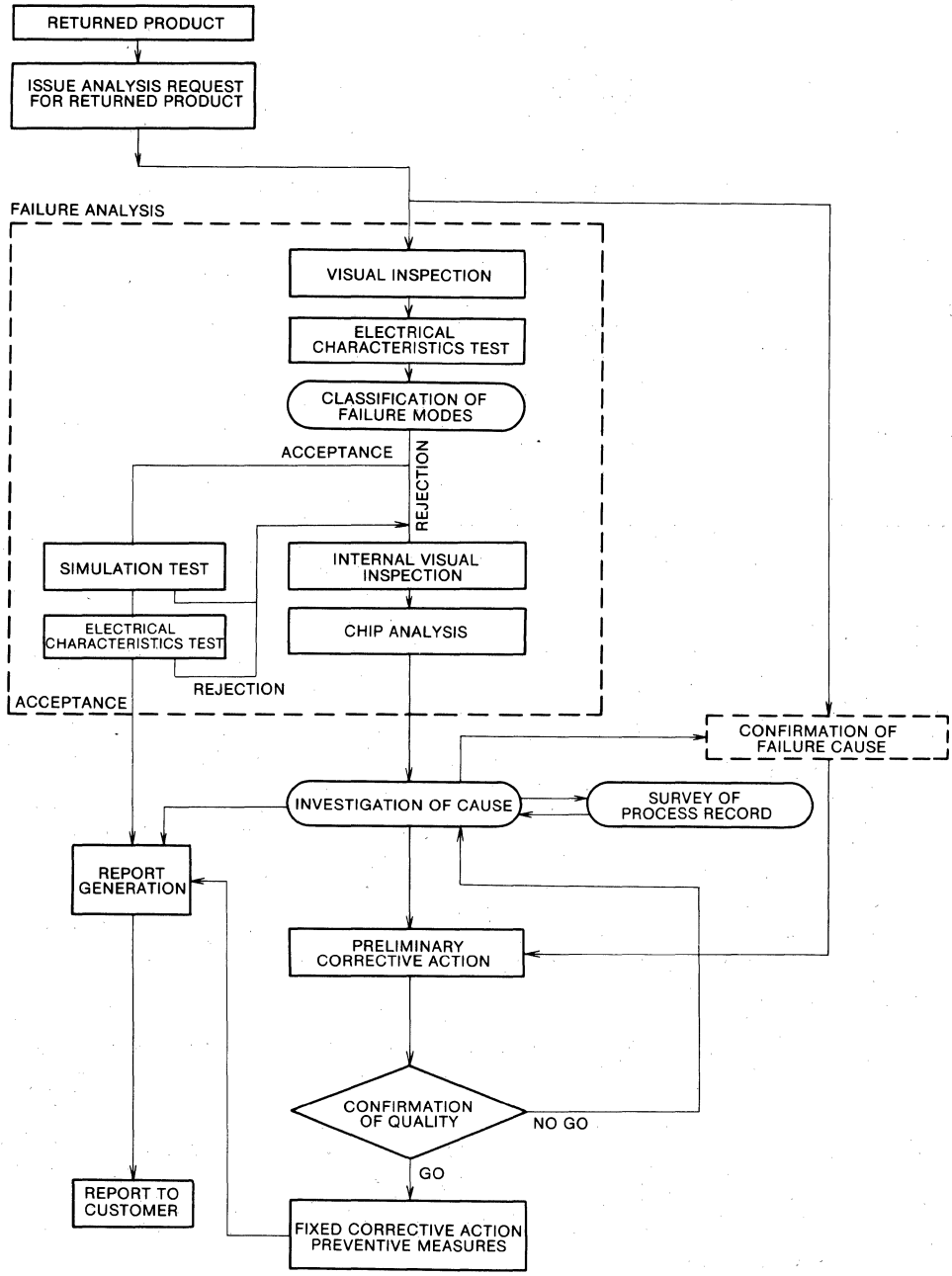


Fig.2 PROCEDURE OF RETURNED PRODUCT CONTROL

3 RELIABILITY TEST RESULTS

The reliability test results for Mitsubishi Single-chip 16-bit Microcomputers are shown in Table 2.
Table 2 shows the result of endurance tests of high temperature operation life and high temperature storage life test and the results of the environment tests of thermal stress, high temperature/high humidity and pressure cooker test for the single-chip 16-bit Microcomputer.

Table 2 ENDURANCE and ENVIRONMENTAL TEST RESULTS

Test	Series	Type Number	Test Condition	Number of Samples	Device Hours (Hours)	Number of Failures
High Temperature Operation Life	MELPS 7700	M37702M2BXXXFP M37702M4BXXXFP M37703M4BXXXSP M37720S1AFP M37730S2BFP M37730S2BSP M37732S4BFP M37702E2AXXXFP M37703E4AXXXSP	125℃ 7 V	198	200000	0
High Temperature Storage Life	MELPS 7700	ditto	150℃	198	200000	0
Low Temperature Storage Life	MELPS 7700	M37702M2BXXXFP M37702E4AXXXFP	−55℃	44	88000	0
High Temperature High Humidity Life	MELPS 7700	M37702M2BXXXFP M37702M4BXXXFP M37703M4BXXXSP M37720S1AFP M37730S2BFP M37730S2BSP M37732S4BFP M37702E2AXXXFP M37703E4AXXXSP	85℃85%RH 5.5V	198	200000	0

Test	Series	Type Number	Test Condition	Number of Samples	Number of Failures	
					96Hours	240Hours
Pressure Cooker	MELPS 7700	ditto	121℃100%RH	198	0	0

Test	Series	Type Number	Test Condition	Number of Samples	Number of Failures	
					10Cycles	100Cycles
Temperature Cycling	MELPS 7700	ditto	−65℃ 30min 150℃ 30min	198	0	0

4 FAILURE ANALYSIS

Accelerated reliability tests are applied to observe failures casued by temperature, voltage, humidity, current, mechanical stress and those combined stresses on chips and packages.

Examples of typical failure modes are shown below.

(1) Wire Bonding Failure by Thermal Stress

Figure 3, Figure 4 and Figure 5 are examples of a failure occured by high temperature storage test of 225℃, 1000hours.

Au-Al intermetallic formation, so-called "Purple plague" by thermal overstress makes Au wire lift off from aluminum metallization. The activation energy of this failure mode is estimated at approximately 1.0eV and no failure has been observed so far in practical uses.

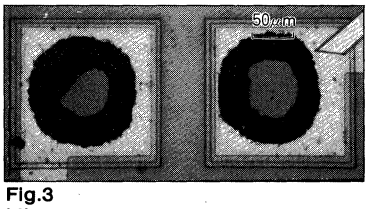


Fig.3
Micrograph of lifted Au ball trace on Al bonding pad

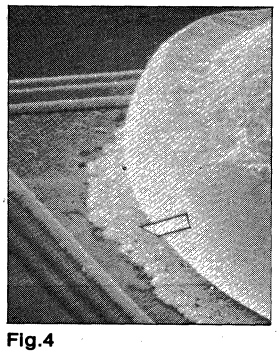


Fig.4
Au-Al plague formation on bonding pad

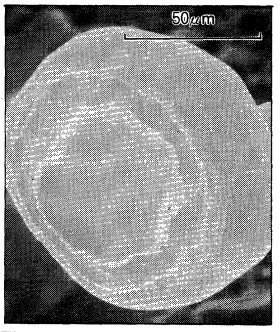


Fig.5
Lifted Au wire ball base

(2) Aluminum Corrosion Failure by Temperature/Humidity Stress.

Figure 6, Figure 7 and Figure 8 are examples of corroded failure of aluminum metallization of plastic encapsulated IC after accelerated temperature/humidity storage test (pressure cooker test) of 121℃, 100%RH, 1000hours duration.

Aluminum bonding pad is dissolved by penetrated water from plastic package, and chlorine concentration is observed on corroded aluminum bonding pad as shown in Figure 8.

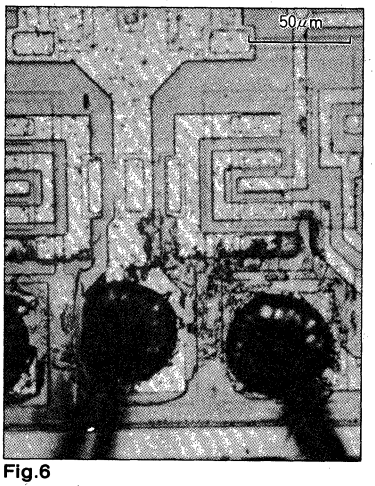


Fig.6
Micrograph of corroded Aluminum metallization

Fig.7 Enlarged micrograph of corroded Aluminum bonding pad

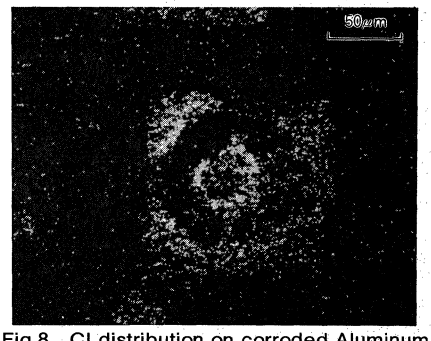


Fig.8 Cl distribution on corroded Aluminum bonding pad

(3) Destructive Failure by Electrical Overstress
Surge voltage marginal tests have been performed to reproduce the electrical overstress failure in field uses. Figure 9 and Figure 10 are examples of failure observed by surge voltage test. The trace of destruction is verified as the aluminum bridge by X-ray micro analysis.

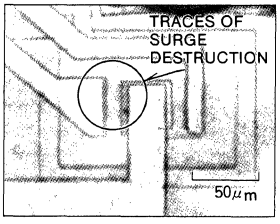


Fig.9 Micrograph of surge voltage destruction

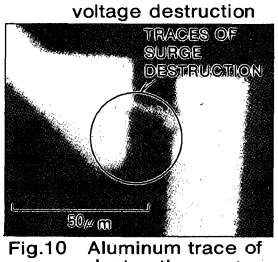


Fig.10 Aluminum trace of destructive spot

(4) Aluminum Electromigration
Figure 11 shows an open circuit of aluminum metallization in high current density region caused by accelerated operation life test. This failure is due to aluminum electromigration. Voids and hillock have been formed in aluminum metallization by high current density operation.

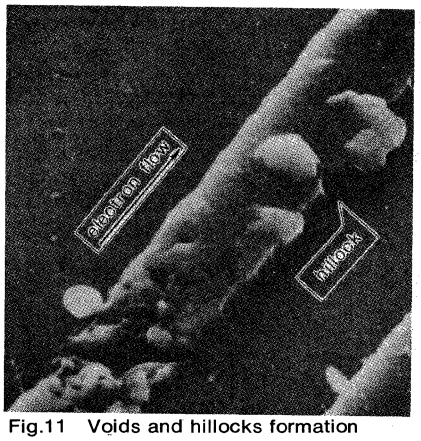


Fig.11 Voids and hillocks formation by Aluminum electromigration

5 SUMMARY

The Mitsubishi quality assurance system and examples of reliability control have been discussed. The customer's interests and requirements for high reliability IC & LSI are increasing significantly. To satisfy customer's expectancy. Mitsubishi as an IC vendor, would like to make perpetual efforts in the following areas.

(1) Emphasis on built-in reliability at design stage and reliability evaluation to investigate latent failure modes and acceleration factors.
(2) Execution of periodical endurance, environment and mechanical test to verify reliability target and realize higher reliability.
(3) Focus on development of advanced failure analysis techniques. Detail failure analysis, intensive corrective action and quick response to customer's analysis request.
(4) Collection of customer's quality data in qualification, incoming inspection, production and field use to improve PPM, fraction defective and FIT, failure rate.

Mitsubishi would highly appreciate if the customer would provide quality and reliability data of incoming inspection or field failure rate essential to verify and improve the quality/reliability of IC & LSI.

PRECAUTIONS IN HANDLING MOS IC/LSIs

A MOS transistor has a very thin oxide insulator under the gate electrode on the silicon substrate. It is operated by altering the conductance (g_m) between source and drain to control mobile charges in the channel formed by the applied gate voltage.

If a high voltage were applied to a gate terminal, the insulator-film under the gate electrode could be destroyed, and all Mitsubishi MOS IC/LSIs contain internal protection circuits at each input terminal to prevent this. It is inherently necessary to apply reverse bias to the P-N junctions of a MOS IC/LSI.

Under certain conditions, however, it may be impossible to completely avoid destruction of the thin insulator-film due to the application of unexpectedly high voltage or thermal destruction due to excessive current from a forward biased P-N junction. Therefore the following recommendations should be followed in handling MOS devices.

1. KEEPING VOLTAGE AND CURRENT TO EACH TERMINAL BELOW MAXIMUM RATINGS

1. The recommended ranges of operating conditions provide adequate safety margins. Operating within these limits will assure maximum equipment performance and quality.
2. Forward bias should not be applied to any terminal since excessive current may cause thermal destruction.
3. Output terminals should not be connected directly to the power supply. Short-circuiting of a terminal to a power supply having low impedance may cause burn-out of the internal leads or thermal destruction due to excessive current.

2. KEEPING ALL TERMINALS AT THE SAME POTENTIAL DURING TRANSPORT AND STORAGE

When MOS IC/LSIs are not in use, both input and output terminals can be in a very high impedance state so that they are easily subjected to electrostatic induction from AC fields of the surrounding space or from charged objects in their vicinity. For this reason, MOS IC/LSIs should be protected from electrostatic charges while being transported and stored by conductive rubber foam, aluminum foil, shielded boxes or other protective precautions.

3. KEEPING ELECTRICAL EQUIPMENT, WORK TABLES AND OPERATING PERSONNEL AT THE SAME POTENTIAL

1. All electric equipment, work table surfaces and operating personnel should be grounded. Work tables should be covered with copper or aluminum plates of good conductivity, and grounded. One method of grounding personnel, after making sure that there is no potential difference with electrical equipment, is by the use of a wristwatch metallic ring, etc. attached around the wrist and grounded in series with a 1M Ω resistor. Be sure that the grounding meets national regulations on personnel safety.
2. Current leakage from electric equipment must be prevented not only for personnel safety, but also to avert the destruction of MOS IC/LSIs, as described above. Items such as testers, curve-tracers and synchroscopes must be checked for current leakage before being grounded.

4. PRECAUTIONS FOR MOUNTING OF MOS IC/LSIs

1. The printed wiring lines between input and output terminals of MOS IC/LSIs should not be close to or parallel to high-voltage or high-power signal lines. Turning power on while the device is short-circuited, either by a solder bridge made during assembly or by a probe during adjusting and testing, may cause maximum ratings to be exceeded, which can result in the destruction of the device.
2. When input/output, or input and/or output, terminals of MOS IC/LSIs (now open-circuits) are connected, we must consider the possibility of current leakage and take precautions similar to §2 above. To reduce such undesirable trouble, it is recommended that an interface circuit be inserted at the input or output terminal, or a resistor with a resistance that does not exceed the output driving capability of the MOS IC/LSI be inserted between the power supply and the ground.
3. A filter circuit should be inserted in the AC power supply line to absorb surges which can frequently be strong enough to destroy a MOS IC/LSI.
4. Terminal connections should be made as described in the catalog while being careful to meet specifications.
5. Ungrounded metal plates should not be placed near input or output terminals of any MOS IC/LSIs, since destruction of the insulation may result if they become electrostatically charged.
6. Equipment cases should provide shielding from electrostatic charges for more reliable operation. When a plastic case is used, it is desirable to coat the inside of the case with conductive paint and to ground it. This is considered necessary even for battery-operated equipment.

MELPS 7700 16-BIT MICROCOMPUTERS

2

M37702M2-XXXFP, M37702M2AXXXFP
M37702M2BXXXFP, M37702S1FP
M37702S1AFP, M37702S1BFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

DESCRIPTION

The M37702M2-XXXFP is a single-chip microcomputers designed with high-performance CMOS silicon gate technology. This is housed in a 80-pin plastic molded QFP. This single-chip microcomputer has a large 16M bytes address space, three instruction queue buffers, and two data buffers for high-speed instruction execution. The CPU is a 16-bit parallel processor that can also be switched to perform 8-bit parallel processing. This microcomputer is suitable for office, business, and industrial equipment controller that require high-speed processing of large data.

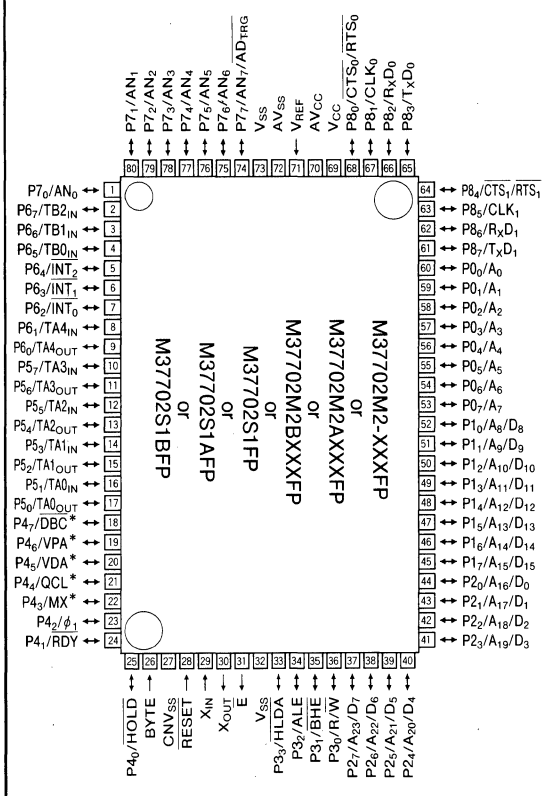
The differences between M37702M2-XXXFP, M37702M2A XXXFP, M37702M2BXXXFP, M37702S1FP, M37702S1AFP and M37702S1BFP are the ROM size and the external clock input frequency as shown below. Therefore, the following descriptions will be for the M37702M2-XXXFP unless otherwise noted.

Type name	ROM size	External clock input frequency
M37702M2-XXXFP	16K bytes	8 MHz
M37702M2AXXXFP	16K bytes	16MHz
M37702M2BXXXFP	16K bytes	25MHz
M37702S1FP	External	8 MHz
M37702S1AFP	External	16MHz
M37702S1BFP	External	25MHz

FEATURES

- Number of basic instructions 103
- Memory size ROM ..... 16K bytes
  RAM ..... 512 bytes
- Instruction execution time
  M37702M2-XXXFP, M37702S1FP
  (The fastest instruction at 8MHz frequency) 500ns
  M37702M2AXXXFP, M37702S1AFP
  (The fastest instruction at 16MHz frequency) 250ns
  M37702M2BXXXFP, M37702S1BFP
  (The fastest instruction at 25MHz frequency) 160ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 8MHz frequency)
  ..... 30mW (Typ.)
- Interrupts ..... 19 types 7 levels
- Multiple function 16-bit timer ..... 5+3
- UART (may also be synchronous) 2
- 8-bit A-D converter ..... 8-channel inputs
- 12-bit watchdog timer
- Programmable input/output
  (ports P0, P1, P2, P3, P4, P5, P6, P7, P8) 68

PIN CONFIGURATION (TOP VIEW)



Outline 80P6N

*: Used in the evaluation chip mode only

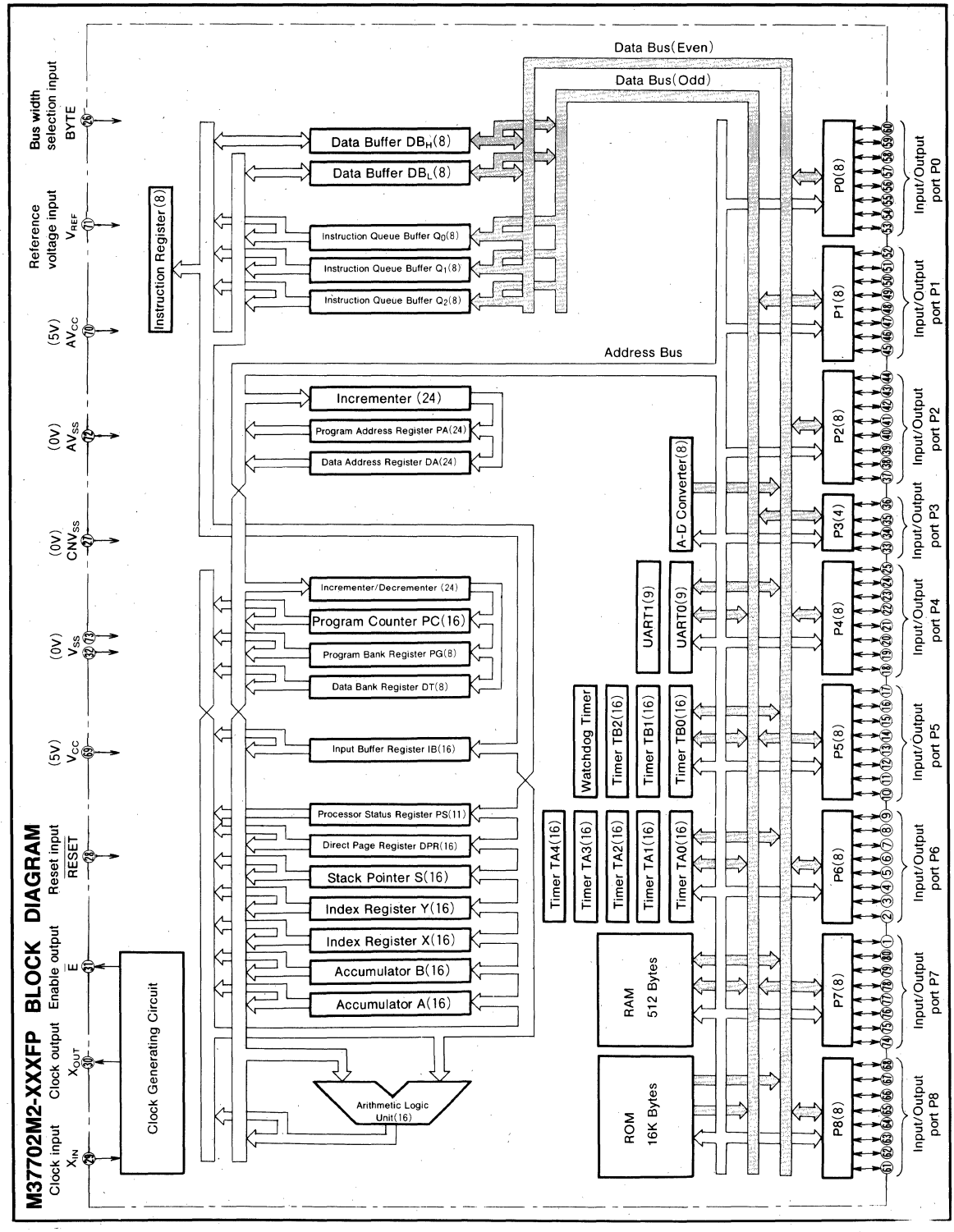
APPLICATION

Control devices for office equipment such as copiers, printers, typewriters, facsimiles, word processors, and personal computers

Control devices for industrial equipment such as ME, NC, communication and measuring instruments.

M37702M2-XXXFP, M37702M2AXXXFP
M37702M2BXXXFP, M37702S1FP
M37702S1AFP, M37702S1BFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER



MITSUBISHI ELECTRIC

M37702M2-XXXFP, M37702M2AXXXFP
M37702M2BXXXFP, M37702S1FP
M37702S1AFP, M37702S1BFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

FUNCTIONS OF M37702M2-XXXFP

Parameter		Functions
Number of basic instructions		103
Instruction execution time	M37702M2-XXXFP, M37702S1FP	500ns (the fastest instructions, at 8MHz frequency)
	M37702M2AXXXFP, M37702S1AFP	250ns (the fastest instructions, at 16MHz frequency)
	M37702M2BXXXFP, M37702S1BFP	160ns (the fastest instructions, at 25MHz frequency)
Memory size	ROM	16K bytes
	RAM	512 bytes
Input/Output ports	P0~P2, P4~P8	8-bit× 8
	P3	4-bit× 1
Multi-function timers	TA0, TA1, TA2, TA3, TA4	16-bit× 5
	TB0, TB1, TB2	16-bit× 3
Serial I/O		(UART or clock synchronous serial I/O)×2
A-D converter		8-bit× 1 (8 channels)
Watchdog timer		12-bit× 1
Interrupts		3 external types, 16 internal types (Each interrupt can be set the priority levels to 0 ~ 7.)
Clock generating circuit		Built-in(externally connected to a ceramic resonator or quartz crystal resonator)
Supply voltage		5 V±10%
Power dissipation		30mW(at external 8 MHz frequency)
Input/Output characteristic	Input/Output voltage	5 V
	Output current	5 mA
Memory expansion		Maximum 16M bytes
Operating temperature range		−20~85℃
Device structure		CMOS high-performance silicon gate process
Package		80-pin plastic molded QFP

PIN DESCRIPTION

Pin	Name	Input/Output	Functions
V_{CC}, V_{SS}	Power supply		Supply 5 V±10% to V_{CC} and 0 V to V_{SS}.
CNV_{SS}	CNV_{SS} input	Input	This pin controls the processor mode. Connect to V_{SS} for single-chip mode, and to V_{CC} for external ROM types.
$\overline{RESET}$	Reset input	Input	To enter the reset state, this pin must be kept at a "L" condition should be maintained for the required time.
X_{IN}	Clock input	Input	These are I/O pins of internal clock generating circuit. Connect a ceramic or quartz crystal resonator between X_{IN} and X_{OUT}. When an external clock is used, the clock source should be connected to the X_{IN} pin and the X_{OUT} pin should be left open.
X_{OUT}	Clock output	Output	
$\overline{E}$	Enable output	Output	Data or instruction read and data write are performed when output from this pin is "L".
BYTE	Bus width selection input	Input	In memory expansion mode or microprocessor mode, this pin determines whether the external data bus is 8-bit width or 16-bit width. The width is 16 bits when "L" signal inputs and 8 bits when "H" signal inputs.
AV_{CC} AV_{SS}	Analog supply input		Power supply for the A-D converter. Connect AV_{CC} to V_{CC} and AV_{SS} to V_{SS} externally.
V_{REF}	Reference voltage input	Input	This is reference voltage input pin for the A-D converter.
$P0_0$~$P0_7$	I/O port P0	I/O	In single-chip mode, port P0 becomes an 8-bit I/O port. An I/O directional register is available so that each pin can be programmed for input or output. These ports are in input mode when reset . Address(A_7~A_0)is output in memory expansion mode or microprocessor mode.
$P1_0$~$P1_7$	I/O port P1	I/O	In single-chip mode, these pins have the same functions as port P0. When the BYTE pin is set to "L" in memory expansion mode or microprocessor mode and external data bus is 16-bit width, high-order data (D_{15}~D_8)is input or output when $\overline{E}$ output is "L" and an address (A_{15}~A_8) is output when $\overline{E}$ output is "H". If the BYTE pin is "H" that is an external data bus is 8-bit width, only address(A_{15}~A_8)is output.
$P2_0$~$P2_7$	I/O port P2	I/O	In single-chip mode, these pins have the same functions as port P0. In memory expansion mode or microprocessor mode low-order data(D_7~D_0) is input or output when $\overline{E}$ output is "L" and an address(A_{23}~A_{16})is output when $\overline{E}$ output is "H".
$P3_0$~$P3_3$	I/O port P3	I/O	In single-chip mode , these pins have the same functions as port P0. In memory expansion mode or microprocessor mode, R/$\overline{W}$, $\overline{BHE}$, ALE, and $\overline{HLDA}$ signals are output.
$P4_0$~$P4_7$	I/O port P4	I/O	In single-chip mode, these pins have the same functions as port P0. In memory expansion mode or microprocessor mode, $P4_0$ and $P4_1$ become $\overline{HOLD}$ and $\overline{RDY}$ input pin respectively. Functions of other pins are the same as in single-chip mode. In single-chip mode or memory expansion mode, port $P4_2$ can be programed for ϕ_1 output pin divided the clock to X_{IN} pin by 2. In microprocessor mode, $P4_2$ always has the function as ϕ_1 output pin.
$P5_0$~$P5_7$	I/O port P5	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as I/O pins for timer A0, timer A1, timer A2 and timer A3.
$P6_0$~$P6_7$	I/O port P6	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as I/O pins for timer A4, external interrupt input $\overline{INT_0}$, $\overline{INT_1}$ and $\overline{INT_2}$ pins, and input pins for timer B0, timer B1 and timer B2.
$P7_0$~$P7_7$	I/O port P7	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as analog input AN_0~AN_7 input pins. $P7_7$ also has an A-D conversion trigger input function.
$P8_0$~$P8_7$	I/O port P8	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as R_XD, T_XD, CLK, $\overline{CTS}$/$\overline{RTS}$ pins for UART 0 and UART 1.

BASIC FUNCTION BLOCKS

The M37702M2-XXXFP contains the following devices on a single chip: ROM and RAM for storing instructions and data, CPU for processing, bus interface unit (which controls instruction prefetch and data read/write between CPU and memory), timers, UART, A-D converter, and other peripheral devices such as I/O ports. Each of these devices are described below.

MEMORY

The memory map is shown in Figure 1. The address space is 16M bytes from addresses 0_{16} to FFFFFF$_{16}$. The address space is divided into 64K bytes units called banks. The banks are numbered from 0_{16} to FF$_{16}$.

Built-in ROM, RAM and control registers for built-in peripheral devices are assigned to bank 0.

The 16K bytes area from addresses C000$_{16}$ to FFFF$_{16}$ is the built-in ROM. Addresses FFD6$_{16}$ to FFFF$_{16}$ are the RESET and interrupt vector addresses and contain the interrupt vectors. Refer to the section on interrupts for details.

The 512 bytes area from addresses 80$_{16}$ to 27F$_{16}$ contains the built-in RAM. In addition to storing data, the RAM is used as stack during a subroutine call, or interrupts.

Assigned to addresses 0_{16} to 7F$_{16}$ are peripheral devices such as I/O ports, A-D converter, UART, timer, and interrupt control registers.

A 256 bytes direct page area can be allocated anywhere in bank 0 using the direct page register DPR. In direct page addressing mode, the memory in the direct page area can be accessed with two words thus reducing program steps.

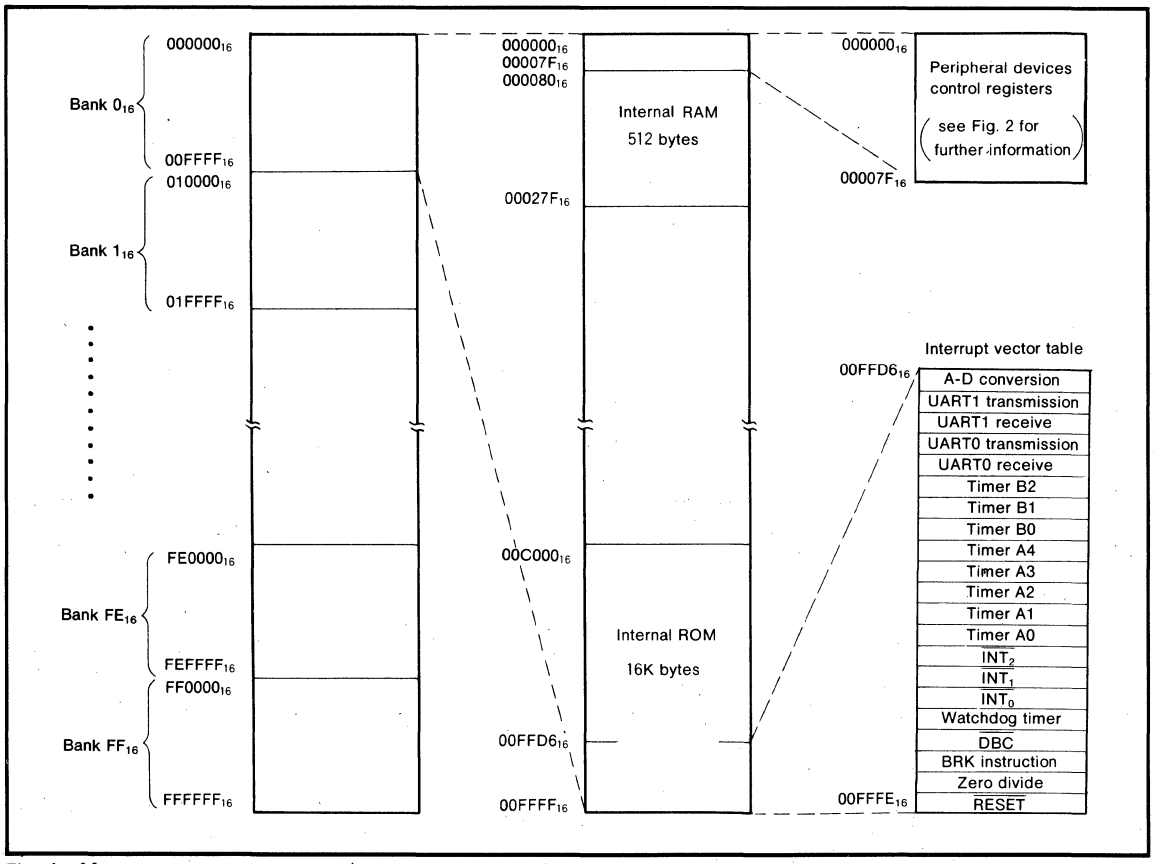


Fig. 1 Memory map

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

Address (Hexadecimal notation)	
000040	Count start flag
000041	
000042	One shot start flag
000043	
000044	Up-down flag
000045	
000046	Timer A0
000047	
000048	Timer A1
000049	
00004A	Timer A2
00004B	
00004C	Timer A3
00004D	
00004E	Timer A4
00004F	
000050	Timer B0
000051	
000052	Timer B1
000053	
000054	Timer B2
000055	
000056	Timer A0 mode register
000057	Timer A1 mode register
000058	Timer A2 mode register
000059	Timer A3 mode register
00005A	Timer A4 mode register
00005B	Timer B0 mode register
00005C	Timer B1 mode register
00005D	Timer B2 mode register
00005E	Processor mode register
00005F	
000060	Watchdog timer
000061	Watchdog timer frequency selection flag
000062	
000063	
000064	
000065	
000066	
000067	
000068	
000069	
00006A	
00006B	
00006C	
00006D	
00006E	
00006F	
000070	A-D conversion interrupt control register
000071	UART0 transmission interrupt control register
000072	UART0 receive interrupt control register
000073	UART1 transmission interrupt control register
000074	UART1 receive interrupt control register
000075	Timer A0 interrupt control register
000076	Timer A1 interrupt control register
000077	Timer A2 interrupt control register
000078	Timer A3 interrupt control register
000079	Timer A4 interrupt control register
00007A	Timer B0 interrupt control register
00007B	Timer B1 interrupt control register
00007C	Timer B2 interrupt control register
00007D	$\overline{INT_0}$ interrupt control register
00007E	$\overline{INT_1}$ interrupt control register
00007F	$\overline{INT_2}$ interrupt control regiter

Fig. 2 Location of peripheral devices and interrupt control registers

CENTRAL PROCESSING UNIT (CPU)

The CPU has ten registers and is shown in Figure 3. Each of these registers is described below.

ACCUMULATOR A (A)

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

Data operations such as calculations, data transfer, input/output, etc., is executed mainly through the accumulator.

ACCUMULATOR B (B)

Accumulator B has the same functions as accumulator A, but the use of accumulator B requires more instruction bytes and execution cycles than accumulator A.

INDEX REGISTER X (X)

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

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

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

INDEX REGISTER Y (Y)

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

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

Also, when executing a block transfer instruction MVP or MVN, the content of index register Y indicates the low-order 16 bits of the destination address. The second byte of the MVP and MVN is the high-order 8 bits of the destination data address.

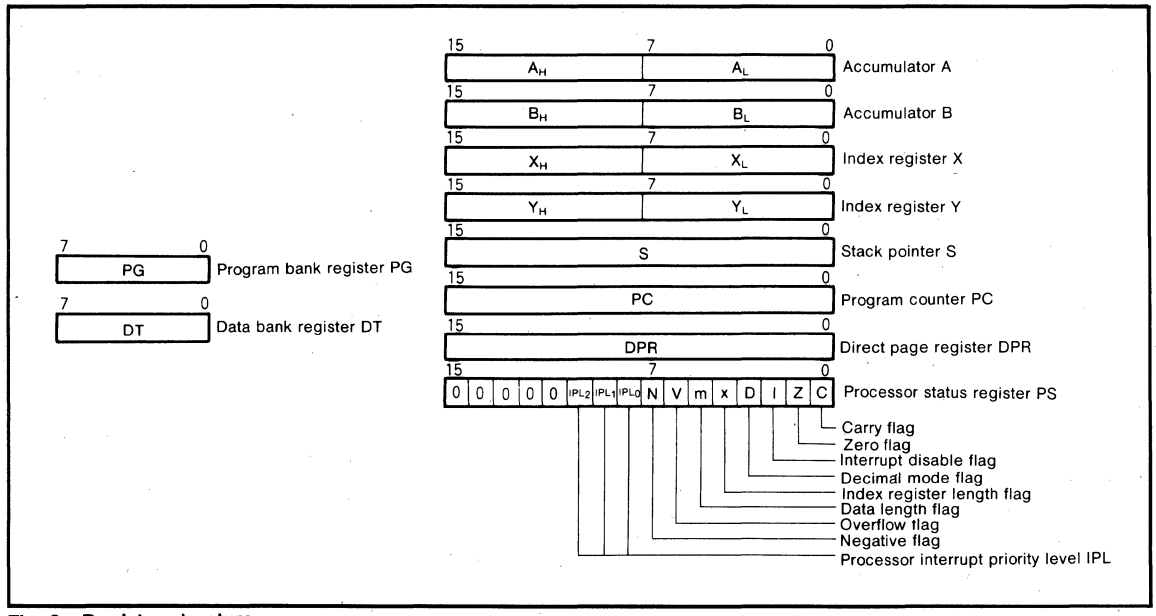


Fig. 3 Register structure

STACK POINTER (S)

Stack pointer (S) is an 16-bit register. It is used during a subroutine call or interrupts. It is also used during stack, stack pointer relative, or stack pointer relative indirect indexed Y addressing mode.

PROGRAM COUNTER (PC)

Program counter (PC) is a 16-bit counter that indicates the low-order 16 bits of the next program memory address to be executed. These is a bus interface unit between the program memory and the CPU, so that the program memory is accessed through bus interface unit. This is described later.

PROGRAM BANK REGISTER (PG)

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

DATA BANK REGISTER (DT)

Data bank register (DT) is an 8-bit register. With some addressing modes, a part of the data bank register (DT) is used to specify a memory address. The contents of data bank register (DT) is used as the high-order 8 bits of a 24-bit address. Addressing modes that use the data bank register (DT) are direct indirect, direct indexed X indirect, direct indirect indexed Y, absolute, absolute bit, absolute indexed X, absolute indexed Y, absolute bit relative, and stack pointer relative indirect indexed Y.

DIRECT PAGE REGISTER (DPR)

Direct page register (DPR) is a 16-bit register. Its contents is used as the base address of a 256-byte direct page area. The direct page area is allocated in bank 0, but when the contents of DPR is $FF01_{16}$ or greater, the direct page area spans across bank 0 and bank 1. All direct addressing modes use the contents of the direct page register (DPR) to generate the data address. If the low-order 8 bits of the direct page register (DPR) is "00_{16}", the number of cycles required to generate an address is minimized. Normally the low-order 8 bits of the direct page register (DPR) is set to "00_{16}".

PROCESSOR STATUS REGISTER (PS)

Processor status register (PS) is an 11-bit register. It consists of a flag to indicate the result of operation and CPU interrupt levels.

Branch operations can be performed by testing the flags C, Z, V, and N.

The details of each processor status register bit are described below.

1. Carry flag (C)

The carry flag contains the carry or borrow generated by the ALU after an arithmetic operation. This flag is also affected by shift and rotate instructions. This flag can be set and reset directly with the SEC and CLC instructions or with the SEP and CLP instructions.

2. Zero flag (Z)

This zero flag is set if the result of an arithmetic operation or data transfer is zero and reset if it is not. This flag can be set and reset directly with the SEP and CLP instructions.

3. Interrupt disable flag (I)

When the interrupt disable flag is set to "1", all interrupts except watchdog timer, $\overline{\text{DBC}}$, and software interrupt are disabled. This flag is set to "1" automatically when there is an interrupt. It can be set and reset directly with the SEI and CLI instructions or SEP and CLP instructions.

4. Decimal mode flag (D)

The decimal mode flag determines whether addition and subtraction are performed as binary or decimal. Binary arithmetic is performed when this flag is "0". If it is "1", decimal arithmetic is performed with each word treated as two or four digit decimal. Arithmetic operation is performed using four digits when the data length flag m is "0" and with two digits when it is "1". (Decimal operation is possible only with the ADC and SBC instructions.) This flag can be set and reset with the SEP and CLP instructions.

5. Index register length flag (x)

The index register length flag determines whether index register X and index register Y are used as 16-bit registers or as 8-bit registers. The registers are used as 16-bit registers when flag x is "0" and as 8-bit registers when it is "1". This flag can be set and reset with the SEP and CLP instructions.

6. Data length flag (m)

The data length flag determines whether the data length is 16-bit or 8-bit. The data length is 16-bit when flag m is "0" and 8-bit when it is "1". This flag can be set and reset with the SEM and CLM instructions or with the SEP and CLP instructions.

7. Overflow flag (V)

The overflow flag has meaning when addition or subtraction is performed a word as signed binary number. When the data length flag m is "0", the overflow flag is set when the result of addition or subtraction is outside the range between −32768 and +32767. When the data length flag m is "1", the overflow flag is set when the result of addition or subtraction is outside the range between −128 and +127. It is reset in all other cases. The overflow flag can also be set and reset directly with the SEP, and CLV or CLP instructions.

8. Negative flag (N)

The negative flag is set when the result of arithmetic operation or data transfer is negative (If data length flag m is "0", when data bit 15 is "1". If data length flag m is "1", when data bit 7 is "1".) It is reset in all other cases. It can also be set and reset with the SEP and CLP instructions.

9. Processor interrupt priority level (IPL)

The processor interrupt priority level (IPL) consists of 3 bits and determines the priority of processor interrupts from level 0 to level 7. Interrupt is enabled when the interrupt priority of the device requesting interrupt (set using the interrupt control register) is higher than the processor interrupt priority. When interrupt is enabled, the current processor interrupt priority level is saved in a stack and the processor interrupt priority level is replaced by the interrupt priority level of the device requesting the interrupt. Refer to the section on interrupts for more details.

BUS INTERFACE UNIT

The CPU operates on an internal clock frequency which is obtained by dividing the external clock frequency $f_{(X_{IN})}$ by two. This frequency is twice the bus cycle frequency. In order to speed-up processing, a bus interface unit is used to pre-fetch instructions when the data bus is idle. The bus interface unit synchronizes the CPU and the bus and prefetches instructions. Figure 4 shows the relationship between the CPU and the bus interface unit. The bus interface unit has a program address register, a 3-byte instruction queue buffer, a data address register, and a 2-byte data buffer.

The bus interface unit obtains an instruction code from memory and stores it in the instruction queue buffer, obtains data from memory and stores it in the data buffer, or writes the data from the data buffer to the memory.

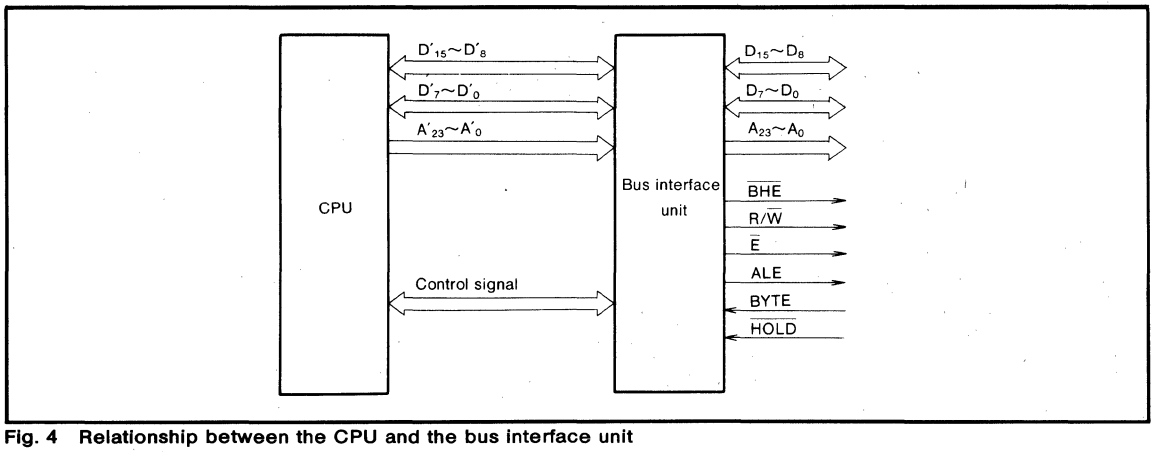


Fig. 4 Relationship between the CPU and the bus interface unit

The bus interface unit operates using one of the waveforms (1) to (6) shown in Figure 5. The standard waveforms are (1) and (2).

The ALE signal is used to latch only the address signal from the multiplexed signal containing data and address.

The $\overline{E}$ signal becomes "L" when the bus interface unit reads an instruction code or data from memory or when it writes data to memory. Whether to perform read or write is controlled by the R/$\overline{W}$ signal. Read is performed when the R/$\overline{W}$ signal is "H" state and write is performed when it is "L" state.

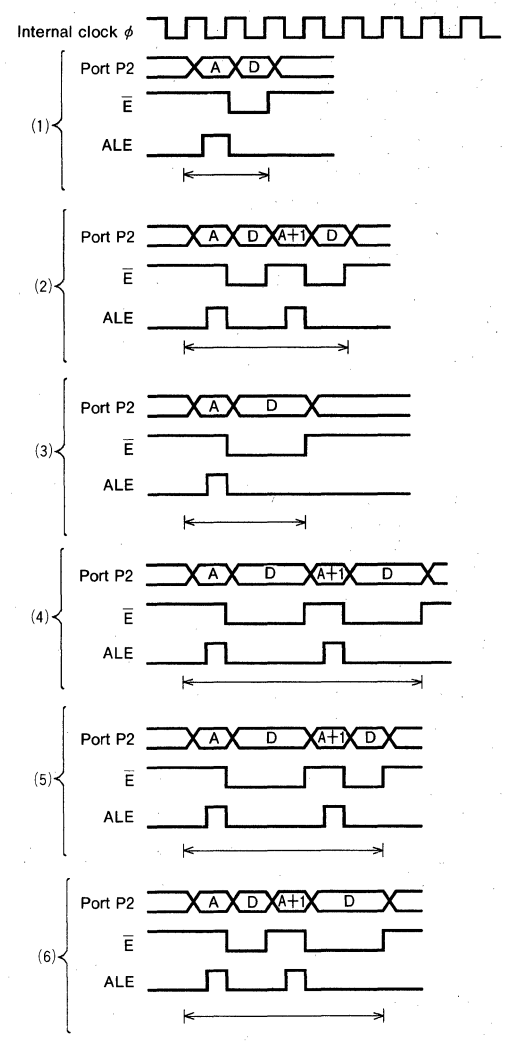
Waveform (1) in Figure 5 is used to access a single byte or two bytes simultaneously. To read or write two bytes simultaneously, the first address accessed must be even. Furthermore, when accessing an external memory area in memory expansion mode or microprocessor mode, set the bus width selection input pin BYTE to "L". (external data bus width to 16 bits) The internal memory area is always treated as 16-bit bus width regardless of BYTE.

When performing 16-bit data read or write, if the conditions for simultaneously accessing two bytes are not satisfied, waveform (2) is used to access each byte one by one. However, when prefetching the instruction code, if the address of the instruction code is odd, waveform (1) is used, and only one byte is read in the instruction queue buffer.

The signals A_0 and $\overline{BHE}$ in Figure 5 are used to control these cases: 1-byte read from even address, 1-byte read from odd address, 2-byte simultaneous read from even and odd addresses, 1-byte write to even address, 1-byte write to odd address, or 2-byte simultaneous write to even and odd addresses. The A_0 signal that is the address bit 0 is "L" when an even number address is accessed. The $\overline{BHE}$ signal becomes "L" when an odd number address is accessed.

The bit 2 of processor mode register (address $5E_{16}$) is the wait bit. When this bit is set to "0", the "L" width of $\overline{E}$ signal is 2 times as long when accessing an external memory area in memory expansion mode or microprocessor mode. However, the "L" width of $\overline{E}$ signal is not extended when an internal memory area is accessed. When the wait bit is "1", the "L" width of $\overline{E}$ signal is not extended for any access. Waveform (3) is an expansion of the "L" width of $\overline{E}$ signal in waveform (1). Waveform (4),(5), and (6) are expansion of each "L" width of $\overline{E}$ signal in waveform (2), first half of waveform (2), and the last half of waveform (2) respectively.

Instruction code read, data read, and data write are described below.



A : Address
D : Data
These waveforms are at the memory expansion mode and the microprocessor mode.

Signal \ Access method	Access 2-byte simultaneously	Access even address 1-byte	Access odd address 1-byte
A_0	"L"	"L"	"H"
$\overline{BHE}$	"L"	"H"	"L"

Fig. 5 Relationship between access method and signals A_0 and $\overline{BHE}$

Instruction code read will be described first.
The CPU obtains instruction codes from the instruction queue buffer and executes them. The CPU notifies the bus interface unit that it is requesting an instruction code during an instruction code request cycle. If the requested instruction code is not yet stored in the instruction queue buffer, the bus interface unit halts the CPU until it can store more instructions than requested in the instruction queue buffer.
Even if there is no instruction code request from the CPU, the bus interface unit reads instruction codes from memory and stores them in the instruction queue buffer when the instruction queue buffer is empty or when only one instruction code is stored and the bus is idle on the next cycle.
This is referred to as instruction pre-fetching.
Normally, when reading an instruction code from memory, if the accessed address is even the next odd address is read together with the instruction code and stored in the instruction queue buffer.
However, in memory expansion mode or microprocessor mode, if the bus width switching pin BYTE is "H", external data bus width is 8 bits and the address to be read is in external memory area is odd, only one byte is read and stored in the instruction queue buffer. Therefore, waveform (1) or (3) in Figure 5 is used for instruction code read.
Data read and write are described below.
The CPU notifies the bus interface unit when performing data read or write. At this time, the bus interface unit halts the CPU if the bus interface unit is already using the bus or if there is a request with higher priority. When data read or write is enabled, the bus interface unit uses one of the waveforms from (1) to (6) in Figure 5 to perform the operation.
During data read, the CPU waits until the entire data is stored in the data buffer. The bus interface unit sends the address received from the CPU to the address bus. Then it reads the memory when the $\overline{E}$ signal is "L" and stores the result in the data buffer.
During data write, the CPU writes the data in the data buffer and the bus interface unit writes it to memory. Therefore, the CPU can proceed to the next step without waiting for write to complete. The bus interface unit sends the address received from the CPU to the address bus. Then when the $\overline{E}$ signal is "L", the bus interface unit sends the data in the data buffer to the data bus and writes it to memory.

INTERRUPTS

Table 1 shows the interrupt types and the corresponding interrupt vector addresses. Reset is also treated as a type of interrupt and is discussed in this section, too.

$\overline{DBC}$ is an interrupt used during debugging.

Interrupts other than reset, $\overline{DBC}$, watchdog timer, zero divide, and BRK instruction all have interrupt control registers. Table 2 shows the addresses of the interrupt control registers and Figure 6 shows the bit configuration of the interrupt control register.

The interrupt request bit is automatically cleared by the hardware during reset or when processing an interrupt. Also, interrupt request bits other than $\overline{DBC}$ and watchdog timer can be cleared by software.

$\overline{INT_2}$ to $\overline{INT_0}$ are external interrupts and whether to cause an interrupt at the input level (level sense) or at the edge (edge sense) can be selected with the level sense/edge sense selection bit. Furthermore, the polarity of the interrupt input can be selected with polarity selection bit.

Timer and UART interrupts are described in the respective section.

The priority of interrupts when multiple interrupts are caused simultaneously is partially fixed by hardware, but, it can also be adjusted by software as shown in Figure 7. The hardware priority is fixed the following:

reset > $\overline{DBC}$ > watchdog timer > other interrupts

Table 1. Interrupt types and the interrupt vector addresses

Interrupts	Vector addresses
A-D conversion	$00FFD6_{16}$ $00FFD7_{16}$
UART1 transmit	$00FFD8_{16}$ $00FFD9_{16}$
UART1 receive	$00FFDA_{16}$ $00FFDB_{16}$
UART0 transmit	$00FFDC_{16}$ $00FFDD_{16}$
UART0 receive	$00FFDE_{16}$ $00FFDF_{16}$
Timer B2	$00FFE0_{16}$ $00FFE1_{16}$
Timer B1	$00FFE2_{16}$ $00FFE3_{16}$
Timer B0	$00FFE4_{16}$ $00FFE5_{16}$
Timer A4	$00FFE6_{16}$ $00FFE7_{16}$
Timer A3	$00FFE8_{16}$ $00FFE9_{16}$
Timer A2	$00FFEA_{16}$ $00FFEB_{16}$
Timer A1	$00FFEC_{16}$ $00FFED_{16}$
Timer A0	$00FFEE_{16}$ $00FFEF_{16}$
$\overline{INT_2}$ external interrupt	$00FFF0_{16}$ $00FFF1_{16}$
$\overline{INT_1}$ external interrupt	$00FFF2_{16}$ $00FFF3_{16}$
$\overline{INT_0}$ external interrupt	$00FFF4_{16}$ $00FFF5_{16}$
Watchdog timer	$00FFF6_{16}$ $00FFF7_{16}$
$\overline{DBC}$ (unusable)	$00FFF8_{16}$ $00FFF9_{16}$
Break instruction	$00FFFA_{16}$ $00FFFB_{16}$
Zero divide	$00FFFC_{16}$ $00FFFD_{16}$
Reset	$00FFFE_{16}$ $00FFFF_{16}$

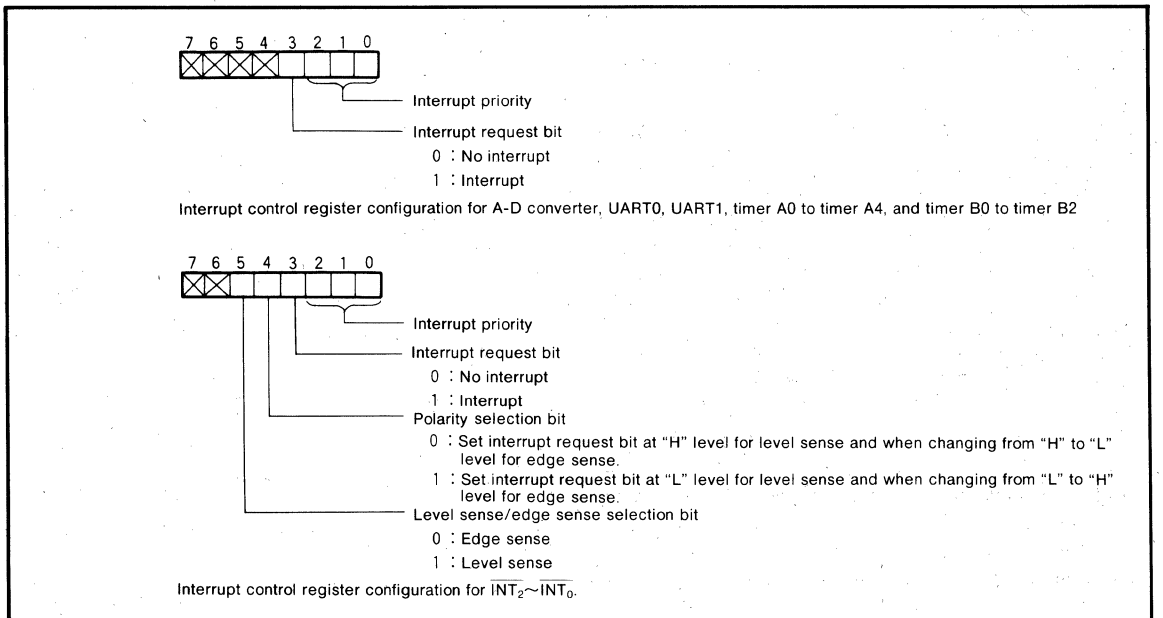


Fig. 6 Interrupt control register configuration

Table 2. Addresses of interrupt control registers

Interrupt control registers	Addresses
A-D conversion interrupt control register	000070_{16}
UART0 transmit interrupt control register	000071_{16}
UART0 receive interrupt control register	000072_{16}
UART1 transmit interrupt control register	000073_{16}
UART1 receive interrupt control register	000074_{16}
Timer A0 interrupt control register	000075_{16}
Timer A1 interrupt control register	000076_{16}
Timer A2 interrupt control register	000077_{16}
Timer A3 interrupt control register	000078_{16}
Timer A4 interrupt control register	000079_{16}
Timer B0 interrupt control register	$00007A_{16}$
Timer B1 interrupt control register	$00007B_{16}$
Timer B2 interrupt control register	$00007C_{16}$
$\overline{INT_0}$ interrupt control register	$00007D_{16}$
$\overline{INT_1}$ interrupt control register	$00007E_{16}$
$\overline{INT_2}$ interrupt control register	$00007F_{16}$

Interrupts caused by a BRK instruction and when dividing by zero are software interrupts and are not included in this list.

Other interrupts previously mentioned are A-D converter, UART, Timer, INT interrupts. The priority of these interrupts can be changed by changing the priority level in the corresponding interrupt control register by software.

Figure 8 shows a diagram of the interrupt priority resolution circuit. When an interrupt is caused, the each interrupt device compares its own priority with the priority from above and if its own priority is higher, then it sends the priority below and requests the interrupt. If the priorities are the same, the one above has priority.

This comparison is repeated to select the interrupt with the highest priority among the interrupts that are being requested. Finally the selected interrupt is compared with the processor interrupt priority level (IPL) contained in the processor status register (PS) and the request is accepted if it is higher than IPL and the interrupt disable flag I is "0". The request is not accepted if flag I is "1". The reset, $\overline{DBC}$, and watchdog timer interrupts are not affected by the interrupt disable flag I.

When an interrupt is accepted, the contents of the processor status register (PS) is saved to the stack and the interrupt disable flag I is set to "1".

Furthermore, the interrupt request bit of the accepted interrupt is cleared to "0" and the processor interrupt priority level (IPL) in the processor status register (PS) is replaced by the priority level of the accepted interrupt.

Therefore, multi-level priority interrupts are possible by resetting the interrupt disable flag I to "0" and enable further interrupts.

For reset, $\overline{DBC}$, watchdog timer, zero divide, and BRK instruction interrupts, which do not have an interrupt control register, the processor interrupt level (IPL) is set as shown in Table 3.

Priority resolution is performed by latching the interrupt request bit and interrupt priority level so that they do not change. They are sampled at the first half and latched at the last half of the operation code fetch cycle.

Because priority resolution takes some time, no sampling pulse is generated for a certain interval even if it is the next operation code fetch cycle.

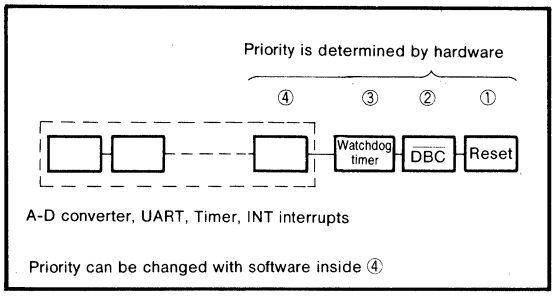


Fig. 7 Interrupt priority

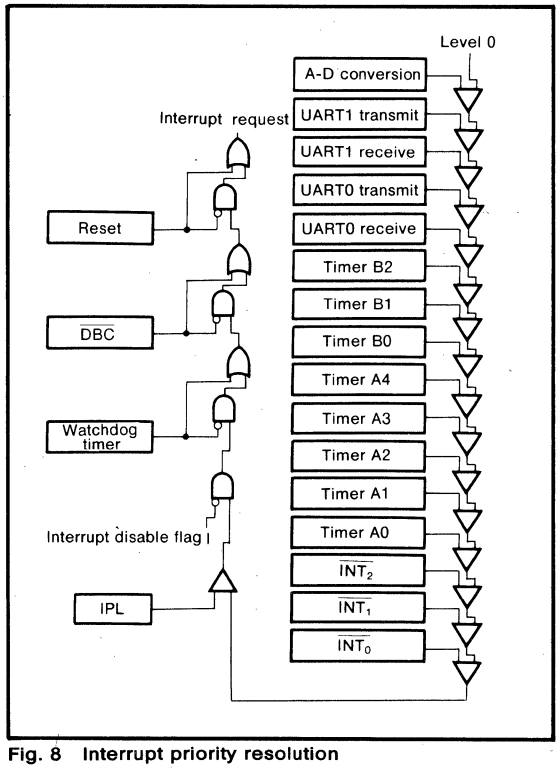


Fig. 8 Interrupt priority resolution

As shown in Figure 9, there are three different interrupt priority resolution time from which one is selected by software. After the selected time has elapsed, the highest priority is determined and is processed after the currently executing instruction has been completed.

The time is selected with bits 4 and 5 of the processor mode register (address $5E_{16}$) shown in Figure 10. Table 4 shows the relationship between these bits and the number of cycles. After a reset, the processor mode register is initialized to "00_{16}" and therefore, the longest time is selected.

However, the shortest time may be selected by software.

Table 3. Value set in processor interrupt level (IPL) during an interrupt

Interrupt types	Setting value
Reset	0
DBC	7
Watchdog timer	7
Zero divide	Not change value of IPL.
BRK instruction	Not change value of IPL.

Table 4. Relationship between priority level evaluation time selection bit and number of cycles

Priority level resolution time selection bit		Number of cycles
Bit 5	Bit 4	
0	0	7 cycles of ϕ
0	1	4 cycles of ϕ
1	0	2 cycles of ϕ

ϕ : internal clock

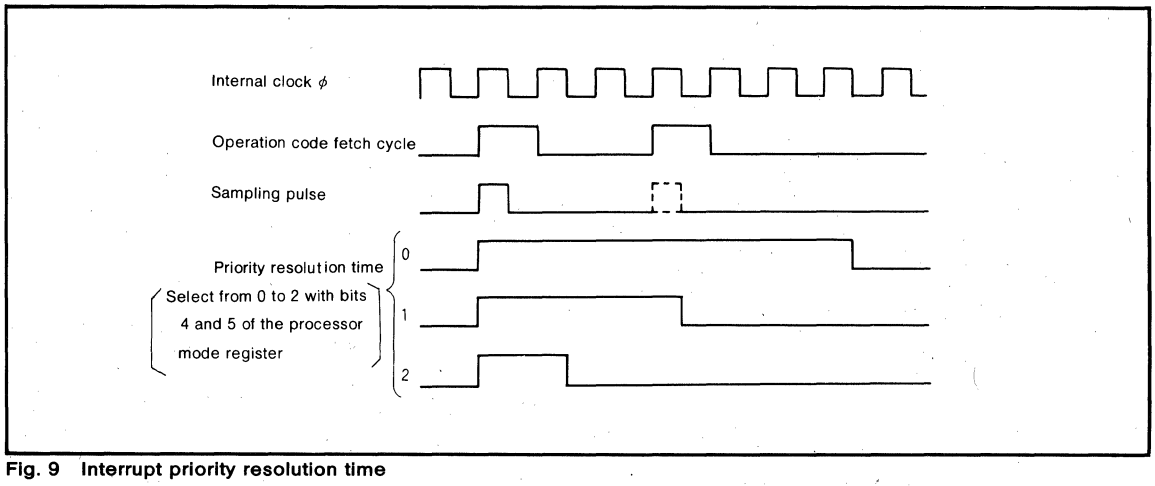


Fig. 9 Interrupt priority resolution time

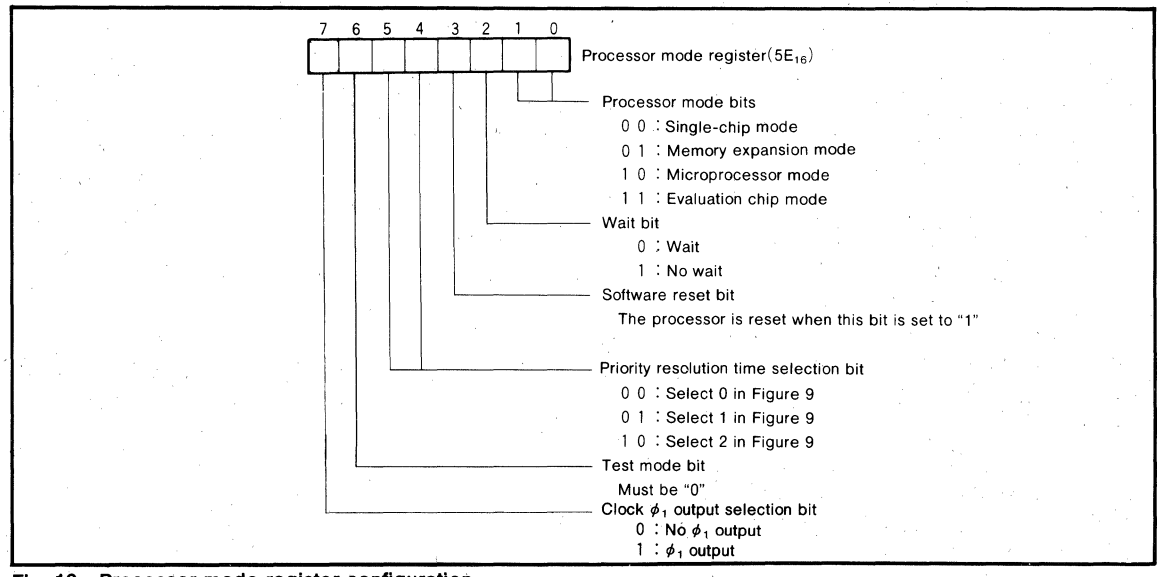


Fig. 10 Processor mode register configuration

TIMER

There are eight 16-bit timers. They are divided by type into timer A(5) and timer B(3).

The timer I/O pins are shared with I/O pins for port P5 and P6. To use these pins as timer input pins, the data direction register bit corresponding to the pin must be cleared to "0" to specify input mode.

Using this timer, confirm the function as this timer is different a little from M37700M2-XXXFP's.

TIMER A

Figure 11 shows a block diagram of timer A.

Timer A has four modes; timer mode, event counter mode, one-shot pulse mode, and pulse width modulation mode. The mode is selected with bits 0 and 1 of the timer Ai mode register (i = 0 to 4). Each of these modes is described below.

(1) Timer mode [00]

Figure 12 shows the bit configuration of the timer Ai mode register during timer mode. Bits 0, 1, and 5 of the timer Ai mode register must always be "0" in timer mode.

Bit 3 is ignored if bit 4 is "0".

Bits 6 and 7 are used to select the timer counter source.

The counting of the selected clock starts when the count start flag is "1" and stops when it is "0".

Figure 13 shows the bit configuration of the count start flag.

The counter is decremented, an interrupt is caused and the interrupt request bit in the timer Ai interrupt control register is set when the contents becomes 0000_{16}. At the same time, the contents of the reload register is transferred to the counter and count is continued.

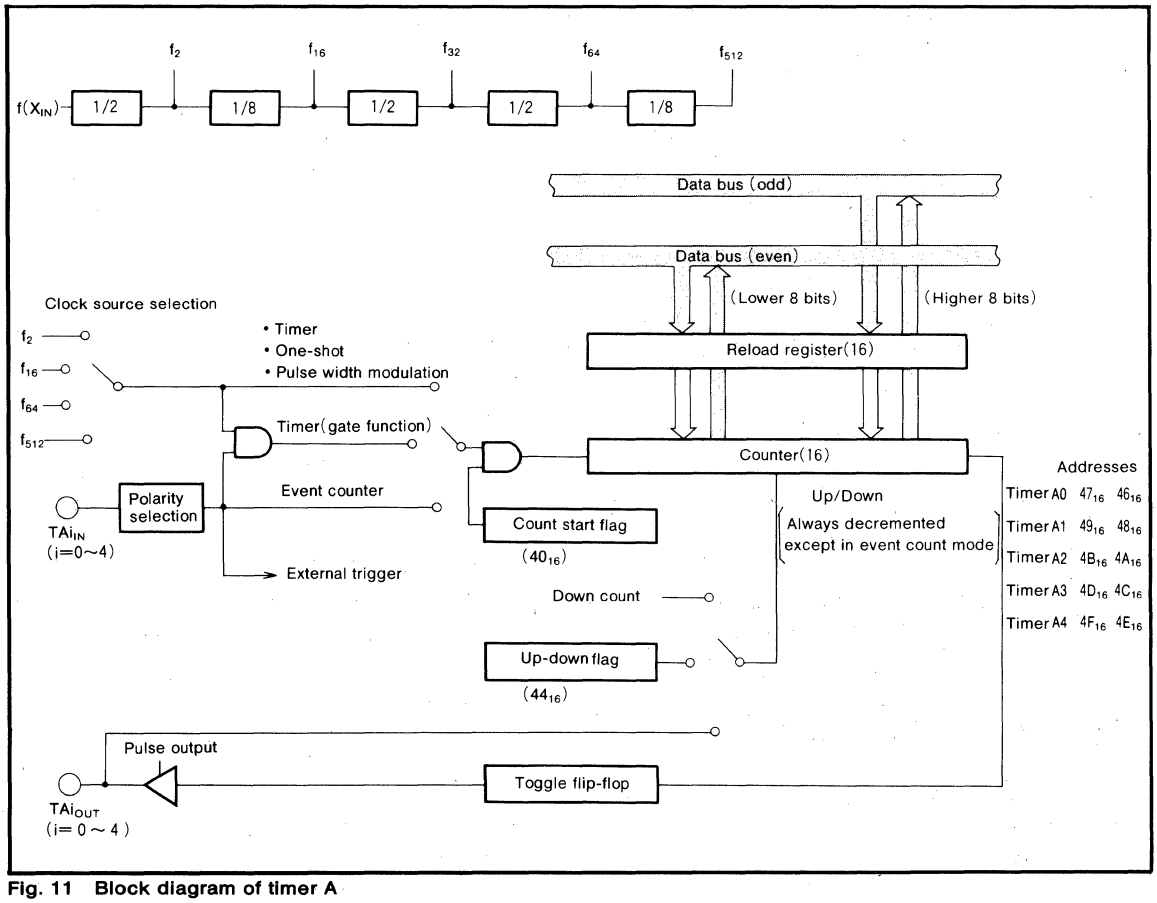


Fig. 11 Block diagram of timer A

When bit 2 of the timer Ai mode register is "1", the output is generated from TAi_{OUT} pin. The output is toggled each time the contents of the counter reaches to 0000_{16}. When the contents of the count start flag is "0", "L" is output from TAi_{OUT} pin.

When bit 2 is "0", TAi_{OUT} can be used as a normal port pin.

When bit 4 is "0", TAi_{IN} can be used as a normal port pin.

When bit 4 is "1", counting is performed only while the input signal from the TAi_{IN} pin is "H" or "L" as shown in Figure 14. Therefore, this can be used to measure the pulse width of the TAi_{IN} input signal. Whether to count while the input signal is "H" or while it is "L" is determined by bit 3. If bit 3 is "1", counting is performed while the TAi_{IN} pin input signal is "H" and if bit 3 is "0", counting is performed while it is "L".

Note that the duration of "H" or "L" on the TAi_{IN} pin must be two or more cycles of the timer count source.

When data is written to timer Ai register with timer Ai halted, the same data is also written to the reload register and the counter. When data is written to timer Ai which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time. The contents of the counter can be read at any time.

When the value set in the timer Ai register is n, the timer frequency dividing ratio is $1/(n+1)$.

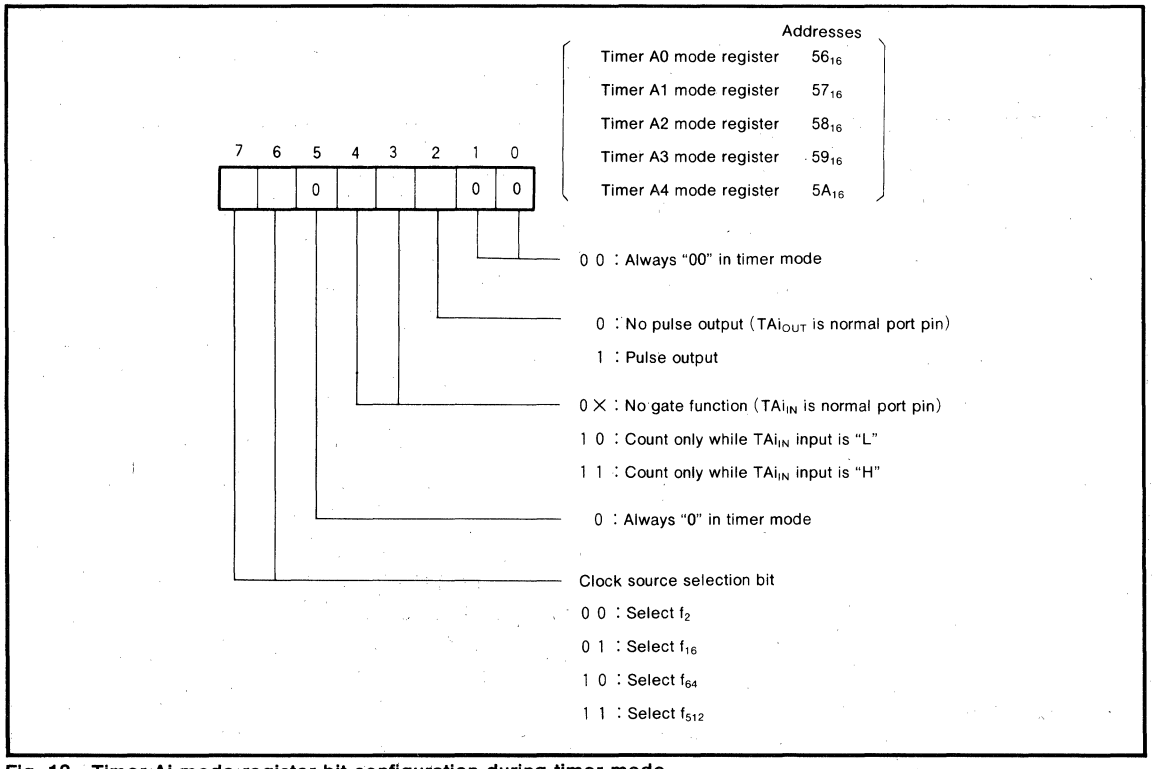


Fig. 12 Timer Ai mode register bit configuration during timer mode

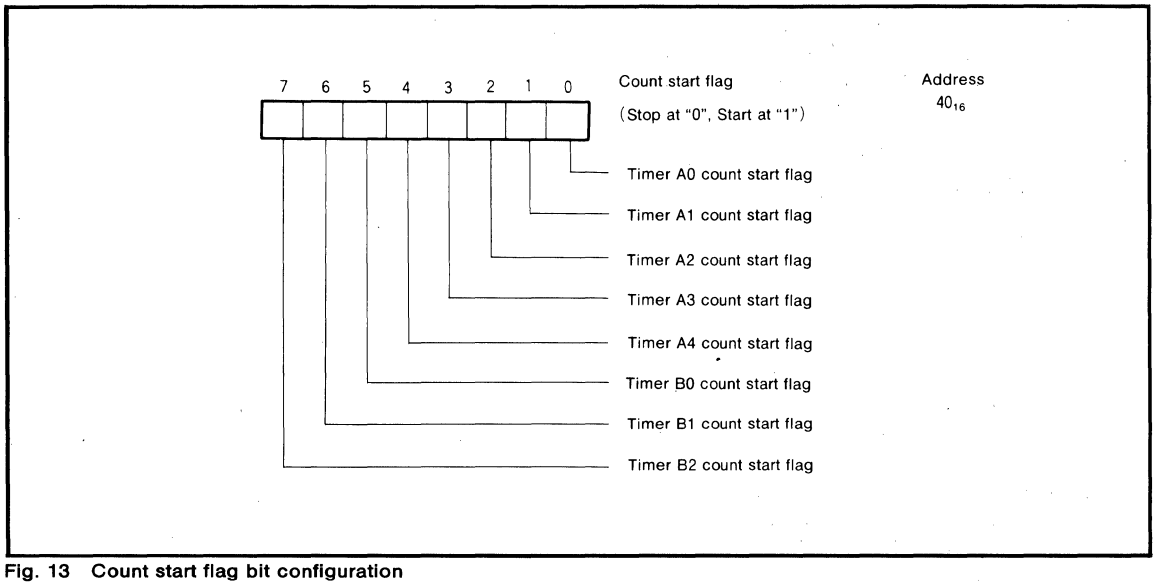


Fig. 13 Count start flag bit configuration

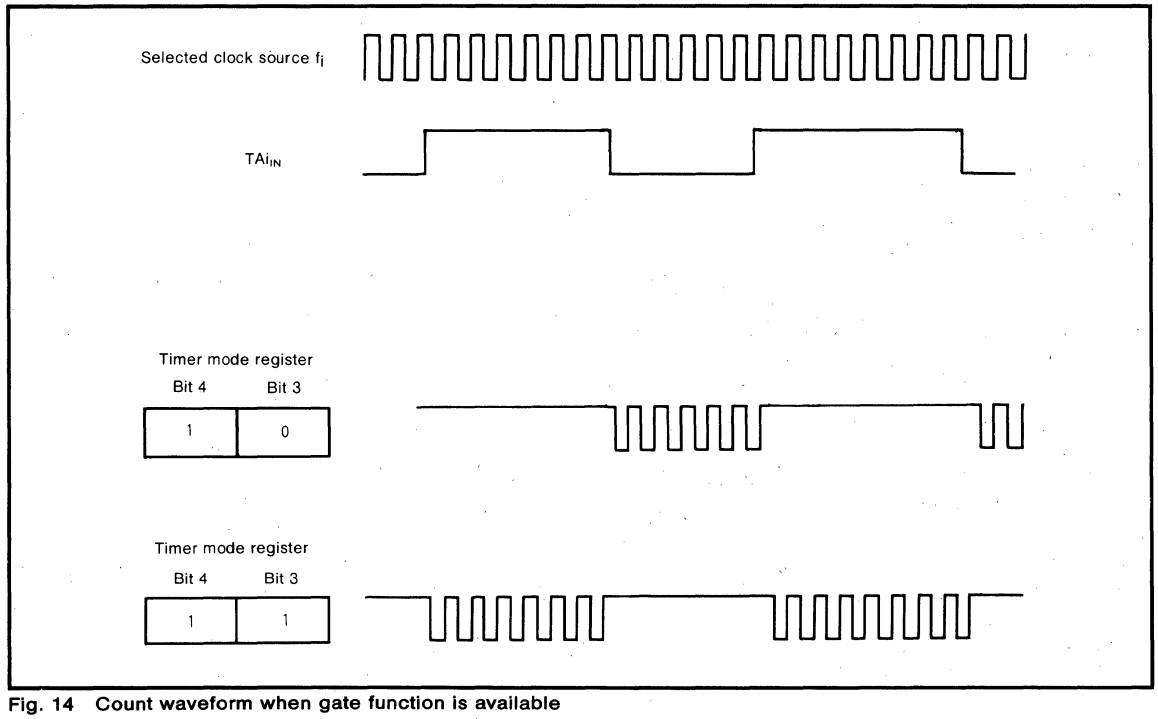


Fig. 14 Count waveform when gate function is available

(2) Event counter mode [01]

Figure 15 shows the bit configuration of the timer Ai mode register during event counter mode. In event counter mode, the bit 0 of the timer Ai mode register must be "1" and bit 1 and 5 must be "0".

The input signal from the TAi_{IN} pin is counted when the count start flag shown in Figure 13 is "1" and counting is stopped when it is "0".

Count is performed at the fall of the input signal when bit 3 is "0" and at the rise of the signal when it is "1".

In event counter mode, whether to increment or decrement the count can be selected with the up-down flag or the input signal from the TAi_{OUT} pin.

When bit 4 of the timer Ai mode register is "0", the up-down flag is used to determine whether to increment or decrement the count (decrement when the flag is "0" and increment when it is "1"). Figure 16 shows the bit configuration of the up-down flag.

When bit 4 of the timer Ai mode register is "1", the input signal from the TAi_{OUT} pin is used to determine whether to increment or decrement the count. However, note that bit 2 must be "0" if bit 4 is "1" because if bit 2 is "1", TAi_{OUT} pin becomes an output pin with pulse output.

The count is decremented when the input signal from the TAi_{OUT} pin is "L" and incremented when it is "H". Determine the level of the input signal from the TAi_{OUT} pin before valid edge is input to the TAi_{IN} pin.

An interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set when the counter reaches 0000_{16} (decrement count) or $FFFF_{16}$ (increment count). At the same time, the contents of the reload register is transferred to the counter and the count is continued.

When bit 2 is "1" and the counter reaches 0000_{16} (decrement count) or $FFFF_{16}$ (increment count), the waveform reversing polarity is output from TAi_{OUT} pin.

If bit 2 is "0", TAi_{OUT} pin can be used as a normal port pin. However, if bit 4 is "1" and the TAi_{OUT} pin is used as an output pin, the output from the pin changes the count direction. Therefore, bit 4 should be "0" unless the output from the TAi_{OUT} pin is to be used to select the count direction.

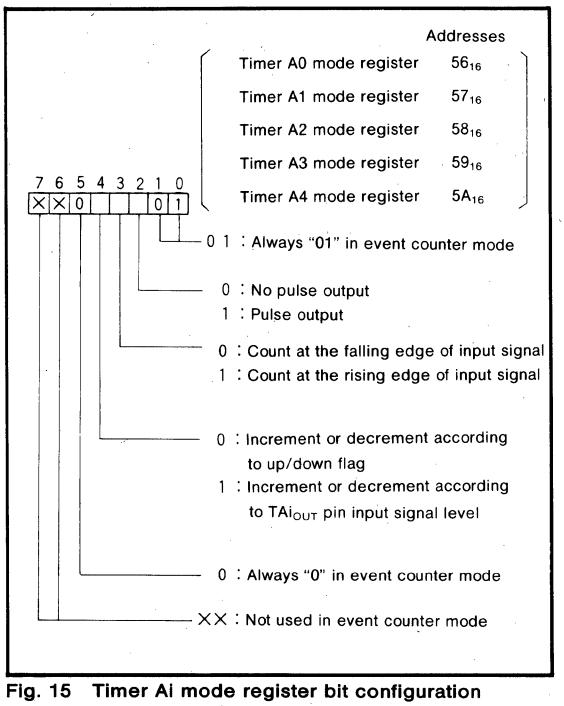


Fig. 15 Timer Ai mode register bit configuration during event counter mode

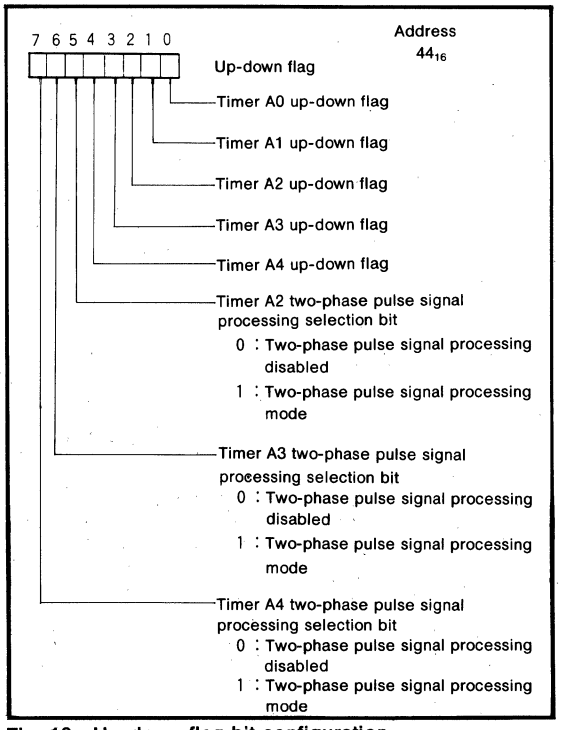


Fig. 16 Up-down flag bit configuration

Data write and data read are performed in the same way as for timer mode. That is, when data is written to timer Ai halted, it is also written to the reload register and the counter. When data is written to timer Ai which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time. The counter can be read at any time.

In event counter mode, whether to increment or decrement the counter can also be determined by supplying two-phase pulse input with phase shifted by 90° to timer A2, A3, or A4. There are two types of two-phase pulse processing operations. One uses timers A2 and A3, and the other uses timer A4. In either processing operation, two-phase pulse is input in the same way, that is, pulses out of phase by 90° are input at the TAj_{OUT} (j=2 to 4) pin and TAj_{IN} pin.

When timers A2 and A3 are used, as shown in Figure 17, the count is incremented when a rising edge is input to the TAk_{IN} pin after the level of TAk_{OUT} (k=2, 3) pin changes from "L" to "H", and when the falling edge is inserted, the count is decremented.

For timer A4, as shown in Figure 18, when a phase related pulse with a rising edge input to the $TA4_{IN}$ pin is input after the level of $TA4_{OUT}$ pin changes from "L" to "H", the count is incremented at the respective rising edge and falling edge of the $TA4_{OUT}$ pin and $TA4_{IN}$ pin.

When a phase related pulse with a falling edge input to the $TA4_{OUT}$ pin is input after the level of $TA4_{IN}$ pin changes from "H" to "L", the count is decremented at the respective rising edge and falling edge of the $TA4_{IN}$ pin and $TA4_{OUT}$ pin. When performing this two-phase pulse signal processing, timer Aj mode register bit 0 and bit 4 must be set to "1" and bits 1, 2, 3, and 5 must be "0". Bits 6 and 7 are ignored. Note that bits 5, 6, and 7 of the up-down flag register (44_{16}) are the two-phase pulse signal processing selection bit for timer A2, A3, and A4 respectively. Each timer operates in normal event counter mode when the corresponding bit is "0" and performs two-phase pulse signal processing when it is "1".

Count is started by setting the count start flag to "1". Data write and read are performed in the same way as for normal event counter mode. Note that the direction register of the input port must be set to input mode because two-phase pulse signal is input. Also, there can be no pulse output in this mode.

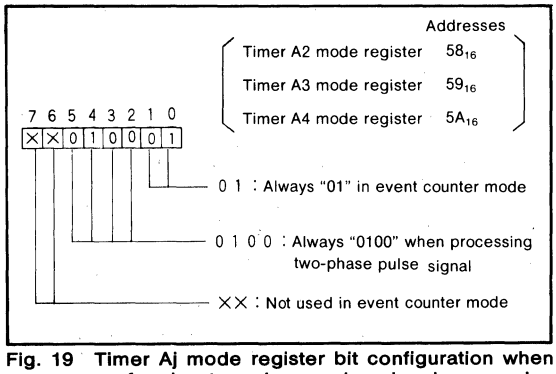


Fig. 19 Timer Aj mode register bit configuration when performing two-phase pulse signal processing in event counter mode

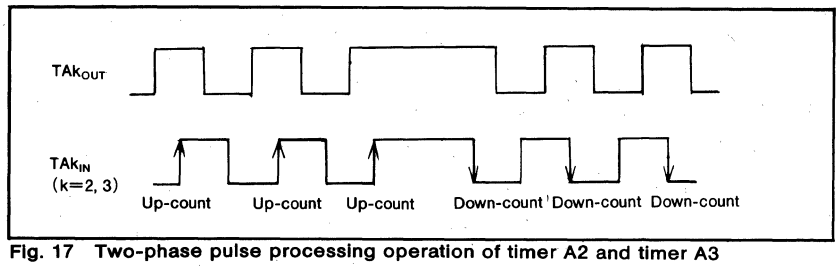


Fig. 17 Two-phase pulse processing operation of timer A2 and timer A3

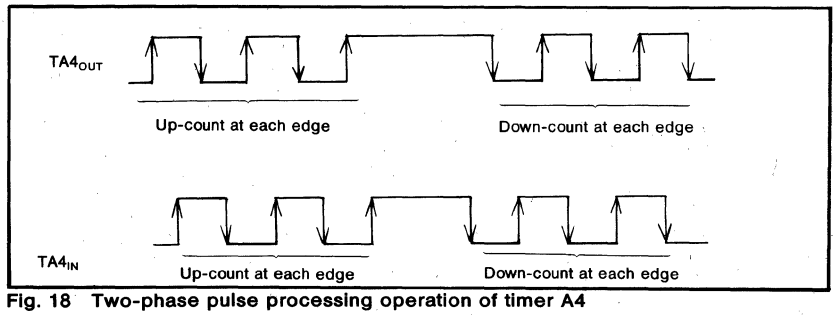


Fig. 18 Two-phase pulse processing operation of timer A4

(3) One-shot pulse mode (10)

Figure 20 shows the bit configuration of the timer Ai mode register during one-shot pulse mode. In one-shot pulse mode, bit 0 and bit 5 must be "0" and bit 1 and bit 2 must be "1".

The trigger is enabled when the count start flag is "1". The trigger can be generated by software or it can be input from the TAi_{IN} pin. Software trigger is selected when bit 4 is "0" and the input signal from the TAi_{IN} pin is used as the trigger when it is "1".

Bit 3 is used to determine whether to trigger at the fall of the trigger signal or at the rise. The trigger is at the fall of the trigger signal when bit 3 is "0" and at the rise of the trigger signal when it is "1".

Software trigger is generated by setting the bit in the one-shot start flag corresponding to each timer.

Figure 21 shows the bit configuration of the one-shot start flag.

As shown in Figure 22, when a trigger signal is received, the counter counts the clock selected by bits 6 and 7.

If the contents of the counter is not 0000_{16}, the TAi_{OUT} pin goes "H" when a trigger signal is received. The count direction is decrement.

When the counter reaches 0001_{16}, The TAi_{OUT} pin goes "L" and count is stopped. The contents of the reload register is transferred to the counter. At the same time, an interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set. This is repeated each time a trigger signal is received. The output pulse width is

$$\frac{1}{\text{pulse frequency of the selected clock}} \times (\text{counter's value at the time of trigger}).$$

If the count start flag is "0", TAi_{OUT} goes "L". Therefore, the value corresponding to the desired pulse width must be written to timer Ai before setting the timer Ai count start flag.

As shown in Figure 23, a trigger signal can be received before the operation for the previous trigger signal is completed. In this case, the contents of the reload register is transferred to the counter by the trigger and then that value is decremented.

Except when retriggering while operating, the contents of the reload register is not transferred to the counter by triggering.

When retriggering, there must be at least one timer count source cycle before a new trigger can be issued.

Data write is performed to the same way as for timer mode. When data is written in timer Ai halted, it is also written to the reload register and the counter.

When data is written to timer Ai which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time.

Undefined data is read when timer Ai is read.

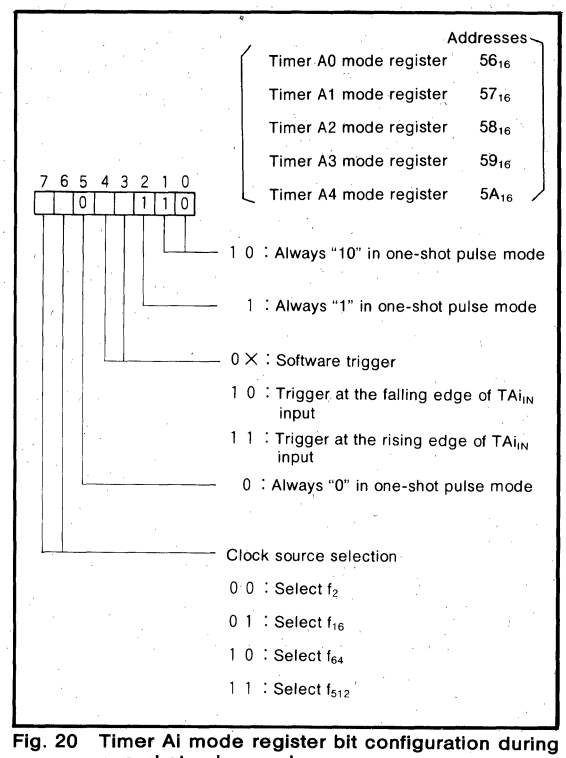


Fig. 20 Timer Ai mode register bit configuration during one-shot pulse mode

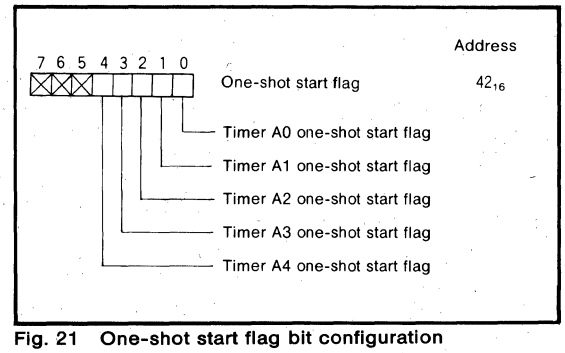


Fig. 21 One-shot start flag bit configuration

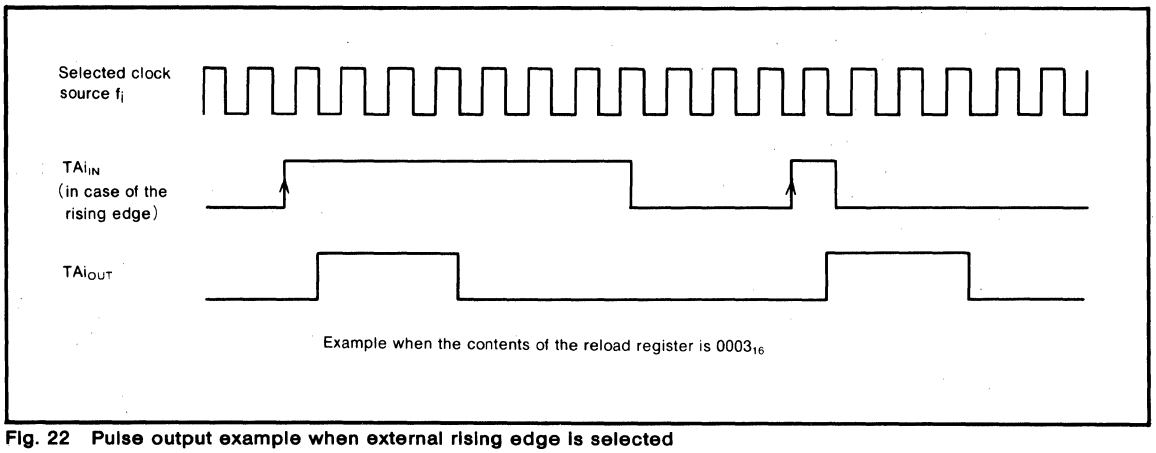


Fig. 22 Pulse output example when external rising edge is selected

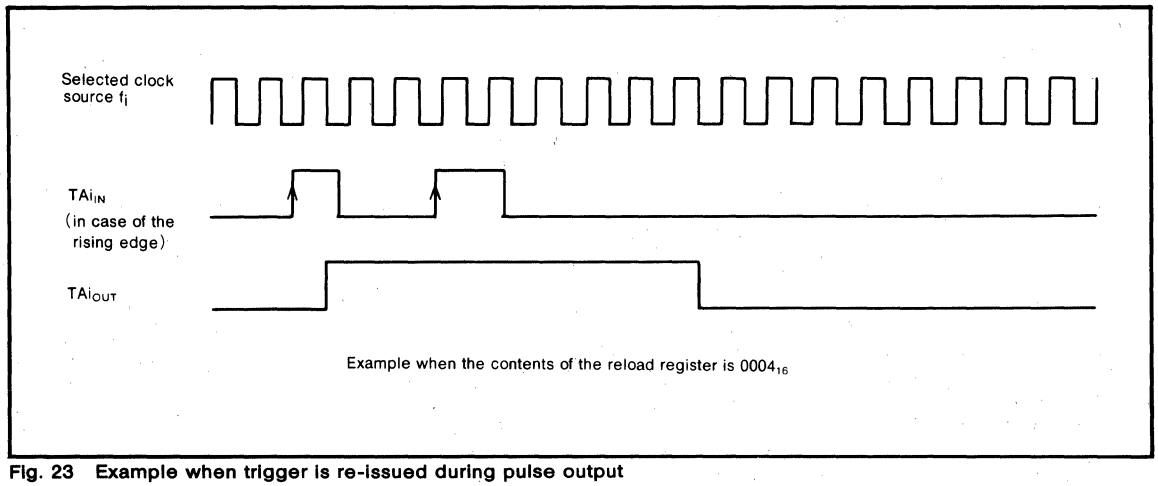


Fig. 23 Example when trigger is re-issued during pulse output

(4) Pulse width modulation mode (11)

Figure 24 shows the bit configuration of the timer Ai mode register during pulse width modulation mode. In pulse width modulation mode, bits 0, 1, and 2 must be set to "1".

Bit 5 is used to determine whether to perform 16-bit length pulse width modulator or 8-bit length pulse width modulator. 16-bit length pulse width modulator is performed when bit 5 is "0" and 8-bit length pulse width modulator is performed when it is "1". The 16-bit length pulse width modulator is described first.

The pulse width modulator can be started with a software trigger or with an input signal from a TAi_{IN} pin (external trigger).

The software trigger mode is selected when bit 4 is "0". Pulse width modulator is started and pulse is output from TAi_{OUT} when the timer Ai start flag is set to "1".

The external trigger mode is selected when bit 4 is "1". Pulse width modulator starts when a trigger signal is input from the TAi_{IN} pin when the timer Ai start flag is "1". Whether to trigger at the fall or rise of the trigger signal is determined by bit 3. The trigger is at the fall of the trigger signal when bit 3 is "0" and at the rise when it is "1".

When data is written to timer Ai with the pulse width modulator halted, it is written to the reload register and the counter.

Then when the timer Ai start flag is set to "1" and a software trigger or an external trigger is issued to start modulation, the waveform shown in Figure 25 is output continuously. Once modulation is started, triggers are not accepted. If the value in the reload register is m, the duration "H" of pulse is

$$\frac{1}{\text{selected clock frequency}} \times m$$

and the output pulse period is

$$\frac{1}{\text{selected clock frequency}} \times (2^{16}-1).$$

An interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set at each fall of the output pulse.

The width of the output pulse is changed by updating timer data. The update can be performed at any time. The output pulse width is changed at the rise of the pulse after data is written to the timer.

The contents of the reload register are transferred to the counter just before the rise of the next pulse so that the pulse width is changed from the next output pulse.

Undefined data is read when timer Ai is read.

The 8-bit length pulse width modulator is described next.

The 8-bit length pulse width modulator is selected when the timer Ai mode register bit 5 is "1".

The reload register and the counter are both divided into 8-bit halves.

The low order 8 bits function as a prescaler and the high order 8 bits function as the 8-bit length pulse width modulator. The prescaler counts the clock selected by bits 6 and 7. A pulse is generated when the counter reaches 0000_{16} as shown in Figure 26. At the same time, the contents of the reload register is transferred to the counter and count is continued.

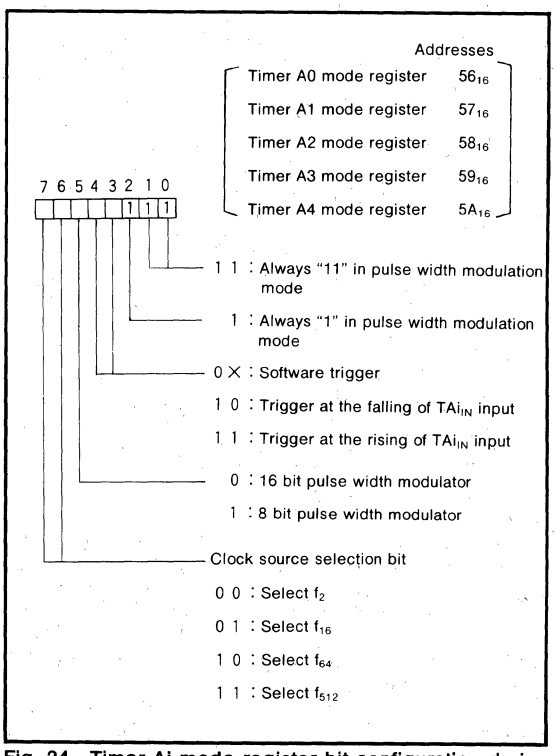


Fig. 24 Timer Ai mode register bit configuration during pulse width modulation mode

Therefore, if the low order 8-bit of the reload register is n, the period of the generated pulse is

$$\frac{1}{\text{selected clock frequency}} \times (n+1).$$

The high order 8-bit function as an 8-bit length pulse width modulator using this pulse as input. The operation is the same as for 16-bit length pulse width modulator except that the length is 8 bits. If the high order 8-bit of the reload register is m, the duration "H" of pulse is

$$\frac{1}{\text{selected clock frequency}} \times (n+1) \times m.$$

And the output pulse period is

$$\frac{1}{\text{selected clock frequency}} \times (n+1) \times (2^8-1).$$

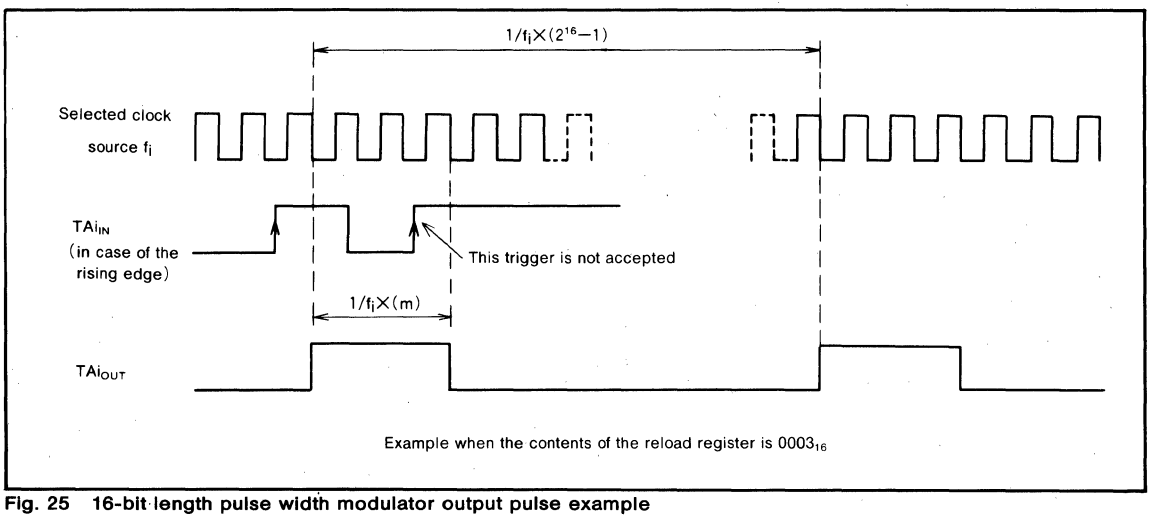


Fig. 25 16-bit length pulse width modulator output pulse example

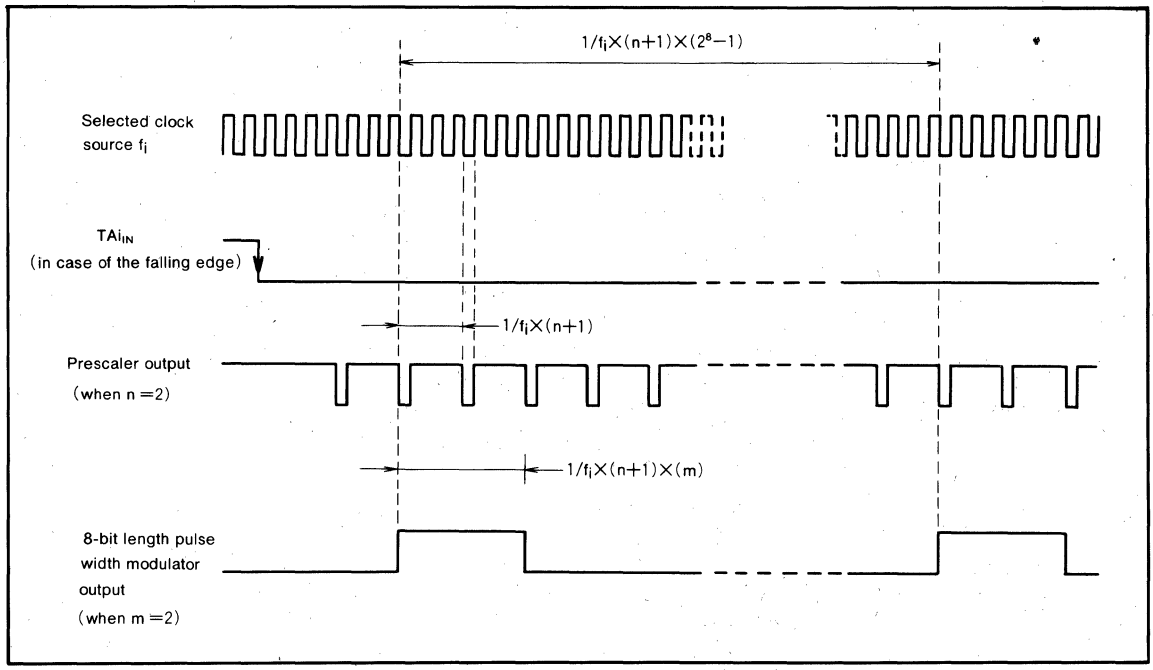


Fig. 26 8-bit length pulse width modulator output pulse example

TIMER B

Figure 27 shows a block diagram of timer B.

Timer B has three modes; timer mode, event counter mode, and pulse period measurement/pulse width measurement mode. The mode is selected with bits 0 and 1 of the timer Bi mode register (i = 0 to 2). Each of these modes is described below.

(1) Timer mode [00]

Figure 28 shows the bit configuration of the timer Bi mode register during timer mode. Bits 0, and 1 of the timer Bi mode register must always be "0" in timer mode.

Bits 6 and 7 are used to select the clock source. The counting of the selected clock starts when the count start flag "1" and stops when "0".

As shown in Figure 13, the timer Bi count start flag is at the same address as the timer Ai count start flag. The count is decremented, an interrupt occurs, and the interrupt request bit in the timer Bi interrupt control register is set when the contents becomes 0000_{16}. At the same time, the contents of the reload register is stored in the counter and count is continued.

Timer Bi does not have a pulse output function or a gate function like timer A.

When data is written to timer Bi halted, it is written to the reload register and the counter. When data is written to timer Bi which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time. The contents of the counter can be read at any time.

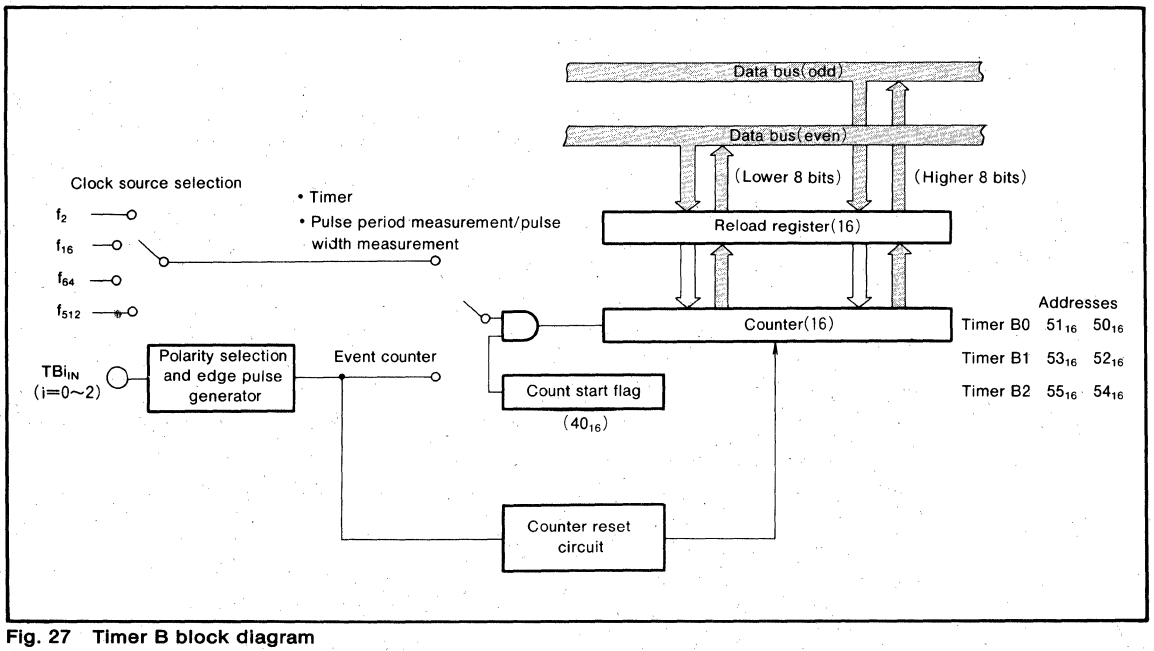


Fig. 27 Timer B block diagram

(2) Event counter mode [01]

Figure 29 shows the bit configuration of the timer Bi mode register during event counter mode. In event counter mode, the bit 0 in the timer Bi mode egister must be "1" and bit 1 must be "0".

The input signal from the TBi_{IN} pin is counted when the count start flag is "1" and counting is stopped when it is "0". Count is performed at the fall of the input signal when bits 2, and 3 are "0" and at the rise of the input signal when bit 3 is "0" and bit 2 is "1".

When bit 3 is "1" and bit 2 is "0", count is performed at the rise and fall of the input signal.

Data write, data read and timer interrupt are performed in the same way as for timer mode.

(3) Pulse period measurement/pulse width measurement mode [10]

Figure 30 shows the bit configuration of the timer Bi mode register during pulse period measurement/pulse width measurement mode.

In pulse period measurement/pulse width measurement mode, bit 0 must be "0" and bit 1 must be "1". Bits 6 and 7 are used to select the clock source. The selected clock is counted when the count start flag is "1" and counting stops when it is "0".

The pulse period measurement mode is selected when bit 3 is "0". In pulse period measurement mode, the selected clock is counted during the interval starting at the fall of the input signal from the TBi_{IN} pin to the next fall or at the rise of the input signal to the next rise and the result is stored in the reload register. In this case, the reload register acts as a buffer register.

When bit 2 is "0", the clock is counted from the fall of the input signal to the next fall. When bit 2 is "1", the clock is counted from the rise of the input signal to the next rise.

In the case of counting from the fall of the input signal to the next fall, counting is performed as follows. As shown in Figure 31, when the fall of the input signal from TBi_{IN} pin is detected, the contents of the counter is transferred to the reload register. Next the counter is cleared and count is started from the next clock. When the fall of the next input signal is detected, the contents of the counter is transferred to the reload register once more, the counter is cleared, and the count is started. The period from the fall of the input signal to the next fall is measured in this way.

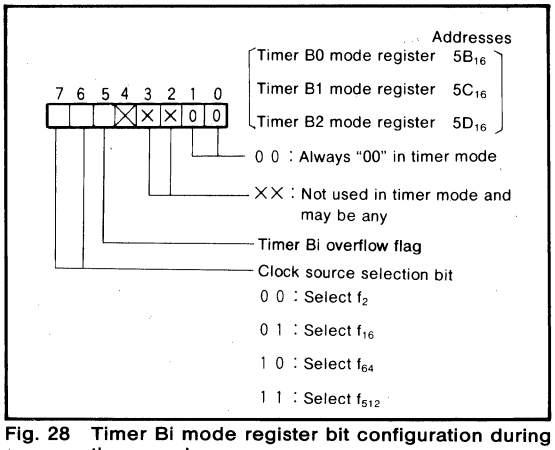


Fig. 28 Timer Bi mode register bit configuration during timer mode

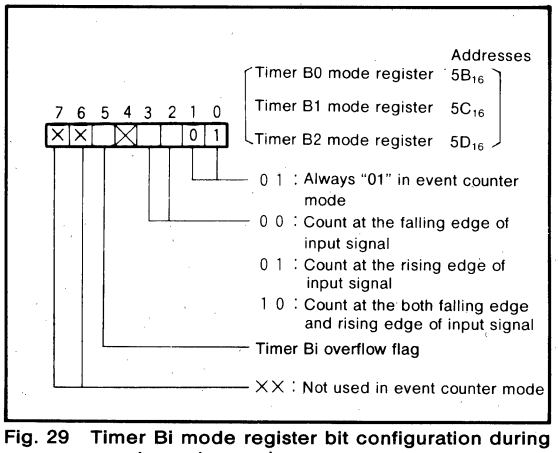


Fig. 29 Timer Bi mode register bit configuration during event counter mode

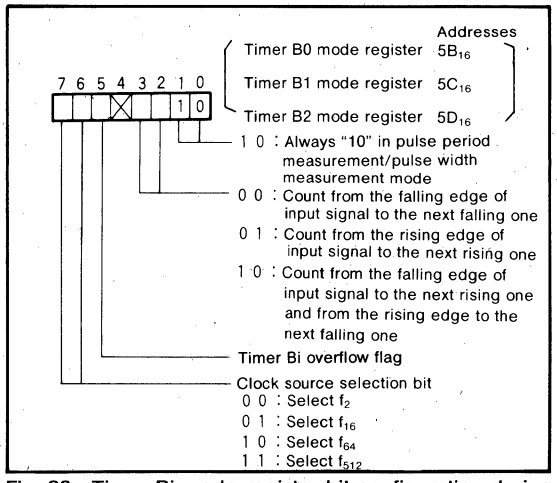


Fig. 30 Timer Bi mode register bit configuration during pulse period measurement/pulse width measurement mode

After the contents of the counter is transferred to the reload register, an interrupt request signal is generated and the interrupt request bit in the timer Bi interrupt control register is set. However, no interrupt request signal is generated when the contents of the counter is transferred first time to the reload register after the count start flag is set to "1".

When bit 3 is "1", the pulse width measurement mode is selected. Pulse width measurement mode is similar to pulse period measurement mode except that the clock is counted from the fall of the TBi_{IN} pin input signal to the next rise or from the rise of the input signal to the next fall as shown in Figure 32.

When timer Bi is read, the contents of the reload register is read.

Note that in this mode, the interval between the fall of the TBi_{IN} pin input signal to the next rise or from the rise to the next fall must be at least two cycles of the timer count source.

Timer Bi overflow flag which is bit 5 of timer Bi mode register is set to "1" when the timer Bi counter reaches 0000_{16}. This flag is cleared by writing to corresponding timer Bi mode register. This bit is set to "1" at reset.

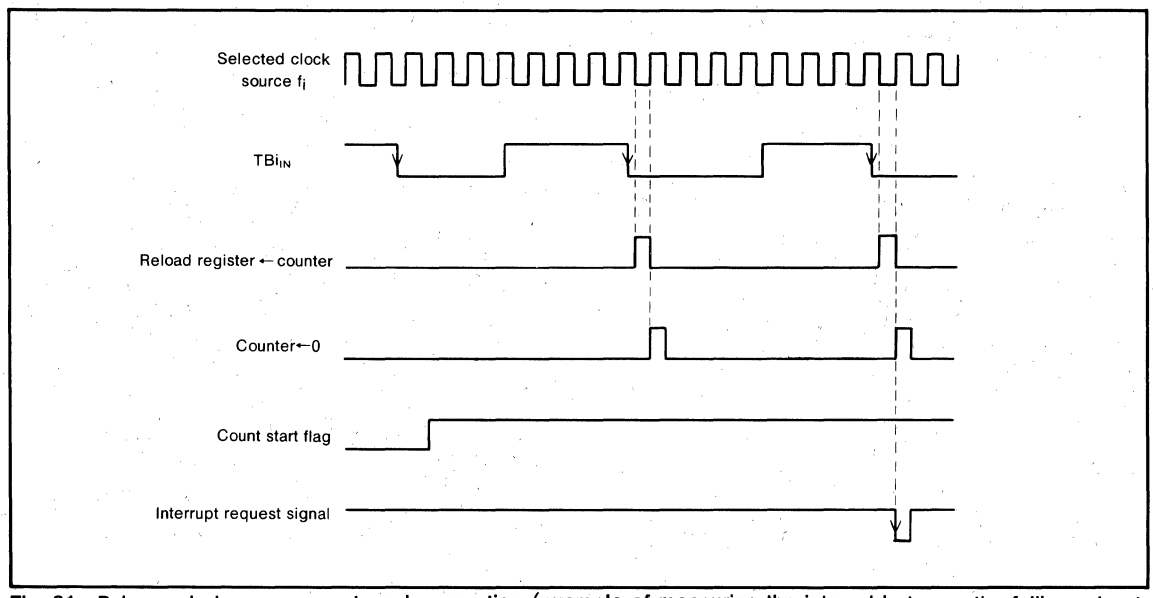


Fig. 31 Pulse period measurement mode operation (example of measuring the interval between the falling edge to next falling one)

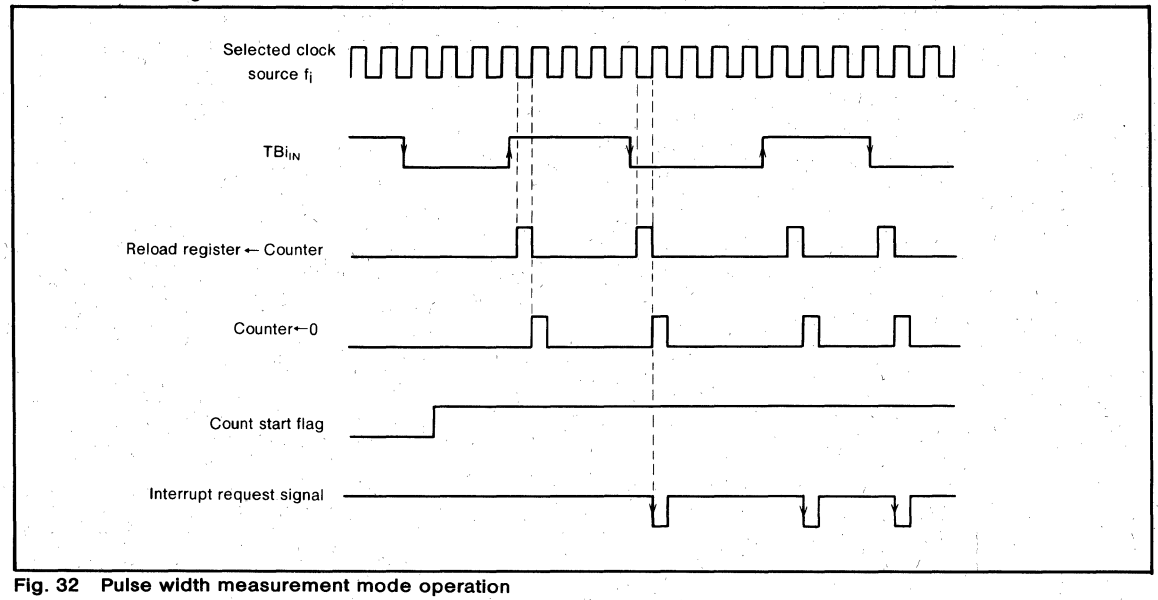


Fig. 32 Pulse width measurement mode operation

SERIAL I/O PORTS

Two independent serial I/O ports are provided. Figure 33 shows a block diagram of the serial I/O ports.

Bits 0, 1, and 2 of the UARTi (i = 0, 1) Transmit/Receive mode register shown in Figure 34 are used to determine whether to use port P8 as parallel port, clock synchronous serial I/O port, or asynchronous (UART) serial I/O port using start and stop bits.

Figures 35 and 36 show the connections of receiver/transmitter according to the mode.

Figure 37 shows the bit configuration of the UARTi transmit/receive control register.

Each communication method is described below.

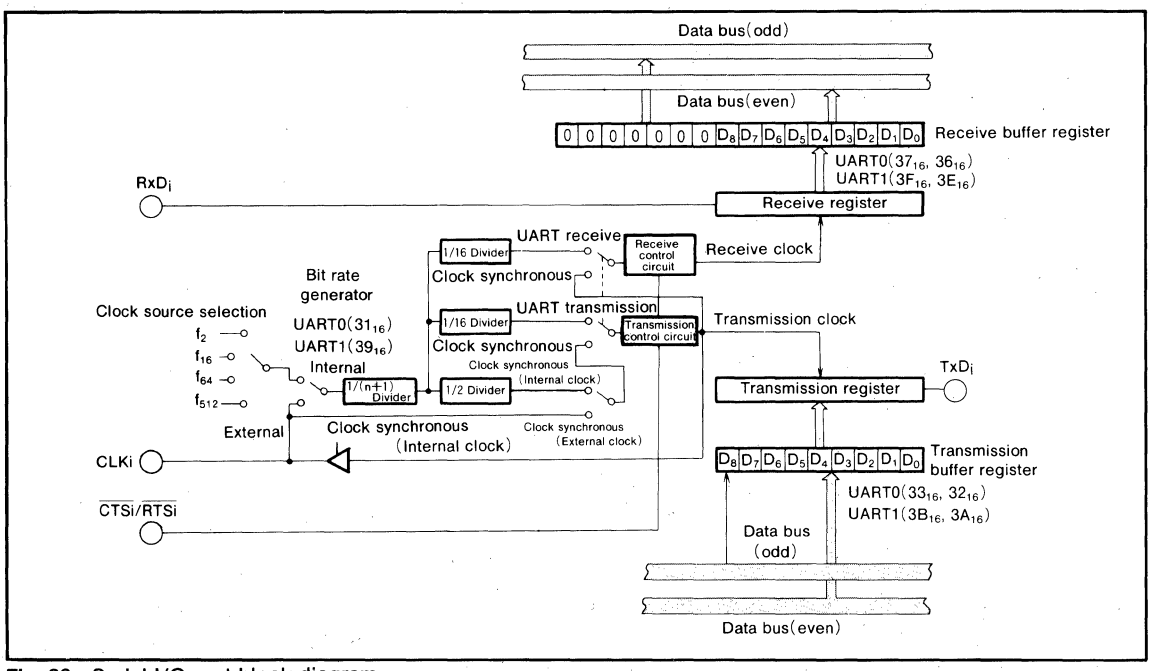


Fig. 33 Serial I/O port block diagram

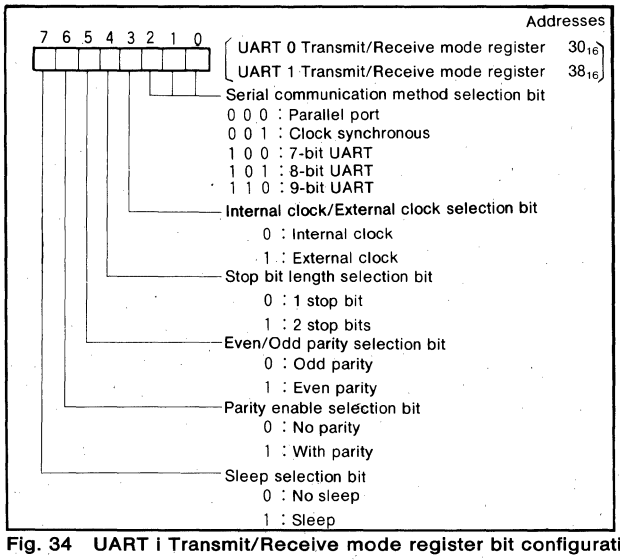


Fig. 34 UART i Transmit/Receive mode register bit configuration

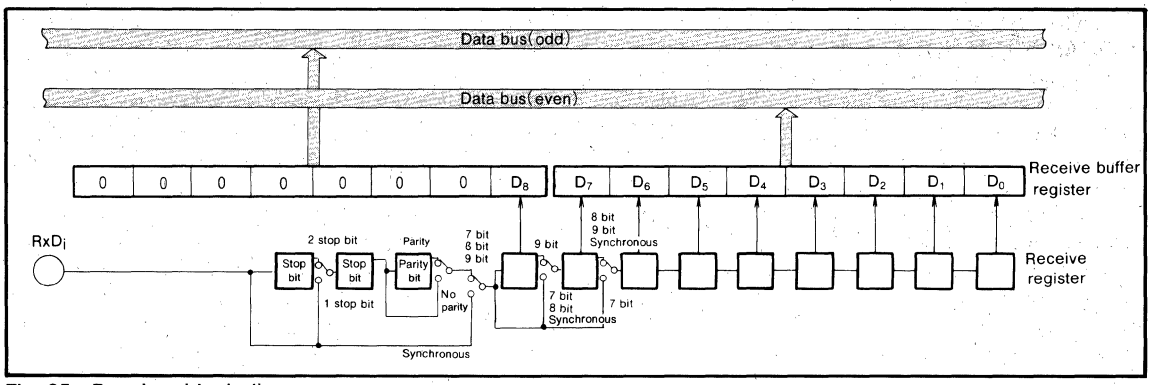


Fig. 35 Receiver block diagram

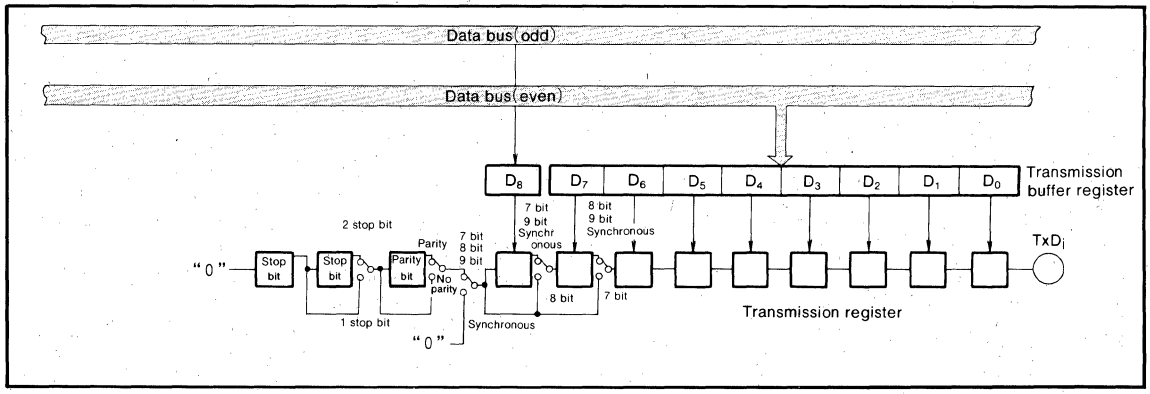


Fig. 36 Transmitter block diagram

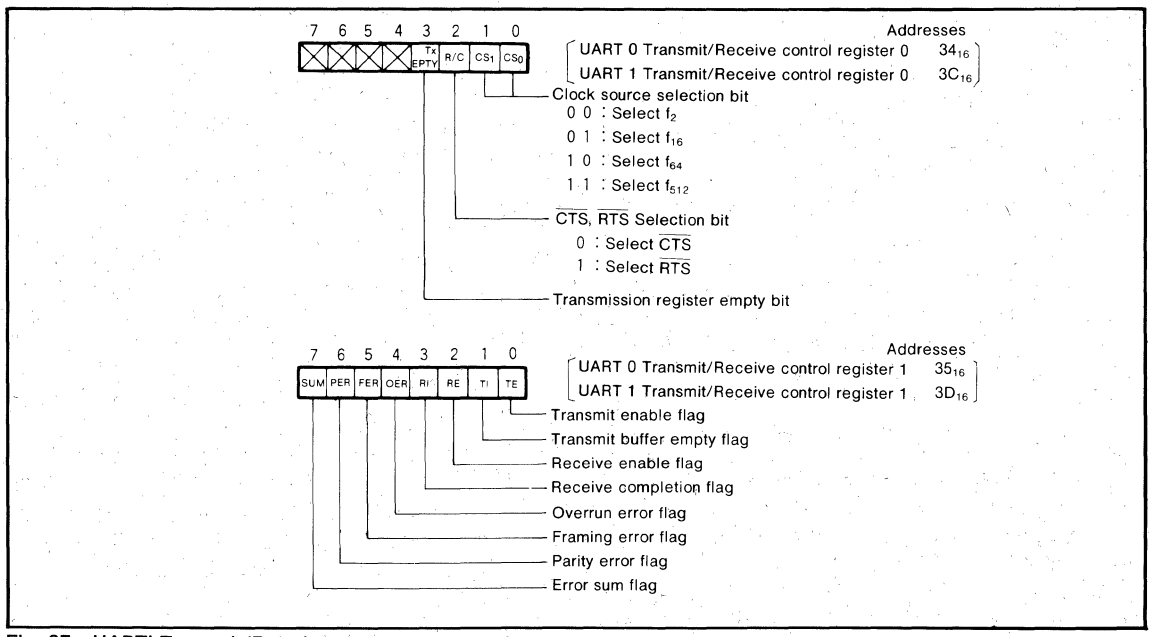


Fig. 37 UARTi Transmit/Receive control register bit configuration

CLOCK SYNCHRONOUS SERIAL COMMUNICATION

A case where communication is performed between two clock synchronous serial I/O ports as shown in Figure 38 will be described. (The transmission side will be denoted by subscript j and the receiving side will be denoted by subscript k.)

Bit 0 of the UARTj transmit/receive mode register and UARTk transmit/receive mode register must be set to "1" and bits 1 and 2 must be "0". The length of the transmission data is fixed at 8 bits.

Bit 3 of the UARTj transmit/receive mode register of the clock sending side is cleared to "0" to select the internal clock. Bit 3 of the UARTk transmit/receive mode register of the clock receiving side is set to "1" to select the external clock. Bits 4, 5 and 6 are ignored in clock synchronous mode. Bit 7 must always be "0".

The clock source is selected by bit 0 (CS_0) and bit 1 (CS_1) of the clock sending side UARTj transmit/receive control register 0. As shown in Figure 33, the selected clock is divided by (n + 1), then by 2, passed through a transmisson control circuit, and output as transmisson clock CLKj. Therefore, when the selected clock is fi,

$$\text{Bit Rate} = fi / \{(n+1) \times 2\}$$

On the clock receiving side, the CS_0 and CS_1 bits of the UARTk transmit/receive control register are ignored because an external clock is selected.

The bit 2 of the clock sending side UARTj transmit/receive control register is clear to "0"to select $\overline{CTS}j$ input. The bit 2 of the clock receiving side is set to "1" to select $\overline{RTS}k$ output. $\overline{CTS}$, and $\overline{RTS}$ signals are described later.

Transmission

Transmission is started when the bit 0 (TEj flag) of UARTj transmit/receive control register 1 is "1", bit 1 is (TIj flag) of one is "0", and $\overline{CTS}j$ input is "L". As shown in Figure 39, data is output from TxDj pin when transmission clock CLKj changes from "H" to "L". The data is output from the least significant bit.

The TIj flag indicates whether the transmission buffer register is empty or not. It is cleared to "0" when data is written in the transmission buffer register and set to "1" when the contents of the transmission buffer register is transferred to the transmission register.

When the transmission register becomes empty after the contents has been transmitted, data is transferred automatically from the transmission buffer register to the transmission register if the next transmission start condition is satisfied. If the bit 2 of UARTj transmit/receive control register 0 is "1", $\overline{CTS}j$ input is ignored and transmission start is controlled only by the TEj flag and TIj flag. Once transmission has started, the TEj flag, TIj flag, and $\overline{CTS}j$ signals are ignored until data transmission completes. Therefore, transmission is not interrupt when $\overline{CTS}j$ input is changed to "H" during transmission.

The transmission start condition indicated by TEj flag, TIj flag, and $\overline{CTS}j$ is checked while the T_{ENDj} signal shown in Figure 39 is "H". Therefore, data can be transmitted continuously if the next transmission data is written in the transmission buffer register and TIj flag is cleared to "0" before the T_{ENDj} signal goes "H".

The bit 3 (TxEPTYj flag) of UARTj transmit/receive control register 0 changes to "1" at the next cycle after the T_{ENDj} signal goes "H" and changes to "0" when transmission starts. Therefore, this flag can be used to determine whether data transmission has completed.

When the TIj flag changes from "0" to "1", the interrupt request bit in the UARTj transmission interrupt control register is set to "1".

Receive

Receive starts when the bit 2 (RE_K flag) of $UART_K$ transmit/receive control register 1 is set to "1".

The $\overline{RTS}_K$ output is "H" when the RE_K flag is "0" and goes "L" when the RE_K flag changed to "1". It goes back to "H" when receive starts. Therefore, the $\overline{RTS}_K$ output can be used to determine whether the receive register is ready to receive. It is ready when $\overline{RTS}_K$ output is "L".

The data from the RxD_K pin is retrieved and the contents of the receive register is shifted by 1 bit each time the transmission clock CLKj changes from "L" to"H". When an 8-bit data is received, the contents of the receive register is transferred to the receive buffer register and the bit 3 (RI_K flag) of $UART_K$ transmit/receive control register 1 is set to "1". In other words, the setting of the RI_K flag indicates that the receive buffer register contains the received data. At this point, $\overline{RTS}j$ output goes "L" to indicate that the next data can be received. When the RI_K flag changes from "0" to "1", the interrupt request bit in the $UART_K$ receive interrupt control register is set to "1". Bit 4 (OER_K flag) of $UART_K$ transmit/receive control register is set to "1" when the next data is transferred from the receive register to the receive buffer register while RI_K flag is "1", and indicates that the next data was transferred to the receive register before the contents of the receive buffer register was read. RI_K and OER_K flags are cleared automatically to "0" when the low-order byte of the receive buffer register is read. The OER_K flag is also cleared when the RE_K flag is cleared. Bit 5 (FER_K flag), bit 6 (PER_K flag), and bit 7 (SUM_K flag) are ignored in clock synchronous mode.

As shown in Figure 33, with clock synchronous serial communication, data cannot be received unless the transmitter is operating because the receive clock is created from the transmission clock. Therefore, the transmitter must be operating even when there is no data to be sent from $UART_K$ to UARTj.

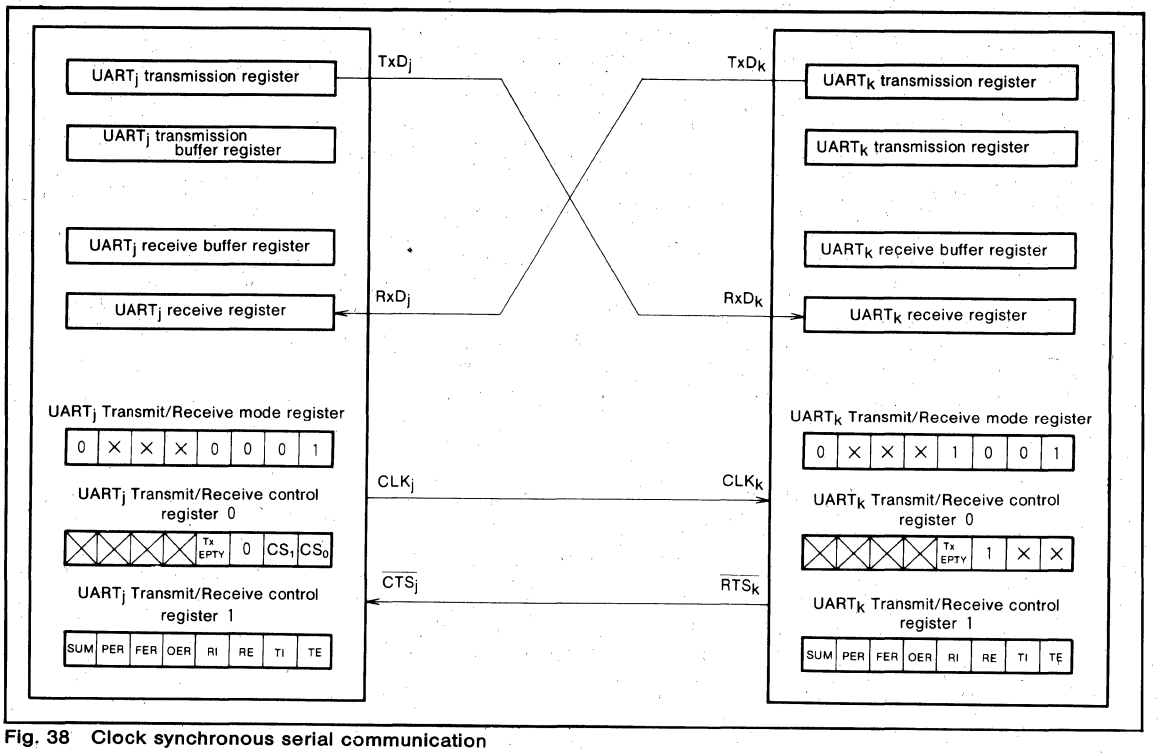


Fig. 38 Clock synchronous serial communication

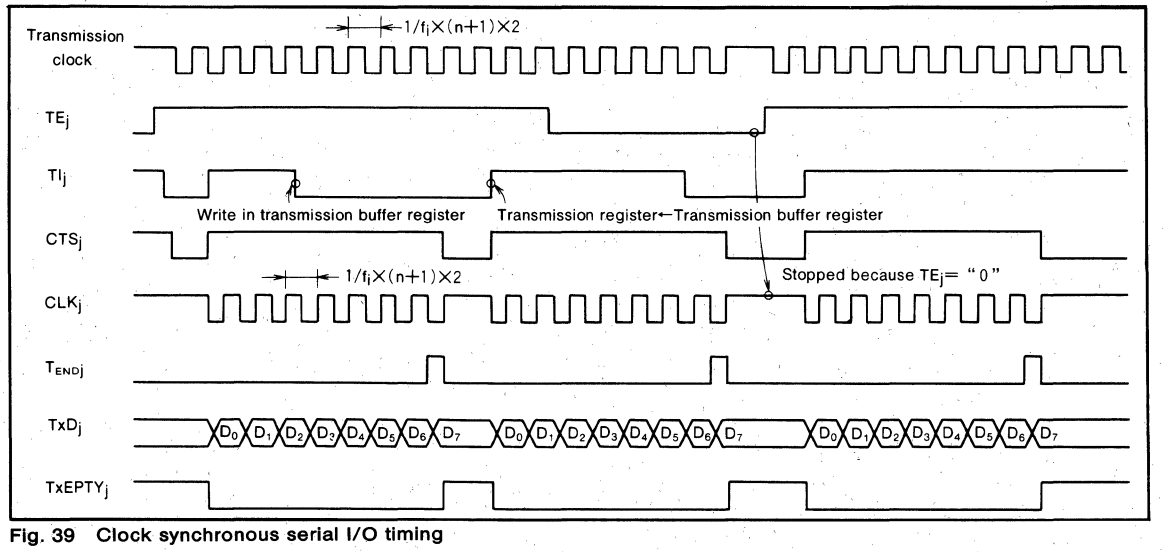


Fig. 39 Clock synchronous serial I/O timing

ASYNCHRONOUS SERIAL COMMUNICATION

Asynchronous serial communication can be performed using 7-, 8-, or 9-bit length data. The operation is the same for all data lengths. The following is the description for 8-bit asynchronous communication.

With 8-bit asynchronous communication, the bit 0 of UARTi transmit/receive mode register is "1", the bit 1 is "0", and the bit 2 is "1".

Bit 3 is used to select an internal clock or an external clock. If bit 3 is "0", an internal clock is selected and if bit 3 is "1", then external clock is selected. If an internal clock is selected, the bit 0 (CS_0) and bit 1 (CS_1) of UARTi transmit/receive control register 0 are used to select the clock source. When an internal clock is selected for asynchronous serial communication, the CLK pin can be used as a normal I/O pin.

The selected internal or external clock is divided by (n+1), then by 16, and passed through a control circuit to create the UART transmission clock or UART receive clock.

Therefore, the transmission speed can be changed by changing the contents n of the bit rate generator. If the selected clock is an internal clock fi or an external clock f_{EXT},

$$\text{Bit Rate} = (f_i \text{ or } f_{EXT}) / \{(n+1) \times 16\}$$

Bit 4 is the stop bit length selection bit to select 1 stop bit or 2 stop bits.

The bit 5 is a selection bit of odd parity or even parity.

In the odd parity mode, the parity bit is adjusted so that the sum of the 1's in the data and parity bit is always odd.

In the even parity mode, the parity bit is adjusted so that the sum of the 1's in the data and parity bit is always even.

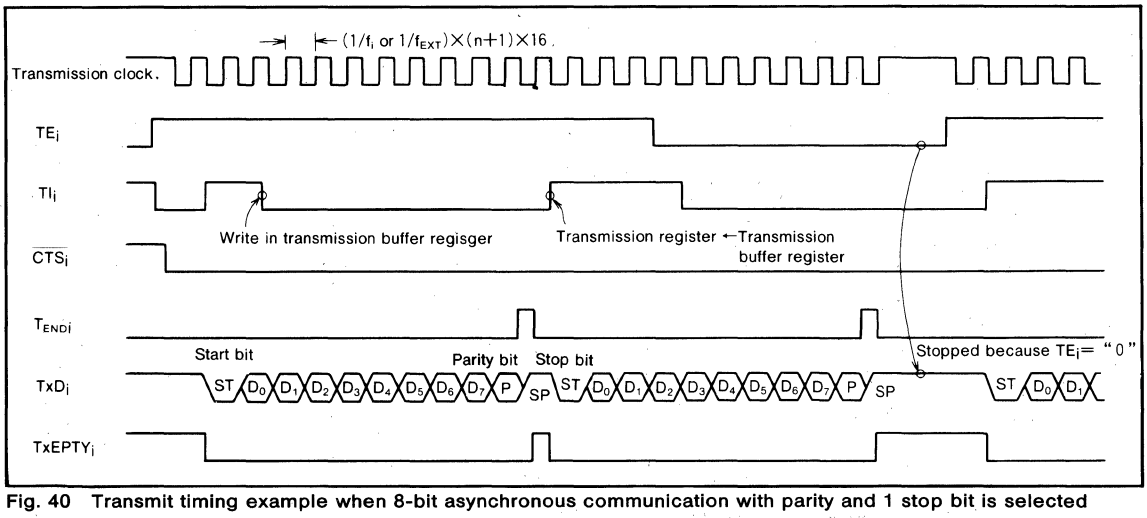


Fig. 40 Transmit timing example when 8-bit asynchronous communication with parity and 1 stop bit is selected

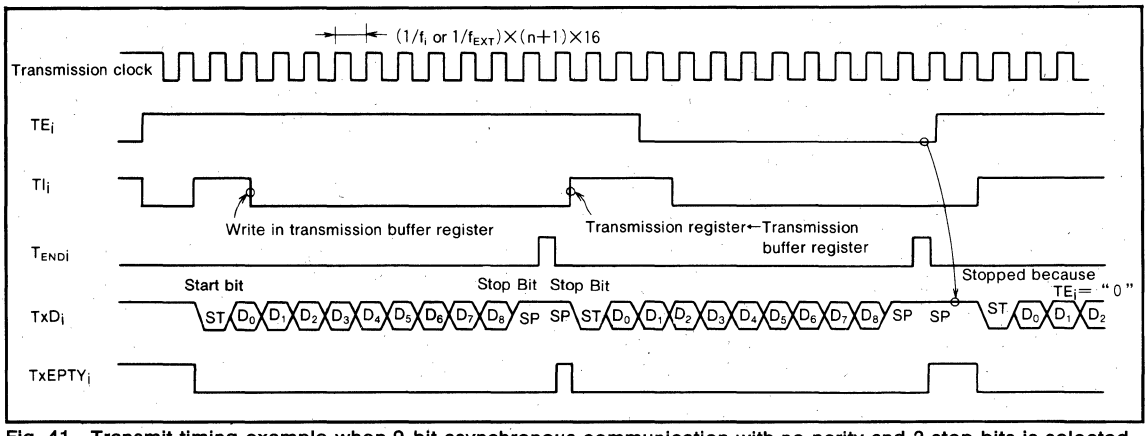


Fig. 41 Transmit timing example when 9-bit asynchronous communication with no parity and 2 stop bits is selected

Bit 6 is the parity bit selection bit which indicates whether to add parity bit or not.
Bits 4 to 6 should be set or reset according to the data format of the communicating devices.
Bit 7 is the sleep selection bit. The sleep mode is described later.
The UART$_i$ transmit/receive control regisger 0 bit 2 is used to determine whether to use $\overline{CTS_i}$ input or $\overline{RTS_i}$ output. $\overline{CTS_i}$ input used if bit 2 is "0" and $\overline{RTS_i}$ output is used if bit 2 is "1".
If $\overline{CTS_i}$ input is selected, the user can control whether to stop or start transmission by external $\overline{CTS_i}$ input. $\overline{RTS_i}$ will be described later.

Transmission

Transmission is started when the bit 0 (TEi flag) of UART$_i$ transmit/receive control register 1 is "1", the bit 1 (TI$_i$ flag) is "0", and $\overline{CTS_i}$ input is "L" if $\overline{CTS_i}$ input is selected. As shown in Figure 40 and 41, data is output from the TxD$_i$ pin with the stop bit and parity bit specified by the bits 4 to 6 of UART$_i$ transmit/receive mode register bits. The data is output from the least significant bit.
The TI$_i$ flag indicates whether the transmission butter is empty or not. It is cleared to "0" when data is written in the transmission buffer and set to "1" when the contents of the transmission buffer register is transferred to the transmission register.

When the transmission register becomes empty after the contents has been transmitted, data is transferred automatically form the transmission buffer register to the transmission register if the next transmission start condition is satisfied.
Once transmission has started, the TE$_i$ flag, TI$_i$ flag, and $\overline{CTS_i}$ signal (if $\overline{CTS_i}$ input is selected) are ignored until data transmission is completed.
Therefore, transmission does not stop until it completes even if the TE$_i$ flag is cleared during transmission.
The transmission start condition indicated by TE$_i$ flag, TI$_i$ flag, and $\overline{CTS_i}$ is checked while the T_{ENDi} signal shown in Figure 40 is "H". Therefore, data can be transmitted continuously if the next transmission data is written in the transmission butter register and TI$_i$ flag is cleared to 0 before the T_{ENDi} signal goes "H".
The bit 3 (TxEPTY$_i$ flag) of UART$_i$ transmit/receive control register 0 changes to "1" at the next cycle after the T_{ENDi} signal goes "H" and changes to "0" when transmission starts. Therefore, this flag can be used to determine whether data transmision is completed.
When the TI$_i$ flag changes from "0" to "1", the interrupt request bit in the UART$_i$ transmissoin interrupt control register is set to "1".

Receive

Receive is enabled when the bit 2 (RE$_i$ flag) of UART$_i$ transmit/receive control register 1 is set. As shown in Figure 42, the frequency divider circuit at the receiving end begin to work when a start bit is arrived and the data is received.

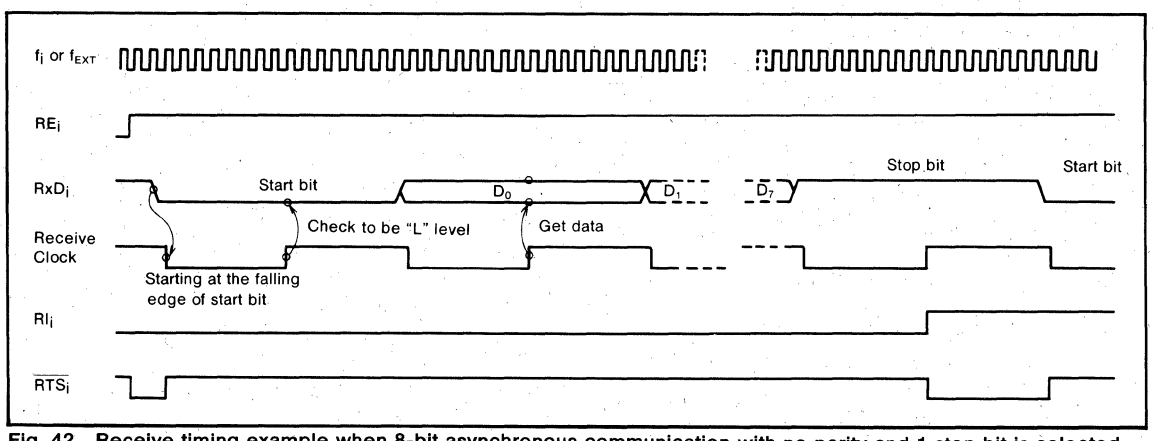


Fig. 42 Receive timing example when 8-bit asynchronous communication with no parity and 1 stop bit is selected

If $\overline{RTS_i}$ output is selected by setting the bit 2 of UART$_i$ transmit/receive control register 0 to "1", the $\overline{RTSi}$ output is "H" when the RE$_i$ flag is "0". When the RE$_i$ flag changes to "1", the $\overline{RTSi}$ output goes "L" to indicate receive ready and returns to "H" once receive has started. In other words, $\overline{RTS_i}$ output can be used to determine externally whether the receive register is ready to receive.

The entire transmission data bits are received when the start bit passes the final bit of the receive block shown in Figure 35. At this point, the contents of the receive register is transferred to the receive buffer register and the bit 3 of UART$_i$ transmit/receive control register 1 is set. In other words, the RI$_i$ flag indicates that the receive buffer register contains data when it is set. If $\overline{RTS_i}$ output is selected, $\overline{RTS_i}$ output goes "L" to indicate that the register is ready to receive the next data.

The interrupt request bit in the UART$_i$ receive interrupt control register is set when the RI$_i$ flag changes from "0" to "1".

The bit 4 (OER$_i$ flag) of UART$_i$ transmission control register 1 is set when the next data is transferred from the receive register to the receive buffer register while the RI$_i$ flag is "1". In other words when an overrun error occurs. If the OER$_i$ flag is "1", it indicates that the next data has been transferred to the receive buffer register before the contents of the receive butter register has been read.

Bit 5 (FER$_i$ flag) is set when the number of stop bits is less than required (framing error).

Bit 6 (PER$_i$ flag) is set when a parity error occurs.

Bit 7 (SUM$_i$ flag) is set when either the OER$_i$ flag, FER$_i$ flag, or the PER$_i$ flag is set. Therefore, the SUM$_i$ flag can be used to determine whether there is an error.

The setting of the RI$_i$ flag, OER$_i$ flag, FER$_i$ flag, and the PERi flag is performed while transferring the contents of the receive register to the receive buffer register. The RI$_i$ OER$_i$, FER$_i$, PER$_i$, and SUM$_i$ flags are cleared when the low order byte of the receive buffer register is read or when the RE$_i$ flag is cleared.

Sleep mode

The sleep mode is used to communicate only between certain microcomputers when multiple microcomputers are connected through serial I/O.

The sleep mode is entered when the bit 7 of UART$_i$ transmit/receive mode register is set.

The operation of the sleep mode for an 8-bit asynchronous communication is described below.

When sleep mode is selected, the contents of the receive register is not transferred to the receive buffer register if bit 7 (bit 6 if 7-bit asynchronous communication and bit 8 if 9-bit asychronous communication) of the received data is "0". Also the RI$_i$, OER$_i$, FER$_i$, PER$_i$, and the SUM$_i$ flag are unchanged. Therefore, the interrupt request bit of the UART$_i$ receive interrupt control register is also unchanged.

Normal receive operation takes place when bit 7 of the received data is "1".

The following is an example of how the sleep mode can be used.

The main microcomputer first sends data with bit 7 set to "1" and bits 0 to 6 set to the address of the subordinate microcomputer which wants to communicate with. Then all subordinate microcomputers receive the same data. Each subordinate microcomputer checks the received data, clears the sleep bit if bits 0 to 6 are its own address and sets the sleep bit if not. Next the main microcomputer sends data with bit 7 cleared. Then the microcomputer with the sleep bit cleared will receive the data, but the microcomputer with the sleep bit set will not. In this way, the main microcomputer is able to communicate with only the designated microcomputer.

A-D CONVERTER

The A-D converter is an 8-bit successive approximation converter.

Figure 43 shows a block diagram of the A-D converter and Figure 44 shows the bit configuration of the A-D control register. The frequency of the A-D converter operating clock ϕ_{AD} is selected by the bit 7 of the A-D control register. When bit 7 is "0", ϕ_{AD} is the clock frequency divided by 8. That is, $\phi_{AD}=f(X_{IN})/8$. When bit 7 is "1", ϕ_{AD} is the clock frequency divided by 4 and ϕ_{AD} is $=f(X_{IN})/4$. The ϕ_{AD} during A-D conversion must be 250kHz minimum because the comparator consists of a capacity coupling amplifier.

The operating mode is selected by the bits 3 and 4 of A-D control register. The available operating modes are one-shot, repeat, single sweep, and repeat sweep.

The bit of data direction register bit corresponding to the A-D converter pin must be "0" (input mode) because the analog input port is shared with port P7.

The operation of each mode is described below.

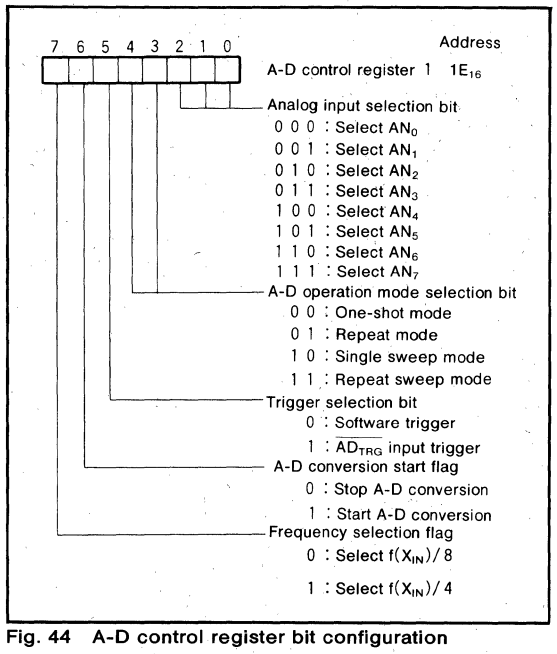


Fig. 44 A-D control register bit configuration

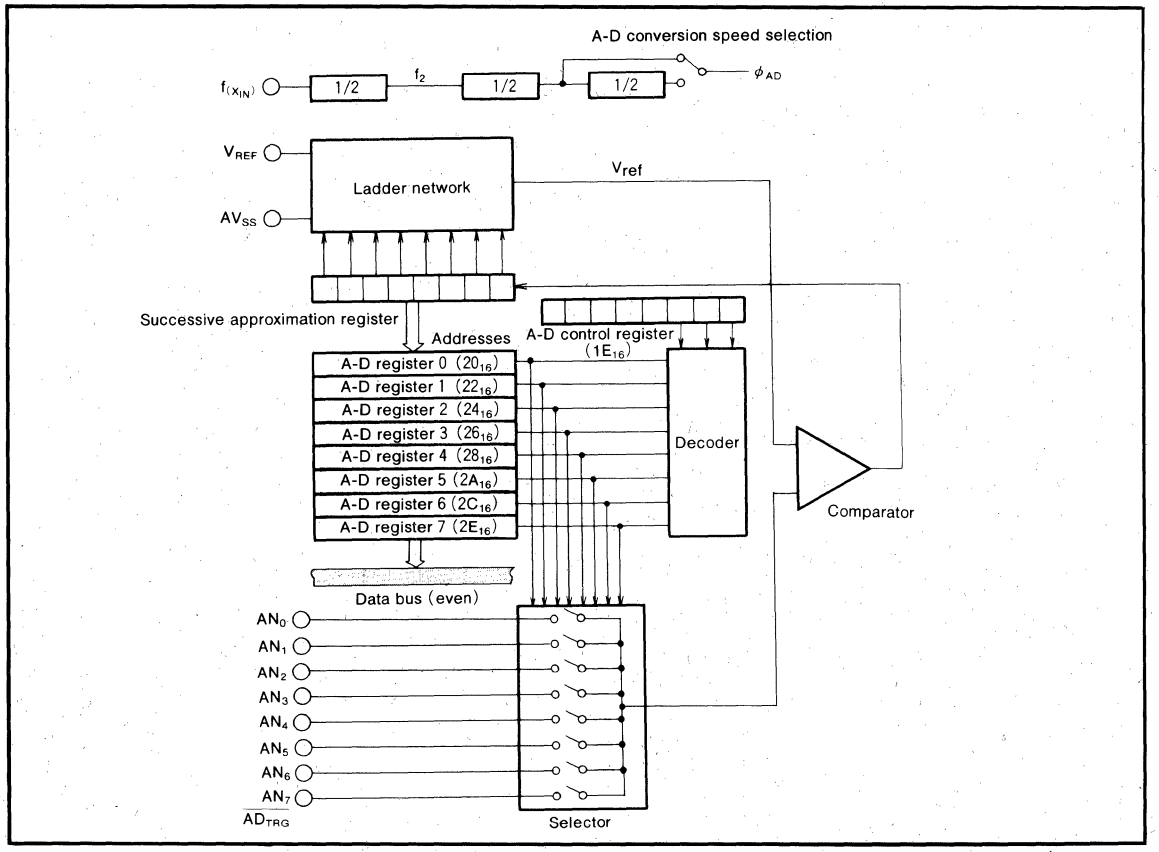


Fig. 43 A-D converter block diagram

(1) One-shot mode [00]

The A-D conversion pins are selected with the bit 0 to 2 of A-D control register. A-D conversion can be started by a software trigger or by an external trigger.

A software trigger is selected when the bit 5 of A-D control register is "0" and an external trigger is selected when it is "1".

When a software trigger is selected, A-D conversion is started when bit 6 (A-D conversion start flag) is set. A-D conversion ends after 57 ϕ_{AD} cycles and an interrupt request bit is set in the A-D conversion interrupt control register. At the same time, A-D control register bit 6 (A-D conversion start flag) is cleared and A-D conversion stops. The result of A-D conversion is stored in the A-D register corresponding to the selected pin.

If an external trigger is selected, A-D conversion starts when the A-D conversion start flag is "1" and the $\overline{AD_{TRG}}$ input changes from "H" to "L". In this case, the pins that can be used for A-D conversion are AN_0 to AN_6 because the $\overline{AD_{TRG}}$ pin is shared with the analog voltage input pin AN_7.

The operation is the same as with software trigger except that the A-D conversion start flag is not cleared after A-D conversion and a retrigger can be available during A-D conversion.

(2) Repeat mode [01]

The operation of this mode is the same as the operation of one-shot mode except that when A-D conversion of the selected pin is complete and the result is stored in the A-D register, conversion does not stop, but is repeated. Also, no interrupt request is issued in this mode. Furthermore, if software trigger is selected, the A-D conversion start flag is not cleared. The contents of the A-D register can be read at any time.

(3) Single sweep mode [10]

In the sweep mode, the number of analog input pins to be swept can be selected. Analog input pins are selected by bits 1 and 0 of the A-D sweep pin selection register ($1F_{16}$ address) shown in Figure 45. Two pins, four pins, six pins, or eight pins can be selected as analog input pins, depending on the contents of these bits.

A-D conversion is performed only for selected input pins. After A-D conversion is performed for input of AN_0 pin, the conversion result is stored in A-D register 0, and in the same way, A-D conversion is performed for selected pins one after another. After A-D conversion is performed for all selected pins, the sweep is stopped.

A-D conversion can be started with a software trigger or with an external trigger input. A software trigger is selected when bit 5 is "0" and an external trigger is selected when it is "1".

When a software trigger is selected, A-D conversion is started when A-D control register bit 6 (A-D conversion start flag) is set. When A-D conversion of all selected pins end, an interrupt request bit is set in the A-D conversion interrupt control register. At the same time, A-D control register bit 6 (A-D conversion start flag) is cleared and A-D conversion stops.

When an external trigger is selected, A-D conversion starts when the A-D conversion start flag is "1" and the $\overline{AD_{TRG}}$ input changes from "H" to "L". In this case, the A-D conversion result of the trigger input itself is stored in the A-D register 7 because the $\overline{AD_{TRG}}$ pin is shared with AN_7 pin.

The operation is the same as done by software trigger except that the A-D conversion start flag is not cleared after A-D conversion and a retrigger can be available during A-D conversion.

(4) Repeat sweep mode [11]

The difference with the single sweep mode is that A-D conversion does not stop after converting from the AN_0 pin to the selected pins, but repeats again from the AN_0 pin. The repeat is performed among the selected pins. Also, no interrupt request is generated. Furthermore, if software trigger is selected, the A-D conversion start flag is not cleared. The A-D register can be read at any time.

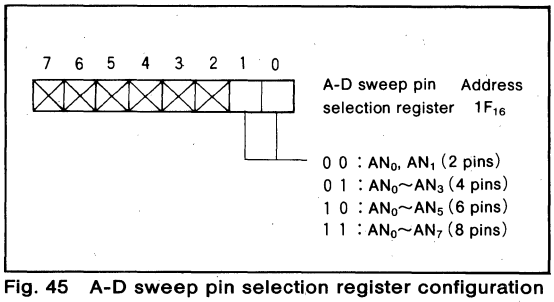


Fig. 45 A-D sweep pin selection register configuration

WATCHDOG TIMER

The watchdog timer is used to detect unexpected execution sequence caused by software run-away.

Figure 46 shows a block diagram of the watchdog timer. The watchdog timer consists of a 12-bit binary counter.

The watchdog timer counts the clock frequency divided by 32 (f_{32}) or by 512 (f_{512}). Whether to count f_{32} or f_{512} is determined by the watchdog timer frequency selection flag shown in Figure 47. f_{512} is selected when the flag is "0" and f_{32} is selected when it is "1". The flag is cleared after reset.

FFF_{16} is set in the watchdog timer when "L" or $2V_{CC}$ is applied to the $\overline{RESET}$ pin, STP instruction is executed, data is written to the watchdog timer, or the most significant bit of the watchdog timer become "0".

After FFF_{16} is set in the watchdog timer, the contents of watchdog timer is decremented by one at every cycle of selected frequency f_{32} or f_{512}, and after 2048 counts, the most significant bit of watchdog timer become "0", and a watchdog timer interrupt request bit is set, and FFF_{16} is preset in the watchdog timer.

Normally, a program is written so that data is written in the watchdog timer before the most significant bit of the watchdog timer become "0". If this routine is not executed due to unexpected program execution, the most significant bit of the watchdog timer become eventually "0" and an interrupt is generated.

The processor can be reset by setting the bit 3 (software reset bit) of processor mode register described in Figure 10 in the interrupt section and generating a reset pulse.

The watchdog timer stops its function when the $\overline{RESET}$ pin voltage is raised to double the V_{CC} voltage.

The watchdog timer can also be used to recover from when the clock is stopped by the STP instruction. Refer to the section on clock generation circuit for more details.

The watchdog timer hold the contents during a hold state and the frequency is stopped to input.

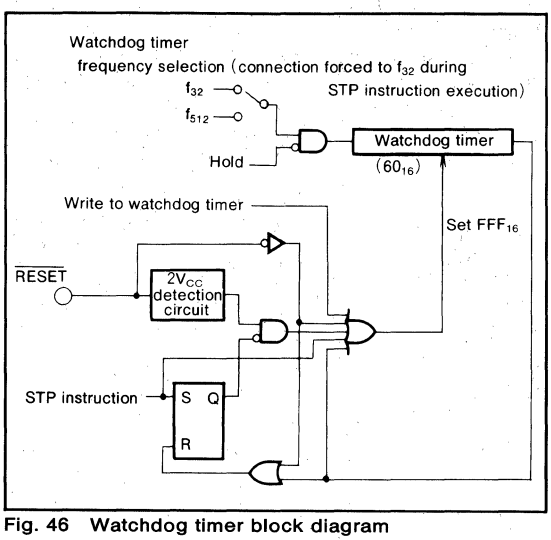


Fig. 46 Watchdog timer block diagram

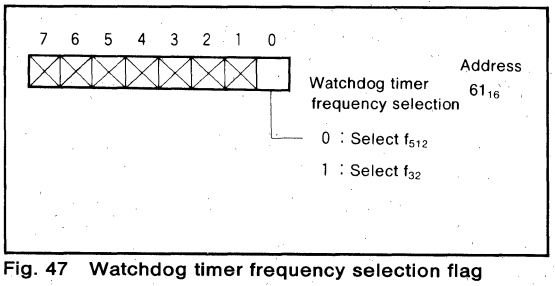


Fig. 47 Watchdog timer frequency selection flag

RESET CIRCUIT

Reset occurs when the $\overline{\text{RESET}}$ pin is returned to "H" level after holding it at "L" level when the power voltage is at 5V ±10%. Program execution starts at the address formed by setting the address pins $A_{23} \sim A_{16}$ to 00_{16}, $A_{15} \sim A_8$ to the contents of address $FFFF_{16}$, and $A_7 \sim A_0$ to the contents of address $FFFE_{16}$.

Figure 48 shows the status of the internal registers when a reset occurs.

Figure 49 shows an example of a reset circuit. The reset input voltage must be held 0.9V or lower when the power voltage reaches 4.5V.

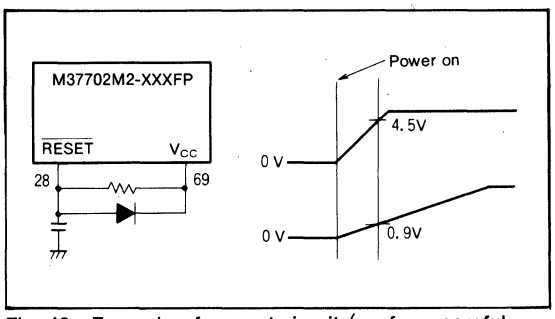


Fig. 49 Example of a reset circuit (perform careful evaluation at the system design level before using)

		Address	
(1)	Port P0 data directional register	$(04_{16})\cdots$	00_{16}
(2)	Port P1 data directional register	$(05_{16})\cdots$	00_{16}
(3)	Port P2 data directional register	$(08_{16})\cdots$	00_{16}
(4)	Port P3 data directional register	$(09_{16})\cdots$	××××0000
(5)	Port P4 data directional register	$(0C_{16})\cdots$	00_{16}
(6)	Port P5 data directional register	$(0D_{16})\cdots$	00_{16}
(7)	Port P6 data directional register	$(10_{16})\cdots$	00_{16}
(8)	Port P7 data directional register	$(11_{16})\cdots$	00_{16}
(9)	Port P8 data directional register	$(14_{16})\cdots$	00_{16}
(10)	A-D control register	$(1E_{16})\cdots$	00000???
(11)	A-D sweep pin selection register	$(1F_{16})\cdots$	××××××11
(12)	UART 0 Transmit/Receive mode register	$(30_{16})\cdots$	00_{16}
(13)	UART 1 Transmit/Receive mode register	$(38_{16})\cdots$	00_{16}
(14)	UART 0 Transmit/Receive control register 0	$(34_{16})\cdots$	××××1000
(15)	UART 1 Transmit/Receive control register 0	$(3C_{16})\cdots$	××××1000
(16)	UART 0 Transmit/Receive control register 1	$(35_{16})\cdots$	00000010
(17)	UART 1 Transmit/Receive control register 1	$(3D_{16})\cdots$	00000010
(18)	Count start flag	$(40_{16})\cdots$	00_{16}
(19)	One-shot start flag	$(42_{16})\cdots$	××××0000
(20)	Up-down flag	$(44_{16})\cdots$	00_{16}
(21)	Timer A0 mode register	$(56_{16})\cdots$	00_{16}
(22)	Timer A1 mode register	$(57_{16})\cdots$	00_{16}
(23)	Timer A2 mode register	$(58_{16})\cdots$	00_{16}
(24)	Timer A3 mode register	$(59_{16})\cdots$	00_{16}
(25)	Timer A4 mode register	$(5A_{16})\cdots$	00_{16}
(26)	Timer B0 mode register	$(5B_{16})\cdots$	001×0000
(27)	Timer B1 mode register	$(5C_{16})\cdots$	001×0000
(28)	Timer B2 mode register	$(5D_{16})\cdots$	001×0000

		Address	
(29)	Processor mode register	$(5E_{16})\cdots$	00_{16}
(30)	Watchdog timer	$(60_{16})\cdots$	FFF_{16}
(31)	Watchdog timer frequency selection flag	$(61_{16})\cdots$	×××××××0
(32)	A-D conversion interrupt control register	$(70_{16})\cdots$	××××0000
(33)	UART 0 transmission interrupt control register	$(71_{16})\cdots$	××××0000
(34)	UART 0 receive interrupt control register	$(72_{16})\cdots$	××××0000
(35)	UART 1 transmission interrupt control register	$(73_{16})\cdots$	××××0000
(36)	UART 1 receive interrupt control register	$(74_{16})\cdots$	××××0000
(37)	Timer A0 interrupt control register	$(75_{16})\cdots$	××××0000
(38)	Timer A1 interrupt control register	$(76_{16})\cdots$	××××0000
(39)	Timer A2 interrupt control register	$(77_{16})\cdots$	××××0000
(40)	Timer A3 interrupt control register	$(78_{16})\cdots$	××××0000
(41)	Timer A4 interrupt control register	$(79_{16})\cdots$	××××0000
(42)	Timer B0 interrupt control register	$(7A_{16})\cdots$	××××0000
(43)	Timer B1 interrupt control register	$(7B_{16})\cdots$	××××0000
(44)	Timer B2 interrupt control register	$(7C_{16})\cdots$	××××0000
(45)	$\overline{\text{INT}}_0$ interrupt control register	$(7D_{16})\cdots$	××000000
(46)	$\overline{\text{INT}}_1$ interrupt control register	$(7E_{16})\cdots$	××000000
(47)	$\overline{\text{INT}}_2$ interrupt control register	$(7F_{16})\cdots$	××000000
(48)	Processor status register PS		000??000??
(49)	Program bank register PG		00_{16}
(50)	Program counter PC_H		Content of $FFFF_{16}$
(51)	Program counter PC_L		Content of $FFFE_{16}$
(52)	Direct page register DPR		0000_{16}
(53)	Data bank register DT		00_{16}

Contents of other registers and RAM are not initialized and should be initialized by software.

Fig. 48 Microcomputer internal status during reset

INPUT/OUTPUT PINS

Ports P8 to P0 all have a data direction register and each bit can be programmed for input or output. A pin becomes an output pin when the corresponding data direction register is set and an input pin when it is cleared.

When pin programmed for output, the data is written to the port latch and it is output to the output pin. When a pin is programmed for output, the contents of the port latch is read instead of the value of the pin. Therefore, a previously output value can be read correctly even when the output "L" voltage is raised due to reasons such as directly driving an LED.

A pin programmed for input is floating and the value input to the pin can be read. When a pin is programmed for input, the data is written only in the port latch and the pin stays floating.

If an input/output pin is not used as an output port, clear the bit of the corresponding data direction register so that the pin become input mode.

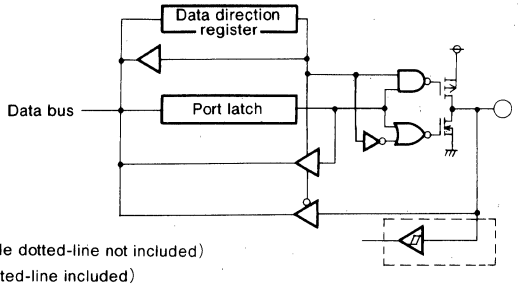
Figure 50 shows a block diagram of ports P8 to P0 in single-chip mode and the $\overline{E}$ pin output.

In memory expansion mode, microprocessor mode, and evaluation chip mode, ports P4 to P0 are also used as address, data, and control signal pins.

Refer to the section on processor modes for more details.

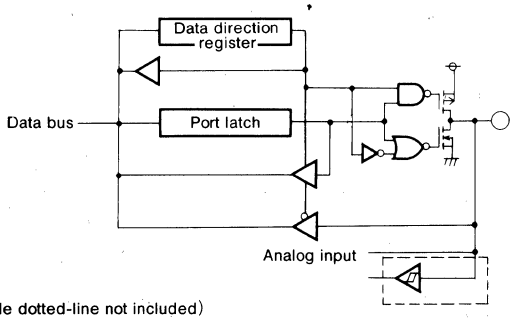
• Port $P0_7$~$P0_0$, $P1_7$~$P1_0$, $P2_7$~$P2_0$, $P3_3$~$P3_0$, $P4_6$~$P4_2$ (Inside dotted-line not included)

Port $P4_0$, $P4_1$, $P4_7$, $P5_7$, $P6_7$~$P6_1$, $P8_2$, $P8_6$ (Inside dotted-line included, but $P8_2$, $P8_6$ are without hysterisis)



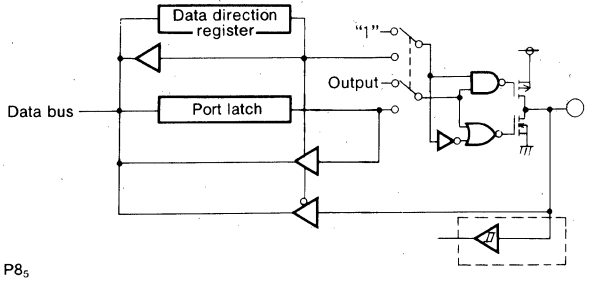
• Port $P7_6$~$P7_0$ (Inside dotted-line not included)

• Port $P7_7$ (Inside dotted-line included)

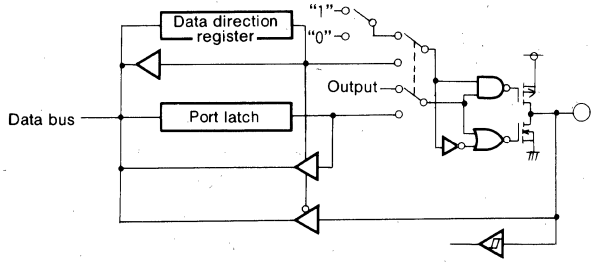


• Port $P8_3$, $P8_7$ (Inside dotted-line not included)

Port $P5_0$~$P5_6$, $P6_0$ (Inside dotted-line included)



• Port $P8_0$, $P8_1$, $P8_4$, $P8_5$



• $\overline{E}$

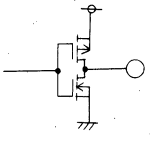


Fig. 50 Block diagram for ports P8 to P0 in single-chip mode and the $\overline{E}$ pin output

PROCESSOR MODE

The bits 0 and 1 of processor mode register as shown in Figure 51 are used to select any mode of single-chip mode, memory expansion mode, microprocessor mode, and evaluation chip mode.

Ports P3 to P0 and a part of port P4 are used as address, data, and control signal I/O pins except in single-chip mode.

Figure 52 shows the functions of ports P4 to P0 in each mode.

The external memory area changes when the mode changes.

Figure 53 shows the memory map for each mode.

Refer to Figure 1 for the memory map of the single-chip mode. The external memory area can be accessed except in single-chip mode. The accessing of the external memory is affected by the BYTE pin and the bit 2 (wait bit) of processor mode register . These will be described next.

•BYTE pin

When accessing the external memory, the level of the BYTE pin is used to determine whether to use the data bus as 8-bit width or 16-bit width.

The data bus width is 8 bits when the level of the BYTE pin is "H" and port P2 becomes the data I/O pin.

The data bus width is 16 bits when the level of the BYTE pin is "L" and ports P1 and P2 become the data I/O pins.

When accessing the internal memory, the data bus width is always 16 bits regardless of the BYTE pin level.

An exclusive mode in the evaluation chip mode allows the BYTE pin level to be set to $2 \cdot V_{CC}$. In this case, the operation is slightly different from the above. This is described in the evaluation chip mode section.

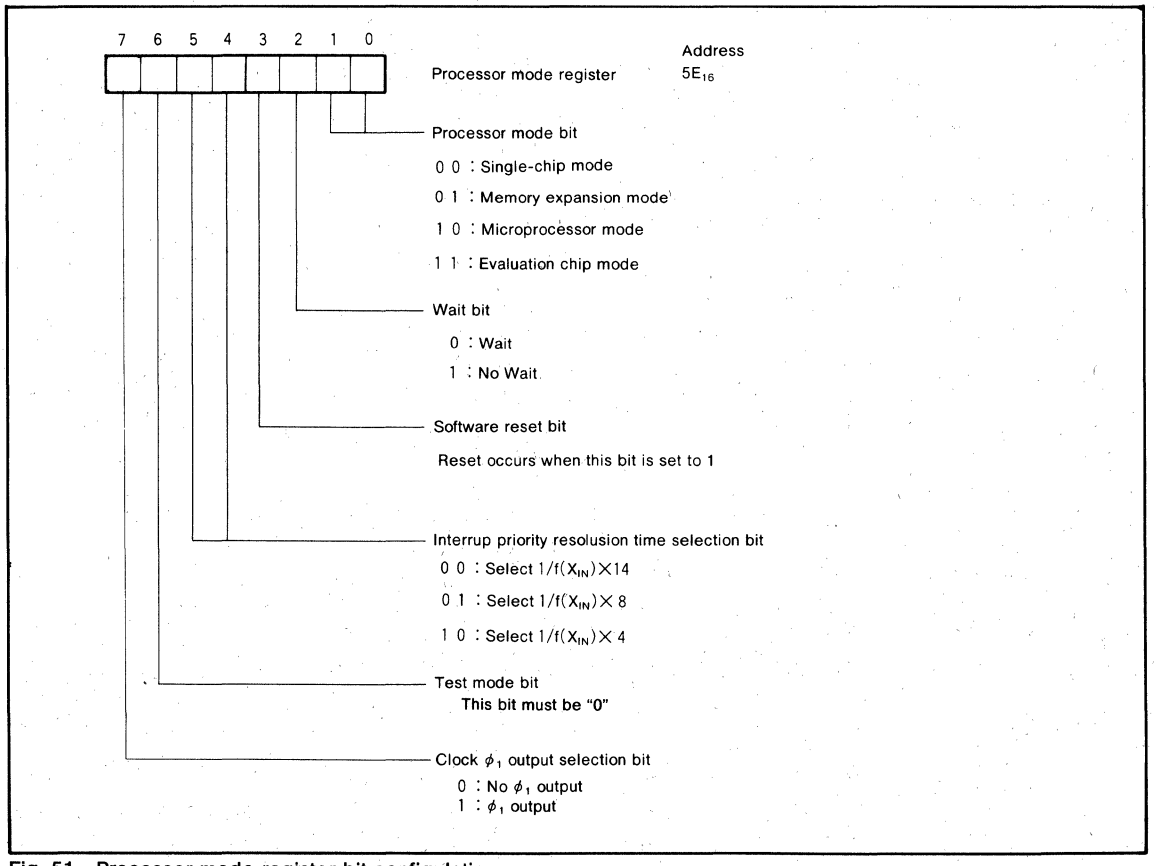


Fig. 51 Processor mode register bit configulation

Port \ Mode	CM_1 = 0, CM_0 = 0 Single-chip Mode	CM_1 = 0, CM_0 = 1 Memory Expansion Mode	CM_1 = 1, CM_0 = 0 Microprocessor Mode	CM_1 = 1, CM_0 = 1 Evaluation Chip Mode
Port P0	$\overline{E}$ $P0_7$ ~ $P0_0$: I/O Port	$\overline{E}$ $P0_7$ ~ $P0_0$: Address A_7~A_0	Same as left	Same as left
Port P1, BYTE= "L"	$\overline{E}$ $P1_7$ ~ $P1_0$: I/O Port	$\overline{E}$ A_{15}~A_8 $P1_7$ ~ $P1_0$: Address / Data(odd)	Same as left	Same as left
Port P1, BYTE= "H" or 2·V_{CC} (Evaluation chip mode only.)	$P1_7$ ~ $P1_0$: I/O Port	$\overline{E}$ $P1_7$ ~ $P1_0$: Address A_{15}~A_8	Same as left	$\overline{E}$ A_{15}~A_8 Address / Data(odd) Port P4, P5 and their direction registers are treated as 16-bit wide bus. If BYTE=2·V_{CC}, the internal ROM area is also treated as 16-bit wide bus.
Port P2, BYTE= "L"	$\overline{E}$ $P2_7$ ~ $P2_0$: I/O Port	$\overline{E}$ A_{23}~A_{16} $P2_7$ ~ $P2_0$: Address / Data (even)	Same as left	Same as left
Port P2, BYTE= "H" or 2·V_{CC} (Evaluation chip mode only.)		$\overline{E}$ A_{23}~A_{16} $P2_7$ ~ $P2_0$: Address / Data (even, odd)	Same as left	$\overline{E}$ A_{23}~A_{16} $P2_7$ ~ $P2_0$: Address / Data (even, odd) Same as for Port P1
Port P3	$\overline{E}$ $P3_3$ ~ $P3_0$: I/O Port	$\overline{E}$ $P3_3$: $\overline{HLDA}$ $P3_2$: ALE $P3_1$: $\overline{BHE}$ $P3_0$: R/$\overline{W}$	Same as left	Same as left
Port P4	$\overline{E}$ $P4_7$ ~ $P4_0$: I/O Port * When processor mode register bit 7 = "0"	$\overline{E}$ $P4_7$ ~ $P4_2$: I/O Port $P4_1$: $\overline{RDY}$ $P4_0$: $\overline{HOLD}$ * When processor mode register bit 7 = "0"	—	$\overline{E}$ $P4_7$: $\overline{DBC}$ $P4_6$: VPA $P4_5$: VDA $P4_4$: QCL $P4_3$: MX $P4_2$: ϕ_1 $P4_1$: $\overline{RDY}$ $P4_0$: $\overline{HOLD}$
	$P4_2$: ϕ_1 Same as above except $P4_2$ * When processor mode register bit 7 = "1"	$P4_2$: ϕ_1 Same as above except $P4_2$ * When processor mode register bit 7 = "1"	Same as left in spite of processor mode register bit 7	

Fig. 52 Processor mode and ports P4 to P0 functions

●Wait bit

As shown in Figure 54, when the external memory area is accessed with the processor mode register bit 2 (wait bit) cleared to "0", the "L" width of $\overline{E}$ signal becomes twice compared with no wait (the wait bit is "1"). The wait bit is cleared to "0" at reset.

The accessing of internal memory area is performed in no wait mode regardless of the wait bit.

The processor modes are described below.

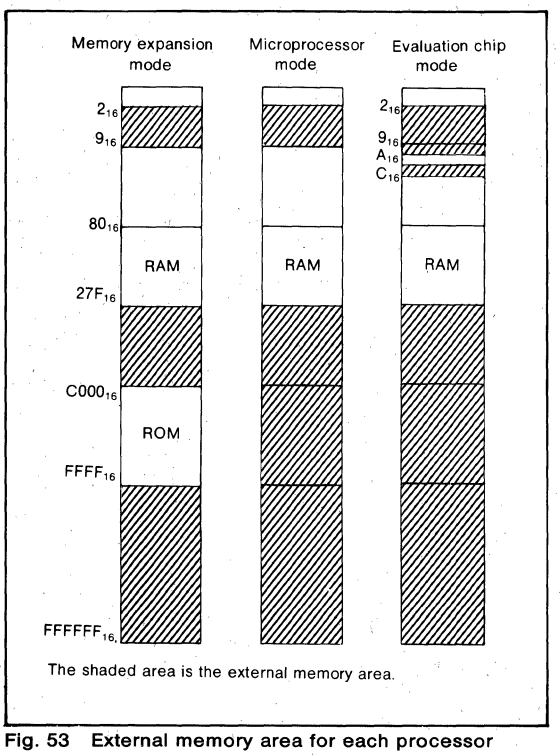


Fig. 53 External memory area for each processor mode

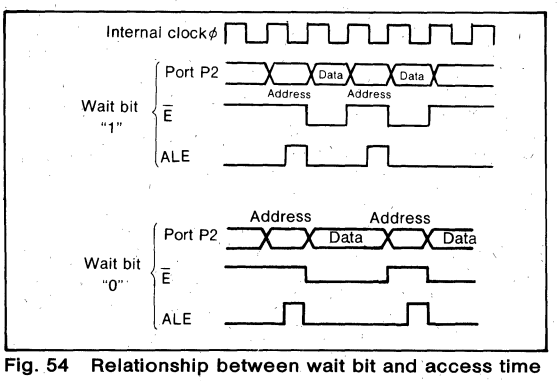


Fig. 54 Relationship between wait bit and access time

(1) Single-chip mode [00]

single-chip mode is entered by connecting the CNV_{SS} pin to V_{SS} and starting from reset. Ports P4 to P0 all function as normal I/O ports. Port $P4_2$ can be the ϕ_1 output pin divided the clock to X_{IN} pin by 2 by setting bit 7 of processor mode register to "1"

(2) Memory expansion mode [01]

Memory expansion mode is entered by setting the processor mode bits to "01" after connecting the CNV_{SS} pin to V_{SS} and starting from reset.

Port P0 becomes an address output pin and loses its I/O port function.

Port P1 has two functions depending on the level of the BYTE pin.

When the BYTE pin level is "L", port P1 functions as an address output pin while $\overline{E}$ is "H" and as an odd address data I/O pin while $\overline{E}$ is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L". In this case the I/O port function is lost.

When the BYTE pin level "H", port P1 functions as an address output pin and loses its I/O port function.

Port P2 has two functions depending on the level of the BYTE pin.

When the BYTE pin level is "L", port P2 functions as an address output pin while $\overline{E}$ is "H" and as an even address data I/O pin while $\overline{E}$ is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

When the BYTE pin level is "H", port P2 functions as an address output pin while $\overline{E}$ is "H" and as an even and odd address data I/O pin while $\overline{E}$ is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L". In this case the I/O port function is lost.

Ports $P3_0$, $P3_1$, $P3_2$, and $P3_3$ become R/$\overline{W}$, $\overline{BHE}$, ALE, and $\overline{HLDA}$ output pin respectively and lose their I/O port functions.

R/$\overline{W}$ is a read/write signal which indicates a read when it is "H" and a write when it is "L".

$\overline{BHE}$ is a byte high enable signal which indicates that an odd address is accessed when it is "L".

Therefore, two bytes at even and odd addresses are accessed simultaneously if address A_0 is "L" and $\overline{BHE}$ is "L".

ALE is an address latch enable signal used to latch the address signal from a multiplexed signal of address and data. The latch is transparent while ALE is "H" to let the address signal pass through and held while ALE is "L".

$\overline{HLDA}$ is a hold acknowledge signal and is used to notify externally when the microcomputer receives $\overline{HOLD}$ input and enters into hold state.

Ports $P4_0$ and $P4_1$ become $\overline{HOLD}$ and $\overline{RDY}$ input pin respectively and lose their output pin function, but the input pin function remains.

$\overline{HOLD}$ is a hold request signal. It is an input signal used to put the microcomputer in hold state. $\overline{HOLD}$ input is accepted when the internal clock ϕ falls from "H" level to "L" level while the bus is not used. Ports P0, P1, P2, $P3_0$, and $P3_1$ are floating while the microcomputer stays in hold state. These ports are floating after one cycle of the internal clock ϕ later than $\overline{HLDA}$ signal changes to "L" level. At the removing of hold state, these ports are removed from floating state after one cycle of ϕ later than $\overline{HLDA}$ signal changes to "H" level.

$\overline{RDY}$ is a ready signal. If this signal goes "L", the internal clock ϕ stops at "L". When ϕ_1 output from port $P4_2$ is selected by setting bit 7 of processor mode register to "1", ϕ_1 output keeps on. $\overline{RDY}$ is used when slow external memory is attached.

(3) Microprocessor mode [10]

Microprocessor mode is entered by connecting the CNV_{SS} pin to V_{CC} and starting from reset. It can also be entered by programming the processor mode bits to "10" after connecting the CNV_{SS} pin to V_{SS} and starting from reset. This mode is similar to memory expansion mode except that internal ROM is disabled and an external memory is required, and ϕ_1 from port $P4_2$ is always output in spite of bit 7 of processor mode register.

(4) Evaluation chip mode [11]

Evaluation chip mode is entered by applying voltage twice the V_{CC} voltage to the CNV_{SS} pin. This mode is normally used for evaluation tools.

The functions of ports P0 and P3 are the same as in memory expansion mode.

Port P1 functions as an address output pin while $\overline{E}$ is "H" and as data I/O pin of odd addresses while $\overline{E}$ is "L" regardless of the BYTE pin level. However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

Port P2 function as an address output pin while $\overline{E}$ is "H" and as data I/O pin of even addresses while $\overline{E}$ is "L" when the BYTE pin level is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

When the BYTE pin level is "H" or $2 \cdot V_{CC}$, port P2 functions as an address output pin while $\overline{E}$ is "H" and as data I/O pin of even and odd addresses while $\overline{E}$ is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

Port P4 and its data direction register which are located at address $0A_{16}$ and $0C_{16}$ are treated differently in evaluation chip mode. When these addresses are accessed, the data bus width is treated as 16 bits regardless of the BYTE pin level, and the access cycle is treated as internal memory regardless of the wait bit.

When a voltage twice the V_{CC} voltage is applied to the BYTE pin, the addresses corresponding to the internal ROM area are also treated as 16-bit data bus.

The functions of ports $P4_0$ and $P4_1$ are the same as in memory expansion mode.

Ports $P4_2$ to $P4_6$ become ϕ_1, MX, QCL, VDA, and VPA output pins respectively. Port $P4_7$ becomes the $\overline{DBC}$ input pin.

ϕ_1 from port $P4_2$ divided the clock to X_{IN} pin by 2 is always output in spite of bit 7 of processor mode register.

The MX signal normally contains the contents of flag m, but the contents of flag x is output if the CPU is using flag x.

QCL is the queue buffer clear signal. It becomes "H" when the instruction queue buffer is cleared, for example, when a jump instruction is executed.

VDA is the valid data address signal. It becomes "H" while the CPU is reading data from data buffer or writing data to data buffer. It also becomes "H" when the first byte of the instruction (operation code) is read from the instruction queue buffer.

VPA is the valid program address signal. It becomes "H" while the CPU is reading an instruction code from the instruction queue buffer.

$\overline{DBC}$ is the debug control signal and is used for debugging.

Table 5 shows the relationship between the CNV_{SS} pin input levels and processor modes.

Table 5. Relationship between the CNV_{SS} pin input levels and processor modes

CNV_{SS}	Mode	Description
V_{SS}	• Single-chip • Memory expansion • Microprocessor • Evaluation chip	Single-chip mode upon starting after reset. Other modes can be selected by changing the processor mode bit by software.
V_{CC}	• Microprocessor • Evaluation chip	Microprocessor mode upon starting after reset. Evaluation chip mode can be selected by changing the processor mode bit by software.
$2 \cdot V_{CC}$	• Evaluation chip	• Evaluation chip mode only.

CLOCK GENERATING CIRCUIT

Figure 55 shows a block diagram of the clock generator.

When an STP instruction is executed, the internal clock ϕ stops oscillating at "L" level. At the same time, FFF_{16} is written to watchdog timer and the watchdog timer input connection is forced to f_{32}. This connection is broken and connected to the input determined by the watchdog timer frequency selection flag when the most significant bit of the watchdog timer is cleared or reset.

Oscillation resumes when an interrupt is received, but the internal clock ϕ remains at "L" level until the most significant bit of the watchdog timer is cleared. This is to avoid the unstable interval at the start of oscillation when using a ceramic resonator.

When a WIT instruction is executed, the internal clock ϕ stops at "L" level, but the oscillator does not stop. The clock is restarted when an interrupt is received. Instructions can be executed immediately because the oscillator is not stopped.

The stop or wait state is released when an interrupt is received or when reset is issued. Therefore, interrupts must be enabled before executing a STP or WIT instruction.

Figure 56 shows a circuit example using a ceramic (or quartz crystal) resonator. Use the manufacturer's recommended values for constants such as capacitance which differ for each resonator. Figure 57 shows an example of using an external clock signal.

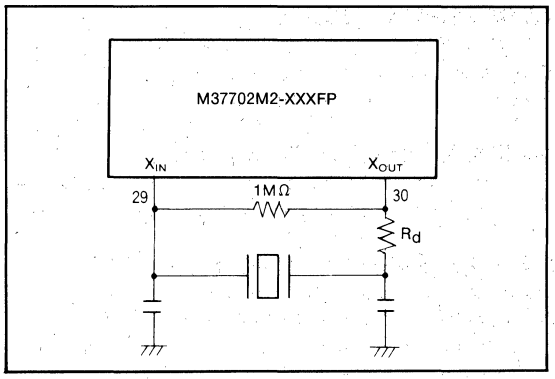


Fig. 56 Circuit using a ceramic resonator

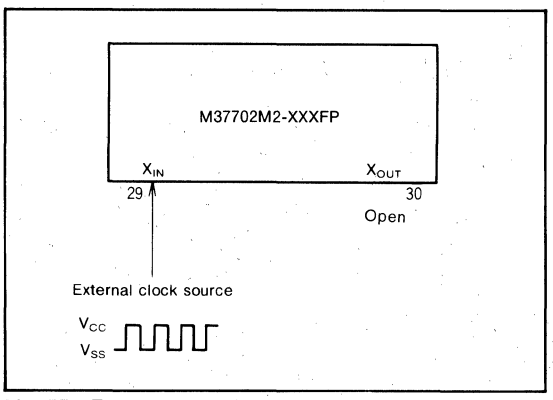


Fig. 57 External clock input circuit

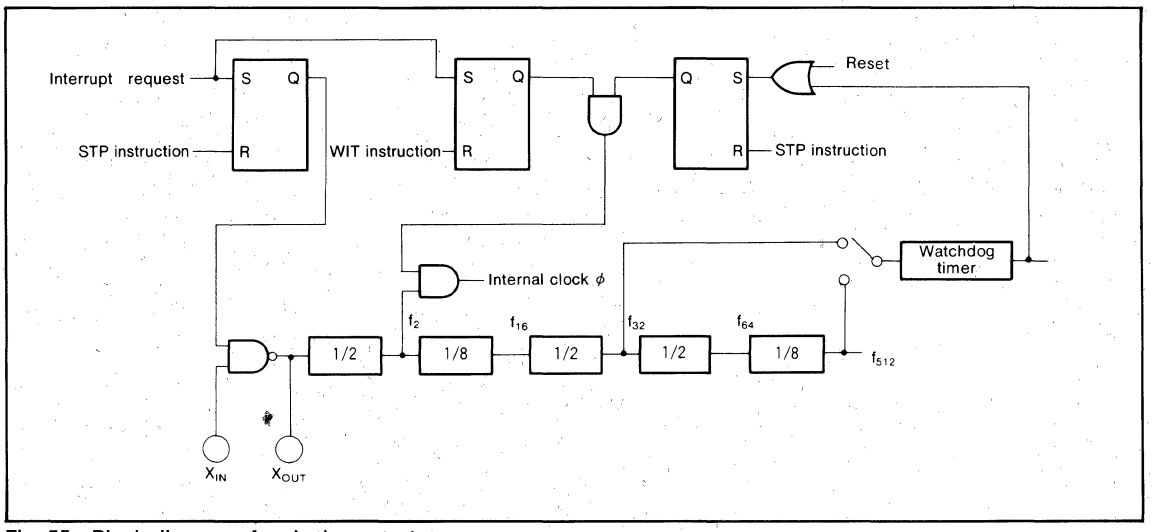


Fig. 55 Block diagram of a clock generator

ADDRESSING MODES

The M37702M2-XXXFP has 28 powerful addressing modes. Refer to the MELPS 7700 addressing mode description for the details of each addressing mode.

MACHINE INSTRUCTION LIST

The M37702M2-XXXFP has 103 machine instructions. Refer to the MELPS 7700 machine instruction list for details.

DATA REQUIRED FOR MASK ORDERING

Please send the following data for mask orders.

(1) M37702M2-XXXFP mask ROM order confirmation form
(2) 80P6N mark specification form
(3) ROM data (EPROM 3 sets)

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Supply voltage		−0.3∼7	V
AV_{CC}	Analog supply voltage		−0.3∼7	V
V_I	Input voltage $\overline{RESET}$, CNV_{SS}, BYTE		−0.3∼12	V
V_I	Input voltage $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_3$, $P4_0$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, V_{REF}, X_{IN}		−0.3∼V_{CC}+0.3	V
V_O	Output voltage $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_3$, $P4_0$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, X_{OUT}, $\overline{E}$		−0.3∼V_{CC}+0.3	V
P_d	Power dissipation	T_a=25℃	300	mW
T_{opr}	Operating temperature		−20∼85	℃
T_{stg}	Storage temperature		−40∼150	℃

RECOMMENDED OPERATING CONDITIONS (V_{CC}=5V±10%, T_a=−20∼85℃, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
V_{CC}	Supply voltage	4.5	5.0	5.5	V
AV_{CC}	Analog supply voltage		V_{CC}		V
V_{SS}	Supply voltage		0		V
AV_{SS}	Analog supply voltage		0		V
V_{IH}	High-level input voltage $P0_0$∼$P0_7$, $P3_0$∼$P3_3$, $P4_0$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	0.8V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage $P1_0$∼$P1_7$, $P2_0$∼$P2_7$ (in single-chip mode)	0.8V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage $P1_0$∼$P1_7$, $P2_0$∼$P2_7$ (in memory expansion mode and microprocessor mode)	0.5V_{CC}		V_{CC}	V
V_{IL}	Low-level input voltage $P0_0$∼$P0_7$, $P3_0$∼$P3_3$, $P4_0$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	0		0.2V_{CC}	V
V_{IL}	Low-level input voltage $P1_0$∼$P1_7$, $P2_0$∼$P2_7$ (in single-chip mode)	0		0.2V_{CC}	V
V_{IL}	Low-level input voltage $P1_0$∼$P1_7$, $P2_0$∼$P2_7$ (in memory expansion mode and microprocessor mode)	0		0.16V_{CC}	V
$I_{OH(peak)}$	High-level peak output current $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_3$, $P4_0$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$			−10	mA
$I_{OH(avg)}$	High-level average output current $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_3$, $P4_0$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$			−5	mA
$I_{OL(peak)}$	Low-level peak output current $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_3$, $P4_0$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$			10	mA
$I_{OL(avg)}$	Low-level average output current $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_3$, $P4_0$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$			5	mA
$f(X_{IN})$	External clock frequency input — M37702M2-XXXFP, M37702S1FP			8	MHz
	External clock frequency input — M37702M2AXXXFP, M37702S1AFP			16	MHz
	External clock frequency input — M37702M2BXXXFP, M37702S1BFP			25	MHz

Note 1. Average output current is the average value of a 100ms interval.
2. The sum of $I_{OL(peak)}$ for ports P0, P1, P2, P3, and P8 must be 80mA or less, the sum of $I_{OH(peak)}$ for ports P0, P1, P2, P3, and P8 must be 80mA or less, the sum of $I_{OL(peak)}$ for ports P4, P5, P6, and P7 must be 80mA or less, and the sum of $I_{OH(peak)}$ for ports P4, P5, P6, and P7 must be 80mA or less.

M37702M2-XXXFP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=8MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$	I_{OH}=−10mA	3			V
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$	I_{OH}=−400μA	4.7			V
V_{OH}	High-level output voltage $P3_2$	I_{OH}=−10mA	3.1			V
		I_{OH}=−400μA	4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA	3.4			V
		I_{OH}=−400μA	4.8			
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$	I_{OL}=10mA			2	V
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$	I_{OL}=2mA			0.45	V
V_{OL}	Low-level output voltage $P3_2$	I_{OL}=10mA			1.9	V
		I_{OL}=2mA			0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA			1.6	V
		I_{OL}=2mA			0.4	
$V_{T+}-V_{T-}$	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA4_{IN}$, $TB0_{IN}$~$TB2_{IN}$, $\overline{INT0}$~$\overline{INT2}$, $\overline{AD_{TRG}}$, $\overline{CTS0}$, $\overline{CTS1}$, CLK_0, CLK_1		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis $\overline{RESET}$		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=5V			5	μA
I_{IL}	Low-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=0V			−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	In single-chip mode output only pin is open and other pins are V_{SS} during reset. f(X_{IN})=8MHz, square waveform		6	12	mA
		T_a=25℃ when clock is stopped.			1	μA
		T_a=85℃ when clock is stopped.			20	

A-D CONVERTER CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=8MHz, unless otherwise noted)

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

M37702M2AXXXFP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=16MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits Min.	Limits Typ.	Limits Max.	Unit
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$	I_{OH}=−10mA	3			V
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$	I_{OH}=−400μA	4.7			V
V_{OH}	High-level output voltage $P3_2$	I_{OH}=−10mA	3.1			V
		I_{OH}=−400μA	4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA	3.4			V
		I_{OH}=−400μA	4.8			
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$	I_{OL}=10mA			2	V
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$	I_{OL}=2mA			0.45	V
V_{OL}	Low-level output voltage $P3_2$	I_{OL}=10mA			1.9	V
		I_{OL}=2mA			0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA			1.6	V
		I_{OL}=2mA			0.4	
$V_{T+}-V_{T-}$	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA4_{IN}$, $TB0_{IN}$~$TB2_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, AD_{TRG}, $\overline{CTS_0}$, $\overline{CTS_1}$, CLK_0, CLK_1		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis $\overline{RESET}$		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=5V			5	μA
I_{IL}	Low-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=0V			−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	In single-chip mode output only pin is open and other pins are V_{SS} during reset. — f(X_{IN})=16MHz, square waveform		12	24	mA
		T_a=25℃ when clock is stopped.			1	μA
		T_a=85℃ when clock is stopped.			20	

A-D CONVERTER CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=16MHz, unless otherwise noted)

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

M37702M2BXXXFP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=25MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$	I_{OH}=−10mA	3			V
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$	I_{OH}=−400μA	4.7			V
V_{OH}	High-level output voltage $P3_2$	I_{OH}=−10mA	3.1			V
		I_{OH}=−400μA	4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA	3.4			V
		I_{OH}=−400μA	4.8			
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$	I_{OL}=10mA			2	V
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$	I_{OL}=2mA			0.45	V
V_{OL}	Low-level output voltage $P3_2$	I_{OL}=10mA			1.9	V
		I_{OL}=2mA			0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA			1.6	V
		I_{OL}=2mA			0.4	
$V_{T+}-V_{T-}$	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA4_{IN}$, $TB0_{IN}$~$TB2_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, $\overline{AD_{TRG}}$, $\overline{CTS_0}$, $\overline{CTS_1}$, CLK_0, CLK_1		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis $\overline{RESET}$		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=5V			5	μA
I_{IL}	Low-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=0V			−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	In single-chip mode output only pin is open and other pins are V_{SS} during reset. f(X_{IN})=25MHz, square waveform		19	38	mA
		T_a=25℃ when clock is stopped.			1	μA
		T_a=85℃ when clock is stopped.			20	

A-D CONVERTER CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=25MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	V_{REF}=V_{CC}			8	Bits
—	Absolute accuracy	V_{REF}=V_{CC}			±3	LSB
R_{LADDER}	Ladder resistance	V_{REF}=V_{CC}	2		10	kΩ
t_{CONV}	Conversion time		9.12			μs
V_{REF}	Reference voltage		2		V_{CC}	V
V_{IA}	Analog input voltage		0		V_{REF}	V

TIMING REQUIREMENTS (V_{CC}=5V±10%, V_{SS}=0V, T_a=25℃, unless otherwise noted)

External clock input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
t_C	External clock input cycle time	125		62		40		ns
$t_{W(H)}$	External clock input high-level pulse width	50		25		15		ns
$t_{W(L)}$	External clock input low-level pulse width	50		25		15		ns
t_r	External clock rise time		20		10		8	ns
t_f	External clock fall time		20		10		8	ns

Single-chip mode

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{SU(P0D-E)}$	Port P0 input setup time	200		100		60		ns
$t_{SU(P1D-E)}$	Port P1 input setup time	200		100		60		ns
$t_{SU(P2D-E)}$	Port P2 input setup time	200		100		60		ns
$t_{SU(P3D-E)}$	Port P3 input setup time	200		100		60		ns
$t_{SU(P4D-E)}$	Port P4 input setup time	200		100		60		ns
$t_{SU(P5D-E)}$	Port P5 input setup time	200		100		60		ns
$t_{SU(P6D-E)}$	Port P6 input setup time	200		100		60		ns
$t_{SU(P7D-E)}$	Port P7 input setup time	200		100		60		ns
$t_{SU(P8D-E)}$	Port P8 input setup time	200		100		60		ns
$t_{h(E-P0D)}$	Port P0 input hold time	0		0		0		ns
$t_{h(E-P1D)}$	Port P1 input hold time	0		0		0		ns
$t_{h(E-P2D)}$	Port P2 input hold time	0		0		0		ns
$t_{h(E-P3D)}$	Port P3 input hold time	0		0		0		ns
$t_{h(E-P4D)}$	Port P4 input hold time	0		0		0		ns
$t_{h(E-P5D)}$	Port P5 input hold time	0		0		0		ns
$t_{h(E-P6D)}$	Port P6 input hold time	0		0		0		ns
$t_{h(E-P7D)}$	Port P7 input hold time	0		0		0		ns
$t_{h(E-P8D)}$	Port P8 input hold time	0		0		0		ns

Memory expansion mode and microprocessor mode

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{SU(P1D-E)}$	Port P1 input setup time	60		45		30		ns
$t_{SU(P2D-E)}$	Port P2 input setup time	60		45		30		ns
$t_{SU(RDY-\phi_1)}$	$\overline{RDY}$ input setup time	70		60		55		ns
$t_{SU(HOLD-\phi_1)}$	$\overline{HOLD}$ input setup time	70		60		55		ns
$t_{h(E-P1D)}$	Port P1 input hold time	0		0		0		ns
$t_{h(E-P2D)}$	Port P2 input hold time	0		0		0		ns
$t_{h(\phi_1-RDY)}$	$\overline{RDY}$ input hold time	0		0		0		ns
$t_{h(\phi_1-HOLD)}$	$\overline{HOLD}$ input hold time	0		0		0		ns

Timer A input (Count input in event counter mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	250		125		80		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	125		62		40		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	125		62		40		ns

Timer A input (Gating input in timer mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	1000		500		320		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	500		250		160		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	500		250		160		ns

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

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	500		250		160		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	250		125		80		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	250		125		80		ns

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

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	250		125		80		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	250		125		80		ns

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

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(UP)}$	TAi_{OUT} input cycle time	5000		2500		2000		ns
$t_{W(UPH)}$	TAi_{OUT} input high-level pulse width	2500		1250		1000		ns
$t_{W(UPL)}$	TAi_{OUT} input low-level pulse width	2500		1250		1000		ns
$t_{SU(UP-T_{IN})}$	TAi_{OUT} input setup time	1000		500		400		ns
$t_{h(T_{IN}-UP)}$	TAi_{OUT} input hold time	1000		500		400		ns

Timer B input (Count input in event counter mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	TBi_{IN} input cycle time (one edge count)	250		125		80		ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width (one edge count)	125		62		40		ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width (one edge count)	125		62		40		ns
$t_{C(TB)}$	TBi_{IN} input cycle time (both edges count)	500		250		160		ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width (both edges count)	250		125		80		ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width (both edges count)	250		125		80		ns

Timer B input (Pulse period measurement mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	TBi_{IN} input cycle time	1000		500		320		ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width	500		250		160		ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width	500		250		160		ns

Timer B input (Pulse width measurement mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	TBi_{IN} input cycle time	1000		500		320		ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width	500		250		160		ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width	500		250		160		ns

A-D trigger input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(AD)}$	$\overline{AD_{TRG}}$ input cycle time (minimum allowable trigger)	2000		1000		1000		ns
$t_{W(ADL)}$	$\overline{AD_{TRG}}$ input low-level pulse width	250		125		125		ns

Serial I/O

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(CK)}$	CLK_i input cycle time	500		250		200		ns
$t_{W(CKH)}$	CLK_i input high-level pulse width	250		125		100		ns
$t_{W(CKL)}$	CLK_i input low-level pulse width	250		125		100		ns
$t_{d(C-Q)}$	TxD_i output delay time		150		90		80	ns
$t_{h(C-Q)}$	TxD_i hold time	30		30		30		ns
$t_{su(D-C)}$	RxD_i input setup time	60		30		20		ns
$t_{h(C-D)}$	RxD_i input hold time	90		90		90		ns

External interrupt $\overline{INT_i}$ input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{W(INH)}$	$\overline{INT_i}$ input high-level pulse width	250		250		250		ns
$t_{W(INL)}$	$\overline{INT_i}$ input low-level pulse width	250		250		250		ns

SWITCHING CHARACTERISTICS (V_{CC}=5V±10%, V_{SS}=0V, T_a=25℃, unless otherwise noted)

Single-chip mode

Symbol	Parameter	Test conditions	Limits						Unit
			8MHz		16MHz		25MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(E-P0Q)}$	Port P0 data output delay time	Fig. 58		200		100		80	ns
$t_{d(E-P1Q)}$	Port P1 data output delay time			200		100		80	ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			200		100		80	ns
$t_{d(E-P3Q)}$	Port P3 data output delay time			200		100		80	ns
$t_{d(E-P4Q)}$	Port P4 data output delay time			200		100		80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			200		100		80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			200		100		80	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time			200		100		80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			200		100		80	ns

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

Symbol	Parameter	Test conditions	Limits						Unit
			8MHz		16MHz		25MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(P0A-E)}$	Port P0 address output delay time	Fig. 58	100		30		12		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE="L")			110		70		45	ns
$t_{PXZ(E-P1Z)}$	Port P1 floating start delay time (BYTE="L")			5		5		5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time		100		30		12		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time		80		24		5		ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			110		70		45	ns
$t_{PXZ(E-P2Z)}$	Port P2 floating start delay time			5		5		5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time		100		30		12		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time		80		24		5		ns
$t_{d(\phi_1-HLDA)}$	$\overline{HLDA}$ output delay time			100		50		50	ns
$t_{d(ALE-E)}$	ALE output delay time		4		4		4		ns
$t_{W(ALE)}$	ALE pulse width		90		35		22		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time		100		30		20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time		100		30		20		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	30	0	20	0	18	ns
$t_{h(E-P0A)}$	Port P0 address hold time		50		25		18		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE="L")		9		9		9		ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE="L")		50		25		18		ns
$t_{PZX(E-P1Z)}$	Port P1 floating release delay time (BYTE="L")		50		25		18		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE="H")		50		25		18		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time		9		9		9		ns
$t_{h(E-P2Q)}$	Port P2 data hold time		50		25		18		ns
$t_{PZX(E-P2Z)}$	Port P2 floating release delay time		50		25		18		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		18		18		18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		18		18		18		ns
$t_{W(EL)}$	$\overline{E}$ pulse width		220		95		50		ns

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

Symbol	Parameter	Test conditions	Limits						Unit
			8MHz		16MHz		25MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(P0A-E)}$	Port P0 address output delay time	Fig. 58	100		30		12		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE="L")			110		70		45	ns
$t_{PXZ(E-P1Z)}$	Port P1 floating start delay time (BYTE="L")			5		5		5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time		100		30		12		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time		80		24		5		ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			110		70		45	ns
$t_{PXZ(E-P2Z)}$	Port P2 floating start delay time			5		5		5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time		100		30		12		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time		80		24		5		ns
$t_{d(\phi_1-HLDA)}$	$\overline{HLDA}$ output delay time			100		50		50	ns
$t_{d(ALE-E)}$	ALE output delay time		4		4		4		ns
$t_{W(ALE)}$	ALE pulse width		90		35		22		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time		100		30		20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time		100		30		20		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	30	0	20	0	18	ns
$t_{h(E-P0A)}$	Port P0 address hold time		50		25		18		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE="L")		9		9		9		ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE="L")		50		25		18		ns
$t_{PZX(E-P1Z)}$	Port P1 floating release delay time (BYTE="L")		50		25		18		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE="H")		50		25		18		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time		9		9		9		ns
$t_{h(E-P2Q)}$	Port P2 data hold time		50		25		18		ns
$t_{PZX(E-P2Z)}$	Port P2 floating release delay time		50		25		18		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		18		18		18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		18		18		18		ns
$t_{W(EL)}$	$\overline{E}$ pulse width		470		220		130		ns

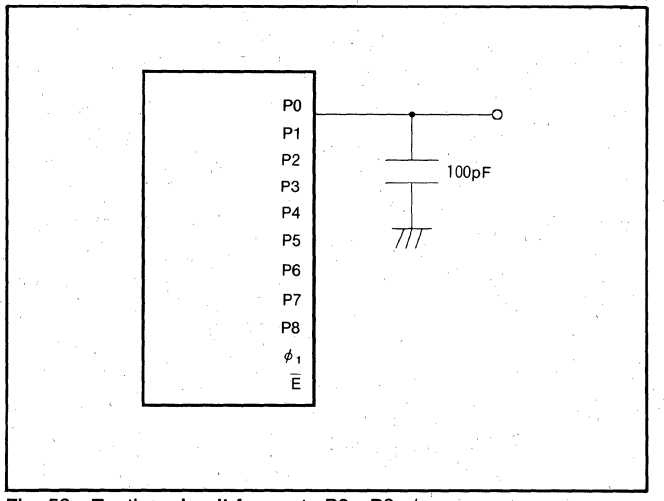
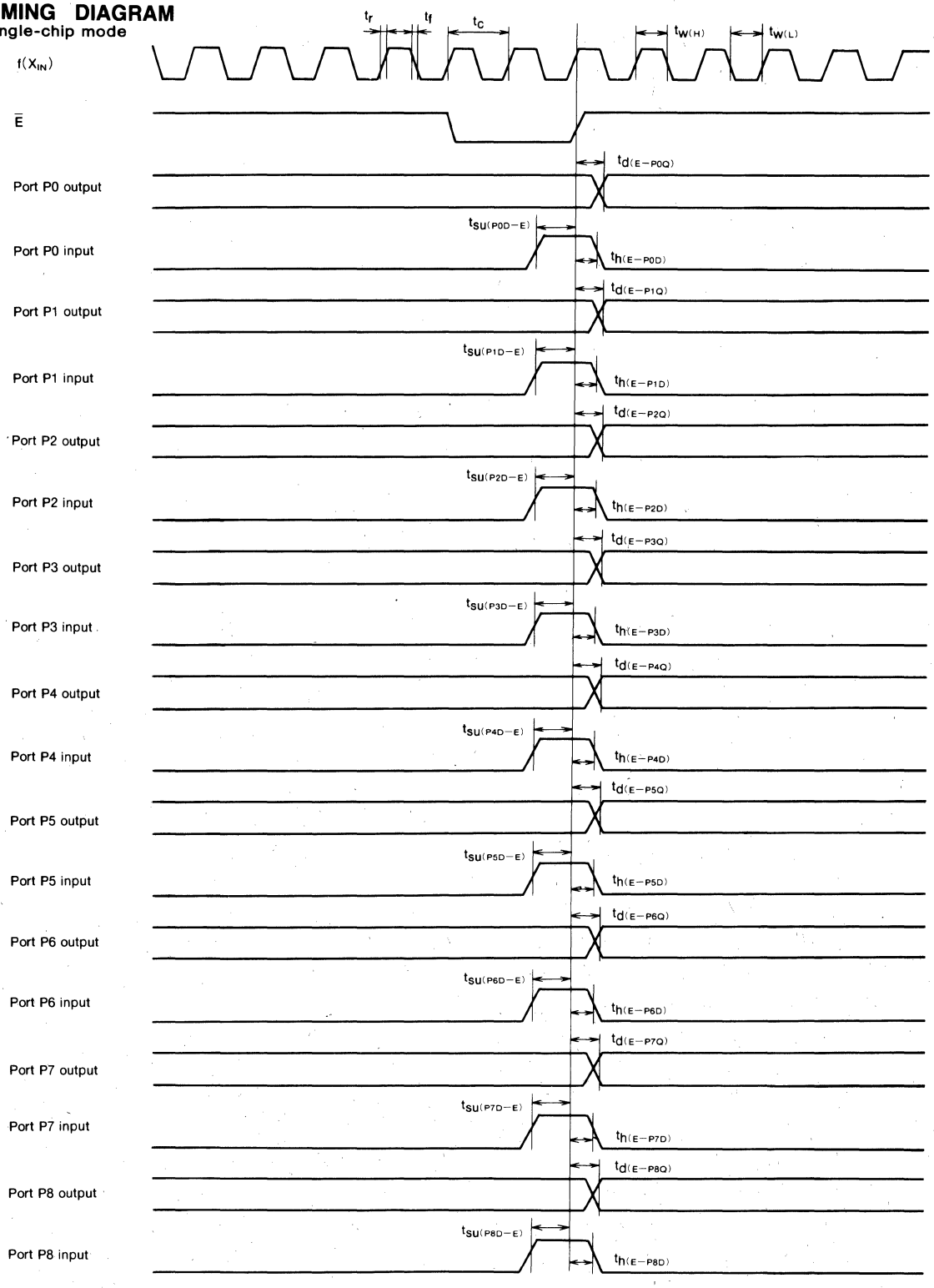
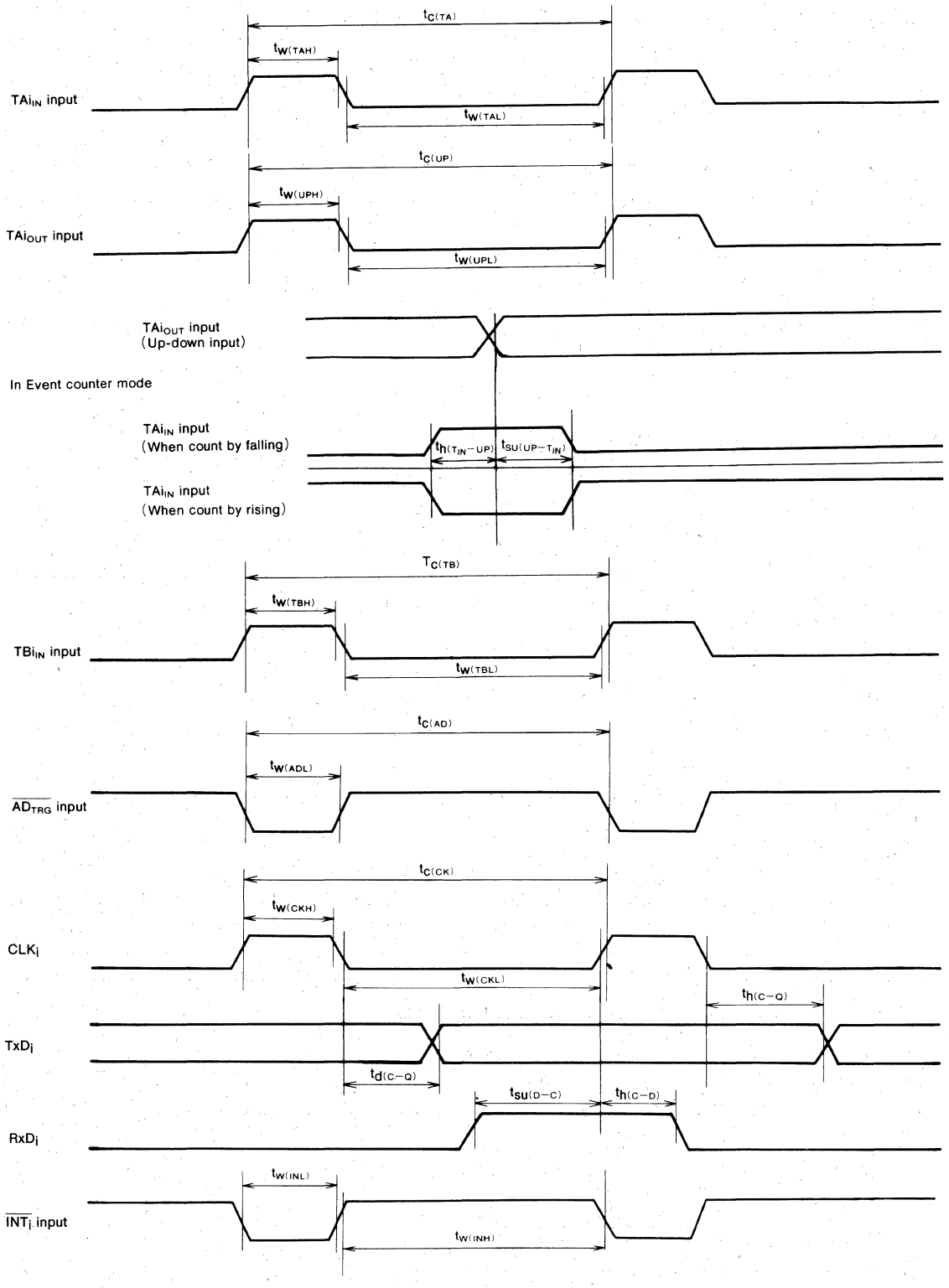


Fig. 58 Testing circuit for ports P0~P8, ϕ_1

TIMING DIAGRAM

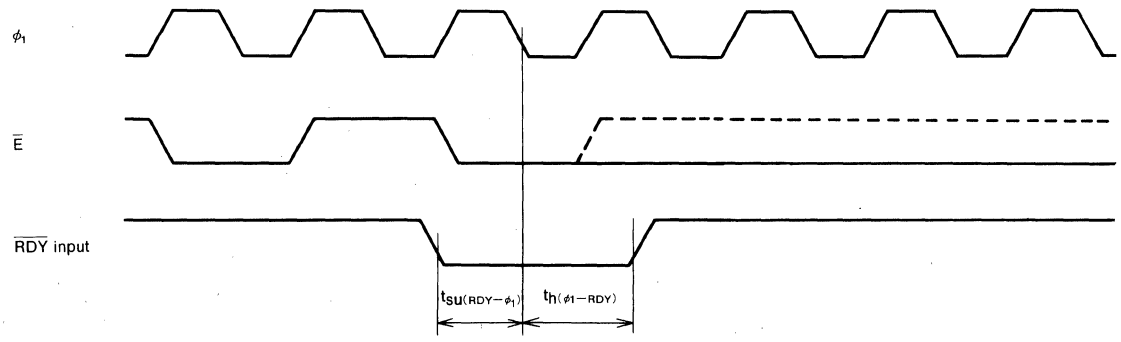
Single-chip mode



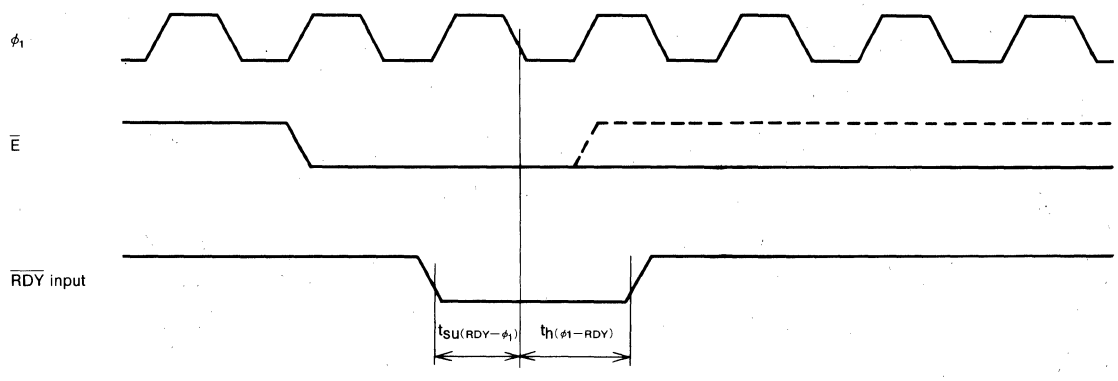

$t_{C(TA)}$
$t_{W(TAH)}$
TAi_{IN} input
$t_{W(TAL)}$
$t_{C(UP)}$
$t_{W(UPH)}$
TAi_{OUT} input
$t_{W(UPL)}$
TAi_{OUT} input
(Up-down input)
In Event counter mode
TAi_{IN} input
(When count by falling)
$t_{h(T_{IN}-UP)}$
$t_{SU(UP-T_{IN})}$
TAi_{IN} input
(When count by rising)
$T_{C(TB)}$
$t_{W(TBH)}$
TBi_{IN} input
$t_{W(TBL)}$
$t_{C(AD)}$
$t_{W(ADL)}$
$\overline{AD_{TRG}}$ input
$t_{C(CK)}$
$t_{W(CKH)}$
CLK_i
$t_{W(CKL)}$
$t_{h(C-Q)}$
TxD_i
$t_{d(C-Q)}$
$t_{su(D-C)}$
$t_{h(C-D)}$
RxD_i
$t_{W(INL)}$
$\overline{INT_i}$ input
$t_{W(INH)}$

Memory expansion mode and microprocessor mode

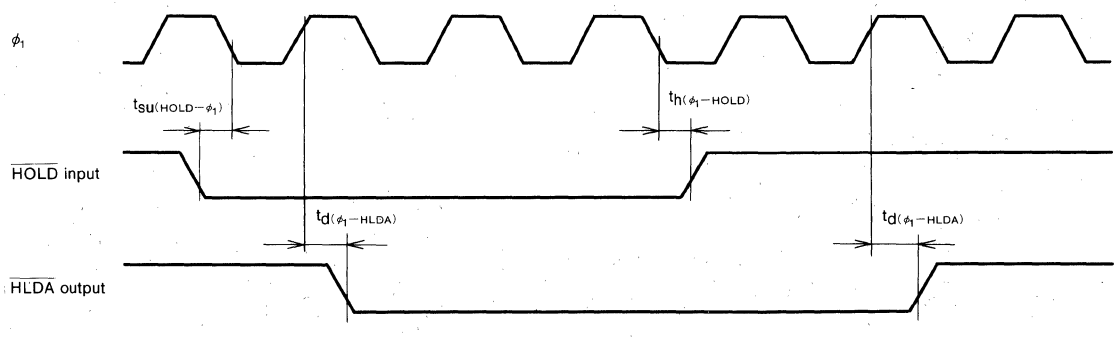
(When wait bit = "1")



(When wait bit = "0")



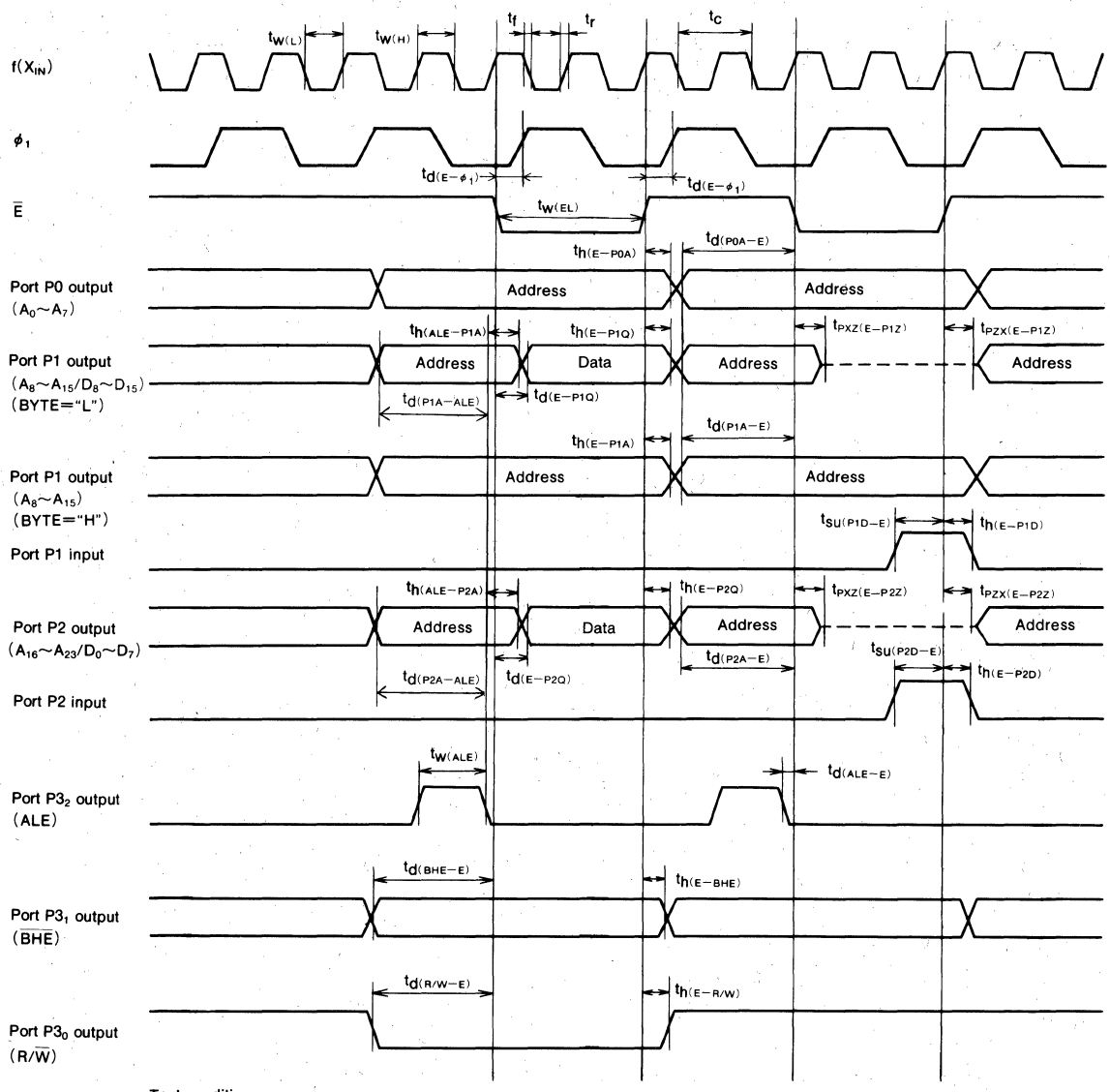
(When wait bit = "1" or "0" in common)



Test conditions

- V_{CC} = 5V±10%
- Input timing voltage : V_{IL} = 1.0V, V_{IH} = 4.0V
- Output timing voltage : V_{OL} = 0.8V, V_{OH} = 2.0V

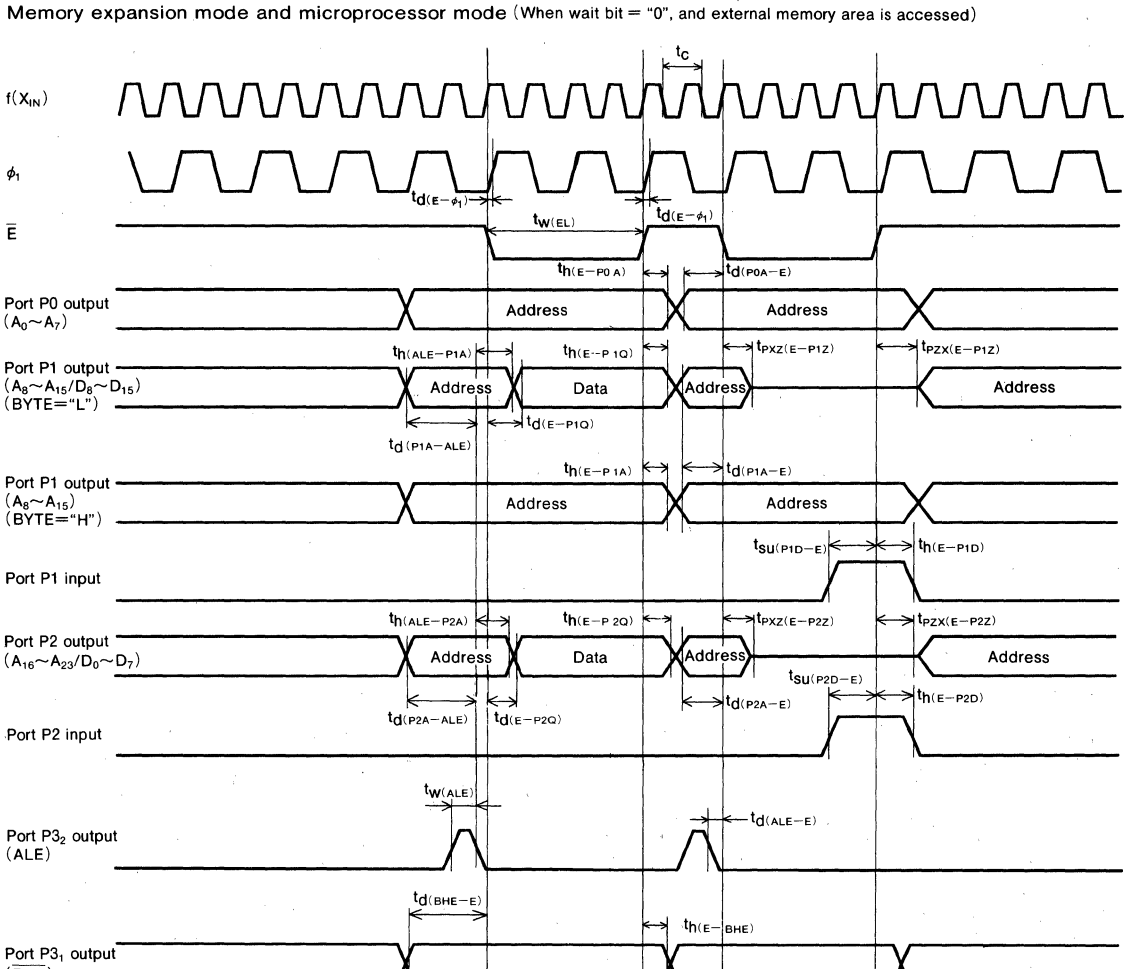
Memory expansion mode and microprocessor mode (When wait bit="1")



Test conditions

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

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



Test conditions

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

M37702M4-XXXFP,M37702M4AXXXFP,M37702M4BXXXFP
M37702S4FP,M37702S4AFP,M37702S4BFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

DESCRIPTION

The M37702M4-XXXFP is a single-chip microcomputer designed with high-performance CMOS silicon gate technology. This is housed in a 80-pin plastic molded QFP. This single-chip microcomputer has a large 16M bytes address space, three instruction queue buffers, and two data buffers for high-speed instruction execution. The CPU is a 16-bit parallel processor that can also be switched to perform 8-bit parallel processing. This microcomputer is suitable for office, business and industrial equipment controller that require high-speed processing of large data.

The differences between M37702M4-XXXFP, M37702M4AXXXFP, M37702M4BXXXFP, M37702S4FP, M37702S4AFP and M37702S4BFP are the ROM size and the external clock input frequency as shown below. Therefore, the following descriptions will be for the M37702M4-XXXFP unless otherwise noted.

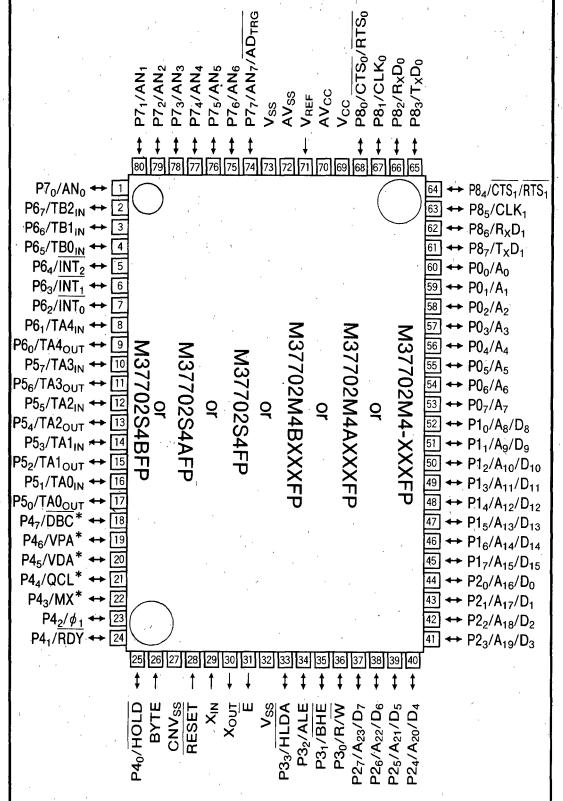
Type name	ROM size	External clock input frequency
M37702M4-XXXFP	32K bytes	8 MHz
M37702M4AXXXFP	32K bytes	16MHz
M37702M4BXXXFP	32K bytes	25MHz
M37702S4FP	External	8 MHz
M37702S4AFP	External	16MHz
M37702S4BFP	External	25MHz

The M37702M4-XXXFP has the same functions as the M37702M2-XXXFP except for the memory size.

FEATURES

- Number of basic instructions 103
- Memory size ROM ..... 32K bytes
  RAM ..... 2048 bytes
- Instruction execution time
  M37702M4-XXXFP, M37702S4FP
  (The fastest instruction at 8 MHz frequency) 500ns
  M37702M4AXXXFP, M37702S4AFP
  (The fastest instruction at 16 MHz frequency) 250ns
  M37702M4BXXXFP, M37702S4BFP
  (The fastest instruction at 25 MHz frequency) 160ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 8 MHz frequency)
  ..... 30mW (Typ.)
- Interrupts ..... 19 types 7 levels
- Multiple function 16-bit timer ..... 5+3
- UART (may also be synchronous) 2
- 8-bit A-D converter ..... 8-channel inputs
- 12-bit watchdog timer
- Programmable input/output
  (ports P0, P1, P2, P3, P4, P5, P6, P7, P8) 68

PIN CONFIGURATION (TOP VIEW)



Outline 80P6N

*: Used in the evaluation chip mode only

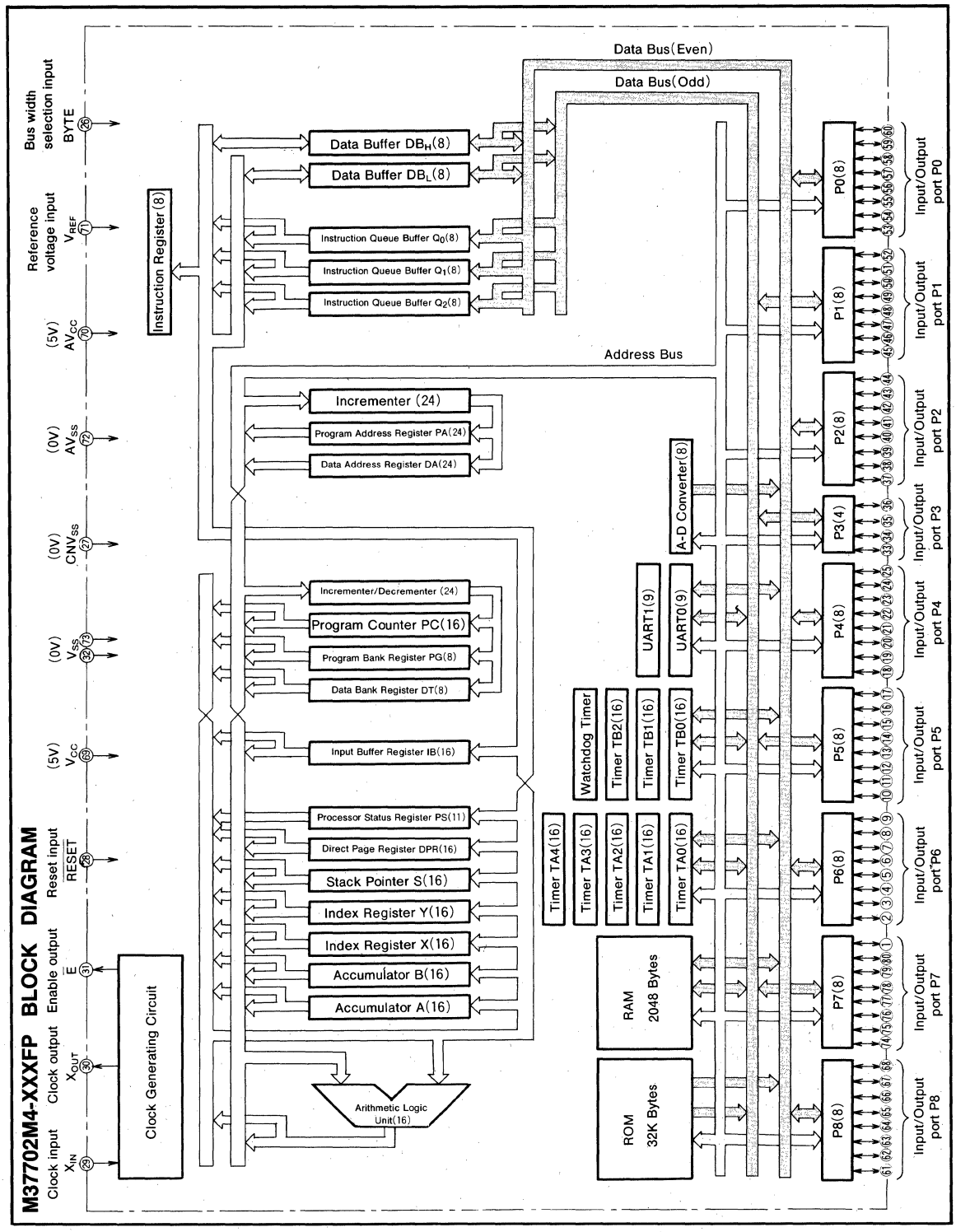
APPLICATION

Control devices for office equipment such as copiers, printers, typewriters, facsimiles, word processors and personal computers

Control devices for industrial equipment such as ME, NC, communication, and measuring instruments

M37702M4-XXXFP,M37702M4AXXXFP,M37702M4BXXXFP
M37702S4FP,M37702S4AFP,M37702S4BFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER



THE FUNCTIONS AND CHARACTERISTICS

The M37702M4-XXXFP has the same functions and characteristics as the M37702M2-XXXFP except for the ROM and RAM size. Refer to the section on the M37702M2-XXXFP. The memory map is shown in Figure 1.

ADDRESSING MODES

The M37702M4-XXXFP has 28 powerful addressing modes. Refer to the MELPS 7700 addressing mode description for the details of each addressing mode.

MACHINE INSTRUCTION LIST

The M37702M4-XXXFP has 103 machine instructions. Refer to the MELPS 7700 machine instruction list for details.

DATA REQUIRED FOR MASK ORDERING

Please send the following data for mask orders.

(1) M37702M4-XXXFP mask ROM order confirmation form
(2) 80P6N mark specification form
(3) ROM data (EPROM 3 sets)

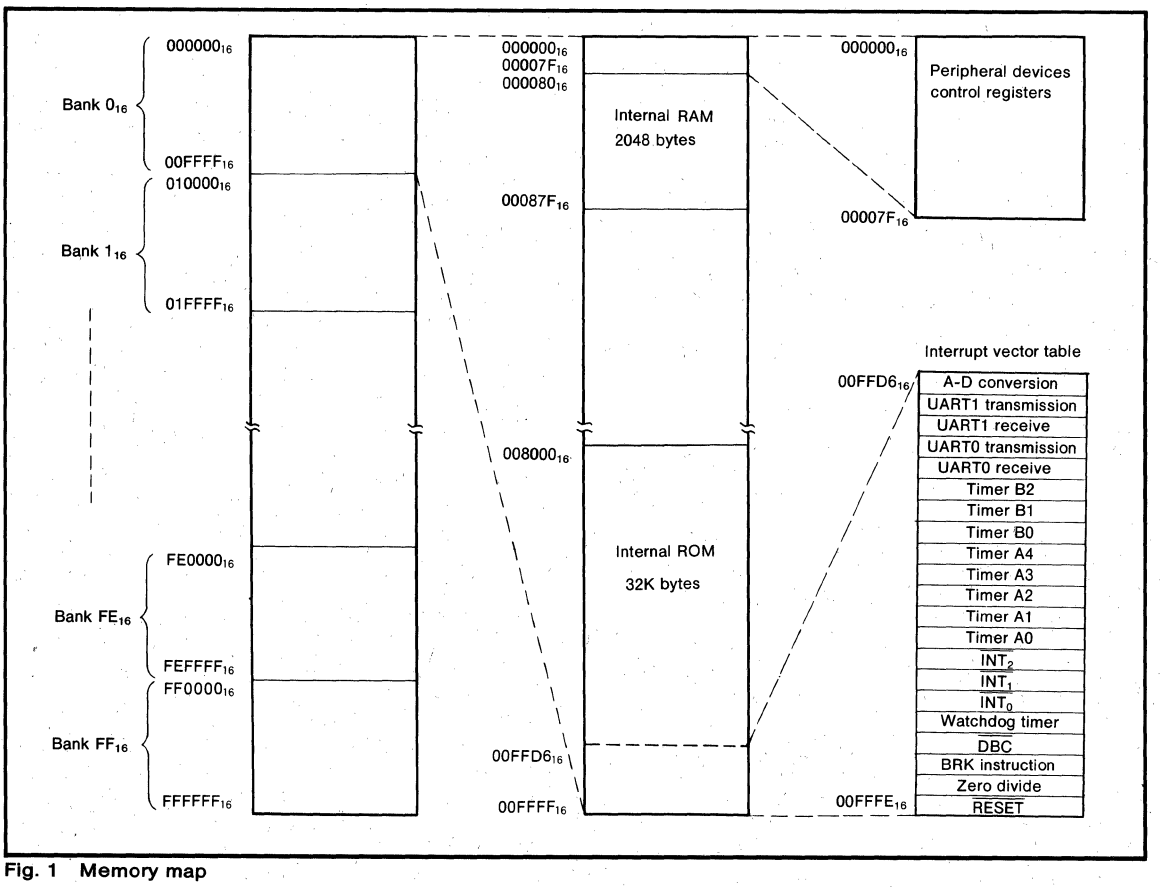


Fig. 1 Memory map

M37703M2-XXXSP,M37703M2AXXXSP,M37703M2BXXXSP M37703S1SP,M37703S1ASP,M37703S1BSP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

DESCRIPTION

The M37703M2-XXXSP is single-chip microcomputer designed with high-performance CMOS silicon gate technology. This is housed in a 64-pin shrink plastic molded DIP. This single-chip microcomputer has a large 16M bytes address space, three instruction queue buffers, and two data buffers for high-speed instruction execution. The CPU is a 16-bit parallel processor that can also be switched to perform 8-bit parallel processing. This microcomputer is suitable for office, business and industrial equipment controller that require high-speed processing of large data.

The differences between M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP, M37703S1SP, M37703S1ASP and M37703S1BSP are the ROM size and the external clock input frequency as shown below. Therefore, the following descriptions will be for the M37703M2-XXXSP unless otherwise noted.

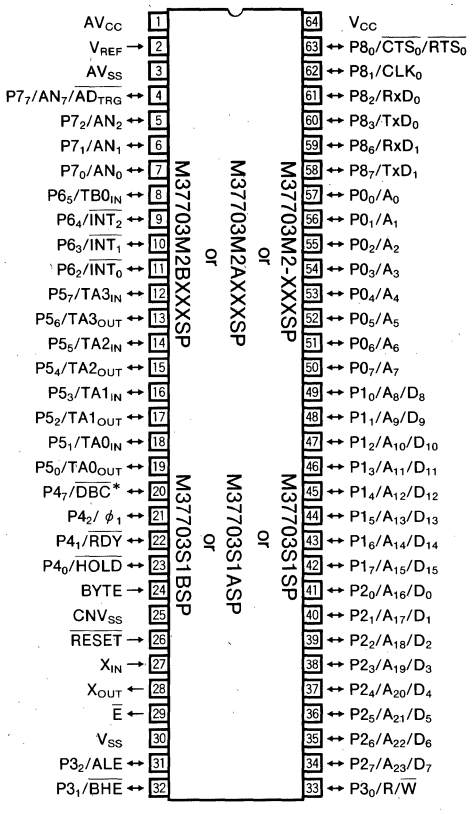
Type name	ROM size	External clock input frequency
M37703M2-XXXSP	16K bytes	8 MHz
M37703M2AXXXSP	16K bytes	16MHz
M37703M2BXXXSP	16K bytes	25MHz
M37703S1SP	External	8 MHz
M37703S1ASP	External	16MHz
M37703S1BSP	External	25MHz

The M37703M2-XXXSP cuts down the pins of M37702M2-XXXFP. Refer to the section on M37702M2-XXXFP for the functional differences.

FEATURES

- Number of basic instructions 103
- Memory size ROM ..... 16K bytes
  RAM ..... 512 bytes
- Instruction execution time
  M37703M2-XXXSP, M37703S1SP
  (The fastest instruction at 8 MHz frequency) 500ns
  M37703M2AXXXSP, M37703S1ASP
  (The fastest instruction at 16 MHz frequency) 250ns
  M37703M2BXXXSP, M37703S1BSP
  (The fastest instruction at 25 MHz frequency) 160ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 8 MHz frequency)
  ..... 30mW (Typ.)
- Interrupts ..... 19 types 7 levels
- Multiple function 16-bit timer ..... 5+3
- UART (may also be synchronous) 2
- 8-bit A-D converter ..... 4-channel inputs
- 12-bit watchdog timer
- Programmable input/output
  (ports P0, P1, P2, P3, P4, P5, P6, P7, P8) 53

PIN CONFIGURATION (TOP VIEW)



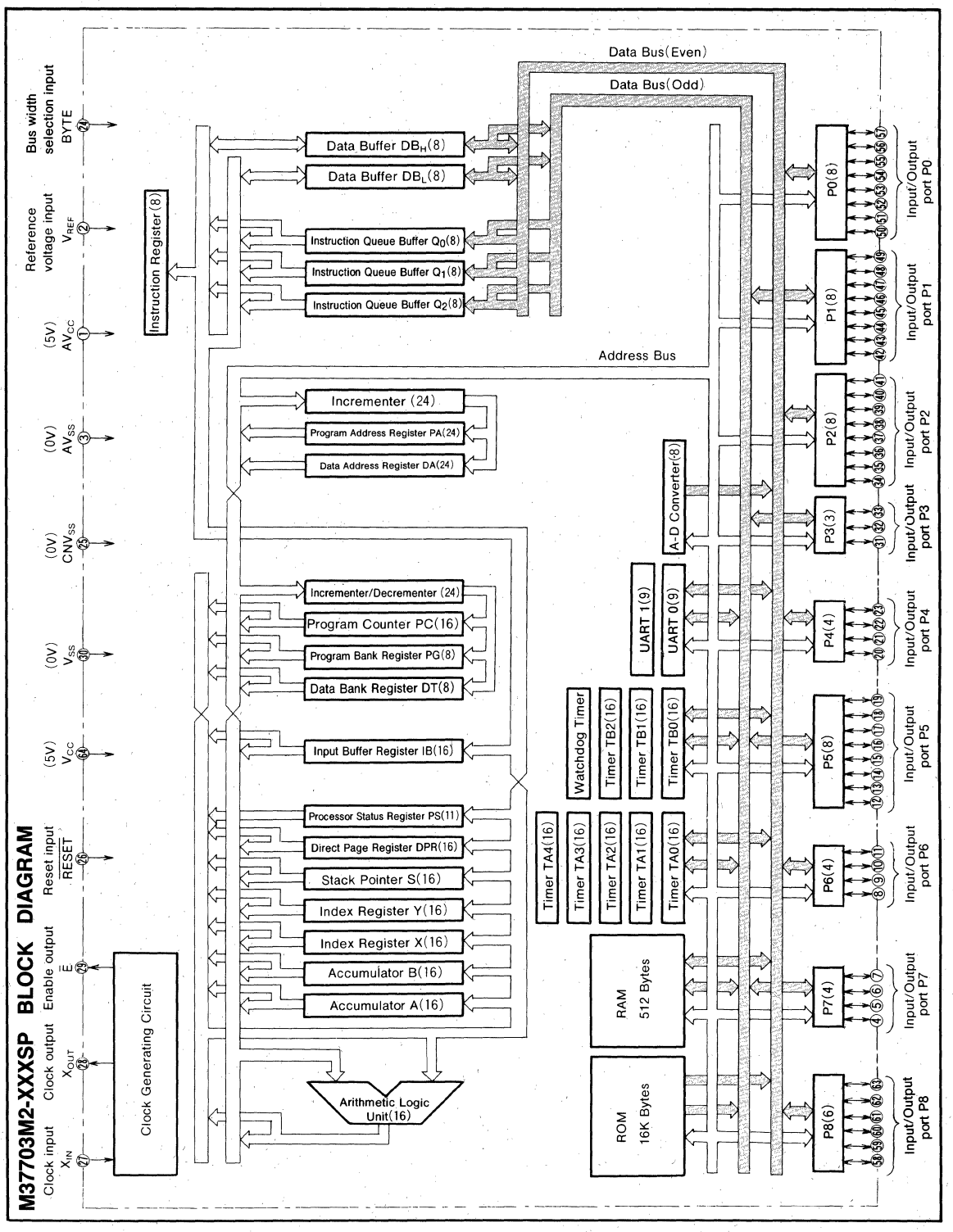
Outline 64P4B

*: Used in the evaluation chip mode only

APPLICATION

Control devices for office equipment such as copiers, printers, typewriters, facsimiles, word processors, and personal computers

Control devices for industrial equipment such as ME, NC, communication, and measuring instruments


M37703M2-XXXSP BLOCK DIAGRAM
Clock input
Clock output
Enable output
Reset input
Reference voltage input
Bus width selection input
Data Bus(Even)
Data Bus(Odd)
Address Bus
Data Buffer DBH(8)
Data Buffer DBL(8)
Instruction Register(8)
Instruction Queue Buffer Q0(8)
Instruction Queue Buffer Q1(8)
Instruction Queue Buffer Q2(8)
Incrementer (24)
Program Address Register PA(24)
Data Address Register DA(24)
Incrementer/Decrementer (24)
Program Counter PC(16)
Program Bank Register PG(8)
Data Bank Register DT(8)
Input Buffer Register IB(16)
Processor Status Register PS(11)
Direct Page Register DPR(16)
Stack Pointer S(16)
Index Register Y(16)
Index Register X(16)
Accumulator B(16)
Accumulator A(16)
Arithmetic Logic Unit(16)
Clock Generating Circuit
A-D Converter(8)
UART 1(9)
UART 0(9)
Watchdog Timer
Timer TB2(16)
Timer TB1(16)
Timer TB0(16)
Timer TA4(16)
Timer TA3(16)
Timer TA2(16)
Timer TA1(16)
Timer TA0(16)
RAM 512 Bytes
ROM 16K Bytes
P0(8)
P1(8)
P2(8)
P3(3)
P4(4)
P5(8)
P6(4)
P7(4)
P8(6)
Input/Output port P0
Input/Output port P1
Input/Output port P2
Input/Output port P3
Input/Output port P4
Input/Output port P5
Input/Output port P6
Input/Output port P7
Input/Output port P8

FUNCTIONS OF M37703M2-XXXSP

Parameter		Functions
Number of basic instructions		103
Instruction execution time	M37703M2-XXXSP, M37703S1SP	500ns (the fastest instructions, at 8MHz frequency)
	M37703M2AXXXSP, M37703S1ASP	250ns (the fastest instructions, at 16MHz frequency)
	M37703M2BXXXSP, M37703S1BSP	160ns (the fastest instructions, at 25MHz frequency)
Memory size	ROM	16K bytes
	RAM	512 bytes
Input/Output ports	P0, P1, P2, P5	8-bit× 4
	P8	6-bit× 1
	P4, P6, P7	4-bit× 3
	P3	3-bit× 1
Multi-function timers	TA0, TA1, TA2, TA3, TA4	16-bit× 5 (4 Input/Output functions)
	TB0, TB1, TB2	16-bit× 3 (1 Input function)
Serial I/O		UART×2(One can be set clock synchronous serial I/O.)
A-D converter		8-bit× 1 (4 channels)
Watchdog timer		12-bit× 1
Interrupts		3 external types, 16 internal types (Each interrupt can be set the priority levels to 0 ~ 7.)
Clock generating circuit		Built-in(externally connected to a ceramic resonator or quartz crystal resonator)
Supply voltage		5 V±10%
Power dissipation		30mW(at external 8 MHz frequency)
Input/Output characteristic	Input/Output voltage	5 V
	Output current	5 mA
Memory expansion		Maximum 16M bytes
Operating temperature range		−20~85℃
Device structure		CMOS high-performance silicon gate process
Package		64-pin shrink plastic molded DIP

PIN DESCRIPTION

Pin	Name	Input/Output	Functions
V_{CC}, V_{SS}	Power supply		Supply 5 V±10% to V_{CC} and 0 V to V_{SS}.
CNV_{SS}	CNV_{SS} input	Input	This pin controls the processor mode. Connect to V_{SS} for single-chip mode, and to V_{CC} for external ROM types.
$\overline{RESET}$	Reset input	Input	To enter the reset state, this pin must be kept at a "L" condition should be maintained for the required time.
X_{IN}	Clock input	Input	These are I/O pins of internal clock generating circuit. Connect a ceramic or quartz crystal resonator between X_{IN} and X_{OUT}. When an external clock is used, the clock source should be connected to the X_{IN} pin and the X_{OUT} pin should be left open.
X_{OUT}	Clock output	Output	
$\overline{E}$	Enable output	Output	Data or instruction read and data write are performed when output from this pin is "L".
BYTE	Bus width selection input	Input	In memory expansion mode or microprocessor mode, this pin determines whether the external data bus is 8-bit width or 16-bit width. The width is 16 bits when "L" signal inputs and 8 bits when "H" signal inputs.
AV_{CC}, AV_{SS}	Analog supply input		Power supply for the A-D converter. Connect AV_{CC} to V_{CC} and AV_{SS} to V_{SS} externally.
V_{REF}	Reference voltage input	Input	This is reference voltage input pin for the A-D converter.
$P0_0$~$P0_7$	I/O port P0	I/O	In single-chip mode, port P0 becomes an 8-bit I/O port. An I/O directional register is available so that each pin can be programmed for input or output. These ports are in input mode when reset . Address(A_7~A_0)is output in memory expansion mode or microprocessor mode.
$P1_0$~$P1_7$	I/O port P1	I/O	In single-chip mode, these pins have the same functions as port P0. When the BYTE pin is set to "L" in memory expansion mode or microprocessor mode and external data bus is 16-bit width, high-order data (D_{15}~D_8)is input or output when $\overline{E}$ output is "L" and an address (A_{15}~A_8) is output when $\overline{E}$ output is "H". If the BYTE pin is "H" that is an external data bus is 8-bit width, only address(A_{15}~A_8)is output.
$P2_0$~$P2_7$	I/O port P2	I/O	In single-chip mode, these pins have the same functions as port P0. In memory expansion mode or microprocessor mode low-order data(D_7~D_0) is input or output when $\overline{E}$ output is "L" and an address(A_{23}~A_{16})is output when $\overline{E}$ output is "H".
$P3_0$~$P3_2$	I/O port P3	I/O	In single-chip mode , these pins have the same functions as port P0. In memory expansion mode or microprocessor mode, R/$\overline{W}$, $\overline{BHE}$, and ALE signals are output.
$P4_0$~$P4_2$, $P4_7$	I/O port P4	I/O	In single-chip mode, these pins have the same functions as port P0. In memory expansion mode or microprocessor mode, $P4_0$ and $P4_1$ become $\overline{HOLD}$ and $\overline{RDY}$ input pin respectively. Functions of other pins are the same as in single-chip mode. In single-chip mode or memory expansion mode, port $P4_2$ can be programed for ϕ_1 output pin divided the clock to X_{IN} pin by 2. In microprocessor mode, $P4_2$ always has the function as ϕ_1 output pin.
$P5_0$~$P5_7$	I/O port P5	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as I/O pins for timer A0, timer A1, timer A2, and timer A3.
$P6_2$~$P6_5$	I/O port P6	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as external interrupt input $\overline{INT_0}$, $\overline{INT_1}$, and $\overline{INT_2}$ pins, and input pins for timer B0.
$P7_0$~$P7_2$, $P7_7$	I/O port P7	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as analog input AN_0~AN_2 and AN_7 input pins. $P7_7$ also has an A-D conversion trigger input function.
$P8_0$~$P8_3$, $P8_6$, $P8_7$	I/O port P8	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as R_XD, T_XD, CLK, $\overline{CTS}$/$\overline{RTS}$ pins for UART 0, and as RxD, TxD pins for UART 1.

BASIC FUNCTION BLOCKS

The functional differences between the M7703M2-XXXSP and M37702M2-XXXFP are described below. The M37703M2-XXXSP has the same functions as the M37702M2-XXXFP, except these points. Refer to the section on the M37702M2-XXXFP.

A-D CONVERTER

Analog signals are input through four channels, AN_0, AN_1, AN_2 and AN_7. In one-shot mode and repeat mode, select one on AN_0, AN_1, AN_2, and AN_7 as analog input by the analog input selection bits (bits 2, 1 and 0) of A-D control register. Set the bits of the directional registers for ports corresponding to analog input channels AN_3, AN_4, AN_5 and AN_6 not having pins to "1" (output mode) and output "0" to the ports.

In the single sweep mode and repeat sweep mode, the M37703M2-XXXSP operates the same as the M37702M2-XXXFP. Set the directional register bits of ports corresponding to AN0, AN1, AN2, and AN7 to "0" (input mode), and the bits of the directional registers for ports corresponding to AN_3, AN_4, AN_5 and AN_6 not having pins to "1" (output mode), and output "0" to the ports. In the single sweep mode and repeat sweep mode, the contents of A-D register bits corresponding to analog input channels AN_3, AN_4, AN_5, and AN_6 not having pins are undefined.

TIMER

Since timer A4 has no input/output function and timer B1, B2 have no input function, timers A4, B1 and B2 operate only in timer mode. Therefore, only clock source can be selected by the bits 7 and 6 of timer mode register for each of timers A4, B1 and B2. The bits of timer mode register must be "0" except for the clock source selection bits. Other timers A0, A1, A2, A3 and B0 have the same functions as the M37702M2-XXXFP.

SERIAL I/O

UART1 has only the asynchronous serial communication function and no clock synchronous serial communication function. Therefore, do not select the clock synchronous serial communication function ("001") by the serial communication method selection bits (bits 2, 1 and 0) of UART1 transmit/receive mode register. Since UART1 does not have the functions of $\overline{CTS}$ and $\overline{RTS}$, the $\overline{CTS}$ and $\overline{RTS}$ selection bit (bit 2) of UART1 transmit/receive control register 0 must always be "1". UART0 has the same function as the M37702M2-XXXFP.

INPUT/OUTPUT PINS

The port registers and directional registers for ports P4, P6, P7 and P8 have eigtht bits, the directional register bits having no pins must always be set to the output mode. Since port $P3_3$ is not available as a pin although it has port register and directional register, port $P3_3$ must be set to the output mode.

ADDRESSING MODES

The M37703M2-XXXSP has 28 powerful addressing modes. Refer to the MELPS 7700 addressing mode description for the details of each addressing mode.

MACHINE INSTRUCTION LIST

The M37703M2-XXXSP has 103 machine instructions. Refer to the MELPS 7700 machine instruction list for details.

DATA REQUIRED FOR MASK ORDERING

Please send the following data for mask orders.

(1) M37703M2-XXXSP mask ROM order confirmation form
(2) 64P4B mark specification form
(3) ROM data (EPROM 3 sets)

Table 1. The functional differences between the M37703M2-XXXSP and M37702M2-XXXFP

Parameter	M37703M2-XXXSP	M37702M2-XXXFP
Input/Output ports	P0~P2, P5 ····· 8-bit× 4 P8 ····· 6-bit× 1 P4, P6, P7 ····· 4-bit× 3 P3 ····· 3-bit× 1 (without $\overline{HLDA}$)	P0~P2, P4~P8 ····· 8-bit× 8 P3 ····· 4-bit× 1 (with $\overline{HLDA}$)
Timer	Timer A ····· with Input/Output ports 16-bit× 4 only timer mode 16-bit× 1 Timer B ····· with Input ports 16-bit× 1 only timer mode 16-bit× 2	Timer A ····· with Input/Output ports 16-bit× 5 Timer B ····· with Input ports 16-bit× 3
Serial I/O	(UART or clock synchronous serial I/O)× 1 UART× 1	(UART or clock synchronous serial I/O)× 2
A-D converter	8-bit× 1 (4 channels)	8-bit× 1 (8 channels)

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Supply voltage		−0.3∼7	V
AV_{CC}	Analog supply voltage		−0.3∼7	V
V_I	Input voltage $\overline{RESET}$, CNV_{SS}, BYTE		−0.3∼12	V
V_I	Input voltage $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_2$, $P4_0$∼$P4_2$, $P4_7$, $P5_0$∼$P5_7$, $P6_2$∼$P6_5$, $P7_0$∼$P7_2$, $P7_7$, $P8_0$∼$P8_3$, $P8_6$, $P8_7$, V_{REF}, X_{IN}		−0.3∼V_{CC}+0.3	V
V_O	Output voltage $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_2$, $P4_0$∼$P4_2$, $P4_7$, $P5_0$∼$P5_7$, $P6_2$∼$P6_5$, $P7_0$∼$P7_2$, $P7_7$, $P8_0$∼$P8_3$, $P8_6$, $P8_7$, X_{OUT}, $\overline{E}$		−0.3∼V_{CC}+0.3	V
P_d	Power dissipation	T_a=25℃	1000	mW
T_{opr}	Operating temperature		−20∼85	℃
T_{stg}	Storage temperature		−40∼150	℃

RECOMMENDED OPERATING CONDITIONS (V_{CC}=5V±10%, T_a=−20∼85℃, unless otherwise noted)

Symbol	Parameter		Limits Min.	Limits Typ.	Limits Max.	Unit
V_{CC}	Supply voltage		4.5	5.0	5.5	V
AV_{CC}	Analog supply voltage			V_{CC}		V
V_{SS}	Supply voltage			0		V
AV_{SS}	Analog supply voltage			0		V
V_{IH}	High-level input voltage $P0_0$∼$P0_7$, $P3_0$∼$P3_2$, $P4_0$∼$P4_2$, $P4_7$, $P5_0$∼$P5_7$, $P6_2$∼$P6_5$, $P7_0$∼$P7_2$, $P7_7$, $P8_0$∼$P8_3$, $P8_6$, $P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE		0.8V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage $P1_0$∼$P1_7$, $P2_0$∼$P2_7$ (in single-chip mode)		0.8V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage $P1_0$∼$P1_7$, $P2_0$∼$P2_7$ (in memory expansion mode and microprocessor mode)		0.5V_{CC}		V_{CC}	V
V_{IL}	Low-level input voltage $P0_0$∼$P0_7$, $P3_0$∼$P3_2$, $P4_0$∼$P4_2$, $P4_7$, $P5_0$∼$P5_7$, $P6_2$∼$P6_5$, $P7_0$∼$P7_2$, $P7_7$, $P8_0$∼$P8_3$, $P8_6$, $P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE		0		0.2V_{CC}	V
V_{IL}	Low-level input voltage $P1_0$∼$P1_7$, $P2_0$∼$P2_7$ (in single-chip mode)		0		0.2V_{CC}	V
V_{IL}	Low-level input voltage $P1_0$∼$P1_7$, $P2_0$∼$P2_7$ (in memory expansion mode and microprocessor mode)		0		0.16V_{CC}	V
$I_{OH(peak)}$	High-level peak output current $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_2$, $P4_0$∼$P4_2$, $P4_7$, $P5_0$∼$P5_7$, $P6_2$∼$P6_5$, $P7_0$∼$P7_2$, $P7_7$, $P8_0$∼$P8_3$, $P8_6$, $P8_7$				−10	mA
$I_{OH(avg)}$	High-level average output current $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_2$, $P4_0$∼$P4_2$, $P4_7$, $P5_0$∼$P5_7$, $P6_2$∼$P6_5$, $P7_0$∼$P7_2$, $P7_7$, $P8_0$∼$P8_3$, $P8_6$, $P8_7$				−5	mA
$I_{OL(peak)}$	Low-level peak output current $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_2$, $P4_0$∼$P4_2$, $P4_7$, $P5_0$∼$P5_7$, $P6_2$∼$P6_5$, $P7_0$∼$P7_2$, $P7_7$, $P8_0$∼$P8_3$, $P8_6$, $P8_7$				10	mA
$I_{OL(avg)}$	Low-level average output current $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_2$, $P4_0$∼$P4_2$, $P4_7$, $P5_0$∼$P5_7$, $P6_2$∼$P6_5$, $P7_0$∼$P7_2$, $P7_7$, $P8_0$∼$P8_3$, $P8_6$, $P8_7$				5	mA
$f(X_{IN})$	External clock frequency input	M37703M2-XXXSP, M37703S1SP			8	MHz
		M37703M2AXXXSP, M37703S1ASP			16	
		M37703M2BXXXSP, M37703S1ASP			25	

Note 1. Average output current is the average value of a 100ms interval.
2. The sum of $I_{OL(peak)}$ for ports P0, P1, P2, P3 and P8 must be 80mA or less, the sum of $I_{OH(peak)}$ for ports P0, P1, P2, P3 and P8 must be 80mA or less, the sum of $I_{OL(peak)}$ for ports P4, P5, P6 and P7 must be 80mA or less, and the sum of $I_{OH(peak)}$ for ports P4, P5, P6 and P7 must be 80mA or less.

M37703M2-XXXSP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=8MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits Min.	Limits Typ.	Limits Max.	Unit
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$	I_{OH}=−10mA	3			V
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$	I_{OH}=−400μA	4.7			V
V_{OH}	High-level output voltage $P3_2$	I_{OH}=−10mA	3.1			V
		I_{OH}=−400μA	4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA	3.4			V
		I_{OH}=−400μA	4.8			
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$	I_{OL}=10mA			2	V
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$	I_{OL}=2mA			0.45	V
V_{OL}	Low-level output voltage $P3_2$	I_{OL}=10mA			1.9	V
		I_{OL}=2mA			0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA			1.6	V
		I_{OL}=2mA			0.4	
V_{T+}−V_{T-}	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA3_{IN}$, $TB0_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, AD_{TRG}, $\overline{CTS_0}$, CLK_0		0.4		1	V
V_{T+}−V_{T-}	Hysteresis $\overline{RESET}$		0.2		0.5	V
V_{T+}−V_{T-}	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_2$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=5V			5	μA
I_{IL}	Low-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_2$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=0V			−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	In single-chip mode output only pin is open and other pins are V_{SS} during reset. f(X_{IN})=8MHz, square waveform		6	12	mA
		T_a=25℃ when clock is stopped.			1	μA
		T_a=85℃ when clock is stopped.			20	

A-D CONVERTER CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=8MHz, unless otherwise noted)

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

M37703M2AXXXSP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=16MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits Min.	Limits Typ.	Limits Max.	Unit
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$	I_{OH}=−10mA	3			V
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$	I_{OH}=−400μA	4.7			V
V_{OH}	High-level output voltage $P3_2$	I_{OH}=−10mA	3.1			V
		I_{OH}=−400μA	4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA	3.4			V
		I_{OH}=−400μA	4.8			
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$	I_{OL}=10mA			2	V
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$	I_{OL}=2mA			0.45	V
V_{OL}	Low-level output voltage $P3_2$	I_{OL}=10mA			1.9	V
		I_{OL}=2mA			0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA			1.6	V
		I_{OL}=2mA			0.4	
$V_{T+}-V_{T-}$	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA3_{IN}$, $TB0_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, $\overline{AD_{TRG}}$, $\overline{CTS_0}$, CLK_0		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis $\overline{RESET}$		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_2$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=5V			5	μA
I_{IL}	Low-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_2$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=0V			−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	In single-chip mode output only pin is open and other pins are V_{SS} during reset. f(X_{IN})=16MHz, square waveform		12	24	mA
		T_a=25℃ when clock is stopped.			1	μA
		T_a=85℃ when clock is stopped.			20	

A-D CONVERTER CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=16MHz, unless otherwise noted)

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

M37703M2BXXXSP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=25MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits Min.	Limits Typ.	Limits Max.	Unit
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$	I_{OH}=−10mA	3			V
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$	I_{OH}=−400μA	4.7			V
V_{OH}	High-level output voltage $P3_2$	I_{OH}=−10mA	3.1			V
		I_{OH}=−400μA	4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA	3.4			V
		I_{OH}=−400μA	4.8			
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$	I_{OL}=10mA			2	V
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$	I_{OL}=2mA			0.45	V
V_{OL}	Low-level output voltage $P3_2$	I_{OL}=10mA			1.9	V
		I_{OL}=2mA			0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA			1.6	V
		I_{OL}=2mA			0.4	
V_{T+}−V_{T-}	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA3_{IN}$, $TB0_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, AD_{TRG}, $\overline{CTS_0}$, CLK_0		0.4		1	V
V_{T+}−V_{T-}	Hysteresis $\overline{RESET}$		0.2		0.5	V
V_{T+}−V_{T-}	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_2$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=5V			5	μA
I_{IL}	Low-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_2$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=0V			−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	In single-chip mode output only pin is open and other pins are V_{SS} during reset. f(X_{IN})=25MHz, square waveform		19	38	mA
		T_a=25℃ when clock is stopped.			1	μA
		T_a=85℃ when clock is stopped.			20	

A-D CONVERTER CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=25MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits Min.	Limits Typ.	Limits Max.	Unit
—	Resolution	V_{REF}=V_{CC}			8	Bits
—	Absolute accuracy	V_{REF}=V_{CC}			±3	LSB
R_{LADDER}	Ladder resistance	V_{REF}=V_{CC}	2		10	kΩ
t_{CONV}	Conversion time		9.12			μs
V_{REF}	Reference voltage		2		V_{CC}	V
V_{IA}	Analog input voltage		0		V_{REF}	V

TIMING REQUIREMENTS (V_{CC}=5V±10%, V_{SS}=0V, T_a=25℃, unless otherwise noted)

External clock input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
t_C	External clock input cycle time	125		62		40		ns
$t_{W(H)}$	External clock input high-level pulse width	50		25		15		ns
$t_{W(L)}$	External clock input low-level pulse width	50		25		15		ns
t_r	External clock rise time		20		10		8	ns
t_f	External clock fall time		20		10		8	ns

Single-chip mode

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{SU(P0D-E)}$	Port P0 input setup time	200		100		60		ns
$t_{SU(P1D-E)}$	Port P1 input setup time	200		100		60		ns
$t_{SU(P2D-E)}$	Port P2 input setup time	200		100		60		ns
$t_{SU(P3D-E)}$	Port P3 input setup time	200		100		60		ns
$t_{SU(P4D-E)}$	Port P4 input setup time	200		100		60		ns
$t_{SU(P5D-E)}$	Port P5 input setup time	200		100		60		ns
$t_{SU(P6D-E)}$	Port P6 input setup time	200		100		60		ns
$t_{SU(P7D-E)}$	Port P7 input setup time	200		100		60		ns
$t_{SU(P8D-E)}$	Port P8 input setup time	200		100		60		ns
$t_{h(E-P0D)}$	Port P0 input hold time	0		0		0		ns
$t_{h(E-P1D)}$	Port P1 input hold time	0		0		0		ns
$t_{h(E-P2D)}$	Port P2 input hold time	0		0		0		ns
$t_{h(E-P3D)}$	Port P3 input hold time	0		0		0		ns
$t_{h(E-P4D)}$	Port P4 input hold time	0		0		0		ns
$t_{h(E-P5D)}$	Port P5 input hold time	0		0		0		ns
$t_{h(E-P6D)}$	Port P6 input hold time	0		0		0		ns
$t_{h(E-P7D)}$	Port P7 input hold time	0		0		0		ns
$t_{h(E-P8D)}$	Port P8 input hold time	0		0		0		ns

Memory expansion mode and microprocessor mode

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{SU(P1D-E)}$	Port P1 input setup time	60		45		30		ns
$t_{SU(P2D-E)}$	Port P2 input setup time	60		45		30		ns
$t_{SU(RDY-\phi_1)}$	$\overline{RDY}$ input setup time	70		60		55		ns
$t_{SU(HOLD-\phi_1)}$	$\overline{HOLD}$ input setup time	70		60		55		ns
$t_{h(E-P1D)}$	Port P1 input hold time	0		0		0		ns
$t_{h(E-P2D)}$	Port P2 input hold time	0		0		0		ns
$t_{h(\phi_1-RDY)}$	$\overline{RDY}$ input hold time	0		0		0		ns
$t_{h(\phi_1-HOLD)}$	$\overline{HOLD}$ input hold time	0		0		0		ns

Timer A input (Count input in event counter mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	250		125		80		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	125		62		40		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	125		62		40		ns

Timer A input (Gating input in timer mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	1000		500		320		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	500		250		160		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	500		250		160		ns

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

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	500		250		160		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	250		125		80		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	250		125		80		ns

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

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	250		125		80		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	250		125		80		ns

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

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(UP)}$	TAi_{OUT} input cycle time	5000		2500		2000		ns
$t_{W(UPH)}$	TAi_{OUT} input high-level pulse width	2500		1250		1000		ns
$t_{W(UPL)}$	TAi_{OUT} input low-level pulse width	2500		1250		1000		ns
$t_{su(UP-T_{IN})}$	TAi_{OUT} input setup time	1000		500		400		ns
$t_{h(T_{IN}-UP)}$	TAi_{OUT} input hold time	1000		500		400		ns

Timer B input (Count input in event counter mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	$TB0_{IN}$ input cycle time (one edge count)	250		125		80		ns
$t_{W(TBH)}$	$TB0_{IN}$ input high-level pulse width (one edge count)	125		62		40		ns
$t_{W(TBL)}$	$TB0_{IN}$ input low-level pulse width (one edge count)	125		62		40		ns
$t_{C(TB)}$	$TB0_{IN}$ input cycle time (both edges count)	500		250		160		ns
$t_{W(TBH)}$	$TB0_{IN}$ input high-level pulse width (both edges count)	250		125		80		ns
$t_{W(TBL)}$	$TB0_{IN}$ input low-level pulse width (both edges count)	250		125		80		ns

Timer B input (Pulse period measurement mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	$TB0_{IN}$ input cycle time	1000		500		320		ns
$t_{W(TBH)}$	$TB0_{IN}$ input high-level pulse width	500		250		160		ns
$t_{W(TBL)}$	$TB0_{IN}$ input low-level pulse width	500		250		160		ns

Timer B input (Pulse width measurement mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	$TB0_{IN}$ input cycle time	1000		500		320		ns
$t_{W(TBH)}$	$TB0_{IN}$ input high-level pulse width	500		250		160		ns
$t_{W(TBL)}$	$TB0_{IN}$ input low-level pulse width	500		250		160		ns

A-D trigger input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(AD)}$	$\overline{AD_{TRG}}$ input cycle time (minimum allowable trigger)	2000		1000		1000		ns
$t_{W(ADL)}$	$\overline{AD_{TRG}}$ input low-level pulse width	250		125		125		ns

Serial I/O

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(CK)}$	CLK_0 input cycle time	500		250		200		ns
$t_{W(CKH)}$	CLK_0 input high-level pulse width	250		125		100		ns
$t_{W(CKL)}$	CLK_0 input low-level pulse width	250		125		100		ns
$t_{d(C-Q)}$	TxD_0 output delay time		150		90		80	ns
$t_{h(C-Q)}$	TxD_0 hold time	30		30		30		ns
$t_{su(D-C)}$	RxD_0 input setup time	60		30		20		ns
$t_{h(C-D)}$	RxD_0 input hold time	90		90		90		ns

External interrupt $\overline{INT_i}$ input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{W(INH)}$	$\overline{INT_i}$ input high-level pulse width	250		250		250		ns
$t_{W(INL)}$	$\overline{INT_i}$ input low-level pulse width	250		250		250		ns

SWITCHING CHARACTERISTICS (V_{CC}=5V±10%, V_{SS}=0V, T_a=25℃, unless otherwise noted)

Single-chip mode

Symbol	Parameter	Test conditions	Limits						Unit
			8MHz		16MHz		25MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(E-P0Q)}$	Port P0 data output delay time	Fig. 1		200		100		80	ns
$t_{d(E-P1Q)}$	Port P1 data output delay time			200		100		80	ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			200		100		80	ns
$t_{d(E-P3Q)}$	Port P3 data output delay time			200		100		80	ns
$t_{d(E-P4Q)}$	Port P4 data output delay time			200		100		80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			200		100		80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			200		100		80	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time			200		100		80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			200		100		80	ns

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

Symbol	Parameter	Test conditions	Limits						Unit
			8MHz		16MHz		25MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(P0A-E)}$	Port P0 address output delay time	Fig. 1	100		30		12		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE="L")			110		70		45	ns
$t_{PXZ(E-P1Z)}$	Port P1 floating start delay time (BYTE="L")			5		5		5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time		100		30		12		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time		80		24		5		ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			110		70		45	ns
$t_{PXZ(E-P2Z)}$	Port P2 floating start delay time			5		5		5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time		100		30		12		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time		80		24		5		ns
$t_{d(ALE-E)}$	ALE output delay time		4		4		4		ns
$t_{W(ALE)}$	ALE pulse width		90		35		22		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time		100		30		20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time		100		30		20		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	30	0	20	0	18	ns
$t_{h(E-P0A)}$	Port P0 address hold time		50		25		18		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE="L")		9		9		9		ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE="L")		50		25		18		ns
$t_{PZX(E-P1Z)}$	Port P1 floating release delay time (BYTE="L")		50		25		18		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE="H")		50		25		18		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time		9		9		9		ns
$t_{h(E-P2Q)}$	Port P2 data hold time		50		25		18		ns
$t_{PZX(E-P2Z)}$	Port P2 floating release delay time		50		25		18		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		18		18		18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		18		18		18		ns
$t_{W(EL)}$	$\overline{E}$ pulse width		220		95		50		ns

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

Symbol	Parameter	Test conditions	Limits						Unit
			8MHz		16MHz		25MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(P0A-E)}$	Port P0 address output delay time	Fig. 1	100		30		12		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE="L")			110		70		45	ns
$t_{PXZ(E-P1Z)}$	Port P1 floating start delay time (BYTE="L")			5		5		5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time		100		30		12		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time		80		24		5		ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			110		70		45	ns
$t_{PXZ(E-P2Z)}$	Port P2 floating start delay time			5		5		5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time		100		30		12		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time		80		24		5		ns
$t_{d(ALE-E)}$	ALE output delay time		4		4		4		ns
$t_{W(ALE)}$	ALE pulse width		90		35		22		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time		100		30		20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time		100		30		20		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	30	0	20	0	18	ns
$t_{h(E-P0A)}$	Port P0 address hold time		50		25		18		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE="L")		9		9		9		ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE="L")		50		25		18		ns
$t_{PZX(E-P1Z)}$	Port P1 floating release delay time (BYTE="L")		50		25		18		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE="H")		50		25		18		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time		9		9		9		ns
$t_{h(E-P2Q)}$	Port P2 data hold time		50		25		18		ns
$t_{PZX(E-P2Z)}$	Port P2 floating release delay time		50		25		18		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		18		18		18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		18		18		18		ns
$t_{W(EL)}$	$\overline{E}$ pulse width		470		220		130		ns

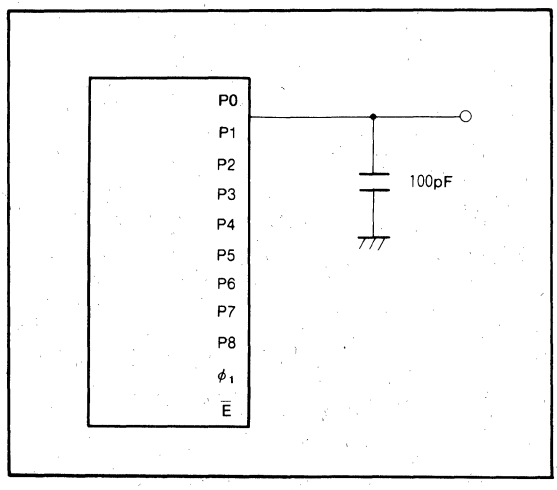
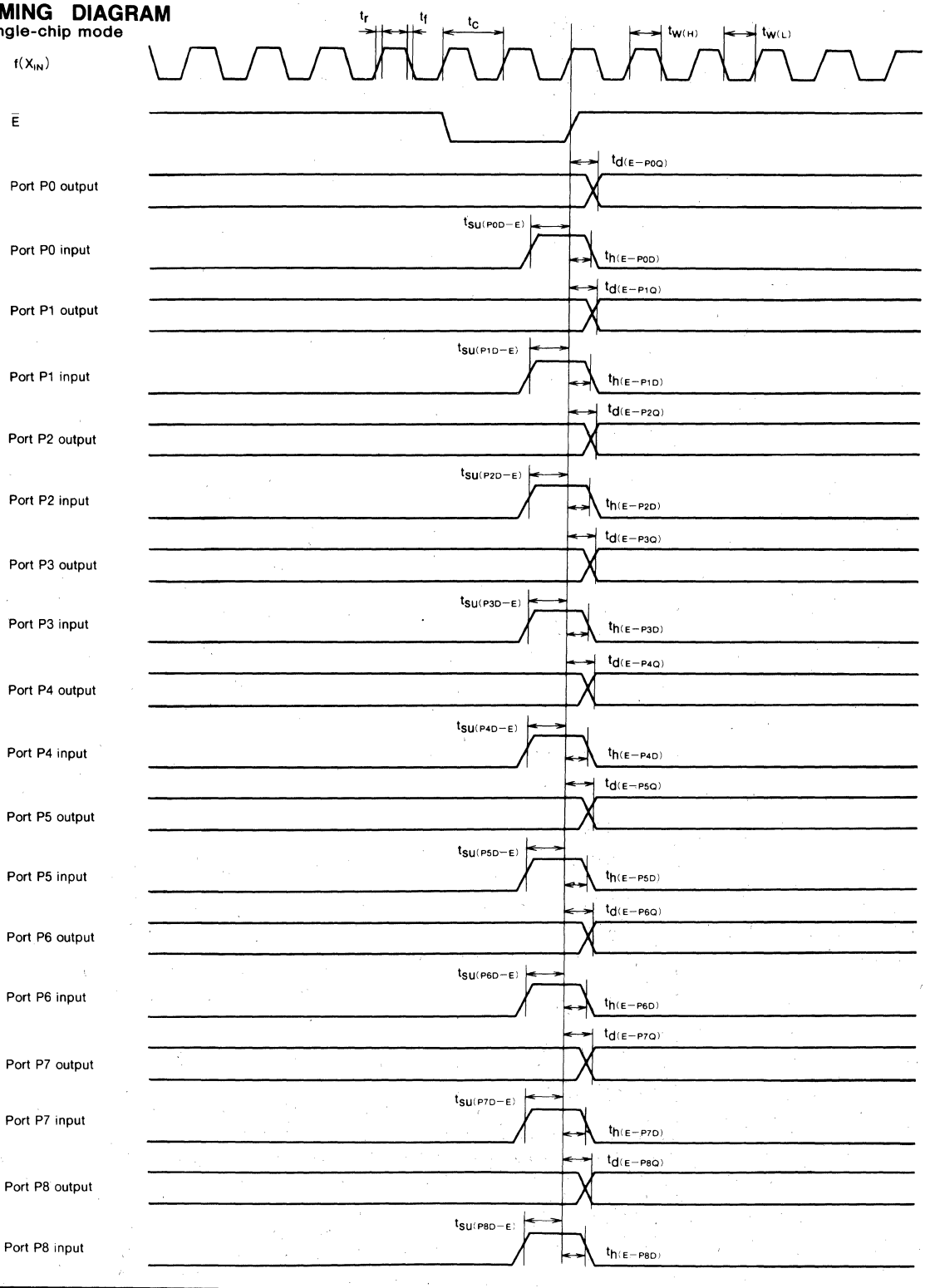
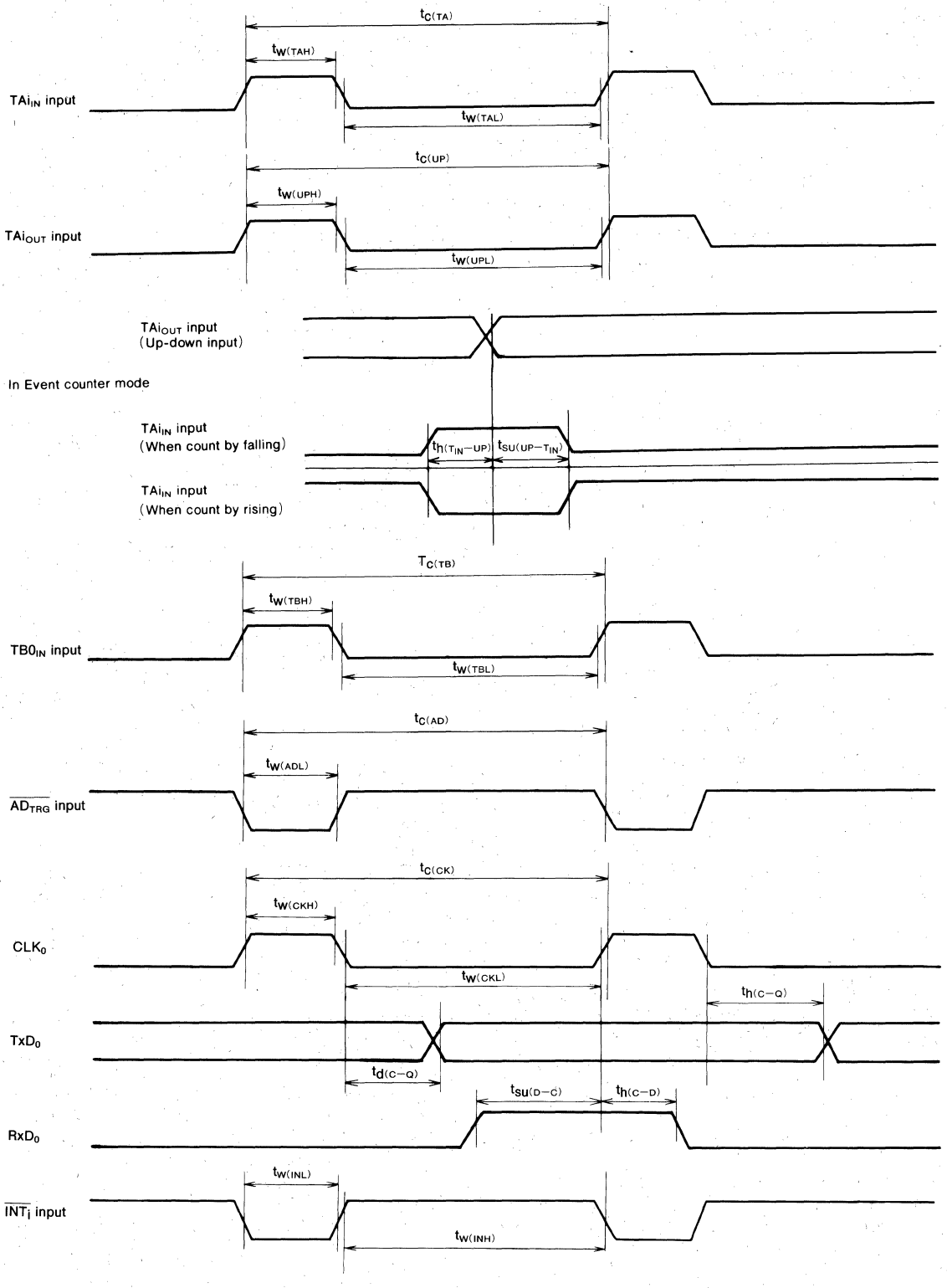


Fig. 1 Testing circuit for ports P0~P8, ϕ_1

TIMING DIAGRAM

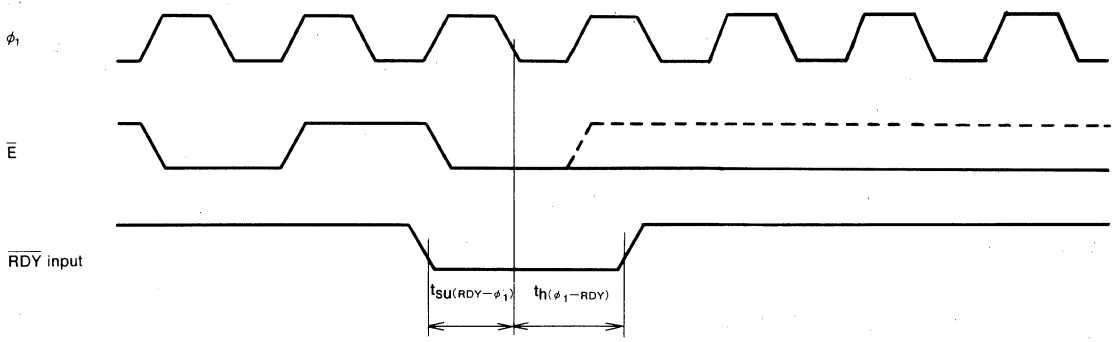
Single-chip mode



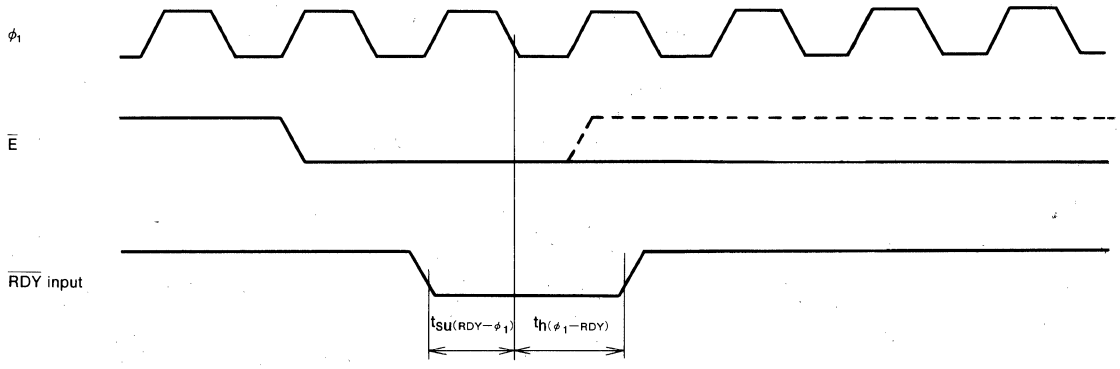

$t_{C(TA)}$
$t_{W(TAH)}$
TAi_{IN} input
$t_{W(TAL)}$
$t_{C(UP)}$
$t_{W(UPH)}$
TAi_{OUT} input
$t_{W(UPL)}$
TAi_{OUT} input
(Up-down input)
In Event counter mode
TAi_{IN} input
(When count by falling)
$t_{h(T_{IN}-UP)}$
$t_{SU(UP-T_{IN})}$
TAi_{IN} input
(When count by rising)
$T_{C(TB)}$
$t_{W(TBH)}$
$TB0_{IN}$ input
$t_{W(TBL)}$
$t_{C(AD)}$
$t_{W(ADL)}$
$\overline{AD_{TRG}}$ input
$t_{C(CK)}$
$t_{W(CKH)}$
CLK_0
$t_{W(CKL)}$
$t_{h(C-Q)}$
TxD_0
$t_{d(C-Q)}$
$t_{su(D-C)}$
$t_{h(C-D)}$
RxD_0
$t_{W(INL)}$
$\overline{INT_i}$ input
$t_{W(INH)}$

Memory expansion mode and microprocessor mode

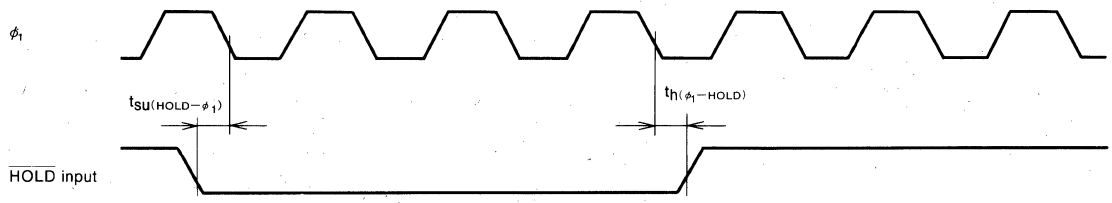
(When wait bit = "1")



(When wait bit = "0")



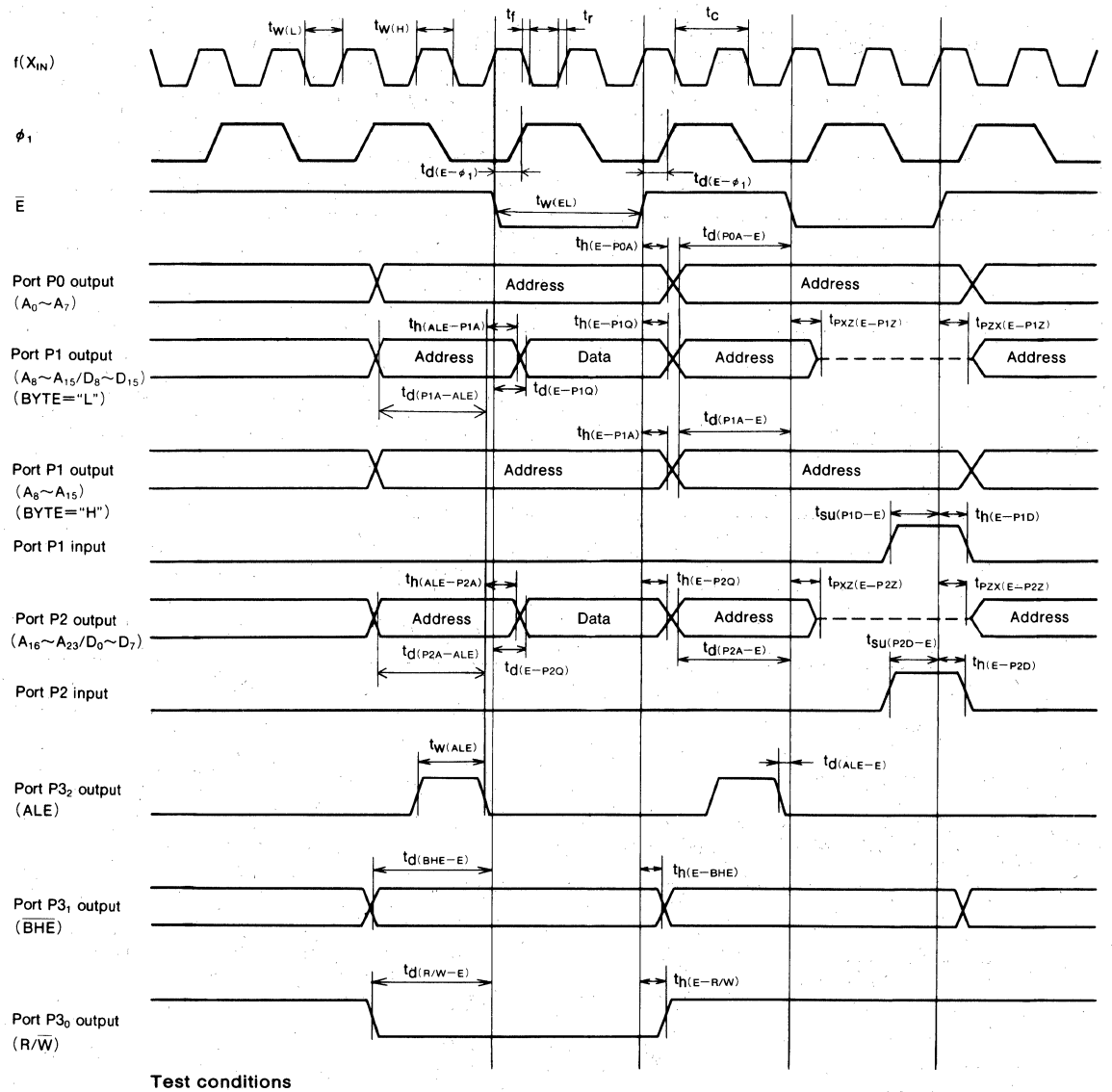
(When wait bit = "1" or "0" in common)



Test conditions

- V_{CC} = 5 V±10%
- Input timing voltage : V_{IL}=1.0V, V_{IH}=4.0V

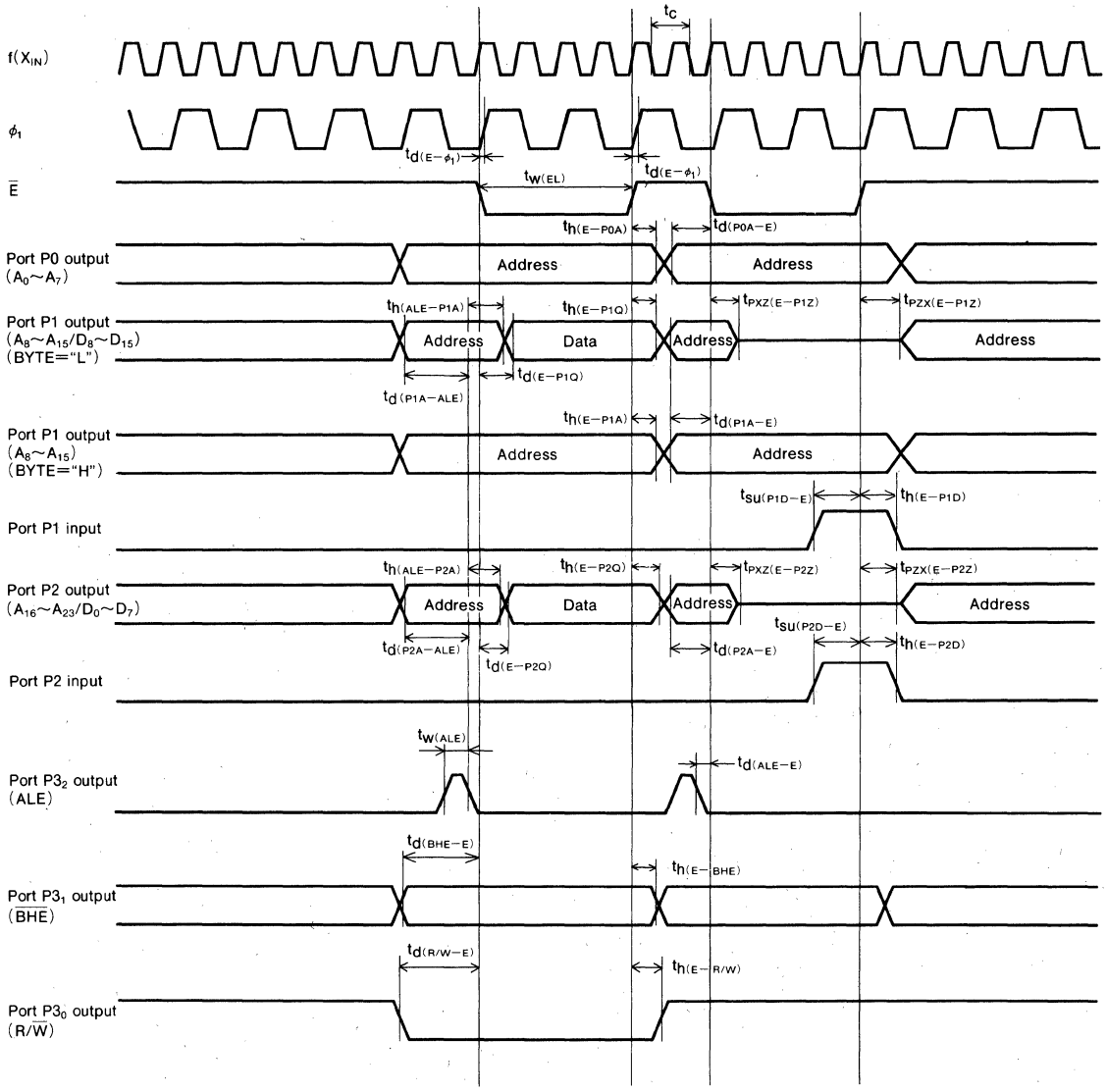
Memory expansion mode and microprocessor mode (When wait bit="1")



Test conditions

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

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



Test conditions

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

M37703M4-XXXSP,M37703M4AXXXSP,M37703M4BXXXSP M37703S4SP,M37703S4ASP,M37703S4BSP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

DESCRIPTION

The M37703M4-XXXSP is a single-chip microcomputer designed with high-performance CMOS silicon gate technology. This is housed in a 64-pin shrink plastic molded DIP. This single-chip microcomputer has a large 16M bytes address space, three instruction queue buffers, and two data buffers for high-speed instruction execution. The CPU is a 16-bit parallel processor that can also be switched to perform 8-bit parallel processing. This microcomputer is suitable for office, business and industrial equipment controller that require high-speed processing of large data.

The differences between M37703M4-XXXSP, M37703M4A XXXSP, M37703M4BXXXSP, M37703S4SP, M37703S4ASP and M37703S4BSP are the ROM size and the external clock input frequency as shown below. Therefore, the following descriptions will be for the M37703M4-XXXSP unless otherwise noted.

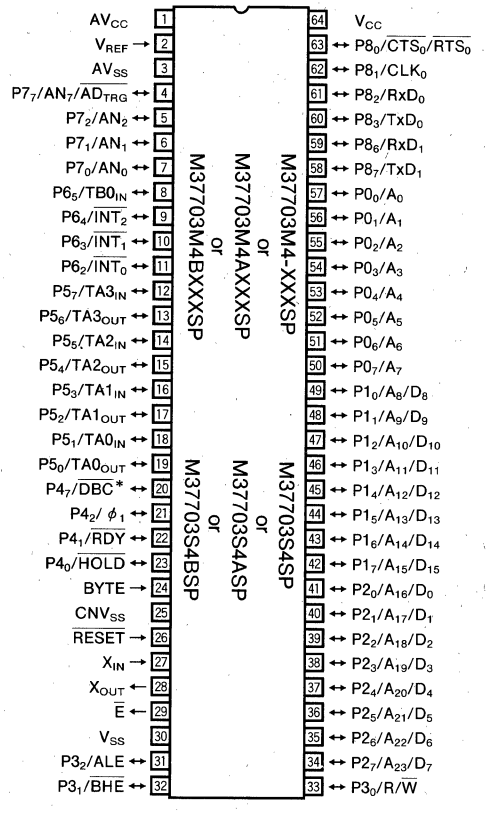
Type name	ROM size	External clock input frequency
M37703M4-XXXSP	32K bytes	8 MHz
M37703M4AXXXSP	32K bytes	16MHz
M37703M4BXXXSP	32K bytes	25MHz
M37703S4SP	External	8 MHz
M37703S4ASP	External	16MHz
M37703S4BSP	External	25MHz

The M37703M4-XXXSP has the same functions as the M37703M2-XXXSP except for the memory size.

FEATURES

- Number of basic instructions 103
- Memory size ROM ..... 32K bytes
  RAM ..... 2048 bytes
- Instruction execution time
  M37703M4-XXXSP, M37703S4SP
  (The fastest instruction at 8 MHz frequency) 500ns
  M37703M4AXXXSP, M37703S4ASP
  (The fastest instruction at 16 MHz frequency) 250ns
  M37703M4BXXXSP, M37703S4BSP
  (The fastest instruction at 25 MHz frequency) 160ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 8 MHz frequency)
  ..... 30mW (Typ.)
- Interrupts ..... 19 types 7 levels
- Multiple function 16-bit timer ..... 5+3
- UART (may also be synchronous) 2
- 8-bit A-D converter ..... 4-channel inputs
- 12-bit watchdog timer
- Programmable input/output
  (ports P0, P1, P2, P3, P4, P5, P6, P7, P8) 53

PIN CONFIGURATION (TOP VIEW)



Outline 64P4B

*: Used in the evaluation chip mode only

APPLICATION

Control devices for office equipment such as copiers, printers, typewriters, facsimiles, word processors, and personal computers

Control devices for industrial equipment such as ME, NC, communication, and measuring instruments

THE FUNCTIONS AND CHARACTERISTICS

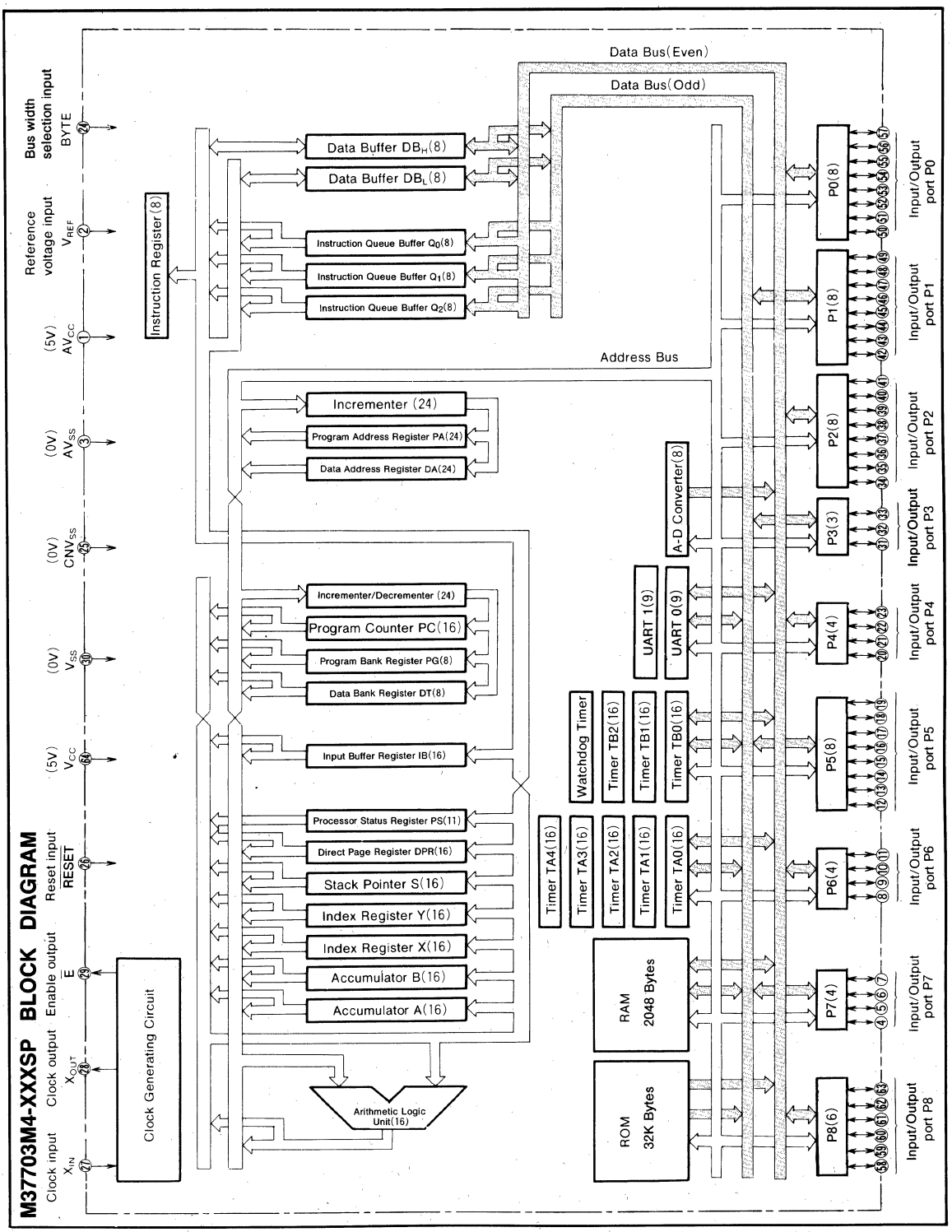
The M37703M4-XXXSP has the same functions and characteristics as the M37703M2-XXXSP except for the ROM and RAM size. Refer to the section on the M37703M2-XXXSP.

DATA REQUIRED FOR MASK ORDERING

Please send the following data for mask orders.

(1) M37703M4-XXXSP mask ROM order confirmation form
(2) 64P4B mark specification form
(3) ROM data (EPROM 3 sets)

M37703M4-XXXSP BLOCK DIAGRAM



M37720S1FP,M37720S1AFP

16-BIT CMOS MICROCOMPUTER

DESCRIPTION

The M37720S1FP and M37720S1AFP are 16-bit microcomputers designed with high-performance CMOS silicon gate technology. These are housed in a 100-pin plastic molded QFP.

These microcomputers have a large 16M bytes address space, three instruction queue buffers, and two data buffers for high-speed instruction execution. They also incorporate a 4-channel DMA controller and a DRAM controller which has a refresh function. The CPU is a 16-bit parallel processor that can also be switched to perform 8-bit parallel processing. These microcomputers are suitable for office, business, and industrial equipment controller that require high-speed processing of large data.

The differences between M37720S1FP and M37720S1AFP is the external clock input frequency as shown below. Therefore, the following descriptions will be for the M37720S1FP unless otherwise noted.

Type	ROM size	External clock input frequency
M37720S1FP	External	8 MHz
M37720S1AFP	External	16MHz

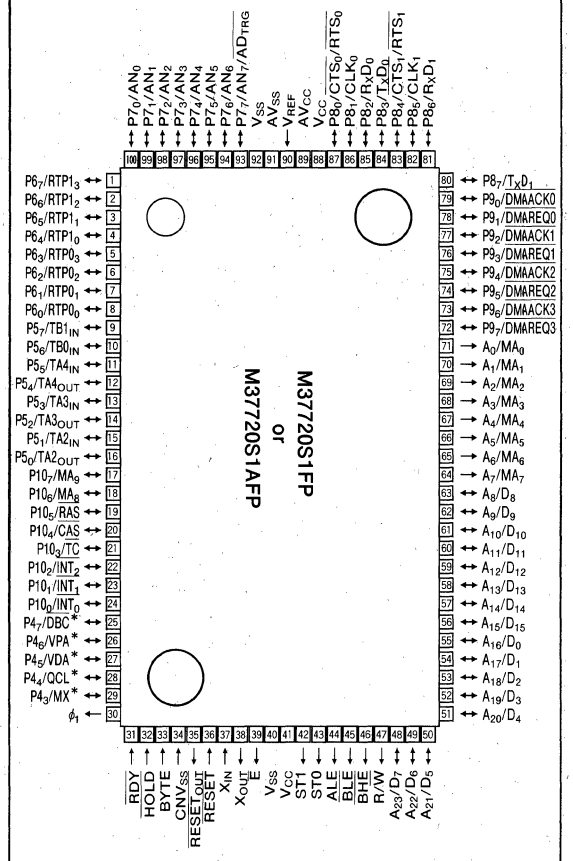
FEATURES

- Number of basic instructions 103
- Memory size RAM ..... 512 bytes
  ROM ..... External
- Instruction execution time
  M37720S1FP
  (The fastest instruction at 8MHz frequency)
  ..... 500ns
  M37720S1AFP
  (The fastest instruction at 16MHz frequency)
  ..... 250ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 8MHz frequency)
  ..... 45mW(Typ.)
- Interrupts ..... 23 types, 7 levels
- Multiple function 16-bit timer ..... 5+3
- UART (may also be synchronous) 2
- 8-bit A-D converter ..... 8-channel inputs
- DMA controller 4-channel
- DRAM controller
- 4-bit real-time output 2-channel
- Watchdog timer
- Programmable input/output
  (ports P4, P5, P6, P7, P8, P9, P10) 53

APPLICATION

Copiers, printers, typewriters, facsimiles, optical disk units, hard disk drives, telecommunications equipment such as mobile radio and ISDN terminal units, and office automation (OA) equipment including personal computers.

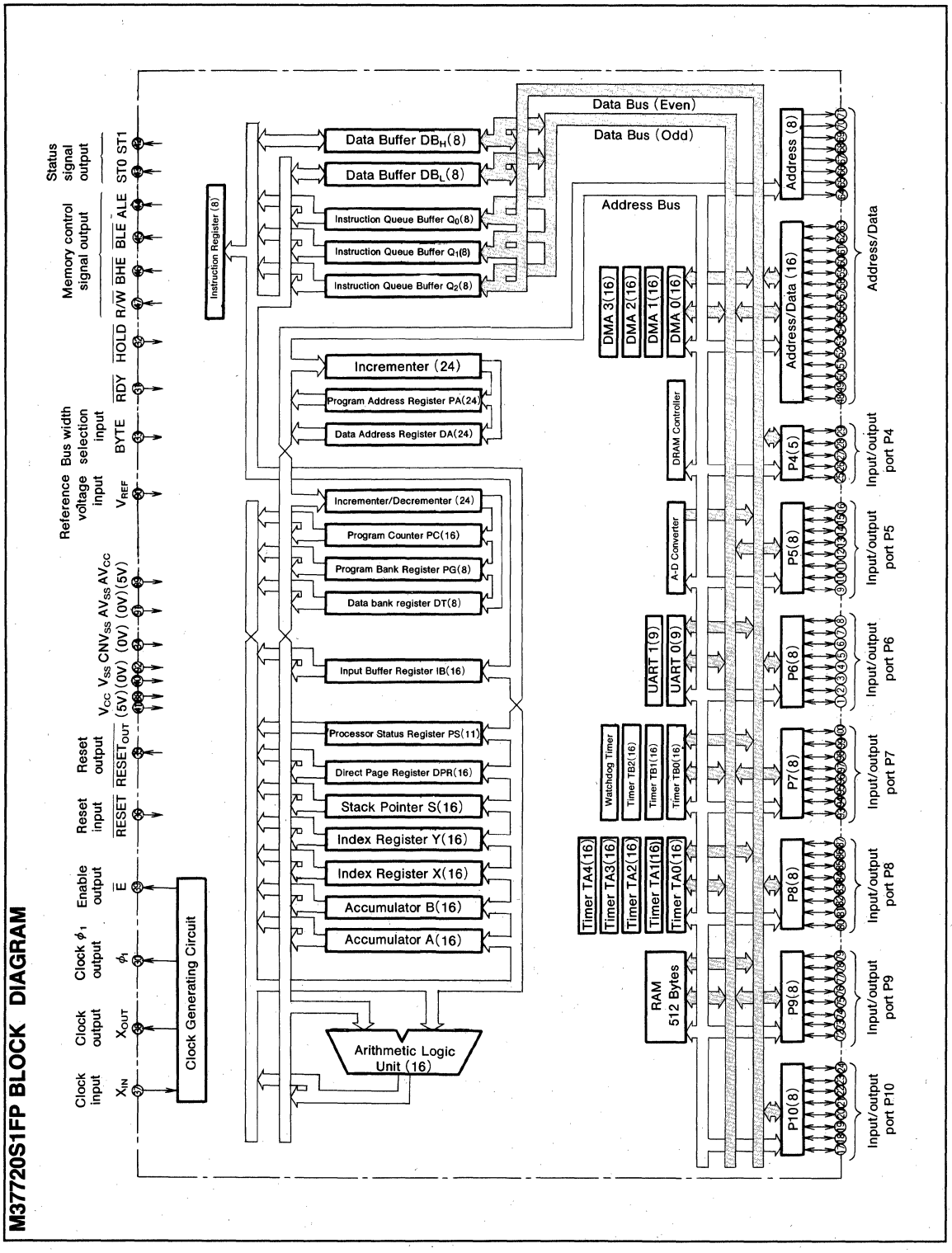
PIN CONFIGURATION (TOP VIEW)



Outline 100P6S

* : Used in the evaluation chip mode only

M37720S1FP BLOCK DIAGRAM



FUNCTIONS OF M37720S1FP

Parameter		Functions
Number of basic instructions		103
Instruction execution time	M37720S1FP	500ns (the fastest instructions, at 8MHz frequency)
	M37720S1AFP	250ns (the fastest instructions, at 16MHz frequency)
Memory size	RAM	512 bytes
	ROM	External
Input/Output ports	P5~P10	8-bit× 6
	P4	5-bit× 1
Multi-function timers	TA0, TA1, TA2, TA3, TA4	16-bit× 5
	TB0, TB1, TB2	16-bit× 3
Serial I/O		(UART or clock synchronous serial I/O)×2
A-D converter		8-bit× 1 (8 channels)
Watchdog timer		12-bit× 1
DMA controller		4 channels Maximum transfer rate : 8M bytes/second
DRAM controller		"$\overline{CAS}$ before $\overline{RAS}$ refresh" system 8-bit refresh timer incorporated
Real-time output		4 bit×2 channels
Interrupts		3 external types, 20 internal types (Each interrupt can be set the priority levels to 0 ~ 7.)
Clock generating circuit		Built-in (externally connected to a ceramic resonator or quartz crystal resonator)
Supply voltage		5 V±10%
Power dissipation		45mW(at external 8 MHz frequency)
Input/Output characteristic	Input/Output voltage	5 V
	Output current	5 mA
Memory expansion		Maximum 16M bytes
Operating temperature range		−20~85℃
Device structure		CMOS high-performance silicon gate process
Package		100-pin plastic molded QFP

PIN DESCRIPTION (1)

Pin	Name	Input/Output	Functions
V_{CC}, V_{SS}	Power supply		Supply 5 V±10% to V_{CC} and 0 V to V_{SS}.
CNV_{SS}	CNV_{SS} input	Input	Connect to V_{SS} or V_{CC}.
$\overline{RESET}$	Reset input	Input	To enter the reset state, this pin must be kept at a "L" condition should be maintained for the required time.
$\overline{RESETout}$	Reset output	Output	This pin outputs the response of reset input. When input to $\overline{RESET}$ pin is "L", this pin outputs "L". And output from this pin returns "H" after the release of reset. When the software reset bit is set to "1", this pin outputs "L".
X_{IN}	Clock input	Input	This is an input pin for internal clock generating circuit. Connect a ceramic or quartz crystal resonator between X_{IN} and X_{OUT}. When an external clock is used, the clock source should be connected to the X_{IN} pin and the X_{OUT} pin should be left open.
X_{OUT}	Clock output	Output	This is an output pin for internal clock generating circuit. When a resonator is used, the resonator should be connected between this pin and the X_{IN} pin.
$\overline{E}$	Enable output	Output	Data or instruction read and data write are performed when output from this pin is "L".
BYTE	Bus width selection input	Input	This pin determines whether the external data bus is 8-bit width or 16-bit width. The width is 16 bits when "L" signal inputs and 8 bits when "H" signal inputs.
ST0, ST1	Status signal output	Output	The bus use status output is generated in 2-bit code. Refresh: ST1 0, ST0 0 Hold: ST1 0, ST0 1 DMA: ST1 1, ST0 0 CPU: ST1 1, ST0 1
AV_{CC}, AV_{SS}	Analog supply input		Power supply for the A-D converter. Connect AV_{CC} to V_{CC} and AV_{SS} to V_{SS} externally.
V_{REF}	Reference voltage input	Input	This is reference voltage input pin for the A-D converter.
R/$\overline{W}$, $\overline{BHE}$, ALE, $\overline{BLE}$	Memory control signal output	Output	These pins are for R/$\overline{W}$, $\overline{BHE}$, ALE, and $\overline{BLE}$ output pins.
ϕ1	Clock ϕ1 output	Output	This is the ϕ1 output pin which is divided the clock to X_{IN} pin by 2.
$\overline{HOLD}$	$\overline{HOLD}$ input	Input	This is an input pin for the $\overline{HOLD}$ request signal.
$\overline{RDY}$	$\overline{RDY}$ input	Input	This is an input pin for the $\overline{RDY}$ signal.
A_0/MA_0~ A_7/MA_7	Address low-order/ DRAM address	Output	These are output pins for the 8 low-order bits of addresses. When the DRAM is to be accessed, the row and column addresses are generated by means of time division multiplexing.
A_8/D_8~ A_{15}/D_{15}	Address medium-order/ data high-order	I/O	In cases where an external data bus width of 16 bits is used with the BYTE pin set to the "L" (Low) level, the high-order data (D_{15}~D_8) input/output is effected at the "L" $\overline{E}$ output level, and the address output (A_{15}~A_8) is generated at the "H" (High) $\overline{E}$ output level. In cases where an external data bus width of 8 bits is used with the BYTE pin set to the "H" level, only the address (A_{15}~A_8) output is generated.
A_{16}/D_0~ A_{23}/D_7	Address high-order/ data low-order	I/O	When the $\overline{E}$ output is "L", the low-order data (D_7~D_0) input/output is effected. When the $\overline{E}$ output is "H", the address output (A_{23}~A_{16}) is generated.
$P4_3$~$P4_7$	I/O port P4	I/O	Port P4 is an 5-bit I/O port. As it has an I/O directional register, it is possible to perform programming to determine whether each bit serves as an input pin or an output pin.
$P5_0$~$P5_7$	I/O port P5	I/O	Port P5 is an 8-bit I/O port. It has basically the same functions as port P4. In addition, these pins also function as I/O pins for timer A2, timer A3 and timer A4, and input pins for timer B0 and timer B1.

PIN DESCRIPTION (2)

Pin	Name	Input/Output	Functions
$P6_0$~$P6_7$	I/O port P6	I/O	Port P6 is an 8-bit I/O port. It has basically the same functions as port P4. In addition, pins $P6_0$ through $P6_3$ and pins $P6_4$ through $P6_7$ are capable of functioning as a 4-bit real-time output, respectively.
$P7_0$~$P7_7$	I/O port P7	I/O	Port P7 is an 8-bit I/O port. It has basically the same functions as port P4. In addition, these pins also function as input pins for analog inputs AN_0 through AN_7. Pin $P7_7$ also has an A-D conversion trigger input function.
$P8_0$~$P8_7$	I/O port P8	I/O	Port P8 is an 8-bit I/O port. It has basically the same functions as port P4. In addition, these pins also function as R_XD, T_XD, CLK, and $\overline{CTS}/\overline{RTS}$ pins for UART0 and UART1.
$P9_0$~$P9_7$	I/O port P9	I/O	Port P9 is an 8-bit I/O port. It has basically the same functions as port P4. In addition, these pins also function as input pins for DMA request, and output pins for DMA acknowledge signal.
$P10_0$~$P10_7$	I/O port P10	I/O	Port P10 is an 8-bit I/O port. It has basically the same functions as port P4. In addition, these pins also function as input pins for $\overline{INT_0}/\overline{INT_1}/\overline{INT_2}$, and I/O pin for $\overline{TC}$, and output pins for $\overline{RAS}$, $\overline{CAS}$, MA_8 and MA_9.

BASIC FUNCTION BLOCKS

The M37720S1FP contains the following devices on a chip: RAM for storing instructions and data, CPU for processing, bus interface unit (which controls instruction prefetch and data read/write between CPU and memory), timers, UART, A-D converter, and other peripheral devices such as I/O ports. ROM is not incorporated. Each of these devices are described below.

MEMORY

The memory map is shown in Figure 1. The address space is 16M bytes from addresses 0_{16} to FFFFFF$_{16}$. The address space is divided into 64K bytes units called banks. The banks are numbered from 0_{16} to FF$_{16}$.

Built-in RAM, and control registers for built-in peripheral devices are assigned to bank 0_{16}.

Addresses FFCE$_{16}$ to FFFF$_{16}$ are the RESET and interrupt vector addresses and store the interrupt vectors. Be sure to set up external ROM for the interrupt vector table. Refer to the section on interrupts for details.

The 512 bytes area from addresses 80_{16} to $27F_{16}$ contains the built-in RAM. In addition to storing data, the RAM is used as stack during a subroutine call, or interrupts.

Assigned to addresses 0_{16} to $7F_{16}$ are peripheral devices such as I/O ports, A-D converter, UART, timer, and interrupt control registers. Registers necessary for DMA control are assigned to addresses 1FC0$_{16}$ through 1FFF$_{16}$.

A 256 bytes direct page area can be allocated anywhere in bank 0_{16} using the direct page register DPR. In direct page addressing mode, the memory in the direct page area can be accessed with two words thus reducing program steps.

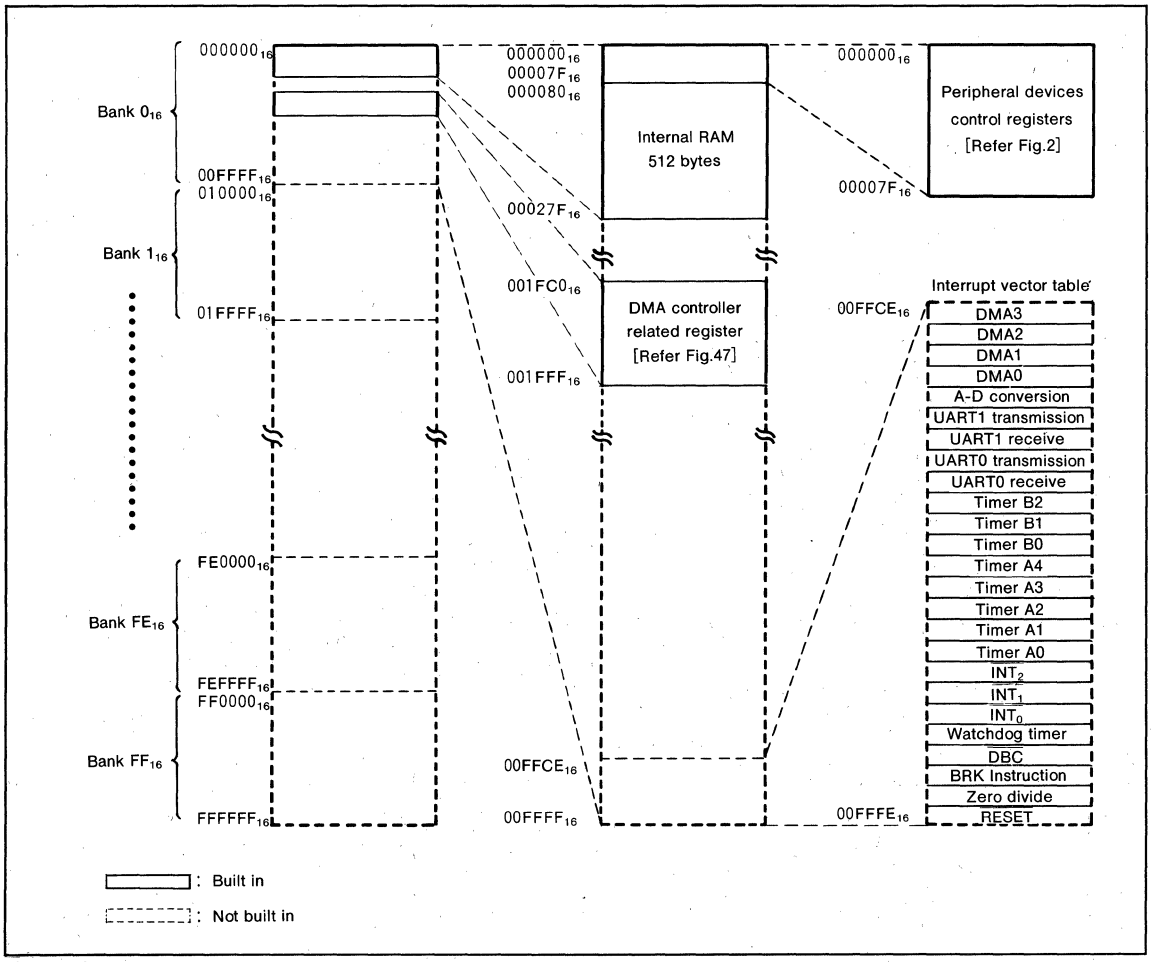


Fig.1 Memory map

Address (Hexadecimal notation)	
000000	
000001	
000002	
000003	
000004	
000005	
000006	
000007	
000008	
000009	
00000A	Port P4
00000B	Port P5
00000C	Port P4 data direction register
00000D	Port P5 data direction register
00000E	Port P6
00000F	Port P7
000010	Port P6 data direction register
000011	Port P7 data direction register
000012	Port P8
000013	Port P9
000014	Port P8 data direction register
000015	Port P9 data direction register
000016	Port P10
000017	
000018	Port P10 data direction register
000019	
00001A	Pulse output data register 0
00001B	
00001C	Pulse output data register 1
00001D	
00001E	A-D control register
00001F	A-D sweep pin selection register
000020	A-D register 0
000021	
000022	A-D register 1
000023	
000024	A-D register 2
000025	
000026	A-D register 3
000027	
000028	A-D register 4
000029	
00002A	A-D register 5
00002B	
00002C	A-D register 6
00002D	
00002E	A-D register 7
00002F	
000030	UART0 transmit/receive mode register
000031	UART0 bit rate register
000032	UART0 transmission buffer register
000033	
000034	UART0 transmit/receive control register 0
000035	UART0 transmit/receive control register 1
000036	UART0 receive buffer register
000037	
000038	UART1 transmit/receive mode register
000039	UART1 bit rate register
00003A	UART1 transmission buffer register
00003B	
00003C	UART1 transmit/receive control register 0
00003D	UART1 transmit/receive control register 1
00003E	UART1 receive buffer register
00003F	

Address (Hexadecimal notation)	
000040	Count start flag
000041	
000042	One shot start flag
000043	
000044	Up-down flag
000045	
000046	Timer A0
000047	
000048	Timer A1
000049	
00004A	Timer A2
00004B	
00004C	Timer A3
00004D	
00004E	Timer A4
00004F	
000050	Timer B0
000051	
000052	Timer B1
000053	
000054	Timer B2
000055	
000056	Timer A0 mode register
000057	Timer A1 mode register
000058	Timer A2 mode register
000059	Timer A3 mode register
00005A	Timer A4 mode register
00005B	Timer B0 mode register
00005C	Timer B1 mode register
00005D	Timer B2 mode register
00005E	Processor mode register
00005F	
000060	Watchdog timer
000061	Watchdog timer frequency selection flag
000062	Real-time output control register
000063	
000064	DRAM control register
000065	
000066	Refresh timer
000067	
000068	DMAC control register L
000069	DMAC control register H
00006A	
00006B	
00006C	DMA0 interrupt control register
00006D	DMA1 interrupt control register
00006E	DMA2 interrupt control register
00006F	DMA3 interrupt control register
000070	A-D conversion interrupt control register
000071	UART0 transmission interrupt control register
000072	UART0 receive interrupt control register
000073	UART1 transmission interrupt control register
000074	UART1 receive interrupt control register
000075	Timer A0 interrupt control register
000076	Timer A1 interrupt control register
000077	Timer A2 interrupt control register
000078	Timer A3 interrupt control register
000079	Timer A4 interrupt control register
00007A	Timer B0 interrupt control register
00007B	Timer B1 interrupt control register
00007C	Timer B2 interrupt control register
00007D	$\overline{INT_0}$ interrupt control register
00007E	$\overline{INT_1}$ interrupt control register
00007F	$\overline{INT_2}$ interrupt control register

Fig.2 Location of peripheral devices and interrupt control registers

CENTRAL PROCESSING UNIT (CPU)

The CPU has ten registers and is shown in Figure 3. Each of these registers is described below.

ACCUMULATOR A (A)

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

Data operations such as calculations, data transfer, input/output, etc., is executed mainly through the accumulator.

ACCUMULATOR B (B)

Accumulator B has the same functions as accumulator A, but the use of accumulator B requires more instruction bytes and execution cycles than accumulator A.

INDEX REGISTER X (X)

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

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

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

INDEX REGISTER Y (Y)

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

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

Also, when executing a block transfer instruction MVP or MVN, the content of index register Y indicates the low-order 16 bits of the destination address. The second byte of the MVP and MVN is the high-order 8 bits of the destination data address.

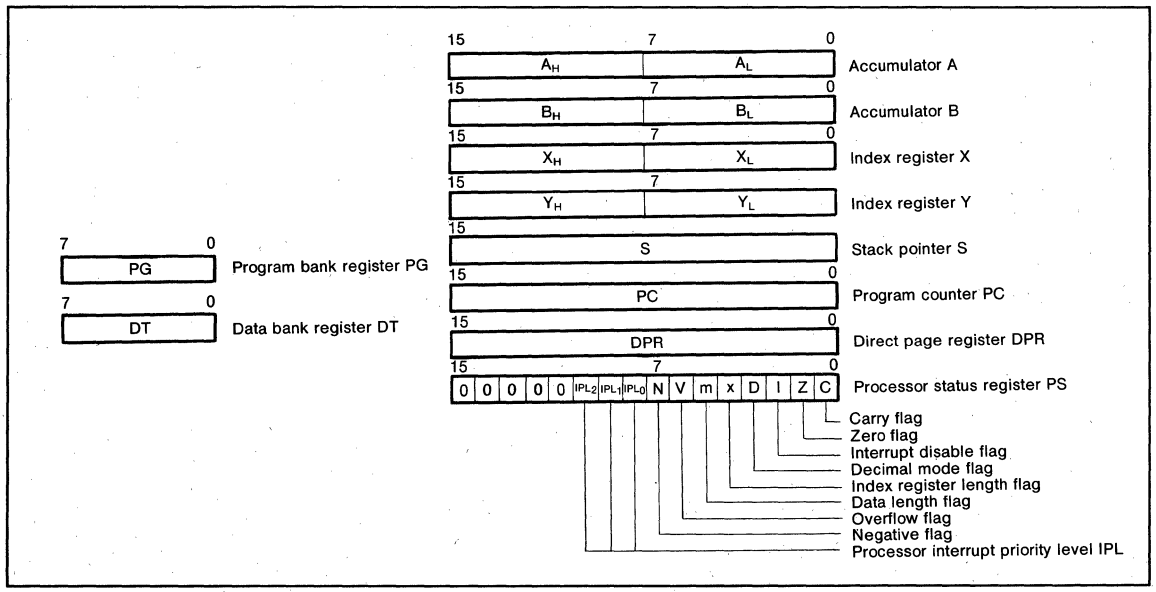


Fig.3 Register structure

STACK POINTER (S)

Stack pointer (S) is a 16-bit register. It is used during a subroutine call or interrupts. It is also used during stack, stack pointer relative, or stack pointer relative indirect indexed Y addressing mode.

For stack area selection, use bit 7 stack bank selection bit of the processor mode register (address $5E_{16}$). When the bit is set to "0", the stack area is set to bank 0_{16}. When the bit is "1", the stack area is set to bank FF_{16}. This bit defaults to 0 upon resetting.

PROGRAM COUNTER (PC)

Program counter (PC) is a 16-bit counter that indicates the low-order 16 bits of the next program memory address to be executed. There is a bus interface unit between the program memory and the CPU, so that the program memory is accessed through bus interface unit. That is described later.

PROGRAM BANK REGISTER (PG)

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

DATA BANK REGISTER (DT)

Data bank register (DT) is an 8-bit register. With some addressing modes, a part of the data bank register (DT) is used to specify a memory address.

The contents of data bank register (DT) is used as the high-order 8 bits of a 24-bit address. Addressing modes that use the data bank register (DT) are direct indirect, direct indexed X indirect, direct indirect indexed Y, absoulute, absolute bit, absolute indexed X, absolute indexed Y, absolute bit relative, and stack pointer relative indirect indexed Y.

DIRECT PAGE REGISTER (DPR)

Direct page register (DPR) is a 16-bit register. Its contents is used as the base address of a 256byte direct page area. The direct page area is allocated in bank 0_{16}, but when the contents of DPR is $FF01_{16}$ or greater, the direct page area spans across bank 0_{16} and bank 1_{16}. All direct addressing modes use the contents of the direct page register (DPR) to generate the data address. If the low-order 8 bits of the direct page register (DPR) is "00_{16}", the number of cycles required to generate the address is minimized. Therefore, nomally the low-order 8 bits of the direct page register (DPR) is set to "00_{16}".

PROCESSOR STATUS REGISTER (PS)

Processor status register (PS) is an 11-bit register. It consists of a flag to indicate the result of operation and CPU interrupt levels.

Branch operations can be performed by testing the flags, C, Z, V, and N.

The details of each processor status register bit are described below.

1. Carry flag (C)

The carry flag contains the carry or borrow generated by the ALU after an arithmetic operation. This flag is also affected by shift and rotate instructions. This flag can be set and reset directly with the SEC and CLC instructions or with the SEP and CLP instructions.

2. Zero flag (Z)

This zero flag is set if the result of an arithmetic operation or data transfer is zero and reset if it is not. This flag can be set and reset directly with the SEP and CLP instructions.

3. Interrupt disable flag (I)

When the interrupt disable flag is set to "1", all interrupts except watchdog timer, $\overline{DBC}$, and software interrupt are disabled. This flag is set to "1" automatically when interrupts are serviced. It can be set and reset directly with the SEI and CLI instructions or SEP and CLP instructions.

4. Decimal mode flag (D)

The decimal mode flag determines whether addition and subtraction are performed as binary or decimal.

Binary arithmetic is performed when this flag is "0". If it is "1", decimal arithmetic is performed with each word treated as two or four digit decimal. Arithmetic operation is performed using four digits when the data length flag m is "0" and with two digits when it is "1". (Decimal operation is possible only with the ADC and SBC instructions.) This flag can be set and reset with the SEP and CLP instructions.

5. Index register length flag (X)

The index register length flag determines whether index register X and index register Y are used as 16-bit registers or as 8-bit registers. The registers are used as 16-bit registers when flag x is "0" and as 8-bit registers when it is "1". This flag can be set and reset with the SEP and CLP instructions.

6. Data length flag (m)

The data length flag determines whether the data length is 16-bit or 8-bit. The data length is 16-bit when flag m is "0" and 8-bit when it is "1". This flag can be set and reset with the SEM and CLM instructions or with the SEP and CLP instructions.

7. Overflow flag (V)

The overflow flag has meaning when addition or subtraction is performed as signed binary number. When the data length flag m is "0", the overflow flag is set when the result of addition or subtraction is outside the range between −32768 and +32767. When the data length flag m is "1", the overflow flag is set when the result of addition or subtraction is outside the range between −128 and +127. It is reset in all other cases. The overflow flag can also be set and reset directly with the SEP, and CLV or CLP instructions.

8. Negative flag (N)

The negative flag is set when the result of arithmetic operation or data transfer is negative (If data length flag m is "0", when data bit 15 is "1". If data length flag m is "1", when data bit 7 is "1".) It is reset in all other cases. It can also be set and reset with the SEP and CLP instructions.

9. Processor interrupt priority level (IPL)

The processor interrupt priority level (IPL) consists of 3 bits and determines the priority of processor interrupts from level 0 to level 7. Interrupt is enabled when the interrupt priority (set using the interrupt control register) of the device requesting interrupt is higher than the processor interrupt priority. When interrupt is enabled, the current processor interrupt priority level is saved in a stack and the processor interrupt priority level is replaced by the interrupt priority level of the device requesting the interrupt. Refer to the section on interrupts for more details.

BUS INTERFACE UNIT

The CPU operates on an internal clock frequency which is obtained by dividing the external clock frequency f(X_{IN}) by two. This frequency is twice the bus cycle frequency. In order to speed-up processing, a bus interface unit is used to pre-fetch instructions when the data bus is idle. The bus interface unit synchronizes the CPU and the bus access timing and pre-feches instructions. Figure 4 shows the relationship between the CPU and the bus interface unit. The bus interface unit has a program address register, a 3-byte instruction queue buffer, a data address register, and a 2-byte data buffer.

The bus interface unit obtains an instruction code from memory and stores it in the instruction queue buffer, obtains data from memory and stores it in the data buffer, or writes the data from the data buffer to the memory.

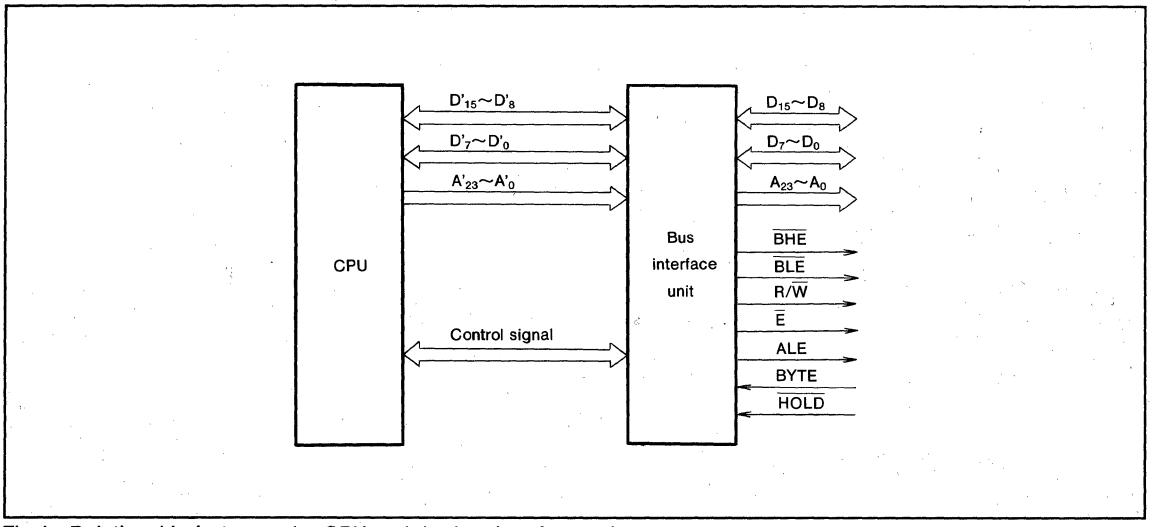


Fig.4 Relationship between the CPU and the bus interface unit

The bus interface unit operates using one of the waveforms (1) to (6) shown in Figure 5. The standard waveforms are (1) and (2).

The ALE signal is used to latch only the address signal from the multiplexed signal containing data and address.

The $\overline{E}$ signal becomes "L" when the bus interface unit reads an instruction code or data from memory or when it writes data to memory. Whether to perform read or write is controlled by the R/$\overline{W}$ signal. Read is performed when the R/$\overline{W}$ signal is "H" state and write is performed when it is "L" state.

Waveform (1) in Figure 5 is used to access a single byte or two bytes simultaneously. To read or write two bytes simultaneously, the first address accessed must be even. Furthermore, when accessing an external memory area, set the bus width selection input pin BYTE to "L". (external data bus width to 16 bits) The internal memory area is always treated as 16-bit bus width regardless of BYTE.

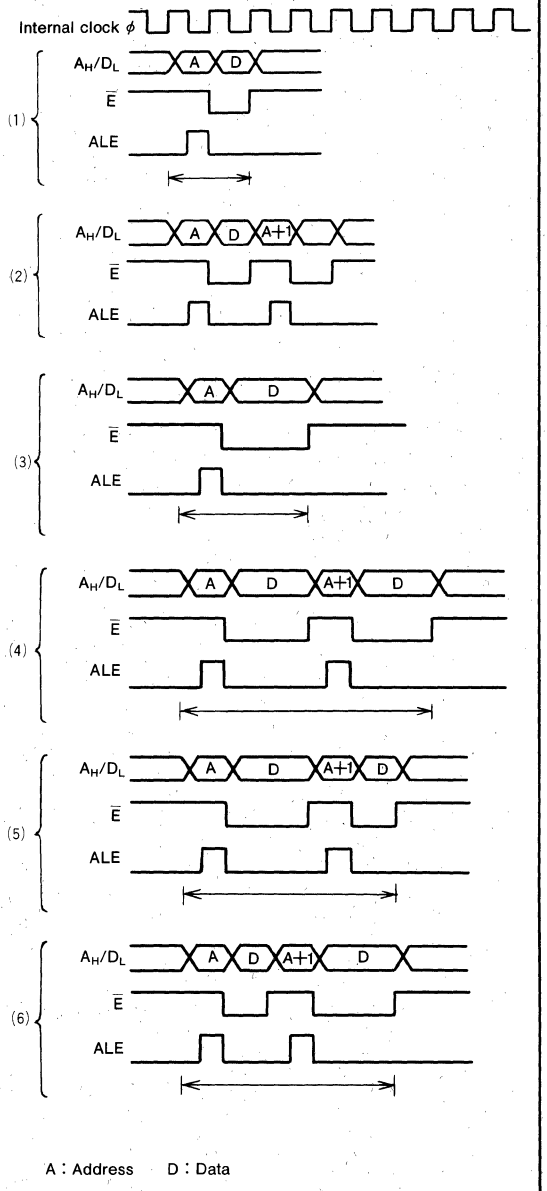
When performing 16-bit data read or write, if the conditions for simultaneously accessing two bytes are not satisfied, waveform (2) is used to access each byte one by one. However, when prefetching the instruction code, if the address of the instruction code is odd, only one byte is read in the instruction queue buffer.

The signals $\overline{BLE}$ and $\overline{BHE}$ in Figure 5 are used to control these cases: 1-byte read from even address, 1-byte read from odd address, 2-byte simultaneous read from even and odd addresses, 1-byte write to even address, 1-byte write to odd address, or 2-byte simultaneous write to even and odd addresses.

The $\overline{BLE}$ signal becomes "L" when an even number address is accessed. The $\overline{BHE}$ signal becomes "L" when an odd number address is accessed.

The bit 2 of processor mode register (address $5E_{16}$) is the wait bit. When this bit is set to "0", the "L" width of $\overline{E}$ signal is extended 2 times as long when accessing an external memory area in microprocessor mode. However, the "L" width of $\overline{E}$ signal is not extended when an internal memory area is accessed. When the wait bit is "1", the "L" width of $\overline{E}$ signal is not extended for any access. Waveform (3) is an expansion of the "L" width of $\overline{E}$ signal in waveform (1). Waveform (4), (5), and (6) are expansion of the "L" width of each $\overline{E}$ signal in waveform (2), first half of waveform (2), and the last half of waveform (2) respectively.

Instruction code read, data read, and data write are described below.



Signal \ Access method	Access 2-byte simultaneously	Access even address 1-byte	Access odd address 1-byte
$\overline{BLE}$	"L"	"L"	"H"
$\overline{BHE}$	"L"	"H"	"L"

Fig.5 Relationship between access method and signals $\overline{BLE}$ and $\overline{BHE}$

Instruction code read will be described first.
The CPU obtains instruction codes from the instruction queue buffer and executes them. The CPU notifies the bus interface unit that it is requesting an instruction code during an instruction code request cycle. If the requested instruction code is not yet stored in the instruction queue buffer, the bus interface unit halts the CPU until it can store more instructions than requested in the instruction queue buffer. Even if there is no instruction code request from the CPU, the bus interface unit reads instruction codes from memory and stores them in the instruction queue buffer when the instruction queue buffer is empty or when only one instruction code is stored and the bus is idle on the next cycle.
This is referred to as instruction pre-fetching.
Normally, when reading an instruction code from memory, if the accessed address is even the next odd address is read together with the instruction code and stored in the instruction queue buffer.
However, if the bus width switching pin BYTE is "H", external data bus width is 8 bits and the address to be read is in external memory area, or the addresses to be read is odd, only one byte is read and stored in the instruction queue buffer. Therefore, waveform (1) or (3) in Figure 5 in used for instruction code read.
Data read and write are described below.
The CPU notifies the bus interface unit when performing data read or write. At this time, the bus interface unit halts the CPU if the bus interface unit is already using the bus or if there is a request with higher priority. When data read or write is enabled, the bus interface unit uses one of the waveforms from (1) to (6) in Figure 5 to perform the operation.
During data read, the CPU waits until the entire data is stored in the data buffer. The bus interface unit sends the address received from the CPU to the address bus. Then it reads the memory when the $\overline{E}$ signal is "L" and stores the result in the data buffer.
During data write, the CPU writes the data in the data buffer and the bus interface unit writes it to memory. Therefore, the CPU can proceed to the next step without waiting for write to complete. The bus interface unit sends the address received from the CPU to the address bus. Then when the $\overline{E}$ signal is "L", the bus interface unit sends the data in the data buffer to the data bus writes it to memory.

INTERRUPTS

Table 1 shows the interrupt sources and the corresponding interrupt vector addresses. Reset is also treated as a kind of interrupt and is discussed in this section, too.

$\overline{DBC}$ is an interrupt used for debugging.

Interrupts other than reset, $\overline{DBC}$, watchdog timer, zero divide, and BRK instruction all have interrupt control registers. Table 2 shows the addresses of the interrupt control registers and Figure 6 shows the bit configuration of the interrupt control register. The interrupt request bit is automatically cleared by the hardware during reset or when processing an interrupt. Also, Interrupt request bits other than $\overline{DBC}$ and watchdog timer can be cleared by software.

$\overline{INT_2}$ to $\overline{INT_0}$ are external interrupts and whether to cause an interrupt at the input level (level sense) or at the edge (edge sense) can be selected with the level sense/edge sense selection bit. Furthermore, the polarity of the interrupt input can be selected with polarity selection bit.

Timer and UART interrupts described in the respective section.

The priority of interrupts when multiple interrupts are caused simultaneously is partially fixed by hardware, but, it can also be adjusted by software as shown in Figure 7. The hardware priority is fixed the following:

reset>$\overline{DBC}$>watchdog timer>other interrupts

Table 1. Interrupt sources and the interrupt vector addresses

Interrupts	Vector addresses	
DMA3	$00FFCE_{16}$	$00FFCF_{16}$
DMA2	$00FFD0_{16}$	$00FFD1_{16}$
DMA1	$00FFD2_{16}$	$00FFD3_{16}$
DMA0	$00FFD4_{16}$	$00FFD5_{16}$
A-D conversion	$00FFD6_{16}$	$00FFD7_{16}$
UART1 transmit	$00FFD8_{16}$	$00FFD9_{16}$
UART1 receive	$00FFDA_{16}$	$00FFDB_{16}$
UART0 transmit	$00FFDC_{16}$	$00FFDD_{16}$
UART0 receive	$00FFDE_{16}$	$00FFDF_{16}$
Timer B2	$00FFE0_{16}$	$00FFE1_{16}$
Timer B1	$00FFE2_{16}$	$00FFE3_{16}$
Timer B0	$00FFE4_{16}$	$00FFE5_{16}$
Timer A4	$00FFE6_{16}$	$00FFE7_{16}$
Timer A3	$00FFE8_{16}$	$00FFE9_{16}$
Timer A2	$00FFEA_{16}$	$00FFEB_{16}$
Timer A1	$00FFEC_{16}$	$00FFED_{16}$
Timer A0	$00FFEE_{16}$	$00FFEF_{16}$
$\overline{INT_2}$ external interrupt	$00FFF0_{16}$	$00FFF1_{16}$
$\overline{INT_1}$ external interrupt	$00FFF2_{16}$	$00FFF3_{16}$
$\overline{INT_0}$ external interrupt	$00FFF4_{16}$	$00FFF5_{16}$
Watchdog timer	$00FFF6_{16}$	$00FFF7_{16}$
$\overline{DBC}$ (unusable)	$00FFF8_{16}$	$00FFF9_{16}$
Break instruction	$00FFFA_{16}$	$00FFFB_{16}$
Zero divide	$00FFFC_{16}$	$00FFFD_{16}$
Reset	$00FFFE_{16}$	$00FFFF_{16}$

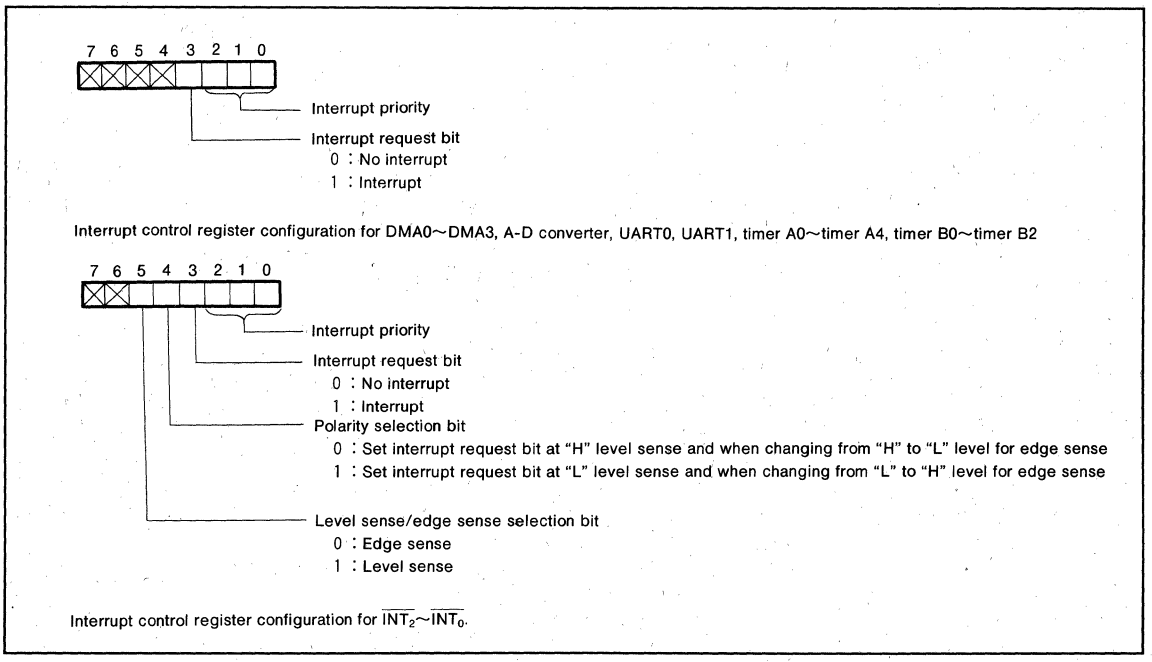


Fig.6 Interrupt control register configuration

Table 2. Address of interrupt control registers

Interrupts	Addresses
DMA0 interrupt control register	$00006C_{16}$
DMA1 interrupt control register	$00006D_{16}$
DMA2 interrupt control register	$00006E_{16}$
DMA3 interrupt control register	$00006F_{16}$
A-D conversion interrupt control register	000070_{16}
UART0 transmit interrupt control register	000071_{16}
UART0 receive interrupt control register	000072_{16}
UART1 transmit interrupt control register	000073_{16}
UART1 receive interrupt control register	000074_{16}
Timer A0 interrupt control register	000075_{16}
Timer A1 interrupt control register	000076_{16}
Timer A2 interrupt control register	000077_{16}
Timer A3 interrupt control register	000078_{16}
Timer A4 interrupt control register	000079_{16}
Timer B0 interrupt control register	$00007A_{16}$
Timer B1 interrupt control register	$00007B_{16}$
Timer B2 interrupt control register	$00007C_{16}$
$\overline{INT_0}$ interrupt control register	$00007D_{16}$
$\overline{INT_1}$ interrupt control register	$00007E_{16}$
$\overline{INT_2}$ interrupt control register	$00007F_{16}$

Interrupts caused by a BRK instruction and when dividing by zero are software interrupts and are not included in this list.

Other interrupts previously mentioned are DMA, A-D converter, UART, Timer, $\overline{INT}$ interrupts. The priority of these interrupts can be changed by changing the priority level in the corresponding interrupt control register by software.

Figure 8 shows a diagram of the interrupt priority resolution circuit. When an interrupt is caused, the each interrupt device compares its own priority with the priority from above and if its own priority is higher, then it sends the priority below and requests the interrupt. If the priorities are the same, the one above has priority.

This comparison is repeated to select the interrupt with the highest priority among the interrupts that are being requested. Finally the selected interrupt is compared with the processor interrupt priority level (IPL) contained in the processor status register (PS) and the request is accepted if it is higher than IPL and the interrupt disable flag I is "0". The request is not accepted if flag I is "1". The reset, $\overline{DBC}$, and watchdog timer interrupts are not affected by the interrupt disable flag I.

When an interrupt is accepted, the contents of the processor status register (PS) is saved to the stack and the interrupt disable flag I is set to "1".

Furthermore, the interrupt request bit of the accepted interrupt is cleared to "0" and the processor interrupt priority level (IPL) in the processor status register (PS) is replaced by the priority level of the accepted interrupt.

Therefore, multi-level priority interrupts are possible by resetting the interrupt disable flag I to "0" and enable further interrupts.

For reset, $\overline{DBC}$, watchdog timer, zero divide, and BRK instruction interrupts, which do not have an interrupt control register, the processor interrupt level (IPL) is set as shown in Table 3.

Priority resolution is performed by latching the interrupt request bit and interrupt priority level so that they do not change. They are sampled at the first half and latched at the last half of the operation code fetch cycle.

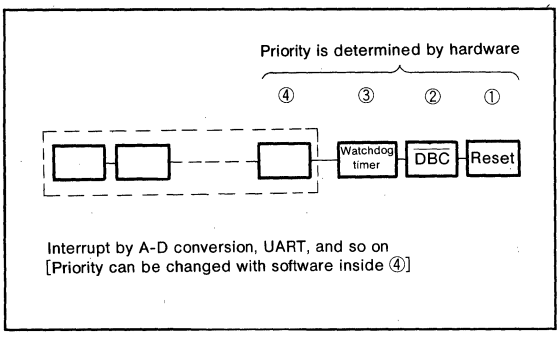


Fig.7 Interrupt priority

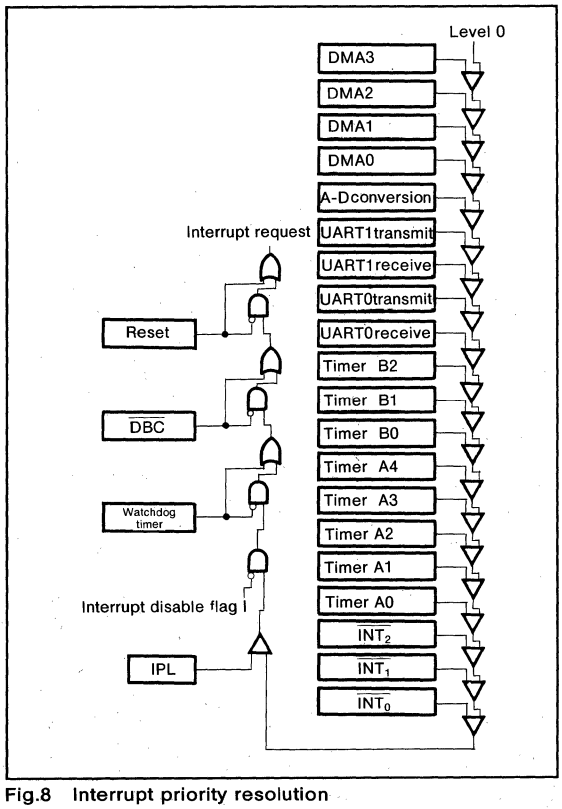


Fig.8 Interrupt priority resolution

Because priority resolution takes some time, no sampling pulse is generated for a certain interval even if it is the next operation code fetch cycle.

As shown in Figure 9, there are three different interrupt priority resolution time from which one is selected by software. After the selected time has elapsed, the highest priority is determined and is processed after the currently executing instruction has been completed.
The time is selected with bits 4 and 5 of the processor mode register (address $5E_{16}$) shown in Figure 10. Table 4 shows the relationship between these bits and the number of cycles. After a reset, the processor mode register is initialized to "00_{16}" and therefore, the longest time is selected.
However, the shortest time may be selected by software.

Table 3. Value set in processor interrupt level(IPL)during an interrupt

Interrupt types	Setting value
Reset	0
$\overline{DBC}$	7
Watchdog timer	7
Zero divide	Not change value of IPL.
BRK instruction	Not change value of IPL.

Table 4. Relationship between priority level evaluation time selection bit and number of cycles

Priority level resolution time selection bit		Number of cycles
Bit 5	Bit 4	
0	0	7 cycles of ϕ
0	1	4 cycles of ϕ
1	0	2 cycles of ϕ

ϕ : Internal clock

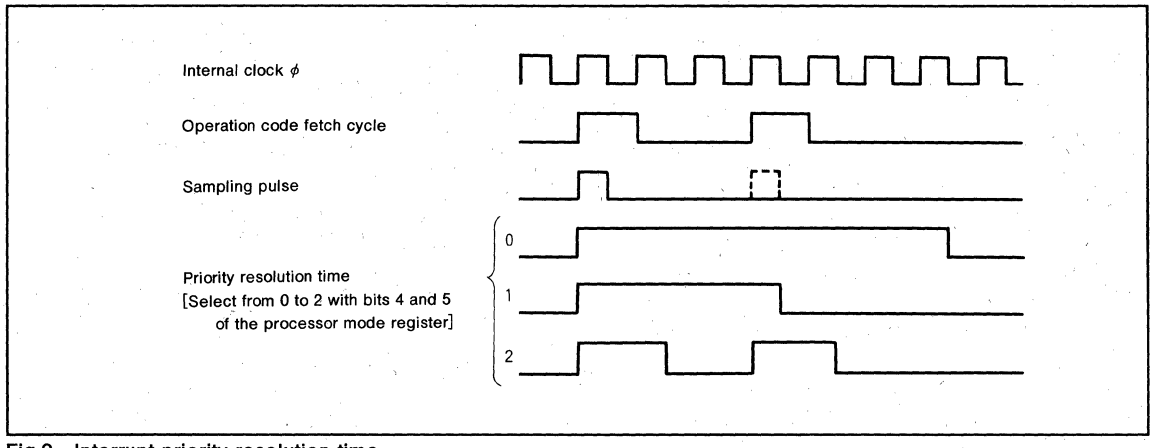


Fig.9 Interrupt priority resolution time

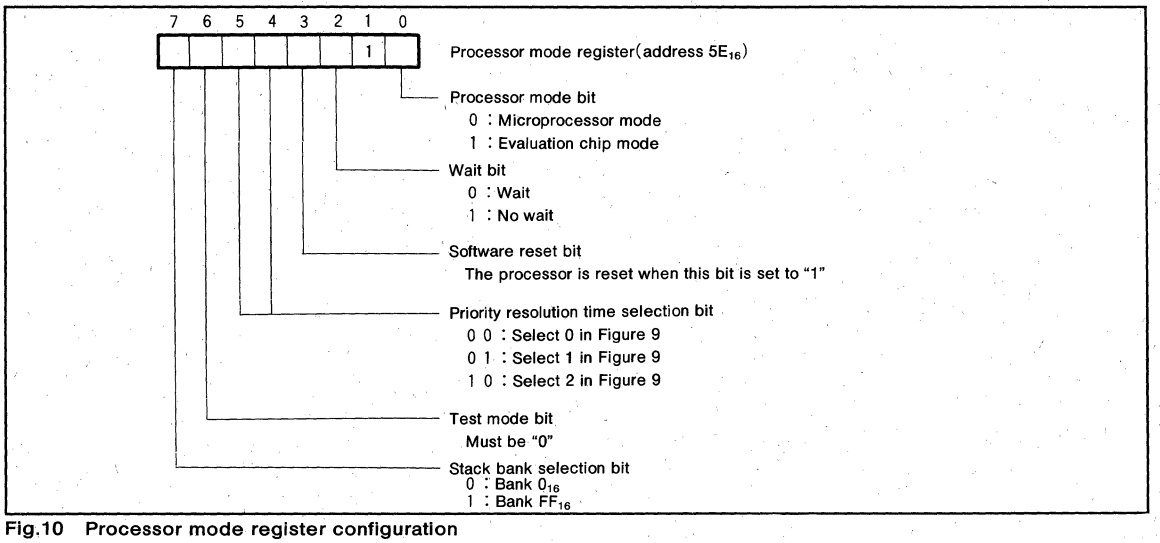


Fig.10 Processor mode register configuration

TIMER

There are eight 16-bit timers. They are divided by type into timer A(5) and timer B(3).

The timer I/O pins are shared with I/O pins for port P5. To use these pins as timer input pins, the direction register bit corresponding to the pin must be cleared to "0" to specify input mode.

TIMER A

Figure 11 shows a block diagram of timer A.

Timer A has four modes; timer mode, event counter mode, one-shot pulse mode, and pulse width modulation mode. The mode is selected with bits 0 and 1 of the timer Ai mode register (i=0 to 4).

Each of these modes is described below.

(1) Timer mode [00]

Figure 12 shows the bit configuration of the timer Ai mode register during timer mode. Bits 0, 1, and 5 of the timer Ai mode register must always be "0" in timer mode.

Bit 3 is ignored if bit 4 is "0".

Bits 6 and 7 are used to select the timer counter source.

The counting of the selected clock starts when the count start flag is "1" and stops when it is "0".

Figure 13 shows the bit configuration of the count start flag.

The counter is decremented, an interrupt is caused and the interrupt request bit in the timer Ai interrupt control register is set when the contents becomes 0000_{16}. At the same time, the contents of the reload register is transferred to the counter and count is continued.

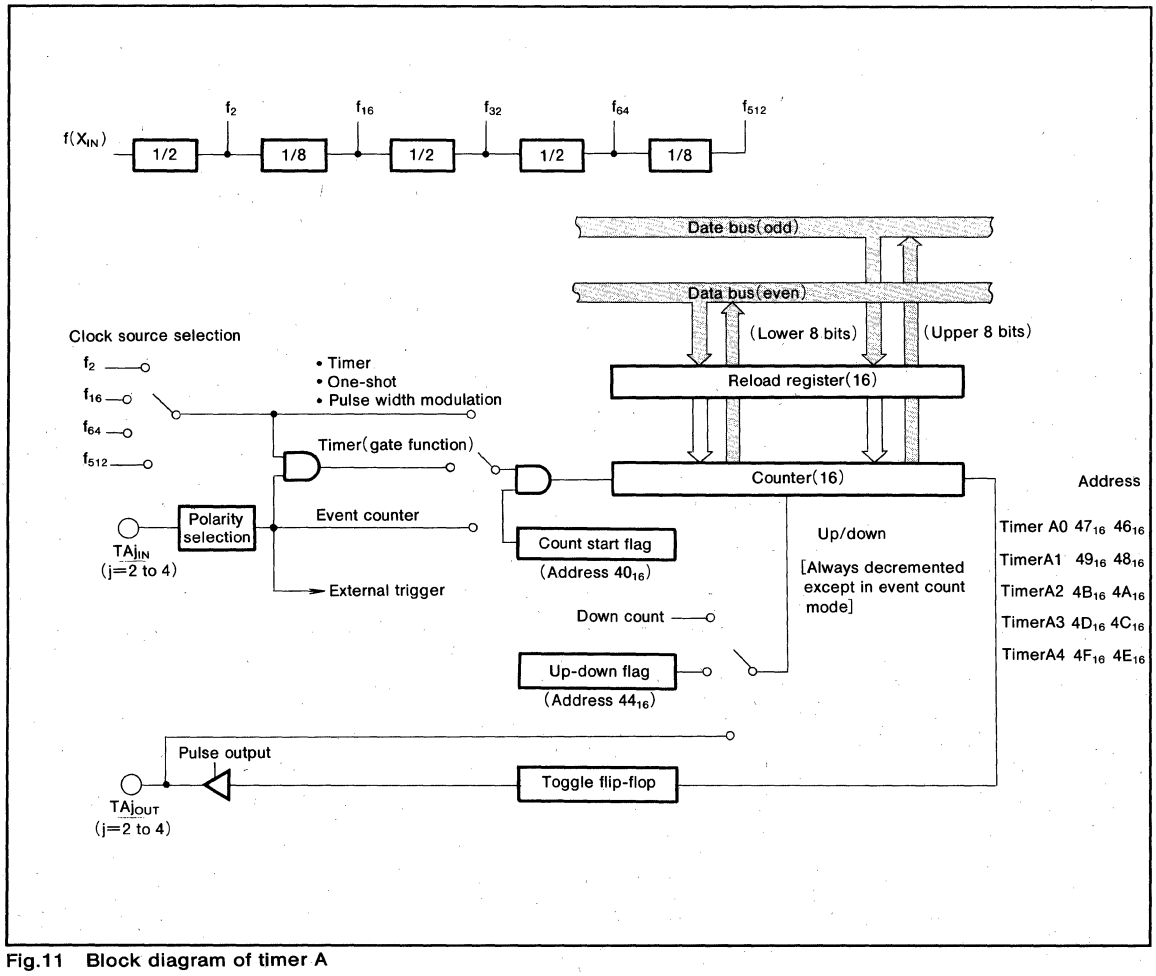


Fig.11 Block diagram of timer A

When bit 2 of the timer Ai mode register is "1", the output is generated from TAj_{OUT}(j=2 to 4) pin. The output is toggled each time the contents of the counter reaches to 0000_{16}. When the contents of the count start flag is "0","L" is output from TAj_{OUT} pin.
When bit 2 is "0", TAj_{OUT} can be used as a normal port pin.
When bit 4 is "0", TAj_{IN} can be used as a normal port pin.
When bit 4 is "1", counting is performed only while the input signal from the TAj_{IN} pin is "H" or "L" as shown in Figure 14. Therefore, this can be used to measure the pulse width of the TAj_{IN} input signal. Whether to count while the input signal is "H" or while it is "L" is determined by bit 3. If bit 3 is "1", counting is performed while the TAj_{IN} pin input signal is "H" and if bit 3 is "0", counting is performed while it is "L".

Note that the duration of "H" or "L" on the TAj_{IN} pin must be two or more cycles of the timer count source.
When data is written into timer Ai while it is not operating, the data is written into the reload register and counter. On the other hand, when data is written into timer Ai while it is operating, the data is written into the reload register only and not into the counter. When reloading is initiated next, new data is reloaded from the reload register into the counter for operation continuation. The contents of the counter can be read at any time.
When the value set in the timer Ai register is n, the timer frequency dividing ratio is 1/(n+1).
As timers A0 and A1 do not have the TA_{IN} and TA_{OUT} pin, be sure to use the timer mode.

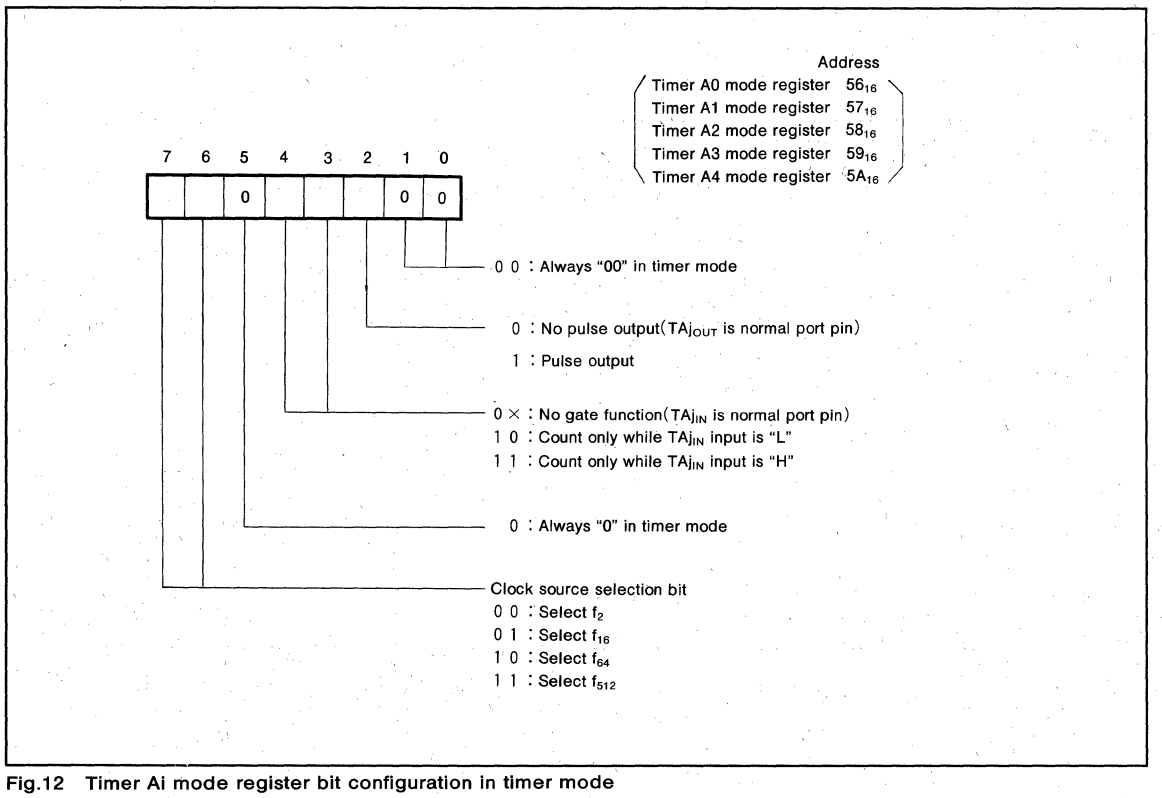


Fig.12 Timer Ai mode register bit configuration in timer mode

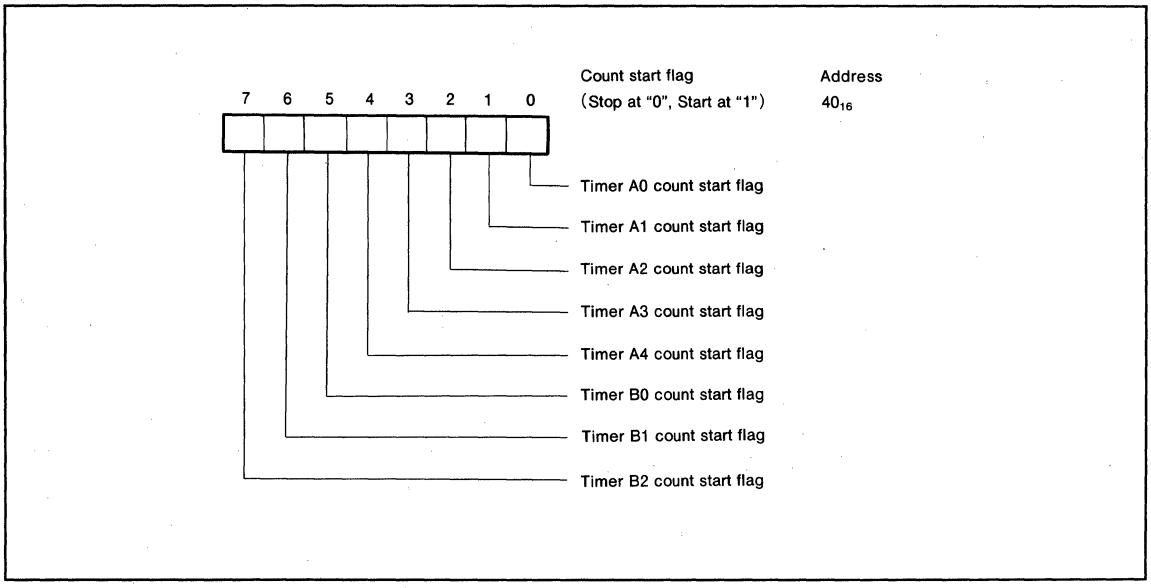


Fig.13 Count start flag bit configuration

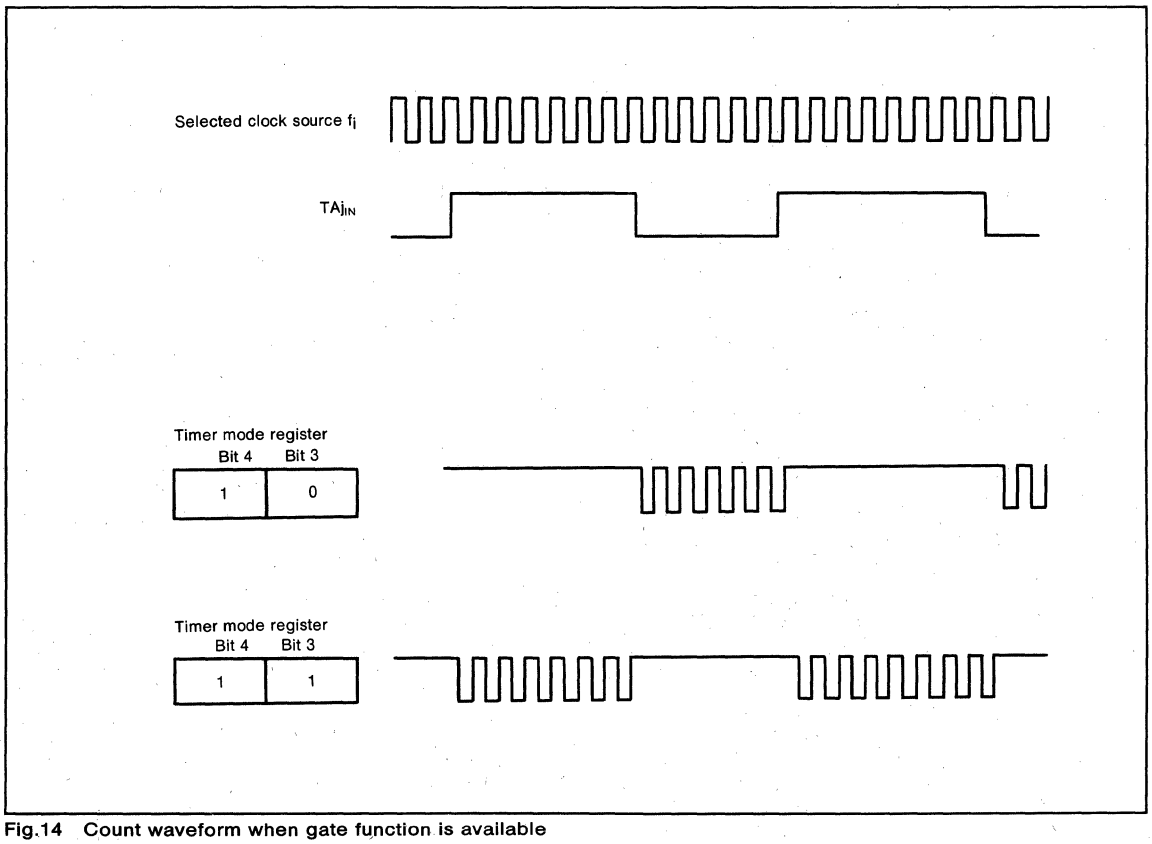


Fig.14 Count waveform when gate function is available

(2) Event counter mode [01]

Figure 15 shows the bit configuration of the timer Aj mode register during event counter mode. In event counter mode, the bit 0 of the timer Aj(j=2 to 4) mode register must be "1" and bit 1 and bit 5 must be "0".

The input signal from the TAj_{IN} pin is counted when the count start flag shown in Figure 13 is "1" and counting is stopped when it is "0".

Count is performed at the fall of the input signal when bit 3 is "0" and at the rise of the signal when it is "1".

In event counter mode, whether to increment or decrement the count can be selected with the up-down flag or the input signal from the TAj_{OUT} pin. When bit 4 of the timer Aj mode register is "0", the up-down flag is used to determine whether to increment or decrement the count (decrement when the flag is "0" and increment when it is "1"). Figure 16 shows the bit configuration of the up-down flag. When bit 4 of the timer Aj mode register is "1", the input signal from the TAj_{OUT} pin is used to determine whether to increment or decrement the count. However, note that bit 2 must be "0" if bit 4 is "1" because if bit 2 is "1", TAj_{OUT} pin becomes an output pin with pulse output.

Determine the level of the input signal from the TAj_{OUT} pin before valid edge is input to the TAj_{IN} pin.

The count is decremented when the input signal from the TAj_{OUT} pin is "L" and incremented when it is "H".

An Interrupt request signal is generated and the interrupt request bit in the timer A_j interrupt control register is set when the counter reaches 0000_{16} (decrement count) or $FFFF_{16}$ (increment count). At the same time, the contents of the reload register is transferred to the counter and the count is continued.

When bit 2 is "1" and the counter reaches 0000_{16} (decrement count) or $FFFF_{16}$ (increment count), the waveform reversing polarity is output from TAj_{OUT} pin.

If bit 2 is "0", TAj_{OUT} pin can be used as a normal port pin. However, if bit 4 is "1" and the TAj_{OUT} pin is used as an output pin, the output from the pin changes the count direction. Therefore, bit 4 should be "0" unless the output from the TAj_{OUT} pin is to be used to select the count direction.

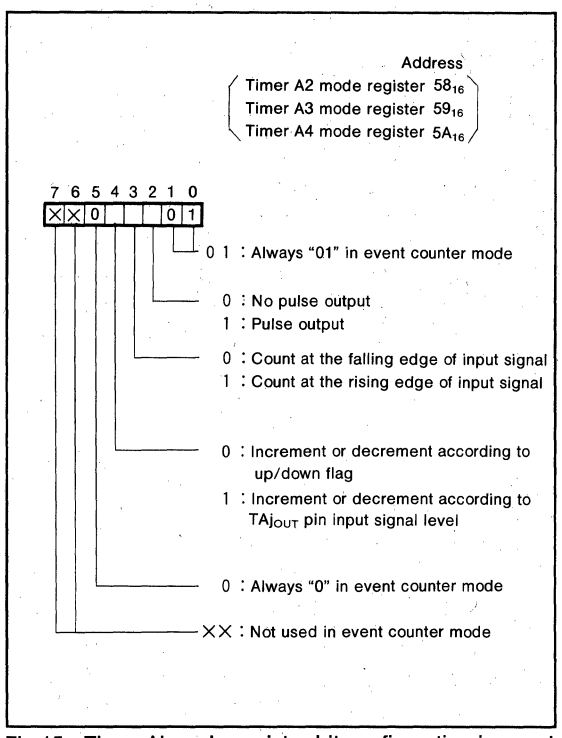


Fig.15 Timer Aj mode register bit configuration in event counter mode

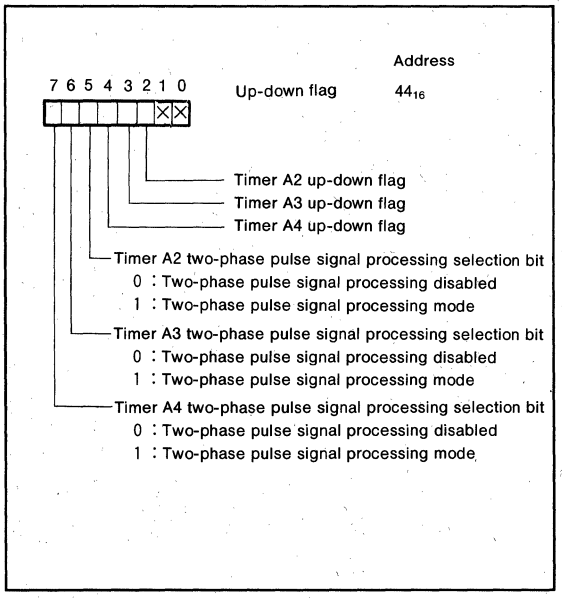


Fig.16 Up-down flag bit configuration

Data write and data read are performed in the same way as for timer mode. That is, when data is written to timer Aj while it is not operating, it is also written to the reload register and the counter. When data is written to timer Aj while it is operating, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time. The counter can be read at any time.

In event counter mode, whether to increment or decrement the counter can also be determined by supplying two-phase pulse input with phase shifted by 90° to timer A2, A3, or A4. There are two types of two-phase pulse processing operations. One uses timers A2 and A3, and the other uses timer A4. In either processing operation, two-phase pulse is input in the same way, that is, pulses out of phase by 90° are input at the TAj_{OUT} (j=2 to 4) pin and TAj_{IN} pin.

When timers A2 and A3 are used, as shown in Figure 17, the count is incremented when a rising edge is entered to the TAk_{IN} pin after the level of TAk_{OUT} (k = 2, 3) pin changes from "L" to "H", and when the falling edge is inserted, the count is decremented.

For timer A4, as shown in Figure 18, when a phase related pulse with a rising edge input to the $TA4_{IN}$ pin is entered after the level of $TA4_{OUT}$ pin changes from "L" to "H", the count is incremented at the respective rising edge and falling edge of the $TA4_{OUT}$ and $TA4_{IN}$ pins.

When a phase related pulse with a falling edge input to the $TA4_{OUT}$ pin is entered after the level of $TA4_{IN}$ pin changes from "H" to "L", the count is decremented at the respective rising edge and falling edge of the $TA4_{IN}$ and $TA4_{OUT}$ pins.

When performing this two-phase pulse signal processing, timer Aj mode register bit 0 and bit 4 must be set to "1" and bits 1, 2, 3, and 5 must be "0" (refer the Figure 19). Bits 6 and 7 are ignored. Note that bits 5, 6, and 7 of the up-down flag register (44_{16}) are the two-phase pulse signal processing selection bit for timer A2, A3, and A4 respectively. Each timer operates in normal event counter mode when the corresponding bit is "0" and performs two-phase pulse signal processing when it is "1".

Count is started by setting the count start flag to "1". Data write and read are performed in the same way as for normal event counter mode. Note that the direction register of the input port must be set to input mode because two-phase pulse signal is input. Also, there can be no pulse output in this mode.

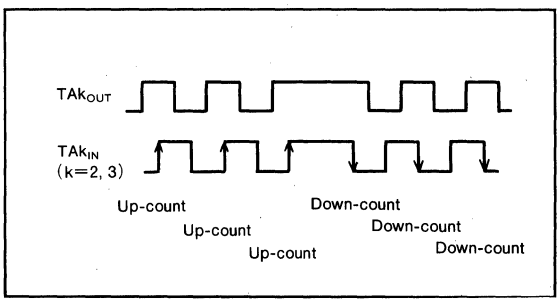


Fig.17 Two-phase pulse processing operation of timers A2 and A3

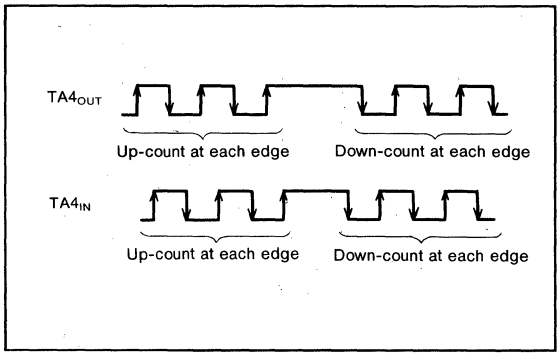


Fig.18 Two-phase pulse processing operation of timer A4

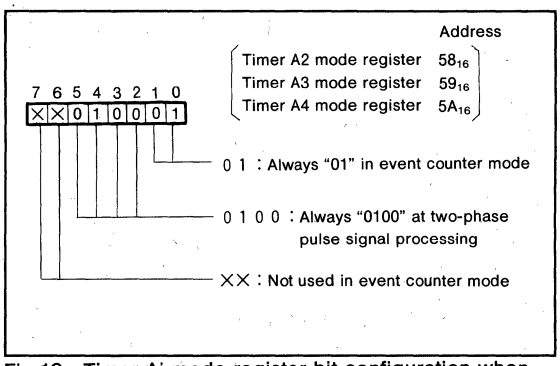


Fig.19 Timer Aj mode register bit configuration when performing two-phase pulse signal processing in event counter mode

(3) One-shot pulse mode [10]

Figure 20 shows the bit configuration of the timer Aj mode register during one-shot pulse mode. In one-shot pulse mode, bit 0 and bit 5 must be "0" and bit 1 and bit 2 must be "1".

The trigger is enabled when the count start flag is "1". The trigger can be generated by software or it can be input from the TAj_{IN} pin. Software trigger is selected when bit 4 is "0" and the input signal from the TAj_{IN} pin is used as the trigger when it is "1". Bit 3 is used to determine whether to trigger at the fall of the trigger signal or at the rise. The trigger is at the fall of the trigger signal when bit 3 is "0" and at the rise of the trigger signal when it is "1". Software trigger is generated by setting the bit in the one-shot start flag corresponding to each timer. Figure 21 shows the bit configuration of the one-shot start flag. Bit 7 of the one-shot start flag must always be "0".

As shown in Figure 22, when a trigger signal is received, the counter counts the clock selected by bits 6 and 7.

If the contents of the counter is not 0000_{16}, the TAj_{OUT} pin goes "H" when a trigger signal is received. The count direction is decrement.

When the counter reaches 0001_{16}. The TAj_{OUT} pin goes "L" and count is stopped. The contents of the reload register is transferred to the counter. At the same time, an interrupt request signal is generated and the interrupt request bit in the timer Aj interrupt control register is set. This is repeated each time a trigger signal is received. The output pulse width is

$$\left(\frac{1}{\text{pulse frequency of the selected clock}}\right) \times (\text{counter's value at the time of trigger}).$$

If the count start flag is "0", TAj_{OUT} goes "L". Therefore, the value corresponding to the desired pulse width must be written to timer Aj before setting the timer Aj count start flag.

As shown in Figure 23, a trigger signal can be received before the operation for the previous trigger signal is completed. In this case, the contents of the reload register is transferred to the counter by the trigger and then that value is decremented. Except when retriggering while operating, the contents of the reload register is not transferred to the counter by triggering.

When retriggering, there must be at least one timer count source cycle before a new trigger can be issued.

Data write is performed to the same way as for timer mode.

When data is written in timer Aj while it is not operating, it is also written to the reload register and the counter.

When data is written to timer Aj while it is operating, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time.

Undefined data is read, when timer Aj is read.

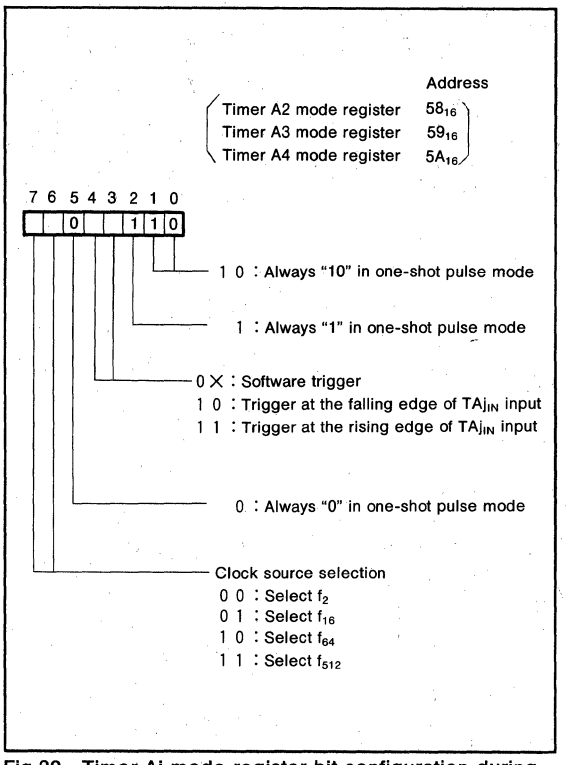


Fig.20 Timer Aj mode register bit configuration during one-shot pulse mode

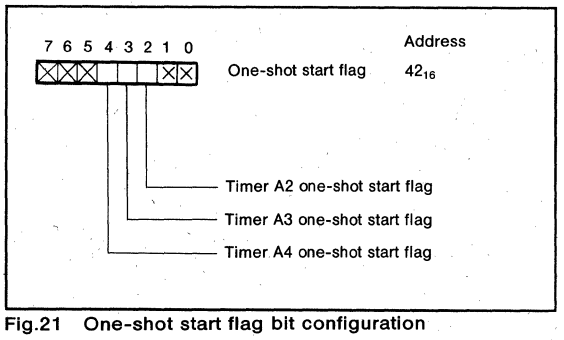


Fig.21 One-shot start flag bit configuration

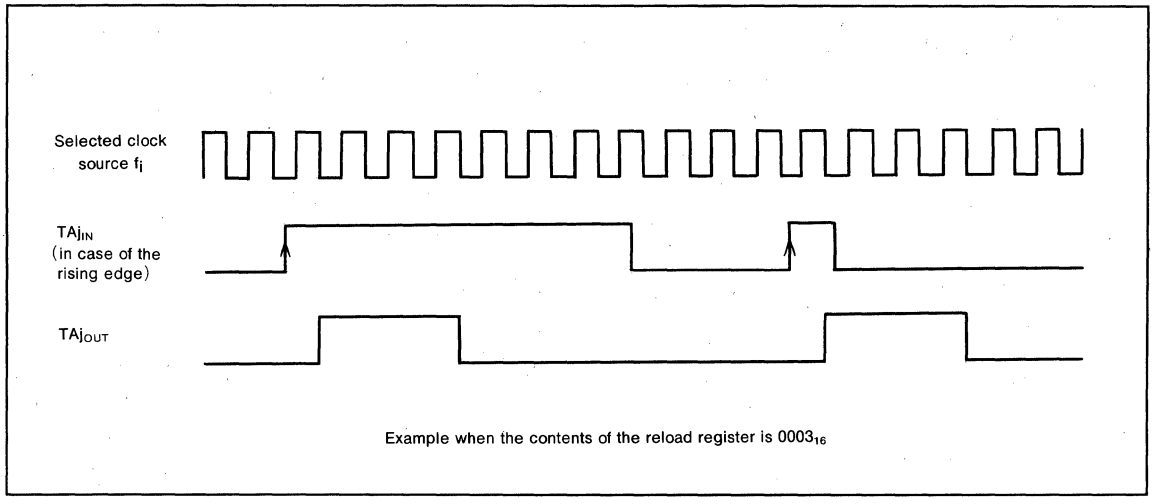


Fig.22 Pulse output example when external rising edge is selected

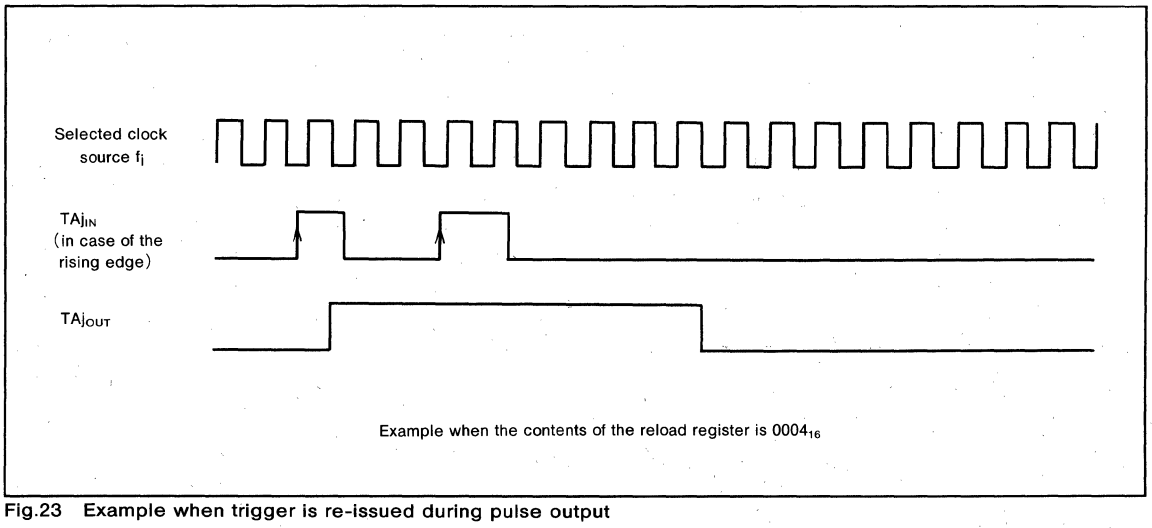


Fig.23 Example when trigger is re-issued during pulse output

(4) Pulse width modulation mode [11]

Figure 24 shows the bit configuration of the timer Aj mode register during pulse width modulation mode. In pulse width modulation mode, bits 0, 1, and 2 must be set to "1".

Bit 5 is used to determine whether to perform 16-bit length pulse width modulator or 8-bit length pulse width modulator. 16-bit length pulse width modulator is performed when bit 5 is "0" and 8-bit length pulse width modulator is performed when it is "1". The 16-bit length pulse width modulator is described first.

The pulse width modulator can be started with a software trigger or with an input signal from a TAj_{IN} pin (external trigger).

The software trigger mode is selected when bit 4 is "0".

Pulse width modulator is started and pulse is output from TAj_{OUT} when the timer Aj start flag is set to "1".

The external trigger mode is selected when bit 4 is "1".

Pulse width modulator starts when a trigger signal is input from the TAj_{IN} pin when the timer Aj start flag is "1".

Whether to trigger at the fall or rise of the trigger signal is determined by bit 3. The trigger is at the fall of the trigger signal when bit 3 is "0" and at the rise when it is "1".

When data is written to timer Aj with the pulse width modulator while it is not operating, it is written to the reload register and the counter.

Then when the timer Aj start flag is set to "1" and a software trigger or an external trigger is issued to start modulation, the waveform shown in Figure 25 is output continuously. Once modulation is started, triggers are not accepted. If the value in the reload register is m, the duration "H" of pulse is

$$\frac{1}{\text{selected clock frequency}} \times m.$$

and the output pulse period is

$$\frac{1}{\text{selected clock frequency}} \times (2^{16}-1).$$

An interrupt request signal is generated and the interrupt request bit in the timer Aj interrupt control register is set at each fall of the output pulse.

The width of the output pulse is changed by updating timer data. The update can be performed at any time. The output pulse width is changed at the rise of the pulse after data is written to the timer. The data is written to the reload register, but not to the counter.

The contents of the reload register are transferred to the counter just before the rise of the next pulse so that the pulse width is changed from the next output pulse.

Undefined data is read, when timer Aj is read.

The 8-bit length pulse width modulator is described next.

The 8-bit length pulse width modulator is selected when the timer Aj mode register bit 5 is "1".

The reload register and the counter are both divided into 8-bit halves.

The low order 8 bits function as a prescaler and the high order 8 bits function as the 8-bit length pulse width modulator. The prescaler counts the clock selected by bits 6 and 7. A pulse is generated when the counter reaches 0000_{16} as shown in Figure 26. At the same time, the contents of the reload register is transferred to the counter and count is continued.

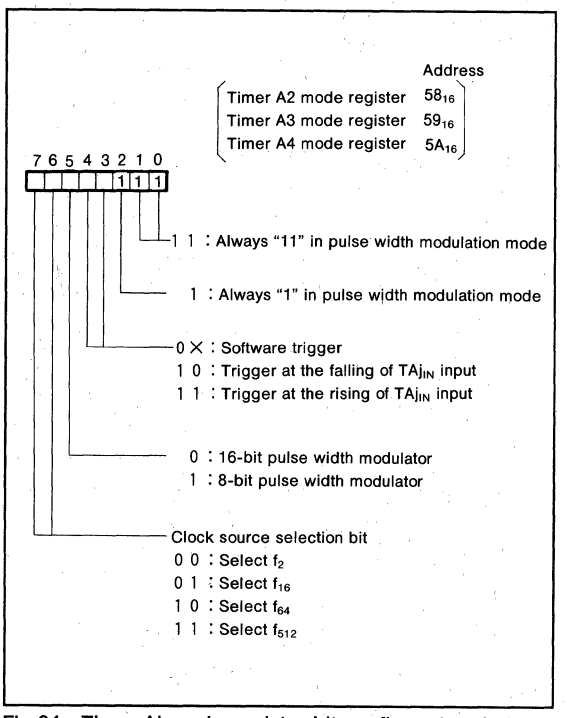


Fig.24 Timer Aj mode register bit configuration during pulse width modulation mode

Therefore, if the low order 8-bit of the reload register is n, the period of the generated pulse is

$$\frac{1}{\text{selected clock frequency}} \times (n+1).$$

The high order 8-bit function as an 8-bit length pulse width modulator using this pulse as input. The operation is the same as for 16-bit length pulse width modulator except that the length is 8 bits. If the high order 8-bit of the reload register is m, the duration "H" of pulse is

$$\frac{1}{\text{selected clock frequency}} \times (n+1) \times m.$$

And the output pulse period is

$$\frac{1}{\text{selected clock frequency}} \times (n+1) \times (2^8-1).$$

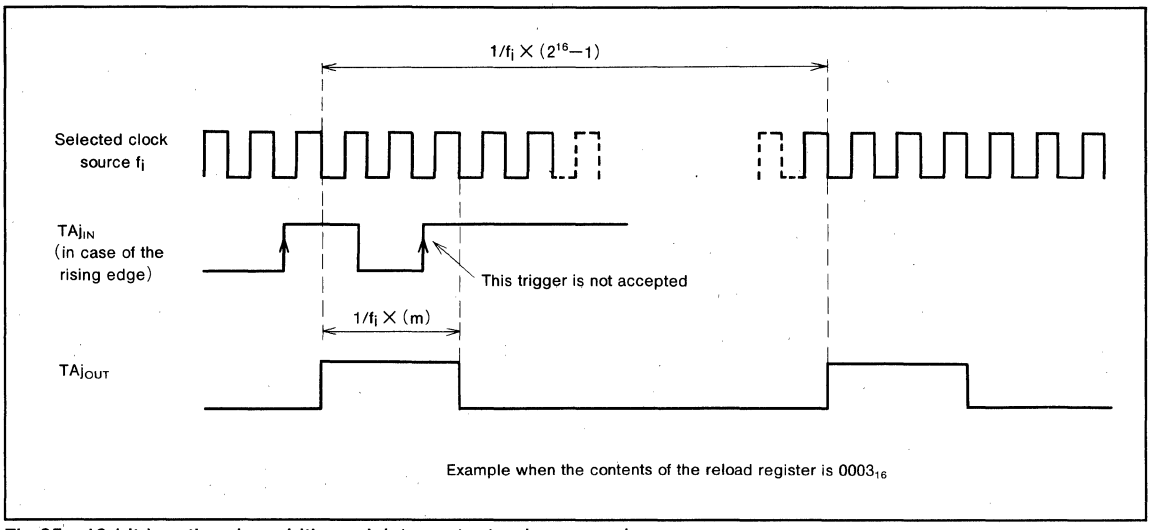


Fig.25 16-bit length pulse width modulator output pulse example

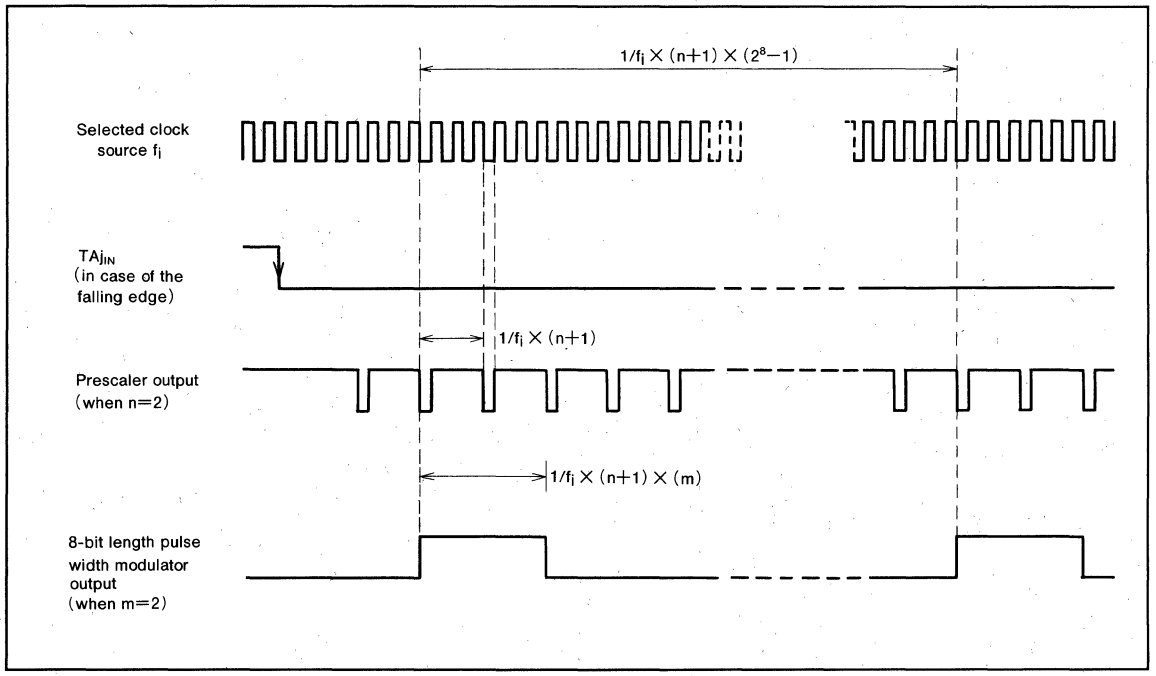


Fig.26 8-bit length pulse width modulator output pulse example

TIMER B

Figure 27 shows a block diagram of timer B.

Timer B has three modes; timer mode, event counter mode, and pulse period measurement/pulse width measurement mode. The mode is selected with bits 0 and 1 of the timer Bi mode register (i = 0 to 2). Each of these modes is described below.

(1) Timer mode [00]

Figure 28 shows the bit configuration of the timer Bi mode register during timer mode. Bits 0, and 1 of the timer Bi mode register must always be "0" in timer mode.

Bits 6 and 7 are used to select the clock source. The counting of the selected clock starts when the count start flag "1" and stops when "0".

As shown in Figure 13, the timer Bi count start flag is at the same address as the timer Ai count start flag. The count is decremented, an interrupt occurs, and the interrupt request bit in the timer Bi interrupt control register is set when the contents becomes 0000_{16}. At the same time, the contents of the reload register is stored in the counter and count is continued.

Timer Bi does not have a pulse output function or a gate function like timer A.

When data is written to timer Bi while it is not operating, it is written to the reload register and the counter. When data is written to timer Bi while it is operating, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time.

The contents of the counter can be read at any time. As timer B2 does not have the TB_{IN} pin, be sure to use the timer mode.

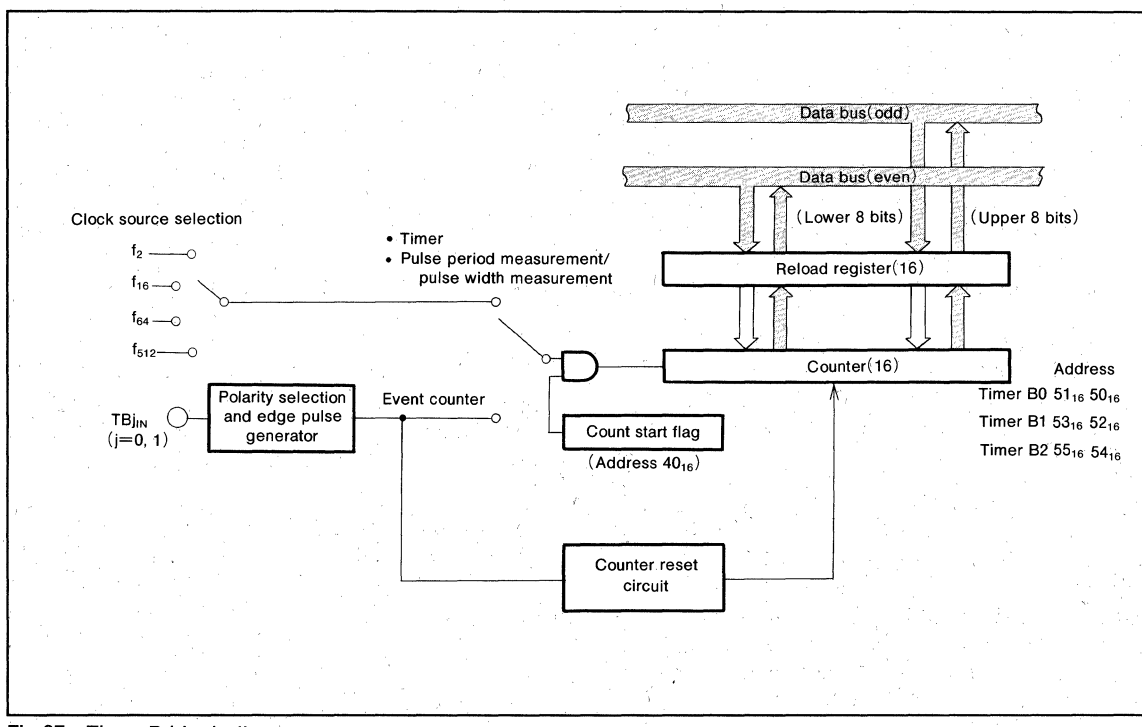


Fig.27 Timer B block diagram

MITSUBISHI ELECTRIC

(2) Event counter mode [01]

Figure 29 shows the bit configuration of the timer Bj(j=0, 1) mode register during event counter mode. In event counter mode, the bit 0 in the tiimer Bj mode register must be "1" and bit 1 must be "0". The input signal from the TBj_{IN} pin is counted when the count start flag is "1" and counting is stopped when it is "0".

Count is performed at the fall of the input signal when bits 2 and 3 are "0", and at the rise of the input signal when bit 3 is "0" and bit 2 is "1".

When bit 3 is "1" and bit 2 is "0", count is performed at the rise and fall of the input signal.

Data write, data read and timer interrupt are performed in the same way as for timer mode.

(3) Pulse period measurement/pulse width measurement mode [10]

Figure 30 shows the bit configuration of the timer Bj mode register during pulse period measurement/pulse width measurement mode.

In pulse period measurement/pulse width measurement mode, bit 0 must be "0" and bit 1 must be "1". Bits 6 and 7 are used to select the clock source. The selected clock is counted when the count start flag is "1" and counting stops when it is "0".

The pulse period measurement mode is selected when bit 3 is "0". In pulse period measurement mode, the selected clock is counted during the interval starting at the fall of the input signal from the TBj_{IN} pin to the next fall or at the rise of the input signal to the next rise and the result is stored in the reload register. In this case, the reload register acts as a buffer register.

When bit 2 is "0", the clock is counted from the fall of the input signal to the next fall. When bit 2 is "1", the clock is counted from the rise of the input signal to the next rise.

In the case of counting from the fall of the input signal to the next fall, counting is performed as follows. As shown in Figure 31, when the fall of the input signal from TBj_{IN} pin is detected, the contents of the counter is transferred to the reload register. Next the counter is cleared and count is started from the next clock. When the fall of the next input signal is detected, the contents of the counter is transferred to the reload register once more, the counter is cleared, and the count is started. The period from the fall of the input signal to the next fall is measured in this way.

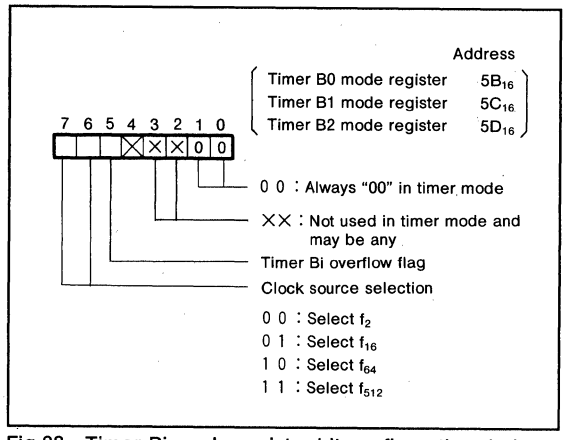


Fig.28 Timer Bi mode register bit configuration during timer mode

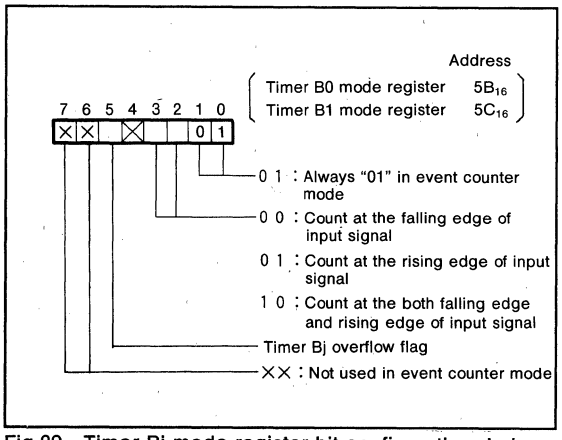


Fig.29 Timer Bj mode register bit configuration during event counter mode

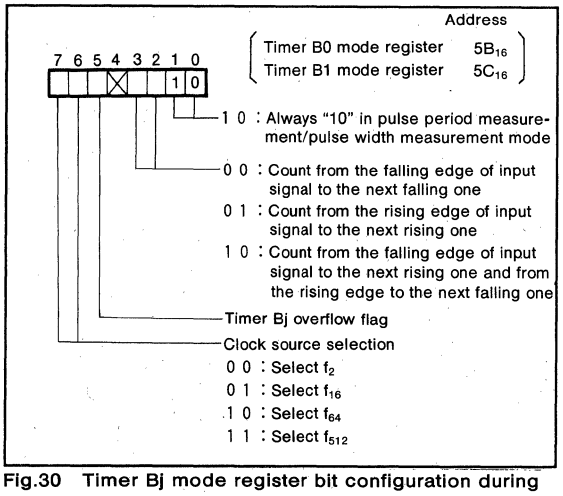


Fig.30 Timer Bj mode register bit configuration during pulse period measurement/pulse width measurement mode

After the contents of the counter is transferred to the reload register, an interrupt request signal is generated and the interrupt request bit in the timer Bj interrupt control register is set. However, no interrupt request signal is generated when the contents of the counter is transferred first time to the reload register after the count start flag is set to "1".

When bit 3 is "1", the pulse width measurement mode is selected. Pulse width measurement mode is similar to pulse period measurement mode except that clock is counted from the fall of the TBj_{IN} pin input signal to the next rise or from the rise of the input signal to the next fall as shown in Figure 32.

When timer Bj is read, the contents of the reload register is read.

Note that in this mode, the interval between the fall of the TBj_{IN} pin input signal to the next rise or from the rise to the next fall must be at least two cycles of the timer count source.

Timer Bi overflow flag which is bit 5 of timer Bi mode register is set to "1" when the timer Bi counter reach 0000_{16}.

This flag is cleared by writing to corresponding timer Bi mode register.

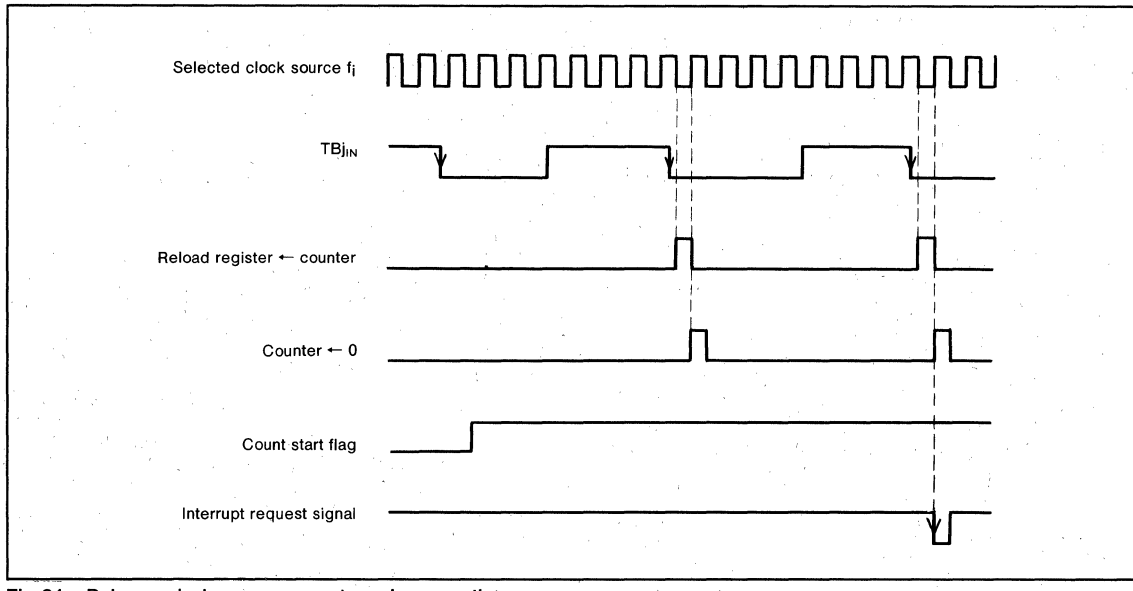


Fig.31 Pulse period measurement mode operation (example of measuring the interval between the falling edge to next falling one)

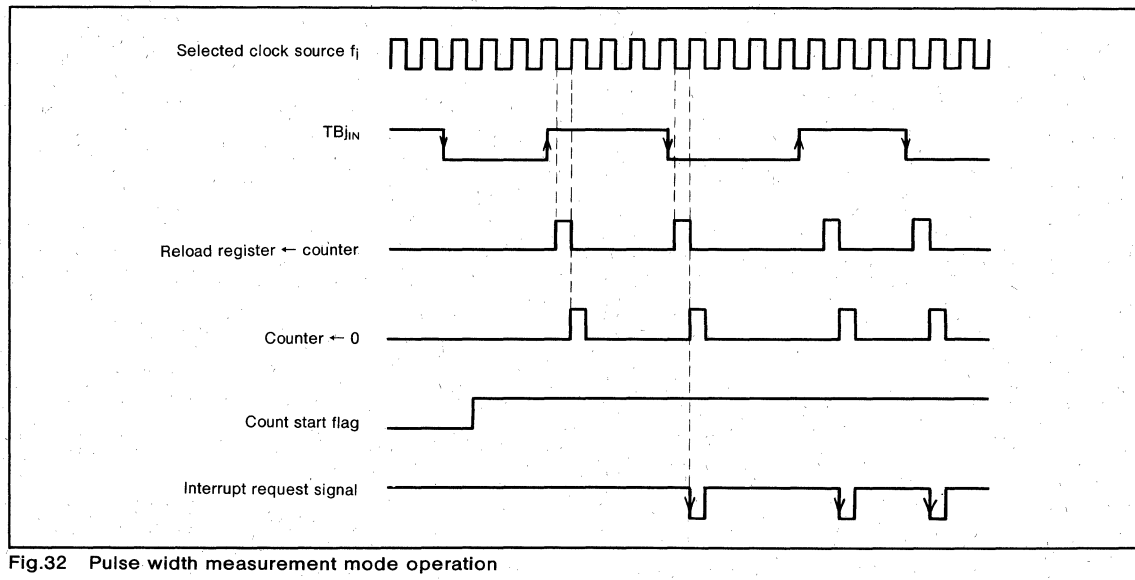


Fig.32 Pulse width measurement mode operation

SERIAL I/O PORTS

Two independent serial I/O ports are provided.

Figure 33 shows a block diagram of the serial I/O ports.

Bits 0, 1. and 2 of the UARTi (i = 0, 1) Transmit/Receive mode register shown in Figure 34 are used to determine whether to use port P8 as parallel port, clock synchronous serial I/O port, or asynchronous (UART) serial I/O port using start and stop bits.

Figures 35 and 36 show connections of receiver/transmitter according to the mode.

Figure 37 shows the bit configuration of the UARTi transmit/receive control register.

Each communication method is described below.

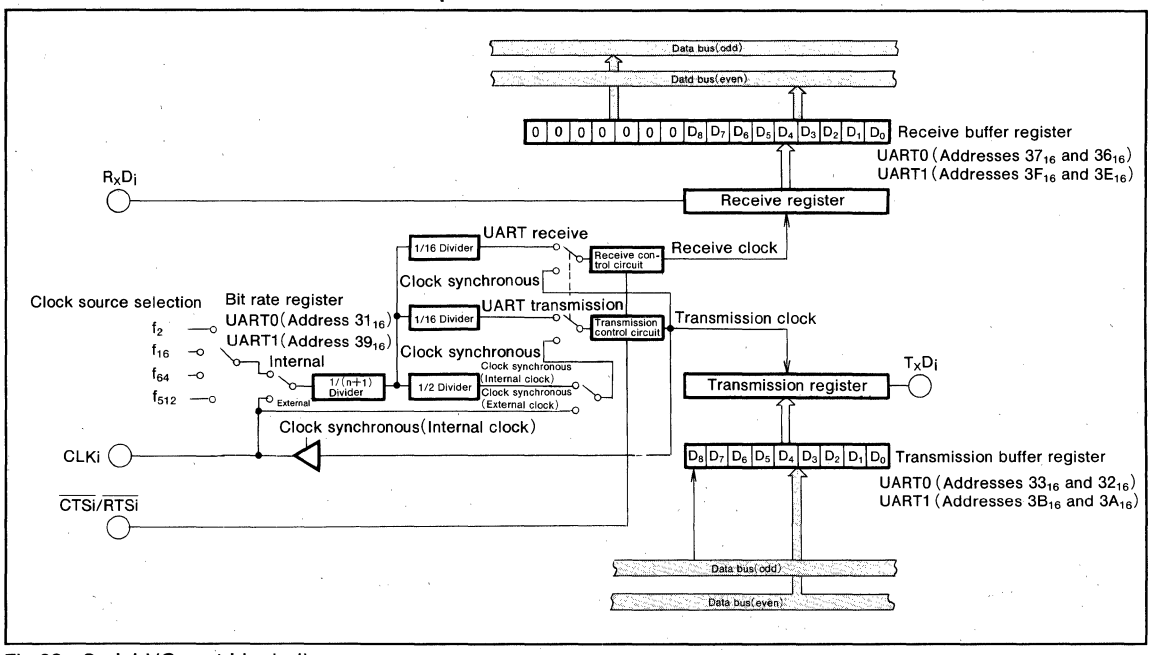


Fig.33 Serial I/O port block diagram

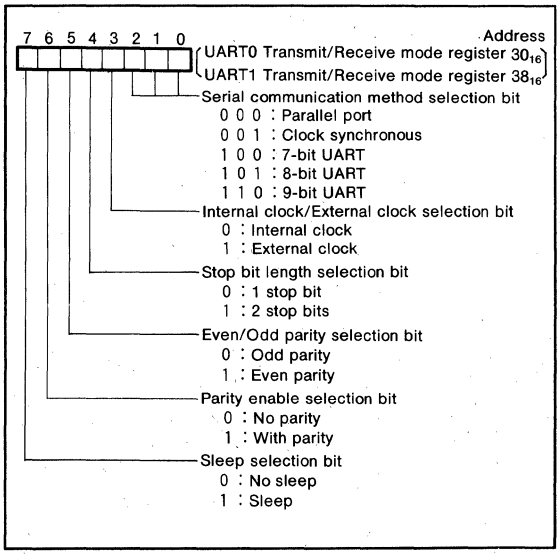


Fig.34 UARTi Transmit/Receive mode register bit configuration

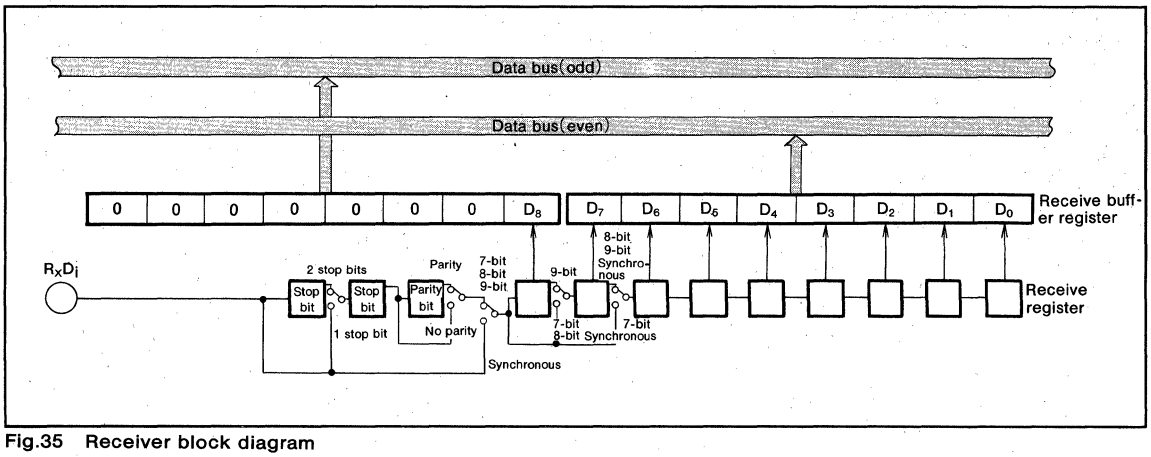


Fig.35 Receiver block diagram

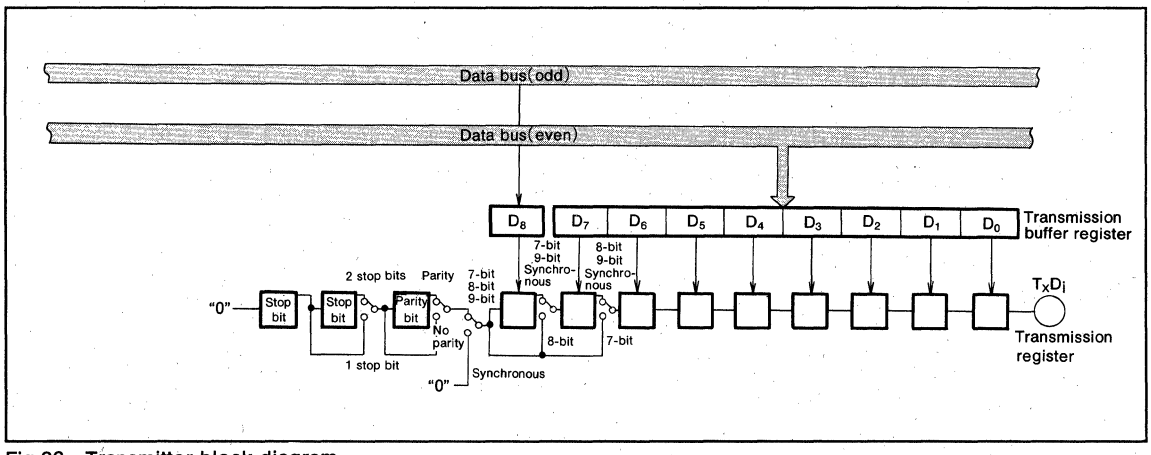


Fig.36 Transmitter block diagram

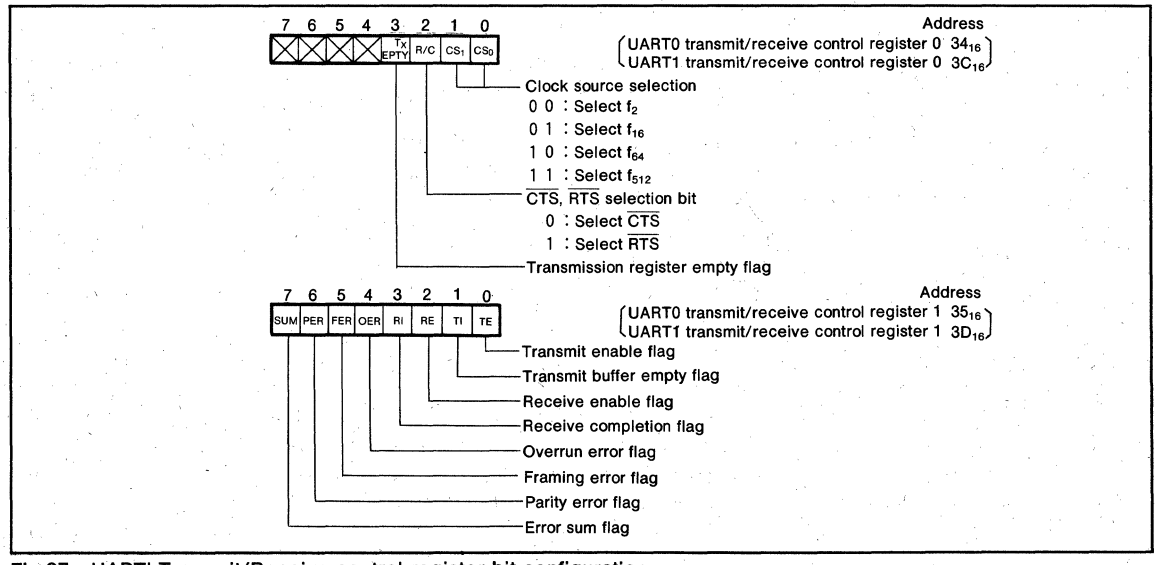


Fig.37 UARTi Transmit/Receive control register bit configuration

CLOCK SYNCHRONOUS SERIAL COMMUNICATION

A case where communication is performed between two clock synchronous serial I/O ports as shows in Figure 38 will be described. (The transmission side will be denoted by subscript j and the receiving side will be denoted by subscript k.)

Bit 0 of the $UART_j$ transmit/receive mode register and $UART_k$ transmit/receive mode register must be set to "1" and bits 1 and 2 must be "0". The length of the transmission data is fixed at 8 bits.

Bit 3 of the $UART_j$ transmit/receive mode register of the clock sending side is cleared to "0" to select the internal clock. Bit 3 of the $UART_k$ transmit/receive mode register of the clock receiving side is set to "1" to select the external clock. Bits 4, 5 and 6 are ignored in clock synchronous mode. Bit 7 must always be "0".

The clock source is selected by bit 0 (CS_0) and bit 1 (CS_1) of the clock sending side $UART_j$ transmit/receive control register 0. As shown in Figure 33, the selected clock is divided by $(n+1)$, then by 2, passed through a transmission control circuit, and output as transmission clock CLK_j. Therefore, when the selected clock is fi,

$$\text{Bit Rate} = fi / \{(n+1) \times 2\}$$

On the clock receiving side, the CS_0 and CS_1 bits of the $UART_k$ transmit/receive control register are ignored because an external clock is selected.

The bit 2 of the clock sending side $UART_j$ transmit/receive control register is clear to "0" to select $\overline{CTS_j}$ input. The bit 2 of the clock receiving side is set to "1" to select $\overline{RTS_k}$ output. $\overline{CTS}$, and $\overline{RTS}$ signals are described later.

Transmission

Transmission is started when the bit 0 (TE_j flag) of $UART_j$ transmit/receive control register 1 is "1", bit 1 is (TI_j flag) of one is "0", and $\overline{CTS_j}$ input is "L". As shown in Figure 39, data is output from TxDj pin when transmission clock CLK_j changes from "H" to "L". The data is output from the least significant bit.

The TIj flag indicates whether the transmission buffer register is empty or not. It is cleared to "0" when data is written in the transmission buffer register and set to "1" when the contents of the transmission buffer register is transferred to the transmission register.

When the transmission register becomes empty after the contents has been transmitted, data is transferred automatically from the transmission buffer register to the transmission register if the next transmission start condition is satisfied. If the bit 2 of $UART_j$ transmit/receive control register 0 is "1", $\overline{CTS_j}$ input is ignored and transmission start is controlled only by the TE_j flag and TI_j flag. Once transmission has started, the TE_j flag, TI_j flag, and $\overline{CTS_j}$ signals are ignored unit data transmission completes. Therefore, transmission is not interrupt when $\overline{CTS_j}$ input is changed to "H" during transmission.

The transmission start condition indicated by TE_j flag, TI_j flag, and $\overline{CTS_j}$ is checked while the T_{ENDj} signal shown in Figure 39 is "H". Therefore, data can be transmitted continuously if the next transmission data is written in the transmission buffer register and TI_j flag is cleared to "0" before the T_{ENDj} signal goes "H".

The bit 3 ($TxEPTY_j$ flag) of $UART_j$ transmit/receive control register 0 changes to "1" at the next cycle after the T_{ENDj} signal goes "H" and changes to "0" when transmission starts. Therefore, this flag can be used to determine whether data transmission has completed.

When the TI_j flag changes from "0" to "1", the interrupt request bit in the $UART_j$ transmission interrupt control register is set to "1".

Receive

Receive starts when the bit 2 (RE_k flag) of $UART_k$ transmit/receive control register 1 is set to "1".

The $\overline{RTS_k}$ output is "H" when the RE_k flag is "0" and goes "L" when the RE_k flag changed to "1". It goes back to "H" when receive starts. Therefore, the $\overline{RTS_k}$ output can be used to determine whether the receive register is ready to receive. It is ready when $\overline{RTS_k}$ output is "L".

The data from the RxD_k pin is retrieved and the contents of the receive register is shifted by 1 bit each time the transmission clock CLK_j changes from "L" to "H". When an 8-bit data is received, the contents of the receive register is transferred to the receive buffer register and the bit 3 (RI_k flag) of $UART_k$ transmit/receive control register 1 is set to "1". In other words, the setting of the RI_k flag indicates that the receive buffer register contains the received data. At this point, $\overline{RTS_k}$ output goes "L" to indicate that the next data can be received. When the RI_k flag changes from "0" to "1", the interrupt request bit in the $UART_k$ receive interrupt control register is set to "1". Bit 4 (OER_k flag) of $UART_k$ transmit/receive control register is set to "1" when the next data is transferred from the receive register to the receive buffer register while RI_k flag is "1", and indicates that the next data was transferred to the receive register before the contents of the receive buffer register was read. RI_k and OER_k flags are cleared automatically to "0" when the loworder byte of the receive buffer register is read. The OER_k flag is also cleared when the RE_k flag is cleared. Bit 5 (FER_k flag), bit 6 (PER_k flag), and bit 7 (SUM_k flag) are ignored in clock synchronous mode.

As shown in Figure 33, with clock synchronous serial communication, data cannot be received unless the transmitter is operating because the receive clock is created from the transmission clock. Therefore, the transmitter must be operating even when there is no data to be sent from $UART_k$ to $UART_j$.

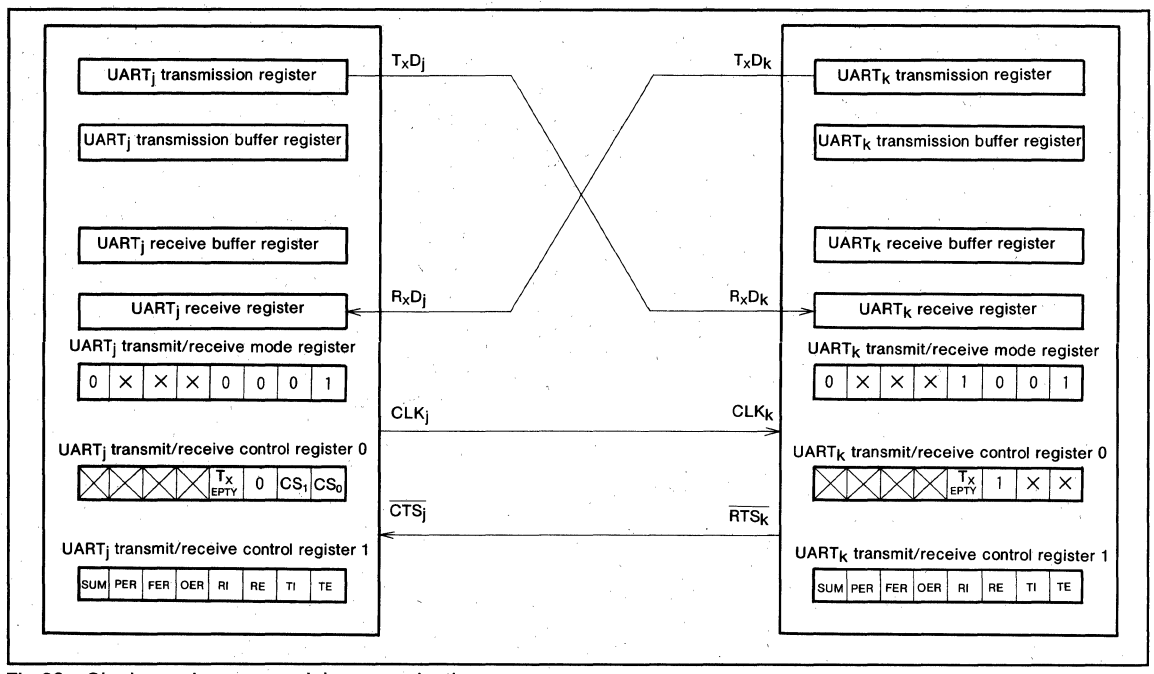


Fig.38 Clock synchronous serial communication

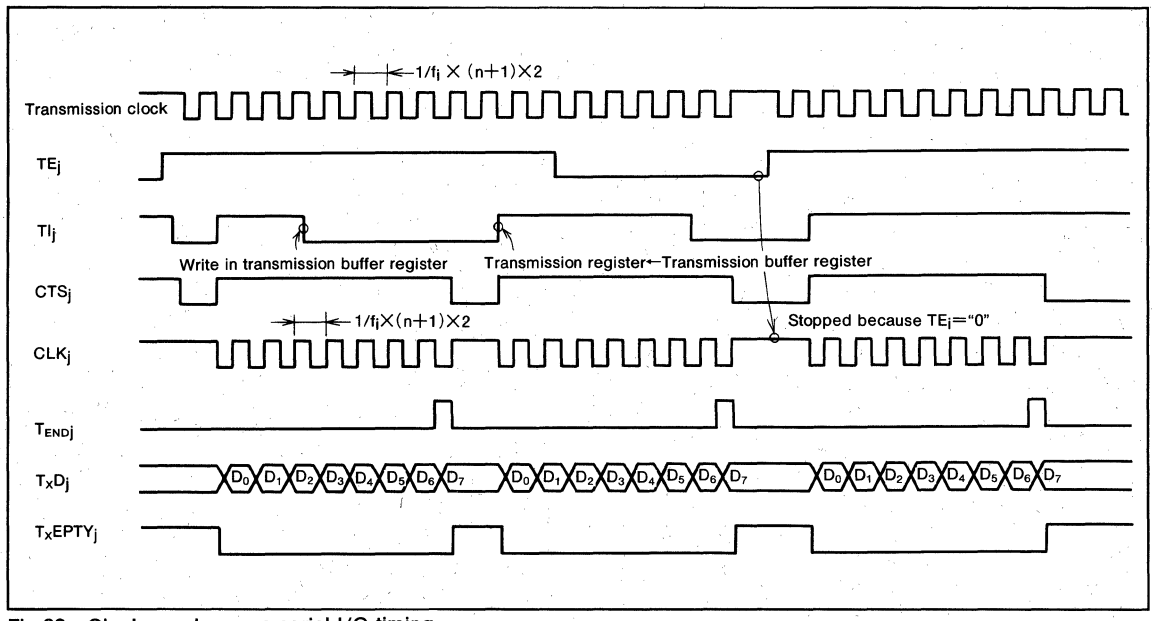


Fig.39 Clock synchronous serial I/O timing

MITSUBISHI ELECTRIC

ASYNCHRONOUS SERIAL COMMUNICATION

Asynchronous serial communication can be performed using 7-, 8-, or 9-bit length data. The operation is the same for all data lengths. The following is the description for 8-bit asynchronous communication.

With 8-bit asynchronous communication, the bit 0 of UARTi transmit/receive mode register is "1", the bit 1 is "0", and the bit 2 is "1".

Bit 3 is used to select an internal clock or an external clock. If bit 3 is "0", an internal clock is selected and if bit 3 is "1", then external clock is selected. If an internal clock is selected, the bit 0 (CS_0) and bit 1 (CS_1) of UARTi transmit/receive control register 0 are used to select the clock source. When an internal clock is selected for asynchronous serial communication, the CLK pin can be used as a normal I/O pin.

The selected internal or external clock is divided by (n+1), then by 16, and passed through a control circuit to create the UART transmission clock or UART receive clock.

Therefore, the transmission speed can be changed by changing the contents n of the bit rate generator. If the selected clock is an internal clock fi or an external clock f_{EXT},

$$\text{Bit Rate} = (\text{fi or } f_{EXT}) / \{(n+1) \times 16\}$$

Bit 4 is the stop bit length selection bit to select 1 stop bit or 2 stop bits.

The bit 5 is a selectiion bit of odd parity or even parity.

In the odd parity mode, the parity bit is adjusted so that the sum of the 1's in the data and parity bit is always odd.

In the even parity mode, the parity bit is adjusted so that the sum of the 1's in the data and parity bit is always even.

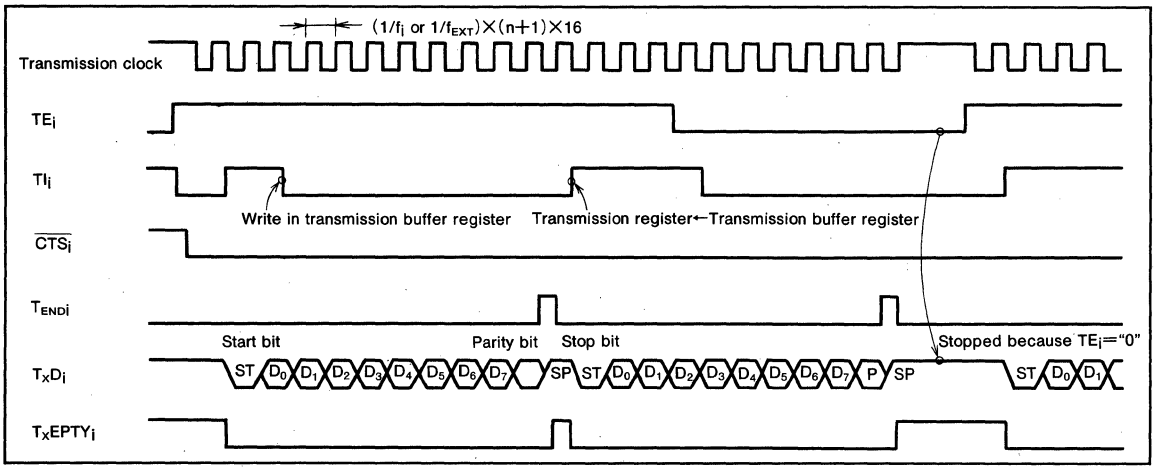


Fig.40 Transmit timing example when 8-bit asynchronous communication with parity and 1 stop bit is selected

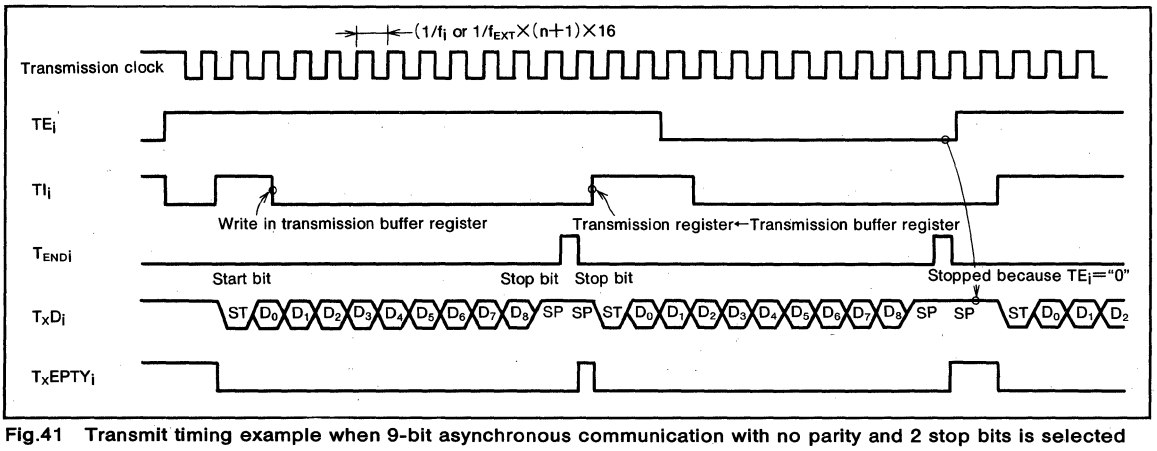


Fig.41 Transmit timing example when 9-bit asynchronous communication with no parity and 2 stop bits is selected

Bit 6 is parity bit selection bit which indicates whether to add parity bit or not.
Bits 4 to 6 should be set or reset according to the data format of the communicating devices.
Bit 7 is the sleep selection bit. The sleep mode is described later.
The UARTi transmit/receive control register 0 bit 2 is used to determine whether to use $\overline{CTS_i}$ input or $\overline{RTS_i}$ output.
$\overline{CTS_i}$ input used if bit 2 is "0" and $\overline{RTS_i}$ output is used if bit 2 is "1".
If $\overline{CTS_i}$ input is selected, the user can control whether to stop or start transmission by external $\overline{CTS_i}$ input. $\overline{RTS_i}$ will be described later.

Transmission

Transmission is started when the bit 0 (TE_i flag) of UARTi transmit/receive control register 1 is "1", the bit 1 (TI_i flag) is "0", and $\overline{CTS_i}$ input is "L" if $\overline{CTS_i}$ input is selected. As shown in Figure 40 and 41, data is output from the TxD_i pin with the stop bit and parity bit specified by the bits 4 to 6 of UARTi transmit/receive mode register bits. The data is output from the least significant bit.
The TI_i flag indicates whether the transmission buffer is empty or not. It is cleared to "0" when data is written in the transmission buffer and set to "1" when the contents of the transmission buffer register is transferred to the transmission register.
When the transmission register becomes empty after the contents has been transmitted, data is transferred automatically from the transmission buffer register to the transmission register if the next transmission start condition is satisfied.
Once transmission has started, the TE_i flag, TI_i flag, and $\overline{CTS_i}$ signal (if $\overline{CTS_i}$ input is selected) are ignored until data transmission is completed.
Therefore, transmission does not stop until it completes even if the TE_i flag is cleared during transmission.
The transmission start condition indicated by TE_i flag, TI_i flag, and $\overline{CTS_i}$ is checked while the T_{ENDi} signal shown in Figure 40 is "H". Therefore, data can be transmitted continuously if the next transmission data is written in the transmission buffer register and TI_i flag is cleared to 0 before the T_{ENDi} signal goes "H".
The bit 3 ($TxEPTY_i$ flag) of UARTi transmit/receive control register 0 changes to "1" at the next cycle after the T_{ENDi} signal goes "H" and changes to "0" when transmission starts. Therefore, this flag can be used to determine whether data transmission is completed.
When the TI_i flag changes from "0" to "1", the interrupt request bit in the UARTi transmission interrupt control register is set to "1".

Receive

Receive is enabled when the bit 2 (RE_i flag) of UARTi transmit/receive control register 1 is set. As shown in Figure 42, the frequency divider circuit at the receiving end begin to work when a start bit is arrived and the data is received.

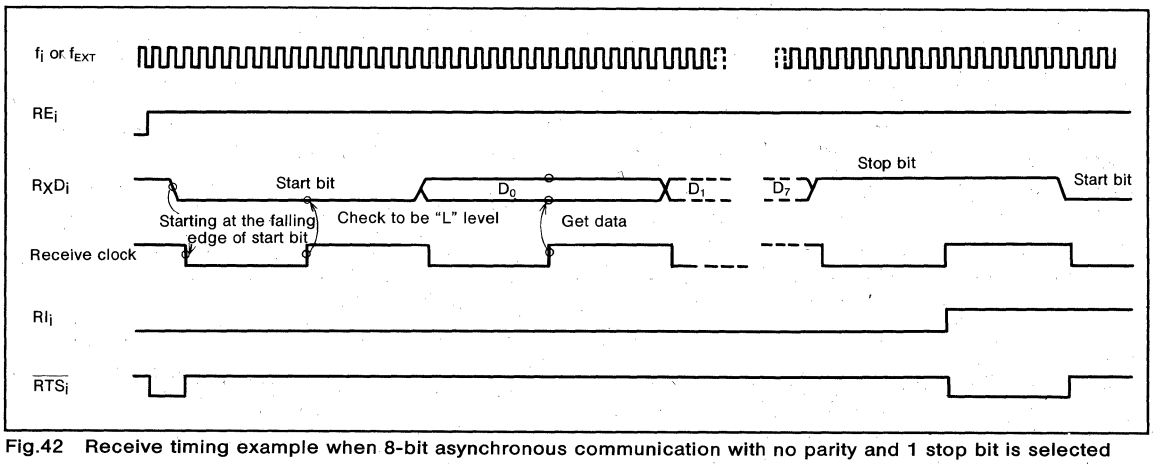


Fig.42 Receive timing example when 8-bit asynchronous communication with no parity and 1 stop bit is selected

If $\overline{RTS_i}$ output is selected by setting the bit 2 of UARTi transmit/receive control register 0 to "1", the $\overline{RTS_i}$ output is "H" when the RE_i flag is "0". When the RE_i flag changes to "1", the $\overline{RTS_i}$ output goes "L" to indicate receive ready and returns to "H" once receive has started. In other words, $\overline{RTS_i}$ output can be used to determine externally whether the receive register is ready to receive.

The entire transmission data bits are received when the start bit passes the final bit of the receive block shown in Figure 35. At this point, the contents of the receive register is transferred to the receive buffer register and the bit 3 of UARTi transmit/receive control register 1 is set. In other words, the RI_i flag indicates that the receive buffer register contains data when it is set. If $\overline{RTS_i}$ output is selected, $\overline{RTS_i}$ output goes "L" to indicate that the register is ready to receive the next data.

The interrupt request bit in the UARTi receive interrupt control register is set when the RI_i flag changes from "0" to "1".

The bit 4 (OER_i flag) of UARTi transmission control register 1 is set when the next data is transferred from the receive register to the receive buffer register while the RI_i flag is "1". In other words when an overrun error occurs. If the OER_i flag is "1", it indicates that the next data has been transferred to the receive buffer register before the contents of the receive buffer register has been read. Bit 5 (FER_i flag) is set when the number of stop bits is less than required (framing error).

Bit 6 (PER_i flag) is set when a parity error occurs.

Bit 7 (SUM_i flag) is set when either the OER_i flag, FER_i flag, or the PER_i flag is set. Therefore, the SUM_i flag can be used to determine whether there is an error.

The setting of the RI_i flag, OER_i flag, FER_i flag, and the PER_i flag is performed while transferring the contents of the receive register to the receive buffer register. The RI_i, OER_i, FER_i, and SUM_i flags are cleared when the low order byte of the receive buffer register is read or when the RE_i flag is cleared.

Sleep mode

The sleep mode is used to communicate only between certain microcomputers when multiple microcomputers are connected through serial I/O.

The sleep mode is entered when the bit 7 of UARTi transmit/receive mode register is set.

The operation of the sleep mode for an 8-bit asynchronous communication is described below.

When sleep mode is selected, the contents of the receive register is not transferred to the receive buffer register if bit 7 (bit 6 if 7-bit asynchronous communication and bit 8 if 9-bit asynchronous communication) of the received data is "0". Also the RI_i, OER_i, FER_i, and the SUM_i flag are unchanged. Therefore, the interrupt request bit of the UARTi receive interrupt control register is also unchanged.

Normal receive operation takes place when bit 7 of the received data is "1".

The following is an example of how the sleep mode can be used.

The main microcomputer first sends data with bit 7 set to "1" and bits 0 to 6 set to the address of the subordinate microcomputer which wants to communicate with. Then all subordinate microcomputers receive the same data. Each subordinate microcomputer checks the received data with software, clears the sleep bit if bits 0 to 6 are its own address and sets the sleep bit if not. Next the main microcomputer sends data with bit 7 cleared. Then the microcomputer with the sleep bit cleared will receive the data, but the microcomputer with the sleep bit set will not. In this way, the main microcomputer is able to communicate with only the designated microcomputer.

A-D CONVERTER

The A-D converter is an 8-bit successive approximation converter.

Figure 43 shows a block diagram of the A-D converter and Figure 44 shows the bit configuration of the A-D control register. The frequency of the A-D converter operating clock ϕ_{AD} is selected by the bit 7 of the A-D control register. When bit 7 is "0", ϕ_{AD} is the clock frequency divided by 8. That is $\phi_{AD}=f(X_{IN})/8$. When bit 7 is "1", ϕ_{AD} is the clock frequency divided by 4 and ϕ_{AD} is $=f(X_{IN})/4$. The ϕ_{AD} during A-D conversion must be 250KHz minimum because the comparator consists of a capacity coupling amplifier.

The operating mode is selected by the bits 3 and 4 of A-D control register. The available operating modes are one-shot, repeat, single sweep, and repeat sweep.

The bit of data direction register bit corresponding to the A-D converter pin must be "0" (input mode) because the analog input port is shared with port P7.

The operation of each mode is described below.

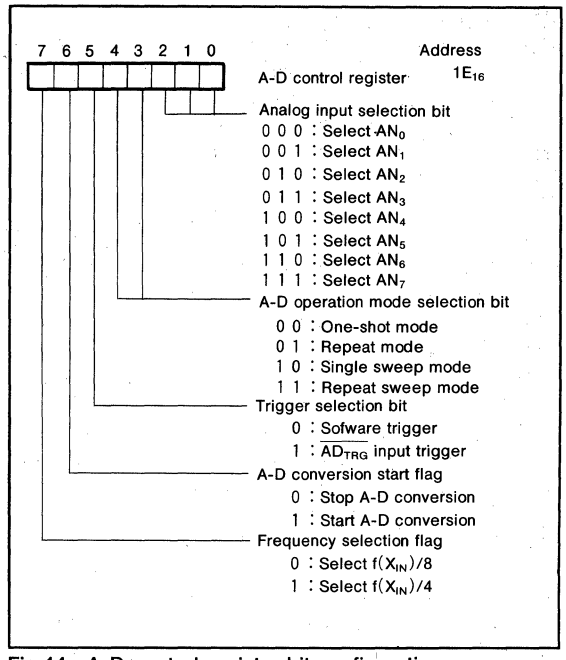


Fig.44 A-D control register bit configuration

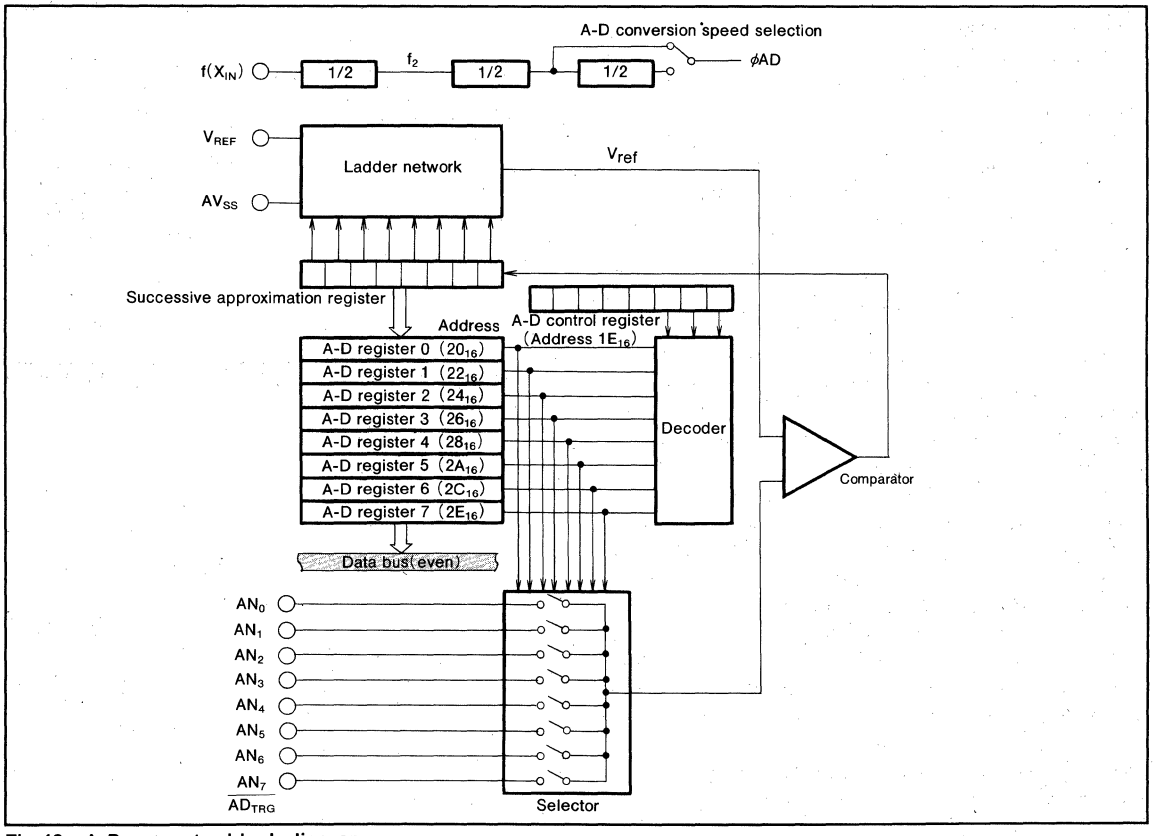


Fig.43 A-D converter block diagram

MITSUBISHI ELECTRIC

(1) One-shot mode [00]

The A-D conversion pins are selected with the bit 0 to 2 of A-D control register. A-D conversion can be started by a. software trigger or by an external trigger.

A software trigger is selected when the bit 5 of A-D control register is "0" and an external trigger is selected when it is "1".

When a software trigger is selected, A-D conversion is started when bit 6 (A-D conversion start flag) is set. A-D conversion ends after 57 ϕ_{AD} cycles and an interrupt request bit is set in the A-D conversion interrupt control register. At the same time, A-D control register bit 6 (A-D conversion start flag) is cleared and A-D conversion stops. The result of A-D conversion is stored in the A-D register corresponding to the selected pin.

If an external trigger is selected, A-D conversion starts when the A-D conversion start flag is "1" and the $\overline{AD_{TRG}}$ input changes from "H" to "L". In this case, the pins that can be used for A-D conversion are AN_0 to AN_6 because the $\overline{AD_{TRG}}$ pin is shared with the analog voltage input pin AN7. The operation is the same as with software trigger except that the A-D conversion start flag is not cleared after A-D conversion and a retrigger can be available during A-D conversion.

(2) Repeat mode [01]

The operation of this mode is the same as the operation of one-shot mode except that when A-D conversion of the selected pin is complete and the result is stored in the A-D register, conversion dose not stop, but is repeated. Also, no interrupt request is issued in this mode. Furthermore, if software trigger is selected, the A-D conversion start flag is not cleared. The contents of the A-D register can be read at any time.

(3) Single sweep mode [10]

In the sweep mode, the number of analog input pins to be swept can be selected. Analog input pins are selected by bits 1 and 0 of the A-D sweep pin selection register (address $1F_{16}$) shown in Figure 45. Two pins, four pins, six pins, or eight pins can be selected as analog input pins, depending on the contents of these bits.

A-D conversion is performed only for selected input pins. After A-D conversion is performed for input of AN_0 pin, the conversion result is stored in A-D register 0, and in the same way, A-D conversion is performed for selected pins one after another. After A-D conversion is performed for all selected pins, the sweep is stopped.

A-D conversion can be started with a software trigger or with an external trigger input. A software trigger is selected when bit 5 is "0" and an external trigger is selected when it is "1".

When a software trigger is selected, A-D conversion is started when A-D control register bit 6 (A-D conversion start flag) is set. When A-D conversion of all selected pins end, an interrupt bit is set In the A-D conversion interrupt control register. At the same time, A-D control register bit 6 (A-D conversion start flag) is cleared and A-D conversion stops.

When an external trigger is selected, A-D conversion starts when the A-D conversion start flag is "1" and the $\overline{AD_{TRG}}$ input changes from "H" to "L". In this case, the A-D conversion result of the trigger input itself is stored in the A-D register 7 because the $\overline{AD_{TRG}}$ pin is shared with AN_7 pin.

The operation is the same as done by software trigger except that the A-D conversion start flag is not cleared after A-D conversion and a retrigger can be available during A-D conversion.

(4) Repeat sweep mode [11]

The difference with the single sweep mode is that A-D conversion dose not stop after converting from the AN_0 pin to the selected pins, but repeats again from the AN_0 pin. The repeat is performed among the selected pins. Also, no interrupt request is generated. Furthermore, if software trigger is selected, the A-D conversion start flag is not cleared. The A-D register can be read at any time.

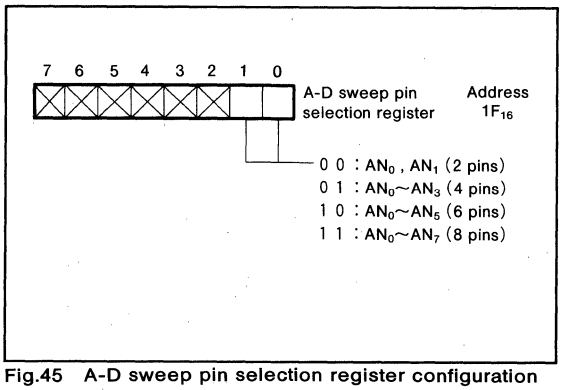


Fig.45 A-D sweep pin selection register configuration

DMA CONTROLLER

The DMA (direct memory access) controller is a 4-channel controller that is provided to carry out data transfer from memory to memory, memory to I/O, or I/O to memory at high speed without using the CPU.

Figure 46 shows the DMA controller block diagram. Figure 47 shows the DMA control related register memory map. Figure 48 shows the DMAC control register bit configuration.

The DMAC control register consists of 16 bits. Bit 0 is a priority selection bit and bit 1 is a terminal count $\overline{TC}$ pin validity bit. Bits 4 through 7 are DMA request flags. These bits can be read to determine whether DMA requests occur on individual channels. Bits 8 through 11 are software DMA request bits, and used to be occured the DMA request by software. Bits 12 through 15 are DMA permission bits. The DMA request is accepted only when the DMA permission bits are "1". All these bits default to 0 upon resetting.

Figure 49 shows the DMAi control register (i=0 to 3) bit configuration. Each channel of the DMAi control register consists of 8 bits. Bits 0 through 3 are used for DMA request source selection.

Bit 4 determines whether the edge or level sensing function is to be used for selecting the source of the request from the DMA request input $\overline{DMAREQi}$ pin. Bit 5 is a DMA acknowledge output $\overline{DMAACKi}$ pin validity bit. When this bit is "0", the $\overline{DMAACKi}$ pin is invalid. When the bit is "1", on the other hand, the pin is valid.

Figure 50 shows the DMAi mode register bit configuration. Each channel of the DMAi mode register (i=0 to 3) consists of 16 bits. Further details are given below.

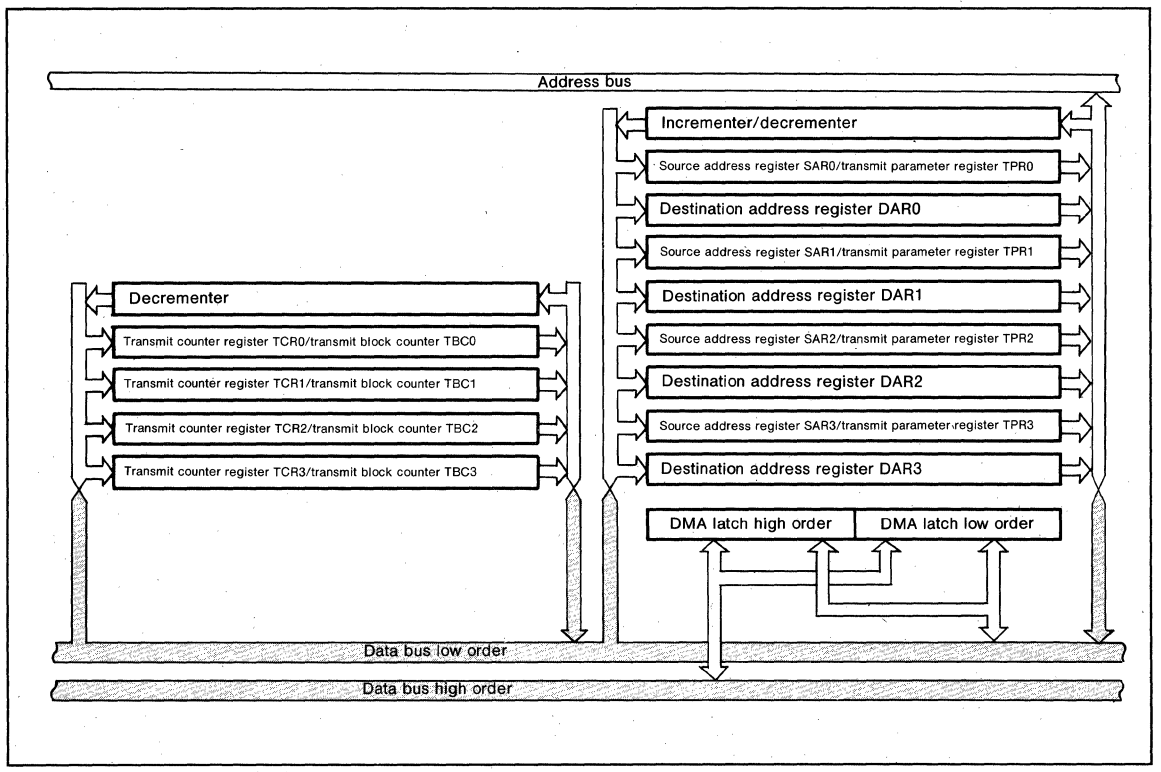


Fig.46 DMA controller block diagram

Address(Hexadecimal notation)	
000068	DMAC control register L
000069	DMAC control register H
001FC0	Source address register 0 L
001FC1	Source address register 0 M
001FC2	Source address register 0 H
001FC3	
001FC4	Destination address register 0 L
001FC5	Destination address register 0 M
001FC6	Destination address register 0 H
001FC7	
001FC8	Transmit counter register 0 L
001FC9	Transmit counter register 0 M
001FCA	Transmit counter register 0 H
001FCB	
001FCC	DMA0 mode register L
001FCD	DMA0 mode register H
001FCE	DMA0 control register
001FCF	
001FD0	Source address register 1 L
001FD1	Source address register 1 M
001FD2	Source address register 1 H
001FD3	
001FD4	Destination address register 1 L
001FD5	Destination address register 1 M
001FD6	Destination address register 1 H
001FD7	
001FD8	Transmit counter register 1 L
001FD9	Transmit counter register 1 M
001FDA	Transmit counter register 1 H
001FDB	
001FDC	DMA1 mode register L
001FDD	DMA1 mode register H
001FDE	DMA1 control register
001FDF	

Address(Hexadecimal notation)	
001FE0	Source address register 2 L
001FE1	Source address register 2 M
001FE2	Source address register 2 H
001FE3	
001FE4	Destination address register 2 L
001FE5	Destination address register 2 M
001FE6	Destination address register 2 H
001FE7	
001FE8	Transmit counter register 2 L
001FE9	Transmit counter register 2 M
001FEA	Transmit counter register 2 H
001FEB	
001FEC	DMA2 mode register L
001FED	DMA2 mode register H
001FEE	DMA2 control register
001FEF	
001FF0	Source address register 3 L
001FF1	Source address register 3 M
001FF2	Source address register 3 H
001FF3	
001FF4	Destination address register 3 L
001FF5	Destination address register 3 M
001FF6	Destination address register 3 H
001FF7	
001FF8	Transmit counter register 3 L
001FF9	Transmit counter register 3 M
001FFA	Transmit counter register 3 H
001FFB	
001FFC	DMA3 mode register L
001FFD	DMA3 mode register H
001FFE	DMA3 control register
001FFF	

Fig.47 DMA controller related register map

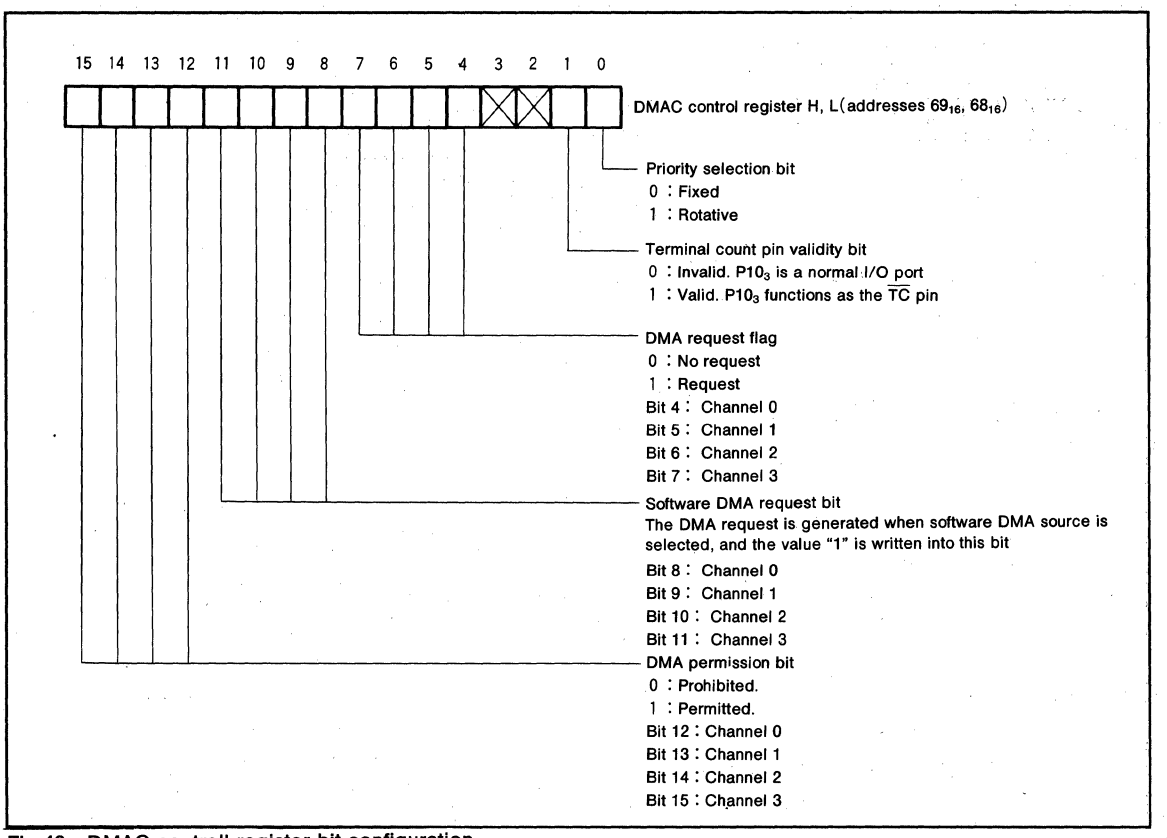


Fig.48 DMAC controll register bit configuration

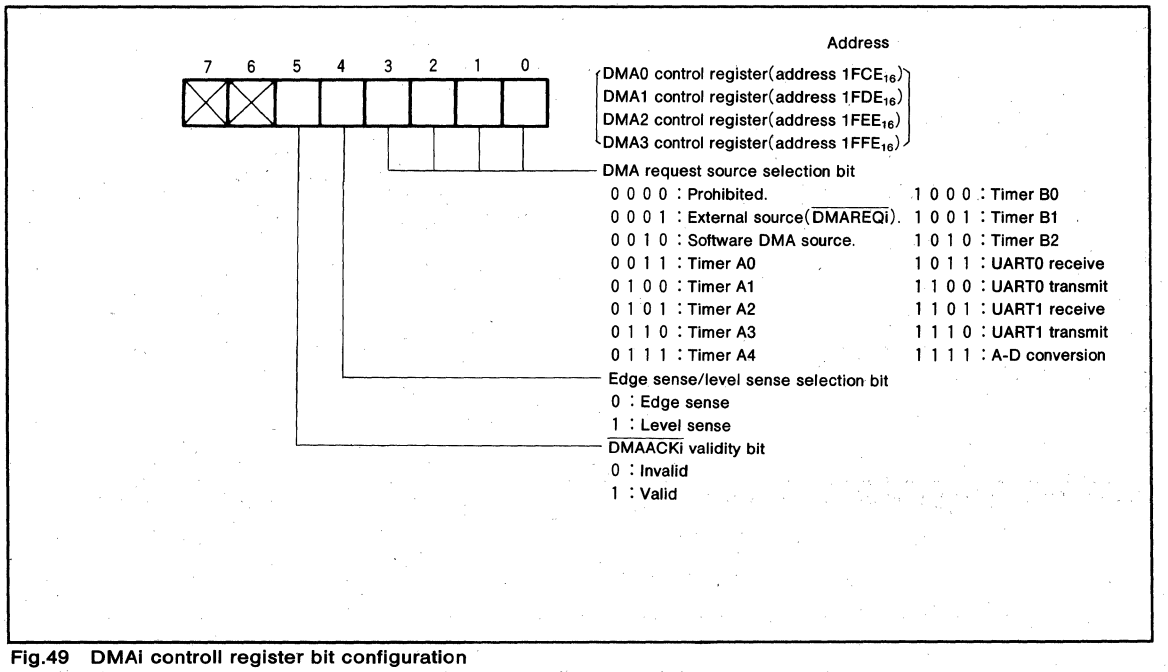


Fig.49 DMAi controll register bit configuration

MITSUBISHI ELECTRIC

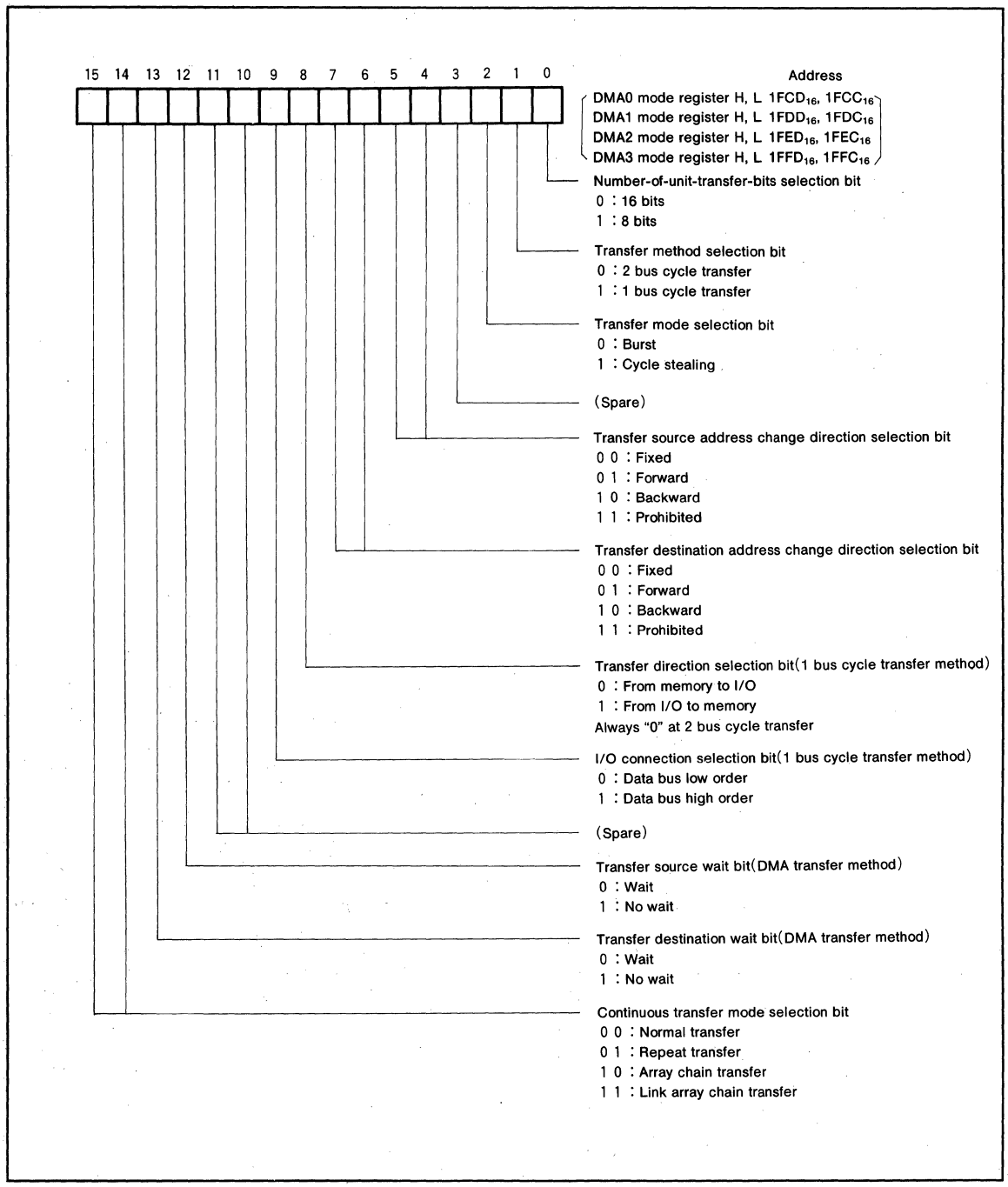


Fig.50 DMAi mode register bit configuration

Pin descriptions

The $\overline{\text{DMAREQi}}$, $\overline{\text{DMAACKi}}$, and $\overline{\text{TC}}$ pins are used for DMA transfer.

$\overline{\text{DMAREQi}}$ (i=0 to 3) is a DMA request input pin, and port $P9_1$, $P9_3$, $P9_5$ and $P9_7$ share with $\overline{\text{DMAREQ0}}$, $\overline{\text{DMAREQ1}}$, $\overline{\text{DMAREQ2}}$ and $\overline{\text{DMAREQ3}}$ pins respectively, and used when a request for DMA transfer is made from external equipment. When the DMAi control register DMA request source selection bits (bits 3 through 0) are set to "0001", the input from this pin becomes the DMA request signal. To use the $\overline{\text{DMAREQi}}$ pin function, set the associated bit of the port P9 direction register to input.

$\overline{\text{DMAACKi}}$ (i=0 to 3) is a DMA acknowledge output pin, and port $P9_0$, $P9_2$, $P9_4$ and $P9_6$ share with $\overline{\text{DMAACK0}}$, $\overline{\text{DMAACK1}}$, $\overline{\text{DMAACK2}}$ and $\overline{\text{DMAACK3}}$ pins respectively. When bit 5 ($\overline{\text{DMAACKi}}$ validity bit) of the DMAi control register of each channel is set to "1", $\overline{\text{DMAACKi}}$ serves as the dedicated $\overline{\text{DMAACKi}}$ signal output pins. During DMA transfer, the operating channel acknowledge signal output is generated no matter whether the 1 bus transfer or 2 bus transfer method is employed. When the acknowledge signal is not to be used, set the $\overline{\text{DMAACKi}}$ validity bit to "0" so that $\overline{\text{DMAACKi}}$ can be used as the I/O pin.

$\overline{\text{TC}}$ is a terminal count pin and $\overline{\text{TC}}$ pin shares with port $P10_3$. When the value "1" is written at bit 1 of the DMA control register, the $\overline{\text{TC}}$ pin takes effect. At this time, the $\overline{\text{TC}}$ pin serves as the N-channel open drain output terminal. When the transfer counter register or transfer block counter value is "0", the $\overline{\text{TC}}$ pin generates a one-cycle ϕ "L" level output.

When the $\overline{\text{TC}}$ pin validity bit is "1", any ongoing channel DMA transfer can be stopped by changing the level of the input from the $\overline{\text{TC}}$ pin from "H" to "L".

Data transfer method

Two different data transfer method are used: two bus cycle transfer and one bus cycle transfer. The two bus cycle transfer method is effective for memory-to-memory data transfer, whereas the one bus cycle transfer method is effective for memory-to-I/O or I/O to memory data transfer. Two transfer method are detailed below.

(1) Two bus cycle transfer

When bit 1 of the DMAi mode register shown in Figure 50 is set to "0", the two bus cycle transfer method is selected. This method accomplishes one unit of transfer by performing one reading bus cycle and one writing bus cycle. One unit of transfer refers to the number of bits that can be transferred by one DMA transfer operation. It is determined by bit 0 of the DMAi mode register. When the bit is set to "0", one transfer unit consists of 16 bits (2 bytes). When the bit is "1", on the other hand, one transfer unit consists of 8 bits (1 byte).

At two bus cycle transfer, bit 8 of DMAi mode register must be "0".

Figure 51 shows an example of two bus cycle transfer. In the reading cycle, the transfer source (memory 1) address is output to the address bus, its address data is read in one-transfer units, and stored in the DMA controller DMA data latch (16-bit). In the writing cycle, on the other hand, the transfer destination address is output to the address bus and the data stored in the DMA data latch is written at the transfer destination (memory 2) address. If the read/write operation is not completed by one session, it is divided into two sessions (for details see under Bus cycle at DMA transfer).

When two bus cycle transfer is effected in 16-bit units with an external bus width of 16 bits employed, the following operations cannot be performed.

1. The transfer source and destination are DRAM, and an odd-numbered address designated as the first address for the backward transfer source address change direction.
2. The transfer destination is DRAM, and an odd-numbered address designated as the first address for the backward transfer destination address change direction.

(2) One bus cycle transfer

When bit 1 of the DMAi mode register is set to "1", the one bus cycle transfer method is selected.

When data transfer is to be made between I/O and memory, this system reads from memory at the same time it writes into I/O, or vice versa, to carry out data transfer at high speed. Bit 8 of the DMAi mode register determines whether one bus cycle transfer is to be made from memory to I/O or from I/O to memory. When this bit is set to "0", data transfer is made from memory to I/O. When the bit is "1", on the other hand, data transfer is made from I/O to memory.

Figures 52 through 55 show examples of one bus cycle transfer. Figure 52 shows a memory-to-I/O DMA transfer in cases where the transfer is effected in 16-bit units with an external bus width of 16 bits employed. The memory (transfer source) address is output to the address bus. The R/$\overline{\text{W}}$ signal is "H", and the $\overline{\text{BLE}}$ and $\overline{\text{BHE}}$ signals are "L". The read operation is performed at both even- and odd-numbered addresses. This ensures that the memory data is read into the data bus high-order area ($D_8 \sim D_{15}$) and low-order area ($D_0 \sim D_7$). At the same time, chip selection is made in accordance with the acknowledge signal from the $\overline{\text{DMAACKi}}$ pin, and the data read from memory is directly acquired when the $\overline{\text{E}}$ signal level rises. In this manner, data is transferred from memory to I/O in one bus cycle.

In data transfer from I/O to memory, chip selection is made according to the acknowledge signal from the $\overline{\text{DMAACKi}}$ pin and data is read into the data bus from I/O. At the same time, the memory (transfer destination) address is output to the address bus, and the R/$\overline{\text{W}}$ signal goes Low to write data into memory. To receive the one bus cycle transfer

service for DMA purposes, however, the external circuitry must be formed in such a manner that the read and write signal inputs for I/O are the reversal of the M37720S1FP's read and write signal outputs. Figure 53 shows the data transfer data flow of Figure 52.

When the address changes in forward direction, data 1 and data 2 are transferred to the I/O's 8 low-order bits and 8 high-order bits, respectively, in the first cycle of DMA transfer. In the next cycle of DMA transfer, data 3 and data 4 are transferred to the I/O's 8 low-order bits and 8 high-order bits, respectively. In this manner, the memory data are sequentially transferred to the I/O.

If the address changes in backward direction, on the other hand, data 10 and data 9 are transferred to the I/O's 8 high-order bits and 8 low-order bits, respectively, in the first DMA transfer cycle.

When one bus cycle transfer is effected in 16-bit units with an external bus width of 16 bits employed, as shown in Figure 52, 16 bits of data are simultaneously read into the data bus. Therefore, determine the transfer start address so that transfer begins with an even-numbered address when the address change mode is forward or that transfer begins with an odd-numbered address when the address change mode is backward.

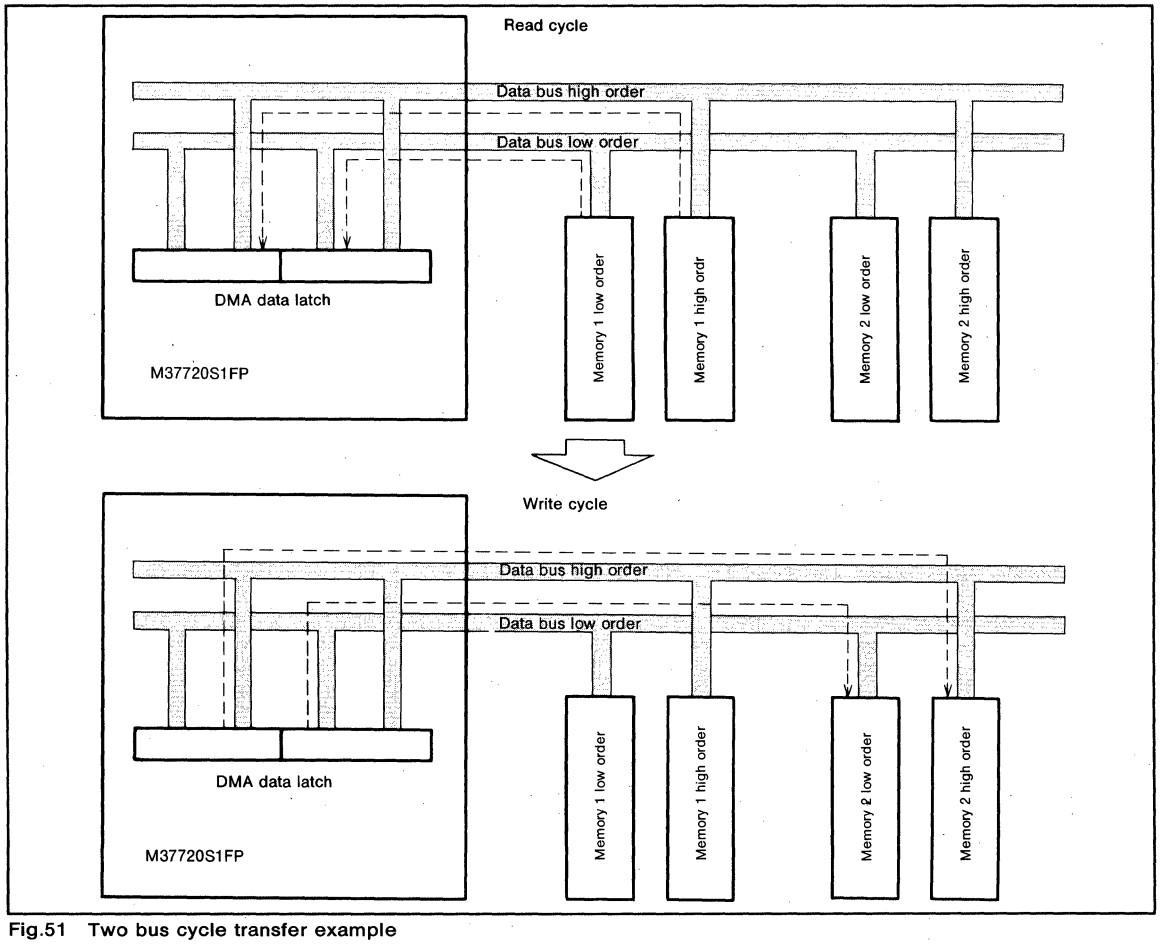


Fig.51 Two bus cycle transfer example

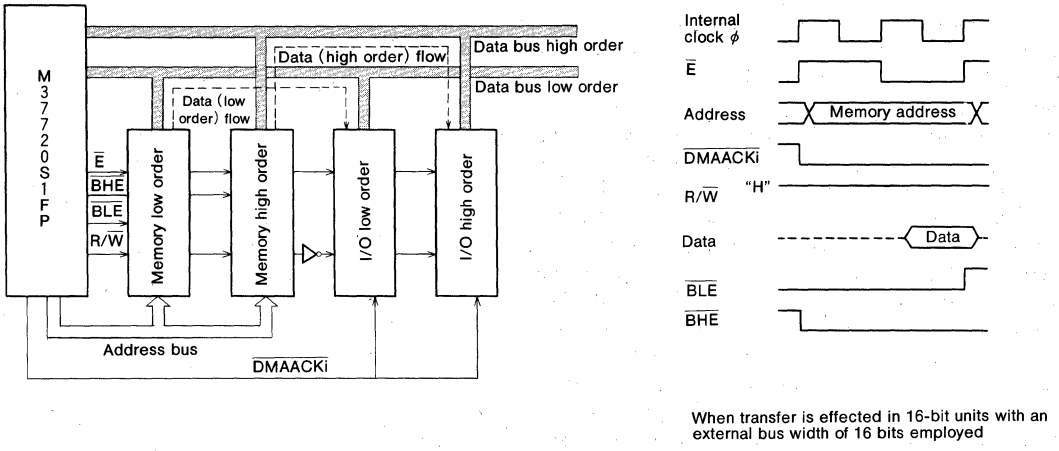


Fig.52 1 bus cycle transfer example(memory to I/O)(1)

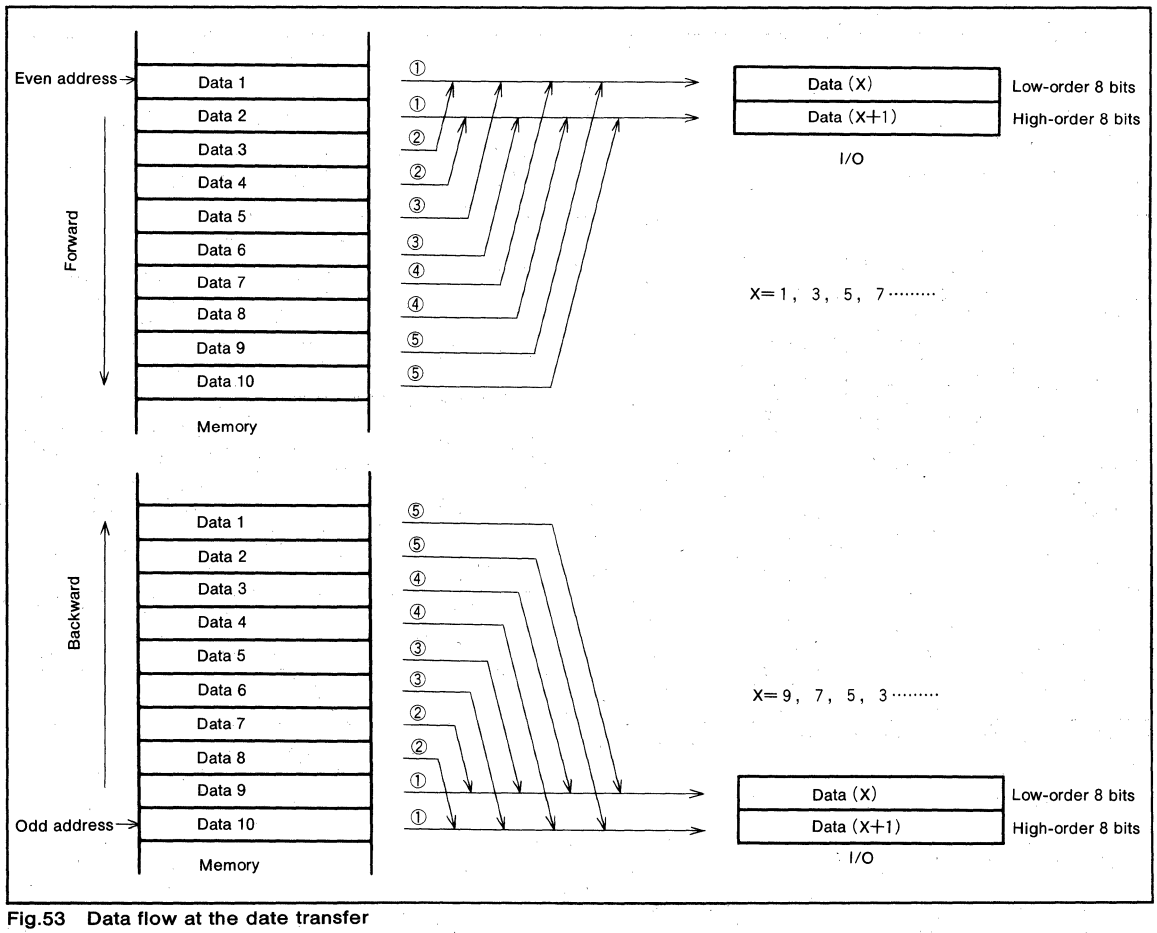
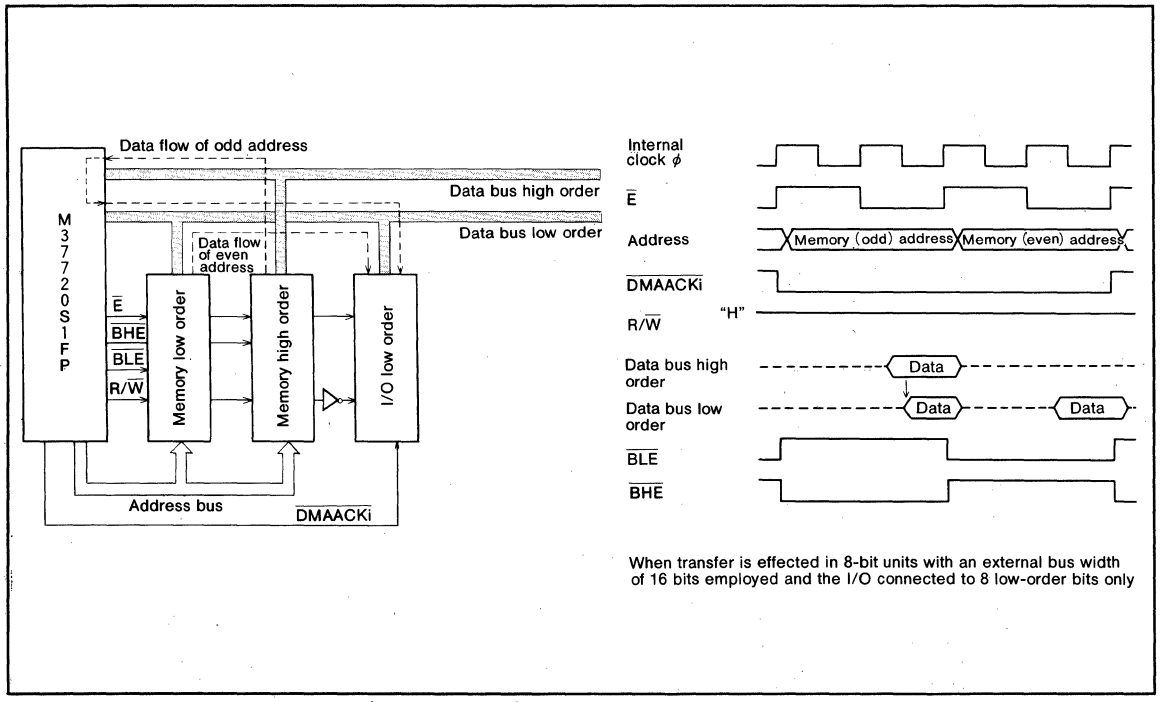


Fig.53 Data flow at the date transfer

Figures 54 and 55 indicate the cases where transfer is made in 8-bit units with an external bus width of 16 bits employed.

Figure 54 shows the cases where an external bus width of 16 bits is employed with the I/O connected to the 8 low-order bits only. Connect the odd-numbered address memories to data bus high-order area ($D_8 \sim D_{15}$) and the even-numbered address memories to data bus low-order area ($D_0 \sim D_7$). When data is transferred from an odd-numbered memory address to I/O, the data is read from the memory high-order to the data bus high-order. The read data first goes into the chip, is copied by the DMA controller to the data bus low-order, and then output outside. Therefore, the data output to the data bus low-order is the same as that for the high-order. For the I/O, chip selection is made according to the $\overline{DMAACKi}$ signal so that the data output to the data bus low-order is acquired. When data is transferred from an even-numbered address memory to I/O, the data read from the memory low-order is directly acquired by the I/O. At this time, M37720S1FP data bus pins D_0 to D_7 are floating.



When transfer is effected in 8-bit units with an external bus width of 16 bits employed and the I/O connected to 8 low-order bits only

Fig.54 1 bus cycle transfer example(memoly to I/O)(2)

Figure 55 shows the case where the I/O is connected to the 8 high-order bits only. When data is transferred from an odd-numbered memory address to I/O, the data read from the memory high-order is directly acquired by the I/O. At this time, M37720S1FP data bus pins D_8 through D_{15} are floating. When data is transferred from an even-numbered memory address to I/O, the data is read from the memory low-order to the data bus low-order. The read data first goes into the chip, is copied by the DMA controller to the data bus high-order, and then output outside. The I/O acquires that data. In one bus cycle transfer, data bus pin ($D_0 \sim D_7$, $D_8 \sim D_{15}$) I/O changeover is automatically effected to initiate DMA transfer no matter whether the 8-bit I/O is connected to the low-or high-order of the data bus. In the above data copy between the chip's internal data buses, there is limitation on the transfer rate.

When one bus cycle transfer is effected with an external bus width of 8 bit, I/O connection selection bit must be "0".

When one bus cycle transfer is effected in 16-bit units with an external bus width of 16 bits employed, the following operation cannot be performed.

1. The object memory is DRAM, and an odd-numbered address designated as the first address for the backward transfer address change direction. Figure 56 shows the data bus status when one bus cycle transfer is made under the conditions indicated in Table 5.

Table 5. Data bus conditions for one bus cycle DMA transfer

Number	Memory	External bus width	Transfer unit	$\overline{BLE}$	$\overline{BHE}$	I/O position
①	External	16-bit	16-bit	0	0	Upper/Lower
②	Internal	16-bit	16-bit	0	0	Upper/Lower
③	External	16-bit	8-bit	0	1	Lower
④	External	16-bit	8-bit	0	1	Upper
⑤	External	16-bit	8-bit	1	0	Lower
⑥	External	16-bit	8-bit	1	0	Upper
⑦	Internal	16-bit	8-bit	0	1	Lower
⑧	Internal	16-bit	8-bit	0	1	Upper
⑨	Internal	16-bit	8-bit	1	0	Lower
⑩	Internal	16-bit	8-bit	1	0	Upper
⑪	External	8-bit	8-bit	—	—	Lower
⑫	Internal	8-bit	8-bit	—	—	Lower

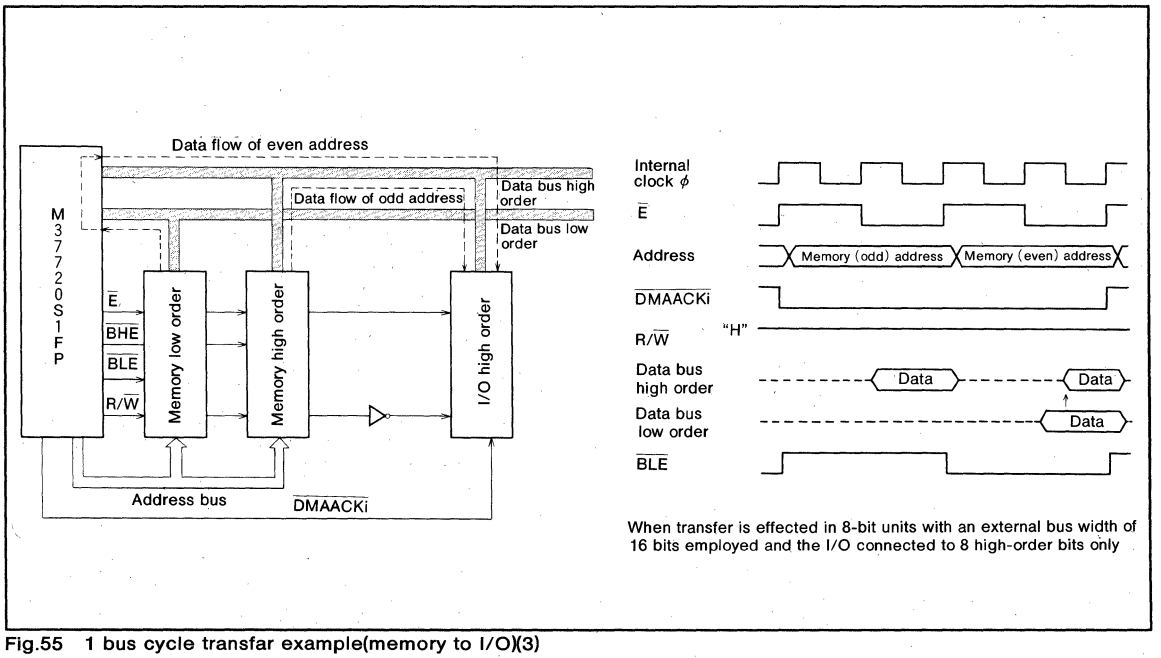


Fig.55 1 bus cycle transfar example(memory to I/O)(3)

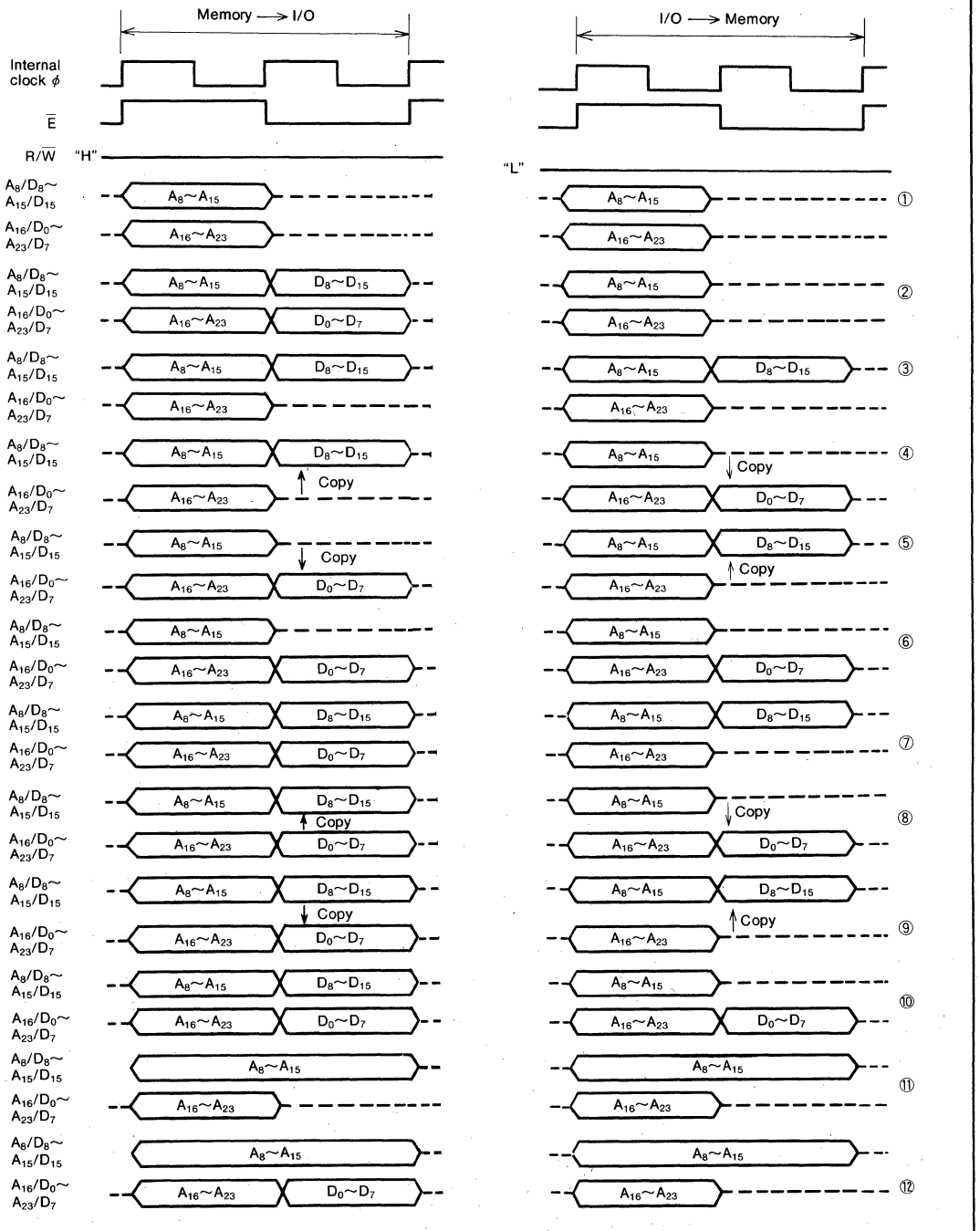


Fig.56 Data bus status under various conditions of one bus cycle DMA transfer

DMA request sources

One out of fifteen DMA request sources can be selected for each channel. There are a total of fifteen DMA request sources. The internal DMA request sources are the A-D conversion, UART0 transmit/receive, UART1 transmit/receive, timer A4, timer A3, timer A2, timer A1, timer A0, timer B2, timer B1, timer B0 and the software DMA source by programs. The external DMA request source is one associated with $\overline{\text{DMAREQi}}$ pin input. For DMA request source selection, use the DMAi control register DMAi request source selection bits (bits 0 through 3) as shown in Figure 49. Table 6 indicates the contents of the bits and DMA request sources. The request timing is the same as that for interrupt.

When the software DMA source is selected with the DMA request source selection bits, the DMA request flag is set to "1" by writing the value "1" for the DMAC control register software DMA request bits (bits 8 through 11). When the DMA request flag is "1", the software DMA request bits are automatically cleared. When the external source is selected with the DMA request source selection bits, the input from the $\overline{\text{DMAREQi}}$ pin sets the DMA request flag to "1". The DMA transfer request will not be accepted until both the DMAC control register DMA permission bit and DMA request flag are "1". Therefore, if the DMA permission bit is "0" while the DMA request flag is "1", the DMA request will not be accepted. Note that DMA permission bit is cleared to "0" upon resetting. Therefore, after DMA transfer parameter and other data setup, set to "1" the DMA permission bit of a DMA channel to be rendered valid. This assures that the transfer request bit of that channel takes effect, making is possible to effect DMA transfer.

Table 6. Relationship between DMA request source selection bits (bits 3 to 0) and DMA request sources

b3	b2	b1	b0	Cause
0	0	0	0	Prohibited
0	0	0	1	External source ($\overline{\text{DMAREQi}}$)
0	0	1	0	Software DMA source
0	0	1	1	Timer A0
0	1	0	0	Timer A1
0	1	0	1	Timer A2
0	1	1	0	Timer A3
0	1	1	1	Timer A4
1	0	0	0	Timer B0
1	0	0	1	Timer B1
1	0	1	0	Timer B2
1	0	1	1	UART0 receive
1	1	0	0	UART0 transmit
1	1	0	1	UART1 receive
1	1	1	0	UART1 transmit
1	1	1	1	A-D conversion

Transfer mode

Two DMA transfer modes are available: burst transfer mode and cycle steal transfer mode. Mode selection is made variously for all channeles, using bit 2 of the DMAi mode register. When this bit is set to "0", the burst transfer mode is selected. This mode is automatically selected upon resetting.

(1) Burst transfer mode

When a DMA request is request is received from a certain channel in the burst transfer mode, the DMA request from another channel will not be accepted until the DMA transfer on the former channel is completed. In the burst transfer mode, either the edge sense or level sense mode can be selected only when the input from the $\overline{\text{DMAREQi}}$ pin (external source) is chosen as the request source.

When DMAi control register bit 4 is set to "0", the edge sense mode is selected. The edge sense mode is automatically selected upon resetting. In the edge sense mode, the DMA request flag is set to "1" when the input from the $\overline{\text{DMAREQi}}$ pin falls. In the burst transfer edge sense mode, the DMA request flag is cleared to "0" when any of the following conditions is met.

1. The channel i DMA permission bit is cleared to "0" (forced termination of transfer).
2. The channel i DMA request flag is cleared to "0".
3. Channel i DMA transfer is wholly terminated.
4. The "L" level is entered to the $\overline{\text{TC}}$ pin during channel i transfer execution (forced termination of transfer).

Figure 57 shows an example of edge sense mode burst transfer. In this example, the $\overline{\text{DMAREQi}}$ pin input (external source) is chosen as the DMA request source. When the $\overline{\text{DMAREQi}}$ pin input changes from the "H" to "L" level during CPU operation, the DMA request flag is set to "1" and the DMA controller acquires the right of bus use and initiates transfer. From high to low, the bus use priority is for DRAM refresh, $\overline{\text{HOLD}}$, DMA controller, and CPU. Therefore, if the request is made by the DRAM refresh, which has a higher priority than the DMA controller, the DMA controller halts any ongoing transfer operation at the end of the current transfer cycle and transfers the bus use right to the DRAM controller as shown in Figure 57. Upon getting the bus use right, the DRAM controller generates the refresh cycle. When refreshing is terminated, the DMA controller resumes the execution of the interrupted DMA transfer at the point of interruption. Once a DMA request is received in the burst transfer mode, no request from another channel is accepted unless DMA transfer is entirely terminated or the transfer operation is brought to a forced stop. Therefore, even when the request flag of channel 0, which has a high priority, is set to "1" in the middle of transfer as shown in Figure 57, such a request will not be accepted (the priority is explained in the next section). When channel 1 DMA transfer is completed, the bus use right is once trans-

ferred to the CPU, and the DMA transfer request from channel 0 is later accepted at the end of the current bus cycle.

When bit 4 of the DMAi control register is set to "1", the level sense mode is selected. The level sense mode can only be used for the DMA request from the $\overline{\text{DMAREQi}}$ pin. When selecting other sources, be sure to select the edge sense mode.

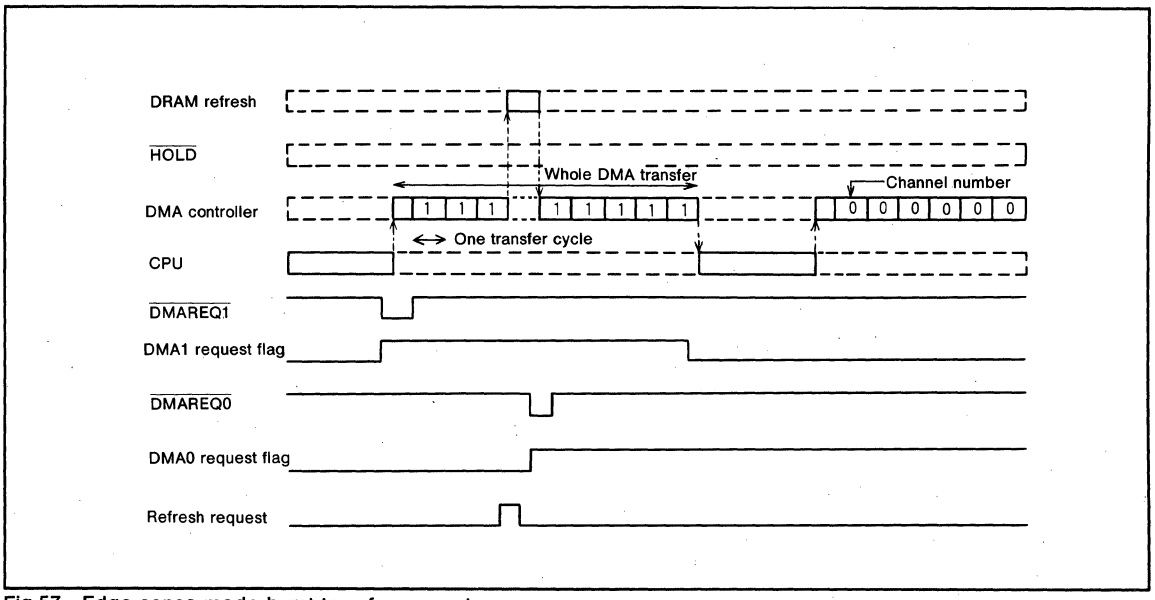


Fig.57 Edge sense mode burst transfer example

In the level sence mode, the DMAi request flag is set to "1" to initiate DMA transfer only while the $\overline{\text{DMAREQi}}$ pin input is "L". If the $\overline{\text{DMAREQi}}$ pin input level returns to "H" in the middie of transfer, the DMA operation is interrupted at the end of the current transfer cycle so that the bus use right is returned to the CPU. At this time, the DMA permission bit is not cleared. When the $\overline{\text{DMAREQi}}$ pin input goes Low, the transfer operation is resumed at the address subsequent to the point of interruption. In the level sense mode, the DMA request flag varies only with the level of the input from the $\overline{\text{DMAREQi}}$ pin. Therefore, while the $\overline{\text{DMAREQi}}$ pin input level is Low, the DMA request flag remains to be "1" even if transfer is completed.

Figure 58 shows an example of level sense mode burst transfer. When the channel 1 $\overline{\text{DMAREQi}}$ pin input level changes from "H" to "L" during CPU operation, the DMA request flag is set to "1" so that the DMA controller acquires the bus use right and initiates transfer. When the $\overline{\text{DMAREQi}}$ pin input returns to the "H" level, the DMA1 request flag is cleared. This causes the DMA transfer operation to be interrupted and returns the bus use right to the CPU. When the $\overline{\text{DMAREQ1}}$ pin input is returned to the "L" level again and the DMA1 request flag set to "1", DMA transfer is resumed at the point of interruption. As with the edge sense mode, no transfer request from a different channel is accepted while DMA transfer is executed for a certain channel. For instance, if the channel 1 request flag is set to "0" and the request flag of channel 0, which has a higher priority, is set to "1", the new request will not be accepted. When channel 1 DMA transfer is wholly terminated, the DMA permission bit is set to "0" so that the bus use right is transfered to the CPU. The DMA transfer request from channel 0 is then accepted at the end of a bus cycle.

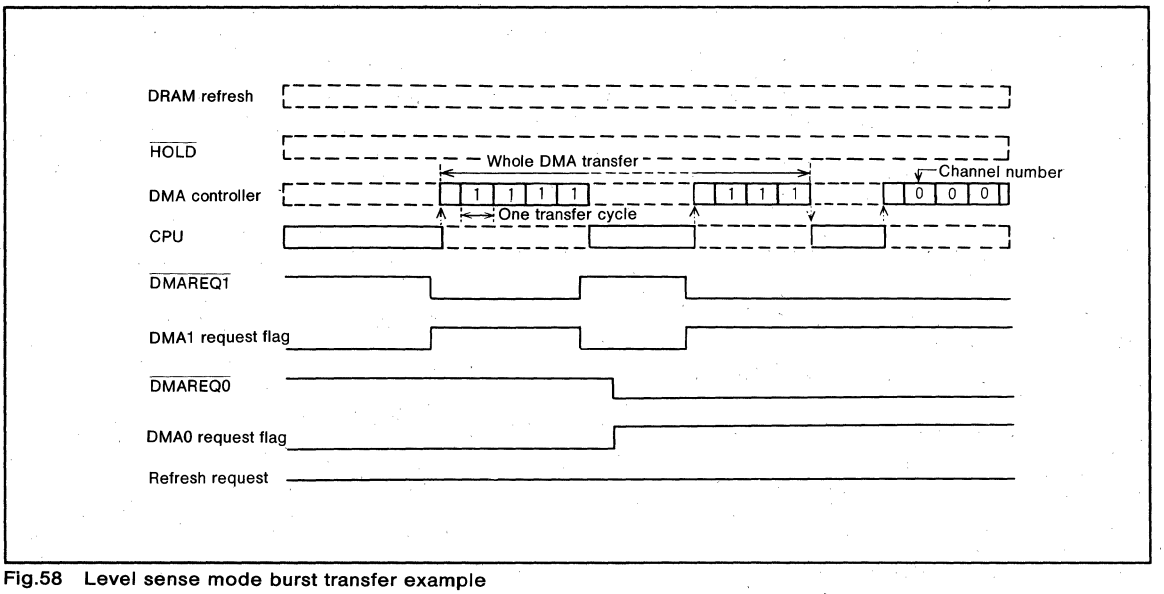


Fig.58 Level sense mode burst transfer example

(2) Cycle steal transfer mode

When bit 2 of the DMAi mode register is set to "1", the cycle steal transfer mode is selected. In the cycle steal transfer mode, be sure to select the edge sense mode.

When the DMA request occurs in the cycle steal transfer mode, the DMA request flag is set to "1" as in the burst transfer mode. When the DMA request from a channel is accepted, DMA transfer starts. However, the DMA request flag is automatically cleared at the beginning of the first DMA transfer cycle. Therefore, if there is no channel DMA request at the end of one-unit transfer, the DMA controller returns the bus use right to the CPU. If there is a DMA request from any channel, the DMA controller continues to use the bus and initiates DMA transfer for the channel. In the cycle steal transfer mode, the priorities of the channels are considered at all times to assure that the DMA request from a channel having the highest priority is accepted to initiate DMA transfer execution. Transfer is effected in one transfer unit segments. However, the DMA permission bit will not be cleared even if the DMA request flag is cleared. Therefore, when the DMA request flag is set to "1", transfer is resumed at the point of interruption. When the transfer counter value is "0" in normal transfer, or both of the transfer counter and transfer block counter value is "0" in array chain transfer, the DMA permission bit is cleared to terminate the whole DMA transfer operation.

Figure 59 shows an example of cycle steal transfer. When the $\overline{\text{DMAREQi}}$ pin input changes from "H" to "L", the DMA1 request flag is set to "1" and the DMA controller acquires the bus use right to initiate DMA transfer. The DMA1 request flag is cleared when the channel 1 transfer cycle starts. Therefore, if there is no DMA transfer request from the other channels, the DMA controller returns the bus use right to the CPU at the end of one transfer cycle. In the example showed in Figure 59, however, DMA0 transfer cycle execution continues because the channel 0 request flag is set to "1". When the DMA0 transfer cycle is terminated, the DMA request flags of all channels are set to "0" so that the DMA controller returns the bus use right to the CPU. When the DMA1 request flag is set to "1", only one cycle of transfer operation is performed. Even if the DMA1 request flag is cleared at this time, the DMA1 request flag is set to "1" again to effect continued transfer as long as the $\overline{\text{DMAREQi}}$ pin input goes Low before the end of the next transfer cycle. In cycle steal transfer, the priorities of individual channels are examined at the end of each transfer cycle. Therefore, if the request is sent from channel 0, which has a higher priority than channel 1, channel 0 transfer is executed first. Further, if the refresh request is sent from the DRAM controller, such a request for bus use takes precedence as it has a high priority.

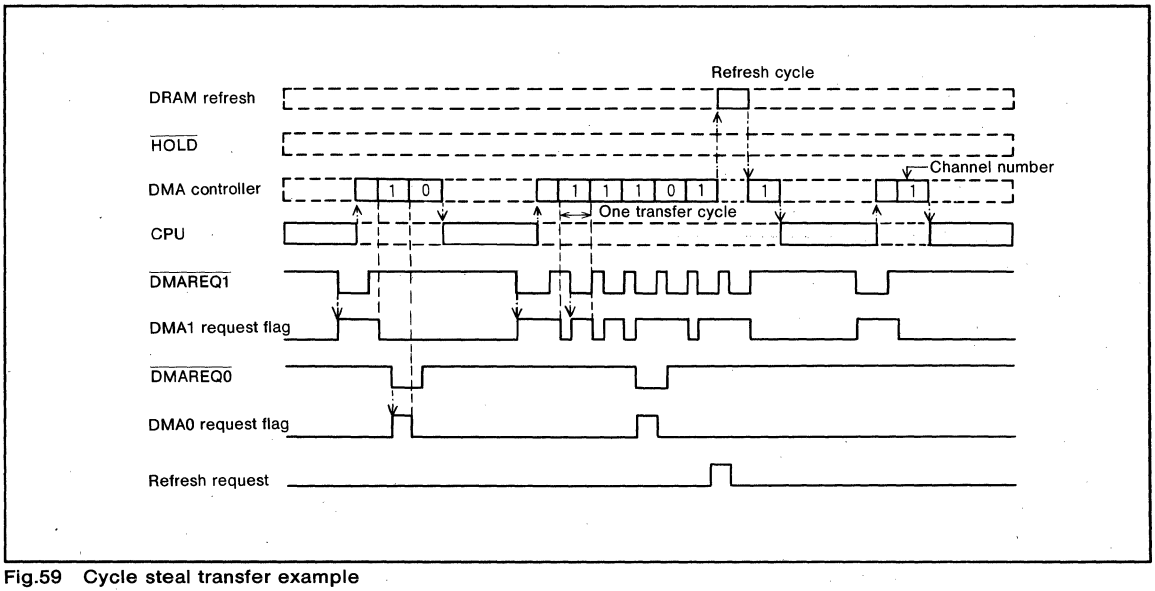


Fig.59 Cycle steal transfer example

Priority

Priorities are assigned to all DMA channels. Either the fixed or rotative priority can be selected. When bit 0 (priority selection bit) of the DMAC control register is set to "0", the fixed priority is selected. Note that the fixed priority is automatically chosen upon resetting. In the fixed priority, the channels are given fixed priorities and DMA transfer is executed in the order of priority. From high to low, the priorities are assigned to channels 0, 1, 2, and 3. As indicated in Figure 61, the priorities are checked when the first DMA request is received in the burst transfer mode or at each cycle end in the cycle steal transfer mode.

When bit 0 of the DMAC control register is set to "1", the rotative priority is used. From high to low, the initial priorities are assigned to channels 0, 1, 2, and 3 as is the case with the fixed priority. When the DMA transfer for one channel is normally terminated with the rotative priority employed, the priorities are rotated in such a manner that the channel, for which transfer has just been completed, has the lowest priority. For example, when channel 0 transfer is normally terminated as shown in Figure 60, the priorities are rotated upon completion of transfer so that the new priorities are, in decreasing order, channel 1, channel 2, channel 3, and channel 0. The priorities remain unchanged when DMA transfer is forcibly terminated by $\overline{TC}$ pin input or DMA permission bit clearing.

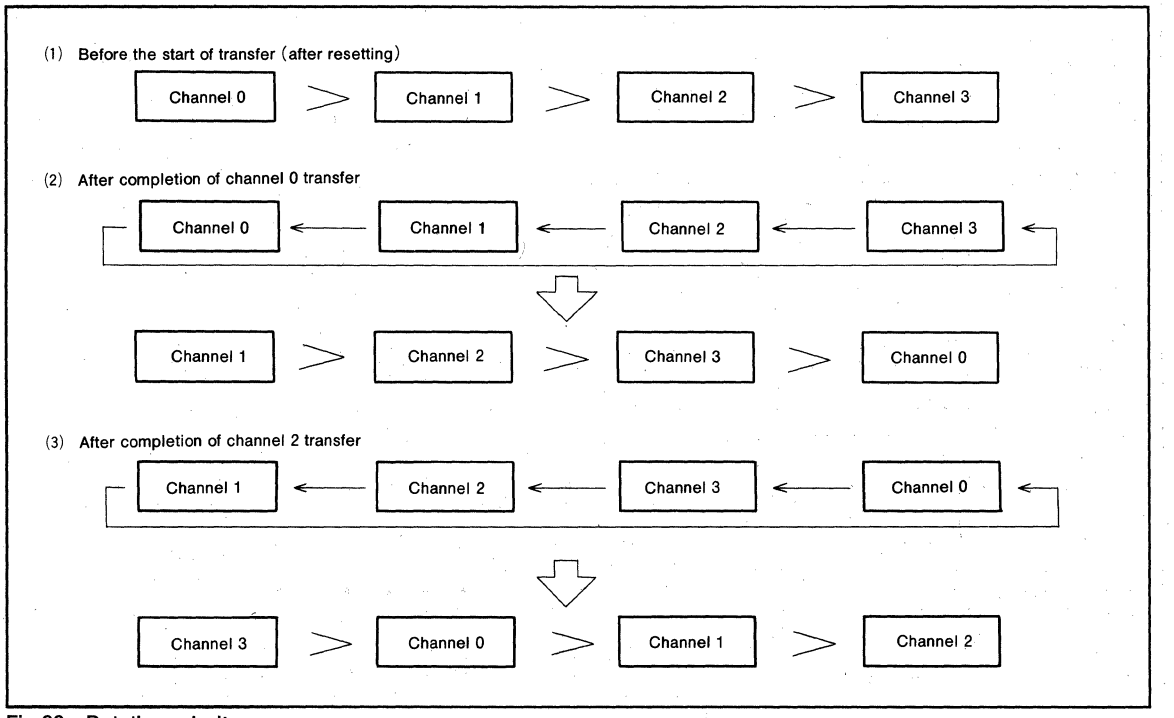
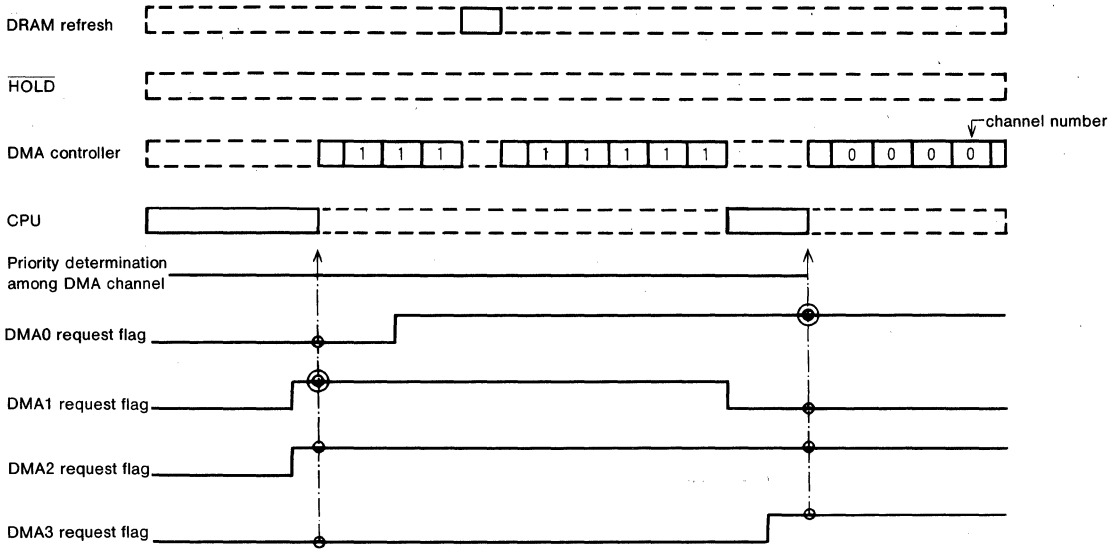
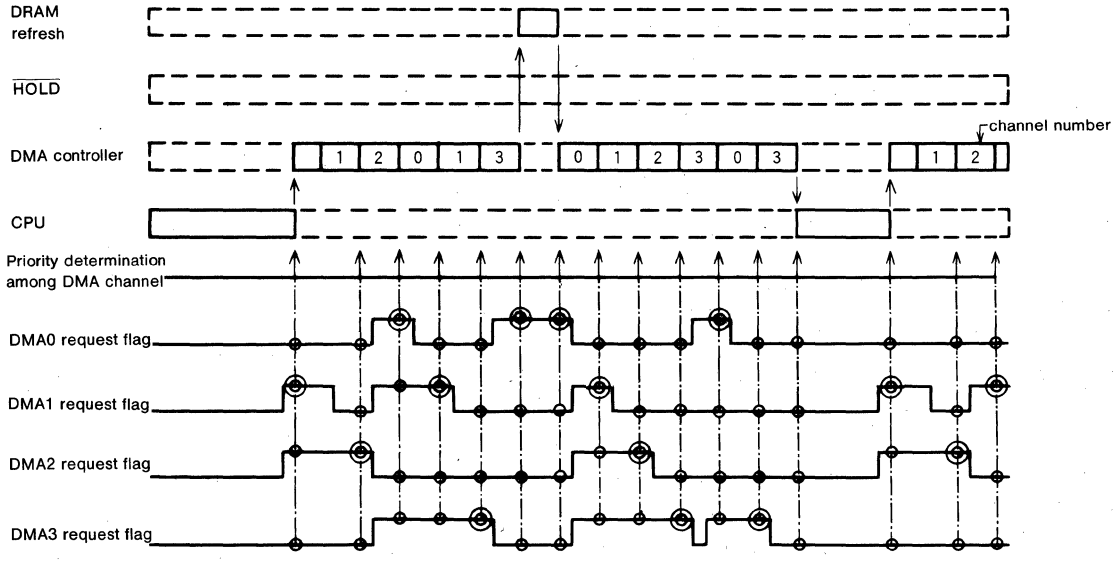


Fig.60 Rotative priority



Priority resolution at burst transfer mode



Priority resolution at cycle steal transfer mode

◎ : Channel having the highest priority

Fig.61 Channel priority determination timing (Fixed priority)

Transfer address change direction

The address change direction in DMA transfer can be designated independently for the transfer source and destination. Choices are "forward", "backward", and "fixed". When the forward direction is chosen, the address increments. When the backward direction is selected, the address decrements. When the fIxed direction is selected, the address is fixed(2 bytes when one transfer unit consists of 16 bits or 1 byte when one transfer unit consists of 8 bits) and does not change. Use bits 4 and 5 of the DMAi mode register shown in Figure 50 to specify the transfer address change direction for the source. For the destination, use bits 6 and 7.

Figure 62 shows an example transfer address change mode when two bus cycle transfer is effected in 16-bit units. Figure 62-(1) shows an example case where both the DMAi mode register transfer source and destination address change directions are set to "forward". In this type of setup, data are transferred from memory 1 to memory 2 in the ①, ②, ③, and ④ order. Figure 62-(2) shows a case where the transfer source address change direction is "forward" and the destination addresses are "fixed". In this setup, the memory 1 data are called up in the forward address change direction and written at fixed memory 2 addresses in one transfer unit segments. Figure 62-(3) shows a case where the transfer source address change direction is forward and the destination address change direction is backward. Figure 62-(4) shows a case where the transfer source address change direction is backward and the destination addresses are fixed. In this setup, the memory 1 data are written at fixed memory 2 addresses in one transfer unit segments in the ①, ②, and ③ order.

As explained above, three different address change modes are selectable for each of the transfer source and destination. A total of nine different address change direction combinations are available.

In one bus cycle transfer, the memory side address change direction depends on the memory bits. For data transfer from memory to I/O, therefore, use bits 4 and 5 (transfer source address change direction selection bits) of the DMAi mode register to determine the memory side (transfer source) address change direction. This is not affected by bits 6 and 7 (transfer destination address change direction selection bits). For data transfer from I/O to memory, use bits 6 and 7 of the DMAi mode register to set the memory side (transfer destination) address change direction. This is not affected by bits 4 and 5.

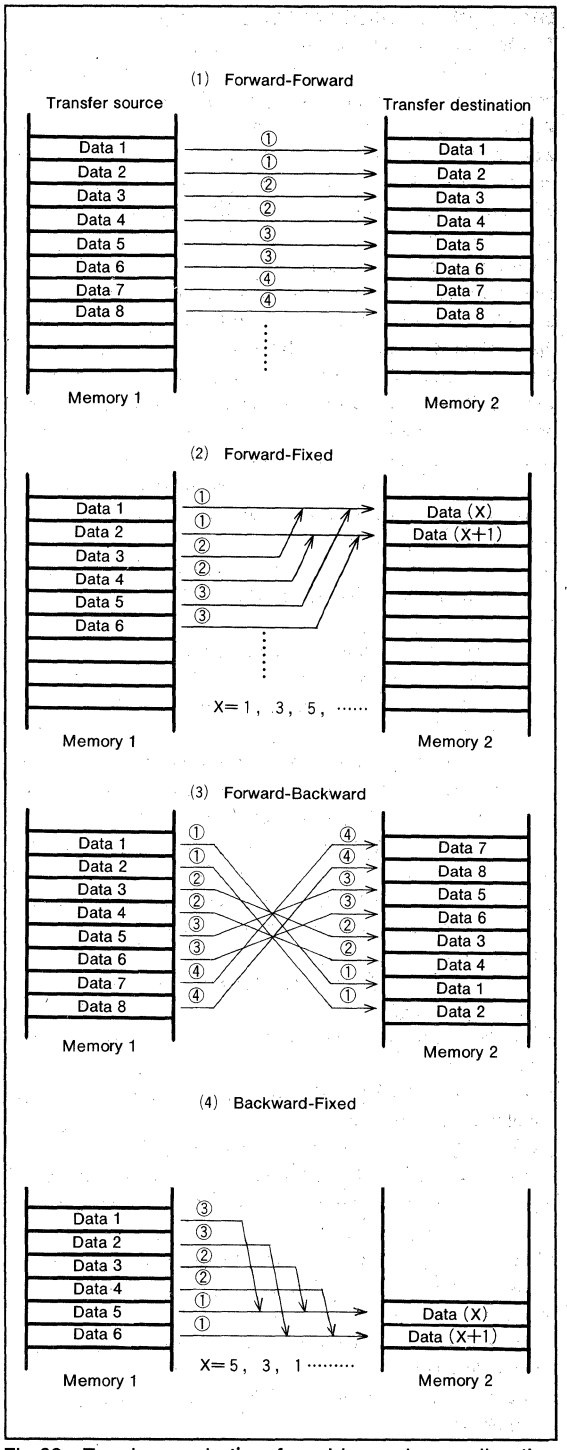


Fig.62 Two bus cycle transfer address change direction examples (when one transfer unit consists of 16 bits)

Bus cycle at DMA transfer

The memory or I/O read/write cycle (hereinafter referred to as the bus cycle) is basically the same as the bus cycle of CPU. Figure 63 shows six examples of bus cycle.

When transfer is made in 8-bit (1-byte) units, the operating waveform is as shown in Figure 63-(1). Bit 12 of the DMAi mode register is the transfer source wait bit and bit 13 is the transfer destination wait bit for DMA transfer. When the transfer source wait bit is set to "0" in cases where the DMA controller performs a read operation, the $\overline{E}$ signal "L" output period is extended as shown in Figure 63-(2) so that the "L" width of $\overline{E}$ signal is extended 2 times as long. When the transfer destination wait bit is set to "0" in cases where the DMA controller performs a write operation, the "L" width of $\overline{E}$ signal is extended 2 times as long. All channels have DMAi mode register, it is possible to determine whether or not to extend the access time depending on the speed of the memory space element to be accessed. The DMA controller wait bit is effective for both the internal and external memory areas.

When transfer is made in 16-bit (2-byte) units, the operating waveform looks like Figure 63-(1) if the following two conditions are met.

1. The associated channel transfer address change direction is either forward or fixed.
2. The first address of the memory to be accessed is even-numbered.

However, when the transfer source or destination wait bit is "0", the DRAM area is being accessed, or transfer parameters are read in the array chain or link array chain transfer mode, the $\overline{E}$ signal "L" period is extended so that the operating waveform looks like Figure 63-(2).

When transfer is made in 16-bit (2-byte) units, the operating waveform looks like Figure 63-(3) if the following two conditions are met.

1. The associated channel transfer address change direction is backward.
2. The first address of the memory to be accessed is odd-numbered.

In this case, although the two bytes can be accessed simultaneously, one additional internal clock cycle ϕ is needed. However, when the transfer source or destination wait bit is "0" or the DRAM area is being accessed, the $\overline{E}$ signal "L" period is extended so that the operating waveform looks like Figure 63-(4).

When transfer is made in 16-bit (2-byte) units, the operating waveform looks like Figure 63-(5) if the following two conditions are met.

1. The associated channel transfer address change direction is forward.
2. The first address of the memory to be accessed is odd-numbered.

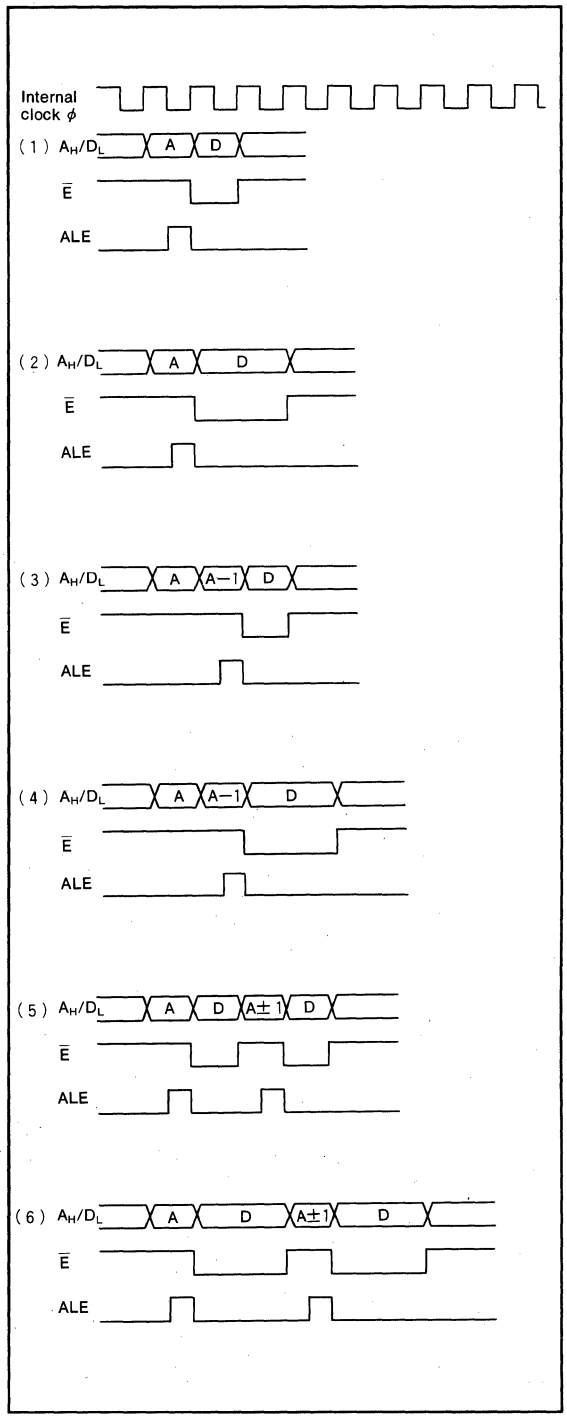


Fig.63 Bus cycle example

The same waveform results if the following two conditions are met.

1. The associated channel transfer address change direction is backward.
2. The first address of the memory to be accessed is even-numbered.

In these cases, accessing is done in 1 byte units only. Therefore, accessing is conducted in two sessions. However, if the transfer source or destination wait bit is "0" or the DRAM area is being accessed, the $\overline{E}$ signal "L" period is extended. If the wait bits of the transfer source and destination are both "0", the operating waveform looks like Figure 63-(6).

DMA continuous transfer

(1) Normal transfer

With the normal transfer function, it is possible to effect DMA transfer of the preselected number of bytes. As shown in Figure 50, first set up the number of one-transfer unit bits, transfer method, transfer mode, and transfer address change direction of the DMAi mode register, and then write the transfer source block first transfer address (the block's lowest-order address in the forward or fixed transfer address change direction or the block's highest-order address in the backward address change direction) into the source address register (hereinafter referred to as the SAR). Further, write the destination block first transfer address (the lowest-order address in the forward or fixed transfer address change direction or the highest-order address in the backward transfer address change direction) into the destination address register (hereinafter referred to as the DAR). Also write the desired number of bytes to be transferred, into the transfer counter register (hereinafter referred to as the TCR). Write the value 1 or more into TCR. The SAR, DAR, and TCR have 24 bits each. Be sure to write into all these bits. The SAR, DAR, and TCR are located at the addresses shown in Figure 47. The next step is to perform DMA source and other setups for the DMAi control register shown in Figure 49. Set up bit 0 (priority selection bit) and bit 1 (terminal count validity bit) of the DMAC control register shown in Figure 48,and finally set the DMAC control register DMA permission bit to "1" so as to make the DMA request acceptable.

When the contents of TCR are "0", the terminal count $\overline{TC}$ signal output is generated, and at the same time, the interrupt request bit of the DMA interrupt control register is set to "1".

To halt transfer, enter the "L" level to the $\overline{TC}$ pin or write the value "0" in the DMA permission bit position. At this time, the interrupt request bit of the DMA interrupt control register is not set.

An example normal transfer operation performed under the following conditions is presented below.

- Transfer unit: 16 bits
- Transfer method: 2 bus cycle transfer
- Transfer mode: Burst transfer mode (edge sense)
- Transfer source address change direction: Forward.
- Transfer destination address change direction: Forward
- Transfer source wait: None
- Transfer destination wait: None
- DMA source: External input ($\overline{DMAREQi}$)

When the $\overline{DMAREQi}$ pin input level changes from "H" to "L", the DMA request flag is set to "1" so that the DMA transfer request is accepted.

At this time, if there is no refresh request from the DRAM controller or no $\overline{HOLD}$ request is made from the $\overline{HOLD}$ pin, the DMA request is accepted. If no request is received from a channel having a higher priority than the accepted DMA channel, the DMA transfer for that accepted channel starts. If there is a DMA request from a channel having a higher priority, the DMA transfer for such a channel takes precedence.

As two bus cycle transfer is made, a read operation is performed in the first bus cycle. First the address written into the SAR is output to the address bus and then inputted into the incrementor/decrementor (hereinafter referred to as the I/D). The I/D adds 1 or 2 to the entered address and outputs the result back to the SAR. If one 16-bit transfer operation is not enough to complete the read operation, the read operation is divided into two sessions to achieve the purpose. The operation performed so far is called the read cycle and the R/$\overline{W}$ signal goes High. The data from memory or I/O is stored in the data latch within the DMA controller.

The write operation is performed in the next bus cycle. First the address written in the DAR is output to the address bus and then inputted into the I/D. The I/D adds 1 or 2 to the entered address and outputs the result back to the DAR. If one 16-bit transfer operation is not enough to complete the write operation, the write operation is divided into two sessions to achieve the purpose. The operation performed so far is called the write cycle and the R/$\overline{W}$ signal goes Low. The data stored in the DMA controller data latch in the read cycle is output to the data bus in the write cycle and written into the destination memory or I/O. The operations performed so far comprise one data transfer unit. In two bus cycle transfer, the read and write cycle combination is called the DMA transfer cycle. DMA transfer is executed by repeating the DMA transfer cycle.

In two bus cycle transfer, the TCR varies in the read cycle. The remaining transfer bytes are read from the TCR in concurrence with address output from SAR in the read cycle, and inputted into the decrementor (hereinafter referred to as the D). The D subtracts 1 or 2 from the number of remaining bytes and outputs the result back to the TCR. In this manner, the contents of the TCR decrease after each one-unit transfer. When the number of remaining bytes,

which is read from the TCR, becomes "0", the DMA controller outputs the terminal count $\overline{TC}$ signal to the $\overline{TC}$ pin, and at the same time, sets the interrupt request bit of the DMA interrupt control register to "1". At this time, the DMA permission bit is cleared. As the burst transfer mode is selected in this example, the DMA request flag is also cleared.

To halt transfer before completion, enter the "L" level to the $\overline{TC}$ pin ($P10_3$) or write the value "0" in the DMA permission bit.

In normal transfer, the first values written in the SAR, DAR, and TCR are retained in the internal latches. Therefore, if DMA transfer is to be effected under the same conditions, it can be initiated simply by setting the DMA permission bit to "1".

Figure 64 shows the normal transfer timing diagram.

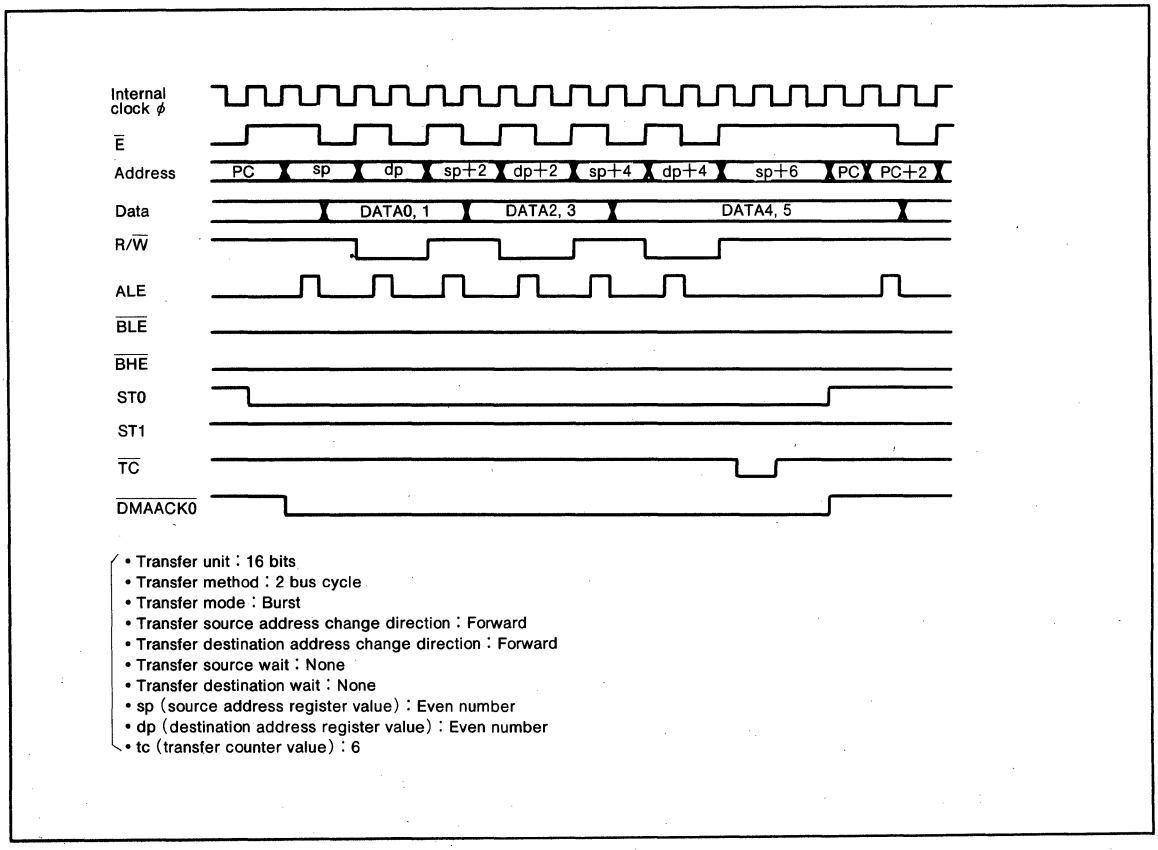


Fig.64 Timing diagram of normal transfer

(2) Repeat transfer

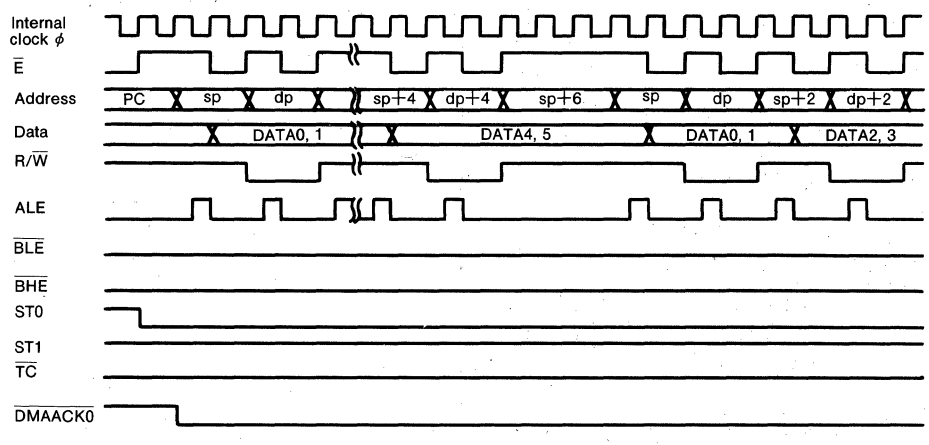
The term "repeat transfer" refers to DMA transfer in which normal transfer is repeated. First set up the number of unit transfer bits, transfer method, transfer mode, and transfer address change direction of the DMAi mode register. Next write the transfer source block transfer start address in the SAR and the transfer destination transfer start address in the DAR. Further, write the number of bytes to be transferred, into the TCR, and set up the DMAi control register and DMAC control register. The DMA request is now acceptable. When the DMA request is received in this state, DMA transfer starts. Even when the number of remaining bytes (to be read from the TCR) becomes 0, the DMA permission bit is not cleared. When the burst transfer function is activated, the DMA request flag is not cleared either. When the cycle steal transfer function is selected, the DMA request flag is cleared at the end of each one-unit transfer.

The first values written in the SAR, DAR, and TCR are retained in the internal latches. The contents of the latches are transferred to the SAR, DAR and TCR at the end of the last transfer cycle. Therefore, when the burst transfer function is selected, the transfer operation is repeated starting with the first written values. When the cycle steal transfer function is selected, on the other hand, these values are used as the initial values and transfer is initiated each time the DMA request flag is set to "1".

To terminate this type of transfer, enter the "L" level to the $\overline{TC}$ pin or write the value "0" in the DMA permission bit.

In repeat transfer, $\overline{TC}$ signal output and the setting to the interrupt request bit of the DMA interrupt control register are not performed.

Figure 65 shows the repeat transfer timing diagram.



- Transfer unit : 16 bits
- Transfer method : 2 bus cycle
- Transfer mode : Burst
- Transfer source address change direction : Forward
- Transfer destination address change direction : Forward
- Transfer source wait : None
- Transfer destination wait : None
- sp (source address register value) : Even number
- dp (destination address register value) : Even number
- tc (transfer counter value) : 6

Fig.65 Timing diagram of repeat transfer

(3) Array chain transfer

The term "array chain transfer" refers to the DMA transfer in which one channel is used for the data transfer for two or more memory blocks.

Three parameters necessary for transfer, that is, the transfer source start address, transfer destination start address, and the number of transfer bytes, must be sequentially written into the transfer parameter memory. The transfer parameter memory can be located in an arbitrary position in the memory space. Figure 66 shows an array chain transfer mode transfer parameter memory configuration example. All the transfer parameters of the memory block to be transferred must be written into the transfer parameter memory. The transfer parameter write format is indicated in Figure 67. For one bus cycle transfer, the I/O side parameters are not needed. For transfer from memory to I/O, for instance, consecutively write the transfer source start address and the number of transfer bytes only as shown in Figure 68. As the transfer destination start address needs not be written, it is possible to save on the transfer time and transfer parameter memory area.

In normal or repeat transfer, the first values written in the SAR, DAR, and TCR are retained in the internal latches. In array chain transfer and link array chain transfer, however, the latches perform different functions.

The SAR latch serves as the transfer parameter register (hereinafter referred to as the TPR) which indicates the first address of the transfer parameter memory. The TCR latch serves as the transfer block counter (hereinafter referred to as the TBC) which indicates the number of transfer blocks. In array chain transfer and link array chain transfer, writing a value at an SAR address causes that value to be written in the TPR, and writing a value at a TCR address causes that value to be written in the TBC.

The array chain transfer operations are detailed below.

As is the case with the other types of transfer, first set up the DMAi mode register, DMAi control register, and DMAC control register. Write the first address of the transfer parameter memory into the SAR. This value is then written into the TPR. Be sure that an even-numbered address is designated as the first address. Nothing needs to be written into the DAR. Into the TCR, on the other hand, write the number of memory blocks to be transferred. This number is then written into the TBC. When the DMA permission bit is set to "1" after completion of the above-mentioned setup, DMA transfer is enable.

In array chain transfer, the transfer parameters are first read from the transfer parameter memory and then written into the SAR, DAR, and TCR. This operations state is called the "array state". Figure 69 shows a two bus cycle transfer mode array state example. The DMA controller outputs the first address of the transfer parameter memory to the address bus, and sequentially stores the read data into the SAR, DAR, and TCR.

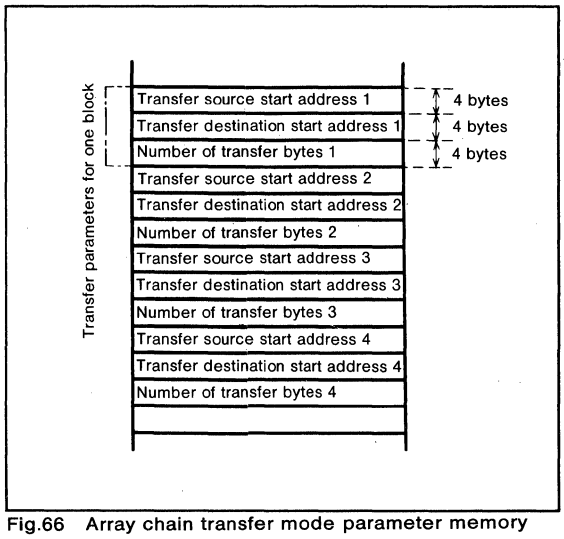


Fig.66 Array chain transfer mode parameter memory configuration example

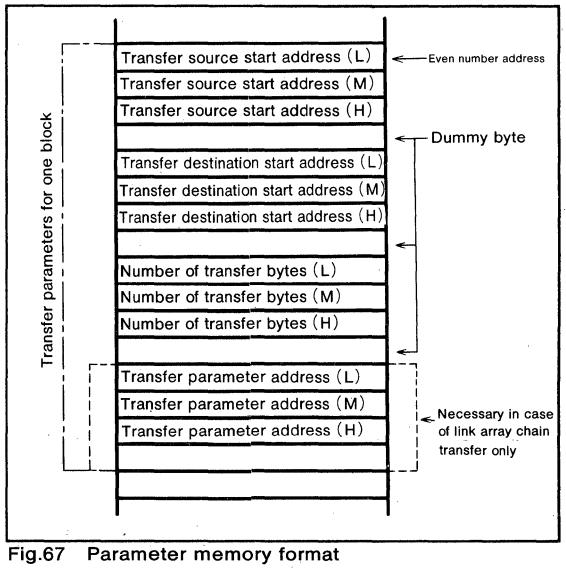


Fig.67 Parameter memory format

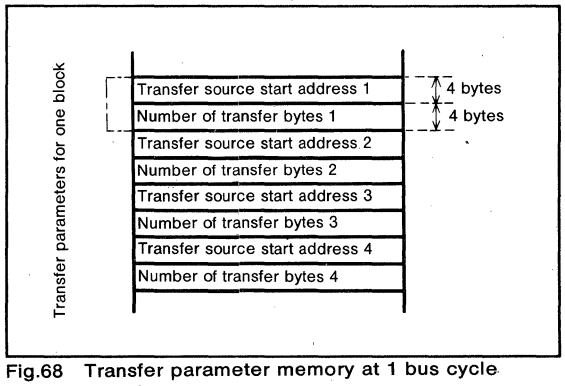


Fig.68 Transfer parameter memory at 1 bus cycle transfer

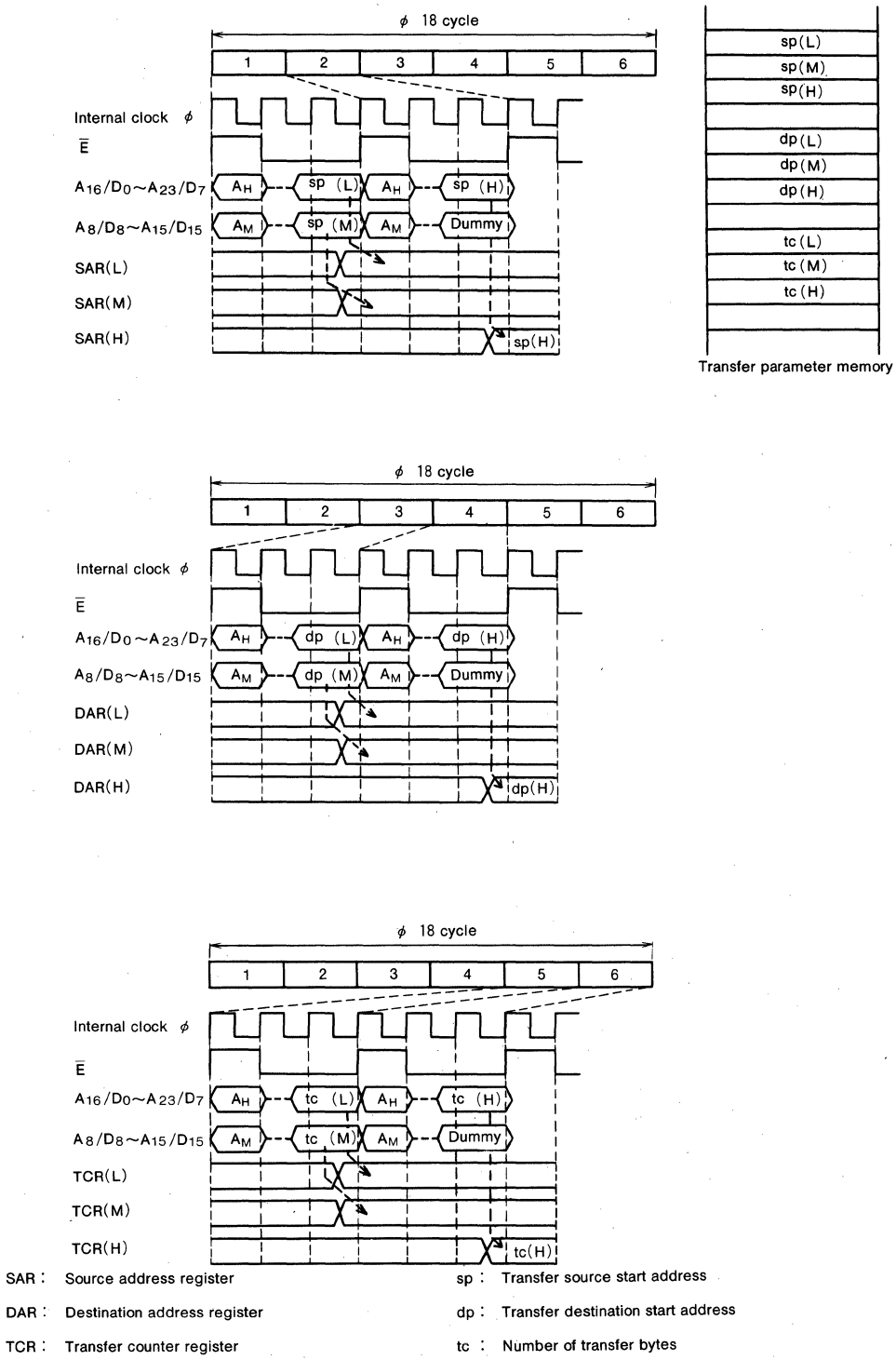
When the transfer parameters for one block are completely stored, the contents of the TBC are decremented by 1 and then the first DMA transfer starts in accordance with the stored parameters.

In contrast to the array state, the state in which DMA transfer is made is called the transfer state. In the transfer state, the same operations are performed as in normal transfer. At the end of each one-unit transfer, the contents of the transfer counter are decremented by 1 in 8-bit transfer or by 2 in 16-bit transfer. Even when the contents of the transfer counter are 0, the DMA request flag and DMA permission bit are not cleared and the array state of the next block occurs.

When the contents of the TBC are 0 at the beginning of the array state, the entire transfer operation is considered to be completed, and the "L" level is output to the $\overline{TC}$ pin to clear the DMA request flag and DMA permission bit and terminate array chain transfer. At the same time, the interrupt request bit of the DMA interrupt control register is set to "1".

In the cycle steal transfer at the array chain transfer mode, one array state and the transfer cycle of one-unit transfer are performed by one DMA request.

Figure 70 shows the array chain transfer timing diagram.



SAR : Source address register
DAR : Destination address register
TCR : Transfer counter register

sp : Transfer source start address
dp : Transfer destination start address
tc : Number of transfer bytes

Fig.69 Array state example at 2 bus cycle transfer

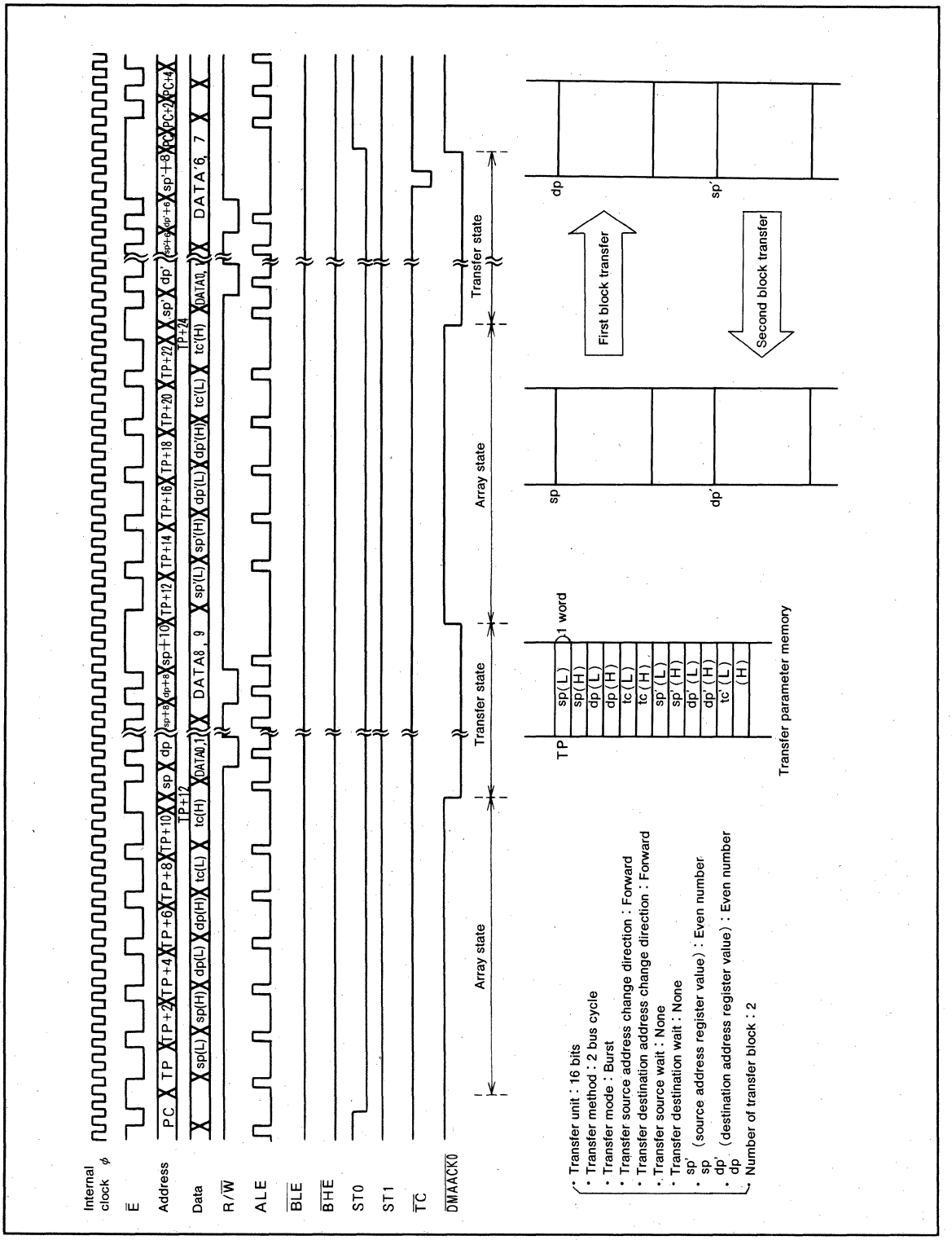


Fig.70 Array chain transfer timing diagram

(4) Link array chain transfer

Figure 71 shows the link array chain transfer parameter memory configuration. As shown in this figure, not only the transfer source start address, transfer destination start address, and number of transfer bytes, but also the first address of the memory block which contains the next transfer parameters is stored. In the transfer parameter of the last block, set "000000_{16}" as the first address of the next transfer parameter. In one bus cycle transfer, the I/O side parameters are not needed.

As is the case with the other types of transfer, the DMAi mode register, DMAi control register, and DMAC control register must be set up for link array chain transfer purposes. Into the SAR, write the first address of the memory block that stores the parameters for the first transfer. This address value is then written into the TPR. Be sure that an evennumbered address is designated as the first address. Nothing needs to be written in the DAR. Write a numerical value other than zero into the TCR. When the DMA permission bit is set to "1" after completion of the above setup, DMA transfer is enable.

In link array chain transfer, the transfer parameters are first read from the transfer parameter memory and then written into the SAR, DAR, and TCR. Further, the first address of the memory block that contains the next parameters is written into the TPR. In link array chain transfer, the state so far is referred to as the array state.

The DMA controller sequentially outputs to the address bus the addresses of the memory block that stores the transfer parameters, beginning with the first address. The read data are sequentially stored into the SAR, DAR, and TCR, and then the first address of the memory block containing the next parameters is written into the TPR. DMA transfer is made in accordance with the parameters read from the transfer parameter memory. The transfer state is the same as in normal transfer. The contents of the TCR are decremented by 1 or 2 at the end of each one-unit transfer. Even when the contents of the TCR are 0, the DMA request flag and DMA permission bit are not cleared to "0" but the array state starts again. When the contents of the TPR are 0 at this time, however, the "L" level is output to the $\overline{TC}$ pin to clear the DMA request flag and DMA permission bit and terminate link array chain transfer. At the same time, the interrupt request bit of the DMA interrupt control register is set to "1".

In the cycle steal transfer at the rink array chain transfer mode, one array state and the transfer cycle of one-unit transfer are performed by one DMA request.

Figure 72 shows the link array chain transfer timing diagram.

Note that the following operations cannot be performed in array chain transfer or link array chain transfer.

1. A 16-bit transfer operation performed with an external width of 8 bits employed and the fixed transfer address change direction selected for either the transfer source or destination
2. A 16-bit transfer operation performed with an external width of 16 bits employed, the fixed transfer address change direction selected, and an odd-numbered address designated as the first address for either the transfer source or destination.

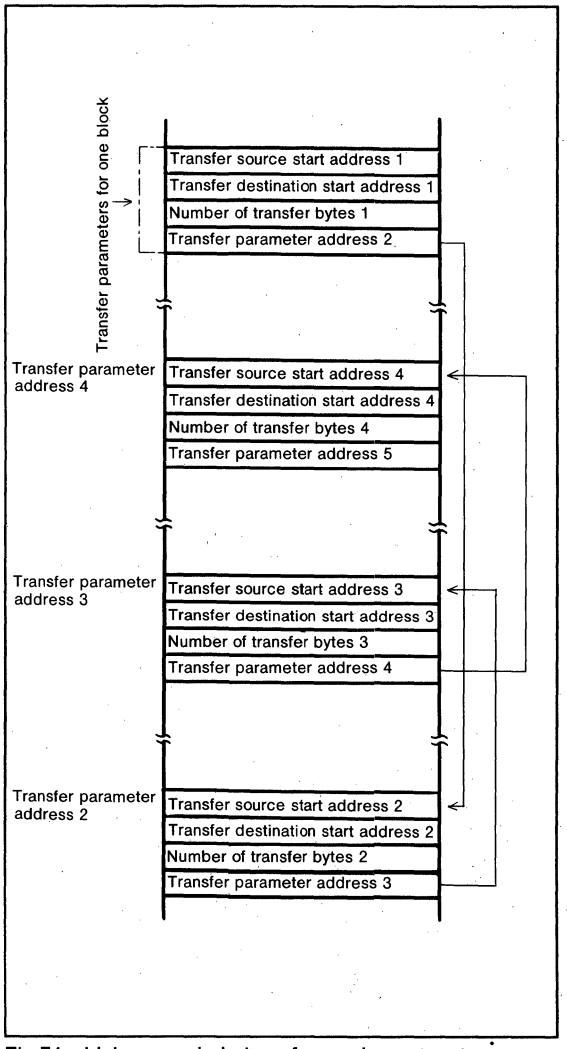


Fig.71 Link array chain transfer mode parameter memory configuration example

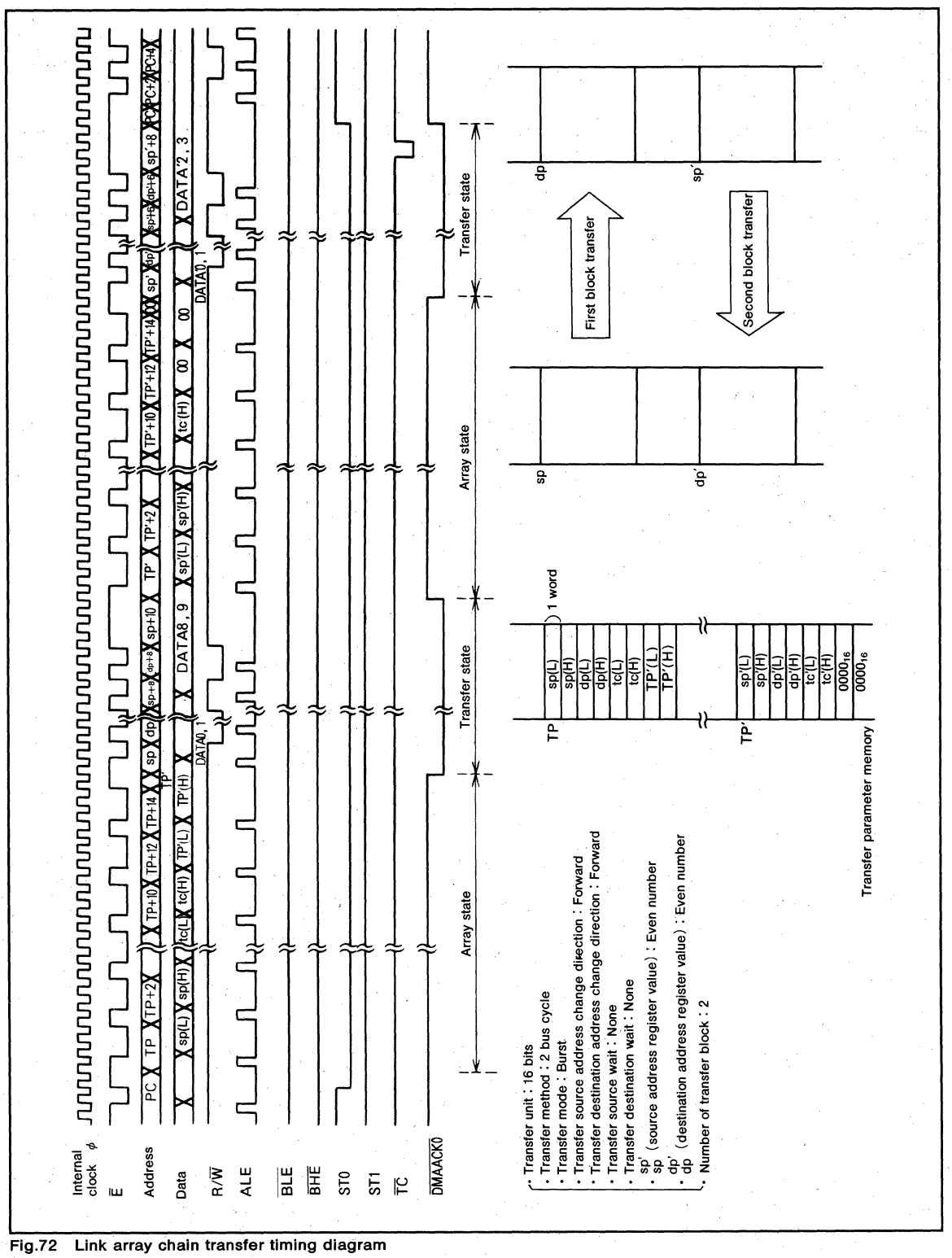


Fig.72 Link array chain transfer timing diagram

MITSUBISHI ELECTRIC

DRAM CONTROLLER

The DRAM controller directly controls the DRAM (dynamic random access memory). The DRAM control pins are located at ports $P10_4/\overline{CAS}$, $P10_5/\overline{RAS}$, $P10_6/MA_8$, $P10_7/MA_9$, and A_0/MA_0 through $A_7/MA7$.

Figure 73 shows the DRAM controller block diagram. Figure 74 shows the DRAM control register (address 64_{16}) bit configuration. Bit 7 of the DRAM control register is a DRAM validity bit. When this bit is set to "1", port $P10_4$ serves as the $\overline{CAS}$ (column address strobe) signal output pin, port 10_5 serves as the $\overline{RAS}$ (Row address strobe) signal output pin, port $P10_6$ serves as the MA_8 output pin, and $P10_7$ serves as the MA_9 output terminal. Upon resetting, the DRAM validity bit is set to "0". When using the DRAM controller, be sure to set this bit to "1".

Bits 0 through 3 of the DRAM control register are the DRAM area selection bits. These bits are used to designate the DRAM area in 1M-byte units. When the DRAM area selection bits (bits 3 through 0) are set to "0000", the DRAM controller concludes that there is no DRAM area. This state prevails upon resetting. When the bits are set to "0001", the DRAM area is set to 1M-byte and the system automatically judges that the area between addresses $F00000_{16}$ and $FFFFFF_{16}$ is provided as the DRAM area. When the bits are set to "1111", the DRAM area is set to 15M bytes and the DRAM controller automatically concludes that the area between addresses 100000_{16} and $FFFFFF_{16}$ is the DRAM area.

When the area to be accessed is judged to the DRAM area, the DRAM controller generates timing signals $\overline{RAS}$ and $\overline{CAS}$ for latching the address in the bus cycle, and outputs addresses A_0 through A_{20} to MA_0 (A_0 pin) through MA_7 (A_7 pin), MA_8, and MA_9 by means of time division multiplexing. The time division multiplexing method varies with the external bus connection. Table 7 shows the address time-division multiplexing DRAM controller for DRAM accessing.

It should also be noted that when the DRAM area is accessed, the "L" level period of $\overline{E}$ is increased twofold without respect to the contents of the wait bit. This also holds true when the DRAM area is accessed by the CPU or DMA controller.

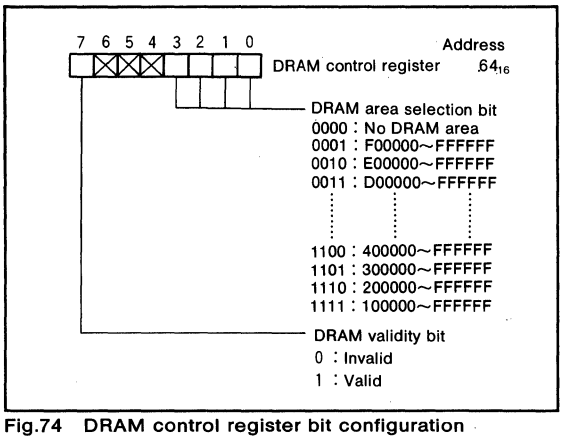


Fig.74 DRAM control register bit configuration

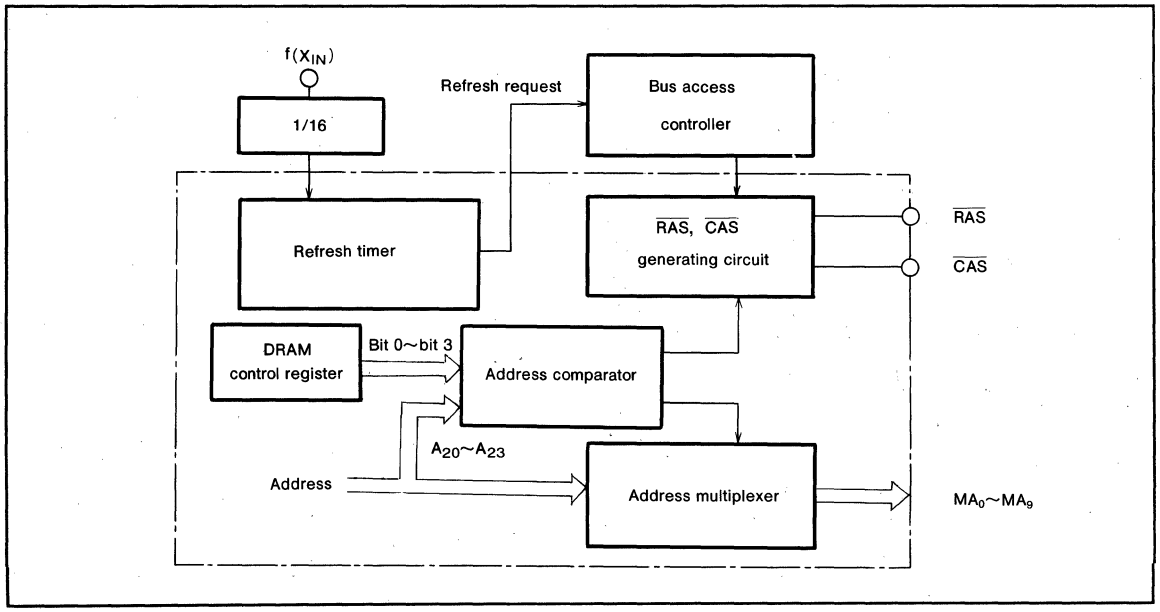


Fig.73 DRAM controller block diagram

The DRAM controller incorporates an independent 8-bit refresh timer. The refresh timer is based on decrement count and its clock source employs a clock that is derived by dividing $f(X_{IN})$ by sixteen. The refresh interval can be given as desired by writing a value (not "0") in the refresh timer (address 66_{16}). The value must be set in the refresh timer before the DRAM validity bit is set to "1". Generally, the value written in the refresh timer is expressed by the following formula.

Timer value n=refresh interval [ns] $\times$ $f(X_{IN})/16-1$

When the contents of the refresh timer is 00_{16}, the refresh timer generates the refresh request signal. The refresh request signal sends the bus use request to the bus access controller. When the request is accepted, the system stops the CPU and DMA controller and sends the response signal to the $\overline{RAS}/\overline{CAS}$ generator circuit. The $\overline{RAS}/\overline{CAS}$ generator circuit generates the refresh cycle.

If the CPU or DMA controller is using the data bus, refresh cycle generation does not take place even when the refresh request is made. When a WIT or STP instruction is executed, the refresh operation is not performed. Figure 75 shows an example refresh operation that is performed when the refresh request is made during DMA transfer. Figure 76 shows an example refresh operation that is performed when the refresh request is made during CPU operation (bus accessing). Both examples represent the cases where the response delay time is maximized. Table 8 shows the delay time that denotes the interval between the instant at which the contents of the refresh timer become 00_{16} and the instant at which the refresh cycle is inserted. This table indicates the maximum and minimum values prevailing when the $\overline{RDY}$ input is not provided. If the access time is to be increased by $\overline{RDY}$ input, the time delay resulting from such an access time increase must be added to the maximum values in the table.

Figure 77 shows the DRAM accessing bus cycle timing diagram. The DRAM controller generates three types of bus cycles: read, write, and refresh bus cycles.

In the read cycle, the $\overline{RAS}$ pin level falls from "H" to "L", and then the $\overline{CAS}$ pin level falls from "H" to "L" at the end of 1/2 the internal clock cycle ϕ. The address bus changes from the row address to the column address during the time interval between $\overline{RAS}$ pin level fall and $\overline{CAS}$ pin level fall.

In the write cycle, the $\overline{CAS}$ pin level falls one internal clock cycle ϕ after $\overline{RAS}$ pin level fall.

In the refresh cycle, the $\overline{CAS}$ pin level falls 1/2 internal clock cycle ϕ earlier than $\overline{RAS}$ pin level fall. Refresh occurs as the $\overline{CAS}$ pin level falls earlier than the $\overline{RAS}$ pin level. This method is called the "$\overline{CAS}$ before $\overline{RAS}$ refresh method". Be sure that the selected DRAM permits "$\overline{CAS}$ before $\overline{RAS}$ refresh".

Table 7. DRAM accessing address time-division multiplexing method

	Regular bus	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	$P10_6$	$P10_7$
	DRAM access	MA_0	MA_1	MA_2	MA_3	MA_4	MA_5	MA_6	MA_7	MA_8	MA_9
External 8-bit	Row address	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_{16}	A_{18}
	Column address	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{17}	A_{19}
External 16-bit	Row address	A_{16}	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_{18}	A_{20}
	Column address	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{17}	A_{19}

Table 8. Refresh wait time

Cause of bus use	Number of delay cycles	At 16MHz	At 8MHz
CPU	1.5~6.5	187.5ns~812.5ns (wait provided) 187.5ns~562.5ns (wait not provided)	375ns~1625ns (wait provided) 375ns~1125ns (wait not provided)
DMA controller	1.5~12.5	187.5ns~1562.5ns (wait provided) 187.5ns~1062.5ns (wait not provided)	375ns~3125ns (wait provided) 375ns~2125ns (wait not provided)
$\overline{HOLD}$ state	1.5	187.5ns	375ns

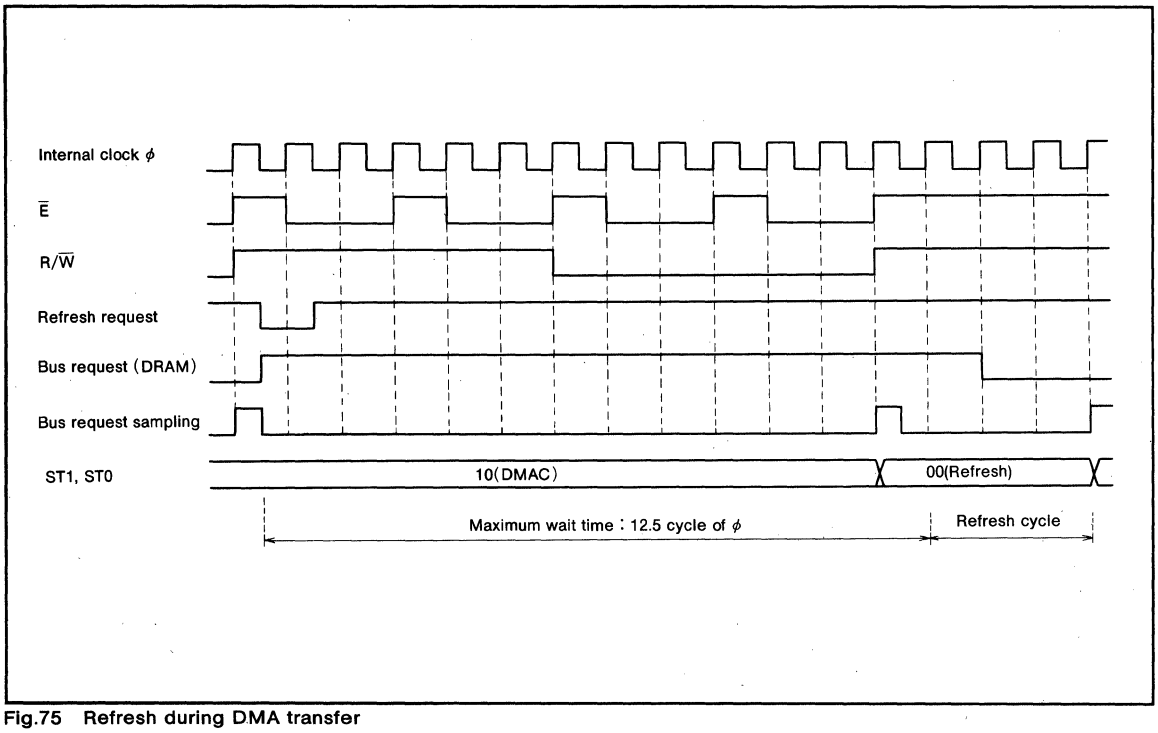


Fig.75 Refresh during DMA transfer

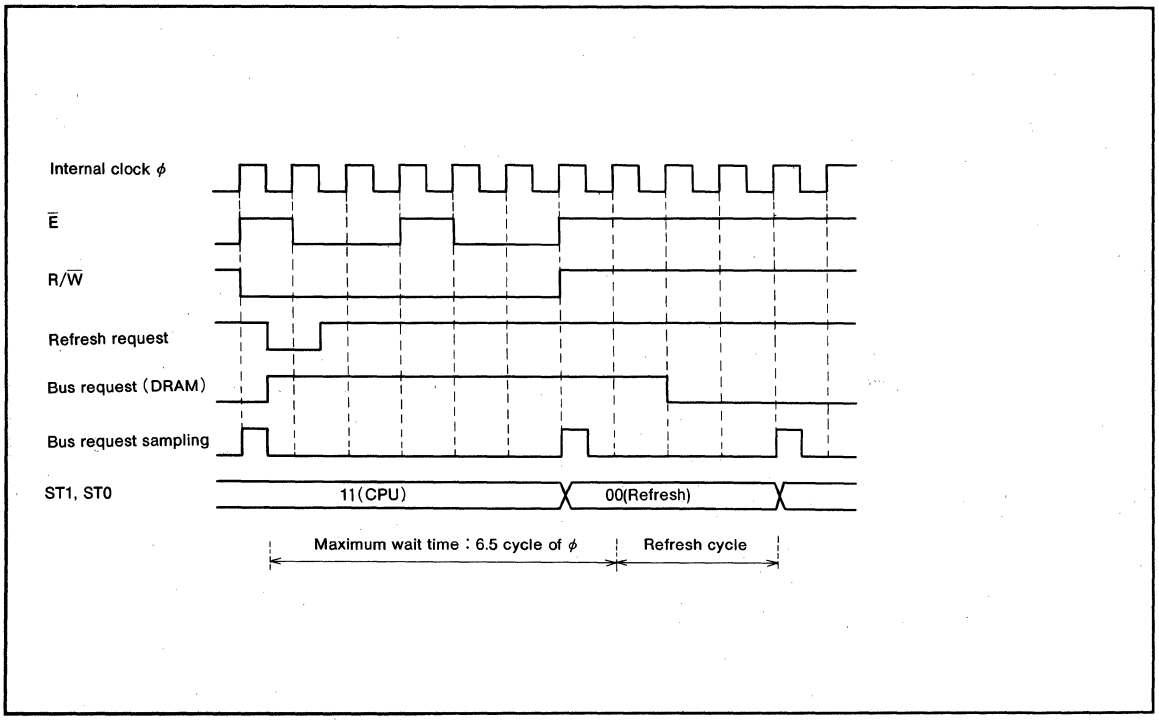
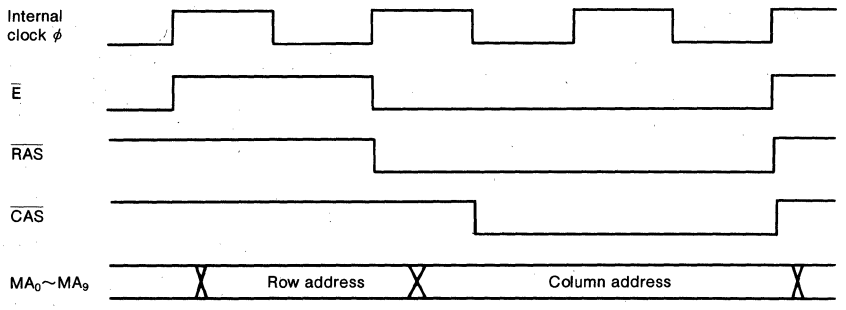
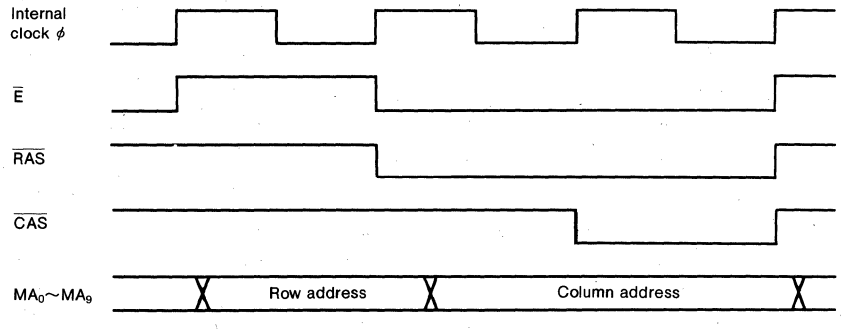


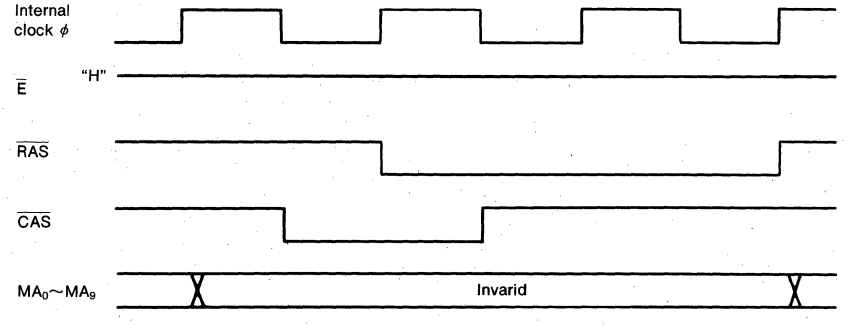
Fig.76 Refresh during CPU operation



(1) Timing diagram at reading



(2) Timing diagram at writing



(3) Timing diagram at refreshing

Fig.77 DRAM access timing diagram

REAL-TIME OUTPUT

The M37720S1FP has a 4-bit, 2-channel real-time output function which is shared with port P6 as the output port. As with the other ports, the port P6 direction register is in the input state after resetting and therefore functions as a regular I/O port. To use the real-time output function, be sure to set the port P6 direction register for output. Ports $P6_0$ through $P6_3$ correspond to channel 0, whereas ports 6_4 through 6_7 correspond to channel 1.

Figure 78 shows the real-time output control register (address 62_{16}) bit configuration. Bits 0 and 1 are the real-time output validity bits for the channels. Bit 0 relates to channel 0, and bit 1 relates to channel 1. When the value "1" is written in the positions of these bits, the real-time output function is enabled. These bits are "0" upon resetting.

Figure 80 shows the real-time output block diagram. In the real-time output mode, the values written in the pulse output data register are not output immediately. However, such an output is delivered to the pin when timer A0 (channel 0) or timer A1 (channel 1) underflows. Before underflow occurrence, the previously written data are output.

Figure 79 shows the pulse output data register bit configuration. When the real-time output 0 validity bit is "0", the data written in the port P6 latch (address E_{16}) is output to ports $P6_0$ through $P6_3$. When the real-time output 0 validity bit is "1", the values written in the bit 0 to bit 3 positions of pulse output data register 0 (address $1A_{16}$) are output to ports $P6_0$ ($RTP0_0$) through $P6_3$ ($RTP0_3$), respectively.

When the real-time output 1 validity bit is "0", the values written for bits 4 through 7 of the port P6 latch (address E_{16}) are output to ports P64 through $P6_7$, respectively. When the real-time output 1 validity bit is "1", on the other hand, the values written in pulse output data register 1 (address $1C_{16}$) are output to ports $P6_4$ ($RTP1_0$) through $P6_7$ (RTP_{13}).

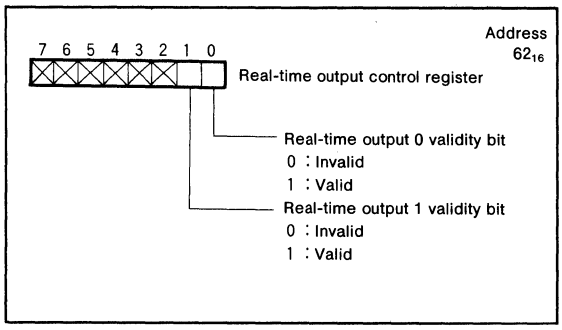


Fig.78 Real-time output control register bit configuration

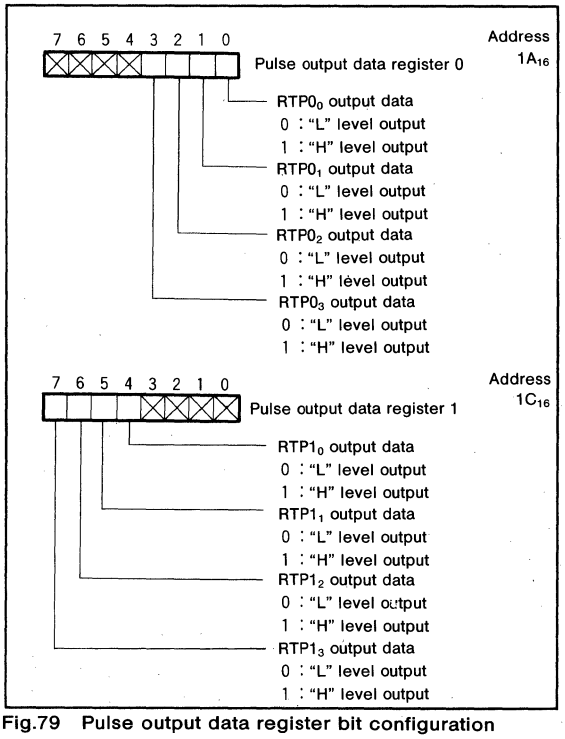


Fig.79 Pulse output data register bit configuration

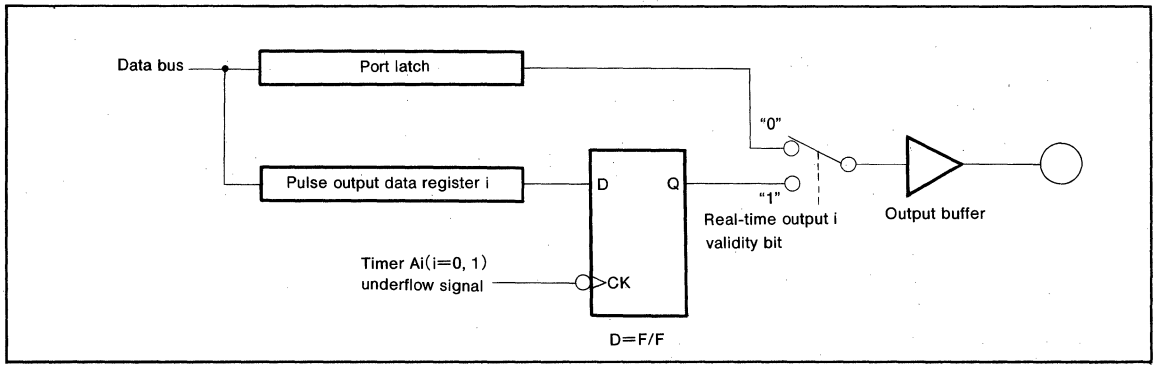


Fig.80 Real-time output block diagram

WATCHDOG TIMER

The watchdog timer is used to detect unexpected execution sequence caused by software run-away. Figure 81 shows a block diagram of the watchdog timer.

The watchdog timer consists of a 12-bit binary counter.

The watchdog timer counts the clock frequency divided by 32 (f_{32}) or by 512 (f_{512}). Whether to count f_{32} or f_{512} is determined by the watchdog timer frequency selection flag shown in Figure 82. f_{512} is selected when the flag is "0" and f_{32} is selected when it is "1". The flag is cleared after reset.

FFF_{16} is set in the watchdog timer when "L" or $2V_{CC}$ is applied to the $\overline{RESET}$ pin, STP instruction is executed, data is written to the watchdog timer, or the most significant bit of the watchdog timer become "0".

After FFF_{16} is set in the watchdog timer, the contents of watchdog timer is decremented by one at every cycle of selected frequency f_{32} or f_{512}, and after 2048 counts, the most significant bit of watchdog timer become "0", and a watchdog timer interrupt request bit is set, and FFF_{16} is preset in the watchdog timer.

Normally, a program is written so that data is written in the watchdog timer before the most significant bit of the watchdog timer become "0". If this routine is not executed due to unexpected program execution, the most significant bit of the watchdog timer become eventually "0" and an interrupt is generated.

The processor can be reset by setting the bit 3 (software reset bit) of processor mode register described in Figure 10 in the interrupt section and generating a reset pulse.

The watchdog timer stops its function when the $\overline{RESET}$ pin voltage is raised to double the V_{CC} voltage.

The watchdog timer can also be used to recover from when the clock is stopped by the STP instruction. Refer to the section on clock generation circuit for more details.

The watchdog timer is also in the hold state during hold, DMA transfer, or refresh. Therefore, the input to the watchdog timer is prohibited.

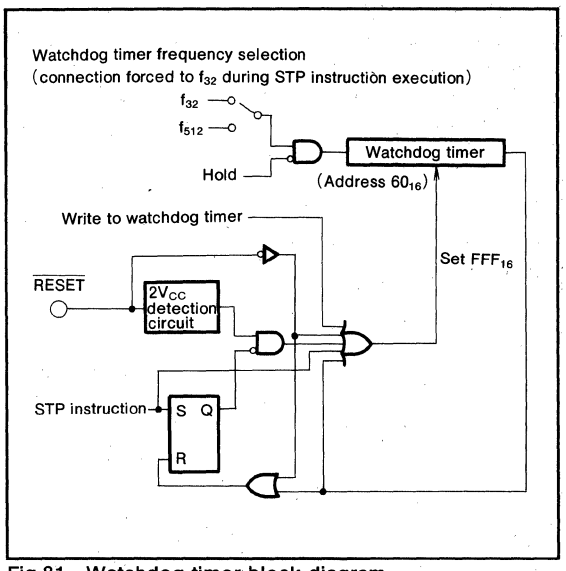


Fig.81 Watchdog timer block diagram

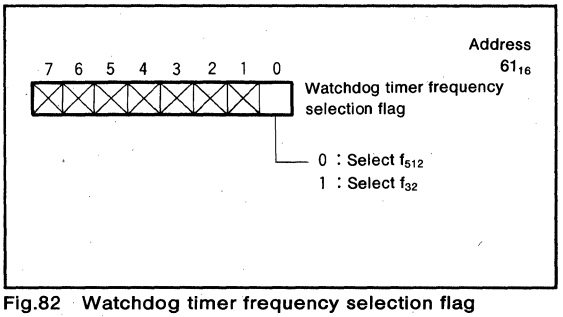


Fig.82 Watchdog timer frequency selection flag

RESET CIRCUIT

Reset occurs when the $\overline{\text{RESET}}$ pin is returned to "H" level after holding it at "L" level when the power voltage is at 5V ±10%. Program execution starts at the address formed by setting the address pins A_{23} ~ A_{16} to 00_{16}, A_{15} ~ A_8 to the contents of address $FFFF_{16}$, and A_7 ~ A_0 to the contents of address $FFFE_{16}$.

Figure 83 shows the status of the internal registers when a reset occurs.

(1)	Port P4 data direction register	($0C_{16}$)···	0 0 0 0 0 X X X
(2)	Port P5 data direction register	($0D_{16}$)···	00_{16}
(3)	Port P6 data direction register	(10_{16})···	00_{16}
(4)	Port P7 data direction register	(11_{16})···	00_{16}
(5)	Port P8 data direction register	(14_{16})···	00_{16}
(6)	Port P9 data direction register	(15_{16})···	00_{16}
(7)	Port P10 data direction register	(18_{16})···	00_{16}
(8)	A-D control register	($1E_{16}$)···	0 0 0 0 0 ? ? ?
(9)	A-D sweep pin selection register	($1F_{16}$)···	X X X X X X 1 1
(10)	UART0 Transmit/Receive mode register	(30_{16})···	00_{16}
(11)	UART1 Transmit/Receive mode register	(38_{16})···	00_{16}
(12)	UART0 Transmit/Receive control register 0	(34_{16})···	X X X X 1 0 0 0
(13)	UART1 Transmit/Receive control register 0	($3C_{16}$)···	X X X X 1 0 0 0
(14)	UART0 Transmit/Receive control register 1	(35_{16})···	0 0 0 0 0 0 1 0
(15)	UART1 Transmit/Receive control register 1	($3D_{16}$)···	0 0 0 0 0 0 1 0
(16)	Count start flag	(40_{16})···	00_{16}
(17)	One-shot start flag	(42_{16})···	X X X 0 0 0 0 0
(18)	Up-down flag	(44_{16})···	00_{16}
(19)	Timer A0 mode register	(56_{16})···	00_{16}
(20)	Timer A1 mode register	(57_{16})···	00_{16}
(21)	Timer A2 mode register	(58_{16})···	00_{16}
(22)	Timer A3 mode register	(59_{16})···	00_{16}
(23)	Timer A4 mode register	($5A_{16}$)···	00_{16}
(24)	Timer B0 mode register	($5B_{16}$)···	0 0 1 X 0 0 0 0
(25)	Timer B1 mode register	($5C_{16}$)···	0 0 1 X 0 0 0 0
(26)	Timer B2 mode register	($5D_{16}$)···	0 0 1 X 0 0 0 0
(27)	Processor mode register	($5E_{16}$)···	0 0 0 0 0 0 1 0
(28)	Watchdog timer	(60_{16})···	FFF_{16}
(29)	Watchdog timer frequency selection flag	(61_{16})···	X X X X X X X 0
(30)	Real-time output control register	(62_{16})···	X X X X X X 0 0
(31)	DRAM control register	(64_{16})···	0 X X X 0 0 0 0
(32)	DMAC control register L	(68_{16})···	0 0 0 0 X X 0 0
(33)	DMAC control register H	(69_{16})···	00_{16}
(34)	DMA0 interrupt control register	($6C_{16}$)···	X X X X 0 0 0 0
(35)	DMA1 interrupt control register	($6D_{16}$)···	X X X X 0 0 0 0
(36)	DMA2 interrupt control register	($6E_{16}$)···	X X X X 0 0 0 0
(37)	DMA3 interrupt control register	($6F_{16}$)···	X X X X 0 0 0 0
(38)	A-D conversion interrupt control register	(70_{16})···	X X X 0 0 0 0 0
(39)	UART 0 transmission interrupt control register	(71_{16})···	X X X 0 0 0 0 0
(40)	UART0 receive interrupt control register	(72_{16})···	X X X 0 0 0 0 0
(41)	UART1 transmission interrupt control register	(73_{16})···	X X X 0 0 0 0 0
(42)	UART1 receive interrupt control register	(74_{16})···	X X X 0 0 0 0 0
(43)	Timer A0 interrupt control register	(75_{16})···	X X X 0 0 0 0 0
(44)	Timer A1 interrupt control register	(76_{16})···	X X X 0 0 0 0 0
(45)	Timer A2 interrupt control register	(77_{16})···	X X X 0 0 0 0 0
(46)	Timer A3 interrupt control register	(78_{16})···	X X X 0 0 0 0 0
(47)	Timer A4 interrupt control register	(79_{16})···	X X X 0 0 0 0 0
(48)	Timer B0 interrupt control register	($7A_{16}$)···	X X X 0 0 0 0 0
(49)	Timer B1 interrupt control register	($7B_{16}$)···	X X X 0 0 0 0 0
(50)	Timer B2 interrupt control register	($7C_{16}$)···	X X X 0 0 0 0 0
(51)	$\overline{\text{INT}_0}$ interrupt control register	($7D_{16}$)···	X X 0 0 0 0 0 0
(52)	$\overline{\text{INT}_1}$ interrupt control register	($7E_{16}$)···	X X 0 0 0 0 0 0
(53)	$\overline{\text{INT}_2}$ interrupt control register	($7F_{16}$)···	X X 0 0 0 0 0 0
(54)	DMA0 mode register L	($1FCC_{16}$)···	00_{16}
(55)	DMA0 mode register H	($1FCD_{16}$)···	00_{16}
(56)	DMA0 control register	($1FCE_{16}$)···	X X 0 0 0 0 0 0
(57)	DMA1 mode register L	($1FDC_{16}$)···	00_{16}
(58)	DMA1 mode register H	($1FDD_{16}$)···	00_{16}
(59)	DMA1 control register	($1FDE_{16}$)···	X X 0 0 0 0 0 0
(60)	DMA2 mode register L	($1FEC_{16}$)···	00_{16}
(61)	DMA2 mode register H	($1FED_{16}$)···	00_{16}
(62)	DMA2 control register	($1FEE_{16}$)···	X X 0 0 0 0 0 0
(63)	DMA3 mode register L	($1FFC_{16}$)···	00_{16}
(64)	DMA3 mode register H	($1FFD_{16}$)···	00_{16}
(65)	DMA3 control register	($1FFE_{16}$)···	X X 0 0 0 0 0 0
(66)	Processor status register PS		0 0 0 ? ? 0 0 0 1 ? ?
(67)	Program bank register PG		00_{16}
(68)	Program counter PC_H		Contents of address $FFFF_{16}$
(69)	Program counter PC_L		Contents of address $FFFE_{16}$
(70)	Direct page register DPR		0000_{16}
(71)	Data bank register DT		00_{16}

Contents of other registers and RAM are not initialized and should be initialized by software.

Fig.83 Microcomputer internal status during reset

Figure 84 shows an example of a reset circuit. The reset input voltage must be held 0.9V or lower when the power voltage reaches 4.5V.

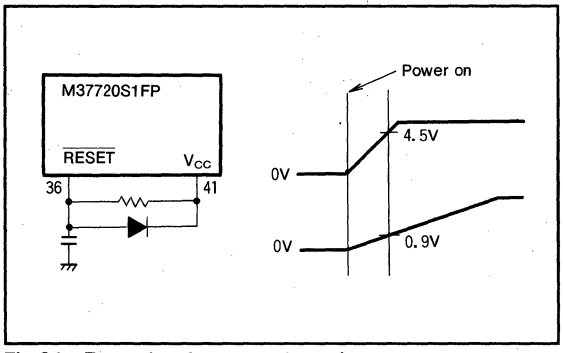


Fig.84 Example of a reset circuit (perform careful evaluation at the system design before using)

INPUT/OUTPUT PINS

Ports P10 to P4 all have a data direction register and each bit can be programmed for input or output. A pin becomes an output pin when the corresponding data direction register is set and an input pin when it is cleared.

When pin programmed for output, the data is written to the port latch and it is output to the output pin. When a pin is programmed for output, the contents of the port latch is read instead of the value of the pin. Therefore, a previously output value can be read correctly even when the output "L" voltage is raised due to reasons such as directly driving an LED.

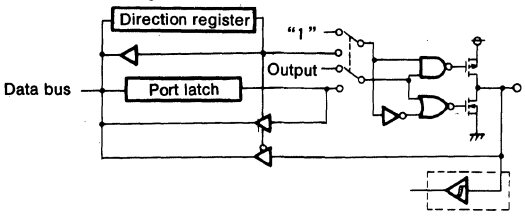
A pin programmed for input is floating and the value input to the pin can be read. When a pin is programmed for input, the data is written only in the port latch and the pin stays floating.

If an input/output pin is not used as an output port, clear the bit of the corresponding data direction register so that the pin become input mode.

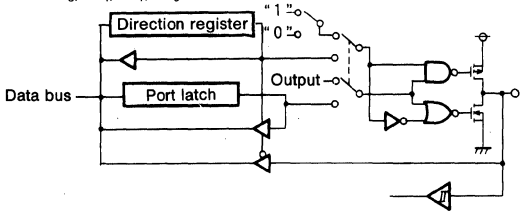
Figure 85 presents the port P10 to port P4 block diagrams and $\overline{E}$ pin output format.

In the evaluation chip mode, port P4 is also used as the control signal terminal. Refer to the section on evaluation chip mode for more details.

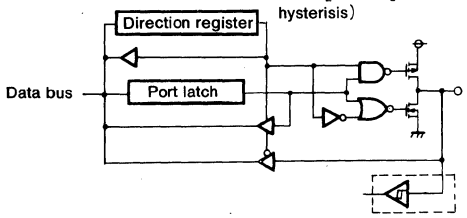
• Ports $P8_3$, $P8_7$, $P9_6$, $P9_4$, $P9_2$, $P9_0$, $P10_7$, $P10_6$, $P10_5$, $P10_4$ (Inside dotted-line not included)
Ports $P5_4$, $P5_2$, $P5_0$ (Inside dotted-line included)



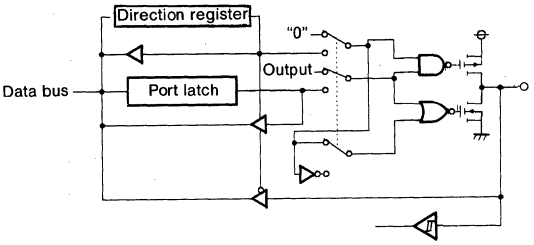
• Ports $P8_5$, $P8_4$, $P8_1$, $P8_0$



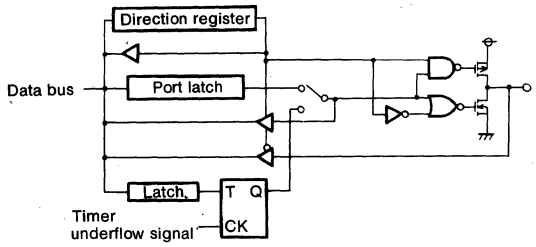
• Ports $P4_6$~$P4_3$ (Inside dotted-line not included)
Ports $P4_7$, $P5_7$, $P5_6$, $P5_5$, $P5_3$, $P5_1$, $P8_6$, $P8_2$, $P9_7$, $P9_5$, $P9_3$, $P9_1$, $P10_2$, $P10_1$, $P10_0$ (Inside dotted-line included, but $P8_2$ and $P8_6$ are without hysterisis)



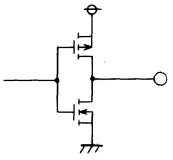
• Port $P10_3$



• Ports $P6_7$~$P6_0$



• $\overline{E}$



• Ports $P7_6$~$P7_0$ (Inside dotted-line not included)
Port $P7_7$ (Inside dotted-line included)

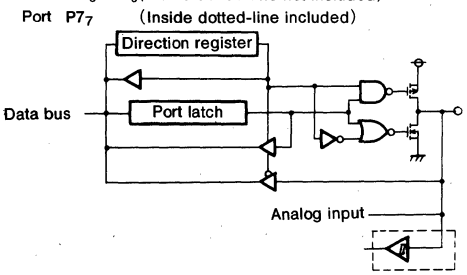


Fig.85 Block diagram for ports P10 to P4 in microprocessor mode and the $\overline{E}$ pin output

PROCESSOR MODE

Either the microprocessor mode or evaluation chip mode is selected in accordance with the contents of bit 0 of the processor mode register shown in Figure 86. Bit 1 must always be set to 1.

Figure 87 shows the port P4, A_7/MA_7 to A_0/MA_0, A_{15}/D_{15} to A_8/D_8, A_{23}/D_7 to A_{16}/D_0, R/$\overline{W}$, $\overline{BHE}$, $\overline{BLE}$, ALE, $\overline{RDY}$, $\overline{HOLD}$, and ϕ_1 functions in the two different processor modes.

The external memory area varies with the mode. Figure 88 presents the memory map in each processor mode. When the external memory is to be accessed, it is influenced by the BYTE pin and processor mode register bit 2 (wait bit) as explained below.

BYTE pin

When accessing the external memory, the level of the BYTE pin is used to determine whether to use the data bus as 8-bit width or 16-bit width.

The data bus width is 8 bits when the level of the BYTE pin is "H".

The data bus width is 16 bits when the level of the BYTE pin is "L".

When accessing the internal memory, the data bus width is always 16 bits regardless of the BYTE pin level.

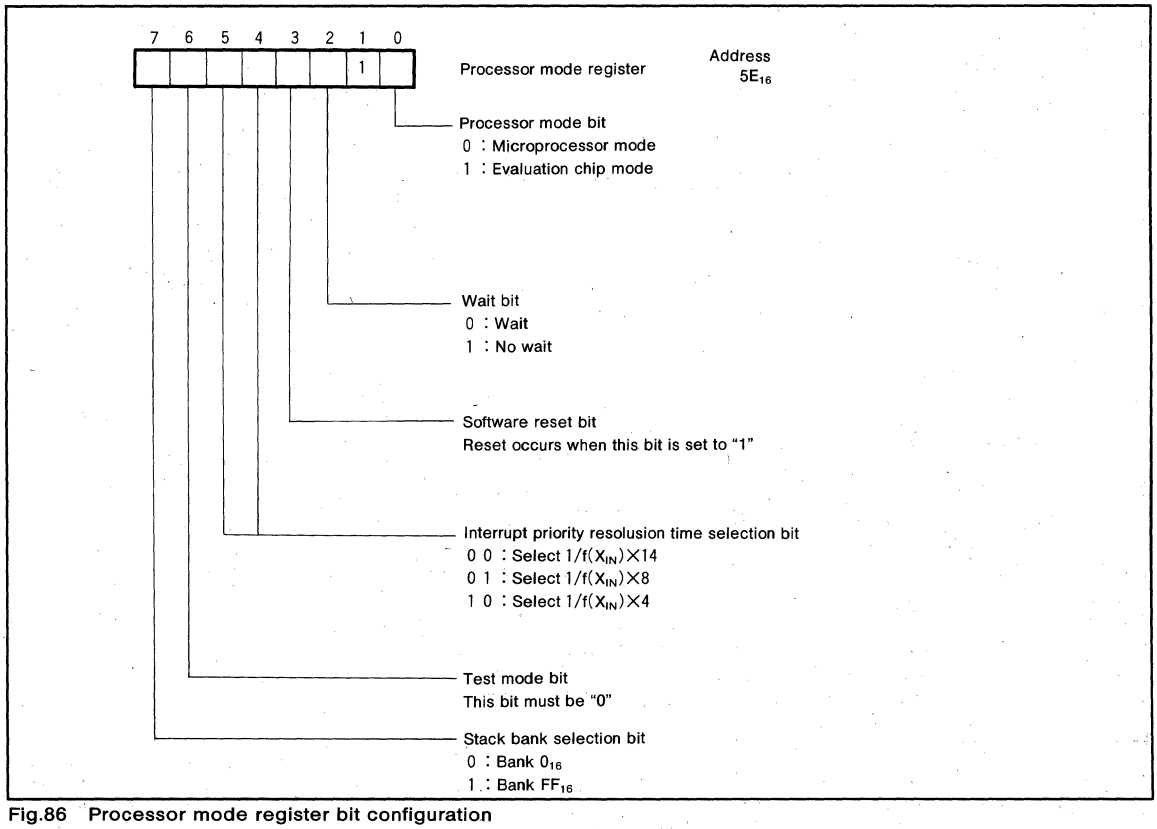


Fig.86 Processor mode register bit configuration

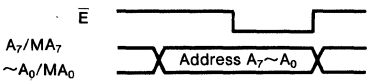
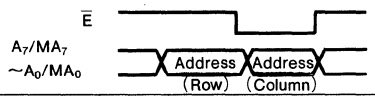
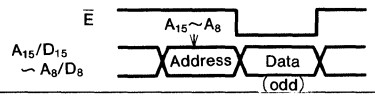
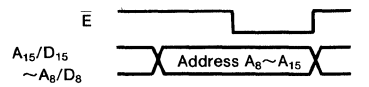
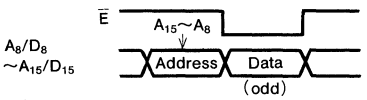
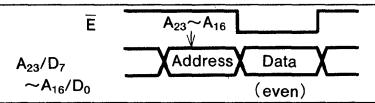
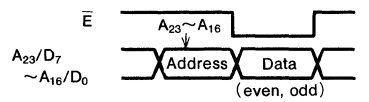
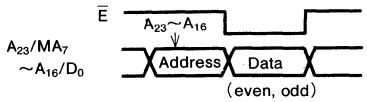
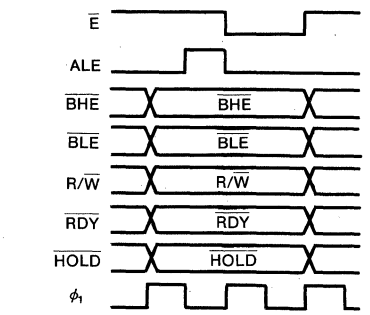
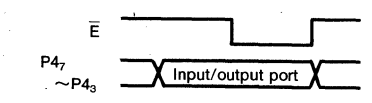
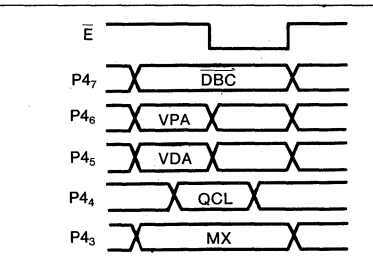
Port \ Mode		CM1=1, CM0=0: Microprocessor mode	CM1=1, CM0=1: Evaluation chip mode
A_7/MA_7 ≀ A_0/MA_0	Access except DRAM		Same as left
	Access DRAM		Same as left
A_{15}/D_{15} ≀ A_8/D_8	BYTE="L"		Same as left
	BYTE="H"		 Port P4, P5 and their direction registers are treated as 16-bit bus width.
A_{23}/D_7 ≀ A_{16}/D_0	BYTE="L"		Same as left
	BYTE="H"		 Same as for A_{15}/D_{15}~A_8/D_8
ALE $\overline{BHE}$ $\overline{BLE}$ R/$\overline{W}$ $\overline{RDY}$ $\overline{HOLD}$ ϕ_1			Same as left
Port P4			

Fig.87 Processor mode and ports P4, A_7/MA_7~A_0/MA_0, A_{15}/D_{15}~A_8/D_8, A_{23}/D_7~A_{16}/D_0, and ALE, R/$\overline{W}$, $\overline{BHE}$, $\overline{BLE}$, $\overline{RDY}$, $\overline{HOLD}$, ϕ1 functions

Wait bit

As shown in Figure 89, when the external memory area is accessed with the processor mode register bit 2 (wait bit) cleared to "0", the "L" width of $\overline{E}$ signal is becomes twice compared with no wait (the wait bit is "1"). The wait bit is cleared during reset.

When the CPU is operating, the accessing of internal memory area is performed in no wait mode regardless of the wait bit.

The processor modes are described below.

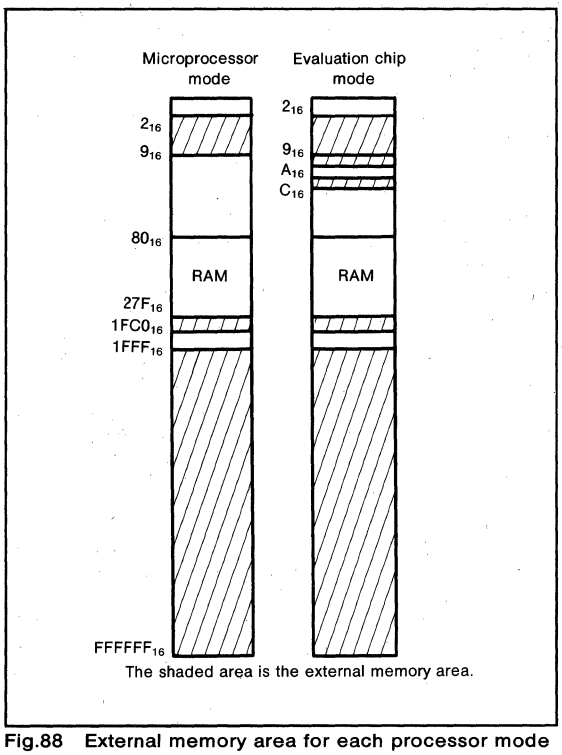


Fig.88 External memory area for each processor mode

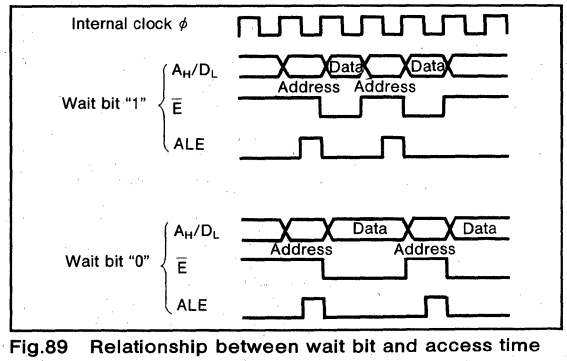


Fig.89 Relationship between wait bit and access time

(1) Microprocessor mode [0]

When the system is reset with the CNV_{SS} pin connected to V_{SS} or V_{CC}, this mode is selected.

Ports A_{15}/D_{15} through A_8/D_8 may be in either of the following two conditions depending on the BYTE pin level.

When the BYTE pin level is "L", the ports serve

as the address output terminals while $\overline{E}$ is at the "H" level or

as the odd-numbered address data I/O terminals while $\overline{E}$ is at the "L" level.

However, if the internal memory area is read, external data is ignored from outside while $\overline{E}$ is at the "L" level.

When the BYTE pin level is "H", the ports serve as the address output terminals.

Ports A_{23}/D_7 through A_{16}/D_0 may be in either of the following two conditions depending on the BYTE pin level.

When the BYTE pin level is "L", the ports serve

as the address output terminals while $\overline{E}$ is at the "H" level or

as the even-numbered address data I/O terminals while $\overline{E}$ is at the "L" level.

However, if the internal memory area is read, external data is ignored from outside while $\overline{E}$ is at the "L" level.

When the BYTE pin level is "H", the ports serve

as the address output terminals while $\overline{E}$ is at the "H" level or

as the even- and odd-numbered data I/O terminals while $\overline{E}$ is at the "L" level.

However, if the internal memory area is read, external data is ignored from outside while $\overline{E}$ is at the "L" level.

R/$\overline{W}$ is the read/write signal. It performs a read operation when it is at the "H" level or a write operation when it is at the "L" level.

$\overline{BHE}$ is the byte high enable signal. When it is at the "L" level, it indicates that an odd-numbered address is being accessed. $\overline{BLE}$ is the byte low enable signal. When it is at the "L" level, it indicates that a even-numbered address is being accessed. Therefore, if $\overline{BLE}$ is at the "L" level and $\overline{BHE}$ is at the "L" level, it means that two bytes at an odd-numbered address and an even-numbered address are being accessed simultaneously.

ALE is the address latch enable signal. It is used for latching the address signal from a multiplexed signal containing addresses and data. When ALE is at the "H" level, the latch opens to pass off the address signal. When ALE is at the "L" level, it retains the address signal.

$\overline{HOLD}$ is the hold request signal. It is an input signal used to hold the microcomputer. $\overline{HOLD}$ input is accepted when the internal clock ϕ falls from "H" level to "L" level while the bus is not used. ST1 and ST0 notify externally when microcomputer enters into hold state. In the hold state, ST1 = 0, ST0 = 1 and A_7/MA_7 through A_0/MA_0, A_{15}/D_{15} through A_8/D_8, A_{23}/D_7 through A_{16}/D_0, R/$\overline{W}$, $\overline{BHE}$, and $\overline{BLE}$ are floating. These ports are floating after one cycle of the in-

ternal clock ϕ later than the status signal changes to [ST1 =0, ST0=1] . At the removing of hold state, these ports are removed from floating state after one cycle of internal clock ϕ later than the status signal changes to the value except [ST1=0, ST0=1] .

$\overline{RDY}$ is the ready signal. When it is at the "L" level, the internal clock ϕ stops at the "L" level. The instruction execution of CPU and the bus access of the bus interface unit are also stopped. The clock ϕ_1 output is continued. $\overline{RDY}$ is used When a slow external memory is installed.

(2) Evaluation chip mode [1]

The evaluation chip mode prevails when the CNV_{SS} pin is connected to $2V_{CC}$. This mode is usually used for evaluation equipment.

A_7/MA_7 to A_0/MA_0, ALE, $\overline{BHE}$, $\overline{BLE}$, $R/\overline{W}$, $\overline{RDY}$, $\overline{HOLD}$, and ϕ_1 perform the same operations as in the microprocessor mode.

Ports A_{15}/D_{15} through A_8/D_8 function irrespective of the BYTE pin level. They serve

as the address output terminals while $\overline{E}$ is at the "H" level or

as the odd-numbered address data I/O terminals while $\overline{E}$ is at the "L" level.

However, if the internal memory area is read, external data is ignored from outside while $\overline{E}$ is at the "L" level.

When the BYTE pin level is "L", ports A_{23}/D_7 through A_{16}/D_0 serve

as the address output terminals while $\overline{E}$ is at the "H" level or

as the even-numbered address data I/O terminals while $\overline{E}$ is at the "L" level.

However, if the internal memory area is read, external data is ignored from outside while $\overline{E}$ is at the "L" level.

When the BYTE pin level is "H", on the other hand, ports A_{23}/D_7 through A_{16}/D_0 serve

as the address output terminals while $\overline{E}$ is at the "H" level or

as the even- and odd-numbered address data I/O terminals while $\overline{E}$ is at the "L" level.

However, if the internal memory area is read, external data is ignored from outside while $\overline{E}$ is at the "L" level.

Port P4, port P5, port P4 and port P5 direction register must be accessed in 8-bit unit.

The port P4 and its direction register addresses (addresses $0A_{16}$ and $0C_{16}$) are treated special in the evaluation chip mode. When these addresses are accessed, the data bus width is considered to be 16 bits without regard to the BYTE pin level. Further, the access cycle is treated as the internal memory without regard to the wait bit.

Ports $P4_3$ through $P4_6$ serve as the MX, QCL, VDA, and VPA output terminals, respectively. Port $P4_7$ serves as the $\overline{DBC}$ input terminal.

The MX signal usually carries the contents of flag m. When the CPU uses flag x, however, the MX signal carries the contents of flag x.

QCL is the queue buffer clear signal. It goes High when the command queue buffer is to be cleared as a command code previously entered in the command queue buffer cannot be used because of jump or other command execution.

VDA is the valid data address signal. It is at the "H" level while the CPU reads data from or writes data into the data buffer. It also goes High when the command first byte (the operation code) is read from the queue buffer.

VPA is the valid program address signal. It is High while the CPU reads a command code from the command queue buffer.

$\overline{DBC}$ is the debug control signal which is used for debugging.

Table 9 summarizes the relationship between the CNV_{SS} pin input level and processor mode.

Table 9. Relationship between the CNV_{SS} pin input levels and processor modes

CNV_{SS}	Mode	Description
V_{SS} or V_{CC}	• Microprocessor • Evaluation chip	Microprocessor mode upon starting after reset. Evaluation chip mode can be selected by changing the processor mode bit by software.
$2V_{CC}$	• Evaluation chip	Evaluation chip mode only

CLOCK GENERATING CIRCUIT

Figure 90 shows a block diagram of the clock generator.

When a STP instruction is executed, the internal clock ϕ stops oscillating at "L" level. At the same time, FFF_{16} is written to watchdog timer and the watchdog timer input connection is forced to f_{32}. This connection is broken and connected to the input determined by the watchdog timer frequency selection flag when the most significant bit of the watchdog timer is cleared or reset.

Oscillation restarts when an interrupt is received, but the internal clock ϕ remains at "L" level until the most significant bit of the watchdog timer is cleared. This is to avoid the unstable interval at the start of oscillation when using a ceramic resonator. When a WIT instruction is executed, the internal clock ϕ stops at "L" level, but the oscillator does not stop. The clock is restarted when an interrupt is received. Instructions can be executed immediately because the oscillator is not stopped.

The stop or wait state is released when an interrupt is received or when reset is issued. Therefore, interrupts must be enabled before executing a STP or WIT instruction.

Figure 91 shows a circuit example using a ceramic (or quartz crystal) resonator. Use the manufacturer's recommended values for constants such as capacitance which differ for each resonator. Figure 92 shows an example of using an external clock signal.

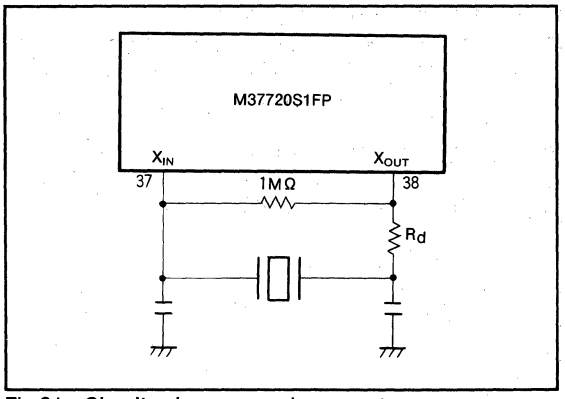


Fig.91 Circuit using a ceramic resonator

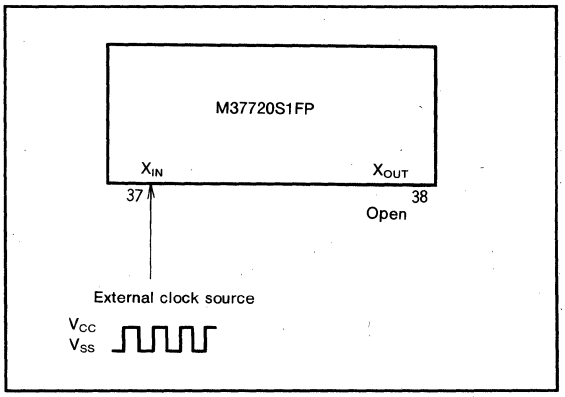


Fig.92 External clock input circuit

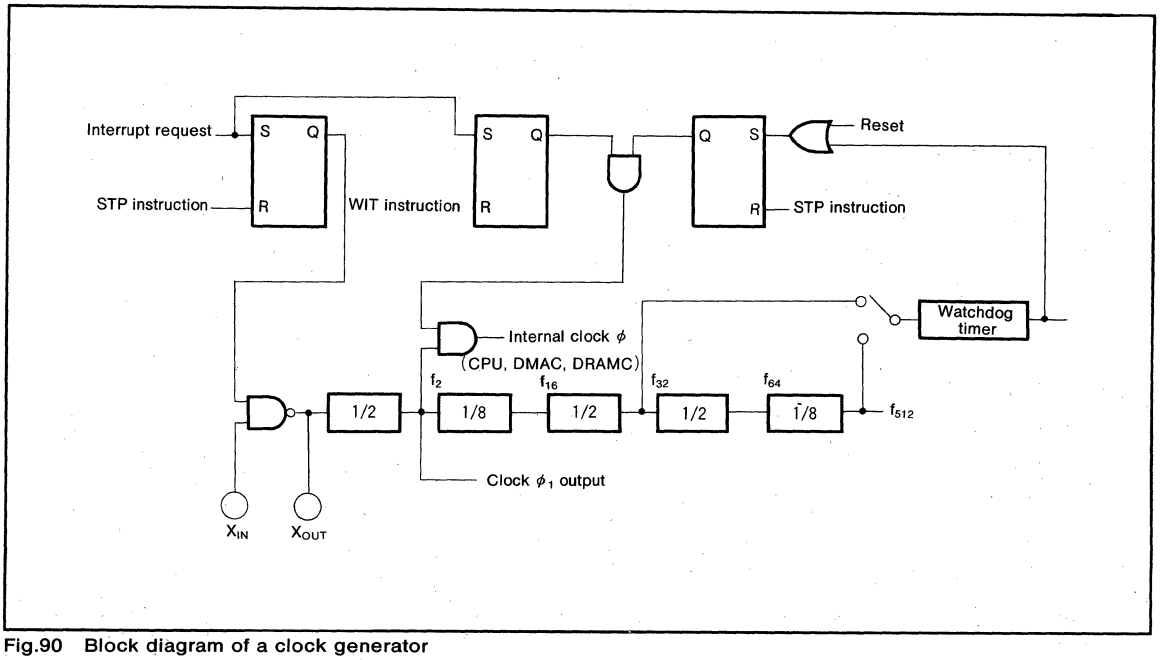


Fig.90 Block diagram of a clock generator

MITSUBISHI ELECTRIC

ADDRESSING MODES

The M37720S1FP has 28 powerful addressing modes. Refer to the MELPS 7700 addressing mode description for the details of each addressing mode.

MACHINE INSTRUCTION LIST

The M37720S1FP has 103 machine instructions. Refer to the MELPS 7700 machine instruction list for details.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Supply voltage		−0.3∼7	V
AV_{CC}	Analog supply voltage		−0.3∼7	V
V_I	Input voltage $\overline{RESET}$, CNV_{SS}, BYTE		−0.3∼12	V
V_I	Input voltage A_8/D_8∼A_{15}/D_{15}, A_{16}/D_0∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, $P9_0$∼$P9_7$, $P10_0$∼$P10_7$, $\overline{RDY}$, $\overline{HOLD}$, X_{IN}, V_{REF}		−0.3∼V_{CC}+0.3	V
V_O	Output voltage A_0/MA_0∼A_7/MA_7, A_8/D_8∼A_{15}/D_{15}, A_{16}/D_0∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, $P9_0$∼$P9_7$, $P10_0$∼$P10_7$, ϕ_1, $\overline{RESET_{OUT}}$, X_{OUT}, $\overline{E}$, ST0, ST1, ALE, $\overline{BLE}$, $\overline{BHE}$, R/$\overline{W}$		−0.3∼V_{CC}+0.3	V
P_d	Power dissipation	T_a=25℃	300	mW
T_{opr}	Operating temperature		−20∼85	℃
T_{stg}	Storage temperature		−40∼150	℃

RECOMMENDED OPERATING CONDITIONS (V_{CC}=5V±10%, T_a=−20∼85℃, unless otherwise noted)

Symbol	Parameter		Limits Min.	Limits Typ.	Limits Max.	Unit
V_{CC}	Supply voltage		4.5	5.0	5.5	V
AV_{CC}	Analog supply voltage			V_{CC}		V
V_{SS}	Supply voltage			0		V
AV_{SS}	Analog supply voltage			0		V
V_{IH}	High-level input voltage $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, $P9_0$∼$P9_7$, $P10_0$∼$P10_7$, $\overline{RDY}$, $\overline{HOLD}$, BYTE, CNV_{SS}, X_{IN}, V_{REF}		0.8V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage A_8/D_8∼A_{15}/D_{15}, A_{16}/D_0∼A_{23}/D_7		0.5V_{CC}		V_{CC}	V
V_{IL}	Low-level input voltage $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, $P9_0$∼$P9_7$, $P10_0$∼$P10_7$, $\overline{RDY}$, $\overline{HOLD}$, BYTE, CNV_{SS}, X_{IN}, V_{REF}		0		0.2V_{CC}	V
V_{IL}	Low-level input voltage A_8/D_8∼A_{15}/D_{15}, A_{16}/D_0∼A_{23}/D_7		0		0.16V_{CC}	V
$I_{OH(peak)}$	High-level peak output current A_0/MA_0∼A_7/MA_7, A_8/D_8∼A_{15}/D_{15}, A_{16}/D_0∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, $P9_0$∼$P9_7$, $P10_0$∼$P10_7$, ϕ_1, $\overline{RESET_{OUT}}$, ST0, ST1, ALE, $\overline{BLE}$, $\overline{BHE}$, R/$\overline{W}$				−10	mA
$I_{OH(avg)}$	High-level average output current A_0/MA_0∼A_7/MA_7, A_8/D_8∼A_{15}/D_{15}, A_{16}/D_0∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, $P9_0$∼$P9_7$, $P10_0$∼$P10_7$, ϕ_1, $\overline{RESET_{OUT}}$, ST0, ST1, ALE, $\overline{BLE}$, $\overline{BHE}$, R/$\overline{W}$				−5	mA
$I_{OL(peak)}$	Low-level peak output current A_0/MA_0∼A_7/MA_7, A_8/D_8∼A_{15}/D_{15}, A_{16}/D_0∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, $P9_0$∼$P9_7$, $P10_0$∼$P10_7$, ϕ_1, $\overline{RESET_{OUT}}$, ST0, ST1, ALE, $\overline{BLE}$, $\overline{BHE}$, R/$\overline{W}$				10	mA
$I_{OL(avg)}$	Low-level average output current A_0/MA_0∼A_7/MA_7, A_8/D_8∼A_{15}/D_{15}, A_{16}/D_0∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, $P9_0$∼$P9_7$, $P10_0$∼$P10_7$, ϕ_1, $\overline{RESET_{OUT}}$, ST0, ST1, ALE, $\overline{BLE}$, $\overline{BHE}$, R/$\overline{W}$				5	mA
$f(X_{IN})$	External clock frequency input	M37720S1FP			8	MHz
		M37720S1AFP			16	

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

2. The sum of $I_{OL(peak)}$ for ports P8, P9, A_0/MA_0∼A_7/MA_7, A_8/D_8∼A_{15}/D_{15}, A_{16}/D_0∼A_{23}/D_7, ST0, ST1, ALE, $\overline{BLE}$, $\overline{BHE}$ and R/$\overline{W}$ must be 80mA or less, the sum of $I_{OH(peak)}$ for ports P8, P9, A_0/MA_0∼A_7/MA_7, A_8/D_8∼A_{15}/D_{15}, A_{16}/D_0∼A_{23}/D_7, ST0, ST1, ALE, $\overline{BLE}$, $\overline{BHE}$ and R/$\overline{W}$ must be 80mA or less, the sum of $I_{OL(peak)}$ for ports P4, P5, P6, P7, P10 and ϕ_1 must be 80mA or less, the sum of $I_{OH(peak)}$ for ports P4, P5, P6, P7, P10 and ϕ_1 must be 80mA or less.

M37720S1FP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=8MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	High-level output voltage A_0/MA_0~A_7/MA_7, A_8/D_8~A_{15}/D_{15}, A_{16}/D_0~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, $P9_0$~$P9_7$, $P10_0$~$P10_7$, ϕ_1, $\overline{RESET_{OUT}}$, ST0, ST1, ALE, $\overline{BLE}$, $\overline{BHE}$, R/$\overline{W}$	I_{OH}=−10mA	3			V
V_{OH}	High-level output voltage A_0/MA_0~A_7/MA_7, A_8/D_8~A_{15}/D_{15}, A_{16}/D_0~A_{23}/D_7, MA_8, MA_9, $\overline{RAS}$, $\overline{CAS}$, ϕ_1, ST0, ST1, $\overline{BLE}$, $\overline{BHE}$, R/$\overline{W}$	I_{OH}=−400μA	4.7			V
V_{OH}	High-level output voltage ALE	I_{OH}=−10mA	3.1			V
		I_{OH}=−400μA	4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA	3.4			V
		I_{OH}=−400μA	4.8			
V_{OL}	Low-level output voltage A_0/MA_0~A_7/MA_7, A_8/D_8~A_{15}/D_{15}, A_{16}/D_0~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, $P9_0$~$P9_7$, $P10_0$~$P10_7$, ϕ_1, $\overline{RESET_{OUT}}$, ST0, ST1, ALE, $\overline{BLE}$, $\overline{BHE}$, R/$\overline{W}$	I_{OL}=10mA			2	V
V_{OL}	Low-level output voltage A_0/MA_0~A_7/MA_7, A_8/D_8~A_{15}/D_{15}, A_{16}/D_0~A_{23}/D_7, MA_8, MA_9, $\overline{RAS}$, $\overline{CAS}$, ϕ_1, ST0, ST1, $\overline{BLE}$, $\overline{BHE}$, R/$\overline{W}$	I_{OL}=2mA			0.45	V
V_{OL}	Low-level output voltage ALE	I_{OL}=10mA			1.9	V
		I_{OL}=2mA			0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA			1.6	V
		I_{OL}=2mA			0.4	
$V_{T+}-V_{T-}$	Hysteresis $\overline{RDY}$, $\overline{HOLD}$, $TA2_{IN}$~$TA4_{IN}$, $TB0_{IN}$, $TB1_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, $\overline{AD_{TRG}}$, $\overline{CTS_0}$, $\overline{CTS_1}$, CLK_0, CLK_1 $\overline{DMAREQ0}$~$\overline{DMAREQ3}$, $\overline{TC}$		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis $\overline{RESET}$		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current A_8/D_8~A_{15}/D_{15}, A_{16}/D_0~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, $P9_0$~$P9_7$, $P10_0$~$P10_7$, $\overline{RDY}$, $\overline{HOLD}$, BYTE, CNV_{SS}, X_{IN}, $\overline{RESET}$	V_I=5V			5	μA
I_{IL}	Low-level input current A_8/D_8~A_{15}/D_{15}, A_{16}/D_0~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, $P9_0$~$P9_7$, $P10_0$~$P10_7$, $\overline{RDY}$, $\overline{HOLD}$, BYTE, CNV_{SS}, X_{IN}, $\overline{RESET}$	V_I=0V			−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped	2			V
I_{CC}	Power supply current	f(X_{IN})=8MHz square waveform		9	18	mA
		T_a=25℃ when clock is stopped			1	μA
		T_a=85℃ when clock is stopped			20	

M37720S1AFP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=16MHz)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	High-level output voltage A_0/MA_0~A_7/MA_7, A_8/D_8~A_{15}/D_{15}, A_{16}/D_0~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, $P9_0$~$P9_7$, $P10_0$~$P10_7$, ϕ_1, $\overline{RESET_{OUT}}$, ST0, ST1, ALE, $\overline{BLE}$, $\overline{BHE}$, R/$\overline{W}$	I_{OH}=−10mA	3			V
V_{OH}	High-level output voltage A_0/MA_0~A_7/MA_7, A_8/D_8~A_{15}/D_{15}, A_{16}/D_0~A_{23}/D_7, MA_8, MA_9, $\overline{RAS}$, $\overline{CAS}$, ϕ_1, ST0, ST1, $\overline{BLE}$, $\overline{BHE}$, R/$\overline{W}$	I_{OH}=−400μA	4.7			V
V_{OH}	High-level output voltage ALE	I_{OH}=−10mA	3.1			V
		I_{OH}=−400μA	4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA	3.4			V
		I_{OH}=−400μA	4.8			
V_{OL}	Low-level output voltage A_0/MA_0~A_7/MA_7, A_8/D_8~A_{15}/D_{15}, A_{16}/D_0~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, $P9_0$~$P9_7$, $P10_0$~$P10_7$, ϕ_1, $\overline{RESET_{OUT}}$, ST0, ST1, ALE, $\overline{BLE}$, $\overline{BHE}$, R/$\overline{W}$	I_{OL}=10mA			2	V
V_{OL}	Low-level output voltage A_0/MA_0~A_7/MA_7, A_8/D_8~A_{15}/D_{15}, A_{16}/D_0~A_{23}/D_7, MA_8, MA_9, $\overline{RAS}$, $\overline{CAS}$, ϕ_1, ST0, ST1, $\overline{BLE}$, $\overline{BHE}$, R/$\overline{W}$	I_{OL}=2mA			0.45	V
V_{OL}	Low-level output voltage ALE	I_{OL}=10mA			1.9	V
		I_{OL}=2mA			0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA			1.6	V
		I_{OL}=2mA			0.4	
$V_{T+}-V_{T-}$	Hysteresis $\overline{RDY}$, $\overline{HOLD}$, $TA2_{IN}$~$TA4_{IN}$, $TB0_{IN}$, $TB1_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, AD_{TRG}, $\overline{CTS_0}$, $\overline{CTS_1}$, CLK_0, CLK_1, $\overline{DMAREQ0}$~$\overline{DMAREQ3}$, $\overline{TC}$		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis $\overline{RESET}$		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current A_8/D_8~A_{15}/D_{15}, A_{16}/D_0~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, $P9_0$~$P9_7$, $P10_0$~$P10_7$, $\overline{RDY}$, $\overline{HOLD}$, BYTE, CNV_{SS}, X_{IN}, $\overline{RESET}$	V_I=5V			5	μA
I_{IL}	Low-level input current A_8/D_8~A_{15}/D_{15}, A_{16}/D_0~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, $P9_0$~$P9_7$, $P10_0$~$P10_7$, $\overline{RDY}$, $\overline{HOLD}$, BYTE, CNV_{SS}, X_{IN}, $\overline{RESET}$	V_I=0V			−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped	2			V
I_{CC}	Power supply current	f(X_{IN})=16MHz square waveform		18	36	mA
		T_a=25℃ when clock is stopped			1	μA
		T_a=85℃ when clock is stopped			20	

A-D CONVERTER CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits				Unit
			8MHz		16MHz		
			Min.	Max.	Min.	Max.	
—	Resolution	$V_{REF}=V_{CC}$		8		8	Bits
—	Absolute accuracy	$V_{REF}=V_{CC}$		±3		±3	LSB
R_{LADDER}	Ladder resistance	$V_{REF}=V_{CC}$	2	10	2	10	kΩ
t_{CONV}	Conversion time		28.5		14.5		μs
V_{REF}	Reference voltage		2	V_{CC}	2	V_{CC}	V
V_{IA}	Analog input voltage		0	V_{REF}	0	V_{REF}	V

TIMING REQUIREMENTS (V_{CC}=5V±10%, V_{SS}=0V, T_a=25℃, unless otherwise noted)

External clock input

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
t_C	External clock input cycle time	125		62		ns
$t_{W(H)}$	External clock input high-level pulse width	50		25		ns
$t_{W(L)}$	External clock input low-level pulse width	50		25		ns
t_r	External clock input rise time		20		10	ns
t_f	External clock input fall time		20		10	ns

Microprocessor mode

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
$t_{SU(P4D-E)}$	Port P4 input setup time	200		100		ns
$t_{SU(P5D-E)}$	Port P5 input setup time	200		100		ns
$t_{SU(P6D-E)}$	Port P6 input setup time	200		100		ns
$t_{SU(P7D-E)}$	Port P7 input setup time	200		100		ns
$t_{SU(P8D-E)}$	Port P8 input setup time	200		100		ns
$t_{SU(P9D-E)}$	Port P9 input setup time	200		100		ns
$t_{SU(P10D-E)}$	Port P10 input setup time	200		100		ns
$t_{h(E-P4D)}$	Port P4 input hold time	0		0		ns
$t_{h(E-P5D)}$	Port P5 input hold time	0		0		ns
$t_{h(E-P6D)}$	Port P6 input hold time	0		0		ns
$t_{h(E-P7D)}$	Port P7 input hold time	0		0		ns
$t_{h(E-P8D)}$	Port P8 input hold time	0		0		ns
$t_{h(E-P9D)}$	Port P9 input hold time	0		0		ns
$t_{h(E-P10D)}$	Port P10 input hold time	0		0		ns
$t_{SU(DH-E)}$	Data high-order input setup time	60		45		ns
$t_{SU(DL-E)}$	Data low-order input setup time	60		45		ns
$t_{SU(RDY-\phi_1)}$	$\overline{RDY}$ input setup time	70		60		ns
$t_{SU(HOLD-\phi_1)}$	$\overline{HOLD}$ input setup time	70		60		ns
$t_{h(E-DH)}$	Data high-order input hold time	0		0		ns
$t_{h(E-DL)}$	Data low-order input hold time	0		0		ns
$t_{h(\phi_1-RDY)}$	$\overline{RDY}$ input hold time	0		0		ns
$t_{h(\phi_1-HOLD)}$	$\overline{HOLD}$ input hold time	0		0		ns

Timer A input (count input in event counter mode)

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAj_{IN} input cycle time	250		125		ns
$t_{W(TAH)}$	TAj_{IN} input high-level pulse width	125		62		ns
$t_{W(TAL)}$	TAj_{IN} input low-level pulse width	125		62		ns

Timer A input (gating input in timer mode)

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAj_{IN} input cycle time	1000		500		ns
$t_{W(TAH)}$	TAj_{IN} input high-level pulse width	500		250		ns
$t_{W(TAL)}$	TAj_{IN} input low-level pulse width	500		250		ns

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

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAj_{IN} input cycle time	500		250		ns
$t_{W(TAH)}$	TAj_{IN} input high-level pulse width	250		125		ns
$t_{W(TAL)}$	TAj_{IN} input low-level pulse width	250		125		ns

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

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
$t_{W(TAH)}$	TAj_{IN} input high-level pulse width	250		125		ns
$t_{W(TAL)}$	TAj_{IN} input low-level pulse width	250		125		ns

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

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
$t_{C(UP)}$	TAj_{OUT} input cycle time	5000		2500		ns
$t_{W(UPH)}$	TAj_{OUT} input high-level pulse width	2500		1250		ns
$t_{W(UPL)}$	TAj_{OUT} input low-level pulse width	2500		1250		ns
$t_{SU(UP-T_{IN})}$	TAj_{OUT} input setup time	1000		500		ns
$t_{H(T_{IN}-UP)}$	TAj_{OUT} input hold time	1000		500		ns

Timer B input (count input in event counter mode)

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
$t_{C(TB)}$	TBj_{IN} input cycle time (one-edge count)	250		125		ns
$t_{W(TBH)}$	TBj_{IN} input high-level pulse width (one-edge count)	125		62		ns
$t_{W(TBL)}$	TBj_{IN} input low-level pulse width (one-edge count)	125		62		ns
$t_{C(TB)}$	TBj_{IN} input cycle time (both-edges count)	500		250		ns
$t_{W(TBH)}$	TBj_{IN} input high-level pulse width (both-edges count)	250		125		ns
$t_{W(TBL)}$	TBj_{IN} input low-level pulse width (both-edges count)	250		125		ns

Timer B input (pulse period measurement mode)

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
$t_{C(TB)}$	TBj_{IN} input cycle time	1000		500		ns
$t_{W(TBH)}$	TBj_{IN} input high-level pulse width	500		250		ns
$t_{W(TBL)}$	TBj_{IN} input low-level pulse width	500		250		ns

Timer B input (pulse width measurement mode)

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
$t_{C(TB)}$	TBj_{IN} input cycle time	1000		500		ns
$t_{W(TBH)}$	TBj_{IN} input high-level pulse width	500		250		ns
$t_{W(TBL)}$	TBj_{IN} input low-level pulse width	500		250		ns

A-D trigger input

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
$t_{C(AD)}$	$\overline{AD_{TRG}}$ input cycle time (minimum allowable trigger)	2000		1000		ns
$t_{W(ADL)}$	$\overline{AD_{TRG}}$ input low-level pulse width	250		125		ns

Serial I/O

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
$t_{C(CK)}$	CLK_i input cycle time	500		250		ns
$t_{W(CKH)}$	CLK_i input high-level pulse width	250		125		ns
$t_{W(CKL)}$	CLK_i input low-level pulse width	250		125		ns
$t_{d(C-Q)}$	TxD_i output delay time		150		90	ns
$t_{h(C-Q)}$	TxD_i hold time	30		30		ns
$t_{su(D-C)}$	RxD_i input setup time	60		30		ns
$t_{h(C-D)}$	RxD_i input hold time	90		90		ns

External interrupt input

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
$t_{W(INH)}$	$\overline{INT_i}$ input high-level pulse width	250		250		ns
$t_{W(INL)}$	$\overline{INT_i}$ input low-level pulse width	250		250		ns

SWITCHING CHARACTERISTICS (V_{CC}=5V±10%, V_{SS}=0V, T_a=25℃, unless otherwise noted)

Microprocessor mode

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
$t_{d(E-P4Q)}$	Port P4 data output delay time		200		100	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time		200		100	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time		200		100	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time		200		100	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time		200		100	ns
$t_{d(E-P9Q)}$	Port P9 data output delay time		200		100	ns
$t_{d(E-P10Q)}$	Port P10 data output delay time		200		100	ns

Test conditions are shown in Fig. 93.

When wait bit is "1"

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
$t_{d(AL-E)}$	Address low-order output delay time	100		30		ns
$t_{d(E-DHQ)}$	Data high-order output delay time (BYTE="L")		110		70	ns
$t_{pxz(E-DHZ)}$	Floating start delay time (BYTE="L")		5		5	ns
$t_{d(AM-E)}$	Address medium-order output delay time	100		30		ns
$t_{d(AM-ALE)}$	Address medium-order output delay time	80		24		ns
$t_{d(E-DLQ)}$	Data low-order output delay time		110		70	ns
$t_{pxz(E-DLZ)}$	Floating start delay time		5		5	ns
$t_{d(AH-E)}$	Address high-order output delay time	100		30		ns
$t_{d(AH-ALE)}$	Address high-order output delay time	80		24		ns
$t_{d(ALE-E)}$	ALE output delay time	4		4		ns
$t_{w(ALE)}$	ALE pulse width	90		35		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time	100		30		ns
$t_{d(BLE-E)}$	$\overline{BLE}$ output delay time	100		30		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time	100		30		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time	0	30	0	20	ns
$t_{h(E-AL)}$	Address low-order hold time	50		25		ns
$t_{h(ALE-AM)}$	Address medium-order hold time (BYTE="L")	9		9		ns
$t_{h(E-DHQ)}$	Data high-order hold time (BYTE="L")	50		25		ns
$t_{pzx(E-DHZ)}$	Floating clear delay time (BYTE="L")	50		25		ns
$t_{h(E-AM)}$	Address medium-order hold time (BYTE="H")	50		25		ns
$t_{h(ALE-AH)}$	Address high-order hold time	9		9		ns
$t_{h(E-DLQ)}$	Data low-order hold time	50		25		ns
$t_{pzx(E-DLZ)}$	Floating clear delay time	50		25		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time	18		18		ns
$t_{h(E-BLE)}$	$\overline{BLE}$ hold time	50		25		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time	18		18		ns
$t_{d(\phi_1-STi)}$	ST0, ST1 output delay time		100		50	ns
$t_{w(EL)}$	$\overline{E}$ pulse width	220		95		ns

Test conditions are shown in Fig. 93.

When wait bit is "0", and external memory area is accessed

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
$t_{d(AL-E)}$	Address low-order output delay time	100		30		ns
$t_{d(E-DHQ)}$	Data high-order output delay time (BYTE="L")		110		70	ns
$t_{pxz(E-DHZ)}$	Floating start delay time (BYTE="L")		5		5	ns
$t_{d(AM-E)}$	Address medium-order output delay time	100		30		ns
$t_{d(AM-ALE)}$	Address medium-order output delay time	80		24		ns
$t_{d(E-DLQ)}$	Data low-order output delay time		110		70	ns
$t_{pxz(E-DLZ)}$	Floating start delay time		5		5	ns
$t_{d(AH-E)}$	Address high-order output delay time	100		30		ns
$t_{d(AH-ALE)}$	Address high-order output delay time	80		24		ns
$t_{d(ALE-E)}$	ALE output delay time	4		4		ns
$t_{w(ALE)}$	ALE pulse width	90		35		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time	100		30		ns
$t_{d(BLE-E)}$	$\overline{BLE}$ output delay time	100		30		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time	100		30		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time	0	30	0	20	ns
$t_{h(E-AL)}$	Address low-order hold time	50		25		ns
$t_{h(ALE-AM)}$	Address medium-order hold time (BYTE="L")	9		9		ns
$t_{h(E-DHQ)}$	Data high-order hold time (BYTE="L")	50		25		ns
$t_{pzx(E-DHZ)}$	Floating clear delay time (BYTE="L")	50		25		ns
$t_{h(E-AM)}$	Address medium-order hold time (BYTE="H")	50		25		ns
$t_{h(ALE-AH)}$	Address high-order hold time	9		9		ns
$t_{h(E-DLQ)}$	Data low-order hold time	50		25		ns
$t_{pzx(E-DLZ)}$	Floating clear delay time	50		25		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time	18		18		ns
$t_{h(E-BLE)}$	$\overline{BLE}$ hold time	50		25		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time	18		18		ns
$t_{d(\phi_1-STi)}$	ST0, ST1 output delay time		100		50	ns
$t_{w(EL)}$	$\overline{E}$ pulse width	470		220		ns

Test conditions are shown in Fig. 93.

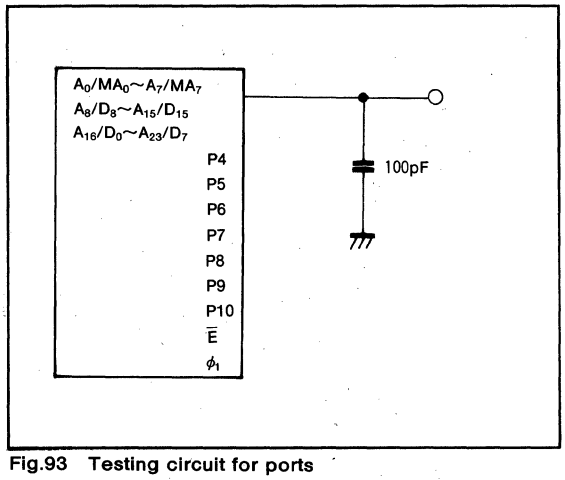


Fig.93 Testing circuit for ports

DRAM CONTROL SWITCHING CHARACTERISTICS (V_{CC}=5V±10%, V_{SS}=0V, T_a=25℃, unless otherwise noted)

Read state

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
$t_{W(RASL)}$	$\overline{RAS}$ low-level pulse width	460		210		ns
$t_{W(CASL)}$	$\overline{CAS}$ low-level pulse width	330		160		ns
$t_{W(RASH)}$	$\overline{RAS}$ high-level pulse width	210		90		ns
$t_{d(RAS-CAS)}$	$\overline{RAS}$–$\overline{CAS}$ delay time	100		40		ns
$t_{d(RA-RAS)}$	Row address delay time before $\overline{RAS}$	100		10		ns
$t_{h(RAS-RA)}$	Row address hold time after $\overline{RAS}$	15		15		ns
$t_{d(CA-CAS)}$	Column address delay time before $\overline{CAS}$	0		0		ns
$t_{h(CAS-CA)}$	Column address hold time after $\overline{CAS}$	330		160		ns
$t_{d(R/W-RAS)}$	R/$\overline{W}$ delay time before $\overline{RAS}$	100		30		ns
$t_{h(CAS-R/W)}$	R/$\overline{W}$ hold time after $\overline{CAS}$	0		0		ns
$t_{d(E-RASL)}$	$\overline{RAS}$ delay time after E low-level		30		30	ns
$t_{d(E-CASL)}$	$\overline{CAS}$ delay time after E low-level		170		100	ns
$t_{d(E-RASH)}$	$\overline{RAS}$ delay time after E high-level	0	20	0	20	ns
$t_{d(E-CASH)}$	$\overline{CAS}$ delay time after E high-level	0	20	0	20	ns

Test conditions are shown in Fig. 93.

Write state

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
$t_{W(RASL)}$	$\overline{RAS}$ low-level pulse width	460		210		ns
$t_{W(CASL)}$	$\overline{CAS}$ low-level pulse width	210		100		ns
$t_{W(RASH)}$	$\overline{RAS}$ high-level pulse width	210		90		ns
$t_{d(RAS-CAS)}$	$\overline{RAS}$–$\overline{CAS}$ delay time	210		105		ns
$t_{d(RA-RAS)}$	Row address delay time before $\overline{RAS}$	100		10		ns
$t_{h(RAS-RA)}$	Row address hold time after $\overline{RAS}$	15		15		ns
$t_{d(CA-CAS)}$	Column address delay time before $\overline{CAS}$	40		40		ns
$t_{h(CAS-CA)}$	Column address hold time after $\overline{CAS}$	225		100		ns
$t_{d(R/W-RAS)}$	R/$\overline{W}$ delay time before $\overline{RAS}$	100		30		ns
$t_{h(CAS-R/W)}$	R/$\overline{W}$ hold time after $\overline{CAS}$	0		0		ns
$t_{d(E-RASL)}$	$\overline{RAS}$ delay time after E low-level		30		30	ns
$t_{d(E-CASL)}$	$\overline{CAS}$ delay time after E low-level		310		170	ns
$t_{d(E-RASH)}$	$\overline{RAS}$ delay time after E high-level	0	20	0	20	ns
$t_{d(E-CASH)}$	$\overline{CAS}$ delay time after E high-level	0	20	0	20	ns

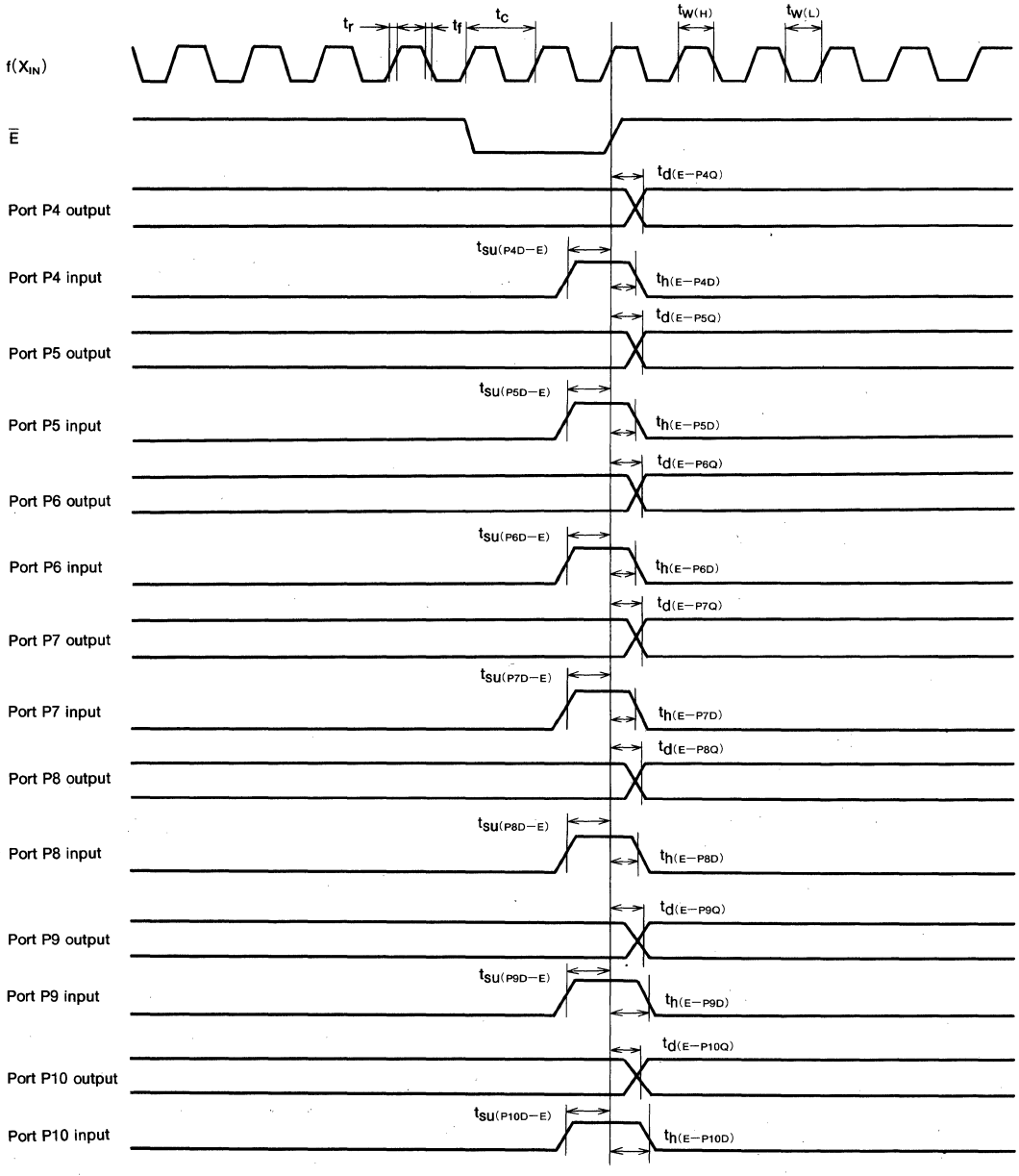
Test conditions are shown in Fig. 93.

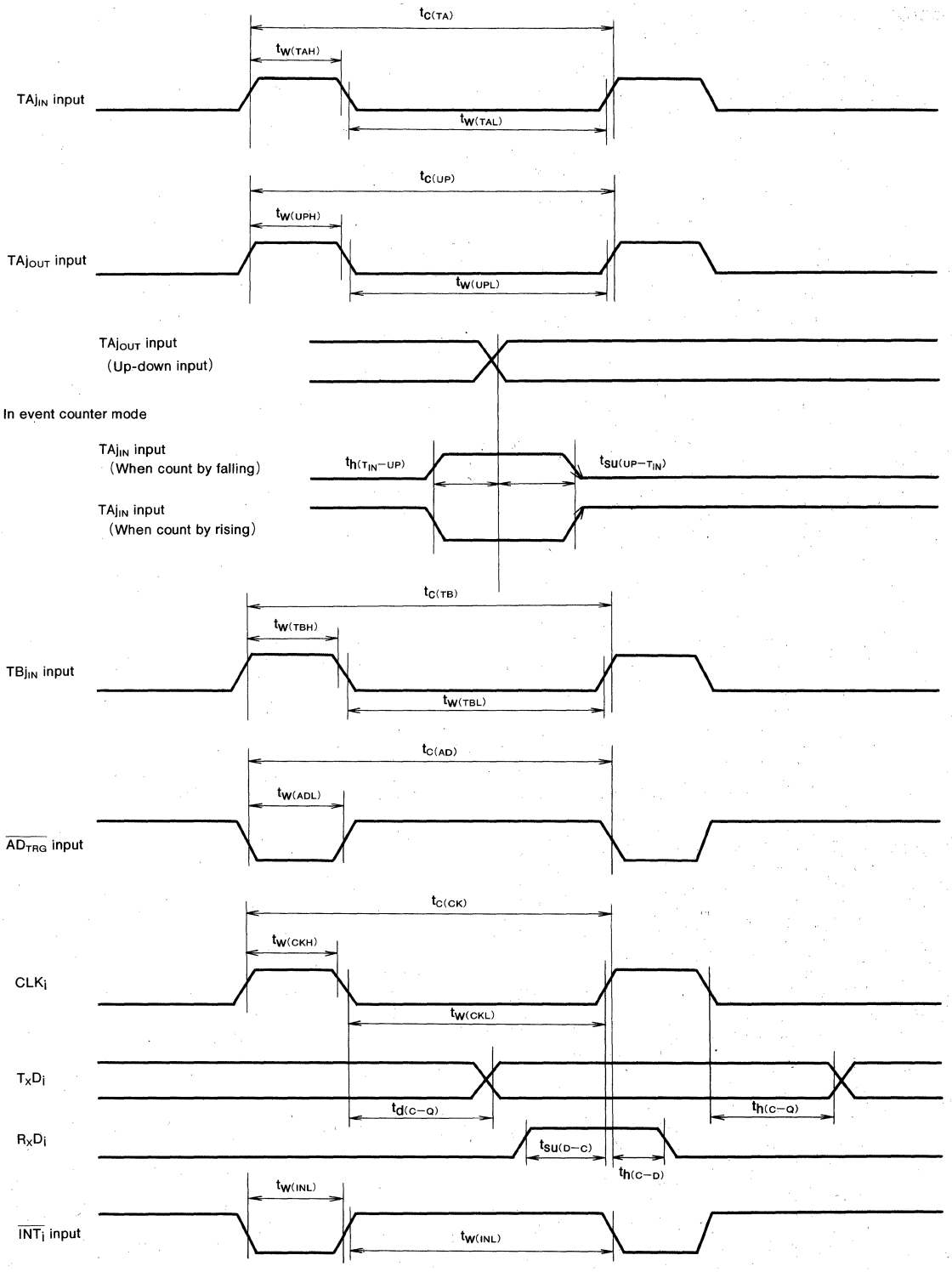
Refresh state

Symbol	Parameter	Limits				Unit
		8MHz		16MHz		
		Min.	Max.	Min.	Max.	
$t_{W(RASL)}$	$\overline{RAS}$ low-level pulse width	460		210		ns
$t_{d(CAS-RAS)}$	$\overline{CAS}$–$\overline{RAS}$ delay time	100		40		ns
$t_{h(RAS-CAS)}$	$\overline{CAS}$ hold time after $\overline{RAS}$	90		40		ns

Test conditions are shown in Fig. 93.

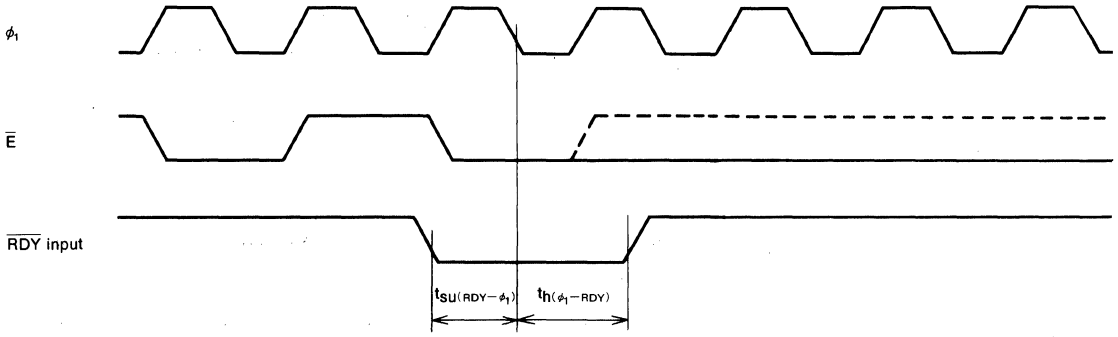
TIMING DIAGRAM



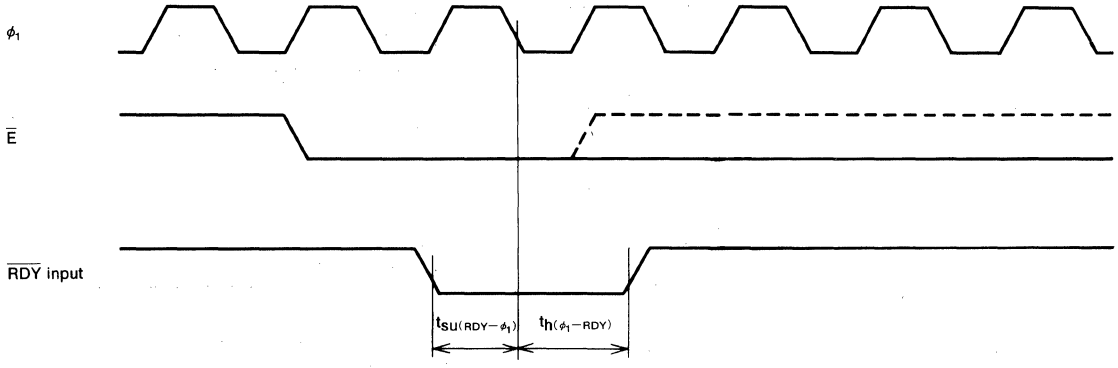


MITSUBISHI ELECTRIC

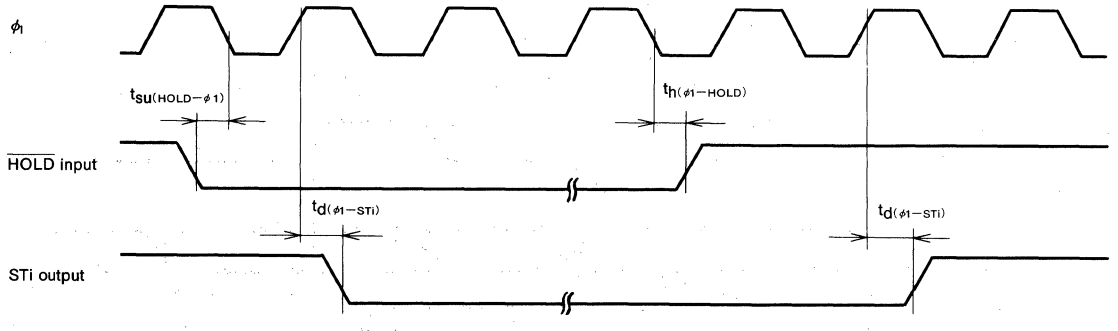
Microprocessor mode (When wait bit="1")



(When wait bit="0")



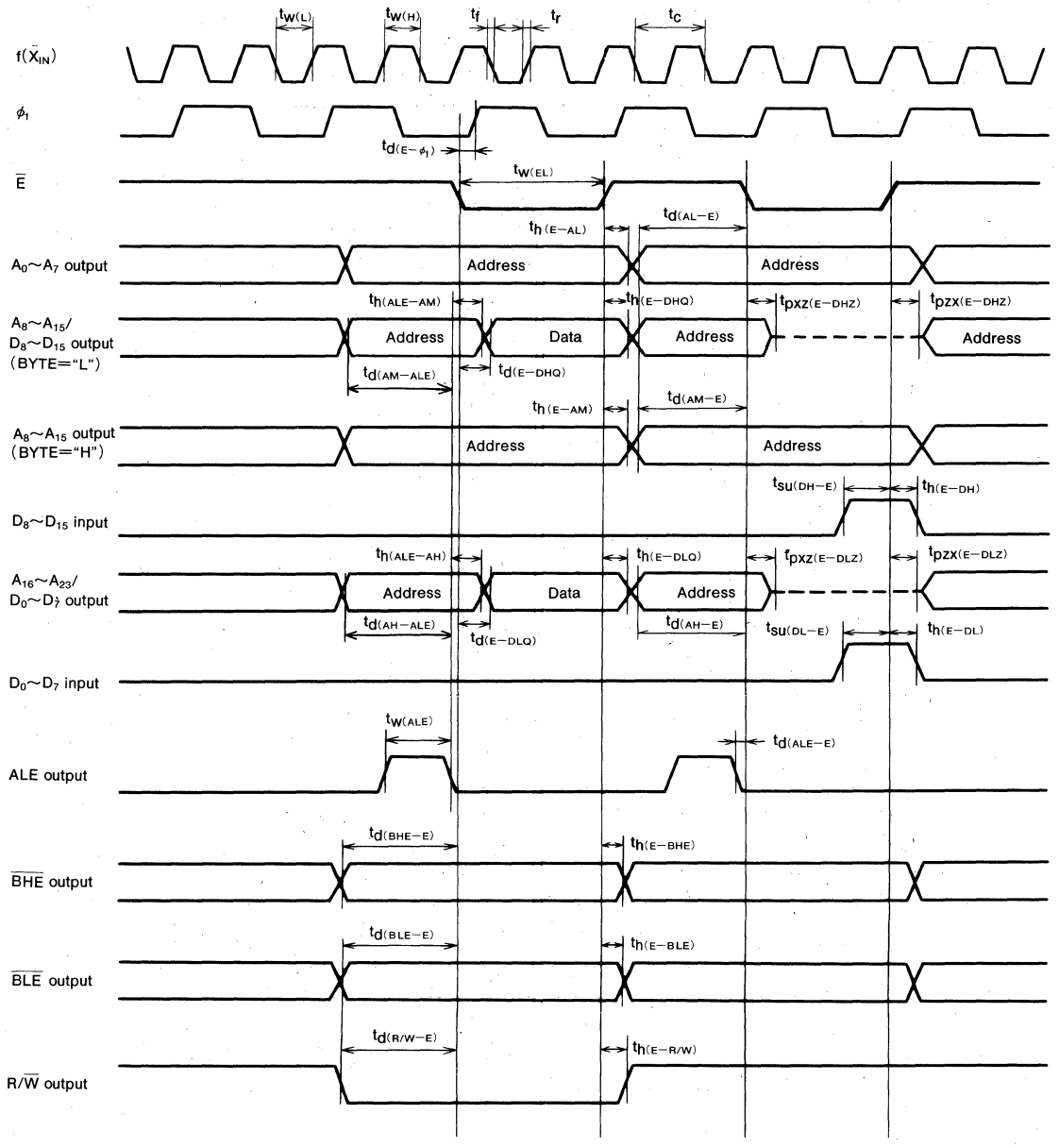
(When wait bit="1" or "0" in common)



Test conditions

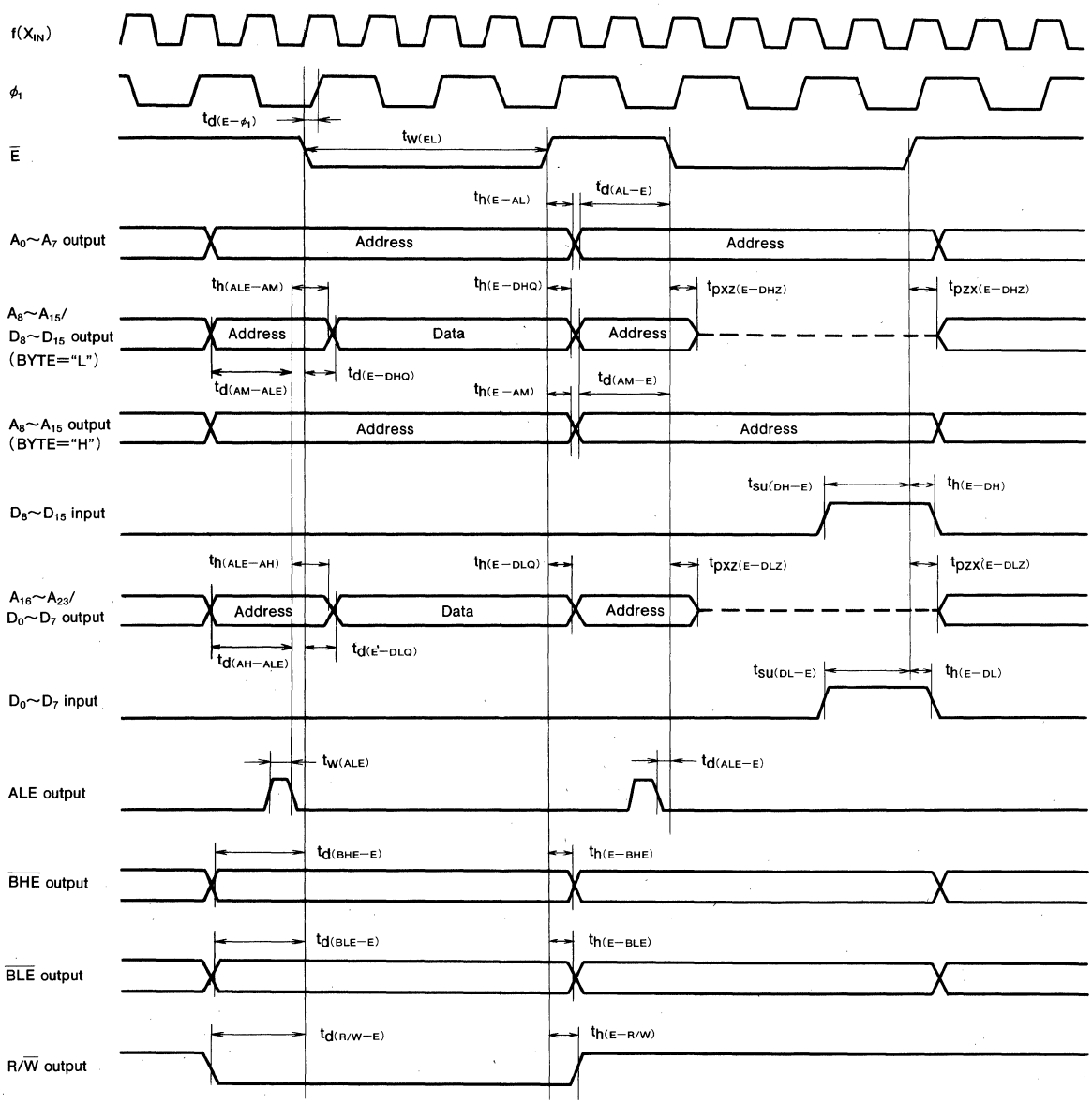
- V_{CC}=5V±10%
- Input timing voltage : V_{IL}=1.0V, V_{IH}=4.0V
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V

Microprocessor mode (When wait bit="1")



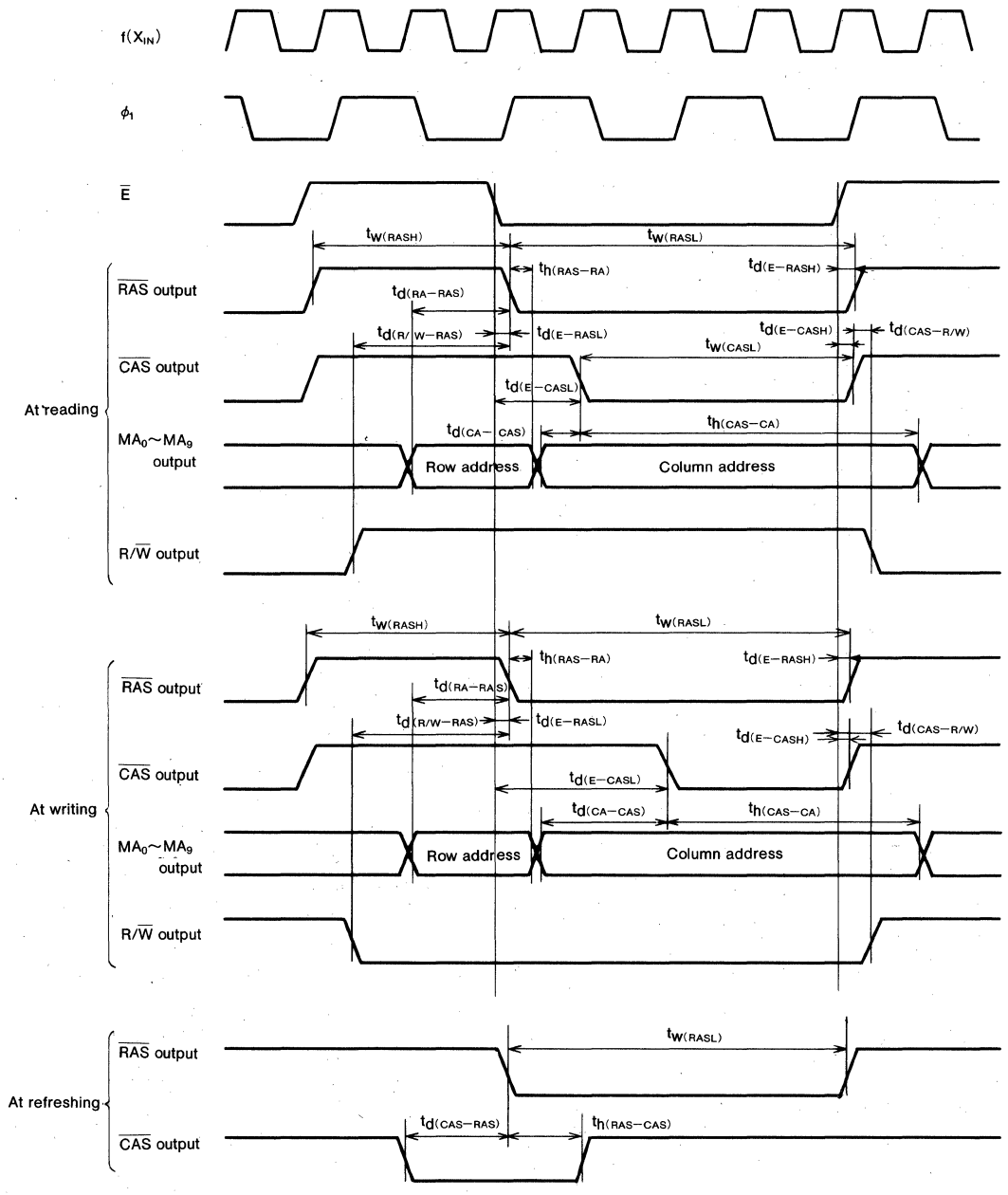
Test conditions
- V_{CC}= 5 V±10%
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V
- D_0~D_{15} input : V_{IL}=0.8V, V_{IH}=2.5V

Microprocessor mode (When wait bit="0", and external memory area is accessed)



Test conditions
- V_{CC}= 5 V±10%
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V
- D_0~D_{15} input : V_{IL}=0.8V, V_{IH}=2.5V

When DRAM control



Test conditions
- V_{CC}= 5 V±10%
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V
- D_0~D_{15} input : V_{IL}=0.8V, V_{IH}=2.5V

M37730S2FP,M37730S2AFP,M37730S2BFP
M37730S2SP,M37730S2ASP,M37730S2BSP

16-BIT CMOS MICROCOMPUTER

DESCRIPTION

The M37730S2FP, M37730S2AFP, and M37730S2BFP are 16-bit microcomputers designed with high-performance CMOS silicon gate technology. These are housed in a 64-pin plastic molded QFP. These microcomputers have a large 16M bytes address space, three instruction queue buffers, and two data buffers for high-speed instruction execution. The CPU is a 16-bit parallel processor that can also be switched to perform 8-bit parallel processing. These microcomputers are suitable for office, business, and industrial equipment controller that require high-speed processing of large data.

M37730S2SP, M37730S2ASP, and M37730S2BSP are also prepared. These are housed in a 64-pin shrink plastic molded DIP.

The differences of these types are the external clock input frequency and package. Therefore, the following descriptions will be for the M37730S2FP unless otherwise noted.

Type name	External clock input frequency	Package
M37730S2FP	8 MHz	64P6N
M37730S2AFP	16MHz	64P6N
M37730S2BFP	25MHz	64P6N
M37730S2SP	8 MHz	64P4B
M37730S2ASP	16MHz	64P4B
M37730S2BSP	25MHz	64P4B

FEATURES

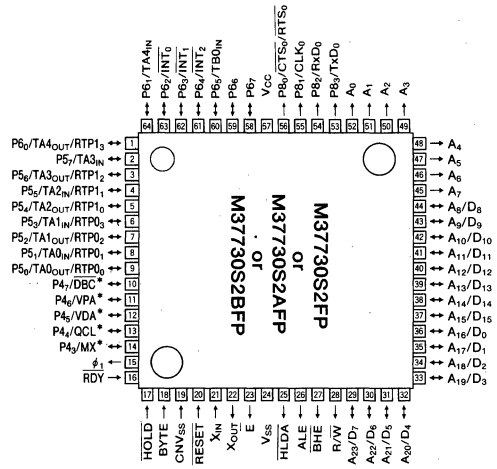
- Number of basic instructions 103
- Memory size RAM ..... 1024 bytes
- Instruction execution time
  M37730S2FP, M37730S2SP
  (The fastest instruction at 8 MHz frequency) 500ns
  M37730S2AFP, M37730S2ASP
  (The fastest instruction at 16 MHz frequency) 250ns
- M37730S2BFP, M37730S2BSP
  (The fastest instruction at 25 MHz frequency) 160ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 8 MHz frequency)
  ..... 30mW (Typ.)
- Interrupts ..... 14 types 7 levels
- Multiple function 16-bit timer ..... 5+1
- UART (may also be synchronous) 1
- 12-bit watchdog timer
- Programmable input/output
  (ports P4, P5, P6, P8) 25
- Pulse output port ..... 4-bit×2

APPLICATION

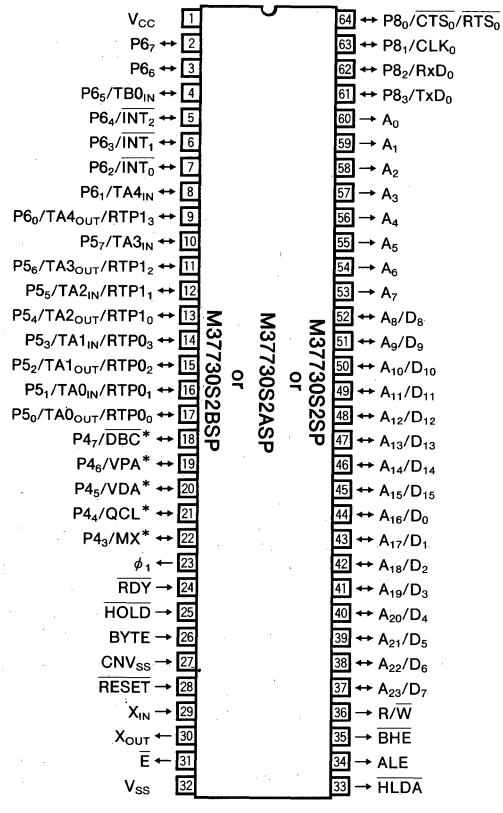
Control devices for office equipment such as copiers, printers, typewriters, facsimiles, word processors, personal computers, and HDD.

Control devices for industrial equipment such as ME, NC, communication and measuring instruments.

PIN CONFIGURATION (TOP VIEW)



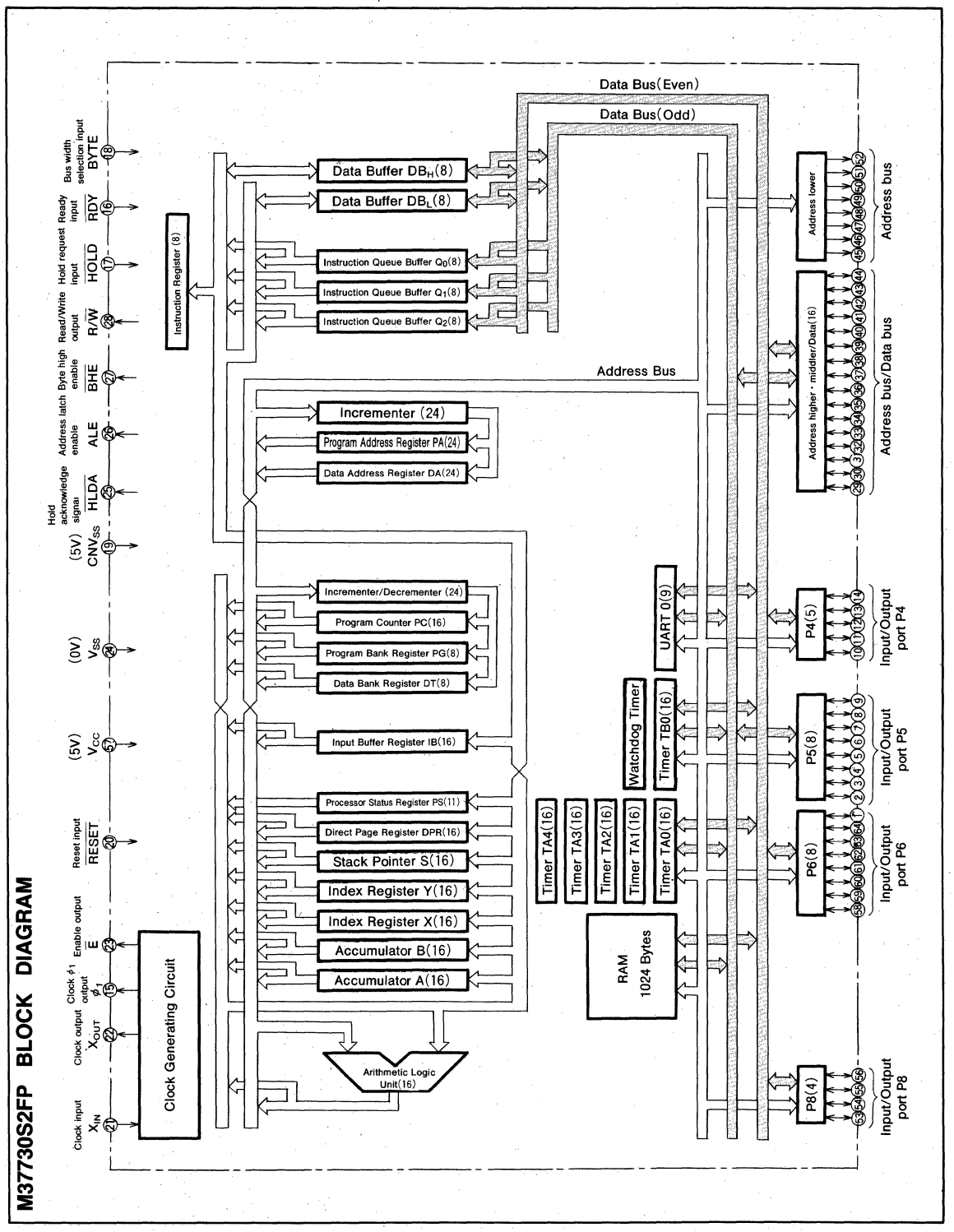
Outline 64P6N



Outline 64P4B

* : Used in the evaluation chip mode only

M37730S2FP BLOCK DIAGRAM



MITSUBISHI ELECTRIC

FUNCTIONS OF M37730S2FP

Parameter		Functions
Number of basic instructions		103
Instruction execution time	M37730S2FP, M37730S2SP	500ns (the fastest instructions, at 8MHz frequency)
	M37730S2AFP, M37730S2ASP	250ns (the fastest instructions, at 16MHz frequency)
	M37730S2BFP, M37730S2BSP	160ns (the fastest instructions, at 25MHz frequency)
Memory size	RAM	1024 bytes
Input/Output ports	P5, P6	8-bit× 2
	P4	5-bit× 1
	P8	4-bit× 1
Multi-function timers	TA0, TA1, TA2, TA3, TA4	16-bit× 5
	TB0	16-bit× 1
Serial I/O		(UART or clock synchronous serial I/O)× 1
Watchdog timer		12-bit× 1
Interrupts		3 external types, 11 internal types (Each interrupt can be set the priority levels to 0 ~ 7.)
Clock generating circuit		Built-in(externally connected to a ceramic resonator or quartz crystal resonator)
Supply voltage		5 V±10%
Power dissipation		30mW(at external 8 MHz frequency)
Input/Output characteristic	Input/Output voltage	5 V
	Output current	5 mA
Memory space		16M bytes
Operating temperature range		−20~85℃
Device structure		CMOS high-performance silicon gate process
Package	M37730S2FP, M37730S2AFP, M37730S2BFP	64-pin plastic molded QFP
	M37730S2SP, M37730S2ASP, M37730S2BSP	64-pin shrink plastic molded DIP

PIN DESCRIPTION

Pin	Name	Input/Output	Functions
V_{CC}, V_{SS}	Power supply		Supply 5 V±10% to V_{CC} and 0 V to V_{SS}.
CNV_{SS}	CNV_{SS} input	Input	Connect to V_{CC}.
$\overline{RESET}$	Reset input	Input	To enter the reset state, this pin must be kept at a "L" condition should be maintained for the required time.
X_{IN}	Clock input	Input	These are I/O pins of internal clock generating circuit. Connect a ceramic or quartz crystal resonator between X_{IN} and X_{OUT}. When an external clock is used, the clock source should be connected to the X_{IN} pin and the X_{OUT} pin should be left open.
X_{OUT}	Clock output	Output	
$\overline{E}$	Enable output	Output	Data or instruction read and data write are performed when output from this pin is "L".
BYTE	Bus width selection input	Input	This pin determines whether the external data bus is 8-bit width or 16-bit width. The width is 16 bits when "L" signal inputs and 8 bits when "H" signal inputs.
ϕ_1	Clock output	Output	This pin outputs the clock ϕ_1 which is divided the clock to X_{IN} pin by 2.
$\overline{RDY}$	Ready	Input	This is ready input pin. This is an input pin for the $\overline{RDY}$ signal. Internal clock stops while this signal is "L".
$\overline{HOLD}$	Hold request input	Input	This is an input pin for $\overline{HOLD}$ request signal. The microcomputer enters into hold state while this signal is "L".
$\overline{HLDA}$	Hold acknowledge output	Output	This is an output pin for $\overline{HLDA}$ signal, indicates the hold state.
R/$\overline{W}$	Read/Write output	Output	"H" indicates the read status and "L" indicates the write status.
$\overline{BHE}$	Byte high enable output	Output	"L" is output when an odd-numbered address is accessed.
ALE	Address latch enable output	Output	This is used to retrieve only the address data from address data and data multiplex signal.
A_0~A_7	Address (low-order) output	Output	Address (A_7~A_0) is output.
A_8/D_8~A_{15}/D_{15}	Address (middle-order) output/Data (high-order) I/O	I/O	In case the BYTE pin is "L" and an external data bus is 16-bit width, high-order data (D_{15}~D_8) is input or output when $\overline{E}$ output is "L" and an address (A_{15}~A_8) is output when $\overline{E}$ output is "H". In case the BYTE pin is "H" and an external data bus is 8-bit width, only address (A_{15}~A_8) is output.
A_{16}/D_0~A_{23}/D_7	Address (high-order) output/Data (low-order) I/O	I/O	Low-order data (D_7~D_0) is input or output when $\overline{E}$ output is "L", and an address (A_{23}~A_{16}) is output when $\overline{E}$ output is "H".
$P4_3$~$P4_7$	I/O port P4	I/O	Port P4 is a 5-bit I/O port. This port has an data direction register and each pin can be programmed for input or output. This port is in input mode when reset.
$P5_0$~$P5_7$	I/O port P5	I/O	Port P5 is a 8-bit I/O port. This port has an data direction register and each pin can be programmed for input or output. This port is in input mode when reset. These pins also function as I/O pins for timer A0, timer A1, timer A2 and timer A3.
$P6_0$~$P6_7$	I/O port P6	I/O	In addition to having the same functions as port P5, these pins also function as I/O pins for timer A4, external interrupt input $\overline{INT_0}$, $\overline{INT_1}$ and $\overline{INT_2}$ pins, and input pin for timer B0.
$P8_0$~$P8_3$	I/O port P8	I/O	Port P8 is a 4-bit I/O port. This port has an data direction register and each pin can be programmed for input or output. This port is in input mode when reset. These pins also function as RxD, TxD, CLK, $\overline{CTS}/\overline{RTS}$ pins for UART0.

M37730S2FP,M37730S2AFP,M37730S2BFP
M37730S2SP,M37730S2ASP,M37730S2BSP

BASIC FUNCTION BLOCKS

The M37730S2FP contains the following devices on a chip: RAM for storing data, CPU for processing, bus interface unit (which controls instruction prefetch and data read/write between CPU and memory), timers, UART, and other peripheral devices such as I/O ports. Each of these devices are described below.

MEMORY

The memory map is shown in Figure 1. The address space is 16M bytes from addresses 0_{16} to FFFFFF$_{16}$. The address space is divided into 64K bytes units called banks. The banks are numbered from 0_{16} to FF$_{16}$.

Built-in RAM and control registers for built-in peripheral devices are assigned to bank 0_{16}.

Addresses FFDC$_{16}$ to FFFF$_{16}$ are the RESET and interrupt vector addresses and contain the interrupt vectors. Use ROM for memory of this address. Refer to the section on interrupts for details.

The 1024 bytes area from addresses 80_{16} to 47F$_{16}$ contains the built-in RAM. In addition to storing data, the RAM is used as stack during a subroutine call, or interrupts.

Assigned to addresses 0_{16} to 7F$_{16}$ are peripheral devices such as I/O ports, UART, timer, and interrupt control registers.

A 256 bytes direct page area can be allocated anywhere in bank 0_{16} using the direct page register DPR. In direct page addressing mode, the memory in the direct page area can be accessed with two words thus reducing program steps.

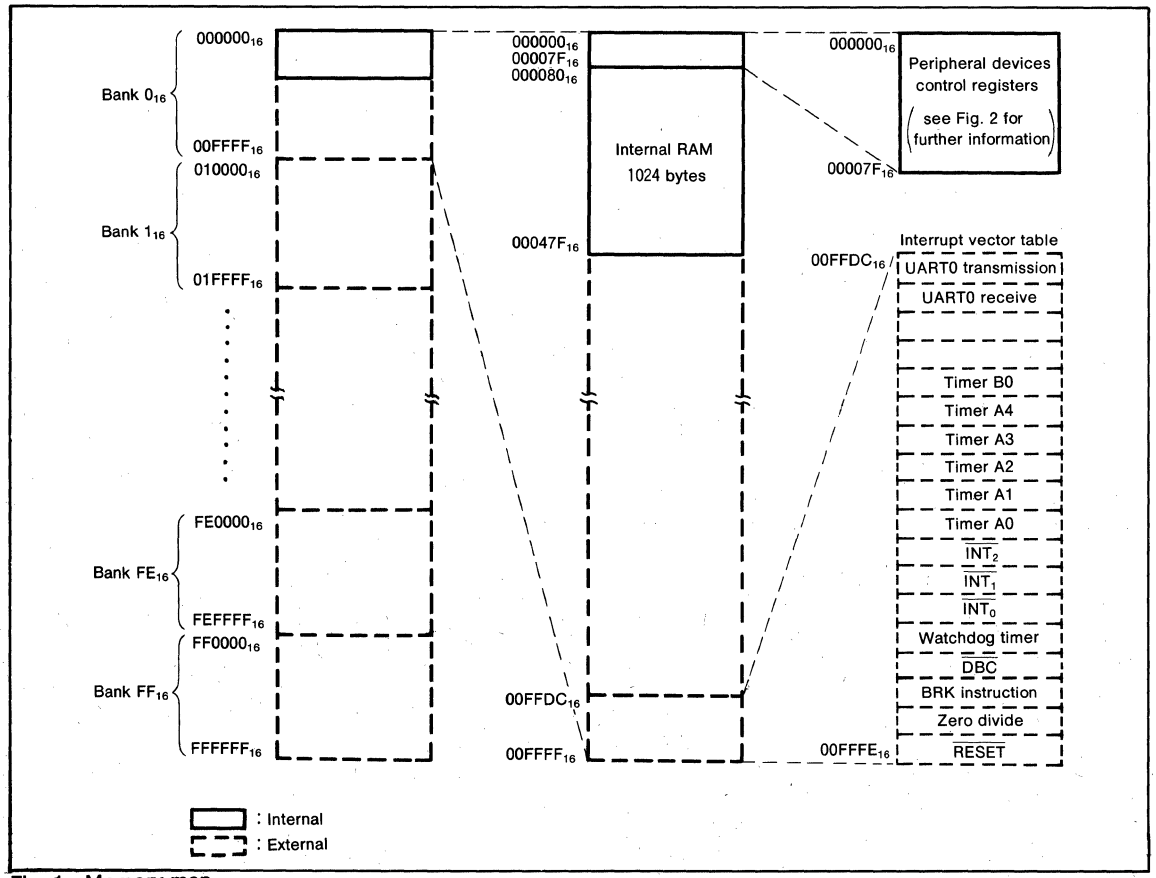


Fig. 1 Memory map

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

Address (Hexadecimal notation)	
000040	Count start flag
000041	
000042	One shot start flag
000043	
000044	Up-down flag
000045	
000046	Timer A0
000047	
000048	Timer A1
000049	
00004A	Timer A2
00004B	
00004C	Timer A3
00004D	
00004E	Timer A4
00004F	
000050	Timer B0
000051	
000052	
000053	
000054	
000055	
000056	Timer A0 mode register
000057	Timer A1 mode register
000058	Timer A2 mode register
000059	Timer A3 mode register
00005A	Timer A4 mode register
00005B	Timer B0 mode register
00005C	
00005D	
00005E	Processor mode register
00005F	
000060	Watchdog timer
000061	Watchdog timer frequency selection flag
000062	Waveform output mode register
000063	
000064	Pulse output data register 1
000065	Pulse output data register 0
000066	
000067	
000068	
000069	
00006A	
00006B	
00006C	
00006D	
00006E	
00006F	
000070	
000071	UART0 transmission interrupt control register
000072	UART0 receive interrupt control register
000073	
000074	
000075	Timer A0 interrupt control register
000076	Timer A1 interrupt control register
000077	Timer A2 interrupt control register
000078	Timer A3 interrupt control register
000079	Timer A4 interrupt control register
00007A	Timer B0 interrupt control register
00007B	
00007C	
00007D	$\overline{INT_0}$ interrupt control register
00007E	$\overline{INT_1}$ interrupt control register
00007F	$\overline{INT_2}$ interrupt control regiter

Fig. 2. Location of peripheral devices and interrupt control registers

CENTRAL PROCESSING UNIT (CPU)

The CPU has ten registers and is shown in Figure 3. Each of these registers is described below.

ACCUMULATOR A (A)

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

Data operations such as calculations, data transfer, input/output, etc., is executed mainly through the accumulator.

ACCUMULATOR B (B)

Accumulator B has the same functions as accumulator A, but the use of accumulator B requires more instruction bytes and execution cycles than accumulator A.

INDEX REGISTER X (X)

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

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

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

INDEX REGISTER Y (Y)

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

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

Also, when executing a block transfer instruction MVP or MVN, the content of index register Y indicates the low-order 16 bits of the destination address. The second byte of the MVP and MVN is the high-order 8 bits of the destination data address.

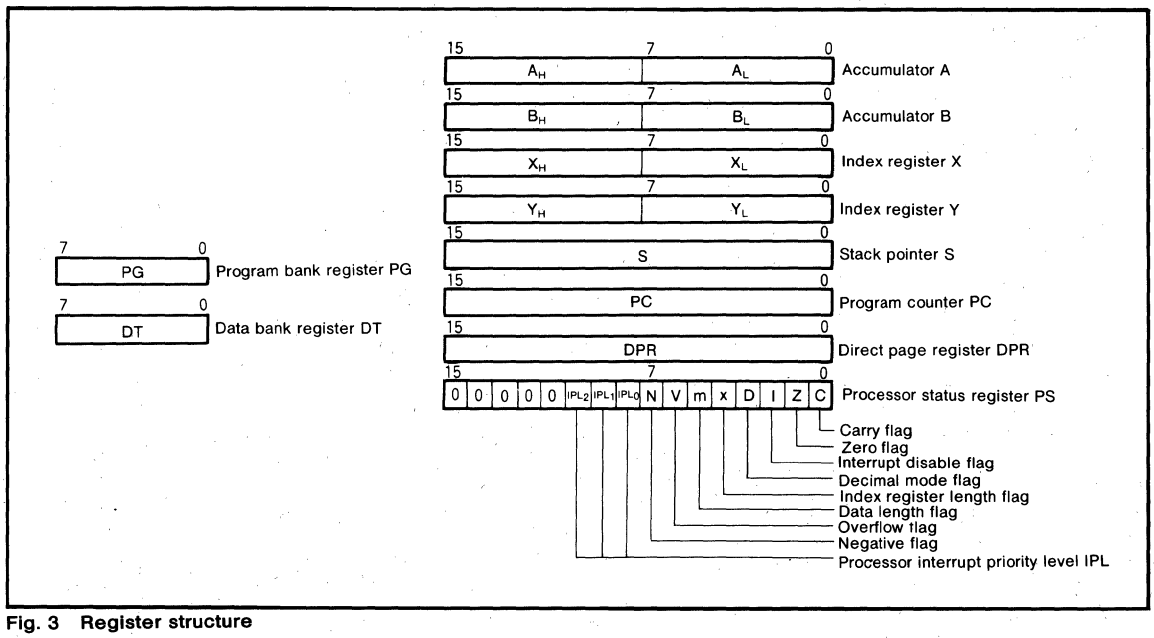


Fig. 3 Register structure

STACK POINTER (S)

Stack pointer (S) is an 16-bit register. It is used during a subroutine call or interrupts. It is also used during stack, stack pointer relative, or stack pointer relative indirect indexed Y addressing mode.

PROGRAM COUNTER (PC)

Program counter (PC) is a 16-bit counter that indicates the low-order 16 bits of the next program memory address to be executed. These is a bus interface unit between the program memory and the CPU, so that the program memory is accessed through bus interface unit. This is described later.

PROGRAM BANK REGISTER (PG)

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

DATA BANK REGISTER (DT)

Data bank register (DT) is an 8-bit register. With some addressing modes, a part of the data bank register (DT) is used to specify a memory address. The contents of data bank register (DT) is used as the high-order 8 bits of a 24-bit address. Addressing modes that use the data bank register (DT) are direct indirect, direct indexed X indirect, direct indirect indexed Y, absolute, absolute bit, absolute indexed X, absolute indexed Y, absolute bit relative, and stack pointer relative indirect indexed Y.

DIRECT PAGE REGISTER (DPR)

Direct page register (DPR) is a 16-bit register. Its contents is used as the base address of a 256-byte direct page area. The direct page area is allocated in bank 0, but when the contents of DPR is $FF01_{16}$ or greater, the direct page area spans across bank 0 and bank 1. All direct addressing modes use the contents of the direct page register (DPR) to generate the data address. If the low-order 8 bits of the direct page register (DPR) is "00_{16}", the number of cycles required to generate an address is minimized. Normally the low-order 8 bits of the direct page register (DPR) is set to "00_{16}".

PROCESSOR STATUS REGISTER (PS)

Processor status register (PS) is an 11-bit register. It consists of a flag to indicate the result of operation and CPU interrupt levels.

Branch operations can be performed by testing the flags C, Z, V, and N.

The details of each processor status register bit are described below.

1. Carry flag (C)

The carry flag contains the carry or borrow generated by the ALU after an arithmetic operation. This flag is also affected by shift and rotate instructions. This flag can be set and reset directly with the SEC and CLC instructions or with the SEP and CLP instructions.

2. Zero flag (Z)

This zero flag is set if the result of an arithmetic operation or data transfer is zero and reset if it is not. This flag can be set and reset directly with the SEP and CLP instructions.

3. Interrupt disable flag (I)

When the interrupt disable flag is set to "1", all interrupts except watchdog timer, $\overline{\text{DBC}}$, and software interrupt are disabled. This flag is set to "1" automatically when these is an interrupt. It can be set and reset directly with the SEI and CLI instructions or SEP and CLP instructions.

4. Decimal mode flag (D)

The decimal mode flag determines whether addition and subtraction are performed as binary or decimal. Binary arithmetic is performed when this flag is "0". If it is "1", decimal arithmetic is performed with each word treated as two or four digit decimal. Arithmetic operation is performed using four digits when the data length flag m is "0" and with two digits when it is "1". (Decimal operation is possible only with the ADC and SBC instructions.) This flag can be set and reset with the SEP and CLP instructions.

5. Index register length flag (x)

The index register length flag determines whether index register X and index register Y are used as 16-bit registers or as 8-bit registers. The registers are used as 16-bit registers when flag x is "0" and as 8-bit registers when it is "1". This flag can be set and reset with the SEP and CLP instructions.

6. Data length flag (m)

The data length flag determines whether the data length is 16-bit or 8-bit. The data length is 16-bit when flag m is "0" and 8-bit when it is "1". This flag can be set and reset with the SEM and CLM instructions or with the SEP and CLP instructions.

7. Overflow flag (V)

The overflow flag has meaning when addition or subtraction is performed a word as signed binary number. When the data length flag m is "0", the overflow flag is set when the result of addition or subtraction is outside the range between −32768 and +32767. When the data length flag m is "1", the overflow flag is set when the result of addition or subtraction is outside the range between −128 and +127. It is reset in all other cases. The overflow flag can also be set and reset directly with the SEP, and CLV or CLP instructions.

8. Negative flag (N)

The negative flag is set when the result of arithmetic operation or data transfer is negative (If data length flag m is "0", when data bit 15 is "1". If data length flag m is "1", when data bit 7 is "1".) It is reset in all other cases. It can also be set and reset with the SEP and CLP instructions.

9. Processor interrupt priority level (IPL)

The processor interrupt priority level (IPL) consists of 3 bits and determines the priority of processor interrupts from level 0 to level 7. Interrupt is enabled when the interrupt priority of the device requesting interrupt (set using the interrupt control register) is higher than the processor interrupt priority. When interrupt is enabled, the current processor interrupt priority level is saved in a stack and the processor interrupt priority level is replaced by the interrupt priority level of the device requesting the interrupt. Refer to the section on interrupts for more details.

BUS INTERFACE UNIT

The CPU operates on an internal clock frequency which is obtained by dividing the external clock frequency $f_{(X_{IN})}$ by two. This frequency is twice the bus cycle frequency. In order to speed-up processing, a bus interface unit is used to pre-fetch instructions when the data bus is idle. The bus interface unit synchronizes the CPU and the bus and prefetches instructions. Figure 4 shows the relationship between the CPU and the bus interface unit. The bus interface unit has a program address register, a 3-byte instruction queue buffer, a data address register, and a 2-byte data buffer.

The bus interface unit obtains an instruction code from memory and stores it in the instruction queue buffer, obtains data from memory and stores it in the data buffer, or writes the data from the data buffer to the memory.

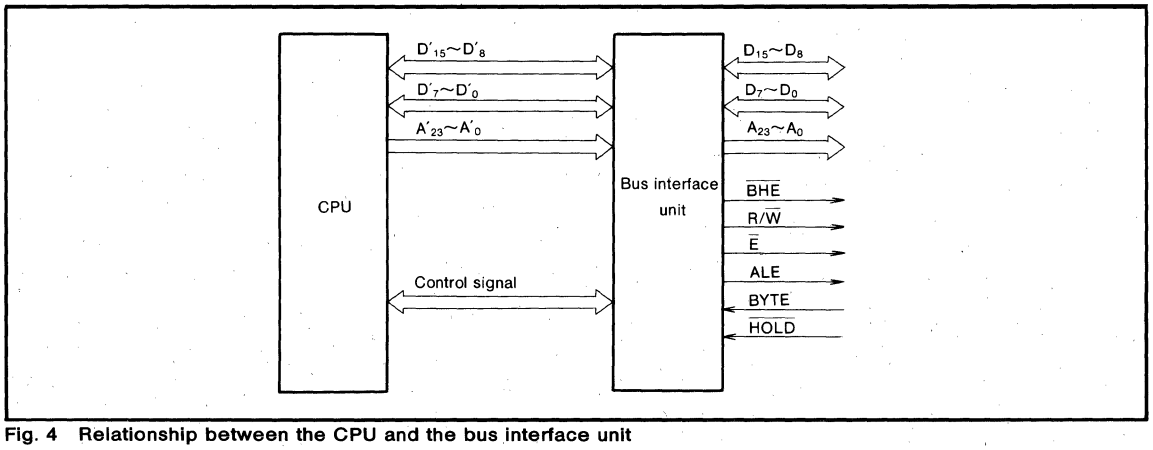


Fig. 4 Relationship between the CPU and the bus interface unit

The bus interface unit operates using one of the waveforms (1) to (6) shown in Figure 5. The standard waveforms are (1) and (2).

The ALE signal is used to latch only the address signal from the multiplexed signal containing data and address.

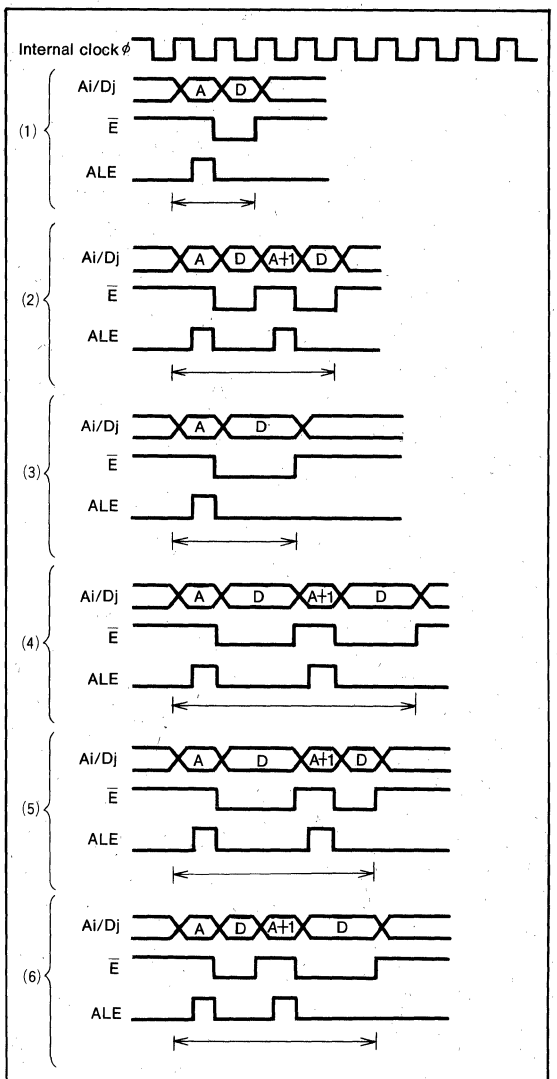
The $\overline{E}$ signal becomes "L" when the bus interface unit reads an instruction code or data from memory or when it writes data to memory. Whether to perform read or write is controlled by the R/$\overline{W}$ signal. Read is performed when the R/$\overline{W}$ signal is "H" state and write is performed when it is "L" state.

Waveform (1) in Figure 5 is used to access a single byte or two bytes simultaneously. To read or write two bytes simultaneously, the first address accessed must be even. Furthermore, when accessing an external memory area, set the bus width selection input pin BYTE to "L". (external data bus width to 16 bits) The internal memory area is always treated as 16-bit bus width regardless of BYTE.

When performing 16-bit data read or write, if the conditions for simultaneously accessing two bytes are not satisfied, waveform (2) is used to access each byte one by one. However, when prefetching the instruction code, if the address of the instruction code is odd, waveform (1) is used, and only one byte is read in the instruction queue buffer.

The signals A_0 and $\overline{BHE}$ in Figure 5 are used to control these cases: 1-byte read from even address, 1-byte read from odd address, 2-byte simultaneous read from even and odd addresses, 1-byte write to even address, 1-byte write to odd address, or 2-byte simultaneous write to even and odd addresses. The A_0 signal that is the address bit 0 is "L" when an even number address is accessed. The $\overline{BHE}$ signal becomes "L" when an odd number address is accessed.

The bit 2 of processor mode register (address $5E_{16}$) is the wait bit. When this bit is set to "0", the "L" width of $\overline{E}$ signal is 2 times as long when accessing an external memory area. However, the "L" width of $\overline{E}$ signal is not extended when an internal memory area is accessed. When the wait bit is "1", the "L" width of $\overline{E}$ signal is not extended for any access. Waveform (3) is an expansion of the "L" width of $\overline{E}$ signal in waveform (1). Waveform (4),(5), and (6) are expansion of each "L" width of $\overline{E}$ signal in waveform (2), first half of waveform (2), and the last half of waveform (2) respectively.



Signal \ Access method	Access 2-byte simultaneously	Access even address 1-byte	Access odd address 1-byte
A_0	"L"	"L"	"H"
$\overline{BHE}$	"L"	"H"	"L"

Fig. 5 Relationship between access method and signals A_0 and $\overline{BHE}$

Instruction code read, data read, and data write are described below.

Instruction code read will be described first.

The CPU obtains instruction codes from the instruction queue buffer and executes them. The CPU notifies the bus interface unit that it is requesting an instruction code during an instruction code request cycle. If the requested instruction code is not yet stored in the instruction queue buffer, the bus interface unit halts the CPU until it can store more instructions than requested in the instruction queue buffer.

Even if there is no instruction code request from the CPU, the bus interface unit reads instruction codes from memory and stores them in the instruction queue buffer when the instruction queue buffer is empty or when only one instruction code is stored and the bus is idle on the next cycle.

This is referred to as instruction pre-fetching.

Normally, when reading an instruction code from memory, if the accessed address is even the next odd address is read together with the instruction code and stored in the instruction queue buffer.

However, if the bus width switching pin BYTE is "H", external data bus width is 8 bits and the address to be read is in external memory area is odd, only one byte is read and stored in the instruction queue buffer. Therefore, waveform (1) or (3) in Figure 5 is used for instruction code read.

Data read and write are described below.

The CPU notifies the bus interface unit when performing data read or write. At this time, the bus interface unit halts the CPU if the bus interface unit is already using the bus or if there is a request with higher priority. When data read or write is enabled, the bus interface unit uses one of the waveforms from (1) to (6) in Figure 5 to perform the operation.

During data read, the CPU waits until the entire data is stored in the data buffer. The bus interface unit sends the address received from the CPU to the address bus. Then it reads the memory when the $\overline{E}$ signal is "L" and stores the result in the data buffer.

During data write, the CPU writes the data in the data buffer and the bus interface unit writes it to memory. Therefore, the CPU can proceed to the next step without waiting for write to complete. The bus interface unit sends the address received from the CPU to the address bus. Then when the $\overline{E}$ signal is "L", the bus interface unit sends the data in the data buffer to the data bus and writes it to memory.

INTERRUPTS

Table 1 shows the interrupt types and the corresponding interrupt vector addresses. Reset is also treated as a type of interrupt and is discussed in this section, too.

$\overline{DBC}$ is an interrupt used during debugging.

Interrupts other than reset, $\overline{DBC}$, watchdog timer, zero divide, and BRK instruction all have interrupt control registers. Table 2 shows the addresses of the interrupt control registers and Figure 6 shows the bit configuration of the interrupt control register.

The interrupt request bit is automatically cleared by the hardware during reset or when processing an interrupt. Also, interrupt request bits other than $\overline{DBC}$ and watchdog timer can be cleared by software.

$\overline{INT_2}$ to $\overline{INT_0}$ are external interrupts and whether to cause an interrupt at the input level (level sense) or at the edge (edge sense) can be selected with the level sense/edge sense selection bit. Furthermore, the polarity of the interrupt input can be selected with polarity selection bit.

Timer and UART interrupts are described in the respective section.

The priority of interrupts when multiple interrupts are caused simultaneously is partially fixed by hardware, but, it can also be adjusted by software as shown in Figure 7. The hardware priority is fixed the following:

reset > $\overline{DBC}$ > watchdog timer > other interrupts

Table 1. Interrupt types and the interrupt vector addresses

Interrupts	Vector addresses	
UART0 transmit	$00FFDC_{16}$	$00FFDD_{16}$
UART0 receive	$00FFDE_{16}$	$00FFDF_{16}$
Timer B0	$00FFE4_{16}$	$00FFE5_{16}$
Timer A4	$00FFE6_{16}$	$00FFE7_{16}$
Timer A3	$00FFE8_{16}$	$00FFE9_{16}$
Timer A2	$00FFEA_{16}$	$00FFEB_{16}$
Timer A1	$00FFEC_{16}$	$00FFED_{16}$
Timer A0	$00FFEE_{16}$	$00FFEF_{16}$
$\overline{INT_2}$ external interrupt	$00FFF0_{16}$	$00FFF1_{16}$
$\overline{INT_1}$ external interrupt	$00FFF2_{16}$	$00FFF3_{16}$
$\overline{INT_0}$ external interrupt	$00FFF4_{16}$	$00FFF5_{16}$
Watchdog timer	$00FFF6_{16}$	$00FFF7_{16}$
$\overline{DBC}$ (unusable)	$00FFF8_{16}$	$00FFF9_{16}$
Break instruction	$00FFFA_{16}$	$00FFFB_{16}$
Zero divide	$00FFFC_{16}$	$00FFFD_{16}$
Reset	$00FFFE_{16}$	$00FFFF_{16}$

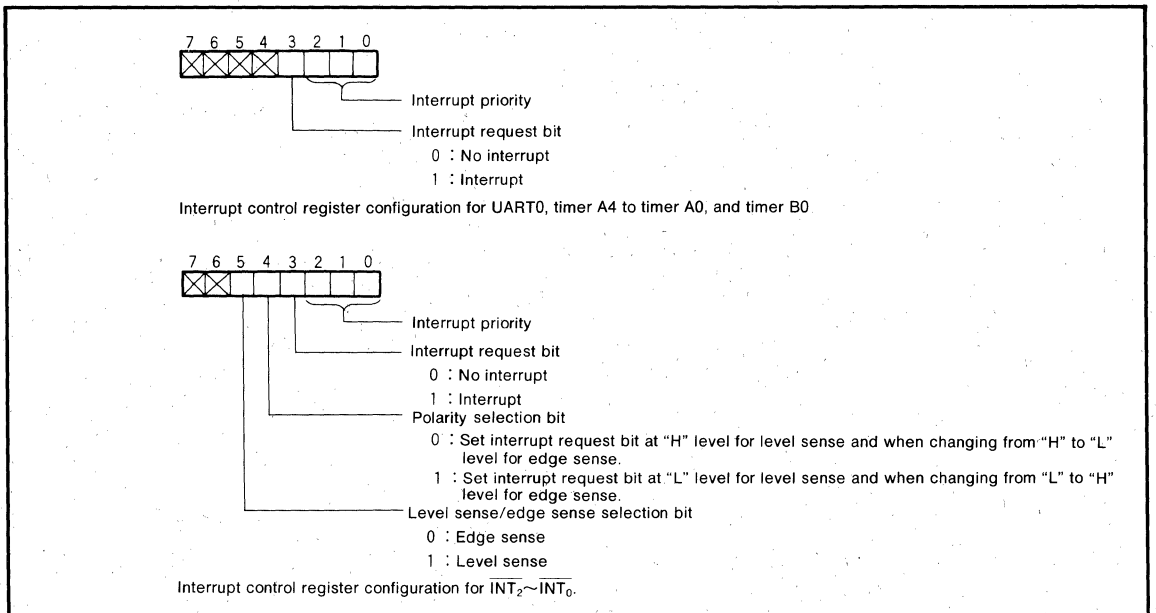


Fig. 6 Interrupt control register configuration

Table 2. Addresses of interrupt control registers

Interrupt control registers	Addresses
UART0 transmit interrupt control register	000071_{16}
UART0 receive interrupt control register	000072_{16}
Timer A0 interrupt control register	000075_{16}
Timer A1 interrupt control register	000076_{16}
Timer A2 interrupt control register	000077_{16}
Timer A3 interrupt control register	000078_{16}
Timer A4 interrupt control register	000079_{16}
Timer B0 interrupt control register	$00007A_{16}$
$\overline{INT_0}$ interrupt control register	$00007D_{16}$
$\overline{INT_1}$ interrupt control register	$00007E_{16}$
$\overline{INT_2}$ interrupt control register	$00007F_{16}$

Interrupts caused by a BRK instruction and when dividing by zero are software interrupts and are not included in this list.

Other interrupts previously mentioned are UART, Timer, INT interrupts. The priority of these interrupts can be changed by changing the priority level in the corresponding interrupt control register by software.

Figure 8 shows a diagram of the interrupt priority resolution circuit. When an interrupt is caused, the each interrupt device compares its own priority with the priority from above and if its own priority is higher, then it sends the priority below and requests the interrupt. If the priorities are the same, the one above has priority.

This comparison is repeated to select the interrupt with the highest priority among the interrupts that are being requested. Finally the selected interrupt is compared with the processor interrupt priority level (IPL) contained in the processor status register (PS) and the request is accepted if it is higher than IPL and the interrupt disable flag I is "0". The request is not accepted if flag I is "1". The reset, $\overline{DBC}$, and watchdog timer interrupts are not affected by the interrupt disable flag I.

When an interrupt is accepted, the contents of the processor status register (PS) is saved to the stack and the interrupt disable flag I is set to "1".

Furthermore, the interrupt request bit of the accepted interrupt is cleared to "0" and the processor interrupt priority level (IPL) in the processor status register (PS) is replaced by the priority level of the accepted interrupt.

Therefore, multi-level priority interrupts are possible by resetting the interrupt disable flag I to "0" and enable further interrupts.

For reset, $\overline{DBC}$, watchdog timer, zero divide, and BRK instruction interrupts, which do not have an interrupt control register, the processor interrupt level (IPL) is set as shown in Table 3.

Priority resolution is performed by latching the interrupt request bit and interrupt priority level so that they do not change. They are sampled at the first half and latched at the last half of the operation code fetch cycle.

Because priority resolution takes some time, no sampling pulse is generated for a certain interval even if it is the next operation code fetch cycle.

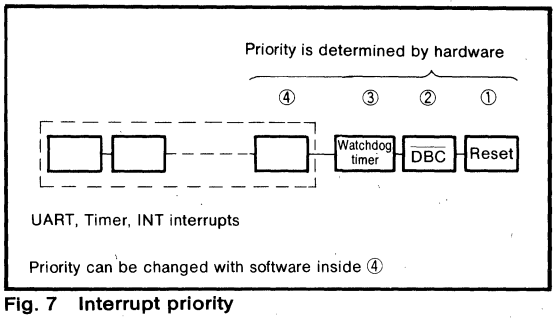


Fig. 7 Interrupt priority

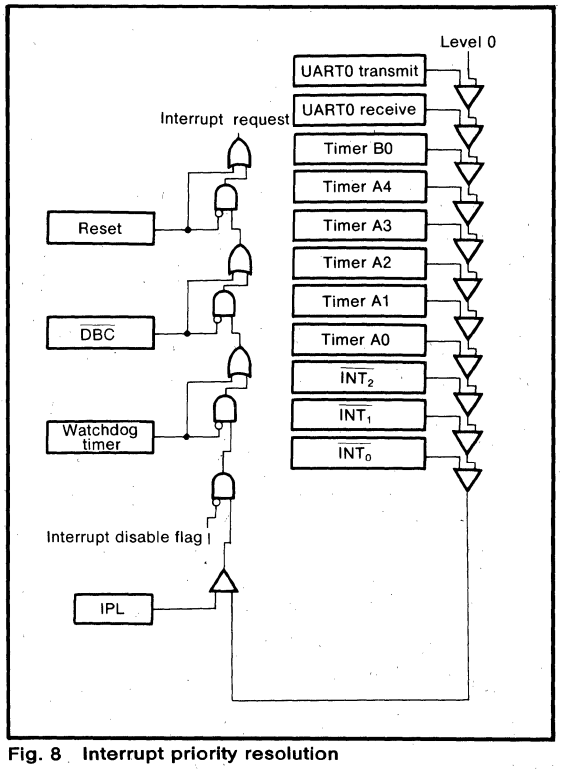


Fig. 8 Interrupt priority resolution

As shown in Figure 9, there are three different interrupt priority resolution time from which one is selected by software. After the selected time has elapsed, the highest priority is determined and is processed after the currently executing instruction has been completed.

The time is selected with bits 4 and 5 of the processor mode register (address $5E_{16}$) shown in Figure 10. Table 4 shows the relationship between these bits and the number of cycles. After a reset, the processor mode register is initialized to "00_{16}" and therefore, the longest time is selected.

However, the shortest time may be selected by software.

Table 3. Value set in processor interrupt level (IPL) during an interrupt

Interrupt types	Setting value
Reset	0
DBC	7
Watchdog timer	7
Zero divide	Not change value of IPL.
BRK instruction	Not change value of IPL.

Table 4. Relationship between priority level evaluation time selection bit and number of cycles

Priority level resolution time selection bit		Number of cycles
Bit 5	Bit 4	
0	0	7 cycles of ϕ
0	1	4 cycles of ϕ
1	0	2 cycles of ϕ

ϕ : internal clock

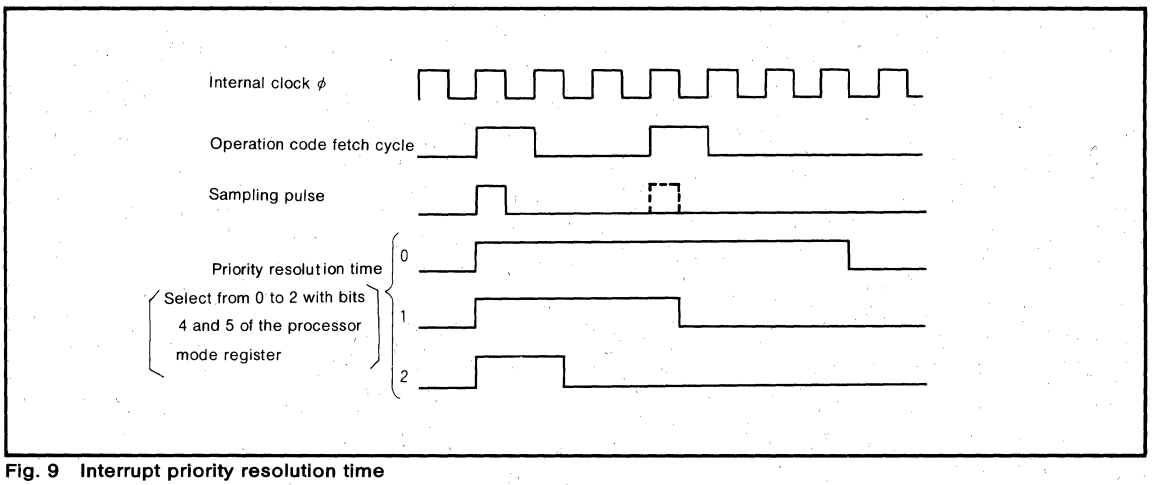


Fig. 9 Interrupt priority resolution time

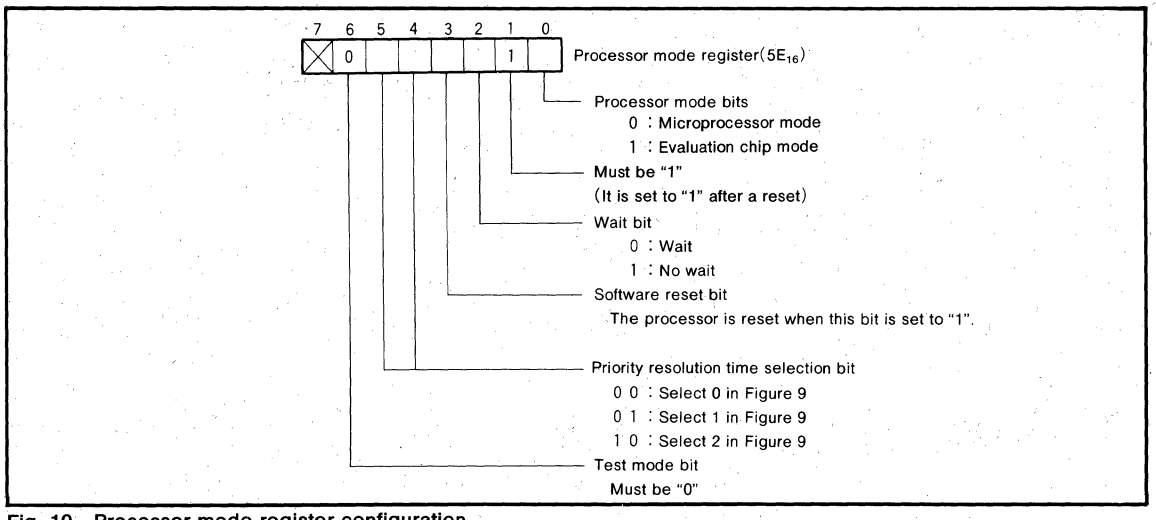


Fig. 10 Processor mode register configuration

TIMER

There are six 16-bit timers. They are divided by type into timer A(5) and timer B(1).

The timer I/O pins are shared with I/O pins for port P5 and P6. To use these pins as timer input pins, the data direction register bit corresponding to the pin must be cleared to "0" to specify input mode.

TIMER A

Figure 11 shows a block diagram of timer A.

Timer A has four modes; timer mode, event counter mode, one-shot pulse mode, and pulse width modulation mode. The mode is selected with bits 0 and 1 of the timer Ai mode register (i = 0 to 4). Each of these modes is described below.

(1) Timer mode (00)

Figure 12 shows the bit configuration of the timer Ai mode register during timer mode. Bits 0, 1, and 5 of the timer Ai mode register must always be "0" in timer mode.

Bit 3 is ignored if bit 4 is "0".

Bits 6 and 7 are used to select the timer counter source.

The counting of the selected clock starts when the count start flag is "1" and stops when it is "0".

Figure 13 shows the bit configuration of the count start flag.

The counter is decremented, an interrupt is caused and the interrupt request bit in the timer Ai interrupt control register is set when the contents becomes 0000_{16}. At the same time, the contents of the reload register is transferred to the counter and count is continued.

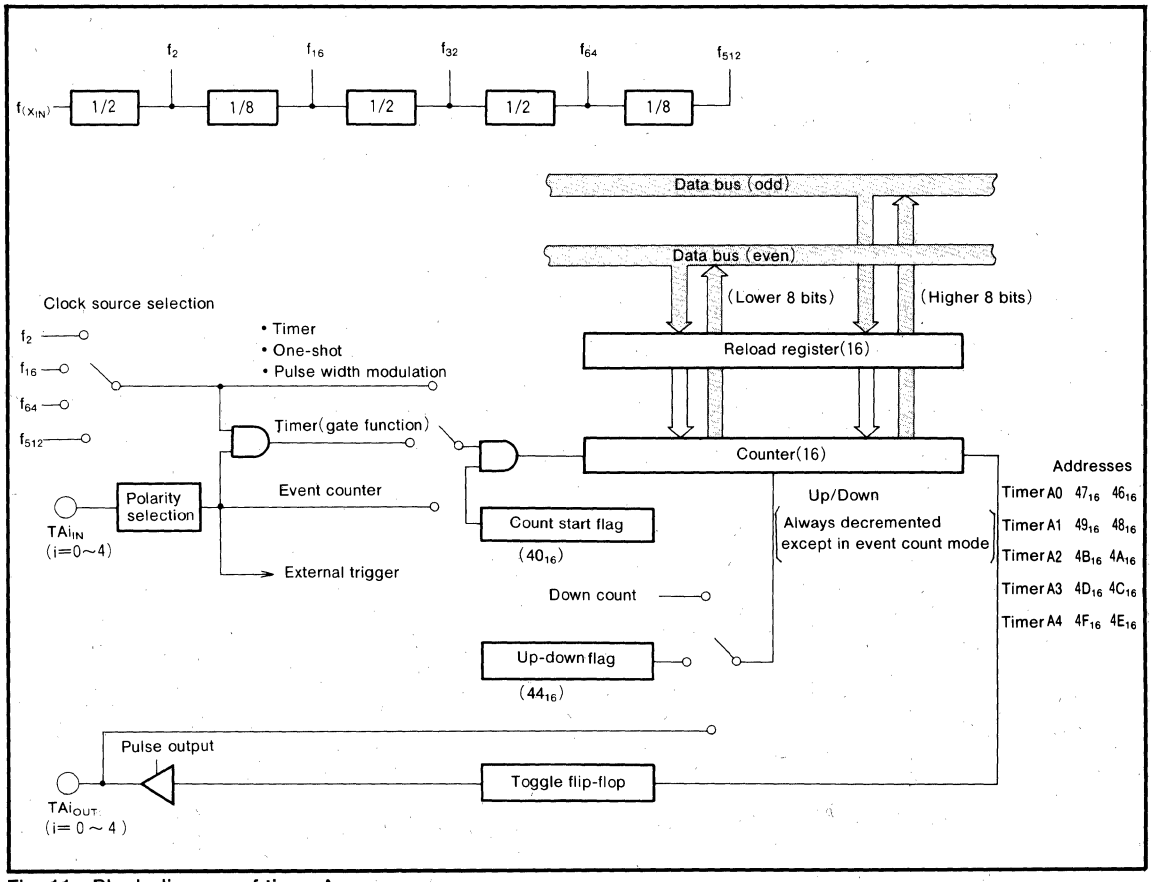


Fig. 11 Block diagram of timer A

When bit 2 of the timer Ai mode register is "1", the output is generated from TAi_{OUT} pin. The output is toggled each time the contents of the counter reaches to 0000_{16}. When the contents of the count start flag is "0", "L" is output from TAi_{OUT} pin.

When bit 2 is "0", TAi_{OUT} can be used as a normal port pin.

When bit 4 is "0", TAi_{IN} can be used as a normal port pin.

When bit 4 is "1", counting is performed only while the input signal from the TAi_{IN} pin is "H" or "L" as shown in Figure 14. Therefore, this can be used to measure the pulse width of the TAi_{IN} input signal. Whether to count while the input signal is "H" or while it is "L" is determined by bit 3. If bit 3 is "1", counting is performed while the TAi_{IN} pin input signal is "H" and if bit 3 is "0", counting is performed while it is "L".

Note that the duration of "H" or "L" on the TAi_{IN} pin must be two or more cycles of the timer count source.

When data is written to timer Ai register with timer Ai halted, the same data is also written to the reload register and the counter. When data is written to timer Ai which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time. The contents of the counter can be read at any time.

When the value set in the timer Ai register is n, the timer frequency dividing ratio is $1/(n+1)$.

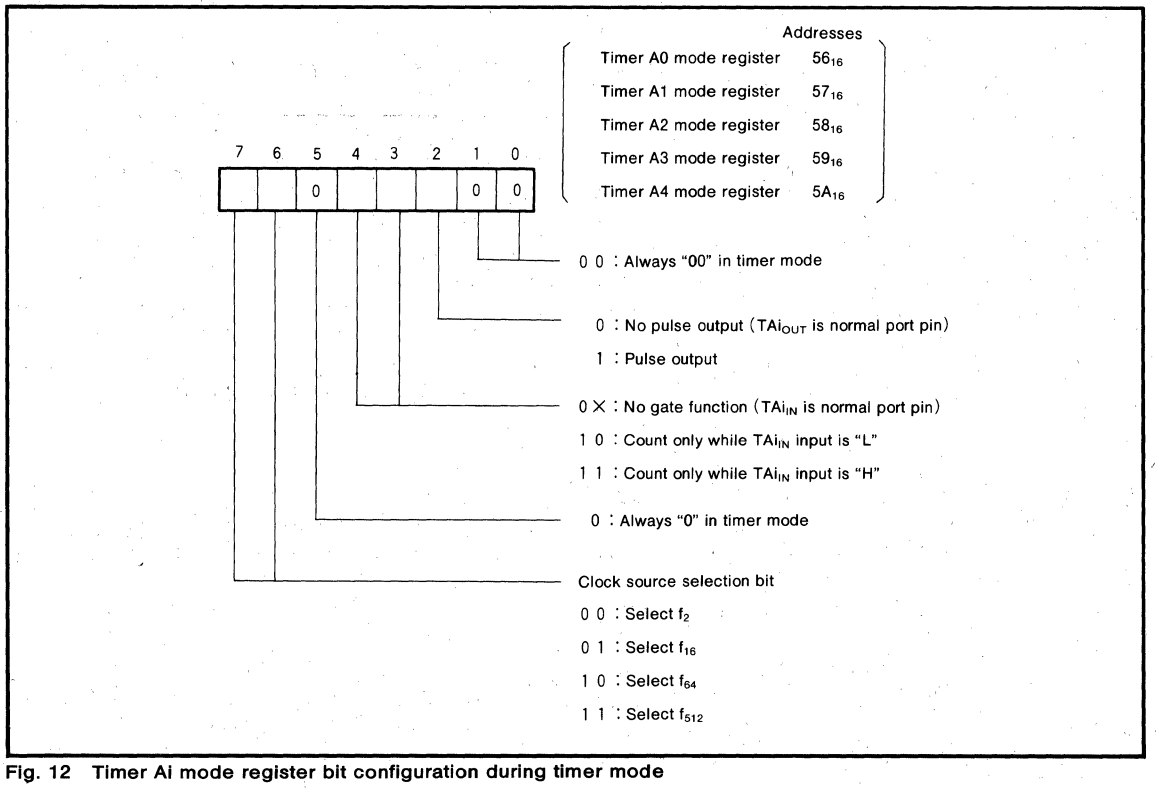


Fig. 12 Timer Ai mode register bit configuration during timer mode

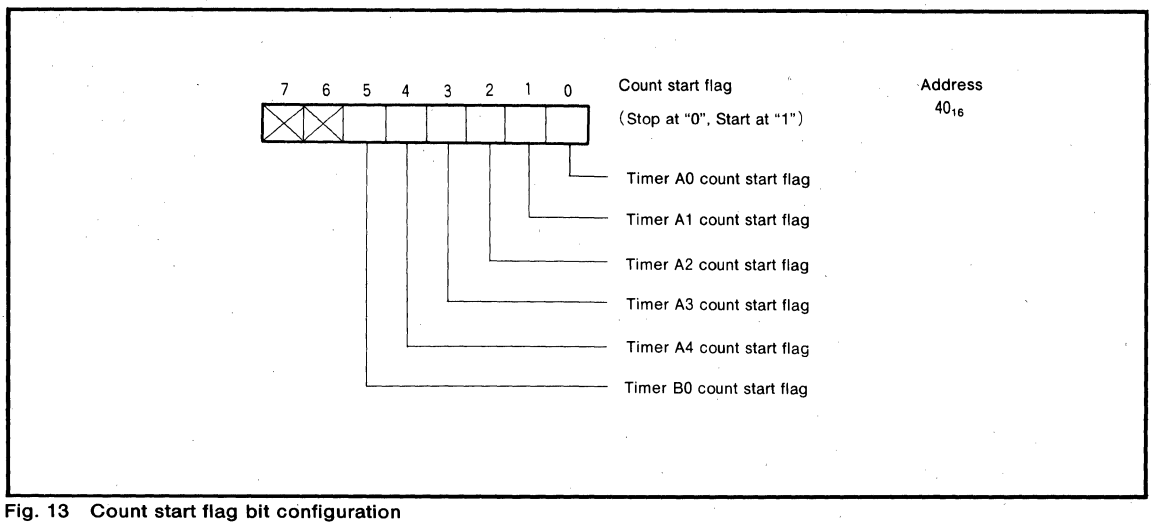


Fig. 13 Count start flag bit configuration

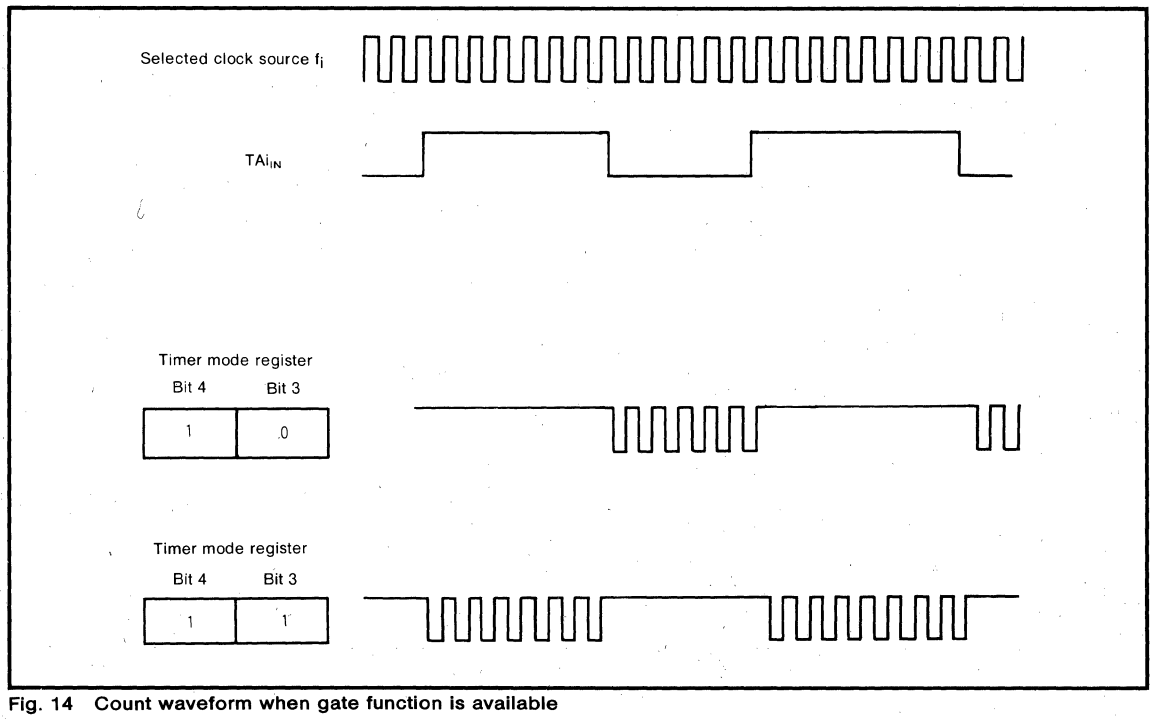


Fig. 14 Count waveform when gate function is available

(2) Event counter mode (01)

Figure 15 shows the bit configuration of the timer Ai mode register during event counter mode. In event counter mode, the bit 0 of the timer Ai mode register must be "1" and bit 1 and 5 must be "0".

The input signal from the TAi_{IN} pin is counted when the count start flag shown in Figure 13 is "1" and counting is stopped when it is "0".

Count is performed at the fall of the input signal when bit 3 is "0" and at the rise of the signal when it is "1".

In event counter mode, whether to increment or decrement the count can be selected with the up-down flag or the input signal from the TAi_{OUT} pin.

When bit 4 of the timer Ai mode register is "0", the up-down flag is used to determine whether to increment or decrement the count (decrement when the flag is "0" and increment when it is "1"). Figure 16 shows the bit configuration of the up-down flag.

When bit 4 of the timer Ai mode register is "1", the input signal from the TAi_{OUT} pin is used to determine whether to increment or decrement the count. However, note that bit 2 must be "0" if bit 4 is "1" because if bit 2 is "1", TAi_{OUT} pin becomes an output pin with pulse output.

The count is decremented when the input signal from the TAi_{OUT} pin is "L" and incremented when it is "H". Determine the level of the input signal from the TAi_{OUT} pin before valid edge is input to the TAi_{IN} pin.

An interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set when the counter reaches 0000_{16} (decrement count) or $FFFF_{16}$ (increment count). At the same time, the contents of the reload register is transferred to the counter and the count is continued.

When bit 2 is "1" and the counter reaches 0000_{16} (decrement count) or $FFFF_{16}$ (increment count), the waveform reversing polarity is output from TAi_{OUT} pin.

If bit 2 is "0", TAi_{OUT} pin can be used as a normal port pin. However, if bit 4 is "1" and the TAi_{OUT} pin is used as an output pin, the output from the pin changes the count direction. Therefore, bit 4 should be "0" unless the output from the TAi_{OUT} pin is to be used to select the count direction.

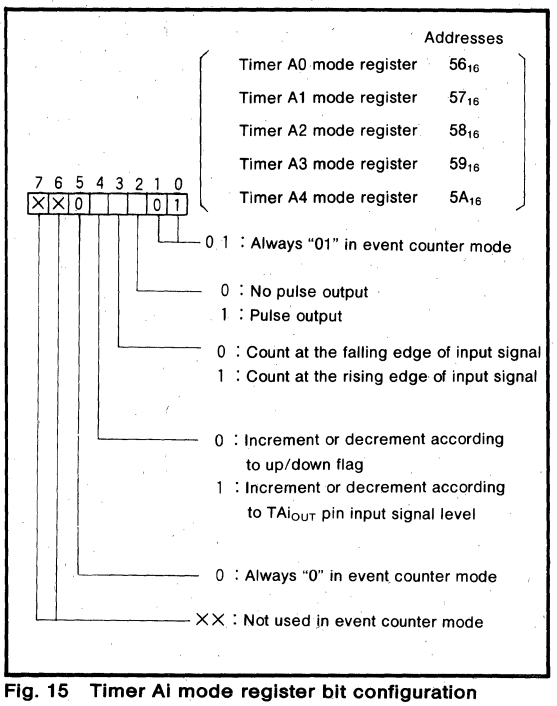


Fig. 15 Timer Ai mode register bit configuration during event counter mode

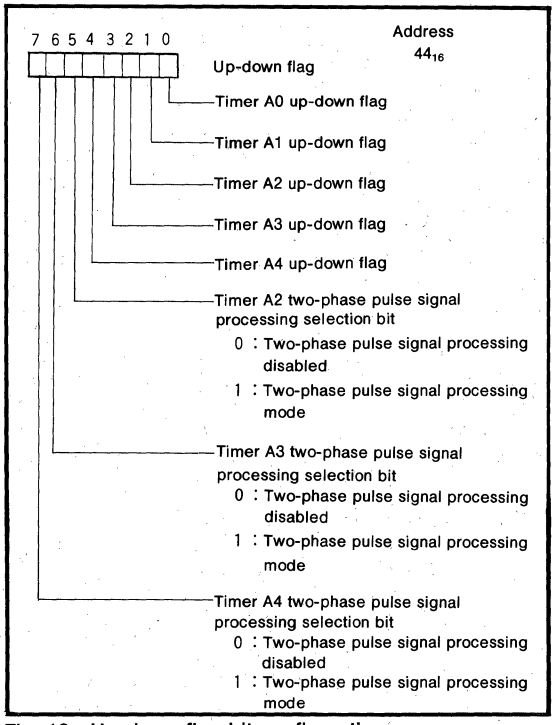


Fig. 16 Up-down flag bit configuration

Data write and data read are performed in the same way as for timer mode. That is, when data is written to timer Ai halted, it is also written to the reload register and the counter. When data is written to timer Ai which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time. The counter can be read at any time.

In event counter mode, whether to increment or decrement the counter can also be determined by supplying two-phase pulse input with phase shifted by 90° to timer A2, A3, or A4. There are two types of two-phase pulse processing operations. One uses timers A2 and A3, and the other uses timer A4. In either processing operation, two-phase pulse is input in the same way, that is, pulses out of phase by 90° are input at the TAj_{OUT} (j=2 to 4) pin and TAj_{IN} pin.

When timers A2 and A3 are used, as shown in Figure 17, the count is incremented when a rising edge is input to the TAk_{IN} pin after the level of TAk_{OUT} (k=2, 3) pin changes from "L" to "H", and when the falling edge is inserted, the count is decremented.

For timer A4, as shown in Figure 18, when a phase related pulse with a rising edge input to the $TA4_{IN}$ pin is input after the level of $TA4_{OUT}$ pin changes from "L" to "H", the count is incremented at the respective rising edge and falling edge of the $TA4_{OUT}$ pin and $TA4_{IN}$ pin.

When a phase related pulse with a falling edge input to the $TA4_{OUT}$ pin is input after the level of $TA4_{IN}$ pin changes from "H" to "L", the count is decremented at the respective rising edge and falling edge of the $TA4_{IN}$ pin and $TA4_{OUT}$ pin. When performing this two-phase pulse signal processing, timer Aj mode register bit 0 and bit 4 must be set to "1" and bits 1, 2, 3, and 5 must be "0". Bits 6 and 7 are ignored. Note that bits 5, 6, and 7 of the up-down flag register (44_{16}) are the two-phase pulse signal processing selection bit for timer A2, A3, and A4 respectively. Each timer operates in normal event counter mode when the corresponding bit is "0" and performs two-phase pulse signal processing when it is "1".

Count is started by setting the count start flag to "1". Data write and read are performed in the same way as for normal event counter mode. Note that the direction register of the input port must be set to input mode because two-phase pulse signal is input. Also, there can be no pulse output in this mode.

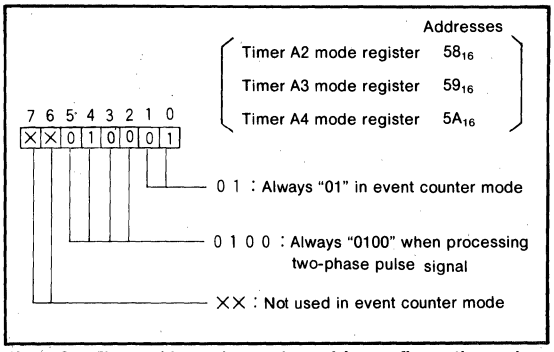


Fig. 19 Timer Aj mode register bit configuration when performing two-phase pulse signal processing in event counter mode

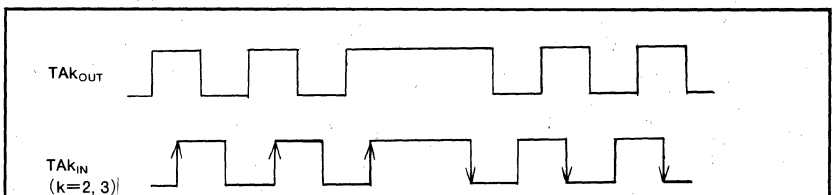


Fig. 17 Two-phase pulse processing operation of timer A2 and timer A3

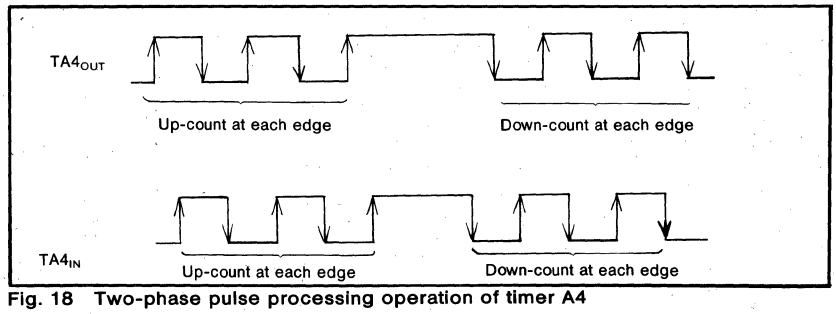


Fig. 18 Two-phase pulse processing operation of timer A4

(3) One-shot pulse mode (10)

Figure 20 shows the bit configuration of the timer Ai mode register during one-shot pulse mode. In one-shot pulse mode, bit 0 and bit 5 must be "0" and bit 1 and bit 2 must be "1".

The trigger is enabled when the count start flag is "1". The trigger can be generated by software or it can be input from the TAi_{IN} pin. Software trigger is selected when bit 4 is "0" and the input signal from the TAi_{IN} pin is used as the trigger when it is "1".

Bit 3 is used to determine whether to trigger at the fall of the trigger signal or at the rise. The trigger is at the fall of the trigger signal when bit 3 is "0" and at the rise of the trigger signal when it is "1".

Software trigger is generated by setting the bit in the one-shot start flag corresponding to each timer.

Figure 21 shows the bit configuration of the one-shot start flag.

As shown in Figure 22, when a trigger signal is received, the counter counts the clock selected by bits 6 and 7.

If the contents of the counter is not 0000_{16}, the TAi_{OUT} pin goes "H" when a trigger signal is received. The count direction is decrement.

When the counter reaches 0001_{16}, The TAi_{OUT} pin goes "L" and count is stopped. The contents of the reload register is transferred to the counter. At the same time, an interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set. This is repeated each time a trigger signal is received. The output pulse width is

$$\frac{1}{\text{pulse frequency of the selected clock}} \times (\text{counter's value at the time of trigger}).$$

If the count start flag is "0", TAi_{OUT} goes "L". Therefore, the value corresponding to the desired pulse width must be written to timer Ai before setting the timer Ai count start flag.

As shown in Figure 23, a trigger signal can be received before the operation for the previous trigger signal is completed. In this case, the contents of the reload register is transferred to the counter by the trigger and then that value is decremented.

Except when retriggering while operating, the contents of the reload register is not transferred to the counter by triggering.

When retriggering, there must be at least one timer count source cycle before a new trigger can be issued.

Data write is performed to the same way as for timer mode. When data is written in timer Ai halted, it is also written to the reload register and the counter.

When data is written to timer Ai which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time.

Undefined data is read when timer Ai is read.

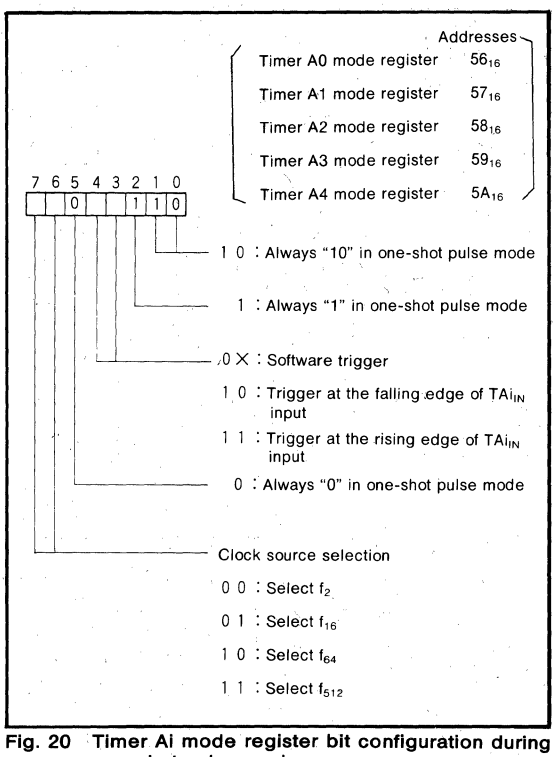


Fig. 20 Timer Ai mode register bit configuration during one-shot pulse mode

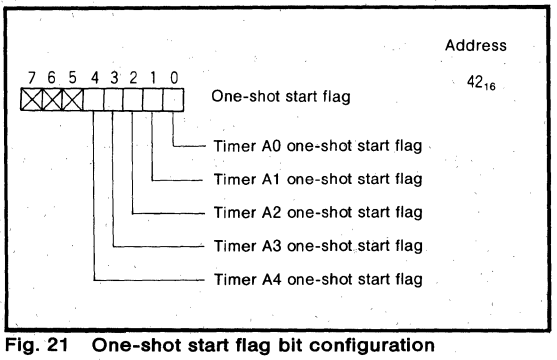


Fig. 21 One-shot start flag bit configuration

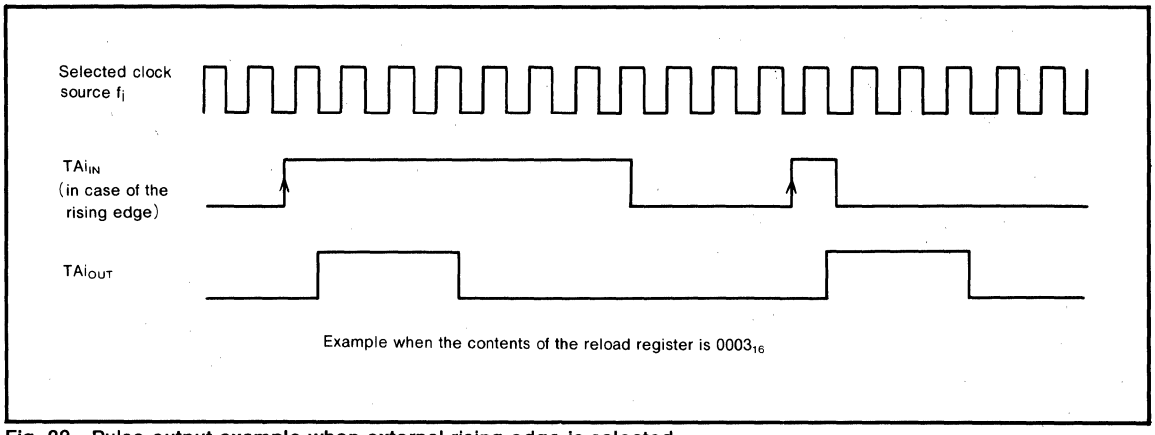


Fig. 22 Pulse output example when external rising edge is selected

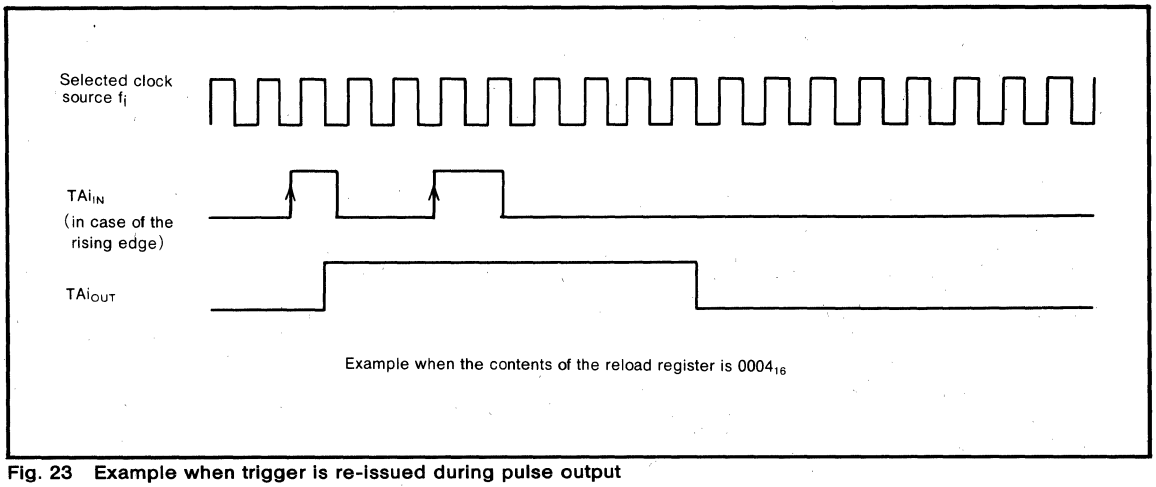


Fig. 23 Example when trigger is re-issued during pulse output

(4) Pulse width modulation mode (11)

Figure 24 shows the bit configuration of the timer Ai mode register during pulse width modulation mode. In pulse width modulation mode, bits 0, 1, and 2 must be set to "1".

Bit 5 is used to determine whether to perform 16-bit length pulse width modulator or 8-bit length pulse width modulator. 16-bit length pulse width modulator is performed when bit 5 is "0" and 8-bit length pulse width modulator is performed when it is "1". The 16-bit length pulse width modulator is described first.

The pulse width modulator can be started with a software trigger or with an input signal from a TAi_{IN} pin (external trigger).

The software trigger mode is selected when bit 4 is "0". Pulse width modulator is started and pulse is output from TAi_{OUT} when the timer Ai start flag is set to "1".

The external trigger mode is selected when bit 4 is "1". Pulse width modulator starts when a trigger signal is input from the TAi_{IN} pin when the timer Ai start flag is "1". Whether to trigger at the fall or rise of the trigger signal is determined by bit 3. The trigger is at the fall of the trigger signal when bit 3 is "0" and at the rise when it is "1".

When data is written to timer Ai with the pulse width modulator halted, it is written to the reload register and the counter.

Then when the time Ai start flag is set to "1" and a software trigger or an external trigger is issued to start modulation, the waveform shown in Figure 25 is output continuously. Once modulation is started, triggers are not accepted. If the value in the reload register is m, the duration "H" of pulse is

$$\frac{1}{\text{selected clock frequency}} \times m$$

and the output pulse period is

$$\frac{1}{\text{selected clock frequency}} \times (2^{16}-1).$$

An interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set at each fall of the output pulse.

The width of the output pulse is changed by updating timer data. The update can be performed at any time. The output pulse width is changed at the rise of the pulse after data is written to the timer.

The contents of the reload register are transferred to the counter just before the rise of the next pulse so that the pulse width is changed from the next output pulse.

Undefined data is read when timer Ai is read.

The 8-bit length pulse width modulator is described next.

The 8-bit length pulse width modulator is selected when the timer Ai mode register bit 5 is "1".

The reload register and the counter are both divided into 8-bit halves.

The low order 8 bits function as a prescaler and the high order 8 bits function as the 8-bit length pulse width modulator. The prescaler counts the clock selected by bits 6 and 7. A pulse is generated when the counter reaches 0000_{16} as shown in Figure 26. At the same time, the contents of the reload register is transferred to the counter and count is continued.

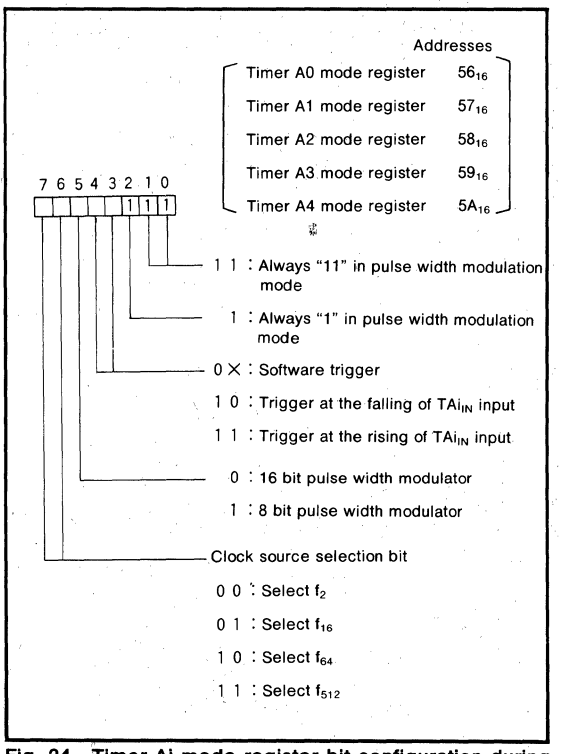


Fig. 24 Timer Ai mode register bit configuration during pulse width modulation mode

Therefore, if the low order 8-bit of the reload register is n, the period of the generated pulse is

$$\frac{1}{\text{selected clock frequency}} \times (n+1).$$

The high order 8-bit function as an 8-bit length pulse width modulator using this pulse as input. The operation is the same as for 16-bit length pulse width modulator except that the length is 8 bits. If the high order 8-bit of the reload register is m, the duration "H" of pulse is

$$\frac{1}{\text{selected clock frequency}} \times (n+1) \times m.$$

And the output pulse period is

$$\frac{1}{\text{selected clock frequency}} \times (n+1) \times (2^8-1).$$

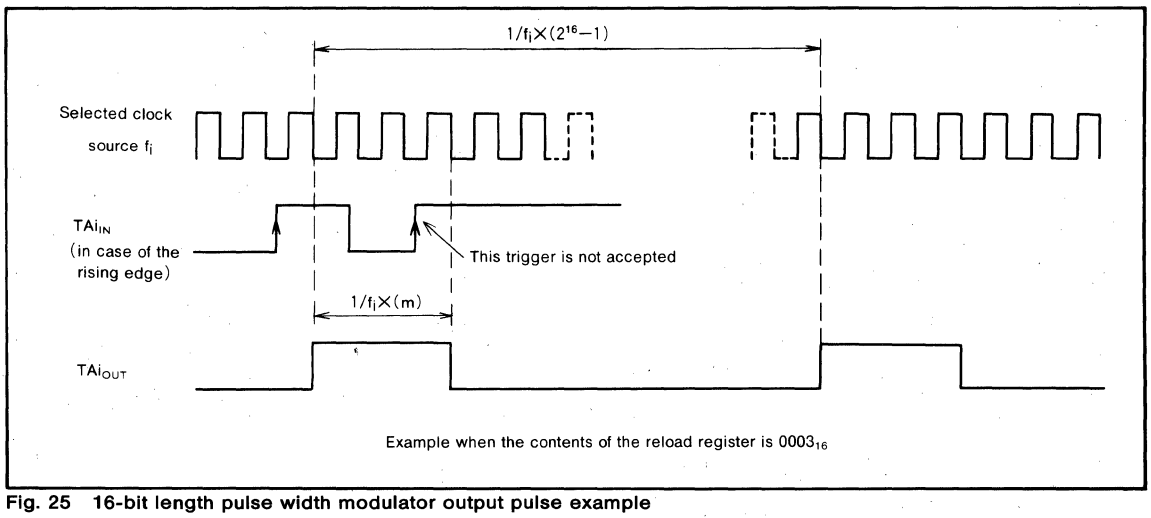


Fig. 25 16-bit length pulse width modulator output pulse example

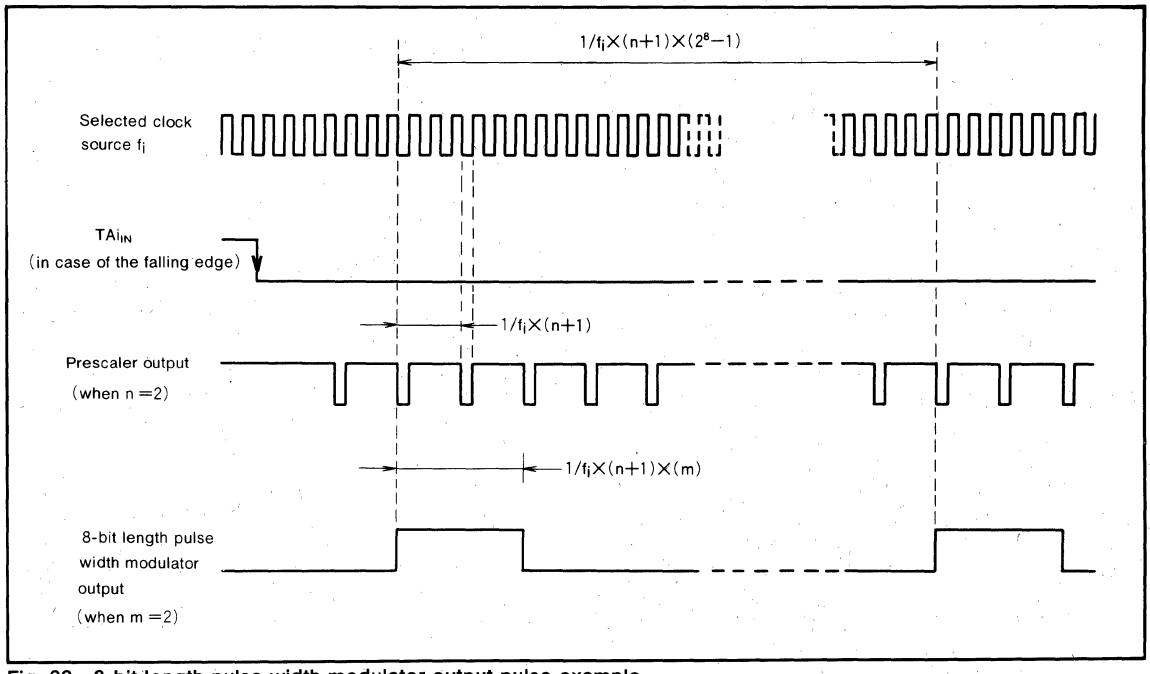


Fig. 26 8-bit length pulse width modulator output pulse example

TIMER B

Figure 27 shows a block diagram of timer B.

Timer B has three modes; timer mode, event counter mode, and pulse period measurement/pulse width measurement mode. The mode is selected with bits 0 and 1 of the timer B0 mode register. Each of these modes is described below.

(1) Timer mode (00)

Figure 28 shows the bit configuration of the timer B0 mode register during timer mode. Bits 0, and 1 of the timer B0 mode register must always be "0" in timer mode.

Bits 6 and 7 are used to select the clock source. The counting of the selected clock starts when the count start flag "1" and stops when "0".

As shown in Figure 13, the timer B0 count start flag is at the same address as the timer Ai count start flag. The count is decremented, an interrupt occurs, and the interrupt request bit in the timer B0 interrupt control register is set when the contents becomes 0000_{16}. At the same time, the contents of the reload register is stored in the counter and count is continued.

Timer B0 does not have a pulse output function or a gate function like timer A.

When data is written to timer B0 halted, it is written to the reload register and the counter. When data is written to timer B0 which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time. The contents of the counter can be read at any time.

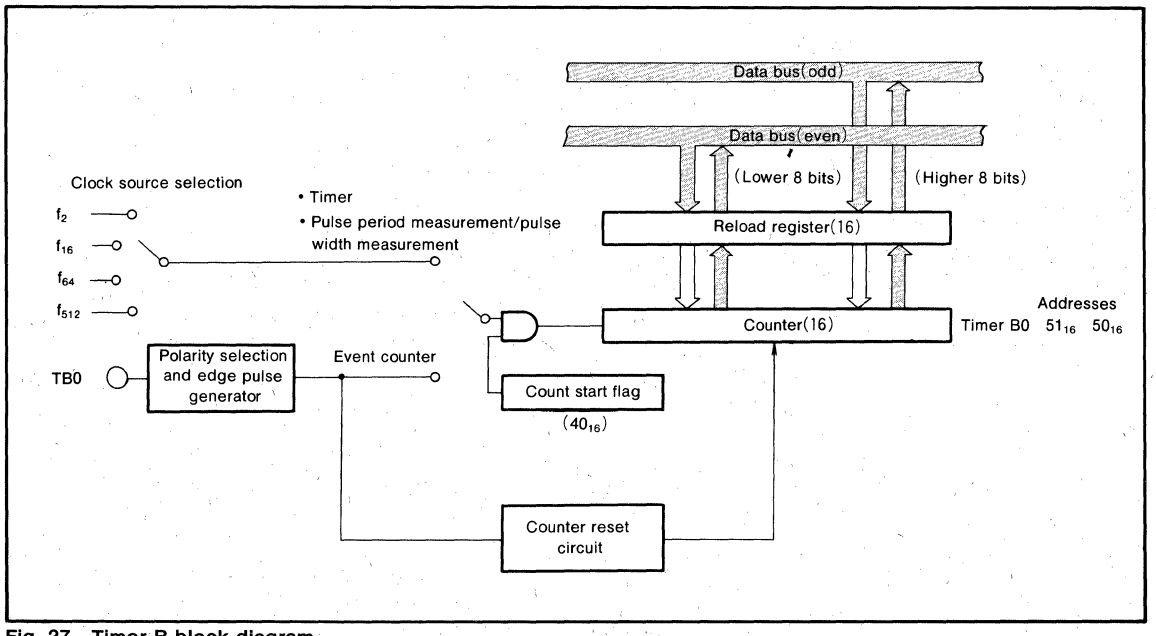


Fig. 27 Timer B block diagram

(2) Event counter mode (01)

Figure 29 shows the bit configuration of the timer B0 mode register during event counter mode. In event counter mode, the bit 0 in the timer B0 mode register must be "1" and bit 1 must be "0".

The input signal from the $TB0_{IN}$ pin is counted when the count start flag is "1" and counting is stopped when it is "0". Count is performed at the fall of the input signal when bits 2, and 3 are "0" and at the rise of the input signal when bit 3 is "0" and bit 2 is "1".

When bit 3 is "1" and bit 2 is "0", count is performed at the rise and fall of the input signal.

Data write, data read and timer interrupt are performed in the same way as for timer mode.

(3) Pulse period measurement/pulse width measurement mode (10)

Figure 30 shows the bit configuration of the timer B0 mode register during pulse period measurement/pulse width measurement mode.

In pulse period measurement/pulse width measurement mode, bit 0 must be "0" and bit 1 must be "1". Bits 6 and 7 are used to select the clock source. The selected clock is counted when the count start flag is "1" and counting stops when it is "0".

The pulse period measurement mode is selected when bit 3 is "0". In pulse period measurement mode, the selected clock is counted during the interval starting at the fall of the input signal from the $TB0_{IN}$ pin to the next fall or at the rise of the input signal to the next rise and the result is stored in the reload register. In this case, the reload register acts as a buffer register.

When bit 2 is "0", the clock is counted from the fall of the input signal to the next fall. When bit 2 is "1", the clock is counted from the rise of the input signal to the next rise.

In the case of counting from the fall of the input signal to the next fall, counting is performed as follows. As shown in Figure 31, when the fall of the input signal from $TB0_{IN}$ pin is detected, the contents of the counter is transferred to the reload register. Next the counter is cleared and count is started from the next clock. When the fall of the next input signal is detected, the contents of the counter is transferred to the reload register once more, the counter is cleared, and the count is started. The period from the fall of the input signal to the next fall is measured in this way.

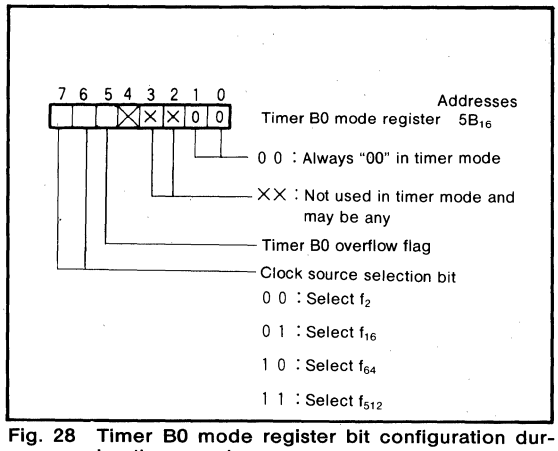


Fig. 28 Timer B0 mode register bit configuration during timer mode

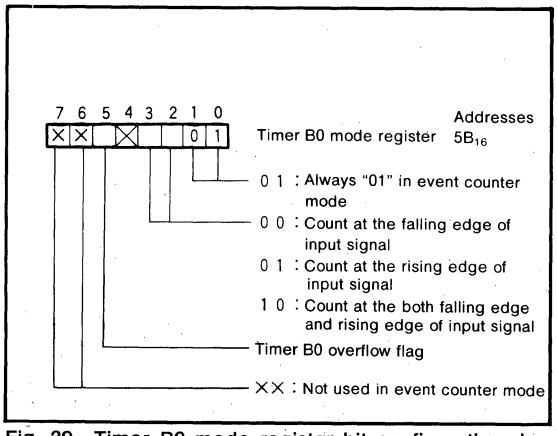


Fig. 29 Timer B0 mode register bit configuration during event counter mode

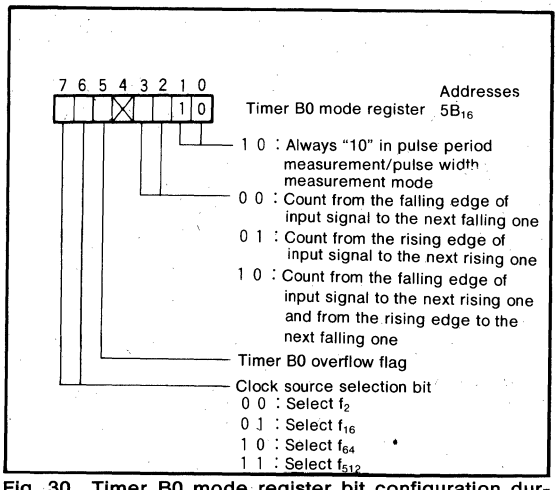


Fig. 30 Timer B0 mode register bit configuration during pulse period measurement/pulse width measurement mode

After the contents of the counter is transferred to the reload register, an interrupt request signal is generated and the interrupt request bit in the timer B0 interrupt control register is set. However, no interrupt request signal is generated when the contents of the counter is transferred first time to the reload register after the count start flag is set to "1".

When bit 3 is "1", the pulse width measurement mode is selected. Pulse width measurement mode is similar to pulse period measurement mode except that the clock is counted from the fall of the $TB0_{IN}$ pin input signal to the next rise or from the rise of the input signal to the next fall as shown in Figure 32.

When timer B0 is read, the contents of the reload register is read.

Note that in this mode, the interval between the fall of the $TB0_{IN}$ pin input signal to the next rise or from the rise to the next fall must be at least two cycles of the timer count source.

Timer B0 overflow flag which is bit 5 of timer B0 mode register is set to "1" when the timer B0 counter reaches 0000_{16}. This flag is cleared by writing to corresponding timer B0 mode register. This bit is set to "1" at reset.

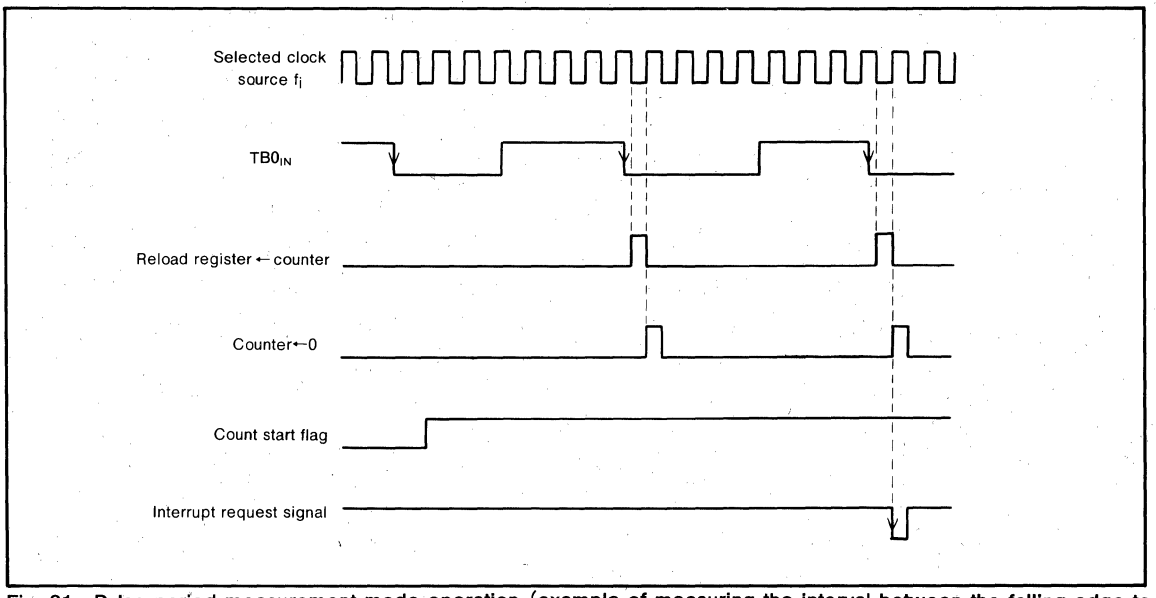


Fig. 31 Pulse period measurement mode operation (example of measuring the interval between the falling edge to next falling one)

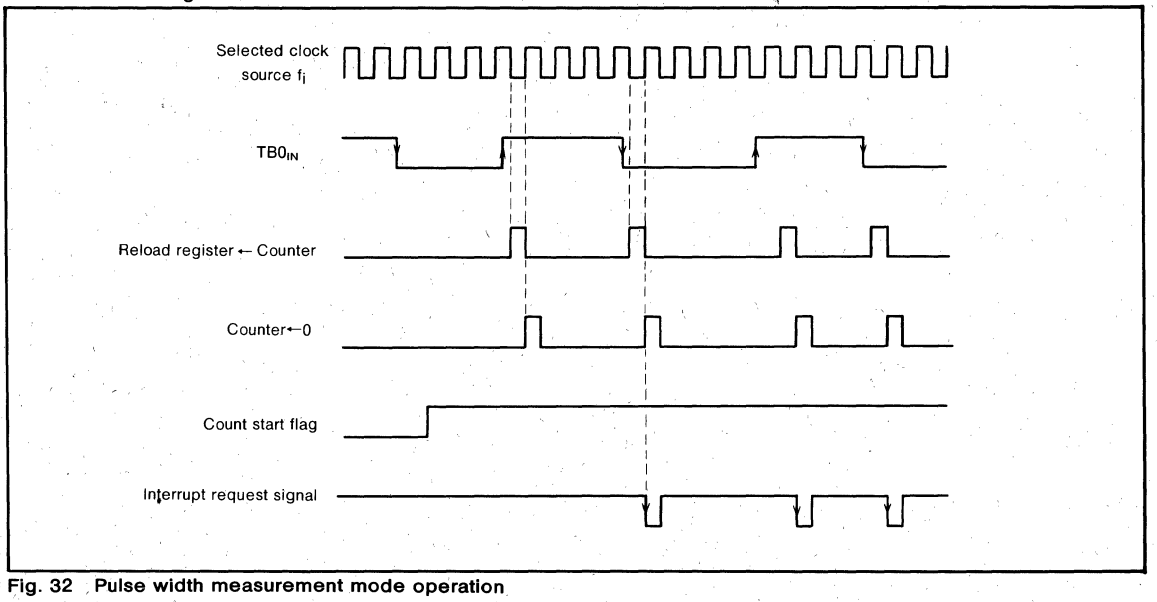


Fig. 32 Pulse width measurement mode operation

Pulse output port mode

Figure 33 shows a block diagram for pulse output port mode. In the pulse output port mode, two pairs of four-bit pulse output ports are used. Whether using pulse output port or not can be selected by waveform output selection bit (bit 0, bit 1) of waveform output mode register (62_{16} address) shown in Figure 34. When bit 0 of waveform output selection bit is set to "1", ports $P6_0$, $P5_6$, $P5_5$ and $P5_4$ are used as pulse output ports (RTP1 selected), and when bit 1 of waveform output selection bit is set to "1", ports $P5_3$, $P5_2$, $P5_1$, and $P5_0$ are used as pulse output ports (RTP0 selected). When bits 1 and 0 of waveform output selection bit are set to "1", ports $P6_0$, $P5_6$, $P5_5$, and $P5_4$, and ports $P5_3$, $P5_2$, $P5_1$ and $P5_0$ are used as pulse output ports (RTP1 and RTP0 selected).

The ports not used as pulse output ports can be used as normal parallel ports or timer input/output.

In the pulse output port mode, set timers A2 and A0 to timer mode as timers A2 and A0 are used. Figure 35 shows the bit configuration of timer A0, A2 mode registers in pulse output port mode.

Data can be set in each bit of the pulse output data register corresponding to four ports selected as pulse output ports. Figure 36 shows the bit configuration of the pulse output data register. The contents of the pulse output data register 1 (low-order four bits of 64_{16} address) corresponding to ports $P6_0$, $P5_6$, $P5_5$ and $P5_4$ is output to the ports each time the counter of timer A2 becomes 0000_{16}. The contents of the pulse output data register 0 (low-order four bits of 65_{16} address) corresponding to ports $P5_3$, $P5_2$, $P5_1$, and $P5_0$ is output to the ports each time the counter of timer A0 becomes 0000_{16}.

When "0" is written to a specified bit of the pulse output data register, "L" level is output to the corresponding pulse output port when the counter of corresponding timer becomes 0000_{16}, and when "1" is written, "H" level is output to the pulse output port.

Pulse width modulation can be applied to each pulse output port. Since pulse width modulation involves the use of timers A3 and A1, activate these timers in pulse width modulation mode. When a certain bit of the pulse output register is "1", pulse width modulation is output from the pulse output port when the counter of the corresponding timer becomes 0000_{16}.

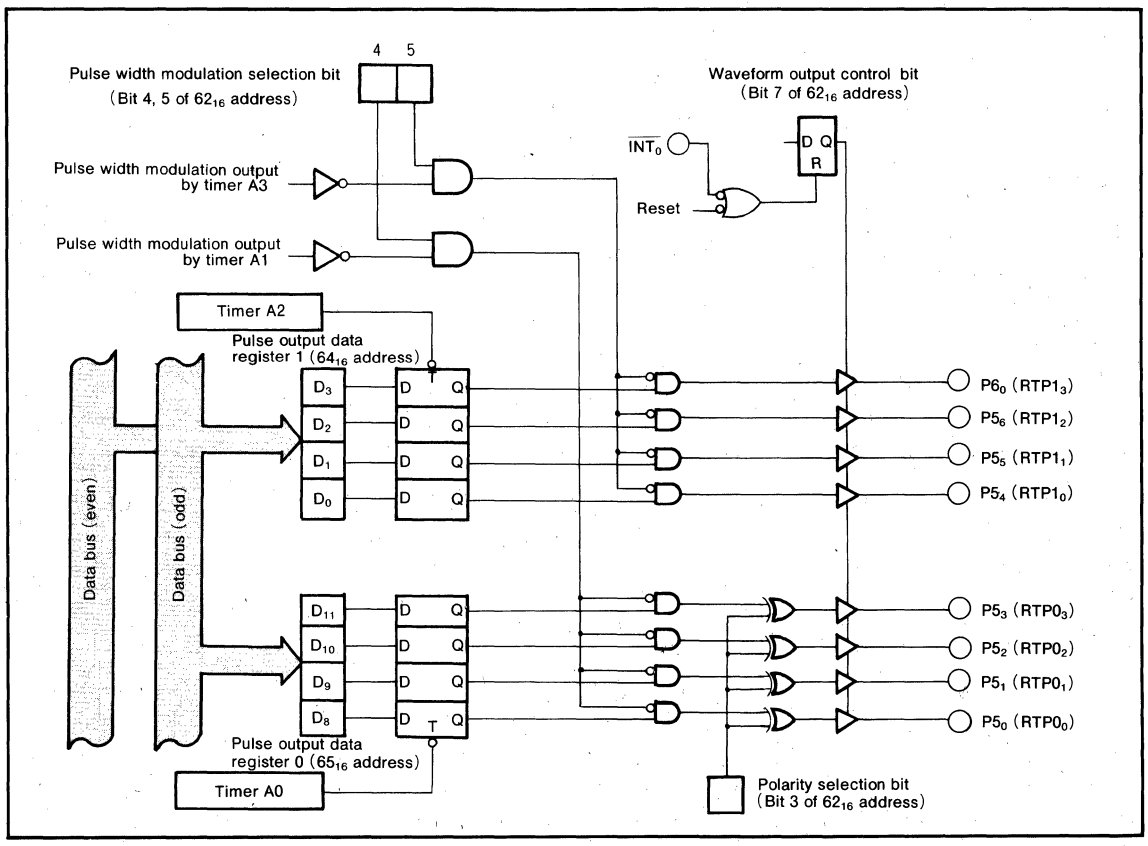


Fig. 33 Block diagram for pulse output port mode

Ports $P6_0$, $P5_6$, $P5_5$ and $P5_4$ are applied pulse width modulation by timer A3 by setting the pulse width modulation selection bit by timer A3 (bit 5) of the waveform output mode register to "1".

Ports $P5_3$, $P5_2$, $P5_1$ and $P5_0$ are applied pulse width modulation by timer A1 by setting the pulse width modulation selection bit by timer A1 (bit 4) of the waveform output mode register to "1".

The contents of the pulse output data register 0 can be reversed and output to pulse output ports $P5_3$, $P5_2$, $P5_1$ and $P5_0$ by the polarity selection bit (bit 3) of the waveform output mode register. When the polarity selection bit is "0", the contents of the pulse output data register 0 is output unchangeably, and when "1", the contents of the pulse output data register 0 is reversed and output. When pulse width modulation is applied, likewise the polarity reverse to pulse width modulation can be selected by the polarity selection bit.

Figure 37 shows example of waveforms in pulse output port mode.

Ports selecting the pulse output port mode can control output by the waveform output control bit (bit 7) of the waveform output mode register (62_{16} address).

When the waveform output control bit is set to "1", a waveform is output from the port. When this bit is set to "0", waveform output from the port is stopped and the port is placed in floating state.

This bit can be set to "0" by instructions, by inputting a falling edge to the $\overline{INT_0}$ pin, or reset.

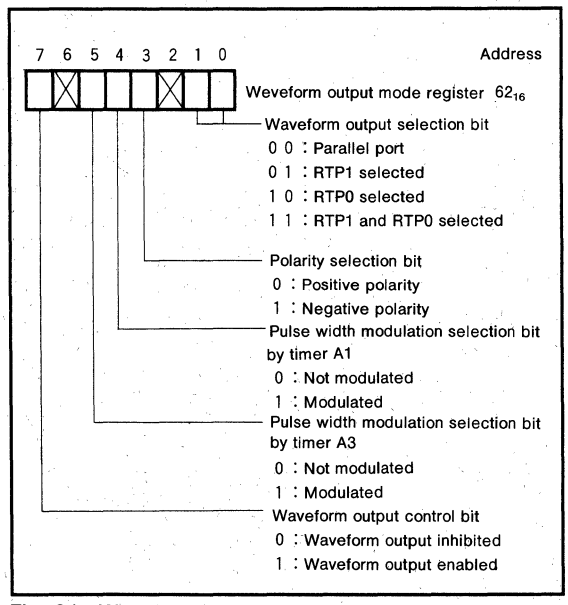


Fig. 34 Waveform output mode register bit configuration

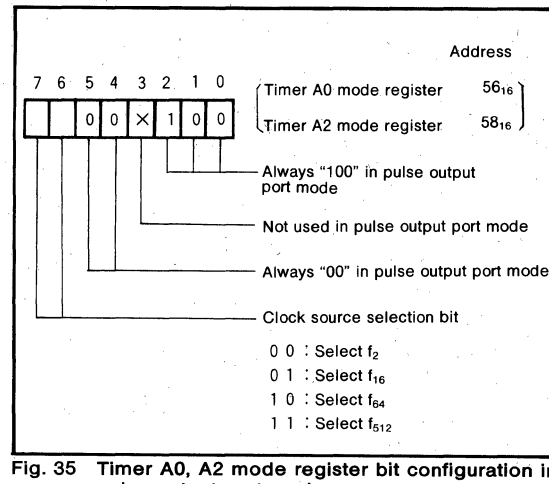


Fig. 35 Timer A0, A2 mode register bit configuration in pulse output port mode

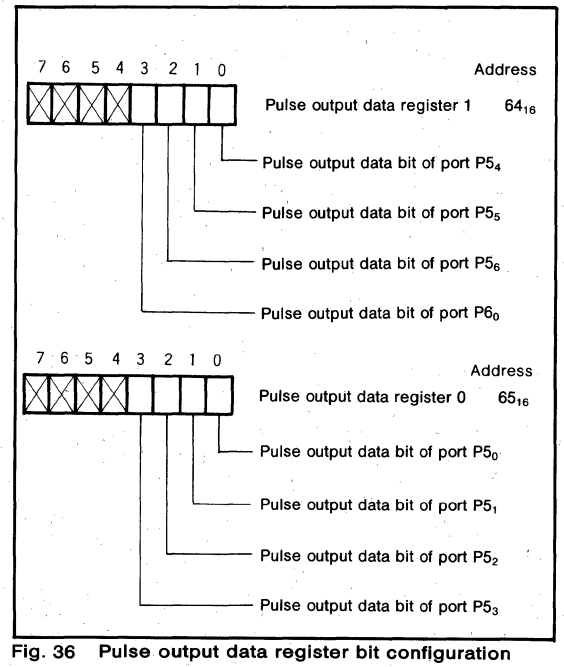


Fig. 36 Pulse output data register bit configuration

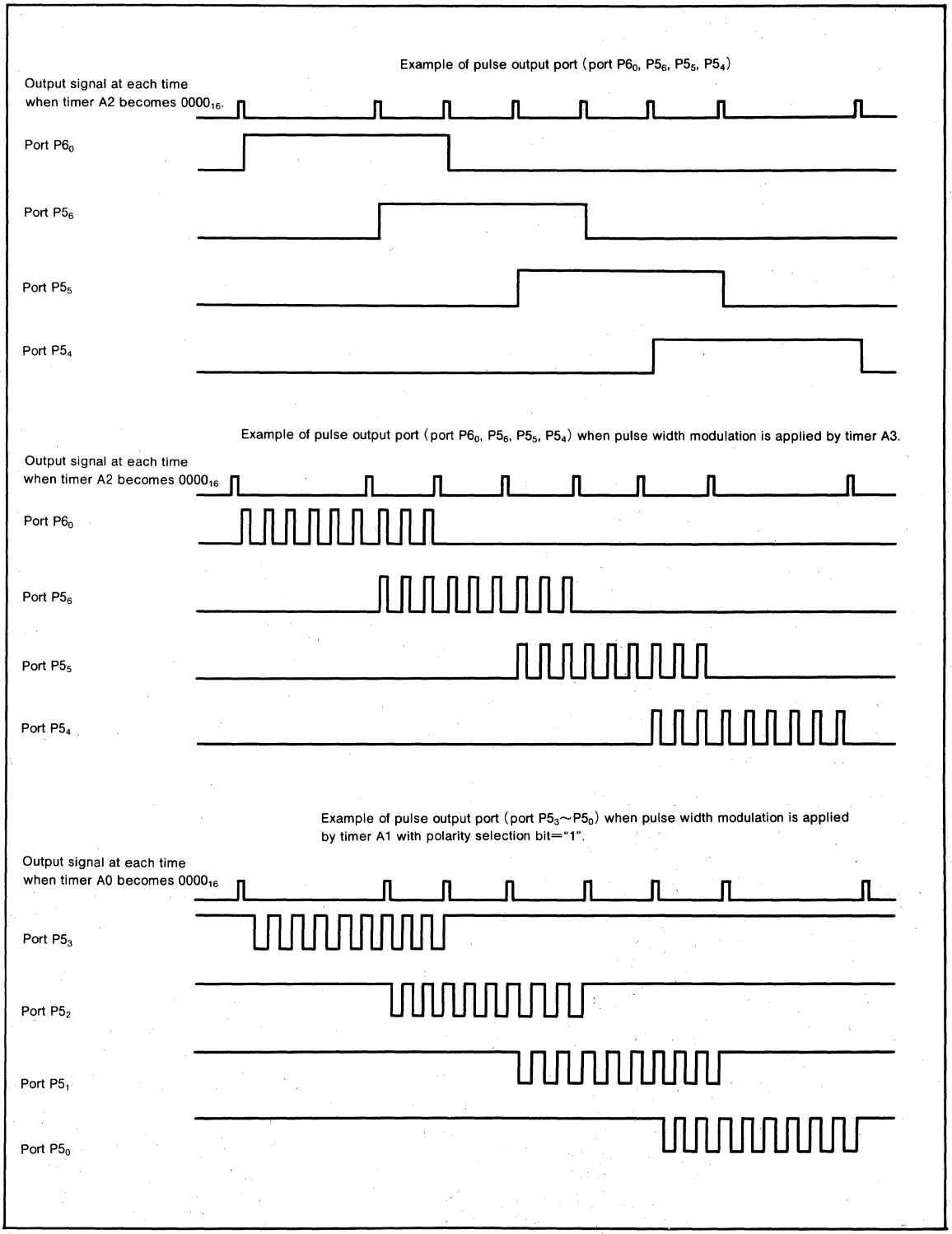


Fig. 37 Example of waveforms in pulse output port mode

SERIAL I/O PORTS

One serial I/O port is provided. Figure 38 shows a block diagram of the serial I/O port.

Bits 0, 1, and 2 of the UART0 Transmit/Receive mode register shown in Figure 39 are used to determine whether to use port P8 as parallel port, clock synchronous serial I/O port, or asynchronous (UART) serial I/O port using start and stop bits.

Figures 40 and 41 show the connections of receiver/transmitter according to the mode.

Figure 42 shows the bit configuration of the UART0 transmit/receive control register.

Each communication method is described below.

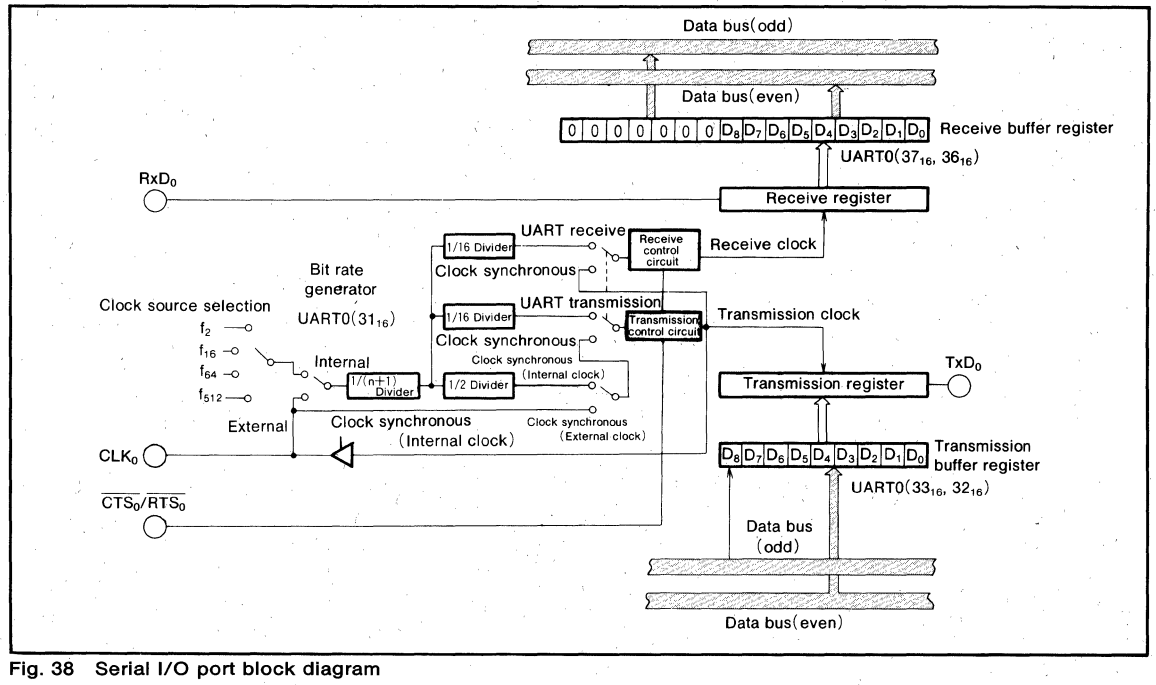


Fig. 38 Serial I/O port block diagram

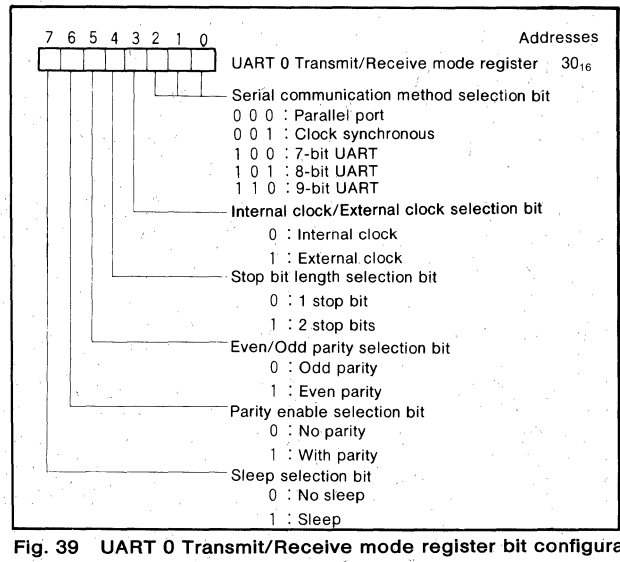


Fig. 39 UART 0 Transmit/Receive mode register bit configuration

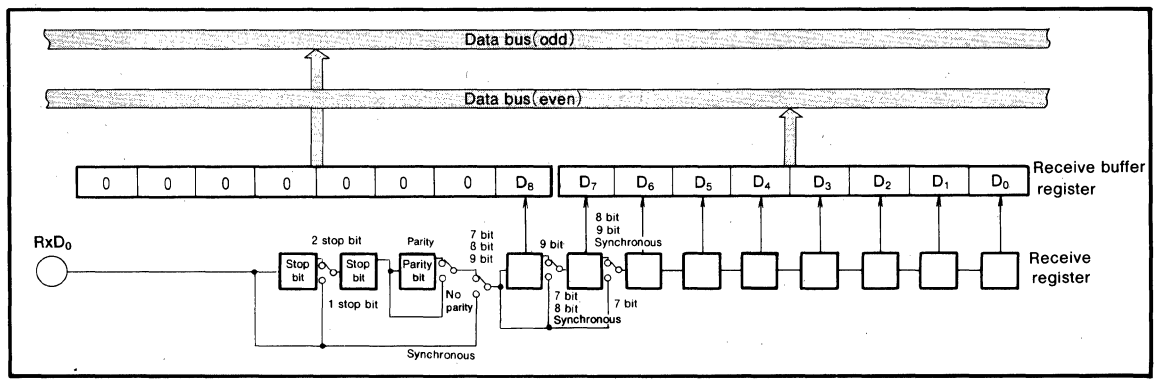


Fig. 40 Receiver block diagram

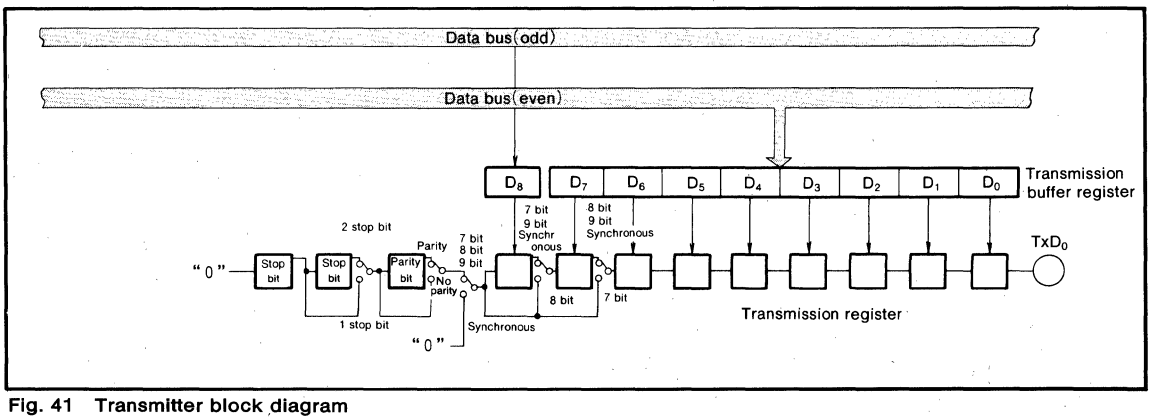


Fig. 41 Transmitter block diagram

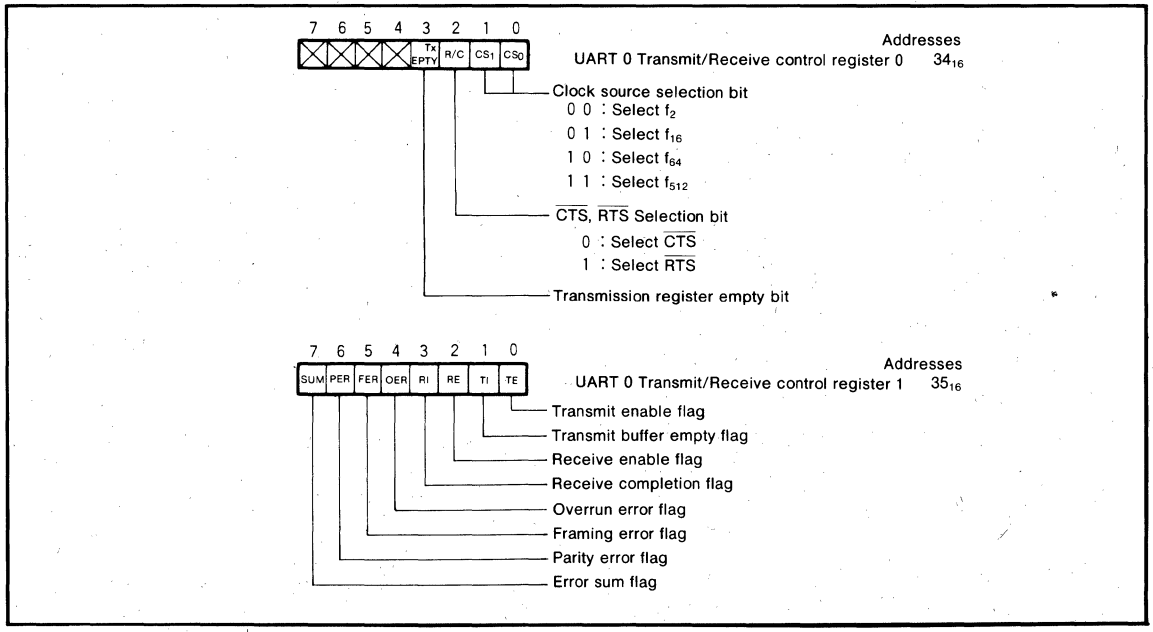


Fig. 42 UART 0 Transmit/Receive control register bit configuration

CLOCK SYNCHRONOUS SERIAL COMMUNICATION

A case where communication is performed between two clock synchronous serial I/O ports as shown in Figure 43 will be described. (The transmission side will be denoted by subscript j and the receiving side will be denoted by subscript k.)

Bit 0 of the UARTj transmit/receive mode register and UARTk transmit/receive mode register must be set to "1" and bits 1 and 2 must be "0". The length of the transmission data is fixed at 8 bits.

Bit 3 of the UARTj transmit/receive mode register of the clock sending side is cleared to "0" to select the internal clock. Bit 3 of the UARTk transmit/receive mode register of the clock receiving side is set to "1" to select the external clock. Bits 4, 5 and 6 are ignored in clock synchronous mode. Bit 7 must always be "0".

The clock source is selected by bit 0 (CS_0) and bit 1 (CS_1) of the clock sending side UARTj transmit/receive control register 0. As shown in Figure 38, the selected clock is divided by (n +1), then by 2, passed through a transmisson control circuit, and output as transmisson clock CLKj. Therefore, when the selected clock is fi,

$$\text{Bit Rate} = fi/\{(n+1)\times 2\}$$

On the clock receiving side, the CS_0 and CS_1 bits of the UARTk transmit/receive control register are ignored because an external clock is selected.

The bit 2 of the clock sending side UARTj transmit/receive control register is clear to "0" to select $\overline{CTS}j$ input. The bit 2 of the clock receiving side is set to "1" to select $\overline{RTS}k$ output. $\overline{CTS}$, and $\overline{RTS}$ signals are described later.

Transmission

Transmission is started when the bit 0 (TEj flag) of UARTj transmit/receive control register 1 is "1", bit 1 is (TIj flag) of one is "0", and $\overline{CTS}j$ input is "L". As shown in Figure 44, data is output from TxDj pin when transmission clock CLKj changes from "H" to "L". The data is output from the least significant bit.

The TIj flag indicates whether the transmission buffer register is empty or not. It is cleared to "0" when data is written in the transmission buffer register and set to "1" when the contents of the transmission buffer register is transferred to the transmission register.

When the transmission register becomes empty after the contents has been transmitted, data is transferred automatically from the transmission buffer register to the transmission register if the next transmission start condition is satisfied. If the bit 2 of UARTj transmit/receive control register 0 is "1", $\overline{CTS}j$ input is ignored and transmission start is controlled only by the TEj flag and TIj flag. Once transmission has started, the TEj flag, TIj flag, and $\overline{CTS}j$ signals are ignored until data transmission completes. Therefore, transmission is not interrupt when $\overline{CTS}j$ input is changed to "H" during transmission.

The transmission start condition indicated by TEj flag, TIj flag, and $\overline{CTS}j$ is checked while the T_{ENDj} signal shown in Figure 44 is "H". Therefore, data can be transmitted continuously if the next transmission data is written in the transmission buffer register and TIj flag is cleared to "0" before the T_{ENDj} signal goes "H".

The bit 3 (TxEPTYj flag) of UARTj transmit/receive control register 0 changes to "1" at the next cycle after the T_{ENDj} signal goes "H" and changes to "0" when transmission starts. Therefore, this flag can be used to determine whether data transmission has completed.

When the TIj flag changes from "0" to "1", the interrupt request bit in the UARTj transmission interrupt control register is set to "1".

Receive

Receive starts when the bit 2 (RE_K flag) of $UART_K$ transmit/receive control register 1 is set to "1".

The $\overline{RTS}_K$ output is "H" when the RE_K flag is "0" and goes "L" when the RE_K flag changed to "1". It goes back to "H" when receive starts. Therefore, the $\overline{RTS}_K$ output can be used to determine whether the receive register is ready to receive. It is ready when $\overline{RTS}_K$ output is "L".

The data from the RxD_K pin is retrieved and the contents of the receive register is shifted by 1 bit each time the transmission clock CLKj changes from "L" to"H". When an 8-bit data is received, the contents of the receive register is transferred to the receive buffer register and the bit 3 (RI_K flag) of $UART_K$ transmit/receive control register 1 is set to "1". In other words, the setting of the RI_K flag indicates that the receive buffer register contains the received data. At this point, $\overline{RTS}j$ output goes "L" to indicate that the next data can be received. When the RI_K flag changes from "0" to "1", the interrupt request bit in the $UART_K$ receive interrupt control register is set to "1". Bit 4 (OER_K flag) of $UART_K$ transmit/receive control register is set to "1" when the next data is transferred from the receive register to the receive buffer register while RI_K flag is "1", and indicates that the next data was transferred to the receive register before the contents of the receive buffer register was read. RI_K and OER_K flags are cleared automatically to "0" when the low-order byte of the receive buffer register is read. The OER_K flag is also cleared when the RE_K flag is cleared. Bit 5 (FER_K flag), bit 6 (PER_K flag), and bit 7 (SUM_K flag) are ignored in clock synchronous mode.

As shown in Figure 38, with clock synchronous serial communication, data cannot be received unless the transmitter is operating because the receive clock is created from the transmission clock. Therefore, the transmitter must be operating even when there is no data to be sent from $UART_K$ to UARTj.

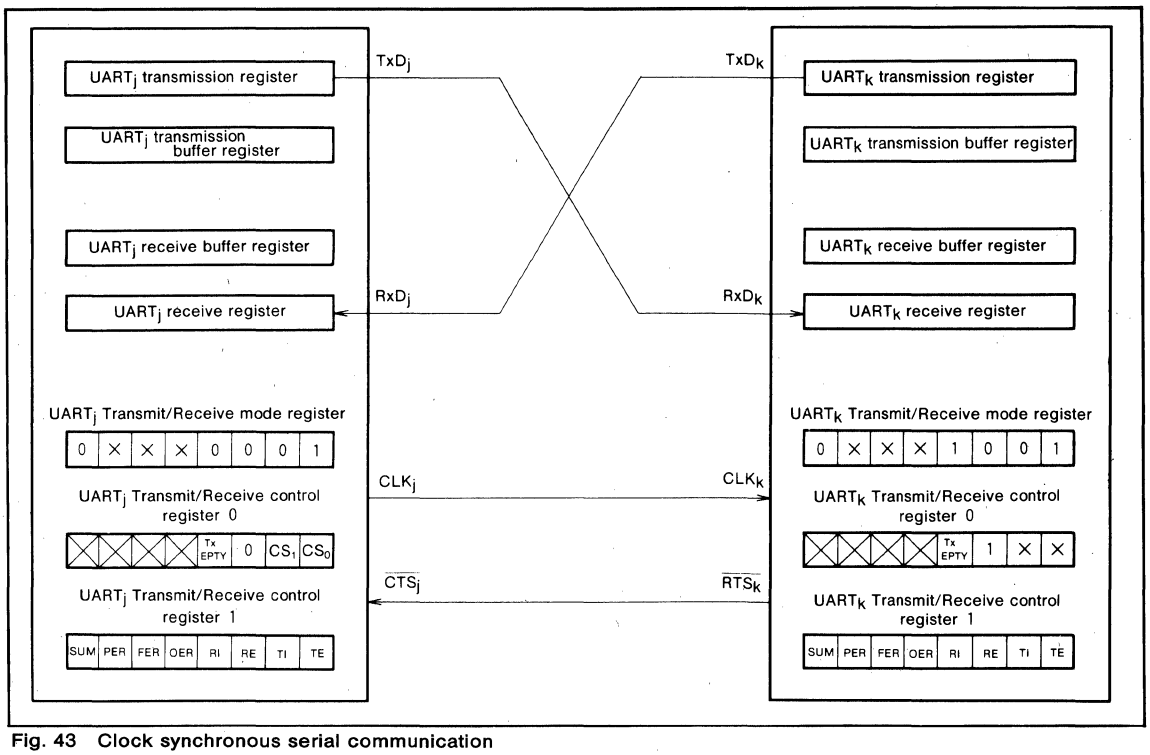


Fig. 43 Clock synchronous serial communication

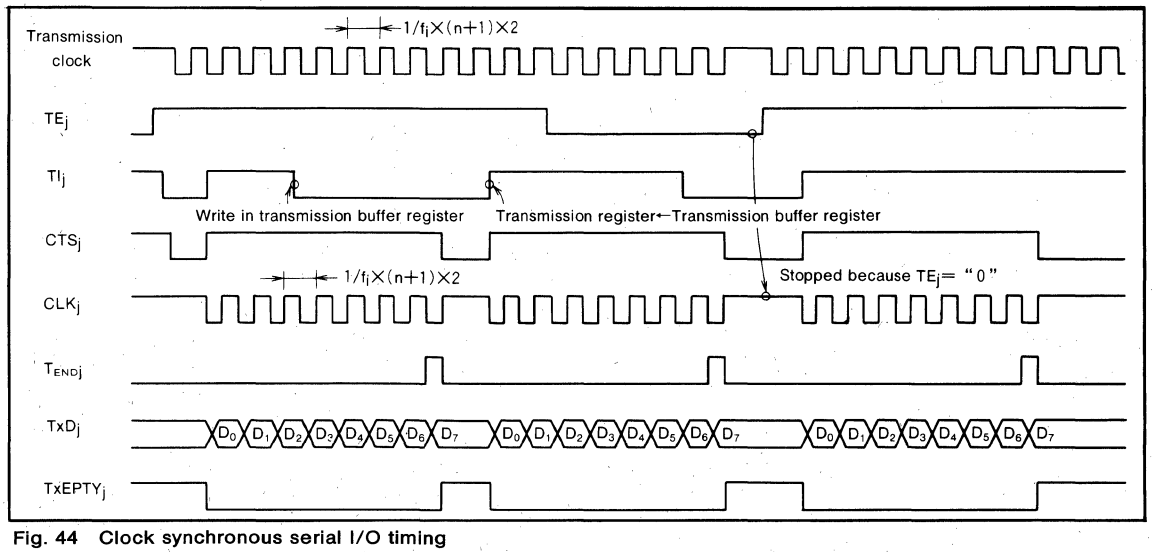


Fig. 44 Clock synchronous serial I/O timing

ASYNCHRONOUS SERIAL COMMUNICATION

Asynchronous serial communication can be performed using 7-, 8-, or 9-bit length data. The operation is the same for all data lengths. The following is the description for 8-bit asynchronous communication.

With 8-bit asynchronous communication, the bit 0 of UART0 transmit/receive mode register is "1", the bit 1 is "0", and the bit 2 is "1".

Bit 3 is used to select an internal clock or an external clock. If bit 3 is "0", an internal clock is selected and if bit 3 is "1", then external clock is selected. If an internal clock is selected, the bit 0 (CS_0) and bit 1 (CS_1) of UART0 transmit/receive control register 0 are used to select the clock source. When an internal clock is selected for asynchronous serial communication, the CLK_0 pin can be used as a normal I/O pin.

The selected internal or external clock is divided by (n+1), then by 16, and passed through a control circuit to create the UART transmission clock or UART receive clock.

Therefore, the transmission speed can be changed by changing the contents n of the bit rate generator. If the selected clock is an internal clock fi or an external clock f_{EXT},

$$\text{Bit Rate} = (f_i \text{ or } f_{EXT}) / \{(n+1) \times 16\}$$

Bit 4 is the stop bit length selection bit to select 1 stop bit or 2 stop bits.

The bit 5 is a selection bit of odd parity or even parity.

In the odd parity mode, the parity bit is adjusted so that the sum of the 1's in the data and parity bit is always odd.

In the even parity mode, the parity bit is adjusted so that the sum of the 1's in the data and parity bit is always even.

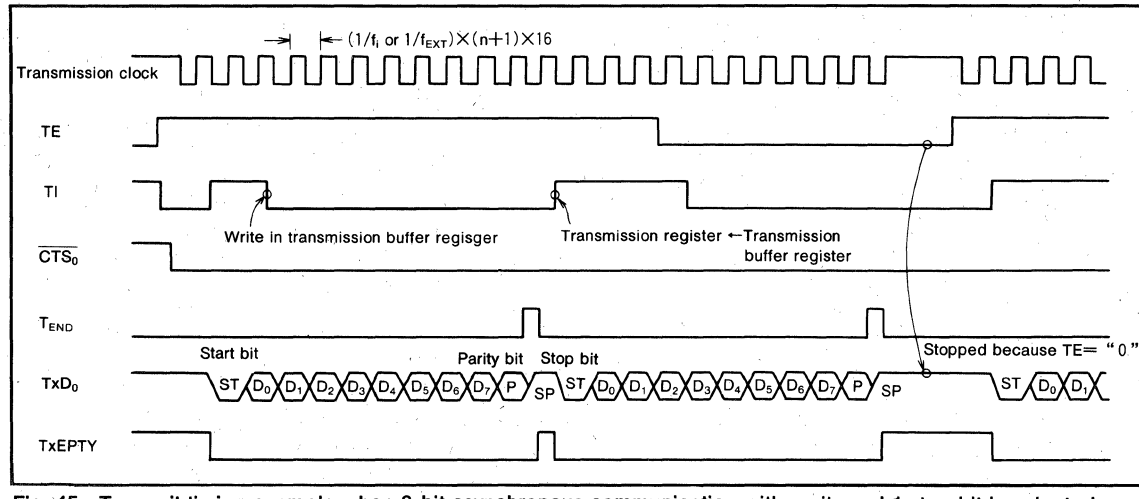


Fig. 45 Transmit timing example when 8-bit asynchronous communication with parity and 1 stop bit is selected

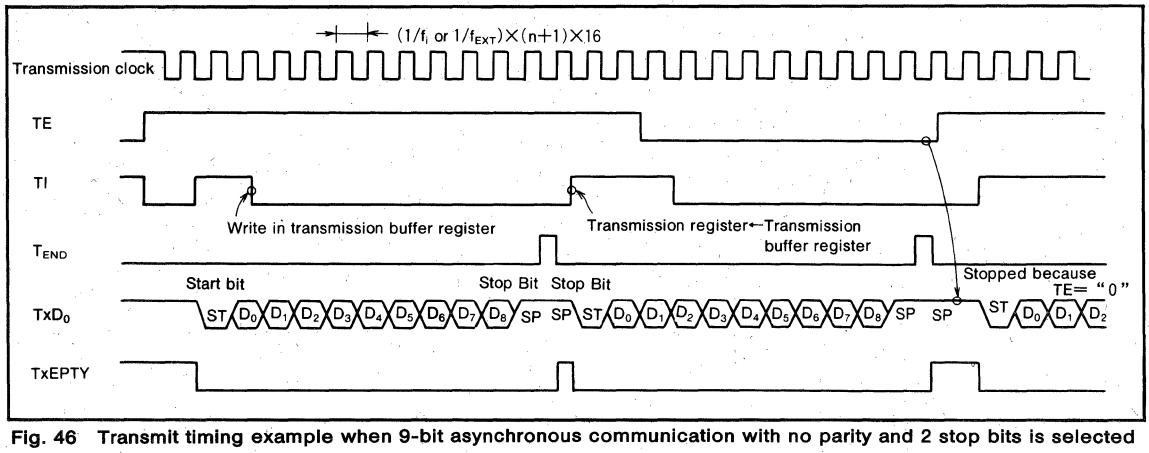


Fig. 46 Transmit timing example when 9-bit asynchronous communication with no parity and 2 stop bits is selected

Bit 6 is the parity bit selection bit which indicates whether to add parity bit or not.
Bits 4 to 6 should be set or reset according to the data format of the communicating devices.
Bit 7 is the sleep selection bit. The sleep mode is described later.
The UART0 transmit/receive control regisger 0 bit 2 is used to determine whether to use $\overline{CTS_0}$ input or $\overline{RTS_0}$ output. $\overline{CTS_0}$ input used if bit 2 is "0" and $\overline{RTS_0}$ output is used if bit 2 is "1".
If $\overline{CTS_0}$ input is selected, the user can control whether to stop or start transmission by external $\overline{CTS_0}$ input. $\overline{RTS_0}$ will be described later.

Transmission

Transmission is started when the bit 0 (TE flag) of UART0 transmit/receive control register 1 is "1", the bit 1 (TI flag) is "0", and $\overline{CTS_0}$ input is "L" if $\overline{CTS_0}$ input is selected. As shown in Figure 45 and 46, data is output from the TxD_0 pin with the stop bit and parity bit specified by the bits 4 to 6 of UART0 transmit/receive mode register bits. The data is output from the least significant bit.
The TI flag indicates whether the transmission butter is empty or not. It is cleared to "0" when data is written in the transmission buffer and set to "1" when the contents of the transmission buffer register is transferred to the transmission register.
When the transmission register becomes empty after the contents has been transmitted, data is transferred automatically form the transmission buffer register to the transmission register if the next transmission start condition is satisfied.
Once transmission has started, the TE flag, TI flag, and $\overline{CTS_0}$ signal (if $\overline{CTS_0}$ input is selected) are ignored until data transmission is completed.
Therefore, transmission does not stop until it completes even if the TE flag is cleared during transmission.
The transmission start condition indicated by TE flag, TI flag, and $\overline{CTS_0}$ is checked while the T_{END} signal shown in Figure 45 is "H". Therefore, data can be transmitted continuously if the next transmission data is written in the transmission butter register and TI flag is cleared to 0 before the T_{END} signal goes "H".
The bit 3 (TxEPTY flag) of UART0 transmit/receive control register 0 changes to "1" at the next cycle after the T_{END} signal goes "H" and changes to "0" when transmission starts. Therefore, this flag can be used to determine whether data transmision is completed.
When the TI flag changes from "0" to "1", the interrupt request bit in the UART0 transmissoin interrupt control register is set to "1".

Receive

Receive is enabled when the bit 2 (RE flag) of UART0 transmit/receive control register 1 is set. As shown in Figure 47, the frequency divider circuit at the receiving end begin to work when a start bit is arrived and the data is received.

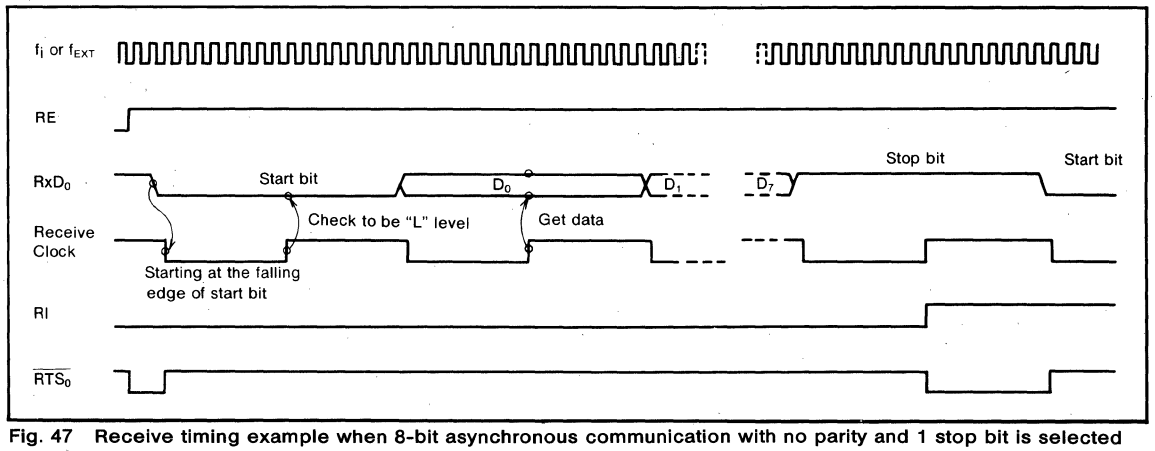


Fig. 47 Receive timing example when 8-bit asynchronous communication with no parity and 1 stop bit is selected

If $\overline{RTS_0}$ output is selected by setting the bit 2 of UART0 transmit/receive control register 0 to "1", the $\overline{RTS_0}$ output is "H" when the RE flag is "0". When the RE flag changes to "1", the $\overline{RTS_0}$ output goes "L" to indicate receive ready and returns to "H" once receive has started. In other words, $\overline{RTS_0}$ output can be used to determine externally whether the receive register is ready to receive.

The entire transmission data bits are received when the start bit passes the final bit of the receive block shown in Figure 40. At this point, the contents of the receive register is transferred to the receive buffer register and the bit 3 of UART0 transmit/receive control register 1 is set. In other words, the RI flag indicates that the receive buffer register contains data when it is set. If $\overline{RTS_0}$ output is selected, $\overline{RTS_0}$ output goes "L" to indicate that the register is ready to receive the next data.

The interrupt request bit in the UART0 receive interrupt control register is set when the RI flag changes from "0" to "1".

The bit 4 (OER flag) of UART0 transmission control register 1 is set when the next data is transferred from the receive register to the receive buffer register while the RI flag is "1". In other words when an overrun error occurs. If the OER flag is "1", it indicates that the next data has been transferred to the receive buffer register before the contents of the receive butter register has been read.

Bit 5 (FER flag) is set when the number of stop bits is less than required (framing error).

Bit 6 (PER flag) is set when a parity error occurs.

Bit 7 (SUM flag) is set when either the OER flag, FER flag, or the PER flag is set. Therefore, the SUM flag can be used to determine whether there is an error.

The setting of the RI flag, OER flag, FER flag, and the PER flag is performed while transferring the contents of the receive register to the receive buffer register. The OER, FER, PER, and SUM flags are cleared when the low order byte of the receive buffer register is read or when the RE flag is cleared.

Sleep mode

The sleep mode is used to communicate only between certain microcomputers when multiple microcomputers are connected through serial I/O.

The sleep mode is entered when the bit 7 of UART0 transmit/receive mode register is set.

The operation of the sleep mode for an 8-bit asynchronous communication is described below.

When sleep mode is selected, the contents of the receive register is not transferred to the receive buffer register if bit 7 (bit 6 if 7-bit asynchronous communication and bit 8 if 9-bit asychronous communication) of the received data is "0". Also the RI, OER, FER, PER, and the SUM flag are unchanged. Therefore, the interrupt request bit of the UART0 receive interrupt control register is also unchanged.

Normal receive operation takes place when bit 7 of the received data is "1".

The following is an example of how the sleep mode can be used.

The main microcomputer first sends data with bit 7 set to "1" and bits 0 to 6 set to the address of the subordinate microcomputer which wants to communicate with. Then all subordinate microcomputers receive the same data. Each subordinate microcomputer checks the received data, clears the sleep bit if bits 0 to 6 are its own address and sets the sleep bit if not. Next the main microcomputer sends data with bit 7 cleared. Then the microcomputer with the sleep bit cleared will receive the data, but the microcomputer with the sleep bit set will not. In this way, the main microcomputer is able to communicate with only the designated microcomputer.

WATCHDOG TIMER

The watchdog timer is used to detect unexpected execution sequence caused by software run-away.

Figure 48 shows a block diagram of the watchdog timer. The watchdog timer consists of a 12-bit binary counter.

The watchdog timer counts the clock frequency divided by 32 (f_{32}) or by 512 (f_{512}). Whether to count f_{32} or f_{512} is determined by the watchdog timer frequency selection flag shown in Figure 49. f_{512} is selected when the flag is "0" and f_{32} is selected when it is "1". The flag is cleared after reset.

FFF_{16} is set in the watchdog timer when "L" or $2V_{CC}$ is applied to the $\overline{RESET}$ pin, STP instruction is executed, data is written to the watchdog timer, or the most significant bit of the watchdog timer become "0".

After FFF_{16} is set in the watchdog timer, the contents of watchdog timer is decremented by one at every cycle of selected frequency f_{32} or f_{512}, and after 2048 counts, the most significant bit of watchdog timer become "0", and a watchdog timer interrupt request bit is set, and FFF_{16} is preset in the watchdog timer.

Normally, a program is written so that data is written in the watchdog timer before the most significant bit of the watchdog timer become "0". If this routine is not executed due to unexpected program execution, the most significant bit of the watchdog timer become eventually "0" and an interrupt is generated.

The processor can be reset by setting the bit 3 (software reset bit) of processor mode register described in Figure 10 in the interrupt section and generating a reset pulse.

The watchdog timer stops its function when the $\overline{RESET}$ pin voltage is raised to double the V_{CC} voltage.

The watchdog timer can also be used to recover from when the clock is stopped by the STP instruction. Refer to the section on clock generation circuit for more details.

The watchdog timer hold the contents during a hold state and the frequency is stopped to input.

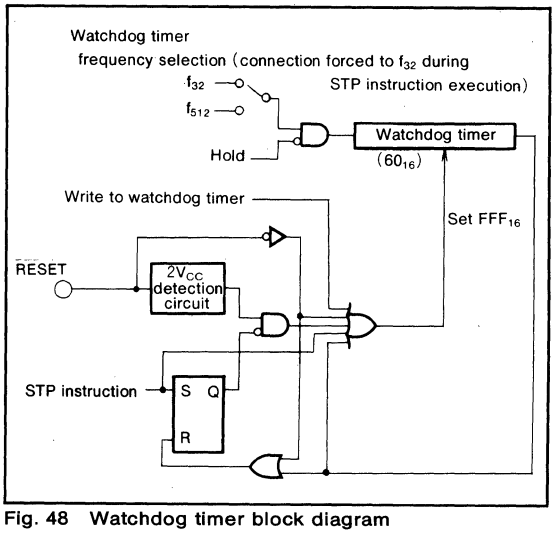


Fig. 48 Watchdog timer block diagram

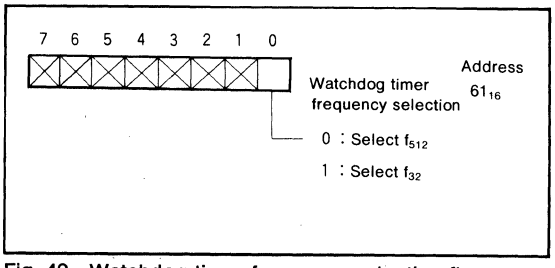


Fig. 49 Watchdog timer frequency selection flag

RESET CIRCUIT

Reset occurs when the $\overline{\text{RESET}}$ pin is returned to "H" level after holding it at "L" level when the power voltage is at 5V ±10%. Program execution starts at the address formed by setting the address pins A_{23} ~ A_{16} to 00_{16}, A_{15} ~ A_8 to the contents of address $FFFF_{16}$, and A_7 ~ A_0 to the contents of address $FFFE_{16}$.

Figure 50 shows the status of the internal registers when a reset occurs.

Figure 51 shows an example of a reset circuit. The reset input voltage must be held 0.9V or lower when the power voltage reaches 4.5V.

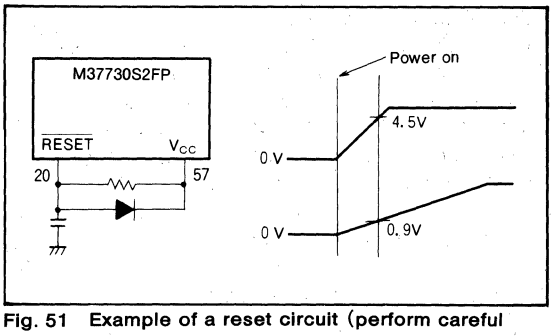


Fig. 51 Example of a reset circuit (perform careful evaluation at the system design level before using)

		Address	
(1)	Port P4 data directional register	($0C_{16}$)···	00_{16}
(2)	Port P5 data directional register	($0D_{16}$)···	00_{16}
(3)	Port P6 data directional register	(10_{16})···	00_{16}
(4)	Port P8 data directional register	(14_{16})···	00_{16}
(5)	UART 0 Transmit/Receive mode register	(30_{16})···	00_{16}
(6)	UART 0 Transmit/Receive control register 0	(34_{16})···	X X X 1 0 0 0
(7)	UART 0 Transmit/Receive control register 1	(35_{16})···	0 0 0 0 0 0 1 0
(8)	Count start flag	(40_{16})···	X X 0 0 0 0 0 0
(9)	One-shot start flag	(42_{16})···	X X 0 0 0 0 0 0
(10)	Up-down flag	(44_{16})···	00_{16}
(11)	Timer A0 mode register	(56_{16})···	00_{16}
(12)	Timer A1 mode register	(57_{16})···	00_{16}
(13)	Timer A2 mode register	(58_{16})···	00_{16}
(14)	Timer A3 mode register	(59_{16})···	00_{16}
(15)	Timer A4 mode register	($5A_{16}$)···	00_{16}
(16)	Timer B0 mode register	($5B_{16}$)···	0 0 1 X 0 0 0 0
(17)	Processor mode register	($5E_{16}$)···	X 0 0 0 0 0 1 0
(18)	Watchdog timer	(60_{16})···	FFF_{16}
(19)	Watchdog timer frequency selection flag	(61_{16})···	X X X X X X X 0
(20)	Waveform output mode register	(62_{16})···	0 X 0 0 0 X 0 0
(21)	UART 0 transmission interrupt control register	(71_{16})···	X X X 0 0 0 0 0
(22)	UART 0 receive interrupt control register	(72_{16})···	X X X 0 0 0 0 0

		Address	
(23)	Timer A0 interrupt control register	(75_{16})···	X X X X 0 0 0 0
(24)	Timer A1 interrupt control register	(76_{16})···	X X X X 0 0 0 0
(25)	Timer A2 interrupt control register	(77_{16})···	X X X X 0 0 0 0
(26)	Timer A3 interrupt control register	(78_{16})···	X X X X 0 0 0 0
(27)	Timer A4 interrupt control register	(79_{16})···	X X X X 0 0 0 0
(28)	Timer B0 interrupt control register	($7A_{16}$)···	X X X X 0 0 0 0
(29)	$\overline{\text{INT}}_0$ interrupt control register	($7D_{16}$)···	X X 0 0 0 0 0 0
(30)	$\overline{\text{INT}}_1$ interrupt control register	($7E_{16}$)···	X X 0 0 0 0 0 0
(31)	$\overline{\text{INT}}_2$ interrupt control register	($7F_{16}$)···	X X 0 0 0 0 0 0
(32)	Processor status register PS		0 0 0 ? ? 0 0 0 1 ? ?
(33)	Program bank register PG		00_{16}
(34)	Program counter PC_H		Content of $FFFF_{16}$
(35)	Program counter PC_L		Content of $FFFE_{16}$
(36)	Direct page register DPR		0000_{16}
(37)	Data bank register DT		00_{16}

Contents of other registers and RAM are not initialized and should be initialized by software.

Fig. 50 Microcomputer internal status during reset

INPUT/OUTPUT PINS

Ports P4, P5, P6, P8 all have a data direction register and each bit can be programmed for input or output. A pin becomes an output pin when the corresponding data direction register is set and an input pin when it is cleared.

When pin programmed for output, the data is written to the port latch and it is output to the output pin. When a pin is programmed for output, the contents of the port latch is read instead of the value of the pin. Therefore, a previously output value can be read correctly even when the output "L" voltage is raised due to reasons such as directly driving an LED.

A pin programmed for input is floating and the value input to the pin can be read. When a pin is programmed for input, the data is written only in the port latch and the pin stays floating.

If an input/output pin is not used as an output port, clear the bit of the corresponding data direction register so that the pin become input mode.

Figure 52 shows a block diagram of ports P4, P5, P6, P8 and the $\overline{E}$ pin output.

In evaluation chip mode, port P4 is also used as control signal pins.

Refer to the section on processor modes for more details.

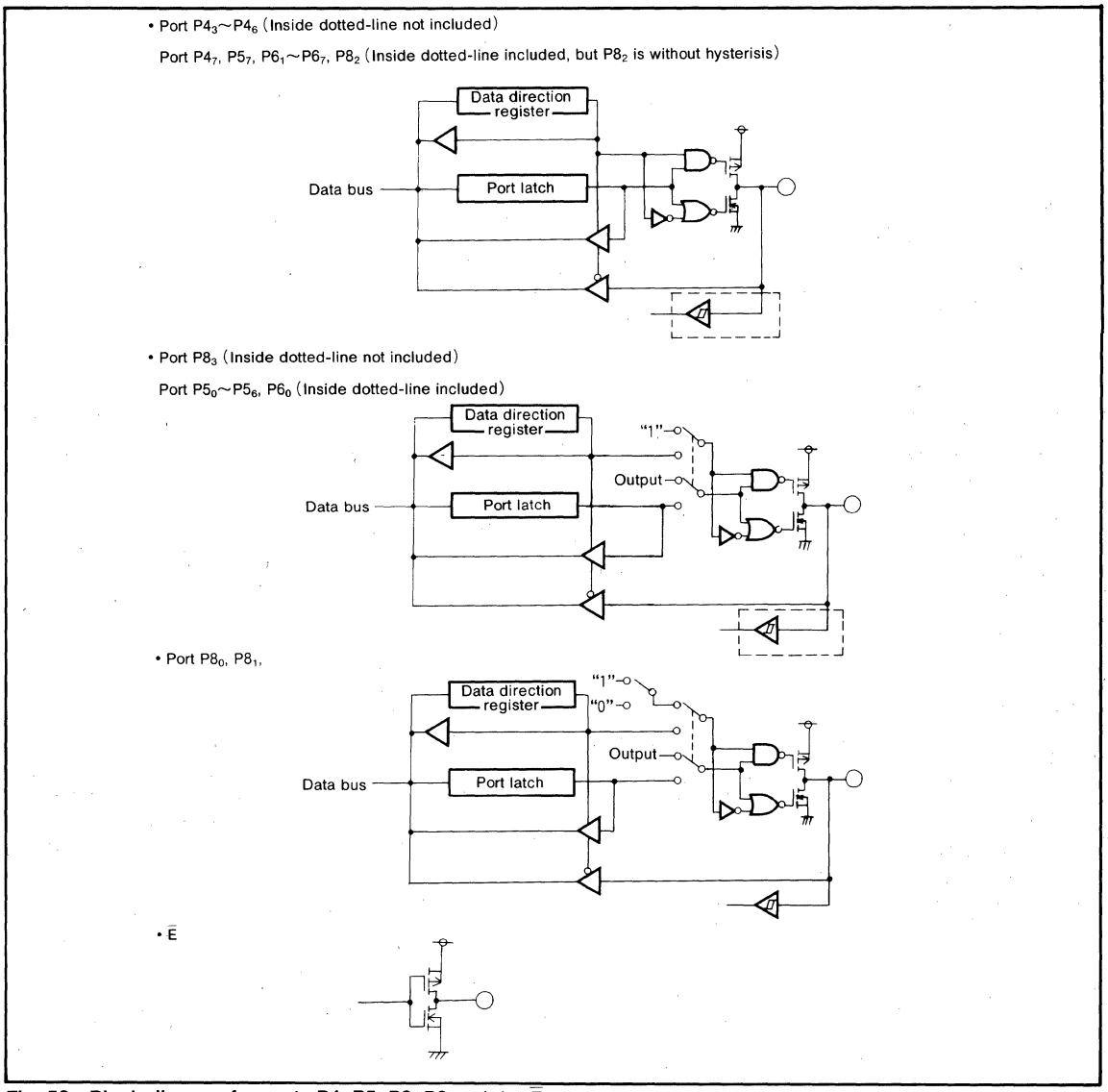


Fig. 52 Block diagram for ports P4, P5, P6, P8 and the $\overline{E}$ pin output

PROCESSOR MODE

The bits 0 of processor mode register as shown in Figure 53 is used to select either, microprocessor mode, or evaluation chip mode.

Figure 54 shows the functions of A_0 to A_7 pins, A_8/D_8 to A_{23}/D_7 pins, and port P4 in each mode.

The external memory area changes when the mode changes.

Figure 55 shows the memory map for each mode.

The accessing of the external memory is affected by the BYTE pin and the bit 2 (wait bit) of processor mode register . These will be described next.

•BYTE pin

When accessing the external memory, the level of the BYTE pin is used to determine whether to use the data bus as 8-bit width or 16-bit width.

The data bus width is 8 bits when the level of the BYTE pin is "H" and A_{16}/D_0 to A_{23}/D_7 become the data I/O pin.

The data bus width is 16 bits when the level of the BYTE pin is "L" and A_{16}/D_0 to A_{23}/D_7 pins and A_8/D_8 to A_{15}/D_{15} pins become the data I/O pins.

When accessing the internal memory, the data bus width is always 16 bits regardless of the BYTE pin level.

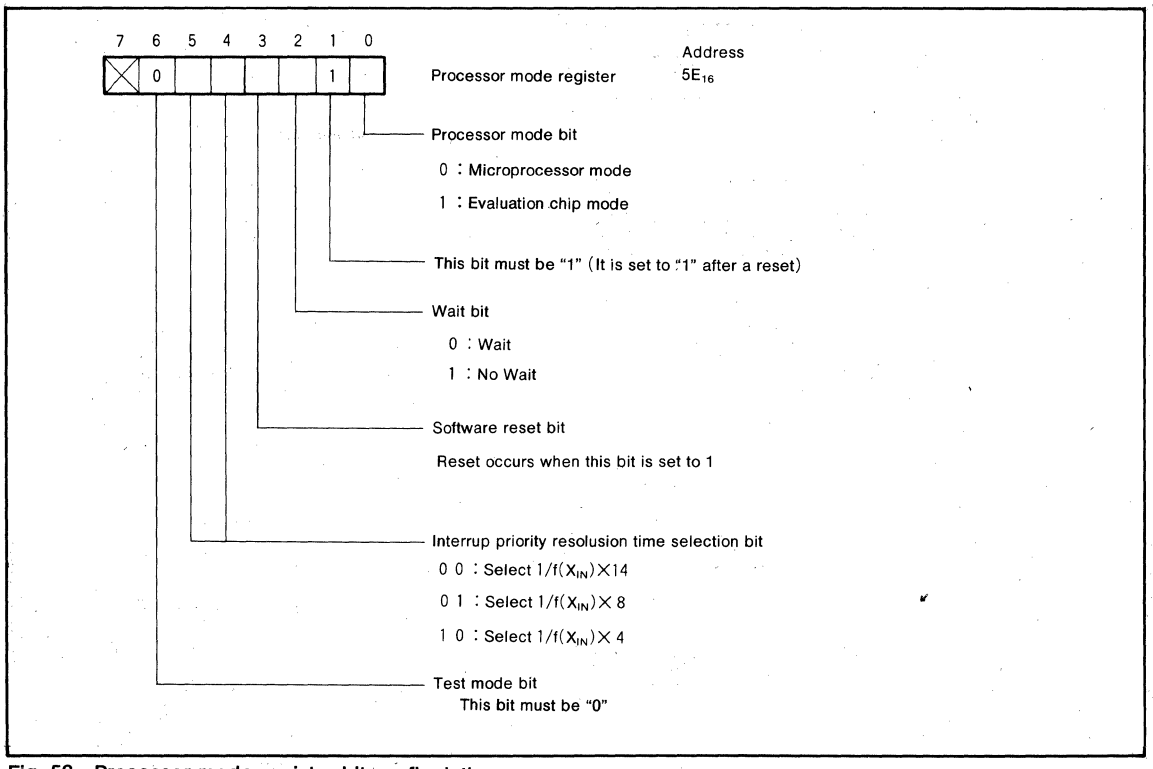


Fig. 53 Processor mode register bit configulation

Port \ Mode		CM_1 = 1, CM_0 = 0: Microprocessor Mode	CM_1 = 1, CM_0 = 1: Evaluation Chip Mode
A_0~A_7		$\overline{E}$, A_0 ~ A_7: Address A_7~A_0	Same as left
A_8/D_8 ~ A_{15}/D_{15}	BYTE = "L"	$\overline{E}$, A_8/D_8 ~ A_{15}/D_{15}: A_{15}~A_8 Address, Data (odd)	Same as left
	BYTE = "H"	$\overline{E}$, A_8/D_8 ~ A_{15}/D_{15}: Address A_{15}~A_8	$\overline{E}$, A_8/D_8 ~ A_{15}/D_{15}: A_{15}~A_8 Address, Data (odd) Port P4, P5 and their direction registers are treated as 16-bit wide bus.
A_{16}/D_0 ~ A_{23}/D_7	BYTE = "L"	$\overline{E}$, A_{16}/D_0 ~ A_{23}/D_7: A_{23}~A_{16} Address, Data (even)	Same as left
	BYTE = "H"	$\overline{E}$, A_{16}/D_0 ~ A_{23}/D_7: A_{23}~A_{16} Address, Data (even, odd)	$\overline{E}$, A_{16}/D_0 ~ A_{23}/D_7: A_{23}~A_{16} Address, Data (even, odd) Same as A_8/D_8 to A_{15}/D_{15}
Port P4		$\overline{E}$, $P4_7$ ~ $P4_3$: I/O Port	$\overline{E}$, $P4_7$: $\overline{DBC}$, $P4_6$: VPA, $P4_5$: VDA, $P4_4$: QCL, $P4_3$: MX

Fig. 54 Processor mode and A_0 to A_7 pins, A_8/D_8 to A_{23}/D_7 pins and port P4 functions

●Wait bit

As shown in Figure 56, when the external memory area is accessed with the processor mode register bit 2 (wait bit) cleared to "0", the "L" width of $\overline{E}$ signal becomes twice compared with no wait (the wait bit is "1"). The wait bit is cleared during reset.

The accessing of internal memory area is performed in no wait mode regardless of the wait bit.

The processor modes are described below.

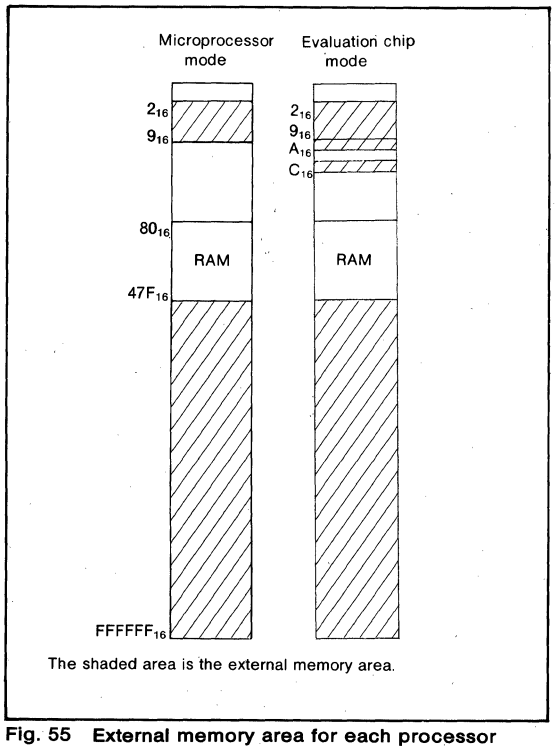


Fig. 55 External memory area for each processor mode

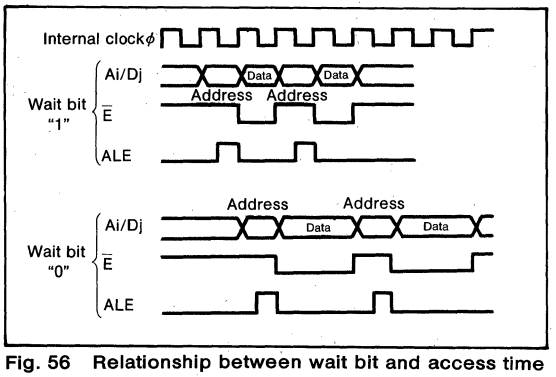


Fig. 56 Relationship between wait bit and access time

(1) Microprocessor mode 〔10〕

Microprocessor mode is entered by connecting the CNV_{SS} pin to V_{CC} and starting from reset.

A_8/D_8 to A_{15}/D_{15} pins have two functions depending on the level of the BYTE pin.

When the BYTE pin level is "L", A_8/D_8 to A_{15}/D_{15} pins function as an address output pin while $\overline{E}$ is "H" and as an odd address data I/O pin while $\overline{E}$ is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

When the BYTE pin level "H", A_8/D_8 to A_{15}/D_{15} pins function as an address output pin.

A_{16}/D_0 to A_{23}/D_7 pins have two functions depending on the level of the BYTE pin.

When the BYTE pin level is "L", $A_{16}/D_0 \sim A_{23}/D_7$ pins function as an address output pin while $\overline{E}$ is "H" and as an even address data I/O pin while $\overline{E}$ is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

When the BYTE pin level is "H", $A_{16}/D_0 \sim A_{23}/D_7$ pins functions as an address output pin while $\overline{E}$ is "H" and as an even and odd address data I/O pin while $\overline{E}$ is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

R/$\overline{W}$ is a read/write signal which indicates a read when it is "H" and a write when it is "L".

$\overline{BHE}$ is a byte high enable signal which indicates that an odd address is accessed when it is "L".

Therefore, two bytes at even and odd addresses are accessed simultaneously if address A_0 is "L" and $\overline{BHE}$ is "L".

ALE is an address latch enable signal used to latch the address signal from a multiplexed signal of address and data. The latch is transparent while ALE is "H" to let the address signal pass through and held while ALE is "L".

$\overline{HLDA}$ is a hold acknowledge signal and is used to notify externally when the microcomputer receives $\overline{HOLD}$ input and enters into hold state.

$\overline{HOLD}$ is a hold request signal. It is an input signal used to put the microcomputer in hold state. $\overline{HOLD}$ input is accepted when the internal clock ϕ falls from "H" level to "L" level while the bus is not used. A_0 to A_7 pins, A_8/D_8 to A_{23}/D_7 pins, R/$\overline{W}$ pin and $\overline{BHE}$ pin are floating while the microcomputer stays in hold state. These ports are floating after one cycle of the internal clock ϕ later than $\overline{HLDA}$ signal changes to "L" level. At the removing of hold state, these ports are removed from floating state after one cycle of ϕ later than $\overline{HLDA}$ signal changes to "H" level.

$\overline{RDY}$ is a ready signal. If this signal goes "L", the internal clock ϕ stops at "L". ϕ_1 output from clock ϕ_1 output pin doesn't stop. $\overline{RDY}$ is used when slow external memory is attached.

(2) Evaluation chip mode [11]

Evaluation chip mode is entered by applying voltage twice the V_{CC} voltage to the CNV_{SS} pin. This mode is normally used for evaluation tools.

A_8/D_8 to A_{15}/D_{15} functions as an address output pin while $\overline{E}$ is "H" and as data I/O pin of odd addresses while $\overline{E}$ is "L" regardless of the BYTE pin level. However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

A_{16}/D_0 to A_{23}/D_7 function as an address output pin while $\overline{E}$ is "H" and as data I/O pin of even addresses while $\overline{E}$ is "L" when the BYTE pin level is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

When the BYTE pin level is "H", A_{16}/D_0 to A_{23}/D_7 functions as an address output pin while $\overline{E}$ is "H" and as data I/O pin of even and odd addresses while $\overline{E}$ is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

Port P4 and its data direction register which are located at address $0A_{16}$ and $0C_{16}$ are treated differently in evaluation chip mode. When these addresses are accessed, the data bus width is treated as 16 bits regardless of the BYTE pin level, and the access cycle is treated as internal memory regardless of the wait bit.

Ports $P4_3$ to $P4_6$ become MX, QCL, VDA, and VPA output pins respectively. Port $P4_7$ becomes the $\overline{DBC}$ input pin.

The MX signal normally contains the contents of flag m, but the contents of flag x is output if the CPU is using flag x.

QCL is the queue buffer clear signal. It becomes "H" when the instruction queue buffer is cleared, for example, when a jump instruction is executed.

VDA is the valid data address signal. It becomes "H" while the CPU is reading data from data buffer or writing data to data buffer. It also becomes "H" when the first byte of the instruction (operation code) is read from the instruction queue buffer.

VPA is the valid program address signal. It becomes "H" while the CPU is reading an instruction code from the instruction queue buffer.

$\overline{DBC}$ is the debug control signal and is used for debugging.

Table 5 shows the relationship between the CNV_{SS} pin input levels and processor modes.

Table 5. Relationship between the CNV_{SS} pin input levels and processor modes

CNV_{SS}	Mode	Description
V_{CC}	• Microprocessor • Evaluation chip	Microprocessor mode upon starting after reset. Evaluation chip mode can be selected by changing the processor mode bit by software.
$2 \cdot V_{CC}$	• Evaluation chip	• Evaluation chip mode only.

CLOCK GENERATING CIRCUIT

Figure 57 shows a block diagram of the clock generator.

When an STP instruction is executed, the internal clock ϕ stops oscillating at "L" level. At the same time, FFF_{16} is written to watchdog timer and the watchdog timer input connection is forced to f_{32}. This connection is broken and connected to the input determined by the watchdog timer frequency selection flag when the most significant bit of the watchdog timer is cleared or reset.

Oscillation resumes when an interrupt is received, but the internal clock ϕ remains at "L" level until the most significant bit of the watchdog timer is cleared. This is to avoid the unstable interval at the start of oscillation when using a ceramic resonator.

When a WIT instruction is executed, the internal clock ϕ stops at "L" level, but the oscillator does not stop. The clock is restarted when an interrupt is received. Instructions can be executed immediately because the oscillator is not stopped.

The stop or wait state is released when an interrupt is received or when reset is issued. Therefore, interrupts must be enabled before executing a STP or WIT instruction.

Figure 58 shows a circuit example using a ceramic (or quartz crystal) resonator. Use the manufacturer's recommended values for constants such as capacitance which differ for each resonator. Figure 59 shows an example of using an external clock signal.

ADDRESSING MODES

The M37730S2FP has 28 powerful addressing modes. Refer to the MELPS 7700 addressing mode description for the details of each addressing mode.

MACHINE INSTRUCTION LIST

The M37730S2FP has 103 machine instructions. Refer to the MELPS 7700 machine instruction list for details.

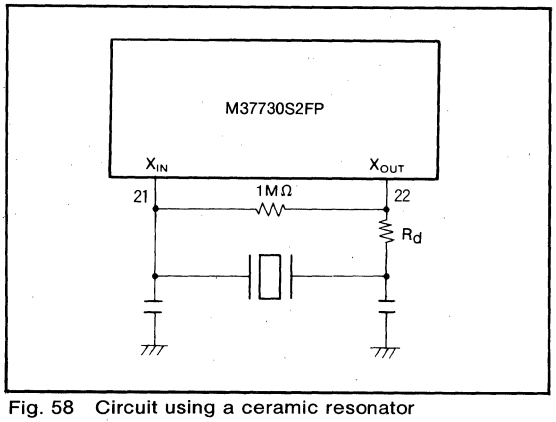


Fig. 58 Circuit using a ceramic resonator

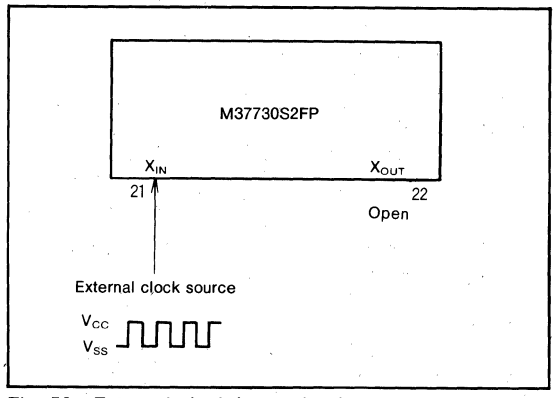


Fig. 59 External clock input circuit

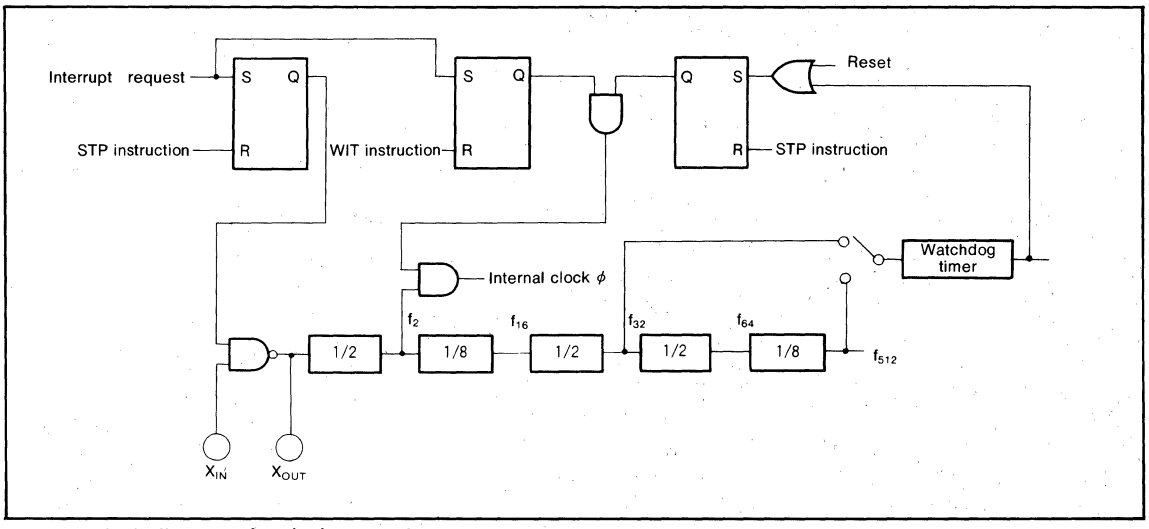


Fig. 57 Block diagram of a clock generator

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Supply voltage		−0.3∼7	V
V_I	Input voltage $\overline{RESET}$, CNV_{SS}, BYTE		−0.3∼12	V
V_I	Input voltage A_8/D_8∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P8_0$∼$P8_3$, X_{IN}, $\overline{HOLD}$, $\overline{RDY}$		−0.3∼V_{CC}+0.3	V
V_O	Output voltage A_0∼A_7, A_8/D_8∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P8_0$∼$P8_3$, X_{OUT}, $\overline{E}$, ϕ_1, $\overline{HLDA}$, ALE, $\overline{BHE}$, R/$\overline{W}$		−0.3∼V_{CC}+0.3	V
P_d	Power dissipation	T_a=25℃	300 (Note)	mW
Topr	Operating temperature		−20∼85	℃
Tstg	Storage temperature		−40∼150	℃

Note. In case of shrink plastic molded DIP, rating of power dissipation is 1000mW.

RECOMMENDED OPERATING CONDITIONS (V_{CC}=5V±10%, T_a=−20∼85℃, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V_{CC}	Supply voltage		4.5	5.0	5.5	V
V_{SS}	Supply voltage			0		V
V_{IH}	High-level input voltage $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P8_0$∼$P8_3$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE, $\overline{HOLD}$, $\overline{RDY}$		0.8V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage A_8/D_8∼A_{23}/D_7		0.5V_{CC}		V_{CC}	V
V_{IL}	Low-level input voltage $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P8_0$∼$P8_3$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE, $\overline{HOLD}$, $\overline{RDY}$		0		0.2V_{CC}	V
V_{IL}	Low-level input voltage A_8/D_8∼A_{23}/D_7		0		0.16V_{CC}	V
$I_{OH(peak)}$	High-level peak output current A_0∼A_7, A_8/D_8∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P8_0$∼$P8_3$, ϕ_1, $\overline{HLDA}$, ALE, $\overline{BHE}$, R/$\overline{W}$				−10	mA
$I_{OH(avg)}$	High-level average output current A_0∼A_7, A_8/D_8∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P8_0$∼$P8_3$, ϕ_1, $\overline{HLDA}$, ALE, $\overline{BHE}$, R/$\overline{W}$				−5	mA
$I_{OL(peak)}$	Low-level peak output current A_0∼A_7, A_8/D_8∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P8_0$∼$P8_3$, ϕ_1, $\overline{HLDA}$, ALE, $\overline{BHE}$, R/$\overline{W}$				10	mA
$I_{OL(avg)}$	Low-level average output current A_0∼A_7, A_8/D_8∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P8_0$∼$P8_3$, ϕ_1, $\overline{HLDA}$, ALE, $\overline{BHE}$, R/$\overline{W}$				5	mA
f(X_{IN})	External clock frequency input	M37730S2FP, M37730S2SP			8	MHz
		M37730S2AFP, M37730S2ASP			16	
		M37730S2BFP, M37730S2BSP			25	

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

2. The sum of $I_{OL(peak)}$ for A_0∼A_7, A_8/D_8∼A_{23}/D_7, $\overline{HLDA}$, ALE, BHE, R/$\overline{W}$, and port P8 must be 80mA or less,
the sum of $I_{OH(peak)}$ for A_0∼A_7, A_8/D_8∼A_{23}/D_7, $\overline{HLDA}$, ALE, BHE, R/$\overline{W}$, and port P8 must be 80mA or less,
the sum of $I_{OL(peak)}$ for ports P4, P5, P6, and ϕ_1 must be 80mA or less, and
the sum of $I_{OH(peak)}$ for ports P4, P5, P6, and ϕ_1 must be 80mA or less.

M37730S2FP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=8MHz, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{OH}	High-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P8_0$~$P8_3$, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OH}=−10mA		3			V
V_{OH}	High-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OH}=−400μA		4.7			V
V_{OH}	High-level output voltage ALE	I_{OH}=−10mA		3.1			V
		I_{OH}=−400μA		4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA		3.4			V
		I_{OH}=−400μA		4.8			
V_{OL}	Low-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P8_0$~$P8_3$, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OL}=10mA				2	V
V_{OL}	Low-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OL}=2mA				0.45	V
V_{OL}	Low-level output voltage ALE	I_{OL}=10mA				1.9	V
		I_{OL}=2mA				0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA				1.6	V
		I_{OL}=2mA				0.4	
$V_{T+}-V_{T-}$	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA4_{IN}$, $TB0_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, $\overline{CTS_0}$, CLK_0			0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis $\overline{RESET}$			0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X_{IN}			0.1		0.3	V
I_{IH}	High-level input current A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P8_0$~$P8_3$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE, $\overline{HOLD}$, $\overline{RDY}$	V_I=5V				5	μA
I_{IL}	Low-level input current A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P8_0$~$P8_3$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE, $\overline{HOLD}$, $\overline{RDY}$	V_I=0V				−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.		2			V
I_{CC}	Power supply current	Output only pin is open and other pins are V_{SS} during reset.	f(X_{IN})=8MHz, square waveform		6	12	mA
			T_a=25℃ when clock is stopped.			1	μA
			T_a=85℃ when clock is stopped.			20	

M37730S2AFP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=16MHz, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{OH}	High-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P8_0$~$P8_3$, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OH}=−10mA		3			V
V_{OH}	High-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OH}=−400μA		4.7			V
V_{OH}	High-level output voltage ALE	I_{OH}=−10mA		3.1			V
		I_{OH}=−400μA		4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA		3.4			V
		I_{OH}=−400μA		4.8			
V_{OL}	Low-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P8_0$~$P8_3$, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OL}=10mA				2	V
V_{OL}	Low-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OL}=2mA				0.45	V
V_{OL}	Low-level output voltage ALE	I_{OL}=10mA				1.9	V
		I_{OL}=2mA				0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA				1.6	V
		I_{OL}=2mA				0.4	
$V_{T+}-V_{T-}$	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA4_{IN}$, $TB0_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, $\overline{CTS_0}$, CLK_0			0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis $\overline{RESET}$			0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X_{IN}			0.1		0.3	V
I_{IH}	High-level input current A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P8_0$~$P8_3$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE, $\overline{HOLD}$, $\overline{RDY}$	V_I=5V				5	μA
I_{IL}	Low-level input current A_8/D_8~A_{23}/D_7 $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P8_0$~$P8_3$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE, $\overline{HOLD}$, $\overline{RDY}$	V_I=0V				−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.		2			V
I_{CC}	Power supply current	Output only pin is open and other pins are V_{SS} during reset.	f(X_{IN})=16MHz, square waveform		12	24	mA
			T_a=25℃ when clock is stopped.			1	μA
			T_a=85℃ when clock is stopped.			20	

M37730S2BFP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=25MHz, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{OH}	High-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P8_0$~$P8_3$, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OH}=−10mA		3			V
V_{OH}	High-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OH}=−400μA		4.7			V
V_{OH}	High-level output voltage ALE	I_{OH}=−10mA		3.1			V
		I_{OH}=−400μA		4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA		3.4			V
		I_{OH}=−400μA		4.8			
V_{OL}	Low-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P8_0$~$P8_3$, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OL}=10mA				2	V
V_{OL}	Low-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OL}=2mA				0.45	V
V_{OL}	Low-level output voltage ALE	I_{OL}=10mA				1.9	V
		I_{OL}=2mA				0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA				1.6	V
		I_{OL}=2mA				0.4	
$V_{T+}-V_{T-}$	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA4_{IN}$, $TB0_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, $\overline{CTS_0}$, CLK_0			0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis $\overline{RESET}$			0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X_{IN}			0.1		0.3	V
I_{IH}	High-level input current A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P8_0$~$P8_3$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE, $\overline{HOLD}$, $\overline{RDY}$	V_I=5V				5	μA
I_{IL}	Low-level input current A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P8_0$~$P8_3$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE, $\overline{HOLD}$, $\overline{RDY}$	V_I=0V				−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.		2			V
I_{CC}	Power supply current	Output only pin is open and other pins are V_{SS} during reset.	f(X_{IN})=25MHz, square waveform		19	38	mA
			T_a=25℃ when clock is stopped.			1	μA
			T_a=85℃ when clock is stopped.			20	

TIMING REQUIREMENTS (V_{CC}=5V±10%, V_{SS}=0V, T_a=25℃, unless otherwise noted)

External clock input

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
t_C	External clock input cycle time	125		62		40		ns
$t_{W(H)}$	External clock input high-level pulse width	50		25		15		ns
$t_{W(L)}$	External clock input low-level pulse width	50		25		15		ns
t_r	External clock rise time		20		10		8	ns
t_f	External clock fall time		20		10		8	ns

Microprocessor mode

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{SU(DH-E)}$	Data high-order input setup time	60		45		30		ns
$t_{SU(DL-E)}$	Data low-order input setup time	60		45		30		ns
$t_{SU(P4D-E)}$	Port P4 input setup time	200		100		60		ns
$t_{SU(P5D-E)}$	Port P5 input setup time	200		100		60		ns
$t_{SU(P6D-E)}$	Port P6 input setup time	200		100		60		ns
$t_{SU(P8D-E)}$	Port P8 input setup time	200		100		60		ns
$t_{SU(RDY-\phi_1)}$	$\overline{RDY}$ input setup time	70		60		55		ns
$t_{SU(HOLD-\phi_1)}$	$\overline{HOLD}$ input setup time	70		60		55		ns
$t_{h(E-DH)}$	Data high-order input hold time	0		0		0		ns
$t_{h(E-DL)}$	Data low-order input hold time	0		0		0		ns
$t_{h(E-P4D)}$	Port P4 input hold time	0		0		0		ns
$t_{h(E-P5D)}$	Port P5 input hold time	0		0		0		ns
$t_{h(E-P6D)}$	Port P6 input hold time	0		0		0		ns
$t_{h(E-P8D)}$	Port P8 input hold time	0		0		0		ns
$t_{h(\phi_1-RDY)}$	$\overline{RDY}$ input hold time	0		0		0		ns
$t_{h(\phi_1-HOLD)}$	$\overline{HOLD}$ input hold time	0		0		0		ns

Timer A input (Count input in event counter mode)

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	250		125		80		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	125		62		40		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	125		62		40		ns

Timer A input (Gating input in timer mode)

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	1000		500		320		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	500		250		160		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	500		250		160		ns

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

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	500		250		160		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	250		125		80		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	250		125		80		ns

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

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	250		125		80		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	250		125		80		ns

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

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(UP)}$	TAi_{OUT} input cycle time	5000		2500		2000		ns
$t_{W(UPH)}$	TAi_{OUT} input high-level pulse width	2500		1250		1000		ns
$t_{W(UPL)}$	TAi_{OUT} input low-level pulse width	2500		1250		1000		ns
$t_{SU(UP-T_{IN})}$	TAi_{OUT} input setup time	1000		500		400		ns
$t_{h(T_{IN}-UP)}$	TAi_{OUT} input hold time	1000		500		400		ns

Timer B input (Count input in event counter mode)

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	$TB0_{IN}$ input cycle time (one edge count)	250		125		80		ns
$t_{W(TBH)}$	$TB0_{IN}$ input high-level pulse width (one edge count)	125		62		40		ns
$t_{W(TBL)}$	$TB0_{IN}$ input low-level pulse width (one edge count)	125		62		40		ns
$t_{C(TB)}$	$TB0_{IN}$ input cycle time (both edges count)	500		250		160		ns
$t_{W(TBH)}$	$TB0_{IN}$ input high-level pulse width (both edges count)	250		125		80		ns
$t_{W(TBL)}$	$TB0_{IN}$ input low-level pulse width (both edges count)	250		125		80		ns

Timer B input (Pulse period measurement mode)

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	$TB0_{IN}$ input cycle time	1000		500		320		ns
$t_{W(TBH)}$	$TB0_{IN}$ input high-level pulse width	500		250		160		ns
$t_{W(TBL)}$	$TB0_{IN}$ input low-level pulse width	500		250		160		ns

Timer B input (Pulse width measurement mode)

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	$TB0_{IN}$ input cycle time	1000		500		320		ns
$t_{W(TBH)}$	$TB0_{IN}$ input high-level pulse width	500		250		160		ns
$t_{W(TBL)}$	$TB0_{IN}$ input low-level pulse width	500		250		160		ns

Serial I/O

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(CK)}$	CLK_0 input cycle time	500		250		200		ns
$t_{W(CKH)}$	CLK_0 input high-level pulse width	250		125		100		ns
$t_{W(CKL)}$	CLK_0 input low-level pulse width	250		125		100		ns
$t_{d(C-Q)}$	TxD_0 output delay time		150		90		80	ns
$t_{h(C-Q)}$	TxD_0 hold time	30		30		30		ns
$t_{su(D-C)}$	RxD_0 input setup time	60		30		20		ns
$t_{h(C-D)}$	RxD_0 input hold time	90		90		90		ns

External interrupt $\overline{INT_i}$ input

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{W(INH)}$	$\overline{INT_i}$ input high-level pulse width	250		250		250		ns
$t_{W(INL)}$	$\overline{INT_i}$ input low-level pulse width	250		250		250		ns

SWITCHING CHARACTERISTICS (V_{CC}=5V±10%, V_{SS}=0V, T_a=25℃, unless otherwise noted)

Microprocessor mode (when wait bit = "1")

Symbol	Parameter	Test conditions	Limits						Unit
			8 MHz		16MHz		25MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(AL-E)}$	Address low-order output delay time	Fig. 60	100		30		12		ns
$t_{d(E-DHQ)}$	Data high-order output delay time (BYTE="L")			110		70		45	ns
$t_{PXZ(E-DHZ)}$	Floating start delay time (BYTE="L")			5		5		5	ns
$t_{d(AM-E)}$	Address middle-order output delay time		100		30		12		ns
$t_{d(AM-ALE)}$	Address middle-order output delay time		80		24		5		ns
$t_{d(E-DLQ)}$	Data low-order output delay time			110		70		45	ns
$t_{PXZ(E-DLZ)}$	Floating start delay time			5		5		5	ns
$t_{d(AH-E)}$	Address high-order output delay time		100		30		12		ns
$t_{d(AH-ALE)}$	Address high-order output delay time		80		24		5		ns
$t_{d(\phi_1-HLDA)}$	$\overline{HLDA}$ output delay time			100		50		50	ns
$t_{d(ALE-E)}$	ALE output delay time		4		4		4		ns
$t_{W(ALE)}$	ALE pulse width		90		35		22		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time		100		30		20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time		100		30		20		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	30	0	20	0	18	ns
$t_{h(E-AL)}$	Address low-order hold time		50		25		18		ns
$t_{h(ALE-AM)}$	Address middle-order hold time (BYTE="L")		9		9		9		ns
$t_{h(E-DHQ)}$	Data high-order hold time (BYTE="L")		50		25		18		ns
$t_{PZX(E-DHZ)}$	Floating release delay time (BYTE="L")		50		25		18		ns
$t_{h(E-AM)}$	Address middle-order hold time (BYTE="H")		50		25		18		ns
$t_{h(ALE-AH)}$	Address high-order hold time		9		9		9		ns
$t_{h(E-DLQ)}$	Data low-order hold time		50		25		18		ns
$t_{PZX(E-DLZ)}$	Floating release delay time		50		25		18		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		18		18		18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		18		18		18		ns
$t_{d(E-P4Q)}$	Port P4 data output delay time			200		100		80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			200		100		80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			200		100		80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			200		100		80	ns
$t_{W(EL)}$	$\overline{E}$ pulse width		220		95		50		ns

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

Symbol	Parameter	Test conditions	Limits						Unit
			8 MHz		16MHz		25MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(AL-E)}$	Address low-order output delay time	Fig.60	100		30		12		ns
$t_{d(E-DHQ)}$	Data high-order output delay time (BYTE="L")			110		70		45	ns
$t_{PXZ(E-DHZ)}$	Floating start delay time (BYTE="L")			5		5		5	ns
$t_{d(AM-E)}$	Address middle-order output delay time		100		30		12		ns
$t_{d(AM-ALE)}$	Address middle-order output delay time		80		24		5		ns
$t_{d(E-DLQ)}$	Data low-order output delay time			110		70		45	ns
$t_{PXZ(E-DLZ)}$	Floating start delay time			5		5		5	ns
$t_{d(AH-E)}$	Address high-order output delay time		100		30		12		ns
$t_{d(AH-ALE)}$	Address high-order output delay time		80		24		5		ns
$t_{d(\phi_1-HLDA)}$	$\overline{HLDA}$ output delay time			100		50		50	ns
$t_{d(ALE-E)}$	ALE output delay time		4		4		4		ns
$t_{W(ALE)}$	ALE pulse width		90		35		22		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time		100		30		20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time		100		30		20		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	30	0	20	0	18	ns
$t_{h(E-AL)}$	Address low-order hold time		50		25		18		ns
$t_{h(ALE-AM)}$	Address middle-order hold time (BYTE="L")		9		9		9		ns
$t_{h(E-DHQ)}$	Data high-order hold time (BYTE="L")		50		25		18		ns
$t_{PZX(E-DHZ)}$	Floating release delay time (BYTE="L")		50		25		18		ns
$t_{h(E-AM)}$	Address middle-order hold time (BYTE="H")		50		25		18		ns
$t_{h(ALE-AH)}$	Address high-order hold time		9		9		9		ns
$t_{h(E-DLQ)}$	Data low-order hold time		50		25		18		ns
$t_{PZX(E-DLZ)}$	Floating release delay time		50		25		18		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		18		18		18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		18		18		18		ns
$t_{d(E-P4Q)}$	Port P4 data output delay time			200		100		80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			200		100		80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			200		100		80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			200		100		80	ns
$t_{W(EL)}$	$\overline{E}$ pulse width		470		220		130		ns

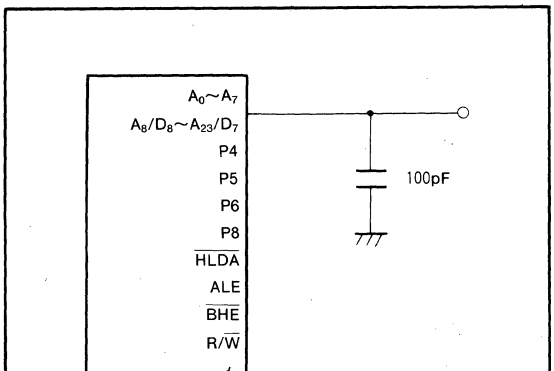
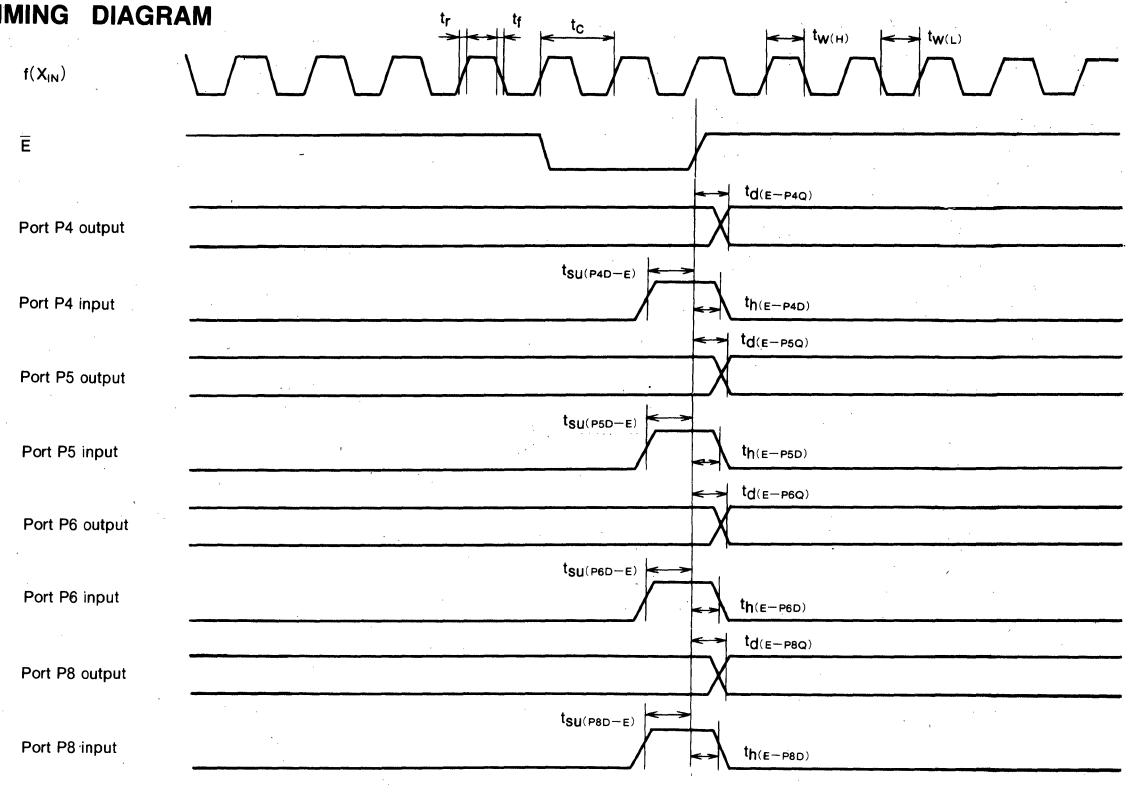
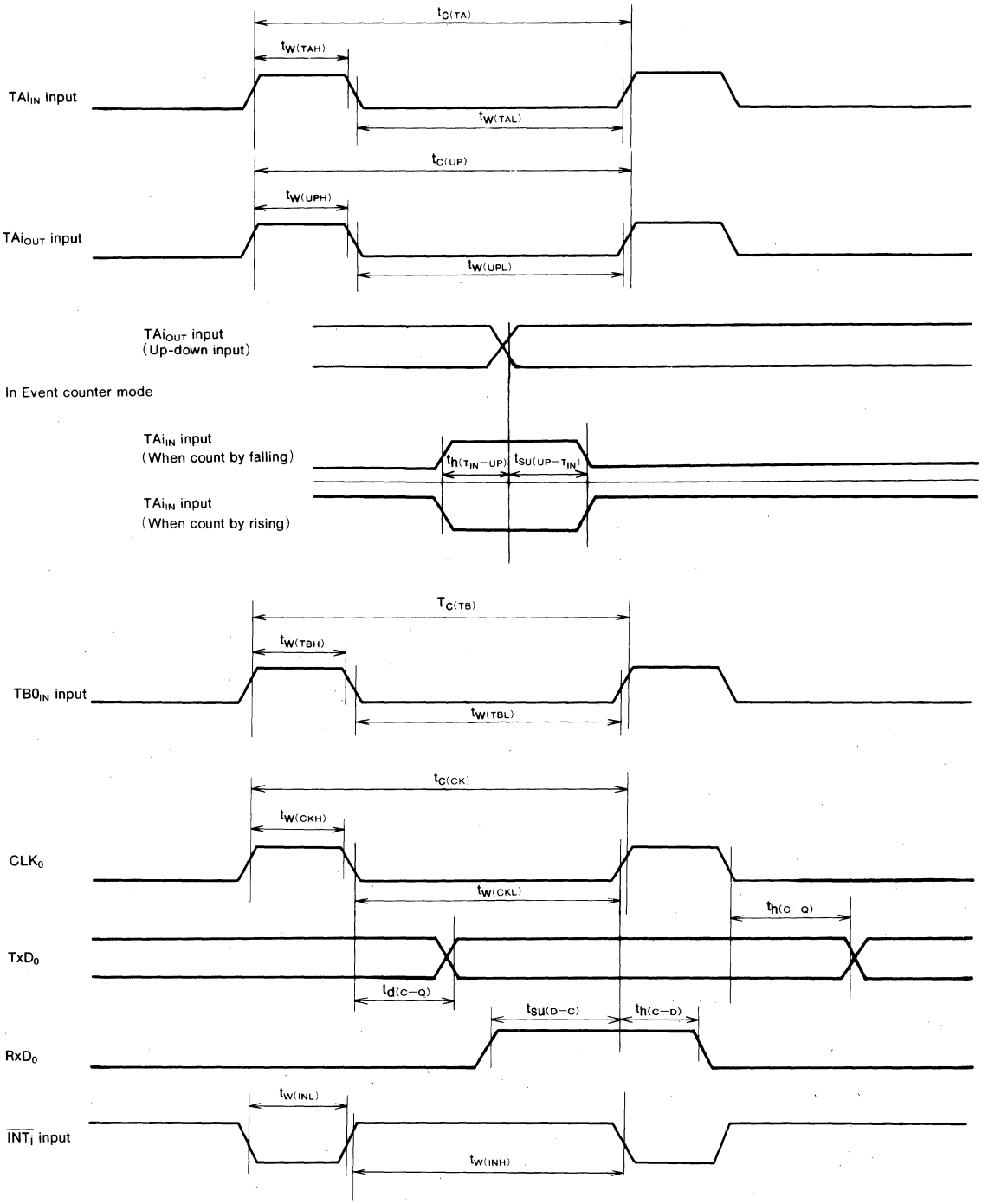


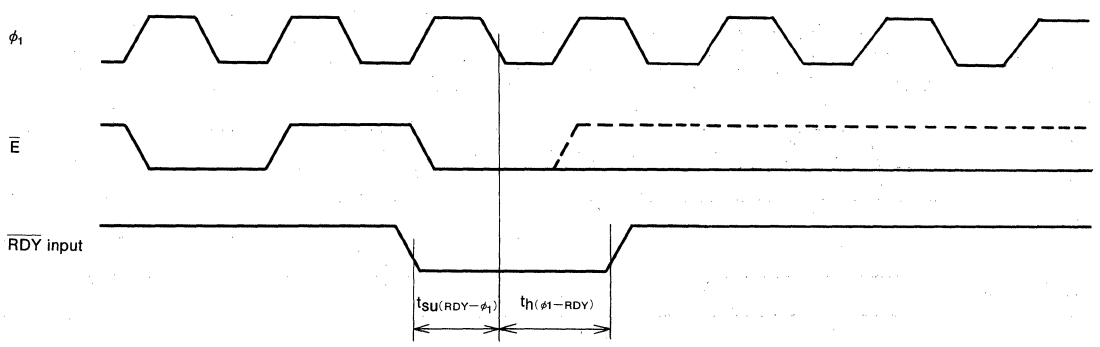
Fig. 60 Testing circuit for each terminal

TIMING DIAGRAM

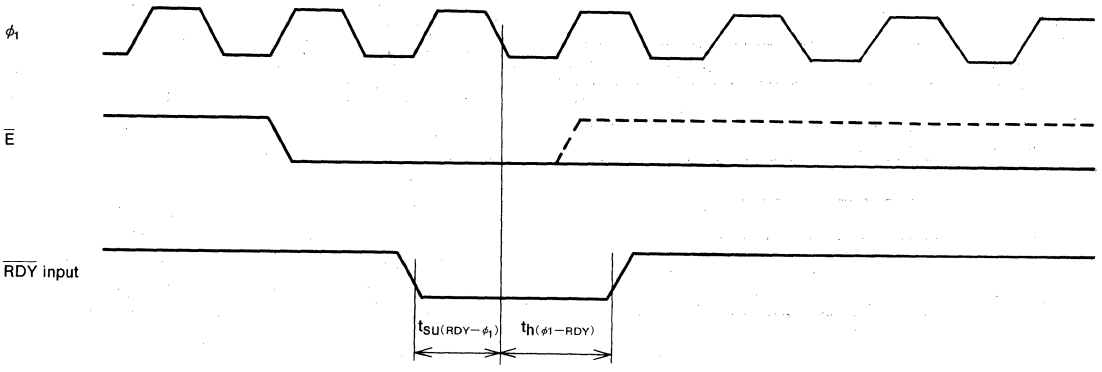



$t_{C(TA)}$
$t_{W(TAH)}$
TAi_{IN} input
$t_{W(TAL)}$
$t_{C(UP)}$
$t_{W(UPH)}$
TAi_{OUT} input
$t_{W(UPL)}$
TAi_{OUT} input
(Up-down input)
In Event counter mode
TAi_{IN} input
(When count by falling)
$t_{h(T_{IN}-UP)}$
$t_{SU(UP-T_{IN})}$
TAi_{IN} input
(When count by rising)
$T_{C(TB)}$
$t_{W(TBH)}$
$TB0_{IN}$ input
$t_{W(TBL)}$
$t_{C(CK)}$
$t_{W(CKH)}$
CLK_0
$t_{W(CKL)}$
$t_{h(C-Q)}$
TxD_0
$t_{d(C-Q)}$
$t_{su(D-C)}$
$t_{h(C-D)}$
RxD_0
$t_{W(INL)}$
$\overline{INT_i}$ input
$t_{W(INH)}$

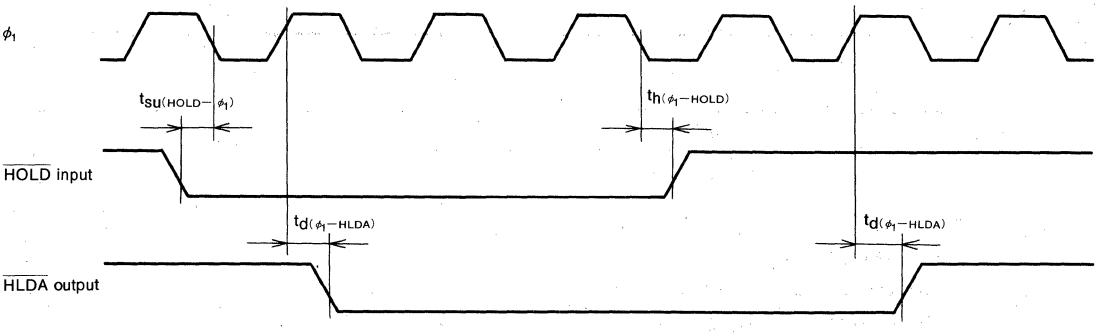
Microprocessor mode (When wait bit = "1")



Microprocessor mode (When wait bit = "0")



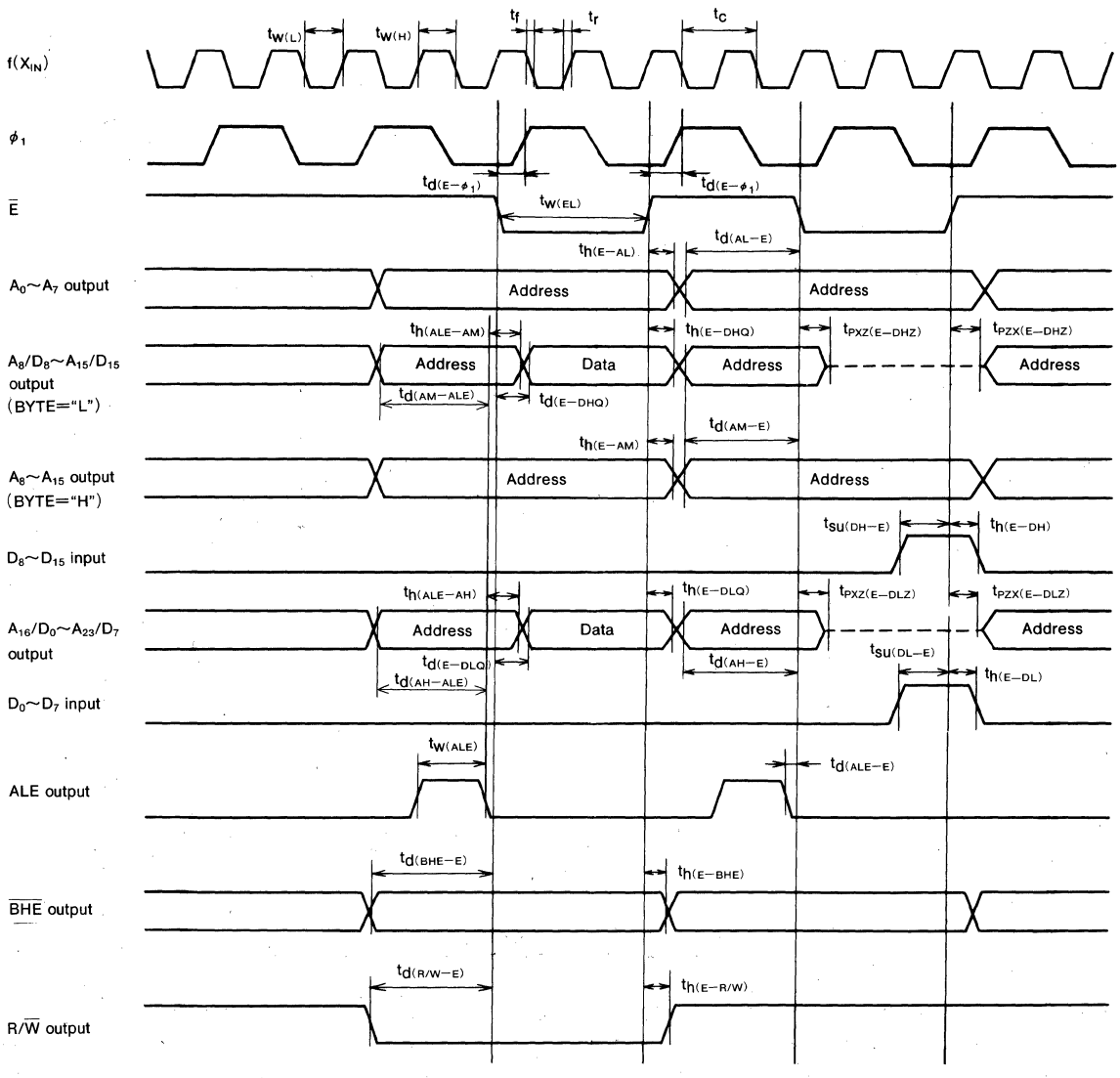
Microprocessor mode (When wait bit = "1" or "0" in common)



Test conditions

- V_{CC} = 5 V±10%
- Input timing voltage : V_{IL}=1.0V, V_{IH}=4.0V
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V

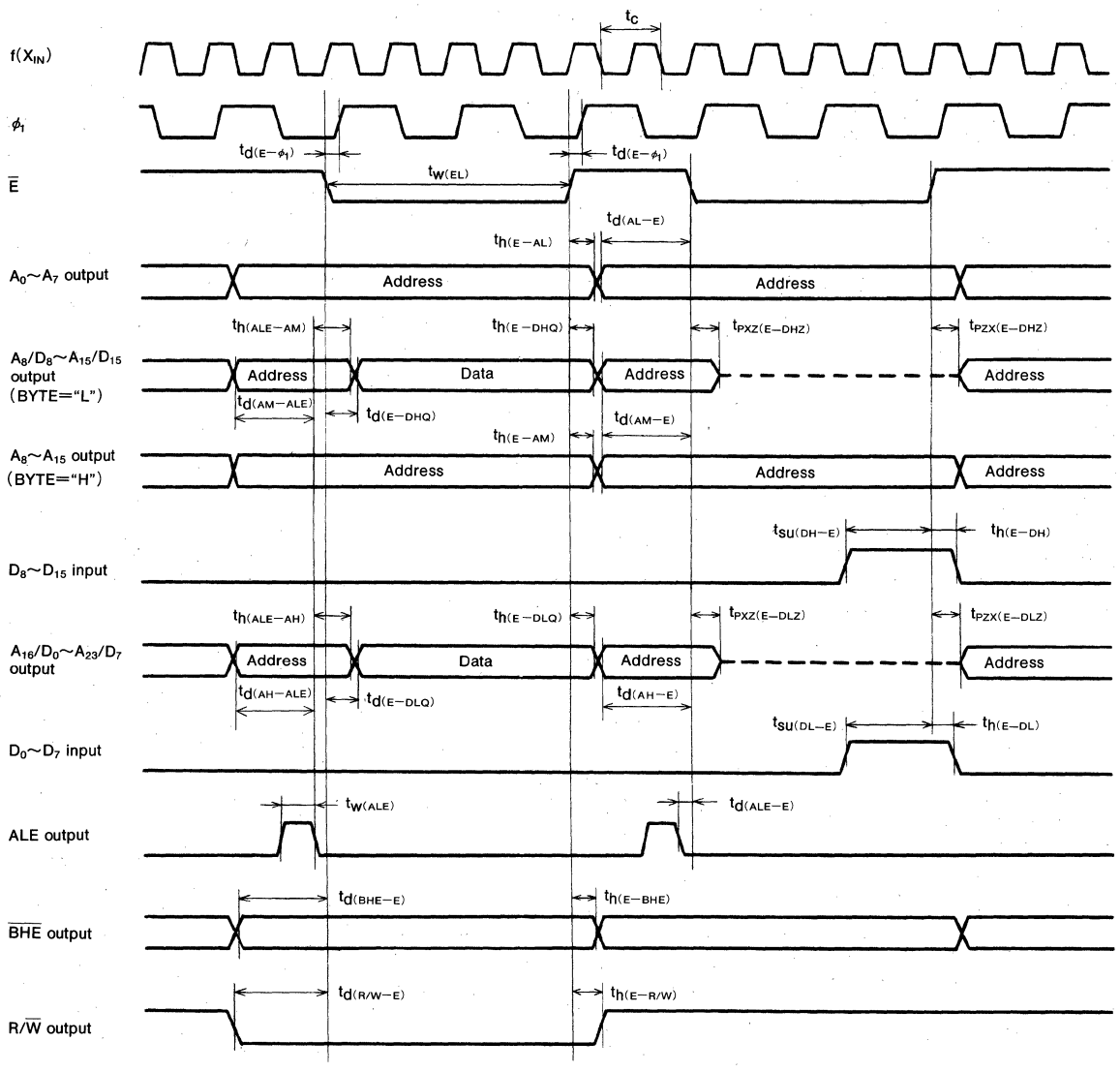
Microprocessor mode (When wait bit = "1")



Test conditions

- V_{CC}= 5 V±10%
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V
- D_0~D_{15} input : V_{IL}=0.8V, V_{IH}=2.5V

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



Test conditions

- V_{CC}= 5 V±10%
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V
- D_0~D_{15} input : V_{IL}=0.8V, V_{IH}=2.5V

M37732S4FP, M37732S4AFP M37732S4BFP

16-BIT CMOS MICROCOMPUTER

DESCRIPTION

The M37732S4FP is a single-chip microcomputer designed with high-performance CMOS silicon gate technology. This is housed in a 80-pin plastic molded QFP. This single-chip microcomputer has a large 16M bytes address space, three instruction queue buffers, and two data buffers for high-speed instruction execution. The CPU is a 16-bit parallel processor that can also be switched to perform 8-bit parallel processing. This microcomputer is suitable for office, business, and industrial equipment controller that require high-speed processing of large data.

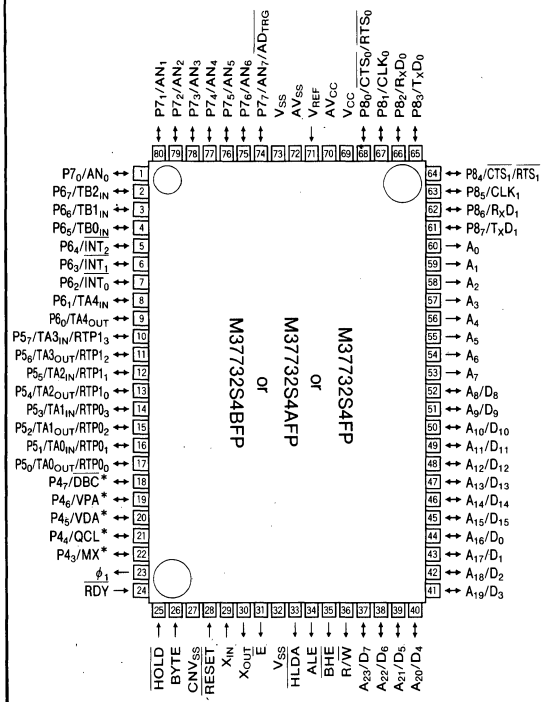
The differences between M37732S4FP, M37732S4AFP and M37732S4BFP are the external clock input frequency as shown below. Therefore, the following descriptions will be for the M37732S4FP unless otherwise noted.

Type name	ROM size	External clock input frequency
M37732S4FP	External	8 MHz
M37732S4AFP	External	16MHz
M37732S4BFP	External	25MHz

FEATURES

- Number of basic instructions 103
- Memory size RAM ..... 2048 bytes
- Instruction execution time
  M37732S4FP
  (The fastest instruction at 8MHz frequency) 500ns
  M37732S4AFP
  (The fastest instruction at 16MHz frequency) 250ns
  M37732S4BFP
  (The fastest instruction at 25MHz frequency) 160ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 8MHz frequency)
  ..... 30mW (Typ.)
- Interrupts ..... 19 types 7 levels
- Multiple function 16-bit timer ..... 5+3
- UART (may also be synchronous) 2
- 8-bit A-D converter ..... 8-channel inputs
- 12-bit watchdog timer
- Programmable input/output
  (ports P4, P5, P6, P7, P8) 37
- Pulse output port ..... 4-bit×2

PIN CONFIGURATION (TOP VIEW)



Outline 80P6N

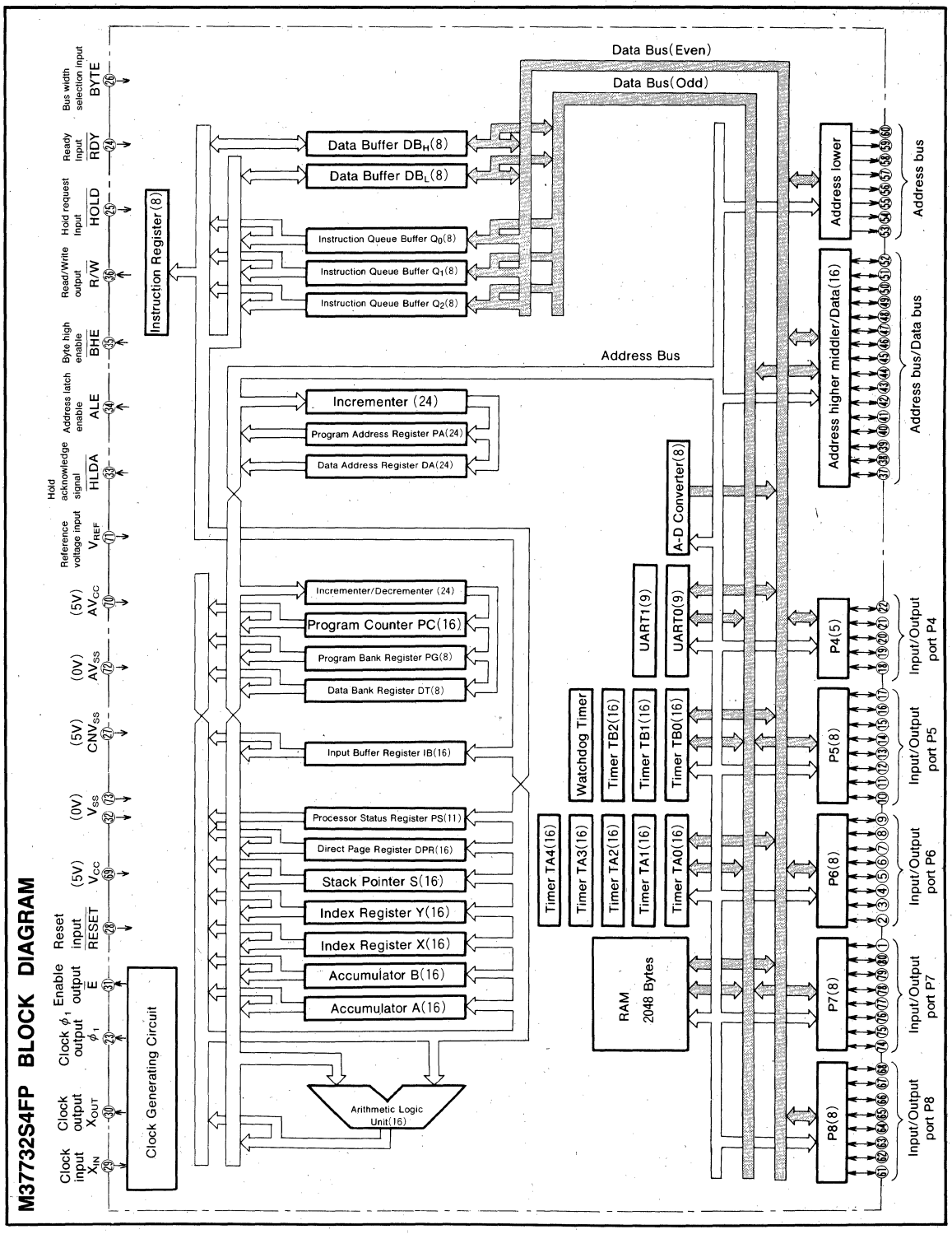
*: Used in the evaluation chip mode only

APPLICATION

Control devices for office equipment such as copiers, printers, typewriters, facsimiles, word processors, and personal computers

Control devices for industrial equipment such as ME, NC, communication and measuring instruments.

M37732S4FP BLOCK DIAGRAM



MITSUBISHI ELECTRIC

FUNCTIONS OF M37732S4FP

Parameter		Functions
Number of basic instructions		103
Instruction execution time	M37732S4FP	500ns (the fastest instructions, at 8MHz frequency)
	M37732S4AFP	250ns (the fastest instructions, at 16MHz frequency)
	M37732S4BFP	160ns (the fastest instructions, at 25MHz frequency)
Memory size	RAM	2048 bytes
Input/Output ports	P5～P8	8-bit× 4
	P4	5-bit× 1
Multi-function timers	TA0, TA1, TA2, TA3, TA4	16-bit× 5
	TB0, TB1, TB2	16-bit× 3
Serial I/O		(UART or clock synchronous serial I/O)×2
A-D converter		8-bit× 1 (8 channels)
Watchdog timer		12-bit× 1
Interrupts		3 external types, 16 internal types (Each interrupt can be set the priority levels to 0～7.)
Clock generating circuit		Built-in(externally connected to a ceramic resonator or quartz crystal resonator)
Supply voltage		5 V±10%
Power dissipation		30mW(at external 8 MHz frequency)
Input/Output characteristic	Input/Output voltage	5 V
	Output current	5 mA
Memory expansion		16M bytes
Operating temperature range		−20～85℃
Device structure		CMOS high-performance silicon gate process
Package		80-pin plastic molded QFP

PIN DESCRIPTION

Pin	Name	Input/Output	Functions
V_{CC}, V_{SS}	Power supply		Supply 5 V±10% to V_{CC} and 0 V to V_{SS}.
CNV_{SS}	CNV_{SS} input	Input	Connect to V_{CC}.
$\overline{RESET}$	Reset input	Input	To enter the reset state, this pin must be kept at a "L" condition should be maintained for the required time.
X_{IN}	Clock input	Input	These are I/O pins of internal clock generating circuit. Connect a ceramic or quartz crystal resonator between X_{IN} and X_{OUT}. When an external clock is used, the clock source should be connected to the X_{IN} pin and the X_{OUT} pin should be left open.
X_{OUT}	Clock output	Output	
$\overline{E}$	Enable output	Output	Data or instruction read and data write are performed when output from this pin is "L".
BYTE	Bus width selection input	Input	This pin determines whether the external data bus is 8-bit width or 16-bit width. The width is 16 bits when "L" signal inputs and 8 bits when "H" signal inputs.
AV_{CC}, AV_{SS}	Analog supply input		Power supply for A-D converter. Connect AV_{CC} to V_{CC}, and AV_{SS} to V_{SS} externally.
V_{REF}	Reference voltage input	Input	This is reference voltage input pin for A-D converter.
ϕ_1	Clock output	Output	This pin outputs the clock ϕ_1 which is divided the clock to X_{IN} pin by 2.
$\overline{RDY}$	Ready	Input	This is ready input pin. This is an input pin for the $\overline{RDY}$ signal. Internal clock stops while this signal is "L".
$\overline{HOLD}$	Hold request input	Input	This is an input pin for $\overline{HOLD}$ request signal. The microcomputer enters into hold state while this signal is "L".
$\overline{HLDA}$	Hold acknowledge output	Output	This is an output pin for $\overline{HLDA}$ signal, indicates the hold state.
R/$\overline{W}$	Read/Write output	Output	"H" indicates the read status and "L" indicates the write status.
$\overline{BHE}$	Byte high enable output	Output	"L" is output when an odd-numbered address is accessed.
ALE	Address latch enable output	Output	This is used to retrieve only the address data from address data and data multiplex signal.
A_0~A_7	Address (low-order) output	Output	Address (A_7~A_0) is output.
A_8/D_8~A_{15}/D_{15}	Address (middle-order) output/Data (high-order) I/O	I/O	In case the BYTE pin is "L" and an external data bus is 16-bit width, high-order data (D_{15}~D_8) is input or output when $\overline{E}$ output is "L" and an address (A_{15}~A_8) is output when $\overline{E}$ output is "H". In case the BYTE pin is "H" and an external data bus is 8-bit width, only address (A_{15}~A_8) is output.
A_{16}/D_0~A_{23}/D_7	Address (high-order) output/Data (low-order) I/O	I/O	Low-order data (D_7~D_0) is input or output when $\overline{E}$ output is "L", and an address (A_{23}~A_{16}) is output when $\overline{E}$ output is "H".
$P4_3$~$P4_7$	I/O port P4	I/O	Port P4 is a 5-bit I/O port. This port has an data direction register and each pin can be programmed for input or output. This port is in input mode when reset.
$P5_0$~$P5_7$	I/O port P5	I/O	Port P5 is a 8-bit I/O port. This port has an data direction register and each pin can be programmed for input or output. This port is in input mode when reset. These pins also function as I/O pins for timer A0, timer A1, timer A2 and timer A3.
$P6_0$~$P6_7$	I/O port P6	I/O	In addition to having the same functions as port P5, these pins also function as I/O pins for timer A4, external interrupt input $\overline{INT_0}$, $\overline{INT_1}$ and $\overline{INT_2}$ pins, and input pin for timer B0, timer B1 and timer B2.
$P7_0$~$P7_7$	I/O port P7	I/O	Port P7 is a 8-bit I/O port. This port has an data direction register and each pin can be programmed for input or output. This port is input mode when reset. These pins also function as analog input AN_0~AN_7 input pins. $P7_7$ also has an A-D conversion trigger input function.
$P8_0$~$P8_7$	I/O port P8	I/O	Port P8 is a 8-bit I/O port. This port has an data direction register and each pin can be programmed for input or output. This port is in input mode when reset. These pins also function as RxD, TxD, CLK, $\overline{CTS}/\overline{RTS}$ pins for UART0 and UART1.

BASIC FUNCTION BLOCKS

The M37732S4FP contains the following devices on a single chip: RAM for storing instructions and data, CPU for processing, bus interface unit (which controls instruction prefetch and data read/write between CPU and memory), timers, UART, A-D converter, and other peripheral devices such as I/O ports. Each of these devices are described below.

MEMORY

The memory map is shown in Figure 1. The address space is 16M bytes from addresses 0_{16} to $FFFFFF_{16}$. The address space is divided into 64K bytes units called banks. The banks are numbered from 0_{16} to FF_{16}.

Built-in RAM and control registers for built-in peripheral devices are assigned to bank 0_{16}.

Addresses $FFD6_{16}$ to $FFFF_{16}$ are the RESET and interrupt vector addresses and contain the interrupt vectors. Use ROM for memory of this address. Refer to the section on interrupts for details.

The 2048 bytes area from addresses 80_{16} to $87F_{16}$ contains the built-in RAM. In addition to storing data, the RAM is used as stack during a subroutine call, or interrupts.

Assigned to addresses 0_{16} to $7F_{16}$ are peripheral devices such as I/O ports, A-D converter, UART, timer, and interrupt control registers.

A 256 bytes direct page area can be allocated anywhere in bank 0_{16} using the direct page register DPR. In direct page addressing mode, the memory in the direct page area can be accessed with two words thus reducing program steps.

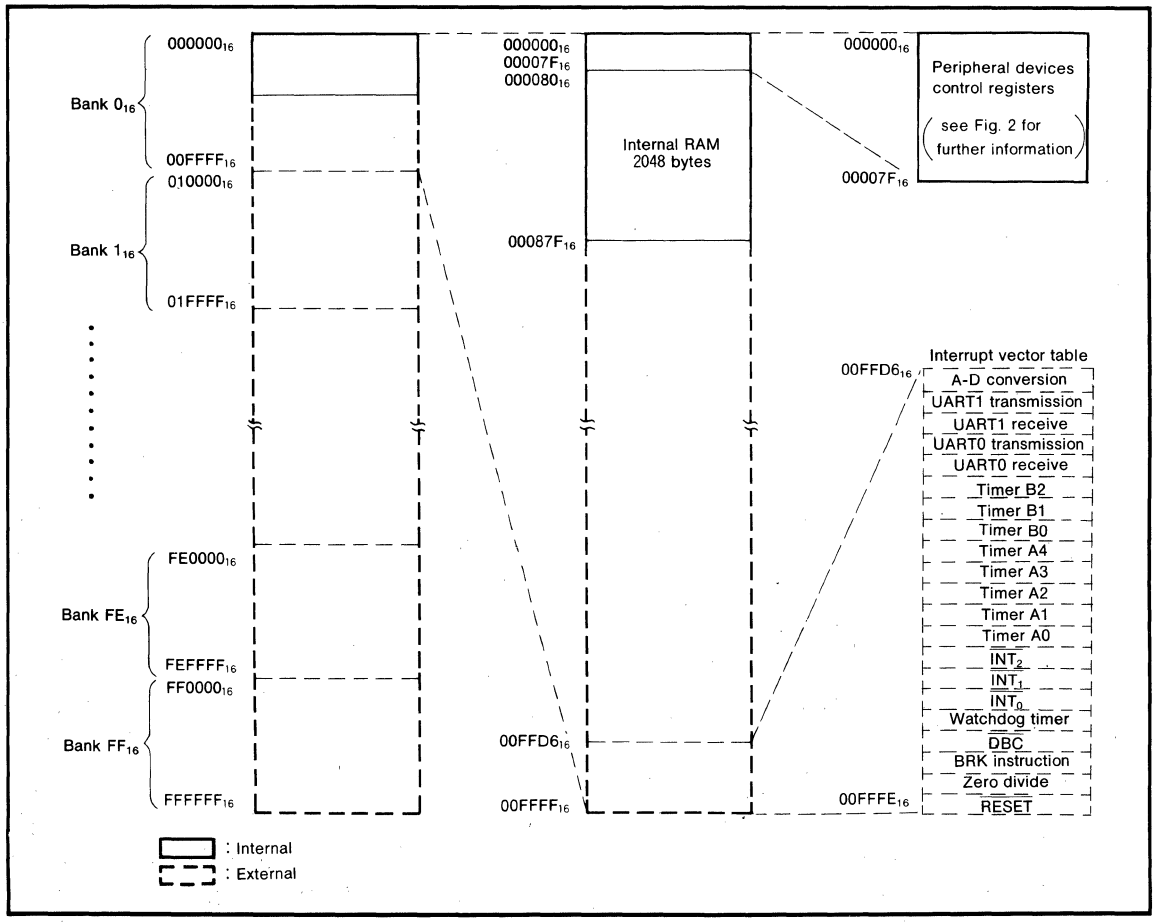


Fig. 1 Memory map

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

Address (Hexadecimal notation)	
000040	Count start flag
000041	
000042	One shot start flag
000043	
000044	Up-down flag
000045	
000046	Timer A0
000047	
000048	Timer A1
000049	
00004A	Timer A2
00004B	
00004C	Timer A3
00004D	
00004E	Timer A4
00004F	
000050	Timer B0
000051	
000052	Timer B1
000053	
000054	Timer B2
000055	
000056	Timer A0 mode register
000057	Timer A1 mode register
000058	Timer A2 mode register
000059	Timer A3 mode register
00005A	Timer A4 mode register
00005B	Timer B0 mode register
00005C	Timer B1 mode register
00005D	Timer B2 mode register
00005E	Processor mode register
00005F	
000060	Watchdog timer
000061	Watchdog timer frequency selection flag
000062	Waveform output mode register
000063	
000064	Pulse output data register 1
000065	Pulse output data register 0
000066	
000067	
000068	
000069	
00006A	
00006B	
00006C	
00006D	
00006E	
00006F	
000070	A-D conversion interrupt control register
000071	UART0 transmission interrupt control register
000072	UART0 receive interrupt control register
000073	UART1 transmission interrupt control register
000074	UART1 receive interrupt control register
000075	Timer A0 interrupt control register
000076	Timer A1 interrupt control register
000077	Timer A2 interrupt control register
000078	Timer A3 interrupt control register
000079	Timer A4 interrupt control register
00007A	Timer B0 interrupt control register
00007B	Timer B1 interrupt control register
00007C	Timer B2 interrupt control register
00007D	$\overline{INT_0}$ interrupt control register
00007E	$\overline{INT_1}$ interrupt control register
00007F	$\overline{INT_2}$ interrupt control regiter

Fig. 2. Location of peripheral devices and interrupt control registers

CENTRAL PROCESSING UNIT (CPU)

The CPU has ten registers and is shown in Figure 3. Each of these registers is described below.

ACCUMULATOR A (A)

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

Data operations such as calculations, data transfer, input/output, etc., is executed mainly through the accumulator.

ACCUMULATOR B (B)

Accumulator B has the same functions as accumulator A, but the use of accumulator B requires more instruction bytes and execution cycles than accumulator A.

INDEX REGISTER X (X)

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

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

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

INDEX REGISTER Y (Y)

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

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

Also, when executing a block transfer instruction MVP or MVN, the content of index register Y indicates the low-order 16 bits of the destination address. The second byte of the MVP and MVN is the high-order 8 bits of the destination data address.

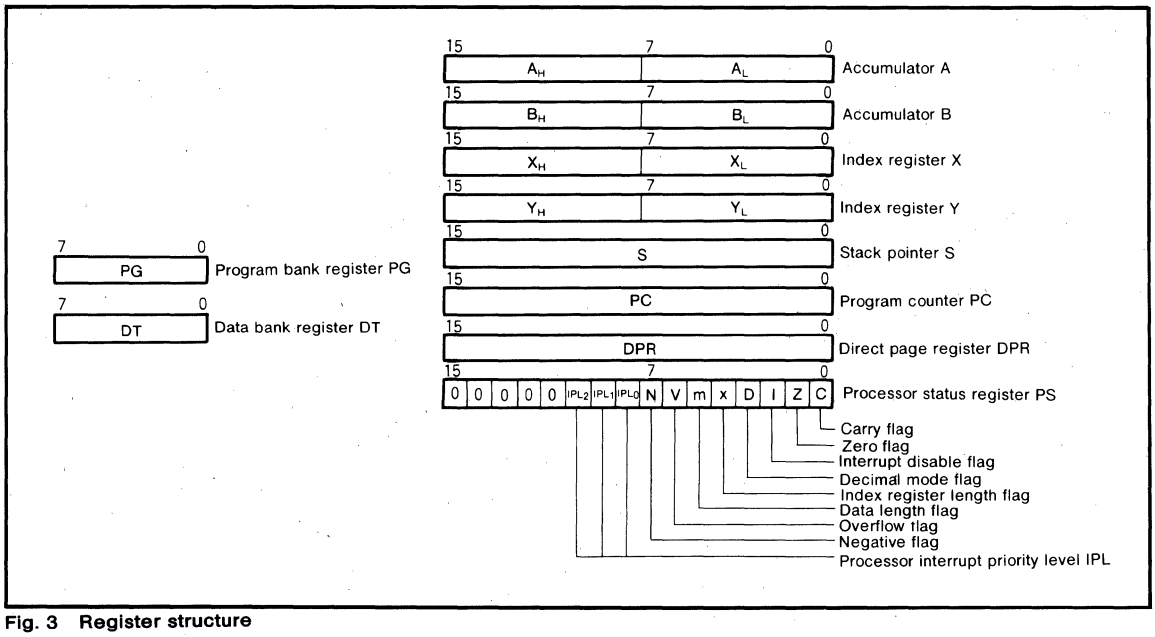


Fig. 3 Register structure

STACK POINTER (S)

Stack pointer (S) is an 16-bit register. It is used during a subroutine call or interrupts. It is also used during stack, stack pointer relative, or stack pointer relative indirect indexed Y addressing mode.

PROGRAM COUNTER (PC)

Program counter (PC) is a 16-bit counter that indicates the low-order 16 bits of the next program memory address to be executed. These is a bus interface unit between the program memory and the CPU, so that the program memory is accessed through bus interface unit. This is described later.

PROGRAM BANK REGISTER (PG)

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

DATA BANK REGISTER (DT)

Data bank register (DT) is an 8-bit register. With some addressing modes, a part of the data bank register (DT) is used to specify a memory address. The contents of data bank register (DT) is used as the high-order 8 bits of a 24-bit address. Addressing modes that use the data bank register (DT) are direct indirect, direct indexed X indirect, direct indirect indexed Y, absolute, absolute bit, absolute indexed X, absolute indexed Y, absolute bit relative, and stack pointer relative indirect indexed Y.

DIRECT PAGE REGISTER (DPR)

Direct page register (DPR) is a 16-bit register. Its contents is used as the base address of a 256-byte direct page area. The direct page area is allocated in bank 0, but when the contents of DPR is $FF01_{16}$ or greater, the direct page area spans across bank 0 and bank 1. All direct addressing modes use the contents of the direct page register (DPR) to generate the data address. If the low-order 8 bits of the direct page register (DPR) is "00_{16}", the number of cycles required to generate an address is minimized. Normally the low-order 8 bits of the direct page register (DPR) is set to "00_{16}".

PROCESSOR STATUS REGISTER (PS)

Processor status register (PS) is an 11-bit register. It consists of a flag to indicate the result of operation and CPU interrupt levels.

Branch operations can be performed by testing the flags C, Z, V, and N.

The details of each processor status register bit are described below.

1. Carry flag (C)

The carry flag contains the carry or borrow generated by the ALU after an arithmetic operation. This flag is also affected by shift and rotate instructions. This flag can be set and reset directly with the SEC and CLC instructions or with the SEP and CLP instructions.

2. Zero flag (Z)

This zero flag is set if the result of an arithmetic operation or data transfer is zero and reset if it is not. This flag can be set and reset directly with the SEP and CLP instructions.

3. Interrupt disable flag (I)

When the interrupt disable flag is set to "1", all interrupts except watchdog timer, $\overline{\text{DBC}}$, and software interrupt are disabled. This flag is set to "1" automatically when there is an interrupt. It can be set and reset directly with the SEI and CLI instructions or SEP and CLP instructions.

4. Decimal mode flag (D)

The decimal mode flag determines whether addition and subtraction are performed as binary or decimal. Binary arithmetic is performed when this flag is "0". If it is "1", decimal arithmetic is performed with each word treated as two or four digit decimal. Arithmetic operation is performed using four digits when the data length flag m is "0" and with two digits when it is "1". (Decimal operation is possible only with the ADC and SBC instructions.) This flag can be set and reset with the SEP and CLP instructions.

5. Index register length flag (x)

The index register length flag determines whether index register X and index register Y are used as 16-bit registers or as 8-bit registers. The registers are used as 16-bit registers when flag x is "0" and as 8-bit registers when it is "1". This flag can be set and reset with the SEP and CLP instructions.

6. Data length flag (m)

The data length flag determines whether the data length is 16-bit or 8-bit. The data length is 16-bit when flag m is "0" and 8-bit when it is "1". This flag can be set and reset with the SEM and CLM instructions or with the SEP and CLP instructions.

7. Overflow flag (V)

The overflow flag has meaning when addition or subtraction is performed a word as signed binary number. When the data length flag m is "0", the overflow flag is set when the result of addition or subtraction is outside the range between −32768 and +32767. When the data length flag m is "1", the overflow flag is set when the result of addition or subtraction is outside the range between −128 and +127. It is reset in all other cases. The overflow flag can also be set and reset directly with the SEP, and CLV or CLP instructions.

8. Negative flag (N)

The negative flag is set when the result of arithmetic operation or data transfer is negative (If data length flag m is "0", when data bit 15 is "1". If data length flag m is "1", when data bit 7 is "1".) It is reset in all other cases. It can also be set and reset with the SEP and CLP instructions.

9. Processor interrupt priority level (IPL)

The processor interrupt priority level (IPL) consists of 3 bits and determines the priority of processor interrupts from level 0 to level 7. Interrupt is enabled when the interrupt priority of the device requesting interrupt (set using the interrupt control register) is higher than the processor interrupt priority. When interrupt is enabled, the current processor interrupt priority level is saved in a stack and the processor interrupt priority level is replaced by the interrupt priority level of the device requesting the interrupt. Refer to the section on interrupts for more details.

BUS INTERFACE UNIT

The CPU operates on an internal clock frequency which is obtained by dividing the external clock frequency $f_{(X_{IN})}$ by two. This frequency is twice the bus cycle frequency. In order to speed-up processing, a bus interface unit is used to pre-fetch instructions when the data bus is idle. The bus interface unit synchronizes the CPU and the bus and prefetches instructions. Figure 4 shows the relationship between the CPU and the bus interface unit. The bus interface unit has a program address register, a 3-byte instruction queue buffer, a data address register, and a 2-byte data buffer.

The bus interface unit obtains an instruction code from memory and stores it in the instruction queue buffer, obtains data from memory and stores it in the data buffer, or writes the data from the data buffer to the memory.

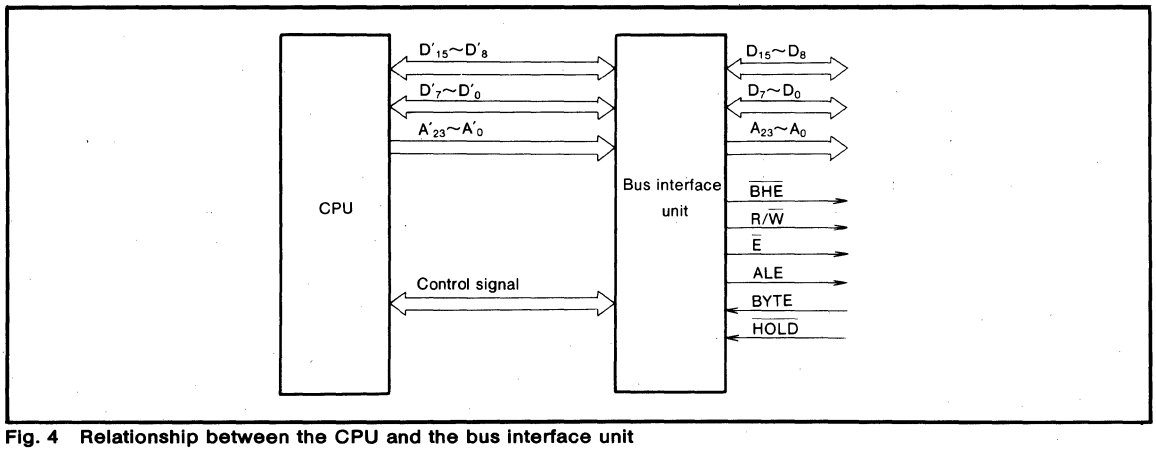


Fig. 4 Relationship between the CPU and the bus interface unit

The bus interface unit operates using one of the waveforms (1) to (6) shown in Figure 5. The standard waveforms are (1) and (2).

The ALE signal is used to latch only the address signal from the multiplexed signal containing data and address.

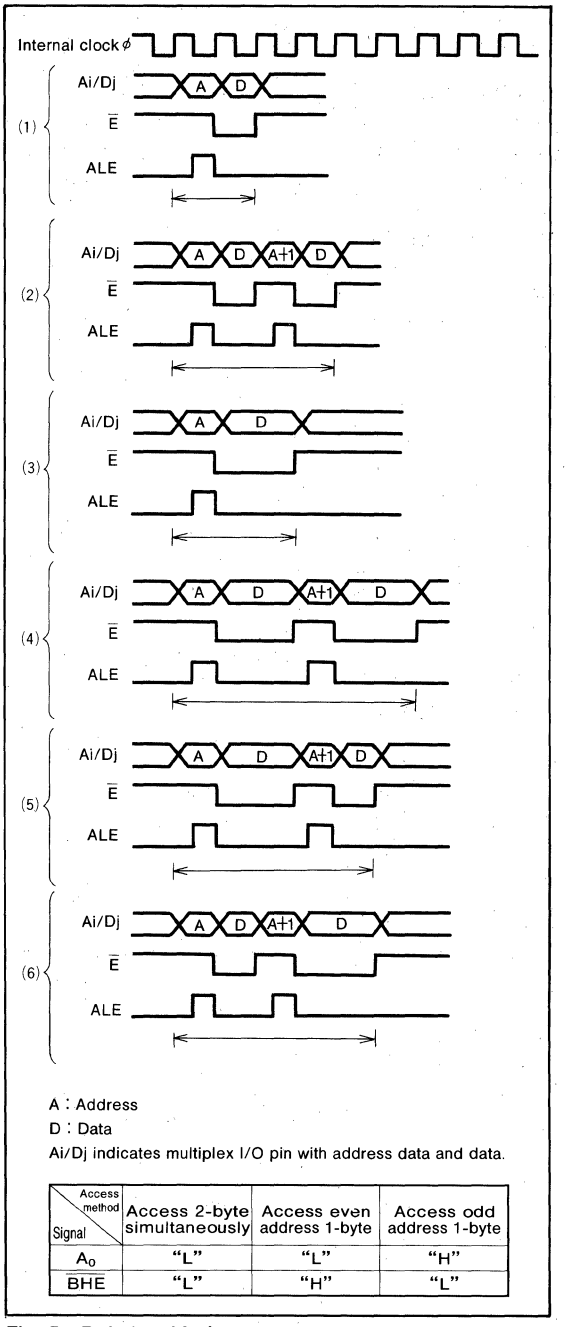
The $\overline{E}$ signal becomes "L" when the bus interface unit reads an instruction code or data from memory or when it writes data to memory. Whether to perform read or write is controlled by the R/$\overline{W}$ signal. Read is performed when the R/$\overline{W}$ signal is "H" state and write is performed when it is "L" state.

Waveform (1) in Figure 5 is used to access a single byte or two bytes simultaneously. To read or write two bytes simultaneously, the first address accessed must be even. Furthermore, when accessing an external memory area, set the bus width selection input pin BYTE to "L". (external data bus width to 16 bits) The internal memory area is always treated as 16-bit bus width regardless of BYTE.

When performing 16-bit data read or write, if the conditions for simultaneously accessing two bytes are not satisfied, waveform (2) is used to access each byte one by one. However, when prefetching the instruction code, if the address of the instruction code is odd, waveform (1) is used, and only one byte is read in the instruction queue buffer.

The signals A_0 and $\overline{BHE}$ in Figure 5 are used to control these cases: 1-byte read from even address, 1-byte read from odd address, 2-byte simultaneous read from even and odd addresses, 1-byte write to even address, 1-byte write to odd address, or 2-byte simultaneous write to even and odd addresses. The A_0 signal that is the address bit 0 is "L" when an even number address is accessed. The $\overline{BHE}$ signal becomes "L" when an odd number address is accessed.

The bit 2 of processor mode register (address $5E_{16}$) is the wait bit. When this bit is set to "0", the "L" width of $\overline{E}$ signal is 2 times as long when accessing an external memory area. However, the "L" width of $\overline{E}$ signal is not extended when an internal memory area is accessed. When the wait bit is "1", the "L" width of $\overline{E}$ signal is not extended for any access. Waveform (3) is an expansion of the "L" width of $\overline{E}$ signal in waveform (1). Waveform (4),(5), and (6) are expansion of each "L" width of $\overline{E}$ signal in waveform (2), first half of waveform (2), and the last half of waveform (2) respectively.



Access method / Signal	Access 2-byte simultaneously	Access even address 1-byte	Access odd address 1-byte
A_0	"L"	"L"	"H"
$\overline{BHE}$	"L"	"H"	"L"

Fig. 5 Relationship between access method and signals A_0 and $\overline{BHE}$

Instruction code read, data read, and data write are described below.

Instruction code read will be described first.

The CPU obtains instruction codes from the instruction queue buffer and executes them. The CPU notifies the bus interface unit that it is requesting an instruction code during an instruction code request cycle. If the requested instruction code is not yet stored in the instruction queue buffer, the bus interface unit halts the CPU until it can store more instructions than requested in the instruction queue buffer.

Even if there is no instruction code request from the CPU, the bus interface unit reads instruction codes from memory and stores them in the instruction queue buffer when the instruction queue buffer is empty or when only one instruction code is stored and the bus is idle on the next cycle.

This is referred to as instruction pre-fetching.

Normally, when reading an instruction code from memory, if the accessed address is even the next odd address is read together with the instruction code and stored in the instruction queue buffer.

However, if the bus width switching pin BYTE is "H", external data bus width is 8 bits and the address to be read is in external memory area is odd, only one byte is read and stored in the instruction queue buffer. Therefore, waveform (1) or (3) in Figure 5 is used for instruction code read.

Data read and write are described below.

The CPU notifies the bus interface unit when performing data read or write. At this time, the bus interface unit halts the CPU if the bus interface unit is already using the bus or if there is a request with higher priority. When data read or write is enabled, the bus interface unit uses one of the waveforms from (1) to (6) in Figure 5 to perform the operation.

During data read, the CPU waits until the entire data is stored in the data buffer. The bus interface unit sends the address received from the CPU to the address bus. Then it reads the memory when the $\overline{E}$ signal is "L" and stores the result in the data buffer.

During data write, the CPU writes the data in the data buffer and the bus interface unit writes it to memory. Therefore, the CPU can proceed to the next step without waiting for write to complete. The bus interface unit sends the address received from the CPU to the address bus. Then when the $\overline{E}$ signal is "L", the bus interface unit sends the data in the data buffer to the data bus and writes it to memory.

INTERRUPTS

Table 1 shows the interrupt types and the corresponding interrupt vector addresses. Reset is also treated as a type of interrupt and is discussed in this section, too.

$\overline{DBC}$ is an interrupt used during debugging.

Interrupts other than reset, $\overline{DBC}$, watchdog timer, zero divide, and BRK instruction all have interrupt control registers. Table 2 shows the addresses of the interrupt control registers and Figure 6 shows the bit configuration of the interrupt control register.

The interrupt request bit is automatically cleared by the hardware during reset or when processing an interrupt. Also, interrupt request bits other than $\overline{DBC}$ and watchdog timer can be cleared by software.

$\overline{INT_2}$ to $\overline{INT_0}$ are external interrupts and whether to cause an interrupt at the input level (level sense) or at the edge (edge sense) can be selected with the level sense/edge sense selection bit. Furthermore, the polarity of the interrupt input can be selected with polarity selection bit.

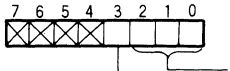
Timer and UART interrupts are described in the respective section.

The priority of interrupts when multiple interrupts are caused simultaneously is partially fixed by hardware, but, it can also be adjusted by software as shown in Figure 7. The hardware priority is fixed the following:

reset > $\overline{DBC}$ > watchdog timer > other interrupts

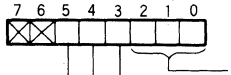
Table 1. Interrupt types and the interrupt vector addresses

Interrupts	Vector addresses
A-D conversion	$00FFD6_{16}$ $00FFD7_{16}$
UART1 transmit	$00FFD8_{16}$ $00FFD9_{16}$
UART1 receive	$00FFDA_{16}$ $00FFDB_{16}$
UART0 transmit	$00FFDC_{16}$ $00FFDD_{16}$
UART0 receive	$00FFDE_{16}$ $00FFDF_{16}$
Timer B2	$00FFE0_{16}$ $00FFE1_{16}$
Timer B1	$00FFE2_{16}$ $00FFE3_{16}$
Timer B0	$00FFE4_{16}$ $00FFE5_{16}$
Timer A4	$00FFE6_{16}$ $00FFE7_{16}$
Timer A3	$00FFE8_{16}$ $00FFE9_{16}$
Timer A2	$00FFEA_{16}$ $00FFEB_{16}$
Timer A1	$00FFEC_{16}$ $00FFED_{16}$
Timer A0	$00FFEE_{16}$ $00FFEF_{16}$
$\overline{INT_2}$ external interrupt	$00FFF0_{16}$ $00FFF1_{16}$
$\overline{INT_1}$ external interrupt	$00FFF2_{16}$ $00FFF3_{16}$
$\overline{INT_0}$ external interrupt	$00FFF4_{16}$ $00FFF5_{16}$
Watchdog timer	$00FFF6_{16}$ $00FFF7_{16}$
$\overline{DBC}$ (unusable)	$00FFF8_{16}$ $00FFF9_{16}$
Break instruction	$00FFFA_{16}$ $00FFFB_{16}$
Zero divide	$00FFFC_{16}$ $00FFFD_{16}$
Reset	$00FFFE_{16}$ $00FFFF_{16}$



Interrupt priority
Interrupt request bit
0 : No interrupt
1 : Interrupt

Interrupt control register configuration for A-D converter, UART0, UART1, timer A0 to timer A4, and timer B0 to timer B2



Interrupt priority
Interrupt request bit
0 : No interrupt
1 : Interrupt
Polarity selection bit
0 : Set interrupt request bit at "H" level for level sense and when changing from "H" to "L" level for edge sense.
1 : Set interrupt request bit at "L" level for level sense and when changing from "L" to "H" level for edge sense.
Level sense/edge sense selection bit
0 : Edge sense
1 : Level sense

Interrupt control register configuration for $\overline{INT_2}$~$\overline{INT_0}$

Fig. 6 Interrupt control register configuration

Table 2. Addresses of interrupt control registers

Interrupt control registers	Addresses
A-D conversion interrupt control register	000070_{16}
UART0 transmit interrupt control register	000071_{16}
UART0 receive interrupt control register	000072_{16}
UART1 transmit interrupt control register	000073_{16}
UART1 receive interrupt control register	000074_{16}
Timer A0 interrupt control register	000075_{16}
Timer A1 interrupt control register	000076_{16}
Timer A2 interrupt control register	000077_{16}
Timer A3 interrupt control register	000078_{16}
Timer A4 interrupt control register	000079_{16}
Timer B0 interrupt control register	$00007A_{16}$
Timer B1 interrupt control register	$00007B_{16}$
Timer B2 interrupt control register	$00007C_{16}$
$\overline{INT_0}$ interrupt control register	$00007D_{16}$
$\overline{INT_1}$ interrupt control register	$00007E_{16}$
$\overline{INT_2}$ interrupt control register	$00007F_{16}$

Interrupts caused by a BRK instruction and when dividing by zero are software interrupts and are not included in this list.

Other interrupts previously mentioned are A-D converter, UART, Timer, INT interrupts. The priority of these interrupts can be changed by changing the priority level in the corresponding interrupt control register by software.

Figure 8 shows a diagram of the interrupt priority resolution circuit. When an interrupt is caused, the each interrupt device compares its own priority with the priority from above and if its own priority is higher, then it sends the priority below and requests the interrupt. If the priorities are the same, the one above has priority.

This comparison is repeated to select the interrupt with the highest priority among the interrupts that are being requested. Finally the selected interrupt is compared with the processor interrupt priority level (IPL) contained in the processor status register (PS) and the request is accepted if it is higher than IPL and the interrupt disable flag I is "0". The request is not accepted if flag I is "1". The reset, $\overline{DBC}$, and watchdog timer interrupts are not affected by the interrupt disable flag I.

When an interrupt is accepted, the contents of the processor status register (PS) is saved to the stack and the interrupt disable flag I is set to "1".

Furthermore, the interrupt request bit of the accepted interrupt is cleared to "0" and the processor interrupt priority level (IPL) in the processor status register (PS) is replaced by the priority level of the accepted interrupt.

Therefore, multi-level priority interrupts are possible by resetting the interrupt disable flag I to "0" and enable further interrupts.

For reset, $\overline{DBC}$, watchdog timer, zero divide, and BRK instruction interrupts, which do not have an interrupt control register, the processor interrupt level (IPL) is set as shown in Table 3.

Priority resolution is performed by latching the interrupt request bit and interrupt priority level so that they do not change. They are sampled at the first half and latched at the last half of the operation code fetch cycle.

Because priority resolution takes some time, no sampling pulse is generated for a certain interval even if it is the next operation code fetch cycle.

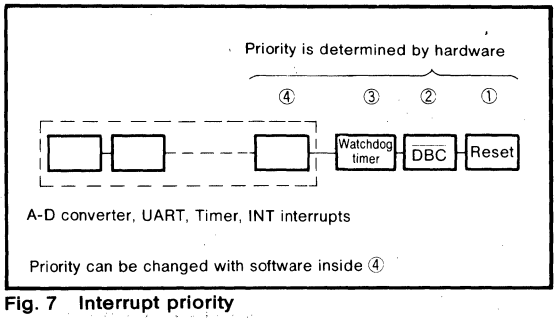


Fig. 7 Interrupt priority

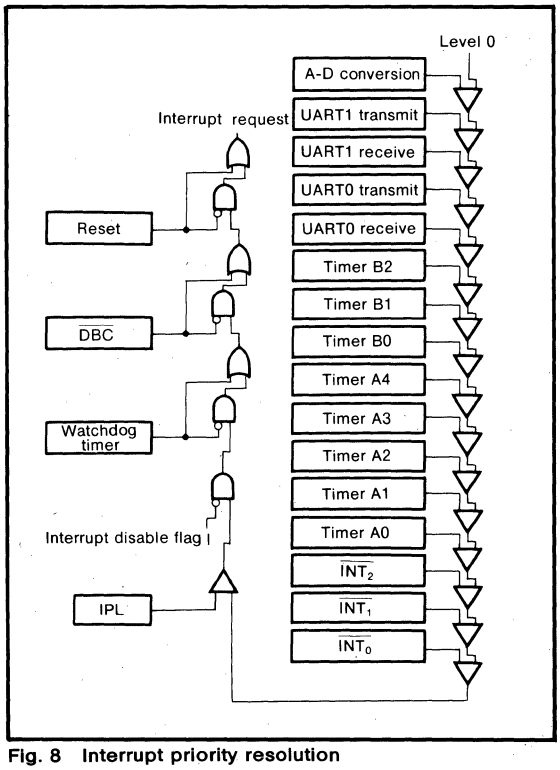


Fig. 8 Interrupt priority resolution

As shown in Figure 9, there are three different interrupt priority resolution time from which one is selected by software. After the selected time has elapsed, the highest priority is determined and is processed after the currently executing instruction has been completed.

The time is selected with bits 4 and 5 of the processor mode register (address $5E_{16}$) shown in Figure 10. Table 4 shows the relationship between these bits and the number of cycles. After a reset, the processor mode register is initialized to "00_{16}" and therefore, the longest time is selected.

However, the shortest time may be selected by software.

Table 3. Value set in processor interrupt level (IPL) during an interrupt

Interrupt types	Setting value
Reset	0
DBC	7
Watchdog timer	7
Zero divide	Not change value of IPL.
BRK instruction	Not change value of IPL.

Table 4. Relationship between priority level evaluation time selection bit and number of cycles

Priority level resolution time selection bit		Number of cycles
Bit 5	Bit 4	
0	0	7 cycles of ϕ
0	1	4 cycles of ϕ
1	0	2 cycles of ϕ

ϕ : internal clock

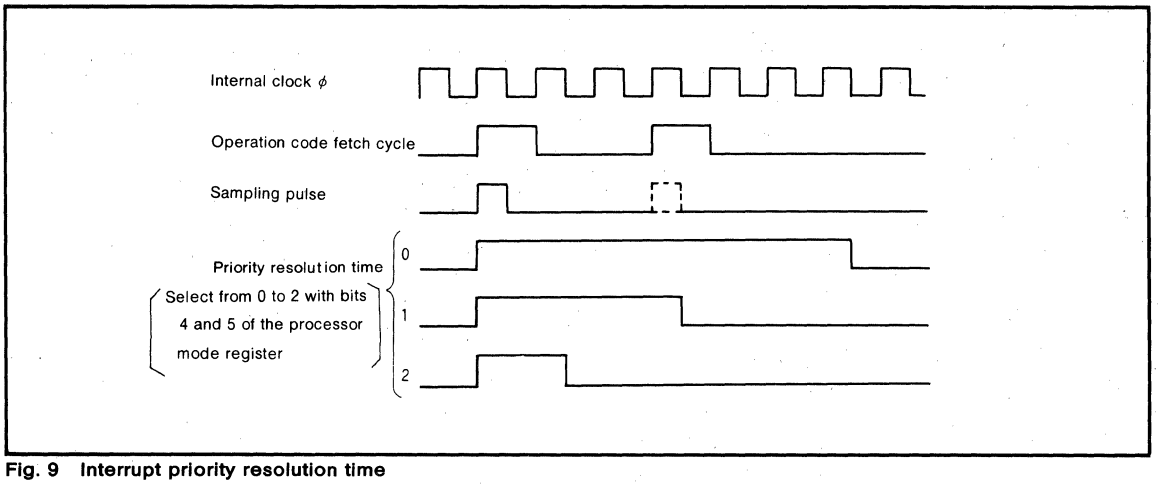


Fig. 9 Interrupt priority resolution time

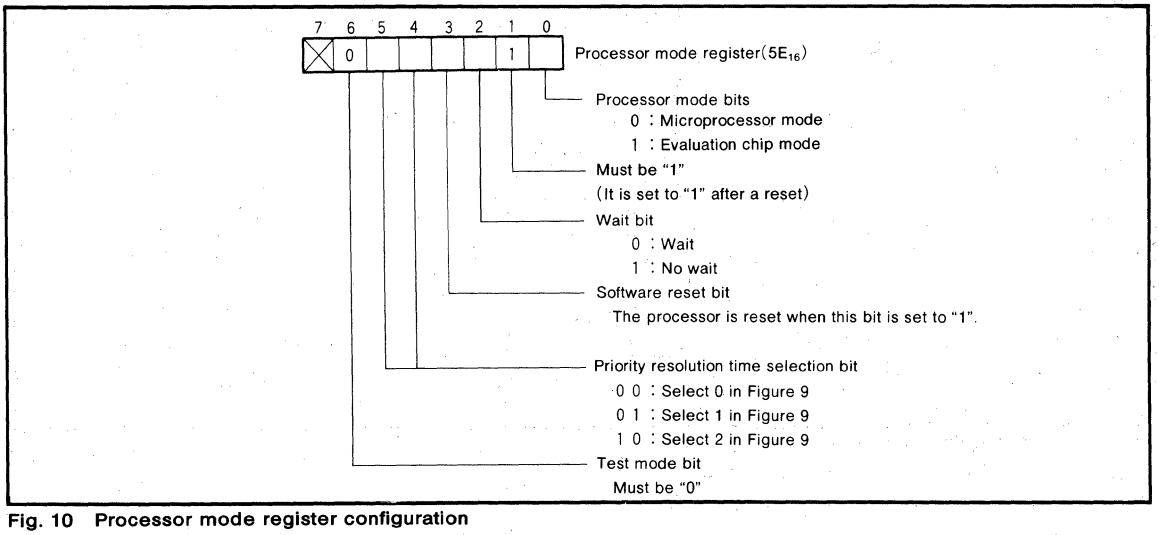


Fig. 10 Processor mode register configuration

TIMER

There are eight 16-bit timers. They are divided by type into timer A(5) and timer B(3).

The timer I/O pins are shared with I/O pins for port P5 and P6. To use these pins as timer input pins, the data direction register bit corresponding to the pin must be cleared to "0" to specify input mode.

TIMER A

Figure 11 shows a block diagram of timer A.

Timer A has four modes; timer mode, event counter mode, one-shot pulse mode, and pulse width modulation mode. The mode is selected with bits 0 and 1 of the timer Ai mode register (i = 0 to 4). Each of these modes is described below.

(1) Timer mode [00]

Figure 12 shows the bit configuration of the timer Ai mode register during timer mode. Bits 0, 1, and 5 of the timer Ai mode register must always be "0" in timer mode.

Bit 3 is ignored if bit 4 is "0".

Bits 6 and 7 are used to select the timer counter source.

The counting of the selected clock starts when the count start flag is "1" and stops when it is "0".

Figure 13 shows the bit configuration of the count start flag.

The counter is decremented, an interrupt is caused and the interrupt request bit in the timer Ai interrupt control register is set when the contents becomes 0000_{16}. At the same time, the contents of the reload register is transferred to the counter and count is continued.

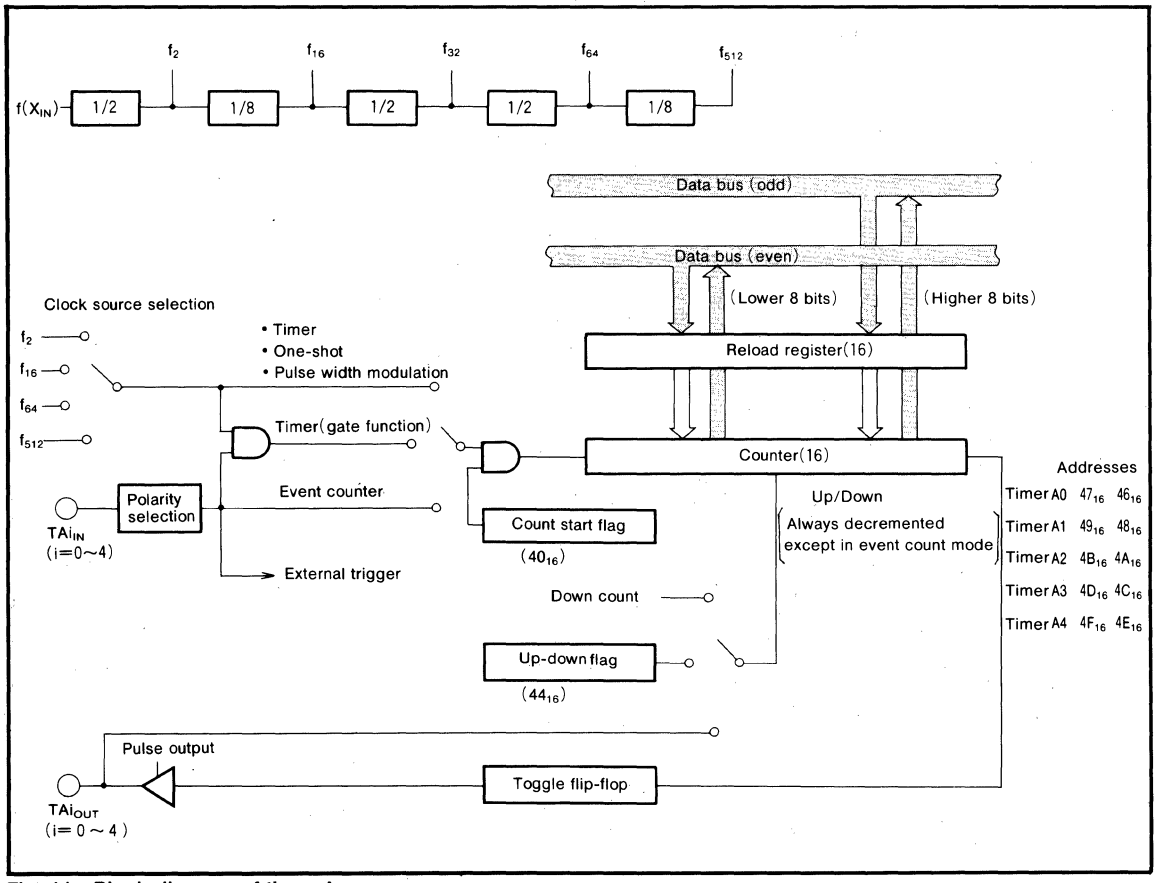


Fig. 11 Block diagram of timer A

When bit 2 of the timer Ai mode register is "1", the output is generated from TAi_{OUT} pin. The output is toggled each time the contents of the counter reaches to 0000_{16}. When the contents of the count start flag is "0", "L" is output from TAi_{OUT} pin.
When bit 2 is "0", TAi_{OUT} can be used as a normal port pin.
When bit 4 is "0", TAi_{IN} can be used as a normal port pin.
When bit 4 is "1", counting is performed only while the input signal from the TAi_{IN} pin is "H" or "L" as shown in Figure 14. Therefore, this can be used to measure the pulse width of the TAi_{IN} input signal. Whether to count while the input signal is "H" or while it is "L" is determined by bit 3. If bit 3 is "1", counting is performed while the TAi_{IN} pin input signal is "H" and if bit 3 is "0", counting is performed while it is "L".
Note that the duration of "H" or "L" on the TAi_{IN} pin must be two or more cycles of the timer count source.
When data is written to timer Ai register with timer Ai halted, the same data is also written to the reload register and the counter. When data is written to timer Ai which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time. The contents of the counter can be read at any time.
When the value set in the timer Ai register is n, the timer frequency dividing ratio is $1/(n+1)$.

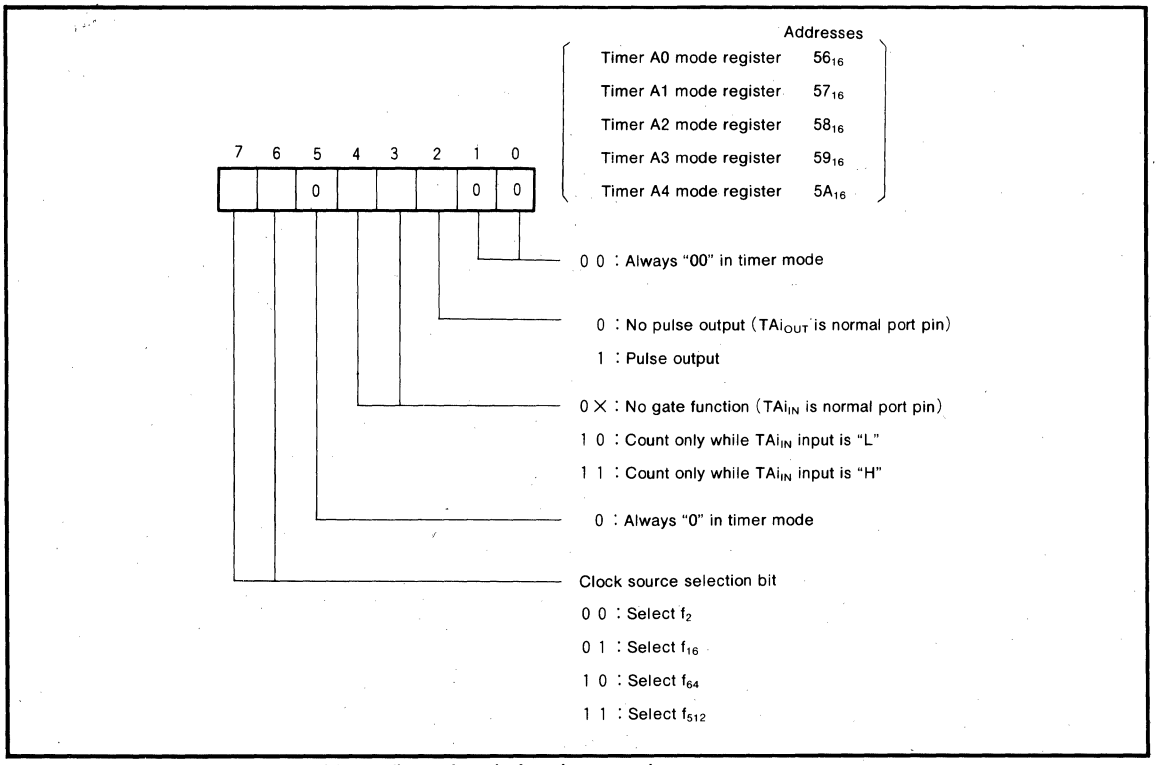


Fig. 12 Timer Ai mode register bit configuration during timer mode

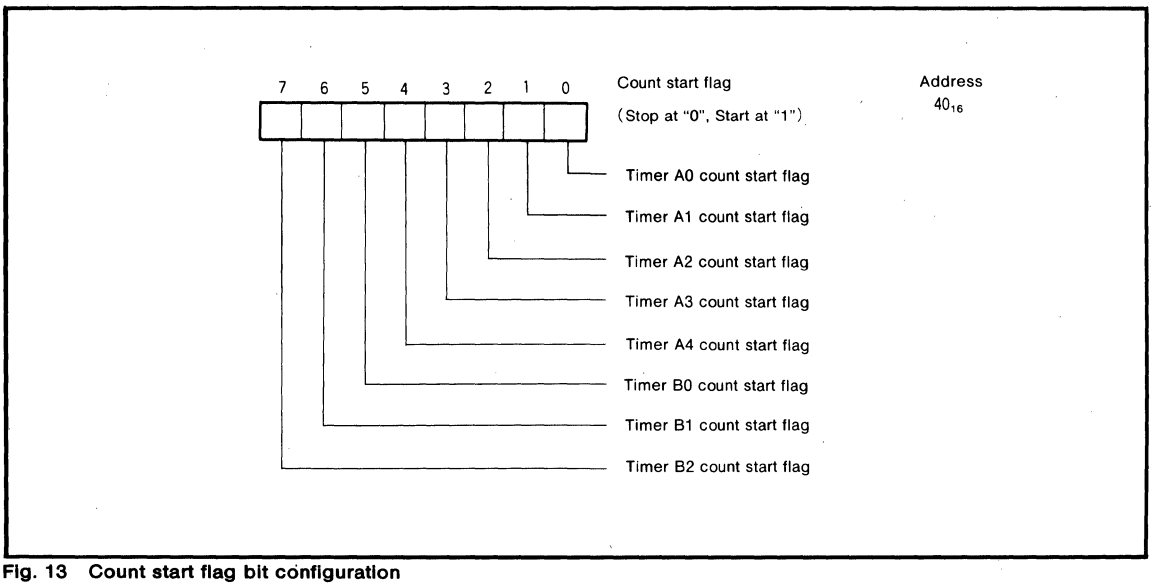


Fig. 13 Count start flag bit configuration

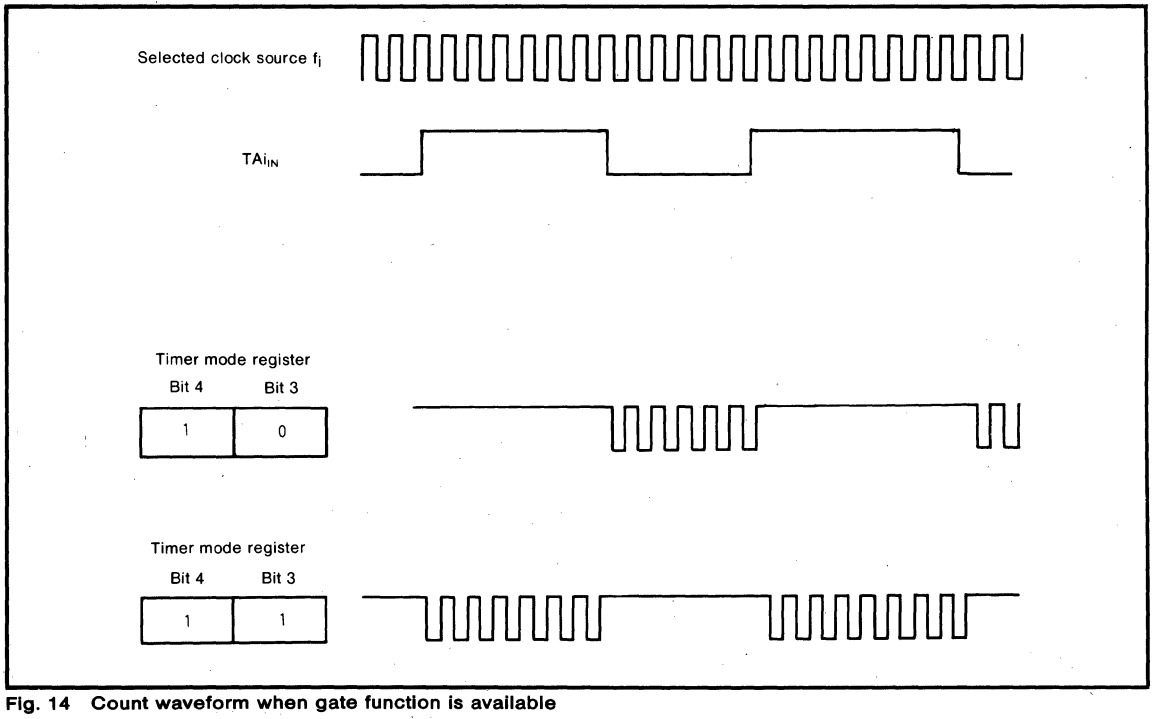


Fig. 14 Count waveform when gate function is available

(2) Event counter mode [01]

Figure 15 shows the bit configuration of the timer Ai mode register during event counter mode. In event counter mode, the bit 0 of the timer Ai mode register must be "1" and bit 1 and 5 must be "0".

The input signal from the TAi_{IN} pin is counted when the count start flag shown in Figure 13 is "1" and counting is stopped when it is "0".

Count is performed at the fall of the input signal when bit 3 is "0" and at the rise of the signal when it is "1".

In event counter mode, whether to increment or decrement the count can be selected with the up-down flag or the input signal from the TAi_{OUT} pin.

When bit 4 of the timer Ai mode register is "0", the up-down flag is used to determine whether to increment or decrement the count (decrement when the flag is "0" and increment when it is "1"). Figure 16 shows the bit configuration of the up-down flag.

When bit 4 of the timer Ai mode register is "1", the input signal from the TAi_{OUT} pin is used to determine whether to increment or decrement the count. However, note that bit 2 must be "0" if bit 4 is "1" because if bit 2 is "1", TAi_{OUT} pin becomes an output pin with pulse output.

The count is decremented when the input signal from the TAi_{OUT} pin is "L" and incremented when it is "H". Determine the level of the input signal from the TAi_{OUT} pin before valid edge is input to the TAi_{IN} pin.

An interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set when the counter reaches 0000_{16} (decrement count) or $FFFF_{16}$ (increment count). At the same time, the contents of the reload register is transferred to the counter and the count is continued.

When bit 2 is "1" and the counter reaches 0000_{16} (decrement count) or $FFFF_{16}$ (increment count), the waveform reversing polarity is output from TAi_{OUT} pin.

If bit 2 is "0", TAi_{OUT} pin can be used as a normal port pin. However, if bit 4 is "1" and the TAi_{OUT} pin is used as an output pin, the output from the pin changes the count direction. Therefore, bit 4 should be "0" unless the output from the TAi_{OUT} pin is to be used to select the count direction.

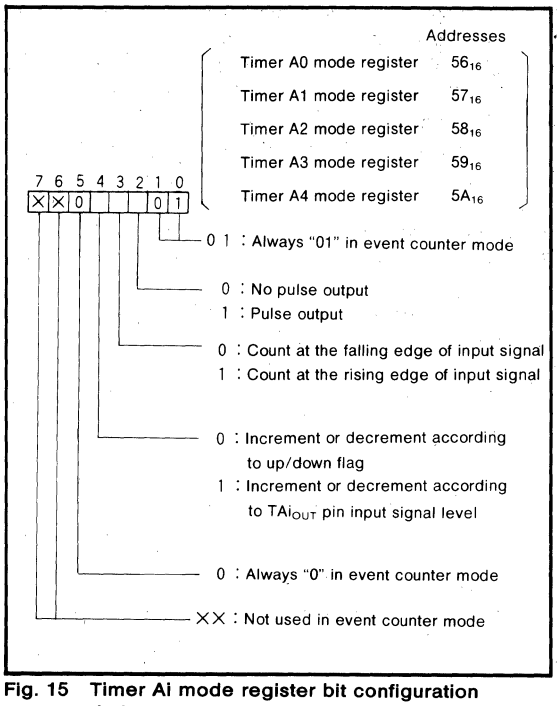


Fig. 15 Timer Ai mode register bit configuration during event counter mode

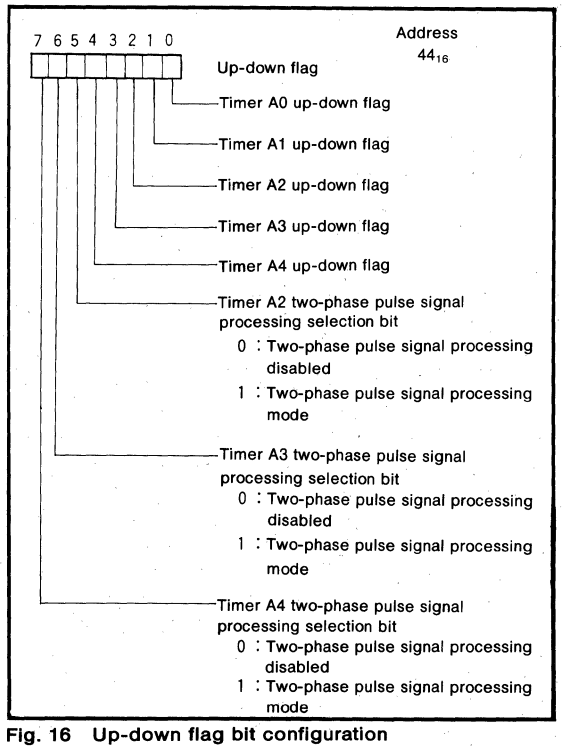


Fig. 16 Up-down flag bit configuration

Data write and data read are performed in the same way as for timer mode. That is, when data is written to timer Ai halted, it is also written to the reload register and the counter. When data is written to timer Ai which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time. The counter can be read at any time.

In event counter mode, whether to increment or decrement the counter can also be determined by supplying two-phase pulse input with phase shifted by 90° to timer A2, A3, or A4. There are two types of two-phase pulse processing operations. One uses timers A2 and A3, and the other uses timer A4. In either processing operation, two-phase pulse is input in the same way, that is, pulses out of phase by 90° are input at the TAj_{OUT} (j=2 to 4) pin and TAj_{IN} pin.

When timers A2 and A3 are used, as shown in Figure 17, the count is incremented when a rising edge is input to the TAk_{IN} pin after the level of TAk_{OUT} (k=2, 3) pin changes from "L" to "H", and when the falling edge is inserted, the count is decremented.

For timer A4, as shown in Figure 18, when a phase related pulse with a rising edge input to the $TA4_{IN}$ pin is input after the level of $TA4_{OUT}$ pin changes from "L" to "H", the count is incremented at the respective rising edge and falling edge of the $TA4_{OUT}$ pin and $TA4_{IN}$ pin.

When a phase related pulse with a falling edge input to the $TA4_{OUT}$ pin is input after the level of $TA4_{IN}$ pin changes from "H" to "L", the count is decremented at the respective rising edge and falling edge of the $TA4_{IN}$ pin and $TA4_{OUT}$ pin. When performing this two-phase pulse signal processing, timer Aj mode register bit 0 and bit 4 must be set to "1" and bits 1, 2, 3, and 5 must be "0". Bits 6 and 7 are ignored. Note that bits 5, 6, and 7 of the up-down flag register (44_{16}) are the two-phase pulse signal processing selection bit for timer A2, A3, and A4 respectively. Each timer operates in normal event counter mode when the corresponding bit is "0" and performs two-phase pulse signal processing when it is "1".

Count is started by setting the count start flag to "1". Data write and read are performed in the same way as for normal event counter mode. Note that the direction register of the input port must be set to input mode because two-phase pulse signal is input. Also, there can be no pulse output in this mode.

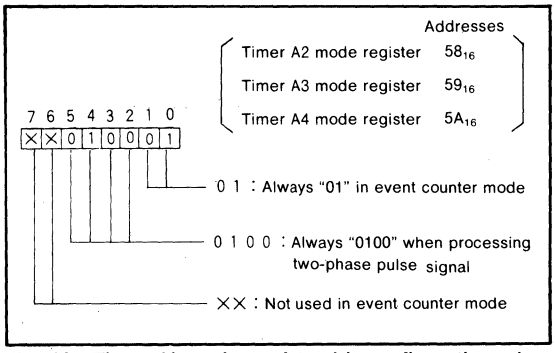


Fig. 19 Timer Aj mode register bit configuration when performing two-phase pulse signal processing in event counter mode

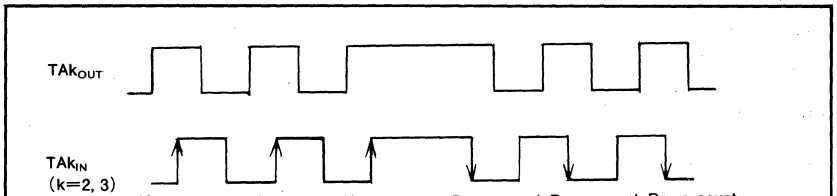


Fig. 17 Two-phase pulse processing operation of timer A2 and timer A3

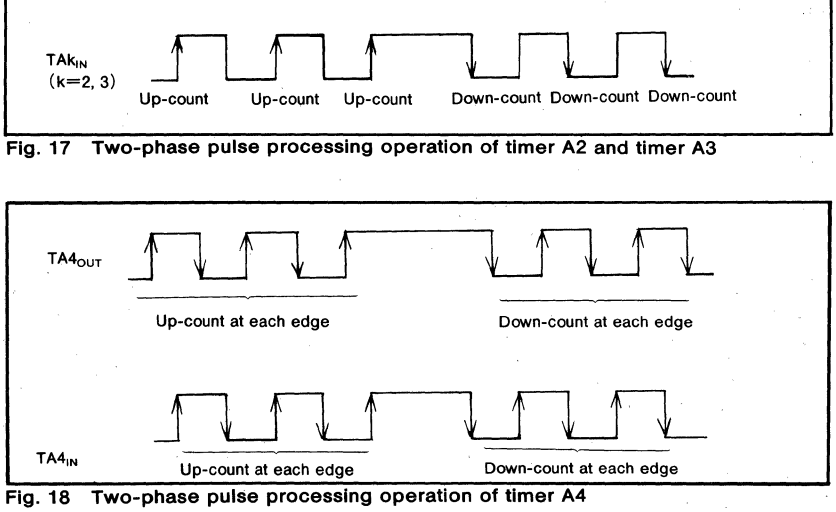


Fig. 18 Two-phase pulse processing operation of timer A4

(3) One-shot pulse mode (10)

Figure 20 shows the bit configuration of the timer Ai mode register during one-shot pulse mode. In one-shot pulse mode, bit 0 and bit 5 must be "0" and bit 1 and bit 2 must be "1".

The trigger is enabled when the count start flag is "1". The trigger can be generated by software or it can be input from the TAi_{IN} pin. Software trigger is selected when bit 4 is "0" and the input signal from the TAi_{IN} pin is used as the trigger when it is "1".

Bit 3 is used to determine whether to trigger at the fall of the trigger signal or at the rise. The trigger is at the fall of the trigger signal when bit 3 is "0" and at the rise of the trigger signal when it is "1".

Software trigger is generated by setting the bit in the one-shot start flag corresponding to each timer.

Figure 21 shows the bit configuration of the one-shot start flag.

As shown in Figure 22, when a trigger signal is received, the counter counts the clock selected by bits 6 and 7.

If the contents of the counter is not 0000_{16}, the TAi_{OUT} pin goes "H" when a trigger signal is received. The count direction is decrement.

When the counter reaches 0001_{16}, The TAi_{OUT} pin goes "L" and count is stopped. The contents of the reload register is transferred to the counter. At the same time, an interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set. This is repeated each time a trigger signal is received. The output pulse width is

$$\frac{1}{\text{pulse frequency of the selected clock}} \times (\text{counter's value at the time of trigger}).$$

If the count start flag is "0", TAi_{OUT} goes "L". Therefore, the value corresponding to the desired pulse width must be written to timer Ai before setting the timer Ai count start flag.

As shown in Figure 23, a trigger signal can be received before the operation for the previous trigger signal is completed. In this case, the contents of the reload register is transferred to the counter by the trigger and then that value is decremented.

Except when retriggering while operating, the contents of the reload register is not transferred to the counter by triggering.

When retriggering, there must be at least one timer count source cycle before a new trigger can be issued.

Data write is performed to the same way as for timer mode. When data is written in timer Ai halted, it is also written to the reload register and the counter.

When data is written to timer Ai which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time.

Undefined data is read when timer Ai is read.

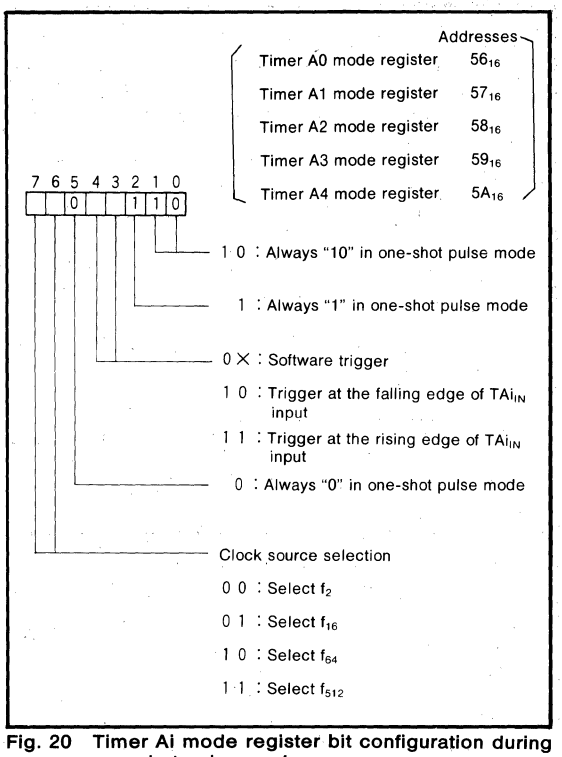


Fig. 20 Timer Ai mode register bit configuration during one-shot pulse mode

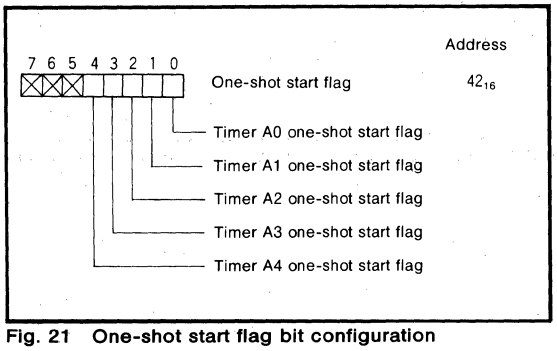


Fig. 21 One-shot start flag bit configuration

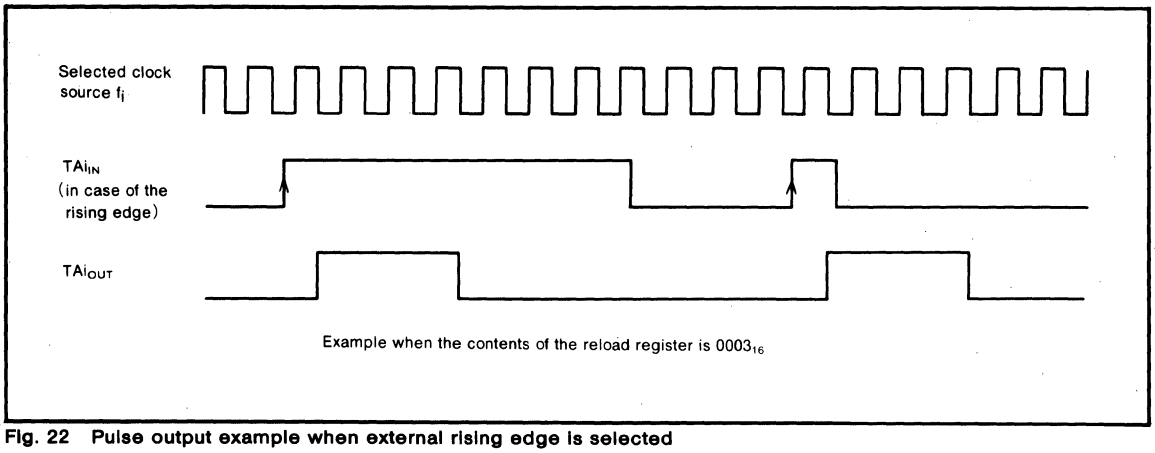


Fig. 22 Pulse output example when external rising edge is selected

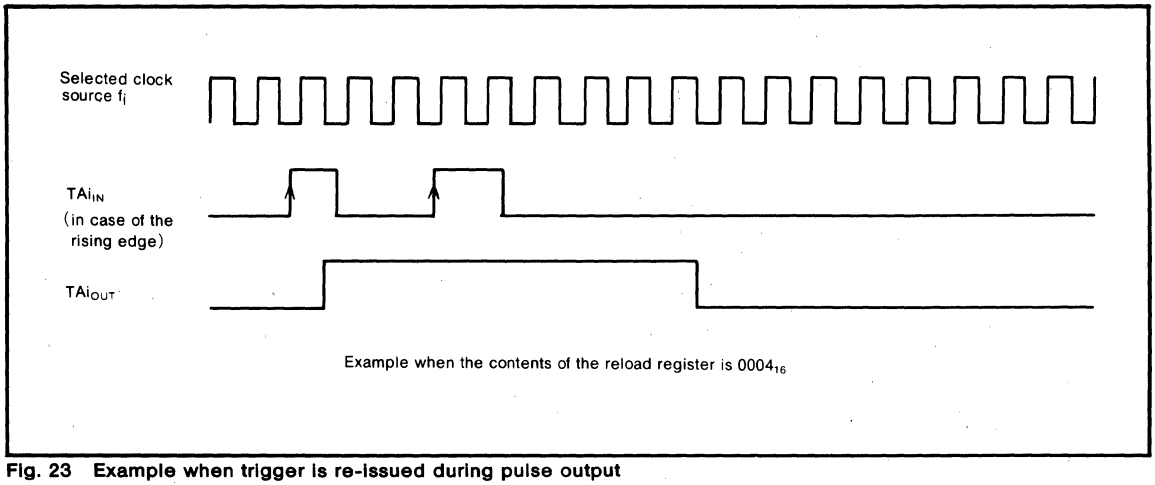


Fig. 23 Example when trigger is re-issued during pulse output

(4) Pulse width modulation mode [11]

Figure 24 shows the bit configuration of the timer Ai mode register during pulse width modulation mode. In pulse width modulation mode, bits 0, 1, and 2 must be set to "1".

Bit 5 is used to determine whether to perform 16-bit length pulse width modulator or 8-bit length pulse width modulator. 16-bit length pulse width modulator is performed when bit 5 is "0" and 8-bit length pulse width modulator is performed when it is "1". The 16-bit length pulse width modulator is described first.

The pulse width modulator can be started with a software trigger or with an input signal from a TAi_{IN} pin (external trigger).

The software trigger mode is selected when bit 4 is "0". Pulse width modulator is started and pulse is output from TAi_{OUT} when the timer Ai start flag is set to "1".

The external trigger mode is selected when bit 4 is "1". Pulse width modulator starts when a trigger signal is input from the TAi_{IN} pin when the timer Ai start flag is "1". Whether to trigger at the fall or rise of the trigger signal is determined by bit 3. The trigger is at the fall of the trigger signal when bit 3 is "0" and at the rise when it is "1".

When data is written to timer Ai with the pulse width modulator halted, it is written to the reload register and the counter.

Then when the time Ai start flag is set to "1" and a software trigger or an external trigger is issued to start modulation, the waveform shown in Figure 25 is output continuously. Once modulation is started, triggers are not accepted. If the value in the reload register is m, the duration "H" of pulse is

$$\frac{1}{\text{selected clock frequency}} \times m$$

and the output pulse period is

$$\frac{1}{\text{selected clock frequency}} \times (2^{16}-1).$$

An interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set at each fall of the output pulse.

The width of the output pulse is changed by updating timer data. The update can be performed at any time. The output pulse width is changed at the rise of the pulse after data is written to the timer.

The contents of the reload register are transferred to the counter just before the rise of the next pulse so that the pulse width is changed from the next output pulse.

Undefined data is read when timer Ai is read.

The 8-bit length pulse width modulator is described next.

The 8-bit length pulse width modulator is selected when the timer Ai mode register bit 5 is "1".

The reload register and the counter are both divided into 8-bit halves.

The low order 8 bits function as a prescaler and the high order 8 bits function as the 8-bit length pulse width modulator. The prescaler counts the clock selected by bits 6 and 7. A pulse is generated when the counter reaches 0000_{16} as shown in Figure 26. At the same time, the contents of the reload register is transferred to the counter and count is continued.

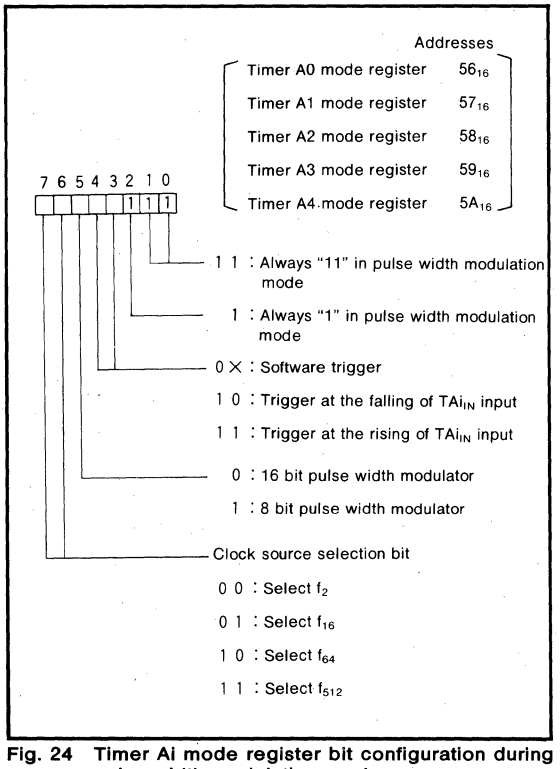


Fig. 24 Timer Ai mode register bit configuration during pulse width modulation mode

Therefore, if the low order 8-bit of the reload register is n, the period of the generated pulse is

$$\frac{1}{\text{selected clock frequency}} \times (n+1).$$

The high order 8-bit function as an 8-bit length pulse width modulator using this pulse as input. The operation is the same as for 16-bit length pulse width modulator except that the length is 8 bits. If the high order 8-bit of the reload register is m, the duration "H" of pulse is

$$\frac{1}{\text{selected clock frequency}} \times (n+1) \times m.$$

And the output pulse period is

$$\frac{1}{\text{selected clock frequency}} \times (n+1) \times (2^8-1).$$

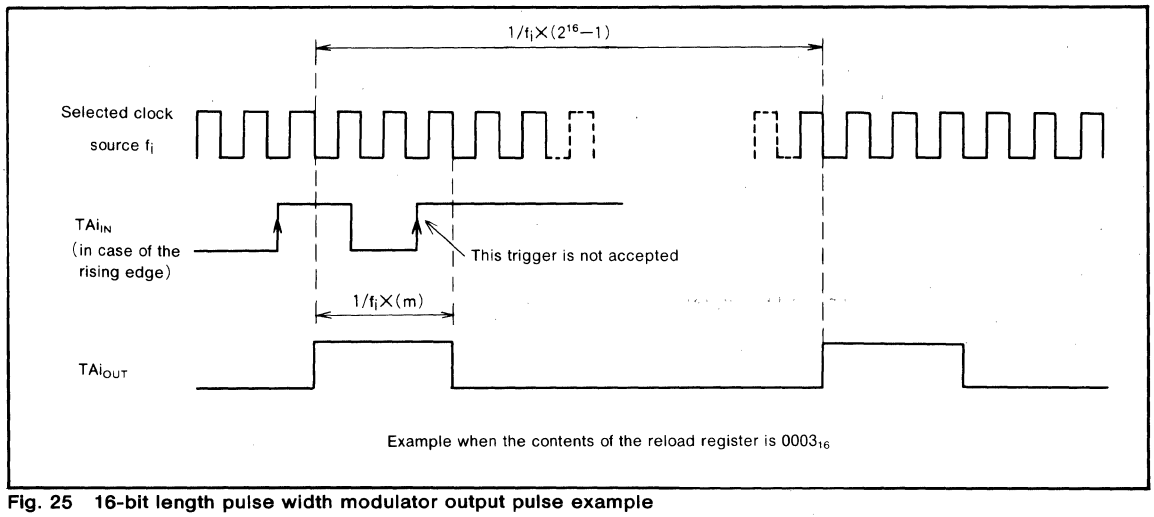


Fig. 25 16-bit length pulse width modulator output pulse example

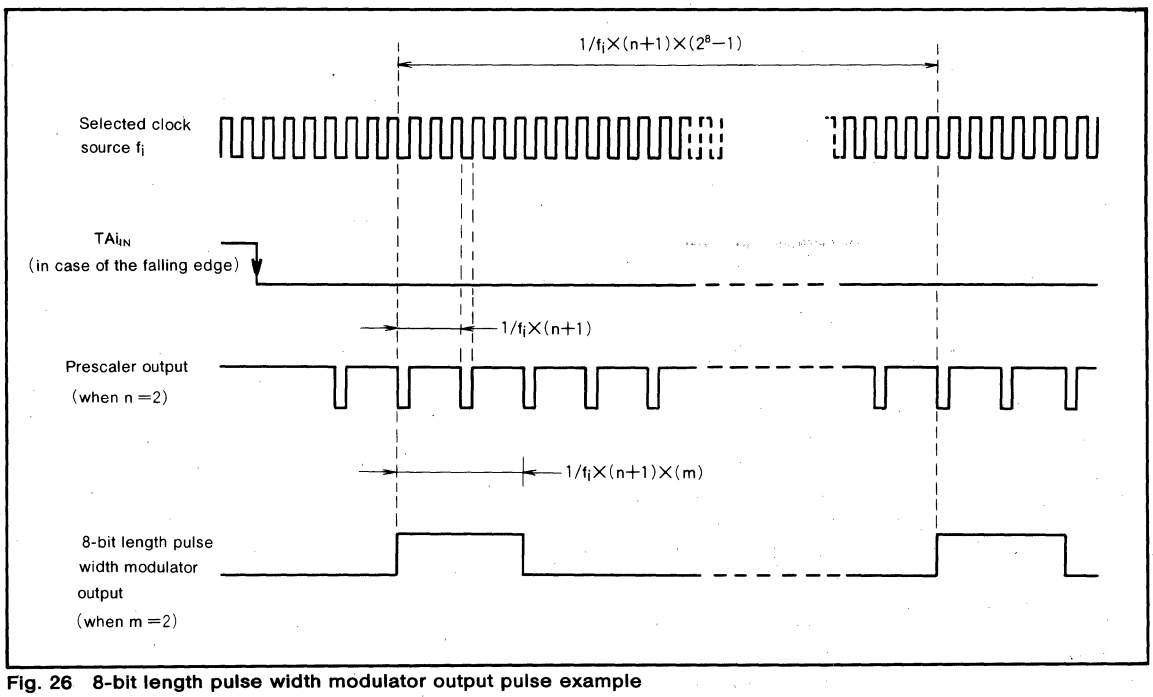


Fig. 26 8-bit length pulse width modulator output pulse example

TIMER B

Figure 27 shows a block diagram of timer B.

Timer B has three modes; timer mode, event counter mode, and pulse period measurement/pulse width measurement mode. The mode is selected with bits 0 and 1 of the timer Bi mode register (i = 0 to 2). Each of these modes is described below.

(1) Timer mode [00]

Figure 28 shows the bit configuration of the timer Bi mode register during timer mode. Bits 0, and 1 of the timer Bi mode register must always be "0" in timer mode.

Bits 6 and 7 are used to select the clock source. The counting of the selected clock starts when the count start flag "1" and stops when "0".

As shown in Figure 13, the timer Bi count start flag is at the same address as the timer Ai count start flag. The count is decremented, an interrupt occurs, and the interrupt request bit in the timer Bi interrupt control register is set when the contents becomes 0000_{16}. At the same time, the contents of the reload register is stored in the counter and count is continued.

Timer Bi does not have a pulse output function or a gate function like timer A.

When data is written to timer Bi halted, it is written to the reload register and the counter. When data is written to timer Bi which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time. The contents of the counter can be read at any time.

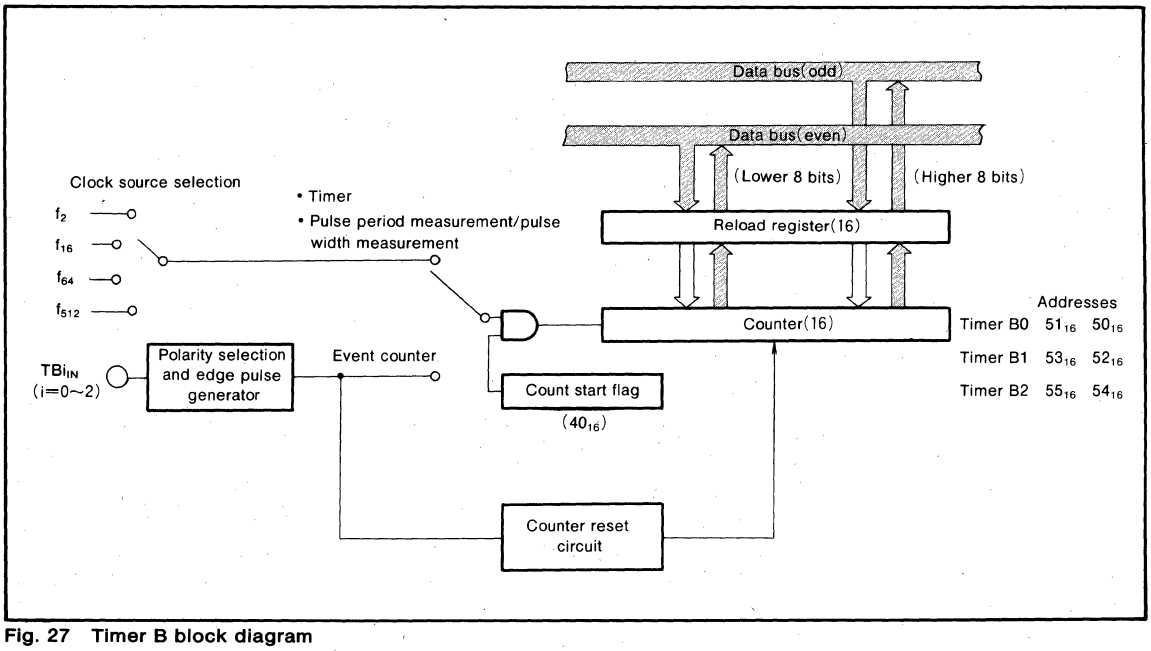


Fig. 27 Timer B block diagram

(2) Event counter mode [01]

Figure 29 shows the bit configuration of the timer Bi mode register during event counter mode. In event counter mode, the bit 0 in the timer Bi mode register must be "1" and bit 1 must be "0".

The input signal from the TBi_{IN} pin is counted when the count start flag is "1" and counting is stopped when it is "0". Count is performed at the fall of the input signal when bits 2, and 3 are "0" and at the rise of the input signal when bit 3 is "0" and bit 2 is "1".

When bit 3 is "1" and bit 2 is "0", count is performed at the rise and fall of the input signal.

Data write, data read and timer interrupt are performed in the same way as for timer mode.

(3) Pulse period measurement/pulse width measurement mode [10]

Figure 30 shows the bit configuration of the timer Bi mode register during pulse period measurement/pulse width measurement mode.

In pulse period measurement/pulse width measurement mode, bit 0 must be "0" and bit 1 must be "1". Bits 6 and 7 are used to select the clock source. The selected clock is counted when the count start flag is "1" and counting stops when it is "0".

The pulse period measurement mode is selected when bit 3 is "0". In pulse period measurement mode, the selected clock is counted during the interval starting at the fall of the input signal from the TBi_{IN} pin to the next fall or at the rise of the input signal to the next rise and the result is stored in the reload register. In this case, the reload register acts as a buffer register.

When bit 2 is "0", the clock is counted from the fall of the input signal to the next fall. When bit 2 is "1", the clock is counted from the rise of the input signal to the next rise.

In the case of counting from the fall of the input signal to the next fall, counting is performed as follows. As shown in Figure 31, when the fall of the input signal from TBi_{IN} pin is detected, the contents of the counter is transferred to the reload register. Next the counter is cleared and count is started from the next clock. When the fall of the next input signal is detected, the contents of the counter is transferred to the reload register once more, the counter is cleared, and the count is started. The period from the fall of the input signal to the next fall is measured in this way.

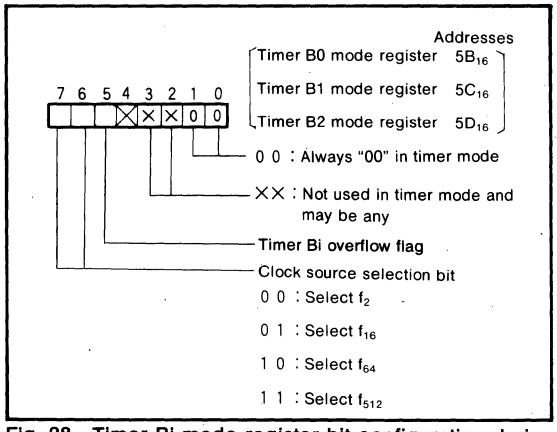


Fig. 28 Timer Bi mode register bit configuration during timer mode

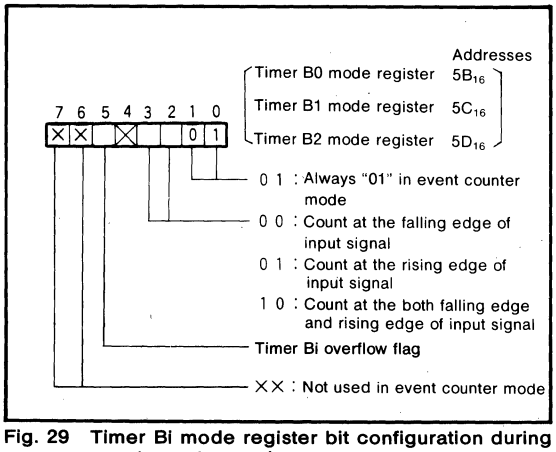


Fig. 29 Timer Bi mode register bit configuration during event counter mode

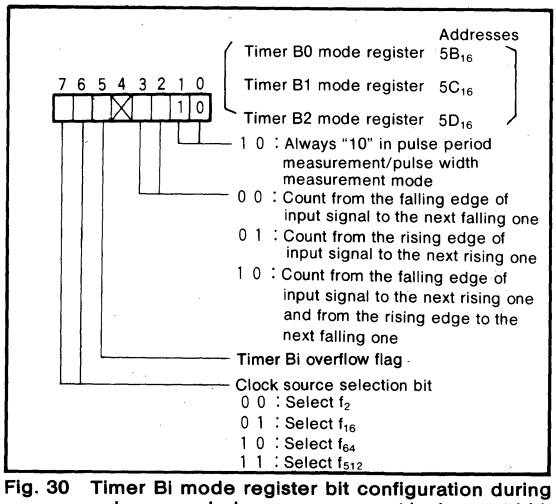


Fig. 30 Timer Bi mode register bit configuration during pulse period measurement/pulse width measurement mode

After the contents of the counter is transferred to the reload register, an interrupt request signal is generated and the interrupt request bit in the timer Bi interrupt control register is set. However, no interrupt request signal is generated when the contents of the counter is transferred first time to the reload register after the count start flag is set to "1".

When bit 3 is "1", the pulse width measurement mode is selected. Pulse width measurement mode is similar to pulse period measurement mode except that the clock is counted from the fall of the TBi_{IN} pin input signal to the next rise or from the rise of the input signal to the next fall as shown in Figure 32.

When timer Bi is read, the contents of the reload register is read.

Note that in this mode, the interval between the fall of the TBi_{IN} pin input signal to the next rise or from the rise to the next fall must be at least two cycles of the timer count source.

Timer Bi overflow flag which is bit 5 of timer Bi mode register is set to "1" when the timer Bi counter reaches 0000_{16}. This flag is cleared by writing to corresponding timer Bi mode register. This bit is set to "1" at reset.

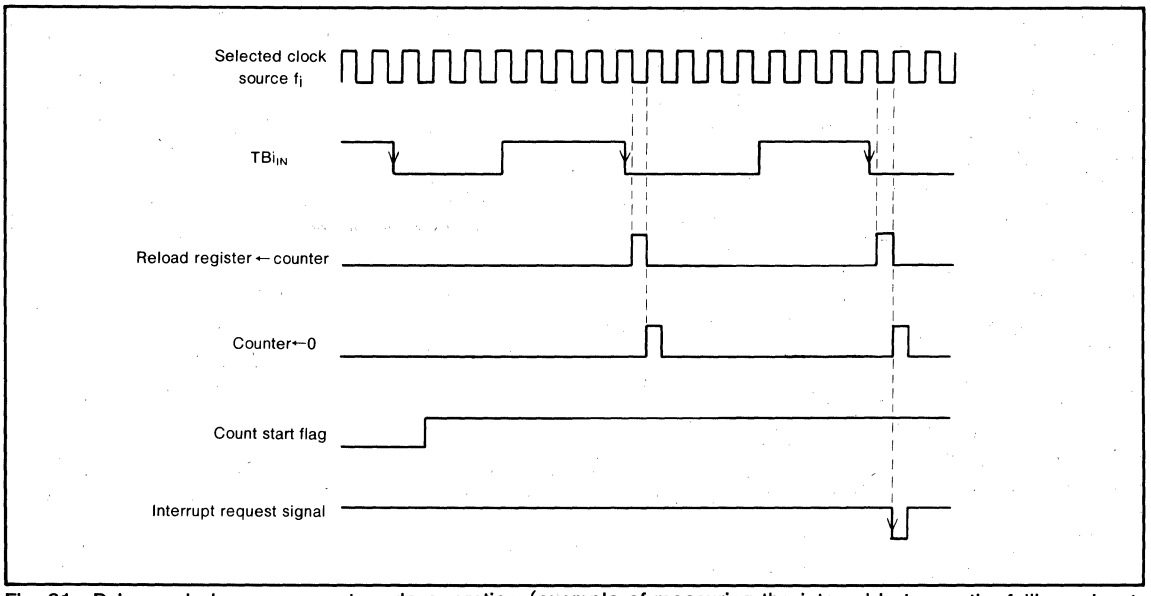


Fig. 31 Pulse period measurement mode operation (example of measuring the interval between the falling edge to next falling one)

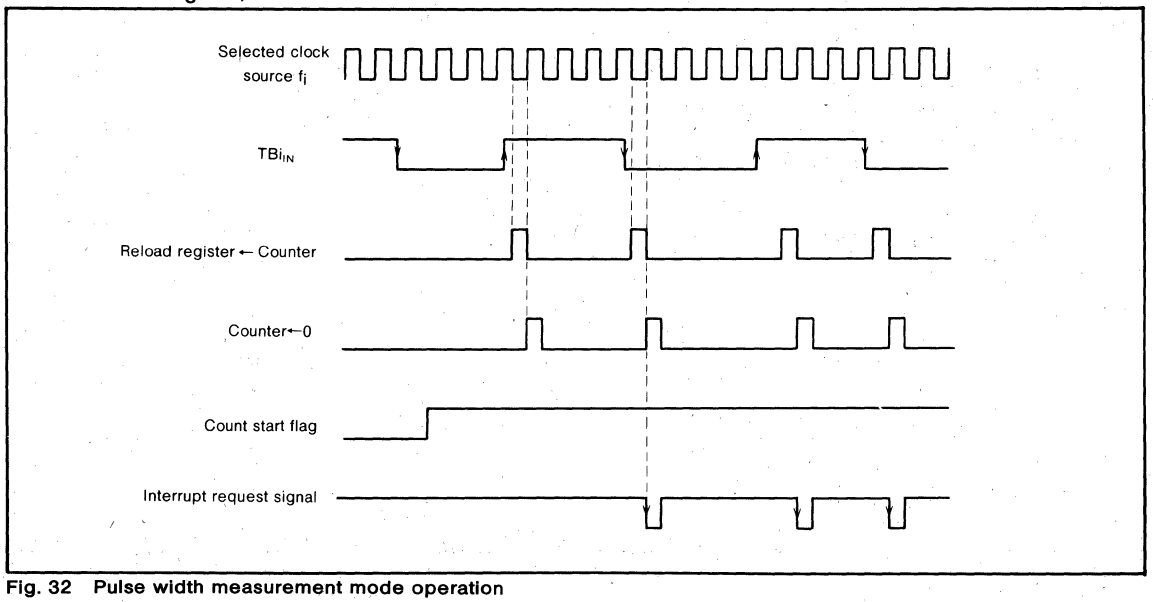


Fig. 32 Pulse width measurement mode operation

Pulse output port mode

Figure 33 shows a block diagram for pulse output port mode. In the pulse output port mode, two pairs of four-bit pulse output ports are used. Whether using pulse output port or not can be selected by waveform output selection bit (bit 0, bit 1) of waveform output mode register (62_{16} address) shown in Figure 34. When bit 0 of waveform output selection bit is set to "1", ports $P5_7$, $P5_6$, $P5_5$ and $P5_4$ are used as pulse output ports (RTP1 selected), and when bit 1 of waveform output selection bit is set to "1", ports $P5_3$, $P5_2$, $P5_1$, and $P5_0$ are used as pulse output ports (RTP0 selected). When bits 1 and 0 of waveform output selection bit are set to "1", ports $P5_7$, $P5_6$, $P5_5$, and $P5_4$, and ports $P5_3$, $P5_2$, $P5_1$ and $P5_0$ are used as pulse output ports (RTP1 and RTP0 selected).

The ports not used as pulse output ports can be used as normal parallel ports or timer input/output.

In the pulse output port mode, set timers A2 and A0 to timer mode as timers A2 and A0 are used. Figure 35 shows the bit configuration of timer A0, A2 mode registers in pulse output port mode.

Data can be set in each bit of the pulse output data register corresponding to four ports selected as pulse output ports. Figure 36 shows the bit configuration of the pulse output data register. The contents of the pulse output data register 1 (low-order four bits of 64_{16} address) corresponding to ports $P5_7$, $P5_6$, $P5_5$ and $P5_4$ is output to the ports each time the counter of timer A2 becomes 0000_{16}. The contents of the pulse output data register 0 (low-order four bits of 65_{16} address) corresponding to ports $P5_3$, $P5_2$, $P5_1$, and $P5_0$ is output to the ports each time the counter of timer A0 becomes 0000_{16}.

When "0" is written to a specified bit of the pulse output data register, "L" level is output to the corresponding pulse output port when the counter of corresponding timer becomes 0000_{16}, and when "1" is written, "H" level is output to the pulse output port.

Pulse width modulation can be applied to each pulse output port. Since pulse width modulation involves the use of timers A3 and A1, activate these timers in pulse width modulation mode. When a certain bit of the pulse output register is "1", pulse width modulation is output from the pulse output port when the counter of the corresponding timer becomes 0000_{16}.

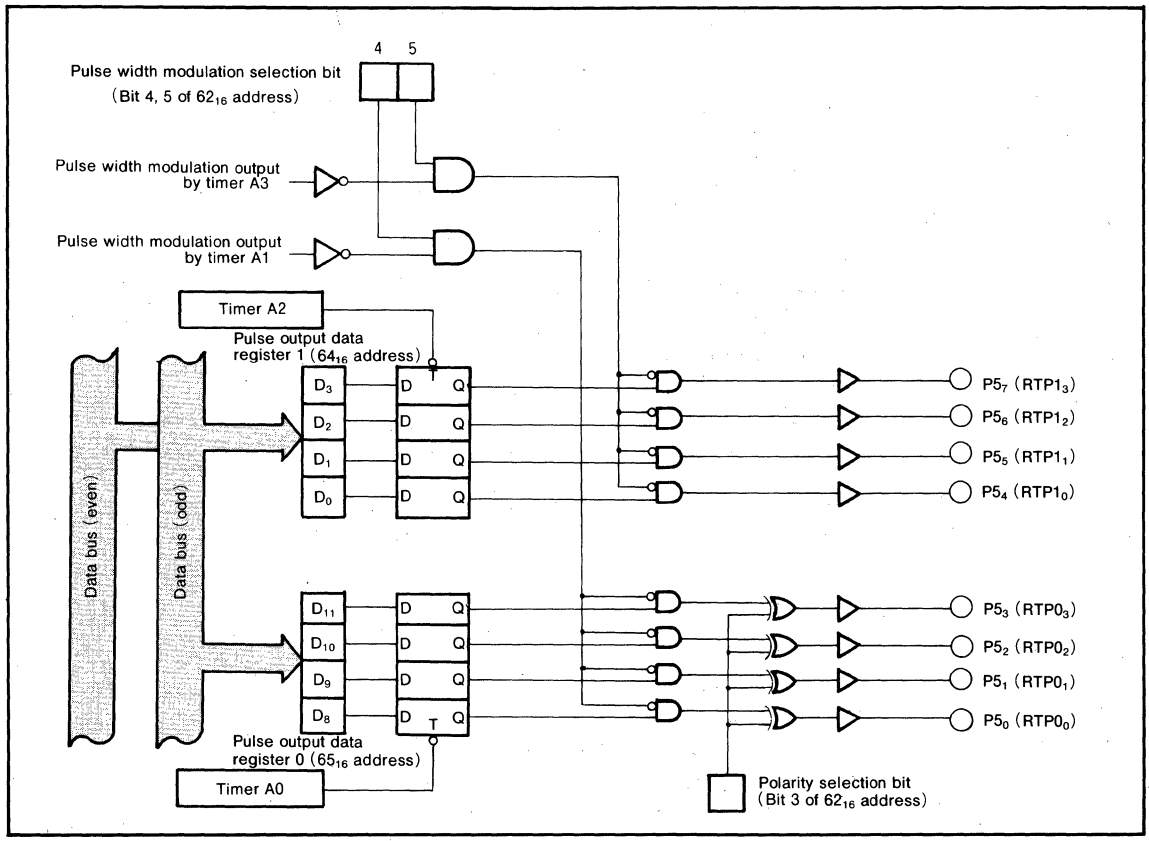


Fig. 33 Block diagram for pulse output port mode

Ports $P5_7$, $P5_6$, $P5_5$ and $P5_4$ are applied pulse width modulation by timer A3 by setting the pulse width modulation selection bit by timer A3 (bit 5) of the waveform output mode register to "1".

Ports $P5_3$, $P5_2$, $P5_1$ and $P5_0$ are applied pulse width modulation by timer A1 by setting the pulse width modulation selection bit by timer A1 (bit 4) of the waveform output mode register to "1".

The contents of the pulse output data register 0 can be reversed and output to pulse output ports $P5_3$, $P5_2$, $P5_1$ and $P5_0$ by the polarity selection bit (bit 3) of the waveform output mode register. When the polarity selection bit is "0", the contents of the pulse output data register 0 is output unchangeably, and when "1", the contents of the pulse output data register 0 is reversed and output. When pulse width modulation is applied, likewise the polarity reverse to pulse width modulation can be selected by the polarity selection bit.

Figure 37 shows example of waveforms in pulse output port mode.

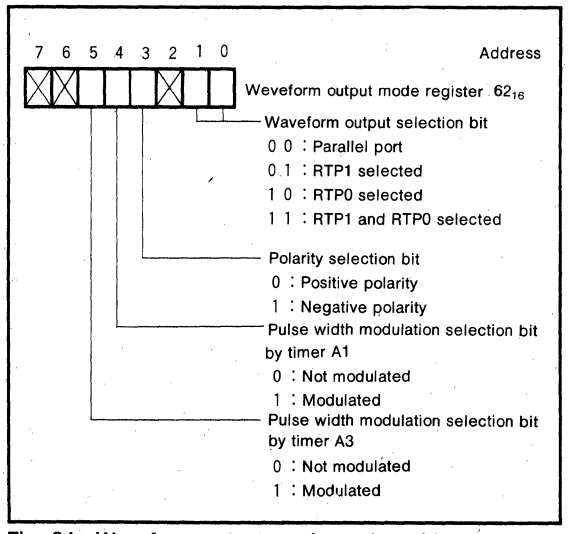


Fig. 34 Waveform output mode register bit configuration

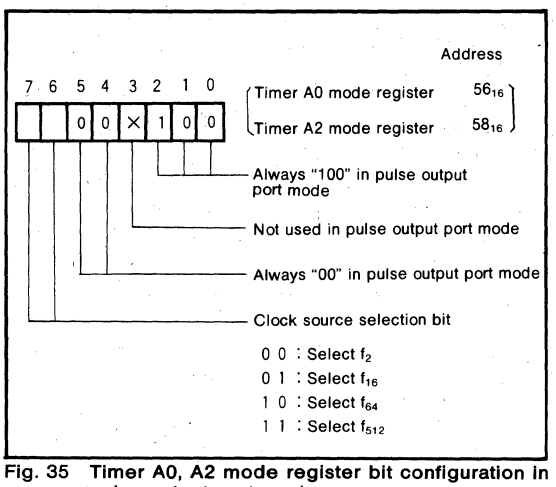


Fig. 35 Timer A0, A2 mode register bit configuration in pulse output port mode

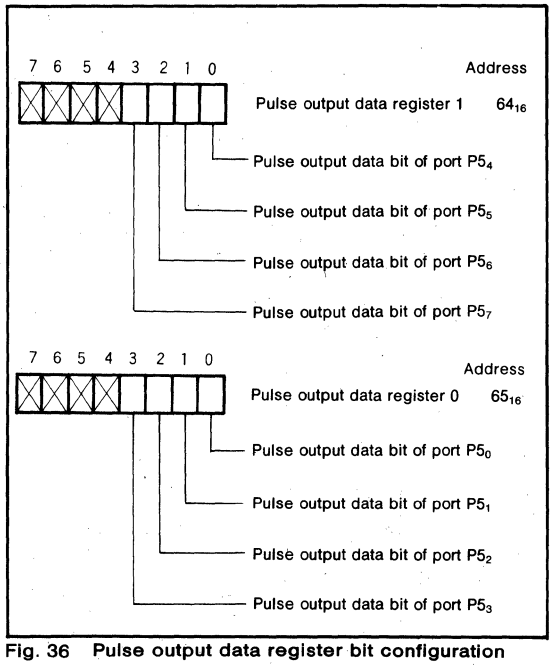


Fig. 36 Pulse output data register bit configuration

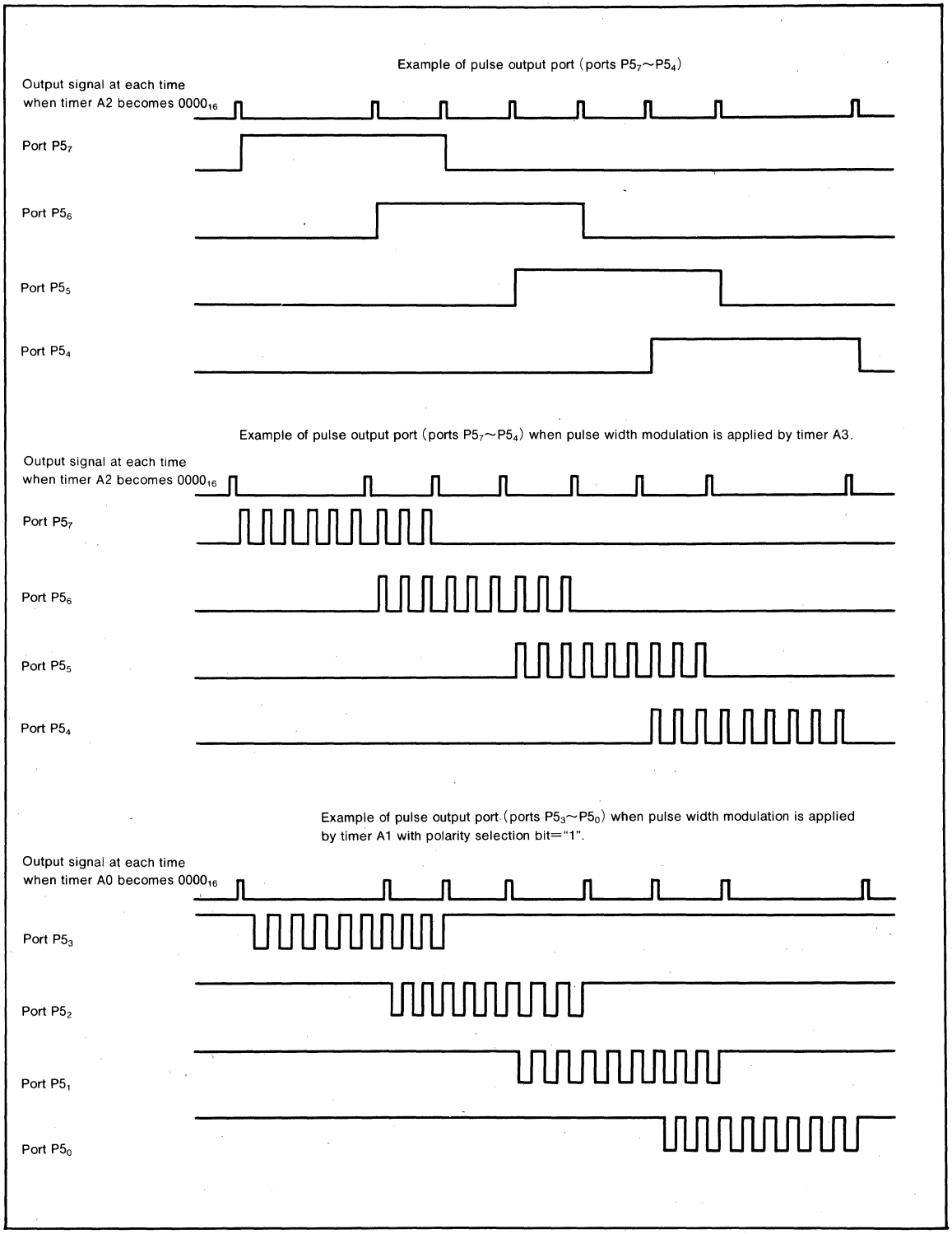


Fig. 37 Example of waveforms in pulse output port mode

SERIAL I/O PORTS

Two independent serial I/O ports are provided. Figure 38 shows a block diagram of the serial I/O ports.

Bits 0, 1, and 2 of the UARTi (i = 0, 1) Transmit/Receive mode register shown in Figure 39 are used to determine whether to use port P8 as parallel port, clock synchronous serial I/O port, or asynchronous (UART) serial I/O port using start and stop bits.

Figures 40 and 41 show the connections of receiver/transmitter according to the mode.

Figure 42 shows the bit configuration of the UARTi transmit/receive control register.

Each communication method is described below.

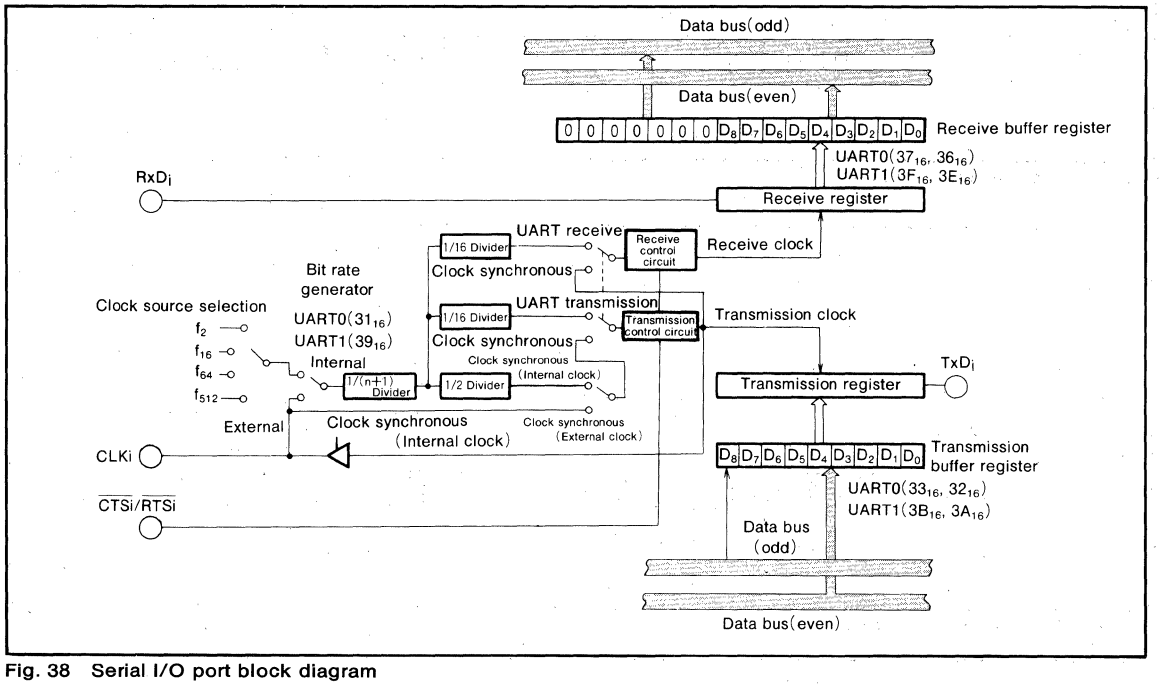


Fig. 38 Serial I/O port block diagram

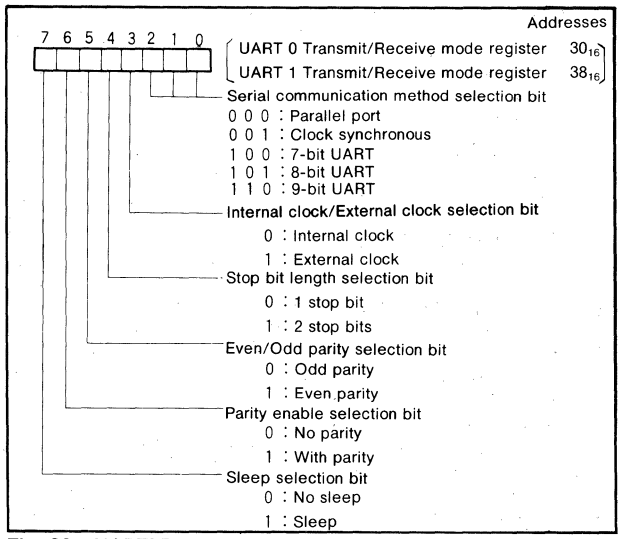


Fig. 39 UARTi Transmit/Receive mode register bit configuration

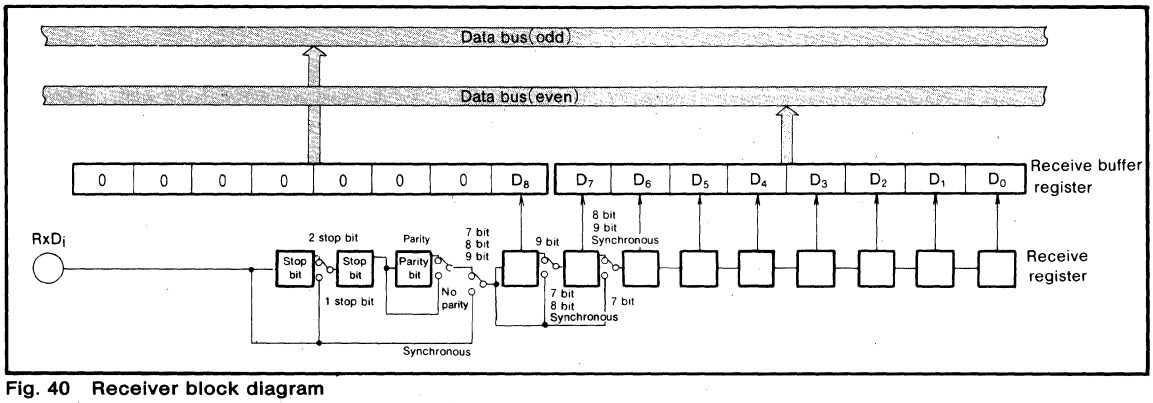


Fig. 40 Receiver block diagram

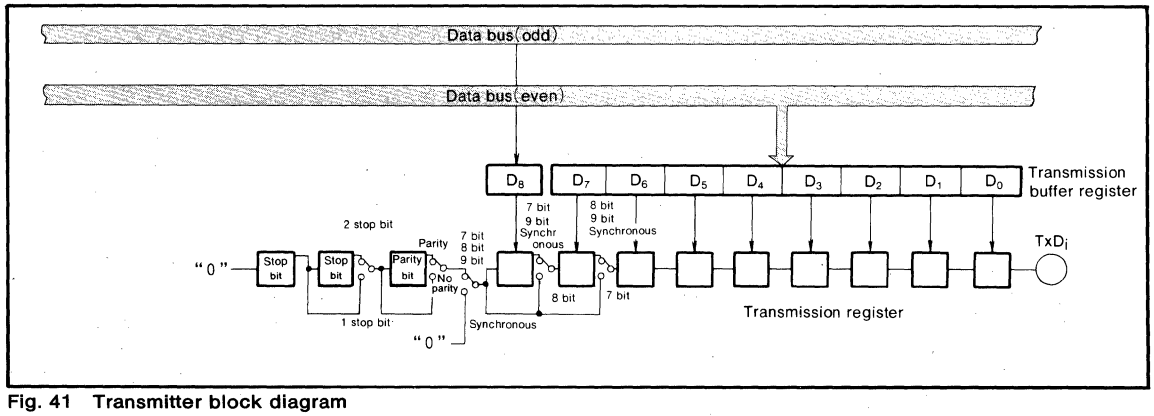


Fig. 41 Transmitter block diagram

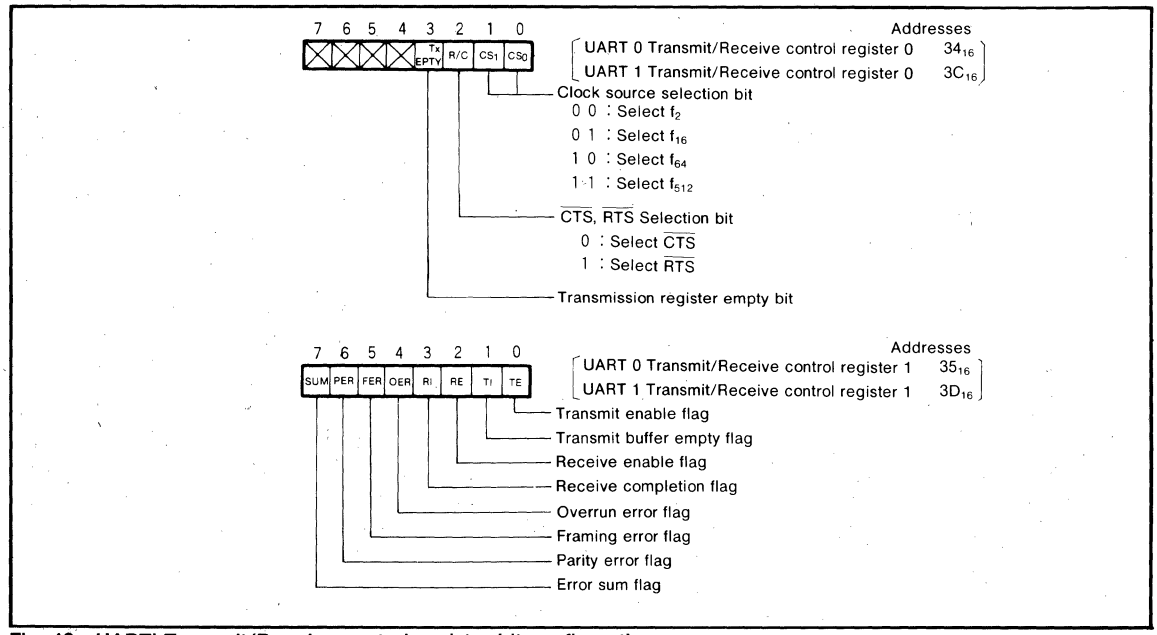


Fig. 42 UARTi Transmit/Receive control register bit configuration

CLOCK SYNCHRONOUS SERIAL COMMUNICATION

A case where communication is performed between two clock synchronous serial I/O ports as shown in Figure 43 will be described. (The transmission side will be denoted by subscript j and the receiving side will be denoted by subscript k.)

Bit 0 of the UARTj transmit/receive mode register and UARTk transmit/receive mode register must be set to "1" and bits 1 and 2 must be "0". The length of the transmission data is fixed at 8 bits.

Bit 3 of the UARTj transmit/receive mode register of the clock sending side is cleared to "0" to select the internal clock. Bit 3 of the UARTk transmit/receive mode register of the clock receiving side is set to "1" to select the external clock. Bits 4, 5 and 6 are ignored in clock synchronous mode. Bit 7 must always be "0".

The clock source is selected by bit 0 (CS_0) and bit 1 (CS_1) of the clock sending side UARTj transmit/receive control register 0. As shown in Figure 38, the selected clock is divided by (n + 1), then by 2, passed through a transmisson control circuit, and output as transmisson clock CLKj. Therefore, when the selected clock is fi,

$$\text{Bit Rate} = fi / \{(n+1) \times 2\}$$

On the clock receiving side, the CS_0 and CS_1 bits of the UARTk transmit/receive control register are ignored because an external clock is selected.

The bit 2 of the clock sending side UARTj transmit/receive control register is clear to "0"to select $\overline{CTS}j$ input. The bit 2 of the clock receiving side is set to "1" to select $\overline{RTS}k$ output. $\overline{CTS}$, and $\overline{RTS}$ signals are described later.

Transmission

Transmission is started when the bit 0 (TEj flag) of UARTj transmit/receive control register 1 is "1", bit 1 is (Tlj flag) of one is "0", and $\overline{CTS}j$ input is "L". As shown in Figure 44, data is output from TxDj pin when transmission clock CLKj changes from "H" to "L". The data is output from the least significant bit.

The Tlj flag indicates whether the transmission buffer register is empty or not. It is cleared to "0" when data is written in the transmission buffer register and set to "1" when the contents of the transmission buffer register is transferred to the transmission register.

When the transmission register becomes empty after the contents has been transmitted, data is transferred automatically from the transmission buffer register to the transmission register if the next transmission start condition is satisfied. If the bit 2 of UARTj transmit/receive control register 0 is "1", $\overline{CTS}j$ input is ignored and transmission start is controlled only by the TEj flag and Tlj flag. Once transmission has started, the TEj flag, Tlj flag, and $\overline{CTS}j$ signals are ignored until data transmission completes. Therefore, transmission is not interrupt when $\overline{CTS}j$ input is changed to "H" during transmission.

The transmission start condition indicated by TEj flag, Tlj flag, and $\overline{CTS}j$ is checked while the T_{ENDj} signal shown in Figure 44 is "H". Therefore, data can be transmitted continuously if the next transmission data is written in the transmission buffer register and Tlj flag is cleared to "0" before the T_{ENDj} signal goes "H".

The bit 3 (TxEPTYj flag) of UARTj transmit/receive control register 0 changes to "1" at the next cycle after the T_{ENDj} signal goes "H" and changes to "0" when transmission starts. Therefore, this flag can be used to determine whether data transmission has completed.

When the Tlj flag changes from "0" to "1", the interrupt request bit in the UARTj transmission interrupt control register is set to "1".

Receive

Receive starts when the bit 2 (RE_K flag) of $UART_K$ transmit/receive control register 1 is set to "1".

The $\overline{RTS}_K$ output is "H" when the RE_K flag is "0" and goes "L" when the RE_K flag changed to "1". It goes back to "H" when receive starts. Therefore, the $\overline{RTS}_K$ output can be used to determine whether the receive register is ready to receive. It is ready when $\overline{RTS}_K$ output is "L".

The data from the RxD_K pin is retrieved and the contents of the receive register is shifted by 1 bit each time the transmission clock CLKj changes from "L" to"H". When an 8-bit data is received, the contents of the receive register is transferred to the receive buffer register and the bit 3 (RI_K flag) of $UART_K$ transmit/receive control register 1 is set to "1". In other words, the setting of the RI_K flag indicates that the receive buffer register contains the received data. At this point, $\overline{RTS}j$ output goes "L" to indicate that the next data can be received. When the RI_K flag changes from "0" to "1", the interrupt request bit in the $UART_K$ receive interrupt control register is set to "1". Bit 4 (OER_K flag) of $UART_K$ transmit/receive control register is set to "1" when the next data is transferred from the receive register to the receive buffer register while RI_K flag is "1", and indicates that the next data was transferred to the receive register before the contents of the receive buffer register was read. RI_K and OER_K flags are cleared automatically to "0" when the low-order byte of the receive buffer register is read. The OER_K flag is also cleared when the RE_K flag is cleared. Bit 5 (FER_K flag), bit 6 (PER_K flag), and bit 7 (SUM_K flag) are ignored in clock synchronous mode.

As shown in Figure 38, with clock synchronous serial communication, data cannot be received unless the transmitter is operating because the receive clock is created from the transmission clock. Therefore, the transmitter must be operating even when there is no data to be sent from $UART_K$ to UARTj.

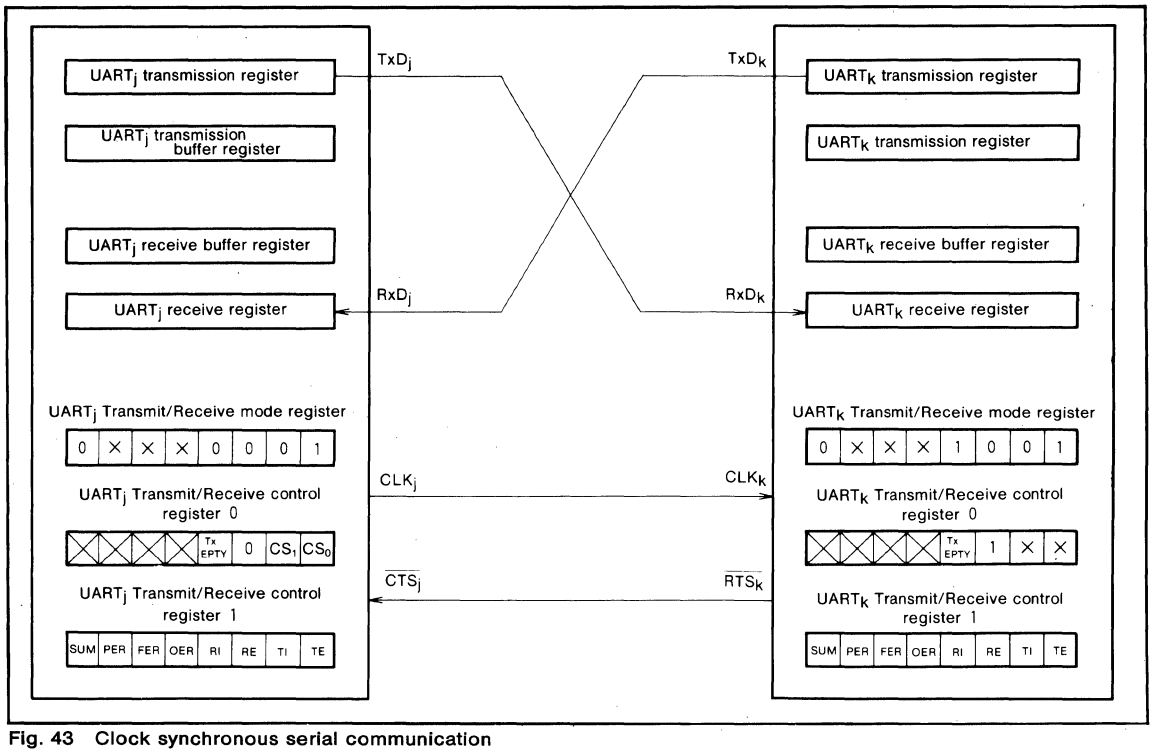


Fig. 43 Clock synchronous serial communication

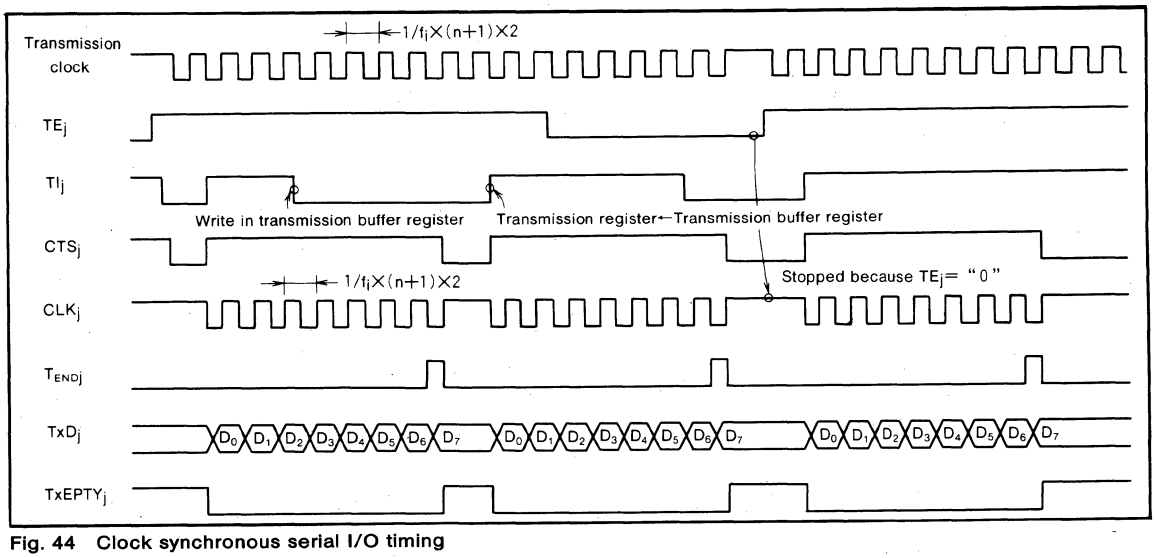


Fig. 44 Clock synchronous serial I/O timing

ASYNCHRONOUS SERIAL COMMUNICATION

Asynchronous serial communication can be performed using 7-, 8-, or 9-bit length data. The operation is the same for all data lengths. The following is the description for 8-bit asynchronous communication.

With 8-bit asynchronous communication, the bit 0 of UARTi transmit/receive mode register is "1", the bit 1 is "0", and the bit 2 is "1".

Bit 3 is used to select an internal clock or an external clock. If bit 3 is "0", an internal clock is selected and if bit 3 is "1", then external clock is selected. If an internal clock is selected, the bit 0 (CS_0) and bit 1 (CS_1) of UARTi transmit/receive control register 0 are used to select the clock source. When an internal clock is selected for asynchronous serial communication, the CLK pin can be used as a normal I/O pin.

The selected internal or external clock is divided by (n+1), then by 16, and passed through a control circuit to create the UART transmission clock or UART receive clock.

Therefore, the transmission speed can be changed by changing the contents n of the bit rate generator. If the selected clock is an internal clock fi or an external clock f_{EXT},

$$\text{Bit Rate} = (f_i \text{ or } f_{EXT}) / \{(n+1) \times 16\}$$

Bit 4 is the stop bit length selection bit to select 1 stop bit or 2 stop bits.

The bit 5 is a selection bit of odd parity or even parity.

In the odd parity mode, the parity bit is adjusted so that the sum of the 1's in the data and parity bit is always odd.

In the even parity mode, the parity bit is adjusted so that the sum of the 1's in the data and parity bit is always even.

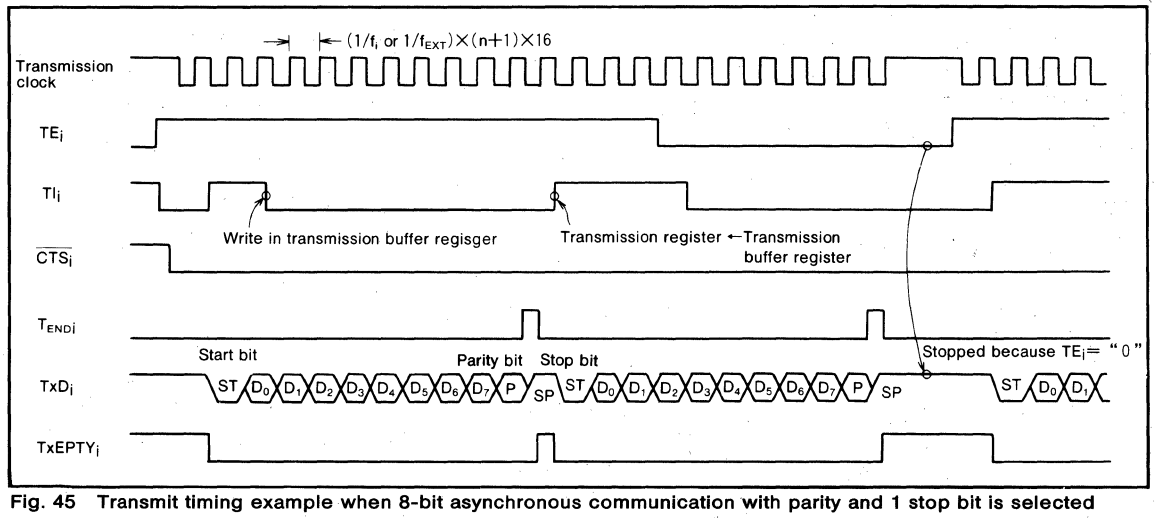


Fig. 45 Transmit timing example when 8-bit asynchronous communication with parity and 1 stop bit is selected

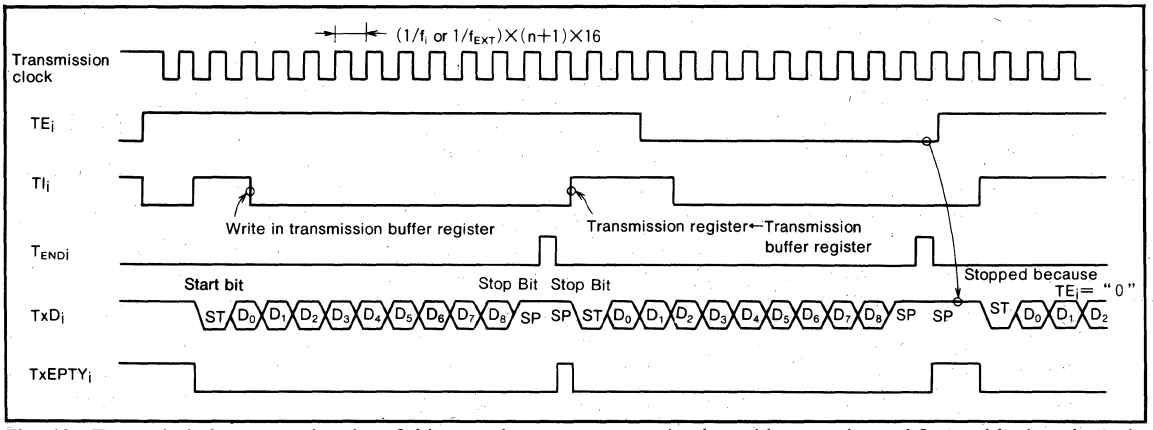


Fig. 46 Transmit timing example when 9-bit asynchronous communication with no parity and 2 stop bits is selected

Bit 6 is the parity bit selection bit which indicates whether to add parity bit or not.
Bits 4 to 6 should be set or reset according to the data format of the communicating devices.
Bit 7 is the sleep selection bit. The sleep mode is described later.
The UART$_i$ transmit/receive control regisger 0 bit 2 is used to determine whether to use $\overline{CTS_i}$ input or $\overline{RTS_i}$ output. $\overline{CTS_i}$ input used if bit 2 is "0" and $\overline{RTS_i}$ output is used if bit 2 is "1".
If $\overline{CTS_i}$ input is selected, the user can control whether to stop or start transmission by external $\overline{CTS_i}$ input. $\overline{RTS_i}$ will be described later.

Transmission

Transmission is started when the bit 0 (TEi flag) of UART$_i$ transmit/receive control register 1 is "1", the bit 1 (TI$_i$ flag) is "0", and $\overline{CTS_i}$ input is "L" if $\overline{CTS_i}$ input is selected. As shown in Figure 45 and 46, data is output from the TxD$_i$ pin with the stop bit and parity bit specified by the bits 4 to 6 of UART$_i$ transmit/receive mode register bits. The data is output from the least significant bit.
The TI$_i$ flag indicates whether the transmission butter is empty or not. It is cleared to "0" when data is written in the transmission buffer and set to "1" when the contents of the transmission buffer register is transferred to the transmission register.
When the transmission register becomes empty after the contents has been transmitted, data is transferred automatically form the transmission buffer register to the transmission register if the next transmission start condition is satisfied.
Once transmission has started, the TE$_i$ flag, TI$_i$ flag, and $\overline{CTS_i}$ signal (if $\overline{CTS_i}$ input is selected) are ignored until data transmission is completed.
Therefore, transmission does not stop until it completes even if the TE$_i$ flag is cleared during transmission.
The transmission start condition indicated by TE$_i$ flag, TI$_i$ flag, and $\overline{CTS_i}$ is checked while the T_{ENDi} signal shown in Figure 45 is "H". Therefore, data can be transmitted continuously if the next transmission data is written in the transmission butter register and TI$_i$ flag is cleared to 0 before the T_{ENDi} signal goes "H".
The bit 3 (TxEPTY$_i$ flag) of UART$_i$ transmit/receive control register 0 changes to "1" at the next cycle after the T_{ENDi} signal goes "H" and changes to "0" when transmission starts. Therefore, this flag can be used to determine whether data transmision is completed.
When the TI$_i$ flag changes from "0" to "1", the interrupt request bit in the UART$_i$ transmissoin interrupt control register is set to "1".

Receive

Receive is enabled when the bit 2 (RE$_i$ flag) of UART$_i$ transmit/receive control register 1 is set. As shown in Figure 47, the frequency divider circuit at the receiving end begin to work when a start bit is arrived and the data is received.

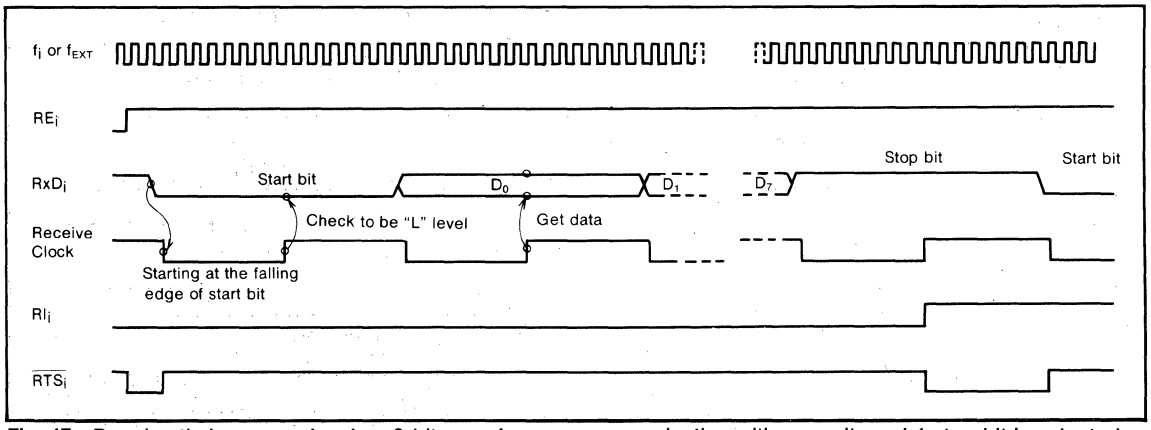


Fig. 47 Receive timing example when 8-bit asynchronous communication with no parity and 1 stop bit is selected

If $\overline{RTS_i}$ output is selected by setting the bit 2 of UART$_i$ transmit/receive control register 0 to "1", the $\overline{RTSi}$ output is "H" when the RE$_i$ flag is "0". When the RE$_i$ flag changes to "1", the $\overline{RTSi}$ output goes "L" to indicate receive ready and returns to "H" once receive has started. In other words, $\overline{RTS_i}$ output can be used to determine externally whether the receive register is ready to receive.

The entire transmission data bits are received when the start bit passes the final bit of the receive block shown in Figure 40. At this point, the contents of the receive register is transferred to the receive buffer register and the bit 3 of UART$_i$ transmit/receive control register 1 is set. In other words, the RI$_i$ flag indicates that the receive buffer register contains data when it is set. If $\overline{RTS_i}$ output is selected, $\overline{RTS_i}$ output goes "L" to indicate that the register is ready to receive the next data.

The interrupt request bit in the UART$_i$ receive interrupt control register is set when the RI$_i$ flag changes from "0" to "1".

The bit 4 (OER$_i$ flag) of UART$_i$ transmission control register 1 is set when the next data is transferred from the receive register to the receive buffer register while the RI$_i$ flag is "1". In other words when an overrun error occurs. If the OER$_i$ flag is "1", it indicates that the next data has been transferred to the receive buffer register before the contents of the receive butter register has been read.

Bit 5 (FER$_i$ flag) is set when the number of stop bits is less than required (framing error).

Bit 6 (PER$_i$ flag) is set when a parity error occurs.

Bit 7 (SUM$_i$ flag) is set when either the OER$_i$ flag, FER$_i$ flag, or the PER$_i$ flag is set. Therefore, the SUM$_i$ flag can be used to determine whether there is an error.

The setting of the RI$_i$ flag, OER$_i$ flag, FER$_i$ flag, and the PERi flag is performed while transferring the contents of the receive register to the receive buffer register. The RI$_i$ OER$_i$, FER$_i$, PER$_i$, and SUM$_i$ flags are cleared when the low order byte of the receive buffer register is read or when the RE$_i$ flag is cleared.

Sleep mode

The sleep mode is used to communicate only between certain microcomputers when multiple microcomputers are connected through serial I/O.

The sleep mode is entered when the bit 7 of UART$_i$ transmit/receive mode register is set.

The operation of the sleep mode for an 8-bit asynchronous communication is described below.

When sleep mode is selected, the contents of the receive register is not transferred to the receive buffer register if bit 7 (bit 6 if 7-bit asynchronous communication and bit 8 if 9-bit asychronous communication) of the received data is "0". Also the RI$_i$, OER$_i$, FER$_i$, PER$_i$, and the SUM$_i$ flag are unchanged. Therefore, the interrupt request bit of the UART$_i$ receive interrupt control register is also unchanged.

Normal receive operation takes place when bit 7 of the received data is "1".

The following is an example of how the sleep mode can be used.

The main microcomputer first sends data with bit 7 set to "1" and bits 0 to 6 set to the address of the subordinate microcomputer which wants to communicate with. Then all subordinate microcomputers receive the same data. Each subordinate microcomputer checks the received data, clears the sleep bit if bits 0 to 6 are its own address and sets the sleep bit if not. Next the main microcomputer sends data with bit 7 cleared. Then the microcomputer with the sleep bit cleared will receive the data, but the microcomputer with the sleep bit set will not. In this way, the main microcomputer is able to communicate with only the designated microcomputer.

A-D CONVERTER

The A-D converter is an 8-bit successive approximation converter.

Figure 48 shows a block diagram of the A-D converter and Figure 49 shows the bit configuration of the A-D control register. The frequency of the A-D converter operating clock ϕ_{AD} is selected by the bit 7 of the A-D control register. When bit 7 is "0", ϕ_{AD} is the clock frequency divided by 8. That is, $\phi_{AD}=f(X_{IN})/8$. When bit 7 is "1", ϕ_{AD} is the clock frequency divided by 4 and ϕ_{AD} is $=f(X_{IN})/4$. The ϕ_{AD} during A-D conversion must be 250kHz minimum because the comparator consists of a capacity coupling amplifier.

The operating mode is selected by the bits 3 and 4 of A-D control register. The available operating modes are one-shot, repeat, single sweep, and repeat sweep.

The bit of data direction register bit corresponding to the A-D converter pin must be "0" (input mode) because the analog input port is shared with port P7.

The operation of each mode is described below.

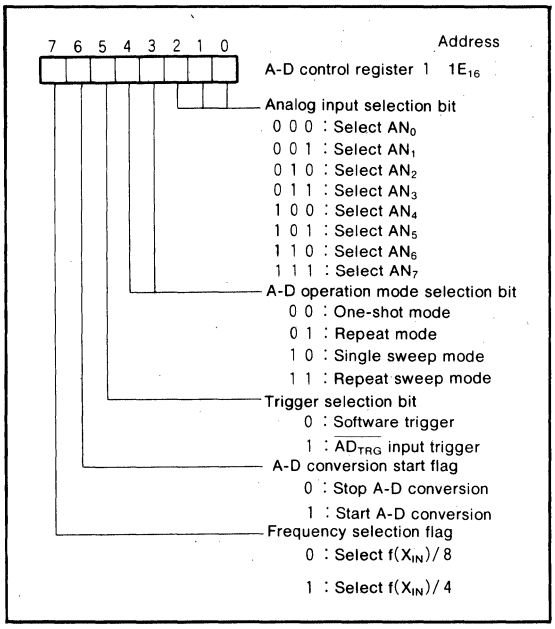


Fig. 49 A-D control register bit configuration

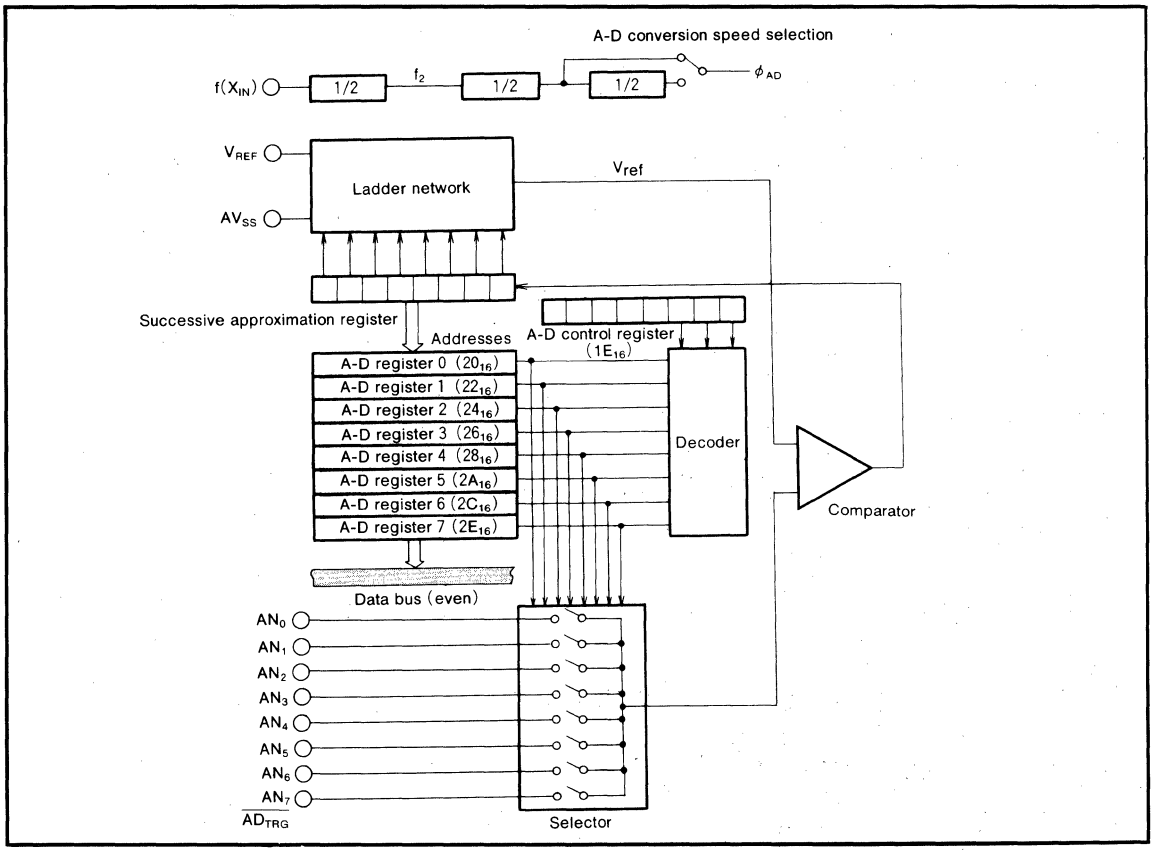


Fig. 48 A-D converter block diagram

(1) One-shot mode [00]

The A-D conversion pins are selected with the bit 0 to 2 of A-D control register. A-D conversion can be started by a software trigger or by an external trigger.

A software trigger is selected when the bit 5 of A-D control register is "0" and an external trigger is selected when it is "1".

When a software trigger is selected, A-D conversion is started when bit 6 (A-D conversion start flag) is set. A-D conversion ends after 57 ϕ_{AD} cycles and an interrupt request bit is set in the A-D conversion interrupt control register. At the same time, A-D control register bit 6 (A-D conversion start flag) is cleared and A-D conversion stops. The result of A-D conversion is stored in the A-D register corresponding to the selected pin.

If an external trigger is selected, A-D conversion starts when the A-D conversion start flag is "1" and the $\overline{AD_{TRG}}$ input changes from "H" to "L". In this case, the pins that can be used for A-D conversion are AN_0 to AN_6 because the $\overline{AD_{TRG}}$ pin is shared with the analog voltage input pin AN_7.

The operation is the same as with software trigger except that the A-D conversion start flag is not cleared after A-D conversion and a retrigger can be available during A-D conversion.

(2) Repeat mode [01]

The operation of this mode is the same as the operation of one-shot mode except that when A-D conversion of the selected pin is complete and the result is stored in the A-D register, conversion does not stop, but is repeated. Also, no interrupt request is issued in this mode. Furthermore, if software trigger is selected, the A-D conversion start flag is not cleared. The contents of the A-D register can be read at any time.

(3) Single sweep mode [10]

In the sweep mode, the number of analog input pins to be swept can be selected. Analog input pins are selected by bits 1 and 0 of the A-D sweep pin selection register ($1F_{16}$ address) shown in Figure 50. Two pins, four pins, six pins, or eight pins can be selected as analog input pins, depending on the contents of these bits.

A-D conversion is performed only for selected input pins. After A-D conversion is performed for input of AN_0 pin, the conversion result is stored in A-D register 0, and in the same way, A-D conversion is performed for selected pins one after another. After A-D conversion is performed for all selected pins, the sweep is stopped.

A-D conversion can be started with a software trigger or with an external trigger input. A software trigger is selected when bit 5 is "0" and an external trigger is selected when it is "1".

When a software trigger is selected, A-D conversion is started when A-D control register bit 6 (A-D conversion start flag) is set. When A-D conversion of all selected pins end, an interrupt request bit is set in the A-D conversion interrupt control register. At the same time, A-D control register bit 6 (A-D conversion start flag) is cleared and A-D conversion stops.

When an external trigger is selected, A-D conversion starts when the A-D conversion start flag is "1" and the $\overline{AD_{TRG}}$ input changes from "H" to "L". In this case, the A-D conversion result of the trigger input itself is stored in the A-D register 7 because the $\overline{AD_{TRG}}$ pin is shared with AN_7 pin.

The operation is the same as done by software trigger except that the A-D conversion start flag is not cleared after A-D conversion and a retrigger can be available during A-D conversion.

(4) Repeat sweep mode [11]

The difference with the single sweep mode is that A-D conversion does not stop after converting from the AN_0 pin to the selected pins, but repeats again from the AN_0 pin. The repeat is performed among the selected pins. Also, no interrupt request is generated. Furthermore, if software trigger is selected, the A-D conversion start flag is not cleared. The A-D register can be read at any time.

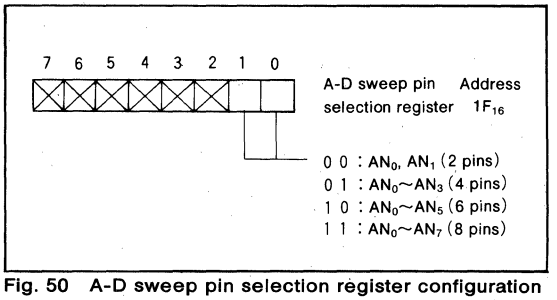


Fig. 50 A-D sweep pin selection register configuration

WATCHDOG TIMER

The watchdog timer is used to detect unexpected execution sequence caused by software run-away.

Figure 51 shows a block diagram of the watchdog timer. The watchdog timer consists of a 12-bit binary counter.

The watchdog timer counts the clock frequency divided by 32 (f_{32}) or by 512 (f_{512}). Whether to count f_{32} or f_{512} is determined by the watchdog timer frequency selection flag shown in Figure 52. f_{512} is selected when the flag is "0" and f_{32} is selected when it is "1". The flag is cleared after reset.

FFF_{16} is set in the watchdog timer when "L" or $2V_{CC}$ is applied to the $\overline{RESET}$ pin, STP instruction is executed, data is written to the watchdog timer, or the most significant bit of the watchdog timer become "0".

After FFF_{16} is set in the watchdog timer, the contents of watchdog timer is decremented by one at every cycle of selected frequency f_{32} or f_{512}, and after 2048 counts, the most significant bit of watchdog timer become "0", and a watchdog timer interrupt request bit is set, and FFF_{16} is preset in the watchdog timer.

Normally, a program is written so that data is written in the watchdog timer before the most significant bit of the watchdog timer become "0". If this routine is not executed due to unexpected program execution, the most significant bit of the watchdog timer become eventually "0" and an interrupt is generated.

The processor can be reset by setting the bit 3 (software reset bit) of processor mode register described in Figure 10 in the interrupt section and generating a reset pulse.

The watchdog timer stops its function when the $\overline{RESET}$ pin voltage is raised to double the V_{CC} voltage.

The watchdog timer can also be used to recover from when the clock is stopped by the STP instruction. Refer to the section on clock generation circuit for more details.

The watchdog timer hold the contents during a hold state and the frequency is stopped to input.

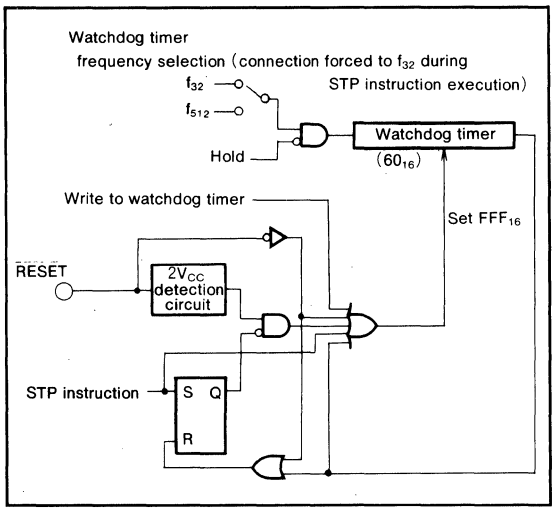


Fig. 51 Watchdog timer block diagram

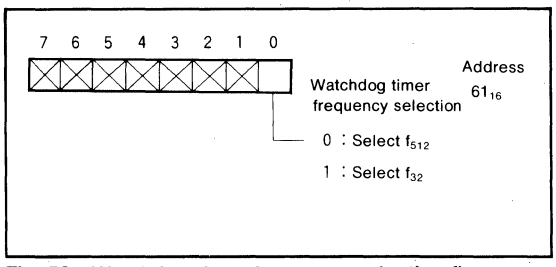


Fig. 52 Watchdog timer frequency selection flag

RESET CIRCUIT

Reset occurs when the $\overline{\text{RESET}}$ pin is returned to "H" level after holding it at "L" level when the power voltage is at 5V ±10%. Program execution starts at the address formed by setting the address pins A_{23} ~ A_{16} to 00_{16}, A_{15} ~ A_8 to the contents of address $FFFF_{16}$, and A_7 ~ A_0 to the contents of address $FFFE_{16}$.

Figure 53 shows the status of the internal registers when a reset occurs.

Figure 54 shows an example of a reset circuit. The reset input voltage must be held 0.9V or lower when the power voltage reaches 4.5V.

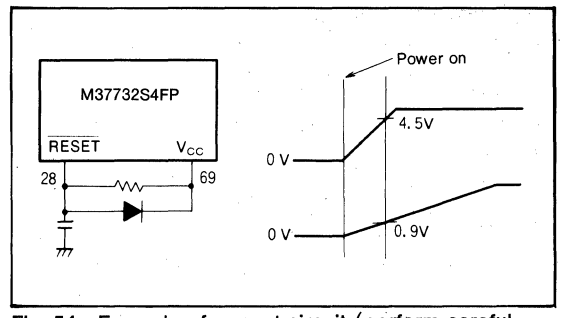


Fig. 54 Example of a reset circuit (perform careful evaluation at the system design level before using)

		Address	
(1)	Port P4 data directional register	($0C_{16}$)···	00_{16}
(2)	Port P5 data directional register	($0D_{16}$)···	00_{16}
(3)	Port P6 data directional register	(10_{16})···	00_{16}
(4)	Port P7 data directional register	(11_{16})···	00_{16}
(5)	Port P8 data directional register	(14_{16})···	00_{16}
(6)	A-D control register	($1E_{16}$)···	0 0 0 0 0 ? ? ?
(7)	A-D sweep pin selection register	($1F_{16}$)···	X X X X X X 1 1
(8)	UART 0 Transmit/Receive mode register	(30_{16})···	00_{16}
(9)	UART 1 Transmit/Receive mode register	(38_{16})···	00_{16}
(10)	UART 0 Transmit/Receive control register 0	(34_{16})···	X X X X 1 0 0 0
(11)	UART 1 Transmit/Receive control register 0	($3C_{16}$)···	X X X X 1 0 0 0
(12)	UART 0 Transmit/Receive control register 1	(35_{16})···	0 0 0 0 0 0 1 0
(13)	UART 1 Transmit/Receive control register 1	($3D_{16}$)···	0 0 0 0 0 0 1 0
(14)	Count start flag	(40_{16})···	00_{16}
(15)	One-shot start flag	(42_{16})···	X X X 0 0 0 0 0
(16)	Up-down flag	(44_{16})···	00_{16}
(17)	Timer A0 mode register	(56_{16})···	00_{16}
(18)	Timer A1 mode register	(57_{16})···	00_{16}
(19)	Timer A2 mode register	(58_{16})···	00_{16}
(20)	Timer A3 mode register	(59_{16})···	00_{16}
(21)	Timer A4 mode register	($5A_{16}$)···	00_{16}
(22)	Timer B0 mode register	($5B_{16}$)···	0 0 1 X 0 0 0 0
(23)	Timer B1 mode register	($5C_{16}$)···	0 0 1 X 0 0 0 0
(24)	Timer B2 mode register	($5D_{16}$)···	0 0 1 X 0 0 0 0
(25)	Processor mode register	($5E_{16}$)···	X 0 0 0 0 0 1 0
(26)	Watchdog timer	(60_{16})···	FFF_{16}
(27)	Watchdog timer frequency selection flag	(61_{16})···	X X X X X X X 0

		Address	
(28)	Waveform output mode register	(62_{16})···	X 0 0 0 X 0 0
(29)	A-D conversion interrupt control register	(70_{16})···	X X X 0 0 0 0
(30)	UART 0 transmission interrupt control register	(71_{16})···	X X X 0 0 0 0
(31)	UART 0 receive interrupt control register	(72_{16})···	X X X 0 0 0 0
(32)	UART 1 transmission interrupt control register	(73_{16})···	X X X 0 0 0 0
(33)	UART 1 receive interrupt control register	(74_{16})···	X X X 0 0 0 0
(34)	Timer A0 interrupt control register	(75_{16})···	X X X 0 0 0 0
(35)	Timer A1 interrupt control register	(76_{16})···	X X X 0 0 0 0
(36)	Timer A2 interrupt control register	(77_{16})···	X X X 0 0 0 0
(37)	Timer A3 interrupt control register	(78_{16})···	X X X 0 0 0 0
(38)	Timer A4 interrupt control register	(79_{16})···	X X X 0 0 0 0
(39)	Timer B0 interrupt control register	($7A_{16}$)···	X X X 0 0 0 0
(40)	Timer B1 interrupt control register	($7B_{16}$)···	X X X 0 0 0 0
(41)	Timer B2 interrupt control register	($7C_{16}$)···	X X X 0 0 0 0
(42)	$\overline{\text{INT}}_0$ interrupt control register	($7D_{16}$)···	X X 0 0 0 0 0
(43)	$\overline{\text{INT}}_1$ interrupt control register	($7E_{16}$)···	X X 0 0 0 0 0
(44)	$\overline{\text{INT}}_2$ interrupt control register	($7F_{16}$)···	X X 0 0 0 0 0
(45)	Processor status register PS		0 0 0 ? ? 0 0 0 1 ? ?
(46)	Program bank register PG		00_{16}
(47)	Program counter PC_H		Content of $FFFF_{16}$
(48)	Program counter PC_L		Content of $FFFE_{16}$
(49)	Direct page register DPR		0000_{16}
(50)	Data bank register DT		00_{16}

Contents of other registers and RAM are not initialized and should be initialized by software.

Fig. 53 Microcomputer internal status during reset

MITSUBISHI ELECTRIC

INPUT/OUTPUT PINS

Ports P8 to P4 all have a data direction register and each bit can be programmed for input or output. A pin becomes an output pin when the corresponding data direction register is set and an input pin when it is cleared.

When pin programmed for output, the data is written to the port latch and it is output to the output pin. When a pin is programmed for output, the contents of the port latch is read instead of the value of the pin. Therefore, a previously output value can be read correctly even when the output "L" voltage is raised due to reasons such as directly driving an LED.

A pin programmed for input is floating and the value input to the pin can be read. When a pin is programmed for input, the data is written only in the port latch and the pin stays floating.

If an input/output pin is not used as an output port, clear the bit of the corresponding data direction register so that the pin become input mode.

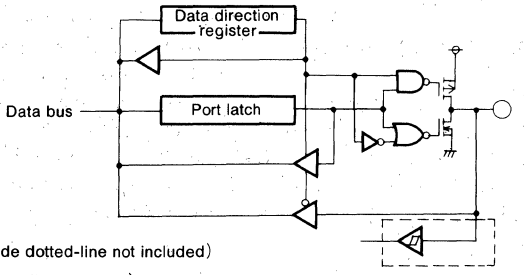
Figure 55 shows a block diagram of ports P8 to P4 and the $\overline{E}$ pin output.

In evaluation chip mode, port P4 is also used as control signal pins.

Refer to the section on processor modes for more details.

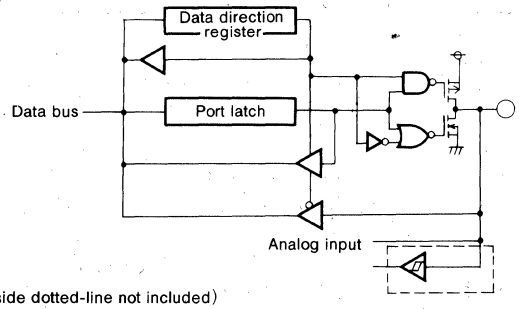
• Ports $P4_6$~$P4_3$ (Inside dotted-line not included)

Ports $P4_7$, $P5_7$, $P6_7$~$P6_1$, $P8_2$, $P8_6$ (Inside dotted-line included, but $P8_2$, $P8_6$ are without hysterisis)



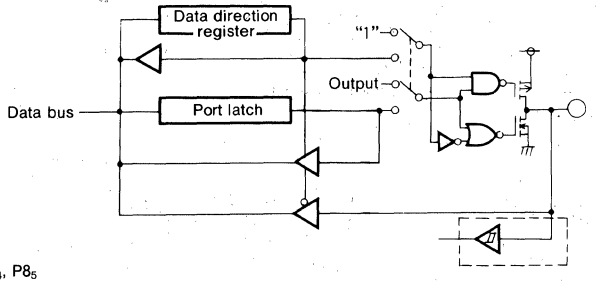
• Ports $P7_6$~$P7_0$ (Inside dotted-line not included)

• Port $P7_7$ (Inside dotted-line included)

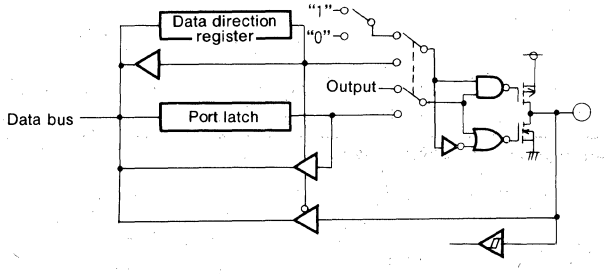


• Ports $P8_3$, $P8_7$ (Inside dotted-line not included)

Ports $P5_0$~$P5_6$, $P6_0$ (Inside dotted-line included)



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



• $\overline{E}$

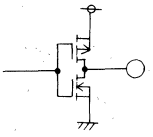


Fig. 55 Block diagram for ports P8 to P4 and the $\overline{E}$ pin output

PROCESSOR MODE

The bit 0 of processor mode register as shown in Figure 56 is used to select either, microprocessor mode, or evaluation chip mode.

Figure 57 shows the functions of A_0 to A_7 pins, A_8/D_8 to A_{23}/D_7 pins, and port P4 in each mode.

The external memory area changes when the mode changes.

Figure 58 shows the memory map for each mode.

The accessing of the external memory is affected by the BYTE pin and the bit 2 (wait bit) of processor mode register . These will be described next.

●BYTE pin

When accessing the external memory, the level of the BYTE pin is used to determine whether to use the data bus as 8-bit width or 16-bit width.

The data bus width is 8 bits when the level of the BYTE pin is "H" and A_{16}/D_0 to A_{23}/D_7 become the data I/O pin.

The data bus width is 16 bits when the level of the BYTE pin is "L" and A_{16}/D_0 to A_{23}/D_7 pins and A_8/D_8 to A_{15}/D_{15} pins become the data I/O pins.

When accessing the internal memory, the data bus width is always 16 bits regardless of the BYTE pin level.

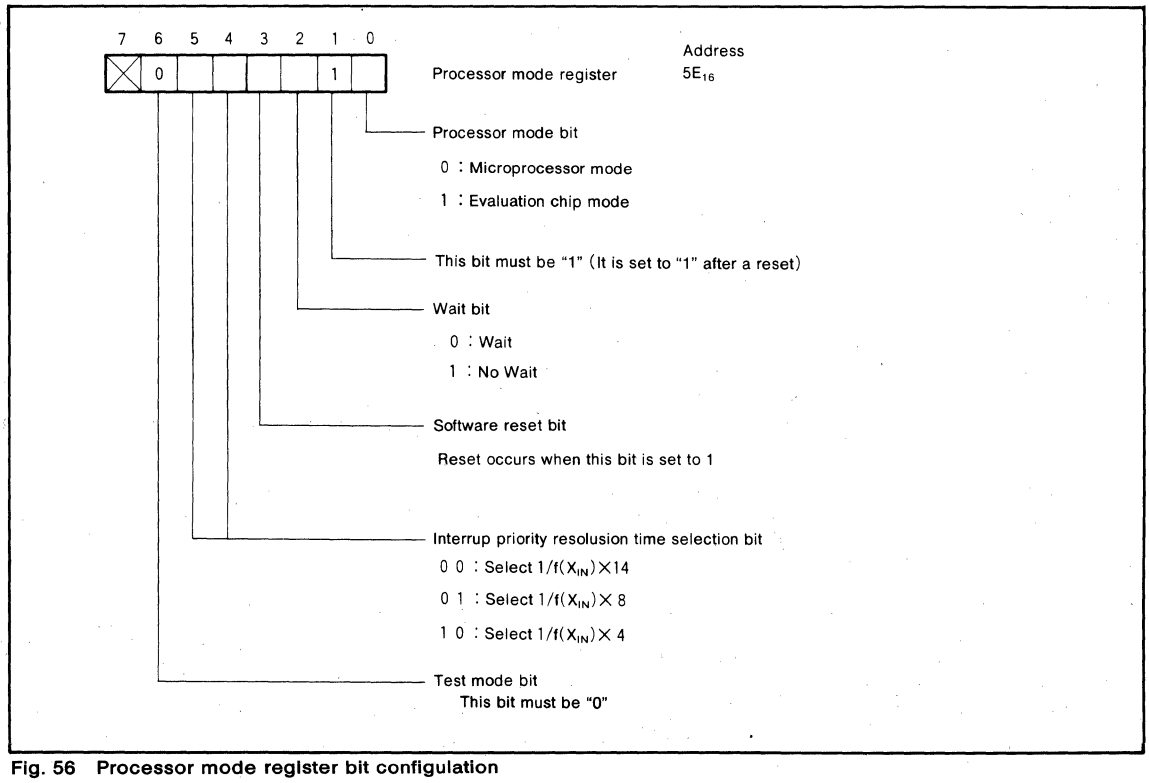


Fig. 56 Processor mode register bit configulation

Port \ Mode		Microprocessor Mode	Evaluation Chip Mode
	CM_1	1	1
	CM_0	0	1
$A_0 \sim A_7$		$\overline{E}$, A_0 ∫ A_7: Address $A_7 \sim A_0$	Same as left
A_8/D_8 ∫ A_{15}/D_{15}	BYTE = "L"	$\overline{E}$, A_8/D_8 ∫ A_{15}/D_{15}: $A_{15} \sim A_8$, Address, Data (odd)	Same as left
A_8/D_8 ∫ A_{15}/D_{15}	BYTE = "H"	$\overline{E}$, A_8/D_8 ∫ A_{15}/D_{15}: Address $A_{15} \sim A_8$	$\overline{E}$, A_8/D_8 ∫ A_{15}/D_{15}: $A_{15} \sim A_8$, Address, Data (odd) Ports P4, P5 and their direction registers are treated as 16-bit wide bus.
A_{16}/D_0 ∫ A_{23}/D_7	BYTE = "L"	$\overline{E}$, A_{16}/D_0 ∫ A_{23}/D_7: $A_{23} \sim A_{16}$, Address, Data (even)	Same as left
A_{16}/D_0 ∫ A_{23}/D_7	BYTE = "H"	$\overline{E}$, A_{16}/D_0 ∫ A_{23}/D_7: $A_{23} \sim A_{16}$, Address, Data (even, odd)	$\overline{E}$, A_{16}/D_0 ∫ A_{23}/D_7: $A_{23} \sim A_{16}$, Address, Data (even, odd) Same as A_8/D_8 to A_{15}/D_{15}
Port P4		$\overline{E}$, $P4_7$ ∫ $P4_3$: I/O Port	$\overline{E}$, $P4_7$: $\overline{DBC}$, $P4_6$: VPA, $P4_5$: VDA, $P4_4$: QCL, $P4_3$: MX

Fig. 57 Processor mode and A_0 to A_7 pins, A_8/D_8 to A_{23}/D_7 pins and port P4 functions

•Wait bit

As shown in Figure 59, when the external memory area is accessed with the processor mode register bit 2 (wait bit) cleared to "0", the "L" width of $\overline{E}$ signal becomes twice compared with no wait (the wait bit is "1"). The wait bit is cleared during reset.

The accessing of internal memory area is performed in no wait mode regardless of the wait bit.

The processor modes are described below.

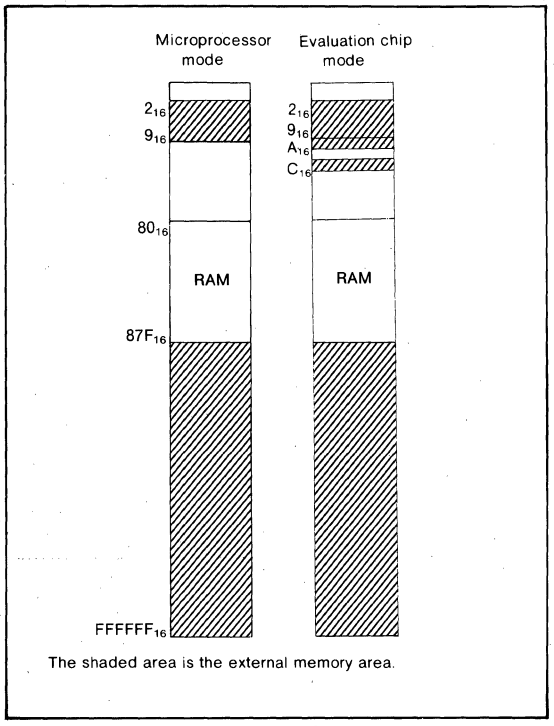


Fig. 58 External memory area for each processor mode

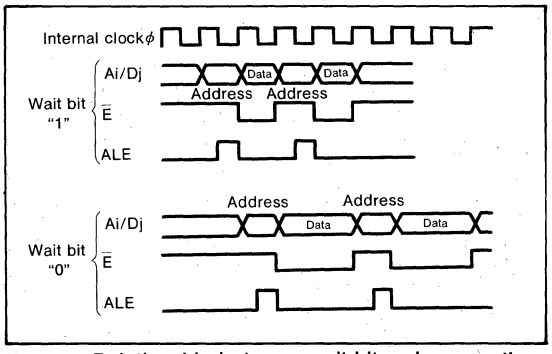


Fig. 59 Relationship between wait bit and access time

(1) Microprocessor mode [10]

Microprocessor mode is entered by connecting the CNV_{SS} pin to V_{CC} and starting from reset.

A_8/D_8 to A_{15}/D_{15} pins have two functions depending on the level of the BYTE pin.

When the BYTE pin level is "L", A_8/D_8 to A_{15}/D_{15} pins function as an address output pin while $\overline{E}$ is "H" and as an odd address data I/O pin while $\overline{E}$ is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

When the BYTE pin level "H", A_8/D_8 to A_{15}/D_{15} pins function as an address output pin.

A_{16}/D_0 to A_{23}/D_7 pins have two functions depending on the level of the BYTE pin.

When the BYTE pin level is "L", $A_{16}/D_0 \sim A_{23}/D_7$ pins function as an address output pin while $\overline{E}$ is "H" and as an even address data I/O pin while $\overline{E}$ is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

When the BYTE pin level is "H", $A_{16}/D_0 \sim A_{23}/D_7$ pins functions as an address output pin while $\overline{E}$ is "H" and as an even and odd address data I/O pin while $\overline{E}$ is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

R/$\overline{W}$ is a read/write signal which indicates a read when it is "H" and a write when it is "L".

$\overline{BHE}$ is a byte high enable signal which indicates that an odd address is accessed when it is "L".

Therefore, two bytes at even and odd addresses are accessed simultaneously if address A_0 is "L" and $\overline{BHE}$ is "L".

ALE is an address latch enable signal used to latch the address signal from a multiplexed signal of address and data. The latch is transparent while ALE is "H" to let the address signal pass through and held while ALE is "L".

$\overline{HLDA}$ is a hold acknowledge signal and is used to notify externally when the microcomputer receives $\overline{HOLD}$ input and enters into hold state.

$\overline{HOLD}$ is a hold request signal. It is an input signal used to put the microcomputer in hold state. $\overline{HOLD}$ input is accepted when the internal clock ϕ falls from "H" level to "L" level while the bus is not used. A_0 to A_7 pins, A_8/D_8 to A_{23}/D_7 pins, R/$\overline{W}$ pin and $\overline{BHE}$ pin are floating while the microcomputer stays in hold state. These ports are floating after one cycle of the internal clock ϕ later than $\overline{HLDA}$ signal changes to "L" level. At the removing of hold state, these ports are removed from floating state after one cycle of ϕ later than $\overline{HLDA}$ signal changes to "H" level.

$\overline{RDY}$ is a ready signal. If this signal goes "L", the internal clock ϕ stops at "L". ϕ_1 output from clock ϕ_1 output pin doesn't stop. $\overline{RDY}$ is used when slow external memory is attached.

(2) Evaluation chip mode [11]

Evaluation chip mode is entered by applying voltage twice the V_{CC} voltage to the CNV_{SS} pin. This mode is normally used for evaluation tools.

A_8/D_8 to A_{15}/D_{15} functions as an address output pin while $\overline{E}$ is "H" and as data I/O pin of odd addresses while $\overline{E}$ is "L" regardless of the BYTE pin level. However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

A_{16}/D_0 to A_{23}/D_7 function as an address output pin while $\overline{E}$ is "H" and as data I/O pin of even addresses while $\overline{E}$ is "L" when the BYTE pin level is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

When the BYTE pin level is "H", A_{16}/D_0 to A_{23}/D_7 functions as an address output pin while $\overline{E}$ is "H" and as data I/O pin of even and odd addresses while $\overline{E}$ is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

Port P4 and its data direction register which are located at address $0A_{16}$ and $0C_{16}$ are treated differently in evaluation chip mode. When these addresses are accessed, the data bus width is treated as 16 bits regardless of the BYTE pin level, and the access cycle is treated as internal memory regardless of the wait bit.

Ports $P4_3$ to $P4_6$ become MX, QCL, VDA, and VPA output pins respectively. Port $P4_7$ becomes the $\overline{DBC}$ input pin.

The MX signal normally contains the contents of flag m, but the contents of flag x is output if the CPU is using flag x.

QCL is the queue buffer clear signal. It becomes "H" when the instruction queue buffer is cleared, for example, when a jump instruction is executed.

VDA is the valid data address signal. It becomes "H" while the CPU is reading data from data buffer or writing data to data buffer. It also becomes "H" when the first byte of the instruction (operation code) is read from the instruction queue buffer.

VPA is the valid program address signal. It becomes "H" while the CPU is reading an instruction code from the instruction queue buffer.

$\overline{DBC}$ is the debug control signal and is used for debugging.

Table 5 shows the relationship between the CNV_{SS} pin input levels and processor modes.

Table 5. Relationship between the CNV_{SS} pin input levels and processor modes

CNV_{SS}	Mode	Description
V_{CC}	• Microprocessor • Evaluation chip	Microprocessor mode upon starting after reset. Evaluation chip mode can be selected by changing the processor mode bit by software.
$2 \cdot V_{CC}$	• Evaluation chip	• Evaluation chip mode only.

CLOCK GENERATING CIRCUIT

Figure 60 shows a block diagram of the clock generator.

When an STP instruction is executed, the internal clock ϕ stops oscillating at "L" level. At the same time, FFF_{16} is written to watchdog timer and the watchdog timer input connection is forced to f_{32}. This connection is broken and connected to the input determined by the watchdog timer frequency selection flag when the most significant bit of the watchdog timer is cleared or reset.

Oscillation resumes when an interrupt is received, but the internal clock ϕ remains at "L" level until the most significant bit of the watchdog timer is cleared. This is to avoid the unstable interval at the start of oscillation when using a ceramic resonator.

When a WIT instruction is executed, the internal clock ϕ stops at "L" level, but the oscillator does not stop. The clock is restarted when an interrupt is received. Instructions can be executed immediately because the oscillator is not stopped.

The stop or wait state is released when an interrupt is received or when reset is issued. Therefore, interrupts must be enabled before executing a STP or WIT instruction.

Figure 61 shows a circuit example using a ceramic (or quartz crystal) resonator. Use the manufacturer's recommended values for constants such as capacitance which differ for each resonator. Figure 62 shows an example of using an external clock signal.

ADDRESSING MODES

The M37732S4FP has 28 powerful addressing modes.

Refer to the MELPS 7700 addressing mode description for the details of each addressing mode.

MACHINE INSTRUCTION LIST

The M37732S4FP has 103 machine instructions. Refer to the MELPS 7700 machine instruction list for details.

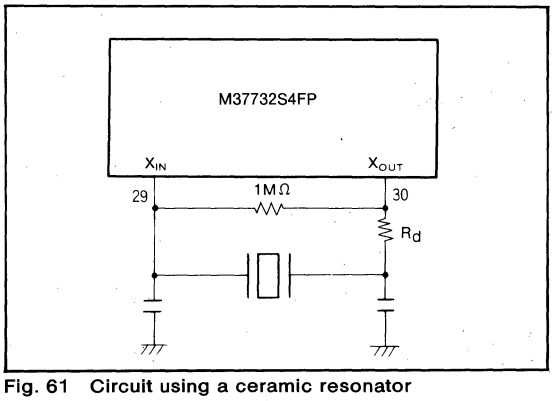


Fig. 61 Circuit using a ceramic resonator

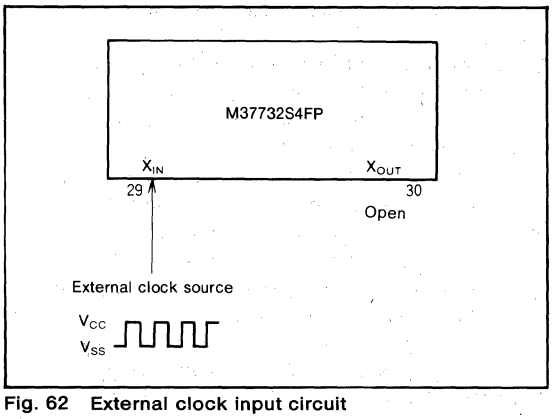


Fig. 62 External clock input circuit

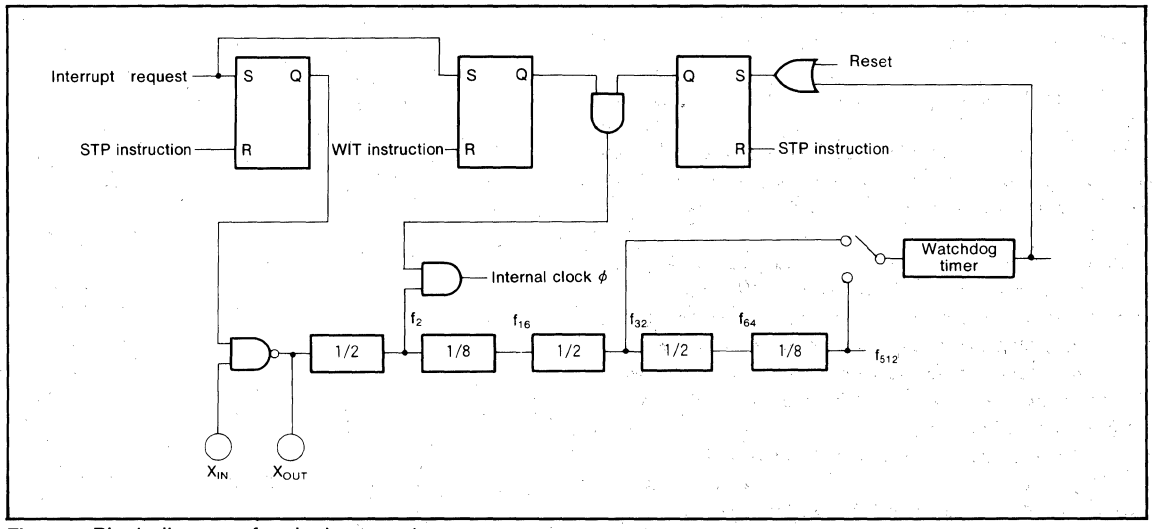


Fig. 60 Block diagram of a clock generator

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Supply voltage		−0.3∼7	V
AV_{CC}	Analog supply voltage		−0.3∼7	V
V_I	Input voltage $\overline{RESET}$, CNV_{SS}, BYTE		−0.3∼12	V
V_I	Input voltage A_0∼A_7, A_8/D_8∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, V_{REF}, X_{IN}, $\overline{HOLD}$, $\overline{RDY}$		−0.3∼V_{CC}+0.3	V
V_O	Output voltage A_0∼A_7, A_8/D_8∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, X_{OUT}, $\overline{E}$, ϕ_1, $\overline{HLDA}$, ALE, $\overline{BHE}$, $R/\overline{W}$		−0.3∼V_{CC}+0.3	V
P_d	Power dissipation	T_a=25℃	300	mW
T_{opr}	Operating temperature		−20∼85	℃
T_{stg}	Storage temperature		−40∼150	℃

RECOMMENDED OPERATING CONDITIONS (V_{CC}=5V±10%, T_a=−20∼85℃, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V_{CC}	Supply voltage		4.5	5.0	5.5	V
AV_{CC}	Analog supply voltage			V_{CC}		V
V_{SS}	Supply voltage			0		V
AV_{SS}	Analog supply voltage			0		V
V_{IH}	High-level input voltage $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE, $\overline{HOLD}$, $\overline{RDY}$		0.8V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage A_8/D_8∼A_{23}/D_7		0.5V_{CC}		V_{CC}	V
V_{IL}	Low-level input voltage $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE, $\overline{HOLD}$, $\overline{RDY}$		0		0.2V_{CC}	V
V_{IL}	Low-level input voltage A_8/D_8∼A_{23}/D_7		0		0.16V_{CC}	V
$I_{OH(peak)}$	High-level peak output current A_0∼A_7, A_8/D_8∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, ϕ_1, $\overline{HLDA}$, ALE, $\overline{BHE}$, $R/\overline{W}$				−10	mA
$I_{OH(avg)}$	High-level average output current A_0∼A_7, A_8/D_8∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, ϕ_1, $\overline{HLDA}$, ALE, $\overline{BHE}$, $R/\overline{W}$				−5	mA
$I_{OL(peak)}$	Low-level peak output current A_0∼A_7, A_8/D_8∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, ϕ_1, $\overline{HLDA}$, ALE, $\overline{BHE}$, $R/\overline{W}$				10	mA
$I_{OL(avg)}$	Low-level average output current A_0∼A_7, A_8/D_8∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, ϕ_1, $\overline{HLDA}$, ALE, $\overline{BHE}$, $R/\overline{W}$				5	mA
$f(X_{IN})$	External clock frequency input	M37732S4FP			8	MHz
		M37732S4AFP			16	
		M37732S4BFP			25	

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

2. The sum of $I_{OL(peak)}$ for ports A_0∼A_7, A_8/D_8∼A_{23}/D_7, $\overline{HLDA}$, ALE, $\overline{BHE}$, $R/\overline{W}$, and P8 must be 80mA or less, the sum of $I_{OH(peak)}$ for ports A_0∼A_7, A_8/D_8∼A_{23}/D_7, $\overline{HLDA}$, ALE, $\overline{BHE}$, $R/\overline{W}$, and P8 must be 80mA or less, the sum of $I_{OL(peak)}$ for ports P4, P5, P6, P7, ϕ_1 must be 80mA or less, and the sum of $I_{OH(peak)}$ for ports P4, P5, P6, P7, ϕ_1 must be 80mA or less.

M37732S4FP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=8MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	High-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OH}=−10mA	3			V
V_{OH}	High-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OH}=−400μA	4.7			V
V_{OH}	High-level output voltage ALE	I_{OH}=−10mA	3.1			V
		I_{OH}=−400μA	4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA	3.4			V
		I_{OH}=−400μA	4.8			
V_{OL}	Low-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OL}=10mA			2	V
V_{OL}	Low-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OL}=2mA			0.45	V
V_{OL}	Low-level output voltage ALE	I_{OL}=10mA			1.9	V
		I_{OL}=2mA			0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA			1.6	V
		I_{OL}=2mA			0.4	
$V_{T+}-V_{T-}$	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA4_{IN}$, $TB0_{IN}$~$TB2_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, $\overline{AD_{TRG}}$, $\overline{CTS_0}$, $\overline{CTS_1}$, CLK_0, CLK_1		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis $\overline{RESET}$		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE, $\overline{HOLD}$, $\overline{RDY}$	V_I=5V			5	μA
I_{IL}	Low-level input current A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE, $\overline{HOLD}$, $\overline{RDY}$	V_I=0V			−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	Output only pin is open and other pins are V_{SS} during reset. f(X_{IN})=8MHz, square waveform		6	12	mA
		T_a=25℃ when clock is stopped.			1	μA
		T_a=85℃ when clock is stopped.			20	

A-D CONVERTER CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=8MHz, unless otherwise noted)

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

M37732S4AFP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=16MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits Min.	Limits Typ.	Limits Max.	Unit
V_{OH}	High-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, ϕ_1, HLDA, $\overline{BHE}$, R/$\overline{W}$	I_{OH}=−10mA	3			V
V_{OH}	High-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, ϕ_1, HLDA, $\overline{BHE}$, R/$\overline{W}$	I_{OH}=−400μA	4.7			V
V_{OH}	High-level output voltage ALE	I_{OH}=−10mA	3.1			V
		I_{OH}=−400μA	4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA	3.4			V
		I_{OH}=−400μA	4.8			
V_{OL}	Low-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, ϕ_1, HLDA, $\overline{BHE}$, R/$\overline{W}$	I_{OL}=10mA			2	V
V_{OL}	Low-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, ϕ_1, HLDA, $\overline{BHE}$, R/$\overline{W}$	I_{OL}=2mA			0.45	V
V_{OL}	Low-level output voltage ALE	I_{OL}=10mA			1.9	V
		I_{OL}=2mA			0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA			1.6	V
		I_{OL}=2mA			0.4	
$V_{T+}-V_{T-}$	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA4_{IN}$, $TB0_{IN}$~$TB2_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, AD_{TRG}, $\overline{CTS_0}$, $\overline{CTS_1}$, CLK_0, CLK_1		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis $\overline{RESET}$		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE, $\overline{HOLD}$, $\overline{RDY}$	V_I=5V			5	μA
I_{IL}	Low-level input current A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE, $\overline{HOLD}$, $\overline{RDY}$	V_I=0V			−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	Output only pin is open and other pins are V_{SS} during reset. f(X_{IN})=16MHz, square waveform		12	24	mA
		T_a=25℃ when clock is stopped.			1	μA
		T_a=85℃ when clock is stopped.			20	

A-D CONVERTER CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=16MHz, unless otherwise noted)

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

M37732S4BFP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=25MHz, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{OH}	High-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, ϕ_1, HLDA, BHE, R/W	I_{OH}=−10mA		3			V
V_{OH}	High-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, ϕ_1, HLDA, BHE, R/W	I_{OH}=−400μA		4.7			V
V_{OH}	High-level output voltage ALE	I_{OH}=−10mA		3.1			V
		I_{OH}=−400μA		4.8			
V_{OH}	High-level output voltage E	I_{OH}=−10mA		3.4			V
		I_{OH}=−400μA		4.8			
V_{OL}	Low-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, ϕ_1, HLDA, BHE, R/W	I_{OL}=10mA				2	V
V_{OL}	Low-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, ϕ_1, HLDA, BHE, R/W	I_{OL}=2mA				0.45	V
V_{OL}	Low-level output voltage ALE	I_{OL}=10mA				1.9	V
		I_{OL}=2mA				0.43	
V_{OL}	Low-level output voltage E	I_{OL}=10mA				1.6	V
		I_{OL}=2mA				0.4	
V_{T+}−V_{T-}	Hysteresis HOLD, RDY, $TA0_{IN}$~$TA4_{IN}$, $TB0_{IN}$~$TB2_{IN}$, INT_0~INT_2, AD_{TRG}, CTS_0, CTS_1, CLK_0, CLK_1			0.4		1	V
V_{T+}−V_{T-}	Hysteresis RESET			0.2		0.5	V
V_{T+}−V_{T-}	Hysteresis X_{IN}			0.1		0.3	V
I_{IH}	High-level input current A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, RESET, CNV_{SS}, BYTE, HOLD, RDY	V_I=5V				5	μA
I_{IL}	Low-level input current A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, RESET, CNV_{SS}, BYTE, HOLD, RDY	V_I=0V				−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.		2			V
I_{CC}	Power supply current	Output only pin is open and other pins are V_{SS} during reset.	f(X_{IN})=25MHz, square waveform		19	38	mA
			T_a=25℃ when clock is stopped.			1	μA
			T_a=85℃ when clock is stopped.			20	

A-D CONVERTER CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=25MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	V_{REF}=V_{CC}			8	Bits
—	Absolute accuracy	V_{REF}=V_{CC}			±3	LSB
R_{LADDER}	Ladder resistance	V_{REF}=V_{CC}	2		10	kΩ
t_{CONV}	Conversion time		9.12			μs
V_{REF}	Reference voltage		2		V_{CC}	V
V_{IA}	Analog input voltage		0		V_{REF}	V

TIMING REQUIREMENTS (V_{CC}=5V±10%, V_{SS}=0V, T_a=25℃, unless otherwise noted)

External clock input

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
t_C	External clock input cycle time	125		62		40		ns
$t_{W(H)}$	External clock input high-level pulse width	50		25		15		ns
$t_{W(L)}$	External clock input low-level pulse width	50		25		15		ns
t_r	External clock rise time		20		10		8	ns
t_f	External clock fall time		20		10		8	ns

Microprocessor mode

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{SU(DH-E)}$	Data high-order input setup time	60		45		30		ns
$t_{SU(DL-E)}$	Data low-order input setup time	60		45		30		ns
$t_{SU(P4D-E)}$	Port P4 input setup time	200		100		60		ns
$t_{SU(P5D-E)}$	Port P5 input setup time	200		100		60		ns
$t_{SU(P6D-E)}$	Port P6 input setup time	200		100		60		ns
$t_{SU(P7D-E)}$	Port P7 input setup time	200		100		60		ns
$t_{SU(P8D-E)}$	Port P8 input setup time	200		100		60		ns
$t_{SU(RDY-\phi_1)}$	$\overline{RDY}$ input setup time	70		60		55		ns
$t_{SU(HOLD-\phi_1)}$	$\overline{HOLD}$ input setup time	70		60		55		ns
$t_{h(E-DH)}$	Data high-order input hold time	0		0		0		ns
$t_{h(E-DL)}$	Data low-order input hold time	0		0		0		ns
$t_{h(E-P4D)}$	Port P4 input hold time	0		0		0		ns
$t_{h(E-P5D)}$	Port P5 input hold time	0		0		0		ns
$t_{h(E-P6D)}$	Port P6 input hold time	0		0		0		ns
$t_{h(E-P7D)}$	Port P7 input hold time	0		0		0		ns
$t_{h(E-P8D)}$	Port P8 input hold time	0		0		0		ns
$t_{h(\phi_1-RDY)}$	$\overline{RDY}$ input hold time	0		0		0		ns
$t_{h(\phi_1-HOLD)}$	$\overline{HOLD}$ input hold time	0		0		0		ns

Timer A input (Count input in event counter mode)

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	250		125		80		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	125		62		40		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	125		62		40		ns

Timer A input (Gating input in timer mode)

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	1000		500		320		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	500		250		160		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	500		250		160		ns

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

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	500		250		160		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	250		125		80		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	250		125		80		ns

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

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	250		125		80		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	250		125		80		ns

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

Symbol	Parameter	Limits						Unit
		8 MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(UP)}$	TAi_{OUT} input cycle time	5000		2500		2000		ns
$t_{W(UPH)}$	TAi_{OUT} input high-level pulse width	2500		1250		1000		ns
$t_{W(UPL)}$	TAi_{OUT} input low-level pulse width	2500		1250		1000		ns
$t_{su(UP-T_{IN})}$	TAi_{OUT} input setup time	1000		500		400		ns
$t_{h(T_{IN}-UP)}$	TAi_{OUT} input hold time	1000		500		400		ns

Timer B input (Count input in event counter mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	TBi_{IN} input cycle time (one edge count)	250		125		80		ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width (one edge count)	125		62		40		ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width (one edge count)	125		62		40		ns
$t_{C(TB)}$	TBi_{IN} input cycle time (both edges count)	500		250		160		ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width (both edges count)	250		125		80		ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width (both edges count)	250		125		80		ns

Timer B input (Pulse period measurement mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	TBi_{IN} input cycle time	1000		500		320		ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width	500		250		160		ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width	500		250		160		ns

Timer B input (Pulse width measurement mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	TBi_{IN} input cycle time	1000		500		320		ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width	500		250		160		ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width	500		250		160		ns

A-D trigger input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(AD)}$	$\overline{AD_{TRG}}$ input cycle time (minimum allowable trigger)	2000		1000		1000		ns
$t_{W(ADL)}$	$\overline{AD_{TRG}}$ input low-level pulse width	250		125		125		ns

Serial I/O

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(CK)}$	CLK_i input cycle time	500		250		200		ns
$t_{W(CKH)}$	CLK_i input high-level pulse width	250		125		100		ns
$t_{W(CKL)}$	CLK_i input low-level pulse width	250		125		100		ns
$t_{d(C-Q)}$	TxD_i output delay time		150		90		80	ns
$t_{h(C-Q)}$	TxD_i hold time	30		30		30		ns
$t_{SU(D-C)}$	RxD_i input setup time	60		30		20		ns
$t_{h(C-D)}$	RxD_i input hold time	90		90		90		ns

External interrupt $\overline{INT_i}$ input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{W(INH)}$	$\overline{INT_i}$ input high-level pulse width	250		250		250		ns
$t_{W(INL)}$	$\overline{INT_i}$ input low-level pulse width	250		250		250		ns

SWITCHING CHARACTERISTICS (V_{CC}=5V±10%, V_{SS}=0V, T_a=25℃, unless otherwise noted)

Microprocessor mode (when wait bit = "1")

Symbol	Parameter	Test conditions	Limits						Unit
			8 MHz		16MHz		25MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(AL-E)}$	Address low-order output delay time	Fig. 63	100		30		12		ns
$t_{d(E-DHQ)}$	Data high-order output delay time (BYTE="L")			110		70		45	ns
$t_{PXZ(E-DHZ)}$	Floating start delay time (BYTE="L")			5		5		5	ns
$t_{d(AM-E)}$	Address middle-order output delay time		100		30		12		ns
$t_{d(AM-ALE)}$	Address middle-order output delay time		80		24		5		ns
$t_{d(E-DLQ)}$	Data low-order output delay time			110		70		45	ns
$t_{PXZ(E-DLZ)}$	Floating start delay time			5		5		5	ns
$t_{d(AH-E)}$	Address high-order output delay time		100		30		12		ns
$t_{d(AH-ALE)}$	Address high-order output delay time		80		24		5		ns
$t_{d(\phi_1-HLDA)}$	$\overline{HLDA}$ output delay time			100		50		50	ns
$t_{d(ALE-E)}$	ALE output delay time		4		4		4		ns
$t_{W(ALE)}$	ALE pulse width		90		35		22		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time		100		30		20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time		100		30		20		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	30	0	20	0	18	ns
$t_{h(E-AL)}$	Address low-order hold time		50		25		18		ns
$t_{h(ALE-AM)}$	Address middle-order hold time (BYTE="L")		9		9		9		ns
$t_{h(E-DHQ)}$	Data high-order hold time (BYTE="L")		50		25		18		ns
$t_{PZX(E-DHZ)}$	Floating release delay time (BYTE="L")		50		25		18		ns
$t_{h(E-AM)}$	Address middle-order hold time (BYTE="H")		50		25		18		ns
$t_{h(ALE-AH)}$	Address high-order hold time		9		9		9		ns
$t_{h(E-DLQ)}$	Data low-order hold time		50		25		18		ns
$t_{PZX(E-DLZ)}$	Floating release delay time		50		25		18		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		18		18		18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		18		18		18		ns
$t_{d(E-P4Q)}$	Port P4 data output delay time			200		100		80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			200		100		80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			200		100		80	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time			200		100		80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			200		100		80	ns
$t_{W(EL)}$	$\overline{E}$ pulse width		220		95		50		ns

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

Symbol	Parameter	Test conditions	Limits						Unit
			8 MHz		16MHz		25MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(AL-E)}$	Address low-order output delay time	Fig.63	100		30		12		ns
$t_{d(E-DHQ)}$	Data high-order output delay time (BYTE="L")			110		70		45	ns
$t_{PXZ(E-DHZ)}$	Floating start delay time (BYTE="L")			5		5		5	ns
$t_{d(AM-E)}$	Address middle-order output delay time		100		30		12		ns
$t_{d(AM-ALE)}$	Address middle-order output delay time		80		24		5		ns
$t_{d(E-DLQ)}$	Data low-order output delay time			110		70		45	ns
$t_{PXZ(E-DLZ)}$	Floating start delay time			5		5		5	ns
$t_{d(AH-E)}$	Address high-order output delay time		100		30		12		ns
$t_{d(AH-ALE)}$	Address high-order output delay time		80		24		5		ns
$t_{d(\phi_1-HLDA)}$	$\overline{HLDA}$ output delay time			100		50		50	ns
$t_{d(ALE-E)}$	ALE output delay time		4		4		4		ns
$t_{W(ALE)}$	ALE pulse width		90		35		22		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time		100		30		20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time		100		30		20		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	30	0	20	0	18	ns
$t_{h(E-AL)}$	Address low-order hold time		50		25		18		ns
$t_{h(ALE-AM)}$	Address middle-order hold time (BYTE="L")		9		9		9		ns
$t_{h(E-DHQ)}$	Data high-order hold time (BYTE="L")		50		25		18		ns
$t_{PZX(E-DHZ)}$	Floating release delay time (BYTE="L")		50		25		18		ns
$t_{h(E-AM)}$	Address middle-order hold time (BYTE="H")		50		25		18		ns
$t_{h(ALE-AH)}$	Address high-order hold time		9		9		9		ns
$t_{h(E-DLQ)}$	Data low-order hold time		50		25		18		ns
$t_{PZX(E-DLZ)}$	Floating release delay time		50		25		18		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		18		18		18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		18		18		18		ns
$t_{d(E-P4Q)}$	Port P4 data output delay time			200		100		80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			200		100		80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			200		100		80	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time			200		100		80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			200		100		80	ns
$t_{W(EL)}$	$\overline{E}$ pulse width		470		220		130		ns

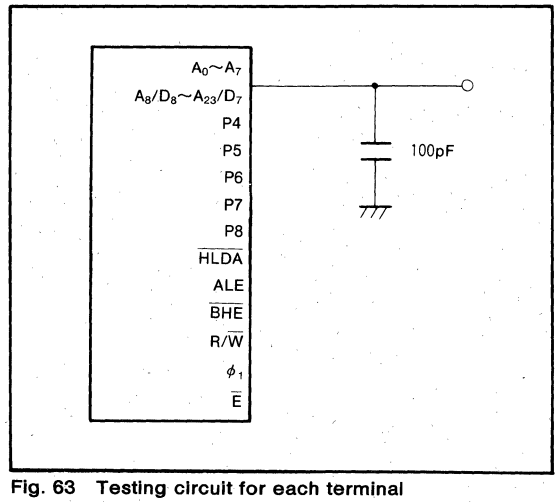
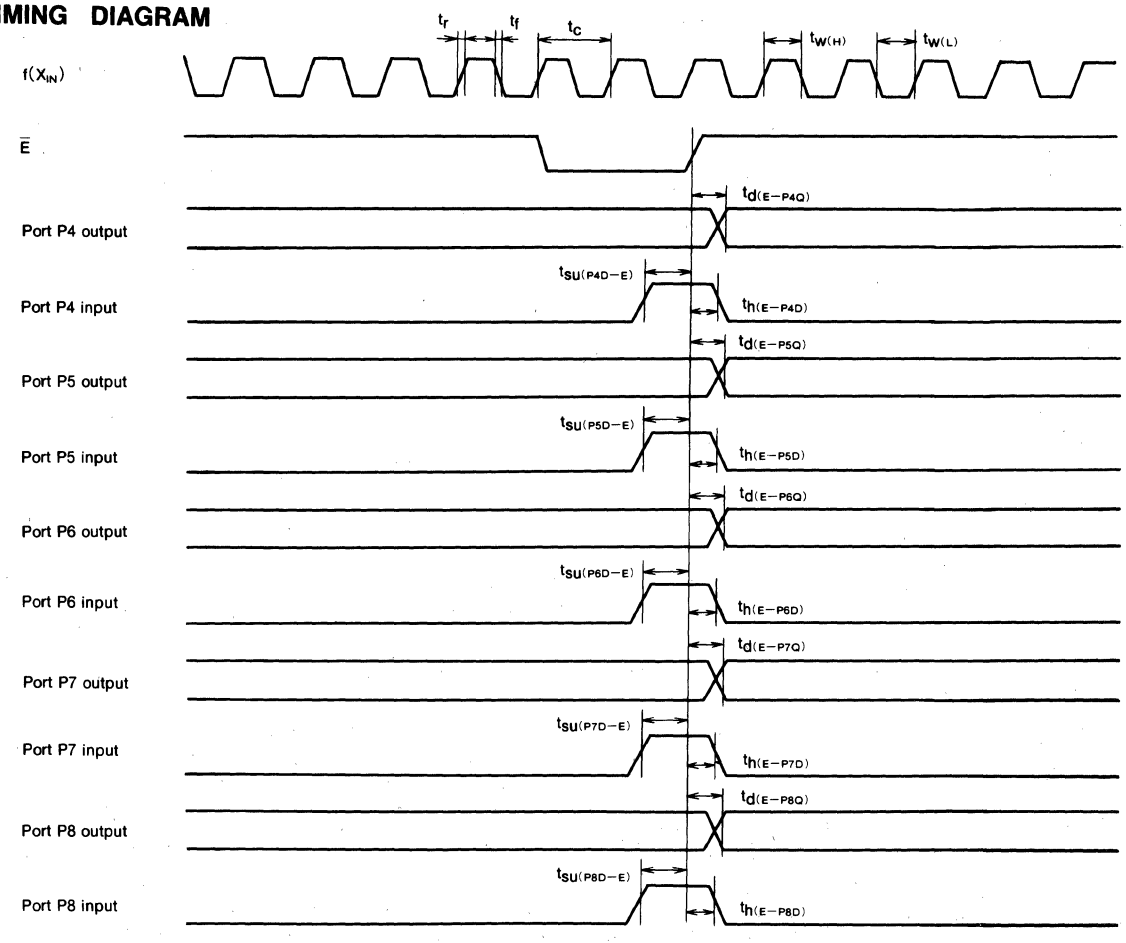
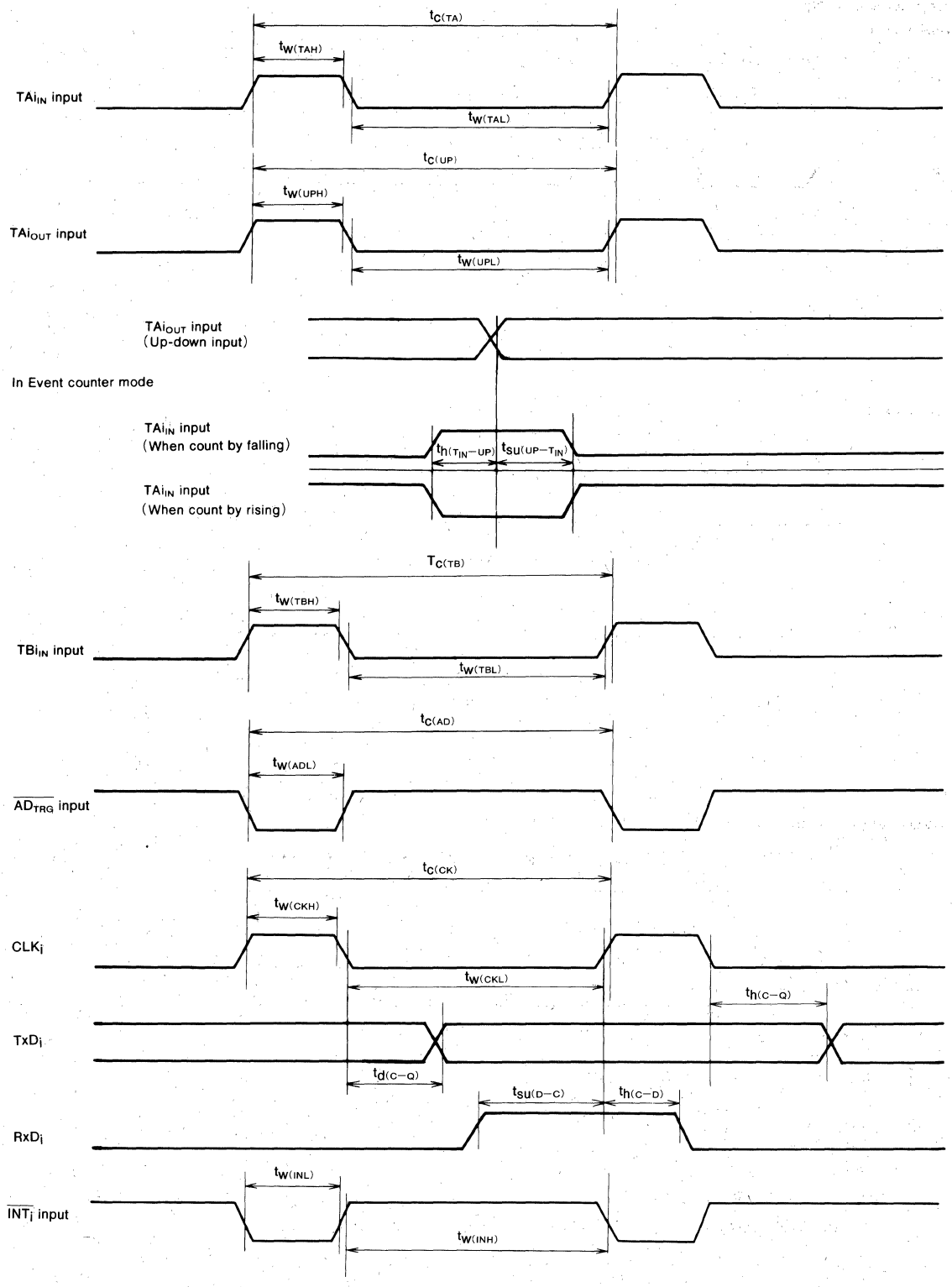


Fig. 63 Testing circuit for each terminal

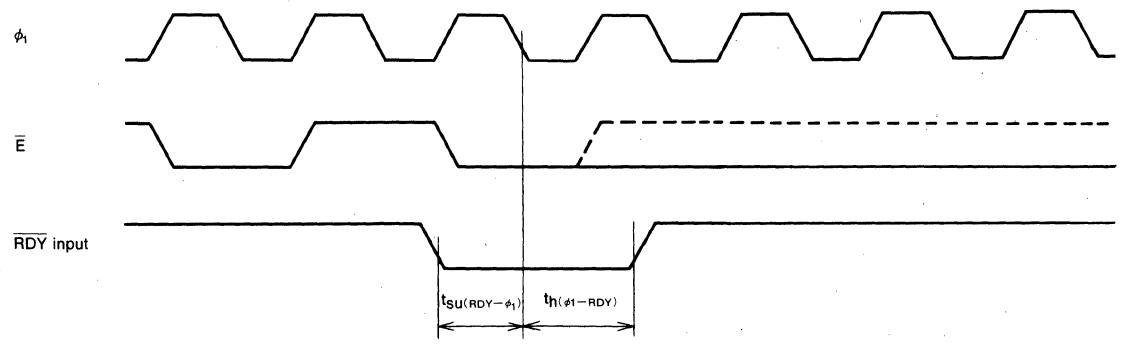
TIMING DIAGRAM



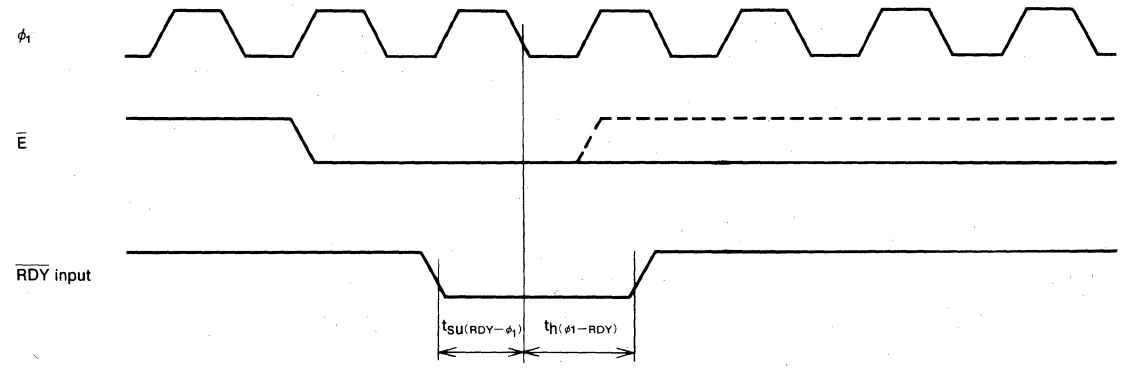

$t_{C(TA)}$
$t_{W(TAH)}$
TAi_{IN} input
$t_{W(TAL)}$
$t_{C(UP)}$
$t_{W(UPH)}$
TAi_{OUT} input
$t_{W(UPL)}$
TAi_{OUT} input
(Up-down input)
In Event counter mode
TAi_{IN} input
(When count by falling)
$t_{h(T_{IN}-UP)}$
$t_{su(UP-T_{IN})}$
TAi_{IN} input
(When count by rising)
$T_{C(TB)}$
$t_{W(TBH)}$
TBi_{IN} input
$t_{W(TBL)}$
$t_{C(AD)}$
$t_{W(ADL)}$
$\overline{AD_{TRG}}$ input
$t_{C(CK)}$
$t_{W(CKH)}$
CLK_i
$t_{W(CKL)}$
$t_{h(C-Q)}$
TxD_i
$t_{d(C-Q)}$
$t_{su(D-C)}$
$t_{h(C-D)}$
RxD_i
$t_{W(INL)}$
$\overline{INT_i}$ input
$t_{W(INH)}$

Microprocessor mode

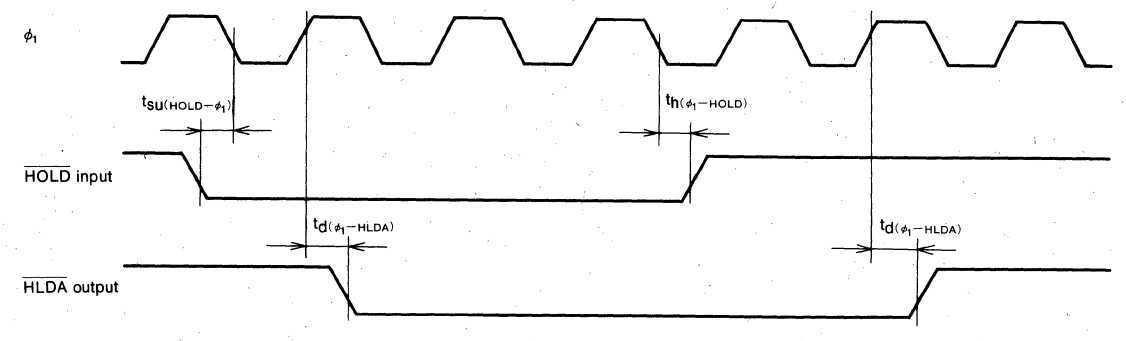
(When wait bit = "1")



(When wait bit = "0")



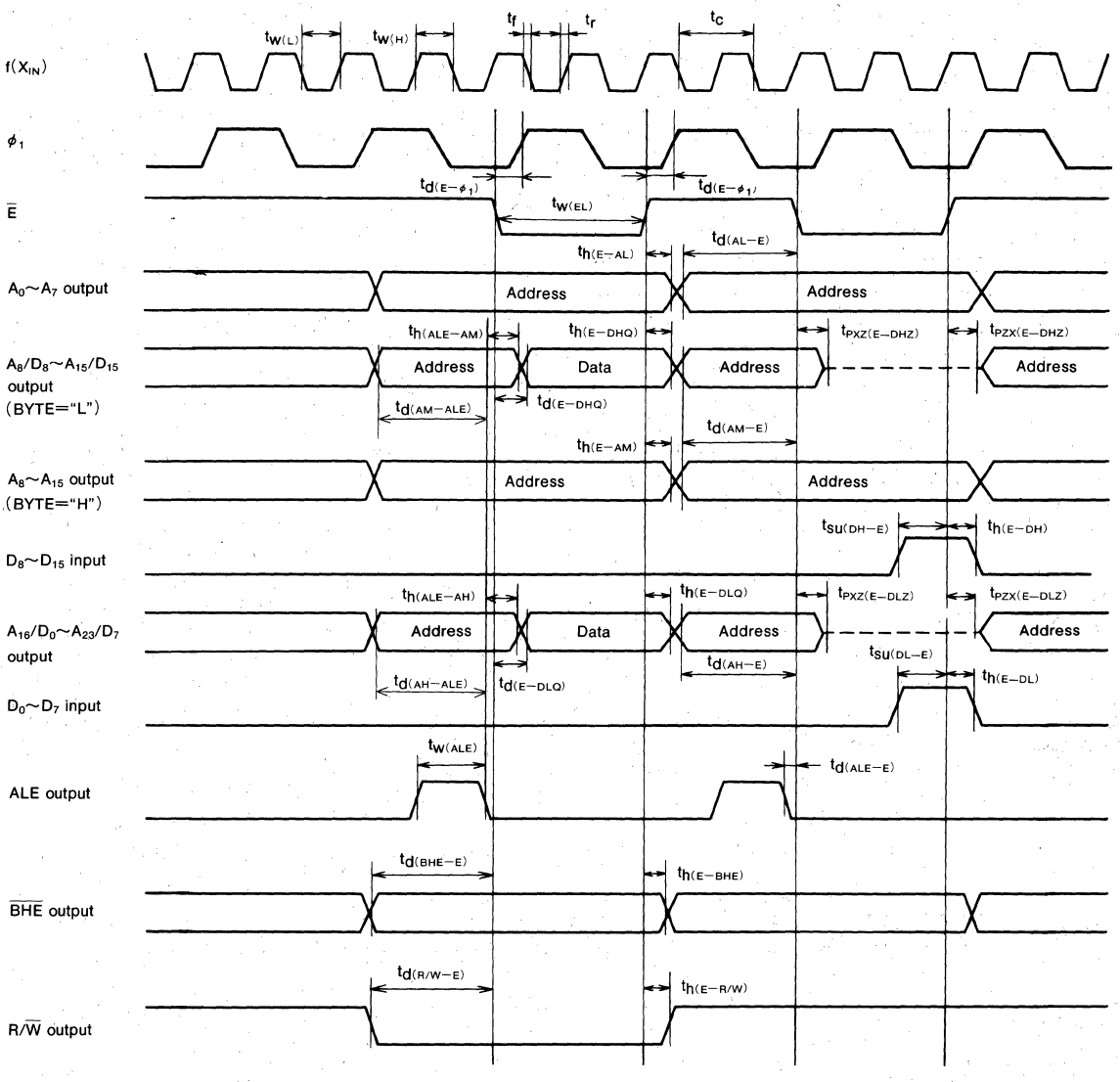
(When wait bit = "1" or "0" in common)



Test conditions

- V_{CC}= 5 V±10%
- Input timing voltage : V_{IL}=1.0V, V_{IH}=4.0V
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V

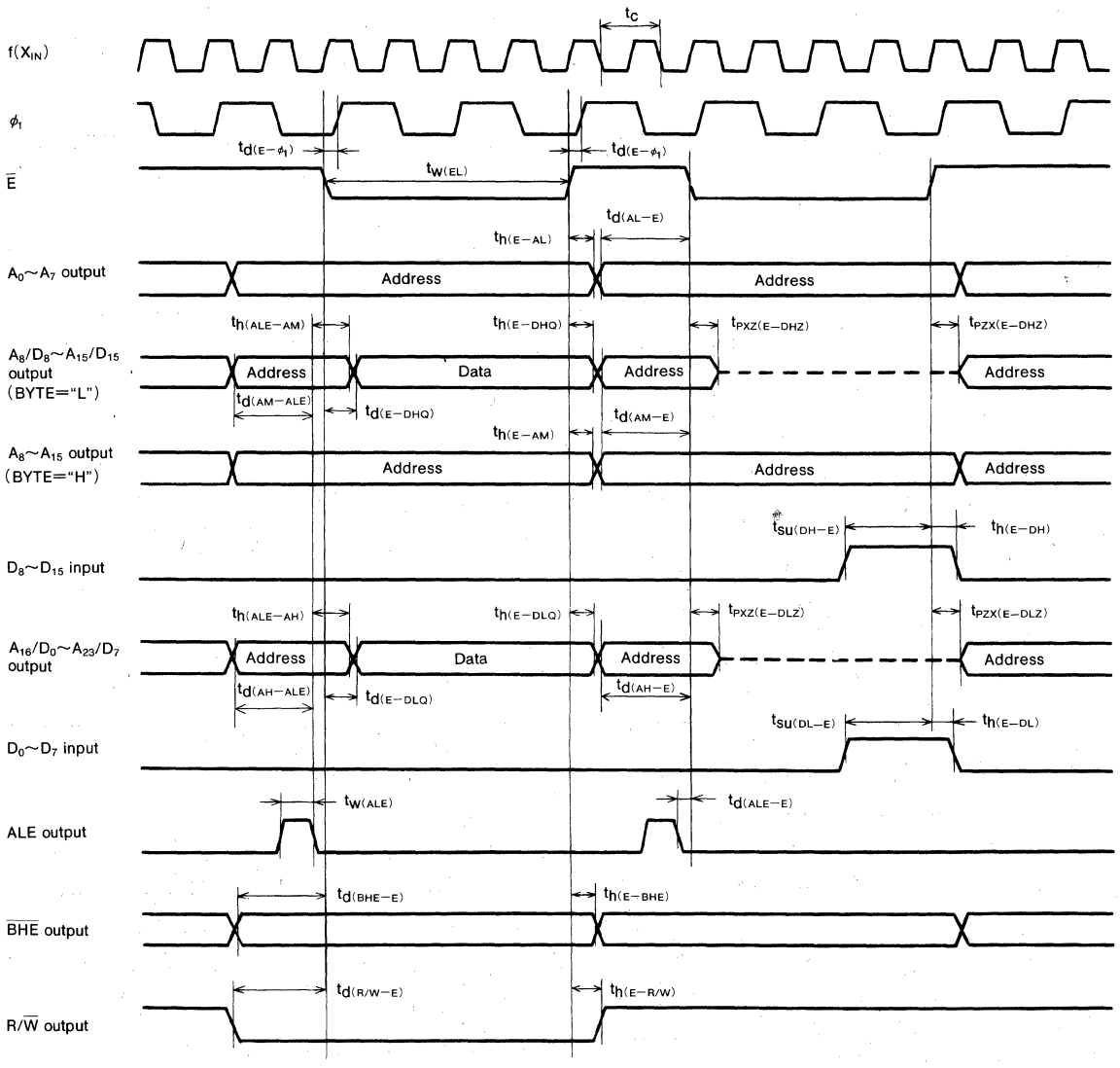
Microprocessor mode (When wait bit = "1")



Test conditions

- V_{CC}= 5 V±10%
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V
- D_0~D_{15} input : V_{IL}=0.8V, V_{IH}=2.5V

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



Test conditions

- V_{CC} = 5 V±10%
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V
- D_0~D_{15} input : V_{IL}=0.8V, V_{IH}=2.5V

ADDRESSING MODES

The MELPS 7700 microcomputers support 28 different addressing modes, offering extremely versatile and powerful memory accessing capability.

When executing an instruction, the address of the memory location from which the data required for arithmetic operation is to be retrieved or to which the result of arithmetic operation is to be stored must be specified address during program execution. Addressing refers to the method of specifying the memory address.

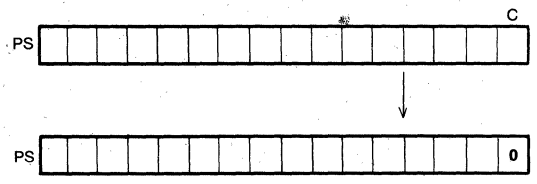
Actual addressing modes are now described by type.

Mode : **Implied addressing mode**

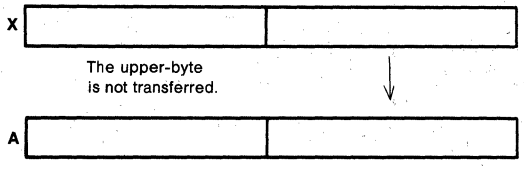
Function : The single-instruction inherently address an internal register.

Instruction : **BRK, CLC, CLI, CLM, CLV, DEX, DEY, INX, INY, NOP, RTI, RTL, RTS, SEC, SEI, SEM, STP, TAD, TAS, TAX, TAY, TBD, TBS, TBX, TBY, TDA, TDB, TSA, TSB, TSX, TXA, TXB, TXS, TXY, TYA, TYB, TYX, WIT, XAB**

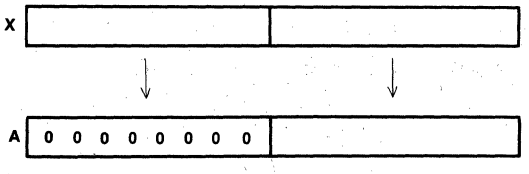
ex. : Mnemonic **CLC** Machine code 18_{16}



ex. : Mnemonic **TXA** **(m = 1, x = 0)** Machine code $8A_{16}$



ex. : Mnemonic **TXA** **(m = 0, x = 1)** Machine code $8A_{16}$



(Note) When the data length differ between the transfer-from and transfer-to locations, data is transferred at the data length for the transfer-to location. If, however, the index register is specified as the transfer-to location and the x flag is set to 1, 0016 is sent as the upper byte value.

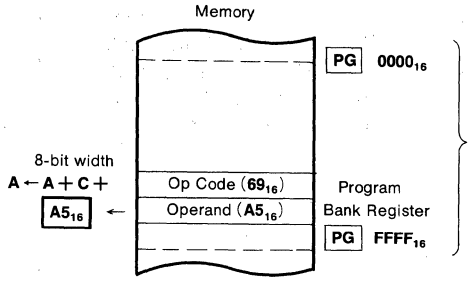
Mode : Immediate addressing mode

Function : A portion of the instruction is the actual data. Such instruction code may cross over the bank boundary.

Instruction : **ADC, AND, CLP, CMP, CPX, CPY, DIV, EOR, LDA, LDT, LDX, LDY, MPY, ORA, RLA, SBC, SEP**

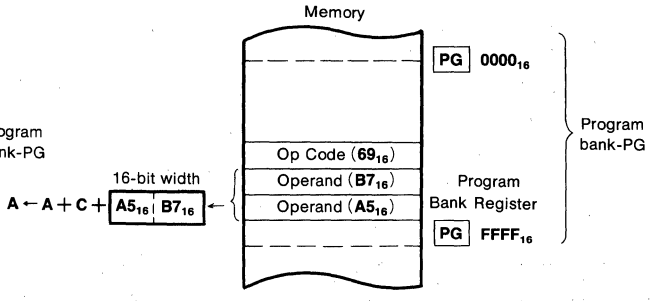
ex. : Mnemonic
ADC A,#0A5H
(m = 1)

Machine code
69_{16} $A5_{16}$



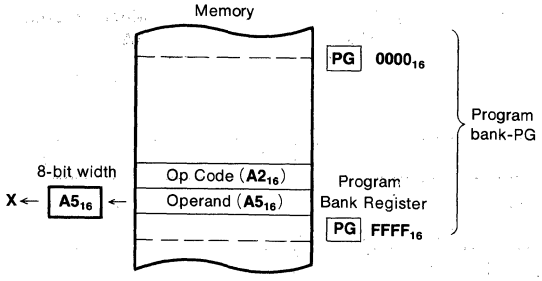
ex. : Mnemonic
ADC A,#0A5B7H
(m = 0)

Machine code
69_{16} $B7_{16}$ $A5_{16}$



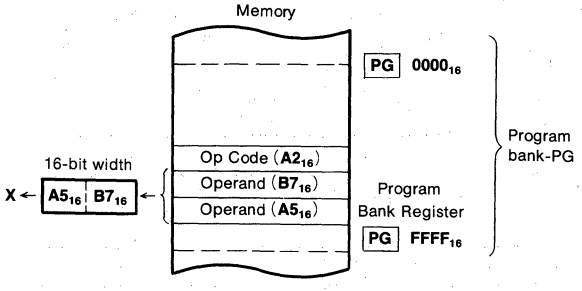
ex. : Mnemonic
LDX #0A5H
(x = 1)

Machine code
$A2_{16}$ $A5_{16}$



ex. : Mnemonic
LDX #0A5B7H
(x = 0)

Machine code
$A2_{16}$ $B7_{16}$ $A5_{16}$



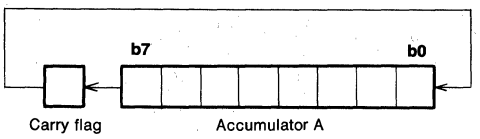
Mode : Accumulator addressing mode

Function : The contents of accumulator are the actual data.

Instruction : **ASL, DEC, INC, LSR, ROL, ROR**

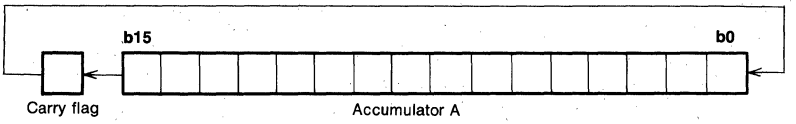
ex. : Mnemonic
ROL A
(m = 1)

Machine code
$2A_{16}$



ex. : Mnemonic
ROL A
(m = 0)

Machine code
$2A_{16}$

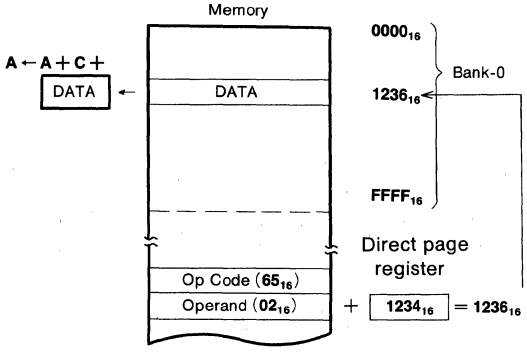


Mode : Direct addressing mode

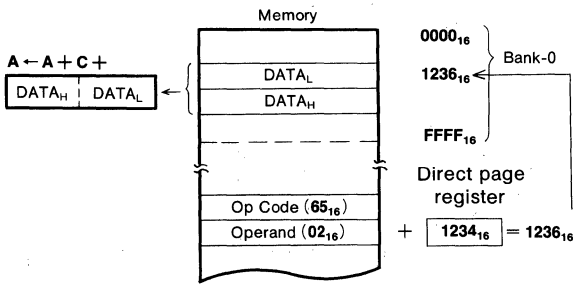
Function : The contents of the bank-0 memory location specified by the result of adding the second byte of the instruction to the contents of the direct page register become the actual data. If, however, addition of the instruction's second byte to the direct page register's contents result in a value that exceeds the bank-0 range, the specified location will be in bank-1

Instruction : **ADC, AND, ASL, CMP, CPX, CPY, DEC, DIV, EOR, INC, LDA, LDM, LDX, LDY, LSR, MPY, ORA, ROL, ROR, SBC, STA, STX, STY**

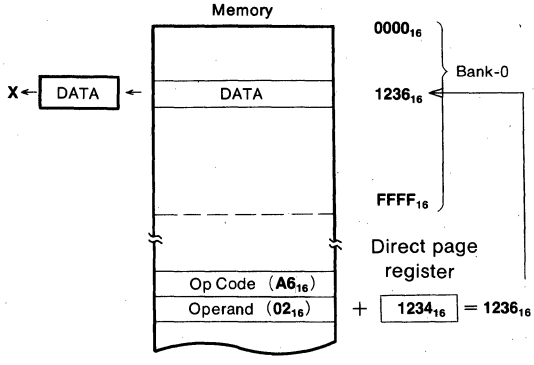
ex. : Mnemonic **ADC A,02H** (**m = 1**) Machine code **65_{16} 02_{16}**



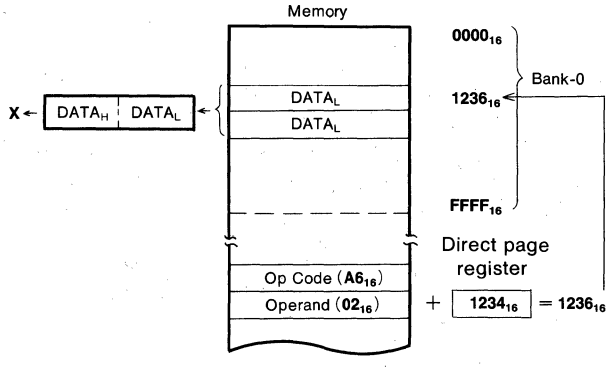
ex. : Mnemonic **ADC A,02H** (**m = 0**) Machine code **65_{16} 02_{16}**



ex. : Mnemonic **LDX 02H** (**x = 1**) Machine code **$A6_{16}$ 02_{16}**



ex. : Mnemonic **LDX 02H** (**x = 0**) Machine code **$A6_{16}$ 02_{16}**



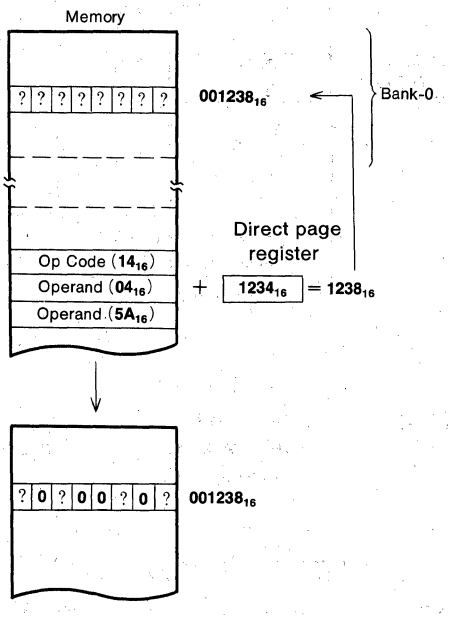
Mode : Direct bit addressing mode

Function : Specifies the bank-0 memory location by the value obtained by adding the instruction's second byte to the direct page register's contents, and specifies the positions of multiple bits in the memory location by the bit pattern in the third and fourth bytes of the instruction (third byte only when the m flag is set to 1). If, however, addition of the instruction's second byte to the direct page register's contents result in a value that exceeds the bank-0 range, the specified location will be in bank-1

Instruction : **CLB, SEB**

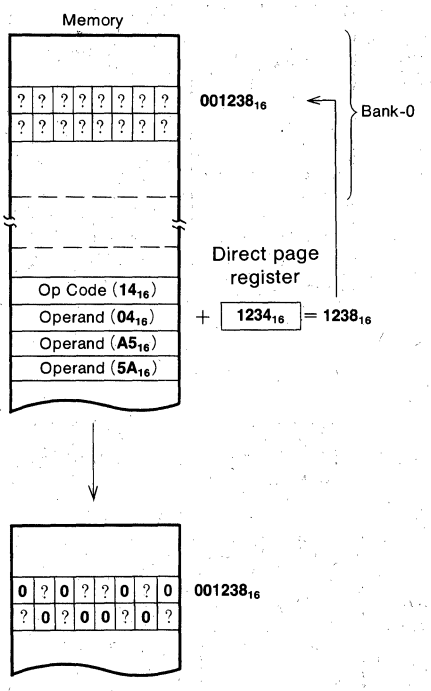
ex. : Mnemonic
CLB #5AH, 04H
(m = 1)

Machine code
14_{16} 04_{16} $5A_{16}$



ex. : Mnemonic
CLB #5AA5H, 04H
(m = 0)

Machine code
14_{16} 04_{16} $A5_{16}$ $5A_{16}$



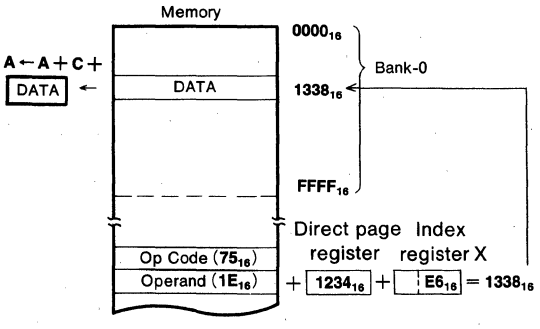
MITSUBISHI ELECTRIC

Mode : **Direct indexed X addressing mode**

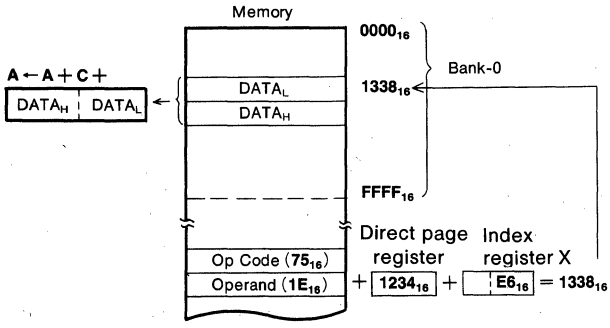
Function : The contents of the bank-0 memory location specified by the result of adding the second byte of the instruction, the contents of the direct page register and the contents of the index register X become the actual data. If, however, addition of the instruction's second byte, the direct page register's contents and the index register X's contents results in a value that exceeds the bank-0 or bank-1 range, the specified location will be in bank-1 or bank-2.

Instruction : **ADC, AND, ASL, CMP, DEC, DIV, EOR, INC, LDA, LDM, LDY, LSR, MPY, ORA, ROL, ROR, SBC, STA, STY**

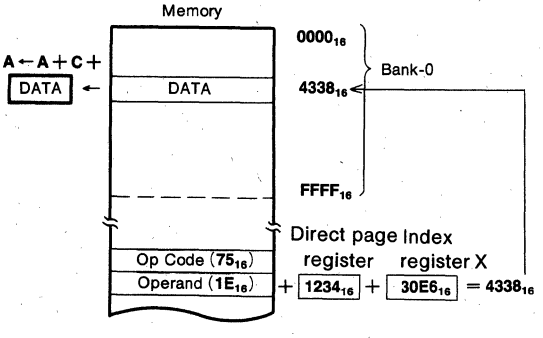
ex. : Mnemonic
ADC A,1EH,X
(m = 1, x = 1)
Machine code
75_{16} $1E_{16}$



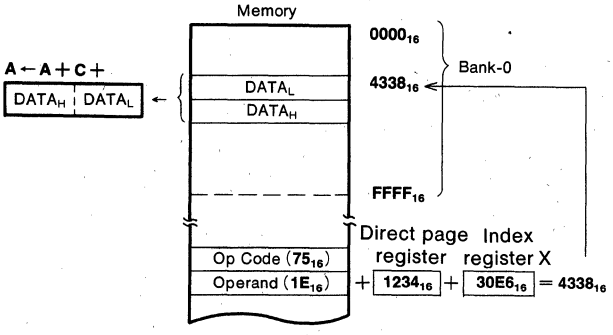
ex. : Mnemonic
ADC A,1EH,X
(m = 0, x = 1)
Machine code
75_{16} $1E_{16}$



ex. : Mnemonic
ADC A,1EH,X
(m = 1, x = 0)
Machine code
75_{16} $1E_{16}$

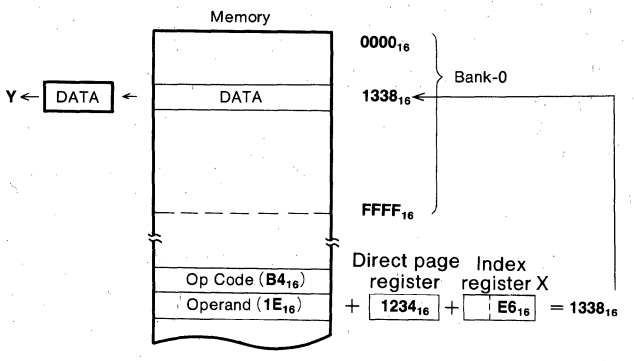


ex. : Mnemonic
ADC A,1EH,X
(m = 0, x = 0)
Machine code
75_{16} $1E_{16}$



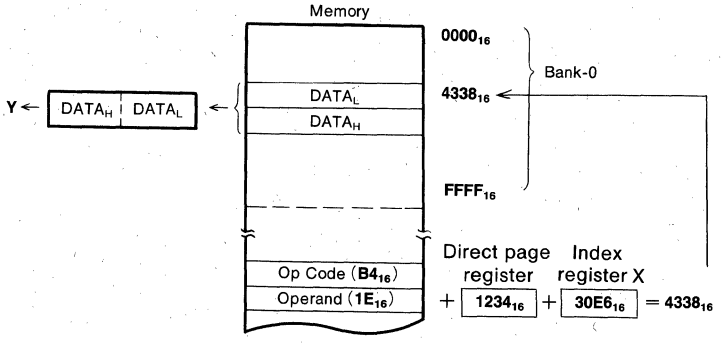
ex. : Mnemonic
LDY 1EH,X
(x = 1)

Machine code
$B4_{16}$ $1E_{16}$



ex. : Mnemonic
LDY 1EH,X
(x = 0)

Machine code
$B4_{16}$ $1E_{16}$



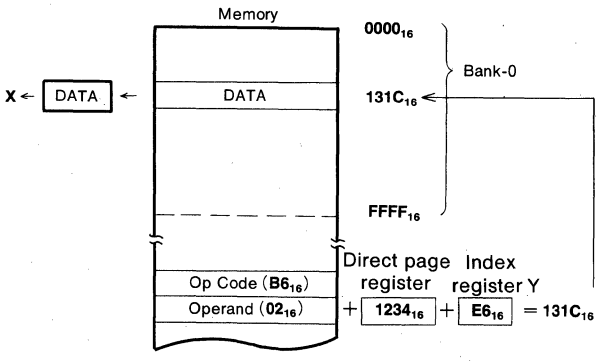
MITSUBISHI ELECTRIC

Mode : **Direct indexed Y addressing mode**

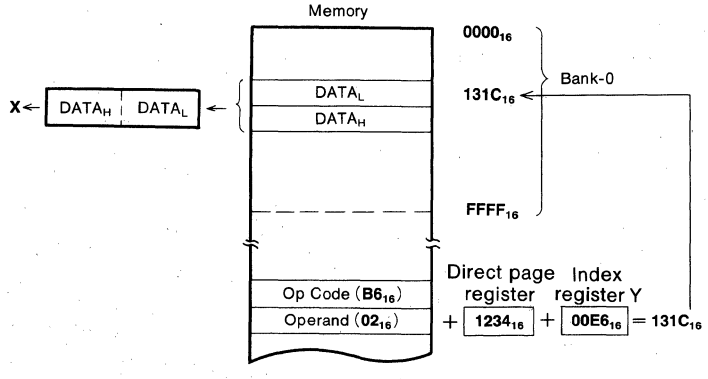
Function : The contents of the bank-0 memory location specified by the result of adding the second byte of the instruction, the contents of the direct page register and the contents of the index register Y become the actual data. If, however, addition of the instruction's second byte, the direct page register's contents and the index register Y's contents results in a value that exceeds the bank-0 or bank-1 range, the specified location will be in bank-1 or bank-2.

Instruction : **LDX, STX**

ex. : Mnemonic — **LDX 02H,Y** (**x = 1**)
Machine code — **$B6_{16}$ 02_{16}**



ex. : Mnemonic — **LDX 02H,Y** (**x = 0**)
Machine code — **$B6_{16}$ 02_{16}**

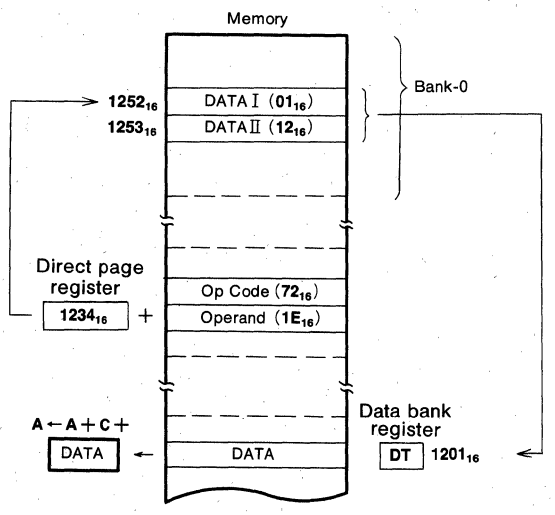


Mode : **Direct indirect addressing mode**

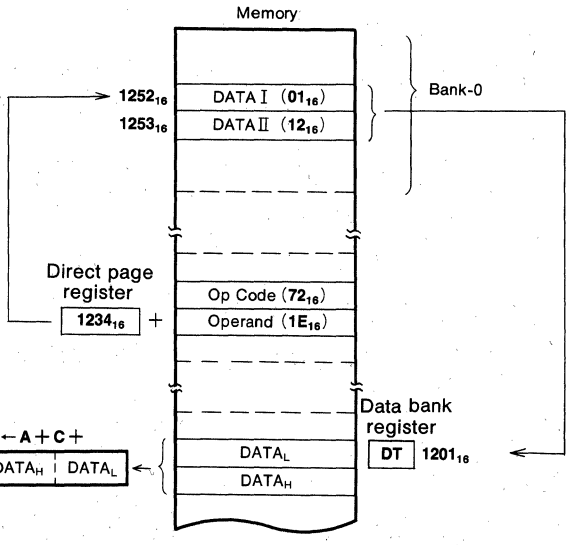
Function : The value obtained by adding the instruction's second byte to the contents of the direct page register specifies 2 adjacent bytes in memory bank-0, and the contents of these bytes in memory bank-DT (DT is contents of data bank register) become the actual data. If, however, the value obtained by adding the instruction's second byte and the direct page register's contents exceeds the bank-0 range, the specified location will be in bank-1.

Instruction : **ADC, AND, CMP, DIV, EOR, LDA, MPY, ORA, SBC, STA**

ex. : Mnemonic **ADC A,(1EH) (m = 1)** Machine code 72_{16} $1E_{16}$



ex. : Mnemonic **ADC A,(1EH) (m = 0)** Machine code 72_{16} $1E_{16}$



MITSUBISHI ELECTRIC

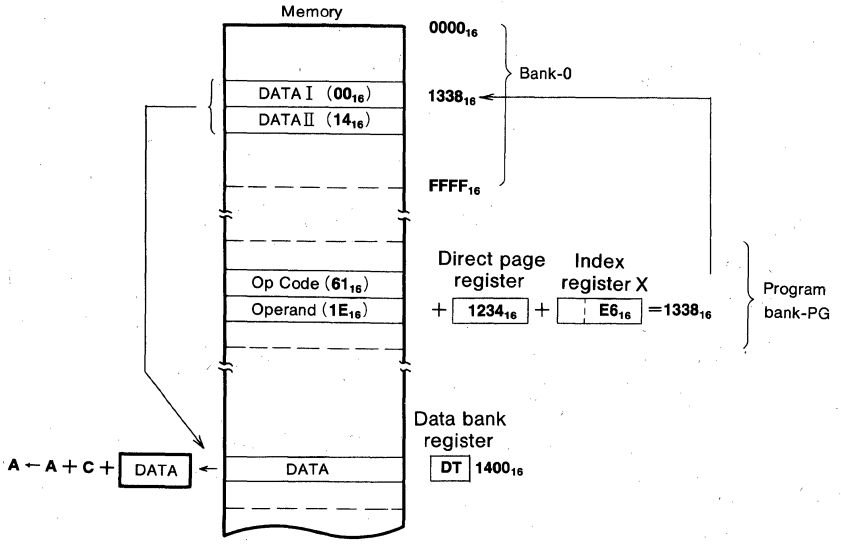
Mode : Direct indexed X indirect addressing mode

Function : The value obtained by adding the instruction's second byte, the contents of the direct page register and the contents of the index register X specifies 2 adjacent bytes in memory bank-0, and the contents of these bytes in memory bank-0, and the contents of these bytes in memory bank-DT (DT is contents of data bank register) become the actual data. If, however, the value obtained by adding the instruction's second byte, the direct page register's contents and the index register X's contents exceeds the bank-0 or bank-1 range, the specified location will be in bank-1 or bank-2

Instruction : **ADC, AND, CMP, DIV, EOR, LDA, MPY, ORA, SBC, STA**

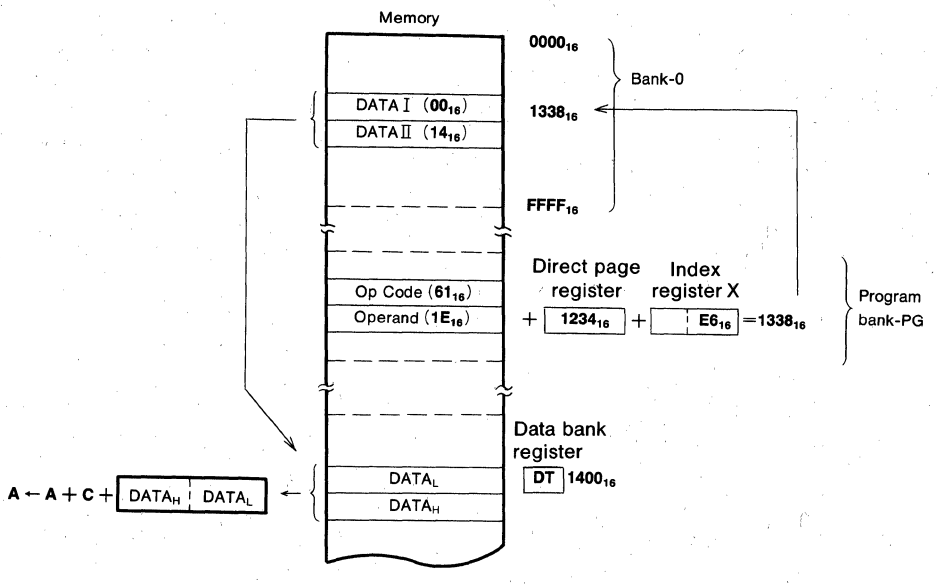
ex. : Mnemonic
ADC A, (1EH, X)
(m = 1, x = 1)

Machine code
61_{16} $1E_{16}$



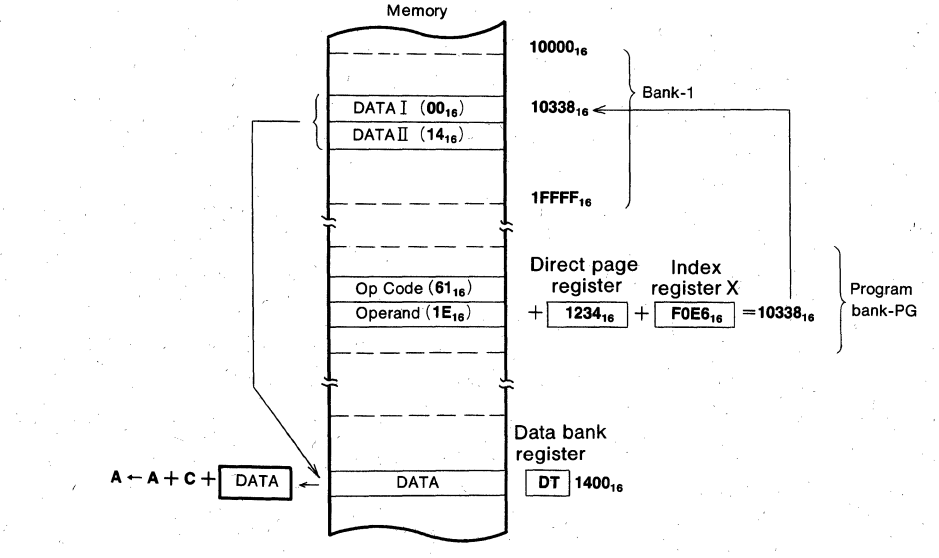
ex. : Mnemonic
ADC A, (1EH, X)
(m = 0, x = 1)

Machine code
61_{16} $1E_{16}$



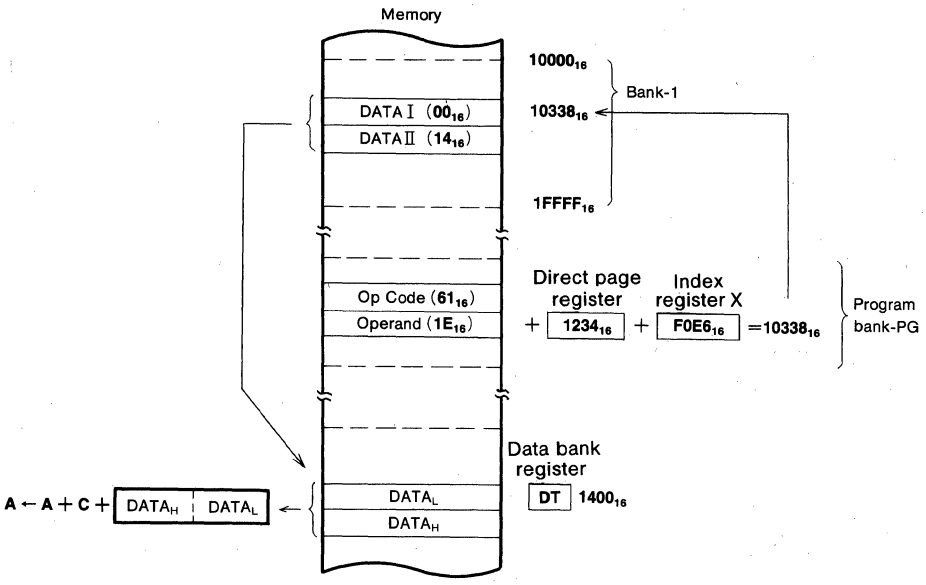
ex. : Mnemonic
ADC A, (1EH, X)
(m = 1, x = 0)

Machine code
61_{16} $1E_{16}$



ex. : Mnemonic
ADC A, (1EH, X)
(m = 0, x = 0)

Machine code
61_{16} $1E_{16}$



Mode : Direct indirect indexed Y addressing mode

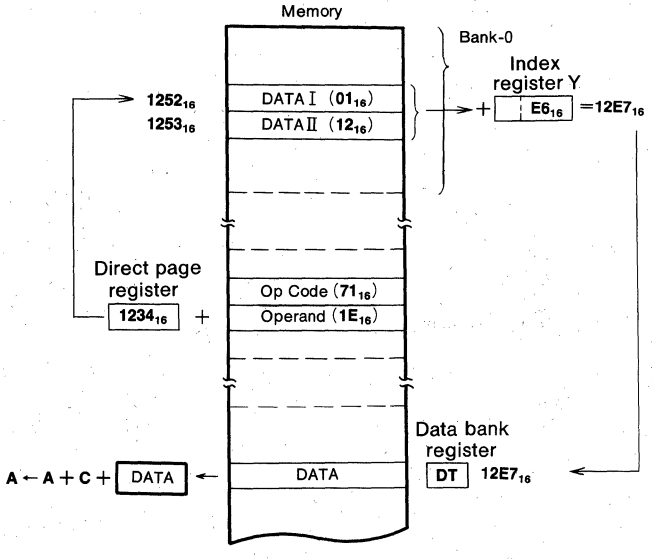
Function : The value obtained by adding the instruction's second byte and the contents of the direct page register specifies 2 adjacent bytes in memory bank-0.
The value obtained by adding the contents of these bytes and the contents of the index register Y specifies address of the actual data in memory bank-DT (DT is contents of data bank register). If, however, the value obtained by adding the contents of the instruction's second byte and the direct page register exceeds the bank-0 range, the specified location will be in bank-1. Also, if addition of the contents of memory and index register Y generate a carry, the bank number will be 1 larger than the contents of the data bank register.

Instruction : **ADC, AND, CMP, DIV, EOR, LDA, MPY, ORA, SBC, STA**

ex. : Mnemonic — Machine code

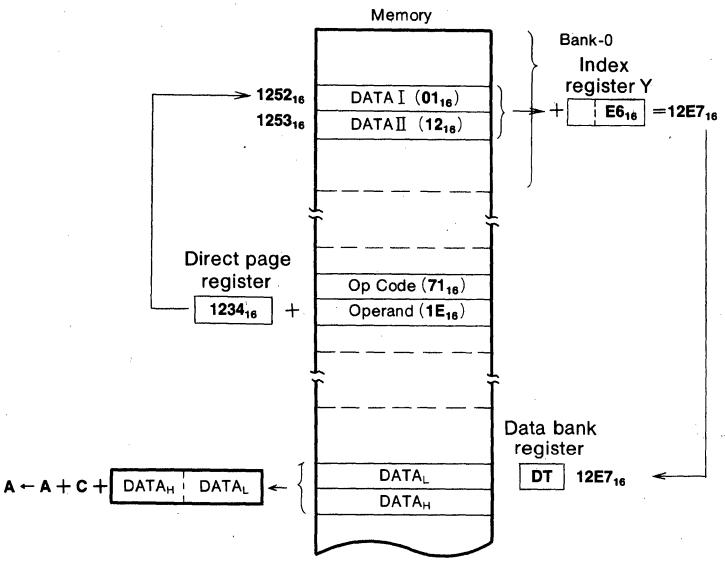
ADC A, (1EH),Y — 71_{16} $1E_{16}$

(m = 1, x = 1)



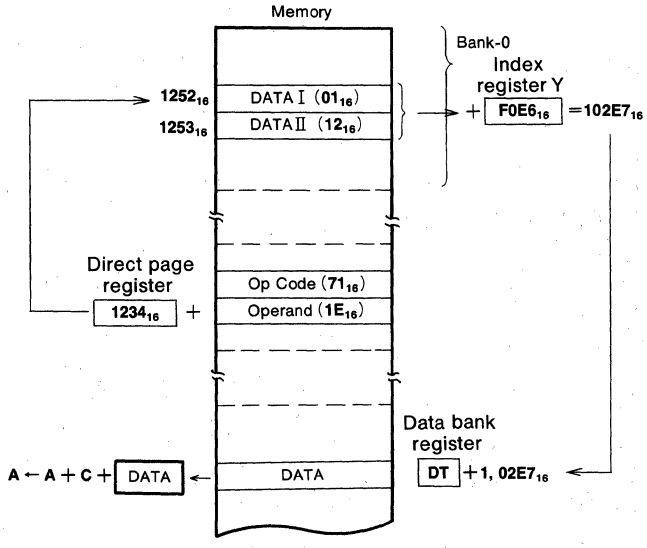
ex. : Mnemonic
ADC A, (1EH), Y
(m = 0, x = 1)

Machine code
71_{16} $1E_{16}$



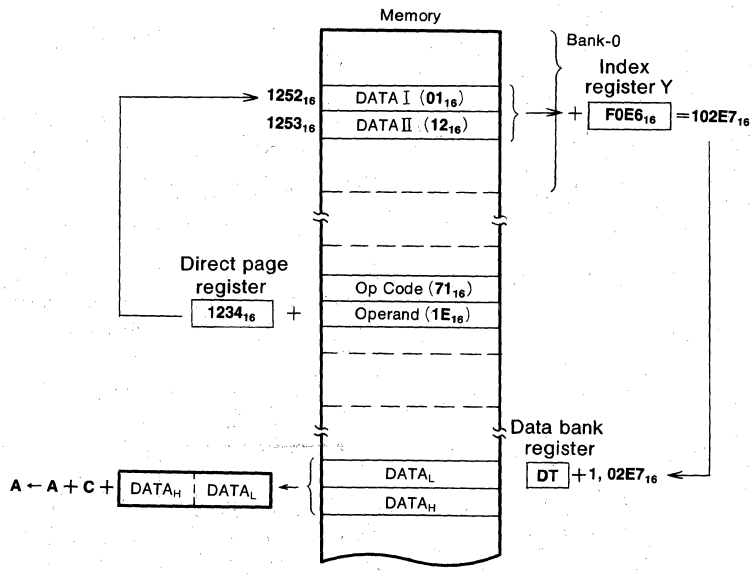
ex. : Mnemonic
ADC A, (1EH), Y
(m = 1, x = 0)

Machine code
71_{16} $1E_{16}$



ex. : Mnemonic
ADC A, (1EH), Y
(m = 0, x = 0)

Machine code
71_{16} $1E_{16}$

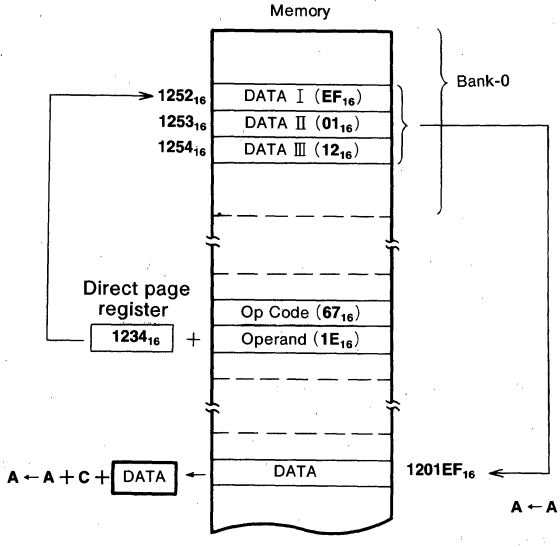


Mode : **Direct indirect long addressing mode**

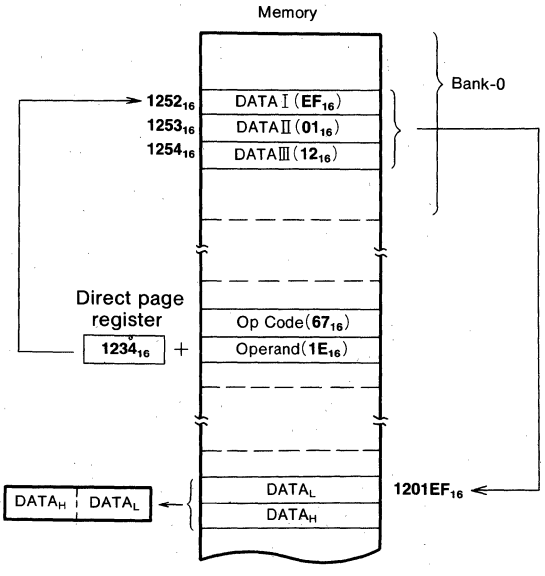
Function : The value obtained by adding the instruction's second byte and the contents of the direct page register specifies 3 adjacent bytes in memory bank-0, and the contents of these bytes specify the address of the memory location that contains the actual data. If, however, the value obtained by adding the contents of the instruction's second byte and the direct page register exceeds the bank-0 range, the specified location will be in bank-1. The 3 adjacent bytes memory location may be spread over two different banks.

Instruction : **ADC, AND, CMP, DIV, EOR, LDA, MPY, ORA, SBC, STA**

ex. : Mnemonic **ADCL A, (1EH) (m=1)** Machine code 67_{16} $1E_{16}$



ex. : Mnemonic **ADCL A, (1EH) (m=0)** Machine code 67_{16} $1E_{16}$



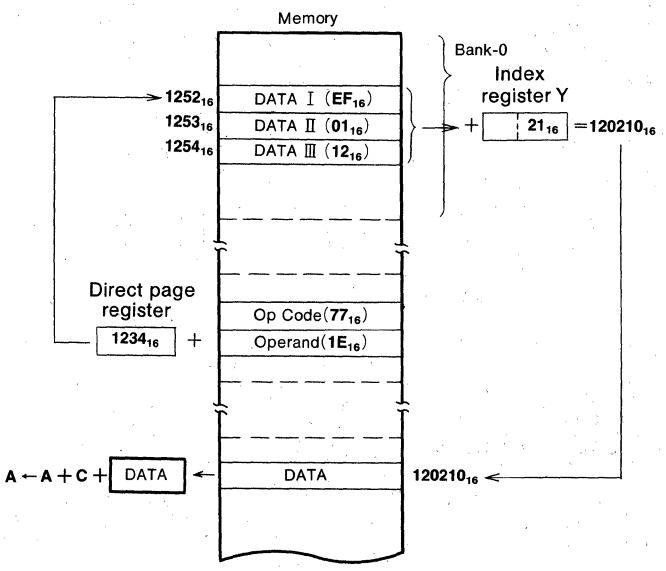
Mode : Direct indirect long indexed Y addressing mode

Function : The value obtained by adding the instruction's second byte and the contents of the direct page register specifies 3 adjacent bytes in memory bank-0, and the value obtained by adding the contents of these bytes and the contents of the index register Y specifies the address of the memory location where the actual data is stored. If, however, the value obtained by adding the contents of the instruction's second byte and the direct page register exceeds the bank-0 range, the specified location will be in bank-1. The 3 adjacent bytes memory location may be spread over two different banks.

Instruction : **ADC, AND, CMP, DIV, EOR, LDA, MPY, ORA, SBC, STA**

ex. : Mnemonic
ADCL A,(1EH), Y
(m=1, x=1)

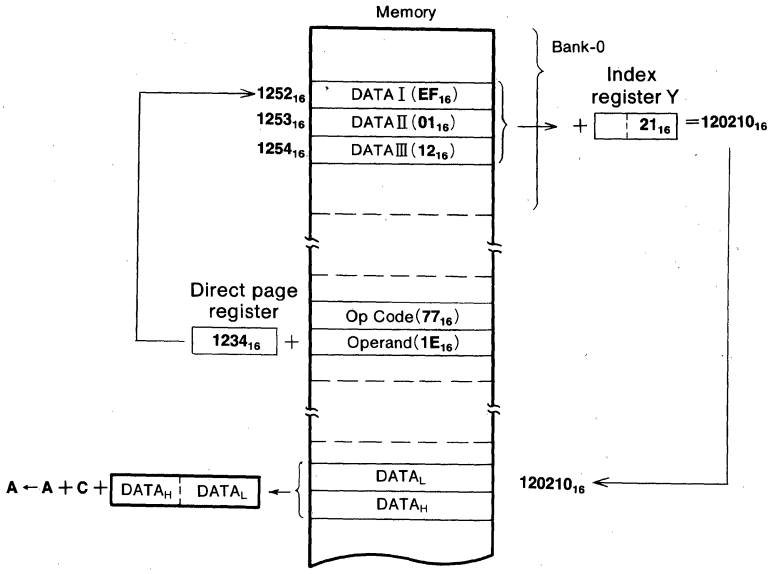
Machine code
77_{16} $1E_{16}$



MITSUBISHI ELECTRIC

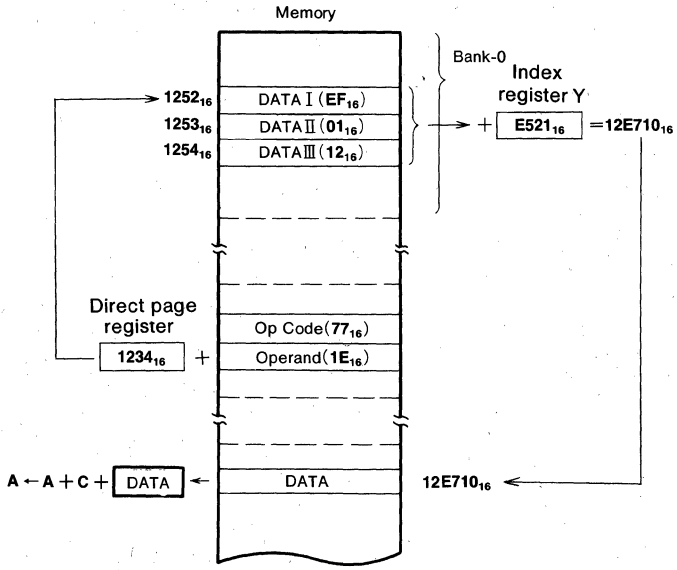
ex. : Mnemonic
ADCL A,(1EH), Y
(m=0, x=1)

Machine code
77_{16} $1E_{16}$



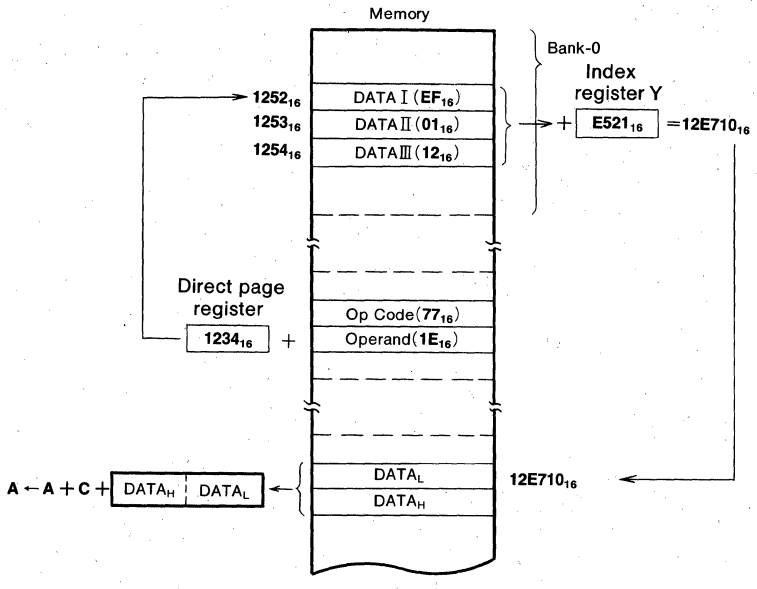
ex. : Mnemonic
ADCL A,(1EH), Y
(m=1, x=0)

Machine code
77_{16} $1E_{16}$



ex. : Mnemonic
ADCL A,(1EH), Y
(m=0, x=0)

Machine code
77_{16} $1E_{16}$



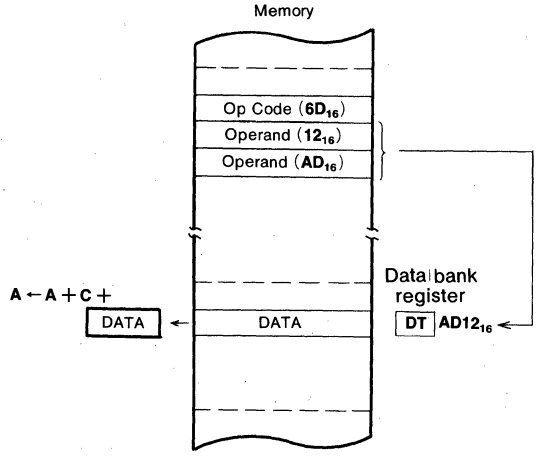
MITSUBISHI ELECTRIC

Mode : **Absolute addressing mode**

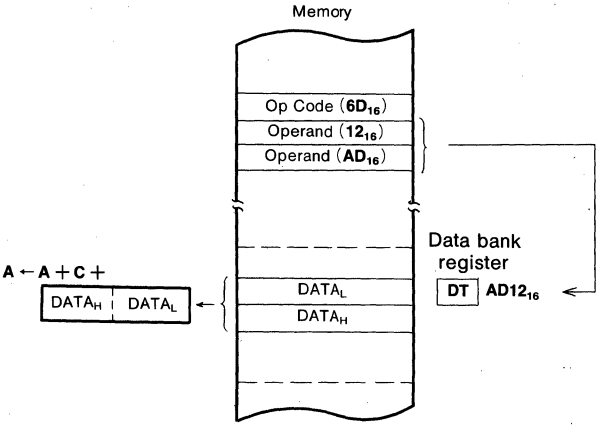
Function : The contents of the memory locations specified by the instruction's second and third bytes and the contents of the data bank register are the actual data. Note that, in the cases of the JMP and JSR instructions, the instructions' second and third byte contents are transferred to the program counter.

Instruction : **ADC, AND, ASL, CMP, CPX, CPY, DEC, DIV, EOR, INC, JMP, JSR, LDA, LDM, LDX, LDY, LSR, MPY, ORA, ROL, ROR, SBC, STA, STX, STY**

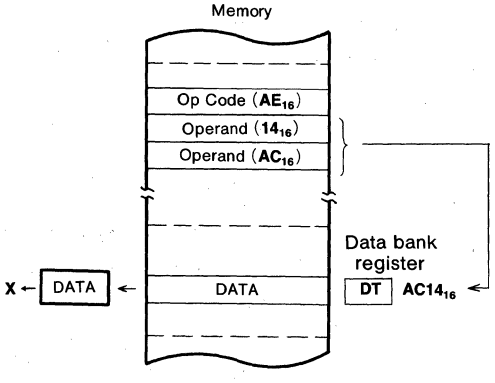
ex. : Mnemonic **ADC A, 0AD12H** (**m=1**) Machine code $6D_{16}$ 12_{16} AD_{16}



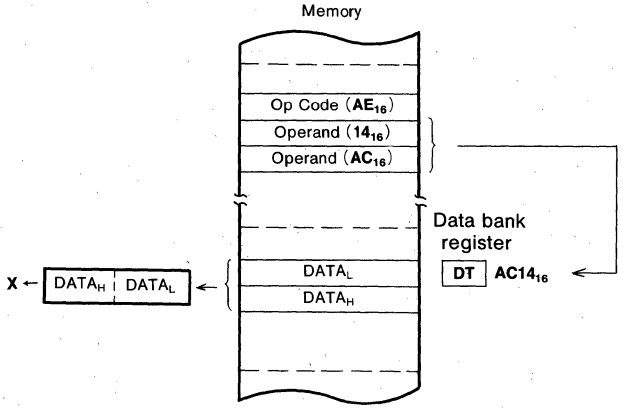
ex. : Mnemonic **ADC A, 0AD12H** (**m=0**) Machine code $6D_{16}$ 12_{16} AD_{16}



ex. : Mnemonic **LDX 0AC14H** (**x=1**) Machine code AE_{16} 14_{16} AC_{16}

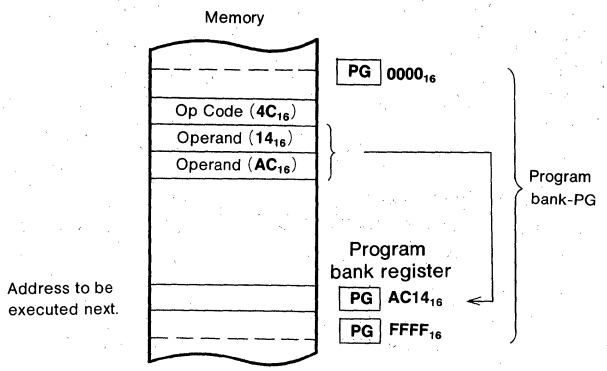


ex. : Mnemonic **LDX 0AC14H** (**x=0**) Machine code AE_{16} 14_{16} AC_{16}



ex. : Mnemonic — Machine code

JMP 0AC14H — **$4C_{16}$ 14_{16} AC_{16}**



Program bank register contents are not affected.

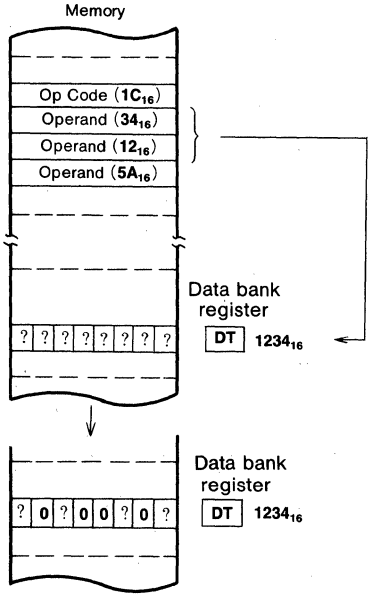
Mode : Absolute bit addressing mode

Function : The contents of the instruction's second and third bytes and the contents of the data bank register specify the memory locations, and data for multiple bit positions in the memory locations are specified by a bit pattern specified in the instruction's fourth and fifth bytes (the fourth byte only if the m flag is set to 1).

Instruction : **CLB, SEB**

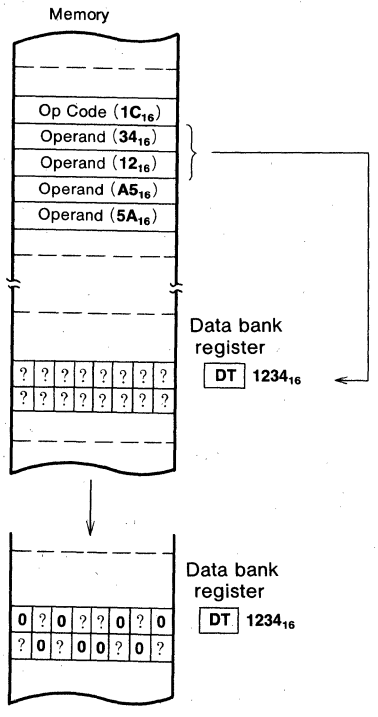
ex. : Mnemonic
CLB #5AH, 1234H
(m=1)

Machine code
$1C_{16}$ 34_{16} 12_{16} $5A_{16}$



ex. : Mnemonic
CLB #5AA5H, 1234H
(m=0)

Machine code
$1C_{16}$ 34_{16} 12_{16} $A5_{16}$ $5A_{16}$



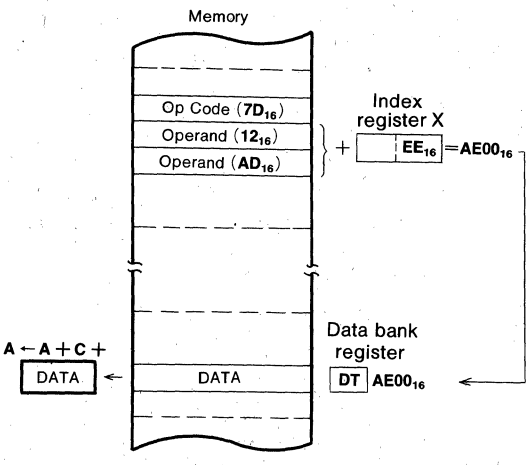
Mode : Absolute indexed X addressing mode

Function : The contents of the memory locations specified by a value resulting from addition of a 16-bit numeric value expressed by the instruction's second and third bytes with the contents of the index register X and the contents of the data bank register are the actual data. If, however, addition of the numeric value expressed by the instruction's second and third bytes with the contents of the index register X generates a carry, the bank number will be 1 larger than the contents of the data bank register.

Instruction : **ADC ,AND ASL, CMP, DEC, DIV, EOR, INC, LDA, LDM, LDY, LSR, MPY, ORA, ROL, ROR, SBC, STA**

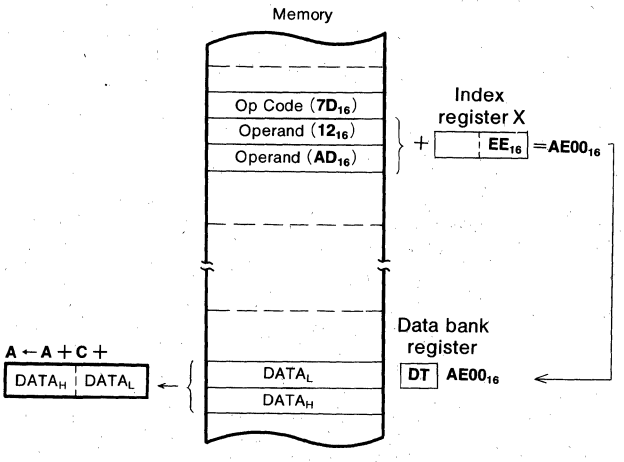
ex. : Mnemonic
ADC A, 0AD12H, X
(m=1, x=1)

Machine code
$7D_{16}$ 12_{16} AD_{16}

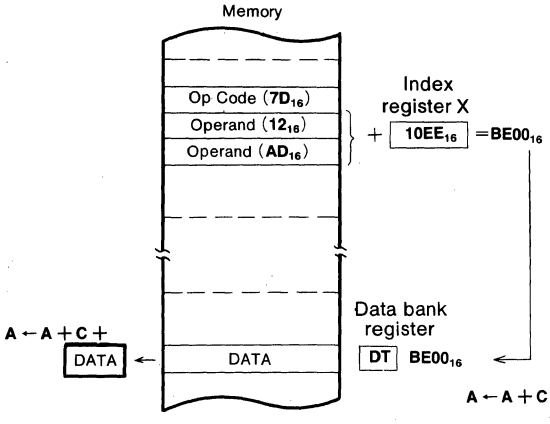


ex. : Mnemonic
ADC A, 0AD12H, X
(m=0, x=1)

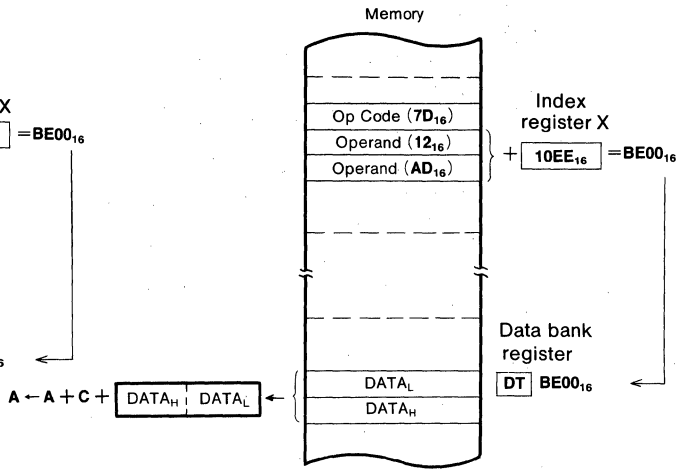
Machine code
$7D_{16}$ 12_{16} AD_{16}



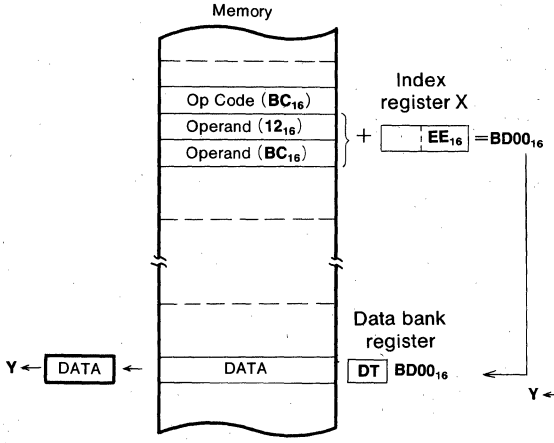
ex. : Mnemonic **ADC A, 0AD12H, X** (**m=1, x=0**) Machine code $7D_{16}\ 12_{16}\ AD_{16}$



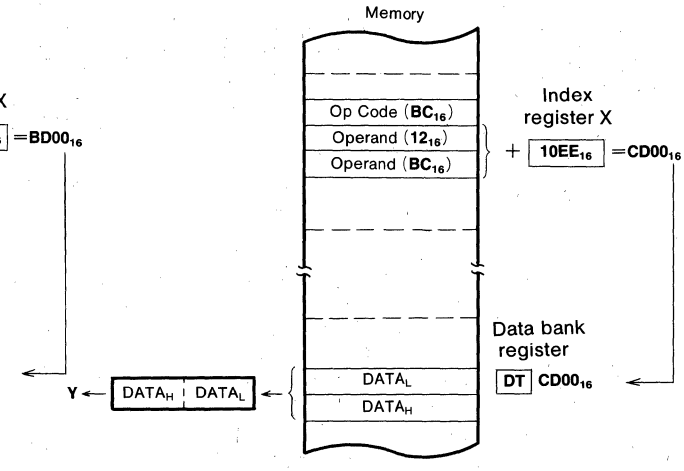
ex. : Mnemonic **ADC A, 0AD12H, X** (**m=0, x=0**) Machine code $7D_{16}\ 12_{16}\ AD_{16}$



ex. : Mnemonic **LDY 0BC12H, X** (**x=1**) Machine code $BC_{16}\ 12_{16}\ BC_{16}$



ex. : Mnemonic **LDY 0BC12H, X** (**x=0**) Machine code $BC_{16}\ 12_{16}\ BC_{16}$

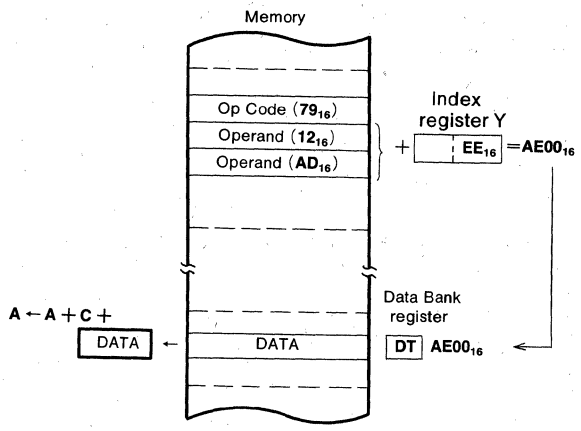


Mode : Absolute indexed Y addressing mode

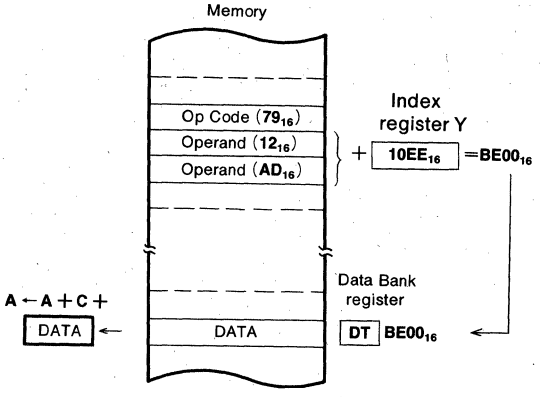
Function : The contents of the memory locations specified by a value resulting from addition of a 16-bit numeric value expressed by the instruction's second and third bytes with the contents of the index register Y and the contents of the data bank register are the actual data. If, however, addition of the numeric value expressed by the instruction's second and third bytes with the contents of the index register Y generates a carry, the bank number will be 1 larger than the contents of the data bank register.

Instruction : **ADC, AND, CMP, DIV, EOR, LDA, LDX, MPY, ORA, SBC, STA**

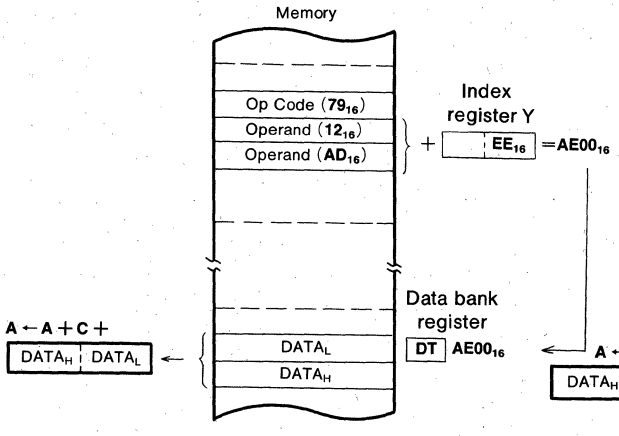
ex. : Mnemonic **ADC A, 0AD12H, Y** (**m=1, x=1**) Machine code 79_{16} 12_{16} AD_{16}



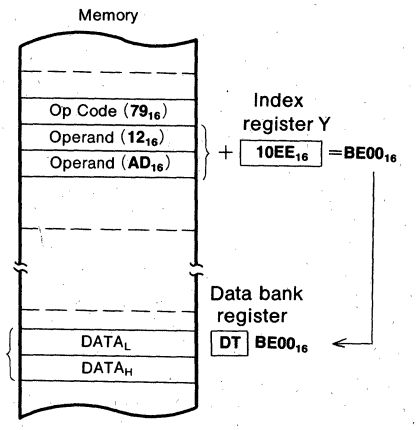
ex. : Mnemonic **ADC A, 0AD12H, Y** (**m=1, x=0**) Machine code 79_{16} 12_{16} AD_{16}



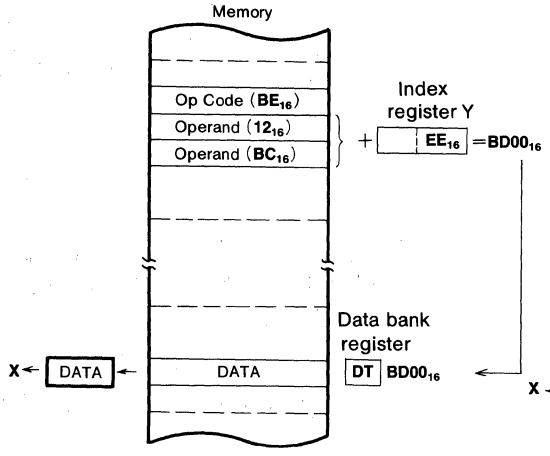
ex. : Mnemonic **ADC A, 0AD12H, Y** (**m=0, x=1**) Machine code 79_{16} 12_{16} AD_{16}



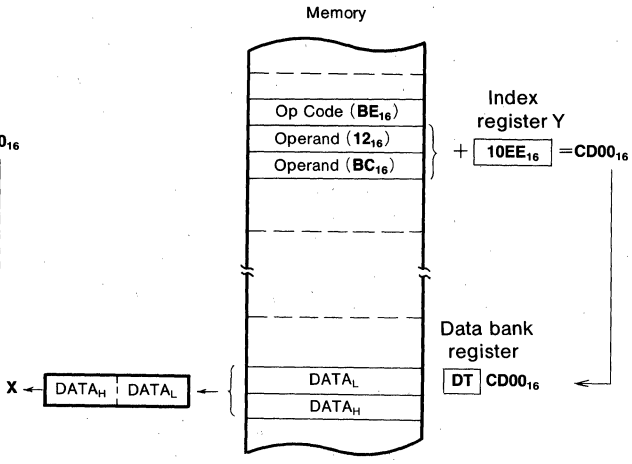
ex. : Mnemonic **ADC A, 0AD12H, Y** (**m=0, x=0**) Machine code 79_{16} 12_{16} AD_{16}



ex. : Mnemonic **LDX 0BC12H, Y** (x=1) Machine code BE_{16} 12_{16} BC_{16}



ex. : Mnemonic **LDX 0BC12H, Y** (x=0) Machine code BE_{16} 12_{16} BC_{16}



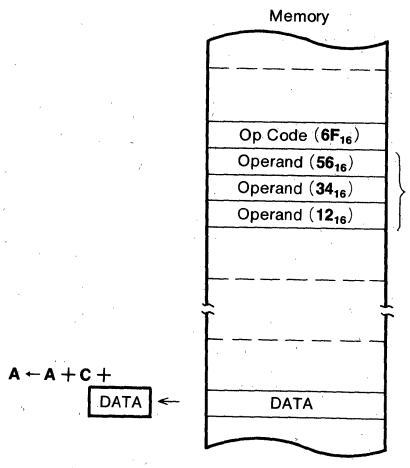
Mode : Absolute long addressing mode

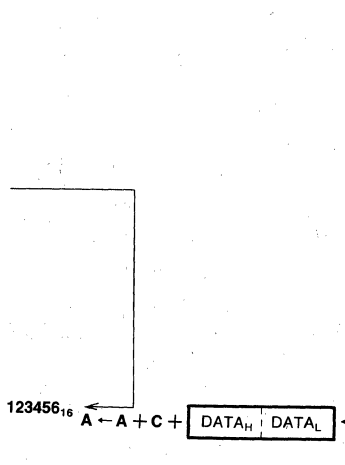
Function : The contents of the memory locations specified by the instruction's second, third and fourth bytes become the actual data.Note that, in the cases of the JMP and JSR instructions, the instructions' second and third byte contents are transferred to the program counter and the fourth byte contents are transferred to the program bank register.

Instruction : **ADC, AND, CMP, DIV, EOR, JMP, JSR, LDA, MPY, ORA, SBC, STA**

ex. : Mnemonic
ADC A, 123456H
(m=1)

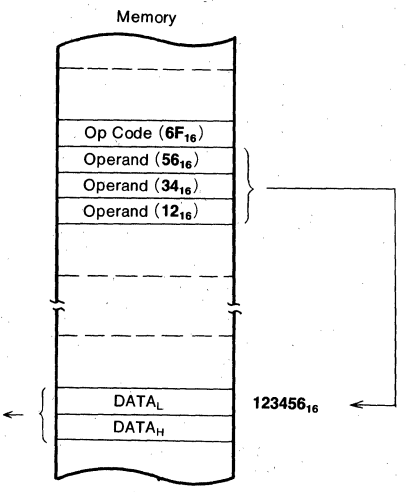
Machine code
$6F_{16}$ 56_{16} 34_{16} 12_{16}





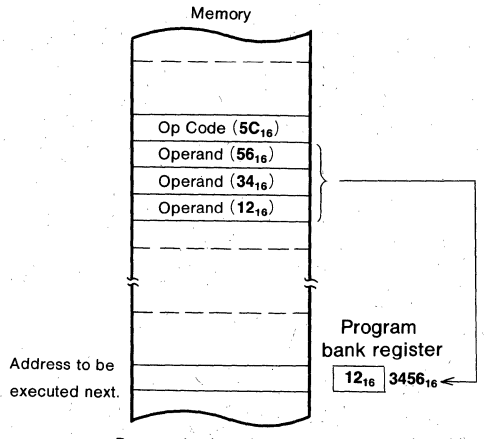
ex. : Mnemonic
ADC A, 123456H
(m=0)

Machine code
$6F_{16}$ 56_{16} 34_{16} 12_{16}



ex. : Mnemonic
JMP 123456H

Machine code
$5C_{16}$ 56_{16} 34_{16} 12_{16}



Program bank register contents are replaced by the third operand.

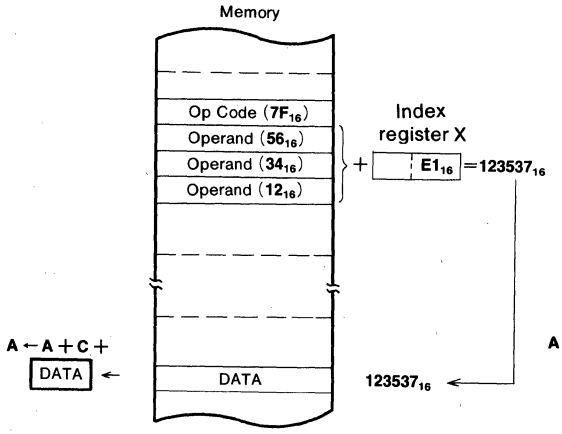
MITSUBISHI ELECTRIC

Mode : Absolute long indexed X addressing mode

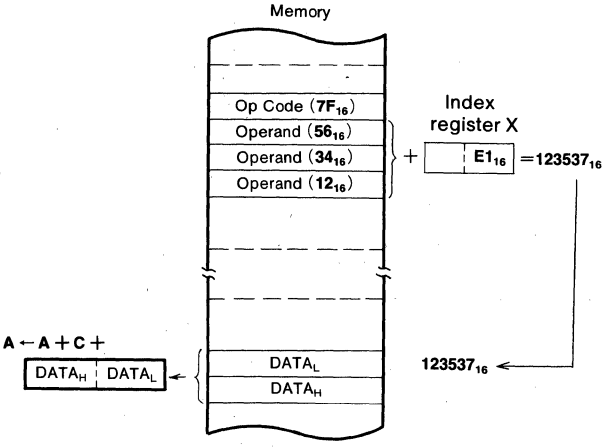
Function : The contents of the memory location specified by adding the numeric value expressed by the instruction's second, third and fourth bytes with the contents of the index register X are the actual data.

Instruction : **ADC, AND, CMP, DIV, EOR, LDA, MPY, ORA, SBC, STA**

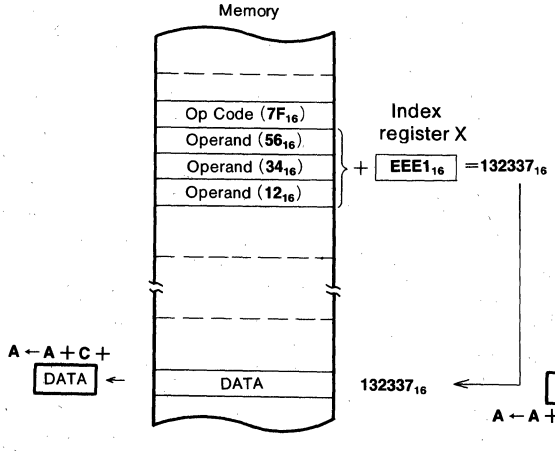
ex. : Mnemonic **ADC A, 123456H, X** **(m=1, x=1)** Machine code $7F_{16}$ 56_{16} 34_{16} 12_{16}



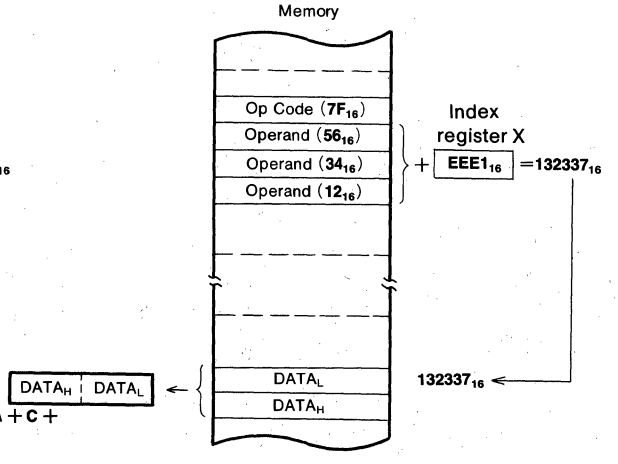
ex. : Mnemonic **ADC A, 123456H, X** **(m=0, x=1)** Machine code $7F_{16}$ 56_{16} 34_{16} 12_{16}



ex. : Mnemonic **ADC A, 123456H, X** **(m=0, x=1)** Machine code $7F_{16}$ 56_{16} 34_{16} 12_{16}



ex. : Mnemonic **ADC A, 123456H, X** **(m=0, x=0)** Machine code $7F_{16}$ 56_{16} 34_{16} 12_{16}

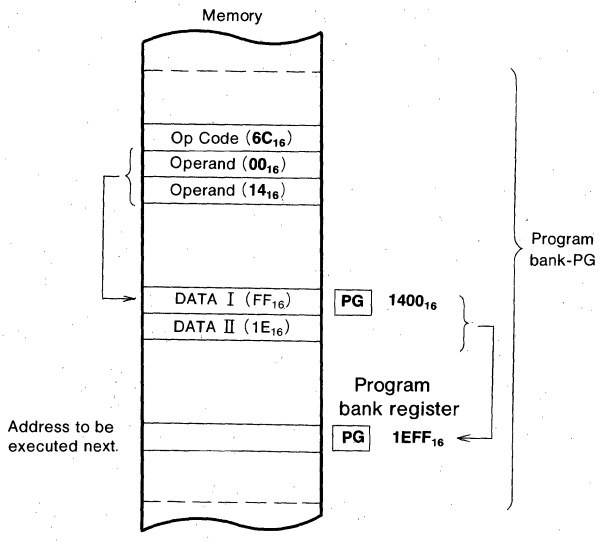


Mode : Absolute indirect addressing mode

Function : The instruction's second and third bytes specify 2 adjacent bytes in memory, and the contents of these bytes specify the address within the same program bank to which a jump is to be made.

Instruction : **JMP**

ex. : Mnemonic **JMP(1400H)** Machine code **$6C_{16}$ 00_{16} 14_{16}**



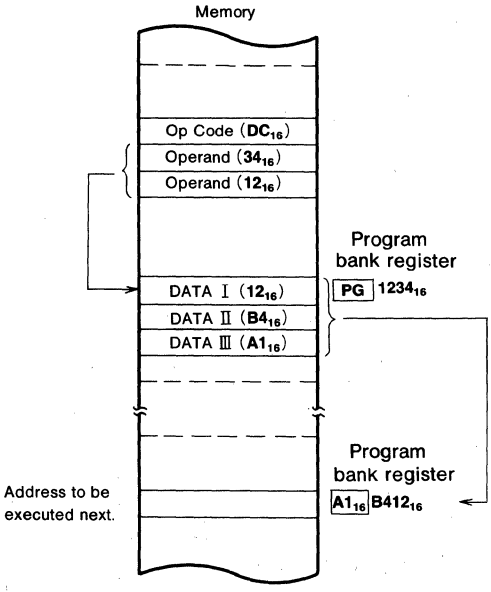
MITSUBISHI ELECTRIC

Mode : **Absolute indirect long addressing mode**

Function : The instruction's second and third bytes specify 3 adjacent bytes in memory, and the contents of these bytes specify the address to which a jump is to be made.

Instruction : **JMP**

ex. : Mnemonic **JMPL(1234H)** Machine code **DC_{16} 34_{16} 12_{16}**



DATA Ⅲ is loaded in the program bank register.

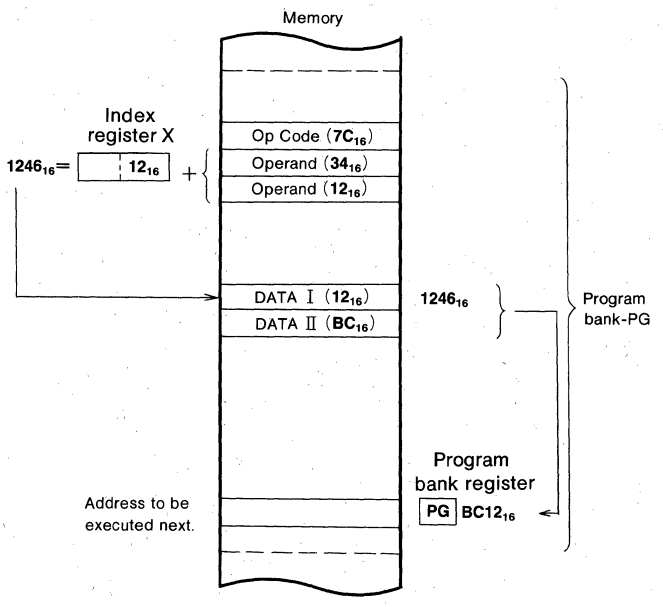
Mode : Absolute indexed X indirect addressing mode

Function : The value obtained by adding the instruction's second and third bytes and the contents of the index register X spcifies 2 adjacent bytes in memory, and the contents of these bytes specify the address to which a jump is to be made.

Instruction : **JMP, JSR**

ex. : Mnemonic **JMP(1234H, X)** **(x=1)**

Machine code **$7C_{16}$ 34_{16} 12_{16}**

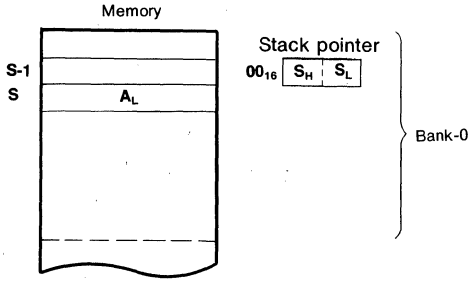


Mode : **Stack addressing mode**

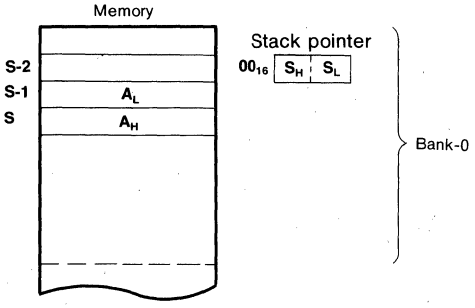
Function : Register contents are saved to or restored from the memory location specified by the stack pointer. The stack pointer is set in bank-0.

Instruction : **PEA, PEI, PER, PHA, PHB, PHD, PHG, PHP, PHT, PHX, PHY, PLA, PLB, PLD, PLP, PLT, PLX, PLY, PSH, PUL**

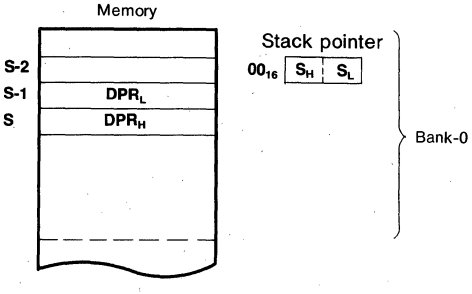
ex. : Mnemonic **PHA** (**m=1**) Machine code 48_{16}



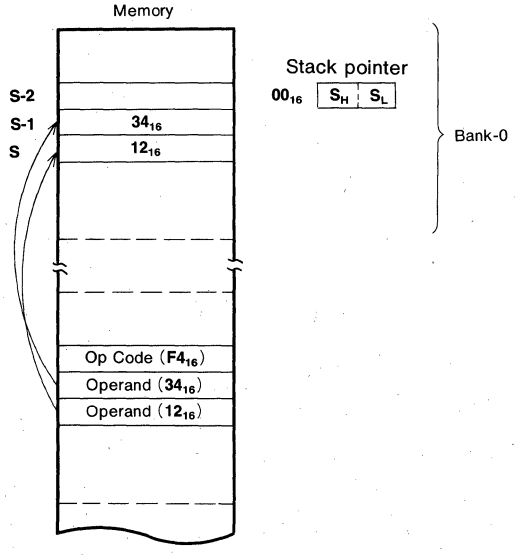
ex. : Mnemonic **PHA** (**m=0**) Machine code 48_{16}



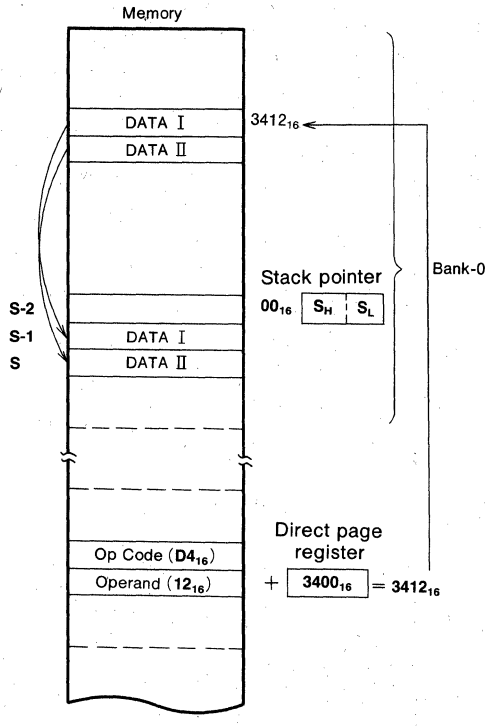
ex. : Mnemonic **PHD** Machine code $0B_{16}$



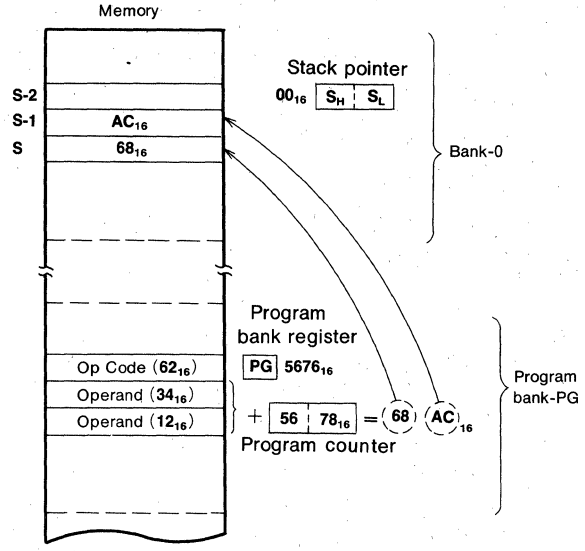
ex. : Mnemonic **PEA # 1234H** Machine code $F4_{16}$ 34_{16} 12_{16}



ex. : Mnemonic **PEI # 12H** Machine code **$D4_{16}$ 12_{16}**



ex. : Mnemonic **PER # 1234H** Machine code **62_{16} 34_{16} 12_{16}**



MITSUBISHI ELECTRIC

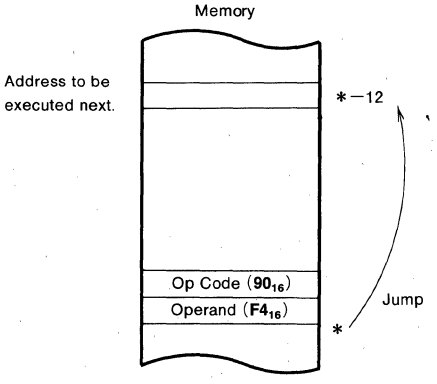
Mode : **Relative addressing mode**

Function : Branching occurs to the address specified by the value resulting from addition of the contents of the program counter and the instruction's second byte. In the case of a long branch by the BRA instruction, a 15-bit signed numeric value formed by the contents of the instruction's second and third bytes is added to the program counter contents. If the addition generates a carry or borrow, 1 is added to or subtracted from the program bank register.

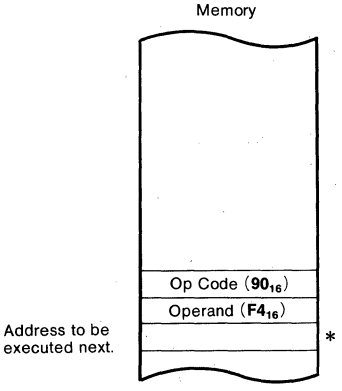
Instruction : **BCC, BCS, BEQ, BMI, BNE, BPL, BRA, BVC, BVS**

ex. : Mnemonic **BCC ＊−12** Machine code **90_{16} $F4_{16}$**

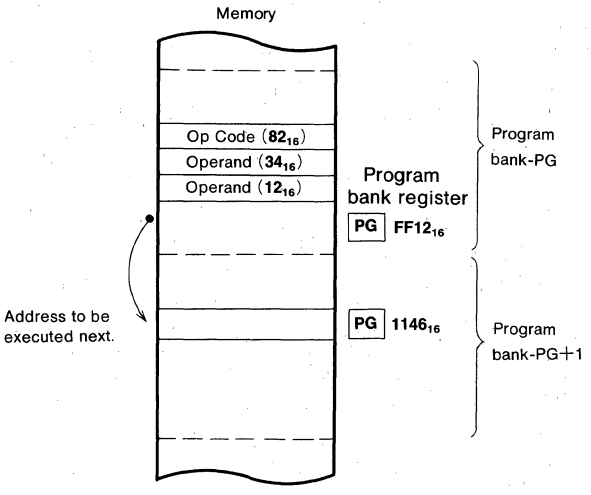
Branches to the address ＊−12 if the carry flag (C) has been cleared.



Advances to the address ＊ if the carry flag (C) has been set.



ex. : Mnemonic **BRA 1234H** Machine code **82_{16} 34_{16} 12_{16}**



Mode : Direct bit relative addressing mode

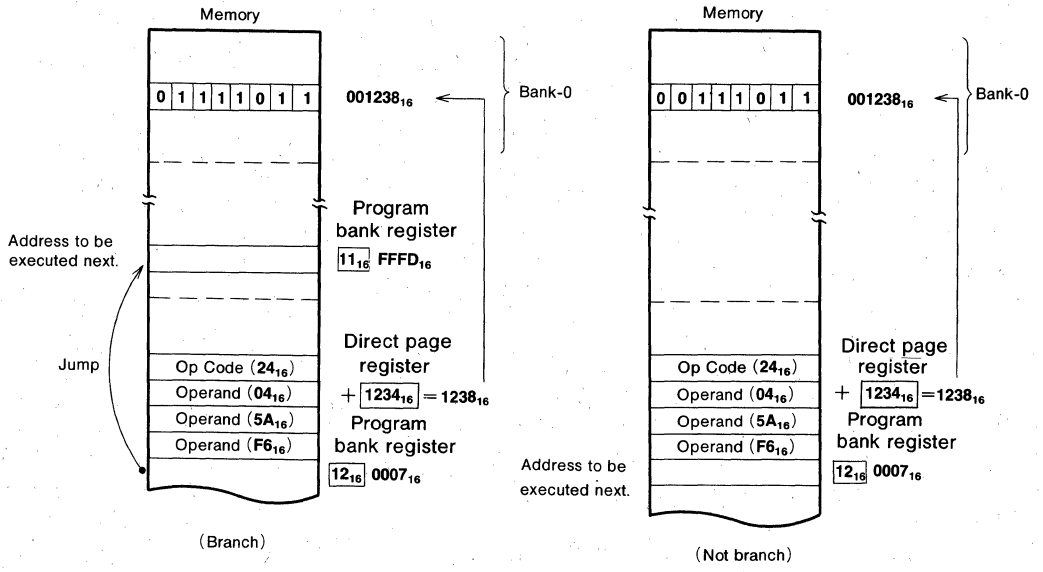
Function : Specifies the bank-0 memory location by the value obtained by adding the instruction's second byte to the direct page register's contents, and specifies the positions of multiple bits in the memory location by the bit pattern in the third and fourth bytes (the third byte only if the m flag is set to 1). Then, if the specified bits all satisfy the branching conditions, the instruction's fifth byte (or the fourth byte if the m flag is set to 1) is added to the program counter as a signed value, generating the branching destination address. If, however, addition of the instruction's second byte to the direct page register's contents result in a value that exceeds the bank-0 range, the specified location will be in bank-1.

Instruction : **BBC, BBS**

ex. : Mnemonic — Machine code

BBS #5AH, 04H, 0F6H — 24_{16} 04_{16} $5A_{16}$ $F6_{16}$

(m=1)

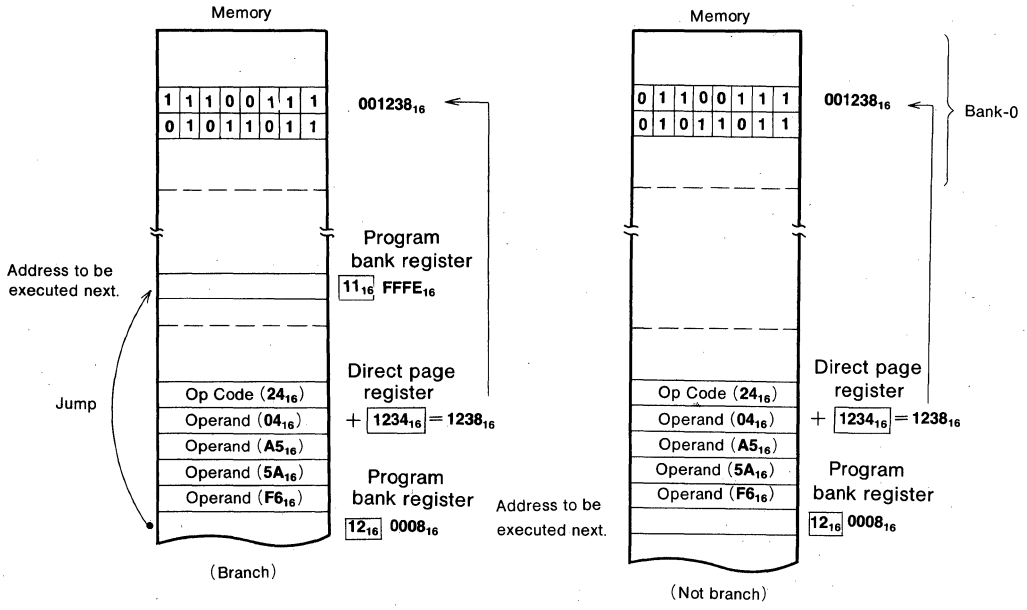


MITSUBISHI ELECTRIC

ex. : Mnemonic — Machine code

BBS #5AA5H, 04H, 0F6H — 24_{16} 04_{16} $A5_{16}$ $5A_{16}$ $F6_{16}$

(m=0)



Mode : **Absolute bit relative addressing mode**

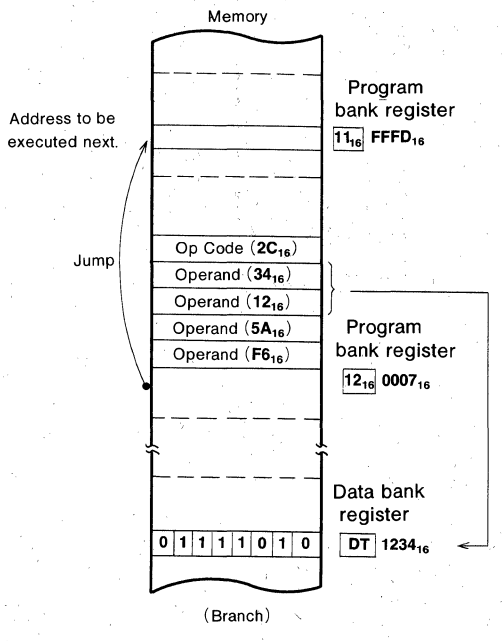
Function : The instruction's second and third bytes and the contents of the data bank register specify the memory location, and data for the memory location's multiple bits is specified by a bit pattern in the instruction's fourth and fifth bytes (the fourth byte only if the m flag is set to 1). Then , if the specified bits all satisfy the branching conditions, the instruction's sixth byte (or the fifth byte if the m flag is set to 1) is added to the program counter as a signed value, generating the branching destination address.

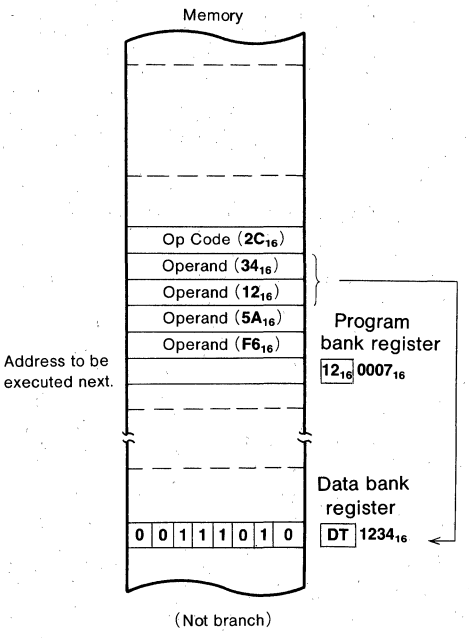
Instruction : **BBC, BBS**

ex. : Mnemonic — Machine code

BBS #5AH, 1234H, 0F6H — $2C_{16}\ 34_{16}\ 12_{16}\ 5A_{16}\ F6_{16}$

(m=1)

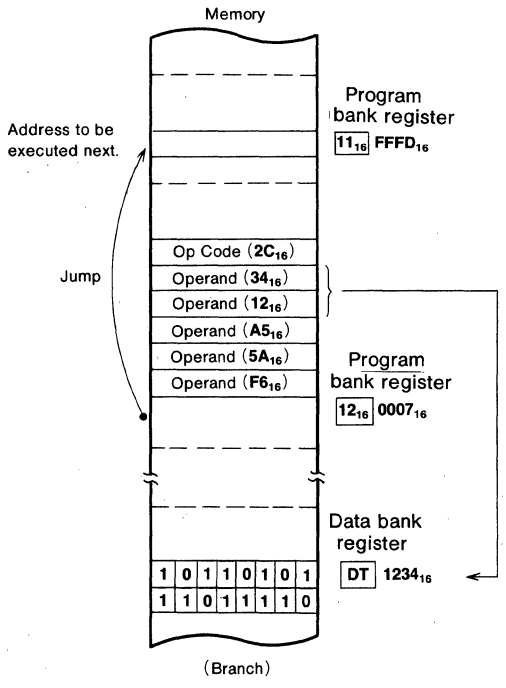


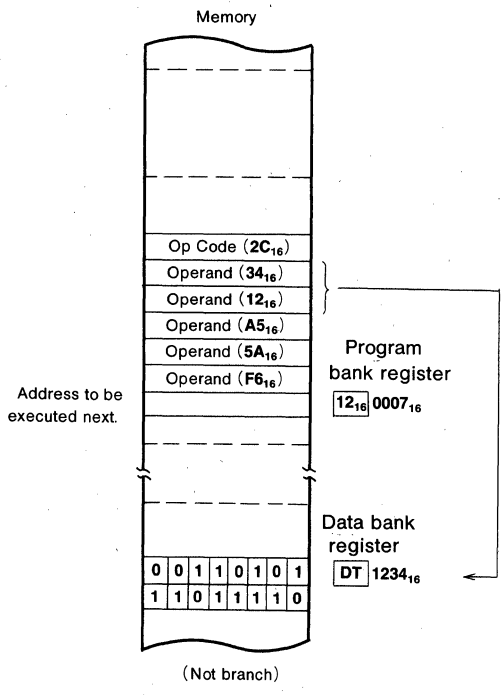


MITSUBISHI ELECTRIC

ex. : Mnemonic
BBS #5AA5H, 1234H, 0F6H
(m=0)

Machine code
$2C_{16}$ 34_{16} 12_{16} $A5_{16}$ $5A_{16}$ $F6_{16}$



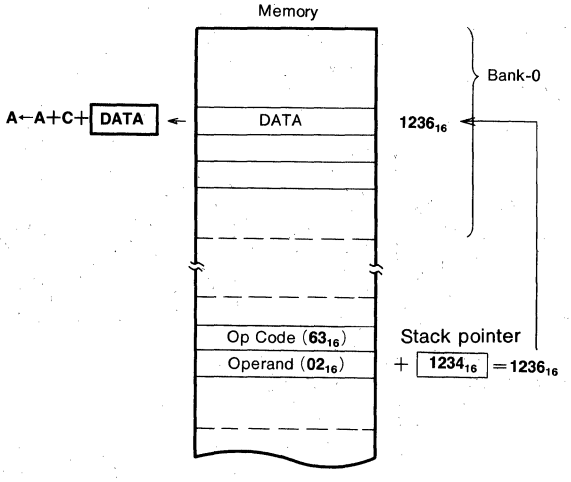


Mode : **Stack pointer relative addressing mode**

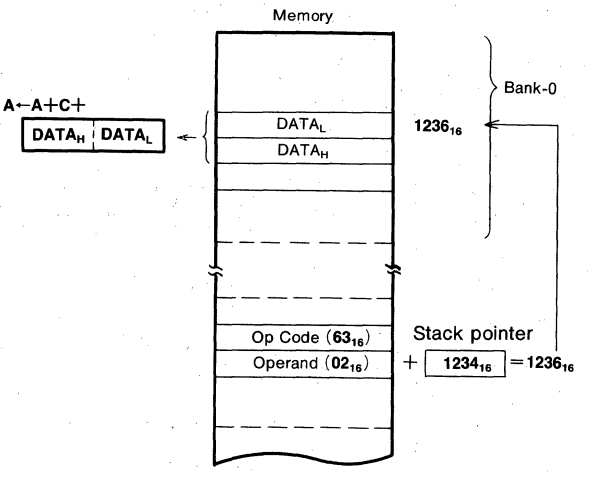
Function : The contents of a bank-0 memory location specified by the value resulting from addition of the instruction's second byte and the contents of the stack pointer become the actual data. If, however, the value obtained by adding the contents of the instruction's second byte and the stack pointer's contents exceeds the bank-0 range, the specified location will be in bank-1.

Instruction : **ADC, AND, CMP, DIV, EOR, LDA, MPY, ORA, SBC, STA**

ex. : Mnemonic **ADC A, 02H, S** (**m=1**) Machine code 63_{16} 02_{16}



ex. : Mnemonic **ADC A, 02H, S** (**m=0**) Machine code 63_{16} 02_{16}



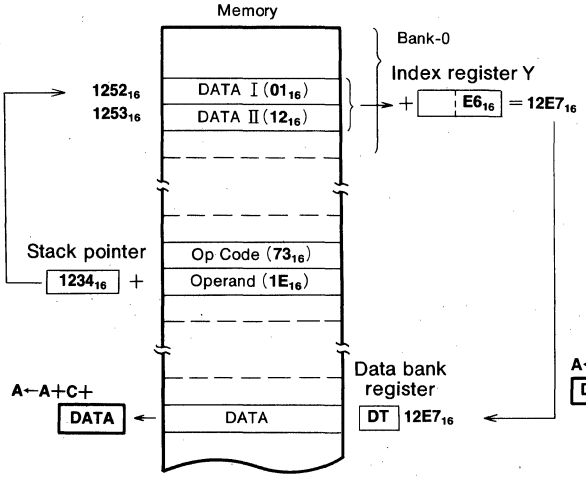
MITSUBISHI ELECTRIC

Mode : Stack pointer relative indirect indexed Y addressing mode

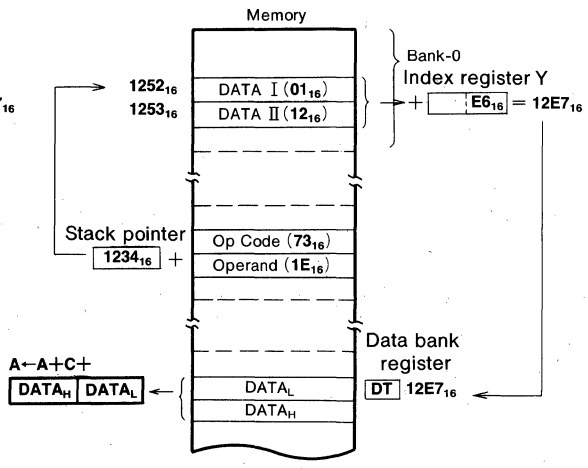
Function : The value obtained by adding the instruction's second byte and the contents of the stack pointer specifies 2 adjacent bytes in memory. The value obtained by adding the contents of these bytes and the contents of the index register Y specifies address of the actual data in memory bank-DT (DT is contents of data bank register). If addition of the 2 bytes in memory with the contents of the index register Y generate a carry, the bank number will be 1 larger than the contents of the data bank register.

Instruction : **ADC, AND, CMP, DIV, EOR, LDA, MPY, ORA, SBC, STA**

ex. : Mnemonic **ADC A,(1EH, S), Y** (**m=1, x=1**) Machine code 73_{16} $1E_{16}$



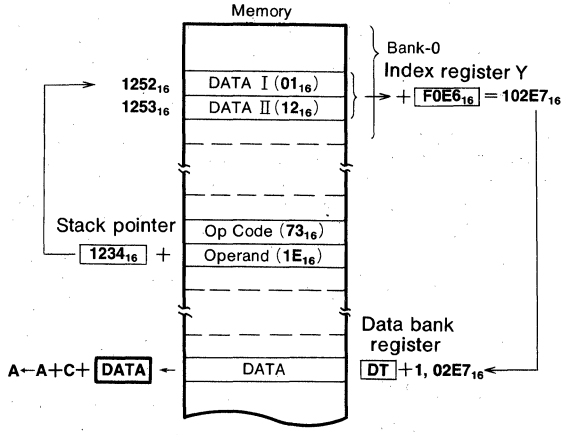
ex. : Mnemonic **ADC A,(1EH, S), Y** (**m=0, x=1**) Machine code 73_{16} $1E_{16}$



ex. : Mnemonic — Machine code

ADC A, (1EH, S), Y — 73_{16} $1E_{16}$

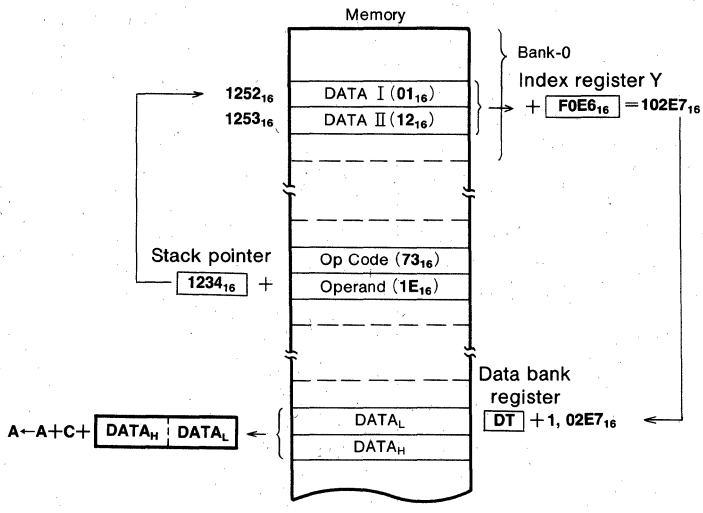
(m=1,x=0)



ex. : Mnemonic — Machine code

ADC A, (1EH, S), Y — 73_{16} $1E_{16}$

(m=0, x=0)



MITSUBISHI ELECTRIC

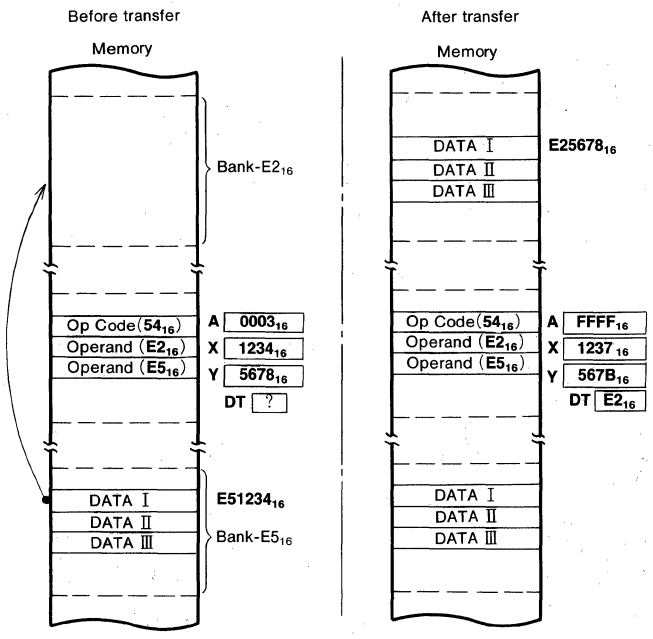
Mode : **Block transfer addressing mode**

Function : The instruction's second byte specifies the transfer-to data bank, and the contents of the index register Y specify the transfer - to address within the data bank. The instruction's third byte specifies the transfer-from data bank, and the contents of the index register X specify the address in the data bank where the data to be transferred is stored. The contents of the accumulator A constitute the number of bytes to be transfeered. Upon termination of transfer, the contents of the data bank register will specify the transfer-to data bank. The MVN instruction is used for transfer to lower address location. In this case, the contents of the index registers X and Y are incremented each time data is transferred. The MVP instruction is used for transfer to higher address location. In this case, the contents of the index registers X and Y are decremented each time data is transferred. The block of data to be transferred may cross over the bank boundary.

Instruction : **MVN, MVP**

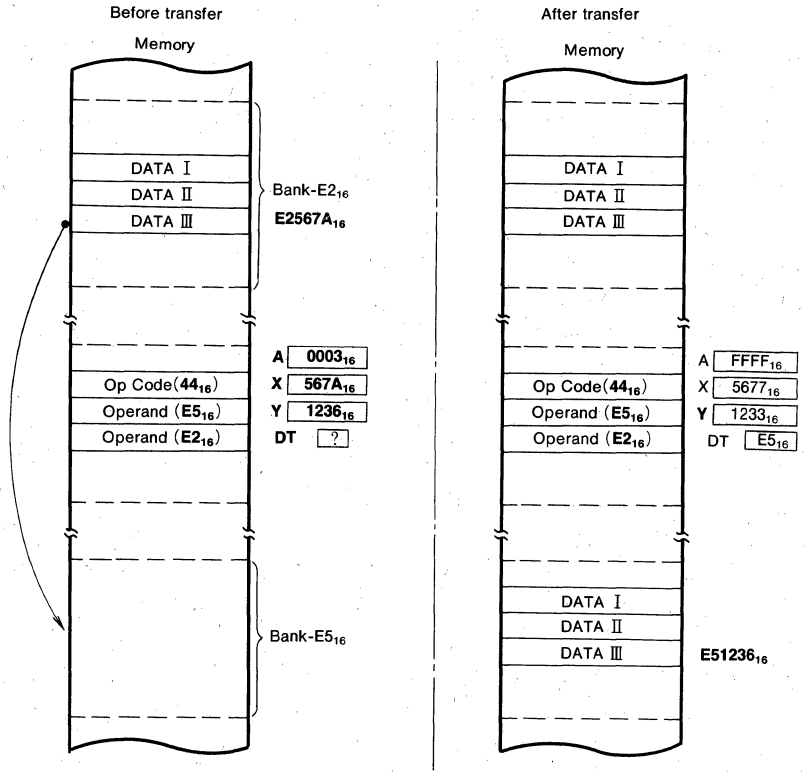
ex. : Mnemonic — Machine code

MVN 0E2H, 0E5H — **54_{16} $E2_{16}$ $E5_{16}$**



ex. : Mnemonic — Machine code

MVP 0E5H, 0E2H — **44_{16} $E5_{16}$ $E2_{16}$**



MELPS 7700 INSTRUCTION CODE TABLE

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

INSTRUCTION CODE TABLE-1

D_7~D_4 \ D_3~D_0	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	BRK	ORA A,(DIR,X)		ORA A,SR	SEB DIR,b	ORA A,DIR	ASL DIR	ORA A,L(DIR)	PHP	ORA A,IMM	ASL A	PHD	SEB ABS,b	ORA A,ABS	ASL ABS	ORA A,ABL
0001	1	BPL	ORA A,(DIR),Y	ORA A,(DIR)	ORA A,(SR),Y	CLB DIR,b	ORA A,DIR,X	ASL DIR,X	ORA A,L(DIR),Y	CLC	ORA A,ABS,Y	DEC A	TAS	CLB ABS,b	ORA A,ABS,X	ASL ABS,X	ORA A,ABL,X
0010	2	JSR ABS	AND A,(DIR,X)	JSR ABL	AND A,SR	BBS DIR,b,R	AND A,DIR	ROL DIR	AND A,L(DIR)	PLP	AND A,IMM	ROL A	PLD	BBS ABS,b,R	AND A,ABS	ROL ABS	AND A,ABL
0011	3	BMI	AND A,(DIR),Y	AND A,(DIR)	AND A,(SR),Y	BBC DIR,b,R	AND A,DIR,X	ROL DIR,X	AND A,L(DIR),Y	SEC	AND A,ABS,Y	INC A	TSA	BBC ABS,b,R	AND A,ABS,X	ROL ABS,X	AND A,ABL,X
0100	4	RTI	EOR A,(DIR,X)	Note 1	EOR A,SR	MVP	EOR A,DIR	LSR DIR	EOR A,L(DIR)	PHA	EOR A,IMM	LSR A	PHG	JMP ABS	EOR A,ABS	LSR ABS	EOR A,ABL
0101	5	BVC	EOR A,(DIR),Y	EOR A,(DIR)	EOR A,(SR),Y	MVN	EOR A,DIR,X	LSR DIR,X	EOR A,L(DIR),Y	CLI	EOR A,ABS,Y	PHY	TAD	JMP ABL	EOR A,ABS,X	LSR ABS,X	EOR A,ABL,X
0110	6	RTS	ADC A,(DIR,X)	PER	ADC A,SR	LDM DIR	ADC A,DIR	ROR DIR	ADC A,L(DIR)	PLA	ADC A,IMM	ROR A	RTL	JMP (ABS)	ADC A,ABS	ROR ABS	ADC A,ABL
0111	7	BVS	ADC A,(DIR),Y	ADC A,(DIR)	ADC A,(SR),Y	LDM DIR,X	ADC A,DIR,X	ROR DIR,X	ADC A,L(DIR),Y	SEI	ADC A,ABS,Y	PLY	TDA	JMP (ABS,X)	ADC A,ABS,X	ROR ABS,X	ADC A,ABL,X
1000	8	BRA REL	STA A,(DIR,X)	BRA REL	STA A,SR	STY DIR	STA A,DIR	STX DIR	STA A,L(DIR)	DEY	Note 2	TXA	PHT	STY ABS	STA A,ABS	STX ABS	STA A,ABL
1001	9	BCC	STA A,(DIR),Y	STA A,(DIR)	STA A,(SR),Y	STY DIR,X	STA A,DIR,X	STX DIR,Y	STA A,L(DIR),Y	TYA	STA A,ABS,Y	TXS	TXY	LDM ABS	STA A,ABS,X	LDM ABS,X	STA A,ABL,X
1010	A	LDY IMM	LDA A,(DIR,X)	LDX IMM	LDA A,SR	LDY DIR	LDA A,DIR	LDX DIR	LDA A,L(DIR)	TAY	LDA A,IMM	TAX	PLT	LDY ABS	LDA A,ABS	LDX ABS	LDA A,ABL
1011	B	BCS	LDA A,(DIR),Y	LDA A,(DIR)	LDA A,(SR),Y	LDY DIR,X	LDA A,DIR,X	LDX DIR,Y	LDA A,L(DIR),Y	CLV	LDA A,ABS,Y	TSX	TYX	LDY ABS,X	LDA A,ABS,X	LDX ABS,Y	LDA A,ABL,X
1100	C	CPY IMM	CMP A,(DIR,X)	CLP IMM	CMP A,SR	CPY DIR	CMP A,DIR	DEC DIR	CMP A,L(DIR)	INY	CMP A,IMM	DEX	WIT	CPY ABS	CMP A,ABS	DEC ABS	CMP A,ABL
1101	D	BNE	CMP A,(DIR),Y	CMP A,(DIR)	CMP A,(SR),Y	PEI	CMP A,DIR,X	DEC DIR,X	CMP A,L(DIR),Y	CLM	CMP A,ABS,Y	PHX	STP	JMP L(ABS)	CMP A,ABS,X	DEC ABS,X	CMP A,ABL,X
1110	E	CPX IMM	SBC A,(DIR,X)	SEP IMM	SBC A,SR	CPX DIR	SBC A,DIR	INC DIR	SBC A,L(DIR)	INX	SBC A,IMM	NOP	PSH	CPX ABS	SBC A,ABS	INC ABS	SBC A,ABL
1111	F	BEQ	SBC A,(DIR),Y	SBC A,(DIR)	SBC A,(SR),Y	PEA	SBC A,DIR,X	INC DIR,X	SBC A,L(DIR),Y	SEM	SBC A,ABS,Y	PLX	PUL	JSR (ABS,X)	SBC # A,ABS,X	INC ABS,X	SBC A,ABL,X

Note 1 : 42_{16} specifies the contents of the INSTRUCTION CODE TABLE-2.
About the second word's codes, refer to the INSTRUCTION CODE TABLE-2.

2 : 89_{16} specifies the contents of the INSTRUCTION CODE TABLE-3.
About the third word's codes, refer to the INSTRUCTION CODE TABLE-2.

INSTRUCTION CODE TABLE-2 (The first word's code of each instruction is 42_{16})

D_7~D_4 \ D_3~D_0	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0		ORA B,(DIR,X)		ORA B,SR		ORA B,DIR		ORA B,L(DIR)		ORA B,IMM	ASL B			ORA B,ABS		ORA B,ABL
0001	1		ORA B,(DIR),Y	ORA B,(DIR)	ORA B,(SR),Y		ORA B,DIR,X		ORA B,L(DIR),Y		ORA B,ABS,Y	DEC B	TBS		ORA B,ABS,X		ORA B,ABL,X
0010	2		AND B,(DIR,X)		AND B,SR		AND B,DIR		AND B,L(DIR)		AND B,IMM	ROL B			AND B,ABS		AND B,ABL
0011	3		AND B,(DIR),Y	AND B,(DIR)	AND B,(SR),Y		AND B,DIR,X		AND B,L(DIR),Y		AND B,ABS,Y	INC B	TSB		AND B,ABS,X		AND B,ABL,X
0100	4		EOR B,(DIR,X)		EOR B,SR		EOR B,DIR		EOR B,L(DIR)	PHB	EOR B,IMM	LSR B			EOR B,ABS		EOR B,ABL
0101	5		EOR B,(DIR),Y	EOR B,(DIR)	EOR B,(SR),Y		EOR B,DIR,X		EOR B,L(DIR),Y		EOR B,ABS,Y		TBD		EOR B,ABS,X		EOR B,ABL,X
0110	6		ADC B,(DIR,X)		ADC B,SR		ADC B,DIR		ADC B,L(DIR)	PLB	ADC B,IMM	ROR B			ADC B,ABS		ADC B,ABL
0111	7		ADC B,(DIR),Y	ADC B,(DIR)	ADC B,(SR),Y		ADC B,DIR,X		ADC B,L(DIR),Y		ADC B,ABS,Y		TDB		ADC B,ABS,X		ADC B,ABL,X
1000	8		STA B,(DIR,X)		STA B,SR		STA B,DIR		STA B,L(DIR)			TXB			STA B,ABS		STA B,ABL
1001	9		STA B,(DIR),Y	STA B,(DIR)	STA B,(SR),Y		STA B,DIR,X		STA B,L(DIR),Y	TYB	STA B,ABS,Y				STA B,ABS,X		STA B,ABL,X
1010	A		LDA B,(DIR,X)		LDA B,SR		LDA B,DIR		LDA B,L(DIR)	TBY	LDA B,IMM	TBX			LDA B,ABS		LDA B,ABL
1011	B		LDA B,(DIR),Y	LDA B,(DIR)	LDA B,(SR),Y		LDA B,DIR,X		LDA B,L(DIR),Y		LDA B,ABS,Y				LDA B,ABS,X		LDA B,ABL,X
1100	C		CMP B,(DIR,X)		CMP B,SR		CMP B,DIR		CMP B,L(DIR)		CMP B,IMM				CMP B,ABS		CMP B,ABL
1101	D		CMP B,(DIR),Y	CMP B,(DIR)	CMP B,(SR),Y		CMP B,DIR,X		CMP B,L(DIR),Y		CMP B,ABS,Y				CMP B,ABS,X		CMP B,ABL,X
1110	E		SBC B,(DIR,X)		SBC B,SR		SBC B,DIR		SBC B,L(DIR)		SBC B,IMM				SBC B,ABS		SBC B,ABL
1111	F		SBC B,(DIR),Y	SBC B,(DIR)	SBC B,(SR),Y		SBC B,DIR,X		SBC B,L(DIR),Y		SBC B,ABS,Y				SBC B,ABS,X		SBC B,ABL,X

INSTRUCTION CODE TABLE-3 (The first word's code of each instruction is 89_{16})

D_7~D_4 \ D_3~D_0	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0		MPY (DIR,X)		MPY SR		MPY DIR		MPY L(DIR)		MPY IMM				MPY ABS		MPY ABL
0001	1		MPY (DIR),Y	MPY (DIR)	MPY (SR),Y		MPY DIR,X		MPY L(DIR),Y		MPY ABS,Y				MPY ABS,X		MPY ABL,X
0010	2		DIV (DIR,X)		DIV SR		DIV DIR		DIV L(DIR)	XAB	DIV IMM				DIV ABS		DIV ABL
0011	3		DIV (DIR),Y	DIV (DIR)	DIV (SR),Y		DIV DIR,X		DIV L(DIR),Y		DIV ABS,Y				DIV ABS,X		DIV ABL,X
0100	4										RLA IMM						
0101	5																
0110	6																
0111	7																
1000	8																
1001	9																
1010	A																
1011	B																
1100	C			LDT IMM													
1101	D																
1110	E																
1111	F																

MACHINE INSTRUCTIONS

Symbol	Function	Details	Addressing mode																														
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
ADC (Note 1,2)	A_{CC},C ← A_{CC}+M+C	Adds the carry, the accumulator and the memory contents. The result is entered into the accumulator. When the D flag is "0", binary additions is done, and when the D flag is "1", decimal addition is done.				69	2	2				65	4	2				75	5	2				72	6	2	61	7	2	71	8	2
						42 69	4	3				42 65	6	3				42 75	7	3				42 72	8	3	42 61	9	3	42 71	10	3
AND (Note 1,2)	A_{CC} ← A_{CC}∧M	Obtains the logical product of the contents of the accumulator and the contents of the memory. The result is entered into the accumulator.				29	2	2				25	4	2				35	5	2				32	6	2	21	7	2	31	8	2
						42 29	4	3				42 25	6	3				42 35	7	3				42 32	8	3	42 21	9	3	42 31	10	3
ASL (Note 1)	m=0 C ← b_{15} ··· b_0 ← 0 m=1 C ← b_7 ··· b_0 ← 0	Shifts the accumulator or the memory contents one bit to the left. "0" is entered into bit 0 of the accumulator or the memory. The contents of bit 15 (bit 7 when the m flag is "1") of the accumulator or memory before shift is entered into the C flag.							0A	2	1	06	7	2				16	7	2												
									42 0A	4	2																					
BBC (Note 3,5)	Mb=0 ?	Tests the specified bit of the memory. Branches when all the contents of the specified bit is "0".																															
BBS (Note 3,5)	Mb=1 ?	Tests the specified bit of the memory. Branches when all the contents of the specified bit is "1".																															
BCC (Note 3)	C=0 ?	Branches when the contents of the C flag is "0".																															
BCS (Note 3)	C=1 ?	Branches when the contents of the C flag is "1".																															
BEQ (Note 3)	Z=1 ?	Branches when the contents of the Z flag is "1".																															
BMI (Note 3)	N=1 ?	Branches when the contents of the N flag is "1".																															
BNE (Note 3)	Z=0 ?	Branches when the contents of the Z flag is "0".																															
BPL (Note 3)	N=0 ?	Branches when the contents of the N flag is "0".																															
BRA (Note 4)	PC←PC±offset PG←PG+1 (carry occured) PG←PG−1 (borrow occured)	Jumps to the address indicated by the program counter plus the offset value.																															
BRK	PC←PC+2 M(S)←PG S←S−1 M(S)←PC_H S←S−1 M(S)←PC_L S←S−1 M(S)←PS_H S←S−1 M(S)←PS_L S←S−1 I←1 PC_L←AD_L PC_H←AD_H PG←00_{16}	Executes software interruption.	00	15	2																												
BVC (Note 3)	V=0 ?	Branches when the contents of the V flag is "0".																															
BVS (Note 3)	V=1 ?	Branches when the contents of the V flag is "1".																															
CLB (Note 5)	Mb←0	Makes the contents of the specified bit in the memory "0".													14	8	3																
CLC	C←0	Makes the contents of the C flag "0".	18	2	1																												
CLI	I←0	Makes the contents of the I flag "0".	58	2	1																												
CLM	m←0	Makes the contents of the m flag "0".	D8	2	1																												
CLP	PSb←0	Specifies the bit position in the processor status register by the bit pattern of the second byte in the instruction, and sets "0" in that bit.				C2	4	2																									
CLV	V←0	Makes the contents of the V flag "0".	B8	2	1																												
CMP (Note 1,2)	A_{CC}−M	Compares the contents of the accumulator with the contents of the memory.				C9	2	2				C5	4	2				D5	5	2				D2	6	2	C1	7	2	D1	8	2
						42 C9	4	3				42 C5	6	3				42 D5	7	3				42 D2	8	3	42 C1	9	3	42 D1	10	3

MITSUBISHI ELECTRIC

Addressing mode																																																						Processor status register										
L(DIR)			L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
67	10	2	77	11	2	6D	4	3				7D	6	3	79	6	3	6F	6	4	7F	7	4																							63	5	2	73	8	2				•	•	•	N	V	•	•	•	•	Z	C
42 67	12	3	42 77	13	3	42 6D	6	4				42 7D	8	4	42 79	8	4	42 6F	8	5	42 7F	9	5																							42 63	7	3	42 73	10	3														
27	10	2	37	11	2	2D	4	3				3D	6	3	39	6	3	2F	6	4	3F	7	4																							23	5	2	33	8	2				•	•	•	N	•	•	•	•	•	Z	•
42 27	12	3	42 37	13	3	42 2D	6	4				42 3D	8	4	42 39	8	4	42 2F	8	5	42 3F	9	5																							42 23	7	3	42 33	10	3														
						0E	7	3				1E	8	3																																								•	•	•	N	•	•	•	•	•	Z	C
																																							34	7	4	3C	8	5										•	•	•	•	•	•	•	•	•	•	•
																																							24	7	4	2C	8	5										•	•	•	•	•	•	•	•	•	•	•
																																				90	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				B0	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				F0	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				30	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				D0	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				10	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				80	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				82	4	3																										
																																																						•	•	•	•	•	•	•	•	•	•	•
																																				50	4	2																•	•	•	•	•	•	•	•	•	•	•
																																				70	4	2																•	•	•	•	•	•	•	•	•	•	•
									1C	9	4																																											•	•	•	•	•	•	•	•	•	•	•
																																																						•	•	•	•	•	•	•	•	•	•	0
																																																						•	•	•	•	•	•	•	•	0	•	•
																																																						•	•	•	•	•	0	•	•	•	•	•
																																																						•	•	•	Specified flag becomes "0".							
																																																						•	•	•	•	0	•	•	•	•	•	•
C7	10	2	D7	11	2	CD	4	3				DD	6	3	D9	6	3	CF	6	4	DF	7	4																							C3	5	2	D3	8	2				•	•	•	N	•	•	•	•	•	Z	C
42 C7	12	3	42 D7	13	3	42 CD	6	4				42 DD	8	4	42 D9	8	4	42 CF	8	5	42 DF	9	5																							42 C3	7	3	42 D3	10	3														

Symbol	Function	Details	Addressing mode																																					
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y										
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#									
CPX (Note 2)	X−M	Compares the contents of the index register X with the contents of the memory.				E0	2	2				E4	4	2																											
CPY (Note 2)	Y−M	Compares the contents of the index register Y with the contents of the memory.				C0	2	2				C4	4	2																											
DEC (Note 1)	A_{CC}←A_{CC}−1 or M←M−1	Decrements the contents of the accumulator or memory by 1.							1A	2	1	C6	7	2				D6	7	2																					
									42 1A	4	2																																
DEX	X←X−1	Decrements the contents of the index register X by 1.	CA	2	1																																				
DEY	Y←Y−1	Decrements the contents of the index register Y by 1.	88	2	1																																				
DIV (Note 2,10)	A(quotient)←B,A/M B(remainder)	The numeral that places the contents of accumulator B to the higher order and the contents of accumulator A to the lower order is divided by the contents of the memory. The quotient is entered into accumulator A and the remainder into accumulator B.				89 29	27	3				89 25	29	3				89 35	30	3				89 32	31	3	89 21	32	3	89 31	33	3									
EOR (Note 1,2)	A_{CC}←A_{CC}∀M	Logical exclusive sum is obtained of the contents of the accumulator and the contents of the memory. The result is placed into the accumulator.				49	2	2				45	4	2				55	5	2				52	6	2	41	7	2	51	8	2									
						42 49	4	3				42 45	6	3				42 55	7	3				42 52	8	3	42 41	9	3	42 51	10	3									
INC (Note 1)	A_{CC}←A_{CC}+1 or M←M+1	Increments the contents of the accumulator or memory by 1.							3A	2	1	E6	7	2				F6	7	2																					
									42 3A	4	2																																
INX	X←X+1	Increments the contents of the index register X by 1.	E8	2	1																																				
INY	Y←Y+1	Increments the contents of the index register Y by 1.	C8	2	1																																				
JMP	ABS PC_L ← AD_L PC_H ← AD_H ABL PC_L ← AD_L PC_H ← AD_H PG←AD_G (ABS) PC_L ←(AD_H, AD_L) PC_H ←(AD_H, AD_L+1) L(ABS) PC_L ←(AD_H, AD_L) PC_H ←(AD_H, AD_L+1) PG ←(AD_H, AD_L+2) (ABS, X) PC_L ←(AD_H, AD_L+X) PC_H ←(AD_H, AD_L+X +1)	Places a new address into the program counter and jumps to that new address.																																						
JSR	ABS M(S)← PC_H S←S−1 M(S)← PC_L S←S−1 PC_L ← AD_L PC_H ← AD_H ABL M(S)← PG S←S−1 M(S)← PC_H S←S−1 M(S)←PC_L S ← S−1 PC_L ← AD_L PC_H ← AD_H PG ← AD_G (ABS, X) M(S)← PC_H S←S−1 M(S)← PC_L S←S−1 PC_L ←(AD_H, AD_L+X) PC_H ←(AD_H, AD_L+X +1)	Saves the contents of the program counter (also the contents of the program bank register for ABL) into the stack, and jumps to the new address.																																						

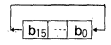
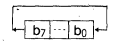
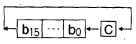
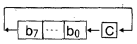
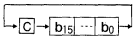
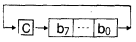
Addressing mode																																																						Processor status register										
L(DIR)			L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
						EC	4	3																																														•	•	•	N	•	•	•	•	•	Z	C
						CC	4	3																																														•	•	•	N	•	•	•	•	•	Z	C
						CE	7	3				DE	8	3																																								•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
89 27	35	3	89 37	36	3	89 2D	29	4				89 3D	31	4	89 39	31	4	89 2F	31	5	89 3F	32	5																							89 23	30	3	89 33	33	3				•	•	•	N	V	•	•	•	•	Z	C
47	10	2	57	11	2	4D	4	3				5D	6	3	59	6	3	4F	6	4	5F	7	4																							43	5	2	53	8	2				•	•	•	N	•	•	•	•	•	Z	•
42 47	12	3	42 57	13	3	42 4D	6	4				42 5D	8	4	42 59	8	4	42 4F	8	5	42 5F	9	5																							42 43	7	3	42 53	10	3														
						EE	7	3				FE	8	3																																								•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
						4C	2	3										5C	4	4				6C	4	3	DC	8	3	7C	6	3																						•	•	•	•	•	•	•	•	•	•	•
						20	6	3										22	8	4										FC	8	3																						•	•	•	•	•	•	•	•	•	•	•

Symbol	Function	Details	Addressing mode																																
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y					
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
LDA (Note 1,2)	$A_{CC} \leftarrow M$	Enters the contents of the memory into the accumulator.				A9	2	2				A5	4	2				B5	5	2				B2	6	2	A1	7	2	B1	8	2
						42 A9	4	3				42 A5	6	3				42 B5	7	3				42 B2	8	3	42 A1	9	3	42 B1	10	3
LDM (Note 5)	$M \leftarrow IMM$	Enters the immediate value into the memory.										64	4	3				74	5	3															
LDT	$DT \leftarrow IMM$	Enters the immediate value into the data bank register.				89 C2	5	3																											
LDX (Note 2)	$X \leftarrow M$	Enters the contents of the memory into index register X.				A2	2	2				A6	4	2							B6	5	2												
LDY (Note 2)	$Y \leftarrow M$	Enters the contents of the memory into index register Y.				A0	2	2				A4	4	2				B4	5	2															
LSR (Note 1)	m=0 0→[b_{15} ··· b_0]→C m=1 0→[b_7 ··· b_0]→C	Shifts the contents of the accumulator or the contents of the memory one bit to the right. The bit 0 of the accumulator or the memory is entered into the C flag. "0" is entered into bit 15 (bit 7 when the m flag is "1".)							4A	2	1	46	7	2				56	7	2															
									42 4A	4	2																									
MPY (Note 2,11)	B, A←A*M	Multiplies the contents of accumulator A and the contents of the memory. The higher order of the result of operation are entered into accumulator B, and the lower order into accumulator A.				89 09	16	3				89 05	18	3				89 15	19	3				89 12	20	3	89 01	21	3	89 11	22	3
MVN (Note 8)	Mn+i←Mm+i	Transmits the data block. The transmission is done from the lower order address of the block.																																	
MVP (Note 9)	Mn−i←Mm−i	Transmits the data block. Transmission is done form the higher order address of the data block.																																	
NOP	PC←PC+1	Advances the program counter, but performs nothing else.	EA	2	1																														
ORA (Note 1,2)	$A_{CC} \leftarrow A_{CC} \vee M$	Logical sum per bit of the contents of the accumulator and the contents of the memory is obtained. The result is entered into the accumulator.				09	2	2				05	4	2				15	5	2				12	6	2	01	7	2	11	8	2
						42 09	4	3				42 05	6	3				42 15	7	3				42 12	8	3	42 01	9	3	42 11	10	3
PEA	$M(S) \leftarrow IMM_2$ $S \leftarrow S-1$ $M(S) \leftarrow IMM_1$ $S \leftarrow S-1$	The 3rd and the 2nd bytes of the instruction are saved into the stack, in this order.																																	
PEI	$M(S) \leftarrow M((DPR)+IMM+1)$ $S \leftarrow S-1$ $M(S) \leftarrow M((DPR)+IMM)$ $S \leftarrow S-1$	Specifies 2 sequential bytes in the direct page in the 2nd byte of the instruction, and saves the contents into the stack.																																	
PER	$EAR \leftarrow PC+IMM_2IMM_1$ $M(S) \leftarrow EAR_H$ $S \leftarrow S-1$ $M(S) \leftarrow EAR_L$ $S \leftarrow S-1$	Regards the 2nd and 3rd bytes of the instruction as 16-bit numerals, adds them to the program counter, and saves the result into the stack.																																	
PHA	m=0 $M(S) \leftarrow A_H$ $S \leftarrow S-1$ $M(S) \leftarrow A_L$ $S \leftarrow S-1$ m=1 $M(S) \leftarrow A_L$ $S \leftarrow S-1$	Saves the contents of accumulator A into the stack.																																	
PHB	m=0 $M(S) \leftarrow B_H$ $S \leftarrow S-1$ $M(S) \leftarrow B_L$ $S \leftarrow S-1$ m=1 $M(S) \leftarrow B_L$ $S \leftarrow S-1$	Saves the contents of accumulator B into the stack.																																	

Addressing mode																																																						Processor status register										
L(DIR)			L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
A7	10	2	B7	11	2	AD	4	3				BD	6	3	B9	6	3	AF	6	4	BF	7	4																							A3	5	2	B3	8	2				•	•	•	N	•	•	•	•	•	Z	•
42 A7	12	3	42 B7	13	3	42 AD	6	4				42 BD	8	4	42 B9	8	4	42 AF	8	5	42 BF	9	5																							42 A3	7	3	42 B3	10	3														
						9C	5	4				9E	6	4																																								•	•	•	•	•	•	•	•	•	•	•
																																																						•	•	•	•	•	•	•	•	•	•	•
						AE	4	3							BE	6	3																																					•	•	•	N	•	•	•	•	•	Z	•
						AC	4	3				BC	6	3																																								•	•	•	N	•	•	•	•	•	Z	•
						4E	7	3				5E	8	3																																								•	•	•	0	•	•	•	•	•	Z	C
89 07	24	3	89 17	25	3	89 0D	18	4				89 1D	20	4	89 19	20	4	89 0F	20	5	89 1F	21	5																							89 03	19	3	89 13	22	3				•	•	•	N	•	•	•	•	•	Z	0
																																																				54	$7+\frac{i}{2}\times7$	3	•	•	•	•	•	•	•	•	•	•	•
																																																				44	$9+\frac{i}{2}\times7$	3	•	•	•	•	•	•	•	•	•	•	•
																																																							•	•	•	•	•	•	•	•	•	•	•
07	10	2	17	11	2	0D	4	3				1D	6	3	19	6	3	0F	6	4	1F	7	4																							03	5	2	13	8	2				•	•	•	N	•	•	•	•	•	Z	•
42 07	12	3	42 17	13	3	42 0D	6	4				42 1D	8	4	42 19	8	4	42 0F	8	5	42 1F	9	5																							42 03	7	3	42 13	10	3														
																																	F4	5	3																					•	•	•	•	•	•	•	•	•	•	•
																																	D4	5	2																					•	•	•	•	•	•	•	•	•	•	•
																																	62	5	3																					•	•	•	•	•	•	•	•	•	•	•
																																	48	4	1																					•	•	•	•	•	•	•	•	•	•	•
																																	42 48	6	2																					•	•	•	•	•	•	•	•	•	•	•

Symbol	Function	Details	Addressing mode																																
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
PHD	M(S)←DPR_H S←S−1 M(S)←DPR_L S←S−1	Saves the contents of the direct page register into the stack.																																	
PHG	M(S)←PG S←S−1	Saves the contents of the program bank register into the stack.																																	
PHP	M(S)←PS_H S←S−1 M(S)←PS_L S←S−1	Saves the contents of the program status register into the stack.																																	
PHT	M(S)←DT S←S−1	Saves the contents of the data bank register into the stack.																																	
PHX	x=0 M(S)←X_H S←S−1 M(S)←X_L S←S−1 x=1 M(S)←X_L S←S−1	Saves the contents of the index register X into the stack.																																	
PHY	x=0 M(S)←Y_H S←S−1 M(S)←Y_L S←S−1 x=1 M(S)←Y_L S←S−1	Saves the contents of the index register Y into the stack.																																	
PLA	m=0 S←S+1 A_L←M(S) S←S+1 A_H←M(S) m=1 S←S+1 A_L←M(S)	Restores the contents of the stack on the accumulator A.																																	
PLB	m=0 S←S+1 B_L←M(S) S←S+1 B_H←M(S) m=1 S←S+1 B_L←M(S)	Restores the contents of the stack on the accumulator B.																																	
PLD	S←S+1 DPR_L←M(S) S←S+1 DPR_H←M(S)	Restores the contents of the stack on the direct page register.																																	
PLP	S←S+1 PS_L←M(S) S←S+1 PS_H←M(S)	Restores the contents of the stack on the processor status register.																																	
PLT	S←S+1 DT←M(S)	Restores the contents of the stack on the data bank register.																																	
PLX	x=0 S←S+1 X_L←M(S) S←S+1 X_H←M(S) x=1 S←S+1 X_L←M(S)	Restores the contents of the stack on the index register X.																																	

Addressing mode																																																			Processor status register										
L(DIR)			L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																																	0B	4	1																•	•	•	•	•	•	•	•	•	•	•
																																	4B	3	1																•	•	•	•	•	•	•	•	•	•	•
																																	08	4	1																•	•	•	•	•	•	•	•	•	•	•
																																	8B	3	1																•	•	•	•	•	•	•	•	•	•	•
																																	DA	4	1																•	•	•	•	•	•	•	•	•	•	•
																																	5A	4	1																•	•	•	•	•	•	•	•	•	•	•
																																	68	5	1																•	•	•	N	•	•	•	•	•	Z	•
																																	42 68	7	2																•	•	•	N	•	•	•	•	•	Z	•
																																	2B	5	1																•	•	•	•	•	•	•	•	•	•	•
																																	28	6	1																Value saved in stack.										
																																	AB	6	1																•	•	•	N	•	•	•	•	•	Z	•
																																	FA	5	1																•	•	•	N	•	•	•	•	•	Z	•

Symbol	Function	Details	Addressing mode																														
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
PLY	x=0 S←S+1 Y_L←M(S) S←S+1 Y_H←M(S) x=1 S←S+1 Y_L←M(S)	Restores the contents of the stack on the index register Y.																															
PSH (Note 6)	M(S)←A, B, X···	Saves the registers among accumulator, index register, direct page register, data bank register, program bank register, or processor status register, specified by the bit pattern of the second byte of the instruction into the stack.																															
PUL (Note 7)	A, B, X···←M(S)	Restores the contents of the stack to the registers among accumulator, index register, direct page register, data bank register, or processor status register, specified by the bit pattern of the second byte of the instruction.																															
RLA (Note 13)	m=0 n bit rotate left  m=1 n bit rotate left 	Rotates the contents of the accumulator A, n bits to the left.				89 49	6 + i	3																									
ROL (Note 1)	m=0  m=1 	Links the accumulator or the memory to C flag, and rotates result to the left by 1 bit.							2A 42 2A	2 4	1 2	26	7	2				36	7	2												
ROR (Note 1)	m=0  m=1 	Links the accumulator or the memory to C flag, and rotates result to the right by 1 bit.							6A 42 6A	2 4	1 2	66	7	2				76	7	2												
RTI	S←S+1 PS_L←M(S) S←S+1 PS_H←M(S) S←S+1 PC_L←M(S) S←S+1 PC_H←M(S) S←S+1 PG←M(S)	Returns from the interruption routine.	40	11	1																												
RTL	S←S+1 PC_L←M(S) S←S+1 PC_H←M(S) S←S+1 PG←M(S)	Returns from the subroutine. The contents of the program bank register are also restored.	6B	8	1																												
RTS	S←S+1 PC_L←M(S) S←S+1 PC_H←M(S)	Returns from the subroutine. The contents of the program bank register are not restored.	60	5	1																												
SBC (Note 1,2)	A_{CC}, C←A_{CC}−M−$\overline{C}$	Subtracts the contents of the memory and the borrow from the contents of the accumulator.				E9 42 E9	2 4	2 3				E5 42 E5	4 6	2 3				F5 42 F5	5 7	2 3				F2 42 F2	6 8	2 3	E1 42 E1	7 9	2 3	F1 42 F1	8 10	2 3

Addressing mode																																																						Processor status register										
L(DIR)			L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
																																	7A	5	1																			•	•	•	N	•	•	•	•	•	Z	•
																																	EB	12 + $2i_1+i_2$	2																			•	•	•	•	•	•	•	•	•	•	•
																																	FB	14 + $3i_1+4i_2$	2																			If restored the contents of PS, it becomes its value. And the other case is no change.										
																																																					•	•	•	•	•	•	•	•	•	•	•
						2E	7	3				3E	8	3																																								•	•	•	N	•	•	•	•	•	Z	C
						6E	7	3				7E	8	3																																								•	•	•	N	•	•	•	•	•	Z	C
																																																							Value saved in stack.										
																																																					•	•	•	•	•	•	•	•	•	•	•
																																																					•	•	•	•	•	•	•	•	•	•	•
E7	10	2	F7	11	2	ED	4	3				FD	6	3	F9	6	3	EF	6	4	FF	7	4																						E3	5	2	F3	8	2				•	•	•	N	V	•	•	•	•	Z	C
42 E7	12	3	42 F7	13	3	42 ED	6	4				42 FD	8	4	42 F9	8	4	42 EF	8	5	42 FF	9	5																						42 E3	7	3	42 F3	10	3														

Symbol	Function	Details	Addressing mode																																
			IMP			IMM			A			DIR			DIR,b			DIR,X			DIR,Y			(DIR)			(DIR,X)			(DIR),Y		
			op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#
SEB (Note 5)	Mb←1	Makes the contents of the specified bit in the memory "1".													04	8	3																	
SEC	C←1	Makes the contents of the C flag "1".	38	2	1																													
SEI	I←1	Makes the contents of the I flag "1".	78	2	1																													
SEM	m←1	Makes the contents of the m flag "1".	F8	2	1																													
SEP	PSb←1	Set the specified bit of the processor status register's lower byte (PS_L) to "1".				E2	3	2																										
STA (Note 1)	M←A_{CC}	Stores the contents of the accumulator into the memory.										85	4	2				95	5	2				92	7	2	81	7	2	91	7	2
												42 85	6	3				42 95	7	3				42 92	9	3	42 81	9	3	42 91	9	3
STP		Stops the oscillation of the oscillator.	DB	3	1																													
STX	M←X	Stores the contents of the index register X into the memory.										86	4	2							96	5	2											
STY	M←Y	Stores the contents of the index register Y into the memory.										84	4	2				94	5	2														
TAD	DPR←A	Transmits the contents of the accumulator A to the direct page register.	5B	2	1																													
TAS	S←A	Transmits the contents of the accumulator A to the stack pointer.	1B	2	1																													
TAX	X←A	Transmits the contents of the accumulator A to the index register X.	AA	2	1																													
TAY	Y←A	Transmits the contents of the accumulator A to the index register Y.	A8	2	1																													
TBD	DPR←B	Transmits the contents of the accumulator B to the direct page register.	42 5B	4	2																													
TBS	S←B	Transmits the contents of the accumulator B to the stack pointer.	42 1B	4	2																													
TBX	X←B	Transmits the contents of the accumulator B to the index register X.	42 AA	4	2																													
TBY	Y←B	Transmits the contents of the accumulator B to the index register Y.	42 A8	4	2																													
TDA	A←DPR	Transmits the contents of the direct page register to the accumulator A.	7B	2	1																													
TDB	B←DPR	Transmits the contents of the direct page register to the accumulator B.	42 7B	4	2																													
TSA	A←S	Transmits the contents of the stack pointer to the accumulator A.	3B	2	1																													
TSB	B←S	Transmits the contents of the stack pointer to the accumulator B.	42 3B	4	2																													
TSX	X←S	Transmits the contents of the stack pointer to the index register X.	BA	2	1																													
TXA	A←X	Transmits the contents of the index register X to the accumulator A.	8A	2	1																													
TXB	B←X	Transmits the contents of the index register X to the accumulator B.	42 8A	4	2																													
TXS	S←X	Transmits the contents of the index register X to the stack pointer.	9A	2	1																													
TXY	Y←X	Transmits the contents of the index register X to the index register Y.	9B	2	1																													
TYA	A←Y	Transmits the contents of the index register Y to the accumulator A.	98	2	1																													
TYB	B←Y	Transmits the contents of the index register Y to the accumulator B.	42 98	4	2																													
TYX	X←Y	Transmits the contents of the index register Y to the index register X.	BB	2	1																													
WIT		Stops the internal clock.	CB	3	1																													
XAB	A⇄B	Exchanges the contents of the accumulator A and the contents of the accumulator B.	89 28	6	2																													

Addressing mode																																																						Processor status register										
L(DIR)			L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	op	n	#	IPL			N	V	m	x	D	I	Z	C
									0C	9	4																																											•	•	•	•	•	•	•	•	•	•	•
																																																						•	•	•	•	•	•	•	•	•	•	1
																																																						•	•	•	•	•	•	•	•	1	•	•
																																																						•	•	•	•	•	1	•	•	•	•	•
																																																						•	•	•	Specified flag becomes "1".							
87	10	2	97	11	2	8D	5	3				9D	5	3	99	5	3	8F	6	4	9F	7	4																						83	5	2	93	8	2				•	•	•	•	•	•	•	•	•	•	•
42 87	12	3	42 97	13	3	42 8D	7	4				42 9D	7	4	42 99	7	4	42 8F	8	5	42 9F	9	5																						42 83	7	3	42 93	10	3														
																																																						•	•	•	•	•	•	•	•	•	•	•
						8E	5	3																																														•	•	•	•	•	•	•	•	•	•	•
						8C	5	3																																														•	•	•	•	•	•	•	•	•	•	•
																																																						•	•	•	•	•	•	•	•	•	•	•
																																																						•	•	•	•	•	•	•	•	•	•	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	•	•	•	•	•	•	•	•
																																																						•	•	•	•	•	•	•	•	•	•	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	•	•	•	•	•	•	•	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	•	•	•	•	•	•	•	•
																																																						•	•	•	N	•	•	•	•	•	Z	•

Symbols in machine instructions table

Symbol	Description	Symbol	Description
IMP	Implied addressing mode	$\veebar$	Exclusive OR
IMM	Immediate addressing mode	—	Negation
A	Accumulator addressing mode	←	Movement to the arrow direction
DIR	Direct addressing mode	A_{CC}	Accumulator
DIR, b	Direct bit addressing mode	A_{CCH}	Accumulator's upper 8 bits
DIR, X	Direct indexed X addressing mode	A_{CCL}	Accumulator's lower 8 bits
DIR, Y	Direct indexed Y addressing mode	A	Accumulator A
(DIR)	Direct indirect addressing mode	A_H	Accumulator A's upper 8 bits
(DIR, X)	Direct indexed X indirect addressing mode	A_L	Accumulator A's lower 8 bits
(DIR), Y	Direct indirect indexed Y addressing mode	B	Accumulator B
L (DIR)	Direct indirect long addressing mode	B_H	Accumulator B's upper 8 bits
L (DIR), Y	Direct indirect long indexed Y addressing mode	B_L	Accumulator B's lower 8 bits
ABS	Absolute addressing mode	X	Index register X
ABS, b	Absolute bit addressing mode	X_H	Index register X's upper 8 bits
ABS, X	Absolute indexed X addressing mode	X_L	Index register X's lower 8 bits
ABS, Y	Absolute indexed Y addressing mode	Y	Index register Y
ABL	Absolute long addressing mode	Y_H	Index register Y's upper 8 bits
ABL, X	Absolute long indexed X addressing mode	Y_L	Index register Y's lower 8 bits
(ABS)	Absolute indirect addressing mode	S	Stack pointer
L (ABS)	Absolute indirect long addressing mode	PC	Program counter
(ABS, X)	Absolute indexed X indirect addressing mode	PC_H	Program counter's upper 8 bits
STK	Stack addressing mode	PC_L	Program counter's lower 8 bits
REL	Relative addressing mode	PG	Program bank register
DIR, b, REL	Direct bit relative addressing mode	DT	Data bank register
ABS, b, REL	Absolute bit relative addressing mode	DPR	Direct page register
SR	Stack pointer relative addressing mode	DPR_H	Direct page register's upper 8 bits
(SR), Y	Stack pointer relative indirect indexed Y addressing mode	DPR_L	Direct page register's lower 8 bits
		PS	Processor status register
BLK	Block transfer addressing mode	PS_H	Processor status register's upper 8 bits
C	Carry flag	PS_L	Processor status register's lower 8 bits
Z	Zero flag	PS_b	Processor status register's b-th bit
I	Interrupt disable flag	M(S)	Contents of memory at address indicated by stack pointer
D	Decimal operation mode flag		
x	Index register length selection flag	M_b	b-th memory location
m	Data length selection flag	AD_G	Value of 24-bit address's upper 8-bit (A_{23}~A_{16})
V	Overflow flag	AD_H	Value of 24-bit address's middle 8-bit (A_{15}~A_8)
N	Negative flag	AD_L	Value of 24-bit address's lower 8-bit (A_7~A_0)
IPL	Processor interrupt priority level	op	Operation code
+	Addition	n	Number of cycle
−	Subtraction	#	Number of byte
*	Multiplication	i	Number of transfer byte or rotation
/	Division	i_1, i_2	Number of registers pushed or pulled
∧	Logical AND		
∨	Logical OR		

The number of cycles shown in the table is described in case of the fastest mode for each instruction. The number of cycles shown in the table is calculated for DPR_L=0. The number of cycles in the addressing mode concerning the DPR when $DPR_L \neq 0$ must be incremented by 1.
The number of cycles shown in the table differs according to the bytes fetched into the instruction queue buffer, or according to whether the memory read/write address is odd or even. It also differs when the external region memory is accessed by BYTE="H".

Note 1. The operation code at the upper row is used for accumulator A, and the operation at the lower row is used for accumulator B.

Note 2. When setting flag m=0 to handle the data as 16-bit data in the immediate addressing mode, the number of bytes increments by 1.

Note 3. The number of cycles increments by 2 when branching.

Note 4. The operation code on the upper row is used for branching in the range of −128~+127, and the operation code on the lower row is used for branching in the range of −32768~+32767.

Note 5. When handling 16-bit data with flag m=0, the byte in the table is incremented by 1.

Note 6.

Type of register	A	B	X	Y	DPR	DT	PG	PS
Number of cycles	2	2	2	2	2	1	1	2

The number of cycles corresponding to the register to be pushed are added. The number of cycles when no pushing is done is 12. i_1 indicates the number of registers among A, B, X, Y, DPR, and PS to be saved, while i_2 indicates the number of registers among DT and PG to be saved.

Note 7.

Type of register	A	B	X	Y	DPR	DT	PS
Number of cycles	3	3	3	3	4	3	3

The number of cycles corresponding to the register to be pulled are added. The number of cycles when no pulling is done is 14. i_1 indicates the number of registers among A, B, X, Y, DT, and PS to be restored, while i_2=1 when DPR is to be restored.

Note 8. The number of cycles is the case when the number of bytes to be transfered is even.
When the number of bytes to be transfered is odd, the number is calculated as;

$$7 + (i/2) \times 7 + 4$$

Note that, (i/2) shows the integer part when i is divided by 2.

Note 9. The number of cycles is the case when the number of bytes to be transfered is even.
When the number of bytes to be transfered is odd, the number is calculated as;

$$9 + (i/2) \times 7 + 5$$

Note that, (i/2) shows the integer part when i is divided by 2.

Note 10. The number of cycles is the case in the 16-bit÷8-bit operation. The number of cycles is incremented by 16 for 32-bit÷16-bit operation.

Note 11. The number of cycles is the case in the 8-bit×8-bit operation. The number of cycles is incremented by 8 for 16-bit ×16-bit operation.

Note 12. When setting flag x=0 to handle the data as 16-bit data in the immediate addressing mode, the number of bytes increments by 1.

Note 13. When flag m is 0, the byte in the table is incremented by 1.

PROGRAMMABLE ROM MICROCOMPUTERS

3

M37702E2-XXXFP,M37702E2AXXXFP,M37702E2BXXXFP
M37702E2FS,M37702E2AFS,M37702E2BFS

PROM VERSION of M37702M2-XXXFP,M37702M2AXXXFP,M37702M2BXXXFP

DESCRIPTION

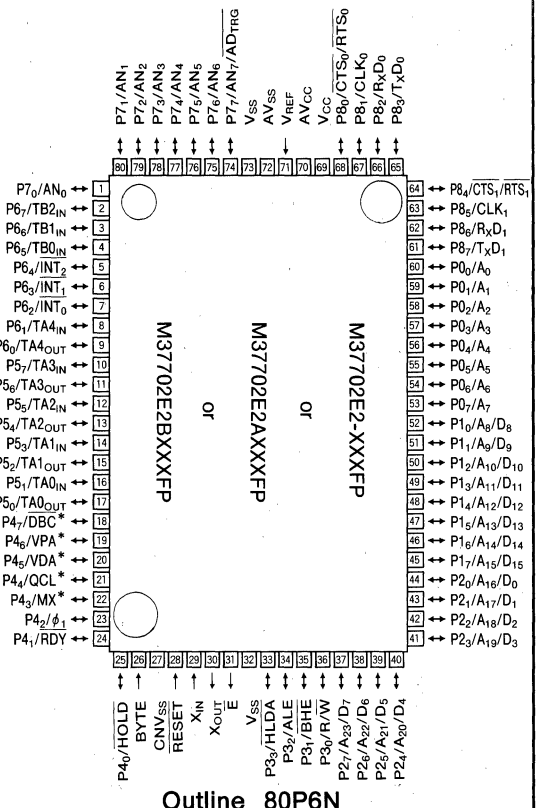
The M37702E2-XXXFP, M37702E2AXXXFP and M37702E2BXXXFP are single-chip microcomputers designed with high-performance CMOS silicon gate technology. These are housed in a 80-pin plastic molded QFP. The features of these chips are similar to those of the M37702M2-XXXFP, M37702M2AXXXFP and M37702M2BXXXFP except that these chips have a 16K-byte PROM built in.

These single-chip microcomputers have a large 16M bytes address space, 3-byte instruction queue buffers, and 2-byte data buffers for high-speed instruction execution. The CPU is a 16-bit parallel processor that can also be switched to perform 8-bit parallel processing. These microcomputers are suitable for office, business and industrial equipment controller that require high-speed processing of large data. Since general purpose PROM writers can be used for the built-in PROM, this chip is suitable for small quantity production runs. The M37702E2FS (8MHz version), M37702E2AFS (16MHz version) and M37702E2BFS (25MHz version) with erasable ROM that are housed in a windowed ceramic LCC are also provided.

The differences between the M37702E2-XXXFP, M37702E2AXXXFP and M37702E2BXXXFP are the external clock input frequency as shown below. Therefore, the following descriptions will be for the M37702E2-XXXFP unless otherwise noted.

Type name	External clock input frequency
M37702E2-XXXFP	8 MHz
M37702E2AXXXFP	16MHz
M37702E2BXXXFP	25MHz

PIN CONFIGURATION (TOP VIEW)



Outline 80P6N

*: Used in the evaluation chip mode only

DISTINCTIVE FEATURES

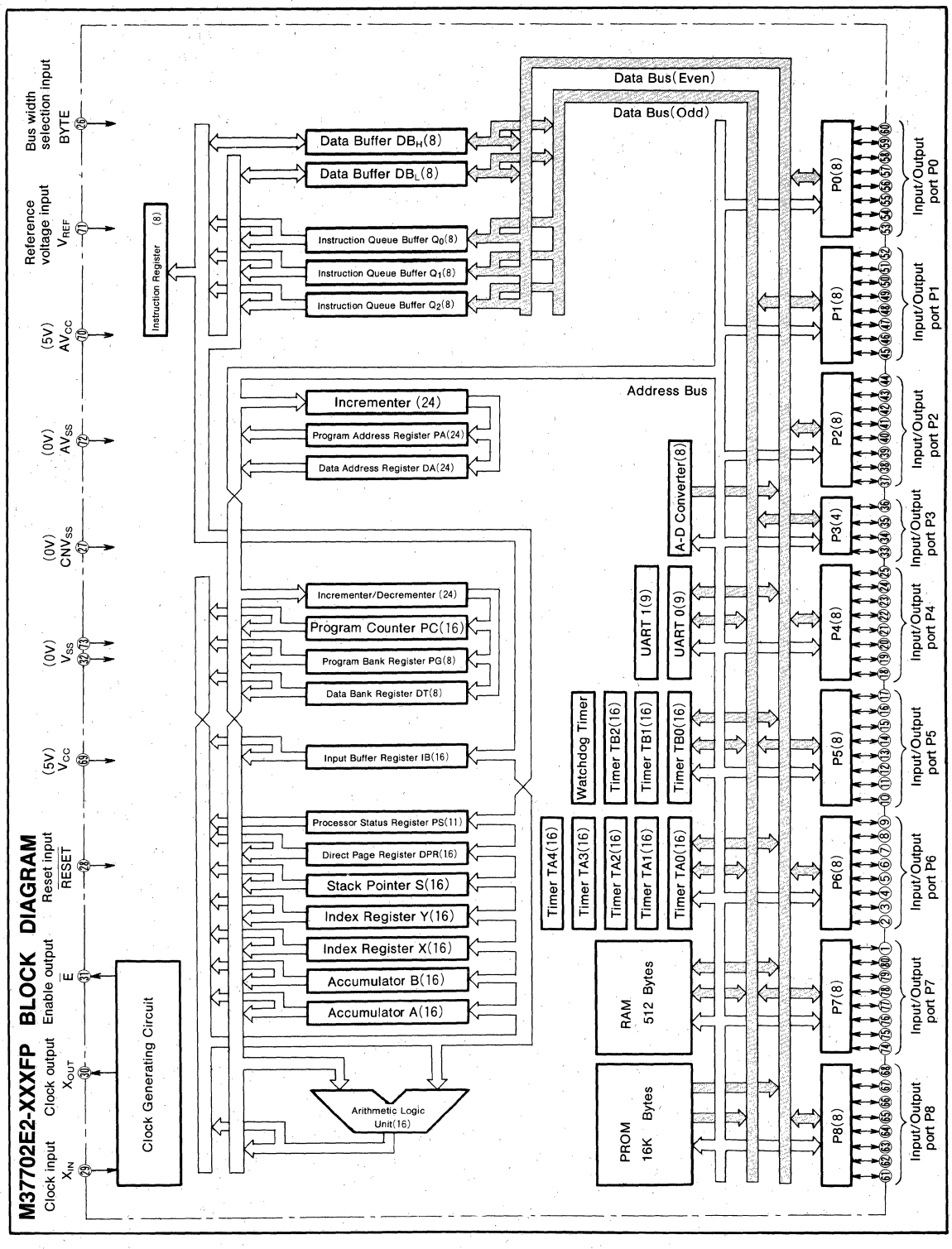
- Number of basic instructions 103
- Memory size PROM ..... 16K bytes
  RAM ..... 512 bytes
- Instruction execution time
  M37702E2-XXXFP
  (The fastest instruction at 8 MHz frequency) 500ns
  M37702E2AXXXFP
  (The fastest instruction at 16 MHz frequency) 250ns
  M37702E2BXXXFP
  (The fastest instruction at 25 MHz frequency) 160ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 8 MHz frequency) ..... 30mW (Typ.)
- Interrupts ..... 19 types 7 levels
- Multiple function 16-bit timer ..... 5+3
- UART (may also be synchronous) 2
- 8-bit A-D converter ..... 8-channel inputs
- 12-bit watchdog timer
- Programmable input/output
  (ports P0, P1, P2, P3, P4, P5, P6, P7, P8) 68

APPLICATION

Control devices for office equipment such as copiers, printers, typewriters, facsimiles, word processors, and personal computers

Control devices for industrial equipment such as ME and NC, communication, and measuring instruments

PROM VERSION of M37702M2-XXXFP,M37702M2AXXXFP,M37702M2BXXXFP



MITSUBISHI ELECTRIC

FUNCTIONS OF M37702E2-XXXFP

Parameter		Functions
Number of basic instructions		103
Instruction execution time	M37702E2-XXXFP, M37702E2FS	500ns (the fastest instructions, at 8MHz frequency)
	M37702E2AXXXFP, M37702E2AFS	250ns (the fastest instructions, at 16MHz frequency)
	M37702E2BXXXFP, M37702E2BFS	160ns (the fastest instructions, at 25MHz frequency)
Memory size	PROM	16K bytes
	RAM	512 bytes
Input/Output ports	P0～P2, P4～P8	8-bit× 8
	P3	4-bit× 1
Multi-function timers	TA0, TA1, TA2, TA3, TA4	16-bit× 5
	TB0, TB1, TB2	16-bit× 3
Serial I/O		(UART or clock synchronous serial I/O)×2
A-D converter		8-bit× 1 (8 channels)
Watchdog timer		12-bit× 1
Interrupts		3 external types, 16 internal types (Each interrupt can be set the priority levels to 0～7.)
Clock generating circuit		Built-in(externally connected to a ceramic resonator or quartz crystal resonator)
Supply voltage		5 V±10%
Power dissipation		30mW(at external 8 MHz frequency)
Input/Output characteristic	Input/Output voltage	5 V
	Output current	5 mA
Memory expansion		Maximum 16M bytes
Operating temperature range		−20～85℃
Device structure		CMOS high-performance silicon gate process
Package	M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP	80-pin plastic molded QFP
	M37702E2FS, M37702E2AFS,M37702E2BFS	80-pin ceramic LCC (with a window)

PIN DESCRIPTION (NORMAL MODE)

Pin	Name	Input/Output	Functions
V_{CC}, V_{SS}	Power supply		Supply 5 V±10% to V_{CC} and 0 V to V_{SS}.
CNV_{SS}	CNV_{SS} input	Input	This pin controls the processor mode. Connect to V_{SS} for single-chip mode.
$\overline{RESET}$	Reset input	Input	To enter the reset state, this pin must be kept at a "L" condition which should be maintained for the required time.
X_{IN}	Clock input	Input	These are I/O pins of internal clock generating circuit. Connect a ceramic or quartz crystal resonator between X_{IN} and X_{OUT}. When an external clock is used, the clock source should be connected to the X_{IN} pin and the X_{OUT} pin should be left open.
X_{OUT}	Clock output	Output	
$\overline{E}$	Enable output	Output	Data or instruction read and data write are performed when output from this pin is "L".
BYTE	Bus width selection input	Input	In memory expansion mode or microprocessor mode, this pin determines whether the external data bus is 8-bit width or 16-bit width. The width is 16 bits when "L" signal inputs and 8 bits when "H" signal inputs.
AV_{CC}, AV_{SS}	Analog supply input		Power supply for the A-D converter. Connect AV_{CC} to V_{CC} and AV_{SS} to V_{SS} externally.
V_{REF}	Reference voltage input	Input	This is reference voltage input pin for the A-D converter.
$P0_0$~$P0_7$	I/O port P0	I/O	In single-chip mode, port P0 becomes an 8-bit I/O port. An I/O directional register is available so that each pin can be programmed for input or output. These ports are in input mode when reset . Address(A_7~A_0)is output in memory expansion mode or microprocessor mode.
$P1_0$~$P1_7$	I/O port P1	I/O	In single-chip mode, these pins have the same functions as port P0. When the BYTE pin is set to "L" in memory expansion mode or microprocessor mode and external data bus is 16-bit width, high-order data (D_{15}~D_8)is input or output when $\overline{E}$ output is "L" and an address (A_{15}~A_8) is output when $\overline{E}$ output is "H". If the BYTE pin is "H" that is an external data bus is 8-bit width, only address(A_{15}~A_8)is output.
$P2_0$~$P2_7$	I/O port P2	I/O	In single-chip mode, these pins have the same functions as port P0. In memory expansion mode or microprocessor mode low-order data(D_7~D_0) is input or output when $\overline{E}$ output is "L" and an address(A_{23}~A_{16})is output when $\overline{E}$ output is "H".
$P3_0$~$P3_3$	I/O port P3	I/O	In single-chip mode , these pins have the same functions as port P0. In memory expansion mode or microprocessor mode, R/$\overline{W}$, $\overline{BHE}$, ALE, and $\overline{HLDA}$ signals are output.
$P4_0$~$P4_7$	I/O port P4	I/O	In single-chip mode, these pins have the same functions as port P0. In memory expansion mode or microprocessor mode, $P4_0$ and $P4_1$ become $\overline{HOLD}$ and $\overline{RDY}$ input pin respectively. Functions of other pins are the same as in single-chip mode. In single-chip mode or memory expansion mode,port $P4_2$ can be programed for ϕ_1 output pin divided the clock to X_{IN} pin by 2. In microprocessor mode, $P4_2$ always has the function as ϕ_1 output pin.
$P5_0$~$P5_7$	I/O port P5	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as I/O pins for timer A0, timer A1, timer A2, and timer A3.
$P6_0$~$P6_7$	I/O port P6	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as I/O pins for timer A4, external interrupt input $\overline{INT_0}$, $\overline{INT_1}$, and $\overline{INT_2}$ pins, and input pins for timer B0, timer B1, and timer B2.
$P7_0$~$P7_7$	I/O port P7	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as analog input AN_0~AN_7 input pins. $P7_7$ also has an A-D conversion trigger input function.
$P8_0$~$P8_7$	I/O port P8	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as R_XD, T_XD, CLK, $\overline{CTS}$/$\overline{RTS}$ pins for UART 0 and UART 1.

PIN DESCRIPTION (EPROM MODE)

Pin	Name	Input/Output	Functions
V_{CC}, V_{SS}	Power supply		Supply 5 V±10% to V_{CC} and 0 V to V_{SS}.
CNV_{SS}	V_{PP} input	Input	Connect to V_{PP} when programming or verifing.
BYTE	V_{PP} input	Input	Connect to V_{PP} when programming or verifing.
$\overline{RESET}$	Reset input	Input	Connect to V_{SS}.
X_{IN}	Clock input	Input	Connect a ceramic resonator between X_{IN} and X_{OUT}.
X_{OUT}	Clock output	Output	
$\overline{E}$	Enable output	Output	Keep open.
AV_{CC}, AV_{SS}	A-D power supply	Input	Connect AV_{CC} to V_{CC} and AV_{SS} to V_{SS}.
V_{REF}	Reference voltage input	Input	Connect to V_{SS}.
$P0_0$~$P0_7$	Address input (A_0~A_7)	Input	Port P0 functions as the lower 8 bits address input (A_0~A_7).
$P1_0$~$P1_7$	Address input (A_8~A_{14})	Input	Port $P1_0$~$P1_6$ functions as the higher 7 bits address input (A_8~A_{14}). Connect $P1_7$ to V_{CC}.
$P2_0$~$P2_7$	Data I/O (D_0~D_7)	I/O	Port P2 functions as the 8 bits data bus (D_0~D_7).
$P3_0$~$P3_3$	Input port P3	Input	Connect to V_{SS}.
$P4_0$~$P4_7$	Input port P4	Input	Connect to V_{SS}.
$P5_0$~$P5_7$	Control signal input	Input	$P5_1$ and $P5_2$ functions as $\overline{OE}$ and $\overline{CE}$ input pin. Connect $P5_0$, $P5_3$, $P5_4$ and $P5_5$ to V_{CC}. Connect $P5_6$ and $P5_7$ to V_{SS}.
$P6_0$~$P6_7$	Input port P6	Input	Connect to V_{SS}.
$P7_0$~$P7_7$	Input port P7	Input	Connect to V_{SS}.
$P8_0$~$P8_7$	Input port P8	Input	Connect to V_{SS}.

EPROM MODE

The M37702E2-XXXFP features an EPROM mode in addition to its normal modes. When the RESET signal level is "L", the chip automatically enters the EPROM mode. Table 1 list the correspondence between pins and Fig. 1 shows the pin connections in the EPROM mode.

When in the EPROM mode, ports P0, P1, P2, $P5_1$, $P5_2$, CNV_{SS} and BYTE are used for the EPROM (equivalent to the M5M27C256K). When in this mode, the built-in PROM can be written to or read from using these pins in the same way as with the M5M27C256K.

This chip does not have Device Identifer Mode, so that set the corresponding program algorithm. The program area should specify address 4000_{16}~$7FFF_{16}$ for the M37702E2-XXXFP.

Connect the clock which is either ceramic resonator or external clock to X_{IN} pin and X_{OUT} pin.

Caution :

Describing in this section, the built-in PROM can be written to or read in the same way as with the M5M27C256K (256K mode).

But in the future, for M37702E2BXXXFP and M37702E2BFS, 1M mode way becomes standard.

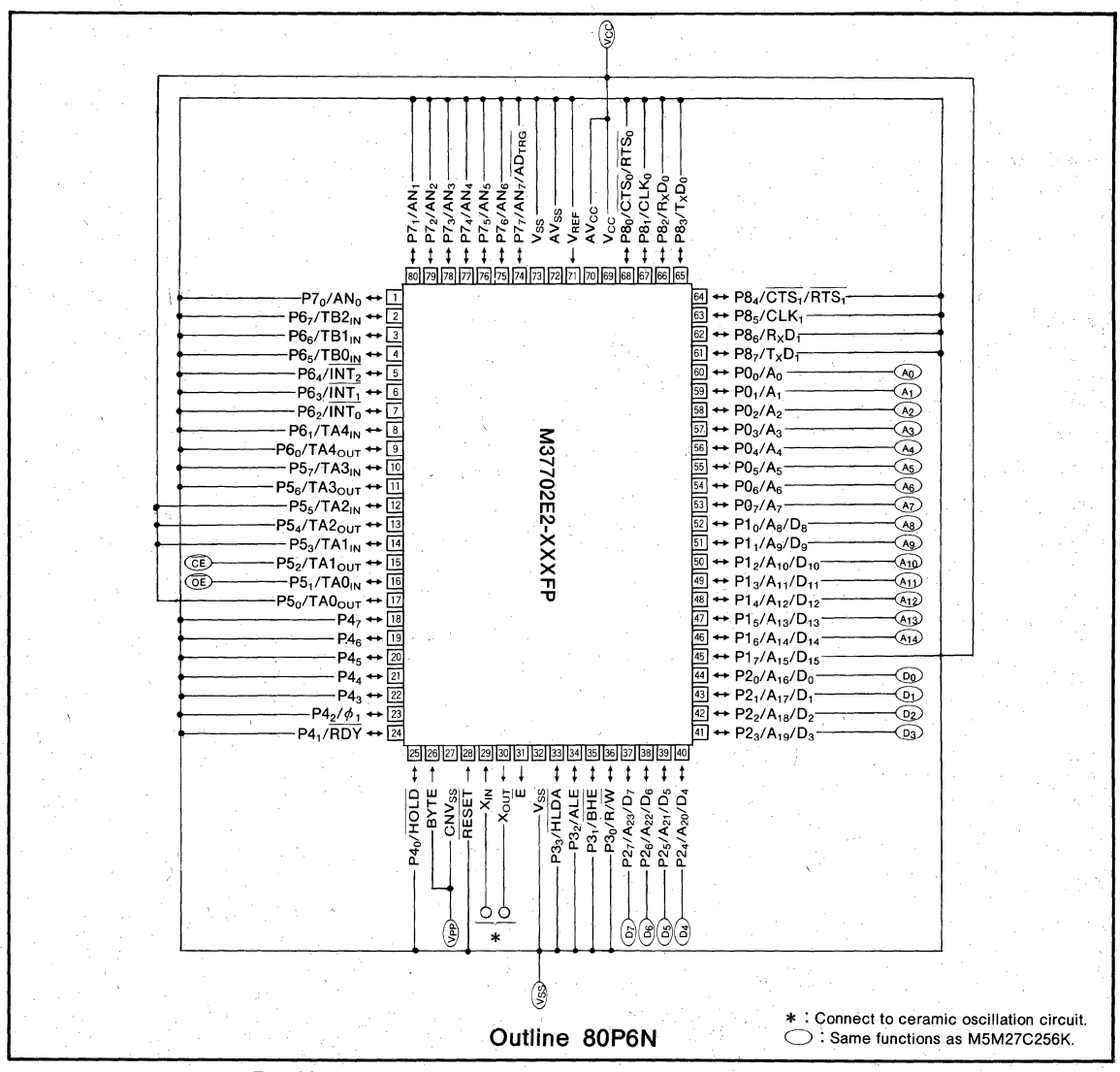


Fig. 1 Pin connection in EPROM programming mode

Table 1 Pin function in EPROM programming mode

	M37702E2-XXXFP	M5M27C256K
V_{CC}	V_{CC}	V_{CC}
V_{PP}	CNV_{SS}, BYTE	V_{PP}
V_{SS}	V_{SS}	V_{SS}
Address input	Ports P0, $P1_0$~$P1_6$	A_0~A_{14}
Data I/O	Port P2	D_0~D_7
$\overline{CE}$	$P5_2$	$\overline{CE}$
$\overline{OE}$	$P5_1$	$\overline{OE}$

FUNCTION IN EPROM MODE

Reading

To read the EPROM, set the $\overline{CE}$ and $\overline{OE}$ pins to a "L" level. Input the address of the data (A_0~A_{14}) to be read and the data will be output to the I/O pins D_0~D_7. The data I/O pins will be floating when either the $\overline{CE}$ or $\overline{OE}$ pins are in the "H" state.

Writing

To write to the EPROM, set the $\overline{OE}$ pin to a "H" level. The CPU will enter the program mode when V_{PP} is applied to the V_{PP} pin. The address to be written to is selected with pins A_0~A_{14}, and the data to be written is input to pins D_0~D_7. Set the $\overline{CE}$ pin to a "L" level to being writing.

Erasing

To erase data on this chip, use an ultraviolet light source with a 2537 Angstrom wave length. The minimum radiation power necessary for erasing is 15W·s/cm^2.
(M37702E2FS, M37702E2AFS, M37702E2BFS)

FAST PROGRAMMING ALGORITHM

To program the M37702E2-XXXFP with fast programming algorithm, first set V_{CC}=6V, V_{PP}=12.5, and set the address to "0". Apply a 1ms write pulse, check that the data can be read, and if it cannot be read OK, repeat the procedure, applying a 1ms write pulse and checking that the data can be read until it can be read OK. Record the accumulated number of pulse applied (N) before the data can be read OK, and then write the data again, applying a further three times this number of pulses (3×N ms).
When this series of write operations is complete, increment the address, and continue to repeat the procedure above until the last address has been reached.
Finally, when all addresses have been written, read with V_{CC}=V_{PP}=5V (or V_{CC}=V_{PP}=5.25V).

Table 2 I/O signal in each mode

Mode \ Pin	$\overline{CE}$	$\overline{OE}$	V_{PP}	V_{CC}	Data I/O
Read-out	V_{IL}	V_{IL}	5 V	5 V	Output
Output Disable	V_{IL}	V_{IH}	5 V	5 V	Floating
	V_{IH}	X	5 V	5 V	Floating
Programming	V_{IL}	V_{IH}	12.5V	6 V	Input
Programming Verify	V_{IH}	V_{IL}	12.5V	6 V	Output
Program Disable	V_{IH}	V_{IH}	12.5V	6 V	Floating

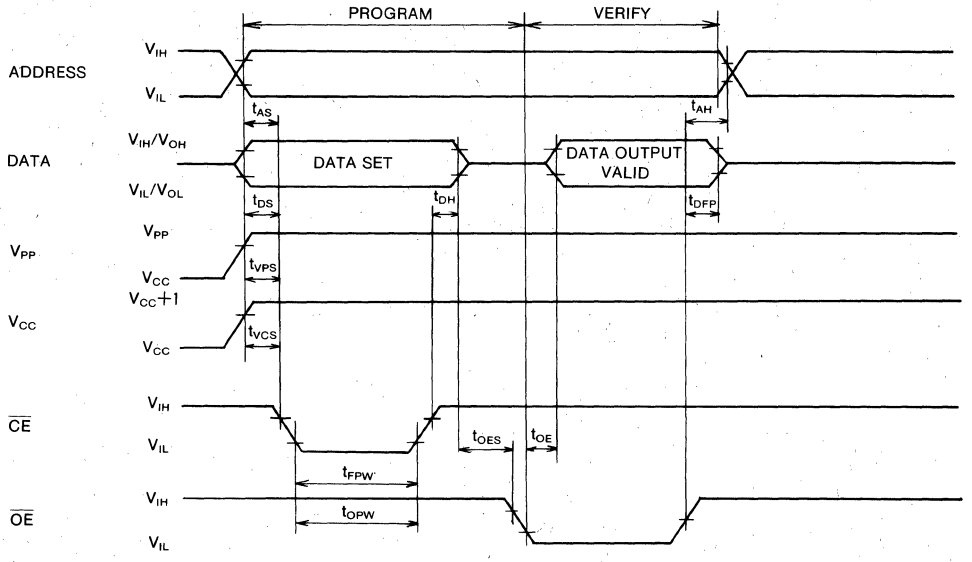
Note 1 : An X indicates either V_{IL} or V_{IH}.

Program operation

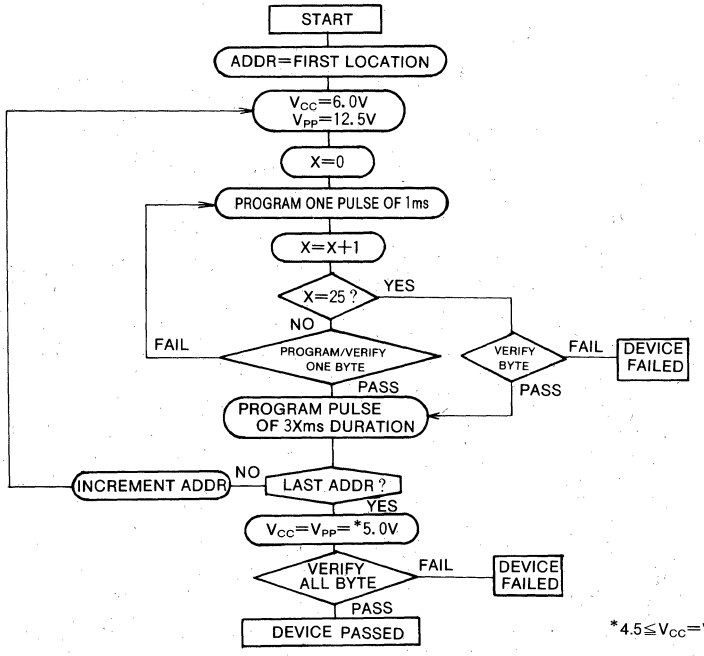
AC ELECTRICAL CHARACTERISTICS (T_a=25±5℃, V_{CC}=6V±0.25V, V_{PP}=12.5±0.3V, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
t_{AS}	Address setup time		2			μs
t_{OES}	$\overline{OE}$ setup time		2			μs
t_{DS}	Data setup time		2			μs
t_{AH}	Address hold time		0			μs
t_{DH}	Data hold time		2			μs
t_{DFP}	Output enable to output float delay		0		130	ns
t_{VCS}	V_{CC} setup time		2			μs
t_{VPS}	V_{PP} setup time		2			μs
t_{FPW}	$\overline{CE}$ initial program pulse width		0.95	1	1.05	ms
t_{OPW}	$\overline{CE}$ over program pulse width		2.85		78.75	ms
t_{OE}	Data valid from $\overline{OE}$				150	ns

AC waveforms



Fast programming algorithm flow chart



SAFETY INSTRUCTIONS

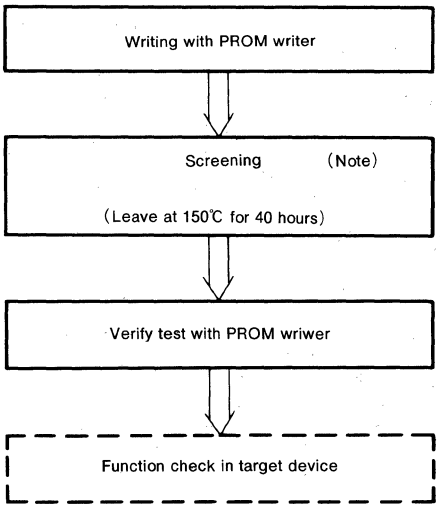
(1) Sunlight and fluorescent lamp contain light that can erase written information. When using in read mode, be sure to cover the transparent glass portion with a seal or other materials (ceramic package product).

(2) Mitsubishi Electric corp. provides the seal for covering the transparent glass. Take care that the seal does not touch the read pins (ceramic package product).

(3) Clean the transparent glass before erasing. Fingers' flat and paste disturb the passage of ultraviolet rays and may affect badly the erasure capability (ceramic package product).

(4) A high voltage is used for writing. Take care that over-voltage is not applied. Take care especially at power on.

(5) The programmable M37702E2FP, M37702E2AFP and M37702E2BFP that are shipped in blank are also provided. For the M37702E2FP, M37702E2AFP and M37702E2BFP, Mitsubishi Electric corp. does not perform PROM write test and screening in the assembly process and following processes. To improve reliability after write, performing write and test according to the flow below before use is recommended.



Note : Never expose to 150℃ exceeding 100 hours.

(6) Use a fit IC socket to mountain the ceramic package product except for evaluation. Settle the ceramic package in an IC socket with silicon resin and the like, surely.

BASIC FUNCTION BLOCKS

Since these processors operate in exactly the same way as the M37702M2-XXXFP, refer to the section on the M37702M2-XXXFP.

ADDRESSING MODES

The M37702E2-XXXFP has 28 powerful addressing modes. Refer to the MELPS 7700 addressing mode description for the details of each addressing mode.

MACHINE INSTRUCTION LIST

The M37702E2-XXXFP has 103 machine instructions. Refer to the MELPS 7700 machine instruction list for details.

DATA REQUIRED FOR PROM ORDERING

Please send the following data for writing to PROM.

(1) M37702E2-XXXFP writing to PROM order confirmation form

(2) 80P6N mark specification form

(3) ROM data (EPROM 3sets)

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Supply voltage		−0.3∼7	V
AV_{CC}	Analog supply voltage		−0.3∼7	V
V_I	Input voltage $\overline{RESET}$, CNV_{SS}, BYTE		−0.3∼12	V
V_I	Input voltage $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_3$, $P4_0$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, V_{REF}, X_{IN}		−0.3∼V_{CC}+0.3	V
V_O	Output voltage $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_3$, $P4_0$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, X_{OUT}, $\overline{E}$		−0.3∼V_{CC}+0.3	V
P_d	Power dissipation	T_a=25℃	300	mW
T_{opr}	Operating temperature		−20∼85	℃
T_{stg}	Storage temperature		−40∼150	℃

RECOMMENDED OPERATING CONDITIONS (V_{CC}=5V±10%, T_a=−20∼85℃, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
V_{CC}	Supply voltage	4.5	5.0	5.5	V
AV_{CC}	Analog supply voltage		V_{CC}		V
V_{SS}	Supply voltage		0		V
AV_{SS}	Analog supply voltage		0		V
V_{IH}	High-level input voltage $P0_0$∼$P0_7$, $P3_0$∼$P3_3$, $P4_0$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	0.8V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage $P1_0$∼$P1_7$, $P2_0$∼$P2_7$ (in single-chip mode)	0.8V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage $P1_0$∼$P1_7$, $P2_0$∼$P2_7$ (in memory expansion mode and microprocessor mode)	0.5V_{CC}		V_{CC}	V
V_{IL}	Low-level input voltage $P0_0$∼$P0_7$, $P3_0$∼$P3_3$, $P4_0$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	0		0.2V_{CC}	V
V_{IL}	Low-level input voltage $P1_0$∼$P1_7$, $P2_0$∼$P2_7$ (in single-chip mode)	0		0.2V_{CC}	V
V_{IL}	Low-level input voltage $P1_0$∼$P1_7$, $P2_0$∼$P2_7$ (in memory expansion mode and microprocessor mode)	0		0.16V_{CC}	V
$I_{OH(peak)}$	High-level peak output current $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_3$, $P4_0$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$			−10	mA
$I_{OH(avg)}$	High-level average output current $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_3$, $P4_0$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$			−5	mA
$I_{OL(peak)}$	Low-level peak output current $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_3$, $P4_0$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$			10	mA
$I_{OL(avg)}$	Low-level average output current $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_3$, $P4_0$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$			5	mA
$f(X_{IN})$	External clock frequency input — M37702E2-XXXFP, M37702E2FS			8	MHz
	External clock frequency input — M37702E2AXXXFP, M37702E2AFS			16	MHz
	External clock frequency input — M37702E2BXXXFP, M37702E2BFS			25	MHz

Note 1. Average output current is the average value of a 100ms interval.
2. The sum of $I_{OL(peak)}$ for ports P0, P1, P2, P3, and P8 must be 80mA or less, the sum of $I_{OH(peak)}$ for ports P0, P1, P2, P3, and P8 must be 80mA or less, the sum of $I_{OL(peak)}$ for ports P4, P5, P6, and P7 must be 80mA or less, and the sum of $I_{OH(peak)}$ for ports P4, P5, P6, and P7 must be 80mA or less.

M37702E2-XXXFP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=8MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$	I_{OH}=−10mA	3			V
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$	I_{OH}=−400μA	4.7			V
V_{OH}	High-level output voltage $P3_2$	I_{OH}=−10mA	3.1			V
		I_{OH}=−400μA	4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA	3.4			V
		I_{OH}=−400μA	4.8			
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$	I_{OL}=10mA			2	V
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$	I_{OL}=2mA			0.45	V
V_{OL}	Low-level output voltage $P3_2$	I_{OL}=10mA			1.9	V
		I_{OL}=2mA			0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA			1.6	V
		I_{OL}=2mA			0.4	
V_{T+}−V_{T-}	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA4_{IN}$, $TB0_{IN}$~$TB2_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, AD_{TRG}, $\overline{CTS_0}$, $\overline{CTS_1}$, CLK_0, CLK_1		0.4		1	V
V_{T+}−V_{T-}	Hysteresis $\overline{RESET}$		0.2		0.5	V
V_{T+}−V_{T-}	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=5V			5	μA
I_{IL}	Low-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=0V			−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	In single-chip mode output only pin is open and other pins are V_{SS} during reset. f(X_{IN})=8MHz, square waveform		6	12	mA
		T_a=25℃ when clock is stopped.			1	μA
		T_a=85℃ when clock is stopped.			20	

A-D CONVERTER CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=8MHz, unless otherwise noted)

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

M37702E2AXXXFP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=16MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$	I_{OH}=−10mA	3			V
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$	I_{OH}=−400μA	4.7			V
V_{OH}	High-level output voltage $P3_2$	I_{OH}=−10mA	3.1			V
		I_{OH}=−400μA	4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA	3.4			V
		I_{OH}=−400μA	4.8			
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$	I_{OL}=10mA			2	V
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$	I_{OL}=2mA			0.45	V
V_{OL}	Low-level output voltage $P3_2$	I_{OL}=10mA			1.9	V
		I_{OL}=2mA			0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA			1.6	V
		I_{OL}=2mA			0.4	
V_{T+}−V_{T-}	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA4_{IN}$, $TB0_{IN}$~$TB2_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, AD_{TRG}, $\overline{CTS_0}$, $\overline{CTS_1}$, CLK_0, CLK_1		0.4		1	V
V_{T+}−V_{T-}	Hysteresis $\overline{RESET}$		0.2		0.5	V
V_{T+}−V_{T-}	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=5V			5	μA
I_{IL}	Low-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=0V			−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	In single-chip mode output only pin is open and other pins are V_{SS} during reset. f(X_{IN})=16MHz, square waveform		12	24	mA
		T_a=25℃ when clock is stopped.			1	μA
		T_a=85℃ when clock is stopped.			20	

A-D CONVERTER CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=16MHz, unless otherwise noted)

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

M37702E2BXXXFP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=25MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits Min.	Limits Typ.	Limits Max.	Unit
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$	I_{OH}=−10mA	3			V
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$	I_{OH}=−400μA	4.7			V
V_{OH}	High-level output voltage $P3_2$	I_{OH}=−10mA	3.1			V
		I_{OH}=−400μA	4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA	3.4			V
		I_{OH}=−400μA	4.8			
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$	I_{OL}=10mA			2	V
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P3_3$	I_{OL}=2mA			0.45	V
V_{OL}	Low-level output voltage $P3_2$	I_{OL}=10mA			1.9	V
		I_{OL}=2mA			0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA			1.6	V
		I_{OL}=2mA			0.4	
$V_{T+}-V_{T-}$	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA4_{IN}$, $TB0_{IN}$~$TB2_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, $\overline{AD_{TRG}}$, $\overline{CTS_0}$, $\overline{CTS_1}$, CLK_0, CLK_1		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis $\overline{RESET}$		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=5V			5	μA
I_{IL}	Low-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_3$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=0V			−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	In single-chip mode output only pin is open and other pins are V_{SS} during reset. f(X_{IN})=25MHz, square waveform		19	38	mA
		T_a=25℃ when clock is stopped.			1	μA
		T_a=85℃ when clock is stopped.			20	

A-D CONVERTER CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=25MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits Min.	Limits Typ.	Limits Max.	Unit
—	Resolution	V_{REF}=V_{CC}			8	Bits
—	Absolute accuracy	V_{REF}=V_{CC}			±3	LSB
R_{LADDER}	Ladder resistance	V_{REF}=V_{CC}	2		10	kΩ
t_{CONV}	Conversion time		9.12			μs
V_{REF}	Reference voltage		2		V_{CC}	V
V_{IA}	Analog input voltage		0		V_{REF}	V

TIMING REQUIREMENTS (V_{CC}=5V±10%, V_{SS}=0V, T_a=25°C, unless otherwise noted)

External clock input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
t_C	External clock input cycle time	125		62		40		ns
$t_{W(H)}$	External clock input high-level pulse width	50		25		15		ns
$t_{W(L)}$	External clock input low-level pulse width	50		25		15		ns
t_r	External clock rise time		20		10		8	ns
t_f	External clock fall time		20		10		8	ns

Single-chip mode

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{su(P0D-E)}$	Port P0 input setup time	200		100		60		ns
$t_{su(P1D-E)}$	Port P1 input setup time	200		100		60		ns
$t_{su(P2D-E)}$	Port P2 input setup time	200		100		60		ns
$t_{su(P3D-E)}$	Port P3 input setup time	200		100		60		ns
$t_{su(P4D-E)}$	Port P4 input setup time	200		100		60		ns
$t_{su(P5D-E)}$	Port P5 input setup time	200		100		60		ns
$t_{su(P6D-E)}$	Port P6 input setup time	200		100		60		ns
$t_{su(P7D-E)}$	Port P7 input setup time	200		100		60		ns
$t_{su(P8D-E)}$	Port P8 input setup time	200		100		60		ns
$t_{h(E-P0D)}$	Port P0 input hold time	0		0		0		ns
$t_{h(E-P1D)}$	Port P1 input hold time	0		0		0		ns
$t_{h(E-P2D)}$	Port P2 input hold time	0		0		0		ns
$t_{h(E-P3D)}$	Port P3 input hold time	0		0		0		ns
$t_{h(E-P4D)}$	Port P4 input hold time	0		0		0		ns
$t_{h(E-P5D)}$	Port P5 input hold time	0		0		0		ns
$t_{h(E-P6D)}$	Port P6 input hold time	0		0		0		ns
$t_{h(E-P7D)}$	Port P7 input hold time	0		0		0		ns
$t_{h(E-P8D)}$	Port P8 input hold time	0		0		0		ns

Memory expansion mode and microprocessor mode

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{su(P1D-E)}$	Port P1 input setup time	60		45		30		ns
$t_{su(P2D-E)}$	Port P2 input setup time	60		45		30		ns
$t_{su(RDY-\phi_1)}$	$\overline{RDY}$ input setup time	70		60		55		ns
$t_{su(HOLD-\phi_1)}$	$\overline{HOLD}$ input setup time	70		60		55		ns
$t_{h(E-P1D)}$	Port P1 input hold time	0		0		0		ns
$t_{h(E-P2D)}$	Port P2 input hold time	0		0		0		ns
$t_{h(\phi_1-RDY)}$	$\overline{RDY}$ input hold time	0		0		0		ns
$t_{h(\phi_1-HOLD)}$	$\overline{HOLD}$ input hold time	0		0		0		ns

Timer A input (Count input in event counter mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	250		125		80		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	125		62		40		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	125		62		40		ns

Timer A input (Gating input in timer mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	1000		500		320		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	500		250		160		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	500		250		160		ns

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

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	500		250		160		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	250		125		80		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	250		125		80		ns

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

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	250		125		80		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	250		125		80		ns

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

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(UP)}$	TAi_{OUT} input cycle time	5000		2500		2000		ns
$t_{W(UPH)}$	TAi_{OUT} input high-level pulse width	2500		1250		1000		ns
$t_{W(UPL)}$	TAi_{OUT} input low-level pulse width	2500		1250		1000		ns
$t_{SU(UP-T_{IN})}$	TAi_{OUT} input setup time	1000		500		400		ns
$t_{h(T_{IN}-UP)}$	TAi_{OUT} input hold time	1000		500		400		ns

Timer B input (Count input in event counter mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	TBi_{IN} input cycle time (one edge count)	250		125		80		ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width (one edge count)	125		62		40		ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width (one edge count)	125		62		40		ns
$t_{C(TB)}$	TBi_{IN} input cycle time (both edges count)	500		250		160		ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width (both edges count)	250		125		80		ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width (both edges count)	250		125		80		ns

Timer B input (Pulse period measurement mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	TBi_{IN} input cycle time	1000		500		320		ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width	500		250		160		ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width	500		250		160		ns

Timer B input (Pulse width measurement mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	TBi_{IN} input cycle time	1000		500		320		ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width	500		250		160		ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width	500		250		160		ns

A-D trigger input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(AD)}$	$\overline{AD_{TRG}}$ input cycle time (minimum allowable trigger)	2000		1000		1000		ns
$t_{W(ADL)}$	$\overline{AD_{TRG}}$ input low-level pulse width	250		125		125		ns

Serial I/O

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(CK)}$	CLK_i input cycle time	500		250		200		ns
$t_{W(CKH)}$	CLK_i input high-level pulse width	250		125		100		ns
$t_{W(CKL)}$	CLK_i input low-level pulse width	250		125		100		ns
$t_{d(C-Q)}$	TxD_i output delay time		150		90		80	ns
$t_{h(C-Q)}$	TxD_i hold time	30		30		30		ns
$t_{su(D-C)}$	RxD_i input setup time	60		30		20		ns
$t_{h(C-D)}$	RxD_i input hold time	90		90		90		ns

External interrupt $\overline{INT_i}$ input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{W(INH)}$	$\overline{INT_i}$ input high-level pulse width	250		250		250		ns
$t_{W(INL)}$	$\overline{INT_i}$ input low-level pulse width	250		250		250		ns

SWITCHING CHARACTERISTICS (V_{CC}=5V±10%, V_{SS}=0V, T_a=25℃, unless otherwise noted)

Single-chip mode

Symbol	Parameter	Test conditions	Limits						Unit
			8MHz		16MHz		25MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(E-P0Q)}$	Port P0 data output delay time	Fig. 2		200		100		80	ns
$t_{d(E-P1Q)}$	Port P1 data output delay time			200		100		80	ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			200		100		80	ns
$t_{d(E-P3Q)}$	Port P3 data output delay time			200		100		80	ns
$t_{d(E-P4Q)}$	Port P4 data output delay time			200		100		80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			200		100		80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			200		100		80	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time			200		100		80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			200		100		80	ns

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

Symbol	Parameter	Test conditions	Limits						Unit
			8MHz		16MHz		25MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(P0A-E)}$	Port P0 address output delay time	Fig. 2	100		30		12		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE="L")			110		70		45	ns
$t_{PXZ(E-P1Z)}$	Port P1 floating start delay time (BYTE="L")			5		5		5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time		100		30		12		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time		80		24		5		ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			110		70		45	ns
$t_{PXZ(E-P2Z)}$	Port P2 floating start delay time			5		5		5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time		100		30		12		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time		80		24		5		ns
$t_{d(\phi_1-HLDA)}$	$\overline{HLDA}$ output delay time			100		50		50	ns
$t_{d(ALE-E)}$	ALE output delay time		4		4		4		ns
$t_{W(ALE)}$	ALE pulse width		90		35		22		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time		100		30		20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time		100		30		20		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	30	0	20	0	18	ns
$t_{h(E-P0A)}$	Port P0 address hold time		50		25		18		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE="L")		9		9		9		ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE="L")		50		25		18		ns
$t_{PZX(E-P1Z)}$	Port P1 floating release delay time (BYTE="L")		50		25		18		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE="H")		50		25		18		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time		9		9		9		ns
$t_{h(E-P2Q)}$	Port P2 data hold time		50		25		18		ns
$t_{PZX(E-P2Z)}$	Port P2 floating release delay time		50		25		18		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		18		18		18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		18		18		18		ns
$t_{W(EL)}$	$\overline{E}$ pulse width		220		95		50		ns

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

Symbol	Parameter	Test conditions	Limits						Unit
			8MHz		16MHz		25MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(P0A-E)}$	Port P0 address output delay time	Fig. 2	100		30		12		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE="L")			110		70		45	ns
$t_{PXZ(E-P1Z)}$	Port P1 floating start delay time (BYTE="L")			5		5		5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time		100		30		12		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time		80		24		5		ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			110		70		45	ns
$t_{PXZ(E-P2Z)}$	Port P2 floating start delay time			5		5		5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time		100		30		12		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time		80		24		5		ns
$t_{d(\phi_1-HLDA)}$	$\overline{HLDA}$ output delay time			100		50		50	ns
$t_{d(ALE-E)}$	ALE output delay time		4		4		4		ns
$t_{W(ALE)}$	ALE pulse width		90		35		22		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time		100		30		20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time		100		30		20		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	30	0	20	0	18	ns
$t_{h(E-P0A)}$	Port P0 address hold time		50		25		18		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE="L")		9		9		9		ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE="L")		50		25		18		ns
$t_{PZX(E-P1Z)}$	Port P1 floating release delay time (BYTE="L")		50		25		18		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE="H")		50		25		18		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time		9		9		9		ns
$t_{h(E-P2Q)}$	Port P2 data hold time		50		25		18		ns
$t_{PZX(E-P2Z)}$	Port P2 floating release delay time		50		25		18		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		18		18		18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		18		18		18		ns
$t_{W(EL)}$	$\overline{E}$ pulse width		470		220		130		ns

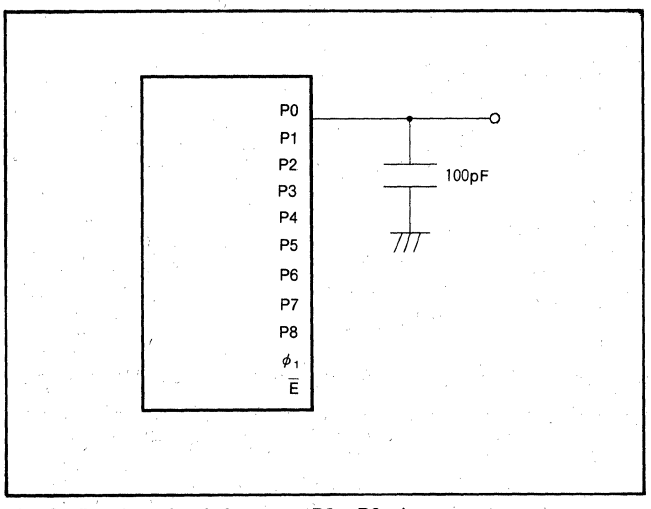
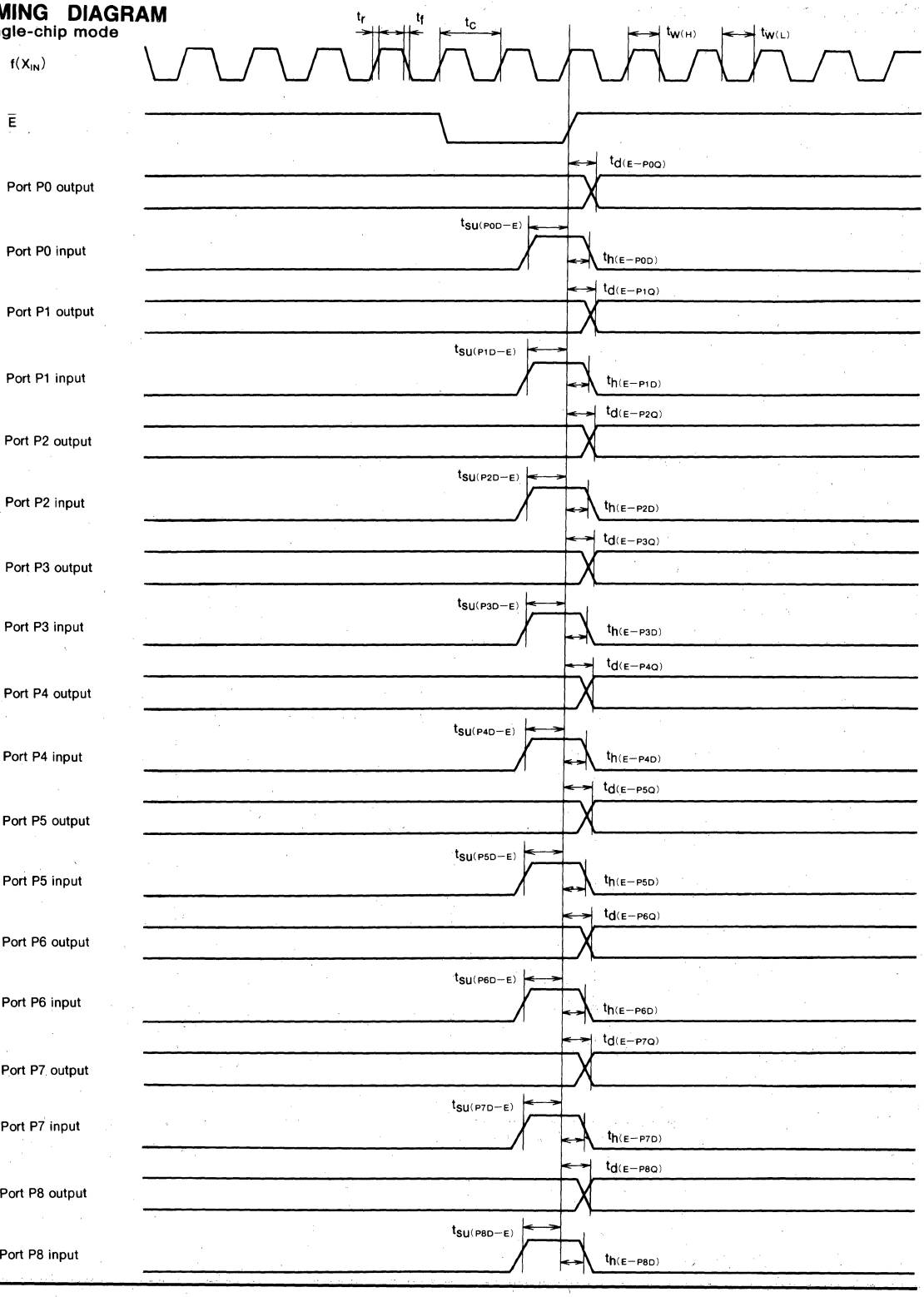
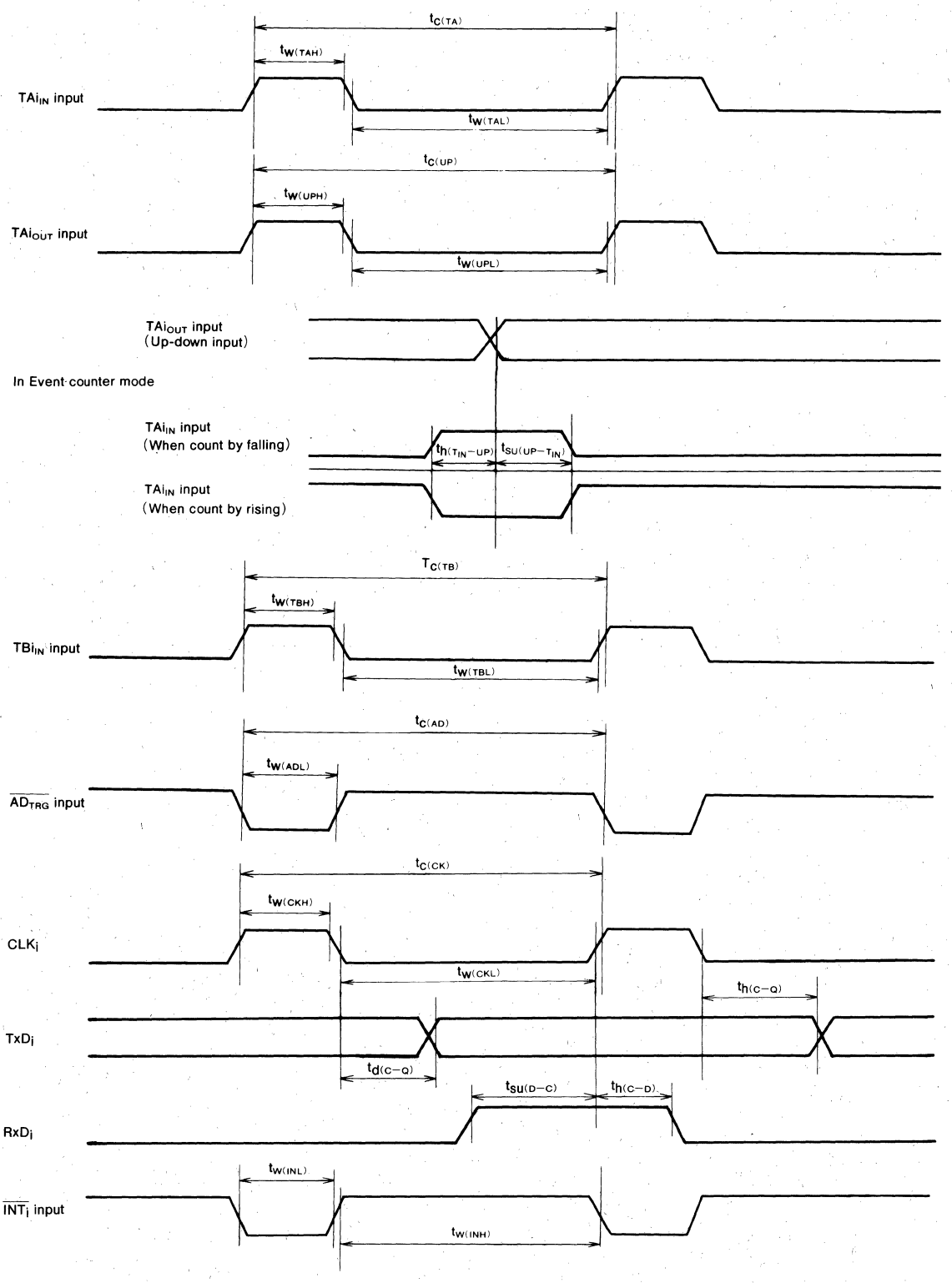


Fig. 2 Testing circuit for ports P0~P8, ϕ_1

TIMING DIAGRAM

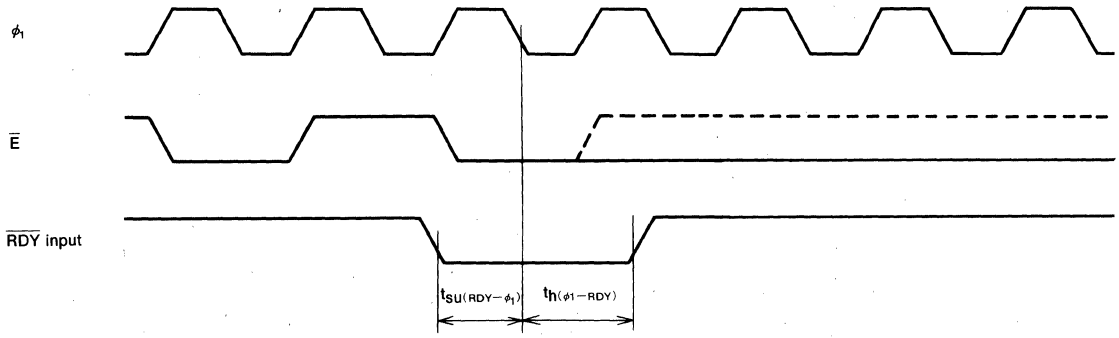
Single-chip mode



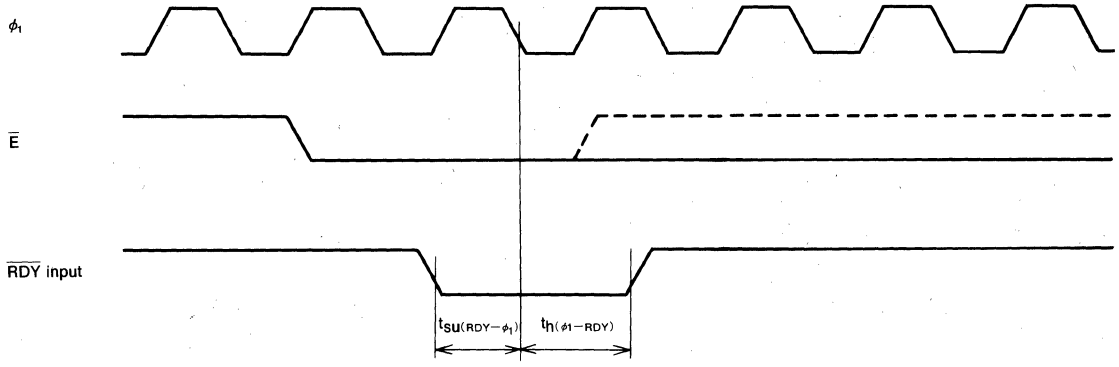

$t_{C(TA)}$
$t_{W(TAH)}$
TAi_{IN} input
$t_{W(TAL)}$
$t_{C(UP)}$
$t_{W(UPH)}$
TAi_{OUT} input
$t_{W(UPL)}$
TAi_{OUT} input
(Up-down input)
In Event counter mode
TAi_{IN} input
(When count by falling)
$t_{h(T_{IN}-UP)}$
$t_{SU(UP-T_{IN})}$
TAi_{IN} input
(When count by rising)
$T_{C(TB)}$
$t_{W(TBH)}$
TBi_{IN} input
$t_{W(TBL)}$
$t_{C(AD)}$
$t_{W(ADL)}$
$\overline{AD_{TRG}}$ input
$t_{C(CK)}$
$t_{W(CKH)}$
CLK_i
$t_{W(CKL)}$
$t_{h(C-Q)}$
TxD_i
$t_{d(C-Q)}$
$t_{su(D-C)}$
$t_{h(C-D)}$
RxD_i
$t_{W(INL)}$
$\overline{INT_i}$ input
$t_{W(INH)}$

Memory expansion mode and microprocessor mode

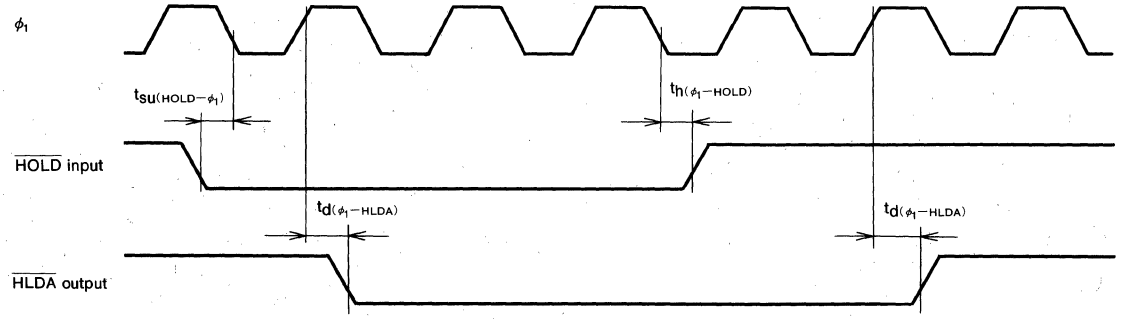
(When wait bit = "1")



(When wait bit = "0")



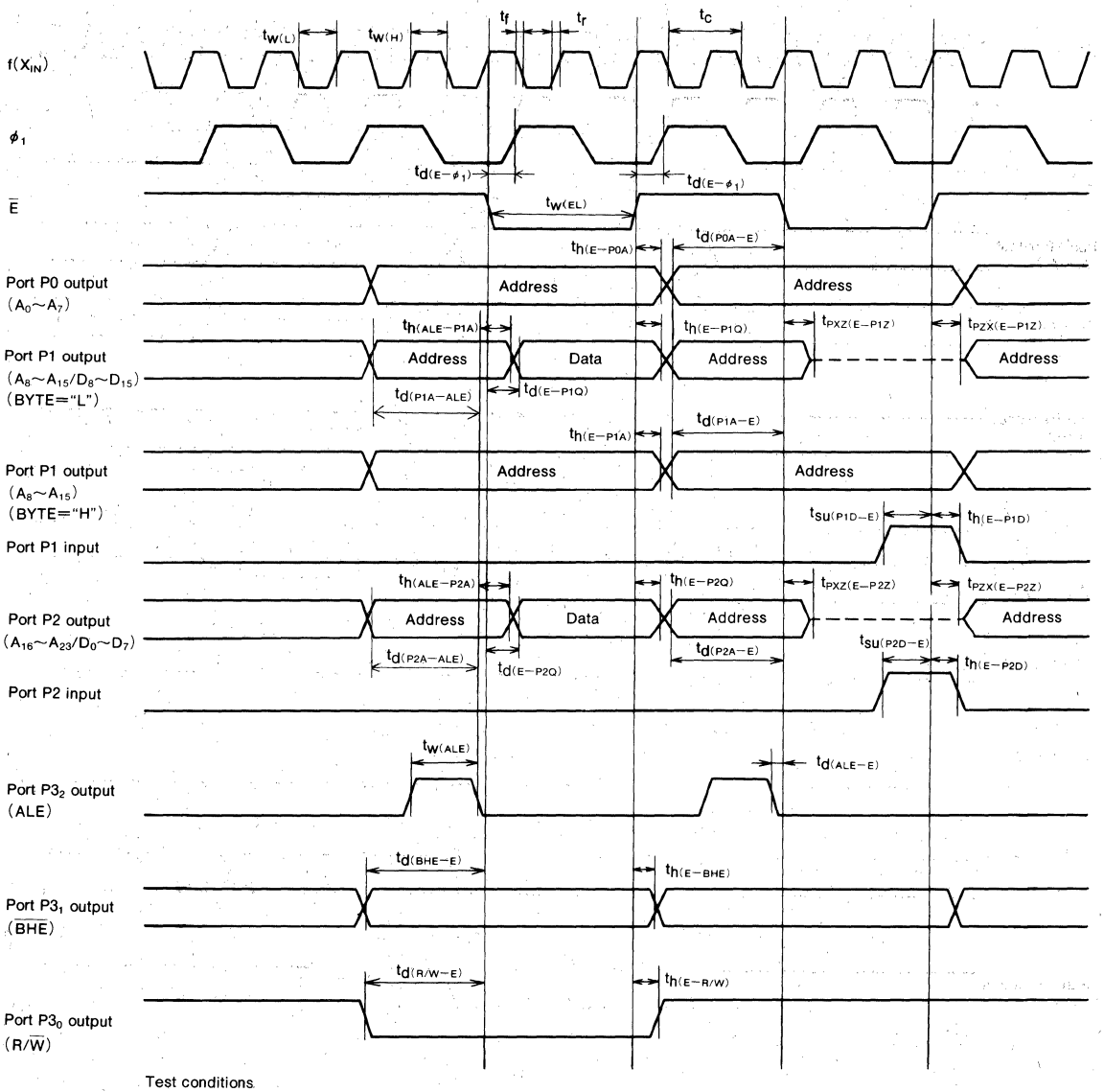
(When wait bit = "1" or "0" in common)



Test conditions

- V_{CC}= 5 V±10%
- Input timing voltage : V_{IL}=1.0V, V_{IH}=4.0V
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V

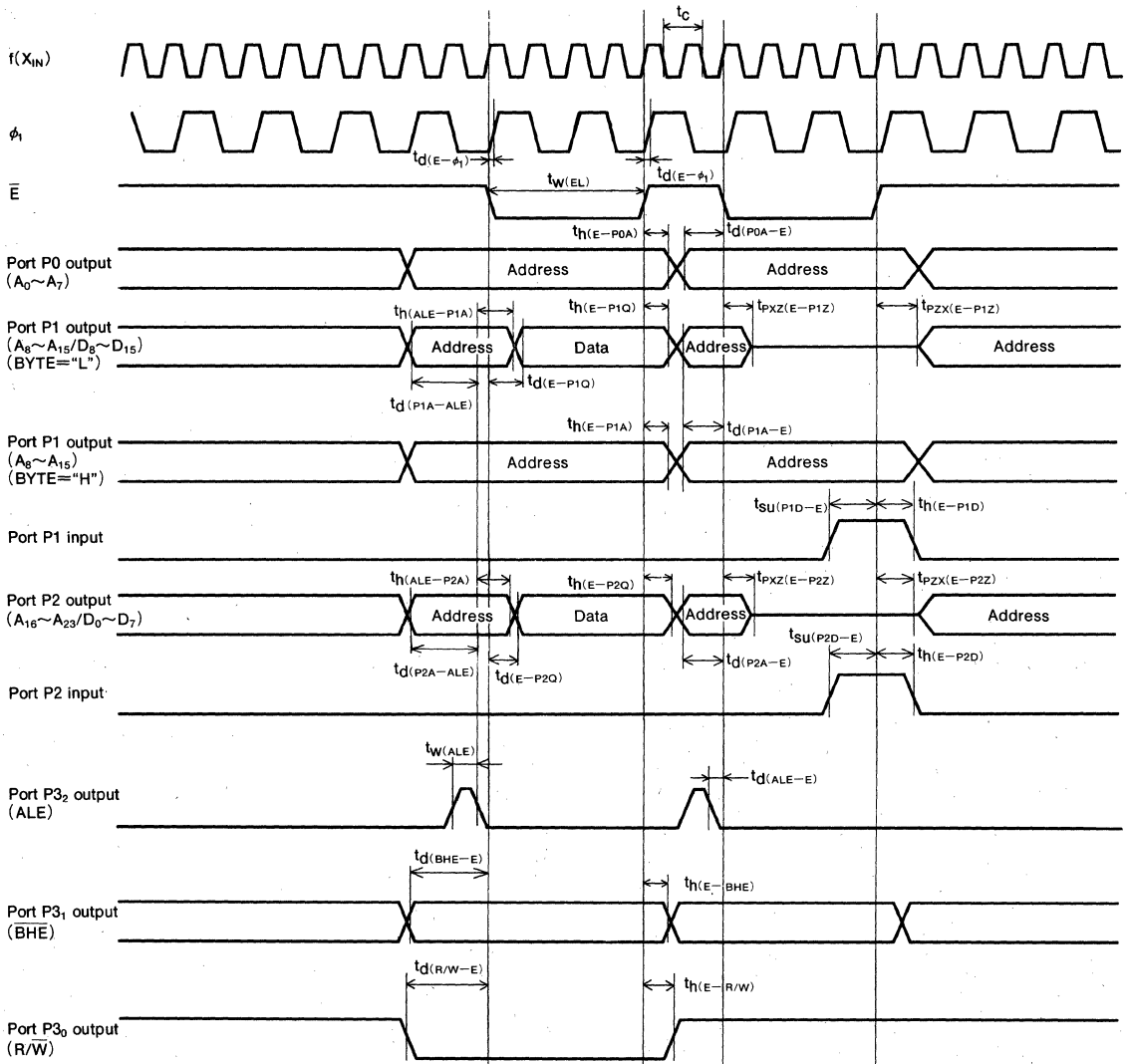
Memory expansion mode and microprocessor mode (When wait bit="1")



Test conditions

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

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



Test conditions

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

M37702E4-XXXFP,M37702E4AXXXFP,M37702E4BXXXFP
M37702E4FS,M37702E4AFS,M37702E4BFS

PROM VERSION of M37702M4-XXXFP,M37702M4AXXXFP,M37702M4BXXXFP

DESCRIPTION

The M37702E4-XXXFP, M37702E4AXXXFP and M37702E4BXXXFP are single-chip microcomputers designed with high-performance CMOS silicon gate technology. These are housed in a 80-pin plastic molded QFP. The features of these chips are similar to those of the M37702M4-XXXFP, M37702M4AXXXFP and M37702M4BXXXFP except that these chips have a 32K-byte PROM built in.

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

Since general purpose PROM writers can be used for the built-in PROM, these chips are suitable for small quantity production runs. The M37702E4FS (8MHz version), M37702E4AFS (16MHz version) and M37702E4BFS (25MHz version) with erasable ROM that are housed in a windowed ceramic LCC are also provided.

The differences between M37702E4-XXXFP, M37702E4AXXXFP and M37702E4BXXXFP are the external clock input frequency as shown below. Therefore, the following descriptions will be for the M37702E4-XXXFP unless otherwise noted.

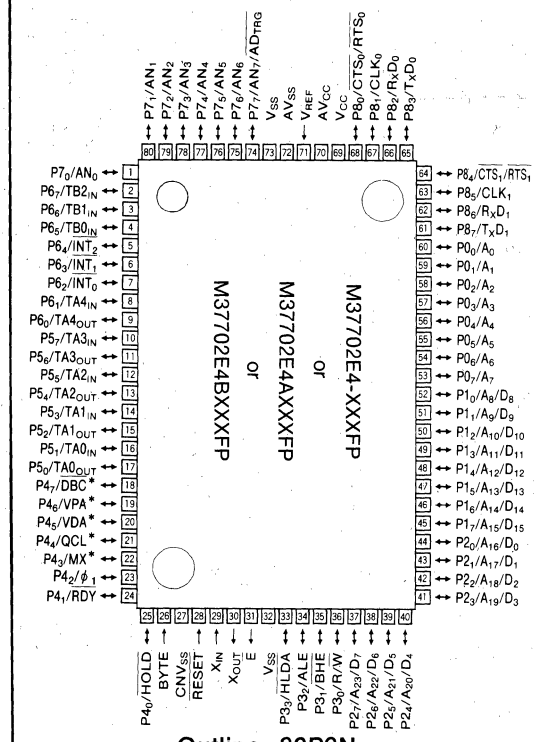
Type name	External clock input frequency
M37702E4-XXXFP	8 MHz
M37702E4AXXXFP	16MHz
M37702E4BXXXFP	25MHz

The M37702E4-XXXFP has the same functions as the M37702E2-XXXFP except for the memory size.

FEATURES

- Number of basic instructions 103
- Memory size PROM ..... 32K bytes
  RAM ..... 2048 bytes
- Instruction execution time
  M37702E4-XXXFP
  (The fastest instruction at 8 MHz frequency) 500ns
  M37702E4AXXXFP
  (The fastest instruction at 16 MHz frequency) 250ns
  M37702E4BXXXFP
  (The fastest instruction at 25 MHz frequency) 160ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 8 MHz frequency)
  ..... 30mW (Typ.)
- Interrupts ..... 19 types 7 levels
- Multiple function 16-bit timer ..... 5+3
- UART (may also be synchronous) 2
- 8-bit A-D converter ..... 8-channel inputs
- 12-bit watchdog timer
- Programmable input/output
  (ports P0, P1, P2, P3, P4, P5, P6, P7, P8) 68

PIN CONFIGURATION (TOP VIEW)



Outline 80P6N

*: Used in the evaluation chip mode only

APPLICATION

Control devices for office equipment such as copiers, printers, typewriters, facsimiles, word processors, and personal computers

Control devices for industrial equipment such as ME, NC, communication, and measuring instruments

THE FUNCTIONS AND CHARACTERISTICS

The M37702E4-XXXFP has the same functions and characteristics as the M37702E2-XXXFP except for the PROM and RAM size. Refer to the section on the M37702E2-XXXFP.

DATA REQUIRED FOR PROM ORDERING

Please send the following data for writing to PROM.

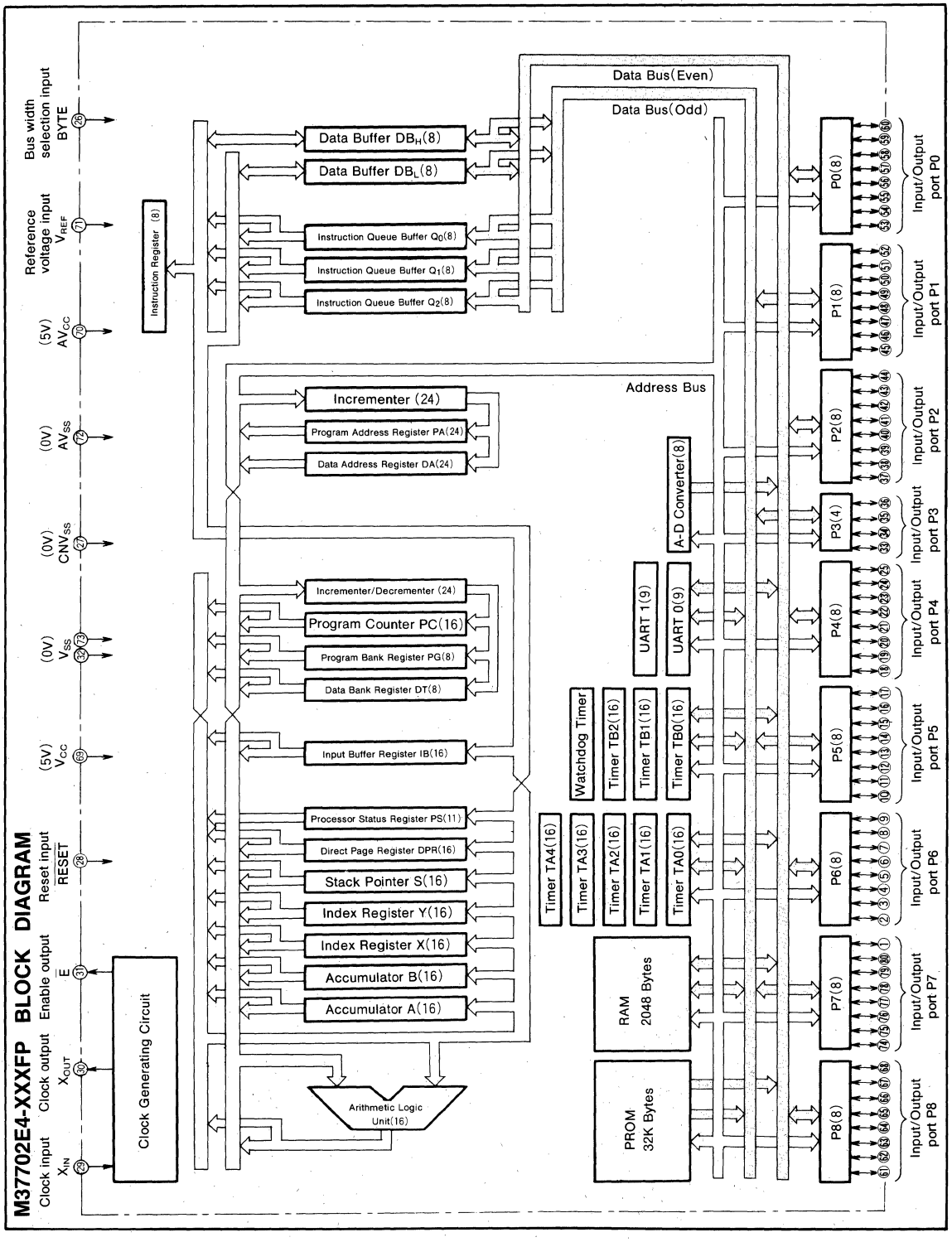
(1) M37702E4-XXXFP writing to PROM order confirmation form

(2) 80P6N mark specification form

(3) ROM data (EPROM 3sets)

MITSUBISHI ELECTRIC

PROM VERSION of M37702M4-XXXFP,M37702M4AXXXFP,M37702M4BXXXFP



M37703E2-XXXSP, M37703E2AXXXSP M37703E2BXXXSP

PROM VERSION of M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP

DESCRIPTION

The M37703E2-XXXSP, M37703E2AXXXSP and M37703E2BXXXSP are single-chip microcomputers designed with high-performance CMOS silicon gate technology. These are housed in a 64-pin shrink plastic molded DIP. The features of these chips are similar to those of the M37703M2-XXXSP, M37703M2AXXXSP and M37703M2BXXXSP except that this chip has a 16K-byte PROM built in.

These single-chip microcomputers have a large 16M bytes address space, three instruction queue buffers, and two data buffers for high-speed instruction execution. The CPU is a 16-bit parallel processor that can also be switched to perform 8-bit parallel processing. These microcomputers are suitable for office, business and industrial equipment controller that require high-speed processing of large data. Since general purpose PROM writers can be used for the built-in PROM, this chip is suitable for small quantity production runs.

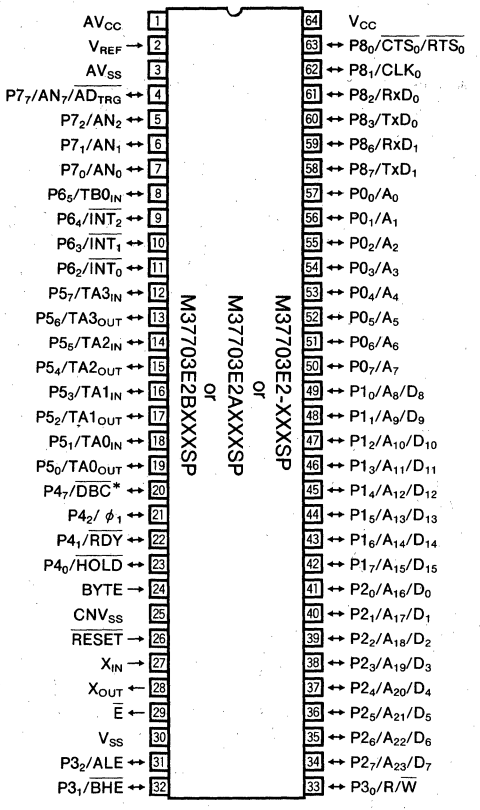
The differences between M37703E2-XXXSP, M37703E2AXXXSP and M37703E2BXXXSP are the external clock input frequency as shown below. Therefore, the following descriptions will be for the M37703E2-XXXSP unless otherwise noted.

Type name	External clock input frequency
M37703E2-XXXSP	8 MHz
M37703E2AXXXSP	16MHz
M37703E2BXXXSP	25MHz

FEATURES

- Number of basic instructions 103
- Memory size PROM (one time) ..... 16K bytes
  RAM ..... 512 bytes
- Instruction execution time
  M37703E2-XXXSP
  (The fastest instruction at 8 MHz frequency) 500ns
  M37703E2AXXXSP
  (The fastest instruction at 16 MHz frequency) 250ns
  M37703E2BXXXSP
  (The fastest instruction at 25 MHz frequency) 160ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 8 MHz frequency)
  ..... 30mW (Typ.)
- Interrupts ..... 19 types 7 levels
- Multiple function 16-bit timer ..... 5+3
- UART (may also be synchronous) 2
- 8-bit A-D converter ..... 4-channel inputs
- 12-bit watchdog timer
- Programmable input/output
  (ports P0, P1, P2, P3, P4, P5, P6, P7, P8) 53

PIN CONFIGURATION (TOP VIEW)



Outline 64P4B

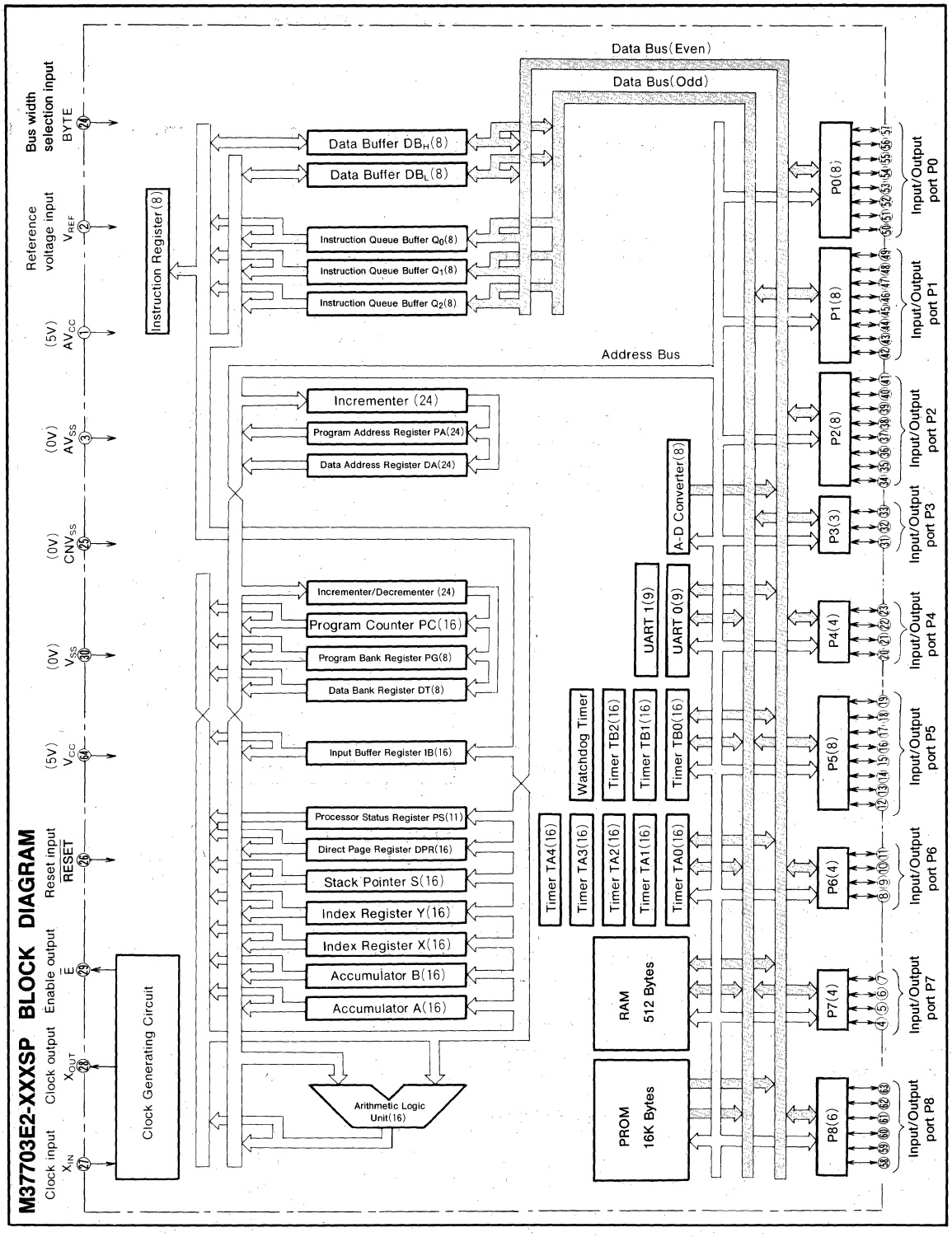
*: Used in the evaluation chip mode only

APPLICATION

Control devices for office equipment such as copiers, printers, typewriters, facsimiles, word processors, and personal computers

Control devices for industrial equipment such as ME, NC, communication, and measuring instruments

PROM VERSION of M37703M2-XXXSP,M37703M2AXXXSP,M37703M2BXXXSP



FUNCTIONS OF M37703E2-XXXSP

Parameter		Functions
Number of basic instructions		103
Instruction execution time	M37703E2-XXXSP	500ns (the fastest instructions, at 8MHz frequency)
	M37703E2AXXXSP	250ns (the fastest instructions, at 16MHz frequency)
	M37703E2BXXXSP	160ns (the fastest instructions, at 25MHz frequency)
Memory size	PROM	16K bytes
	RAM	512 bytes
Input/Output ports	P0, P1, P2, P5	8-bit× 4
	P8	6-bit× 1
	P4, P6, P7	4-bit× 3
	P3	3-bit× 1
Multi-function timers	TA0, TA1, TA2, TA3, TA4	16-bit× 5 (4 Input/Output functions)
	TB0, TB1, TB2	16-bit× 3 (1 Input function)
Serial I/O		UART× 2 (One can be set clock synchronous serial I/O.)
A-D converter		8-bit× 1 (4 channels)
Watchdog timer		12-bit× 1
Interrupts		3 external types, 16 internal types (Each interrupt can be set the priority levels to 0 ~ 7.)
Clock generating circuit		Built-in(externally connected to a ceramic resonator or quartz crystal resonator)
Supply voltage		5 V±10%
Power dissipation		30mW(at external 8 MHz frequency)
Input/Output characteristic	Input/Output voltage	5 V
	Output current	5 mA
Memory expansion		Maximum 16M bytes
Operating temperature range		−20~85℃
Device structure		CMOS high-performance silicon gate process
Package		64-pin shrink plastic molded DIP

PIN DESCRIPTION (NORMAL MODE)

Pin	Name	Input/Output	Functions
V_{CC}, V_{SS}	Power supply		Supply 5 V±10% to V_{CC} and 0 V to V_{SS}.
CNV_{SS}	CNV_{SS} input	Input	This pin controls the processor mode. Connect to V_{SS} for single-chip mode.
$\overline{RESET}$	Reset input	Input	To enter the reset state, this pin must be kept at a "L" condition should be maintained for the required time.
X_{IN}	Clock input	Input	These are I/O pins of internal clock generating circuit. Connect a ceramic or quartz crystal resonator between X_{IN} and X_{OUT}. When an external clock is used, the clock source should be connected to the X_{IN} pin and the X_{OUT} pin should be left open.
X_{OUT}	Clock output	Output	
$\overline{E}$	Enable output	Output	Data or instruction read and data write are performed when output from this pin is "L".
BYTE	Bus width selection input	Input	In memory expansion mode or microprocessor mode, this pin determines whether the external data bus is 8-bit width or 16-bit width. The width is 16 bits when "L" signal inputs and 8 bits when "H" signal inputs.
AV_{CC}, AV_{SS}	Analog supply input		Power supply for the A-D converter. Connect AV_{CC} to V_{CC} and AV_{SS} to V_{SS} externally.
V_{REF}	Reference voltage input	Input	This is reference voltage input pin for the A-D converter.
$P0_0$~$P0_7$	I/O port P0	I/O	In single-chip mode, port P0 becomes an 8-bit I/O port. An I/O directional register is available so that each pin can be programmed for input or output. These ports are in input mode when reset . Address(A_7~A_0)is output in memory expansion mode or microprocessor mode.
$P1_0$~$P1_7$	I/O port P1	I/O	In single-chip mode, these pins have the same functions as port P0. When the BYTE pin is set to "L" in memory expansion mode or microprocessor mode and external data bus is 16-bit width, high-order data (D_{15}~D_8)is input or output when $\overline{E}$ output is "L" and an address (A_{15}~A_8) is output when $\overline{E}$ output is "H". If the BYTE pin is "H" that is an external data bus is 8-bit width, only address(A_{15}~A_8)is output.
$P2_0$~$P2_7$	I/O port P2	I/O	In single-chip mode, these pins have the same functions as port P0. In memory expansion mode or microprocessor mode low-order data(D_7~D_0) is input or output when $\overline{E}$ output is "L" and an address(A_{23}~A_{16})is output when $\overline{E}$ output is "H".
$P3_0$~$P3_2$	I/O port P3	I/O	In single-chip mode , these pins have the same functions as port P0. In memory expansion mode or microprocessor mode, R/$\overline{W}$, $\overline{BHE}$, and ALE signals are output.
$P4_0$~$P4_2$, $P4_7$	I/O port P4	I/O	In single-chip mode, these pins have the same functions as port P0. In memory expansion mode or microprocessor mode, $P4_0$ and $P4_1$ become $\overline{HOLD}$ and $\overline{RDY}$ input pin respectively. Functions of other pins are the same as in single-chip mode. In single-chip mode or memory expansion mode, port $P4_2$ can be programed for ϕ_1 output pin divided the clock to X_{IN} pin by 2. In microprocessor mode, $P4_2$ always has the function as ϕ_1 output pin.
$P5_0$~$P5_7$	I/O port P5	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as I/O pins for timer A0, timer A1, timer A2, and timer A3.
$P6_2$~$P6_5$	I/O port P6	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as external interrupt input $\overline{INT_0}$, $\overline{INT_1}$, and $\overline{INT_2}$ pins, and input pins for timer B0.
$P7_0$~$P7_2$, $P7_7$	I/O port P7	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as analog input AN_0~AN_2 and AN_7 input pins. $P7_7$ also has an A-D conversion trigger input function.
$P8_0$~$P8_3$, $P8_6$, $P8_7$	I/O port P8	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as R_XD, T_XD, CLK, $\overline{CTS}$/$\overline{RTS}$ pins for UART 0, and as RxD, TxD pins for UART 1.

PIN DESCRIPTION (EPROM MODE)

Pin	Name	Input/Output	Functions
V_{CC}, V_{SS}	Power supply		Supply 5 V±10% to V_{CC} and 0 V to V_{SS}.
CNV_{SS}	V_{PP} input	Input	Connect to V_{PP} when programming or verifing.
BYTE	V_{PP} input	Input	Connect to V_{PP} when programming or verifing.
$\overline{RESET}$	Reset input	Input	Connect to V_{SS}.
X_{IN}	Clock input	Input	Connect a ceramic resonator between X_{IN} and X_{OUT}.
X_{OUT}	Clock output	Output	
$\overline{E}$	Enable output	Output	Keep open.
AV_{CC}, AV_{SS}	A-D power supply		Connect AV_{CC} to V_{CC} and AV_{SS} to V_{SS}.
V_{REF}	Reference voltage input	Input	Connect to V_{SS}.
$P0_0$~$P0_7$	Address input (A_0~A_7)	Input	Port P0 functions as the lower 8 bits address input (A_0~A_7).
$P1_0$~$P1_7$	Address input (A_8~A_{14})	Input	Port $P1_0$~$P1_6$ functions as the higher 7 bits address input (A_8~A_{14}). Connect $P1_7$ to V_{CC}.
$P2_0$~$P2_7$	Data I/O (D_0~D_7)	I/O	Port P2 functions as the 8 bits data bus (D_0~D_7).
$P3_0$~$P3_2$	Input port P3	Input	Connect to V_{SS}.
$P4_0$~$P4_2$, $P4_7$	Input port P4	Input	Connect to V_{SS}.
$P5_0$~$P5_7$	Control signal input	Input	$P5_1$ and $P5_2$ functions as $\overline{OE}$ and $\overline{CE}$ input pin. Connect $P5_0$, $P5_3$, $P5_4$ and $P5_5$ to V_{CC}. Connect $P5_6$ and $P5_7$ to V_{SS}.
$P6_2$~$P6_5$	Input port P6	Input	Connect to V_{SS}.
$P7_0$~$P7_2$, $P7_7$	Input port P7	Input	Connect to V_{SS}.
$P8_0$~$P8_3$, $P8_6$, $P8_7$	Input port P8	Input	Connect to V_{SS}.

PROM VERSION of M37703M2-XXXSP,M37703M2AXXXSP,M37703M2BXXXSP

EPROM MODE

The M37703E2-XXXSP features an EPROM mode in addition to its normal modes. When the $\overline{RESET}$ signal level is "L", the chip automatically enters the EPROM mode. Table 1 list the correspondence between pins and Fig. 1 gives the pin connections in the EPROM mode.

When in the EPROM mode, ports P0, P1, P2, $P5_1$, $P5_2$, CNV_{SS} and BYTE are used for the EPROM (equivalent to the M5M27C256K). When in this mode, the built-in PROM can be written to or read from using these pins in the same way as with the M5M27C256K.

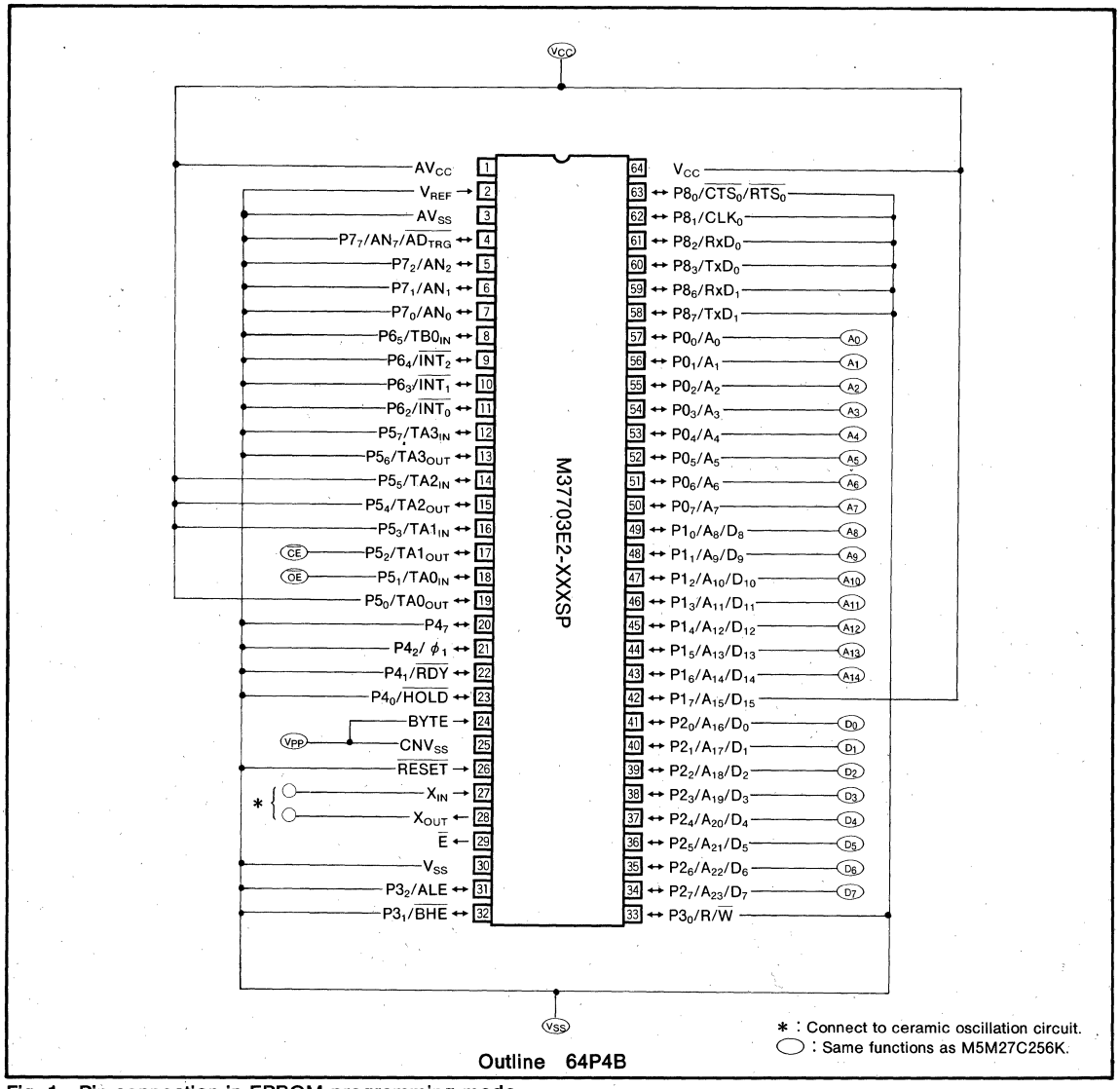
This chip does not have Device Identifier Mode, so that set the corresponding program algorithm. The program area should specify address 4000_{16}~$7FFF_{16}$ for the M37703E2-XXXSP.

Connect the clock which is either ceramic resonator or external clock to X_{IN} pin and X_{OUT} pin.

Caution :

Describing in this section, the built-in PROM can be written to or read in the same way as with the M5M27C256K (256K mode).

But in the future, for M37703E2BXXXSP, 1M mode may become standard.



Outline 64P4B

Fig. 1 Pin connection in EPROM programming mode

Table 1. Pin function in EPROM programming mode

	M37703E2-XXXSP	M5M27C256K
V_{CC}	V_{CC}	V_{CC}
V_{PP}	CNV_{SS}, BYTE	V_{PP}
V_{SS}	V_{SS}	V_{SS}
Address input	Ports P0, $P1_0$~$P1_6$	A_0~A_{14}
Data I/O	Port P2	D_0~D_7
$\overline{CE}$	$P5_2$	$\overline{CE}$
$\overline{OE}$	$P5_1$	$\overline{OE}$

FUNCTION IN EPROM MODE

Reading

To read the PROM, set the $\overline{CE}$ and $\overline{OE}$ pins to a "L" level. Input the address of the data (A_0~A_{14}) to be read and the data will be output to the I/O pins D_0~D_7. The data I/O pins will be floating when either the $\overline{CE}$ or $\overline{OE}$ pins are in the "H" state.

Writing

To write to the PROM, set the $\overline{OE}$ pin to a "H" level. The CPU will enter the program mode when V_{PP} is applied to the V_{PP} pin. The address to be written to is selected with pins A_0~A_{14}, and the data to be written is input to pins D_0 ~D_7. Set the $\overline{CE}$ pin to a "L" level to being writing.

FAST PROGRAMMING ALGORITHM

To program the M37703E2-XXXSP with fast programming algorithm, first set V_{CC}=6V, V_{PP}=12.5, and set the address to "0". Apply a 1ms write pulse, check that the data can be read, and if it cannot be read OK, repeat the procedure, applying a 1ms write pulse and checking that the data can be read until it can be read OK. Record the accumulated number of pulse applied (N) before the data can be read OK, and then write the data again, applying a further three times this number of pulses (3×N ms).

When this series of write operations is complete, increment the address, and continue to repeat the procedure above until the last address has been reached.

Finally, when all addresses have been written, read with V_{CC}=V_{PP}=5V (or V_{CC}=V_{PP}=5.25V).

Table 2. I/O signal in each mode

Pin / Mode	$\overline{CE}$	$\overline{OE}$	V_{PP}	V_{CC}	Data I/O
Read-out	V_{IL}	V_{IL}	5 V	5 V	Output
Output Disable	V_{IL}	V_{IH}	5 V	5 V	Floating
	V_{IH}	X	5 V	5 V	Floating
Programming	V_{IL}	V_{IH}	12.5V	6 V	Input
Programming Verify	V_{IH}	V_{IL}	12.5V	6 V	Output
Program Disable	V_{IH}	V_{IH}	12.5V	6 V	Floating

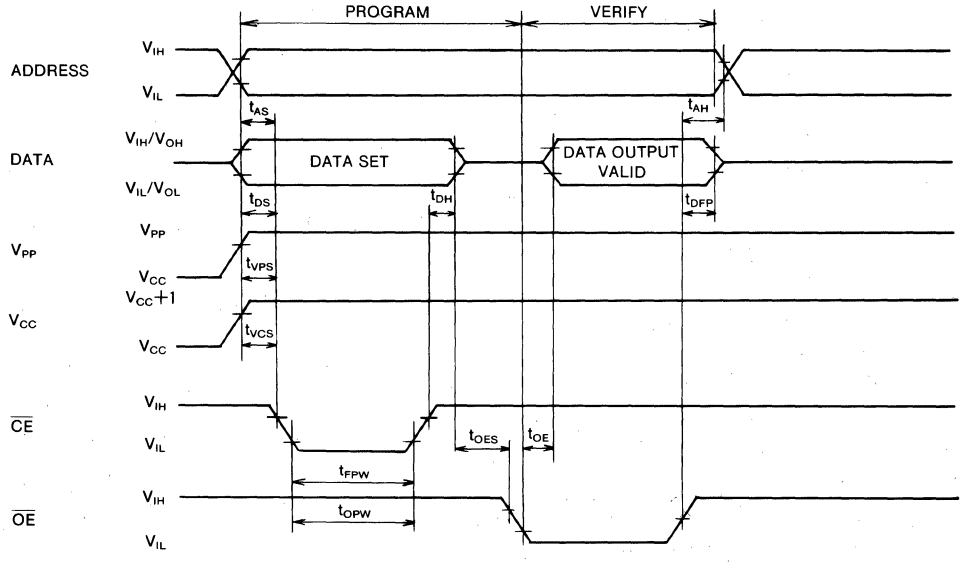
Note 1 : An X indicates either V_{IL} or V_{IH}.

Program operation

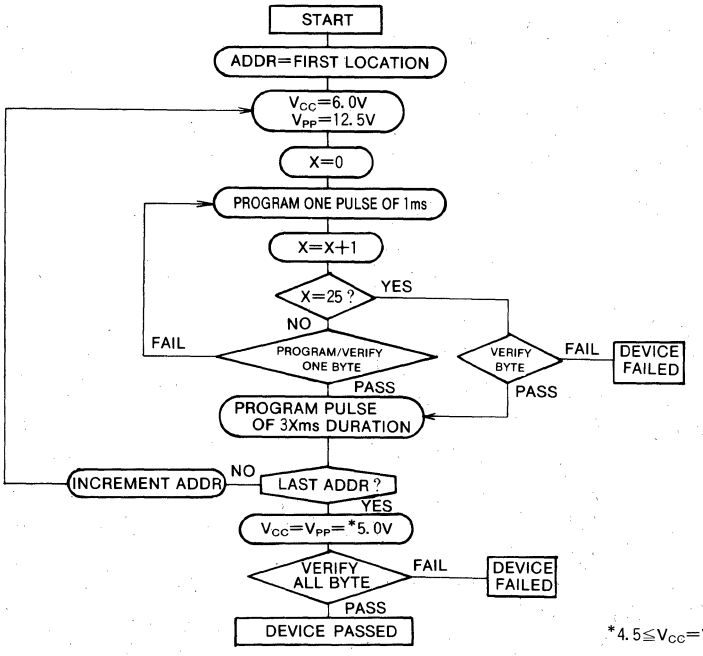
AC ELECTRICAL CHARACTERISTICS (T_a=25±5℃, V_{CC}=6V±0.25V, V_{PP}=12.5±0.3V, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
t_{AS}	Address setup time		2			μs
t_{OES}	$\overline{OE}$ setup time		2			μs
t_{DS}	Data setup time		2			μs
t_{AH}	Address hold time		0			μs
t_{DH}	Data hold time		2			μs
t_{DFP}	Output enable to output float delay		0		130	ns
t_{VCS}	V_{CC} setup time		2			μs
t_{VPS}	V_{PP} setup time		2			μs
t_{FPW}	$\overline{CE}$ initial program pulse width		0.95	1	1.05	ms
t_{OPW}	$\overline{CE}$ over program pulse width		2.85		78.75	ms
t_{OE}	Data valid from $\overline{OE}$				150	ns

AC waveforms



Fast programming algorithm flow chart

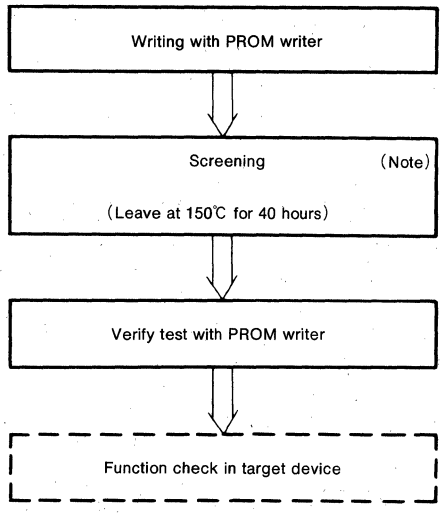


*4.5≦V_{CC}=V_{PP}≦5.5V

PROM VERSION of M37703M2-XXXSP,M37703M2AXXXSP,M37703M2BXXXSP

CAUTION: UNITS SHIPPED AS BLANKS

The programmable M37703E2SP, M37703E2ASP and M37703E2ESP that are shipped in blank are also provided. For the M37703E2SP, M37703E2ASP and M37703E2BSP, Mitsubishi Electric corp. does not perform PROM write test and screening following the assembly processes. To improve reliability after write, performing write and test according to the flow below before use is recommended.



Note : Never expose to 150℃ exceeding 100 hours.

BASIC FUNCTION BLOCKS

Since these processors operate in exactly the same way as the M37703M2-XXXSP, refer to the section on the M37703M2-XXXSP.

ADDRESSING MODES

The M37703E2-XXXSP has 28 powerful addressing modes. Refer to the MELPS 7700 addressing mode description for the details of each addressing mode.

MACHINE INSTRUCTION LIST

The M37703E2-XXXSP has 103 machine instructions. Refer to the MELPS 7700 machine instruction list for details.

DATA REQUIRED FOR PROM ORDERING

Please send the following data for writing to PROM .

(1) M37703E2-XXXSP writing to PROM order confirmation form
(2) 64P4B mark specification form for one time PROM
(3) ROM data (EPROM 3sets)

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Supply voltage		−0.3∼7	V
AV_{CC}	Analog supply voltage		−0.3∼7	V
V_I	Input voltage $\overline{RESET}$, CNV_{SS}, BYTE		−0.3∼12(Note1)	V
V_I	Input voltage $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_2$, $P4_0$∼$P4_2$, $P4_7$, $P5_0$∼$P5_7$, $P6_2$∼$P6_5$, $P7_0$∼$P7_2$, $P7_7$, $P8_0$∼$P8_3$, $P8_6$, $P8_7$, V_{REF}, X_{IN}		−0.3∼V_{CC}+0.3	V
V_O	Output voltage $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_2$, $P4_0$∼$P4_2$, $P4_7$, $P5_0$∼$P5_7$, $P6_2$∼$P6_5$, $P7_0$∼$P7_2$, $P7_7$, $P8_0$∼$P8_3$, $P8_6$, $P8_7$, X_{OUT}, $\overline{E}$		−0.3∼V_{CC}+0.3	V
P_d	Power dissipation	T_a=25℃	1000	mW
T_{opr}	Operating temperature		−20∼85	℃
T_{stg}	Storage temperature		−40∼150	℃

Note 1. Input voltage for CNV_{SS} and BYTE pins is 13V in writing to PROM.

RECOMMENDED OPERATING CONDITIONS (V_{CC}=5V±5%, T_a=−20∼85℃, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V_{CC}	Supply voltage		4.5	5.0	5.5	V
AV_{CC}	Analog supply voltage			V_{CC}		V
V_{SS}	Supply voltage			0		V
AV_{SS}	Analog supply voltage			0		V
V_{IH}	High-level input voltage $P0_0$∼$P0_7$, $P3_0$∼$P3_2$, $P4_0$∼$P4_2$, $P4_7$, $P5_0$∼$P5_7$, $P6_2$∼$P6_5$, $P7_0$∼$P7_2$, $P7_7$, $P8_0$∼$P8_3$, $P8_6$, $P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE		0.8V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage $P1_0$∼$P1_7$, $P2_0$∼$P2_7$ (in single-chip mode)		0.8V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage $P1_0$∼$P1_7$, $P2_0$∼$P2_7$ (in memory expansion mode and microprocessor mode)		0.5V_{CC}		V_{CC}	V
V_{IL}	Low-level input voltage $P0_0$∼$P0_7$, $P3_0$∼$P3_2$, $P4_0$∼$P4_2$, $P4_7$, $P5_0$∼$P5_7$, $P6_2$∼$P6_5$, $P7_0$∼$P7_2$, $P7_7$, $P8_0$∼$P8_3$, $P8_6$, $P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE		0		0.2V_{CC}	V
V_{IL}	Low-level input voltage $P1_0$∼$P1_7$, $P2_0$∼$P2_7$ (in single-chip mode)		0		0.2V_{CC}	V
V_{IL}	Low-level input voltage $P1_0$∼$P1_7$, $P2_0$∼$P2_7$ (in memory expansion mode and microprocessor mode)		0		0.16V_{CC}	V
$I_{OH(peak)}$	High-level peak output current $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_2$, $P4_0$∼$P4_2$, $P4_7$, $P5_0$∼$P5_7$, $P6_2$∼$P6_5$, $P7_0$∼$P7_2$, $P7_7$, $P8_0$∼$P8_3$, $P8_6$, $P8_7$				−10	mA
$I_{OH(avg)}$	High-level average output current $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_2$, $P4_0$∼$P4_2$, $P4_7$, $P5_0$∼$P5_7$, $P6_2$∼$P6_5$, $P7_0$∼$P7_2$, $P7_7$, $P8_0$∼$P8_3$, $P8_6$, $P8_7$				−5	mA
$I_{OL(peak)}$	Low-level peak output current $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_2$, $P4_0$∼$P4_2$, $P4_7$, $P5_0$∼$P5_7$, $P6_2$∼$P6_5$, $P7_0$∼$P7_2$, $P7_7$, $P8_0$∼$P8_3$, $P8_6$, $P8_7$				10	mA
$I_{OL(avg)}$	Low-level average output current $P0_0$∼$P0_7$, $P1_0$∼$P1_7$, $P2_0$∼$P2_7$, $P3_0$∼$P3_2$, $P4_0$∼$P4_2$, $P4_7$, $P5_0$∼$P5_7$, $P6_2$∼$P6_5$, $P7_0$∼$P7_2$, $P7_7$, $P8_0$∼$P8_3$, $P8_6$, $P8_7$				5	mA
$f(X_{IN})$	External clock frequency input	M37703E2-XXXSP			8	MHz
		M37703E2AXXXSP			16	
		M37703E2BXXXSP			25	

Note 2. Average output current is the average value of a 100ms interval.

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

M37703E2-XXXSP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=8MHz, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits Min.	Limits Typ.	Limits Max.	Unit
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$	I_{OH}=−10mA		3			V
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$	I_{OH}=−400μA		4.7			V
V_{OH}	High-level output voltage $P3_2$	I_{OH}=−10mA		3.1			V
		I_{OH}=−400μA		4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA		3.4			V
		I_{OH}=−400μA		4.8			
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$	I_{OL}=10mA				2	V
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$	I_{OL}=2mA				0.45	V
V_{OL}	Low-level output voltage $P3_2$	I_{OL}=10mA				1.9	V
		I_{OL}=2mA				0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA				1.6	V
		I_{OL}=2mA				0.4	
V_{T+}−V_{T-}	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA3_{IN}$, $TB0_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, $\overline{AD_{TRG}}$, $\overline{CTS_0}$, CLK_0			0.4		1	V
V_{T+}−V_{T-}	Hysteresis $\overline{RESET}$			0.2		0.5	V
V_{T+}−V_{T-}	Hysteresis X_{IN}			0.1		0.3	V
I_{IH}	High-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_2$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=5V				5	μA
I_{IL}	Low-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_2$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=0V				−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.		2			V
I_{CC}	Power supply current	In single-chip mode output only pin is open and other pins are V_{SS} during reset.	f(X_{IN})=8MHz, square waveform		6	12	mA
			T_a=25℃ when clock is stopped.			1	μA
			T_a=85℃ when clock is stopped.			20	

A-D CONVERTER CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=8MHz, unless otherwise noted)

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

M37703E2AXXXSP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=16MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$	I_{OH}=−10mA	3			V
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$	I_{OH}=−400μA	4.7			V
V_{OH}	High-level output voltage $P3_2$	I_{OH}=−10mA	3.1			V
		I_{OH}=−400μA	4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA	3.4			V
		I_{OH}=−400μA	4.8			
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$	I_{OL}=10mA			2	V
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$	I_{OL}=2mA			0.45	V
V_{OL}	Low-level output voltage $P3_2$	I_{OL}=10mA			1.9	V
		I_{OL}=2mA			0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA			1.6	V
		I_{OL}=2mA			0.4	
V_{T+}−V_{T-}	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA3_{IN}$, $TB0_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, $\overline{AD_{TRG}}$, $\overline{CTS_0}$, CLK_0		0.4		1	V
V_{T+}−V_{T-}	Hysteresis $\overline{RESET}$		0.2		0.5	V
V_{T+}−V_{T-}	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_2$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=5V			5	μA
I_{IL}	Low-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_2$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=0V			−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	In single-chip mode output only pin is open and other pins are V_{SS} during reset. f(X_{IN})=16MHz, square waveform		12	24	mA
		T_a=25℃ when clock is stopped.			1	μA
		T_a=85℃ when clock is stopped.			20	

A-D CONVERTER CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=16MHz, unless otherwise noted)

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

M37703E2BXXXSP

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=25MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits Min.	Limits Typ.	Limits Max.	Unit
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$	I_{OH}=−10mA	3			V
V_{OH}	High-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$	I_{OH}=−400μA	4.7			V
V_{OH}	High-level output voltage $P3_2$	I_{OH}=−10mA	3.1			V
		I_{OH}=−400μA	4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA	3.4			V
		I_{OH}=−400μA	4.8			
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$	I_{OL}=10mA			2	V
V_{OL}	Low-level output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$, $P3_1$	I_{OL}=2mA			0.45	V
V_{OL}	Low-level output voltage $P3_2$	I_{OL}=10mA			1.9	V
		I_{OL}=2mA			0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA			1.6	V
		I_{OL}=2mA			0.4	
$V_{T+}-V_{T-}$	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA3_{IN}$, $TB0_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, $\overline{AD_{TRG}}$, $\overline{CTS_0}$, CLK_0		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis $\overline{RESET}$		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_2$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=5V			5	μA
I_{IL}	Low-level input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_2$, $P4_0$~$P4_2$, $P4_7$, $P5_0$~$P5_7$, $P6_2$~$P6_5$, $P7_0$~$P7_2$, $P7_7$, $P8_0$~$P8_3$, $P8_6$, $P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	V_I=0V			−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	In single-chip mode output only pin is open and other pins are V_{SS} during reset. f(X_{IN})=25MHz, square waveform		19	38	mA
		T_a=25℃ when clock is stopped.			1	μA
		T_a=85℃ when clock is stopped.			20	

A-D CONVERTER CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=25MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits Min.	Limits Typ.	Limits Max.	Unit
—	Resolution	V_{REF}=V_{CC}			8	Bits
—	Absolute accuracy	V_{REF}=V_{CC}			±3	LSB
R_{LADDER}	Ladder resistance	V_{REF}=V_{CC}	2		10	kΩ
t_{CONV}	Conversion time		9.12			μs
V_{REF}	Reference voltage		2		V_{CC}	V
V_{IA}	Analog input voltage		0		V_{REF}	V

TIMING REQUIREMENTS (V_{CC}=5V±10%, V_{SS}=0V, T_a=25℃, unless otherwise noted)

External clock input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
t_C	External clock input cycle time	125		62		40		ns
$t_{W(H)}$	External clock input high-level pulse width	50		25		15		ns
$t_{W(L)}$	External clock input low-level pulse width	50		25		15		ns
t_r	External clock rise time		20		10		8	ns
t_f	External clock fall time		20		10		8	ns

Single-chip mode

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{SU(P0D-E)}$	Port P0 input setup time	200		100		60		ns
$t_{SU(P1D-E)}$	Port P1 input setup time	200		100		60		ns
$t_{SU(P2D-E)}$	Port P2 input setup time	200		100		60		ns
$t_{SU(P3D-E)}$	Port P3 input setup time	200		100		60		ns
$t_{SU(P4D-E)}$	Port P4 input setup time	200		100		60		ns
$t_{SU(P5D-E)}$	Port P5 input setup time	200		100		60		ns
$t_{SU(P6D-E)}$	Port P6 input setup time	200		100		60		ns
$t_{SU(P7D-E)}$	Port P7 input setup time	200		100		60		ns
$t_{SU(P8D-E)}$	Port P8 input setup time	200		100		60		ns
$t_{h(E-P0D)}$	Port P0 input hold time	0		0		0		ns
$t_{h(E-P1D)}$	Port P1 input hold time	0		0		0		ns
$t_{h(E-P2D)}$	Port P2 input hold time	0		0		0		ns
$t_{h(E-P3D)}$	Port P3 input hold time	0		0		0		ns
$t_{h(E-P4D)}$	Port P4 input hold time	0		0		0		ns
$t_{h(E-P5D)}$	Port P5 input hold time	0		0		0		ns
$t_{h(E-P6D)}$	Port P6 input hold time	0		0		0		ns
$t_{h(E-P7D)}$	Port P7 input hold time	0		0		0		ns
$t_{h(E-P8D)}$	Port P8 input hold time	0		0		0		ns

Memory expansion mode and microprocessor mode

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{SU(P1D-E)}$	Port P1 input setup time	60		45		30		ns
$t_{SU(P2D-E)}$	Port P2 input setup time	60		45		30		ns
$t_{SU(RDY-\phi_1)}$	$\overline{RDY}$ input setup time	70		60		55		ns
$t_{SU(HOLD-\phi_1)}$	$\overline{HOLD}$ input setup time	70		60		55		ns
$t_{h(E-P1D)}$	Port P1 input hold time	0		0		0		ns
$t_{h(E-P2D)}$	Port P2 input hold time	0		0		0		ns
$t_{h(\phi_1-RDY)}$	$\overline{RDY}$ input hold time	0		0		0		ns
$t_{h(\phi_1-HOLD)}$	$\overline{HOLD}$ input hold time	0		0		0		ns

Timer A input (Count input in event counter mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	250		125		80		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	125		62		40		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	125		62		40		ns

Timer A input (Gating input in timer mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	1000		500		320		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	500		250		160		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	500		250		160		ns

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

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TA)}$	TAi_{IN} input cycle time	500		250		160		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	250		125		80		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	250		125		80		ns

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

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	250		125		80		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	250		125		80		ns

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

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(UP)}$	TAi_{OUT} input cycle time	5000		2500		2000		ns
$t_{W(UPH)}$	TAi_{OUT} input high-level pulse width	2500		1250		1000		ns
$t_{W(UPL)}$	TAi_{OUT} input low-level pulse width	2500		1250		1000		ns
$t_{su(UP-T_{IN})}$	TAi_{OUT} input setup time	1000		500		400		ns
$t_{h(T_{IN}-UP)}$	TAi_{OUT} input hold time	1000		500		400		ns

Timer B input (Count input in event counter mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	$TB0_{IN}$ input cycle time (one edge count)	250		125		80		ns
$t_{W(TBH)}$	$TB0_{IN}$ input high-level pulse width (one edge count)	125		62		40		ns
$t_{W(TBL)}$	$TB0_{IN}$ input low-level pulse width (one edge count)	125		62		40		ns
$t_{C(TB)}$	$TB0_{IN}$ input cycle time (both edges count)	500		250		160		ns
$t_{W(TBH)}$	$TB0_{IN}$ input high-level pulse width (both edges count)	250		125		80		ns
$t_{W(TBL)}$	$TB0_{IN}$ input low-level pulse width (both edges count)	250		125		80		ns

Timer B input (Pulse period measurement mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	$TB0_{IN}$ input cycle time	1000		500		320		ns
$t_{W(TBH)}$	$TB0_{IN}$ input high-level pulse width	500		250		160		ns
$t_{W(TBL)}$	$TB0_{IN}$ input low-level pulse width	500		250		160		ns

Timer B input (Pulse width measurement mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	$TB0_{IN}$ input cycle time	1000		500		320		ns
$t_{W(TBH)}$	$TB0_{IN}$ input high-level pulse width	500		250		160		ns
$t_{W(TBL)}$	$TB0_{IN}$ input low-level pulse width	500		250		160		ns

A-D trigger input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(AD)}$	$\overline{AD_{TRG}}$ input cycle time (minimum allowable trigger)	2000		1000		1000		ns
$t_{W(ADL)}$	$\overline{AD_{TRG}}$ input low-level pulse width	250		125		125		ns

Serial I/O

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(CK)}$	CLK_0 input cycle time	500		250		200		ns
$t_{W(CKH)}$	CLK_0 input high-level pulse width	250		125		100		ns
$t_{W(CKL)}$	CLK_0 input low-level pulse width	250		125		100		ns
$t_{d(C-Q)}$	TxD_0 output delay time		150		90		80	ns
$t_{h(C-Q)}$	TxD_0 hold time	30		30		30		ns
$t_{su(D-C)}$	RxD_0 input setup time	60		30		20		ns
$t_{h(C-D)}$	RxD_0 input hold time	90		90		90		ns

External interrupt $\overline{INT_i}$ input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{W(INH)}$	$\overline{INT_i}$ input high-level pulse width	250		250		250		ns
$t_{W(INL)}$	$\overline{INT_i}$ input low-level pulse width	250		250		250		ns

SWITCHING CHARACTERISTICS (V_{CC}=5V±10%, V_{SS}=0V, T_a=25℃, unless otherwise noted)

Single-chip mode

Symbol	Parameter	Test conditions	Limits						Unit
			8MHz		16MHz		25MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(E-P0Q)}$	Port P0 data output delay time	Fig. 1		200		100		80	ns
$t_{d(E-P1Q)}$	Port P1 data output delay time			200		100		80	ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			200		100		80	ns
$t_{d(E-P3Q)}$	Port P3 data output delay time			200		100		80	ns
$t_{d(E-P4Q)}$	Port P4 data output delay time			200		100		80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			200		100		80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			200		100		80	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time			200		100		80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			200		100		80	ns

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

Symbol	Parameter	Test conditions	Limits						Unit
			8MHz		16MHz		25MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(P0A-E)}$	Port P0 address output delay time	Fig. 1	100		30		12		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE="L")			110		70		45	ns
$t_{PXZ(E-P1Z)}$	Port P1 floating start delay time (BYTE="L")			5		5		5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time		100		30		12		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time		80		24		5		ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			110		70		45	ns
$t_{PXZ(E-P2Z)}$	Port P2 floating start delay time			5		5		5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time		100		30		12		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time		80		24		5		ns
$t_{d(ALE-E)}$	ALE output delay time		4		4		4		ns
$t_{W(ALE)}$	ALE pulse width		90		35		22		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time		100		30		20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time		100		30		20		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	30	0	20	0	18	ns
$t_{h(E-P0A)}$	Port P0 address hold time		50		25		18		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE="L")		9		9		9		ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE="L")		50		25		18		ns
$t_{PZX(E-P1Z)}$	Port P1 floating release delay time (BYTE="L")		50		25		18		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE="H")		50		25		18		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time		9		9		9		ns
$t_{h(E-P2Q)}$	Port P2 data hold time		50		25		18		ns
$t_{PZX(E-P2Z)}$	Port P2 floating release delay time		50		25		18		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		18		18		18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		18		18		18		ns
$t_{W(EL)}$	$\overline{E}$ pulse width		220		95		50		ns

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

Symbol	Parameter	Test conditions	Limits						Unit
			8MHz		16MHz		25MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(P0A-E)}$	Port P0 address output delay time	Fig. 1	100		30		12		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE="L")			110		70		45	ns
$t_{PXZ(E-P1Z)}$	Port P1 floating start delay time (BYTE="L")			5		5		5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time		100		30		12		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time		80		24		5		ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			110		70		45	ns
$t_{PXZ(E-P2Z)}$	Port P2 floating start delay time			5		5		5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time		100		30		12		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time		80		24		5		ns
$t_{d(ALE-E)}$	ALE output delay time		4		4		4		ns
$t_{W(ALE)}$	ALE pulse width		90		35		22		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time		100		30		20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time		100		30		20		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	30	0	20	0	18	ns
$t_{h(E-P0A)}$	Port P0 address hold time		50		25		18		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE="L")		9		9		9		ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE="L")		50		25		18		ns
$t_{PZX(E-P1Z)}$	Port P1 floating release delay time (BYTE="L")		50		25		18		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE="H")		50		25		18		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time		9		9		9		ns
$t_{h(E-P2Q)}$	Port P2 data hold time		50		25		18		ns
$t_{PZX(E-P2Z)}$	Port P2 floating release delay time		50		25		18		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		18		18		18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		18		18		18		ns
$t_{W(EL)}$	$\overline{E}$ pulse width		470		220		130		ns

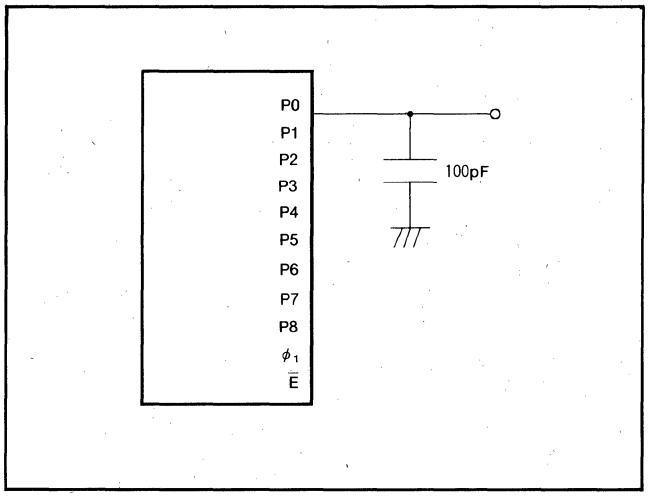
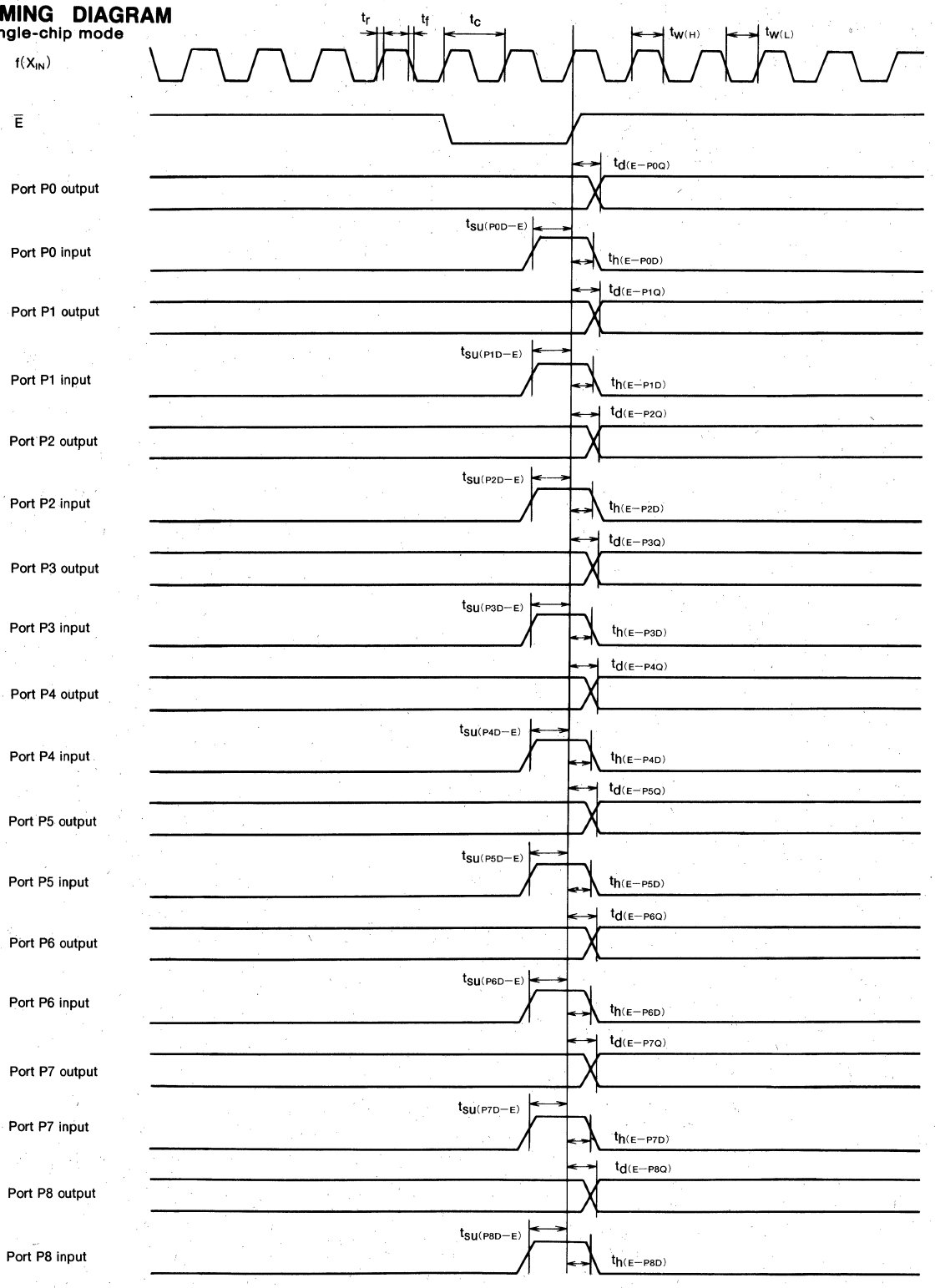
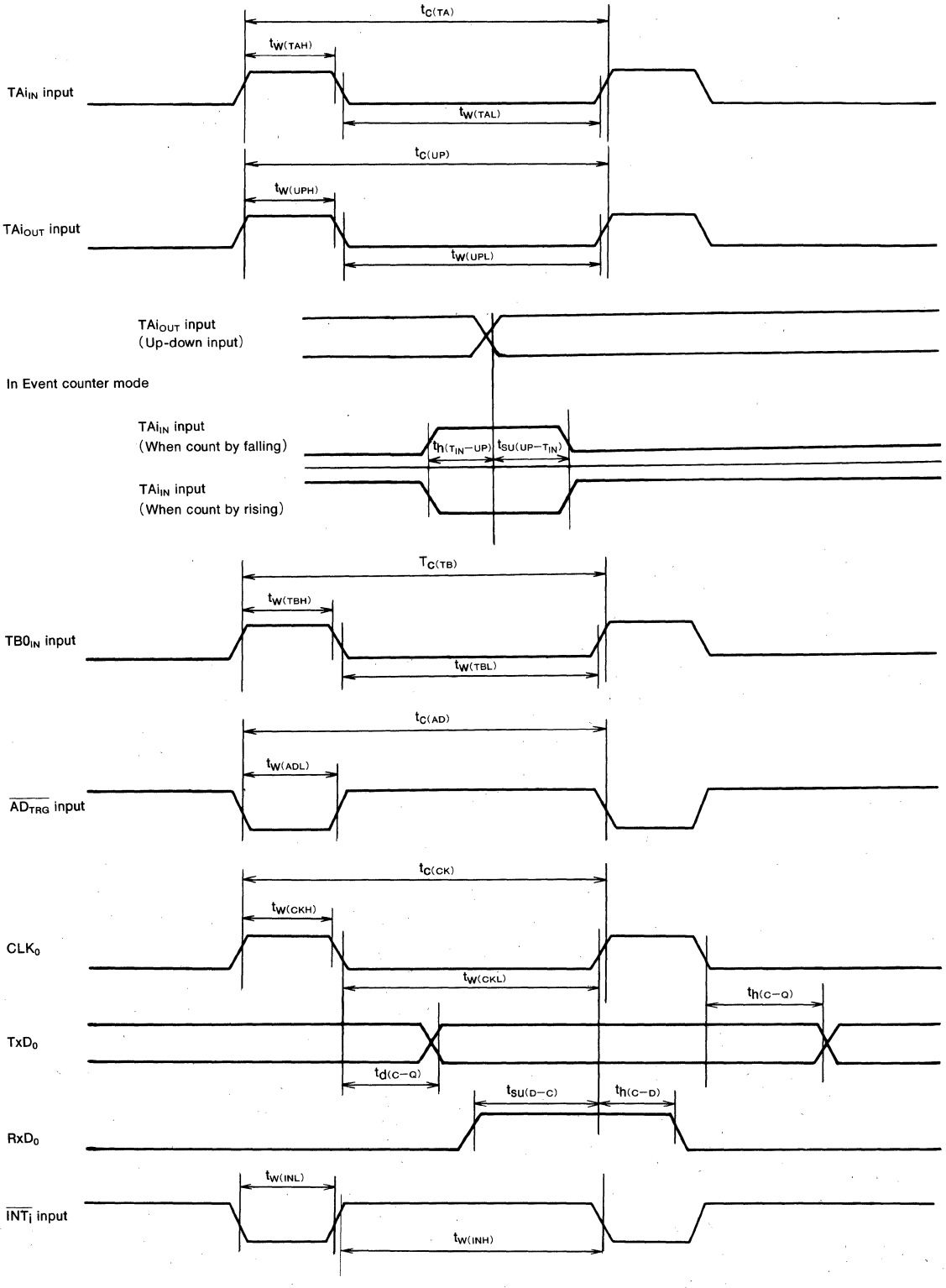


Fig. 1 Testing circuit for ports P0~P8, ϕ_1

TIMING DIAGRAM

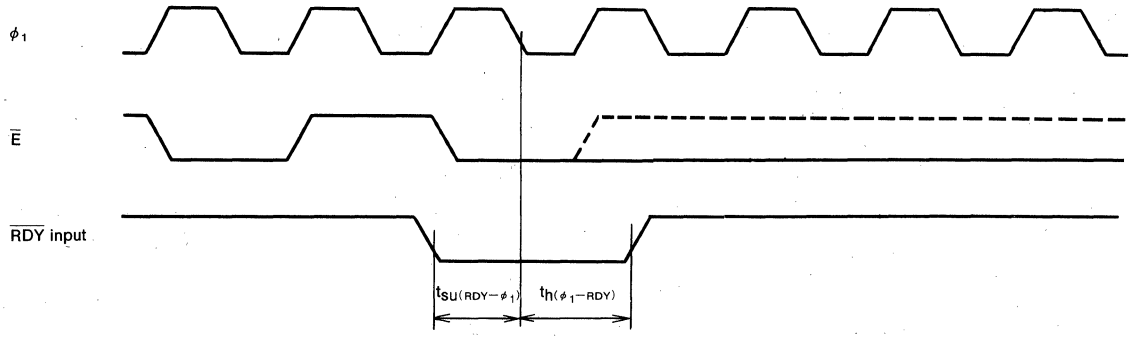
Single-chip mode



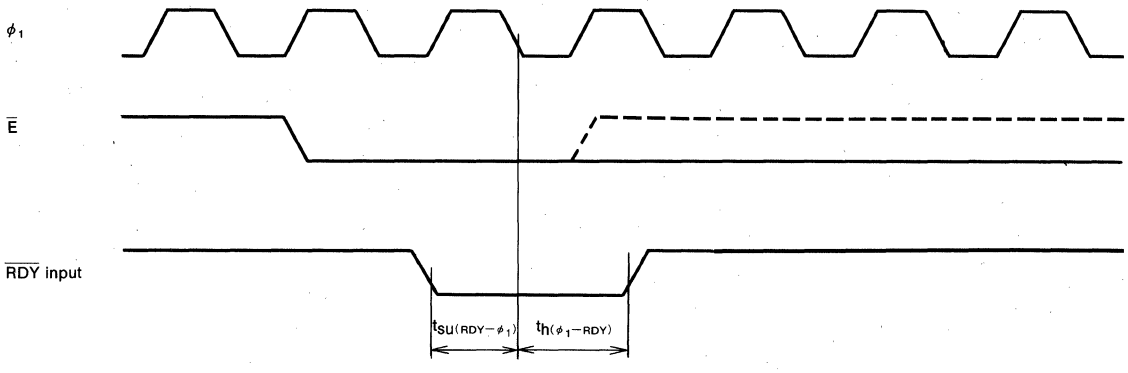

$t_{C(TA)}$
$t_{W(TAH)}$
TAi_{IN} input
$t_{W(TAL)}$
$t_{C(UP)}$
$t_{W(UPH)}$
TAi_{OUT} input
$t_{W(UPL)}$
TAi_{OUT} input
(Up-down input)
In Event counter mode
TAi_{IN} input
(When count by falling)
$t_{h(T_{IN}-UP)}$ $t_{SU(UP-T_{IN})}$
TAi_{IN} input
(When count by rising)
$T_{C(TB)}$
$t_{W(TBH)}$
$TB0_{IN}$ input
$t_{W(TBL)}$
$t_{C(AD)}$
$t_{W(ADL)}$
$\overline{AD_{TRG}}$ input
$t_{C(CK)}$
$t_{W(CKH)}$
CLK_0
$t_{W(CKL)}$
$t_{h(C-Q)}$
TxD_0
$t_{d(C-Q)}$
$t_{SU(D-C)}$
$t_{h(C-D)}$
RxD_0
$t_{W(INL)}$
$\overline{INT_i}$ input
$t_{W(INH)}$

Memory expansion mode and microprocessor mode

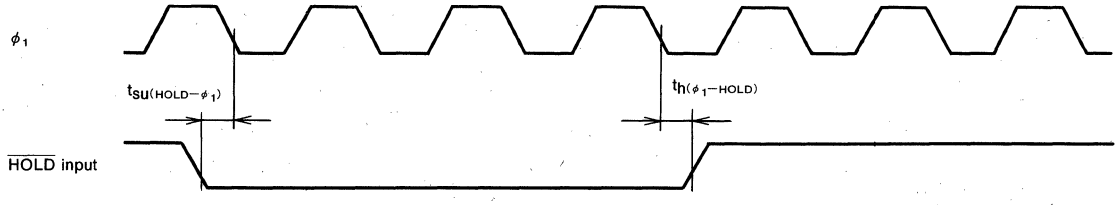
(When wait bit = "1")



(When wait bit = "0")



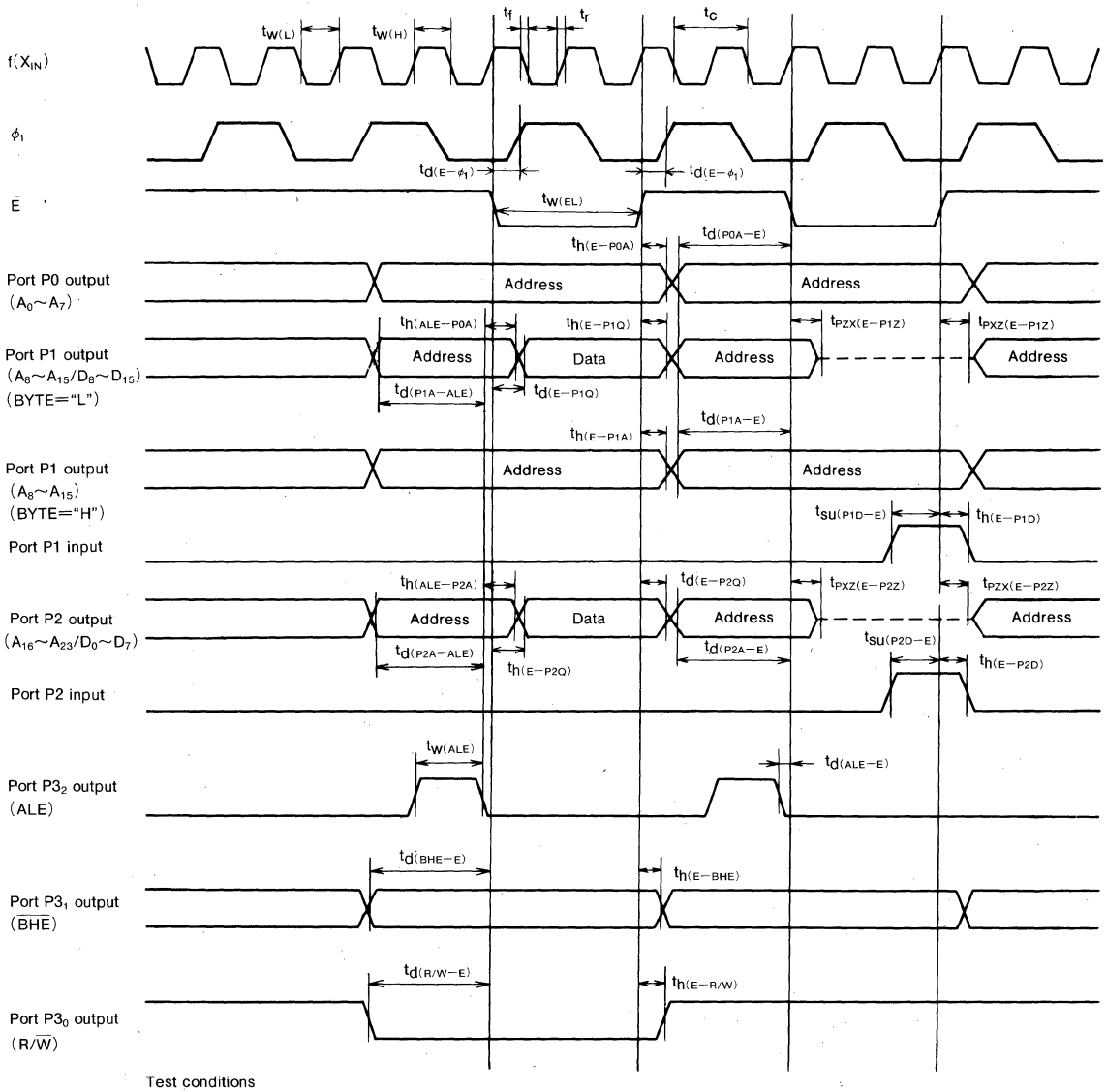
(When wait bit = "1" or "0" in common)



Test conditions

- V_{CC}= 5 V±10%
- Input timing voltage : V_{IL}=1.0V, V_{IH}=4.0V

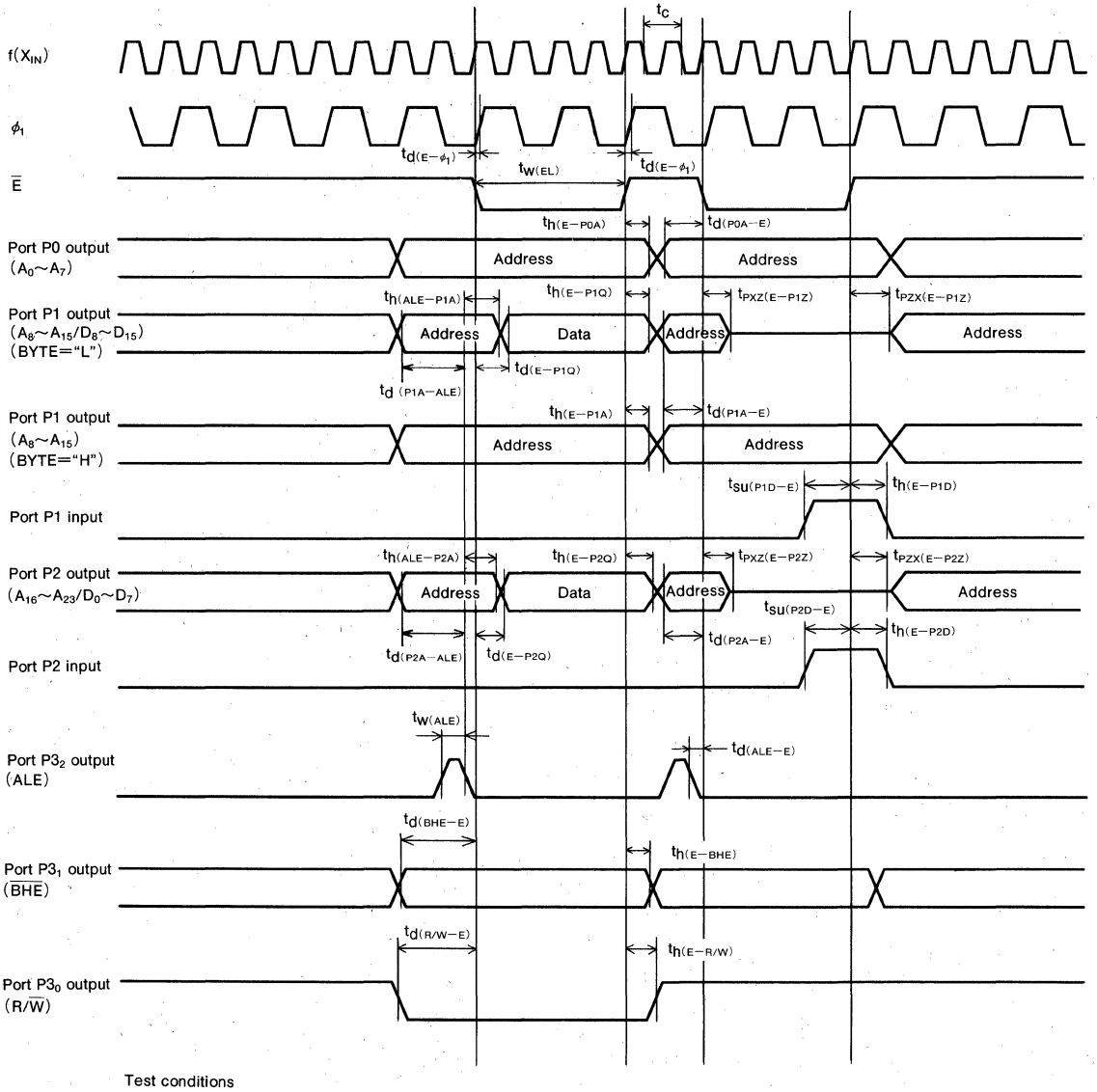
Memory expansion mode and microprocessor mode (When wait bit = "1")



Test conditions

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

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



Test conditions

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

M37703E4-XXXSP, M37703E4AXXXSP M37703E4BXXXSP

PROM VERSION of M37703M4-XXXSP, M37703M4AXXXSP, M37703M4BXXXSP

DESCRIPTION

The M37703E4-XXXSP, M37703E4AXXXSP and M37703E4BXXXSP are single-chip microcomputers designed with high-performance CMOS silicon gate technology. These are housed in a 64-pin shrink plastic molded DIP. The features of these chips are similar to those of the M37703M4-XXXSP, M37703M4AXXXSP and M37703M4BXXXSP except that these chips have a 32K-byte PROM built in.

These single-chip microcomputers have a large 16M bytes address space, three instruction queue buffers, and two data buffers for high-speed instruction execution. The CPU is a 16-bit parallel processor that can also be switched to perform 8-bit parallel processing. These microcomputers are suitable for office, business and industrial equipment controller that require high-speed processing of large data. Since general purpose PROM writers can be used for the built-in PROM, these chips are suitable for small quantity production runs.

The differences between M37703E4-XXXSP, M37703E4AXXXSP and M37703E4BXXXSP are the external clock input frequency as shown below. Therefore, the following descriptions will be for the M37703E4-XXXSP unless otherwise noted.

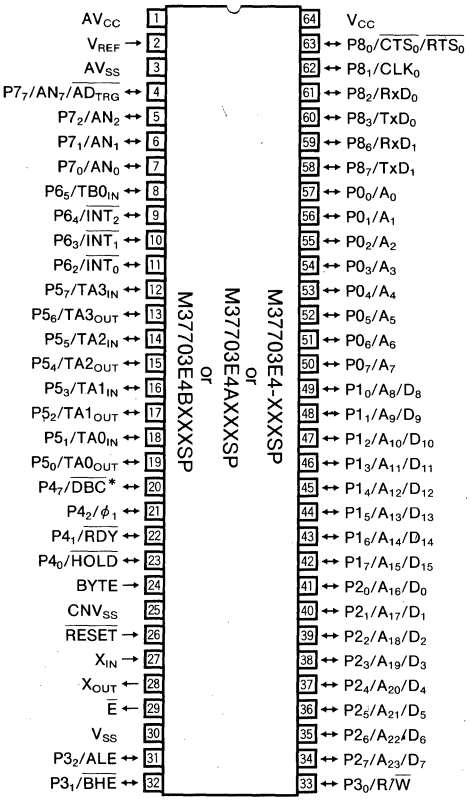
Type name	External clock input frequency
M37703E4-XXXSP	8 MHz
M37703E4AXXXSP	16MHz
M37703E4BXXXSP	25MHz

The M37703E4-XXXSP has the same functions as the M37703E2-XXXSP except for the memory size.

FEATURES

- Number of basic instructions 103
- Memory size PROM (one time) ..... 32K bytes
  RAM ..... 2048 bytes
- Instruction execution time
  M37703E4-XXXSP
  (The fastest instruction at 8 MHz frequency) 500ns
  M37703E4AXXXSP
  (The fastest instruction at 16 MHz frequency) 250ns
  M37703E4BXXXSP
  (The fastest instruction at 25 MHz frequency) 160ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 8 MHz frequency)
  ..... 30mW (Typ.)
- Interrupts ..... 19 types 7 levels
- Multiple function 16-bit timer ..... 5+3
- UART (may also be synchronous) 2
- 8-bit A-D converter ..... 4-channel inputs
- 12-bit watchdog timer
- Programmable input/output
  (ports P0, P1, P2, P3, P4, P5, P6, P7, P8) 53

PIN CONFIGURATION (TOP VIEW)



Outline 64P4B

* : Used in the evaluation chip mode only

APPLICATION

Control devices for office equipment such as copiers, printers, typewriters, facsimiles, word processors, and personal computers

Control devices for industrial equipment such as ME, NC, communication, and measuring instruments

THE FUNCTIONS AND CHARACTERISTICS

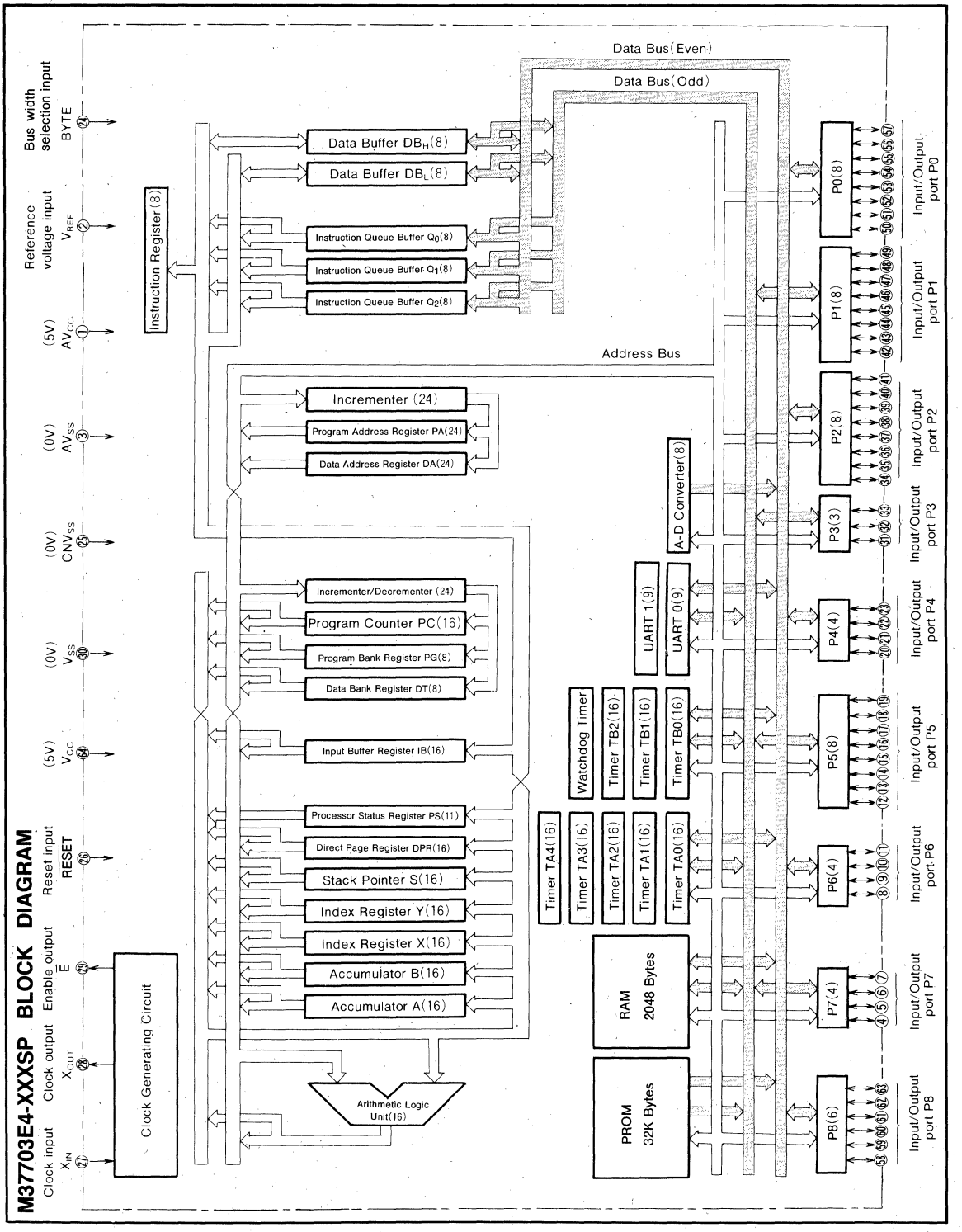
The M37703E4-XXXSP has the same functions and characteristics as the M37703E2-XXXSP except for the ROM and RAM size. Refer to the section on the M37703E2-XXXSP.

DATA REQUIRED FOR PROM ORDERING

Please send the following data for writing to PROM.

(1) M37703E4-XXXSP writing to PROM order confirmation form
(2) 64P4B mark specification form for one time PROM
(3) ROM data (EPROM 3sets)

PROM VERSION of M37703M4-XXXSP,M37703M4AXXXSP,M37703M4BXXXSP



MITSUBISHI ELECTRIC

APPENDICES

MELPS 7700 MASK ROM ORDERING METHOD

MASK ROM ORDERING METHOD

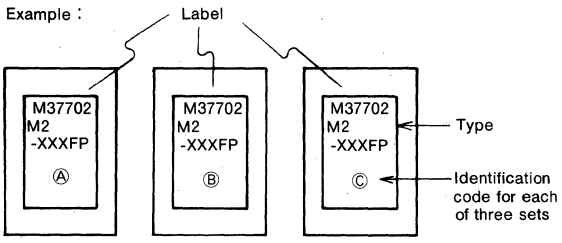
Mitsubishi Electric corp. accepts order to transfer EPROM supplied program data into the mask ROMs in single-chip 16-bit microcomputers. When placing such order, please submit the information described below.

1. Mask ROM Order Confirmation Form ····· 1 set
   (There is a specific form to be used for each model.)
2. Data to be written into mask ROM ····· EPROM
   (Please provide three sets containing the identical data.)
3. Mark Specification Form ····· 1 set

NOTES

(1) Acceptable EPROM type
Any EPROM made by Mitsubishi Electric corp. that is listed in the Mask ROM Order Confirmation Form may be used.

(2) EPROM window labeling
Please write the model name and the identification code (A, B, C) on the label for each of the three sets of data EPROMs provided.



(3) Calculation and indication of check sum code
Please calculate the total number of data in words in the EPROM, and write the number in 4-digit hexadecimal form in the check sum code field of the Mask ROM Order Confirmation Form.

(4) Options
Refer to the appropriate data book entry and write the desired options on the Mask ROM Order Confirmation Form.

(5) Marking specification method
The permissible marking specifications differ depending on the shape of package. Please fill out the Marking Specification Form and attach it to the Mask ROM Order Confirmation Form.

OUTLINE OF ORDER PROCESSING

Mitsubishi Electric corp. will produce the mask ROM if at least two of the three EPROM sets submitted contain identical data.

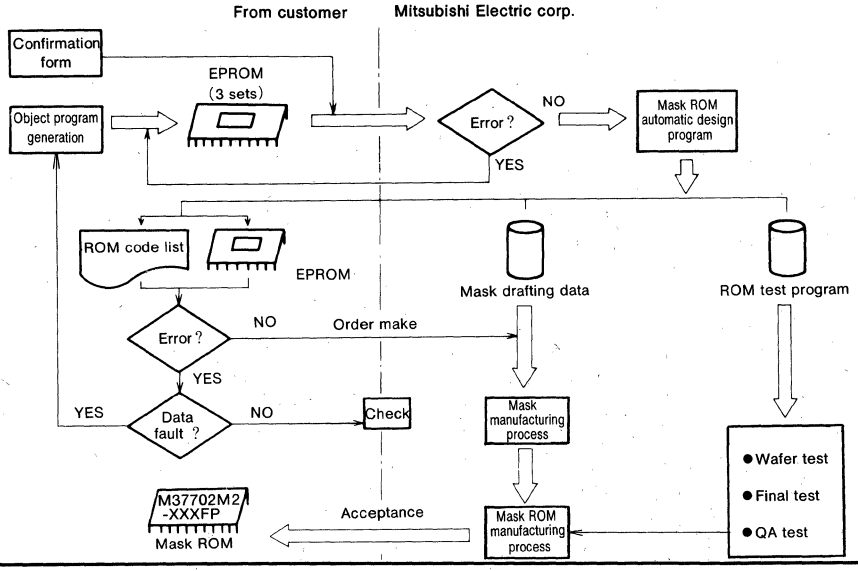
If we find error when the submitted EPROMs are compared, we will contact your representative. Thus, we assume responsibility only when we produce the mask ROMs that contain data other than the data correctly provided by the customer.

Mitsubishi Electric corp. uses an automatic mask ROM design program to generated the forlling:

1 : Drafting data for mask ROM production;
2 : ROM code listing or EPROM for mask ROM production error check work;
3 : Mask ROM test program.

The chart below shows the flow of mask ROM production.

MELPS 7700 MASK ROM DEVELOPMENT CAD SYSTEM



GZZ—SH02—45A〈99A0〉

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

MELPS 7700 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER M37702M2-XXXFP
MITSUBISHI ELECTRIC

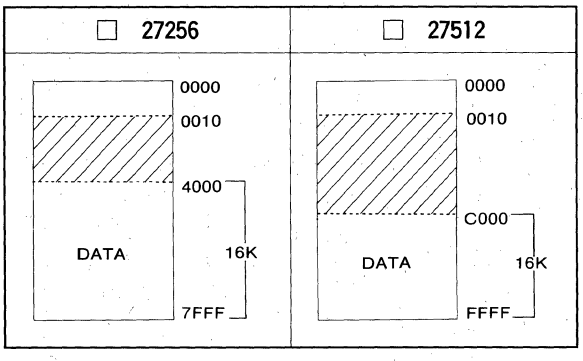
Note : Please fill in all items marked※

※	Customer	Company name	TEL ( )	Issuance signatures	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 on 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) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options. This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

	Address		Address		Address
4D	0	2D	8	Option data	10
33	1	FF	9		
37	2	FF	A		
37	3	FF	B		
30	4	FF	C		
32	5	FF	D		
4D	6	FF	E		
32	7	FF	F		

※2. STP instruction option
One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered. Check @ in the appropriate box.

☐ STP instruction enable [01_{16}] Address 10_{16}
☐ STP instruction disable [00_{16}] Address 10_{16}

※3. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702M2-XXXFP) and attach to the Mask ROM Order Confirmation Form.

※4. Comments

GZZ—SH02—46A〈99A0〉

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

MELPS 7700 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER M37702M2AXXXFP
MITSUBISHI ELECTRIC

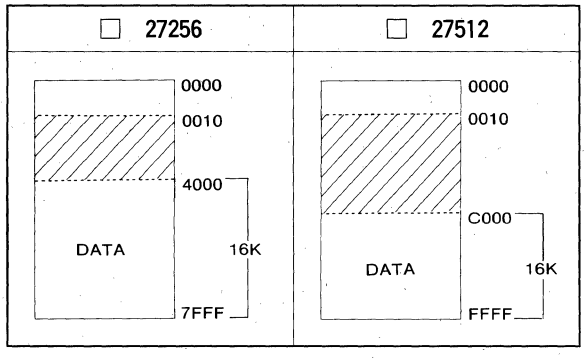
Note : Please fill in all items marked※

※	Customer	Company name	TEL ( )	Issuance signatures	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 on 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) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options. This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

	Address		Address		Address
4D	0	41	8	Option data	10
33	1	FF	9		
37	2	FF	A		
37	3	FF	B		
30	4	FF	C		
32	5	FF	D		
4D	6	FF	E		
32	7	FF	F		

※2. STP instruction option
One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered. Check @ in the appropriate box.

☐ STP instruction enable [01_{16}] Address 10_{16}
☐ STP instruction disable [00_{16}] Address 10_{16}

※3. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702M2AXXXFP) and attach to the Mask ROM Order Confirmation Form.

※4. Comments

MELPS 7700 MASK ROM ORDERING METHOD

GZZ—SH02—47A〈99A0〉

MELPS 7700 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER M37702M2BXXXFP
MITSUBISHI ELECTRIC

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

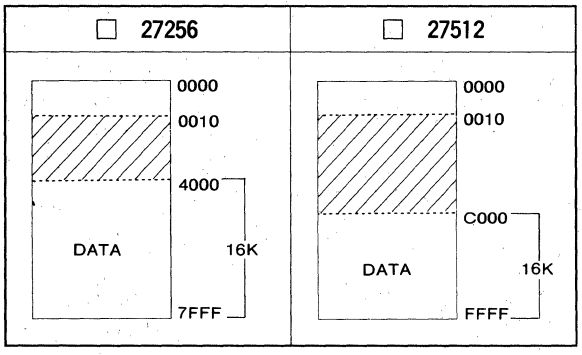
Note : Please fill in all items marked※

※	Customer	Company name	TEL ( )	Issuance signatures	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 on 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) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options. This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

	Address		Address		Address
4D	0	42	8	Option data	10
33	1	FF	9		
37	2	FF	A		
37	3	FF	B		
30	4	FF	C		
32	5	FF	D		
4D	6	FF	E		
32	7	FF	F		

※2. STP instruction option
One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered. Check @ in the appropriate box.

☐ STP instruction enable [01_{16}] Address 10_{16}
☐ STP instruction disable [00_{16}] Address 10_{16}

※3. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702M2BXXXFP) and attach to the Mask ROM Order Confirmation Form.

※4. Comments

MITSUBISHI ELECTRIC

MELPS 7700 MASK ROM ORDERING METHOD

GZZ—SH02—36A〈98A0〉

MELPS 7700 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER M37702M4-XXXFP
MITSUBISHI ELECTRIC

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

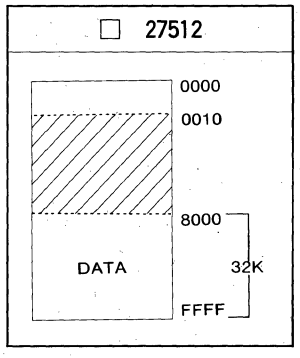
					Issuance signatures	Responsible officer	Supervisor
※	Customer	Company name	TEL ( )				
		Date issued	Date :				

※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 on 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) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options. This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

	Address		Address		Address
4D	0	2D	8	Option data	10
33	1	FF	9		
37	2	FF	A		
37	3	FF	B		
30	4	FF	C		
32	5	FF	D		
4D	6	FF	E		
34	7	FF	F		

※2. STP instruction option

One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered. Check @ in the appropriate box.

☐ STP instruction enable 01_{16} Address 10_{16}
☐ STP instruction disable 00_{16} Address 10_{16}

※3. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702M4-XXXFP) and attach to the Mask ROM Order Confirmation Form.

※4. Comments

GZZ—SH02—37A〈98A0〉

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

MELPS 7700 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER M37702M4AXXXFP
MITSUBISHI ELECTRIC

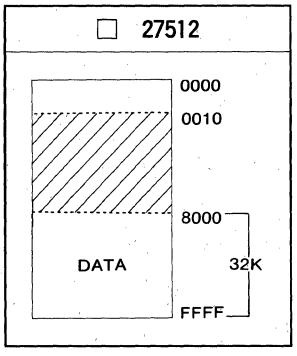
Note : Please fill in all items marked※

※ Customer	Company name	TEL ( )	Issuance signatures	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 on 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) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options. This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

	Address		Address		Address
4D	0	41	8	Option data	10
33	1	FF	9		
37	2	FF	A		
37	3	FF	B		
30	4	FF	C		
32	5	FF	D		
4D	6	FF	E		
34	7	FF	F		

※2. STP instruction option
One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered. Check @ in the appropriate box.

☐ STP instruction enable [01_{16}] Address 10_{16}
☐ STP instruction disable [00_{16}] Address 10_{16}

※3. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702M4AXXXFP) and attach to the Mask ROM Order Confirmation Form.

※4. Comments

MELPS 7700 MASK ROM ORDERING METHOD

GZZ—SH02—38A〈98A0〉

MELPS 7700 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER M37702M4BXXXFP
MITSUBISHI ELECTRIC

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

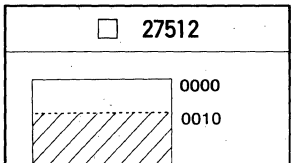
※	Customer	Company name	TEL ( )	Issuance signatures	Responsible officer	Supervisor
		Date issued	Date :			

※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 on 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) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options. This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

Address		Address		Address
4D	0	42	8	Option data 10
33	1	FF	9	
37	2	FF	A	
37	3	FF	B	
30	4	FF	C	
32	5	FF	D	
4D	6	FF	E	
34	7	FF	F	

※2. STP instruction option

One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered. Check @ in the appropriate box.

☐ STP instruction enable 01_{16} Address 10_{16}
☐ STP instruction disable 00_{16} Address 10_{16}

※3. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702M4BXXXFP) and attach to the Mask ROM Order Confirmation Form.

※4. Comments

GZZ—SH02—51A〈99A0〉

MELPS 7700 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER M37703M2-XXXSP
MITSUBISHI ELECTRIC

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

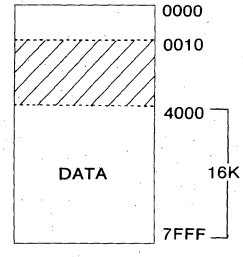
※	Customer	Company name	TEL ( )	Issuance signatures	Responsible officer	Supervisor
		Date issued	Date :			

※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 on 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.

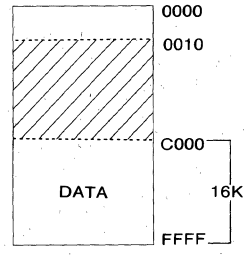
Checksum code for entire EPROM areas [ ][ ][ ][ ] (hexadecimal notation)

EPROM Type :

☐ 27256



☐ 27512



(1) Set "FF_{16}" in the shaded area.
(2) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options. This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

	Address		Address		Address
4D	0	2D	8	Option data	10
33	1	FF	9		
37	2	FF	A		
37	3	FF	B		
30	4	FF	C		
33	5	FF	D		
4D	6	FF	E		
32	7	FF	F		

※2. STP instruction option
One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered. Check @ in the appropriate box.

☐ STP instruction enable 01_{16} Address 10_{16}
☐ STP instruction disable 00_{16} Address 10_{16}

※3. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form (for M37703M2-XXXSP) and attach to the Mask ROM Order Confirmation Form.

※4. Comments

MELPS 7700 MASK ROM ORDERING METHOD

GZZ—SH02—52A〈99A0〉

MELPS 7700 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER M37703M2AXXXSP
MITSUBISHI ELECTRIC

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

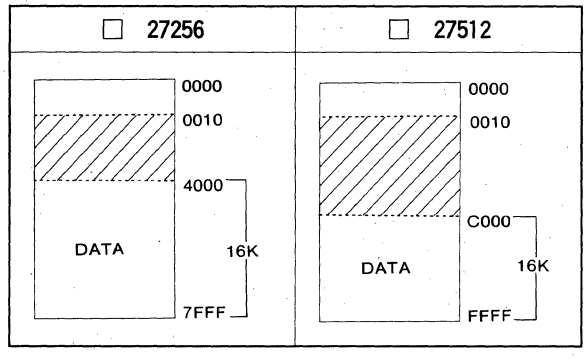
※ Customer	Company name	TEL ( )	Issuance signatures	Responsible officer	Supervisor
	Date issued	Date :			

※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 on 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) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options. This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

	Address		Address		Address
4D	0	41	8	Option data	10
33	1	FF	9		
37	2	FF	A		
37	3	FF	B		
30	4	FF	C		
33	5	FF	D		
4D	6	FF	E		
32	7	FF	F		

※2. STP instruction option

One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered. Check @ in the appropriate box.

☐ STP instruction enable [01_{16}] Address 10_{16}
☐ STP instruction disable [00_{16}] Address 10_{16}

※3. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form (for M37703M2AXXXSP) and attach to the Mask ROM Order Confirmation Form.

※4. Comments

GZZ—SH02—53A〈99A0〉

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

MELPS 7700 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER M37703M2BXXXSP
MITSUBISHI ELECTRIC

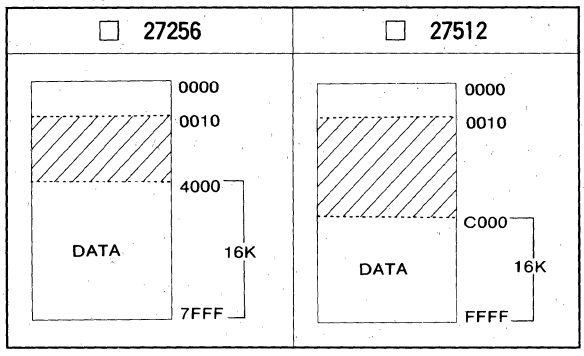
Note : Please fill in all items marked※

					Responsible officer	Supervisor
※	Customer	Company name	TEL ( )	Issuance signatures		
		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 on 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) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options. This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

	Address		Address		Address
4D	0	42	8	Option data	10
33	1	FF	9		
37	2	FF	A		
37	3	FF	B		
30	4	FF	C		
33	5	FF	D		
4D	6	FF	E		
32	7	FF	F		

※2. STP instruction option
One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered. Check @ in the appropriate box.

☐ STP instruction enable 01_{16} Address 10_{16}
☐ STP instruction disable 00_{16} Address 10_{16}

※3. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form (for M37703M2BXXXSP) and attach to the Mask ROM Order Confirmation Form.

※4. Comments

MELPS 7700 MASK ROM ORDERING METHOD

GZZ—SH02—42A〈98A0〉

MELPS 7700 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER M37703M4-XXXSP
MITSUBISHI ELECTRIC

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

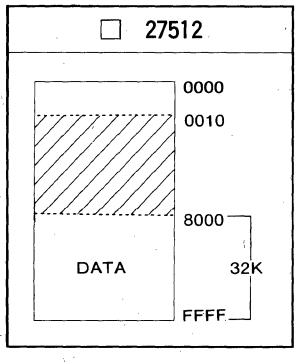
※	Customer	Company name	TEL ( )	Issuance signatures	Responsible officer	Supervisor
		Date issued	Date :			

※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 on 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) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options. This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

	Address		Address		Address
4D	0	2D	8	Option data	10
33	1	FF	9		
37	2	FF	A		
37	3	FF	B		
30	4	FF	C		
33	5	FF	D		
4D	6	FF	E		
34	7	FF	F		

※2. STP instruction option

One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered. Check @ in the appropriate box.

☐ STP instruction enable 01_{16} Address 10_{16}
☐ STP instruction disable 00_{16} Address 10_{16}

※3. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form (for M37703M4-XXXSP) and attach to the Mask ROM Order Confirmation Form.

※4. Comments

MELPS 7700 MASK ROM ORDERING METHOD

GZZ—SH02—43A〈98A0〉

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

MELPS 7700 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER M37703M4AXXXSP
MITSUBISHI ELECTRIC

Note : Please fill in all items marked※

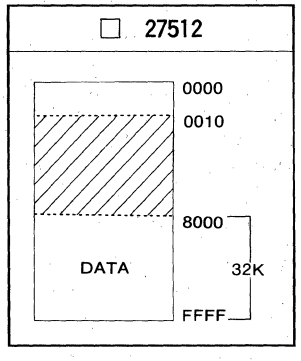
※	Customer	Company name	TEL ( )	Issuance signatures	Responsible officer	Supervisor
		Date issued	Date :			

※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 on 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) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options. This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

	Address		Address		Address
4D	0	41	8	Option data	10
33	1	FF	9		
37	2	FF	A		
37	3	FF	B		
30	4	FF	C		
33	5	FF	D		
4D	6	FF	E		
34	7	FF	F		

※2. STP instruction option

One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered. Check @ in the appropriate box.

☐ STP instruction enable [01_{16}] Address 10_{16}
☐ STP instruction disable [00_{16}] Address 10_{16}

※3. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form (for M37703M4AXXXSP) and attach to the Mask ROM Order Confirmation Form.

※4. Comments

MITSUBISHI ELECTRIC

MELPS 7700 MASK ROM ORDERING METHOD

GZZ—SH02—44A〈98A0〉

MELPS 7700 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER M37703M4BXXXSP
MITSUBISHI ELECTRIC

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

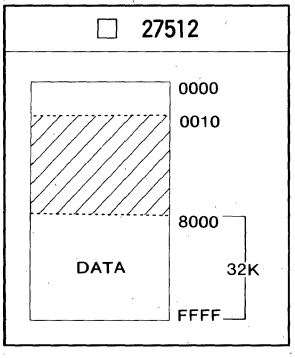
※	Customer	Company name	TEL ( )	Issuance signatures	Responsible officer	Supervisor
		Date issued	Date :			

※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 on 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) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options. This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

	Address		Address		Address
4D	0	42	8	Option data	10
33	1	FF	9		
37	2	FF	A		
37	3	FF	B		
30	4	FF	C		
33	5	FF	D		
4D	6	FF	E		
34	7	FF	F		

※2. STP instruction option

One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered. Check @ in the appropriate box.

☐ STP instruction enable [01_{16}] Address 10_{16}
☐ STP instruction disable [00_{16}] Address 10_{16}

※3. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form (for M37703M4BXXXSP) and attach to the Mask ROM Order Confirmation Form.

※4. Comments

MELPS 7700 PROM ORDERING METHOD

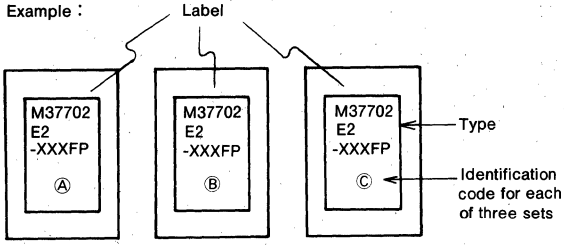
PROM ORDERING METHOD

Mitsubishi Electric corp. accepts order to transfer EPROM supplied program data into the one time PROMs in single-chip 16-bit microcomputers. When placing such order, please submit the information described below.

1. Writing to PROM Order Confirmation Form ····· 1 set (There is a specific form to be used for each model.)
2. Data to be written into PROM built in ····· EPROM (Please provide three sets containing the identical data.)
3. Mark Specification Form ····· 1 set

NOTES

(1) Acceptable EPROM type
Any EPROM made by Mitsubishi Electric corp. that is listed in the Writing to PROM Order Confirmation Form may be used.



(2) EPROM window labeling
Please write the model name and the identification code (A, B, C) on the label for each of the three sets of data EPROMs provided.

(3) Calculation and indication of check sum code
Please calculate the total number of data in words in the EPROM, and write the number in 4-digit hexadecimal form in the check sum code field of the Writing to PROM Order Confirmation Form.

(4) Marking specification method
The permissible marking specifications differ depending on the shape of package. Please fill out the Marking Specification Form and attach it to the Writing to PROM Order Confirmation Form.

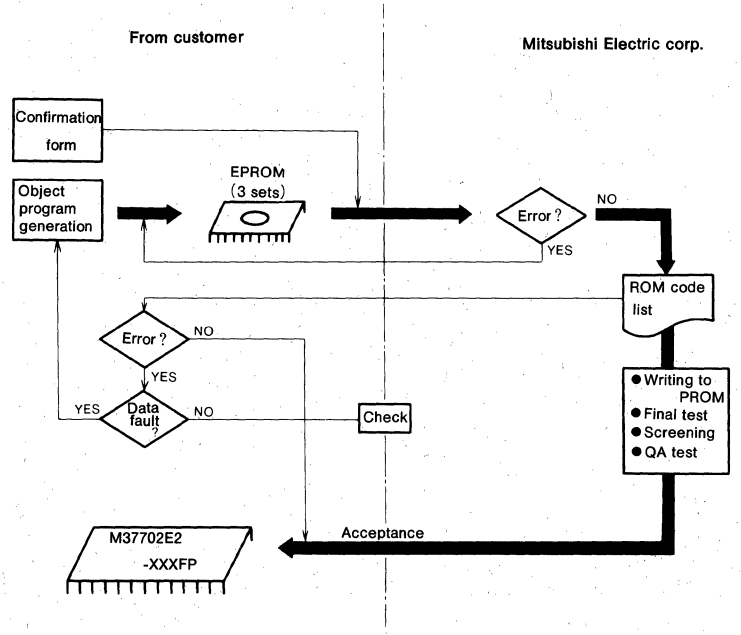
OUTLINE OF ORDER PROCESSING

Mitsubishi Electric corp. will produce Writing to PROM if at least two of the three EPROM sets submitted contain identical data.

If we find error when the submitted EPROMs are compared, we will contact your representative. Thus, we assume responsibility only when we produce Writing to PROMs that contain data other than the data correctly provided by the customer.

The chart below shows the flow of one time PROM production.

MELPS 7700 ONE TIME PROM DEVELOPMENT CAD SYSTEM



MITSUBISHI ELECTRIC

GZZ—SH02—54A〈99A0〉

ROM number	

MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37702E2-XXXFP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

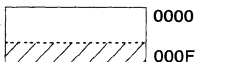
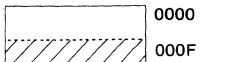
Note : Please fill in all items marked※

※ Customer	Company name	TEL ( )	Issuance signatures	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 writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM 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 :

☐ 27256	☐ 27512
	

(1) Set "FF_{16}" in the shaded area.
(2) Address 0_{16} to $0F_{16}$ are the area for storing the data on model designation. This area must be written with the data shown below.
Address and data are written in hexadecimal notation.

	Address		Address
4D	0	2D	8
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
32	5	FF	D
45	6	FF	E
32	7	FF	F

※2. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702E2-XXXFP) and attach to the Writing to PROM Order Confirmation Form.

※3. Comments

MELPS 7700 PROM ORDERING METHOD

GZZ—SH02—55A〈99A0〉

ROM number	

MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37702E2AXXXFP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

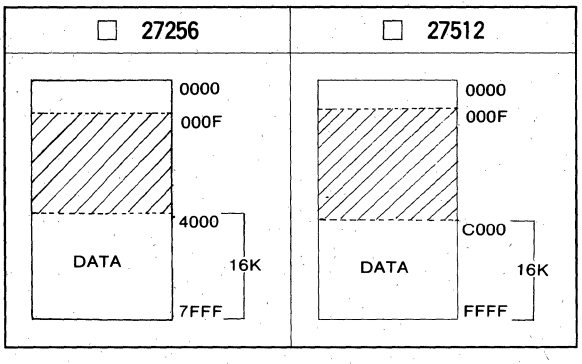
※	Customer	Company name	TEL ( )	Issuance signatures	Responsible officer	Supervisor
		Date issued	Date :			

※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 writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM 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) Address 0_{16} to $0F_{16}$ are the area for storing the data on model designation. This area must be written with the data shown below.
Address and data are written in hexadecimal notation.

	Address		Address
4D	0	41	8
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
32	5	FF	D
45	6	FF	E
32	7	FF	F

※2. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702E2AXXXFP) and attach to the Writing to PROM Order Confirmation Form.

※3. Comments

MELPS 7700 PROM ORDERING METHOD

GZZ—SH04—07A〈0ZA0〉

ROM number	

MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37702E2BXXXFP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

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

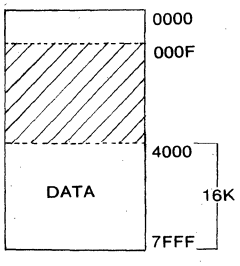
※1. Confirmation

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 writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM 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.

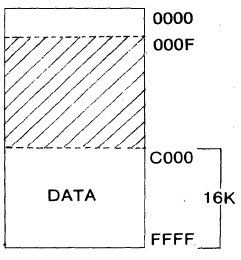
Checksum code for entire EPROM areas | | | | | (hexadecimal notation)

EPROM Type :

☐ 27256



☐ 27512



(1) Set "FF_{16}" in the shaded area.
(2) Address 0_{16} to $0F_{16}$ are the area for storing the data on model designation. This area must be written with the data shown below.
Address and data are written in hexadecimal notation.

	Address		Address
4D	0	42	8
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
32	5	FF	D
45	6	FF	E
32	7	FF	F

※2. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702E2BXXXFP) and attach to the Writing to PROM Order Confirmation Form.

※3. Comments

MELPS 7700 PROM ORDERING METHOD

GZZ—SH03—67A〈07A0〉

ROM number	

MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37702E4-XXXFP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

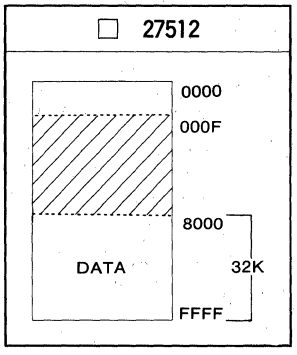
※	Customer	Company name	TEL ( )	Issuance signatures	Responsible officer	Supervisor
		Date issued	Date :			

※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 writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM 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) Address 0_{16} to $0F_{16}$ are the area for storing the data on model designation. This area must be written with the data shown below.
Address and data are written in hexadecimal notation.

	Address		Address
4D	0	2D	8
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
32	5	FF	D
45	6	FF	E
34	7	FF	F

※2. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702E4-XXXFP) and attach to the Writing to PROM Order Confirmation Form.

※3. Comments

MITSUBISHI ELECTRIC

GZZ—SH03—70A〈07A0〉

ROM number	

MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37702E4AXXXFP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

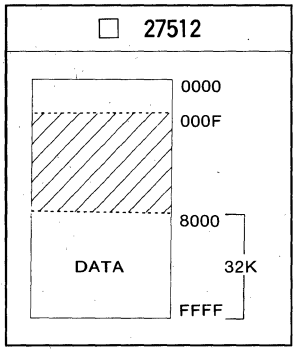
Note : Please fill in all items marked※

※	Customer	Company name	TEL ( )	Issuance signatures	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 writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM 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) Address 0_{16} to $0F_{16}$ are the area for storing the data on model designation. This area must be written with the data shown below.
Address and data are written in hexadecimal notation.

	Address		Address
4D	0	41	8
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
32	5	FF	D
45	6	FF	E
34	7	FF	F

※2. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702E4AXXXFP) and attach to the Writing to PROM Order Confirmation Form.

※3. Comments

MELPS 7700 PROM ORDERING METHOD

GZZ—SH03—38A〈01A0〉

ROM number	

MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37702E4BXXXFP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

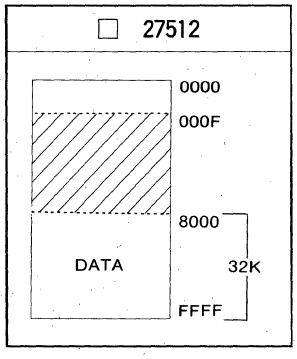
※ Customer	Company name	TEL ( )	Issuance signatures	Responsible officer	Supervisor
	Date issued	Date :			

※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 writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM 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) Address 0_{16} to $0F_{16}$ are the area for storing the data on model designation. This area must be written with the data shown below.
Address and data are written in hexadecimal notation.

	Address		Address
4D	0	42	8
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
32	5	FF	D
45	6	FF	E
34	7	FF	F

※2. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702E4BXXXFP) and attach to the Writing to PROM Order Confirmation Form.

※3. Comments

MITSUBISHI ELECTRIC

GZZ—SH02—58A〈99A0〉

MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37703E2-XXXSP
MITSUBISHI ELECTRIC

ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

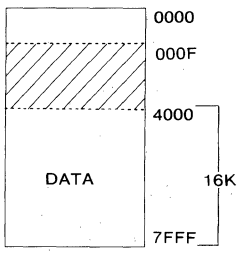
※	Customer	Company name	TEL ( )	Issuance signatures	Responsible officer	Supervisor
		Date issued	Date :			

※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 writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM 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.

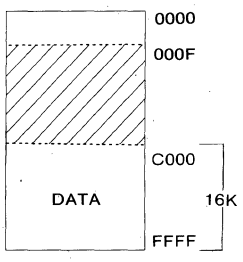
Checksum code for entire EPROM areas | | | | | (hexadecimal notation)

EPROM Type :

☐ 27256



☐ 27512



(1) Set "FF_{16}" in the shaded area.
(2) Address 0_{16} to $0F_{16}$ are the area for storing the data on model designation. This area must be written with the data shown below.
Address and data are written in hexadecimal notation.

	Address		Address
4D	0	2D	8
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
33	5	FF	D
45	6	FF	E
32	7	FF	F

※2. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form for one time PROM (for M37703E2-XXXSP) and attach to the Writing to PROM Order Confirmation Form.

※3. Comments

MELPS 7700 PROM ORDERING METHOD

GZZ—SH02—59A〈99A0〉

ROM number	

MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37703E2AXXXSP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

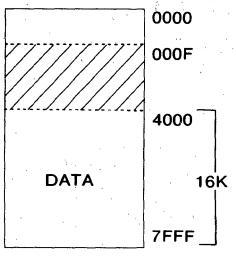
※	Customer	Company name	TEL ( )	Issuance signatures	Responsible officer	Supervisor
		Date issued	Date :			

※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 writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM 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.

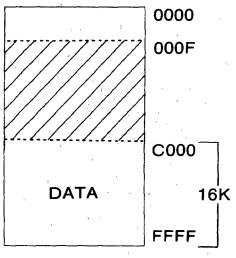
Checksum code for entire EPROM areas [ ][ ][ ][ ] (hexadecimal notation)

EPROM Type :

☐ 27256



☐ 27512



(1) Set "FF_{16}" in the shaded area.
(2) Address 0_{16} to $0F_{16}$ are the area for storing the data on model designation. This area must be written with the data shown below.
Address and data are written in hexadecimal notation.

	Address		Address
4D	0	41	8
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
33	5	FF	D
45	6	FF	E
32	7	FF	F

※2. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form for one time PROM (for M37703E2AXXXSP) and attach to the Writing to PROM Order Confirmation Form.

※3. Comments

GZZ—SH04—08A〈0ZA0〉

MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37703E2BXXXSP
MITSUBISHI ELECTRIC

ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

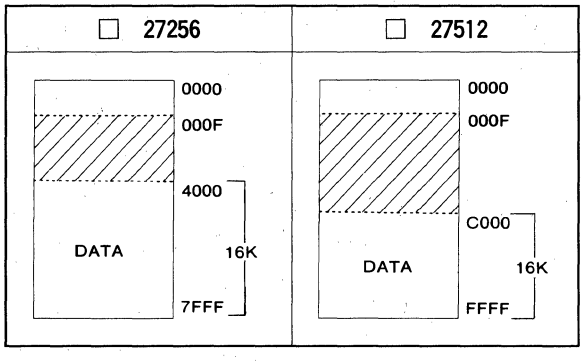
Note : Please fill in all items marked※

※	Customer	Company name	TEL ( )	Issuance signatures	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 writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM 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) Address 0_{16} to $0F_{16}$ are the area for storing the data on model designation. This area must be written with the data shown below.
Address and data are written in hexadecimal notation.

	Address		Address
4D	0	42	8
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
33	5	FF	D
45	6	FF	E
32	7	FF	F

※2. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form for one time PROM (for M37703E2BXXXSP) and attach to the Writing to PROM Order Confirmation Form.

※3. Comments

GZZ—SH03—69A〈07A0〉

ROM number	

MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37703E4-XXXSP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

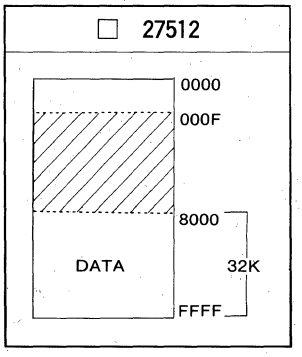
Note : Please fill in all items marked※

※	Customer	Company name	TEL ( )	Issuance signatures	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 writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM 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) Address 0_{16} to $0F_{16}$ are the area for storing the data on model designation. This area must be written with the data shown below.
Address and data are written in hexadecimal notation.

	Address		Address
4D	0	2D	8
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
33	5	FF	D
45	6	FF	E
34	7	FF	F

※2. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form for one time PROM (for M37703E4-XXXSP) and attach to the Writing to PROM Order Confirmation Form.

※3. Comments

MELPS 7700 PROM ORDERING METHOD

GZZ—SH03—68A〈07A0〉

ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37703E4AXXXSP
MITSUBISHI ELECTRIC

Note : Please fill in all items marked※

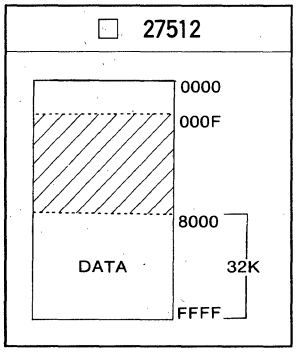
※	Customer	Company name	TEL ( )	Issuance signatures	Responsible officer	Supervisor
		Date issued	Date :			

※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 writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM 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) Address 0_{16} to $0F_{16}$ are the area for storing the data on model designation. This area must be written with the data shown below.
Address and data are written in hexadecimal notation.

	Address		Address
4D	0	41	8
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
33	5	FF	D
45	6	FF	E
34	7	FF	F

※2. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form for one time PROM (for M37703E4AXXXSP) and attach to the Writing to PROM Order Confirmation Form.

※3. Comments

MELPS 7700 PROM ORDERING METHOD

GZZ—SH03—41A〈01A0〉

ROM number	

MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37703E4BXXXSP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

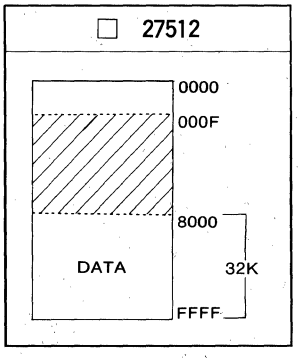
Note : Please fill in all items marked※

※	Customer	Company name	TEL ( )	Issuance signatures	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 writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM 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) Address 0_{16} to $0F_{16}$ are the area for storing the data on model designation. This area must be written with the data shown below.
Address and data are written in hexadecimal notation.

	Address		Address
4D	0	42	8
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
33	5	FF	D
45	6	FF	E
34	7	FF	F

※2. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form for one time PROM (for M37703E4BXXXSP) and attach to the Writing to PROM Order Confirmation Form.

※3. Comments

MARK SPECIFICATION FORM

MARK SPECIFICATION FORM

Mark specification format differs depending on the package type.

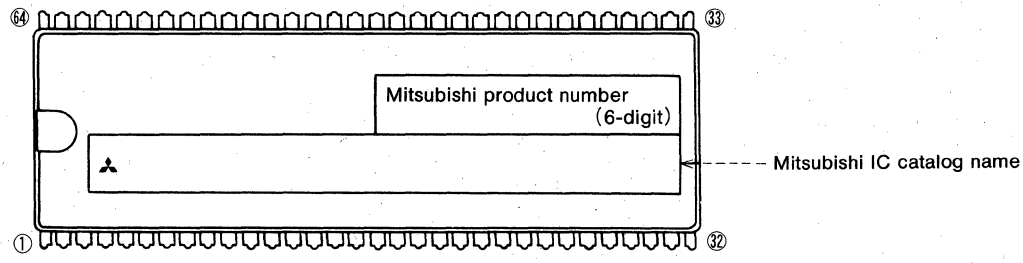
Fill out the Mark Specification Form for the package type being ordered, and submit the form with the Mask ROM Confirmation Form.

64P4B (64-PIN SHRINK DIP) MARK SPECIFICATION FORM

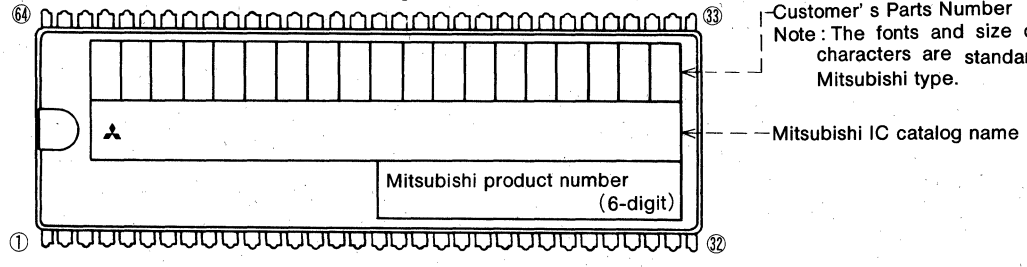
Mitsubishi IC catalog name

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



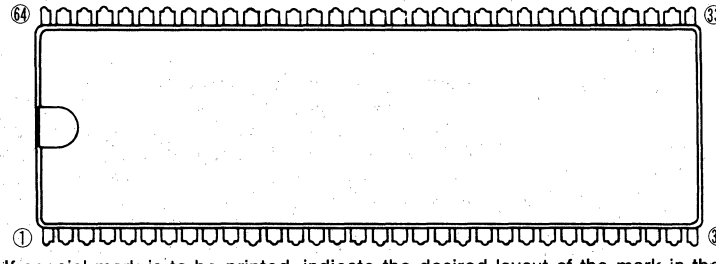
B. Customer' s Parts Number + Mitsubishi IC Catalog Name



Note1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type.
3 : Customer' s parts number can be up to 19 alphanumeric characters for capital letters, hyphens, commas, periods and so on.
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 special mark is to be printed, indicate the desired layout of the mark in the upper figure. The layout will be duplicated technically as close as possible. Mitsubishi product number (6-digit) and Mask ROM number (3-digit) are always marked for sorting the products.
2 : If special character fonts (e. g., customer's trade mark logo) must be used in special mark, check the box on the right.
For the new special character fonts a clean font original (ideally logo drawing) must be submitted.

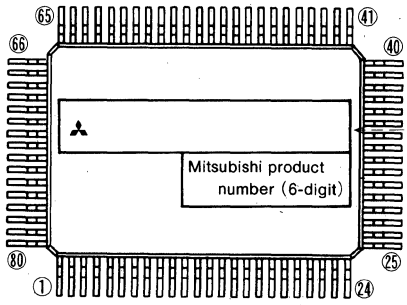
Special character fonts required ☐

80P6N (80-PIN QFP) MARK SPECIFICATION FORM

Mitsubishi IC catalog name

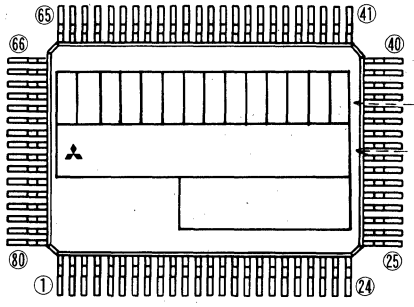
Please choose one of the marking types below (A, B, C), and enter the Mitsubishi IC catalog name and the special mark (if needed).

A. Standard Mitsubishi Mark



Mitsubishi IC catalog name

B. Customer's Parts Number + Mitsubishi IC Catalog Name



Customer's Parts Number

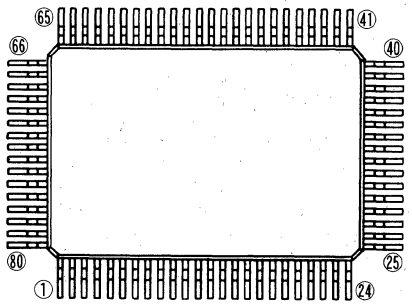
Note : The fonts and size of characters are standard Mitsubishi type.

Mitsubishi IC catalog name

Note1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type.
3 : Customer's parts number can be up to 14 alphanumeric characters for capital letters, hyphens, commas, periods and so on.
4 : If the Mitsubishi logo is not required, check the box below.

Mitsubishi logo is not required ☐

C. Special Mark Required



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

Special character fonts required ☐

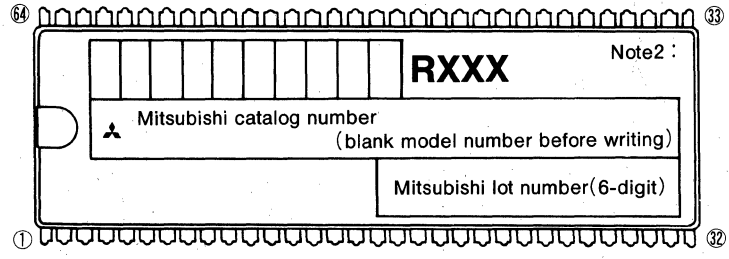
64P4B(64-PIN SHRINK DIP)MARK SPECIFICATION FORM for one time PROM version microcomputers

Enter the catalog number of the microcomputer for which this mark specification is intended.(If you do not know the ROM code number, enter XXX in its place.)

the catalog number of the microcomputer | M |

A. Standard Mitsubishi Mark

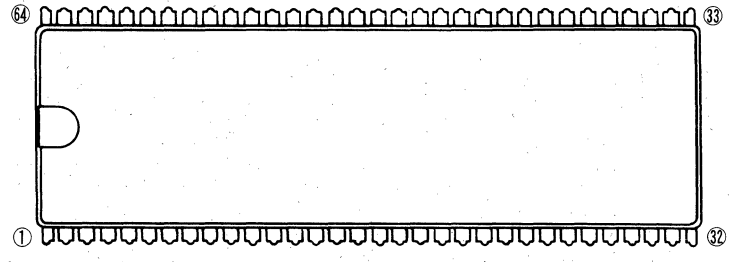
Customer specified part number will be printed together with the ROM code number on the top line.
Enter the desired part number left aligned in the box below.(up to 10 characters)



Note1 : The following characters can be used in the part number :
Uppercase alphabet, numbers, ampersand, hyphen, period, comma, +, /, (,), ©
(©will be printed at 1,5X character width)
2 : XXX is the ROM code number.

B. Special Mark Required

If you desire anything other than the standard Mitsubishi mark, it will be treated as a special mark.
Special marks will take longer to produce and should be avoided if possible.
If a special mark is to be printed, indicate the desired layout of the mark in the figure below.The layout will be duplicated as closely as possible.



Note1 : If the customer's trademark logo must be used in the special mark, please submit a clean original logo.
Note that special marks require extra cost and time to produce.

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.
25 Floor, Leighton Centre,
77, Leighton Road. Causeway Bay.
Hong Kong
Telex: 60800 MELCO HX
Telephone: (5) 773901-3
Facsimile: (5) 895-3104

SINGAPORE
MELCO SALES SINGAPORE PTE. LTD.
230 Upper Bukit Timah Road # 03-01/15
Hock Soon Industrial Complex
Singapore 2158
Telex: RS 20845 MELCO
Telephone: 4695255
Facsimile: 4695347

TAIWAN
MELCO-TAIWAN CO., Ltd.
1st fl., Chung-Ling Bldg.,
363, Sec. 2, Fu-Hsing S Road,
Taipei R.O.C.
Telephone: (02) 735-3030
Facsimile: (02) 735-6771
Telex: 25433 CHURYO "MELCO-TAIWAN"

U.S.A.

NORTHWEST
Mitsubishi Electronics America, Inc.
1050 East Arques Avenue
Sunnyvale, CA 94086
Telephone: (408) 730-5900
Facsimile: (408) 730-4972

SAN DIEGO
Mitsubishi Electronics America, Inc.
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

GERMANY
Mitsubishi Electric Europe GmbH
Headquarters:
Gothear Str. 8
4030 Ratingen 1, Germany
Telex: 8585070 MED D
Telephone: (02102) 4860
Facsimile: (02102) 486-115

Munich Office:
Arabellastraße 31
8000 München 81, Germany
Telex: 5214820
Telephone: (089) 919006-09
Facsimile: (089) 9101399

FRANCE
Mitsubishi Electric Europe GmbH
55, Avenue de Colmar
92563 Rueil Malmaison Cedex
Telex: 632326
Telephone: 47087871
Facsimile: 47513622

ITALY
Mitsubishi Electric Europe GmbH
Centro Direzionale Colleoni
Palazzo Cassiopea 1
20041 Agrate Brianza I-Milano
Telephone: (039) 636011
Facsimile: (039) 6360120

SWEDEN
Mitsubishi Electric Europe GmbH
Lastbilsvägen 6B
5-19149 Sollentuna, Sweden
Telex: 10877 (meab S)
Telephone: (08) 960468
Facsimile: (08) 966877

U.K.
Mitsubishi Electric (U.K.) Ltd.
Travellers Lane
Hatfield
Herts AL10 8XB, England, U.K.
Telephone: (0044) 7072 76100
Facsimile: (0044) 7072 78692

AUSTRALIA
Mitsubishi Electric Australia Pty. Ltd.
73-75, Epping Road, North Ryde,
P.O. Box 1567, Macquarie Centre,
N.S.W., 2113, Australia
Telex: MESYD AA 26614
Telephone: (02) (888) 5777
Facsimile: (02) (887) 3635

MITSUBISHI ELECTRIC

MITSUBISHI DATA BOOK
SINGLE-CHIP 16-BIT MICROCOMPUTERS

Jun. First Edition 1991

Editioned by
Committee of editing of Mitsubishi Semiconductor Data Book

Published by
Mitsubishi Electric Corp., Semiconductor Division

MITSUBISHI SEMICONDUCTORS
SINGLE-CHIP 16-BIT MICROCOMPUTERS

Enlarged edition

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

H-D0492-A KI-9106 Printed in Japan (ROD)

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