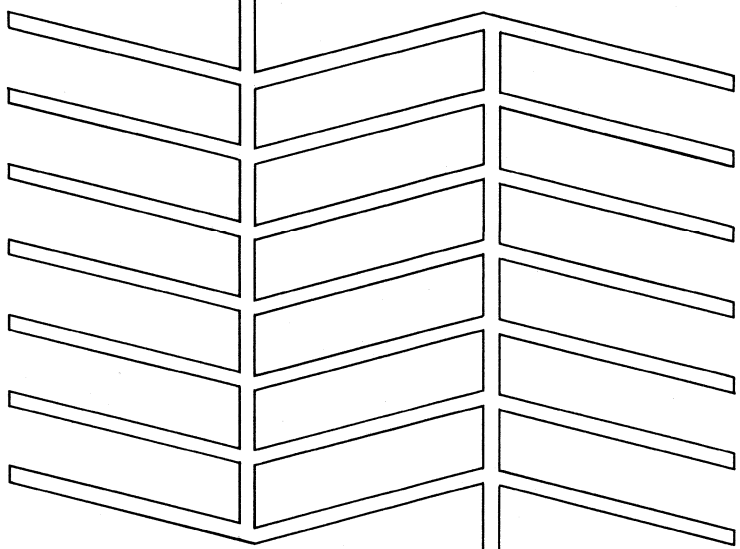
1981

MITSUBISHI LSI DATA BOOK

SUPPLEMENT TO '80 LSI 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.

MELPS and MELCS are registered trademarks of Mitsubishi Electric Corporation.

PREFACE

Thank you for your continued patronage of Mitsubishi Electric and our semiconductor products.

Semiconductor devices are a mainstay of the burgeoning electronics industry, where they are finding more and more applications, and meeting demands for increased sophistication and diversification of performance and function.

We have already published the 1980 Mitsubishi LSI Data Book, but after then a number of new products are announced, then we have prepared the supplement to the 1980 Mitsubishi LSI Data Book.

This supplement provides detailed specifications of new products, including 16K-bit static RAMs, 32K-bit EPROMs, MELPS 42 single-chip CMOS 4-bit microcomputer, MELPS 8-48 single-chip 8-bit microcomputers, MELPS 86 16-bit microprocessor, board computers and their development and support system, along with additional MOS LSI devices.

We hope you will let us know of any mistakes or omissions that come to your attention, and any suggestions you might have on improving the usefulness of this data book.

January, 1981

Kimio Sato, General Manager
Semiconductors Division
Mitsubishi Electric Corporation

CONTENTS

CONTENTS

INDEXES 1

INDEX BY FUNCTION

Type (Note 1)	Former designation	Circuit function and organization	Application notes	Structure (Note 2)	Ambient operating temp. Ta(°C)
Static RAMs					
M58725P, S-15 * **M58725P, S** *	— —	16384-Bit(2048×8) Static RAM	Power down by $\overline{CS}$	N,Si,ED	0~70
Field-Programmable ROMs (EPROMs)					
M5L2732K * **M5L2732K-6** *	— —	32768-Bit (4096×8) Erasable and Electrically Reprogrammable ROM	Electrical programming ultraviolet erasing	N,Si,FA	0~70
Single-Chip Microcomputers					
M58496-XXXP **	—	Single-Chip 4-Bit CMOS Microcomputer	77 instructions mask-prog ROM 1K-word by 10-Bit, RAM 128-word by 4-Bit	C,Al	—10~70
M5L8048-XXXP *	—	Single-Chip 8-Bit Microcomputer	96 instructions mask-prog ROM 1K-word by 8-bit RAM 64-word by 8-bit	N,Si,ED	0~70
M5L8049-XXXP *	—	Single-Chip 8-Bit Microcomputer	96 instructions mask-prog ROM 2K-word by 8-bit RAM 128-word by 8-bit	N,Si,ED	0~70
M5L8039P-6 *	—	Single-Chip 8-Bit Microcomputer	96 instructions RAM 128-word by 8-bit	N,Si,ED	0~70
M5L8748S **	—	Single-Chip 8-Bit Microcomputer with EPROM	96 instructions EPROM 1K-word by 8-bit RAM 64-word by 8-bit	N,Si,ED	0~70
Microprocessors					
M5L8086S **	—	16-Bit Microprocessor	97 instructions 1M-Byte direct access 16-bit CPU	N,Si,ED	0~70
LSIs for Peripheral Circuits					
M58990P **	—	8-Bit, 8-channel A-D Converter		C,Si	0~70
M5C6847P-1 *	—	Video Display Generator	4 different alphanumeric and 8 different graphic display modes	N,Si,ED	0~70
M5L8155P *	—	2048-Bit Static RAM with I/O Ports and Timer	256-word×8-Bit RAM with 22 I/O low pins and 14-Bits counter/timer CE=low active	N,Si,ED	0~70
M5L8156P *	—	2048-Bit Static RAM with I/O Ports and Timer	256 word×8-Bit RAM with 22 I/O pins and 14-Bit counter/timer CE=high active	N,Si,ED	0~70
M5L8259AP *	—	Programmable Interrupt Controller	8 vectored priority interrupts	N,Si,ED	0~70
M5W1791-02P **	—	Floppy Disk Formatter/Controller	Single and double density formats	N,Si,ED	0~70
General Purpose MOS LSIs					
M50110XP *	—	30-Function Remete-Control Transmitter		C, Al	—30~70
M50115XP *	—	120-Function Remote-Control Transmitter		C, Al	—30~70
M50111XP *	—	120-Function Remote-Control Receiver		C, Al	—30~70
M50116XP *	—	120-Function Remote-Control Receiver		C, Al	—30~70
M50117XP *	—	120-Function Receiver Remote-Control		C, Al	—30~70

Note 1: * =New product. ** =Under development C=CMOS. ED=Enhancement depletion mode. FA=FAMOS.
2: Al=Aluminum gate. B =Bipolar. S=Schottkey. Si =Silicon gate
N=N-channel. P =Pchannel.

MITSUBISHI ELECTRIC

INDEX BY FUNCTION

Supply voltage				Clock voltage $V_I(\phi)$	Electrical characteristics					Package (Note3)	Interchangeable products		Page
V_{DD}	V_{CC}	V_{SS} GND	V_{BB}		Typ pwr dissi-pasion (mW)	Max. access time (ns)	Max. cycle time (ns)	Max. fre-quency (MHz)	TTL com-pati-bility		Mfr.	Type	
—	5V±10%	0V	—	—	200	150	150	—	Yes	24P1 24S1	TI	TMS4016-15	4-3
					200	200	200					TMS4016	
—	5V±5%	0V	—	—	400	450	—	—	Yes	24K10	INTEL	D2732	5-3
					400	550	—					D2732-6	
—	5V±5%	0V	—	—	5	—	7700	4.2	Yes	72P2	—		6-3
—	5V±10%	0V	—	—	325	—	—	6	Yes	40P1	INTEL	P8048	7-21
—	5V±10%	0V	—	—	500	—	—	6	Yes	40P1	INTEL	P8049	7-33
—	5V±10%	0V	—	—	500	—	—	6	Yes	40P1	INTEL	P8039	7-33
—	5V±5%	0V	—	—	500	—	—	6	Yes	40S10	INTEL	C8748	7-25
—	5V±10%	0V	—	3.9V	1375	—	—	5	Yes	40S1	INTEL	C8086	8-3
—	5V±10%	0V	—	—	—	—	—	—	Yes	28P4	NS	ADC0808	9-3
—	5V±5%	0V	—	2.4V	500	—	—	3.85	Yes	40P1	MOTOROLA	MC6847-1	9-5
—	5V±5%	0V	—	—	500	—	—	—	Yes	40P1	INTEL	P8155	9-15
—	5V±5%	0V	—	—	500	—	—	—	Yes	40P1	INTEL	P8156	9-23
—	5V±10%	0V	—	—	275	—	—	—	Yes	28P4	INTEL	P8259A	9-31
—	5V±5%	0V	—	—	—	300	—	—	Yes	40P1	WESTERN DIGITAL	FD1791-02B	9-45
2.2~8V	—	0V	—	—	—	—	—	—	—	16P4	—	—	10-3
2.2~8V	—	0V	—	—	—	—	—	—	—	18P4	—	—	10-3
4.5~8V	—	0V	—	—	—	—	—	—	Yes	16P4	—	—	10-9
4.4~8V	—	0V	—	—	—	—	—	—	Yes	18P4	—	—	10-9
4.5~8V	—	0V	—	—	—	—	—	—	Yes	18P4	—	—	10-9

Note3: Package code: **24** **S** **1**

- 24 — Numder of pins
- S — Package structure
  K=Glass-sealed ceramic. P=Molded plastic. S=Metal-sealed ceramic
- 1 — Package outline
  1=DIL without fin. 2=Flat without fin.
  4=DIL without fin.(improved). 10=DIL w/quartz lid.

INDEX BY FUNCTION

Type (Note4)	Function	Application notes	Memory capacity RAM (bytes)	Memory capacity ROM (bytes)	I/O port (bits)	Ambient operating temp Ta (°C)	Supply voltage (V)	Dimensions (l×w×h) (mm)	Page
Microcomputer System									
PCA8506 *	MELCS 85/2 Memory and Parallel I/O Expansion Board	For PCA8501 PCA8540	12K(Note1)		48	0~55	5	125×145×17	11-3
PCA8507 *	MELCS 85/2 Memory and Serial I/O Expansion Board	For PCA8501,PCA8540	12K(Note1)		1 (serial)	0~55	12.5,—12	125×145×17	11-7
PCA8520G01 * **PCA8520G02** *	MELCS 85/3 Voice Generating Single-Board Computer	Using M5L8085AP	256	16K (Note2)	24	0~55	5,—5	125×145×20	11-11
PCA8540G01* **PCA8540G02***	MELCS 85/2 Color TV Display Single-Board Computer	Using M5L8085AP and M5C6847P-1	256	4K (Note3)	22	5~40	5,—5	125×145×20	11-15

Note 1: The standard product contains neither M5L2716K 2K-byte EPROMs or M58725P 2K-byte static RAMs.
2: The PCA8520G01 does not contains M5L2716K EPROMs.
The PCA8520G01 contains standard voice stored in eight M5L2716K EPROMs.
3: The standard product contains no M5L2716K EPROMs.
4: * : New Product;

Microcomputer Support Systems

PC0400	MELCS 4/1 Portable Development Support System	For M58840-XXXP	—	—	—	0~55	AC100	370×360×140	12-3

MELPS Software

Program	Program code number	Normal shipping media	Source language
MELPS 42 Cross Assembler	GB1AS0010	Magnetic tape	FORTRAN (part in assembler)
MELPS 42 Paper-Tape Generation Program for PROM Writers	GB1SP0006	Magnetic tape	FORTRAN (part in assembler)

ORDERING INFORMATION AND PACKAGE OUTLINES

2

2

FUNCTION CODE

Mitsubishi integrated circuits may be ordered using the following simplified alphanumeric type-codes which define the function of the ICs and the package style.

For Mitsubishi Original Products

Example: **M 5 89 81 S – 45**

M: Mitsubishi integrated circuit prefix

Temperature range
- **5**: Standard industrial/commercial (0 to 70/75°C or −20 to 85°C).
- **9**: High reliability (military)

Series designation using 1 or 2 alpha-numeric characters.
- **0**: CMOS
- **1**: Linear circuit
- **3**: TTL
- **10~19**: Linear circuit
- **32~33**: TTL (equivalent to Texas Instruments' SN74 series)
- **41~47**: TTL
- **84**: CMOS
- **85**: P-channel silicon-gate MOS
- **86**: P-channel aluminum-gate MOS
- **87**: N-channel silicon-gate MOS
- **88**: P-channel aluminum-gate ED-MOS
- **89**: CMOS
- **9**: DTL
- **50~52**: Schottky TTL (equivalent to Texas Instruments' SN74S series)

Circuit function identification code using 2 digits.

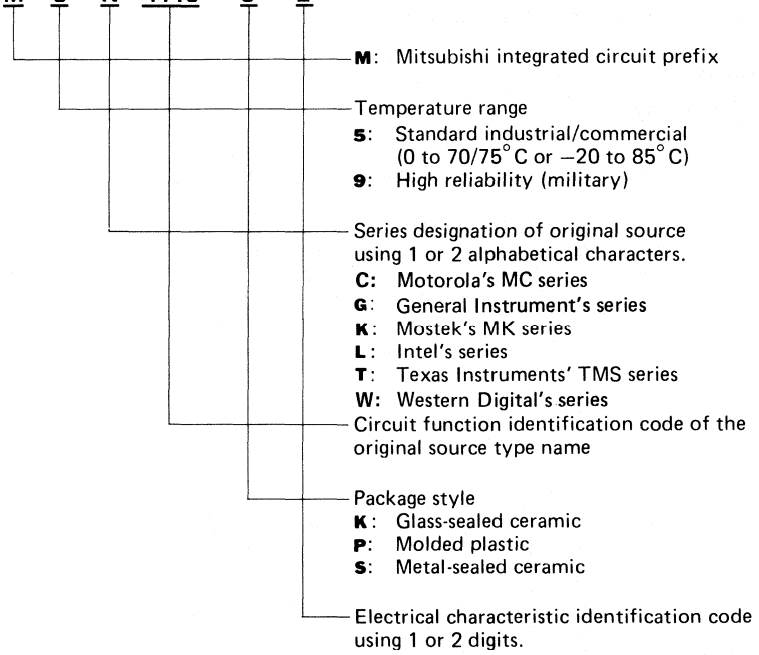
Package style
- **K**: Glass-sealed ceramic
- **P**: Molded plastic
- **S**: Metal-sealed ceramic

Electrical characteristic identification code using 1 or 2 digits.

ORDERING INFORMATION

For Second Source Products

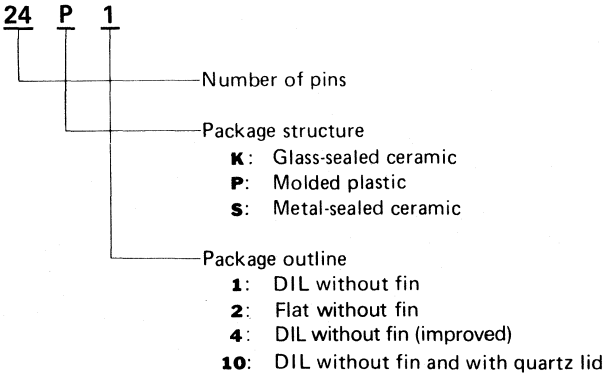
Example: M 5 K 4116 S – 2



PACKAGE CODE

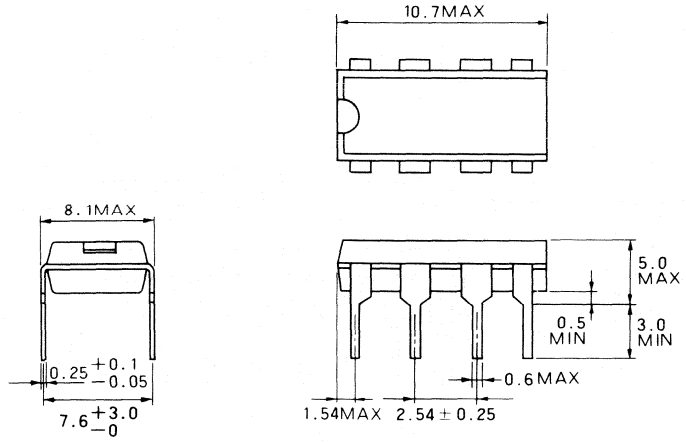
Package style may be specified by using the following simplified alphanumeric code.

Example: 24 P 1

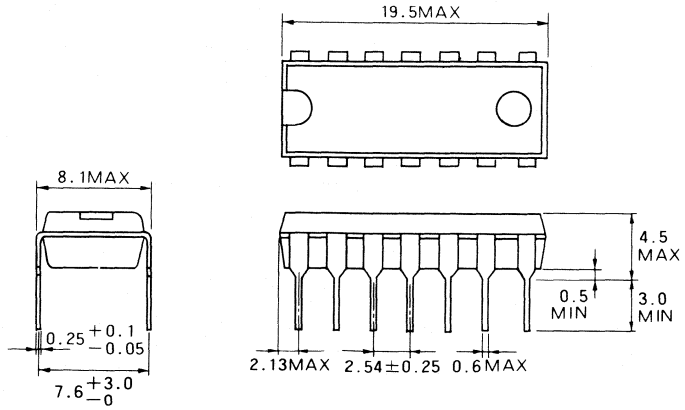


2

TYPE 8P1 8-PIN MOLDED PLASTIC DIL

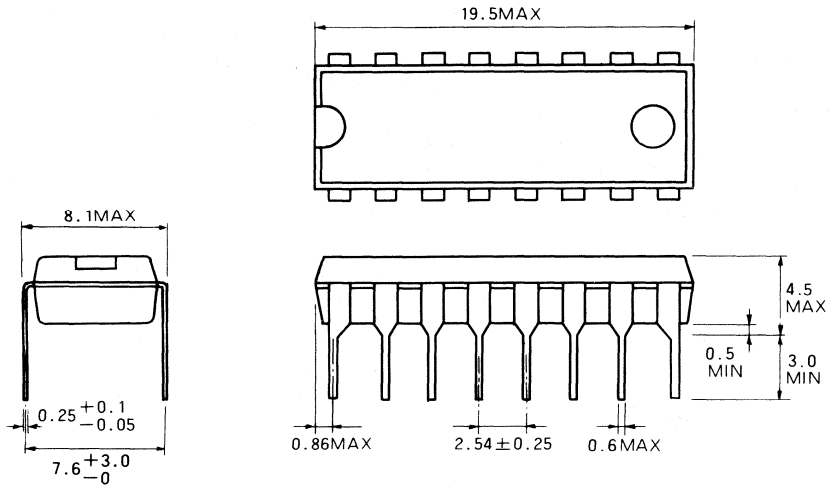


TYPE 14P4 14-PIN MOLDED PLASTIC DIL

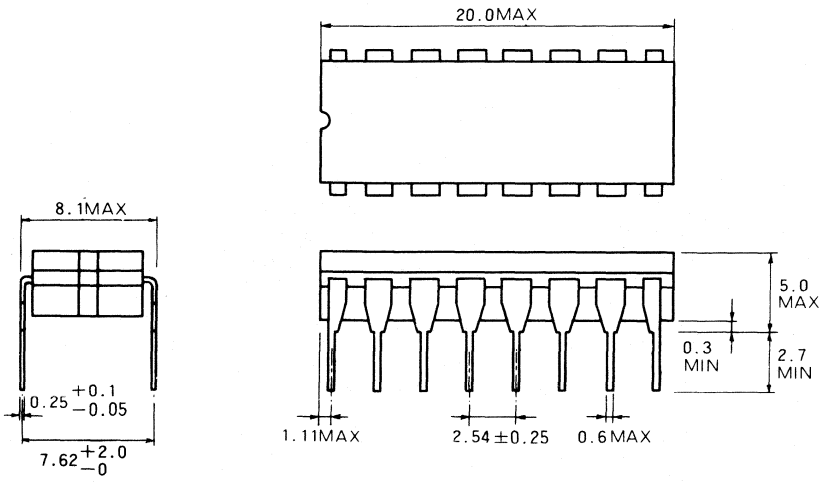


PACKAGE OUTLINES

TYPE 16P4 16-PIN MOLDED PLASTIC DIL



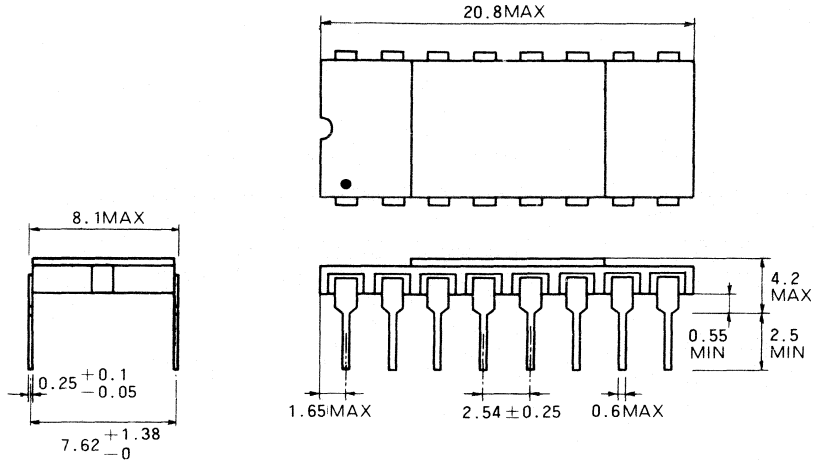
TYPE 16K1 16-PIN GLASS-SEALED CERAMIC DIL



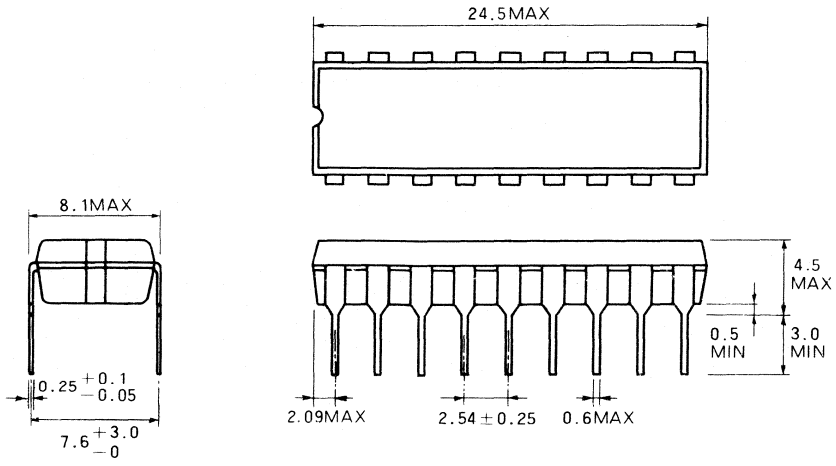
MITSUBISHI ELECTRIC

2

TYPE 16S1 16-PIN METAL-SEALED CERAMIC DIL

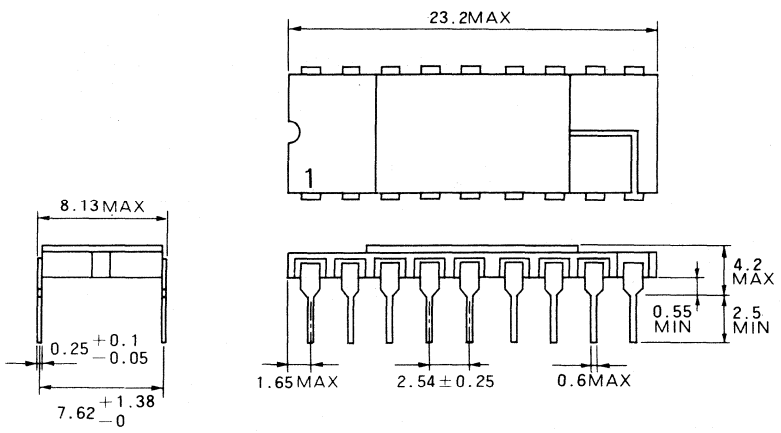


TYPE 18P4 18-PIN MOLDED PLASTIC DIL

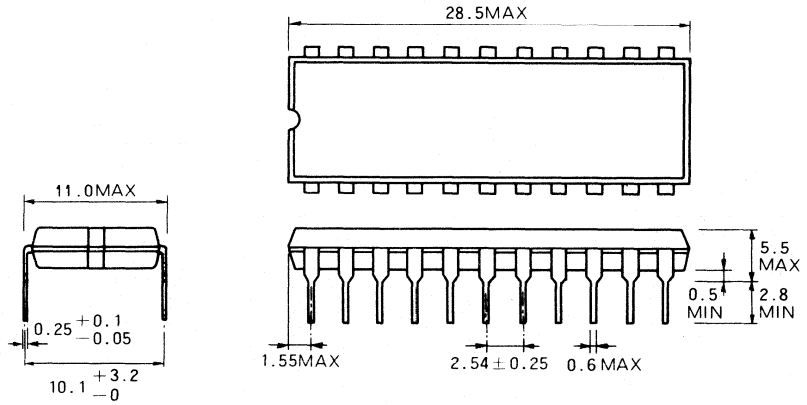


PACKAGE OUTLINES

TYPE 18S1 18-PIN METAL-SEALED CERAMIC DIL



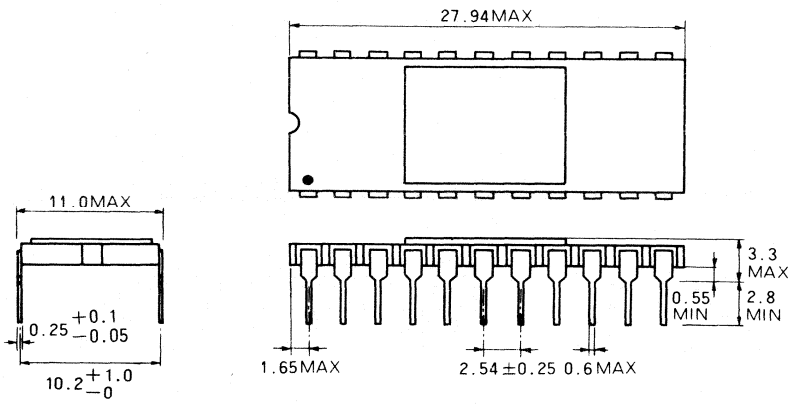
TYPE 22P1 22-PIN MOLDED PLASTIC DIL



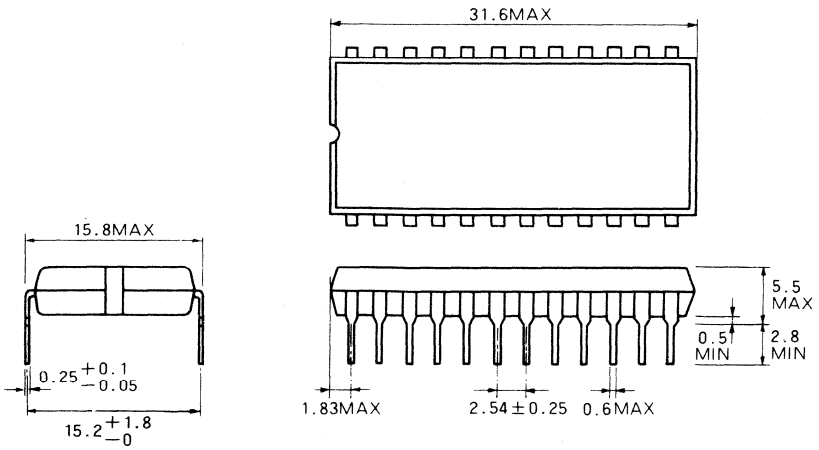
MITSUBISHI ELECTRIC

2

TYPE 22S1 22-PIN METAL-SEALED CERAMIC DIL

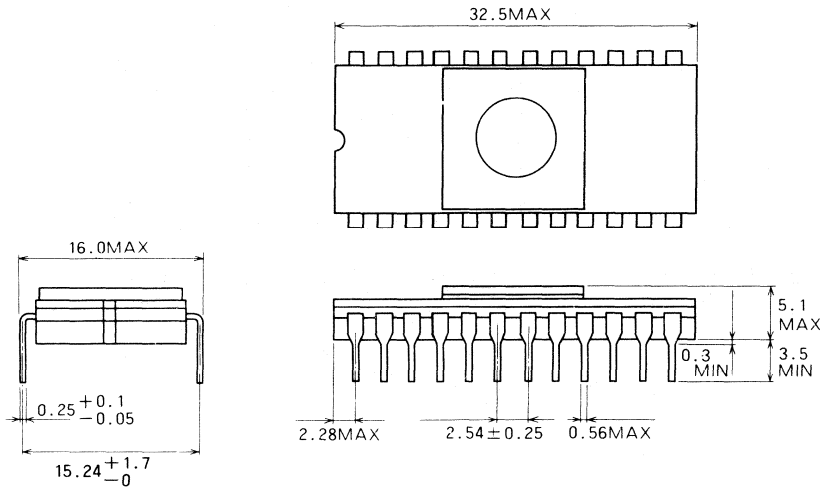


TYPE 24P1 24-PIN MOLDED PLASTIC DIL

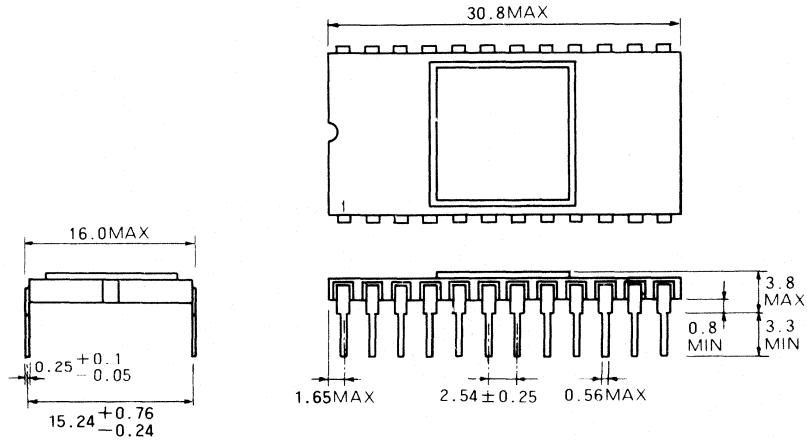


PACKAGE OUTLINES

TYPE 24K10 24-PIN METAL-SEALED CERAMIC DIL WITH QUARTZ LID



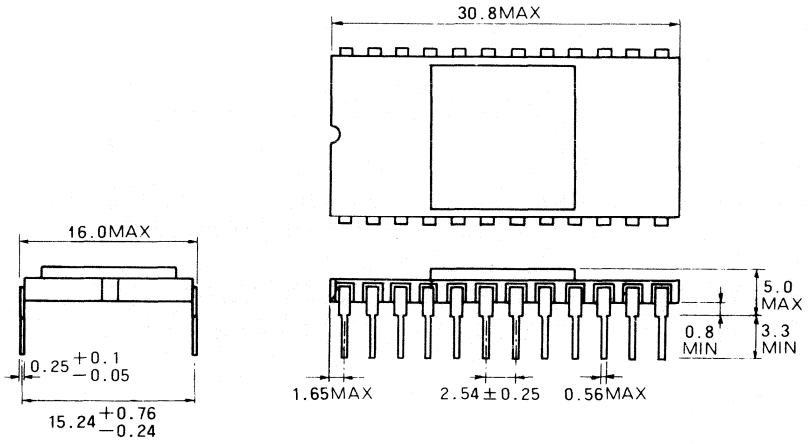
TYPE 24S1 24-PIN METAL-SEALED CERAMIC DIL



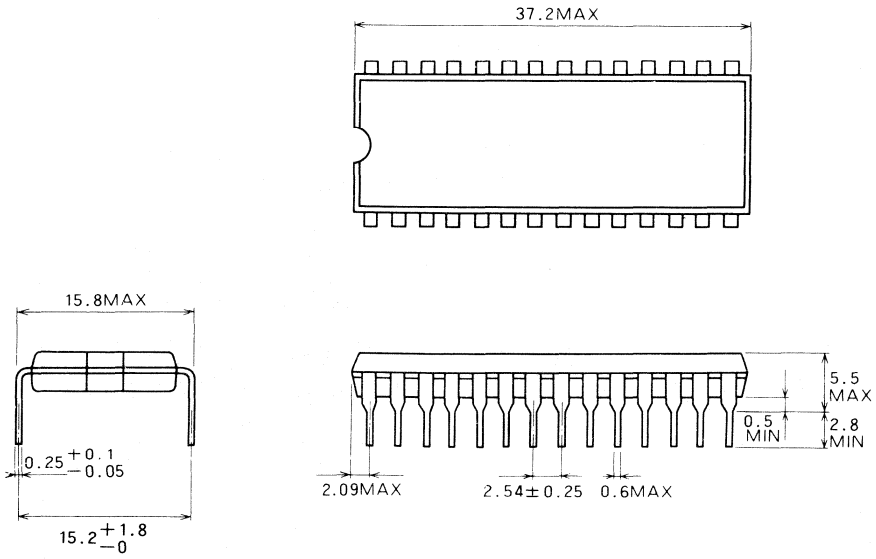
MITSUBISHI ELECTRIC

2

TYPE 24S10 24-PIN METAL-SEALED CERAMIC DIL WITH QUARTZ LID

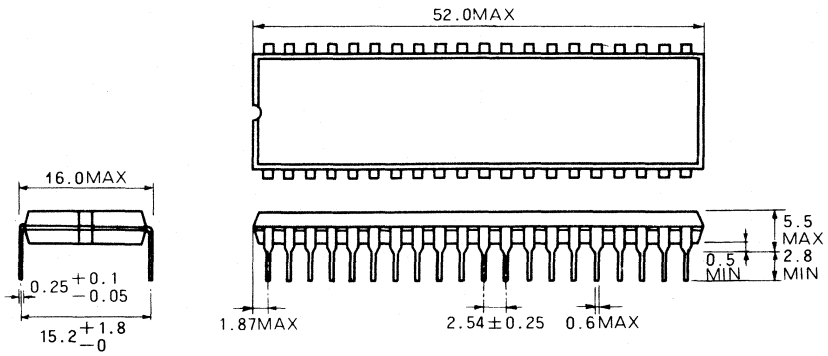


TYPE 28P4 28-PIN MOLDED PLASTIC DIL

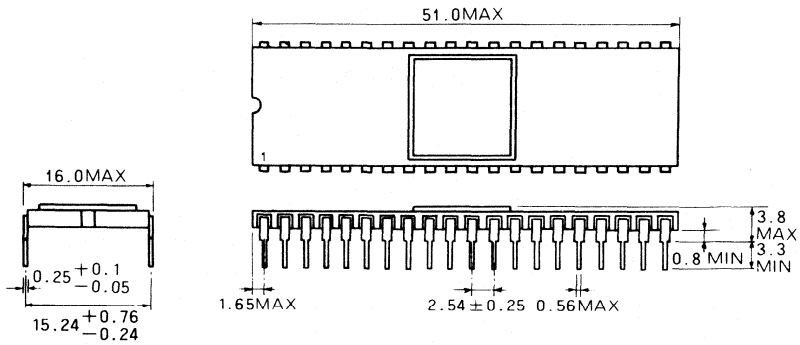


PACKAGE OUTLINES

TYPE 40P1 40-PIN MOLDED PLASTIC DIL



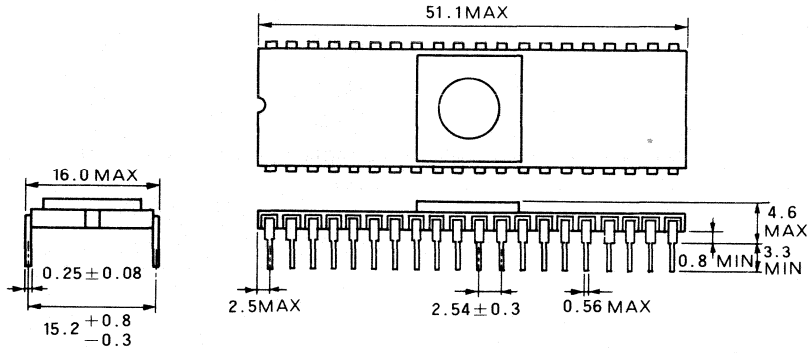
TYPE 40S1 40-PIN METAL-SEALED CERAMIC DIL



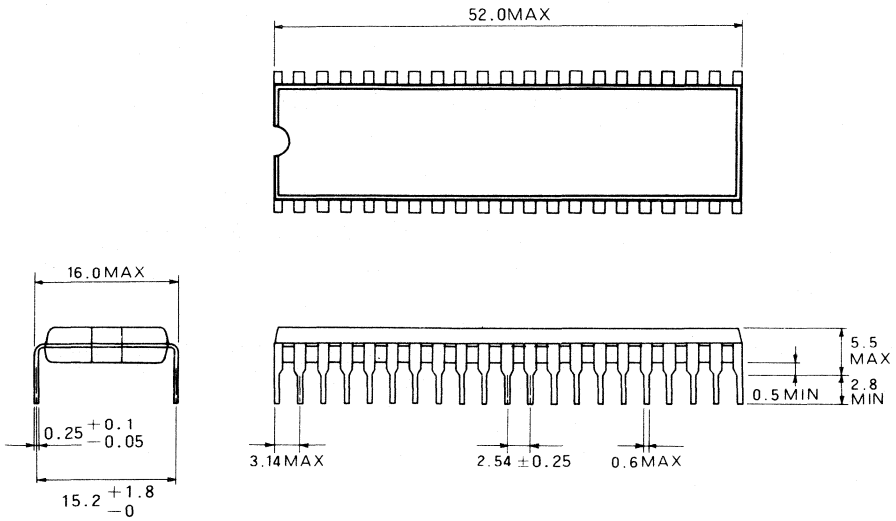
MITSUBISHI ELECTRIC

2

TYPE 40S10 40-PIN METAL-SEALED CERAMIC DIL WITH QUARTZ LID

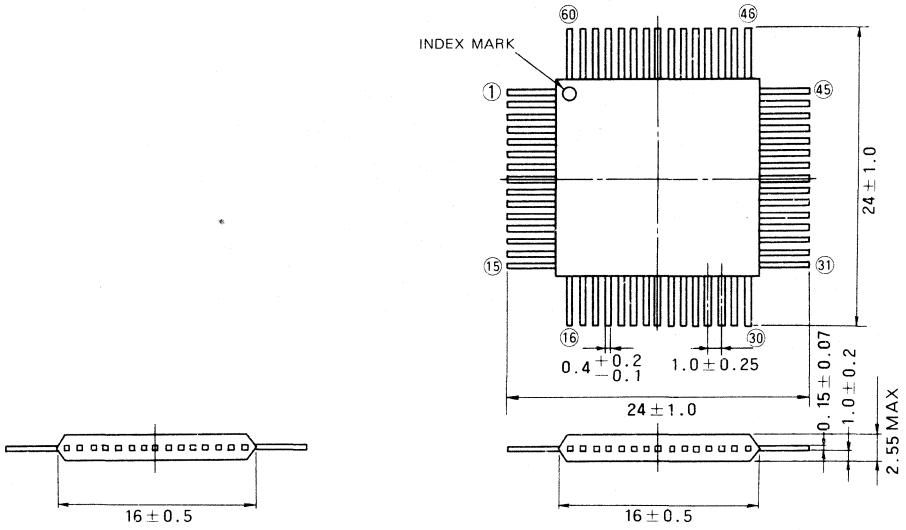


TYPE 42P1 42-PIN MOLDED PLASTIC DIL

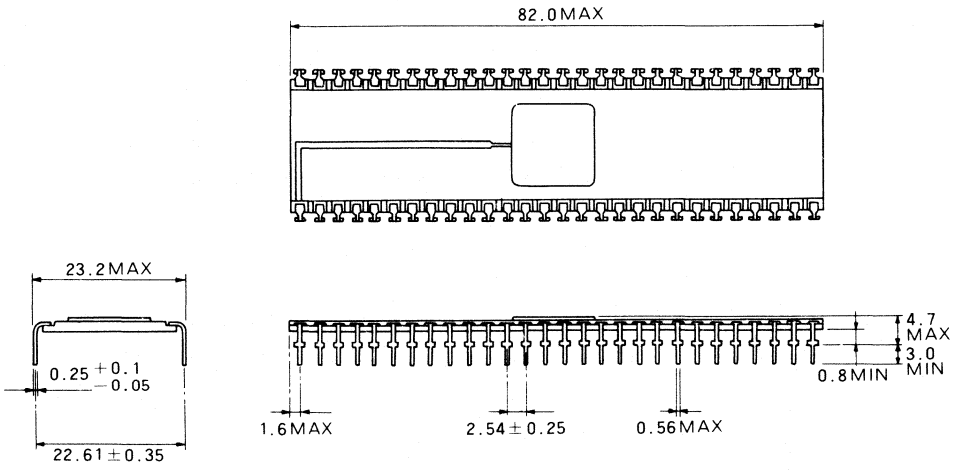


PACKAGE OUTLINES

TYPE 60P2 60-PIN MOLDED PLASTIC FLAT



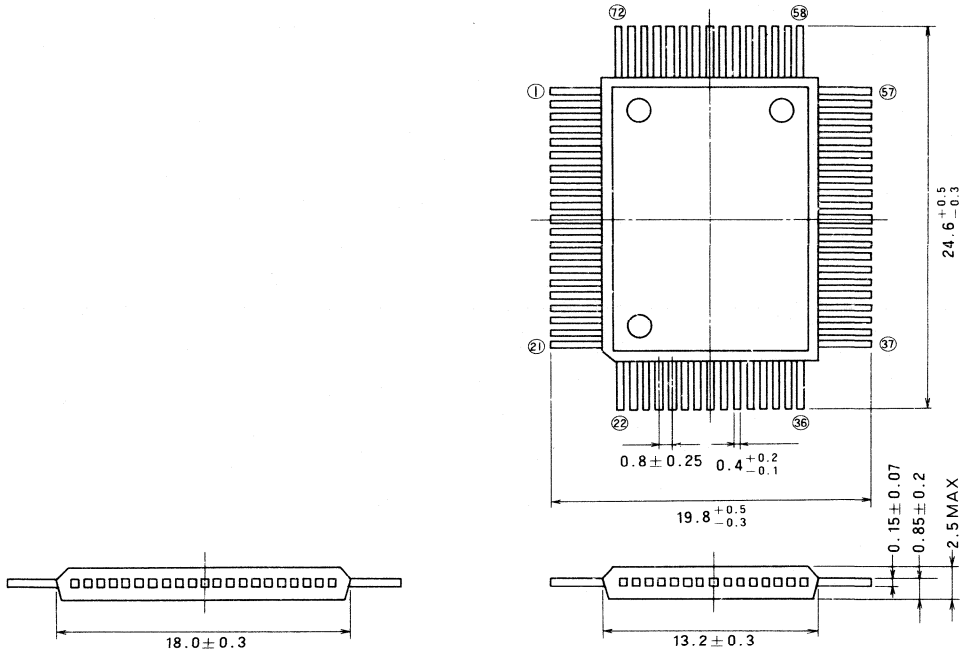
TYPE 64S1 64-PIN METAL-SEALED CERAMIC DIL



MITSUBISHI ELECTRIC

2

TYPE 72P2 72-PIN MOLDED PLASTIC FLAT



GENERAL INFORMATION

3

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 pheripheral 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 a general symbol of the form:-

$t_{A(BC-DC)F}$ ..... (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

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

SYMBOLOGY

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 Vcc
$I_{CC(AV)}$		Avarage supply current from Vcc
$I_{CC(PD)}$		Power-down supply current from Vcc
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}	$t_{C(RMR)}$	Read-modify-write cycle time—the time interval between teh 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

3

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
$t_{d(CAS\text{-}W)}$	$t_{d(CAS\text{-}WR)}$	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
$t_{dis(R\text{-}Q)}$	$t_{dis(R\text{-}DA)}$	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
t_{DLH}		Low-level to high-level delay time } output is going to the low (high) level and when the device is driven and loaded by specified 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 time during which a signal at a specified input terminal 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
$t_{h(CAS\ RAS)}$		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 output is going to the low (high) level and when the device is driven and loaded by typical devices of stated type
t_{PLH}		Low-level to high-level propagation time }
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

SYMBOLOGY

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
$t_{su(RA-CAS)}$		Column 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
$t_{w(W)}$	$t_{w(WR)}$	Wrtie pulse width
$t_{w(\phi)}$		Clock pulse width
Ta		Ambient temperature
Topr		Operating temperature
Tstg		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

RANDOM-ACCESS MEMORIES

4

M58725P, S; P-15, S-15

16 384-BIT (2048-WORD BY 8-BIT) STATIC RAM

DESCRIPTION

This is a family of 2048-word by 8-bit static RAMs, fabricated with the N-channel silicon-gate MOS process and designed for simple interfacing. These devices operate on a single 5V supply, as does TTL, and are directly TTL-compatible.

The input and output terminals are common, and an $\overline{OE}$ terminal is provided. $\overline{S}$ controls the power-down feature.

FEATURES

- Fast access time:
  - M58725P, S: 200ns (max)
  - M58725 P-15, S-15: 150ns (max)
- Low power dissipation:
  - Active: 250mW (typ)
  - Stand by: 25mW (typ)
- Power down by $\overline{S}$
- Single 5V supply voltage (±10% tolerance)
- Requires neither external clock nor refreshing
- All inputs and outputs are directly TTL compatible
- All outputs are three-state, with OR-tie capability
- Easy memory expansion by chip-select ($\overline{S}$) input
- Common data DQ terminals.
- Same pin configuration as M5L2716K 16 384-bit EPROM

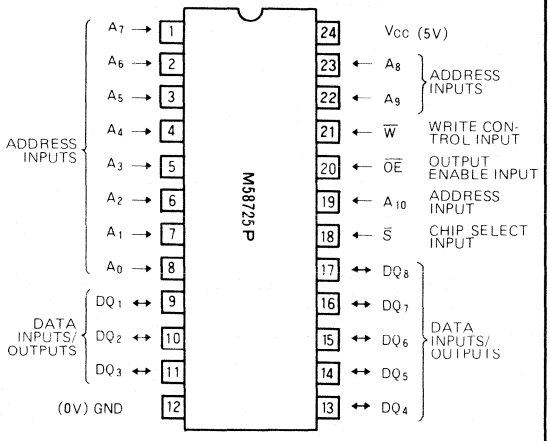
APPLICATION

- Small-capacity memory units

FUNCTION

These devices provide common data input and output terminals. During a write cycle, when a location is designated by address signals $A_0 \sim A_{10}$ the $\overline{OE}$ signal is kept high to keep the DQ terminals in the input mode, signal $\overline{W}$ goes low, and the data of the DQ signal at that time is written.

PIN CONFIGURATION (TOP VIEW)



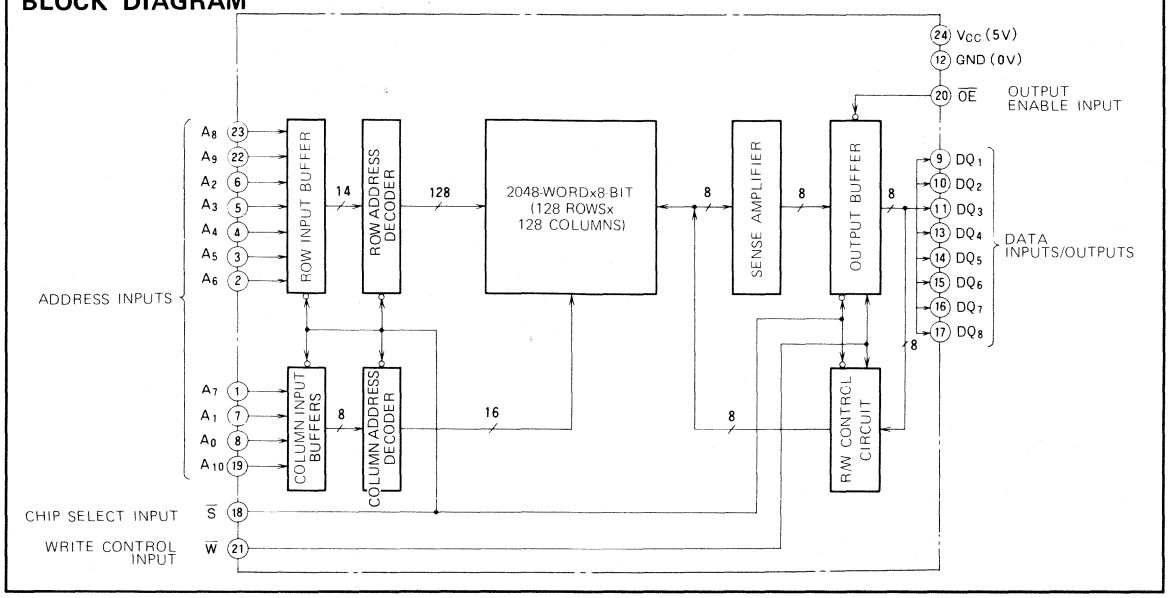
Outline 24P1 (M58725P)
24S1 (M58725S)

During a read cycle, when a location is designated by address signals $A_0 \sim A_{10}$ the $\overline{OE}$ signal is kept low to keep the DQ terminals in the output mode, signal $\overline{W}$ goes high, and the data of the designated address is available at the I/O terminals.

When signal $\overline{S}$ is high, the chip is in the non-selectable state, disabling both reading and writing. In this case the output is in the floating (high-impedance) state, useful for OR-ties with other output terminals.

Signal $\overline{S}$ controls the power down feature. When $\overline{S}$ goes high power dissipation is reduced to 1/10 of active power. The access time from $\overline{S}$ is equivalent to the address access time.

BLOCK DIAGRAM



16 384-BIT (2048-WORD BY 8-BIT) STATIC RAM

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Test conditions	Limits	Unit
V_{CC}	Supply voltage	With respect to GND	−0.5 ~ 7	V
V_I	Input voltage		−0.5 ~ 7	V
V_O	Output voltage		−0.5 ~ 7	V
P_d	Maximum power dissipation	Ta = 25°C	1000	mW
Topr	Operating free-air ambient temperature range		0 ~ 70	°C
Tstg	Storage temperature range		−65 ~ 150	°C

RECOMMENDED OPERATING CONDITIONS (Ta = 0 ~ 70°C, unless otherwise noted.)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{CC}	Supply voltage	4.5	5	5.5	V
V_{IL}	Low-level input voltage	−0.5		0.8	V
V_{IH}	High-level input voltage	2		6	V

ELECTRICAL CHARACTERISTICS (Ta = 0 ~ 70°C, V_{CC} = 5V ± 10%, unless otherwise noted.)

Symbol	Parameter	Test conditions		Limits			Unit
				Min	Typ	Max	
V_{IH}	High-level input voltage			2		6	V
V_{IL}	Low-level input voltage			−0.5		0.8	V
V_{OH}	High-level output voltage	I_{OH} = −1mA, V_{CC} = 4.5V		2.4			V
V_{OL}	Low-level output voltage	I_{OL} = 3.2mA				0.4	V
I_I	Input current	V_I = 0 ~ 5.5V				10	μA
I_{OZH}	Off-state high-level output current	$V_{I(\overline{S})}$ = 2V, V_O = 2.4V ~ V_{CC}				10	μA
I_{OZL}	Off-state low-level output current	$V_{I(\overline{S})}$ = 2V, V_O = 0.4V				−10	μA
I_{CC1}	Supply current from V_{CC}	V_I = 5.5V, $V_{I(\overline{S})}$ = 0.8V, outputs open	Ta = 25°C		50		mA
			Ta = 0°C			90	mA
I_{CC2}	Stand-by current	V_I = 5.5V, $V_{I(\overline{S})}$ = 2V, outputs open	Ta = 25°C		5	10	mA
			Ta = 70°C		7	15	mA
C_i	Input capacitance, all inputs	V_I = GND, Vi = 25mVrms, f = 1MHz			3	5	pF
C_o	Output capacitance	V_O = GND, V_O = 25mVrms, f = 1MHz			5	8	pF

Note 1: Current flowing into an IC is positive, out is negative.

SWITCHING CHARACTERISTICS (For Read Cycle) (Ta = 0 ~ 70°C, V_{CC} = 5V ± 10%, unless otherwise noted.)

Symbol	Parameter	M58725P-15, S-15			M58725P, S			Unit
		Limits			Limits			
		Min	Typ	Max	Min	Typ	Max	
$t_{C(R)}$	Read cycle time	150			200			ns
$t_{a(A)}$	Address access time			150			200	ns
$t_{a(S)}$	Chip select access time			150			200	ns
$t_{a(OE)}$	Output enable access time			50			60	ns
$t_{V(A)}$	Data valid time after address	20			20			ns
$t_{PXZ(S)}$	Output disable time after chip select			50			60	ns
$t_{PZX(S)}$	Output active time after chip select	10			20			ns
t_{PU}	Power up time after chip selection	0			0			ns
t_{PD}	Power down time after chip deselection			60			80	ns

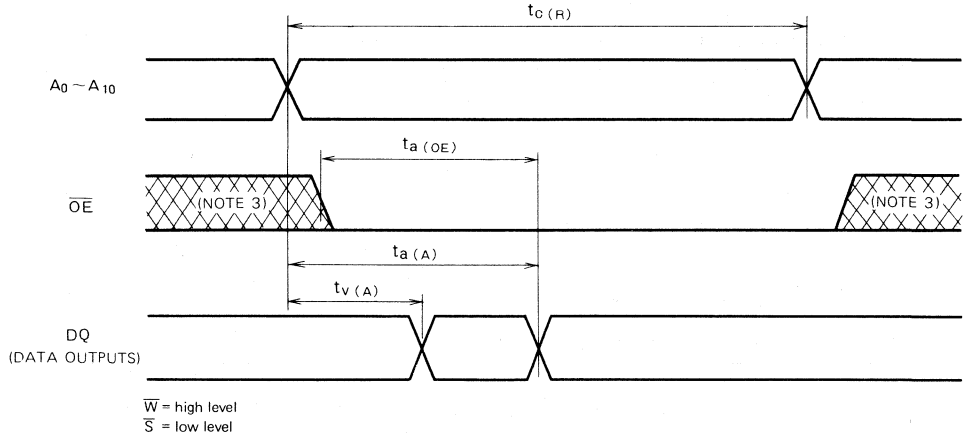
TIMING REQUIREMENTS (For Write Cycle) ($Ta = 0 \sim 70°C$, $V_{CC} = 5V \pm 10\%$, unless otherwise noted.)

Symbol	Parameter	M58725P-15, S-15 Limits			M58725P, S Limits			Unit
		Min	Typ	Max	Min	Typ	Max	
$t_{c(W)}$	Write cycle time	150			200			ns
$t_{su(S)}$	Chip select setup time	100			120			ns
$t_{su(A)}$	Address setup time	20			20			ns
$t_{w(W)}$	Write pulse width	80			100			ns
t_{wr}	Write recovery time	0			0			ns
$t_{su(OE)}$	Output enable setup time	40			40			ns
$t_{su(D)}$	Data setup time	60			60			ns
$t_{h(D)}$	Data hold time	10			10			ns
$t_{PXZ(OE)}$	Output disable time after output enable			40			40	ns
$t_{PXZ(W)}$	Output disable time after write enable			40			40	ns

4

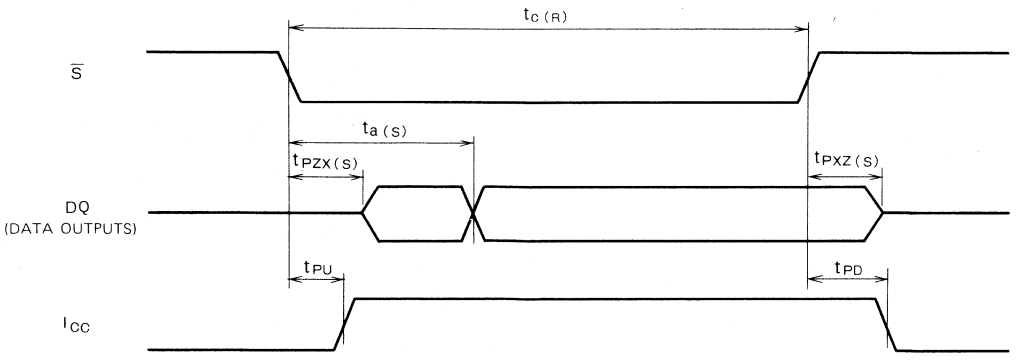
TIMING DIAGRAMS (Note 2)

Read Cycle 1



$\overline{W}$ = high level
$\overline{S}$ = low level

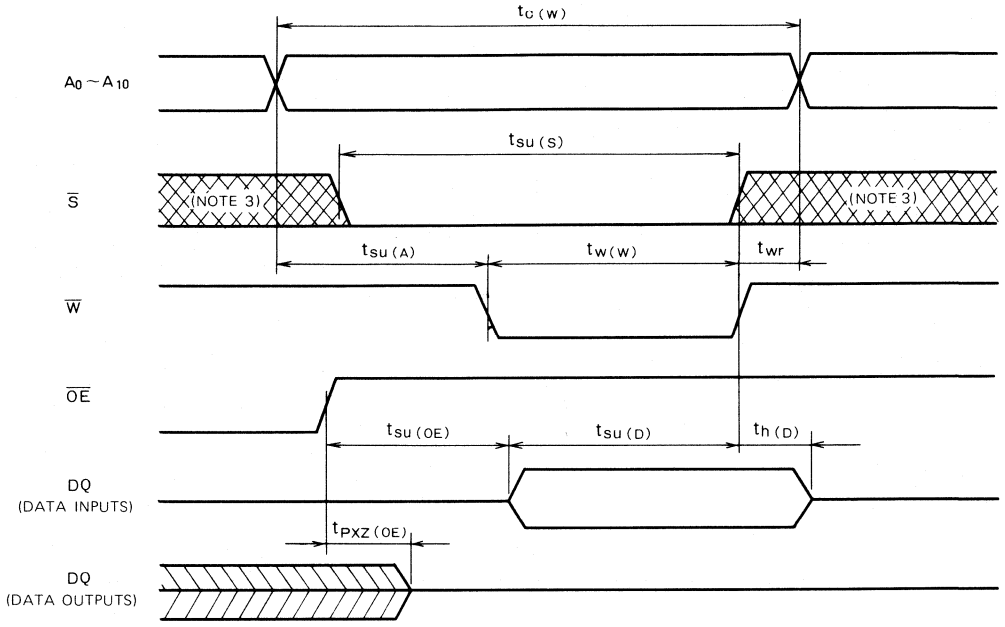
Read Cycle 2



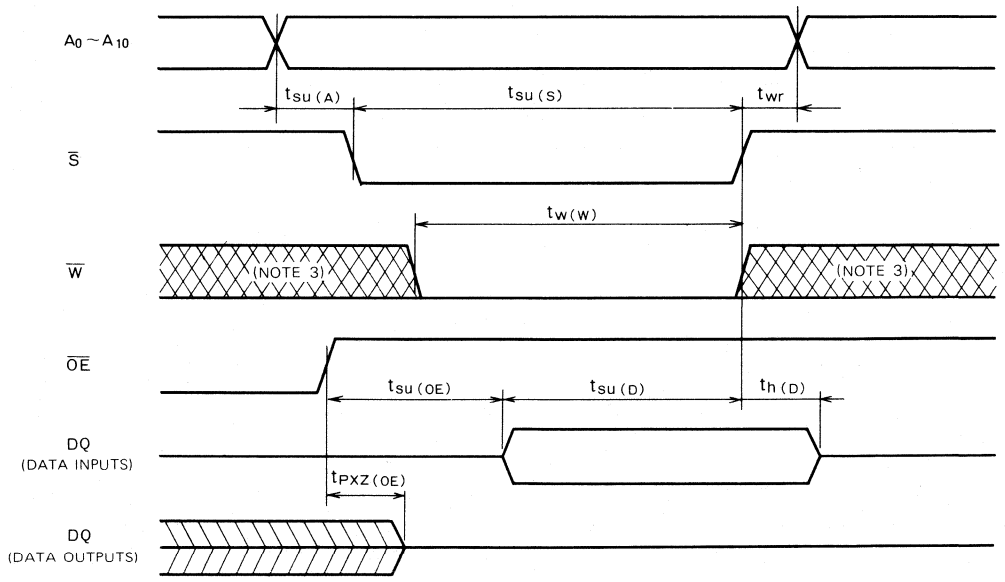
$\overline{W}$ = high level
OE = low level

16 384-BIT (2048-WORD BY 8-BIT) STATIC RAM

Write Cycle ($\overline{W}$ Control Mode)



Write Cycle 2 ($\overline{S}$ Control Mode)



Note 2: Test conditions

Input pulse level	0.8~2V
Input pulse rise time	10ns
Input pulse fall time	10ns
Reference level	
Input	1.5V
Output	1.5V

Load = 1TTL, C_L = 100pF

Note 3: Hatching indicates the state is don't care.

MITSUBISHI ELECTRIC

READ-ONLY MEMORIES

5

M5L 2732K, K-6

32 768-BIT(4096-WORD BY 8-BIT) ERASABLE AND ELECTRICALLY REPROGRAMMABLE ROM

DESCRIPTION

These are ultraviolet-light erasable and electrically reprogrammable 32 768-bit (4096-word by 8-bit) EPROMS. They incorporate N-channel silicon-gate MOS technology, and are designed for microprocessor programming applications.

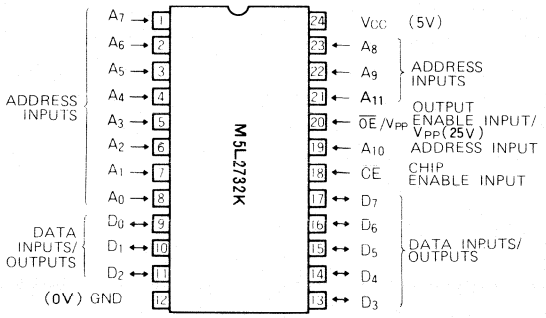
FEATURES

- Fast programming: 200s/32 768 bits (typ)
- Access time M5L 2732K: 450ns (max)
  M5L 2732K-6: 550ns (max)
- Static circuits are used throughout
- Inputs and outputs TTL-compatible in read and program modes
- Single 5V power supply for read mode
  (25V power supply required for program)
- Low power dissipation: Operating: 787mW (max)
  Standby: 157mW (max)
- Single-location programming
  (requires one 50ms pulse/address)
- Interchangeable with Intel's 2732 in pin configuration

APPLICATION

- Computers and peripheral equipment

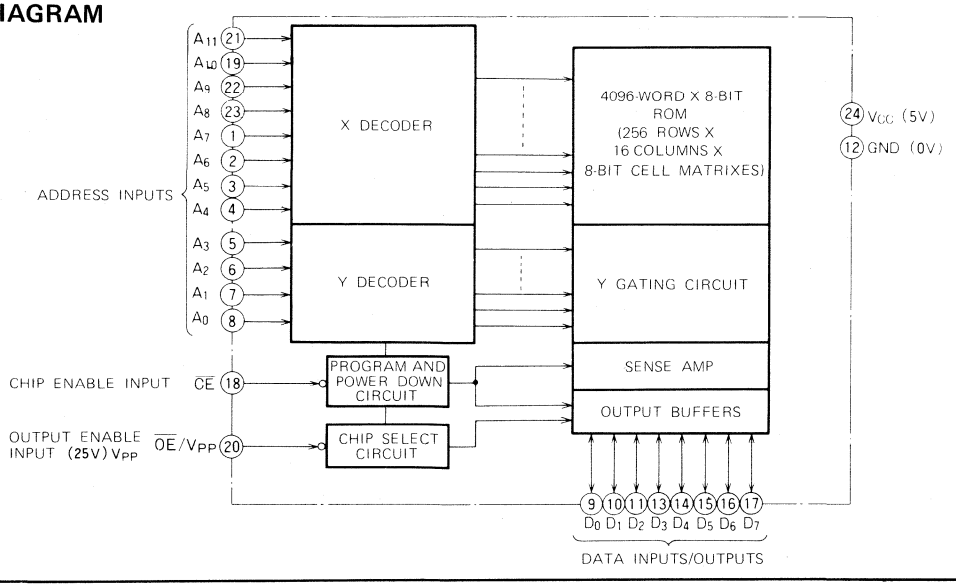
PIN CONFIGURATION (TOP VIEW)



Outline 24K10

5

BLOCK DIAGRAM



M5L 2732K, K-6

32 768-BIT(4096-WORD BY 8-BIT) ERASABLE AND ELECTRICALLY REPROGRAMMABLE ROM

FUNCTION

Read

Set the $\overline{CE}$ and $\overline{OE}$ terminals to the read mode (low-level). Low-level input to $\overline{CE}$ and $\overline{OE}$ and address signals to the address inputs ($A_0 \sim A_{11}$) make the data contents of the designated address location available at the data inputs/outputs ($D_0 \sim D_7$). When the $\overline{CE}$ or $\overline{OE}$ signal is high, data inputs/outputs ($D_0 \sim D_7$) are in a floating state.

When the $\overline{CE}$ signal is high, the device is in the standby mode or power-down mode.

Programming

The chip enters the programming mode when 25V is supplied to the $\overline{OE}/V_{PP}$ input. A location is designated by address signals $A_0 \sim A_{11}$, and the data to be programmed must be applied at 8 bits in parallel to the data inputs $D_0 \sim D_7$. A program pulse, an active low pulse, to the $\overline{CE}$ at this state will effect the programming operation. Only one programming is required, but its width must satisfy the condition $45ms \leqq t_{W(CE)} \leqq 55ms$.

Erase

Erase is effected by exposure to ultraviolet light with a wavelength of 2537Å at an intensity of approximately $15Ws/cm^2$.

Mode selection

(Unit: V)

Mode \ Pin	$\overline{CE}$	$\overline{OE}/V_{PP}$	V_{CC}	
Read	V_{IL}	V_{IL}	5	Output
Deselect	$V_{IL} \sim V_{IH}$	V_{IH}	5	Floating
Power down	V_{IH}	$V_{IL} \sim V_{IL}$	5	Floating
Program	Pulsed V_{IH} to V_{IL}	25	5	Input
Program verify	V_{IL}	V_{IL}	5	Output
Program inhibit	V_{IH}	25	5	Floating

HANDLING PRECAUTIONS

1. Sunlight and fluorescent light may contain ultraviolet light sufficient to erase the programmed information. For any operation in the read mode, the transparent window should be covered with opaque tape.
2. High voltages are used when programming, and the conditions under which is it performed must be carefully controlled to prevent the application of excessively high voltages. Specifically, the voltage applied to V_{PP} should be kept below 26V including overshoot. Special precautions should be taken at the time of power-on.
3. Before erasing, clean the surface of the transparent lid to remove completely oily impurities or paste, which may impede irradiation and affect the erasing characteristics.

32 768-BIT(4096-WORD BY 8-BIT) ERASABLE AND ELECTRICALLY REPROGRAMMABLE ROM

ABSOLUTE MAXIMUM RATING

Symbol	Parameter	Conditions	Limits	Unit
V_{I1}	Input voltage, $\overline{OE}/V_{PP}$ input	With respect to GND	−0.3 ~ 26.5	V
V_{I2}	Input voltage, V_{CC}, address, $\overline{CE}$, data inputs		−0.3 ~ 6	V
T_{opr}	Operating free air temperature range		0 ~ 70	℃
T_{stg}	Storage temperature range		−65 ~ 125	℃

READ OPERATION

Recommended Operating Conditions (Ta = 0 ~ 70℃, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{CC}	Supply voltage	4.75	5	5.25	V
GND	Supply voltage		0		V
V_{IL}	Low-level input voltage	−0.1		0.8	V
V_{IH}	High-level input voltage	2.2		V_{CC} + 1	V

Electrical Characteristics (Ta = 0 ~ 70℃, V_{CC} = 5V + 5%, unless otherwise noted.)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ (Note 1)	Max	
I_{IL1}	High-level input current, address, $\overline{CE}$ input	V_I = 5.25V			10	μA
I_{IL2}	High-level input current, $\overline{OE}/V_{PP}$ input	V_I = 4.75V			10	μA
I_{OZ}	Off-state output current	V_O = 5.25V, $\overline{OE}$ = 5 V			10	μA
I_{CC1}	Supply current from V_{CC} (standby)	$\overline{CE} = V_{IH}$, $\overline{OE} = V_{IL}$		15	30	mA
I_{CC2}	Supply current from V_{CC} (operating)	$\overline{OE} = \overline{CE} = V_{IL}$		85	150	mA
V_{OL}	Low-level output voltage	I_{OL} = 2.1mA			0.45	V
V_{OH}	High-level output voltage	I_{OH} = −400μA	2.4			V

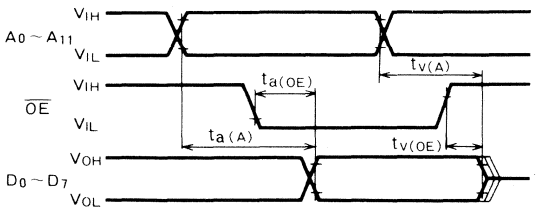
Switching Characteristics (Ta = 0 ~ 70℃, V_{CC} = 5 V ± 5 %, unless otherwise noted.)

Symbol	Parameter		Test conditions		Limits			Unit
					Min	Typ (Note 1)	Max	
$t_{a(A)}$	Address access time	M5L 2732K	$\overline{OE} = \overline{CE} = V_{IL}$	$t_r \leqq 20$ns			450	ns
		M5L 2732K−6		$t_f \leqq 20$ns			550	ns
$t_{a(CE)}$	Chip enable access time	M5L 2732K	$\overline{OE} = V_{IL}$	V_{IL} = 0.8V			450	ns
		M5L 2732K−6		V_{IH} = 2.2V			550	ns
$t_{a(OE)}$	Output enable access time	M5L 2732K	$\overline{CE} = V_{IL}$	Load:		100	150	ns
		M5L 2732K−6		100pF+1TTL			200	ns
$t_{v(OE)}$	Data valid time after output enable		$\overline{OE} = V_{IL}$		0		100	ns
$t_{v(CE)}$	Data valid time after chip select		$\overline{CE} = V_{IL}$		0		100	ns
$t_{v(A)}$	Data valid time after address		$\overline{OE} = \overline{CE} = V_{IL}$		0			ns

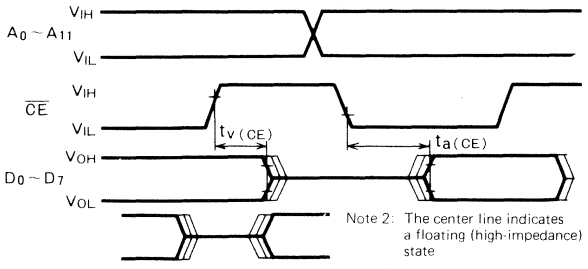
Note 1: at Ta = 25°C and normal supply voltage.

TIMING DIAGRAMS (Read Operation)

When power-Down Mode Not Used



Power-Down Mode



Note 2: The center line indicates a floating (high-impedance) state

32 768-BIT(4096-WORD BY 8-BIT) ERASABLE AND ELECTRICALLY REPROGRAMMABLE ROM

PROGRAM MODE

Recommended Operating Conditions (Ta = 25 ± 5 °C, unless otherwise noted.)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{CC}	Supply voltage	4.75	5	5.25	V
V_{PP}	Supply voltage	24	25	26	V
GNG	Supply voltage		0		V
V_{IL}	Low-level input voltage	−0.1		0.8	V
V_{IH}	High-level input voltage	2.2		V_{CC} + 1	V

Electrical Characteristics (Ta = 25 ± 5 °C, V_{CC} = 5 V ± 5 %, V_{PP} = 25 ± 1 V, unless otherwise noted.)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
I_{IL}	High-level input current, address, $\overline{CE}$ inputs	V_{IN} = 5.25V			10	μA
I_{PP}	Supply current from V_{PP}	$\overline{CE}$ = V_{IL}			30	mA
I_{CC}	Supply current from V_{CC}				150	mA

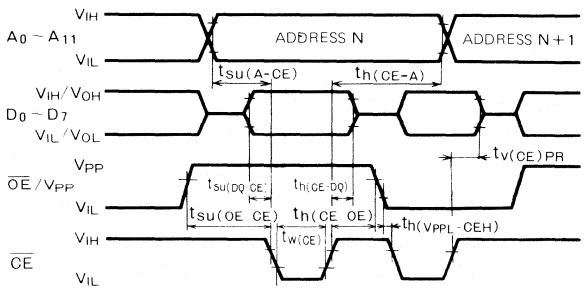
Timing Requirements (Ta = 25 ± 5 °C, V_{CC} = 5 V ± 5 %, V_{PP} = 25 ± 1 V, unless otherwise noted.)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
$t_{su(A\ CE)}$	Address setup time before chip enable		2			μs
$t_{su(OE\ CE)}$	Output enable setup time before chip enable		2			μs
$t_{su(DQ\ CE)}$	Data input setup time before chip enable		2			μs
$t_{h(CE\ A)}$	Address hold time after chip enable		2			μs
$t_{h(CE\ OE)}$	Output enable hold time after chip enable		2			μs
$t_{h(CE\ DQ)}$	Data input hold time after chip enable		2			μs
$t_{h(V_{PP}L\text{-}CEH)}$	Chip enable high hold time after V_{PP} low		2			μs
$t_{w(CE)}$	Chip enable pulse width		45	50	55	ms

Switching Characteristics (Ta = 25 ± 5 °C, V_{CC} = 5 V ± 5 %, V_{PP} = 25 ± 1 V, unless otherwise noted.)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
$t_{v(CE)PR}$	Data valid time after chip enable in program mode		0		120	ns

Timing Diagram (for Program and Verify)



MITSUBISHI ELECTRIC

MELPS 42 MICROCOMPUTERS

6

SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

DESCRIPTION

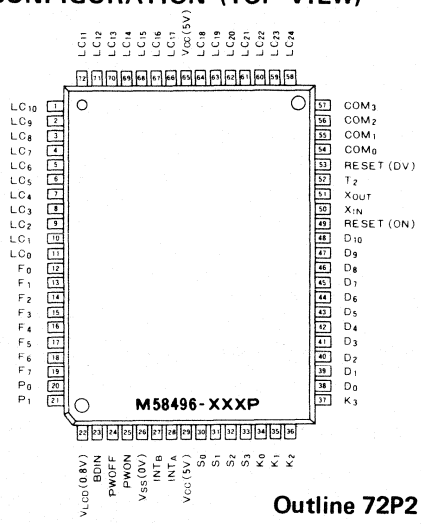
The M58496-XXXP is a single-chip 4-bit microcomputer fabricated using CMOS technology. Its features are liquid crystal display direct drive circuit, current saving circuit for back-up of a 22-stage frequency divider and RAM.

This device is designed for applications in which clock and liquid crystal display functions are included and where the low-power dissipation achieved by CMOS is especially important.

FEATURES

- Single 5V power supply
- Basic machine instructions 77
- Basic instruction execution time (at 4.2Mhz) 7.7μs
- Memory capacity: ROM: 2048 words x 10 bits
  Internal RAM: 128 words x 4 bits
  External RAM: 256 words x 4 bits
- Internal crystal oscillation circuit
- Internal 22-stage frequency divider
- Low voltage detector circuit
- Internal current saving circuit while idling
- Subroutine nesting . . . . . 3 levels
- Internal timer: Prescaler: 7 bits
  Timer: 4 bits
- Output ports for liquid crystal display
  segment signal (port LC): 25 bits
  common signal (port COM): 4 bits
- I/O Ports (ports K and S) . . . . . 2x4 bits
- Output port (port D) . . . . . 11x1 bit

PIN CONFIGURATION (TOP VIEW)



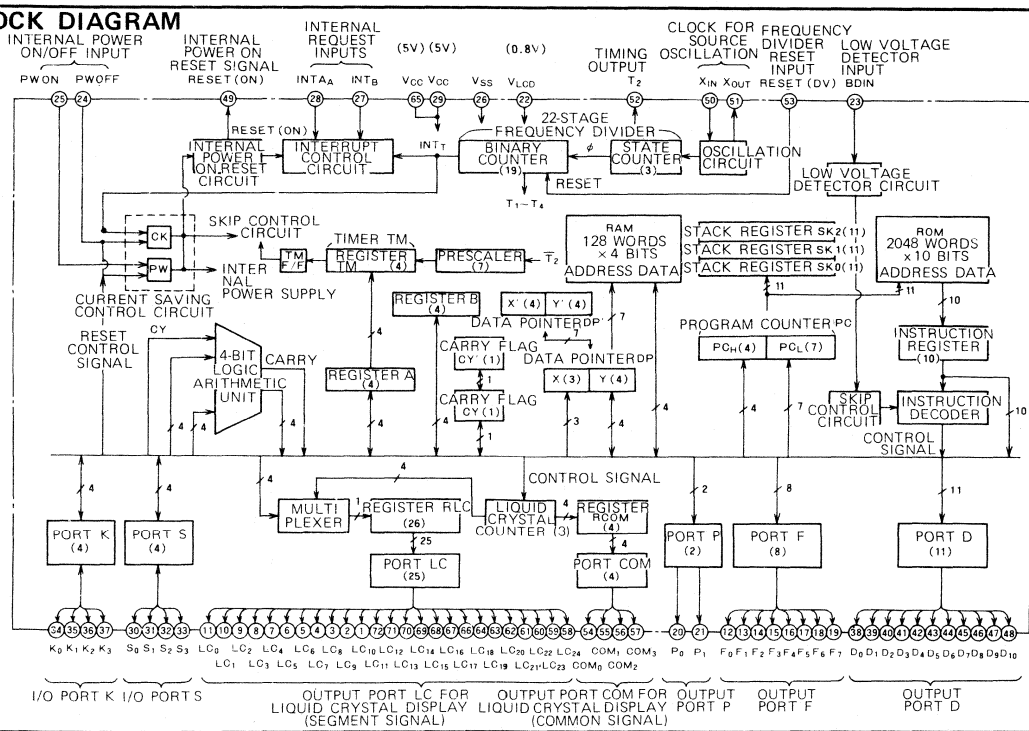
Outline 72P2

- Input port (port F) . . . . . 8x1 bit
- Output port (port P) . . . . . 2x1 bit
- Interrupt function . . . . . 4 factors, 1 level

APPLICATIONS

- Electronic cash registers and calculators with printer
- Office machines, intelligent terminals and data terminals
- Electronic Games
- Electronic coin and changer machines
- Sewing machines

BLOCK DIAGRAM



SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

FUNCTION

The M58496-XXXP consists of mask ROM and RAM, a 4-bit arithmetic logic unit, crystal oscillation circuit, 22-stage frequency divider, power saving circuit, low voltage detector circuit, 4-bit timer, interrupt circuit and a liquid crystal display direct drive circuit. The RAM capacity can easily be expanded by the external connection of 256-word by 4-bit CMOS RAM.

The ROM storage is organized as 16 pages of 128 words which is used mainly for programs. Addressing the ROM is done through the program counter. The address register is structured as a 7-bit address register and a 4-bit page register. The address register is counted up as nonbranching instructions are executed. When a nonbranching instruction at address 127 on a page is executed an overflow of the address register is produced. This carry (overflow) is disregarded so the page register is not counted up and the next instruction to be executed will come from address 0 on the same page.

The return addresses for subroutines and interrupts are saved in one of the 11-bit stack registers. The stack consists of 3 stack registers (SK2, SK1, SK0) and when interrupts are expected 1 level should be reserved for interrupt processing. When an interrupt request is accepted control is transferred to fixed addresses as follows: in case of an internal power on reset signal (RESET (ON)) the program is set to page 0 address 0, for the INT_A signal it is set to page 0 address 2, for the INT_B signal it is set to page 0 address 4 and for the output signal INT_T (second signal) of the 22-stage frequency divider it is set to page 0 address 8.

The internal RAM which is configured as 8 files of 16 words is used for data storage and each word can be addressed. The internal RAM is addressed by a 7-bit data pointer. The internal RAM can be augmented by external RAM consisting of up to 16 files of 16 words. The external RAM is addressed by the 8-bit combined register Y (4 bits) and register B (4 bits).

RAM addressing, register-to-register transfers, RAM-to-accumulator transfers, arithmetic operations, input/output operations and timer operation are performed mainly through register A (accumulator).

The current saving circuit used in conjunction with the 22-stage frequency divider and RAM can be controlled by the PWOFF input and instruction.

The low voltage detector circuit is also active while the power source is a battery. Low voltage is sensed by the program and an indication can be output.

The output ports for direct drive of the liquid crystal display are port LC (25 terminals) and port COM (4 terminals). The liquid crystal display can be driven by 1/4 duty, 1/3 bias or 1/3 duty, 1/3 bias.

Output port D consists of 11 individually latched bits that can be used to output not only 1-bit data but can also output data such as the contents of register Y of the data pointer and 8-bit addresses for external RAM.

Output port F consists of 8 individually latched bits that can be used to output data. It can be set or reset by instructions.

Output port P consists of 2 terminals through which a synchronous signal of 1 machine cycle width can be output by instruction.

The combined 7-bit output of ports F and P can be used to directly fetch the contents of ROM addressed by the data field of an instruction.

The I/O ports K and S consist of 4 terminals through which data can be transferred to and from register A.

PERFORMANCE SPECIFICATIONS

Item			Performance
Number of basic instructions			77
Execution time of basic instructions			7.7μs (V_{CC}=5V, f=4.1943MHz)
Clock frequency			250~525kHz
Memory Capacity	ROM		2048 words x 10 bits
	Internal RAM		128 words x 4 bits
	External RAM		256 words x 4 bits
I/O Port	LC	Liquid crystal display output	25 x 1 bit (Note 1)
	COM		4 terminals x 4 bits
	K	Input	4 bits
		Output	4 bits (Note 2)
	S	Input	4 bits
		Output	4 bits (Note 2)
	D	Output	11x 1 bit (open drain)
	F	Output	8 x 1 bit (Note 2)
	P	Output	2 x 1 bit (Note 2)
Frequency divider			22-stage built in
Current saving circuit			Built in
Low voltage detector			Built in
Subroutine nesting			3 levels (including 1 level of interrupt)
Interrupt request			4 factors, 1 level
Clock generation circuit			Built in (4.1943 MHz crystal oscillator external)
Input/output port	Output voltage		6V (max)
	Output current		-0.5 mA (max)
Power supply voltage:	V_{CC}		5V (nom)
	V_{SS}		0V
Liquid crystal display driving supply voltage			0.8V (nom)
Element structure			CMOS
Package			72-pin plastic molded flat package
Power dissipation (open output terminals)	In operation		5mW (V_{CC}=5V, 525 kHz)
	In idle		1.5mW (V_{CC}=5V, 525 kHz)

Note 1: Port LC can be extended to a maximum of 26 bits, but other ports are eliminated.

2: Ports K, S, F, and P are connected to high-impedance pull-down resistors. When high driving current is required, external resistors are required.

MITSUBISHI ELECTRIC

PIN DESCRIPTION

Pin	Name	Input or output	At reset (internal power-on)	Function
X_{IN}	Source oscillation clock input	Input	—	Incorporates the clock oscillation circuit, for setting the frequency. An oscillation reference device such as a crystal oscillator is connected between X_{IN} and X_{OUT}. When an external clock is used, connect the clock oscillation source to the X_{IN} pin and leave the X_{OUT} pin open.
X_{OUT}	Source oscillation clock output	Output	—	
PWON	Internal power on input	Input	—	Incorporates the power saving circuit. Its control inputs are PWON and PWOFF. The 22-stage frequency divider and RAM are put in the idle state by a PWOFF input.
PWOFF	Internal power off input	Input	—	
RESET (DV)	Frequency divider reset input	Input	—	Incorporates the 22-stage frequency divider as the crystal oscillation reference device. This is a reset input for the frequency divider.
BDIN	Low voltage detector input	Input	—	The low voltage detector circuit is built in. A resistor should be connected to the BDIN pin for voltage sensing.
INT_A	Interrupt request A signal	Input	Interrupt disable	This input signal is for an interrupt request. The request is accepted on the rising edge of the signal. Besides these external input signals, an interrupt request INT_T from the 22-stage frequency divider output signal is sensed as an interrupt.
INT_B	Interrupt request B signal	Input	Interrupt disable	
$LC_0 \sim LC_{24}$	Liquid crystal display segment output	Output	—	Incorporates the liquid crystal display direct drive circuit. It is suitable for liquid crystal display at 1/4 duty and 1/3 bias. The output ports for direct drive of the liquid crystal display are port LC ($LC_0 \sim LC_{24}$) and port COM ($COM_0 \sim COM_3$).
$COM_0 \sim COM_3$	Liquid crystal display common output	Output	—	
V_{LCD}	Power supply for liquid crystal display	—	—	This is the power supply terminal for a liquid crystal display. It includes the bias resistor for the segment and common signals.
$D_0 \sim D_{10}$	Output port D	Output	—	This output port consists of 11 bits. Each output is individually latched and can be selected to be set or reset by the contents of register Y. Also 8 bits of the port can be used to fetch 8-bit addresses for external RAM.
$F_0 \sim F_7$	Output port F	Output	Low level	The output port consists of 8 bits. Each output is individually latched and can be set or reset by instructions.
P_0, P_1	Output port P	Output	Low level	This output port consists of 2 bits from which 1 synchronous signal of 1 machine cycle width can be output per instruction. The immediate 7-bit field of an instruction can be output through this port in combination with 5 bits of port F.
$K_0 \sim K_3$	Input/output port K	Input/output	Low level	Ports K and S are 4-bit latched input/output ports through which data can be transferred to and from register A. When output is low-level the output will be high-impedance so it can be used as an input port.
$S_0 \sim S_3$	Input/output S	Input/output	Low level	
T_2	Timing output	Output	—	The timing output is used for testing the device.
RESET (ON)	Internal power-on reset signal	Output	High level	When the internal power supply is switched on, a built in automatic reset circuit generates a high-level reset signal that resets the I/O ports.

6

OPERATIONS

Program Counter PC

The program counter is an 11-bit address register. The high-order 4 bits designate the page number and as a group are called PC_H. The low-order 7 bits designate the address on the page and as a group are called PC_L. The PC designates the address of the 2048 words by 10-bit mask-programmable ROM. The ROM is organized into 16 pages of 128 words. As instructions are fetched from ROM, PC_L is incremented so that unless there is a branch executed instructions are fetched and executed in sequence. Care must be taken when the last instruction on a page (address 127) is executed because when PC_L is incremented it becomes zero with a carry, but the carry is disregarded so the next instruction to be fetched will be the start of the same page. Therefore to move to the next page PC_H must modified by using branch instructions such as BL, BML, BLA and BMLA.

Pages 14 and 15 are special pages designed to accommodate subroutines. Subroutines starting on page 14 can be called by 1-word instructions BM or BMA. These instructions automatically load PC_H to designate page 14 and in addition the return address and control status are saved so they can be restored when the subroutine transfers control back to the main program. If the instructions BM or BMA are executed on page 14, they execute a branch within page 14 without saving any information. If the instructions B or BA are executed on page 14, they execute a branch to page 15.

Stack Registers SK0, SK1, SK2

The 3-level stack register consists of 11-bit registers for storing the contents of the program counter when control is transferred from the main program to a subroutine or interrupt. When control is transferred back to the main program, the PC can be restored. There are 3 levels, but when 1 level is saved for interrupts it leaves 2 levels for subroutine nesting.

Data Pointers DP, DP'

The data pointer is a 7-bit register used to designate the address of RAM or the bit position of output port D. The data pointer is composed of the 3-bit register X and the 4-bit register Y. Internal RAM is organized as 8 files of 16 words. Register X designates the file and register Y designates the word position of a file or the bit position of output port D.

The data pointer DP' is selected by software during interrupt processing to leave the contents of DP unchanged (saves the DP).

External RAM is organized as 16 files of 16 words that can be added to the system to expand memory. Register Y designates the word position of a file while register B designates the file.

Register A (accumulator) and Carry Flags CY, CY'

Register A is the 4-bit accumulator forming the heart of the 4-bit microcomputer. Data processing operations such as arithmetic, transfer, exchange, conversion, and input/output are executed principally through this register.

The carry flags are to store the carry or borrow from the most significant bit of the arithmetic unit resulting from executing the various instructions. It can be tested and used for various purposes. In principle it acts as a 1-bit flag.

The carry flag CY is selected by software to leave the contents of CY unchanged (saves the CY).

Register B (Auxiliary Register)

Register B is a 4-bit register used for temporary storage of 4-bit data. It also is used to designate the file number of external RAM.

Arithmetic Logic Unit (ALU)

The arithmetic logic unit performs 4-bit arithmetic and logical operations. The heart of the ALU is a 4-bit adder and the logic circuit associated with it. It performs operations such as additions, complement conversions, logic arithmetic comparisons and bit processing.

Frequency Divider and Timer

The frequency divider divides the basic oscillation frequency into 22 stages. It is connected to the basic oscillation device through X_{IN} and X_{OUT}. The frequency divider generates the interrupt request signal INT_T to the interrupt control circuit. The frequency divider sets flag CK for controlling the power saving circuit.

Basic oscillation for the timer is the timing signal T_2. The timer is composed of a 7-bit prescaler and a 4-bit counter. Timer flag TMF/F is set when a timer overflows, and is sensed by the TTM instruction. The 4-bit timer counter is set by the STM instruction. Prescaler and timer flag are reset at the same time.

Power Saving Circuit

The power saving circuit is controlled by the CK flag and PW. Its output is input to the internal power supply reset circuit and generates an interrupt request signal RESET (ON). Control is transferred unconditionally to address 0 on page 0 and resets the I/O ports. The interrupt request

signal RESET (ON) generates on the rising edge of internal power supply on reset output. Internal power supply is switched off by the external terminal and stop instruction, but power is maintained to the following circuits:

1. Internal data memory (RAM)
2. Clock oscillation circuit
3. 22-stage frequency divider
4. Low voltage detector circuit
5. Power saving circuit

Low Voltage Detector Circuit

The low voltage detector circuit connects the resistor for sensing voltage to the BDIN terminal. A falling voltage level is sensed by the program and can be displayed by using apt output port.

Interrupt Functions

The M58496-XXXP has internal circuits to process interrupt requests from 4 single level sources. The 4 interrupt request sources are external interrupt signals INT_A and INT_B, internal power supply reset output RESET (ON), output INT_T from the 22-stage frequency divider. Interrupt requests INT_A, INT_B and INT_T are enabled by the instructions EIA, EIB and EIT respectively and disabled by the instruction DIA, DIB and DIT respectively. Interrupt requests from the internal power supply through reset output RESET (ON) cannot be disabled and will cause an interrupt whenever received.

During the interrupt enable state an interrupt request by INT_A or INT_B is accepted on the rising edge of the signal. When an interrupt request is received during the interrupt disable state it is latched, but is not executed. When the disable is removed thereafter by executing the corresponding interrupt enable instruction, the interrupt request will be accepted immediately and control transferred to the interrupt routine because the request was latched. A current interrupt request, held by latching during interrupt disable state is reset when the corresponding interrupt disable instruction is executed.

One level of the 3-level stack register is required when interrupt programs are used. This leaves 2 levels available for subroutine processing. After an interrupt is processed control is returned to the main program by executing a return instruction such as RTI. Care must be taken after starting an interrupt program to save the contents the data pointer DP, register A, carry flag and any other registers used, so the contents can be restored before returning to the main program. The contents must be saved and restored by the interrupt program.

When an interrupt request is accepted the program counter, interrupt enable flag and skip flag are affected as follows:

(1) Program counter
The contents (the current program address) are stored in the stack register. Control is transferred to address 0 on page 0 by a RESET (ON) interrupt, to address 2 on page 0 by an INT_A interrupt, to address 4 on page by an INT_B interrupt or to address 8 on page 0 by an INT_T interrupt by setting the control counter to 00, 02, 04 or 08 respectively. When control is transferred to address 0 page 0, the instruction is invalid and is not executed, so the first instruction is executed from address 1 on page 0.

(2) Interrupt enable flags
When an interrupt request is accepted additional interrupts are disabled until the accepted interrupt is processed. Except that a RESET (ON) interrupt may be accepted at any time.

(3) Skip flags
The skip flags are used to indicate an instruction skip and the NOP state for instructions LXY and LA are saved. A special stack is provided for saving these flags.

6

General-Purpose I/O ports K, S, F, P and D

These 4-bit or 1-bit general-purpose registers are used for such things as data transfer between register A, instruction transfers, 1-bit transfers as selected by register Y, storing 7-bit immediate field data of instructions fetched from ROM, and data transfers between external RAM. Each output has a latch and its output circuit contains an open drain resistor or a pulldown resistor (high-impedance).

I/O ports K, S

Ports K and S are 4-bit latched I/O ports, that can transfer data to and from register A. Output latches are reset by the DIKS instruction when the port is being used as an input port.

Output port F

Port F is an 8-bit latched output port, that has independent latches for each bit. The individual bits can be set by the SF instruction and reset by the RF instruction.

Output port P

Port P is a 2-bit latched output port, that is usually in low-level, but can output the machine cycle high-level synchronous signal by SP_0 or SP_1 instructions. The 7 bits (F_4~F_0, P_1, P_0) can be used for direct fetching of the immediate field of the OTRO instruction.

Output port D

Port D is an 11-bit latched output port, that has independent latches for each bit. The contents for register Y indicate the individual bit to be set by the SD instruction or to be reset by the RD instruction. The 8-bit address of external memory (RAM) is output through this port.

Liquid Crystal Display Drive Circuit

The liquid crystal display direct drive circuit is composed of the following units. A block diagram of the units is shown in Fig. 1.

1 Control counter for the liquid crystal display
This is an octal counter composed of 3 bits and is counted down by the ELC instruction. The contents of the counter select 1 bit of register A and transfer data in order to the segment register RLC by the TLC instruction and determines the frame frequency for the liquid crystal display by transferring the contents of the counter to common register RCOM.

2 Register A
This 4-bit register is the accumulator. Its function is to control data processing, arithmetic operations control functions and input/output of the microcomputer.

3 Segment register RLC
The 26-bit segment register stores selected 1-bit data from register A by execution of the TLC instruction. It shifts 1 bit in order and stores the segment signals for the liquid crystal display device.

4 Common register RCOM
The 4-bit common register stores the common signal for the liquid crystal display. The input for the common register is the converted contents of the control counter for the liquid crystal display.

5 Port LC
The 26-bit latched port LC stores data in parallel by the ELC or DLC instruction from the segment register RLC. A bias resistor provides for the output at 2 levels and the 25 low-order bits are output as standard type. The high-order bit is not output to an external terminal.

6 Port COM
Port COM has 4 bits of latched storage. The data is transferred in parallel by the ELC or DLC instruction through the common register (RCOM). The outputs of this port have 3 biased levels by means of bias resistors.

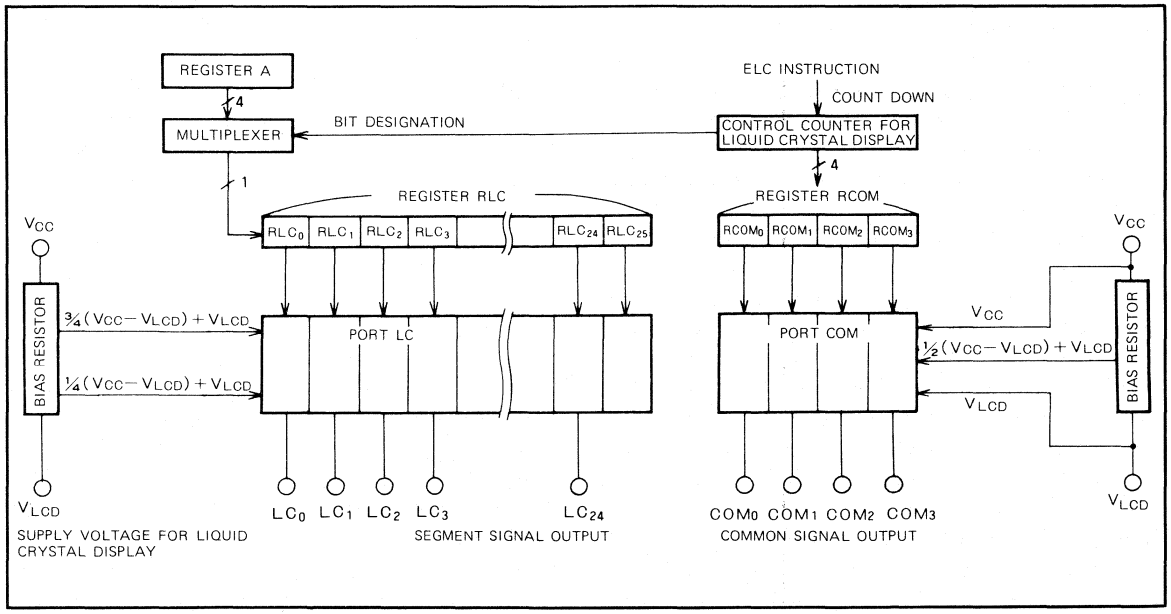
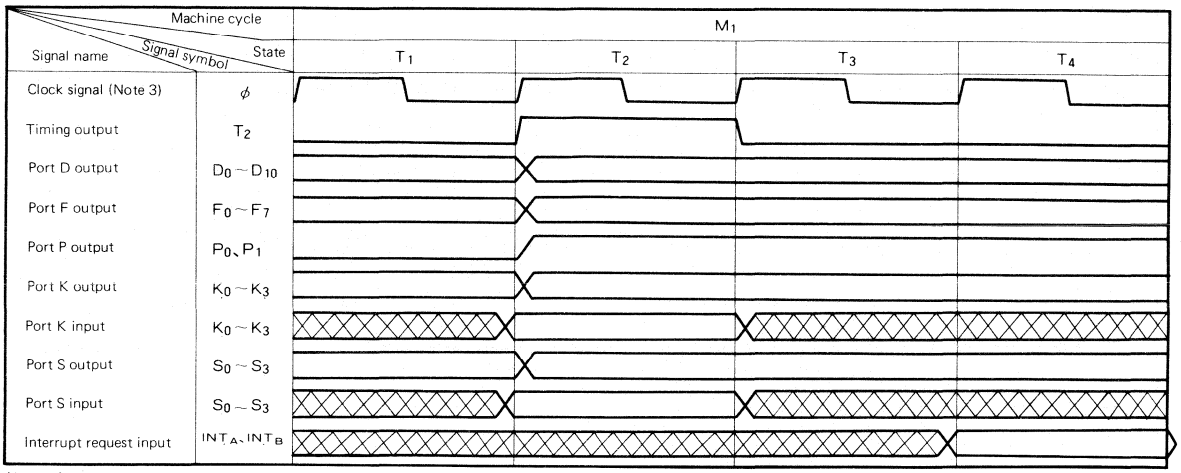


Fig. 1 Liquid crystal display drive circuit block diagram

BASIC TIMING CHART



Note 3: Internal clock signal which is 1/8 of basic oscillation frequency.
4: ▨ indicates an invalid signal input.

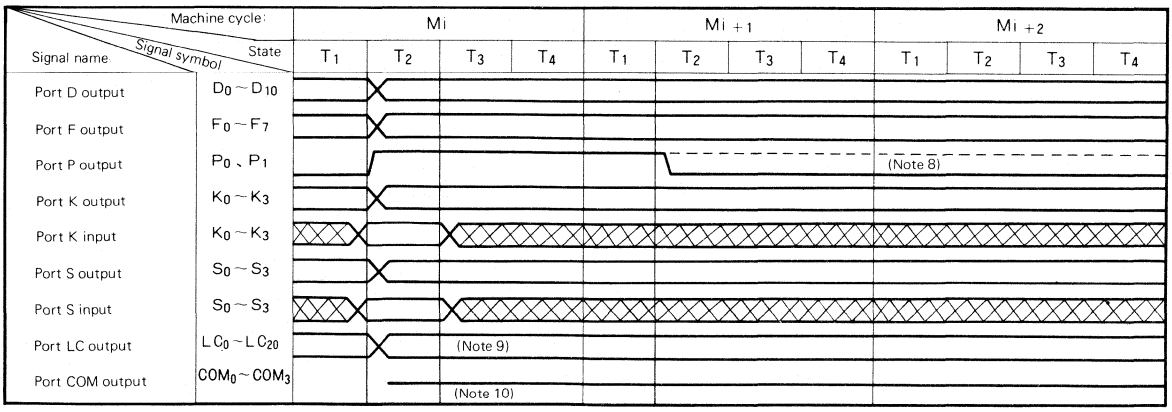
6

INSTRUCTION FETCH TIMING

Machine cycle / Instruction cycle / State	Mi				Mi +1				Mi +2			
	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4
Instruction fetch						(Note 5)						
Instruction execution								(Note 6)				

Note 5: Instruction fetch time can differ depending on the types of the instructions.
6: The instruction which was fetched in the preceding cycle is executed.
7: The execution of the instruction and addressing of ROM and RAM are performed simultaneously.

I/O INSTRUCTION EXECUTION TIMING



Note 8: When an OTRO instruction is executed, the output is latched.
9: Output voltage of port LC depends upon power supply V_{LCD} for the liquid crystal display.
10: Output voltage of port COM has 3 levels depending on the power supply V_{LCD} for the liquid crystal display.

SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

BRANCH AND SUBROUTINE CALL INSTRUCTION EXECUTION TIMING

Operation \ Machine cycle / State	M_i				M_{i+1}				M_{i+2}			
	T_1	T_2	T_3	T_4	T_1	T_2	T_3	T_4	T_1	T_2	T_3	T_4
Instruction B_{xy} (to be operated as the branch instruction, when the instruction BM or BMA was not executed before).												
Program counter		$(PC_L) \leftarrow xy$	$(PC_L) \leftarrow (PC_L)+1$				$(PC_L) \leftarrow (PC_L)+1$					
ROM address		(ROM address) $\leftarrow$ (PC)				(ROM address) $\leftarrow$ (PC)						
Execution of program	Execution of the branch instruction				Execution of the instruction stored in the branched address							
Instruction B_{xy} (to be operated as the branch instruction to page 15, when the instruction BM or BMA was executed before).												
Program counter		$(PC_H) \leftarrow 15$ $(PC_L) \leftarrow xy$	$(PC_L) \leftarrow (PC_L)+1$				$(PC_L) \leftarrow (PC_L)+1$					
ROM address		(ROM address) $\leftarrow$ (PC)				(ROM address) $\leftarrow$ (PC)						
Execution of program	Execution of the branch instruction				Execution of the instruction stored in the branched address on page 15							
Instruction BM_{xy} (subroutine call instruction).												
Program counter		$(PC_H) \leftarrow 14$ $(PC_L) \leftarrow xy$	$(PC_L) \leftarrow (PC_L)+1$				$(PC_L) \leftarrow (PC_L)+1$					
ROM address		(ROM address) $\leftarrow$ (PC)				(ROM address) $\leftarrow$ (PC)						
Stack register		$(SK_2) \leftarrow (SK_1) \leftarrow (SK_0) \leftarrow (PC)$										
Execution of program	Execution of the subroutine call instruction				Execution of the instruction stored in the subroutine called address							
Instruction BL p,xy (branch instruction).												
Program counter		(Temporary register) $\leftarrow P$	$(PC_L) \leftarrow (PC_L)+1$			$(PC_H) \leftarrow$ (Temporary register) $(PC_L) \leftarrow xy$	$(PC_L) \leftarrow (PC_L)+1$					
ROM address		(ROM address) $\leftarrow$ (PC)				(ROM address) $\leftarrow$ (PC)				(ROM address) $\leftarrow$ (PC)		
Execution of program	Page number is stored temporarily				Execution of branch instruction				Execution of the instruction stored in the branched address			
Instruction BML p, xy (subroutine call instruction).												
Program counter		(Temporary register) $\leftarrow P$	$(PC_L) \leftarrow (PC_L)+1$			$(PC_H) \leftarrow$ (Temporary register) $(PC_L) \leftarrow xy$	$(PC_L) \leftarrow (PC_L)+1$					
ROM address		(ROM address) $\leftarrow$ (PC)				(ROM address) $\leftarrow$ (PC)				(ROM address) $\leftarrow$ (PC)		
Stack register						$(SK_2) \leftarrow (SK_1) \leftarrow (SK_0) \leftarrow (PC)$						
Execution of program	Page number is stored temporarily				Execution of the subroutine call instruction				Execution of the instruction stored in the subroutine called address			

Note 11: The instructions BA, BMA, BLA and BMLA have the same execution timing as B, BM, BL and BML respectively as shown. The only difference is that $(PC_L) \leftarrow xy$ is replaced by $(PC_L) \leftarrow x(A)$.

INTERRUPT EXECUTION TIMING

Operation \ Machine cycle / State		M_{i-1}	M_i				M_{i+1}				M_{i+2}			
		T_4	T_1	T_2	T_3	T_4	T_1	T_2	T_3	T_4	T_1	T_2	T_3	T_4
Interrupt request input	INT_A (Note 13)													
Program counter	(PC)				$(PC_L) \leftarrow (PC_L)+1$				$(PC_L) \leftarrow (PC_L)$		$(PC_H) \leftarrow 0$ $(PC_L) \leftarrow 2$		$(PC_L) \leftarrow (PC_L)+1$	
ROM address				(ROM address) $\leftarrow$ (PC)				(ROM address) $\leftarrow$ (PC)				(ROM address) $\leftarrow$ (PC)		
Stack register												$(SK_2) \leftarrow (SK_1) \leftarrow (SK_0) \leftarrow (PC)$		
Execution of program												no execution (skip)		

Note 12: When the instruction executed in the machine cycle M_{i+1} is a BL, BML, BLA or BMLA, the value of address 2 of page 0 is stored in the program counter during M_{i+3}.

13: The interrupt request input INT_B has the same execution timing as INT_A. If the input is low level in the machine cycle M_{i-1} and high level in the machine cycle M_i, the interrupt is executed during the interrupt enable state.

INSTRUCTION CODE LIST

$D_9 \sim D_4$ (Hexadecimal notation) / $D_3 \sim D_0$		00 0000	00 0001	00 0010	00 0011	00 0100	00 0101	00 0110	00 0111	00 1000	00 1001	00 1010	00 1011	00 1100	00 1101	00 1110 \ 00 1111	01 0000 \ 01 0111	01 1000 \ 01 1111	10 0000 \ 10 0111	10 1000 \ 10 1111	11 0000 \ 11 0111	11 1000 \ 11 1111
		0 0	0 1	0 2	0 3	0 4	0 5	0 6	0 7	0 8	0 9	0 A	0 B	0 C	0 D	0E ~ 0F	10 ~ 17	18 ~ 1F	20 ~ 27	28 ~ 2F	30 ~ 37	38 ~ 3F
0000	0	NOP	TLC	INY	SZB 0	SEY 0	SEI 0	SF 0	BL BLA BML BMLA	—	RAR	TAM 0	XAMD 0	A 0	LA 0	—	OTRO	LXY	BM	BMA	B	BA
0001	1	SCOM	DIKS	DEY	SZB 1	SEY 1	SEI 1	SF 1	BL BLA BML BMLA	—	—	TAM 1	XAMD 1	A 1	LA 1	—	OTRO	LXY	BM	BMA	B	BA
0010	2	EIA	SFK	XDP	SZB 2	SEY 2	SEI 2	SF 2	BL BLA BML BMLA	*	IK	TAM 2	XAMD 2	A 2	LA 2	—	OTRO	LXY	BM	BMA	B	BA
0011	3	DIA	SFS	TYA	SZB 3	SEY 3	SEI 3	SF 3	BL BLA BML BMLA	SEAM	IS	TAM 3	XAMD 3	A 3	LA 3	—	OTRO	LXY	BM	BMA	B	BA
0100	4	EIB	*	SC	RT	SEY 4	SEI 4	SF 4	BL BLA BML BMLA	*	TBA	TAM 4	XAMD 4	A 4	LA 4	—	OTRO	LXY	BM	BMA	B	BA
0101	5	DIB	DLC	RC	RTS	SEY 5	SEI 5	SF 5	BL BLA BML BMLA	TAY	—	TAM 5	XAMD 5	A 5	LA 5	—	OTRO	LXY	BM	BMA	B	BA
0110	6	DETS	*	XC	RTI	SEY 6	SEI 6	SF 6	BL BLA BML BMLA	AND	XAB	TAM 6	XAMD 6	A 6	LA 6	—	OTRO	LXY	BM	BMA	B	BA
0111	7	DETR	ELC	*	*	SEY 7	SEI 7	SF 7	BL BLA BML BMLA	EXL	TAB	TAM 7	XAMD 7	A 7	LA 7	—	OTRO	LXY	BM	BMA	B	BA
1000	8	EIT	SP0	*	*	SEY 8	SEI 8	RF 0	BL BLA BML BMLA	*	SB 0	XAM 0	XAMI 0	A 8	LA 8	—	OTRO	LXY	BM	BMA	B	BA
1001	9	DIT	*	SD	*	SEY 9	SEI 9	RF 1	BL BLA BML BMLA	CMA	SB 1	XAM 1	XAMI 1	A 9	LA 9	—	OTRO	LXY	BM	BMA	B	BA
1010	A	STM	SP1	*	*	SEY 10	SEI 10	RF 2	BL BLA BML BMLA	AM	SB 2	XAM 2	XAMI 2	A 10	LA 10	—	OTRO	LXY	BM	BMA	B	BA
1011	B	POF2	*	*	*	SEY 11	SEI 11	RF 3	BL BLA BML BMLA	*	SB 3	XAM 3	XAMI 3	A 11	LA 11	—	OTRO	LXY	BM	BMA	B	BA
1100	C	POF1	OTAD	*	*	SEY 12	SEI 12	RF 4	BL BLA BML BMLA	*	RB 0	XAM 4	XAMI 4	A 12	LA 12	—	OTRO	LXY	BM	BMA	B	BA
1101	D	SDET	*	RD	*	SEY 13	SEI 13	RF 5	BL BLA BML BMLA	*	RB 1	XAM 5	XAMI 5	A 13	LA 13	—	OTRO	LXY	BM	BMA	B	BA
1110	E	TTM	ADRT	*	*	SEY 14	SEI 14	RF 6	BL BLA BML BMLA	AMC	RB 2	XAM 6	XAMI 6	A 14	LA 14	—	OTRO	LXY	BM	BMA	B	BA
1111	F	TCK	TPW	*	SZC	SEY 15	SEI 15	RF 7	BL BLA BML BMLA	AMCS	RB 3	XAM 7	XAMI 7	A 15	LA 15	—	OTRO	LXY	BM	BMA	B	BA

Note 14: This list shows the machine codes and corresponding machine instructions. $D_3 \sim D_0$ indicate the low-order 4 bits of the machine code and $D_9 \sim D_4$ indicate the high-order 6 bits. Hexadecimal numbers are also shown that represent the codes. An instruction may consist of one or two words, but only the first word is listed. Code combination indicated with asterisk (*) and bar (—) must not be used.

Two-word instructions

	Second word
BL	11 0xxx yyyy
BLA	11 1xxx XXXX
BML	10 0xxx yyyy
BMLA	10 1xxx XXXX

6

SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

MACHINE INSTRUCTIONS

Item / Classification	Mnemonic	Instruction code D_9D_8	$D_7D_6D_5D_4$	$D_3D_2D_1D_0$	Hexadecimal	No. of words	No. of cycles	Functions	Skip conditions	Flag CY
RAM address	**LXY x,y**	0 1	1 x x x	y y y y	18y + x	1	1	(X)←x, where, x=0~7 (Y)←y, where, y=0~15	Consecutively described	×
	INY	0 0	0 0 1 0	0 0 0 0	020	1	1	(Y)←(Y)+1	—	×
	DEY	0 0	0 0 1 0	0 0 0 1	021	1	1	(Y)←(Y)−1	—	×
Register to register transfer	**TAB**	0 0	1 0 0 1	0 1 1 1	097	1	1	(A)←(B)	—	×
	TBA	0 0	1 0 0 1	0 1 0 0	094	1	1	(B)←(A)	—	×
	XAB	0 0	1 0 0 1	0 1 1 0	096	1	1	(A)↔(B)	—	×
	TAY	0 0	1 0 0 0	0 1 0 1	085	1	1	(A)←(Y)	—	×
	TYA	0 0	0 0 1 0	0 0 1 1	023	1	1	(Y)←(A)	—	×
	XDP	0 0	0 0 1 0	0 0 1 0	022	1	1	(DP)↔(DP′)	—	×
RAM to accumulator transfer	**TAM j**	0 0	1 0 1 0	0 j j j	0Aj	1	1	(A)←(M(DP)) (X)←(X)∀j, where, j=0~7	—	×
	XAM j	0 0	1 0 1 0	1 j j j	0A8 + j	1	1	(A)↔(M(DP)) (X)←(X)∀j, where, j=0~7	—	×
	XAMD j	0 0	1 0 1 1	0 j j j	0Bj	1	1	(A)↔(M(DP)), (Y)←(Y)−1 (X)←(X)∀j, where, j=0~7	(Y)=15	×
	XAMI j	0 0	1 0 1 1	1 j j j	0B8 + j	1	1	(A)↔(M(DP)), (Y)←(Y)+1 (X)←(X)∀j, where, j=0~7	(Y)=0	×
Arithmetic	**LA n**	0 0	1 1 0 1	n n n n	0Dn	1	1	(A)←n, where, n=0~15	Consecutively described	×
	AM	0 0	1 0 0 0	1 0 1 0	08A	1	1	(A)←(A)+(M(DP))	—	×
	AMC	0 0	1 0 0 0	1 1 1 0	08E	1	1	(A)←(A)+(M(DP))+(CY) (CY)← Carry	—	0/1
	AMCS	0 0	1 0 0 0	1 1 1 1	08F	1	1	(A)←(A)+(M(DP))+(CY) (CY)← Carry	(CY)=0	0/1
	A n	0 0	1 1 0 0	n n n n	0Cn	1	1	(A)←(A)+n, where, n=0~15	Carry = 0	×
	SC	0 0	0 0 1 0	0 1 0 0	024	1	1	(CY)←1	—	1
	RC	0 0	0 0 1 0	0 1 0 1	025	1	1	(CY)←0	—	0
	XC	0 0	0 0 1 0	0 1 1 0	026	1	1	(CY)↔(CY′)	—	(CY′)
	SZC	0 0	0 0 1 1	1 1 1 1	03F	1	1	Skip if (CY)=0	(CY)=0	×
	AND	0 0	1 0 0 0	0 1 1 0	086	1	1	(A)←(A)∧(M(DP))	—	×
	EXL	0 0	1 0 0 0	0 1 1 1	087	1	1	(A)←(A)∀(M(DP))	—	×
	CMA	0 0	1 0 0 0	1 0 0 1	089	1	1	(A)←($\overline{A}$)	—	×
	RAR	0 0	1 0 0 1	0 0 0 0	090	1	1	(An−1)←(An) (CY)←(A0), (A3)←(CY)	—	(A0)
Bit manipulation	**SB i**	0 0	1 0 0 1	1 0 i i	098 + i	1	1	(Mi(DP))←1, where, i=0~3	—	×
	RB i	0 0	1 0 0 1	1 1 i i	09C + i	1	1	(Mi(DP))←0, where, i=0~3	—	×
	SZB i	0 0	0 0 1 1	0 0 i i	03i	1	1	Skip if (Mi(DP))=0, where, i=0~3	(Mi(DP))=0 where, i=0~3	×
Compare	**SEAM**	0 0	1 0 0 0	0 0 1 1	083	1	1	Skip if (M(DP))=(A)	(M(DP))=(A)	×
	SEY y	0 0	0 1 0 0	y y y y	04y	1	1	Skip if (Y)=y, where, y=0~15	(Y)=y, where, y=0~15	×
	SEI n	0 0	0 1 0 1	n n n n	05n	1	1	Skip if (A)=n, where, n=0~15	(A)=n, where, n=0~15	×
	SCOM	0 0	0 0 0 0	0 0 0 1	001	1	1	Skip if (SCA=0) and (SCB=0)	SCA=0 and SCB=0	×

MITSUBISHI ELECTRIC

Item / Classification	Mnemonic	Instruction code D_9D_8	$D_7D_6D_5D_4$	$D_3D_2D_1D_0$	Hexa-decimal	No. of words	No. of cycles	Functions	Skip conditions	Flag CY
Branch	**B xy**	1 1	0 x x x	y y y y	**3xy**	1	1	$(PC_L) \leftarrow 16x + y$	—	×
	(Note 15)							$(PC_H) \leftarrow 15$, $(PC_L) \leftarrow 16x + y$		
	BL pxy	0 0	0 1 1 1	p p p p	**07p**	2	2	$(PC_H) \leftarrow p$	—	×
		1 1	0 x x x	y y y y	**3xy**			$(PC_L) \leftarrow 16x + y$		
	BA xX	1 1	1 x x x	X X X X	**38X + x**	1	1	$(PC_L) \leftarrow 16x + (A)$	—	×
	(Note 15)							$(PC_H) \leftarrow 15$, $(PC_L) \leftarrow 16x + (A)$		
	BLA pxX	0 0	0 1 1 1	p p p p	**07p**	2	2	$(PC_H) \leftarrow p$	—	×
		1 1	1 x x x	X X X X	**38X + x**			$(PC_L) \leftarrow 16x + (A)$		
Subroutine call	**BM xy**	1 0	0 x x x	y y y y	**2xy**	1	1	$(SK2) \leftarrow (SK1) \leftarrow (SK0) \leftarrow (PC)$ $(PC_H) \leftarrow 14$, $(PC_L) \leftarrow 16x + y$	—	×
	(Note 15)							$(PC_H) \leftarrow 14$, $(PC_L) \leftarrow 16x + y$		
	BML pxy	0 0	0 1 1 1	p p p p	**07p**	2	2	$(SK2) \leftarrow (SK1) \leftarrow (SK0) \leftarrow (PC)$	—	×
		1 0	0 x x x	y y y y	**2xy**			$(PC_H) \leftarrow p$, $(PC_L) \leftarrow 16x + y$		
	BMA xX	1 0	1 x x x	X X X X	**28X + x**	1	1	$(SK2) \leftarrow (SK1) \leftarrow (SK0) \leftarrow (PC)$ $(PC_H) \leftarrow 14$, $(PC_L) \leftarrow 16x + (A)$	—	×
	(Note 15)							$(PC_H) \leftarrow 14$, $(PC_L) \leftarrow 16x + (A)$		
	BMLA pxX	0 0	0 1 1 1	p p p p	**07p**	2	2	$(SK2) \leftarrow (SK1) \leftarrow (SK0) \leftarrow (PC)$	—	×
		1 0	1 x x x	X X X X	**28X + x**			$(PC_H) \leftarrow p$, $(PC_L) \leftarrow 16x + (A)$		
Return	**RTI**	0 0	0 0 1 1	0 1 1 0	**036**	1	1	$(PC) \leftarrow (SK0) \leftarrow (SK1) \leftarrow (SK2)$ Restore interrupt skip flags	—	×
	RT	0 0	0 0 1 1	0 1 0 0	**034**	1	1	$(PC) \leftarrow (SK0) \leftarrow (SK1) \leftarrow (SK2)$	—	×
	RTS	0 0	0 0 1 1	0 1 0 1	**035**	1	1	$(PC) \leftarrow (SK0) \leftarrow (SK1) \leftarrow (SK2)$	Unconditional	×
Input/output	**DIKS**	0 0	0 0 0 1	0 0 0 1	**011**	1	1	Port K and S go to floating state	—	×
	IK	0 0	1 0 0 1	0 0 1 0	**092**	1	1	$(A) \leftarrow (K)$	—	×
	IS	0 0	1 0 0 1	0 0 1 1	**093**	1	1	$(A) \leftarrow (S)$	—	×
	SFK	0 0	0 0 0 1	0 0 1 0	**012**	1	1	$(K) \leftarrow (A)$	—	×
	SFS	0 0	0 0 0 1	0 0 1 1	**013**	1	1	$(S) \leftarrow (A)$	—	×
	SD	0 0	0 0 1 0	1 0 0 1	**029**	1	1	$(D(Y)) \leftarrow 1$, where, $0 \leqq (Y) \leqq 10$	—	×
	RD	0 0	0 0 1 0	1 1 0 1	**02D**	1	1	$(D(Y)) \leftarrow 0$, where, $0 \leqq (Y) \leqq 10$	—	×
	ADRT	0 0	0 0 0 1	1 1 1 0	**01E**	1	1	$(D) \leftarrow 0$	—	×
	OTAD	0 0	0 0 0 1	1 1 0 0	**01C**	1	1	$(D_7 \sim D_4) \leftarrow (B)$ $(D_3 \sim D_0) \leftarrow (Y)$	—	×
	SF m	0 0	0 1 1 0	0 m m m	**06m**	1	1	$(Fm) \leftarrow 1$, where, $m = 0 \sim 7$	—	×
	RF m	0 0	0 1 1 0	1 m m m	**068 + m**	1	1	$(Fm) \leftarrow 0$, where, $m = 0 \sim 7$	—	×
	OTRO mn	0 1	0 m m m	n n n n	**1mn**	1	1	$(F_0 \sim F_3) \leftarrow n$, where, $n = 0 \sim 15$ $(F_4, P_0, P_1) \leftarrow m$, where, $m = 0 \sim 7$	—	×
	SP0	0 0	0 0 0 1	1 0 0 0	**018**	1	1	$(P_0) \leftarrow 1$	—	×
	SP1	0 0	0 0 0 1	1 0 1 0	**01A**	1	1	$(P_1) \leftarrow 1$	—	×
	TLC	0 0	0 0 0 1	0 0 0 0	**010**	1	1	$(R(LC_0)) \leftarrow (Ai)$, where, $i = 0 \sim 3$ $(R(LC_{n+1})) \leftarrow (R(LC_n))$	—	×
	ELC	0 0	0 0 0 1	0 1 1 1	**017**	1	1	$(P(LC_n)) \leftarrow (R(LC_{n+1}))$ $(P(COM_n)) \leftarrow (R(COM_n))$	—	×
	DLC	0 0	0 0 0 1	0 1 0 1	**015**	1	1	$(P(LC_n)) \leftarrow (R(LC_{n+1}))$ $(P(COM)) \leftarrow \frac{1}{2}(V_{CC} - V_{LCD}) + V_{LCD}$	—	×

6

SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

Item / Classification	Mnemonic	Instruction code D_9D_8	$D_7D_6D_5D_4$	$D_3D_2D_1D_0$	Hexadecimal	No. of words	No. of cycles	Functions	Skip conditions	Flag CY
Interrupt	**EIA**	0 0	0 0 0 0	0 0 1 0	**002**	1	1	Enables interruption of INT_A signal.	—	×
	DIA	0 0	0 0 0 0	0 0 1 1	**003**	1	1	Disables interruption of INT_A signal.	—	×
	EIB	0 0	0 0 0 0	0 1 0 0	**004**	1	1	Enables interruption of INT_B signal.	—	×
	DIB	0 0	0 0 0 0	0 1 0 1	**005**	1	1	Disables interruption of INT_B signal.	—	×
	EIT	0 0	0 0 0 0	1 0 0 0	**008**	1	1	Enables interruption of INT_T signal.	—	×
	DIT	0 0	0 0 0 0	1 0 0 1	**009**	1	1	Disables interruption of INT_T signal.	—	×
Timer	**STM**	0 0	0 0 0 0	1 0 1 0	**00A**	1	1	(TM)←(A), (TM F/F)←0 7-bit prescaler presetting	—	×
	TTM	0 0	0 0 0 0	1 1 1 0	**00E**	1	1	Skip if (TM F/F) = 1	(TM F/F) = 1	×
Power supply control	**TCK**	0 0	0 0 0 0	1 1 1 1	**00F**	1	1	Skip if (CK F/F) = 1	(CK F/F) = 1	×
	POF1	0 0	0 0 0 0	1 1 0 0	**00C**	1	1	(CK F/F)←0	—	×
	POF2	0 0	0 0 0 0	1 0 1 1	**00B**	1	1	(PW F/F)←0	—	×
	TPW	0 0	0 0 0 1	1 1 1 1	**01F**	1	1	Skip if (PW F/F) = 1	(PW F/F) = 1	×
	DETS	0 0	0 0 0 0	0 1 1 0	**006**	1	1	(DET F/F)←1	—	×
	DETR	0 0	0 0 0 0	0 1 1 1	**007**	1	1	(DET F/F)←0	—	×
	SDET	0 0	0 0 0 0	1 1 0 1	**00D**	1	1	Skip if (BD_{OUT}) = 1	(BD_{OUT}) = 1	×
Misc.	**NOP**	0 0	0 0 0 0	0 0 0 0	**000**	1	1	No operation	—	×

Note 15: Instructions B, BA, BM or BMA execute the second function of the functions column when executed, provided that none of instructions RT, RTS, BL, BML, BLA or BMLA was executed after execution of instruction BM or BMA.

16: When the M58496-XXXP generates a skip it is not necessary to increment the program counter so no additional cycles are required for execution.

Symbol	Meaning	Symbol	Meaning
A	4-bit register (accumulator)	P(COMn)	Common output port for liquid crystal display
Ai	Indicates the bits of register A. Where i=1~3	P(LCn)	Segment output port for liquid crystal display
B	4-bit auxiliary register	PW F/F	1-bit power supply control flag display
BD_{OUT}	Battery detector signal	R(COMn)	Common register for liquid crystal display (4 bits)
CK F/F	1-bit 1-second flag	R(LCn)	Segment register for liquid crystal display (25 bits)
CY	1-bit carry flag	S	4-bit I/O port
CY′	1-bit carry flag	SCA	Output of bit A of control counter for liquid crystal display
D	11-bit output port	SCB	Output of bit B of control counter for liquid crystal display
Di	Indicates the bits of port D. Where i=0~3	SK0	11-bit stack register
D(Y)	The bit of port D addressed by Y	SK1	11-bit stack register
DP	7-bit data pointer composed of register Y, X	SK2	11-bit stack register
		TM	4-bit timer/counter
Y, Y′	4-bit register	TM F/F	1-bit timer/counter flag
X, X′	3-bit register	xx	2-bit binary variable
DP′	7-bit data pointer	yyyy	4-bit binary variable
DET F/F	1-bit battery detector flag	mmm	3-bit binary variable
F	8-bit output port	nnnn	4-bit binary variable
Fi	Indicates the bits of port F. Where i=0~7	ii	2-bit binary variable
K	4-bit I/O port	jjj	3-bit binary variable
M(DP)	4-bit data of memory addressed by data pointer DP	XXXX	4-bit unknown binary variable (the value does not affect execution)
		←	Indicates direction of data flow
Mi(DP)	A bit of data of memory addressed by data pointer DP where i=0~3	( )	Indicates contents of register memory, etc.
		∀	Exclusive OR
PC	11-bit program acounter composed of PC_L, PC_H	∧	AND
		—	Negation
PC_L	Low-order 7 bits of the program counter	X	Indicates flag is unaffected by instruction execution
PC_H	High-order 4 bits of the program counter	xy	Label used to indicate the address
P_0	4-bit output port	C	Hexadecimal number C + binary number-X
P_1	4-bit output port	+	
		x	

MELPS 8-48 MICROCOMPUTERS

7

DESCRIPTION

The MELPS 8-48 LSI family is a low-cost high-performance single-chip microcomputer series. The functions have been integrated. For example the CPU, ROM, RAM, I/O ports, timer and other circuits are all on one chip. The MELPS 8-48 family has the following three configurations to meet the requirements of different applications for various ROM and RAM capacities.

M5L8048-XXXP

ROM	1024 bytes
RAM	64 bytes
I/O	27 pins

M5L8049-XXXP

ROM	2048 bytes
RAM	128 bytes
I/O	27 pins

M5L8748S

EPROM	1024 bytes
RAM	64 bytes
I/O	27 pins

Timer and interrupt inputs are also built into these chips. The program memory capacity can easily be expanded to 4K bytes. The M5L8243P input/output expander chip can be used to extend the I/O capability. The family of microcomputers allows designers to fabricate systems for applications simply and quickly.

The M5L8048-XXXP contains 1K bytes of read only memory and the M5L8049-XXXP contains 2K bytes. The contents of the memory is set by a mask during manufacture. This makes it practical to mass produce ROMs containing customer developed programs.

The M5L8748S contains 1K bytes of EPROM and is pin-compatible with the M5L8048-XXXP. Its memory can be electrically written and changed by the user. This chip can be used while a system is being developed and subject to modifications. Once the system has been checked out and the program debugged, the program can be masked in the M5L8048-XXXP.

A cross assembler, the MELPS 8-48, is available for use with this family of microcomputers. Designers will find the assembler convinient and easy to use.

BASIC FUNCTION BLOCKS

Program Memory (ROM)

The M5L8048-XXXP and M5L8748S contain 1024 bytes of ROM, in the case of the M5L8748S, it is EPROM and its contents can easily be changed by the user. The M5L8049-XXXP contains 2048 bytes of ROM. The program for the users application is stored in this ROM. Addresses 0, 3, 7 of the ROM are reserved for special functions. Table 1 shows the meaning and function of these three special addresses.

Table 1 Reserved, defined addresses and their meanings and functions

Address	Meaning and function
0	The first instruction executed after a system reset.
3	The first instruction executed after an external interrupt is accepted.
7	The first instruction executed after a timer interrupt is accepted.

The ROM can be used to store constants and other 8-bit fixed data in addition to the program. Instructions such as MOVP A, @A and MOVP3 A, @A can be used to access the constants and data. The data could be in the form of tables, and can be easily looked up.

7

Data Memory (RAM)

The M5L8048-XXXP and M5L8748S contain 64 bytes of RAM. The M5L8049-XXXP contains 128 bytes of RAM. The RAM is used for data storage and manipulation and is divided into sections for more efficient processing. Addresses 0~7 and 24~31 form two banks of general purpose registers that can be directly addressed. Addresses 0~7 compose bank 0 and are numbered R0~R7. Addresses 24~31 compose bank 1 and are also numbered R0~R7. Only one bank is active at a time. The instructions SEL RB0 and SEL RB1 are used to select the working bank. Fig. 1 shows the division of the RAM and its mapping.

Addresses 8~23 compose an 8-level program counter stack. The details for using the stack will be found in the "Program Counter and Stack" section. Please refer to that section for details.

The remaining section, addresses 32 and above, must be accessed indirectly using the general-purpose registers R0 or R1. Of course all addresses can be indirectly addressed using the general-purpose registers R0 and R1.

A good practice to simplify programming is to reserve general-purpose register bank 0 for use of the main program and register bank 1 for interrupt programs. For example if register bank 0 (addressed 0~7) is reserved for processing data by the main program, when an interrupt is accepted the first instruction would be to switch the working registers from bank 0 to bank 1. This would save the data of the main program (addresses 0~7). The interrupt program

FUNCTION OF MELPS 8-48 MICROCOMPUTERS

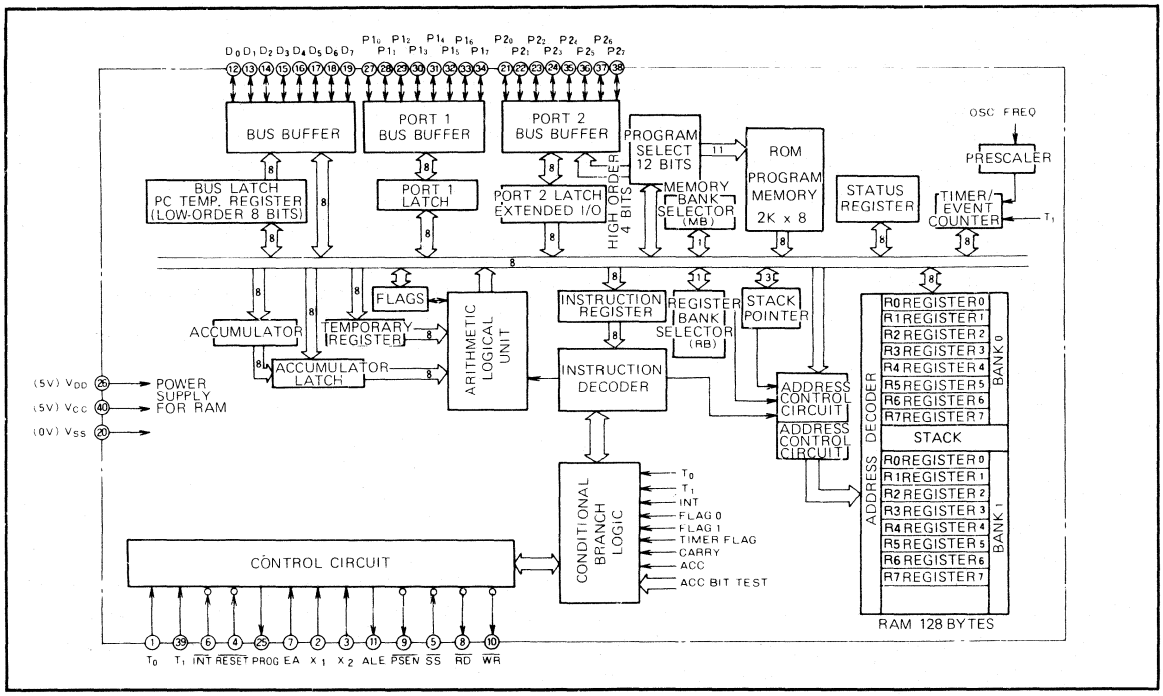


Fig. 1 Block diagram of M5L8039P, M5L8048-XXXP and M5L8049-XXXP

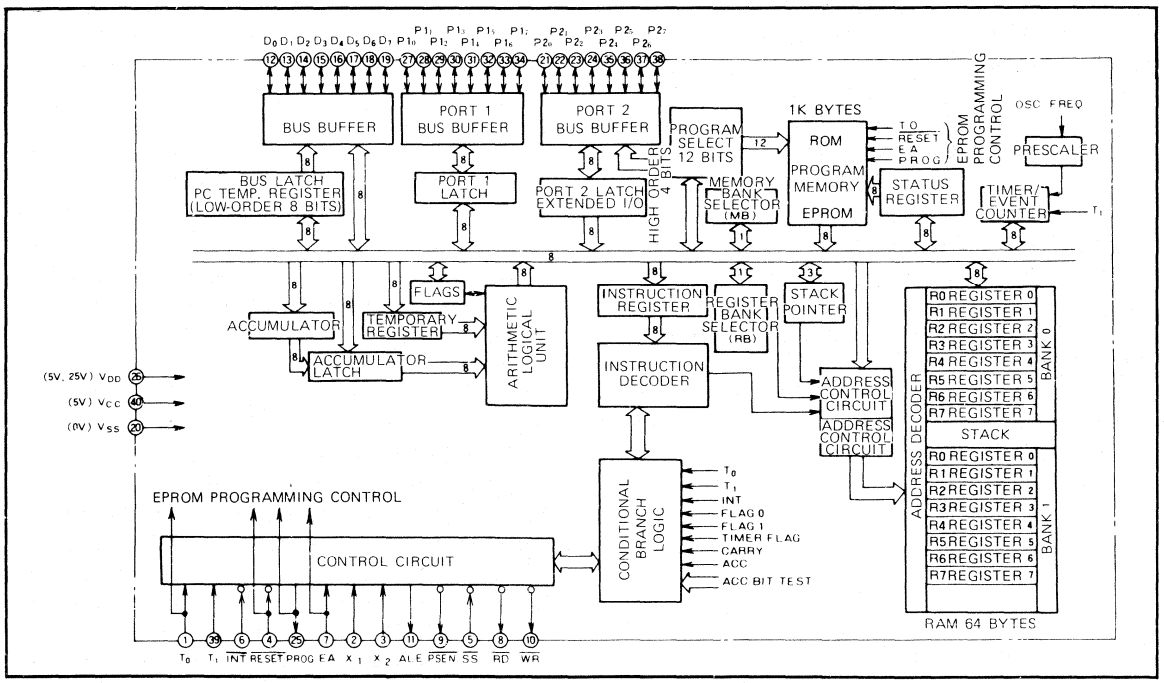


Fig. 2 Block diagram of M5L8748S

can then freely use register bank1 (addresses 24~31) without destroying or altering data of the main program. When the interrupt processing is complete and control is returned to the main program by the RETR instruction, register bank 0 (in this example) is automatically restored as the working register bank at the same time the main program counter is restored.

Addresses 0~31 have special functions, but when not all of the registers are required, the ones not needed can be used for general storage. This includes both banks of general-purpose registers and the stack.

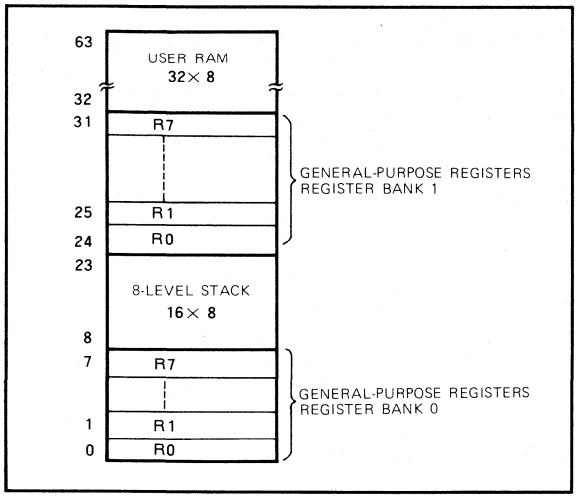


Fig. 1 Data memory (RAM)

PROGRAM COUNTER (PC) AND STACK (SK)

The MELPS 8-48 program counter is composed of a 12-bit binary counter as shown in Fig. 2. The low-order 10 bits can address 1024 bytes of memory. When the high-order 2 bits are zero, the internal, on chip memory is accessed. The high-order 2 bits can have the values 1~3, which allows the user to add up to three banks of 1024 bytes. The program counter can address up to 4096 bytes of memory.

Addresses 8~23 of RAM are used for the stack (program counter stack). The stack provides an easy and automatic means of saving the program counter and other control information when an interrupt is accepted or a subroutine is called. For example, if control is with the main program and an interrupt is accepted, the contents of the 12-bit PC (program counter) is saved in the top of the stack, so it can be restored when control is returned to the main program. In addition to the PC, the high-order 4 bits of the PSW (program status word) are saved in the stack and restored along with the PC. A total of 16 bits are saved, the 12-bit PC and 4 bits of the PSW. A 3-bit stack pointer is associated with the stack. This pointer is a part of the PSW and indicates the top of the stack. The stack pointer indicates the next empty location (top of the stack), in case of an empty stack the top of the stack is the bottom of the stack. The data memory addresses associated with the stack pointer along with the data storage sequence are shown in Fig. 3.

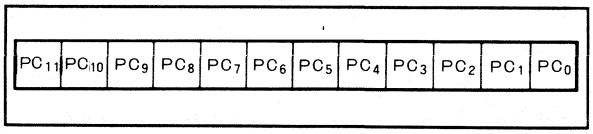


Fig. 2 Program counter

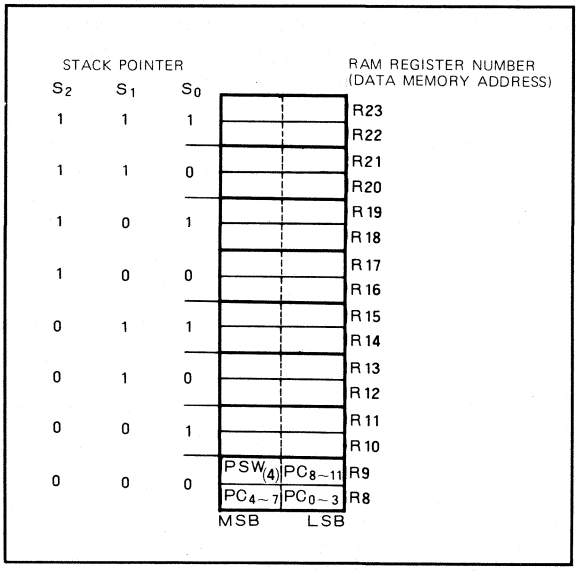


Fig. 3 Relation between the program counter stack and the stack pointer

7

FUNCTION OF MELPS 8-48 MICROCOMPUTERS

PROGRAM STATUS WORD (PSW)

The PSW (program status word) is stored in 8 bits of register storage. The configuration of the PSW is shown in Fig. 4. The high-order 4 bits of the PSW are stored in the stack, along with the PC, when an interrupt is accepted or a subroutine call executed. When control is returned to the main program by RETR both the PC and the high-order 4 bits of PSW are restored. When control is returned by RET only the PC is restored, so care must be taken to assure that the contents of the PSW was not unintentionally changed.

The order and meaning of the 8 bits of the PSW are shown below.

Bit 0~2: Stack pointer (S_0, S_1, S_2)
Bit 3: Unused (always 1)
Bit 4: Working register bank indicator
Bit 5: Flag 0 (value is set by the user and can be tested)
Bit 6: Auxiliary carry (AC) (it is set/reset by instructions ADD and ADC and used by instruction DA A).
Bit 7: Carry bit (C) (indicates an overflow after execution)

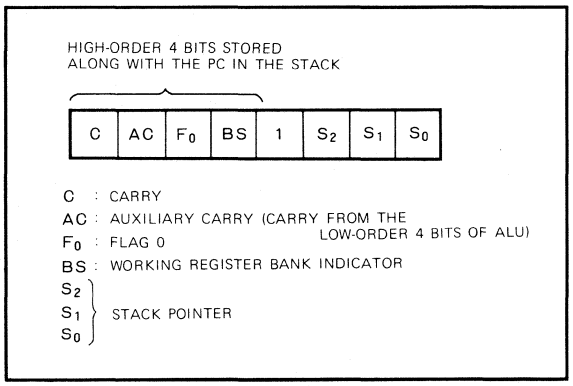


Fig. 4 Program status word

I/O PORTS

The MELPS 8-48 has three 8-bit ports, which are called data bus, port 1 and port 2.

Port 1 and Port 2

Ports 1 and 2 and both 8-bit ports with identical properties. The output data of these ports are retained and do not change until another output is loaded into them. When used as inputs the input data is not retained so the input signals must be maintained until an input instruction is executed and completed.

Ports 1 and 2 so-called quasi-bidirectional ports have a special circuit configuration to accomplish this. The special circuit is shown in Fig. 5. All terminals of ports 1 and 2 can be used for input or output.

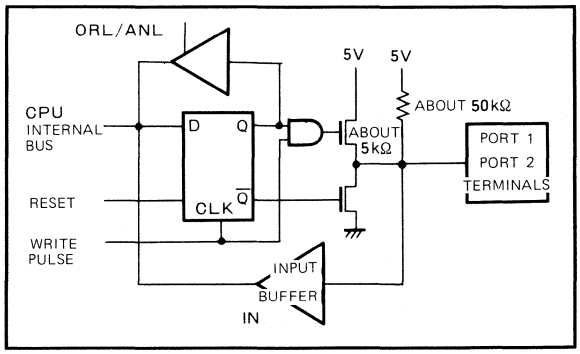


Fig. 5 I/O ports 1 and 2 circuit

Internal on chip pull-up resistors are provided for all the ports. Through the use of pull-up resistors, TTL standard high-level or low-level signals can be supplied. Therefore each terminal can be used for both input and output. To shorten switching time from low-level to high-level, when 1s are output, a device of about 5kΩ or lower is inserted for a short time (about 500ns when using a 6MHz crystal oscillator).

A port used for input must output all 1s before it reads the data from the input terminal. After resetting, a port is set to an input port and remains in this state, therefore it is not necessary to output all 1s if it is to be used for input. In short a port being used for output must output 1s before it can be used for input.

The individual terminals of quasi-bidirectional ports can be used for input or output. Therefore some terminals can be in the input mode while the remaining terminals of a port are in the output mode. This capability of ports 1 and 2 is convenient for inputting or outputting 1-bit or data with few bits. The logical instructions ANL and ORL can easily be used to manipulate the input or output of these ports.

Data Bus (Port 0)

The data bus is an 8-bit bidirectional port, which is used with I/O strobed signals. When the data bus is used for output the output data is latched, but if it is used for input the data is not latched. Unlike ports 1 and 2, which can have individual terminals in the input or output mode, all terminals of the data bus are in the input or output mode.

When the data bus is used as a static port the OUTL instruction can be used to output data and the INS instruction to input data. Strobe pulse $\overline{RD}$ is generated while the INS instruction is being executed or $\overline{WR}$ while OUTL is being executed.

The data bus read/write using MOVX instructions, but then the data bus is a bidirectional port. To write into the data bus a $\overline{WR}$ signal is generated and the data is valid when $\overline{WR}$ goes high. When reading from the data bus, an $\overline{RD}$ signal is generated. The input levels must be maintained until $\overline{RD}$ goes high. When the data bus is not reading/writing, it is in the high-impedance state.

CONDITIONAL JUMPS USING TERMINALS T_0, T_1 and $\overline{INT}$

Conditional jump instructions are used to alter program depending on internal and external conditions (states). Details of the jump instructions for the MELPS 8-48 can be found in the section on machine instructions.

The input signal status of T_0, T_1 and $\overline{INT}$ can be checked by the conditional jump instructions. These input terminals, through conditional jump instructions such as JTO and JNTO, can be used to control a program. Programs and processing time can be reduced by being able to test data in input terminal rather than reading the data into a register and then testing it in the register.

Terminal T_0, T_1 and $\overline{INT}$ have other functions and uses that are not related to conditional jump instructions. The details of these other functions and uses can be found in the section on terminal functions.

INTERRUPT

The CPU recognizes an external interrupt by a low-level state at the $\overline{INT}$ terminal. A "Wired-OR" connection can be used for checking multiple interrupts.

The $\overline{INT}$ terminal is tested for an interrupt request at the ALE signal output of every machine cycle. When an interrupt is recognized and accepted, control is transferred to the interrupt handling program. This is accomplished by an unconditional jump to address 3 of program memory, which is the start of the interrupt handling program, at the same time the program counter and 4 high-order bits of PSW are automatically moved to the top of the stack.

The interrupt level is one, so the next interrupt cannot be accepted until the current interrupt processing has been completed. The RETR instruction terminates the interrupt processing. That is to say, the next interrupt can not be accepted until the RETR instruction is executed. The next interrupt can be accepted at the start of the second cycle of the RETR instruction (2-cycle instruction). Time/event counter overflow which causes an interupt request also will not be accepted.

After the processing for an interrupt is completed control is returned to the main program. This is accomplished by executing RETR which restores the program counter and PSW automatical and checks $\overline{INT}$ and the time/event counter overflow for an interrupt request. If there is an interrupt request, the control will not be returned to the main program but will be transferred to the interrupt handling program.

An external interrupt has a higher priority than a timer interrupt. This means that, if an external and timer interrupt request are generated at the same time, the external interrupt has the priority and will be accepted first.

When a second level of external interrupt is required, the timer interrupt, if not being used, can provide this. The procedure for this is to first disable the timer interrupt, set the timer/event counter to FF_{16} and put the CPU in the event counter mode. After this has been done, if T_1 input is changed to low-level from high-level, an interrupt is generated in address 7.

Terminal $\overline{INT}$ can also be tested using a conditional jump instruction. For more details on this procedure, check the "Conditional Jumps Using Terminals T_0, T_1 and $\overline{INT}$" section.

FUNCTION OF MELPS 8-48 MICROCOMPUTERS

TIMER/EVENT COUNTER

The timer/event counter for the MELPS 8-48 is an 8-bit counter, that is used to measure time delays or count external events. The same counter is used to measure time delays or count external events by simply changing the input to the counter.

The counter can be initialized by executing an MOV T, A instruction. The value of the counter can be read for checking by executing an MOV A, T instruction. Reset will stop the counting but the counter is not cleared, so counting can be resumed.

The largest number the counter can contain is FF_{16}. If it is incremented by 1 when it contains FF_{16}, the counter will be reset to 0, the overflow flag is set and a timer interrupt request is generated.

The conditional jump instruction JTF can be used to test the overflow flag. Care must be used in executing the JTF instruction because the overflow flag is cleared (reset) when executed. When a timer interrupt is accepted, the control is transferred to address 7 of program memory.

When both a timer and external interrupt request are generated at the same time, the external interrupt is given priority and will be accepted first by automatically jumping to address 3 of program memory. The timer interrupt request is kept and will be processed when the external interrupt has been completed and a PETR is executed. A latched timer interrupt request is cancelled when a timer interrupt request is generated. A timer interrupt request can be disabled by executing a DIS TCNTI instruction.

The STRT CNT instruction is used to change the counter to an event counter. Then terminal T_1 signal becomes the input to the event counter and an event is counted each full cycle (low-high-low one event). The maximum rate that can be counted is one time in 3 machine cycles (7.5μs when using 6MHz crystal). The high-level at T_1 must be maintained at least 1/5 of the cycle time (500ns when using 6MHz crystal).

The STRT T instruction is used to change the counter to a timer. The internal clock signal becomes the input to the timer. The internal clock is 1/32 of 400kHz (when using 6MHz crystal) or 12.5kHz. The timer is therefore counted up every 80μs. Fig. 6 shows the timer/event counter.

The counter can be initialized by executing an MOV T, A instruction. The timer can be used to measure 80μs~ 20ms in multiples of 80μs. When it is necessary to measure over 20ms (maximum count 256x80μs) of delay time the number of overflows,one every 20ms, can be counted by the program. To measure times of less than 80μs; external clock pulses can be input through T_1 while the counter is in the event counter mode. Every third (or more) ALE signal can be used instead of an external clock.

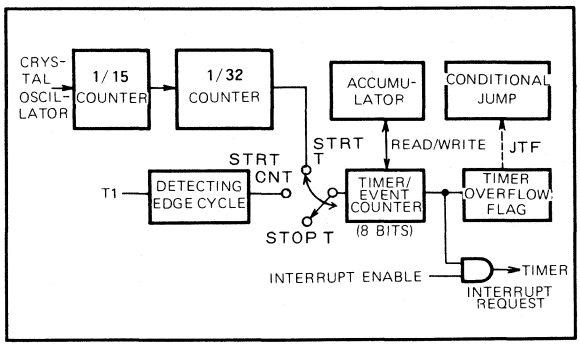


Fig. 6 Timer/event counter

MELPS 8-48 CYCLE TIMING

The output of the state counter is 1/3 the input frequency from the oscillator. When a 6MHz crystal is used for input, the output would be 2MHz (500ns). A CLK signal is generated every 500ns (one state cycle) which is used for the demarcation of each machine state. The instruction ENT0 CLK will output the CLK signal through terminal T_0. The input of the cycle counter is CLK (state cycle) and the output is an ALE signal which is generated every 5 state cycles.

Fig. 8 Shows the relationship between clock and generated cycles.

One machine cycle contains 5 states with a CLK signal for demarcation of each state. The MELPS 8-48 instructions are executed in one machine cycle or two machine cycles. An instruction cycle can be one or two machine cycles as shown in Fig. 9.

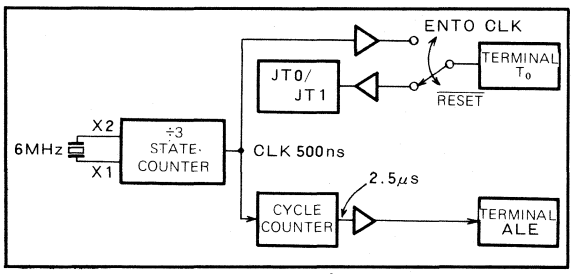


Fig. 7 Clocking cycle generation

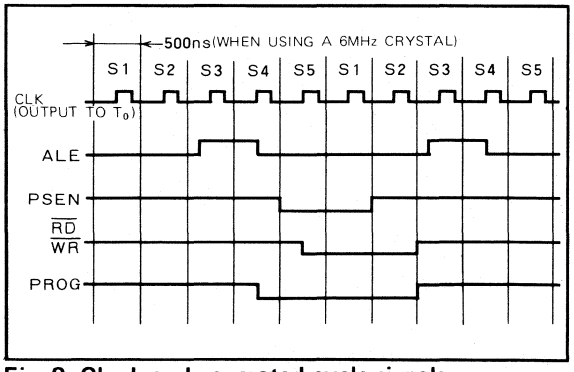


Fig. 8 Clock and generated cycle signals

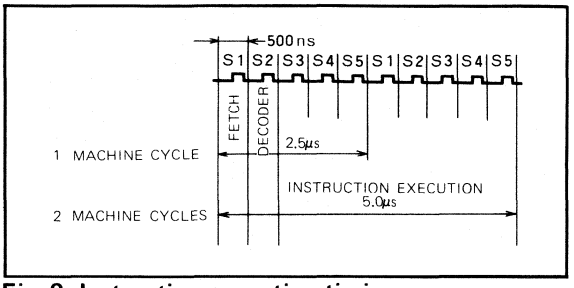


Fig. 9 Instruction execution timing

RESET

The reset terminal is for resetting the CPU. A Schmitt trigger circuit along with a pull-up register are connected to it on the chip. A reset can easily be generated by attaching a 1μF as capacitor as shown in Fig. 10. An external reset pulse applied at RESET must remain at low-level for at least 50ms after power has been turned on and reached its normal level.

The reset function causes the following initialization within the CPU.

1. Program counter is reset to 0.
2. Stack pointer is reset to 0.
3. Register bank is reset to 0.
4. Memory bank is reset to 0.
5. Data bus is cleared to high-impedance state.
6. Ports 1 and 2 are reset to input mode.
7. External and timer interrupts are reset to disable state.
8. Timer is stopped.
9. Timer overflow flag is cleared.
10. Flags F_0 and F_1 are cleared.
11. Clock output for terminal T_0 is disabled.

Note 1: On the M5L8748S the RESET terminal, in addition to being used for the reset function, is also used when reading and writing data in the EPROM on the chip. Details on this will be found in the section on reading and writing data in the M5L8748S.

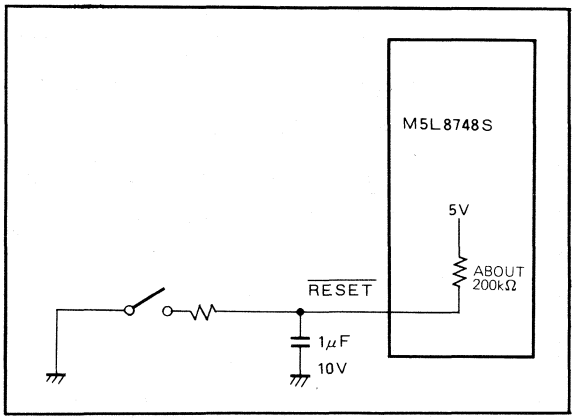


Fig. 10 Example of a reset circuit

7

FUNCTION OF MELPS 8-48 MICROCOMPUTERS

SINGLE-STEP OPERATION

The terminal $\overline{SS}$ on the MELPS 8-48 is provided to facilitate single-step operation. In single-step operation, the CPU stops after the execution of each instruction is completed and the memory address (12 bits) of the next instruction to be fetched is output through the data bus (8 bits) plus the low-order 4 bits of port 2 (P_{20}~P_{23}). The user can use this to trace the flow of this program instruction by instruction and will find this an aid in program debugging. Single-step operation is controlled through $\overline{SS}$ and ALE as shown in Fig. 11

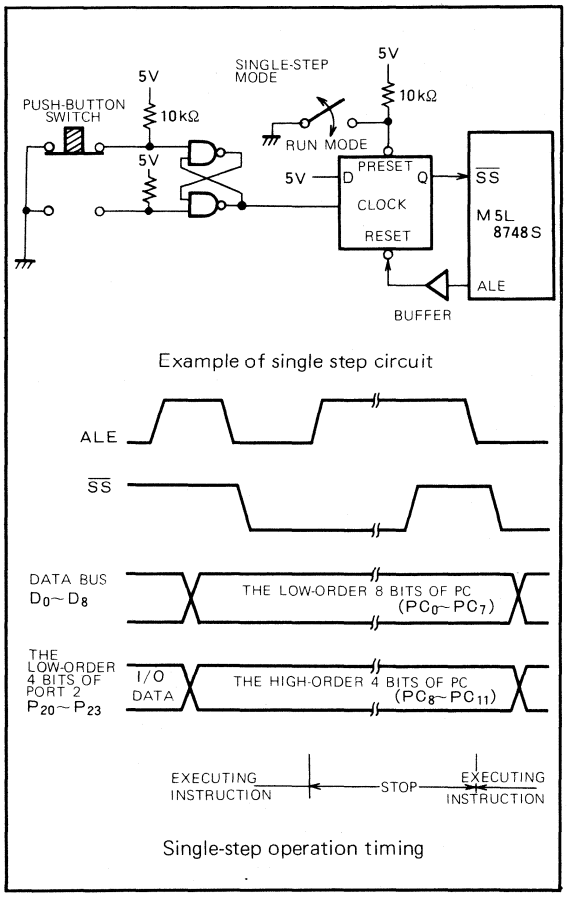


Fig. 11 Single-step operation circuit and timing

A type D flip-flop with preset and reset terminals, as shown in Fig. 11, is used to generate the signal for $\overline{SS}$. When the preset terminal goes to low-level, $\overline{SS}$ goes to high-level, which puts the CPU in RUN mode. When the preset terminal is grounded it goes to high-level. Then $\overline{SS}$ goes to low-level. When $\overline{SS}$ goes to low-level, the CPU stops. Then when the push-button switch is pushed, a pulse is sent to the clock terminal of the type D flip-flop which turns $\overline{SS}$ to high-level. When $\overline{SS}$ goes to high-level the CPU fetches the next instruction and begins to execute it, but then an ALE signal is sent to the reset terminal of the type D flip-flop which turns $\overline{SS}$ to low-level. The CPU again stops as soon as execution of the current instruction is completed. When the push-button switch is again pushed, the cycle is repeated and the CPU is in single-step operation as shown in Fig. 12. While the CPU is stopped in single-step operation, the data bus and the low-order 4 bits of port 2 are used to output the memory address of the next instruction to be fetched. This interferes with input and output, but essential input/output can be latched by using the rising edge of ALE as clock.

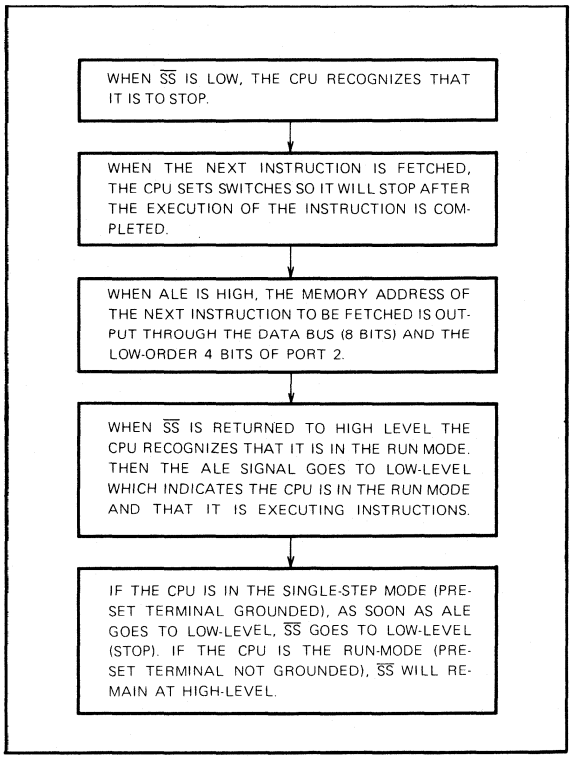


Fig. 12 CPU operation in single-step mode

Central Processing Unit (CPU)

Central Processing Unit (CPU) is composed of an 8-bit parallel arithmetic unit, accumulator, flag flip-flop and instruction decoder. The 8-bit parallel arithmetic unit has circuitry to perform the four basic arithmetic operations (plus, minus, multiply and divide) as well as logical operations such as AND and OR. The flag flip-flop is used to indicate status such as carry and zero. The accumulator contains one of the operations and the result is usually retained in the accumulator.

MACHINE INSTRUCTIONS

Item / Type	Mnemonic	Instruction code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Hexa-decimal	Bytes	Cycles	Function	Effected carry C	Effected carry AC	Effected carry Note	Description
Transfer	**MOV A, #n**	0 0 1 0 0 0 1 1 n_7 n_6 n_5 n_4 n_3 n_2 n_1 n_0	23 n	2	2	(A)←n				Transfers data n to register A.
	MOV A, PSW	1 1 0 0 0 1 1 1	C7	1	1	(A)←(PSW)				Transfers the contents of the program status word to register A.
	MOV A, Rr	1 1 1 1 1 r_2 r_1 r_0	F8 + r	1	1	(A)←(Rr) r = 0 ~ 7				Transfers the contents of register R_r to register A.
	MOV A, @Rr	1 1 1 1 0 0 0 r_0	F0 + r	1	1	(A)←(M(Rr)) r = 0 ~ 1				Transfers the contents of memory location, of the current page, whose address is in register R_r to register A.
	MOV PSW, A	1 1 0 1 0 1 1 1	D7	1	1	(PSW)←(A) (C)←(A_7), (AC)←(A_6)	○	○		Transfers the contents of register A to the program status word.
	MOV STS, A (Note 3)	1 0 0 1 0 0 0 0	90	1	1	(STS)←(A) (ST_4 ~ ST_7)←(A_4 ~ A_7)			4	Transfers the contents of register A to the system status register.
	MOV Rr, A	1 0 1 0 1 r_2 r_1 r_0	A8 + r	1	1	(Rr)←(A) r = 0 ~ 7				Transfers the contents of register A to register R_r.
	MOV Rr, #n	1 0 1 1 1 r_2 r_1 r_0 n_7 n_6 n_5 n_4 n_3 n_2 n_1 n_0	B8 + r n	2	2	(Rr)←n r = 0 ~ 7				Transfers data n to register R_r.
	MOV @Rr, A	1 0 1 0 0 0 0 r_0	A0 + r	1	1	(M(Rr))←(A) r = 0 ~ 1				Transfers the contents of register A to memory location, of the current page, whose address is in register R_r.
	MOV @Rr, #n	1 0 1 1 0 0 0 r_0 n_7 n_6 n_5 n_4 n_3 n_2 n_1 n_0	B0 + r n	2	2	(M(Rr))←n r = 0 ~ 1				Transfers data n to memory location, of the current page, whose address is in register R_r.
	MOVP A, @A	1 0 1 0 0 0 1 1	A3	1	2	(A)←(M(A))				Transfers the data of memory location, of the current page, whose address is in register A to register A.
	MOVP3 A, @A	1 1 1 0 0 0 1 1	E3	1	2	(A)←(M(page 3, A))				Transfers the data of memory location, of page 3, whose address is in register A to register A.
	MOVX @Rr, A (Note 2)	1 0 0 1 0 0 0 r_0	90 + r	1	2	(Mx(Rr))←(A) r = 0 ~ 1				Transfers the contents of register A to memory location, of the current page, whose address is in register R_r.
	MOVX A, @Rr (Note 2)	1 0 0 0 0 0 0 r_0	80 + r	1	2	(A)←(Mx(Rr)) r = 0 ~ 1				Transfers the contents of memory location, of the current page, whose address is in register R_r to register A.
	XCH A, Rr	0 0 1 0 1 r_2 r_1 r_0	28 + r	1	1	(A)⟷(Rr) r = 0 ~ 7				Exchanges the contents of register R_r with the contents of register A.
	XCH A, @Rr	0 0 1 0 0 0 0 r_0	20 + r	1	1	(A)⟷(M(Rr)) r = 0 ~ 1				Exchanges the contents of memory location, of the current page, whose address is in register R_r with the contents of register A.
	XCHD A, @Rr	0 0 1 1 0 0 0 r_0	30 + r	1	1	(A_0 ~ A_3)⟷(M(Rr_0 ~ Rr_3)) r = 0 ~ 1				Exchanges the contents of the low-order four bits of register A with the low-order four bits of memory location, of the current page, whose address is in register R_r.
Arithmetic	**ADD A, #n**	0 0 0 0 0 0 1 1 n_7 n_6 n_5 n_4 n_3 n_2 n_1 n_0	03 n	2	2	(A)←(A)+n	○	○	1	Adds data n to the contents of register A and sets the carry flags to 1 if there is an overflow otherwise resets the carry flags to 0. The result is stored in register A.
	ADD A, Rr	0 1 1 0 1 r_2 r_1 r_0	68 + r	1	1	(A)←(A)+(Rr) r = 0 ~ 7	○	○	1	Adds the contents of register R_r to the contents of register A and sets the carry flags to 1 if there is an overflow otherwise resets the carry flags to 0. The result is stored in register A.
	ADD A, @Rr	0 1 1 0 0 0 0 r_0	60 + r	1	1	(A)←(A)+(M(Rr)) r = 0 ~ 1	○	○	1	Adds the contents of register A and the contents of memory location, of the current page, whose address is in register A and sets the carry flags to 1 if there is an overflow otherwise resets the carry flags to 0. The result is stored in register A.
	ADDC A, #n	0 0 0 1 0 0 1 1 n_7 n_6 n_5 n_4 n_3 n_2 n_1 n_0	13 n	2	2	(A)←(A)+n+(C)	○	○	1	Adds the carry and data n to the contents of register A and sets the carry flags to 1 if there is an overflow otherwise resets the carry flags to 0. The result is stored in register A.
	ADDC A, Rr	0 1 1 1 1 r_2 r_1 r_0	78 + r	1	1	(A)←(A)+(Rr)+(C) r = 0 ~ 7	○	○	1	Adds the carry and the contents of register R_r to the contents of register A and sets the carry flags to 1 if there is an overflow otherwise resets the carry flags to 0. The result is stored in register A.
	ADDC A, @Rr	0 1 1 1 0 0 0 r_0	70 + r	1	1	(A)←(A)+(M(Rr))+(C) r = 0 ~ 1	○	○	1	Adds the carry and the contents of memory location, of the current page, whose address is in register R_r to the contents of register A and sets the carry flags to 1 if there is an overflow otherwise resets the carry flags to 0. The result is stored in register A.

7

FUNCTION OF MELPS 8-48 MICROCOMPUTERS

Item / Type	Mnemonic	Instruction code $D_7 D_6 D_5 D_4$ $D_3 D_2 D_1 D_0$	Hexa-decimal	Bytes	Cycles	Function	Effected carry C	Effected carry AC	Effected carry Note	Description
Arithmetic	**ANL A, #n**	0 1 0 1 0 0 1 1 $n_7 n_6 n_5 n_4$ $n_3 n_2 n_1 n_0$	5 3 n	2	2	(A) ← (A) ∧ n				The logical product of the contents of register A and data n, is stored in register A.
	ANL A, Rr	0 1 0 1 1 $r_2 r_1 r_0$	5 8 + r	1	1	(A) ← (A) ∧ (Rr) r = 0 ~ 7				The logical product of the contents of register A and the contents of register R_r, is stored in register A.
	ANL A, @Rr	0 1 0 1 0 0 0 r_0	5 0 + r	1	1	(A) ← (A) ∧ (M(Rr)) r = 0 ~ 1				The logical product of the contents of register A and the contents of memory location, of the current page, whose address is in register R_r, is stored in register A.
	ORL A, #n	0 1 0 0 0 0 1 1 $n_7 n_6 n_5 n_4$ $n_3 n_2 n_1 n_0$	4 3 n	2	2	(A) ← (A) ∨ n				The logical sum of the contents of register A and data n, is stored in register A.
	ORL A, Rr	0 1 0 0 1 $r_2 r_1 r_0$	4 8 + r	1	1	(A) ← (A) ∨ (Rr) r = 0 ~ 7				The logical sum of the contents of register A and the contents of register R_r is stored in register A.
	ORL A, @Rr	0 1 0 0 0 0 0 r_0	4 0 + r	1	1	(A) ← (A) ∨ (M(Rr)) r = 0 ~ 1				The logical sum of the contents of register A and the contents of memory location, of the current page, whose address is in register R_r, is stored in register A.
	XRL A, #n	1 1 0 1 0 0 1 1 $n_7 n_6 n_5 n_4$ $n_3 n_2 n_1 n_0$	D 3 n	2	2	(A) ← (A) ∀ n				The exclusive OR of the contents of register A and data n, is stored in register A.
	XRL A, Rr	1 1 0 1 1 $r_2 r_1 r_0$	D 8 + r	1	1	(A) ← (A) ∀ (Rr) r = 1 ~ 7				The exclusive OR of the contents of register A and the contents of register R_r is stored in register A.
	XRL A, @Rr	1 1 0 1 0 0 0 r_0	D 0 + r	1	1	(A) ← (A) ∀ (M(Rr)) r = 0 ~ 1				The exclusive OR of the contents of register A and the contents of memory location, of the current page, whose address is in register R_r, is stored in register A.
	INC A	0 0 0 1 0 1 1 1	1 7	1	1	(A) ← (A) + 1				Increments the contents of register A by 1. The result is stored in register A, and the carries are unchanged.
	DEC A	0 0 0 0 0 1 1 1	0 7	1	1	(A) ← (A) − 1				Decrements the contents of register A by 1. The result is stored in register A, and the carries are unchanged.
	CLR A	0 0 1 0 0 1 1 1	2 7	1	1	(A) ← 0				Clears the contents of register A, resets to 0.
	CPL A	0 0 1 1 0 1 1 1	3 7	1	1	(A) ← $(\overline{A})$				Forms 1's complement of register A, and stores it in register A.
	DA A	0 1 0 1 0 1 1 1	5 7	1	1	(A) ← (A) 10 Hexadecimal	○	○	1	The contents of register A is converted to binary coded decimal notion, and it is stored in register A. If the contents of register A are more than 99 the carry flags are set to 1 otherwise they are reset to 0.
	SWAP A	0 1 0 0 0 1 1 1	4 7	1	1	$(A_4 \sim A_7) \longleftrightarrow (A_0 \sim A_3)$				Exchanges the contents of bits 0~3 of register A with the contents of bits 4~7 of register A.
	RL A	1 1 1 0 0 1 1 1	E 7	1	1	$(A_{n+1}) \leftarrow (A_n)$ $(A_0) \leftarrow (A_7)$ n = 0 ~ 6				Shifts the contents of register A left one bit. A_7 the MSB is rotated to A_0 the LSB.
	RLC A	1 1 1 1 0 1 1 1	F 7	1	1	$(A_{n+1}) \leftarrow (A_n)$ $(A_0) \leftarrow (C)$ $(C) \leftarrow (A_7)$ n = 0 ~ 6	○			Shifts the contents of register A left one bit. A_7 the MSB is shifted to the carry flag and the carry flag is shifted to A_0 the LSB.
	RR A	0 1 1 1 0 1 1 1	7 7	1	1	$(A_n) \leftarrow (A_{n+1})$ $(A_7) \leftarrow (A_0)$ n = 0 ~ 6				Shifts the contents of register A right one bit. A_0 the LSB is rotated to A_7 the MSB.
	RRC A	0 1 1 0 0 1 1 1	6 7	1	1	$(A_n) \leftarrow (A_{n+1})$ $(A_7) \leftarrow (C)$ $(C) \leftarrow (A_0)$ n = 0 ~ 6	○			Shifts the contents of register A right one bit. A_0 the LSB is shifted to the carry flag and the carry flag is shifted to A_7 the MSB.
Register arithmetic	**INC Rr**	0 0 0 1 1 $r_2 r_1 r_0$	1 8 + r	1	1	(Rr) ← (Rr) + 1 r = 0 ~ 7				Increments the contents of register R_r by 1. The result is stored in register R_r and the carries are unchanged.
	INC @Rr	0 0 0 1 0 0 0 r_0	1 0 + r	1	1	(M(Rr)) ← (M(Rr)) + 1 r = 0 ~ 1				Increments the contents of the memory location, of the current page, whose address is in register R_r by 1. Register R_r uses bit 0~5.
	DEC Rr	1 1 0 0 1 $r_2 r_1 r_0$	C 8 + r	1	1	(Rr) ← (Rr) − 1 r = 0 ~ 7				Decrements the contents of register R_r by 1. The result is stored in register R_r and the carries are unchanged.

FUNCTION OF MELPS 8-48 MICROCOMPUTERS

Item / Type	Mnemonic	Instruction code		Bytes	Cycles	Function	Effected carry			Description
		$D_7 D_6 D_5 D_4$	$D_3 D_2 D_1 D_0$ Hexa-decimal				C	AC	Note	
Jump	**JBb m**	$b_7 b_6 b_5 1$ / $m_7 m_6 m_5 m_4$	0 0 1 0 / $m_3 m_2 m_1 m_0$ — 12 + b×2 / m	2	2	$(A_b)=1$ then $(PC_0 \sim PC_7) \leftarrow m$ $(A_b)=0$ then $(PC) \leftarrow (PC)+2$ $b_7 b_6 b_5 = 0 \sim 7$				Jumps to address m of the current page when bit b of register A is 1. Executes the next instruction when bit b of register A is 0.
	JNIBF m (Note 3)	1 1 0 1 / $m_7 m_6 m_5 m_4$	0 1 1 0 / $m_3 m_2 m_1 m_0$ — D6 / m	2	2	(IBF) = 0 then $(PC_0 \sim PC_7) \leftarrow m$				Jumps to address m of the current page when IBF is 0, otherwise the next instruction is executed.
	JOBF m (Note 3)	1 0 0 0 / $m_7 m_6 m_5 m_4$	0 1 1 0 / $m_3 m_2 m_1 m_0$ — 86 / m	2	2	(OBF) = 1 then $(PC_0 \sim PC_7) \leftarrow m$				Jumps to address m of the current page when OBF is 0, otherwise the next instruction is executed.
	JTF m	0 0 0 1 / $m_7 m_6 m_5 m_4$	0 1 1 0 / $m_3 m_2 m_1 m_0$ — 16 / m	2	2	(TF) = 1 then $(PC_0 \sim PC_7) \leftarrow m$ (TF) = 0 then $(PC) \leftarrow (PC)+2$				Jumps to address m of the current page when the overflow flag of the timer is 1 otherwise the next instruction is executed. Flag is cleared after executing.
	JNI m (Note 3)	1 0 0 0 / $m_7 m_6 m_5 m_4$	0 1 1 0 / $m_3 m_2 m_1 m_0$ — 86 / m	2	2	(INT) = 0 then $(PC_0 \sim PC_7) \leftarrow m$ (INT) = 1 then $(PC) \leftarrow (PC)+2$				Jumps to address m of the current page when external interrupt terminal is low-level, otherwise the next instruction is executed.
	JMP m	$m_{10} m_9 m_8$ 0 / $m_7 m_6 m_5 m_4$	0 1 0 0 / $m_3 m_2 m_1 m_0$ — 04 + $(m_8 \sim m_{10})$ ×2 / m	2	2	$(PC_8 \sim PC_{10}) \leftarrow m_8 \sim m_{10}$ $(PC_0 \sim PC_7) \leftarrow m_0 \sim m_7$ $(PC_{11}) \leftarrow (MBF)$				Jumps to address m on page m_{10} m_9 m_8 in the memory bank indicated by MBF.
	JMPP @A	1 0 1 1 / 0 0 1 1	B 3	1	2	$(PC_0 \sim PC_7) \leftarrow (M(A))$				Jumps to the memory location, of the current page, whose address is in register A. But when the instruction executed was in address 255, jumps to next page.
	DJNZ Rr, m	1 1 1 0 / 1 r_2 r_1 r_0	E 8 + r / m	2	2	$(Rr) \leftarrow (Rr)-1$ r = 0 ~ 7 $(Rr) \neq 0$ then $(PC_0 \sim PC_7) \leftarrow m$ $(Rr) = 0$ then $(PC) \leftarrow (PC)+2$				Decrements the contents of register R_r by 1. Jumps to address m of the current page when the result is not 0, otherwise the next instruction is executed.
	JC m	1 1 1 1 / $m_7 m_6 m_5 m_4$	0 1 1 0 / $m_3 m_2 m_1 m_0$ — F6 / m	2	2	(C) = 1 then $(PC_0 \sim PC_7) \leftarrow m$ (C) = 0 then $(PC) \leftarrow (PC)+2$				Jumps to address m of the current page if the carry flag C is 1, otherwise the next instruction is executed.
	JNC m	1 1 1 0 / $m_7 m_6 m_5 m_4$	0 1 1 0 / $m_3 m_2 m_1 m_0$ — E6 / m	2	2	(C) = 0 then $(PC_0 \sim PC_7) \leftarrow m$ (C) = 1 then $(PC) \leftarrow (PC)+2$				Jumps to address m of the current page if the carry flag C is 0, otherwise the next instruction is executed.
	JZ m	1 1 0 0 / $m_7 m_6 m_5 m_4$	0 1 1 0 / $m_3 m_2 m_1 m_0$ — C6 / m	2	2	(A) = 0 then $(PC_0 \sim PC_7) \leftarrow m$ (A) ≠ 0 then $(PC) \leftarrow (PC)+2$				Jumps to address m of the current page when the contents of register A are 0, otherwise the next instruction is executed.
	JNZ m	1 0 0 1 / $m_7 m_6 m_5 m_4$	0 1 1 0 / $m_3 m_2 m_1 m_0$ — 96 / m	2	2	(A) ≠ 0 then $(PC_0 \sim PC_7) \leftarrow m$ (A) = 0 then $(PC) \leftarrow (PC)+2$				Jumps to address m of the current page when the contents of register A are not 0, otherwise the next instruction is executed.
	JT0 m	0 0 1 1 / $m_7 m_6 m_5 m_4$	0 1 1 0 / $m_3 m_2 m_1 m_0$ — 36 / m	2	2	$(T_0) = 1$ then $(PC_0 \sim PC_7) \leftarrow m$ $(T_0) = 0$ then $(PC) \leftarrow (PC)+2$				Jumps to address m of the current page when flag T_0 is 1 otherwise the next instruction is executed.
	JNT0 m	0 0 1 0 / $m_7 m_6 m_5 m_4$	0 1 1 0 / $m_3 m_2 m_1 m_0$ — 26 / m	2	2	$(T_0) = 0$ then $(PC_0 \sim PC_7) \leftarrow m$ $(T_0) = 1$ then $(PC) \leftarrow (PC)+2$				Jumps to address m of the current page when flag T_0 is 0, otherwise the next instruction is executed.
	JT1 m	0 1 0 1 / $m_7 m_6 m_5 m_4$	0 1 1 0 / $m_3 m_2 m_1 m_0$ — 56 / m	2	2	$(T_1) = 1$ then $(PC_0 \sim PC_7) \leftarrow m$ $(T_1) = 0$ then $(PC) \leftarrow (PC)+2$				Jumps to address m of the current page when flag T_1 is 1. otherwise the next instruction is executed.
	JNT1 m	0 1 0 0 / $m_7 m_6 m_5 m_4$	0 1 1 0 / $m_3 m_2 m_1 m_0$ — 46 / m	2	2	$(T_1) = 0$ then $(PC_0 \sim PC_7) \leftarrow m$ $(T_1) = 1$ then $(PC) \leftarrow (PC)+2$				Jumps to address m of the current page when flag T_1 is 0. otherwise the next instruction is executed.
	JF0 m	1 0 1 1 / $m_7 m_6 m_5 m_4$	0 1 1 0 / $m_3 m_2 m_1 m_0$ — B6 / m	2	2	$(F_0) = 1$ then $(PC_0 \sim PC_7) \leftarrow m$ $(F_0) = 0$ then $(PC) \leftarrow (PC)+2$				Jumps to address m of the current page when flag F_0 is 1.
	JF1 m	0 1 1 1 / $m_7 m_6 m_5 m_4$	0 1 1 0 / $m_3 m_2 m_1 m_0$ — 76 / m	2	2	$(F_1) = 1$ then $(PC_0 \sim PC_7) \leftarrow m$ $(F_1) = 0$ then $(PC) \leftarrow (PC)+2$				Jumps to address m of the current page when flag F_1 is 1.
Flag control	**CLR C**	1 0 0 1	0 1 1 1 — 97	1	1	$(C) \leftarrow 0$	○			Clears the carry flag C, resets it to 0. AC is not affected.
	CPL C	1 0 1 0	0 1 1 1 — A7	1	1	$(C) \leftarrow (\overline{C})$	○			Complements the carry flag C. AC is not affected.
	CLR F_0	1 0 0 0	0 1 0 1 — 85	1	1	$(F_0) \leftarrow 0$				Clears the flag F_0, resets it to 0.
	CPL F_0	1 0 0 1	0 1 0 1 — 95	1	1	$(F_0) \leftarrow (\overline{F_0})$				Complements the flag F_0.
	CLR F_1	1 0 1 0	0 1 0 1 — A5	1	1	$(F_1) \leftarrow 0$				Clears flag F_1 resets it to 0.
	CPL F_1	1 0 1 1	0 1 0 1 — B5	1	1	$(F_1) \leftarrow (\overline{F_1})$				Complements the flag F_1.

7

FUNCTION OF MELPS 8-48 MICROCOMPUTERS

Item / Type	Mnemonic	Instruction code $D_7 D_6 D_5 D_4$	Instruction code $D_3 D_2 D_1 D_0$	Hexa-decimal	Bytes	Cycles	Function	Effected carry C	Effected carry AC	Effected carry Note	Description
Subroutine call	**CALL m**	$m_{10} m_9 m_8$ 1	0 1 0 0 $m_7 m_6 m_5 m_4$	14 + $(m_8 - m_{10}) \times 2$ $m_3 m_2 m_1 m_0$ m	2	2	((SP)) ← (PC) (PSW4 ~ PSW7) (SP) ← (SP) + 1 (PC_{0-10}) ← m (PC_{11}) ← MBF				Calls subroutine from address m. The program counter and the 4 high-order bits of the PSW are stored in the address indicated by the stack pointer (SP). The SP is incremented by 1 and m is transferred to PC_0 ~ PC_{10} and the MBF is transferred to PC_{11}.
	RET	1 0 0 0	0 0 1 1	83	1	2	(SP) ← (SP) − 1 (PC) ← ((SP))				The SP is decremented by 1. The program counter is restored to the saved setting in the stack indicated by the stack pointer. The PSW is not changed and interrupt disabled is maintained.
	RETR	1 0 0 1	0 0 1 1	93	1	2	(SP) ← (SP) − 1 (PC) (PSW4 ~ PSW7) ← ((SP))				The SP is decremented by 1. The program counter and the 4 high-order bits of the PSW are restored with the saved data in the stack indicated by the stack pointer. The interrupt becomes enabled after the execution is completed.
Input/Output	**IN A, Pp**	0 0 0 0	1 0 $p_1 p_0$	08 + p	1	2	(A) ← (Pp) p = 1 ~ 2				Loads the contents of P_p to register A.
	OUTL Pp, A	0 0 1 1	1 0 $p_1 p_0$	38 + p	1	2	(Pp) ← (A) p = 1 ~ 2				Output latches the contents of register A to P_p
	ANL Pp, #n	1 0 0 1 $n_7 n_6 n_5 n_4$	1 0 $p_1 p_0$ $n_3 n_2 n_1 n_0$	98 + p n	2	2	(Pp) ← (Pp) ∧ n p = 1 ~ 2				Logical ANDs the contents of P_p and data n. Outputs the result to P_p
	ORL Pp, #n	1 0 0 0 $n_7 n_6 n_5 n_4$	1 0 $p_1 p_0$ $n_3 n_2 n_1 n_0$	88 + p n	2	2	(Pp) ← (Pp) ∨ n p = 1 ~ 2				Logical ORs the contents of P_p and data n. Outputs the result to P_p
	INS A, BUS (Note 2)	0 0 0 0	1 0 0 0	08	1	2	(A) ← (BUS)				Enters the contents of data bus (port 0) to register A
	OUTL BUS, A (Note 2)	0 0 0 0	0 0 1 0	02	1	2	(BUS) ← (A)				Output latches the contents of register A data to data bus (port 0)
	ANL BUS, #n (Note 2)	1 0 0 1 $n_7 n_6 n_5 n_4$	1 0 0 0 $n_3 n_2 n_1 n_0$	98 n	2	2	(BUS) ← (BUS) ∧ n				Logical ANDs the contents of data bus (port 0) and data n. Outputs the result to data bus (port 0)
	ORL BUS, #n (Note 2)	1 0 0 0 $n_7 n_6 n_5 n_4$	1 0 0 0 $n_3 n_2 n_1 n_0$	88 n	2	2	(BUS) ← (BUS) ∨ n				Logical ORs the contents of data bus (port 0) and data n. Outputs the result to data bus (port 0)
	IN A, DBB (Note 3)	0 0 1 0	0 0 1 0	22	1	1	(A) ← (DBB)				Enters the contents of data bus buffer (DBB) in register A.
	OUT DBB, A (Note 3)	0 0 0 0	0 0 1 0	02	1	1	(DBB) ← (A)				Outputs the contents of register A to data bus buffer (DBB). 0BF is set.
	MOVD A, Pp	0 0 0 0	1 1 $p_1 p_0$	0C + $p_1 p_0$	1	2	(A_0 ~ A_3) ← (Pp_0 ~ Pp_3) (A_4 ~ A_7) ← 0 p = 4 ~ 7				Inputs the contents of P_p to the low-order 4 bits of register A and inputs 0 to the high-order 4 bits of register A.
	MOVD Pp, A	0 0 1 1	1 1 $p_1 p_0$	3C + $p_1 p_0$	1	2	(Pp_0 ~ Pp_3) ← (A_0 ~ A_3) p = 4 ~ 7				Outputs the low-order 4 bits of register A to P_p.
	ANLD Pp, A	1 0 0 1	1 1 $p_1 p_0$	9C + $p_1 p_0$	1	2	(Pp_0 ~ Pp_3) ← (Pp_0 ~ Pp_3) ∧ (A_0 ~ A_3) p = 4 ~ 7				Logical ANDs the 4 low-order bits of register A and the contents of P_p. P_p contains the result.
	ORLD Pp, A	1 0 0 0	1 1 $p_1 p_0$	8C + $p_1 p_0$	1	2	(Pp_0 ~ Pp_3) ← (Pp_0 ~ Pp_3) ∨ (A_0 ~ A_3) p = 4 ~ 7				Logical ORs the 4 low-order bits of register A and the contents of P_p. P_p contains the result.

P_p's used for multiplying 8243 ports are P_4 ~ P_7.
Correspondence to p_2, p_1 is shown below.
P4 ··· $p_1 p_2$ = 00
P5 ··· $p_1 p_2$ = 01
P6 ··· $p_1 p_2$ = 10
P7 ··· $p_1 p_2$ = 11

FUNCTION OF MELPS 8-48 MICROCOMPUTERS

Type \ Item	Mnemonic	Instruction code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Hexa-decimal	Bytes	Cycles	Function	Effected carry C	AC	Note	Description
Control	**EN I**	0 0 0 0 0 1 0 1	05	1	1	(INTF) ← 1				Enables outside interrupt.
	DIS I	0 0 0 1 0 1 0 1	15	1	1	(INTF) ← 0				Disables outside interrupt.
	SEL RB_0	1 1 0 0 0 1 0 1	C5	1	1	(BS) ← 0				Selects working register bank 0.
	SEL RB_1	1 1 0 1 0 1 0 1	D5	1	1	(BS) ← 1				Selects working register bank 1.
	SEL MB_0 (Note 2)	1 1 1 0 0 1 0 1	E5	1	1	(MBF) ← 0				Selects memory bank 0.
	SEL MB_1 (Note 2)	1 1 1 1 0 1 0 1	F5	1	1	(MBF) ← 1				Selects memory bank 1.
	ENTO CLK (Note 2)	0 1 1 1 0 1 0 1	75	1	1					Enables output of clock signal from terminal T_0
	EN DMA (Note 3)	1 1 1 0 0 1 0 1	E5	1	1					Enables DMA hand shake lines.
	EN FLAGS (Note 3)	1 1 1 1 0 1 0 1	F5	1	1	($P2_4$) ← (OBF) ($P2_5$) ← (IBF)				Enables interrupts from master.
Timer/event counter control	**MOV A, T**	0 1 0 0 0 0 1 0	42	1	1	(A) ← (T)				Transfers the contents of timer/event counter to register A.
	MOV T, A	0 1 1 0 0 0 1 0	62	1	1	(T) ← (A)				Transfers the contents of register A to timer/event counter.
	STRT T	0 1 0 1 0 1 0 1	55	1	1					Starts timer operation of timer/event counterm. Minimum count cycle is 80μs.
	STRT CNT	0 1 0 0 0 1 0 1	45	1	1					Starts operation as event counter of time/event counter. Counts up when terminated T_1 changes to input high-level for input low-level. Minimum count cycle is 7.5μs.
	STOP TCNT	0 1 1 0 0 1 0 1	65	1	1					Stops operation of timer or event counter.
	EN TCNTI	0 0 1 0 0 1 0 1	25	1	1	(TCNTF) ← 1				Enables interrupt of timer/event counter.
	DIS TCNTI	0 0 1 1 0 1 0 1	35	1	1	(TCNTF) ← 0				Disables interrupt of timer/event counter. Resets interrupt flip-flop of CPU which is set during the CPU stands-by. Timer overflow flag isn't affected.
Misc.	**NOP**	0 0 0 0 0 0 0 0	00	1	1					No operation. Execution time is 1 cycle.

Note 1: Executing an instruction may produce a carry (overflow or underflow). The carry may be disregarded (lost) or it may be transferred to C/AC (saved). The saving of a carry is not shown in the function equations, but is instead shown in the carry columns C and AC. The detail affection of carries for instructions ADD ADDC and DA is as follows:

(C) ← 1 at overflow of the accumulator is produced.
(C) ← 0 at no overflow of the accumulator is produced.
(AC) ← 1 at overflow of the bit 3 of the accumulator.
(AC) ← 0 at no overflow.

2: These instructions are available in M5L8039P, M5L8048-XXXP, M5L8049-XXXP and M5L8748S.

3: These instructions are available only in M5L8041A-XXXP.

4: The contents of ST_4~ST_7 is read when the host computer reads the status of M5L8041A-XXXP.

7

FUNCTION OF MELPS 8-48 MICROCOMPUTERS

Symbol	Details	Symbol	Details
A	8 bit register (accumlator)	PC	Program counter
A_0~ A_3	The low-order 4 bits of the register A	PC_0~PC_7	The low-order 8 bits of the program counter
A_4~ A_7	The high-order 4 bits of the register A	PC_8~PC_{10}	The high-order 3 bits of the program counter
A_0~An, A_{n+1}	The bits of the register A	PSW	Program status word
b	The value of the bits 5~7 of the first byte machine code		
$b_7b_6b_5$	The bits 5~7 of the first byte machine code	Rr	Register designator
BS	Register bank select	r	Register number
BUS	Corresponds to the port 0 (bus I/O port)	r_0	The value of bit 0 of the machine code
		$r_2r_1r_0$	The value of bits 0~2 of the machine code
AC	Auxiliary carry flag	$s_2s_1s_0$	The value of bits 0~2 of the stack pointer
C	Carry flag	SP	Stack pointer
DBB	Data bus buffer	ST_4~ST_7	Bits 4~7 of the status register
		STS	System status
F_0	Flag 0	T	Timer/event counter
F_1	Flag 1	T_0	Test pin 0
INTF	Interrupt flag	T_1	Test pin 1
IBF	Input buffer full flag	TCNTF	Timer/event counter overflow interrupt flag
m	The value of the 11-bit address	TF	Timer flag
$m_7m_6m_5m_4m_3m_2m_1m_0$	The second byte (low-order 8 bits) machine code of the 11-bit address		
$m_{10}m_9$ m_8	The bits 5~7 of the first byte (high-order 3 bits) machine code of the 11-bit address	#	Symbol to indicate the immediate data
(M (A))	The content of the memory location addressed by the register A	@	Symbol to indicate the content of the memory location addressed by the register
(M (Rr))	The content of the memory location addressed by the register Rr		
(Mx(Rr))	The content of the external memory location addressed by the register Rr	←	Shows direction of data flow
MBF	Memory bank flag	↔	Exchanges the contents of data
n	The value of the immediate data	( )	Contents of register, memory location or flag
$n_7n_6n_5n_4n_3n_2n_1n_0$	The immediate data of the second byte machine code	∧	Logical AND
OBF	Output buffer full flag	∨	Inclusive OR
		∀	Exclusive OR
p	Port number	—	Negation
Pp	Port designator	○	Content of flag is set or reset after execution
p_1p_0	The bits of the machine code corresponding to the port number		

MITSUBISHI ELECTRIC

Instruction Code List (M5L8039P, M5L8048-XXXP, M5L8049-XXXP, M5L8748S)

D_3~D_0 \ D_7~D_4	Hexa-decimal	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	0	NOP	INC @R0	XCH A, @R0	XCHD A, @R0	ORL A, @R0	ANL A, @R0	ADD A, @R0	ADDC A, @R0	MOVX A, @R0	MOVX @R0, A	MOV @R0, A	MOV @R0, #n		XRL A, @R0		MOV A, @R0
0001	1		INC @R1	XCH A, @R1	XCHD A, @R1	ORL A, @R1	ANL A, @R1	ADD A, @R1	ADDC A, @R1	MOVX A, @R1	MOVX @R1, A	MOV @R1, A	MOV @R1, #n		XRL A, @R1		MOV A, @R1
0010	2	OUTL BUS, A	JB0 m		JB1 m	MOV A, T	JB2 m	MOV T, A	JB3 m		JB4 m		JB5 m		JB6 m		JB7 m
0011	3	ADD A, #n	ADDC A, #n	MOV A, #n		ORL A, #n	ANL A, #n			RET	RETR	MOVP A, @A	JMPP @A		XRL A, #n	MOVP3 A, @A	
0100	4	JMP 0XX	CALL 0XX	JMP 1XX	CALL 1XX	JMP 2XX	CALL 2XX	JMP 3XX	CALL 3XX	JMP 4XX	CALL 4XX	JMP 5XX	CALL 5XX	JMP 6XX	CALL 6XX	JMP 7XX	CALL 7XX
0101	5	EN I	DIS I	EN TCNTI	DIS TCNTI	STRT CNT	STRT T	STOP TCNT	ENTO CLK	CLR F0	CPL F0	CLR F1	CPL F1	SEL RB0	SEL RB1	SEL MB0	SEL MB1
0110	6		JTF m	JNT0 m	JT0 m	JNT1 m	JT1 m		JF1 m	JNI m	JNZ m		JF0 m	JZ m		JNC m	JC m
0111	7	DEC A	INC A	CLR A	CPL A	SWAP A	DA A	RRC A	RR A		CLR C	CPL C		MOV A, PSW	MOV PSW, A	RL A	RLC A
1000	8	INS A, BUS	INC R0	XCH A, R0		ORL A, R0	ANL A, R0	ADD A, R0	ADDC A, R0	ORL BUS, #n	ANL BUS, #n	MOV R0, A	MOV R0, #n	DEC R0	XRL A, R0	DJNZ R0, m	MOV A, R0
1001	9	IN A, P1	INC R1	XCH A, R1	OUTL P1, A	ORL A, R1	ANL A, R1	ADD A, R1	ADDC A, R1	ORL P1, #n	ANL P1, #n	MOV R1, A	MOV R1, #n	DEC R1	XRL A, R1	DJNZ R1, m	MOV A, R1
1010	A	IN A, P2	INC R2	XCH A, R2	OUTL P2, A	ORL A, R2	ANL A, R2	ADD A, R2	ADDC A, R2	ORL P2, #n	ANL P2, #n	MOV R2, A	MOV R2, #n	DEC R2	XRL A, R2	DJNZ R2, m	MOV A, R2
1011	B		INC R3	XCH A, R3		ORL A, R3	ANL A, R3	ADD A, R3	ADDC A, R3			MOV R3, A	MOV R3, #n	DEC R3	XRL A, R3	DJNZ R3, m	MOV A, R3
1100	C	MOVD A, P4	INC R4	XCH A, R4	MOVD P4, A	ORL A, R4	ANL A, R4	ADD A, R4	ADDC A, R4	ORLD P4, A	ANLD P4, A	MOV R4, A	MOV R4, #n	DEC R4	XRL A, R4	DJNZ R4, m	MOV A, R4
1101	D	MOVD A, P5	INC R5	XCH A, R5	MOVD P5, A	ORL A, R5	ANL A, R5	ADD A, R5	ADDC A, R5	ORLD P5, A	ANLD P5, A	MOV R5, A	MOV R5, #n	DEC R5	XRL A, R5	DJNZ R5, m	MOV A, R5
1110	E	MOVD A, P6	INC R6	XCH A, R6	MOVD P6, A	ORL A, R6	ANL A, R6	ADD A, R6	ADDC A, R6	ORLD P6, A	ANLD R6, A	MOV P6, A	MOV R6, #n	DEC R6	XRL A, R6	DJNZ R6, m	MOV A, R6
1111	F	MOVD A, P7	INC R7	XCH A, R7	MOVD P7, A	ORL A, R7	ANL A, R7	ADD A, R7	ADDC A, R7	ORLD P7, A	ANLD P7, A	MOV R7, A	MOV R7, #n	DEC R7	XRL A, R7	DJNZ R7, m	MOV A, R7

2-byte, 2-cycle instruction

1-byte, 2-cycle instruction

7

FUNCTION OF MELPS 8-48 MICROCOMPUTERS

Instruction Code List (M5L8041A-XXXP)

D_3~D_0 Hexa-decimal \ D_7~D_4		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	NOP	INC @R0	XCH A, @R0	XCHD A, @R0	ORL A, @R0	ANL A, @R0	ADD A, @R0	ADDC A, @R0		MOV STS.A	MOV @R0, A	MOV @R0,#n		XRL A, @R0		MOV A, @R0
0001	1		INC @R1	XCH A, @R1	XCHD A, @R1	ORL A, @R1	ANL A, @R1	ADD A, @R1	ADDC A, @R1			MOV @R1, A	MOV @R1,#n		XRL A, @R1		MOV A, @R1
0010	2	OUT DBB,A	JB0 m	IN A,DBB	JB1 m	MOV A,T	JB2 m	MOV T,A	JB3 m		JB4 m		JB5 m		JB6 m		JB7 m
0011	3	ADD A, #n	ADDC A, #n	MOV A, #n		ORL A, #n	ANL A, #n			RET	RETR	MOVP A, @A	JMPP @A		XRL A, #n	MOVP3 A, @A	
0100	4	JMP 0XX	CALL 0XX	JMP 1XX	CALL 1XX	JMP 2XX	CALL 2XX	JMP 3XX	CALL 3XX	JMP 4XX	CALL 4XX	JMP 5XX	CALL 5XX	JMP 6XX	CALL 6XX	JMP 7XX	CALL 7XX
0101	5	EN I	DIS I	EN TCNTI	DIS TCNTI	STRT CNT	STRT T	STOP TCNT		CLR F0	CPL F0	CLR F1	CPL F1	SEL RB0	SEL RB1	EN DMA	EN FLAGS
0110	6		JTF m	JNT0 m	JT0 m	JNT1 m	JT1 m		JF1 m	JOBF m	JNZ m		JF0 m	JZ m	JNIBF	JNC m	JC m
0111	7	DEC A	INC A	CLR A	CPL A	SWAP A	DA A	RRC A	RR A		CLR C	CPL C		MOV A,PSW	MOV PSW,A	RL A	RLC A
1000	8		INC R0	XCH A, R0		ORL A, R0	ANL A, R0	ADD A, R0	ADDC A, R0			MOV R0, A	MOV R0, #n	DEC R0	XRL A, R0	DJNZ R0, m	MOV A, R0
1001	9	IN A, P1	INC R1	XCH A, R1	OUTL P1, A	ORL A, R1	ANL A, R1	ADD A, R1	ADDC A, R1	ORL P1, #n	ANL P1, #n	MOV R1, A	MOV R1, #n	DEC R1	XRL A, R1	DJNZ R1, m	MOV A, R1
1010	A	IN A, P2	INC R2	XCH A, R2	OUTL P2, A	ORL A, R2	ANL A, R2	ADD A, R2	ADDC A, R2	ORL P2, #n	ANL R2, #n	MOV R2, A	MOV R2, #n	DEC R2	XRL A, R2	DJNZ R2, m	MOV A, R2
1011	B		INC R3	XCH A, R3		ORL A, R3	ANL A, R3	ADD A, R3	ADDC A, R3			MOV R3, A	MOV R3, #n	DEC R3	XRL A, R3	DJNZ R3, m	MOV A, R3
1100	C	MOVD A, P4	INC R4	XCH A, R4	MOVD P4, A	ORL A, R4	ANL A, R4	ADD A, R4	ADDC A, R4	ORLD P4, A	ANLD P4, A	MOV R4, A	MOV R4, #n	DEC R4	XRL A, R4	DJNZ R4, m	MOV A, R4
1101	D	MOVD A, P5	INC R5	XCH A, R5	MOVD P5, A	ORL A, R5	ANL A, R5	ADD A, R5	ADDC A, R5	ORLD P5, A	ANLD P5, A	MOV R5, A	MOV R5, #n	DEC R5	XRL A, R5	DJNZ R5, m	MOV A, R5
1110	E	MOVD A, P6	INC R6	XCH A, R6	MOVD P6, A	ORL A, R6	ANL A, R6	ADD A, R6	ADDC A, R6	ORLD P6, A	ANLD P6, A	MOV R6, A	MOV R6, #n	DEC R6	XRL A, R6	DJNZ R6, m	MOV A, R6
1111	F	MOVD A, P7	INC R7	XCH A, R7	MOVD P7, A	ORL A, R7	ANL A, R7	ADD R7, A	ADDC A, R7	ORLD P7, A	ANLD P7, A	MOV R7, A	MOV R7, #n	DEC R7	XRL A, R7	DJNZ R7, m	MOV A, R7

2-byte 2-cycle instruction

1-byte 2-cycle instruction

DEVELOPMENT OF MASK-PROGRAMMABLE ROMs

GENERAL INFORMATION

This information explains how to specify the object program for the automatic design system for mask ROMs. This system for mask ROM production has been developed to accept a customer's object program specifications for the automatic design system for a mask ROM.

The main segments of the automatic design system are:

1. The plotter instructions for mask production.
2. A check list for verifing that the customer's specifications have been met.
3. A test program to assure that the production ROMs meet specifications.

An EPROM in which a program is stored is used for a customer's specifications. A separate (set of) EPROM(s) should be produced for each object program.

Three sets of EPROM(s) should be supplied with the confirmation material.

EPROM SPECIFICATIONS

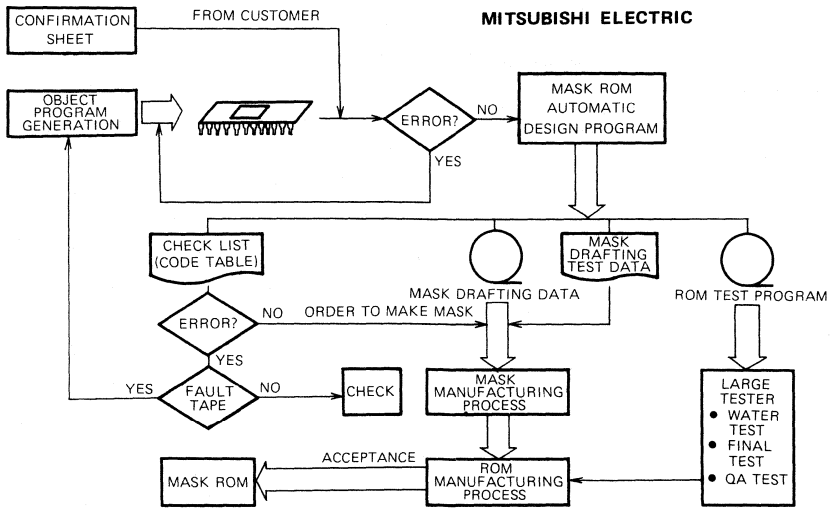
1. The Mitsubishi M5L2708K, M5L2716K, M5L2732K or M5L8748S are standard, but Intel 2708, 2716, 2732, 8748 or equivalent devices may be used.
2. The high-level data of both data outputs and address inputs of the supplied EPROM will be programmed as '1', and low-level as '0'.
3. All the data stored in the EPROM are considered as valid and processed to make masks.
4. The object program specifications should be stored on each of a set of 3 EPROMs. During the mask ROM development, the data from each address of the 3 EPROMs are compared. If 2 or 3 of the 3 values compared are equal, the value will be considered valid and programmed into mask ROM. If the 3 values are all different, it is an error condition and the mask ROM is not produced.

ITEMS TO CONFIRM FOR ORDERING

1. Specify the type number M5L8048-XXXP or M5L8049-XXXP. The 3-digit number XXX will be assigned by Mitsubishi.
2. Cleary indicate the type number of EPROM and address designation letter symbols A and B on the supplied EPROMs. For the M5L8049-XXXP, 2 sets of EPROMs are required when the object program is contained in M5L2708K EPROMs or equivalent devices. To identify these 2 sets of EPROMs, address designation letter symbols A and B are used which are specified in the confirmation material.

7

MASK ROM DEVELOPMENT FLOW CHART



DEVELOPMENT OF MASK-PROGRAMMABLE ROMs

MELPS 8-48 MASK-PROGRAMMABLE ROM CONFIRMATION MATERIAL

SINGLE-CHIP 8-BIT MICROCOMPUTERS M5L8048-XXXP, M5L8049-XXXP

MITSUBISHI ELECTRIC

		Signature
Customer Company name _____ Company address _____	Tel _____	Prepared
Company contact _____	Date _____	Approved

The single-chip microcomputer type number to order and the type of EPROMs to be supplied should be specified by checking √ in the boxes. Three sets of EPROMs should be supplied.

Single-chip microcomputer type number \ EPROM type number	2708	2716	2732	8748
☐ M5L8048-XXXP	☐ A(000_{16}~$3FF_{16}$)	☐ A(000_{16}~$3FF_{16}$)	☐ A(000_{16}~$3FF_{16}$)	☐ A(000_{16}~$3FF_{16}$)
☐ M5L8049-XXXP	☐ A(000_{16}~$3FF_{16}$) B(400_{16}~$7FF_{16}$)	☐ A(000_{16}~$7FF_{16}$)	☐ A(000_{16}~$7FF_{16}$)	–

Note 1: The high-level data of both data outputs and address inputs of the supplied EPROM will be programmed as '1', and low-level as '0'.
2: Cleary indicate the type number of EPROMs and address designation letter symbols A and B on the supplied EPROMs.
3: The data of the addresses in parenthesises on the EPROM are programmed onto the ROM.
4: The data from each PROM in the set is compared and if 2 of the 3 are equal, the equal value will be programmed into the ROM. When the 3 values are different programming is halted and the customer is notified of the error. The error report will show the address and data.

CUSTOMER'S IDENTIFICATION MARK

If you require a special identification mark, please specify in the following format.

Mitsubishi IC type number

Note 5 A mark field should start with the box at the extreme right
6 The identification mark should be no more than 12 characters consisting of alphanumeric characters (except J.I. and O) or dashes

COMMENTS

MITSUBISHI ELECTRIC

M5L8048-XXXP

SINGLE-CHIP 8-BIT MICROCOMPUTER

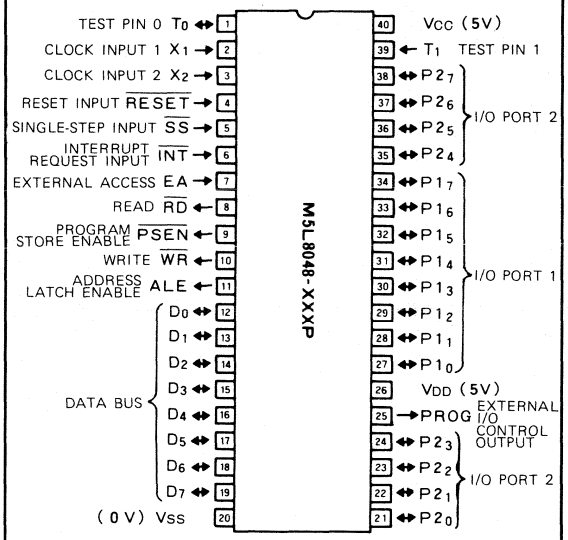
DESCRIPTION

The M5L8048-XXXP is an 8-bit parallel microcomputer frabricated on a single chip using high-speed N-channel silicon-gate ED-MOS technology.

FEATURES

- Single 5V power supply
- Instruction cycle ..... 2.5μs (min)
- Basic machine instructions: 96
  - 1-byte instructions: 68
  - 2-byte instructions: 28
- Direct addressing ..... up to 4096 bytes
- Internal ROM ..... 1024 bytes
- Internal RAM ..... 64 bytes
- Built-in timer/event counter ..... 8 bits
- I/O Ports ..... 27 lines
- Easily expandable Memory and I/O:
- Subroutine nesting ..... 8 levels
- External and timer/event counter interrupt . 1 level each
- Low power standby mode:
- External RAM ..... 256 bytes
- Interchangeable with Intel's P8048 in pin configuration and electrical characteristics

PIN CONFIGURATION (TOP VIEW)



Outline 40P1

7

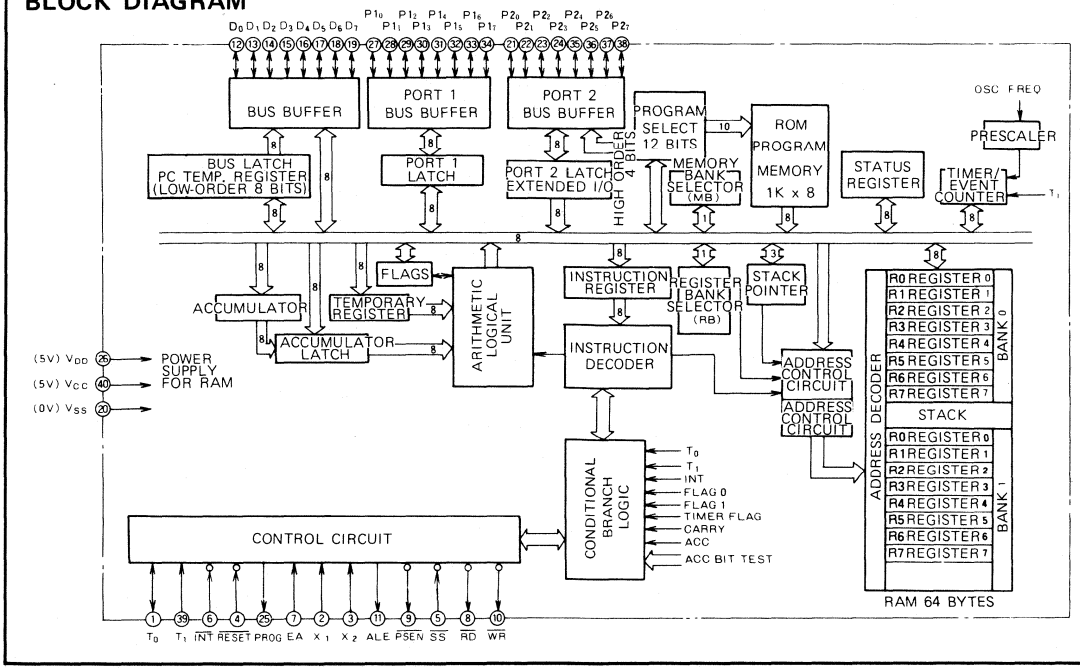
APPLICATION

Control processor or CPU for a wide variety of applications

FUNCTION

The M5L8048-XXXP LSI is an integrated 8-bit CPU, with memory (ROM, RAM) and timer/event counter interrupt all contained on a single chip.

BLOCK DIAGRAM



SINGLE-CHIP 8-BIT MICROCOMPUTER

PIN DESCRIPTION

Pin	Name	Input or Output	Function
V_{SS}	Ground		Normally connected to ground (0V).
V_{CC}	Main power supply		Connected to 5V power supply.
V_{DD}	Power supply		① Connected to 5V power supply. ② Used for memory hold when V_{CC} is cut.
PROG	Program	Output	Strobe signal for M5L8243P I/O Expander.
$P1_0 \sim P1_7$	Port 1	Input/output	Quasi-bidirectional port. When used as an input port, FF_{16} must first be output to this port. After reset, when not used as an output port nothing can be output.
$P2_0 \sim P2_7$	Port 2	Input/output	① The same as port 1.
		Output	② $P2_0 \sim P2_3$ output the high-order 4 bits of the program counter when using external program memory.
		Input/output	③ $P2_0 \sim P2_3$ serve as a 4-bit I/O expander bus for the M5L8243P.
$D_0 \sim D_7$	Data bus	Input/output	① Provides true bidirectional bus transfer of instructions and data between the CPU and external memory. Synchronizing is done with signals $\overline{RD}/\overline{WR}$. The output data is latched.
			② When using external program memory the output of the low-order 8 bits of the program counter are synchronized with ALE. After that the transfer of the instruction code or data from external program memory is synchronized with $\overline{PSEN}$.
			③ The output of addresses for data using external data memory is synchronized with ALE. After that the transfer of data with the external data memory is synchronized with $\overline{RD}/\overline{WR}$. (MOVX A, @Rr and MOVX @Rr, A)
T_0	Test pin 0	Input	① Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions. (JT0 m and JNT0 m)
		Output	② Used for outputting the internal clock signal. (ENT0 CLK)
T_1	Test pin 1	Input	① Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions. (JT1 m and JNT1 m) ② When enabled event signals are transferred to the timer/event counter. (STRT CNT)
$\overline{INT}$	Interrupt	Input	① Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions. (JN1 m) ② Used for external interrupt to CPU.
$\overline{RD}$	Read control	Output	Read control signal used when the CPU requests data from external data memory or external devices to be transferred to the data bus. (MOVX A, @Rr and INS A, BUS)
$\overline{WR}$	Write control	Output	Write control signal used when the CPU sends data through the data bus to external data memory or external device. (MOVX @R, A and OUTL BUS, A)
$\overline{RESET}$	Reset	Input	Control used to initialize the CPU.
ALE	Address latch enable	Output	A signal used for latching the address on the data bus. An ALE signal occurs once during each cycle.
$\overline{PSEN}$	Program store enable	Output	Strobe signal to fetch external program memory.
$\overline{SS}$	Single step	Input	Control signal used in conjunction with ALE to stop the CPU through each instruction, in the single step mode.
EA	External access	Input	① Normally maintained at 0V. ② When the level is raised to 5V, external memory will be accessed even when the address is less than 400_{16} (1024).
X_1, X_2	Crystal inputs	Input	External crystal oscillator or RC circuit input for generating internal clock signals. An external clock signal can be input through X_1 or X_2.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{CC}	Supply voltage	With respect to V_{SS}	−0.5～7	V
V_{DD}	Supply voltage		−0.5～7	V
V_I	Input voltage		−0.5～7	V
V_O	Output voltage		−0.5～7	V
Pd	Power dissipation	Ta＝25℃	1.5	W
Topr	Operating free-air temperature range		0～70	℃
Tstg	Storage temperature range		−65～150	℃

RECOMMENDED OPERATING CONDITIONS (Ta＝0～70℃, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{CC}	Supply voltage	4.5	5	5.5	V
V_{DD}	Supply voltage	4.5	5	5.5	V
V_{SS}	Supply voltage		0		V
V_{IH1}	High-level input voltage, except X1, X2 and $\overline{RESET}$	2		V_{CC}	V
V_{IH2}	High-level input voltage, except X1, X2 and $\overline{RESET}$	3.8		V_{CC}	V
V_{IL}	Low-level input voltage	−0.5		0.8	V

ELECTRICAL CHARACTERISTICS (Ta＝0～70℃, $V_{CC}=V_{DD}=5V\pm10\%$, $V_{SS}=0V$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{OL}	Low-level output voltage, BUS, $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE	I_{OL}＝2mA			0.45	V
V_{OL1}	Low-level output voltage, except the above and PROG	I_{OL}＝1.6mA			0.45	V
V_{OL2}	Low-level output voltage, PROG	I_{OL}＝1mA			0.45	V
V_{OH}	High-level output voltage, BUS, $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE	$I_{OH}=-100\mu A$	2.4			V
V_{OH1}	High-level output voltage, except the above	$I_{OH}=-50\mu A$	2.4			V
I_{IL}	Input leak current, T1, $\overline{INT}$	$V_{SS}\leqq V_{IN}\leqq V_{CC}$	−10		10	μA
I_{OL}	Output leak current, BUS, T0 high-impedance state	$V_{SS}+0.45\leqq V_{IN}\leqq V_{CC}$	−10		10	μA
I_{LI1}	Input current during low-level input, port	V_{IL}＝0.8V		−0.2		mA
I_{LI2}	Input current during low-level input, $\overline{RESET}$, $\overline{SS}$	V_{IL}＝0.8V		−0.05		mA
I_{DD}	Supply current from V_{DD}			10	20	mA
$I_{DD}+I_{CC}$	Supply current from V_{DD} and V_{CC}			65	135	mA

TIMING REQUIREMENTS (Ta＝0～70℃, $V_{CC}=V_{DD}=5V\pm10\%$, $V_{SS}=0V$, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Limits			Unit
			Min	Typ	Max	
t_C	Cycle time	t_{CY}	2.5		15.0	μs
$t_{h\ (PSEN-D)}$	Data hold time after $\overline{PSEN}$	t_{DR}	0		200	ns
$t_{h\ (R-D)}$	Data hold time after $\overline{RD}$	t_{DR}	0		200	ns
$t_{su\ (PSEN-D)}$	Data setup time after $\overline{PSEN}$	t_{RD}			500	ns
$t_{su\ (R-D)}$	Data setup time after $\overline{RD}$	t_{RD}			500	ns
$t_{su\ (A-D)}$	Data setup time after address	t_{AD}			950	ns
$t_{su\ (PROG-D)}$	Data setup time after PROG	t_{PR}			810	ns
$t_{h\ (PROG-D)}$	Data hold time before PROG	t_{PF}	0		150	ns

Note 1: The input voltage level of the input voltage is V_{IL}=0.45V and V_{IH}=2.4V.

SINGLE-CHIP 8-BIT MICROCOMPUTER

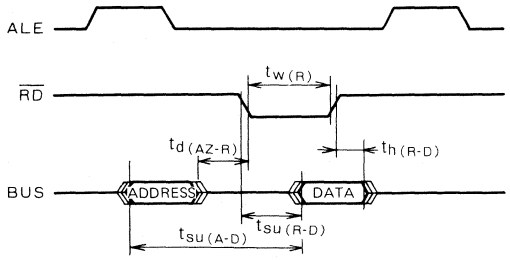
SWITCHING CHARACTERISTICS (T_a=0~70℃, V_{CC}=V_{DD}=5V±10%, V_{SS}=0V, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Limits			Unit
			Min	Typ	Max	
$t_{W(ALE)}$	ALE pulse width	t_{LL}	400			ns
$t_{d(A-ALE)}$	Delay time, address to ALE signal	t_{AL}	120			ns
$t_{v(ALE-A)}$	Address valid time after ALE	t_{LA}	80			ns
$t_{W(PSEN)}$	$\overline{PSEN}$ pulse width	t_{CC}	700			ns
$t_{W(R)}$	$\overline{RD}$ pulse width	t_{CC}	700			ns
$t_{W(W)}$	$\overline{WR}$ pulse width	t_{CC}	700			ns
$t_{d(Q-W)}$	Delay time, data to $\overline{WR}$ signal	t_{DW}	500			ns
$t_{v(W-Q)}$	Data valid time after $\overline{WR}$	t_{WD}	120			ns
$t_{d(A-W)}$	Delay time, address to $\overline{WR}$ signal	t_{AW}	230			ns
$t_{d(AZ-R)}$	Delay time, address disable to $\overline{RD}$ signal	t_{AFC}	0			ns
$t_{d(AZ-PSEN)}$	Delay time, address disable to $\overline{PSEN}$ signal	t_{AFC}	0			ns
$t_{d(PC-PROG)}$	Delay time, port control to PROG signal	t_{CP}	110			ns
$t_{v(PROG-PC)}$	Port control valid time after PROG	t_{PC}	100			ns
$t_{p(Q-PROG)}$	Delay time, data to PROG signal	t_{DP}	250			ns
$t_{v(PROG-Q)}$	Data valid time after PROG	t_{PD}	65			ns
$t_{W(PROGL)}$	PROG low pulse width	t_{PP}	1200			ns
$t_{d(Q-ALE)}$	Delay time, data to ALE signal	t_{PL}	350			ns
$t_{v(ALE-Q)}$	Data valid time after ALE	t_{LP}	150			ns

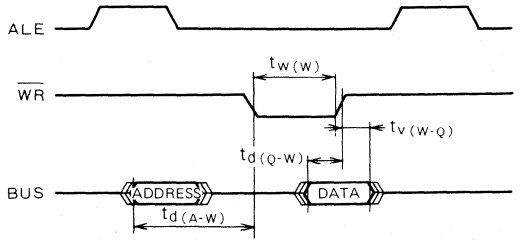
Note 2: Conditions of measurement: control output C_L=80pF
data bus output, port output C_L=150pF, t_C=2.5μS
3: Reference levels for the input/output voltages are low level=0.8V and high level=2V

TIMING DIAGRAM

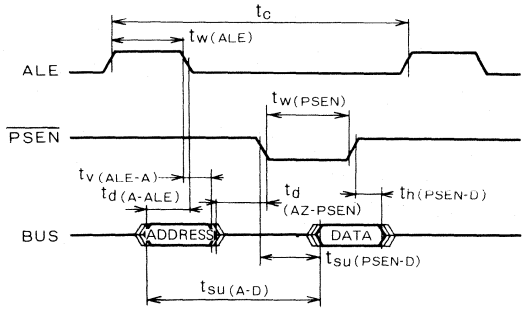
Read from External Data Memory



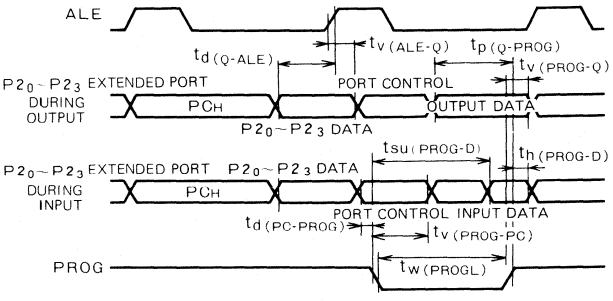
Write to External Data Memory



Instruction Fetch from External Program Memory



Port 2



MITSUBISHI ELECTRIC

SINGLE-CHIP 8-BIT MICROCOMPUTER WITH EPROM

DESCRIPTION

The M5L8748S is an 8-bit parallel microcomputer frabricated on a single-chip using high-speed N-channel silicon-gate ED-MOS technology. This contains ultraviolet-light erasable and electrically reprogrammable ROM (EPROM) on a chip, so it is easy to change the program stored in the EPROM.

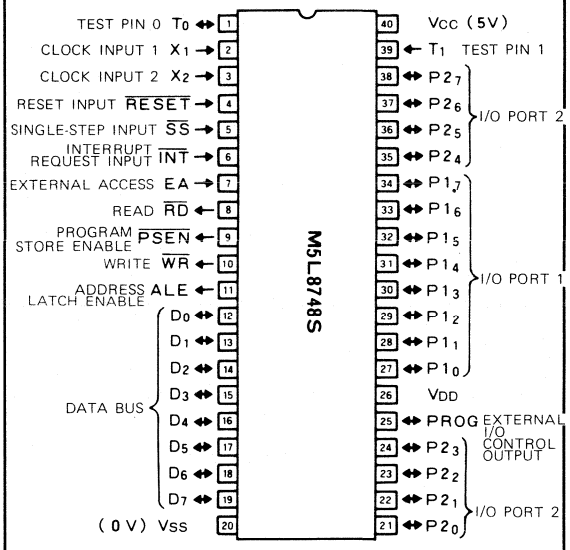
FEATURES

- Single 5V power supply
- Instruction cycle . . . . . 2.5μs (min)
- Basic machine instructions 96
  - 1-byte instructions: 68
  - 2-byte instructions: 28
- Direct addressing . . . . . up to 4096 bytes
- Internal EPROM . . . . . 1024 bytes
- Internal RAM . . . . . 64 bytes
- Built-in timer/event counter . . . . . 8 bits
- I/O Ports . . . . . 27 lines
- Easily expandable memory and I/O:
- Subroutine nesting . . . . . 8 levels
- External and timer/event counter interrupt . 1 level each
- External RAM . . . . . 256 bytes
- Interchangeable with the Intel's D8748 in pin configuration and electrical characteristics

APPLICATIONS

- A CPU for special repetitive processing or control for which a small number of units are to be produced.
- A debugging CPU for program, application and system design development
- A CPU for prototype and preproduction systems prior to factory-programmed mask ROM production

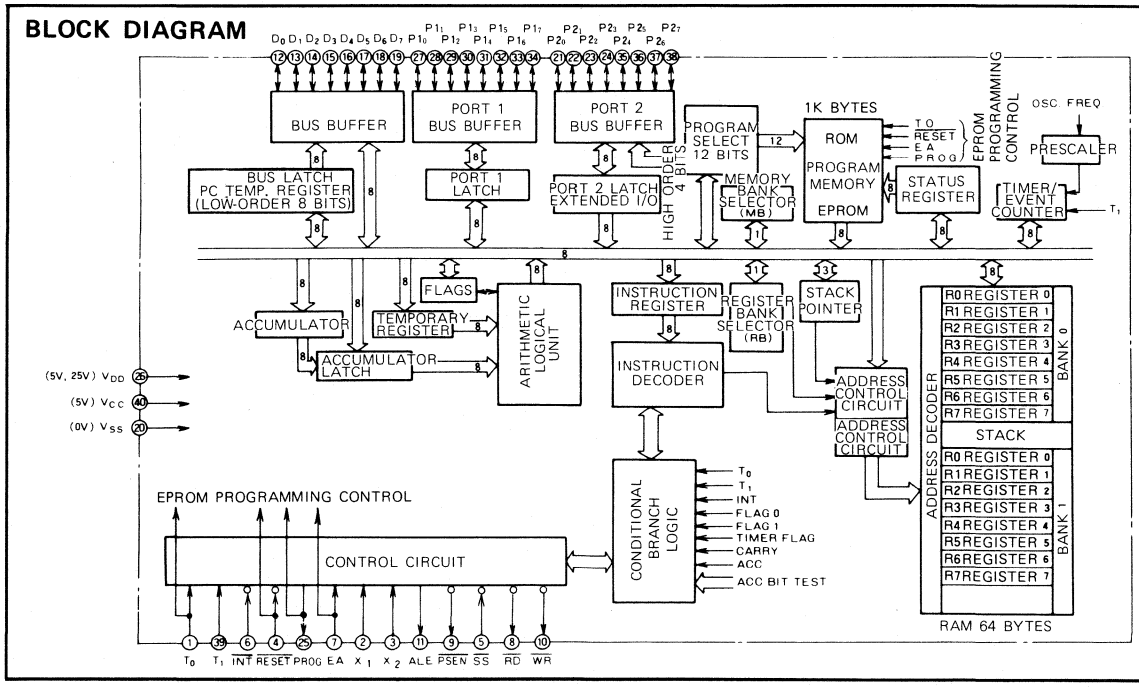
PIN CONFIGURATION (TOP VIEW)



Outline 40S10

7

BLOCK DIAGRAM



SINGLE-CHIP 8-BIT MICROCOMPUTER WITH EPROM

PIN DESCRIPTION

Pin	Name	Input or Output	Function
V_{SS}	Ground		Normally connected to ground (0V).
V_{CC}	Main power supply		Connected to 5V power supply.
V_{DD}	Program power supply		① Normally connected to 5V power supply. ② When programming to EPROM, 25V is required.
PROG	Program	Input	① Used to supply 25V program pulses (50 ms width) from an outside source when programming to EPROM.
		Output	② Strobe signal for M5L8243P I/O Expander.
$P1_0 \sim P1_7$	Port 1	Input/output	Quasi-bidirectional port. When used as an input port, FF_{16} must first be output to this port. After reset, when not used as an output port nothing can be output.
$P2_0 \sim P2_7$	Port 2	Input/output	① The same as port 1.
		Output	② $P2_0 \sim P2_3$ output the high-order 4 bits of the program counter when using external program memory.
		Input/output	③ $P2_0 \sim P2_3$ serve as a 4-bit I/O expander bus for the M5L8243P.
$D_0 \sim D_7$	Data bus	Input/output	① Provides true bidirectional bus transfer of instructions and data between the CPU and external memory. Synchronizing is done with signals $\overline{RD}/\overline{WR}$. The output data is latched.
			② When using external program memory the output of the low-order 8 bits of the program counter are synchronized with ALE. After that the transfer of the instruction code or data from external program memory is synchronized with PSEN.
			③ The output of addresses for data using external data memory is synchronized with ALE. After that the transfer of data with the external data memory is synchronized with $\overline{RD}/\overline{WR}$. (MOVX A, @Rr and MOVX @Rr, A)
T_0	Test pin 0	Input	① Control signal from an external source for conditonal jumping in a program. Jumping is dependent on external conditions. (JT0 m and JNT0 m)
		Output	② Used for outputting the internal clock signal. (ENT0 CLK)
T_1	Test pin 1	Input	① Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions. (JT1 m and JNT1 m)
			② When enabled event signals are transferred to the timer/event counter. (STRT CNT)
$\overline{INT}$	Interrupt	Input	① Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions. (JN1 m)
			② Used for external interrupt to CPU.
$\overline{RD}$	Read control	Output	Read control signal used when the CPU requests data from external data memory or external devices to be transferred to the data bus. (MOVX A, @Rr and INS A, BUS)
$\overline{WR}$	Write control	Output	Write control signal used when the CPU sends data through the data bus to external data memory or external devices. (MOVX @R, A and OUTL BUS, A)
$\overline{RESET}$	Reset	Input/output	① Control used to initialize the CPU.
			② Latch signal for the EPROM address when programming to EPROM and for reading from EPROM (verify mode).
ALE	Address latch enable	Output	A signal used for latching the address on the data bus. An ALE signal occurs once during each cycle.
$\overline{PSEN}$	Program store enable	Output	Strobe signal used to fetch from external program memory.
$\overline{SS}$	Single step	Input	Control signal used in conjunction with ALE to stop program execution at the finish of each instruction, in the single step mode.
EA	External access	Input	① Normally maintained at 0V. ② When the level is raised to 5V, external memory will be accessed even when the address is less than 400_{16} (1024). ③ When in the programming mode for the EPROM a 25V power supply must be available at this terminal.
X_1, X_2	Crystal inputs	Input	External crystal oscillator or RC circuit input for generating internal clock signals. An external clock signal can be input through X_1 or X_2.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{CC}	Supply voltage	With respect to V_{SS}	−0.5～7	V
V_{DD}	Supply voltage		−0.5～26.5	V
V_I	Input voltage		−0.5～7	V
V_O	Output voltage, all outputs except ϕ_1 and ϕ_2		−0.5～7	V
Pd	Power dissipation	Ta=25℃	1.5	W
Topr	Operating free-air temperature range		0～70	℃
Tstg	Storage temperature range		−65～150	℃

RECOMMENDED OPERATING CONDITIONS

CPU Operation (Ta=0～70℃, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{CC}	Supply voltage	4.75	5	5.25	V
V_{DD}	Supply voltage, except programming EPROM	4.75	5	5.25	V
V_{DD}	Supply voltage, programming EPROM	24	25	26	V
V_{SS}	Supply voltage		0		V
V_{IH1}	High-level input voltage, except X_1, X_2, $\overline{RESET}$	2		Vcc	V
V_{IH2}	High-level input voltage, X_1, X_2, $\overline{RESET}$	3.8		Vcc	V
V_{IL}	Low-level input voltage	−0.5		0.8	V

EPROM PROGRAMMING (Ta=25±5℃, Vcc=5V±5%, VDD=25±1V, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
$V_{DD(H)}$	High-level program supply voltage	24.		26	V
$V_{DD(L)}$	Low-level program supply voltage	4.75		5.25	V
$V_{IH(PROG)}$	High-level program pulse input voltage	21.5		24.5	V
$V_{IL(PROG)}$	Low-level program pulse input voltage			0.2	V
$V_{EA(H)}$	High-level EA input voltage	21.5		24.5	V
$V_{EA(L)}$	Low-level EA input voltage			5.25	V

ELECTRICAL CHARACTERISTICS

CPU Operation (Ta=0～70℃, $V_{CC}=V_{DD}=5V\pm5\%$, $V_{SS}=0V$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{OL}	Low-level output voltage, BUS, $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE	$I_{OL}=2mA$			0.45	V
V_{OL1}	Low-level output voltage, except the above and PROG	$I_{OL}=1.6mA$			0.45	V
V_{OL2}	Low-level output voltage, PROG	$I_{OL}=1mA$			0.45	V
V_{OH}	High-level output voltage, BUS, $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE	$I_{OH}=-100\mu A$	2.4			V
V_{OH1}	High-level output voltage, except the above	$I_{OH}=-50\mu A$	2.4			V
I_{IL}	Input leak current, T1, $\overline{INT}$	$V_{SS}\leqq V_{IN}\leqq V_{CC}$	−10		10	μA
I_{LI1}	Low-level input current, ports	$V_{IL}=0.8V$		−0.2		mA
I_{LI2}	Low-level input current, $\overline{RESET}$, $\overline{SS}$	$V_{IL}=0.8V$		−0.05		mA
I_{DD}	Supply current from V_{DD}	Ta=25°C		10	20	mA
$I_{DD}+I_{CC}$	Supply current from V_{DD} and V_{CC}	Ta=25°C		65	135	mA

EPROM PROGRAMMING (Ta=25±5℃, Vcc=5V±5%, VDD=25±1V, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
I_{DD}	Supply current from V_{DD}				30	mA
$I_{IH(PROG)}$	High-level input current, PROG				16	mA
$I_{IH(EA)}$	High-level input current, EA				1	mA

7

TIMING REQUIREMENTS

Read/Write of External Memory (Ta = 0 ~ 70°C, $V_{CC} = V_{DD} = 5V \pm 5\%$, $V_{SS} = 0V$, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Limits			Unit
			Min	Typ	Max	
t_C	Cycle time	t_{CY}	2.5		15.0	μs
$t_{h(PSEN-D)}$	Data hold time after $\overline{PSEN}$	t_{DR}	0		200	ns
$t_{h(R-D)}$	Data hold time after $\overline{RD}$	t_{DR}	0		200	ns
$t_{su(PSEN-D)}$	Data setup time after $\overline{PSEN}$	t_{RD}			500	ns
$t_{su(R-D)}$	Data setup time after $\overline{RD}$	t_{RD}			500	ns
$t_{su(A-D)}$	Data setup time after address	t_{AD}			950	ns

Note 1: The input voltage level is V_{IL}=0.45 and V_{IH}=2.4V.

Port 2 (Ta = 0 ~ 70°C, $V_{CC} = V_{DD} = 5V \pm 5\%$, $V_{SS} = 0V$, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Limits			Units
			Min	Typ	Max	
$t_{su(PROG-D)}$	Data setup time after PROG	t_{PR}			810	ns
$t_{h(PROG-D)}$	Data hold time after PROG	t_{PF}	0		150	ns

Note 2: The input voltage level of the input voltage is V_{IL} = 0.45V and V_{IH}=2.4V.

EPROM PROGRAMMING (Ta = 25 ± 5°C, V_{CC} = 5V ± 5%, V_{DD} = 25V ± 1V, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Limits			Unit
			Min	Typ	Max	
$t_{su(A-RES)}$	Address setup time before $\overline{RESET}$	t_{AW}	$4t_C$			
$t_{h(RES-A)}$	Address hold time after $\overline{RESET}$	t_{WA}	$4t_C$			
$t_{su(D-PROG)}$	Data setup time before PROG	t_{DW}	$4t_C$			
$t_{h(PROG-D)}$	Data hold time after PROG	t_{WD}	$4t_C$			
$t_{h(T_0-RESH)}$	$\overline{RESET}$ high hold time after T_0 (verify mode)	t_{PH}	$4t_C$			
$t_{su(V_{DD}-PROG)}$	V_{DD} setup time before PROG	t_{VDDW}	$4t_C$			
$t_{h(PROG-V_{DD})}$	V_{DD} hold time after PROG	t_{VDDH}	0			ns
$t_{W(PROG)}$	PROG pulse width	t_{PW}	50		60	ms
$t_{su(T_0-RES)}$	Setup time before $\overline{RES}$	t_{TW}	$4t_C$			
$t_{h(VDD-T_0)}$	Hold time after V_{DD}	t_{WT}	$4t_C$			
$t_{W(RES)}$	$\overline{RESET}$ pulse width	t_{WW}	$4t_C$			

Note 3: CPU cycle time t_c requires 5μs min.
4: Rise time (t_r) and fall time (t_f) of V_{DD} and PROG should be within the range of 0.5~2μs.
5: RESET setup time for the positive-going EA requires 4 t_c min.

SWITCHING CHARACTERISTICS

Read/Write of External Memory (T_a = 0 ~ 70℃, V_{CC} = V_{DD} = 5V ± 5%, V_{SS} = 0V, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Limits			Unit
			Min	Typ	Max	
$t_{W(ALE)}$	ALE pulse width	t_{LL}	400			ns
$t_{d(A-ALE)}$	Delay time, address to ALE signal	t_{AL}	120			ns
$t_{v(ALE-A)}$	Address valid time after ALE	t_{LA}	80			ns
$t_{W(\overline{PSEN})}$	$\overline{PSEN}$ pulse width	t_{CC}	700			ns
$t_{W(R)}$	$\overline{RD}$ pulse width	t_{CC}	700			ns
$t_{W(W)}$	$\overline{WR}$ pulse width	t_{CC}	700			ns
$t_{d(Q-W)}$	Delay time, data to $\overline{WR}$ signal	t_{DW}	500			ns
$t_{v(W-Q)}$	Data valid time after $\overline{WR}$	t_{WD}	120			ns
$t_{d(A-W)}$	Delay time, address to $\overline{WR}$ signal	t_{AW}	230			ns
$t_{d(AZ-R)}$	Delay time, address floating to $\overline{RD}$ signal	t_{AFC}	0			ns
$t_{d(AZ-\overline{PSEN})}$	Delay time, address floating to $\overline{PSEN}$ signal	t_{AFC}	0			ns

Note 6: Conditions of measurement: control output C_L=80pF
data bus output C_L=150pF, t_c=2.5μs
7: Reference level for the input/output voltage is low level=0.8V and high level=2V.

Port 2 (T_a = 0 ~ 70℃, V_{CC} = V_{DD} = 5V ± 5%, V_{SS} = 0V, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Limits			Unit
			Min	Typ	Max	
$t_{d(PC-PROG)}$	Delay time, port control to $\overline{PROG}$ signal	t_{CP}	110			ns
$t_{v(PROG-PC)}$	Port control valid time after $\overline{PROG}$	t_{PC}	100			ns
$t_{p(Q-PROG)}$	Delay time, data to $\overline{PROG}$ signal	t_{DP}	250			ns
$t_{v(PROG-Q)}$	Data valid time after $\overline{PROG}$	t_{PD}	65			ns
$t_{W(PROGL)}$	$\overline{PROG}$ low-level pulse width	t_{PP}	1200			ns
$t_{d(Q-ALE)}$	Delay time, data to ALE signal	t_{PL}	350			ns
$t_{v(ALE-Q)}$	Data valid time after ALE	t_{LP}	150			ns

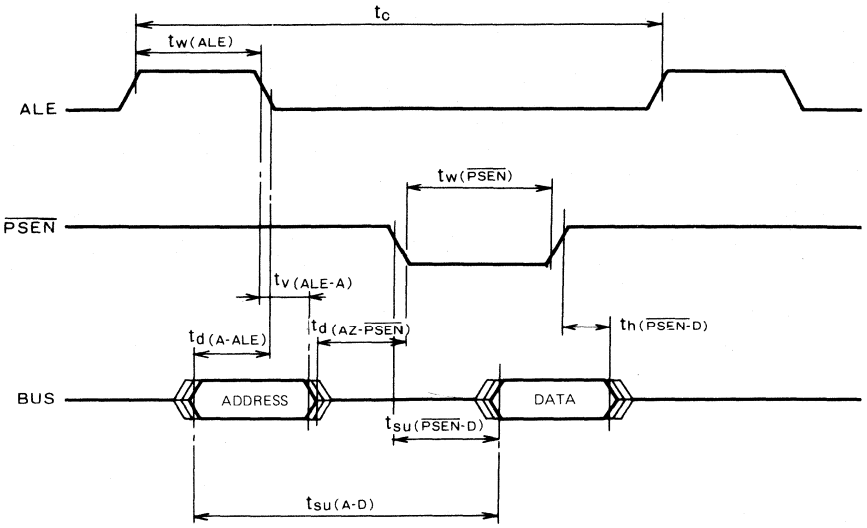
Note 8: Condition of measurement is C_L=150pF, t_c=2.5μs
9: Reference level for the input/output voltage is low level=0.8V and high level=2V.

EPROM PROGRAMMING (T_a = 25 ± 5℃, V_{CC} = 5V ± 5%, V_{DD} = 25V ± 1V, unless otherwise noted)

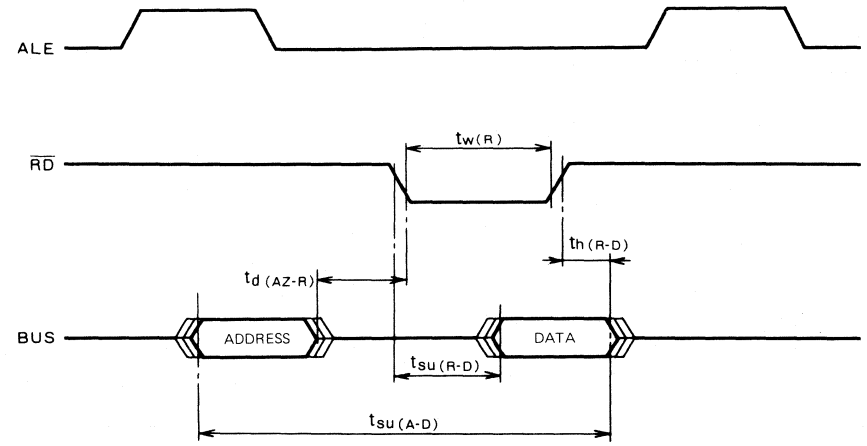
Symbol	Parameter	Alternative symbol	Limits			Unit
			Min	Typ	Max	
$t_{p(T_0-Q)}$	Propagation time between T_0 and data.	t_{DO}			4tc	

TIMING DIAGRAM

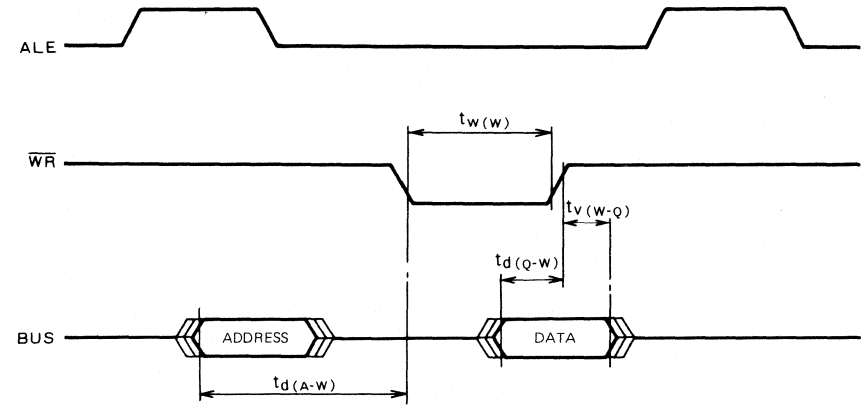
Instruction fetch from external program memory



Reading from external data memory

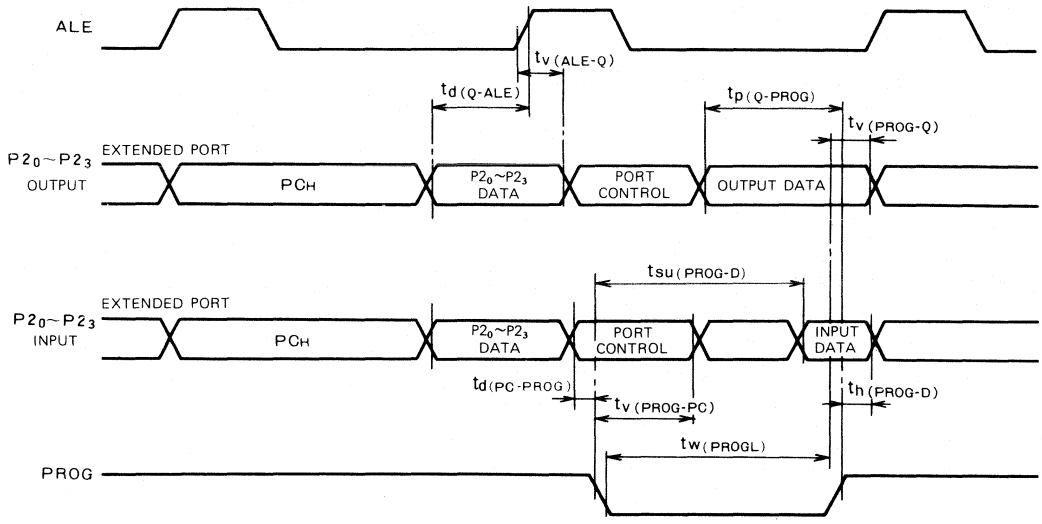


Writing to external data memory

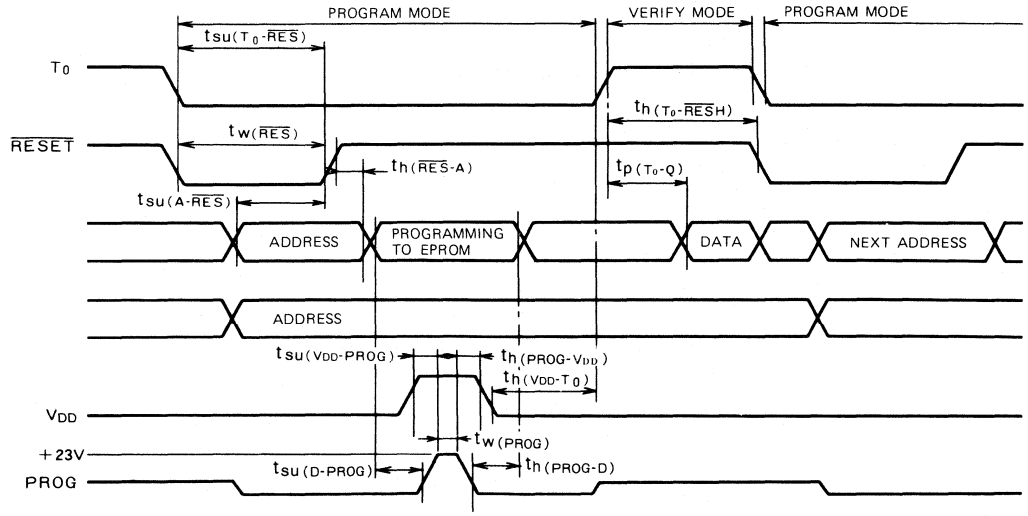


MITSUBISHI ELECTRIC

Port 2



PROGRAMMING to EPROM



7

PROGRAMMING TO EPROM AND ERASING

Details of how to program data onto the EPROM for the M5L8748S is shown in Fig. 1. The EPROM can be erased by approximately 15 Ws/cm^2 of exposure to high-intensity 2537Å short-wave ultraviolet rays.

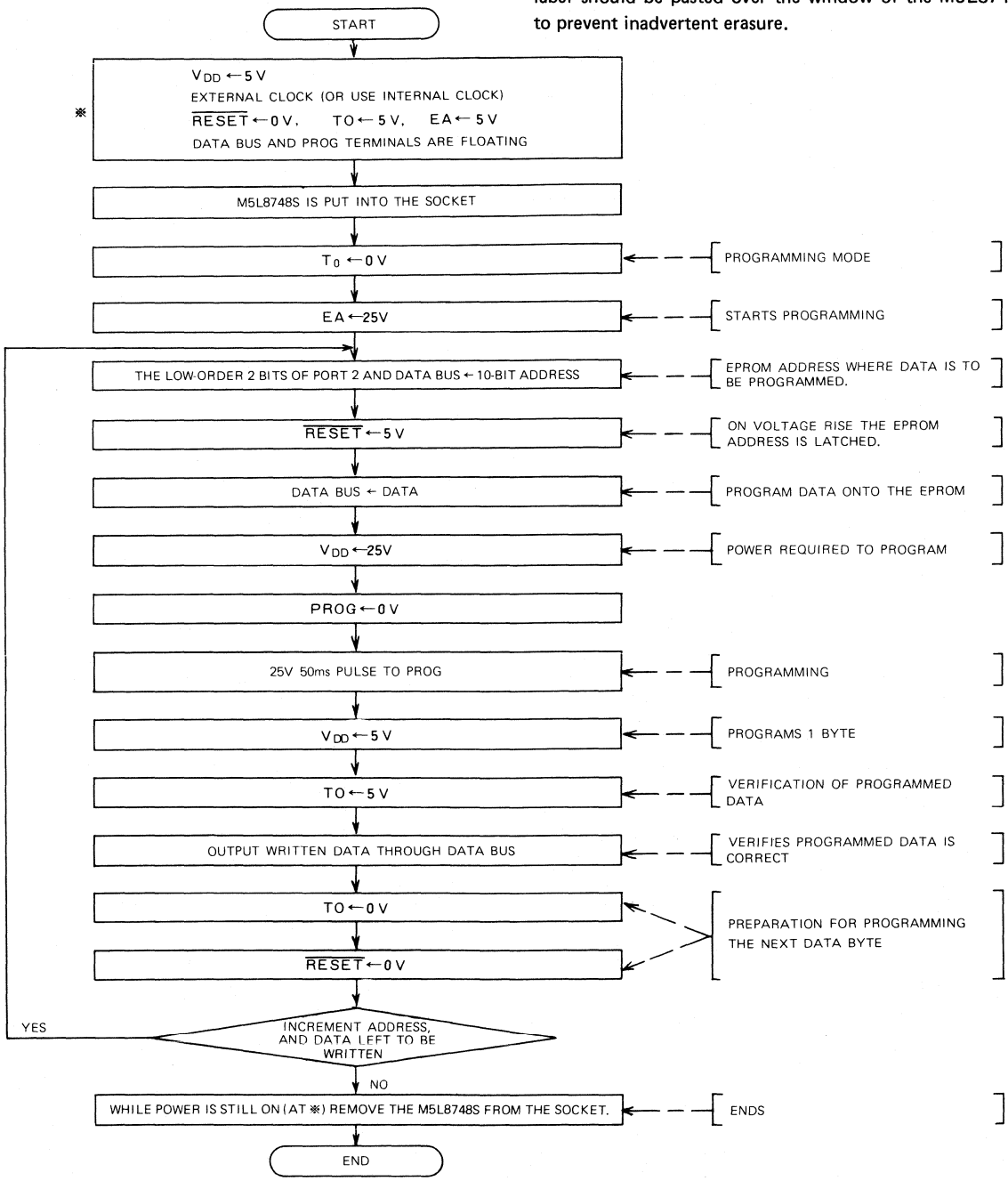
For example: the S-52 ultraviolet lamp has an intensity of 17,000μW/cm^2 at a distance of 2.4 cm from the lamp.

The necessary exposure would be:

$$\frac{15\text{Ws/cm}^2}{17{,}000\mu\text{W/cm}^2} = 900 \text{ seconds} = 15 \text{ minutes}$$

Once data has been entered on the EPROM an opaque label should be pasted over the window of the M5L8748S to prevent inadvertent erasure.

Fig. 1 Flowchart of programming onto the EPROM



M5L8049-XXXP, M5L8039P-6

SINGLE-CHIP 8-BIT MICROCOMPUTER

DESCRIPTION

The M5L8049-XXXP and M5L8039P-6 are 8-bit parallel microcomputers frabicated on a single chip using high-speed N-channel silicon gate ED-MOS technology.

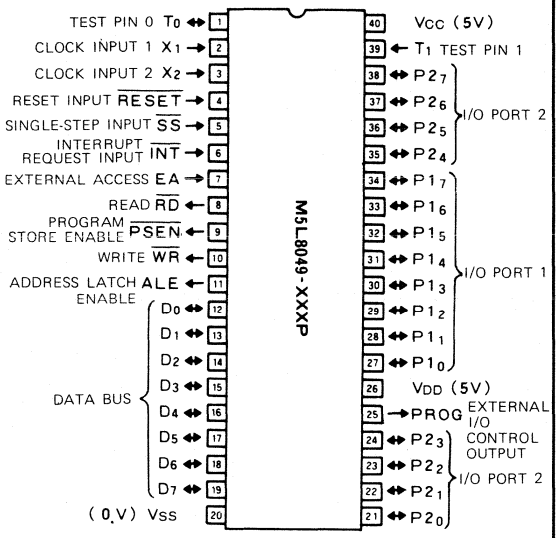
FEATURES

- Single 5V power supply
- Instruction cycle ..... 2.5μs (min.)
- Basic machine instructions 96
  - 1-byte instructions: 68
  - 2-byte instructions: 28
- Direct addressing ..... up to 4096 bytes
- Internal ROM (for M5L8049-XXXP) ..... 2048 bytes
- Internal RAM ..... 128 bytes
- Built-in timer/event counter ..... 8 bits
- I/O Ports ..... 27 lines
- Easily expandable Memory and I/O:
- Subroutine nesting ..... 8 levels
- External and timer/event counter interrupt . 1 level each
- Low power standby mode
- External RAM ..... 256 bytes
- M5L8049-XXXP/M5L8039P-6 are interchangeable with Intel's P8049 (6MHz operation)/P8039-6 in pin configuration and electrical characteristics.

APPLICATION

- Control processor or CPU for a wide variety of applications

PIN CONFIGURATION (TOP VIEW)

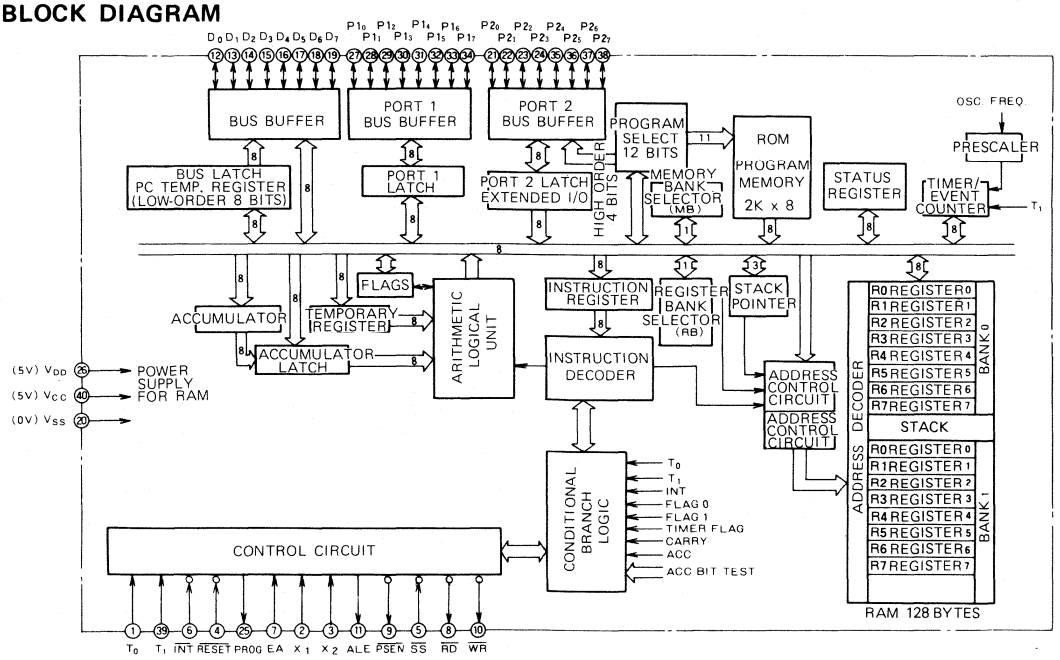


Outline 40P1

FUNCTION

The M5L8049-XXXP and M5L8039P-6 are integrated 8-bit CPUs, with memory (ROM, RAM) and timer/event counter interrupt all contained on a single chip.

BLOCK DIAGRAM



SINGLE-CHIP 8-BIT MICROCOMPUTER

PIN DESCRIPTION

Pin	Name	Input or Output	Function
V_{SS}	Ground		Normally connected to ground (0V)
V_{CC}	Main power supply		Connected to 5V power supply
V_{DD}	Power supply		① Connected to 5V power supply ② Used for memory hold when V_{CC} is cut
PROG	Program	Output	Strobe signal for M5L8243P I/O Expander
$P1_0 \sim P1_7$	Port 1	Input/output	Quasi-bidirectional port. When used as an input port, FF_{16} must first be output to this port. After reset, when not used as an output port nothing can be output.
$P2_0 \sim P2_7$	Port 2	Input/output	① The same as port 1
		Output	② $P2_0 \sim P2_3$ output the high-order 4 bits of the program counter when using external program memory
		Input/output	③ $P2_0 \sim P2_3$ serve as a 4-bit I/O expander bus for the M5L8243P
$D_0 \sim D_7$	Data bus	Input/output	① Provides true bidirectional bus transfer of instructions and data between the CPU and external memory. Synchronizing is done with signals $\overline{RD}/\overline{WR}$. The output data is latched.
			② When using external program memory the output of the low-order 8 bits of the program counter are synchronized with ALE. After that the transfer of the instruction code or data from external program memory is synchronized with $\overline{PSEN}$.
			③ The output of addresses for data using external data memory is synchronized with ALE. After that the transfer of data with the external data memory is synchronized with $\overline{RD}/\overline{WR}$. (MOVX A, @Rr and MOVX @Rr, A)
T_0	Test pin 0	Input	① Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions. (JT0 m and JNT0 m)
		Output	② Used for outputting the internal clock signal. (ENT0 CLK)
T_1	Test pin 1	Input	① Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions. (JT1 m and JNT1 m) ② When enabled event signals are transferred to the timer/event counter. (STRT CNT)
$\overline{INT}$	Interrupt	Input	① Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions. (JN1 m) ② Used for external interrupt to CPU
$\overline{RD}$	Read control	Output	Read control signal used when the CPU requests data from external data memory or external devices to be transferred to the data bus. (MOVX A, @Rr and INS A, BUS)
$\overline{WR}$	Write control	Output	Write control signal used when the CPU sends data through the data bus to external data memory or external device. (MOVX @R, A and OUTL BUS, A)
$\overline{RESET}$	Reset	Input	Control used to initialize the CPU.
ALE	Address latch enable	Output	A signal used for latching the address on the data bus. An ALE signal occurs once during each cycle.
$\overline{PSEN}$	Program store enable	Output	Strobe signal to fetch external program memory.
$\overline{SS}$	Single step	Input	Control signal used in conjunction with ALE to stop the CPU through each instruction, in the single step mode.
EA	External access	Input	① Normally maintained at 0V ② When the level is raised to 5V, external memory will be accessed even when the address is less than 400_{16} (2048).
X_1, X_2	Crystal inputs	Input	External crystal oscillator or RC circuit input for generating internal clock signals. An external clock signal can be input through X_1 or X_2.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{CC}	Supply voltage	With respect to V_{SS}	−0.5～7	V
V_{DD}	Supply voltage		−0.5～7	V
V_I	Input voltage		−0.5～7	V
V_O	Output voltage		−0.5～7	V
Pd	Power dissipation	Ta＝25℃	1.5	W
Topr	Operating free-air temperature range		0～70	℃
Tstg	Storage temperature range		−65～150	℃

RECOMMENDED OPERATING CONDITIONS (Ta＝0～70℃, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{CC}	Supply voltage	4.5	5	5.5	V
V_{DD}	Supply voltage	4.5	5	5.5	V
V_{SS}	Supply voltage		0		V
V_{IH1}	High-level input voltage, except for X_1, X_2, $\overline{RESET}$	2		V_{CC}	V
V_{IH2}	High-level input voltage, X_1, X_2, $\overline{RESET}$	3.8		V_{CC}	V
V_{IL}	Low-level input voltage	−0.5		0.8	V

ELECTRICAL CHARACTERISTICS (Ta＝0～70℃, $V_{CC}=V_{DD}=5V\pm10\%$, $V_{SS}=0V$, unless otherwise noted)

Symbol	Parameter	Conditions	Limits			Unit
			Min	Typ	Max	
V_{OL}	Low-level output voltage, BUS, $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE	I_{OL}＝2mA			0.45	V
V_{OL1}	Low-level output voltage, except for the above and PROG	I_{OL}＝1.6mA			0.45	V
V_{OL2}	Low-level output voltage PROG	I_{OL}＝1mA			0.45	V
V_{OH}	High-level output voltage, BUS, $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE	$I_{OH}=-100\mu A$	2.4			V
V_{OH1}	High-level output voltage, except for the above	$I_{OH}=-50\mu A$	2.4			V
I_{IL}	Input leak current, T1, $\overline{INT}$	$V_{SS}\leqq V_{IN}\leqq V_{CC}$	−10		10	μA
I_{OL}	Output leak current, BUS, TO, high-impedance state	$V_{SS}+0.45\leqq V_{IN}\leqq V_{CC}$	−10		10	μA
I_{LI1}	Input current during low-level input, port	V_{IL}＝0.8V		−0.2		mA
I_{LI2}	Input current during low-level input, $\overline{RESET}$, $\overline{SS}$	V_{IL}＝0.8V		−0.05		mA
I_{DD}	Supply current from V_{DD}	Ta＝25℃		25	50	mA
$I_{DD}+I_{CC}$	Supply current from V_{DD} and V_{CC}	Ta＝25℃		100	170	mA

TIMING REQUIREMENTS (Ta＝0～70℃, $V_{CC}=V_{DD}=5V\pm10\%$, $V_{SS}=0V$, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Limits			Unit
			Min	Typ	Max	
t_C	Cycle time	t_{CY}	2.5		15.0	μs
$t_{h(PSEN-D)}$	Data hold time after $\overline{PSEN}$	t_{DR}	0		200	ns
$t_{h(R-D)}$	Data hold time after $\overline{RD}$	t_{DR}	0		200	ns
$t_{su(PSEN-D)}$	Data setup time after $\overline{PSEN}$	t_{RD}			500	ns
$t_{su(R-D)}$	Data setup time after $\overline{RD}$	t_{RD}			500	ns
$t_{su(A-D)}$	Data setup time after address	t_{AD}			950	ns
$t_{su(PROG-D)}$	Data setup time after PROG	t_{PR}			810	ns
$t_{h(PROG-D)}$	Data hold time before PROG	t_{PF}	0		150	ns

Note 1: The input voltage level of the input voltage is V_{IL}=0.45V and V_{IH}=2.4V.

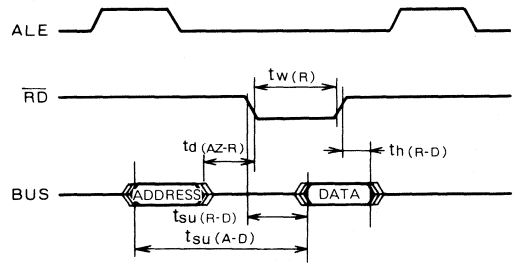
SWITCHING CHARACTERISTICS ($Ta=0\sim70℃$, $V_{CC}=V_{DD}=5V\pm10\%$, $V_{SS}=0V$, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Limits			Unit
			Min	Typ	Max	
$t_{W(ALE)}$	ALE pulse width	t_{LL}	400			ns
$t_{d(A-ALE)}$	Delay time, address to ALE signal	t_{AL}	150			ns
$t_{v(ALE-A)}$	Address valid time after ALE	t_{LA}	80			ns
$t_{W(PSEN)}$	$\overline{PSEN}$ pulse width	t_{CC}	700			ns
$t_{W(R)}$	$\overline{RD}$ pulse width	t_{CC}	700			ns
$t_{W(W)}$	$\overline{WR}$ pulse width	t_{CC}	700			ns
$t_{d(Q-W)}$	Delay time, data to $\overline{WR}$ signal	t_{DW}	500			ns
$t_{v(W-Q)}$	Data valid time after $\overline{WR}$	t_{WD}	120			ns
$t_{d(A-W)}$	Delay time, address to $\overline{WR}$ signal	t_{AW}	230			ns
$t_{d(AZ-R)}$	Delay time, address disable to $\overline{RD}$ signal	t_{AFC}	0			ns
$t_{d(AZ-PSEN)}$	Delay time, address disable to $\overline{PSEN}$ signal	t_{AFC}	0			ns
$t_{d(PC-PROG)}$	Delay time, port control to PROG signal	t_{CP}	110			ns
$t_{v(PROG-PC)}$	Port control valid time after PROG	t_{PC}	130			ns
$t_{p(Q-PROG)}$	Delay time, data to PROG signal	t_{DP}	220			ns
$t_{v(PROG-Q)}$	Data valid time after PROG	t_{PD}	65			ns
$t_{W(PROGL)}$	PROG low pulse width	t_{PP}	1510			ns
$t_{d(Q-ALE)}$	Delay time, data to ALE signal	t_{PL}	400			ns
$t_{v(ALE-Q)}$	Data valid time after ALE	t_{LP}	150			ns

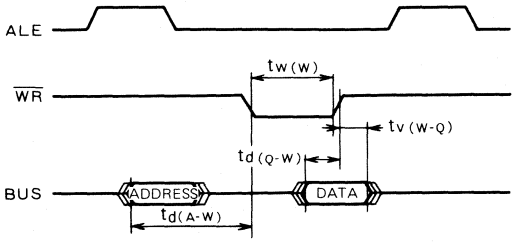
Note 2: Conditions of measurement: control output C_L=80pF
data bus output, port output C_L=150pF, t_C=2.5μS
3: Reference levels for the input/output voltages are low level=0.8V and high level=2V.

TIMING DIAGRAM

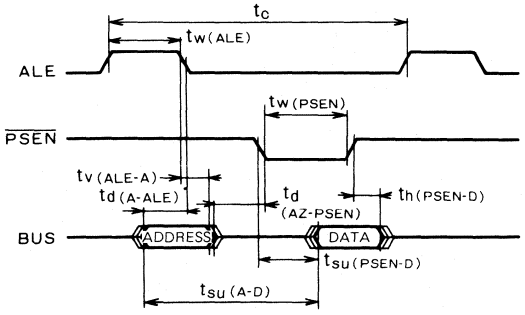
Read from External Data Memory



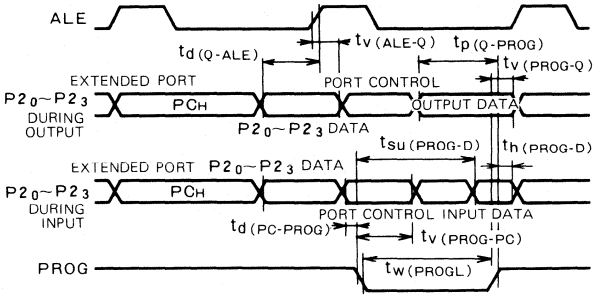
Write to External Data Memory



Instruction Fetch from External Program Memory



Port 2



MITSUBISHI ELECTRIC

MELPS 86 MICROPROCESSORS

8

16-BIT PARALLEL MICROPROCESSOR

DESCRIPTION

The M5L8086S is a 16-bit parallel microprocessor fabricated using high-speed N-channel silicon-gate ED-MOS techology. It requires a single 5V power supply and has a maximum basic clock rate of 5MHz.

The M5L8086S is upward compatible, both in hardware and software, with the M5L8080AP, S and M5L8085AP, S therefore it can replace either of these devices. It has higher performance because of additional and more powerful operation and addressing functions and instructions.

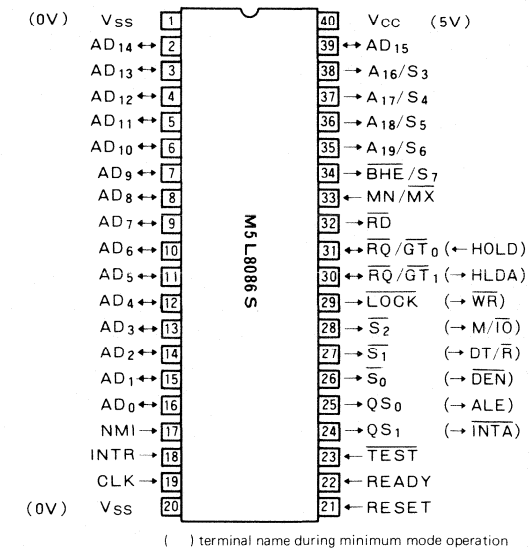
FEATURES

- Direct addressing: 1M byte
- Instruction set upward compatible with that of M5L–8080AP, S
- Enlarged powerful addressing: 24 modes
- On chip 16-bit registers: 14 registers
- Arithmetic operations include multiplication and division, signed or unsigned and 8-bit or 16-bit operands.
- Basic clock rate: 5MHz (max.)
- Multi-CPU functions
- Single 5V power supply
- Interchangeable with the Intel 8086 in pin configuration and electrical characteristics

APPLICATIONS

Central processing unit for 16-bit microcomputer and control units

PIN CONFIGURATION (TOP VIEW)



( ) terminal name during minimum mode operation

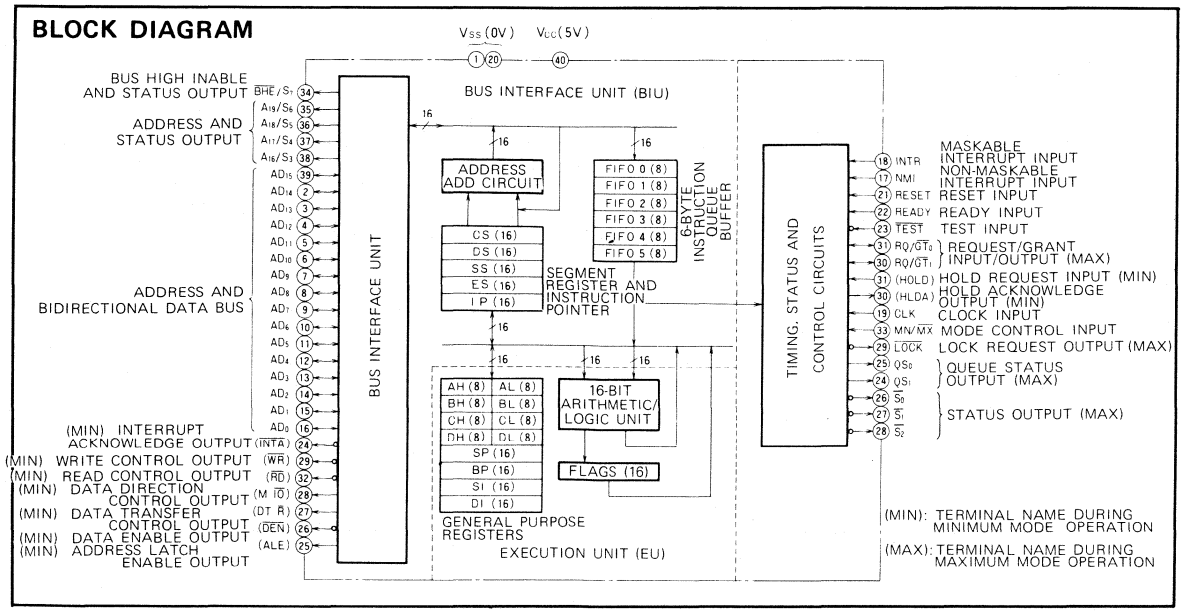
Outline 40 S1

FUNCTIONS

The M5L8086S has a minimum and maximum mode, which allows the composition to be selected to match the scale of the system in which it is used. The internal function consists of execution unit (EU) and bus interface unit (BIU). The BIU controls the 6-byte instruction queue, while generating addresses, and decodes instructions to be executed by the EU. Each unit operates asynchronously and can

8

BLOCK DIAGRAM



16-BIT PARALLEL MICROPROCESSOR

access the instruction queue.

The pipeline architecture increased the throughput of the system. The ability to select 8-bit bytes or 16-bit words by using terminals A_0 and $\overline{BHE}$, allows more efficient use of memory. This along with a large direct addressable memory (up to 1 M bytes) makes it practical to process large complicated programs. Two kinds of external interrupt input are provided. The INTR is a maskable interrupt input for the normal interrupt applications, while the NMI is a nonmaskable interrupt for the use of a higher priority interrupt such as power down. In addition to external interrupts, internal interrupts can be initiated by software with the overflow and so on.

PIN DESCRIPTIONS

Pins which have the same functions in minimum or maximum mode

Pin	Name	Input or Output	Functional description
AD_0~AD_{15}	Address and data bas	Input/output	AD_0~AD_{15} is used as both an address bus (A_0~A_{15}) and a data bus (D_0~D_{15}). Though time sharing it outputs addresses during T_1 state and outputs data during T_2, T_3, T_W, T_4 states.
A_{19}/S_6 \ A_{16}/S_3	Address and status	Output	The high-order 4 bits (A_{16}~A_{19}) and status (S_3~S_6) are output using time sharing techniques. The address bits are output during T_1 state and data are output during T_2, T_3, T_W, T_4 states. The status bits S_3 and S_4 determine which segment register is used in the bus cycle as follows: S_4 S_3 0 0 extra segment 0 1 stack segment 1 0 code segment or none 1 1 data segment while S_5 shows the interrupt enable flag and starts the beginning of a clock cycle. Status bit S_6 is aways 0.
$\overline{BHE}/S_7$	Bus high enable and status	Output	Bus high enable ($\overline{BHE}$) and status are output using time sharing techniques. Bus high enable is output during T_1 state and status is output during T_2, T_3, T_W, T_4 states. $\overline{BHE}$ along with A_0 is used to select byte or word unit processing. The selection is as shown below. $\overline{BHE}$ A_0 0 0 word processing (16 bits) 0 1 high-order byte processing (8 bits) 1 0 low-order byte processing (8 bits) 1 1 undefined This pin goes to low-level during the first clock cycle of an interrupt acknowledge cycle. S_7 is a spare status bit.
$\overline{RD}$	Read control	Output	An active "L" signal indicates read timing from memory or an I/O port.
READY	Ready	Input	Signal indicating data transfer to or from memory and I/O device. When the READY signal is at low level the CPU waits for the signal to go high level. When the signal is at high level the CPU ends the read or write.
INTR	Maskable interrupt request	Input	This signal is sampled at the final clock cycle of each instruction for its level. Enable can be masked by software to inhibit interrupts. An interrupt vector of 256 types can be made using an M5L8259A.
$\overline{TEST}$	Test	Input	The CPU samples this pin while in the wait state. As the result of executing a WAIT instruction this pin is at high level. If the pin is still at high level when sampled the CPU continues to idle until it goes to low level and when that happens the CPU will resume operation.
NMI	Non-maskable interrupt request	Input	This signal is sampled during the final clock cycle of an instruction execution cycle. It is used for urgent interrupts such as power down. A type 2 interrupt is generated by this signal.
RESET	Reset	Input	This signal is used to initialized the CPU. When used it must be maintained at high level for 4 clock cycles to be effective.
CLK	Clock	Input	This signal is used for internal clocking. It is normally attached to the clock output of a M5L8284P or similar device.

Pin Description During Minimum Mode

Pin	Name	Input or output	Functional description
M/$\overline{IO}$	Data direction control	Output	This pin indicates whether the CPU is accessing memory or an I/O device at the time.
$\overline{WR}$	Write control	Output	This signal is used for timing when writing data to external memory or I/O device.
$\overline{INTA}$	Interrupt acknowledge	Output	This pin is used as the read strobe for the interrupt vector on the data bus during the interrupt acknowledge cycle.
ALE	Address latch enable	Output	This signal is the output strobe from the CPU for write address. This is output using time sharing techniques to an external latch.
DT/$\overline{R}$	Data transfer control	Output	This signal indicates the direction of data transfer between the data bus buffer and an external device.
$\overline{DEN}$	Data enable	Output	This signal enables the external data bus buffer.
HOLD	Hold request	Input	When a hold request is received by the CPU it will enter the hold state and surrender control of the data bus at the end of the current instruction execution cycle.
HLDA	Hold acknowledge	Output	This signal shows that the CPU bus accepted a hold request from a peripheral device and that control of the data bus has been surrendered to the peripheral device.

Pin Description During Maximum Mode

Pin	Name	Input or Output	Function description
$\overline{S_0}$ $\overline{S_2}$, $\overline{S_1}$,	Status	Output	$\overline{S_2}$ $\overline{S_1}$ $\overline{S_0}$ 0 0 0 Interrupt acknowledge 0 0 1 Read I/O port 0 1 0 Write I/O port 0 1 1 Hold 1 0 0 Instruction fetch 1 0 1 Read memory 1 1 0 Write memory 1 1 1 Passive cycle
$\overline{RQ}/\overline{GT_0}$ $\overline{RQ}/\overline{GT_1}$	Request/Grant	Input/output	This pin is used by other local bus masters to input a hold request to the CPU and then used to output acknowledge. $\overline{RQ}/\overline{GT_0}$ has higher priority than $\overline{RQ}/\overline{GT_1}$.
$\overline{LOCK}$	Lock request	Output	This signal forbids the use of the system bus by any other system bus masters when the CPU is using the system bus.
QS_1, QS_0	Queue status	Output	The status signal is used for indicating queue operations. QS_0 QS_1 0 0 No operation 0 1 fetch first byte (operation code) of the instruction 1 0 clear the contents of queue 1 1 fetch the next byte of the instruction

8

16-BIT PARALLEL MICROPROCESSOR

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{CC}	Supply voltage	With respect to GND	−1.0～7	V
V_I	Input voltage		−1.0～7	V
Pd	Maximum power dissipation	Ta=25°C	2.5	W
Topr	Operating free-air ambient temperature range		0～70	℃
Tstg	Storage temperature range		−65～150	℃

RECOMMENDED OPERATING CONDITIONS (Ta=0～70°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{CC}	Supply voltage	4.5	5	5.5	V
V_{IH}	High-level input voltage	2.0		V_{CC}+0.5	V
V_{IL}	Low-level input voltage	−0.5		0.8	V
$V_{IH(\phi)}$	High-level clock input voltage	3.9		V_{CC}+1.0	V
$V_{IL(\phi)}$	Low-level clock input voltage	−0.5		0.6	V

ELECTRICAL CHARACTERISTICS (Ta=0～70°C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{OL}	Low-level output voltage	I_{OL}=2mA			0.45	V
V_{OH}	High-level output voltage	I_{OH}=−400μA	2.4			V
I_{CC}	Supply current	Ta=25°C			340	mA
I_{LI}	Input leak current	0V<V_I<V_{CC}			±10	μA
I_{LO}	Output leak current	0.45≦V_O≦V_{CC}			±10	μA
Ci	Input capacitance	f_C=1MHz			10	pF
C_O	Output capacitance	f_C=1MHz			20	pF

TIMING REQUIREMENTS DURING MINIMUM MODE (Ta=0～70°C, V_{CC}=5V±10%, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{C(\phi)}$	Clock cycle time	TCLCL		200		500	ns
$t_{W(\phi L)}$	Clock input low-level pulse width	TCLCH		$\frac{2}{3}t_{C(\phi)}$−15			ns
$t_{W(\phi H)}$	Clock input high-level pulse width	TCHCL		$\frac{1}{3}t_{C(\phi)}$+2			ns
$t_{r(\phi)}$	Clock input rise time	TCH1CH2	V_{IL}=1.0V, V_{IH}=3.5V			10	ns
$t_{f(\phi)}$	Clock input fall time	TCL2CL1	V_{IL}=1.0V, V_{IH}=3.5V			10	ns
$t_{su(DQ-\phi)}$	Data input setup time before clock	TDVCL		30			ns
$t_{h(\phi-DQ)}$	Data input hold time after clock	TCLDZ		10			ns
$t_{su(RDY-\phi)}$	RDY setup time before clock (Note 1, 2)	TRIVCL		35			ns
$t_{h(\phi-RDY)}$	RDY hold time after clock (Note 1, 2)	TCLRIX		0			ns
$t_{su(READY-\phi)}$	READY setup time before clock	TRYHCH		$\frac{2}{3}t_{C(\phi)}$−15			ns
$t_{h(\phi-READY)}$	READY hold time after clock	TCHRYX		30			ns
$t_{su(READY-\phi)}$	READY data invalid setup time before clock (Note 3)	TRYLCL		−8			ns
$t_{su(HOLD-\phi)}$	HOLD setup time before clock	THVCH		35			ns
$t_{su(INTR-\phi)}$ $t_{su(NMI-\phi)}$ $t_{su(TEST-\phi)}$	INTR, NMI, $\overline{TEST}$ setup time before clock (Note 2)	TINVCH		30			ns

SWITCHING CHARACTERISTICS DURING MINIMUM MODE ($T_a = 0 \sim 70°C$, $V_{CC} = 5V \pm 10\%$, $C_L = 20 \sim 100pF$, unless otherwise noted)

Symbol	Parameter	Alternative	Conditions	Limits			Unit
				Min	Typ	Max	
$t_{PXV(\phi\text{-}A)}$	Propagation time, clock to address valid	TCLAV		15		110	ns
$t_{PVX(\phi\text{-}A)}$	Address hold time after clock	TCLAX		10			ns
$t_{PVZ(\phi\text{-}A)}$	Propagation time, clock to address float	TCLAZ		$t_{PVX(A\text{-}\phi)}$		80	ns
$t_{W(ALE)}$	Address iatch enable pulse width	TLHLL		$t_{W(\phi L)}-20$			ns
$t_{PLH(\phi\text{-}ALE)}$	Propagation time, clock to address latch enable	TCLLH				80	ns
$t_{PHL(\phi\text{-}ALE)}$	Propagation time, clock to address latch enable	TCHLL				85	ns
$t_{PVZ(ALE\text{-}A)}$	Propagation time, address latch enable to address float	TLLAZ		$t_{W(\phi H)}-10$			
$t_{PXV(\phi\text{-}DQ)}$	Propagation time, clock to data valid	TCLDV		15		110	ns
$t_{PVZ(\phi\text{-}DQ)}$	Propagation time, clock to data float	TCHDZ		$t_{h(A\text{-}\phi)}$		85	ns
$t_{h(\overline{WR}\text{-}DQ)}$	Data hold time after write	TWHDZ		$t_{W(\phi L)}-30$			ns
$t_{PHL(\phi\text{-}DEN)}$ $t_{PHL(\phi\text{-}\overline{WR})}$ $t_{PHL(\phi\text{-}INTA)}$	Propagation time, clock to data enable, clock to write, clock to $\overline{INTA}$	TCVCTV		10		110	ns
$t_{PHL(\phi\text{-}DT/\overline{R})}$ $t_{PLH(\phi\text{-}DT/\overline{R})}$ $t_{PHL(\phi\text{-}M/\overline{IO})}$ $t_{PHL(\phi\text{-}M/\overline{IO})}$	Propagation time, clock to data send and return control signal, clock to data transfer control signal	TCHCTV		15		110	ns
$t_{PLH(\phi\text{-}\overline{DEN})}$ $t_{PLH(D\text{-}\overline{WR})}$	Propagation time, clock to data enable and write	TCVCTX		10		110	ns
$t_{PHL(A\text{-}\overline{RD})}$	Propagation time, address float to read	TAZRL		0			ns
$t_{PHL(\phi\text{-}\overline{RD})}$	Propagation time, clock to read	TCLRL		10		165	ns
$t_{PLH(\phi\text{-}\overline{RD})}$	Propagation time, clock to read	TCLRH		10		150	ns
$t_{PZV(\overline{RD}\text{-}A)}$	Next cycle address propagation time after read	TRHAV		$t_{C(\phi)}-45$			ns
$t_{PLH(\phi\text{-}HLDA)}$	HLDA propagation time after clock.	TCLHAV		10		160	ns

Note 1: Signal at M5L8284P is shown for reference.
2: Setup time required to be recognized at next clock
3: Requirement during T2 state

TIMING REQUIREMENTS DURING MAXIMUM MODE ($T_a = 0 \sim 70°C$, $V_{CC} = 5V \pm 10\%$, unless otherwise noted)

Symbol	Parameter	Alternative	Conditions	Limits			Unit
				Min	Typ	Max	
$t_{C(\phi)}$	Clock cycle	TCLCL		200		500	ns
$t_{W(\phi L)}$	Clock input low-level pulse width	TCLCH		$\frac{2}{3}t_{C(\phi)}-15$			ns
$t_{W(\phi H)}$	Clock input high-level pulse width	TCHCL		$\frac{1}{3}t_{C(\phi)}+2$			ns
$t_{r(\phi)}$	Clock input rise time	TCH1CH2	$V_{IL}=1.0V$ $V_{IH}=3.5V$			10	ns
$t_{f(\phi)}$	Clock input fall time	TCL2CL1	$V_{IL}=1.0V$ $V_{IH}=3.5V$			10	ns
$t_{su(DQ\text{-}\phi)}$	Data input setup time before clock	TDVCL		30			ns
$t_{h(\phi\text{-}DA)}$	Data input hold time after clock	TCLDZ		10			ns
$t_{su(READY\text{-}\phi)}$	Ready setup time before clock	TRYHCH		35			ns
$t_{h(\phi\text{-}READY)}$	Ready hold time after clock	TCHRYX		0			ns
$t_{su(READY\text{-}\phi)}$	Ready invalid setup time before clock (Note 6)	TRYLCL		$\frac{2}{3}t_{C(\phi)}-15$			ns
$t_{su(RDY\text{-}\phi)}$	RDY setup time before clock (Note 4, 5)	TR1VCL		35			ns
$t_{h(\phi\text{-}RDY)}$	RDY hold time after clock (Note 4, 5)	TCLRIX		40			ns
$t_{su(INTR\text{-}\phi)}$ $t_{su(NMI\text{-}\phi)}$ $t_{su(\overline{TEST}\text{-}\phi)}$	INTRO, NMI, $\overline{TEST}$ setup time before clock	TINVCH		30			ns
$t_{su(\overline{RQ}/\overline{GT}\text{-}\phi)}$	$\overline{RQ}/\overline{GT}$ setup time before clock	TGVCH		30			ns
$t_{h(\phi\text{-}\overline{RQ})}$	$\overline{RQ}$ hold time after clock	TCHGX		30			ns

8

16-BIT PARALLEL MICROPROCESSOR

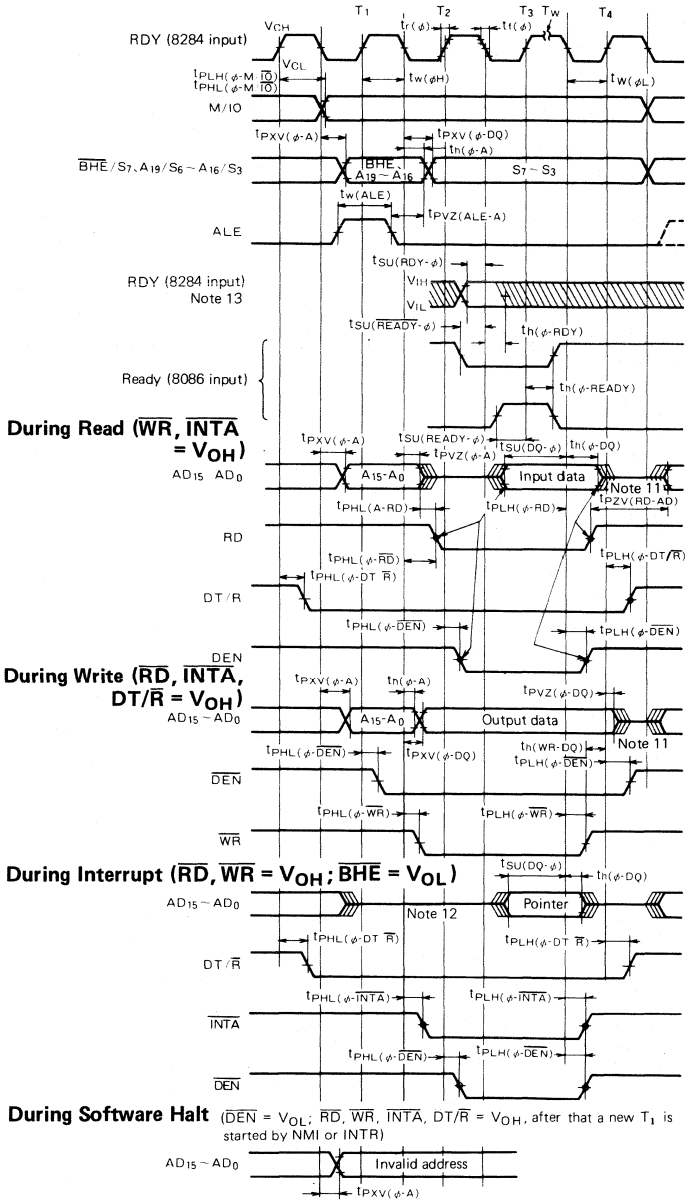
SWITCHING CHARACTERISTICS

MAXIMUM MODE (2) ($Ta = 0 \sim 70°C$, $V_{CC} = 5V \pm 10\%$, $C_L = 20 \sim 100pF$, unless otherwise noted)

Symbol	Parameter	Alternative	Conditions	Limits			Unit
				Min	Typ	Max	
t_{PHL} (ϕ-$\overline{MROC}$) t_{PHL} (ϕ-$\overline{IORC}$) t_{PHL} (ϕ-$\overline{AIOWC}$) t_{PHL} (ϕ-$\overline{AMWC}$) t_{PHL} (ϕ-$\overline{INTA}$) t_{PHL} (ϕ-$\overline{MWTC}$) t_{PHL} (ϕ-$\overline{IOWC}$)	Propagation time, clock to $\overline{MRDC}$, $\overline{IORC}$, $\overline{AIOWC}$, $\overline{AMWC}$, $\overline{INTA}$, $\overline{MWTC}$, $\overline{IOWC}$ (Note 4)	TCLML	$C_L = 80pF$	10		35	ns
t_{PLH} (ϕ-$\overline{MRDC}$) t_{PLH} (ϕ-$\overline{IORC}$) t_{PLH} (ϕ-$\overline{AIOWC}$) t_{PLH} (ϕ-$\overline{AMWC}$) t_{PLH} (ϕ-$\overline{INTA}$) t_{PLH} (ϕ-$\overline{MWTC}$) t_{PLH} (ϕ-$\overline{IOWC}$)	Propagation time, clock to $\overline{MRDC}$, $\overline{IORC}$, $\overline{AIOWC}$, $\overline{AMWC}$, $\overline{INTA}$, $\overline{MWTC}$, $\overline{IOWC}$ (Note 4)	TCLMH		10		35	ns
t_{PVZ} (RDY-S)	Propagation time, RDY to status 3~7 float (Note 6)	TRYHSH				110	ns
t_{PHL} (ϕ-$\overline{S}$)		TCHSV		10		110	ns
t_{PLH} (ϕ-$\overline{S}$)	Propagation time, clock to status 0~2	TCLSH		10		130	ns
t_{PLH} (ϕ-QS) t_{PHL} (ϕ-QS) t_{PXV} (ϕ-A) t_{PHL} (ϕ-$\overline{LOCK}$) t_{PLH} (ϕ-$\overline{LOCK}$)	Propagation time, clock to queue status, address lock	TCLAV		15		110	ns
t_{PVX} (ϕ-A)	Propagation time, clock to address	TCLAX		10			ns
t_{PVZ} (ϕ-A)	Propagation time, clock to address float	TCLAZ		$t_{PVX(\phi\text{-}A)}$		80	ns
t_{PLH} ($\overline{S}$-ALE)	Propagation time, status 0~2 to address latch enable (Note 4)	TSVLH				15	ns
t_{PLH} ($\overline{S}$-$\overline{MCE}$) t_{PLH} ($\overline{S}$-$\overline{PDEN}$)	Propagation time, status 0~2 to MCE, $\overline{PDEN}$ (Note 4)	TSVMCH				15	ns
t_{PLH} (ϕ-ALE)	Propagation time, clock to address latch enable (Note 4)	TCLLH				15	ns
t_{PLH} (ϕ-$\overline{MCE}$) t_{PLH} (ϕ-$\overline{PDEN}$)	Propagation time, clock to MCE, $\overline{PDEN}$ (Note 4)	TCLMCH				15	ns
t_{PHL} (ϕ-ALE)	Propagation time, clock to address latch enable (Note 4)	TCHLL				15	ns
t_{PHL} (ϕ-MCE) t_{PHL} (ϕ-PDEN)	Propagation time, clock to MCE, $\overline{PDEN}$ (Note 4)	TCLMCL				15	ns
t_{PXV} (ϕ-DQ) t_{PXV} (ϕ-S)	Propagation time, clock to data and status 3~7 valid	TCLDV		15		110	ns
t_{PVZ} (ϕ-DQ) t_{PVZ} (ϕ-S)	Propagation time, clock to data and status 3~7 float	TCHDZ		$t_{PVX(\phi\text{-}A)}$		85	ns
t_{PLH} (ϕ-$\overline{DEN}$)	Propagation time, clock to $\overline{DEN}$ (Note 4)	TCVNV		5		45	ns
t_{PHL} (ϕ-$\overline{DEN}$)	Propagation time, clock to $\overline{DEN}$ (Note 4)	TCVNX		10		45	ns
t_{PHL} (A-$\overline{RD}$)	Propagation time, address float to read	TAZRL		0			ns
t_{PHL} (ϕ-$\overline{RD}$)	Propagation time, clock to read	TCLRL		10		165	ns
t_{PLH} (ϕ-$\overline{RD}$)	Propagation time, clock to read	TCLRH		10		150	ns
t_{PZV} ($\overline{RD}$-A)	Propagation time, invalid read to next address	TRHAV		$t_{C(\phi)}$-45			ns
t_{PHL} (ϕ-DT/$\overline{R}$)	Propagation time, clock to data S/R control (Note 4)	TCHDTL				50	ns
t_{PLH} (ϕ-DT/$\overline{R}$)	Propagation time, clock to data S/R control (Note 4)	TCHDTH				30	ns
t_{PHL} (ϕ-$\overline{GT}$)	Propagation time, clock to data S/R control (Note 4)	TCLGL	$C_L = 30pF$			85	ns
t_{PLH} (ϕ-$\overline{GT}$)	Propagation time, clock to data S/R control (Note 4)	TCLGH	$C_L = 30pF$			85	ns

Note 4: Signal of M5L8284P is shown for reference.
5: Setup time required to be recognized at next clock.
6: Applies only to T_3 and wait states.
7: Applies only to T_2 states.

TIMING DIAGRAM (During Minimum Mode)



Note 8: the center line indicates floating (high-impedance) state.

9: Input signal is entered within the range of V_{OL}~V_{OH} unless otherwise noted.

10: When the T_W state is entered the RDY signal is sampled near the end of T_2, T_3 and T_W.

11: Only when the M5L8086S enters a hold acknowledge cycle does the local bus go to a floating state after a write cycle.

12: An interrupt cycle requires 2 clock cycles. The AD bus goes to a floating state during the second cycle of an interrupt.

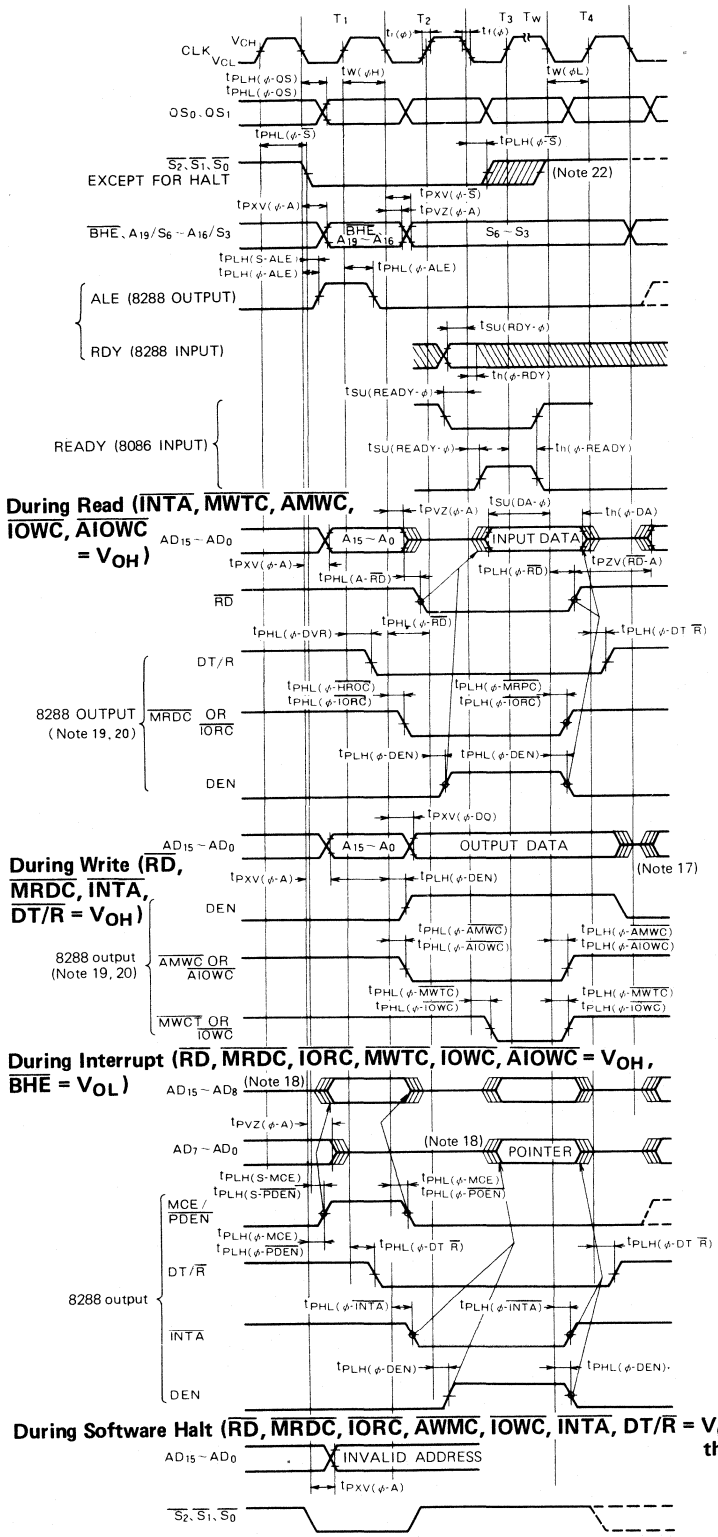
13: Signals of the M5L8284P are shown for reference.

14: All timing signals are tested at 1.5V unless otherwise noted.

8

16-BIT PARALLEL MICROPROCESSOR

TIMING DIAGRAM (During Maximum Mode)



Note 15: Input signals are entered within the range V_{OL}~V_{OH} unless otherwise noted.

16: When the T_w state is entered the RDY signal is sampled near the end of T_2, T_3 and T_w.

17: Only when the M5L8086S enters a hold acceptance cycle does the local bus go to a floaling state after a write cycle.

18: An interrupt cycle requires 2 cycles. The AD bus goes to a floating state during the second cycle of an interrupt.

19: Signals of the M5L8284P and M5L8288 are shown for reference.

20: The M5L8288 sends a command and a control signal soon after the CEN signal.

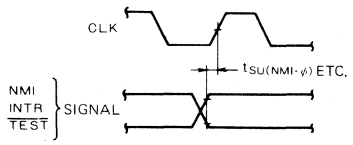
21: All timing signals are tested at 1.5V unless otherwise noted.

22: Status is invalid just before T_4 state.

MITSUBISHI ELECTRIC

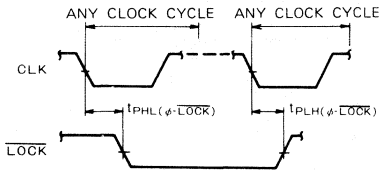
TIMING DIAGRAM

Asynchronous Signal Recognition Timing

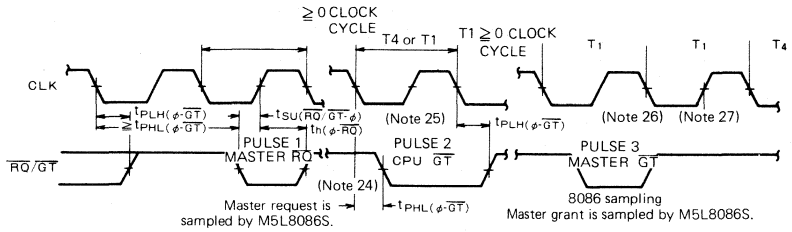


Note 23: Setup time required for being recognized in the next cycle.

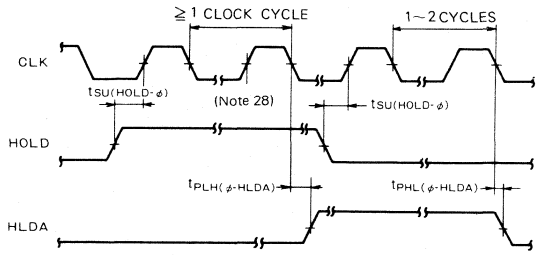
Bus Lock Signal Timing (For Maximum Mode Only)



Request/Grant Sequence Timing (For Maximum Mode Only)



Hold Acknowledge Timing (For Minimum Mode Only)



Note 24: S_2, S_1 and S_0 are changed to floating from the states of (1, 1, 1) at this edge.
25: AD bus, $\overline{RD}$ and $\overline{LOCK}$ are changed to floating at this edge.
26: S_2, S_1 and S_0 of the other master are changed to floating from the states of (1, 1, 1) at this edge.
27: AD bus, $\overline{RD}$ and $\overline{LOCK}$ of the other master are changed to floating at this edge.
28: Bus is changed to floating at this edge.

8

16-BIT PARALLEL MICROPROCESSOR

MACHINE INSTRUCTION INSTRUCTION SET SUMMARY

Item / Type of instruction		Mnemonic	Instruction code								Hexadecimal notation
			D_7 D_6 D_5 D_4	D_3 D_2 D_1 D_0	D_7 D_6	D_5 D_4 D_3	D_2 D_1 D_0				
Data transfers	General transfers	MOV EA1/r1, EA2/r2	1 0 0 0	1 0 d W	MOD	REG	R/M				88~8B
			[DISP-L]		[DISP-H]						
		MOV r1, r2	1 0 0 0	1 0 d W	1 1	REG	R/M				
		MOV r1, EA2	1 0 0 0	1 0 1 W	MOD	REG	R/M				
			[DISP-L]		[DISP-H]						
		MOV EA1, r2	1 0 0 0	1 0 0 W	MOD	REG	R/M				
			[DISP-L]		[DISP-H]						
		MOV EA1/r1, DATA	1 1 0 0	0 1 1 W	MOD	0 0 0	R/M				C6~C7
			[DISP-L]		[DISP-H]						
			[DATA-L]		[DATA-H]						
		MOV r1, DATA	1 1 0 0	0 1 1 W	1 1	0 0 0	R/M				
			[DATA-L]		[DATA-H]						
		MOV EA1, DATA	1 1 0 0	0 1 1 W	MOD	0 0 0	R/M				
			[DISP-L]		[DISP-H]						
			[DATA-L]		[DATA-H]						
		MOV r1, DATA	1 0 1 1	W REG	[DATA-L]						B0~BF
			[DATA-H]								
		MOV Acc, ADDR	1 0 1 0	0 0 0 W	[ADDR-L]						A0~A1
			[ADDR-H]								
		MOV ADDR, Acc	1 0 1 0	0 0 1 W	[ADDR-L]						A2~A3
			[ADDR-H]								
		MOV SEG, EA2/r2	1 0 0 0	1 1 1 0	MOD	0 SR	R/M				8E
			[DISP-L]		[DISP-H]						
		MOV SEG, r2	1 0 0 0	1 1 1 0	1 1	0 SR	R/M				
		MOV SEG, EA2	1 0 0 0	1 1 1 0	MOD	0 SR	R/M				
			[DISP-L]		[DISP-H]						
		MOV r1/EA1, SEG	1 0 0 0	1 1 0 0	MOD	0 SR	R/M				8C
			[DISP-L]		[DISP-H]						
		MOV r1, SEG	1 0 0 0	1 1 0 0	1 1	0 SR	R/M				
		MOV EA1, SEG	1 0 0 0	1 1 0 0	MOD	0 SR	R/M				
			[DISP-L]		[DISP-H]						
		XCHG r1, EA2/r2	1 0 0 0	0 1 1 W	MOD	REG	R/M				86~87
			[DISP-L]		[DISP-H]						
		XCHG r1, r2	1 0 0 0	0 1 1 W	1 1	REG	R/M				
		XCHG r1, EA2	1 0 0 0	0 1 1 W	MOD	REG	R/M				
			[DISP-L]		[DISP-H]						
		XCHG AX, r2	1 0 0 1	0 REG							90~97
		XLAT m	1 1 0 1	0 1 1 1							D7
	Stack operations	PUSH EA1/r1	1 1 1 1	1 1 1 1	MOD	1 1 0	R/M				FF
			[DISP-L]		[DISP-H]						
		PUSH r1	1 1 1 1	1 1 1 1	1 1	1 1 0	R/M				
		PUSH EA1	1 1 1 1	1 1 1 1	MOD	1 1 0	R/M				
			[DISP-L]		[DISP-H]						
		PUSH r1	0 1 0 1	0 REG							50~57
		PUSH SEG	0 0 0 SR	1 1 0							06, 0E, 16, 1E
		POP EA1/r1	1 0 0 0	1 1 1 1	MOD	0 0 0	R/M				8F
			[DISP-L]		[DISP-H]						
		POP r1	1 0 0 0	1 1 1 1	1 1	1 0 0	R/M				
		POP EA1	1 0 0 0	1 1 1 1	MOD	0 0 0	R/M				
			[DISP-L]		[DISP-H]						
		POP r1	0 1 0 1	1 REG							58~5F
		POP SEG	0 0 0 SR	1 1 1							07, 0F, 17, 1F
	Input/output	IN Acc, Port	1 1 1 0	0 1 0 W	[PORT]						E4~E5
		IN Acc, DX	1 1 1 0	1 1 0 W							EC~ED
		OUT Port, Acc	1 1 1 0	0 1 1 W	[PORT]						E6~E7
		OUT DX, Acc	1 1 1 0	1 1 1 W							EE~EF
	Address transfers	LEA r1, EA2	1 0 0 0	1 1 0 1	MOD	REG	R/M				8D
			[DISP-L]		[DISP-H]						
		LDS r1, EA2	1 1 0 0	0 1 0 1	MOD	REG	R/M				C5
			[DISP-L]		[DISP-H]						
		LES r1, EA2	1 1 0 0	0 1 0 0	MOD	REG	R/M				C4
			[DISP-L]		[DISP-H]						
	Flag transfers	LAHF	1 0 0 1	1 1 1 1							9F
		SAHF	1 0 0 1	1 1 1 0							9E
		POPF	1 0 0 1	1 1 0 1							9D
		PUSHF	1 0 0 1	1 1 0 0							9C

Clock cycles	Bytes in the code	Bus cycles	Function description	Flags								
				OF	DF	IF	TF	SF	ZF	AF	PF	CF
				X	X	X	X	X	X	X	X	X
2	2	0	(r1)←(r2)									
8+EA	2~4	1	(r1)←(EA2) MOD≠11									
9+EA	2~4	1	(EA1)←(r2) MOD≠11									
				X	X	X	X	X	X	X	X	X
4	3~4	0	(r1)←DATA									
10+EA	3~6	1	(EA1)←DATA MOD≠11									
4	2~3	0	(r1)←DATA	X	X	X	X	X	X	X	X	X
10	3	1	(A_{CC})←(ADDR)	X	X	X	X	X	X	X	X	X
10	3	1	(ADDR)←(A_{CC})	X	X	X	X	X	X	X	X	X
			When SR = 01: undefined	X	X	X	X	X	X	X	X	X
2	2	0	(SEG)←(r2)									
8+EA	2~4	1	(SEG)←(EA2) MOD≠11									
				X	X	X	X	X	X	X	X	X
2	2	0	(r1)←(SEG)									
9+EA	2~4	1	(EA1)←(SEG) MOD≠11									
				X	X	X	X	X	X	X	X	X
4	2	0	(r1)←→(r2)									
17+EA	2~4	1	(r1)←→(EA2) MOD≠11									
3	1	0	(AX)←→(r2)	X	X	X	X	X	X	X	X	X
11	1	1	(AL)←((BX)+(AL)) : (m)	X	X	X	X	X	X	X	X	X
				X	X	X	X	X	X	X	X	X
11	2	1	(SP)←(SP)−2, ((SP)+1 : (SP))←(r1)									
16+EA	2~4	2	(SP)←(SP)−2, ((SP)+1 : (SP))←(EA1) MOD≠11									
10	1	1	(SP)←(SP)−2, ((SP)+1 : (SP))←(r1)	X	X	X	X	X	X	X	X	X
10	1	1	(SP)←(SP)−2, ((SP)+1 : (SP))←(SEG) When SR = 01: undefined	X	X	X	X	X	X	X	X	X
				X	X	X	X	X	X	X	X	X
8	2	1	(r1)←((SP)+1 : (SP)), (SP)←(SP)+2									
17+EA	2~4	2	(EA1)←((SP)+1 : (SP)), (SP)←(SP)+2 MOD≠11									
8	1	1	(r1)←((SP)+1 : (SP)), (SP)←(SP)+2	X	X	X	X	X	X	X	X	X
8	1	1	(SEG)←((SP)+1 : (SP)), (SP)←(SP)+2 When SR = 01 : undefined	X	X	X	X	X	X	X	X	X
10	2	1	(A_{CC})←(Port)	X	X	X	X	X	X	X	X	X
8	1	1	(A_{CC})←((DX))	X	X	X	X	X	X	X	X	X
10	2	1	(Port)←(A_{CC})	X	X	X	X	X	X	X	X	X
8	1	1	((DX))←(A_{CC})	X	X	X	X	X	X	X	X	X
2+EA	2~4	0	(r1)←EA2 When MOD = 11: undefined	X	X	X	X	X	X	X	X	X
16+EA	2~4	2	(r1)←(EA2), (DS)←(EA2+2) When MOD = 11: undefined	X	X	X	X	X	X	X	X	X
16+EA	2~4	2	(r1)←(EA2), (ES)←(EA2+2) When MOD = 11: undefined	X	X	X	X	X	X	X	X	X
4	1	0	(AH)←(SF) : (ZF) : X : (AF) : X : (PF) : X : (CF)	X	X	X	X	X	X	X	X	X
4	1	0	(SF) : (ZF) : X : (AF) : X : (PF) : X : (CF)←(AH)	X	X	X	X	○	○	○	○	○
8	1	1	(FR)←((SP)+1 : (SP)), (SP)←(SP)+2	○	○	○	○	○	○	○	○	○
10	1	1	(SP)←(SP)−2, ((SP)+1 : (SP))←(FR)	X	X	X	X	X	X	X	X	X

8

16-BIT PARALLEL MICROPROCESSOR

Item / Type of instruction		Mnemonic	Instruction code			Hexadecimal notation
			D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0		
Arithmetic instructions	Addition and related	[ADD EA1/r1, EA2/r2]	0 0 0 0 0 0 d W [DISP-L]	MOD REG R/M [DISP-H]		00~03
		ADD r1,r2	0 0 0 0 0 0 d W	1 1 REG R/M		
		ADD r1, EA2	0 0 0 0 0 0 1 W [DISP-L]	MOD REG R/M [DISP-H]		
		ADD EA1,r2	0 0 0 0 0 0 0 W [DISP-L]	MOD REG R/M [DISP-H]		
		[ADD EA1/r1, DATA]	1 0 0 0 0 0 S W [DISP-L] [DATA-L]	MOD 0 0 0 R/M [DISP-H] [DATA-H]		80~83
		ADD r1, DATA	1 0 0 0 0 0 S W [DATA-L]	1 1 0 0 0 R/M [DATA-H]		
		ADD EA1, DATA	1 0 0 0 0 0 S W [DISP-L] [DATA-L]	MOD 0 0 0 R/M [DISP-H] [DATA-H]		
		ADD Acc, DATA	0 0 0 0 0 1 0 W [DATA-H]	[DATA-L]		04~05
		[ADC EA1/r1, EA2/r2]	0 0 0 1 0 0 d W [DISP-L]	MOD REG R/M [DISP-H]		10~13
		ADC r1,r2	0 0 0 1 0 0 d W	1 1 REG R/M		
		ADC r1, EA2	0 0 0 1 0 0 1 W [DISP-L]	MOD REG R/M [DISP-H]		
		ADC EA1,r2	0 0 0 1 0 0 0 W [DISP-L]	MOD REG R/M [DISP-H]		
		[ADC EA1/r1, DATA]	1 0 0 0 0 0 S W [DISP-L] [DATA-L]	MOD 0 1 0 R/M [DISP-H] [DATA-H]		80~83
		ADC r1, DATA	1 0 0 0 0 0 S W [DATA-L]	1 1 0 1 0 R/M [DATA-H]		
		ADC EA1, DATA	1 0 0 0 0 0 S W [DISP-L] [DATA-L]	MOD 0 1 0 R/M [DISP-H] [DATA-H]		
		ADC Acc, DATA	0 0 0 1 0 1 0 W [DATA-H]	[DATA-L]		14~15
		[INC EA1/r1]	1 1 1 1 1 1 1 W [DISP-L]	MOD 0 0 0 R/M [DISP-H]		FE~FF
		INC r1	1 1 1 1 1 1 1 W	1 1 0 0 0 R/M		
		INC EA1	1 1 1 1 1 1 1 W [DISP-L]	MOD 0 0 0 R/M [DISP-H]		
		INC r1	0 1 0 0 0 REG			40~47
		AAA	0 0 1 1 0 1 1 1			37
		DAA	0 0 1 0 0 1 1 1			27

MITSUBISHI ELECTRIC

Clock cycles	Bytes in the code	Bus cycles	Function description	Flags (OF DF IF TF SF ZF AF PF CF)
				○ X X X ○ ○ ○ ○ ○
3	2	0	(r1)← (r1)+(r2)	
9+EA	2~4	1	(r1)← (r1)+(EA2) MOD≠11	
16+EA	2~4	2	(EA1)← (EA1)+(r2) MOD≠11	
			When S:W = 01 then DATA is DATA-L and DATA-H When S:W = 11 then DATA is DATA-L and DATA-H is filled with the sign of DATA-L (sign extended)	○ X X X ○ ○ ○ ○ ○
4	3~4	0	(r1)← (r1)+DATA	
17+EA	3~6	2	(EA1)← (EA1)+DATA MOD≠11	
4	2~3	0	(Acc)← (Acc)+DATA	○ X X X ○ ○ ○ ○ ○
				○ X X X ○ ○ ○ ○ ○
3	2	0	(r1)← (r1)+(r2) +1	
9+EA	2~4	1	(r1)← (r1)+(EA2) +1 MOD≠11	
16+EA	2~4	2	(EA1)← (EA1)+(r2) +1 MOD≠11	
			When S:W = 01 then DATA is DATA-L and DATA-H When S:W = 11 then DATA is DATA-L and DATA-H is filled with the sign of DATA-L (sign extended)	○ X X X ○ ○ ○ ○ ○
4	3~4	0	(r1)← (r1)+DATA +1	
17+EA	3~6	2	(EA1)← (EA1)+DATA +1	
4	2~3	0	(Acc)← (Acc)+DATA +1	○ X X X ○ ○ ○ ○
				○ X X X ○ ○ ○ ○
3	2	0	(r1)← (r1)+1	
15+EA	2~4	2	(EA1)← (EA1)+1 MOD≠11	
2	1	0	(r1)← (r1)+1	○ X X X ○ ○ ○ ○ X
4	1	0	When (AL) $0F_{16}$ or (CF) = 1: (AL)← (AL)+(AL)+60_{16} (AH)← (AH)+1, (AF)← 1, (CF)← (AF) (AL)← (AL) ∧ $0F_{16}$	△ X X X △ △ ○ △ ○
4	1	0	When (AL) $0F_{16}$ or (CF) = 1: (AL)← (AL)+(AL)+60_{16} (CF)← (AF)∨(CF), (AF)←1 When (AL) > $9F_{16}$ or (CF) = 1: (AL)← (AL)+60_{16} (CF)←1	△ X X X ○ ○ ○ ○ ○

8

16-BIT PARALLEL MICROPROCESSOR

Type of instruction	Mnemonic	Instruction code			Hexadecimal notation
		D_7 D_6 D_5 D_4	D_3 D_2 D_1 D_0	D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	
Arithmetic instructions (Cont'd) / Subtraction and related	[SUB EA1/r1, EA2/r2]	0 0 1 0	1 0 d W	MOD REG R/M	28~2B
		[DISP-L]		[DISP-H]	
	SUB r1, r2	0 0 1 0	1 0 d W	1 1 REG R/M	
	SUB r1, EA2	0 0 1 0	1 0 1 W	MOD REG R/M	
		[DISP-L]		[DISP-H]	
	SUB EA1, r2	0 0 1 0	1 0 0 W	MOD REG R/M	
		[DISP-L]		[DISP-H]	
	[SUB EA1/r1, DATA]	1 0 0 0	0 0 S W	MOD 1 0 1 R/M	80~83
		[DISP-L]		[DISP-H]	
		[DATA-L]		[DATA-H]	
	SUB r1, DATA	1 0 0 0	0 0 S W	1 1 1 0 1 R/M	
		[DATA-L]		[DATA-H]	
	SUB EA1, DATA	1 0 0 0	0 0 S W	MOD 1 0 1 R/M	
		[DISP-L]		[DISP-H]	
		[DATA-L]		[DATA-H]	
	SUB Acc, DATA	0 0 1 0	1 1 0 W	[DATA-L]	2C~2D
		[DATA-H]			
	[SBB EA1/r1, EA2/r2]	0 0 0 1	1 0 d W	MOD REG R/M	18~1B
		[DISP-L]		[DISP-H]	
	SBB r1, r2	0 0 0 1	1 0 d W	1 1 REG R/M	
	SBB r1, EA2	0 0 0 1	1 0 1 W	MOD REG R/M	
		[DISP-L]		[DISP-H]	
	SBB EA1, r2	0 0 0 1	1 0 0 W	MOD REG R/M	
		[DISP-L]		[DISP-H]	
	[SBB EA1/r1, DATA]	1 0 0 0	0 0 S W	MOD 0 1 1 R/M	80~83
		[DISP-L]		[DISP-H]	
		[DATA-L]		[DATA-H]	
	SBB r1, DATA	1 0 0 0	0 0 S W	1 1 0 1 1 R/M	
		[DATA-L]		[DATA-H]	
	SBB EA1, DATA	1 0 0 0	0 0 S W	MOD 0 1 1 R/M	
		[DISP-L]		[DISP-H]	
		[DATA-L]		[DATA-H]	
	SBB Acc, DATA	0 0 0 1	1 1 0 W	[DATA-L]	1C~1D
		[DATA-H]			
	[DEC EA1/r1]	1 1 1 1	1 1 1 W	MOD 0 0 1 R/M	FE~FF
		[DISP-L]		[DISP-H]	
	DEC r1	1 1 1 1	1 1 1 W	1 1 0 0 1 R/M	
	DEC EA1	1 1 1 1	1 1 1 W	MOD 0 0 1 R/M	
		[DISP-L]		[DISP-H]	
	DEC r1	0 1 0 0	1 REG		48~4F
	[NEG EA1/r1]	1 1 1 1	0 1 1 W	MOD 0 1 1 R/M	F6~F7
		[DISP-L]		[DISP-H]	
	NEG r1	1 1 1 1	0 1 1 W	1 1 0 1 1 R/M	
	NEG EA1	1 1 1 1	0 1 1 W	MOD 0 1 1 R/M	
		[DISP-L]		[DISP-H]	
	[CMP EA1/r1, EA2/r2]	0 0 1 1	1 0 d W	MOD REG R/M	38~3B
		[DISP-L]		[DISP-H]	
	CMP r1, r2	0 0 1 1	1 0 d W	1 1 REG R/M	
	CMP r1, EA2	0 0 1 1	1 0 1 W	MOD REG R/M	
		[DISP-L]		[DISP-H]	
	CMP EA1, r2	0 0 1 1	1 0 0 W	MOD REG R/M	
		[DISP-L]		[DISP-H]	
	[CMP EA1/r1, DATA]	1 0 0 0	0 0 S W	MOD 1 1 1 R/M	80~83
		[DISP-L]		[DISP-H]	
		[DATA-L]		[DATA-H]	
	CMP r1, DATA	1 0 0 0	0 0 S W	1 1 1 1 1 R/M	
		[DATA-L]		[DATA-H]	
	CMP EA1, DATA	1 0 0 0	0 0 S W	MOD 1 1 1 R/M	
		[DISP-L]		[DISP-H]	
		[DATA-L]		[DATA-H]	
	CMP Acc, DATA	0 0 1 1	1 1 0 W	[DATA-L]	3C~3D
		[DATA-H]			
	AAS	0 0 1 1	1 1 1 1		3F
	DAS	0 0 1 0	1 1 1 1		2F

MITSUBISHI ELECTRIC

Clock cycles	Bytes in the code	Bus cycles	Function description	Flags (OF DF IF TF SF ZF AF PF CF)
				○ X X X ○ ○ ○ ○ ○
3	2	0	(r1)←(r1)−(r2)	
9+EA	2~4	1	(r1)←(r1)−(EA2) MOD≠11	
16+EA	2~4	2	(EA1)←(EA1)−(r2) MOD≠11	
			When S:W = 01 then DATA is DATA-L and DATA-H When S:W = 11 then DATA is DATA-L and the signs of DATA-L are extended to form 16-bit operand.	○ X X X ○ ○ ○ ○ ○
4	3~4	0	(r1)←(r1)−DATA	
17+EA	3~6	2	(EA1)←(EA1)−DATA MOD≠11	
4	2~3	0	(Acc) ←(Acc)−DATA	○ X X X ○ ○ ○ ○ ○
				○ X X X ○ ○ ○ ○ ○
3	2	0	(r1)←(r1)−(r2) −1 when (CF) = 1	
9+EA	2~4	1	(r1)←(r1)−(EA2) −1 when (CF) = 1 MOD≠11	
16+EA	2~4	2	(EA1)←(EA1)−(r2) −1 when (CF) = 1 MOD≠11	
			When S:W = 01 then DATA is DATA-L and DATA-H When S:W = 11 then DATA is DATA-L and the sign of DATA-L is extended to form a 16-bit operand.	○ X X X ○ ○ ○ ○ ○
4	3~4	0	(r1)←(r1)−DATA −1 when (CF) = 1	
17+EA	3~6	2	(EA1)←(EA1)−DATA −1 when (CF) = 1 MOD≠11	
4	2~3	0	(Acc)←(Acc)−DATA when (CF) = 1	○ X X X ○ ○ ○ ○ ○
				○ X X X ○ ○ ○ ○ X
3	2	0	(r1)←(r1)−1	
15+EA	2~4	2	(EA1)←(EA1)−1 MOD≠11	
2	1	0	(r1)←(r1)−1	○ X X X ○ ○ ○ ○ X
			When W=0 (SRC)=FFH When W=1 (SRC)=FFFFH	○ X X X ○ ○ ○ ○ 1
3	2	0	(r1)←(SRC)−(r1), (r1)←0−(r1)	
16+EA	2~4	4	(EA1)←(SRC)−(EA1), (EA1)←0−(SRC) MOD≠11	
				○ X X X ○ ○ ○ ○ ○
3	2	0	(r1)−(r2)	
9+EA	2~4	1	(r1)−(EA2) MOD≠11	
16+EA	2~4	2	(EA1)−(r2) MOD≠11	
			When S:W = 01 then DATA is DATA-L and DATA-H When S:W = 11 then DATA is DATA-L and the signs of DATA-L are extended to form 16-bit operand.	○ X X X ○ ○ ○ ○ ○
4	3~4	0	(r1)−DATA	
17+EA	3~6	2	(EA1)−DATA MOD≠11	
4	2~3	0	(Acc)−DATA	○ X X X ○ ○ ○ ○ ○
4	1	0	When ((AL) ∨ $0F_{16}$) or (AF) = 1: When (AL) ∨ $0F_{16}$ or (AF) = 1: (AL)←(AL) −6 (AH)←(AH)−1, (AF)←1, (CF)←(AF) (AL)←(AL) ∨ $0F_{16}$	△ X X X ∧ ∧ ○ ∧ ○
4	1	0	When ((AL) ∨ $0F_{16}$) or (AF) = 1: When (AL) ∨ $0F_{16}$ or (AF) = 1: (AL)←(AL) −6 (CF)←(AF)∨(CF), (AF)←1 When (AL) > $9F_{16}$ or (CF) = 1: (AL)←(AL) −60_{16} (CF)←1	△ X X X ○ ○ ○ ○ ○

8

16-BIT PARALLEL MICROPROCESSOR

Type of instruction		Mnemonic	Instruction code						Hexadecimal notation
			D_7 D_6 D_5 D_4	D_3 D_2 D_1 D_0	D_7 D_6	D_5 D_4 D_3	D_2 D_1 D_0		
Arithmetic instructions (cont'd)	Multiplication and related	[MUL EA1/r1]	1 1 1 1	0 1 1 W	MOD	1 0 0	R/M		F6~F7
		[DISP-L]			[DISP-H]				
		MUL r1	1 1 1 1	0 1 1 0	1 1	1 0 0	R/M		
			1 1 1 1	0 1 1 1	1 1	1 0 0	R/M		
		MUL EA1	1 1 1 1	0 1 1 0	MOD	1 0 0	R/M		
			[DISP-L]		[DISP-H]				
			1 1 1 1	0 1 1 1	MOD	1 0 0	R/M		
			[DISP-L]		[DISP-H]				
		[IMUL EA1/r1]	1 1 1 1	0 1 1 W	MOD	1 0 1	R/M		F6~F7
			[DISP-L]		[DISP-H]				
		IMUL r1	1 1 1 1	0 1 1 0	1 1	1 0 1	R/M		
			1 1 1 1	0 1 1 1	1 1	1 0 1	R/M		
		IMUL EA1	1 1 1 1	0 1 1 0	MOD	1 0 1	R/M		
			[DISP-L]		[DISP-H]				
			1 1 1 1	0 1 1 1	MOD	1 0 1	R/M		
			[DISP-L]		[DISP-H]				
		AAM	1 1 0 1	0 1 0 0	0 0	0 0 1	0 1 0		D4
	Division and related	[DIV EA1/r1]	1 1 1 1	0 1 1 W	MOD	1 1 0	R/M		F6~F7
			[DISP-L]		[DISP-H]				
		DIV r1	1 1 1 1	0 1 1 0	1 1	1 1 0	R/M		
			1 1 1 1	0 1 1 1	1 1	1 1 0	R/M		
		DIV EA1	1 1 1 1	0 1 1 0	MOD	1 1 0	R/M		
			[DISP-L]		[DISP-H]				
			1 1 1 1	0 1 1 1	MOD	1 1 0	R/M		
			[DISP-L]		[DISP-H]				
		[IDIV EA1/r1]	1 1 1 1	0 1 1 W	MOD	1 1 1	R/M		F6~F7
			[DISP-L]		[DISP-H]				
		IDIV r1	1 1 1 1	0 1 1 0	1 1	1 1 1	R/M		
			1 1 1 1	0 1 1 1	1 1	1 1 1	R/M		
		IDIV EA1	1 1 1 1	0 1 1 0	MOD	1 1 1	R/M		
			[DISP-L]		[DISP-H]				
			1 1 1 1	0 1 1 1	MOD	1 1 1	R/M		
			[DISP-L]		[DISP-H]				
		AAD	1 1 0 1	0 1 0 1	0 0	0 0 1	0 1 0		D5
		CBW	1 0 0 1	1 0 0 0					98
		CWD	1 0 0 1	1 0 0 1					99

MITSUBISHI ELECTRIC

Clock cycles	Bytes in the code	Bus cycles	Function description	Flags OF	DF	IF	TF	SF	ZF	AF	PF	CF
			(EXT) ← overflow digit of operation; when (EXT) = 0: (CF) ← 0	○	X	X	X	△	△	△	△	○
			When (EXT) ≠ 0: (0F) ← (CF), (CF) ← 1 − (CF)									
70 ~ 77	2	0	When W = 0: (EXT) = (AH)									
			(AX) ← (AL) * (r1)									
118 ~ 133	2	0	When W = 1: (EXT) = (DX)									
			(DX : AX) ← (AX) * (r1)									
(76 ~ 83) + EA	2 ~ 4	1	When W = 0: (EXT) = (AH) MOD ≠ 11									
			(AX) ← (AL) * (EA1)									
(124 ~ 139) + EA	2 ~ 4	1	When W = 1: (EXT) = (DX) MOD ≠ 11									
			(DX : AX) ← (AX) * (EA1)									
			(EXT) ← overflow digit of operation; when (LOW) changes to (EXT) by extending the sign bit of (LOW): (CF) ← 0	○	X	X	X	△	△	△	△	○
			Otherwise: (0F) ← (CF), (CF) ← 1									
80 ~ 98	2	0	When W = 0: (EXT) = (AH), (LOW) = (AL)									
			(AX) ← (AL) * (r1)									
128 ~ 154	2	0	When W = 1: (EXT) = (DX), (LOW) = (AX)									
			(DX : AX) ← (AX) * (r1)									
(86 ~ 104) + EA	2 ~ 4	1	When W = 0: (EXT) = (AH), (LOW) = (AL) MOD ≠ 11									
			(AX) ← (AL) * (EA1)									
(134 ~ 160) + EA	2 ~ 4	1	When W = 1: (EXT) = (DX), (LOW) = (AX) MOD ≠ 11									
			(DX : AX) ← (AX) * (EA1)									
83	2	0	(AH) ← (AL) ÷ $0A_{16}$,	△	X	X	X	○	○	△	○	△
			(AL) ← remainder									
			(temp) ← dividend; when W = 0: MAX = FF_{16};	△	X	X	X	△	△	△	△	△
			When W = 1: MAX = $FFFF_{16}$; (temp) ÷ (EA1/r1)									
			When results of the division are larger than MAX, an interrupt of TYPE = 0									
			is generated, (SP) ← (SP) − 2, ((SP) + 1: (SP)) ← flag									
			(IF) ← 0, (TF) ← 0, (SP) ← (SP) −2, ((SP) + 1: (SP)) ← (CS)									
			(CS) ← contents of address 2, (SP) ← (SP) −2, ((SP) +1: (SP)) ← IP									
			(IP) ← contents of address 0, the result of the division is undefined.									
80 ~ 90	2	0	(AL) ← (AX) ÷ (r1), (AH) ← Remainder									
144 ~ 162	2	0	(AX) ← (DX : AX) ÷ (r1), (DX) ← Remainder									
(86 ~ 96) + EA	2 ~ 4	1	(AL) ← (AX) ÷ (EA1) (AH) ← Remainder MOD ≠ 11									
(150 ~ 168) + EA	2 ~ 4	1	(AX) ← (DX : AX) ÷ (EA1), (DX) ← Remainder MOD ≠ 11									
			(temp) ← dividend; when W = 0: MAX = $7F_{16}$	△	X	X	X	△	△	△	△	△
			When W = 1: MAX = $7FFF_{16}$; MIN = 81_{16}									
			when the result of the division is positive and over MAX or when negative									
			and more negative than MIN an interrupt of TYPE = 0 is generated and									
			the result of the division is undefined.									
101 ~ 112	2	0	(AL) ← (AX) ÷ (r1), (AH) ← Remainder									
165 ~ 184	2	0	(AX) ← (DX : AX) ÷ (r1), (DX) ← Remainder									
(107 ~ 118) + EA	2 ~ 4	1	(AL) ← (AX) ÷ (EA1), (AH) ← Remainder MOD ≠ 11									
(171 ~ 190) + EA	2 ~ 4	1	(AX) ← (DX : AX) ÷ (EA1), (DX) ← Remainder MOD ≠ 11									
60	2	0	(AL) ← (AH) * $0A_{16}$ + (AL), (AH) ← 0	△	X	X	X	○	○	△	○	△
2	1	0	When (AL) < 80_{16}: (AH) ← 0	X	X	X	X	X	X	X	X	X
			When (AL) ≥ 80_{16}: (AH) ← FF_{16} (extended sign bit)									
5	1	0	When (AX) < 8000_{16}: (DX) ← 0	X	X	X	X	X	X	X	X	X
			When (AX) ≥ 8000_{16}: (DX) ← $FFFF_{16}$ (extended sign bit)									

8

16-BIT PARALLEL MICROPROCESSOR

Type of instruction		Mnemonic	Instruction code																Hexadecimal notation
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
Bit manipulation	Logic	[NOT EA1/r1]	1	1	1	1	0	1	1	W	MOD		0	1	0	R/M			F6~F7
		[DISP-L								]	[DISP-H							]	
		NOT r1	1	1	1	1	0	1	1	W	1	1	0	1	0	R/M			
		NOT EA1	1	1	1	1	0	1	1	W	MOD		0	1	0	R/M			
		[DISP-L								]	[DISP-H							]	
		[AND EA1/r1, EA2/r2]	0	0	1	0	0	0	d	W	MOD		REG			R/M			20~23
		[DISP-L								]	[DISP-H							]	
		AND r1, r2	0	0	1	0	0	0	d	W	1	1	REG			R/M			
		AND r1, EA2	0	0	1	0	0	0	1	W	MOD		REG			R/M			
		[DISP-L								]	[DISP-H							]	
		AND EA1, r2	0	0	1	D	0	0	d	W	MOD		REG			R/M			
		[DISP-L								]	[DISP-H							]	
		[AND EA1/r1, DATA]	1	0	0	0	0	0	0	W	MOD		1	0	0	R/M			80~81
		[DISP-L								]	[DISP-H							]	
		[DATA-L								]	[DATA-H							]	
		AND r1, DATA	1	0	0	0	0	0	0	W	1	1	1	0	0	R/M			
		[DATA-L								]	[DATA-H							]	
		AND EA1, DATA	1	0	0	0	0	0	0	W	MOD		1	0	0	R/M			
		[DISP-L								]	[DISP-H							]	
		[DATA-L								]	[DATA-H							]	
		AND Acc, DATA	0	0	1	0	0	1	0	W	[DATA-L							]	24~25
		[DATA-H								]									
		[TEST EA1/r1, EA2/r2]	1	0	0	0	0	1	0	W	MOD		REG			R/M			84~85
		[DISP-L								]	[DISP-H							]	
		TEST r1, r2	1	0	0	0	0	1	0	W	1	1	REG			R/M			
		TEST r1, EA2 / TEST EA1, r2	1	0	0	0	0	1	0	W	MOD		REG			R/M			
		[DISP-L								]	[DISP-H							]	
		[TEST EA1/r1, DATA]	1	1	1	1	0	1	1	W	MOD		0	0	0	R/M			F6~F7
		[DISP-L								]	[DISP-H							]	
		[DATA-L								]	[DATA-H							]	
		TEST r1, DATA	1	1	1	1	0	1	1	W	1	1	0	0	0	R/M			
		[DATA-L								]	[DATA-H							]	
		TEST EA1, DATA	1	1	1	1	0	1	1	W	MOD		0	0	0	R/M			
		[DISP-L								]	[DISP-H							]	
		[DATA-L								]	[DATA-H							]	
		TEST Acc, DATA	1	0	1	0	1	0	0	W	[DATA-L							]	A8~A9
		[DATA-H								]									
		[OR EA1/r1, EA2/r2]	0	0	0	0	1	0	d	W	MOD		REG			R/M			08~0B
		[DISP-L								]	[DISP-H							]	
		OR r1, r2	0	0	0	0	1	0	d	W	1	1	REG			R/M			
		OR r1, EA2	0	0	0	0	1	0	1	W	MOD		REG			R/M			
		[DISP-L								]	[DISP-H							]	
		OR EA1, r2	0	0	0	0	1	0	0	W	MOD		REG			R/M			
		[DISP-L								]	[DISP-H							]	
		[OR EA1/r1, DATA]	1	0	0	0	0	0	0	W	MOD		0	0	1	R/M			80~81
		[DISP-L								]	[DISP-H							]	
		[DATA-L								]	[DATA-H							]	
		OR r1, DATA	1	0	0	0	0	0	0	W	1	1	0	0	1	R/M			
		[DATA-L								]	[DATA-H							]	
		OR EA1, DATA	1	0	0	0	0	0	0	W	MOD		0	0	1	R/M			
		[DISP-L								]	[DISP-H							]	
		[DATA-L								]	[DATA-H							]	
		OR Acc, DATA	0	0	0	0	1	1	0	W	[DATA-L							]	0C~0D
		[DATA-H								]									
		[XOR EA1/r1, EA2/r2]	0	0	1	1	0	0	d	W	MOD		REG			R/M			30~33
		[DISP-L								]	[DISP-H							]	
		XOR r1, r2	0	0	1	1	0	0	d	W	1	1	REG			R/M			
		XOR r1, EA2	0	0	1	1	0	0	1	W	MOD		REG			R/M			
		[DISP-L								]	[DISP-H							]	
		XOR EA1, r2	0	0	1	1	0	0	0	W	MOD		REG			R/M			
		[DISP-L								]	[DISP-H							]	
		[XOR EA1/r1, DATA]	1	0	0	0	0	0	0	W	MOD		1	1	0	R/M			80~81
		[DISP-L								]	[DISP-H							]	
		[DATA-L								]	[DATA-H							]	
		XOR r1, DATA	1	0	0	0	0	0	0	W	1	1	1	1	0	R/M			
		[DATA-L								]	[DATA-H							]	
		XOR EA1, DATA	1	0	0	0	0	0	0	W	MOD		1	1	0	R/M			
		[DISP-L								]	[DISP-H							]	
		[DATA-L								]	[DATA-H							]	
		XOR Acc, DATA	0	0	1	1	0	1	0	W	[DATA-L							]	34~35
		[DATA-H								]									

Clock cycles	Bytes in the code	Bus cycles	Function description	Flags OF	DF	IF	TF	SF	ZF	AF	PF	CF
			When W = 0: (SRC) = FF_{16} When W = 1: (SRC) = $FFFF_{16}$	X	X	X	X	X	X	X	X	X
3	2	0	(r1) ← (SRC) − (r1)									
16+EA	2~4	1	(EA1) ← (SRC) − (EA1) MOD≠11									
			After execution of the instruction (CF) ← 0, (OF) ← 0	0	X	X	X	○	○	△	○	0
3	2	0	(r1) ← (r1) ∧ (r2)									
9+EA	2~4	1	(r1) ← (r1) ∧ (EA2) MOD≠11									
16+EA	2~4	2	(EA1) ← (EA1) ∧ (r2) MOD≠11									
			After execution of the instruction (CF) ← 0, (OF) ← 0	0	X	X	X	○	○	△	○	0
4	3~4	0	(r1) ← (r1) ∧ DATA									
17+EA	3~6	2	(EA1) ← (EA1) ∧ DATA MOD≠11									
4	2~3	0	(Acc) ← (Acc) ∧ DATA After execution of the instruction (CF) ← 0, (OF) ← 0	0	X	X	X	○	○	△	○	0
			After execution of the instruction (CF) ← 0, (OF) ← 0	0	X	X	X	○	○	△	○	0
3	2	0	(r1) ← (r1) ∧ (r2)									
9+EA	2~4	1	(r1) ← (r1) ∧ (EA2) (EA1) ← (EA1) ∧ (r2) MOD≠11									
			After execution of the instruction (CF) ← 0, (OF) ← 0	0	X	X	X	○	○	△	○	0
5	3~4	0	(r1) ← (r1) ∧ DATA									
11+EA	3~6	2	(EA1) ← (EA1) ∧ DATA MOD≠11									
4	2~3	0	(Acc) ∧ DATA, (CF) ← 0, (OF) ← 0	0	X	X	X	○	○	△	○	0
			After execution of the instruction (CF) ← 0, (OF) ← 0	0	X	X	X	○	○	△	○	0
3	2	0	(r1) ← (r1) ∨ (r2)									
9+EA	2~4	1	(r1) ← (r1) ∨ (EA2) MOD≠11									
16+EA	2~4	2	(EA1) ← (EA1) ∨ (r2) MOD≠11									
			After execution of the instruction (CF) ← 0, (OF) ← 0	0	X	X	X	○	○	△	○	0
4	3~4	0	(r1) ← (r1) ∨ DATA									
17+EA	3~6	2	(EA1) ← (EA1) ∨ DATA MOD≠11									
4	2~3	0	(Acc) ← (Acc) ∨ DATA, (CF) ← 0, (OF) ← 0	0	X	X	X	○	○	△	○	0
			After execution of the instruction (CF) ← 0, (OF) ← 0	0	X	X	X	○	○	△	○	0
3	2	0	(r1) ← (r1) ∀ (r2)									
9+EA	2~4	1	(r1) ← (r1) ∀ (EA2) MOD≠11									
16+EA	2~4	2	(EA1) ← (EA1) ∀ (r2) MOD≠11									
			After execution of the instruction (CF) ← 0, (OF) ← 0	0	X	X	X	○	○	△	○	0
4	3~4	0	(r1) ← (r1) ∀ DATA									
17+EA	3~6	2	(EA1) ← (EA1) ∀ DATA MOD≠11									
4	2~3	0	(Acc) ← (Acc) ∀ DATA, (CF) ← 0, (OF) ← 0	0	X	X	X	○	○	△	○	0

8

16-BIT PARALLEL MICROPROCESSOR

MACHINE INSTRUCTION INSTRUCTION SET SUMMARY

Item / Type of instruction		Mnemonic	Instruction code D_7 D_6 D_5 D_4	D_3 D_2 D_1 D_0	D_7 D_6	D_5 D_4 D_3	D_2 D_1 D_0	Hexadecimal notation
Bit manipulation (cont'd)	Shifts	{SHL/SAL EA1/r1, 1/CL}	1 1 0 1	0 0 V W	MOD	1 0 0	R/M	D0~D3
			{DISP-L}		{DISP-H}			
		SHL/SAL r1, 1	1 1 0 1	0 0 0 W	1 1	1 0 0	R/M	
		SHL/SAL r1, CL	1 1 0 1	0 0 1 W	1 1	1 0 0	R/M	
		SHL/SAL EA1, 1	1 1 0 1	0 0 0 W	MOD	1 0 0	R/M	
			{DISP-L}		{DISP-H}			
		SHL/SAL EA1, CL	1 1 0 1	0 0 1 W	MOD	1 0 0	R/M	
			{DISP-L}		{DISP-H}			
		{SHR EA1/r1, 1/CL}	1 1 0 1	0 0 V W	MOD	1 0 0	R/M	D0~D3
			{DISP-L}		{DISP-H}			
		SHR r1, 1	1 1 0 1	0 0 0 W	1 1	1 0 1	R/M	
		SHR r1, CL	1 1 0 1	0 0 1 W	1 1	1 0 1	R/M	
		SHR EA1, 1	1 1 0 1	0 0 0 W	MOD	1 0 1	R/M	
			{DISP-L}		{DISP-H}			
		SHR EA1, CL	1 1 0 1	0 0 1 W	MOD	1 0 1	R/M	
			{DISP-L}		{DISP-H}			
		{SAR EA1/r1, 1/CL}	1 1 0 1	0 0 V W	MOD	1 1 1	R/M	D0~D3
			{DISP-L}		{DISP-H}			
		SAR r1, 1	1 1 0 1	0 0 0 W	1 1	1 1 1	R/M	
		SAR r1, CL	1 1 0 1	0 0 1 W	1 1	1 1 1	R/M	
		SAR EA1, 1	1 1 0 1	0 0 0 W	MOD	1 1 1	R/M	
			{DISP-L}		{DISP-H}			
		SAR EA1, CL	1 1 0 1	0 0 1 W	MOD	1 1 1	R/M	
			{DISP-L}		{DISP-H}			
	Rotates	{ROL EA1/r1, 1/CL}	1 1 0 1	0 0 V W	MOD	0 0 0	R/M	D0~D3
			{DISP-L}		{DISP-H}			
		ROL r1, 1	1 1 0 1	0 0 0 W	1 1	0 0 0	R/M	
		ROL r1, CL	1 1 0 1	0 0 1 W	1 1	0 0 0	R/M	
		ROL EA1, 1	1 1 0 1	0 0 0 W	MOD	0 0 0	R/M	
			{DISP-L}		{DISP-H}			
		ROL EA1, CL	1 1 0 1	0 0 1 W	MOD	0 0 0	R/M	
			{DISP-L}		{DISP-H}			
		{ROR EA1/r1, 1/CL}	1 1 0 1	0 0 V W	MOD	0 0 1	R/M	D0~D3
			{DISP-L}		{DISP-H}			
		ROR r1, 1	1 1 0 1	0 0 0 W	1 1	0 0 1	R/M	
		ROR r1, CL	1 1 0 1	0 0 1 W	1 1	0 0 1	R/M	
		ROR EA1, 1	1 1 0 1	0 0 0 W	MOD	0 0 1	R/M	
			{DISP-L}		{DISP-H}			
		ROR EA1, CL	1 1 0 1	0 0 1 W	MOD	0 0 1	R/M	
			{DISP-L}		{DISP-H}			
		{RCL EA1/r1, 1/CL}	1 1 0 1	0 0 V W	MOD	0 1 0	R/M	D0~D3
			{DISP-L}		{DISP-H}			
		RCL r1, 1	1 1 0 1	0 0 0 W	1 1	0 1 0	R/M	
		RCL r1, CL	1 1 0 1	0 0 1 W	1 1	0 1 0	R/M	
		RCL EA1, 1	1 1 0 1	0 0 0 W	MOD	0 1 0	R/M	
			{DISP-L}		{DISP-H}			
		RCL EA1, CL	1 1 0 1	0 0 1 W	MOD	0 1 0	R/M	
			{DISP-L}		{DISP-H}			
		{RCR EA1/r1, 1/CL}	1 1 0 1	0 0 V W	MOD	0 1 1	R/M	D0~D3
			{DISP-L}		{DISP-H}			
		RCR r1, 1	1 1 0 1	0 0 0 W	1 1	0 1 1	R/M	
		RCR r1, CL	1 1 0 1	0 0 1 W	1 1	0 1 1	R/M	
		RCR EA1	1 1 0 1	0 0 0 W	MOD	0 1 1	R/M	
			{DISP-L}		{DISP-H}			
		RCR EA1, CL	1 1 0 1	0 0 1 W	MOD	0 1 1	R/M	
			{DISP-L}		{DISP-H}			

MITSUBISHI ELECTRIC

Clock cycles	Bytes in the code	Bus cycles	Function description	Flags OF	DF	IF	TF	SF	ZF	AF	PF	CF
2 8+4/bit 15+EA 20+EA+4/bit	2 2 2~4 2~4	0 0 2 2	When V = 0: COUNT ← 1 if the high-order bit of (EA1/r1) = (CF) : (OF) ← 0 if the high-order bit of (EA1/r1) ≠ (CF) : (OF) ← 1 When V = 1: COUNT ← (CL), (OF) is undefined Shift one bit as indicated below and reduce COUNT by 1, Repeat until COUNT becomes 0. CF ← (EA1/r1) ← 0	○	X	X	X	X	X	X	X	○
2 8+4/bit 15+EA 20+EA+4/bit	2 2 2~4 2~4	0 0 2 2	When V = 0: COUNT ← 1 if the high-order bits of (EA1/r1) are equal : (OF) ← 0 if the high-order bits of (EA1/r1) are not equal: (OF) ← 1 When V = 1: COUNT ← (CL), (OF) is undefined Shift one bit as indicated below and reduce COUNT by 1, Repeat until COUNT becomes 0. 0 → (EA1/r1) → CF	○	X	X	X	X	X	X	X	○
2 8+4/bit 15+EA 20+EA+4/bit	2 2 2~4 2~4	0 0 2 2	When V = 0: COUNT ← 1, (OF) ← 0 When V = 1: COUNT ← (CL), (OF) is undefined Shift one bit as indicated below and reduce COUNT by 1, Repeat until COUNT becomes 0. (EA1/r1) → CF	○	X	X	X	X	X	X	X	○
2 8+4/bit 15+EA 20+EA+4/bit	2 2 2~4 2~4	0 0 2 2	When V = 0: COUNT ← 1 if the high-order bit of (EA1/r1) = (CF) : (OF) ← 0 if the high-order bit of (EA1/r1) ≠ (CF) : (OF) ← 1 When V = 1: COUNT ← (CL), (OF) is undefined Rotate one bit as indicated below and reduce COUNT by 1. Repeat until COUNT becomes 0. CF ← (EA1/r1)	○	X	X	X	X	X	X	X	○
2 8+4/bit 15+EA 20+EA+4/bit	2 2 2~4 2~4	0 0 2 2	When V = 0: COUNT ← 1 if the high-order bits of (EA1/r1) are equal: (OF) ← 0 if the high-order bits of (EA1/r1) are not equal: (OF) ← 1 When V = 1: COUNT ← (CL), (OF) is undefined Rotate one bit as indicated below and reduce COUNT by 1. Repeat until COUNT becomes 0. (EA1/r1) → CF	○	X	X	X	X	X	X	X	○
2 8+4/bit 15+EA 20+EA+4/bit	2 2 2~4 2~4	0 0 2 2	When V = 0: COUNT ← 1 if the high-order bits of (EA1/r1) are equal: (OF) ← 0 if the high-order bits of (EA1/r1) are not equal: (OF) ← 1 When V = 1: COUNT ← (CL), (OF) is undefined Rotate one bit as indicated below and reduce COUNT by 1. Repeat until COUNT becomes 0. CF ← (EA1/r1)	○	X	X	X	X	X	X	X	○
2 8+4/bit 15+EA 20+EA+4/bit	2 2 2~4 2~4	0 0 2 2	When V = 0: COUNT ← 1 if the high-order bits of (EA1/r1) are equal: (OF) ← 0 if the high-order bits of (EA1/r1) are not equal: (OF) ← 1 When V = 1: COUNT ← (CL), (OF) is undefined Rotate one bit as indicated below and reduce COUNT by 1. Repeat until COUNT becomes 0. (EA1/r1) → CF	○	X	X	X	X	X	X	X	○

8

16-BIT PARALLEL MICROPROCESSOR

Type of instruction \ Item		Mnemonic	Instruction code														Hexadecimal notation
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
String manipulations	Repeat prefix	[REP]	1	1	1	1	0	0	1	Z									F2 ~ F3
		REPE/REPZ	1	1	1	1	0	0	1	1									
		REPNE/REPNZ	1	1	1	1	0	0	1	0									
	Transmission	MOVS MEM1, MEM2	1	0	1	0	0	1	0	W									A4 ~ A5
	Comparison	CMPS MEM1, MEM2	1	0	1	0	0	1	1	W									A6 ~ A7
	Scan	SCAS MEM	1	0	1	0	1	1	1	W									AE ~ AF
	Load	LODS MEM	1	0	1	0	1	1	0	W									AC ~ AD
	Store	STOS MEM	1	0	1	0	1	0	1	W									AA ~ AB

Clock cycles (Note 29)	Bytes in the code	Bus cycles (Note 29)	Function description	Flags OF	DF	IF	TF	SF	ZF	AF	PF	CF
2 2	1 1	0 0	Register CX becomes a counter. The prefixed instruction is executed and CX is counted down by 1. The execution of the prefixed instruction and counting down of CX is repeated until CX becomes 0. The instructions SCAS and CMPS, which may alter some flags will not repeat when the value of Z ≠ (ZF)	X	X	X	X	X	X	X	X	X
18 9+17/LOOP	1	2 2/LOOP	(DI) = MEM1, (ST) = MEM2 When W = 0: ((DI)) ← ((SI)) if (DF) = 0: (DI) ← (DI)+2, (SI) ← (SI)+2 if (DF) = 1: (DI) ← (DI)−2, (SI) ← (SI)−2 When W = 1: ((DI+1:DI)) ← ((SI+1:SI)) if (DF) = 0: (DI) ← (DI)+2, (SI) ← (SI)+2 if (DF) = 1: (DI) ← (DI)−2, (SI) ← (SI)−2	X	X	X	X	X	X	X	X	X
22 9+22/LOOP	1	2 2/LOOP	(SI) = MEM1, (DI) = MEM2 When W = 0: ((SI)) ← ((DI)) if (DF) = 0: (DI) ← (DI)+1, (SI) ← (SI)+1 if (DF) = 1: (DI) ← (DI)−1, (SI) ← (SI)−1 When W = 1: ((SI+1:SI)) ← ((DI+1:DI)) if (DF) = 0: (DI) ← (DI)+2, (SI) ← (SI)+2 if (DF) = 1: (DI) ← (DI)−2, (SI) ← (SI)−2	○	X	X	X	○	○	○	○	○
15 9+15/LOOP	1	1 1/LOOP	(DI) = MEM When W = 0: (AL) ← ((DI)) if (DF) = 0: (DI) ← (DI)+1 if (DF) = 1: (DI) ← (DI)−1 When W = 1: (AX) ← ((DI+1:DI)) if (DF) = 0: (DI) ← (DI)+2 if (DF) = 1: (DI) ← (DI)−2	○	X	X	X	○	○	○	○	○
12 9+13/LOOP	1	1 1/LOOP	(SI) = MEM When W = 0: (AL) ← ((SI)) if (DF) = 0: (SI) ← (SI)+1 if (DF) = 1: (SI) ← (SI)−1 When W = 1: (AX) ← ((SI+1:SI)) if (DF) = 0: (SI) ← (SI)+2 if (DF) = 1: (SI) ← (SI)−2	X	X	X	X	X	X	X	X	X
11 9+10/LOOP	1	1 1/LOOP	(DI) = MEM When W = 0: ((DI)) ← (AL) if (DF) = 0: (DI) ← (DI)+1 if (DF) = 1: (DI) ← (DI)−1 When W = 1: ((DI+1:DI)) ← (AX) if (DF) = 0: (DI) ← (DI)+2 if (DF) = 1: (DI) ← (DI)−2	X	X	X	X	X	X	X	X	X

Note 29: The number of clock and bus cycles depend on the number of time an instruction is repeated.
The numbers shown are the number per loop and to determine the number of cycles the listed figures must be multiplied by the number of times the instruction is repeated.

8

Type of instruction \ Item		Mnemonic	Instruction code			Hexadecimal notation
			D_7 D_6 D_5 D_4	D_3 D_2 D_1 D_0	D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	
Control transfer	Unconditional jump	JMP DISP16	1 1 1 0 [DISP-H	1 0 0 1]	[DISP-L]	E9
		JMP DISP8	1 1 1 0	1 0 1 1	[DISP-L]	EB
		JMP EA1/r1	1 1 1 1 [DISP-L	1 1 1 1]	MOD 1 0 0 R/M [DISP-H]	FF
		JMP FAR-LABEL	1 1 1 0 [Offset-H [Seg-H	1 0 1 0]]	[Offset-L] [Seg-L]	EA
		JMP EA1/r1	1 1 1 1 [DISP-L	1 1 1 1]	MOD 1 0 1 R/M [DISP-H]	FF
	Call	CALL NEAR-PROC	1 1 1 0 [DISP-H	1 0 0 0]	[DISP-L]	E8
		CALL EA1/r1	1 1 1 1 [DISP-L	1 1 1 1]	MOD 0 1 0 R/M [DISP-H]	FF
		CALL FAR-PROC	1 0 0 1 [Offset-H [Seg-H	1 0 1 0]]	[Offset-L] [Seg-L]	9A
		CALL EA1/r1	1 1 1 1 [DISP-L	1 1 1 1]	MOD 0 1 1 R/M [DISP-H]	FF
	Return	RET	1 1 0 0	0 0 1 1		C3
		RET DATA	1 1 0 0 [DATA-H	0 0 1 0]	[DATA-L]	C2
		RET	1 1 0 0	1 0 1 1		CB
		RET DATA	1 1 0 0 [DATA-H	1 0 1 0]	[DATA-L]	CA

Clock cycles	Bytes in the code	Bus cycles	Function description	Flags OF	DF	IF	TF	SF	ZF	AF	PF	CF
15	3	0	Jump within current segment (DEST) ← (IP) + DISP	X	X	X	X	X	X	X	X	X
15	2	0	Jump within current segment	X	X	X	X	X	X	X	X	X
18 + EA	2~4	1 0	When MOD = 11: (IP) ← (r1) When MOD ≠ 11: (IP) ← (EA1)	X	X	X	X	X	X	X	X	X
15	5	0	Jump to other segment (IP) ← Offset, (CS) ← Seg	X	X	X	X	X	X	X	X	X
24 + EA	2~4	2	Jump to other segment (IP) ← (EA1/r1) (CS) ← (EA1 + 1/r1 + 1)	X	X	X	X	X	X	X	X	X
11	3	1	Call within current segment (DEST) ← (IP) + DISP	X	X	X	X	X	X	X	X	X
13 + EA	2~4	2	Call within current segment When MOD = 11: (DEST) ← r1 When MOD ≠ 11: (DEST) ← (EA1)	X	X	X	X	X	X	X	X	X
20	5	2	Call to other segment (DEST) ← Offset, (SEG) ← Seg.	X	X	X	X	X	X	X	X	X
29 + EA	2~4	4	Call to other segment (DEST) ← (EA1/r1), (SEG) ← (EA1 + 1/r1 + 1) (SP) ← (SP) − 2 ((SP) + 1 : (SP)) ← (CS) (CS) ← (SEG) (SP) ← (SP) − 2 ((SP) + 1 : (SP)) ← (IP) (IP) ← (DEST)	X	X	X	X	X	X	X	X	X
8	1	1	Return within current segment (IP) ← ((SP) + 1 : (SP)), (SP) ← (SP) + 2	X	X	X	X	X	X	X	X	X
12	3	1	Return within current segment (IP) ← ((SP) + 1 : (SP)), (SP) ← (SP) + DATA	X	X	X	X	X	X	X	X	X
18	1	2	Return to other segment (IP) ← ((SP) + 1 : (SP)), (SP) ← (SP) + 2 (CS) ← ((SP) + 1 : (SP)), (SP) ← (SP) + 2	X	X	X	X	X	X	X	X	X
17	3	2	Return to other segment (IP) ← ((SP) + 1 : (SP)), (SP) ← (SP) + 2 (CS) ← ((SP) + 1 : (SP)), (SP) ← (SP) + DATA	X	X	X	X	X	X	X	X	X

8

16-BIT PARALLEL MICROPROCESSOR

Item / Type of instruction		Mnemonic	Instruction code			Hexadecimal notation
			D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	
Control transfer (cont'd)	Condition jump	JE/JZ LABEL	0 1 1 1 0 1 0 0	[DISP]	74
		JL/JNGE LABEL	0 1 1 1 1 1 0 0	[DISP]	7C
		JLE/JNG LABEL	0 1 1 1 1 1 1 0	[DISP]	7E
		JB/JNAE LABEL	0 1 1 1 0 0 1 0	[DISP]	72
		JBE/JNA LABEL	0 1 1 1 0 1 1 0	[DISP]	76
		JP/JPE LABEL	0 1 1 1 1 0 1 0	[DISP]	7A
		JO LABEL	0 1 1 1 0 0 0 0	[DISP]	70
		JS LABEL	0 1 1 1 1 0 0 0	[DISP]	78
		JNE/JNZ LABEL	0 1 1 1 0 1 0 1	[DISP]	75
		JNL/JGE LABEL	0 1 1 1 1 1 0 1	[DISP]	7D
		JNLE/JG LABEL	0 1 1 1 1 1 1 1	[DISP]	7F
		JNB/JAE LABEL	0 1 1 1 0 0 1 1	[DISP]	73
		JNBE/JA LABEL	0 1 1 1 0 1 1 1	[DISP]	77
		JNP/JPO LABEL	0 1 1 1 1 0 1 1	[DISP]	7B
		JNO LABEL	0 1 1 1 0 0 0 1	[DISP]	71
		JNS LABEL	0 1 1 1 1 0 0 1	[DISP]	79
	Internal control	LOOP LABEL	1 1 1 0 0 0 1 0	[DISP]	E2
		LOOPZ/LOOPE LABEL	1 1 1 0 0 0 0 1	[DISP]	E1
		LOOPNZ/LOOPNE LABEL	1 1 1 0 0 0 0 0	[DISP]	E0
		JCXZ LABEL	1 1 1 0 0 0 1 1	[DISP]	E3

Clock cycles	Bytes in the code	Bus cycles	Function description	Flags OF	DF	IF	TF	SF	ZF	AF	PF	CF
16	2	0	When (ZF) = 1: (IP) ← (IP) + DISP (extends sign bit)	X	X	X	X	X	X	X	X	X
4	2	0	When (ZF) = 0: (IP) ← (IP) + 2 (executes the next inst.)									
16	2	0	When (SF) ∀ (OF) = 1: (IP) ← (IP) + DISP (extends sign bit)	X	X	X	X	X	X	X	X	X
4	2	0	When (SF) ∀ (OF) = 0: (IP) ← (IP) + 2 (executes the next inst.)									
16	2	0	When (SF) ∀ (OF) ∨ (ZF) = 1: (IP) ← (IP) + DISP (extends sign bit)	X	X	X	X	X	X	X	X	X
4	2	0	When (SF) ∀ (OF) ∨ (ZF) = 0: (IP) ← (IP) + 2 (executes the next inst.)									
16	2	0	When (CF) = 1: (IP) ← (IP) + DISP (extends sign bit)	X	X	X	X	X	X	X	X	X
4	2	0	When (CF) = 0: (IP) ← (IP) + 2 (executes the next inst.)									
16	2	0	When (CF) ∨ (ZF) = 1: (IP) ← (IP) + DISP (extends sign bit)	X	X	X	X	X	X	X	X	X
6	2	0	When (CF) ∨ (ZF) = 0: (IP) ← (IP) + 2 (executes the next inst.)									
16	2	0	When (PF) = 1: (IP) ← (IP) + DISP (extends sign bit)	X	X	X	X	X	X	X	X	X
4	2	0	When (PF) = 0: (IP) ← (IP) + 2 (executes the next inst.)									
16	2	0	When (OF) = 1: (IP) ← (IP) + DISP (extends sign bit)	X	X	X	X	X	X	X	X	X
4	2	0	When (OF) = 0: (IP) ← (IP) + 2 (executes the next inst.)									
16	2	0	When (SF) = 1: (IP) ← (IP) + DISP (extends sign bit)	X	X	X	X	X	X	X	X	X
4	2	0	When (SF) = 0: (IP) ← (IP) + 2 (executes the next inst.)									
16	2	0	When (ZF) = 0: (IP) ← (IP) + DISP (extends sign bit)	X	X	X	X	X	X	X	X	X
4	2	0	When (ZF) = 1: (IP) ← (IP) + 2 (executes the next inst.)									
16	2	0	When (SF) ∀ (OF) = 0: (IP) ← (IP) + DISP (extends sign bit)	X	X	X	X	X	X	X	X	X
4	2	0	When (SF) ∀ (OF) = 1: (IP) ← (IP) + 2 (executes the next inst.)									
16	2	0	When (SF) ∀ (OF) ∨ (ZF) = 0: (IP) ← (IP) + DISP (extends sign bit)	X	X	X	X	X	X	X	X	X
4	2	0	When (SF) ∀ (OF) ∨ (ZF) = 1: (IP) ← (IP) + 2 (executes the next inst.)									
16	2	0	When (CF) = 0: (IP) ← (IP) + DISP (extends sign bit)	X	X	X	X	X	X	X	X	X
4	2	0	When (CF) = 1: (IP) ← (IP) + 2 (executes the next inst.)									
16	2	0	When (CF) ∨ (ZF) = 0: (IP) ← (IP) + DISP (extends sign bit)	X	X	X	X	X	X	X	X	X
4	2	0	When (CF) ∨ (ZF) = 1: (IP) ← (IP) + 2 (executes the next inst.)									
16	2	0	When (PF) = 0: (IP) ← (IP) + DISP (extends sign bit)	X	X	X	X	X	X	X	X	X
4	2	0	When (PF) = 1: (IP) ← (IP) + 2 (executes the next inst.)									
16	2	0	When (OF) = 0: (IP) ← (IP) + DISP (extends sign bit)	X	X	X	X	X	X	X	X	X
4	2	0	When (OF) = 1: (IP) ← (IP) + 2 (executes the next inst.)									
16	2	0	When (SF) = 0: (IP) ← (IP) + DISP (extends sign bit)	X	X	X	X	X	X	X	X	X
4	2	0	When (SF) = 1: (IP) ← (IP) + 2 (executes the next inst.)									
17	2	0	(CX) ← (CX) −1	X	X	X	X	X	X	X	X	X
5	2	0	When (CX) = 0: (IP) ← (IP) + DISP (extends sign bit) When (CX) = 1: (IP) ← (IP) + 2 (executes the next inst.)									
18	2	0	(CX) ← (CX) −1	X	X	X	X	X	X	X	X	X
6	2	0	When (ZF) = 1 and (CX) ≠ 0: (IP) ← (IP) + DISP (extends sign bit) When (ZF) = 0 or (CX) = 0: (IP) ← (IP) + 2 (executes the next inst.)									
19	2	0	(CX) ← (CX) −1	X	X	X	X	X	X	X	X	X
5	2	0	When (ZF) = 0 and (CX) ≠ 0: (IP) ← (IP) + DISP (extends sign bit) When (ZF) = 1 or (CX) = 0: (IP) ← (IP) + 2 (executes the next inst.)									
18	2	0	When (CX) = 0: (IP) ← (IP) + DISP (extends sign bit)	X	X	X	X	X	X	X	X	X
6	2	0	When (CX) = 1: (IP) ← (IP) + 2 (executes the next inst.)									

8

16-BIT PARALLEL MICROPROCESSOR

Item / Type of instruction		Mnemonic	Instruction code																Hexadecimal notation
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
Control transfer (cont'd)	Interrupt	[INT type] INT type 3 INT (any)	1	1	0	0	1	1	0	V	[When V = 1: Specify type]								CC~CD
			1	1	0	0	1	1	0	0									
			1	1	0	0	1	1	0	1	[type]								
		INTO	1	1	0	0	1	1	1	0									CE
		IRET	1	1	0	0	1	1	1	1									CF
Processor control	Flag	CLC	1	1	1	1	1	0	0	0									F8
		CMC	1	1	1	1	0	1	0	1									F5
		STC	1	1	1	1	1	0	0	1									F9
		CLD	1	1	1	1	1	1	0	0									FC
		STD	1	1	1	1	1	1	0	1									FD
		CLI	1	1	1	1	1	0	1	0									FA
		STI	1	1	1	1	1	0	1	1									FB
	Miscellaneous	HLT	1	1	1	1	0	1	0	0									F4
		WAIT	1	0	0	1	1	0	1	1									9B
		ESC	1	1	0	1	1	X	X	X	MOD	X	X	X	R/M				D8~DF
			[DISP-L				]				[DISP-H				]				
		LOCK	1	1	1	1	0	0	0	0									F0
		NOP	1	0	0	1	0	0	0	0									90

Note 30: The preceding tables are summaries of details of the instructions of the M5L8086S. Basic instructions with variation are shown in brackets "[ ]" in the mnemonic column followed by the variations.
Instructions are from 1 to 6 bytes in length. Details of the first byte are given in the left half of the instruction code column, the second byte in the right half, the third byte below the first, the fourth byte below the second, the fifth byte below the third and the sixth byte below the fourth.
The hexadecimal column shows the value of the first byte of an instruction. When in has a single value the single value is shown. When it has a range of values, the range is shown.

Clock cycles	Bytes in the code	Bus cycles	Function description	Flags OF	DF	IF	TF	SF	ZF	AF	PF	CF
52 51	1 2	5 5	When V = 0: Type = 3 When V = 1: Type = type (0 ~ 255) (SP) ← (SP) − 2, ((SP) + 1 : (SP)) ← Flag (IF) ← 0, (TF) ← 0, (SP) ← (SP) − 2 ((SP) + 1 : (SP)) ← (CS), (CS) ← (type * 4 + 2) (SP) ← (SP) − 2, ((SP) + 1 : (SP)) ← (IP) (IP) ← (type * 4)	X	X	0	0	X	X	X	X	X
4 53	1 1	0 5	When (OF) = 0: No operation When (OF) = 1: (SP) ← (SP) − 2, ((SP) + 1 : (SP)) ← Flag (IF) ← 0, (TF) ← 0 (SP) ← (SP) − 2, ((SP) + 1 : (SP)) ← (CS) (CS) ← 12_{16} (SP) ← (SP) − 2, ((SP) + 1 : (IP) ← 10_{16}	X	X	0	0	X	X	X	X	X
24	1	3	Return from interrupt routine (IP) ← ((SP) + 1 : (SP)), (SP) ← (SP) + 2 (CS) ← ((SP) + 1 : (SP)), (SP) ← (SP) + 2 (Flag) ← ((SP) + 1 : (SP)), (SP) ← (SP) + 2	○	○	○	○	○	○	○	○	○
2	1	0	(CF) ← 0	X	X	X	X	X	X	X	X	0
2	1	0	When (CF) = 0: (CF) ← 1 (complement CF) When (CF) = 1: (CF) ← 0	X	X	X	X	X	X	X	X	○
2	1	0	(CF) ← 1	X	X	X	X	X	X	X	X	1
2	1	0	(DF) ← 0	X	1	X	X	X	X	X	X	X
2	1	0	(DF) ← 1	X	1	X	X	X	X	X	X	X
2	1	0	(IF) ← 0	X	X	0	X	X	X	X	X	X
2	1	0	(IF) ← 1	X	X	1	X	X	X	X	X	X
2	1	0	When this instruction is executed the CPU is put in the HALT state. An interrupt or RESET will take the CPU out of the HALT state.	X	X	X	X	X	X	X	X	X
3	1	0	The CPU is kept in the wait state until the TEST pin is V_{OL}. When	X	X	X	X	X	X	X	X	X
2 8+EA	2 2 ~ 4	0 1	AAA and BBB are not specified When MOD = 11: No operation When MOD ≠ 11: (DATA BUS) ← (EA1)	X	X	X	X	X	X	X	X	X
2	1	0	When this is prefixed to an instruction, during the execution of the instruction a bus lock is output through the LOCK pin.	X	X	X	X	X	X	X	X	X
3	1	0	NO OPERATION	X	X	X	X	X	X	X	X	X

8

16-BIT PARALLEL MICROPROCESSOR

SYMBOLS USED AND THEIR MEANING

Symbol		Symbol	
Acc	Accumulator (AX, AL or AH)	LABEL	Label name
ADDR	Memory address	V	Inclusive OR
DATA	Immediate data (data which is part of the instruction)	∀	Exclusive OR
DISP	Displacement	∧	Logical AND
d	When d = 0 source is REG, destination is EA (R/M)	−	Subtraction
	When d = 1 source is EA (R/M0, destination is REG	+	Addition
EA	Effective address	*	Multiplication
EA1, EA2	First effective address, second effective address	÷	Division
Port	I/O port	←	Direction of data transfer
r1, r2	First register, second register	↔	Exchange of data
SR	Segment register code	( )	Contents of register or memory
SEG	Segment register (CS, DS, SS and ES)	V1:V2	Pair of register or data treated as one unit
W	When 0: byte processing When 1: word processing	×	Does not affect the flag or instruction
−L	Suffix indicating low-order 8 bits	○	Flag may be changed by the instruction
−H	Suffix indicating high-order 8 bits	∧	Flag is undefined after execution of the instruction
DEST	Transfer address (Destination address)	FR	Flag register

ADDRESSING MODE AND REGISTER

1. Instruction Format

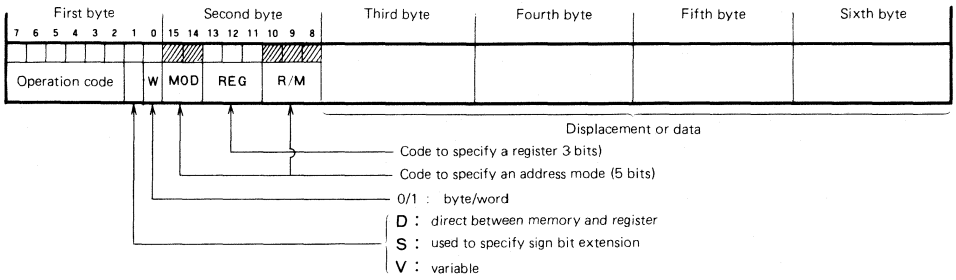


Table 1 EA Determination based on R/M and MOD

MOD / R/M	Memory mode			Register mode	
	Calculation for EA (effective address)			11	
	00	01	10	W=0	W=1
0 0 0	(BX)+(SI)	(BX)+(SI)+D8	(BX)+(SI)+D16	AL	AX
0 0 1	(BX)+(DI)	(BX)+(DI)+D8	(BX)+(DI)+D16	CL	CX
0 1 0	(BP)+(SI)	(BP)+(SI)+D8	(BP)+(SI)+D16	DL	DX
0 1 1	(BP)+(DI)	(BP)+(DI)+D8	(BP)+(DI)+D16	BL	BX
1 0 0	(SI)	(SI)+D8	(SI)+D16	AH	SP
1 0 1	(DI)	(DI)+D8	(DI)+D16	CH	BP
1 1 0	Direct address	(BP)+D8	(BP)+D16	DH	SI
1 1 1	(BX)	(BX)+D8	(BX)+D16	BH	DI

Note 31: D8: 8-bit displacement variable, D16: 16-bit displacement variable

Table 2 Register code

REG	W=0	W=1
0 0 0	AL	AX
0 0 1	CL	CX
0 1 0	DL	DX
0 1 1	BL	BX
1 0 0	AH	SP
1 0 1	CH	BP
1 1 0	DH	SI
1 1 1	BH	DI

Segment override prefix

0	0	1	SR	1	1	0

Segment register code

SR	SEG
0 0	ES
0 1	CS
1 0	SS
1 1	DS

MITSUBISHI ELECTRIC

2. Effective Address (EA) Calculation Time

EA configuration		Segment register used	Computing time
Displacement only	Direct address	DS	6 (clocks)
Base or index register	BP	SS	
	BX, SI, DI	DS	5
Displacement + base or index register	(BP + D8 or D16)	SS	
	(BX + D8 or D16, S1 + D8 or D16, D1 + D8 or D16)	DS	9
Base register + index register	BP + DI	SS	7
	BX + SI	DS	
	BP + SI	SS	8
	BX + DI	DS	
Displacement + base register + index register	BP + DI + D8 or D16	SS	11
	BX + SI + D8 or D16	DS	
	BP + SI + D8 or D16	SS	12
	BX + DI + D8 or D16	DS	

Note 32: When the segment override prefix is used the segment register used (column 2) is changed to the segment register specified by [0|0|1|SR|1|1|0] and 2 clock cycles must be added to the time (column 3) above.

3. Flag Register

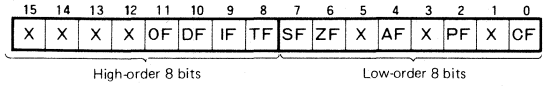


Table 3 Flag code and name

OF	overflow flag	When an arithmetic overflow occurs, when 2 operands are exclusive ORed up to the high-order bit and the high-order bit bit is 1, OF is set.
DF	direction flag	
IF	interrupt enable flag	
TF	trap flag	
SF	sign flag	When the high-order bit is 1, SF is set.
ZF	zero flag	When the result is zero, ZF is set.
AF	auxiliary flag	When there is a borrow from the low-order 4 bits, AF is set.
PF	parity flag	When the number of 1's in the low-order 8 bits is even, PF is set.
CF	carry flag	When a carry is generated from the high-order bit, CF is set.

OF, SF, ZF, AF, PF and CF are set/reset after the operation is completed.

8

16-BIT PARALLEL MICROPROCESSOR

8086 INSTRUCTION SET MATRIX

D_7–D_4 / Hexadecimal notation / D_3–D_0		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	ADD b,ear	ADC b,ear	AND b,ear	XOR b,ear	INC AX	PUSH AX	—	JO	See table below	NOP	MOV AL←m	MOV AL←i	—	See table below	LOOPNZ /LOOPNE	LOCK
0001	1	ADD w,ear	ADC w,ear	AND w,ear	XOR w,ear	INC CX	PUSH CX	—	JNO		XCHG CX	MOV AX←m	MOV CL←i	—		LOOPZ/ LOOPE	—
0010	2	ADD b,rea	ADC b,rea	AND b,rea	XOR b,rea	INC DX	PUSH DX	—	JB/ JNAE		XCHG DX	MOV AL←m	MOV DL←i	RET (i+SP)		LOOP	REP z=0
0011	3	ADD w,rea	ADC w,rea	AND w,rea	XOR w,rea	INC BX	PUSH BX	—	JNB/ JAE		XCHG BX	MOV AX←m	MOV BL←i	RET		JCXZ	REP z=1
0100	4	ADD b,ia	ADC b,i	AND b,i	XOR b,i	INC SP	PUSH SP	—	JE/ JZ	TEST b,ea	XCHG SP	MOVS b	MOV AH←i	LES	AAM	IN b	HLT
0101	5	ADD w,ia	ADC w,i	AND w,i	XOR w,i	INC BP	PUSH BP	—	JNE/ JNZ	TEST w,ea	XCHG BP	MOVS w	MOV CH←i	LDS	AAD	IN w	CMC
0110	6	PUSH ES	PUSH SS	SEG ES	SEG SS	INC SI	PUSH SI	—	JBE/ JNA	XCHG b,ea	XCHG SI	CMPS b	MOV DH←i	MOV b,ea,i	—	OUT b	See table below
0111	7	POP ES	POP SS	DAA	AAA	INC DI	PUSH DI	—	JNBE/ JA	XCHG w,ea	XCHG DI	CMPS w	MOV BH←i	MOV w,ea,i	XLAT	OUT w	
1000	8	OR b,ear	SBB b,ear	SUB b,ear	CMP b,ear	DEC AX	POP AX	—	JS	MOV b,ear	CBW	TEST b,i,a	MOV AX←i	—	ESC 0	CALL d	CLC
1001	9	OR w,ear	SBB w,ear	SUB w,ear	CMP w,ear	DEC CX	POP CX	—	JNS	MOV w,ear	CWD	TEST w,i,a	MOV CX←i	—	ESC 1	JMP d	STC
1010	A	OR b,rea	SBB b,rea	SUB b,rea	CMP b,rea	DEC DX	POP DX	—	JP/ JPE	MOV b,rea	CALL I,d	STOS b	MOV DX←i	RET I,(i+SP)	ESC 2	JMP I,d	CLI
1011	B	OR w,rea	SBB w,rea	SUB w,rea	CMP w,rea	DEC BX	POP BX	—	JNP/ JPO	MOV w,rea	WAIT	STOS w	MOV BX←i	RET I	ESC 3	JMP si,d	STI
1100	C	OR b,i	SBB b,i	SUB b,i	CMP b,i	DEC SP	POP SP	—	JL/ JNGE	MOV easr	PUSHF	LODS b	MOV SP←i	INT type 3	ESC 4	IN b,v=1	CLD
1101	D	OR w,i	SBB w,i	SUB w,i	CMP w,i	DEC BP	POP BP	—	JNL/ JGE	LEA	POPF	LODS w	MOV BP←i	INT (any)	ESC 5	IN w,v=1	STD
1110	E	PUSH CS	PUSH DS	SEG CS	SEG DS	DEC SI	POP SI	—	JLE/ JNG	MOV srea	SAHF	SCAS b	MOV SI←i	INTO	ESC 6	OUT b,v=1	See table below
1111	F	—	POP DS	DAS	AAS	DEC DI	POP DI	—	JNLE /JG	POP ea	LAHF	SCAS w	MOV DI←i	IRET	ESC 7	OUT w,v=1	

TABLE GROUP INSTRUCTION CODE LIST

mod□r/m	000	001	010	011	100	101	110	111
immed	ADD	OR	ADC	SBB	AND	SUB	XOR	CMP
Shift	ROL	ROR	RCL	RCR	SHL/ SAL	SHR	—	SAR
Grp 1	TEST	—	NOT	NEG	MUL	IMUL	DIV	IDIV
Grp 2	INC	DEC	CALL id	CALL I,id	JMP id	JMP I,id	PUSH	—

Note 33: Special symbols used only in the "Instruction set matrix" and the "Group Instruction Code List".

EA➡effective address (including register mode), REG⬅register

b : byte operation
w : Word operation
– : this code should not be used because the result and function are undefined
a : accumulator
d : direct address
ea : calculation of EA
ear : processing results of EA and REG are transferred to EA
easr : (EA) ← (SR)
i : immediate data
ia : immediate data and accumulator
id : indirect address
is : immediate data in sign extended form
I : segment is included in the jump
m : memory
rea : processing results of REG and EA are transferred to REG
si : sign of 8 byte displacement is extended
srea : (SR) ← (EA)
v : variable
z : z bit
← : shows direction of transfer.

Note 34: The length of instructions varies from 1 byte (8 bits) to 6 bytes. The "Instruction Set Matrix" is ordered by the hexadecimal value of the first byte of the instruction. The instruction and its operands (an instruction may have no operand) are listed in mnemonic or symbolic form. The group instructions (those instructions with different functions depending on bit D_5, D_4, D_3 in the second word of the instruction) are shown in the "Group Instruction Code List".

LSIs FOR PERIPHERAL CIRCUITS

9

8-BIT 8-CHANNEL A-D CONVERTER

DESCRIPTION

The M58990P A-D converter is used to convert analog signals to 8-bit digital values. The A-D converter is fabricated using silicon-gates and CMOS technology. The M58990P can selectively multiplex 8 channels of analog input.

FEATURES

- Single 5V power supply
- Conversion resolution of 8 bits
- Broad range of analog input voltages: 0V ~ V_{CC}
- Conversion time: 100μs
- Conversion by successive approximation
- Can be used online through the data bus of a microprocessor
- The I/O pins can be connected directly to TTL circuits
- Interchangeable with NS's ADC0808 (in pin configuration)

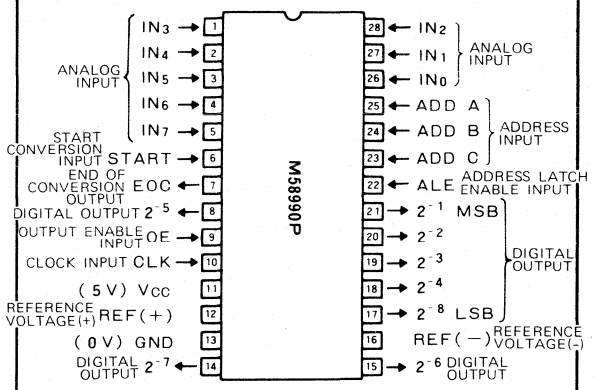
APPLICATION

- Used with microcomputers to control analog systems

FUNCTION

The M58990P has eight analog input terminals that are selected by the input signals to the 3 address terminals (ADD A ~ ADD C). The address signals of these terminals are read and latched in the internal address latches by the ALE signal. When the OE terminal is at low-level, the output terminals 2^{-1} ~ 2^{-8} are in a floating state so they can be connected directly to the data bus of a microcomputer.

PIN CONFIGURATION (TOP VIEW)

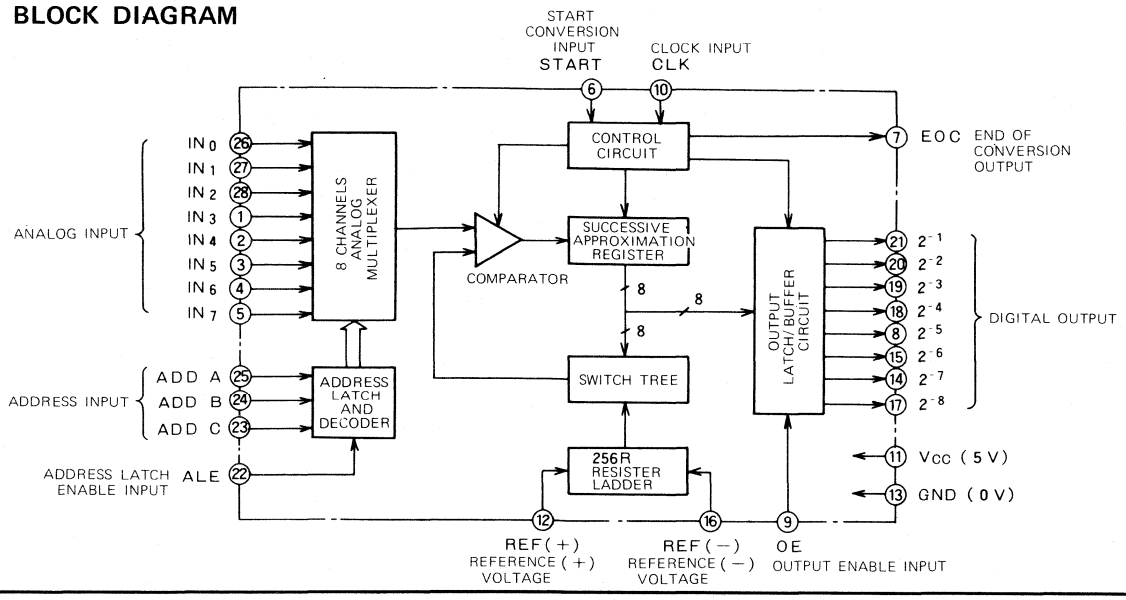


Outline 28P4

The input terminal START is used to call for the start of an analog to digital conversion and a signal is output through terminal EOC when the conversion is completed.

9

BLOCK DIAGRAM



8-BIT 8-CHANNEL A-D CONVERTER

PIN DESCRIPTIONS

Pin	Name	Input or Output	Functions
IN_0 ~ IN_7	Analog signal	Input	These are analog signal input pins. Which of the 8 inputs is selected, is determined by ADD A ~ ADD C. An analog voltage applied at the selected pin is converted to a digital value in the range of $2^{-1} \sim 2^{-8}$ and output.
ADD A ~ ADD C	Address signal	Input	The input is used for selecting which of the 8 terminals $IN_0 \sim IN_7$ is to be converted from analog to digital. The address input through ADD A ~ ADD C is read to the address latch by the riseing edge of ALE.
ALE	Address latch enable signal	Input	This is the strobe signal which causes the address signal input through ADD A ~ ADD C to be read and latched for use as an internal address.
REF(+)	Reference voltage (+)	Input	This is one of the input terminals for the reference voltage that is applied to the 256R resistor ladder circuit. The other terminal is REF (–) and the voltage levels of these two inputs must meet the condition: REF (+) > REF (–).
REF(–)	Reference voltage (–)	Input	This is one of the input terminals for the reference voltage that is applied to the 256R resistor ladder circuit. The other terminal is REF (+) and the voltage levels of these two inputs must meet the condition: REF (+) > REF (–).
OE	Output enable signal	Input	The signal at this pin controls the digital output. When the signal is low-level, pins $2^{-1} \sim 2^{-8}$ are in a floating state. When it is high-level, the data is output.
2^{-1} ~ 2^{-8}	Digital signal	Output	The analog signal, which was input through $IN_0 \sim IN_7$, is converted to digital data and is output from these terminals. When OE is low-level, these terminals are floating. When OE is high-level, the converted digital data is output. The MSB is 2^{-1} and the LSB is 2^{-8}
EOC	End of conversion signal	Output	This terminals is used to indicate the completion of an analog to digital conversion. It is reset by a START signal (high-level to low-level) and is set on completion of the conversion (low-level to high-level). This output is normally used to generate an interrupt request for the CPU.
START	Start conversion signal	Input	The input signal at this terminal is used to start a conversion cycle by setting the successive approximation register. The successive approximation register is reset by rising from low-level to high-level and conversion is started after being set by falling from high-level to low-level.
CLK	Clock input	Input	The signal at this terminal is the basic clocking signal used to determine internal timing.

M5C6847P-1

VIDEO DISPLAY GENERATOR

DESCRIPTION

The M5C6847P-1 is a color or monochrome television interface device, frabricated using N-channel silicon gate ED-MOS technology. The M5C6847P-1 has a 64-character (6-bit ASCII code) generator and memory interface.

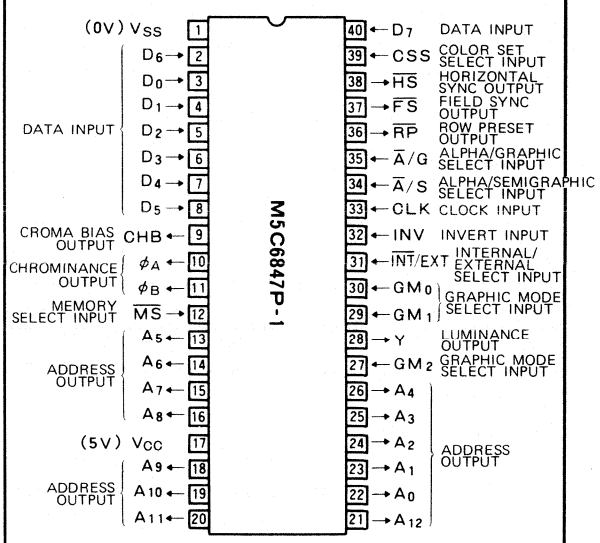
FEATURES

- Can be easily connected to the MELPS 85 series 8-bit CPUs.
- Alphanumeric display: 4 modes
- Graphic display: 8 modes
- Can connect directly with the M51342P RF modulator
- Alphanumeric display: 32 characters per line by 16 lines
- Character generator for 64 ASCII characters
- Can be used with an external character generator
- Generates composite video signals
- Generates intensity signal Y, color signal R-Y (ϕA) and B-Y (ϕB)
- Display RAM capacity (depends on mode): 512~6K bytes
- Single 5V power supply
- Interchangeable with the Motorola's MC6847P in pin configuration

APPLICATION

- Microcomputer system or terminals using a color or monochrome CRT.

PIN CONFIGURATION (TOP VIEW)



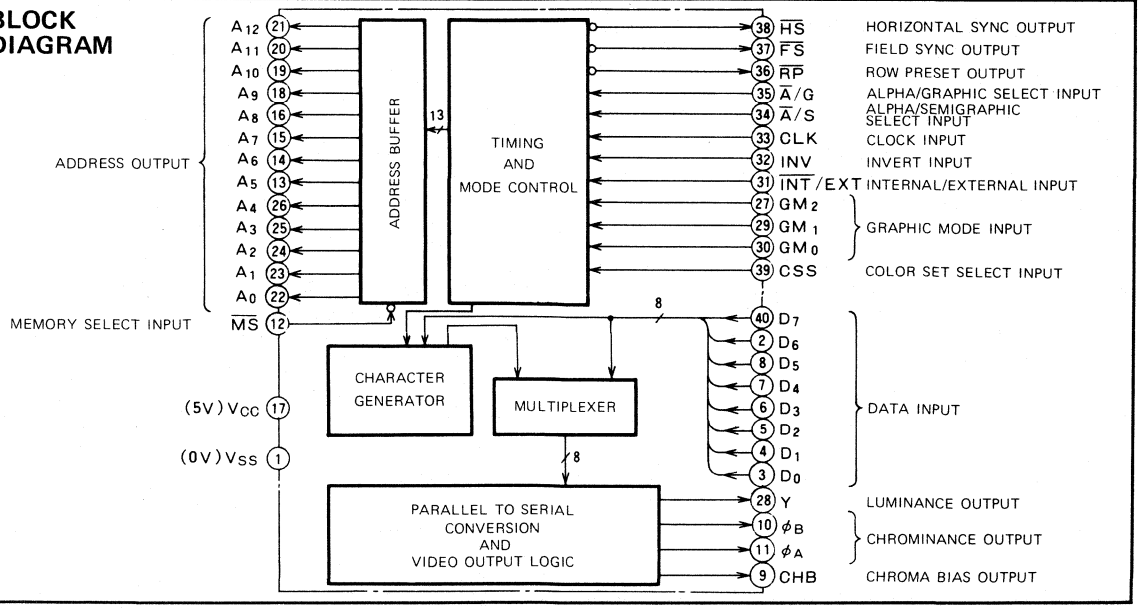
Outline 40P1

FUNCTION

The picture on the television set is composed of the synchronization signals such as horizontal synchronization signal, vertical synchronization signal and color burst signal, and syncronizing serial data. M5C6847P-1 can generate these signals. The information or data to be shown on the screen is written in the display memory by the CPU. (When the picture is to be composed on a CRT) the data for one screen in the display memory is read in the order of the scan cycles and synchronization signals are added. This

9

BLOCK DIAGRAM



serial is sent to the RF modulator. The M5C6847P-1 performs these functions by reading the display memory in the order of the CRT scan, adding the required synchronization signals such as luminance signal, color signal and then transferring the data stream serially to the RF modulator.

OPERATION

Address Outputs (A_{12}~A_0)

Thirteen address lines are used by the M5C6847P-1 to access the display memory (refresh memory). The starting address of the display memory is located at the upper-left corner of the display screen. As the television sweeps from the left to right and top to bottom, the VDG increments the RAM display address. The address lines are TTL-compatible and may be forced in a high-impedance state when input $\overline{MS}$ goes low.

Data Input (D_7~D_0)

Eight TTL-compatible data lines are used to input data from the display memory to be processed by the M5C-6847P-1. The data is interpreted and transformed into video analog level signals.

Video Output (Y, ϕ_A, ϕ_B, CHB)

These video outputs are used to transfer luminance and color information of pictures displayed on television with standard NTSC systems. These outputs can be directly connected to the RF modulator M51342P.

Luminance Output (Y)

The luminance output is a 6-level analog output. The six level analog outputs contain composite, blank, and four levels of video intensity.

Chrominance Output (ϕ_A)

The chrominance output ϕ_A is a 3-level analog output. The signal is used in combination with ϕ_B and Y to specify one of eight colors.

Chrominance Output (ϕ_B)

The chrominance output ϕ_B is a 4-level analog output. These levels of the signal are used in combination with ϕ_A and Y to specify one of eight colors. The other level is used to specify the time of the color burst reference signal.

Chroma Bias Output (CHB)

The chroma bias output is a single level analog output that provides the DC reference for chrominance outputs.

Synchronization Input ($\overline{MS}$, CLK)

Memory Select Input ($\overline{MS}$)

This is a TTL compatible input. When it goes low-level, address outputs (A_{12}~A_0) are forced in high-impedance state. When other devices such as the CPU access the display memory, it must be kept at low-level to prevent interference.

Clock input (CLK)

The clock input requires a 3.579545 MHz clock with a duty cycle of 50±5%. The M51342P RF modulator may be used to supply the 3.579545 MHz clock.

Synchronization output ($\overline{FS}$, $\overline{HS}$, $\overline{RP}$)

The synchronization outputs $\overline{FS}$, $\overline{HS}$ and $\overline{RP}$ are TTL-compatible and provide circuits, exterior to the M5C6847P-1 states.

Table 1 Operation modes

$\overline{A}/G$	$\overline{A}/S$	$\overline{INT}/EXT$	INV	GM_2	GM_1	GM_0	Mode
0	0	0	0	X	X	X	Internal alphanumerics
0	0	0	1	X	X	X	Internal alphanumerics inverted
0	0	1	0	X	X	X	External alphanumerics
0	0	1	1	X	X	X	External alphanumerics inverted
0	1	0	X	X	X	X	Semigraphics 4
0	1	1	X	X	X	X	Semigraphics 6
1	X	X	X	0	0	0	64× 64 Color graphics
1	X	X	X	0	0	1	128× 64 Graphics
1	X	X	X	0	1	0	128× 64 Color graphics
1	X	X	X	0	1	1	128× 96 Graphics
1	X	X	X	1	0	0	128× 96 Color graphics
1	X	X	X	1	0	1	128×192 Graphics
1	X	X	X	1	1	0	128×192 Color graphics
1	X	X	X	1	1	1	256×192 Graphics

Note 1: X is "don't care" bit

Table 2 Alphanumeric mode display memory, color and display element

Mode	Memory capacity (bits)	Color	Display elements
Internal alphanumerics	512×8	2	8 dots, 12 dots, Character is 5 x 7 dots
External alphanumerics	512×8	2	8 dots, 12 dots
Semigraphics 4	512×8	8	Elements 64×32
Semigraphics 6	512×8	4	Elements 64×48

Table 3 Graphic mode display memory, color and display element

Mode	Memory capacity (bits)	Color	Display elements
64× 64 Color graphics	1K×8	4	64×64
128× 64 Graphics	1K×8	2	128×64
128× 64 Color graphics	2K×8	4	
128× 96 Graphics	2K×8	2	128×96
128× 96 Color graphics	3K×8	4	
128×192 Graphics	3K×8	2	128×192
128×192 Color graphics	6K×8	4	
256×192 Graphics	6K×8	2	256×192

Field synchronization output ($\overline{FS}$)

The high to low transition of the $\overline{FS}$ output coincides with the end of active display area. The low to high transition of $\overline{FS}$ coincides with the trailing edge of the vertical synchronization pulse. The CPU should not access display memory while $\overline{FS}$ is at low-level to avoid undesired flicker on the screen.

Horizontal synchronization output ($\overline{HS}$)

This signal is used for horizontal synchronization on the CRT. A fall from high-level to low-level indicates the leading edge of the horizontal synchronization signal.

Row preset output ($\overline{RP}$)

This signal can be used when an external character generator ROM that is used with the VDG. An external 4-bit binary counter must also be added to supply row selection.

The counter is clocked by the $\overline{HS}$ signal and cleared by the $\overline{RP}$ signal. See Table 4 ② for details.

Mode Control Inputs ($\overline{A}$/G, $\overline{A}$/S, $\overline{INT}$/EXT, GM_2, GM_1, GM_0, CSS and INV)

These eight TTL-compatible input signals are used to determine and control the operational modes of the M5C6847P-1. Outline and details of the operational modes are shown in Table 1~3.

Alphanumeric mode

A screen in the alphanumeric mode is composed of 32 characters x 16 lines. Each character occupies space equivalent to an 8 x 12 dot matrix. The internal character generator can generate 64 characters (6-bit ASCII). Each character is formed by a 5 x 7 dot matrix. The low-order 6 bits of the 8-bit data input are used to select 1 of 64 characters and the remaining 2 bits can be used to implement the CSS and INV signal inputs. Operation in this mode requires a display memory of a least 512 bytes.

Semigraphic 4 mode

A screen in the semigraphics 4 mode is composed of 64 x 32 display elements. A display element is a 4 x 6 dot matrix; that is to say, each 8 x 12 character dot matrix is split into 4 display elements, each display element being a 4 x 6 dot matrix. The low-order 4 bits of the 8-bit data input correspond to the 4 display elements of a character. Three data bits of the remaining 4 bits may be used to select one of eight colors for the entire character box. The extra bit is available to switch the operation mode. Operation in this mode requires a display memory of at least 512 bytes.

Semigraphics 6 mode

A screen in the semigraphics 6 mode is composed of 64 x 48 display elements. A display element is a 4 x 4 dot matrix; that is to say, each 8 x 12 character dot matrix is split into 6 display elements, each display element being a 4 x 4 dot matrix. The low-order 6 bits of the 8-bit data input to the 6 display elements of a character and the remaining 2 bits are used to determine color. Operation in this mode requires a display memory of at least 512 bytes.

Full Graphic Modes

There are 8 full graphic modes. The border color (green or white) is selected by the level of the CSS signal. The CSS pin selects one of two sets of four colors in the four color graphic modes.

Color Graphic Mode 64 x 64

A screen in the 64 x 64 color graphic mode is composed of 64 x 64 display elements. Each display element can be 1 of 4 colors. Operation in this mode requires a display memory of at least 1024 bytes.

Graphic mode 128 x 64

A screen in the 128 x 64 graphic mode is composed of 128 x 64 display elements. Each display element can be green or white depending on the level of the CSS signal. Operation in this mode requires a display memory of at least 1024 bytes.

Color graphic mode 128 x 64

A screen in the 128 x 64 color graphic mode is composed of 128 x 64 display elements. Each display element can be 1 of 4 colors. Operation in this mode requires a display memory of at least 2048 bytes.

Graphic mode 128 x 96

A screen in the 128 x 96 graphic mode is composed of 128 x 96 picture elements. Each display element can be geeen or white depending on the level of the CSS signal. Operation in this mode requires a display memory of at least 2048 bytes.

Color graphic mode 128 x 96

A screen in the 128 x 96 color graphic mode is composed of 128 x 96 display elements. Each display element can be 1 of 4 colors. Operation in this mode requires a display memory of at least 3072 bytes.

Graphic mode 128 x 192

A screen in the 128 x 192 graphic mode is composed of 128 x 192 display elements. Each display element can be green or white depending on the level of the CSS signal. Operation in this mode requires a display memory of at least 3072 bytes.

Color graphic mode 128 x 192

A screen in the 128 x 192 color graphic mode is composed of 128 x 192 display elements. Each picture element can be 1 of 4 colors. Operation in this mode requires a display memory of at least 6144 bytes.

Graphic mode 256 x 192

A screen in the 256 x 192 graphic mode is composed of 256 x 192 display elements. Each display element can be green or white depending on the level of the CSS signal. Operation in this mode requires a display memory of at least 6144 bytes.

Details of the 8 graphic modes are shown in Table 4 which gives more information in an easy to understand form.

9

Table 4 Operational characteristics in the various graphic modes

	Input Pin									Color			TV Screen (1 screen is composed of 256x192 dots)		Data Bus	Display Mode
	$\overline{MS}$	$\overline{A}$/G	$\overline{A}$/S	$\overline{INT}$/EXT	GM_2	GM_1	GM_0	CSS	INV	Character Color	Background	Border	Mode	Display Elements		
①	1	0	0	0	X	X	X	0	0	green	black	black	16 lines of 32 characters	5x7 Dots 1 Character	D_7 6 5 4 3 2 1 0 ASCII Code	Alphanumeric mode
									1	black	green					
								1	0	orange	black	black				
									1	black	orange					
②	1	0	0	1	X	X	X	0	0	green	black	black	16 lines of 32 characters	8x12 Dots 1 Character	External ROM Code	Alphanumeric mode
									1	black	green					
								1	0	orange	black	black				
									1	black	orange					
③	1	0	1	0	X	X	X	X	X	$D_{0\sim3}$ D_6 D_5 D_4 0 X X X black 1 0 0 0 green 1 0 0 1 yellow 1 0 1 0 blue 1 0 1 1 red 1 1 0 0 white 1 1 0 1 cyan 1 1 1 0 magenta 1 1 1 1 orange		black	64×32 display elements	4 Dots: D_3 D_2 / D_1 D_0 } 6 Dots All 4 picture elements of the character group are the same color. The color intensity is 0 (black) or 1 (full color).	Color Luminance	Semigraphics 4 mode
④	1	0	1	1	X	X	X	0	X	D_{0-5} D_7 D_6 0 X X black 1 0 0 green 1 0 1 yellow 1 1 0 blue 1 1 1 red		black	64×48 display elements	4 Dots: D_5 D_4 / D_3 D_2 / D_1 D_0 } 4 Dots All 6 picture elements of the character group are the same color. The color intensity is 0 (black) or 1 (full color).	Color Luminance	Semigraphics 6 mode
								1		0 X X black 1 0 0 white 1 0 1 cyan 1 1 0 magenta 1 1 1 orange						
⑤	1	1	X	X	0	0	0	0	X	D_7 D_6 (D_5, D_4, D_3, D_2, D_1, D_0) 0 0 green 0 1 yellow 1 0 blue 1 1 red		green	64×64 display elements	2 Dots: E_3 E_2 E_1 E_0 } 3 Dots	Color E_3 E_2 E_1 E_0	Color graphic mode 64x64
								1		0 0 white 0 1 cyan 1 0 magenta 1 1 orange		white				
⑥	1	1	X	X	0	0	1	0	X	D_7 (D_6, D_5, D_4, D_3, D_2, D_1, D_0) 0 black 1 green		green	128×64 display elements	2 Dots: 7 6 5 4 3 2 1 0 } 3 Dots	Luminance	Graphic mode 128x64
								1		0 black 1 white		white				
⑦	1	1	X	X	0	1	0	0	X	The same as ⑤		green	128×64 display elements	2 Dots: E_3 E_2 E_1 E_0 } 3 Dots	Color E_3 E_2 E_1 E_0	Graphic mode 128x64
								1				white				
⑧	1	1	X	X	0	1	1	0	X	The same as ⑥		green	128×96 display elements	2 Dots: 7 6 5 4 3 2 1 0 } 2 Dots	Luminance	Graphic mode 128x96
								1				white				
⑨	1	1	X	X	1	0	0	0	X	The same as ⑤		green	128×96 display elements	2 Dots: E_3 E_2 E_1 E_0 } 2 Dots	Color E_3 E_2 E_1 E_0	Color graphic mode 128x96
								1				white				
⑩	1	1	X	X	1	0	1	0	X	The same as ⑥		green	128×192 display elements	2 Dots: 7 6 5 4 3 2 1 0 } 1 Dot	Luminance	Graphic mode 128x192
								1				white				
⑪	1	1	X	X	1	1	0	0	X	The same as ⑤		green	128×192 display elements	2 Dots: E_3 E_2 E_1 E_0 } 1 Dot	Color E_3 E_2 E_1 E_0	Color graphic mode 128x192
								1				white				
⑫	1	1	X	X	1	1	1	0	X	The same as ⑥		green	256×192 display elements	1 Dot: 7 6 5 4 3 2 1 0 } 1 Dot	Luminance	Graphic mode 256x192
								1				white				

MITSUBISHI ELECTRIC

Internal Character Generator

The M5C6847P-1 generates the 64 standard ASCII characters in a 5 x 7 dot matrix form. It generates the 64 standard ASCII characters according to a 6-bit code. The code for each character is showed in Table 5.

Table 5 M5C6847P-1 character set

Code						Character
D_5	D_4	D_3	D_2	D_1	D_0	
0	0	0	0	0	0	@
0	0	0	0	0	1	A
0	0	0	0	1	0	B
0	0	0	0	1	1	C
0	0	0	1	0	0	D
0	0	0	1	0	1	E
0	0	0	1	1	0	F
0	0	0	1	1	1	G
0	0	1	0	0	0	H
0	0	1	0	0	1	I
0	0	1	0	1	0	J
0	0	1	0	1	1	K
0	0	1	1	0	0	L
0	0	1	1	0	1	M
0	0	1	1	1	0	N
0	0	1	1	1	1	O
0	1	0	0	0	0	P
0	1	0	0	0	1	Q
0	1	0	0	1	0	R
0	1	0	0	1	1	S
0	1	0	1	0	0	T
0	1	0	1	0	1	U
0	1	0	1	1	0	V
0	1	0	1	1	1	W
0	1	1	0	0	0	X
0	1	1	0	0	1	Y
0	1	1	0	1	0	Z
0	1	1	0	1	1	[
0	1	1	1	0	0	¥
0	1	1	1	0	1	]
0	1	1	1	1	0	↑
0	1	1	1	1	1	←

Code						Character
D_5	D_4	D_3	D_2	D_1	D_0	
1	0	0	0	0	0	SP
1	0	0	0	0	1	!
1	0	0	0	1	0	"
1	0	0	0	1	1	#
1	0	0	1	0	0	$
1	0	0	1	0	1	%
1	0	0	1	1	0	&
1	0	0	1	1	1	'
1	0	1	0	0	0	(
1	0	1	0	0	1	)
1	0	1	0	1	0	*
1	0	1	0	1	1	+
1	0	1	1	0	0	,
1	0	1	1	0	1	-
1	0	1	1	1	0	.
1	0	1	1	1	1	/
1	1	0	0	0	0	0
1	1	0	0	0	1	1
1	1	0	0	1	0	2
1	1	0	0	1	1	3
1	1	0	1	0	0	4
1	1	0	1	0	1	5
1	1	0	1	1	0	6
1	1	0	1	1	1	7
1	1	1	0	0	0	8
1	1	1	0	0	1	9
1	1	1	0	1	0	:
1	1	1	0	1	1	;
1	1	1	1	0	0	<
1	1	1	1	0	1	=
1	1	1	1	1	0	>
1	1	1	1	1	1	?

EXAMPLE OF DISPLAY ON CRT

The M5C6847P-1 can be used to generate characters for display on a video screen. An example of a display is shown in Fig. 1.

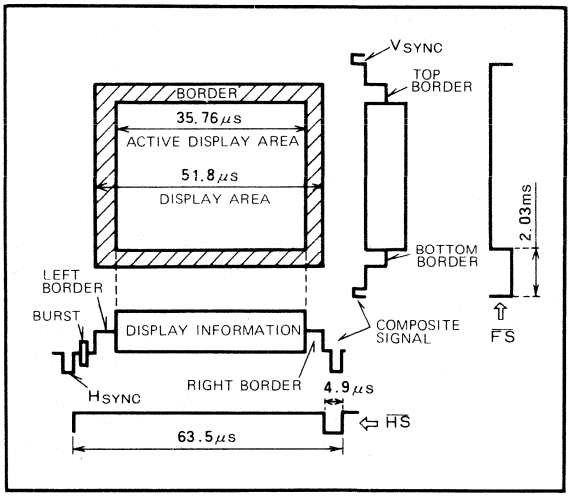


Fig. 1 Example of a display by a M5C6847P-1

APPLICATION EXAMPLE

One example of interfacing a M5C6847P-1 with a television set for home use is shown in Fig. 2. A M5L8085AP is used as the CPU in the example shown. The CPU executes the programs to control display and write the information for one screen into display memory. The M5C6847P-1 performs the main functions of interfacing with the CRT such as synchronizing scan, reading the display information from the display memory while adding necessary synchronization signals and sending to the RF modulator.

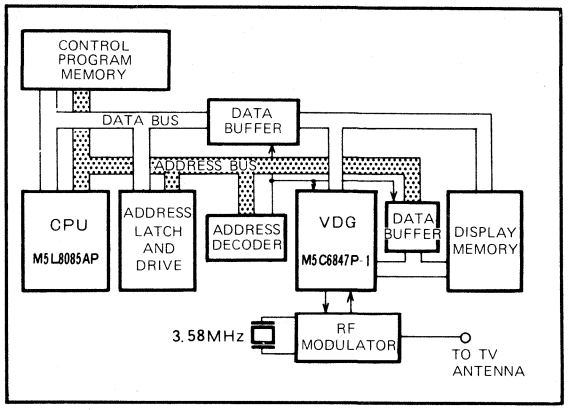


Fig. 2 Application example using the M5C6847P-1

9

VIDEO DISPLAY GENERATOR

A schematic for using the M5C6847P-1 with the M51342P RF modulator is shown in Fig. 3. M51342 requires ±5V power supplies. The video signal and chroma signal from the M5C6847P-1 can be modulated with the sound signal to form a RF signal that appears the same as the television antenna input signal. The video amp circuit to enable direct connection to a M5C6847P-1 is shown in Fig. 4. This can be connected to the monochrome video monitor. In this case, the inpedance is 75Ω.

Four levels of brightness (black, low, medium and high) can display a clear picture.

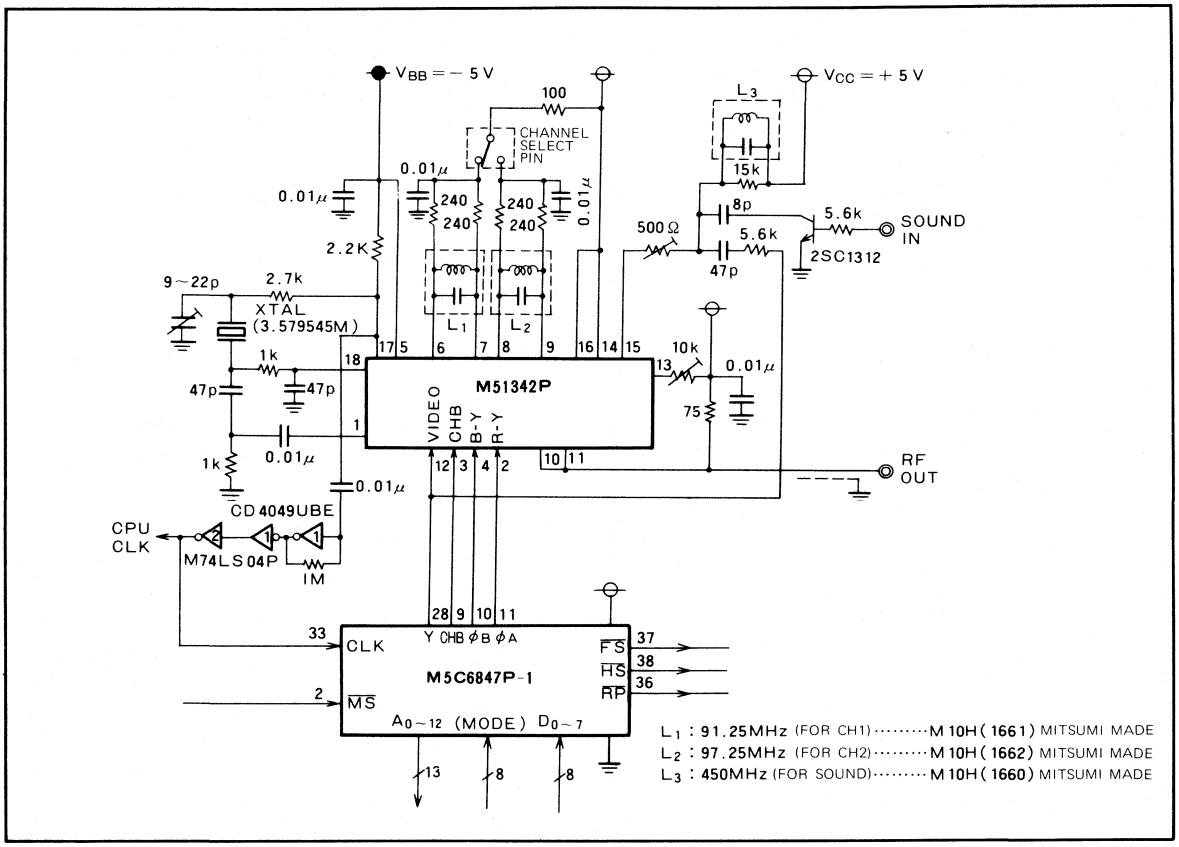


Fig. 3 Schematic for using the M51342P (RF modulator) with the M5C6847P-1

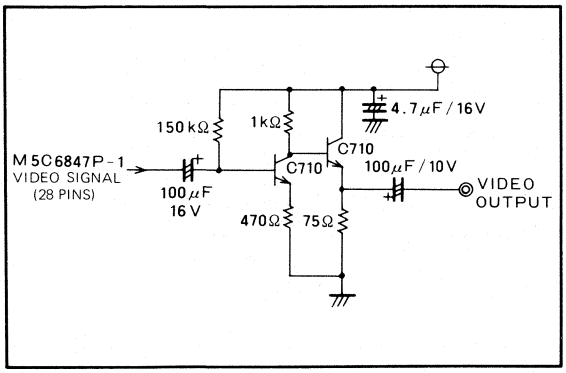


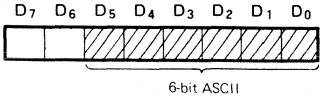
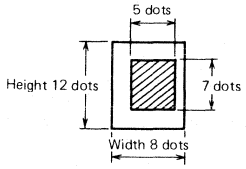
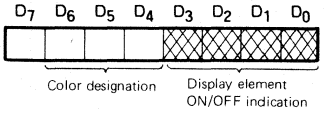
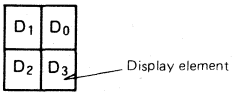
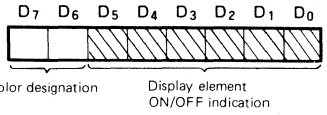
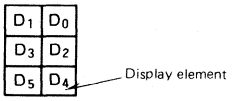
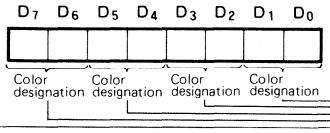
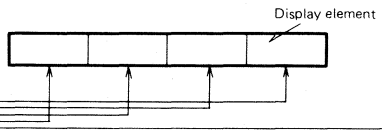
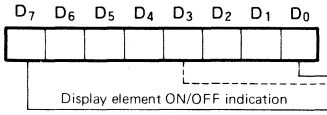
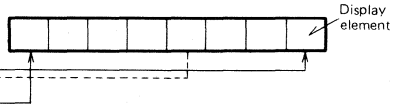
Fig. 4 Video amp circuit

MITSUBISHI ELECTRIC

Data and Display Relation

The relation between data and 5 display modes is shown in Table 6.

Table 6 Data and display relation

Mode	Data	Display
Character		
Semigraphic 4		
Semigraphic 6		
Color-graphic (4 colors)		
Graphic (2 colors)		

9

VIDEO DISPLAY GENERATOR

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{CC}	Supply voltage		−0.3～7	V
V_I	Input voltage	With respect to V_{SS}	−0.3～7	V
V_O	Output voltage		−0.3～7	V
Pd	Power dissipation	Ta＝25°C	1000	mW
Topr	Operating free-air temperature range		0～70	°C
Tstg	Storage temperature range		−65～150	°C

RECOMMENDED OPERATING CONDITIONS (Ta＝0～70°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{CC}	Supply voltage	4.75	5	5.25	V
V_{SS}	Supply voltage		0		V
$V_{IH(\phi)}$	High-level input voltage, clock	2.4		V_{CC}	V
V_{IH}	High-level input voltage	2		V_{CC}	V
$V_{IL(\phi)}$	Low-level input voltage, clock	−0.3		0.4	V
V_{IL}	Low-level input voltage	−0.3		0.8	V

ELECTRICAL CHARACTERISTICS (Ta＝0～70°C, V_{CC}＝5V±5%, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{OH}	High-level output voltage, except for ϕ_A, ϕ_B, Y, and CHB output	V_{SS}＝0V, I_{OH}＝−100μA, C_L＝30pF	2.4			V
V_{OL}	Low-level output voltage, except for ϕ_A, ϕ_B, Y and CHB output	V_{SS}＝0V, I_{OL}＝1.6mA, C_L＝30pF			0.4	V
I_{IH}	High-level input current	V_{SS}＝0V, V_I＝5.25V	−10		10	μA
I_{IL}	Low-level output current	V_{SS}＝0V, V_I＝0V	−10		10	μA
I_{OZ}	Output floating leak current	V_{SS}＝0V, V_I＝0.4V, MS＝0.4V	−10		10	μA
I_{CC}	Supply current from V_{CC}	V_{SS}＝0V			150	mA
Ci	Input capacitance	V_I＝0V, f＝1MHz, Ta＝25°C			10	pF
C_O	Output capacitance				20	pF
V_{CHB}	Chroma bias voltage	V_{SS}＝0V, C_L＝20pF, R_L＝200kΩ		0.6V_{CC}		V
$V_{\phi A,H}$	ϕ_A chrominance high-level output voltage			V_{CHB}＋0.16V_{CC}		V
$V_{\phi A,M}$	ϕ_A chrominance medium-level output voltage			V_{CHB}		V
$V_{\phi A,L}$	ϕ_A chrominance low-level output voltage			V_{CHB}−0.16V_{CC}		V
$V_{\phi B,H}$	ϕ_B chrominance high-level output voltage			V_{CHB}＋0.16V_{CC}		V
$V_{\phi B,M}$	ϕ_B chrominance medium-level output voltage			V_{CHB}		V
$V_{\phi B,B}$	ϕ_B chrominance burst-level output voltage			V_{CHB}−0.08V_{CC}		V
$V_{\phi B,L}$	ϕ_B chrominance low-level output voltage			V_{CHB}−0.16V_{CC}		V
V_{YSYNC}	Luminance sync output voltage			0.74V_{CC}		V
V_{YBLANK}	Luminance blank output voltage			0.85 V_{YSYNC}		V
V_{YBLACK}	Luminance black output voltage			0.81 V_{YSYNC}		V
$V_{YW(H)}$	White luminance high-level output voltage			0.62 V_{YSYNC}		V
$V_{YW(M)}$	White luminance medium-level output voltage			0.69 V_{YSYNC}		V
$V_{YW(L)}$	White luminance low-level output voltage			0.77 V_{YSYNC}		V

TIMING REQUIREMENTS ($Ta = 0 \sim 70°C$, $V_{CC} = 5V \pm 5\%$, $V_{SS} = 0V$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
$f_{C(\phi)}$	Clock frequency		3.579535	3.579545	3.579555	MHz
f_{DUTY}	Clock duty ratio		45	50	55	%
$t_{r(\phi)}$	Clock rise time				10	ns
$t_{f(\phi)}$	Clock fall time				10	ns
$t_{a(A\text{-}D)I}$	Address access time of display memory	Internal character mode			900	ns
$t_{a(A\text{-}D)E}$	Address access time of display memory + Address access time of external character ROM	External character mode			900	ns

SWITCHING CHARACTERISTICS

Composite video and chroma ($Ta = 0 \sim 70°C$, $V_{CC} = 5V \pm 5\%$, $V_{SS} = 0V$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
$t_{W(YSYNC)}$	Luminance output synchronization signal pulse width			4.89		μs
$t_{W(YFP)}$	Luminance output front pot signal pulse width			1.96		μs
$t_{W(YHBLANK)}$	Luminance output horizontal blank signal pulse width			11.73		μs
$t_{r(YHSYNC)}$	Luminance output horizontal synchronization signal rise time				250	ns
$t_{f(YHSYNC)}$	Luminance output horizontal synchronization signal fall time				250	ns
$t_{r(YHBLANK)}$	Luminance output horizontal blank signal rise time				340	ns
$t_{f(YHBLANK)}$	Luminance output horizontal blank signal fall time				340	ns

CHROMA

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
$t_{r(\phi A)}$	ϕA chrominance output rise time			60		ns
$t_{f(\phi A)}$	ϕA chrominance output fall time			60		ns
$t_{r(\phi B)}$	ϕB chrominance output rise time			60		ns
$t_{f(\phi B)}$	ϕB chrominance output fall time			60		ns
$t_{PHL(SYNC\text{-}BURST)}$	ϕB chrominance output propagation time after luminance syncronization signal output			980		ns
$t_{W(BURST)}$	ϕB chrominance output burst signal pulse width			2.93		μs
$t_{r(BURST)}$	ϕB chrominance output burst signal rise time			60		ns
$t_{f(BURST)}$	ϕB chrominance output burst signal fall time			60		ns
$t_{PHL(Y\text{-}CH)}$ / $t_{PLH(Y\text{-}CH)}$	Chrominance propagation time after luminance output			0		ns

MISCELLANEOUS

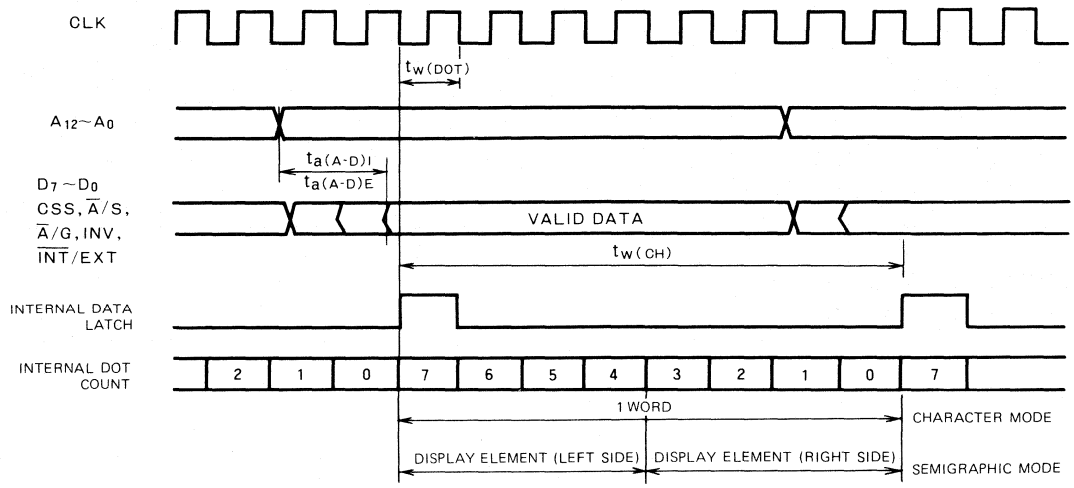
Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
$t_{W(FS)}$	Field syncronization pulse width			2.03		ms
$t_{W(RP)}$	Row preset pulse width			980		ns
$t_{PHL(HS\text{-}RP)}$	$\overline{RP}$ propagation time after $\overline{HS}$			980		ns
$t_{W(HS)}$	Horizontal syncronization pulse width			4.9		μs
$t_{W(CH)}$	Character width			1.12		μs
$t_{W(DOT)}$	Dot width			140		ns

9

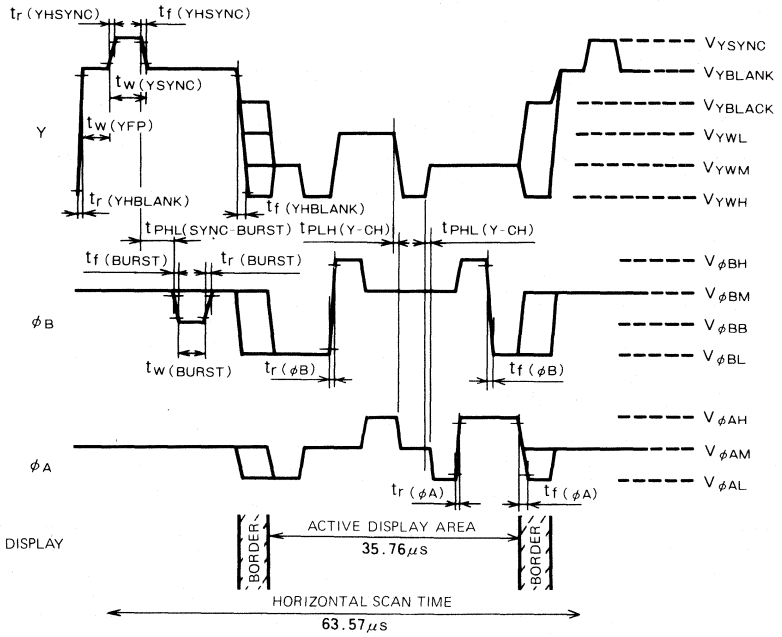
VIDEO DISPLAY GENERATOR

TIMING DIAGRAM

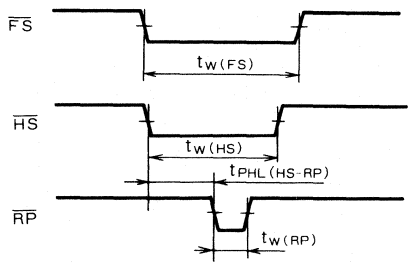
Display memory access



Composite video and chroma



Miscellaneous timing



2048-BIT STATIC RAM WITH I/O PORTS AND TIMER

DESCRIPTION

The M5L8155P is a 2K-bit RAM (256-word by 8-bit) fabricated with the N-channel silicon-gate ED-MOS technology. This IC has 3 I/O ports and a 14-bit counter/timer which make it a good choice to extend the functions of an 8-bit microcomputer. It is incased in a 40-pin plastic DIL package and operates with a single 5V power supply.

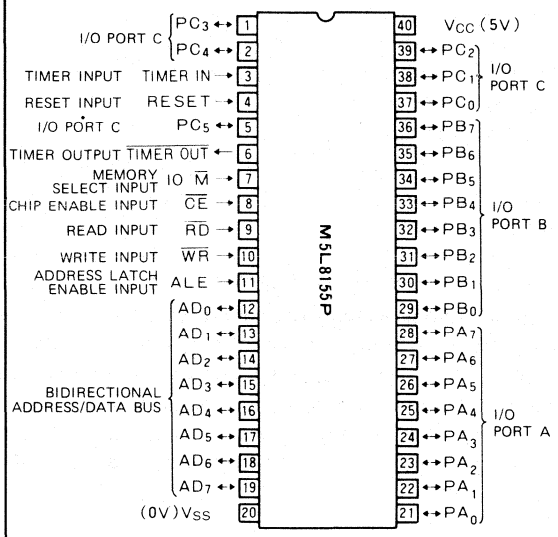
FEATURES

- Compatible with MELPS 85 devices
- Static RAM: 256 words by 8 bits
- Programmable 8-bit I/O port: 2
- Programmable 6-bit I/O port: 1
- Programmable counter/timer: 14 bits
- Multiplexed address/data bus
- Single 5V power supply
- Interchangeable with Intel's P8155 in pin
- Configuration and electrical characteristics

APPLICATION

- Extension of I/O ports and timer function for MELPS 8/85 and MELPS 8-48 devices

PIN CONFIGURATION (TOP VIEW)



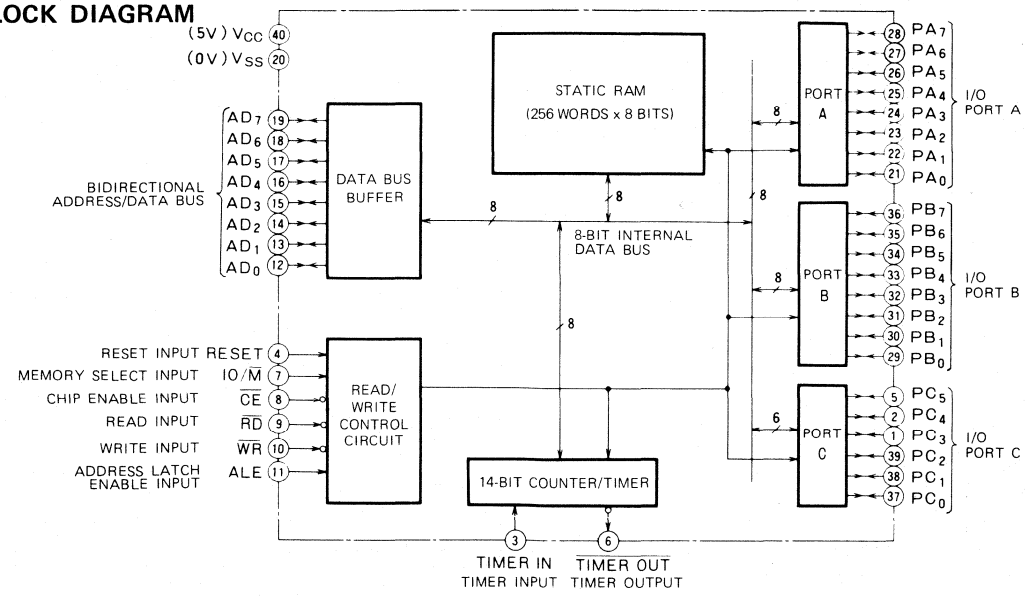
Outline 40P1

FUNCTION

The M5L8155P is composed of RAM, I/O ports and counter/timer. The RAM is a 2K-bit static RAM organized as 256 words by 8 bits. The I/O ports consist of 2 programmable 8-bit ports and 1 programmable 6-bit port. The terminals of the 6-bit port can be programmed to function as control terminals for the 8-bit ports, so that the 8-bit ports can be operated in a handshake mode. The counter/timer is composed of 14 bits that can be used to count down (events or time) and it can generate square wave pulses that can be used for counting and timing.

9

BLOCK DIAGRAM



2048-BIT STATIC RAM WITH I/O PORTS AND TIMER

OPERATION

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to transfer the data while input or output instructions are being executed by the CPU. Command and address information is also transferred through the data bus buffer.

Read/Write Control Logic

The read/write control logic controls the transfer of data by interpreting I/O control bus output signals ($\overline{RD}$, $\overline{WR}$, IO/$\overline{M}$ and ALE) along with CPU signal ($\overline{CE}$). RESET signal is also used to control the transfer of data and commands.

Bidirectional Address/Data Bus ($AD_0 \sim AD_7$)

The bidirectional address/data bus is a 3-state 8-bit bus. The 8-bit address is latched in the internal latch by the falling edge of ALE. Then if IO/$\overline{M}$ input signal is at high-level, the address of I/O port, counter/timer, or command register is selected. If it is at low-level, memory address is selected.

The 8-bit address data is transferred by read input ($\overline{RD}$) or write input ($\overline{WR}$).

Chip Enable Input ($\overline{CE}$)

When $\overline{CE}$ is at low-level, the address information on address/data bus is stored in the M5L8155P

Read Input ($\overline{RD}$)

When $\overline{RD}$ is at low-level the data bus buffer is active. If IO/$\overline{M}$ input signal is at low-level, the contents of RAM are read through the address/data bus. If IO/$\overline{M}$ input is at high-level, the selected contents of I/O port or counter/timer are read through the address/data bus.

Write Input ($\overline{WR}$)

When $\overline{WR}$ is at low-level, the data on the address/data bus are written into RAM if IO/$\overline{M}$ is at low-level, or if IO/$\overline{M}$ is at high-level they are written into I/O port, counter/timer or command register.

Address Latch Enable Input (ALE)

An address on the address/data bus along with the levels of CE and IO/$\overline{M}$ are latched in the M5L8155P on the falling edge of ALE.

IO/Memory Input (IO/$\overline{M}$)

When IO/$\overline{M}$ is at low-level, the RAM is selected, while at high-level the I/O port, counter/timer or command register are selected.

I/O Port A ($PA_0 \sim PA_1$)

Port A is an 8-bit general-purpose I/O port. Input/output setting is controlled by the system software.

I/O Port B ($PB_0 \sim PB_7$)

Port B is an 8-bit general-purpose I/O port. Input/output setting is controlled by the system software.

I/O Port C ($PC_0 \sim PC_5$)

Port C is a 6-bit I/O port that can also be used to output control signals of port A (PA) or port B (PB). The functions of port C are controlled by the system software. When port C is used to output control signals of ports A or B the assignment of the signals to the pins is as shown in Table 1.

Table 1 Pin assignment of control signals of port C

Pin	Function	
PC_5	$\overline{B\ STB}$	(port B strobe)
PC_4	B BF	(port B buffer full)
PC_3	B INTR	(port B interrupt)
PC_2	$\overline{A\ STB}$	(port A strobe)
PC_1	A BF	(port A buffer full)
PC_0	A INTR	(port A interrupt)

Timer Input (TIMER IN)

The signal at this input terminal is used by the counter/timer for counting events or time. (3MHz max.)

Timer Output ($\overline{TIMER\ OUT}$)

A square wave signal or pulse from the counter/timer is output through this pin when in the operation mode.

Command Register (8 bits)

The command register is an 8-bit latched register. The low-order 4 bits (bits 0 ~ 3) are used for controlling and determination of the mode of the ports. Bits 4 and 5 are used as interrupt enable flags for ports A and B when port C is used as a control port. Bits 6 and 7 are used for controlling the counter/timer. The contents of the command register are rewritten by output instructions (address I/O XXXXX000).

Details of the functions of the individual bits of the command register are shown in Fig. 1.

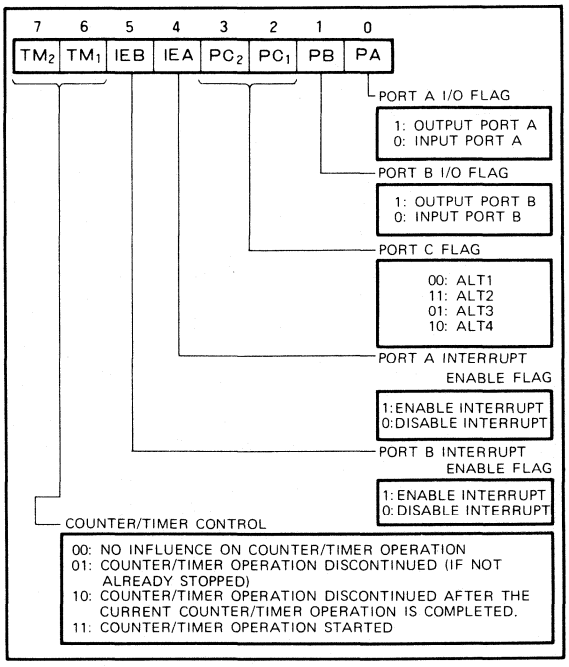


Fig. 1 Bit functions of the command register

MITSUBISHI ELECTRIC

Status Register (7 bits)

The status register is a 7-bit latched register. The low-order 5 bits (bits 0 ~ 4) are used as status flags for the I/O ports. Bit 6 is as a status flag for the counter/timer. The contents of the status register are transferred into the CPU by reading (INPUT instruction, address I/O XXXXX000). Details of the functions of the individual bits of the status register are shown in Fig. 2.

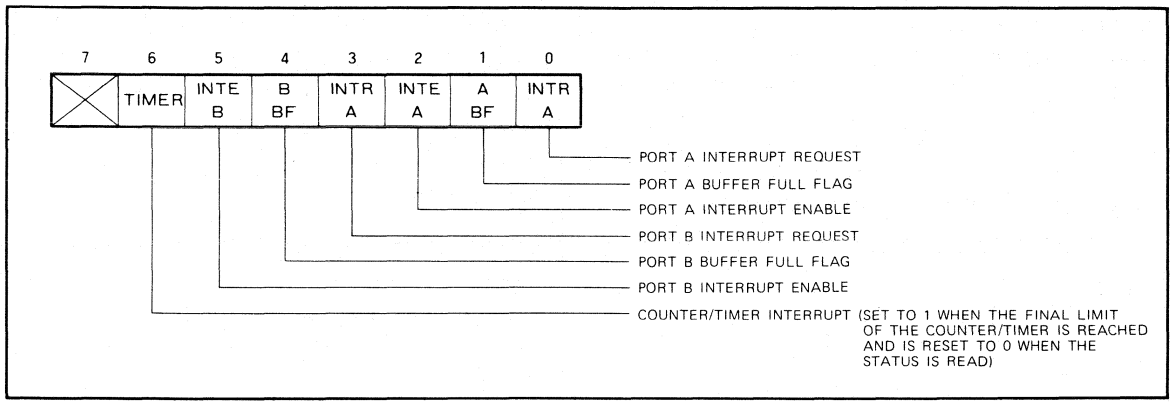


Fig. 2 Bit functions of the status register

I/O Ports

Command/status registers (8 bits/7 bits)

These registers are assigned address XXXXX000. When executing an OUTPUT instruction, the contents of the command register are rewritten. When executing an INPUT instruction the contents of the status register are read.

Port A Register (8 bits)

Port A register is assigned address XXXXX001. This register can be programmed as an input or output by setting the appropriate bits of the command register as shown in Fig. 1.

Port A can be operated in basic or strobe mode and is assigned I/O terminal $PA_0 \sim PA_7$.

Port B Register (8 bits)

Port B register is assigned address XXXXX010. As with Port A register, this register can be programmed as an input or output by setting the appropriate bits of the command register as shown in Fig. 1. Port B can be operated in basic or strobe mode and is assigned I/O terminals $PB_0 \sim PB_7$.

Port C Register (6 bits)

Port C register is assigned address XXXXX011. This port is used for controlling input/output operations of ports A and B by selectively setting bits 2 and 3 of the command register as shown in Fig. 1. Details of the functions of the various setting of bits 2 and 3 are shown in Table 2. Port C is assigned I/O terminals $PC_0 \sim PC_5$ and when used as port control signals, the 3 low-order bits are assigned for port A while the 3 high-order bits are assigned for port B.

9

Table 2 Functions of port C

State / Terminal	ALT1	ALT2	ALT 3	ALT 4
PC_5	Input	Output	Output	$\overline{\text{B STB}}$ (port B strobe)
PC_4	Input	Output	Output	B BF (port B buffer full)
PC_3	Input	Output	Output	B INTR (port B interrupt)
PC_2	Input	Output	$\overline{\text{A STB}}$ (port A strobe)	$\overline{\text{A STB}}$ (port A strobe)
PC_1	Input	Output	A BF (port A buffer full)	A BF (port A buffer full)
PC_0	Input	Output	A INTR (port A interrupt)	A INTR (port A interrupt)

2048-BIT STATIC RAM WITH I/O PORTS AND TIMER

Configuration of Ports

A block diagram of 1 bit of ports A and B is shown in Fig. 3. While port A or B is programmed as an output port, if the port is addressed by an input instruction, the contents of the selected port can be read. When a port is put in input mode, the output latch is cleared and writing into the output latch is disabled. Therefore when a port is changed to output mode from input mode, low-level signals are output through the port. When a reset signal is applied, all 3 ports (PA, PB, and PC) will be input ports and their output latches are cleared. Port C has the same configuration as ports A and B in modes ALT1 and ALT2.

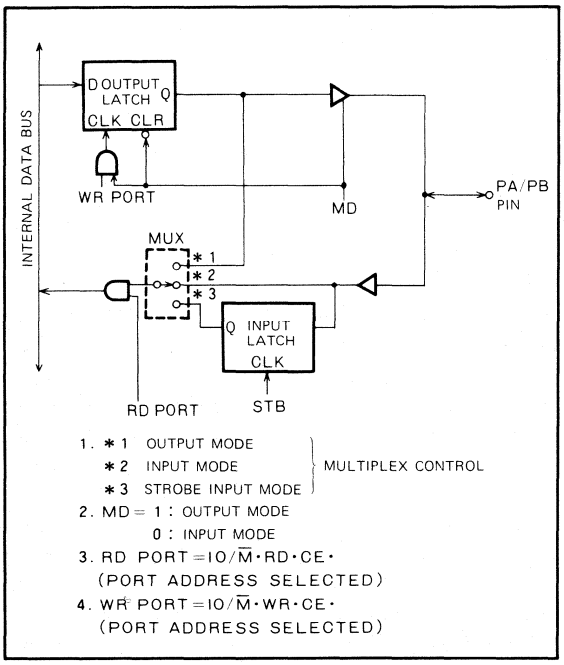


Fig. 3 Configuration for 1 bit of port A or B

The basic functions of the I/O ports are shown in Table 3. The control signal levels to ports A and B, when port C is programmed as a control port, are shown in Table 4.

Table 3 Basic functions of I/O ports

Address	$\overline{RD}$	$\overline{WR}$	Function
XXXXX000	0	1	AD bus ← status register
	1	0	Command register ← AD bus
XXXXX001	0	1	AD bus ← port A
	1	0	Port A ← AD bus
XXXXX010	0	1	AD bus ← port B
	1	0	Port B ← AD bus
XXXXX011	0	1	AD bus ← port C
	1	0	Port C ← AD bus

Table 4 Port control signal levels at ALT3 and ALT4

Control signal	Input mode	Output mode
BF	"L"	"L"
INTR	"L"	"H"
$\overline{STB}$	Input	Input

Counter/Timer

The counter/timer is a 14-bit counting register plus 2 mode flags. The register has two sections: address I/O XXXXX100 is assigned to the low-order 8 bits and address I/O XXXXX101 is assigned to the high-order 8 bits. The low-order bits 0 ~ 13 are used for counting or timing. The counter is initialized by the program and then counted down to zero. The initial setting can range from 2_{16} to $3FFF_{16}$. Bits 14 and 15 are used as mode flags.

The mode flags select 1 of 4 modes with functions as follow:

- Mode 0: Outputs high-level signal during the former half of the counter operation
  Outputs low-level signal during the latter half of the counter operation
- Mode 1: Outputs square wave signals as in mode 0
- Mode 2: Outputs a low-level pulse during the final count down
- Mode 3: Outputs a low-level pulse during each final count down

Starting and stopping the counter/timer is controlled by bits 6 and 7 of the command register (see Fig. 1 for details). The format and timer modes of the counter/timer register are shown in Fig. 4 and Table 5.

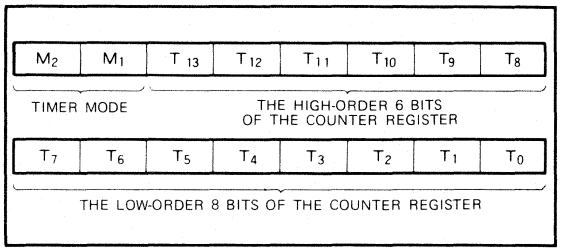


Fig. 4 Format of counter/timer

The counter/timer is not influenced by a reset, but counting is discontinued. To resume counting, a start command must be written into the command register as shown in Fig. 1. While operating 2n+1 count down in mode 0, a high-level signal is output during the n+1 counting and a low-level signal is output during the n counting.

Table 5 Timer mode

M_2	M_1	Timer operation
0	0	Outputs high-level signal during the former half of the counter operation Outputs low-level signal during the latter half of the counter operation (mode 0)
0	1	Outputs square wave signals as in mode 0 (mode 1)
1	0	Outputs a low-level pulse during the final count down (mode 2)
1	1	Outputs a low-level pulse during each final count down (mode 3)

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{CC}	Supply voltage	With respect to V_{SS}	−0.5 ~ 7	V
V_I	Input voltage		−0.5 ~ 7	V
V_O	Output voltage		−0.5 ~ 7	V
Pd	Maximum power dissipation	Ta = 25°C	1.5	W
Topr	Operating free-air temperature range		0 ~ 70	°C
Tstg	Storage temperature range		−65 ~ 150	°C

RECOMMENDED OPERATING CONDITIONS (Ta = 0 ~ 70°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{CC}	Supply voltage	4.75	5	5.25	V
V_{SS}	Power-supply voltage		0		V
V_{IL}	Low-level input voltage	−0.5		0.8	V
V_{IH}	High-level input voltage	2		V_{CC} + 0.5	V

ELECTRICAL CHARACTERISTICS (Ta = 0 ~ 70°C, V_{CC} = 5V ± 5%, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{OH}	High-level output voltage	V_{SS} = 0V, I_{OH} = −400μA	2.4			V
V_{OL}	Low-level output voltage	V_{SS} = 0V, I_{OL} = 2mA			0.45	V
I_I	Input leak current	V_{SS} = 0V, V_I = 0 ~ V_{CC}	−10		10	μA
$I_{I(CE)}$	Input leak current, $\overline{CE}$ pin	V_{SS} = 0V, V_I = 0 ~ V_{CC}	−100		100	μA
I_{OZ}	Output floating leak current	V_{SS} = 0V, V_I = 0.45 ~ V_{CC}	−10		10	μA
C_I	Input capacitance	V_{IL} = 0V, f = 1MHz, 25mVrms, Ta = 25°C			10	pF
$C_{I/O}$	Input/output terminal capacitance	$V_{I/OL}$ = 0V, f = 1MHz, 25mVrms, Ta = 25°C			20	pF
I_{CC}	Supply current from V_{CC}	V_{SS} = 0V			180	mA

Note 5 Current flowing into an IC is positive, out is negative.

2048-BIT STATIC RAM WITH I/O PORTS AND TIMER

TIMING REQUIREMENTS (Ta = 0 ~ 70°C, V_{CC} = 5V ± 5%, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{su(A-L)}$	Address setup time before latch	t_{AL}		50			ns
$t_{h(L-A)}$	Address hold time after latch	t_{LA}		80			ns
$t_{h(L-RWH)}$	Read/write hold time after latch	t_{LC}		100			ns
$t_{w(L)}$	Latch pulse width	t_{LL}		100			ns
$t_{h(RW-L)}$	Latch hold time after read/write	t_{CL}		20			ns
$t_{w(RWL)}$	Read/write low-level pulse width	t_{CC}		250			ns
$t_{su(D-W)}$	Data setup time before write	t_{DW}		150			ns
$t_{h(W-D)}$	Data hold time after write	t_{WD}		0			ns
$t_{w(RWH)}$	Read/write high-level pulse width	t_{RV}		300			ns
$t_{su(P-R)}$	Port setup time before read	t_{PR}		70			ns
$t_{h(R-P)}$	Port hold time after read	t_{RP}		50			ns
$t_{w(STB)}$	Strobe pulse width	t_{SS}		200			ns
$t_{su(P-STB)}$	Port setup time before strobe	t_{PSS}		50			ns
$t_{h(STB-P)}$	Port hold time after strobe	t_{PHS}		120			ns
$t_{w(\phi H)}$	Timer input high-level pulse width	t_2		120			ns
$t_{w(\phi L)}$	Timer input low-level pulse width	t_1		80			ns
$t_{c(\phi)}$	Timer input cycle time	t_{CYC}		320			ns
$t_{r(\phi)}$	Timer input rise time	t_r				30	ns
$t_{f(\phi)}$	Timer input fall time	t_f				30	ns

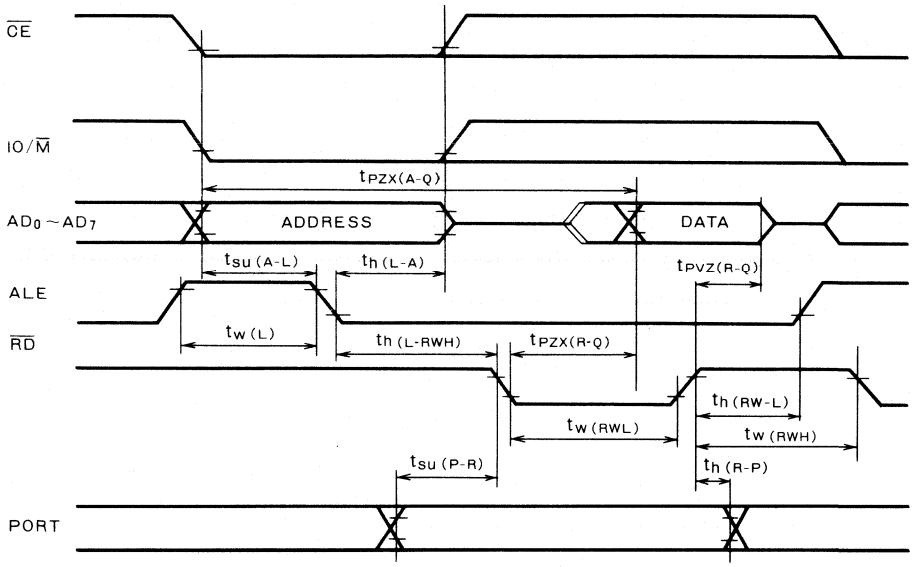
SWITCHING CHARACTERISTICS (Ta = 0 ~ 70°C, V_{CC} = 5V ± 5%, unless otherwise noted.)

Symbol	Parameter	Alternative symbol		Limits			Unit
				Min	Typ	Max	
$t_{PZX(R-Q)}$	Propagation time from read to data output	t_{RD}				170	ns
$t_{PZX(A-Q)}$	Propagation time from address to data output	t_{AD}				400	ns
$t_{PVZ(R-Q)}$	Propagation time from read to data floating (Note 7)	t_{RDF}				100	ns
$t_{PHL(W-P)}$	Propagation time from write to data output	t_{WP}				400	ns
$t_{PLH(W-P)}$		t_{WP}					
$t_{PLH(STB-BF)}$	Propagation time from strobe to BF flag	t_{SBF}				400	ns
$t_{PHL(R-BF)}$	Propagation time from read to BF flag	t_{RBE}				400	ns
$t_{PLH(STB-INTR)}$	Propagation time from strobe to interrupt	t_{SI}				400	ns
$t_{PHL(R-INTR)}$	Propagation time from read to interrupt	t_{RDI}				400	ns
$t_{PHL(STB-BF)}$	Propagation time from strobe to BF flag	t_{SBE}				400	ns
$t_{PLH(W-BF)}$	Propagation time from write to BF flag	t_{WBF}				400	ns
$t_{PHL(W-INTR)}$	Propagation time from write to interrupt	t_{WI}				400	ns
$t_{PHL(\phi-OUT)}$	Propagation time from timer input to timer output	t_{TL}				400	ns
$t_{PLH(\phi-OUT)}$		t_{TH}					

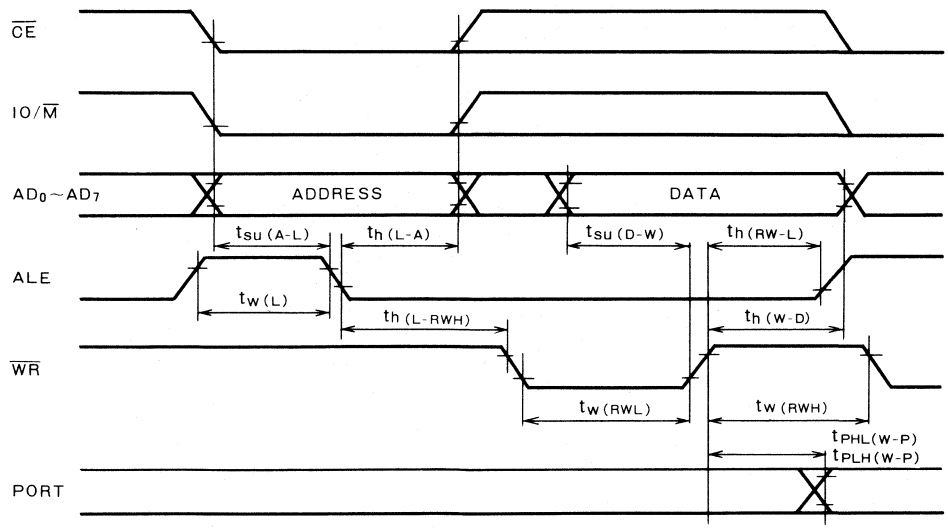
Note 6: Measurement conditions C = 150pF
7: Measurement conditions of note 6 are not applied.

TIMING DIAGRAM (reference level, high-level=2V, low-level=0.8V)

Basic Input



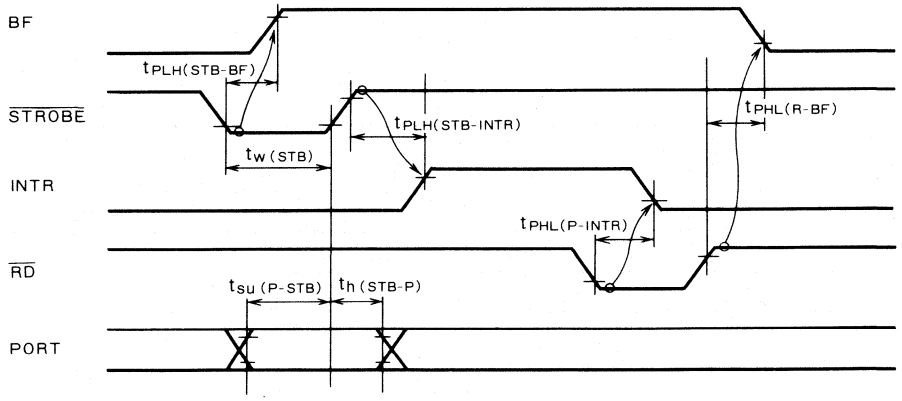
Basic Output



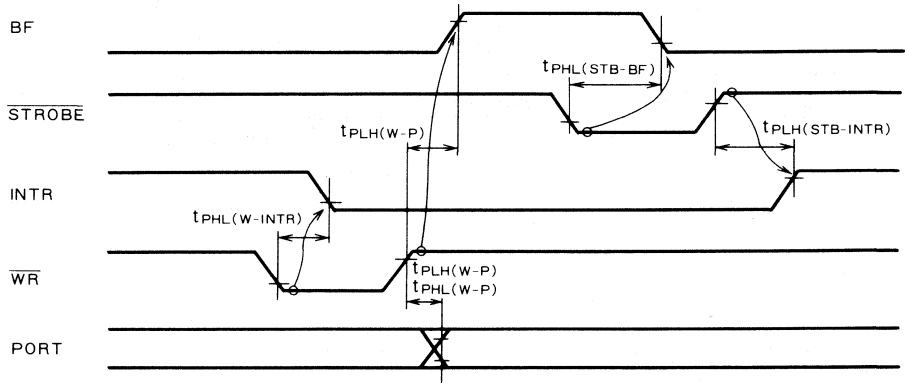
9

2048-BIT STATIC RAM WITH I/O PORTS AND TIMER

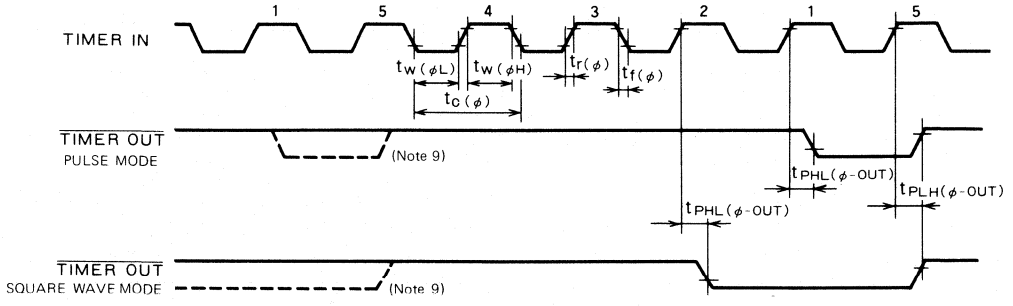
Strobed Input



Strobed Output



Timer (Note 8)



Note 8: The wave form is shown counting down from 5 to 1.
9: As long as the M1 mode flag of the timer register is at high-level, pulses are continuously output.

MITSUBISHI ELECTRIC

2048-BIT STATIC RAM WITH I/O PORTS AND TIMER

DESCRIPTION

The M5L8156P is a 2K-bit RAM (256-word by 8-bit) fabricated with the N-channel silicon-gate ED-MOS technology. This IC has 3 I/O ports and a 14-bit counter/timer which make it a good choice to extend the functions of an 8-bit microcomputer. It is incased in a 40-pin plastic DIL package and operates with a single 5V power supply.

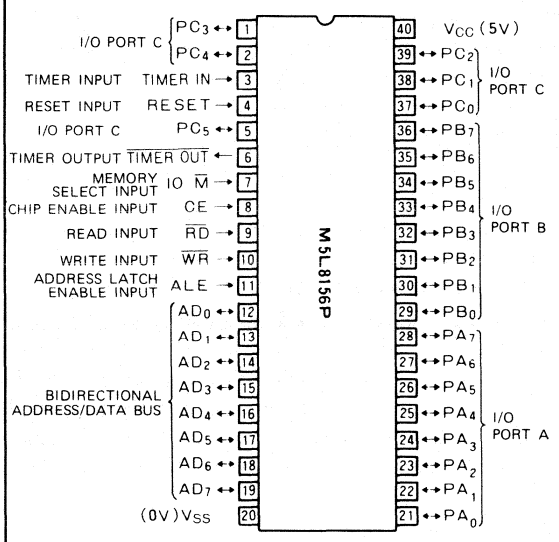
FEATURES

- Compatible with MELPS 85 devices
- Static RAM: 256 words by 8 bits
- Programmable 8-bit I/O port: 2
- Programmable 6-bit I/O port: 1
- Programmable counter/timer: 14 bits
- Multiplexed address/data bus
- Single 5V power supply
- Interchangeable with Intel's P8156 in pin
- Configuration and electrical characteristics

APPLICATION

- Extension of I/O ports and timer function for MELPS 8/85 and MELPS 8-48 devices

PIN CONFIGURATION (TOP VIEW)

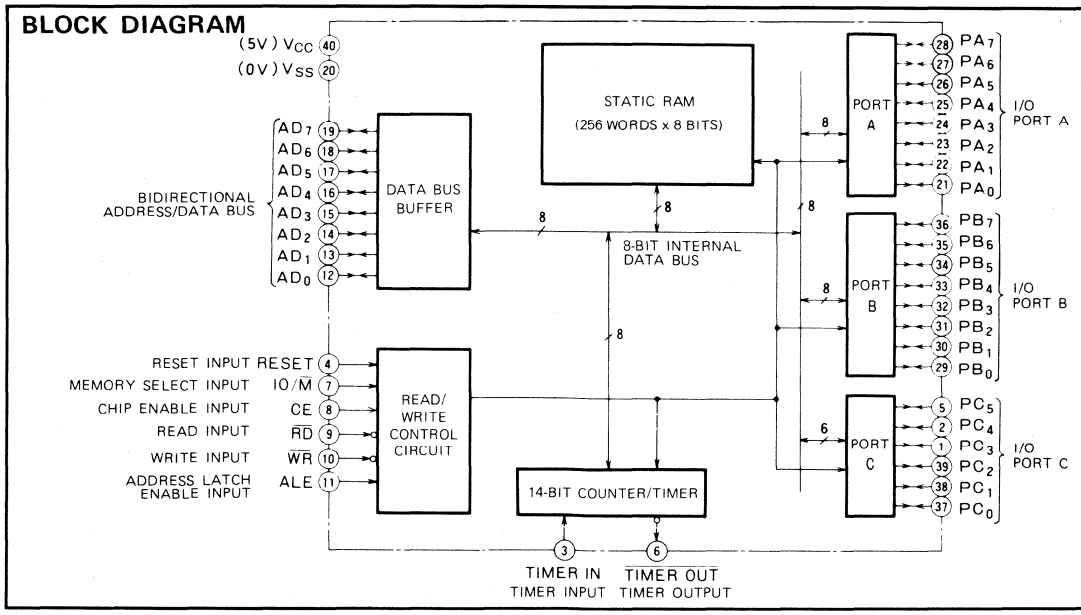


Outline 40P1

FUNCTION

The M5L8156P is composed of RAM, I/O ports and counter/timer. The RAM is a 2K-bit static RAM organized as 256 words by 8 bits. The I/O ports consist of 2 programmable 8-bit ports and 1 programmable 6-bit port. The terminals of the 6-bit port can be programmed to function as control terminals for the 8-bit ports, so that the 8-bit ports can be operated in a handshake mode. The counter/timer is composed of 14 bits that can be used to count down (events or time) and it can generate square wave pulses that can be used for counting and timing.

BLOCK DIAGRAM



9

2048-BIT STATIC RAM WITH I/O PORTS AND TIMER

OPERATION

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to transfer the data while input or output instructions are being executed by the CPU. Command and address information is also transferred through the data bus buffer.

Read/Write Control Logic

The read/write control logic controls the transfer of data by interpreting I/O control bus output signals ($\overline{RD}$, $\overline{WR}$, IO/$\overline{M}$ and ALE) along with CPU signal (CE). RESET signal is also used to control the transfer of data and commands.

Bidirectional Address/Data Bus ($AD_0 \sim AD_7$)

The bidirectional address/data bus is a 3-state 8-bit bus. The 8-bit address is latched in the internal latch by the falling edge of ALE. Then if IO/$\overline{M}$ input signal is at high-level, the address of I/O port, counter/timer, or command register is selected. If it is at low-level, memory address is selected.

The 8-bit address data is transferred by read input ($\overline{RD}$) or write input ($\overline{WR}$).

Chip Enable Input (CE)

When CE is at high-level, the address information on address/data bus is stored in the M5L8156P.

Read Input ($\overline{RD}$)

When $\overline{RD}$ is at low-level the data bus buffer is active. If IO/$\overline{M}$ input signal is at low-level, the contents of RAM are read through the address/data bus. If IO/$\overline{M}$ input is at high-level, the selected contents of I/O port or counter/timer are read through the address/data bus.

Write Input ($\overline{WR}$)

When $\overline{WR}$ is at low-level, the data on the address/data bus are written into RAM if IO/$\overline{M}$ is at low-level, or if IO/$\overline{M}$ is at high-level they are written into I/O port, counter/timer or command register.

Address Latch Enable Input (ALE)

An address on the address/data bus along with the levels of CE and IO/$\overline{M}$ are latched in the M5L8156P on the falling edge of ALE.

IO/Memory Input (IO/$\overline{M}$)

When IO/$\overline{M}$ is at low-level, the RAM is selected, while at high-level the I/O port, counter/timer or command register are selected.

I/O Port A ($PA_0 \sim PA_1$)

Port A is an 8-bit general-purpose I/O port. Input/output setting is controlled by the system software.

I/O Port B ($PB_0 \sim PB_7$)

Port B is an 8-bit general-purpose I/O port. Input/output setting is controlled by the system software.

I/O Port C ($PC_0 \sim PC_5$)

Port C is a 6-bit I/O port that can also be used to output control signals of port A (PA) or port B (PB). The functions of port C are controlled by the system software. When port C is used to output control signals of ports A or B the assignment of the signals to the pins is as shown in Table 1.

Table 1 Pin assignment of control signals of port C

Pin	Function	
PC_5	$\overline{B\ STB}$	(port B strobe)
PC_4	B BF	(port B buffer full)
PC_3	B INTR	(port B interrupt)
PC_2	$\overline{A\ STB}$	(port A strobe)
PC_1	A BF	(port A buffer full)
PC_0	A INTR	(port A interrupt)

Timer Input (TIMER IN)

The signal at this input terminal is used by the counter/timer for counting events or time. (3MHz max.)

Timer Output ($\overline{TIMER\ OUT}$)

A square wave signal or pulse from the counter/timer is output through this pin when in the operation mode.

Command Register (8 bits)

The command register is an 8-bit latched register. The low-order 4 bits (bits 0 ~ 3) are used for controlling and determination of the mode of the ports. Bits 4 and 5 are used as interrupt enable flags for ports A and B when port C is used as a control port. Bits 6 and 7 are used for controlling the counter/timer. The contents of the command register are rewritten by output instructions (address I/O XXXXX000).

Details of the functions of the individual bits of the command register are shown in Fig. 1.

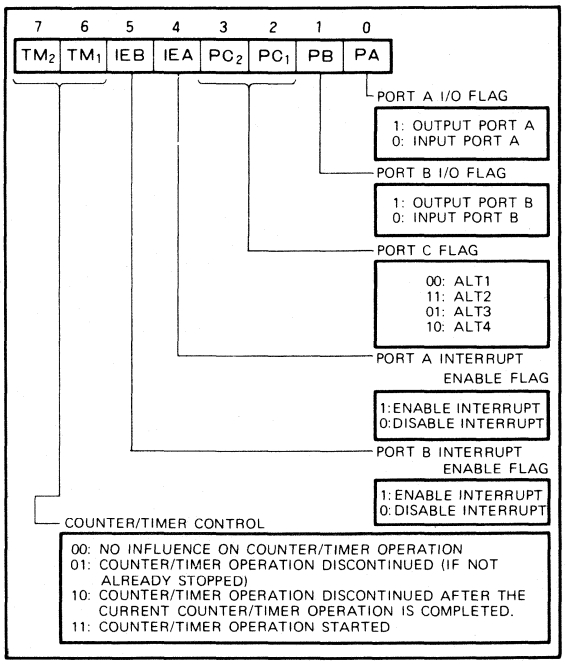


Fig. 1 Bit functions of the command register

Status Register (7 bits)

The status register is a 7-bit latched register. The low-order 5 bits (bits 0 ~ 4) are used as status flags for the I/O ports. Bit 6 is as a status flag for the counter/timer. The contents of the status register are transferred into the CPU by reading (INPUT instruction, address I/O XXXXX000). Details of the functions of the individual bits of the status register are shown in Fig. 2.

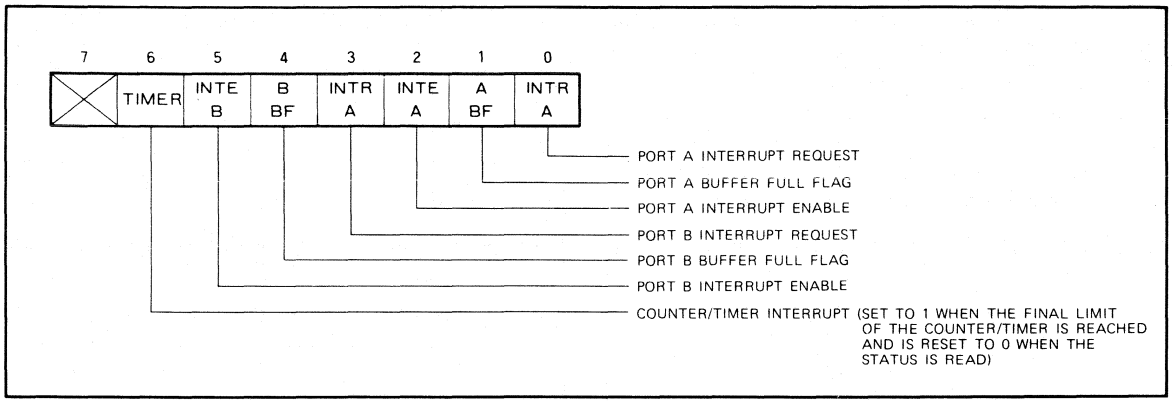


Fig. 2 Bit functions of the status register

I/O Ports

Command/status registers (8 bits/7 bits)

These registers are assigned address XXXXX000. When executing an OUTPUT instruction, the contents of the command register are rewritten. When executing an INPUT instruction the contents of the status register are read.

Port A Register (8 bits)

Port A register is assigned address XXXXX001. This register can be programmed as an input or output by setting the appropriate bits of the command register as shown in Fig. 1.

Port A can be operated in basic or strobe made and is assigned I/O terminal $PA_0 \sim PA_7$.

Port B Register (8 bits)

Port B register is assigned address XXXXX010. As with Port A register, this register can be programmed as an input or output by setting the appropriate bits of the command register as shown in Fig. 1. Port B can be operated in basic or strobe mode and is assigned I/O terminals $PB_0 \sim PB_7$.

Port C Register (6 bits)

Port C register is assigned address XXXXX011. This port is used for controlling input/output operations of ports A and B by selectively setting bits 2 and 3 of the command register as shown in Fig. 1. Details of the functions of the various setting of bits 2 and 3 are shown in Table 2. Port C is assigned I/O terminals $PC_0 \sim PC_5$ and when used as port control signals, the 3 low-order bits are assigned for port A while the 3 high-order bits are assigned for port B.

9

Table 2 Functions of port C

State / Terminal	ALT1	ALT2	ALT 3	ALT 4
PC_5	Input	Output	Output	$\overline{\text{B STB}}$ (port B strobe)
PC_4	Input	Output	Output	B BF (port B buffer full)
PC_3	Input	Output	Output	B INTR (port B interrupt)
PC_2	Input	Output	$\overline{\text{A STB}}$ (port A strobe)	$\overline{\text{A STB}}$ (port A strobe)
PC_1	Input	Output	A BF (port A buffer full)	A BF (port A buffer full)
PC_0	Input	Output	A INTR (port A interrupt)	A INTR (port A interrupt)

2048-BIT STATIC RAM WITH I/O PORTS AND TIMER

Configuration of Ports

A block diagram of 1 bit of ports A and B is shown in Fig. 3. While port A or B is programmed as an output port, if the port is addressed by an input instruction, the contents of the selected port can be read. When a port is put in input mode, the output latch is cleared and writing into the output latch is disabled. Therefore when a port is changed to output mode from input mode, low-level signals are output through the port. When a reset signal is applied, all 3 ports (PA, PB, and PC) will be input ports and their output latches are cleared. Port C has the same configuration as ports A and B in modes ALT1 and ALT2.

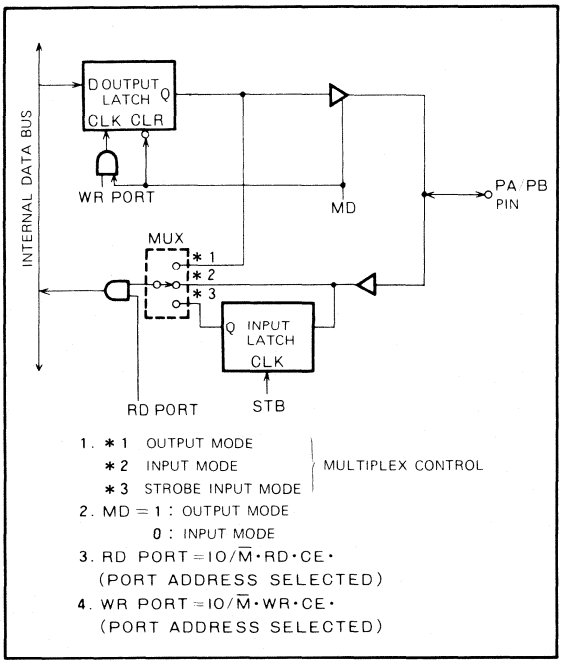


Fig. 3 Configuration for 1 bit of port A or B

The basic functions of the I/O ports are shown in Table 3. The control signal levels to ports A and B, when port C is programmed as a control port, are shown in Table 4.

Table 3 Basic functions of I/O ports

Address	$\overline{RD}$	$\overline{WR}$	Function
XXXXX000	0	1	AD bus ← status register
	1	0	Command register ← AD bus
XXXXX001	0	1	AD bus ← port A
	1	0	Port A ← AD bus
XXXXX010	0	1	AD bus ← port B
	1	0	Port B ← AD bus
XXXXX011	0	1	AD bus ← port C
	1	0	Port C ← AD bus

Table 4 Port control signal levels at ALT3 and ALT4

Control signal	Input mode	Output mode
BF	"L"	"L"
INTR	"L"	"H"
$\overline{STB}$	Input	Input

Counter/Timer

The counter/timer is a 14-bit counting register plus 2 mode flags. The register has two sections: address I/O XXXXX100 is assigned to the low-order 8 bits and address I/O XXXXX101 is assigned to the high-order 8 bits. The low-order bits 0 ~ 13 are used for counting or timing. The counter is initialized by the program and then counted down to zero. The initial setting can range from 2_{16} to $3FF_{16}$. Bits 14 and 15 are used as mode flags.

The mode flags select 1 of 4 modes with functions as follow:

- Mode 0: Outputs high-level signal during the former half of the counter operation
  Outputs low-level signal during the latter half of the counter operation
- Mode 1: Outputs square wave signals as in mode 0
- Mode 2: Outputs a low-level pulse during the final count down
- Mode 3: Outputs a low-level pulse during each final count down

Starting and stopping the counter/timer is controlled by bits 6 and 7 of the command register (see Fig. 1 for details). The format and timer modes of the counter/timer register are shown in Fig. 4 and Table 5.

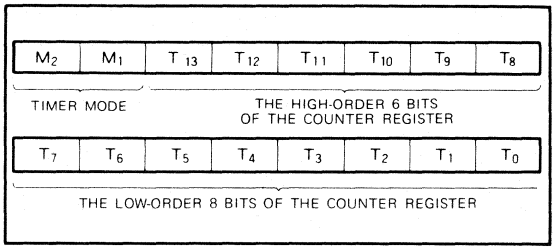


Fig. 4 Format of counter/timer

The counter/timer is not influenced by a reset, but counting is discontinued. To resume counting, a start command must be written into the command register as shown in Fig. 1. While operating 2n+1 count down in mode 0, a high-level signal is output during the n+1 counting and a low-level signal is output during the n counting.

Table 5 Timer mode

M_2	M_1	Timer operation
0	0	Outputs high-level signal during the former half of the counter operation Outputs low-level signal during the latter half of the counter operation (mode 0)
0	1	Outputs square wave signals as in mode 0 (mode 1)
1	0	Outputs a low-level pulse during the final count down (mode 2)
1	1	Outputs a low-level pulse during each final count down (mode 3)

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{CC}	Supply voltage	With respect to V_{SS}	−0.5 ~ 7	V
V_I	Input voltage		−0.5 ~ 7	V
V_O	Output voltage		−0.5 ~ 7	V
Pd	Maximum power dissipation	Ta = 25°C	1.5	W
Topr	Operating free-air temperature range		0 ~ 70	°C
Tstg	Storage temperature range		−65 ~ 150	°C

RECOMMENDED OPERATING CONDITIONS (Ta = 0 ~ 70°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{CC}	Supply voltage	4.75	5	5.25	V
V_{SS}	Power-supply voltage		0		V
V_{IL}	Low-level input voltage	−0.5		0.8	V
V_{IH}	High-level input voltage	2		V_{CC}+0.5	V

ELECTRICAL CHARACTERISTICS (Ta = 0 ~ 70°C, V_{CC} = 5V ± 5%, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{OH}	High-level output voltage	V_{SS} = 0V, I_{OH} = −400μA	2.4			V
V_{OL}	Low-level output voltage	V_{SS} = 0V, I_{OL} = 2mA			0.45	V
I_I	Input leak current	V_{SS} = 0V, V_I = 0 ~ V_{CC}	−10		10	μA
$I_{I(CE)}$	Input leak current, CE pin	V_{SS} = 0V, V_I = 0 ~ V_{CC}	−100		100	μA
I_{OZ}	Output floating leak current	V_{SS} = 0V, V_I = 0.45 ~ V_{CC}	−10		10	μA
C_I	Input capacitance	V_{IL} = 0V, f = 1MHz, 25mVrms, Ta = 25°C			10	pF
$C_{I/O}$	Input/output terminal capacitance	$V_{I/OL}$ = 0V, f = 1MHz, 25mVrms, Ta = 25°C			20	pF
I_{CC}	Supply current from V_{CC}	V_{SS} = 0V			180	mA

Note 5 Current flowing into an IC is positive, out is negative.

2048-BIT STATIC RAM WITH I/O PORTS AND TIMER

TIMING REQUIREMENTS ($Ta = 0 \sim 70°C$, $V_{CC} = 5V \pm 5\%$, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{su(A-L)}$	Address setup time before latch	t_{AL}		50			ns
$t_{h(L-A)}$	Address hold time after latch	t_{LA}		80			ns
$t_{h(L-RWH)}$	Read/write hold time after latch	t_{LC}		100			ns
$t_{w(L)}$	Latch pulse width	t_{LL}		100			ns
$t_{h(RW-L)}$	Latch hold time after read/write	t_{CL}		20			ns
$t_{w(RWL)}$	Read/write low-level pulse width	t_{CC}		250			ns
$t_{su(D-W)}$	Data setup time before write	t_{DW}		150			ns
$t_{h(W-D)}$	Data hold time after write	t_{WD}		0			ns
$t_{w(RWH)}$	Read/write high-level pulse width	t_{RV}		300			ns
$t_{su(P-R)}$	Port setup time before read	t_{PR}		70			ns
$t_{h(R-P)}$	Port hold time after read	t_{RP}		50			ns
$t_{w(STB)}$	Strobe pulse width	t_{SS}		200			ns
$t_{su(P-STB)}$	Port setup time before strobe	t_{PSS}		50			ns
$t_{h(STB-P)}$	Port hold time after strobe	t_{PHS}		120			ns
$t_{w(\phi H)}$	Timer input high-level pulse width	t_2		120			ns
$t_{w(\phi L)}$	Timer input low-level pulse width	t_1		80			ns
$t_{c(\phi)}$	Timer input cycle time	t_{CYC}		320			ns
$t_{r(\phi)}$	Timer input rise time	t_r				30	ns
$t_{f(\phi)}$	Timer input fall time	t_f				30	ns

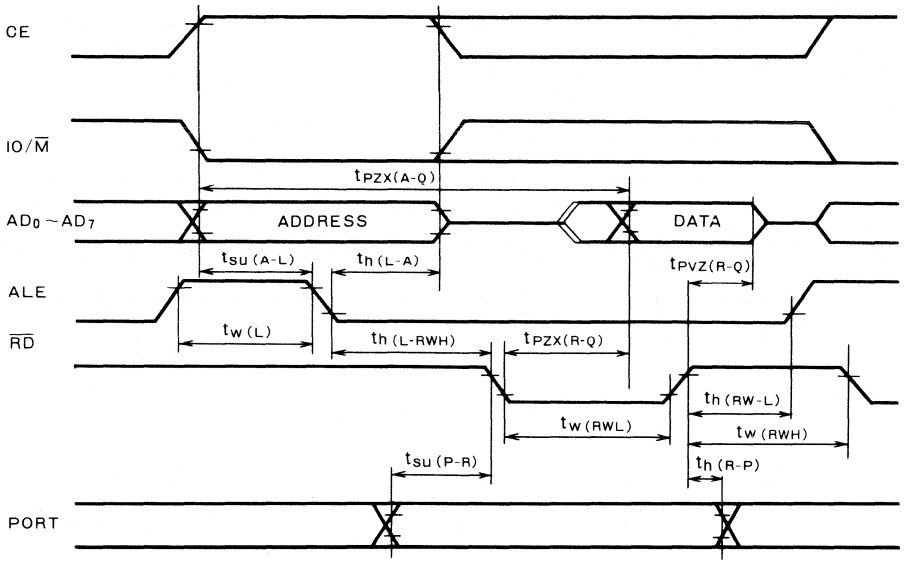
SWITCHING CHARACTERISTICS ($Ta = 0 \sim 70°C$, $V_{CC} = 5V \pm 5\%$, unless otherwise noted.)

Symbol	Parameter	Alternative symbol		Limits			Unit
				Min	Typ	Max	
$t_{PZX(R-Q)}$	Propagation time from read to data output	t_{RD}				170	ns
$t_{PZX(A-Q)}$	Propagation time from address to data output	t_{AD}				400	ns
$t_{PVZ(R-Q)}$	Propagation time from read to data floating (Note 7)	t_{RDF}				100	ns
$t_{PHL(W-P)}$	Propagation time from write to data output	t_{WP}				400	ns
$t_{PLH(W-P)}$		t_{WP}					
$t_{PLH(STB-BF)}$	Propagation time from strobe to BF flag	t_{SBF}				400	ns
$t_{PHL(R-BF)}$	Propagation time from read to BF flag	t_{RBE}				400	ns
$t_{PLH(STB-INTR)}$	Propagation time from strobe to interrupt	t_{SI}				400	ns
$t_{PHL(R-INTR)}$	Propagation time from read to interrupt	t_{RDI}				400	ns
$t_{PHL(STB-BF)}$	Propagation time from strobe to BF flag	t_{SBE}				400	ns
$t_{PLH(W-BF)}$	Propagation time from write to BF flag	t_{WBF}				400	ns
$t_{PHL(W-INTR)}$	Propagation time from write to interrupt	t_{WI}				400	ns
$t_{PHL(\phi-OUT)}$	Propagation time from timer input to timer output	t_{TL}				400	ns
$t_{PLH(\phi-OUT)}$		t_{TH}					

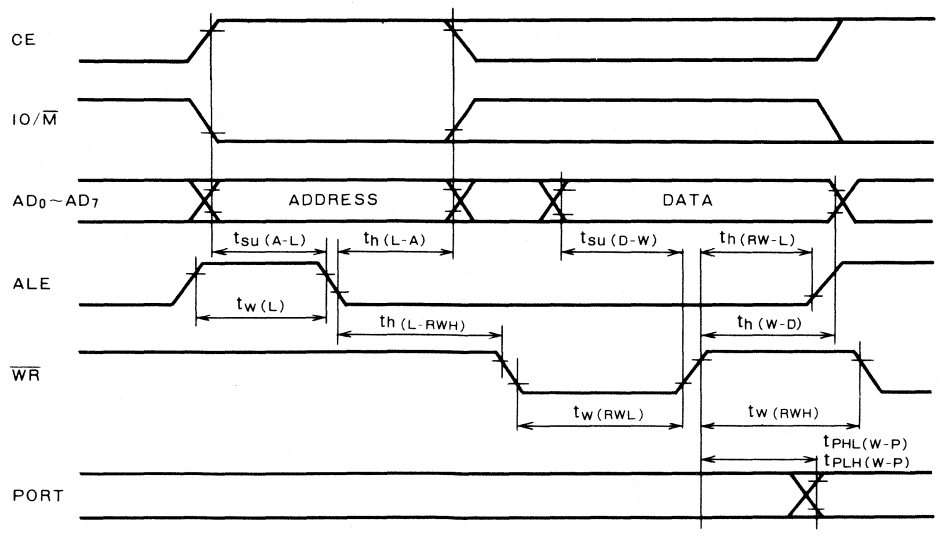
Note 6: Measurement conditions C = 150pF
7: Measurement conditions of note 6 are not applied.

TIMING DIAGRAM (reference level, high-level=2V, low-level=0.8V)

Basic Input



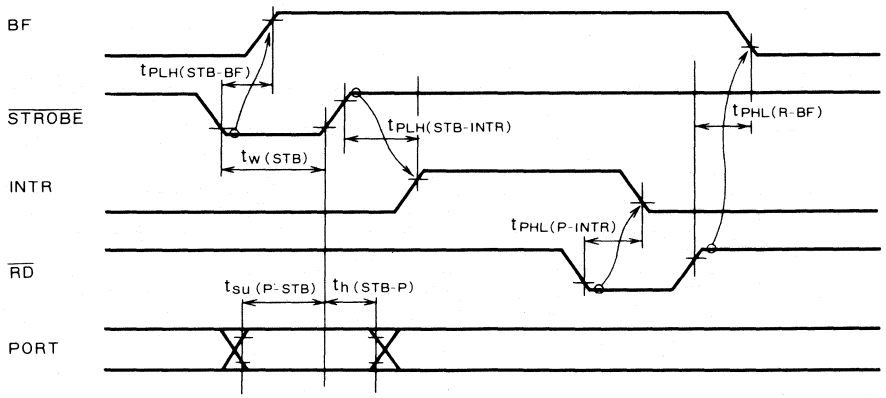
Basic Output



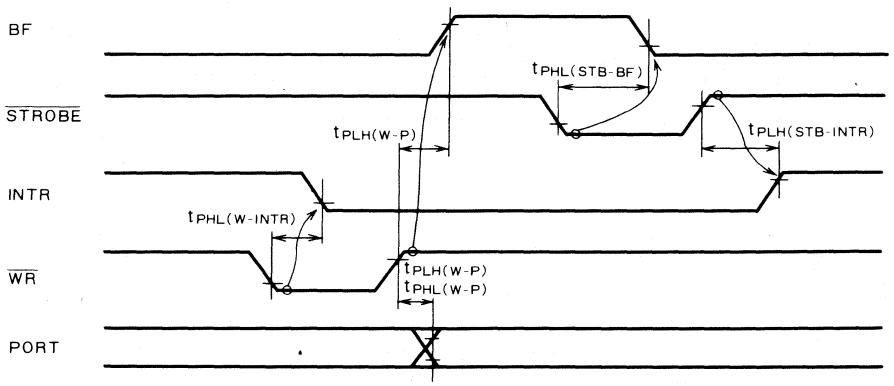
9

2048-BIT STATIC RAM WITH I/O PORTS AND TIMER

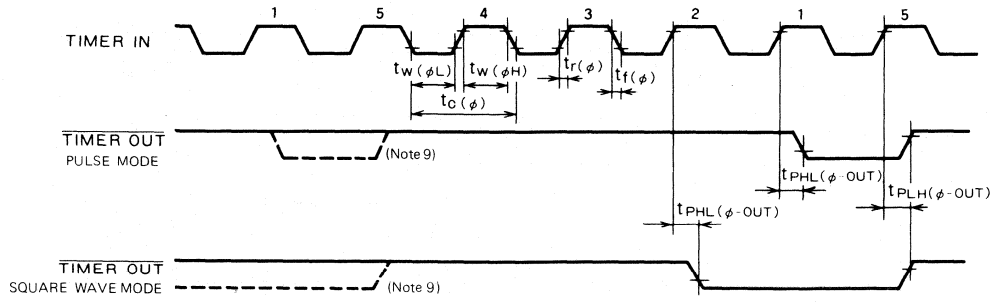
Strobed Input



Strobed Output



Timer (Note 8)



Note 8: The wave form is shown counting down from 5 to 1.
9: As long as the M1 mode flag of the timer register is at high-level, pulses are continuously output.

MITSUBISHI ELECTRIC

PROGRAMMABLE INTERRUPT CONTROLLER

DESCRIPTION

The M5L8259P is a programmable LSI for interrupt control. It is fabricated using N-channel silicon-gate ED-MOS technology and is designed to be used easily in connection with an M5L8080AP, M5L8085AP or M5L8086S.

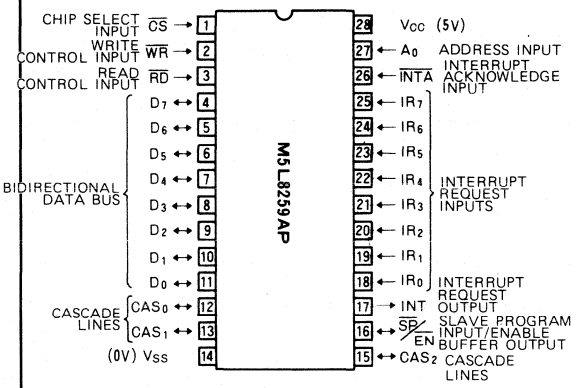
FEATURES

- Single 5V power supply
- CALL instruction to the CPU is generated automatically
- Priority, interrupt mask and vectored address for each interrupt request input are programmable
- Up to 64 levels of interrupt requests can be controlled by cascading with M5L8259AP
- Polling functions
- TTL compatible
- Interchangeable with Intels P8259A in pin configuration and electrical characteristics.

APPLICATIONS

- The M5L8259AP can be used as an interrupt controller for CPUs M5L8080AP, M5L8085AP and M5L8086S

PIN CONFIGURATION (TOP VIEW)



Outline 28P4

FUNCTIONS

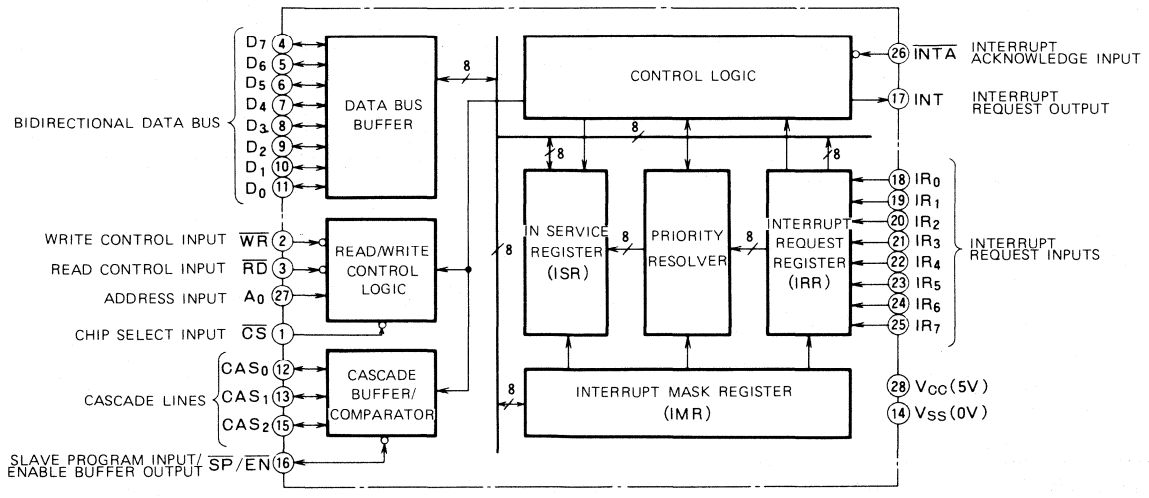
The M5L8259AP is a device specifically designed for use in real time, interrupt driven microcomputer systems. It manages eight levels or request and has built-in features for expandability to other M5L8259AP's.

The priority and interrupt mask can be changed or reconfigured at any time by the main program.

When an interrupt is generated because of an interrupt request at 1 of the pins, the M5L8259AP based on the mask and priority will output an INT to the CPU. After that, when an $\overline{INTA}$ signal is received from the CPU or the system controller, a CALL instruction and a programmed vector address is released onto the data bus.

9

BLOCK DIAGRAM



PROGRAMMABLE INTERRUPT CONTROLLER

PIN DESCRIPTION

Symbol	Pin name	Input or output	Functional significance
$\overline{CS}$	Chip select input	Input	This input is active at low-level, but may be at high-level during interrupt request input and interrupt processing.
$\overline{WR}$	Write control input	Input	Command write control input from the CPU
$\overline{RD}$	Read control input	Input	Data read control input for the CPU
$D_7 \sim D_0$	Bidirectional data bus	Input/ output	Data and commands are transmitted through this bidirectional data bus to and from the CPU.
$CAS_2 \sim CAS_0$	Cascade lines	Input/ output	These pins are outputs for a master and inputs for a slave. And these pins of the master will be able to address each individual slave. The master will enable the corresponding slave to release the device routine address during bytes 2 and 3 of $\overline{INTA}$.
$\overline{SP}/\overline{EN}$	Slave program input/ Enable buffer output	Input/ output	SP: In normal mode, a master is designated when $\overline{SP}/\overline{EN}$=1 and a slave is designated when $\overline{SP}/\overline{EN}$=0. EN: In the buffered mode, whenever the M5L8259AP's data bus output is enabled, its $\overline{SP}/\overline{EN}$ pin will go low.
INT	Interrupt request output	Output	This pin goes high whenever a valid interrupt is asserted.
$IR_7 \sim IR_0$	Interrupt request input	Input	The asynchronous interrupt inputs are active at high-level. The interrupt mask and priority of each interrupt input can be changed at any time. When using edge triggered mode, the rising edge (low to high) of the interrupt request and the high-level must be held until the first $\overline{INTA}$. For level triggered mode, the high-level must be held until the first $\overline{INTA}$.
$\overline{INTA}$	Interrupt acknowledge input	Input	When an interrupt acknowledge ($\overline{INTA}$) from the CPU is received, the M5L8259AP releases a CALL instruction or vectored address onto the data bus.
A_0	A_0 address input	Input	This pin is normally connected to one of the address lines and acts in conjunction with the $\overline{CS}$, $\overline{WR}$ and $\overline{RD}$ when writing commands or reading status registers.

OPERATION

The M5L8259AP is interfaced with a standard system bus as shown in Fig. 1 and operates as an interrupt controller.

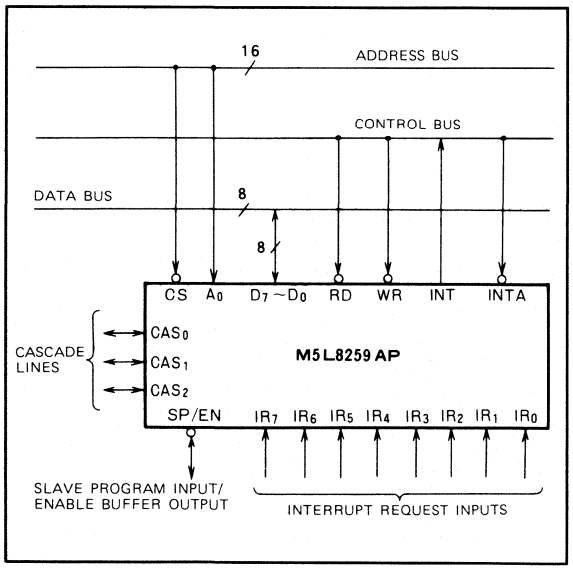


Fig. 1 The M5L8259AP interfaces to standard system bus.

Table 1 M5L8259AP basic operation

A_0	D_4	D_3	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Input operation (read)
0			0	1	0	IRR, ISR or interrupting level→data bus
1			0	1	0	IMR→Data bus
						Output operation (write)
0	0	0	1	0	0	Data bus→OCW2
0	0	1	1	0	0	Data bus→OCW3
0	1	X	1	0	0	Data bus →ICW1
1	X	X	1	0	0	Data bus→OCW1, ICW2, ICW3, ICW4
						Disable function
X	X	X	1	1	0	Data bus → High-impedance
X	X	X	X	X	1	Data bus → High-impedance

Interrupt Sequence

1. When the CPU is an M5L8080AP or M5L8085AP:
   (1) When one or more of the interrupt request inputs are raised high, the corresponding IRR bit(s) for the high-level inputs will be set.
   (2) Mask state and priority levels are considered and, if appropriate, the M5L8259AP sends an INT signal to the CPU.
   (3) The acknowledgement of the CPU to the INT signal, the CPU issues an $\overline{INTA}$ pulse to the M5L-8259AP.
   (4) The ISR bit corresponding to the interrupt request input is set upon receiving an $\overline{INTA}$ from the CPU, and the corresponding IRR bit is reset. A CALL instruction is released onto the data bus.
   (5) A CALL is a 3-byte instruction, so additional $\overline{INTA}$ pulses are issued to the M5L8259AP from the CPU.
   (6) These two $\overline{INTA}$ pulses allow the M5L8259AP to release the program address onto the data bus. The low-order 8-bit vectored address is released at the second $\overline{INTA}$ pulse and the high-order 8-bit vectored address is released at the third $\overline{INTA}$ pulse.
   (7) This completes the 3-byte CALL instruction and the interrupt routine will be serviced. The ISR bit is reset at the end of the third $\overline{INTA}$ pulse in the AEOI mode. In the other modes the ISR bit is not reset until an EOI command is issued.
2. When the CPU is an M5L8086S:
   (1) When one or more of the interrupt request inputs are raised high, the corresponding IRR bit(s) for the high-level inputs will be set.
   (2) Mask state and priority levels are considered and if appropriated, the M5L8259AP sends an INT signal to the CPU.
   (3) As an acknowledgement to the INT signal, the CPU issues an $\overline{INTA}$ pulse to the M5L8259AP.
   (4) The ISR bit corresponding to the interrupt request input is set upon receiving the first $\overline{INTA}$ pulse from the CPU, and the corresponding IRR bit is reset. The M5L8259AP does not drive the data bus, and the data bus goes to high-impedance state.
   (5) When the second $\overline{INTA}$ pulse is issued from the CPU an 8-bit pointer is released onto the data bus.
   (6) This completes the interrupt cycle and the interrupt routine will be serviced. The ISR bit is reset at the end of the second $\overline{INTA}$ pulse in the AEOI mode. In the other modes the ISR bit is not reset until an EOI command is issued from the CPU.

The interrupt request input must be held at high-level until the first $\overline{INTA}$ pulse is issued. If it is allowed to return to low-level before the first $\overline{INTA}$ pulse is issued, an interrupt request in IR_7 is executed. However, in this case the ISR bit is not set.

Interrupt sequence outputs

1. When the CPU is a M5L8080AP or M5L8085AP:
   A CALL instruction is released onto the data bus when the first $\overline{INTA}$ pulse is issued. The low-order 8 bits of the vectored address are released when the second $\overline{INTA}$ pulse is issued, and the high-order 8 bits are released when the third $\overline{INTA}$ pulse is issued. The format of these three outputs is shown in Table 2.

Table 2 Formats of interrupt CALL instruction and vectored address

First $\overline{INTA}$ pulse (CALL instruction)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	0	0	1	1	0	1

Second $\overline{INTA}$ pulse (low-order 8-bit of vectored address)

IR	Interval = 4							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
IR_7	A_7	A_6	A_5	1	1	1	0	0
IR_6	A_7	A_6	A_5	1	1	0	0	0
IR_5	A_7	A_6	A_5	1	0	1	0	0
IR_4	A_7	A_6	A_5	1	0	0	0	0
IR_3	A_7	A_6	A_5	0	1	1	0	0
IR_2	A_7	A_6	A_5	0	1	0	0	0
IR_1	A_7	A_6	A_5	0	0	1	0	0
IR_0	A_7	A_6	A_5	0	0	0	0	0

IR	Interval = 8							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
IR_7	A_7	A_6	1	1	1	0	0	0
IR_6	A_7	A_6	1	1	0	0	0	0
IR_5	A_7	A_6	1	0	1	0	0	0
IR_4	A_7	A_6	1	0	0	0	0	0
IR_3	A_7	A_6	0	1	1	0	0	0
IR_2	A_7	A_6	0	1	0	0	0	0
IR_1	A_7	A_6	0	0	1	0	0	0
IR_0	A_7	A_6	0	0	0	0	0	0

Third $\overline{INTA}$ pulse (high-order 8 bits of vectored address)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8

9

2. When the CPU is a M5L8086S:
   The data bus goes to a high-impedance state when the first $\overline{INTA}$ pulse is issued. Then the pointer $T_7 \sim T_0$ is released when the next $\overline{INTA}$ pulse is issued. The content of the pointer $T_7 \sim T_0$ is shown in Table 3. The $T_2 \sim T_0$ are a binary code corresponding to the interrupt request level, $A_{10} \sim A_5$ are unused and ADI mode control is ignored.

Table 3 Contents of interrupt pointer

Second $\overline{INTA}$ pulse (8-bit pointer)

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
IR_7	T_7	T_6	T_5	T_4	T_3	1	1	1
IR_6	T_7	T_6	T_5	T_4	T_3	1	1	0
IR_5	T_7	T_6	T_5	T_4	T_3	1	0	1
IR_4	T_7	T_6	T_5	T_4	T_3	1	0	0
IR_3	T_7	T_6	T_5	T_4	T_3	0	1	1
IR_2	T_7	T_6	T_5	T_4	T_3	0	1	0
IR_1	T_7	T_6	T_5	T_4	T_3	0	0	1
IR_0	T_7	T_6	T_5	T_4	T_3	0	0	0

Interrupt Request Register (IRR), In-service Register (ISR)

As interrupt requests are received at inputs $IR_7 \sim IR_0$, the corresponding bits of IRR are set and as an interrupt request is serviced the corresponding bit of ISR is set. The IRR is used to store all the interrupt levels which are requesting service, and the ISR is used to store all the interrupt levels which are being serviced. The status of these two registers can be read. These two registers are connected through the priority resolver.

An interrupt request received by IR_n is acknowledged on the leading edge when in the edge triggered mode or it is acknowledged on the level when in the level triggered mode. After that an INT signal is released and the interrupt request signal is latched in the corresponding IRR bit if the high-level is held until the first $\overline{INTA}$ pulse is issued. It is important to remember that the interrupt request signal must be held at high-level until the first $\overline{INTA}$ pulse is issued.

The interrupt request latching in the IRR causes a signal to be sent to the priority resolver unless it is masked out. When the priority resolver receives the signals it selects the highest priority interrupt request latched in IRR. The ISR is set when the first $\overline{INTA}$ pulse is issued while the corresponding bit of IRR is reset and the other bits of IRR are unaffected.

The bit of ISR that was set is not reset during the interrupt routine, but is reset at the end of the routine by the EOI command (end of interrupt) or by the last $\overline{INTA}$ pulse in AEOI mode.

Priority Resolver

The priority resolver examines all of the interrupt requests set in IRR to determine and selects the highest priority. The ISR bit corresponding to the selected (highest priority) request is set by the $\overline{INTA}$ pulse.

Interrupt Mask Register (IMR)

The contents of the interrupt mask register are used to mask out (disable) interrupt requests of selected interrupt request pins. Each terminal is independently masked so that masking a high priority interrupt does not influence the lower or higher priority interrupts. Therefore the contents of IMR selectively enable reading.

Interrupt Request Output (INT)

The interrupt request output connects directly to the interrupt input of the CPU. The output level is compatible with the input level required for the CPUs.

Interrupt Acknowledge Input ($\overline{INTA}$)

The CALL instruction and vectored address are released onto the data bus by the $\overline{INTA}$ pulse.

Data Bus Buffer

The data bus buffer is a 3-state bidirectional data bus buffer that is used to interface with the system bus. Write commands to the M5L8259AP, CALL instructions, vectored addresses, status information, etc. are transferred through the data bus buffer.

Read/Write Control Logic

The read/write control logic is used to control functions such as receiving commands from the CPU and supplying status information to the data bus.

Chip Select ($\overline{CS}$)

The M5L8259AP is selected (enabled) when $\overline{CS}$ is at low-level, but during interrupt request input or interrupt processing it may be high-level.

Write Control Input ($\overline{WR}$)

When $\overline{WR}$ goes to low-level the M5L8259AP can then write.

Read Control Input ($\overline{RD}$)

When $\overline{RD}$ goes low status information in the internal register of the M5L8259AP can be read through the data bus.

Address Input (A_0)

The address input is normally connected with one of the address lines and is used along with $\overline{WR}$ and $\overline{RD}$ to control write commands and reading status information.

Cascade Buffer/Comparator

The cascade buffer/comparator stores or compares identification codes. The three cascade lines are output when the M5L8259AP is a master or input when it is a slave. The identification code on the cascade lines select it as master or slave.

PROGRAMMING THE M5L8259AP

The M5L8259AP is programmed through the Initialization Command Word (ICW) and the operational command word (OCW). The following explains the functions of these two commands.

Initialization Command Words (ICSs)

The initialization command word is used for the initial setting of the M5L8259AP. There are 4 commands in this group and the following explains the details of these four commands.

ICW1

The meaning of the bits of ICW1 is explained in Fig. 3 along with the functions. ICW1 contains vectored address bits A_7~A_5, a flag indicating whether interrupt input is edge triggered or level triggered, CALL address interval, whether a single M5L8259AP or the cascade mode is used, and whether ICW4 is required or not.

Whenever a command is issued with A_0=0 and D_4=1, this is interpreted as ICW1 and the following will automatically occur.

(a) The interrupt mask register (IMR) is cleared.
(b) The interrupt request input IR_7 is assigned the lowest priority.
(c) The identification code for slave mode is set to 7.
(d) The special mask mode is cleared and the status read is set to the interrupt request register (IRR).
(e) When IC4=0 all bits in ICW4 are set to zero.

ICW2

ICW2 contains vectored address bits A_{15}~A_8 or interrupt type T_7~T_3, and the format is shown in Fig. 3.

ICW3

When SNGL=1 it indicates that only a single M5L8259AP is used in the system, in which case ICW3 is not valid. When SNGL=0, ICW3 is valid and indicates cascade connections with other M5L8259AP devices. In the master mode, a "1" is set for each slave.

When the CPU is an M5L8080AP or M5L8085AP the CALL instruction is released from the master at the first $\overline{INTA}$ pulse and the vectored address is released onto the data bus from the slave at the second and third $\overline{INTA}$ pulses.

When the CPU is a M5L8086S the master and slave are in high-impedance at the same time and the pointer is released onto the data bus from the slave at the next $\overline{INTA}$ pulse.

The master mode is specified when $\overline{SP}/\overline{EM}$ pin is high-level or BUF=1 and M/S=1 in ICW4, and slave mode is specified when $\overline{SP}/\overline{EM}$ pin is low-level or BUF=1 and M/S=0 in ICW4. In the slave mode, three bits ID_2~ID_0 identify the slave. And then when the slave code released on the cascade lines from the master, matches the assigned ID code, the vectored address is released by it onto the data bus at the next $\overline{INTA}$ pulse.

ICW4

Only when IC4=1 in ICW1 is ICW4 valid. Otherwise all bits are set to zero. When ICW4 is valid it specifies special fully nested mode, buffer mode master/slave, automatic EOI and microprocessor mode. The format of ICW4 is shown in Fig. 3.

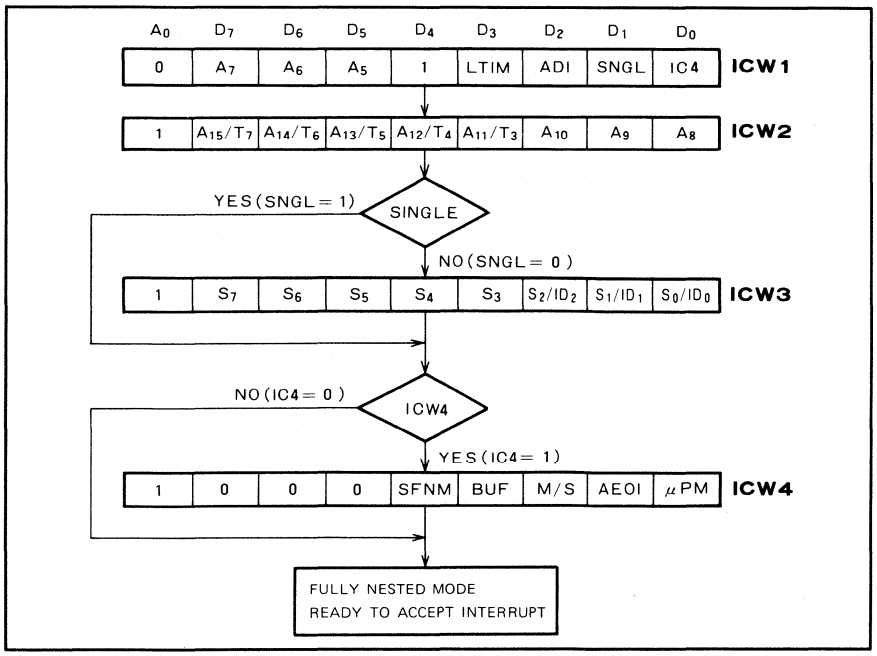


Fig. 2 Initialization sequence

9

PROGRAMMABLE INTERRUPT CONTROLLER

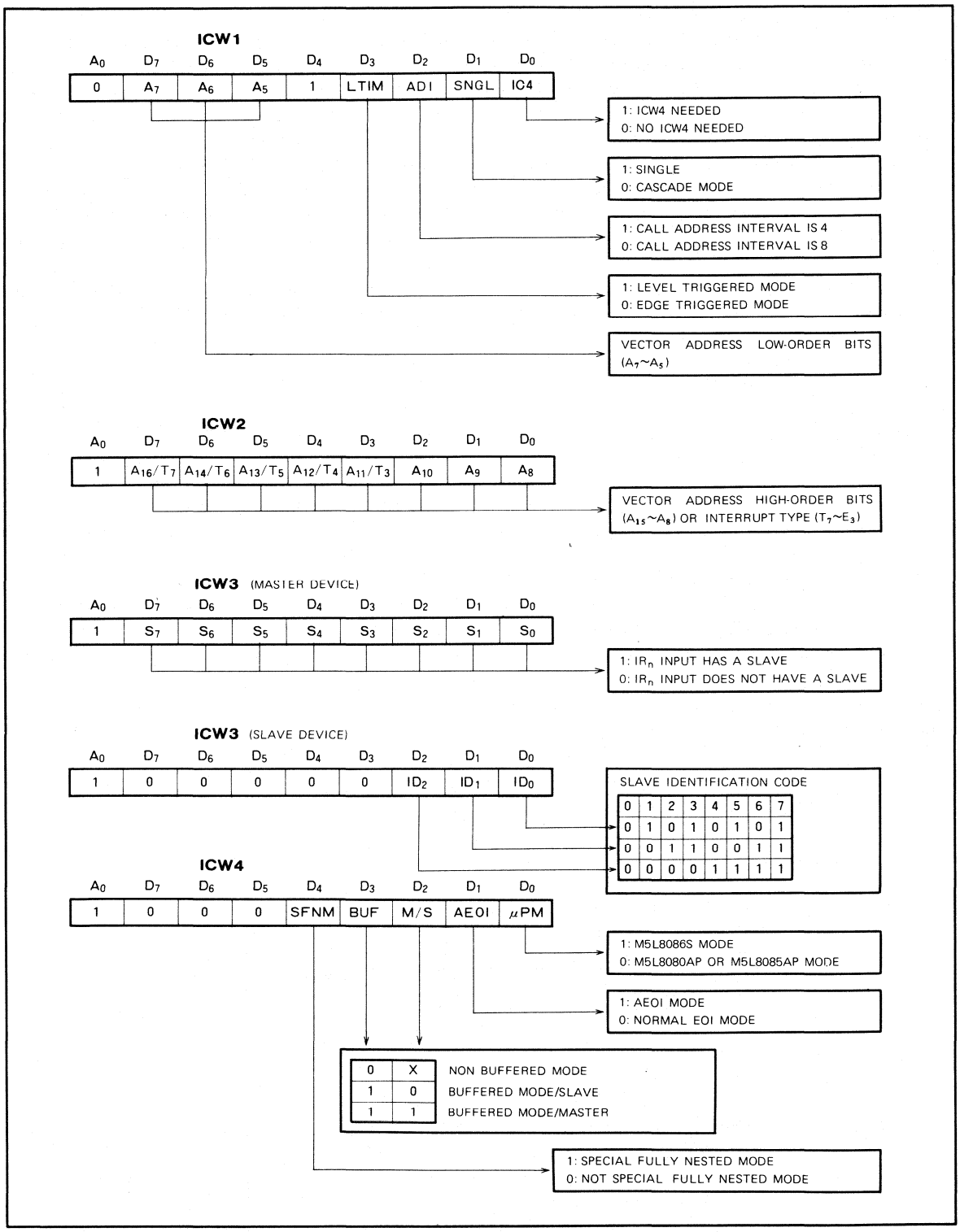


Fig. 3 Initialization command word format

Operation Command Words (OCWs)

The operation command words are used to change the contents of IMR, the priority of interrupt request inputs and the special mask. After the ICW are programmed into the M5L8259AP, the device is ready to accept interrupt requests. There are three types of OCWs; explanation of each follows, and the format of OCWs is shown in Fig. 4.

OCW1

The meaning of the bits of OCW1 are explained in Fig. 4 along with their functions. Each bit of IMR can be independently changed (set or reset) by OCW1.

OCW2

The OCW2 is used for issuing EOI commands to the M5L-8259AP and for changing the priority of the interrupt request inputs.

OCW3

The OCW3 is used for specifying special mask mode, poll mode and status register read.

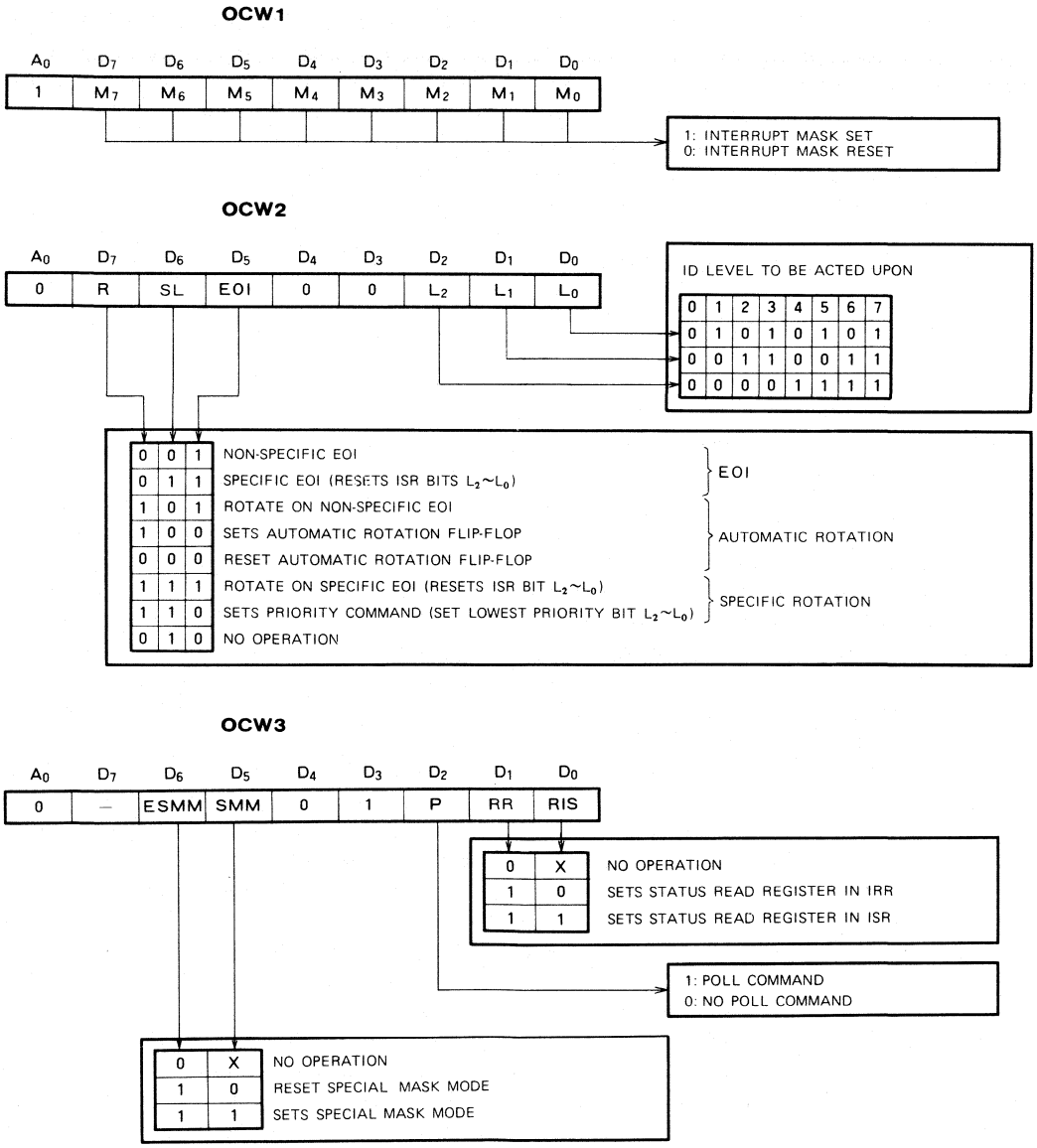


Fig. 4 Operation command word format

PROGRAMMABLE INTERRUPT CONTROLLER

FUNCTION OF COMMAND

Interrupt masks

The mask register contains a mask for each individual interrupt request. These interrupt masks can be changed by programming using OCW1.

Special mask mode

When an interrupt request is acknowledged and the ISR bit corresponding to the interrupt request is not reset by EOI command (which means an interrupt service routine is executing) lower priority interrupt requests are ignored.

In special mask mode interrupt requests received at interrupt request inputs which are masked by OCW1 are disabled, but interrupts at all levels that are not masked are possible. This means that in the mask mode all level of interrupts are possible or individual inputs can be selectively programmed so all interrupts at the selected inputs are disabled. The masks are stored in IMR and special mask is set/reset by executing OCW3.

Buffered mode

The buffered mode will structure the M5L8259AP to send an enable signal on $\overline{SP}/\overline{EN}$ to enable the data bus buffer, when the data bus requires the data bus buffer or when cascading mode is used. In this mode, when data bus output of the M5L8259AP is enabled, the $\overline{SP}/\overline{EN}$ output becomes active. This allows the M5L8259AP to be programmed whether it is a master or a slave by software. The buffered mode is set/reset by executing ICW4.

Fully nested mode

The fully nested mode is the mode when no mode is specified and is the usual operational mode. In this mode, the priority of interrupt request terminals is fixed from the lowest IR_7 to the highest IR_0. When an interrupt request is acknowledged the CALL instruction and vectored address are released onto the data bus. At the same time the ISR bit corresponding to the accepted interrupt request is set. This ISR bit remains set until it is reset by the input of an EOI command or until the trailing edge of last $\overline{INTA}$ pulse in AEOI mode. While an interrupt service routine is being executed, interrupt requests of same or lower priority are disabled while the bit of ISR remains set. The priorities can be changed by OCW2.

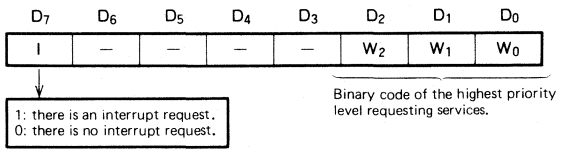
Special fully nested mode

The special fully nested mode will be used when cascading is used and this mode will be programmed to the master by ICW4. The special fully nested mode is the same as the fully nested mode with the following two exceptions.

1. When an interrupt from a certain slave is being serviced, this slave is not locked out from the master priority logic. Higher priority interrupts within the slave will be recognized by the master and the master will initiate an interrupt request to the CPU. In general in the normal fully nested mode, a serviced slave is locked out from the master's priority, and so higher priority interrupts from the same slave are not serviced.
2. When an interrupt from a certain slave is being serviced the software must check ISR to determine if there are additional interrupts requests to be serviced. If the ISR bit is 0 the EOI command may be sent to the master too. But if it is not 0 the EOI command should not be sent to the master.

Poll command

The poll mode is useful when the internal enable flip-flop of the microprocessor is reset, and interrupt input is disabled. Service to the device is achieved by a programmer initiative using a poll command. In the poll mode the M5L8259AP at the next $\overline{RD}$ pulse puts 8 bits on the data bus which indicates whether there is an interrupt request and reads the priority level. The format of the information on the data bus is as shown below.



When I=0 (no interrupt request), $W_2 \sim W_0$ is 111. The poll is valid from $\overline{WR}$ to $\overline{RD}$ and interrupt is frozen. This mode can be used for processing common service routines for interrupts from more than one line and does not require any $\overline{INTA}$ sequence. Poll command is issued by setting P=1 in OCW3.

End of interrupt (EOI) and specific EOI (SEOI)

An EOI command is required by the M5L8259AP to reset the ISR bit. So an EOI command must be issued to the M5L8259AP before returning from an interrupt service routine.

When AEOI is selected in ICW4, the ISR bit can be reset at the trailing edge of the last $\overline{INTA}$ pulse. When AEOI is not selected the ISR bit is reset by the EOI command issued to the M5L8259AP before returning from an interrupt service routine. When programmed in the cascade mode the EOI command must be issued to the master once and to corresponding slave once.

There are two forms of EOI command, specific EOI and non-specific EOI. When the M5L8259AP is used in the fully nested mode, the ISR bit being serviced is reset by the EOI command. When the non-specific EOI is issued the M5L8259AP will automatically reset the highest ISR bit of those that are set. Other ISR bits are reset by a specific EOI and the bit to be reset is specified in the EOI by the program. The SEOI is useful in modes other than free nested

mode. When the M5L8259AP is in special mask mode ISR bits masked in IMR are not reset by EOI. EOI and SEOI are selected when OCW2 is executed.

Automatic EOI (AEOI)

In the AEOI mode the M5L8259AP executes non-specific EOI command automatically at the trailing edge of the last $\overline{INTA}$ pulse. The AEOI mode is not required within a single M5L8259AP, but it is useful when a nested multilevel interrupt structure is expected. When AEOI=1 in ICW4, the M5L8259AP is put in AEOI mode continuously until reprogrammed in ICW4.

Automatic rotation

The automatic rotation mode is used in applications where many interrupt requests of the same level are expected such as multichannel communication systems. In this mode when an interrupt request is serviced, that request is assigned the lowest priority so that if there are other interrupt requests they will have higher priorities. This means that the next request on the interrupt request being serviced must wait until the other interrupt requests are serviced (worst case is waiting for all 7 of the other controllers to be serviced). The priority and serving status are rotated as shown in Fig. 5.

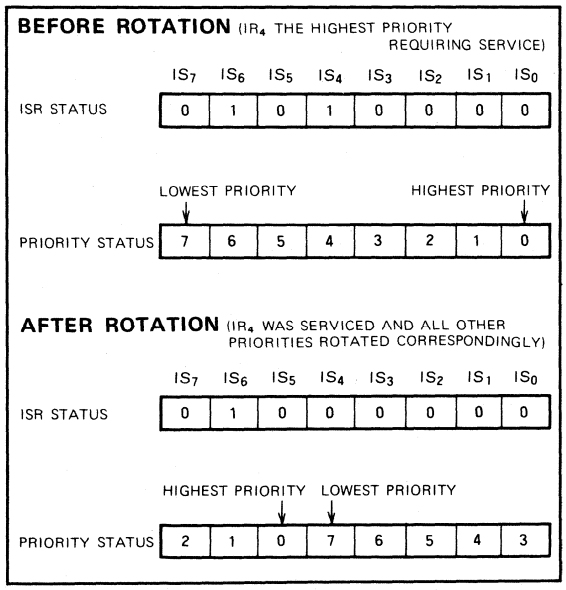


Fig. 5 An example of priority rotation

Automatic rotation mode is selected when R=1, EOI=1, SL=0 in OCW2. The internal priority status is changed by EOI or AEOI commands. The rotation priority A flip-flop is set by R=1, EOI=0 and SL=0 which is useful when the M5L8259AP is used in the AEOI mode.

Specific rotation

Specific rotation gives the user versatile capabilities in interrupt controlled operations. It serves in those applications in which a specific device's interrupt priority must be altered. As opposed to automatic rotation which automatically sets priorities, specific rotation is completely user controlled. That is, the user selects the interrupt level that is to receive lowest or highest priority. Priority changes can be executed during an EOI command.

Level triggered mode/Edge triggered mode

Selection of level or edge triggered mode of the M5L8259AP is made by ICW1. When using edge triggered mode not only is a transition from low to high required, but the high-level must be held until the first $\overline{INTA}$. If the high-level is not held until the first $\overline{INTA}$, the interrupt request will be treated as if it were input on IR_7, except that the ISR bit is not set. When level triggered mode is used the functions are the same as edge triggered mode except that the transition from low to high is not required to trigger the interrupt request.

In the level triggered mode and using AEOI mode together, if the high-level is held too long the interrupt will occur immediately. To avoid this situation interrupts should be kept disabled until the end of the service routine or until the IR input returns low. In the edge triggered mode this type of mistake is not possible because the interrupt request is edge triggered.

Reading the M5L8259AP internal status

The contents of IRR and ISR can be read by the CPU with status read. When an OCW3 is issued to the M5L8259AP and an $\overline{RD}$ pulse issued the contents of IRR or ISR can be released onto the data bus. A special command is not required to read the contents of IMR. The contents of IMR can be released onto the data bus by issuing an $\overline{RD}$ pulse when A_0=1. There is no need to issue a read register command every time the IRR or ISR is to be read. Once a read register command is received by the M5L8259AP, it remains valid until it is changed. Remember that the programmer must issue a poll command every time to check whether there is an interrupt request and read the priority level. Polling overrides status read when P=1, RR=1 in OCW3.

9

PROGRAMMABLE INTERRUPT CONTROLLER

Cascading

The M5L8259AP can be interconnected in a system of one master with up to eight slaves to handle up to 64 priority levels. A system of three units that can be used with the M5L8080AP or M5L8085AP is shown in Fig. 6.

The master can select a slave by outputting its identification code through the three cascade lines. The INT output of each slave is connected to the master interrupt request inputs. When an interrupt request of one of the slaves is to be serviced the master outputs the identification code of the slave through the cascade lines, so the slave will release the vectored address on the next $\overline{INTA}$ pulse.

The cascade lines of the master are nomally low, and will contain the slave identification code from the trailing edge of the first $\overline{INTA}$ pulse to the leading edge of the last $\overline{INTA}$ pulse. The master and slave can be programmed to work in different modes. ICWs, and EOI commands must be issued twice: once for the master and once for the corresponding slave. Each CS of the M5L8259AP requires an address decoder.

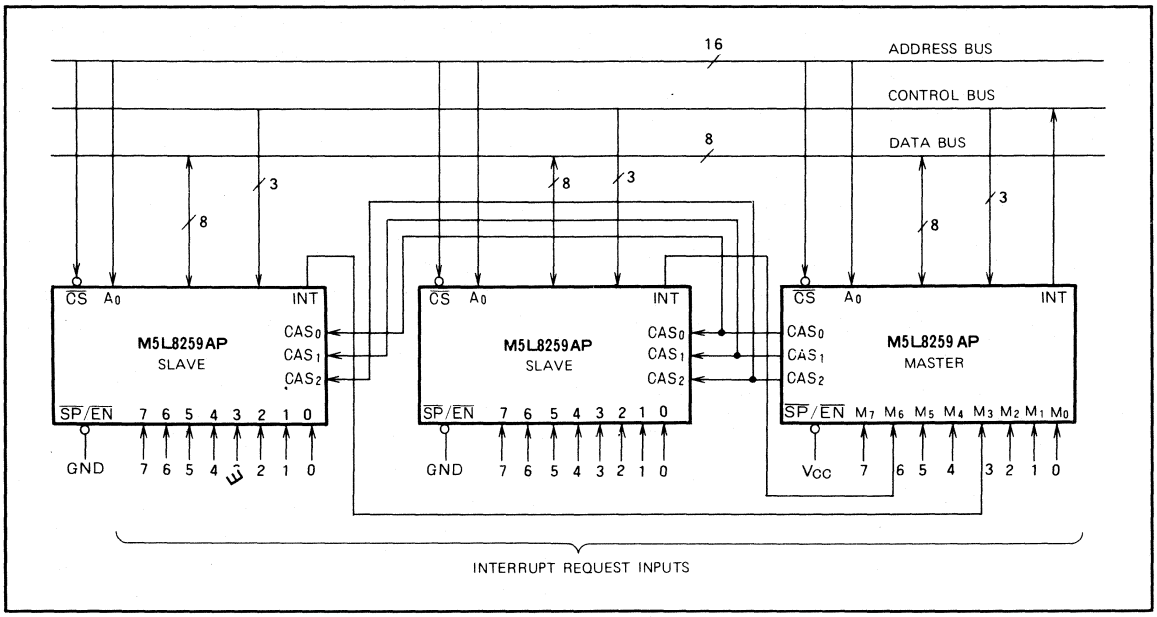


Fig. 6 Cascading the M5L8259AP

INSTRUCTION SET

Item / Number	Mnemonic	Instruction code									Function			
		A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	ICW4 required?	Interval	Single	Trigger
1	**ICW1 A**	0	A_7	A_6	A_5	1	0	1	1	0	N	4	Y	E
2	**ICW1 B**	0	A_7	A_6	A_5	1	1	1	1	0	N	4	Y	L
3	**ICW1 C**	0	A_7	A_6	A_5	1	0	1	0	0	N	4	N	E
4	**ICW1 D**	0	A_7	A_6	A_5	1	1	1	0	0	N	4	N	L
5	**ICW1 E**	0	A_7	A_6	0	1	0	0	1	0	N	8	Y	E
6	**ICW1 F**	0	A_7	A_6	0	1	1	0	1	0	N	8	Y	L
7	**ICW1 G**	0	A_7	A_6	0	1	0	0	0	0	N	8	N	E
8	**ICW1 H**	0	A_7	A_6	0	1	1	0	0	0	N	8	N	L
9	**ICW1 I**	0	A_7	A_6	A_5	1	0	1	1	1	Y	4	Y	E
10	**ICW1 J**	0	A_7	A_6	A_5	1	1	1	1	1	Y	4	Y	L
11	**ICW1 K**	0	A_7	A_6	A_5	1	0	1	0	1	Y	4	N	E
12	**ICW1 L**	0	A_7	A_6	A_5	1	1	1	0	1	Y	4	N	L
13	**ICW1 M**	0	A_7	A_6	0	1	0	0	1	1	Y	8	Y	E
14	**ICW1 N**	0	A_7	A_6	0	1	1	0	1	1	Y	8	Y	L
15	**ICW1 O**	0	A_7	A_6	0	1	0	0	0	1	Y	8	N	E
16	**ICW1 P**	0	A_7	A_6	0	1	1	0	0	1	Y	8	N	L
17	**ICW2**	1	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	8-bit vectored address			
18	**ICW3 M**	1	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0	Slave connections (master mode)			
19	**ICW3 S**	1	0	0	0	0	0	ID_2	ID_1	ID_0	Slave identification code (slave mode)			
											SFNM	BUF	AEOI	8086
20	**ICW4 A**	1	0	0	0	0	0	0	0	0	N	N	N	N
21	**ICW4 B**	1	0	0	0	0	0	0	0	1	N	N	N	Y
22	**ICW4 C**	1	0	0	0	0	0	0	1	0	N	N	Y	N
23	**ICW4 D**	1	0	0	0	0	0	0	1	1	N	N	Y	Y
24	**ICW4 E**	1	0	0	0	0	0	1	0	0	N	N	N	N
25	**ICW4 F**	1	0	0	0	0	0	1	0	1	N	N	N	Y
26	**ICW4 G**	1	0	0	0	0	0	1	1	0	N	N	Y	N
27	**ICW4 H**	1	0	0	0	0	0	1	1	1	N	N	Y	Y
28	**ICW4 I**	1	0	0	0	0	1	0	0	0	N	Y S	N	N
29	**ICW4 J**	1	0	0	0	0	1	0	0	1	N	Y S	N	Y
30	**ICW4 K**	1	0	0	0	0	1	0	1	0	N	Y S	Y	N
31	**ICW4 L**	1	0	0	0	0	1	0	1	1	N	Y S	Y	Y
32	**ICW4 M**	1	0	0	0	0	1	1	0	0	N	Y M	N	N
33	**ICW4 N**	1	0	0	0	0	1	1	0	1	N	Y M	N	Y
34	**ICW4 O**	1	0	0	0	0	1	1	1	0	N	Y M	Y	N
35	**ICW4 P**	1	0	0	0	0	1	1	1	1	N	Y M	Y	Y
36	**ICW4 NA**	1	0	0	0	1	0	0	0	0	Y	N	N	N
37	**ICW4 NB**	1	0	0	0	1	0	0	0	1	Y	N	N	Y
38	**ICW4 NC**	1	0	0	0	1	0	0	1	0	Y	N	Y	N
39	**ICW4 ND**	1	0	0	0	1	0	0	1	1	Y	N	Y	Y
40	**ICW4 NE**	1	0	0	0	1	0	1	0	0	Y	N	N	N
41	**ICW4 NF**	1	0	0	0	1	0	1	0	1	Y	N	N	Y
42	**ICW4 NG**	1	0	0	0	1	0	1	1	0	Y	N	Y	N
43	**ICW4 NH**	1	0	0	0	1	0	1	1	1	Y	N	Y	Y
44	**ICW4 NI**	1	0	0	0	1	1	0	0	0	Y	Y S	N	N
45	**ICW4 NJ**	1	0	0	0	1	1	0	0	1	Y	Y S	N	Y
46	**ICW4 NK**	1	0	0	0	1	1	0	1	0	Y	Y S	Y	N
47	**ICW4 NL**	1	0	0	0	1	1	0	1	1	Y	Y S	Y	Y
48	**ICW4 NM**	1	0	0	0	1	1	1	0	0	Y	Y M	N	N
49	**ICW4 NN**	1	0	0	0	1	1	1	0	1	Y	Y M	N	Y
50	**ICW4 NO**	1	0	0	0	1	1	1	1	0	Y	Y M	Y	N
51	**ICW4 NP**	1	0	0	0	1	1	1	1	1	Y	Y M	Y	Y
52	**OCW1**	1	M_7	M_6	M_5	M_4	M_3	M_2	M_1	M_0	Interrupt mask			
53	**OCW2 E**	0	0	0	1	0	0	0	0	0	EOI			
54	**OCW2 SE**	0	0	1	1	0	0	L_2	L_1	L_0	SEOI			
55	**OCW2 RE**	0	1	0	1	0	0	0	0	0	Rotate on Non-Specific EOI command (Automatic rotation)			
56	**OCW2 RSE**	0	1	1	1	0	0	L_2	L_1	L_0	Rotate on Specific EOI command (Specific rotation)			
57	**OCW2 R**	0	1	0	0	0	0	0	0	0	Rotate in AEOI Mode (SET)			
58	**OCW2 CR**	0	0	0	0	0	0	0	0	0	Rotate in AEOI Mode (CLEAR)			
59	**OCW2 RS**	0	1	1	0	0	0	L_2	L_1	L_0	Set priority without EOI			
60	**OCW3 P**	0	0	0	0	0	1	1	0	0				
61	**OCW3 RIS**	0	0	0	0	0	1	0	1	1				
62	**OCW3 RR**	0	0	0	0	0	1	0	1	0				
63	**OCW3 SM**	0	0	1	1	0	1	0	0	0				
64	**OCW3 RSM**	0	0	1	0	0	1	0	0	0				

Note: Y: yes, N: no, E: edge, L: level, M: master, S: slave

9

PROGRAMMABLE INTERRUPT CONTROLLER

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{CC}	Supply voltage		−0.5～7	V
V_I	Input voltage		−0.5～7	V
V_O	Output voltage		−0.5～7	V
Pd	Power dissipation		1000	mW
Topr	Operating free-air temperature range		0～70	°C
Tstg	Storage temperature range		−65～150	°C

RECOMMENDED OPERATING CONDITIONS (T_a=0～70°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{CC}	Supply voltage	4.5	5	5.5	V
V_{SS}	Supply voltage		0		V
V_{IH}	High-level input voltage	2		V_{CC}+0.5	V
V_{IL}	Low-level input voltage	−0.5		0.8	V

ELECTRICAL CHARACTERISTICS (T_a=0～70°C, V_{CC}, 5V±10%, V_{SS}=0V, unless otherwise noted.)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{OH}	High-level output voltage	I_{OH}=−400μA	2.4			V
$V_{OH(INT)}$	High-level output voltage, interrupt request output	I_{OH}=−100μA	3.5			V
V_{OL}	Low-level output voltage	I_{OL}=2.2mA			0.45	V
I_{CC}	Supply current from V_{CC}				85	mA
I_{IH}	High-level input current	V_I=V_{CC}	−10		10	μA
I_{IL}	Low-level input current	V_I=0V	−10		10	μA
I_{OZ}	Off-state output current	V_{SS}=0, V_I=0.45～5.5V	−10		10	μA
$I_{IH(IR)}$	High-level input current, interrupt request inputs	V_I=V_{CC}			10	μA
$I_{IL(IR)}$	Low-level input current, interrupt request inputs	V_I=0V	−300			μA
Ci	Output capacitance	V_{CC}=V_{SS}, f=1MHz, 25mVrms, Ta=25°C			10	pF
$C_{I/O}$	Input/output capacitance	V_{CC}=V_{SS}, f=1MHz, 25mVrms, Ta=25°C			20	pF

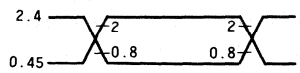
TIMING REQUIREMENTS (T_a=0～70°C, V_{CC}=5V±10%, V_{SS}=0V, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Limits			Unit
			Min	Typ	Max	
$t_{w(W)}$	Write pulse width	t_{WLWH}	290			ns
$t_{su(A-W)}$	Address setup time before write	t_{AHWL}	0			ns
$t_{h(W-A)}$	Address hold time after write	t_{WHAX}	0			ns
$t_{su(DQ-W)}$	Data setup time before write	t_{DVWH}	240			ns
$t_{h(W-DQ)}$	Data hold time after write	t_{WHDX}	0			ns
$t_{w(R)}$	Read pulse width	t_{RLRH}	235			ns
$t_{su(A-R)}$	Address setup time before read	t_{AHRL}	0			ns
$t_{h(R-A)}$	Address hold time after read	t_{RHAX}	0			ns
$t_{w(IR)}$	Interrupt request input width, low-level time, edge triggered mode	t_{JLJH}	100			ns
$t_{su(CAS-INTA)}$	Cascade setup time after $\overline{INTA}$ (slave)	t_{CVIAL}	55			ns
$t_{rec(W)}$	Write recovery time	t_{WHRL}	190			ns
$t_{rec(R)}$	Read recovery time	t_{RHRL}	160			ns

SWITCHING CHARACTERISTICS ($Ta=0\sim70°C$, $V_{CC}=5V\pm10\%$, $V_{SS}=0V$, unless otherwise noted)

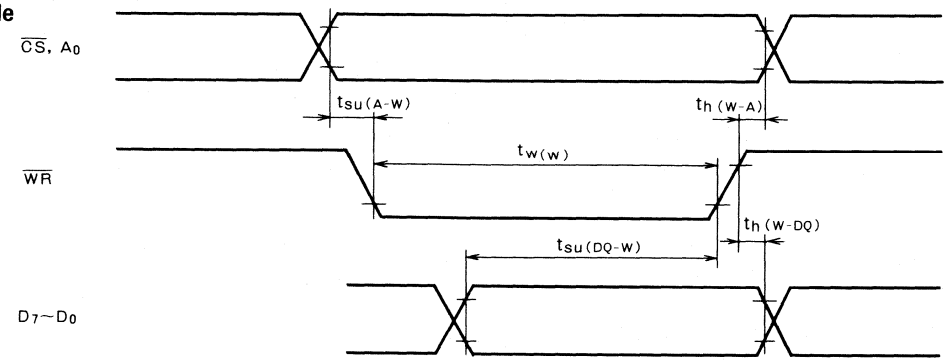
Symbol	Parameter	Alternative symbol	Limits			Unit
			Min	Typ	Max	
$t_{PZV(R-DQ)}$	Data output enable time after read	t_{RLDV}			200	ns
$t_{PVZ(R-DQ)}$	Data output disable time after read	t_{RHDZ}			100	ns
$t_{PZV(A-DQ)}$	Data output enable time after address	t_{AHDV}			200	ns
$t_{PHL(R-EN)}$	Propagation time from read to enable signal output	t_{RLEL}			125	ns
$t_{PLH(R-EN)}$	Propagation time from read to disable signal output	t_{RHEH}			150	ns
$t_{PLH(IR-INT)}$	Propagation time from interrupt request input to interrupt request output	t_{JHIH}			350	ns
$t_{PLV(INTA-CAS)}$	Propagation time from $\overline{INTA}$ to cascade output (master)	t_{IALCV}			565	ns
$t_{PZV(CAS-DQ)}$	Data output enable time after cascade output (slave)	t_{CVDV}			300	ns

Note 1: $\overline{INTA}$ signal is considered read signal
$\overline{CS}$ signal is considered address signal
Input pulse level 0.45~2.4V
Input pulse rise time 20ns
Input pulse fall time 20ns
Reference level input $V_{IH}=2V$, $V_{IL}=0.8V$
output $V_{OH}=2V$, $V_{OL}=0.8V$
Load capacitance $C_L=100pF$, where $\overline{SP}/\overline{EN}$ pin is 15pF

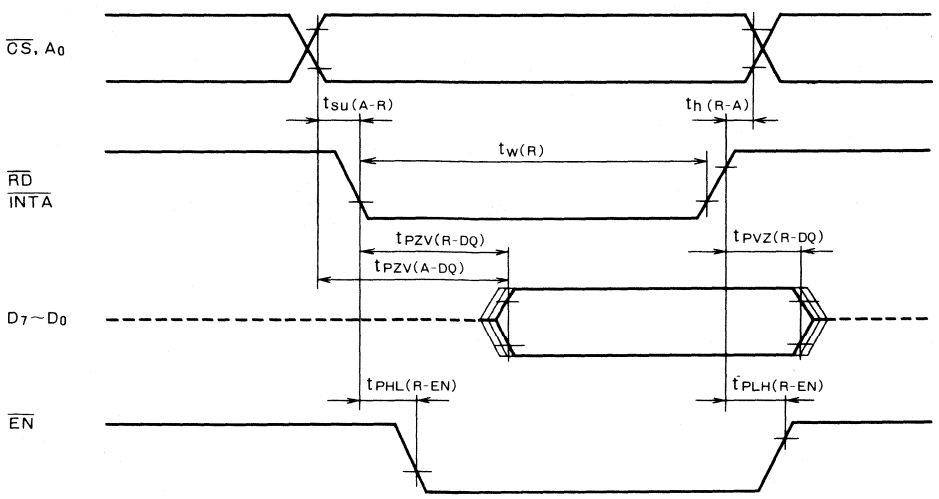


TIMING DIAGRAM

Write Mode



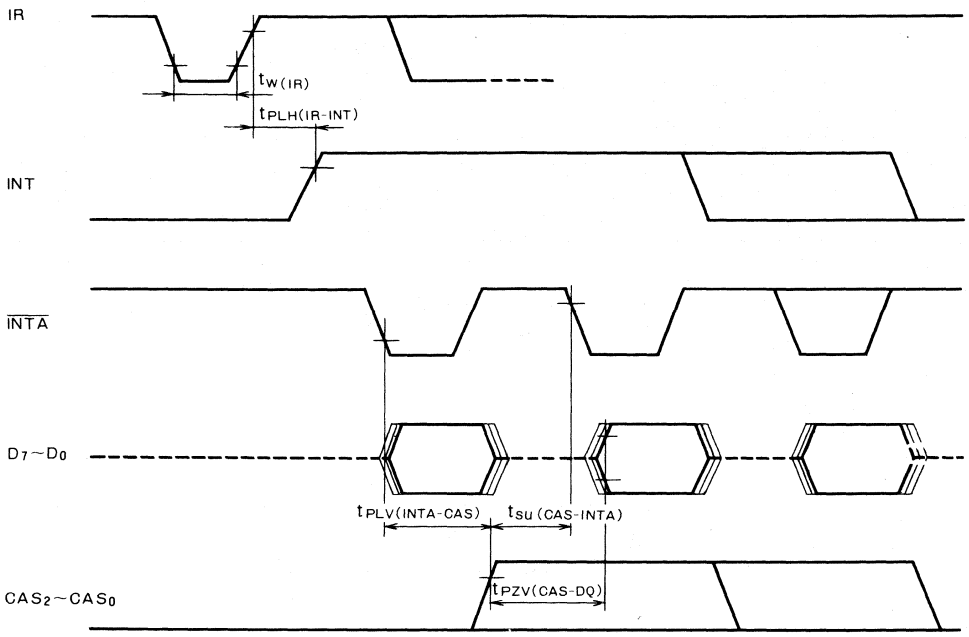
Read Mode



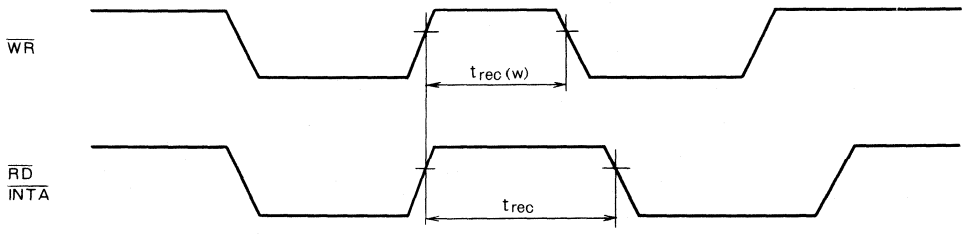
9

PROGRAMMABLE INTERRUPT CONTROLLER

Interrupt Sequence



Other Timing



Note 1: M5L8086S mode
2: M5L8080AP/M5L8085AP mode
3: M5L8086S mode is in high-impedance state, pointer is released during the next $\overline{INTA}$.
When in single M5L8080AP/M5L8085AP mode, data is released by all $\overline{INTA}$s. When master, CALL instruction is released during the first $\overline{INTA}$, high impedance state during the second and third $\overline{INTA}$. When slave, high impedance state during the first $\overline{INTA}$, vectored address is released during the second and third $\overline{INTA}$.

DESCRIPTION

The M5W1791-02P is a floppy disk formatter/controller device which accommodates single and double density formats. The device is designed for use with microprocessors or microcomputers. The device is fabricated with the N-channel silicon gate ED-MOS technology and is packaged in a 40-pin DIL package.

FEATURES

- Single 5V power supply
- Accommodate single and double density formats
  IBM 3740 single density format
  IBM system 34 double density format
- Selectable sector length (128,256,512 or 1024 bytes/sector)
- Side select compare
- Single/multiple sector read or write with automatic sector search
- Selectable track to track stepping time
- Write precompensation
- DMA or programmed data transfers
- Window extension
- Interchangeable with Western Digital's FD1791-02 in function except for V_{DD} power supply and pin configuration

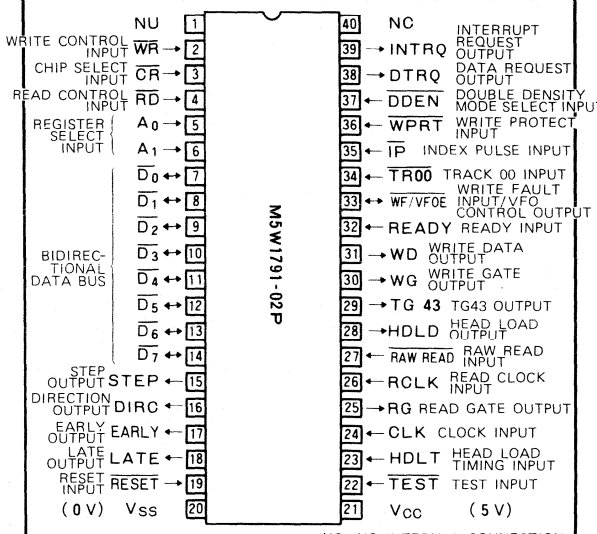
APPLICATIONS

- Single or double density floppy disk drive formatter/controller
- 8-inch or mini floppy disk interface

FUNCTION

The M5W1791-02P is a floppy disk formatter/controller that can be used with most microprocessor or microcomputer systems. The hardware of the M5W1791-02P consists of a floppy disk interface, a CPU interface and a PLA control logic. The total chip can be programmed by eleven 8-bit commands. The floppy disk interface portion performs the communication with the floppy disk drive under control of the PLA control logic. The CPU interface portion has five registers — command, dara, status, track and sector register — and communicates with the CPU through the data bus. These functions are also controlled by the PLA.

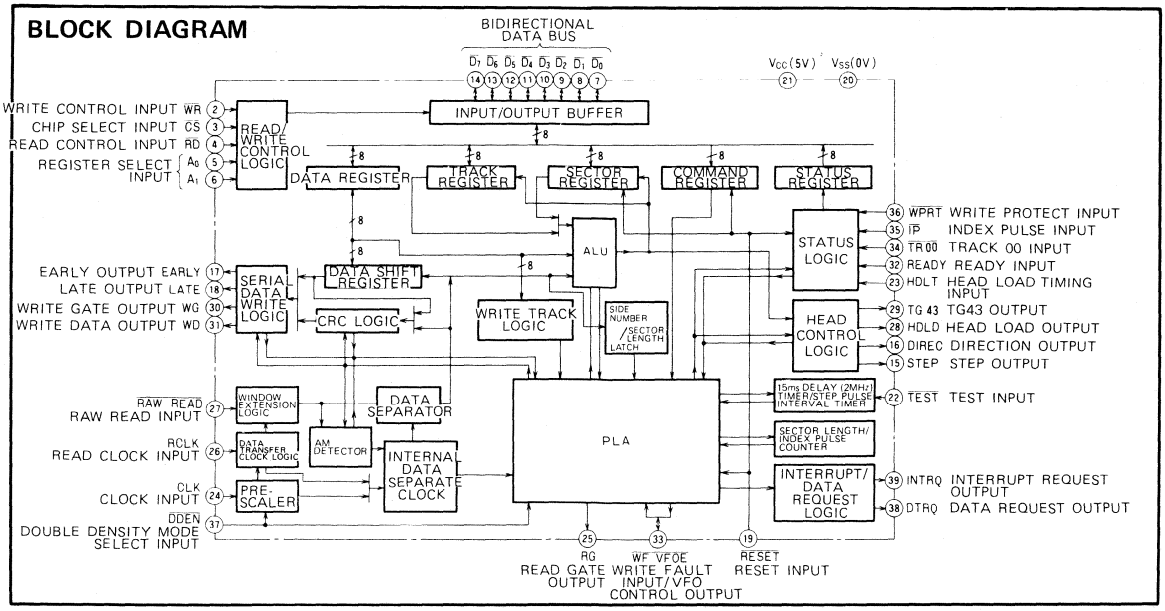
PIN CONFIGURATION (TOP VIEW)



NC: NO INTERNAL CONNECTION
NU: NON-USABLE

Outline 40P1

BLOCK DIAGRAM



FLOPPY DISK FORMATTER/CONTROLLER

PIN DESCRIPTION

Pin	Name	Input or output	Functions
NU	Non-usable terminal		NU (pin 1) is internally connected to the back gate bias generater, so it must remain open.
NC	No internal connection		NC (pin 40) is not internally connected.
$\overline{RESET}$	Reset input	Input	Reset input (Active low). The device is reset by this signal and automatically loads 0316 into the command register. The not-ready-status bit is also reset by this signal. When reset input is made to be high, the device executes restore command unless READY is active and the device loads 0116 to the sector register.
$\overline{WR}$	Write control input	Input	Write signal from a master CPU (Active low).
$\overline{CS}$	Chip select input	Input	Chip select (Active low).
$\overline{RD}$	Read control input	Input	Read signal from a master CPU (Active low).
A_0, A_1	Register select input	Input	Register select inputs. These inputs select the register under the control of the $\overline{RD}$ and $\overline{WR}$. A_1 A_0 $\overline{RD}$ $\overline{WR}$ 0 0 STATUS REGISTER COMMAND REGISTER 0 1 TRACK REGISTER TRACK REGISTER 1 0 SECTOR REGISTER SECTOR REGISTER 1 1 DATA REGISTER DATA REGISTER
$\overline{D_0}$–$\overline{D_7}$	Bidirectional data bus	In/Out	Three-state, inverted bidirectional data bus.
CLK	Clock input	Input	Clock input to generate internal timing, 2MHz for 8-inch drives, 1MHz for mini drives.
DTRQ	Data request output	Output	DTRQ is an open drain output, so pull up to V_{CC} by the 10k resistor. In the disk read mode, DTRQ indicates that data is assembled in the data register. In the disk write mode, it indicates that the data register is empty. DTRQ is reset by the read data or write data operation.
INTRQ	Interrupt request output	Output	INTRQ is also a open drain output, so pull up to V_{CC} by the 10k resistor. INTRQ becomes active at the completion of any command and is reset when the CPU reads the status or writes the command.
STEP	Step output	Output	Step pulse output (Active high).
DIRC	Direction output	Output	Direction output. High level means the head is stepping in and low level means the head is stepping out.
EARLY	Early output	Output	This signal is used for write precompensation. It indicates that the write data pulse should be shifted early.
LATE	Late output	Output	This signal is also used for write precompensation. It indicates that the write data pulse should be shifted late.
$\overline{TEST}$	Test input	Input	This input is only used for test purposes, so user must tie it to V_{CC} or leave it open unless using voice coil actuated motors.
HDLT	Head load timing input	Input	When the device finds high level on this input, the device assumes that the head is engaged on the media. Active high.
RG	Read gate output	Output	This signal shows the external data separator that the synchfield is detected.

Pin	Name	Input or output	Functions
RCLK	Read clock input	Input	This signal is internally used for the data window. Phasing relation to raw read data is specified but polarity (RCLK high or low) is not important.
$\overline{\text{RAW}}$ $\overline{\text{READ}}$	Raw read input	Input	This input signal from the drive shall be low for each recorded flux transition.
HDLD	Head load output	Output	This output signal controls the loading of the head of the drive. The head must be loaded on the media by this high-level output.
TG43	TG 43 output	Output	This output is valid only during disk read/write operation and it shows the position of the head. High level on this output indicates that the head is positioned between track 44 to 76.
WG	Write gate output	Output	This signal becomes active before disk write operations are to occur.
WD	Write data output	Output	This signal consists of data bits and clock bits. It becomes active for every flux transition. Active high.
READY	Ready input	Input	This signal shows the device the drive is ready. In the disk read/write operation except for TYPE 1 command operation, low-level input terminates current operation and the device generates the INTRQ. In the TYPE 1 command operation, this signal is neglected. Not ready bit in the status register is the inverted form of this input.
$\overline{\text{WF}}$/$\overline{\text{VFOE}}$	Write fault input/ VFO enable output	In/Out	This is a bidirectional signal. It becomes write fault input when WG is active. In the disk write operation, low level signal on this input terminates the write operation and makes INTRQ active. This signal also appears in the status register as the write fault bit. When WG is inactive, this signal works as VFO enable output. $\overline{\text{VFOE}}$ output is also an open drain type, so pull it up to V_{CC} and never input active write fault signal while WG is inactive.
$\overline{\text{TR00}}$	Track 00 input	Input	This signal indicates that the head is located on the track 00 to the device. Active low.
$\overline{\text{IP}}$	Index pulse input	Input	This input indicates to the device that an index hole of the diskette has been encountered.
$\overline{\text{WPRT}}$	Write protect input	Input	Low level signal on this input informs the device that the drive is in the write protected state. Before disk write operations, this signal is sampled and an active low signal will terminate the current command and set INTRQ. The write protect status bit in the status register is also set.
$\overline{\text{DDEN}}$	Double density mode select input	Input	This input determines the device operation mode. When DDEN=0, double density mode is selected. When DDEN=1, single density mode is selected.

9

FLOPPY DISK FORMATTER/CONTROLLER

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{CC}	Supply voltage	With respect to V_{SS}	−0.5∼7	V
V_I	Input voltage		−0.5∼7	V
V_O	Output voltage		−0.5∼7	V
Pd	Power dissipation	Ta=25°C	1000	mW
Topr	Operating free-air temperature range		0∼70	°C
Tstg	Storage temperature range		−65∼150	°C

RECOMMENDED OPERATING CONDITIONS (Ta=0∼70°C, unless otherwise noted)

Symbol	Paramter	Limits			Unit
		Min	Nom	Max	
V_{CC}	Supply voltage	4.75	5	5.25	V
V_{SS}	Supply voltage		0		V
V_{IH}	High-level input voltage	2			V
V_{IL}	Low-level input voltage	V_{SS}−0.5		0.8	V

ELECTRICAL CHARACTERISTICS (Ta=0∼70°C, V_{CC}=5V±5%, unless otherwise noted)

Symbol	Parameter	Test condition	Limits			Unit
			Min	Type	Max	
V_{OH}	High-level output voltage	I_{OH}=−100μA	2.4			V
V_{OL}	Low-level output voltage	I_{OL}=1.6mA			0.45	V
I_{CC}	Supply current				100	mA
I_I	Input current, HLT, $\overline{TEST}$, $\overline{WF}$, $\overline{WPRT}$ and $\overline{DDEN}$	V_I=V_{CC}∼0V	−100		100	μA
	Input current, other inputs	V_I=V_{CC}∼0V	−10		10	μA
I_{OZ}	Off-state output current	V_I=V_{CC}∼0V	−10		10	μA

TIMING REQUIREMENTS ($Ta = 0 \sim 70°C$, $V_{CC} = 5V \pm 5\%$, $V_{SS} = 0V$, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
t_{su} (A-R) t_{su} (CS-R)	Address setup time before read and chip select	TSET		50			ns
t_h (R-A) t_h (R-CS)	Address hold time after read and chip select	THLD		10			ns
t_w (R)	Read pulse width	TRE	$C_L = 50pf$	400			ns
t_{su} (A-W) t_{su} (CS-W)	Address setup time before write and chip select	TSET		50			ns
t_h (W-A) t_h (W-CS)	Address hold time after write and chip select	THLD		10			ns
t_w (W)	Write pulse width	TWE		350			ns
t_{su} (DQ-W)	Data setup time before write	TDS		250			ns
t_h (W-DQ)	Data hold time after write	TDH		70			ns
t_w (RR)	Raw read pulse width	Tpw	(Note 1, 2)	100	200		ns
t_c (RR)	Raw read cycle time	tbc	(Note 3)		1500	1800	ns
t_w (RCLK)	Read clock high-level width	Ta	(Note 4, 5)	800			ns
t_w (RCLK)	Read clock low-level width	Tb	(Note 4, 5)	800			ns
t_c (RCLK)	Read clock cycle time	Tc			1500	1800	ns
t_h (RCLK-RR)	Read clock hold time before raw read	T_{X1}		40			ns
t_h (RR-RCLK)	Read clock hold time after raw read	T_{X2}	(Note 1)	40			ns
t_w (WD)	Write data pulse width	Twp	FM	450	500	550	ns
			MFM	150	200	250	ns
t_c (WD)	Write data cycle time	Tbc			2, 3, 4		μs
t_w (ϕ)	Clock high-level pulse width	TCD_1		230	250	20000	ns
t_w (ϕ)	Clock low-level pulse width	TCD_2		200	250	20000	ns
t_w (RESET)	Reset pulse width	TMR		50			μs
t_w (IP)	Index pulse width	TIP	(Note 5)	10			μs
t_w (WF)	Write fault pulse width	TWF	(Note 5)	10			μs

SWITCHING CHARACTERISTICS ($Ta = 0 \sim 70°C$, $V_{CC} = 5V \pm 5\%$, $V_{SS} = 0V$, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
t_{PLH} (WG-WD)	Propagation time from write gate to write data	Twg	FM		2		μs
			MFM		1		μs
t_{PLH} (E-WD) t_{PLH} (L-WD)	Propagation time from early or late to write data	Ts	MFM	125			ns
t_{PHL} (WD-E) t_{PHL} (WD-L)	Propagation time from write data to early or late	Th	MFM	125			ns
t_{PHL} (WD-WG)	Propagation time from write data to write gate	Twf	FM		2		μs
			MFM		1		μs
t_{PZV} (R-DQ)	Output enable time after read	TDACC	$C_L = 50pf$			350	ns
t_{PVZ} (R-DQ)	Output disable time after read	TDOH	$C_L = 50pf$	50		150	ns
t_{PHL} (R-DRQ)	Propagation time from read to DRQ	TDRR(RD)			400	500	ns
t_{PHL} (R-INTRQ)	Propagation time from read to INTRQ	TIRR(RD)	(Note 5)		500	3000	ns
t_{PHL} (W-DRQ)	Propagation time from write to DRQ	TDRR(WR)			400	500	ns
t_{PHL} (W-INTRQ)	Propagation time from write to INTRQ	TIRR(WR)	(Note 5)		500	3000	ns
t_w (STP)	Step pulse width	TSTP	(Note 5)	2 or 4			μs
t_{PLH} (DIR-STP)	Propagation time from direction to step	TDIR	(Note 5)	12			μs

Note 1 : The pulse of Raw $\overline{READ}$ may be any width if pulse is entirely within RCLK. When the pulse occurs in the RCLK window, RAW READ pulse width must be less than 300 ns for MFM mode and 600 ns for FM mode at CLK=2MHz. Times double for 1MHz.

2 : 100 ns pulse width is recommended for the Raw $\overline{READ}$ pulse in 8 MFM mode.

3 : $\overline{RAW\ READ}$ cycle time $T_{C(RR)}$ and WD cycle time $T_{C(WD)}$ is normally $2\mu s$ in MFM and $4\mu s$ in FM. Times double when CLK=1MHz.

4 : The polarity of RCLK during Raw $\overline{READ}$ is not important.

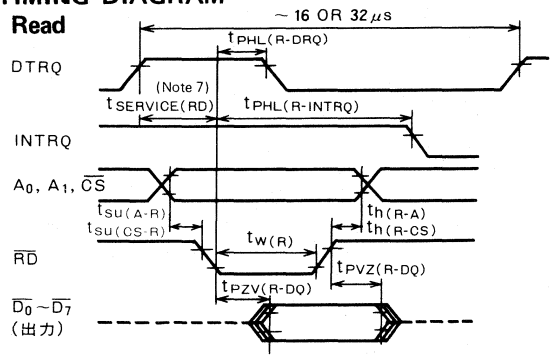
5 : Times double when CLK=1MHz.

9

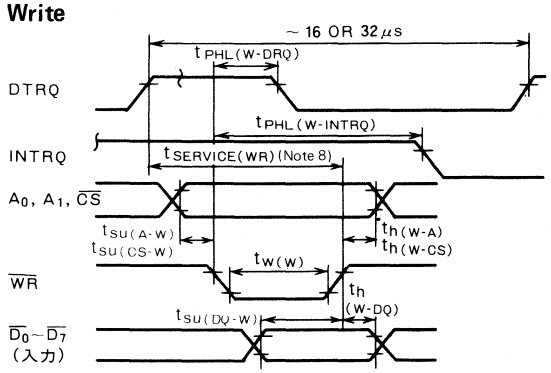
FLOPPY DISK FORMATTER/CONTROLLER

TIMING DIAGRAM

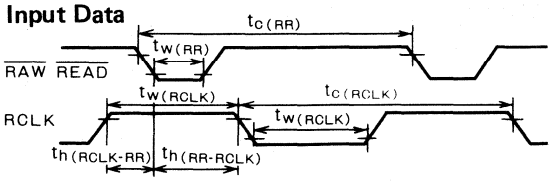
Read



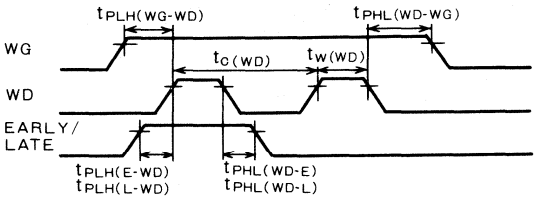
Write



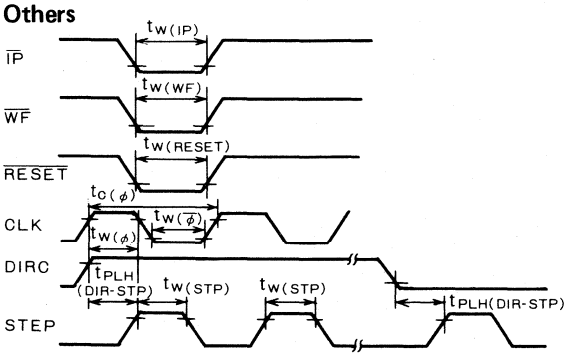
Input Data



Write Data



Others



Note 7: t SERVICE (RD) maximum value; FM: 27.5μs, MFM: 13.5μs
8: t SERVICE (WR) maximum value; FM: 28μs, MFM: 14μs

MITSUBISHI ELECTRIC

GENERAL-PURPOSE MOS LSIs

10

M50110XP, M50115XP

30- OR 120-FUNCTION REMOTE-CONTROL TRANSMITTERS

DESCRIPTION

The M50110XP and M50115XP are remote-control transmitter circuits manufactured by aluminum-gate CMOS technology for use in television receivers, audio equipment and other devices using infrared for transmission. The M50110-XP conveys 30 different commands on the basis of a 10-bit PCM code, while the M50115XP conveys 120 different commands. These transmitters are intended to be used in conjunction with an M50111XP, M50116XP or M50117-XP receiver. The X in each type corresponds to blank, A, B or C, which are respectively used for audio equipment, TV and VTR, air conditioners and other applications, or video-disk equipment.

FEATURES

Type	Remote-control function
M50110XP	30
M50115XP	120

- Single power supply
- Wide supply voltage range: 2.2V~8V
- Low power dissipation:
  Idle state (V_{DD}=3V): 3mW (typ)
  3μW (max)
- Has many functions and various uses
- Low-cost LC or ceramic oscillator used for reference frequency
- Low external component count
- Low transmitter duty cycle for minimal power consumption
- High-speed transmission

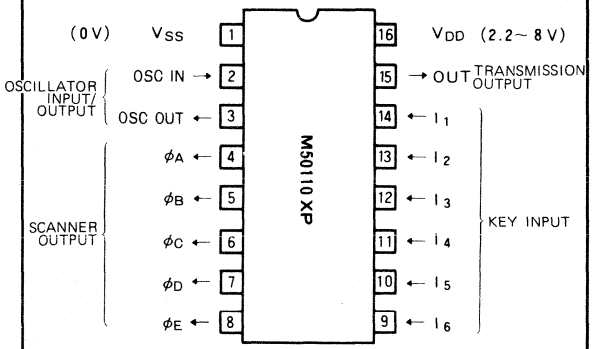
APPLICATION

- Remote-control transmitter for audio equipment, TV, VTR and video-disk equipment

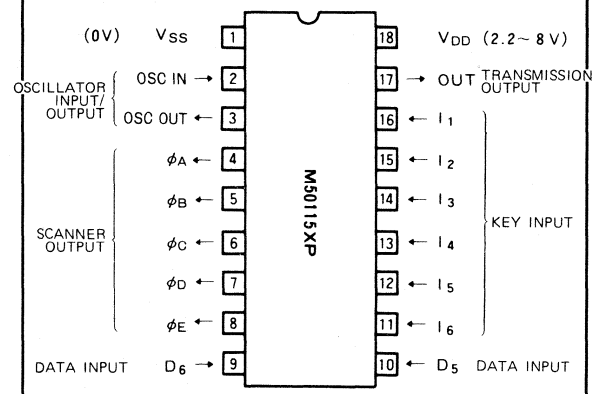
FUNCTION

The M50110XP and M50115XP transmitter circuits for infrared remote-control systems consist of an oscillator, a timing generator, a scanner, a key-in encoder, an instruction decoder, a code modulator and an output buffer. In M50110XP with a 6x5 keyboard matrix 30 commands can be transmitted by 10-bit PCM codes. In M50115XP, with a 6x5 keyboard matrix and two data inputs 120 commands can be transmitted. Oscillation is stopped when none of the keys are depressed in order to minimize power consumption.

PIN CONFIGURATION (TOP VIEW)



Outline 16P4



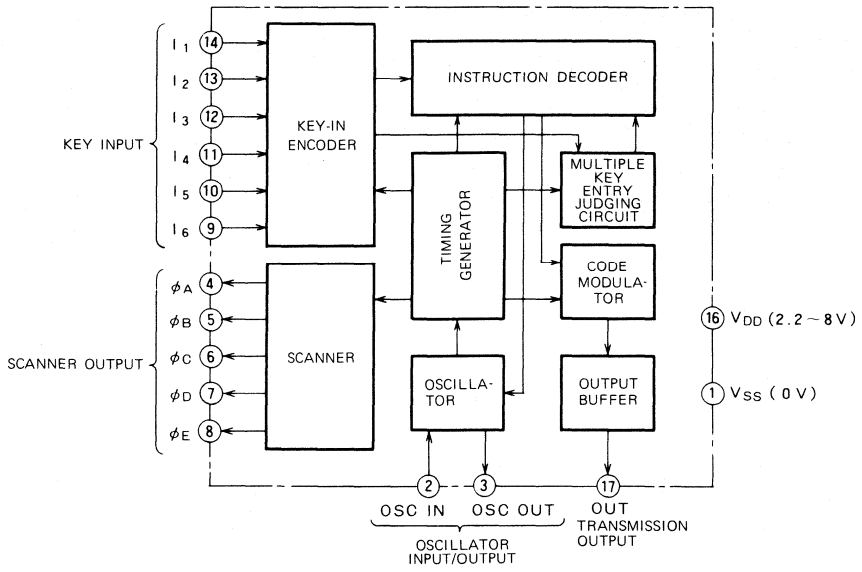
Outline 18P4

10

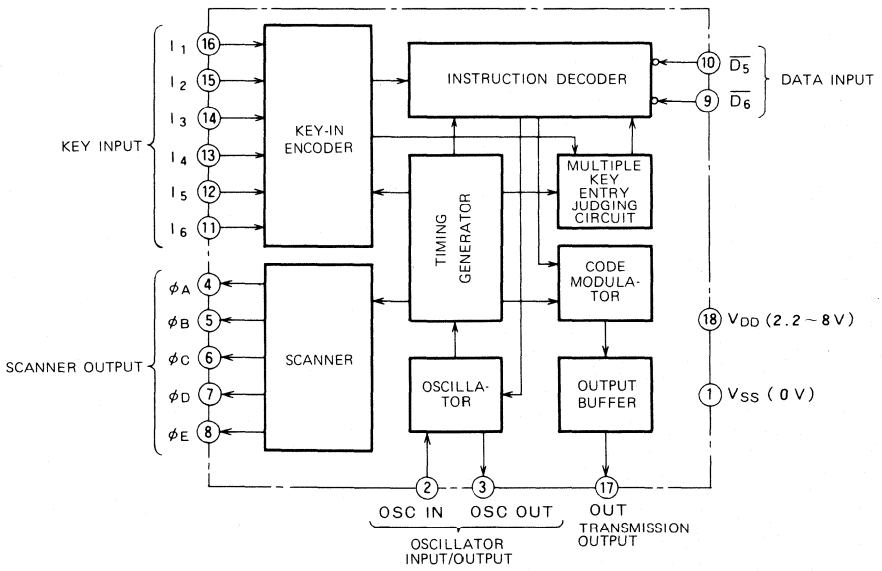
30- OR 120-FUNCTION REMOTE-CONTROL TRANSMITTERS

BLOCK DIAGRAM

M50110XP



M50115XP



MITSUBISHI ELECTRIC

FUNCTION

Oscillator

As the oscillator is on chip, oscillation frequency is easily obtained by connecting an external LC network or ceramic resonator between the OSC IN and OSC OUT terminals. Fig. 1 and Fig. 2 show examples of typical oscillation circuits.

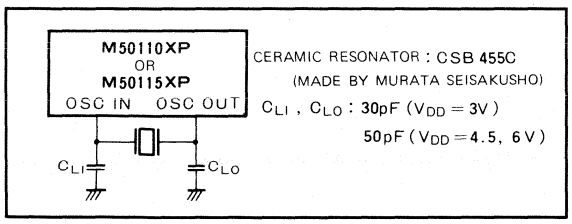


Fig. 1 An example of an oscillator (using a ceramic resonator)

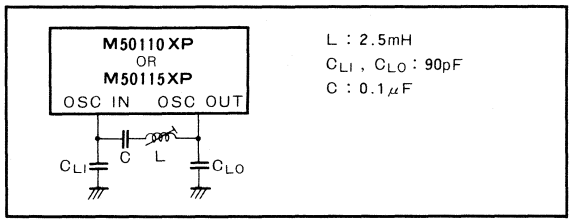


Fig. 2 An example of an oscillator (using an LC network)

Setting the oscillation frequency to 480kHz (or 455kHz) will also set the signal transmission carrier wave to 400kHz (or 38kHz).

Power consumption is minimized by stopping oscillation in the oscillator when none of the keys is depressed.

Key input and data input

In the M50110XP, 30 different commands can be sent through a 6x5 keyboard matrix, consisting of inputs I_1~I_6 and scanner outputs I_A~I_E. In the M50115XP, 120 different commands can be sent because two data inputs, $\overline{D_5}$ and $\overline{D_6}$, are also used.

Table 2 shows the relationship between the keyboard matrix and the transmission code.

1 Key code, type number and use

Key code			Type number	Use
K_0	K_1	K_2		
0	0	0	M50110P M50115P	Remote control for audio equipment
1	0	0	M50110AP M50115AP	Remote control for TV and VTR
0	1	0	M50110BP M50115BP	Remote control for air conditioners and other application
0	0	1	M50110CP M50115CP	Remote control for video-disk equipment

Table 2 Relation between the keyboard matrix and the transmission code names

	ϕA	ϕB	ϕC	ϕD	ϕE
I_6	A−1	A−2	A−3	A−4	A−5
I_5	A−6	A−7	A−8	A−9	A−10
I_4	A−11	A−12	A−13	A−14	A−15
I_3	B−0	B−1	B−2	B−3	B−4
I_2	B−5	B−6	B−7	B−8	B−9
I_1	B−10	B−11	B−12	B−13	B−14

Table 3 Relation between the transmission code names and the transmission codes

Transmission code name	Transmission				
	D_0	D_1	D_2	D_3	D_4
A−1	1	0	0	0	0
A−2	0	1	0	0	0
A−3	1	1	0	0	0
A−4	0	0	1	0	0
A−5	1	0	1	0	0
A−6	0	1	1	0	0
A−7	1	1	1	0	0
A−8	0	0	0	1	0
A−9	1	0	0	1	0
A−10	0	1	0	1	0
A−11	1	1	0	1	0
A−12	0	0	1	1	0
A−13	1	0	1	1	0
A−14	0	1	1	1	0
A−15	1	1	1	1	0
B−0	0	0	0	0	1
B−1	1	0	0	0	1
B−2	0	1	0	0	1
B−3	1	1	0	0	1
B−4	0	0	1	0	1
B−5	1	0	1	0	1
B−6	0	1	1	0	1
B−7	1	1	1	0	1
B−8	0	0	0	1	1
B−9	1	0	0	1	1
B−10	0	1	0	1	1
B−11	1	1	0	1	1
B−12	0	0	1	1	1
B−13	1	0	1	1	1
B−14	0	1	1	1	1

10

30- OR 120-FUNCTION REMOTE-CONTROL TRANSMITTERS

Transmission Commands

In the M50110XP, 30 commands can be transmitted by 10-bit PCM codes (K_0~K_2, D_0~D_6), and in the M50115XP, 120 commands can be transmitted. The first three bits K_0~K_2, which are key codes between transmitters and receivers, correspond to type numbers and uses. Relation between key codes, type numbers and uses of remote control systems is shown in Table 1.

The next five bits D_0~D_4 correspond to the 6x5 keyboard matrix. Relation between transmission codes and their name is shown in Table 2.

The last two bits, D_5 and D_6, are controlled by the data inputs D_5 and D_6. When terminal D_5 or D_6 is open or high level, data code D_5 or D_6 becomes "0", and when terminal D_5 or D_6 is low level, code data D_5 or D_6 becomes "1".

In the M50110XP, the data bits D_5 and D_6 are fixed in "0." To prevent spurious operation, the codes are designed so that there is no transmission code whose data bits D_0~D_6 are all "0" or "1."

Transmission Coding

When oscillation frequency f_{osc} is 480kHz, transmission of data code is executed as follows: when f_{osc} is other than 480kHz, the period is multiplied by 480kHz/f_{osc} and its frequency by f_{osc}/480kHz.

A single pulse is amplitude-modulated by a carrier of 40kHz, and the pulse width is 0.25ms. Therefore a single pulse consists of 10 clock pulses of 40kHz (see Fig. 3).

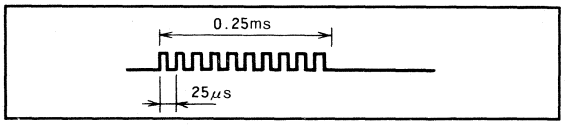


Fig. 3 A single pulse modulated onto carrier (40kHz)

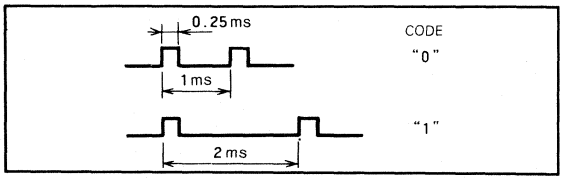


Fig. 4 Distinction between the bits "1" and "0"

The distinction between "0" and "1" bits is made by the pulse interval between two pulses, with an 1ms interval corresponding to "0", and a 2ms interval representing "1" (see Fig. 4).

One command word is composed of 10 bits, that is, of 11 pulses, and it is transmitted in the 24ms cycle while a matrix switch is depressed (see Fig. 5).

As mentioned above, adopting of this code means that the period during which output is high level (i.e. signal emitting LED is lit) is shorter than in continuous wave transmission. Indeed the LED is on for only half of the 11-pulse period or 1.375ms, which is 5.7% of the 24ms entire cycle. This not only saves in total power consumption, but it also improves LED reliability. That is to say, emission can be increased on the same power consumption.

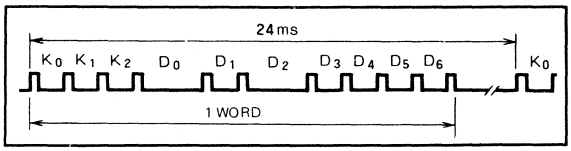


Fig. 5 Synthesis of one word (the code below shows 0001010000)

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{DD}	Supply voltage	With respect to GND	−0.3～9	V
V_I	Input voltage		$V_{SS} \leqq V_I \leqq V_{DD}$	V
V_O	Output voltage		$V_{SS} \leqq V_O \leqq V_{DD}$	V
Pd	Maximum power dissipation	Ta＝25°C	300	mW
Topr	Operating free-air temperature range		−30～70	°C
Tstg	Storage temperature range		−40～125	°C

RECOMMENDED OPERATING CONDITIONS (Ta＝−30～70°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{DD}	Supply voltage	2.2		8	V
V_{IH}	High-level input voltage	$0.7 \times V_{DD}$		V_{DD}	V
V_{IL}	Low-level input voltage	0		$0.3 \times V_{DD}$	V
f_{OSC}	Oscillation frequency		455		kHz
			480		kHz

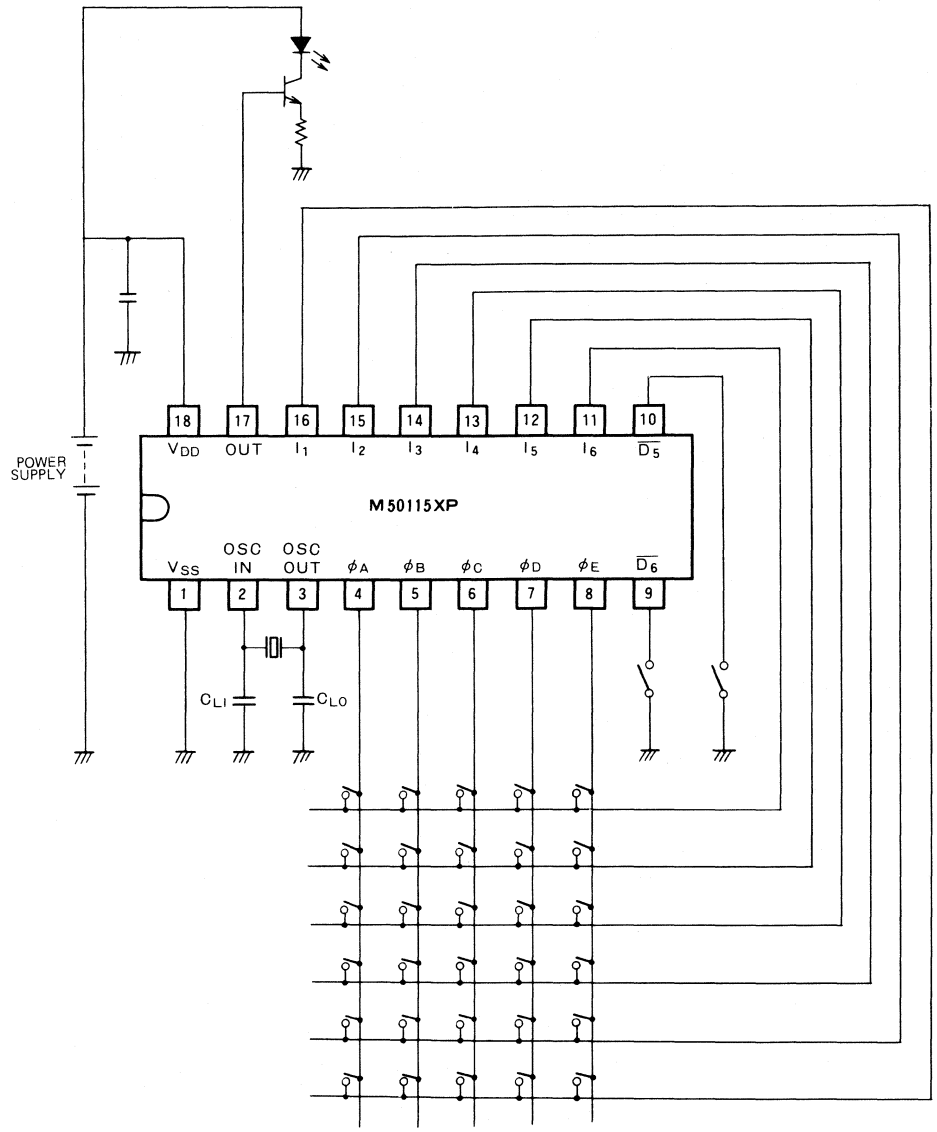
ELECTRICAL CHARACTERISTICS (Ta＝25°C, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits			Unit
				Min	Typ	Max	
V_{DD}	Operational supply voltage	Ta＝−30～70°C, f_{OSC}＝455kHz		2.2		8	V
I_{DD}	Supply voltage during operation	f_{OSC}＝455kHz	V_{DD}＝3 V		0.1	0.5	mA
			V_{DD}＝6 V		0.5	2	mA
I_{DD}	Supply voltage during non-operation	V_{DD}＝3 V				1	μA
		V_{DD}＝8 V				5	μA
R_I	Pull-up resistances, I_1～I_6				20		kΩ
I_{OL}	Low-level output currents, ϕ_A～ϕ_E	V_{DD}＝3 V, V_{OL}＝0.9V		0.18	0.6		mA
		V_{DD}＝6 V, V_{OL}＝1.8V		0.7	3		mA
I_{OH}	High-level output current, OUT	V_{DD}＝3 V, V_{OH}＝2 V		−2	−5		mA
		V_{DD}＝6 V, V_{OH}＝4 V		−8	−16		mA

10

30- OR 120-FUNCTION REMOTE-CONTROL TRANSMITTERS

An example of an application circuit (M50115XP)



MITSUBISHI ELECTRIC

M50111XP, M50116XP, M50117XP

30~120-FUNCTION REMOTE-CONTROL RECEIVER

DESCRIPTION

The M50111XP, M50116XP and M50117XP are remote control receiver circuits manufactured by aluminum-gate CMOS technology for use in television receivers, audio equipment and other applications using infrared transmission. The systems can receive 30~120 different 10-bit PCM code commands by remote control.

The M50111XP, M50116XP and M50117XP are designed for use with an M50110XP or M50115XP transmitter. The X in each type number corresponds to blank, A, B or C, which are respectively used for audio equipment, TV and VTR, air conditioner and other applications, or video-disk equipment.

FEATURES

Type	Remote-control function		Parallel outputs
	Serial data	Parallel data	
M50111XP	120	30	D_0~D_3, $\overline{STA}$, $\overline{STB}$
M50116XP	120	60	D_0~D_3, $\overline{STA}$~$\overline{STD}$
M50117XP	120	120	D_0~D_7, FF

- Single power supply
- Wide power supply voltage range: 4.5V~8V
- Low power dissipation
- Low-cost LC or ceramic oscillator used for frequency reference
- Information is transmitted by pulse code modulation
- High speed reception
- Superior noise immunity — instructions are not executed unless the same code is received two or more times in succession
- Single transmission frequency (40kHz or 38kHz) for carrier wave
- Many functions and various uses
- Large tolerance in operating frequency between the transmitter and the receiver
- Can be simply connected to a microcomputer

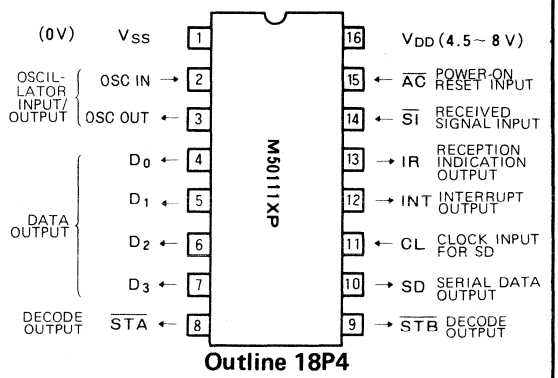
APPLICATION

- Remote control receivers for audio equipment, TV, VTR, air conditioners, video-disk equipment and similar devices

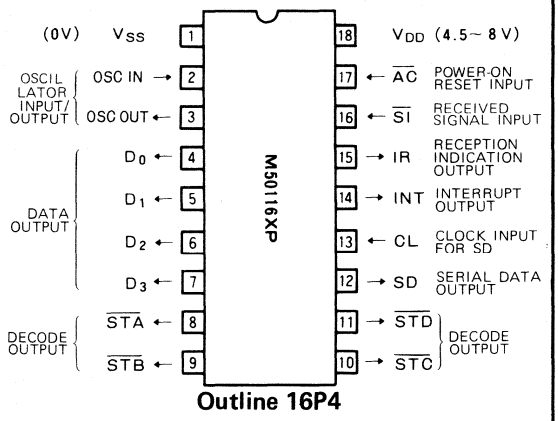
FUNCTION

The M50111XP, M50116XP and M50117XP receivers for infrared remote control systems consist of an oscillator, a timing generator, a demodulator, an error prevention circuit, a reception state decision circuit, a serial data processor, a shift register, a received signal input circuit, power-on reset circuit and other circuits. The M50111XP, M50116XP and M50117XP are designed to decode and execute instructions after 2 successive receptions of the identical instruction code. This provides positive assurance that noise will not be executed as instructions.

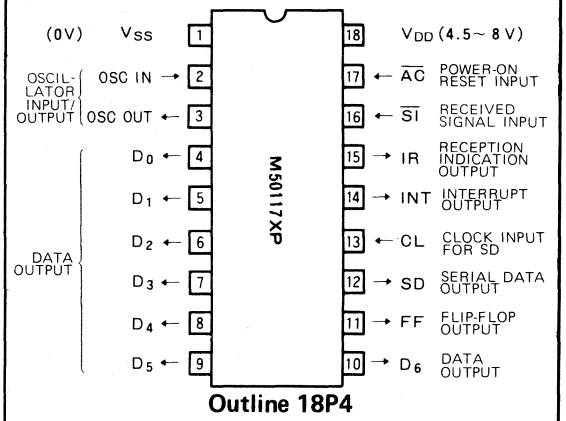
PIN CONFIGURATION (TOP VIEW)



Outline 18P4



Outline 16P4



Outline 18P4

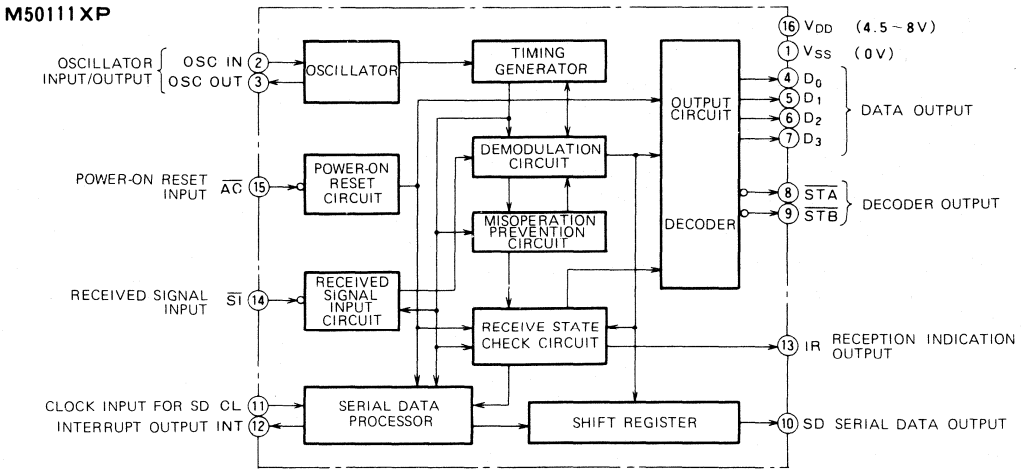
With the data outputs D_0~D_6 and the decode outputs $\overline{STA}$~$\overline{STD}$, M50111XP can process 30 different instructions, the M50116XP can process 60 different instructions and the M50117XP can process 120 different instructions. With a serial data output SD, 120 different instructions can be processed by any of the receivers.

10

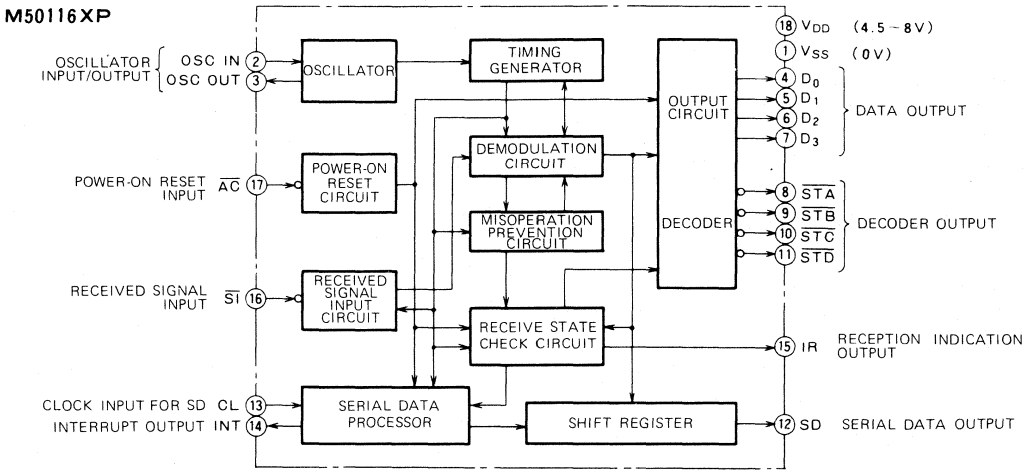
30～120-FUNCTION REMOTE-CONTROL RECEIVER

BLOCK DIAGRAM

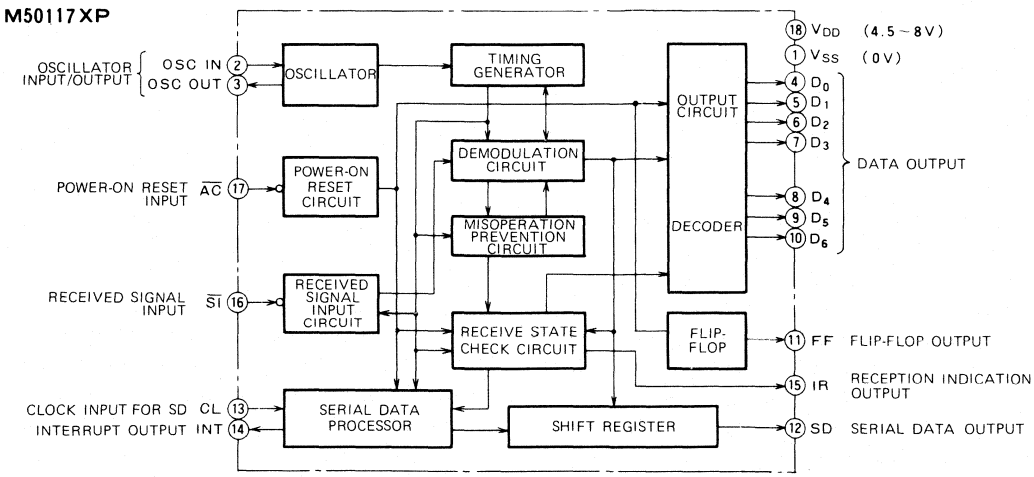
M50111XP



M50116XP



M50117XP



MITSUBISHI ELECTRIC

FUNCTION

Oscillator

As the oscillator is on chip, oscillation frequency is easily obtained by connecting an external LC network or ceramic resonator between the OSC IN and OSC OUT terminals. Fig. 1 and Fig. 2 show examples of typical oscillation circuit.

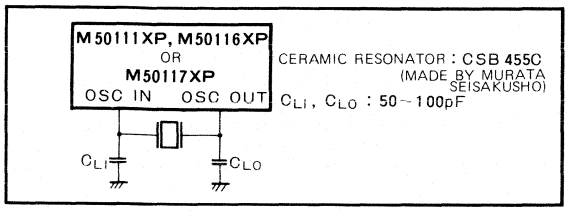


Fig. 1 An example of an oscillator (using a ceramic resonator)

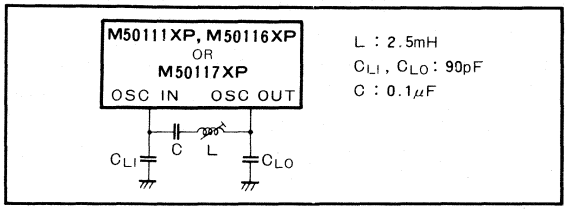


Fig. 2 An example of an oscillator (using an LC network)

When oscillation frequency f_{osc} is 480kHz, execution is as follows:

Received Signal Input Circuit and Demodulation Circuit

The received signal, sensed by the photo detector, is amplified and an integrated signal is supplied through $\overline{SI}$ to be processed by the received signal input circuit, and then it is sent to the demodulation circuit. In the demodulation circuit, the pulse interval of the signal is analyzed and then converted to a digital code. Fig. 3 shows the relationship between the $\overline{SI}$ input wave form, codes and data.

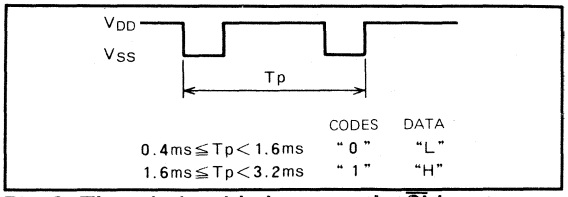


Fig. 3 The relationship between the $\overline{SI}$ input wave form, code and data

When the input pulse interval to the $\overline{SI}$ input is 3.2 ms or longer, it will be assumed to be the end of a word, but if the interval is finally 50 ms or longer it will be accepted as the end of the command transmission and the device will be put in the idle state. In the idle state, the data outputs D_0～D_6 and the reception indication output IR goes to low-level and the decoder outputs $\overline{STA}$～$\overline{STD}$ go to high-level.

Misoperation Prevention Circuit

Any signal whose low-level interval at $\overline{SI}$ input is less than 50～100 μs is not accepted as a transmission signal.

When a pulse interval T_p is less than 0.4 ms, the misoperation prevention circuit resets to idle state to prevent an error. When all data codes D_0～D_4 are supplied as 0 or 1, it resets to idle state.

Receive State Check Circuit

The reception indication output IR becomes high-level after receiving the same transmission code 2 or more times in succession. Therefore reception states of an instruction from the transmitter can be indicated by an LED connected to the output IR.

Reception Code, Data Output, Decode Output and Flip-flop Output

Data outputs D_0～D_6 correspond to D_0～D_6 of the transmission codes. When a code is 0, the data output will be low-level, and when a code is 1, the data output will be high-level, while decode outputs $\overline{STA}$～$\overline{STD}$ correspond to transmission codes D_4, D_5 as shown in Table 1. When the transmission codes D_0～D_6 are 1010000, the flip-flop output FF will go to high-level, and when the codes are 0101000, the output FF will go to low-level.

Table 2 shows the relationship between key codes and type numbers, and examples of their use.

Table 1 The relationship between the decode outputs and the transmission codes D_4, D_5

Transmission code		Decode output			
D_4	D_5	$\overline{STA}$	$\overline{STB}$	$\overline{STC}$	$\overline{STD}$
0	0	L	H	H	H
1	0	H	L	H	H
0	1	H	H	L	H
1	1	H	H	H	L

Table 2 The relationship between be key codes, types numbers examples of their use

Key code			Type number	Use
K_0	K_1	K_2		
0	0	0	M50111P M50116P M50117P	Remote control for audio equipment
1	0	0	M50111AP M50116AP M50117AP	Remote control for TV, VTR
0	1	0	M50111BP M50116BP M50117BP	Remote control for air conditioners and others
0	0	1	M50111CP M50116CP M50117CP	Remote control for video-disk equipment

10

30 ~ 120-FUNCTION REMOTE-CONTROL RECEIVER

Serial Data Processor

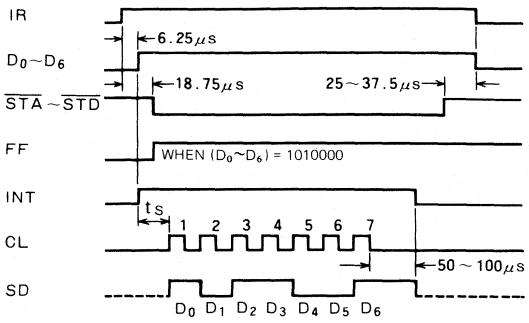
When an identical code is received twice, the reception indication output IR is turned from low-level to high-level and then after 6.25μs delay the interrupt output INT is turned from low-level to high-level (see the timing diagram). When pulses are supplied to the clock input CL for SD while the INT output is high-level, the received data are sent from the serial data output SD. These data are synchronized with the rising edge of the CL input pulses. Thus the contents of the transmission code can be read, if the SD output is decided at the falling edge of the CL input pulses.

The SD output is a three-state output, which is usually in the disabled state (high impedance). After an interrupt output INT goes to high-level, the disabled state is absolved at the first low-to-high transmission of a CL input pulses. And then the data D_0~D_6 is serially sent, and after 50~100μs from the seventh high-to-low transmission of CL input pulses, the SD output is again put in the disabled state and at the same time the INT output goes to low-level.

Power-on Reset Circuit

Attaching a capacitor to the terminal $\overline{AC}$, the power-on reset function can be activated when power supply is applied to the IC. When the $\overline{AC}$ input is turned to low-level, the data outputs D_0~D_6, the reception indication output IR, the interrupt output INT and the flip-flop output FF go to low-level, the decode outputs $\overline{STA}$~$\overline{STD}$ go to high-level and the serial data output SD is put in disabled state.

Timing Diagram



After the INT output becomes high-level, when the received code is not identical to the previously received code before the first fall of the CL input, the INT output is returned to low-level; at the same time the $\overline{STA}$~$\overline{STD}$ outputs become high level and the SD output become a disabled state. After the INT output goes to high-level, when received codes are not identical after the first fall of the CL input, the data D_0~D_6 are sent and then the INT output goes to low-level after 50~100μs from the seventh fall of CL input pulses and the SD output is put in the disabled state.

The time t_s from the rising edge of the INT output to the rising edge of the CL input must be at least 6.25μs.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{DD}	Supply voltage	With respect to V_{SS}	−0.3～9	V
V_I	Input voltage		$V_{SS} \leqq V_I \leqq V_{DD}$	V
V_O	Output voltage		$V_{SS} \leqq V_O \leqq V_{DD}$	V
Pd	Maximum power dissipation	Ta＝25°C	300	mW
Topr	Operating free-air temperature range		−30～70	°C
Tstg	Storage temperature range		−40～125	°C

RECOMMENDED OPERATING CONDITIONS (Ta＝−30～70°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{DD}	Supply voltage	4.5		8	V
V_{IH}	High-level input voltage	$0.7 \times V_{DD}$		V_{DD}	V
V_{IL}	Low-level input voltage	0		$0.3 \times V_{DD}$	V
f_{OSC}	Oscillation frequency		455		kHz
			480		kHz

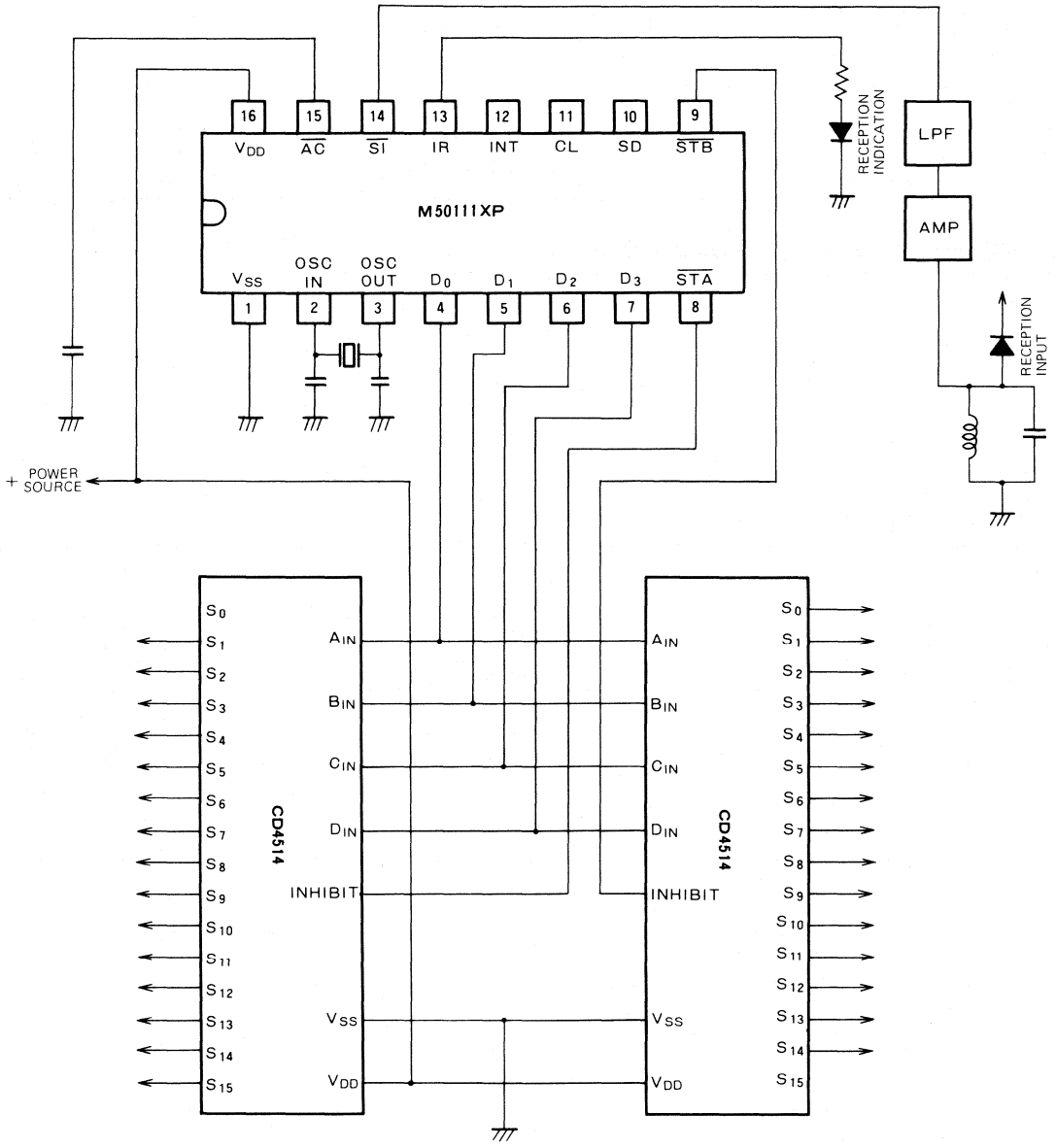
ELECTRICAL CHARACTERISTICS (Ta＝25°C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{DD}	Operational supply voltage	Ta＝−30～70°C, f_{OSC}＝455kHz	4.5		8	V
I_{DD}	Supply current from V_{DD}	V_{DD}＝5V, f_{OSC}＝455kHz		0.3	1.0	mA
I_{OH}	High-level output current, SD	V_{DD}＝4.5V, V_{OH}＝2.4V	−2	−6		mA
I_{OH}	High-level output current, INT, IR	V_{DD}＝4.5V, V_{OH}＝2.4V	−1	−3		mA
I_{OH}	High-level output current, D_0～D_6, $\overline{STA}$～$\overline{STD}$, FF	V_{DD}＝4.5V, V_{OH}＝2.4V	−0.5	−1.5		mA
I_{OL}	Low-level output current, D_0～D_6, $\overline{STA}$～$\overline{STD}$, FF, SD, INT, IR	V_{DD}＝4.5V, V_{OL}＝0.4V	1.6	3.2		mA
R_I	Pull-up resistance, $\overline{SI}$			20		kΩ
R_I	Pull-up resistance, $\overline{AC}$			48		kΩ
R_I	Pull-down resistance, CL			63		kΩ

10

30～120-FUNCTION REMOTE-CONTROL RECEIVER

APPLICATION EXAMPLE M50111XP



MITSUBISHI ELECTRIC

MICROCOMPUTER SYSTEMS

11

MELCS 85/2 MEMORY AND PARALLEL I/O EXPANSION BOARD

DESCRIPTION

The PCA8506 memory and parallel I/O expansion board is designed to be used with the PCA8501 or PCA8540 single-board computer. Memory, parallel I/O ports, and a timer are assembled on a 145 x 125 mm printed circuit board. The PCA8506 can easily be attached to the PCA8501 or PCA8540 single-board computer by using a bus-extension connector.

FEATURES

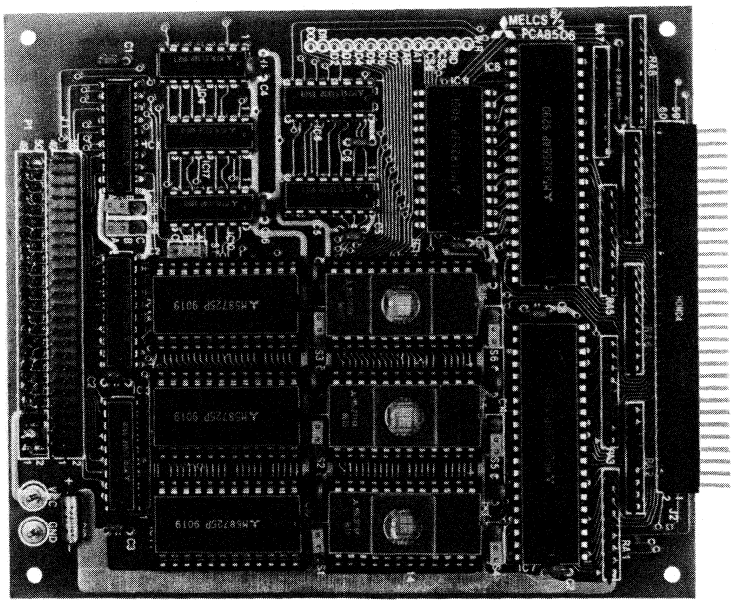
- Expansion board consisting of memory, parallel I/O ports, and a timer
- Memory capacity: 12K bytes
  (expandable in units of 2K bytes RAM or 2K bytes ROM)
- Programmable ports: 48 bits (8 bits x 6 ports)
- Programmable timer: 16 bits x 3
- Power supplied from the PCA8501 or PCA8540
- Compact dimensions (LxWxH): 125x145x17 mm

APPLICATIONS

- Personal computer expansion module
- Control equipment module

FUNCTION

The PCA8506 expansion board consists of up to 12K bytes memory, six 8-bit parallel I/O ports, along with 3 16-bit counters for timer application. The memory can easily be expanded in units of 2K bytes up to 12K bytes using any combination of M5L2716K EPROMs or M58725P static RAMs. The parallel I/O ports consist of 2 (programmable peripheral interfaces) (PPIs) each composed of 3 8-bit I/O ports. The timer consists of a programmable interval timer (PIT) which has 3 16-bit timer counters.



11

MELCS 85/2 MEMORY AND PARALLEL I/O EXPANSION BOARD

OPERATION

The address bus of the CPU is connected to other boards through the address bus buffer. The data bus is connected to the data input/output pins of memory, I/O, and a timer through the bidirectional data bus buffer. The data bus buffer is in an active state only when an IC device on the board is selected. The buffer is ready for output to external units only when the read signal RDCL from the CPU goes low.

Six 24-pin sockets are provided for memory, which are designed for M5L2716K EPROMs. Since the M5L2716K is compatible with the M58725P except for V_{PP}/WR (pin 21), if pin 21 is switched on the connector corresponding to a socket, a M58725P static RAM can be used in place of a M5L2716K EPROM in that socket. It is therefore possible to mix ROMs and RAMs in any order desired by the user.

Since addresses have been allocated on the memory map for the 2 parallel I/O ports and a timer the contents can be read and written in the same way memory is accessed.

All ports of PPIs are initiated to the input mode after the power is turned on, and remain in this mode until a control word is written. As soon as the counter is set by the control word, following the operation mode, the timer will begin to count.

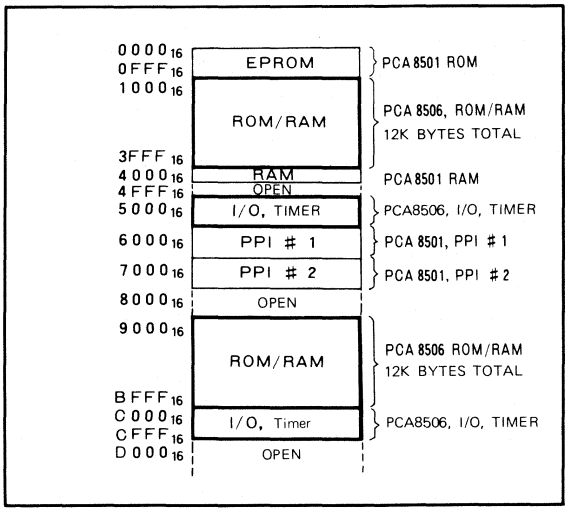
DIMENSIONS

(LxWxH) 125x145x17 mm

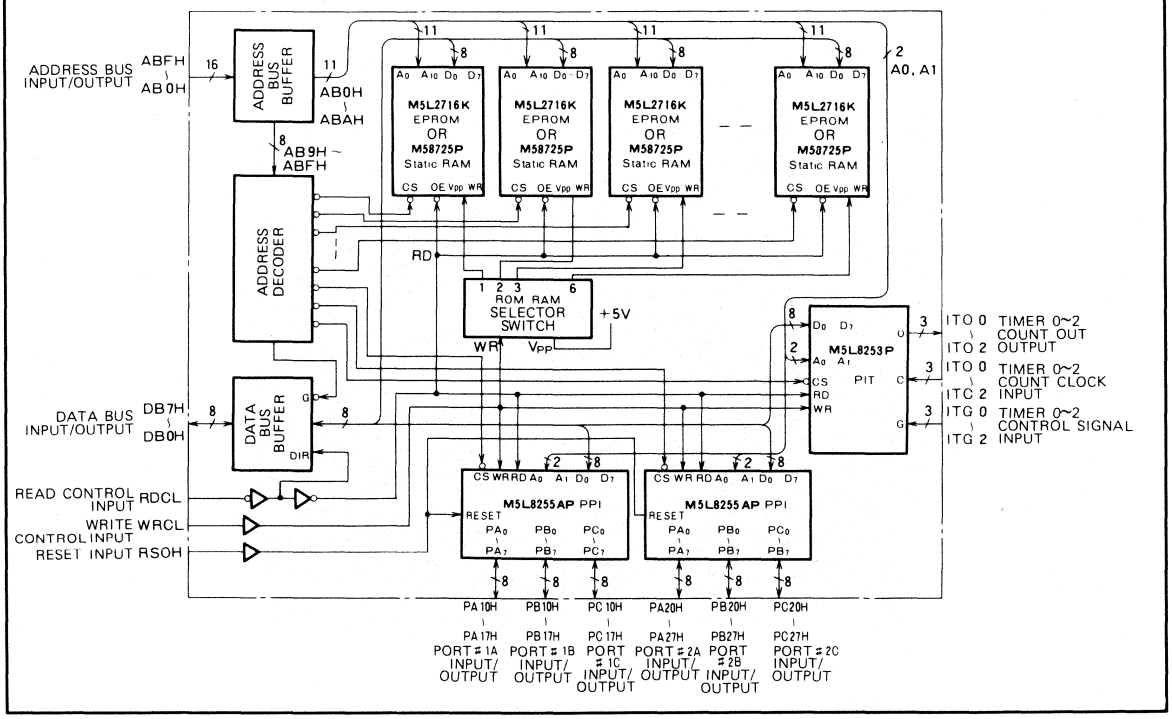
MEMORY AND I/O ADDRESSING

Both memory and I/O addresses can select two areas. When this board is added, different address areas from those of the main board should be selected.

MEMORY MAP



BLOCK DIAGRAM



MITSUBISHI ELECTRIC

SPECIFICATIONS

Memory Address and Memory Capacity

Memory Address (Note 1)

1 : 1000_{16}~$17FF_{16}$
2 : 1800_{16}~$1FFF_{16}$
3 : 2000_{16}~$27FF_{16}$
4 : 2800_{16}~$2FFF_{16}$
5 : 3000_{16}~$37FF_{16}$
6 : 3800_{16}~$3FFF_{16}$

Memory Capacity

#1: 2K bytes (only the socket is supplied)
#2: 2K bytes (only the socket is supplied)
#3: 2K bytes (only the socket is supplied)
#4: 2K bytes (only the socket is supplied)
#5: 2K bytes (only the socket is supplied)
#6: 2K bytes (only the socket is supplied)

Either the M5L2716K EPROM or M58725P RAM can be used in the sockets.

I/O and Timer Addresses and I/O Capacity

I/O and timer addresses (Note 1)

Name		Signal designation	Address
Port #1	PA	PA10H~PA17H	5000_{16}
	PB	PB10H~PB17H	5001_{16}
	PC	PC10C~PC17H	5002_{16}
	CW	Control Word	5003_{16}
Port #2	PA	PA20H~PA27H	5100_{16}
	PB	PB20H~PB27H	5101_{16}
	PC	PC20H~PC27H	5102_{16}
	CW	Control Word	5103_{16}
Timer	COUNTER 0	Interval timer 0	5200_{16}
	COUNTER 1	Interval timer 1	5201_{16}
	COUNTER 2	Interval timer 2	5202_{16}
	CW	Control Word	5203_{16}

Note 1: The address area can be altered by using an inline connector as follows:
Memory 9000_{16}~$BFFF_{16}$
I/O and timer $CXXX_{16}$

I/O Capacity

Port #1 : 8 bits x 3 ports = 24 bits
Port #2 : 8 bits x 3 ports = 24 bits

Interface

Bus : All signals are TTL compatible (fanout LS TTL 1 gate).

I/O and Timer : All signals are TTL compatible.

Connectors

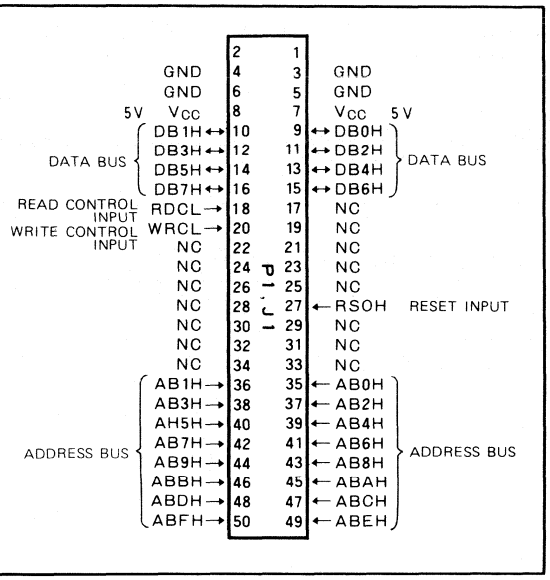
1. P1 (for bus) : Straight dip-type, 50 pins.
2. J1 (for bus) : Straight pin header, T-type, 50 pins.
3. J2 (for I/O) : Angle pin header, L-type, 60 pins.

Power

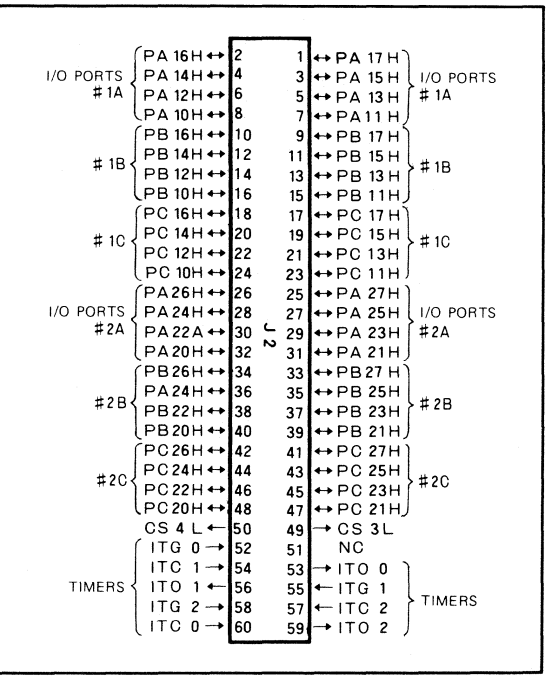
5V, 1A maximum (when six M5L2716Ks are loaded).

PIN CONFIGURATION

Connectors P1 and J1



Connector J2



11

MELCS 85/2 MEMORY AND PARALLEL I/O EXPANSION BOARD

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{CC}	Supply voltage	With respect to GND	0~6.5	V
V_I	Input voltage		5.5	V
V_O	Output voltage		5.5	V
Topr	Operating free-air ambient temperature range		0~55	°C
Tstg	Storage temperature range		−30~70	°C

RECOMMENDED OPERATING CONDITIONS (Ta=0~55°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{CC}	Supply voltage	4.75	5	5.25	V
V_{IH}	Low-level input voltage	2			V
V_{IL}	High-level input voltage			0.8	V

ELECTRICAL CHARACTERISTICS (Ta=0~55°C, V_{CC}=5V±5%, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{OH}	High-level output voltage PA11H~PC27H output	$I_{OH}=-50\mu A$	2.4			V
V_{OH}	High-level output voltage, IT00~IT02 output	$I_{OH}=-150\mu A$	2.4			V
V_{OL}	Low-level output voltage, PA11H~PC27H output	I_{OL}=1.6mA			0.4	V
V_{OL}	Low-level output voltage, IT00~IT02 output	I_{OL}=1.6mA			0.4	V

MELCS 85/2 MEMORY AND SERIAL I/O EXPANSION BOARD

DESCRIPTION

The PCA8507 memory and serial I/O expansion board is designed to be used with the PCA8501 or PCA8540 single-board computer. Memory, a serial I/O port and a timer are assembled on a 145 x 125 mm printed circuit board. The PCA8507 can easily be attached to the PCA8501 or PCA-8540 single-board computer by using a bus-extension connector.

FEATURES

- Expansion board consists of memory, a serial I/O and a timer
- Memory capacity: 12K bytes
  (expandable in units of 2K bytes RAM or 2K bytes ROM)
- Serial I/O port and TTL, RS-232-C interface
- Programmable timer, 16 bits x 3
- Power supply from the PCA8501 or PCA8540
- Compact, dimensions (LxWxH): 125x145x17mm

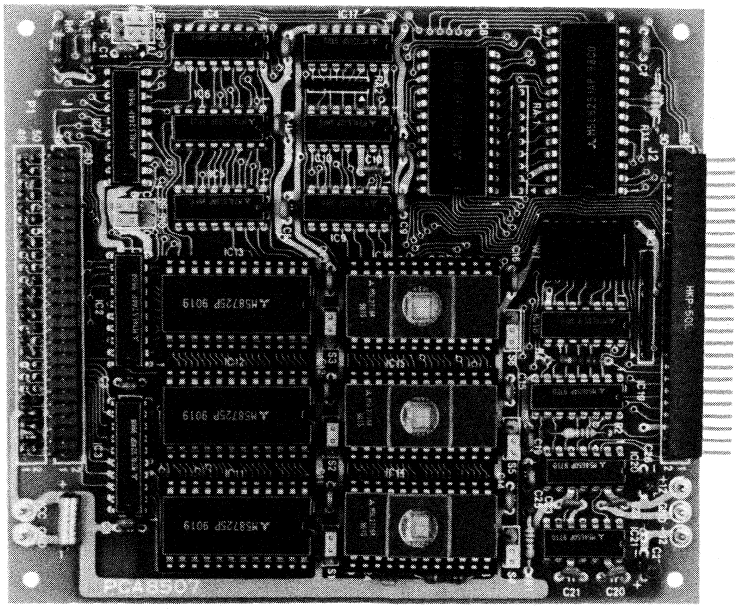
APPLICATIONS

- Personal computer expansion module
- Control equipment module
- Data terminal module

FUNCTION

The PCA8507 expansion board consists of up to 12K bytes of memory, serial I/O port, interface for TTL level and RS-232-C output, along with 3 16-bit counters for timer application.

The memory can easily be expanded in units of 2K bytes up to 12K bytes using any combination of M5L-2716K EPROMs and M5825P static RAMs. The serial I/O port consists of a universal synchronous asynchronous receiver transmitter (USART) for changing parallel/serial and formatting the string in the specified format. Interfaces are provided between the USART and the TTL level or RS-232-C output. The interface is selected by a jumper connection. The timer consists of a programmable interval timer (PIT) which has 3 16-bit timer counters. One of the timers is used by the USART for controlling the baud rate of serial data transfer.



MELCS 85/2 MEMORY AND SERIAL I/ O EXPANSION BOARD

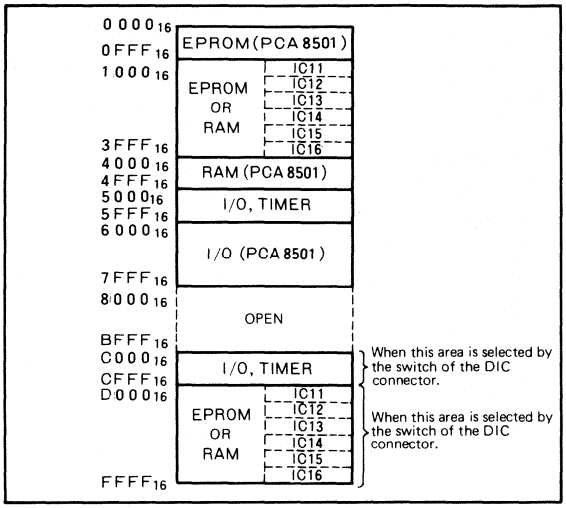
OPERATION

The address bus of the CPU is connected to other boards through the address bus buffer. The data bus is connected to the data input/output pins of memory, I/O, and a timer through the bidirectional data bus buffer. The data bus buffer is in an active state only when an IC device on the board is selected. The buffer is ready for output to external units only when the read signal RDCL from the CPU goes low.

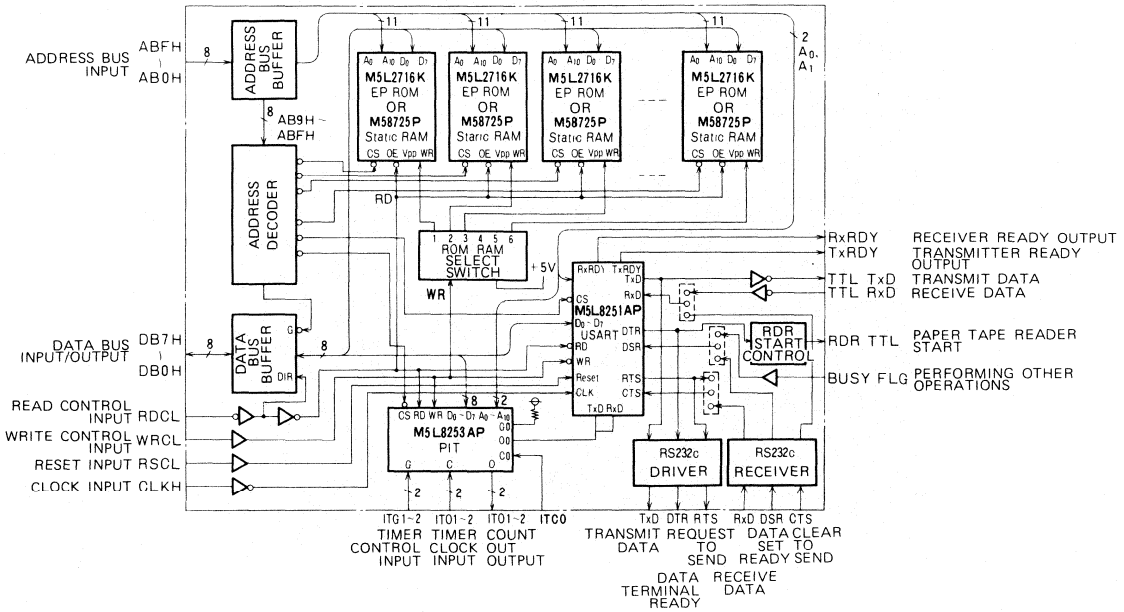
Six 24-pin sockets are provided for memory, which are designed for M5L2716K EPROMs. Since the M5L2716K is compatible with the M58725P except for V_{PP}/WR (pin 21), if pin 21 is switched on the connector corresponding to a socket, an M58725P static RAM can be used in place of an M5L2716K EPROM in that socket. It is therefore possible to mix ROMs and RAMs in any order desired by the user.

Since addresses have been allocated on the memory map for the USART as a serial I/O port and a timer the contents can be read and written in the same way memory is accessed. TTL and standard RS-232-C interfaces are built on the board to interface between the serial I/O port and peripheral devices. Selection of one or the other interface is done by a jumper in the jumper socket. The timer consists of 3 16-bit counters. One of the counters is used as a clock by the USART in setting the baud rate.

MEMORY MAP



BLOCK DIAGRAM



MITSUBISHI ELECTRIC

MELCS 85/2 MEMORY AND SERIAL I/O EXPANSION BOARD

SPECIFICATIONS

Memory Address and Memory Capacity

Memory Address (Note 1)

1 : 1000_{16}~$17FF_{16}$
2 : 1800_{16}~$1FFF_{16}$
3 : 2000_{16}~$27FF_{16}$
4 : 2800_{16}~$2FFF_{16}$
5 : 3000_{16}~$37FF_{16}$
6 : 3800_{16}~$3FFF_{16}$

Memory Capacity

#1: 2K bytes (only the socket is supplied)
#2: 2K bytes (only the socket is supplied)
#3: 2K bytes (only the socket is supplied)
#4: 2K bytes (only the socket is supplied)
#5: 2K bytes (only the socket is supplied)
#6: 2K bytes (only the socket is supplied)

Either the M5L2716K EPROM or M58725P RAM can be used in the sockets.

I/O and Timer Addresses and I/O Capacity

I/O and timer addresses (Note 1)

Name			Signal designation	Address
Serial port	TD		Parallel data	5000_{16}
	CW		Control word	5001_{16}
Timer	COUNTER	0	Interval timer 0	5100_{16}
	〃	1	Interval timer 1	5101_{16}
	〃	2	Interval timer 2	5102_{16}
	CW		Control word	5103_{16}

Note 1 The address area can be altered by using an inline connector as follows:
Memory $D000_{16}$~$FFFF_{16}$
I/O and Timer $CXXX_{16}$

INTERFACE

Bus : All signals are TTL compatible (fanout LS TTL 1 gate).
Timer : All signals are TTL compatible (fanout TTL 1 gate).
Serial I/O : TTL level or RS-232-C standard interface.

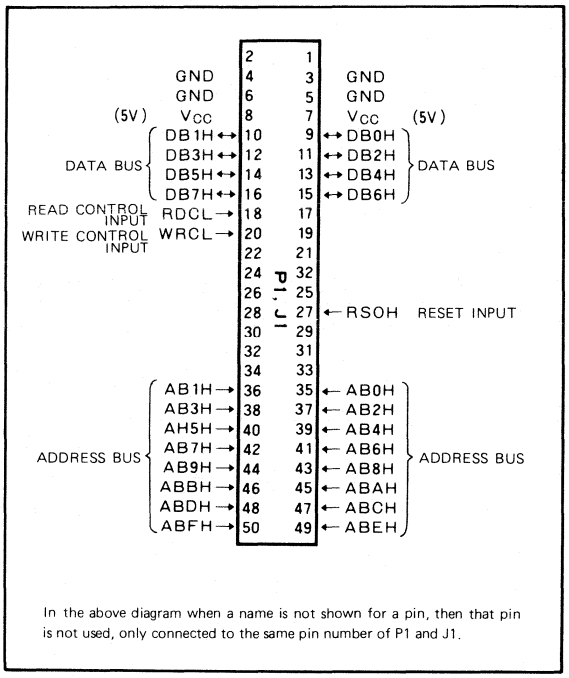
CONNECTORS

1. P1 (for bus): Straight dip-type, 50 pins
2. J1 (for bus): Straight pin header, T-type, 50 pins
3. J2 (for I/O): Angle pin header, L-type, 50 pins

POWER

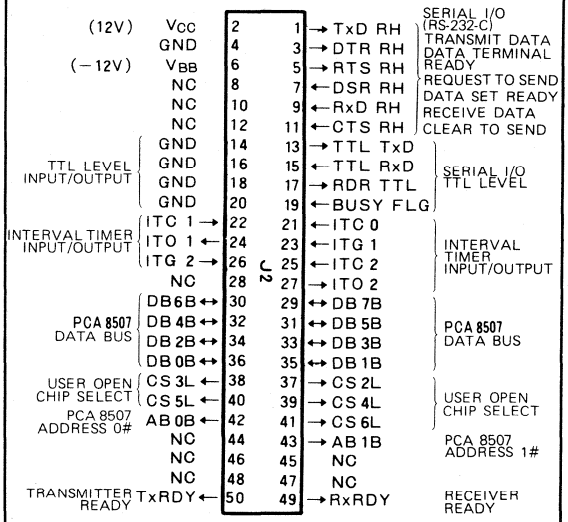
5V, 1A maximum (when six M5L2716Ks are loaded)
±12V (when used as an RS-232-C interface)

PIN CONFIGURATION (Connectors P1 and J1)



In the above diagram when a name is not shown for a pin, then that pin is not used, only connected to the same pin number of P1 and J1.

Connector J2



11

MELCS 85/2 MEMORY AND SERIAL I/O EXPANSION BOARD

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{CC}	Supply voltage	With respect to GND	0～6.5	V
V_{DD}	Supply voltage (plus supply for RS-232-C)		15	V
V_{BB}	Supply voltage (minus supply for RS-232-C)		−15	V
V_I	Input voltage		5.5	V
V_O	Output voltage		5.5	V
Topr	Operating free-air ambient temperature range		0～55	V
Tstg	Storage temperature range		−30～70	°C

RECOMMENDED OPERATING CONDITIONS (Ta=0～55°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{CC}	Supply voltage	4.75	5	5.25	V
V_{IH}	High-level input voltage	3		V_{CC}	V
V_{IL}	Low-level input voltage	0		0.65	V
V_{DD}	Supply voltage (plus supply for RS-232-C)	10.8	12	13.2	V
V_{BB}	Supply voltage (minus supply for RS-232-C)	−13.2	−12	−10.8	V

ELECTRICAL CHARACTERISTICS (Ta=0～55°C, V_{CC}=5V±5%, unless otherwise noted)

Symbol	Parameter		Test conditions	Limits			Unit
				Min	Typ	Max	
V_{OH}	High-level output voltage	DB0B～DB7B	I_{OH}=−3mA	2.4			V
	High-level output voltage	AB0B, AB1B	I_{OH}=−3mA	2.4			V
	High-level output voltage	IT01, IT02	I_{OH}=−150μA	2.4			V
	High-level output voltage	CS2L～CS6L	I_{OH}=−400μA	2.7			V
	High-level output voltage	TxDRH, DTRRH, RTSRH	V_{CC+}=10.8V, V_{CC-}=−13.2V V_{IL}=0.8V, R_L=3～7kΩ	5			V
V_{OL}	Low-level output voltage	DB0B～DB7B	I_{OL}=12mA			0.4	V
	Low-level output voltage	AB0B, AB1B	I_{OL}=12mA			0.4	V
	Low-level output voltage	IT01, IT02	I_{OL}=1.6mA			0.45	V
	Low-level output voltage	CS2L～CS6L	I_{OL}=4mA			0.4	V
	Low-level output voltage	TxDRH, DTRRH, RTSRH	V_{CC+}=10.8V, V_{CC-}=−10.8V V_{IH}=2V, R_L=3～7kΩ			−5	V

PCA8520G01, G02

MELCS 85/3 VOICE GENERATING SINGLE-BOARD COMPUTER

DESCRIPTION

The PCA8520 is a voice generating single-board computer. It consists of an 8-bit M5L8085AP microprocessor, memory, I/O interface, voice reproducing IC, and is fabricated on a single 125 x 145 mm printed circuit board. Voice data is first recorded in EPROMs and is changed into voice data through delta modulation system.

FEATURES

Type	Contents
PCA8520G01	Single-board computer only
PCA8520G02	PCA8520G01 single-board computer . . . . . 1 pc. M5L2716K (007) EPROM for control program storage . . . . . 1 pc. (008~014) EPROMs for voice data storage . . . . . 7 pcs. Speaker . . . . . 1 pc. Instruction manual . . . . . 1 vol.

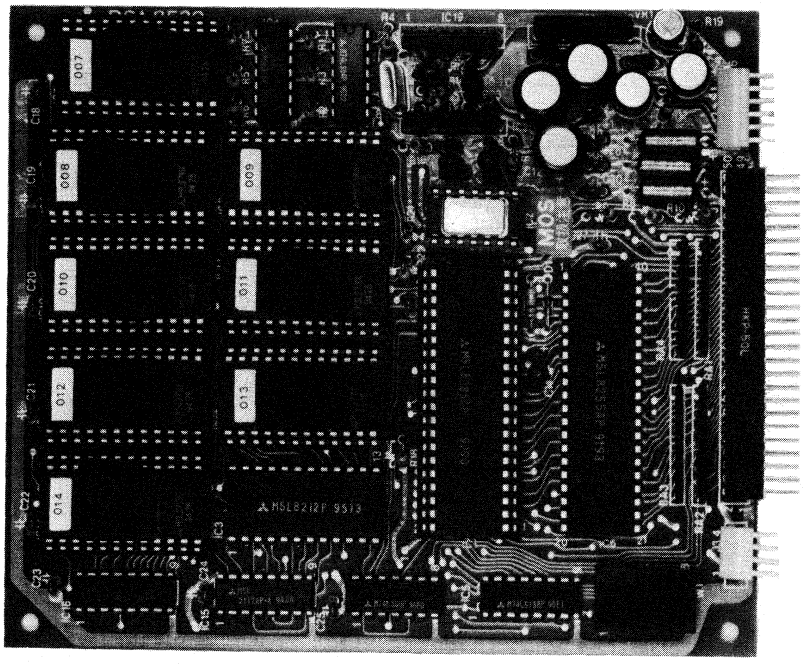
- A single-board computer complete with CPU, memory, I/O interface and voice reproducing IC.
- Storage capacity of the EPROM:
  using M5L2716K: 16K bytes (max)
  using M5L2732K: 32K bytes (max)
- Storage capacity of the RAM: 256 bytes
- I/O interface: 24 bits (8 bits x 3)
- Voice recording time:
  using M5L2716K: 9 seconds (max)
  using M5L2732K: 18 seconds (max)
- Voice maximum output power (at V_{CC2} = 9V): 1W (typ)
- Compact dimensions (LWH) 125x145x20 mm

APPLICATIONS

- An alarm device to be used in factories, offices, etc.
- A recorded sales message device
- An audio output information device
- A device to give voice operation instructions
- Numerical value response for measurement instruments and calculators

FUNCTION

The PCA8520 is a single-board computer with a voice generating function, and is designed around Mitsubishi's M5L8085AP CPU, its LSI family, and voice reproducing IC. It comes with 16K bytes (M5L2716K x 8) or 32K bytes (M5L2732K x 8) of read-only memory and 256 bytes (M5L2112AP) of random-access memory. The



11

PCA8520G01, G02

MELCS 85/3 VOICE GENERATING SINGLE-BOARD COMPUTER

PCA8520 has 1 M5L8255AP programmable peripheral interface (PPI) which offers 24 bits (8 bits x 3) of programmable I/O port.

Voice reproducing is performed through an IC for delta demodulation, a low-pass filter, and a power amplifier. Voice data in the ROM can be sent to the voice reproducing circuit by program control, and then output with 1W of power.

A nine-second message can be output when using 8 M5L2716Ks. An eighteen-second message is possible when using 8 M5L2732Ks. Voice data can be output at both syllable- and word-levels, and can be edited under program control.

OPERATION

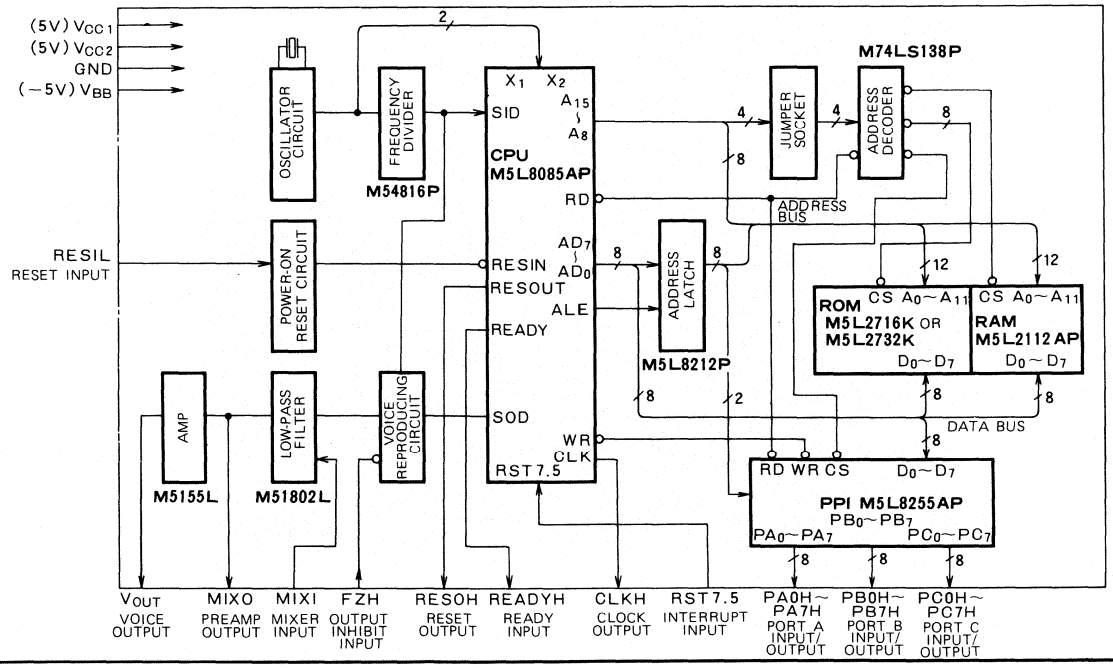
The M5L8085AP CPU executes programs stored in the ROM synchronizing with a quartz ocillator clock. The frequency of this clock is divided by 256 and is supplied to the SID terminal of the CPU and the input of the IC for delta demodulation. The voice can be generated using voice data. This voice data, which is stored in the ROM, is converted parallel to serial and is sent to the SOD terminal in sequence from the most-significant bit.

The M5L2112AP RAM can be used as a data stack, etc. The M5L8255AP PPI can be used for external data inputs or outputs.

BLOCK DIAGRAM NOTATION

Name	Function
Reset circuit	A system reset signal is generated when power is turned on.
Oscillator circuit	The clock is supplied to the CPU and the frequency divider circuit.
Frequency divider circuit	The frequency of the oscillator clock is devided by 256 and is supplied to the voice reproducing circuit.
CPU	Executes the program
Address latch	As the data and low-order address signals are sent from AD_0~AD_7 terminals of the CPU using time-sharing technique, only the address signal is latched into the address latch circuit by the ALE timing signal.
Jumper socket	M5L2716Ks or M5L2732Ks can be selected by simply changing the jumper wire in the jumper socket.
Address decoder	Generates the selection signal of a ROM, RAM, and PPI decoding the high-order bits of the address signal
ROM	Memory to store program and voice data
RAM	Memory to store data stack, temporaly data, etc.
PPI	Is used for external data inputs and outputs
Voice reproducing circuit	Reproduces voice waveform from the digital signal which is sent from the SOD terminal of the CPU
Low-pass filter	This filter only passes low-frequency voice signals.
Amplifier	Voice signal passed through the low-pass filter is amplified to 1W.

BLOCK DIAGRAM



MITSUBISHI ELECTRIC

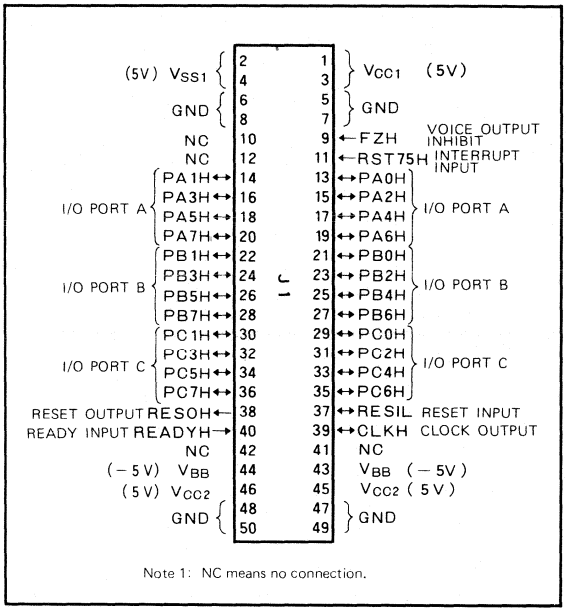
SPECIFICATIONS

Item		Contents
Method		CVSD Method
CPU device		Mitsubishi M5L8085AP
Cycle time		Basic instruction time 2.2μs (at 3.6MHz crystal oscillator frequency)
Memory	ROM	16K bytes (max) using M5L2716Ks address 0000_{16}~$3FFF_{16}$
	ROM	32K bytes (max) using M5L2732Ks address 0000_{16}~$7FFF_{16}$
	RAM	256 bytes address $C000_{16}$~$C0FF_{16}$
I/O interface		Programmable I/O ports: 8 bits x 3 ports (PPI M5L8255AP) address 8000_{16}~8003_{16}
Voice recording time		9 seconds using M5L2716Ks (max) 18 seconds using M5L2732Ks (max)
Voice maximum output power		1W (V_{CC2}=5V, THD=10% f=1kHz)
Interrupt		1 interrupt, 1 level
Auxillary units		MELCS 85/1 microcomputer console voice data unit
Power supply		5V (Two power sources: V_{CC1}, V_{CC2}), -5V
Connectors		Angle pin, header type 50 pins (for the I/O ports) Angle pin, header type 6 pins (for voice output) Angle pin, header type 4 pins (for power)
Physical dimensions		(LxWxH): 125x145x20mm

CONNECTORS

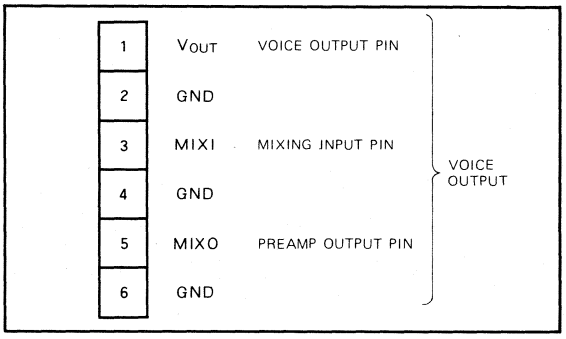
I/O Ports: (Connector J1)
angle pin header L-type 50 pins
Power: (Connector J2)
angle pin header L-type 4 pins
Voice Output: (Connector J3)
angle pin header L-type 6 pins

PIN CONFIGURATION Connector J1

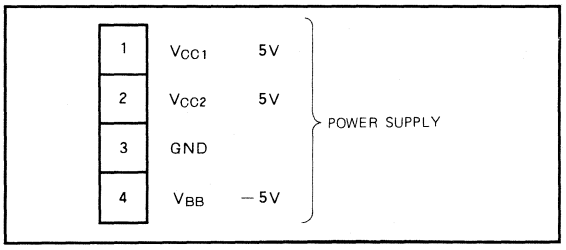


Note 1: NC means no connection.

Connector J2



Connector J3



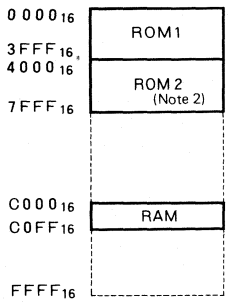
I/O ADDRESS

	PPI			
	Port A	Port B	Port C	C, W
I/O Address	80_{16}	81_{16}	82_{16}	83_{16}

11

MELCS 85/3 VOICE GENERATING SINGLE-BOARD COMPUTER

MEMORY ADDRESS MAP



Note 2: ROM2 is additional storage area when 8 M5L2732Ks are used.
3: ROM is fully decoded, but RAM and PPI are not.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{CC1}	Supply voltage	With respec to GND	0～7	V
V_{CC2}	Supply voltage		0～15	V
V_{BB}	Supply voltage		−15～0	V
V_I	Input voltage		5.5	V
V_O	Output voltage		0～5.5	V
Topr	Operational free-air ambient temperature range		0～55	℃
Tstg	Storage temperature range		−30～70	℃

RECOMMENDED OPERATING CONDITIONS (Ta＝0～55℃, unless otherwise noted.)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{CC1}	Supply voltage	4.75	5	5.25	V
V_{CC2}	Supply voltage	4	5	12	V
V_{BB}	Supply voltage	−18	−5	−4	V
V_{IH}	High-level input voltage	2			V
V_{IL}	Low-level input voltage			0.8	V

ELECTRICAL CHARACTERISTICS (Ta＝0～70℃, unless otherwise noted.)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{OH}	High-level output voltage, RES0H, CLKH	$I_{OH}=-400\,\mu A$	2.4			V
V_{OL}	Low-level output voltage, RES0H, CLKH	$I_{OL}=2mA$			0.45	V
V_{OH}	High-level output voltage, PA0H～PC7H	$I_{OH}=-200\,\mu A$	2.4			V
V_{OL}	Low-level output voltage, PA0H～PC7H	$I_{OL}=1.7mA$			0.45	V
V_{IH}	High-level input voltage, RESIL		2.4		V_{CC} +0.5	V
V_{IL}	Low-level input voltage, RESIL		−0.3		0.8	V
V_{IH}	High-level input voltage, READY, RST75H		2.2		V_{CC} +0.5	V
V_{IL}	Low-level input voltage, READY, REST75H		−0.3		0.8	V
P_O	Voice maximum output power,	THD＝10%, f＝1kHz, $V_{CC2}=9V$, $R_L=8\Omega$, Ta＝25℃	0.7	1		W
I_{CC1}	Supply current from V_{CC1}	When 8 M5L2716K EPROMs are used.		450	900	mA
I_{CC2}	Supply current from V_{CC2}				400	mA
I_{BB}	Supply current from V_{BB}				100	mA

MITSUBISHI ELECTRIC

MELCS 85/2 COLOR TV DISPLAY SINGLE-BOARD COMPUTER

DESCRIPTION

The PCA8540 is a single-board computer of the MELPS 85 LSI family. The TV interface is fabricated on a single 125 x 145 mm printed circuit board. It provides for screen displaying with a resolution of 256 x 192 elements maximum in 2 colors, up to 8 colors in semigraphic 4, or up to 64 ASCII coded characters. A simple connection to the antenna terminal allows it to be used with a home color TV receiver. The PCA8540 also produces composite video signals that can be connected directly to the video monitor.

FEATURES

Type	Function
PCA8540G01	For home-use TV with output of NTSC system signals for Japan Channel 1 or 2 Contains no EPROMs Contains only one M58725P for screen memory
PCA8540G02	For video monitor TV with monochrome video monitor signals Contains no EPROMs Contains only one M58725P for screen memory

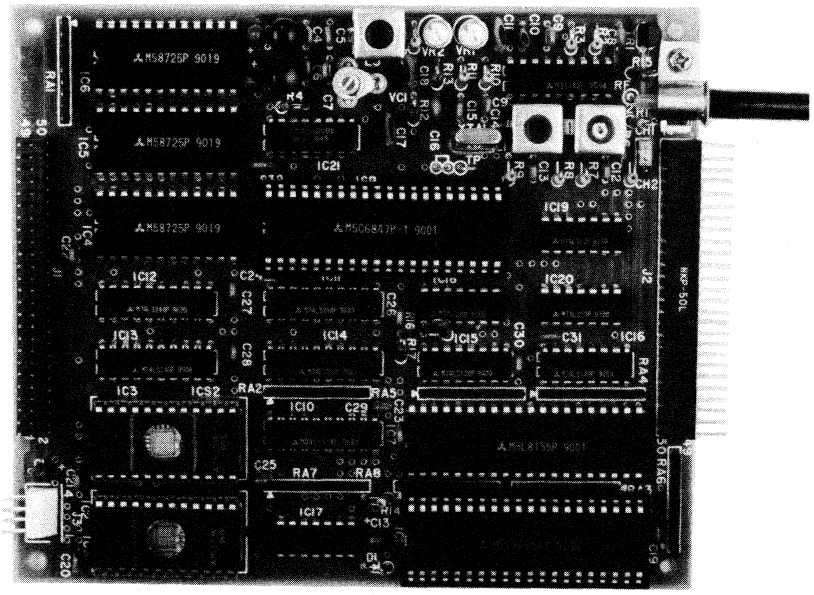
- A single-board computer complete with CPU, memory, I/O and TV interface
- Enables up to 256(H) x 192(V) elements graphic display on a home color TV receiver (or monochrome video monitor)
- Up to 64 characters can be displayed
- The 64 ASCII characters are stored on an internal character generator ROM and can be displayed together with semigraphics 4 mode
- Provide 9 colors on screen: green, yellow, blue, red, light gray, cyan, magenta, orange and black
- ROM 4K bytes (max) + RAM 256 bytes or ROM 2K bytes + RAM 2.25K bytes
- Programmable I/O port with timer: 22 bits
- Compact: dimensions (LxWxH): 125x145x20 mm
- Expandable memory and I/O (using memory I/O expansion board PCA8506 or PCA8507)

APPLICATION

- TV games
- Data terminal with graphic capability
- Display terminal for microcomputer systems
- Commercial advertising display
- Slave computer for a MELCS 85/2 system

FUNCTION

The PCA8540 is a single-board computer, with color TV display capabilities designed to be compatible with the Mitsubishi M5L8085AP CPU and its LSI family as well as the VDG (video display generator) LSI M5C6847P-1. The PCA8540 comes with 4K bytes of ROM + 256 bytes



MELCS 85/2 COLOR TV DISPLAY SINGLE-BOARD COMPUTER

of RAM or 2K bytes of ROM + 2,304 bytes of RAM along with 3 I/O ports (22 bits). ROM, RAM and I/O are provided by the M5L2716K x 2 + M5L8155A or M5L2716K + M58725P + M5L8155A.

The TV interface of the G01 system consists of a VDG LSI M5C6847P-1 (TV interface), an M51342P RF modulator IC and 3 M58725P 16K static RAMs which are used for screen display memory. The G02 system has a video amp circuit instead of an M51342P.

As the various display modes can be programmed using an M5C6847P-1, the following can be displayed.

- Character display, pattern stored on internal ROM
- Reverse character display (one character)
- Semigraphics 4 (up to 8 colors)
- Semigraphics 6 (up to 4 colors)
- 64 x 64 4 colors
- 128 x 64 2 colors
- 128 x 64 4 colors
- 128 x 96 2 colors
- 128 x 96 4 colors
- 128 x 192 2 colors
- 128 x 192 4 colors
- 256 x 192 2 colors

The PCA8506 and PCA8507 are used, for memory I/O expansion boards, to expand to a maximum of 16K bytes of ROM or RAM, an RS-232-C serial interface can be used.

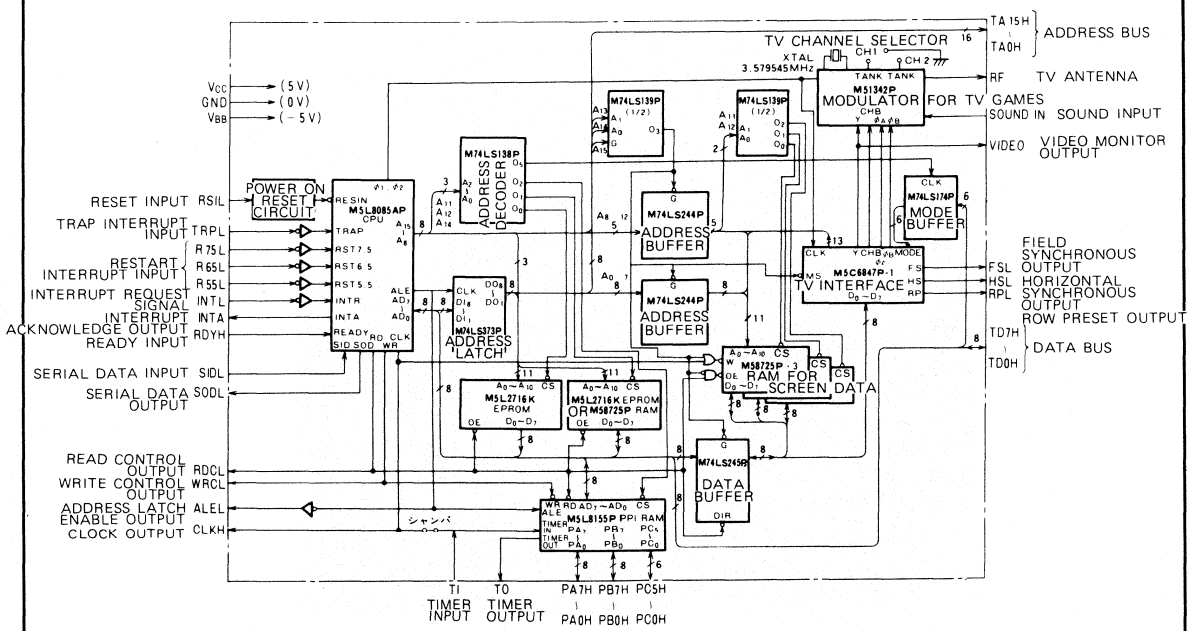
OPERATIONS

The program for the M5L8085AP CPU is normally stored on 2 M5L2716K EPROMs (2 x 2K bytes) and an M5L-8155P RAM (256 bytes) but 1 M5L2716K EPROM can be replaced by an M58725P RAM (2K bytes). Data transmission to and from external sources is done through the ports of the M5L8155P.

There is a data buffer between the M5C6847P-1 and the CPU on the address and data bus. This allows the M5C-6847P-1 to operate independently of the CPU when reading information from the M58725P RAM for screen data. It adds synchronous signal before it is output serially to the M51342P TV game modulator. The signal includes the intensity and color signals which are modulated by the M51342P into NTSC system TV signals for channel 1 or 2. The M5C6847P-1's composite video signal can be used for input to the monochrome video monitor.

When the CPU accesses the RAM, addresses 6000_{16}~$77FF_{16}$ for screen data, $\overline{MS}$ of the M5C6847P-1 will be at low-level and the address output will be in high-impedance state. During this period the CPU can change the contents of the RAM for screen data. The CPU can also change the display mode of the M5C6847P-1 through the data bus by accessing mode set address 4800_{16}.

BLOCK DIAGRAM (PCA8540G01)



MITSUBISHI ELECTRIC

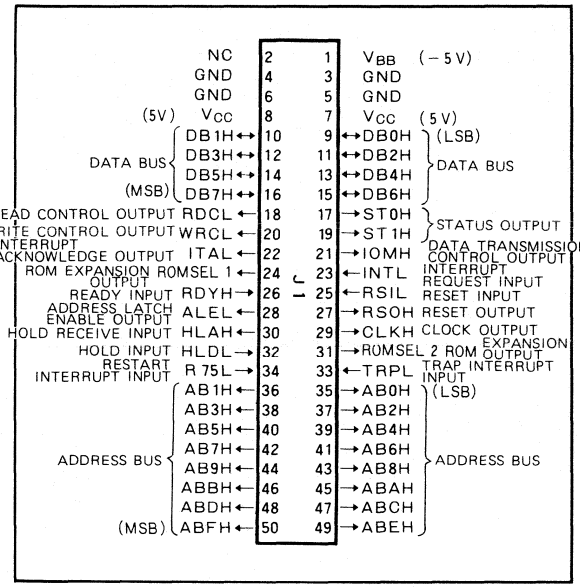
MELCS 85/2 COLOR TV DISPLAY SINGLE-BOARD COMPUTER

SPECIFICATIONS

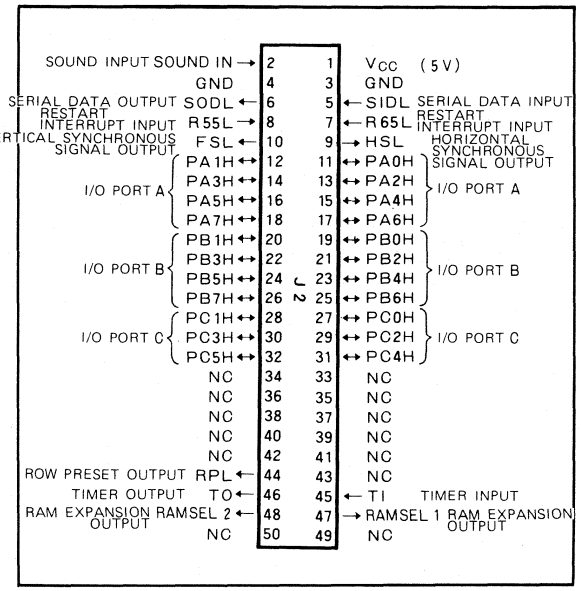
Item	Description
Method	8-bit parallel operation
CPU Component	Mitsubishi's M5L8085AP (equivalent to the intel 8085A)
Cycle time	Basic instruction time 2.23μs (at clock frequency 1.79 MHz)
Memory	EPROM 4K bytes (M5L2716K x 2) Address 0000_{16}~$0FFF_{16}$ or 2K bytes (M5L2716K x 1) (Note 1) Address 0000_{16}~$07FF_{16}$ RAM 256 bytes (M5L8155P) Address 4000_{16}~$40FF_{16}$ or 2304 bytes (M5L8155P + M58725P) Address 0800_{16}~$0FFF_{16}$ (Note 1) 4000_{16}~$40FF_{16}$ Screen Memory (Note 2) 6K bytes (M5872P x 3) Address 6000_{16}~$77FF_{16}$
I/O	Programmable port 22 bits (M5L8155P) Address 4100_{16}~4105_{16} Serial input/output Opens SID, SOD of CPU
Video output	G01: NTSC system, Japan, channel 1 or 2 G02: Monochrome composite video monitor signal
Display method	Priority CPU
Interrupt	5-level (INTR, RST55, RST65, RST75, TRAP)
Support device	PCA0803 (program checker) can be used PC8500 (portable microcomputer console) can be used.
Power supply	G01: 5V ±5%, -5V ±5% G02: 5V ±5%
Applicable connector	Straight pin header 50 pins (for bus extension) Angle pin header 50 pins (for I/O port)
Physical dimensions	(L x W x H): 125 x 145 x 20 mm

Note 1: By switch of ROM/RAM connector.
2: 0.5K bytes are used for screen data and 5.5K bytes can be used for data.

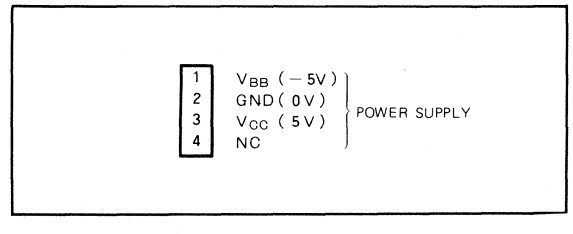
PIN CONFIGURATION Connector J1



Connector J2



Connector J3



11

MELCS 85/2 COLOR TV DISPLAY SINGLE-BOARD COMPUTER

INTERRELATION BETWEEN EACH MODE AND THE SCREEN

$\overline{A}/S$ (D_7)	INV (D_6)	$\overline{A}/G$ (D_5)	$\overline{I}/E$ (D_4)	CSS (D_3)	GM2 (D_2)	GM1 (D_1)	GM0 (D_0)	Color (Note 5) Character	Color (Note 5) Background	Color (Note 5) Border	Display	Data	Mode	Memory Capacity (Bytes)
0	0	0	0	0	X	X	X	Green	Black	Black	5 x 7 Dot matrix 1 char.	$D_7D_6D_5D_4D_3D_2D_1D_0$ (Note 4) (Note 3)	Alpha numeric 32 characters x 16 lines	0.5K
	1							Black	Green					
0	0	0	0	1	X	X	X	Orange	Black	Black				
	1							Black	Orange					
1	X	0	0	X	X	X	X	8 Color ①		Black	D_3 D_2 / D_1 D_0	(Note 4) Color Luminance	Semigraphic 4 64 x 32 picture elements	0.5K
X	X	0	1	0	X	X	X	4 Color ②		Black	D_5 D_4 / D_3 D_2 / D_1 D_0	Color Luminance	Semigraphic 6 64 x 48 picture elements	0.5K
X	X	0	1	1	X	X	X	4 Color ③						
X	X	1	X	0	0	0	0	4 Color ④		Green	$E_3 \sim E_0$	Color E_3 E_2 E_1 E_0	64×64 Color graphic	1K
X	X	1	X	1	0	0	0	4 Color ⑤		Dark gray				
X	X	1	X	0	0	0	1	2 Color ⑥		Green	$D_7 \sim D_0$	Luminance	128×64 Graphic	1K
X	X	1	X	1	0	0	1	2 Color ⑦		Dark gray				
X	X	1	X	0	0	1	0	4 Color ④		Green	$E_3 \sim E_0$	Color E_3 E_2 E_1 E_0	128×64 Color graphic	2K
X	X	1	X	1	0	1	0	4 Color ⑤		Dark gray				
X	X	1	X	0	0	1	1	2 Color ⑥		Green	$D_7 \sim D_0$	Luminance	128×96 Graphic	2K
X	X	1	X	1	0	1	1	2 Color ⑦		Dark gray				
X	X	1	X	0	1	0	0	4 Color ④		Green	$E_3 \sim E_0$	Color E_3 E_2 E_1 E_0	128×96 Color graphic	3K
X	X	1	X	1	1	0	0	4 Color ⑤		Dark gray				
X	X	1	X	0	1	0	1	2 Color ⑥		Green	$D_7 \sim D_0$	Luminance	128×192 Graphic	3K
X	X	1	X	1	1	0	1	2 Color ⑦		Dark gray				
X	X	1	X	0	1	1	0	4 Color ④		Green	$E_3 \sim E_0$	Color E_3 E_2 E_1 E_0	128×192 Color graphic	6K
X	X	1	X	1	1	1	0	4 Color ⑤		Dark gray				
X	X	1	X	0	1	1	1	2 Color ⑥		Green	$D_7 \sim D_0$	Luminance	256×192 Graphic	6K
X	X	1	X	1	1	1	1	2 Color ⑦		Dark gray				

Note 3: INV (reverse of character) is determined by D_6 of data (when D_6 1→ INV 1, 0→ 0)

4: When $\overline{I}/E$ (D_4) = 0, $\overline{A}/S$ is determined by D_7 of data (1 = Semigraphics 4 mode, 0 = Character mode)

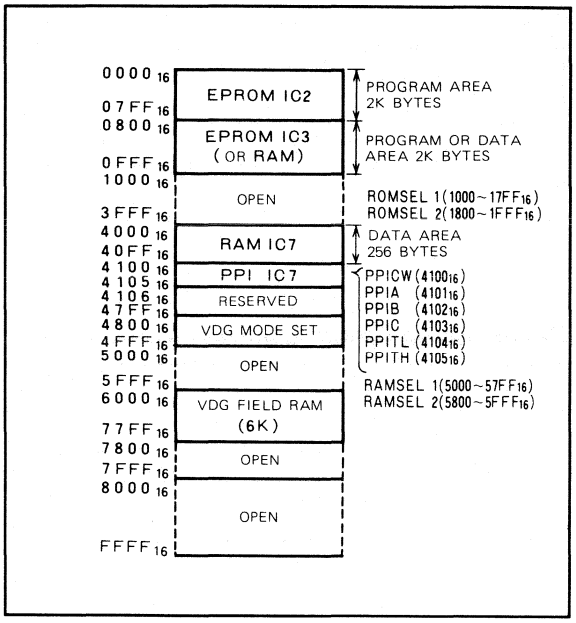
5: Details regarding color are on the next page.

MITSUBISHI ELECTRIC

COLOR DETAILS

	Color data				
	D_6	D_5	D_4	$D_3 \sim D_0$	
8 Colors ①	X	X	X	0	Black
	0	0	0	1	Green
	0	0	1	1	Yellow
	0	1	0	1	Blue
	0	1	1	1	Red
	1	0	0	1	Dark gray
	1	0	1	1	Cyan
	1	1	0	1	Magenta
	1	1	1	1	Orange
	CSS	D_7	D_6	$D_5 \sim D_0$	
4 Colors ②	0	X	X	0	Black
	0	0	0	1	Green
	0	0	1	1	Yellow
	0	1	0	1	Blue
	0	1	1	1	Red
	1	X	X	0	Black
4 Colors ③	1	0	0	1	Dark gray
	1	0	1	1	Cyan
	1	1	0	1	Magenta
	1	1	1	1	Orange
	CSS	D_7	D_6	(D_5D_4, D_3D_2, D_1D_0)	
4 Colors ④	0	0	0		Green
	0	0	1		Yellow
	0	1	0		Blue
	0	1	1		Red
4 Colors ⑤	1	0	0		Dark gray
	1	0	1		Cyan
	1	1	0		Magenta
	1	1	1		Orange
	CSS	D_7	($D_6 \sim D_0$)		
2 Colors ⑥	0	0			Black
	0	1			Green
2 Colors ⑦	1	0			Black
	1	1			Dark gray

MEMORY ADDRESS MAP



MEMORY CAPACITY AND I/O EXPANSION

The capacity of the PCA8540 can be easily expanded by the addition of other boards such as the PCA8506 or PCA8507.

PCA8506 (ROM, RAM and Parallel I/O Extension)

Features

- Memory capacity . . . . . 12K bytes (Note 6)
- Programmable ports 48 bits . . . . . 8 bits x 6
- Programmable timers . . . . . 16 bits x 3
- Small size, dimensions (LxWxH) . . . . 145x125x17 mm

PCA8507 (ROM, RAM and Serial I/O Extension)

Features

- Memory capacity . . . . . 12K bytes (Note 6)
- Serial port (RS-232-C or TTL Level) . . . . . 1 port
- Programmable timers . . . . . 16 bits x 3
- Small size, dimensions (LxWxH) . . . . 145x125x17 mm

Note 6: The memory can easily be expanded in units of 2K bytes up to 12K bytes using any combination of M5L2716K EPROMs and M58725P static RAMs.

11

MELCS 85/2 COLOR TV DISPLAY SINGLE-BOARD COMPUTER

Absolute Maximum Ratings

Symbol	Parameter		Conditions	Limits	Unit
V_{CC}	Supply voltage		With respect to GND	0～7	V
V_{BB}	Supply voltage			0.3～−6.5	V
V_I	Input voltage			5.5	V
Topr	Operational free-air ambient temperature range	PCA8540 G01	Logical circuit	5～50	°C
		PCA8540 G02	Logical circuit	0～55	°C
Tstg	Storage temperature range			−10～70	°C

Recommended Operating Conditions (Ta = 5～40°C, unless otherwise noted.)

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{CC}	Supply voltage	4.75	5	5.25	V
V_{BB}	Supply voltage	−5.25	−5	−4.75	V
V_{IH}	High-level input voltage	2			V
V_{IL}	Low-level input voltage			0.8	V

Electrical Characteristics (V_{CC} = 5V ± 5%, V_{BB} = −5V + 5%, Ta = 25°C, unless otherwise noted.)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{OH}	High-level output voltage, PA0H～PC5H	I_{OH} = −50μA	2.4			V
V_{OH}	High-level output voltage, AB0H～AB7H	I_{OH} = −900μA	3.65			V
V_{OH}	High-level output voltage, AB8H～ABFH	I_{OH} = −300μA	2.4			V
V_{OH}	High-level output voltage, RS0H, CLKH, HLAH, ALEL	I_{OH} = −300μA	2.4			V
V_{OH}	High-level output, other outputs	I_{OH} = −400μA	2.4			V
V_{OL}	Low-level output voltage, PA0H～PC5H	I_{OL} = 1.8mA			0.4	V
V_{OL}	Low-level output voltage, AB0H～AB7H	I_{OL} = 16mA			0.5	V
V_{OL}	Low-level output voltage, AB8H～ABFH	I_{OL} = 1.9mA			0.45	V
V_{OL}	Low-level output voltage, CLKH, ALEL	I_{OL} = 8mA			0.4	V
V_{OL}	Low-level output, other outputs	I_{OL} = 1.9mA			0.4	V
I_{CC}	Supply current from V_{CC}	When 2 EPROMs are loaded.		0.6	1.3	A
I_{BB}	Supply current from V_{BB}	When 2 EPROMs are loaded.		0.05	0.2	A
f_{CLK}	CPU clock frequency			1.79		MHz
f_{CH1}	RF output frequency 1			91.25		MHz
f_{CH2}	RF output frequency 2			97.25		MHz
f_{SUB}	Color sub-carrier frequency			3.579545		MHz

MICROCOMPUTER SUPPORT SYSTEMS

12

MELCS 4/1 PORTABLE DEVELOPMENT SUPPORT SYSTEM

DESCRIPTION

The PC0400 was designed for development support for the M58840-XXXP single-chip 4-bit microcomputer, and for the debugging and maintenance of systems using the M58840-XXXP.

FEATURES

- RAM-base debugging capability
- Program execution and breakpoint capability from any address
- Single-step operating capability
- Display/alter capability for contents of all registers and flags in the CPU
- Write/erase capability for the M5L2708K EPROM
- Connection of users' systems by an accessory adaptor card or by cable
- Comes with convenient carrying case

APPLICATION

- The development, design, inspection and maintenance of hardware and software for systems using the M58850-XXXP.

FUNCTION

As can be seen in the block diagram, the PC0400 is composed of the following hardware:

1. Monitor CPU and monitor ROM
2. Input ports
3. Output ports
4. Keyboard/display
5. EPROM writer and eraser
6. Program memory and control circuit
7. Debug CPU and peripheral circuits

The debug CPU makes use of the M58842S MELPS 4 system evaluation device, supervised and controlled by the M5L8085AP monitor CPU. The program memory stores the memory under development, and is connected to the monitor CPU for program setting and confirmation of its contents, and to the debug CPU for program execution. The control circuits control the address bus and data bus of the program memory.

The EPROM writer is used to store the debugged program in the EPROM, accepting and displaying inputs from the keyboard.

Interfacing with users' systems is normally accomplished by means of a J4 42-pin connector. A J3 40-pin connector is used to interface with the touch-keyboard.

MELCS 4/1 PORTABLE DEVELOPMENT SUPPORT SYSTEM

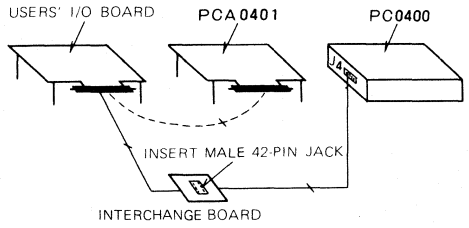
OPERATION

As can be seen in the application diagrams the PC0400 is used in cases where the PCA0401 MELCS 4 single-board system-evaluation computer is applied to system evaluation, and is connected by its J4 connector via a relay board to the PMP connector of the PCA0401.

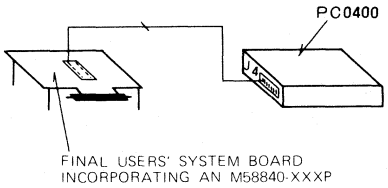
When the PCA0401 is not used, the PC0400 is used as an in-circuit emulator, connected by a J4 connector to the 42-pin socket of the users' system CPU.

Application Diagrams

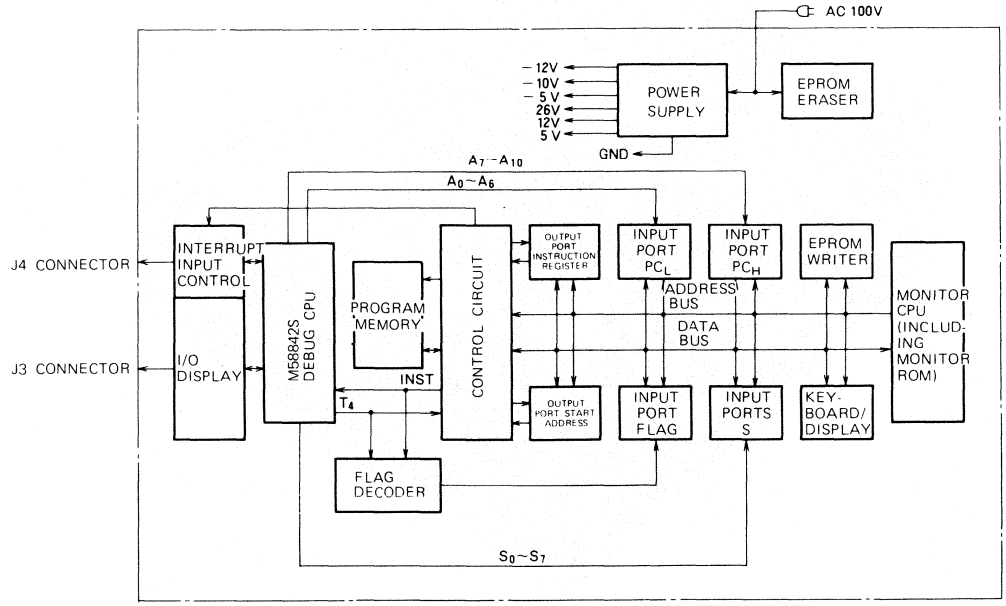
a) Prototype debugging



b) Final product debugging

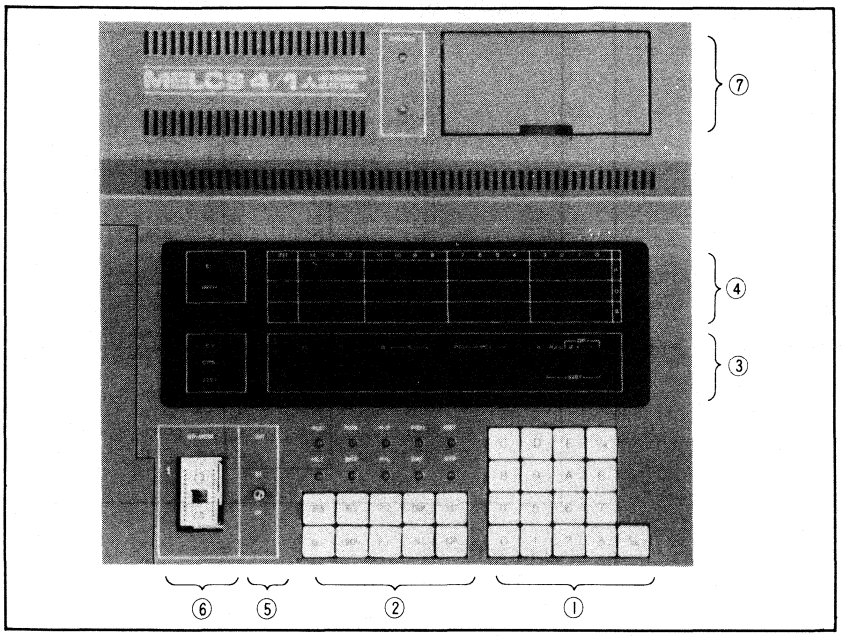


BLOCK DIAGRAM



MITSUBISHI ELECTRIC

Panel Operation



① Data Keys : 16 keys for hexadecimal correction entering, address, and other data, plus and E/N key for mode designation.

② Command Keys : 10 keys labelled as follows:

Name of key	Function
RST	For resetting the M58842S debug CPU, also used for cancelling monitor mode designations and breakpoints.
RUN	For starting programs from the desired address
STP	For executing programs from the desired address (single step) every instruction cycle.
HLT	For halting program execution
BKP	For designating and confirming breakpoints
PGM	For confirmation or alteration of the contents of the program memory
DAT	For confirmation or alteration of the contents of the data memory of the debug CPU.
REG	For confirmation of data in all registers, and alteration of data in the register displayed on ③
P-L	Loads contents of EPROM to program memory
P-G	Writes contents of the program RAM to the EPROM.

③ 7-Segment LEDs: 13 LEDs for the display of debug CPU status.

④ LESs: 40 dot LEDs for the display of the internal status and external status of the debug CPU.

⑤ Interrupt Control: For the debug CPU. Two positions: EI (enable int) and DI (disable int)

⑥ EPROM Socket: When occupied by the M5L2708K, S EPROM (equivalent to Intel's 2708), allows movement of data from the EPROM to the program memory and vice versa.

⑦ EPROM Eraser and Erase Switch : When the EPROM is inserted in the eraser and the switched pressed, a lamp lights and erasing begins and proceeds automatically by timer for approximately 25 min.

12

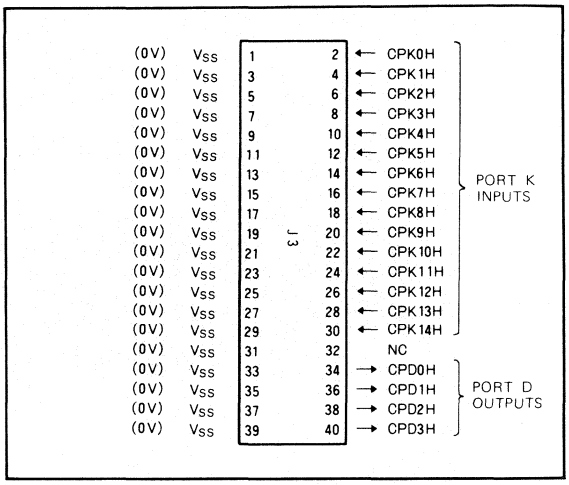
MELCS 4/1 PORTABLE DEVELOPMENT SUPPORT SYSTEM

SPECIFICATIONS

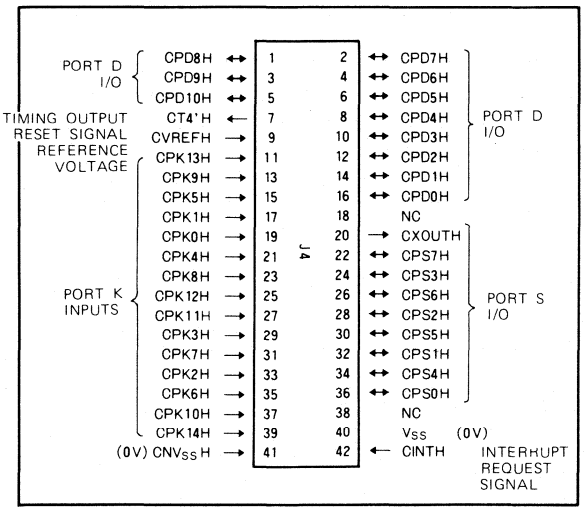
Organization		4-bit parallel processing
CPUs	Monitor	M5L8085AP
	Debug	M58842S
Cycle time		10μs @ 600kHz operation
Clock frequency		300 ~ 600kHz, variable with internal CR/ceramic filter switching
Program memory		Six M5L2114LP (2K words x 9 bits) RAMs Address: $3000_{16} \sim 3FFF_{16}$ Program loaded in RAMs from EPROM
Connectors		J4 42-pin for interfacing with users systems J3 40-pin for interfacing with touch-keyboard J1 40-pin for internal board connections J2 40-pin for internal board connections J5 19-pin for power supply
Dip switches	DS1, DS2	DS1, DS2 for K port inputs without using the touch keys. When keys are used all 16 circuits are off.
	DS3	DS3 for CR/ceramic switching of the clock, and internal/external switching of V_{REF}
Keyboard/ display		Keyboard 27 keys Display 13 7-segment LEDs and 40 dot LEDs
External interrupt control switch		Toggle switch for disable/ enable
EPROM eraser		Erasing time approx. 25min
Input power supply		100 ± 10% VAC 50/60Hz
Operating ambient temp.		0 °C ~ 55°C
Storage temperature		−30°C ~ 70°C
Accessories		Power supply cable 1 User interface cable with DIL connector 1 Adaptor card 1
Overall dimensions (carrying case)		363 x 363 x 135 mm
Weight		9 kg

PIN CONFIGURATIONS

Connector J3



Connector J4



MITSUBISHI ELECTRIC

MICROCOMPUTER SOFTWARE

13

CROSS ASSEMBLER

DESCRIPTION

The MELPS 42 cross assembler has been prepared for the development of application programs suitable for equipment using the M58496-XXXP single-chip 4-bit microcomputer.

This cross assembler not only provides many pseudo instructions, control commands, and control data for improving programming efficiency, but it also provides program versatility for changing instruction codes and functions.

FEATURES OF THE CROSS ASSEMBLER

- 3 types of control data
- Instruction codes and functions easily changed
- Catalogs the control data in disk storage
- Constants can also be expressed in non-decimal notations
- Printouts available from the tables and cross-reference lists
- Execution computer: MELCOM 70 (memory capacity more than 24K words, monitor BDOS)
- Implementation language: FORTRAN IV (parts are written in assembly language)

FEATURES OF THE ASSEMBLY LANGUAGE

- 6 pseudo instructions
- 77 machine instructions
- Decimal numbers can be used to define the constants of the machine instruction operand field.

INPUT/OUTPUT MEDIA

- Source input : Punched cards and magnetic disk
- Control data input : Punched cards and magnetic disk
- Control data command : Punched cards
- Execution command : System typewriter keyboard
- Object output : Magnetic disk
- Output lists : Line printer

FUNCTIONS

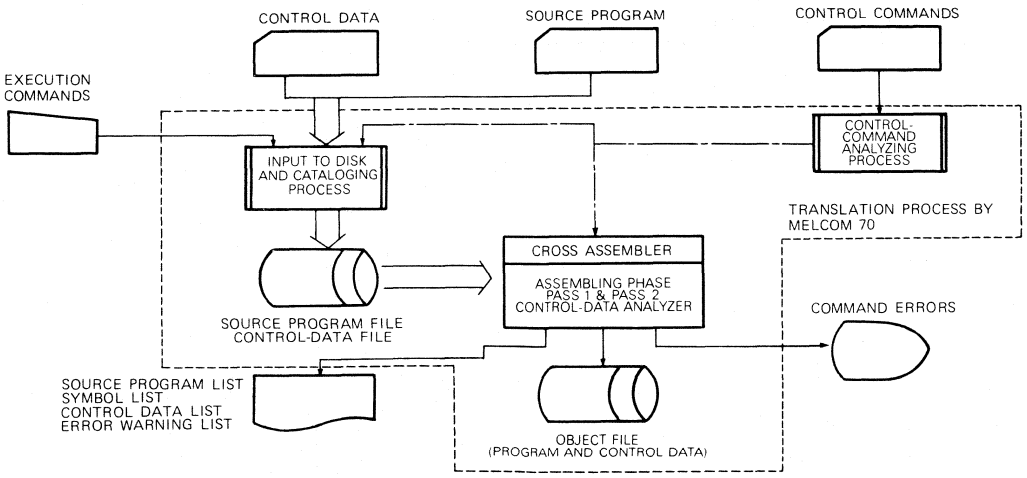
This cross assembler converts source programs written in the MELPS 42 assembly language to machine instruction codes that are filed in disk storage in the form of binary absolute object codes.

The MELPS 42 cross assembler is a 2-pass translator that provides data and control command analysis along with cataloging functions.

Modifying the instruction code and setting mnemonic tables and numeric tables to constants can easily be accomplished by means of the control data. In this way, programming versatility is provided for changing functions, allowing the user free selection in defining the mnemonics of the machine instructions, etc.

The standard version of the MELPS 42 assembly language has 7 assembler control commands (see Table 1). In addition 6 pseudo instructions (Table 2) can be used in the source language program.

PROCESSING SYSTEM



PROGRAM ORDERING INFORMATION

Program name	Ordering number	Program and software manuals included	
MELPS 42 cross assembler	GBIAS0010	MELPS 42 Cross Assembler Manual	GBM-SR00-31A<03A0>

13

CROSS ASSEMBLER

CROSS ASSEMBLER

This cross assembler facilitates assembly by the use of the control commands shown in Table 1. Basically, it requires only the source program and control commands input by punched cards with control data being utilized only when necessary. All input is stored and filed in disk storage. The control data is processed by the control command analyzing processor, and the symbol table is created in pass 1. This is followed by pass 2, where each instruction is converted to machine language, while control data, labels and the assembly list are printed out as specified by the control commands. On the assembly list, the control commands, sequence numbers, location numbers and addresses are printed out, along with error and warning messages, followed by the ROM page list and the cross-reference list.

OBJECT LANGUAGE

The object file is composed of a name section and a text section.

The name section is filed on sector 0 of the object file and stores overall information such as the total number of instructions in the text section, control data, file name, source program file name, size of a single page and the module name.

The text section contains the data that controlled the conversion of the source program to instruction codes and other related data necessary for execution by the simulator.

ASSEMBLY LANGUAGE

The assembly language that the MELPS 42 cross assembler accepts consists of machine instructions and pseudo instructions.

1. Machine Instructions

There are 77 basic machine instructions. These are converted to their corresponding machine codes and then assembled into an object program. For the mnemonics, instruction codes and their functional descriptions, please refer to the data sheet provided for the M58496-XXXP single-chip 4-bit microcomputer.

2. Pseudo Instructions

Although the pseudo instructions are written in the source program together with machine instructions, they are not converted to instruction codes but are used to control the assembler. The instruction codes will be written in the ROM.

The assembler-control instructions, numeric symbols defining instructions and list control instructions are among the pseudo instructions.

The pseudo instructions are shown in Table 2.

Table 1 Assembler control commands

Command		Format	Function
Execution start		/ / / **RUN**	Starts execution of the cross assembler
Execution end		/ / / **END**	Terminates execution of the cross assembler
Input/output function assignment		/ / / **ASMB4, x, y, z**	Assignment of assembly execution and control data and assembly listings x = (A / P) y = (L / N) z = (L / N) **x** : Assembly control **A** : Assembly needed **P** : Designation of cataloging function **y** : Assembly listing **z** : Control data listing **L** : Listing needed **N** : No listing needed
File assignment control	Control data	/ / / **CDISK, XXXXX**	Assignment of the control file name (max. 6 characters)
	Source program	/ / / **SDISK, XXXXX**	Assignment of the source program file name (max. 6 characters)
	Object	/ / / **BDISK, XXXXX**	Assignment of the object file name (max. 6 characters)
Input/output device assignment		/ / / **INPUT, x, y**	Assignment of input device for the control data and source program x = (C / D / N) y = (C / D / N) **x** : Control data input **y** : Source program input **C** : Punched card input **D** : Disk input **N** : Control data no input

Table 2 Pseudo instructions

Classification	Mnemonic	Instruction	Function
Assembler control instruction	TTL	Program title declaration	Declares the program title
	PAGE	Program counter paging	Sets the counter to the top address of the next page
	ORG	Program counter setting	Sets the counter to the top address of the program
	END	End declaration	Declares the end of the program
Symbol value equivalence instruction	EQU	Symbol value setting	Sets a numeral value to the specific numeral symbol
List control instruction	EJE	Page eject declaration	Advances the printout form to the next page during output

3. Language Format

The following format should be used in coding programs in this cross assembler.

The single-line statement is composed of the label, instruction, operand, comment, and identification fields. The format of the source statement is fixed as indicated in Fig. 1. Although the constant is usually a decimal number, it may be expressed by hexadecimal notation when defined by pseudo instructions.

An asterisk (*) in the first column of a line indicates that the entire statement is used as a comment field.

The following are valid characters for use in statements:

Alphabetics: A~Z

Numberics: 0~9

Special characters: ; = , ▾ @ $ + − * / ! & ( ) .
% < > ? (space)

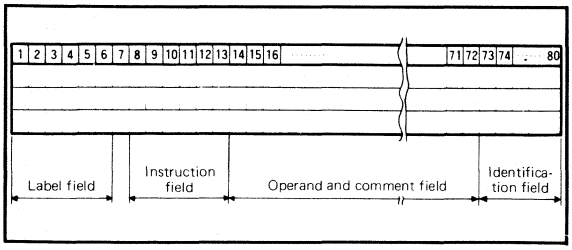


Fig. 1 Source statement format

(1) Label field

The value of the program counter at that time is set to the label. The number of characters used for a label is limited to a maximum of 6, and any of the alphanumerics and special characters specified above can be used. However, an asterisk (*) cannot be used in the first column of the label field.

(2) Instruction field

Mnemonic codes are written in this field, left-justified. For pseudo instructions, any of the mnemonics among the assembler-control instructions, numeric symbol definition instructions and list-control instructions may be used.

(3) Operand field

Parameters of the instruction are specified in this field. This field contains the label, defined symbol, or numerical value. The operand is stated from the 14th column, left-justified.

(4) Comment field

Whenever the operand is followed by more than one space to the end of the statement, the successive columns may be used for comments.

(5) Identification field

The use of this field is optional. Many find it convenient to use this field for a sequential identification card number.

CROSS ASSEMBLER

ASSEMBLY LIST FORMAT

A source program prepared and assembled in the format indicated in the preceding paragraph may produce source, symbol table, cross reference, and ROM page list printouts. The format of an assembly list produced as an example is shown in Fig. 2. Please note that pages, locations, and object codes are indicated in hexadecimal notation.

MESSAGE FORMAT

Error and warning messages are printed out on the assembly list. In the case of errors, the message is printed out under the respective statement in the following format.

$$$$$$$ERROR xxx$

where "x x x" indicates the type of error by a numerical code.

In the case of warnings, the following message is printed between SEQ (sequential number) and LOC (location number):

* **Wx** * (where "**x**" indicates the degree of warning)

In addition the total number of errors and warnings are printed on the last line of the assembly list. The cross-reference list, however, will not be produced when any errors are indicated.

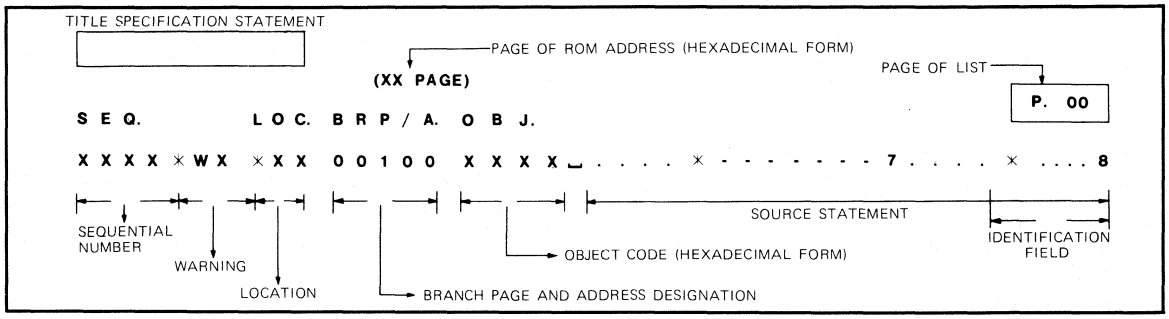


Fig. 2 Assembly list format

Example of an assembly list

An actual example of an assembly list for an assembly made with the MELPS 42 cross assembler is shown in Fig. 3.

```
EXAMPLE PROGRAM                                                          P. 1
                        ( 00 PAGE )
SEQ.  LOC.BRP/A. OBJ. ....*....1....*....2....*SOURCE STATEMENT....*....5....*....6....*....7....*....8

  1                       TTL   EXAMPLE PROGRAM ------------------------ (1) EXA00010
  2                       ORG   0,0 ------------------------------------ (2) EXA00020
  3                 * FILE DATA EXCHANGE ------------------------------- (3) EXA00030
  4                 *                                                     EXA00040
  5                 DIGMAX EQU   13          DIGMAX=13 ----------------- (4) EXA00050
  7    01 0E/00  100        BM    XCG02       EXCHANGE F0 & F2 ---------- (5) EXA00070
  8    02 3E/01  101        BM    XCG13       EXCHANGE F1 & F3 ---------- (6) EXA00080
  9    03 9E/07  107        BM    XCG23       EXCHANGE F2 & F3 ---------- (7) EXA00090
 10    04        000        NOP
 11                 *                                                     (8) EXA00100

 12                       ORG   E,0                                        EXA00110
 13                 * SUBROUTINE FILE EXCHANGE                            EXA00120
 14                 *                                                     EXA00130
 15                 *     EXCHANGE FILE M(2,X,0-DIGMAX)                   EXA00140
 16                 *                                                     EXA00150
 17    00        0CD  XCG02  LXY   0,DIGMAX    EXCHANGE F0 (0-DIGMAX) & F2(0-DIGMAX) (9) EXA00160
 18    01        0DD  XCG13  LXY   1,DIGMAX    EXCHANGE F1 (0-DIGMAX) & F3(0-DIGMAX) (10) EXA00170
 19                 *                                                     EXA00180
 20    02        066  LBL4   TAM   2                                       EXA00190
 21    03        062        XAM   2                                       EXA00200
 22    04        068        XAMD  0                                       EXA00210
 23*WO*05        102        BM    LBL4        BM IS EQUIVALENT WITH B ON PAGE 14 (11) EXA00220
 24    06        044        RT                                            EXA00230
 25                 *                                                     EXA00930
 26    07        0ED  XCG23  LXY   2,13        EXCHANGE F2 (0-DIGMAX) & F3(0-DIGMAX) EXA01010
 27                 *     COMMON ROUTINE START                            EXA01020
 28                 *                                                     EXA01030
 29    08        065  LBL5   TAM   1                                       EXA01040
 30    09        061        XAM   1                                       EXA01050
 31    0A        068        XAMD  0                                       EXA01060
 32*WO*0B        108        BM    LBL5                                    EXA01070
 33    0C        044        RT                                            EXA01080
 34                       END                                             EXA01100
```

① The program name is declared as "EXAMPLE PROGRAM"
② It shows that the start of the program was set to page 0 address 0 by means of the program counter setting instruction.
③ An asterisk (*) in the first column indicates that the entire statement is a comment.
④ Numeric value 13 (decimal number) is assigned to the symbol DIGMAX by means of the symbol value equivalence instruction.
⑤ The label XCG02 is assigned by means of the BM instruction during the assembly process, and calls the subroutine starting at page 14 address 00.
⑥ The label XCG13 is assigned by means of the BM instruction during the assembly process, and calls the subroutine starting at page 14 address 01.
⑦ The label XCG23 is assigned by means of the BM instruction during the assembly process, and calls the subroutine starting at page 14 address 07.
⑧ This whole statement line is used as a comment field.
⑨ The numerical value 0 is loaded in register X of the data pointer and 13 (decimal number) in register Y by means of the LXY instruction. As written, the results of this LXY instruction are nullified by the results of the following LXY instruction.
⑩ The numerical value 1 is loaded in register X of the data pointer and 13 (decimal number) in register Y by means of the LXY instruction.
⑪ The BM instruction in this case assigns the branch address of the label LBL4 to address 02 of page 14.

MITSUBISHI ELECTRIC

PAPER-TAPE GENERATION PROGRAM FOR PROM WRITERS

DESCRIPTION

MELPS 42 PROM writer paper-tape generation programs are used to convert the absolute binary object program generated by the MELPS 42 cross assembler into another format that can be used in a PROM writer. The program is output on paper tape in the new format.

With this program, a binary object program can easily be converted to hexadecimal object format that can be programmed directly into a PROM. It can produce paper tapes that meet the requirements of Takeda Riken's and Minato Electronics' PROM writers.

FEATURES

- Outputs the binary object program in the disk storage to paper tape in hexadecimal format
- Paper-tape output can be partitioned with a simple control command
- May be used in conjunction with the MELPS 42 cross assembler
- Execution computer: MELCOM 70 Minicomputer (memory capacity more than 16K words, monitor BDOS)
- Programming language: FORTRAN IV (parts are written in assembly language)

INPUT/OUTPUT MEDIA

- Input : Cartridge disk storage
- Output : Paper tape (ASCII code, even parity)
- Control command input : Through the keyboard of the system typewriter
- Messages : System typewriter printout

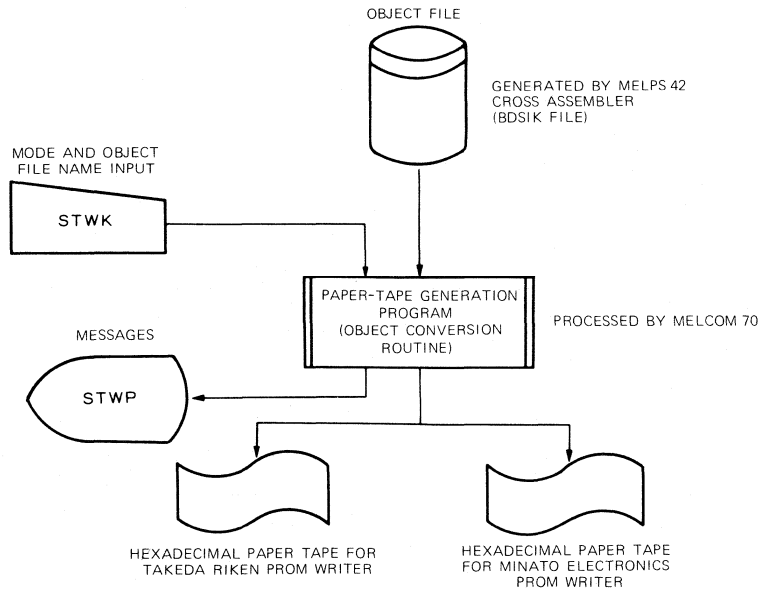
APPLICATIONS

- For preparing programs for 1K words X 8-bit EPROMs (M5L2708K, S), etc., which are to be programmed by PROM writers supplied by Takeda Riken or Minato Electronics.

FUNCTIONS

This program is used for converting the absolute binary object format programs generated by the MELPS 42 cross assembler to hexadecimal object format compatible with the PROM writers manufactured by Takeda Riken (T310) and Minato Electronics (model 1380). The paper-tape output is partitioned in accordance with PROM capacity (number of bytes).

PROCESSING SYSTEM



PROGRAM ORDERING INFORMATION

Program name	Ordering number	Program and software manuals included
MELPS 42 paper-tape generation program for PROM writer	GBISP 0006	MELPS 42 paper-tape generation program for PROM writer manual GBM-SR 00-33A <03AO>

PAPER-TAPE GENERATION PROGRAM FOR PROM WRITERS

PROGRAM PROCESSING

The program has conversion routines for Takeda Riken's and Minato Electronics' PROM writers. Select T_1 mode (for Takeda Riken's PROM writer) or M_1 mode (for Minato Electronics' PROM writer) through the system typewriter keyboard. Then the object program is converted to paper tape compatible with the selected PROM writer. When a (BDISK file) file name is called, a paper tape is output for the PROM writer. When a number of programs are to be converted from the same file, successive calls can be made until all the programs are converted. Termination of the job is directed with the E command, and control is then returned to the monitor.

The object file consists of name and text segments. The data to be converted is contained in the text segment. Instruction codes, stored after sector 1 of the disk, that corresponds to machine instructions are converted to hexadecimal codes and output to paper tape.

Example of Hexadecimal Paper-Tape Format

This program can generate paper tapes for Takeda Riken's PROM writer or Minato Electronics' PROM writer. Examples of both formats are shown in Figs. 1 and 2.

Example of Object Conversion

The program at present can output 1K-word units of paper tape up to a total of 4K words. An example is shown in Fig. 3.

Error Processing

When an error is encountered during object conversion, a message will be printed out in the following format:

$$$$$$$$ xxx$

where, XXX indicates the error code.

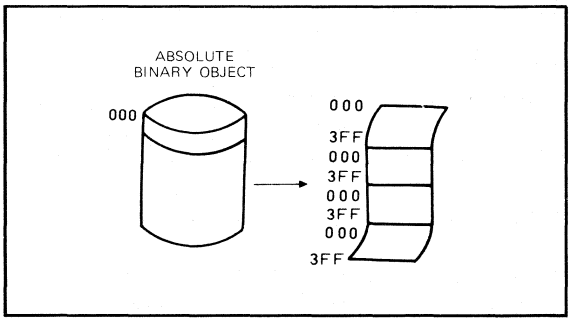


Fig. 3 Example of object conversion

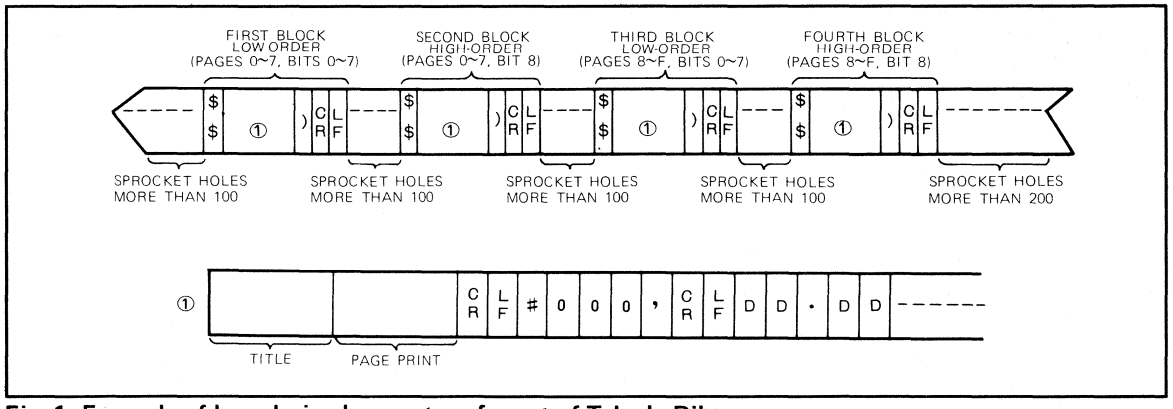


Fig. 1 Example of hexadecimal paper-tape format of Takeda Riken

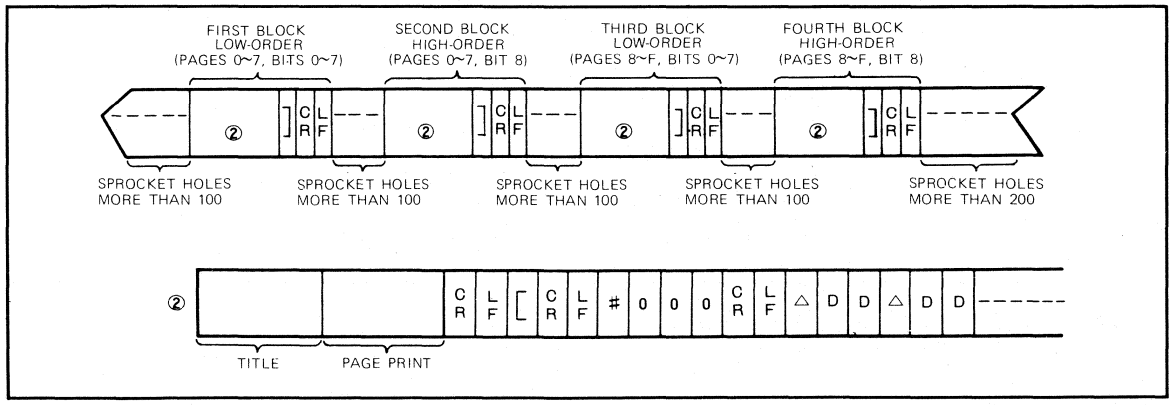


Fig. 2 Example of hexadecimal paper-tape format of Minato Electronics

MITSUBISHI ELECTRIC

CONTACT ADDRESSES FOR FURTHER INFORMATION

JAPAN
Electronics Overseas Division
Mitsubishi Electric Corporation
2-3, Marunouchi 2-chome
Chiyoda-ku, Tokyo 100, Japan
Telex: 24532 MELCO J
Telephone: (03) 218-3473
(03) 218-3499

HONG KONG
Ryoden Electric Engineering Co., Ltd.
22nd fl., Leighton Centre
77, Leighton Rd.
Causeway Bay, Hong Kong
Telex: 73411 RYODEN HX
Telephone: (5) 790-7021

TAIWAN
Mitsubishi Electric Corporation
Taipei Representative Office
Room 1303, 13th fl., Huei Fong Bldg.
27, Sec. 3, Chung Shan N. Road
Taipei, R.O.C.
Telex: 11211 MITSUBISHI
Telephone: (597) 3111

U.S.A.
Mitsubishi Electronics America, Inc.
2200 West Artesia Blvd.
Compton CA 90220, U.S.A.
Telex: 698246 MELA CMTN
Telephone: (213) 979-6055
(213) 979-6371

Mitsubishi Electronics America, Inc.
100 Wade Ave.
South Plainfield, NJ 07080, U.S.A.
Telex: 833244 MELA SOPH
Telephone: (201) 753-1600
(800) 631-5377

Mitsubishi Electronics America, Inc.
1230 Oakmead Parkway
Suite 206 Sunnyvale CA 94086 U.S.A.
Telex: 172296 MELA SUVL
Telephone: (408) 730-5900

WEST GERMANY
Mitsubishi Electric Europe GmbH
Brandenburger Str. 40
4030 Ratingen, West Germany
Telex: 8585070 MED D
Telephone: (02102) 44089~44094

U.K.
Mitsubishi Electric (U.K.) Ltd.
Otterspool Way, Watford, Hertfordshire
WD2 8LD, U.K.
Telex: 927908
Telephone: (923) 40566~9