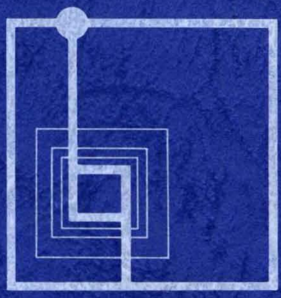
MITSUBISHI DATA BOOK 1984

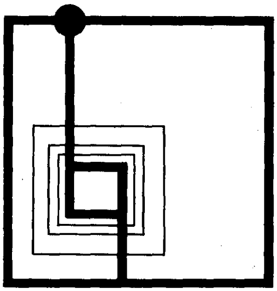
SINGLE-CHIP MICROCOMPUTERS



MITSUBISHI
ELECTRIC

MITSUBISHI DATA BOOK 1984

SINGLE-CHIP MICROCOMPUTERS



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.

CONTENTS

8 APPENDICES

GUIDANCE 1

INDEX BY FUNCTION

Type	Circuit function and organization	Structure	Supply voltage (V)	Electrical characteristics				Package	Interchangeable products	Page
				Typ pwr dissipation (mW)	Max. access time (ns)	Min cycle time (ns)	Max. fre-quency (MHz)			

■MELPS 760 Microcomputers

M50760-XXXP	1K-Byte Mask-Prog. ROM	C,Si	5±10%	2	—	10μs	0.4	20P4	—	2—3
M50761-XXXP	0.5K-Byte Mask-Prog. ROM	C,Si	5±10%	2	—	10μs	0.4	20P4	—	2—3

■MELPS 740 Microcomputers

M50740-XXXSP	3K-Byte Mask-Prog. ROM	C,Si	5±10%	20	—	2μs	4	52P4B	—	3—3

■MELPS 8-48 Microcomputers

M5L8048-XXXP	1K-Byte Mask-Prog. ROM	N,Si,ED	5±10%	325	—	2500	6	40P4	i8048	4—21
M5L8035LP	External ROM Type,64-Byte RAM	N,Si,ED	5±10%	325	—	2500	6	40P4	i8035L	4—21
M5L8049-XXXP	2K-Byte Mask-Prog. ROM	N,Si,ED	5±10%	500	—	1360	11	40P4	i8049	4—25
M5L8049-XXXP-8				500	—	1875	8		—	
M5L8049-XXXP-6				500	—	2500	6		—	
M5L8039P-11	External ROM Type,128-Byte RAM	N,Si,ED	5±10%	500	—	1360	11	40P4	i8039	4—25
M5L8039P-8				500	—	1875	8		—	
M5L8039P-6				500	—	2500	6		i8039-6	
M5L8049H-XXXP ⁑	2K-Byte Mask-Prog. ROM	N,Si,ED	5±10%	250	—	1360	11	40P4	i8049H	4—29
M5L8039HLP ⁑	External ROM Type,128-Byte RAM	N,Si,ED	5±10%	250	—	1360	11	40P4	i8039HL	4—29
M5M80C49-XXXP*	2K-Byte Mask-Prog. ROM	C,Si	5±10%	25	—	2500	6	40P4	—	4—44
M5M80C39P-6 *	External ROM Type,128-Byte RAM	C,Si	5±10%	25	—	2500	6	40P4	—	4—44
M5M8050H-XXXP	4K-Byte Mask-Prog. ROM	N,Si,ED	5±10%	350	—	1360	11	40P4	—	4—34
M5M8040HP *	External ROM Type,256-Byte RAM	N,Si,ED	5±10%	350	—	1360	11	40P4	—	4—34
M5M8050L-XXXP*	4K-Byte Mask-Prog. ROM	N,Si,ED	5±10%	250	—	2500	6	40P4	—	4—39
M5M8040LP *	External ROM Type,256-Byte RAM	N,Si,ED	5±10%	250	—	2500	6	40P4	—	4—39

■MELPS 8-41 Slave Microcomputers

M5L8041A-XXXP	1K-Byte Mask-Prog. ROM	N,Si,ED	5±10%	300	—	2500	6	40P4	i8041A	5—25
M5L8042-XXXP *	2K-Byte Mask-Prog. ROM	N,Si,ED	5±10%	300	—	1250	12	40P4	i8042	5—32

■LSIs for Peripheral Circuits

M50780SP	I/O Expander (CE="H" active)	C,Al	3~14	—	—	—	—	40P4B	TMS1025C	6—3
M50781SP	I/O Expander (CE="H" active)	C,Al	3~14	—	—	—	—	28P4B	TMS1024C	6—3
M50782SP	I/O Expander ($\overline{CE}$="L" active)	C,Al	3~14	—	—	—	—	40P4B	—	6—3
M50783SP	I/O Expander ($\overline{CE}$="L" active)	C,Al	3~14	—	—	—	—	28P4B	—	6—3
M50784SP	Input Expander	C,Al	4~14	—	—	—	—	28P4B	—	6—9
M50786SP	I/O Expander ($\overline{CE}$="L" active)	C,Al	4~14	—	—	—	—	40P4B	—	6—11
M50790SP	I/O Expander	C,Al	4~14	—	—	—	—	52P4B	—	6—16
M5L8243P	I/O Expander	N,Si,ED	5V±10%	—	—	—	—	24P4	i8243	6—26
M5M82C43P *	I/O Expander	C,Si	5V±10%	—	—	—	—	24P4	—	6—32
M5L8155P	2048-Bit Static RAM with I/O Ports and Timer ($\overline{CE}$="L" active)	N,Si,ED	5 ±5%	500	—	—	—	40P4	i8155	6—39
M5L8156P	2048-Bit Static RAM with I/O Ports and Timer (CE="H" active)	N,Si,ED	5 ±5%	500	—	—	—	40P4	i8156	6—47

Note : Al=Aluminum gate. C=CMOS. ED=Enhancement depletion mode. N=N-channel. P=P-channel. Si=Silicon gate.
* : New product ⁑ : Under development

Development Support Systems

Development Support unit Type				Cross-assembler unit		Debugging unit		Evaluation boards
				Hardware	Software	Main unit	Special boards	
	CMOS 4-bit	MELPS 760	M50760-XXXP	PC9000	Cross-assembler software for PC9000	PC4000	PCA4060	PCA4360
			M50761-XXXP					M50760-PGYS
	CMOS 8-bit	MELPS 740	M50740-XXXSP	PC9100	PC9008 ASM		PCA4040	PCA4340
			M50741-XXXSP					M50740-PGYS
		MELPS 8-48	M5M80C49-XXXP	PC9000	Cross-assembler software for PC9000		PCA8400	PCA8403
			M5M80C39P-6					—
	NMOS 8-bit		M5L8048-XXXP					—
			M5L8035LP					—
			M5L8049-XXXP					PCA8403
			M5L8049-XXXP-8					PCA8403
			M5L8049-XXXP-6					PCA8403
			M5L8039P-11					—
			M5L8039P-8					—
			M5L8039P-6					—
			M5M8050H-XXXP					PCA8403
			M5M8050L-XXXP					M5M8050H-PGYS
			M5L8049H-XXXP			—	—	—
			M5L8039HLP			—	—	—
		MELPS 8-41	M5L8041A-XXXP			—	—	—
			M5L8042-XXXP			—	—	—

FUNCTION CODE

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

For Mitsubishi Original Products

Example : M 5 07 60 - 001 P - 2

M : Mitsubishi integrated prefix

Temperature range
5 : Standard industrial/commercial (0 to 70/75℃ or −20 to 86℃).
9 : High reliability

Series designation using 1 or 2 alphanumeric characters.
01～09 : CMOS
1 : Linear circuit
3 : TTL
10～19 : Linear circuit
32～33 : TTL (equivalent to Texas Instruments' SN74 series)
41～47 : TTL
81 : P-channel aluminum-gate MOS
84 : COMS
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 : COMS
9 : DTL
S0～S2 : Schottky TTL (equivalent to Texas Instruments' SN74S series)

Circuit function identification code using 2 digits.

Mask ROM number.

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 L 8041A - 001 P - 1

M : Mitsubishi integrated circuit prefix

Temperature range
5 : Standard industrial/commercial (0 to 70/75℃ or −20 to 85℃)
9 : High reliability

Series designation of original source using 1 or 2 alphabetical characters.
C : Motorola's MC series
G : General Instrument's series
K : Mostek's MK series
L : Intel's series
T : Texas Instruments' TMS series
W : Western Digital's series

Circuit function identification code of the original source type name

Mask ROM number

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

Electrical characteristic identification code using 1 or 2 digits.

PACKAGE CODE

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

Example : 42 P 4 B

Number of pins

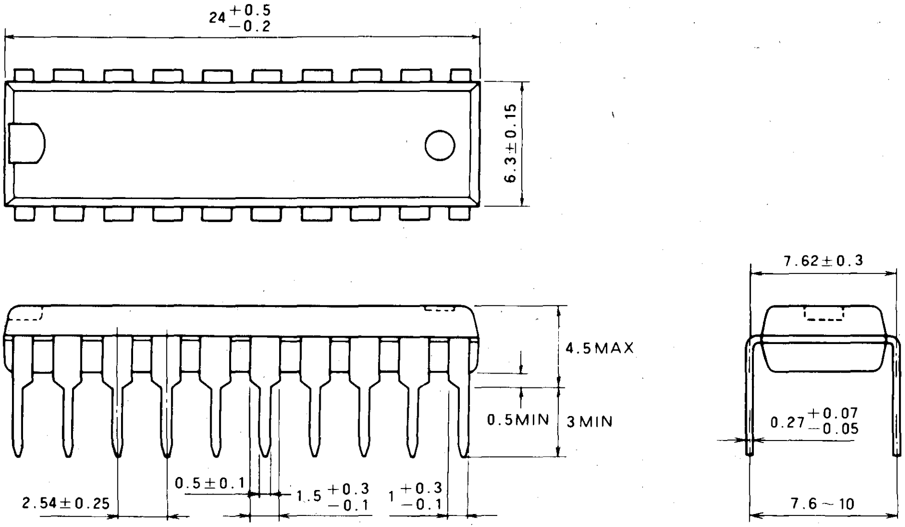
Package structure
K : Glass-sealed ceramic
P : Molded plastic
S : Metal-sealed ceramic

Package outline
4 : DIL without fin (improved)
10 : DIL without fin and with quartz lid

Other package outline
B : Shrink DIL package

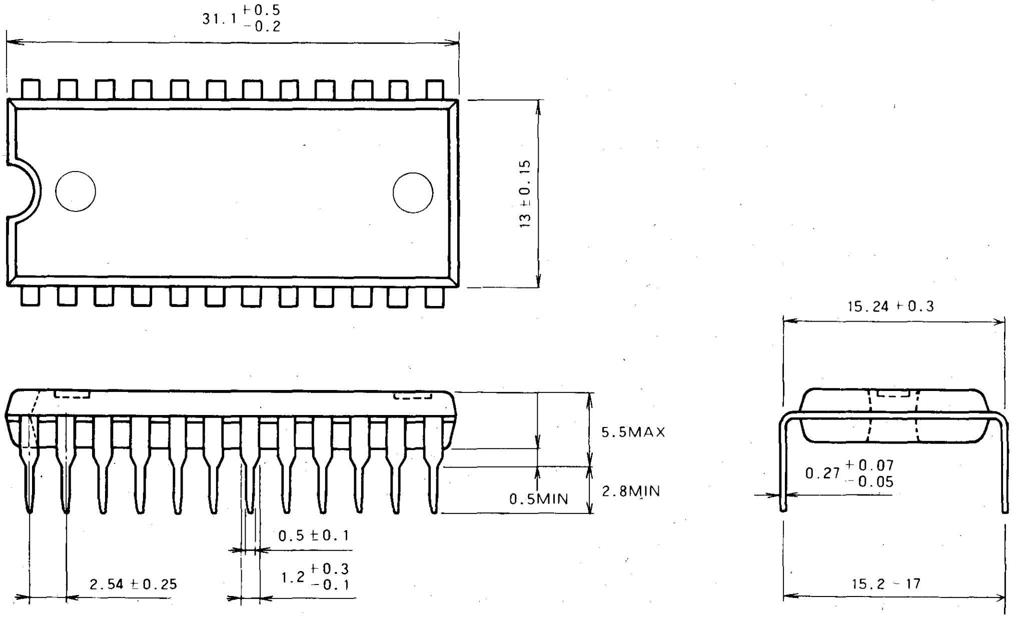
TYPE 20P4 20-PIN MOLDED PLASTIC DIL

Dimension in mm



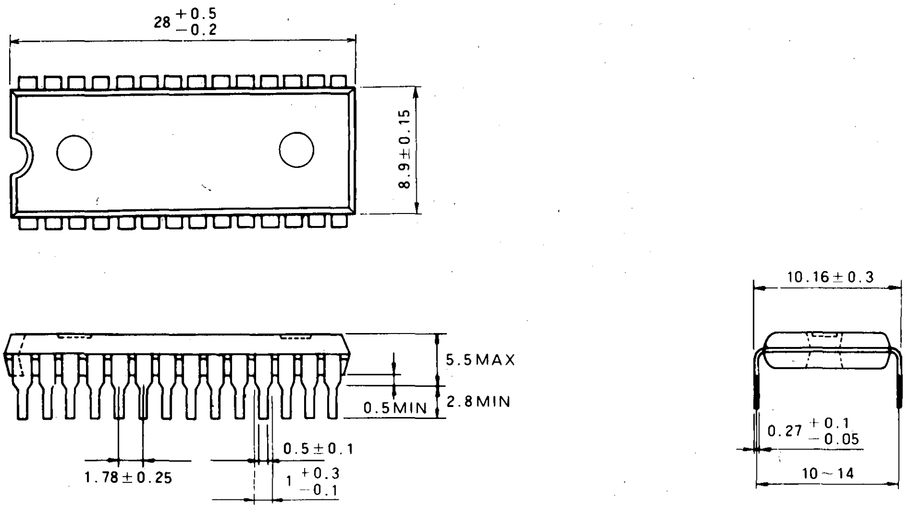
TYPE 24P4 24-PIN MOLDED PLASTIC DIL

Dimension in mm



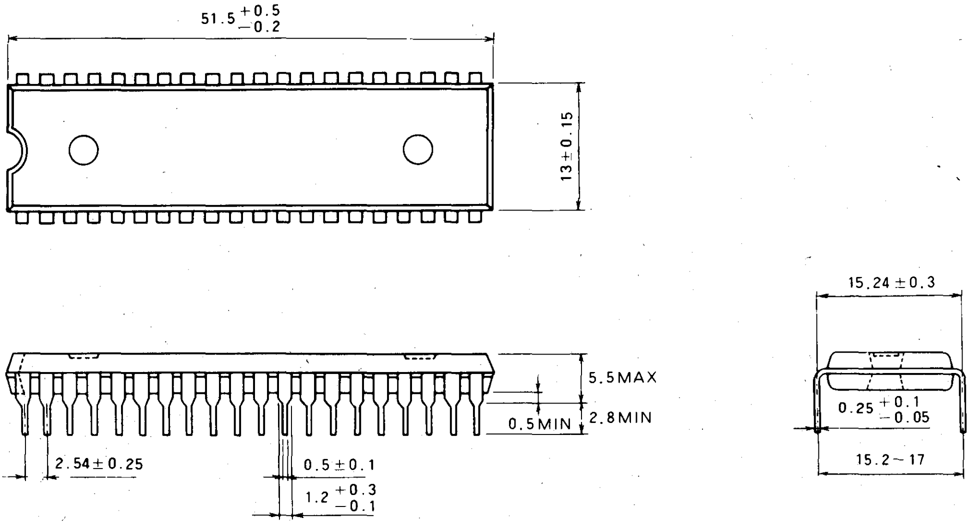
TYPE 28P4B 28-PIN MOLDED PLASTIC DIL(LEAD PITCH 1.78mm)

Dimension in mm



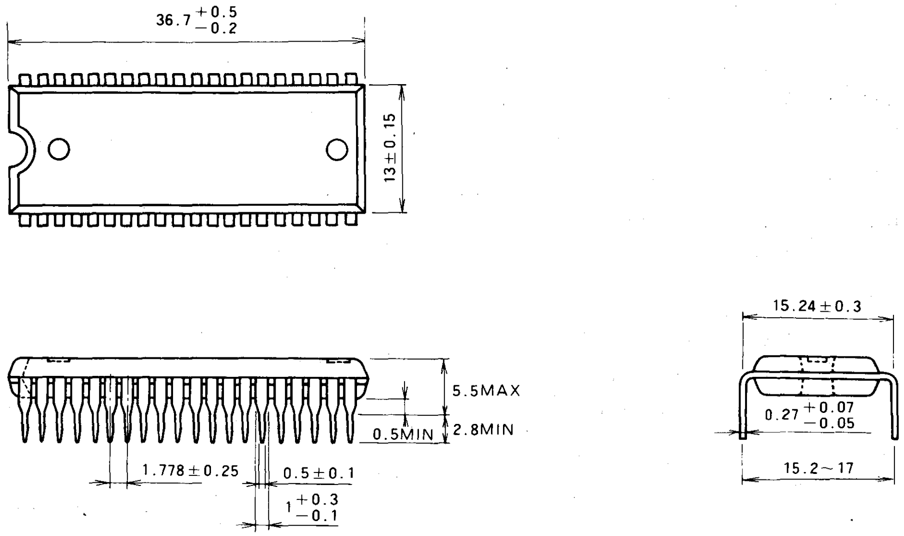
TYPE 40P4 40-PIN MOLDED PLASTIC DIL

Dimension in mm



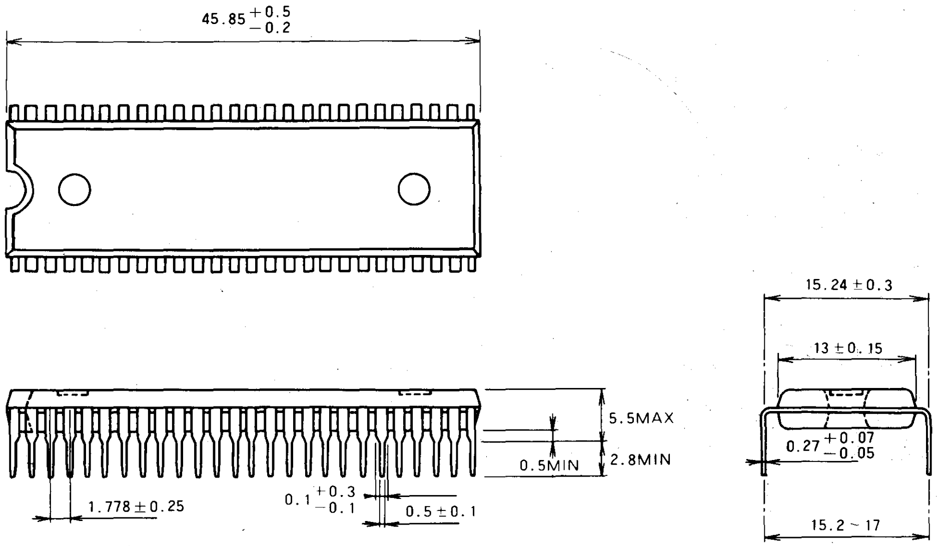
TYPE 40P4B 40-PIN MOLDED PLASTIC DIL (LEAD PITCH 1.78mm)

Dimension in mm



TYPE 52P4B 52-PIN MOLDED PLASTIC DIL (LEAD PITCH 1.78mm)

Dimension in mm



LETTER SYMBOLS FOR THE DYNAMIC PARAMETERS

1. INTRODUCTION

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

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

2. LETTER SYMBOLS

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

2.1. General Form

The dynamic parameters are represented by 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.

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

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

6. SUBSCRIPT F (For Additional Information)

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

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

FOR DIGITAL INTEGRATED CIRCUITS

New symbol	Former symbol	Parameter—definition
C_i		Input capacitance
C_o		Output capacitance
$C_{i/o}$		Input/output terminal capacitance
$C_{i(\phi)}$		Input capacitance of clock input
f		Frequency
$f_{(\phi)}$		Clock frequency
I		Current—the current into an integrated circuit terminal is defined as a positive value and the current out of a terminal is defined as a negative value
I_{BB}		Supply current from V_{BB}
$I_{BB(AV)}$		Average supply current from V_{BB}
I_{CC}		Supply current from 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 capabilty 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

SYMBOLOGY

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
$t_{d(CAS-RAS)}$		Delay time, column address strobe to row address strobe
$t_{d(CAS-W)}$	$t_{d(CAS-WR)}$	Delay time, column address strobe to write
$t_{d(RAS-CAS)}$		Delay time, row address strobe to column address strobe
$t_{d(RAS-W)}$	$t_{d(RAS-WR)}$	Delay time, row address strobe to write
$t_{dis(R-Q)}$	$t_{dis(R-DA)}$	Output disable time after read
$t_{dis(S)}$	$t_{PXZ(CS)}$	Output disable time after chip select
$t_{dis(W)}$	$t_{PXZ(WR)}$	Output disable time after write
t_{DHL}		High-level to low-level delay time } the time interval between specified reference points on the input and on the output pulses, when the output is going to the low (high) level and when the device is driven and loaded by specified networks.
t_{DLH}		Low-level to high-level delay time
$t_{en(A-Q)}$	$t_{PZV(A-DQ)}$	Output enable time after address
$t_{en(R-Q)}$	$t_{PZV(R-DQ)}$	Output enable time after read
$t_{en(S-Q)}$	$t_{PZX(CS-DQ)}$	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
$t_{h(A-E)}$	$t_{h(AD-CE)}$	Chip enable hold time after address
$t_{h(A-PR)}$	$t_{h(AD-PRO)}$	Program hold time after address
$t_{h(CAS-CA)}$		Column address hold time after column address strobe
$t_{h(CAS-D)}$	$t_{h(CAS-DA)}$	Data-in hold time after column address strobe
$t_{h(CAS-Q)}$	$t_{h(CAS-OUT)}$	Data-out hold time after column address strobe
$t_{h(CAS-RAS)}$		Row address strobe hold time after column address strobe
$t_{h(CAS-W)}$	$t_{h(CAS-WR)}$	Write hold time after column address strobe
$t_{h(D)}$	$t_{h(DA)}$	Data-in hold time
$t_{h(D-PR)}$	$t_{h(DA-PRO)}$	Program hold time after data-in
$t_{h(E)}$	$t_{h(CE)}$	Chip enable hold time
$t_{h(E-D)}$	$t_{h(CE-DA)}$	Data-in hold time after chip enable
$t_{h(E-G)}$	$t_{h(CE-OE)}$	Output enable hold time after chip enable
$t_{h(R)}$	$t_{h(RD)}$	Read hold time
$t_{h(RAS-CA)}$		Column address hold time after row address strobe
$t_{h(RAS-CAS)}$		Column address strobe hold time after row address strobe
$t_{h(RAS-D)}$	$t_{h(RAS-DA)}$	Data-in hold time after row address strobe
$t_{h(RAS-W)}$	$t_{h(RAS-WR)}$	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
$t_{h(W-CAS)}$	$t_{h(WR-CAS)}$	Column address strobe hold time after write
$t_{h(W-D)}$	$t_{h(WR-DA)}$	Data-in hold time after write
$t_{h(W-RAS)}$	$t_{h(WR-RAS)}$	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
$t_{su(A-E)}$	$t_{su(AD-CE)}$	Chip enable setup time before address
$t_{su(A-W)}$	$t_{su(AD-WR)}$	Write setup time before address
$t_{su(CA-RAS)}$		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

QUALITY ASSURANCE AND RELIABILITY TESTING

1. PLANNING

In recent years, advances in integrated circuits have been rapid, with increasing density and speed accompanied by decreasing cost. Because of these advances, it is now practical and economically justifiable to use these devices in systems of greater complexity and in which they were previously considered too expensive. All of these advances add up to increased demand.

We at Mitsubishi foresaw this increased demand and organized our production facilities to meet it. We also realized that simply increasing production to meet the demand was not enough and that positive steps would have to be taken to assure the reliability of our products.

This realization resulted in development of our Quality Assurance System. The system has resulted in improved products, and Mitsubishi is able to supply its customers' needs with ICs of high reliability and stable quality. This system is the key to future planning for improved design, production and quality assurance.

2. QUALITY ASSURANCE SYSTEM

The Quality Assurance System imposes quality controls on Mitsubishi products from the initial conception of a new product to the final delivery of the product to the customer. A diagram of the total system is shown in Fig. 1. For ease of understanding, the system is divided into three stages.

2.1 Quality Assurance in the Design Stage

The characteristics of the breadboard devices are carefully checked to assure that all specifications are met. Standard integrated circuits and high-quality discrete components are used. During the design stage, extensive use is made of a sophisticated CAD program, which is updated to always include the latest state-of-the-art techniques.

2.2 Quality Assurance in the Limited-Manufacturing Stage

Rigid controls are maintained on the environment, incoming material and manufacturing equipment such as tools and test equipment. The products and materials used are subjected to stringent tests and inspections as they are manufactured. Wafer production is closely monitored.

Finally, a tough quality assurance test and inspection is made before the product is released for delivery to a customer. This final test includes a complete visual inspection and electrical characteristics tests. A sampling technique is used to conduct tests under severe operating conditions to assure that the products meet reliability specifications.

2.3 Quality Assurance in the Full Production Stage

Full production of a product is not started until it has been confirmed that it can be manufactured to meet quality and reliability specifications. The controls, tests and inspection procedures developed in §2.2 are continued. The closest monitoring assures that they are complied with.

3. RELIABILITY CONTROL

3.1 Reliability Tests

The newly established Reliability Center for Electronic Components of Japan has established a qualification system for electronic components. Reliability test methods and procedures are developed to mainly meet MIL-STD-883 and JIS C 7022 specifications. Details of typical tests used on Mitsubishi ICs are shown in Table 1.

Table 1 Typical reliability test items and conditions

Group	Item	Test condition
1	High temperature operating life	Maximum operating ambienj temperature 1000h
	High temperature storage life	Maximum storage temperature 1000h
	Humidity (steady state) life	65°C 95%RH 500h
2	Soldering heat	260°C 10s
	Thermal shock	0~100°C 15 cycles, 10min/cycle
	Temperature cycle	Minimum to maximum storage temperature, 10 cycles of 1h/cycle
3	Soldering	230°C, 5s, use rosin flux
	Lead integrity	Tension: 340g 30s Bending stress: 225g, ±30°, 3 times
	Vibration	20G, X, Y, Z each direction, 4 times 100~2000Hz–4 min/cycle
	Shock	1500G, 0.5ms in X_1, Y_1 and Z_1 direction, 5 times.
	Constant acceleration	20000G, Y_1 direction, 1 min

3.2 Failure Analysis

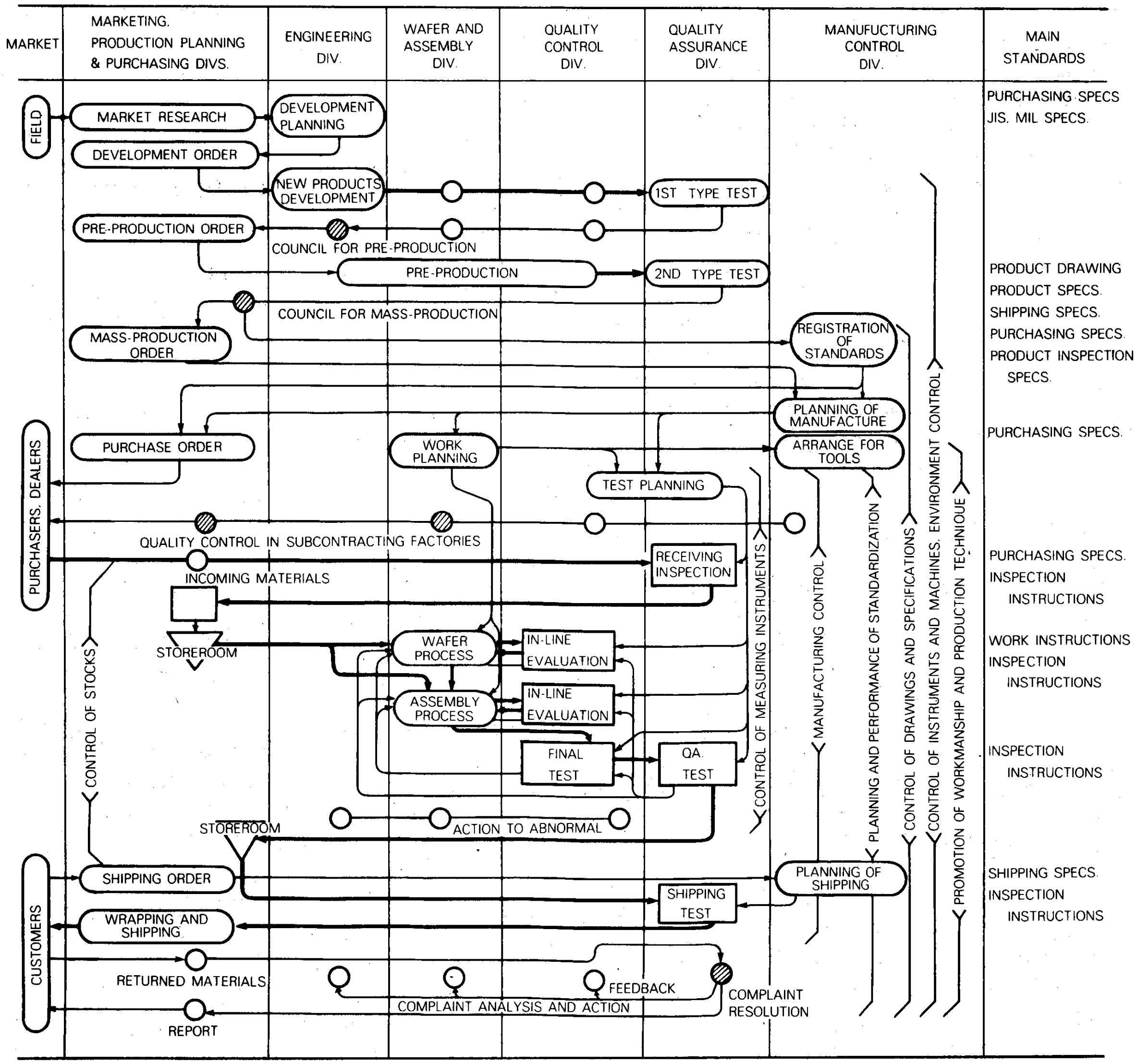
Devices that have failed during reliability or acceleration tests are analyzed to determine the cause of failure. This information is fed back to the process engineering section and manufacturing section so that improvements can be made to increase reliability. A summary of failure analysis procedures is shown in Table 2.

Table 2 Summary of failure analysis procedures

Step	Description
1. External examination	○ Inspection of leads, plating, soldering and welding ○ Inspection of materials, sealing, package and marking ○ Visual inspection of other items of the specifications ○ Use of stereo microscopes, metallurgical microscopes, X-ray photographic equipment, fine leakage and gross leakage testers in the examination
2. Electrical tests	○ Checking for open circuits, short circuits and parametric degradation by electrical parameter measurement ○ Observation of characteristics by a synchroscope or a curve tracer and checking of important physical characteristics by electrical characteristics ○ Stress tests such as environmental or life tests, if required
3. Internal examination	○ Removal of the cover of the device, the optical inspection of the internal structure of the device ○ Checking of the silicon chip surface ○ Measurement of electrical characteristics by probes, if applicable ○ Use of SEM, XMA and infrared microscanner if required
4. Chip analysis	○ Use of metallurgical analysis techniques to supplement analysis of the internal examination ○ Slicing for cross-sectional inspection ○ Analysis of oxide film defects ○ Analysis of diffusion defects

QUALITY ASSURANCE AND RELIABILITY TESTING

Fig. 1 Quality assurance system



▨ MAIN DIVISION ○ CONCERNED DIVISION ▬▶ FLOW OF MATERIALS, PARTS, AND PRODUCTS ⟶ FLOW OF INFORMATION

PRECAUTIONS IN HANDLING MOS IC/LSIs

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

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

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

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

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

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

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

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

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

4. PRECAUTIONS FOR MOUNTING OF MOS IC/LSIs

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

MELPS 760 MICROCOMPUTERS

M50760-XXXP/M50761-XXXP

SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

DESCRIPTION

The M50760-XXXP and M50761-XXXP are single-chip 4-bit microcomputers fabricated using CMOS technology. They come in a 20-pin plastic molded DIL package.

Differences between M50760-XXXP and M50761-XXXP.

Name	ROM capacity	RAM capacity
M50760-XXXP	1024 word	48 word
M50761-XXXP	512 word	32 word

The details given below relate to the M50760-XXXP.

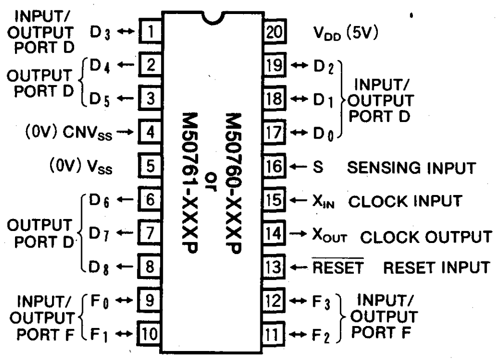
FEATURES

- Basic machine instructions ············ 37
- Instruction execution time (1-word instruction at a clock frequency of 400kHz) ············ 10μs
- Memory capacity ROM ············ 1024 words × 8 bits
  RAM ············ 48 words × 4 bits
- Single 5V power supply
- Timer (7-bit timer)
- Input/output ports (port F) ············ 4
- Output ports (port D) ············ 9 (including 4 I/O ports)
- Sensing input (port S)

APPLICATIONS

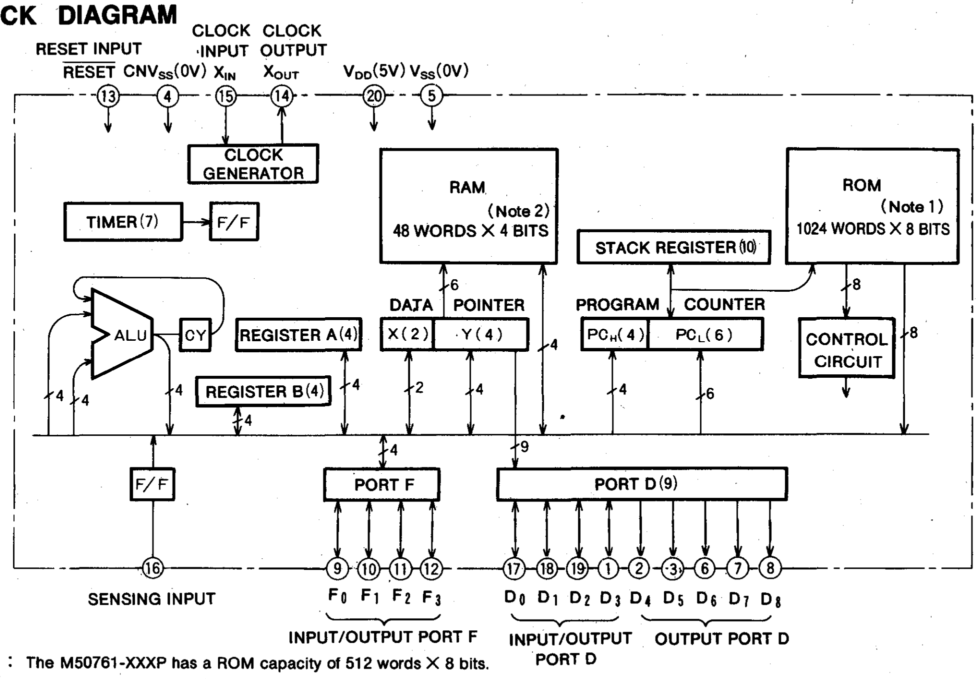
- VTRs, audio equipment and TVs
- Air conditioners, refrigerators, rice cookers
- Remote-controlled receivers/transmitters
- Electronic toys
- Input/output circuits as sub-microcomputers

PIN CONFIGURATION (TOP VIEW)



Outline 20P4

BLOCK DIAGRAM



Note 1 : The M50761-XXXP has a ROM capacity of 512 words × 8 bits.
2 : The M50761-XXXP has a RAM capacity of 32 words × 4 bits.

PERFORMANCE SPECIFICATIONS

Parameter			Performance	
			M50760-XXXP	M50761-XXXP
Number of basic instructions			37	37
Execution time of basic instruction (1-word instruction)			10μs (with 400kHz clock frequency)	10μs (with 400kHz clock frequency)
Clock frequency			200kHz~400kHz	200kHz~400kHz
Memory capacity	ROM		1024 words × 8 bits	512 words × 8 bits
	RAM		48 words × 4 bits	32 words × 4 bits
Input/output ports (14 pins)	F	Input	4 bits × 1	4 bits × 1
		Output	4 bits × 1	4 bits × 1
	D	Output (input)	1 bit × 9 (input 4 bits × 1)	1 bit × 9 (input 4 bits × 1)
	S	Input	1 bit × 1	1 bit × 1
Timer			7-bit timer	7-bit timer
Subroutine nesting			1 level	1 level
Clock generator			Built-in (Externally connected resistor or ceramic resonator)	Built-in (Externally connected resistor or ceramic resonator)
Port output characteristics	Port D_4~D_8		Output current 12mA (n-channel open drain)	Output current 12mA (n-channel open drain)
	Port D_0~D_3		Output current 5mA (n-channel open drain)	Output current 5mA (n-channel open drain)
	Port F		Output current 5mA (n-channel open drain)	Output current 5mA (n-channel open drain)
Supply voltage			5V (typ)	5V (typ)
Device structure			CMOS silicon gate	CMOS silicon gate
Package			20-pin plastic molded DIL package	20-pin plastic molded DIL package
Power dissipation (excluding ports)			2mW (typ)	2mW (typ)

PIN DESCRIPTION

Pin	Name	Input or output	Function
V_{DD}	Supply voltage	In	Positive power supply pin.
V_{SS}	Ground	In	Ground pin.
F_0~F_3	Input/output port F	In/out	Port F is a 4-bit output-latched input/output port. N-channel transistor open-drain circuits are featured for the outputs. When the port F output latch is programmed to (1), the output floats (High-impedance state), thereby enabling use of the port for input.
D_0~D_3	Input/output port D	In/out	Port D consists of 9 bits, each of which is individually latched. N-channel transistor open-drain circuits are featured for the outputs. Port D_0—D_3 pins have a 4-bit input function and when the output latch is programmed to (1), the output floats (high-impedance state) thereby enabling use of the port for input.
D_4~D_8		Out	
S	Sensing input S	In	This pin has an active rising edge. When the S pin signal changes from low to high, the flag is set (1). Not only in case that the flag is set "(1)", whennever you want to, you can test it. This enables testing and flag clearing using an instruction. You can test and clear it by using an instruction. The pin can be modified to a level active input pin with a mask option.
X_{IN}	Clock input	In	These are the clock input and output pins to which an external resistor is connected for RC oscillation of the clock generator or a ceramic resonator is connected. When an external clock is used, connect the source to X_{IN} and leave X_{OUT} open.
X_{OUT}	Clock output	Out	
$\overline{RESET}$	Reset input	In	The device is reset when a low-level singnal is applied for 2 or more machine cycles.
CNV_{SS}	CNV_{SS} input	In	This pin is connected to V_{SS} and must have a low-level input applied to it (0V).

BASIC FUNCTION BLOCKS

Program Memory (ROM)

This 1024-word × 8-bit mark programmable ROM can be programmed with machine instruction codes in accordance with the customer's specifications.

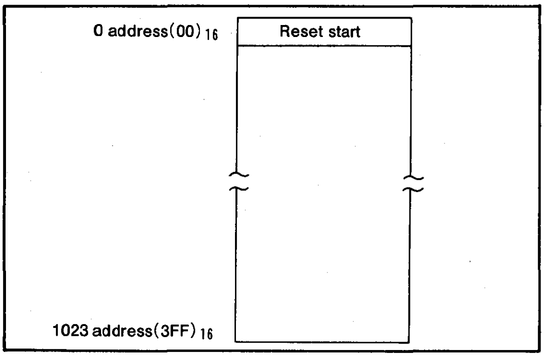


Fig.1 ROM address map

Program Counter (PC)

This counter is to specify ROM addresses and the sequence of read-out of instructions stored in the ROM. It is a 10-bit polynomial counter.

p_3	p_2	p_1	p_0	a_5	a_4	a_3	a_2	a_1	a_0

Fig.2 Program counter configuration

Note : 10-bit configuration is also featured for the program counter in the M50761-XXXP.

Stack Register (SK)

This is used to temporarily store the contents of the PC while executing subroutines until the program returns to its main routine.

Data Memory (RAM)

This 48-word × 4-bit (192-bit) RAM is used to store both processing and control data. One RAM word consists of 4 bits with bit manipulation possible over the entire storage area.

X	0				1				2				3			
Y \ Bit	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0
0																
1																
⋮																
10																
11																

Fig.3 RAM address map

Note : Y = 0~7 for the M50761-XXXP.

Data Pointer (DP)

These registers are used to designate RAM addresses and bit positions for output port D. Register X (the 2 most significant bits of the DP) designates the RAM file: register Y (the 4 least significant bits) designates the digit position of the RAM file.

4-Bit Arithmetic Logic Unit

This unit executes 4-bit arithmetic and logic operations. It performs subtraction, addition and logical comparisons.

Register A and Carry Flag CY

Register A is a 4-bit accumulator that constitutes the basis for arithmetic operations. The carry flag CY is used to store carry or overflow after execution of arithmetic and logical operations by the arithmetic unit. It may also be used as a 1-bit flag.

Register B

Register B is composed of 4 bits and can be used as a 4-bit temporary storage register.

Timer

The timer is implemented using a 7-bit counter which, after release of reset, starts counting and divides the machine cycle by 100. It also sets the flag. Counting starts again after the flag has been set.

The skip instruction (SNZT) can be used to test the flag. Both the timer and the flag can be reset using the system reset and reset instruction (RSTM).

Input/Output Ports

(1) Port F

This 4-bit port is controlled for output and input by the OFA and IAF instructions respectively. When using a bit for input, that bit output latch must first be set to (1) to achieve the high-impedance mode.

(2) Port D

This port consists of 9 bits which can be used for both output and input functions by using the SD/RD and IAD instructions respectively. The output section provides individual bit latching and the contents of register Y can be used to designate a single bit of port D for output.

$D_0 \sim D_3$ pins have a 4-bit input function. When using the port for input, the $D_0 \sim D_3$ output latch must first be set to (1) to achieve the high-impedance mode.

The outputs are n-channel open-drain circuits.

(3) Port S

This is a rising edge active sensing port. The flag is set (1) when the S pin signal changes from low to high. The skip instruction (SNZS) may be used for flag testing, and the flag is reset by the execution of this instruction. The flag is also reset with system resetting.

The port can also be used as a level active input pin with a mask option, in which case the flag (S) is ineffective.

Pin S can be tested using the skip instruction (SNZS) and skipping occurs when the pin is in the high-level mode.

Reset

When the $\overline{\text{RESET}}$ pin is kept low for at least 2 machine cycles, the reset state is enabled. After resetting,when the input is driven high, the program execution will begin at address 0.

When the reset state is enable, the following operations are performed.

(1) The program counter is set to address 0.
(2) The two flags (timer flag, sensing input flag) are reset.
(3) All output latchs of port D are set to (1) (high impedance)
(4) All output latchs of port F are set to (1) (high impedance)

Fig.4 shows an example of the power-on resetting circuit

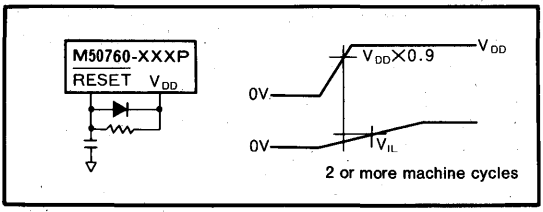


Fig.4 Power-on resetting circuit

Clock Generator Circuit

A clock generator circuit has been built in to allow control of the frequency by means of an externally connected resistor or ceramic resonator. In addition, an external clock signal may be applied at the X_{IN} pin, leaving the X_{OUT} pin open. External connection of the resister or ceramic resonater should be specified as a mask option.

Circuit examples are shown in Fig.5

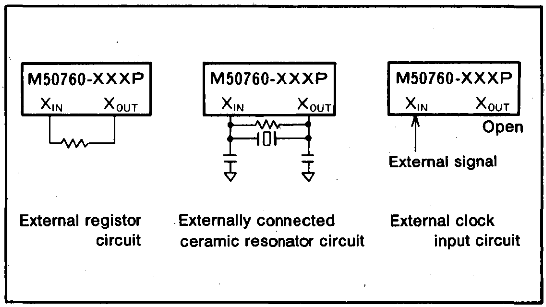

M50760-XXXP
X_IN X_OUT
M50760-XXXP
X_IN X_OUT
M50760-XXXP
X_IN X_OUT
Open
External signal
External registor circuit
Externally connected ceramic resonator circuit
External clock input circuit

Fig.5 Clock generator circuits

Documentation Required upon Ordering

The following information should be provided when ordering a custom mask.

(1) M50760-XXXP mask confirmation sheet
(2) ROM data EPROM 3 sets

Mask Options

The following mask options are available, specifiable at the time of initial ordering.

(1) Specify whether a resistor or ceramic resonator is to be used for the clock generator circuit.
(2) Specify whether edge sensing or level sensing is to be provided for port S.

INSTRUCTION CODE TABLE

D3~D0	D8~D4 Hexadecimal number	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	RB 0	LY 0	BL	LA 0	A 0	TAM	IAF	BM	BM	BM	BM	B	B	B	B
0001	1	—	RB 1	LY 1	BL	LA 1	A 1	—	OFA								
0010	2	SZC	RB 2	LY 2	BL	LA 2	A 2	—	—								
0011	3	SNZS	RB 3	LY 3	BL	LA 3	A 3	—	—								
0100	4	SNZT	SB 0	LY 4	BL	LA 4	A 4	XAMI	INY								
0101	5	RSTM	SB 1	LY 5	BL	LA 5	A 5	—	TBA								
0110	6	RC	SB 2	LY 6	BL	LA 6	A 6	—	TYA								
0111	7	SC	SB 3	LY 7	*BL	LA 7	A 7	—	RAR								
1000	8	LX 0	SZB 0	LY 8	BML	LA 8	A 8	—	CMA								
1001	9	LX 1	SZB 1	LY 9	BML	LA 9	A 9	—	IAD								
1010	A	LX 2	SZB 2	LY 10	BML	LA 10	A 10	—	TAY								
1011	B	LX 3	SZB 3	LY 11	BML	LA 11	A 11	—	TAB								
1100	C	—	RD	LY 12	BML	LA 12	A 12	XAM	AMC								
1101	D	—	SD	LY 13	BML	LA 13	A 13	—	—								
1110	E	—	—	LY 14	BML	LA 14	A 14	—	SEA								
1111	F	—	RT	LY 15	*BML	LA 15	A 15	—	SEAM								

Note 1 : An instruction may cosist of one or two word but only the first word is listed.
2 : The BL and BML codes marked with an asterisk are not available with the M50761-XXXP.

MACHINE INSTRUCTIONS

Parameter / Type of instruction	Mnemonic	Instruction code			No. of words	No. of cycles	Functions
		D_7 D_6 D_5 D_4	D_3 D_2 D_1 D_0	16mal notation			
Register-to-register transfers	TAB	0 1 1 1	1 0 1 1	7 B	1	1	(A)←(B)
	TBA	0 1 1 1	0 1 0 1	7 5	1	1	(B)←(A)
	TAY	0 1 1 1	1 0 1 0	7 A	1	1	(A)←(Y)
	TYA	0 1 1 1	0 1 1 0	7 6	1	1	(Y)←(A)
RAM addresses	LY y	0 0 1 0	y y y y	2 y	1	1	(Y)←y, y=0~15
	LX x	0 0 0 0	1 0 x x	0 8+x	1	1	(X)←x, x=0~3
	INY	0 1 1 1	0 1 0 0	7 4	1	1	(Y)←(Y)+1
RAM-accumulator transfers	TAM	0 1 1 0	0 0 0 0	6 0	1	1	(A)←(M(DP))
	XAM	0 1 1 0	1 1 0 0	6 C	1	1	(A)↔(M(DP))
	XAMI	0 1 1 0	0 1 0 0	6 4	1	1	(A)↔(M(DP)), (Y)←(Y)+1
Arithmetic operations	LA n	0 1 0 0	n n n n	4 n	1	1	(A)←n, n=0~15
	A n	0 1 0 1	n n n n	5 n	1	1	(A)←(A)+n, n=0~15
	AMC	0 1 1 1	1 1 0 0	7 C	1	1	(A)←(A)+(M(DP))+(CY) (CY)←carry
	SC	0 0 0 0	0 1 1 1	0 7	1	1	(CY)←1
	RC	0 0 0 0	0 1 1 0	0 6	1	1	(CY)←0
	SZC	0 0 0 0	0 0 1 0	0 2	1	1	(CY)=0 ?
	CMA	0 1 1 1	1 0 0 0	7 8	1	1	(A)←($\bar{A}$)
	RAR	0 1 1 1	0 1 1 1	7 7	1	1	→ $\boxed{CY}$ → $\boxed{A_3\ A_2\ A_1\ A_0}$ →
Bit operations	SB j	0 0 0 1	0 1 j j	1 4+j	1	1	(Mj(DP))←1, j=0~3
	RB j	0 0 0 1	0 0 j j	1 0+j	1	1	(Mj(DP))←0, j=0~3
	SZB j	0 0 0 1	1 0 j j	1 8+j	1	1	(Mj(DP))=0? j=0~3
Comparisons	SEAM	0 1 1 1	1 1 1 1	7 F	1	1	(A)=(M(DP))?
	SEA n	0 1 1 1	1 1 1 0	7 E +	2	2	(A)=n?
		0 1 0 0	n n n n	4 n			
Jumps	B a	1 1 a_5 a_4	a_3 a_2 a_1 a_0	C+a a	1	1	$(PC_L)\leftarrow a_5 \sim a_0$
	BL p,a	0 0 1 1	0 p_2 p_1 p_0	3 0+p	2	2	$(PC_H)\leftarrow p_3 \sim p_0$
		0 p_3 a_5 a_4	a_3 a_2 a_1 a_0	p+a a			$(PC_L)\leftarrow a_5 \sim a_0$
	BM a	1 0 a_5 a_4	a_3 a_2 a_1 a_0	8+a a	1	1	(SK)←PC, $(PC_H)\leftarrow 0$ $(PC_L)\leftarrow a_5 \sim a_0$
	BML p,a	0 0 1 1	1 p_2 p_1 p_0	3 8+p	2	2	(SK)←PC,
		0 p_3 a_5 a_4	a_3 a_2 a_1 a_0	p+a a			$(PC_H)\leftarrow p_3 \sim p_0$ $(PC_L)\leftarrow a_5 \sim a_0$
	RT	0 0 0 1	1 1 1 1	1 F	1	1	(PC)←(SK)
Timer operations	SNZT	0 0 0 0	0 1 0 0	0 4	1	1	(T)=1? After skip (T)←0
	RSTM	0 0 0 0	0 1 0 1	0 5	1	1	Timer reset, (T)←0

Skip conditions	Flag CY	Description of operation
—		Transfers contents of resister B to register A.
—		Transfers contents of register A to register B.
—		Transfers contents of register Y to register A.
—		Transfers contents of register A to resister Y.
Written successively		Loads value of "y" into register Y.
		When LY is written successively the first is executed and successive ones are skipped.
Written successively		Loads value of "x" into register X.
		When LX is written successively the first is executed and successive ones are skipped.
(Y)= 0		Increments contents of register Y by 1. Skips next instruction when new contents of register Y are "0".
—		Transfers the RAM contents addressed by the active DP to register A.
—		Exchanges the contents of the RAM and register A.
(Y)= 0		Exchanges the contents of the RAM and register A.
		The contents of register Y are incremented by 1 and when the result is "0", the next instruction is skipped.
Written successively		Loads the value n in the instruction into register A.
		When LA is written successively, the first is executed and successive ones are skipped.
carry = 0		Adds value of n to register A. The contents of flag CY remain unchanged.
		The next instruction is skipped unless any carry is produced.
—	0/1	Adds the contents of the RAM and flag CY to register A.
		The result is stored in register A and the carry in the active flag CY.
—	1	Sets active flag CY.
—	0	Resets active flag CY.
(CY)= 0		Skips next instruction when the contents of the active flag CY are "0".
—		Stores complement of register A in register A.
—	A_0	Rotates contents of register A and flag CY to right.
—		Sets the jth bit of the RAM addressed by the active DP.
—		Resets the jth bit of the RAM addressed by the active DP.
(Mj(DP))= 0		Skips next instruction when the contents of the jth bit of the RAM addressed by active DP are "0".
(A)=(M(DP))		Skips next instruction when the contents of register A are equal to the RAM contents addressed by the active DP.
(A)= n		Skips next instruction when the contents of register A are equal to the value n in the instruction.
—		Jumps to the address indicated by (a_5—a_0), while PC_H remains unchanged.
—		Jumps to the address indicated by (p_3—p_0, a_5—a_0).
—		Calls subroutine starting from the address indicated by (0,a_5—a_0), replaced by 0 and PC_L has been replaced by (a_5—a_0).
—		Calls subroutine starting from the address indicated by (p_3—p_0, a_5—a_0).
—		Returns from subroutine to main routine.
(T)= 1		Skips the next instruction if timer flag (T) is "1".
		Timer flag is reset when the instruction is skipped.
—		Resets timer and timer flag (T)

MACHINE INSTRUCTIONS

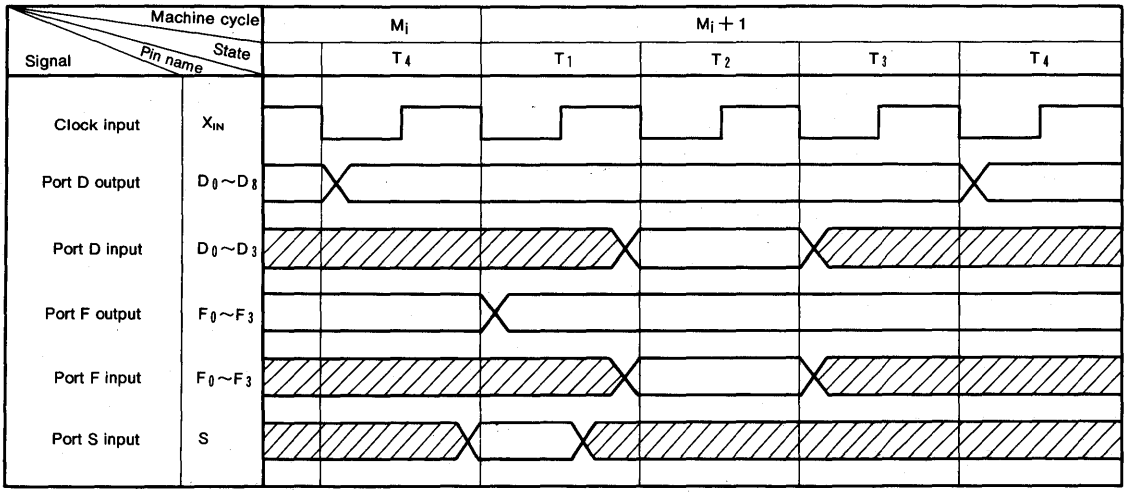
Parameter Type of instruction	Mnemonic	Instruction code									No. of words	No. of cycles	Functions
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	16mal notation			
Input/output operations	SD	0	0	0	1	1	1	0	1	1 D	1	1	(D(Y))←1
	RD	0	0	0	1	1	1	0	0	1 C	1	1	(D(Y))←0
	IAF	0	1	1	1	0	0	0	0	7 0	1	1	(A)←(F)
	OFA	0	1	1	1	0	0	0	1	7 1	1	1	(F)←(A)
	SNZS	0	0	0	0	0	0	1	1	0 3	1	1	(S)=1 ? After skipping (S)←0
	IAD	0	1	1	1	1	0	0	1	7 9	1	1	(A)←(D)
Misc.	NOP	0	0	0	0	0	0	0	0	0 0	1	1	(PC)←(PC)+1

Symbol	Contents	Symbol	Contents
A	4-bit register (accumulator)	D	9-bit port
B	4-bit register	F	4-bit port
X	2-bit register	←	Shows direction of data flow
Y	4-bit register	( )	Indicates contents of register, memory, etc.
DP	6-bit data pointer, combination of register XY	xx	2-bit binary variable
PC_H	The high-order 4 bits of the program counter	yyyy	4-bit binary variable
PC_L	The low-order 6 bits of the program counter	nnnn	4-bit binary constant
PC	10-bit program counter, combination of PC_H, PC_L	jj	2-bit binary constant
SK	10-bit stack register	—	
CY	1-bit carry flag	aaaaaa	Label used to indicate address
T	1-bit timer overflow flag	pppp	Label used to indicate address
S	1-bit port		

Note : When a skip has occurred, the next instruction only is ignored and the program counter is not incremented by 2. Therefore, the number of cycles does not change in accordance with the existence or non-existence of skip.

Skip conditions	Flag CY	Description of operation
—		Sets the bit of port D that is designated by register Y.
—		Resets the bit of port D that is designated by register Y.
—		Transfers port F input to register A.
—		Outputs contents of register A to port F.
(S) = 1		The next instruction is skipped when port S flag (S) is "1". Flag (S) is reset when the instruction is skipped.
—		Transfers port $D_0 \sim D_3$ input to register A.
—		No operation.

BASIC TIMING DIAGRAM



Note 1 ; The crosshatch area indicates invalid input.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{DD}	Supply voltage		−0.3~7	V
V_I	Input voltage, X_{IN}, $\overline{RESET}$		−0.3~ V_{DD} + 0.3	V
V_I	Input voltage, ports F, D_0~D_3, S		−0.3~11	V
V_O	Output voltage, X_{OUT}		−0.3~ V_{DD} + 0.3	V
V_O	Output voltage, ports F, D	Output transistors cut-off	−0.3~11	V
Pd	Power dissipation	T_a = 25°C	400	mW
Topr	Operating temperature		−10~70	℃
Tstg	Storage temperature		−40~125	℃

RECOMMENDED OPERATING CONDITIONS (T_a = −10~70℃, unless otherwise noted)

Symbol	Parameter	Conditions	Limits			Unit
			Min	Nom	Max	
V_{DD}	Supply voltage		4		6	V
V_{SS}	Supply voltage			0		V
V_{IH}	High-level input voltage, port F, D_0~D_3, S		0.7XV_{DD}		10	V
V_{IH}	High-level input voltage, $\overline{RESET}$, X_{IN}		0.7XV_{DD}		V_{DD}	V
V_{IL}	Low-level input voltage		0		0.3XV_{DD}	V
$I_{OL(peak)}$	Low-level peak output current, port D_4~D_8				24	mA
$I_{OL(peak)}$	Low-level peak output current, ports F, D_0~$_3$				10	mA
$I_{OL(avg)}$	Low-level average output current, ports D_4~D_8	(Note 1)			12	mA
$I_{OL(avg)}$	Low-level average output current, ports F, D_0~D_3	(Note 1)			5	mA
$f_{(\phi)}$	Internal clock oscillation frequency		200		400	kHz

Note 1 : The low-level average output currents $I_{OL\ (avg)}$are average values in 100ms period.

ELECTRICAL CHARACTERISTICS (T_a = −10~70℃, V_{DD} = 5 V±10%, f = 200~400kHz)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{OL}	Low-level output voltage, port F, D_0~D_3	I_{OL} = 1.6mA			0.4	V
V_{OL}	Low-level output voltage, ports D_4~D_8	I_{OL} = 12mA			2	V
I_{IH}	High-level input current, ports F, D_0~D_8, S	V_I = 10V			10	μA
I_{IH}	High-level input current, X_{OUT}, $\overline{RESET}$	V_I = V_{DD}			10	μA
$I_{IH}(\phi)$	High-level input current, X_{IN}	V_I = V_{DD}			10	μA
I_{IL}	Low-level input current, F, D_0~D_3, S, X_{OUT}, $\overline{RESET}$	V_I = 0 V			−10	μA
$I_{IL}(\phi)$	Low-level input current, X_{IN}	V_I = 0 V			−10	μA
C_i	Clock input capacitance	f = 1 MHz		7	10	pF
I_{DD}	Supply current	f = 400kHz		400	900	μA

MELPS 740 MICROCOMPUTERS

3

MITSUBISHI MICROCOMPUTERS

M50740-XXXSP

SINGLE-CHIP 8-BIT CMOS MICROCOMPUTER

DESCRIPTION

The M50740-XXXSP is a single-chip 8-bit microcomputer fabricated using CMOS technology and housed in a 52-pin shrink plastic molded DIL package. It is designed to suit for controlling home electrical appliances and consumer equipment with a simple instruction where the ROM and RAM use the same memory area.

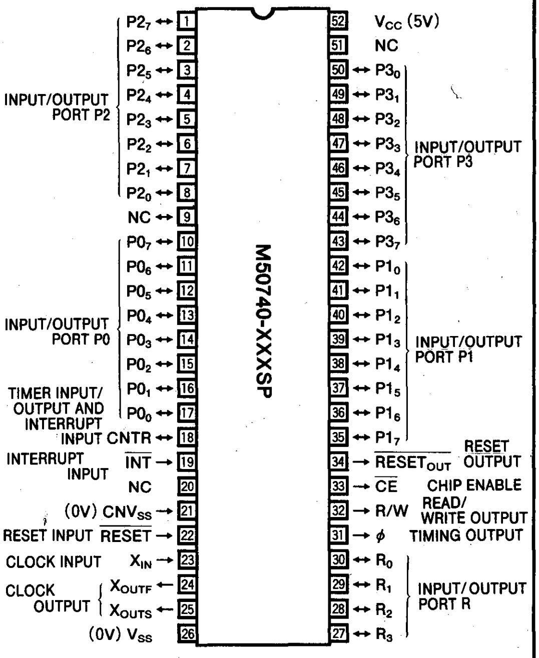
FEATURES

● Basic machine instructions	70
● Memory capacity ROM	3072 bytes
RAM	96 bytes
● Instruction execution time (with shortest instruction, at 4MHz)	2μs (min.)
● Single power supply	5V ±10%
● Low power dissipation	15mW during operation (4MHz)
● Subroutine nesting	48 levels (max.)
● Interrupts	5 sources, 32 levels (max.)
● 8-bit timers	3
● Programmable input/output (ports P0, P1, P2, P3)	32

APPLICATION

VTRs, tuners and audio equipment

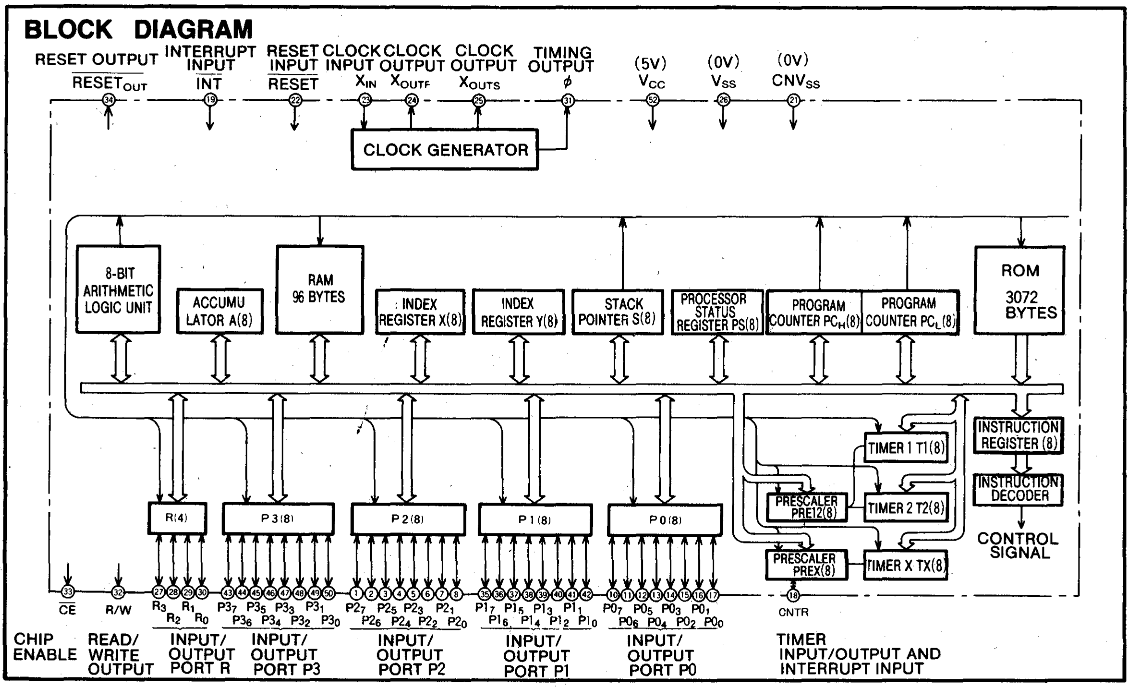
PIN CONFIGURATION (TOP VIEW)



Outline 52P4B

NC : NO CONNECTION

BLOCK DIAGRAM



PERFORMANCE SPECIFICATIONS

Parameter			Performance
Number of basic instructions			70
Execution time of basic instruction			2μs (with shortest instructions, 4MHz clock frequency)
Clock frequency			4MHz
Memory capacity	ROM		3072 bytes
	RAM		96 bytes
Input/output ports	$\overline{\mathrm{INT}}$	Input	1 bit × 1
	P0, P1, P2, P3	Input/output	8 bits × 4
	R	Input/output	4 bits × 1
	CNTR	Input/output	1 bit × 1
Timers			8-bit prescalers x 2 + 8-bit timers x 3
Subroutine nesting			Max. 48 level
Interrupts			External interrupts (2), internal timer interrupts (3)
Clock generator			Built-in (externally connected RC circuit, ceramic or quartz resonator)
Supply voltage	During operation		5V ± 10%
Power dissipation	High-speed operation		15 mW (at 4MHz clock frequency)
	Low-speed operation		100 μW (at 20kHz clock frequency)
Input/output characteristics	Input/output withstanding voltage		12V (Ports P0, P1, P2, $\overline{\mathrm{INT}}$, CNTR)
	Output current		10mA (Ports P0, P1, P2, P3)
Memory expansion			Possible
Ambient operating temperature			−10～70℃
Device structure			CMOS silicon gate process
Package			52-pin shrink plastic molded DIL

PIN DESCRIPTION

Pin	Name	Input or output	Function
V_{CC} V_{SS}	Supply voltage	In	5V ±10% supplied to V_{CC}, 0V supplied to V_{SS}
CNV$_{SS}$	CNV$_{SS}$ input	In	To be connected to V_{SS}.
$\overline{\mathrm{RESET}}$	Reset input	In	When this input is kept low for at least 2μs, the reset state is enabled.
X_{IN}	Clock input	In	The clock generator circuit is built-in. For setting the oscillation frequency, either connect the external RC circuit to X_{IN} and X_{OUTS} or X_{OUTF} or connect a ceramic or quartz resonator across X_{IN} and X_{OUTS}. When using an external clock source connect the clock generator source to X_{IN}, leaving X_{OUTF} and X_{OUTS} open. For details, refer to the section on the clock generator circuit.
X_{OUTS}	Clock output	Out	Internal clock generator output. An RC circuit or ceramic or quartz resonator is connected between this output and X_{IN} to control the oscillation frequency. For details, refer to the section on the clock generator circuit.
X_{OUTF}	Clock output	Out	Internal clock generator output. An RC circuit is connected between this output and X_{IN} to control the oscillation frequency. For details, refer to the section on the clock generator circuit.
ϕ	Timing output	Out	Timing output
CNTR	Timer input/output and interrupt input	In/out	Timer X input/output pin and interrupt input pin.
$\overline{\mathrm{INT}}$	Interrupt input	In	Interrupt input pin.
$P0_0$～$P0_7$	Input/output port P0	In/out	This 8-bit input/output port has a direction register and for each bit the port is programmed to serve for input or output. The input mode is established during resetting. N-channel open-drain circuits are used for the outputs. For details, refer to the section on the input/output pins.
$P1_0$～$P1_7$	Input/output port P1	In/out	This is an 8-bit input/output port with virtually the same functions as those of port P0.
$P2_0$～$P2_7$	Input/output port P2	In/out	This is an 8-bit input/output port with virtually the same functions as those of port P0.
$P3_0$～$P3_7$	Input/output port P3	In/out	This is an 8-bit input/output port with virtually the same functions as those of port P0, but p-channel open-drain circuits are used for the outputs.
R_0～R_3	Input/output port R	In/out	This 4-bit input/output port is used for connection with the I/O expander.
R/W	Read/write output	Out	Read/write signal output for I/O expander.
$\overline{\mathrm{CE}}$	Chip enable output	Out	Chip enable signal output for I/O expander.
$\overline{\mathrm{RESET}}_{OUT}$	Reset output	Out	Reset signal output for I/O expander.

BASIC FUNCTION BLOCKS

Memory

Fig. 1 shows the memory map. The 3072-byte ROM extends from 1400_{16} to $1FFF_{16}$. The area from $1F00_{16}$ to $1FFF_{16}$ includes special addresses, and when the special page addressing mode is used with the JSR instruction, subroutines on these pages can be called with two bytes. The area from $1FF4_{16}$ to $1FFF_{16}$ includes the reset and interrupt vector addresses. For details, refer to the section on interrupts.

The addresses from 0000_{16} to $00FF_{16}$ are own as the zero page and access to this page can be achieved with two bytes by using the zero page addressing mode, which reduces the number of programming steps. The memories used frequently, such as the RAM, input/output ports and timers, are allocated to the zero page.

From 0000_{16} to $005F_{16}$ is the RAM space and the size is 96 bytes. Apart from storing data, the RAM is also used as a stack for subroutine calls or interrupts.

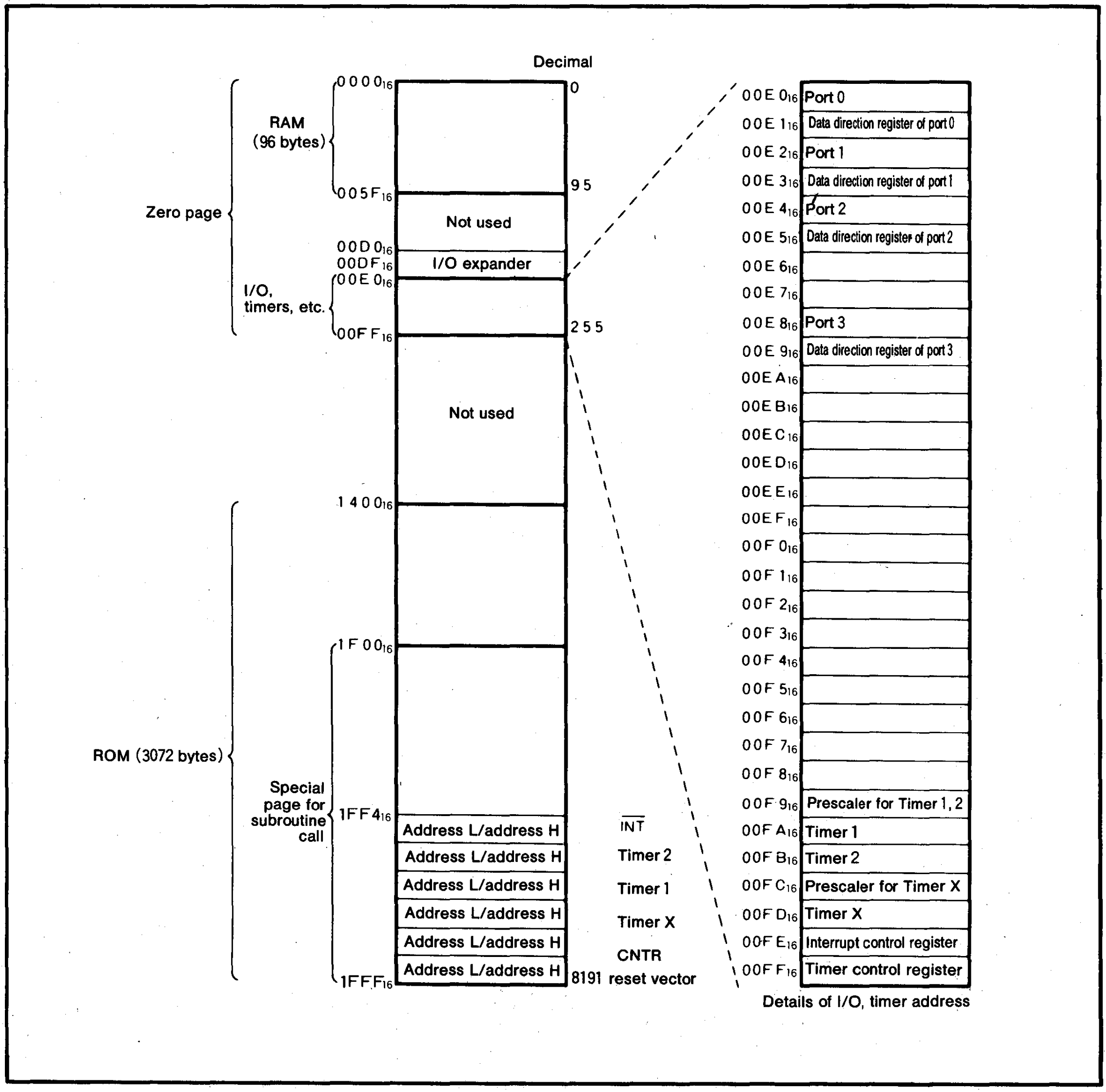


Fig.1 Memory layout

CPU

Six registers, as shown in Fig. 2, are contained inside the CPU. Each of these register is now described in turn.

Accumulator A

The accumulator is the 8-bit register and is heart of the microcomputer. Arithmetic and logic operations, transfers and processing of input/output and other data are performed centering on this register.

Index Register X

This is an 8-bit register. In the index addressing mode where this register serves as the index register, the contents of this register and the contents of the program counter is added and the result is actual address. When flag T in the program status register is "1," the contents of index register X become the other operand address.

Index Register Y

This is also an 8-bit register. In the index addressing mode where this register serves as the index register, the contents of this register and the contents of the program counter is added and the result is actual address.

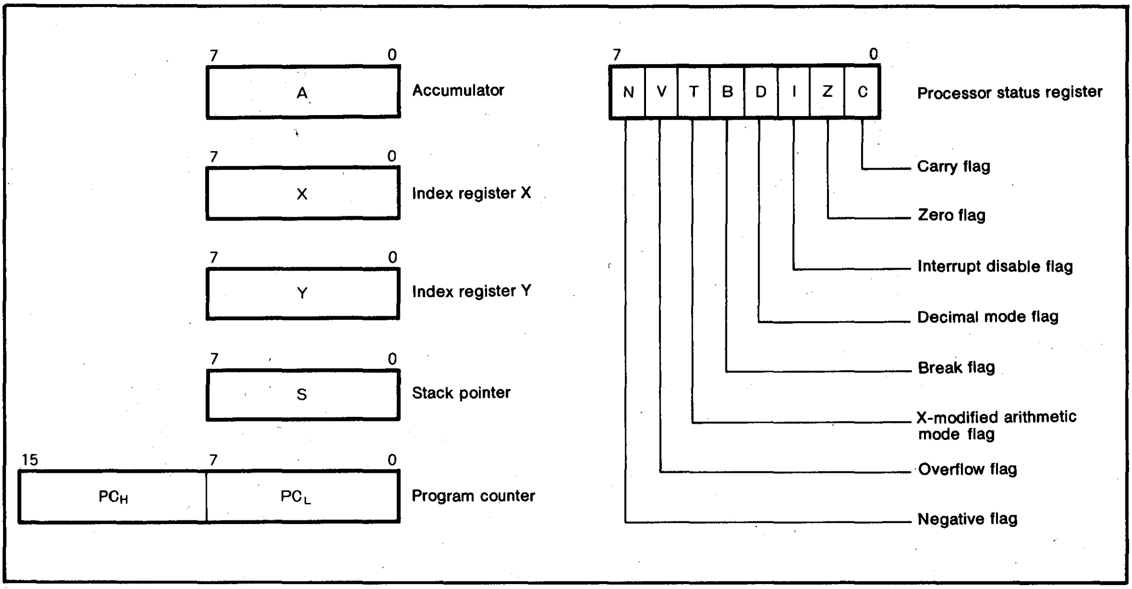


Fig.2 Register configuration

Stack Pointer S

The stack pointer is an 8-bit register used for calling subroutines and for interrupts. When an interrupt is acknowledged, the high-order contents of the program counter are saved in the address where the low-order address are the contents of the stack pointer and the high-order address are "0," the contents of the stack pointer are then decremented by 1, the low-order contents of the program counter are saved in the address where the low-order address are the contents of the stack pointer and the high-order address are "0," the contents of the stack pointer are then further decremented by 1, and the contents of the program status register are saved in the address where the low-order address are the contents of the stack pointer and the high-order address are "0."

This operation is performed automatically when an interrupt is acknowledged. The RTI instruction is used to return from the interrupt routine, and when it is executed, the stack pointer is incremented by 1 and returned in the reverse sequence to that described above. Since the contents of accumulator are not saved automatically, the PHA instruction must be used for this purpose. When the PHA instruction is executed, the contents of the accumulator are saved in the address where the low-order address are the contents of the stack pointer and the high-order address are "0," and the contents of the stack pointer are decremented by 1. The accumulator is returned by the PLA instruction. When this instruction is executed, the contents of the stack pointer are incremented by 1 and the contents of the address where the low-order address are the stack pointer contents and the high-order address are "0" enter the accumulator.

Similarly, the contents of the program status register are saved and returned by the PHP and PLP instructions respectively. With a subroutine call, program counter saving only is performed and this necessitates saving on the program for registers which must not be destroyed. The RTS instruction is employed to return from the subroutine.

Program Counter PC

This is a 16-bit counter consisting of PC_H and PC_L, both is 8 bit register. PC_H is 8 bit register, but only 5 bits are actually used. The program counter specifies the address of the program memory which is to be executed next.

Processor Status Register PS

This 8-bit register consists of the flags that hold the status immediately after arithmetic and logic operations. The C, Z, V and N flags can be tested and branched using the branch instructions. Each bit of the register is described in detail below.

1. Carry Flag C

The carry flag C is used to store carry or overflow after execution of arithmetic and logic operations by the arithmetic logic unit. It also undergoes change with the shift and rotate instructions. It can be set or reset directly using the SEC and CLC instructions.

2. Zero Flag Z

This flag is set when the results of data transfer or arithmetic and logic operations are "0" and reset when they are not "0."

3. Interrupt Disable Flag I

This flag disables all interrupts when its contents are "1." When an interrupt is acknowledged, its contents are automatically made "1." The flag can be set or reset by the program using the SEI and CLI instructions.

4. Decimal Mode Flag D

This flag determines whether additions and subtractions are to be undertaken by the binary or decimal mode. The ordinary binary mode is used when its contents are "0", while 1 word is processed as a 2-digit decimal number when its contents are "1." Decimal corrections are performed automatically. The SED and CLD instructions are used for setting and resetting.

5. Break Flag B

Operation is the same for interrupts when the BRK instruction is executed. This instruction is used for debugging programs. The BRK instruction interrupt vector and the interrupt vector of the lowest order of priority are located in the same address. In order to discriminate whether or not an interrupt has occurred with the BRK instruction, the contents of flag B are set to "1" when interrupted by the BRK instruction; at all other times, the contents are set to "0" and saved. It is possible to ascertain whether an interrupt has occurred with BRK by investigating the bit saved in the interrupt routine.

6. X-modified Arithmetic Mode Flag

Arithmetic and logic operations are performed between the accumulator and memory when the flag T contents is "0." When this bit is "1," the accumulator is bypassed and operations are performed directly between the memories. The results of such operations between memory 1 and memory 2 enter memory 1. The memory 1 address is specified by the contents of index register X; the memory 2 address is specified by the ordinary addressing mode. The SET and CLT instructions are used for setting and resetting flag T.

7. Overflow Flag V

This flag is significant when in the addition and subtruction a single word is treated as a signed binary number. It is set when the results of an addition or subtraction exceed +127 or −128. Apart from this, the 6th bit of the memory subject to the execution of the BIT instruction enters the overflow flag when this instruction is executed. The CLV instruction is used to clear the overflow flag. A setting instruction is not provided.

8. Negative Flag N

This flag is set when the results of an arithmetic or logic operation or of data transfer are negative (7th bit is "1"). In addition, the 7th bit of the memory subject to the BIT instruction enters the negative flag when this instruction is executed. Instructions to set and reset this flag are not provided.

Table 1 Interrupt vector addresses and priority

Interrupt source	Priority	Vector address
$\overline{\text{RESET}}$	1	1FFF, 1FFE
CNTR	2	1FFD, 1FFC
Timer X	3	1FFB, 1FFA
Timer 1	4	1FF9, 1FF8
Timer 2	5	1FF7, 1FF6
$\overline{\text{INT}}$ (BRK)	6	1FF5, 1FF4

INTERRUPTS

Interrupts include the interrupt from pin CNTR, the timer X interrupts, timer 1 interrupt, timer 2 interrupt, the interrupt from pin $\overline{\text{INT}}$ and the interrupt based on the BRK instruction. The interrupts are vector interrupts and Table 1 shows the vector table and priority. Resetting take the same action as interrupt and so it is descried here.

When an interrupt is acknowledged, the registers are saved, as described in the above section on stack pointer S, the interrupt disable flag I is set and a jump is made to the address indicated by the contents of the vector table. The interrupt request bit is automatically cleared. Resetting is not disabled by any condition. Interrupts (exclusive of resetting) are not acknowledged when the interrupt disable flag has been set. The interrupts from pin CNTR, timer X, timer 1, timer 2 and $\overline{\text{INT}}$ can be controlled individually by the interrupt control and timer control registers. This is shown in Fig. 3. When the interrupt enable bit is "1," when the interrupt request bit is "1" and when the interrupt disable flag I is "0," the interrupt is acknowledged. When the level of pins CNTR and $\overline{\text{INT}}$ change from high to low or when the contents of timer X, timer 1 or timer 2 reach to "0," the corresponding interrupt request bits are set.

These bits can be reset by programming but cannot be set. The interrupt enable bit can be set and reset by programming. Whether interrupt is caused by the BRK instruction, can be verified by checking break flag B which has been saved, as mentioned in the section on the break flag B.

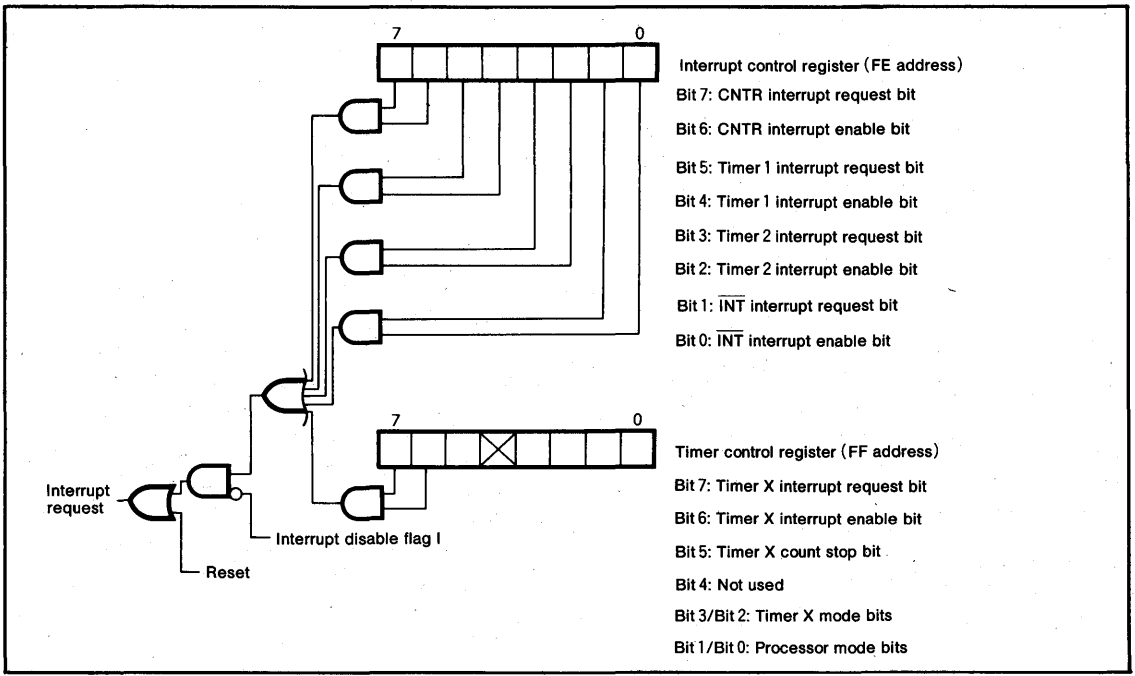

7
0
Interrupt control register (FE address)
Bit 7: CNTR interrupt request bit
Bit 6: CNTR interrupt enable bit
Bit 5: Timer 1 interrupt request bit
Bit 4: Timer 1 interrupt enable bit
Bit 3: Timer 2 interrupt request bit
Bit 2: Timer 2 interrupt enable bit
Bit 1: $\overline{\text{INT}}$ interrupt request bit
Bit 0: $\overline{\text{INT}}$ interrupt enable bit
7
0
Timer control register (FF address)
Bit 7: Timer X interrupt request bit
Bit 6: Timer X interrupt enable bit
Bit 5: Timer X count stop bit
Bit 4: Not used
Bit 3/Bit 2: Timer X mode bits
Bit 1/Bit 0: Processor mode bits
Interrupt request
Interrupt disable flag I
Reset

Fig.3 Interrupt control

Timers

There are 3 timers: timer X, timer 1 and timer 2. Timer X has four modes which are selected by the value of the timer X mode bits (bit 2 and bit 3) in the timer control register. When the timer count stop bit (bit 5) is set to "1," all four timer X modes stop. Fig. 4 is a block diagram of timers X, 1 and 2. Timer 1 and timer 2 have a common prescaler composed of 8 bits. The frequency division ratio is determined by the prescaler contents. This ratio is 1/(n+2) when the prescaler latch contents are made n decimally. All the timers have 8-bit timer latches. The countdown system is featured for the timers, and the timer latch contents are reloaded into the timer at the following cycle when the counter contents reach to "0."

When the timer contents reach to "0," the interrupt request bit (on the interrupt control register or on the timer control register located in the FE_{16} or FF_{16} address respectively) corresponding to the timer is set to "1." Any number except "0" should be entered in the prescaler latch and timer latch.

Refer to the section on interrupts for details. The four modes of timer X are now described.

(1) Timer mode (00)
In this mode the frequency produced by dividing the oscillation frequency by 16, is counted. When the timer contents reach to "0," the interrupt request bit is set to "1," the timer latch contents are re-loaded and the count is continued.

(2) Pulse output mode (01)
Every time the timer contents reach to "0," the signal on the pin CNTR changes the polarity.

(3) Event counter mode (10)
Operation is the same as in the timer mode except for counting the signal from pin CNTR.

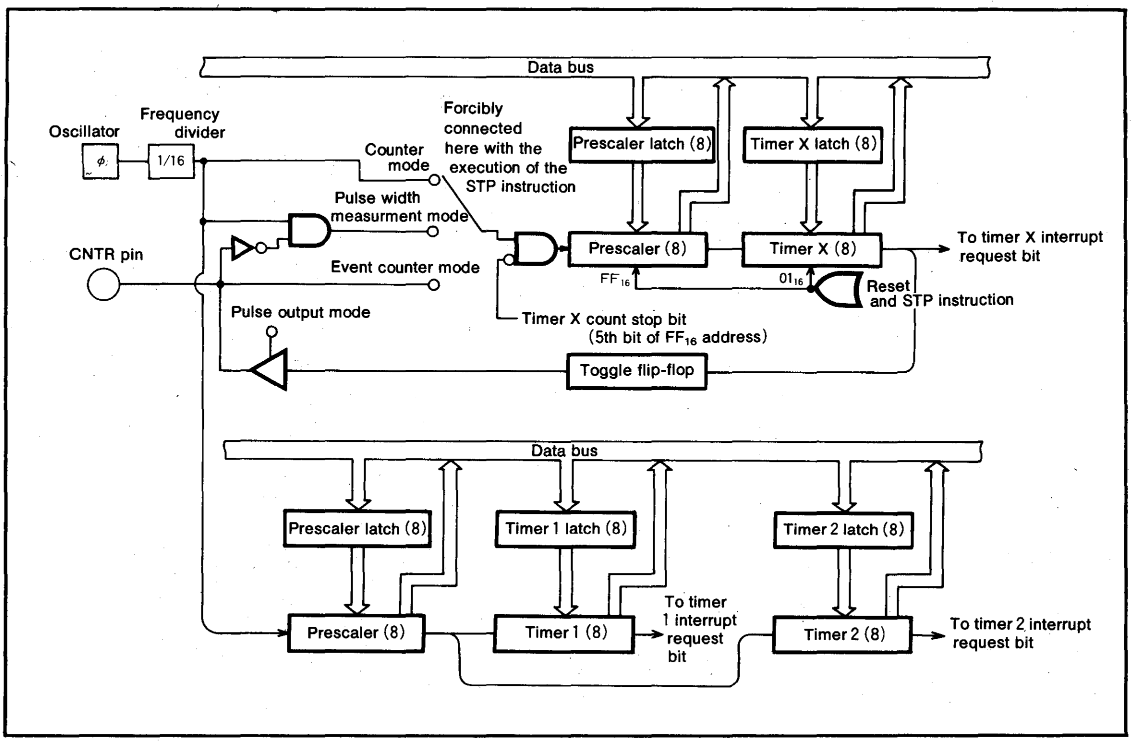


Fig.4 Block diagram of timer X, timer 1, timer 2

(4) Pulse width measurment mode (11)

The frequency, produced by dividing the oscillation frequency by 16, is counted only while the pin CNTR level is low. When the counter contents reach to "0," the interrupt request bit is set to "1," the timer latch contents are re-loaded and the count is continued.

Fig. 5 shows the relationship between the timer control register contents and the timer modes.

Also shown are the processor mode and other bits.

When reset or the STP instruction is executed, the timer X prescaler is set in FF_{16} and the timer X latch is set in 01_{16}. When the STP instruction is executed, the frequency produced by dividing the oscillation frequency by 16 serves as the timer X prescaler input, regardless of the timer X mode bit. This mode is released either when the timer X interrupt request bit is set to "1" or when resetting is accomplished and resume the mode determined by the timer X mode bit. For details on the operation of the STP instruction, reference should be made to the section on the oscillator circuit.

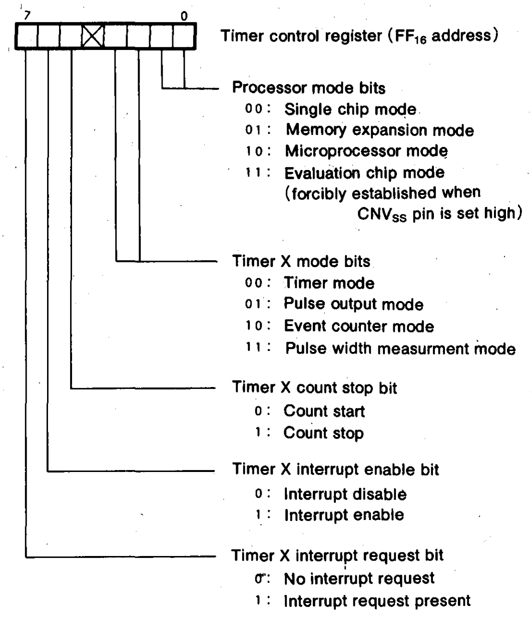
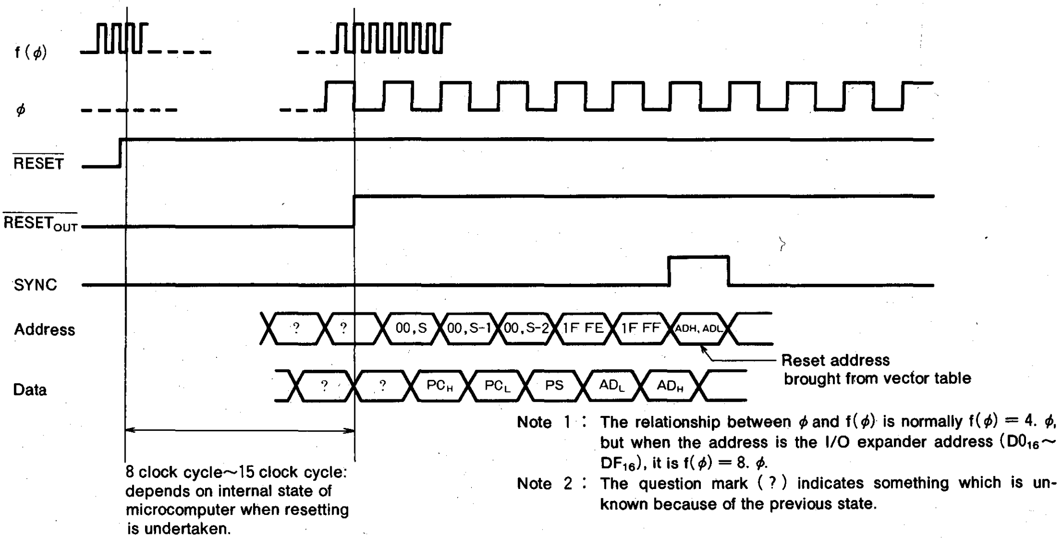


Fig.5 Configuration of timer control register


f (ϕ)
ϕ
RESET
RESETOUT
SYNC
Address
?
?
00,S
00,S-1
00,S-2
1F FE
1F FF
ADH, ADL
Reset address brought from vector table
Data
?
?
PCH
PCL
PS
ADL
ADH
8 clock cycle~15 clock cycle: depends on internal state of microcomputer when resetting is undertaken.

Note 1 : The relationship between ϕ and $f(\phi)$ is normally $f(\phi) = 4. \phi$, but when the address is the I/O expander address ($D0_{16}$~DF_{16}), it is $f(\phi) = 8. \phi$.

Note 2 : The question mark (?) indicates something which is unknown because of the previous state.

Fig.6 Timing diagram during resetting

Reset Circuit

When a supply voltage of 5V ±10% is being supplied to the. M50740-XXXSP and the $\overline{\text{RESET}}$ pin is returned to the high level after being kept at the low level for 2μs or more, the reset is released in accordance with the sequence shown in Fig. 6, and the program starts from the address which is derived from the contents of the address $1FFF_{16}$ and $1FFE_{16}$,high-order address is the contents of address $1FFF_{16}$ and low-order address is the contents of address $1FFE_{16}$. When resetting is accomplished, the internal state of the microcomputer is as shown in Fig. 7.

Fig. 8 shows an example of the reset circuit.

The reset input voltage should be set to less than 0.6V at that point when the supply voltage is passing through 4.5V.

	Address	
(1) Data direction register of port 0	($E1_{16}$) …	00_{16}
(2) Data direction register of port 1	($E3_{16}$) …	00_{16}
(3) Data direction register of port 2	($E5_{16}$) …	00_{16}
(4) Data direction register of port 3	($E9_{16}$) …	00_{16}
(5) Prescaler X	(FC_{16}) …	FF_{16}
(6) Timer X	(FD_{16}) …	01_{16}
(7) Interrupt control register	(FE_{16}) …	00_{16}
(8) Timer control register	(FF_{16}) …	00_{16}
(9) Interrupt disable flag on the processor status register	(PS) …	| | | | | |1| | |
(10) Program counter	(PC_H) …	contents of the address $1FFF_{16}$
	(PC_L) …	contents of the address $1FFE_{16}$
(11) The oscillator output is connected to pin X_{OUTF} as with the state established after the FST instruction has been executed.		

Fig.7 Internal state of microcomputer after resetting

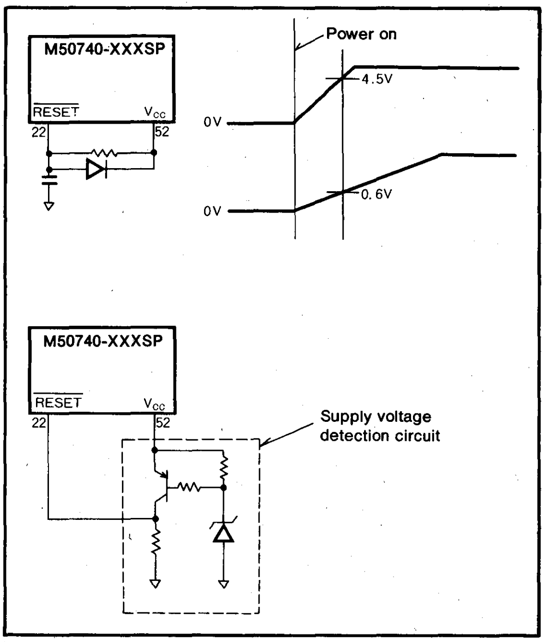


Fig.8 Example of reset circuit

Input/Output Pins

(1) Port P0

This port is an 8-bit input/output port with n-channel open-drain outputs. As shown in the memory map of Fig. 1, port P0 is treated as the memory of address $E0_{16}$ on the zero page. Port P0 has a data direction register (address $E1_{16}$ on zero page) and programming can be undertaken for individual bit to use the port for input or output. The pins where the data direction register is programmed to "1" are for output and those where the register is programmed to "0" are for input. The data written into the pin programmed as an output pin are written into the port latch and supplied direct to the output pin. When reading the data from a pin programmed as an output pin, it is not the output pin contents which are read but the port latch contents. Consequently, since the LED or other similar part is driven directly, the value output previously can be read correctly even if the low-level output voltage rises. The pin programmed as an input pin remains floating, so external signal can be read. When data are written, they are written into the port latch only and the pin remains floating.

(2) Port P1
This has the same functions as port P0.

(3) Port P2
This has the same functions as port P0.

(4) Port P3
Apart from the fact that this port has p-channel open-drain outputs, its functions are the same as those for port P0. Fig. 9 is a block diagram of port P0~P3.
Also indicated are the output structure of port R, CNTR, ϕ, R/W, $\overline{\text{CE}}$ and $\overline{\text{RESET}_{\text{OUT}}}$.

(5) Port R
This port is for exchanging data with the I/O expander. When ϕ is high, the port address of the I/O expander is sent; when it is low, data are sent to or received from the expander. The above data and addresses are effective only when pin $\overline{\text{CE}}$ is low. Fig. 10 is a timing diagram.

(6) $\overline{\text{CE}}$
This pin is set low when the address becomes the I/O expander address ($D0_{16}$~DF_{16}). It is used to inform the I/O expander that the port R address or data is effective.

(7) R/W
This is set low while writing is being executed, and it is used to inform the I/O expander that either writing or reading is being undertaken.

(8) ϕ
Normally output to this pin is a signal with a frequency produced by dividing the clock frequency by 4. However, when pin $\overline{\text{CE}}$ is low, an output with a frequency which is one-eighth of the clock frequency is output. The pin is used to provide synchronization with the I/O expander.

(9) $\overline{\text{RESET}_{\text{OUT}}}$
When the $\overline{\text{RESET}}$ pin is set low, this pin also goes low. When the $\overline{\text{RESET}}$ pin is set high, the pin also goes high after between 8 and 15 clock cycles (this depends on the internal state of the microcomputer). The $\overline{\text{RESET}_{\text{OUT}}}$ pin itself is used to reset the I/O expander.

(10) $\overline{\text{INT}}$
When an input which changes the level from high to low is applied to this interrupt input pin, the $\overline{\text{INT}}$ interrupt request bit (bit 1 of address FE_{16}) is set to "1."

(11) CNTR
This pin serves both as the timer X input/output pin and as the interrupt input pin. When an input which changes its level from high to low is applied, the CNTR interrupt request bit (bit 7 of address FE_{16}) is set to "1." The pin serves as the external pulse input pin in the event counter mode. In the pulse output mode a pulse which reverses its polarity is output every time the timer X contents are reach to "0." In the pulse width measument mode, the pulse to be measured is supplied to this pin.

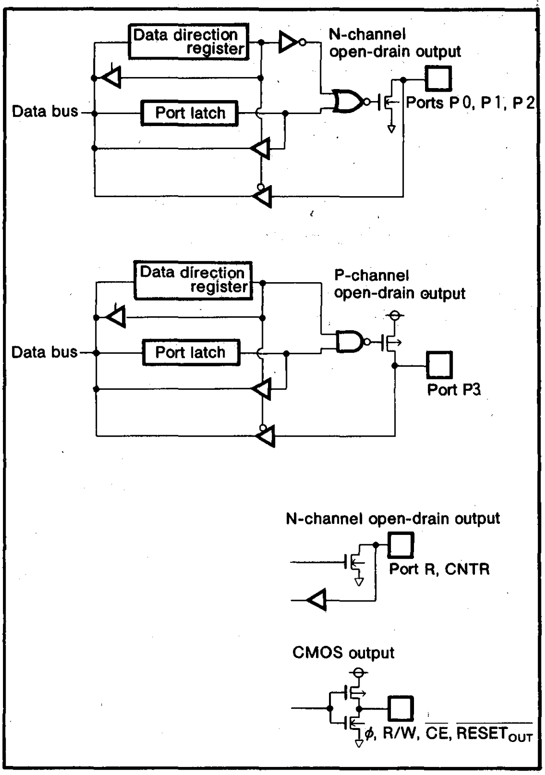

Data direction register
N-channel open-drain output
Data bus
Port latch
Ports P0, P1, P2
Data direction register
P-channel open-drain output
Data bus
Port latch
Port P3
N-channel open-drain output
Port R, CNTR
CMOS output
ϕ, R/W, $\overline{\text{CE}}$, $\overline{\text{RESET}_{\text{OUT}}}$

Fig.9 Block diagram of port P0~P3 (single chip mode) and output formats of port R, CNTR, R/W, $\overline{\text{CE}}$ and $\overline{\text{RESET}_{\text{OUT}}}$

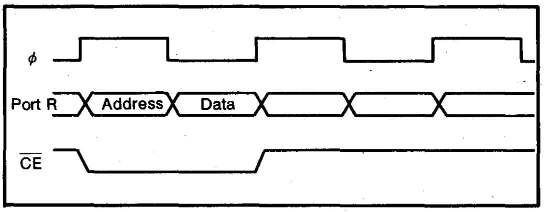


Fig.10 Timing diagram of port R

Clock Generator Circuit

The clock generator circuit is built-in, as shown in Fig. 11. When the STP instruction is executed, oscillation is stopped with the internal clock ϕ in the high-level. Furthermore, FF_{16} is set in prescaler X and 01_{16} in timer X, and the output, one-sixteenth of the oscillator output, is forcibly connected to the prescaler X input. This connection is released when, as mentioned in the timer section, timer X overflows or when resetting is accomplished. Oscillation re-starts when an interrupt is acknowledged but the internal clock ϕ remains high until timer X overflows. Only when timer X overflows is the internal clock ϕ supplied. This is because time is required for the oscillation to rise when a ceramic resonator or similar part is employed.

When the FST instruction is executed, SW_{OSC} closes and when the SLW instruction is executed, it opens. These instructions are used when RC oscillation is emploied and the oscillation frequency is changed. SW_{OSC} is closed during resetting.

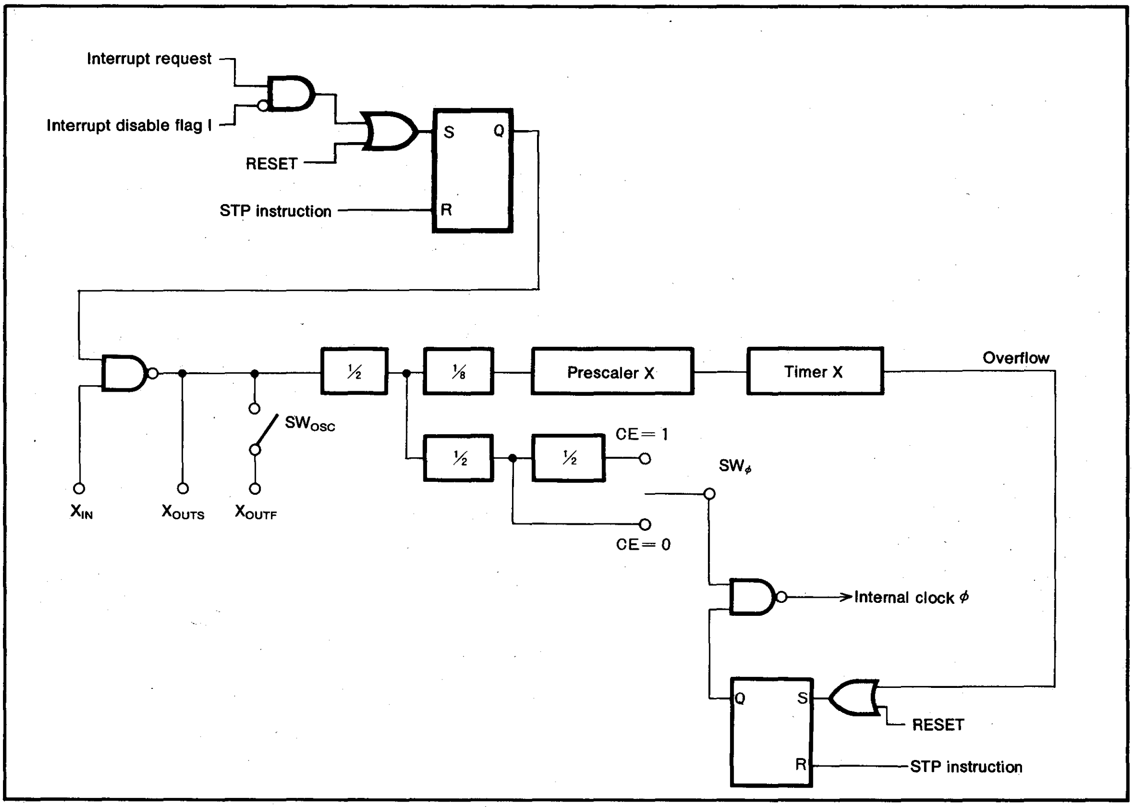


Fig.11 Block diagram of clock generator circuit

When the address becomes the I/O expander address ($D0_{16}$ ~ DF_{16}), SWϕ is connected to the output (CE=1) which is one-eighth of the oscillation frequency and at all other times it is connected to the output (CE=0) which is one-fourth of the same frequency. This is because a margin in terms of time is given to the signal exchange with the I/O expander.

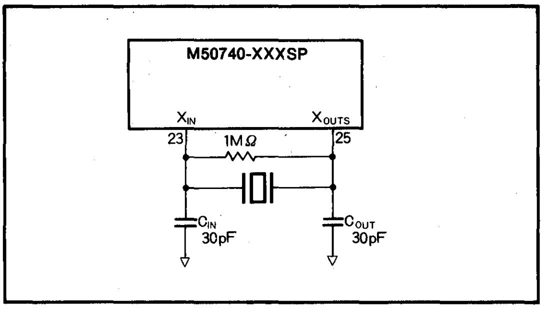


Fig.12 Externally connected ceramic resonator circuit

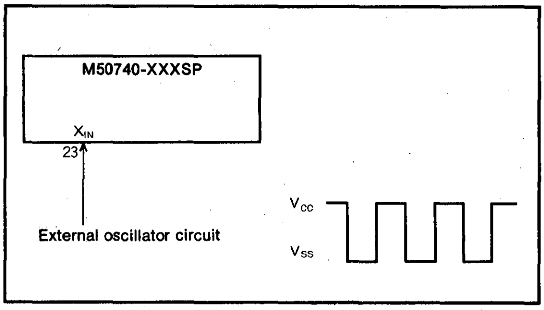


Fig.13 External clock input circuit

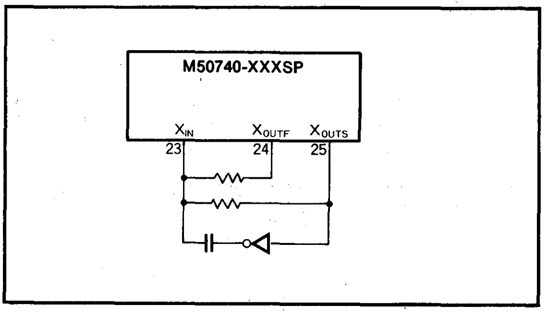


Fig.14 External RC circuit

Figs. 12~14 give examples of clock generator circuits. The clock signal is produced if a ceramic resonator (or quartz crystal) is externally connected. X_{OUTF} is left open. The capacitance and other constants depend on the resonator itself and the values recommended by the manufacturer in question should be used.

When the external clock source is used it should be applied to the X_{IN} pin with pins X_{OUTS} and X_{OUTF} left open.

An inverter is required externally for RC oscillation.

Addressing Modes

The M50740-XXXSP has 17 addressing modes and an extremely powerful memory access capability.

When extracting data required for arithmetic and logic operations from the memory or when storing the results of such operations in a memory using the appropriate instructions for this purpose, the memory address must be specified. Even when jumping to an address during a program, that particular address must be specified. The specification of the memory address is called addressing. The data required for addressing and the registers involved are now described. The M50740-XXXSP's instructions can be classified into three kinds, as shown in Fig. 15, by the byte number in the program memory required for configuring the instruction: 1-byte, 2-byte and 3-byte instructions. In each case, the first byte is known as the "operation code" which forms the basis of the instruction. The second or third byte is called the "operand" which affects the addressing. The contents of index registers X and Y also effect the addressing.

However many the addressing modes, there is no difference in the sense that a particular memory is specified. What differs is whether the operand or the index register contents or a combination of both should be used to specify the memory or jump destination. Based on these 3 methods, the range of variation is increased and the M50740-XXXSP's operation is enhanced by combinations of the bit operation instructions, jump instruction and arithmetic instructions. The accumulator or register is specified with a 1-byte instruction and so there is no operand byte, which is the part specifying the memory.

Actual addressing modes are now described by type.

3

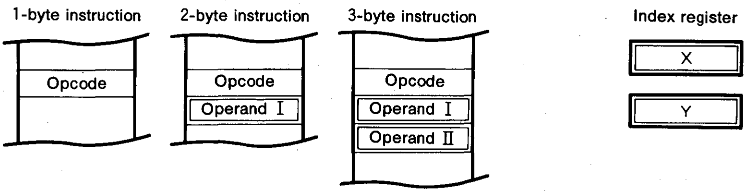
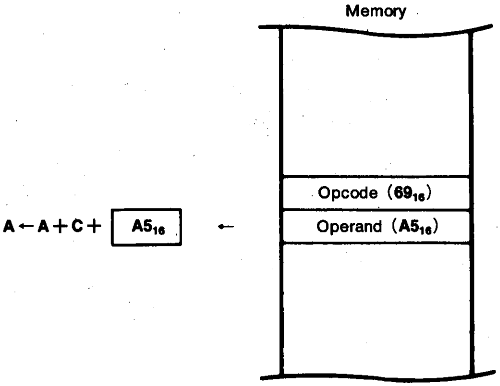


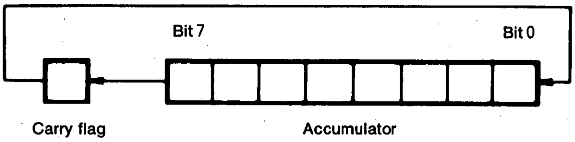
Fig.15 Instruction byte configuration

Name	: **Immediate addressing mode**
Function	: Operand follow immediate after opcode.
Instructions	: **ADC, AND, CMP, CPX, CPY, EOR, LDA, LDX, LDY, ORA, SBC**
Example	: Mnemonic Machine code
	ADC #$A5 **$69_{16}$ $A5_{16}$**

*This symbol designates the immediate addressing mode.



Name	: **Accumulator addressing mode**
Function	: Operation is performed on accumulator.
Instructions	: **ASL, DEC, INC, LSR, ROL, ROR**
Example	: Mnemonic Machine code
	ROL A **$2A_{16}$**

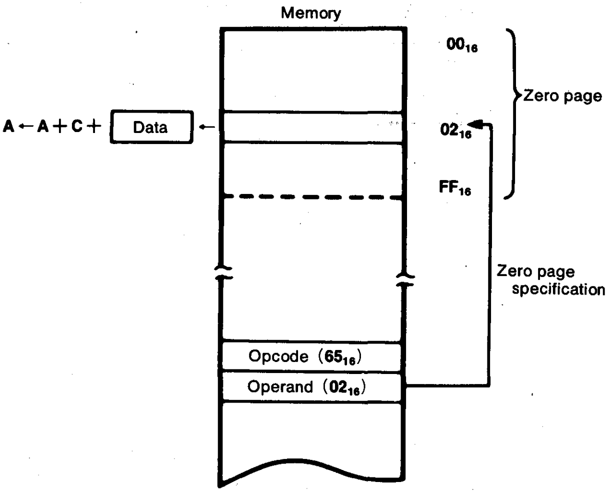


Name : Zero page addressing mode

Function : Operation is performed on the zero page memory (00_{16}~FF_{16})

Instructions : **ADC, AND, ASL, BIT, CMP, COM, CPX, CPY, DEC, EOR, INC, LDA, LDM, LDX, LDY, LSR, ORA, ROL, ROR, RRF, SBC, STA, STX, STY, TST**

Example : Mnemonic	Machine code
ADC \$02	**65_{16} 02_{16}**

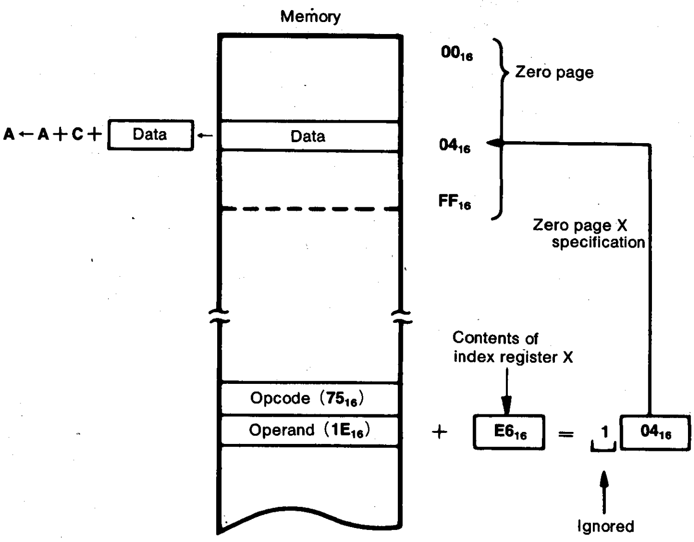


Name : Zero page X addressing mode

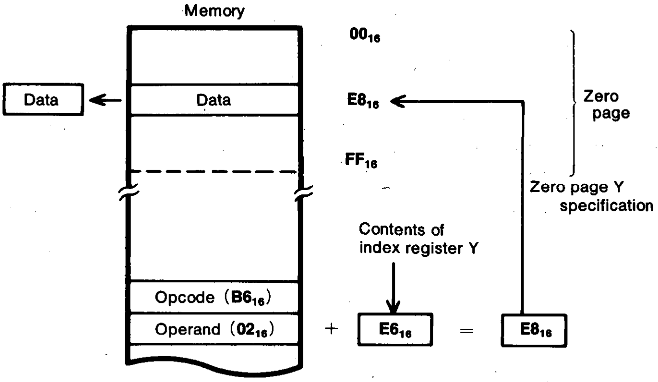
Function : Operation is performed on the memory which address is specified by adding the operand and contents of index register X.

Instructions : **ADD, AND, ASL, CMP, DEC, EOR, INC, LDA, LDY, LSR, ORA, ROL, ROR, SBC, STA, STY**

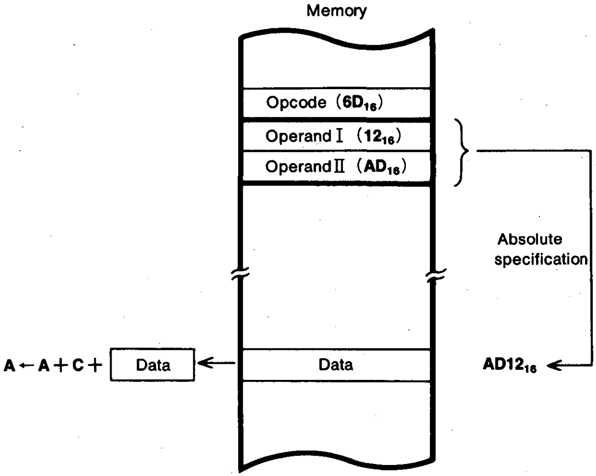
Example : Mnemonic	Machine code
ADC \$1E,X	**75_{16} $1E_{16}$**



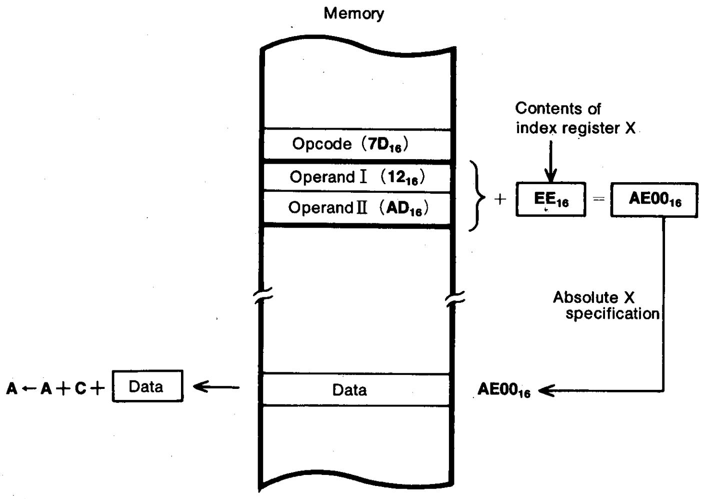
Name : Zero page Y addressing mode
Function : Operation is performed on the memory which address is specified by adding the operand and contents of index register Y.
Instructions : **LDX, STX**
Example : Mnemonic Machine code
LDX $02,Y **$B6_{16}$ 02_{16}**



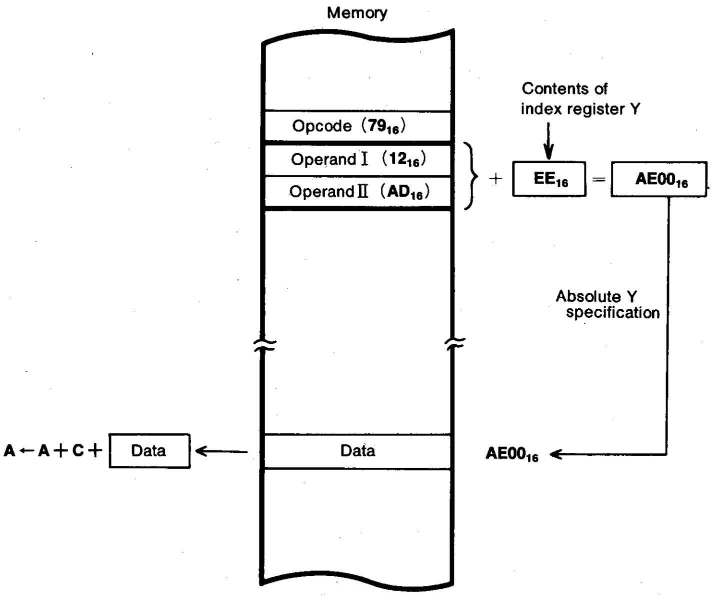
Name : Absolute addressing mode
Function : Operation is performed on the memory which address is specified by first and second operand.
Instructions : **ADC, AND, ASL, BIT, CMP, CPX, CPY, DEC, EOR, INC, JMP, JSR, LDA, LDX, LDY, LSR, ORA, ROL, ROR, SBC, STA, STX, STY**
Example : Mnemonic Machine code
ADC $AD12 **$6D_{16}$ 12_{16} AD_{16}**



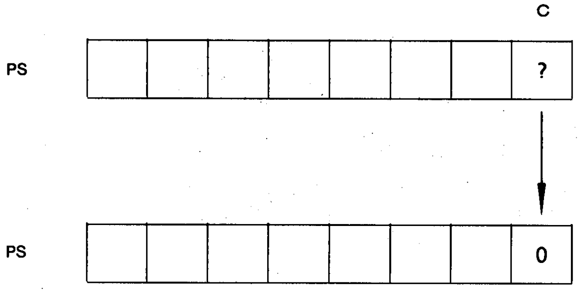
Name	: Absolute X addressing mode
Function	: Operation is performed on the memory which address is specified by adding the contents of index register X and value indicated first and second operand.
Instructions	: **ADC, AND, ASL, CMP, DEC, EOR, INC, LDA, LDY, LSR, ORA, ROL, ROR, SBC, STA**
Example	: Mnemonic / Machine code
	ADC $AD12,X / **$7D_{16}$ 12_{16} AD_{16}**



Name	: Absolute Y addressing mode
Function	: Operation is performed on the memory which address is specified by adding the contents of index register Y and value indicated first and second operand.
Instructions	: **ADC, AND, CMP, EOR, LDA, LDX, ORA, SBC, STA**
Example	: Mnemonic / Machine code
	ADC $AD12,Y / **$79_{16}$ 12_{16} AD_{16}**



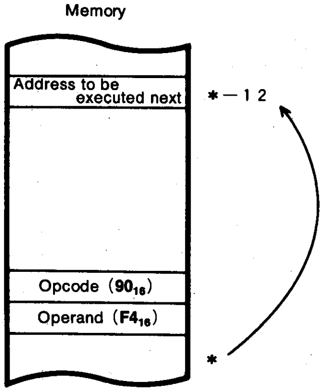
Name : **Implied addressing mode**
Function : Implied addressing mode need no operand.
Instructions : **BRK, CLC, CLD, CLI, CLT, CLV, DEX, DEY, FST, INX, INY, NOP, PHA, PHP, PLA, PLP, RTI, RTS, SEC, SED, SEI, SET, SLW, STP, TAX, TSX, TAY, TXA, TYA**
Example : Mnemonic   Machine code
CLC   **18_{16}**



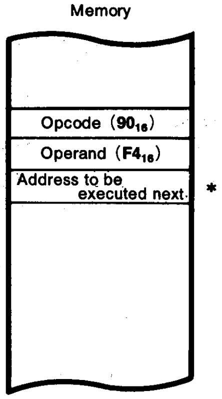
Carry flag reset

Name : **Relative addressing mode**
Function : Jumps to address which is produced by adding the contents of program counter and the contents of operand.
Instructions : **BCC, BCS, BEQ, BMI, BNE, BPL, BRA, BVC, BVS**
Example : Mnemonic   Machine code
BCC $*-12$ **90_{16} $F4_{16}$**

Jumps to − 12 address when carry flag(c) is cleared.



Proceed to next address when carry flag(c) is set.

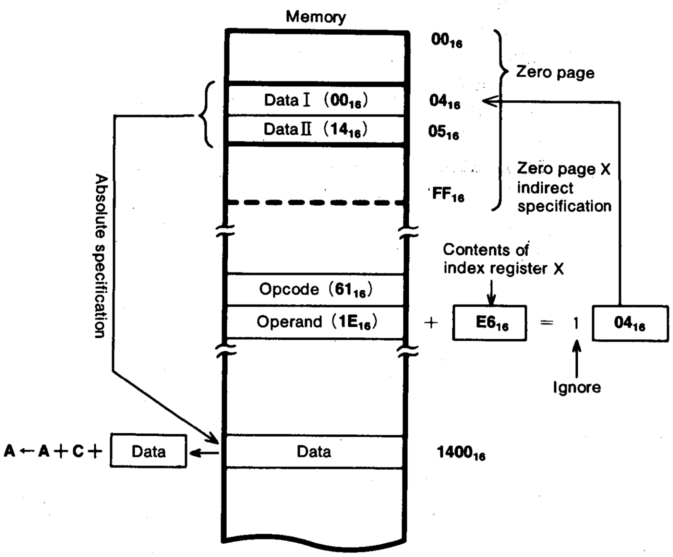


Name : Indirect X addressing mode

Function : Operation is performed on the memory at address indicated by contents of consecutive 2 byte memory which first address is formed by adding operand and contents of index register X.

Instructions : **ADC, AND, CMP, EOR, LDA, ORA, SBC, STA**

Example : Mnemonic Machine code
ADC ($1E,X) **$61_{16}$ $1E_{16}$**



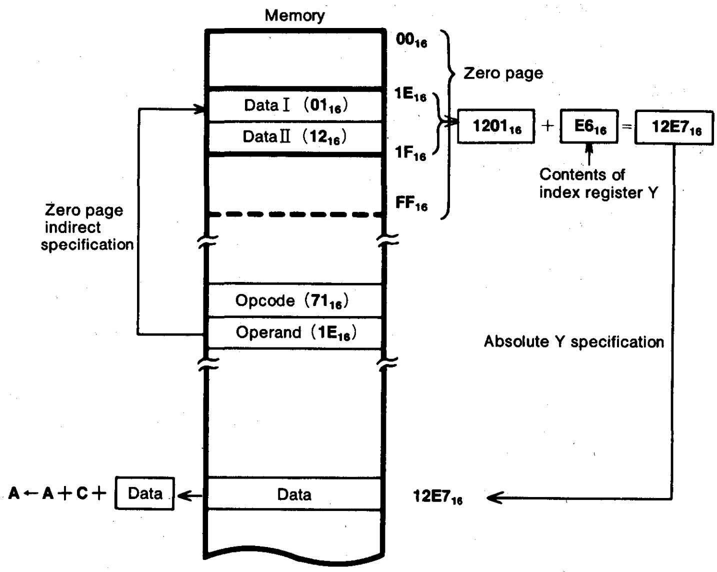
In this example, 00_{16} as data I and 14_{16} as data II have been stored beforehand.

Name : Indirect Y addressing mode

Function : Operation is performed on the memory addressed by adding the contents of index register Y and contents of consecutive 2 byte zero page memory which first address is specified by operand.

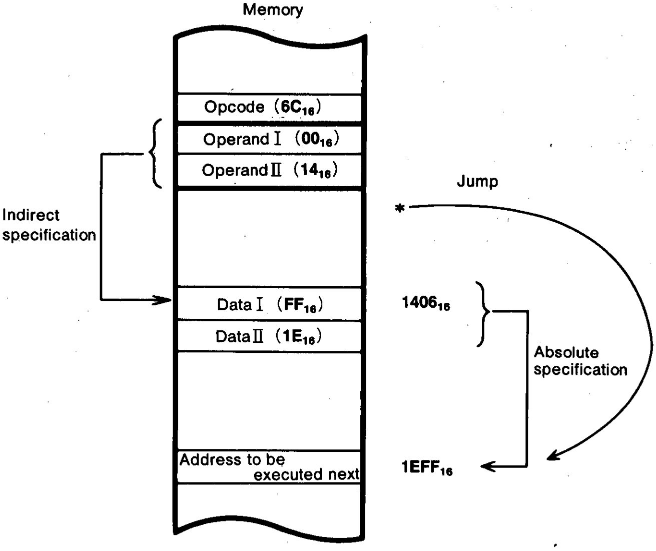
Instructions : **ADC, AND, CMP, EOR, LDA, ORA, SBC, STA**

Example : Mnemonic Machine code
ADC ($1E),Y **$71_{16}$ $1E_{16}$**



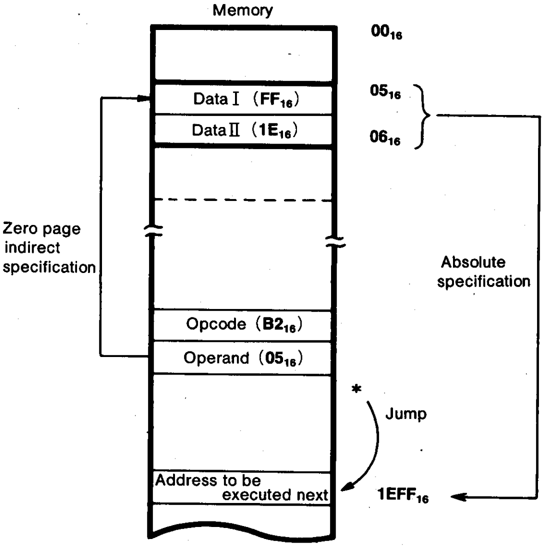
In this example, 00_{16} as data I and 12_{16} as Data II have been stored beforehand.

Name	: Indirect absolute addressing mode
Function	: Specifies consecutive 2-byte memories by contents of first and second operand and jumps to address indicated by contents of these memories.
Instructions	: **JMP, JSR**
Example	: Mnemonic　　Machine code
	JMP $1400　　**$6C_{16}$ 00_{16} 14_{16}**



In this example, FF_{16} as data I and $1E_{16}$ as data II have been stored beforehand.

Name	: Zero page indirect absolute addressing mode
Function	: Specifies consecutive 2-byte memories in zero page area by operand contents and jumps to adddress indicated by contents of these memories.
Instructions	: **JMP, JSR**
Example	: Mnemonic　　Machine code
	JMP $05　　**$B2_{16}$ 05_{16}**



In this example, FF_{16} as data I and $1E_{16}$ as data II have been stored beforehand.

Name : **Special page addressing mode**

Function : Jumps to address in special page area. 8 high-order address and 8 low-order address of jump distination is $1F_{16}$ and contents of operand respectively.

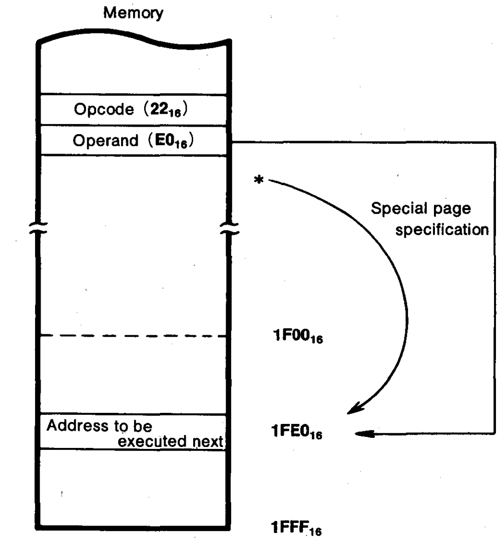
Instruction : **JSR**

Example : Mnemonic Machine code

JSR ¥$1FE0 **$22_{16}$ $E0_{16}$**

↑

*This symbol denotes special page mode.



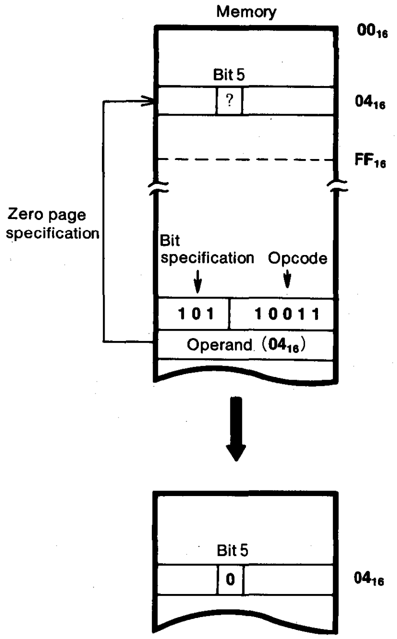
Name : Zero page bit addressing mode

Function : Operation is performed on the bit specified by 3 high-order bits of opcode, memory address containing this bit is specified by operand.

Instructions : **CLB, SEB**

Example : Mnemonic Machine code

CLB 5,$04 **$BF_{16}$ 04_{16}**



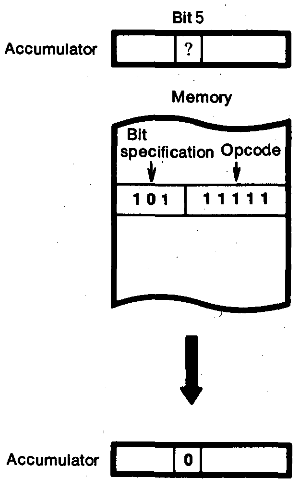
Name : Accumulator bit addressing mode

Function : Specifies bit in accumulator by 3 high-order bits of opcode.

Instructions : **CLB, SEB**

Example : Mnemonic Machine code

CLB 5,A **BB$_{16}$**



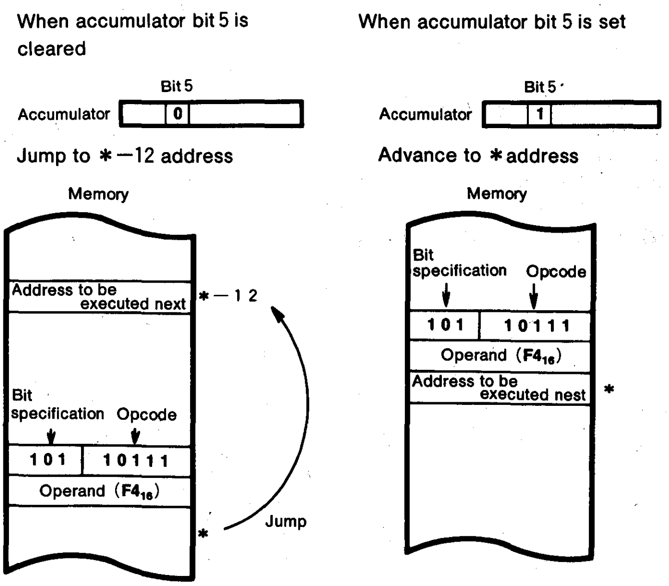
Name : Zero page bit addressing mode

Function : Operation is performed on the bit specified by 3 high-order bits of opcode, memory address containing this bit is specified by operand.

Instructions : **CLB, SEB**

Example : Mnemonic Machine code

CLB 5,$04 **BF$_{16}$ 04$_{16}$**



Name : **Zero page bit relative addressing mode**

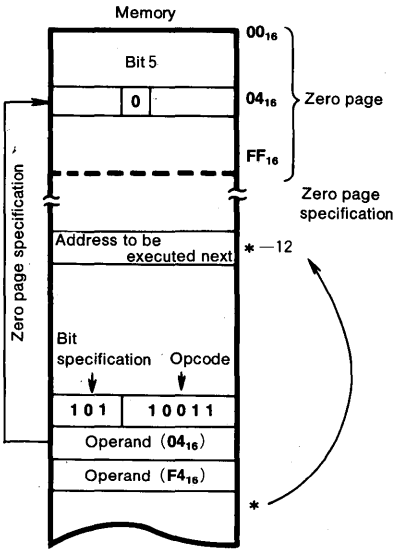
Function : Operation is performed on the bit specified by 3 high-order bits of opcode, memory address containing this bit is specified by first operand and, depending on the state of this special bit, jumps to the address indicated by the value produced by adding the second operand contents to the contents of the program counter.

Instructions : **BBC, BBS**

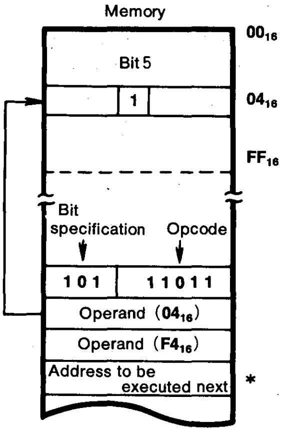
Example : Mnemonic Machine code

BBC 5,$04,−12 **B7$_{16}$ 04$_{16}$ F4$_{16}$**

Jump to ∗ − 12 address when 04$_{16}$ address bit 5 is cleared.



Advance to ∗ address when 04$_{16}$ address bit 5 is set.



Documentation Required for Ordering a Custom Mask

The following information should be provided when ordering a custom mask:

(1) M50740-XXXSP mask confirmation sheet

(2) ROM data EPROM 3 sets

Programming Precautions

(1) The frequency division ratio of the timers and prescalers is not 1/(n+1) but 1/(n+2).

(2) Select any numerical value except 0 for the set values of the timers and prescalers.

(3) Even when the BBC or BBS instruction is executed immediately after the contents of the interrupt request bit has been changed by the program, the execution is still valid for the contents prior to the change. This means that for execution keyed to the contents subsequent to the change, the instruction should be executed after one of more instructions.

(4) Data should be read from the timers and prescalers while there is no change in the prescaler input.

(5) The decimal mode flag D is set to "1" and the ADC or SBC instruction is executed with decimal arithmetic and logic operations. In this case, the SEC or CLC instruction should be executed after one or more instructions from the ADC or SBC instruction.

MACHINE INSTRUCTIONS

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT,A			ZP			BIT,ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
ADC (Note 1)	When T=0 A←A+M+C When T=1 M(X)←M(X)+M+C	Adds the carry, accumulator and memory contents. The results are entered into the accumulator. Adds the contents of the memory in the address indicated by index register X, the contents of the memory specified by the addressing modes in the columns on the right, and the contents of the carry. The results are entered into the memory at the address indicated by index register X.				69	2	2							65	3	2			
AND (Note 1)	When T=0 A←A∧M When T=1 M(X)←M(X)∧M	"AND-s" the accumulator and memory contents. The results are entered into the accumulator. "AND-s" the contents of the memory of the address indicated by index register X and the contents of the memory specified by the addressing modes in the columns on the right. The results are entered into the memory at the address indicated by index register X.				29	2	2							25	3	2			
ASL	7 0 C ← □ ←0	1-bit shifts the contents of accumulator or contents of memory to the left. "0" enters 0th bit of memory or accumulator and the contents of the 7th bit enter carry flag.							0A	2	1				06	5	2			
BBC (Note 4)	A_b or M_b=0?	Branches when the contents of the bit specified in the accumulator or memory are "0".										13+2i	4	2				17+2i	5	3
BBS (Note 4)	A_b or M_b=1?	Branches when the contents of the bit specified in the accumulator or memory are "1".										03+2i	4	2				07+2i	5	3
BCC (Note 4)	C=0?	Branches when the contents of carry flag are "0".																		
BCS (Note 4)	C=1?	Branches when the contents of carry flag are "1".																		
BEQ (Note 4)	Z=1?	Branches when the contents of zero flag are "1".																		
BIT	A∧M	"AND-s" the contents of accumulator and memory. The results are not entered anywhere.													24	3	2			
BMI (Note 4)	N=1?	Branches when the contents of negative flag are "1".																		
BNE (Note 4)	Z=0?	Branches when the contents of zero flag are "0".																		
BPL (Note 4)	N=0?	Branches when the contents of negative flag are "0".																		
BRA	PC←PC±offset	Jumps to address where offset has been added to the program counter.																		
BRK	B←1 $M(S) \leftarrow PC_H$ S←S−1 $M(S) \leftarrow PC_L$ S←S−1 M(S)←PS S←S−1 $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$	Executes software interrupt.	00	7	1															

Addressing mode																																	Processor status register							
ZP,X			ZP,Y			ABS			ABS,X			ABS,Y			IND			ZP,IND			IND,X			IND,Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
75	4	2				6D	4	3	7D	5	3	79	5	3							61	6	2	71	6	2							N	V	•	•	•	•	Z	C
35	4	2				2D	4	3	3D	5	3	39	5	3							21	6	2	31	6	2							N	•	•	•	•	•	Z	•
16	6	2				0E	6	3	1E	7	3																						N	•	•	•	•	•	Z	C
																																	•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
																											90	2	2				•	•	•	•	•	•	•	•
																											B0	2	2				•	•	•	•	•	•	•	•
																											F0	2	2				•	•	•	•	•	•	•	•
						2C	4	3																									M_7	M_6	•	•	•	•	Z	•
																											30	2	2				•	•	•	•	•	•	•	•
																											D0	2	2				•	•	•	•	•	•	•	•
																											10	2	2				•	•	•	•	•	•	•	•
																											80	4	2				•	•	•	•	•	•	•	•
																																	•	•	•	1	•	1	•	•

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT,A			ZP			BIT,ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
BVC (Note 4)	V=0?	Branches when the contents of overflow flag are "0."																		
BVS (Note 4)	V=1?	Branches when the contents of overflow flag are "1."																		
CLB	A_b or M_b←0	Clears the contents of the bit specified in the accumulator or memory to "0."										1B + 2i	2	1				1F + 2i	5	2
CLC	C←0	Clears the contents of the carry flag to "0."	18	2	1															
CLD	D←0	Clears the contents of decimal mode flag to "0."	D8	2	1															
CLI	I←0	Clears the contents of interrupt disable flag to "0."	58	2	1															
CLT	T←0	Clears the contents of X-modified arithmetic mode flag to "0."	12	2	1															
CLV	V←0	Clears the contents overflow flag to "0."	B8	2	1															
CMP (Note 3)	When T=0 A−M When T=1 M(X)−M	Compares the contents of accumulator and memory. Compares the contents of the memory specified by addressing modes in the columns on the right with the contents of the address indicated by index register X.				C9	2	2							C5	3	2			
COM	M←$\overline{M}$	Formes one's complement of contents of memory, and store it into memory.													44	5	2			
CPX	X−M	Compares the contents of index register X and memory.				E0	2	2							E4	3	2			
CPY	Y−M	Compares the contents of index register Y and memory.				C0	2	2							C4	3	2			
DEC	A←A−1 or M←M−1	Decrements the contents of accumulator or memory by 1.							1A	2	1				C6	5	2			
DEX	X←X−1	Decrements the contents of index register X by 1.	CA	2	1															
DEY	Y←Y−1	Decrements the contents of index register Y by 1.	88	2	1															
EOR (Note 1)	When T=0 $A \leftarrow A \veebar M$ When T=1 $M(X) \leftarrow M(X) \veebar M$	"Exclusive-ORs" the contents of accumulator and memory. The results are stored into the accumulator. "Exclusive-ORs" the contents of the memory specified by the addressing modes in the columns on the right and the contents of the memory at the address indicated by index register X. The results are stored into the memory at the address indicated by index register X.				49	2	2							45	3	2			
FST		Connects oscillator output to X_{OUTF}.	E2	2	1															
INC	A←A+1 or M←M+1	Increments the contents of accumulator or memory by 1.							3A	2	1				E6	5	2			
INX	X←X+1	Increments the contents of index register X by 1.	E8	2	1															
INY	Y←Y+1	Increments the contents of index register Y by 1.	C8	2	1															

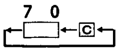
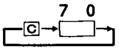
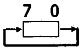
Addressing mode																																	Processor status register							
ZP,X			ZP,Y			ABS			ABS,X			ABS,Y			IND			ZP,IND			IND,X			IND,Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
																											50	2	2				•	•	•	•	•	•	•	•
																											70	2	2				•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	0
																																	•	•	•	•	0	•	•	•
																																	•	•	•	•	•	0	•	•
																																	•	•	0	•	•	•	•	•
																																	•	0	•	•	•	•	•	•
D5	4	2				CD	4	3	DD	5	3	D9	5	3							C1	6	2	D1	6	2							N	•	•	•	•	•	Z	C
																																	N	•	•	•	•	•	Z	•
						EC	4	3																									N	•	•	•	•	•	Z	C
						CC	4	3																									N	•	•	•	•	•	Z	C
D6	6	2				CE	6	3	DE	7	3																						N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
55	4	2				4D	4	3	5D	5	3	59	5	3							41	6	2	51	6	2							N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•
F6	6	2				EE	6	3	FE	7	3																						N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•

3

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT,A			ZP			BIT,ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
JMP	If addressing mode is ABS $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ If addressing mode is IND $PC_L \leftarrow (AD_H, AD_L)$ $PC_H \leftarrow (AD_H, AD_L+1)$ If addressing mode is ZP, IND $PC_L \leftarrow (00, AD_L)$ $PC_H \leftarrow (00, AD_L+1)$	Jumps to new address.																		
JSR	$M(S) \leftarrow PC_H$ $S \leftarrow S-1$ $M(S) \leftarrow PC_L$ $S \leftarrow S-1$ After executing the above, if addressing mode is ABS, $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ If addressing mode is SP, $PC_L \leftarrow AD_L$ $PC_H \leftarrow FF$ If addressing mode is ZP, IND, $PC_L \leftarrow (00, AD_L)$ $PC_H \leftarrow (00, AD_L+1)$	After storing contents of program counter in stack, and jumps to new address.																		
LDA (Note 2)	When T=0 $A \leftarrow M$ When T=1 $M(X) \leftarrow M$	Load accumulator with contents of memory. Load memory indicated by index register X with contents of memory specified by addressing mode shown in right column.				A9	2	2							A5	3	2			
LDM	$M \leftarrow IMM$	Load memory with immediate value.													3C	4	3			
LDX	$X \leftarrow M$	Load index register X with contents of memory.				A2	2	2							A6	3	2			
LDY	$Y \leftarrow M$	Load index register Y with contents of memory.				A0	2	2							A4	3	2			
LSR	7 0 0 → □ → C	Shift the contents of accumulator or memory to the right by one bit. 0th bit of accumulator or memory is stored in carry, 7th bit is cleared.							4A	2	1				46	5	2			
NOP	$PC \leftarrow PC+1$	No operation.	EA	2	1															
ORA (Note 1)	When T=0 $A \leftarrow A \vee M$ When T=1 $M(X) \leftarrow M(X) \vee M$	Produce the logical OR of the contents of memory and accumulator. The result is stored in accumulator. produce the logical OR of contents of memory indicated by index register X and contents of memory specified by addressing mode shown in right column. The result is stored in memory of address specified by index register X.				09	2	2							05	3	2			

Addressing mode																																	Processor status register							
ZP,X			ZP,Y			ABS			ABS,X			ABS,Y			IND			ZP,IND			IND,X			IND,Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
						4C	3	3							6C	5	3	B2	4	2													•	•	•	•	•	•	•	•
						20	6	3										02	7	2										22	5	2	•	•	•	•	•	•	•	•
B5	4	2				AD	4	3	BD	5	3	B9	5	3							A1	6	2	B1	6	2							N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•
			B6	4	2	AE	4	3				BE	5	3																			N	•	•	•	•	•	Z	•
B4	4	2				AC	4	3	BC	5	3																						N	•	•	•	•	•	Z	•
56	6	2				4E	6	3	5E	7	3																						0	•	•	•	•	•	Z	C
																																	•	•	•	•	•	•	•	•
15	4	2				0D	4	3	1D	5	3	19	5	3							01	6	2	11	6	2							N	•	•	•	•	•	Z	•

3

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT,A			ZP			BIT,ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
PHA	M(S)←A S←S−1	Saves the contents of the accumulator in the memory at the address indicated by the stack pointer and decrements the contents of stack pointer by 1.	48	3	1															
PHP	M(S)←PS S←S−1	Saves the contents of processor status register in the memory at the address indicated by the stack pointer and decrements the contents of stack pointer by 1.	08	3	1															
PLA	S←S+1 A←M(S)	Increments the contents of stack pointer by 1 and pulls from the memory at the address indicated by the stack pointer, and store it in accumulator.	68	4	1															
PLP	S←S+1 PS←M(S)	Increments the contents of stack pointer by 1 and pulls from the memory at the address indicated by the stack pointer, and store it in processor status register.	28	4	1															
ROL		Connects the carry flag and the accumulator or memory and rotates the contents to the left by 1 bit.							2A	2	1				26	5	2			
ROR		Connects the carry flag and the accumulator or memory and rotates the contents to the right by 1 bit.							6A	2	1				66	5	2			
RRF		Rotates the contents of memory to the right by 4 bits.													82	8	2			
RTI	S←S+1 PS←M(S) S←S+1 PC_L←M(S) S←S+1 PC_H←M(S)	Returns from the interrupt routine to the main routine.	40	6	1															
RTS	S←S+1 PC_L←M(S) S←S+1 PC_H←M(S)	Returns from the subroutine to the main routine.	60	6	1															
SBC (Note 1)	When T=0 A←A−M−C When T=1 M(X)←M(X)−M−C	Subtracts the contents of memory and carry flag from the contents of accumulator. The results are stored into the accumulator. Subtracts contents of carry flag and contents of the memory indicated by the addressing modes shown in the columns on the right from the memory at the address indicated by index register X. The results are stored into the memory of the address indicated by index register X.				E9	2	2							E5	3	2			
SEB	A_b or M_b←1	Sets the specified bit contents of accumulator or memory to "1."										0B + 2i	2	1				0F + 2i	5	2
SEC	C←1	Sets the contents of carry flag to "1."	38	2	1															
SED	D←1	Sets the contents of decimal mode flag to "1."	F8	2	1															
SEI	I←1	Sets the contents of interrupt disable flag to "1."	78	2	1															
SET	T←1	Sets the contents of X-modified arithmetic mode flag to "1."	32	2	1															
SLW		Releases the connection between the oscillator output and pin X_{OUTF}.	C2	2	1															

Addressing mode																																	Processor status register							
ZP,X			ZP,Y			ABS			ABS,X			ABS,Y			IND			ZP,IND			IND,X			IND,Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
																																	•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
																																	N	•	•	•	•	•	Z	•
																																	(Value saved in stack)							
36	6	2				2E	6	3	3E	7	3																						N	•	•	•	•	•	Z	C
76	6	2				6E	6	3	7E	7	3																						N	•	•	•	•	•	Z	C
																																	•	•	•	•	•	•	•	•
																																	(Value saved in stack)							
																																	•	•	•	•	•	•	•	•
F5	4	2				ED	4	3	FD	5	3	F9	5	3							E1	6	2	F1	6	2							N	V	•	•	•	•	Z	C
																																	•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	1
																																	•	•	•	•	1	•	•	•
																																	•	•	•	•	•	1	•	•
																																	•	•	1	•	•	•	•	•
																																	•	•	•	•	•	•	•	•

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT,A			ZP			BIT,ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
STA	M←A	Stores the contents of accumulator in the memory.													85	4	2			
STP		Stops the oscillation of the oscillator.	42	2	1															
STX	M←X	Stores the contents of index register X in the memory.													86	4	2			
STY	M←Y	Stores the contents of index register Y in the memory.													84	4	2			
TAX	X←A	Transfers the contents of accumulator to index register X.	AA	2	1															
TAY	Y←A	Transfers the contents of accumulator to index register Y.	A8	2	1															
TST	M=0?	Tests whether the contents of memory are "0" or not.													64	3	2			
TSX	X←S	Transfers the contents of stack pointer to index register X.	BA	2	1															
TXA	A←X	Transfers the contents of index register X to the accumulator.	8A	2	1															
TXS	S←X	Transfers the contents of index register X to the stack pointer.	9A	2	1															
TYA	A←Y	Transfers the contents of index register Y to the accumulator.	98	2	1															

Note 1 : The number of cycles "n" is added by 3 when T is 1.
2 : The number of cycles "n" is added by 2 when T is 1.
3 : The number of cycles "n" is added by 1 when T is 1.
4 : The number of cycles "n" is added by 2 when branching has occurred.

Symbol	Contents	Symbol	Contents
IMP	Implied addressing mode	+	Addition
IMM	Immediate addressing mode	−	Subtraction
A	Accumulator or accumulator addressing mode	∧	AND
		V	OR
BIT, A	Accumulator bit relative addressing mode	⊻	Exclusive-OR
		—	Negation
ZP	Zero page addressing mode	←	Shows direction of data flow
BIT, ZP	Zero page bit relative addressing mode	X	Index register X
		Y	Index register Y
ZP, X	Zero page X addressing mode	S	Stack pointer
ZP, Y	Zero page Y addressing mode	PC	Program counter
ABS	Absolute addressing mode	PS	Processor status register
ABS, X	Absolute X addressing mode	PC_H	8 high-order bits of program counter
ABS, Y	Absolute Y addressing mode	PC_L	8 low-order bits of program counter
IND	Indirect absolute addressing mode	AD_H	8 high-order bits of address
		AD_L	8 low-order bits of address
ZP, IND	Zero page indirect absolute addressing mode	(AD_H, AD_L)	Contents of memory at address indicated by AD_H and AD_L, in AD_H is 8 high-order bits and AD_L is low-order bits.
IND, X	Indirect X addressing mode	$(00, AD_L)$	Contents of address indicated by zero page AD_L
IND, Y	Indirect Y addressing mode	FF	FF in Hexadecimal notation
REL	Relative addressing mode	M	Memory specified by address designation of any addressing mode
SP	Special page addressing mode		
C	Carry flag	M (X)	Memory of address indicated by contents of index register X
Z	Zero flag		
I	Interrupt disable flag	M (S)	Memory of address indicated by contents of stack pointer
D	Decimal mode flag	A_b	1 bit of accumulator
B	Break flag	M_b	1 bit of memory
T	X-modified arithmetic mode flag	OP	Opcode
V	Overflow flag	n	Number of cycles
N	Negative flag	#	Number of bytes

Addressing mode																																	Processor status register							
ZP,X			ZP,Y			ABS			ABS,X			ABS,Y			IND			ZP,IND			IND,X			IND,Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
95	5	2				8D	5	3	9D	6	3	99	6	3							81	7	2	91	7	2							•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
			96	5	2	8E	5	3																									•	•	•	•	•	•	•	•
94	5	2				8C	5	3																									•	•	•	•	•	•	•	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•
																																	N	•	•	•	•	•	Z	•

LIST OF INSTRUCTION CODES

	$D_3 \sim D_0$	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
$D_7 \sim D_4$	Hexadecimal notation	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	BRK	ORA IND, X	JSR ZP, IND	BBS 0, A	—	ORA ZP	ASL ZP	BBS 0, ZP	PHP	ORA IMM	ASL A	SEB 0, A	—	ORA ABS	ASL ABS	SEB 0, ZP
0001	1	BPL	ORA IND, Y	CLT	BBC 0, A	—	ORA ZP, X	ASL ZP, X	BBC 0, ZP	CLC	ORA ABS, Y	DEC A	CLB 0, A	—	ORA ABS, X	ASL ABS, X	CLB 0, ZP
0010	2	JSR ABS	AND IND, X	JSR SP	BBS 1, A	BIT ZP	AND ZP	ROL ZP	BBS 1, ZP	PLP	AND IMM	ROL A	SEB 1, A	BIT ABS	AND ABS	ROL ABS	SEB 1, ZP
0011	3	BMI	AND IND, Y	SET	BBC 1, A	—	AND ZP, X	ROL ZP, X	BBC 1, ZP	SEC	AND ABS, Y	INC A	CLB 1, A	LDM ZP	AND ABS, X	ROL ABS, X	CLB 1, ZP
0100	4	RTI	EOR IND, X	STP	BBS 2, A	COM ZP	EOR ZP	LSR ZP	BBS 2, ZP	PHA	EOR IMM	LSR A	SEB 2, A	JMP ABS	EOR ABS	LSR ABS	SEB 2, ZP
0101	5	BVC	EOR IND, Y	—	BBC 2, A	—	EOR ZP, X	LSR ZP, X	BBC 2, ZP	CLI	EOR ABS, Y	—	CLB 2, A	—	EOR ABS, X	LSR ABS, X	CLB 2, ZP
0110	6	RTS	ADC IND, X	—	BBS 3, A	TST ZP	ADC ZP	ROR ZP	BBS 3, ZP	PLA	ADC IMM	ROR A	SEB 3, A	JMP IND	ADC ABS	ROR ABS	SEB 3, ZP
0111	7	BVS	ADC IND, Y	—	BBC 3, A	—	ADC ZP, X	ROR ZP, X	BBC 3, ZP	SEI	ADC ABS, Y	—	CLB 3, A	—	ADC ABS, X	ROR ABS, X	CLB 3, ZP
1000	8	BRA	STA IND, X	RRF ZP	BBS 4, A	STY ZP	STA ZP	STX ZP	BBS 4, ZP	DEY	—	TXA	SEB 4, A	STY ABS	STA ABS	STX ABS	SEB 4, ZP
1001	9	BCC	STA IND, Y	—	BBC 4, A	STY ZP, X	STA ZP, X	STX ZP, Y	BBC 4, ZP	TYA	STA ABS, Y	TXS	CLB 4, A	—	STA ABS, X	—	CLB 4, ZP
1010	A	LDY IMM	LDA IND, X	LDX IMM	BBS 5, A	LDY ZP	LDA ZP	LDX ZP	BBS 5, ZP	TAY	LDA IMM	TAX	SEB 5, A	LDY ABS	LDA ABS	LDX ABS	SEB 5, ZP
1011	B	BCS	LDA IND, Y	JMP ZP, IND	BBC 5, A	LDY ZP, X	LDA ZP, X	LDX ZP, Y	BBC 5, ZP	CLV	LDA ABS, Y	TSX	CLB 5, A	LDY ABS, X	LDA ABS, X	LDX ABS, Y	CLB 5, ZP
1100	C	CPY IMM	CMP IND, X	SLW	BBS 6, A	CPY ZP	CMP ZP	DEC ZP	BBS 6, ZP	INY	CMP IMM	DEX	SEB 6, A	CPY ABS	CMP ABS	DEC ABS	SEB 6, ZP
1101	D	BNE	CMP IND, Y	—	BBC 6, A	—	CMP ZP, X	DEC ZP, X	BBC 6, ZP	CLD	CMP ABS, Y	—	CLB 6, A	—	CMP ABS, X	DEC ABS, X	CLB 6, ZP
1110	E	CPX IMM	SBC IND, X	FST	BBS 7, A	CPX ZP	SBC ZP	INC ZP	BBS 7, ZP	INX	SBC IMM	NOP	SEB 7, A	CPX ABS	SBC ABS	INC ABS	SEB 7, ZP
1111	F	BEQ	SBC IND, Y	—	BBC 7, A	—	SBC ZP, X	INC ZP, X	BBC 7, ZP	SED	SBC ABS, Y	—	CLB 7, A	—	SBC ABS, X	INC ABS, X	CLB 7, ZP

▩ 3-byte instruction

▨ 2-byte instruction

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{CC}	Supply voltage	With respect to VSS; output transistors cut-off	$-0.3\sim7$	V
V_I	Input voltage, $R_0\sim R_3$, CNV_{SS}, $\overline{\text{RESET}}$, X_{IN}		$-0.3\sim7$	V
V_I	Input voltage, $P3_0\sim P3_7$		$-3.0\sim V_{CC}+0.3$	V
V_I	Input voltage, $\overline{\text{INT}}$ $P0_0\sim P0_7$, $P1_0\sim P1_7$, $P2_0\sim P2_7$, CNTR		$-0.3\sim13$	V
V_O	Output voltage, $R_0\sim R_3$		$-0.3\sim7$	V
V_O	Output voltage, $P3_0\sim P3_7$, X_{OUTF}, X_{OUTS}, ϕ, R/W, $\overline{\text{CE}}$, $\overline{\text{RESET}_{OUT}}$		$-0.3\sim V_{CC}+0.3$	V
V_O	Output voltage, $P0_0\sim P0_7$, $P1_0\sim P1_7$, $P2_0\sim P2_7$ CNTR		$-0.3\sim13$	V
P_d	Power dissipation	$T_a=25$℃	1000	mW
T_{opr}	Operating temperature		$-10\sim70$	℃
T_{stg}	Storage temperature		$-40\sim125$	℃

RECOMMENDED OPERATING CONDITIONS ($T_a=-10\sim70$℃, $V_{CC}=5V\pm10\%$, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min	Typ	Max	
V_{CC}	Supply voltage	4.5	5	5.5	V
V_{SS}	Supply voltage		0		V
V_{IH}	High-level input voltage, $P0_0\sim P0_7$, $P1_0\sim P1_7$, $P2_0\sim P2_7$, $P3_0\sim P3_7$, $R_0\sim R_3$, CNV_{SS}	$0.8V_{CC}$		V_{CC}	V
V_{IH}	High-level input voltage, CNTR, $\overline{\text{INT}}$	$0.8V_{CC}$		V_{CC}	V
V_{IH}	High-level input voltage, $\overline{\text{RESET}}$	$0.48V_{CC}$		V_{CC}	V
V_{IH}	High-level input voltage, X_{IN}	$0.8V_{CC}$		V_{CC}	V
V_{IL}	Low-level input voltage, $P0_0\sim P0_7$, $P1_0\sim P1_7$ $P2_0\sim P2_7$, $P3_0\sim P3_7$, $R_0\sim R_3$, CNV_{SS}	0		$0.2V_{CC}$	V
V_{IL}	Low-level input voltage, CNTR, $\overline{\text{INT}}$	0		$0.2V_{CC}$	V
V_{IL}	Low-level input voltage, $\overline{\text{RESET}}$	0		$0.12V_{CC}$	V
V_{IL}	Low-level input voltage, X_{IN}	0		$0.2V_{CC}$	V
$f_{(\phi)}$	Internal clock oscillation frequency			4	MHz

Note 1 : A high-level input voltage for ports P0, P1, P2, CNTR and $\overline{\text{INT}}$ of up to +12V may be supplied.

ELECTRICAL CHARACTERISTICS ($V_{CC}=5V\pm10\%$, $V_{SS}=0V$, $f_{(\phi)}=4MHz$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{OH}	High-level output voltage, $P3_0\sim P3_7$	$V_{CC}=5V$, $T_a=25$℃ $I_{OH}=-10mA$	3			V
V_{OH}	High-level output voltage, ϕ, R/W, $\overline{\text{CE}}$, $\overline{\text{RESET}_{OUT}}$	$V_{CC}=5V$, $T_a=25$℃ $I_{OH}=-2.5mA$	3			V
V_{OL}	Low-level output voltage, $P0_0\sim P0_7$, $P1_0\sim P1_7$, $P2_0\sim P2_7$ $R_0\sim R_3$, CNTR	$V_{CC}=5V$, $T_a=25$℃ $I_{OL}=10mA$			2	V
V_{OL}	Low-level output voltage, ϕ, R/W, $\overline{\text{CE}}$, $\overline{\text{RESET}_{OUT}}$	$V_{CC}=5V$, $T_a=25$℃ $I_{OL}=5mA$			2	V
$V_{T+}-V_{T-}$	Hysteresis, CNTR, $\overline{\text{INT}}$	$V_{CC}=5V$, $T_a=25$℃	0.3		1	V
$V_{T+}-V_{T-}$	Hysteresis, $\overline{\text{RESET}}$	$V_{CC}=5V$, $T_a=25$℃		0.5	0.7	V
$V_{T+}-V_{T-}$	Hysteresis, X_{IN}	$V_{CC}=5V$, $T_a=25$℃	0.1		0.5	V
I_{IL}	Input leakage current, $P0_0\sim P0_7$, $P1_0\sim P1_7$, $P2_0\sim P2_7$ $\overline{\text{INT}}$, CNTR	$V_{CC}=5V$, $T_a=25$℃ $0\leq V_I\leq12V$	-12		12	μA
I_{IL}	Input leakage current, $P3_0\sim P3_7$, $R_0\sim R_3$, CNV_{SS}, $\overline{\text{RESET}}$, X_{IN}	$V_{CC}=5V$, $T_a=25$℃ $0\leq V_I\leq5V$	-5		5	μA
I_{CC}	Supply current	$P3_0\sim P3_7$: V_{CC}, output pins open V_{SS} for all input and output pins except $P3_0\sim P3_7$ $V_{CC}=5V$ $T_a=25$℃		3	6	mA

TIMING REQUIREMENTS
SINGLE CHIP MODE (V_{CC} = 5V±10%, V_{SS} = 0V, T_a = 25℃, $f_{(\phi)}$ = 4MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
$t_{su\ (P0D-\phi)}$	Port P0 input setup time		270			ns
$t_{su\ (P1D-\phi)}$	Port P1 input setup time		270			ns
$t_{su\ (P2D-\phi)}$	Port P2 input setup time		270			ns
$t_{su\ (P3D-\phi)}$	Port P3 input setup time		270			ns
$t_{su\ (RD-\phi)}$	Port R input setup time		330			ns
$t_{h\ (\phi-P0D)}$	Port P0 input hold time		0			ns
$t_{h\ (\phi-P1D)}$	Port P1 input hold time		0			ns
$t_{h\ (\phi-P2D)}$	Port P2 input hold time		0			ns
$t_{h\ (\phi-P3D)}$	Port P3 input hold time		0			ns
$t_{h\ (\phi-RD)}$	Port R input hold time		0			ns
t_c	External clock input cycle time		250			ns
t_w	External clock input pulse width		75			ns
t_r	External clock rise time				25	ns
t_f	External clock fall time				25	ns

SWITCHING CHARACTERISTICS
SINGLE CHIP MODE (V_{CC} = 5V±10%, V_{SS} = 0V, T_a = 25℃, $f_{(\phi)}$ = 4MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
$t_{d(\phi-P0Q)}$	Port P0 data output delay time	Fig.16			230	ns
$t_{d(\phi-P1Q)}$	Port P1 data output delay time	Fig.16			230	ns
$t_{d(\phi-P2Q)}$	Port P2 data output delay time	Fig.16			230	ns
$t_{d(\phi-P3Q)}$	Port P3 data output delay time	Fig.17			200	ns
$t_{d(\phi-RA)}$	Port R address output delay time	Fig.16			200	ns
$t_{d(\phi-RAF)}$	Port R address output delay time	Fig.16	0		200	ns
$t_{d(\phi-RQ)}$	Port R data output delay time	Fig.16			200	ns
$t_{d(\phi-RQF)}$	Port R data output delay time	Fig.16			200	ns
$t_{d(\phi-CE)}$	$\overline{CE}$ output delay time	Fig.18			200	ns
$t_{d(\phi-RW)}$	R/W output delay time	Fig.18			100	ns

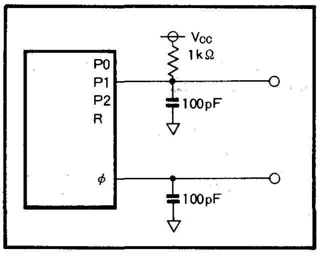


Fig.16 Port P0～P2, R test circuit

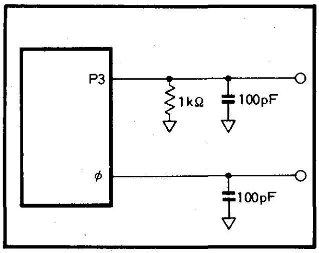


Fig.17 Port P3 test circuit

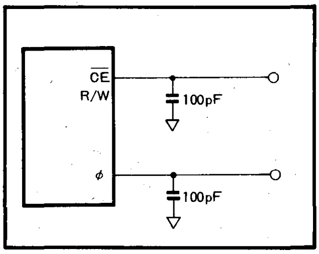
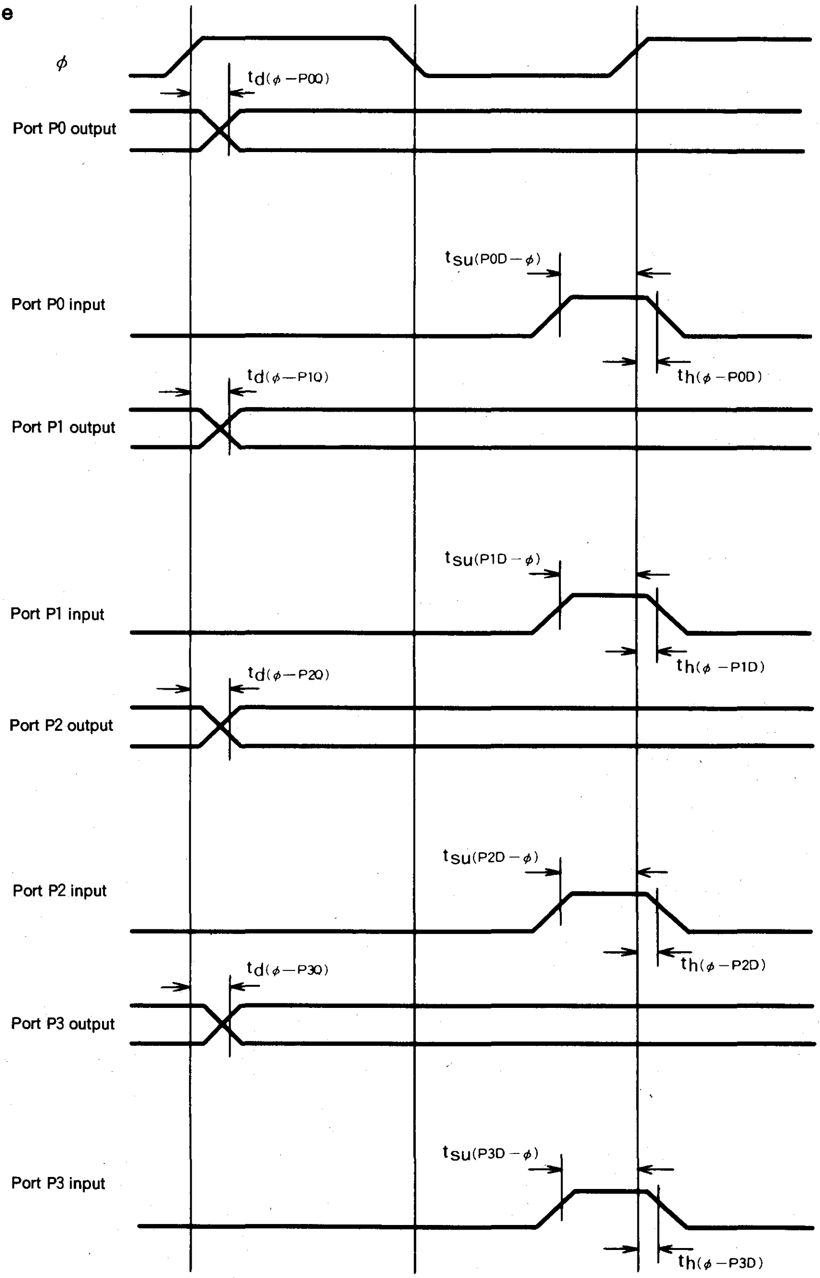


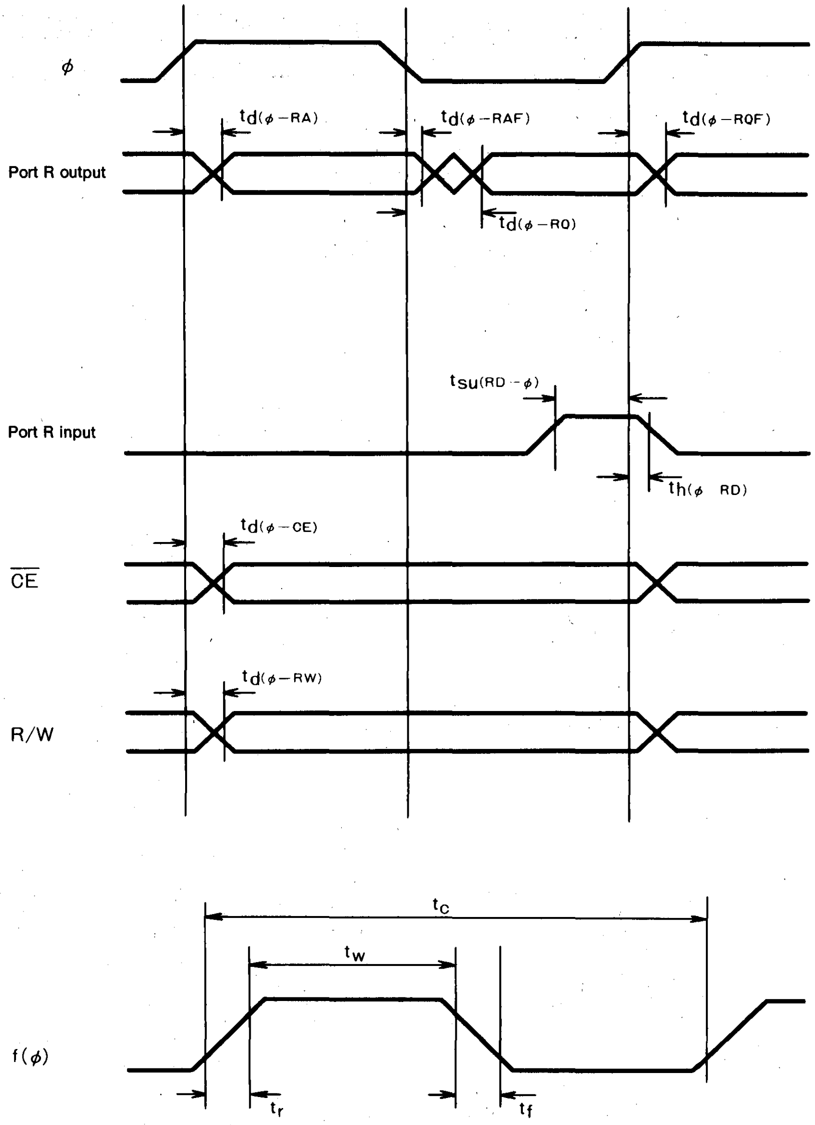
Fig.18 $\overline{CE}$, R/W test circuit

TIMING DIAGRAMS

Single chip mode



Single chip mode (continued evaluation)



MELPS 8-48 MICROCOMPUTERS

MELPS 8-48 MICROCOMPUTERS

FUNCTION OF MELPS 8-48 MICROCOMPUTERS

DESCRIPTION

The MELPS8-48 LSI is a family of cost-efficient single-chip microcomputers in which all of such necessary compoments as the CPU, ROM, RAM, input/output ports, timer etc. are integrated. The MELPS8-48 family consists of the following twelve members of different kinds and ROM/RAM capacities.

Each chip is provided with a timer and interrupt input and its I/O capabilities are simply expanded by use of I/O expanding chip M5L8243P or M5M82C43P (CMOS version), in addition the program memory can also be expanded to 4K bytes.

Each of M5L8048-XXXP, M5L8049-XXXP, M5L8049H-XXXP, M5M8050H-XXXP, M5M8050L-XXXP and M5M80C49-XXXP (CMOS version) has a built-in masked ROM and is suited for mass produciton. M5L8035LP has functions equivalent to M5L8048-XXXP, M5L8039P to M5L8049-XXXP, M5L8039HLP to M5L8049H-XXXP, M5M8040HP to M5M8050H-XXXP, M5M8040LP to M5M8050L-XXXP and M5M80C39P-6 to M5M80C49-XXXP where the program memory (ROM) is set externally.

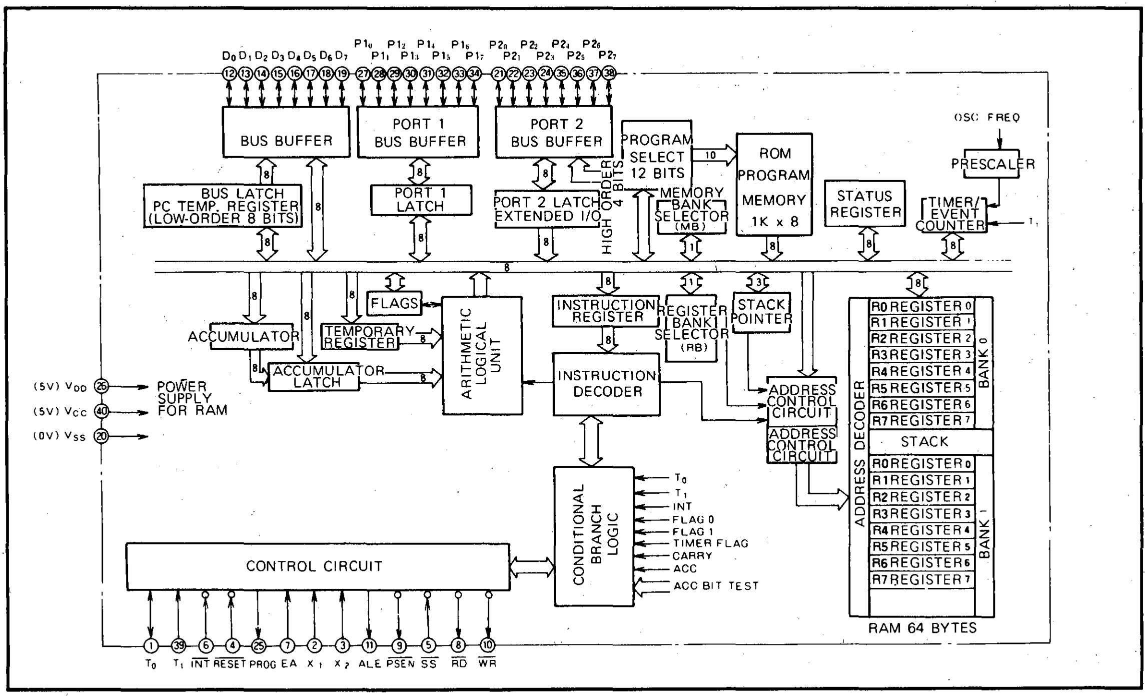
The family is provided with the MELPS8-48 cross assembler as a support for software development.

MELPS 8-48 single-chip microcomputer family

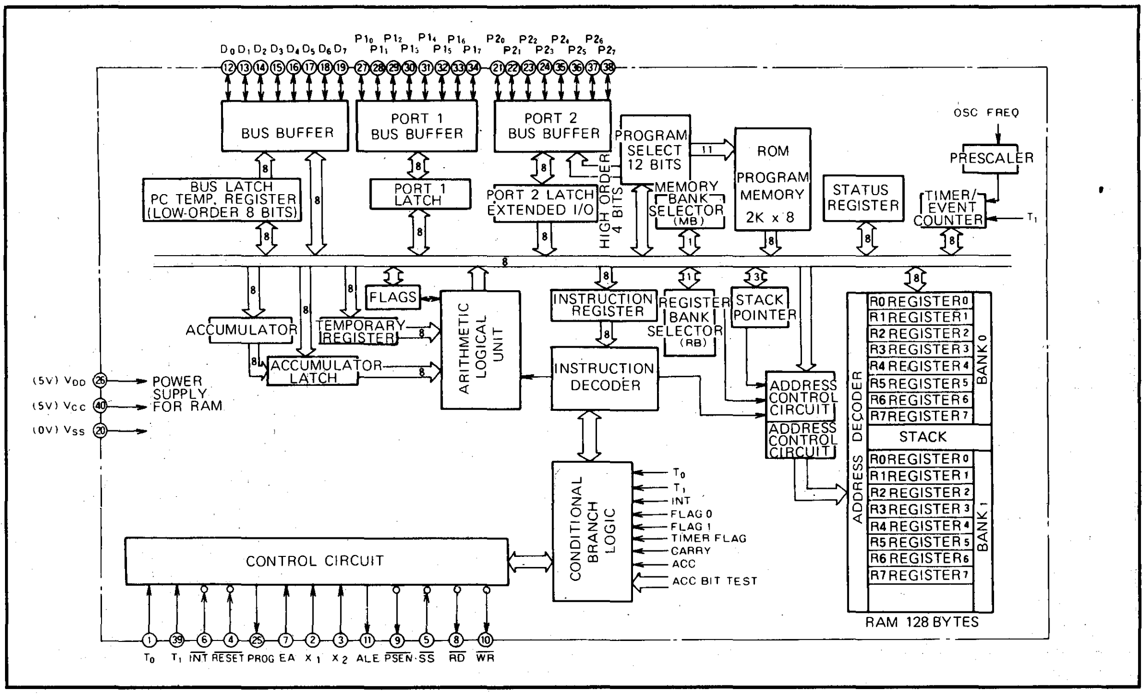
Symbol	Input clock (MHz)	Memory and input/output capacity			Structure
		ROM (Bytes)	RAM (Bytes)	I/O (Port)	
M5L8048-XXXP	6	1K	64	27	ED NMOS
M5L8035LP	6	External	64	27	ED NMOS
M5L8049-XXXP-6	6	2K	128	27	ED NMOS
M5L8049-XXXP-8	8	2K	128	27	ED NMOS
M5L8049-XXXP	8	2K	128	27	ED NMOS
M5L8039P-6	6	External	128	27	ED NMOS
M5L8039P-8	8	External	128	27	ED NMOS
M5L8039P-11	11	External	128	27	ED NMOS
M5L8049H-XXXP	11	2K	128	27	ED NMOS
M5L8039HLP	11	External	128	27	ED NMOS
M5M8050H-XXXP	11	4K	256	27	ED NMOS
M5M8040HP	11	External	256	27	ED NMOS
M5M8050L-XXXP	6	4K	256	27	ED NMOS
M5M8040L-XXXP	6	External	256	27	ED NMOS
M5M80C49-XXXP	6	2K	128	27	CMOS
M5M80C39P-6	6	External	128	27	CMOS

FUNCTION OF MELPS 8-48 MICROCOMPUTERS

M5L8048-XXXP Block Diagram

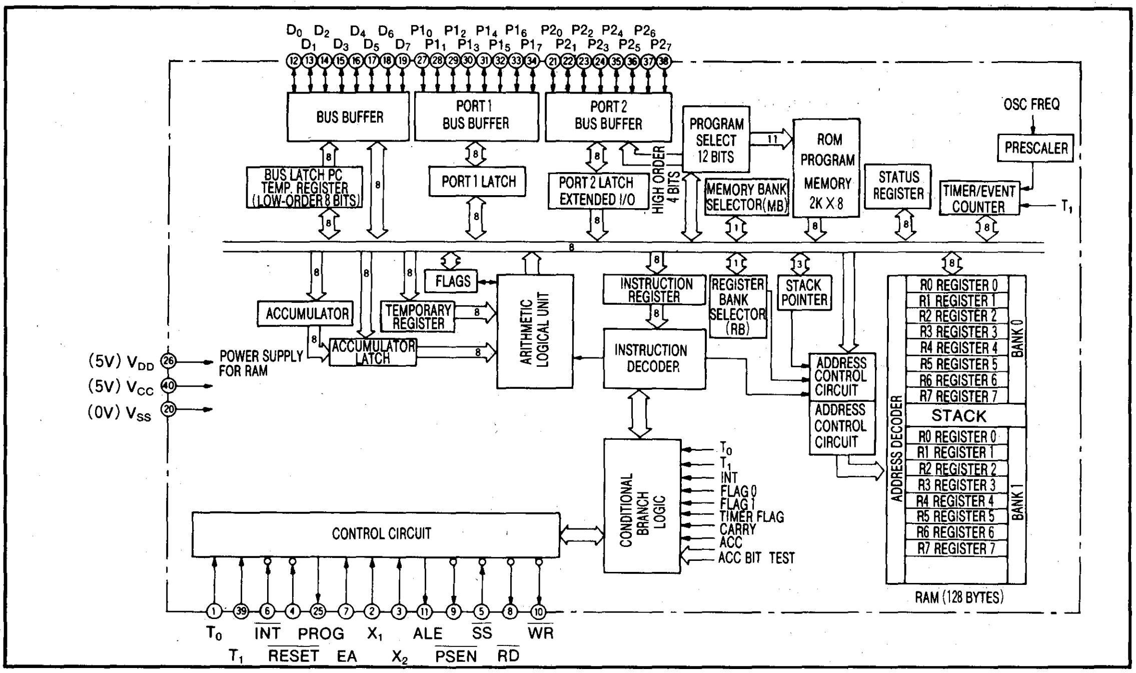


M5L8049-XXXP Block Diagram

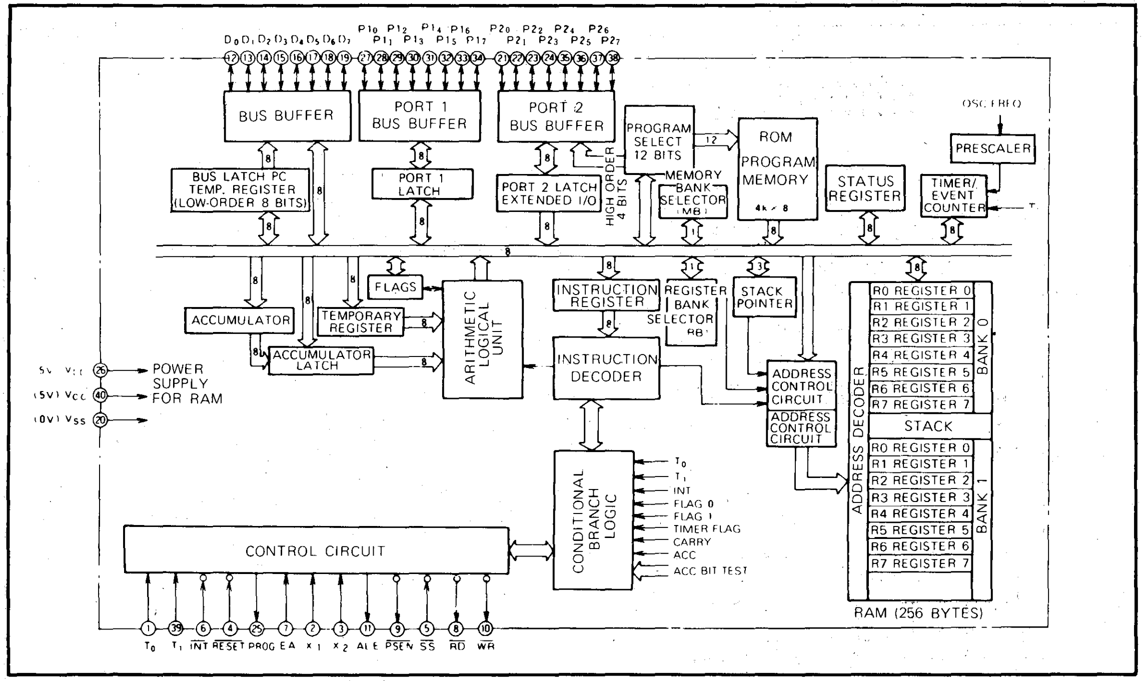


FUNCTION OF MELPS 8-48 MICROCOMPUTERS

M5L8049H-XXXP Block Diagram

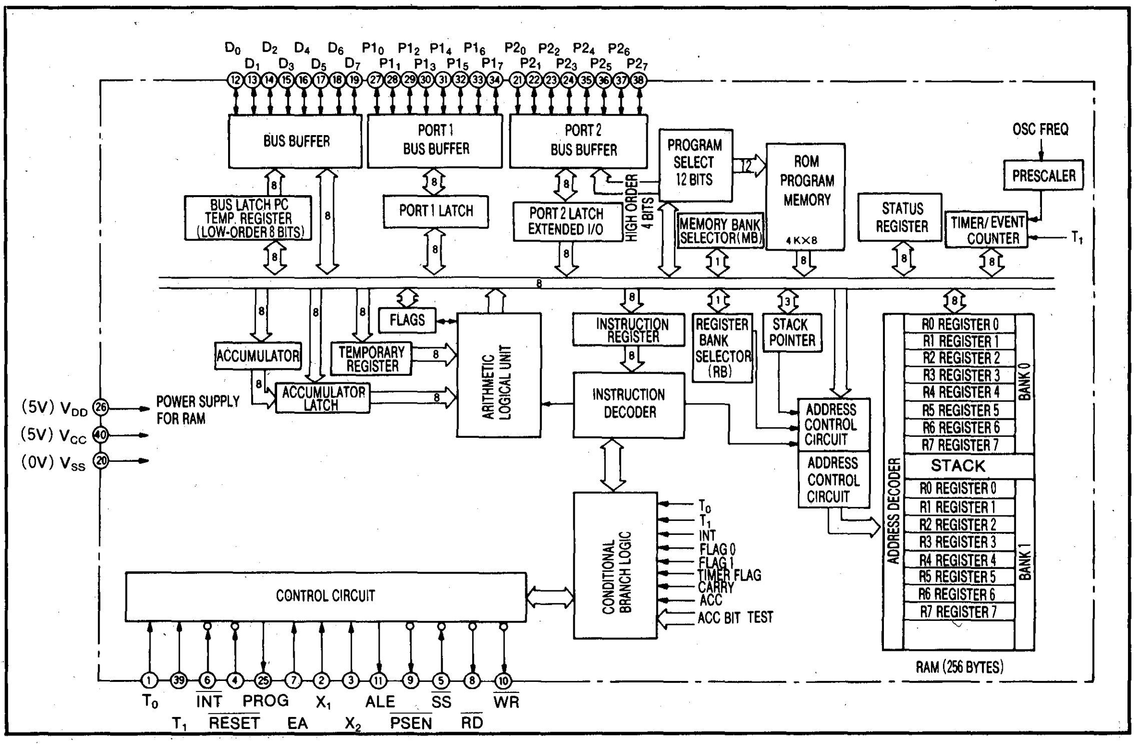


M5M8050H-XXXP Block Diagram

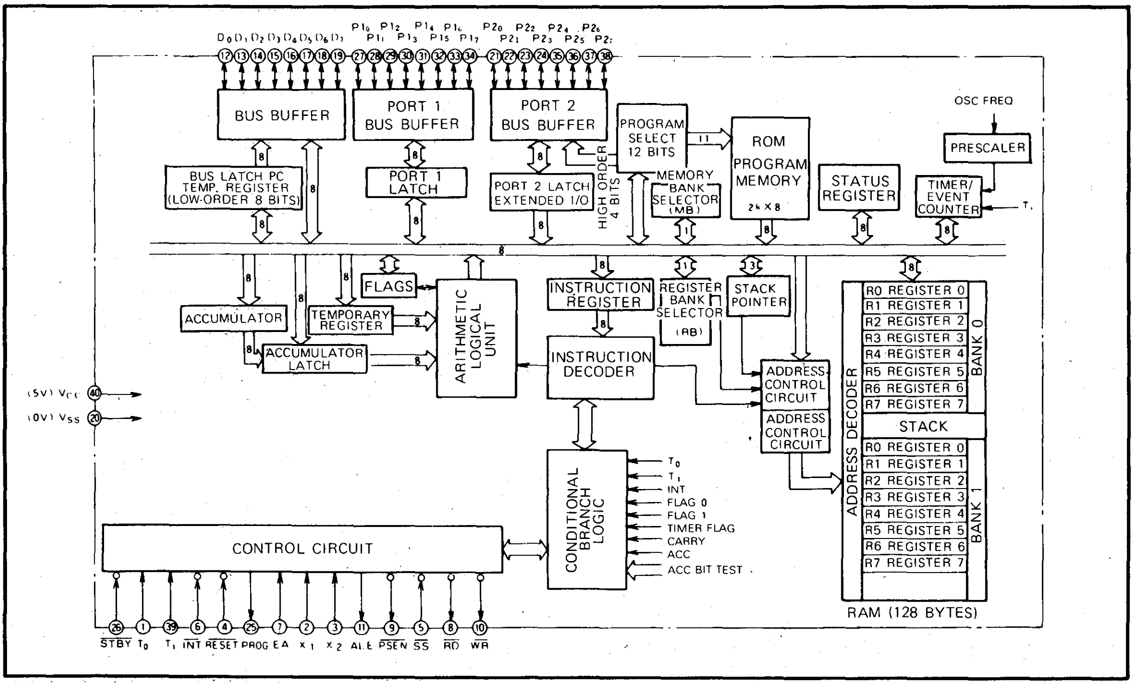


FUNCTION OF MELPS 8-48 MICROCOMPUTERS

M5M8050L-XXXP Block Diagram



M5M80C49-XXXP Block Diagram



BASIC FUNCTION BLOCKS

Program Memory (ROM)

The M5L8048-XXXP contain 1024 bytes of ROM. 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.

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 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.

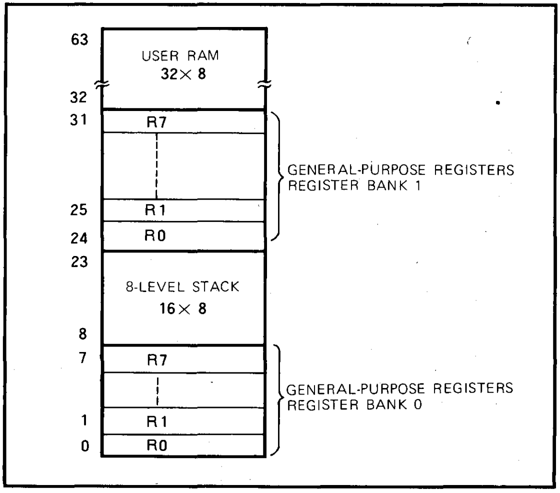


Fig. 4 Data memory (RAM)

PROGRAM COUNTER (PC) AND STACK (SK)

The MELPS 8-48 program counter is composed of a 12-bit binary counter as shown in Fig. 5. 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. 6.

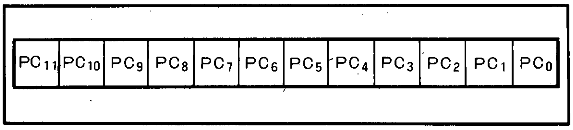


Fig. 5 Program counter

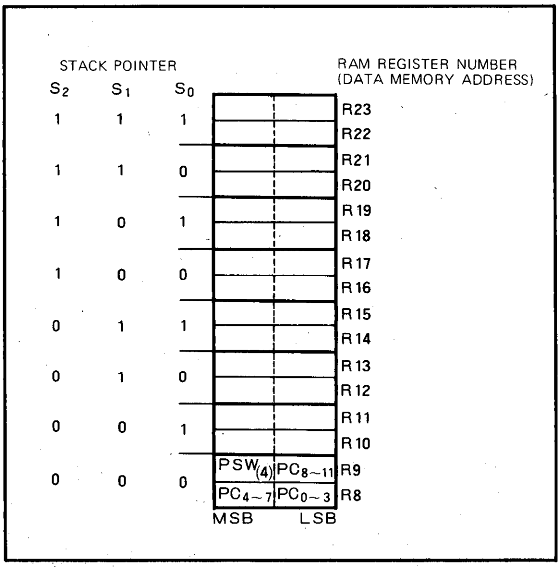


Fig. 6 Relation between the program counter stack and the stack pointer

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. 7. 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
0 = Bank 0
1 = Bank 1
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)

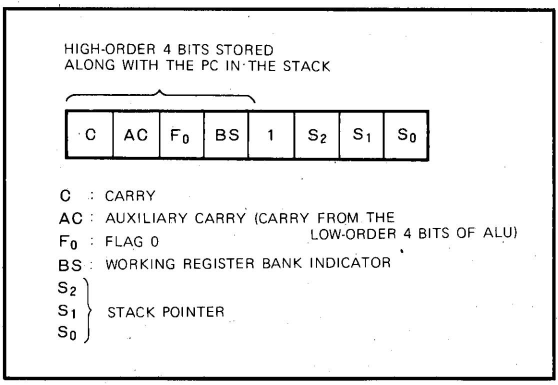

HIGH-ORDER 4 BITS STORED ALONG WITH THE PC IN THE STACK
C
AC
F0
BS
1
S2
S1
S0
C : CARRY
AC : AUXILIARY CARRY (CARRY FROM THE LOW-ORDER 4 BITS OF ALU)
F0 : FLAG 0
BS : WORKING REGISTER BANK INDICATOR
S2, S1, S0 STACK POINTER

Fig. 7 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. 8. All terminals of ports 1 and 2 can be used for input or output.

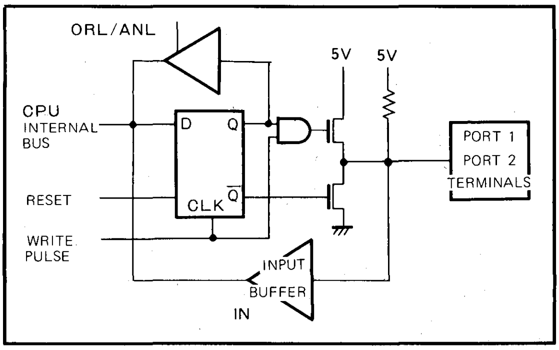


Fig. 8 I/O ports 1 and 2 circuit

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 $\overline{\text{INT}}$ terminal. A "Wired-OR" connection can be used for checking multiple interrupts.

The $\overline{\text{INT}}$ 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 $\overline{\text{INT}}$ 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 $\overline{\text{INT}}$ 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 $\overline{\text{INT}}$" section.

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).

FUNCTION OF MELPS 8-48 MICROCOMPUTERS

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 $80\mu s$. Fig. 9 shows the timer/event counter.

The counter can be initialized by executing an MOV T, A instruction. The timer can be used to measure $80\mu s$~20ms in multiples of $80\mu s$. When it is necessary to measure over 20ms (maximum count $256 x 80\mu s$) of delay time the number of overflows,one every 20ms, can be counted by the program. To measure times of less than $80\mu s$; 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.

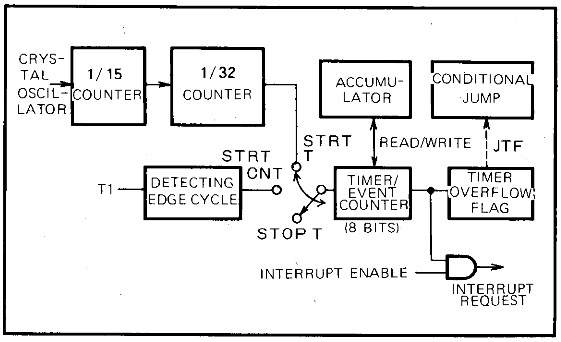


Fig. 9 Timer/event counter

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 ENTO 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. 11 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. 12.

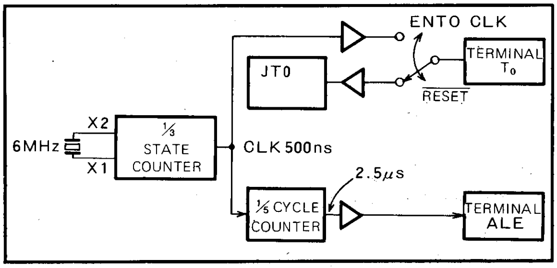


Fig.10 Clocking cycle generation

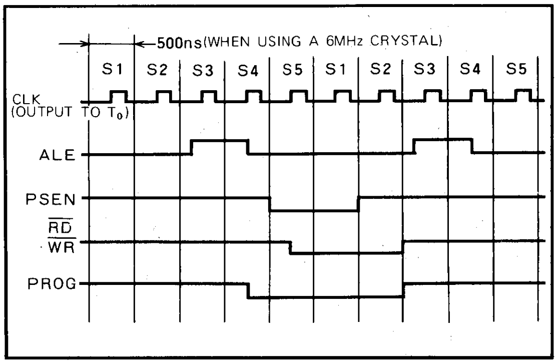


Fig.11 Clock and generated cycle signals

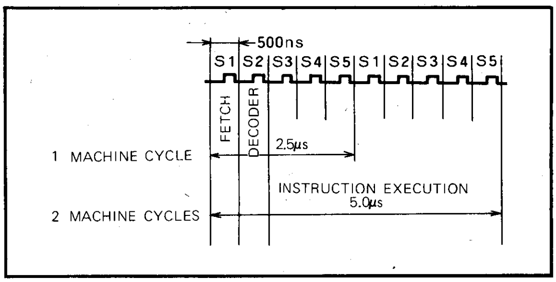


Fig.12 Instruction execution timing

$\overline{\text{RESET}}$

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. 13. An external reset pulse applied at $\overline{\text{RESET}}$ 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 $\overline{\text{RESET}}$ 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.

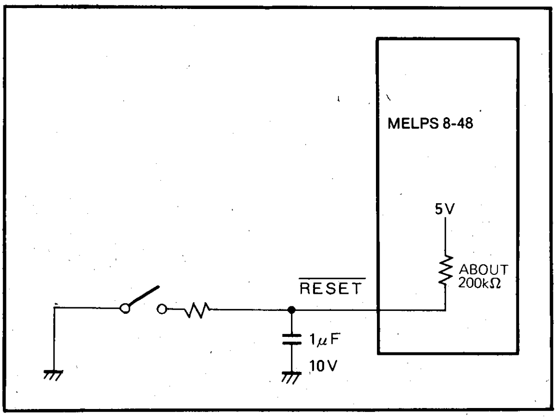
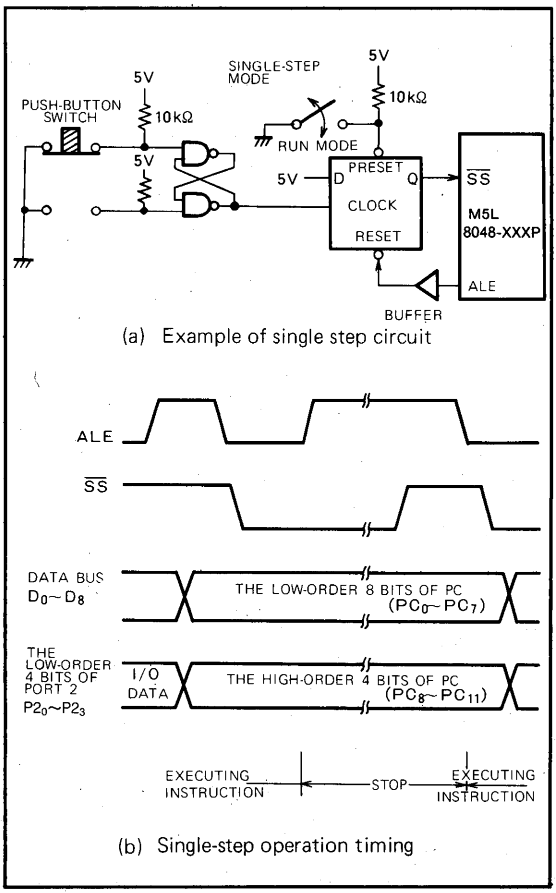


Fig. 13 Example of a reset circuit

SINGLE-STEP OPERATION

The terminal $\overline{\text{SS}}$ 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 $\overline{\text{SS}}$ and ALE as shown in Fig. 14.

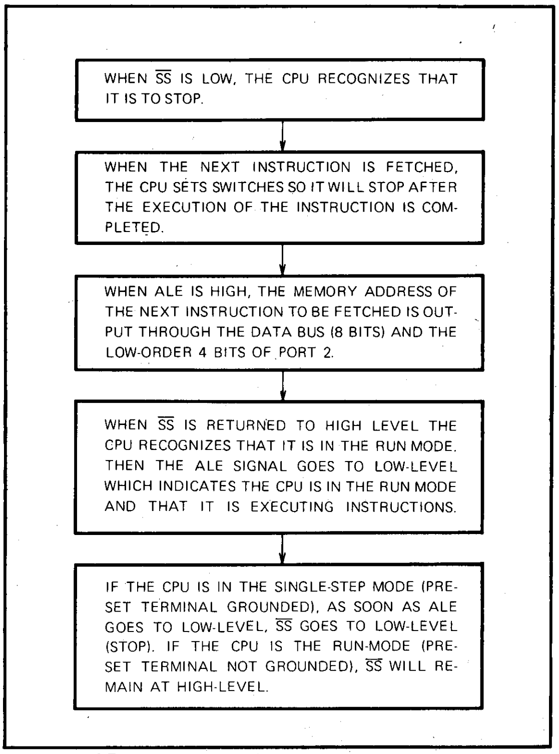


(a) Example of single step circuit

(b) Single-step operation timing

Fig. 14 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 $\overline{\text{SS}}$. When the preset terminal goes to low-level, $\overline{\text{SS}}$ goes to high-level, which puts the CPU in RUN mode. When the preset terminal is grounded it goes to high-level. Then $\overline{\text{SS}}$ goes to low-level. When $\overline{\text{SS}}$ 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 $\overline{\text{SS}}$ to high-level. When $\overline{\text{SS}}$ 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.



WHEN $\overline{SS}$ IS LOW, THE CPU RECOGNIZES THAT IT IS TO STOP.

WHEN THE NEXT INSTRUCTION IS FETCHED, THE CPU SETS SWITCHES SO IT WILL STOP AFTER THE EXECUTION OF THE INSTRUCTION IS COMPLETED.

WHEN ALE IS HIGH, THE MEMORY ADDRESS OF THE NEXT INSTRUCTION TO BE FETCHED IS OUTPUT THROUGH THE DATA BUS (8 BITS) AND THE LOW-ORDER 4 BITS OF PORT 2.

WHEN $\overline{SS}$ IS RETURNED TO HIGH LEVEL THE CPU RECOGNIZES THAT IT IS IN THE RUN MODE. THEN THE ALE SIGNAL GOES TO LOW-LEVEL WHICH INDICATES THE CPU IS IN THE RUN MODE AND THAT IT IS EXECUTING INSTRUCTIONS.

IF THE CPU IS IN THE SINGLE-STEP MODE (PRESET TERMINAL GROUNDED), AS SOON AS ALE GOES TO LOW-LEVEL, $\overline{SS}$ GOES TO LOW-LEVEL (STOP). IF THE CPU IS THE RUN-MODE (PRESET TERMINAL NOT GROUNDED), $\overline{SS}$ WILL REMAIN AT HIGH-LEVEL.

Fig. 15 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.

FUNCTION OF MELPS 8-48 MICROCOMPUTERS

MACHINE INSTRUCTIONS

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	
Transfer	MOV A, #n	0 0 1 0 $n_7n_6n_5n_4$	0 0 1 1 $n_3n_2n_1n_0$	2 3 n	2	2	(A)←n				Transfers data n to register A.
	MOV A, PSW	1 1 0 0	0 1 1 1	C 7	1	1	(A)←(PSW)				Transfers the contents of the program status word to register A.
	MOV A, Rr	1 1 1 1	1 $r_2r_1r_0$	F 8 + 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	F 0 + 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	D 7	1	1	(PSW)←(A) (C)←(A_7), (AC)←(A_6)	○	○		Transfers the contents of register A to the program status word.
	MOV Rr, A	1 0 1 0	1 $r_2r_1r_0$	A 8 + 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 $n_7n_6n_5n_4$	1 r_2 r_1 r_0 $n_3n_2n_1n_0$	B 8 + 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	A 0 + 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 $n_7n_6n_5n_4$	0 0 0 r_0 $n_3n_2n_1n_0$	B 0 + 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	A 3	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	E 3	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	1 0 0 1	0 0 0 r_0	9 0 + 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	1 0 0 0	0 0 0 r_0	8 0 + 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 $r_2r_1r_0$	2 8 + 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	2 0 + 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	3 0 + 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 $n_7n_6n_5n_4$	0 0 1 1 $n_3n_2n_1n_0$	0 3 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 $r_2r_1r_0$	6 8 + 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	6 0 + 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 $n_7n_6n_5n_4$	0 0 1 1 $n_3n_2n_1n_0$	1 3 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 $r_2r_1r_0$	7 8 + 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	7 0 + 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.

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	
Arithmetic	ANL A, #n	0 1 0 1 $n_7 n_6 n_5 n_4$	0 0 1 1 $n_3 n_2 n_1 n_0$	5 3 n	2	2	$(A) \leftarrow (A) \wedge n$				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	$(A) \leftarrow (A) \wedge (Rr)$ $r=0 \sim 7$				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	$(A) \leftarrow (A) \wedge (M(Rr))$ $r=0 \sim 1$				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 $n_7 n_6 n_5 n_4$	0 0 1 1 $n_3 n_2 n_1 n_0$	4 3 n	2	2	$(A) \leftarrow (A) \vee n$				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	$(A) \leftarrow (A) \vee (Rr)$ $r=0 \sim 7$				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	$(A) \leftarrow (A) \vee (M(Rr))$ $r=0 \sim 1$				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 $n_7 n_6 n_5 n_4$	0 0 1 1 $n_3 n_2 n_1 n_0$	D 3 n	2	2	$(A) \leftarrow (A) \forall n$				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	$(A) \leftarrow (A) \forall (Rr)$ $r=1 \sim 7$				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	$(A) \leftarrow (A) \forall (M(Rr))$ $r=0 \sim 1$				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	$(A) \leftarrow (A)+1$				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	$(A) \leftarrow (A)-1$				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	$(A) \leftarrow 0$				Clears the contents of register A, resets to 0.
	CPL A	0 0 1 1	0 1 1 1	3 7	1	1	$(A) \leftarrow (\bar{A})$				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	$(A) \leftarrow (A)$ 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	$(A_{n+1}) \leftarrow (A_n)$ $(A_0) \leftarrow (A_7) \quad n=0 \sim 6$				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	$(A_{n+1}) \leftarrow (A_n)$ $(A_0) \leftarrow (C)$ $(C) \leftarrow (A_7) \quad n=0 \sim 6$	○			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	$(A_n) \leftarrow (A_{n+1})$ $(A_7) \leftarrow (A_0) \quad n=0 \sim 6$				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	$(A_n) \leftarrow (A_{n+1})$ $(A_7) \leftarrow (C)$ $(C) \leftarrow (A_0) \quad n=0 \sim 6$	○			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	$(Rr) \leftarrow (Rr)+1$ $r=0 \sim 7$				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	$(M(Rr)) \leftarrow (M(Rr))+1$ $r=0 \sim 1$				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	$(Rr) \leftarrow (Rr)-1$ $r=0 \sim 7$				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 $m_3 m_2 m_1 m_0$	1 2 + b×2 m	2	2	$(A_b)=1$ then $(PC_0 \sim PC_7) \leftarrow m$ $(A_b)=0$ then $(PC) \leftarrow (PC)+2$ $b_7b_6b_5 = 0 \sim 7$				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.
	JTF m	0 0 0 1 $m_7 m_6 m_5 m_4$	0 1 1 0 $m_3 m_2 m_1 m_0$	1 6 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	1 0 0 0 $m_7 m_6 m_5 m_4$	0 1 1 0 $m_3 m_2 m_1 m_0$	8 6 m	2	2	$(PC_0 \sim PC_7) \leftarrow m$ when $(INT)=0$ $(PC) \leftarrow (PC)+2$ when $(INT)=1$				This instruction causes a jump to the address indicated by the second byte if the external interrupt pin $\overline{INT}$ is low.
	JMP m	$m_{10} m_9 m_8$ 0 $m_7 m_6 m_5 m_4$	0 1 0 0 $m_3 m_2 m_1 m_0$	0 4 + $(m_8 \sim m_{10})$ ×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 $m_7 m_6 m_5 m_4$	1 $r_2 r_1 r_0$ $m_3 m_2 m_1 m_0$	E 8 + r m	2	2	$(Rr) \leftarrow (Rr)-1$ $r=0 \sim 7$ $(Rr) \neq 0$ 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 $m_3 m_2 m_1 m_0$	F 6 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 $m_3 m_2 m_1 m_0$	E 6 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 $m_3 m_2 m_1 m_0$	C 6 m	2	2	$(A)=0$ then $(PC_0 \sim PC_7) \leftarrow m$ $(A) \neq 0$ 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 $m_3 m_2 m_1 m_0$	9 6 m	2	2	$(A) \neq 0$ 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 $m_3 m_2 m_1 m_0$	3 6 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 $m_3 m_2 m_1 m_0$	2 6 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 $m_3 m_2 m_1 m_0$	5 6 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 $m_3 m_2 m_1 m_0$	4 6 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 $m_3 m_2 m_1 m_0$	B 6 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 $m_3 m_2 m_1 m_0$	7 6 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	9 7	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	A 7	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	8 5	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	9 5	1	1	$(F_0) \leftarrow (\overline{F_0})$				Complements the flag F_0.
	CLR F_1	1 0 1 0	0 1 0 1	A 5	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	B 5	1	1	$(F_1) \leftarrow (\overline{F_1})$				Complements the flag F_1.

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		
Subroutine call	**CALL m**	$m_{10} m_9 m_8$ 1 $m_7 m_6 m_5 m_4$	0 1 0 0 $m_3 m_2 m_1 m_0$	1 4 + $(m_8 \sim m_{10})$ ×2 m	2	2	$((SP)) \leftarrow (PC)\ (PSW_4 \sim PSW_7)$ $(SP) \leftarrow (SP)+1$ $(PC_{0-10}) \leftarrow m$ $(PC_{11}) \leftarrow MBF$				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 $PC_0 \sim PC_{10}$ and the MBF is transferred to PC_{11}.	
	RET	1 0 0 0	0 0 1 1	8 3	1	2	$(SP) \leftarrow (SP)-1$ $(PC) \leftarrow ((SP))$				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	9 3	1	2	$(SP) \leftarrow (SP)-1$ $(PC)\ (PSW_4 \sim PSW_7) \leftarrow ((SP))$				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 control	**IN A, Pp**	0 0 0 0	1 0 $p_1 p_0$	0 8 + p	1	2	$(A) \leftarrow (Pp)$ $p=1 \sim 2$				Loads the contents of P_p to register A.	
	OUTL Pp, A	0 0 1 1	1 0 $p_1 p_0$	3 8 + p	1	2	$(Pp) \leftarrow (A)$ $p=1 \sim 2$				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 $p_1 p_0$ $n_3 n_2 n_1 n_0$	9 8 + p n	2	2	$(Pp) \leftarrow (Pp) \wedge n$ $p=1 \sim 2$				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 $p_1 p_0$ $n_3 n_2 n_1 n_0$	8 8 + p n	2	2	$(Pp) \leftarrow (Pp) \vee n$ $p=1 \sim 2$				Logical ORs the contents of P_p and data n. Outputs the result to P_p	
	INS A, BUS	0 0 0 0	1 0 0 0	0 8	1	2	$(A) \leftarrow (BUS)$				Enters the contents of data bus (port 0) to register A	
	OUTL BUS, A	0 0 0 0	0 0 1 0	0 2	1	2	$(BUS) \leftarrow (A)$				Output latches the contents of register A data to data bus (port 0)	
	ANL BUS, #n	1 0 0 1 $n_7 n_6 n_5 n_4$	1 0 0 0 $n_3 n_2 n_1 n_0$	9 8 n	2	2	$(BUS) \leftarrow (BUS) \wedge n$				Logical ANDs. the contents of data bus (port 0) and data n. Outputs the result to data bus (port 0)	
	ORL BUS, #n	1 0 0 0 $n_7 n_6 n_5 n_4$	1 0 0 0 $n_3 n_2 n_1 n_0$	8 8 n	2	2	$(BUS) \leftarrow (BUS) \vee n$				Logical ORs the contents of data bus (port 0) and data n. Outputs the result to data bus (port 0)	
	MOVD A, Pp	0 0 0 0	1 1 $p_1 p_0$	0 C + $p_1 p_0$	1	2	$(A_0 \sim A_3) \leftarrow (Pp_0 \sim Pp_3)$ $(A_4 \sim A_7) \leftarrow 0 \quad p=4 \sim 7$				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.	P_p's used for multiplying 8243 ports are $P_4 \sim P_7$. Correspondence to p_2, p_1 is shown below. $P4 \cdots p_1 p_2 = 00$ $P5 \cdots p_1 p_2 = 01$ $P6 \cdots p_1 p_2 = 10$ $P7 \cdots p_1 p_2 = 11$
	MOVD Pp, A	0 0 1 1	1 1 $p_1 p_0$	3 C + $p_1 p_0$	1	2	$(Pp_0 \sim Pp_3) \leftarrow (A_0 \sim A_3)$ $p=4 \sim 7$				Outputs the low-order 4 bits of register A to P_p.	
	ANLD Pp, A	1 0 0 1	1 1 $p_1 p_0$	9 C + $p_1 p_0$	1	2	$(Pp_0 \sim Pp_3) \leftarrow (Pp_0 \sim Pp_3) \wedge (A_0 \sim A_3)$ $p=4 \sim 7$				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 $p_1 p_0$	8 C + $p_1 p_0$	1	2	$(Pp_0 \sim Pp_3) \leftarrow (Pp_0 \sim Pp_3) \vee (A_0 \sim A_3)$ $p=4 \sim 7$				Logical ORs the 4 low-order bits of register A and the contents of P_p. P_p contains the result.	

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	
Input/Output control	EN I	0 0 0 0	0 1 0 1	0 5	1	1	(INTF) ← 1				Enables outside interrupt.
	DIS I	0 0 0 1	0 1 0 1	1 5	1	1	(INTF) ← 0				Disables outside interrupt.
	SEL RB_0	1 1 0 0	0 1 0 1	C 5	1	1	(BS) ← 0				Selects working register bank 0.
	SEL RB_1	1 1 0 1	0 1 0 1	D 5	1	1	(BS) ← 1				Selects working register bank 1.
	SEL MB_0	1 1 1 0	0 1 0 1	E 5	1	1	(MBF) ← 0				Selects memory bank 0.
	SEL MB_1	1 1 1 1	0 1 0 1	F 5	1	1	(MBF) ← 1				Selects memory bank 1.
	ENTO CLK	0 1 1 1	0 1 0 1	7 5	1	1					Enables output of clock signal from terminal T_0
Timer/event counter control	MOV A, T	0 1 0 0	0 0 1 0	4 2	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	6 2	1	1	(T) ← (A)				Transfers the contents of register A to timer/event counter.
	STRT T	0 1 0 1	0 1 0 1	5 5	1	1					Starts timer operation of timer/event counterm. Minimum count cycle is 80μs.
	STRT CNT	0 1 0 0	0 1 0 1	4 5	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	6 5	1	1					Stops operation of timer or event counter.
	EN TCNTI	0 0 1 0	0 1 0 1	2 5	1	1	(TCNTF) ← 1				Enables interrupt of timer/event counter.
	DIS TCNTI	0 0 1 1	0 1 0 1	3 5	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	0 0	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.

FUNCTION OF MELPS 8-48 MICROCOMPUTERS

Symbol	Meaning	Symbol	Meaning
A	8-bit register (accumulator)	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~A_n, 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	Rcgister 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 ofothe 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 cuntent of the memory location
(M (Rr))	The content of rhe memory location addressed by the register Rr		address by the register
(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	V	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		

Instruction Code List

	$D_7 \sim D_4$	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
$D_3 \sim D_0$	Hexa-decimal	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	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, R_2	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

DESCRIPTION

The M5L8048-XXXP and M5L8035LP are 8-bit parallel microcomputer fabricated on a single chip using high-speed N-channel silicon-gate ED-MOS technology.

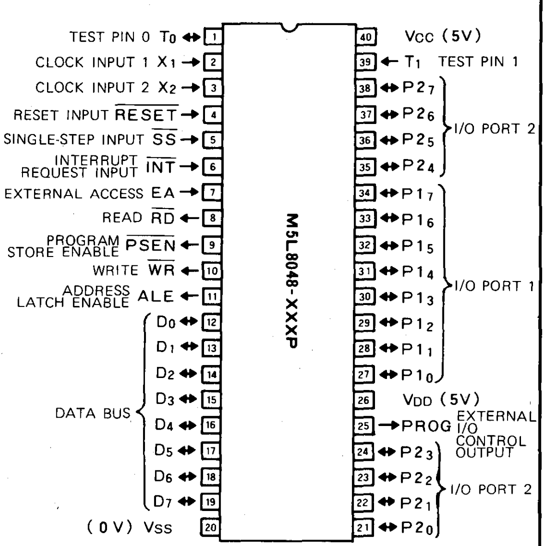
FEATURES

- 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 (for M5L8048-XXXP only)
- 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 i8048 and i8035L in pin configuration and electrical characteristics

APPLICATION

- Control processor or CPU for a wide variety of applications

PIN CONFIGURATION (TOP VIEW)

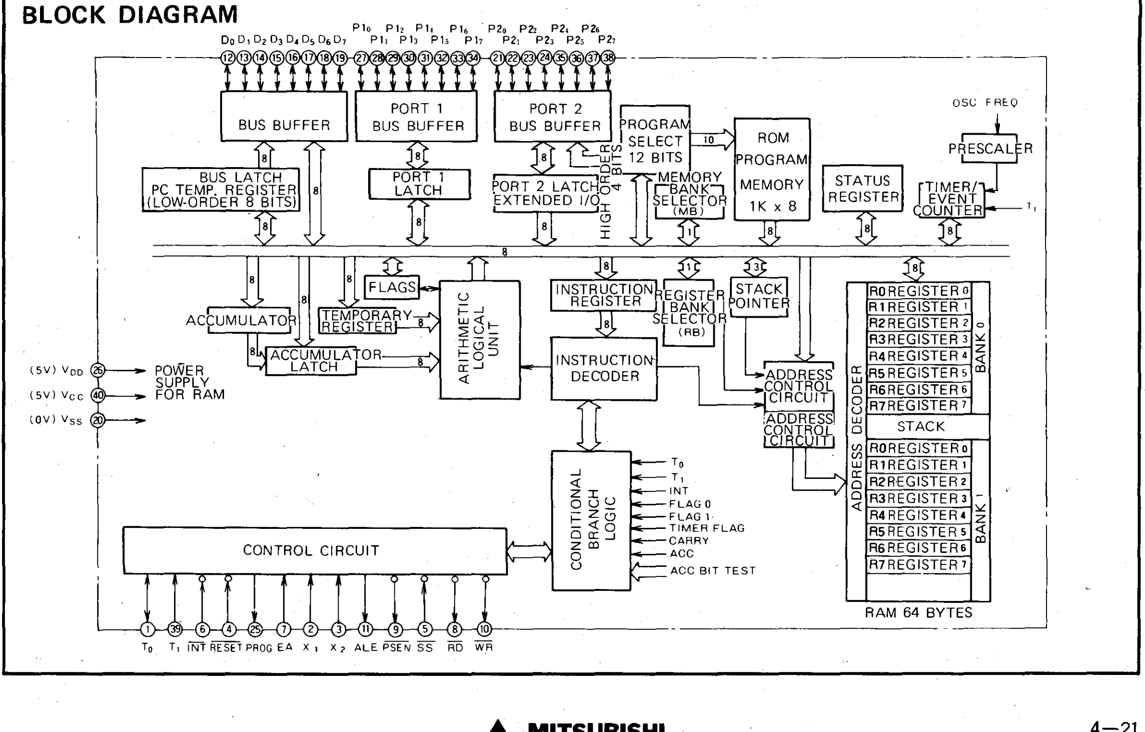


Outline 40P4

FUNCTION

The M5L8048-XXXP and M5L8035LP are integrated 8-bit CPU, with memory (ROM, RAM) and timer/event counter interrupt all contained on a single chip.

BLOCK DIAGRAM



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		①Connected to 5V power supply. ②Used for memory hold when V_{CC} is cut.
T_0	Test pin 0	Input	①Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions (JT0/JNT0).
		Output	②Used for outputting the internal clock signal (ENT0 CLK).
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.
$\overline{\text{RESET}}$	Reset	Input	Control used to initialize the CPU.
$\overline{\text{SS}}$	Single step	Input	Control signal used in conjunction with ALE to stop the CPU through each instruction, in the single-step mode.
$\overline{\text{INT}}$	Interrupt	Input	①Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions (JNI). ②Used for external interrupt to CPU.
EA	External access	Input	①Normally maintained at 0V. ②When the level is raised to 5V, external memory will be accessed even when the address is less than 400_{16} (1024). The M5L8035LP is raised to 5V.
$\overline{\text{RD}}$	Read control	Output	Read control signal used when the CPU requests data from external data memory or external device to be transferred to the data bus. (MOVX A, @R_r, and INS A, BUS)
$\overline{\text{PSEN}}$	Program store enable	Output	Strobe signal to fetch external program memory.
$\overline{\text{WR}}$	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_r, A and OUTL BUS, A)
ALE	Address latch enable	Output	A signal used for latching the address on the data bus. An ALE signal occurs once during each cycle.
$D_0 \sim D_7$	Data bus	Input/output	①Provides true bidirectional bus transfer of instructions and data between the CPU and external memory. Synchronizing is done with signals $\overline{\text{RD}}/\overline{\text{WR}}$. The output data is latched.
			②When 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 the external program memory is synchronized with $\overline{\text{PSEN}}$.
			③The output of addresses for data using the external data memory is synchronized with ALE. After that, the transfer of data with the external data memory is synchronized with $\overline{\text{RD}}/\overline{\text{WR}}$. (MOVX A, @$R_r$, and MOVX @$R_r$, A)
$P2_0 \sim P2_7$	Port 2	Input/output	①Quasi-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 needs to be output.
		Output	②$P2_0 \sim P2_3$ output 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.
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 needs to be output.
T_1	Test pin 1	Input	①Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions (JT1/JNT1). ②When enabled, event signals are transferred to the timer/event counter (STRT CNT).

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		−20～75	℃
Tstg	Storage temperature range		−65～150	℃

RECOMMENDED OPERATING CONDITIONS (Ta＝−20～75℃, 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＝−20～75℃, V_{CC}＝V_{DD}＝5V ± 10%, 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}＝−100μA	2.4			V
V_{OH1}	High-level output voltage, except the above	I_{OH}＝−50μA	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＝−20～75℃, V_{CC}＝V_{DD}＝5V ± 10%, 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.

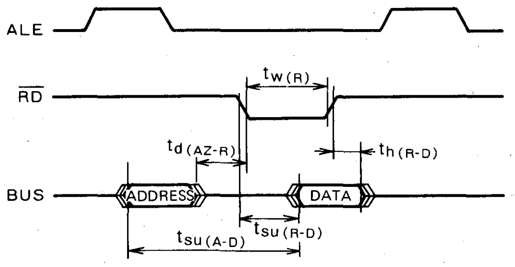
SWITCHING CHARACTERISTICS (Ta = −20~75℃, $V_{CC}=V_{DD}=5V\pm10\%$, $V_{SS}=0V$, unless otherwise noted)

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}	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

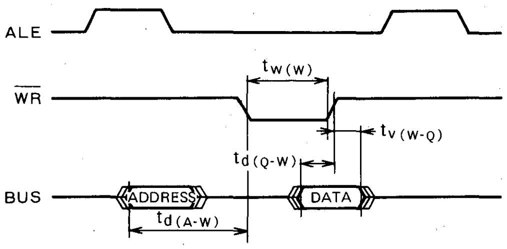
Note 2: Conditions of measurement: control output C_L=80pF
data bus output, port output C_L=150pF, t_c=2.5μs
3: Reference levels for the input/output voltages are low level=0.8V and high level=2V

TIMING DIAGRAM

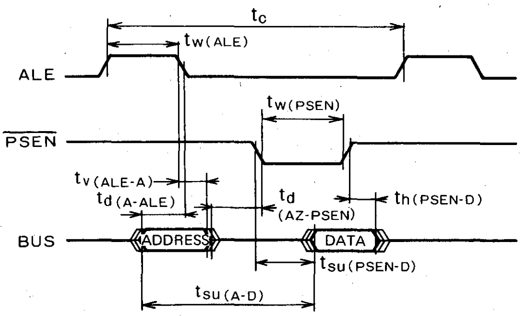
Read from External Data Memory



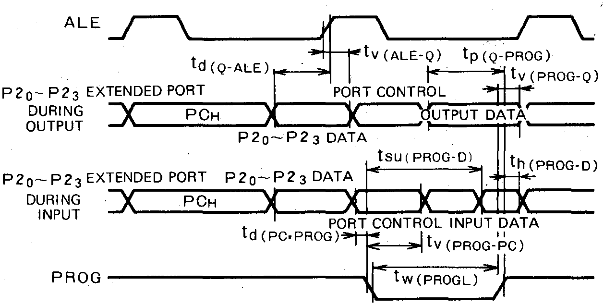
Write to External Data Memory



Instruction Fetch from External Program Memory



Port 2



M5L8049-XXXP,P-8,P-6
M5L8039P-11,P-8,P-6

SINGLE-CHIP 8-BIT MICROCOMPUTER

DESCRIPTION

The M5L8049-XXXP, P-8, P-6 and M5L8039P-11, P-8, P-6 are 8-bit parallel microcomputers fabricated on a single chip using high-speed N-channel silicon gate ED-MOS technology.

Speed \ ROM Type	Internal ROM Type	External ROM Type
11 MHz Type	M5L8049-XXXP	M5L8039P-11
8 MHz Type	M5L8049-XXXP-8	M5L8039P-8
6 MHz Type	M5L8049-XXXP-6	M5L8039P-6

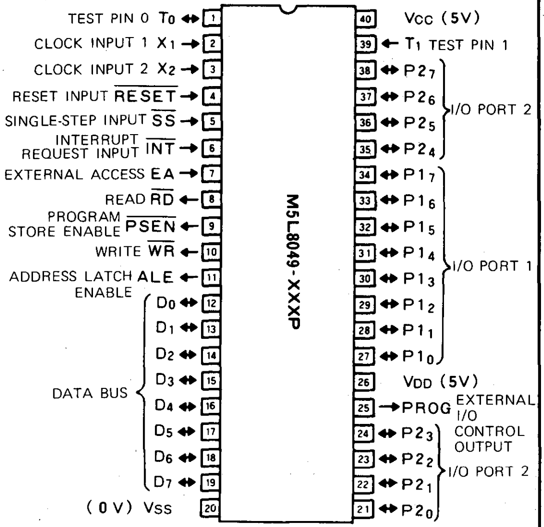
FEATURES

- Single 5V power supply
- Basic machine instructions 96
  - 1-byte instructions: 68
  - 2-byte instructions: 28
- Direct addressing .... up to 4096 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
- External RAM .... 256 bytes
- M5L8049-XXXP/M5L8039P-11, P-6 are interchangeable with i8049/i8039, i8039-6 in pin configuration and electrical characteristics.

APPLICATION

- Control processor or CPU for a wide variety of applications

PIN CONFIGURATION (TOP VIEW)

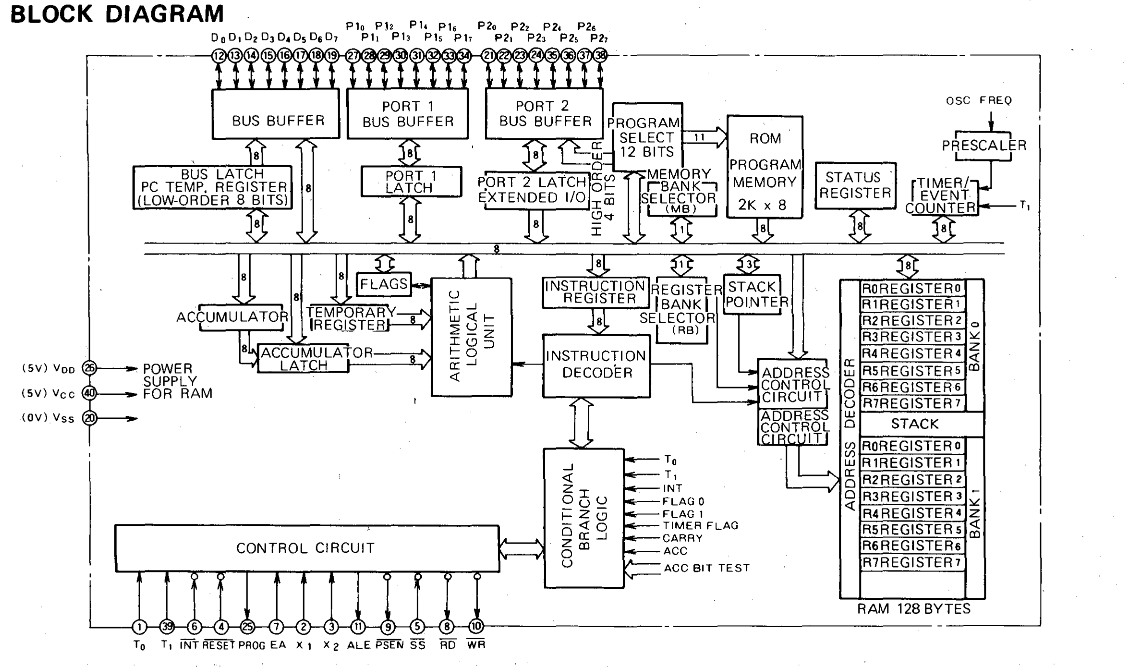


Outline 40P4

FUNCTION

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

BLOCK DIAGRAM



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		①Connected to 5V power supply. ②Used for memory hold when V_{CC} is cut.
T_0	Test pin 0	Input	①Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions (JT0/JNT0).
		Output	②Used for outputting the internal clock signal (ENT0 CLK).
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.
$\overline{\text{RESET}}$	Reset	Input	Control used to initialize the CPU.
$\overline{\text{SS}}$	Single step	Input	Control signal used in conjunction with ALE to stop the CPU through each instruction, in the single-step mode.
$\overline{\text{INT}}$	Interrupt	Input	①Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions (JNI). ②Used for external interrupt to CPU.
EA	External access	Input	①Normally maintained at 0V. ②When the level is raised to 5V, external memory will be accessed even when the address is less than 400_{16} (2048). The M5L8039P is raised to 5V.
$\overline{\text{RD}}$	Read control	Output	Read control signal used when the CPU requests data from external data memory or external device to be transferred to the data bus. (MOVX A, @R_r, and INS A, BUS)
$\overline{\text{PSEN}}$	Program store enable	Output	Strobe signal to fetch external program memory.
$\overline{\text{WR}}$	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_r, A and OUTL BUS, A)
ALE	Address latch enable	Output	A signal used for latching the address on the data bus. An ALE signal occurs once during each cycle.
D_0~D_7	Data bus	Input/output	①Provides true bidirectional bus transfer of instructions and data between the CPU and external memory. Synchronizing is done with signals $\overline{\text{RD}}/\overline{\text{WR}}$. The output data is latched.
			②When 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 the external program memory is synchronized with $\overline{\text{PSEN}}$.
			③The output of addresses for data using the external data memory is synchronized with ALE. After that, the transfer of data with the external data memory is synchronized with $\overline{\text{RD}}/\overline{\text{WR}}$. (MOVX A, @$R_r$, and MOVX @$R_r$, A)
$P2_0$~$P2_7$	Port 2	Input/output	①Quasi-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 needs to be output.
		Output	②$P2_0$~$P2_3$ output high-order 4 bits of the program counter when using external program memory.
		Input/output	③$P2_0$~$P2_3$ serve as a 4-bit I/O expander bus for the M5L8243P.
PROG	Program	Output	Strobe signal for M5L8243P I/O expander.
$P1_0$~$P1_7$	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 needs to be output.
T_1	Test pin 1	Input	①Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions (JT1/JNT1). ②When enabled, event signals are transferred to the timer/event counter (STRT CNT).

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		−20～75	℃
Tstg	Storage temperature range		−65～150	℃

RECOMMENDED OPERATING CONDITIONS (Ta=−20～75℃, 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=−20～75℃, V_{CC}=V_{DD}=5V ± 10%, 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 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}=−100μA	2.4			V
V_{OH1}	High-level output voltage, except for the above	I_{OH}=−50μA	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=−20～75℃, V_{CC}=V_{DD}=5V ± 10%. V_{SS}=0V, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Limits									Unit
			M5L8049-XXXP M5L8039P-11 (Note 2)			M5L8049-XXXP-8 M5L8039P-8			M5L8049-XXXP-6 M5L8039P-6			
			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
tc	Cycle time	t_{CY}	1.36		15.0	1.875		15.0	2.5		15.0	μs
th (PSEN-D)	Data hold time after $\overline{PSEN}$	t_{DR}	0		100	0		150	0		200	ns
th (R-D)	Data hold time after $\overline{RD}$	t_{DR}	0		100	0		150	0		200	ns
tsu (PSEN-D)	Data setup time after $\overline{PSEN}$	t_{RD}			250			350			500	ns
tsu (R-D	Data setup time after $\overline{RD}$	t_{RD}			250			350			500	ns
tsu (A-D)	Data setup time after address	t_{AD}			400			650			950	ns
tsu (PROG-D)	Data setup time after PROG	t_{PR}			650			700			810	ns
th (PROG-D)	Data hold time before PROG	t_{PF}	0		150	0		150	0		150	ns

Note 1 : The input voltage are V_{IL}=0.45V and V_{IH}=2.4V.
2 : Ta=0～70℃

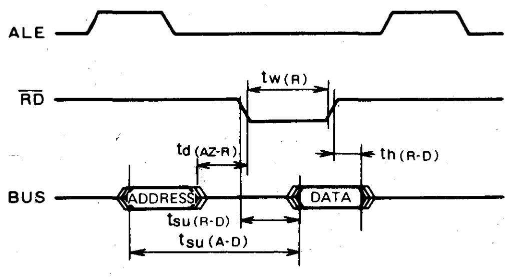
SWITCHING CHARACTERISTICS (T_a=−20~75℃, V_{CC} = V_{DD} = 5V±10%, V_{SS} = 0V, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Limits									Unit
			M5L8049-XXXP M5L8039-11 (Note 2)			M5L8049-XXXP-8 M5L8039P-8			M5L8049-XXXP-6 M5L8039P-6			
			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
$t_{w(ALE)}$	ALE pulse width	t_{LL}	150			300			400			ns
$t_{d(A-ALE)}$	Delay time, address to ALE signal	t_{AL}	70			120			150			ns
$t_{v(ALE-A)}$	Address valid time after ALE	t_{LA}	50			70			80			ns
$t_{w(PSEN)}$	$\overline{PSEN}$ pulse width	t_{CC}	300			500			700			ns
$t_{w(R)}$	$\overline{RD}$ pulse width	t_{CC}	300			500			700			ns
$t_{d(W)}$	$\overline{WR}$ pulse width	t_{CC}	300			500			700			ns
$t_{v(Q-W)}$	Delay time, data to $\overline{WR}$ signal	t_{DW}	250			380			500			ns
$t_{d(W-Q)}$	Data valid time after $\overline{WR}$	t_{WD}	40			80			120			ns
$t_{d(A-W)}$	Delay time, address to $\overline{WR}$ signal	t_{AW}	200			220			230			ns
$t_{d(AZ-R)}$	Delay time, address disable to $\overline{RD}$ signal	t_{AFC}	−10			−5			0			ns
$t_{d(AZ-PSEN)}$	Delay time, address disable to $\overline{PSEN}$ signal	t_{AFC}	−10			−5			0			ns
$t_{d(PC-PROG)}$	Delay time, port control to PROG signal	t_{CP}	100			105			110			ns
$t_{v(PROG-PC)}$	Port control valid time after PROG	t_{PC}	60			100			130			ns
$t_{p(Q-PROG)}$	Delay time, data to PROG signal	t_{DP}	200			210			220			ns
$t_{v(PROG-Q)}$	Data valid time after PROG	t_{PD}	20			45			65			ns
$t_{w(PROGL)}$	PROG low pulse width	t_{PP}	700			1150			1510			ns
$t_{d(Q-ALE)}$	Delay time, data to ALE signal	t_{PL}	150			300			400			ns
$t_{v(ALE-Q)}$	Data valid time after ALE	t_{LP}	20			100			150			ns

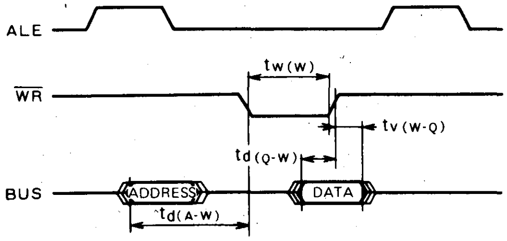
Note 3 : Conditions of measurement: control output C_L=80pF
data bus output, port output C_L=150pF
4 : Reference levels for the input/output voltages are low level=0.8V and high level=2V.

TIMING DIAGRAM

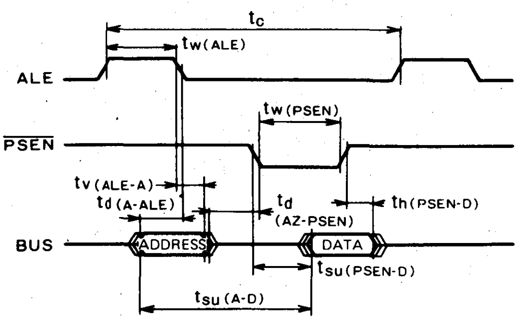
Read from External Data Memory



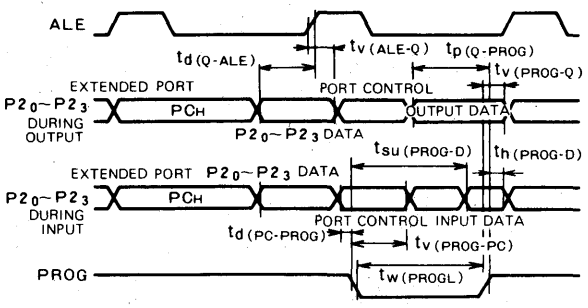
Write to External Data Memory



Instruction Fetch from External Program Memory



Port 2



M5L8049H-XXXP/M5L8039HLP

SINGLE-CHIP 8-BIT MICROCOMPUTER

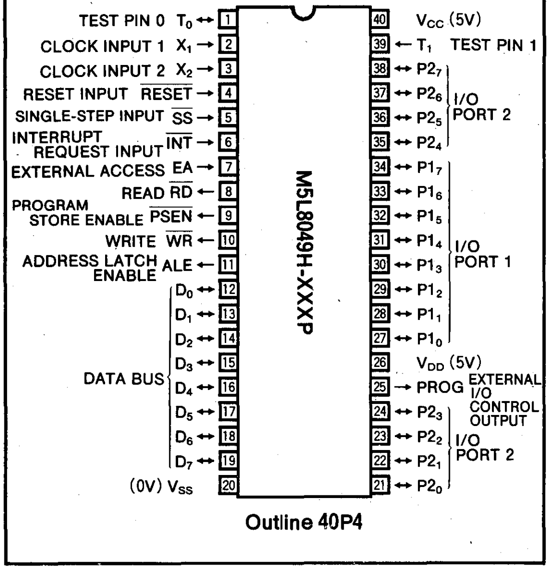
DESCRIPTION

The M5L8049H-XXXP and M5L8039HLP are 8-bit parallel microcomputers fabricated on a single chip using N-channel silicon gate ED-MOS technology.

FEATURES

- Single 5V power supply
- Low power dissipation ··························· 275mW (typ.)
- Instruction cycle ································· 1.36μs (min.)
- Basic machine instructions ·····96(1-byte instructions: 68)
- 4K-bytes memory addressing possible (direct addressing possible in 2K-bytes memory)
- Memory capacity ROM ································ 2K-bytes
  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
- M5L8049H-XXXP/M5L8039HLP are interchangeable with i8049H/i8039HL in pin configuration and electrical characteristics.

PIN CONFIGURATION (TOP VIEW)



Outline 40P4

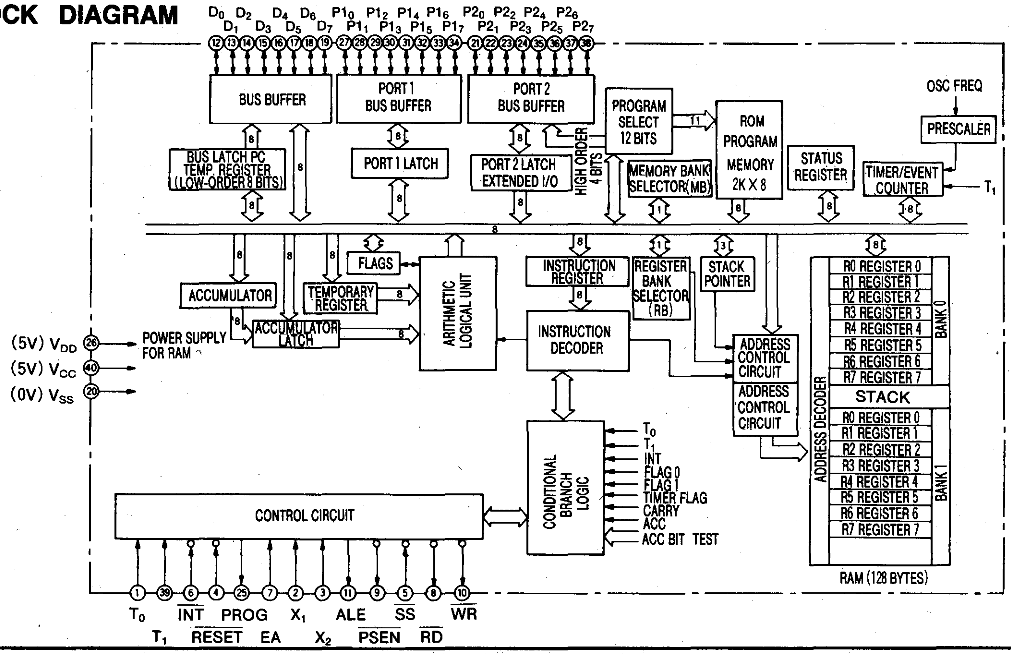
APPLICATION

Control processor or CPU for a wide variety of applications

FUNCTION

The M5L8049H-XXXP and M5L8039HLP are integrated 8-bit CPUs, with memory (ROM (Except M5L8039HLP), RAM) and timer/event counter interrupt all contained on a single chip.

BLOCK DIAGRAM



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		①Connected to 5V power supply. ②Used for memory hold when V_{CC} is cut.
T_0	Test pin 0	Input	①Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions (JT0/JNT0).
		Output	②Used for outputting the internal clock signal (ENT0 CLK).
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.
$\overline{RESET}$	Reset	Input	Control used to initialize the CPU.
$\overline{SS}$	Single step	Input	Control signal used in conjunction with ALE to stop the CPU through each instruction, in the single-step mode.
$\overline{INT}$	Interrupt	Input	①Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions (JNI). ②Used for external interrupt to CPU.
EA	External access	Input	①Normally maintained at 0V. ②When the level is raised to 5V, external memory will be accessed even when the address is less than 400_{16} (2048). The M5L8039HLP is raised to 5V.
$\overline{RD}$	Read control	Output	Read control signal used when the CPU requests data from external data memory or external device to be transferred to the data bus. (MOVX A, @R_r, and INS A, BUS)
$\overline{PSEN}$	Program store enable	Output	Strobe signal to fetch external program memory.
$\overline{WR}$	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_r, A and OUTL BUS, A)
ALE	Address latch enable	Output	A signal used for latching the address on the data bus. An ALE signal occurs once during each cycle.
D_0~D_7	Data bus	Input/output	①Provides 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.
			②When 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 the external program memory is synchronized with $\overline{PSEN}$.
			③The output of addresses for data using the 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, @R_r, and MOVX @R_r, A)
$P2_0$~$P2_7$	Port 2	Input/output	①Quasi-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 needs to be output.
		Output	②$P2_0$~$P2_3$ output high-order 4 bits of the program counter when using external program memory.
		Input/output	③$P2_0$~$P2_3$ serve as a 4-bit I/O expander bus for the M5L8243P.
PROG	Program	Output	Strobe signal for M5L8243P I/O expander.
$P1_0$~$P1_7$	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 needs to be output.
T_1	Test pin 1	Input	①Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions (JT1/JNT1). ②When enabled, event signals are transferred to the timer/event counter (STRT CNT).

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
P_d	Power dissipation	$T_a = 25$℃	1.5	W
T_{opr}	Operating free-air temperature range		0~70	℃
T_{stg}	Storage temperature range		−65~150	℃

RECOMMENDED OPERATING CONDITIONS ($T_a = 0$~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 X_1, X_2 and $\overline{\text{RESET}}$	2		V_{CC}	V
V_{IH2}	High-level input voltage, X_1, X_2 and $\overline{\text{RESET}}$	3.8		V_{CC}	V
V_{IL1}	Low-level input voltage, except X_1, X_2 and $\overline{\text{RESET}}$	−0.5		0.8	V
V_{IL2}	Low-level input voltage, X_1, X_2 and $\overline{\text{RESET}}$	−0.5		0.6	V

ELECTRICAL CHARACTERISTICS ($T_a = 0$~70℃, $V_{CC} = V_{DD} = 5V \pm 10\%$, $V_{SS} = 0V$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{OL}	Low-level output voltage (BUS)	$I_{OL} = 2mA$			0.45	V
V_{OL1}	Low-level output voltage ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, ALE)	$I_{OL} = 1.8\,mA$			0.45	V
V_{OL2}	Low-level output voltage (PROG)	$I_{OL} = 1mA$			0.45	V
V_{OL3}	Low-level output voltage (for other outputs)	$I_{OL} = 1.6mA$			0.45	V
V_{OH}	High-level output voltage (BUS)	$I_{OH} = -400\,\mu A$	2.4			V
V_{OH1}	High-level output voltage ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, ALE)	$I_{OH} = -100\,\mu A$	2.4			V
V_{OH2}	High-level output voltage (for other outputs)	$I_{OH} = -40\,\mu A$	2.4			V
I_I	Input leak current (T_1, $\overline{\text{INT}}$)	$V_{SS} \leqq V_{IN} \leqq V_{CC}$	−10		10	μA
I_{OZ}	Output leak current (BUS, T_0), high-impedance state	$V_{SS} + 0.45 \leqq V_{IN} \leqq V_{CC}$	−10		10	μA
I_{I1}	Input leak current (Port)	$V_{SS} + 0.45 \leqq V_{IN} \leqq V_{CC}$		−0.2	−0.5	mA
I_{I2}	Input leak current ($\overline{\text{RESET}}$, $\overline{\text{SS}}$)	$V_{SS} + 0.45 \leqq V_{IN} \leqq V_{CC}$		−0.05		mA
I_{DD}	Supply current from V_{DD}			5	10	mA
$I_{DD} + I_{CC}$	Supply current from V_{DD} and V_{CC}			50	100	mA

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

Symbol	Parameter	Relationship to cycle time (t_C)	Alternative symbol	Limits			Unit
				Min	Typ	Max	
t_C	Cycle time	1/ (f_{XTAL} ÷ 15)	t_{CY}	1.36		15	μs
$t_{h (PSEN-D)}$	Data hold time after PSEN	1/10 • t_C — 30	t_{DR}	0		110	ns
$t_{h (R-D)}$	Data hold time after RD	1/10 • t_C — 30	t_{DR}	0		110	ns
$t_{su (PSEN-D)}$	Data setup time after PSEN	3/10 • t_C — 200	t_{RD2}			210	ns
$t_{su (R-D)}$	Data setup time after RD	2/5 • t_C — 200	t_{RD1}			350	ns
$t_{su1 (A-D)}$	Data setup time after address (external data memory read cycle)	7/10 • t_C — 220	t_{AD1}			730	ns
$t_{su2 (A-D)}$	Data setup time after address (external program memory read cycle)	1/2 • t_C — 200	t_{AD2}			460	ns
$t_{su (PROG-D)}$	Data setup time after PROG	6/10 • t_C — 120	t_{PR}			700	ns
$t_{h (PROG-D)}$	Data hold time after PROG	1/10 • t_C	t_{PF}	0		140	ns

Note 1 : The input voltage level is V_{IL} = 0.45V and V_{IH} = 2.4V.
2 : f_{XTAL} is the oscillator frequency entered at the crystal input terminals (X_1, X_2).

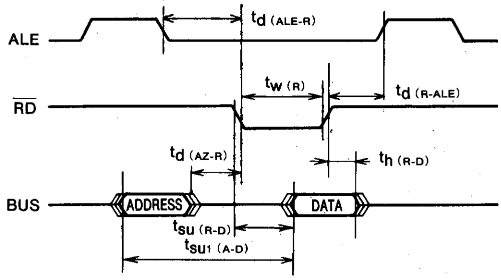
SWITCHING CHARACTERISTICS (T_a = 0~70℃, $V_{CC} = V_{DD}$ = 5V±10%, V_{SS} = 0V, unless otherwise noted)

Symbol	Parameter	Relationship to cycle time (t_C)	Alternative symbol	Limits			Unit
				Min	Typ	Max	
$t_{w (ALE)}$	ALE pulse width	7/30 • t_C — 170	t_{LL}	150			ns
$t_{d (A-ALE)}$	Address to ALE signal delay time	2/15 • t_C — 110	t_{AL}	70			ns
$t_{v (ALE-A)}$	Address valid time after ALE	1/15 • t_C — 40	t_{LA}	50			ns
$t_{w (PSEN)}$	PSEN pulse width	2/5 • t_C — 200	t_{CC2}	350			ns
$t_{w (R)}$	RD pulse width	1/2 • t_C — 200	t_{CC1}	480			ns
$t_{w (W)}$	WR pulse width	1/2 • t_C — 200	t_{CC1}	480			ns
$t_{d (Q-W)}$	Data to WR signal delay time	13/30 • t_C — 200	t_{DW}	390			ns
$t_{v (W-Q)}$	Data valid time after WR	1/15 • t_C — 50	t_{WD}	40			ns
$t_{d (A-W)}$	Address to WR signal delay time	1/3 • t_C — 150	t_{AW}	300			ns
$t_{d (AZ-R)}$	Address disable to RD signal delay time	2/15 • t_C — 40	t_{AFC1}	140			ns
$t_{d (AZ-W)}$	Address disable to WR signal delay time	2/15 • t_C — 40	t_{AFC1}	140			ns
$t_{d (AZ-PSEN)}$	Address disable to PSEN signal delay time	1/30 • t_C — 40	t_{AFC2}	10			ns
$t_{d (ALE-R)}$	ALE to RD signal delay time	1/5 • t_C — 75	t_{LAFC1}	200			ns
$t_{d (ALE-W)}$	ALE to WR signal delay time	1/5 • t_C — 75	t_{LAFC1}	200			ns
$t_{d (ALE-PSEN)}$	ALE to PSEN signal delay time	1/10 • t_C — 75	t_{LAFC2}	60			ns
$t_{d (R-ALE)}$	RD to ALE signal delay time	1/15 • t_C — 40	t_{CA1}	50			ns
$t_{d (W-ALE)}$	WR to ALE signal delay time	1/15 • t_C — 40	t_{CA1}	50			ns
$t_{d (PROG-ALE)}$	PROG to ALE signal delay time	1/15 • t_C — 40	t_{CA1}	50			ns
$t_{d (PSEN-ALE)}$	PSEN to ALE signal delay time	4/15 • t_C — 40	t_{CA2}	320			ns
$t_{d (PC-PROG)}$	Port control to PROG signal delay time	2/15 • t_C — 80	t_{CP}	100			ns
$t_{v (PROG-PC)}$	Port control valid time after PROG	4/15 • t_C — 200	t_{PC}	160			ns
$t_{d (Q-PROG)}$	Data to PROG signal delay time	2/5 • t_C — 150	t_{DP}	400			ns
$t_{v (PROG-Q)}$	Data valid time after PROG	1/10 • t_C — 50	t_{PD}	90			ns
$t_{w (PROGL)}$	PROG low-level pulse width	7/10 • t_C — 250	t_{PP}	700			ns
$t_{d (Q-ALE)}$	Data to ALE signal delay time	4/15 • t_C — 200	t_{PL}	160			ns
$t_{v (ALE-Q)}$	Data valid time after ALE	1/10 • t_C — 100	t_{LP}	40			ns
$t_{d (ALE-Q)}$	Delay time after ALE	3/10 • t_C + 100	t_{PV}			510	ns
$t_{w (T0)}$	T_0 pulse spacing	3/15 • t_C	t_{OPRR}	270			ns

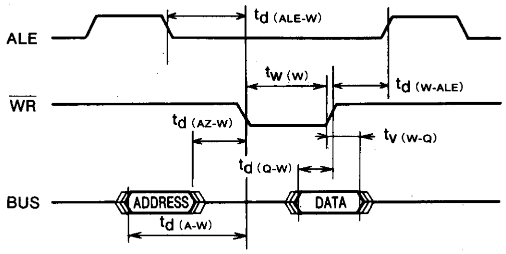
Note 3 : Conditions of measurement: control output C_L = 80pF data bus output, port output C_L = 150pF.
4 : Reference levels for input/output voltages are low-level = 0.8V high-level = 2V.

TIMING DIAGRAM

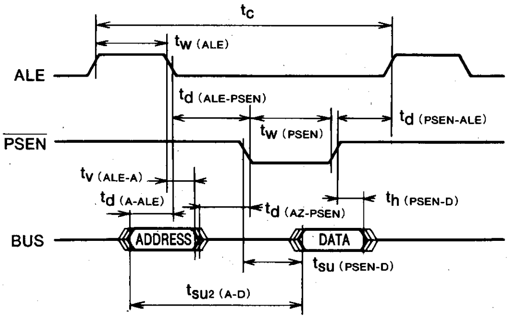
External Data Memory Read



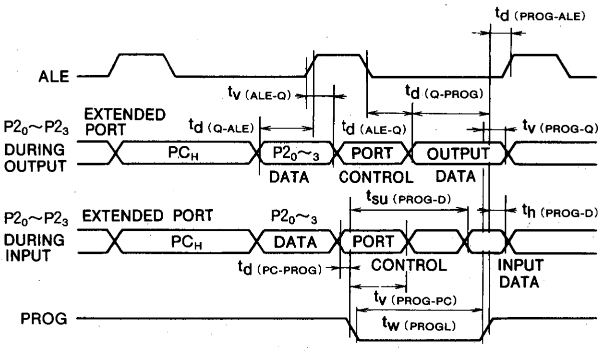
External Data Memory Write



External Program Memory Instruction Fetch



Port 2



MITSUBISHI MICROCOMPUTERS

M5M8050H-XXXP/M5M8040HP

SINGLE-CHIP 8-BIT MICROCOMPUTER

DESCRIPTION

The M5M8050H-XXXP/M5M8040HP is an 8-bit parallel microcomputer fabricated on a single chip using N-channel silicon gate ED-MOS technology.

M5M8050H-XXXP	Internal ROM Type (4K Bytes)
M5M8040HP	External ROM Type

FEATURES

- Single 5V power supply
- Instruction cycle .... 1.36μs (min)
- Basic machine instructions .. 96 (1-byte instructions: 68)
- 4K-bytes memory addressing possible (direct addressing possible in 2K bytes memory)
- Memory capacity: ROM .... 4K bytes
  RAM .... 256 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

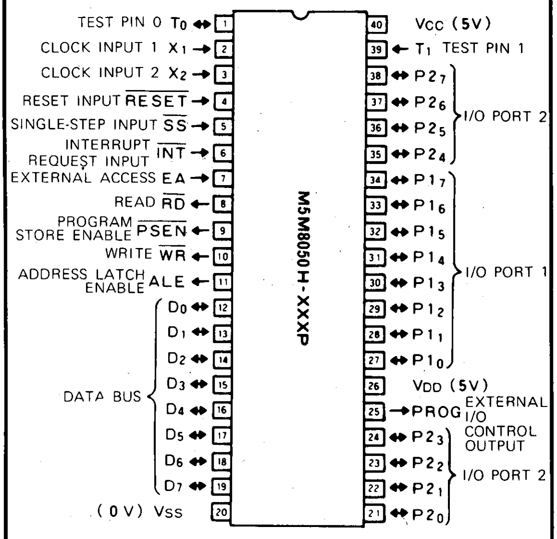
APPLICATION

Control processor or CPU for a wide variety of applications

FUNCTION

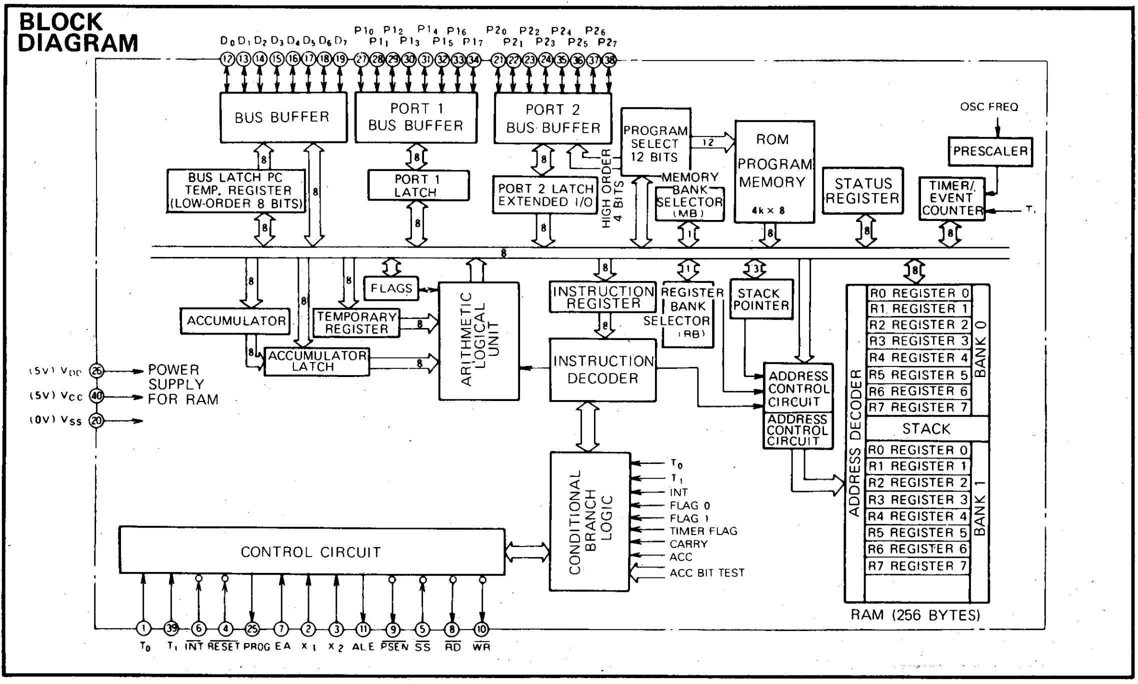
The M5M8050H-XXXP/M5M8040HP is an 8-bit CPU, with memory (ROM, RAM) and timer/event counter interrupt all contained a single chip.

PIN CONFIGURATION (TOP VIEW)



Outline 40P4

BLOCK DIAGRAM



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		①Connected to 5V power supply. ②Used for memory hold when V_{CC} is cut.
T_0	Test pin 0	Input	①Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions (JT0/JNT0).
		Output	②Used for outputting the internal clock signal (ENT0 CLK).
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.
$\overline{\text{RESET}}$	Reset	Input	Control used to initialize the CPU.
$\overline{\text{SS}}$	Single step	Input	Control signal used in conjunction with ALE to stop the CPU through each instruction, in the single-step mode.
$\overline{\text{INT}}$	Interrupt	Input	①Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions (JNI). ②Used for external interrupt to CPU.
EA	External access	Input	①Normally maintained at 0V. ②When the level is raised to 5V, external memory will be accessed. The M5M8040HP is raised to 5V.
$\overline{\text{RD}}$	Read control	Output	Read control signal used when the CPU requests data from external data memory or external device to be transferred to the data bus. (MOVX A, @R_r, and INS A, BUS)
$\overline{\text{PSEN}}$	Program store enable	Output	Strobe signal to fetch external program memory.
$\overline{\text{WR}}$	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_r, A and OUTL BUS, A)
ALE	Address latch enable	Output	A signal used for latching the address on the data bus. An ALE signal occurs once during each cycle.
D_0~D_7	Data bus	Input/output	①Provides true bidirectional bus transfer of instructions and data between the CPU and exterqal memory. Synchronizing is done with signals $\overline{\text{RD}}$/$\overline{\text{WR}}$. The output data is latched.
			②When 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 the external program memory is synchronized with $\overline{\text{PSEN}}$.
			③The output of addresses for data using the external data memory is synchronized with ALE. After that, the transfer of data with the external data memory is synchronized with $\overline{\text{RD}}$/$\overline{\text{WR}}$. (MOVX A, @R_r, and MOVX @R_r, A)
$P2_0$~$P2_7$	Port 2	Input/output	①Quasi-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 needs to be output.
		Output	②$P2_0$~$P2_3$ output high-order 4 bits of the program counter when using external program memory.
		Input/output	③$P2_0$~$P2_3$ serve as a 4-bit I/O expander bus for the M5L8243P.
PROG	Program	Output	Strobe signal for M5L8243P I/O expander.
$P1_0$~$P1_7$	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 needs to be output.
T_1	Test pin 1	Input	①Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions (JT1/JNT1). ②When enabled, event signals are transferred to the timer/event counter (STRT CNT).

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	Outpu t voltage		−0.5～7	V
P_d	Power dissipation	T_a=25℃	1.5	W
T_{opr}	Operating free-air temperature range		0～70	℃
T_{stg}	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 X_1, X_2 and $\overline{RESET}$	2		V_{CC}	V
V_{IH2}	High-level input voltage, X_1, X_2 and $\overline{RESET}$	3.8		V_{CC}	V
V_{IL1}	Low-level input voltage, except X_1, X_2 and $\overline{RESET}$	−0.5		0.8	V
V_{IL2}	Low-level input voltage, X_1, X_2 and $\overline{RESET}$	−0.5		0.6	V

ELECTRICAL CHARACTERISTICS (Ta=0～70℃, VCC=VDD=5V ± 10%, VSS=0V, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{OL}	Low-level output voltage (BUS)	I_{OL}=2mA			0.45	V
V_{OL1}	Low-level output voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	I_{OL}=1.8mA			0.45	V
V_{OL2}	Low-level output voltage (PROG)	I_{OL}=1mA			0.45	V
V_{OL3}	Low-level output voltage (for other outputs)	I_{OL}=1.6mA			0.45	V
V_{OH}	High-level output voltage (BUS)	I_{OH}=−400μA	2.4			V
V_{OH1}	High-level output voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	I_{OH}=−100μA	2.4			V
V_{OH2}	High-level output voltage (for other outputs)	I_{OH}=−40μA	2.4			V
I_I	Input leak current (T_1, $\overline{INT}$)	$V_{SS} \leqq V_{IN} \leqq V_{CC}$	−10		10	μA
I_{OZ}	Output leak current (BUS, T_0) high-impedance state	$V_{SS}+0.45 \leqq V_{IN} \leqq V_{CC}$	−10		10	μA
I_{I1}	Input leak current (PORT)	$V_{SS}+0.45 \leqq V_{IN} \leqq V_{CC}$		− 0.2	−0.5	mA
I_{I2}	Input leak current ($\overline{RESET}$, $\overline{SS}$)	$V_{SS}+0.45 \leqq V_{IN} \leqq V_{CC}$		−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}			70	140	mA

TIMING REQUIREMENTS (Ta=0~70℃, $V_{CC}=V_{DD}$=5V ± 10%, V_{SS}=0 V, unless otherwise noted)

Symbol	Parameter	Relationship to cycle time (t_C)	Alternative symbol	Limits			Unit
				Min	Typ	Max	
t_C	Cycle time	1/(f_{XTAL} ÷ 15)	t_{CY}	1.36		15	μs
$t_{h(PSEN-D)}$	Data hold time after $\overline{PSEN}$	1/10·t_C−30	t_{DR}	0		110	ns
$t_{h(R-D)}$	Data hold time after $\overline{RD}$	1/10·t_C−30	t_{DR}	0		110	ns
$t_{su(PSEN-D)}$	Data setup time after $\overline{PSEN}$	3/10·t_C−200	t_{RD2}			210	ns
$t_{su(R-D)}$	Data setup time after $\overline{RD}$	2/5·t_C−200	t_{RD1}			350	ns
$t_{su1(A-D)}$	Data setup time after address (external data memory read cycle)	7/10·t_C—220	t_{AD1}				ns
$t_{su2(A-D)}$	Data setup time after address (external program memory read cycle)	1/2·t_C—200	t_{AD2}			460	ns
$t_{su(PROG-D)}$	Data setup time after PROG	6/10·t_C− 120	t_{PR}			700	ns
$t_{h(PROG-D)}$	Data hold time after PROG	1/10·t_C	t_{PF}	0		140	ns

Note 1: The input voltages are V_{IL}=0.45V and V_{IH}=2.4V.
2: f_{XTAL} is the oscillator frequency entered at the crystal input terminals (X_1, X_2).

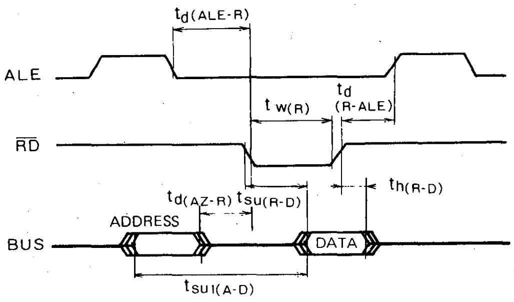
SWITCHING CHARACTERISTICS (Ta=0~70℃, $V_{CC}=V_{DD}$=5V ± 10%, V_{SS}=0 V, unless otherwise noted)

Symbol	Parameter	Relationship to cycle time (t_C)	Alternative symbol	Limits			Unit
				Min	Typ	Max	
$t_{w(ALE)}$	ALE pulse width	7/30·t_C− 170	t_{LL}	150			ns
$t_{d(A-ALE)}$	Delay time, address to ALE signal	2/15·t_C−110	t_{AL}	70			ns
$t_{v(ALE-A)}$	Address valid time after ALE	1/15·t_C−40	t_{LA}	50			ns
$t_{w(PSEN)}$	$\overline{PSEN}$ pulse width	2/5·t_C−200	t_{CC2}	350			ns
$t_{w(R)}$	$\overline{RD}$ pulse width	1/2·t_C−200	t_{CC1}	480			ns
$t_{w(W)}$	$\overline{WR}$ pulse width	1/2·t_C−200	t_{CC1}	480			ns
$t_{d(Q-W)}$	Delay time, data to $\overline{WR}$ signal	13/30·t_C−200	t_{DW}	390			ns
$t_{v(W-Q)}$	Data valid time after $\overline{WR}$	1/15·t_C−50	t_{WD}	40			ns
$t_{d(A-W)}$	Delay time, address to $\overline{WR}$ signal	1/3·t_C—150	t_{AW}	300			ns
$t_{d(AZ-R)}$	Delay time, address disable to $\overline{RD}$ signal	2/15·t_C−40	t_{AFC1}	140			ns
$t_{d(AZ-W)}$	Delay time, address disable to $\overline{WR}$ signal	2/15·t_C−40	t_{AFC1}	140			ns
$t_{d(AZ-PSEN)}$	Delay time, address disable to $\overline{PSEN}$ signal	1/30·t_C−40	t_{AFC2}	10			ns
$t_{d(ALE-R)}$	Delay time, ALE to $\overline{RD}$ signal	1/5·t_C−75	t_{LAFC1}	200			ns
$t_{d(ALE-W)}$	Delay time, ALE to $\overline{WR}$ signal	1/5·t_C−75	t_{LAFC1}	200			ns
$t_{d(ALE-PSEN)}$	Delay time, ALE to $\overline{PSEN}$ signal	1/10·t_C−75	t_{LAFC2}	60			ns
$t_{d(R-ALE)}$	Delay time, $\overline{RD}$ to ALE signal	1/15·t_C−40	t_{CA1}	50			ns
$t_{d(W-ALE)}$	Delay time, $\overline{WR}$ to ALE signal	1/15·t_C−40	t_{CA1}	50			ns
$t_{d(PROG-ALE)}$	Delay time, PROG to ALE signal	1/15·t_C−40	t_{CA1}	50			ns
$t_{d(PSEN-ALE)}$	Delay time, $\overline{PSEN}$ to ALE signal	4/15·t_C−40	t_{CA2}	320			ns
$t_{d(PC-PROG)}$	Delay time, Port control to PROG signal	2/15·t_C−80	t_{CP}	100			ns
$t_{v(PROG-PC)}$	Port control valid time after PROG	4/15·t_C−200	t_{PC}	160			ns
$t_{d(Q-PROG)}$	Delay time, Data to PROG signal	2/5·t_C− 150	t_{DP}	400			ns
$t_{v(PROG-Q)}$	Data valid time after PROG	1/10·t_C−50	t_{PD}	90			ns
$t_{w(PROGL)}$	PROG low pulse width	7/10·t_C−250	t_{PP}	700			ns
$t_{d(Q-ALE)}$	Delay time, Data to ALE signal	4/15·t_C−200	t_{PL}	160			ns
$t_{v(ALE-Q)}$	Data valid time after ALE	1/10·t_C− 100	t_{LP}	40			ns
$t_{d(ALE-Q)}$	Delay time, ALE to data	3/10·t_C +100	t_{PV}			510	ns
$t_{w(T_0)}$	T_0 pulse period	3/15·t_C	t_{OPRR}	270			ns

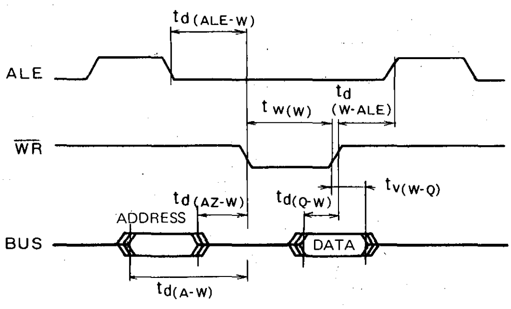
Note 3: Conditions of measurement: control output C_L=80pF data bus output, port output C_L=150pF.
4: Reference levels for input/output voltages are low-level=0.8V and high-level=2V.

TIMING DIAGRAM

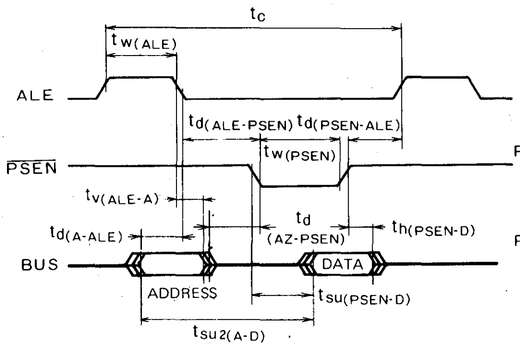
Read from External Data Memory



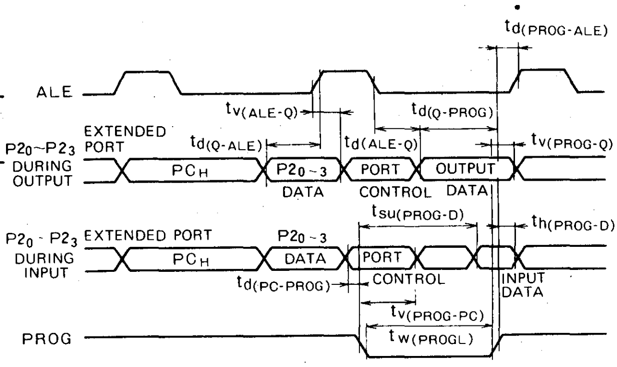
Write to External Data Memory



Instruction Fetch from External Program Memory



Port 2



MITSUBISHI MICROCOMPUTERS

M5M8050L-XXXP/M5M8040LP

SINGLE-CHIP 8-BIT MICROCOMPUTER

DESCRIPTION

The M5M8050L-XXXP and M5M8040LP are 8-bit parallel microcomputers fabricated on a single chip using N-channel silicon gate ED-MOS technology.

FEATURES

- Single 5V power supply
- Low power dissipation ································300mW (typ.)
- Instruction cycle···2.5μs (min.)
- Basic machine instructions ·····96(1-byte instructions: 68)
- 4K-bytes memory addressing possible
  (direct addressing possible in 2K bytes memory)
- Memory capacity: ROM ································· 4K bytes
  RAM·································256 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

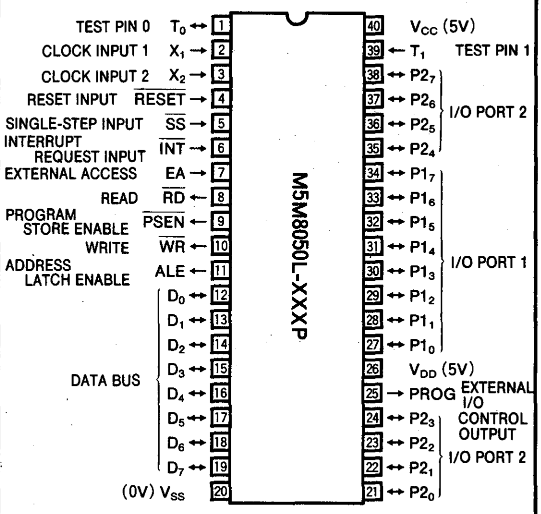
APPLICATION

Control processor or CPU for a wide variety of applications

FUNCTION

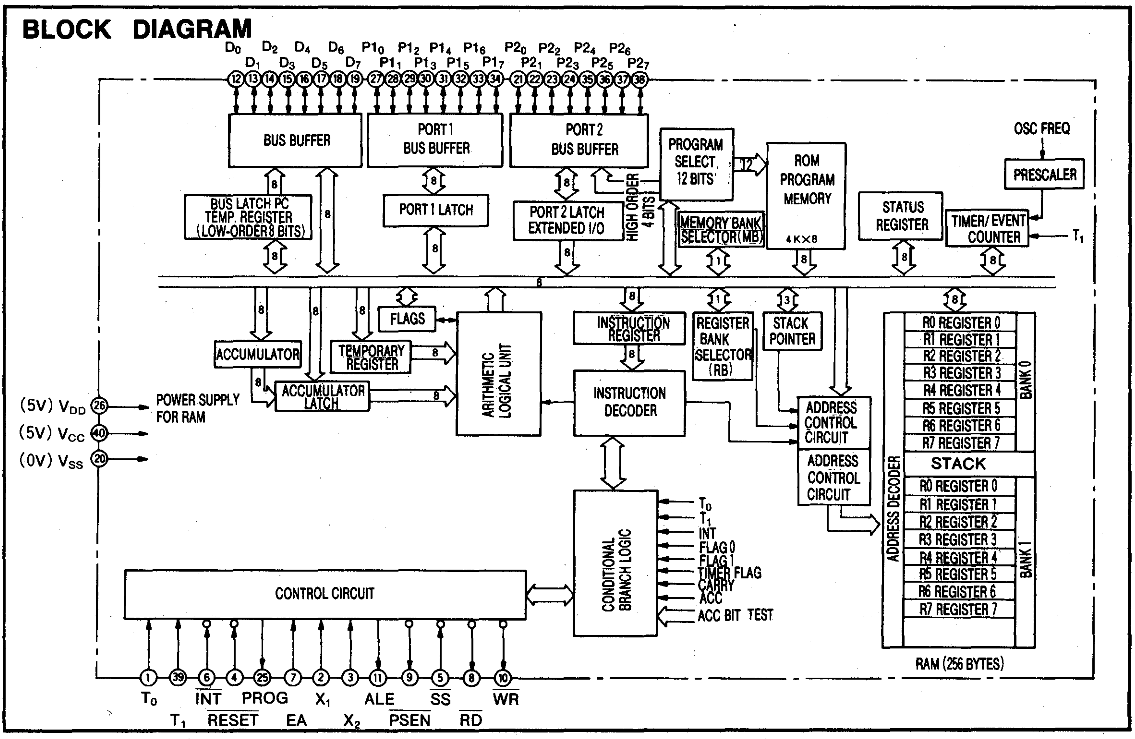
The M5M8050L-XXXP and M5M8040LP are integrated 8 bit CPUs, with memory (ROM (except M5M8040LP), RAM) and timer/event counter interrupt all contained on a single chip.

PIN CONFIGURATION (TOP VIEW)



Outline 40P4

BLOCK DIAGRAM



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		①Connected to 5V power supply. ②Used for memory hold when V_{CC} is cut.
T_0	Test pin 0	Input	①Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions (JT0/JNT0).
		Output	②Used for outputting the internal clock signal (ENT0 CLK).
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.
$\overline{\text{RESET}}$	Reset	Input	Control used to initialize the CPU.
$\overline{\text{SS}}$	Single step	Input	Control signal used in conjunction with ALE to stop the CPU through each instruction, in the single-step mode.
$\overline{\text{INT}}$	Interrupt	Input	①Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions (JNI). ②Used for external interrupt to CPU.
EA	External access	Input	①Normally maintained at 0V. ②When the level is raised to 5V, external program memory will be accessed.
$\overline{\text{RD}}$	Read control	Output	Read control signal used when the CPU requests data from external data memory or external device to be transferred to the data bus. (MOVX A, @R_r, and INS A, BUS)
$\overline{\text{PSEN}}$	Program store enable	Output	Strobe signal to fetch external program memory.
$\overline{\text{WR}}$	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_r, A and OUTL BUS, A)
ALE	Address latch enable	Output	A signal used for latching the address on the data bus. An ALE signal occurs once during each cycle.
$D_0 \sim D_7$	Data bus	Input/output	①Provides true bidirectional bus transfer of instructions and data between the CPU and external memory. Synchronizing is done with signals $\overline{\text{RD}}/\overline{\text{WR}}$. The output data is latched.
			②When 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 the external program memory is synchronized with $\overline{\text{PSEN}}$.
			③The output of addresses for data using the external data memory is synchronized with ALE. After that, the transfer of data with the external data memory is synchronized with $\overline{\text{RD}}/\overline{\text{WR}}$. (MOVX A, @$R_r$, and MOVX @$R_r$, A)
$P2_0 \sim P2_7$	Port 2	Input/output	①Quasi-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 needs to be output.
		Output	②$P2_0 \sim P2_3$ output 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.
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 needs to be output.
T_1	Test pin 1	Input	①Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions (JT1/JNT1). ②When enabled, event signals are transferred to the timer/event counter (STRT CNT).

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
P_d	Power dissipation	$T_a = 25$℃	1.5	W
T_{opr}	Operating free-air temperature range		0~70	℃
T_{stg}	Storage temperature range		−65~150	℃

RECOMMENDED OPERATING CONDITIONS ($T_a = 0$~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 X_1, X_2 and RESET	2		V_{CC}	V
V_{IH2}	High-level input voltage, X_1, X_2 and $\overline{RESET}$	3.8		V_{CC}	V
V_{IL1}	Low-level input voltage, except X_1, X_2 and $\overline{RESET}$	−0.5		0.8	V
V_{IL2}	Low-level input voltage, X_1, X_2 and $\overline{RESET}$	−0.5		0.6	V

ELECTRICAL CHARACTERISTICS ($T_a = 0$~70℃, $V_{CC} = V_{DD} = 5V \pm 10\%$, $V_{SS} = 0V$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{OL}	Low-level output voltage (BUS)	$I_{OL} = 2mA$			0.45	V
V_{OL1}	Low-level output voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	$I_{OL} = 1.8 mA$			0.45	V
V_{OL2}	Low-level output voltage (PROG)	$I_{OL} = 1mA$			0.45	V
V_{OL3}	Low-level output voltage (for other outputs)	$I_{OL} = 1.6mA$			0.45	V
V_{OH}	High-level output voltage (BUS)	$I_{OH} = -400 \mu A$	2.4			V
V_{OH1}	High-level output voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	$I_{OH} = -100 \mu A$	2.4			V
V_{OH2}	High-level output voltage (for other outputs)	$I_{OH} = -40 \mu A$	2.4			V
I_I	Input leak current (T_1, $\overline{INT}$)	$V_{SS} \leqq V_{IN} \leqq V_{CC}$	−10		10	μA
I_{OZ}	Output leak current (BUS, T_0), high-impedance state	$V_{SS} + 0.45 \leqq V_{IN} \leqq V_{CC}$	−10		10	μA
I_{I1}	Input leak current (Port)	$V_{SS} + 0.45 \leqq V_{IN} \leqq V_{CC}$		−0.2	−0.5	mA
I_{I2}	Input leak current ($\overline{RESET}$, $\overline{SS}$)	$V_{SS} + 0.45 \leqq V_{IN} \leqq V_{CC}$		−0.05		mA
I_{DD}	Supply current from V_{DD}			5	10	mA
$I_{DD} + I_{CC}$	Supply current from V_{DD} and V_{CC}				90	mA

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

Symbol	Parameter	Relationship to cycle time (t_C)	Alternative symbol	Limits			Unit
				Min	Typ	Max	
t_C	Cycle time	1/ (f_{XTAL} ÷ 15)	t_{CY}	2.5		15	μs
$t_{h\,(PSEN\text{-}D)}$	Data hold time after PSEN	1/10 · t_C — 30	t_{DR}	0		220	ns
$t_{h\,(R\text{-}D)}$	Data hold time after RD	1/10 · t_C — 30	t_{DR}	0		220	ns
$t_{su\,(PSEN\text{-}D)}$	Data setup time after PSEN	3/10 · t_C — 200	t_{RD2}			550	ns
$t_{su\,(R\text{-}D)}$	Data setup time after RD	2/5 · t_C — 200	t_{RD1}			800	ns
$t_{su1\,(A\text{-}D)}$	Data setup time after address (external data memory read cycle)	7/10 · t_C — 220	t_{AD1}			1530	ns
$t_{su2\,(A\text{-}D)}$	Data setup time after address (external program memory read cycle)	1/2 · t_C — 200	t_{AD2}			1050	ns
$t_{su\,(PROG\text{-}D)}$	Data setup time after PROG	6/10 · t_C — 120	t_{PR}			1380	ns
$t_{h\,(PROG\text{-}D)}$	Data hold time after PROG	1/10 · t_C	t_{PF}	0		250	ns

Note 1 : The input voltage level is V_{IL} = 0.45V and V_{IH} = 2.4V.
2 : f_{XTAL} is the oscillator frequency entered at the crystal input terminals (X_1, X_2).

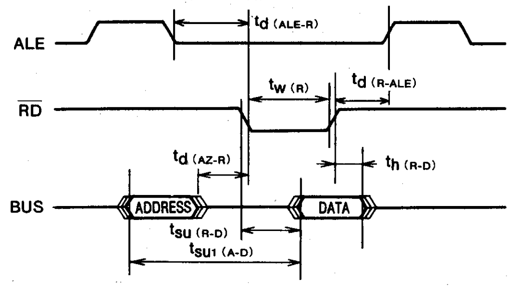
SWITCHING CHARACTERISTICS (T_a = 0~70℃, $V_{CC} = V_{DD}$ = 5V±10%, V_{SS} = 0V, unless otherwise noted)

Symbol	Parameter	Relationship to cycle time (t_C)	Alternative symbol	Limits			Unit
				Min	Typ	Max	
$t_{w\,(ALE)}$	ALE pulse width	7/30 · t_C — 170	t_{LL}	410			ns
$t_{d\,(A\text{-}ALE)}$	Address to ALE signal delay time	2/15 · t_C — 110	t_{AL}	220			ns
$t_{v\,(ALE\text{-}A)}$	Address valid time after ALE	1/15 · t_C — 40	t_{LA}	120			ns
$t_{w\,(PSEN)}$	PSEN pulse width	2/5 · t_C — 200	t_{CC2}	800			ns
$t_{w\,(R)}$	RD pulse width	1/2 · t_C — 200	t_{CC1}	1050			ns
$t_{w\,(W)}$	WR pulse width	1/2 · t_C — 200	t_{CC1}	1050			ns
$t_{d\,(Q\text{-}W)}$	Data to WR signal delay time	13/30 · t_C — 200	t_{DW}	880			ns
$t_{v\,(W\text{-}Q)}$	Data valid time after WR	1/15 · t_C — 50	t_{WD}	120			ns
$t_{d\,(A\text{-}W)}$	Address to WR signal delay time	1/3 · t_C — 150	t_{AW}	680			ns
$t_{d\,(AZ\text{-}R)}$	Address disable to RD signal delay time	2/15 · t_C — 40	t_{AFC1}	290			ns
$t_{d\,(AZ\text{-}W)}$	Address disable to WR signal delay time	2/15 · t_C — 40	t_{AFC1}	290			ns
$t_{d\,(AZ\text{-}PSEN)}$	Address disable to PSEN signal delay time	1/30 · t_C — 40	t_{AFC2}	40			ns
$t_{d\,(ALE\text{-}R)}$	ALE to RD signal delay time	1/5 · t_C — 75	t_{LAFC1}	420			ns
$t_{d\,(ALE\text{-}W)}$	ALE to WR signal delay time	1/5 · t_C — 75	t_{LAFC1}	420			ns
$t_{d\,(ALE\text{-}PSEN)}$	ALE to PSEN signal delay time	1/10 · t_C — 75	t_{LAFC2}	170			ns
$t_{d\,(R\text{-}ALE)}$	RD to ALE signal delay time	1/15 · t_C — 40	t_{CA1}	120			ns
$t_{d\,(W\text{-}ALE)}$	WR to ALE signal delay time	1/15 · t_C — 40	t_{CA1}	120			ns
$t_{d\,(PROG\text{-}ALE)}$	PROG to ALE signal delay time	1/15 · t_C — 40	t_{CA1}	120			ns
$t_{d\,(PSEN\text{-}ALE)}$	PSEN to ALE signal delay time	4/15 · t_C — 40	t_{CA2}	620			ns
$t_{d\,(PC\text{-}PROG)}$	Port control to PROG signal delay time	2/15 · t_C — 80	t_{CP}	250			ns
$t_{v\,(PROG\text{-}PC)}$	Port control valid time after PROG	4/15 · t_C — 200	t_{PC}	460			ns
$t_{d\,(Q\text{-}PROG)}$	Data to PROG signal delay time	2/5 · t_C — 150	t_{DP}	850			ns
$t_{v\,(PROG\text{-}Q)}$	Data valid time after PROG	1/10 · t_C — 50	t_{PD}	200			ns
$t_{w\,(PROGL)}$	PROG low-level pulse width	7/10 · t_C — 250	t_{PP}	1500			ns
$t_{d\,(Q\text{-}ALE)}$	Data to ALE signal delay time	4/15 · t_C — 200	t_{PL}	460			ns
$t_{v\,(ALE\text{-}Q)}$	Data valid time after ALE	1/10 · t_C — 100	t_{LP}	150			ns
$t_{d\,(ALE\text{-}Q)}$	Delay time after ALE	3/10 · t_C + 100	t_{PV}			850	ns
$t_{w\,(T_0)}$	T_0 pulse spacing	3/15 · t_C	t_{OPRR}	500			ns

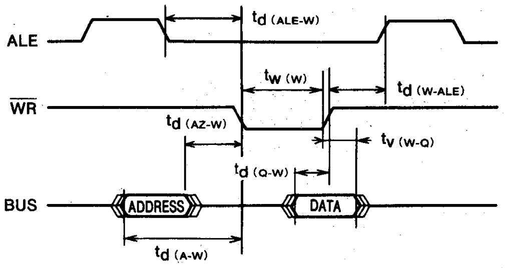
Note 3 : Conditions of measurement: control output C_L = 80pF data bus output, port output C_L = 150pF.
4 : Reference levels for input/output voltages are low-level = 0.8V high-level = 2V.

TIMING DIAGRAM

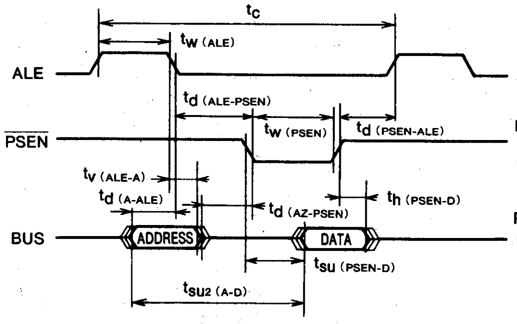
External Data Memory Read



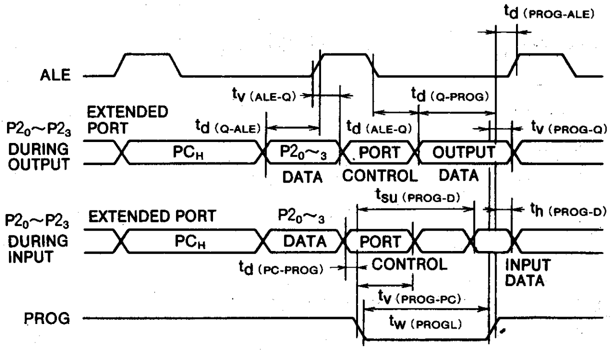
External Data Memory Write



External Program Memory Instruction Fetch



Port 2



MITSUBISHI MICROCOMPUTERS

M5M80C49-XXXP/M5M80C39P-6

SINGLE-CHIP 8-BIT CMOS MICROCOMPUTER

DESCRIPTION

The M5M80C49-XXXP and M5M80C39P-6 are 8-bit parallel microcomputer fabricated on a single chip using silicon gate CMOS technology.

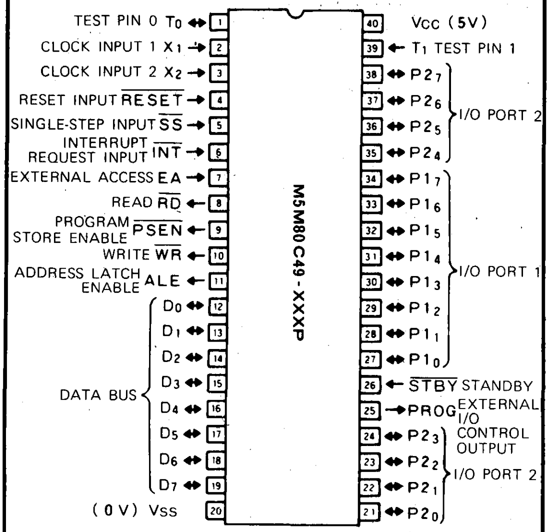
FEATURES

- Single 5V power supply
- Instruction cycle .... 2.5μs (min)
- Basic machine instructions 97
    - 1-byte instructions 69
    - 2-byte instructions 28
- Direct addressing .... up to 4096 bytes
- Internal ROM (except M5M80C39P-6) ..... 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 time/event counter interrupt, 1 level each
- High noise margin
- Low power dissipation modes (Vcc = 5V)
    - Operating 50mW
    - HALT mode 15mW
    - Stand-by 50μW

APPLICATION

Control processor for a wide variety of applications

PIN CONFIGURATION (TOP VIEW)

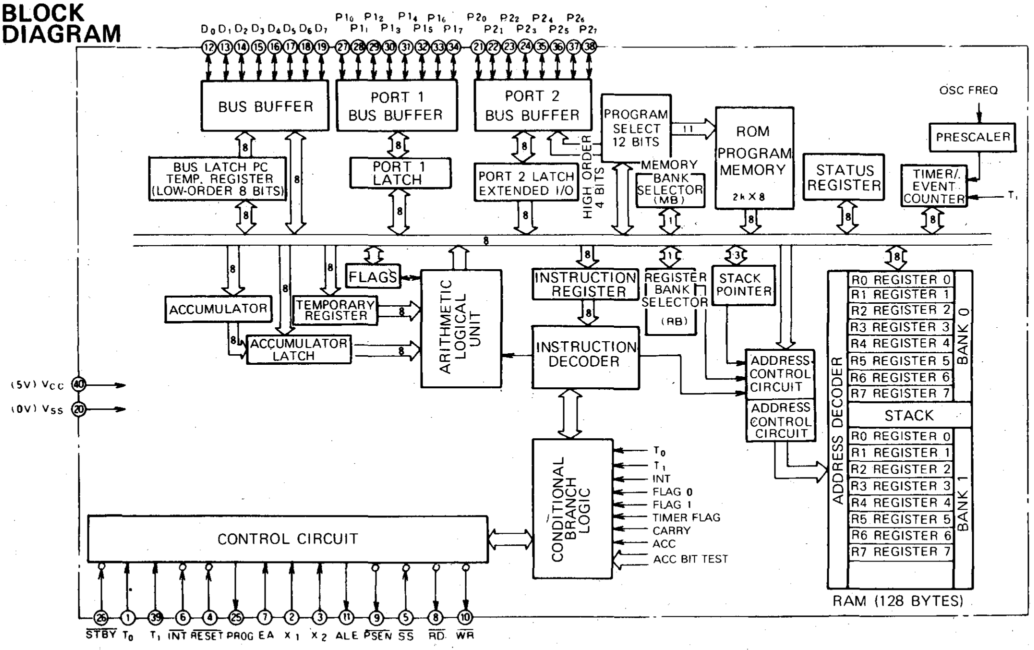


Outline 40P4

FUNCTION

The M5M80C49-XXXP and M5M80C39P-6 are integrated 8-bit CPU, with memory (ROM, RAM) and timer/event counter interrupt all contained on a single chip.

BLOCK DIAGRAM



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.
$\overline{\text{STBY}}$	Standby	Input	①Connected to 5V power supply during normal operation. ②Used when entering the standby mode. Power dissipation is reduced by connecting this to 0V.
T_0	Test pin 0	Input	①Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions (JT0/JNT0).
		Output	②Used for outputting the internal clock signal (ENT0 CLK).
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.
$\overline{\text{RESET}}$	Reset	Input	Control used to initialize the CPU.
$\overline{\text{SS}}$	Single step	Input	Control signal used in conjunction with ALE to stop the CPU through each instruction, in the single-step mode.
$\overline{\text{INT}}$	Interrupt	Input	①Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions (JNI). ②Used for external interrupt to CPU.
EA	External access	Input	①Normally maintained at 0V. ②When the level is raised to 5V, external memory will be accessed even when the address is less than 400_{16} (2048). The M5M80C39P is raised to 5V.
$\overline{\text{RD}}$	Read control	Output	Read control signal used when the CPU requests data from external data memory or external device to be transferred to the data bus. (MOVX A, @R_r, and INS A, BUS)
$\overline{\text{PSEN}}$	Program store enable	Output	Strobe signal to fetch external program memory.
$\overline{\text{WR}}$	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_r, A and OUTL BUS, A)
ALE	Address latch enable	Output	A signal used for latching the address on the data bus. An ALE signal occurs once during each cycle.
$D_0 \sim D_7$	Data bus	Input/output	①Provides true bidirectional bus transfer of instructions and data between the CPU and external memory. Synchronizing is done with signals $\overline{\text{RD}}/\overline{\text{WR}}$. The output data is latched.
			②When 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 the external program memory is synchronized with $\overline{\text{PSEN}}$.
			③The output of addresses for data using the external data memory is synchronized with ALE. After that, the transfer of data with the external data memory is synchronized with $\overline{\text{RD}}/\overline{\text{WR}}$. (MOVX A, @$R_r$, and MOVX @$R_r$, A)
$P2_0 \sim P2_7$	Port 2	Input/output	①Quasi-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 needs to be output.
		Output	②$P2_0 \sim P2_3$ output 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 M5M82C43P.
PROG	Program	Output	Strobe signal for M5M82C43P 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 needs to be output.
T_1	Test pin 1	Input	①Control signal from an external source for conditional jumping in a program. Jumping is dependent on external conditions (JT1/JNT1). ②When enabled, event signals are transferred to the timer/event counter (STRT CNT).

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Condtions	Limits	Unit
V_{CC}	Supply voltage		$V_{SS}-0.3\sim7$	V
V_I	Input voltage		$V_{SS}-0.3\sim V_{CC}+0.3$	V
V_O	Output voltage		$V_{SS}-0.3\sim V_{CC}+0.3$	V
Pd	Power dissipation	$T_a=25°C$	1.5	W
T_{opr}	Operating free-air temperature range		−40～85	℃
T_{stg}	Storage temperature range		−65～150	℃

RECOMMENDED OPERATING CONDITIONS ($T_a=-40\sim85°C$, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min	Typ	Max	
V_{CC}	Supply voltage	4.5	5	5.5	V
V_{SS}	Supply voltage		0		V
V_{IH1}	High-level input voltage, except EA, $\overline{RESET}$, X_1, X_2	$0.7\times V_{CC}$		V_{CC}	V
V_{IH2}	High-level input voltage, EA, $\overline{RESET}$, X_1, X_2	$0.8\times V_{CC}$		V_{CC}	V
V_{IL}	Low-level input voltage, except EA, $\overline{RESET}$, X_1, X_2	V_{SS}		$0.3\times V_{CC}$	V
V_{IL}	Low-level input voltage, EA, $\overline{RESET}$, X_1, X_2	V_{SS}		$0.2\times V_{CC}$	V

ELECTRICAL CHARACTERISTICS ($T_a=-40\sim85°C$, $V_{CC}=5V\pm10\%$, $V_{SS}=0V$, 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_{OH1}	High-level output voltage, except $P1_0\sim P1_7$, $P2_0\sim P2_7$	$I_{OH}=-400\mu A$	$0.75\times V_{CC}$			V
V_{OH2}	High-level output voltage, $P1_0\sim P1_7$, $P2_0\sim P2_7$	$I_{OH}=-1\mu A$	$0.75\times V_{CC}$			V
I_I	Input current, T_1, $\overline{INT}$, $\overline{SS}$, EA, $\overline{STBY}$	$V_{SS}\leqq V_{IN}\leqq V_{CC}$	−10		10	μA
I_{OZ}	Output current, BUS, T_0, high impedance state	$V_{SS}\leqq V_{IN}\leqq V_{CC}$	−10		10	μA
I_{IL1}	Input current during low level, Port	$V_{IL}=V_{SS}$		−50		μA
I_{IL2}	Input current during low level, $\overline{RESET}$	$V_{IL}=V_{SS}$		−50		μA
I_{CC}	Supply current	at 6MHz			10	mA
I_{CC}	Supply current during HALT	at 6MHz (Note 1)			3	mA
I_{CC}	Supply current during STAND BY	(Note 1)			10	μA
$V_{CC(STB)}$	Stand by power supply voltage		2			V

Note 1. BUS, T_0, T_1, EA, $\overline{SS}$, $\overline{STBY}$, $\overline{RESET}$, $\overline{INT}=V_{CC}$ or V_{SS}

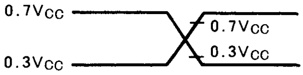
TIMING REQUIREMENT ($T_a=-40\sim85°C$, $V_{CC}=5V\pm10\%$, $V_{SS}=0V$, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Limits			Unit
			Min	Typ	Max	
t_C	Cycle time	t_{CY}	2.5		15	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 after PROG	t_{PF}	0		150	ns

SWITCHING CHARACTERISTICS ($T_a = -40 \sim 85$℃, $V_{CC} = 5V \pm 10\%$, $V_{SS} = 0V$, unless otherwise noted)

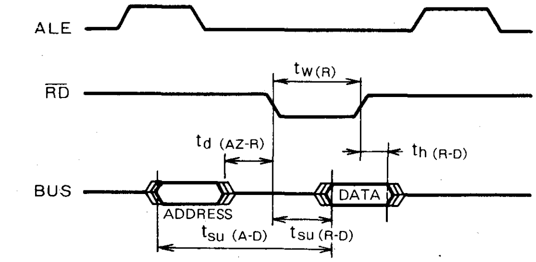
Symbol	Parameter	Alternative symbol	Limits			Unit
			Min	Typ	Max	
$t_{w(ALE)}$	ALE pulse width	t_{LL}	400			ns
$t_{d\ (A-ALE)}$	Delay time, address to ALE signal	t_{AL}	150			ns
$t_{v\ (ALE-A)}$	Address valid time after ALE	t_{LA}	80			ns
$t_{w(PSEN)}$	$\overline{PSEN}$ pulse width	t_{CC}	700			ns
$t_{w(R)}$	$\overline{RD}$ 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 floating to $\overline{RD}$ signal	t_{AFC}	0			ns
$t_{d(AZ-PSEN)}$	Delay time, address floating 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}	140			ns
$t_{d(Q-PROG)}$	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
$t_{d\ (Q-ALE)}$	Data time data to ALE	t_{PL}	400			ns
$t_{v\ (ALE-Q)}$	Data valid time after ALE	t_{LP}	100			ns

Note: Conditions of measurement: control output C_L = 80pF
data bus output, port output C_L = 150pF tc = 2.5μs

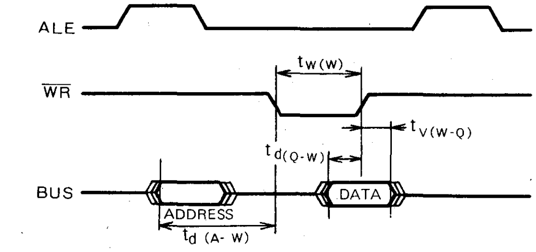


TIMING DIAGRAM

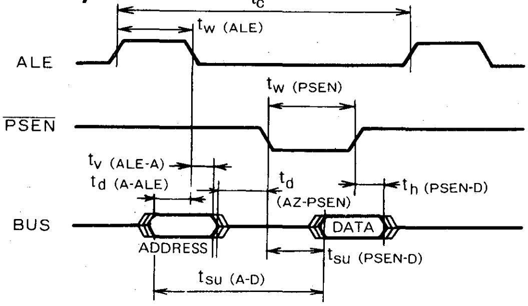
Read from External Data Memory



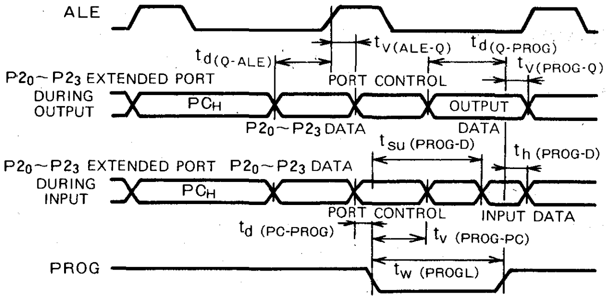
Write to External Data Memory



Instruction Fetch from External Program Memory



Port 2



Low power dissipation mode

(1) HALT mode

It will be in HALT mode when HALT instruction is executed and program execution is stopped. In HALT mode, only the basic clock operates and the others are all in the halt state. MCU keeps the contents of the registers in the state before the execution of HALT instrucitons.

The pin conditions are shown below.

Table 1

Pin	Conditions
Data bus	Output mode : Data output Input mode : High inpedence (Input mode for M5M80C39P-6)
Ports 1, 2	Port data output
ALE	L
PSEN, RD WR, PROG	H
Tϕ	Provided clock is continued
T1 (counter input)	The first pulse input is effective. After the reset of Halt mode, the count continues.
(Timer)	Halt

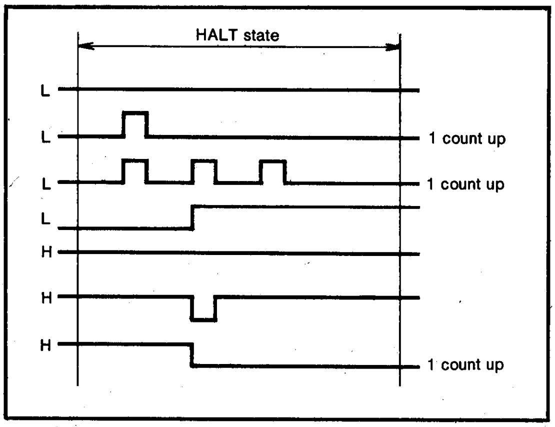


Fig. 1 Counter operation in the HALT mode

The HALT mode can be cleared by the following 2 methods

(I) By RESET input

When RESET input goes "L", an internal state is initialized as same as the normal reset operation, and the program starts from address 0.

(II) By INT input

When INT input gose "L", and if it is in the interupt enable state, the interrupt sequence will start after executing the 2en instruction following HALT instructions.

If is the interrupt disable state, the program exeution starts from the next address to HALT instructions.

(2) Standby mode

It will be in Stnadby mode when STBY input goes "L" after setting RESET input to "L". In standby mode, all operations including clock stop, and only the contents of the buit-in RAM are maintained. For the standby mode reset, let STBY pin "H", Keeping RESET input low, and then let RESET input input "H". After that the internal state is inilialized and program excuition starts from address 0.

Control method of standby mode (Example)

Place the capacitor to RESET pin, as shown in Fig 2, in order to make th standby mode control easier, so that the standby mode can be controlled by only contolling STBY pin.

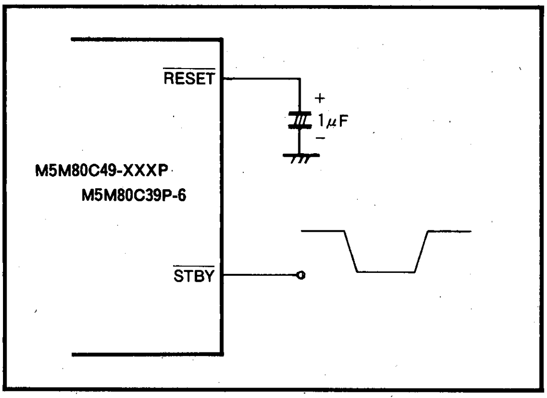


Fig. 2 Control circuit example for standby mode

MELPS 8-41 SLAVE MICROCOMPUTERS

FUNCTIONS OF MELPS 8-41 SLAVE MICROCOMPUTERS

DESCRIPTION

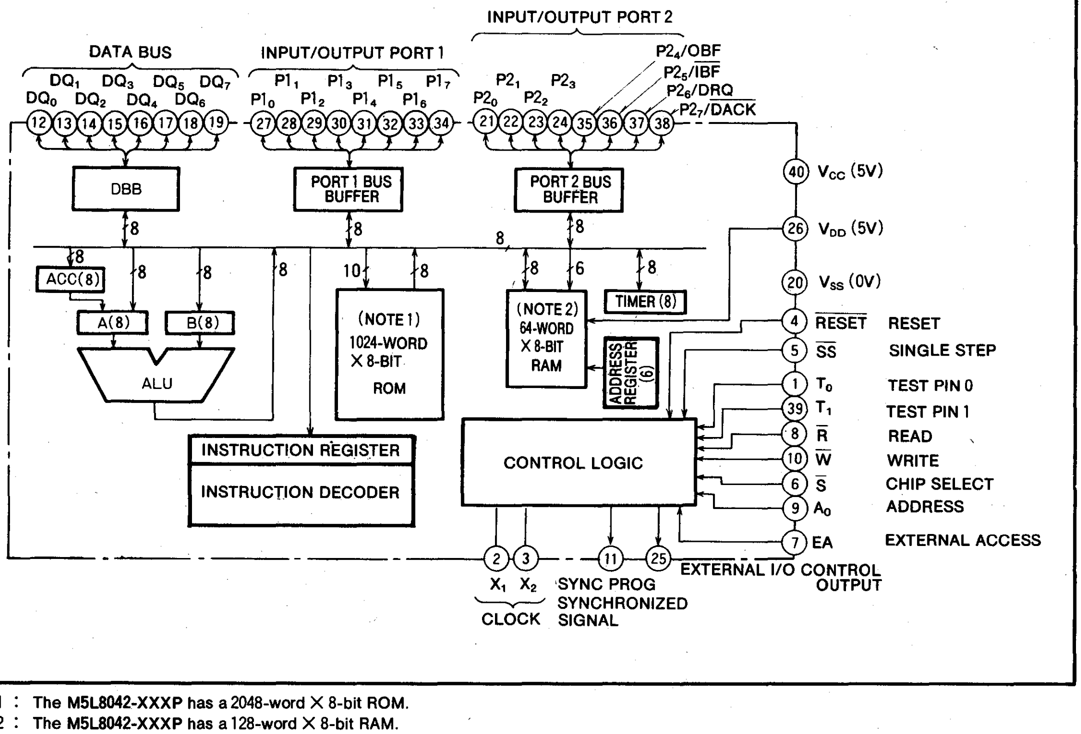
The MELPS 8-41 family is a general-purpose 8-bit CPU peripheral LSI microcomputer series configured internally with a CPU, RAM, ROM, I/O ports, timer and other functions. These microcomputers function independently of external connections but since they operate based on the instructions from the CPU (master CPU) that employs these LSIs as the peripheral LSIs, they are known as slave microcomputers.

Data is passed to and from the master CPU and slave microcomputers asynchronously through the buffer registers built into the MELPS 8-41 and therefore, when seen from the master CPU side, the LSI can be treated in every way like an ordinary peripheral LSI. Since the MELPS 8-41 has a built-in microcomputer, its functions can be changed easily simply by altering the program of the internal ROM.

MELPS 8-41 SLAVE MICROCOMPUTER FAMILY

Type name	Input clock (MHz)	Capacity			Technology used
		ROM (bytes)	RAM (bytes)	I/O (ports)	
M5L8041A-XXXP	6	1024	64	18	ED NMOS
M5L8042-XXXP	12	2048	128	18	ED NMOS

BLOCK DIAGRAM OF MELPS 8-41



Note 1 : The M5L8042-XXXP has a 2048-word X 8-bit ROM.
2 : The M5L8042-XXXP has a 128-word X 8-bit RAM.

BASIC FUNCTION BLOCKS

Program Memory (ROM)

The M5L8041A-XXXP contains a 1042-byte ROM while the M5L8042-XXXP has a built-in 2048-byte ROM. The program for the user application is stored in this ROM. Addresses 0, 3 and 7 of the ROM are reserved for special functions. Table 1 shows the meaning and functions of these 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, based on the timer/event counter, is accepted.

Data Memory (RAM)

The M5L8041A-XXXP has a built-in 64-byte (128 bytes for M5L8042-XXXP) RAM. The RAM is used for data storage and manipulation and it 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 R_0~R_7. Addresses 24~31 compose bank 1 and are also numbered R_0~R_7. 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. The remaining sections, addresses 32 and above, must be accessed indirectly using the general-purpose registers R_0 or R_1. Of course, all addresses can be indirectly accessed using the general-purpose registers R_0 and R_1.

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 (addresses 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 saves the data of the main program (addresses 0~7). The interrupt program can then freely use register bank 1 (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 8 ~ 23 comprise an 8-level program counter stack. More information on using the stack is found in the section on the program counter and stack and so reference should be made here for further details.

The general-purpose registers and program counter stack sections may be used in exactly the same way as the other RAM sections.

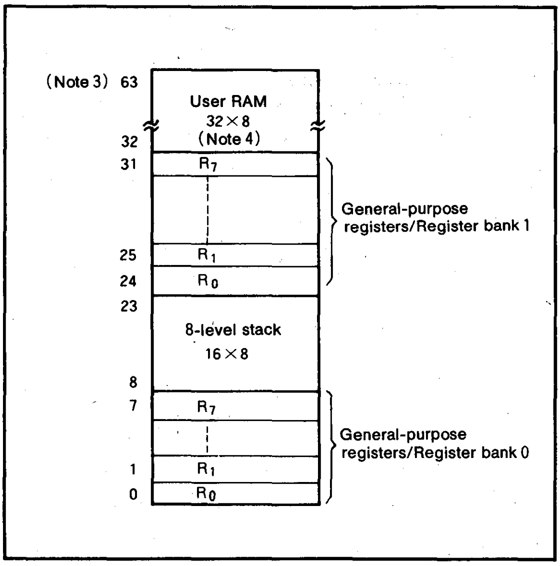


Fig.1 Data memory (RAM)

Note 3 : The corresponding address is 127 for the M5L8042-XXXP.
4 : The corresponding capacity is 96 × 8 for the M5L8042-XXXP.

FUNCTIONS OF MELPS 8-41 SLAVE MICROCOMPUTERS

Program Counter (PC) and Stack (SK)

The M5L8041A-XXXP has a 10-bit (11 bits for the M5L8042-XXXP) program counter which is illustrated in Fig. 2.

When an interrupt or a subroutine call has occurred, the program currently being executed is interrupted and the execution flow transters to the interrupt program or subroutine. When such a condition has been encountered, the value currently stored in the program counter is saved for use when restarting execution of the original program flow. The place where these program counter values are stored is the program counter stack. In addition to the program counter, the high-order 4 bits of the PSW (program status word), which will be described later, are saved in the stack. Addresses 8～23 of the RAM are used for this purpose. 10 bits (or 11 bits for the M5L8042-XXXP) for the PC and 4 bits for the PSW are saved. Therefore, a RAM capacity of 2 bytes (16 bits) is used for each time. This means that it is possible to use the program counter stack with the RAM 8～23 addresses to store both the PSW and program counter on top of each other up to 8 levels. This situation is indicated in Fig. 3.

The 3-bit stack pointer indicates at which level data is being stored in the stack. The stack pointer is also a part of the PSW but it is not stored in the program counter stack. It is automatically incremented by 1 whenever the program counter and PSW are stored in the program counter stack while, conversely, it is decremented by 1 every time stored values are taken out. The stack pointer always shows the positon of the program counter stack which is used as the next storage place. Consequently, when a return is made from a subroutine (using the RET or RETR instruction), the stack pointer is first decremented by 1 and then the contents of the program counter stack indicated by the stack pointer are transferred to the program counter.

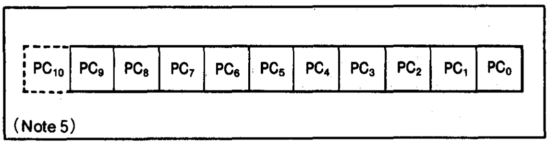


Fig.2 Program counter

Note 5 : PC_{10} for M5L8042-XXXP

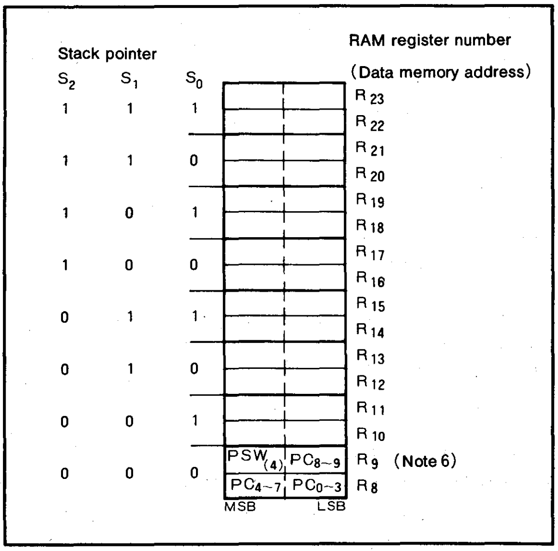


Fig.3 Relationship between program counter stack and stack pointer

Note 6 : PC_8～PC_{10} for M5L8042-XXXP

Program Status Word (PSW)

The PSW (program status word) is stored in 8 bits in the register storage. The configuration 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 ensure that the contents of the PSW are not unintentionally changed.

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

Bit 0~Bit 2 : Stack pointer (S_0, S_1, S_2)
Bit 3 : Not used
Bit 4 : Working register bank indicator
0 = Bank 0
1 = Bank 1
Bit 5 : Flag 0 (value is set by user and can be tested with JFO conditional jump instruction.)
Bit 6 : Auxiliary carry bit (AC). It is set/reset by the ADD and ADDC instructions and used by the DAA decimal compensation instruction.
Bit 7 : Carry bit (CY). This indicates an overflow after an arithmetic or logic operation.

High-order 4 bits are stored along with PC in stack.

CY	AC	F_0	BS	1	S_2	S_1	S_0

CY : Carry
AC : Auxiliary carry (carry from low-order 4 bits of ALU)
F_0 : Flag 0
BS : Working register bank indicator
S_2, S_1, S_0 } Stack pointer

Fig.4 Program status word

I/O Ports

The MELPS 8-41 has two 8-bit ports, called port 1 and port 2.

(1) Port 1 and port 2

Ports 1 and 2 are 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 are so-called quasi-bidirectional ports which have a special circuit configuration to accomplish this purpose. All the pins of the ports can be used for input or for output.

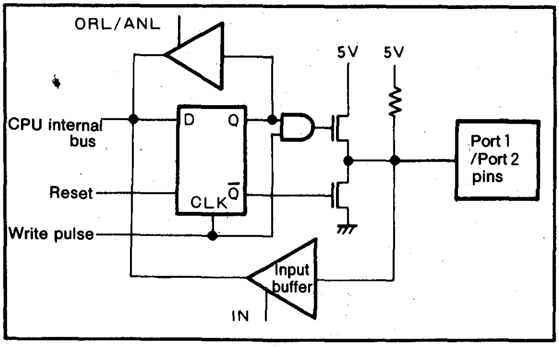


Fig.5 I/O port 1 and 2 circuit

The special circuit is shown in Fig. 5. Internal on-chip pull-up resistors are provided for all the ports for pull-up to 5V. The current required for setting the TTL signal high can be supplied through these pull-up resistors. In addition, the level can be pulled low by the standard TTL output. This means that any pin can be used for both input and output.

To shorten the switching time from a low level to high level, when 1's are output, a device with a relatively low impedance is turned on for a short time (approx. 500ns when a 6MHz crystal oscillator is used).

To use a particular port pin as an input, a logic "1" must first be written to that pin. After resetting, a port is set to an input port and remains in this state.

Therefore, it is not necessary to output all 1's if it is to be used for input. In short, a port being used for output must output 1's before it can be used for input.

FUNCTIONS OF MELPS 8-41 SLAVE MICROCOMPUTERS

The individual terminals of the quasi-bidirectional ports can be used for input or output. Some terminals, therefore, 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 data with few bits. The logical instructions ANL and ORL can easily be used to manipulate the input or output of these ports.

(2) Data bus

The data bus ($DQ_0 \sim DQ_7$) handles the data, commands and statuses between the master CPU and MELPS 8-4 1. It is controlled by the following 4 control signals. Table 2 shows the relationship between the control signals and the data bus.

A_0 : Address input indicating data/command bus buffer registers and status register
$\overline{R}$: Read input
$\overline{W}$: Write input
$\overline{S}$: Chip select input

Table 2 Control signals and data bus

$\overline{S}$	$\overline{R}$	$\overline{W}$	A_0	Data bus mode	Data on data bus
0	0	1	0	Read	Data
0	0	1	1	Read	Status
0	1	0	0	Write	Data
0	1	0	1	Write	Command ($F_1 \leftarrow 1$)
1	×	×	×	High impedance	–

The internal configuration of the data bus is shown in Fig. 7. The functions of the 3 registers indicated (status register, output data bus buffer register and input data/command bus buffer register) are now described in detail.

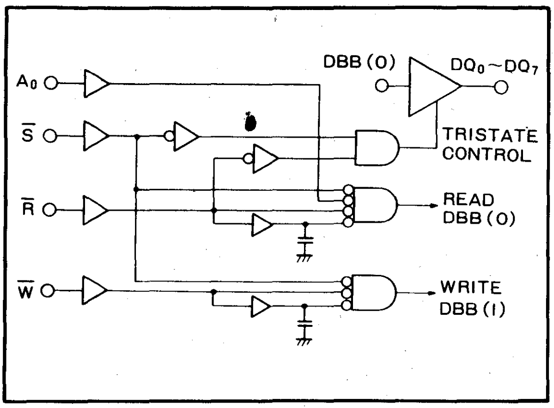


Fig.6 Internal configuration of data bus control

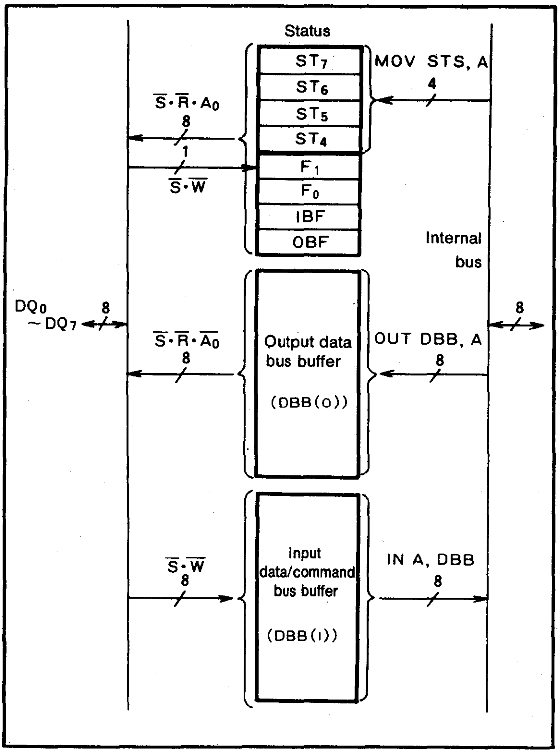

Status
ST_7
ST_6
ST_5
ST_4
F_1
F_0
IBF
OBF
MOV STS, A
4
$\overline{S} \cdot \overline{R} \cdot A_0$
8
1
$\overline{S} \cdot \overline{W}$
Internal bus
$DQ_0 \sim DQ_7$
8
$\overline{S} \cdot \overline{R} \cdot \overline{A_0}$
8
Output data bus buffer
(DBB(O))
OUT DBB, A
8
8
$\overline{S} \cdot \overline{W}$
8
Input data/command bus buffer
(DBB(I))
IN A, DBB
8

Fig.7 Internal configuration of data bus

- Status register

The status register is configured with 8 bits and the high-order 4 bits ($ST_4 \sim ST_7$) can be set as required with a software (MOV STS, A) instructions. The low-order 4 bits (OBF, IBF, F_0, F_1) are set as follows:

OBF (output buffer full)

The OBF flag is automatically set to "1" when the output instruction (OUT DBB, A) is executed inside the MELPS 8-41 and it is cleared when the contents of the output data bus buffer are read by the master CPU.

IBF (input buffer full)

The IBF flag is automatically set to "1" when the data or commands are written into the input data/command bus buffer by the master CPU and it is cleared when the input instruction (IN A, DBB) is executed inside the MELPS 8-41.

F_0 (flag 0)

The F_0 flag is set by the flag setting instructions (CPL F_0, CLR F_0) and it is used to inform the master CPU of the internal state of the MELPS 8-41.

F_1 (flag 1)

When the data or command is input into the input data/command bus buffer by the master CPU, the F_1 flag is set to the condition of the A_0 input.

The F_1 flag is also set by the flag setting instructions (CPL F_1, CLR F_1).

- Output Data Bus Buffer Register

The accumulator (A) contents are transferred to the DBB (0) output data bus buffer register by the OUT DBB, A instruction. Since the OBF flag is set at this time, the master CPU can judge whether the data has been transferred to the register by confirming the state of the OBF flag.

- Input Data/Command Bus Buffer (DBB(1)) Register

When the write request ($\overline{W}$ =0) is generated from the master CPU, the data on the data bus is transferred to the DBB (1) input data/command bus buffer register. Since the IBF flag is set at this time, it is possible to judge whether the data or command has been transferred inside the MELPS 8-41 by confirming the state of this flag.

Conditional Jumps Using Pins T_0, T_1 and Flags IBF, OBF

The conditional jump instructions are used to alter programs, depending on the internal and external conditions (states) of the CPU. Details of the jump instructions can be found in the section on machine instructions.

The input signal status of pins T_0 and T_1, and the states of the IBF and OBF flags can be checked by the conditional jump instructions. These input pins, through conditional jump instructions such as JTO and JNTO, can be used to control a program. This means that programs and processing time can be reduced by being able to test data in the input pin rather than reading the data into a accumulator and then testing it.

Pin T_1 has other functions and uses which are not related to conditional jump instructions. Details of these other functions and uses can be found on the section dealing with pin functions.

Interrupt

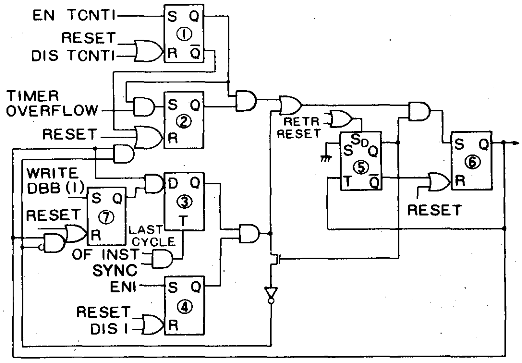
The CPU recognizes an external interrupt by a low-level signal at the $\overline{S}$ and $\overline{W}$ pins. When such an interrupt is accepted, the external interrupt pending flip-flop and IBF flag are set.

Interrupt requests are sampled between the SYNC signal outputs of every machine cycle. When a request is recognized, then as soon as the instruction being executed is terminated, a subroutine call is made to address 3 of the program memory. As with ordinary subroutine calls, the program counter and program status word (PSW) are saved in the program counter stack.

The unconditional jump instructions for enabling a jump to be made to the address where the ordinary interrupt processing program is stored are contained in address 3 of the program memory.

The interrupt level is one so that 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 cannot 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). Timer/event counter overflow which causes an interrupt request will also not be accepted.

Priority is given to the external interrupt when both an external interrupt and timer interrupt have been generated at the same time.



①TIMER INTERRUPT ENABLE FF (TCNTF)
②TIMER INTERRUPT REQUEST FF (TIRF)
③EXTERNAL INTERRUPT LATCH
④EXTERNAL INTERRUPT ENABLE FF (INTF)
⑤INTERRUPT ENABLE FF (IEF)
⑥INTERRUPT ACKNOWLEDGE FF
⑦EXTERNAL INTERRUPT PENDING FF (EIPF)

Fig.8 Interrupt control section configuration

FUNCTIONS OF MELPS 8-41 SLAVE MICROCOMPUTERS

Table 3 Acceptance of interrupts

Item	Conditions		Execution details
Internal interrupt	When TCNTEF (timer INT enable FF) = 1	No external interrupt	Interrupt is executed and call is mode to address 7.
		During external interrupt execution	Interrupt is held.
	When TCNTF (timer INT enable FF) = 0		Interrupt is not executed or held. TF (Timer flag)← 1
External interrupt	When (external INT enable FF) = 1	No timer interrupt	Interrupt is executed and call is made to address 3.
		During timer interrupt execution	Interrupt is not executed but held.
	When (external INT enable FF) = 0		
Timer and external interrupts generated simultaneously	Combination is same as conditions above		External interrupt takes priority and is executed.

When a second level of external interrupt is required, the timer interrupt, if not being used, can provide this. This is done by enabling the timer/event counter interrupt and setting the timer/event counter to FF_{16}. The CPU is placed in the event counter mode. The interrupt is then generated in address 7 by setting the T_1 input from the external source from the high to low level.

The IBF flag can be tested using a conditional jump instruction. For further details, check the section on the conditional jump instructions, pins T_0 and T_1, and the IBF and OBF flags.

Timer / Event Counter

The timer/event counter for the MELPS 8-41 is an 8-bit counter, that is used to measure time delays or count external events but not both. The same counter is used to measure time delays or count external events simply by 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 the MOV A, T instruction. Reset will stop the counting but the counter is not cleared, thus enabling counting to 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 00_{16}, the overflow flag is set and a timer interrupt request is issued. The timer flag can be checked using the JTF conditional branch instruction, and it is cleared by executing the JTF instruction or by resetting the system. When the timer interrupt is accepted, a subroutine call is made to address 7 of the 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 calling to address 3 of the program memory. The timer interrupt request is kept and will be processed when the external interrupt has been completed and a RETR instruction is executed. A latched timer interrupt request is cancelled when a timer interrupt request is generated. The STRT CNT instruction is used to change the counter to an event counter. Then the pin T_1 signal becomes the input to the event counter and events are counted up at the T_1 fall. The maximum rate that can be counted is one time in 3 machine cycles (7.5μs when using a 6MHz crystal). The high-level at T_1 must be maintained at least $\frac{1}{5}$ of the cycle time (500ns with a 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 $\frac{1}{32}$ of 400kHz (with a 6MHz crystal) or 12.5kHz (see Fig. 9). The timer is therefore counted up every 80μs. The counter can be initialized by executing an MOV T, A instruction. Delay times varying from 80μs to 20ms (256 count) can be obtained by detecting the counter overflows. Even times of more than 20ms can be achieved by counting the number of overflows using the program.

A resolution of less than 80μs can be obtained in the event counter mode by supplying an external clock to pin T_1. It is also possible to supply every third (or more) prescaled ALE signal to pin T_1 instead of an external clock.

FUNCTIONS OF MELPS 8-41 SLAVE MICROCOMPUTERS

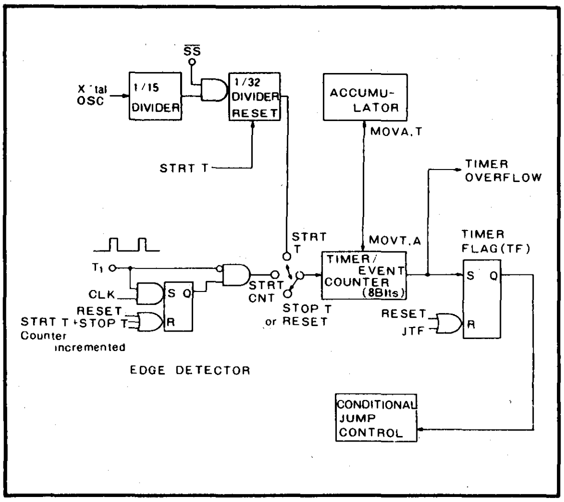


Fig.9 Timer/event counter configuration

Cycle Timing

The output of the state counter is $\frac{1}{3}$ the input frequency from the oscillator, and a CLK signal is produced which determines the times of each machine state (see Fig. 10). During the cycle count the CLK signal is prescaled by $\frac{1}{5}$ and a machine cycle containing 5 states is produced. The MELPS 8-41 instructions are executed in one or two machine cycles. Fig. 12 shows the internal operation with an instruction formed from one machine cycle.

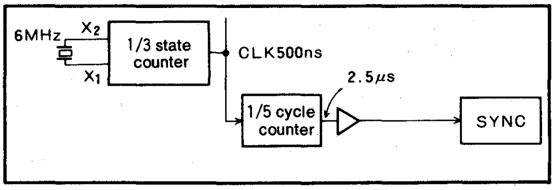


Fig.10 Clock generator circuit

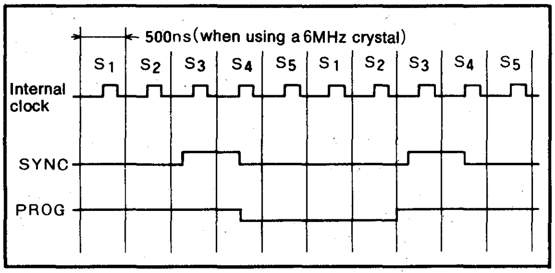


Fig.11 Clock and generated cycle signals

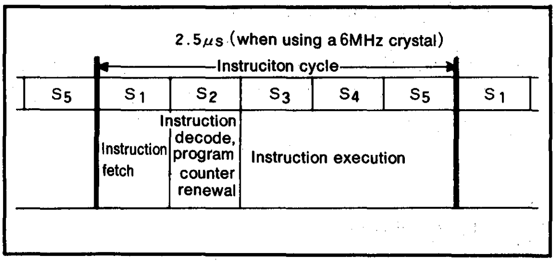

2.5μs (when using a 6MHz crystal)
Instruciton cycle
S5
S1
S2
S3
S4
S5
S1
Instruction fetch
Instruction decode, program counter renewal
Instruction execution

Fig.12 Instruction execution timing

Reset

The RESET pin is for resetting the CPU. A Schmitt trigger circuit along with a pull-up resistor are connected to it on the chip. A sufficiently long pulse can be obtained for resetting by attaching 1μF capacitor as shown in Fig. 13. An external reset pulse applied at RESET must remain at the low level for at least 10ms after the power has been turned on and after it has reached its normal level.

The reset function causes the following initialization within the CPU.

(1) The program counter is reset to 0.
(2) The stack pointer is reset to 0.
(3) The register bank 0 is selected.
(4) Ports 1 and 2 are reset to the input mode.
(5) External and timer interrupts are reset to disable state.
(6) Timer is stopped.
(7) Timer flag is cleared.
(8) Flags F_0 and F_1 are cleared.

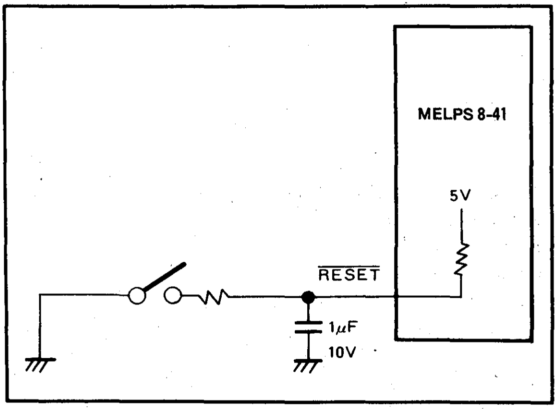
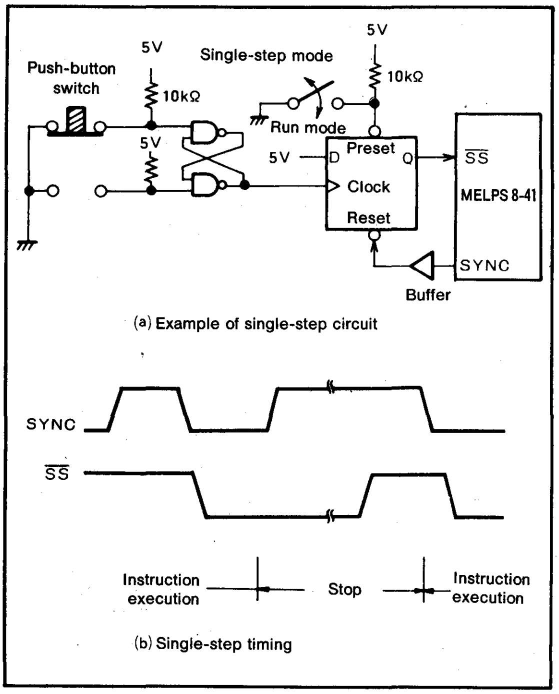


Fig.13 Example of reset circuit

Single-Step Operation

The MELPS 8-41 is provided with an $\overline{SS}$ pin for facilitating single-step operation where the CPU stops after the execution of each instruction is completed. The user can use this to trace the flow of the program, instruction by instruction, and find this to be an aid in program debugging.

$\overline{SS}$ is used in synchronization with the timing of the SYNC signal output from the CPU. Fig. 14 shows the circuit used for single-step operation and the timing involved.



(a) Example of single-step circuit

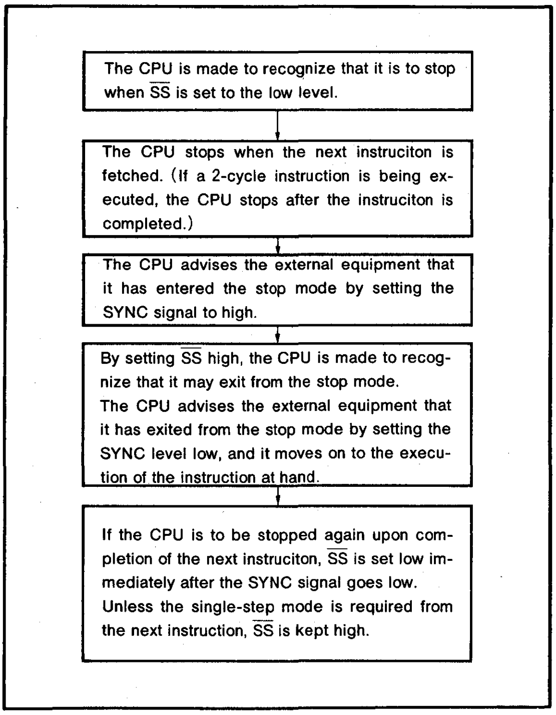
(b) Single-step timing

Fig.14 Single-step operation circuit and timing

A type D flip-flop with preset and reset pins is used to generate the signal for $\overline{SS}$.

When the preset pin is kept low, $\overline{SS}$ goes to the high level, which puts the CPU in the run mode.

For single-step operation the preset pin is switched to the high level and $\overline{SS}$ to the low level. While $\overline{SS}$ is low, the CPU stops. To restart the CPU, a pulse is supplied to the clock pin on the type D flip-flop. This sets $\overline{SS}$ to the high level, and the CPU fetches the next instruction and begins to execute it. Once the CPU starts the execution, the SYNC signal connected to the reset pin of the type D flip-flop is low and so $\overline{SS}$ also goes low. As soon as the CPU finishes executing the instruction, it is again stopped by $\overline{SS}$ going to the low level.



The CPU is made to recognize that it is to stop when $\overline{SS}$ is set to the low level.

The CPU stops when the next instruciton is fetched. (If a 2-cycle instruction is being executed, the CPU stops after the instruciton is completed.)

The CPU advises the external equipment that it has entered the stop mode by setting the SYNC signal to high.

By setting $\overline{SS}$ high, the CPU is made to recognize that it may exit from the stop mode.
The CPU advises the external equipment that it has exited from the stop mode by setting the SYNC level low, and it moves on to the execution of the instruction at hand.

If the CPU is to be stopped again upon completion of the next instruciton, $\overline{SS}$ is set low immediately after the SYNC signal goes low.
Unless the single-step mode is required from the next instruction, $\overline{SS}$ is kept high.

Fig.15 CPU operation in single-step mode

Fig.15 shows the operation of the CPU in the single-step mode.

Central Processing Unit (CPU)

The CPU is composed of an 8-bit parallel arithmetic unit, accumulator, flag flip-flop and instruction decoder. The 8-bit parallel arithmetic unit has circuit to perform the four basic arithmetic operations (addition, subtraction, multiplication and division) as well as logical operations such as AND and OR. The carry, zero and other states generated by these operations are set in the flag flip-flop. The accumulator supplies the operands (HIENZANSUU) to the arithmetic circuit, receives the results from the same circuit and keeps them. The flag flip-flop keeps the carry, zero and other states when various kinds of arithmetic operation instructions are executed.

DMA Control

Ports $P2_6$ and $P2_7$ of the MELPS 8-41 can be used not only as ordinary input/output ports but also for the control signal employed for DMA handshaking. Immediately after resetting, these two ports function as ordinary ports (see Fig. 16).

FUNCTIONS OF MELPS 8-41 SLAVE MICROCOMPUTERS

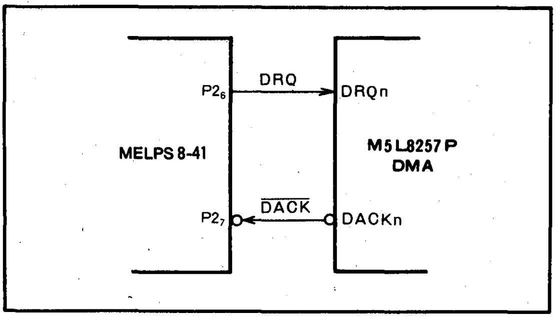


Fig.16 DMA control

When the EN DMA instruction is executed, $P2_6$ becomes the DRQ (DMA request) output. Subsequently, when $P2_6$ is set to "1", DRQ becomes "1" and DMA-based data transfer is requested.
DRQ is cleared when the $\overline{\text{DACK}} \cdot \overline{\text{R}}$, $\overline{\text{DACK}} \cdot \overline{\text{W}}$ or EN DMA instruction is executed.

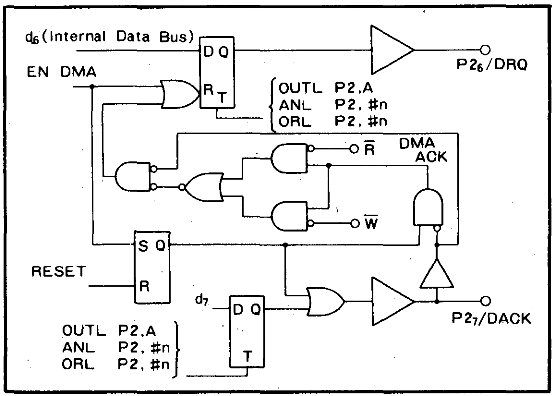


Fig.17 Internal configuration of DMA control

When the EN DMA instruction is executed, $P2_7$ becomes the $\overline{\text{DACK}}$ (DMA acknowledge) input. The $\overline{\text{DACK}}$ input is used as the chip select input for DMA transfer. There is, therefore, no connection with the state of $\overline{\text{S}}$ (chip select) during DMA transfer.

Interrupt Request to Master CPU

Ports $P2_4$ and $P2_5$ of MELPS 8-41 can be used not only as ordinary input/output ports but also as the outputs of the $\overline{\text{IBF}}$ (input buffer full) flag and OBF (output buffer full) flag. Immediately after resetting, both ports function as input ports.

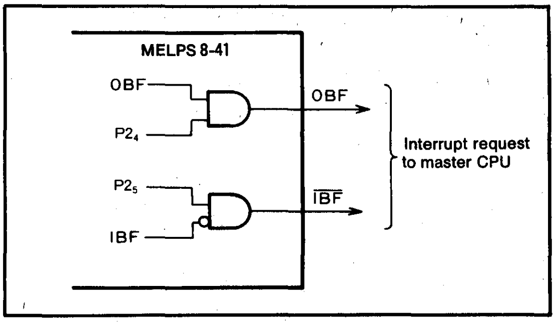


Fig.18 Interrupt request to master CPU

When the EN FLAGS instruction is executed, $P2_4$ functions as the OBF pin and $P2_5$ as the $\overline{\text{IBF}}$ pin. "1" must be output to both pins so that the OBF and $\overline{\text{IBF}}$ flag states are output to each pin, respectively. These states are not output while "0" is output to the pins. The OBF flag output indicates that data has been output to the output data bus buffer register; the $\overline{\text{IBF}}$ flag output indicates that the input data/command bus buffer register is in the data accept enable mode.

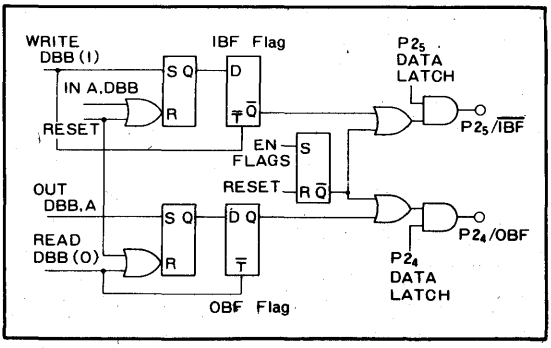


Fig.19 Internal configuration of IBF/OBF

FUNCTIONS OF MELPS 8-41 SLAVE MICROCOMPUTERS

INSTRUCTION CODES

D_3~D_0	Hexadecimal notation \ 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			NOV @ 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 OXX	CALL OXX	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 RBO	SEL RB1	EN DMA	EN FLAOS
0110	6		JTF m	JNT0 m	JT0 m	JNT1 m	JT1 m		JF1 m	JOBF m	JNZ m		JF0 m	JZ m	JNIBF 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		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

FUNCTIONS OF MELPS 8-41 SLAVE MICROCOMPUTERS

MACHINE INSTRUCTIONS

Item / Type	Mnemonic	Instruction code			Bytes	Cycles	Function
		$D_7D_6D_5D_4$	$D_3D_2D_1D_0$	Hexadecimal			
Transfer	MOV A, # n	0 0 1 0 $n_7n_6n_5n_4$	0 0 1 1 $n_3n_2n_1n_0$	2 3 n	2	2	(A) ← n
	MOV A, Rr	1 1 1 1	1 $r_2r_1r_0$	F 8 + r	1	1	(A) ← (Rr) r = 0~7
	MOV Rr, A	1 0 1 0	1 $r_2r_1r_0$	A 8 + r	1	1	(Rr) ← (A) r = 0~7
	MOV Rr, #n	1 0 1 1 $n_7n_6n_5n_4$	1 $r_2r_1r_0$ $n_3n_2n_1n_0$	B 8 + r n	2	2	(Rr) ← n r = 0~7
	XCH A, Rr	0 0 1 0	1 $r_2r_1r_0$	2 8 + r	1	1	(A) ←→ (Rr) r = 0~7
Indirect Addressing Transfer	MOV A, @Rr	1 1 1 1	0 0 0 r_0	F 0 + r	1	1	(A) ← (M (Rr)) r = 0~1
	MOV @Rr, A	1 0 1 0	0 0 0 r_0	A 0 + r	1	1	(M (Rr)) ← (A) r = 0~1
	MOV @Rr, #n	1 0 1 1 $n_7n_6n_5n_4$	0 0 0 r_0 $n_3n_2n_1n_0$	B 0 + r n	2	2	(M (Rr)) ← n r = 0~1
	MOVP A, @A	1 0 1 0	0 0 1 1	A 3	1	2	(A) ← (M (A))
	MOVP3 A, @A	1 1 1 0	0 0 1 1	E 3	1	2	(A) ← (M (page 3, A))
	XCH A, @Rr	0 0 1 0	0 0 0 r_0	2 0 + r	1	1	(A) ←→ (M (Rr)) r = 0~1
	XCHD A, @Rr	0 0 1 1	0 0 0 r_0	3 0 + r	1	1	(A_0~A_3) ←→ (M (Rr_0~Rr_3)) r = 0~1
Status Control	MOV A, PSW	1 1 0 0	0 1 1 1	C 7	1	1	(A) ← (PSW)
	MOV PSW, A	1 1 0 1	0 1 1 1	D 7	1	1	(PSW) ← (A) (C) ← (A_7), (AC) ← (A_6)
	MOV STS, A	1 0 0 1	0 0 0 0	9 0	1	1	(STS) ← (A) (ST_4~ST_7) ← (A_4~A_7)
	CLR C	1 0 0 1	0 1 1 1	9 7	1	1	(C) ← 0
	CPL C	1 0 1 0	0 1 1 1	A 7	1	1	(C) ← ($\overline{C}$)
	CLR F_0	1 0 0 0	0 1 0 1	8 5	1	1	(F_0) ← 0
	CPL F_0	1 0 0 1	0 1 0 1	9 5	1	1	(F_0) ← ($\overline{F_0}$)

FUNCTIONS OF MELPS 8-41 SLAVE MICROCOMPUTERS

Affected carry			Description
C	AC	Note	
			Transfers data n to register A.
			Transfers contents of register Rr to register A.
			Transfers contents of register A to register Rr.
			Transfers data n to register Rr.
			Exchanges contents of register Rr with contents of register A.
			Transfers contents of memory location of current page, whose address is in register Rr, to register A.
			Transfers contents of register A to memory location of current page whose address is in register Rr.
			Transfers data n to memory location of current page whose address is in register Rr.
			Transfers data of memory location of current page whose address is in register A to register A.
			Transfers data of memory location of page 3 whose address is in register A to regsiter A.
			Exchanges contents of memory location of current page whose address is in register Rr with contents of register A.
			Exchanges contents of low-order 4 bits of register with low-order 4 bits of memory location of current page whose address is in register Rr.
			Transfers contents of program status word to register A.
○	○		Transfers contents of register A to program status word.
		2	Transfers contents of register A to status register.
○			Clears carry flag and resets it to 0.
○			Complements contents of carry flag.
			Clears flag F_0 and resets it to 0.
			Complements contents of flag F_0.

FUNCTIONS OF MELPS 8-41 SLAVE MICROCOMPUTERS

Item / Type	Mnemonic	Instruction code			Bytes	Cycles	Function
		$D_7D_6D_5D_4$	$D_3D_2D_1D_0$	Hexadecimal			
Status Control	CLR F_1	1 0 1 0	0 1 0 1	A 5	1	1	$(F_1) \leftarrow 0$
	CPL F_1	1 0 1 1	0 1 0 1	B 5	1	1	$(F_1) \leftarrow (\overline{F}_1)$
Arithmetic	ADD A, #n	0 0 0 0 $n_7n_6n_5n_4$	0 0 1 1 $n_3n_2n_1n_0$	0 3 n	2	2	(A) ← (A) + n
	ADD A, Rr	0 1 1 0	1 r_2 r_1 r_0	6 8 + r	1	1	(A) ← (A) + (Rr) r = 0~7
	ADD A, @Rr	0 1 1 0	0 0 0 r_0	6 0 + r	1	1	(A) ← (A) + (M (Rr)) r = 0~1
	ADDC A, #n	0 0 0 1 $n_7n_6n_5n_4$	0 0 1 1 $n_3n_2n_1n_0$	1 3 n	2	2	(A) ← (A) + n + (C)
	ADDC A, Rr	0 1 1 1	1 r_2 r_1 r_0	7 8 + r	1	1	(A) ← (A) + (Rr) + (C) r = 0~7
	ADDC A, @Rr	0 1 1 1	0 0 0 r_0	7 0 + r	1	1	(A) ← (A) + (M (Rr)) + (C) r = 0~1
	ANL A, #n	0 1 0 1 $n_7n_6n_5n_4$	0 0 1 1 $n_3n_2n_1n_0$	5 3 n	2	2	(A) ← (A) ∧ n
	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
	ANL A, @Rr	0 1 0 1	0 0 0 r_0	5 0 + r	1	1	(A) ← (A) ∧ (M (Rr)) r = 0~1
	ORL A, # n	0 1 0 0 $n_7n_6n_5n_4$	0 0 1 1 $n_3n_2n_1n_0$	4 3 n	2	2	(A) ← (A) ∨ n
	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
	ORL A, @Rr	0 1 0 0	0 0 0 r_0	4 0 + r	1	1	(A) ← (A) ∨ (M (Rr)) r = 0~1
	XRL A, # n	1 1 0 1 $n_7n_6n_5n_4$	0 0 1 1 $n_3n_2n_1n_0$	D 3 n	2	2	(A) ← (A) ⊻ n
	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
	XRL A, @Rr	1 1 0 1	0 0 0 r_0	D 0 + r	1	1	(A) ← (A) ⊻ (M (Rr)) r = 0~1
	INC A	0 0 0 1	0 1 1 1	1 7	1	1	(A) ← (A) + 1
	DEC A	0 0 0 0	0 1 1 1	0 7	1	1	(A) ← (A) − 1

FUNCTIONS OF MELPS 8-41 SLAVE MICROCOMPUTERS

Affected carry			Description
C	AC	Note	
			Clears flag F_1 and resets it to 0.
			Complements contents of flag F_1.
○	○	1	Adds data n to contents of register A and sets carry flags to 1 if there is an overflow; otherwise resets carry flags to 0. The result is stored in register A.
○	○	1	Adds contents of register Rr to contents of register A and set carry flags to 1 if there is an overflow; otherwise resets carry flags to 0. The result is stored in register A.
○	○	1	Adds contents of register A and contents of memory location of current page whose address is in register Rr and sets carry flags to 1 if there is an overflow; otherwise resets carry flags to 0. The result is stored in register A.
○	○	1	Adds carry and data n to contents of register A and sets carry flags to 1 if there is an overflow; otherwise resets carry flags to 0. The result is stored in register A.
○	○	1	Adds carry and contents of register Rr to contents of register A and sets carry flags to 1 if there is an overflow; otherwise resets carry flags to 0. The result is stored in register A.
○	○	1	Adds carry and contents of memory location of current page whose address is in register Rr to contents of register A and sets carry flags to 1 if there is an overflow; otherwise resets carry flags to 0. The result is stored in register A.
			Logical product of contents of register A and data n is stored in register A.
			Logical product of contents of register A and contents of register Rr is stored in register A.
			Logical product of contents of register A and contents of memory location of current page whose address is in register Rr is stored in register A.
			Logical sum of contents of register A and data n is stored in register A.
			Logical sum of contents of register A and contents of register Rr is stored in register A.
			Logical sum of contents of register A and contents of memory location of current page whose address is in register Rr is stored in register A.
			Exclusive OR of contents of register A and data n is stored in register A.
			Exclusive OR of contents of register A and contents of register Rr is stored in register A.
			Exclusive OR of contents of register A and contents of memory location of current page whose address is in register Rr, is stored in register A.
			Increments contents of register A by 1. The result is stored in register A.
			Decrements contents of register A by 1. The result is stored in register A.

FUNCTIONS OF MELPS 8-41 SLAVE MICROCOMPUTERS

Item / Type	Mnemonic	Instruction code			Bytes	Cycles	Function
		$D_7D_6D_5D_4$	$D_3D_2D_1D_0$	Hexadecimal			
Arithmetic	CLR A	0 0 1 0	0 1 1 1	2 7	1	1	(A) ← 0
	CPL A	0 0 1 1	0 1 1 1	3 7	1	1	$(A) \leftarrow (\overline{A})$
	DA A	0 1 0 1	0 1 1 1	5 7	1	1	(A) decimal conversion
Shift	SWAP A	0 1 0 0	0 1 1 1	4 7	1	1	$(A_4 \sim A_7) \longleftrightarrow (A_0 \sim A_3)$
	RL A	1 1 1 0	0 1 1 1	E 7	1	1	$(A_{n+1}) \leftarrow (A_n)$ $(A_0) \leftarrow (A_7)$ $n = 0 \sim 6$
	RLC A	1 1 1 1	0 1 1 1	F 7	1	1	$(A_{n+1}) \leftarrow (A_n)$ $(A_0) \leftarrow (C)$ $(C) \leftarrow (A_7)$ $n = 0 \sim 6$
	RR A	0 1 1 1	0 1 1 1	7 7	1	1	$(A_n) \leftarrow (A_{n+1})$ $(A_7) \leftarrow (A_0)$ $n = 0 \sim 6$
	RRC A	0 1 1 0	0 1 1 1	6 7	1	1	$(A_n) \leftarrow (A_{n+1})$ $(A_7) \leftarrow (C)$ $(C) \leftarrow (A_0)$ $n = 0 \sim 6$
Register arithmetic	INC Rr	0 0 0 1	1 r_2 r_1 r_0	1 8 + r	1	1	$(Rr) \leftarrow (Rr) + 1$ $r = 0 \sim 7$
	INC @Rr	0 0 0 1	0 0 0 r_0	1 0 + r	1	1	$(M(Rr)) \leftarrow (M(Rr)) + 1$ $r = 0 \sim 1$
	DEC Rr	1 1 0 0	1 r_2 r_1 r_0	C 8 + r	1	1	$(Rr) \leftarrow (Rr) - 1$ $r = 0 \sim 7$
Jump	JMP m	$m_{10}m_9m_8$ 0 $m_7m_6m_5m_4$	0 1 0 0 $m_3m_2m_1m_0$	0 4 + $m_{8\sim10}$ m	2	2	$(PC_8 \sim PC_{10}) \leftarrow m_8 \sim m_{10}$ $(PC_0 \sim PC_7) \leftarrow m_0 \sim m_7$
	JMPP @A	1 0 1 1	0 0 1 1	B 3	1	2	$(PC_0 - PC_7) \leftarrow (M(A))$
Conditional Jump	JBb m	$b_7b_6b_5$ 1 $m_7m_6m_5m_4$	0 0 1 0 $m_3m_2m_1m_0$	1 2 + b m	2	2	When $(A_b) = 1$, $(PC_0 \sim PC_7) \leftarrow m$ When $(A_b) = 1$, $(PC) \leftarrow (PC) + 2$ $b_7b_6b_5 = 0 \sim 7$
	JNIBF m	1 1 0 1 $m_7m_6m_5m_4$	0 1 1 0 $m_3m_2m_1m_0$	D 6	2	2	When (IBF) = 0, $(PC_0 - PC_7) \leftarrow m$
	JOBF m	1 0 0 0 $m_7m_6m_5m_4$	0 1 1 0 $m_3m_2m_1m_0$	8 6 m	2	2	When (OBF) = 1, $(PC_0 - PC_7) \leftarrow m$
	JTF m	0 0 0 1 $m_7m_6m_5m_4$	0 1 1 0 $m_3m_2m_1m_0$	1 6 m	2	2	When (TF) = 1, $(PC_0 \sim PC_7) \leftarrow m$ When (TF) = 0, $(PC) \leftarrow (PC) + 2$
	DJNZ Rr, m	1 1 1 0 $m_7m_6m_5m_4$	1 r_2 r_1 r_0 $m_3m_2m_1m_0$	E 8 + r m	2	2	$(Rr) \leftarrow (Rr) - 1$ $r = 0 \sim 7$ When (Rr) ≠ 0, $(PC_0 \sim PC_7) \leftarrow m$ When (Rr) = 0, $(PC) \leftarrow (PC) + 2$
	JC m	1 1 1 1 $m_7m_6m_5m_4$	0 1 1 0 $m_3m_2m_1m_0$	F 6 m	2	2	When (C) = 1, $(PC_0 \sim PC_7) \leftarrow m$ When (C) = 0, $(PC) \leftarrow (PC) + 2$

FUNCTIONS OF MELPS 8-41 SLAVE MICROCOMPUTERS

Affected carry			Description
C	AC	Note	
			Clears contents of register A and resets to 0.
			Forms 1's complement of register A and stores it in register A.
○	○	1	Contents of register A are converted to binary coded decimal notation and stored in register A.
			Exchanges contents of bits 0～3 of register A with contents of bits 4～7 of register A.
			Shifts contents of register A left one bit. MSB A_7 is rotated to LSB A_0.
○			Shifts contents of register A left one bit. MSB A_7 is shifted to carry flag and carry flag is shifted to LSB A_0.
			Shifts contents of register A right one bit. LSB A_0 is rotated to MSB A_7.
○			Shifts contents of register A right one bit. LSB A_0 is shifted to carry flag and carry flag is shifted to MSB A_7.
			Increments contents of register Rr by 1. The result is stored in register Rr.
			Increments contents of memory location of current page whose address is in register Rr by 1.
			Decrements contents of register Rr by 1. The result is stored in register Rr.
			Jumps unconditionally to address m.
			Jumps to memory location of current page whose address is in register A; but when instruction executed was in address 255, jumps to next page.
			Jumps to address m of current page when bit b of register A is 1. Executes next instruction when bit b of register A is 0.
			Jumps to address m of current page when IBF is 0.
			Jumps to address m of current page when OBF is 1.
			Jumps to address m of current page when timer/counter overflow flag is 1; flag is cleared after execution.
			Decrements contents of register Rr by 1; jumps to address m of current page when result is not 0.
			jumps to address m of current page if carry flag is 1.

FUNCTIONS OF MELPS 8-41 SLAVE MICROCOMPUTERS

Item Type	Mnemonic	Instruction code			Bytes	Cycles	Function
		$D_7D_6D_5D_4$	$D_3D_2D_1D_0$	Hexadecimal			
Conditional Jump	JNC m	1 1 1 0 $m_7m_6m_5m_4$	0 1 1 0 $m_3m_2m_1m_0$	E 6 m	2	2	When (C) = 0, $(PC_0 \sim PC_7) \leftarrow m$ When (C) = 1, $(PC) \leftarrow (PC)+2$
	JZ m	1 1 0 0 $m_7m_6m_5m_4$	0 1 1 0 $m_3m_2m_1m_0$	C 6 m	2	2	When (A) = 0, $(PC_0 \sim PC_7) \leftarrow m$ When (A) ≠ 0, $(PC) \leftarrow (PC)+2$
	JNZ m	1 0 0 1 $m_7m_6m_5m_4$	0 1 1 0 $m_3m_2m_1m_0$	9 6 m	2	2	When (A) ≠ 0, $(PC_0 \sim PC_7) \leftarrow m$ When (A) = 0, $(PC) \leftarrow (PC)+2$
	JT0 m	0 0 1 1 $m_7m_6m_5m_4$	0 1 1 0 $m_3m_2m_1m_0$	3 6 m	2	2	When $(T_0) = 1$, $(PC_0 \sim PC_7) \leftarrow m$ When $(T_0) = 0$, $(PC) \leftarrow (PC)+2$
	JNT0 m	0 0 1 0 $m_7m_6m_5m_4$	0 1 1 0 $m_3m_2m_1m_0$	2 6 m	2	2	When $(T_0) = 0$, $(PC_0 \sim PC_7) \leftarrow m$ When $(T_0) = 1$, $(PC) \leftarrow (PC)+2$
	JT1 m	0 1 0 1 $m_7m_6m_5m_4$	0 1 1 0 $m_3m_2m_1m_0$	5 6 m	2	2	When $(T_1) = 1$, $(PC_0 \sim PC_7) \leftarrow m$ When $(T_1) = 0$, $(PC) \leftarrow (PC)+2$
	JNT1 m	0 1 0 0 $m_7m_6m_5m_4$	0 1 1 0 $m_3m_2m_1m_0$	4 6 m	2	2	When $(T_1) = 0$, $(PC_0 \sim PC_7) \leftarrow m$ When $(T_1) = 1$, $(PC) \leftarrow (PC)+2$
	JF0 m	1 0 1 1 $m_7m_6m_5m_4$	0 1 1 0 $m_3m_2m_1m_0$	B 6 m	2	2	When $(F_0) = 1$, $(PC_0 \sim PC_7) \leftarrow m$ When $(F_0) = 0$, $(PC) \leftarrow (PC)+2$
	JF1 m	0 1 1 1 $m_7m_6m_5m_4$	0 1 1 0 $m_3m_2m_1m_0$	7 6 m	2	2	When $(F_1) = 1$, $(PC_0 \sim PC_7) \leftarrow m$ When $(F_1) = 0$, $(PC) \leftarrow (PC)+2$
Subroutine	CALL m	$m_{10}m_9m_8$ 1 $m_7m_6m_5m_4$	0 1 0 0 $m_3m_2m_1m_0$	1 4 + $m_8 \sim m_{10}$ m	2	2	$((SP)) \leftarrow (PC)(PSW_4 \sim PSW_7)$ $(SP) \leftarrow (SP)+1$ $(PC_0 \sim PC_{10}) \leftarrow m$
	RET	1 0 0 0	0 0 1 1	8 3	1	2	$(SP) \leftarrow (SP)-1$ $(PC) \leftarrow ((SP))$
	RETR	1 0 0 1	0 0 1 1	9 3	1	2	$(SP) \leftarrow (SP)-1$ $(PC)(PSW_4 \sim PSW_7) \leftarrow ((SP))$
Input/Output	IN A, P_p	0 0 0 0	1 0 p_1 p_0	0 8 + p	1	2	$(A) - (P_p)$ $p = 1 \sim 2$
	OUTL P_p, A	0 0 1 1	1 0 p_1 p_0	3 8 + p	1	2	$(P_p) \leftarrow (A)$ $P = 1 \sim 2$
	ANL P_p, #n	1 0 0 1 $n_7n_6n_5n_4$	1 0 p_1 p_0 $n_3n_2n_1n_0$	9 8 + p n	2	2	$(P_p) \leftarrow (P_p)An$ $p = 1 \sim 2$
	ORL P_p, #n	1 0 0 0 $n_7n_6n_5n_4$	1 0 p_1 p_0 $n_3n_2n_1n_0$	8 8 + p n	2	2	$(P_p) \leftarrow (P_p)V_n$ $p = 1 \sim 2$
	IN A, DBB	0 0 1 0	0 0 1 0	2 2	1	1	$(A) \leftarrow (DBB)$
	OUT DBB, A	0 0 0 0	0 0 1 0	0 2	1	1	$(DBB) \leftarrow (A)$

FUNCTIONS OF MELPS 8-41 SLAVE MICROCOMPUTERS

Affected carry			Description
C	AC	Note	
			Jumps to address m of current page if carry flag is 0.
			Jumps to address m of current page when contents of register A are 0.
			Jumps to address m of current page when contents of register A are not 0.
			Jumps to address m of current page when flag T_0 is 1.
			Jumps to address m of current page when flag T_0 is 0.
			Jumps to address m of current page when flag T_1 is 1.
			Jumps to address m of current page when flag T_1 is 0.
			Jumps to address m of current page when flag F_0 is 1.
			Jumps to address m of current page when flag F_1 is 1.
			Calls subroutine from address m. The program counter and the high-order 4 bits of PSW are stored in address indicated by stack pointer (SP). SP is incremented by 1 and m is transferred to $PC_0 \sim PC_{10}$.
			SP is decremented by 1. Program counter is restored to saved setting in stack indicated by stack pointer. $PSW_4 \sim PSW_7$ are not changed and interrupt disable is maintained.
			SP is decremented by 1. Program counter and high-order 4 bits of PSW are restored with saved data in stack indicated by stack pointer. Interrupt becomes enabled after execution is completed.
			Loads contents of Pp to register A.
			Output latches contents of register A to Pp.
			Logical product of contents of Pp and data n; outputs result to Pp.
			Logical sum of contents of Pp and data n; outputs result to Pp.
			Enters contents of data bus buffer (DBB) into register A and resets IBF.
			Outputs contents of register A to data bus buffer (DBB) and sets OBF.

FUNCTIONS OF MELPS 8-41 SLAVE MICROCOMPUTERS

Item / Type	Mnemonic	Instruction code			Bytes	Cycles	Function
		$D_7D_6D_5D_4$	$D_3D_2D_1D_0$	Hexadecimal			
I/O Expander Control	MOVD A, Pp	0 0 0 0	1 1 P_1 P_0	0 C + P_1P_0	1	2	$(A_0 \sim A_3) \leftarrow (P_{p0} \sim P_{p3})$ $(A_4 \sim A_7) \leftarrow 0 \quad p = 4 \sim 7$
	MOVD P_p, A	0 0 1 1	1 1 P_1 P_0	3 C + P_1P_0	1	2	$(P_{p0} \sim P_{p3}) \leftarrow (A_0 \sim A_3)$ $P = 4 \sim 7$
	ANLD P_p, A	1 0 0 1	1 1 P_1 P_0	9 C + P_1P_0	1	2	$(P_{p0} \sim P_{p3}) \leftarrow (P_{p0} \sim P_{p3}) \wedge (A_0 \sim A_3)$ $p = 4 \sim 7$
	ORLD P_p, A	1 0 0 0	1 1 P_1 P_0	8 C + P_1P_0	1	2	$(P_{p0} \sim P_{p3}) \leftarrow (P_{p0} \sim P_{p3}) V (A_0 \sim A_3)$ $p = 4 \sim 7$
Timer/Counter Control	MOV A, T	0 1 0 0	0 0 1 0	4 2	1	1	(A) ← (T)
	MOV T, A	0 1 1 0	0 0 1 0	6 2	1	1	(T) ← (A)
	STRT T	0 1 0 1	0 1 0 1	5 5	1	1	
	STRT CNT	0 1 0 0	0 1 0 1	4 5	1	1	
	STOP TCNT	0 1 1 0	0 1 0 1	6 5	1	1	
	EN TCNTI	0 0 1 0	0 1 0 1	2 5	1	1	(TCNTF) ← 1
	DIS TCNTI	0 0 1 1	0 1 0 1	3 5	1	1	(TCNTF) ← 0
Control	EN I	0 0 0 0	0 1 0 1	0 5	1	1	(INTF) ← 1
	DIS I	0 0 0 1	0 1 0 1	1 5	1	1	(INTF) ← 0
	SEL RB_0	1 1 0 0	0 1 0 1	C 5	1	1	(BS) ← 0
	SEL RB_1	1 1 0 1	0 1 0 1	D 5	1	1	(BS) ← 1
	EN DMA	1 1 1 0	0 1 0 1	E 5	1	1	
	EN FLAGS	1 1 1 1	0 1 0 1	F 5	1	1	$(P2_4) \leftarrow (OBF)$ $(P2_5) \leftarrow (IBF)$
Misc.	NOP	0 0 0 0	0 0 0 0	0 0	1	1	

Note 1 : Executing an instruction may produce a carry (overflow or underflow). The carry may be lost or it may be transferred to C or AC. The (○) mark indicates a carry which affects C or AC. The detail affection of carries for instructions ADD, ADDC and DA is as follows:

(C) ← 1   At overflow of accumulator
(C) ← 0   At no overflow of accumulator
(AC) ← 1   At overflow of bit 3 of accumulator
(AC) ← 0   At no overflow

2 : The contents of ST_4-ST_7 are read when host computer reads status of MELPS 8-41.

FUNCTIONS OF MELPS 8-41 SLAVE MICROCOMPUTERS

Affected carry			Description	
C	AC	Note		
			Inputs contents of Pp to low-order 4 bits of register A and inputs 0 to high-order 4 bits of register A.	Pp's used for multiplying 8243 ports are P4~P7. Correspondence to P2, P1 bits is shown below. P4···$P_1P_2 = 00$ P5···$P_1P_2 = 01$ P6···$P_1P_2 = 10$ P7···$P_1P_2 = 11$
			Outputs low-order 4 bits of register A to Pp.	
			Logical product of the low-order 4 bits of register A and contents of Pp; Pp contains result.	
			Logical sum of low-order 4 bits of register A and contents of Pp; Pp contains result.	
			Transfers contents of timer/event counter to register A.	
			Transfers contents of register A to timer/event counter.	
			Starts timer operation of timer/event counter. Count cycle is 480 times master oscillation.	
			Starts operation as event counter of timer/event counter. Counts up when pin T_1 changes from high to low input level.	
			Stops operation of timer or event counter.	
			Enables interrupt of timer/event counter.	
			Disables interrupt of timer/event counter. Resets interrupt flip-flop of CPU which is set during CPU stand-by. Timer flag is not affected.	
			Enables external interrupt.	
			Disables external interrupt.	
			Selects working register bank 0.	
			Selects working register bank 1.	
			Enables DMA handshaking line.	
			Enables master interrupt.	
			No operation. Execution time is 1 machine cycle.	

FUNCTIONS OF MELPS 8-41 SLAVE MICROCOMPUTERS

Item	Details of execution
RESET input low level	TF (Timer Flag) ← 0 TIRF (Timer INT Request FF) ← 0 TCNTF (Timer INT Enable FF) ← 0 INTF (External INT Enable FF) ← 0 IEF (INT Enable FF) ← 1 IBF ← 0 EIPF (External Interrupt Pending FF) ← 0
JTF execution	TF (Timer Flag) ← 0
Timer/event Counter overflow	TF (Timer Flag) ← 1 TCNTE (Timer INT Enable FF) = 1 When TIRF (Timer INT Request FF) ← 1
EN TNCTI execution	TCNTF (Timer INT Enable FF) ← 1
DIS TNCTI execution	TCNTF (Timer INT Enable FF) ← 0
EN I execution	INTF (External INT Enable FF) ← 1
DIS I execution	INTF (External INT Enable FF) ← 0
RETR execution	IEF (INT Enable FF) ← 1

Symbol	Contents	Symbol	Contents
A	8-bit register (accumulator)	PC	Program counter
$A_0 \sim A_3$	Low-order 4 bits of register A	$PC_0 \sim PC_7$	Low-order 8 bits of program counter
$A_4 \sim A_7$	High-order 4 bits of register A	$PC_8 \sim PC_{10}$	High-order 3 bits of program counter
$A_0 \sim A_n$, A_{n+1}	Bits of register A	PSW	Program status word
b	Value of bits 5-7 of first byte machine code	Rr	Register designator
$b_7b_6b_5$	Bits 5-7 of first byte machine code	r	Register number
BS	Register bank select	r_0	Value of bit 0 of machine code
AC	Auxiliary carry flag	$r_2r_1r_0$	Value of bits 0-2 of machine code
C	Carry flag	$s_2s_1s_0$	Value of bits 0-2 of stack pointer
DBB	Data bus buffer	SP	Stack pointer
F_0	Flag 0	ST_4ST_7	Bits 4-7 of status register
F_1	Flag 1	STS	System status
INTF	External interrupt enable flip-flop	T	Timer/event counter
IBF	Input buffer full flag	T_0	Test pin 0
m	Destination address	T_1	Test pin 1
$m_7m_6m_5m_4m_3m_2m_1m_0$	Second byte (low-order 8 bits) machine code	TCNTF	Timer/event counter interrupt flip-flop
	corresponding to destination address	TF	Timer flag
$m_{10}m_9m_8$	Bits 5-7 of first byte (high-order 3 bits) machine code	#	Symbol to indicate immediate data
(M(A))	Content of memory location addressed by register A	@	Symbol to indicate content of memory location
(M(Rr))	Content of memory location addressed by register Rr		addressed by register
(Mx(Rr))	Content of external memory location addressed by	←	Shows direction of data flow
	register Rr	↔	Exchanges contents of data
n	Value of immediate data	( )	Contents of register, memory location or flag
$n_7n_6n_5n_4n_3n_2n_1n_0$	Immediate data of second byte machine code	∧	Logical AND
OBF	Output buffer full flag	∨	Logical OR
p	Port number	∀	Exclusive OR
P_P	Port designator	—	Negation
p_1p_0	Bits of machine code corresponding to port number	○	Content of flag is set or reset after execution

DESCRIPTION

The M5L8041A-XXXP is a general-purpose, programmable interface device deisgned for use with a variety of 8-bit microcomputer systems. This device is fabricated using N-channel sillicon-gate ED-MOS technology.

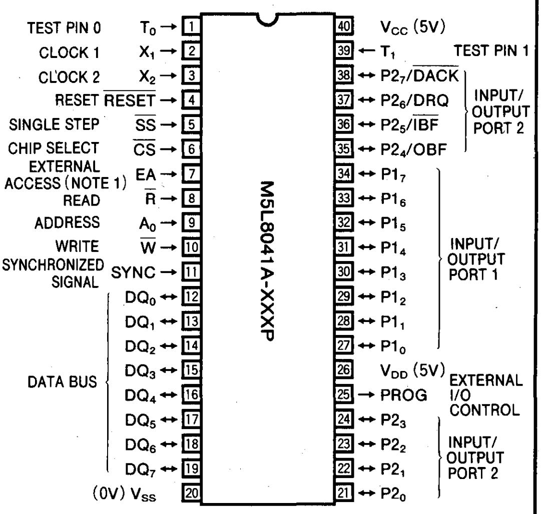
FEATURES

- Mask ROM ································· 1024-word by 8-bit
- Static RAM ································· 64-word by 8-bit
- 18 programmable I/O pins
- Asynchronous data register for interface to master processor
- 8-bit CPU, ROM, RAM, I/O, timer, clock and low power, stand-by mode
- Single 5V supply
- Alternative to custom LSI
- Interchangeable with i 8041A

APPLICATION

Alternative to custom LSI for peripheral interface

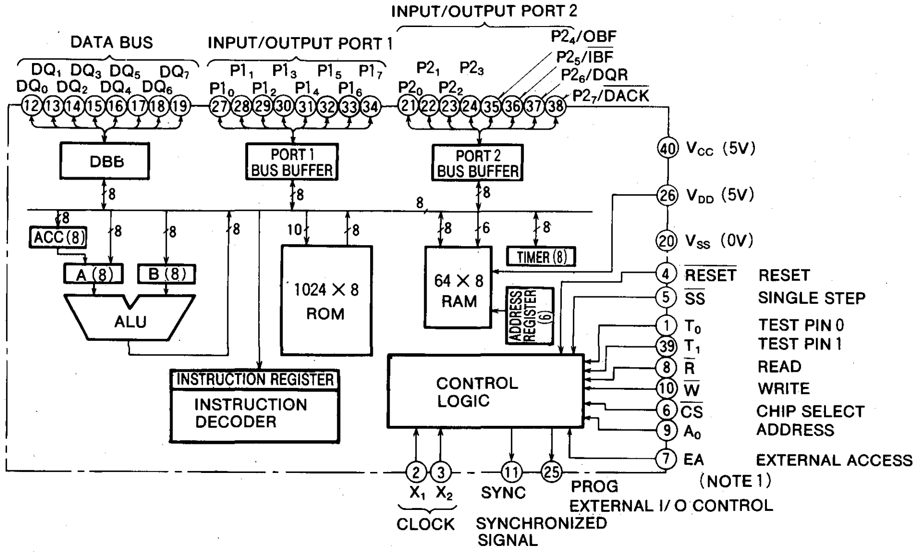
PIN CONFIGURATION (TOP VIEW)



Outline 40P4

Note 1 : Connect to V_{SS} in the operating condition.

BLOCK DIAGRAM



FUNCTION

The M5L8041A-XXXP is designed as an ordinary 8-bit CPU peripheral LSI chip and it contains a small stand-alone microcomputer. Although this microcomputer functions independently, when it is used as a peripheral controller, it is called the slave microcomputer in contrast to the master computer. These two devices can transfer the data alternatively through the buffer register between them. The M5L8041A-XXXP contains the buffer register to use this LSI as a slave microcomputer and it can be accessed in the same way as other standard peripheral devices. Since the M5L8041A-XXXP is a complete microcomputer, it is easy to develop a user-oriented mask-programmed peripheral LSI only by changing the control software.

PIN DESCRIPTION

Pin	Name	Input or output	Function
V_{SS}	Ground	—	Connected to a 0V supply (ground).
V_{CC}	Main power supply	—	Connected to a 5V supply.
V_{DD}	Power supply	—	Connected to a 5V supply.
			Used as a memory hold when V_{CC} is cut off.
T_0	Test pin 0	Input	Provides external control of conditional program jumps (JTO/JNTO instructions).
X_1, X_2	Crystal inputs	Input	An internal clock circuit is provided so that by connecting an RC circuit or crystal to these input pins the clock frequency can be determined. Pins X_1 and X_2 can also be used to input an external clock signal.
$\overline{RESET}$	Reset	Input	CPU initialization input.
$\overline{SS}$	Single step	Input	Used to halt the execution of a command by the CPU. When used in combination with the SYNC signal, the command execution of the CPU can be halted every instruction to enable single step operation.
$\overline{CS}$	Chip select input	Input	Chip select input data bus control.
EA	External access	Input	Normally maintained at 0V.
$\overline{R}$	Read enable signal	Input	Serves as the read signal when the master CPU is accepting data on the data bus from the M5L8041A-XXXP.
A_0	Address input	Input	An address input used to indicate whether the signal on the data bus is data or a command.
$\overline{W}$	Write enable signal	Input	Serves as the write signal when the master CPU is outputting data from the bus to the M5L8041A-XXXP.
SYNC	Sync signal output	Output	Output 1 time for each machine cycle.
DQ_0~DQ_7	Data bus	Input/output	Three-state, bidirectional data bus. Data bus is used to interface the M5L8041A-XXXP to a master system data bus.
$P2_0$~$P2_7$	Port 2	Input/output	Quaisi-bidirectional port. When used as an input port, FF_{16} must first be output to this port. After resetting, however, when not used afterwards as an output port, this is not necessary.
			$P2_0$~$P2_3$ are used when the M5L8243P I/O port expander is used.
PROG	Program	Output	Serves as the strobe signal when the M5L8243P I/O expander is used.
$P1_0$~$P1_7$	Port 1	Input/output	Quaisi-bidirectional port. When used as an input port, FF_{16} must first be output to this port. After resetting, however, when not used afterwards as an output port, this is not necessary.
T_1	Test pin 1	Input	Provides external control of conditional program jumps (JT1/JNT1 instructions).
			Can serve as the input pin for the event counter (STRT CNT instructions).

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
P_d	Power dissipation	$T_a = 25$℃	1500	mW
T_{opr}	Operating temperature range		−20～75	℃
T_{stg}	Storage temperature range		−65～150	℃

RECOMMENDED OPERATING CONDITIONS

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_{IH}	High-level input voltage	2			V
V_{IL}	Low-level input voltage			0.8	V
$f_{(\phi)}$	Operating frequency	1		6	MHz

ELECTRICAL CHARACTERISTICS ($T_a = -20～75$℃, $V_{CC} = 5\,V \pm 10\%$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{IL}	Low-level input voltage		−0.5		0.8	V
V_{IH1}	High-level input voltage (all except 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_{OL1}	Low-level output voltage (DQ_0～DQ_7, SYNC)	$I_{OL} = 2mA$			0.45	V
V_{OL2}	Low-level output voltage (all except DQ_0～DQ_7, SYNC, PROG)	$I_{OL} = 1.6\,mA$			0.45	V
V_{OL3}	Low-level output voltage (PROG)	$I_{OL} = 1mA$			0.45	V
V_{OH1}	High-level output voltage (DQ_0～DQ_7)	$I_{OH} = -400\mu A$	2.4			V
V_{OH2}	High-level output voltage (all other outputs)	$I_{OH} = -50\mu A$	2.4			V
I_I	Input leakage current (T_0, T_1, $\overline{RD}$, $\overline{WR}$, $\overline{CS}$, A_0)	$V_{SS} \leqq V_I \leqq V_{CC}$	−10		10	μA
I_{OZL}	Off-state output leakage current (DQ_0～DQ_7)	$V_{SS} + 0.45 \leqq V_O \leqq V_{CC}$	−10		10	μA
I_{IL1}	Low-level input current ($P1_0$～$P1_7$, $P2_0$～$P2_7$)	$V_{IL} = 0.8V$	−0.5			mA
I_{IL2}	Low-level input current ($\overline{RESET}$, $\overline{SS}$)	$V_{IL} = 0.8V$	−0.2			mA
I_{DD}	Supply current from V_{DD}			6	15	mA
$I_{CC} + I_{DD}$	Total supply current			65	125	mA

TIMING REQUIREMENTS ($T_a = -20 \sim 75°C$, $V_{CC} = 5V \pm 10\%$, unless otherwise noted)

DBB Read

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{C(\phi)}$	Cycle time	t_{CY}		2.5		15	μs
$t_{W(R)}$	Read pulse with	t_{RR}	$t_{C(\phi)} = 2.5\mu s$	250			ns
$t_{su(CS-R)}$	Chip-select setup time befor read	t_{AR}		0			ns
$t_{h(R-CS)}$	Chip-select hold time after read	t_{RA}		0			ns

DBB Write

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{W(W)}$	Write pulse width	t_{WW}		250			ns
$t_{su(CS-W)}$ $t_{su(A0-W)}$	$\overline{CS}$, A_0, setup time before write	t_{AW}		0			ns
$t_{h(W-CS)}$ $t_{h(W-A0)}$	$\overline{CS}$, A_0, hold time after wirte	t_{WA}		0			ns
$t_{su(DQ-W)}$	Data setup time before write	t_{DW}		150			ns
$t_{h(W-DQ)}$	Data hold time.after write	t_{WD}		0			ns

Port 2

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{W(PR)}$	PROG pulse width	t_{PP}		1200			ns
$t_{su(PC-PR)}$	Port control setup time before PROG	t_{CP}	$C_L = 80pF$	110			ns
$t_{h(PR-PC)}$	Port control hold time after PROG	t_{PC}	$C_L = 20pF$	100			ns
$t_{su(Q-PR)}$	Output data setup time before PROG	t_{DP}	$C_L = 80pF$	250			ns
$t_{su(D-PR)}$	Input data hold timer before PROG	t_{PR}	$C_L = 80pF$			810	ns
$t_{h(PR-D)}$	Input data hold time after PROG	t_{PF}	$C_L = 20pF$	0		150	ns

DMA

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{su(DACK-R)}$	Data acknowledge time before read	t_{ACC}		0			ns
$t_{h(R-DACK)}$	Data hold time after read	t_{CAC}		0			ns
$t_{su(DACK-W)}$	Data setup time before write	t_{ACC}		0			ns
$t_{h(W-DACK)}$	Data hold time after write	t_{CAC}		0			ns

Note 1 : Input voltage level $V_{IL} = 0.45V$, $V_{IH} = 2.4V$.

SWITCHING CHARACTERISTICS ($T_a = -20 \sim 75°C$, $V_{CC} = 5V \pm 10\%$, unless otherwise noted)

DBB Read

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{PZX(CS-DQ)}$	Data enable time after $\overline{CS}$	t_{AD}	$C_L = 150 pF$			225	ns
$t_{PZX(A0-DQ)}$	Data enable time after address	t_{AD}	$C_L = 150 pF$			225	ns
$t_{PZX(R-DQ)}$	Data enable time after read	t_{RD}	$C_L = 150 pF$			225	ns
$t_{PXZ(R-DQ)}$	Data disable time after read	t_{DF}				100	ns

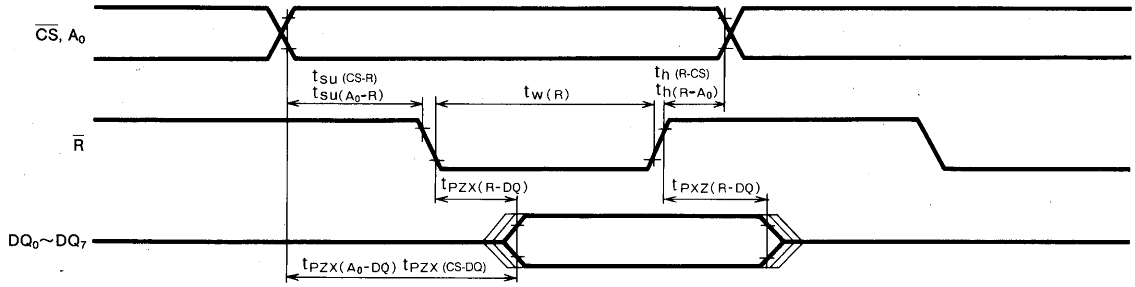
DMA

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{PZX(DACK-DQ)}$	Data enable time after DACK	t_{ACD}	150 pF Load			225	ns
$t_{PHL(R-DRQ)}$	DRQ disable time after read	t_{CRQ}	150 pF Load			200	ns
$t_{PHL(W-DRQ)}$	DRQ disable time after write	t_{CRQ}	150 pF Load			200	ns

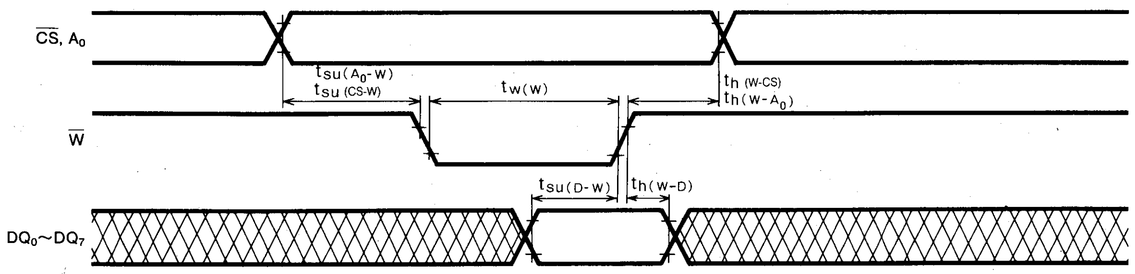
Note 2 : Output voltage discriminating levels, low and high, are 0.8V and 2.0V respectively.

TIMING DIAGRAMS

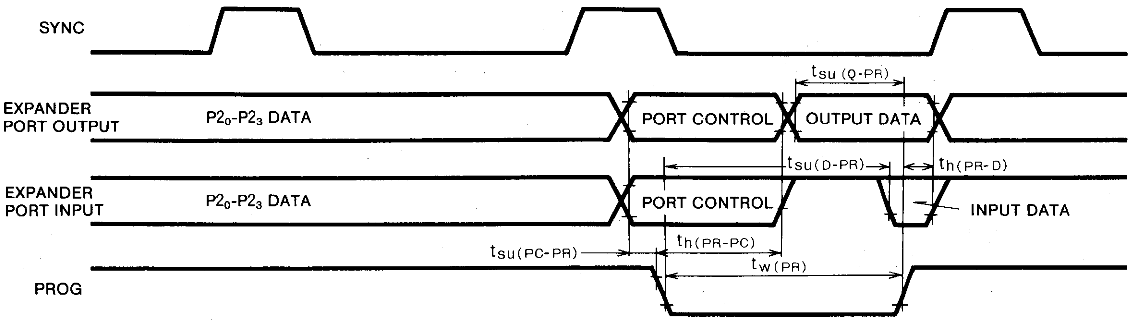
Read



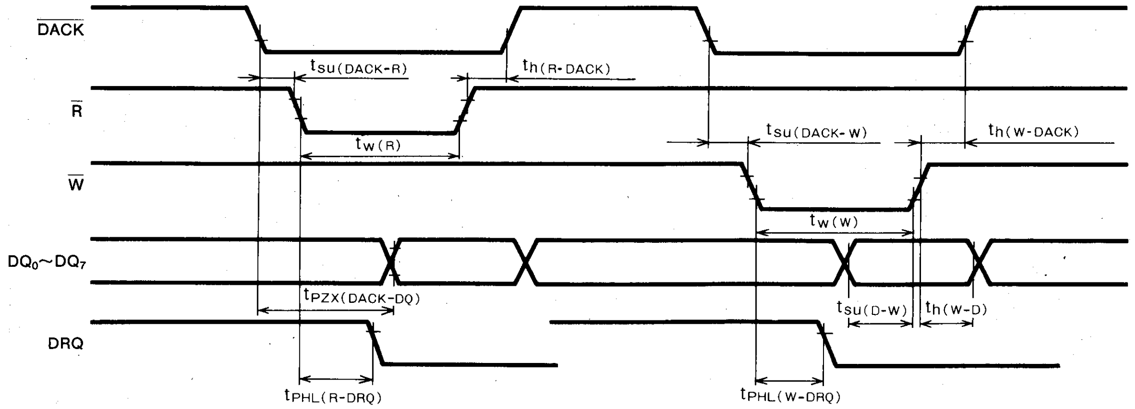
Write



Port 2

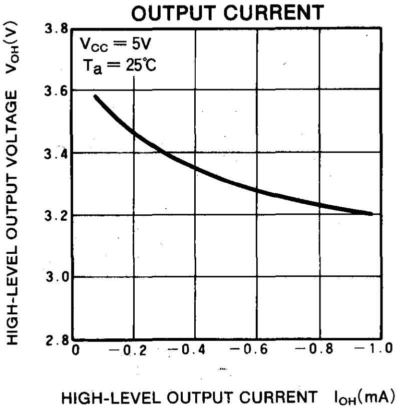


DMA

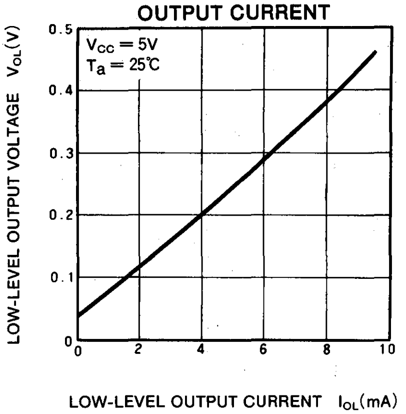


TYPICAL CHARACTERISTICS

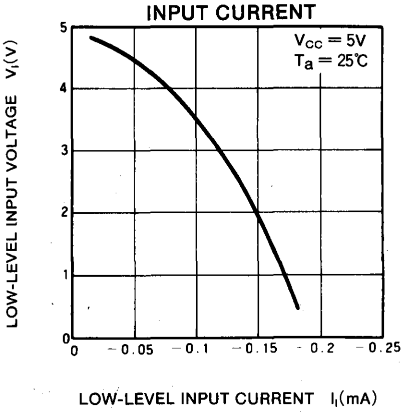
DATA BUS HIGH-LEVEL OUTPUT VOLTAGE VS. HIGH-LEVEL OUTPUT CURRENT



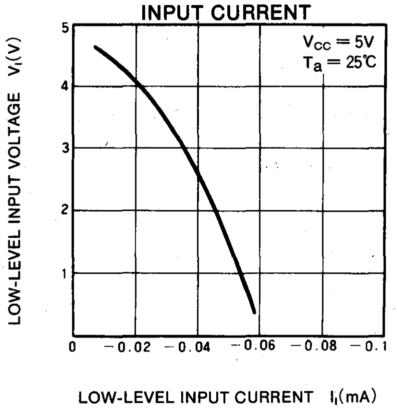
DATA BUS LOW-LEVEL OUTPUT VOLTAGE VS. LOW-LEVEL OUTPUT CURRENT



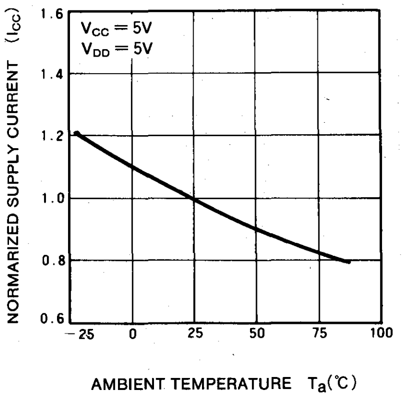
P_1, P_2 LOW-LEVEL INPUT VOLTAGE VS. LOW-LEVEL INPUT CURRENT



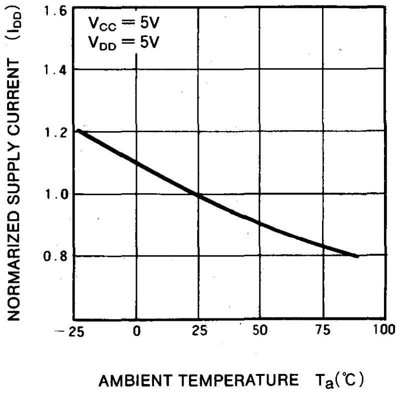
RESET LOW-LEVEL INPUT VOLTAGE VS. LOW-LEVEL INPUT CURRENT



NORMARIZED SUPPLY CURRENT (I_{CC}) VS. AMBIENT TEMPERATURE

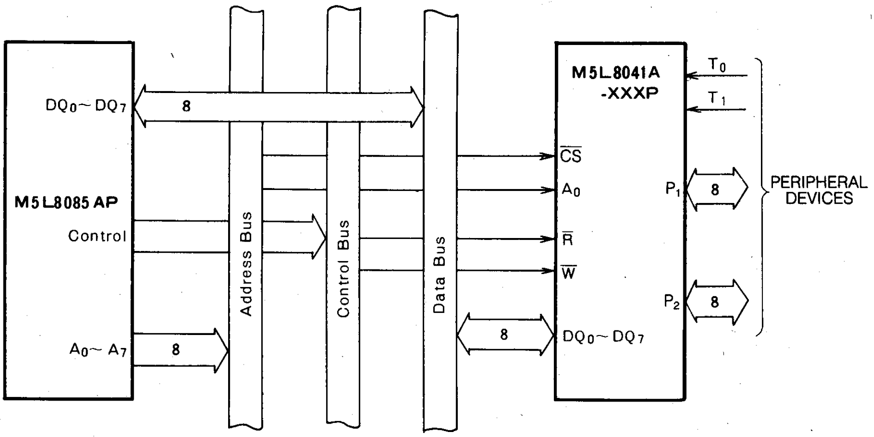


NORMARIZED SUPPLY CURRENT (I_{DD}) VS. AMBIENT TEMPERATURE

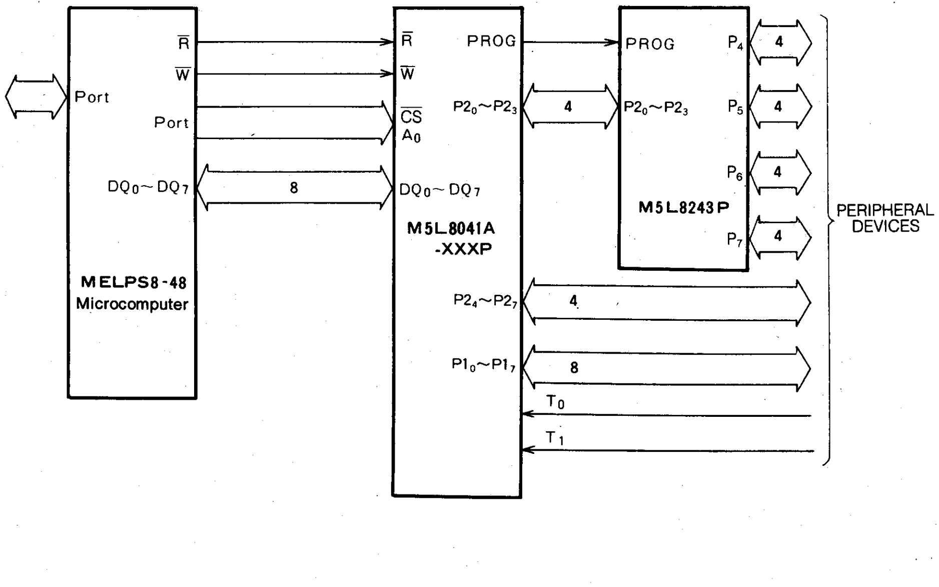


APPLICATION EXAMPLES

(1) Interface with M5L8085AP



(2) Interface with MELPS 8-48 Microcomputer and M5L8243P



M5L8042-XXXP

SLAVE MICROCOMPUTER

DESCRIPTION

The M5L8042-XXXP is a general-purpose programmable interface device designed for use with a variety of 8-bit microcomputer systems. The device is fabricated using n-channel silicon-gate ED-MOS technology.

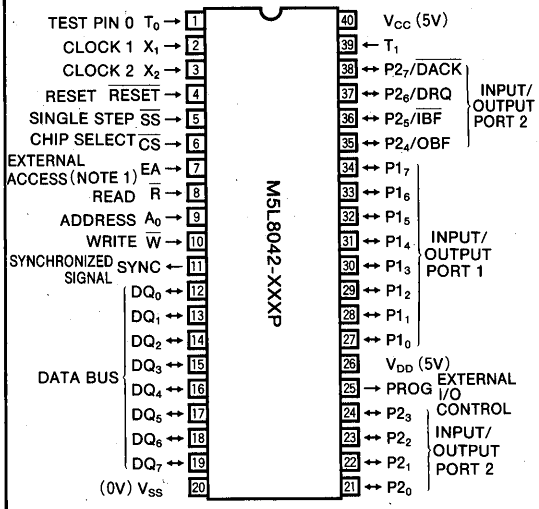
FEATURES

- Mask ROM ···································· 2048-word by 8-bit
- Static RAM ···································· 128-word by 8-bit
- 18 programmable I/O pins
- Asynchronous data register for interface to master processor
- 8-bit CPU, ROM, RAM, I/O, timer, clock and low-power stand-by mode
- Single 5V power supply
- Alternative to custom LSI
- Interchangeable with i 8042

APPLICATION

Alternative to custom LSI for peripheral interfaces

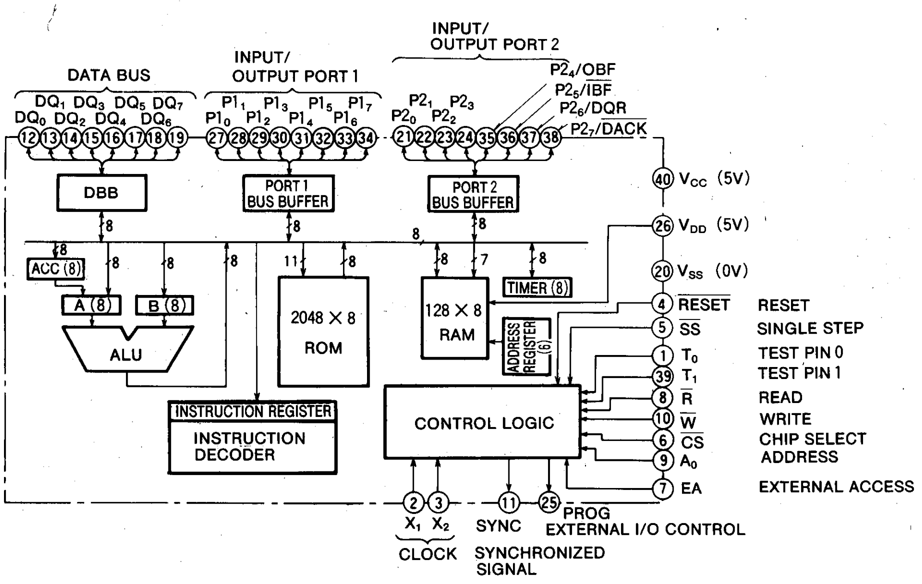
PIN CONFIGURATION (TOP VIEW)



Outline 40P4

Note 1 : Connect to V_{SS} in the operating condition.

BLOCK DIAGRAM



FUNCTION

The M5L8042-XXXP is designed as an ordinary 8-bit CPU peripheral LSI chip and it contains a small stand-alone microcomputer. Although this microcomputer functions independently, when it is used as a peripheral controller, it is called the slave microcomputer in contrast to the master computer. These two devices can transfer the data alternatively through the buffer register between them. The M5L8042-XXXP contains the buffer register to use this LSI as a slave microcomputer and it can be accessed in the same way as other standard peripheral devices. Since the M5L8042-XXXP is a complete microcomputer, it is easy to develop a user-oriented mask-programmed peripheral LSI only by changing the control software.

PIN DESCRIPTION

Pin	Name	Input or output	Function
V_{SS}	Ground	—	Connected to a 0V supply (ground).
V_{CC}	Main power supply	—	Connected to a 5V supply.
V_{DD}	Power supply	—	Connected to a 5V supply.
			Used as a memory hold when V_{CC} is cut off.
T_0	Test pin 0	Input	Provides external control of conditional program jumps (JTO/JNTO instructions).
X_1, X_2	Crystal inputs	Input	An internal clock circuit is provided so that by connecting an RC circuit or crystal to these input pins, the clock frequency can be determined. X_1 and X_2 can also be used to input an external clock signal.
$\overline{RESET}$	Reset	Input	CPU initialization input.
$\overline{SS}$	Single step	Input	Used to halt the execution of a command by the CPU. When used in combination with the SYNC signal, the command execution of the CPU can be halted every instruction to enable single step operation.
$\overline{CS}$	Chip select input	Input	Chip select input for data bus control.
EA	External access	Input	Normally maintained at 0V.
$\overline{R}$	Read enable signal	Input	Serves as the read signal when the master CPU is accepting data on the data bus from the M5L8042-XXXP.
A_0	Address input	Input	An address input used to indicate whether the signal on the data bus is data or a command.
$\overline{W}$	Write enable signal	Input	Serves as the write signal when the master CPU is outputting data from the bus to the M5L8042-XXXP.
SYNC	Sync signal output	Output	Output 1 time for each machine cycle.
$DQ_0 \sim DQ_7$	Data bus	Input/output	Three-state, bidirectional data bus. Data bus is used to interface the M5L8042-XXXP to a master system data bus.
$P2_0 \sim P2_7$	Port 2	Input/output	Quasi-bidirectional port. When used as an input port, FF_{16} must first be output to this port. After resetting, however, when not used afterwards as an output port, this is not necessary.
			$P2_0 \sim P2_3$ are used when the M5L8243P I/O expander is used.
PROG	Program	Output	Serves as the strobe signal when the M5L8243P I/O expander is used.
$P1_0 \sim P1_7$	Port 1	Input/output	Quasi-bidirectional port. When used as an input port, FF_{16} must first be output to this port. After resetting, however, when not used afterwards as an output port, this is not necessary.
T_1	Test pin 1	Input	Provides external control of conditional program jumps (JT1/JNT1 instructions).
			Can serve as the input pin for the event counter (STRT CNT instruction).

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
P_d	Power dissipation	$T_a = 25$℃	1500	mW
T_{opr}	Operating temperature range		0~70	℃
T_{stg}	Storage temperature range		−65~150	℃

RECOMMENDED OPERATING CONDITIONS

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_{IH}	High-level input voltage	2.2			V
V_{IL}	Low-level input voltage			0.8	V
$f(\phi)$	Operating frequency	1		12	MHz

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

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{IL}	Low-level input voltage		−0.5		0.8	V
V_{IH1}	High-level input voltage (all except X_1, X_2, $\overline{RESET}$)		2.2		V_{CC}	V
V_{IH2}	High-level input voltage (X_1, X_2, $\overline{RESET}$)		3.8		V_{CC}	V
V_{OL1}	Low-level output voltage (DQ_0~DQ_7)	$I_{OL} = 2$mA			0.45	V
V_{OL2}	Low-level output voltage ($P1_0$~$P1_7$, $P2_0$~$P2_7$, SYNC)	$I_{OL} = 1.6$ mA			0.45	V
V_{OL3}	Low-level output voltage (PROG)	$I_{OL} = 1$mA			0.45	V
V_{OH1}	High-level output voltage (DQ_0~DQ_7)	$I_{OH} = -400\mu$A	2.4			V
V_{OH2}	High-level output voltage (all other outputs)	$I_{OH} = -50\mu$A	2.4			V
I_I	Input leakage current (T_0, T_1, $\overline{R}$, $\overline{W}$, $\overline{CS}$, A_0, EA)	$V_{SS} \leqq V_I \leqq V_{CC}$	−10		10	μA
I_{OZL}	High-impedance state output leakage current (DQ_0~DQ_7)	$V_{SS} + 0.45 \leqq V_O \leqq V_{CC}$	−10		10	μA
I_{IL1}	Low-level input load current ($P1_0$~$P1_7$, $P2_0$~$P2_7$)	$V_{IL} = 0.8$V	−0.5			mA
I_{IL2}	Low-level input load current ($\overline{RESET}$, $\overline{SS}$)	$V_{IL} = 0.8$V	−0.2			mA
I_{DD}	Supply current from V_{DD}				10	mA
$I_{CC} + I_{DD}$	Total supply current				145	mA

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

DBB Read

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{C(\phi)}$	Cycle time	t_{CY}		1.25		15	μs
$t_{W(R)}$	Read pulse width	t_{RR}	$t_{C(\phi)} = 1.25\,\mu s$	160			ns
$t_{SU(CS-R)}$	Chip select setup time before read	t_{AR}		0			ns
$t_{h(R-CS)}$	Chip select hold time after read	t_{RA}		0			ns

DBB Write

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{W(W)}$	Write pulse width	t_{WW}		160			ns
$t_{SU(CS-W)}$ $t_{SU(A0-W)}$	$\overline{CS}$, A_0, setup time before write	t_{AW}		0			ns
$t_{h(W-CS)}$ $t_{h(W-A0)}$	$\overline{CS}$, A_0, hold time after write	t_{WA}		0			ns
$t_{SU(DQ-W)}$	Data setup time before write	t_{DW}		130			ns
$t_{h(W-DQ)}$	Data hold time after write	t_{WD}		0			ns

Port 2

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{W(PR)}$	PROG pulse width	t_{PP}		700			ns
$t_{SU(PC-PR)}$	Port control setup time before PROG	t_{CP}	$C_L = 80pF$	80			ns
$t_{h(PR-PC)}$	Port control hold time after PROG	t_{PC}	$C_L = 20pF$	60			ns
$t_{SU(Q-PR)}$	Output data setup time before PROG	t_{DP}	$C_L = 80pF$	200			ns
$t_{SU(D-PR)}$	Input data hold time before PROG	t_{PR}	$C_L = 80pF$			650	ns
$t_{h(PR-D)}$	Input data hold time after PROG	t_{PF}	$C_L = 20pF$	0		150	ns

DMA

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{SU(DACK-R)}$	$\overline{DACK}$ setup time before read	t_{ACC}		0			ns
$t_{h(R-DACK)}$	$\overline{DACK}$ hold time after read	t_{CAC}		0			ns
$t_{SU(DACK-W)}$	$\overline{DACK}$ setup time before write	t_{ACC}		0			ns
$t_{h(W-DACK)}$	$\overline{DACK}$ hold time after write	t_{CAC}		0			ns

Note 1 : Input voltage level $V_{IL} = 0.45V$, $V_{IH} = 2.4V$.

SWITCHING CHARACTERISTICS ($T_a = 0 \sim 70°C$, $V_{CC} = 5V \pm 10\%$, unless otherwise noted)

DBB Read

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{PZX(CS-DQ)}$	Data enable time after $\overline{CS}$	t_{AD}	$C_L = 100\,pF$			130	ns
$t_{PZX(A0-DQ)}$	Data enable time after address	t_{AD}	$C_L = 100\,pF$			130	ns
$t_{PZX(R-DQ)}$	Data enable time after read	t_{RD}	$C_L = 100\,pF$			130	ns
$t_{PXZ(R-DQ)}$	Data disable time after read	t_{DF}				85	ns

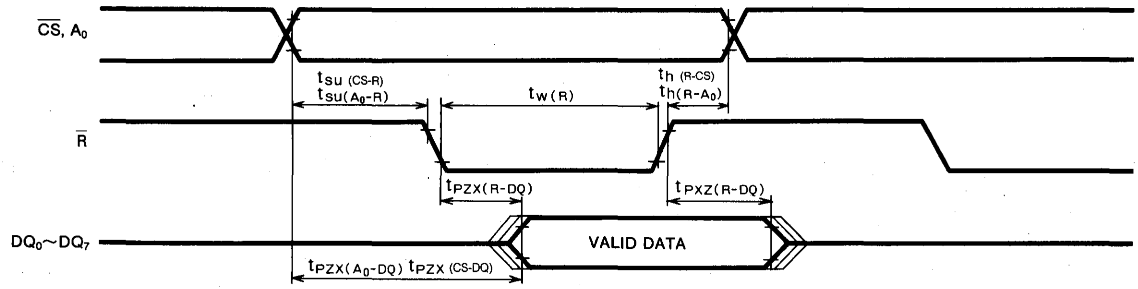
DMA

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{PZX(DACK-DQ)}$	Data enable time after $\overline{DACK}$	t_{ACD}	$C_L = 150\,pF$			130	ns
$t_{PHL(R-DRQ)}$	DRQ disable time after read	t_{CRQ}				90	ns
$t_{PHL(W-DRQ)}$	DRQ disable time after write	t_{CRQ}				90	ns

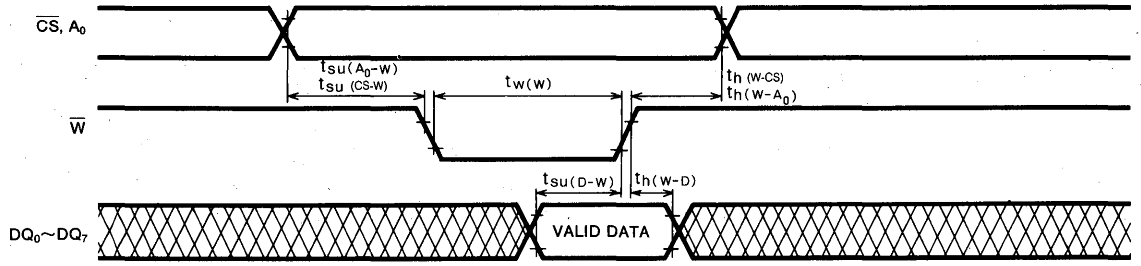
Note 2 : Output voltage discriminating levels, low and high, are 0.8V and 2.0V respectively.

TIMING DIAGRAMS

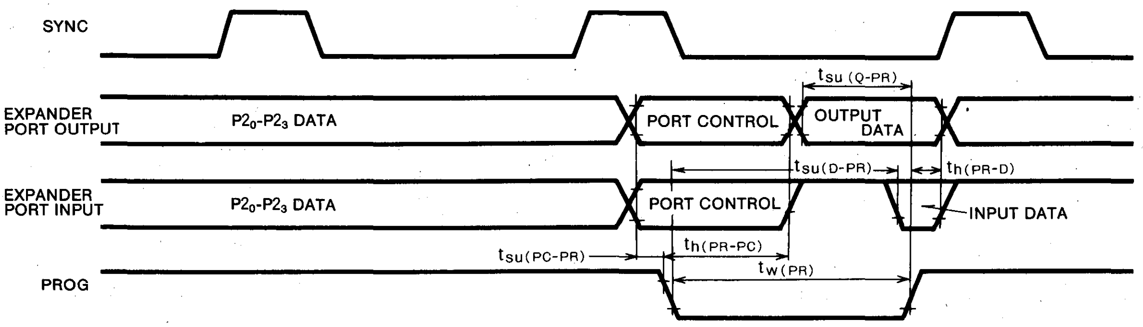
Read



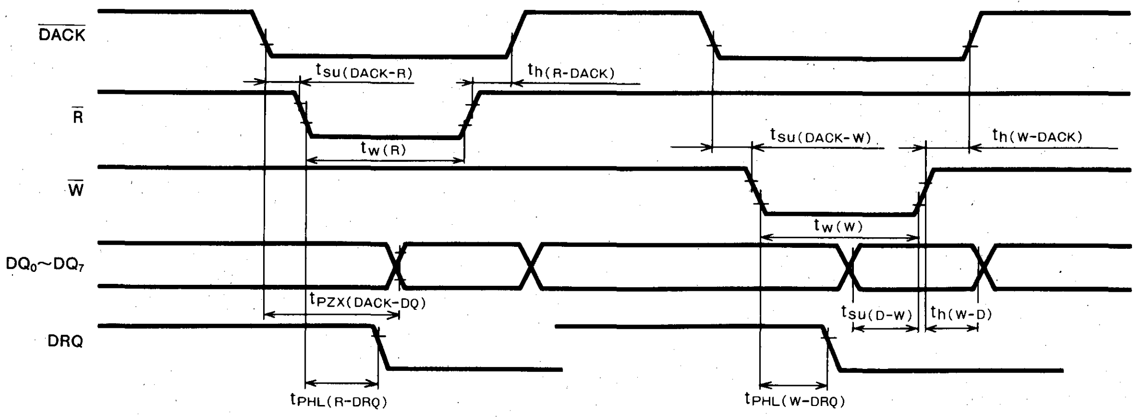
Write



Port 2

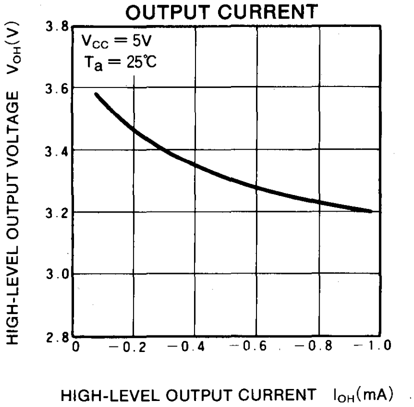


DMA

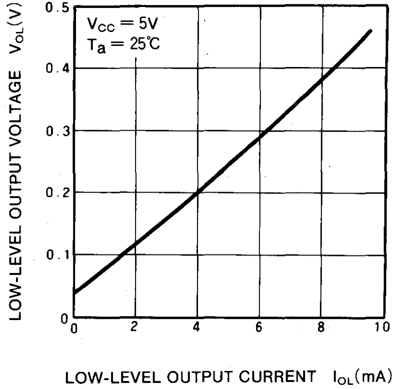


TYPICAL CHARACTERISTICS

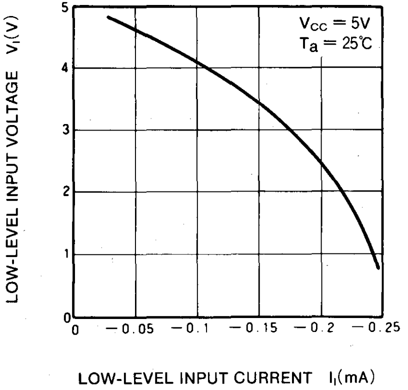
DATA BUS HIGH-LEVEL OUTPUT VOLTAGE VS. HIGH-LEVEL OUTPUT CURRENT



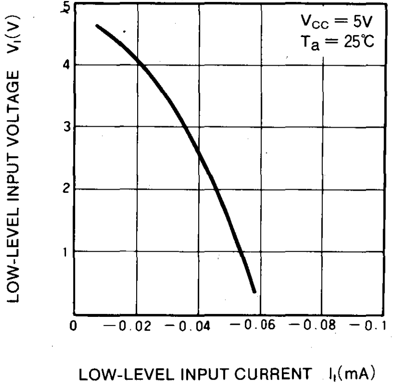
DATA BUS LOW-LEVEL OUTPUT VOLTAGE VS. LOW-LEVEL OUTPUT CURRENT



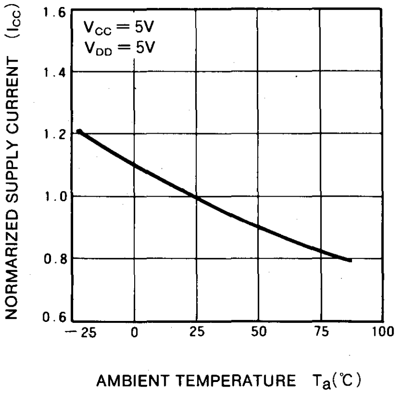
P_1, P_2 LOW-LEVEL INPUT VOLTAGE VS. LOW-LEVEL INPUT CURRENT



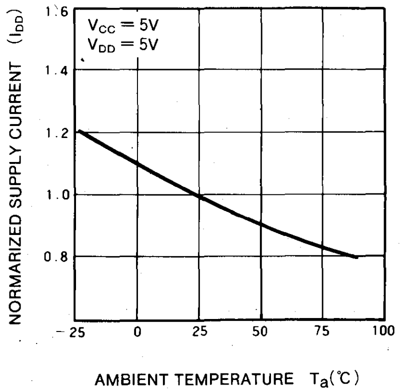
RESET LOW-LEVEL INPUT VOLTAGE VS. LOW-LEVEL INPUT CURRENT



NORMARIZED SUPPLY CURRENT (I_{CC}) VS. AMBIENT TEMPERATURE

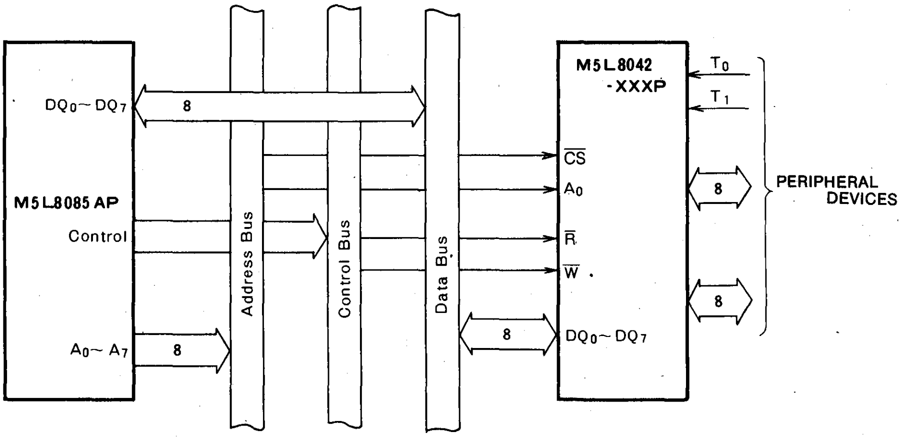


NORMARIZED SUPPLY CURRENT (I_{DD}) VS. AMBIENT TEMPERATURE

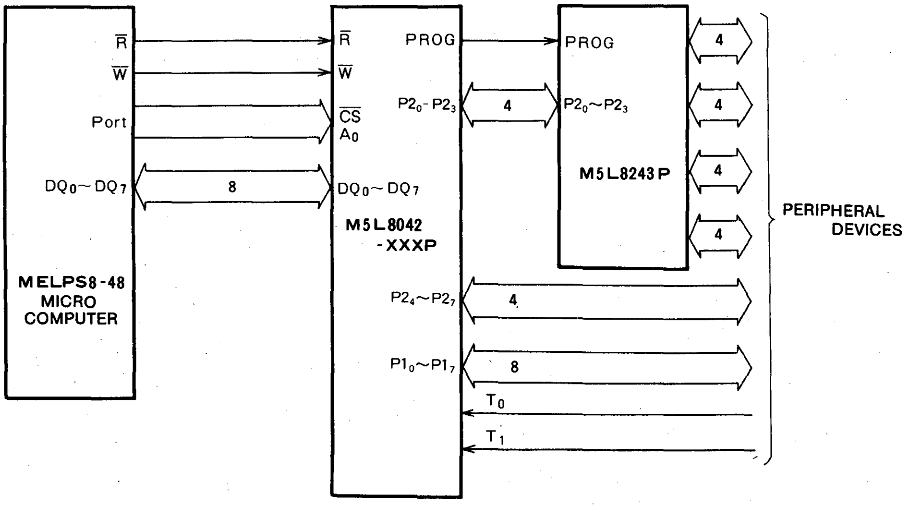


APPLICATION EXAMPLES

(1) Interface with M5L8085AP



(2) Interface with MELPS 8-48 Microcomputer and M5L8243P



LSIs FOR PERIPHERAL CIRCUITS

M50780SP/M50781SP M50782SP/M50783SP

INPUT/OUTPUT EXPANDER

DESCRIPTION

These devices, fabricated using the aluminum gate CMOS process and used for input/output port expansion, are ideal LSIs for connection to the single-chip 4-bit microcomputer series.

The M50780SP and M50782SP are housed in a 40-pin plastic mold DIL package while the M50781SP and M50783SP are housed in a 28–pin plastic mold DIL package.

FEATURES

- Wide operating voltage range . . . . 3 ~ 14V
- Low power dissipation
- Interchangeable with TI's TMS1025C and TMS1024C in terms of pin connections and electrical characteristics (M50780SP and M50781SP)

APPLICATION

I/O expansion for the single-chip microcomputer series

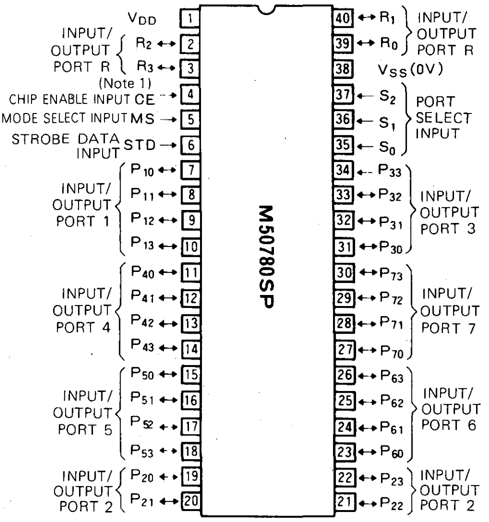
FUNCTION

M50780SP, M50781SP, M50782SP and M50783SP are configured with 4 or 7 groups of input/output ports, 1 group of input/output ports, a port selector circuit and mode control circuit, and operation is possible in the latch or multiplexer mode.

Table 1 Configurations

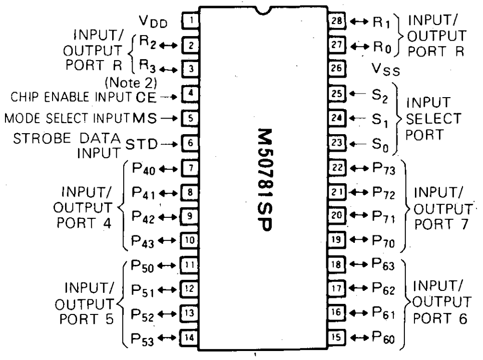
Expander	Outline	CE pin	Requirements for reset	Compatible expanders
M50780SP	40-pin	CE	S_0~S_2 = low STD = (↓)	TMS1025C
M50781SP	28-pin			TMS1024C
M50782SP	40-pin	$\overline{CE}$	S_0~S_2= low STD = high	
M50783SP	28-pin			

PIN CONFIGURATIONS (TOP VIEW)



Outline 40P4B

Note 1: M50782SP has a $\overline{CE}$ pin.



Outline 28P4B

Note 2: M50783SP has a $\overline{CE}$ pin.

Table 2 Pin Description

Symbol	Name	Function
MS	Mode select input	Latch mode at high; multiplexer mode at low.
STD	Strobe input	This input is valid in latch mode and data latched by the fall (↓) are output to output port. Input data are read into latch at high level. At low level latch data are output.
CE ($\overline{CE}$)	Chip enable input	This input is valid in multiplexer mode.
R_0~R_3	Input/output port	4-bit bidirectional input/output ports. Input in latch mode; output in multiplexer mode.
P_{10}~P_{13} P_{20}~P_{23} P_{30}~P_{33} P_{40}~P_{43} P_{50}~P_{53} P_{60}~P_{63} P_{70}~P_{73}	Input/output ports 1~7	4-bit bidirectional input/output ports. Output in latch mode; input in multiplexer mode.

M50780SP、M50781SP

Table 3 Latch Modes

Input					Latch	Output
MS	CE	STD	S_0~S_2	R_0~R_3		Q (P_{n0}, P_{n1}, P_{n2}, P_{n3})
H	X	↓	n	H	H	H (P_{n0}~P_{n3})
H	X	↓	n	L	L	L (P_{n0}~P_{n3})
H	X	L	0, n	X	Q_0	Q_0 (All ports)
H	X	H	0, n	X	Q_0	Q_0(All ports)
H	X	↓	0	X	L	L (All ports)
L	L	X	0, n	X	Q_0	Z (All ports)

Table 4 Multiplexer Modes

Input					Output
MS	CE	STD	S_0~S_2	P_{n0}~P_{n3}	Q (R_0, R_1, R_2, R_3)
L	L	X	0, n	X	Z
L	H	X	n	H	H
L	H	X	n	L	L
L	X	X	0	X	Z

M50782SP、M50783SP

Table 5 Latch Modes

Input					Latch	Output
MS	$\overline{CE}$	STD	S_0~S_2	R_0~R_3		Q (P_{n0}, P_{n1}, P_{n2}, P_{n3})
H	X	H	n	H	— *	H (P_{n0}~P_{n3})
H	X	H	n	L	— *	L (P_{n0}~P_{n3})
H	X	↓	n	H	H	H (P_{n0}~P_{n3})
H	X	↓	n	L	L	L (P_{n0}~P_{n3})
H	X	L	0, n	X	Q_0	Q_0 (All ports)
H	X	H	0	X	— *	L (All ports)
L	H	L	n	X	Q_0	Z (All ports)

* Not latched

Table 6 Multiplexer Modes

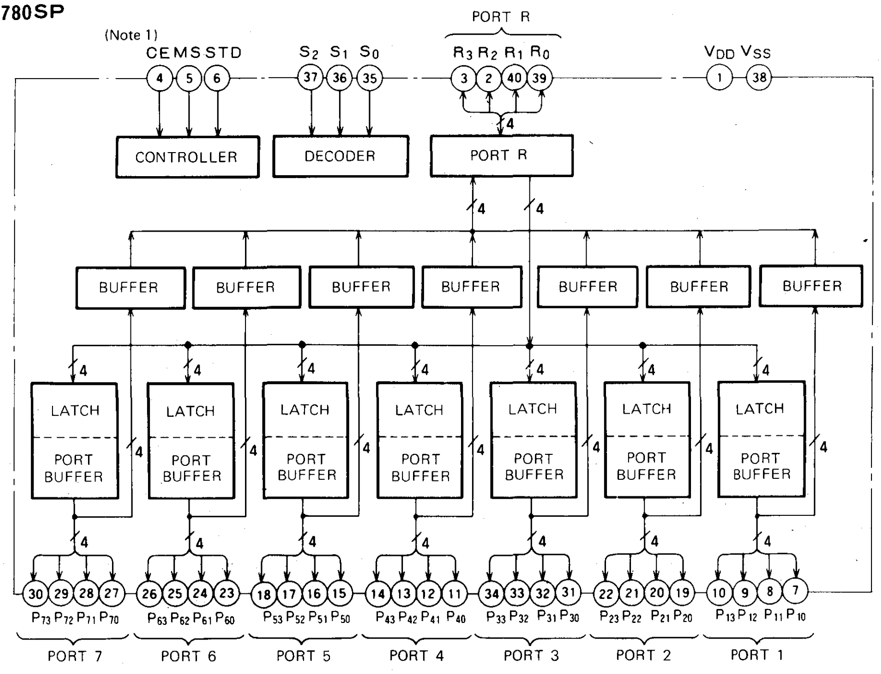
Input					Output
MS	$\overline{CE}$	STD	S_0~S_2	P_{n0}~P_{n3}	Q (R_0, R_1, R_2, R_3)
L	H	X	0, n	X	Z
L	L	X	n	H	H
L	L	X	n	L	L
L	X	X	0	X	Z

X : High or low

Z : High-impedance state

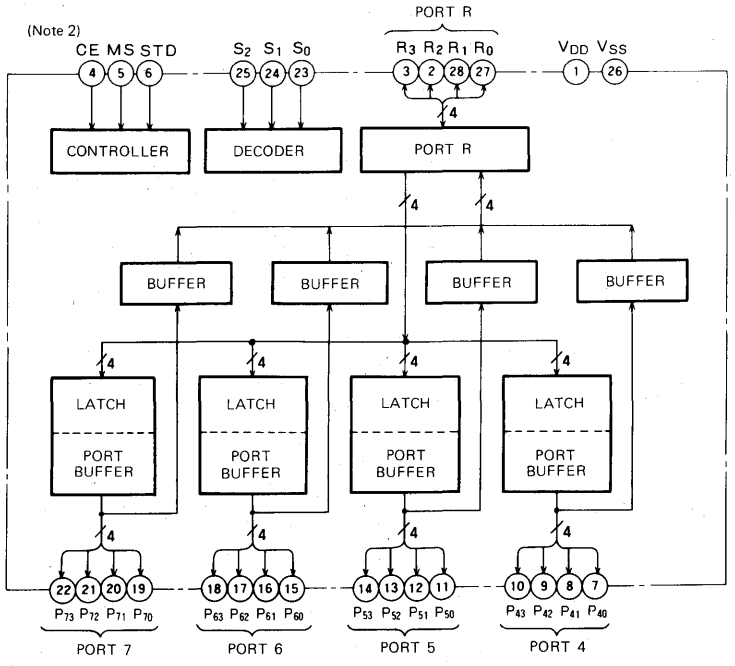
BLOCK DIAGRAMS

M50780SP


PORT R
(Note 1)
CE MS STD
4 5 6
S2 S1 S0
37 36 35
R3 R2 R1 R0
3 2 40 39
VDD VSS
1 38
CONTROLLER
DECODER
PORT R
BUFFER
LATCH
PORT BUFFER
4
30 29 28 27 P73 P72 P71 P70 PORT 7
26 25 24 23 P63 P62 P61 P60 PORT 6
18 17 16 15 P53 P52 P51 P50 PORT 5
14 13 12 11 P43 P42 P41 P40 PORT 4
34 33 32 31 P33 P32 P31 P30 PORT 3
22 21 20 19 P23 P22 P21 P20 PORT 2
10 9 8 7 P13 P12 P11 P10 PORT 1

Note 1: M50782SP has a $\overline{CE}$ pin.

M50781SP


PORT R
(Note 2)
CE MS STD
4 5 6
S2 S1 S0
25 24 23
R3 R2 R1 R0
3 2 28 27
VDD VSS
1 26
CONTROLLER
DECODER
PORT R
BUFFER
LATCH
PORT BUFFER
4
22 21 20 19 P73 P72 P71 P70 PORT 7
18 17 16 15 P63 P62 P61 P60 PORT 6
14 13 12 11 P53 P52 P51 P50 PORT 5
10 9 8 7 P43 P42 P41 P40 PORT 4

Note 2: M50783SP has a $\overline{CE}$ pin.

Table 7 Function Table

				M50780SP, M50782SP						
							M50781SP, M50783SP			
n	S_2	S_1	S_0	P_{10}~P_{13}	P_{20}~P_{23}	P_{30}~P_{33}	P_{40}~P_{43}	P_{50}~P_{53}	P_{60}~P_{63}	P_{70}~P_{73}
1	L	L	H	○						
2	L	H	L		○					
3	L	H	H			○				
4	H	L	L				○			
5	H	L	H					○		
6	H	H	L						○	
7	H	H	H							○

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{DD}	Supply voltage	With respect to V_{SS}	−0.3~15	V
V_I	Input voltage		V_{SS}−0.3~V_{DD}+0.3	V
V_O	Output voltage		V_{SS}−0.3~V_{DD}+0.3	V
Pd	Maximum power dissipation	Ta=25°C	600	mW
Topr	Operating free-air temperature range		−10~70	°C
Tstg	Storage temperature range		−40~125	°C

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{DD}	Supply voltage	3		14	V
V_I	Input voltage	0		V_{DD}	V
V_{IH}	High-level input voltage	V_{DD}×0.7		V_{DD}	V
V_{IL}	Low-level input voltage	0		V_{DD}×0.3	V

ELECTRICAL CHARACTERISTICS (Ta=25°C, V_{DD}=9V, unless otherwise noted)

Symbol	Parameter		Test conditions	Limits			Unit
				Min	Typ	Max	
V_{OH}	High-level output voltage	Port 1 ~ port 7	V_{DD}= 5 V, I_{OH}=−2mA	2.5			V
			V_{DD}= 9 V, I_{OH}=−4mA	6.5			V
			V_{DD}=12V, I_{OH}=−5.5mA	9.5			V
		Port R	V_{DD}= 5 V, I_{OH}=−200μA	4.6			V
			V_{DD}= 9 V, I_{OH}=−350μA	8.6			V
			V_{DD}=12V, I_{OH}=−450μA	11.6			V
V_{OL1}	Low-level output voltage	Port 1 ~ port 7	V_{DD}= 5 V, I_{OL}=1mA			0.4	V
			V_{DD}= 9 V, I_{OL}=1.4mA			0.4	V
			V_{DD}=12V, I_{OL}=1.7mA			0.4	V
		Port R	V_{DD}= 5 V, I_{OL}=250μA			0.4	V
			V_{DD}= 9 V, I_{OL}=450μA			0.4	V
			V_{DD}=12V, I_{OL}=550μA			0.4	V
V_{OL2}	Low-level output voltage	Port 1 ~ port 7	V_{DD}=5V, I_{OL}=4mA			1.8	V
			V_{DD}=9V, I_{OL}=12mA			2.5	V
			V_{DD}=12V, I_{OL}=17mA			3.9	V
I_I	Input current	Port 1 ~ port 7	V_I=0~V_{DD}			10	μA
		Port R	V_I=0~V_{DD}			1	μA
I_{DD}	Supply current		Output pins open			50	μA

TIMING REQUIREMENTS

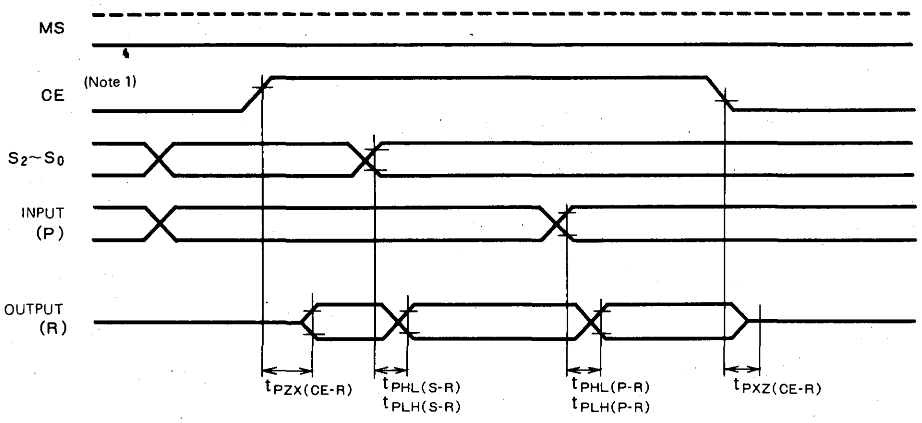
Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
$t_{su(R-STD)}$	Data set-up time for STD input, port R input	V_{DD}= 5 V	0.2			μs
		V_{DD}= 9 V	0.15			
		V_{DD}=12V	0.1			
$t_{su(S-STD)}$	Data set-up time for STD input, S_0~S_2 inputs	V_{DD}= 5 V	1			μs
		V_{DD}= 9 V	0.5			
		V_{DD}=12V	0.2			
t_w	STD input pulse width	V_{DD}= 5 V	0.5			μs
		V_{DD}= 9 V	0.4			
		V_{DD}=12V	0.3			
$t_{h(R-STD)}$	Data hold time for STD input, port R input	V_{DD}= 5 V	0.4			μs
		V_{DD}= 9 V	0.3			
		V_{DD}=12V	0.2			
$t_{h(S-STD)}$	Data hold time for STD input, S_0~S_2 inputs	V_{DD}= 5 V	1			μs
		V_{DD}= 9 V	0.7			
		V_{DD}=12V	0.5			

SWITCHING CHARACTERISTICS

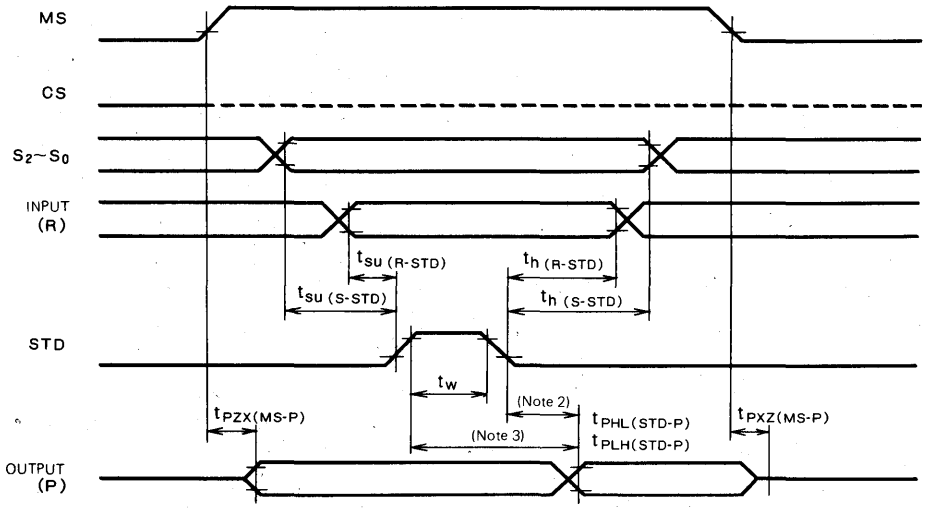
Symbol	Parameter	Test conditions		Limits			Unit
				Min	Typ	Max	
$t_{PZX(CE-P)}$	Valid output delay time for CE input, R output	R_L= 10kΩ C_L=50pF	V_{DD}= 5 V			0.6	μs
			V_{DD}= 9 V			0.5	
			V_{DD}=12V			0.4	
$t_{PZX(MS-P)}$	Valid output delay time for MS input, P_{10}~P_{73} outputs		V_{DD}= 5 V			0.8	μs
			V_{DD}= 9 V			0.6	
			V_{DD}=12V			0.5	
$t_{PXZ(CE-R)}$	Output floating delay time for CE input, R output		V_{DD}= 5 V			0.6	μs
			V_{DD}= 9 V			0.5	
			V_{DD}=12V			0.4	
$t_{PXZ(MS-P)}$	Output floating delay time for MS input, P_{10}~P_{73} outputs		V_{DD}= 5 V			0.8	μs
			V_{DD}= 9 V			0.6	
			V_{DD}=12V			0.5	
$t_{PHL(S-R)}$	Data output high-to-low delay time; S input, R output	R_L=200kΩ C_L=50pF	V_{DD}= 5 V			3	μs
			V_{DD}= 9 V			1	
			V_{DD}=12V			0.7	
$t_{PHL(P-R)}$	Data output high-to-low delay time; P_{10}~P_{73} inputs, R output		V_{DD}= 5 V			1.8	μs
			V_{DD}= 9 V			0.7	
			V_{DD}=12V			0.5	
$t_{PHL(SFD-P)}$	Data output high-to-low delay time; STD input, P output		V_{DD}= 5 V			1.2	μs
			V_{DD}= 9 V			0.5	
			V_{DD}=12V			0.4	
$t_{PLH(S-R)}$	Data output low-to-high delay time; S input, R output		V_{DD}= 5 V			3	μs
			V_{DD}= 9 V			1	
			V_{DD}=12V			0.7	
$t_{PLH(P-R)}$	Data output low-to-high delay time; P_{10}~P_{73} inputs, R output		V_{DD}= 5 V			1.8	μs
			V_{DD}= 9 V			0.7	
			V_{DD}=12V			0.5	
$t_{PLH(STD-P)}$	Data output low-to-high delay time; STD input, P_{10}~P_{73} outputs		V_{DD}= 5 V			1.2	μs
			V_{DD}= 9 V			0.5	
			V_{DD}=12V			0.4	

TIMING DIAGRAM

Multiplex Mode



Latch Mode



Note 1: M50782SP and M50783SP have a $\overline{CE}$ pin.
2: For M50780SP and M50781SP
3: For M50782SP and M50783SP

M50784SP

INPUT EXPANDER

DESCRIPTION

This device, fabricated using the aluminum gate CMOS process and used for input port expansion, is an ideal LSI for connection to the single-chip 4-bit microcomputer series.

It is housed in a 28-pin plastic mold DIL package.

FEATURES

- Wide operating voltage range .... (4 ~ 14V)
- Low power dissipation

APPLICATION

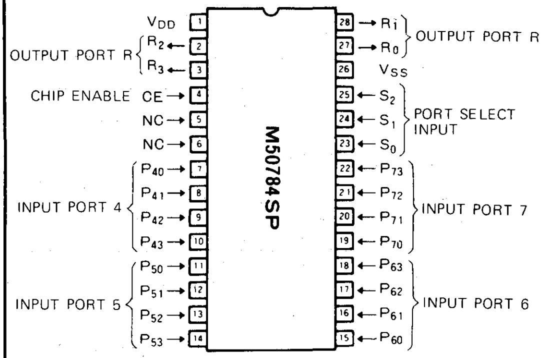
Input port expansion in the single-chip microcomputer series

FUNCTION

M50784SP comprises 4 groups of input ports, 1 group of output ports and a port selector circuit.

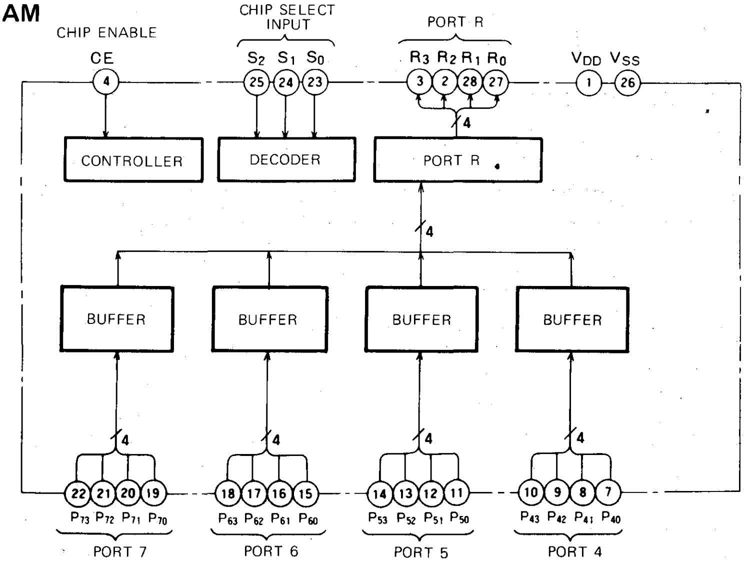
It is particularly attractive for implementing a multiplexer.

PIN CONFIGURATION (TOP VIEW)



Outline 28P4B

BLOCK DIAGRAM



PIN DESCRIPTION

Symbol	Name	Function
CE	Chip enable input	When low, output transistors will be "off" state.
$R_0 \sim R_3$	Output port	4-bit output ports. P-channel open drain outputs.
$P_{40} \sim P_{43}$ $P_{50} \sim P_{53}$ $P_{60} \sim P_{63}$ $P_{70} \sim P_{73}$	Input ports 4 ~ 7	4-bit input ports

Function table (1)

Input			Output
CE	$S_0 \sim S_2$	$Pn_0 \sim Pn_3$	Q (R_0, R_1, R_2, R_3)
L	0, n	X	(L)
H	n	H	H
H	n	L	(L)
X	0	X	(L)

(L) : Transistor "off" state
X : High or low

Function table (2)

n	S_2	S_1	S_0	$P_{40} \sim P_{43}$	$P_{50} \sim P_{53}$	$P_{60} \sim P_{63}$	$P_{70} \sim P_{73}$
4	H	L	L	○			
5	H	L	H		○		
6	H	H	L			○	
7	H	H	H				○

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{DD}	Supply voltage	With respect to V_{SS}	$-0.3 \sim 15$	V
V_I	Input voltage		$V_{SS}-0.3 \sim V_{DD}+0.3$	V
V_O	Output voltage		$V_{SS}-0.3 \sim V_{DD}+0.3$	V
Pd	Maximum power dissipation	Ta = 25°C	600	mW
Topr	Operating free-air temperature range		$-10 \sim 70$	°C
Tstg	Storage temperature range		$-40 \sim 125$	°C

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{DD}	Supply voltage	4		14	V
V_I	Input voltage	0		V_{DD}	V
V_{IH}	High-level input voltage	$V_{DD} \times 0.7$		V_{DD}	V
V_{IL}	Low-level input voltage	0		$V_{DD} \times 0.3$	V

ELECTRICAL CHARACTERISTICS (Ta = 25°C, V_{DD} = 9V, unless otherwise noted)

Symbol	Parameter		Test condition	Limits			Unit
				Min	Typ	Max	
I_{OH}	High-level output current	Port R	$V_{DD}=5V$, $V_{OH}=3.5V$	1.5			mA
			$V_{DD}=9V$, $V_{OH}=7.5V$	2.5			mA
			$V_{DD}=12V$, $V_{OH}=10.5V$	3.5			mA
I_I	Input current	Port 4 ~ Port 7	$V_I=0 \sim V_{DD}$			1	μA
I_{DD}	Supply current		Output pins open			50	μA

M50786SP

INPUT/OUTPUT EXPANDER

DESCRIPTION

The M50786SP is an input/output expander fabricated using aluminum-gate CMOS technology. It is designed especially for connection with the single-chip 4-bit microcomputer series, and it comes in a 40-pin plastic molded DIL package.

FEATURES

- Wide operating voltage range .... 4~14V
- Low power dissipation

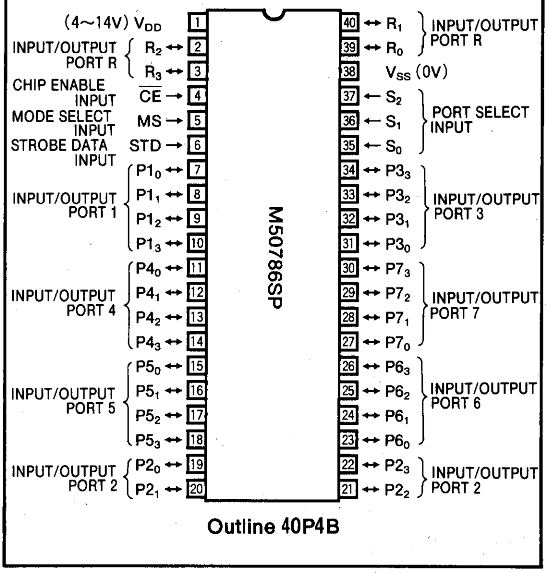
APPLICATION

I/O expansion for the single-chip 4-bit microcomputer series

FUNCTION

The M50786SP is composed of seven groups of I/O ports, one I/O port group, a port selector circuit and a mode control circuit. It can be used in the latch or multiplexer mode. P-channel open-drain output buffers are featured.

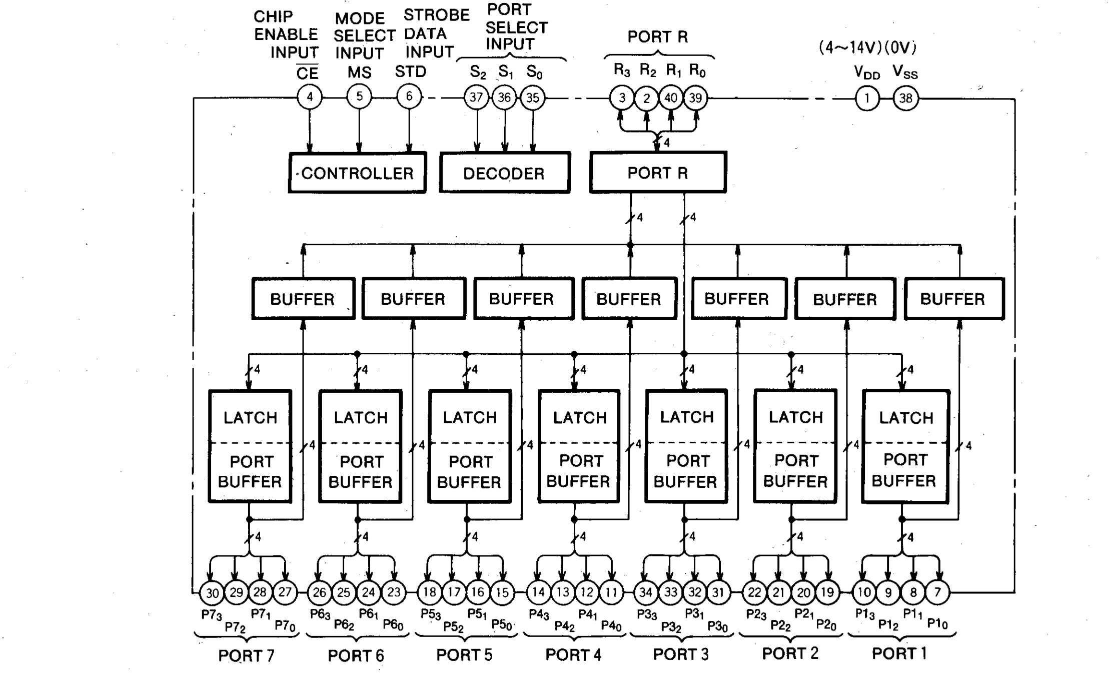
PIN CONFIGURATION (TOP VIEW)



(4~14V) V_{DD} 1
INPUT/OUTPUT PORT R R_2 2
R_3 3
CHIP ENABLE INPUT $\overline{CE}$ 4
MODE SELECT INPUT MS 5
STROBE DATA INPUT STD 6
INPUT/OUTPUT PORT 1 $P1_0$ 7
$P1_1$ 8
$P1_2$ 9
$P1_3$ 10
INPUT/OUTPUT PORT 4 $P4_0$ 11
$P4_1$ 12
$P4_2$ 13
$P4_3$ 14
INPUT/OUTPUT PORT 5 $P5_0$ 15
$P5_1$ 16
$P5_2$ 17
$P5_3$ 18
INPUT/OUTPUT PORT 2 $P2_0$ 19
$P2_1$ 20
M50786SP
40 R_1 INPUT/OUTPUT PORT R
39 R_0
38 V_{SS} (0V)
37 S_2 PORT SELECT INPUT
36 S_1
35 S_0
34 $P3_3$ INPUT/OUTPUT PORT 3
33 $P3_2$
32 $P3_1$
31 $P3_0$
30 $P7_3$ INPUT/OUTPUT PORT 7
29 $P7_2$
28 $P7_1$
27 $P7_0$
26 $P6_3$ INPUT/OUTPUT PORT 6
25 $P6_2$
24 $P6_1$
23 $P6_0$
22 $P2_3$ INPUT/OUTPUT PORT 2
21 $P2_2$

Outline 40P4B

BLOCK DIAGRAM



CHIP ENABLE INPUT $\overline{CE}$
MODE SELECT INPUT MS
STROBE DATA INPUT STD
PORT SELECT INPUT S_2 S_1 S_0
PORT R R_3 R_2 R_1 R_0
(4~14V)(0V) V_{DD} V_{SS}
4 5 6 37 36 35 3 2 40 39 1 38
CONTROLLER
DECODER
PORT R
BUFFER
LATCH
PORT BUFFER
4
30 29 28 27 26 25 24 23 18 17 16 15 14 13 12 11 34 33 32 31 22 21 20 19 10 9 8 7
$P7_3$ $P7_2$ $P7_1$ $P7_0$ PORT 7
$P6_3$ $P6_2$ $P6_1$ $P6_0$ PORT 6
$P5_3$ $P5_2$ $P5_1$ $P5_0$ PORT 5
$P4_3$ $P4_2$ $P4_1$ $P4_0$ PORT 4
$P3_3$ $P3_2$ $P3_1$ $P3_0$ PORT 3
$P2_3$ $P2_2$ $P2_1$ $P2_0$ PORT 2
$P1_3$ $P1_2$ $P1_1$ $P1_0$ PORT 1

PIN DESCRIPTION

Symbol	Name	Function
MS	Mode select input	Latch mode when high; multiplexer mode when low.
STD	Strobe input	Valid in latch mode; data latched by fall (↓) are supplied to output port. When high, input data are read into latch; when low, latch data are output.
$\overline{CE}$	Chip enable input	Input is valid in multiplexer mode.
R_0~R_3	Input/output port	4-bit bidirectional input/output port. Functions as input in latch mode and output in multiplexer mode.
$P1_0$~$P1_3$ $P2_0$~$P2_3$ $P3_0$~$P3_3$ $P4_0$~$P4_3$ $P5_0$~$P5_3$ $P6_0$~$P6_3$ $P7_0$~$P7_3$	Input/output port 1~7	4-bit bidirectional input/output ports that function as output in latch mode and input in multiplexer mode.

LATCH MODE

Input					latch	Output
MS	$\overline{CE}$	STD	S_0~S_2	R_0~R_3		Q (Pn_0, Pn_1, Pn_2, Pn_3)
H	X	H	n	H	—*	H (Pn_0~Pn_3)
H	X	H	n	L	—*	(L) (Pn_0~Pn_3)
H	X	↓	n	H	H	H (Pn_0~Pn_3)
H	X	↓	n	L	L	(L) (Pn_0~Pn_3)
H	X	L	0, n	X	Q_0	Q_0 (all ports)
H	X	H	0	X	—*	(L) (all ports)
L	H	L	n	X	Q_0	(L) (all ports)

*Not latched

MULTIPLEXER MODE

Input					Output
MS	$\overline{CE}$	STD	S_0~S_2	Pn_0~Pn_3	Q (R_0, R_1, R_2, R_3)
L	H	X	0, n	X	Z
L	L	X	n	H	H
L	L	X	n	L	L
L	X	X	0	X	Z

X : Irrelevant
Z : High impedance
(L): Output transistor off-state

FUNCTION TABLE

n	S_2	S_1	S_0	$P1_0$~$P1_3$	$P2_0$~$P2_3$	$P3_0$~$P3_3$	$P4_0$~$P4_3$	$P5_0$~$P5_3$	$P6_0$~$P6_3$	$P7_0$~$P7_3$
1	L	L	H	○						
2	L	H	L		○					
3	L	H	H			○				
4	H	L	L				○			
5	H	L	H					○		
6	H	H	L						○	
7	H	H	H							○

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{DD}	Supply voltage	With respect to V_{SS}	−0.3∼15	V
V_I	Input voltage		V_{SS} −0.3∼V_{DD} +0.3	V
V_O	Output voltage		V_{SS} −0.3∼V_{DD} +0.3	V
P_d	Power dissipation	T_a = 25℃	600	mW
T_{opr}	Operating temperature		−10∼70	℃
T_{stg}	Storage temperature		−40∼125	℃

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Limits			Unit
		Min	Nom	Max	
V_{DD}	Supply voltage	4		14	V
V_I	Input voltage	0		V_{DD}	V
V_{IH}	High-level input voltage	V_{DD}×0.7		V_{DD}	V
V_{IL}	Low-level input voltage	0		V_{DD}×0.3	V

ELECTRICAL CHARACTERISTICS (V_{DD} = 9V, T_a = 25℃, unless otherwise noted)

Symbol	Parameter		Test conditions	Limits			Unit
				Min	Typ	Max	
V_{OH}	High-level output voltage	Ports 1∼7	V_{DD} = 5V, I_{OH} = −1mA	2.5			V
			V_{DD} = 9V, I_{OH} = −3mA	6.5			V
			V_{DD} = 12V, I_{OH} = −5.5mA	9.5			V
		Port R	V_{DD} = 5V, I_{OH} = −100μA	4.6			V
			V_{DD} = 9V, I_{OH} = −250μA	8.6			V
			V_{DD} = 12V, I_{OH} = −450μA	11.6			V
V_{OL}	Low-level output voltage	Port R	V_{DD} = 5V, I_{OL} = 130μA			0.4	V
			V_{DD} = 9V, I_{OL} = 350μA			0.4	V
			V_{DD} = 12V, I_{OL} = 550μA			0.4	V
I_I	Input current	Ports 1∼7	V_I = 0∼V_{DD}			10	μA
		Except ports 1∼7	V_I = 0∼V_{DD}			1	μA
I_{DD}	Supply current		Output pins open			50	μA

TIMING REQUIREMENTS

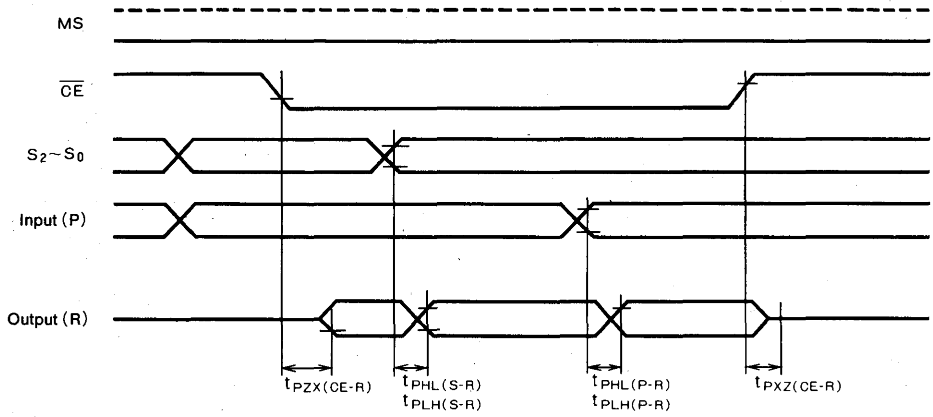
Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
$t_{su\ (R\text{-}STD)}$	Data setup time for STD input, port R input	$V_{DD} = 5V$	0.4			μs
		$V_{DD} = 9V$	0.4			
		$V_{DD} = 12V$	0.4			
$t_{su\ (S\text{-}STD)}$	Data setup time for STD input, inputs $S_0 \sim S_2$	$V_{DD} = 5V$	1			μs
		$V_{DD} = 9V$	0.7			
		$V_{DD} = 12V$	0.5			
t_W	STD input pulse width	$V_{DD} = 5V$	1.5			μs
		$V_{DD} = 9V$	0.5			
		$V_{DD} = 12V$	0.3			
$t_{h\ (R\text{-}STD)}$	Data hold time for STD input, port R input	$V_{DD} = 5V$	0.5			μs
		$V_{DD} = 9V$	0.5			
		$V_{DD} = 12V$	0.5			
$t_{h\ (S\text{-}STD)}$	Data hold time for STD input, port S input	$V_{DD} = 5V$	1			μs
		$V_{DD} = 9V$	0.7			
		$V_{DD} = 12V$	0.5			

SWITCHING CHARACTERISTICS

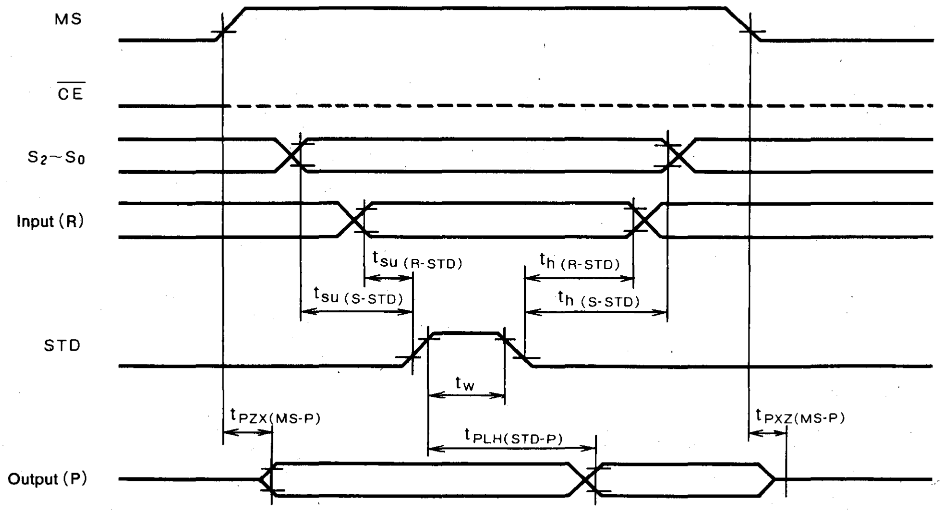
Symbol	Parameter	Test conditions		Limits			Unit
				Min	Typ	Max	
$t_{PZX\ (CE\text{-}R)}$	Valid output delay time for CE input, R output	$R_L = 10k\Omega$ $C_L = 50pF$	$V_{DD} = 5V$			1.5	μs
			$V_{DD} = 9V$			1	
			$V_{DD} = 12V$			0.8	
$t_{PZX\ (MS\text{-}P)}$	Valid output delay time for MS input, $P1_0 \sim P7_3$ output	$R_L = 2k\Omega$ $C_L = 50pF$	$V_{DD} = 5V$			2.4	μs
			$V_{DD} = 9V$			1.8	
			$V_{DD} = 12V$			1.2	
$t_{PXZ\ (CE\text{-}R)}$	Output floating delay time for CE input, R output	$R_L = 10k\Omega$ $C_L = 50pF$	$V_{DD} = 5V$			1.5	μs
			$V_{DD} = 9V$			1	
			$V_{DD} = 12V$			0.8	
$t_{PXZ\ (MS\text{-}P)}$	Output floating delay time for MS input, $P1_0 \sim P7_3$ output	$R_L = 2k\Omega$ $C_L = 50pF$	$V_{DD} = 5V$			2.4	μs
			$V_{DD} = 9V$			1.8	
			$V_{DD} = 12V$			1.2	
$t_{PHL\ (S\text{-}R)}$	Data output high-to-low delay time, S input, R output	$R_L = 200k\Omega$	$V_{DD} = 5V$			1.2	μs
			$V_{DD} = 9V$			2.5	
			$V_{DD} = 12V$			2	
$t_{PHL\ (P\text{-}R)}$	Data output high-to-low delay time, $P1_0 \sim P7_3$ input, R output	$C_L = 50pF$	$V_{DD} = 5V$			3.5	μs
			$V_{DD} = 9V$			1.8	
			$V_{DD} = 12V$			1.2	
$t_{PHL\ (STD\text{-}P)}$	Data output low-to-high delay time, STD input, R output	$R_L = 2k\Omega$ $C_L = 50pF$	$V_{DD} = 5V$			2.4	μs
			$V_{DD} = 9V$			1.8	
			$V_{DD} = 12V$			1.2	
$t_{PLH\ (S\text{-}R)}$	Data output low-to-high delay time, S input, R output	$R_L = 200k\Omega$	$V_{DD} = 5V$			4.2	μs
			$V_{DD} = 9V$			2.5	
			$V_{DD} = 12V$			2	
$t_{PLH\ (P\text{-}R)}$	Data output low-to-high delay time, $P1_0 \sim P7_3$ input, R output	$C_L = 50pF$	$V_{DD} = 5V$			3.5	μs
			$V_{DD} = 9V$			1.8	
			$V_{DD} = 12V$			1.2	
$t_{PLH\ (STD\text{-}P)}$	Data output low-to-high delay time, STD input, $P1_0 \sim P7_3$ output	$R_L = 2k\Omega$ $C_L = 50pF$	$V_{DD} = 5V$			2.8	μs
			$V_{DD} = 9V$			1.8	
			$V_{DD} = 12V$			1.2	

TIMING DIAGRAM

Multiplexer Mode



Latch Mode



M50790SP

INPUT/OUTPUT EXPANDER

DESCRIPTION

The M50790SP is an input/output expander LSI fabricated using aluminum-gate CMOS technology. It is designed especially for connection with the single-chip 8-bit microcomputer series, and it comes in a 52-pin shrink plastic molded DIL package.

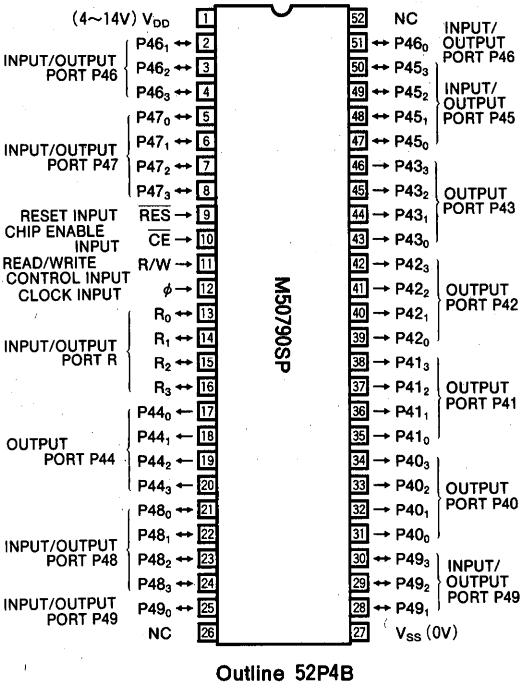
FEATURES

- Wide operating voltage range ························ 4~14V
- Voltage level convertable thanks to n-channel open-drain configuration at I/O port (port R)
- Input/output ports (ports P45~P49) ·············· 4 bits × 5
- Output ports (ports P40~P44) ······················ 4 bits × 5
- High current output
- Output latch data can be read
- Input/output setting possible for each bit individually with 20bits I/O ports

APPLICATION

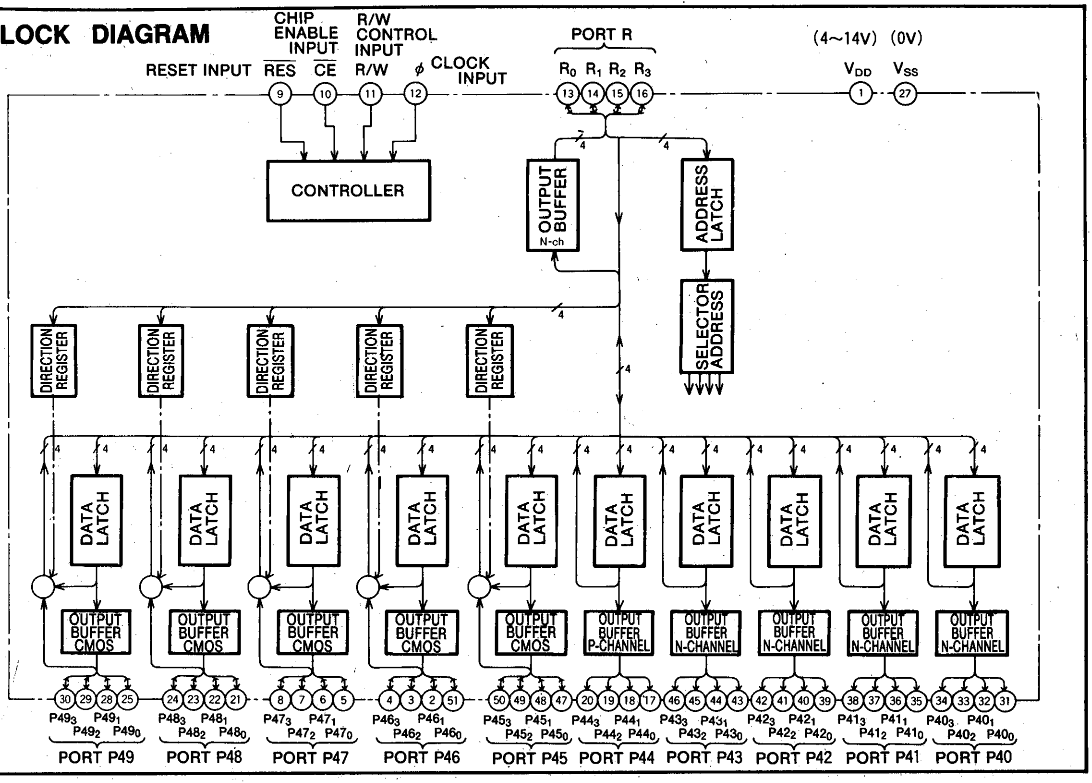
I/O expansion for single-chip 8-bit microcomputer M50740-XXXSP

PIN CONFIGURATION (TOP VIEW)



Outline 52P4B

BLOCK DIAGRAM



FUNCTION

The M50790SP enables expansion from one 4-bit I/O port (port R) group to five 4-bit output port groups and five 4-bit I/O port (port P) groups. The ports are selected as follows: when the clock input ϕ is high, the address is input from port R and when it is low, the data are input (or output) by the same port (port R).

Each I/O port (ports P45 ~ P49) has a direction register (D45~D49) and the input or output can be set for each bit individually.

The contents of all output latches can be read from port R.

PIN DESCRIPTION

Symbol	Name	Input or output	Function
ϕ	Clock input	In	When the input is high, the address of the port to be accessed from port R is designated. The address is latched at the ϕ fall. Conversely, when the input is low, the R port data are input or output.
$\overline{CE}$	Chip enable input	In	If this input is high when ϕ is high, the internal mode is prevented from being changed by external equipment.
R/W	Read/write control input	In	The port R input/output is controlled by the high/low level of this input when ϕ is falling.
$\overline{RES}$	Reset input	In	All the outputs are put into the high-impedance state when this input is low. This means that the open-drain output port latches and direction registers are reset. The CMOS input/output port data latches remain unchanged (see table below).
R	Port R	In/out	Data is sent and received at this 4-bit bidirectional port by the microcomputer's transfer instrucitons. When ϕ is high, the address is read through this port; when low, the data are sent or received through this port.
P40 P41 P42 P43 P44	Output ports 40 Output ports 41 Output ports 42 Output ports 43 Output ports 44	Out	These 4-bit output ports have output latches for each individual bit. The output configuration is n-channel open-drains for P40i~P43i and p-channel open-drains for P44i.
P45 P46 P47 P48 P49	Input/output ports 45 Input/output ports 46 Input/output ports 47 Input/output ports 48 Input/output ports 49	In/out	These 4-bit bidirectional input/output ports have an output latch and direction register for each individual bit and input or output can be designated for each bit. The output configuration is a 3-state CMOS structure.

PORT MODES AFTER RESET INPUT

Port	Direction register	Data latch	Output transistor
P40~P43	—	ALL "H"	N-channel open-drain OFF
P44	—	ALL "L"	P-channel open-drain OFF
P45~P49	ALL low (input)	No change	High-impedance

OPERATION

Address Designation

The address is designated by port R while the clock ϕ signal is high and the address data are latched to the address latch by the ϕ fall (↓). The relationship between each port or address register and the addresses is shown below.

Address and data signs

Address (ϕ = high)				Data (ϕ = low)			
R_3	R_2	R_1	R_0	R_3	R_2	R_1	R_0
L	L	L	L	blank			
L	L	L	H	$P40_3$	$P40_2$	$P40_1$	$P40_0$
L	L	H	L	$P41_3$	$P41_2$	$P41_1$	$P41_0$
L	L	H	H	$P42_3$	$P42_2$	$P42_1$	$P42_0$
L	H	L	L	$P43_3$	$P43_2$	$P43_1$	$P43_0$
L	H	L	H	$P44_3$	$P44_2$	$P44_1$	$P44_0$
L	H	H	L	$P45_3$	$P45_2$	$P45_1$	$P45_0$
L	H	H	H	$D45_3$	$D45_2$	$D45_1$	$D45_0$
H	L	L	L	$P46_3$	$P46_2$	$P46_1$	$P46_0$
H	L	L	H	$D46_3$	$D46_2$	$D46_1$	$D46_0$
H	L	H	L	$P47_3$	$P47_2$	$P47_1$	$P47_0$
H	L	H	H	$D47_3$	$D47_2$	$D47_1$	$D47_0$
H	H	L	L	$P48_3$	$P48_2$	$P48_1$	$P48_0$
H	H	L	H	$D48_3$	$D48_2$	$D48_1$	$D48_0$
H	H	H	L	$P49_3$	$P49_2$	$P49_1$	$P49_0$
H	H	H	H	$D49_3$	$D49_2$	$D49_1$	$D49_0$

Note : D45~D49 are the direction registers of ports P45~49. When D is low, the output is set to the high-impedance state and when high, the CMOS output ON state is established. Designation for each bit in this way is possible. The contents of the direction registers, however, cannot be read out.

Direction Register Setting

For each bit the input/output ports have corresponding direction registers (see table below). When a register is low, the output is set to the high-impedance state and when high, the CMOS output ON mode is established.

Correspondence between I/O ports and direction registers

Output port/input port				Direction register			
$P40_0$	$P40_1$	$P40_2$	$P40_3$	None (Outputs only)			
$P41_0$	$P41_1$	$P41_2$	$P41_3$				
$P42_0$	$P42_1$	$P42_2$	$P42_3$				
$P43_0$	$P43_1$	$P43_2$	$P43_3$				
$P44_0$	$P44_1$	$P44_2$	$P44_3$				
$P45_0$	$P45_1$	$P45_2$	$P45_3$	$D45_0$	$D45_1$	$D45_2$	$D45_3$
$P46_0$	$P46_1$	$P46_2$	$P46_3$	$D46_0$	$D46_1$	$D46_2$	$D46_3$
$P47_0$	$P47_1$	$P47_2$	$P47_3$	$D47_0$	$D47_1$	$D47_2$	$D47_3$
$P48_0$	$P48_1$	$P48_2$	$P48_3$	$D48_0$	$D48_1$	$D48_2$	$D48_3$
$P49_0$	$P49_1$	$P49_2$	$P49_3$	$D49_0$	$D49_1$	$D49_2$	$D49_3$

Output Operation

Output Ports (P40~P44)

In order for the data to be output to the P40 ~ P44 output ports, the address must be designated from port R when ϕ is high. When ϕ is low, the output data must be designated in the same way from port R. In this case, the R/W input is set low with the ϕ fall (↓).

Input/Output Ports (P45~P49)

In the case of the I/O ports the direction register corresponding to the bits must be set high beforehand. As long as the level is high, the output operation can be conducted in exactly the same way as for the P40~P44 ports.

Input Operation

For input the direciton register corresponding to the bits must be set low beforehand. The actual operation consists in designating the input port address from port R when the ϕ input is high and reading the R port data when ϕ is low. In this case, the R/W input is set high with the ϕ fall (↓).

Latch Read Operation

Output Ports (P40~P44)

In order to read the P40~P44 output port latches, the P40~P44 port address is designated when the ϕ input is high and, as soon as R/W is high during the ϕ fall, the data can be read from port R when ϕ is low.

I/O Ports (P45~P49)

In order to read the I/O port latches, the direction register corresponding to the bits must be set high beforehand. (This sets the CMOS outputs ON.) If the direction register is high, then the latch reading operation can be conducted in exactly the same way as with the P40~P44 ports.

Direction Registers (D45~D49)

The contents of the direction registers cannot be read out.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{DD}	Supply voltage	With respect to V_{SS}	−0. 3～15	V
V_I	Input voltage		V_{SS}−0. 3～V_{DD}+0. 3	V
V_O	Output voltage		V_{SS}−0. 3～V_{DD}+0. 3	V
P_d	Power dissipation	T_a = 25℃	600	mW
T_{opr}	Operating temperature		−10～70	℃
T_{stg}	Storage temperature		−40～125	℃

RECOMMENDED OPERATING CONDITIONS (V_{DD} = 9V±10%, T_a = −10～70℃, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Nom	Max	
V_{DD}	Supply voltage		4		14	V
V_I	Input voltage		0		V_{DD}	V
V_{IHP}	High-level input voltage, P45, P46, P47, P48, P49	V_{DD} = 7～14V	V_{DD}X0. 7		V_{DD}	V
V_{ILP}	Low-level input voltage, P45, P46, P47, P48, P49	V_{DD} = 7～14V	0		V_{DD}X0. 3	V
V_{IHR}	High-level input voltage, R, ϕ, R/W, $\overline{CE}$, $\overline{RES}$	V_{DD} = 7～14V	V_{DD}X0. 4		V_{DD}	V
V_{ILR}	Low-level input voltage, R, ϕ, R/W, $\overline{CE}$, $\overline{RES}$	V_{DD} = 7～14V	0		V_{DD}X0. 1	V
$I_{OL(avg)}$	Low-level output average current (R)				5	mA
$I_{OL(avg)}$	Low-level output average current, P40, P41, P42, P43				20	mA
$I_{OH(avg)}$	High-level output average current, (P44)		−10			mA
$I_{OH(avg)}$	High-level output average current, P45, P46, P47, P48, P49		−2			mA
$I_{OL(avg)}$	Low-level output average current, P45, P46, P47, P48, P49				2	mA
$f(\phi)$	M50740-XXXSP internal clock oscillation frequency	V_{DD} = 9V			2. 5	MHz

ELECTRICAL CHARACTERISTICS (V_{DD} = 9V±10%, T_a = −10～70℃, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
I_I	Input current, ϕ, R/W, $\overline{CE}$, $\overline{RES}$				1	μA
I_I	Input current, R	V_I = 0～V_{DD}			1	μA
I_I	Input current, P45, P46, P47, P48, P49				10	μA
V_{OH}	High-level output voltage, P44	I_{OH} = −10mA	V_{DD}−2		V_{DD}	V
V_{OH}	High-level output voltage, P45, P46, P47, P48, P49	I_{OH} = −2mA	V_{DD}−2		V_{DD}	V
V_{OL}	Low-level output voltage, R	I_{OL} = 5mA	0		0. 4	V
V_{OL}	Low-level output voltage, P40, P41, P42, P43	I_{OL} = 20mA	0		2	V
V_{OL}	Low-level output voltage, P45, P46, P47, P48, P49	I_{OL} = 2mA	0		2	V
I_{DD}	Supply current	Output pins open			50	μA

TIMING REQUIREMENTS (Ta = −10~70℃, VDD±10%, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
$t_{su(A-\phi)}$	Address input set-up time	$V_{DD}=9V$	750			ns
		$V_{DD}=12V$	600			ns
$t_{su(D-\phi)}$	Data input set-up time	$V_{DD}=9V$	700			ns
		$V_{DD}=12V$	550			ns
$t_{h(\phi-A)}$	Address hold time after ϕ fall	$V_{DD}=9V$	0			ns
		$V_{DD}=12V$	0			ns
$t_{h(\phi-D)}$	Data hold time after ϕ rise	$V_{DD}=9V$	0			ns
		$V_{DD}=12V$	0			ns
$t_{su(\overline{CE}-\phi)}$	Chip enable set-up time before ϕ fall	$V_{DD}=9V$	550			ns
		$V_{DD}=12V$	400			ns
$t_{h(\phi-\overline{CE})}$	Chip enable hold time after ϕ fall	$V_{DD}=9V$	400			ns
		$V_{DD}=12V$	300			ns
$t_{su(R/W-\phi)}$	Read/write set-up time before ϕ fall	$V_{DD}=9V$	650			ns
		$V_{DD}=12V$	500			ns
$t_{h(\phi-R/W)}$	Read/write hold time after ϕ fall	$V_{DD}=9V$	400			ns
		$V_{DD}=12V$	300			ns
$t_{su(P-\phi)}$	P input set-up time before ϕ fall	$V_{DD}=9V$	750			ns
		$V_{DD}=12V$	600			ns
$t_{h(\phi-P)}$	P input data hold time after ϕ rise	$V_{DD}=9V$	300			ns
		$V_{DD}=12V$	200			ns
$t_{w(\overline{RES})}$	Reset pulse width	$V_{DD}=9V$	400			ns
		$V_{DD}=12V$	300			ns

SWITCHING CHARACTERISTICS ($T_a = -10 \sim 70$℃, $V_{DD} \pm 10\%$, unless otherwise noted)

Symbol	Parameter		Test conditions	Limits			Unit
				Min	Typ	Max	
(During output) $t_{p(\phi\text{-}P)}$	P port output propagation time after ϕ fall (Note 1)	Ports P40~P43	$V_{DD} = 9V$			550	ns
			$V_{DD} = 12V$			400	ns
		Port P44	$V_{DD} = 9V$			700	ns
			$V_{DD} = 2V$			550	ns
		Potrs P45~P49	$V_{DD} = 9V$			900	ns
			$V_{DD} = 12V$			700	ns
(During input) $t_{pxL(\phi\text{-}D)}$	R port output propagation time after ϕ rise		$V_{DD} = 9V$			1000	ns
			$V_{DD} = 12V$			800	ns
$t_{v(\phi\text{-}D)}$	R port data valid time after ϕ rise		$V_{DD} = 9V$	0			ns
			$V_{DD} = 12V$	0			ns
$t_{p(P\text{-}D)}$	Output propagation time, from P input to R port		$V_{DD} = 9V$			900	ns
			$V_{DD} = 12V$			700	ns
(When direction registers are changed)			$V_{DD} = 9V$	0			ns
$t_{v(\phi\text{-}P)}$	P port output data valid time after ϕ fall		$V_{DD} = 12V$	0			ns
$t_{vz(\phi\text{-}P)}$	P port high-impedance state valid time after ϕ fall		$V_{DD} = 9V$	0			ns
			$V_{DD} = 12V$	0			ns
$t_{pxz(D\text{-}P)}$	P port high-impedance state propagation time after ϕ fall		$V_{DD} = 9V$			1200	ns
			$V_{DD} = 12V$			900	ns
$t_{pxv(D\text{-}P)}$	P port valid output propagation time after ϕ fall		$V_{DD} = 9V$			1200	ns
			$V_{DD} = 12V$			900	ns
(During reset) $t_{pvz(\overline{RES}\text{-}P)}$	P port output high-impedance propagation time after reset		$V_{DD} = 9V$			800	ns
			$V_{DD} = 12V$			600	ns
$t_{pvz(\overline{RES}\text{-}R)}$	R port output high-impedance propagation time after reset		$V_{DD} = 9V$			700	ns
			$V_{DD} = 12V$			500	ns
$t_{vz(\overline{RES}\text{-}R)}$	P port output high-impedance valid time after reset rise		$V_{DD} = 9V$	0			ns
			$V_{DD} = 12V$	0			ns

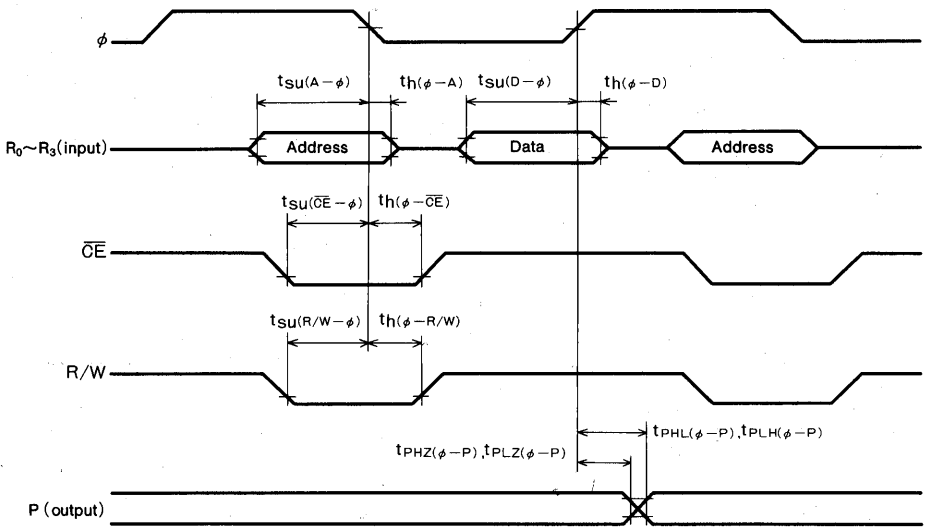
Note 1 : The P port output high-impedance state propagation time after ϕ rise is indicated for the open-drain output ports.

2 : R_0: Port R pull-up resistor (to $V_{CC} = 5V$)
R_1: Port P40~P43 pull-up resistor (to V_{DD})
R_2: Port P44 pull-down resistor (to V_{SS})
R_3: Port p45~P49 load resistor (to V_{DD})

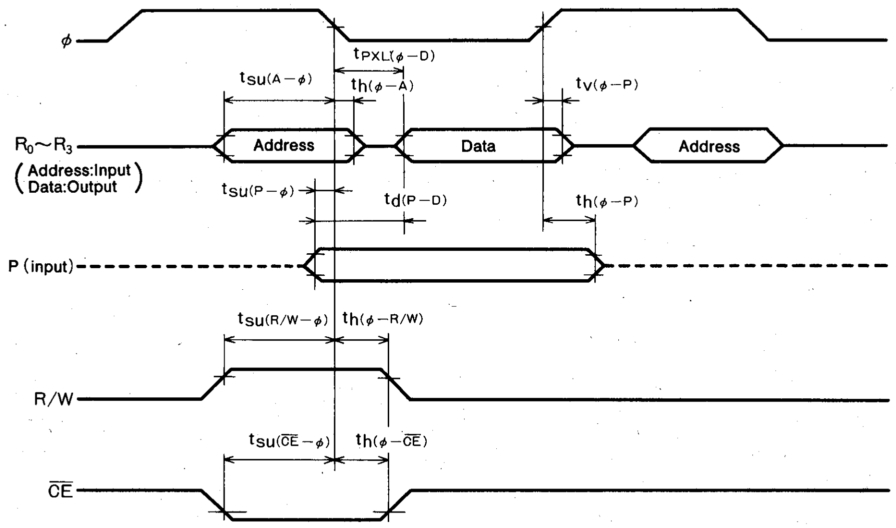
TIMING DIAGRAM (With reference voltage of 0.9 × V_{DD} for high level and 0.1 × V_{DD} for low level)

For R/W, $\overline{CE}$, ϕ and $\overline{RES}$, R_0~R_3, high level is 0.9 × V_{CC} and low level is 0.1 × V_{CC}.

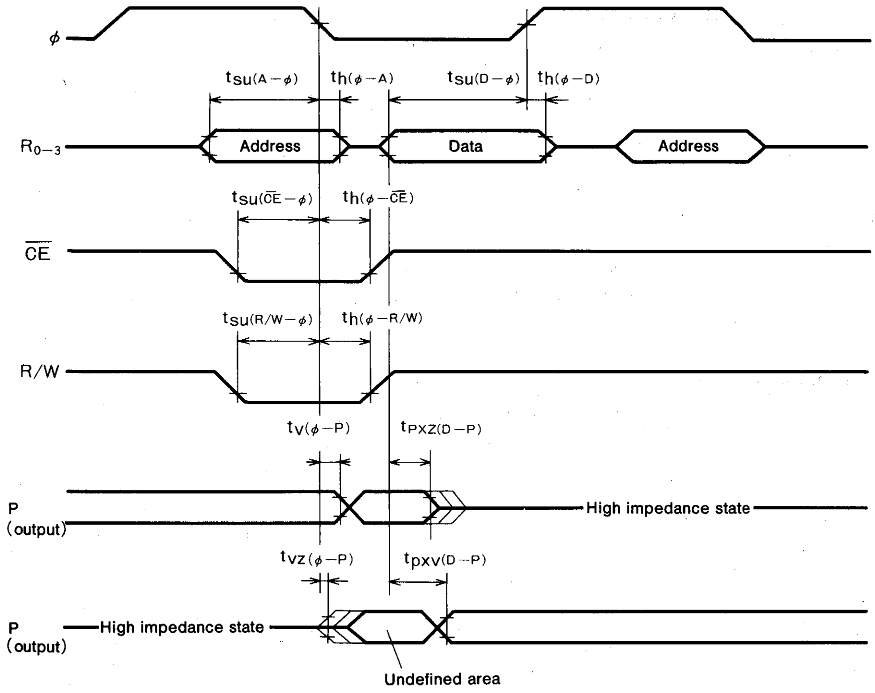
During output



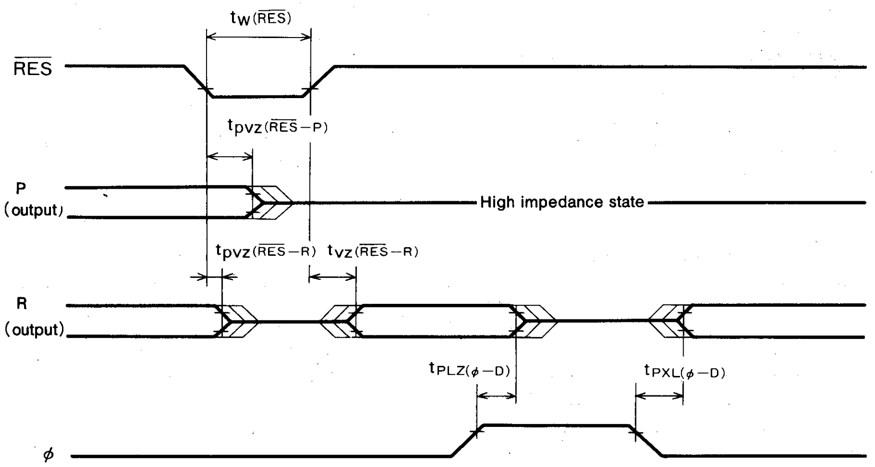
During input



During direction register changing



During resetting



APPLICATION EXAMPLES

Connection with M50740SP

Fig. 1 shows the connections between the M50790SP I/O expander and the M50740-XXXSP. The addresses listed in Table 1 are reserved for the M50790SP, and data read and write operations are possible in the same way as for the internal ports of the M50740-XXXSP. (Only the low-order 4 bits of the data are valid.) The timing and control signals are generated automatically at the M50740-XXXSP end.

The operation is now described. The output ports are set OFF state and the input/output ports are placed in the high-impedance state by the RESET OUT signal from the M50740-XXXSP.

Port inputting or outputting is conducted by the same instructions as those to the M50740-XXXSP's other zero page memory. With output port P40 ~ P44 read operations the contents of the respective output latches can be read out. With input/output ports P45~P49 the contents of the direction register can determine whether the read data come from the output latches or input ports. "1's" for each direction register bit signifies output and "0's" signifies input.

The contents of the direction registers cannot be read out. They must be set using the store instruction. Table 2 shows the codes written into the direction registers and the states of the input/output bits of the ports.

Fig. 2 shows the construction of each of the output ports. When an address shown in Table 1 is accessed, double the normal instruction execution time is required and so care must be taken when calculating the processing time.

This precaution must be taken since a margin is provided for interfacing with the I/O expander.

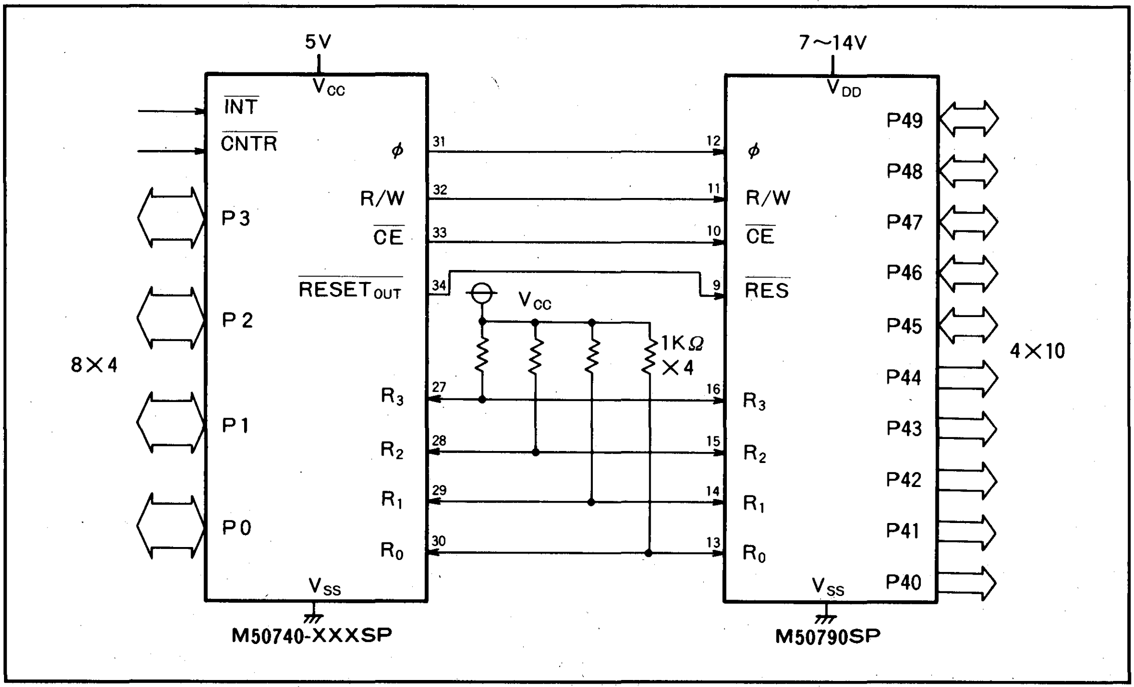


Fig.1 Example of connections between I/O expander (M50790SP) and microcomputer (M50740-XXXSP)

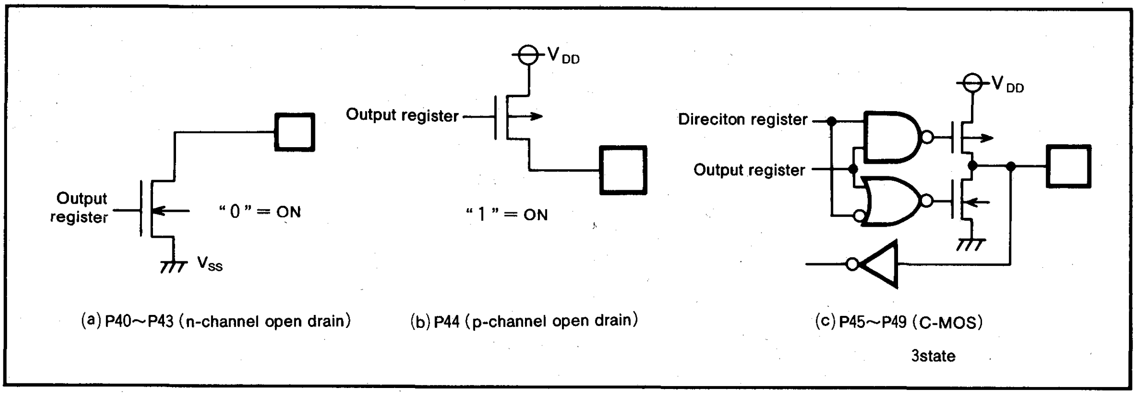
Table 1 Addresses reserved for I/O expander

Address (hexadecimal)	Data bit								Remarks
in zero page	D₇*	D₆*	D₅*	D₄*	D₃	D₂	D₁	D₀	
DF					P49	Direction Register			0: Input, 1: Output
DE					P49				CMOS 3state I/O
DD					P48	Direction Register			0: Input, 1: Output
DC					P48				CMOS 3state I/O
DB					P47	Direction Register			0: Input, 1: Output
DA					P47				CMOS 3state I/O
D9					P46	Direction Register			0: Input; 1: Output
D8					P46				CMOS 3state I/O
D7					P45	Direction Register			0: Input, 1: Output
D6					P45				CMOS 3state I/O
D5					P44				Pch Open Drain Output
D4					P43				Nch OPen Drain Output
D3					P42				〃
D2					P41				〃
D1					P40				〃
D0					*				

* Bits $D_4 \sim D_7$ are ignored when the M50790SP is accessed.

Table 2 Port setting examples

Direction register (low-order 4 bits)				Port P4i bit			
D_3	D_2	D_1	D_0	P 4 i_3	P 4 i_2	P 4 i_1	P 4 i_0
0	0	0	0	Input	Input	Input	Input
0	0	0	1	Input	Input	Input	Output
0	0	1	0	Input	Input	Output	Input
0	0	1	1	Input	Input	Output	Output
0	1	0	0	Input	Output	Input	Input
0	1	0	1	Input	Output	Input	Output
0	1	1	0	Input	Output	Output	Input
0	1	1	1	Input	Output	Output	Output
1	0	0	0	Output	Input	Input	Input
1	0	0	1	Output	Input	Input	Output
1	0	1	0	Output	Input	Output	Input
1	0	1	1	Output	Input	Output	Output
1	1	0	0	Output	Output	Input	Input
1	1	0	1	Output	Output	Input	Output
1	1	1	0	Output	Output	Output	Input
1	1	1	1	Output	Output	Output	Output



(a) P40~P43 (n-channel open drain)

(b) P44 (p-channel open drain)

(c) P45~P49 (C-MOS) 3state

Fig.2 Output port format

MITSUBISHI MICROCOMPUTERS

M5L8243P

INPUT/OUTPUT EXPANDER

DESCRIPTION

The M5L8243P is an input/output expander fabricated using N-channel silicon-gate ED-MOS technology. This device is designed specifically to provide a low-cost means of I/O expansion for the MELPS 8-48 single-chip microcomputers and MELPS 8-41 slave microcomputers.

FEATURES

- 16 Input/output pins (I_{OL} = 5.0mA(max.))
- Simple interface to MELPS 8-48, MELPS 8-41
- Single 5V power supply
- Low power dissipation ······························ 50mW(typ.)
- Interchangeable with i8243 in pin configuration and electrical characteristics

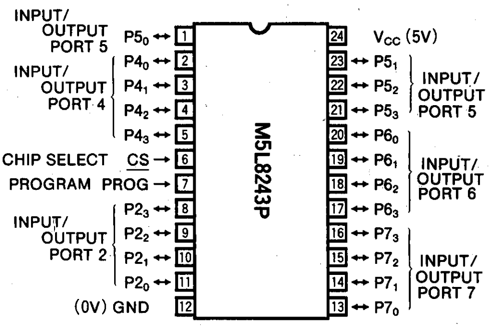
APPLICATION

I/O expansion for the MELPS 8-48 single-chip microcomputers and MELPS 8-41 slave microcomputers.

FUNCTION

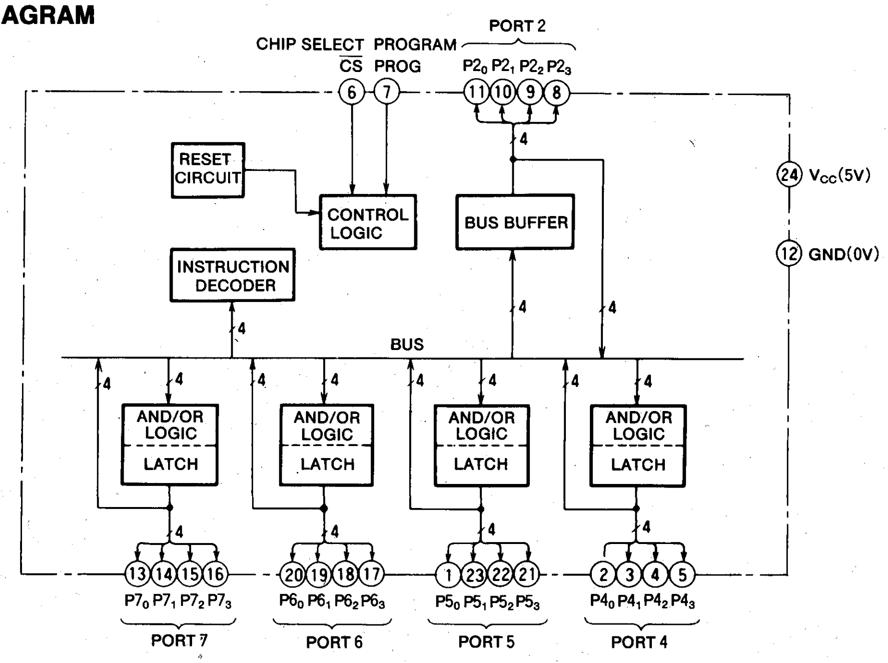
The M5L8243P is designed to provide a low-cost means of I/O expansion for the MELPS 8-41 and the MELPS 8-48. The M5L8243P consists of four 4-bit bidirectional static I/O ports and one 4-bit port which serves as an interface to the MELPS 8-41 and MELPS 8-48. Thus multiple M5L8243Ps can be added to a single master. Using the original instruction set of the master, the M5L8243P serves as the in resident I/O facility. Its I/O ports are accessed by instructions MOVD, ANLD and ORLD.

PIN CONFIGURATION (TOP VIEW)



Outline 24P4

BLOCK DIAGRAM



PIN DESCRIPTION

Symbol	Name	Input or output	Function
$\overline{CS}$	Chip select	In	Chip select input. A high on $\overline{CS}$ causes PROG input to be regarded high inside the M5L8243P, then this inhibits any change of output or internal status.
PROG	Program	In	A high-to-low transition on PROG signifies that address (PORT 4-7) and control are available on PORT 2, and a low-to-high transition signifies that the designated data is available on the disignated port through PORT 2. The designation is shown in Table 1.
$P2_0 \sim P2_3$	Input/output port 2	In/out	The 4-bit bidirectional port contains the address and control bits shown in Table 1 on a high-to-low transition of PROG. During a low-to-high transition it contains the input (output) data on this port.
$P4_0 \sim P4_3$ $P5_0 \sim P5_3$ $P6_0 \sim P6_3$ $P7_0 \sim P7_3$	Input/output port 4 Input/output port 5 Input/output port 6 Input/output port 7	In/out	The 4-bit bidirectional I/O port. May be programmed to be input, low-impedance latched output or a three-state. This port is automatically set output mode when it is written. ANLed or ORLed then continues its mode until next read operation. After reset on a read operation, this port is in high-impedance and input mode.

OPERATION

The M5L8243P is an input/output expander designed specifically for the MELPS 8-41 and MELPS 8-48. The MELPS 8-41 and MELPS 8-48 already have instructions and PROG pin to communicate with the M5L8243P.

An example of the M5L8243P and the MELPS 8-41 or MELPS 8-48 is shown in Fig. 1. The following description of the M5L8243P basic operation is made according to Fig. 1.

Upon initial application of power supply to the device, and then about 500μs after, resident bias circuits become stable and each device is ready to operate. And each port of the M5L8243P is set input mode (high-impedance) by means of a resident power-on initialization circuit.

When the microcomputer begins to execute a transfer instruction

MOVD A , Pi i = 4, 5, 6, 7

which means the value on the port Pi is transferred to the accumulator, then the signals are sent out on the pins PROG and $P2_0 \sim P2_3$ as shown in Timing Diagram.

On the high-to-low transition of the pin PROG, the M5L8243P latches the instructions (ex. 0000) into itself from pins $P2_0 \sim P2_3$ and transfers them to the instruction register (① in Timing Diagram). During the low-level of PROG, the M5L8243P continuously outputs the contents of the specified input (output) port (in this case port P_4) to pins $P2_0 \sim P2_3$ (② in Timing Diagram). The microcomputer, at an appropriate time, latches the level of pins $P2_0 \sim P2_3$ and resumes high-level of PROG.

The next example is the case in which the microcomputer executes

MOVD Pi , A i = 4, 5, 6, 7

the transfer (output) instruction.

In this case, as in the previous case, on the high-to-low transition of the pin PROG, the M5L8243P latches the instructions (ex. 0110) into itself from pins $P2_0 \sim P2_3$ and transfers them to the instruction register (① in Timing Diagram). After this, the microcomputer sends out high to the pin PROG, transferring the data to pin $P2_0 \sim P2_3$ which is an output data to input/output port. Then the M5L8243P transfers the data of pins $P2_0 \sim P2_3$ to the port latch of the designated input/output port (in this case P_6). In a few seconds after a low-to-high transition on the PROG, the designated port (P_6) becomes in an output mode and the data of the port latch are transferred to the port pins (③ in Timing Diagram).

When instructions

ANLD Pi , A
ORLD Pi , A i = 4, 5, 6, 7

are executed, the microcomputer generally operates as same function as MOVD Pi, A.

It only differs in that the data of port latch after ④ in the Timing Diagram is ANDed or ORed with the data of port latch before ④ and the data of pins $P2_0 \sim P2_3$.

When instructions

MOVD Pi , A
ANLD Pi , A
ORLD Pi , A i = 4, 5, 6, 7

are executed toward the port in an output mode, the outputs are generated on the port as soon as low-to-high transition on the PROG occurs.

When the mode of the output port is going to be changed during the execution and the instruction

MOVD A , Pi i = 4, 5, 6, 7

is executed, it is preferable to execute one dummy instruction. Because it takes a little time to turn the designated port into a high-impedance state after high-to-low transition on the PROG, the result may be that the first instruction is not read correctly.

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
P_d	Maximum power dissipation	$T_a = 25$℃	600	mW
T_{opr}	Operating free-air temperature range		−20~75	℃
T_{stg}	Storage temperature range		−65~150	℃

RECOMMENDED OPERATING CONDITIONS ($T_a = -20$~75℃, $V_{CC} = 5V \pm 10\%$, 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 ($T_a = -20$~75℃, $V_{CC} = 5V \pm 10\%$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{OL1}	Low-level output voltage, ports 4~7	$I_{OL} = 5mA$			0.45	V
V_{OL2}	Low-level output voltage, port 7	$I_{OL} = 20 mA$			1	V
V_{OL3}	Low-level output voltage, port 2	$I_{OL} = 0.6mA$			0.45	V
V_{OH1}	High-level output voltage, ports 4~7	$I_{OH} = -240\mu A$	2.4			V
V_{OH2}	High-level output voltage, port 2	$I_{OH} = -100\mu A$	2.4			V
I_{I1}	Input leakage current, ports 4~7	$0V \leqq V_{in} \leqq V_{CC}$	−10		20	μA
I_{I2}	Input leakage current, port 2, CS, PROG	$0V \leqq V_{in} \leqq V_{CC}$	−10		10	μA
I_{CC}	Supply current from V_{CC}			10	20	mA
I_{OL}	Sum of all I_{OL} from 16 outputs	$I_{OL} = 5mA$ ($V_{OL} = 0.45V$) Each pin			80	mA

Table 1 Instruction and address codes

Instruction code	$P2_3$	$P2_2$	Address code	$P2_1$	$P2_0$
Read	0	0	port 4	0	0
Write	0	1	port 5	0	1
ORLD	1	0	port 6	1	0
ANLD	1	1	port 7	1	1

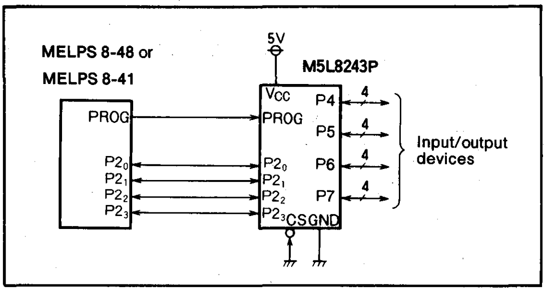


Fig.1 Basic connection

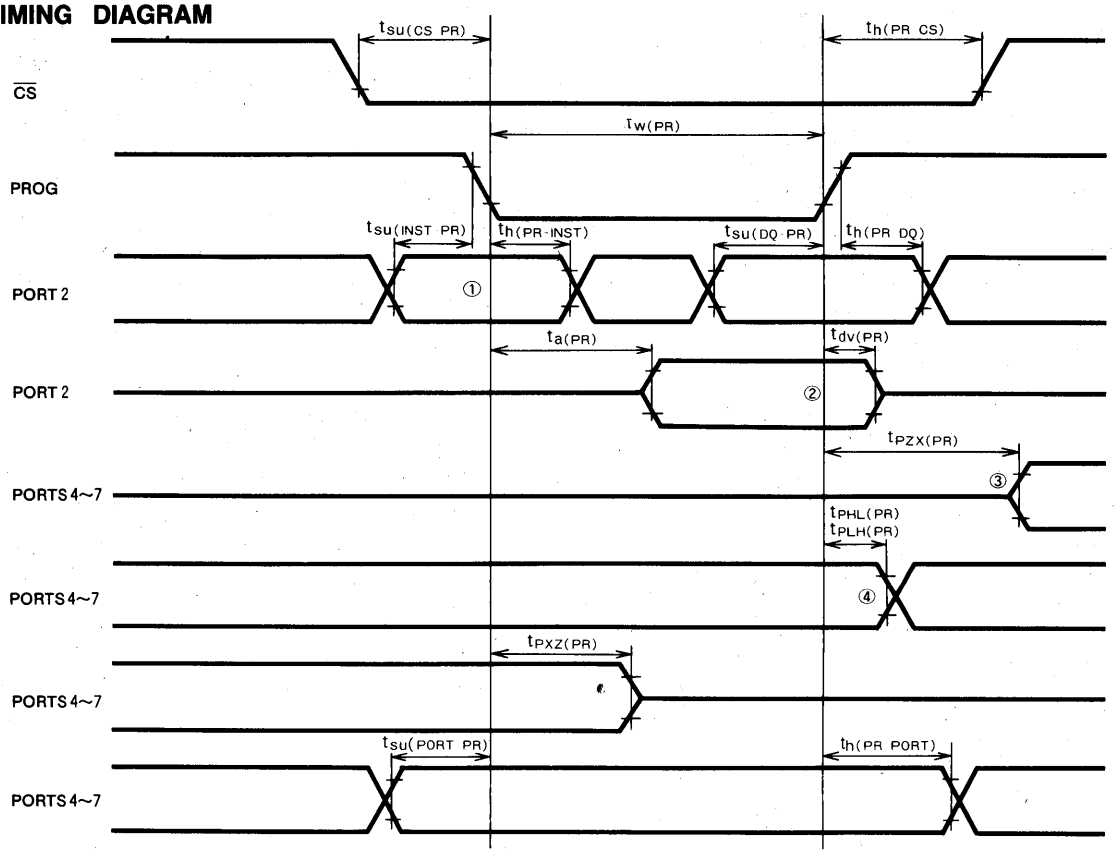
TIMING REQUIREMENTS ($T_a = -20 \sim 75℃$, $V_{CC} = 5V \pm 10\%$, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{su(INST-PR)}$	Instruction code setup time before PROG	t_A	80pF Load	100			ns
$t_{h(PR-INST)}$	Instruction code hold time after PROG	t_B	20pF Load	60			ns
$t_{su(DQ-PR)}$	Data setup time before PROG	t_C	80pF Load	200			ns
$t_{h(PR-DQ)}$	Data hold time after PROG	t_D	20pF Load	20			ns
$t_{w(PR)}$	PROG pulse width	t_K		700			ns
$t_{su(CS-PR)}$	Chip-select setup time before PROG	t_{CS}		50			ns
$t_{h(PR-CS)}$	Chip-select hold time after PROG	t_{CS}		50			ns
$t_{su(PORT-PR)}$	Port setup time before PROG	t_{IP}		100			ns
$t_{h(PR-PORT)}$	Port hold time after PROG	t_{IP}		100			ns

SWITCHING CHARACTERISTICS ($T_a = -20 \sim 75℃$, $V_{CC} = 5V \pm 10\%$, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{a(PR)}$	Data access time after PROG	t_{ACC}	80pF Load	0		650	ns
$t_{dv(PR)}$	Data valid time after PROG	t_H	20pF Load	0		150	ns
$t_{PHL(PR)}$ $t_{PLH(PR)}$	Output valid time after PROG	t_{PO}	100pF Load			700	ns
$t_{PZX(PR)}$ $t_{PXZ(PR)}$	Input/output switching time	—				800	ns

TIMING DIAGRAM



Note 1 : AC test conditions

Input pulse level································ 0.45~2.4V

Input pulse rise time tr (10%~90%) ············ 20ns

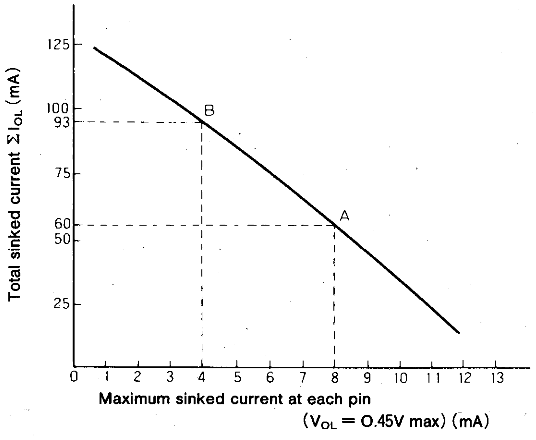
Input pulse fall time tf (10%~90%) ············· 20ns

Reference voltage for switching characteristic measurement.

Output V_{OH}···2V

V_{OL}·· 0.8V

Current Sinking Capability



Each of the 16 I/O lines of the M5L8243P is capable of sinking 5mA simultaneously (V_{OL} = 0.45V max). However, the drive capacity of each line depends upon whether all lines are sinking current simultaneously and on the degree of loading. This is illustrated in the curve shown.

Example

Assuming that the remaining pins are not loaded, how many pins would be able to accommodate 20LSTTL loads (0.4mA) ?

I_{OL} = 0.4mA X 20 = 8mA (sink current for each pin)

ΣI_{OL} = 60mA from curve (POINT A)

(total sinking current)

Number of pins = 60mA ÷ 8mA = 7.5>7

For this case, each of the 7 lines could sink 8mA for a total of 56mA. Since 4mA reserve sinking capability exists, 9 of the I/O lines of the M5L8243P can be divided arbitrarily.

Example

To use 20mA sinking capability at port 7, find the effects on the sinking capabilities of the other I/O lines.

Assume the M5L8243P is driving loads as shown below.

3 lines: −20mA (V_{OL} = 1.0V max, port 7 only)

4 lines: −4mA (V_{OL} = 0.45V max)

9 lines: −1.6mA (V_{OL} = 0.45V max)

Is this within the allowable limit ?

ΣI_{OL} = (20mA × 3) + (4mA × 4) + (1.6mA × 9) = 90.4mA

From the curve we see that with respect to I_{OL} = 4mA, I_{OL} is 93mA (Point B) and that the above load of 90.4mA is within the limit of 93mA.

Note: The sinking current of ports 4 ~ 7 must not exceed 30mA regardless of the value of V_{OL}.

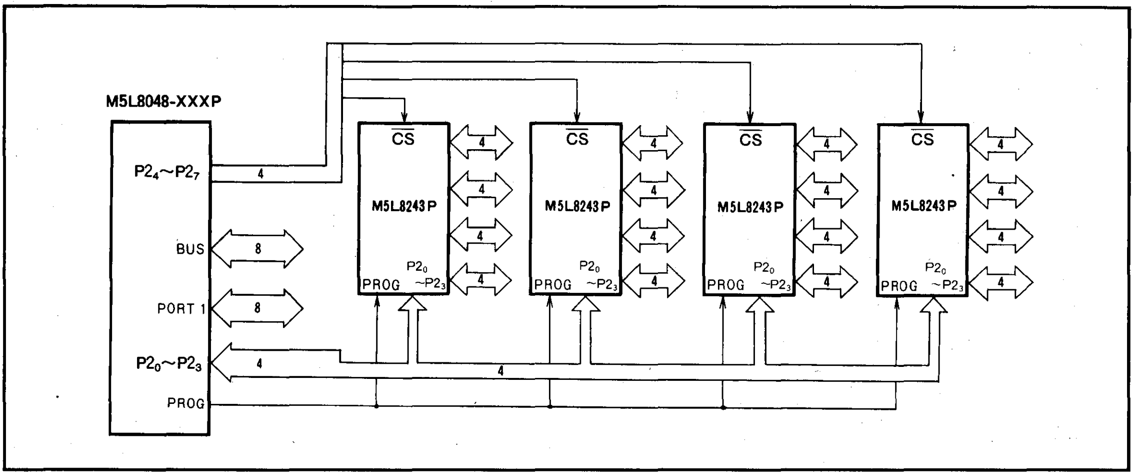


Fig.2 Expansion interface example

DESCRIPTION

The M5M82C43P is an input/output expander fabricated using CMOS silicon-gate technology. This device is designed specifically to provide a low-cost means of I/O expansion for the MELPS8-48 single-chip 8-bit microcomputers and the MELPS8-41 slave microcomputers.

FEATURES

- 16 input/output pins (I_{OL} = 5.0mA (max.))
- Simple interface to MELPS8-48, MELPS8-41
- Single 5V power supply
- Interchangeable with i8243 in pin configuration

APPLICATION

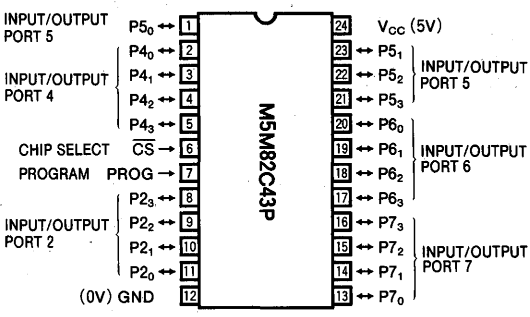
I/O expansion for the MELPS8-48 single-chip microcomputers and the MELPS8-41 slave microcomputers.

FUNCTION

The M5M82C43P is designed to provide a low-cost means of I/O expansion for the MELPS8-41 slave microcomputers and the MELPS8-48 single-chip microcomputers. The M5M82C43P consists of four 4-bit bidirectional static I/O ports and one 4-bit port which serves as an interface to the MELPS8-41 and MELPS8-48. Thus multiple M5M82C43Ps can be added to a single master.

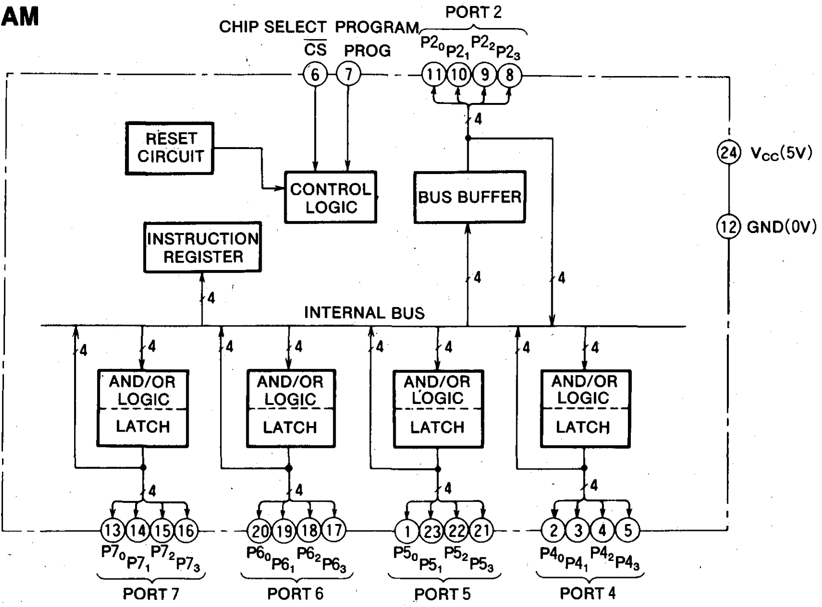
Using the original instruction set of the master, the M5M82C43P serves as the in-resident I/O facility. Its I/O ports are accessed by instructions MOVD, ANLD and ORLD.

PIN CONFIGURATION (TOP VIEW)



Outline 24P4

BLOCK DIAGRAM



PIN DESCRIPTION

Symbol	Name	Input or output	Function
$\overline{CS}$	Chip select	In	Chip select input. A high on $\overline{CS}$ causes PROG input to be regarded high inside the M5M82C43P. This then inhibits any change of output or internal status.
PROG	Program	In	A high-to-low transition on PROG signifies that address (ports 4~7) and control are available on port 2, and a low-to-high transition signifies that the designated data is available on the designated port through port 2. The designation is shown in Table 1.
$P2_0 \sim P2_3$	Input/output port 2	In/out	This 4-bidirectional port contains the address and control bits shown in Table 1 on a high-to-low transition of PROG. During a low-to-high transition, it contains the input (output) data on this port.
$P4_0 \sim P4_3$ $P5_0 \sim P5_3$ $P6_0 \sim P6_3$ $P7_0 \sim P7_3$	Input/output port 4 Input/output port 5 Input/output port 6 Input/output port 7	In/out	4-bit bidirectional I/O ports. May be programmed to be input, low-impedance latched or 3-state. These ports are automatically set to the output mode when written, ANLed or ORLed and this mode continues until the next read operation. After reset on a read operation, this port is placed in the high impedance and input mode.

OPERATION

The M5M82C43P is an input/output expander designed specifically for the MELPS8-41 and MELPS8-48. The MELPS8-41 and MELPS8-48 already have instructions and PROG pin to communicate with the M5M82C43P.

An example of the M5M82C43P and the M5M80C49-XXXP is shown in Fig. 1. The following description of the M5M82C43P basic operation is made according to Fig. 1.

Upon initial application of the power supply to the device, each port of the M5M82C43P is set to the input mode (high-impedance) by means of the resident power-on initialization circuit.

When the microcomputer begins to execute a transfer instruction

MOVD **A, Pi** i = 4, 5, 6, 7

which means the value on the port Pi is transferred to the accumulator, then the signals are sent out on the pins PROG and $P2_0 \sim P2_3$, as shown in the timing diagram.

On the high-to-low transition of pin PROG, the M5M82C43P latches the instructions (e.g. 0000) into itself from pins $P2_0 \sim P2_3$ and transfers them to the instruction register (① in the timing diagram). During the low-level of PROG, the M5M82C43P continuously outputs the contents of the specified input (output) port (in this case, port P4) to pins $P2_0 \sim P2_3$ (② in the timing diagram). The microcomputer, at the appropriate time, latches the level of pins $P2_0 \sim P2_3$ and resumes the high level of PROG.

The next example is the case in which the microcomputer executes

MOVD **PI, A** i = 4, 5, 6, 7

the transfer (output) instruction.

In this case, as in the previous case, on the high-to-low transition of pin PROG, the M5M82C43P latches the instructions (e.g.0110) into itself from pins $P2_0 \sim P2_3$ and transfers them to the instruction register (① in the timing diagram).

After this the microcomputer sends out high to pin PROG, transferring the data to pins $P2_0 \sim P2_3$ which is an output data to the input/output port. Then the M5M82C43P transfers the data of pins $P2_0 \sim P2_3$ to the port latch of the designated input/output port (in this case P6). In a few seconds after a low-to-high transition on the PROG, the designated port (P6) is set to the output mode and the data of the port latch is transferred to the port pins (③ in the timing diagram).

When instructions

ANLD **Pi, A**
ORLD **Pi, A** i = 4, 5, 6, 7

are executed, the microcomputer generally operates as the same function as MOVD Pi, A.

It only differs in that the data of the port latch after ④ in the timing diagram is ANDed or ORed with the data of the port latch before ④ and the data of pins $P2_0 \sim P2_3$.

When instructions

MOVD **Pi, A**
ANLD **Pi, A**
ORLD **Pi, A** i = 4, 5, 6, 7

are executed toward the port in an output mode, the outputs are generated on the port as soon as low-to-high transition on the PROG occurs.

When the mode of the output port is going to be changed during the execution and the instruction

MOVD **A, Pi** i = 4, 5, 6, 7

is executed, it is preferable to execute one dummy instruction. Because it takes a little time to turn the designated port into a high-impedance state after the high-to-low transition on the PROG, the result may be that the first instruction is not read correctly.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Limits	Unit
V_{CC}	Supply voltage		V_{SS}−0.3~7	V
V_I	Input voltage		V_{SS}−0.3~V_{CC}+0.3	V
V_O	Output voltage		V_{SS}−0.3~V_{CC}+0.3	V
P_d	Maximum power dissipation	T_a = 25℃	1000	mW
T_{opr}	Operating free-air temperature range		−40~85	℃
T_{stg}	Storage temperature range		−65~150	℃

RECOMMENDED OPERATING CONDITIONS (T_a = −40~85℃, V_{CC} = 5V±10%, 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	0.7XV_{CC}		V_{CC}	V
V_{IL}	Low-level input voltage	V_{SS}		0.3XV_{CC}	V

ELECTRICAL CHARACTERISTICS (T_a = −40~85℃, V_{CC} = 5V±10%, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min	Typ	Max	
V_{OL1}	Low-level output voltage, ports 4~7	I_{OL} = 5mA			0.45	V
V_{OL2}	Low-level output voltage, port 7	I_{OL} = 20mA			1	V
V_{OL3}	Low-level output voltage, port 2	I_{OL} = 0.6mA			0.45	V
V_{OH1}	High-level output voltage, ports 4~7	I_{OH} = −240μA	0.75XV_{CC}			V
V_{OH2}	High-level output voltage, port 2	I_{OH} = −100μA	0.75XV_{CC}			V
I_{I1}	Input leakage current, ports 4~7	V_{SS} ≦ Vin ≦ V_{CC}	−10		20	μA
I_{I2}	Input leakage current port 2, $\overline{CS}$, PROG	V_{SS} ≦ Vin ≦ V_{CC}	−10		10	μA
I_{CC1}	Supply current (1)	V_{CC} = 5.5V, V_{IN} = V_{CC} or V_{SS} PROG input pulse period = 5μs			2	mA
I_{CC2}	Supply current (2)	V_{CC} = 5.5V, V_{IN} = V_{CC} or V_{SS} PROG = V_{CC}			10	μA
I_{OL}	Sum of all I_{OL} from 16 outputs	I_{OL} = 5mA (V_{OL} = 0.45V) each pin			80	mA

Table 1 Instruction and address codes

Instruction code	$P2_3$	$P2_2$	Address code	$P2_1$	$P2_0$
Read	0	0	Port 4	0	0
Write	0	1	Port 5	0	1
ORLD	1	0	Port 6	1	0
ANLD	1	1	Port 7	1	1

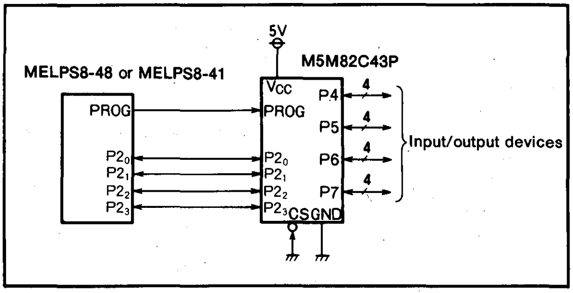


Fig.1 Basic connection

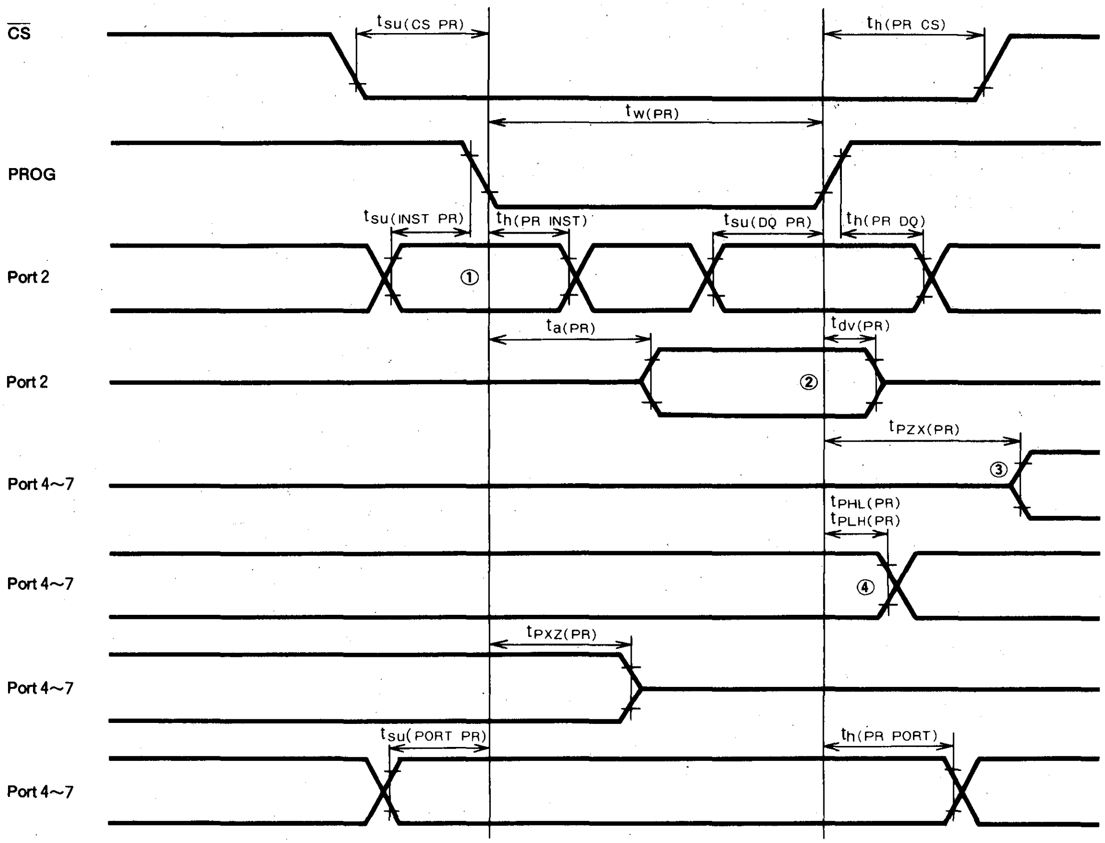
TIMING REQUIREMENTS ($T_a = -40 \sim 85$℃, $V_{CC} = 5V \pm 10\%$, $V_{SS} = 0V$, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{su(INST-PR)}$	Instruction code setup time befor PROG	t_A	$C_L = 80pF$	100			ns
$t_{h(PR-INST)}$	Instruction code hold time after PROG	t_B	$C_L = 20pF$	60			ns
$t_{su(DQ-PR)}$	Data setup time before PROG	t_C	$C_L = 80pF$	200			ns
$t_{h(PR-DQ)}$	Data hold time after PROG	t_D	$C_L = 20pF$	20			ns
$t_{w(PR)}$	PROG pulse with	t_K		700			ns
$t_{su(CS-PR)}$	Chip select setup time before PROG	t_{CS}		50			ns
$t_{h(PR-CS)}$	Chip select hold time after PROG	t_{CS}		50			ns
$t_{su(PORT-PR)}$	Port setup time before PROG	t_{IP}		100			ns
$t_{h(PR-PORT)}$	Port hold time after PROG	t_{IP}		100			ns

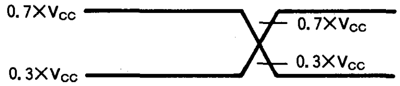
SWITCHING CHARACTERISTICS ($T_a = -40 \sim 85$℃, $V_{CC} = 5V \pm 10\%$, $V_{SS} = 0V$, unless otherwise noted)

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{a(PR)}$	Data access time after PROG	t_{ACC}	$C_L = 80pF$	0		650	ns
$t_{dv(PR)}$	Data valid time after PROG	t_H	$C_L = 20pF$	0		150	ns
$t_{PHL(PR)}$ $t_{PLH(PR)}$	Output valid time after PROG	t_{PO}	$C_L = 100pF$			700	ns
$t_{PZX(PR)}$ $t_{PXZ(PR)}$	Input/output switching time	—				800	ns

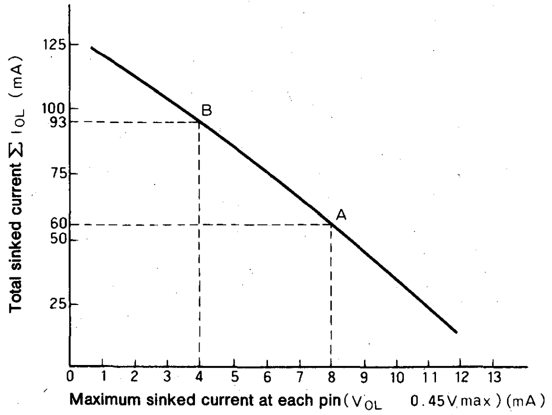
TIMING DIAGRAM



Note 1 : Input pulse level································· 0.3 × V_{CC}~0.7 × V_{CC}
Input pulse rise time tr ·· 20ns
Input pulse fall time tf ·· 20ns
Reference voltage for switching characteristic measurement



Current Sinking Capability



Each of the 16 I/O lines of the M5M82C43P is capable of sinking 5mA simultaneously (V_{OL} = 0.45V max). However, the drive capacity of each line depends upon whether all lines are sinking current simultaneously and on the degree of loading. This is illustrated in the curve shown.

Example:

Assuming that the remaining pins are not loaded, how many pins would be able to accommodate 20LSTTL loads (0.4mA)?

I_{OL} = 0.4mA × 20 = 8mA (sink current for each pin)
ΣI_{OL} = 60mA from curve (point A)
(total sinking current)
Number of pins = 60mA ÷ 8mA = 7.5>7

For this case, each of the 7 lines could sink 8mA for a total of 56mA. Since a 4mA reserve sinking capability exists, 9 of the I/O lines of the M5M82C43P can be divided arbitrarily.

Example:
To use the 20mA sinking capability at port 7, find the effects on the sinking capabilities of the other I/O lines. Assume the M5M82C43P is driving loads as shown below:

3 lines: 20mA (V_{OL} = 1.0V max, port 7 only)
4 lines: 4mA (V_{OL} = 0.45V max)
9 lines: 1.6mA (V_{OL} = 0.45V max)

Is this within the allowable limit ?

$\Sigma\ I_{OL}$ = (20mA × 3) + (4mA × 4) + (1.6mA × 9) = 90.4mA

From the curve it is seen that with respect to I_{OL} = 4mA, I_{OL} is 93mA (point B) and that the above load of 90.4mA is within the limit of 93mA.

Note: The sinking current of port 4 ~ 7 must not exceed 30mA regardless of the value of V_{OL}.

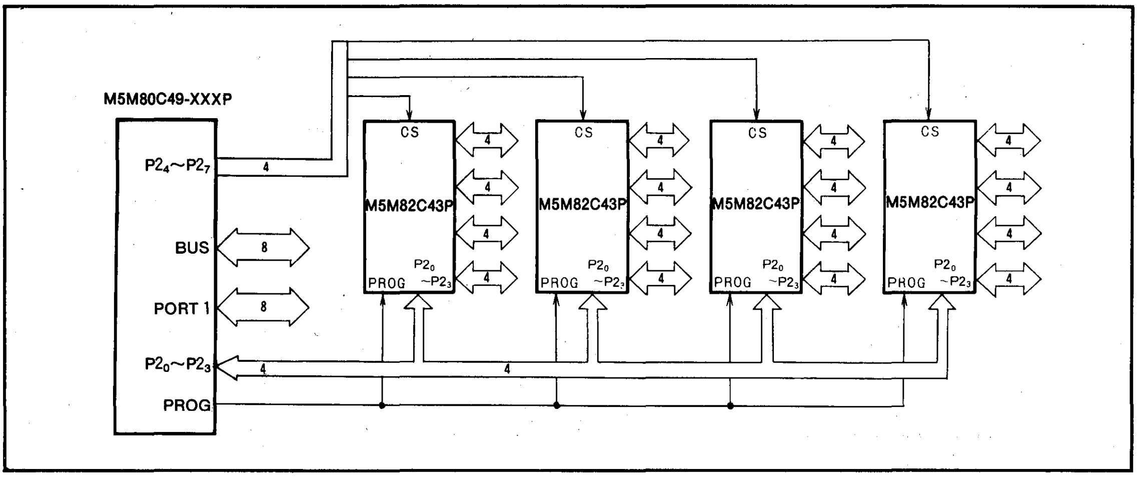
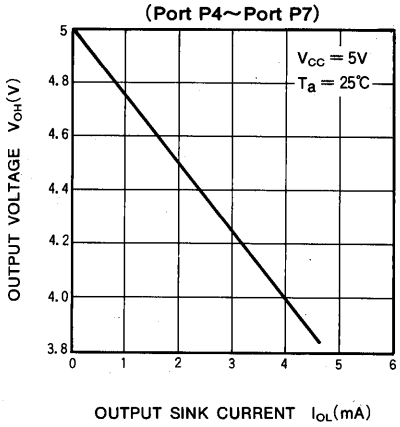
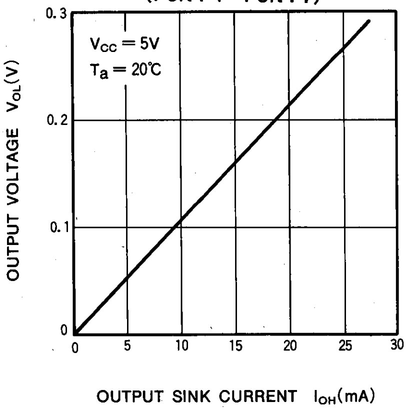


Fig.2 Expansion interface example

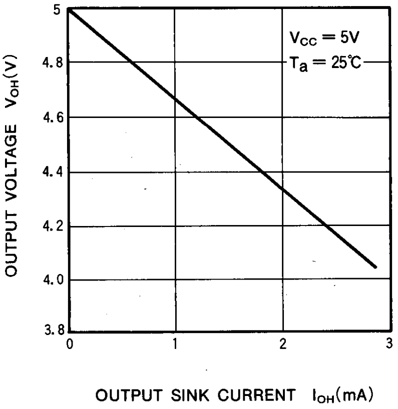
OUTPUT VOLTAGE VS.
OUTPUT SINK CURRENT
(Port P4~Port P7)



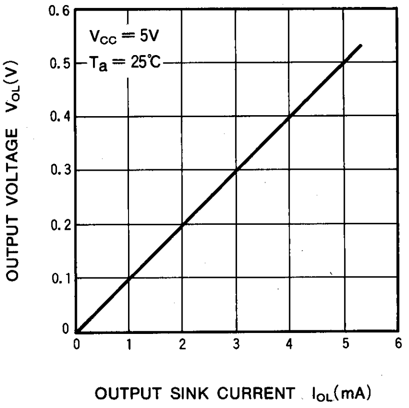
OUTPUT VOLTAGE VS.
OUTPUT SINK CURRENT
(Port P4~Port P7)



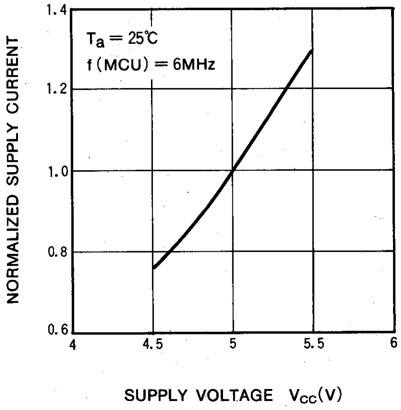
OUTPUT VOLTAGE VS.
OUTPUT SINK CURRENT
(Port P2)



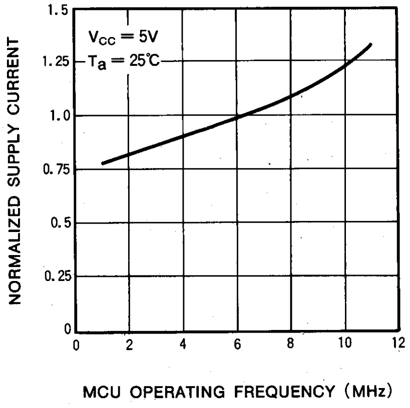
OUTPUT VOLTAGE VS.
OUTPUT SINK CURRENT
(Port P2)



NORMALIZED SUPPLY CURRENT (I_{CC}) VS.
SUPPLY VOLTAGE



NORMALIZED SUPPLY CURRENT (I_{CC}) VS.
MCU OPERATING FREQUENCY



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 Nchannel 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
- Configuration and electrical characteristics

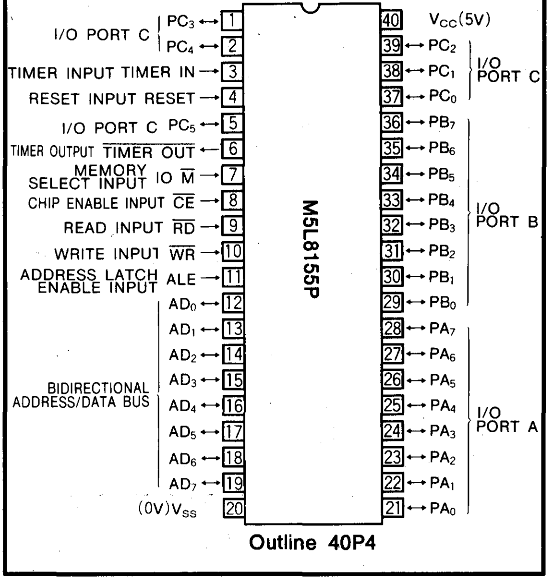
APPLICATION

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

FUNCTION

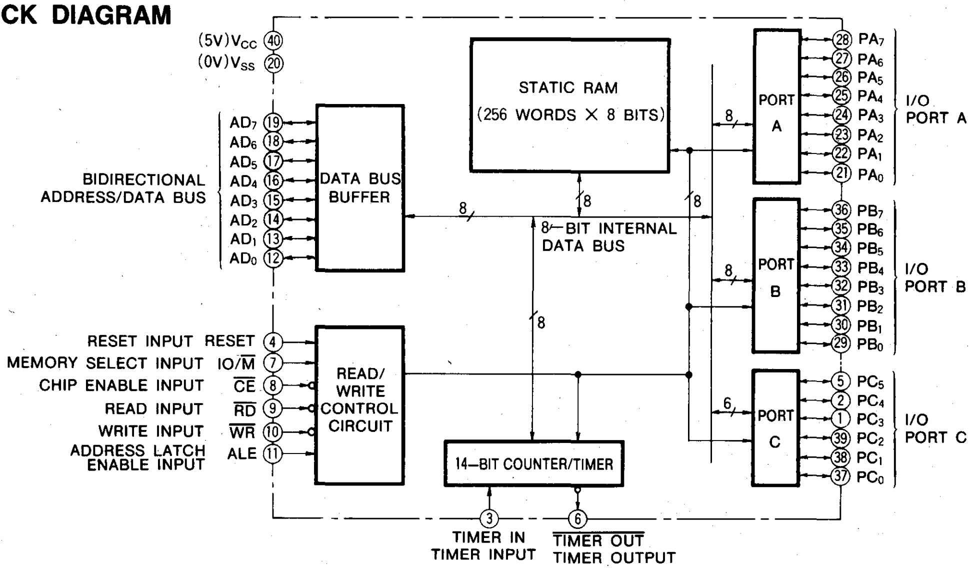
The M5L8155P is composed of RAM, I/O ports and counter/timer. The RAM is a 2K-bit static RAM organized as 256 words by 8 bits. The I/O ports consist of 2 programmable 8-bit ports and 1 programmable 6-bit port. The terminals of the 6-bit port can be programmed to function as control terminals for the 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.

PIN CONFIGURATION (TOP VIEW)



Outline 40P4

BLOCK DIAGRAM



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	$\overline{\text{B STB}}$	(port B strobe)
PC_4	B BF	(port B buffer full)
PC_3	B INTR	(port B interrupt)
PC_2	$\overline{\text{A STB}}$	(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 ($\overline{\text{TIMER OUT}}$)

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 loworder 4 bits (bits 0~3) are used for controlling and determination of 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 Table 2.

Table 2 Bit functions of the command register

Bit	Symbol	Function	
0	PA	PORT A I/O FLAG	1: OUTPUT PORT A 0: INPUT PORT A
1	PB	PORT B I/O FLAG	1: OUTPUT PORT B 0: INPUT PORT B
2	PC_1	PORT C FLAG	00: ALT1 11: ALT2 01: ALT3 10: ALT4
3	PC_2		
4	IEA	PORT A INTERRUPT ENABLE FLAG	1: ENABLE INTERRUPT 0: DISABLE INTERRUPT
5	IEB	PORT B INTERRUPT ENABLE FLAG	1: ENABLE INTERRUPT 0: DISABLE INTERRUPT
6	TM1	COUNTER/TIMER CONTROL 00: NO INFLUENCE ON COUNTER/TIMER OPERATION 01: COUNTER/TIMER OPERATION DISCONTINUED (IF NOT ALREADY STOPPED) 10: COUNTER/TIMER OPERATION DISCONTINUED AFTER THE CURRENT COUNTER/TIMER OPERATION IS COMPLETED 11: COUNTER/TIMER OPERATION STARTED	
7	TM2		

Status Register (7 bits)

The status register is a 7-bit latched register. The loworder 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 Table 3.

Table 3 Bit functions of the status register

Bit	Symbol	Function	
0	INTR A	PORT A INTERRUPT REQUEST	
1	A BF	PORT A BUFFER FULL FLAG	
2	INTE A	PORT A INTERRUPT ENABLE	
3	INTR B	PORT B INTERRUPT REQUEST	
4	B BF	PORT B BUFFER FULL FLAG	
5	INTE B	PORT B INTERRUPT ENABLE	
6	TIMER	COUNTER/TIMER INTERRUPT	(SET TO 1 WHEN THE FINAL LIMIT OF THE COUNTER/TIMER IS REACHED AND IS RESET TO 0 WHEN THE STATUS IS READ)
7	—	THIS BIT IS NOT USED	

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 Table 2.

Port A can be operated in basic or strobe mode and is assigned I/O terminal PA_0~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 Table 2. Port B can be operated in basic or strobe mode and is assigned I/O terminals PB_0~PB_7.

Port C Register (6 bits)

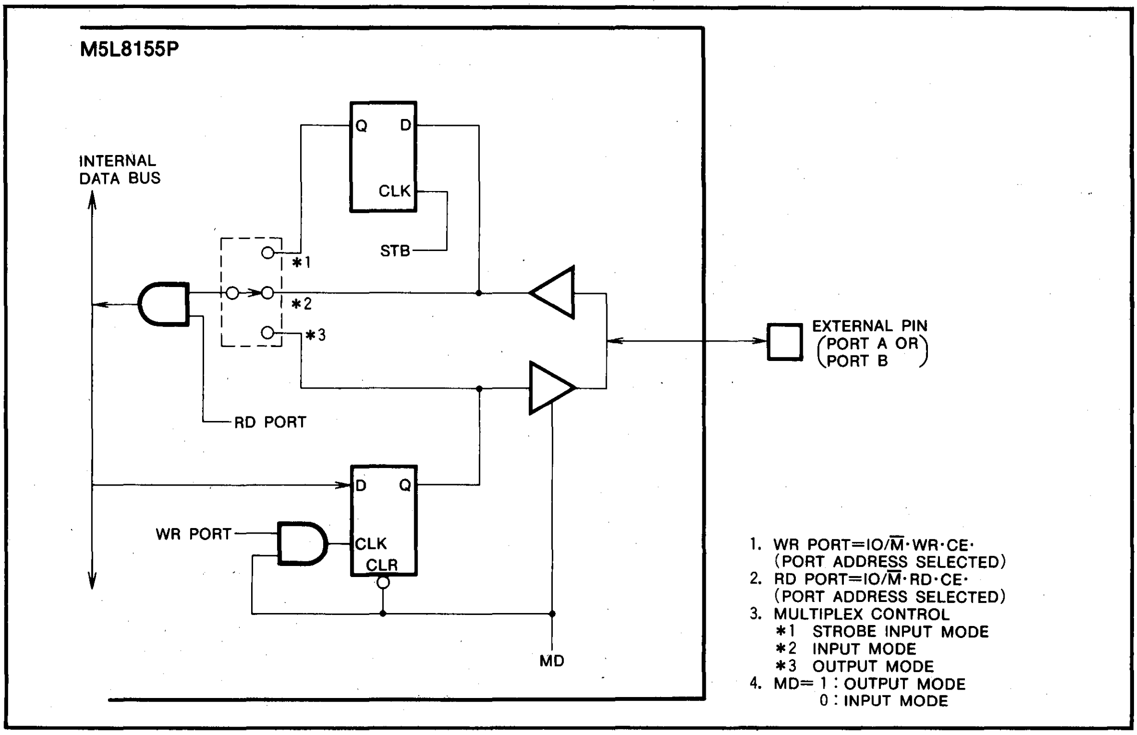
Port C register is assigned address XXXXX011. This port is used for controlling input/output operations of ports A and B by selectively setting bits 2 and 3 of the command register as shown in Table 2. Details of the functions of the various setting of bits 2 and 3 are shown in Table 4. Port C is assigned I/O terminals PC_0 ~ 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.

Table 4 Functions of port C

State Terminal	ALT 1	ALT 2	ALT 3	ALT 4
PC_5	Input	Output	Output	$\overline{\text{B STB}}$ (port B strobe)
PC_4	Input	Output	Output	B BF (port 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)

Configuration of ports

A block diagram of 1 bit of ports A and B is shown in Fig. 1. 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.



1. WR PORT=IO/$\overline{M}$·WR·CE· (PORT ADDRESS SELECTED)
2. RD PORT=IO/$\overline{M}$·RD·CE· (PORT ADDRESS SELECTED)
3. MULTIPLEX CONTROL
   *1 STROBE INPUT MODE
   *2 INPUT MODE
   *3 OUTPUT MODE
4. MD= 1 : OUTPUT MODE
   0 : INPUT MODE

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

Table 5 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 6 Port control signal levels at ALT3 and ALT4

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

The basic functions of the I/O ports are shown in Table 5. The control signal levels to ports A and B, when port C is programmed as a control port, are shown in Table 6.

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

Table 7 Format of counter/timer

Address	Bit Number								Function
	7	6	5	4	3	2	1	0	
XXXXX100	T_7	T_6	T_5	T_4	T_3	T_2	T_1	T_0	THE LOW-ORDER 8 BITS OF THE COUNTER REGISTER
XXXXX101	M_2	M_1	T_{13}	T_{12}	T_{11}	T_{10}	T_9	T_8	M1,M2: TIMER MODE T_8~T_{13}: THE HIGH-ORDER 6 BITS OF THE COUNTER REGISTER

Table 8 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 in mode 0 (mode 1)
1	0	Outputs a low-level pulse during the final count dowm (mode 2)
1	1	Outputs a low-level pulse during each final count down (mode 3)

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 Table 2 for details). The format and timer modes of the counter/timer register are shown in Table 7 and Table 8.

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 Table 2. 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.

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
P_d	Maximum power dissipation	T_a=25℃	1.5	W
T_{opr}	Operating free-air temperature range		0~70	℃
T_{stg}	Storage temperature range		−65~150	℃

RECOMMENDED OPERATING CONDITIONS (T_a=0~70℃, 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 (T_a=0~70℃, 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
$I_{I(\overline{CE})}$	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, T_a=25℃			10	pF
$C_{i/o}$	Input/output terminal capacitance	$V_{I/OL}$=0V, f=1MHz, 25mVrms, T_a=25℃			20	pF
I_{CC}	Supply current from V_{CC}	V_{SS}=0V			180	mA

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

TIMING REQUIREMENTS (T_a=0~70℃, 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

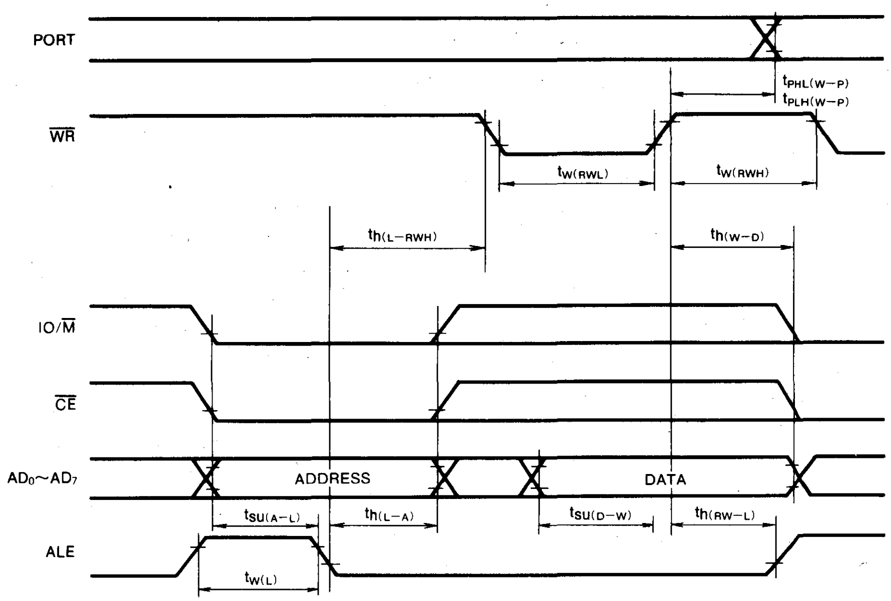
SWITCHING CHARACTERISTICS (T_a=0~70℃, V_{CC}=5V±5%, unless otherwise noted.)

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{PXV(R-DQ)}$	Propagation time from read to data output	t_{RD}				170	ns
$t_{PZX(A-DQ)}$	Propagation time from address to data output	t_{AD}				400	ns
$t_{PVZ(R-DQ)}$	Propagation time from read to data floating (Note 2)	t_{RDF}				100	ns
$t_{PHL(W-P)}$ $t_{PLH(W-P)}$	Propagation time from write to data output	t_{WP} t_{WP}				400	ns
$t_{PLH(STB-BF)}$	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
$t_{PHL(\phi-OUT)}$ $t_{PLH(\phi-OUT)}$	Propagation time from timer input to timer output	t_{TL} t_{TH}				400	ns
$t_{PZX(R-DQ)}$	propagation time from read to data enable	t_{RDE}		10			ns

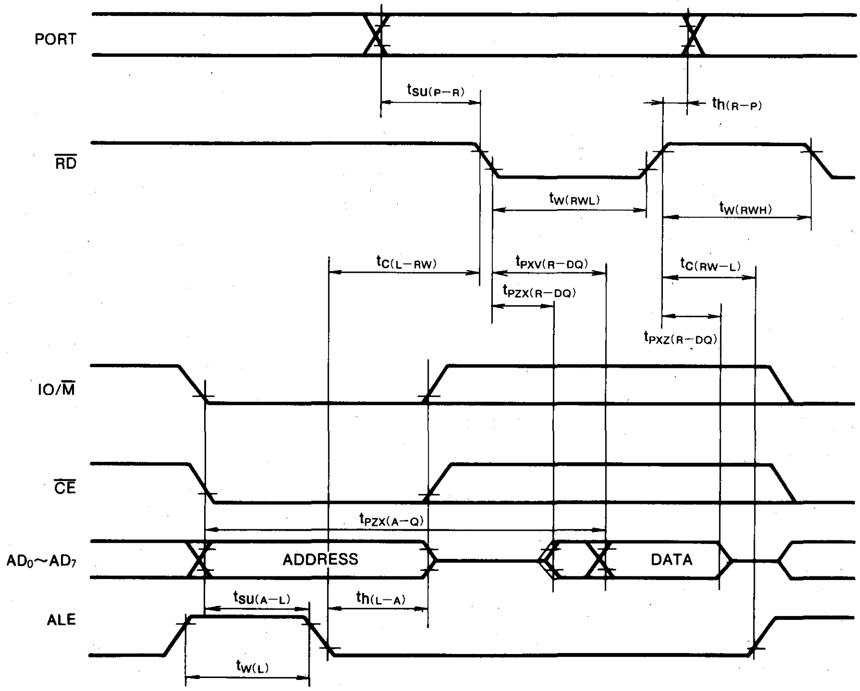
Note 1 : Measurement conditions C=150pF
2 : Measurement conditions of note 1 are not applied.

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

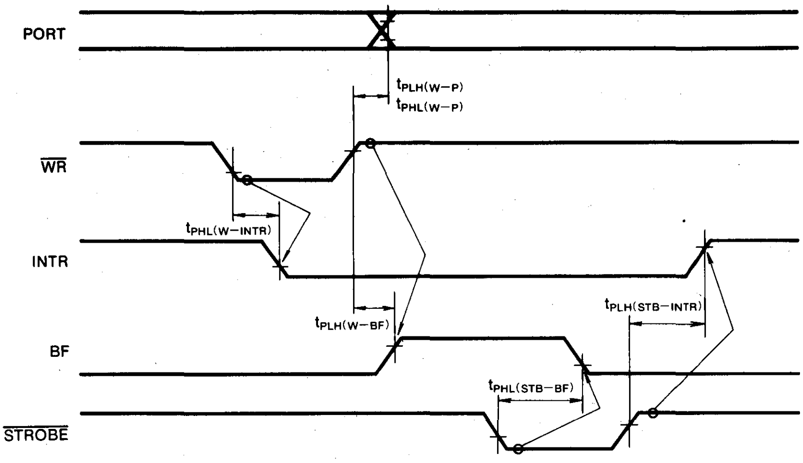
Basic output



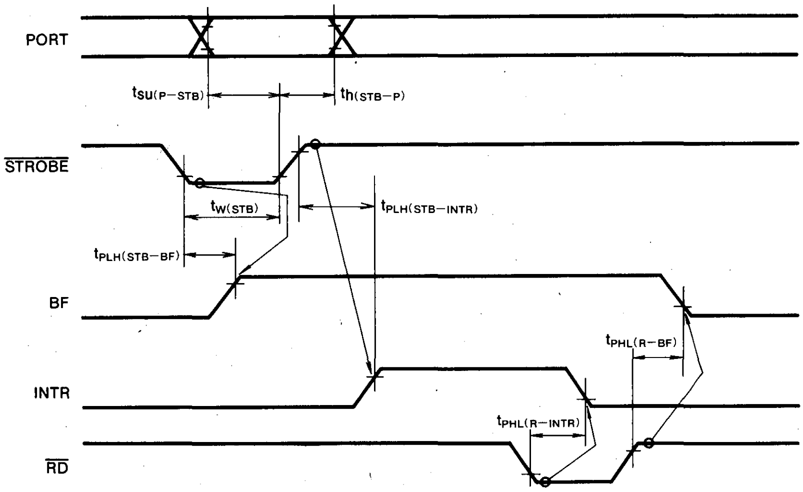
Basic input



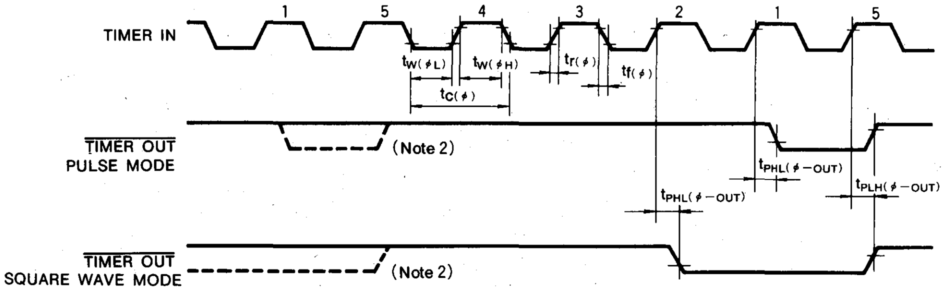
Strobed output



Strobed input



Timer (Note 1)



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

MITSUBISHI MICROCOMPUTERS

M5L8156P

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
- Configuration and electrical characteristics

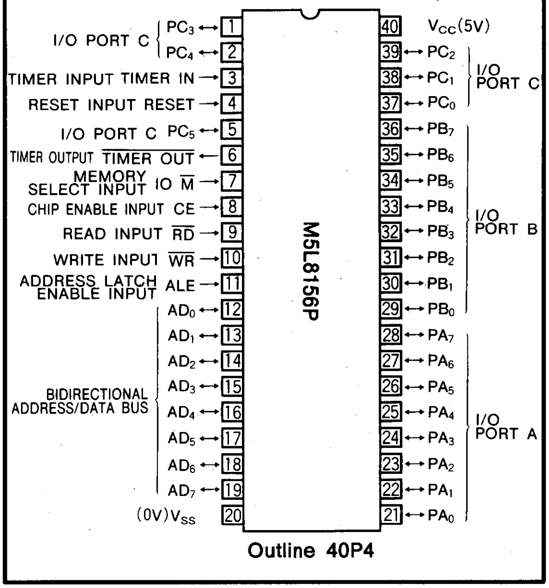
APPLICATION

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

FUNCTION

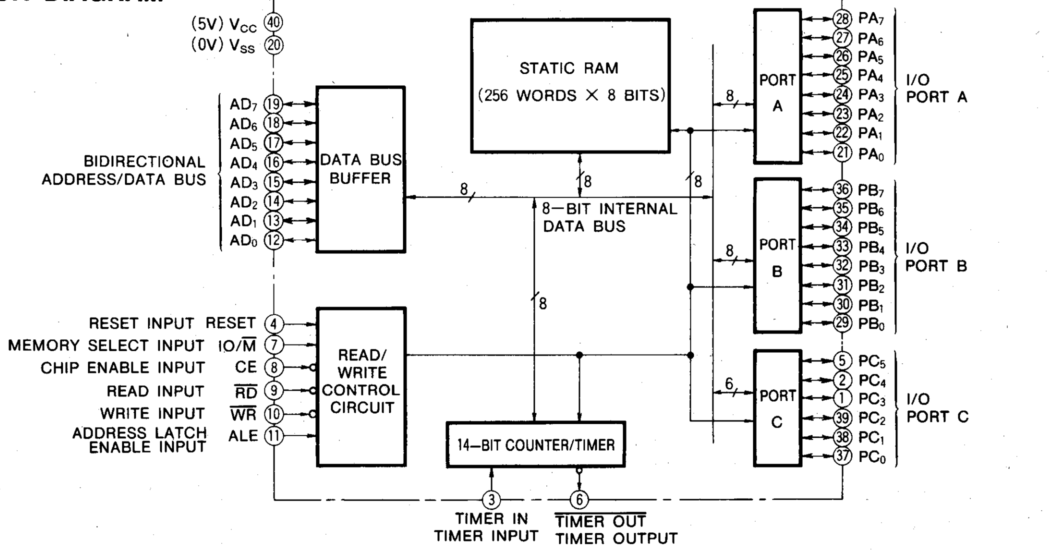
The M5L8156P is composed of RAM, I/O ports and counter/timer. The RAM is a 2K-bit static RAM organized as 256 words by 8 bits. The I/O ports consist of 2 programmable 8-bit ports and 1 programmable 6-bit port. The terminals of the 6-bit port can be programmed to function as control terminals for the 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.

PIN CONFIGURATION (TOP VIEW)



Outline 40P4

BLOCK DIAGRAM



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 ($\overline{\text{RD}}$, $\overline{\text{WR}}$, IO/$\overline{\text{M}}$ and ALE) along with CPU signal ($\overline{\text{CE}}$). RESET signal is also used to control the transfer of data and commands.

Bidirectional Address/Data Bus (AD_0~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/$\overline{\text{M}}$ 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 ($\overline{\text{RD}}$) or write input ($\overline{\text{WR}}$).

Chip Enable Input (CE)

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

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

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

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

When $\overline{\text{XR}}$ is at low-level, the data on the address/data bus are written into RAM if IO/$\overline{\text{M}}$ is at low-level, or if IO/$\overline{\text{M}}$ 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/$\overline{\text{M}}$ are latched in the M5L8156P on the falling edge of ALE.

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

When IO/$\overline{\text{M}}$ 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~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~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~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	$\overline{\text{B STB}}$	(port B strobe)
PC_4	B BF	(port B buffer full)
PC_3	B INTR	(port B interrupt)
PC_2	$\overline{\text{A STB}}$	(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 ($\overline{\text{TIMER OUT}}$)

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 loworder 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 Table 2.

Table 2 Bit functions of the command register

Bit	Symbol	Function	
0	PA	PORT A I/O FLAG	1: OUTPUT PORT A 0: INPUT PORT A
1	PB	PORT B I/O FLAG	1: OUTPUT PORT B 0: INPUT PORT B
2	PC_1	PORT C FLAG	00: ALT1 11: ALT2 01: ALT3 10: ALT4
3	PC_2		
4	IEA	PORT A INTERRUPT ENABLE FLAG	1: ENABLE INTERRUPT 0: DISABLE INTERRUPT
5	IEB	PORT B INTERRUPT ENABLE FLAG	1: ENABLE INTERRUPT 0: DISABLE INTERRUPT
6	TM1	COUNTER/TIMER CONTROL 00: NO INFLUENCE ON COUNTER/TIMER OPERATION 01: COUNTER/TIMER OPERATION DISCONTINUED (IF NOT ALREADY STOPPED) 10: COUNTER/TIMER OPERATION DISCONTINUED AFTER THE CURRENT COUNTER/TIMER OPERATION IS COMPLETED 11: COUNTER/TIMER OPERATION STARTED	
7	TM2		

Status Register (7 bits)

The status register is a 7-bit latched register. The loworder 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 Table 3.

Table 3 Bit functions of the status register

Bit	Symbol	Function
0	INTR A	PORT A INTERRUPT REQUEST
1	A BF	PORT A BUFFER FULL FLAG
2	INTE A	PORT A INTERRUPT ENABLE
3	INTR B	PORT B INTERRUPT REQUEST
4	B BF	PORT B BUFFER FULL FLAG
5	INTE B	PORT B INTERRUPT ENABLE
6	TIMER	COUNTER/TIMER INTERRUPT (SET TO 1 WHEN THE FINAL LIMIT OF THE COUNTER/TIMER IS REACHED AND IS RESET TO 0 WHEN THE STATUS IS READ)
7	—	THIS BIT IS NOT USED

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 Table 2.

Port A can be operated in basic or strobe made and is assigned I/O terminal PA_0~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 Table 2. Port B can be operated in basic or strobe mode and is assigned I/O terminals PB_0~PB_7.

Port C Register (6 bits)

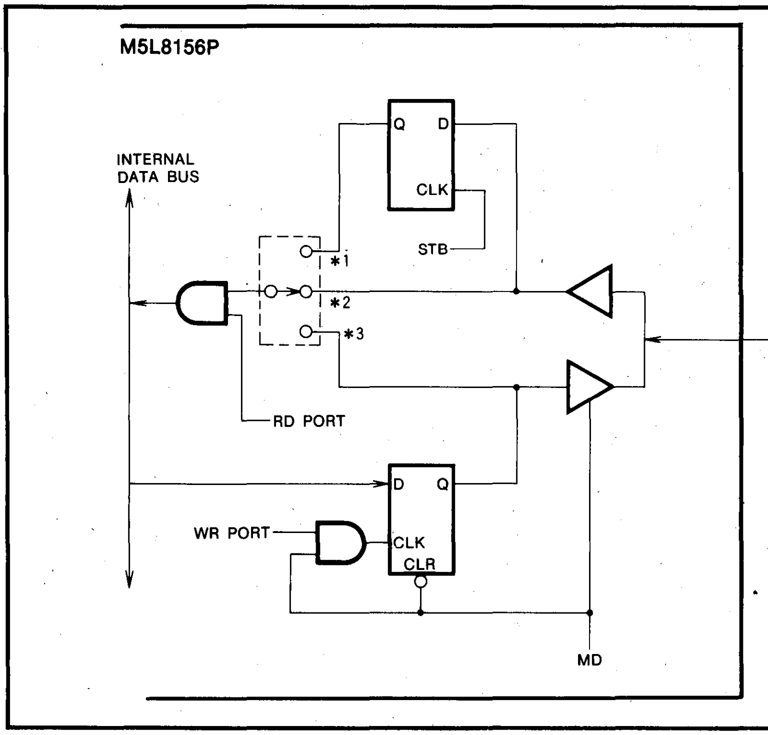
Port C register is assigned address XXXXX011. This port is used for controlling input/output operations of ports A and B by selectively setting bits 2 and 3 of the command register as shown in Table 2. Details of the functions of the various setting of bits 2 and 3 are shown in Table 4. Port C is assigned I/O terminals PC_0 ~ 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.

Table 4 Functions of port C

State Terminal	ALT 1	ALT 2	ALT 3	ALT 4
PC_5	Input	Output	Output	B $\overline{\text{STB}}$ (port B strobe)
PC_4	Input	Output	Output	B BF (port buffer full)
PC_3	Input	Output	Output	B INTR (port B interrupt)
PC_2	Input	Output	A $\overline{\text{STB}}$ (port A strobe)	A $\overline{\text{STB}}$ (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)

Configuration of ports

A block diagram of 1 bit of ports A and B is shown in Fig. 1. 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.



1. WR PORT=IO/$\overline{M}$·WR·CE· (PORT ADDRESS SELECTED)
2. RD PORT=IO/$\overline{M}$·RD·CE· (PORT ADDRESS SELECTED)
3. MULTIPLEX CONTROL
   *1 STROBE INPUT MODE
   *2 INPUT MODE
   *3 OUTPUT MODE
4. MD= 1 : OUTPUT MODE
   0 : INPUT MODE

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

Table 5 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 6 Port control signal levels at ALT3 and ALT4

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

The basic functions of the I/O ports are shown in Table 5. The control signal levels to ports A and B, when port C is programmed as a control port, are shown in Table 6.

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

Table 7 Format of counter/timer

Address	Bit Number								Function
	7	6	5	4	3	2	1	0	
XXXXX100	T_7	T_6	T_5	T_4	T_3	T_2	T_1	T_0	THE LOW-ORDER 8 BITS OF THE COUNTER REGISTER
XXXXX101	M_2	M_1	T_{13}	T_{12}	T_{11}	T_{10}	T_9	T_8	M1,M2: TIMER MODE T_8~T_{13}: THE HIGH-ORDER 6 BITS OF THE COUNTER REGISTER

Table 8 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 dowm (mode 2)
1	1	Outputs a low-level pulse during each final count down (mode 3)

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 Table 2 for details). The format and timer modes of the counter/timer register are shown in Table 7 and Table 8.

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 Table 2. 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.

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
P_d	Maximum power dissipation	T_a=25℃	1.5	W
T_{opr}	Operating free-air temperature range		0~70	℃
T_{stg}	Storage temperature range		−65~150	℃

RECOMMENDED OPERATING CONDITIONS (T_a=0~70℃, 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 (T_a=0~70℃, 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, T_a=25℃			10	pF
$C_{i/o}$	Input/output terminal capacitance	$V_{I/OL}$=0V, f=1MHz, 25mVrms, T_a=25℃			20	pF
I_{CC}	Supply current from V_{CC}	V_{SS}=0V			180	mA

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

TIMING REQUIREMENTS (T_a=0~70℃, 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

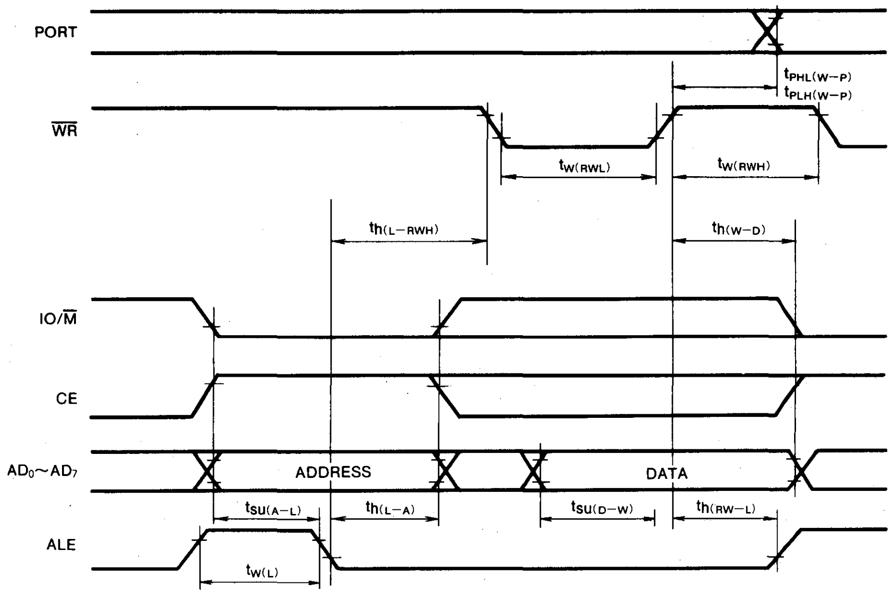
SWITCHING CHARACTERISTICS (T_a=0~70℃, V_{CC}=5V±5%, unless otherwise noted.)

Symbol	Parameter	Alternative symbol	Test conditions	Limits			Unit
				Min	Typ	Max	
$t_{PXV(R-DQ)}$	Propagation time from read to data output	t_{RD}				170	ns
$t_{PZX(A-DQ)}$	Propagation time from address to data output	t_{AD}				400	ns
$t_{PVZ(R-DQ)}$	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}					
$t_{PZX(R-DQ)}$	propagation time from read to data enable	t_{RDE}		10			ns

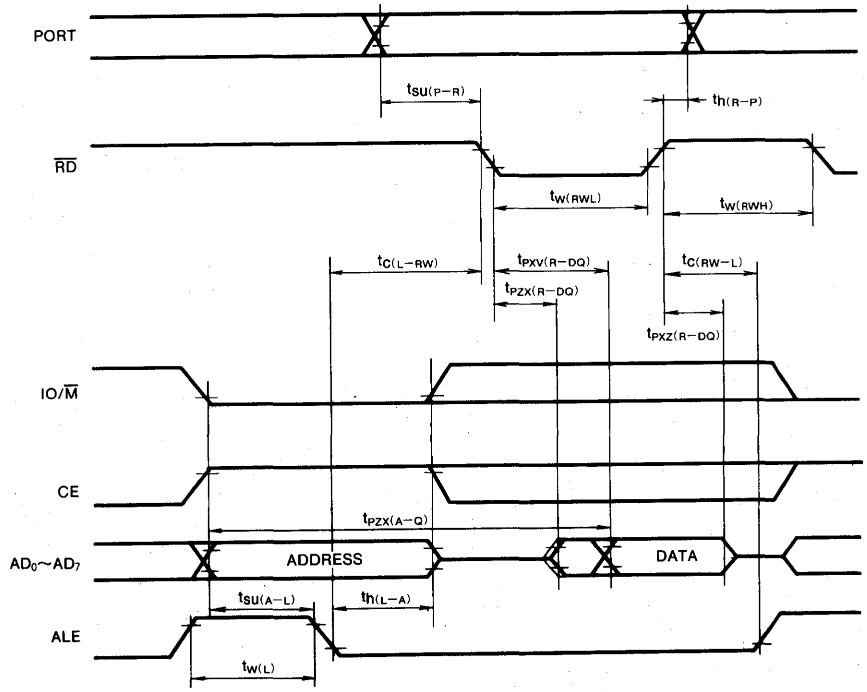
Note 1 : Measurement conditions C=150pF
2 : Measurement conditions of note 6 are not applied.

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

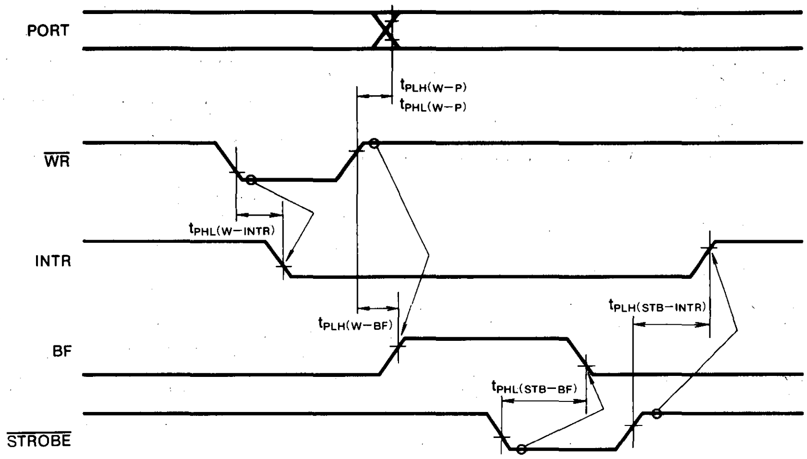
Basic output



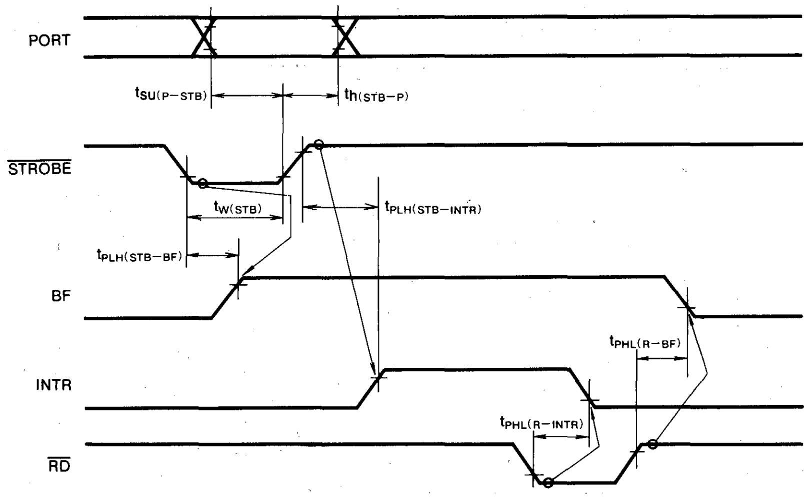
Basic input



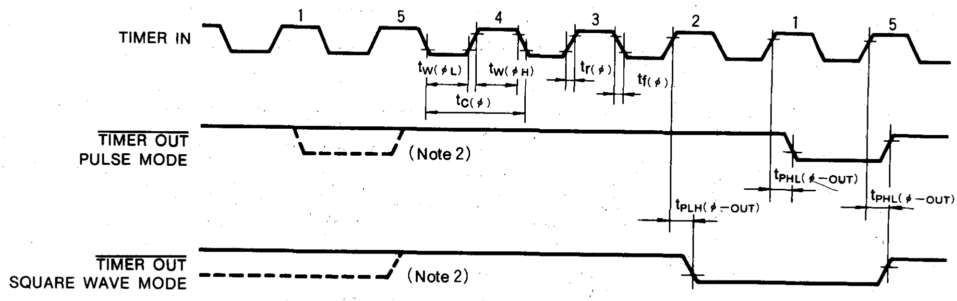
Strobed output



Strobed input



Timer (Note 1)



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

MICROCOMPUTER SUPPORT SYSTEMS

PC4000

DEBUGGING MACHINE

DESCRIPTION

The PC4000 is a debugging machine for use with single-chip microcomputers. It is intended for use as a general purpose debugging machine for support of single-chip microcomputer hardware and software.

FEATURES

- Usable for RAM-based program debugging
- Connectable to the user system via a DIL socket or connector
- Built-in EPROM (2716, 2732) writer function
- Uses serial data transfer for two-way data transfer with the host machine (e.g. PC9000 cross assembler machine)
- Usable with a variety of single-chip microcomputers by simply replacing a single board
- Print out of internal memory contents is possible by means of an external printer
- Easy-to-carry-about in its compact case, provided with an angle stand

APPLICATIONS

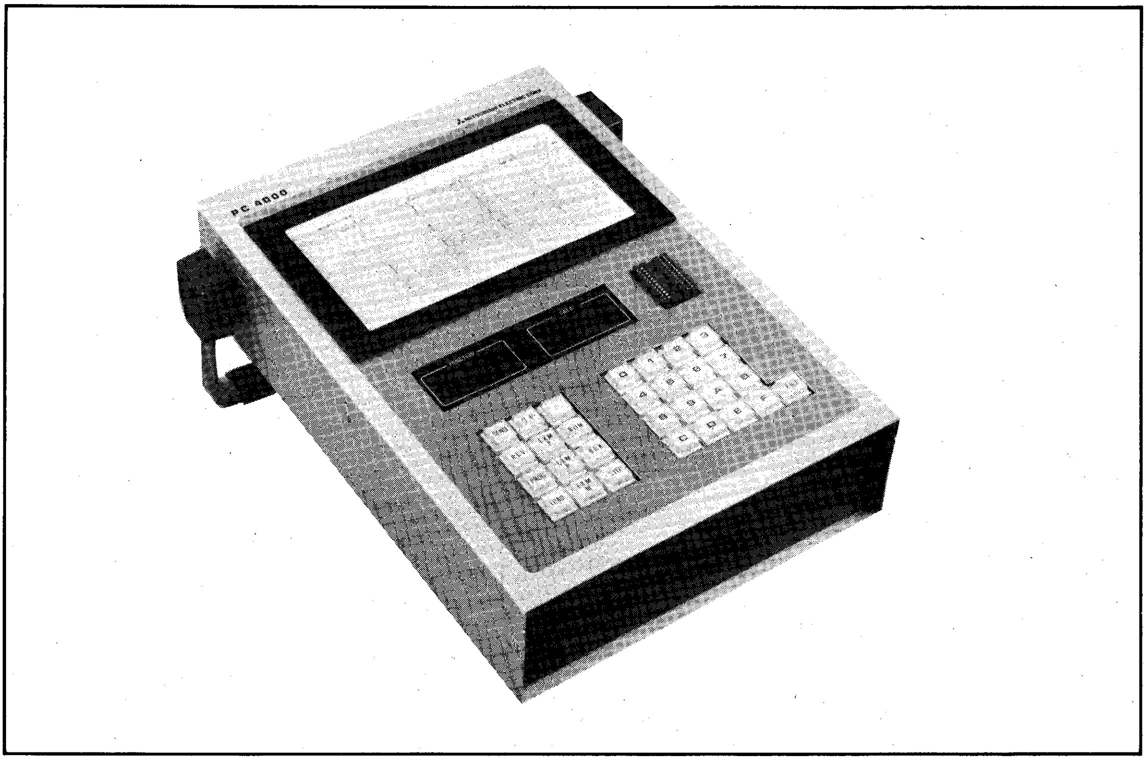
Hardware and software development and program debugging for single-chip microcomputer systems.

CONFIGURATION

The PC4000, as shown in the block diagram, consists of the following hardware elements.

(1) M5L8085AP monitor CPU
(2) Serial data input/output interface circuit
(3) EPROM writer circuit
(4) Program RAM (10 bits x 4K)
(5) Keyboard and LED display circuits
(6) Power supply

The PC4000 is used in conjunction with a dedicated board which allows interface of the PC4000 with the object microcomputer under development. The dedicated board insertion access window is located on the right side of the PC4000. In addition, each dedicated board stores the control program for the monitor CPU. Therefore, when the microcomputer type is changed, the PC4000 can be modified to suit the new type by merely changing the single dedicated board.



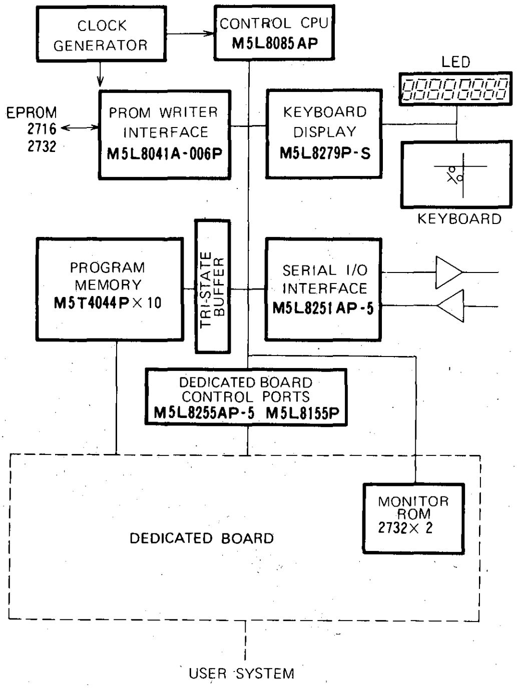
FUNCTIONAL DESCRIPTION

Object programs developed on such devices as the PC9000 cross assembler machine are sent to the PC4000 via the serial input/output interface. The serial data transmission rate can be selected from 1200bps to 9600bps and the interface is a 20mA current loop type. The transmission format is Intel-compatible hexadecimal.

The data in the program memory is executed by the evaluation CPU on the dedicated board. In addition, this memory contents can be written into 2716 or 2732 EPROM devices or data can be read out of such devices via a 24-pin DIL socket.

The keyboard consists of 12 function keys and 16 numerical keys as well as a single entry key. The LED display is an 8-digit display of 7-segment LED elements used to display data for reference while processing is performed.

BLOCK DIAGRAM


CLOCK GENERATOR
CONTROL CPU M5L8085AP
LED
EPROM 2716 2732
PROM WRITER INTERFACE M5L8041A-006P
KEYBOARD DISPLAY M5L8279P-S
KEYBOARD
PROGRAM MEMORY M5T4044P×10
TRI-STATE BUFFER
SERIAL I/O INTERFACE M5L8251AP-5
DEDICATED BOARD CONTROL PORTS M5L8255AP-5 M5L8155P
MONITOR ROM 2732×2
DEDICATED BOARD
USER SYSTEM

DEBUGGING MACHINE

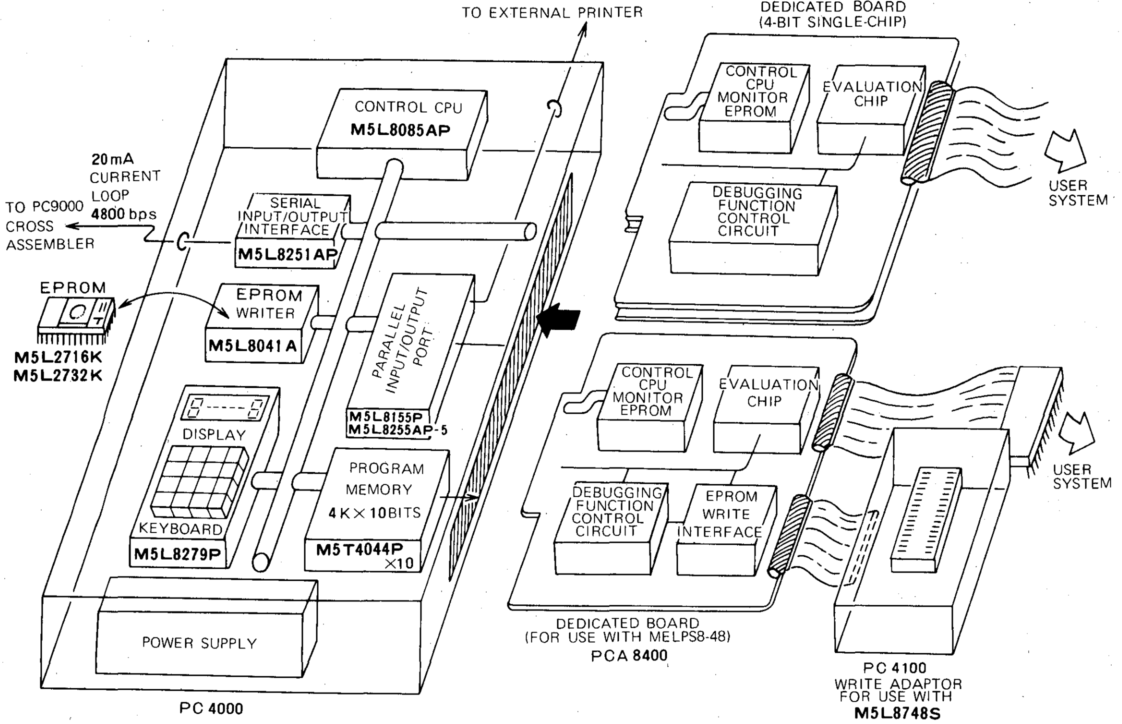
KEY FUNCTIONS (BASIC FUNCTIONS ONLY)

Symbol	Name	Function
SEND	Data transmit key	Converts program memory data to serial data and transmits to an external device
RCV	Data receive key	Receives serial data and writes this data into program memory
PROG	(EPROM) Program key	Writes program memory data into the EPROM inserted in the socket
LOAD	(EPROM) Load key	Sends data from the EPROM inserted in the socket to program memory
PRT	Print key	Data transmit to the optional printer
EXM P	Examine program memory key	Verification/correction of program memory contents
EXM R	Examine register key	Verification/correction of register contents
EXM M	Examine memory key	Verification/correction of RAM contents
RES	Reset key	Reset of program counter
RUN	Run (execute) key	Re-start of program execution at the specified address (real time)
BRK	Break point set key	Sets the break point address
STEP	Single step key	Excutes the program one step at a time
0～F	Numerical keys	Used for input of address and data
ENT	Entry key	Effectively enters input numerical data

SPECIFICATIONS

Item	Specification
Method	The system is used with a dedicated board which includes the evaluation chip to perform in-circuit emulation
Applicable microcomputers	M58840-XXXP M58494-XXXP M58496-XXXP M5L8048-XXXP M5L8049-XXXP and all other Mitsubishi single-chip microcomputers
Program RAM	Built-in, 4K x 10 bits (250ns access time)
Control CPU	M5L8085AP
Built-in EPROM writer circuit	Usable with 2716 or 2732 devices
Display	7-segment LED, 8 digits
Input	Key switches: Commands: 12 keys Numerical: 16 keys Entry: 1 key
Interface	① 20mA current loop serial input/output interface 4800bps, full deplex, one line (Selectable from 1200 to 9600bps) ② Centronix-compatible parallel interface, one line
Monitor function	Monitor programs for the appropriate object microcomputers are written into the two M5L2732K devices mounted on the dedicated board. Basic Functions • Transfer of RAM data with an external system • Read and write of EPROM data • Verification/correction of the built-in program memory (RAM) contents • Execution and halt at any arbitrary program address • Single-step execution of programs • Verification/correction of internal registers, memory, flags
User system connection	Input/output connections to the dedicated board by means of a cable
Dimensions	364 × 257 × 85 mm (excluding handle and key switch tops)
Power supply	AC 100V 100VA
Operating temperature	5～40°C
Storage temperature	−20～60°C

PC4000 CONFIGURATION



PC9000

CROSS ASSEMBLER MACHINE

DESCRIPTION

The PC9000 is a cross assembler machine. It is capable of converting programs for the Mitsubishi single-chip microcomputers written in assembler language to machine language. In addition, it can perform such debugging functions as disassembly and act as an EPROM writer.

FEATURES

- Input of the source program from the keyboard
- An efficient screen editor allows editing of source programs
- Program dump and load to the mini-floppy disk
- Object data write/read for 2708, 2716 and 2732 EPROM devices
- Listing using a Centronix-compatible printer is possible
- Data transmission is possible to the PC4000 debugging machine
- Usable with all types of Mitsubishi single-chip microcomputers
- Compact, desk-top design

APPLICATION

Software development support for Mitsubishi single-chip microcomputers.

FUNCTION

The PC9000 as shown in the configuration diagram consists of the following hardware

(1) Control CPU and bootstrap ROM
(2) 48K byte RAM
(3) 2K byte display screen RAM
(4) 9-inch CRT display circuit
(5) EPROM writer circuit
(6) ASCII keyboard
(7) Hardcopy output by means of an internal mini-printer circuit or an external printer interface circuit
(8) Floppy disk controller (two mini floppy disk drives)
(9) Parallel input/output interface circuit (two lines)
(10) Power supply

An M5L8085AP is used as the control CPU. The keyboard, CRT, mini-floppy disk drives, and printer interfaces are connected by means of a bus line. The keyboard is used for input of commands to the monitor and source program data verification. The 9-inch green CRT display screen is capable of displaying 24 lines of 80 characters. As a printer a 20 column mini-printer is built-in to the PC9000 in addition to the ability to use an 80 column printer having Centronix compatibility via an interface which is available. The built-in mini-printer may be used to output

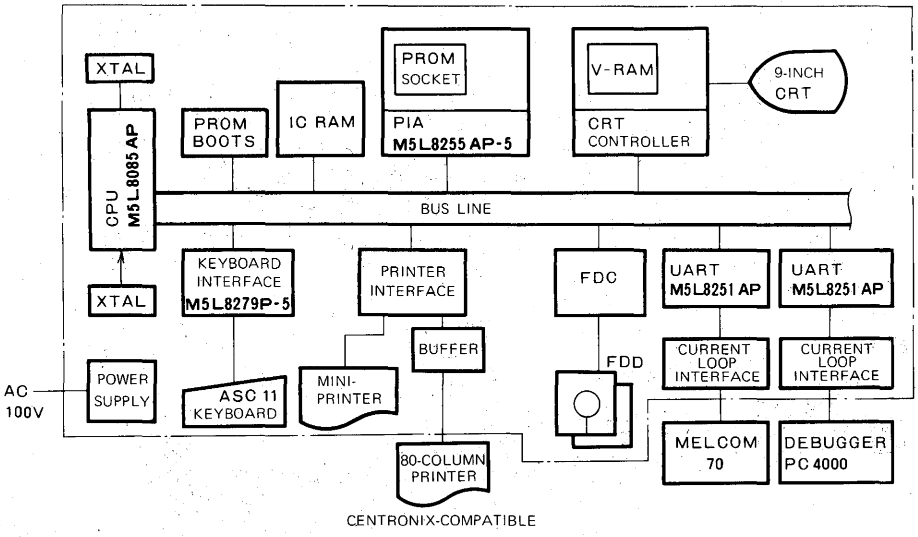
the disassembly results while the external printer may be used to output the assembly listing as well as disassembly listing.

FUNCTIONAL DESCRIPTION

The PC9000 contains the assembler, disassembler, source editor, and EPROM writer functions required for software support of microcomputers. These functions are summarized in the Table.

Function	Effect	Applicable devices
Assemble	Source input: keyboard output: printer, EPROM, data transfer (with debugging unit)	All 4-bit single-chip PMOS, and CMOS microcomputers M5L8048, M5L8049 and M5L8041A 8-bit single-chip microcomputers
Disassemble	Disassembly of the specified file Output: 20 column printer, external printer	Same as above
Source editor	Deletion, insertion, modification, character search, and screen editing	Same as above
PROM writer	EPROM erase check, write, verification, read	M5L2708K, M5L2716K, M5L2732K

PC9000 CONFIGURATION

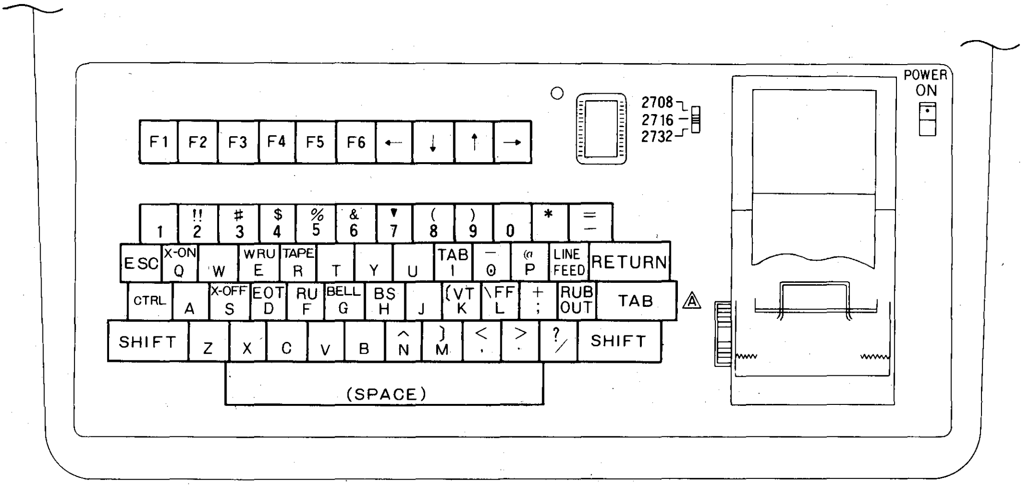


CROSS ASSEMBLER MACHINE

SPECIFICATIONS

Item	Specification
Structure	Desktop-type, single cabinet
CPU	Mitsubishi M5L8085AP (2.45 MHz clock)
IC memory	2K byte ROM (bootstrap area), 48K-byte DRAM, 2K-byte VRAM
Memory device	Mini floppy disk x 2 drives, double-sided, double-density
Display	9-inch green CRT display, 80 lines x 25 characters
Keyboard	Modified ASCII specifications, 2-key lockout
Dedicated printer	5 x 7 dot Matrix thermal printer, 20 columns. 2 lines/s. Paper width: 60mm.
Printer interface	Centronix, parallel interface Interface connector: 36-pin DDK Amphenol
Serial input/output interface	20mA current loop (2 lines)
Data transfer format	MELPS 85 Hexadecimal (equivalent to Intel Hexadecimal)
Applicable microcomputers	MELPS 8−48 (M5L8048-XXXP, M5L8049-XXXP and others) MELPS 4 (M58840-XXXP and others) MELPS 41 (M58494-XXXP) MELPS 42 (M58496-XXXP and others)
Outer dimensions and weight	Desk top-type 470(W) x 290(H) x 490(D), 17kg
Power supply	AC 100V ± 10% 50/60Hz

KEYBOARD ARRANGEMENT


F1 F2 F3 F4 F5 F6 ← ↓ ↑ →
2708
2716
2732
POWER ON
1 !! 2 # 3 $ 4 % 5 & 6 ▼ 7 (8) 9 0 * = −
ESC X-ON Q W WRU E TAPE R T Y U TAB I − 0 @ P LINE FEED RETURN
CTRL A X-OFF S EOT D RU F BELL G BS H J { VT K \ FF L + ; RUB OUT TAB
A
SHIFT Z X C V B ^ N } M < , > . ? / SHIFT
(SPACE)

PC9100

16-BIT CPU SOFTWARE DEVELOPMENT SYSTEM

DESCRIPTION

The PC9100 is a software development support for the MELPS 86 and PCA8600 series. As this has floppy disk drives and a built-in EPROM programmer, a software development support for the MELPS 86 is performed by only connecting a CRT terminal on the market.

The operation system (OS) adopts the propagated ®CP/M-86$^{T.M.}$ and CP/M-80®, therefore many softwares programmed for CP/M are usable.

In addition, with a use of M5L8086S in-circuit emulator PC9110, the operation from a source programming of MELPS 86 to a hardware debugging is executed continuously.

FEATURES

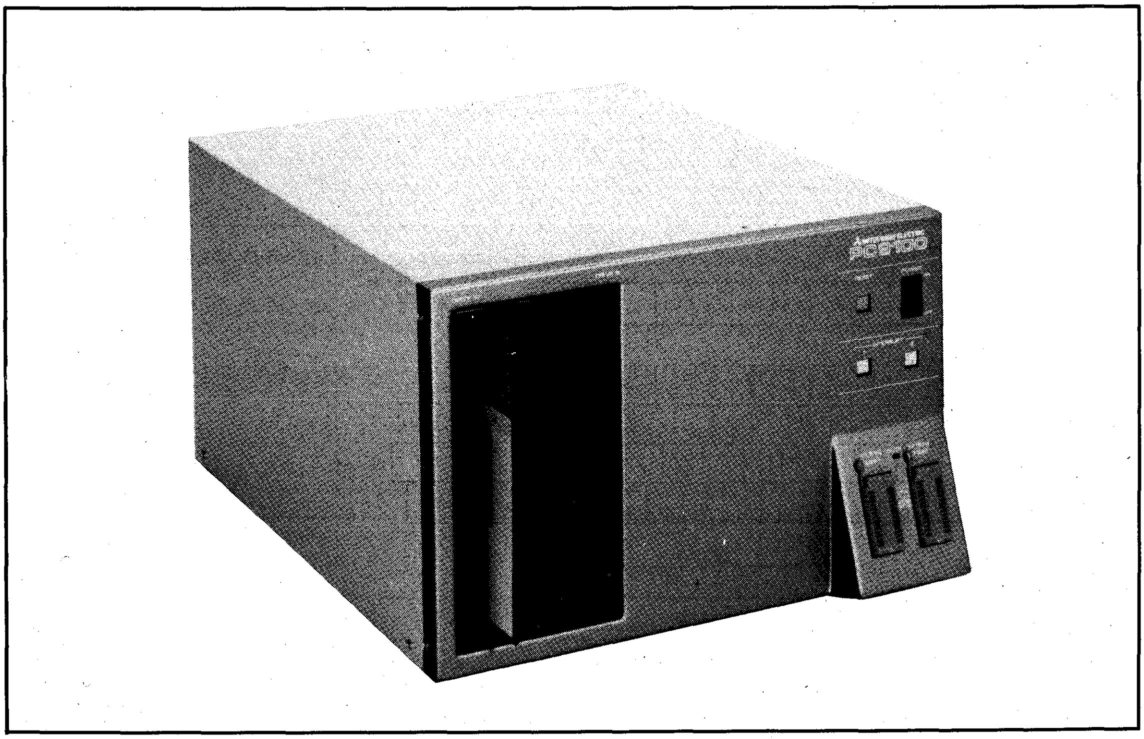
- Compatible with PCA8601 monitor program
- Adoption of CP/M-86, CP/M-80 for OS. Select any OS by a monitor program.
- Capability of executing a developed software for 8080A/8085A and of developing an 8085A software when the CP/M-80 is selected.
- Easy system expansion caused by system bus which adopts IEEE-796 bus and 4 opened card cages.
- Standard 2 drives for double-sided double-density floppy disk (Capacity of 1.2M bytes).
- Applicable drive for double-sided double-density, double-sided single-density, single-sided single-density disks by the switch on the rear panel.
- Capability of programming 2 of 16-bit data at a time by a built-in EPROM programmer (corresponds to M5L2716K, M5L2732K, M5L2764K)
- Usable CRT device, which contains RS232C serial interface, on the market
- Capability of connecting a centronics printer
- Capability of connecting the M5L8086S in-circuit emulator PC9110
- One built-in RS232C serial interface for general purpose
- RAM size .... 72K bytes
- AC 100V
- Compact, light weight

APPLICATION

- Software development support for MELPS 86
- Software development support for MELPS 85
- Personal computer
- Device for data analysis and management
- Base machine for each dedicated system

Note 1. Ⓡ CP/M is a registered trade-mark of Degital Research Inc.
2. IEEE-796 bus is a system bus for microcomputers which are the standardized Intel Ⓡ multi bus by IEEE.

Ⓡ multi bus is a registered trade-mark of Intel.



16-BIT CPU SOFTWARE DEVELOPMENT SYSTEM

FUNCTION

The PC9100 adopts multi bus as a system and mounts a PCA8601 CPU board, PCA8602 64K RAM board, PCA8603 floppy disk controller board (including M5L8085A) of PCA 8600 series and an I/O board for PC9100. 4 multi bus compatible boards are in the card cage slot for easy system's expansion and exclusive.

2 built-in double-sided double-density floppy disks are provided. The second drive (Drive B:) can select either double-sided single-density or single-sided single-density by the switch on the rear panel. After selecting either single density, the CP/M lets the BIOS work automatically by accessing each Drive as C:, D:.

The EPROM programmer consists of 2 DIL sockets for 28-pin and 2 EPROMs are programmed in a 16-bit microcomputer object. Programmable EPROMs are M5L2716K, M5L2732K, M5L2732A type and M5L2764K.

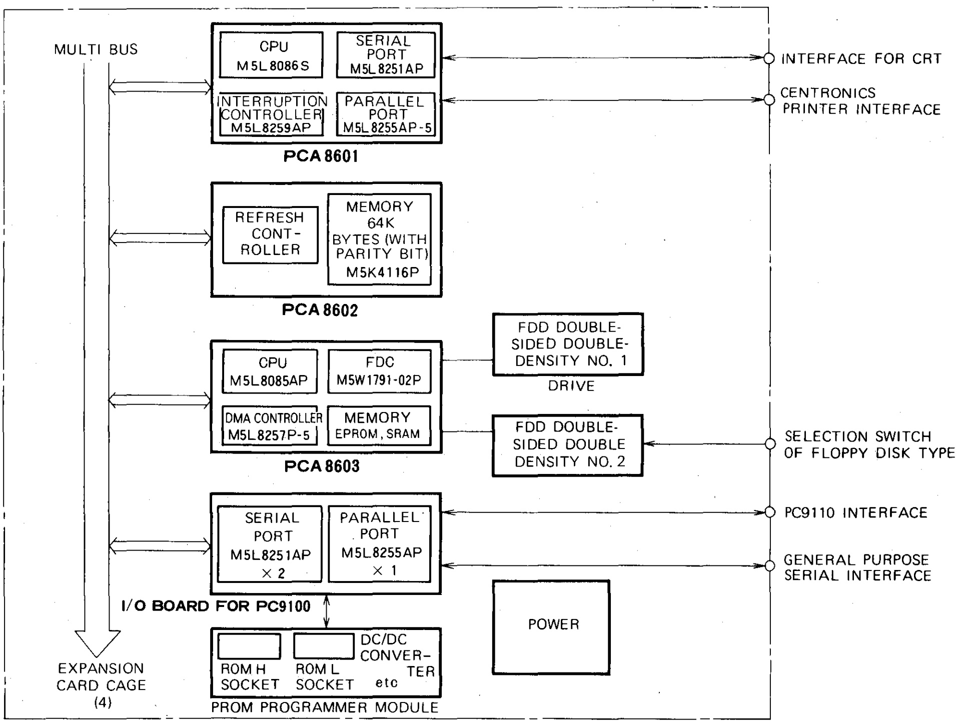
I/O functions are: (1) CRT interface (RS232C standard, Serial operation, 25-pin connector) (2) Printer interface (TTL level, Centronics 36-pin champ connector) (3) PC9110 interface (RS232C standard, Serial operation, 25-pin connector) (4) General purpose serial interface (RS 232C standard, Assignation to RDR: PUN: In CP/M, 25-pin connector)

After turning on the PC9100, the M5L8086S on the PCA8601 will be a bus master and executes a monitor program written in the ROM on the PCA8601. After this, if the G(GO) command is executed, the system will be in CP/M-86 or CP/M-80 mode. When it is the CP/M-86 mode, the system works considering the M5L8086S as master, and commands consist of the CP/M-86, 8086 assembler, 8086 debugger and application software for CP/M-86 which is on the market are able to execute.

When it is the CP/M-80 mode, the commands consist of the CP/M-80, 8080 assembler, 8080 debugger and application software for the CP/M-80 which is on the market are able to execute.

The operation from the OS to the monitor is executed by the reset switch.

BLOCK DIAGRAM



16-BIT CPU SOFTWARE DEVELOPMENT SYSTEM

FUNCTION EXPLANATION (According to CP/M, CP/M86, Mitsubishi original utility S/W)

Function	Effects	Application devices
Assembler	Source input: Source file created by key board input output: List, EPROM, Data communication with emulator	● 8086 ● 8085A
Debugger	Software debugger: Supporter in CP/M-86 For Application S/W in CP/M-86	● 8086
	Software debugger: Supporter in CP/M-80 For Application S/W in CP/M-85	● 8085A
RPOM programmer	Write, read, verification of EPROM (Capability of programming upper bytes and lower bytes at the same time for 8086)	M5L2716K M5L2732K M5L2764K
Data communication	Bi-directional data communication with an external device in file base	RS232C standard
Editor	Supporter in CP/M	Applicable for all uses
Execution of application S/W	Rich applicable S/W which on the market eg. "CIS COBOL86", "PASCAL/M-86" etc. which are on the market as the high quality language for 8086 can be executed in the PC9100. Likewise, the high quality editor (Word star etc.) on the market can be used beside the standard editor.	● 8086 ● 8085A

SPECIFICATION

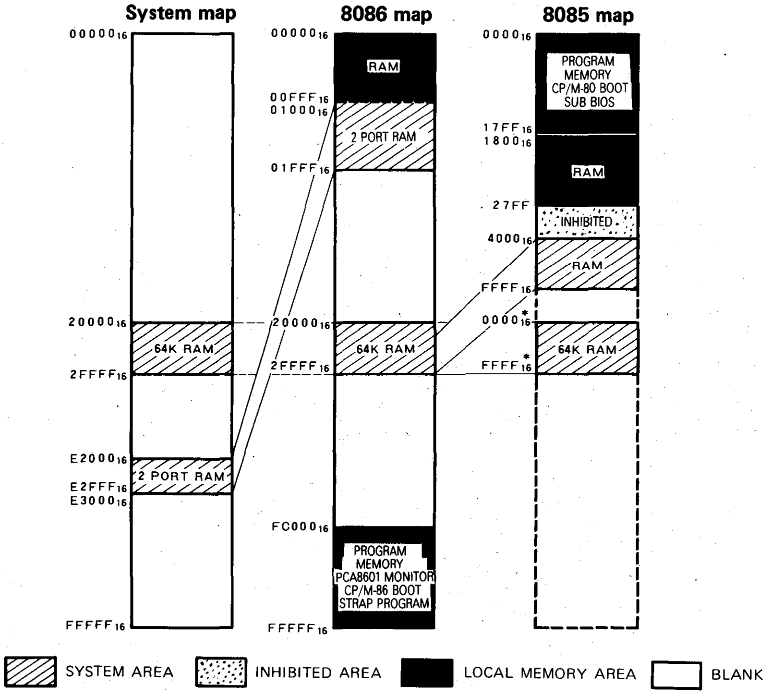
Hardware Specification

Item	Specification
Structure	Desk top, single cabinet (External CRT, Keyboard, printer)
CPU	Mitsubishi M5L8086S 4.9152MHz Mitsubishi M5L8085AP 2.4576MHz
IC memory	PCA8601 Program memory 16K bytes RAM 16K bytes PCA8602 RAM 64K bytes PCA8603 Program memory 4K bytes RAM 2K bytes
Memory device	2 double-sided double-density floppy disks (Single-sided single density is also used by setting the switch on the rear panel)
CRT interface	Serial interface (The electrical characteristics and connector are compatible with the RS232C standard) Transmit speed . . . . . 9600BPS standard
Printer interface	Centronics parallel interface (The DDK unphenol 36-pin is used for a connector)
Interface for PC9110 in-circuit emulator controller	Serial interface (The electrical characteristics and connector are compatible with the RS232C standard) Transmit speed 9600BPS
General purpose serial I/O interface	One port (The electrical characteristics and connector are compatible with the RS232C standard) Transmit speed . . . . . Selectable from 1200/2400/4800/9600BPS 9600BPS standard
PROM programming device	2 multi devices Programmable into M5L2764K, M5L2732A type, M5L2732K, M5L2716K
Applicable microcomputer	MELPS 86, MELPS 85
Capable expansion area	4 of multi bus boards
Outer dimensions	Desk top type 420(W) x 450(D) x 260(H) mm
Operating temperature	5°C ~ 40°C

Software Specification

Item	Specification
Monitor program	Monitor for 8086 which is compatible with the one on the PCA8081
Operation system	CP/M-86 General purpose O.S. for 8086 of Degital Research Inc. CP/M-80 General purpose O.S. for 8080 of Degital Research Inc.
Utility software (8086 base)	ASM86. CMD 8086 Assembler DDT86. CMD CP/M86 Debugger STAT. CMD File status utility SUBMIT. CMD Batch management utility PIP. CMD File exchange utility GENCMD. CMD CMD file generating utility ED. CMD Editor for program generator PROM. CMD PROM programmer controller program DDFMT Disk initialization and disk copy
Utility software (8080 base)	ASM. COM 8080 Assembler DDT. COM CP/M80 Debugger STAT. COM File status utility SUBMIT. COM Batch management utility XSUB. COM Expansion of SUBMIT. COM utility PIP. COM File exchange utility LOAD. COM COM file generating utility ED. COM Editor for program generator DUMP. COM Hex dump utility MOVCPM. COM
Library	BIOS.A86、 DEBLOCK.LIB、 BIOS.ASM、 DUMP.ASM

Memory Map



Note 1. Broken line means that addresses are the same between each area.
Solid line means that each area has the different address from others.
2. The area marked with * will be changed by an installation of a control register.

PC9001CPM

CP/M SOFTWARE FOR PC9000

DESCRIPTION

The PC9001CPM is an option board to turn the cross assemble machine PC9000 to a CP/M® machine.

This is capable of developing the 8085 software and executions of a software for a CP/M and a user program with the PC9000. The CP/M of PC9001 is designed for 48K bytes CP/M Version 2.2.

This software must be used following the software contract.

Note: ® CP/M is a registered trade-mark of Digital Research Inc.

FEATURES

- Execution of a CP/M on the PC9000
- Functions of a debugger and an assembler of 8080A
- Application program which operates on a CP/M of which memory capacitor is below 48K
- Developing and operating of a user program on the PC9000
- Opened internal miniprinter as a user list device
- Selectable letters (CAPITAL/small) by F3 key
- Usable internal serial interface as RDR:, PUN:
- Functions of a standard CP/M editor and a PC9000 original screen editor
- Used for a high quality EPROM programmer by an internal EPROM programmer and an option control program

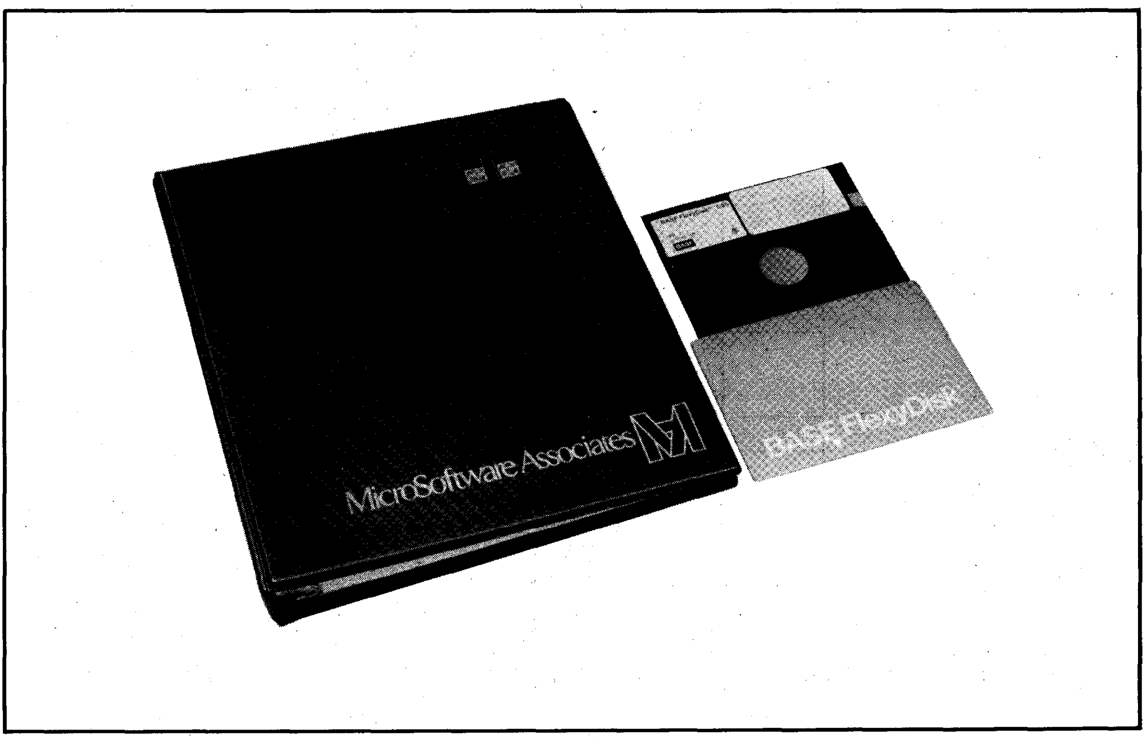
APPLICATIONS

- Supporting develop machine for MELPS 85
- Personal computer
- Data communication terminal

FUNCTIONS

The PC9001CPM contains the EPROM (M5L2716K or M5L2732K) that programs a boot strap program and a basic operation system (BIOS) for CP/M. A floppy disk programmed of a CP/M and an application software, and manuals are also packed in the PC9001 CP/M carton.

Following the manual, exchange the EPROM on the PC9000 main board. The CP/M then initiates when a system starts by the disk. The PC9000 original disk and prescribed disk are compatible after changing a ROM.



PC9001CPM

CP/M SOFTWARE FOR PC9000

SPECIFICATION

Item	Specification
CP/M Version	Version 2.2
RAM size	48K bytes
Console (CON:)	Built-in CRT device and keyboard of the PC9000
Paper tape reader (RDR:)	Built-in MELCOM 70 interface of the PC9000 (Hand shaking operation using a data terminal ready. DTR)
Paper tape punch (PUN:)	Built-in PC4000 interface of the PC9000 (Hand shaking operation using a data set ready. DSR)
List device (LST:)	Centronics external printer (LPT:) Built-in miniprinter of the PC9000 (UL1:) LPT: = standard assign
Floppy disk storage	2 x 320K bytes (A: and B:)
Key board	● Capital/small letter selection by F3 key
PROM programmer	● File load, file save, edit, PROM program, PROM read, verify, padding, block moving ● 8K bytes work area ● Target ROM (2708/2716/2732)
Miscellaneous	● Time-out detector for printer ready ● Automatical warm boot operation by a key input when a programming is begun in the write protected disk ● Automatically effected DTR output of the MELCOM 70 interface by a warm boot

APPLICATION SOFTWARE LIST IN THE PC9001CPM

Program	Function	Supplement
EDIT	Screen Editor	PC9000 original editor
ED *	Line Editor	
ASM *	Assembler for 8080	
LOAD *	Modification of execution style	
DDT *	Debugger for 8080	
PROM	Programming control for 2708/16/32	The program to transfer an assemble result to a PROM by a built-in PROM programmer of PC9000. (Functions of padding and block transfer are programmed.)
DUMP *	Hex Dump	Source program is attached
PIP *	File handling	
STAT *	System status indicator	
SUBMIT *	Submit file management	
XSUB *	Optional SUBMIT	
CMNA 1	MELCOM 70 communication	The communication of object codes by the MELCOM 70 in the Intel HEX format
CMNA 2	PC4000 communication	The communication of object codes by the PC4000 in the Intel HEX format
MSS	General purpose file communication	The communication of file data with another CP/M machine
DISKCOPY	Copy of disk contents	Copy the disk contents from A to B
DDBI	Disk initialize	Initialization of PC9000 formatting for a disk in the market

The program marked with * is the utility program for CP/M.

PC9004

INTERFACE CABLE BETWEEN PC9000 AND PC4000

DESCRIPTION

The PC9004 is an interface cable to connect the cross assemble PC9000 and the debugging machine PC4000 by serial interface.

FEATURES

- High speed data communication for object codes of assembled and debugged results by connecting the PC9000 and the PC4000 with the serial interface.
- File communication by a serial interface between two PC9000s (when CP/M® option is used)

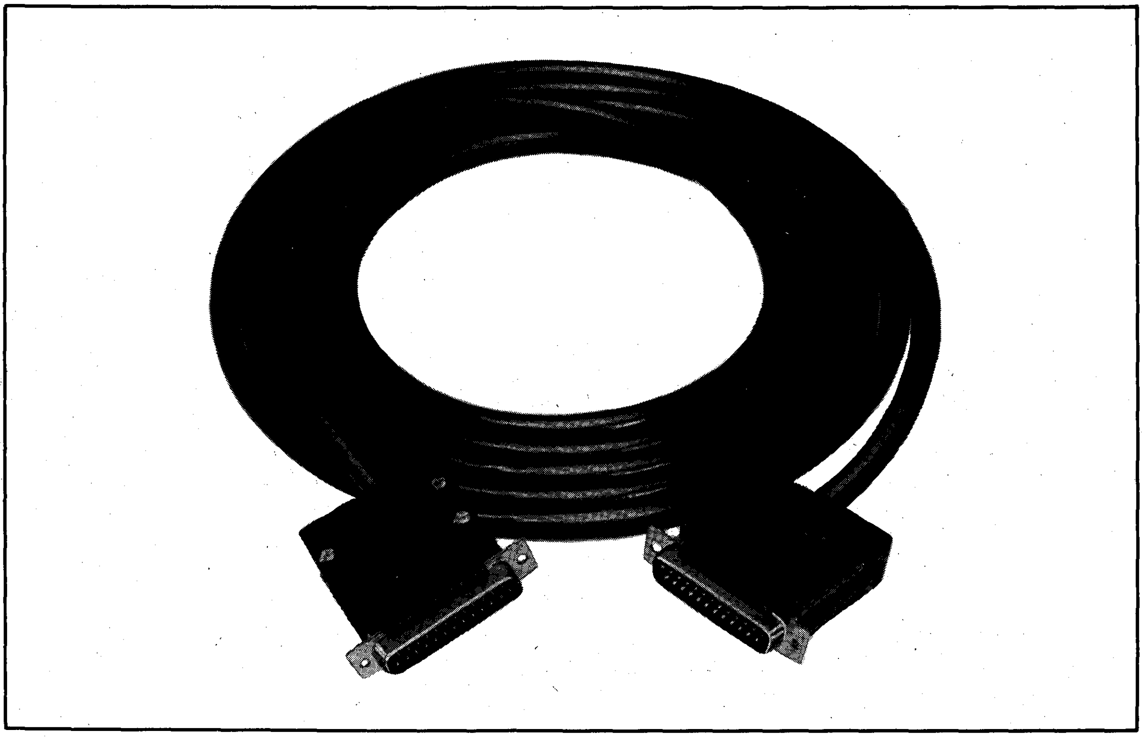
Note: ® CP/M is a registered trade-mark of Digital Research Inc.

FUNCTIONS

The PC9004 makes it possible to execute a data communication used with built-in current loop interface of the PC9000 and the PC4000. For example, to send a 4K bytes object côde in Intel HEX format (at 9600 baud), it takes approximately 17 seconds.

SPECIFICATION

Item	Specification	
Connector	RS232C type connector	
Line	7 lines	
Pin connection	Connector A	Connector B
	9	10
	10	9
	11	12
	12	11
	13	14
	14	13
	22	22



PC9005

PRINTER INTERFACE CABLE

DESCRIPTION

The PC9005 is an interface cable to connect the cross assemble machine PC9000 or the debugging machine PC4000 and a centronics printer.

FEATURES

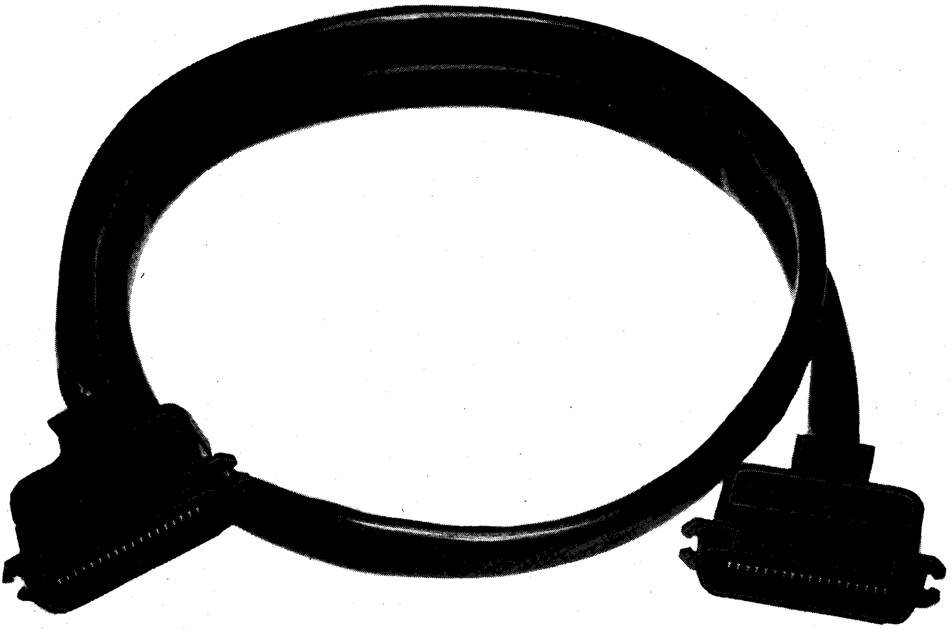
- Capability of outputting an assembled list or a disassembled list by connecting the PC9000 and a printer.
- Capability of outputting a disassembled list or a traced result at the 8080 software debugging by DDT when a CP/M is converted to the PC9000.
- Capability of outputting a disassembled list which corresponds to each microcomputer, an internal memory dump list etc. by connecting the PC4000 and a printer.

FUNCTIONS

The PC9005 is designed based on the centronics specification and is used as an interface cable between the PC9000 or the PC4000 and a centronics printer.

SPECIFICATION

Pin. No.	Signal	Content
1	STB	Strobe output signal of data
2~9	D_0~D_7	Data output signal to a printer
10	$\overline{ACK}$	Input signal of data acknowledgement
11	BUSY	BUSY input signal from a printer
12	P.EMP	Input signal notifying a printer paper empty
31	$\overline{RST}$	Output signal to initiate a printer
32	$\overline{FAULT}$	Input signal to notify printer errors
19~29	GND	Ground



PCA4040

MELPS 740 DEDICATED BOARD (M50740-XXXSP, M50741-XXXSP)

DESCRIPTION

The PCA4040 MELPS 740 dedicated board is for use with the PC4000 debugging machine for the M50740-XXXSP, and M50741-XXXSP single-chip 8-bit microcomputer, and it is used by inserting the board in the PC4000 cabinet.

FEATURES

- Connection to user's system by a flat cable
- Single-step operation and breakpoint operation capability from the PC4000 debugging machine keyboard. Debugging functions such as confirmation and modification of internal register contents.

APPLICATIONS

Development of hardware and software for systems using the MELPS 740 (M50740-XXXSP and M50741-XXXSP) single-chip microcomputer.

CONFIGURATION

As can be seen from the block diagram, the PCA4040 consists of the following hardware.

(1) Evaluation chip (M50740-000SP) and peripheral circuit
(2) EPROM with the PC4000 monitor program
(3) Single-step and breakpoint control circuit
(4) Program memory interface circuit
(5) Input/output buffer/latch circuit

The board and user system can be connected by means of an accessory cable.

FUNCTION

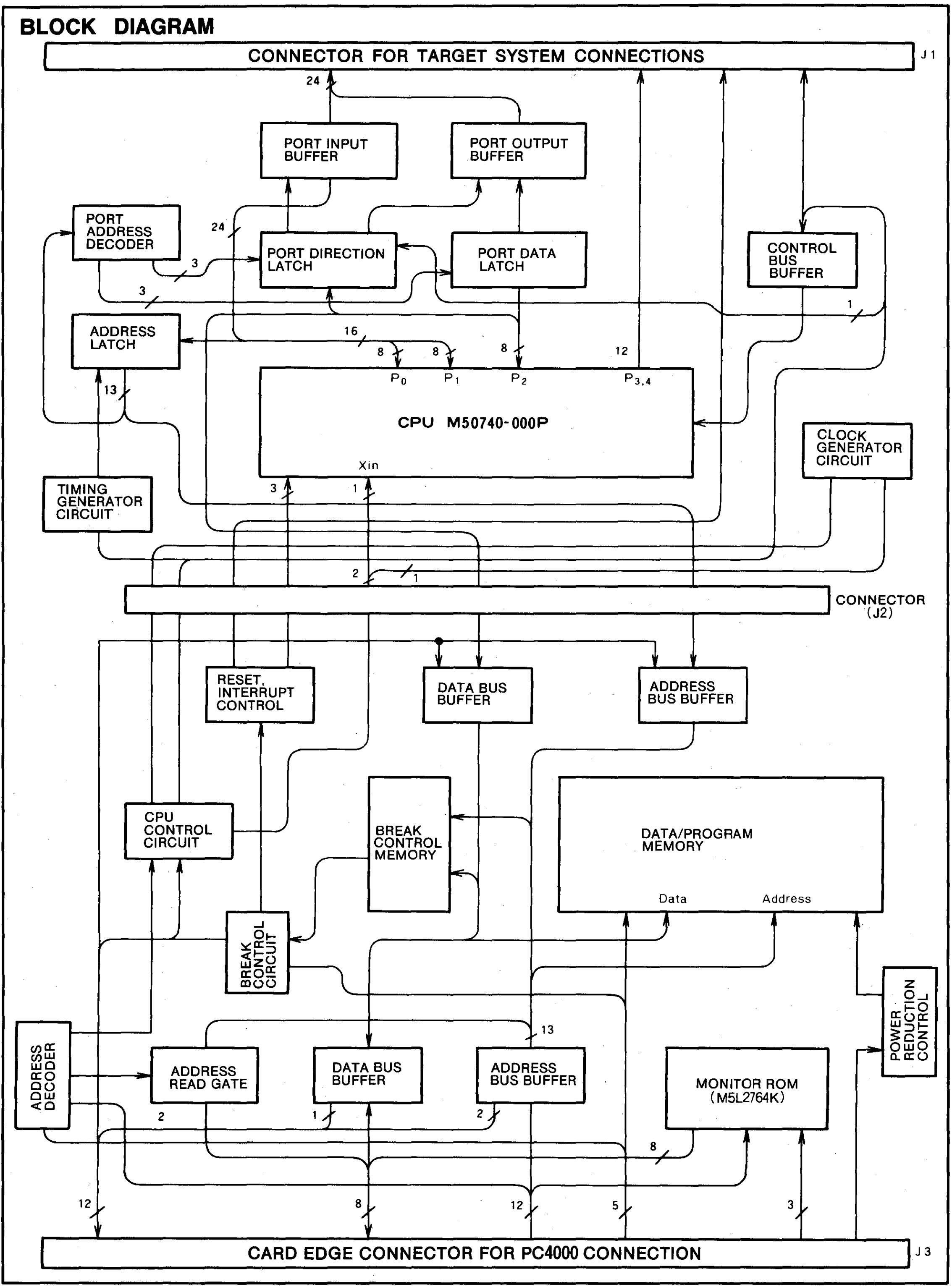
The debugging machine PC4000 operates as a debugging machine for MELPS 740 microcomputers using the monitor program contents of a ROM mounted on the dedicated board. The evaluation chip (M50740-000SP) loaded on the board executes the program stored in the program memory. The internal status of the evaluation chip is read out under monitor CPU control when halted by single-step operation and breakpoint operation.

SPECIFICATIONS

Item		Specification
Applicable microcomputer		M50740-XXXSP, M50741-XXXSP
Clock frequency	C F	4MHz
Applicable debugging machine		PC4000 (connected by a card edge connector J3)
Power supply		Supplied by PC4000
Connection to user's system		Accessory cable
Debugging functions (contents of monitor EPROM)		● Program execution from any address, stop and single-step operation ● Confirmation and change of program RAM contents ● Confirmation and modification of RAM data in evaluation chip and verification and modification of following registers and flags: • Program counter • Index register X • Index register Y • Stack pointer • Accumulator • Processor status register ● Data programing to EPROM and reading

MELPS 740 DEDICATED BOARD (M50740-XXXSP,M50741-XXXSP)

BLOCK DIAGRAM



PCA4060

MELPS 760 DEDICATED BOARD (M50760-XXXP, M50761-XXXP)

DESCRIPTION

The PCA4060 MELPS 760 dedicated board is for use with the PC4000 debugging machine for the M50760-XXXP and M50761-XXXP single-chip 4-bit microcomputers, and it is used by inserting the board in the PC4000 cabinet.

FEATURES

- Connection to user's system by a flat cable
- Single-step operation and breakpoint operation capability from the PC4000 debugging machine keyboard. Debugging functions such as confirmation of internal register contents.

APPLICATIONS

Development of hardware and software for systems using the MELPS 760 (M50760-XXXP, M50761-XXXP) single-chip microcomputer.

CONFIGURATION

As can be seen from the block diagram, the PCA4060 consists of the following hardware.

(1) Evaluation chip (M50760-000P) and peripheral circuit
(2) EPROM with the PC4000 monitor program
(3) Single-step and breakpoint control circuit
(4) Program memory interface circuit
(5) Input/output buffer/latch circuit

The board and user system can be connected by means of an accessory cable.

FUNCTION

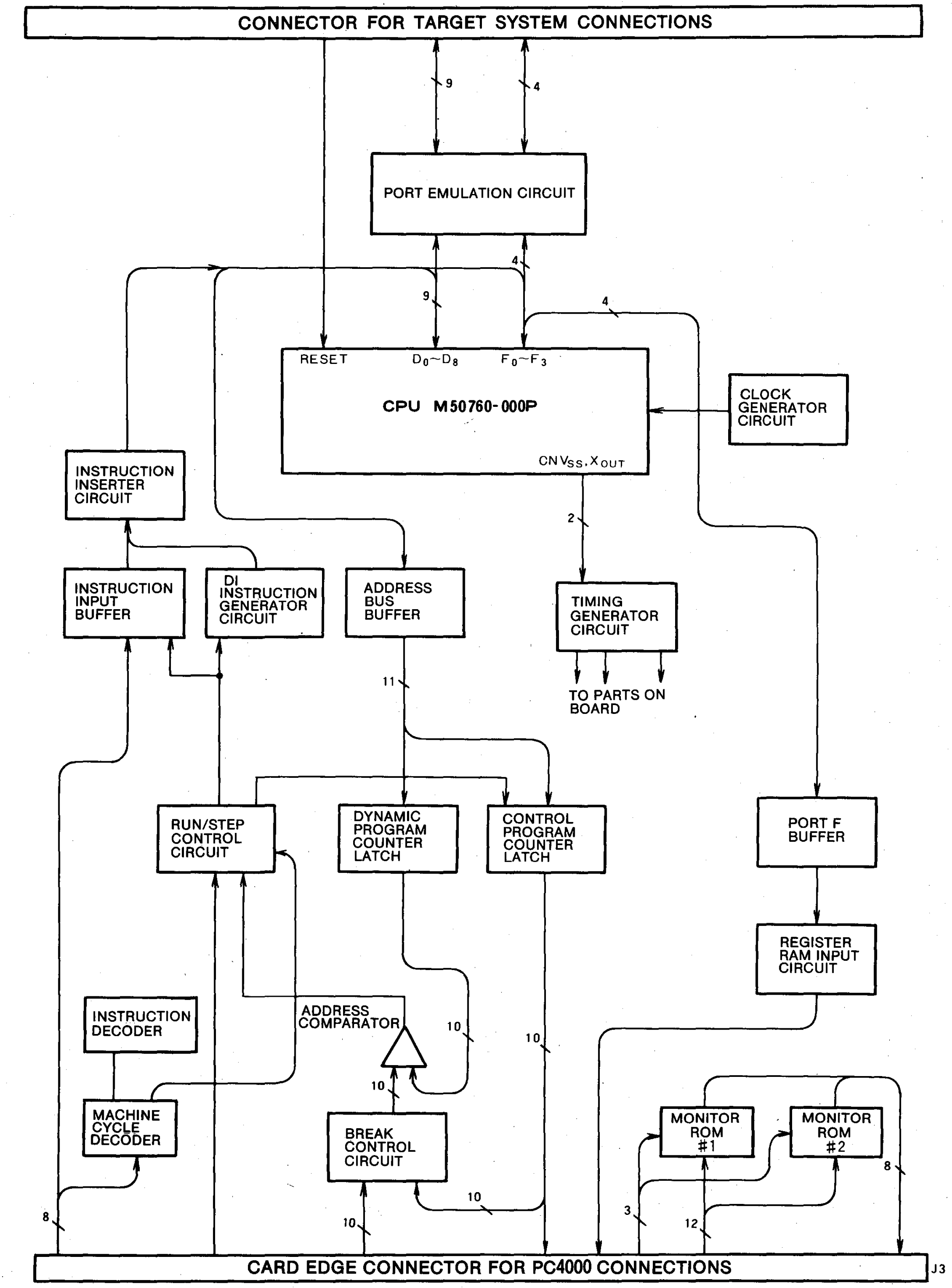
The debugging machine PC4000 operates as a debugging machine for MELPS 760 microcomputers using the monitor program contents of a ROM mounted on the dedicated board. The evaluation chip (M50760-000P) loaded on the board executes the program stored in the program memory in the PC4000 debugging machine.

The internal status of the evaluation chip is read out under monitor CPU control when halted by single-step operation and breakpoint operation.

SPECIFICATIONS

Item		Specification
Applicable microoomputer		M50760-XXXP, M50761-XXXP
Clock frequency	C F	400kHz
	C R	200～400kHz
Applicable debugging machine		PC4000 (connected by a card edge connector J3)
Power supply		Supplied by PC4000
Connection to user's system		Accessory cable
Debugging functions (contents of monitor EPROM)		● Program execution from any address, stop and single-step operation ● Confirmation and change of program RAM contents ● Confirmation and modification of RAM data in evaluation chip and verification and modification of following registers and flags: • Program counter • A register and carry • B register • Data pointers (X, Y) ● Data programing to EPROM and reading

BLOCK DIAGRAM



MITSUBISHI MICROCOMPUTERS

PCA4340

MELPS 740 EVALUATION BOARD (M50740-XXXSP, M50741-XXXSP)

DESCRIPTION

The PCA4340 evaluation board is used as an evaluation board for MELPS 740 8-bit single-chip computers.

When used in the external ROM mode, this board consists of the evaluation chip (M50740-000SP) and the program EPROM (M5L2732K) possessing equivalent functions to the masked ROM M50740-XXXSP and M50741-XXXSP. When creating the mask for a developed program, this board is suitable for verification and running tests.

FEATURES

- Board computer equivalent to M50740-XXXSP
- Simple program modification using an EPROM
- Connection to user's system by means of a cable
- Built-in clock generator

APPLICATIONS

Program and applications equipment development for MELPS 740 8-bit single-chip microcomputers

CONFIGURATION

As can be seen in the block diagram, the PCA4340 consists of the following hardware:

(1) Evaluation chip and peripheral circuit
(2) Program EPROM socket
(3) EPROM power supply circuit

The board and user system can be connected by means of an accessory cable.

FUNCTION

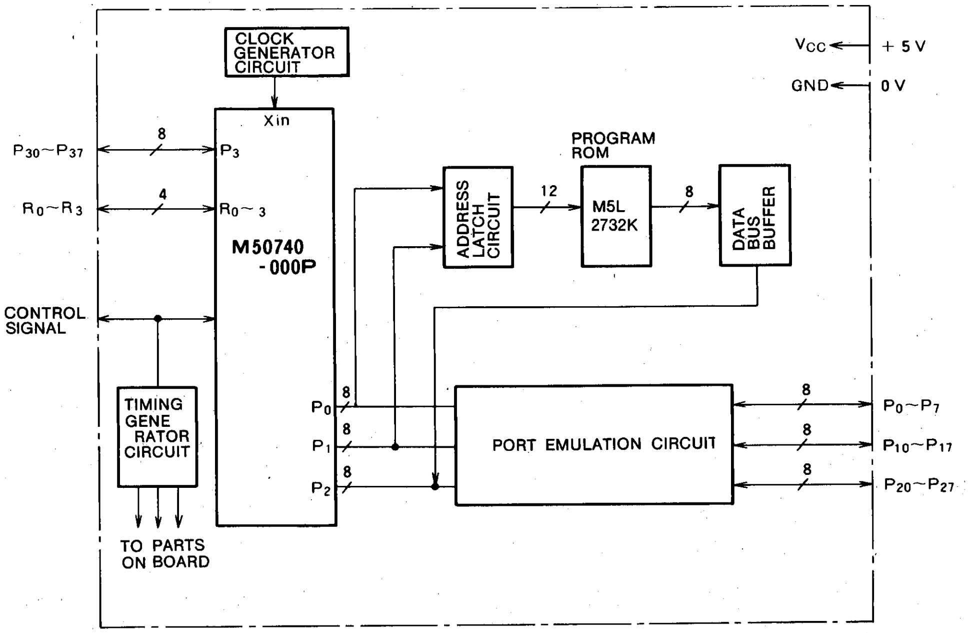
The evaluation chip (M50740-000SP) outputs the value of the program counter, and reads and executes the instruction stored in the appropriate EPROM address.

It is possible to have this board emulate the operation of a single-chip microcomputer.

SPECIFICATIONS

Item		Specification
Applicable microcomputer		M50740-XXXSP, M50741-XXXSP
Clock frequency	C F	4MHz
Cycle time		1μs (with 4MHz clock)
Power supply	Voltage	5V±5%, single
	Current	0.5mA typ. (J1 open, with NOP instruction execution)
Connection with user system		Accessory cable
Connectors	J1	For user system connection (60 pins)
	J2	2-pin angle pin header for power supply
Outer dimensions		200 (L) × 250 (W) × 20 (H) mm

BLOCK DIAGRAM



MELPS 760 EVALUATION BOARD (M50760-XXXP, M50761-XXXP)

DESCRIPTION

The PCA4360 evaluation board is used as an evaluation board for MELPS 760 4-bit single-chip computers.

When used in the external ROM mode, this board consists of the evaluation chip (M50760-000P) and the program EPROM (M5L2716K) possessing equivalent functions to the masked ROM M50760-XXXP and M50761-XXXP. When creating the mask for a developed program, this board is suitable for verification and running tests.

FEATURES

- Board computer equivalent to M50760-XXXP, M50761-XXXP
- Simple program modification using an EPROM
- Connection to user's system by means of a cable
- Built-in clock generator

APPLICATIONS

Program and applications equipment development for MELPS 760 4-bit single-chip microcomputers

CONFIGURATION

As can be seen in the block diagram, the PCA4360 consists of the following hardware:

(1) Evaluation chip and peripheral circuit
(2) Program EPROM socket

The board and user system can be connected by means of an accessory cable.

FUNCTION

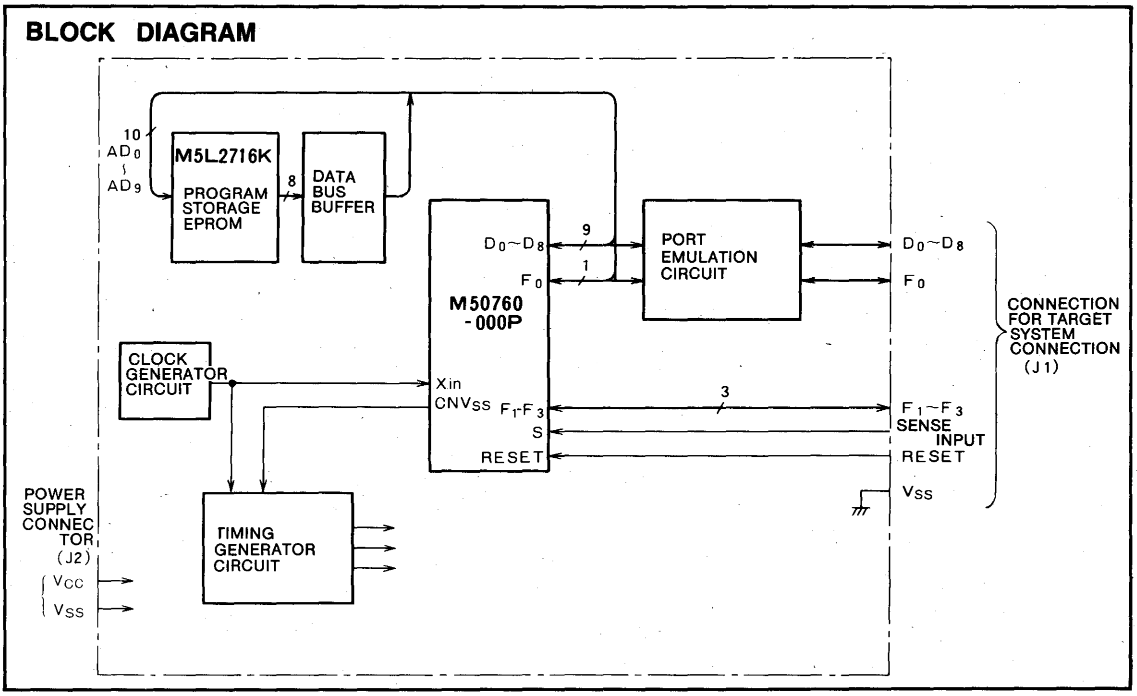
The evaluation chip (M50760-000P) outputs the value of the program counter, and reads and executes the instruction stored in the appropriate EPROM address.

It is possible to have this board emulate the operation of a single-chip microcomputer.

SPECIFICATIONS

Item		Specification
Applicable microcomputer		M50760-XXXP, M50761-XXXP
Clock frequency	C F	400kHz
	C R	200~400kHz
Cycle time		10μs
Power supply	Voltage	5V±5%, single
	Current	0.5mA typ.
Connection with user system		Accessory cable
Connectors	J1	For user system connection (20pin)
	J2	2-pin angle pin header for power supply
Outer dimensions		100 (L) × 150 (W) × 20 (H) mm

BLOCK DIAGRAM



PCA8400

MELPS 8-48 DEDICATED BOARD

DESCRIPTION

The PCA8400 is a dedicated MELPS 8-48 board for use with the PC4000 debugging machine for the 8-bit single-chip microcomputers and is used by inserting the board in the PC4000 cabinet.

FEATURES

- Connection to user's system by means of a 40-pin DIL plug
- Control circuits and connectors for the i 8748 writing adaptor (PC4100)

APPLICATIONS

The development of hardware and software for systems using the MELPS 8-48 8-bit single-chip microcomputers.

CONFIGURATION

As can be seen in the block diagram, the PCA8400 consists of the following hardware:

(1) Evaluation chip (M5L8039P-6) and peripheral circuitry
(2) ROM with the PC4000 monitor program
(3) Single-step and breakpoint control circuit
(4) Program memory interface circuit
(5) Input/output buffer/latch circuit

The PC4000 is connected to this board using a card edge connector and this board is connected to the user system by means of an accessory cable.

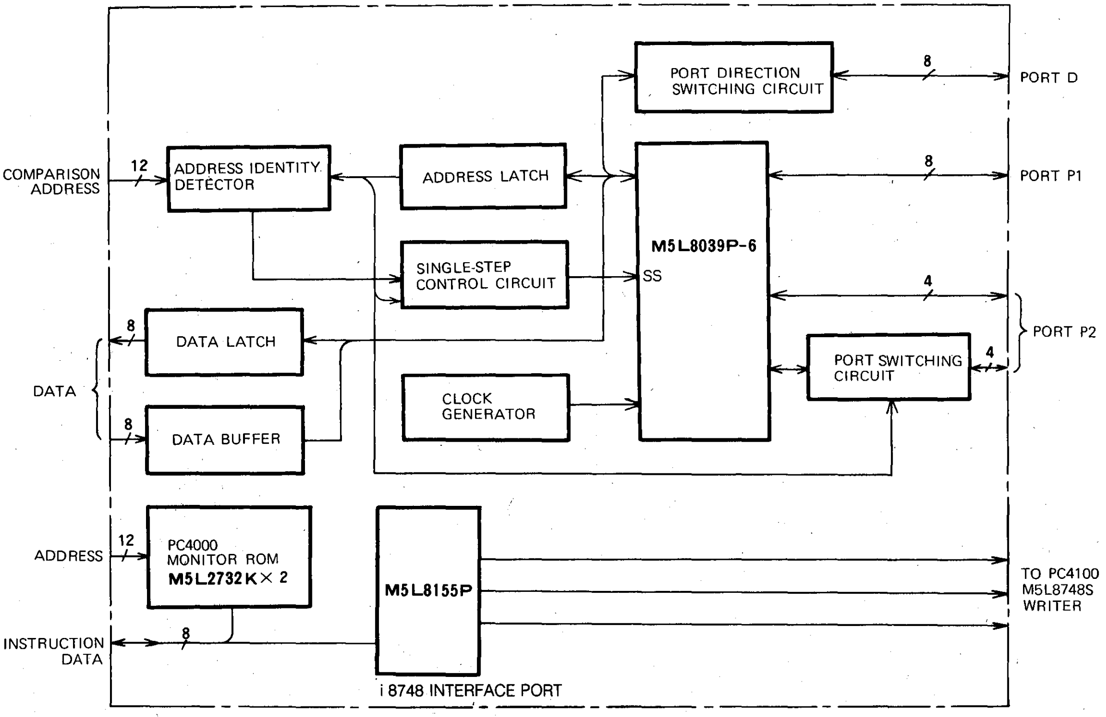
FUNCTION

The debugging machine PC4000 operates as a debugging machine for the MELPS 8-48 using the contents of the monitor ROM mounted on the dedicated board. The evaluation chip (M5L8039P-6) loaded on the board executes the program stored in the program memory in the PC4000 debugging machine.

The internal status of the evaluation chip is read out under monitor CPU control when single-step operation and breakpoint operation are halted.

An interface and connector to enable connection to the M5L8748S writing adaptor PC4100 has been provided, allowing programs to be written and read from the i 8748.

BLOCK DIAGRAM



SPECIFICATIONS

Item		Specification
Applicable microcomputers		**M5L8048-XXXP** **M5L8049-XXXP** **M5M8050H-XXXP** **M5M8050L-XXXP**
Clock frequency	Package	6.144MHz
	Variable range	1~6.144MHz (By changing the oscillator crystal)
Applicable debugging machine		PC4000 (connected by a card edge connector)
Power supply		Supplied from the PC4000 when inserted into the debugging machine
Connection to user's system		By an accessory cable
Debugging functions (contents of monitor EPROM)		● Program execution from any address and halt ● Data writing to EPROM and reading ● Confirmation and change of the contents ofprogram RAM ● Serial data transfer to an external device ● Confirmation and modification of the RAM data in the evaluation chip ● (M5L8039P-6) and the contents of the following registers and flags: ● Program counter ● Accumulator ● PSW
Other		By connecting the PC4100, read and write operations to the i8748 can be performed.

MITSUBISHI MICROCOMPUTERS

PCA8403

MELPS 8-48 EVALUTION BOARD

DESCRIPTION

The PCA8403 evaluation board is used as an evaluation board for MELPS 8-48 8-bit microcomputers.

This board consists basically of the external ROM chip (M5L8039P-11) and EPROM (M5L2732K), possessing equivalent functions to the masked ROM M5L8048-XXXP and M5L8049-XXXP. When creating the mask for a developed program, this board is suitable for program verification and running tests.

FEATURES

- Board computer equivalent to the M5L8048-XXXP, M5L8049-XXXP.
- Simple program modification using an EPROM
- Connection to user's system socket by means of a 40-pin DIL plug
- Built-in clock generator
- Speed up of a CPU using the M5L2764K

APPLICATIONS

Program and applications equipment development for MELPS 8-48 8-bit single-chip microcomputers.

FUNCTION

The evaluation chip (M5L8039P-11) outputs the value of the program counter and reads in instructions from ERROM and executes them.

The board is equivalent in operation to a single-chip microcomputer.

CONFIGURATION

As can be seen in the block diagram, the PCA8403 consists of the following hardware:

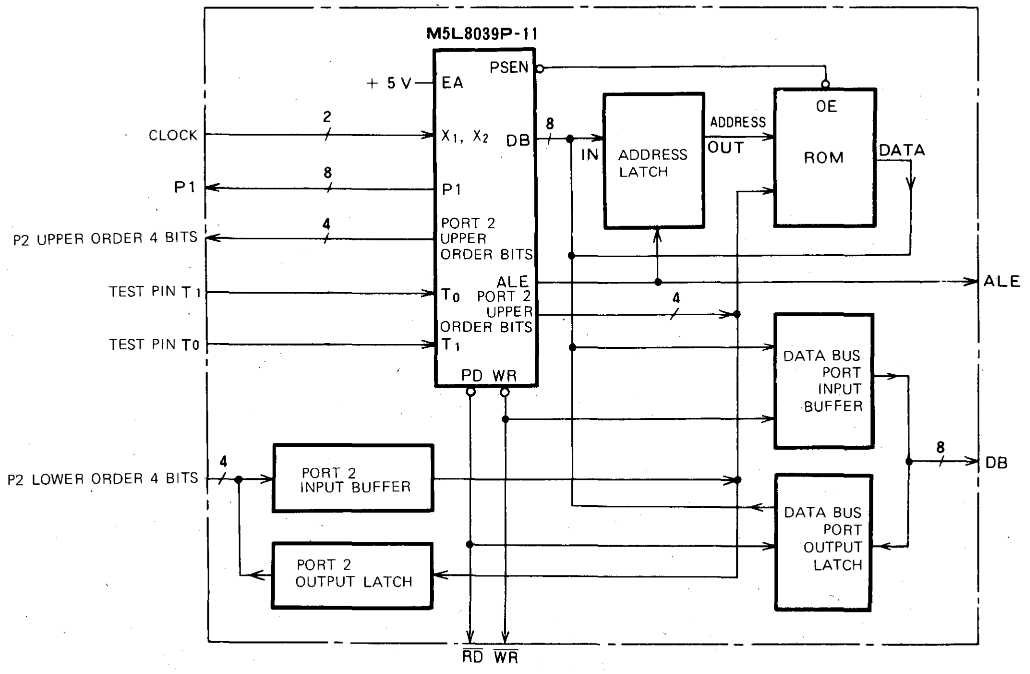
(1) Evaluation chip and peripheral circuitry
(2) Program EPROM socket
(3) EPROM power supply circuit

The board and user system can be connected by means of an accessory cable.

SPECIFICATIONS

Item	Specification
Type	8-bit parallel processor
CPU	**M5L8039P-11** (equivalent to Intel 8039)
Cycle time	Clock supplied by user system (maximum 11 MHz)
Memory	Program memory: 4K bytes (M5L2732K) Data memory: 128 bytes (built-in M5L8039P-11)
I/O	8-bit parallel port x3 Test pin x2
Interrupts	INT pin
Power supply	5V ±5%, 600mA (max)
Connector used	40-pin DIL accessary plag
Outer dimensions	50 (L) x 170 (W) x 35 (H) mm

BLOCK DIAGRAM



APPENDICES

MELPS 760 MASK ROM ORDERING METHOD

MASK ROM ORDERING METHOD

Mitsubishi Electric corp. receives via EPROMs the data for writing programs in the mask ROMs of single-chip 4-bit microcomputer.

When placing an order, submit three sets of one or more than one EPROMs in which the data for one pattern are written together with the prescribed confirmation document (s).

It use of different media is intended, make a request accordingly.

EPROM SPECIFICATIONS

1. M5L2716K, M5L2732K, Intel's 2716 or 2732 may be used, but M5L2716K and M5L2732K are regarded as standard.
2. The data and address of EPROM are processed by regarding the content of "H" as "1".
3. Write in the EPROMs to be submitted the kind of order to be placed as well as the distinction as to address assignment (the address to be assigned is given in the confirmation documents).
4. If a discrepancy (discrepancies) between the EPROMs submitted is found, a notice to that effect will be given.
5. Everything written from the start address of an EPROM onward is treated as data.

MASK ROM PROCESSING SYSTEM

Programs for automatic design of mask ROMs have been prepared and the followings are generated automatically :

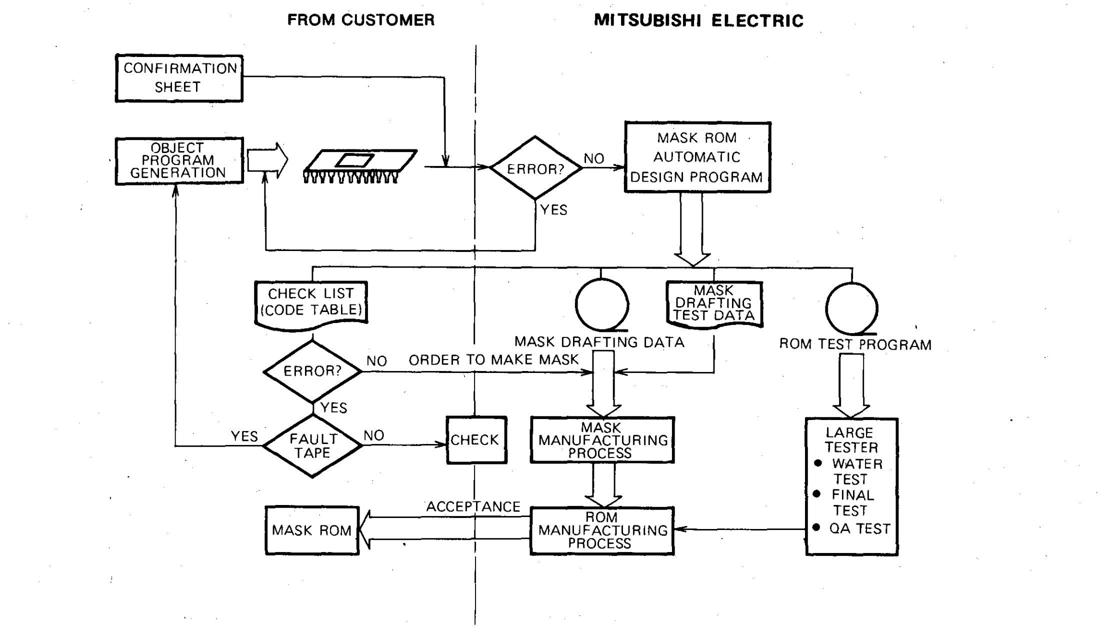
1. Drawing data for mask ROM generation.
2. Proof lists for checking errors in making mask ROMs.
3. Test pograms for large-scale tester.

A CAD system for this shown in the figure below.

ITEMS TO CONFIRM FOR ORDERING

1. Confirmation document for masking··················1 copy
2. Data for ROMs······························ 3 sets in EPROMs

MELPS 760 MASK ROM DEVELOPMENT CAD SYSTEM



MELPS760 MASK-PROGRAMMABLE ROM CONFIRMATION MATERIAL

SINGLE-CHIP 4-BIT MICROCOMPUTERS M50760-XXXSP

MITSUBISHI ELECTRIC

Customer
Company name ______________________________
Company address ______________________________ Tel ____________
Company contact ______________________________ Date ____________

Signature
Prepared
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.

EPROM type number	□2716	□2732
Address of EPROM	□A $(000_{16} \sim 3FF_{16})$	□A $(000_{16} \sim 3FF_{16})$

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 parentheses 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 mard should be no more than 12 characters consisting of alphanumeric characters (except J.I. and O) or dashes.

COMMENTS

MELPS 740 MASK ROM ORDERING METHOD

MASK ROM ORDERING METHOD

Mitsubishi Electric corp. receives via EPROMs the data for writing programs in the mask ROMs of single-chip 8-bit microcomputer.

When placing an order, submit three sets of one or more than one EPROMs in which the data for one pattern are written together with the prescribed confirmation document (s).

It use of different media is intended, make a request accordingly.

EPROM SPECIFICATIONS

1. M5L2716K, M5L2732K, Intel's 2716 or 2732 may be used, but M5L2716K and M5L2732K are regarded as standard.
2. The data and address of EPROM are processed by regarding the content of "H" as "1".
3. Write in the EPROMs to be submitted the kind of order to be placed as well as the distinction as to address assignment (the address to be assigned is given in the confirmation documents).
4. If a discrepancy (discrepancies) between the EPROMs submitted is found, a notice to that effect will be given.
5. Everything written from the start address of an EPROM onward is treated as data.

MASK ROM PROCESSING SYSTEM

Programs for automatic design of mask ROMs have been prepared and the followings are generated automatically :

1. Drawing data for mask ROM generation.
2. Proof lists for checking errors in making mask ROMs.
3. Test pograms for large-scale tester.

A CAD system for this shown in the figure below.

ITEMS TO CONFIRM FOR ORDERING

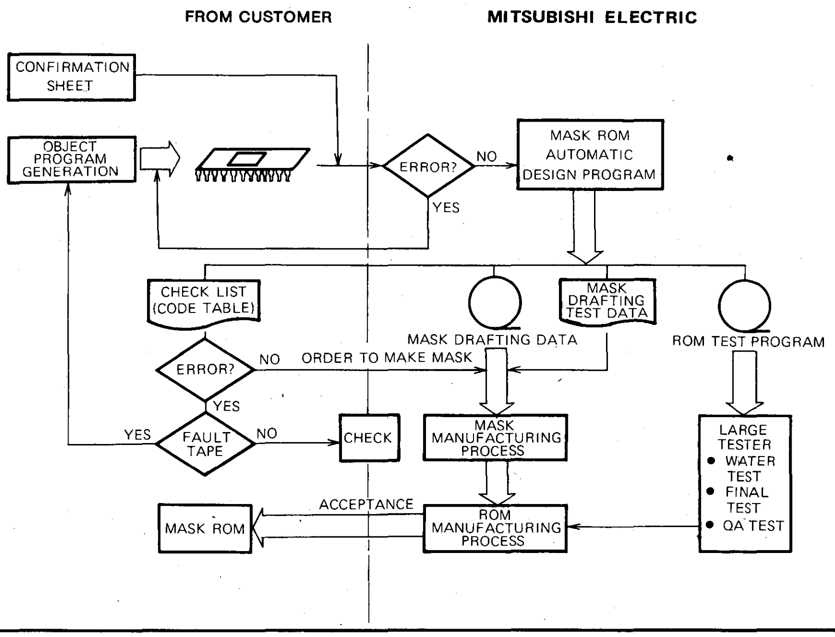
1. Confirmation document for masking··················1 copy
2. Data for ROMs······························ 3 sets in EPROMs

MASK OPTIONS

As for option, make entries in the confirmation documents by referring to the data book.

ROMs are made in accordance with the EPROMs and confirmation document submitted, then the EPROMs will be returned but the documents will not be returned.

MELPS 740 MASK ROM DEVELOPMENT CAD SYSTEM



MELPS740 MASK-PROGRAMMABLE ROM CONFIRMATION MATERIAL

SINGLE-CHIP 8-BIT MICROCOMPUTERS M50740-XXXSP

MITSUBISHI ELECTRIC

Customer Company name ______________________________ Company address ______________________________ Tel __________ Company contact ______________________________ Date __________	Signature Prepared 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.

EPROM type number	□2716	□2732
Address of EPROM	□A (000_{16} ~ $3FF_{16}$)	□A (000_{16} ~ $3FF_{16}$)

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 parentheses 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 mard should be no more than 12 characters consisting of alphanumeric characters (except J.I. and O) or dashes.

COMMENTS

MELPS 8-48 MASK ROM ORDERING METHOD

MASK ROM ORDERING METHOD

Described below is the ordering method applicable when programs submitted by the customer are written into the mask ROMs.

An automatic mask ROM design program is prepared for writing programs into mask ROMs, and (1) the drafting data for mask ROM generation, (2) the reference list for mask ROM preparation error checks and (3) an automatic test protram for the large-scale tester designed to test the mask ROMs are all automatically generated.

When the object program is stored in the MELPS8-48 single-chip microcomputer mask ROM, the order for the object program medium is received as an EPROM form. Consequently, the EPROM or EPROMs which have stored the object program equivalent to one single-chip microcomputer chip should be submitted accompanied by the prescribed confirmation sheets for 3 sets of EPROMs respectively.

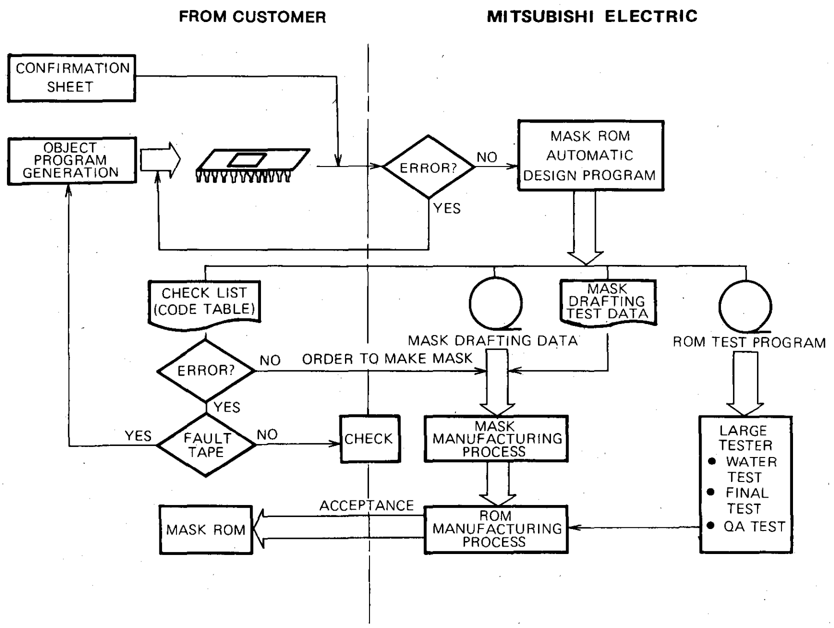
EPROM SPECIRICATIONS

1. Usable EPROMs include Mitsubishi's M5L2716K, M5L2732K or Intel's 2716, 2732, 8748, 8749 or their equivalent. The M5L2716K, M5L2732 and Intel's 8748, 8749 are the standard EPROMs.
2. "High" is treated as 1 for the EPROM data and address.
3. All the data from the head address to the final address are treated as the EPROM's effective data.

CHECKPOINTS

1. Cleary indicate the type number of EPROM.

MELPS 8-48 MASK ROM DEVELOPMENT CAD SYSTEM



MELPS8-48 MASK-PROGRAMMABLE ROM CONFIRMATION MATERIAL

SINGLE-CHIP 8-BIT MICROCOMPUTERS M5L8048-XXXP, M5L8049-XXXP, P-8, P-6, M5L8049H-XXXP, M5M8050H-XXXP, M5M8050L-XXXP, M5M80C49-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.

EPROM type number microcomputer type number	□2716	□2732	
□M5L8048-XXXP	□A (000_{16} ~ $3FF_{16}$)	□A (000_{16} ~ $3FF_{16}$)	□8748
□M5L8049-XXXP □M5L8049-XXXP-8 □M5L8049-XXXP-6	□A (000_{16} ~ $7FF_{16}$)	□A (000_{16} ~ $7FF_{16}$)	□8749
□M5L8049H-XXXP	□A (000_{16} ~ $7FF_{16}$)	□A (000_{16} ~ $7FF_{16}$)	□8749
□M5M8050H-XXXP □M5M8050L-XXXP	□A (000_{16} ~ $7FF_{16}$) □B (800_{16} ~ FFF_{16})	□A (000_{16} ~ FFF_{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 parentheses 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 mard should be no more than 12 characters consisting of alphanumeric characters (except J.I. and O) or dashes.

COMMENTS

MELPS8-48 MASK-PROGRAMMABLE ROM CONFIRMATION MATERIAL

SINGLE-CHIP 8-BIT MICROCOMPUTERS M5M80C49-XXXP

MITSUBISHI ELECTRIC

Customer
Company name ______________________
Company address ______________________ Tel __________
Company contact ______________________ Date __________

Signature
Prepared
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.

EPROM type number microcomputer type number	□2716	□2732	
□M5M80C49P-XXXP	□A (000_{16} ~ $7FF_{16}$)	□A (000_{16} ~ FFF_{16})	□8749

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 parentheses 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 mard should be no more than 12 characters consisting of alphanumeric characters (except J.I. and O) or dashes.

COMMENTS

MELPS 8-41 MASK ROM ORDERING METHOD

MASK ROM ORDERING METHOD

Described below is the ordering method applicable when programs submitted by the customer are written into the mask ROMs.

An automatic mask ROM design program is prepared for writing programs into mask ROMs, and (1) the drafting data for mask ROM generation, (2) the reference list for mask ROM preparation error checks and (3) an automatic test protram for the large-scale tester designed to test the mask ROMs are all automatically generated.

When the object program is stored in the MELPS8-48 single-chip microcomputer mask ROM, the order for the object program medium is received as an EPROM form. Consequently, the EPROM or EPROMs which have stored the object program equivalent to one single-chip microcomputer chip should be submitted accompanied by the prescribed confirmation sheets for 3 sets of EPROMs respectively.

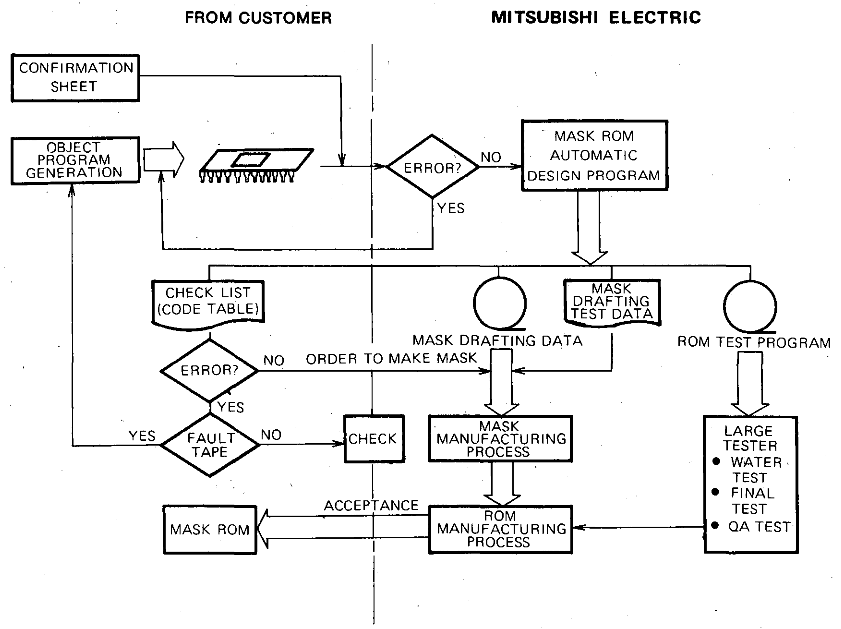
EPROM SPECIRICATIONS

1. Usable EPROMs include Mitsubishi's M5L2716K, M5L2732K, or Intel's 2716, 2732, 8741, 8741A, 8742 or their equivalent. The M5L2716K and M5L2732K are the standard EPROMs.
2. "High" is treated as 1 for the EPROM data and address.
3. All the data from the head address to the final address are treated as the EPROM's effective data.

CHECKPOINTS

1. Cleary indicate the type number of EPROM.

MELPS 8-41 MASK ROM DEVELOPMENT CAD SYSTEM



MELPS 8-41 MASK ROM ORDERING METHOD

MELPS8-41 MASK-PROGRAMMABLE ROM CONFIRMATION MATERIAL

SLAVE COMPUTERS M5L8041A-XXXP, M5L8042-XXXP

MITSUBISHI ELECTRIC

Customer
Company name ______________________________
Company address ______________________________ Tel ____________
Company contact ______________________________ Date ____________

Signature
Prepared
Approved

The Slave computer 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.

EPROM Type number microcomputer type number	□2716	□2732	
□M5L8041A-XXXP	□A ($000_{16} \sim 3FF_{16}$)	□A ($000_{16} \sim 3FF_{16}$)	□8741 □8741A
□M5L8042-XXXP	□A ($000_{16} \sim 7FF_{16}$)	□A ($000_{16} \sim 7FF_{16}$)	□8742

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 parentheses 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 mard should be no more than 12 characters consisting of alphanumeric characters (except J.I. and O) or dashes.

COMMENTS

CONTACT ADDRESSES FOR FURTHER INFORMATION

JAPAN
Electronics Marketing Division
Mitsubishi Electric Corporation
2-3, Marunouchi 2-chome
Chiyoda-ku, Tokyo 100, Japan
Telex: 24532 MELCO J
Telephone: (03) 218-3473
(03) 218-3499
Facsimile: (03) 214-5570

HONG KONG
Ryoden Electric Engineering Co., Ltd.
22nd fl., Leighton Centre
77, Leighton Road
Causeway Bay, Hong Kong
Telex: 73411 RYODEN HX
Telephone: (5) 795-2427
Facsimile: (852) 123-4344

TAIWAN
MELCO Taiwan Co., Ltd.
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.
777 North Pastoria Avenue
Sunnyvale CA 94086, U.S.A.
Telex: 172296 MELA SUVL
Telephone: (408) 730-5900
Facsimile: (408) 730-4972

Mitsubishi Electronics America, Inc
200 Unicorn Park Drive
Woburn, MA 01801, U.S.A.
Telex: 951796 MELASB WOBN
Telephone: (617) 938-1220
Facsimile: (617) 938-1075

Mitsubishi Electronics America, Inc.
3247 West Story Road,
Los Colinas Business Park
Irving, TX 75062, U.S.A.
Telephone: (214) 258-1266
Facsimile: (214) 659-9313

Mitsubishi Electronics America, Inc.
799 North Bierman Circle,
Mt. Prospect, ILL 60056, U.S.A.
Telex: 270636 MESA CHI-MPCT
Telephone: (312) 298-9223～8
Facsimile: (312) 298-9223～8

WEST GERMANY
Mitsubishi Electric Europe GmbH
Brandenburger Str. 40
4030 Ratingen, West Germany
Telex: 8585070 MED D
Telephone: (02102) 4860
Facsimile: (02102) 486-115

U.K.
Mitsubishi Electric (U.K.) Ltd.
18th Floor, Centre Point,
103 New Oxford st.,
London WC1A1EB, U.K.
Telex: 296195 MELCO G
Telephone: (379) 7160
Facsimile: (836) 0699

AUSTRALIA
Mitsubishi Electric Australia Pty. Ltd.
73-75, Epping Road, North Ride,
N.S.W. 2113 Australia
P.O. Box 1567 Macquarie Centre
Telex: MESYO AA 26614
Telephone: (888) 5777
Facsimile: (887) 3635

MITSUBISHI
ELECTRIC

MITSUBISHI DATA BOOK
SINGLE-CHIP MICROCOMPUTERS

March, First Edition 1984

Editioned by
Committee of editing of Mitsubishi Semiconductor Data Book

Published by
Mitsubishi Electric Corp., Semiconductor Division

1984 MITSUBISHI DATA BOOK
SINGLE-CHIP MICROCOMPUTERS

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

H-C5229-A KI-8403 Printed in Japan (ROD)

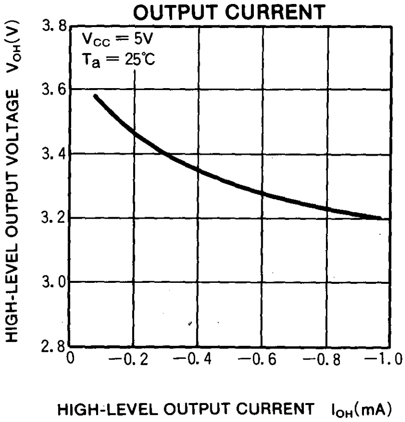
New publication, effective Mar, 1984
Specifications subject to change without notice.

This book contains the typical characteristics of MITSUBISHI MICROCOMPUTER 8-48 series, which are omitted from the data book.

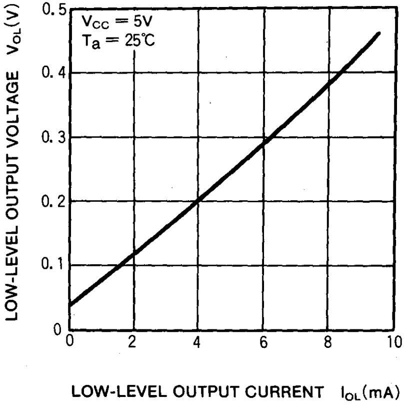
MITSUBISHI ELECTRIC

TYPICAL CHARACTERISTICS

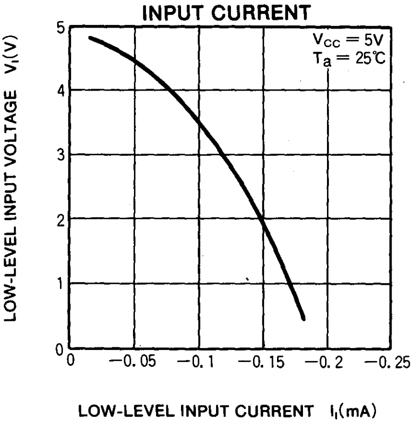
DATA BUS HIGH-LEVEL OUTPUT VOLTAGE VS. HIGH-LEVEL OUTPUT CURRENT



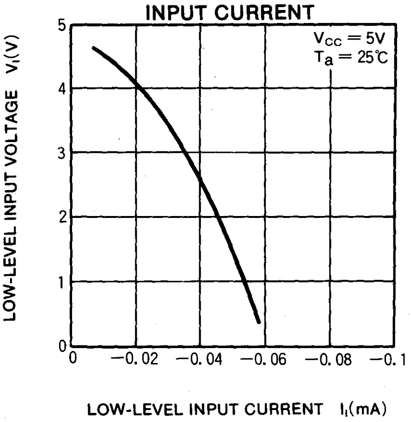
DATA BUS LOW-LEVEL OUTPUT VOLTAGE VS. LOW-LEVEL OUTPUT CURRENT



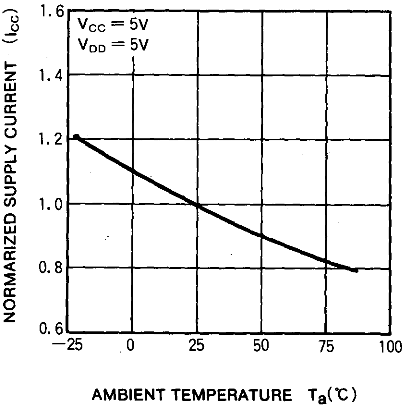
P_1, P_2 LOW-LEVEL INPUT VOLTAGE VS. LOW-LEVEL INPUT CURRENT



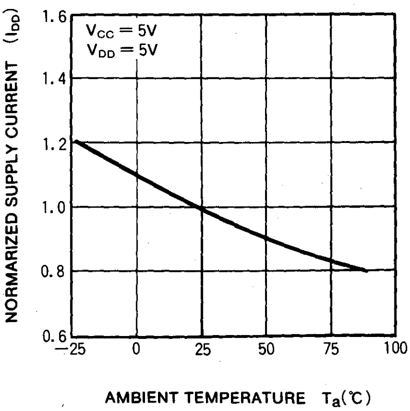
RESET LOW-LEVEL INPUT VOLTAGE VS. LOW-LEVEL INPUT CURRENT



NORMARIZED SUPPLY CURRENT (I_{CC}) VS. AMBIENT TEMPERATURE

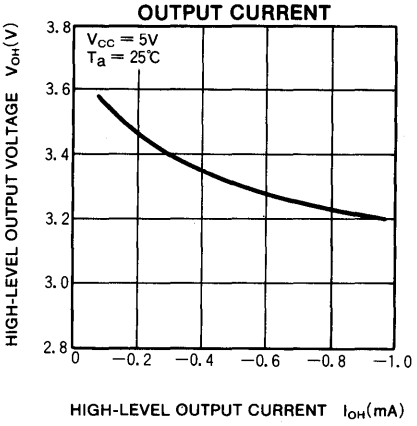


NORMARIZED SUPPLY CURRENT (I_{DD}) VS. AMBIENT TEMPERATURE

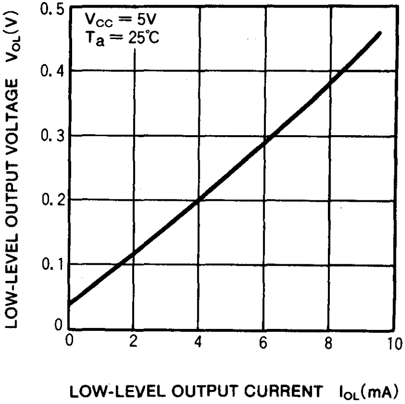


TYPICAL CHARACTERISTICS

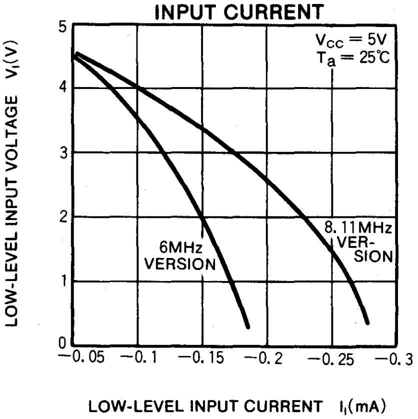
DATA BUS HIGH-LEVEL OUTPUT VOLTAGE VS. HIGH-LEVEL OUTPUT CURRENT



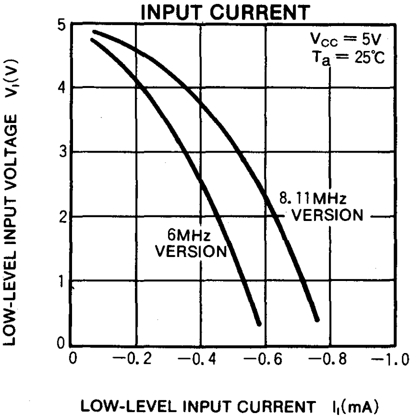
DATA BUS LOW-LEVEL OUTPUT VOLTAGE VS. LOW-LEVEL OUTPUT CURRENT



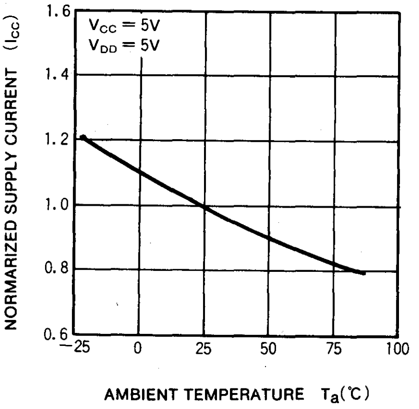
P_1, P_2 LOW-LEVEL INPUT VOLTAGE VS. LOW-LEVEL INPUT CURRENT



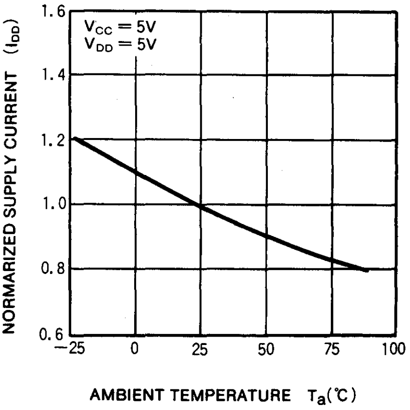
RESET LOW-LEVEL INPUT VOLTAGE VS. LOW-LEVEL INPUT CURRENT



NORMARIZED SUPPLY CURRENT (I_{CC}) VS. AMBIENT TEMPERATURE

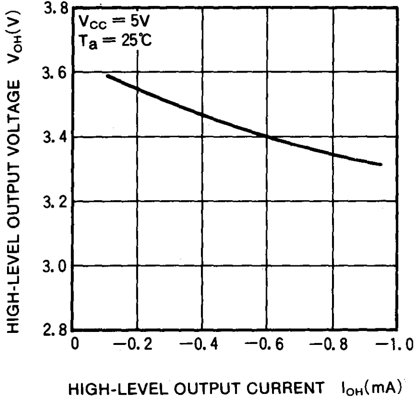


NORMARIZED SUPPLY CURRENT (I_{DD}) VS. AMBIENT TEMPERATURE

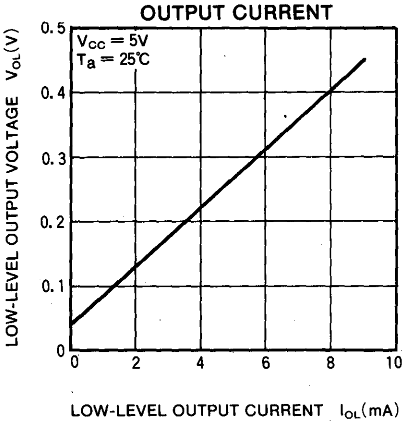


TYPICAL CHARACTERISTICS

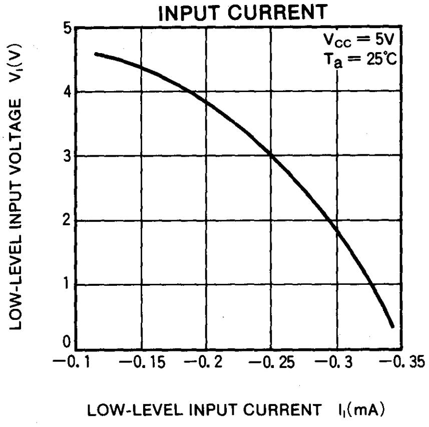
DATA BUS HIGH-LEVEL OUTPUT VOLTAGE VS. HIGH-LEVEL OUTPUT CURRENT



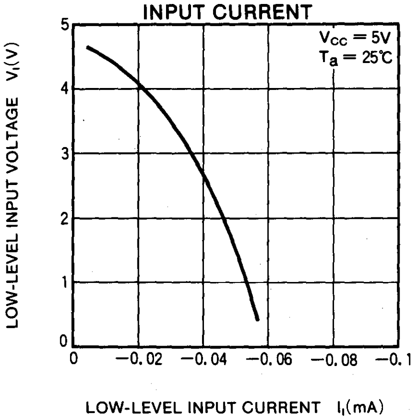
DATA BUS LOW-LEVEL OUTPUT VOLTAGE VS. LOW-LEVEL OUTPUT CURRENT



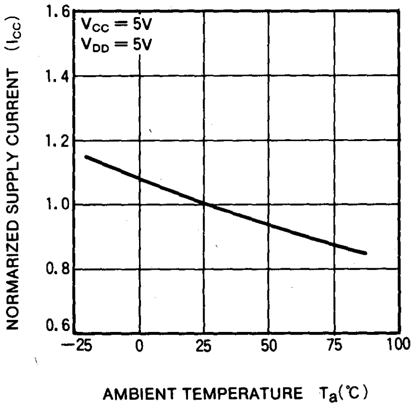
P_1, P_2 LOW-LEVEL INPUT VOLTAGE VS. LOW-LEVEL INPUT CURRENT



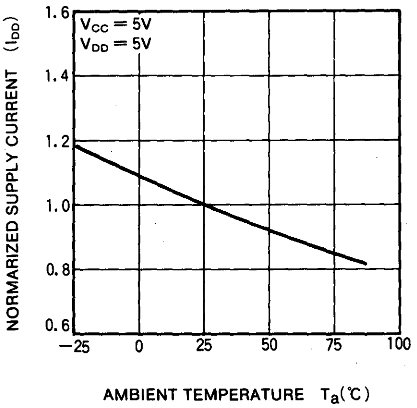
RESET LOW-LEVEL INPUT VOLTAGE VS. LOW-LEVEL INPUT CURRENT



NORMARIZED SUPPLY CURRENT (I_{CC}) VS. AMBIENT TEMPERATURE

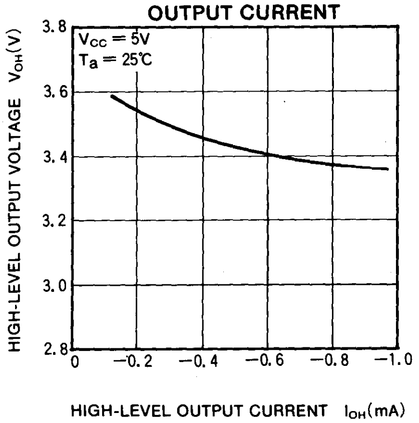


NORMARIZED SUPPLY CURRENT (I_{DD}) VS. AMBIENT TEMPERATURE

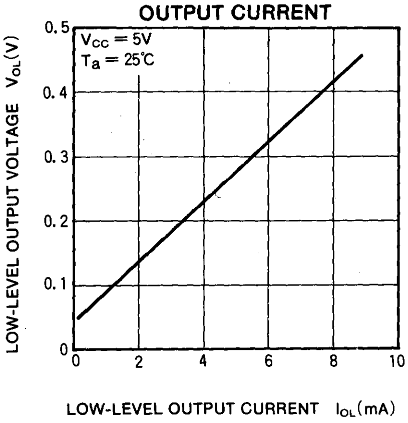


TYPICAL CHARACTERISTICS

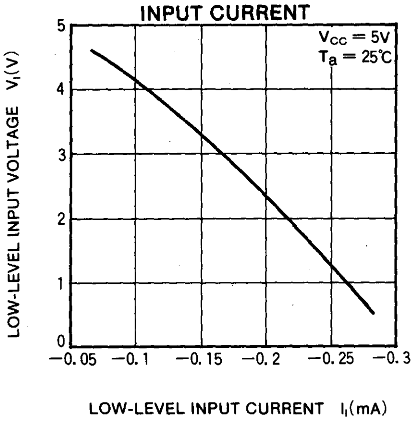
DATA BUS HIGH-LEVEL OUTPUT VOLTAGE VS. HIGH-LEVEL OUTPUT CURRENT



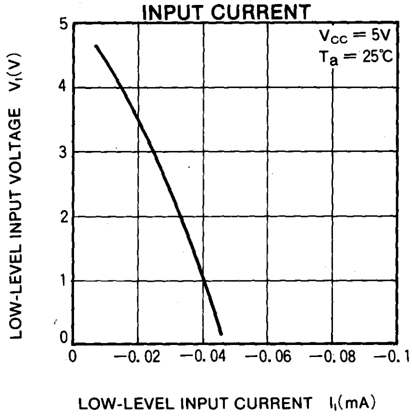
DATA BUS LOW-LEVEL OUTPUT VOLTAGE VS. LOW-LEVEL OUTPUT CURRENT



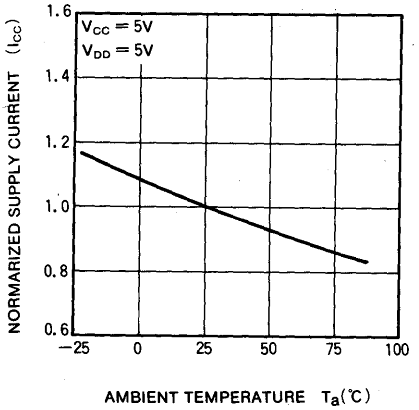
P_1, P_2 LOW-LEVEL INPUT VOLTAGE VS. LOW-LEVEL INPUT CURRENT



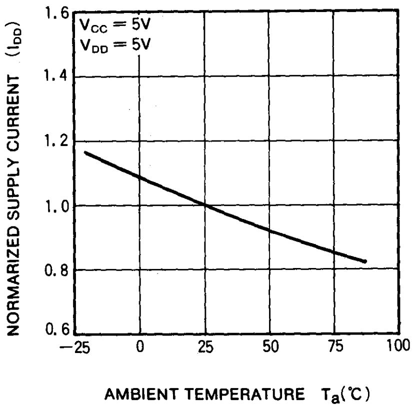
RESET LOW-LEVEL INPUT VOLTAGE VS. LOW-LEVEL INPUT CURRENT



NORMARIZED SUPPLY CURRENT (I_{CC}) VS. AMBIENT TEMPERATURE

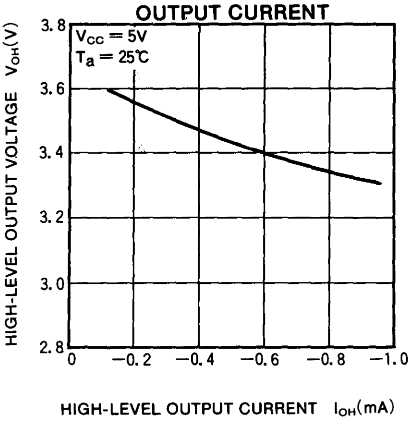


NORMARIZED SUPPLY CURRENT (I_{DD}) VS. AMBIENT TEMPERATURE

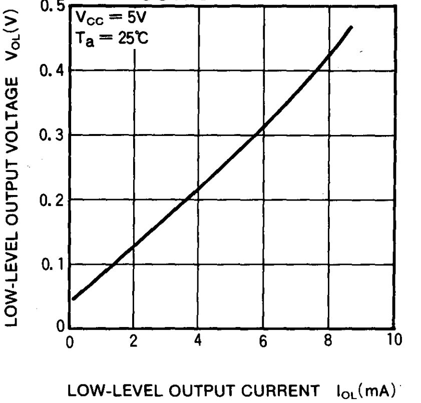


TYPICAL CHARACTERISTICS

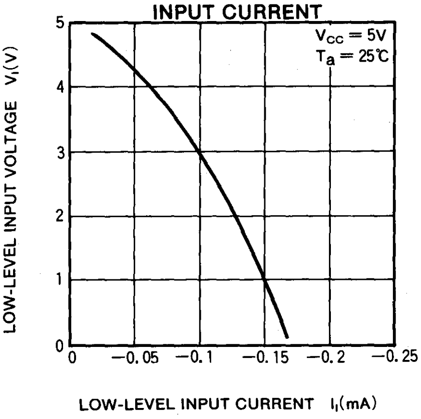
DATA BUS HIGH-LEVEL OUTPUT VOLTAGE VS. HIGH-LEVEL OUTPUT CURRENT



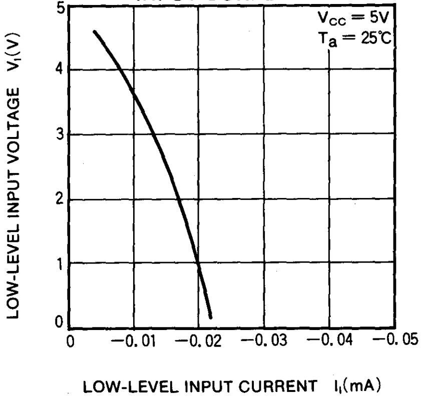
DATA BUS LOW-LEVEL OUTPUT VOLTAGE VS. LOW-LEVEL OUTPUT CURRENT



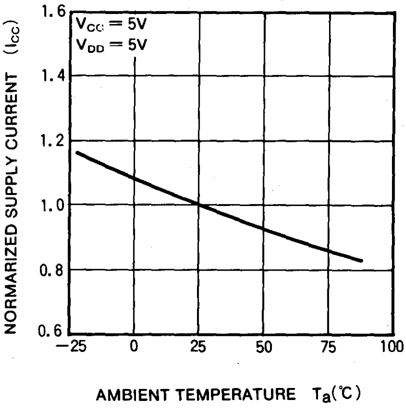
P_1, P_2 LOW-LEVEL INPUT VOLTAGE VS. LOW-LEVEL INPUT CURRENT



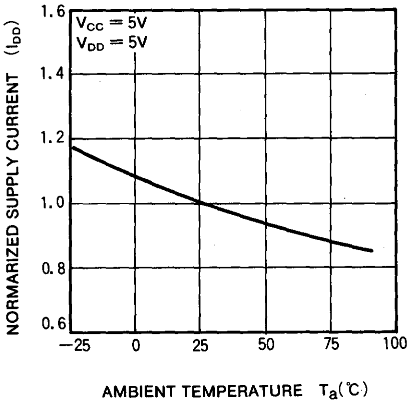
RESET LOW-LEVEL INPUT VOLTAGE VS. LOW-LEVEL INPUT CURRENT



NORMARIZED SUPPLY CURRENT (I_{CC}) VS. AMBIENT TEMPERATURE

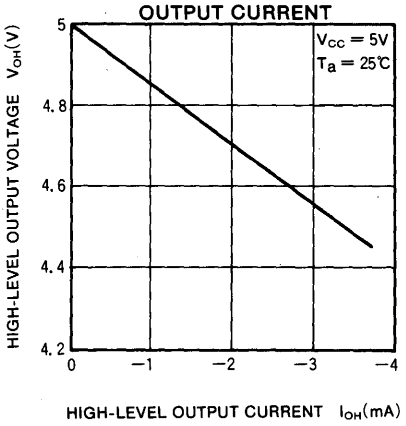


NORMARIZED SUPPLY CURRENT (I_{DD}) VS. AMBIENT TEMPERATURE

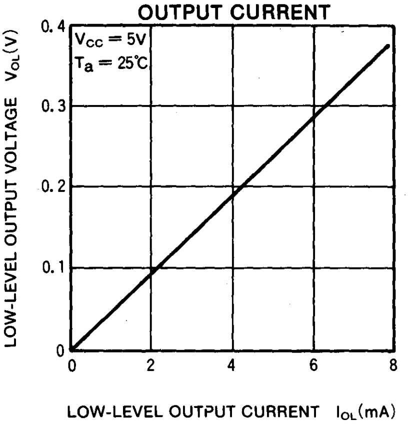


TYPICAL CHARACTERISTICS

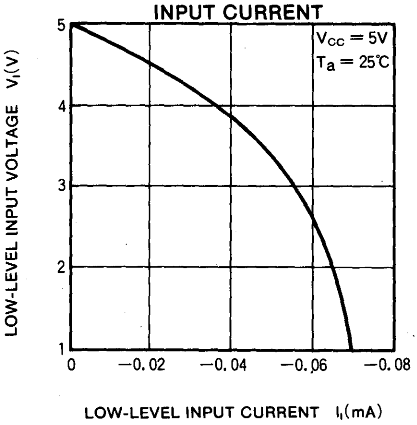
DATA BUS HIGH-LEVEL OUTPUT VOLTAGE VS. HIGH-LEVEL OUTPUT CURRENT



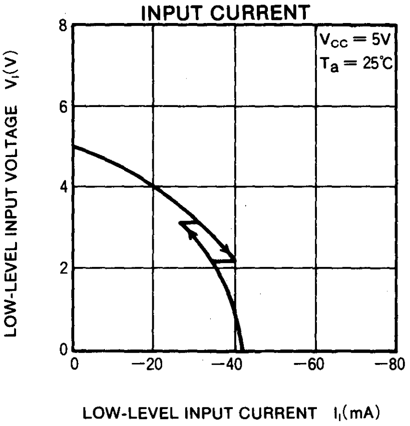
DATA BUS LOW-LEVEL OUTPUT VOLTAGE VS. LOW-LEVEL OUTPUT CURRENT



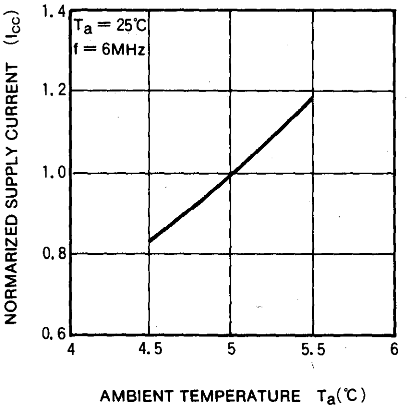
P_1, P_2 LOW-LEVEL INPUT VOLTAGE VS. LOW-LEVEL INPUT CURRENT



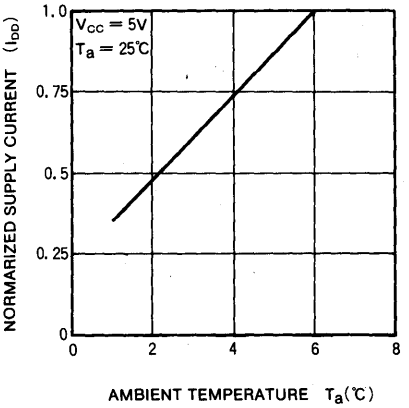
RESET LOW-LEVEL INPUT VOLTAGE VS. LOW-LEVEL INPUT CURRENT



NORMARIZED SUPPLY CURRENT (I_{CC}) VS. AMBIENT TEMPERATURE



NORMARIZED SUPPLY CURRENT (I_{DD}) VS. AMBIENT TEMPERATURE



MEMO

CONTACT ADDRESSES FOR FURTHER INFORMATION

JAPAN
Electronics Marketing Division
Mitsubishi Electric Corporation
2-3, Marunouchi 2-chome
Chiyoda-ku, Tokyo 100, Japan
Telex: 24532 MELCO J
Telephone: (03) 218-3473
(03) 218-3499
Facsimile: (03) 214-5570

HONG KONG
Ryoden Electric Engineering Co., Ltd.
22nd fl., Leighton Centre
77, Leighton Road
Causeway Bay, Hong Kong
Telex: 73411 RYODEN HX
Telephone: (5) 795-2427
Facsimile: (852) 123-4344

TAIWAN
MELCO Taiwan Co., Ltd.
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.
777 North Pastoria Avenue
Sunnyvale CA 94086, U.S.A.
Telex: 172296 MELA SUVL
Telephone: (408) 730-5900
Facsimile: (408) 730-4972

Mitsubishi Electronics America, Inc.
200 Unicorn Park Drive
Woburn, MA 01801, U.S.A.
Telex: 951796 MELASB WOBN
Telephone: (617) 938-1220
Facsimile: (617) 938-1075

Mitsubishi Electronics America, Inc.
3247 West Story Road,
Los Colinas Business Park
Irving, TX 75062, U.S.A.
Telephone: (214) 258-1266
Facsimile: (214) 659-9313

Mitsubishi Electronics America, Inc.
799 North Bierman Circle,
Mt. Prospect, ILL 60056, U.S.A.
Telex: 270636 MESA CHI-MPCT
Telephone: (312) 298-9223∼8
Facsimile: (312) 298-9223∼8

WEST GERMANY
Mitsubishi Electric Europe GmbH
Brandenburger Str. 40
4030 Ratingen, West Germany
Telex: 8585070 MED D
Telephone: (02102) 4860
Facsimile: (02102) 486-115

U.K.
Mitsubishi Electric (U.K.) Ltd.
18th Floor, Centre Point,
103 New Oxford st.,
London WC1A1EB, U.K.
Telex: 296195 MELCO G
Telephone: (379) 7160
Facsimile: (836) 0699

AUSTRALIA
Mitsubishi Electric Australia Pty. Ltd.
73-75, Epping Road, North Ride,
N.S.W. 2113 Australia
P.O. Box 1567 Macquarie Centre
Telex: MESYO AA 26614
Telephone: (888) 5777
Facsimile: (887) 3635

MITSUBISHI
ELECTRIC