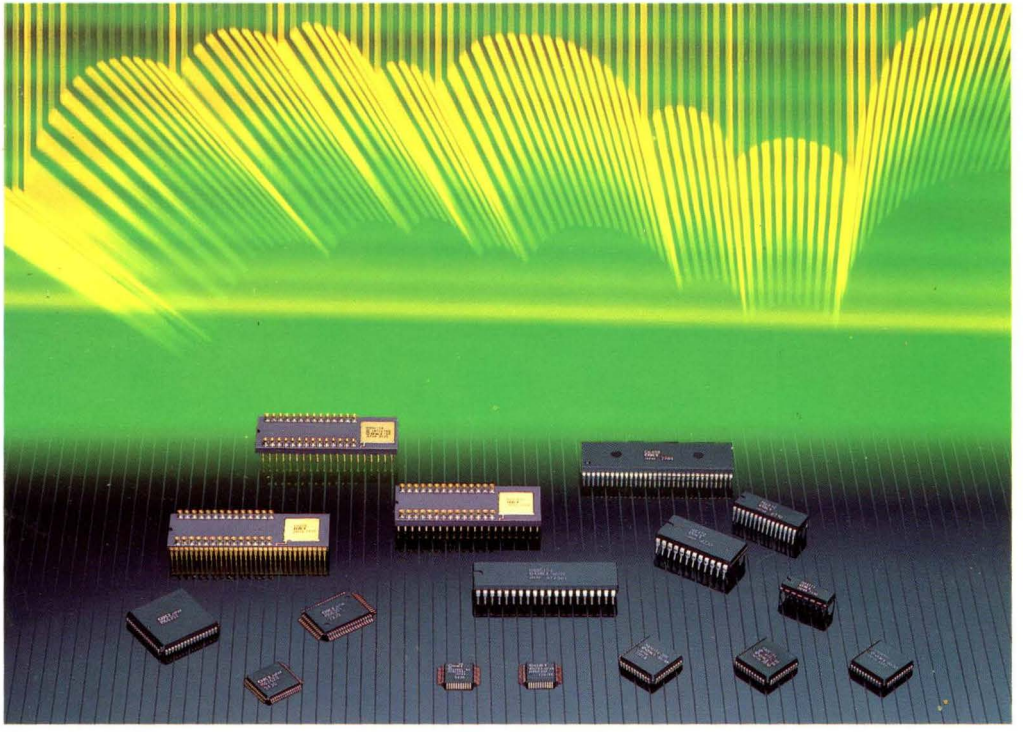
DATA BOOK

OKI

MICROCONTROLLER



OKI Semiconductor

FIFTH EDITION
ISSUE DATE: JUN., 1990

OKI MICROCONTROLLER DATA BOOK ('90-'91)

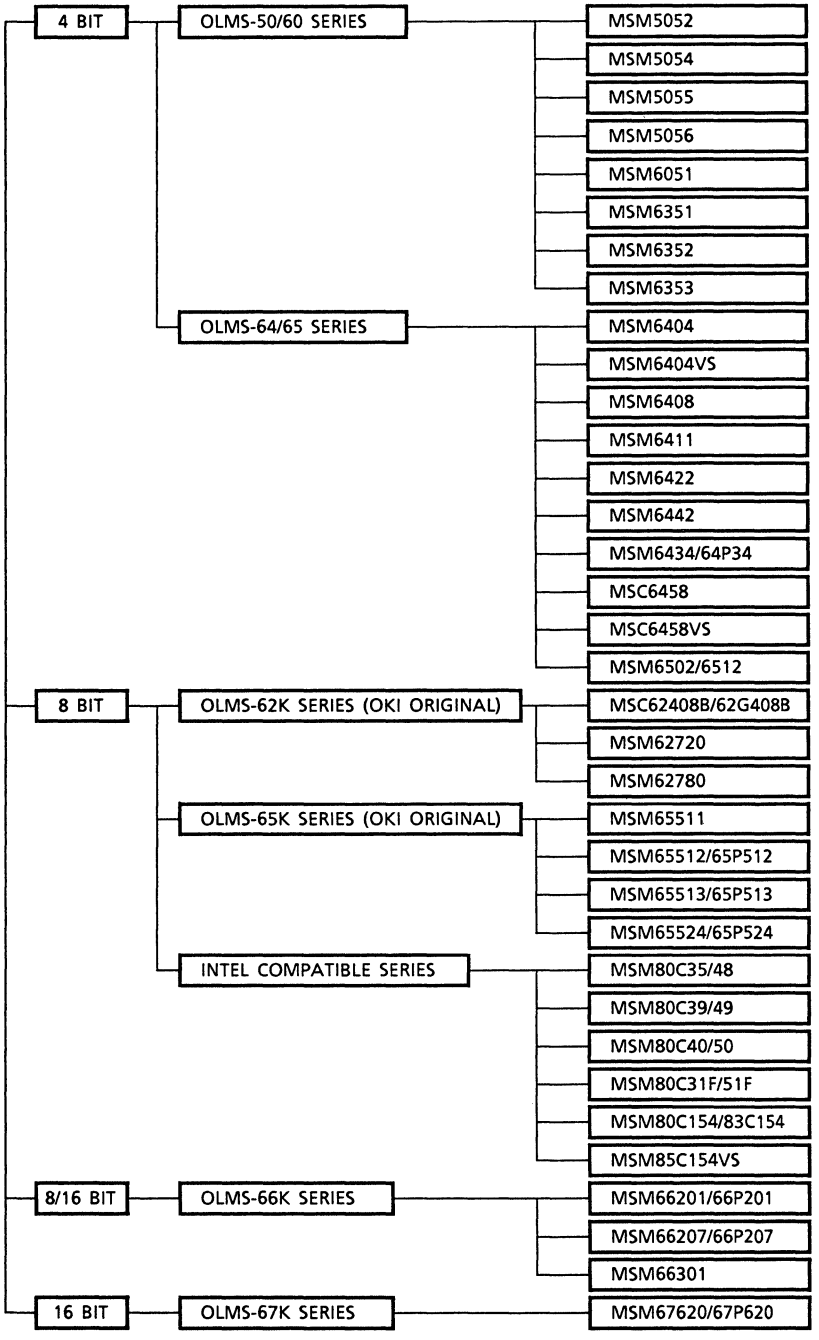
MICRO-CONTROLLERS DATABOOK 1990/1991

CONTENTS

INTRODUCTION 1

- OKI MICROCONTROLLERS LINE-UP
- LINE-UP AND TYPICAL CHARACTERISTIC
- CODE ENTRY
- PROGRAM DEVELOPMENT SYSTEMS

OKI MICROCONTROLLERS LINE-UP



LINE-UP AND TYPICAL CHARACTERISTICS

LOW POWER

OLMS-50/60 SERIES

PRODUCT NAME	ROM	RAM	POWER CONSUMP-TION	FEATURES
MSM5052	1280 x 14	62 x 4	3 µA	52 Seg. LCD Driver
MSM5054	1024 x 14	62 x 4	3 µA	88 Seg. LCD Driver
MSM5055	1792 x 14	96 x 4	3 µA	120 Seg. LCD Driver
MSM5056	1792 x 14	90 x 4	3 µA	76 Seg. LCD Driver
MSM6051	2560 x 14	120 x 4	3 µA	189 Seg. LCD Driver
MSM6352	2048 x 14	640 x 4	0.4 mA	DTMF Generator
MSM6351	4096 x 15	1024 x 4	3 µA	232 Seg. LCD Driver
MSM6353	4096 x 15	1024 x 4	3 µA	Serial Port (Synchronised/ Asynchronised)

HIGH SPEED

OLMS-64/65 SERIES

PRODUCT NAME	ROM	RAM	MACHINE CYCLE	FEATURES
MSM6404	4000 x 8	256 x 4	952 nS	I/O: 36
MSM6408	8096 x 8	256 x 4	1000 nS	I/O: 36
MSM6411	1024 x 8	32 x 4	952 nS	I/O: 11
MSM6422	2048 x 8	64 x 4	952 nS	I/O: 19
MSM6434/ 64P34	4000 x 8	256 x 4	952 nS	A/D Converter
MSM6442	2048 x 8	128 x 4	952 nS	92 Seg. LCD Driver
MSC6458	8192 x 8	512 x 4	930 nS	144 Seg. FLT Driver
MSM6502/ 6512	2000 x 8	128 x 4	91.5 µS	108 Seg. LCD Driver

4 BIT

IC CARD

OLMS-62K SERIES

PRODUCT NAME	ROM	RAM	MACHINE CYCLE	FEATURES
MSM62780	6144 x 8	192 x 8	800 nS	Built-in E^2PROM 8198 x 8 for IC Card
MSM62720	3072 x 8	128 x 8	800 nS	Built-in E^2PROM 2048 x 8 for IC Card
MSM62408 /62G408	15360 x 8	352 x 8	930 nS	384 Seg. FLT Driver

8 BIT

OKI SINGLE CHIP

HIGH PERFORMANCE

INTEL COMPATIBLE SERIES

PRODUCT NAME	ROM	RAM	MACHINE CYCLE
MSM80C35148	1024 x 8	64 x 8	1.36 µS
MSM80C39149	2048 x 8	128 x 8	1.36 µS
MSC80C40150	4096 x 8	256 x 8	2.5 µS
MSM80C51F-1	4096 x 8	128 x 8	750 nS
MSM80C51F	4096 x 8	128 x 8	1 µS
MSM80C31F-1	–	128 x 8	750 nS
MSM80C31F	–	128 x 8	1 µS
MSM83C154-1	188384 x 8	256 x 8	750 nS
MSM83C154	16384 x 8	256 x 8	1 µS
MSM80C154-1	–	256 x 8	750 nS
MSM80C154	–	256 x 8	1 µS

8 BIT

HIGH PERFORMANCE nX SERIES

OLMS-67K SERIES

PRODUCT NAME	ROM	RAM	MACHINE CYCLE
MSM67620	16384 x 16	512 x 8	200 nS
MSM67P620	EPROM 16384 x 16	512 x 8	200 nS

8 BIT

OLMS-66K SERIES

PRODUCT NAME	ROM	RAM	MACHINE CYCLE
MSM66301	16384 x 8	512 x 8	400 nS
MSM66201	16384 x 8	512 x 8	400 nS
MSM66P201	EPROM 16384 x 8	512 x 8	400 nS
MSM66207	32768 x 8	1024 x 8	400 nS
MSM66P207	EPROM 32768 x 8	1024 x 8	400 nS

8/16 BIT

OLMS-65K SERIES

PRODUCT NAME	ROM	RAM	MACHINE CYCLE
MSM65511*	4096 x 8	128 x 8	400 nS
MSM65512*	8197 x 8	256 x 8	400 nS
MSM65P512*	EPROM 8192 x 8	256 x 8	400 nS
MSM65513	8192 x 8	256 x 8	400 nS
MSM65P513	EPROM 8192 x 8	256 x 8	400 nS
MSM65524	16384 x 8	384 x 8	400 nS
MSM65P524	EPROM 16384 x 8	384 x 8	400 n/s

8 BIT

MICROCOMPUTER

*: Under Development

■ OLMS-50/60 4-Bit Series Single-Chip Microcontrollers

Model	Function	Process	Supply Voltage (V_{DD})	Supply Current (I_{DD})	ROM Capacity (Bits)	RAM Capacity (Bits)	Input Ports	Output Ports	LCD Output	Package
MSM5052-XX	Low Power 4 Bit Single Chip Microcontroller with LCD Driver and Thermometer Circuit	CMOS	1.5V	3μA	1280x14	62x4	8	5	26 segments 2 common	DIE FORM
MSM5054-XX	Low Power 4 Bit Single Chip Microcontroller with LCD Driver	CMOS	1.5/3V	3/1.5μA	1024x14	62x4	6	4	44 segments 2 common	DIE FORM
MSM5055-XX	Low Power 4 Bit Single Chip Microcontroller with LCD Driver	CMOS	1.5/3V	3/1.5μA	1792x14	96x4	8	4	60 segments 2 common	DIE FORM
MSM5056-XX	Low Power 4 Bit Single Chip Microcontroller with LCD Driver and Solar Cell Circuit	CMOS	1.5V	3μA	1792x14	90x4	4	4	38 segments 2 common	DIE FORM
MSM6051-XX	Low Power 4 Bit Single Chip Microcontroller with LCD Driver	CMOS	1.5/3V	3/1.5μA	2560x14	120x4	9	4	63 segments 3 common	DIE FORM
MSM6351-XX	Low Power 4 Bit Single Chip Microcontroller with LCD Driver	CMOS	1.5/3V	3μA	4096x15	1024x4	–	–	58 segments 4 common 20 I/O ports	100 QFP or DIE FORM
MSM6352-XX	Low Power 4 Bit Single Chip Microcontroller with DTMF Circuit	CMOS	20/5.5V	400μA	2048x14	640x4	12	12	4 I/O ports	28/40DIP 44 QFP
MSM6353-XX	Low Power 4 Bit Single Chip Microcontroller	CMOS	1.5/3V	3μA	4096x15	1024x4	–	–	20 I/O ports	42SDIP

■ OLMS-64/65 Series 4-Bit Single-Chip Microcontrollers

Model	Function	Process	Supply Voltage (V_{DD})	ROM Capacity (Bits)	RAM Capacity (Bits)	Input Pors	Output Ports	I/O Ports	Package
MSM6404A-XX	High Performance 4 Bit Single Chip Microcontroller	CMOS	5V	4Kx8	256x4	4	—	32	42 DIP 44 QFP
MSM6411B-XX	4 Bit Single Chip Microcontroller	CMOS	5V	1Kx8	32x4	4	—	8	16 DIP
MSM6422-XX	4 Bit Single Chip Microcontroller	CMOS	5V	2Kx8	64x4	4	—	14	24 DIP 24 SOP
MSM6434-XX	4 Bit Single Chip Microcontroller with A/D Converter	CMOS	5V	4Kx8	256x4	1	—	12	30 SDIP
MSM6408-XX	4 Bit Single Chip Microcontroller	CMOS	5V	8Kx8	256x4	4	—	32	42 DIP 44 QFP
MSM6442-XX	4 Bit Single Chip Microcontroller with LCD Driver	CMOS	5V	2Kx8	128x4	4	LCD46 (DC) /92 (1/2 duty) buzzer 1	32	80 QFP
MSC6458-XX	4 Bit Single Chip Microcontroller with FLT Driver	Bi-CMOS	5V	8Kx8	512x4	9	FLT 12(Tim)/ 12(Seg)	24	64 SDIP
MSM6502B-XX	Low Power 4 Bit Single Chip Microcontroller with LCD Driver	CMOS	3V	2Kx8	128x4	4	LCD segment output 108	8	56 QFP (small)
MSM6512-XX	Low Power 4 Bit Single Chip Microcontroller with LCD Driver	CMOS	3V	2Kx8	128x4	4	LCD segment output 108	8	56 QFP (small)

■ OLMS-62K 8-Bit Series Single-Chip Microcontrollers

Model	Function	Process	Supply Voltage (V_{DD})	ROM Capacity (Bits)	E²PROM Capacity (Bits)	RAM Capacity (Bits)	Input Ports	Output Ports	I/O Ports	Package
MSM62720-XX	8 Bit Single Chip Microcontroller with E²PROM	CMOS	5V	3Kx8	2Kx8	128x8	—	—	Serial interface	DIE FORM
MSM62780-XX	8 Bit Single Chip Microcontroller with E²PROM	CMOS	5V	6Kx8	8Kx8	192x8	—	—	Serial interface	DIE FORM
MSC62408-XX* /62G408	8 Bit Single Chip Microcontroller with FLT Driver	Bi-CMOS	5V	15Kx8	—	352x8	9	FLT8 (Tim)/ 24 (Seg)/ 8 (Tim or Seg)	24	80 QFP

* Under development

■ OLMS-65K 8-Bit Series Single-Chip Microcontrollers

Model	Function	Process	Supply Voltage (V_{DD})	ROM Capacity (Bits)	RAM Capacity (Bits)	Input Ports	Output Ports	I/O Ports	Package
MSM65511-XX*	High Performance 8 Bit Single Chip Microcontroller	CMOS	5V	4Kx8	128x8	—	—	32	40 DIP, 44/64 QFP 44 PLCC
MSM65512-XX* /65P512 (OTP)	High Performance 8 Bit Single Chip Microcontroller	CMOS	5V	8Kx8	256x8	—	—	32	40 DIP, 44/64 QFP 44 PLCC
MSM65513-XX* /65P513 (OTP)	High Performance 8 Bit Single Chip Microcontroller	CMOS	5V	8Kx8	256x8	8	—	48	64 SDIP, 64 QFP (small), 68 PLCC
MSM65524-XX* /65P524 (OTP)	High Performance 8 Bit Single Chip Microcontroller	CMOS	5V	16Kx8	384x8	8	—	44	64 SDIP, 64 QFP (small), 68 PLCC

* Under development

■ Intel Compatible 8-Bit Series Single-Chip Microcontrollers

Model	Function	Process	Supply Voltage (V_{DD})	Package	Remarks
MSM80C35	8 Bit Single Chip Microcontroller	CMOS	5V	40 DIP 44 QFP	External ROM
MSM80C39	8 Bit Single Chip Microcontroller	CMOS	5V	40 DIP 44 QFP	External ROM
MSM80C48-XX	8 Bit Single Chip Microcontroller	CMOS	5V	40 DIP 44 QFP	
MSM80C49-XX	8 Bit Single Chip Microcontroller	CMOS	5V	40 DIP 44 QFP	
MSM80C40	8 Bit Single Chip Microcontroller	CMOS	5V	40 DIP 44 QFP	External ROM
MSM80C50-XX	8 Bit Single Chip Microcontroller	CMOS	5V	40 DIP 44 QFP	
MSM80C31F	8 Bit Single Chip Microcontroller	CMOS	5V	40 DIP, 44 QFP 44 PLCC	External ROM
MSM80C51F-XX	8 Bit Single Chip Microcontroller	CMOS	5V	40 DIP, 44 QFP 44 PLCC	
MSM80C154	8 Bit Single Chip Microcontroller	CMOS	5V	40 DIP, 44 QFP 44 PLCC	External ROM
MSM83C154-XX	8 Bit Single Chip Microcontroller	CMOS	5V	44 DIP, 44 QFP 44 PLCC	
MSM85C154	8 Bit Single Chip Microcontroller	CMOS	5V	Ceramic, piggyback	
MSM6404VS	High Performance 4 Bit Single Chip Microcontroller	CMOS	5V	Ceramic, piggyback	
MSM6458VS	4 Bit Single Chip Microcontroller with FLT Driver	Bi-CMOS	5V	Ceramic, piggyback	
MSM62G408	8 Bit Single Chip Microcontroller with FLT Driver	Bi-CMOS	5V	Ceramic, piggyback Under development	
MSM66G301	High Performance 8/16 Bit Single Chip Microcontroller	CMOS	5V	Ceramic, piggyback	

■ OLMS-66K 8/16-Bit Series Single-Chip Microcontrollers

Model	Function	Process	Supply Voltage (V_{DD})	ROM Capacity (Bits)	RAM Capacity (Bits)	Input Ports	Output Ports	I/O Ports	Package
MSM66301-XX	High Performance 8/16 Bit Single Chip Microcontroller	CMOS	5V	16Kx8	512x8	8	—	40	64 SDIP 64 QFP (small)
MSM66201-XX* /66P201 (OTP)	High Performance 8/16 Bit Single Chip Microcontroller	CMOS	5V	16Kx8	512x8	8	—	40	64 SDIP 64 QFP (small)
MSM66207-XX* /66P207 (OTP)	High Performance 8/16 Bit Single Chip Microcontroller	CMOS	5V	32Kx8	1024x8	8	—	40	64 SDIP 64 QFP (small)

* Under development

■ OLMS-67K 16-Bit Series Single-Chip Microcontroller

Model	Function	Process	Supply Voltage (V_{DD})	ROM Capacity (Bits)	RAM Capacity (Bits)	Input Ports	Output Ports	I/O Ports	Package
MSM67620-XX /67P620 (OTP)	High Performance 16 Bit Single Chip Microcontroller	CMOS	5V	16Kx16	512x8	—	—	56	64 SDIP 64 QFP (small) 68 PLCC

CODE ENTRY

The program code ENTERING method is outlined below.

1. USABLE MEDIA

(1) 2 pieces of same type EPROMs containing identical DATA

EPROM specification

- 2716
- 2732
- 27C32
- 27C32A
- 2764
- 27C64
- 27128
- 27256

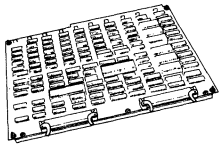
(2) 1 copy of object machine code list

2. SINGLE CHIP MICROCONTROLLER DEVELOPMENT STAGES

USER 1

● Program with OKI development tool.

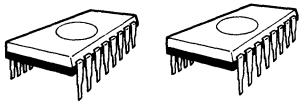
if 800



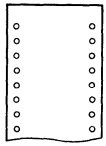
TOOL

USER 2

● Prepare 2 pcs of EPROM and programming list



EPROM



Printout

OKI 3

● Print out identical EPROM data.

OKI 4

● Engineering Sample
Actual production sample of your microcontroller chip prepared for final approval.

OKI 5

● Volume production of single chip microcontrollers.

USER 6

START
PRODUCT/SYSTEM
MANUFACTURE

Program Development System

■ Low Power Series

Object Chip	System Name	Standard Software	
MSM5052	EASE5052/56	EASE host monitor	ASM50 cross assembler
MSM5056			
MSM5054	EASE5054/55		
MSM5055			
MSM6051	EASE6051		
MSM6351	EASE6351/53		
MSM6353			
MSM6352	EASE6352		

Note: Standard software adaptable to following operating systems.
CP/M-80: Various personal computer models
MS-DOS: OKI if800, NEC9801 and equivalents
PC-DOS: IBM PC-XT, AT and 5550

■ OLMS-64 Series

Object Chip	System Name	Adaptor Module	Standard Software		Field Dubugging Tool	Piggyback
MSM6408	EASE6400	—	EASE host monitor	ASM6400	—	
MSM6404A		—			—	MSM6404VS
MSM6422		PAM6422			PEM6422	—
MSM6411B		PAM6411			PEM6411	
MSM6434		PAM6434			PEM6434	
MSM6442		PAM6442			PEM6442	
MSC6458	EASE6458	—		ASM6458	—	MSC6458VS

Note: Standard software adaptable to following operating systems.
EASE, ASM6400: CP/M-80 (various personal computer models)
ASM6458: MS-DOS (for OKI if800, NEC9801 and equivalents)
PC-DOS (for IBM PC-XT, AT and 5550)

■ OLMS-65 Series

Object Chip	System Name	Standard Software	Field Debugging Tool
MSM6502B MSM6512	EASE6502	EASE65 ASM6502	MPB6502EVA

Note: Standard software adaptable to CP/M-80, MS-DOS and PC-DOS.

■ Intel Compatible 8-Bit Series

Object Chip	System Name	Standard Software	Optional Software	Piggyback
MSM80C48	EASE80C49	EASE49 host monitor ASM49 assembler	—	—
MSM80C49				
MSM80C50				
MSM80C51	EASE80C51mkII	EASE host monitor ASM51 assembler	See note 1	MSM85C154VS
MSM83C154				

Notes: 1. Optional software for MSM80C51 family is as follows:
PASM pre-processor, MAC51 relocatable assembler, RL51 linker, LIB51 librarian, SID51 symbolic debugger.

2. Above software can be used in following operating systems:
   - EASE49: CP/M-80, MS-DOS
   - ASM49: CP/M-80, MS-COS
   - EASE, ASM51: CP/M-80 for Oki if800, NEC PC8801, etc.
     MS-DOS for Oki if800, NEC PC9801, etc.
     PC-DOS for IBM PC-XT, AT and IBM5550
   - PASM, MAC51, RL51, LIB51: CP/M-80 for Oki if800, NEC8801, etc.

■ OLMS-62K/ – 65K/ – 66K/ – 67K Series

Object Chip	System Name	Standard Software	Optional Software
MSC62408	OMFICE62408	AS62400, EASE	—
MSM62720 MSM62780	EASE62780	AS62780, EASE	—
MSM65524/65511/ 65512/65513	OMFICE + EVM65524 *	RAS65K, RL65K, LIB65K, SID65K *	
MSM66301	EASE66301	RAS66K, RL66K, LIB66K, SID66K	CC66K * SDB66K *
MSM66201/66207	EASE66201 *		
MSM67620	OMFICE + EVM67620 *	RAS67K, RL67K, LIB67K, SID67K *	CC67K *

* Under development

PACKAGING

PACKAGING

Product	No. of Pin	DIE FORM	DIP	SDIP	PIGGY BACK	SOP/QFP				QFJ (PLCC)
			RS	SS	VS	GS-K	GS-1K	GS-VK	GS-VIK	JS
MSM5052	56(s)	○				○		○		
	80					○				
MSM5054	56(s)	○				○				
	80					○				
MSM5055	80	○				○				
MSM5056	–	○								
MSM6051	–	○								
MSM6351	100	○				○				
MSM6352	28		○							
	40		○							
	42			○						
	44					○		○		
MSM6353	42			○						
	44					○			○	
MSM6404	42	○	○							
	44					○		○	○	
MSM6404VS	42				○					
MSM6408	42	○	○							
	44					○			○	
MSM6411	16	○	○							
	24					○				
MSM6422	24	○	○			○				
MSM6442	80	○				○				
MSM6434	30			○						
	40		○							
	44									○
MSM64P34	30			○						
MSC6458	64			○						
MSC6458VS	64				○					
MSM6502	56(s)	○				○				
MSM6512	56(s)	○				○				
MSC62408B	80								○	
MSC62G408B	80				○					
MSM62720	–	○								
MSM62780	–	○								
MSM65511	40		○							
	44					T.B.D.				○
MSM65512	40		○							
	44					T.B.D.				○
	64					T.B.D.				
MSM65P512	40		○							
	44									○
	64					T.B.D.				
MSM65513	64			○		T.B.D.				
	68									○
MSM65P513	64			○		T.B.D.				
	68									○

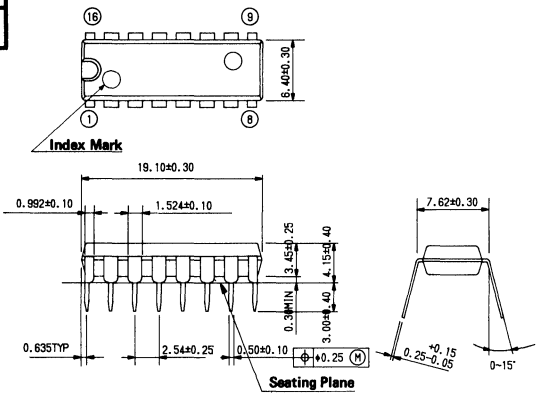
• PACKAGING •

Product	No. of Pin	DIE FORM	DIP	SDIP	PIGGY BACK	SOP/QFP				QFJ (PLCC)
			RS	SS	VS	GS-K	GS-1K	GS-VK	GS-VIK	JS
MSM65524	64			○		T.B.D.				
	68									○
MSM65P524	64			○		T.B.D.				
	68									○
MSM80C35	40		○							
	44					○		○		
MSM80C48	40		○							
	44					○		○		
MSM80C39	40		○							
	44					○		○		
MSM80C49	40		○							
	44					○		○		
MSM80C40	40		○							
	44					○		○		
MSM80C50	40		○							
	44					○		○		
MSM80C31F	40		○							
	44					○			○	○
MSM80C51F	40		○							
	44					○			○	○
MSM80C154	40		○							
	44					○			○	○
MSM83C154	40		○							
	44					○			○	○
MSM85C154VS	40				○					
MSM66201	64			○			○			
	68									○
MSM66P201	64			○						
	68									○
MSM66207	64			○						
	68									○
MSM66P207	64			○						
	68									○
MSM66301	64			○		○				
	68									○
MSM67620	64			○		T.B.D.				
	68									○
MSM67P620	64			○						
	68									○

PLASTIC DIP

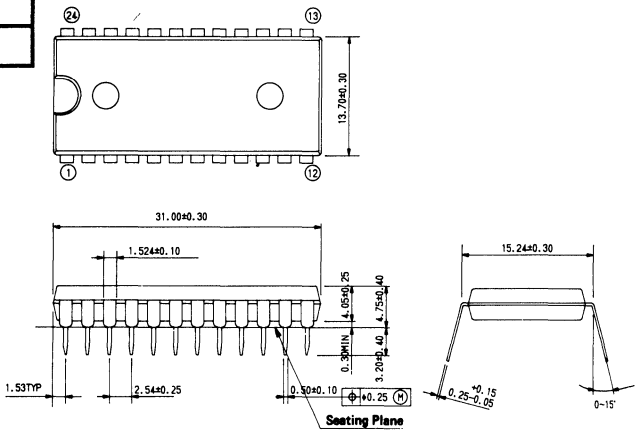
16PIN PLASTIC DIP

DIP16-P-300



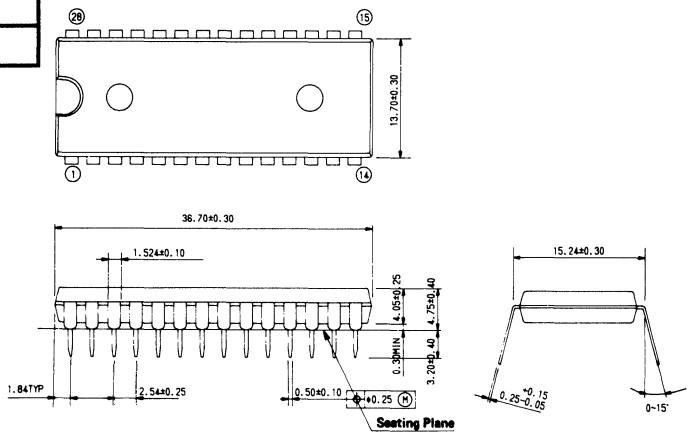
24PIN PLASTIC DIP

DIP24-P-600



28PIN PLASTIC DIP

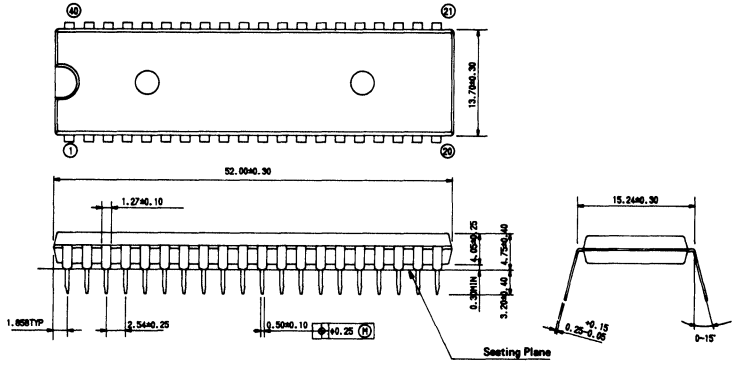
DIP28-P-600



PLASTIC DIP

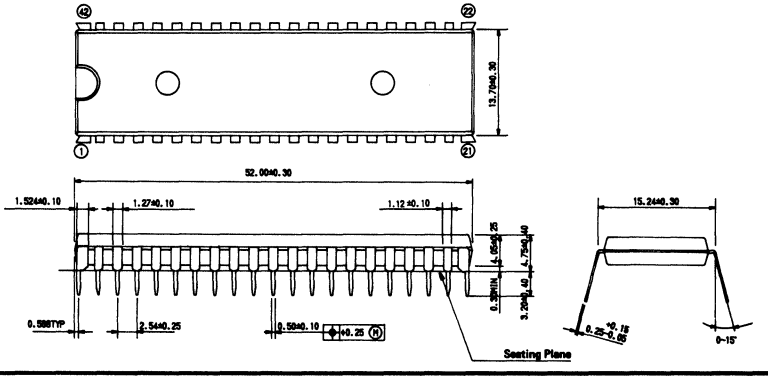
40PIN PLASTIC DIP

DIP40-P-600



42PIN PLASTIC DIP

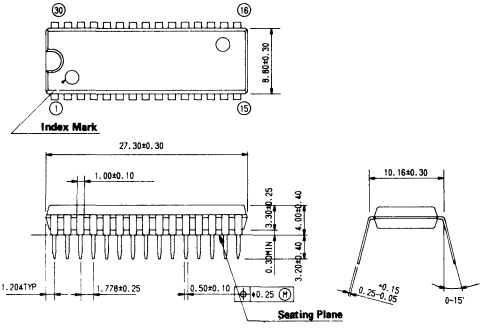
DIP42-P-600



PLASTIC SDIP

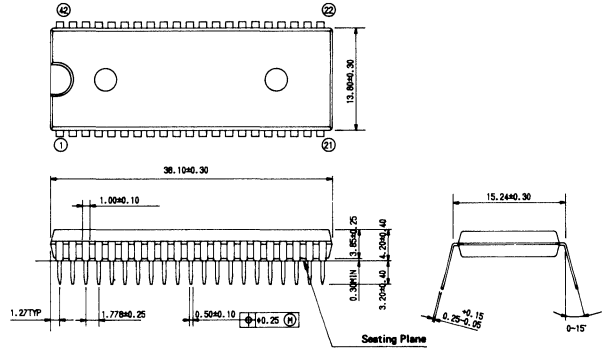
30PIN PLASTIC SHRINK DIP

SDIP30-P-400



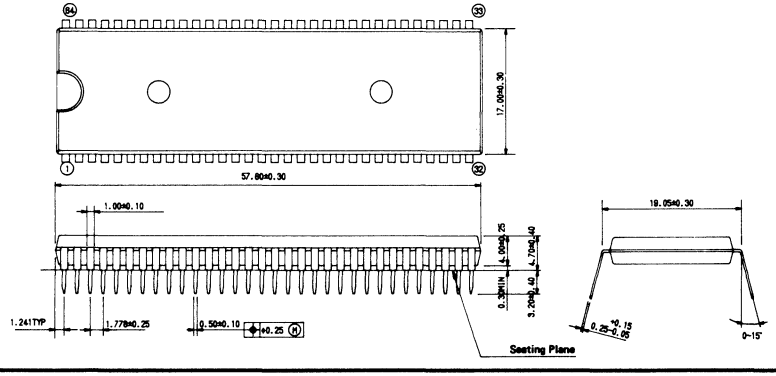
42PIN PLASTIC SHRINK DIP

SDIP42-P-600



64PIN PLASTIC SHRINK DIP

SDIP64-P-750

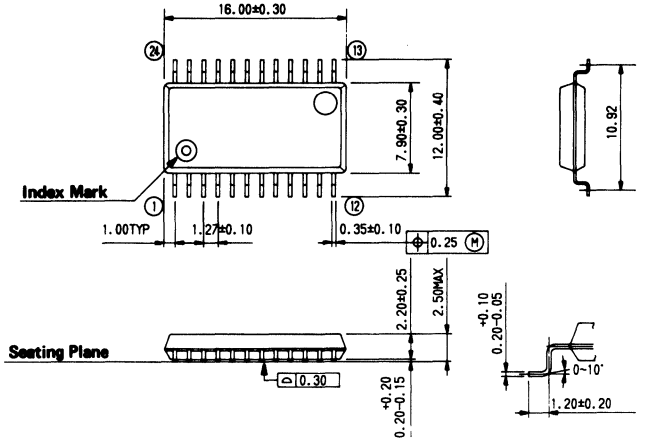


2

PLASTIC SOP

24PIN PLASTIC SOP

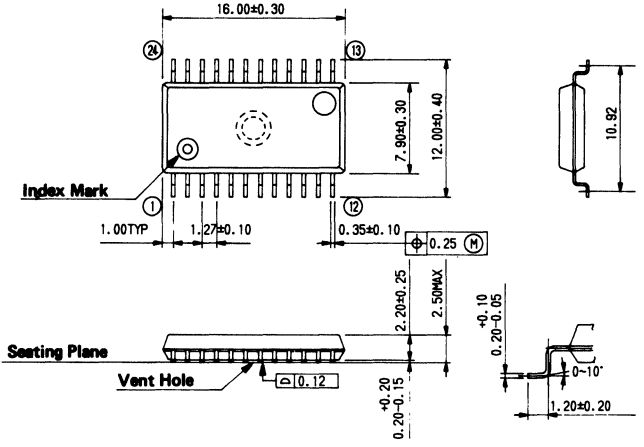
SOP24-P-430-K



PLASTIC SOP

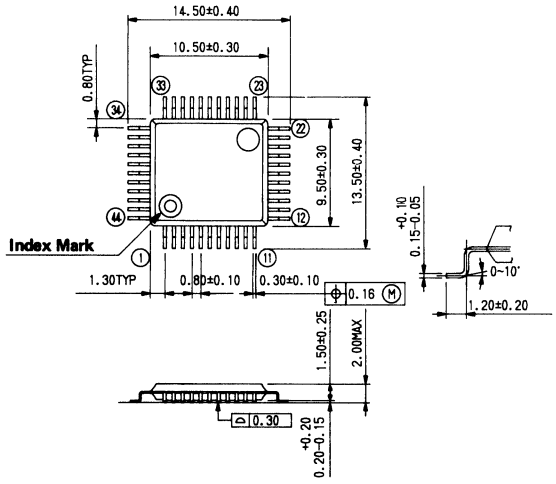
24PIN-V PLASTIC SOP

SOP24-P-430-VK

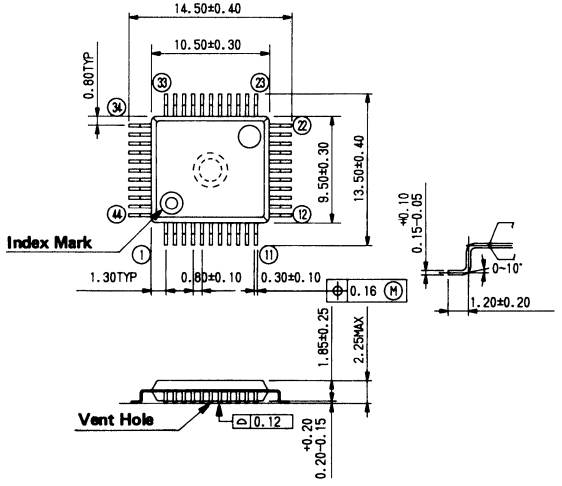


PLASTIC QFP

44PIN PLASTIC QFP
QFP44-P-910-K

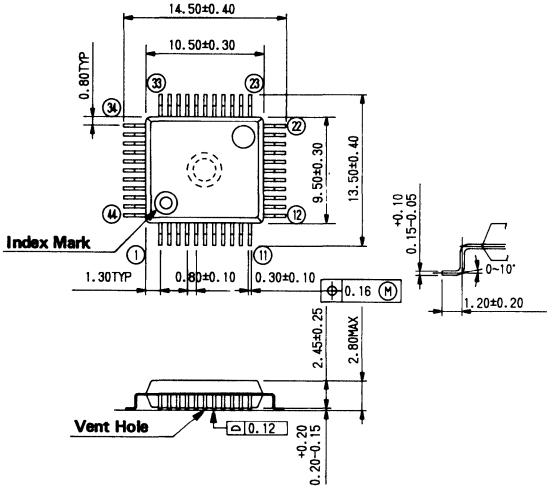


44PIN PLASTIC QFP
QFP44-P-910-VIK

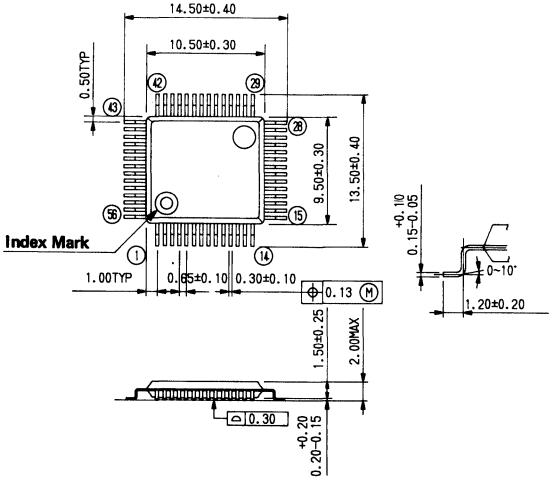


PLASTIC QFP

44PIN-V PLASTIC QFP
QFP44-P-910-VK



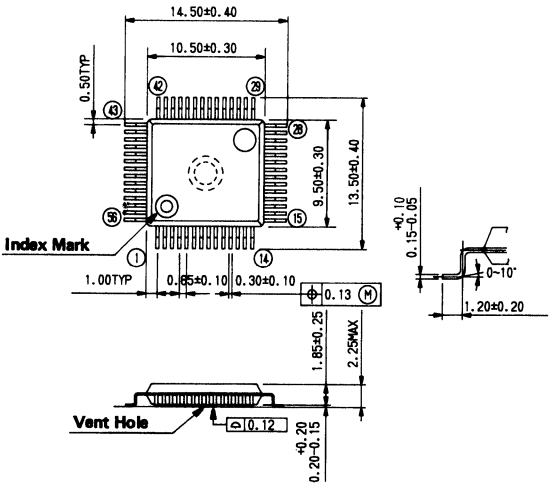
56PIN(S) PLASTIC QFP
QFP56-P-910-K



PLASTIC QFP

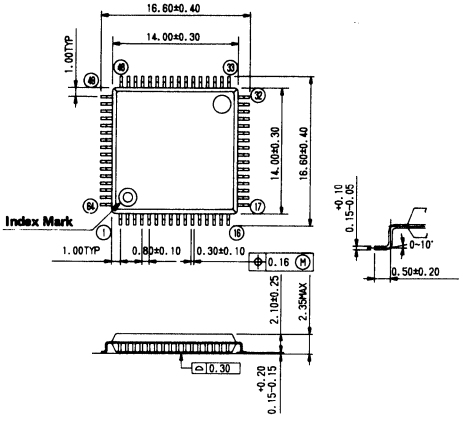
56PIN(S)-V PLASTIC QFP

QFP56-P-910-VK



64PIN PLASTIC QFP

QFP64-P-1414-K

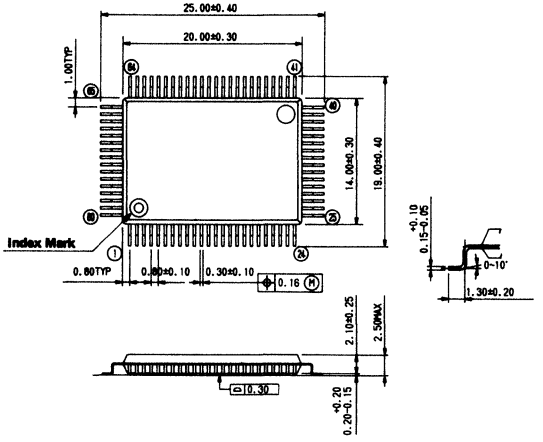


PLASTIC QFP

2

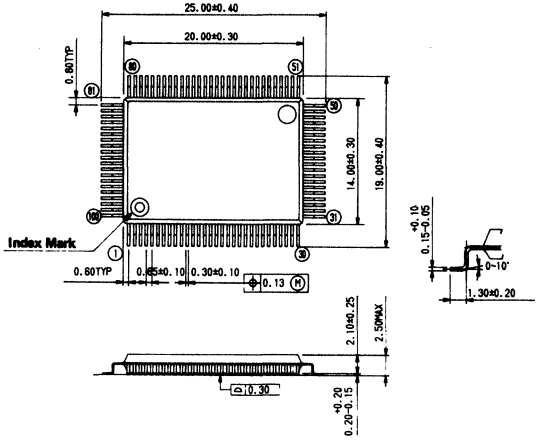
80PIN PLASTIC QFP

QFP80-P-1420-K



100PIN PLASTIC QFP

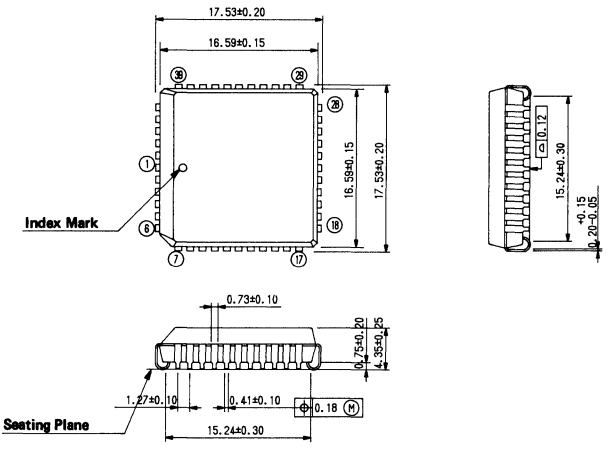
QFP100-P-1420-K



PLASTIC QFJ (PLCC)

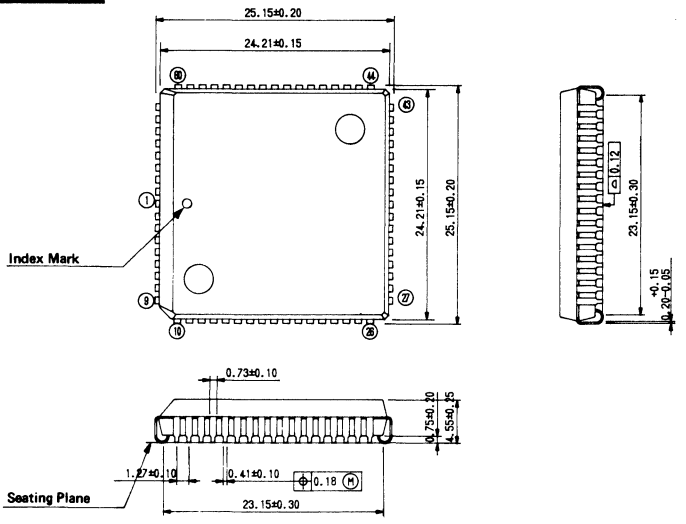
44PIN PLCC

QFJ44-P-S650



68PIN PLCC

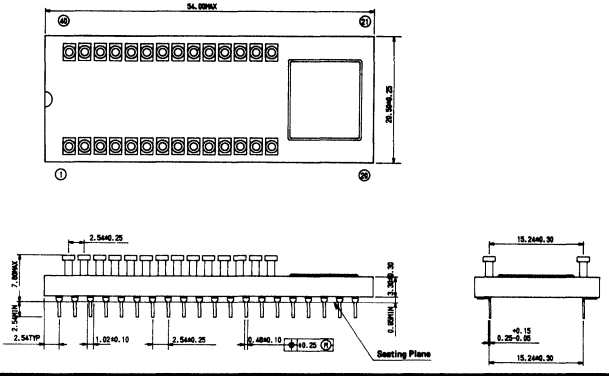
QFJ68-P-S950



CERAMIC PIGGY BACK (Evaluation Sample Only)

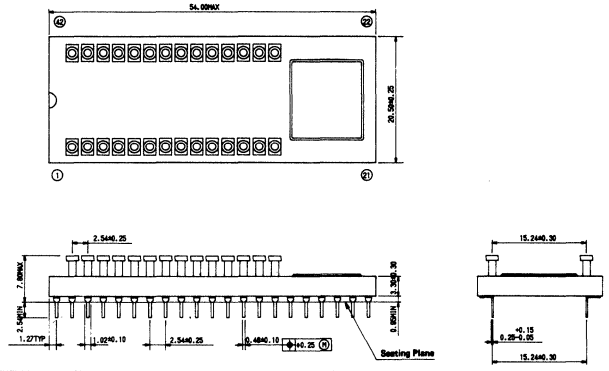
40PIN CERAMIC PIGGY BACK

ADIP40-C-600



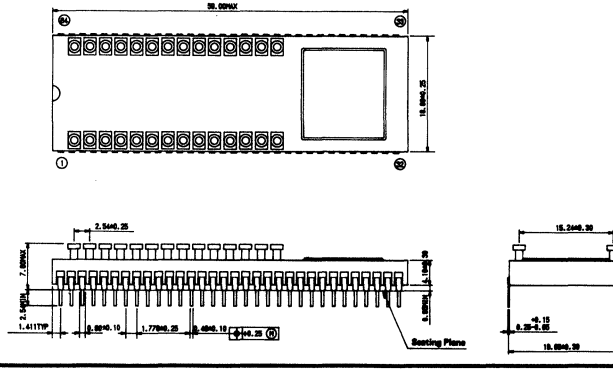
42PIN CERAMIC PIGGY BACK

ADIP42-C-600



64PIN CERAMIC PIGGY BACK

ADIP64-C-750



RELIABILITY INFORMATION

RELIABILITY INFORMATION

1. INTRODUCTION

Semiconductor devices play a leading role in the explosive progress of technology. They use some of the most advanced design and manufacturing technology developed to date. With greater integration, diversity and reliability, their applications have expanded enormously. Their use in large scale computers, control equipment, calculators, electronic games and in many other fields has increased at a fast rate.

A failure in electronic banking or telephone switching equipment, for example, could have far reaching effects and can cause incalculable losses. So, the demand, for stable, high quality memory devices is strong.

We, at Oki are fully aware of this demand. So we have adopted a comprehensive quality assurance system based on the concept of consistency in development, manufacturing and sales.

With the increasing demand for improvement in function, capability and reliability, we will expand our efforts in the future. Our quality assurance system and the underlying concepts are outlined briefy below.

2. QUALITY ASSURANCE SYSTEM AND UNDERLYING CONCEPTS

The quality assurance system employed by Oki can be divided into four major stages: device planning, developmental prototype, production prototype, and mass production. This system is outlined in the following block diagram (Fig. 1).

1) Device planning stage

To manufacture devices that meet market demands and satisfy customer needs, we carefully consider functional and failure rate requirements, utilization form, environment and other conditions. Once we determine the proper type, material and structure, we check the design and manufacturing techniques, and the line processing capacity. Then we prepare the development planning and time schedule.

2) Developmental prototype stage

We determine circuits, pattern design, process settings, assembly techniques and structural requirements during this stage. At the same time, we carry out actual prototype reliability testing.

Since device quality is largely determined during the designing stage, Oki pays careful attention to quality confirmation during this stage.

This is how we do it:

(1) After completion of circuit design (or pattern design), personnel from the design, process technology, production technology, installation technology and reliability departments get together for a thorough review to ensure design quality and to anticipate problems that may occur during mass production. Past experience and know-how guide these discussions.

(2) Since many semiconductor memories involve new concepts and employ high level manufacturing technology, the TEG evaluation test is often used during this stage.
Note: TEG (Test Element Group) refers to the device group designed for stability evaluation of MOS transistors, diodes, resistors, capacitors and other circuit component element used in LSI memories.

(3) Prototypes are subjected to repeated reliability and other special evaluation tests. In addition, the stability and capacity of the manufacturing process are checked.

3) Production prototype stage

During this stage, various tests check the reliability and other special features of the production prototype at the mass production factory level. After confirming the quality of a device, we prepare the various standards required for mass production, and then start production. Although reliability and other special tests performed on the production prototype are much the same as those performed on the developmental prototype, the personnel, facilities and production site differ for the two prototypes, necessitating repeated confirmation tests.

4) Mass production

During the mass production stage, careful management of purchased materials, parts and facilities used during the manufacturing process, measuring equipment, manufacturing conditions and environment is necessary to ensure device quality first stipulated during the designing stages. The manufacturing process (including inspection of the completed device) is followed by a lot guarantee inspection to check that the specified quality is maintained under conditions identical to those under which a customer would actually use the device. This lot guarantee inspection is performed in three different forms as shown below.

(1) Group A tests: appearance, labels, dimensions and electrical characteristics inspection
(2) Group B tests: check of durability under thermal and mechanical environmental stresses, and operating life characteristics
(3) Group C tests: performed periodically to check operational life, etc., on a long term basis.

Note: Like the reliability tests, the group B tests conform to the following standards.
MIL-STD-883B, JIS C 7022, EIAJ-IC-121

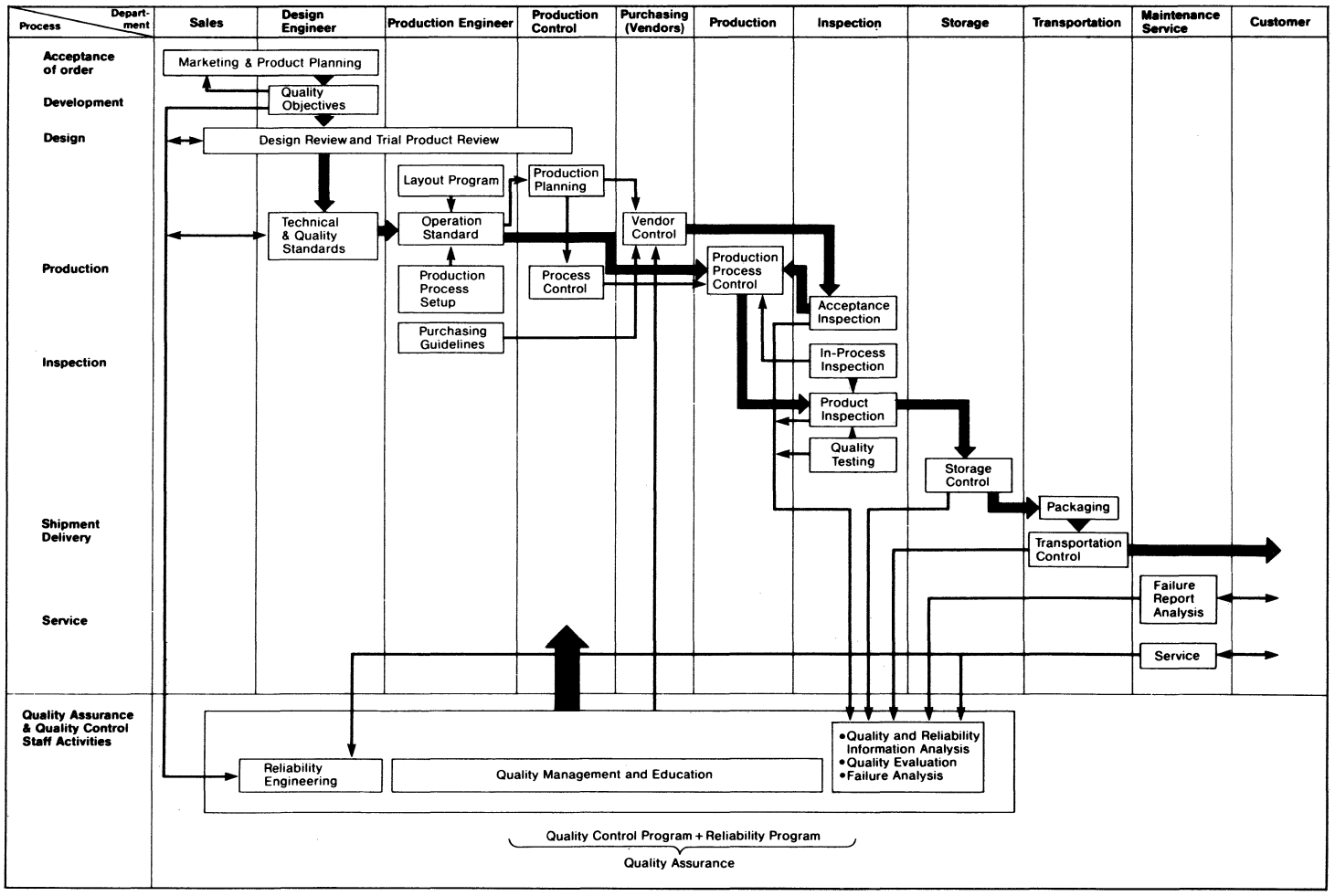


Figure 1 Quality Assurance System

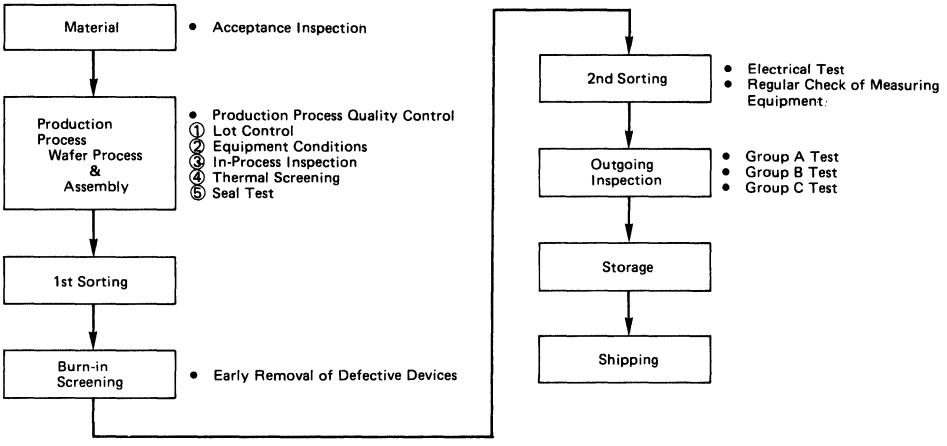


Figure 2 Manufacturing Process

Devices which pass these lot guarantee inspections are stored in a warehouse awaiting shipment to customers. Standards are also set up for handling, storage and transportation during this period, thereby ensuring quality prior to delivery. Figure 2 shows the manufacturing flow of the completed device.

5) At Oki, all devices are subjected to thorough quality checks. If, by chance, a failure does occur after delivery to the customer, defective devices are processed and the problem rectified immediately to minimize the inconvenience to the customer in accordance with the following flowchart.

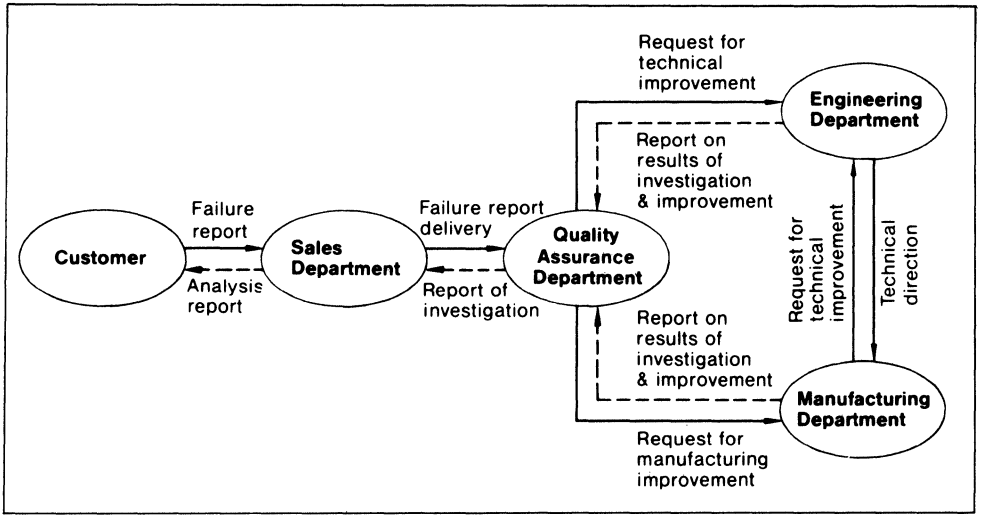
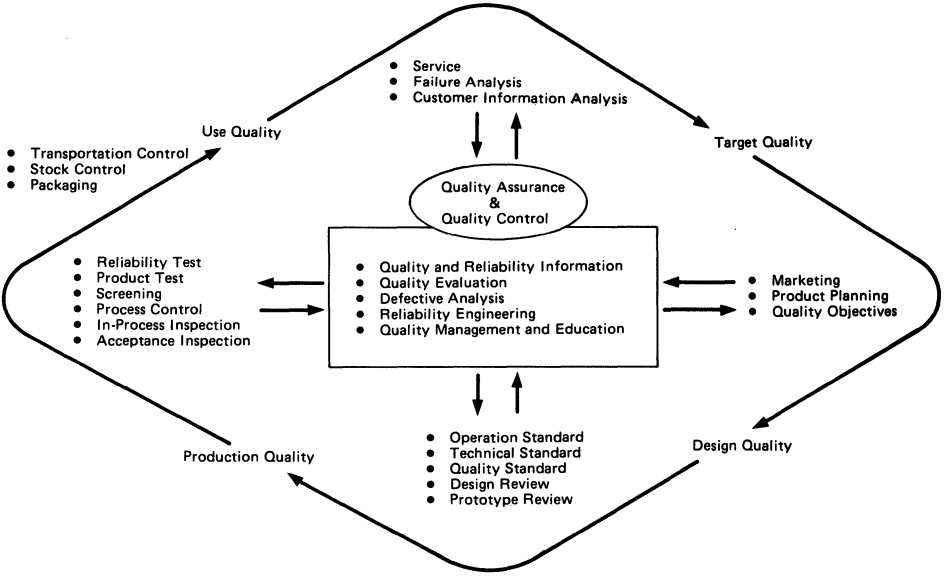


Figure 3 Failure report process



3. EXAMPLE OF RELIABILITY TEST RESULTS

We have outlined the quality assurance system and the underlying concepts employed by Oki. Now, we will give a few examples of the reliability tests performed during the developmental and production prototype stages. All reliability tests performed by Oki conform to the following standards.

MIL-STD-883B, JIS C 7022, EIAJ-IC-121

Since these reliability tests must determine performance under actual working conditions in a short period of time, they are performed under severe test conditions. For example, the 125°C high temperature continuous operation test performed for 1000 hours is equivalent to testing device life from 2 to 300 years of use at Ta = 40°C.
By repeating these accelerated reliability tests, device quality is checked and defects analyzed. The resulting information is extremely useful in improving the manufacturing processes. Some of the more common defects in LSI elements and their analysis are described on next page.

MICROCONTROLLER LIFE TEST RESULTS

Part name		MSM80C31/51-XXRS			MSM80C35/39/48/49-XXRS			MSM83C154-XXRS			Referred standard
Function		8 BIT ONE CHIP MICROCONTROLLER			8 BIT ONE CHIP MICROCONTROLLER			8 BIT ONE CHIP MICROCONTROLLER			
Test item	Test condition	Sample size (pcs)	Test hours	Failures	Sample size (pcs)	Test hours	Failures	Sample size (pcs)	Test hours	Failures	
Operating life test	Ta = 125°C Vcc = 6V	88	2000 (H)	0	88	2000 (H)	0	88	2000 (H)	0	MIL-STD-883C METHOD 1005
Temperature humidity test	Ta = 85°C RH = 85% Vcc = 6V	100	2000 (H)	0	100	2000 (H)	0	100	2000 (H)	0	
Temperature cycling test	−55°C⇄RT⇄150°C (30 min) ↑ (30 min) (5 min)	100	500 (cy)	0	100	300 (cy)	0	100	500 (cy)	0	MIL-STD-883C METHOD 1010
Pressure cooker test	Ta = 121°C RH = 100% 2 atm	50	200 (H)	0	50	200 (H)	0	50	200 (H)	0	

Part name		MSM6404-XXJS			MSM80C31JS			Referred standard
Function		4 BIT ONE CHIP MICROCONTROLLER			8 BIT ONE CHIP MICROCONTROLLER			
Test item	Test condition	Sample size (pcs)	Test hours	Failures	Sample size (pcs)	Test hours	Failures	
Operating life test	Ta = 125°C Vcc = 6V	88	2000 (H)	0	88	2000 (H)	0	MIL-STD-883C METHOD 1005
Temperature humidity test	Ta = 85°C RH = 85% Vcc = 6V	100	2000 (H)	0	100	2000 (H)	0	
Temperature cycling test	−55°C⇄RT⇄150°C (30 min) ↑ (30 min) (5 min)	100	500 (cy)	0	100	500 (cy)	0	MIL-STD-883C MEJTHOD 1010
Pressure cooker test	Ta = 121°C RH = 100% 2 atm	50	200 (H)	0	50	200 (H)	0	

3

MICROCONTROLLER ENVIRONMENTAL TEST RESULTS

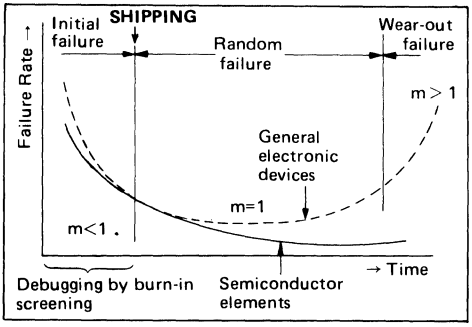
Test item		Test condition	MSM80C31/51-XXRS 8 BIT ONE CHIP MICROCONTROLLER Sample size (pcs)	Failures	MSM80C35/39/48/49-XXRS 8 BIT ONE CHIP MICROCONTROLLER Sample size (pcs)	Failures	MSM83C154-XXRS 8 BIT ONE CHIP MICROCONTROLLER Sample size (pcs)	Failures	Referred standard
Soldering Heat Test		260°C 10 SEC	22	0	22	0	22	0	MIL-STD-883C METHOD 2003
Temperature Cycling Test		−55°C⇄RT⇄150°C (30min) (5min) (30min) 20 cycles	22	0	22	0	22	0	MIL-STD-883C METHOD 1010
Thermal Shock Test		100°C⇄0°C (5min) (5min) 10 cycles	22	0	22	0	22	0	MIL-STD-883C METHOD 1011
Lead Integrity	Tensile	500 g 10 SEC	11	0	11	0	11	0	MIL-STD883C METHOD 2004
	Bending	250 g 90° BEND 3 TIMES							
Solderability		230°C 5 SEC	22	0	22	0	22	0	MIL STD883C METHOD 2003

	Test item	Test condition	MSM6404 XXJS 4 BIT ONE CHIP MICROCONTROLLER Sample size (pcs)	Failures	MSM80C31JS 8 BIT ONE CHIP MICROCONTROLLER Sample size (pcs)	Failures	Referred standard
Thermal Environmental Test	PRE-Bake	Bake (125°C, 6 hrs)	22	0	22	0	
	Soldering Heat Test	Vapor Phase Reflow (215±2°C, 90+10, −0sec) 2 times					
	Temperature Cycling Test	−55°C⇄RT⇄150°C (30min) (5min) (30min) 20 cycles					MIL-STD-883C METHOD 1010
	Thermal Shock Test	100°C⇄0°c (5min) (5min) 10 cycles					MIL-STD-883C METHOD 1011
Other Test	Solderability	Bake (125°C, 24hrs) → Immerse into Flux → Immerse into Solder (215±2°C 10±1sec)	22	0	22	0	

4. SEMICONDUCTOR MEMORY FAILURES

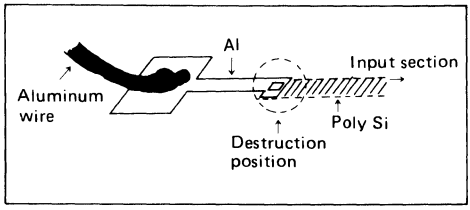
The life-span characteristics of semiconductor elements in general (not only semiconductor IC devices) are described by the curve shown in the diagram below. Although semiconductor memory failures are similar to those of ordinary integrated circuits, the degree of integration (miniaturization), manufacturing complexity and other circuit element factors influence their incidence.

Semiconductor Element Failure Rate Curve



1) Surge Destruction

This is destruction of the input/output stage circuits by external surge currents or static electricity. The accompanying photograph shows a point of contact between aluminum and polysilicon that has been dissolved by a surge current. A hole has formed in the substrate silicon, leading to a short circuit. This kind of failure is traceable in about 30% of defective devices returned to the manufacturer. Despite miniaturization of semiconductor memory component elements (which means the elements themselves are less resistant), these failures usually occur during assembly and other handling operations.
At Oki, all devices are subjected to static electricity intensity tests (under simulated operational conditions) in the development stage to reduce this type of failure. In addition to checking endurance against surge currents, special protective circuits are incorporated in the input and output sections.



2) Oxide Film Insulation Destruction (Pin Holes)

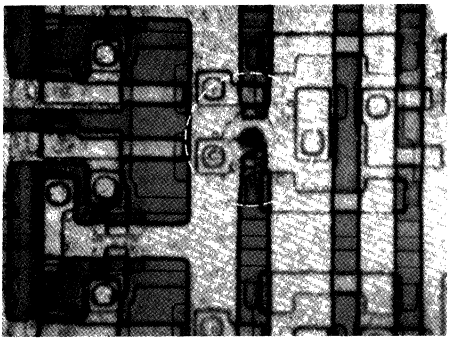
Unlike surge destruction, this kind of failure is caused by manufacturing defects. Locally weakened sections are ruptured when subjected to external electrical stress. Although this problem is accentuated by the miniaturization of circuit elements, it can be resolved by maintaining an ultra-clean manufacturing environment and through 100% burn-in screening.

3) Surface Deterioration due to Ionic Impurities

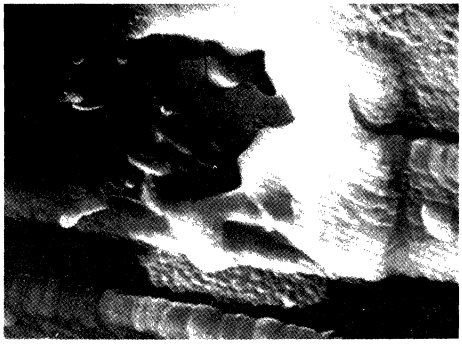
Under some temperature and electric field conditions, charged ionic impurities moving within the oxide film previously resulted in occasional deterioration of silicon surfaces. This problem has been eliminated by new surface stabilization techniques.

4) Photolithographic Defects

Integrated circuits are formed by repeated photographic etching processes. Dust and scratches on the mask (which corresponds to a photographic negative) can cause catastrophic defects. At present, component elements have been reduced in size to the order of 10 cm through miniaturization. However, the size of dust and scratches stays the same. At Oki, a high degree of automation, minimizing human intervention in the process, and unparalleled cleanliness, solves this problem.



Example of surge destruction



Photolithographic Defect

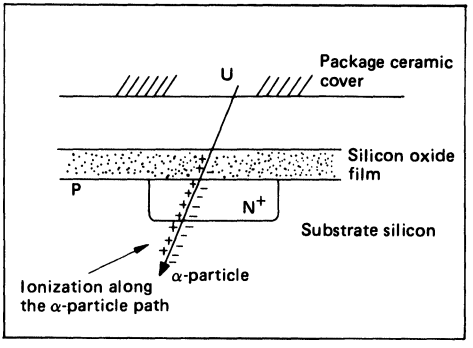
3

5) Aluminum Corrosion

Aluminum corrosion is due to electrolytic reactions caused by the presence of water and minute impurities. When aluminum dissolves, lines break. This problem is unique to the plastic capsules now used widely to reduce costs. Oki has carefully studied the possible cause and effect relationship between structure and manufacturing conditions on the one hand, and the generation of aluminum corrosion on the other. Refinements incorporated in Oki LSIs permit superior endurance to even the most severe high humidity conditions.

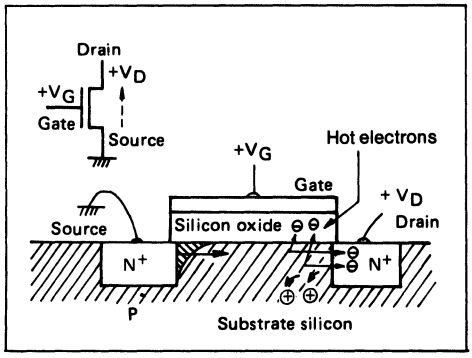
6) Alpha-Particle Soft Failure

This problem occurs when devices are highly miniaturized, such as in 1 megabit RAMs. The inversion of memory cell data by alpha-particle generated by radio-active elements like uranium and thorium (present in minute quantities, measured in ppb) in the ceramic package material causes defects. Since failure is only temporary and normal operation restored quickly, this is referred to as a "soft" failure. At Oki we have eliminated the problem by coating the chip surface of 1 megabit RAMs with a resin which effectively screens out these alpha-particles.



7) Degradation in Performance Characteristics Due to Hot Electrons

With increased miniaturization of circuit elements, internal electric field strength in the channels increases since the applied voltage remains the same at 5V. As a result, electrons flowing in the channels, as shown in the accompanying diagram, tend to enter into the oxide film near the drain, leading to degradation of performance. Although previous low-temperature operation tests have indicated an increase of this failure, we have confirmed by our low-temperature acceleration tests, including checks on test element groups, that no such problem exists in Oki LSIs.



Characteristic deterioration caused by hot electrons

With further progress in the miniaturization of circuit components, failures related to pin hole oxide film destruction and photolithography have increased. To eliminate these defects during manufacturing, Oki has been continually improving its production processes based on reliability tests and information gained from the field. And we subject all devices to high-temperature burn-in screening for 48 to 96 hours to ensure even greater reliability.

DATA SHEET

OLMS-50/60 SERIES

OKI semiconductor

MSM5052

CMOS 4 BIT SINGLE CHIP VERY LOW POWER MICROCONTROLLER WITH TEMPERATURE DETECTION CIRCUIT AND LCD DRIVER

GENERAL DESCRIPTION

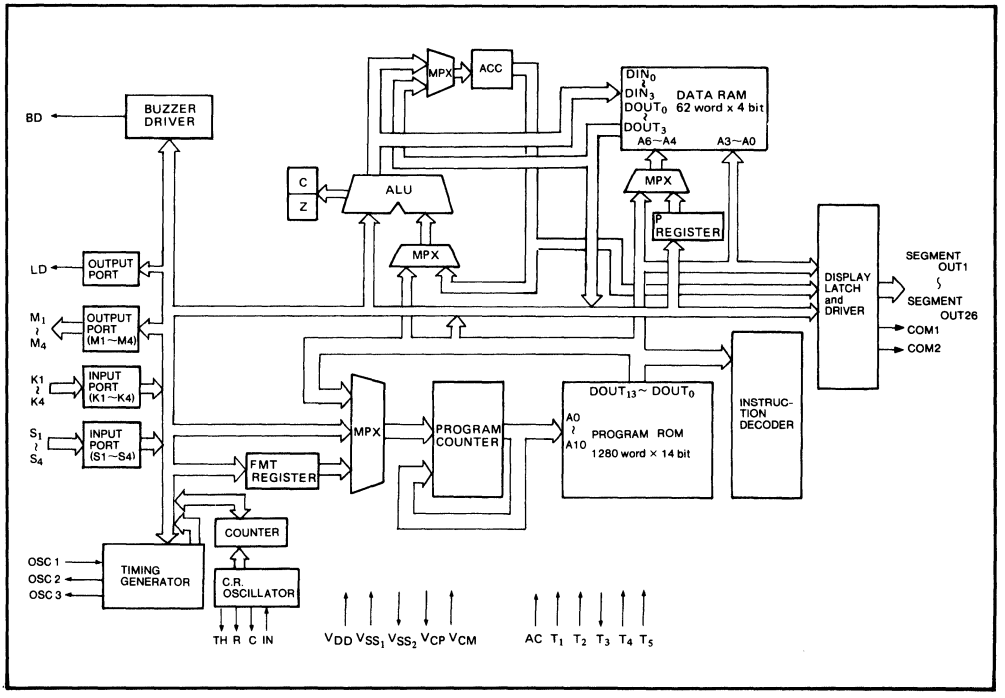
The OKI MSM5052 is a low-power and high-performance single-chip microcontroller employing complementary metal oxide semiconductor technology. Integrated onto a single chip are a 4 bit ALU, 18K bits of mask programmable ROM, 248 bits of data RAM, crystal oscillator, voltage doubler, timer, LCD driver, input port, output port and CR oscillator for temperature detection.

The MSM5052 is widely used in electronic products requiring low power operation, for example, thermometer and clinical thermometer with a time piece.

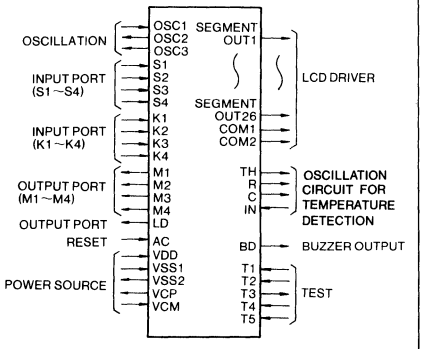
FEATURES

- Low Power Consumption 3 μA Typical
- 1280 × 14 Internal ROM
- 62 × 4 Internal RAM
- 4 × 2 Input Port
- 5 Output Port
- 4 × 4 Key Matrix Input (K1~K4, M1~M4)
- 26 LCD Driver
  (1/2 Duty, 1/2 Bias, 52 Segment)
- 42 Instructions
- 1.5 V Operating Voltage
- 32.768 kHz Crystal Oscillator
- 122.1 μs Instruction Cycle
- −20 to 75°C Operating Temperature
- 61 pad die
- Package:
  56(S) pin plastic QFP (QFP56-P-910-K)
  56(S) pin plastic QFP (QFP56-P-910-VK)
  80 pin plastic QFP (QFP80-P-1420-K)

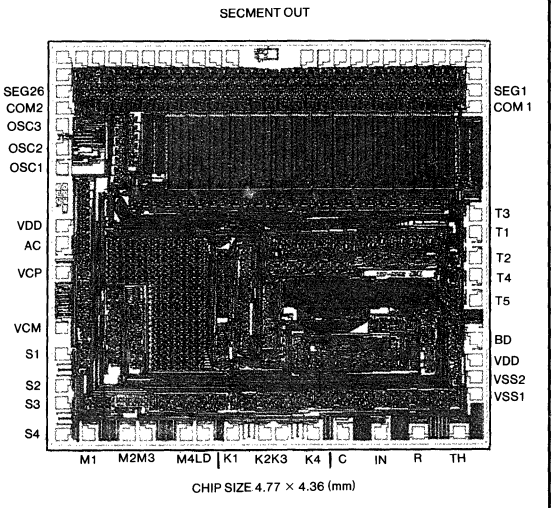
FUNCTIONAL BLOCK DIAGRAM



LOGIC SYMBOL



CHIP PAD LAYOUT



CHIP SIZE 4.77 × 4.36 (mm)

PIN DESCRIPTION

Designation	Function
V_{DD}	Circuit ground potential
V_{SS_1}	Power source (−1.5 V)
V_{SS_2}	Power source for LCD driver (−3.0 V) This terminal is connected to V_{DD} terminal through a 0.1 μF capacitor.
V_{CP}, V_{CM}	Booster capacitor connection terminals V_{CP} terminal is connected to V_{CM} terminal through a 0.1 μF capacitor.
OSC1, OSC3	Input and output terminals of oscillator inverter, 32.768 kHz crystal is connected to these terminals.
T1 ~ T5	Terminals to test internal logic, T1 ~ T3 and T5 are pulled down to V_{SS_1}. T4 is output. Test pins must be normally open.
AC	Terminal to clear internal logic pulled down to V_{SS_1}. After power is turned on, the MSM5052 must be reset by this terminal.
BD	Buzzer output
TH, R, C, IN	Terminal to CR oscillation circuit for temperature detection, fundamental resistor, thermistor, capacitor connection terminal.

FUNCTIONAL DESCRIPTION

A block diagram of the MSM5052 is given on page 91. Each block of logic will be briefly discussed. For more information, please refer to the MSM5052 user's manual.

Program ROM

The MSM5052 addresses up to 1.25 K words of internal mask programmable ROM. Each word consists of 14 bits and all instructions are one word. The instructions are routed to a programmed logic array which generates the signals necessary for control of logic.

Data RAM

Data is organized in 4-bit nibbles. Internal data RAM consists of 62 nibbles.

The RAM is addressed by page address and column address. Normally page address is specified with page register, but direct addressing is available in Page 0.

Column address is directly addressed by operand of various instructions.

ALU

The ALU performs 4-bit parallel operation on RAM and ACC contents, or RAM contents and an immediate digit. It sets or resets the flags (Z, C) depending on the condition.

Program Counter (PC)

The program counter is 11 bits wide and specifies the address of the program ROM.

The PC is incremented by one every execution of the instruction, and then specifies the next instruction to be executed. However, the contents of the PC are rewritten by the execution of the Jump or Branch instruction.

There is no boundary in the ROM, so a Jump or Branch instruction can be put anywhere in the ROM.

Input/Output Port

Input Port (S1 ~ S4)

The input port (S1 ~ S4) is a 4 bit parallel input port. Each pin of the port is pulled down to V_{SS1} by an internal resistor, and the status of the port is fetched by an input instruction.

Input Port (K1 ~ K4)

The input port (K1 ~ K4) is a 4 bit parallel input port. Each pin of the port is pulled down to V_{SS1} by an internal resistor, and the status of the port is fetched by an input instruction.

Output Port (M1 ~ M4)

The output port (M1 ~ M4) is a 4 bit parallel output port. This port consists of data latches and buffers, and the contents of the data latches are rewritten by an output instruction.

Output Port (LD)

The output port (LD) is single output port. This terminal is used for loading of M1 to M4 data.

Display Function

The MSM5052 is provided with a segment output terminal which can directly drive a 1/2 bias, 1/2 duty LCD and the common drive output terminal COM1 and COM2.

The segment drive circuit consists of the display data latch, multiplexer and driver. If the data is sent to the display data latch with a display instruction, the LCD drive waveform is output to the segment drive output terminal.

Time Base

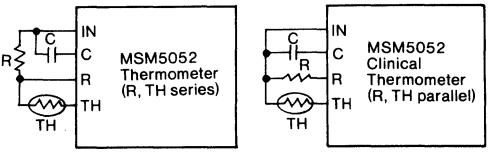
The time base for the CPU is provided by connecting a 32.768 kHz crystal to the OSC1 and OSC3 pins. One machine cycle is 122.1 μs.

A hardware divider up to 1 Hz is provided enabling programs to implement a clock function by counting signals between 16 and 1 Hz.

Temperature Detection Circuit

The temperature detection circuit is composed of an external thermistor, a fundamental resistor, a capacitor and a built-in CR oscillation circuit.

Two types of temperature measurement circuit, as shown below, are available.



ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage 1	$V_{DD} - V_{SS_1}$	Ta = 25°C	−0.3 to +2.0	V
Supply Voltage 2	$V_{DD} - V_{SS_2}$	Ta = 25°C	−0.3 to +4.0	V
Input Voltage	V_{IN_1}	Ta = 25°C	V_{SS_1} −0.3 to +0.3	V
Storage Temperature	Tstg		−55 to 125	°C

OPERATING CONDITIONS

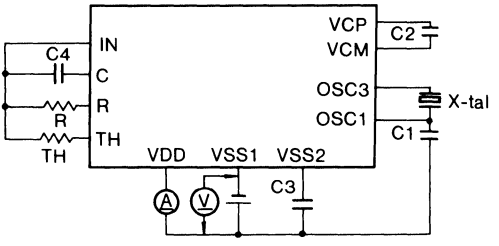
Parameter	Symbol	Limits	Unit
Operating Voltage	$V_{DD} - V_{SS_1}$	1.25 to 1.65	V
Operating Temperature	Topr	−20 to 75	°C

DC CHARACTERISTICS

(V_{DD} =0V, V_{SS_1} = −1.55V, V_{SS_2} = −3.0V, C_I = 30kΩ, Ta = 25°C)

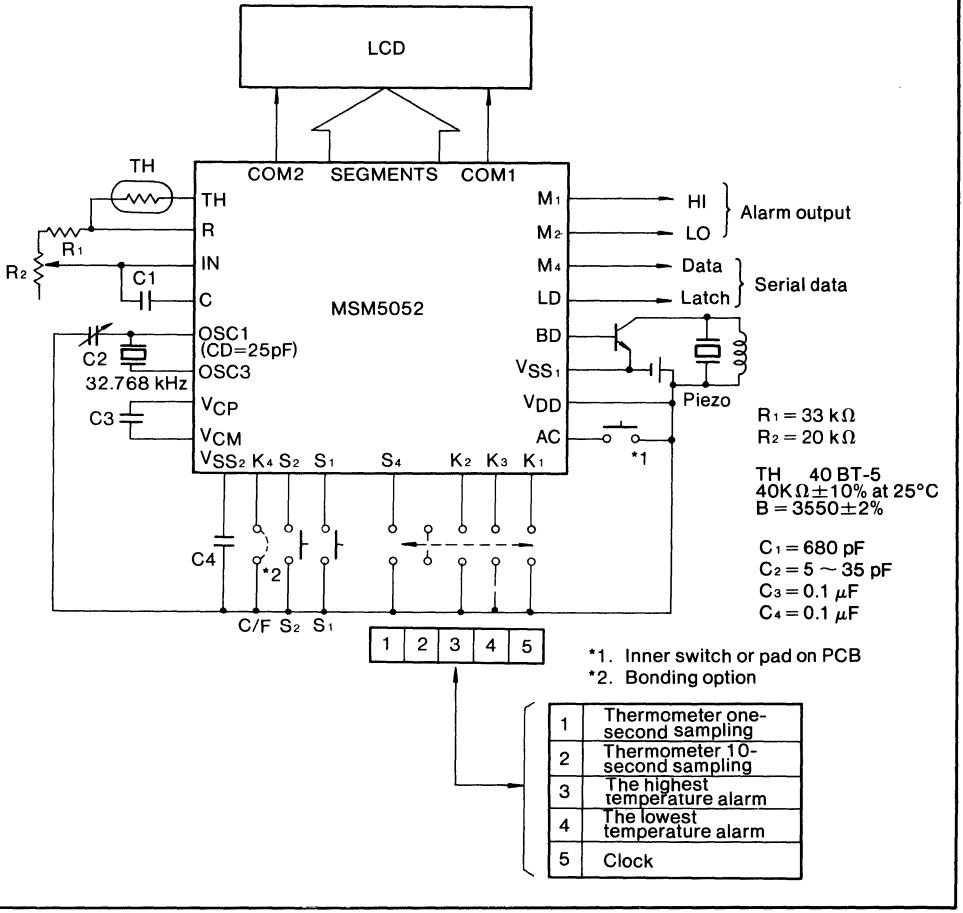
Parameter	Symbol	Condition	Limits			Unit
			Min.	Typ.	Max.	
Power supply current 1	I_{DD_1}	Temperature sampling off	–	3.0	–	μA
Power supply current 2	I_{DD_2}	Temperature sampling on	–	100	–	μA
Oscillation start voltage	$-V_{OSC}$	Within 10 seconds V_{SS_1} terminal	1.45	–	–	V
Output current 1 COM	I_{OH_1}	V_{OH_1} = −0.2V	−4	–	–	μA
	I_{OM_1}	$V_{OM_1} = V_{SS_1} \pm 0.2V$	4/−4	–	–	
	I_{OL_1}	V_{OL_1} = −2.8V	4	–	–	
Output current 2 SEGMENT	I_{OH_2}	V_{OH_2} = −0.2V	−0.4	–	–	μA
	I_{OL_2}	V_{OL_2} = −2.8V	0.4	–	–	
Output current 3 C. R. TH	I_{OH_3}	V_{OH_3} = −0.4 V	−400	–	–	μA
	I_{OL_3}	V_{OL_3} = −1.15V	400	–	–	
Output current 4 M1~M4 LD	I_{OH_4}	V_{OH_4} = −0.4V	−100	–	–	μA
	I_{OL_4}	V_{OL_4} = −1.15V	10	–	–	
Output current 5 BD	I_{OH_5}	V_{OH_5} = −0.4V	−50	–	−500	μA
	I_{OL_5}	V_{OL_5} = −1.15V	4	–	–	
Input current S1~S4 K1~K4	I_{IH_1}	V_{IH_1} = 0V	1	10	100	μA
	I_{IL_1}	V_{IL_1} = −1.55V	–	–	−0.2	
Oscillator built-in capacitor	CD		–	25	–	pF

MEASURING CIRCUIT



C1 : 20 pF
C2, C3 : 0.1 μF
C4 : 3000 pF
X-tal : 32.768 kHz
R, TH : 10 KΩ

TYPICAL APPLICATION



DESCRIPTION OF INSTRUCTIONS

	Mnemonic	Instruction Code														Operation
		13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Arithmetic operation	ADD ACC, AP	0	0	0	0	0	P	0	1	0	0	A				AP ← (AP) + (ACC)
	ADD #D, AP	0	1	1	0	0	P	D				A				AP ← (AP) − D
	ADC AP	0	0	0	0	0	P	0	1	0	1	A				AP ← Decimal adjust {(AP) + (ACC) + (C)}
	SUB ACC, AP	0	0	0	0	1	P	0	1	0	0	A				AP ← (AP) − (ACC)
	SUB #D, AP	0	1	1	0	1	P	D				A				AP ← (AP) − D
	SBC AP	0	0	0	0	1	P	0	1	0	1	A				AP ← Decimal adjust {(AP) − (ACC) − (C)}
	CMP ACC, AP	0	0	0	0	1	P	1	1	1	0	A				(AP) − (ACC)
	CMP #D, AP	0	1	0	1	1	P	D				A				(AP) − D
	INC AP	0	1	1	0	0	P	0	0	0	1	A				AP ← (AP) + 1
	DEC AP	0	1	1	0	1	P	0	0	0	1	A				A ← (AP) − 1
	XOR ACC, AP	0	0	0	0	0	P	0	1	1	1	A				AP ← (AP) ∀ (ACC)
	XOR #D, AP	0	1	1	1	1	P	D				A				AP ← (AP) ∀ D
Bit operation	BIT ACC, AP	0	0	0	0	0	P	1	1	1	0	A				(AP) V $\overline{(ACC)}$
	BIT #D, AP	0	1	0	1	0	P	D				A				(AP) V $\overline{D}$
	BIS ACC, AP	0	0	0	0	0	P	0	1	1	0	A				AP ← (AP) V (ACC)
	BIS #D, AP	0	1	0	0	0	P	D				A				AP ← (AP) V D
	BIC ACC, AP	0	0	0	0	1	P	0	1	1	0	A				AP ← (AP) Λ $\overline{(ACC)}$
	BIC #D, AP	0	1	0	0	1	P	D				A				AP ← (AP) Λ $\overline{D}$
Shift	ASR AP	0	0	0	0	0	P	0	0	1	1	A				⌐→(C) 0 → (AP)⌐
	ASL AP	0	0	0	0	1	P	0	0	1	1	A				(C) ← (AP) ← 0
Flag operation	CLZ	0	0	0	0	0	0	1	0	1	0	0	0	0	0	Z ← 0
	CLC	0	0	0	0	0	0	1	0	0	1	0	0	0	0	C ← 0
	CLA	0	0	0	0	0	0	1	0	1	1	0	0	0	0	Z ← 0, C ← 0
	SEZ	0	0	0	0	1	0	1	0	1	0	0	0	0	0	Z ← 1
	SEC	0	0	0	0	1	0	1	0	0	1	0	0	0	0	C ← 1
	SEA	0	0	0	0	1	0	1	0	1	1	0	0	0	0	Z ← 1, C ← 1
Data transfer	MOV ACC, AP	1	1	1	1	0	P	0	0	0	0	A				AP ← (ACC)
	MOV ACC, AX	1	1	1	1	0	0	0	X			A				AX ← (ACC)
	MOV #D, AP	0	1	1	1	0	P	D				A				AP ← D
	MOV AP, ACC	1	1	1	1	1	P	0	0	0	0	A				ACC ← (AP)
	MOV AX, ACC	1	1	1	1	1	0	0	X			A				ACC ← (AX)
	CHG AP	1	1	1	0	0	P	0	0	0	0	A				(ACC) ⟷ (AP)
	CHG AX	1	1	1	0	0	0	0	X			A				(ACC) ⟷ (AX)

DESCRIPTION OF INSTRUCTIONS (CONT.)

	Mnemonic	Instruction Code														Operation
		13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Jump	JMP adrs	1	0	0	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	PC ← adrs
	JMP @AP	0	0	0	0	0	P	1	1	0	1	A				PC ← (PC) + (AP) + 1
	JMPIO @AP	0	0	0	0	1	P	1	1	0	1	A				PC ← (PC) + [(AP)A7H] + +1
	BEP +n BZE +n	0	0	0	1	1	0	0	1	0	n_4	n_3	n_2	n_1	n_0	PC ← (PC) + n + 1, if Z = 1
	BNE +n BNZ +n	0	0	0	1	1	0	1	1	0	n_4	n_3	n_2	n_1	n_0	PC ← (PC) + n + 1, if Z = 0
	BCS +n BLT +n	0	0	0	1	1	0	0	0	1	n_4	n_3	n_2	n_1	n_0	PC ← (PC) + n + 1, if C = 1
	BCC +n BGE +n	0	0	0	1	1	0	1	0	1	n_4	n_3	n_2	n_1	n_0	PC ← (PC) + n + 1, if C = 0
	BGT +n	0	0	0	1	1	0	1	1	1	n_4	n_3	n_2	n_1	n_0	PC ← (PC) + n + 1, if Z = 0 and C = 0
	BLE +n	0	0	0	1	1	0	0	1	1	n_4	n_3	n_2	n_1	n_0	PC ← (PC) + n + 1, if Z = 1 or C = 1
Input/ Output	INP Port, AP	1	1	0	1	0	P	Port				A				AP ← (Port)
	OUT AP, Port	1	1	0	1	1	P	Port				A				Port ← (AP)
	OUT #D, Port	0	0	0	1	Q	0	Port				D				Port ← D
Display	DSP digit, AP	0	0	1	0	0	P	digit				A				digit ← (AP), (ACC)
	DSPF digit, AP	0	0	1	1	0	P	digit				A				digit ← (AP) via table
CPU control	HALT	0	0	0	1	0	0	0	0	0	0	0	0	0	0	Halt CPU
	NOP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	No Operation

MSM5054

CMOS 4 BIT SINGLE CHIP VERY LOW POWER MICROCONTROLLER WITH LCD DRIVER

GENERAL DESCRIPTION

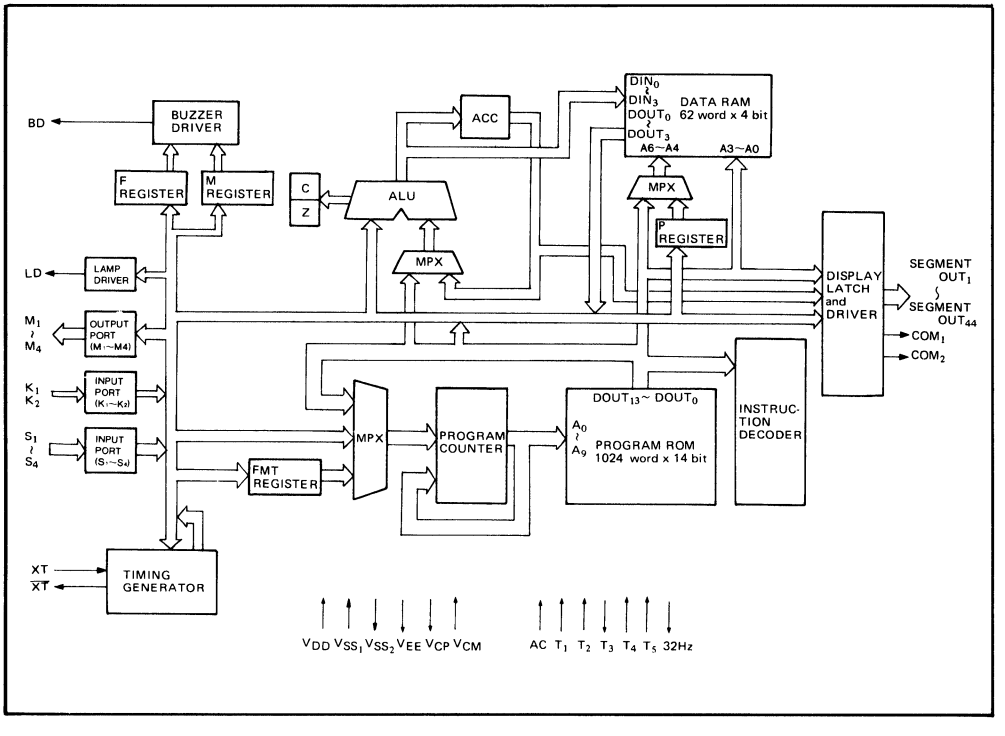
The OKI MSM5054 is a low-power, high-performance single-chip microcontroller employing complementary metal oxide semiconductor technology. Integrated onto a single chip are 4-bit of ALU, 14K bits of mask programmable ROM, 248 bits of data RAM, crystal oscillator, voltage doubler, timer, LCD driver, input port and output port.

The MSM5054 is widely used in electronic products requiring low power operation, for example, Clocks, Timers and Games.

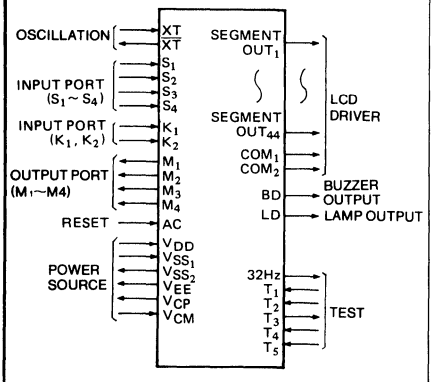
FEATURES

- Low Power Consumption 3 μA Typical
- 1024 × 14 Internal ROM
- 62 × 4 Internal RAM
- 6 Input Port
- 4 Output Port
- 4 × 4 Key Matrix Input (S_1~S_4, M_1~M4)
- 44 LCD Driver
  (1/2 Duty, 1/2 Bias, 88 Segment)
- 40 Instructions
- 1.5 V or 3 V Operating Voltage
  (Masking Option)
- 32.768 kHz Crystal Oscillator
- 122.1 μs Instruction Cycle
- −20 to 75°C Operating Temperature
- 74 pad die
- Package:
  56(S) pin plastic QFP (QFP56-P-910-K)
  80 pin plastic QFP (QFP80-P-1420-K)

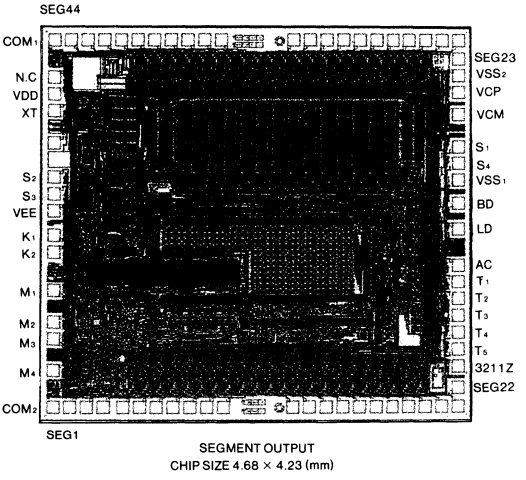
FUNCTIONAL BLOCK DIAGRAM



LOGIC SYMBOL



CHIP PAD LAYOUT



SEGMENT OUTPUT
CHIP SIZE 4.68 × 4.23 (mm)

PIN DESCRIPTION

Designation	Function
V_{DD}	Circuit ground potential
V_{SS_1}	Power source (−1.5 V)
V_{SS_2}	Power source for LCD driver (−3.0 V) This terminal is connected to V_{DD} terminal through a 0.1 μF capacitor.
V_{EE}	Power source for internal logic (−1.5 to −3.0 V) This terminal is connected to V_{DD} terminal through a 0.1 μF capacitor.
V_{CP}, V_{CM}	Booster capacitor connection terminals V_{CP} terminal is connected to V_{CM} terminal through a 0.1 μF capacitor.
XT, $\overline{XT}$	Input and output terminals of oscillator inverter, 32.768 kHz crystal is connected to these terminals.
$T_1 \sim T_5$	Terminals to test internal logic, $T_1 \sim T_3$ and T_5 are pulled down to V_{SS_1}. T_4 is output. Test pins must be normally open.
AC	Terminal to clear internal logic pulled down to V_{SS_1} After power is turned on, the MSM5054 must be reset by this terminal.
BD	Buzzer output
LD	Lamp output

FUNCTIONAL DESCRIPTION

A block diagram of the MSM5054 is given on page 97. Each block of logic will be briefly discussed. For more information, please refer to the MSM5054 user's manual.

Program ROM

The MSM5054 addresses up to 1 K word of internal mask programmable ROM. Each word consists of 14 bits, and all instructions are one word. The instructions are routed to a programmed logic array which generates the signals necessary for control of logic.

Data RAM

Data is organized in 4-bit nibbles. Internal data RAM consists of 62 nibbles.

The RAM is addressed by page address and column address. Normally page address is specified by the page register, but direct addressing is available in Page 0.

Column address is directly addressed by the operand of various instructions.

ALU

The ALU performs 4-bit parallel operation of RAM and ACC contents, or RAM contents and an immediate digit. It sets or resets the flags (Z, C) depending on the condition.

Program Counter (PC)

The program counter is 10 bits wide and specifies the address of the program ROM.

The PC is incremented by one at every execution of the instruction, and then specifies the next instruction to be executed. However, the contents of the PC are rewritten by the execution of the Jump or Branch instruction.

There is no boundary in the ROM, so the Jump or Branch instruction can be put anywhere in the ROM.

Input/Output Port

Input Port (S1 ~ S4)

The input port (S1 ~ S4) is a 4-bit parallel input port. Each pin of the port is pulled down to V_{SS1} by an internal resistor, and the status of the port is fetched by the SWITCH instruction.

Input Port (K1 ~ K4)

The input port (K1 ~ K2) is a 2-bit parallel input port. Each pin of the port is pulled down to V_{SS1} by an internal resistor, and the status of the port is fetched by the KSWITCH instruction.

Output Port (M1 ~ M4)

The output port (M1 ~ M4) is a 4-bit parallel output port. This port consists of data latches and buffers, and the contents of the data latches are rewritten by a matrix instruction.

Display Function

The MSM5054 is provided with a segment output terminal which can directly drive a 1/2 bias, 1/2 duty LCD, and the common drive output terminal COM1 and COM2. The segment drive circuit consists of the display data latch, multiplexer and driver. If the data is sent to the display data latch with a display instruction, the LCD drive waveform is output to the segment drive output terminal.

Time Base

The time base of the CPU is provided by connecting a 32.768 kHz crystal to the XT and $\overline{XT}$ pins. One machine cycle is 122.1 μs.

A hardware divider of up to 1 Hz is provided, enabling programs to implement a clock function by counting signals between 32 and 1 Hz.

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage 1	$V_{DD} - V_{SS_1}$	Ta = 25°C	−0.3 to +2.0	V
Supply Voltage 2	$V_{DD} - V_{SS_2}$	Ta = 25°C	−0.3 to +4.0	V
Supply Voltage 3	$V_{DD} - V_{EE}$	Ta = 25°C	−0.3 to +4.0	V
Input Voltage	V_{IN_1}	Ta = 25°C	V_{SS_1} −0.3 to +0.3	V
Storage Temperature	Tstg		−55 to 125	°C

OPERATING CONDITIONS

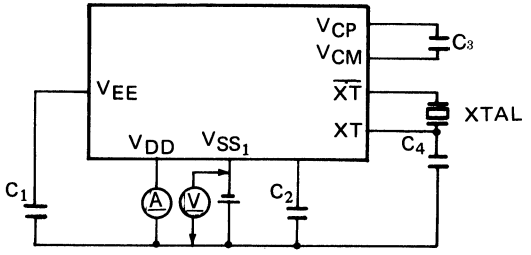
Parameter	Symbol	Limits	Unit
Operating Voltage	$V_{DD} - V_{SS_1}$	1.25 to 1.65	V
Operating Temperature	Topr	−20 to 75	°C

DC CHARACTERISTICS

(V_{DD} =0V, V_{SS_1}, V_{EE} = −1.55V, V_{SS_2} = −3.0V, C_I = 30kΩ, Ta = 25°C)

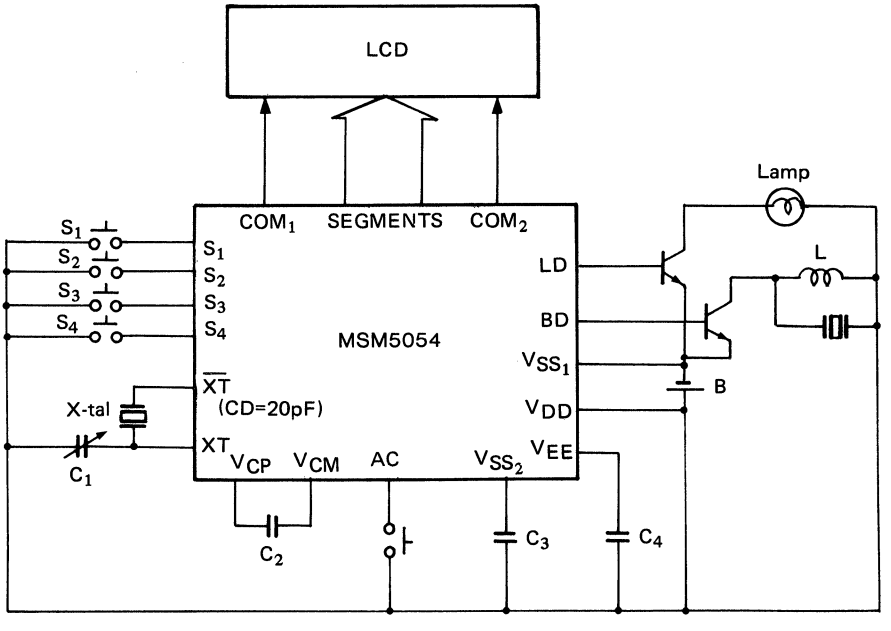
Parameter	Symbol	Condition		Limits			Unit
				Min.	Typ.	Max.	
Power supply current	I_{DD}			−	3.0	−	μA
Oscillation start voltage	$-V_{OSC}$	Within 5 seconds V_{SS_1} terminal		1.45	−	−	V
Output current 1 COM	I_{OH_1}	$V_{OH_1} = -0.2V$		−4	−	−	μA
	I_{OM_1}	$V_{OM_1} = V_{SS_1} \pm 0.2V$		4/−4	−	−	
	I_{OL_1}	$V_{OL_1} = -2.8V$		4	−	−	
Output current 2 SEGMENT	I_{OH_2}	$V_{OH_2} = -0.2V$		−0.4	−	−	μA
	I_{OL_2}	$V_{OL_2} = -2.8V$		0.4	−	−	
Output current 3 BD	I_{OH_3}	$V_{OH_3} = -0.4$ V	$V_{SS_1} = -1.25V$ $V_{EE} = -1.25V$ $V_{SS_2} = -2.4V$	−50	−	−500	μA
	I_{OL_3}	$V_{OL_3} = -0.8V$		2.5	−	7.5	
Output current 4 LD	I_{OH_4}	$V_{OH_4} = -0.55V$	$V_{SS_1} = -1.25V$ $V_{EE} = -2.4V$ $V_{SS_2} = -2.4V$	−21	−	−83	μA
	I_{OL_4}	$V_{OL_4} = -1.15V$		1	−	−	
Output current 5 $M_1 \sim M_4$	I_{OH_5}	$V_{OH_5} = -0.5V$		−100	−	−	μA
	I_{OL_5}	$V_{OL_5} = -1.0V$		1.5	−	7.5	
Input current 1 $S_1 \sim S_4$	I_{IH_1}	$V_{IH_1} = 0V$		1	10	50	μA
	I_{IL_1}	$V_{IL_1} = -1.55V$		−	−	−0.2	
Input current 2 K_1, K_2	I_{IH_2}	$V_{IH_2} = 0V$		2.5	6	12	μA
	I_{IL_2}	$V_{IL_2} = -1.55V$		−	−	−0.2	
Oscillator built-in capacitor	CD			−	20	−	pF

MEASURING CIRCUIT



C_1, C_2, C_3 : 0.1μF
C_4 : 30pF
XTAL : 32.768kHz

TYPICAL APPLICATION



C_1 : 5 ~ 35pF
C_2, C_3, C_4 : 0.1μF
B : 1.5V
L : 20mH

DESCRIPTION OF INSTRUCTIONS

	Mnemonic	Instruction Code																Operation
		13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Arithmetic operation	ADD ACC, AP	0	0	0	0	0	P	0	1	0	0	A				AP ← (AP) + (ACC)		
	ADD #D, AP	0	1	1	0	0	P	D				A				AP ← (AP) + D		
	SUB ACC, AP	0	0	0	0	1	P	0	1	0	0	A				AP ← (AP) − (ACC)		
	SUB #D, AP	0	1	1	0	1	P	D				A				AP ← (AP) − D		
	ADJUST N, AP	1	1	0	0	0	P	$\overline{N}+1$				A				AP ← N adjust {(AP)}		
	CMP ACC, AP	0	0	0	0	1	P	1	1	1	0	A				(AP) − (ACC)		
	CMP #D, AP	0	1	0	1	1	P	D				A				(AP) − D		
	INC AP	0	1	1	0	0	P	0	0	0	1	A				AP ← (AP) + 1		
	DEC AP	0	1	1	0	1	P	0	0	0	1	A				A ← (AP) − 1		
	XOR ACC, AP	0	0	0	0	0	P	0	1	1	1	A				AP ← (AP) ∀ (ACC)		
	XOR #D, AP	0	1	1	1	1	P	D				A				AP ← (AP) ∀ D		
Bit operation	BIT ACC, AP	0	0	0	0	0	P	1	1	1	0	A				(AP) V ($\overline{ACC}$)		
	BIT #D, AP	0	1	0	1	0	P	D				A				(AP) V $\overline{D}$		
	BIS ACC, AP	0	0	0	0	0	P	0	1	1	0	A				AP V (ACC)		
	BIS #D, AP	0	1	0	0	0	P	D				A				(AP) V D		
	BIC ACC, AP	0	0	0	0	1	P	0	1	1	0	A				AP Λ ($\overline{ACC}$)		
	BIC #D, AP	0	1	0	0	1	P	D				A				AP Λ $\overline{D}$		
Shift	ASR AP	0	0	0	0	0	P	0	0	1	1	A				↳(C) 0 → (AP)┘		
	ASL AP	0	0	0	0	1	P	0	0	1	1	A				(C) ← (AP) ← 0		
Flag operation	CLZ	0	0	0	0	0	0	1	0	1	0	0	0	0	0	Z ← 0		
	CLC	0	0	0	0	0	0	1	0	0	1	0	0	0	0	C ← 0		
	CLA	0	0	0	0	0	0	1	0	1	1	0	0	0	0	Z ← 0, C ← 0		
	SEZ	0	0	0	0	1	0	1	0	1	0	0	0	0	0	Z ← 1		
	SEC	0	0	0	0	1	0	1	0	0	1	0	0	0	0	C ← 1		
	SEA	0	0	0	0	1	0	1	0	1	1	0	0	0	0	Z ← 1, C ← 1		
Data transfer	MOV ACC, AP	1	1	1	1	0	P	0	0	0	0	A				AP ← (ACC)		
	MOV ACC, AX	1	1	1	1	0	0	0		X		A				AX ← (ACC)		
	MOV #D, AP	0	1	1	1	0	P	D				A				AP ← D		
	MOV AP, ACC	1	1	1	1	1	P	0	0	0	0	A				ACC ← (AP)		
	MOV AX, ACC	1	1	1	1	1	0	0		X		A				ACC ← (AX)		

DESCRIPTION OF INSTRUCTIONS (CONT.)

	Mnemonic	Instruction Code 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Operation
Jump	JMP Adrs	1 0 0 0 a_9 a_8 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	PC ← Adrs
	JMP @AP	0 0 0 0 0 P 1 1 0 1 A	PC ← (PC) + (AP) + 1
	JMPIO @AP	0 0 0 0 1 P 1 1 0 1 A	PC ← (PC) + {(AP)∧7H} +1
	BEQ +n BZE +n	0 0 0 1 1 0 0 1 0 n_4 n_3 n_2 n_1 n_0	PC ← (PC)+n+1, if Z=1
	BNE +n BNZ +n	0 0 0 1 1 0 1 1 0 n_4 n_3 n_2 n_1 n_0	PC ← (PC)+n+1, if Z=0
	BCS +n	0 0 0 1 1 0 0 0 0 n_4 n_3 n_2 n_1 n_0	PC ← (PC)+n+1, if C=1
	BCC +n	0 0 0 1 1 0 1 0 0 n_4 n_3 n_2 n_1 n_0	PC ← (PC)+n+1, if C=0
Input/Output	SWITCH AP	1 1 0 1 0 P 0 0 0 1 A	AP ← INPUT PORT (S_1 ~ S_4)
	KSWITCH AP	1 1 0 1 0 P 0 0 1 0 A	AP ← INPUT PORT (K_1 ~ K_2)
	MATRIX AP	1 1 0 1 1 P 0 0 1 0 A	OUTPUT PORT (M_1 ~ M_4) ←(AP)
	MATRIX Mn	0 0 0 1 0 0 0 0 1 0 M_4 M_3 M_2 M_1	OUTPUT PORT (M_1~M_4) ← Mn (n=1, 2, 3, 4)
	BUZZER freq., sound	0 0 0 1 0 0 1 1 0 0 b_3 b_2 b_1 b_0	Freq ← freq, Mreg ← sound Buzzer start
	BSO	0 0 0 1 0 0 1 1 0 0 0 0 0 0	Buzzer stop
	LAMP ON/OFF	0 0 0 1 0 0 0 0 0 1 0 0 b_1 b_0	LD ON/OFF
Display	DSP digit, AP	0 0 1 0 0 P digit A	Digit ← (AP), (ACC)
	FORMAT AP	1 1 0 1 1 P 0 0 1 1 A	FMT reg. ← (AP)
	FORMAT N	0 0 0 1 0 0 0 0 1 1 N	FMT reg. ← N
	DSPF digit, AP	0 0 1 1 0 P digit A	Digit ← (AP) via table
CPU Control & Others	HALT	0 0 0 1 0 0 0 0 0 0 0 0 0 0	Halt
	INTENAB 32/16	0 0 0 1 0 0 1 0 1 1 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0	Enable timer
	INTDSAB 32/16	0 0 0 1 0 0 1 0 1 1 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1	Disable timer
	INTMODE AP	1 1 0 1 0 P 0 1 0 0 A	AP ← Interrupt mode
	PAGE A0	1 1 0 1 1 0 0 1 0 1 A	Preg ← (A0)
	PAGE N	0 0 0 1 0 0 0 1 0 1 N	Preg ← N
	RATE AP	1 1 0 1 0 P 1 0 0 1 A	AP ← DIVIDER (8 Hz~1 Hz)
	RSTRATE	0 0 0 1 0 0 1 0 0 0 1 0 0 0	DIVIDER (8 Hz~1 Hz) ← 0
	BACKUP ON/OFF	0 0 0 1 0 0 0 0 0 1 b_3 b_2 0 0	Backup ON/OFF
	NOP	0 0 0 0 0 0 0 0 0 0 0 0 0 0	No operation

OKI semiconductor

MSM5055

CMOS 4 BIT SINGLE CHIP VERY LOW POWER MICROCONTROLLER WITH LCD DRIVER

GENERAL DESCRIPTION

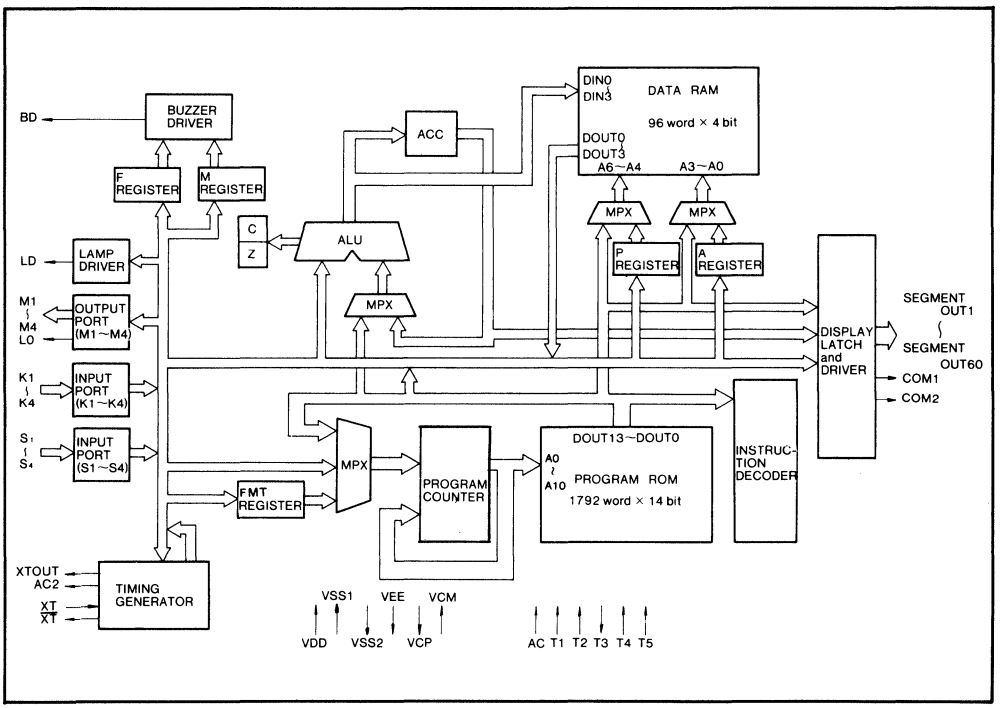
The OKI MSM5055 is a low-power, high-performance single-chip microcontroller employing complementary metal oxide semiconductor technology. Integrated onto a single chip are 4 bits of ALU, 25K bits of mask programmable ROM, 384 bits of data RAM, crystal oscillator, voltage doubler, timer, LCD driver, input port, output port and interface circuit for voice LSI (MSM6212).

The MSM5055 is widely used in electronic products requiring low power operation, for example, multi-functioned watches, voice synthesizer watches and games.

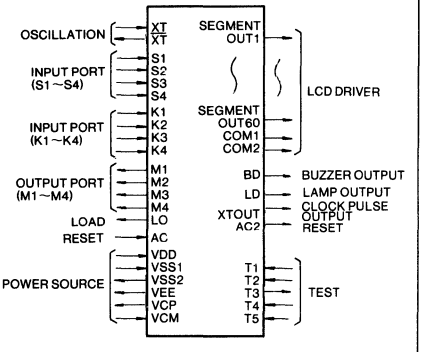
FEATURES

- Low Power Consumption 3 μA Typical
- 1792 × 14 Internal ROM
- 96 × 4 Internal RAM
- 4 × 2 Input Port
- 4 × 1 Output Port
- 4 × 4 Key Matrix Input (K1 ~ K4, M1 ~ M4)
- 60 LCD Driver
  (1/2 Duty, 1/2 Bias, 120 Segment)
- 42 Instructions
- 1.5 V or 3 V Operating Voltage
  (Masking Option)
- 32.768 kHz Crystal Oscillator
- 122.1 μS Instruction Cycle
- −20 to 75°C Operating Temperature
- 94 pad die
- Package:
  80 pin plastic QFP (QFP80-P-1420-K)

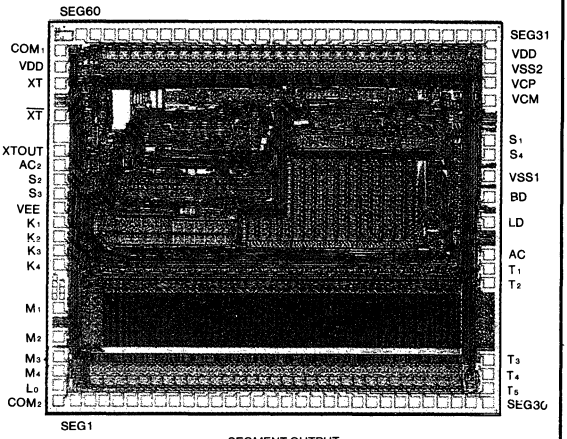
FUNCTIONAL BLOCK DIAGRAM



LOGIC SYMBOL



CHIP PAD LAYOUT



SEGMENT OUTPUT

CHIP SIZE 5.49 × 4.71 (mm)

PIN DESCRIPTION

Designation	Function
V_{DD}	Circuit ground potential
V_{SS1}	Power source (−1.5 V)
V_{SS2}	Power source for LCD driver (−3.0 V) This terminal is connected to V_{DD} terminal through a 0.1 μF capacitor.
V_{EE}	Power source for internal logic (−1.5 to −3.0 V) This terminal is connected to V_{DD} terminal through a 0.1 μF capacitor.
V_{CP}, V_{CM}	Booster capacitor connection terminals V_{CP} terminal is connected to V_{CM} terminal through a 0.1 μF capacitor.
XT, $\overline{XT}$	Input and output terminals of oscillator inverter, 32.768 kHz crystal is connected to these terminals.
T1 ~ T5	Terminals to test internal logic, T1 ~ T3 and T5 are pulled down to V_{SS1}. T4 is output. Test pins must be normally open.
AC	Terminal to clear internal logic pulled down to V_{SS1}. After power is turned on, the MSM5055 must be reset by this terminal.
BD	Buzzer output
LD	Lamp output
LO	Load data terminal of M_1 to M_4
AC2	Reset terminal for external circuit
XT OUT	Clock output for external circuit

FUNCTIONAL DESCRIPTION

A block diagram of the MSM5055 is given on page 104. Each block of logic will be briefly discussed. For more information, please refer to the MSM5055 user's manual.

Program ROM

The MSM5055 addresses up to 1 K word of internal mask programmable ROM. Each word consists of 14 bits, and all instructions are one word. The instructions are routed to a programmed logic array which generates the signals necessary for control of logic.

Data RAM

Data is organized in 4-bit nibbles. Internal data RAM consists of 62 nibbles.

The RAM is addressed by page address and column address. Normally page address is specified by the page register, but direct addressing is available in Page 0.

Column address is directly addressed by the operand of various instructions.

ALU

The ALU performs 4-bit parallel operation of RAM and ACC contents, or RAM contents and an immediate digit. It sets or resets the flags (Z, C) depending on the condition.

Program Counter (PC)

The program counter is 10 bits wide and specifies the address of the program ROM.

The PC is incremented by one at every execution of the instruction, and then specifies the next instruction to be executed. However, the contents of the PC are rewritten by the execution of the Jump or Branch instruction.

There is no boundary in the ROM, so the Jump or Branch instruction can be put anywhere in the ROM.

Input/Output Port

Input Port (S1 ~ S4)

The input port (S1 ~ S4) is a 4-bit parallel input port. Each pin of the port is pulled down to V_{SS1} by an internal resistor, and the status of the port is fetched by the SWITCH instruction.

Input Port (K1 ~ K4)

The input port (K1 ~ K2) is a 2-bit parallel input port. Each pin of the port is pulled down to V_{SS1} by an internal resistor, and the status of the port is fetched by the KSWITCH instruction.

Output Port (M1 ~ M4)

The output port (M1 ~ M4) is a 4-bit parallel output port. This port consists of data latches and buffers, and the contents of the data latches are rewritten by a matrix instruction.

Display Function

The MSM5055 is provided with a segment output terminal which can directly drive a 1/2 bias, 1/2 duty LCD, and the common drive output terminal COM1 and COM2. The segment drive circuit consists of the display data latch, multiplexer and driver. If the data is sent to the display data latch with a display instruction, the LCD drive waveform is output to the segment drive output terminal.

Time Base

The time base of the CPU is provided by connecting a 32.768 kHz crystal to the XT and $\overline{XT}$ pins. One machine cycle is 122.1 μs.

A hardware divider of up to 1 Hz is provided, enabling programs to implement a clock function by counting signals between 32 and 1 Hz.

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage 1	$V_{DD} - V_{SS1}$	Ta = 25°C	−0.3 to +2.0	V
Supply Voltage 2	$V_{DD} - V_{SS2}$	Ta = 25°C	−0.3 to +4.0	V
Supply Voltage 3	$V_{DD} - V_{EE}$	Ta = 25°C	−0.3 to +4.0	V
Input Voltage	V_{IN1}	Ta = 25°C	V_{SS1} −0.3 to +0.3	V
Storage Temperature	Tstg		−55 to 125	°C

OPERATING CONDITIONS

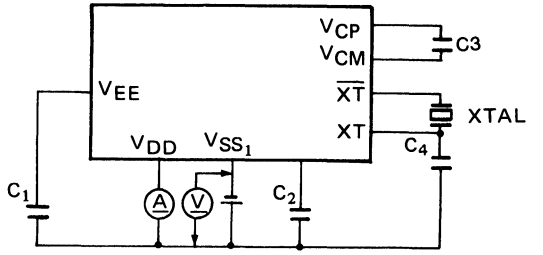
Parameter	Symbol	Limits	Unit
Operating Voltage	$V_{DD} - V_{SS1}$	1.25 to 1.65	V
Operating Temperature	Topr	−20 to 75	°C

DC CHARACTERISTICS

(VDD =0V, V_{SS1}, V_{EE} = −1.55V, V_{SS2} = −3.0V, C_I = 30kΩ, Ta = 25°C)

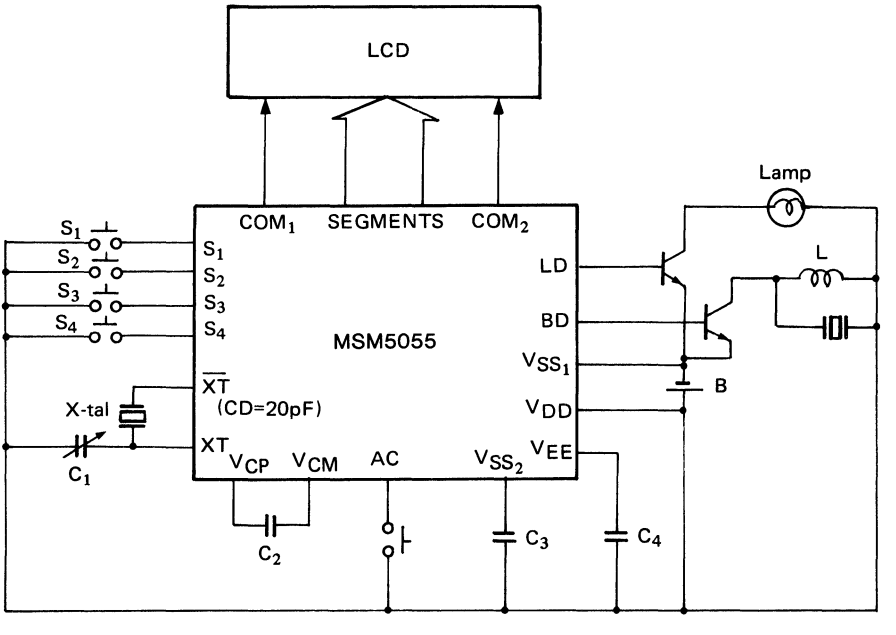
Parameter	Symbol	Conditions		Limits			Unit
				Min.	Typ.	Max.	
Power supply current	I_{DD}			−	3.0	−	μA
Oscillation start voltage	$-V_{OSC}$	Within 5 seconds V_{SS1} terminal		1.45	−	−	V
Output current 1 COM	I_{OH1}	V_{OH1} = −0.2V		−4	−	−	μA
	I_{OM1}	$V_{OM1} = V_{SS1} \pm 0.2V$		4/−4	−	−	
	I_{OL1}	V_{OL1} = −2.8V		4	−	−	
Output current 2 SEGMENT	I_{OH2}	V_{OH2} = −0.2V		−0.4	−	−	μA
	I_{OL2}	V_{OL2} = −2.8V		0.4	−	−	
Output current 3 AC2 LOAD, XTOUT	I_{OH3}	V_{OH3} = −0.5V		−10	−	−	μA
	I_{OL3}	V_{OL3} = −1.05V		10	−	−	
Output current 4 M1~M4	I_{OH4}	V_{OH4} = −0.5V		−100	−	−	μA
	I_{OL4}	V_{OL4} = −1.0V		1.5	−	12.7	
Output current 5 LD	I_{OH5}	V_{OH5} = −0.55V	V_{SS1} = −1.25V V_{EE} = −2.4V V_{SS2} = −2.4V	−21.6	−	−83	μA
	I_{OL5}	V_{OL5} = −1.15V		1	−	−	
Output current 5 BD	I_{OH6}	V_{OH6} = −0.4V	V_{SS1} = VEE =−1.25V V_{SS2}=−2.4V	−50	−	−	μA
	I_{OH6}	V_{OL6} = −0.8V		−	5	−	
Input current 1 S_1 ~ S4	I_{IH1}	V_{IH1} = 0V		1	10	50	μA
	I_{IL1}	V_{IL1} = −1.55V		−	−	−0.2	
Input current 2 K_1~K_4	I_{IH2}	V_{IH2} = 0V		2.5	6	12	μA
	I_{IL2}	V_{IL2} = −1.55V		−	−	−0.2	
Oscillator built-in capacitor	CD			−	20	−	pF

MEASURING CIRCUIT



C1 . C2 . C3 : 0.1 μF
C4 : 30 pF
X-tal : 32.768 kHz

TYPICAL APPLICATION



C1 : 5 ~ 35pF
C2, C3, C4 : 0.1μF
B : 1.5V
L : 20mH

DESCRIPTION OF INSTRUCTIONS

	Mnemonic	Instruction Code														Operation
		13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Arithmetic operation	ADD ACC, AP	0	0	0	0	0	P	0	1	0	0	A				AP ← (AP) + (ACC)
	ADD #D, AP	0	1	1	0	0	P	D				A				AP ← (AP) + D
	SUB ACC, AP	0	0	0	0	1	P	0	1	0	0	A				AP ← (AP) − (ACC)
	SUB #D, AP	0	1	1	0	1	P	D				A				AP ← (AP) − D
	ADJUST N, AP	1	1	0	0	0	P	$\overline{N}+1$				A				AP ← N adjust {(AP)}
	CMP ACC, AP	0	0	0	0	1	P	1	1	1	0	A				(AP) − (ACC)
	CMP #D, AP	0	1	0	1	1	P	D				A				(AP) − D
	INC AP	0	1	1	0	0	P	0	0	0	1	A				AP ← (AP) + 1
	DEC AP	0	1	1	0	1	P	0	0	0	1	A				A ← (AP) − 1
	XOR ACC, AP	0	0	0	0	0	P	0	1	1	1	A				AP ← (AP) $\forall$ (ACC)
	XOR #D, AP	0	1	1	1	1	P	D				A				AP ← (AP) $\forall$ D
Bit operation	BIT ACC, AP	0	0	0	0	0	P	1	1	1	0	A				(AP) V $\overline{(ACC)}$
	BIT #D, AP	0	1	0	1	0	P	D				A				(AP) V $\overline{D}$
	BIS ACC, AP	0	0	0	0	0	P	0	1	1	0	A				AP V (ACC)
	BIS #D, AP	0	1	0	0	0	P	D				A				(AP) V D
	BIC ACC, AP	0	0	0	0	1	P	0	1	1	0	A				AP $\wedge$ $\overline{(ACC)}$
	BIC #D, AP	0	1	0	0	1	P	D				A				AP $\wedge$ $\overline{D}$
Shift	ASR AP	0	0	0	0	0	P	0	0	1	1	A				┌→(C) 0 → (AP)┘
	ASL AP	0	0	0	0	1	P	0	0	1	1	A				(C) ← (AP) ← 0
Flag operation	CLZ	0	0	0	0	0	0	1	0	1	0	0	0	0	0	Z ← 0
	CLC	0	0	0	0	0	0	1	0	0	1	0	0	0	0	C ← 0
	CLA	0	0	0	0	0	0	1	0	1	1	0	0	0	0	Z ← 0, C ← 0
	SEZ	0	0	0	0	1	0	1	0	1	0	0	0	0	0	Z ← 1
	SEC	0	0	0	0	1	0	1	0	0	1	0	0	0	0	C ← 1
	SEA	0	0	0	0	1	0	1	0	1	1	0	0	0	0	Z ← 1, C ← 1
Data transfer	MOV ACC, AP	1	1	1	1	0	P	0	0	0	0	A				AP ← (ACC)
	MOV ACC, AX	1	1	1	1	0	0	0	X			A				AX ← (ACC)
	MOV #D, AP	0	1	1	1	0	P	D				A				AP ← D
	MOV AP, ACC	1	1	1	1	1	P	0	0	0	0	A				ACC ← (AP)
	MOV AX, ACC	1	1	1	1	1	0	0	X			A				ACC ← (AX)
Jump	JMP adrs	1	0	0	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	PC ← adrs
	JMP @AP	0	0	0	0	0	P	1	1	0	1	A				PC ← (PC) + (AP) + 1
	JMPIO @AP	0	0	0	0	1	P	1	1	0	1	A				PC ← (PC) + {(AP) $\wedge$ 7H} +1
	BEQ +n BZE +n	0	0	0	1	1	0	0	1	0	n_4	n_3	n_2	n_1	n_0	PC ← (PC)+n+1, if Z=1

DESCRIPTION OF INSTRUCTIONS (CONT.)

	Mnemonic	Instruction Code 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Operation
Jump	BNE +n BNZ +n	0 0 0 1 1 0 1 1 0 n_4 n_3 n_2 n_1 n_0	PC ← (PC)+n+1, if Z=0
	BCS +n	0 0 0 1 1 0 0 0 0 n_4 n_3 n_2 n_1 n_0	PC ← (PC)+n+1, if C=1
	BCC +n	0 0 0 1 1 0 1 0 0 n_4 n_3 n_2 n_1 n_0	PC ← (PC)+n+1, if C=0
Input/Output	SWITCH AP	1 1 0 1 0 P 0 0 0 1 A	AP ← INPUT PORT (S1 ~ S4)
	KSWITCH AP	1 1 0 1 0 P 0 0 1 0 A	AP ← INPUT PORT (K1 ~ K4)
	MATRIX AP	1 1 0 1 1 P 0 0 1 0 A	OUTPUT PORT (M1 ~ M4) ←(AP)
	MATRIX Mn	0 0 0 1 0 0 0 0 1 0 M_4 M_3 M_2 M_1	OUTPUT PORT (M1 ~M4) ← Mn (n=1, 2, 3, 4)
	XTCP ON/OFF	0 0 0 1 0 0 1 0 0 0 0 0 b_1 b_0	XTOUT ON/OFF
	FREQ N	0 0 0 1 0 0 1 1 0 1 N	Freq ← N
	BUZZER sound	0 0 0 1 0 0 1 1 0 0 b_3 b_2 1 0	Mreg ← sound, Buzzer start
	BSO	0 0 0 1 0 0 1 1 0 0 0 0 0 0	Buzzer stop
	LAMP ON/OFF	0 0 0 1 0 0 0 0 0 1 0 0 b_1 b_0	LD ON/OFF
Display	DSP digit, AP	0 0 1 0 0 P digit A	Digit (Low part) ← (AP), (ACC)
	DSPH digit, AP	0 0 1 0 1 P digit A	Digit (High part) ← (AP), (ACC)
	FORMAT AP	1 1 0 1 1 P 0 0 1 1 A	FMT reg. ← (AP)
	FORMAT N	0 0 0 1 0 0 0 0 1 1 N	FMT reg. ← N
	DSPF digit, AP	0 0 1 1 0 P digit A	Digit (Low part) ← (AP) via table
	DSPFH digit, AP	0 0 1 1 1 P digit A	Digit (High part) ← (AP) via table
CPU Control & Others	HALT	0 0 0 1 0 0 0 0 0 0 0 0 0 0	Halt
	INTENAB 32/16	0 0 0 1 0 0 1 0 1 1 1 0 0 0	Enable timer
		0 0 0 1 0 0 0 1 0 0 0 0 1 0	
	INTDSAB 32/16	0 0 0 1 0 0 1 0 1 1 0 1 0 0	Disable timer
		0 0 0 1 0 0 0 1 0 0 0 0 0 1	
	INTMODE AP	1 1 0 1 0 P 0 1 0 0 A	AP ← Interrupt mode
	PAGE A0	1 1 0 1 1 0 0 1 0 1 A	Preg ← (A0)
	PAGE N	0 0 0 1 0 0 0 1 0 1 N	Preg ← N
	ADRS AP	1 1 0 1 1 P 0 1 1 0 A	Areg ← (AP)
	ADRS N	0 0 0 1 0 0 0 1 1 0 N	Areg ← N
	RATE AP	1 1 0 1 0 P 1 0 0 1 A	AP ← DIVIDER (8 Hz~1 Hz)
	RSTRATE	0 0 0 1 0 0 1 0 0 0 1 0 0 0	DIVIDER (8 Hz~1 Hz) ← 0
	BACKUP ON/OFF	0 0 0 1 0 0 0 0 0 1 b_3 b_2 0 0	Backup ON/OFF
	NOP	0 0 0 0 0 0 0 0 0 0 0 0 0 0	No operation

MSM5056

CMOS 4BIT SINGLE CHIP VERY LOW POWER MICROCONTROLLER WITH LCD DRIVER

GENERAL DESCRIPTION

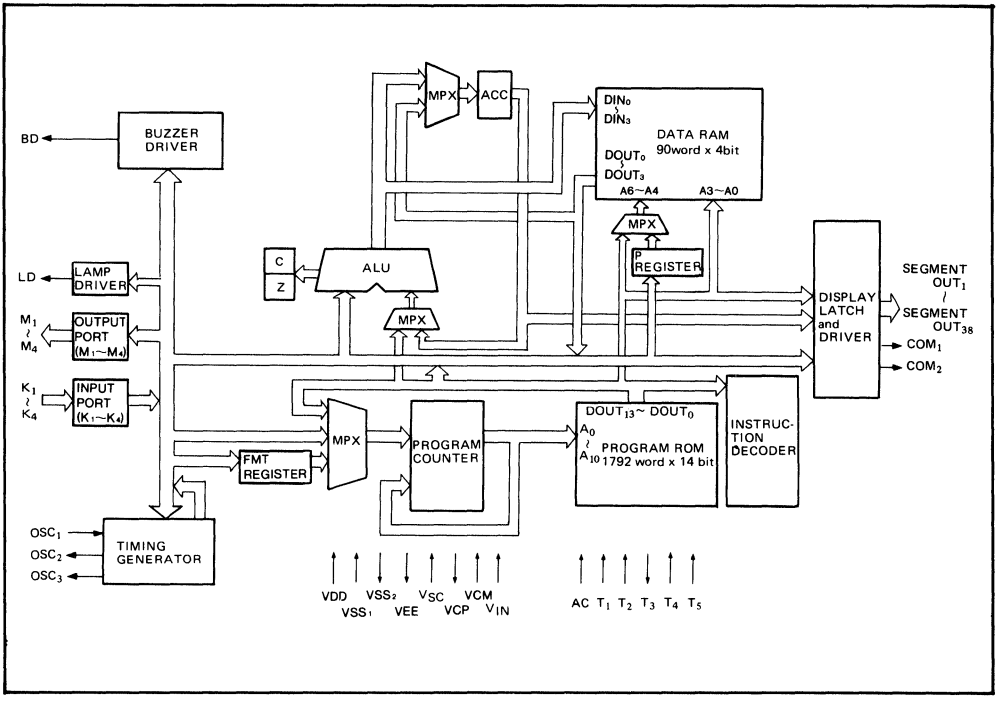
The OKI MSM5056 is a low-power, high-performance single-chip microcontroller employing complementary metal oxide semiconductor technology. Integrated onto a single chip are a 4-bit ALU, 25K bits of mask programmable ROM, 360 bits of data RAM, crystal oscillator, voltage doubler, timer, LCD driver, input port, output port and overcharge protection circuit for connection to a solar cell.

The MSM5056 is widely used in electronic products requiring low power operation, for example, solar calculator watches and games.

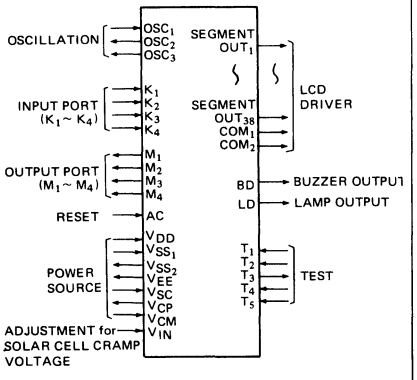
FEATURES

- Low Power Consumption 3 μA Typical
- 1792 × 14 Internal ROM
- 90 × 4 Internal RAM
- 4 Input Port
- 4 Output Port
- 4 × 4 Key Matrix Input (K_1~K_4, M_1~M_4)
- 38 LCD Driver
  (1/2 Duty, 1/2 Bias, 88 Segment)
- 42 Instructions
- 1.5 V Operating Voltage
  (The solar cell can be connected.)
- 32.768 kHz Crystal Oscillator
- 122.1 μs Instruction Cycle
- −20 to 75°C Operating Temperature
- 68 pad die

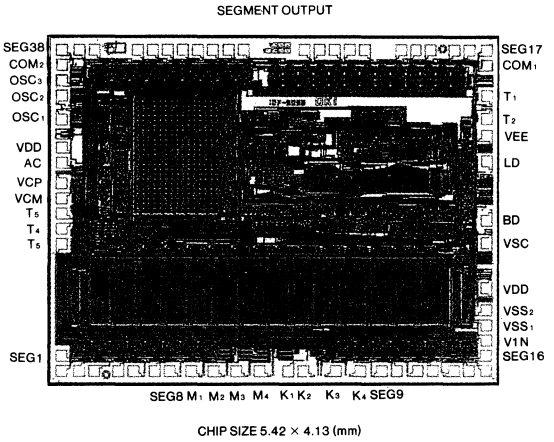
FUNCTIONAL BLOCK DIAGRAM



LOGIC SYMBOL



CHIP PAD LAYOUT



CHIP SIZE 5.42 × 4.13 (mm)

PIN DESCRIPTION

Designation	Function
V_{DD}	Circuit ground potential
V_{SS_1}	Power source (−1.5 V)
V_{SC}	Solar cell connection terminal
V_{SS_2}	Power source for LCD driver (−3.0 V) This terminal is connected to V_{DD} terminal through a 0.1 μF capacitor.
V_{EE}	Power source for internal logic (−1.5 to −3.0 V) This terminal is connected to V_{DD} terminal through a 0.1 μF capacitor.
V_{CP}, V_{CM}	Booster capacitor connection terminals V_{CP} terminal is connected to V_{CM} terminal through a 0.1 μF capacitor.
XT, $\overline{XT}$	Input and output terminals of oscillator inverter, 32.768 kHz crystal is connected to these terminals.
$T_1 \sim T_5$	Terminals to test internal logic, $T_1 \sim T_3$ and T_5 are pulled down to V_{SS_1}. T_4 is output. Test pins must be normaly open.
AC	Terminal to clear internal logic pulled down to V_{SS_1}. After power is turned on, the MSM5056 must be reset by this terminal.
BD	Buzzer output
LD	Lamp output
V_{IN}	Adjustment for solar cell cramp voltage This terminal is connected to V_{SS_1} terminal through 50 ~ 200 kΩ resistor.

FUNCTIONAL DESCRIPTION

A block diagram of the MSM5056 is given on page 111. Each block of logic will be briefly discussed. For more information, please refer to the MSM5056 user's manual.

Program ROM

The MSM5056 will address up to 1.75 K words of internal mask programmable ROM. Each word consists of 14 bits and all instructions are one word. The instructions are routed to a programmed logic array which generates the signals necessary for control of logic.

Data RAM

Data is organized in 4 bit nibbles. Internal data RAM consists of 90 nibbles.

The RAM is addressed by page address and column address. Normally page address is specified with the page register, but direct addressing is available at Page 0.

Column address is directly addressed by the operands of various instructions.

ALU

The ALU performs 4-bit parallel operation of RAM and AC contents, or RAM contents and an immediate digit. It sets or resets the flags (Z, C) depending on the condition.

Program Counter (PC)

The program counter is an 11-bit wide counter and specifies the address of the program ROM.

The PC is incremented by one at every execution of an instruction, and then specifies the next instruction to be executed. However, the contents of the PC are rewritten by the execution of a Jump or Branch instruction.

There is no boundary in the ROM, so a Jump or Branch instruction can be put anywhere in the ROM.

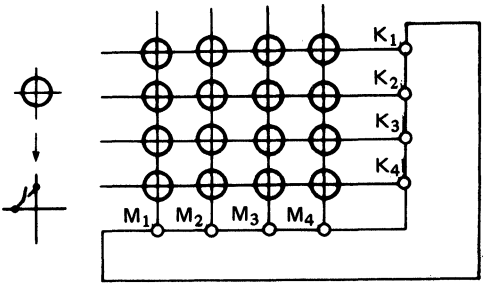
Input/Output Port

Input Port (K1 ~ K2)

The input port (K1 ~ K4) is a 4-bit parallel input port. Each pin of the port is pulled down to V_{SS1} by an internal resistor, and the status of the port can be fetched by an input instruction.

Output Port (M1 ~ M4)

The output port (M1 ~ M4) is a 4-bit parallel output port. This port consists of data latches and buffers. The contents of data latches are rewritten by an output instruction. A key matrix is used in combination with K1 to K4.



Display Function

The MSM5056 is provided with a segment output terminal which can directly drive a 1/2 bias, 1/2 duty LCD and common drive output terminals. COM1 and COM2. The segment drive circuit consists of the display data latch, multiplexer and driver. If the data is sent to the display data latch with the display instruction, the LCD drive waveform is output to the segment drive output terminal.

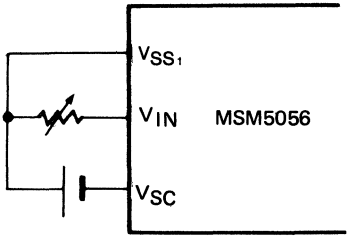
Time Base

Time base of the CPU is provided by connecting a 32.768 kHz crystal to the OSC1 and OSC3 pin. One machine cycle is 122.1 μs.

A hardware divider up to 1 Hz is provided enabling programs to implement and a clock function by counting signals between 16 and 1 Hz.

Solar Cell Overcharge Protection Circuit

When a solar cell is connected to prolong the usefull life of the battery, a resistor is inserted between the V_{IN} pin and V_{SS1} to adjust the overcharge protection voltage.



ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage 1	$V_{DD} - V_{SS1}$	Ta = 25°C	−0.3 to +3.0	V
Supply Voltage 2	$V_{DD} - V_{SC}$	Ta = 25°C	−0.3 to +3.5	V
Supply Voltage 3	$V_{DD} - V_{SS2}$	Ta = 25°C	−0.3 to +6.0	V
Supply Voltage 4	$V_{DD} - V_{EE}$	Ta = 25°C	−0.3 to +6.0	V
Input Voltage	V_{IN1}	Ta = 25°C	V_{SS1} −0.3 to +0.3	V
Storage Temperature	Tstg		−55 to 125	°C

OPERATING CONDITIONS

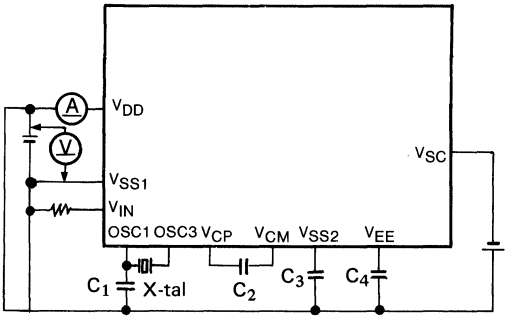
Parameter	Symbol	Limits	Unit
Operating Voltage	$V_{DD} - V_{SS1}$	1.25 to 1.65	V
Operating Temperature	Topr	−20 to 75	°C

DC CHARACTERISTICS

(V_{DD} =0V, V_{SS1}, V_{EE} = −1.55V, V_{SS2} = −3.0V, C_I = 30kΩ, C_G = 30pF, Ta = 25°C)

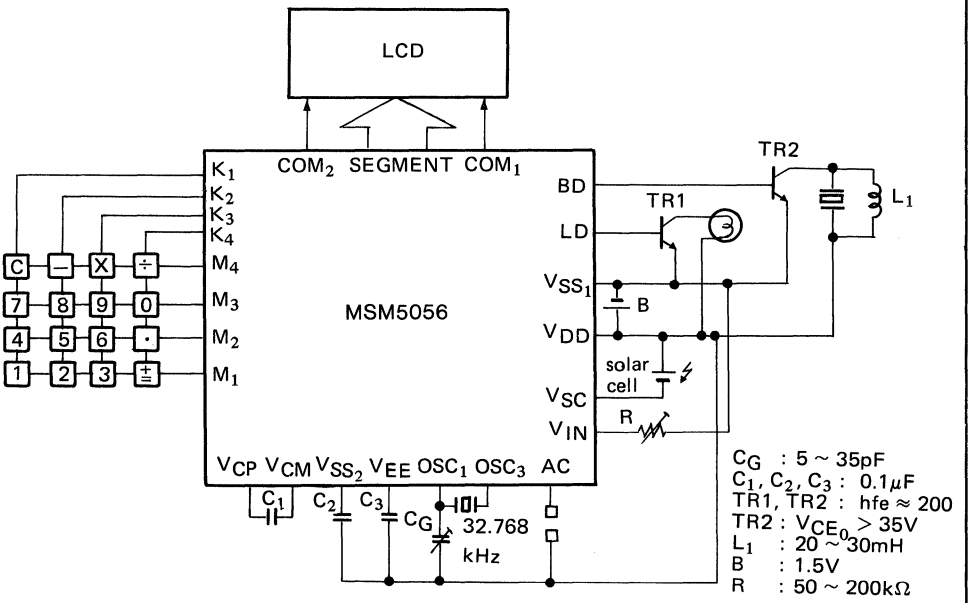
Parameter	Symbol	Condition		Limits			Unit
				Min.	Typ.	Max.	
Operating voltage 1	$-V_{SS1}$	V_{SS1} terminal		1.25	1.55	2.0	V
Operating voltage 2	$-V_{SC}$	V_{SC} terminal		0	2.0	3.0	V
Power supply current	I_{DD}	V_{SS1} terminal		−	3.0	−	μA
Oscillation start voltage	$-V_{OSC}$	Within 10 seconds V_{SS1} terminal		1.45	−	−	V
Output current 1 COM	I_{OH1}	V_{OH1} = −0.2V		−4	−	−	μA
	I_{OM1}	$V_{OM1} = V_{SS1} \pm 0.2V$		4/−4	−	−	
	I_{OL1}	V_{OL1} = −2.8V		4	−	−	
Output current 2 SEGMENT	I_{OH2}	V_{OH2} = −0.2V		−0.4	−	−	μA
	I_{OL2}	V_{OL2} = −2.8V		0.4	−	−	
Output current 3 $M_1 \sim M_4$	I_{OH3}	V_{SS1}, V_{EE} −1.25V	V_{OH3} −0.4 V	−100	−	−	μA
	I_{OL3}	V_{SS2} −2.3V	V_{OL3} −0.85V	3	−	8	
Output current 4 BD	I_{OH4}	V_{OH4} = −0.4V		−50	−100	−200	μA
	I_{OL4}	V_{OL4} = −1.15V		3	10	30	
Input current $K_1 \sim K_4$	I_{IH1}	V_{IN} = −0V		5	10	15	μA
	I_{IL1}	V_{IN} = −1.55V		−	−	−0.2	
Oscillator built-in capacitor	CD			−	20	−	pF
Solar battery cramp resistor	R_{IN}	V_{SS1} = −1.8V V_{IN} terminal		50	−	200	kΩ

MEASURING CIRCUIT



C_1 : 30pF
C_2, C_3, C_4 : 0.1μF
X-tal : 32.768kHz

TYPICAL APPLICATION



C_G : 5 ~ 35pF
C_1, C_2, C_3 : 0.1μF
TR1, TR2 : hfe ≈ 200
TR2 : V_{CE_0} > 35V
L_1 : 20 ~ 30mH
B : 1.5V
R : 50 ~ 200kΩ

DESCRIPTION OF INSTRUCTIONS

	Mnemonic	Instruction Code														Operation
		13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Arithmetic operation	ADD ACC, AP	0	0	0	0	0	P	0	1	0	0		A			AP ← (AP) + (ACC)
	ADD #D, AP	0	1	1	0	0	P		D				A			AP ← (AP) − D
	ADC AP	0	0	0	0	0	P	0	1	0	1		A			AP ← Decimal adjust {(AP) + (ACC) + (C)}
	SUB ACC, AP	0	0	0	0	1	P	0	1	0	0		A			AP ← (AP) − (ACC)
	SUB #D, AP	0	1	1	0	1	P		D				A			AP ← (AP) − D
	SBC AP	0	0	0	0	1	P	0	1	0	1		A			AP ← Decimal adjust {(AP) − (ACC) − (C)}
	CMP ACC, AP	0	0	0	0	1	P	1	1	1	0		A			(AP) − (ACC)
	CMP #D, AP	0	1	0	1	1	P		D				A			(AP) − D
	INC AP	0	1	1	0	0	P	0	0	0	1		A			AP ← (AP) + 1
	DEC AP	0	1	1	0	1	P	0	0	0	1		A			A ← (AP) − 1
	XOR ACC, AP	0	0	0	0	0	P	0	1	1	1		A			AP ← (AP) ∀ (ACC)
	XOR #D, AP	0	1	1	1	1	P		D				A			AP ← (AP) V D
Bit operation	BIT ACC, AP	0	0	0	0	0	P	1	1	1	0		A			(AP) V $\overline{(ACC)}$
	BIT #D, AP	0	1	0	1	0	P		D				A			(AP) V $\overline{D}$
	BIS ACC, AP	0	0	0	0	0	P	0	1	1	0		A			AP ← (AP) V (ACC)
	BIS #D, AP	0	1	0	0	0	P		D				A			AP ← (AP) V D
	BIC ACC, AP	0	0	0	0	1	P	0	1	1	0		A			AP ← (AP) Λ $\overline{(ACC)}$
	BIC #D, AP	0	1	0	0	1	P		D				A			AP ← (AP) Λ $\overline{D}$
Shift	ASR AP	0	0	0	0	0	P	0	0	1	1		A			⌊→(C) 0 → (AP)⌋
	ASL AP	0	0	0	0	1	P	0	0	1	1		A			(C) ← (AP) ← 0
Flag operation	CLZ	0	0	0	0	0	0	1	0	1	0	0	0	0	0	Z ← 0
	CLC	0	0	0	0	0	0	1	0	0	1	0	0	0	0	C ← 0
	CLA	0	0	0	0	0	0	1	0	1	1	0	0	0	0	Z ← 0, C ← 0
	SEZ	0	0	0	0	1	0	1	0	1	0	0	0	0	0	Z ← 1
	SEC	0	0	0	0	1	0	1	0	0	1	0	0	0	0	C ← 1
	SEA	0	0	0	0	1	0	1	0	1	1	0	0	0	0	Z ← 1, C ← 1
Data transfer	MOV ACC, AP	1	1	1	1	0	P	0	0	0	0		A			AP ← (ACC)
	MOV ACC, AX	1	1	1	1	0	0	0		X			A			AX ← (ACC)
	MOV #D, AP	0	1	1	1	0	P		D				A			AP ← D
	MOV AP, ACC	1	1	1	1	1	P	0	0	0	0		A			ACC ← (AP)
	MOV AX, ACC	1	1	1	1	1	0	0		X			A			ACC ← (AX)
	CHG AP	1	1	1	0	0	P	0	0	0	0		A			(ACC) ⟷ (AP)
	CHG AX	1	1	1	0	0	0	0		X			A			(ACC) ⟷ (AX)

DESCRIPTION OF INSTRUCTIONS (CONT.)

	Mnemonic	Instruction Code 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Operation
Jump	JMP adrs	1 0 0 a_{10} a_9 a_8 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	PC ← adrs
	JMP @AP	0 0 0 0 0 P 1 1 0 1 A	PC ← (PC) + (AP) + 1
	JMPIO @AP	0 0 0 0 1 P 1 1 0 1 A	PC ← (PC) + {(AP)∧7H} + +1
	BEQ +n BZE +n	0 0 0 1 1 0 0 1 0 n_4 n_3 n_2 n_1 n_0	PC ← (PC)+n+1, if Z=1
	BNE +n BNZ +n	0 0 0 1 1 0 1 1 0 n_4 n_3 n_2 n_1 n_0	PC ← (PC)+n+1, if Z=0
	BCS +n BLT +n	0 0 0 1 1 0 0 0 1 n_4 n_3 n_2 n_1 n_0	PC ← (PC)+n+1, if C=1
	BCC +n BGE +n	0 0 0 1 1 0 1 0 1 n_4 n_3 n_2 n_1 n_0	PC ← (PC)+n+1, if C=0
	BGT +n	0 0 0 1 1 0 1 1 1 n_4 n_3 n_2 n_1 n_0	PC ← (PC)+n+1, if Z=0 and C=0
	BLE +n	0 0 0 1 1 0 0 1 1 n_4 n_3 n_2 n_1 n_0	PC ← (PC)+n+1, if Z=1 or C=1
Input/Output	INP Port, AP	1 1 0 1 0 P Port A	AP ← (Port)
	OUT AP, Port	1 1 0 1 1 P Port A	Port ← (AP)
	OUT #D, Port	0 0 0 1 0 0 Port D	Port ← D
Display	DSP digit, AP	0 0 1 0 0 P digit A	digit ← (AP), (ACC)
	DSPF digit, AP	0 0 1 1 0 P digit A	digit ← (AP) via table
CPU control	HALT	0 0 0 1 0 0 0 0 0 0 0 0 0 0	Halt CPU
	NOP	0 0 0 0 0 0 0 0 0 0 0 0 0 0	No Operation

MSM6051

CMOS 4BIT HIGH PERFORMANCE SINGLE CHIP VERY LOW POWER MICROCONTROLLER WITH LCD DRIVER

GENERAL DESCRIPTION

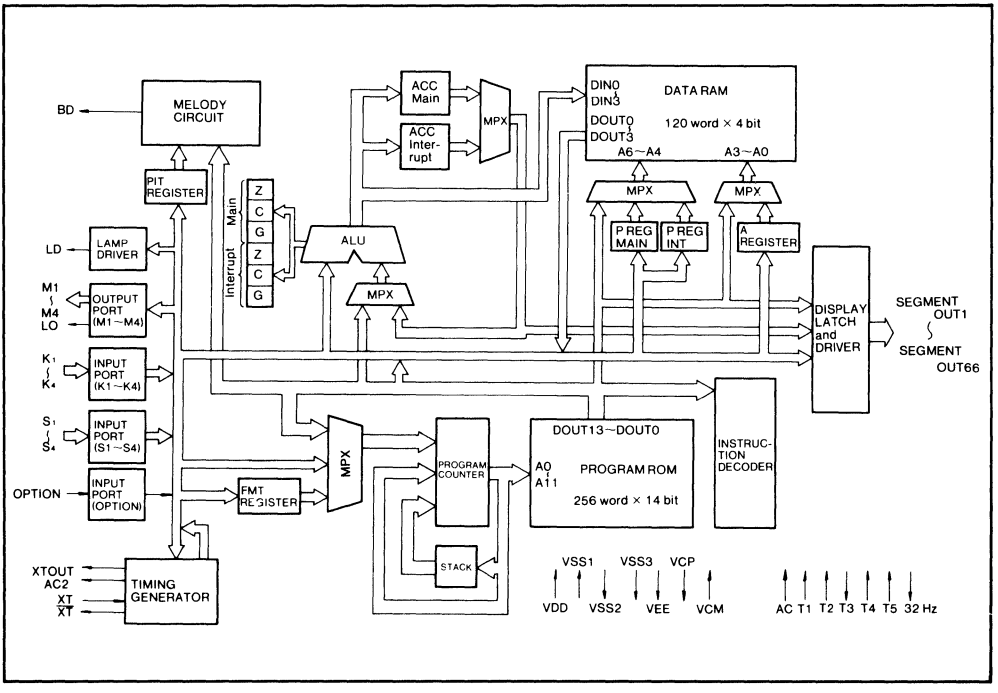
OKI's MSM6051 is a low-power and high-performance single-chip microcontroller employing complementary metal oxide semiconductor technology. Integrated onto a single chip are a 4-bit ALU, 35K bits of mask programmable ROM, 480 bits of data RAM, crystal oscillator, voltage doubler, timer, LCD driver, input port and output port.

The MSM6051 is widely used in electronic products requiring low power operation, for example, stopwatches with lap time memory, calculator watches and handy terminals.

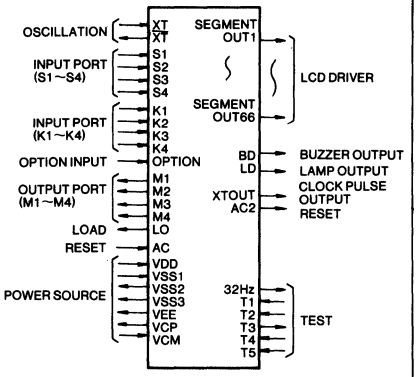
FEATURES

- Low Power Consumption 3 μA Typical
- 2560 × 14 Internal ROM
- 120 × 4 Internal RAM
- 9 Input Port
- 4 Output Port
- 4 × 4 Key Matrix Input (K1~K4, M1~M4)
- 66 LCD Driver (including 3 common) (1/3 Duty, 1/3 Bias, 189 Segment)
- 59 Instructions
- 1.5 V or 3 V Operating Voltage (Masking Option)
- 32.768 kHz crystal Oscillator
- 91.5 μs Instruction Cycle
- −20 to 75°C Operating Temperature
- 101 pad die

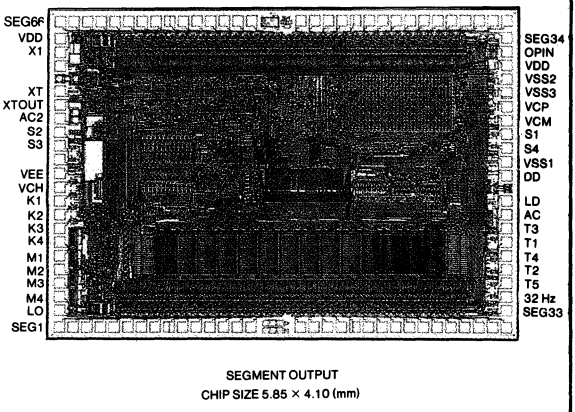
FUNCTIONAL BLOCK DIAGRAM



LOGIC SYMBOL



CHIP PAD LAYOUT



SEGMENT OUTPUT
CHIP SIZE 5.85 × 4.10 (mm)

PIN DESCRIPTION

Designation	Function
V_{DD}	Circuit ground potential
V_{SS_1}	Power source (−1.5 V)
V_{SS_2}	Power source for LCD driver (−3.0 V) This terminal is connected to V_{DD} terminal through a 0.1 μF capacitor.
V_{SS_3}	Power source for LCD driver (−4.5 V) This terminal is connected to V_{DD} terminal through a 0.1 μF capacitor.
V_{EE}	Power source for internal logic (−1.5 to −3.0 V) This terminal is connected to V_{DD} terminal through a 0.1 μF capacitor.
V_{CP}, V_{CM}	Booster capacitor connection terminals V_{CP} terminal is connected to V_{CM} terminal through a 0.1 μF capacitor.
XT, $\overline{XT}$	Input and output terminals of oscillator inverter, 32.768 kHz crystal is connected to these terminals.
$T_1 \sim T_5$	Terminals to test internal logic, $T_1 \sim T_3$ and T_5 are pulled down to V_{SS_1}. T_4 is output. Test pins must be normally open.
AC	Terminal to clear internal logic pulled down to V_{SS_1}. After power is turned on, the MSM6051 must be reset by this terminal.
BD	Buzzer output
LD	Lamp output
LO	Load data terminal of M_1 to M_4.
AC2	Reset terminal for external circuit.
XT OUT	Clock output for external circuit.

FUNCTIONAL DESCRIPTION

A block diagram of the MSM6051 is given on page 118. Each block of logic will be briefly discussed. For more information, please refer to the MSM6051 user's manual.

Program ROM

The MSM6051 addresses up to 2.5 K words of internal mask programmable ROM. Each word consists of 14 bits, and all instructions are one word. The instructions are routed to a programmed logic array which generates the signals necessary for control of logic.

Data RAM

Data is organized in 4 bit nibbles. Internal data RAM consists of 120 nibbles.

The RAM is addressed by page address and column address. Normally page address is specified the with page register, but direct addressing is available at Page 0.

Column address is directly addressed by operand of various instructions.

ALU

The ALU performs 4-bit parallel operation of RAM and ACC contents, or RAM contents and an immediate digit. It sets or resets the flags (Z, C, G) depending on the condition.

Program Counter (PC)

The program counter is 12 bits wide and specifies the address of the program ROM.

The PC is incremented by one at every execution of the instruction, and then specifies the next instruction to be executed. However, the contents of the PC are rewritten by the execution of a Jump, Call or Branch instruction.

There is no boundary in the ROM, so a Jump or Branch instruction can be put anywhere in the ROM.

Stack

The MSM6051 has a 3 level stack apart from data RAM. The contents of the PC are loaded into the stack when a call instruction is executed or an interrupt is generated.

Input/Output Port

Input Port (S1 ~ S4)

The input port (S1 ~ S4) is a 4-bit parallel input port. Each pin of the port is pulled down to V_{SS_1} by an internal resistor, and the status of the port is fetched by a SWITCH instruction.

Input Port (K1 ~ K4)

The input port (K1 ~ K4) is a 4-bit parallel input port. Each pin of the port is pulled down to V_{SS_1} by an internal resistor, and the status of the port is fetched by a KSWITCH instruction.

Input Port (OPIN)

The input port (OPIN) is single input port. OPIN is pulled down to V_{SS_1} by an internal power save circuit, and the status of the port is fetched by an input instruction.

Output Port (M1 ~ M4)

The output port (M1 ~ M4) is a 4-bit parallel output port. This port consists of data latches and buffers, and the contents of data latches are rewritten by a matrix instruction.

Display Function

The MSM6051 is provided with the segment output terminal which can directly drive a 1/3 bias, 1/3 duty LCD, and the common drive output terminals COM1, COM2 and COM3.

The segment drive circuit consists of the display data latch, multiplexer and driver. If the data is sent to the display data latch with the display instruction, the LCD drive waveform is output to the segment drive output terminal.

Time Base

The time base of the CPU is provided by connecting 32.768 kHz crystal to the XT and $\overline{XT}$ pin. One machine cycle is 91.5 μs.

A hardware divider up to 1 Hz is provided enabling programs to implement a clock function by counting signals between 32 and 1 Hz.

Also, a 1/100 second digit counting function is provided as a hardware feature to make for easy implementation of a stopwatch function.

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage 1	$V_{DD} - V_{SS_1}$	Ta = 25°C	−2.0 to +0.3	V
Supply Voltage 2	$V_{DD} - V_{SS_2}$	Ta = 25°C	−4.0 to +0.3	V
Supply Voltage 3	$V_{DD} - V_{SS_3}$	Ta = 25°C	−6.0 to +0.3	V
Supply Voltage 4	$V_{DD} - V_{EE}$	Ta = 25°C	−4.0 to +0.3	V
Input Voltage	V_{IN_1}	Ta = 25°C	V_{SS_1} −0.3 to +0.3	V
Storage Temperature	Tstg		−55 to 125	°C

OPERATING CONDITIONS

Parameter	Symbol	Limits	Unit
Operating Voltage	$V_{DD} - V_{SS_1}$	1.25 to 1.65	V
Operating Temperature	Topr	−20 to 75	°C

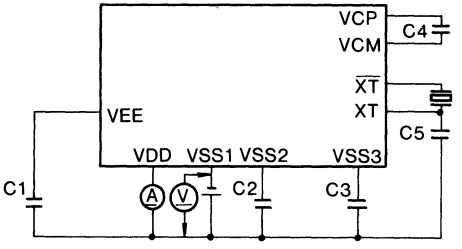
DC CHARACTERISTICS

(V_{DD} =0V, V_{SS_1}, V_{EE} = −1.55V, V_{SS_2} = −3.0V, V_{SS_3} = −4.5V, C_l = 30kΩ, Ta = 25°C)

Parameter	Symbol	Conditions		Limits			Unit
				Min.	Typ.	Max.	
Power supply current	I_{DD}	V_{SS} terminal		−	3.0	−	μA
Oscillation start voltage	$-V_{OSC}$	Within 5 seconds V_{SS_1} terminal		1.45	−	−	V
Output current 1 COM	I_{OH_1}	V_{OH_1} = −0.2V		−4	−	−	μA
	I_{OMH_1}	$V_{OMH_1} = V_{SS_1} \pm 0.2$		4/−4	−	−	
	I_{OML_1}	$V_{OML_1} = V_{SS_2} \pm 0.2$		4/−4	−	−	
	I_{OL_1}	V_{OL_1} = −4.3V		4	−	−	
Output current 2 SEGMENT	I_{OH_2}	V_{OH_2} = −0.2V		−4	−	−	μA
	I_{OMH_2}	$V_{OMH_2} = V_{SS_1} \pm 0.2$		4/−4	−	−	
	I_{OML_2}	$V_{OML_2} = V_{SS_2} \pm 0.2$		4/−4	−	−	
	I_{OL_2}	V_{OL_2} = −4.3V		4	−	−	
Output current 3 AC2 LOAD XTOUT	I_{OH_3}	V_{OH_3} = −0.5 V		−10	−	−	μA
	I_{OL_3}	V_{OL_3} = −1.15V		10	−	−	
Output current 4 $M_1 \sim M_4$	I_{OH_4}	V_{OH_4} = −0.5V		−100	−	−	μA
	I_{OL_4}	V_{OL_4} = −1.0V		2	−	10	
Output current 5 LD	I_{OH_5}	V_{OH_5} = −0.55V	V_{SS_1} = −1.25V, V_{EE} = −2.0V, V_{SS_2} = −2.0V	−12.5	−25	−83	μA
	I_{OL_5}	V_{OL_5} = −0.85V		1	−	−	
Output current 6 BD	I_{OH_6}	V_{OH_6} = −0.55V	V_{SS_1} = −1.25V, V_{EE} = −2.0V, V_{SS_2} = −2.0V	−17	−30	−62	μA
	I_{OL_6}	V_{OL_6} = −0.85V		−	5	−	

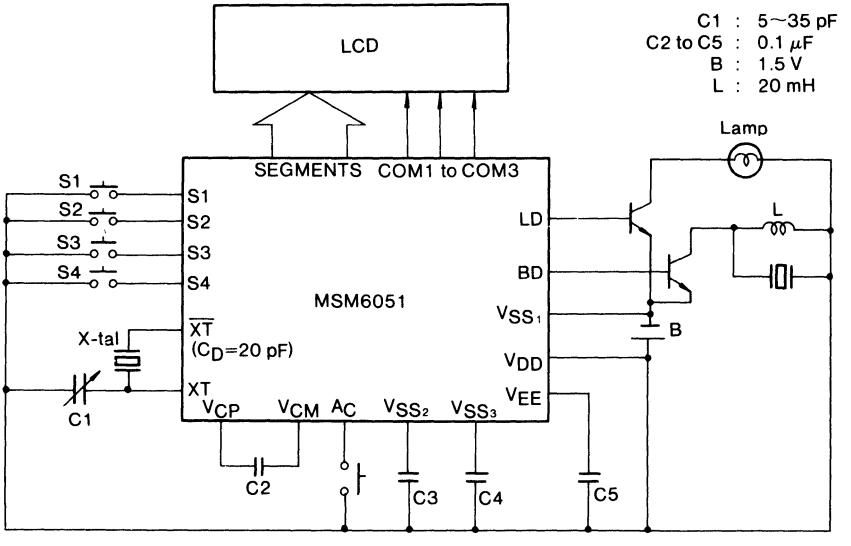
Parameter	Symbol	Conditions	Limits			Unit
			Min.	Typ.	Max.	
Input current 1 S_1~S_4	I_{IH_1}	V_{IH_1} = 0V	2	20	100	μA
	I_{IL_1}	V_{IL_1} = −1.55V	–	–	−0.2	
Input current 2 K_1~K_4	I_{IH_2}	V_{IH_2} = 0V	5	13	26	μA
	I_{IL_2}	V_{IL_2} = −1.55V	–	–	−0.2	
Input current 3 OPIN	I_{IH_3}	V_{IH_3} = 0V	–	30	45	μA
	I_{IL_3}	V_{IL_3} = −1.55V	–	–	−0.2	
Oscillator built-in capacitance	CD		–	20	–	pF

MEASURING CIRCUIT



C1 . C2 . C3 . C4 : 0.1 μF
C5 : 30 pF
X-tal : 32.768 kHz

TYPICAL APPLICATION



C1 : 5~35 pF
C2 to C5 : 0.1 μF
B : 1.5 V
L : 20 mH

DESCRIPTION OF INSTRUCTIONS

	Mnemonic	Instruction Code														Operation
		13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Arithmetic operation	ADD ACC, AP	0	0	0	0	0	P	0	1	0	0	A				AP ← (AP) + (ACC)
	ADD #D, AP	0	1	1	0	0	P	D				A				AP ← (AP) + D
	ADC AP	0	0	0	0	0	P	0	1	0	1	A				AP ← Decimal adjust {(AP) + (ACC) + (C)}
	ADCN AP	1	1	0	0	0	P	N				A				AP ← N adjust {(AP) + (C)}
	SUB ACC, AP	0	0	0	0	1	P	0	1	0	0	A				AP ← (AP) − (ACC)
	SUB #D, AP	0	1	1	0	1	P	D				A				AP ← (AP) − D
	SBC AP	0	0	0	0	1	P	0	1	0	1	A				AP ← Decimal adjust {(AP) − (ACC) − (C)}
	SBCN AP	1	1	0	0	1	P	N				A				AP ← N adjust {(AP) − (C)}
	CMP ACC, AP	0	0	0	0	1	P	1	1	1	0	A				(AP) − (ACC)
	CMP #D, AP	0	1	0	1	1	P	D				A				(AP) − D
	INC AP	1	1	1	0	0	P	0	0	0	0	A				AP ← (AP) + 1
	INC AX	1	1	1	0	0	0	0	X			A				AX ← (AX) + 1
	DEC AP	1	1	1	0	1	P	0	0	0	0	A				AP ← (AP) − 1
	DEC AX	1	1	1	0	1	0	0	X			A				AX ← (AX) − 1
	XOR ACC, AP	0	0	0	0	0	P	0	1	1	1	A				AP ← (AP) ∀ (ACC)
	XOR #D, AP	0	1	1	1	1	P	D				A				AP ← (AP) ∀ D
Bit operation	BIT ACC, AP	0	0	0	0	0	P	1	1	1	0	A				(AP) V ($\overline{ACC}$)
	BIT #D, AP	0	1	0	1	0	P	D				A				(AP) V $\overline{D}$
	BIS ACC, AP	0	0	0	0	0	P	0	1	1	0	A				(AP) V (ACC)
	BIS #D, AP	0	1	0	0	0	P	D				A				(AP) V D
	BIC ACC, AP	0	0	0	0	1	P	0	1	1	0	A				(AP) ∧ ($\overline{ACC}$)
	BIC #D, AP	0	1	0	0	1	P	D				A				(AP) ∧ $\overline{D}$
Rotate/Shift	ROR AP	0	0	0	0	0	P	0	0	1	0	A				⌐→ (C) → (AP) ⌐
	ROL AP	0	0	0	0	1	P	0	0	1	0	A				⌐ (C) ← (AP) ← ⌐
	ASR AP	0	0	0	0	0	P	0	0	1	1	A				⌐→ (C) 0 → (AP) ⌐
	ASL AP	0	0	0	0	1	P	0	0	1	1	A				(C) ← (AP) ← 0
Flag operation	CLZ	0	0	0	0	0	0	1	0	1	0	0	0	0	0	Z ← 0
	CLC	0	0	0	0	0	0	1	0	0	1	0	0	0	0	C ← 0
	CLG	0	0	0	0	0	0	1	0	0	0	0	0	0	0	G ← 0
	CLA	0	0	0	0	0	0	1	0	1	1	0	0	0	0	Z ← 0, C ← 0, G ← 0
	SEZ	0	0	0	0	1	0	1	0	1	0	0	0	0	0	Z ← 1
	SEC	0	0	0	0	1	0	1	0	0	1	0	0	0	0	C ← 1

DESCRIPTION OF INSTRUCTIONS (CONT.)

	Mnemonic	Instruction Code														Operation
		13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	SEG	0	0	0	0	1	0	1	0	0	0	0	0	0	0	G ← 1
	SEA	0	0	0	0	1	0	1	0	1	1	0	0	0	0	Z ← 1, C ← 1, G ← 1
Data transfer	MOV ACC, AP	1	1	1	1	0	P	0	0	0	0	A				AP ← (ACC)
	MOV ACC, AX	1	1	1	1	0	0	0		X		A				AX ← (ACC)
	MOV #D, AP	0	1	1	1	0	P	D				A				AP ← D
	MOV AP, ACC	1	1	1	1	1	P	0	0	0	0	A				ACC ← (AP)
	MOV AX, ACC	1	1	1	1	1	0	0		X		A				ACC ← (AX)
Sub-routine	CALL adrs	1	0	1	1	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	STACK ← (PC), PC ← adrs
	RET	0	0	0	0	0	0	1	1	0	0	0	0	0	0	PC ← (STACK) + 1
	RTI	0	0	0	0	1	0	1	1	0	0	0	0	0	0	PC ← (STACK) + 1, or PC ← (STACK) ACC ↔ ACC', Z ↔ Z', C ↔ C' G ↔ G', Preg ↔ P'reg
Jump	JMP adrs	1	0	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	PC ← Adrs
	JMP @AP	0	0	0	0	0	P	1	1	0	1	A				PC ← (PC) + (AP) + 1
	JMPIO @AP	0	0	0	0	1	P	1	1	0	1	A				PC ← (PC) + {(AP) ∧ 7H} +1
	BEQ +n BZE +n	0	0	0	1	1	0	0	1	0	n_4	n_3	n_2	n_1	n_0	PC ← (PC)+n+1, if Z=1
	BEQ −n BZE −n	0	0	0	1	1	1	0	1	0	n_4	n_3	n_2	n_1	n_0	PC ← (PC) −n, if Z=1
	BNE +n BNZ +n	0	0	0	1	1	0	1	1	0	n_4	n_3	n_2	n_1	n_0	PC ← (PC) +n + 1, if Z=0
	BNE −n BNZ −n	0	0	0	1	1	1	1	1	0	n_4	n_3	n_2	n_1	n_0	PC ← (PC) −n, if Z=0
	BCS +n	0	0	0	1	1	0	0	0	0	n_4	n_3	n_2	n_1	n_0	PC ← (PC)+n+1, if C=1
	BCS −n	0	0	0	1	1	1	0	0	0	n_4	n_3	n_2	n_1	n_0	PC ← (PC) −n, if C=1
	BCC +n	0	0	0	1	1	0	1	0	0	n_4	n_3	n_2	n_1	n_0	PC ← (PC)+n+1, if C=0
	BCC −n	0	0	0	1	1	1	1	0	0	n_4	n_3	n_2	n_1	n_0	PC ← (PC)−n, if C=0
	BGT +n	0	0	0	1	1	0	0	0	1	n_4	n_3	n_2	n_1	n_0	PC ← (PC)+n+1, if G=1
	BGT −n	0	0	0	1	1	1	0	0	1	n_4	n_3	n_2	n_1	n_0	PC ← (PC)−n, if G=1
	BLE +n	0	0	0	1	1	0	1	0	1	n_4	n_3	n_2	n_1	n_0	PC ← (PC)+n+1, if G=0
	BLE −n	0	0	0	1	1	1	1	0	1	n_4	n_3	n_2	n_1	n_0	PC ← (PC)−n, if G=0
	BGE +n	0	0	0	1	1	0	0	1	1	n_4	n_3	n_2	n_1	n_0	PC ← (PC)+n+1, if Z=1 or G=1
	BGE −n	0	0	0	1	1	1	0	1	1	n_4	n_3	n_2	n_1	n_0	PC ← (PC)−n, if Z=1 or G=1
	BLT +n	0	0	0	1	1	0	1	1	1	n_4	n_3	n_2	n_1	n_0	PC ← (PC)+n+1, if Z=0 and G=0
	BLT −n	0	0	0	1	1	1	1	1	1	n_4	n_3	n_2	n_1	n_0	PC ← (PC)−n, if Z=0 and G=0
Input/Output	SWITCH AP	1	1	0	1	0	P	0	0	0	1	A				AP ← INPUT PORT (S1 ~ S4)
	KSWITCH AP	1	1	0	1	0	P	0	0	1	0	A				AP ← INPUT PORT (K1 ~ K4)

DESCRIPTION OF INSTRUCTIONS (CONT.)

	Mnemonic	Instruction Code 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Operation
Input/Output	MATRIX AP	1 1 0 1 1 P 0 0 1 0 A	OUTPUT PORT (M1~M4) ← (AP)
	MATRIX Mn	0 0 0 1 0 0 0 0 1 0 M_4 M_3 M_2 M_1	OUTPUT PORT (M1~M4) ← Mn (n=1, 2, 3, 4)
	XTCP ON/OFF	0 0 0 1 0 0 1 0 0 0 0 0 b_1 b_0	XTOUT ON/OFF
	PITCH N	0 0 0 1 0 0 1 1 0 0 N	PIT reg. ← N
	MSA adrs	1 0 1 0 1 a_8 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	Stack ← (PC), PC ← adrs, Melody start
	MSO	0 0 0 1 0 0 1 0 1 0 0 0 0 0	Melody stop
	LAMP ON/OFF	0 0 0 1 0 0 0 0 0 1 0 0 b_1 b_0	LD ON/OFF
Display	DSP digit, AP	0 0 1 0 0 P digit A	Digit (Low part) ← (AP), (ACC)
	DSPH digit, AP	0 0 1 0 1 P digit A	Digit (High part) ← (AP), (ACC)
	FORMAT AP	1 1 0 1 1 P 0 0 1 1 A	FMT reg. ← (AP)
	FORMAT N	0 0 0 1 0 0 0 0 1 1 N	FMT reg. ← N
	DSPF digit, AP	0 0 1 1 0 P digit A	Digit (Low part) ← (AP) via table
	DSPFH digit, AP	0 0 1 1 1 P digit A	Digit (High part) ← (AP) via table
CPU Control & Others	HALT	0 0 0 1 0 0 0 0 0 0 0 0 0 0	Halt
	INTENAB 32/16/2	0 0 0 1 0 0 1 0 1 1 1 0 0 0	Enable timer interrupt
		0 0 0 1 0 0 0 1 0 0 b_3 0 b_1 0	
	INTDSAB 32/16/2	0 0 0 1 0 0 1 0 1 1 0 1 0 0	Disable timer interrupt
		0 0 0 1 0 0 0 1 0 0 0 b_2 0 b_0	
	ACTIVATE	0 0 0 1 0 0 1 0 0 1 0 0 0 0	Activate main routine
	INTMODE AP	1 1 0 1 0 P 0 1 0 0 A	AP ← Interrupt mode
	KENAB	0 0 0 1 0 0 0 1 1 1 1 0 0 0	Enable INPUT PORT (K1~K4)
	KDSAB	0 0 0 1 0 0 0 1 1 1 0 1 0 0	Disable INPUT PORT (K1~K4)
	STATUS AP	1 1 0 1 0 P 1 0 1 0 A	AP ← Status
	PAGE A0	1 1 0 1 1 0 0 1 0 1 A	Preg ← (A0)
	PAGE N	0 0 0 1 0 0 0 1 0 1 N	Preg ← N
	ADRS AP	1 1 0 1 1 P 0 1 1 0 A	Areg ← (AP)
	ADRS N	0 0 0 1 0 0 0 1 1 0 N	Areg ← N
	RATE AP	1 1 0 1 0 P 1 0 0 1 A	AP ← DIVIDER (8 Hz~1 Hz)
	RSTRATE	0 0 0 1 0 0 1 0 0 0 1 0 0 0	DIVIDER (8 Hz~1 Hz) ← 0
	FLAGIN AP	1 1 0 1 0 P 1 1 1 0 A	AP ← S1 on flag, S2 on flag
	S1RATE AP	1 1 0 1 0 P 1 1 0 0 A	AP ←S1reg.
	S2RATE AP	1 1 0 1 0 P 1 1 0 1 A	AP ←S2reg.
	BACKUP ON/OFF	0 0 0 1 0 0 0 0 0 1 b_3 b_2 0 0	Backup ON/OFF
	NOP	0 0 0 0 0 0 0 0 0 0 0 0 0 0	No operation

MSM6351

CMOS 4BIT HIGH PERFORMANCE AND VERY LOW POWER SINGLE CHIP MICROCONTROLLER WITH LCD DRIVER

GENERAL DESCRIPTION

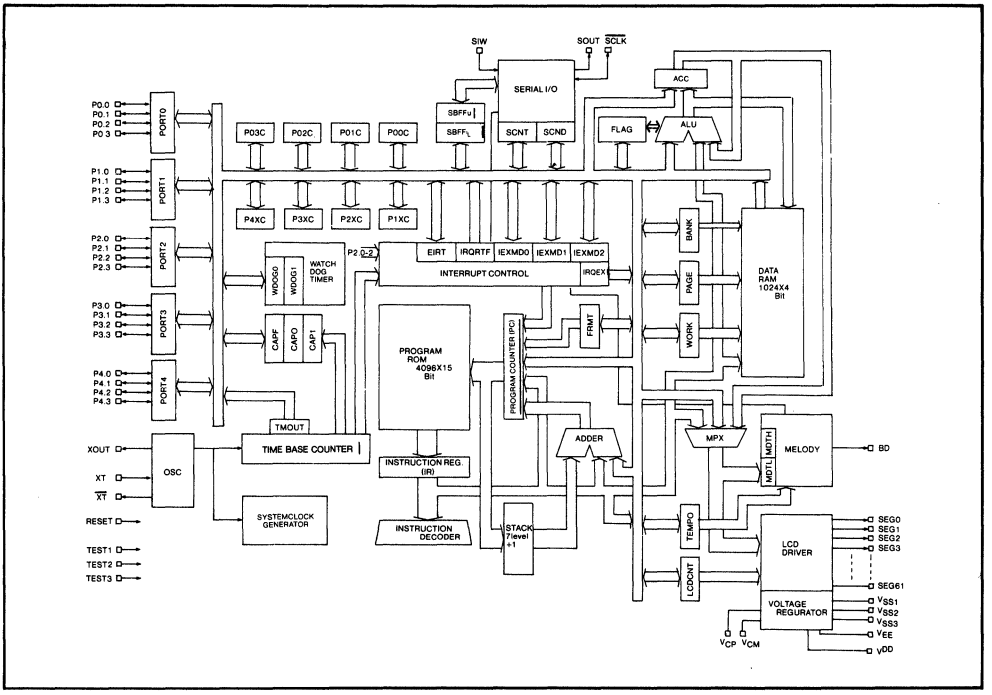
OKI's MSM6351 is a low-power, high-performance single-chip microcontroller employing silicon gate CMOS technology. Integrated onto a single chip are 4-bit ALU, 61K bits of mask programmable ROM, 4096 bits of RAM, 20 bits of I/O port, serial I/O port, time-base counter, LCD driver, 3 interrupts, crystal oscillator and voltage tripler.

The MSM6351 is widely used in electronic products requiring low power consumption, a large number of LCD drivers and a large size of memory.

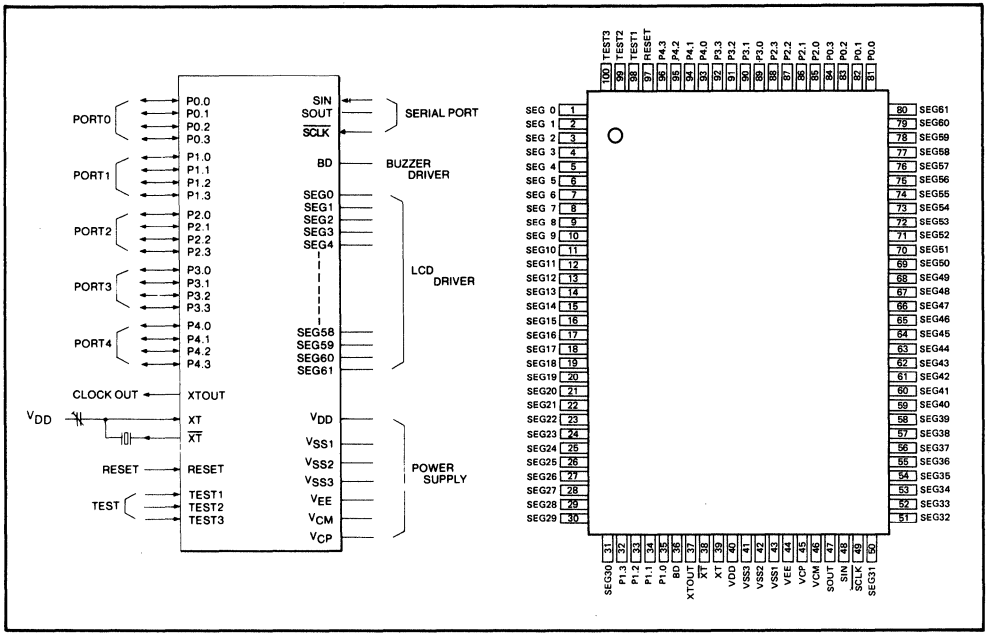
FEATURES

- Low Power Consumption 3 μA Typical
- 4096 × 15 Internal ROM
- 1024 × 4 Internal RAM
- 20 Input/Output Ports
- 62 LCD Drivers
  1/3 Duty, 1/3 Bias or 1/4 Duty 1/3 Bias
  (Selectable by software)
- Serial I/O Port
  8 bits or 5 bits data frame mode
  Asynchronous
  receiver/transmitter mode
  Internal or external clock mode
- 15 stages Time Base Counter
- Watch Dog Timer
- Capture function by external trigger signal
- 3 Interrupt Sources
  Real time interrupt
  External interrupt
  Serial I/O port interrupt
- Melody Circuit
- 65 Instructions
- Sub-routine Nesting: 7 levels
- 32.768 kHz Crystal Oscilator
- Machine Cycle: 61.0 μsec.
- Power Supply: 1.5V or 3.0V (selectable by mask option)
- 100 pad die Silicon gate CMOS Process
- Package:
  100 pin plastic QFP (QFP100-P-1420-K)

FUNCTIONAL BLOCK DIAGRAM



LOGIC SYMBOL



PIN DESCRIPTION

Designation	Function			
P0.0 P0.1 P0.2 P0.3	PORT 0	• 4-bit I/O port 0 The input (*)/output, the existence (*)/ absence of pull-down resistance, and the HALT function release enable/disable condition can be selected for each bit.		Capture trigger signal
P1.0 ~ P1.3	PORT 1	• 4-bit I/O port 1	The input (*)/output, the existence (*)/absence of pull-down resistance, and the HALT function release enable/disable condition can be selected for each bit.	
P2.0 P2.1 P2.2 P2.3	PORT 2	• 4-bit I/O port 2		External interrupt signal
P3.0 ~ P3.3	PORT 3	• 4-bit I/O port 3		
P4.0 ~ P4.3	PORT 4	• 4-bit I/O port 4		
XTOUT	• Oscillator clock output The oscillator clock is output when XTF (forth bit of port P00C) is set to "1".			
$\overline{XT}$ XT	• Oscillator connection terminal			
RESET	• Reset input Input with a pull-down resistance, the system is reset when "1" is input.			
TEST 1 TEST 2 TEST 3	• Test input Input with a pull-down resistance.			
SIN	• Serial port data input			
SOUT	• Serial port data output the circuit is set at high impedance level when "1" is set to HZOUT (the forth bit of port P4XC)			
$\overline{SCLK}$	• Serial port clock input and output the input and output (*) are switched from each other the serial port control register SCNT. In the output mode, the serial clock frequency can be selected from the demultiplied signal (1/1, 1/2 or 1/4 of the system clock)			
BD	• Melody output (buzzer drive output)			
SEG0 ~ SEG61	• LCD drive output LCD drive output with 1/3 bias and 1/3 duty, or 1/3 bias and 1/4 duty. The duty can be switched by LCD control register LCDCT. A maximum of 177 segments can be displayed with the 1/3 duty and 232 with the 1/4 duty.			

* means system reset conditions.

Designation	Function
V_{DD}	• 0V power supply terminal
V_{SS1}	• —1.5V power supply terminal (for the 1.5V specification)
V_{SS2}	• —3.0V power supply terminal (for the 3.0V specification)
V_{SS3}	• —4.5V power supply terminal
V_{EE}	• Internal logic power supply terminal
V_{CM}	• Internal voltage converter capacitor connecting terminal
V_{CP}	

FUNCTIONAL DESCRIPTION

A block diagram of MSM6351 is given on page 128. Each block of logic will be briefly discussed. For further information, please refer to MSM6351/53 user's manual.

Programmable ROM

The programmable ROM has a capacity of 4096 words, each of which is 15 bits long. It is provided with the address space of 000H to FFFH.

In the MSM6351, the programmable ROM is not only used for programming but also used to save the following items:

1) LCD indicator segment conversion table
2) Melody tone data

The program instructions are all made up of one word, thus the ROM can save up to 4096 instructions in it.

Data RAM

The data RAM has a capacity of 1024 words, each consisting of four bits. It is provided with the space for address between 000H and 3FFH. Data is organized in 4 bit nibble.

Page Register (PAGE)

The page register specifies one of 16 pages in each bank of the data RAM.

Bank Register (BANK)

The bank register specifies one of four banks in the data RAM. It is used together with the page register.

Work Specification Register

The work specification register specifies one of 16 pages in bank 0 of the data RAM as the work registers.

Operational Section

The operational section consists of the ALU, accumulator (ACC), and conditional flags C, Z and G.

This operational section performs four-bit computation of the contents of the data RAM with the contents of ACC or the immediate data in the instruction words.

This computation is mainly performed with the data RAM which functions as a register. The resultant data of the computation are input to the data RAM or to ACC (for operation other than comparison).

Program Counter (PC)

The program counter (PC) generates addresses for the program ROM.

The addresses for program ROM are changed according to the instructions executed. These addresses are incremented by one each time the instruction is executed.

When an interrupt is generated, the current address is stored in the STACK. The address is set to 400H, 401H or 402H depending on the type of the interrupt.

These addresses are set to the start addresses of each interrupt routine.

In the MSM6351, the PC also gives LCD indicator "segment conversion table" or melody "tone data" addresses to the program ROM.

The output data of the program ROM whose address is specified by the PC is fetched into the instruction register (IR). If the output data is an instruction, it is decoded by the instruction decoder. Then, control signals to each section are generated.

Ports

The MSM6351 handle the I/O ports, flags and registers collectively as ports. Therefore, each of the I/O ports, flags and registers are selected by specifying their own addresses.

All of these ports are accessed by the INP and OUT instructions.

PORT NAMES, ADDRESSES AND THEIR CONTENTS

Port name	Address	Bit 3	Bit 2	Bit 1	Bit 0	Access mode (*)
PORT0	00	P0.3	P0.2	P0.1	P0.0	R/W
PORT1	01	P1.3	P1.2	P1.1	P1.0	R/W
PORT2	02	P2.3	P2.2	P2.1	P2.0	R/W
PORT3	03	P3.3	P3.2	P3.1	P3.0	R/W
PORT4	04	P4.3	P4.2	P4.1	P4.0	R/W
P00C	05	XTF	HRE00	HZ00	DIR00	R/W
P01C	06	BUF	HRE01	HZ01	DIR01	R/W
P02C	07	–	HRE02	HZ02	DIR02	R/W
P03C	08	–	HRE03	HZ03	DIR03	R/W
P1XC	09	–	HRE1X	HZ1X	DIR1X	R/W
P2XC	0A	HZSOUT	HRE2X	HZ2X	DIR2X	R/W
P3XC	0B	EISIO	HRE3X	HZ3X	DIR3X	R/W
P4XC	0C	5/8	HRE4X	HZ4X	DIR4X	R/W
SBF	0D	(L) d_3	d_2	d_1	d_0	R/W
		(u) d_7	d_6	d_5	d_4	
SCNT	0E	CLKSL1	CLKSL0	MODE	LSB/$\overline{\text{MSB}}$	R/W
SCND	0F	SIOEND	STPErr	ENRC	ENTR	Bit: 3 and 2; R 1 and 0; R/W
IRQEX	10	–	IRQP22	IRQP21	IRQP20	R
EIRT	11	EI256Hz	EI32Hz	EI16Hz	EI1Hz	R/W
IRQRT	12	IRQ256Hz	IRQ32Hz	IRQ16Hz	IRQ1Hz	R
IEXM0	13	–	EIP20	L/E0	P/N0	R/W
IEXM1	14	–	EIP21	L/E1	P/N1	R/W
IEXM2	15	–	EIP22	L/E2	P/N2	R/W
TMOUT	16	15th	14th	13th	12th	R
CAPTR	17	128Hz / 32Hz	256Hz / 64Hz	512Hz / 128Hz	1KHz / 256Hz	R
		–	CAPMD	CAP1F	CAP0F	Bit 2: R/W Other bits: R
FLAG	18	MSTART	G	Z	C	Bit 3: R Other bits: R/W
WDOG	19	W_{03}	W_{02}	W_{01}	W_{00}	R/W
		W_{13}	W_{12}	W_{11}	W_{10}	
FRMT	1A	F_3	F_2	F_1	F_0	R/W
LCDCT	1B	AIION	Duty 3/4	FLM1	FLM0	R/W

PORT NAMES, ADDRESSES AND THEIR CONTENTS (Continued)

Port name	Address	Bit 3	Bit 2	Bit 1	Bit 0	Access mode (*)
TEMPO	1C	P_3	P_2	P_1	P_0	R/W
BANK	1D	b_3	b_2	b_1	b_0	R/W
PAGE	1E	P_3	P_2	P_1	P_0	R/W
WORK	1F	w_3	w_2	w_1	w_0	R/W

Note: • In access mode (*), R denotes "readable" bits and W "writable" bits.
• Bits marked with — denote bits which are not present.

I/O Ports

The MSM6351 is provided with five ports; PORT0 to PORT4. Each port consists of four bits.

Ports are controlled by the I/O port control register.

Each port is accessed by the OUT and INP instruction.

Time Base Counter

The MSM6351 has their built-in time base counters consisting of a 15-stage binary counter. System base clock is input from the time base counter.

Capture Circuit

The MSM6351 is provided with a capture function that stores the 1KHz to 128Hz outputs at stages 5 to 8 of the time base counter or the 256Hz to 32Hz outputs at stages 7 to 10 when P0.0 or P0.1 of I/O port 0 is set at the "H" level.

Watchdog Timer

The MSM6351 has their built-in watchdog timer to prevent program malfunction. The time-out may be set with two types of setting time: 250ms and 2s.

Serial Port (SERIAL I/O)

The MSM6351 has its built-in serial port. It is used for synchronous and asynchronous data communications. A data length of five or eight bits can be selected. Either internal or external clock can be selected as the driving clock. At the end of data transfer, a serial port inerrupt can be generated.

Melody Output Circuit (MELODY)

The melody output circuit automatically outputs melody or buzzer sound.

The melody circuit initiates its operation by the MSA instruction. Automatically fetching the musical note data defined in the program ROM, the MSA instruction outputs the melody from buzzer driving output terminal BD.

Liquid Crystal Display Circuit (LCD DRIVER)

The MSM6351 has its built-in liquid crystal display circuit that can drive the liquid crystal (LCD).

The liquid crystal display consists of the display data register for holding the data to indicate and the display driver. After data is written in the display data register with an display instruction, the display driver automatically fetches data from the display data register to output the driving waveform.

Interruption Controller

There are four types of interruptions as follows:

1) Real-time interruption — Interruption with the time base counter output
2) External interruption — Interruption from PORT2
3) Serial port interruption — Interruption by terminating serial port data transfer
4) Melody interruption — Interruption by requesting melody data

ABSOLUTE MAXIMUM RATING (Target Specification)

V_{DD} = 0V (V_{SS2} = Battery Voltage)

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	Ta = 25°C	−6.0 ~ +0.3	V
Input Voltage	V_{IN}		V_{SS2}−0.3 ~ +0.3	V
Output Voltage*1	V_{01}		V_{SS2}-0.3 ~ +0.3	V
Output Voltage*2	V_{02}		−6.0 ~ +0.3	V
Storage Temperature	T_{STG}		−55 ~ +125	°C

*1 Normal Output
*2 LCD Driver

OPERATING CONDITION (Target Specification)

Parameter	Symbol	Limits	Units
Operating Voltage	$-V_{SS2}$	2.6 ~ 3.5	V
Operating Temperature	Topr	−20 to 70	°C

DC CHARACTERISTICS 3V Li Battery

V_{DD} = 0V, V_{SS1} = −1.5V, V_{SS2} = −3.0V, V_{SS3} = −4.5V, f = 32,768Hz, Cx = 35kΩ Ta = 25°C

Parameter	Symbol	Conditions	Rating			Unit	Terminal Applied
			Min.	Typ.	Max.		
Current Consumption	I_{DD}	*1, *3	–	3.0	–	μA	
Voltage for Oscillation Start	$-V_{OSC}$	Within 2 sec.	–	–	2.4	V	
Output Current 1 (Common, Segment)	$-I_{OH1}$	V_O = −0.2V	4	–	–	μA	SEG0 ~ SEG61
	$\|I_{OMH1}\|$	$V_{OM} = V_{SS1} \pm 0.2V$	4	–	–		
	$\|I_{OML1}\|$	$V_{OM} = V_{SS1} \pm 0.2V$	4	–	–		
	I_{OL1}	V_O = −4.3V	4	–	–		
Output Current 2	$-I_{OH2}$	V_O = −0.5V	500	–	–	μA	PORT0~PORT4 *2 S_{OUT}, $S\overline{CLK}$ X_{TOUT}
	I_{OL2}	V_O = −2.5V	500	–	–		
Output Current 3	$-I_{OH3}$	V_O = −0.5V V_{SS2} = 3.0V	7	–	–	μA	BD
	I_{OL3}	V_O = −2.5V V_{SS2} = 3.0V	20	–	–		
Input Current 1	$-I_{IH1}$	V_I = 0V I/O input, with pull down	150	300	600	μA	PORT0~PORT4
Input Leak Current 2	$\|I_{I2}\|$	V_I = 0V, −3V I/O input, without pull down	–	–	1	μA	PORT0~PORT4 S_{IN}, $S\overline{CLK}$, S_{OUT}
Input Current 3	$-I_{IH3}$	V_I = 0V with pull down	–	25	–	μA	RESET
Input Voltage	$-V_{IH}$		–	–	0.5	V	All input terminals
	$-V_{IH}$		2.5	–	–		

*1. When 3V battery with halver is used (BUF = '0').

*2. PORT0 = P0.0 ~ P0.3, PORT1 = P1.0 ~ P1.3, PORT2 = P2.0 ~ P2.3, PORT 3 = P3.0 ~ P3.3, PORT4 = P4.0 ~ P4.3

*3. This value changes depending on the soft duty (HALT to HALT)

INSTRUCTION LIST

	Mnemonic	Instruction code														Description	Machine cycle
		14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Operation instruction	ADD ACC, REG1	0	0	0	0	0	0	P	0	1	0	0	r_3	r_2	r_1	r_0	(rP), (ACC), (Z), (C) ← (rP) + (ACC)	1
	ADD #i, REG1	0	0	1	1	0	0	P	i_3	i_2	i_1	i_0	r_3	r_2	r_1	r_0	(rP), (ACC), (Z), (C) ← (rP) + i	1
	ADC REG1	0	0	0	0	0	0	P	0	1	0	1	r_3	r_2	r_1	r_0	(rP), (ACC), (Z), (C) ← decimal adj [(rP) + (ACC) + (C)]	1
	ADCN REG1	0	1	1	0	0	0	P	N_3	N_2	N_1	N_0	r_3	r_2	r_1	r_0	(rP), (ACC), (Z), (C) ← N adjust [(rP) + (C)]	1
	SUB ACC, REG1	0	0	0	0	0	1	P	0	1	0	0	r_3	r_2	r_1	r_0	(rP), (ACC), (Z), (C) ← (rP) − (ACC)	1
	SUB #i, REG1	0	0	1	1	0	1	P	i_3	i_2	i_1	i_0	r_3	r_2	r_1	r_0	(rP), (ACC), (Z), (C) ← (rP) − i	1
	SBC REG1	0	0	0	0	0	1	P	0	1	0	1	r_3	r_2	r_1	r_0	(rP), (ACC), (Z), (C) ← decimal adj [(rP) − (ACC) − (C)]	1
	SBCN REG1	0	1	1	0	0	1	P	N_3	N_2	N_1	N_0	r_3	r_2	r_1	r_0	(rP), (ACC), (Z), (C) ← N adjust [(rP) − (C)]	1
	CMP ACC, REG1	0	0	0	0	0	1	P	1	1	1	0	r_3	r_2	r_1	r_0	(Z), (G) ← (rP) − (ACC)	1
	CMP #i, REG1	0	0	1	0	1	1	P	i_3	i_2	i_1	i_0	r_3	r_2	r_1	r_0	(Z), (G) ← (rP) − i	1
	INC REG1	0	0	0	0	0	0	P	0	0	0	1	r_3	r_2	r_1	r_0	(rP), (ACC), (Z), (C) ← (rP) + 1	1
	INCD REG2	1	0	0	0	b_1	b_0	0	P_3	P_2	P_1	P_0	r_3	r_2	r_1	r_0	(rPb), (ACC), (Z), (C) ← (rPb) + 1	1
	DEC REG1	0	0	0	0	0	1	P	0	0	0	1	r_3	r_2	r_1	r_0	(rP), (ACC), (Z), (C) ← (rP) − 1	1
	DECD REG2	1	0	0	1	b_1	b_0	0	P_3	P_2	P_1	P_0	r_3	r_2	r_1	r_0	(rPb), (ACC), (Z), (C) ← (rPb) − 1	1
Bit operation instruction	BIT ACC, REG1	0	0	0	0	0	0	P	1	1	1	0	r_3	r_2	r_1	r_0	$(Z) \leftarrow \overline{(rP_3)} \wedge (ACC_3) \vee \overline{(rP_2)} \wedge (ACC_2) \vee \overline{(rP_1)} \wedge (ACC_1) \vee \overline{(rP_0)} \wedge (ACC_0)$	1
	BIT #i, REG1	0	0	1	0	1	0	P	i_3	i_2	i_1	i_0	r_3	r_2	r_1	r_0	$(Z) \leftarrow \overline{(rP_3)} \wedge i_3 \vee \overline{(rP_2)} \wedge i_2 \vee \overline{(rP_1)} \wedge i_1 \vee \overline{(rP_0)} \wedge i_0$	1
	BIS ACC, REG1	0	0	0	0	0	0	P	0	1	1	0	r_3	r_2	r_1	r_0	(rP), (ACC), (Z) ← (rP) ∨ (ACC)	1
	BIS #i, REG1	0	0	1	0	0	0	P	i_3	i_2	i_1	i_0	r_3	r_2	r_1	r_0	(rP), (ACC), (Z) ← (rP) ∨ i	1
	BIC ACC, REG1	0	0	0	0	0	1	P	0	1	1	0	r_3	r_2	r_1	r_0	$(rP), (ACC), (Z) \leftarrow (rP) \wedge \overline{(ACC)}$	1
	BIC #1, REG1	0	0	1	0	0	1	P	i_3	i_2	i_1	i_0	r_3	r_2	r_1	r_0	$(rP), (ACC), (Z) \leftarrow (rP) \wedge \overline{i}$	1
	XOR ACC, REG1	0	0	0	0	0	0	P	0	1	1	1	r_3	r_2	r_1	r_0	$(rP), (ACC), (Z) \leftarrow (rP) \forall (ACC)$	1
	XOR #i, REG1	0	0	1	1	1	1	P	i_3	i_2	i_1	i_0	r_3	r_2	r_1	r_0	$(rP), (ACC), (Z) \leftarrow (rP) \forall i$	1

INSTRUCTION LIST (Continued)

	Mnemonic	Instruction code															Description	Machine cycle
		14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Rotate instruction	ROR REG1	0	0	0	0	0	0	P	0	0	1	0	r_3	r_2	r_1	r_0	(Z), (ACC) ← [$\overrightarrow{\hookrightarrow (C) \rightarrow (rP) \rightarrow}$]	1
	ROL REG1	0	0	0	0	0	1	P	0	0	1	0	r_3	r_2	r_1	r_0	(Z), (ACC) ← [$\overleftarrow{\leftarrow (C) \leftarrow (rP) \hookleftarrow}$]	1
	ASR REG1	0	0	0	0	0	0	P	0	0	1	1	r_3	r_2	r_1	r_0	(Z), (ACC) ← [O → (rP) → (C)]	1
	ASL REG1	0	0	0	0	0	1	P	0	0	1	1	r_3	r_2	r_1	r_0	(Z), (ACC) ← [(C) ← (rP) ← O]	1
Flag operation instruction	CLG	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	(G) ← O	1
	CLC	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	(C) ← O	1
	CLZ	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	(Z) ← O	1
	CLA	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	(Z), (C), (G) ← O	1
	SEG	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	(G) ← 1	1
	SEC	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	(C) ← 1	1
	SEZ	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	(Z) ← 1	1
	SEA	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0	(Z), (C), (G) ← 1	1
Date transfer instruction	MOV ACC, REG1	0	0	0	0	0	0	P	1	1	1	1	r_3	r_2	r_1	r_0	(rP) ← (ACC)	1
	MOVD ACC, REG2	1	0	1	0	b_1	b_0	0	P_3	P_2	P_1	P_0	r_3	r_2	r_1	r_0	(rPb) ← (ACC)	1
	MOV #i, REG1	0	0	1	1	1	0	P	i_3	i_2	i_1	i_0	r_3	r_2	r_1	r_0	(rP), (ACC), (Z) ← i	1
	MOV REG1, ACC	0	0	0	0	0	1	P	1	1	1	1	r_3	r_2	r_1	r_0	(ACC), (Z) ← (rP)	1
	MOVD REG2, ACC	1	0	1	1	b_1	b_0	0	P_3	P_2	P_1	P_0	r_3	r_2	r_1	r_0	(ACC), (Z) ← (rPb)	1
	EXG REG1	0	0	0	0	0	1	P	0	0	0	0	r_3	r_2	r_1	r_0	(rP) ↔ (ACC)	1
	EXGD REG2	0	1	1	1	b_1	b_0	0	P_3	P_2	P_1	P_0	r_3	r_2	r_1	r_0	(rPb) ↔ (ACC)	1
Subroutine instruction	CALL adrs	1	1	1	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	(STACK) ← (PC), (PC) ← $a_{11} \sim a_0$, (SP) ← (SP) + 1	1
	RET	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	(PC) ← (STACK) + 1 (SP) ← (SP) − 1	1
	RTI	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	(PC) ← (STACK) + 1 (SP) ← (SP) − 1 (at INT routine)	1
Jump instruction	JMP adrs	1	1	0	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	(PC) ← $a_{11} \sim a_0$	1
	JMP @ REG1	0	0	0	0	0	0	P	1	1	0	1	r_3	r_2	r_1	r_0	(PC) ← (PC) + (rP) + 1	1
	JMPIO @ REG1	0	0	0	0	0	1	P	1	1	0	1	r_3	r_2	r_1	r_0	(PC) ← (PC) + 7 ∧ (rP) + 1	1

INSTRUCTION LIST (Continued)

	Mnemonic	Instruction code															Description	Machine cycle
		14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Jump instruction	BGT n	0	0	0	0	1	1	P	0	0	1	n_4	n_3	n_2	n_1	n_0	if (G) = 1 then (PC) ← (PC) + N else (PC) ← (PC) + 1	1
	BLE n	0	0	0	0	1	1	P	1	0	1	n_4	n_3	n_2	n_1	n_0	if (G) = 0 then (PC) ← (PC) + N else (PC) ← (PC) + 1	1
	BCS n	0	0	0	0	1	1	P	0	0	0	n_4	n_3	n_2	n_1	n_0	if (C) = 1 then (PC) ← (PC) + N else (PC) ← (PC) + 1	1
	BCC n	0	0	0	0	1	1	P	1	0	0	n_4	n_3	n_2	n_1	n_0	if (C) = 0 then (PC) ← (PC) + N else (PC) ← (PC) + 1	1
	BEQ n (BZE n)	0	0	0	0	1	1	P	0	1	0	n_4	n_3	n_2	n_1	n_0	if (Z) = 1 then (PC) ← (PC) + N else (PC) ← (PC) + 1	1
	BNE n (BNZ n)	0	0	0	0	1	1	P	1	1	0	n_4	n_3	n_2	n_1	n_0	if (Z) = 0 then (PC) ← (PC) + N else (PC) ← (PC) + 1	1
	BGE n	0	0	0	0	1	1	P	0	1	1	n_4	n_3	n_2	n_1	n_0	if [(G) = 1 or (Z) = 1] then (PC) ← (PC) + N else (PC) → (PC) + 1	1
	BLT n	0	0	0	0	1	1	P	1	1	1	n_4	n_3	n_2	n_1	n_0	if [(G) = 0 and (Z) = 0] then (PC) ← (PC) + N else (PC) ← (PC) + 1	1
Melody start	MSA adrs	0	0	0	0	1	0	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Specifies the first address of note data. ($E00_H$ ~ FFF_H)	2
Display instruction	DSP dig, REG1	0	0	0	1	0	0	P	dig_3	dig_2	dig_1	dig_0	r_3	r_2	r_1	r_0	digit (Low Part) ← (rP), (ACC)	1
	DSPH dig, REG1	0	0	0	1	0	1	P	dig_3	dig_2	dig_1	dig_0	r_3	r_2	r_1	r_0	digit (High Part) ← (rP), (ACC)	1
	DSPF dig, REG1	0	0	0	1	1	0	P	dig_3	dig_2	dig_1	dig_0	r_3	r_2	r_1	r_0	digit (Low Part) ← (rP) via Table	2
	DSPFH dig, REG1	0	0	0	1	1	1	P	dig_3	dig_2	dig_1	dig_0	r_3	r_2	r_1	r_0	digit (High Part) ← (rP) via Table	2
Input/output instruction	OUT REG1, PORT	0	1	0	1	0	y_4	P	y_3	y_2	y_1	y_0	r_3	r_2	r_1	r_0	(PORT y) ← (rP)	1
	OUT #i, PORT	0	1	0	1	1	y_4	0	y_3	y_2	y_1	y_0	i_3	i_2	i_1	i_0	(PORT y) ← i	1
	INP PORT, REG1	0	1	0	0	0	y_4	P	y_3	y_2	y_1	y_0	r_3	r_2	r_1	r_0	(rP), (ACC) ← (PORT y)	1
CPU control	NOP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	No Operation	1
	HALT	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	Halt CPU	1

When P = 0 in bit 8, N = n + 1; when P = 1, N = −n

MSM6352

CMOS 4BIT SINGLE CHIP LOW POWER MICROCONTROLLER FOR TELEPHONE

GENERAL DESCRIPTION

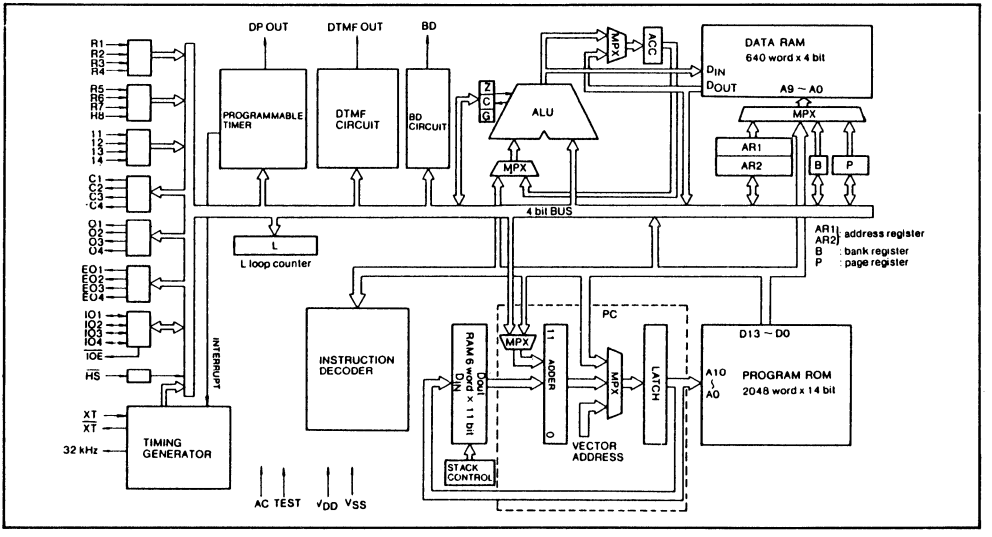
The OKI MSM6352 is a low-power, high-performance single-chip 4-bit microcontroller employing complementary metal oxide semiconductor technology, especially designed for use in sophisticated telephone sets. Integrated onto a single chip are a 4-bit ALU, 28K bits of mask programmable ROM, 2560 bits of data RAM, programmable timer, oscillator, 12-bits of input port, 12-bits of output port and 4-bits of input/output port. In addition to these units, a DTMF generator is provided.

With the MSM6352, sophisticated telephone sets become feasible through a single chip instead

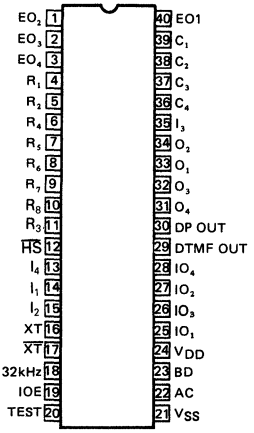
FEATURES

- Low Power Consumption
- 2048 × 14 Internal ROM
- 640 × 4 Internal RAM
- 3 × 4 Input Port
- 3 × 4 Output Port
- 1 × 4 Input/Output Port
- DTMF Generator (Single Tone Mode or Dual Tone Mode)
- Buzzer Sound Output
- 4 Bits Programmable Timer Applicable for Output of Dial Pulse
- Watch Dog Timer
- On Hook Dialing and Off Hook Dialing Function
- Interrupt
  Programmable Timer-Interrupt
  Real Time Interrupt
- 5 Level Stack
- Power Down Mode
- 52 Instruction Set
- Instructions Useful for Data Management (Data Search and Block Data Transfer)
- 2.0 to 5.5V (2.2 to 5.5V at TONE MODE) Operating Voltage
- 3.58 MHz Oscillator
- 17.9 μs Instruction Cycle
- −20 to 75°C Operating Temperature
- 28 Pin DIP, 40 Pin DIP or 44 Pin FLAT
- Software Compatibility with MSM6052
- Package:
  28 pin plastic DIP (DIP28-P-600)
  40 pin plastic DIP (DIP40-P-600)
  42 pin plastic shrink DIP (SDIP42-P-600)
  44 pin plastic QFP (QFP44-P-910-K)
  44 pin plastic QFP (QFP44-P-910-VK)

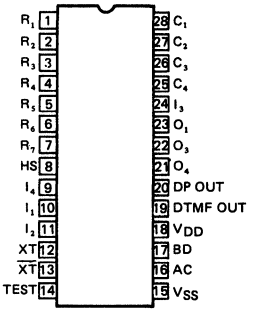
FUNCTIONAL BLOCK DIAGRAM



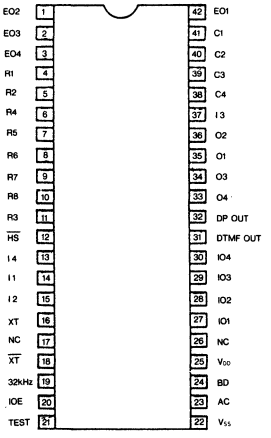
PIN CONFIGURATION



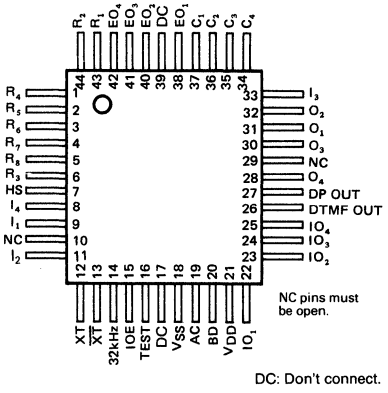
(a) 40 Lead Plastic DIP



(b) 28 Lead Plastic DIP

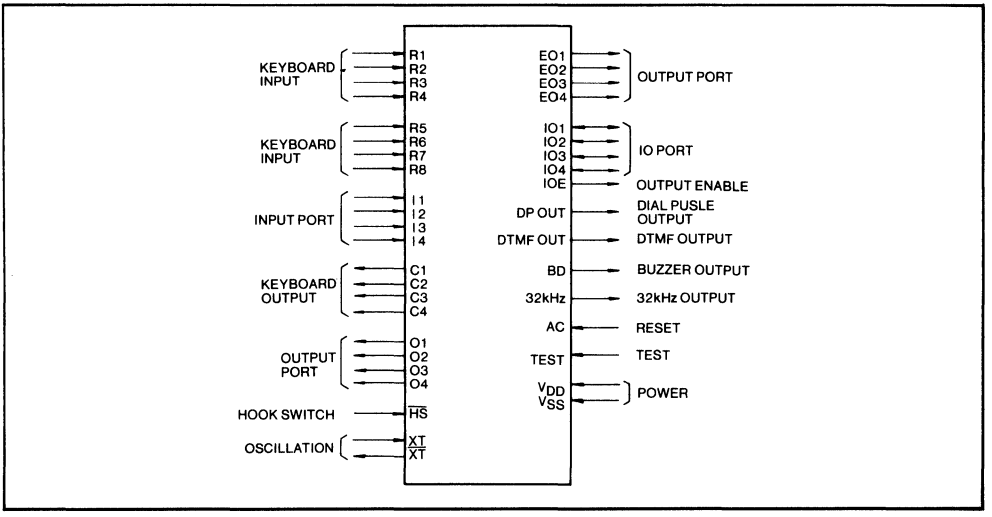


(c) 42 PIN PLASTIC SHRINK DIP



(d) 44 Lead Plastic Flat Package
(NC pin must not be connected to any signal.)

LOGIC SYMBOL



PIN DESCRIPTION

Designation	Function
V_{DD}	Power source
V_{SS}	Circuit ground potential
AC	Terminal to clear internal logic, pulled down to V_{SS}. After power is turned on, the MSM6352 must be reset by this terminal.
TEST	Terminal to test internal logic, pulled down to V_{SS}. This terminal must be open in normal operation.
XT, $\overline{XT}$	Input and output terminals of oscillator inverter. 3.58 MHz ceramic resonator and capacitors is connected to these terminals.
$\overline{HS}$	Input terminal connected to the hook switch, pulled up tp V_{DD}.
DP OUT	Output terminal of dial pulse. Dial pulse rate (10 pps or 20 pps) and Make Break ratio (40% or 33 %) can be selected by software.
DTMF OUT	Output terminal of DTMF signal
BD	Output terminal of buzzer sound
32 kHz	Output terminal of 32 kHz clock
$R_1 \sim R_4$ $R_5 \sim R_8$	Input port pulled down to V_{SS}.
$I_1 \sim I_4$	Input port having clocked pull-down resistor to V_{SS}. Only when this port is accessed, pull-down resistors are connected to this port.
$C_1 \sim C_4$ $O_1 \sim O_4$	Output port
$IO_1 \sim IO_4$	Tri-state bidirectional port
IOE	Output terminal When $IO_1 \sim IO_4$ is accessed, input completion signal (when read) or load signal (when written) is output from IOE terminal.

FUNCTIONAL DESCRIPTION

A block diagram of the MSM6352 is given on page 149. Each block of logic will be briefly discussed. For more information, please refer to the MSM6352 user's manual.

Program ROM

The MSM6352 will address up to 1 K words of internal mask programmable ROM. Each word consists of 14-bits and all instructions are one word. The instructions are routed to a programmed logic array which generales the signals necessary for control of logic.

Data RAM

Data is organized in 4-bit nibbles. Internal data RAM consists of 640 nibbles.

All locations are addressed by 10-bit address registers (AR_1, AR_2), 2-bit bank register (B), 4-bit page register (P) or a part of the instruction's operand.

Program Counter (PC)

The PC is an 11-bit counter to specify the ROM's address. The PC is normally incremented by one by every execution of the instruction, and then specifies the next instruction to be executed. However, Jump, Conditional branch, and Subroutine instructions are exceptions.

When the JMP adrs or CALL adrs instruction is executed, all of the PC contents are rewritten, so jump can be done to any address of the ROM.

Bank Register (B)

The bank register is a 2 bits register which specifies the bank of the RAM. Read/Write operation is performed by the Input/Output instruction.

Page Register (P)

The page register is a 4 bits register which specifies the page of the RAM. Read/Write operation is performed by the Input/Output instruction.

Address Register 1 (AR_1)

The address register AR_1 is a 10 bits register which specifies the RAM's address. This register is used by the RDAR instruction or the MVAR instruction. Read/Write operation is performed by the Input/Output instruction.

Address Register 2 (AR_2)

The address register AR_2 is a 10 bits register which specifies the RAM's address. This register is used by the MVAR instruction. Read/Write operation is performed by the Input/Output instruction.

Loop Counter (L)

The loop counter is a 10 bits down counter which specifies a number of words of the data to be searched or to be moved by the RDAR instruction or the MVAR instruction. Its contents can be rewritten by the output instruction.

ALU, Conditional Flag, ACC

(a) ALU

The ALU performs 4-bits parallel operation of the RAM contents and ACC contents, or the RAM contents and an immediate data. The arithmetic, logic, comparison and rotate operations can be done.

(b) Conditional flag

The zero flag (Z), carry flag (C) and greater flag (G) are provided. These flags are set or reset depending on the operation result and referred to by the conditional branch instruction.

The flag operation instruction enables these flags to be set or reset individually or altogether.

(c) ACC

The ACC is 4-bits register for arithmetic, and equipped with data transfer instruction between ACC and RAM.

Stack

The stack consists of a RAM of 5 words x 11 bits. It is used to save the PC contents when the sub-routine is called or a timer interrupt is generated, and 5-level nesting can be done including a timer interrupt. The PC contents saved in the stack is popped to the PC by the RETURN instruction.

Interrupt

MSM6352 has two kinds of interrupt as below.

- Realtime interrupt
- Programmabletimer interrupt

Stop Mode

Stop mode is established by the execution of the STOP instruction. In the stop mode, oscillation of system clock stops and all operations are suspended, but the RAM contents and all register contents are maintained.

Halt Mode

Halt mode is established by the execution of the HALT instruction and the execution of program of main routine is suspended. In the halt mode, all RAM contents and register contents are maintained.

Timer Activation and Realtime Interrupt Circuit

The timer activation and realtime interrupt circuit are to release HALT mode (timer activiation) and to generate interrupt (Realtime interrupt) at the trailing edge of the 31.21 Hz clock obtained by dividing the 3.579545 MHz system clock by 114688. The timer activation and realtime interrupt circuit can be used for the generation of timer activation and realtime interrupt by setting or resetting the mode setting flag (TMF).

Divider Circuit

The 3 stage binary divider circuit to which 31.21 Hz clock is supplied is provided. The divider circuit's contents can be read by the input instruction (IN2, AP), and at the same time HS input port data is also read. The divider circuit can be reset by the RST instruction.

Programmable Timer and Programmable Timer Interrupt

The programmable timer is used for dial pulse output or timer interrupt generation. This timer consists of control resistor PTL, 1/100 divider circuit, 4 bit presettable down counter PTC, interrupt flag IRQF, interrupt enable flag EIF, selection flag EOF of off and on-hook dialing made, and dial pulse phase selection flag DPE, 1/100 divider circuit, PTC, IRQF, EIF, EOF and DPF are reset at system reset.

DTMF Output Circuit

The DTMF output circuit is to generate DTMF tone signal and is controlled by DTMF and TONE register. Rewriting the contents of the output latch for DTMF circuit by output instructions, 12 kinds of dual or single tones can be output to the DTMF output port. The tone output frequency is selected by the DTMF register.

BD Circuit

The BD circuit generates the square wave which can be used as the confirmation sound, warning sound and so on. 15 kinds of sound (4.66 to 0.82 kHz) are output by an output instruction specifying the frequency.

Watchdog Timer WDT

The watchdog timer is to generate the system reset signal to recover from system ran away trouble.

Input Port ($R_1 \sim R_4$)

$R_1 \sim R_4$ is 4-bits input port, which status is fetched by the input instruction. The port is pulled down to the low level (V_{SS}) by resistor, so it can be used as keyboard input port.

Input ($R_5 \sim R_8$)

$R_5 \sim R_8$ is 4-bits input port, which status is fetched by the input instruction. The port is pulled down to the low level (V_{SS}) by resistor, so it can be used as keyboard input port.

Input Port ($I_1 \sim I_4$)

$I_1 \sim I_4$ is 4-bits input port, which status can be fetched by the input instruction. It is pulled down to low level (V_{SS}) by register via transistor only when it is desired to fetch the port status or input signal is low level.

As input current is restricted, it can be used being fixed at high level (V_{DD}).

$\overline{HS}$ Input Pin

It is one bit input pin, which status can be fetched by the input instruction. It is pulled up to high level (V_{DD}) by resistor. It is used as a hook switch input pin.

Output Port ($C_1 \sim C_4$)

$C_1 \sim C_4$ is 4-bits output port. The contents of the output latch can be rewritten by the output instruction. The low level is output at each output pin after the system is reset. When the HS input pin is open or at high level, the low level is output to each output pin irrespective of the contents of the output latch.

The outputs of $C_1 \sim C_4$ are all CMOS output.

Output Port ($O_1 \sim O_4$)

$O_1 \sim O_4$ is 4-bit output port. The contents of the output latch can be rewritten by the output instruction.

Output latch of O_1 and O_2 are reset and O_3 and O_4 are set at system reset.

Each output from $O_1 \sim O_4$ port is C-MOS.

Output Port $EO_1 \sim EO_4$)

$EO_1 \sim EO_4$ is 4-bits output port. The contents of the output latch can be rewritten by the output instruction. The level is output at each output pin after the system is reset. Each output of $EO_1 \sim EO_4$ is CMOS output.

Input/Output Port ($IO_1 \sim IO_4$)

$IO_1 \sim IO_4$ is 4-bits input/output port. Fetching of the port status and rewriting of the output latch contents can be done by the input/output instruction.

Each Output of $IO_1 \sim IO_4$ is CMOS at output mode.

IOE Output Pin

It is one bit output pin. A load signal is output at this pin when the output latch's ($IO_1 \sim IO_4$) contents are rewritten.

DTMF Output Pin

It is output pin to output DTMF signals. Start and stop of the DTMF output are done by the output instruction.

DP Output Pin

It is an output pin for dial pulse output. Start and stop of the dial pulse output can be done by the output instruction.

The DP OUT pin output is C-MOS output.

BD Output Pin

It is output pin for the buzzer output. The buzzer output can be started and stopped by the output instruction. Output of BD port is CMOS output.

32 kHz Output Pin

It is an output pin to output 31.960 kHz clock (duty: 50%) which is obtained by dividing the 3.579545 MHz system clock by 112. This clock keep outputting as long as system clock oscilation is executed. Output of 32 kHz pin is CMOS out-out.

XT, $\overline{XT}$ Pins

These are input and output pins of the oscillator inverter, and the oscillator circuit is provided with the built in feed back resistor. By connecting to them oscillation of system clock status. 3.579545 MHz ceramic resonator and capacitors.

ELECTRIC CHARACTERISTICS

• Absolute Maximum Ratings

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	Ta = 25°C	−0.3 ~ 7	V
Input Voltage	V_I	Ta = 25°C	−0.3 ~ V_{DD} + 0.3	V
Output Voltage	V_O	Ta = 25°C	−0.3 ~ V_{DD} + 0.3	V
Power Dissipation	P_D	Ta = 25°C	200 max.	mW
Storage Temperature	Tstg	—	−55 ~ +125	°C

• Operating Conditions

Parameter	Symbol	Conditions	Limits	Unit
Operating Voltage	V_{DD}	Pulse Mode f_{OSC} = 3.58 MHz	2.0 ~ 5.5	V
Memory Retention Voltage	V_{DDM}	—	1.2 ~ 5.5	V
Operating Temperature	Topr	—	−20 ~ +75	°C

Note: Operating voltage for tone mode is V_{DD} = 2.2 ~ 5.5V.

• DC Characteristics

(Ta = −20 ~ +75°C)

Parameter	Symbol	Conditions		Supply Voltage	Min.	Typ.	Max.	Unit
"H" Output Current (1)	I_{OH_1}	O_3, O_4	V_{OH} = 2.6V	3.0 V	−0.2	–	–	mA
"L" Output Current (1)	I_{OL_1}	$\overline{DP}$ OUT	V_{OL} = 0.4V	3.0V	0.5	–	–	mA
"H" Output Current (2)	I_{OH_2}	$C_1 \sim C_4$	V_{OH} = 2.6V	3.0V	−1.0	–	–	mA
"L" Output Current (2)	I_{OL_2}		V_{OL} = 0.4V	3.0V	10	–	–	μA
"H" Output Current (3)	I_{OH_3}	O_1, O_2	V_{OH} = 2.6V	3.0V	−2.0	–	–	μA
"L" Output Current (3)	I_{OL_3}	BD	V_{OL} = 0.4V	3.0V	10	–	–	μA
"H" Output Current (4)	I_{OH_4}	$IO_1 \sim IO_4$ IOE $EO_1 \sim EO_4$	V_{OH} = 2.6V	3.0V	−150	–	–	μA
"L" Output Current (4)	I_{OL_4}		V_{OL} = 0.4V	3.0V	300	–	–	μA
"H" Output Current (5)	I_{OH_5}	32kHz	V_{OH} = 2.6V	3.0V	−40	–	–	μA
"L" Output Current (5)	I_{OL_5}		V_{OL} = 0.4V	3.0V	25	–	–	μA
"H" Input Voltage	V_{IH}	–		3.0V	2.2	–	–	V
				5.5V	40	–	–	
"L" Input Voltage	V_{IL}	–		3.0V	–	–	0.8	V
				5.5V	–	–	1.4	
"H" Input Current (1)	I_{IH_1}	$\overline{HS}$	V_{IH} = 5.5V	5.5V	–	–	2	μA
"L" Input Current (1)	I_{IL_1}		V_{IL} = 0V	3.0V	−2.0	–	−180	μA
				5.5V	−40	–	−360	
"H" Input Current (2)	I_{IH_2}	$R_1 \sim R_8$	V_{IH} = 5.5V	5.5V	20	–	180	μA
			V_{IH} = 3.0V	3.0V	10	–	90	
"L" Input Current (2)	I_{IL_2}		V_{IL} = 0V	5.5V	–	–	−2	μA
"H" Input Current (3)	I_{IH_3}	$I_1 \sim I_4$ AC, TEST	V_{IH} = 5.5V	5.5V	60	–	600	μA
			V_{IH} = 3.0V	3.0V	30	–	300	
"L" Input Current (3)	I_{IL_3}		V_{IL} = 0V	5.5V	–	–	−2	μA
"H" Input Current (4)	I_{IH_4}	$IO_1 \sim IO_4$	V_{IH} = 5.5V	5.5V	–	–	2	μA
"L" Input Current (4)	I_{IL_4}		V_{IL} = 0V	5.5V	–	–	−2	μA
Current Consumption (1)	I_{DDP}	Tone output off With no load		2.5V	–	0.25	0.5	mA
				5.0V	–	1.5	2.4	
Current Consumption (2)	I_{DDT}	Tone output on With no load		2.5V	–	1.3	2.4	mA
				5.0V	–	4.2	6.8	
Current Consumption (3)	I_{DDM}	ON HOOK, Ta = 25°C With no load		2.5V	–	–	0.2	μA

• AC Characteristics

(Ta = –20 ~ +75°C)

Parameter	Symbol	Conditions	Supply Voltage	Min.	Typ.	Max.	Unit
Cycle Time	t_{CY}	f = 3.579545MHz	3.0V	–	17.9	–	μs
Tone Output	V_{OUT}	Row side only R_L = 1kΩ	2.2V	–	180	–	mV rms
			4.0V	–	260	–	
			5.5V	–	330	–	
High/Low Level Ratio	dB_{CR}	–	3.0V	1	2	3	dB
			5.5V	1	2	3	
Distorsion Ratio	$\%d_{IS}$	R_L = 1kΩ	3.0V	–	–	5	%
			5.5V	–	–	5	
Switch Input Time	t_{KIN}		–	16	–	–	ms
Rise/Fall Time (1)	t_{TLH_1}	O_3, O_4, DP OUT C_L = 50pF	3.0V	–	–	0.5	μs
	t_{THL_1}		3.0V	–	–	0.5	
Rise/Fall Time (2)	t_{TLH_2}	$C_1 \sim C_4$ C_L = 50pF	3.0V	–	–	0.5	μs
	t_{THL_2}		3.0V	–	–	10	
Rise/Fall Time (3)	t_{TLH_3}	O_1, O_2, BD, 32kHz C_L = 50pF	3.0V	–	–	5	μs
	t_{THL_3}		3.0V	–	–	10	
Rise/Fall Time (4)	t_{TLH_4}	$IO_1 \sim IO_4$, IOE $EO_1 \sim EO_4$ C_L = 50pF	3.0V	–	–	1	μs
	t_{THL_4}		3.0V	–	–	1	

• DTMF Tone Output Freqeuncy

	Standard Frequency (Hz)	Output Freqeuncy (Hz)	Deviation (%)
R1	697	699.1	+0.30
R2	770	766.2	–0.49
R3	852	847.4	–0.54
R4	941	948.0	+0.74
C1	1209	1215.9	+0.57
C2	1336	1331.7	–0.32
C3	1477	1471.9	–0.35

f_{OSC} = 3.579545MHz

	Mnemonic	Instruction Code 13 12	11 10 9 8	7 6 5 4	3 2 1 0		ACC	Zero	Carry	Greater	Description
Arithmetic operation	ADD ACC, AP	0 0	0 0 0 P	0 1 0 0	A	$A=0_H\sim F_H$, P=0 or 1	*	*	*	–	AP ← (AP) + ACC
	ADD #D, AP	0 1	1 0 0 P	D	A	$D=0_H\sim F_H$ $A=0_H\sim F_H$, P=0 or 1	*	*	*	–	AP ← (AP) + D
	ADC AP	0 0	0 0 0 P	0 1 0 1	A	$A=0_H\sim F_H$, P=0 or 1	*	*	*	–	AP ← (AP) + ACC + C
	SUB ACC, AP	0 0	0 0 1 P	0 1 0 0	A	$A=0_H\sim F_H$, P=0 or 1	*	*	*	–	AP ← (AP)– ACC
	SUB #D, AP	0 1	1 0 1 P	D	A	$D=0_H\sim F_H$, $A=0_H\sim F_H$, P=0 or 1	*	*	*	–	AP ← (AP) – D
	SBC AP	0 0	0 0 1 P	0 1 0 1	A	$A=0_H\sim F_H$, P=0 or 1	*	*	*	–	AP ← (AP) – ACC – C
	CMP ACC, AP	0 0	0 0 1 P	1 1 1 0	A	$A=0_H\sim F_H$, P=0 or 1	–	*	–	*	(AP) – ACC
	CMP #D, AP	0 1	0 1 1 P	D	A	$D=0_H\sim F_H$, $A=0_H\sim F_H$, P=0 or 1	–	*	–	*	(AP) – D
	XOR ACC, AP	0 0	0 0 0 P	0 1 1 1	A	$A=0_H\sim F_H$, P=0 or 1	*	*	–	–	AP ← (AP) ∀ ACC
	XOR #D, AP	0 1	1 1 1 P	D	A	$D=0_H\sim F_H$, $A=0_H\sim F_H$, P=0 or 1	*	*	–	–	AP ← (AP) ∀ D
Bit operation	BIT ACC, AP	0 0	0 0 0 P	1 1 1 0	A	$A=0_H\sim F_H$, P=0 or 1	–	*	–	–	(AP) ∨ $\overline{ACC}$
	BIT #D, AP	0 1	0 1 0 P	D	A	$D=0_H\sim F_H$, $A=0_H\sim F_H$, p=0 or 1	–	*	–	–	(AP) ∨ $\overline{D}$
	BIS ACC, AP	0 0	0 0 0 P	1 1 1 0	A	$A=0_H\sim F_H$, P=0 or 1	*	*	–	–	AP ← (AP) ∨ ACC
	BIS #D, AP	0 1	0 0 0 P	D	A	$D=0_H\sim F_H$, $A=0_H\sim F_H$, P=0 or 1	*	*	–	–	AP ← (AP) ∨ D
	BIC ACC; AP	0 0	0 0 1 P	0 1 1 0	A	$A=0_H\sim F_H$, P=0 or 1	*	*	–	–	AP ← (AP) ∧ $\overline{ACC}$
	BIC #D, AP	0 1	0 0 1 P	D	A	$D=0_H\sim F_H$, $A=0_H\sim F_H$, P=0 or 1	*	*	–	–	AP ← (AP) ∧ $\overline{D}$
Rotate	ROR AP	0 0	0 0 0 P	0 0 1 0	A	$A=0_H\sim F_H$, P=0 or 1	*	*	*	–	↳ (AP) → C ⌋
	ROL AP	0 0	0 0 1 P	0 0 1 0	A	$A=0_H\sim F_H$, P=0 or 1	*	*	*	–	⌊ (AP) ← C ↵
	ASR AP	0 0	0 0 0 P	0 0 1 1	A	$A=0_H\sim F_H$, P=0 or 1	*	*	*	–	0 → (AP) → C
	ASL AP	0 0	0 0 1 P	0 0 1 1	A	$A=0_H\sim F_H$, P=0 or 1	*	*	*	–	C ← (AP) ← 0
Flag operation	SEZ	0 0	0 0 1 0	1 0 1 0	0 0 0 0		–	1	–	–	Z ← 1
	CLZ	0 0	0 0 0 0	1 0 1 0	0 0 0 0		–	0	–	–	Z ← 0
	SEC	0 0	0 0 1 0	1 0 0 1	0 0 0 0		–	–	1	–	C ← 1
	CLC	0 0	0 0 0 0	1 0 0 1	0 0 0 0		–	–	0	–	C ← 0
	SEG	0 0	0 0 1 0	1 0 0 0	0 0 0 0		–	–	–	1	G ← 1
	CLG	0 0	0 0 0 0	1 0 0 0	0 0 0 0		–	–	–	0	G ← 0
	SEA	0 0	0 0 1 0	1 0 1 1	0 0 0 0		–	1	1	1	Z ← 1, C ← 1, G ← 1
	CLA	0 0	0 0 0 0	1 0 1 1	0 0 0 0		–	0	0	0	Z ← 0, C ← 0, G ← 0
Data transfer	MOV ACC. AP	1 1	1 1 0 1	0 0 0 0	A	$A=0_H\sim F_H$	–	–	–	–	AP ← ACC
	MOV ACC, AX	1 1	1 1 0 0	X	A	$X=0_H\sim F_H$ $A=0_H\sim F_H$	–	–	–	–	AX ← ACC
	MOV #D, AP	0 1	1 1 0 P	D	A	$D=0_H\sim F_H$ $A=0_H\sim F_H$, P=0 or 1	*	*	–	–	AP ← D
	MOV AP, ACC	1 1	1 1 1 1	0 0 0 0	A	$A=0_H\sim F_H$	*	*	–	–	ACC ← (AP)
	MOV AX, ACC	1 1	1 1 1 0	X	A	$X=0_H\sim F_H$ $A=0_H\sim F_H$	*	*	–	–	ACC ← (AX)
	CHG AP	1 1	1 0 0 1	0 0 0 0	A	$A=0_H\sim F_H$	*	–	–	–	(AP) ↔ ACC
	CHG AX	1 1	1 0 0 0	X	A	$X=0_H\sim F_H$ $A=0_H\sim F_H$	*	–	–	–	(AX) ↔ ACC

	Mnemonic	13 12	11 10 9 8	7 6 5 4	3 2 1 0		ACC	Zero	Carry	Greater	Description
Data transfer	RDAR	1 1	0 0 0 0	0 0 0 0	0 0 0 0		*	*	–	–	ACC ← (AR_1)
	RDAR +(–)	1 1	0 0 0 0	0 0 1 D/I	0 0 0 0	D/I = 0 or 1	*	*	–	–	ACC ← (AR_1), AR_1 ← AR_1 ±1
	RDAR +(–), Z	1 1	0 0 0 0	0 1 0 D/I	0 0 0 0	D/I = 0 or 1	0	1	–	–	ACC ← (AR_1) if (AR_1)=0 then PC ← PC+1 else AR_1 ← AR_1 ±1, repeat
	RDAR +(–), N	1 1	0 0 0 0	1 0 0 D/I	0 0 0 0	D/I = 0 or 1	*	0	–	–	ACC ← (AR_1) if (AR_1)=0 then PC ← PC+1 else AR_1 ← AR_1 ±1, repeat
	RDAR +(–),Z,L	1 1	0 0 1 0	0 1 0 D/I	0 0 0 0	D/I = 0 or 1	*	*	–	–	ACC ← (AR_1), L ← L–1 if (AR_1)=0 or L=0 then PC ← PC+1 else AR_1 ← AR_1 ±1, repeat
	RDAR +(–),N,L	1 1	0 0 1 0	1 0 0 D/I	0 0 0 0	D/I = 0 or 1	*	*	–	–	ACC ← (AR_1), L ← L–1 if (AR_1)=0 or L=0 then PC ← PC+1 else AR_1 ← AR_1 ±1, repeat
	MVAR	1 1	0 1 0 0	0 0 0 0	0 0 0 0		–	*	–	–	(AR_2) ← (AR_1)
	MVAR +(–)	1 1	0 1 0 0	0 0 1 D/I	0 0 0 0	D/I = 0 or 1	–	*	–	–	(AR_2) ← (AR_1), AR_1←AR_1 ±1, AR_2←AR_2 ±1
	MVAR +(–), Z	1 1	0 1 0 0	0 1 0 D/I	0 0 0 0	D/I = 0 or 1	–	1	–	–	(AR_2) ← (AR_1) if (AR_1)=0 then PC ← PC+1 else AR_1←AR_1 ±1, AR_2←AR_2 ±1, repeat
	MVAR +(–), N	1 1	0 1 0 0	1 0 0 D/I	0 0 0 0	D/I = 0 or 1	–	0	–	–	(AR_2) ← (AR_1) if (AR_1)=0 then PC ← PC+1 else AR_1←AR_1 ±1, AR_2←AR_2 ±1, repeat
	MVAR +(–), L	1 1	0 1 1 0	0 0 0 D/I	0 0 0 0	D/I = 0 or 1	–	*	–	–	(AR_2) ← (AR_1), L ← L–1 if L=0 then PC ← PC+1 else AR_1←AR_1 ±1, AR_2←AR_2 ±1, repeat
	MVAR +(–),Z,L	1 1	0 1 1 0	0 1 0 D/I	0 0 0 0	D/I = 0 or 1	–	*	–	–	(AR_2) ← (AR_1), L ← L–1 if (AR_1)=0 or L=0 then PC ← PC+1 else AR_1←AR_1 ±1, AR_2←AR_2 ±1, repeat
	MVAR +(–),N,L	1 1	0 1 1 0	1 0 0 D/I	0 0 0 0	D/I = 0 or 1	–	*	–	–	(AR_2) ← (AR_1), L ← L–1 if (AR_1)=0 or L=0 then PC ← PC+1 else AR_1←AR_1 ±1, AR_2←AR_2 ±1, repeat
Subroutine	CALL adrs	1 0	1 a_{10} a_9 a_8	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0	a_{10}~a_0=000_H~$7FF_H$	–	–	–	–	STACK ← (PC), PC ← adrs
	RET	0 0	0 0 0 0	1 1 0 0	0 0 0 0		–	–	–	–	PC ← (STACK) + 1
	RTI	0 0	0 0 1 0	1 1 0 0	0 0 0 0		–	–	–	–	PC ← (STACK) or PC ← (STACK) + 1
Jump	JMP adrs	1 0	0 a_{10} a_9 a_8	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0	a_{10}~a_0=000_H~$7FF_H$	–	–	–	–	PC ← adrs
	JMP @ AP	0 0	0 0 0 P	1 1 0 1	A	A=0_H~F_H, P=0 or 1	–	–	–	–	PC ← (PC) + (AP) + 1
	JMPIO @ AP	0 0	0 0 1 P	1 1 0 1	A	A=0_H~F_H, P=0 or 1	–	–	–	–	PC ← (PC) + [(AP) A 7] + 1
Branch	BEQ n (BZE n)	1 1	1 0 1 P	0 1 0 n_4	n_3 n_2 n_1 n_0	n_4~n_0=00_H~$1F_H$ P=0 or 1	–	–	–	–	if Z=1 then PC←PC–n or PC←PC+n+1 else PC←PC+1
	BNE n (BNZ n)	1 1	1 0 1 P	1 1 0 n_4	n_3 n_2 n_1 n_0	n_4~n_0=00_H~$1F_H$ P=0 or 1	–	–	–	–	if Z=0 then PC←PC–n or PC←PC+n+1 else PC←PC+1
	BCS n	1 1	1 0 1 P	0 0 0 n_4	n_3 n_2 n_1 n_0	n_4~n_0=00_H~$1F_H$ P=0 or 1	–	–	–	–	if C=1 then PC←PC–n or PC←PC+n+1 else PC←PC+1
	BCC n	1 1	1 0 1 P	1 0 0 n_4	n_3 n_2 n_1 n_0	n_4~n_0=00_H~$1F_H$ P=0 or 1	–	–	–	–	if C=0 then PC←PC–n or PC←PC+n+1 else PC←PC+1
	BGT n	1 1	1 0 1 P	0 0 1 n_4	n_3 n_2 n_1 n_0	n_4~n_0=00_H~$1F_H$ P=0 or 1	–	–	–	–	if G=1 then PC←PC–n or PC←PC+n+1 else PC←PC+1
	BLE n	1 1	1 0 1 P	1 0 1 n_4	n_3 n_2 n_1 n_0	n_4~n_0=00_H~$1F_H$ P=0 or 1	–	–	–	–	if G=0 then pC←PC–n or PC←PC+n+1 else PC←PC+1
	BGE n	1 1	1 0 1 P	0 1 1 n_4	n_3 n_2 n_1 n_0	n_4~n_0=00_H~$1F_H$ P=0 or 1	–	–	–	–	if Z=1 or G=1 then PC←PC–n or PC+n+1 else PC←PC+1
	BLT n	1 1	1 0 1 P	1 1 1 n_4	n_3 n_2 n_1 n_0	n_4~n_0=00_H~$1F_H$ P=0 or 1	–	–	–	–	if Z=0 then PC←PC–n or PC←PC+n+1 and G=0 else PC←PC+1
Input/Output	IN PORT, AP	0 0	0 1 0 P	P_L	A	P_L=0_H~F_H A=0_H~F_H, P=0 or 1	*	*	–	–	AP ← (PORT)
	OUT AP, PORT	0 0	1 0 P_H P	P_L	A	P_L=0_H~F_H, P_H=0 or 1 A=0_H~F_H, P=0 or 1	–	–	–	–	PORT ← (AP)
	OUT #D, PORT	0 0	1 1 P_H 0	P_L	D	P_L=0_H~F_H, P_H=0 or 1 D=0_H~F_H	–	–	–	–	PORT ← D

	Mnemonic	Instruction Code														ACC	Zero	Carry	Greater	Description
		13	12	11	10	9	8	7	6	5	4	3	2	1	0					
CPU Control & Others	STOP	0	0	1	1	1	0	0	0	0	0	0	0	0	0	–	–	–	–	Stop system clock
	HALT	0	0	1	1	1	0	0	0	0	1	0	0	0	0	–	–	–	–	Halt CPU
	ACT	0	0	1	1	1	0	0	0	1	0	0	0	0	0	–	–	–	–	Activate CPU
	EI	0	0	1	1	1	0	0	1	1	0	1	0	0	0	–	–	–	–	Enable timer interrupt
	DI	0	0	1	1	1	0	0	1	1	0	0	1	0	0	–	–	–	–	Disable timer interrupt
	ET	0	0	1	1	1	0	0	1	1	0	0	0	1	0	–	–	–	–	Enable timer activate
	DT	0	0	1	1	1	0	0	1	1	0	0	0	0	1	–	–	–	–	Disable timer activate
	EC	0	0	1	1	1	0	0	1	1	1	1	0	0	0	–	–	–	–	Enable output port ($C_1 \sim C_4$)
	DC	0	0	1	1	1	0	0	1	1	1	0	1	0	0	–	–	–	–	Disable output port ($C_1 \sim C_4$)
	OM	0	0	1	1	1	0	0	1	1	1	0	0	1	0	–	–	–	–	Set I/O port ($IO_1 \sim IO_4$) to output mode
	IM	0	0	1	1	1	0	0	1	1	1	0	0	0	1	–	–	–	–	Set I/O port ($IO_1 \sim IO_4$) to input mode
	RST	0	0	1	1	1	0	1	0	0	1	0	0	0	0	–	–	–	–	Reset divider
	SELINT	0	0	1	1	1	0	1	1	0	0	1	0	0	0	–	–	–	–	Select interrupt mode
	SELHR	0	0	1	1	1	0	1	1	0	0	0	1	0	0	–	–	–	–	Select timer activation mode
	EHLH	0	0	1	1	1	0	1	1	0	0	0	0	1	0	–	–	–	–	Enable hook switch low to high transmission
	DHLH	0	0	1	1	1	0	1	1	0	0	0	0	0	1	–	–	–	–	Disable hook switch low to high transmission
	SELN	0	0	1	1	1	0	1	0	1	1	0	0	1	0	–	–	–	–	Select negative phase
	SELP	0	0	1	1	1	0	1	0	1	1	0	0	0	1	–	–	–	–	Select positive phase
	EO	0	0	1	1	1	0	1	0	1	1	1	0	0	0	–	–	–	–	Enable on-hook dial
	DO	0	0	1	1	1	0	1	0	1	1	0	1	0	0	–	–	–	–	Disable on-hook dial
	NOP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	–	–	–	–	No operation

OLMS-64/65 SERIES

MSM6404

HIGH-SPEED 4-BIT SINGLE CHIP MICROCONTROLLER

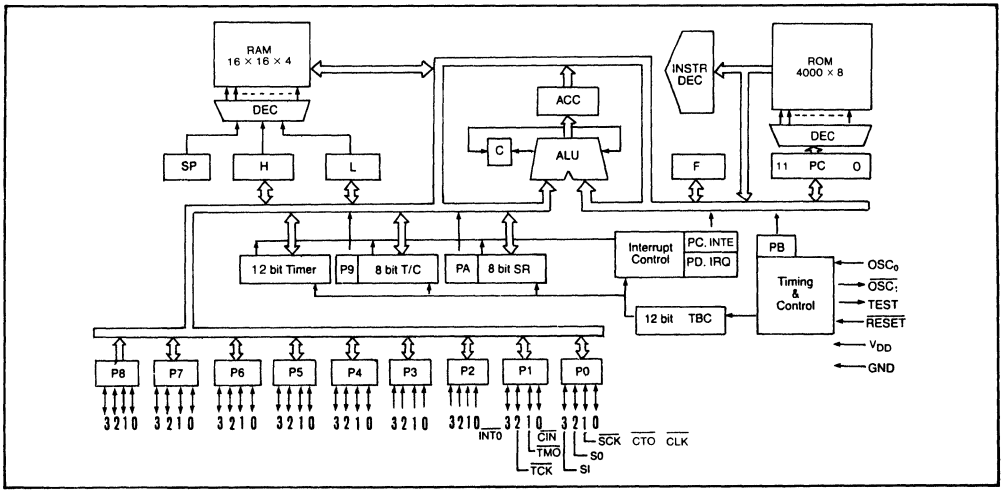
GENERAL DESCRIPTION

The OKI MSM6404 microcontroller is a low power, high-performance single-chip device implemented in complementary metal oxide semiconductor technology. 32K bits of mask program ROM, 1024 bits of data RAM, 36 Input/Output lines, a programmable timer/event-counter, and oscillator are integrated onto one chip. Program memory is byte wide and data-paths are organized in 4 bit nibbles. RAM and I/O lines are bit addressable. 122 instructions include binary, BCD operations; bit set, reset, test; 8 bit I/O; relative jumps; multifunctional instructional (increment, modify, skip) 8 bit wide table output; subroutine call and return.

FEATURES

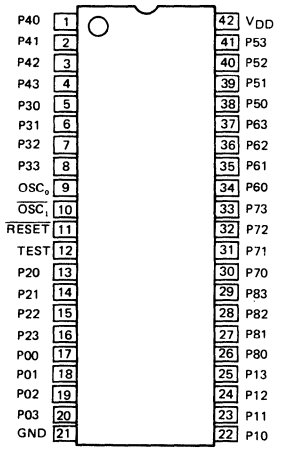
- 4000 × 8 MASK ROM
  An evaluation board is available for up to 8k × 8.
- 256 × 4 RAM (including the stack area)
- 9 × 4 Ports, 36 I/O lines
  4 lines for input ports having a latch, and the other 32 lines for bit operation are available.
- Three built-in counters
  12-bit time-base counter
  12-bit programmable timer
  8-bit high-speed programmable timer/event counter
- Built-in 8-bit serial I/O register (with 3-bit counter)
- Five interrupts with five priority levels (4 internal, 1 external)
- 32 stack levels (in RAM)
- LED direct drive available (8mA x 5 ports at the same time)
- Power down features
- Instruction execution time
  952 ns 4.2 MHz clock
- Instruction systems suitable for control
- 122 instructions
- Mask option
  P60-63 for input port
- Full static operation
- Low power consumption
  TYP 0.4μW at V_{DD}=2V
  TYP 5μW at V_{DD}=5V 0Hz clock
- 5V single power supply, 42-pin DIP or 44-pin FLAT
- Package:
  42 pin plastic DIP (DIP42-P-600)
  44 pin plastic QFP (QFP44-P-910-K)
  44 pin plastic QFP (QFP44-P-910-VK)
  44 pin plastic QFP (QFP44-P-910-VIK)

BLOCK DIAGRAM

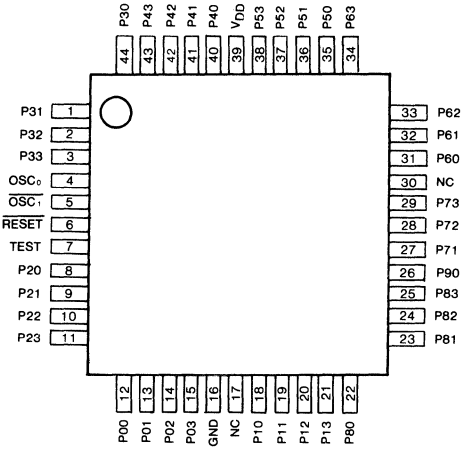


PIN CONFIGURATION

(Top View) 42-pin Plastic DIP



(Top View) 44-pin Plastic Flat Package



PIN DESCRIPTION

Pin name	Input/output	Function		When reset
P00 P01 / $\overline{SCK}$ P02 / SO P03 / SI	Input/output	4-bit input/output port. P01 to P03 are also used as serial interface terminals.		"1"
P10 / $\overline{CIN}$ P11 / $\overline{TMO}$ P12 / $\overline{TCK}$ P13	Input/output	4-bit input port with latch. Built-in pull up register for all bit input.		"1"
P20 / $\overline{INT}$ P21 P22 P23	Input	4-bit input port with a latch. P20 is shared with INT input. (Fall trigger input) P21 ~ 23 are level input. Built-in pull up register for all bit input.		The latch is reset.
P30–33	Input/output	4-bit input/output port		"1"
P40–43	Input/output	4-bit input/output port	8-bit output port	"0"
P50–53	Input/output	4-bit input/output port		"0"
P60–63	Input/output	4-bit input/output port *1		"0"
P70–73	Input/output	4-bit input/output port		"0"
P80–83	Input/output	4-bit input/output port		"0"
OSC_0 $\overline{OSC_1}$	Input/output	X'tal connection terminal for system clock oscillation		Oscillation wave
TEST	Output	(Test terminal for Maker)		Pulse output
$\overline{RESET}$	Input	System reset input terminal		
VDD GND		Power source voltage supply		

Note: When each port is used for output, it is possible to drive one TTL (one input).
*1 Can be made as a port dedicated to input (mask option).

INSTRUCTION LIST

	Mnemonic		Description	Code	Byte	Cycle
Load, Push, Pop	LAI	n	A ← n	9n	1	1
	LLI	n	L ← n	8n	1	1
	LHLI	nn	HL ← nn	15nn	2	2
	LMI	nn	M(w) ← nn	14nn	2	2
	LAL		A ← L	21	1	1
	LLA		L ← A	2D	1	1
	LAH		A ← H	22	1	1
	LHA		H ← A	2E	1	1
	LAM		A ← M	38	1	1
	LMA		M ← A	2F	1	1
	LAM+		A ← M, L ← L+1, Skip if L=0	24	1	1
	LAM−		A ← M, L ← L−1, Skip if L=F	25	1	1
	LMA+		M ← A, L ← L+1, Skip if L=0	26	1	1
	LMA−		M ← A, L ← L−1, Skip if L=F	27	1	1
	LAMM	n_2	A ← M, H ← H V n_2	39−3B	1	1
	LAMD	mm	A ← Md	10mm	2	2
	LMAD	mm	Md ← A	11mm	2	2
	LMTD	mm	Md(w) ← T (M(w), A), T=ROM table	19mm	2	3
	LMCT		M(w) ← CT	3E59	2	2
	LCTM		CT ← M(w)	3E51	2	2
	LMSR		M(w) ← SR	3E5A	2	2
	LSRM		SR ← M(w)	3E52	2	2
	LTMM		TM ← (M(w), A)	3E50	2	2
	PUSH		ST ← C, A, H, L, SP ← SP−4	1C	1	3
	POP		C, A, H, L ←, ST SP ← SP+4	1D	1	3
Exchange	X		A ⟷ M	28	1	1
	XM	n_2	A ⟷ M, H ← H V n_2	29−2B	1	1
	X+		A ⟷ M, L ← L+1, Skip if L=0	3C	1	1
	X−		A ⟷ M, L ← L−1, Skip if L=F	2C	1	1
Increment/ Decrement	INA		A ← A+1, Skip if A=0	30	1	1
	INM		M ← M+1, Skip if M=0	33	1	1
	INL		L ← L+1, Skip if L=0	31	1	1

INSTRUCTION LIST (Continued)

	Mnemonic		Description	Code	Byte	Cycle
Increment/Decrement	INH		H ← H+1, Skip if M = 0	32	1	1
	INMD	mm	Md ← Md+1, Skip if Md = 0	12mm	2	2
	DCA		A ← A−1, Skip if A = F	34	1	1
	DCM		M ← M−1, Skip if M = F	37	1	1
	DCL		L ← L−1, Skip if L = F	35	1	1
	DCH		H ← H−1, Skip if H = F	36	1	1
	DCMD	mm	Md ← Md−1, Skip if Md = F	13mm	2	2
Arithmetic	ADS		A ← A+M, Skip if Cy = 1	02	1	1
	ADCS		A, C ← A+M+C, Skip if Cy=1	01	1	1
	ADC		A, C ← A+M+C	03	1	1
	AIS	n	A ← A+n, Skip if Cy = 1	3E4n	2	2
	DAA		A ← A+6	06	1	1
	DAS		A ← A+10	0A	1	1
	AND		A ← A ∨ M	0D	1	1
	OR		A ← A ∨ M	05	1	1
	EOR		A ← A ∨ M	04	1	1
	CMA		A ← $\overline{A}$	0B	1	1
	CIA		A ← $\overline{A}$+1	0C	1	1
	RAL		Rotate Left with C	0E	1	1
	RAR		Rotate Right with C	0F	1	1
	TC		Skip if C = 1	09	1	1
	SC		C ← 1	07	1	1
	RC		C ← 0	08	1	1
Compare	CAI	n	Skip if A = n	3E0n	2	2
	CLI	n	Skip if L = n	3E2n	2	2
	CPI	p, n	Skip if P_p = n	17pn	2	2
	CMI	n	Skip if M = n	3E1n	2	2
	CAM		Skip if A = M	16	1	1
Bit operation	TAB	n_2	Skip if Abit (n_2) = 1	54–57	1	1
	RAB	n_2	Abit (n_2) ← 0	64-67	1	1
	SAB	n_2	Abit (n_2) ← 1	74–77	1	1
	TMB	n_2	Skip if Mbit (n_2) = 1	58–5B	1	1
	RMB	n_2	Mbit (n_2) ← 0	68–6B	1	1

INSTRUCTION LIST (Continued)

	Mnemonic		Description	Code	Byte	Cycle
Bit operation	SMB	n	Mbit (n_2) ←1	78–7B	1	1
	TFB	n_2	Skip if Fbit (n_2) =1	5C–5F	1	1
	RFB	n_2	Fbit (n_2) ←0	6C-6F	1	1
	SFB	n_2	Fbit (n_2) ←1	7C–7F	1	1
	TPB	n_2	Skip if Pbit (n_2) =1	50–53	1	1
	RPB	n_2	Pbit (n_2) ←0	60–63	1	1
	SPB	n	Pbit (n_2) ←1	70–73	1	1
	TPBD	p n_2	Skip if $P_p$bit (n_2) =1	3D $p_{0\sim3}$	2	2
	RPBD	p n_2	$P_p$bit (n_2) =0	3D $p_{4\sim7}$	2	2
	SPBD	p n_2	$P_p$bit (n_2) =1	3D $p_{8\sim B}$	2	2
Interrupt	MEI		MEIF ←1	3E60	2	2
	MDI		MEIF ←0	3E61	2	2
	EITB		EITBF ←1	3DC9	2	2
	EITM		EITMF ←1	3DCA	2	2
	EICT		EICTF ←1	3DCB	2	2
	EIEX		EIEXF ←1	3DC8	2	2
	DITB		EITBF ←0	3DC5	2	2
	DITM		EITMF ←0	3DC6	2	2
	DICT		EICTF ←0	3DC7	2	2
	DIEX		EIEXF ←0	3DC4	2	2
	TITB		Skip If EITBF =1	3DC1	2	2
	TITM		Skip If EITMF =1	3DC2	2	2
	TICT		Skip If EICTF =1	3DC3	2	2
	TIEX		Skip If EIEXF =1	3DC0	2	2
	TQEX		Skip If IRQEX =1	3D20	2	2
	TQTB		Skip If IRQTB =1	3DD0	2	2
	TQTM		Skip If IRQTM =1	3DD1	2	2
	TQCT		Skip If IRQCT =1	3DD2	2	2
	TQSR		Skip If IRQSR =1	3DD3	2	2
	RQEX		IRQ EX ←0	3D24	2	2
	RQTB		IRQ TB ←0	3DD4	2	2
	RQTM		IRQ TM ←0	3DD5	2	2
	RQCT		IRQ CT ←0	3DD6	2	2
	RQSR		IRQ SR ←0	3DD7	2	2

INSTRUCTION LIST (Continued)

	Mnemonic		Description	Code	Byte	Cycle
Shift resistor Counter	ECT		CTF ←1 (start)	3DBB	2	2
	ESR		SRF ←1 (start)	3DBA	2	2
	DCT		CTF ←0 (stop)	3DB7	2	2
	DSR		SRF ←0 (stop)	3DB6	2	2
	TCT		Skip If CTF =1	3DB3	2	2
	TSR		Skip If SRF =1	3DB2	2	2
Branch	JCP	a_6	PC ←a_6	C0~FF	1	1
	JP	a_{12}	PC ←a_{12}	4a_{12}	2	2
	CZP	a	ST ←PC+1, PC ←2a, SP ←SP−4	Ba	1	4
	CAL	a_{12}	ST ←PC+2, PC ←a_{12}, SP ←SP−4	Aa_{12}	2	4
	OPT		P_5, P_4 ←T (M(w), A), T=ROM table	18	1	3
	RT		PC ←ST, SP ←SP+4	IE	1	4
	RTS		PC ←ST, SP ←SP+4, Skip unconditional	IF	1	4
	JA		PC ←(PC ←A)+1	IA	1	1
	JM		PC ←(M(w), A)	IB	1	2
Input/Output	IP		A ←P	20	1	1
	IPD	p	A ←Pp	3DpD	2	2
	OP		P ←A	23	1	1
	OPD	p	Pp ←A	3DpC	2	2
CPU control	NOP		No Operation	00	1	1
	HALT		Halt CPU	3DB8	2	2
	STOP		Stop Clock	3DB9	2	2

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	Ta = 25°C	−0.3 to 7	V
Input Voltage	V_I		−0.3 to V_{DD}	V
Output Voltage	V_O		−0.3 to V_{DD}	V
Power Dissipation	P_D	Ta = 25°C per package	200 max.	mW
		Ta = 25°C per output	50 max.	mW
Storage Temperature	Tstg	–	−55 to +150	°C

OPERATING RANGE

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	$f_{(OSC)} \leqq 1$ MHz	3 to 6	V
		$f_{(OSC)} \leqq 4.2$ MHz	4.5 to 5.5	V
Data-Hold Voltage	V_{DDH}	f (OSC) = 0 Hz	2 to 6	V
Operating Temperature	T_{OP}	–	−40 to +85	°C
Fan Out	N	MOS Load	15	–
		TTL Load	1	

DC CHARACTERISTICS

(V_{DD} = 5V±10%, Ta = −40 to +85°C)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
"H" Input Voltage*1,*2	V_{IH}	–	2.4	–	V_{DD}	V
"H" Input Voltage*3,*4	V_{IH}	–	3.6	–	V_{DD}	V
"L" Input Voltage	V_{IL}	–	−0.3	–	0.8	V
"H" Output Voltage*1,*5	V_{OH}	I_O = −15μA	4.2	–	–	V
"L" Output Voltage*1	V_{OL}	I_O = 1.6mA	–	–	0.4	V
"L" Output Voltage*5	V_{OL}	I_O = 15μA	–	–	0.4	V
"L" Output Voltage*6	V_{OL}	I_O = 8mA	–	1	2	V
Input Current*3	I_{IH}/I_{IL}	$V_I = V_{DD}/0V$	–	–	15/−15	μA
Input Current*2,*4	I_{IH}/I_{IL}	$V_I = V_{DD}/0V$	–	–	1/−30	μA
"H" Output Current*1	I_{OH}	V_O = 2.4V	−0.1	–	–	mA
"H" Output Current*1	I_{OH}	V_O = 0.4V	–	–	−1.2	mA
Input Capacity	C_I	f=1 MHz Ta=25°C	–	5	–	pF
Output Capacity	C_O		–	7	–	
Current Dissipation (when stop condition)	I_{DDS}	V_{DD}=2V, no load Ta=25°C	–	0.2	5	μA
		No load	–	1	100	μA
Current Dissipation	I_{DD}	Quarts oscillation f=4 MHz, no load	–	6	12	mA

*1 Applied to P0, P1, P3, P4, P5, P6, P7 and P8

*2 Applied to P2 *3 Applied to OSC_0 *4 Applied to $\overline{RESET}$ *5 Applied to $\overline{OSC_1}$

*6 In using LED, total output current should be within the limit of Power dissipitation in "Absolute Maximum Rating."

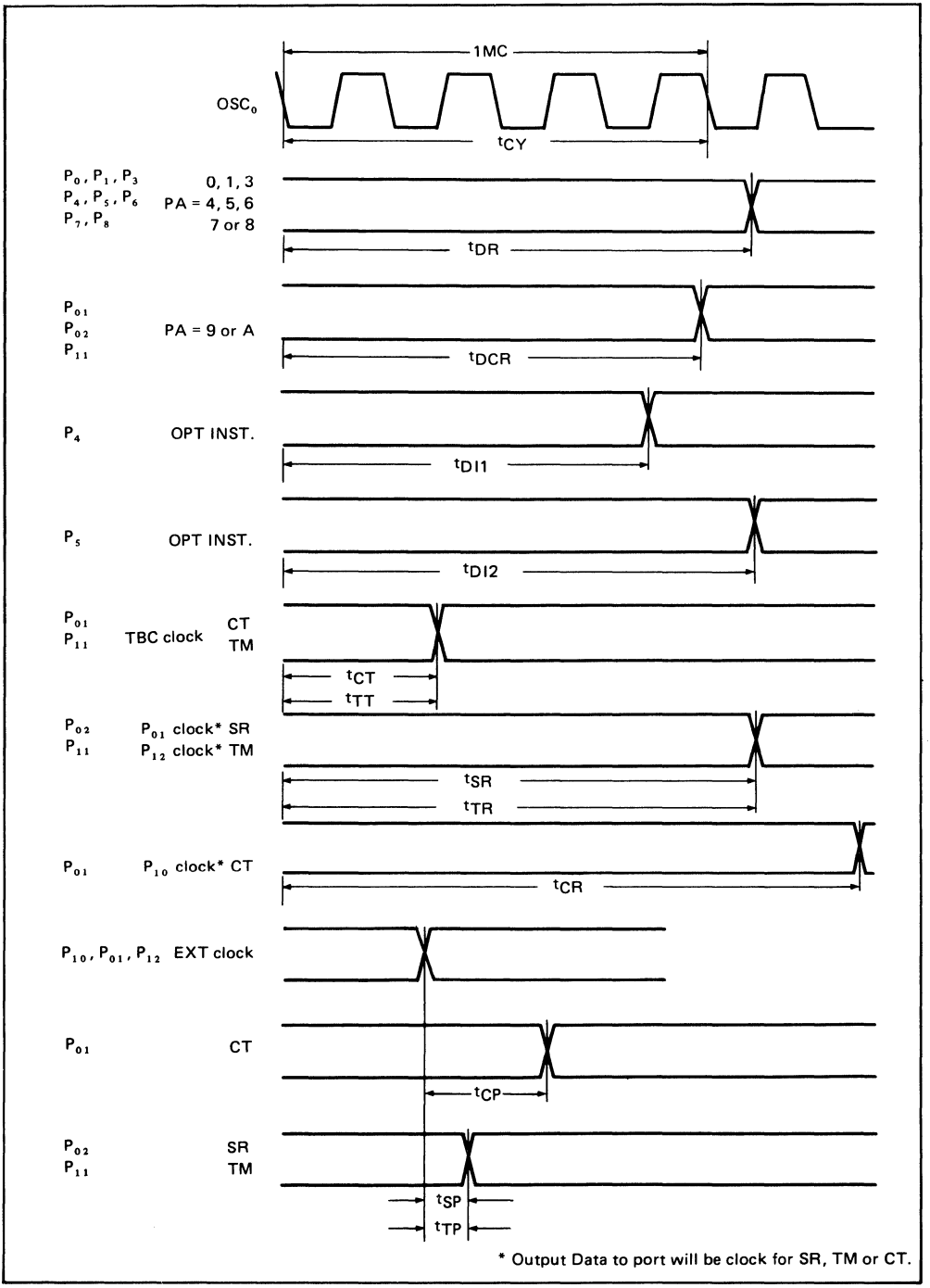
AC CHARACTERISTICS

(V_{DD} = 5V±10%, Ta = −40° to +85°C)

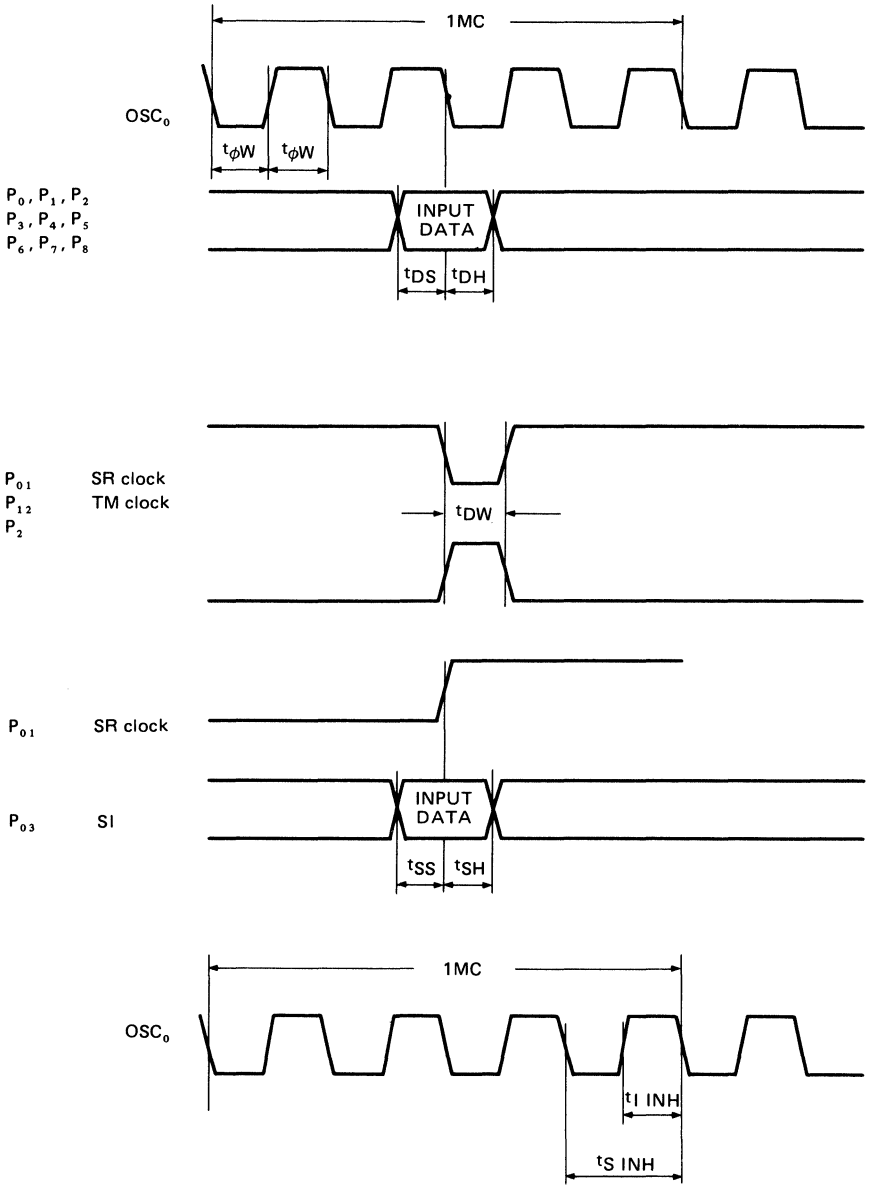
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Clock Pulse Width Clock (OSC)	$t_{\phi W}$	–	119	–	–	nS
Cycle Time	t_{CY}	–	952	–	–	nS
Input Data Setup Time	t_{DS}	–	120	–	–	nS
Input Data Hold Time	t_{DH}	–	120	–	–	nS
Input Data, Input Clock Pulse Width	t_{DW}	–	120	–	–	nS
SR Data Setup Time	t_{SS}	–	120	–	–	nS
SR Data Hold Time	t_{SH}	–	120	–	–	nS
Data Delay Time	t_{DR}	C_L=15pF	–	–	t_{CY}+300	nS
Data Delay Time at Mode Switching	t_{DCR}	C_L=15pF	–	–	7/8 t_{CY}+300	nS
Data Delay Time at OPT Instruction	t_{DI1}	C_L=15pF	–	–	6/8 t_{CY}+300	nS
Data Delay Time at OPT Instruction	t_{DI2}	C_L=15pF	–	–	7/8 t_{CY}+300	nS
CT/TM Data Delay Time using TBC Clock	t_{CT}/t_{TT}	C_L=15pF	–	–	2/8 t_{CY}+360	nS
SR/TM Data Delay Time using PR Clock	t_{SR}/t_{TR}	C_L=15pF	–	–	t_{CY}+480	nS
CT Data Delay Time using PR Clock	t_{CR}	C_L=15pF	–	–	10/8 t_{CY}+480	nS
CT Data Delay Time using External Clock	t_{CP}	C_L=15pF	–	–	2/8 t_{CY}+360	nS
SR/TM Data Delay Time using External Clock	t_{SP}/t_{TP}	C_L=50pF	–	–	360	nS
SR Clock Invalid Time	t_{SINH}	–	2/8 t_{CY}	–	–	nS
INT Invalid Time	t_{IINH}	–	1/8 t_{CY}	–	–	nS

TIMING CHARTS

Output Condition



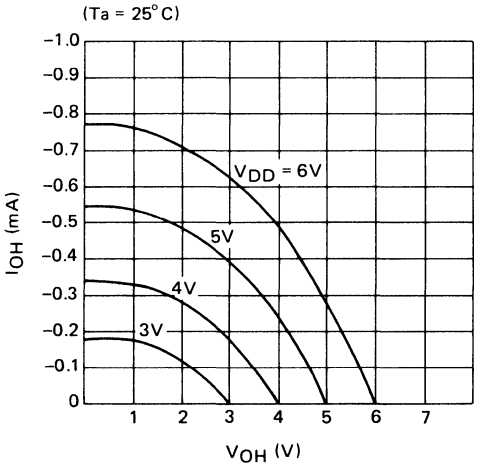
Input Condition



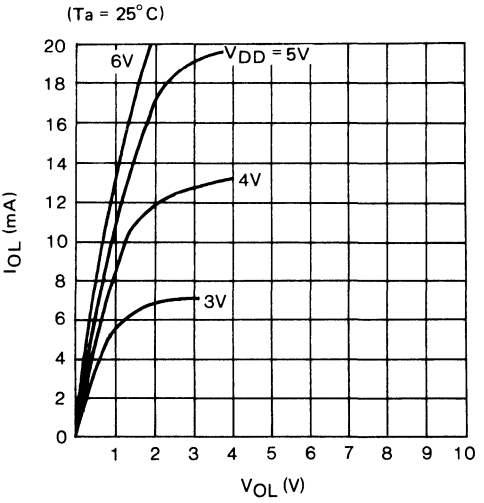
$t_{S\ INH}$: P_{01} (SR clock) INH period during LMSR INST.
(Note: P_{01} is used for clock of SR)

$t_{I\ INH}$: P_{20} (interrupt) INH period during RPB and RPBD INST.

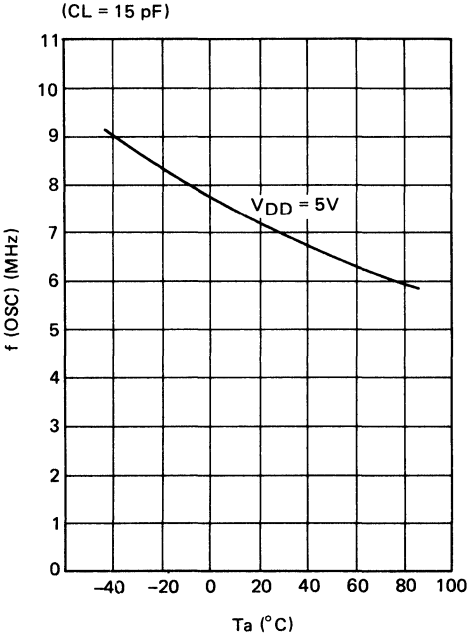
TYP. Current vs Voltage for High State Output (I_{OH}) (V_{OH})



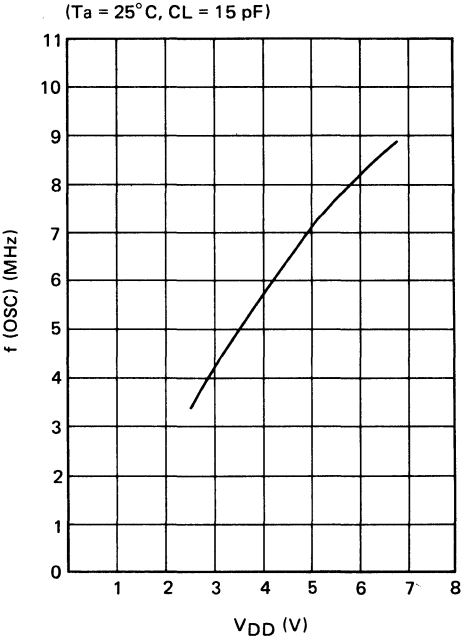
TYP. Current vs. Voltage for Low State Output (I_{OL}) (V_{OL})



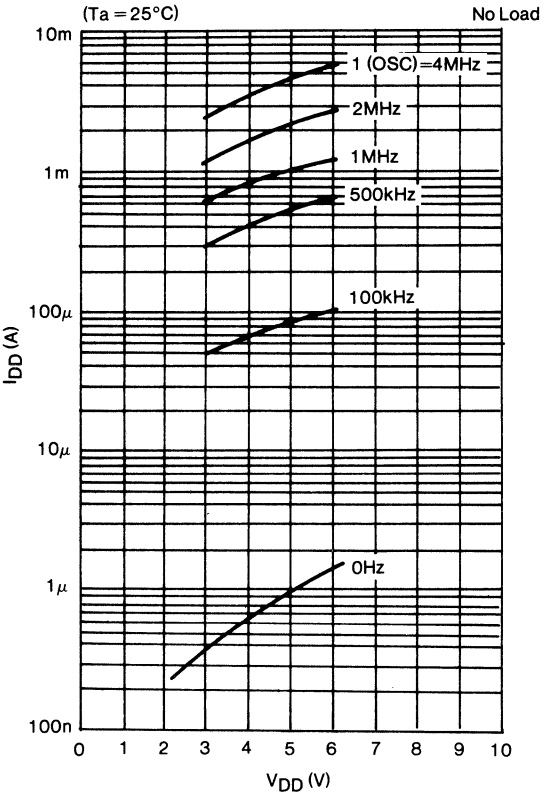
TYP. Maximum Oscillator Frequency vs Temperature f(OSC) (Ta)



TYP. Maximum Oscillator Frequency vs. Supply Voltage f(OSC) (V_{DD})



TYP. Supply Current vs. Supply Voltage
(I_{DD}) (V_{DD})



MSM6404VS

MSM6404 PIGGY BACK

GENERAL DESCRIPTION

The MSM6404VS is a device whose built-in ROM is replaced by external EPROM using the piggy-back method. This device can be used for the evaluation of program.

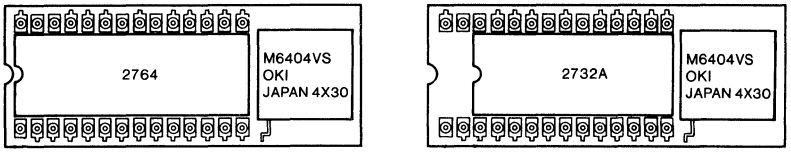
FEATURES

- Supply Voltage: 5V ± 5%
- Frequency: DC ~ 4.2 MHz
- Operating Temperature: 0 ~ 70°C
- 42 pin ceramic piggy back (ADIP42-C-600)

Note: There are a few differences in the electrical characteristics of this chip and the evaluation chip. Please refer to next page for the detail.

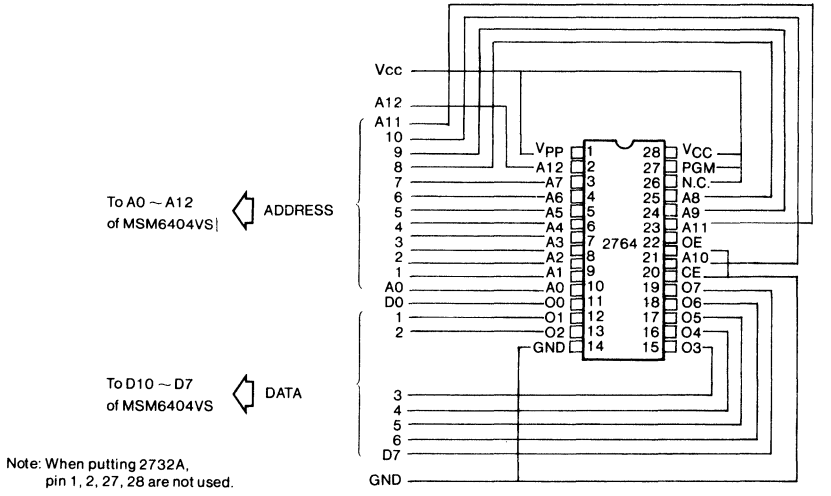
PUTTING METHOD OF ROM

Please refer to drawing below.



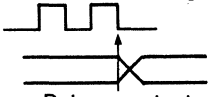
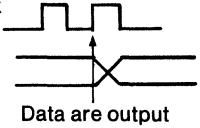
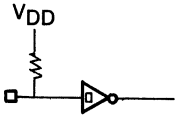
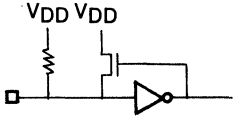
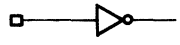
PIN CONFIGURATION

Pin Connection between MSM6404VS and EPROM



Note: When putting 2732A, pin 1, 2, 27, 28 are not used.

DIFFERENCES BETWEEN MSM6404 AND MSM6404VS (PIGGY-BACK)

Item	6404	6404VS (Piggy-Back)
1. Port initialization during reset	Port P0, 1, 3 are set to "1" and port 2, 4, 5, 6, 7, 8 are reset to "0" directly by signal put into the RESET.	Port P0, 1, 3 are set to "1" and port 2, 4, 5, 6,7, 8 are reset to "0" during reset cycle being executed.
2. Timer Operation	After being reset, timer stops counting until data are set in it.	It is undecidable whether the timer starts counting or not after being reset. Therefore, the timer should be initialized by software.
3. Shift registor	Serial Out F/F (SOF/F) is set to "0" after being reset.	It is undecidable whether Serial Out F/F (SOF/F) is set to "0" or "1" after being reset. Therefore Serial Out F/F should be initialized by software.
4. Port input/output timing	Internal clock  Data are input at this moment	Internal clock  Data are input at this moment
	Internal clock Synchronized with falling edge  Data are output at this moment	Internal clock  Data are output at this moment
5. Port input/output (maracteris-tics)	TTL F0=1 (I_{OL} = 1.6 mA 0.4V)	LSTTL F0=1 (I_{OL} = 0.4 mA 0.4V)
	P20~3  V_{DD}	P20~3  V_{DD} V_{DD}
	TTL compatible input P00~P83  (Except P20~3)	CMOS input P00~P83  (Except P20~3)
Available ROM capacity	4K byte	Accessible up to 8K byte
LJP, LCAL instruction	Not available	Available

MSM6408

HIGH-SPEED 4-BIT SINGLE CHIP MICROCONTROLLER

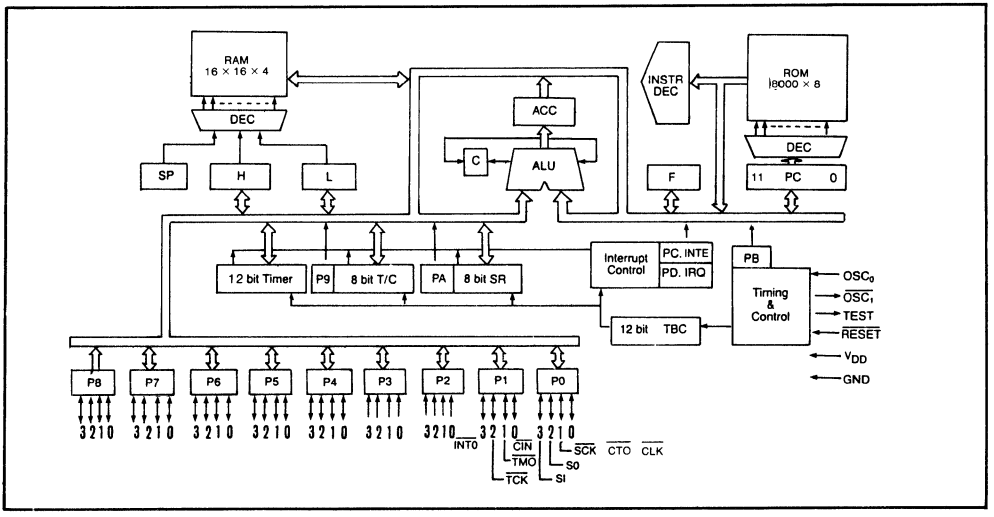
GENERAL DESCRIPTION

The OKI MSM6408 microcontroller is a low power, high-performance single-chip device implemented in complementary metal oxide semiconductor technology. 64K bits of mask program ROM, 1024 bits of data RAM, 36 Input/Output lines, a programmable timer/event-counter, and oscillator are integrated onto one chip. Program memory is byte wide and data-paths are organized in 4-bit nibbles. RAM and I/O lines are bit addressable. 122 instructions include binary, BCD operations; bit set, reset, test; 8-bit I/O; relative jumps; multifunctional instructional (increment, modify, skip) 8-bit wide table output; subroutine call and return.

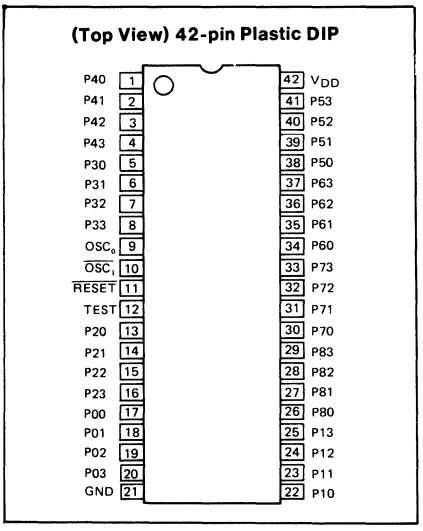
FEATURES

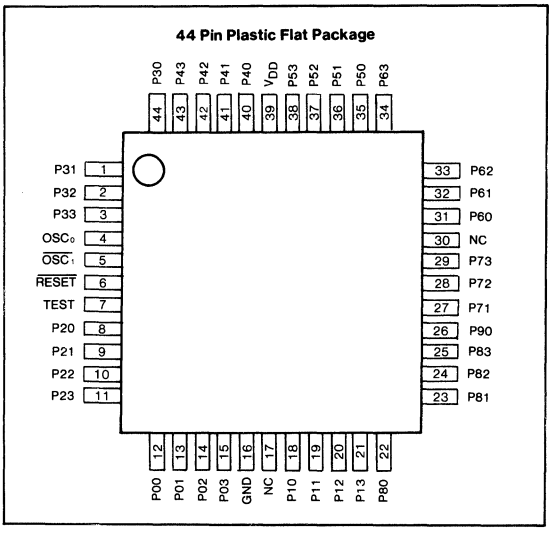
- 8096 × 8 MASK ROM
  An evaluation board is available for up to 8K × 8.
- 256 × 4 RAM (including the stack area)
- 9 × 4 Ports, 36 I/O lines
  4 lines for input ports having a latch, and the other 32 lines for bit operation are available.
- Three built-in counters
  12-bit time-base counter
  12-bit programmable timer
  8-bit high-speed programmable timer/event counter
- Built-in 8-bit serial I/O register (with 3-bit counter)
- Five interrupts with five priority levels (4 internal, 1 external)
- 32 stack levels (in RAM)
- LED direct drive available (8mA x 5 ports at the same time)
- Power down features
- Instruction execution time
  1.0 μs 4.0 MHz clock
- Instruction systems suitable for control
- 122 instructions
- Full static operation
- Low power consumption
  TYP 0.4μW at V_{DD}=2V
  TYP 5μW at V_{DD}=5V 0Hz clock
- 5V single power supply, 42-pin DIP or 44-pin FLAT
- Package:
  42 pin plastic DIP (DIP42-P-600)
  44 pin plastic QFP (QFP44-P-910-K)
  44 pin plastic QFP (QFP44-P-910-VIK)

BLOCK DIAGRAM



PIN CONFIGURATION





PIN DESCRIPTION

Pin name	Input/output	Function	When reset
P00 P01 / SCK P02 / SO P03 / SI	Input/output	4-bit input/output port. P01 to P03 are also used as serial interface terminals.	"1"
P10 / CIN P11 / TMO P12 / TCK P13	Input/output	4-bit input port with latch. Built-in pull up register for all bit input.	"1"
P20 / INT P21 P22 P23	Input	4-bit input port with a latch. P20 is shared with INT input. (Fall trigger input) P21 ~ 23 are level input. Built-in pull up register for all bit input.	The latch is reset.
P30–33	Input/output	4-bit input/output port	"1"
P40–43	Input/output	4-bit input/output port (8-bit output port)	"0"
P50–53	Input/output	4-bit input/output port (8-bit output port)	"0"
P60–63	Input/output	4-bit input/output port *1	"0"
P70–73	Input/output	4-bit input/output port	"0"
P80–83	Input/output	4-bit input/output port	"0"
OSC0 OSC1	Input/output	X'tal connection terminal for system clock oscillation	Oscillation wave
TEST	Output	(Test terminal for Maker)	Pulse output
RESET	Input	System reset input terminal	
VDD GND		Power source voltage supply	

Note: When each port is used for output, it is possible to drive one TTL (one input).
*1 Can be made as a port dedicated to input (mask option).

INSTRUCTION LIST

	Mnemonic		Description	Code	Byte	Cycle
Load, Push, Pop	LAI	n	A ← n	9n	1	1
	LLI	n	L ← n	8n	1	1
	LHLI	nn	HL ← nn	15nn	2	2
	LMI	nn	M(w) ← nn	14nn	2	2
	LAL		A ← L	21	1	1
	LLA		L ← A	2D	1	1
	LAH		A ← H	22	1	1
	LHA		H ← A	2E	1	1
	LAM		A ← M	38	1	1
	LMA		M ← A	2F	1	1
	LAM+		A ← M, L ← L+1, Skip if L=0	24	1	1
	LAM−		A ← M, L ← L−1, Skip if L=F	25	1	1
	LMA+		M ← A, L ← L+1, Skip if L=0	26	1	1
	LMA−		M ← A, L ← L−1, Skip if L=F	27	1	1
	LAMM	n_2	A ← M, H ← H V n_2	39–3B	1	1
	LAMD	mm	A ← Md	10mm	2	2
	LMAD	mm	Md ← A	11mm	2	2
	LMTD	mm	Md(w) ← T (M(w), A), T=ROM table	19mm	2	3
	LMCT		M(w) ← CT	3E59	2	2
	LCTM		CT ← M(w)	3E51	2	2
	LMSR		M(w) ← SR	3E5A	2	2
	LSRM		SR ← M(w)	3E52	2	2
	LTMM		TM ← (M(w), A)	3E50	2	2
	PUSH		ST ← C, A, H, L, SP ← SP−4	1C	1	3
	POP		C, A, H, L, ← ST SP ← SP+4	1D	1	3
Exchange	X		A ⟷ M	28	1	1
	XM	n_2	A ⟷ M, H ← H V n_2	29–2B	1	1
	X+		A ⟷ M, L ← L+1, Skip if L=0	3C	1	1
	X−		A ⟷ M, L ← L−1, Skip if L=F	2C	1	1
Increment/ Decrement	INA		A ← A+1, Skip if A=0	30	1	1
	INM		A ← M+1, Skip if M=0	33	1	1
	INL		L ← L+1, Skip if L =0	31	1	1

INSTRUCTION LIST (Continued)

	Mnemonic		Description	Code	Byte	Cycle
Increment/Decrement	INH		H ← H+1, Skip if M = 0	32	1	1
	INMD	mm	Md ← Md+1, Skip if Md = 0	12mm	2	2
	DCA		A ← A−1, Skip if A = F	34	1	1
	DCM		M ← M−1, Skip if M = F	37	1	1
	DCL		L ← L−1, Skip if L = F	35	1	1
	DCH		H ← H−1, Skip if H = F	36	1	1
	DCMD	mm	Md ← Md−1, Skip if Md = F	13mm	2	2
Arithmetic	ADS		A ← A+M, Skip if Cy = 1	02	1	1
	ADCS		A, C ← A+M+C, Skip if Cy = 1	01	1	1
	ADC		A, C ← A+M+C	03	1	1
	AIS	n	A ← A+n, Skip if Cy = 1	3E4n	2	2
	DAA		A ← A+6	06	1	1
	DAS		A ← A+10	0A	1	1
	AND		A ← A ∨ M	0D	1	1
	OR		A ← A ∨ M	05	1	1
	EOR		A ← A ∨ M	04	1	1
	CMA		A ← $\overline{A}$	0B	1	1
	CIA		A ← $\overline{A}$+1	0C	1	1
	RAL		Rotate Left with C	0E	1	1
	RAR		Rotate Right with C	0F	1	1
	TC		Skip if C = 1	09	1	1
	SC		C ← 1	07	1	1
	RC		C ← 0	08	1	1
Compare	CAI	n	Skip if A = n	3E0n	2	2
	CLI	n	Skip if L = n	3E2n	2	2
	CPI	p, n	Skip if P_p = n	17pn	2	2
	CMI	n	Skip if M = n	3E1n	2	2
	CAM		Skip if A = M	16	1	1
Bit operation	TAB	n_2	Skip if Abit (n_2) = 1	54–57	1	1
	RAB	n_2	Abit (n_2) ← 0	64-67	1	1
	SAB	n_2	Abit (n_2) ← 1	74–77	1	1
	TMB	n_2	Skip if Mbit (n_2) = 1	58–5B	1	1
	RMB	n_2	Mbit (n_2) ← 0	68–6B	1	1

INSTRUCTION LIST (Continued)

	Mnemonic		Description	Code	Byte	Cycle
Bit operation	SMB	n	Mbit (n_2) ←1	78–7B	1	1
	TFB	n_2	Skip if Fbit (n_2) =1	5C–5F	1	1
	RFB	n_2	Fbit (n_2) ←0	6C-6F	1	1
	SFB	n_2	Fbit (n_2) ←1	7C–7F	1	1
	TPB	n_2	Skip if Pbit (n_2) =1	50–53	1	1
	RPB	n_2	Pbit (n_2) ←0	60–63	1	1
	SPB	n	Pbit (n_2) ←1	70–73	1	1
	TPBD	p n_2	Skip if $P_p$bit (n_2) =1	3D $p_{0\sim3}$	2	2
	RPBD	p n_2	$P_p$bit (n_2) =0	3D $p_{4\sim7}$	2	2
	SPBD	p n_2	$P_p$bit (n_2) =1	3D $p_{8\sim B}$	2	2
Interrupt	MEI		MEIF ←1	3E60	2	2
	MDI		MEIF ←0	3E61	2	2
	EITB		EITBF ←1	3DC9	2	2
	EITM		EITMF ←1	3DCA	2	2
	EICT		EICTF ←1	3DCB	2	2
	EIEX		EIEXF ←1	3DC8	2	2
	DITB		EITBF ←0	3DC5	2	2
	DITM		EITMF ←0	3DC6	2	2
	DICT		EICTF ←0	3DC7	2	2
	DIEX		EIEXF ←0	3DC4	2	2
	TITB		Skip If EITBF =1	3DC1	2	2
	TITM		Skip If EITMF =1	3DC2	2	2
	TICT		Skip If EICTF =1	3DC3	2	2
	TIEX		Skip If EIEXF =1	3DC0	2	2
	TQEX		Skip If IRQEX =1	3D20	2	2
	TQTB		Skip If IRQTB =1	3DD0	2	2
	TQTM		Skip If IRQTM =1	3DD1	2	2
	TQCT		Skip If IRQCT =1	3DD2	2	2
	TQSR		Skip If IRQSR =1	3DD3	2	2
	RQEX		IRQ EX ←0	3D24	2	2
	RQTB		IRQ TB ←0	3DD4	2	2
	RQTM		IRQ TM ←0	3DD5	2	2
	RQCT		IRQ CT ←0	3DD6	2	2
	RQSR		IRQ SR ←0	3DD7	2	2

INSTRUCTION LIST (Continued)

	Mnemonic		Description	Code	Byte	Cycle
Shift resistor / Counter	ECT		CTF ←1 (start)	3DBB	2	2
	ESR		SRF ←1 (start)	3DBA	2	2
	DCT		CTF ←0 (stop)	3DB7	2	2
	DSR		SRF ←0 (stop)	3DB6	2	2
	TCT		Skip If CTF =1	3DB3	2	2
	TSR		Skip If SRF =1	3DB2	2	2
Branch	JCP	a_6	PC ←a_6	C0~FF	1	1
	JP	a_{12}	PC ←a_{12}	$4a_{12}$	2	2
	LJP	a_{13}	PC ←a_{13}	3F	3	4
	CZP	a	ST ←PC+1, PC ←2a, SP ←SP−4	Ba	1	4
	CAL	a_{12}	ST ←PC+2, PC ←a_{12}, SP ←SP−4	Aa_{12}	2	4
	RT		PC ←ST, SP ←SP+4	IE	1	4
	RTS		PC ←ST, SP ←SP+4, Skip unconditional	IF	1	4
	JA		PC ←(PC ←A)+1	IA	1	1
	JM		PC ←(M(w), A)	IB	1	2
Input/Output	IP		A ←P	20	1	1
	IPD	p	A ←Pp	3DpD	2	2
	OP		P ←A	23	1	1
	OPD	p	Pp ←A	3DpC	2	2
CPU control	NOP		No Operation	00	1	1
	HALT		Halt CPU	3DB8	2	2
	STOP		Stop Clock	3DB9	2	2

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	Ta = 25°C	−0.3 to 7	V
Input Voltage	V_I		−0.3 to V_{DD}	V
Output Voltage	V_O		−0.3 to V_{DD}	V
Power Dissipation	P_D	Ta = 25°C per package	200 max.	mW
		Ta = 25°C per output	50 max.	mW
Storage Temperature	Tstg	–	−55 to +150	°C

OPERATING RANGE

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	$f_{(OSC)} \leqq 1$ MHz	3 to 6	V
		$f_{(OSC)} \leqq 4.0$ MHz	4.5 to 5.5	V
Data-Hold Voltage	V_{DDH}	f (OSC) = 0 Hz	2 to 6	V
Operating Temperature	T_{OP}	–	−40 to +85	°C
Fan Out	N	MOS Load	15	–
		TTL Load	1	

DC CHARACTERISTICS

(V_{DD} = 5V±10%, Ta = −40 to +85°C)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
"H" Input Voltage*1,*2	V_{IH}	–	2.4	–	V_{DD}	V
"H" Input Voltage*3,*4	V_{IH}	–	3.6	–	V_{DD}	V
"L" Input Voltage	V_{IL}	–	−0.3	–	0.8	V
"H" Output Voltage*1,*5	V_{OH}	I_O = −15μA	4.2	–	–	V
"L" Output Voltage*1	V_{OL}	I_O = 1.6mA	–	–	0.4	V
"L" Output Voltage*5	V_{OL}	I_O = 15μA	–	–	0.4	V
"L" Output Voltage*6	V_{OL}	I_O = 8mA	–	1	2	V
Input Current*3	I_{IH}/I_{IL}	$V_I = V_{DD}/0V$	–	–	15/−15	μA
Input Current*2,*4	I_{IH}/I_{IL}	$V_I = V_{DD}/0V$	–	–	1/−30	μA
"H" Output Current*1	I_{OH}	V_O = 2.4V	−0.1	–	–	mA
"H" Output Current*1	I_{OH}	V_O = 0.4V	–	–	−1.2	mA
Input Capacity	C_I	f=1MHz Ta=25°C	–	5	–	pF
Output Capacity	C_O		–	7	–	
Current Dissipation (when stop condition)	I_{DDS}	V_{DD}=2V, no load Ta=25°C	–	0.2	5	μA
		No load	–	1	100	μA
Current Dissipation	I_{DD}	Quarts oscillation f=4 MHz, no load	–	6	12	mA

*1 Applied to P0, P1, P3, P4, P5, P6, P7 and P8

*2 Applied to P2 *3 Applied to OSC_0 *4 Applied to $\overline{RESET}$ *5 Applied to $\overline{OSC_1}$

*6 In using LED, total output current should be within the limit of Power dissipitation in "Absolute Maximum Rating."

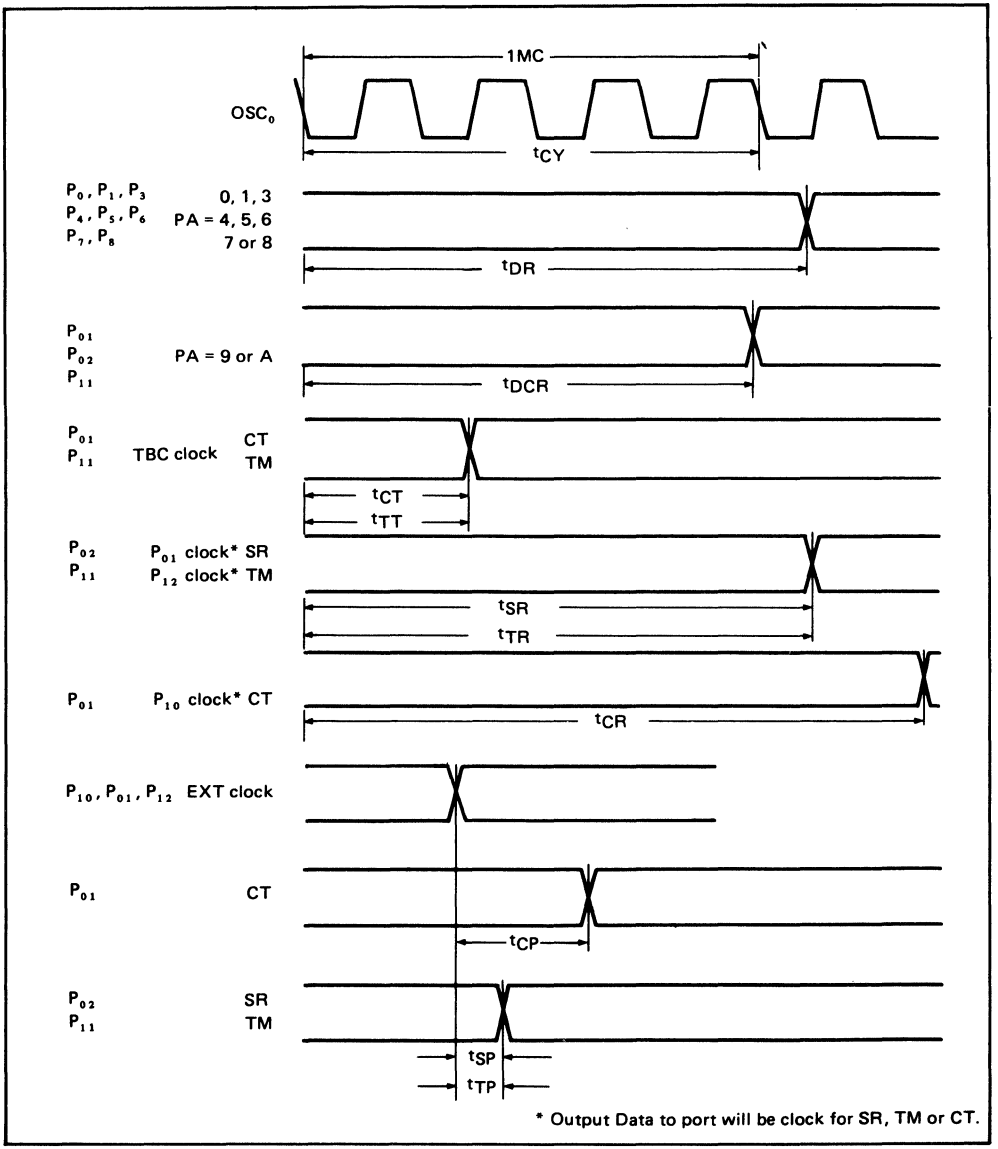
AC CHARACTERISTICS

(V_{DD} = 5V±10%, Ta = −40 to +85°C)

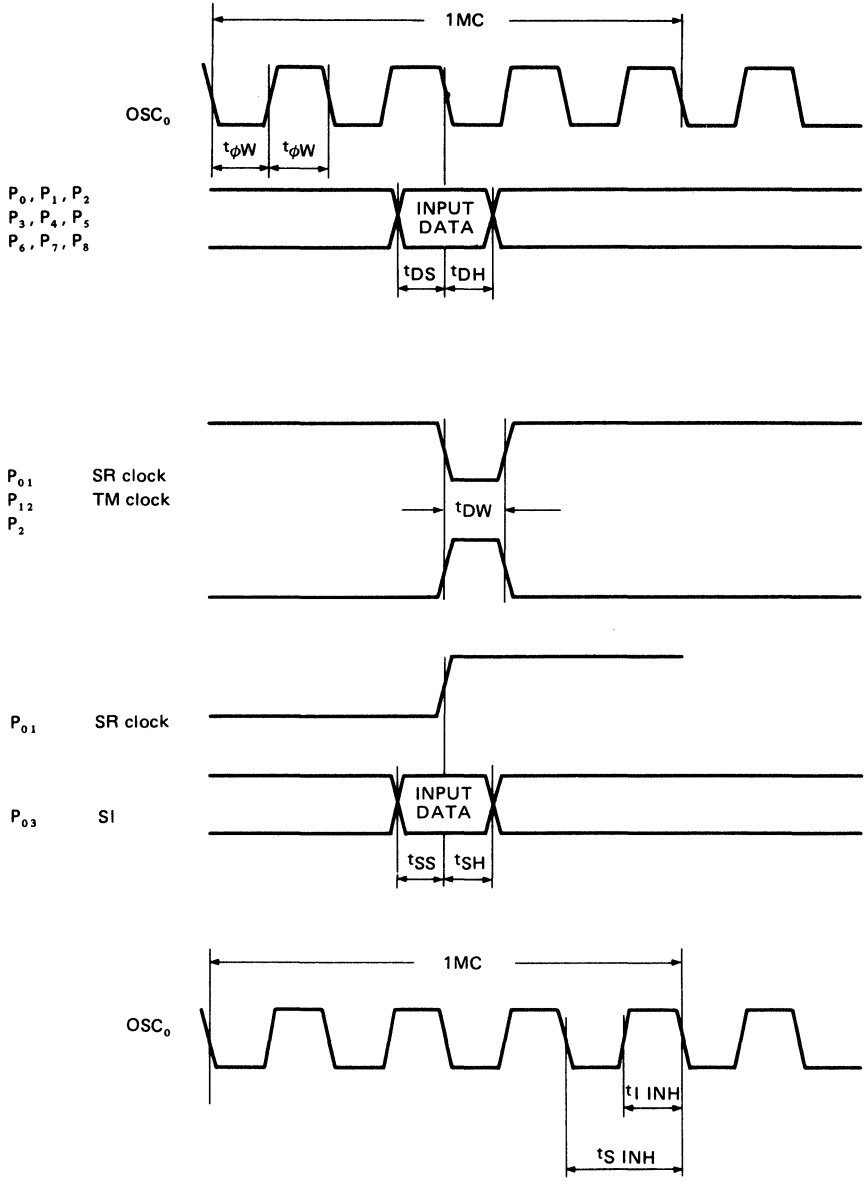
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Clock Pulse Width Clock (OSC)	$t_{\phi W}$	–	125	–	–	nS
Cycle Time	t_{CY}	–	1	–	–	μS
Input Data Setup Time	t_{DS}	–	120	–	–	nS
Input Data Hold Time	t_{DH}	–	120	–	–	nS
Input Data, Input Clock Pulse Width	t_{DW}	–	120	–	–	nS
SR Data Setup Time	t_{SS}	–	120	–	–	nS
SR Data Hold Time	t_{SH}	–	120	–	–	nS
Data Delay Time	t_{DR}	C_L=15pF	–	–	t_{CY}+300	nS
Data Delay Time at Mode Switching	t_{DCR}	C_L=15pF	–	–	7/8 t_{CY}+300	nS
CT/TM Data Delay Time using TBC Clock	t_{CT}/t_{TT}	C_L=15pF	–	–	2/8 t_{CY}+360	nS
SR/TM Data Delay Time using PR Clock	t_{SR}/t_{TR}	C_L=15pF	–	–	t_{CY}+480	nS
CT Data Delay Time using PR Clock	t_{CR}	C_L=15pF	–	–	10/8 t_{CY}+480	nS
CT Data Delay Time using External Clock	t_{CP}	C_L=15pF	–	–	2/8 t_{CY}+360	nS
SR/TM Data Delay Time using External Clock	t_{SP}/t_{TP}	C_L=50pF	–	–	360	nS
SR Clock Invalid Time	t_{SINH}	–	2/8 t_{CY}	–	–	nS
INT Invalid Time	t_{IINH}	–	1/8 t_{CY}	–	–	nS

TIMING CHARTS

Output Condition



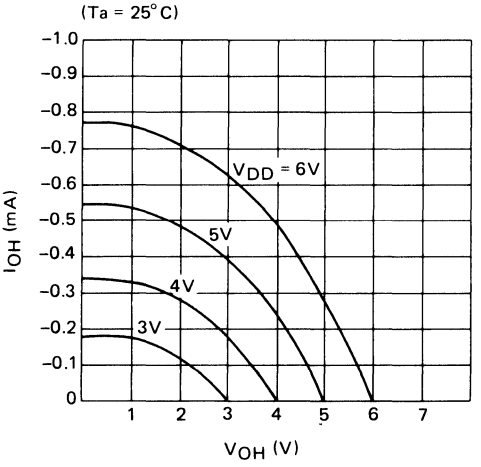
Input Condition



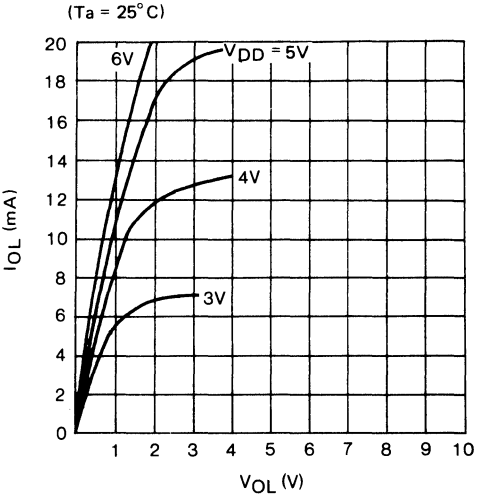
$t_{S\ INH}$: P_{01} (SR clock) INH period during LMSR INST.
(Note: P_{01} is used for clock of SR)

$t_{I\ INH}$: P_{20} (interrupt) INH period during RPB and RPBD INST.

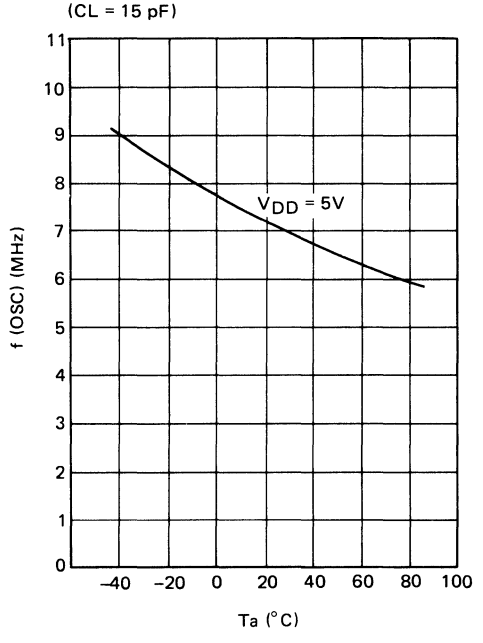
TYP. Current vs Voltage for High State Output (I_{OH}) (V_{OH})



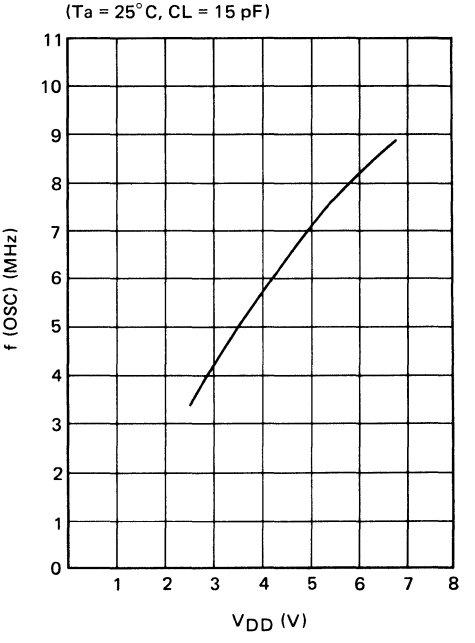
TYP. Current vs. Voltage for Low State Output (I_{OL}) (V_{OL})



TYP. Maximum Oscillator Frequency vs Temperature
f(OSC) (Ta)



TYP. Maximum Oscillator Frequency vs. Supply Voltage
f(OSC) (V_{DD})



OKI semiconductor

MSM6411

CMOS 4-BIT SINGLE CHIP MICROCONTROLLER

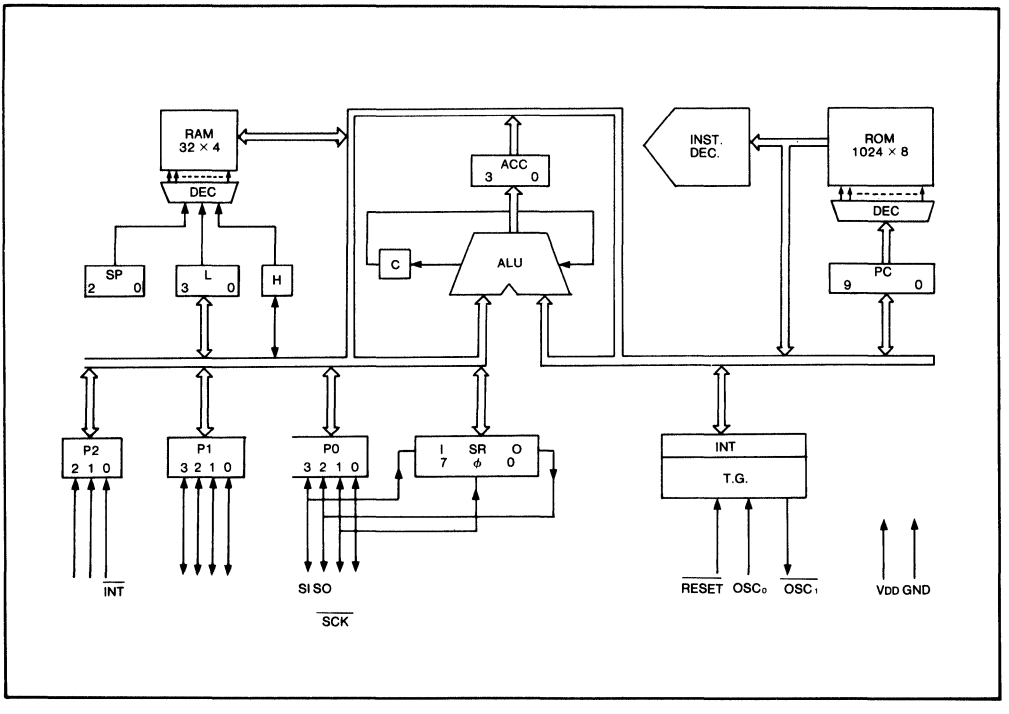
GENERAL DESCRIPTION

The OKI MSM6411 microcontroller is a low-power, high-performance single-chip device implemented in complementary metal oxide semiconductor technology. 1024 × 8 bits of program ROM, 32 × 4 bits of data RAM, 11 Input/Output lines and oscillator. Program memory is byte wide and data paths are organized as 4-bit wide. 63 instructions include binary, logical operations; bit set, reset, test; multifunctional.

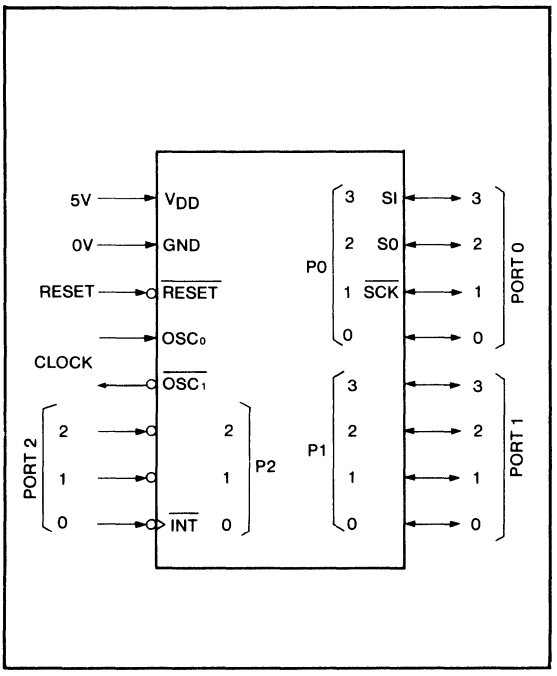
FEATURES

- 1024 × 8 Internal ROM
- 32 × 4 Internal RAM
- 11 I/O Lines (8 I/O Lines, 3 Input Lines)
- 8-bit serial I/O Register
- 2 Interrupt Levels
- 8 Stack Levels
- LED direct drive available (8mA x 5 ports at the same time)
- 952 ns 4.2MHz (V_{DD} 5V±10%)
- 63 Instructions
- Self-Contained Oscillator
- −40 to +85°C Operating Temperature
- 3 to 6V Operating V_{DD}
- Low Power Consumption 5 μW Typical (STOP, V_{DD} = 5V, no load)
- Mask Option Crystal (Ceramic)/CR Oscillator
- Package:
  16 pin plastic DIP (DIP16-P-300)
  24 pin plastic SOP (SOP24-P-430-K)

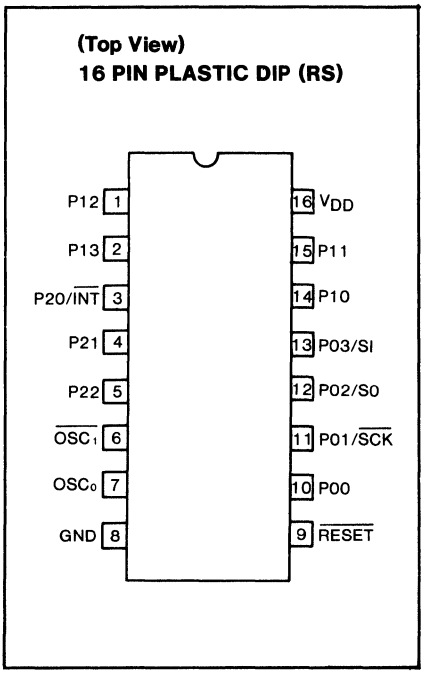
FUNCTIONAL BLOCK DIAGRAM



LOGIC SYMBOL



PIN CONFIGURATION



PIN DESCRIPTION

Designation	Input/Output	Pin No.	Function	Reset
P00 P01/SCK P02/SO P03/SI	Input/Output	10 11 12 13	4 Bits I/O port. P01 to P03 are used both I/O port and terminal of shift resister	"1"
P10 P11 P12 P13	Input/Output	14 15 1 2	4 Bits I/O port	"1"
P20/INT P21 P22	Input	3 4 5	3 Bits input port with latch. P20 is used as both input port and input terminal of INT (input of falling edge trigger).	Latch is reset. ("0")
OSCo	Input	7	Input terminal of system clock. Oscillation circuit consists of OSCo and OSC1.	Clock pulse In
OSC1	Output	6	Oscillation circuit consists of OSCo and OSC1.	
RESET	Input	9	Input terminal of system reset	
VDD GND	Input	16 8	Main power source and circuit GND potential.	

FUNCTIONAL DESCRIPTION

ROM

ROM is organized in 1024 words by 8 bits. It is used to stored developed application programs (instructions). It is addressed by the program counter (PC).

PC

The PC consists of a 10-bit binary counter and is used to address ROM.

Stack and Stack Pointer

An interrupt or CAL instruction causes the contents of the PC to be saved in the stack. The PC is restored from the stack by RT instruction.

All RAM locations (up to 8 levels) are available as the stack. Note that four words of RAM are used for each level.

The stack pointer is a 5-bit up-down counter that points to the address of the next stack to be used. It allows the RAM locations to be used as a push-down stack.

RAM

RAM consists of up to 32 words 4 bits wide. It is addressed by the H- and L-registers or by the contents of the second byte of an instruction.

L-REGISTER

A 4-bit register which specifies RAM locations A3-A0.

H-REGISTER

A 1-bit register which specifies RAM location A4.

ALU

A 4-bit logic circuit that provides arithmetic and logical operations.

ACC

Consisting of a 4-bit register, the accumulator holds the result of operations or the data present on ports.

C FLAG

The flag that holds a carry generated from the result of operations.

INPUT/OUTPUT Ports (2 × 4 bit)

Organized into 4 bits, 2 ports are provided for effecting and controlling data transfer to and from an external source. The ports are selected by codes included in instructions.

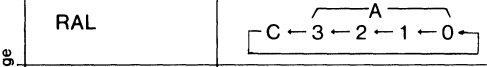
Input Ports (1 × 4 bit)

Contained port 2 (P2), which is an input port with latching function. P20 is set at falling edge of the input signal P21 and P22 are set at "0" level inputs. Also, P20 is used as an interrupt request flag. When P20 is set and an interrupt operation occurs, it is automatically reset.

TIMING CONTROL (TC)

A 0 level on the $\overline{\text{RESET}}$ pin for longer than 2 machine cycles initializes the internal circuitry.

INSTRUCTION LIST

	Mnemonic		Description	Code	Byte	Cycle
Load, Push, Pop	LAI	n	A ← n	90 – 9F	1	1
	LLI	n	L ← n	80 – 8F	1	1
	LHLI	nn	HL ← nn	15nn	2	2
	LAL		A ← L	21	1	1
	LLA		L ← A	2D	1	1
	LAM		A ← M	38	1	1
	LMA		M ← A	2F	1	1
	LAMD	mm	A ← Md	10mm	2	2
	LMAD	mm	Md ← A	11mm	2	2
	LMSR		M(w) ← SR	3E5A	2	2
	LSRM		SR ← M(w)	3E52	2	2
	PUSH		ST ← C, A, H, L, SP ← SP−1	1C	1	3
	POP		C, A, H, L ← ST, SP ← SP+1	1D	1	3
Input Output	IPD	p	A ← Pp	3DpD	2	2
	OPD	p	Pp ← A	3DpC	2	2
Arithmetic	ADS		A ← A+M, SKIP IF Cy = "1"	02	1	1
	ADC		C, A ← C+A+M	03	1	1
	AIS	n	A ← A+n, SKIP IF Cy = "1"	3E4n	2	2
	DAS		A ← A+10	0A	1	1
	AND		A ← A ∧ M	0D	1	1
	EOR		A ← A ∀ M	04	1	1
	CMA		A ← $\overline{A}$	0B	1	1
	CAM		SKIP IF A = M	16	1	1
	SC		C ← "1"	07	1	1
	RC		C ← "0"	08	1	1
	TC		SKIP IF C = "1"	09	1	1
	RAL			0E	1	1
Ex-change	X		A ⟷ M	28	1	1
Increment/ Decrement	INL		L ← L+1, SKIP IF L = "0"	31	1	1
	INH		H ← H+1, SKIP IF H = "0"	32	1	1
	INM		M ← M+1, SKIP IF M = "0"	33	1	1

INSTRUCTION LIST (CONT.)

	Mnemonic		Description	Code	Byte	Cycle
Increment/ Decrement	INMD	mm	Md ← Md+1, SKIP IF Md = "0"	12mm	2	2
	DCL		L ← L−1, SKIP IF L = "F"	35	1	1
	DCH		H ← H−1	36	1	1
	DCM		M ← M−1, SKIP IF M = "F"	37	1	1
Bit operation	TAB	n2	SKIP IF [A bit n2] = "1"	54−57	1	1
	TMB	n2	SKIP IF [M bit n2] = "1"	58-5B	1	1
	RMB	n2	[M bit n2] ← "0"	68−6B	1	1
	SMB	n2	[M bit n2] ← "1"	78−7B	1	1
	TPBD	p, n2	SKIP IF [Pp bit n2] = "1"	3D p0~3	2	2
	RPBD	p, n2	[Pp bit n2] ← "0"	3D p4~7	2	2
	SPBD	p, n2	[Pp bit n2] ← "1"	3D p8~B	2	2
Branch	JCP	a6	PC ← a6	C0~FF	1	1
	JP	a10	PC ← a10	40_43 00~FF	2	2
	CAL	a10	ST ← PC+2, PC ← a10, SP ← SP−1	A0_A3 00~FF	2	4
	RT		PC ← ST, SP ← SP+1	1E	1	4
Interrupt	MEI		MEIF ← "1"	3E60	2	2
	MDI		MEIF ← "0"	3E61	2	2
	EICT		EICTF ← "1"	3DCB	2	2
	EIEX		EIEXF ← "1"	3DC8	2	2
	DICT		EICTF ← "0"	3DC7	2	2
	DIEX		EIEXF ← "0"	3DC4	2	2
	TICT		SKIP IF EICTF = "1"	3DC3	2	2
	TIEX		SKIP IF EIEXF = "1"	3DC0	2	2
	TQEX		SKIP IF IRQEX = "1"	3D20	2	2
	TQSR		SKIP IF IRQSR = "1"	3DD3	2	2
	RQEX		IRQEX ← "0"	3D24	2	2
	RQSR		IRQSR ← "0"	3DD7	2	2
Shift resistor	ESR		SRF ← "1"	3DBA	2	2
	DSR		SRF ← "0"	3DB6	2	2
	TSR		SKIP IF SRF = "1"	3DB2	2	2
CPU control	STOP		STOP CLOCK	3DB9	2	2
	NOP		NO OPERATION	00	1	1

ABSOLUTE MAXIMUM RATING

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	Ta = 25°C	−0.3 ~ 7	V
Input Voltage	V_I		−0.3 ~ V_{DD}	V
Output Voltage	V_O		−0.3 ~ V_{DD}	V
Power Dissipation	P_D	Ta = 25°C per one package	200 max.	mW
		Ta = 25°C per one output	50 max.	mW
Storage Temperature	Tstg		−55 ~ +150	°C

OPERATING CONDITIONS

Parameter	Symbol	Condition	Limits	Unit
Supply Voltage	V_{DD}	f(OSC) ≦ 1MHz	3 ~ 6	V
		f(OSC) ≦ 4.2MHz	4.5 ~ 5.5	V
Data-Hold Voltage	V_{DDH}	f(OSC) = 0Hz	2 ~ 6	V
Operating Temperature	T_{OP}	–	−40 ~ +85	°C
Fan Out	N	MOS Load	15	–
		TTL Load	1	

DC CHARACTERISTICS

(V_{DD} = 5V±10%, Ta = −40 ~ +85°C)

Parameter	Symbol	Condition	Min.	Typ	Max.	Unit
"H" Input Voltage *1, *2	V_{IH}	———	2.4	–	V_{DD}	V
"H" Input Voltage *3, *4	V_{IH}	———	3.6	–	V_{DD}	V
"L" Input Voltage	V_{IL}	———	−0.3	–	0.8	V
"H" Output Voltage *1, *5	V_{OH}	I_O = −15μA	4.2	–	–	V
"L" Output Voltage *1	V_{OL}	I_O = 1.6mA	–	–	0.4	V
"L" Output Voltage *5	V_{OL}	I_O = 15μA	–	–	0.4	V
"L" Output Voltage*6	V_{OL}	I_O = 8mA	–	1	2	V
Input Current *3	I_{IH}/I_{IL}	$V_I = V_{DD}/0V$	–	–	15/−15	μA
Input Current *2, *4	I_{IH}/I_{IL}	$V_I = V_{DD}/0V$	–	–	1/−30	μA
"H" Output Current *1	I_{OH}	V_O = 2.4V	−0.1	–	–	mA
"H" Output Current *1	I_{OH}	V_O = 0.4V	–	–	−1.2	mA
Input Capacitance	C_I	f = 1MHz, Ta = 25°C	–	5	–	pF
Output Capacitance	C_O		–	7	–	pF
Power Consumption (STOP)	I_{DDS}	V_{DD} = 2V, no load Ta = 25°C	–	0.2	5	μA
		No load	–	1	100	μA
Power Consumption	I_{DD}	Crystal oscillation, No load, 4.2MHz	–	6	12	mA

*1 Applied to P0 and P1.

*2 Applied to P2.

*3 Applied to OSC_0

*4 Applied to $\overline{RESET}$

*5 Applied to $\overline{OSC_1}$

*6 In using LED, total output current should be within the limit of Power dissipitation in "Absolute Maximum Rating."

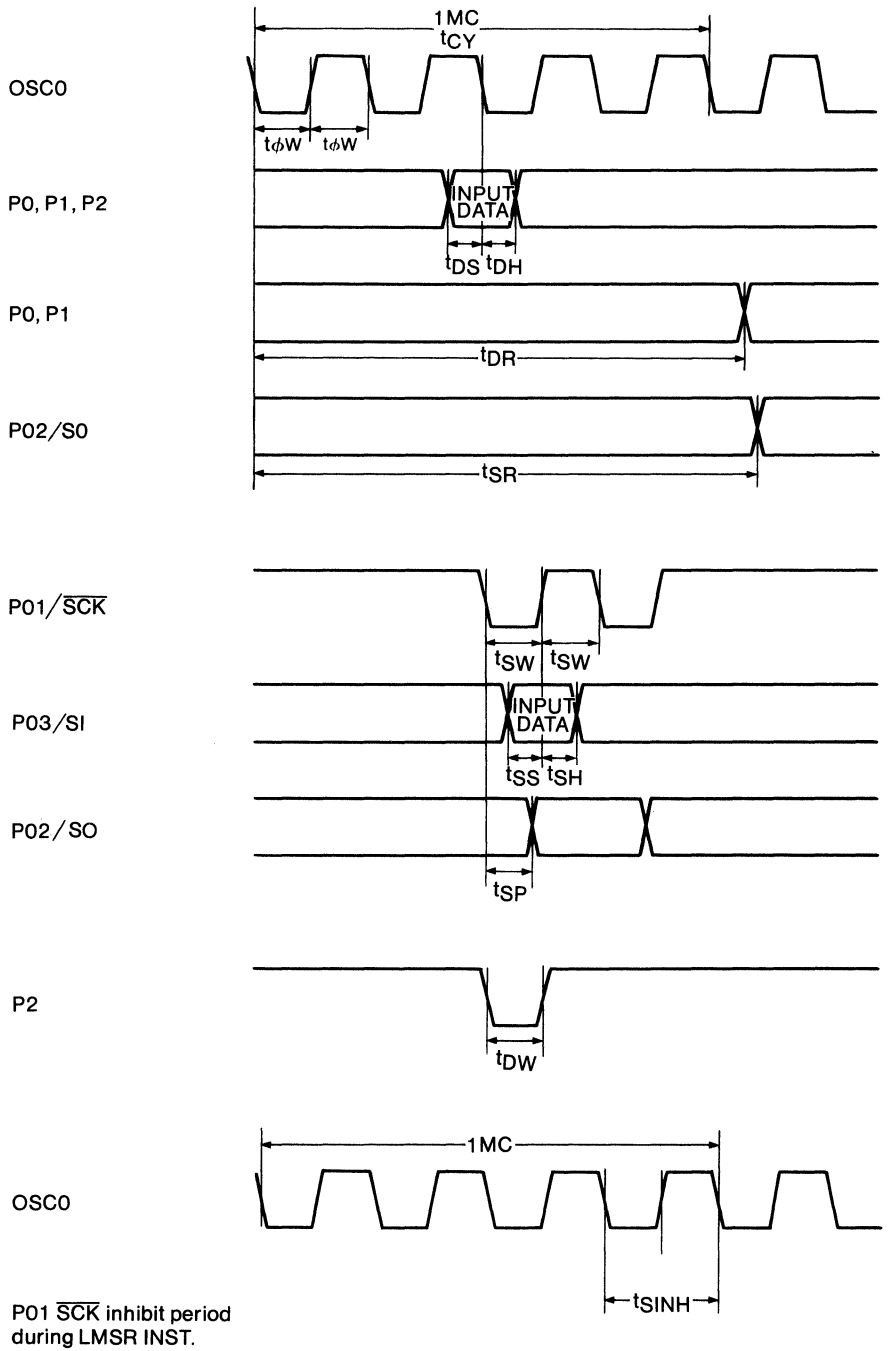
AC CHARACTERISTICS

(V_{DD} = 5V±10%, Ta = −40 ~ +85°C)

Parameter	Symbol	Condition	Min.	Typ	Max.	Unit
Clock (OSC_O) Pulse Width	$t_{\phi W}$	–	119	–	–	ns
Cycle Time	t_{CY}	–	952	–	–	ns
Input Data Setup Time	t_{DS}	–	120	–	–	ns
Input Data Hold Time	t_{DH}	–	120	–	–	ns
Input Data/Input Clock Pulse Width	t_{DW}	–	120	–	–	ns
SR Clock Pulse Width	t_{SW}	–	$t_{\phi W}$	–	–	ns
SR Data Setup Time	t_{SS}	–	120	–	–	ns
SR Data Hold Time	t_{SH}	–	120	–	–	ns
Data Delay Time	t_{DR}	C_L=15pF	–	–	t_{CY} +300	ns
SR Data Delay Time*	t_{SR}	C_L=15pF	–	–	t_{CY} +480	ns
SR Data Delay Time Using External Clock	t_{SP}	C_L=15pF	–	–	360	ns
SR Clock Invalid Time	t_{SINH}	–	2/8 t_{CY}	–	–	ns

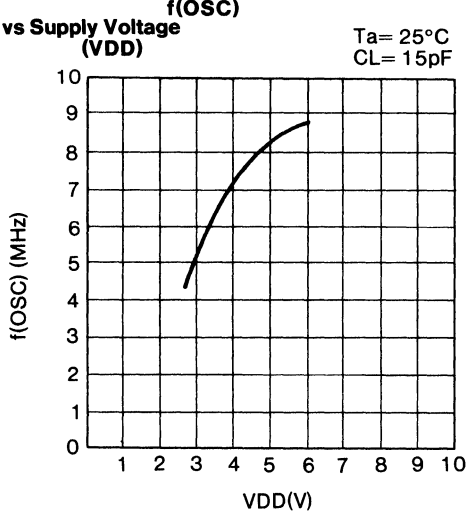
* When SR clock is oscillated by alternate output of "1" or "0" to P01.

TIMING CHART

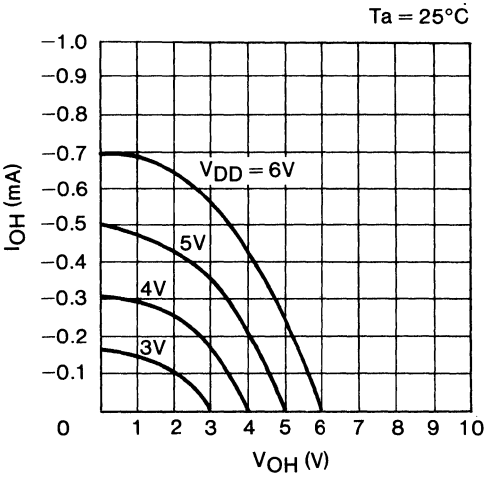


P01 $\overline{SCK}$ inhibit period during LMSR INST.

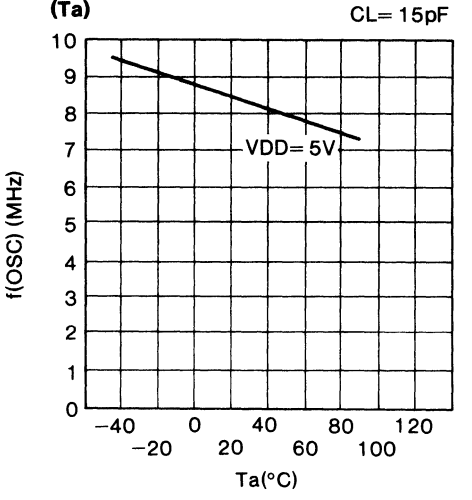
TYP. Maximum Oscillator Frequency f(OSC) vs Supply Voltage (VDD)



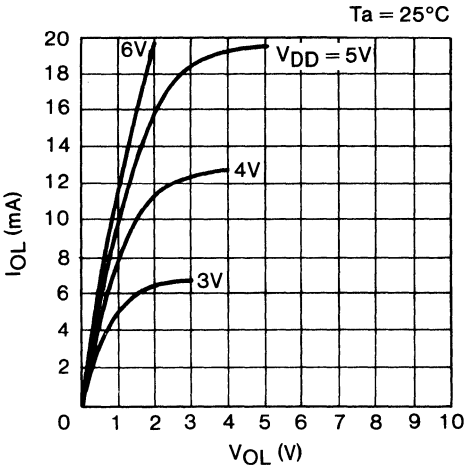
TYP. Current vs Voltage for High State Output (I_{OH}) (V_{OH})



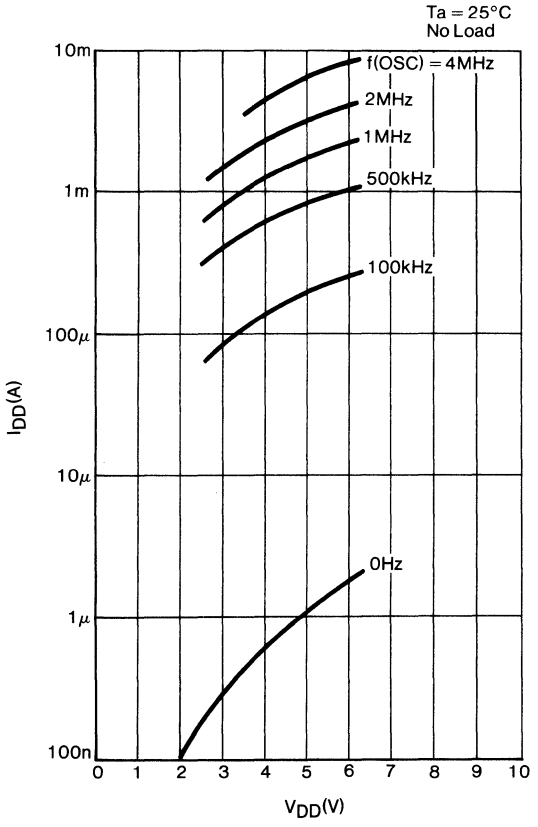
TYP. Maximum Oscillator Frequency f(OSC) vs Temperature (Ta)



TYP. Current vs Voltage for Low State Output (I_{OL}) (V_{OL})



TYP. Supply Current vs Supply Voltage
(I_{DD}) (V_{DD})



MSM6422

CMOS 4-BIT SINGLE CHIP MICROCONTROLLER

GENERAL DESCRIPTION

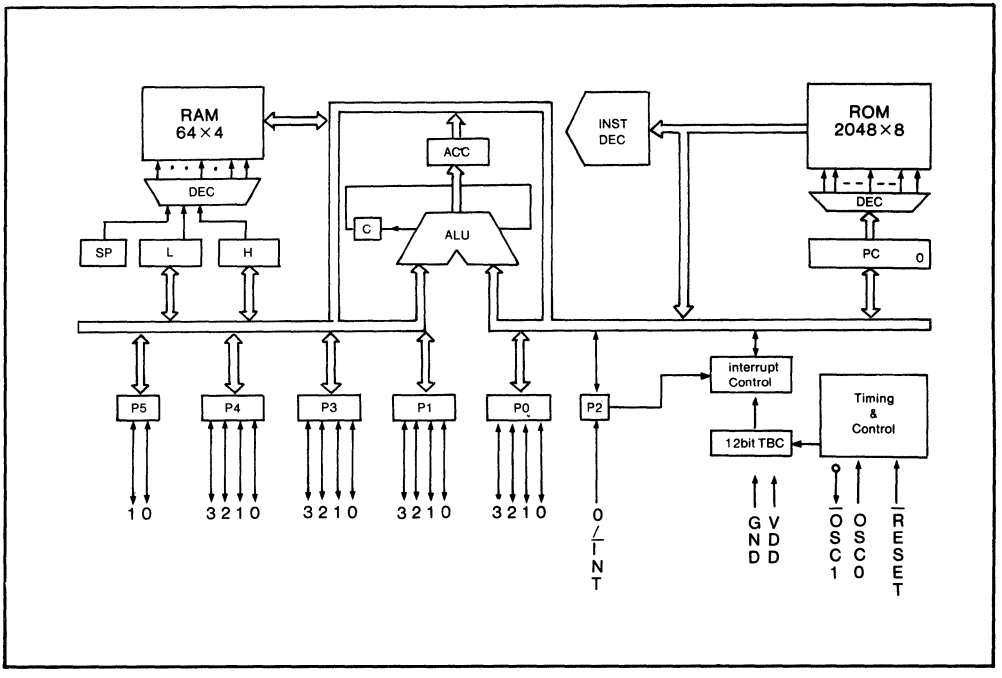
The OKI MSM6422 is a low power, high performance single-chip device implemented in complementary metal oxide semiconductor technology.

Integrated onto a single chip are 16K bits of mask program ROM; 256 bits of data RAM; 18 Input/Output lines and oscillator. Program memory is byte wide and data-paths are organized in 4 bit nibbles. RAM and I/O lines are bit addressable. 63 instructions include Binary, BCD operations; Bit set, Reset, Test; Subroutine call and return.

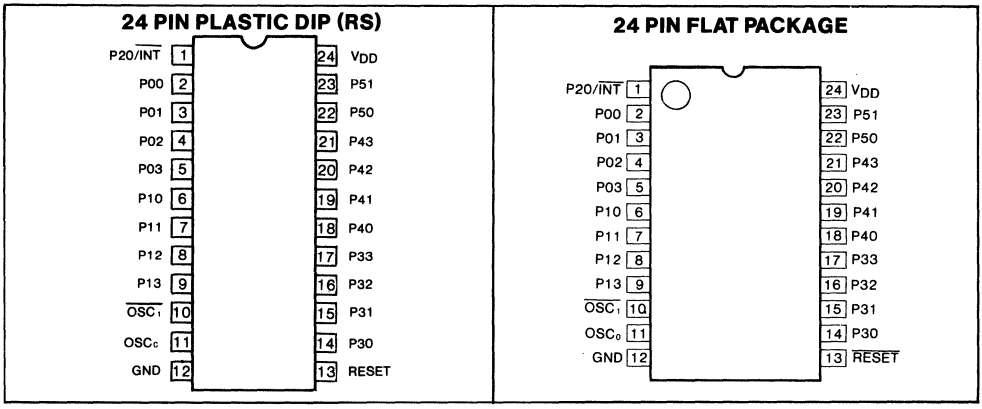
FEATURES

- Low power consumption – 30 mW Typical
- 2048 × 8 Internal ROM
- 64 × 4 Internal RAM
- 18 I/O Lines include 8 Bit Data Bus
- Self-contained Oscillator
- 63 Instructions
- 2 Interrupt Levels
- 16 Stack Levels
- LED direct drive available (8mA x 5 ports at the same time)
- −40 to +85°C Operating Temperature
- 4.5 to 5.5V Operating V_{DD} at 4.2 MHz
  3 to 6V Operating V_{DD} at 1 MHz
- TTL Compatible
- 952ns Cycle Time @ 4.2MHz
  (V_{DD} 5V ±10%)
- Package:
  24 pin plastic DIP (DIP24-P-600)
  24 pin plastic SOP (SOP24-P-430-K)

FUNCTIONAL BLOCK DIAGRAM



LOGIC SYMBOL (Top View)



PIN DESCRIPTION

Terminal symbol	Input/Output	Function	Reset
P00 P01 P02 P03	I/O	4-bit I/O ports (pseudo bidirectional configuration)	"1"
P10 P11 P12 P13	I/O	4-bit I/O ports (pseudo bidirectional configuration)	"1"
P30 P31 P32 P33	I/O	4-bit I/O ports (pseudo bidirectional configuration)	"1"
P40 P41 P42 P43	I/O	4-bit I/O ports (pseudo bidirectional configuration)	"0"
P50 P51	I/O	2-bit I/O ports (pseudo bidirectional configuration)	"0"
P20/$\overline{INT}$	Input	1-bit input port with a latch. Combined use with an interrupt input(falling edge trigger input)	The latch is reset to "0".
OSC_0	Input	System clock (SYSCLK) input terminal. This provides an oscillation circle with $\overline{OSC_1}$ terminal.	–
$\overline{OSC_1}$	Output	System clock output terminal. This provides an oscillation circle with OSC_0 terminal.	–
$\overline{RESET}$	Input	RESET input terminal.	–
V_{DD} GND	Input	Power Supply terminals.	–

FUNCTIONAL DESCRIPTION

Program ROM

Organized into as many as 2,048 words by 8 bits, ROM is used to store developed application programs (instructions). It is addressed by the program counter (PC).

Data RAM

RAM consists of up to 64 words 4 bits wide. It is addressed by the H- and L-registers or by the contents of the second byte of an instruction.

Input/Output Ports

18 input/output port lines are provided for effecting and controlling data transfer to and from an external source. The ports are selected by codes included in instructions.

P20/$\overline{INT}$ PIN (1 line)

A low on this interrupt input pin sets the interrupt request flag. The flag is automatically reset when an external interrupt occurs.
The line can be used as an input port when interrupt is not used.

12-BIT TIME BASE COUNTER (TBC)

The time base counter consists of a 12-bit binary counter. An interrupt request is generated each time an overflow occurs from the division of OSC_0 input signals by 2^{12}.

PROGRAM COUNTER (PC)

The program counter (PC) consists of a 11-bit binary up counter. It is used to address ROM.

STACK AND STACK POINTER (SP)

An interrupt or subroutine call (CAL) causes the contents of the program counter to be saved in the stack. The program counter is restored from the stack by the RT instruction.

All RAM locations (up to 16 levels) are available as the stack. Note that four words of RAM are used for each level.

The stack pointer is a 4-bit up-down counter that points to the address of the next stack to be used. It allows the RAM locations to be used as a push-down stack.

L-REGISTER

A 4-bit register which specifies RAM locations A3-A0.

H-REGISTER

A 4-bit register whose two low-order bits specify RAM locations A5-A4.

ALU

The 4-bit logic circuit that provides arithmetic and logical operations.

ACCUMULATOR (Acc)

Consisting of a 4-bit register, the accumulator holds the result of operations or the data present on ports.

C-FLAG

The flag that holds a carry generated from the result of operations.

TIMING CONTROL (TC)

A 0 level on the $\overline{RESET}$ pin for longer than a predetermined period initializes the internal circuitry and ports.

Clock pulses are supplied to the OSC_0 pin from an external source. A crystal or ceramic oscillator may be connected to OSC_0 and $\overline{OSC_1}$ to form an oscillator circuit to produce clock pulses.

ABSOLUTE MAXIMUM RATING

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	Ta = 25°C	−0.3 ~ 7	V
Input Voltage	V_I		−0.3 ~ V_{DD}	V
Output Voltage	V_O		−0.3 ~ V_{DD}	V
Power Dissipation	P_D	Ta = 25°C per one package	200 max.	mW
		Ta = 25°C per one output	50 max.	mW
Storage Temperature	Tstg	—	−55 ~ +150	°C

OPERATING CONDITIONS

Parameter	Symbol	Condition	Limits	Unit
Supply Voltage	V_{DD}	f(OSC) $\leqq$ 1MHz	3 ~ 6	V
		f(OSC) $\leqq$ 4.2MHz	4.5 ~ 5.5	V
Memory-Hold Voltage	V_{DDH}	–	2 ~ 6	V
Operating Temperature	T_{OP}	–	−40 ~ +85	°C
Fan Out	N	MOS Load	15	–
		TTL Load	1	–

DC CHARACTERISTICS

(V_{DD} = 5V±10%, Ta = −40 ~ +85°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
"H" Input Voltage *1, *2	V_{IH}	–	2.4	–	V_{DD}	V
"H" Input Voltage *3, *4	V_{IH}	–	4.2	–	V_{DD}	V
"L" Input Voltage	V_{IL}	–	−0.3	–	0.8	V
"H" Output Voltage *1, *5	V_{OH}	I_O = −15μA	4.2	–	–	V
"L" Output Voltage *1	V_{OL}	I_O = 1.6mA	–	–	0.4	V
"L" Output Voltage *5	V_{OL}	I_O = 15μA	–	–	0.4	V
"L" Output Voltage*6	V_{OL}	I_O = 8mA	—	1	2	V
Input Current *3	I_{IH}/I_{IL}	$V_I = V_{DD}/0V$	–	–	15/−15	μA
Input Current *2, *4	I_{IH}/I_{IL}	$V_I = V_{DD}/0V$	–	–	1/−30	μA
"H" Output Current *1	I_{OH}	V_O = 2.4V	−0.1	–	–	mA
"H" Output Current *1	I_{OH}	V_O = 0.4V	–	–	−1.2	mA
Input Capacity	C_I	f = 1MHz, Ta = 25°C	–	5	–	pF
Output Capacity	C_O		–	7	–	
Current Consumption (STOP)	I_{DDS}	V_{DD} = 2V, no load Ta = 25°C	–	0.2	5	μA
		No load	–	1	100	μA
Current Consumption	I_{DD}	Crystal oscillation, No load, 4.194304MHz	–	6	12	mA

*1 Applied to P0, P1, P3, P4, and P5

*2 Applied to P2

*3 Applied to OSC_0

*4 Applied to $\overline{RESET}$

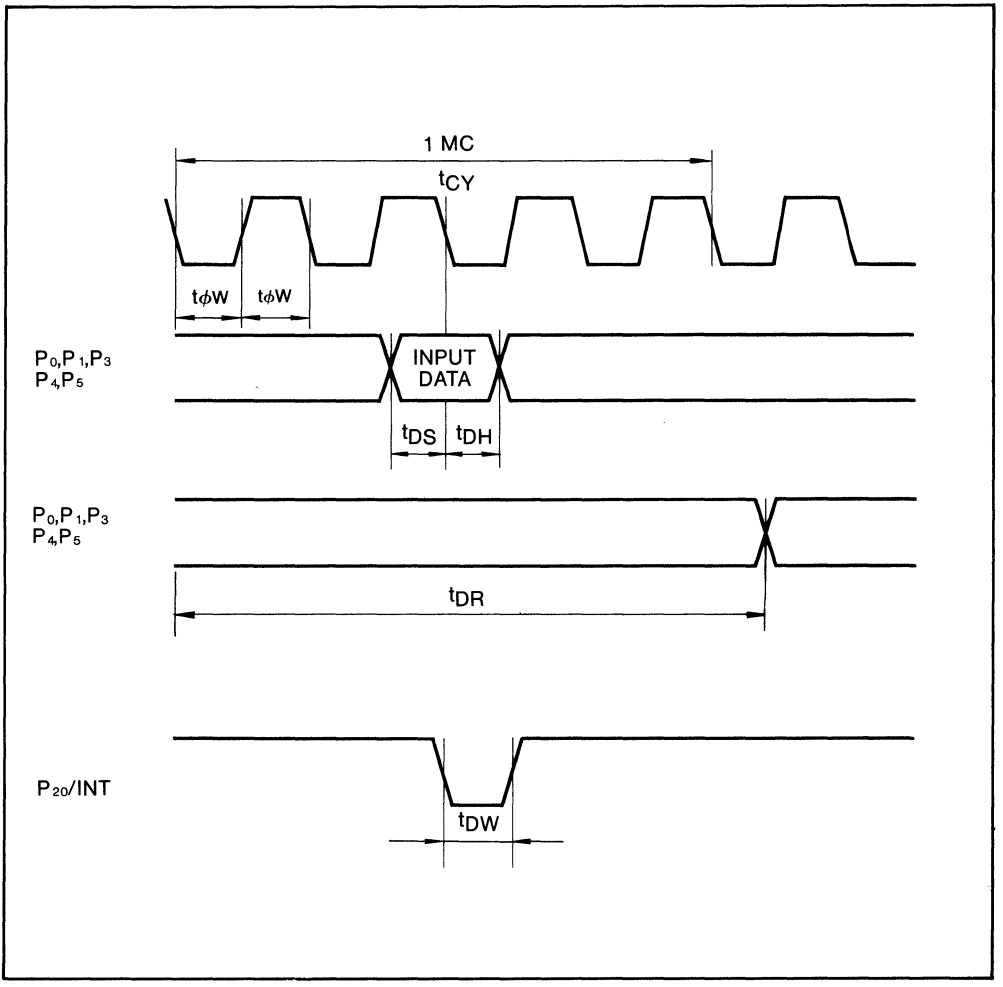
*5 Applied to $\overline{OSC_1}$

*6 In using LED total output current should be within the limit of Power dissipitation in "Absolute Maximum Rating.

AC CHARACTERISTICS

(V_{DD} = 5V±10%, Ta = −40 ~ +85°C)

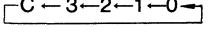
Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Clock (OSC_0) Pulse Width	$t_{\phi W}$	–	119	–	–	ns
Cycle Time	t_{CY}	–	952	–	–	ns
Input Data Setup Time	t_{DS}	–	120	–	–	ns
Input Data Hold Time	t_{DH}	–	120	–	–	ns
Input Data/Input Clock Pulse Width	t_{DW}	–	120	–	–	ns
Data Delay Time	t_{DR}	C_L=15pF	–	–	t_{CY} +300	ns



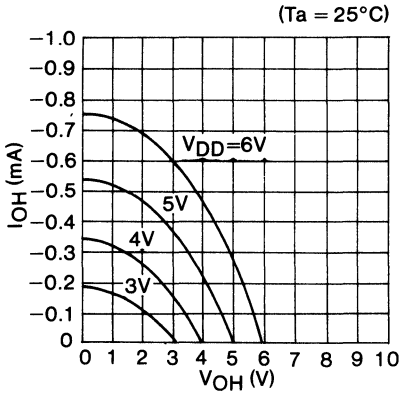
INSTRUCTION SET

	Mnemonic	Hex op code	Byte	Cycle	Description
Load	LAI n	90 – 9F	1	1	Acc ← n
	LLI n	80 – 8F	1	1	L ← n
	LAL	21	1	1	Acc ← L
	LLA	2D	1	1	L ← Acc
	LAH	22	1	1	Acc ← H
	LHA	2E	1	1	H ← Acc
	LAM	38	1	1	Acc ← M
	LMA	2F	1	1	M ← Acc
	X	28	1	1	Acc ← M
	LMI nn	14 · nn	2	2	M(W) ← nn
	LHLI nn	15 · nn	2	2	HL ← nn
	LAMD mm	10 · mm	2	2	Acc ← Md
	LMAD mm	11 · mm	2	2	Md ← Acc
	IPO p	3D · pD	2	2	Acc ← Pp
	OPD p	3D · pC	2	2	Pp ← Acc
Control	MEI	3E · 60	2	2	MEIF ← "1"
	MDI	3E · 61	2	2	MEIF ← "0"
	EIEX	3D · C8	2	2	EIEXF ← "1"
	EITB	3D · C9	2	2	EITBF ← "1"
	DIEX	3D · C4	2	2	EIEXF ← "0"
	DITB	3D · C5	2	2	EITBF ← "0"
	TIEX	3D · C0	2	2	SKIP IF EIEXF="1"
	TITB	3D · C1	2	2	SKIP IF EITBF="1"
	TQEX	3D · 20	2	2	SKIP IF IRQEX="1"
	TQTB	3D · D0	2	2	SKIP IF IRQTB="1"
	RQEX	3D · 24	2	2	IRQEX ← "0"
	RQTB	3D · D4	2	2	IRQTB ← "0"
Increment/ decrement	INL	31	1	1	L ← L+1, SKIP IF L="0"
	INH	32	1	1	H ← H+1, SKIP IF H="0"
	INM	33	1	1	M ← M+1, SKIP IF M="0"
	DCL	35	1	1	L ← L-1, SKIP LF L="F"
	DCH	36	1	1	H ← H-1, SKIP IF H="F"
	DCM	37	1	1	M ← M-1, SKIP IF M="F"
	INMD mm	12 · mm	2	2	Md ← Md+1, SKIP IF Md="0"

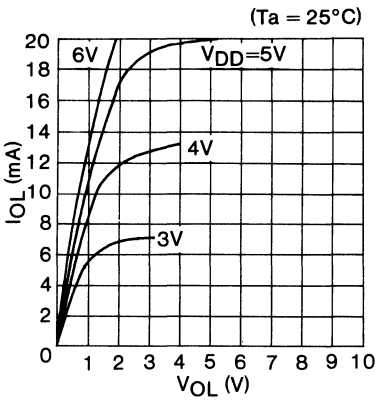
INSTRUCTION SET (CONT.)

	Mnemonic	Hex op code	Byte	Cycle	Description
Bit handling	TAB n2	54 - 57	1	1	SKIP IF (Acc-Bit n2) = "1"
	TMB n2	58 -5B	1	1	SKIP IF (M-Bit n2) = "1"
	RMB n2	68 - 6B	1	1	(M-Bit n2) ← "0"
	SMB n2	78 - 7B	1	1	(M-Bit n2) ← "1"
	TPBD p n2	3D · p0~3	2	2	SKIP IF (Pp-Bit n2) = "1"
	RPBD p n2	3D · p4~7	2	2	(Pp-Bit n2) ← "0"
	SPBD p n2	3D · p8~B	2	2	(Pp-Bit n2) ← "1"
	TC	09	1	1	SKIP IF C = "1"
	RC	08	1	1	C ← "0"
	SC	07	1	1	C ← "1"
Arithmetic	ADS	02	1	1	Acc ← Acc+M, SKIP IF Cy="1"
	ADC	03	1	1	C, Acc ← C+Acc+M
	AIS n	3E · 4n	2	2	Acc ← Acc+n, SKIP IF Cy="1"
	DAS	0A	1	1	Acc ← Acc+10
	AND	0D	1	1	Acc ← AccΛM
	OR	05	1	1	Acc ← AccVM
	EOR	04	1	1	Acc ← Acc∀M
	CMA	0B	1	1	Acc ← $\overline{\text{Acc}}$
	CAM	16	1	1	SKIP IF Acc=M
	CAI n	3E · 0n	2	2	SKIP IF Acc=n
	RAL	0E	1	1	
Branch	JCP a6	C0 - FF	1	1	PC ← a6
	JP a11	40 - 47 00 - FF	2	2	PC ← a11
	CAL a11	A0 - A7 00 - FF	2	4	STACK ← PC+2, PC←a11, SP←SP-1
	RT	1E	1	4	PC ← STACK, SP ← SP+1
Others	PUSH	1C	1	3	STACK ← PC+2, PC←a11, SP←SP-1
	POP	1D	1	3	C, Acc, H, L ← STACK, SP←SP+1
	STOP	3D · B9	2	2	CLOCK STOP
	NOP	00	1	1	NO OPERATION

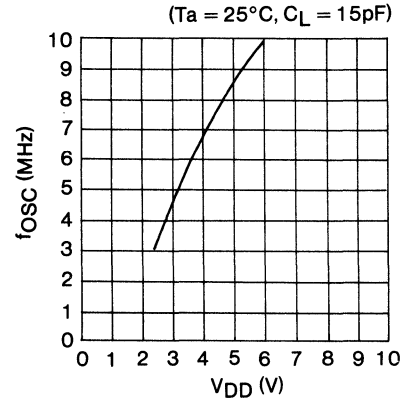
TYP. Current (I_{OH}) vs Voltage (V_{OH}) for High state Output



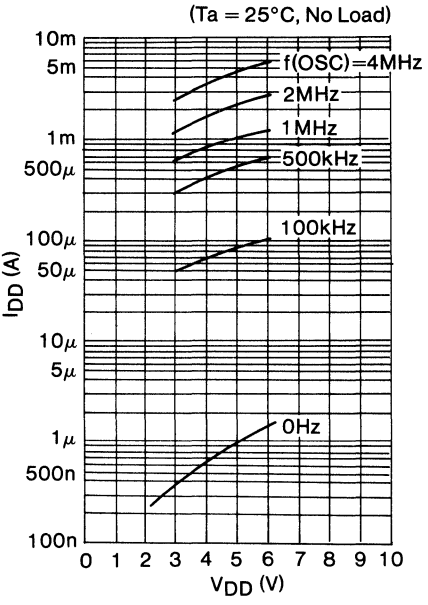
TYP. Current (I_{OL}) vs Voltage (V_{OL}) for Low state Output



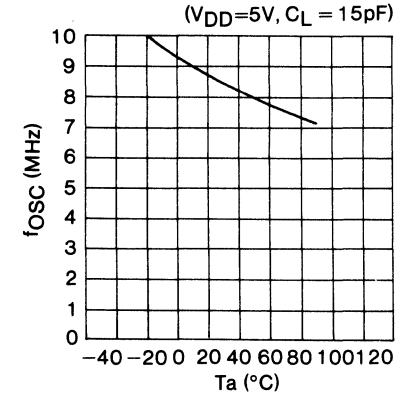
TYP. Maximum Oscillator Frequency f(OSC) vs Supply Voltage (V_D)



TYP. Supply Current (I_{DD}) vs Supply Voltage (V_{DD})



TYP. Maximum Oscillator Frequency f(OSC) vs Temperature (Ta)



MSM6442

CMOS 4-BIT SINGLE CHIP MICROCONTROLLER WITH LCD DRIVER

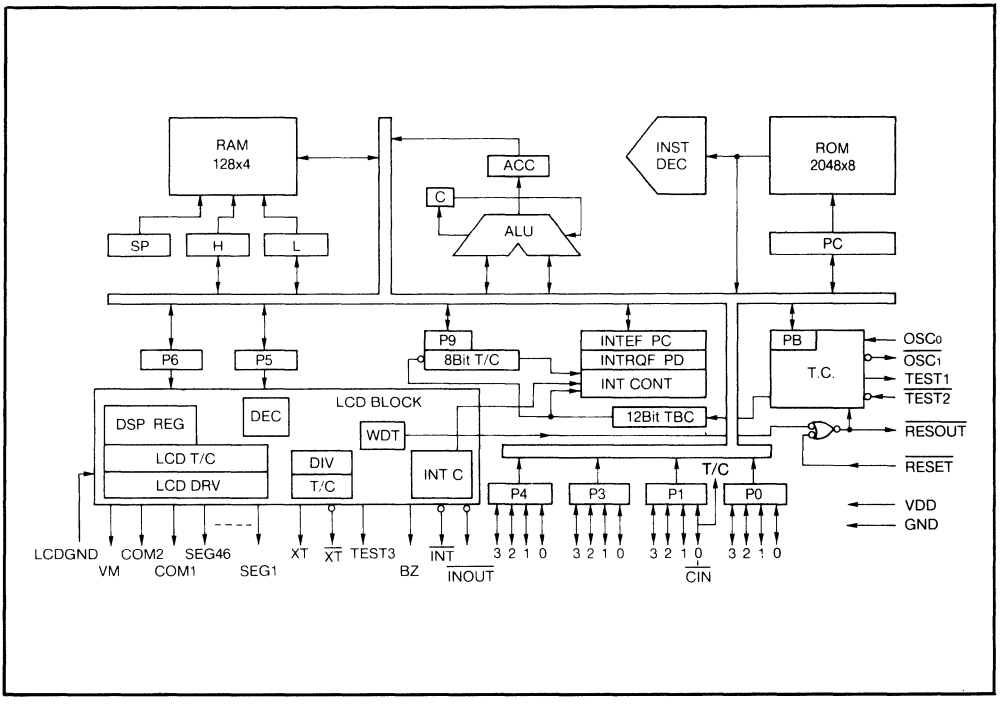
GENERAL DESCRIPTION

The OKI MSM6442 is a low power, high performance single chip device implemented in complementary metal oxide semiconductor technology with 46 segment outputs and 2 commons. Also integrated onto this chip are 16K bits mask program ROM, 512 bits of data RAM, 28 Input/Output lines and oscillator. 71 instructions include binary, BCD, logical operations; bit set, reset, test; subroutine call and return.

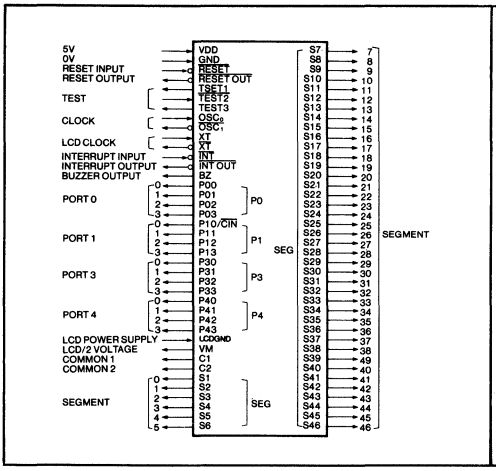
FEATURES

- Low Power Consumption 30mW (typ)
- 2048 × 8 Internal ROM
- 128 × 4 Internal RAM
- Two built-in counters
  12-bit time-base counter
  8-bit programable timer/event counter
- 16 Input/Output Ports and 46 LCD Output Port and 2 Common Output (1/2 Duty, 1/2 Bias)
- LED direct drive available
- Self-contained Oscillator
- 71 Instructions
- 4 Interrupt Levels
- 16 Stack Levels
- −40 to +85°C Operating Temperature
- 4.5 to 5.5V Operating V_{DD} at 4.2 MHz
  3.0 to 6.0V Operating V_{DD} at 1 MHz
- TTL Compatible
- Package:
  80 pin plastic QFP (QFP80-P-1420-K)

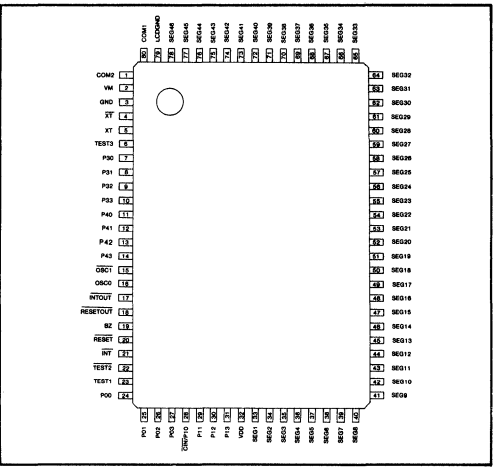
FUNCTIONAL BLOCK DIAGRAM



LOGIC SYMBOL



PIN CONFIGURATION (TOP VIEW)



PIN DESCRIPTION

Terminal	Input/ Output	Function	When reset
P00 ~ P03 P10 ~ P13 P30 ~ P33	Input/ Output	I/O port I/O port (P10 and count input $\overline{CIN}$ are in common) I/O port	"1"
P40 ~ P43	Input/ Output	I/O port	"0"
SEG1 ~ SEG16 SEG17 ~ SEG46	Output*	LCD output port (can be assigned to data output in 4 bit wide) LCD output port	"0"*
COM1 COM2	Output*	LCD common output terminal 1 LCD common output terminal 2	"0"*
$\overline{INT}$	Input	Input port of external interrupt	–
$\overline{INT\ OUT}$	Output	Interrupt output port	"1"
$\overline{RESET}$		Reset input port	–
$\overline{RESET\ OUT}$	Output	Reset output terminal	"1"
BZ	Output	Buzzer pulse output port in 2048 KHz	"0"*
$\overline{OSC_0}$ $\overline{OSC_1}$	Input/ Output	Crystal OSC or ceramic OSC connection Crystal OSC or ceramic OSC connection (System clock)	–
$\overline{XT}$ XT	Input/ Output	32.768 kHz crystal oscillator connection (use for LCD control)	–
$\overline{TEST\ 1}$ $\overline{TEST\ 2}$ TEST 3	– –	TEST terminal 1 (open) (Connected to V_{DD}) TEST terminal 2 (open) TEST terminal 3 (open)	
V_{DD}	Input	Power supply (5V)	–
LCDGND	Input	Power supply for LCD	–
VM	Input/ Output	(V_{DD}–LCDGND)/2 supply voltage output or supply voltage input	"0"*
GND	Input	Power supply (0V)	–

*"0" indicates the LCD GND voltage level

ELECTRIC CHARACTERISTICS

• Absolute Maximum Ratings

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	Ta = 25°C	−0.3 ~ 7	V
Input Voltage	V_I		−0.3 ~ V_{DD}	V
Output Voltage	V_O		−0.3 ~ V_{DD}	V
LCD Voltage	LCDGND		V_{DD} − 9 ~ V_{DD}	V
Storage Temperature	Tstg	—	−55 ~ +150	°C

• Operating Conditions

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	f (OSC) ≦ 1MHz	3 ~ 6	V
		f (OSC) ≦ 4.2MHz	4.5 ~ 5.5	V
LCD Voltage	LCDGND	*1	V_{DD} − 8 ~ 0	V
Memory Retension Voltage	V_{DDH}	Oscillation off	2 ~ 6	V
Operating Temperature	Top	—	−40 ~ +85	°C
LCD Clock Oscillation Frequency	f(XT)	*2	32.768	kHz
Fan Out (I/O Port)	N	MOS Load	15	—
		TTL Load	1	—

*1 Voltage applied to LCD is (V_{DD}−LCDGND).

*2 Oscillation Circuit for LCD Clock (XT, $\overline{XT}$ Port) is for Christal Oscillation only.

DC CHARACTERISTICS

(VDD = 5V ±10%, LCDGND = 0V, Ta = −40 ~ +85°C)

Parameter		Symbol	Condition	MIN	TYP	MAX	Unit
"H" Input Voltage	*1. $\overline{INT}$	VIH	—	2.4	—	VDD	V
	*3.		—	3.6	—	VDD	V
"L" Input Voltage	*1. *4	VIL	—	0	—	0.8	V
"H" Output Voltage	*1. $\overline{OSC1}$	VOH	IO = −15μA	4.2	—	—	V
	*2.		IO = −400μA	2.4	—	—	V
	SEG1~SEG46		IO = −10μA	VDD-0.2	—	—	V
	COM1, COM2		IO = −50μA	VDD-0.2	—	—	V
"L" Output Voltage	*1, *2	VOL	IO = 1.6mA	—	—	0.4	V
	$\overline{OSC1}$		IO = 15mA	—	—	0.4	V
	SEG1~SEG46		IO = 10μA	—	—	0.2	V
	COM1, COM2		IO = 50μA	—	—	0.2	V
"M" Output Voltage	COM1, COM2	VOM	IO ±0.5	VDD/2 0.2	—	VDD/2+0.2	V
"H" Input Current	OSC0	IIH	VI= VDD	—	—	15	μA
	XT			—	—	7	μA
	$\overline{INT}$, $\overline{RESET}$			—	—	1	μA
"L" Input Current	OSC0	IIL	VI = 0V	—	—	−15	μA
	XT			—	—	7	μA
	$\overline{INT}$, $\overline{RESET}$			—	—	−30	μA
"H" Output Current	*1	IOH	VO = 2.4V	−0.1	—	—	mA
			VO = 0.4V	—	—	−1.2	mA
Current Consumption —at stop mode —no oscillation		IDDS	VDD = 2V, TA = 25°C No load Display off XT Port is fixed to "L"	—	0.2	10	μA
			Display off XT Port is fixed to "L"	—	1	100	μA
Current Consumption —at stop mode		IDDL	No load Display off At stop mode f(XT) = 32.768 KHz	—	100	200	μA
Current Consumption		IDD	No load Display off f(osc) = 4.2 MHz f(XT) = 32.768 KHz	—	6	12	mA

*1 Applied to P0, P1, P2, and P4.
*2 Applied to $\overline{INTOUT}$, $\overline{RESET}$, and BZ.
*3 Applied to OSC0, XT, and $\overline{RESET}$.
*4 Applied to XT, $\overline{INT}$, and $\overline{RESET}$.

Note: "M" output voltage is intermediate voltage of the output from common port at dynamic display.

AC CHARACTERISTICS

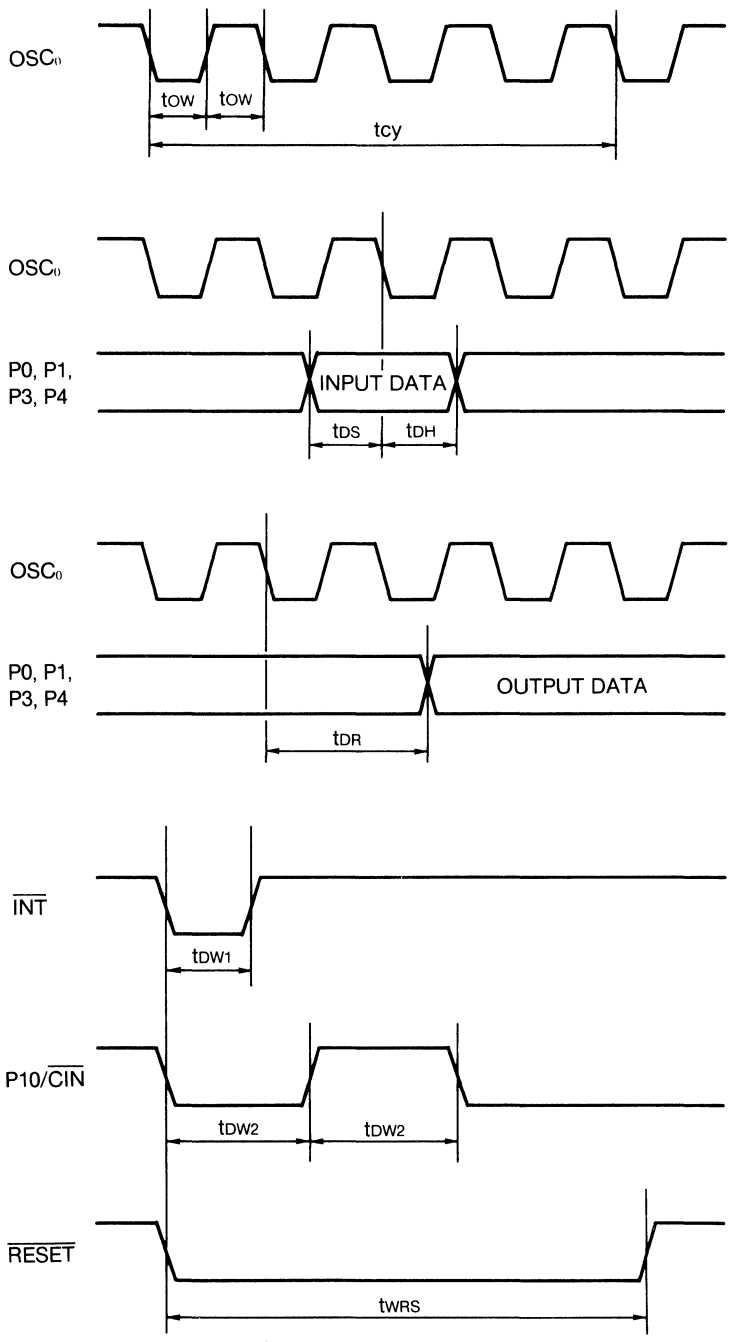
(VDD = 5V ±10%, Ta = −40 ~ +85°)

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Clock (OSC_0) Pulse Width	t_{OW}	—	119	—	—	nS
Cycle Time	t_{CY}	—	952	—	—	nS
Input Data Set-up Time	t_{DS}	—	120	—	—	nS
Input Data Hold Time	t_{DH}	Note 1	120	—	—	nS
INT Input Data Pulse Width	t_{DW1}	—	120	—	—	nS
CT Clock Pulse Width	t_{DW2}	—	2/8tcy÷120	—	—	nS
Data Delay Time	t_{DR}	C_L = 15pF	—	—	300	nS
Reset Input Pulse Width	t_{WRS}	Note 2	2tcy	—	—	nS

Note 1* To release powerdown by inputting "L" level into INT Port, pulse width should be longer than the time for the oscillation stabilization at OSC_0.

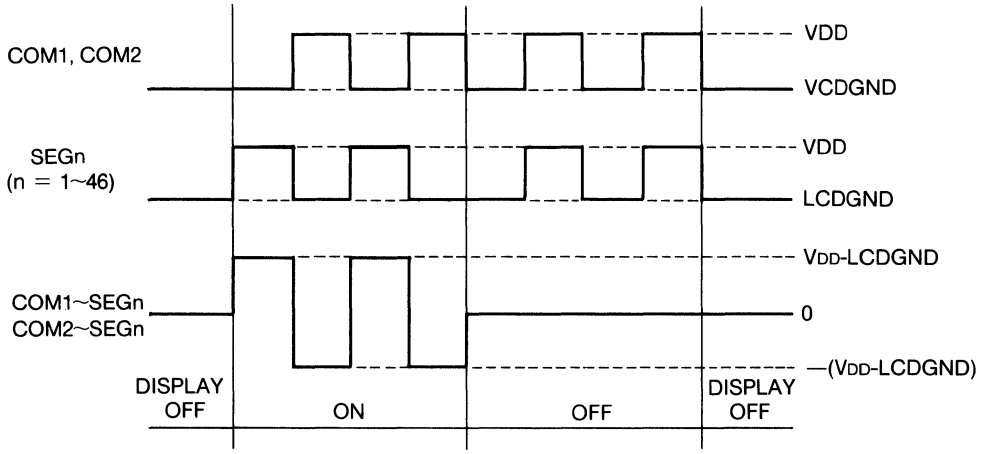
Note 2* The condition of stable oscillation. To release powerdown by reset, pulse width should be longer than the time for oscillation stabilization at OSC_0.

TIMING CHART

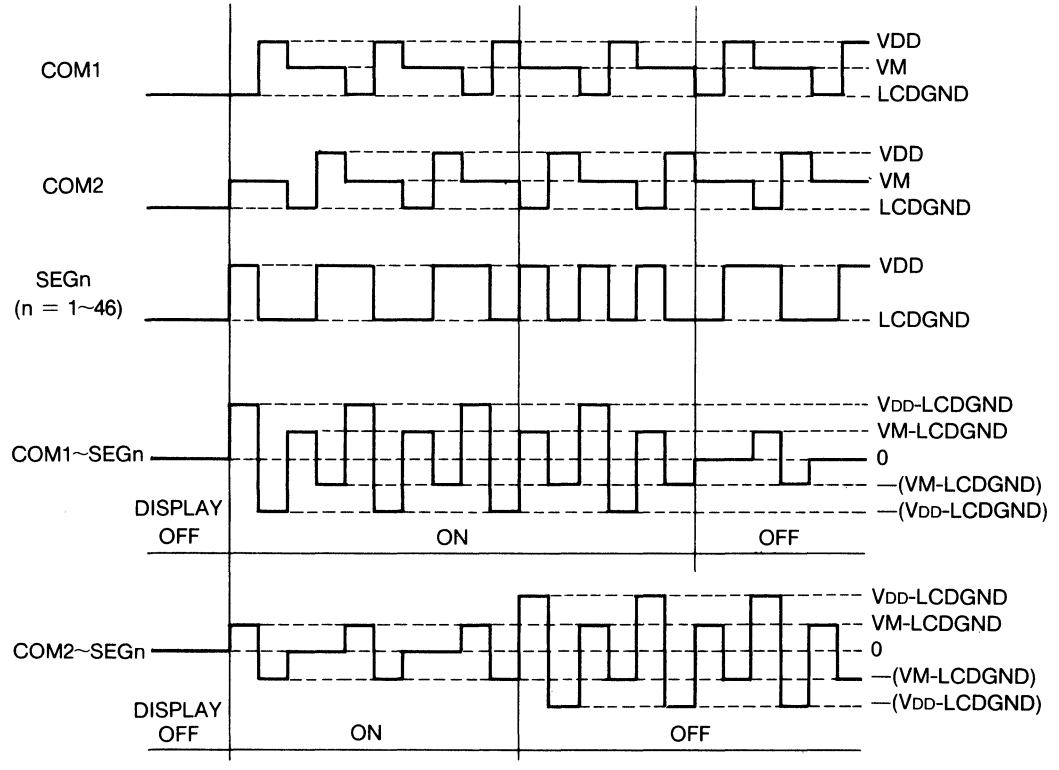


OUTPUT WAVE FORM OF LCD DRIVER

STATIC MODE



DYNAMIC MODE



FUNCTIONAL DESCRIPTION

ROM

Organized into 2048 words by 8 bits, ROM is used to store developed application programs (instructions). It is addressed by the program counter (PC).

PROGRAM COUNTER (PC)

The program counter consists of a 11-bit binary counter. It is used to address ROM.

STACK

An interrupt or CAL instruction causes the contents of the PC to be saved in the stack. Also, the PUSH instruction causes the contents of accumulator, carry-flag, H- and L-register to be saved in it. These are allowed to be restored by the RT instruction or POP instruction.

RAM

Organized into 128 words of 4 bits, RAM is addressed by the H- and L-register or the contents of the second byte of an instruction.

L-REGISTER

A 4-bit register which specifies the row address of RAM and the port-address in the port operation instructions. It is also used as a working register.

H-REGISTER

A 4-bit register which specifies the column address of RAM and is used as a working register.

ALU

A 4-bit logic circuit which provides arithmetic and logical operations.

ACCUMULATOR (ACL)

Consisting of a 4-bit register, the accumulator holds the result of operations or the date present on ports.

C-FLAG

The flag that holds a carry generated from the result of operations.

INPUT/OUTPUT Ports (16 bits)

16 input/output ports are provided for effecting and controlling data transfer to and from an external source. The ports are selected by codes included in the instructions.

12-bit TIME-BASE COUNTER

The time base counter consists of a 12-bit binary counter. It is used to devide the frequency of the OSC_0 input by 2^{12} and generate the interrupt request at every over-flow signal.

8-bit TIMER EVENT COUNTER

The timer event counter consists of a 8-bit counter (8-bit) register, comparing and controlling circuits. It is used to count pulses of an internal or external source. Coincidently, if value between the counter and the register causes interrupt request occur.

LCD DRIVER

The LCD driver is used to effect LCD display by transferring data in a program to the register assigned as port 5 and 6. It is available to select driving in static or dynamic operation (1/2 duty cycle) and frame frequency (128 Hz/64 Hz) and to drive up to 92 segments at 1/2 duty. Also, 16 outputs(SEG1~SEG16)of the segment terminals can be used as normal data outputs.

A standard LCD clock is produced by the oscillation dividing a crystal oscillator (32.768 kHz) connected to XT and $\overline{XT}$ terminals. This is also used as standard clock of displaying, clock interrupting and watch dog timer. (This clock can be also produced by dividing a frequency of 4.194304 MHz. Note the selection of the frame frequency, when the crystal oscillator is used without a frequency of 4.194304 MHz.)

INTERRUPT

As shown below, 1 ~ 4 is available to interrupt;

1. External interrupt at the falling edge of INT signal input
2. Clock interrupt at every second (32.768 kHz crystal oscillator)
3. Time base counter interrupt at the occurance of an overflow of the timer base counter.
4. Timer event counter interrupt coinciding between the signals of the 8-bit counter and register.

Interrupts 1 and 2 are also used to release the power down mode.

WATCH DOG TIMER (WDT)

A timer for detecting the overrunning of the program. This timer produces the overflow signal by dividing the 64 Hz frequency by 4 generated from the oscillation of a frequency of 32.768 kHz. It can be also halted, when unused.

TIMING CONTROL (T.C)

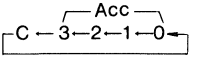
A 0 level on the $\overline{RESET}$ pin for longer than predetermined period initializes the internal circuitry and ports.

Clock pulses are supplied to the OSC_0 pin from an external source. A crystal or ceramic oscillator may be connected to OSC_0 and $\overline{OSC_1}$ to form an oscillator circuit to produce clock pulses.

INSTRUCTION SET

	Mnemonic		Hex op code	Byte	Cycle	Description
Load	LAI	n	90 – 9F	1	1	Acc ← n
	LLI	n	80 – 8F	1	1	L ← n
	LAL		21	1	1	Acc ← L
	LLA		2D	1	1	L ← Acc
	LAH		22	1	1	Acc ← H
	LHA		2E	1	1	H ← Acc
	LAM		38	1	1	Acc ← M
	LMA		2F	1	1	M ← Acc
	X		28	1	1	Acc ⟷ M
	LMI	nn	14 · nn	2	2	M(W) ← nn
	LHLI	nn	15 · nn	2	2	HL ← nn
	LAMD	mm	10 · mm	2	2	Acc ← Md
	LMAD	mm	11 · mm	2	2	Md ← Acc
	LMCT		3E · 59	2	2	M(W) ← CT
	LCTM		3E · 51	2	2	CT ← M(W)
	IPD	p	3D · pD	2	2	Acc ← Pp
	OPD	p	3D · pC	2	2	Pp ← Acc
Control	MEI		3E · 60	2	2	MEIF ← "1"
	MDI		3E · 61	2	2	MEIF ← "0"
	EIEX		3D · C8	2	2	EIEXF ← "1"
	EICT		3D · CB	2	2	EICTF ← "1"
	DIEX		3D · C4	2	2	EIEXF ← "0"
	DICT		3D · C7	2	2	EICTF ← "0"
	TIEX		3D · C0	2	2	SKIP IF EIEXF="1"
	TICT		3D · C3	2	2	SKIP IF EICTF="1"
	TQEX		3D · 20	2	2	SKIP IF IRQEX="1"
	TQCT		3D · D2	2	2	SKIP IF IRQCT="1"
	RQEX		3D · 24	2	2	IRQEX ← "0"
	RQCT		3D · D6	2	2	IRQCT ← "0"
Increment/ decrement	INL		31	1	1	L ← L+1, SKIP IF L="0"
	INH		32	1	1	H ← H+1, SKIP IF H="0"
	INM		33	1	1	M ← M+1, SKIP IF M="0"
	DCL		35	1	1	L ← L−1, SKIP IF L="F"
	DCH		36	1	1	H ← H−1, SKIP IF H="F"
	DCM		37	1	1	M ← M−1, SKIP IF M="M"
	INMD	mm	12 · mm	2	2	Md ← Md+1, SKIP IF Md="0"

INSTRUCTION SET (CONT.)

	Mnemonic		Hex op code	Byte	Cycle	Description
Bit handling	TAB	n2	54 - 57	1	1	SKIP IF (ACC−Bit n2) = "1"
	TPB	n2	50 - 53	1	1	SKIP IF (P−Bit n2) = "1"
	RPB	n2	60 - 63	1	1	(P−Bit n2) ← "0"
	SPB	n2	70 - 73	1	1	(P−Bit n2) ← "1"
	TMB	n2	58 -5B	1	1	SKIP IF (M−Bit n2) = "1"
	RMB	n2	68 - 6B	1	1	(M−Bit n2) ← "0"
	SMB	n2	78 - 7B	1	1	(M−Bit n2) ← "1"
	TPBD p	n2	3D · p0~3	2	2	SKIP IF (Pp−Bit n2) = "1"
	RPBD p	n2	3D · p4~7	2	2	(Pp−Bit n2) ← "0"
	SPBD p	n2	3D · p8~B	2	2	(Pp−Bit n2) ← "1"
	TC		09	1	1	SKIP IF C = "1"
	RC		08	1	1	C ← "0"
	SC		07	1	1	C ← "1"
Arithmetic	ADS		02	1	1	Acc ← Acc+M, SKIP IF Cy="1"
	ADC		03	1	1	C, Acc ← C+Acc+M
	AIS	n	3E · 4n	2	2	Acc ← Acc+n, SKIP IF Cy="1"
	DAS		0A	1	1	Acc ← Acc+10
	AND		0D	1	1	Acc ← Acc∧M
	OR		05	1	1	Acc ← Acc∨M
	EOR		04	1	1	Acc ← Acc¥M
	CMA		0B	1	1	Acc ← Acc
	CAM		16	1	1	SKIP IF Acc=M
	CAI	n	3E · 0n	2	2	SKIP IF Acc=n
	RAL		0E	1	1	
Branch	JCP	a6	C0 - FF	1	1	PC ← a6
	JP	a11	40 _ 47 00 - FF	2	2	PC ← a11
	CAL	a11	A0 _ A7 00 - FF	2	4	STACK ← PC+2, PC←a11, SP←SP−1
	RT		1E	1	4	PC ← STACK, SP ← SP+1
Others	PUSH		1C	1	3	STACK ← C, Acc, H, L, SP←SP−1
	POP		1D	1	3	C, Acc, H, L ← STACK, SP←SP+1
	STOP		3D · B9	2	2	CLOCK STOP
	NOP		00	1	1	NO OPERATION
	ECT		3D·BB	2	2	CTF ← "1" (Counter Start)
	DCT		3D·B7	2	2	CTF ← "0" (Counter Stop)
	TCT		3D·B3	2	2	Skip if CTF = "1"

OKI semiconductor
MSM6434

HIGH-SPEED 4-BIT SINGLE CHIP MICROCONTROLLER WITH A/D CONVERTER

GENERAL DESCRIPTION

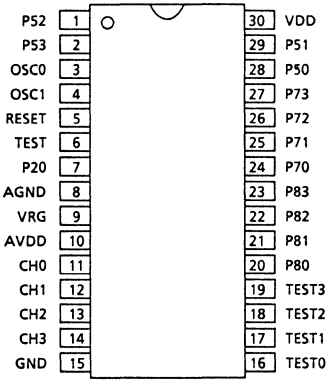
The OKI MSM6434 microcontroller is a low power, high-performance single-chip device implemented in complementary metal oxide semiconductor technology. 32K bits of mask program ROM, 1024 bits of data RAM, 13 Input/Output lines, a programmable timer/event-counter, 8 bit A/D converter, and oscillator are integrated onto one chip. Program memory is byte wide and data-paths are organized in 4 bit nibbles. RAM and I/O lines are bit addressable. 113 instructions include binary, BCD operations; bit set, reset, test; 8 bit I/O; relative jumps; multi functional instructional (increment, modify, skip) 8 bit wide table output; subroutine call and return.

FEATURES

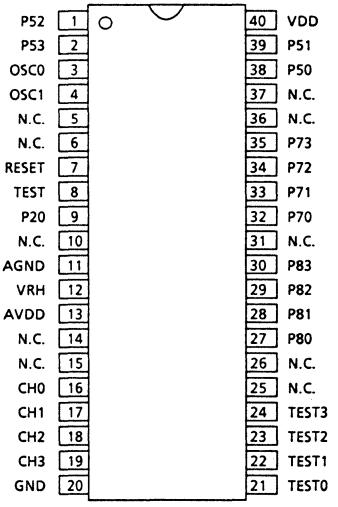
- 4000 x 8 MASK ROM
  An evaluation board is available for up to 8k x 8.
- 256 x 4 RAM (including the stack area)
- 3 x 4, 1 x 1 ports, 13 I/O lines
  1 lines for input ports having a latch, and the other 12 lines for bit operation are available.
- Three built-in counters
  12-bit time-base counter
  12-bit programmable timer
  8-bit high-speed programmable time/event counter
- 4 interrupts with four priority levels (3 internal, 1 external)
- 32 stack levels (in RAM)
- LED direct drive available (8 mA x 3 ports at the same time)
- Power down features
- Instruction execution time
  952 ns 4.2 MHz clock
- Instruction systems suitable for control
- 113 instructions
- 8 bit A/D converter (4 channel)
- Full static operation
- Low power consumption
  TYP 0.4μW at V_{DD} = 2V
  TYP 5μW at V_{DD} = 5V 0Hz clock
- 5V single power supply
- Package:
  30 pin plastic shrink DIP (SDIP30-P-600)
  40 pin plastic DIP (DIP40-P-400)
  44 pin PLCC (QFJ44-P-S650)

PIN CONFIGURATION

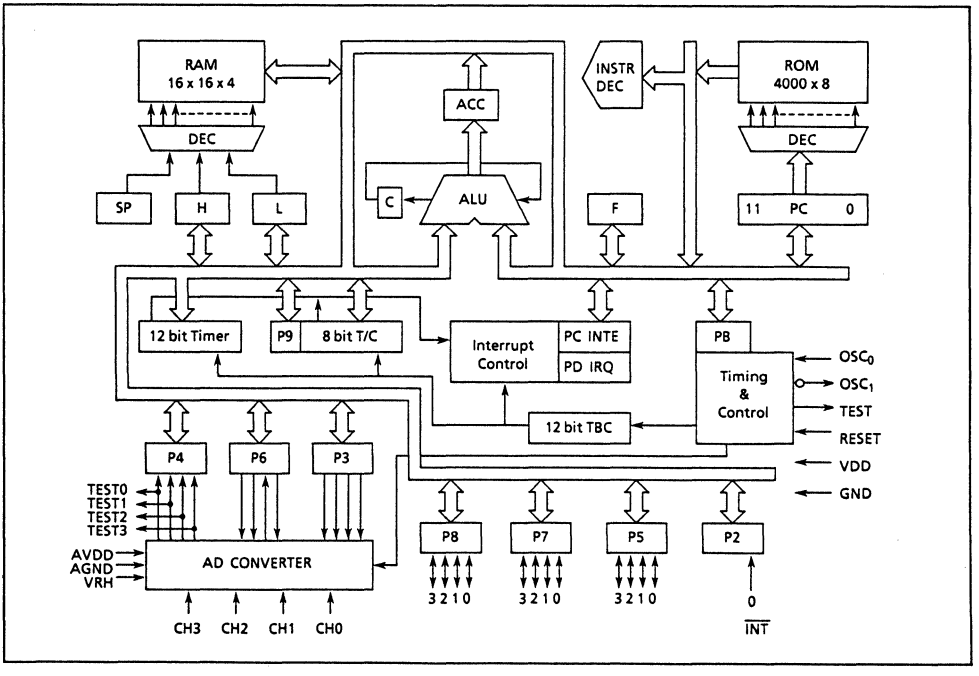
(Top View) 30-Lead Plastic Shrink DIP



(Top View) 40-pin Plastic DIP



BLOCK DIAGRAM



PIN DESCRIPTION

Pin Name	Input/Output	Function	When Reset
P20/$\overline{INT}$	Input	Input port with a latch. P20 is shared with INT input. (Fall trigger input). Built-in pull up register	The latch is reset
P50 - 53	Input/Output	4-bit input/output port	"0"
P70 - 73	Input/Output	4-bit input/output port	"0"
P80 - 83	Input/Output	4-bit input/output port	"0"
$\overline{OSC0}$ $\overline{OSC1}$	Input/Output	X'tal connection terminal for system clock oscillation	Oscillation wave
TEST	Output	(Test terminal for Maker)	Pulse output
TEST0 - 3	Output		Hi-z
RESET	Input	System reset input terminal	"0"
CH0 - 3	Input	Analog voltage input pin	
VRH		Reference voltage input pin for A/D converter	
AVDD AGND		A/D converter power supply	
VDD GND		System power supply	

INSTRUCTION LIST

	Mnemonic	Description	Code	Byte	Cycle
Load, Push, Pop	LAI n	A←n	9n	1	1
	LLI n	L←n	8n	1	1
	LHLI nn	HL←nn	15nn	2	2
	LMI nn	M(w)←nn	14nn	2	2
	LAL	A←L	21	1	1
	LLA	L←A	2D	1	1
	LAH	A←H	22	1	1
	LHA	H←A	2E	1	1
	LAM	A←M	38	1	1
	LMA	M←A	2F	1	1
	LAM +	A←M, L←L + 1, Skip if L = 0	24	1	1

INSTRUCTION LIST (Continued)

	Mnemonic	Description	Code	Byte	Cycle
Load, Push, Pop	LAM –	A←M, L←L – 1, Skip if L = F	25	1	1
	LMA +	M←A, L←L + 1, Skip if L = 0	26	1	1
	LMA –	M←A, L←L – 1, Skip if L = F	27	1	1
	LAMM n_2	A←M, H←HVn_2	39-3B	1	1
	LAMD mm	A←Md	10mm	2	2
	LMAD mm	Md←A	11mm	2	2
	LMTD mm	Md(w)←T(M(w), A), T = ROM table	19mm	2	3
	LMCT	M(w)←CT	3E59	2	2
	LCTM	CT←M(w)	3E51	2	2
	LTMM	TM←(M(w), A)	3E50	2	2
	PUSH	ST←C, A, H, L, SP←SP – 4	1C	1	3
	POP	C, A, H, L←ST SP←SP + 4	1D	1	3
Exchange	X	A↔M	28	1	1
	XM n_2	A↔M, H↔HVn_2	29-2B	1	1
	X +	A↔M, L←L + 1, Skip if L = 0	3C	1	1
	X –	A↔M, L←L – 1, Skip if L = F	2C	1	1
Increment/ Decrement	INA	A←A + 1, Skip if A = 0	30	1	1
	INM	M←M + 1, Skip if M = 0	33	1	1
	INL	L←L + 1, Skip if L = 0	31	1	1
	INH	H←H + 1, Skip if M = 0	32	1	1
	INMD mm	Md←Md + 1, Skip if Md = 0	12mm	2	2
	DCA	A←A – 1, Skip if A = F	34	1	1
	DCM	M←M – 1, Skip if M = F	37	1	1
	DCL	L←L – 1, Skip if L = F	35	1	1
	DCH	H←H – 1, Skip if H = F	36	1	1
	DCMD mm	Md←Md – 1, Skip if Md = F	13mm	2	2
Arithmetic	ADS	A←A + M, Skip if Cy = 1	02	1	1
	ADCS	A, C←A + M + C, Skip if Cy = 1	01	1	1
	ADC	A, C←A + M + C	03	1	1
	AIS n	A←A + n, Skip if Cy = 1	3E4n	2	2
	DAA	A←A + 6	06	1	1
	DAS	A←A + 10	0A	1	1

INSTRUCTION LIST (Continued)

	Mnemonic		Description	Code	Byte	Cycle
Arithmetic	AND		A←A∧M	0D	1	1
	OR		A←A∨M	05	1	1
	EOR		A←A∀M	04	1	1
	CMA		A←$\overline{A}$	0B	1	1
	CIA		A←$\overline{A}$ + 1	0C	1	1
	RAL		Rotate Left with C	0E	1	1
	RAR		Rotate Right with C	0F	1	1
	TC		Skip if C = 1	09	1	1
	SC		C←1	07	1	1
	RC		C←0	08	1	1
Compare	CAI	n	Skip if A = n	3E0n	2	2
	CLI	n	Skip if L = n	3E2n	2	2
	CPI	p, n	Skip if Pp = n	17pn	2	2
	CMI	n	Skip if M = n	3E1n	2	2
	CAM		Skip if A = M	16	1	1
Bit operation	TAB	n_2	Skip if A bit (n_2) = 1	54-57	1	1
	RAB	n_2	A bit (n_2)←0	64-67	1	1
	SAB	n_2	A bit (n_2)←1	74-77	1	1
	TMB	n_2	Skip if M bit (n_2) = 1	58-5B	1	1
	RMB	n_2	M bit (n_2)←0	68-6B	1	1
	SMB	n	M bit (n_2)←1	78-7B	1	1
	TFB	n_2	Skip if F bit (n_2) = 1	5C-5F	1	1
	RFB	n_2	F bit (n_2)←0	6C-6F	1	1
	SFB	n_2	F bit (n_2)←1	7C-7F	1	1
	TPB	n_2	Skip if P bit (n_2) = 1	50-53	1	1
	RPB	n	P bit (n_2)←0	60-63	1	1
	SPB	n	P bit (n_2)←1	70-73	1	1
	TPBD	pn_2	Skip if Pp bit (n_2) = 1	3D $p_{0\sim3}$	2	2
	RPBD	pn_2	Pp bit (n_2)←0	3D $p_{4\sim7}$	2	2
	SPBD	pn_2	Pp bit (n_2)←1	3D $p_{8\sim B}$	2	2

INSTRUCTION LIST (Continued)

	Mnemonic	Description	Code	Byte	Cycle
Interrupt	MEI	MEIF←1	3E60	2	2
	MDI	MEIF←0	3E61	2	2
	EITB	EITBF←1	3DC9	2	2
	EITM	EITMF←1	3DCA	2	2
	EICT	EICTF←1	3DCB	2	2
	EIEX	EIEXF←1	3DC8	2	2
	DITB	EITBF←0	3DC5	2	2
	DITM	EITMF←0	3DC6	2	2
	DICT	EICTF←0	3DC7	2	2
	DIEX	EIEXF←0	3DC4	2	2
	TITB	Skip if EITBF = 1	3DC1	2	2
	TITM	Skip if EITMF = 1	3DC2	2	2
	TICT	Skip if EICTF = 1	3DC3	2	2
	TIEX	Skip if EIEXF = 1	3DC0	2	2
	TQEX	Skip if IRQEX = 1	3D20	2	2
	TQTB	Skip if IRQTB = 1	3DD0	2	2
	TQTM	Skip if IRQTM = 1	3DD1	2	2
	TQCT	Skip if IRQCT = 1	3DD2	2	2
	RQEX	IRQEX←0	3D24	2	2
	RQTB	IRQTB←0	3DD4	2	2
	RQTM	IRQTM←0	3DD5	2	2
	RQCT	IRQCT←0	3DD6	2	2
Counter	ECT	CTF←1 (start)	3DBB	2	2
	DCT	CTF←0 (stop)	3DB7	2	2
	TCT	Skip if CTF = 1	3DB3	2	2
Branch	JCP a_6	PC←a_6	C0~FF	1	1
	JP a_{12}	PC←a_{12}	$4a_{12}$	2	2
	CZP a	ST←PC + 1, PC←2a, SP←SP − 4	Ba	1	4
	CAL a_{12}	ST←PC + 2, PC←a_{12}, SP←SP − 4	Aa_{12}	2	4
	RT	PC←ST, SP←SP + 4	IE	1	4

INSTRUCTION LIST (Continued)

	Mnemonic	Description	Code	Byte	Cycle
Branch	RTS	PC←ST, SP←SP + 4, Skip unconditional	IF	1	4
	JA	PC←(PC←A) + 1	IA	1	1
	JM	PC←(M(w), A)	IB	1	2
Branch	IP	A←P	20	1	1
	IPD P	A←Pp	3DpD	2	2
	OP	P←A	23	1	1
	OPD P	Pp←A	3DpC	2	2
CPU control	NOP	No Operation	00	1	1
	HALT	Halt CPU	3DB8	2	2
	STOP	Stop Clock	3DB9	2	2

ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	Ta = 25°C	– 0.3 to 7	V
	AV_{DD}		– 0.3 to 7 $AV_{DD} = V_{DD}$	V
	VRH		– 0.3 to V_{DD}	V
Input Voltage	V_I		– 0.3 toV_{DD}	V
Output Voltage	V_O		– 0.3 toV_{DD}	V
Power Dissipation	P_D	Ta = 25°C per package	200 max.	mW
		Ta = 25°C per out	50 max.	mW
Storage Temperature	T_{STG}	–	– 55 to + 150	°C

OPERATING RANGE

Item	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	$f_{(OSC)} \leqq 1$ MHz	3 to 6	V
		$f_{(OSC)} \leqq 4.2$ MHz	4.5 to 5.5	V
Data-Hold Voltage	V_{DDH}	$f_{(OSC)} = 0$ Hz	2 to 6	V
Operating Temperature	T_{OP}	–	– 40 to + 85	°C
Fan Out	N	MOS Load	15	–
		TTL Load	1	

DC CHARACTERISTICS

($V_{DD} = AV_{DD} = 5V \pm 10\%$, Ta = – 40 to + 85°C)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
"H" Input voltage *1 *2	V_{IH}	–	2.4	–	V_{DD}	V
"H" Input voltage *3 *4	V_{IH}	–	3.6	–	V_{DD}	V
"L" Input voltage *8	V_{IL}	–	– 0.3	–	0.8	V
"H" Output Voltage *1 *5	V_{OH}	$I_O = -15\mu A$	4.2	–	–	V
"L" Output voltage *1	V_{OL}	$I_O = 1.6$mA	–	–	0.4	V
"L" Output voltage *5	V_{OL}	$I_O = 15\mu A$	–	–	0.4	V
"L" Output voltage *6	V_{OL}	$I_O = 8$mA	–	1	2	V
Input Current *3	I_{IH}/I_{IL}	$V_I = V_{DD}/0V$	–	–	15/– 15	µA
Input Current *2 *4	I_{IH}/I_{IL}	$V_I = V_{DD}/0V$	–	–	1/– 30	µA
"H" Output Current *1	I_{OH}	$V_O = 2.4V$	– 0.1	–	–	mA
"H" Output Current *1	I_{OH}	$V_O = 0.4V$	–	–	– 1.2	mA
Input Capacity	C_I	f = 1MHz, Ta = 25°C	–	5	–	pF
Output Capacity	C_O		–	7	–	

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Current Dissipation (when stop condition)	I_{DDS}	V_{DD} = 2V, no load Ta = 25°C	–	0.2	5	μA
		No load	–	1	100	μA
Current Dissipation	I_{DD}	Quarts oscillation f = 4MHz, no load	–	–	20	mA

*1 Applied to P5, P7 and P8
*2 Applied to P2
*3 Applied to OSC_0
*4 Applied to $\overline{RESET}$
*5 Applied to $\overline{OSC_1}$
*6 In using LED, total output current should be within the limit of Power dissipatation in "Absolute Maximum Rating."

AC CHARACTERISTICS

($V_{DD} = AV_{DD} = 5V \pm 10\%$, Ta = −40 to +85°C)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Clock Pulse Width Clock (OSC)	$t_{\phi W}$	–	119	–	–	ns
Cycle Time	t_{CY}	–	952	–	–	ns
Input Data Setup Time	t_{DS}	–	120	–	–	ns
Input Data Hold Time	t_{DH}	–	120	–	–	ns
Input Data, Input Clock Pulse Width	t_{DH}	–	120	–	–	ns
Data Delay Time	t_{DR}	C_L = 15pF	–	–	t_{CY} + 300	ns
INT Invalid Time	t_{IINH}	–	1/8 t_{CY}	–	–	ns

A/D CONVERSION CHARACTERISTICS

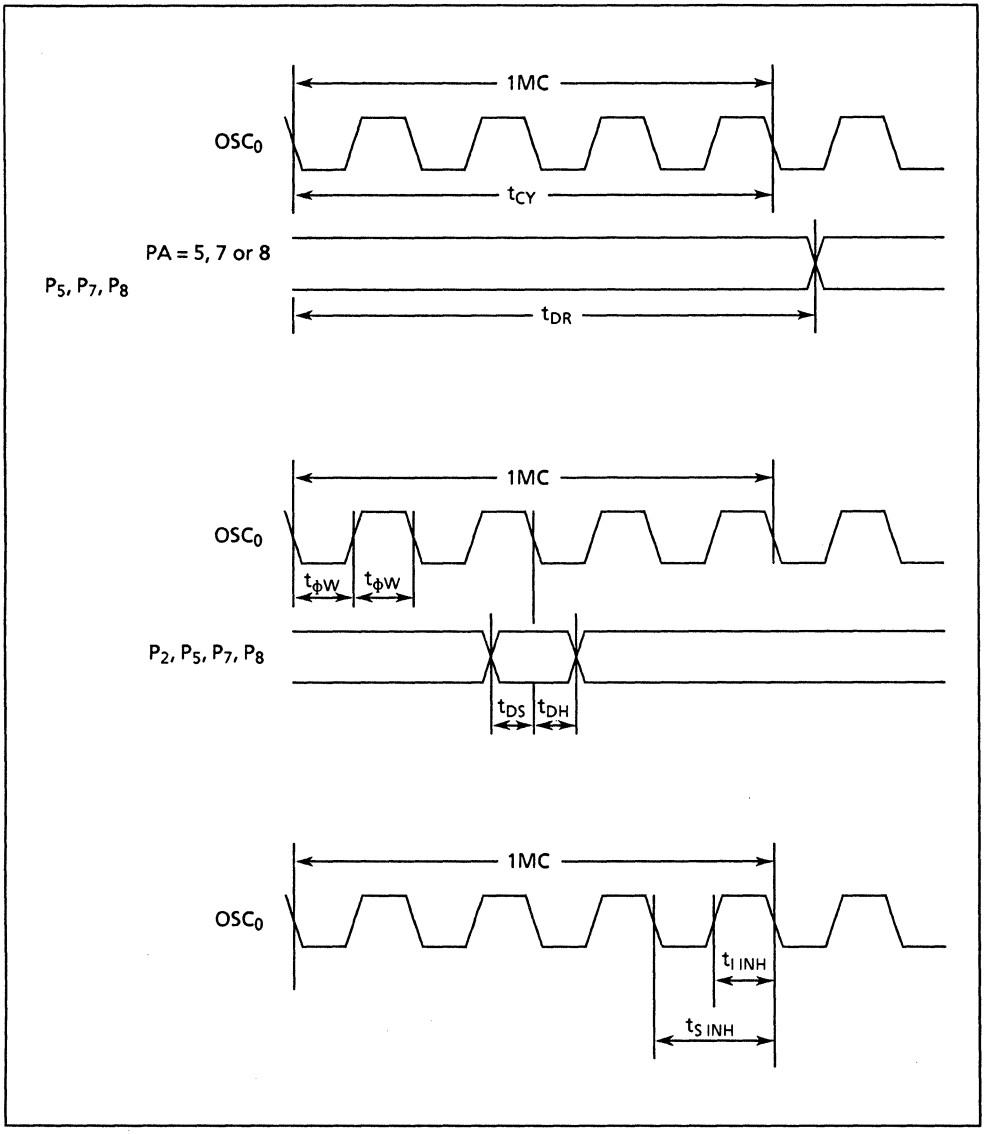
($V_{DD} = AV_{DD} = V_{RH} = 5V \pm 10\%$, GND = AGND = 0V, 1MHz ≦ f(osc) ≦ 4.2MHz, Ta = −40 to +85°C)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Resolution	–	–	8	–	–	Bits
Absolute Accuracy	–	–	–	–	± 1.5	LSB
Conversion Speed	t_{CON}	SPEED = "0"	60t cy	–	–	ns
		SPEED = "1"	120t cy	–	–	ns
Analog channel Input Voltage	VI	–	AGND	–	VRH	V
Analog channel Input Current	ILI	VI ≧ AGND VI ≦ VRH	–	–	± 1	μA
VRH Input Current	I REF	–	–	0.5	1.0	mA

(Note: t cy = 952ns (f(osc) = 4.2MHz))

TIMING CHARTS

Output Condition



OKI semiconductor

MSC6458

OKI 4-BIT 1-CHIP MICROCONTROLLER

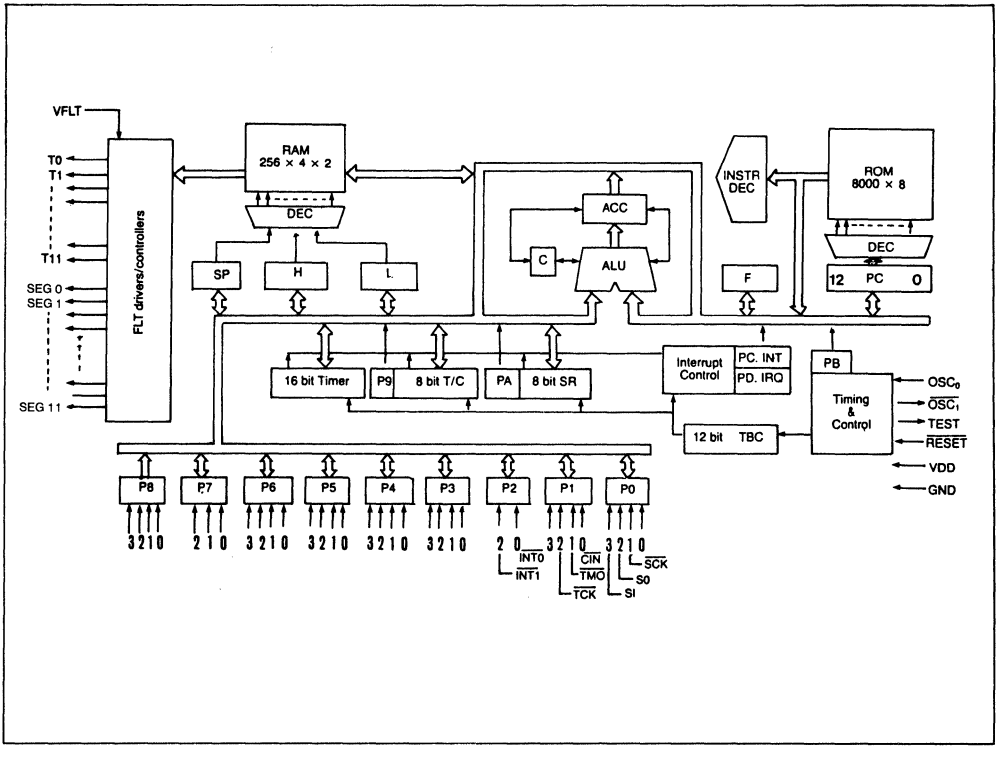
GENERAL DESCRIPTION

The MSC6458 is a high-speed, 4-bit 1-chip microcontroller with built-in FLT drivers/controllers developed to support relatively large control systems.

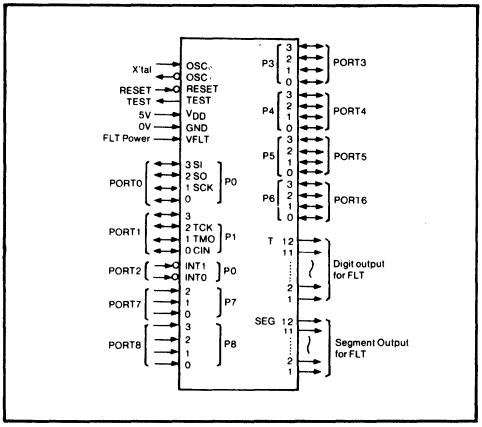
FEATURES

- ROM: 8000 × 8 bits
- RAM: 512 × 4 bits
- Ports: I/O 24 ports (8 having IOL = 20 mA)
  Input 9 (2 also serving as interrupt inputs)
- FLT drivers (Withstand voltage 40V): 12 (IOH = 20mA)
  12 (IOH = 6mA)
- LED direct drive available
- Interrupts: 7 lines (2 external, 5 internal)
- Built-in counters: 12 bits, timebase counter
  16 bits, programmable counter
  8 bits, high-speed programmable timer/event counter
- Serial I/O: Built-in 8-bit SIO register
- Oscillation circuit: Crystal or ceramic oscillation
- Number of instructions: 147
- Cycle time: 930 ns (4.3MHz)
- Operating ranges: 4.5 to 5.5V (4.3MHz)
  Voltage: 3.0 to 6.0V (1MHz)
  Temperature: −40 to +85°C
- Power dissipation (typical)
  (display off): 9mA (5V, 4.3MHz)
  2mA (3V, 1MHz)
- Power down: STOP instruction
- Package: 64-pin shrink DIP
- Package:
  64 pin plastic shrink DIP (SDIP64-P-750)

BLOCK DIAGRAM

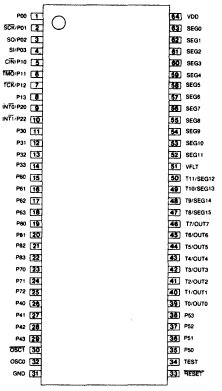


LOGIC SYMBOL



PIN CONFIGURATION (TOP VIEW)

64 PIN PLASTIC SHRINK DIP



PIN DESCRIPTION

Terminal	Input/ Output	Function	When reset
P00 P01/$\overline{SCK}$ P02/SO P03/SI	Input/ Output	I/O port I/O port (also used as serial clock input $\overline{SCK}$) I/O port (also used as serial data output SO) I/O port (also serial data input SI)	"1"
P10/$\overline{CIN}$ P11/$\overline{TMO}$ P12/$\overline{TCK}$ P13	Input/ Output	I/O port (also used as count input $\overline{CIN}$) I/O port (also used as timer output $\overline{TMO}$) I/O port (also used timer clock input $\overline{TCK}$) I/O port	"1"
P20/$\overline{INT0}$ P22/$\overline{INT1}$	Input	Input Port with Latch (falling edge sensitive) also used as interrupt input $\overline{INT0}$ Input Port with Latch ('0' level sensitive) also used as interrupt input $\overline{INT1}$	–
P30 ~ P33	Input/ Output	I/O port	"1"
P60 ~ P63	Input/ Output	I/O port	"0"
P40 ~ P43 P50 ~ P53	Output/ Input	I/O port (I_{OL}=20mA MAX)	"0"
P70 ~ P72 P80 ~ P83	Input	Input port with pull down register Pull down register of P70 ~ P72 can be removed by instruction	
SEG0 ~ SEG11	Output	FLT segment driver (dynamic)	"0"
T11/SEG12 ~ T8/SEG15	Output	FLT segment driver (dynamic)/Timing output	"0"
T7/OUT7 ~ T0/OUT0	Output	FLT segment driver (static)/Timing output	"0"
OSC0 OSC1	Input/ Output	Crystal connection terminal for system clock oscillation	–
$\overline{RESET}$	Input	System reset input	–
TEST	Output	Test pin (Open)	–
VFLT	Input	Power supply for FLT driving	–
VDD GND	Input	System Power Supply	–

FUNCTIONAL DESCRIPTION

1. ROM

The ROM, organized in 8 bits, has a maximum capacity of 8000 bytes.

2. RAM

The RAM is organized in 4 bits per word, with a capacity of 512 words.

It is separated into two banks each 256 words long. Bank selection is accomplished via internal ports. The RAM location in the banks is addressed by the H and L registers or by the second byte of each instruction.

3. Ports (24 I/O, 7 input)

The 24 pseudo-bidirectional I/O ports effect or control the exchange of data with external sources. The ports are specified by the L register or by codes contained in instructions. Ports 4 and 5 may draw IOL up to 20mA.

The seven input ports have built-in pulldown resistors. Up to 84 keys can be scanned by assembling them in key matrices with the timing outputs of the FLT drivers (with 12 segments × 12 timings on display; also during automatic display).

4. Interrupt Input Pins (2 terminals)

The $\overline{INT0}$/P20 and $\overline{INT1}$/P22 pins are interrupt input pins. External interrupt request flags of $\overline{INT0}$/P20 pin and $\overline{INT1}$/P22 pin can be set by using interrupt input pins:

$\overline{INT0}$/P20 pin ... positive edge or negative edge input.

$\overline{INT1}$/P22 pin ... "0" level input.

These flags are automatically reset when the appropriate external interrupts occur. These pins are available for use as input ports when not used as interrupt input pins.

5. FLT Drivers/Controllers (Automatic Display)

The FLT drivers have a withstand voltage of 40V in the positive direction from the GND level. They comprise 12 ports that can draw 20mA as IOH (Timing outputs) and 12 ports that can draw 6mA as such (Segment outputs).

A choice of four display modes is supported as listed below. A display RAM area is allocated as part of the RAM space. Data is automatically displayed when transferred to the display RAM. (Two different display frequencies are selectable.) Static output data can be displayed by controlling the FLT drivers by programming.

Display modes (@4.194304 MHz)

(1) 12 Segments × 12 Timings
1/12 duty (85.3/341.3 Hz)

(2) 16 Segments × 8 Timings
1/8 duty (128/512 Hz)

(3) 16 Segments × 4 Timings +4 output*
1/4 duty (256/1024Hz)

(4) 16 Segments+8 output*
Program controlled

*output: static outputs

6. Stack (STACK) and Stack Pointer (SP)

The PC is saved in the stack when an interrupt occurs or a CAL instruction is executed. It is recovered by the execution of an RT instruction.

One fourth of the RAM space (128 words maximum, 32 levels) is available as a stack area. A 4-word RAM area is used for "one" level in the stack.

The stack pointer is an 8-bit up-down counter (the MSB and 2 bits from LSB being fixed at '1') indicating the next stack address to use. It enables the RAM space to be used as a pushdown stack. Data can also be transferred between stack pointer and the H/L registers.

7. Interrupts

Seven interrupt lines are provided for eight sources and eight levels of interrupts as follows (two external inputs):

(1) Display interrupt
Update to timing signals (positive edge)

(2) External interrupt1
Negative edge on the $\overline{INT0}$/P20 pin

(3) External interrupt2
Positive edge on the $\overline{INT0}$/P20 pin

(4) External interrupt3
'0' input on the $\overline{INT1}$/P22 pin

(5) Timebase interrupt
12-Bit timebase counter overflow

(6) Timer interrupt
16-Bit timer and timer register matched signal

(7) Counter interrupt
8-Bit counter and counter register matched signal

(8) Serial/O interrupt
8-Bit shift register shift end signal

8. 12-Bit Timebase Counter

The timebase counter is made up of a 12-bit binary counter. It generates an interrupt request every time it overflows as a result of dividing the OSC0 input 2^{12}.

9. 16-Bit Programmable Timer/Event Counter

Comprising a 16-bit register, a 16-bit binary counter, a comparator circiut, and a control circuit, the programmable timer generates an interrupt request when the register and counter values are matched.

10. 8-Bit High-Speed Programmable Timmer/Event Counter

The high-speed programmable timer/event counter comprises an 8-bit register, an 8-bit binary counter, a comparator circuit, and a control circuit. Starting and stopping the counter can be controlled by instructions. It generates an interrupt request when the register and counter values are matched.

11. 8-Bit Serial I/O

Serial I/O consists of an 8-bit shift register, a 3-bit shift counter, and a control circuit. It is used for serial data input and output. Serial data input and output takes place synchronized with a shift clock, which is selectable between internal and external clocks. The shift counter automatically terminates a data transfer on counting eight shift clock pulses and generates an interrupt request.

12. Registers (Acc, H, L, F)

The accumulator (Acc) is a 4-bit register used to perform data transfers or calculations with the RAM, other registers, ports and so on.

The H and L registers are each a 4-bit register. They transfer data to and from Acc and SP (stack pointer) and address the RAM. The L register is also used to specify ports to use.

The F register is made up of four independent flip-flops. It can be used as a program "flag" or general-purpose register because each of these flip-flops permits set/reset testing and transferring 4-bit parallel data to and from Acc by instructions.

13. Timing Control (TC)

A '0' input on the $\overline{\text{RESET}}$ pin for a certain period initializes internal circuitry and ports.

As the input side of clock pulses, the OSC0 pin accepts clock pulses from an external source. Clock pulses may also be obtained by configuring an oscillation circuit with a crystal oscillator or ceramic resonator connected to OSC0 and $\overline{\text{OSC1}}$.

Load Instructions, etc.

Mnemonic		Code	Bytes	Cycles	Description
LAI	n	90–9F	1	1	A ← n
LLI	n	80–8F	1	1	L ← n
LHI	n	3E · 7n	2	2	H ← n
LHLI	nn	15 · nn	2	2	HL ← nn
LMI	nn	14 · nn	2	2	M (w) ← nn
LAL		21	1	1	A ← L
LLA		2D	1	1	L ← A
LAH		22	1	1	A ← H
LHA		2E	1	1	H ← A
LAM		38	1	1	A ← M
LMA		2F	1	1	M ← A
LAM+		24	1	1	A ← M, L ← L+1, Skip if L = "0"
LAM–		25	1	1	A ← M, L ← L–1, Skip if L = "F"
LMA+		26	1	1	M ← A, L ← L+1, Skip if L = "0"
LMA–		27	1	1	M ← A, L ← L–1, Skip if L = "F"
LAMM	n2	39–3B	1	1	A ← M, H ← H ⩛ n2
LAMD	mm	10 · mm	2	2	A ← Md
LMAD	mm	11 · mm	2	2	Md ← A
X		28	1	1	A ⟷ M
X+		3C	1	1	A ⟷ M, L ← L+1, Skip if L = "0"
X–		2C	1	1	A ⟷ M, L ← L–1, Skip if L = "F"
XM	n2	29–2B	1	1	A ⟷ M, H ← H ⩛ n2
LMT	mm	19 · mm	2	4	M (w) ← T (Md (w), A)
LAF		3E · 54	2	2	A ← F
LFA		3E · 5C	2	2	F ← A
LHLS		3E · 53	2	2	HL ← SP
LSHL		3E · 5B	2	2	SP ← HL
IP		20	1	1	A ← P
OP		23	1	1	P ← A
IPD	p	3D · pD	2	2	A ← Pp
OPD	p	3D · pC	2	2	Pp ← A
OPT		18	1	3	P4, P5 ← T (M (w), A)

Interrupt Control Instructions

Mnemonic	Code	Bytes	Cycles	Description
MEI	3E · 60	2	2	MEIF ← "1"
MDI	3E · 61	2	2	MEIF ← "0"
EIXD	3D · E8	2	2	EIXDF ← "1"
EIXU	3D · E9	2	2	EIXUF ← "1"
EIXL	3D · EA	2	2	EIXLF ← "1"
EIDP	3D · EB	2	2	EIDPF ← "1"
EITB	3D · D8	2	2	EITBF ← "1"
EITM	3D · D9	2	2	EITMF ← "1"
EICT	3D · DA	2	2	EICTF ← "1"
EISR	3D · DB	2	2	EISRF ← "1"
DIXD	3D · E4	2	2	EIXDF ← "0"
DIXU	3D · E5	2	2	EIXUF ← "0"
DIXL	3D · E6	2	2	EIXLF ← "0"
DIDP	3D · E7	2	2	EIDPF ← "0"
DITB	3D · D4	2	2	EITBF ← "0"
DITM	3D · D5	2	2	EITMF ← "0"
DICT	3D · D6	2	2	EICTF ← "0"
DISR	3D · D7	2	2	EISRF ← "0"
TIXD	3D · E0	2	2	Skip if EIXDF = "1"
TIXU	3D · E1	2	2	Skip if EIXUF = "1"
TIXL	3D · E2	2	2	Skip if EIXLF = "1"
TIDP	3D · E3	2	2	Skip if EIDPF = "1"
TITB	3D · D0	2	2	Skip if EITBF = "1"
TITM	3D · D1	2	2	Skip if EITMF = "1"
TICT	3D · D2	2	2	Skip if EICTF = "1"
TISR	3D · D3	2	2	Skip if EISRF = "1"
TQXD	3D · 20	2	2	Skip if IRQXDF = "1"
TQXU	3D · 21	2	2	Skip if IRQXUF = "1"
TQXL	3D · 22	2	2	Skip if IRQXLF = "1"
TQDP	3D · 23	2	2	Skip if IRQDPF = "1"
TQTB	3D · C0	2	2	Skip if IRQTBF = "1"
TQTM	3D · C1	2	2	Skip if IRQTMF = "1"
TQCT	3D · C2	2	2	Skip if IRQCTF = "1"
TQSR	3D · C3	2	2	Skip if IRQSRF = "1"
RQXD	3D · 24	2	2	IRQXDF ← "0"
RQXU	3D · 25	2	2	IRQXUF ← "0"
RQXL	3D · 26	2	2	IRQXLF ← "0"
RQDP	3D · 27	2	2	IRQDPF ← "0"
RQTB	3D · C4	2	2	IRQTBF ← "0"
RQTM	3D · C5	2	2	IRQTMF ← "0"
RQCT	3D · C6	2	2	IRQCTF ← "0"
RQSR	3D · C7	2	2	IRQSRF ← "0"

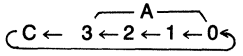
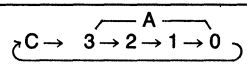
Increment/Decrement Instructions

Mnemonic	Code	Bytes	Cycles	Description
INA	30	1	1	A ← A+1, Skip if A = "0"
INL	31	1	1	L ← L+1, Skip if L = "0"
INH	32	1	1	H ← H+1, Skip if H = "0"
INM	33	1	1	M ← M+1, Skip if M = "0"
DCA	34	1	1	A ← A−1, Skip if A = "F"
DCL	35	1	1	L ← L−1, Skip if L = "F"
DCH	36	1	1	H ← H−1, Skip if H = "F"
DCM	37	1	1	M ← M−1, Skip if M = "F"
INMD mm	12 · mm	2	2	Md ← Md+1, Skip if Md = "0"
DCMD mm	13 · mm	2	2	Md ← Md−1, Skip if Md = "F"

Bit Handling Instructions, etc.

Mnemonic	Code	Bytes	Cycles	Description
TAB n2	54−57	1	1	Skip if A (n2) = "1"
RAB n2	64−67	1	1	A (n2) ← "0"
SAB n2	74−77	1	1	A (n2) ← "1"
TPB n2	50−53	1	1	Skip if P (n2) = "1"
RPB n2	60−63	1	1	P (n2) ← "0"
SPB n2	70−73	1	1	P (n2) ← "1"
TMB n2	58−5B	1	1	Skip if M (n2) = "1"
RMB n2	68−6B	1	1	M (n2) ← "0"
SMB n2	78−7B	1	1	M (n2) ← "1"
TFB n2	5C−5F	1	1	Skip if F (n2) = "1"
RFB n2	6C−6F	1	1	F (n2) ← "0"
SFB n2	7C−7F	1	1	F (n2) ← "1"
TPBD p, n2	3D · p0~3	2	2	Skip if Pp (n2) = "1"
RPBD p, n2	3D · p4~7	2	2	Pp (n2) ← "0"
SPBD p, n2	3D · p8~B	2	2	Pp (n2) ← "1"
TC	09	1	1	Skip if C = "1"
RC	08	1	1	C ← "0"
SC	07	1	1	C ← "1"

Arithmetic Instructions

Mnemonic		Code	Bytes	Cycles	Description
ADCS		01	1	1	C, A ← C+A+M, Skip if C = "1"
ADS		02	1	1	A ← A+M, Skip if Cy = "1"
ADC		03	1	1	C, A ← C+A+M
AIS	n	3E · 4n	2	2	A ← A+n, Skip if Cy = "1"
DAA		06	1	1	A ← A+6
DAS		0A	1	1	A ← A+10
AND		0D	1	1	A ← A ∧ M
OR		05	1	1	A ← A ∨ M
EOR		04	1	1	A ← A ⊻ M
CMA		0B	1	1	A ← $\overline{A}$
CIA		0C	1	1	A ← $\overline{A}$+1
RAL		0E	1	1	
RAR		0F	1	1	
CAM		16	1	1	Skip if A = M
CAI	n	3E · 0n	2	2	Skip if A = n
CMI	n	3E · 1n	2	2	Skip if M = n
CLI	n	3E · 2n	2	2	Skip if L = n
CPI	p, n	17 · pn	2	2	Skip if Pp = n

Branch Instructions, etc.

Mnemonic		Code	Bytes	Cycles	Description
JCP	a6	C0–FF	1	1	PC ← a6
JA		1A	1	2	PC ← (PC ← A) +1
JM		1B	1	2	PC ← (M (w), A)
JP	a12	40_4F 00‾FF	2	2	PC ← a12
CAL	a12	A0_AF 00‾FF	2	4	ST ← PC+2, PC ← a12, SP ← SP−4
CZP	a	Ba	1	4	ST ← PC+1, PC ← 2a, SP ← SP−4
LJP	a13	3F 3F 00–1F 00 FF	3	4	PC ← a13
LCAL	a13	3F 3F 80–9F 00 FF	3	4	ST ← PC+3, PC ← a13, SP ← SP−4
RT		1E	1	4	PC ← ST, SP ← SP+4
RTS		1F	1	4	PC ← ST, SP ← SP+4, then Skip

Counter Control Instructions, etc.

Mnemonic	Code	Bytes	Cycles	Description
LCTM	3E · 51	2	2	CTR ← M (w)
LMCT	3E · 59	2	2	M (w) ← CT
ECT	3D · BB	2	2	CTF ← "1" (Counter Start)
DCT	3D · B7	2	2	CTF ← "0" (Counter Stop)
TCT	3D · B3	2	2	Skip if CTF = "1"
LTMM	3E · 50	2	3	TMR ← M (2w)
LMTM	3E · 58	2	3	M (2w) ← TM
LSRM	3E · 52	2	2	SR ← M (w), SC ← "0" SC: Shift Counter
LMSR	3E · 5A	2	2	M (w) ← SR
ESR	3D · BA	2	2	SRF ← "1" (Shift Register Start)
DSR	3D · B6	2	2	SRF ← "0" (Shift Register Stop)
TSR	3D · B2	2	2	Skip if SRF = "1"

CPU Control Instructions, etc.

Mnemonic	Code	Bytes	Cycles	Description
PUSH	1C	1	3	ST ← C, A, H, L, SP ← SP−4
POP	1D	1	3	C, A, H, L ← ST, SP ← SP+4
HALT	3D · B8	2	2	Halt CPU
STOP	3D · B9	2	2	Stop CPU
NOP	00	1	1	No Operation

Explanations of Instruction Symbols

A	: Accumulator (4-bit)
H	: H register (4-bit)
L	: L register (4-bit)
F	: F register (4-bit)
M	: RAM word addressed by the H and L registers
Md	: RAM word addressed by second byte of an instruction code
M (w)	: Two RAM words addressed by the H and L register/H3-0 and L3-1 (8-bit)
Md (w)	: Two RAM words addressed by second byte of an instruction code (8-bit)
M (2w)	: Four RAM words addressed by the H and L register/H3-0 and L3-2 (16-bit)
ST	: Four RAM words (16-bit) allocated as a stack area
SP	: Stack pointer (8-bit)
PC	: Program counter
P	: Port specified by the L register (4-bit)
Pp	: Port specified by 4 high-order bits of second byte of an instruction code (4-bit)
CTR	: 8-Bit counter/register
CT	: 8-Bit programmable counter
CTF	: Programmable counter start flag
TMR	: 16-Bit timer/register
TM	: 16-Bit programmable timer
SR	: 8-Bit shift register
SRF	: Shift register start flag
(X, Y)	: ROM address data specified by a11-4 as X and a3-0 as Y (12-bit)
T (X, Y)	: ROM table data specified by a11-4 as X and a3-0 as Y (8-bit)
n	: Immediate data (4-bit)
nn	: Immediate data (8-bit)
n2	: Two low-order bits of an instruction code
(n2)	: Bit specified by the two low-order bits of an instruction code
a	: ROM address data
aX	: ROM address data (X-bit)
mm	: RAM address data (8-bit)
C	: Carry flag
Cy	: Flag indicating a carry in a calculation result

ELECTRIC CHARACTERISTICS

• Absolute Maximum Ratings

Parameter	Symbol	Conditions		Limits	Unit
Supply Voltage	V_{DD}			−0.3 ~ 7	V
Indicated Supply Voltage	V_{FLT}	Ta = 25°C		V_{DD} ~ 45	V
Input Voltage	V_I			−0.3 ~ V_{DD}	V
Input Voltage	V_O	Ta = 25°C	Input/output	−0.3 ~ V_{DD}	V
			Indicated output	−0.3 ~ V_{FLT}	V
"H" Output Current (Indicated Output)	I_{OH}	Per pin	SEG0 ~ SEG1	10	mA
			T0 ~ T11	40	mA
			OUT0 ~ OUT7	* 30	mA
		Output terminal total	SEG0 ~ SEG11	72	mA
			T0 ~ T11	72	mA
"L" Output Current (P4, P5)	I_{OL}	Per terminal		20	mA
		P4 total		40	mA
		P5 total		40	mA
Power Dissipation	P_D	Per package		600	mW
		Per input/output terminal		50	mW
Storage Temperature	Tstg	−		−55 ~ +150	°C

* When timing output is used as static output

• Operating Conditions

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	f (osc) ≦ 4.3MHz	4.5 ~ 5.5	V
		f (osc) ≦ 1MHz	3 ~ 6	V
Indicated Supply Voltage	V_{FLT}	−	10 ~ 40	V
Memory Retension Voltage	V_{DDH}	Oscillation off	2 ~ 6	V
Operating Temperature	Topr	−	−40 ~ +85	°C
(Fan Out (Input/Output Port)	N	MOS Load	15	−
		TTL Load	1	−

• DC Characteristics

(V_{DD} = 5V ±10%, Ta = −40 ~ +85°C)

Parameter	Terminal applied	Symbol	Conditions	Min.	Typ.	Max.	Unit
"H" Input Voltage	*1	V_{IH}	−	2.4	−	V_{DD}	V
	OSC0, RESET		−	3.8	−	V_{DD}	V
	P7, P8		−	3.4	−	V_{DD}	V
"L" Input Voltage	*2	V_{IL}	−	0	−	0.8	V
	P7, P8		−	0	−	1.6	V
"H" Output Voltage	*3	V_{OH}	IO = −15μA	4.2	−	−	V
	SEG0 ~ SEG11		IO = −6mA	V_{FLT}-2.5	−	−	V
	T0 ~ T11		IO = −20mA	V_{FLT}-3.5	−	−	V
"L" Output Voltage	P0, P1, P3, P6	V_{OL}	IO = 1.6mA		−	0.4	V
	P4, P5		IO = 10mA		−	0.8	V
	OSC1		IO = 15μA		−	0.4	V
	SEG0 ~ SEG11		IO = 1mA		−	1.6	V
	T0 ~ T11		IO = 1mA		−	1.4	V
"H" Input Current	OSC0	I_{IH}	VI = VDD		−	15	μA
	P2, RESET				−	1	μA
	P7(P73=0), P8				−	60	μA
	P7(P73=1)				−	1	μA

Parameter	Terminal applied	Symbol	Conditions	Min.	Typ.	Max.	Unit
"L" Input Current	OSC0	I_{IL}	VI = 0V	–	–	–15	μA
	P2, $\overline{RESET}$			–	–	–30	μA
	P7, P8			–	–	–1	μA
"H" Output Current	P0, P1, P3, P4, P5, P6	I_{OH}	VO = 2.4V	–0.1	–	–	mA
			VO = 0.4V	–	–	–1.2	mA
Current Consumption		I_{DD}	No load f (osc) = 4.3MHz	–	12	20	mA
Current Consumption (When stop mode condition)		I_{DDS}	No load	–	1	100	μA
			No load V_{DD} = 2V Ta = 25°C	–	0.5	10	μA
Current Consumption (FLT driver section)		I_{FLT}	No load All FLT driver, "L" level	–	2	100	μA

*1. Applied to P0, P1, P2, P3, P4, P5, P6
*2. Applied to P0, P1, P2, P3, P4, P5, P6, OSC0, RESET
*3. Applied to P0, P1, P3, P4, P5, P6, OSC1

• AC Characteristics

(V_{DD} = 5V ±10%, Ta = 40 ~ +85°C)

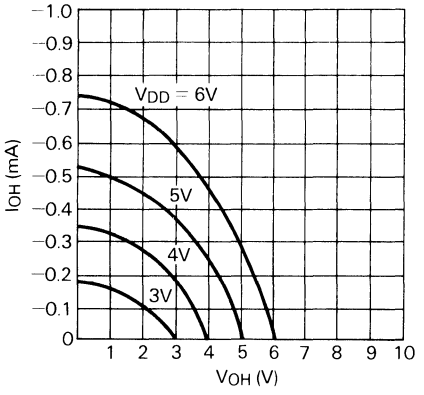
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Clock (O.S.C $_0$) Pulse Width	$t_{\phi W}$	–	116	–	–	nS
Cycle Time	t_{CY}	–	928	–	–	nS
Input Data Setup Time	t_{DS}	–	120	–	–	nS
Input Data Hold Time	t_{DH}	–	120	–	–	nS
P2 Input Data Pulse Width	t_{DWP2}	Note 1	120	–	–	nS
SR Clock. Pulse Width	t_{DW1}	–	120	–	–	nS
CT Clock. Pulse Width	t_{DW2}	–	2/8t_{CY}+120	–	–	nS
TM Clock. Pulse Width	t_{DW3}	–	t_{CY}+120	–	–	nS
SR Data Setup Time	t_{SS}	–	120	–	–	nS
SR Data Hold Time	t_{SH}	–	120	–	–	nS
SR Clock Invalid Time *	t_{SINH}	–	2/8tcy	–	–	nS
Data Delay Time	t_{DR}	C_L = 15pF	–	–	300	nS
SR Clock Delay Time	t_{SP}	C_L = 15pF	–	–	360	nS
Reset Input. Rise Time	t_{WRS}	Note 2	2tcy	–	–	nS
Segment Output. Rise Time	t_{TLHS}	VFLT = 40V	–	–	3	μSS
Segment Output. Rise Time	t_{THLS}	C_{LD} = 15pF	–	–	1	μS
Timing Output. Rise Time	t_{TLHT}	VFLT = 40V	–	–	3	μS
Timing Output. Rise Time	t_{THLT}	C_{LD} = 15pF	–	–	1	μS

*1. When stop mode is to be released by "L" level input from P20/INT0, it is necessary to keep the pulse width of more than oscillation stability time for OSC_0.
*2. This indicates when OSC_0 oscillation is stabilized. However, when stop mode is released by reset input, the pulse width of more than OSC_0 oscillation stability time as requested.
*3. t_{SINH}: When shift register commands LMSR during shift in operation, its inner part will not change if clock, which inputs P01/SCK during t_{SINH} period, changes.

STANDARD CHARACTERISTICS

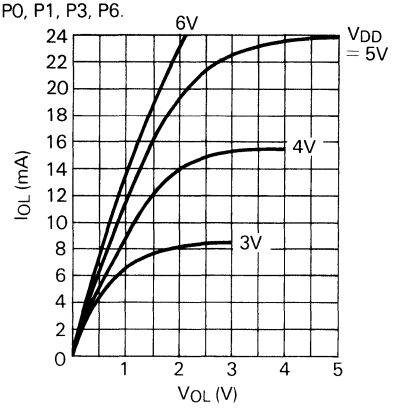
- **"H" Output Current I_{OH} – Output Voltage V_{OH} Characteristics (Ta = 25°C)**

P0, P1, P3, P4, P5, P6

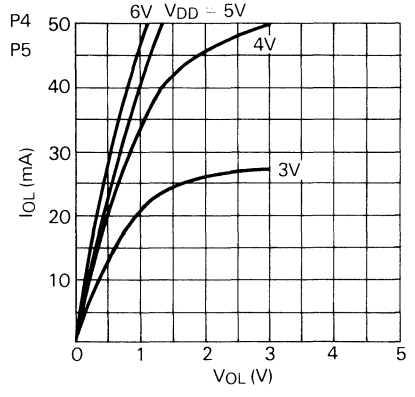


- **"L" Output Current I_{OL}– Output Voltage V_{OL} Characteristics (Ta = 25°C)**

P0, P1, P3, P6.

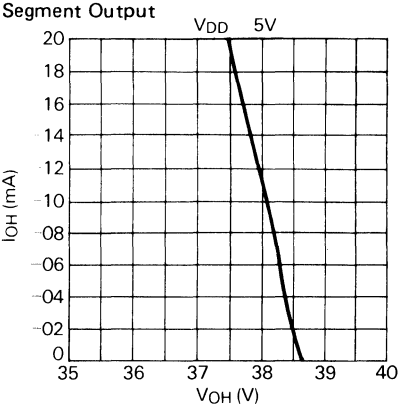


- **"L" Output Current I_{OL}– Output Voltage V_{OL} Characteristics (Ta = 25°)**



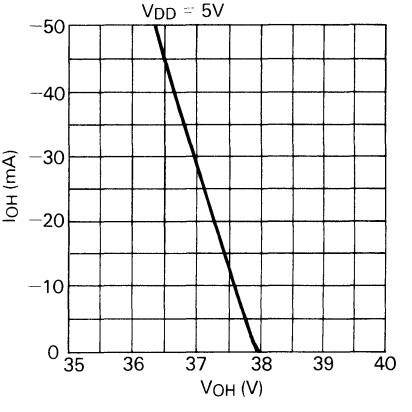
- **"H" Output Current I_{OH}– Output Voltage V_{OH} Characteristics (Ta = 25°C, VFLT = 40V)**

Segment Output



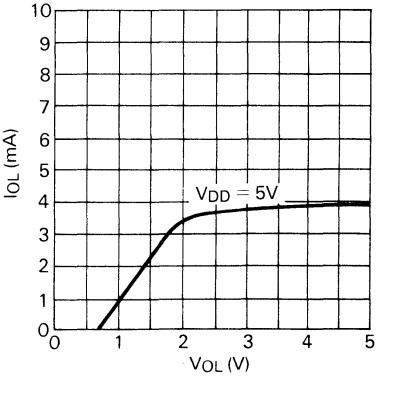
- **"L" Output Current I_{OH}– Output Voltage V_{OL} Characteristics (Ta = 25°C, VFLT = 40V)**

Timing Output



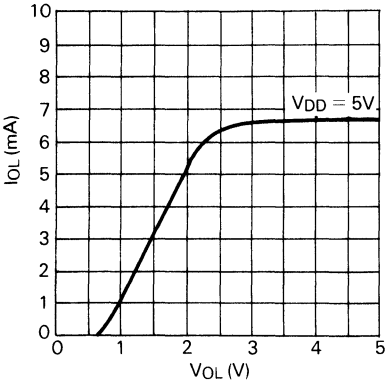
- **"L" Output Current I_{OL}– Output Output Current V_{OL} Characteristics (Ta = 25°C, VFLT = 40V)**

Segment Output

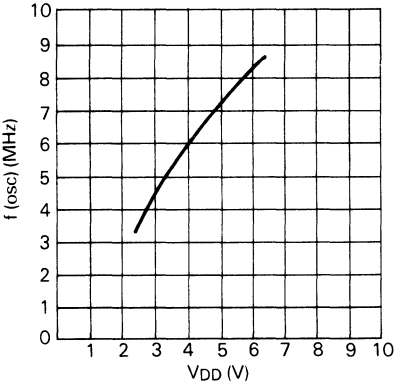


• **"L" Output Current I_{OL} – Output Current V_{OL} Characteristics (Ta = 25°C, VFLT = 40V)**

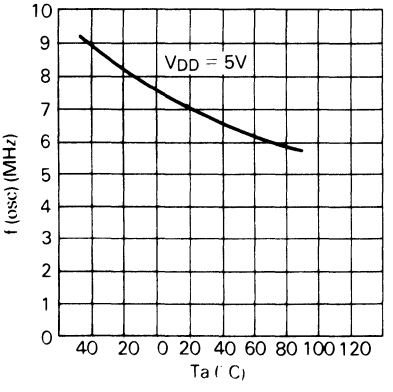
Timing Output



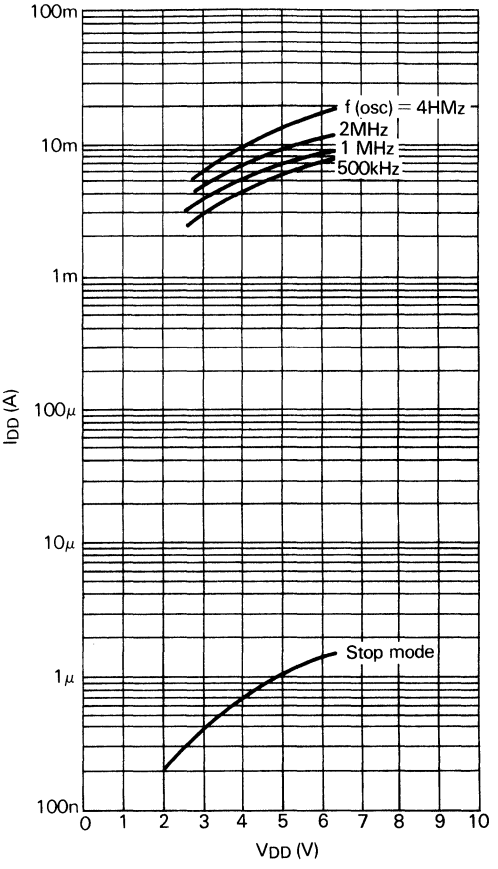
• **Maximum Clock Freqency f(osc) – Supply Voltage V_{DD} Characteristics ($Ta = 25°C$, C_L = 15pF)**



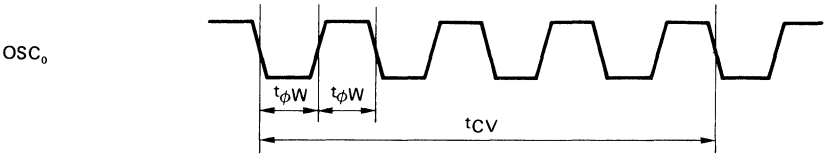
• **Maximum Clock Frequency f(osc) – Ambient Temperature Ta (V_{DD} = 5V, C_L = 15pF)**

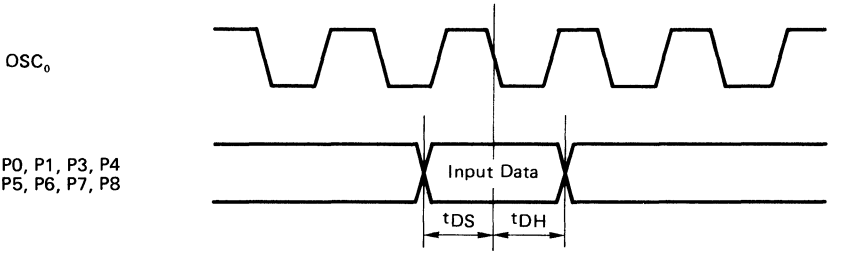


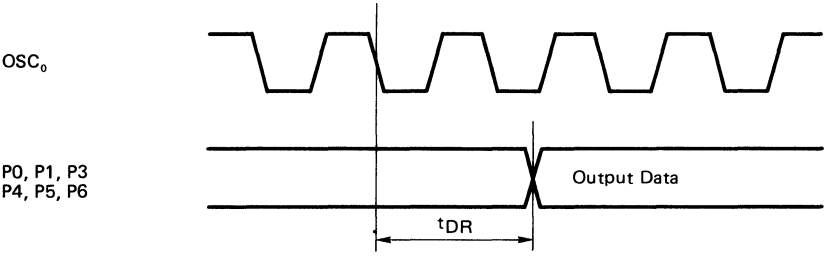
• **Current Consumption I_{DD} – Supply Voltage V_{DD} (Ta = 25°C, No load)**

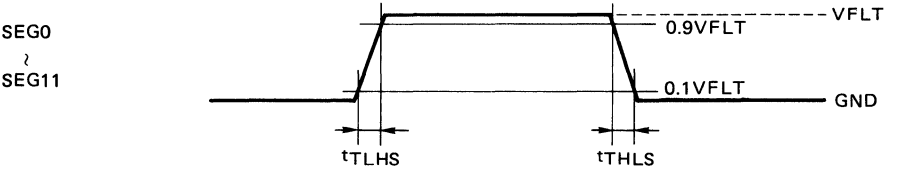


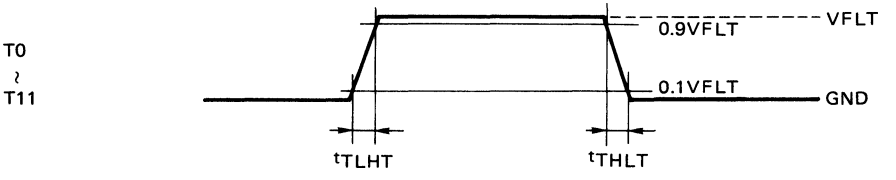
TIMING CHART

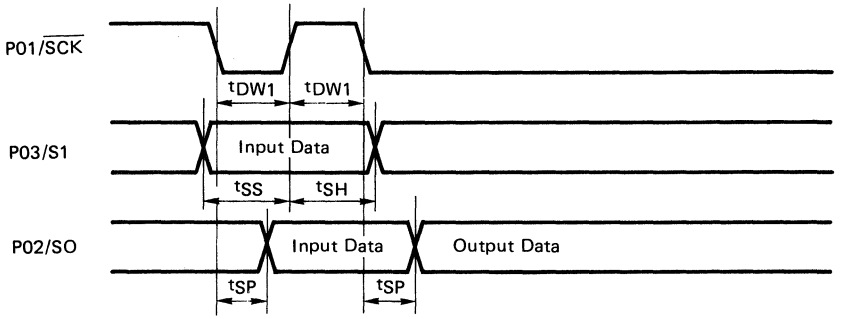


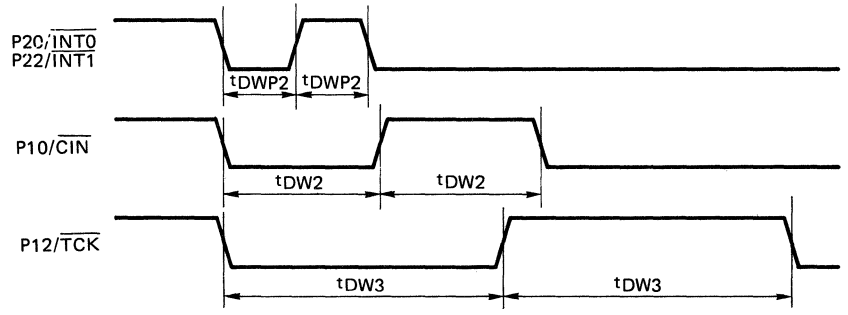


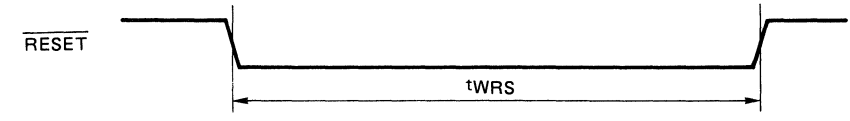


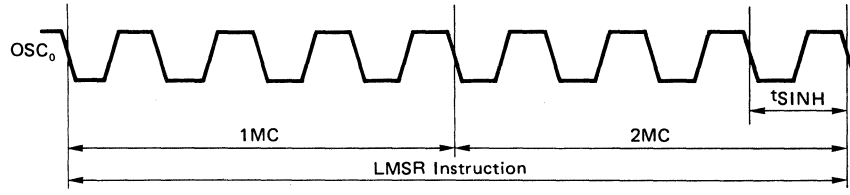





P01/SCK
tDW1
tDW1
P03/S1
Input Data
tSS
tSH
P02/SO
Input Data
Output Data
tSP
tSP


P20/INT0
P22/INT1
tDWP2
tDWP2
P10/CIN
tDW2
tDW2
P12/TCK
tDW3
tDW3


RESET
tWRS


OSC0
tSINH
1MC
2MC
LMSR Instruction

MSC6458VS

MSC6458 PIGGY BACK

GENERAL DESCRIPTION

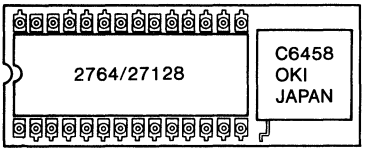
The MSC6458VS is a device whose built-in ROM is replaced by external EPROM using the piggy-back method. This device can be used for the evaluation of program.

FEATURES

- Supply Voltage: 5V±5%
- Frequency: DC ~ 4.3MHz
- 64 pin ceramic piggy back (ADIP64-C-750)
- Operating Temperature: 0 ~ 70°C

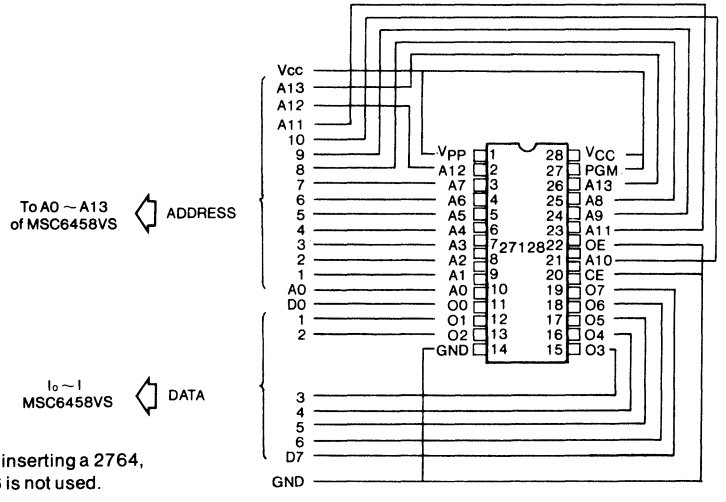
ROM INSERTION

Please refer to drawing below.



PIN CONFIGURATION

Pin Connection between MSC6458 VS and EPROM



Note: When inserting a 2764, pin 26 is not used.

MSM6502/6512

CMOS 4 BIT SINGLE CHIP MICROCONTROLLER WITH LCD DRIVER

GENERAL DESCRIPTION

The OKI MSM6502/6512 is a low-power, high-performance 4 bit single-chip microcontroller implemented in complementary metal oxide semiconductor technology.

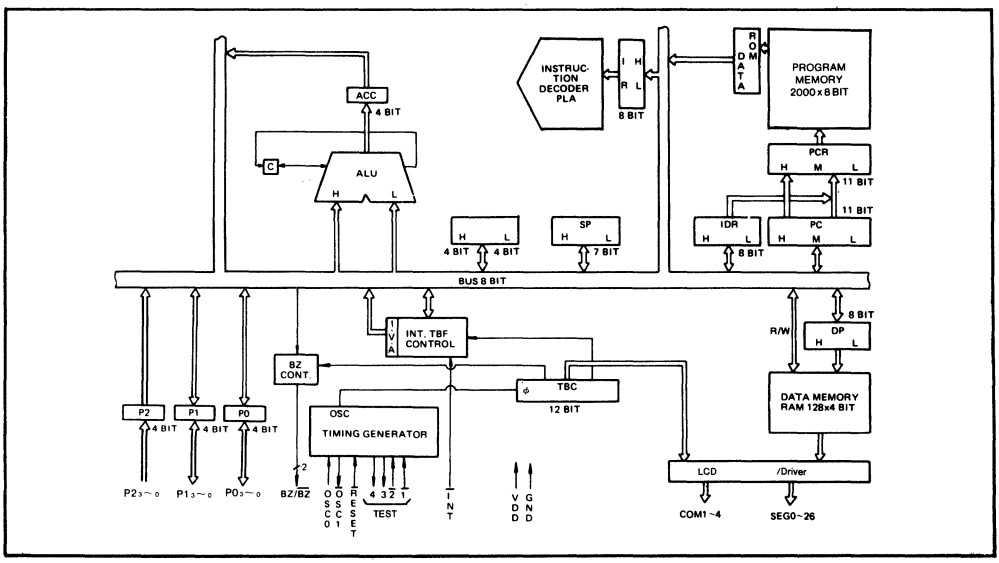
Integrated within this one chip is a 108 (4×27) dot LCD Driver. Also integrated in this chip are 16K bits of mask program ROM, 512 bits of data RAM, Input/Output lines, a programmable timer/counter, and oscillator.

The advantages of the MSM6502/6512 in comparison with the OLMS-40 Series include, among other features, a lower drive voltage, multiplexed interrupts, a larger number of drivable liquid crystal elements, a buzzer output, and larger memories.

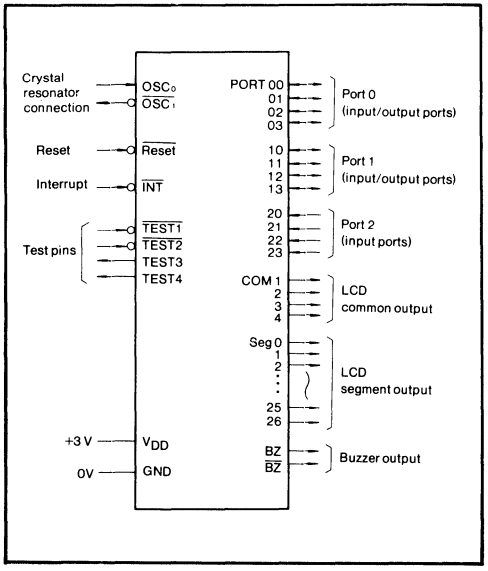
FEATURES

- ROM : 2000 × 8 bit
- RAM : 128 × 4 bit
- Number of Instructions : 68
- Clock Oscillation : Crystal 32.768 kHz
- Cycle Time : 91.5 μs
- Timer Interrupt : Dual (16 & 128 Hz)
- I/O Ports : 4 bit × 2 Port
- Input ports : 4 bit × 1 Port
- LCD Drive : 108 (4×27) picture elements
- Buzzer : 2K/1K/512 Hz/Soft
- Interrupt : Three types (external, two timer types)
- Stack : Nesting RAM stack pointer = 7 bits
- Power down : Halt mode available
- Operating power supply voltage : 2.4V – 3.6V
- Consumption current (V_{DD}=3V, OSC = 32.768 kHz) : MSM6502 : 45μA (Typical) 30μA (Halt mode) MSM6512 : 30μA (Typical) 12μA (Halt mode)
- Package: 56(S) pin plastic QFP (QFP56-P-910-K)

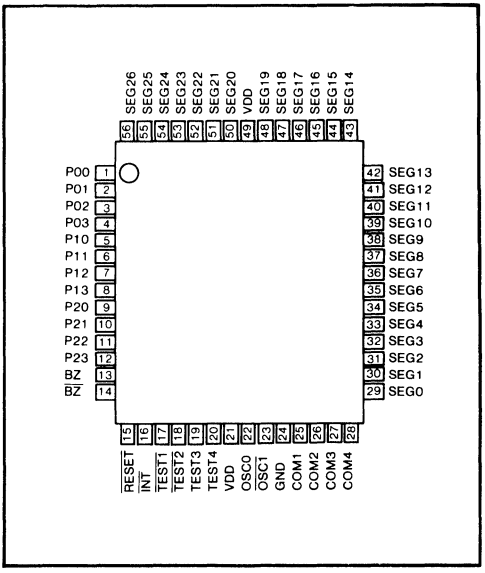
FUNCTIONAL BLOCK DIAGRAM



LOGIC SYMBOL



PIN CONFIGURATION (TOP VIEW)



PIN DESCRIPTION

Designation	Pin No.	Function
GND	24	Circuit GND potential
V_{DD}	21, 49	Main power source
OSC_0	22	Crystal OSC input, internal clock input
$\overline{OSC_1}$	23	Crystal OSC input, internal clock output
P0, P1	1 to 4 5 to 8	Pseudo-bidirectional ports for 4-bits parallel I/O. To input data from these ports, it is necessary to write "1" beforehand. The port to be selected is specified by the L register. The register contents and the corresponding specified ports are listed below. Content of L register → Port Specified: 0,8 → P0 1,9 → P1 2,0AH → P2 3,0BH → P3 4,0CH → P4 5,0DH → P5 6, 7, OEH, OFH → No designation Note: P3, P4, P5 are internal ports.
P2	9 to 12	Input ports for 4-bit parallel input with no latching function.
$\overline{INT}$	16	Input pin to request an interrupt from the external circuit. The input flag is set by the fall of the input signal.

Designation	Pin No.	Function
$\overline{\text{RESET}}$	15	The reset mode starts after "0" is input to the $\overline{\text{RESET}}$ pin for more than 2 machine cycles. The reset signal has priority over all of other signals and performs the following operations automatically: (1) Resets all bits of the PC (Program counter) to "0". (2) Sets all bits of the parallel I/O ports (P00 to P13) to "1". (3) Resets the internal resister (H, L, ACC, C, P3, P4, P5). (4) Resets the skip flag. (5) Resets all bits of the time base counter (TBC). (6) Resets the interrupt request flag (IRQF). (7) Resets the interrupt enable flag (EIF). (8) Resets the master interrupt enable flag (MEIF). (9) Sets all bits of the stack pointer (SP) to "1". (10) Initializes the segment and common outputs. (11) Sets all bits of the index register (IDR) to "1". Since the $\overline{\text{RESET}}$ pin is pulled up to V_{DD} by an internal resister (approx. 800 kΩ), it is possible to achieve power ON/reset by connecting it with an external capacitor.
LCD Drive Pins SEG 0 ~ 26 COM 1 ~ 4	 29 to 48 50 to 56 25 to 28	A special AC waveform designed to comply with liquid crystal properties is required for liquid crystal drive purpose. The MSM6502 is equipped with a 1/4 duty 1/3 bias liquid crystal drive circuit with four common output ports and 27 segments to enable displays of 108 picture elements. On/off selection of picture element displays involves writing "0" or "1" in the corresponding bits in the RAM 00H to thru 1AH display area, and subsequent automatic hardware controlled display. The frame frequency is 64 Hz.
BZ/$\overline{\text{BZ}}$	13, 14	BZ and $\overline{\text{BZ}}$ are used in the generation of alarms and other sounds. The selectable frequencies include three hardware frequencies (TBC output) 512, 1024, and 2048 Hz, and a software type based on P50 data. These frequencies are selected at P3. When one of the hardware frequencies is selected by P3, output of that frequency is continuous. But selection of the software type results in output of P50 contents to generate a melody by program.

INSTRUCTION LIST

Grouping	Mnemonic	Code	Byte	Cycle	Function
Load	LAI n	8n	1	1	Acc ← n
	LLI n	7n	1	1	L ← n
	LHLI n8	6A n8	2	2	HL ← n8
	LXI n8	69 n8	2	2	X ← n8
	LAM	B0	1	1	Acc ← M<HL>
	LAL	B1	1	1	Acc ← L
	LAH	B2	1	1	Acc ← H
	LMA	B4	1	1	M<HL> ← Acc
	LLA	B5	1	1	L ← Acc
	LHA	B6	1	1	H ← Acc
	LMAD m7	1B m7	2	2	M<m7> ← Acc

INSTRUCTION LIST (CONT.)

Grouping	Mnemonic	Code	Byte	Cycle	Function
Load	LAMD m7	1A m7	2	2	Acc ← M<m7>
	LMT	67	1	2	M<HL+1><HL> ← ROM<X>
	PUSH HL	BC	1	2	STACK ← HL, SP ← SP−2
	PUSH CA	BD	1	2	STACK ← C, Acc, SP ← SP−2
	POP HL	BE	1	2	HL ← STACK, SP ← SP+2
	POP CA	BF	1	2	C, Acc ← STACK, SP ← SP+2
Exchange	XAM	B8	1	1	Acc ←→ M<HL>
	XAMD m7	1C m7	2	2	Acc ←→ M<m7>
	XHS	3F	1	1	HL ←→ SP
Increment and decrement	INA	10	1	1	Acc ← Acc + 1, Skip if Acc=0
	INL	11	1	1	L ← L+1, Skip if L=0
	INM	12	1	1	M<HL> ← M<HL> +1, Skip if M<HL> = 0
	INX	5C	1	1	X ← X + 1
	DCA	14	1	1	Acc ← Acc−1, Skip if Acc= F
	DCL	15	1	1	L ← L − 1 Skip if L= F
	DCM	16	1	1	M<HL> ← M<HL> −1, Skip if M<HL> = F
	DCX	5D	1	1	X ← X − 1
Operation	DSC	60	1	1	C,Acc ← C+Acc+$\overline{M}$<HL>, Adjust if C=0
	DAC	61	1	1	C,Acc ← C+Acc+(M<HL> +6), Adjust if C= 0
	ADS	62	1	1	Acc ← Acc+M<HL>, Skip if Cy=1
	ADCS	63	1	1	C,Acc ← C+Acc+M<HL>,Skip if C=1
	AIS n	0n	1	1	Acc ← Acc+n, Skip if Cy=1
	CMA	65	1	1	Acc ← $\overline{Acc}$
	EOR	66	1	1	Acc ← Acc∀M<HL>
	AND	4C	1	1	Acc ← Acc∧M<HL>
	OR	4D	1	1	Acc ← Acc∨M<HL>
	CAM	6B	1	1	Skip if Acc = M<HL>
	CPAL	6C	1	1	Skip if Acc = L
	CAXL	6D	1	1	Skip if Acc = XL
	CAXH	6E	1	1	Skip if Acc = XH
	SC	1F	1	1	C ← "1"
	RC	1E	1	1	C ← "0"
	TC	4E	1	1	Skip if C= "1"

INSTRUCTION LIST (CONT.)

Grouping	Mnemonic	Code	Byte	Cycle	Function
Operation	RAL	18	1	1	(*) ← C ← Acc 3←2←1←0←(*)
	RAR	19	1	1	(*) → C → Acc 3→2→1→0→(*)
Bit Operation	SMB n2	30 ~ 33	1	1	M<HL> bit n2 ← "1"
	SMBD m7, n2	50 m7 ~ 53 m7	2	2	M<m7> bit n2 ← "1"
	SPB n2	20 ~ 23	1	1	P bit n2 ← "1"
	RMB n2	34 ~ 37	1	1	M<HL> bit n2 ← "0"
	RMBD m7, n2	54 m7 ~ 57 m7	2	2	M<m7> bit n2 ← "0"
	RPB n2	24 ~ 27	1	1	P bit n2 ← "0"
	TMB n2	38 ~ 3B	1	1	Skip if M<HL> bit n2 = "1"
	TMBD m7, n2	58 m7 ~ 5B m7	2	2	Skip if M<m7> bit n2 = "1"
	TPB n2	28 ~ 2B	1	1	Skip if P bit n2 = "1"
	TAB n2	2C ~ 2F	1	1	Skip if Acc bit n2 = "1"
	TIRB n2	49 ~ 4B	1	1	if IRQF bit n2="1" Skip & IRQF bit n2 ← "0"
Interrupt	EI	5E	1	1	MEIF ← "1"
	DI	5F	1	1	MEIF ← "0"
Branch	J a11	9000~97CF	2	2	$PC_{10 \sim 0}$ ← a11
	CAL a11	A000~A7CF	2	3	STACK←PC+2, SP←SP-3, $PC_{10 \sim 0}$ ←a11
	JCP a6	C0~FF	1	1	$PC_{5 \sim 0}$ ←a6
	RT	3C	1	2	PC ← STACK, SP←SP+3
	RTS	3D	1	1	PC ← STACK, SP←SP+3, then Skip
Input/ Output	IP	B3	1	1	Acc ← P
	OP	B7	1	1	P ← Acc
CPU Control	HALT	4F	1	1	HLF ← "1"
	NOP	00	1	1	No Operation

ELECTRICAL CHARACTERISTIC

ABSOLUTE MAXIMUM RATING

Parameter	Symbol	Conditions	Limits	Unit
Power Supply Voltage	V_{DD}	Ta = 25°C	−0.3 ~ 7	V
Input Voltage	V_I		−0.3 ~ V_{DD}	V
Output Voltage	Vo		−0.3 ~ V_{DD}	V
Power Dissipation	P_D	Ta = 25°C per package	200	mV
StorageTemperature	Tstg	—	−55 ~ +150	°C

OPERATING RANGE

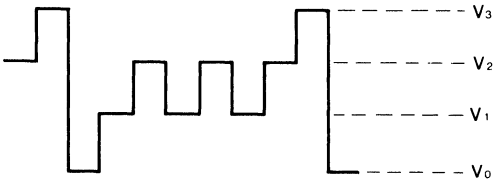
Parameter	Symbol	Conditions	Limits	Unit
Power Supply Voltage	V_{DD}	f(osc)=32.768 kHz	2.4 ~ 3.6	V
Operating Temperature	Top	—	−20 ~ +70	°C

D.C. CHARACTERISTICS

(V_{DD} = 3V, Ta = −20 ~ +70°C)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
"H" Input Voltage	V_{IH}	–		2.6	–	–	V
"L" Input Voltage	V_{IL}	–		–	–	0.4	V
"H" Output Voltage(1)	V_{OH}	I_O = −1.0 mA		2.0	–	–	V
"L" Output Voltage(2)	V_{OL}	I_O = 1.0 mA		–	–	1.0	V
LCD Drive Output Voltage(3)	V_3	MSM6502	I_O = −5 μA	2.8	–	3.0	V
		MSM6512	I_O = −2 μA				
	V_2	MSM6502	I_O = ±2 μA	1.8	–	2.2	V
		MSM6512	I_O = ±0.5 μA				
	V_1	MSM6502	I_O = ±2 μA	0.8	–	1.2	V
		MSM6512	I_O = ±0.5 μA				
	V_0	MSM6502	I_O = 5μA	0.0	–	0.2	V
		MSM6512	I_O = ±2 μA				
OSC_0 Input Current	I_{IH}/I_{IL}	$V_I = V_{DD}/V_I$ = 0V		–	–	2/−2	μA
Input Current(4)	I_{IH}/I_{IL}	$V_I = V_{DD}/V_I$ = 0V		–	–	1/−10	μA
Input Current(5)	I_{IH}/I_{IL}	$V_I = V_{DD}/V_I$ = 0V		–	–	1000/−1	μA
Input Current(6)	I_{IH}/I_{IL}	$V_I = V_{DD}/V_I$ = 0V		–	–	1/−40	μA
P0, P1 "H" Output Current	I_{OH}	V_O = 0 V		–	–	−50	μA
Current Consumption	I_{DD}	MSM6502	f(osc) = 32.768 kHz at no load	–	45	70	μA
		MSM6512		–	30	55	
	I_{DDHLT}	MSM6502	f(osc) = 32.768 kHz at HLT execution	–	30	40	μA
		MSM6512		–	12	25	
	I_{DDS}	MSM6502	Statis	–	15	25	μA
		MSM6512		–	5	15	
Oscillation Start Time	T_{OSC}	–		–	–	10	SEC

(1) Applied to BZ, $\overline{BZ}$
(2) Applied to BZ, $\overline{BZ}$, P0, P1
(3) Applied to COMMON, SEGMENT



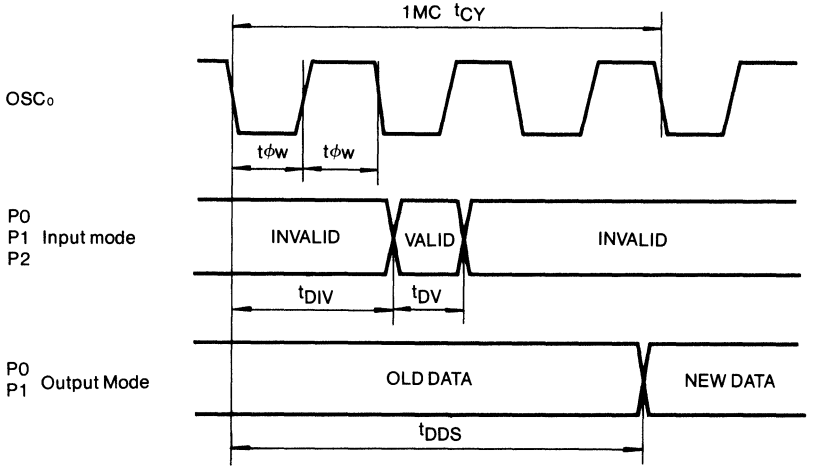
(4) Applied to $\overline{RESET}$, $\overline{INT}$
(5) Applied to P2 (When input is unable)
(6) Applied to P2 (When input is able)

SWITCHING CHARACTERISTIC

(V_{DD} = 3V, Ta = −20 ~ +70°C)

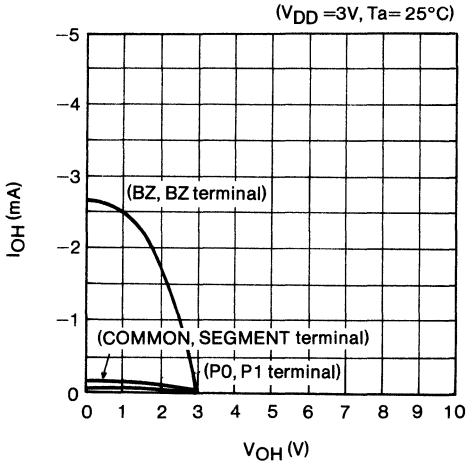
Parameter	Symbol	Conditions	MIN.	TYP	MAX.	Unit
Clock (OSC_0) Pulse Width	$t\phi W$	–	15	–	–	μS
Cycle Time	t_{CY}	–	(1)	–	–	μS
P0 P1 P2 Data Valid Time	t_{DV}	–	(2)	–	–	μS
P0 P1 P2 Data Invalid Time	t_{DIV}	–	–	–	(3)	μS
P0 P1 Data Delay Time	t_{DDS}	C_L = 50 pF	–	–	(4)	μS

(1) $t_{CY} = 3 \times 1/f(osc)$
(2) $t_{DV} = 1/2 \times 1/f(osc)$
(3) $t_{DIV} = 1 \times 1/f(osc) + 10\mu S$
(4) $t_{DDS} = 5/2 \times 1/f(osc) + 15\mu S$

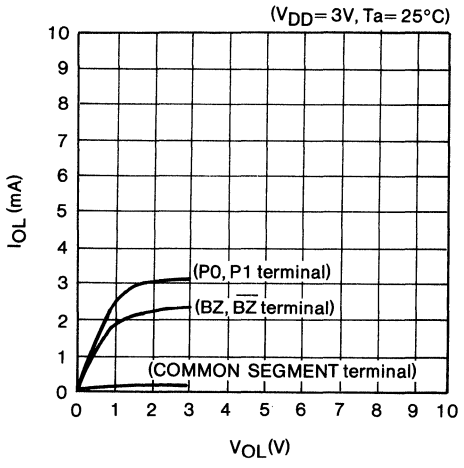


TYPICAL PERFORMANCE CURVES for MSM6502

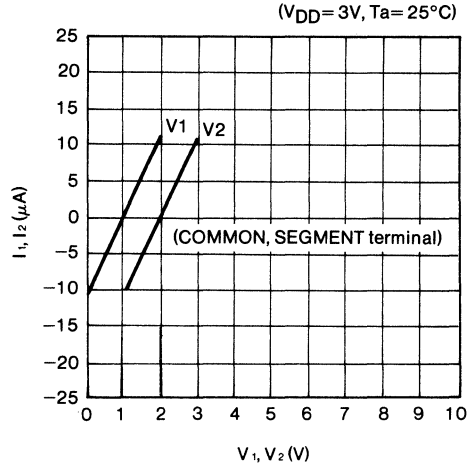
High-level Output Current (I_{OH}) – Output Voltage (V_{OH})



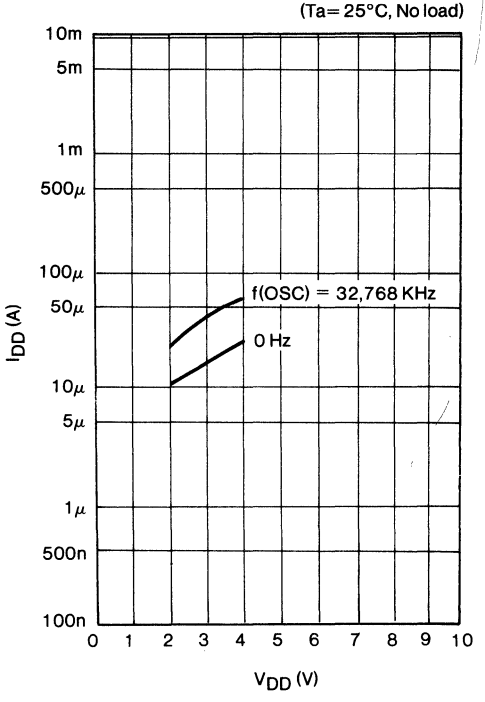
Low-level Output Current (I_{OL}) – Output Voltage (V_{OL})



Middle-level Output Current (I1, I2) – Output Voltage (V1, V2)

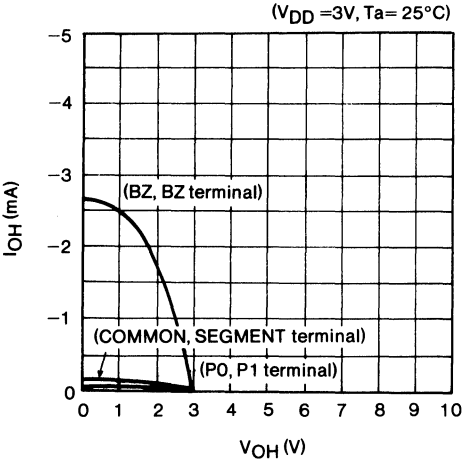


Current Consumption (I_{DD}) – Power Supply Voltage (V_{DD})

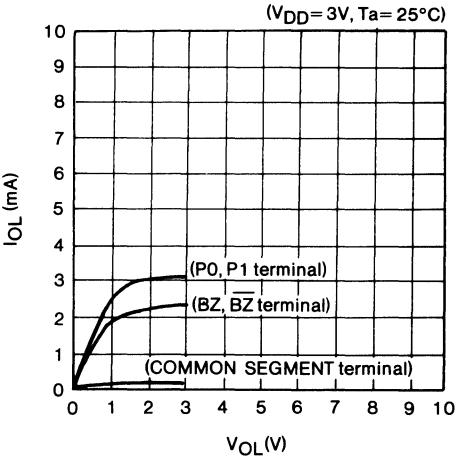


TYPICAL PERFORMANCE CURVES for MSM6512

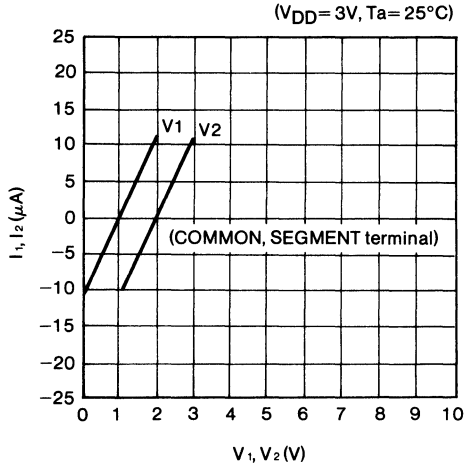
High-level Output Current (I_{OH}) – Output Voltage (V_{OH})



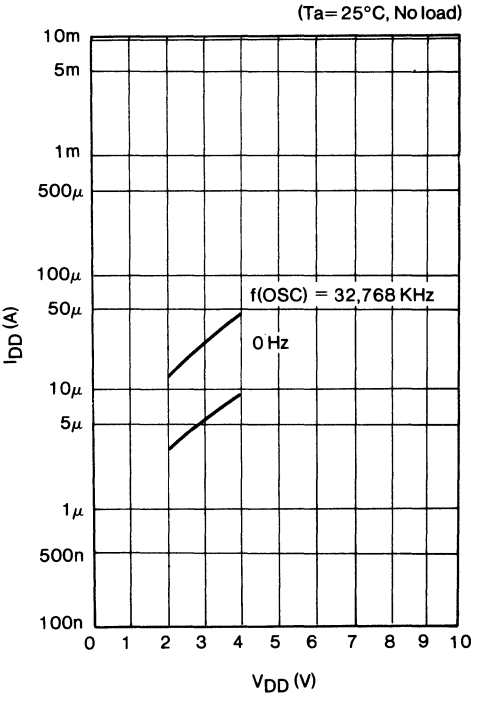
Low-level Output Current (I_{OL}) – Output Voltage (V_{OL})



Middle-level Output Current (I1, I2) – Output Voltage (V1, V2)



Current Consumption (I_{DD}) – Power Supply Voltage (V_{DD})



OLMS-62K SERIES (OKI ORIGINAL)

PRELIMINARY

OKI semiconductor

MSC62408B/62G408B

BI-CMOS SINGLE CHIP MICROCONTROLLER WITH FLT DRIVER

GENERAL DESCRIPTION

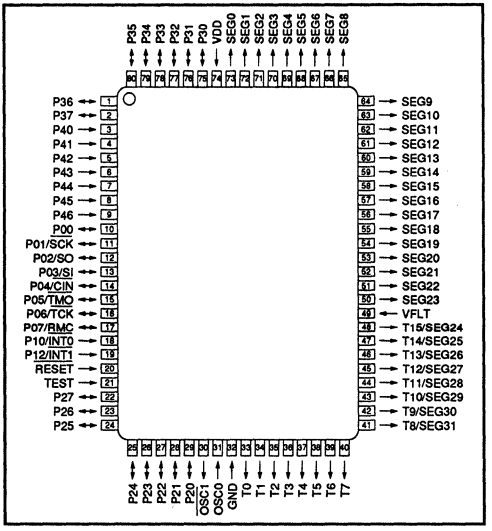
The MSC62408B is a high-speed, high-performance 8-bit single-chip microcontroller with built-in FLT controllers/drivers which is manufactured by the silicon gate Bi-CMOS process. Using the processing capacity of a large capacity ROM, RAM, FLT controllers/drivers, remote control receiving circuit, and 8-bit CPU, the MSC62408B may be widely used in audio apparatuses such as CD, and a MSC62G408B, in which the MSC62408B built-in program memory is replaced with an external ROM using a piggy-back package, can be used for program evaluation.

FEATURES

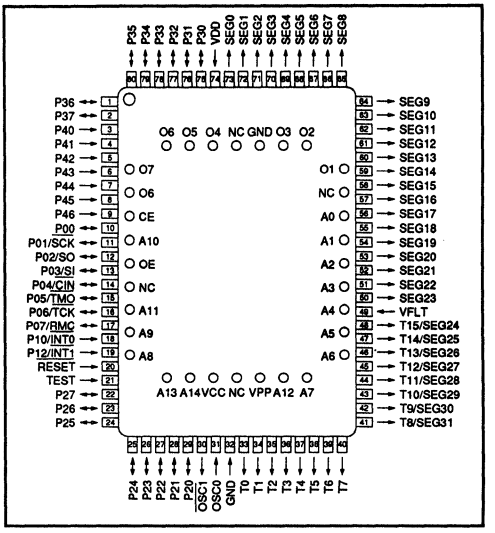
- ROM : 15360 x 8 bits
- RAM : 384 x 8 bits
  (display RAM included)
- Ports : 24 two-way I/O ports
  9 input ports
  (2 ports serving as interrupt inputs)
  8 output ports
  (serving as FLT segment outputs)
- FLT controllers/drivers (automatic display and dimmer functions) :
  24 segment outputs
  8 segment outputs
  8 timing outputs
- Remote control receiving circuit
- Counters :
  12-bit time base counter (free running)
  16-bit auto-reload timer event counter
  8-bit auto-reload timer event counter
- 8-bit shift register
- Interrupts : 7 lines (5 internal, 2 external)
- Cycle time : 928 ns, (4.3 MHz clock)
- Number of instructions : 251
- Many addressing modes
- Standby function
- Perfectly static operation
- Package :
  80pin plastic QFP (MSC62408B)
  (QFP80-P-1420-VIK)
  80pin ceramic piggyback (MSC62G408B)
  (QFP type)

PIN CONFIGURATION

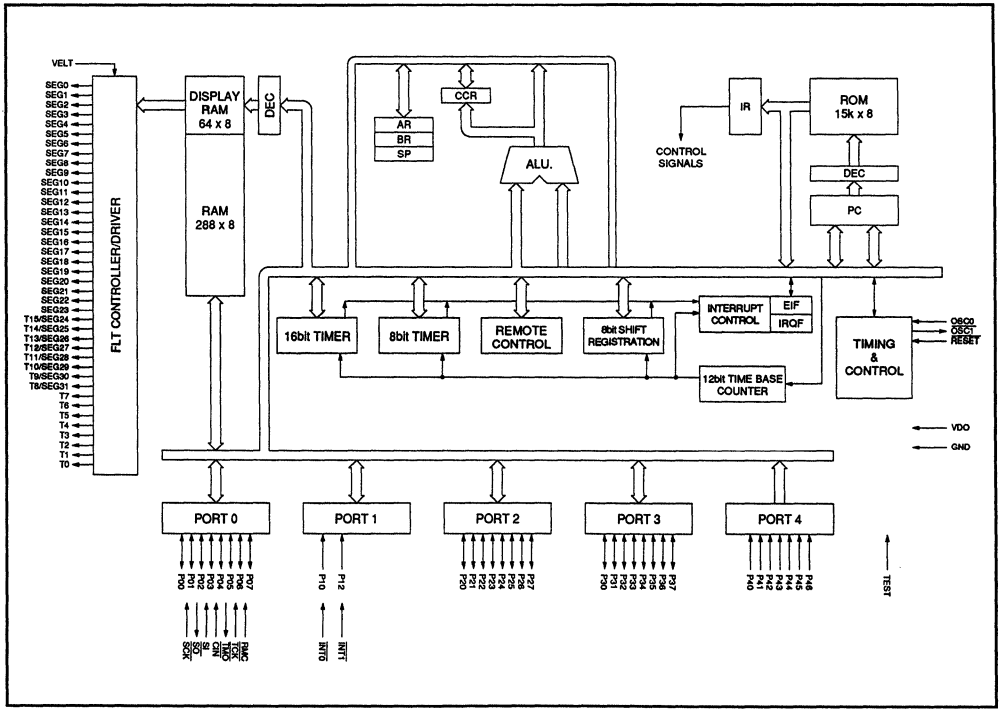
MSC62408B (TOP VIEW)
80 Lead Plastic QFP



MSC62G408B (TOP VIEW)
80 Lead Ceramic Piggy Back Package



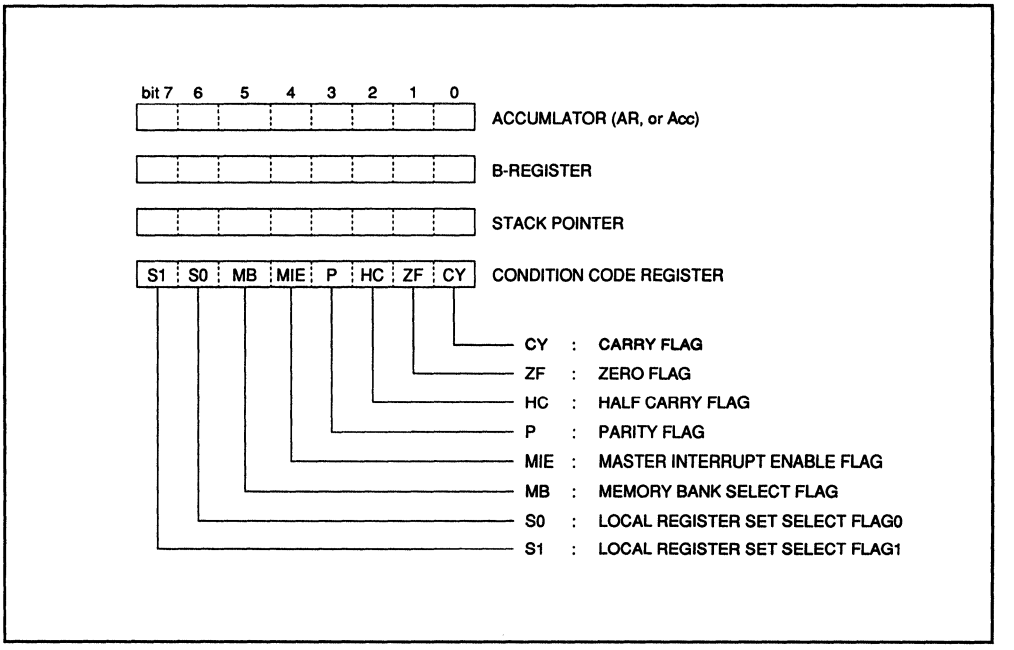
FUNCTIONAL BLOCK DIAGRAM



PIN DESCRIPTION

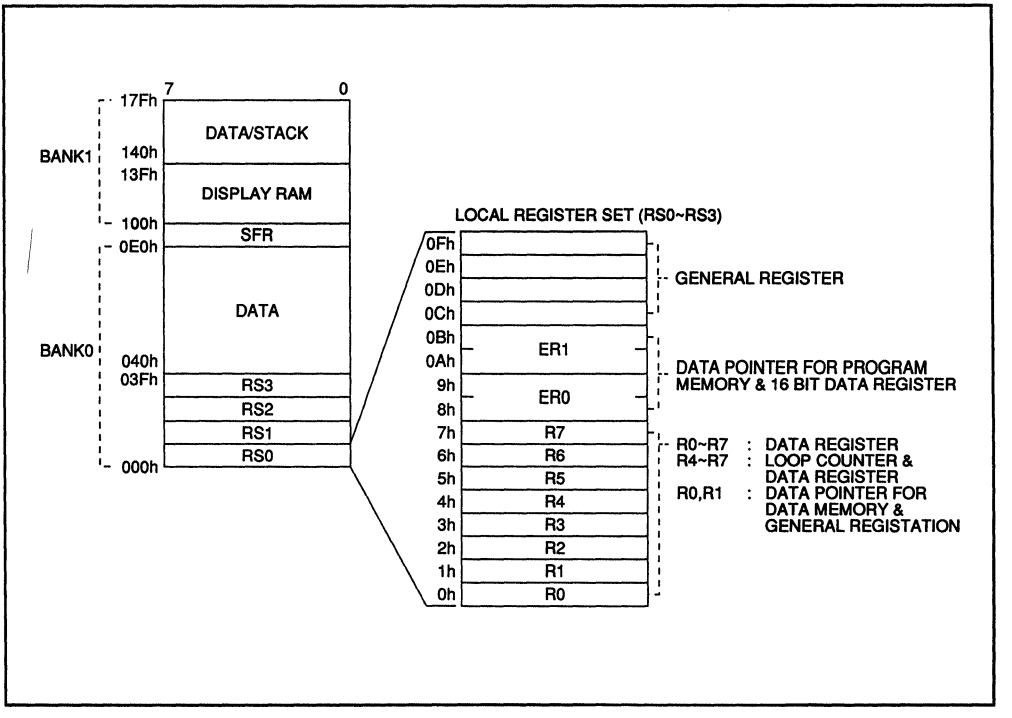
Terminal	Input/Output	Function
P00	Input/Output	I/O port
P01/SCK		I/O port (also used as serial clock input output)
P02/SO		I/O port (also used as serial data output)
P03/SI		I/O port (also serial data input)
P04/CIN		I/O port (also used as count input)
P05/TMO		I/O port (also used as timer output)
P06/TCK		I/O port (also used timer clock input)
P07/RMC		I/O port (also used remort control input)
P10/INT0	Input	Input port with pull up register
P12/INT1		(also used external interrupt input)
P20 ~ P27	Input/Output	I/O port
P30 ~ P37	Input/Output	I/O port
P40 ~ P43	Input	Input port with pull down register
P44 ~ P46		(Pull down register can be removed by instruction)
SEG0/P50 ~ SEG7/P57	Output	FLT segment driver/Port output
SEG8 ~ SEG23	Output	FLT segment driver
T8/SEG31 ~ T15/SEG24	Output	FLT segment driver/Timing output
T0 ~ T7	Output	Timing space output
OSC0	Input	Crystal ceramic connection terminal for system clock oscillation
OSC1	Output	Crystal ceramic connection terminal for system clock oscillation
RESET	Input	System reset input
VFLT	Input	Power supply for FLT driving
VDD	Input	System Power Supply (5V)
GND	Input	System Power Supply (0V)

REGISTER DIAGRAM



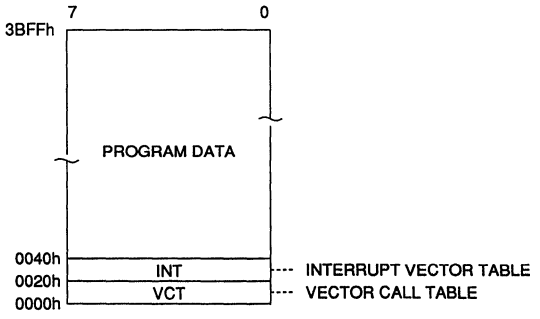
MEMORY MAP

DATA MEMORY



A.C.CHARACTERISTICS

(VDD = 5V± 10%, Ta = 0 to +70°C)



OKI semiconductor

MSM62720

CMOS 8-BIT SINGLE CHIP MICROCONTROLLER WITH 2K BYTES E²PROM

GENERAL DESCRIPTION

The MSM62720 is a CMOS single-chip microcontroller with on-board 2K BYTE E²PROM for applications such as IC-cards, etc.

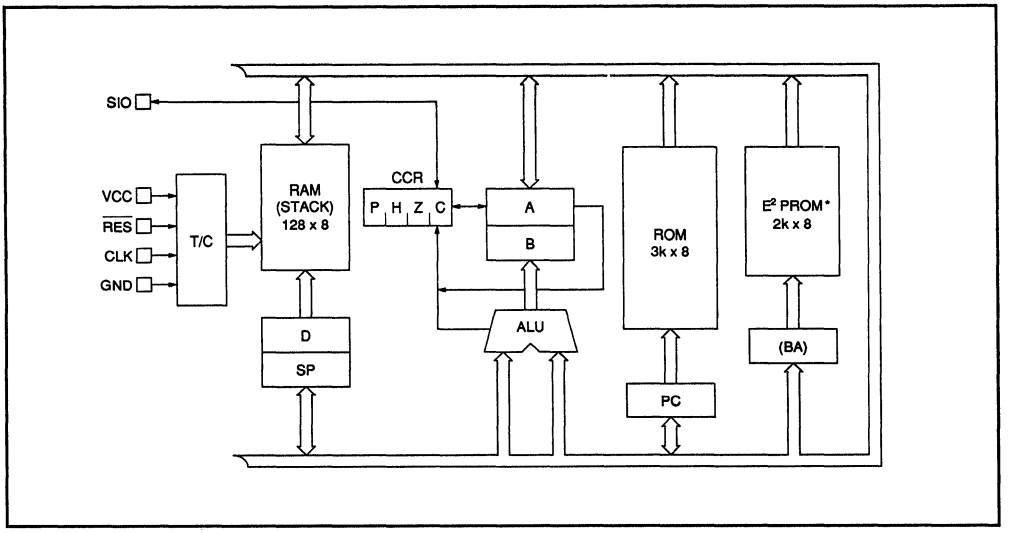
The powerful instruction set consists of 100 instructions including special instructions for IC cards, executed by the 8-bit CPU in 800ns at 5.0MHz clock frequency.

The MSM62720 has improved hardware and software for security. Consequently, this chip suits application such as IC cards of low cost, high security and high reliability.

FEATURES

- Single chip, low power CMOS
- 8-Bit Microcontroller
- 3K Bytes program ROM
- 1K Bytes control ROM
- 2K Bytes E²PROM
- 128 Byte data RAM
- Instruction cycle : 800ns @ 5MHz
- Number of instructions : 100
- Operation current : 4mA typ.
- Operating range : 0 to 70 °C
- Number of pads : 5
- Supply voltage : +5V ±10%
- Small Die size
- Simplified E²PROM write/erase operation by using control ROM
- Various security measures, such as memory data protection, countermeasures against CPU's runaway etc.
- Built-in error correction circuit (ECC)

FUNCTIONAL BLOCK DIAGRAM



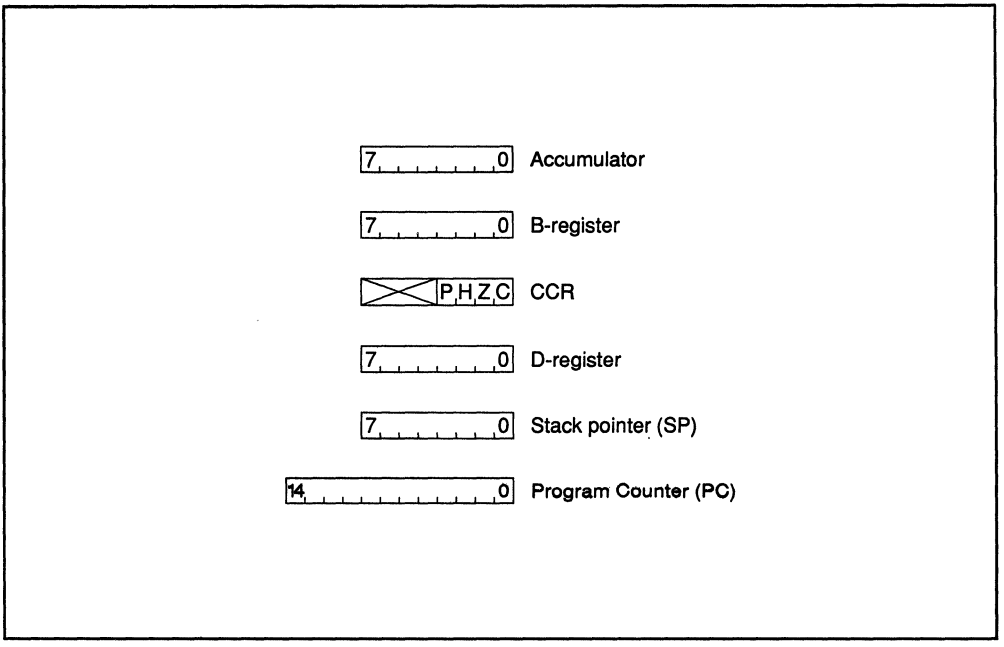
A	:	Accumulator	H	:	Half carry flag (for decimal operations)
ALU	:	Arithmetic circuit	P	:	Parity flag
B	:	B register (auxiliary register)	PC	:	Program counter
BA	:	B register paired with accumulator (B register higher rank)	RAM	:	Data memory
C	:	Carry flag	$\overline{RES}$	:	Reset input pin
CCR	:	Condition code register	SIO	:	Serial input/output pin
CLK	:	Clock input pin	SP	:	Stack pointer
D	:	D register (data pointer)	T/C	:	Timing and control circuit
EEPROM	:	Rewritable read-only memory	Vcc	:	Power supply pin (5V)
GND	:	Power supply pin (0V)	Z	:	Zero flag

* EEPROM is not used as instruction area.

PIN DESCRIPTION

Description	Input/Output	Function
SIO	Input/Output	Serial data input/output port. Quasi bidirectional I/O port. Set "1" level after "Reset". (Pulled up to Vcc by an internal (Approx. 10KΩ) Resister.
Vcc	–	Main power source
GND	–	Circuit GND potential
$\overline{RES}$	Input	Rest input pin is pulled up to Vcc by an internal (Approx. 1.5KΩ) resister.
CLK	Input	Clock input pin is pulled up to Vcc by an internal (Approx. 1.5KΩ) resister.

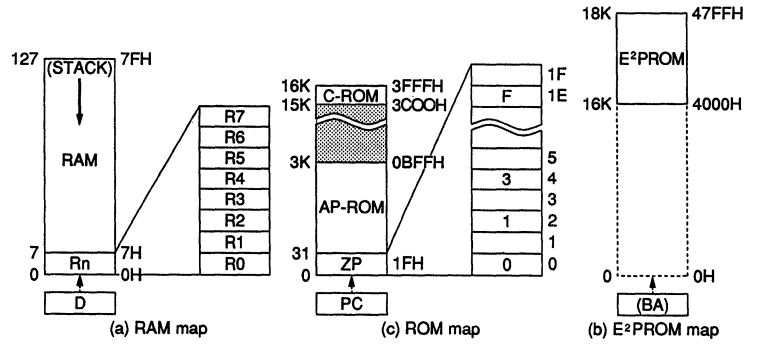
REGISTER DIAGRAM



MEMORY MAP

R7 ~ R4 : Registers which are concerned with DJNZ instruction
R7 ~ R0 : Local registers

: Empty area
C-ROM : Control ROM
ZP : Zero page adress called by CZP instruction



ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	Vcc	Ta = 25°C	-0.5 to 7	V
Input Voltage	VIN	Ta = 25°C	-0.5 to Vcc +0.5	V
Output Voltage	VOUT	Ta = 25°C	-0.5 to Vcc +0.5	V
Storage Temperature	Tstg	–	-55 to +150	°C

OPERATING CONDITIONS

Parameter	Symbol	Limits	Unit
Supply Voltage	Vcc	4.5 to 5.5	V
Operating Temperature	TOP	0 to +70	°C
E^2PROM Data Hold Temperature	TEEH	0 to +70	°C

D.C. CHARACTERISTICS

(Vcc = 5V ±10%, 0 to +70°C)

Parameter		Symbol	Conditions	Min	Typ	Max	Unit
Operating Current		Icc	f = 5MHz	–	6.0	10	mA
Low Input Voltage	CLK	VIL	–	-0.3	–	0.5	V
	$\overline{RES}$			-0.3	–	0.6	
	SIO			-0.3	–	0.8	
High Input Voltage	CLK	VIH	–	2.4	–	Vcc+0.3	V
	$\overline{RES}$			4	–	Vcc+0.3	
	SIO			2.0	–	Vcc+0.3	
Low Output Voltage		VOL	IOL MAX = 1.6mA	0	–	0.4	V
High Output Voltage		VOH	IOH MAX = -100µA	2.4	–	Vcc	V
Input Current (CLK, $\overline{RES}$)		IIH1	Vcc = VI = 5.5V	–	–	10	µA
		IIL1	Vcc=5.5V, VI=0.0V	–	–	-10	µA
Input Current (SIO)		IIH2	Vcc = VI = 5.5V	–	–	10	µA
		IIL2	Vcc=5.5V, VI=0.0V	–	–	-1	µA
Input Capacitance		CI	f = 1MHz Ta = 25°C	–	10	20	pF
Output Capacitance		CO	f = 1MHz Ta = 25°C	–	10	20	pF

Notes : = CLK, $\overline{RES}$ has poll down resistance, and SIO has pull up resistance.

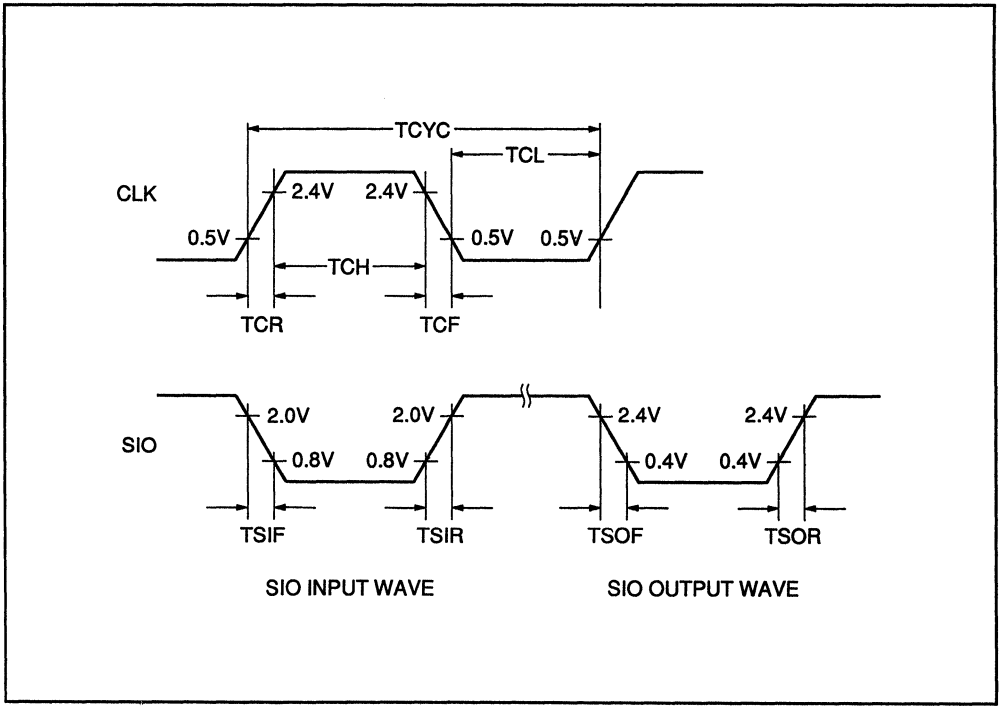
A.C. CHARACTERISTICS

(VDD = 5V ±10%, Ta = 0 to +70°C)

Parameter	Symbol	Min	Typ	Max	Unit
CLK Cycle Time	TCYC	200	–	–	ns
CLK Cycle Width	TCL	0.4*TCYC	–	0.6*TCYC	ns
CLK Cycle High Width	TCH	0.4*TCYC	–	0.6*TCYC	ns
CLK Cycle Rise Time	TCR	–	–	5.0	µs
CLK Cycle Fall Time	TCF	–	–	5.0	µs
$\overline{\text{RES}}$ Pulse Width	TRW	8*TCYC	–	–	µs
SIO INPUT Rise Time	TSIR	–	–	5.0	µs
SIO INPUT Fall Time	TSIF	–	–	5.0	µs
SIO OUTPUT Rise Time	TSOR	–	–	1.0	µs
SIO OUTPUT Fall Time	TSOF	–	–	1.0	µs

Note : = at output load capacitance Co = 30pF

TIMING CHARTS



MSM62780

CMOS 8-BIT SINGLE CHIP MICROCONTROLLER WITH 8K BYTES E²PROM

GENERAL DESCRIPTION

The MSM62780 is a CMOS single-chip microcontroller with on-board 8K BYTE E²PROM for applications such as IC-cards, etc.

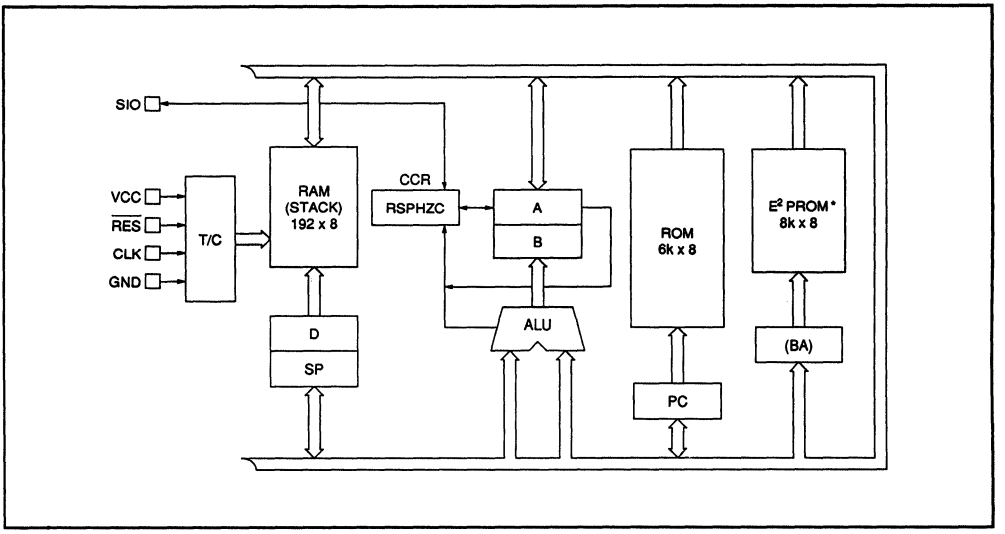
The powerful instruction set consists of 114 instructions including special instructions for IC cards, executed by the 8-bit CPU in 800ns at 5.0MHz clock frequency.

The MSM62780 has improved hardware and software for security. Consequently, this chip suits application such as IC cards of low cost, high security and high reliability.

FEATURES

- Single chip, low power CMOS
- 8-Bit Microcontroller
- 6K Bytes program ROM
- 2K Bytes control ROM
- 8K Bytes E²PROM
- 192 Byte data RAM
- Instruction cycle : 800ns @ 5MHz
- Number of instructions : 114
- Operation current : 8mA typ.
- Operating range : 0 to 70 °C
- Number of pads : 5
- Supply voltage : +5V ±10%
- Small Die size
- Simplified E²PROM write/erase operation by using control ROM
- Various security measures, such as memory data protection, countermeasures against CPU's runaway etc.
- Built-in 32 bytes E²PROM paga write mode
- Built-in error correction circuit (ECC)

FUNCTIONAL BLOCK DIAGRAM



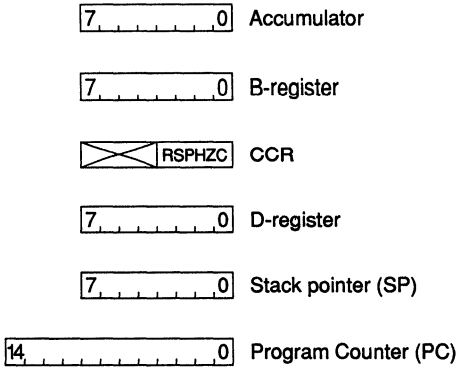
A	:	Accumulator	H	:	Half carry flag (for decimal operations)
ALU	:	Arithmetic circuit	P	:	Parity flag
B	:	B register (auxiliary register)	PC	:	Program counter
BA	:	B register paired with accumulator (B register higher rank)	RAM	:	Data memory
C	:	Carry flag	$\overline{RES}$	:	Reset input pin
CCR	:	Condition code register	SIO	:	Serial input/output pin
CLK	:	Clock input pin	SP	:	Stack pointer
D	:	D register (data pointer)	T/C	:	Timing and control circuit
EEPROM	:	Rewritable read-only memory	Vcc	:	Power supply pin (5V)
GND	:	Power supply pin (0V)	Z	:	Zero flag

* EEPROM is not used as instruction area.

PIN DESCRIPTION

Description	Input/Output	Function
SIO	Input/Output	Serial data input/output port. Quasi bidirectional I/O port. Set "1" level after "Reset". (Pulled up to Vcc by an internal (Approx. 10KΩ) Resister.
Vcc	–	Main power source
GND	–	Circuit GND potential
$\overline{RES}$	Input	Rest input pin is pulled up to Vcc by an internal (Approx. 1.5KΩ) resister.
CLK	Input	Clock input pin is pulled up to Vcc by an internal (Approx. 1.5KΩ) resister.

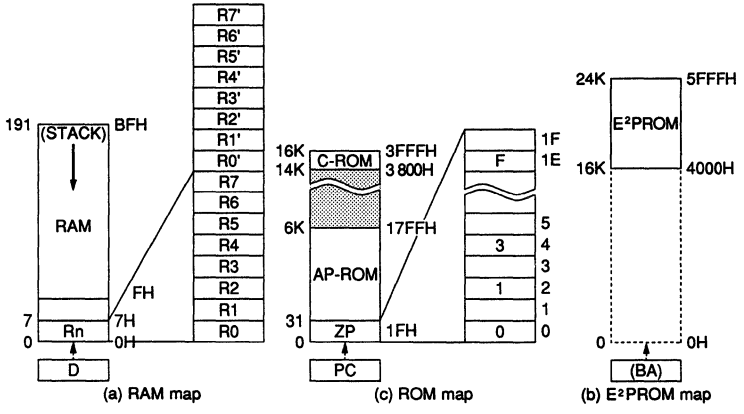
REGISTER DIAGRAM



MEMORY MAP

R7 ~ R4
R7' ~ R4' : Registers which are concerned with DJNZ instruction
R7 ~ R0
R7' ~ R0' : Local registers

: Empty area
C-ROM : Control ROM
ZP : Zero page adress called by CZP instruction



Note : Local register's address will be change by RS bit.

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	Vcc	Ta = 25°C	-0.5 to 7	V
Input Voltage	VIN	Ta = 25°C	-0.5 to Vcc +0.5	V
Output Voltage	VOUT	Ta = 25°C	-0.5 to Vcc +0.5	V
Storage Temperature	Tstg	–	-55 to +150	°C

OPERATING CONDITIONS

Parameter	Symbol	Limits	Unit
Supply Voltage	Vcc	4.5 to 5.5	V
Operating Temperature	TOP	0 to +70	°C
E^2PROM Data Hold Temperature	TEEH	0 to +70	°C

D.C. CHARACTERISTICS

(Vcc = 5V ±10%, 0 to +70°C)

Parameter		Symbol	Conditions	Min	Typ	Max	Unit
Operating Current		Icc	f = 5MHz	–	8.0	16	mA
Low Input Voltage	CLK	VIL	–	-0.3	–	0.5	V
	$\overline{RES}$			-0.3	–	0.6	
	SIO			-0.3	–	0.8	
High Input Voltage	CLK	VIH	–	2.4	–	Vcc+0.3	V
	$\overline{RES}$			4	–	Vcc+0.3	
	SIO			2.0	–	Vcc+0.3	
Low Output Voltage		VOL	IOL MAX = 1.6mA	0	–	0.4	V
High Output Voltage		VOH	IOH MAX = -100µA	2.4	–	Vcc	V
Input Current (CLK, $\overline{RES}$)		IIH1	Vcc = VI = 5.5V	–	–	10	µA
		IIL1	Vcc=5.5V, VI=0.0V	–	–	-10	µA
Input Current (SIO)		IIH2	Vcc = VI = 5.5V	–	–	10	µA
		IIL2	Vcc=5.5V, VI=0.0V	–	–	-1	µA
Input Capacitance		CI	f = 1MHz Ta = 25°C	–	10	20	pF
Output Capacitance		CO	f = 1MHz Ta = 25°C	–	10	20	pF

Notes : = CLK, $\overline{RES}$ has poll down resistance, and SIO has pull up resistance.

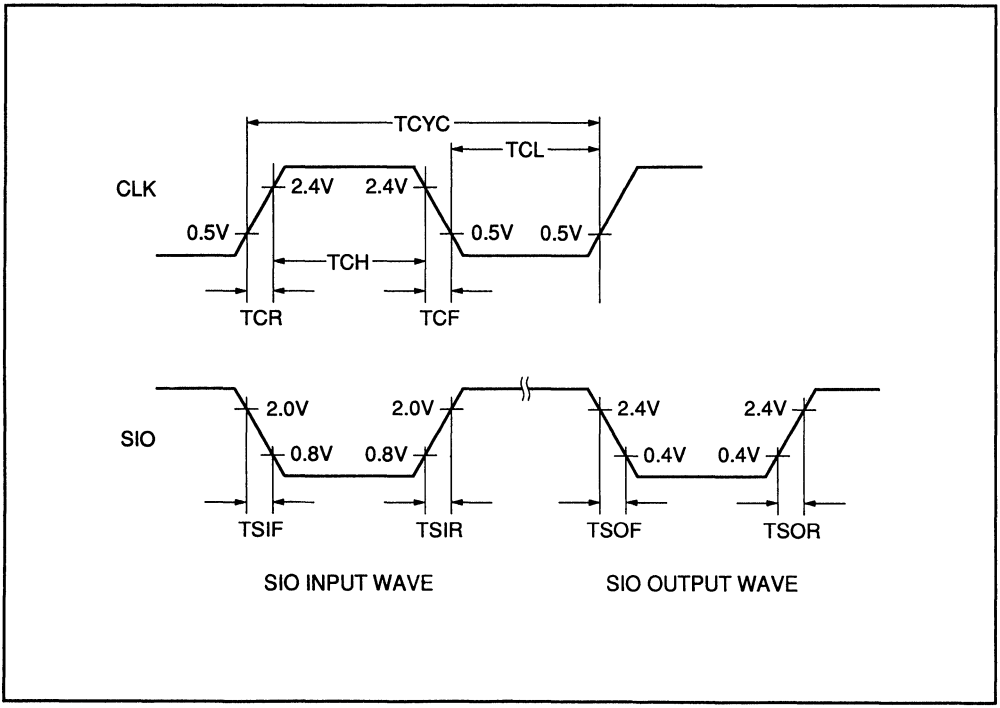
A.C. CHARACTERISTICS

(VDD = 5V ±10%, Ta = 0 to +70°C)

Parameter	Symbol	Min	Typ	Max	Unit
CLK Cycle Time	TCYC	200	–	–	ns
CLK Cycle Width	TCL	0.4*TCYC	–	0.6*TCYC	ns
CLK Cycle High Width	TCH	0.4*TCYC	–	0.6*TCYC	ns
CLK Cycle Rise Time	TCR	–	–	5.0	µs
CLK Cycle Fall Time	TCF	–	–	5.0	µs
$\overline{RES}$ Pulse Width	TRW	8*TCYC	–	–	µs
SIO INPUT Rise Time	TSIR	–	–	5.0	µs
SIO INPUT Fall Time	TSIF	–	–	5.0	µs
SIO OUTPUT Rise Time	TSOR	–	–	1.0	µs
SIO OUTPUT Fall Time	TSOF	–	–	1.0	µs

Note : = at output load capacitance Co = 30pF

TIMING CHARTS



OLMS-65K SERIES (OKI ORIGINAL)

Preliminary

OKI semiconductor
MSM65511

OKI ORIGINAL HIGH PERFORMANCE CMOS 8 BIT SINGLE CHIP MICROCONTROLLER

GENERAL DESCRIPTION

MSM65511 is a high-performance 8-bit single-chip controller that employs Oki's original nX-8/50 CPU core. With a minimum instruction execution time of 400 ns (10MHz clock), the MSM65511 is capable of high-speed processing, and includes 4 Kbytes of program memory, 128 bytes of data memory, timers and serial ports on chip.
Program and system evaluation can be performed using the MSM65P512, which as a version of the MSM65512 that replaces program memory with one-time PROM. The MSM65512 shares upward compatibility with the MSM65511.

OPERATING RANGE

- Operating Frequency : DC ~ 10 MHz
- Operating Voltage : 4.5 ~ 5.5 V
- Operating Temperature : -40 ~ 85°C

FEATURES

- Memory Space : 64 Kbytes

  On-Chip Program Memory : 4 Kbytes
  On-Chip Data Memory : 128 bytes

- Minimum Instruction Execution Cycle:
  400nS @ 10 MHz

- Powerful instruction set:

  81 basic instructions
  8/16-bit operation instructions
  Bit manipulation instructions
  Compound function instructions

- Abundant addressing modes

- I/O ports: 8-bit x 4

- Timers
  8-bit auto-reload timer x 2
  Watchdog timer x 1

- Counters
  Time base counter x 1

- Serial ports

  Shift register x 1

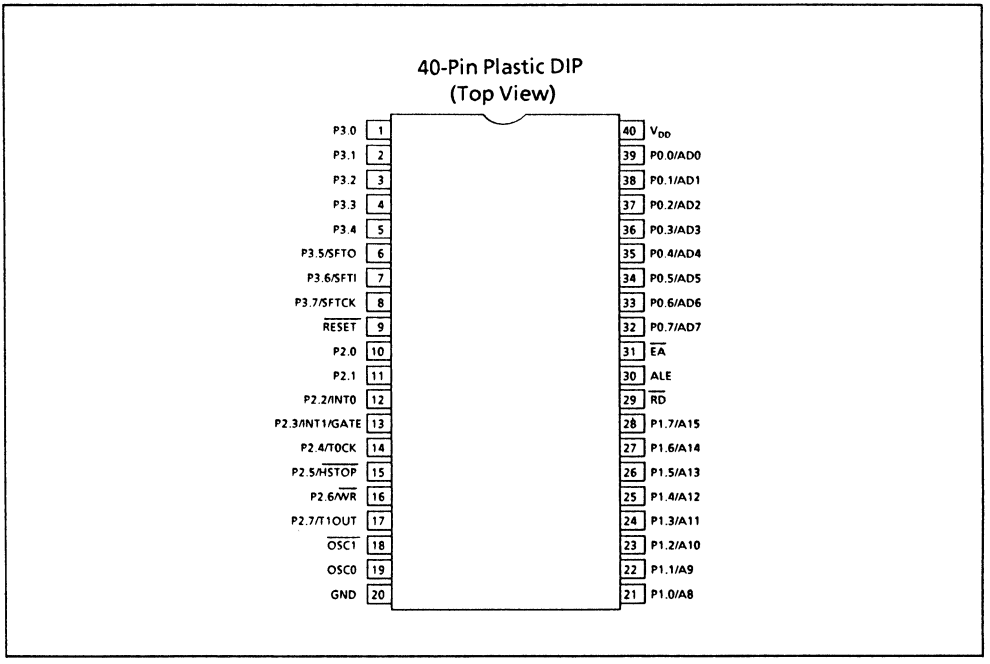
- External interrupts: 2

- Interrupt factors: 6

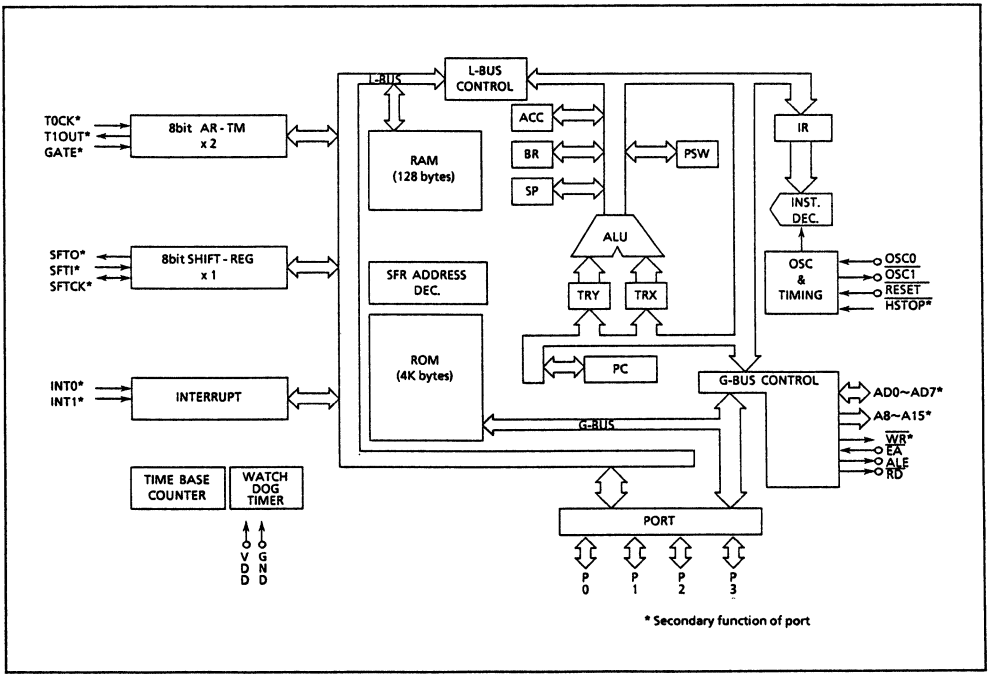
- Package:
  40 pin plastic DIP (DIP40-P-600)
  44 pin plastic QFP (T.B.D)
  44 pin PLCC (QFJ44-P-S650)

* Specifications are subject to change without notice.

PIN CONFIGURATION



FUNCTIONAL BLOCK DIAGRAM



PIN DESCRIPTIONS

Type	Pin Name	I/O	Description
Power supply	V_{DD}		+ 5 V power supply
	GND		0 V ground
Oscillation	OSC0	Input	System clock input pin. Quartz oscillator or ceramic oscillator is connected between OSC0 and $\overline{OSC1}$. For external clock, input at OSC0, leaving $\overline{OSC1}$ open.
	$\overline{OSC1}$	Output	System clock output pin
Control	$\overline{RESET}$	Input	System reset input (program starts from address 0040H); internal pull-up resistance
	$\overline{EA}$	Input	Program memory select input pin. "L" level input for external program memory; "H" level input for internal program memory.
	$\overline{RD}$	Output	Read strobe signal during external memory access
	ALE	Output	Address latch signal during external memory access
Port	PORT 0	I/O	8-bit I/O port During external memory access, becomes address/data bus for address output, instruction fetch or data read/write along with ALE, $\overline{RD}$ and $\overline{WR}$ pins
	PORT 1	I/O	8-bit I/O port Address bus during external memory access
	PORT 2 PORT 3	I/O	8-bit I/O port x2. Secondary functions shown in following table are added for ports 2 and 3.

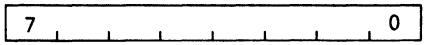
PIN SECONDARY FUNCTIONS

Pin Name	I/O	Description
INT0	Input	P2.2 secondary function. External interrupt 0 input pin.
INT1/GATE	Input	P2.3 secondary functions. External interrupt 1 input pin. Also used as input pin for gate signal for timer 0 count enable /disable.
T0CK	Input	P2.4 secondary function Timer 0 external clock input pin.
$\overline{HSTOP}$	Input	P2.5 secondary function. Hard stop mode input pin; stops system clock oscillation with "L" level input.
$\overline{WR}$	Output	P2.6 secondary function. Write strobe signal output pin during external data memory access.
T1OUT	Output	P2.7 secondary function. Output pin for signal that 2-divided timer 1 overflow.
SFTO	Output	P3.5 secondary function. Shift register data output pin.
SFTI	Input	P3.6 secondary function. Shift register data input signal.
SFTCK	I/O	P3.7 secondary function. Shift register synchronizing clock input/output signal.

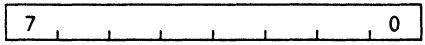
REGISTERS

• Operation Registers

A Register (Accumulator)



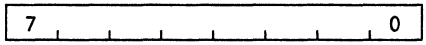
B Register (BR)



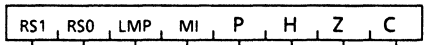
* For 16-bit operation instruction, A register holds low byte data and the B register holds high byte data.

• Special Purpose Registers

Stack pointer (SP)

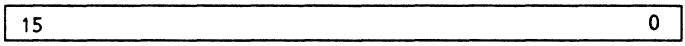


Program status word (PSW)

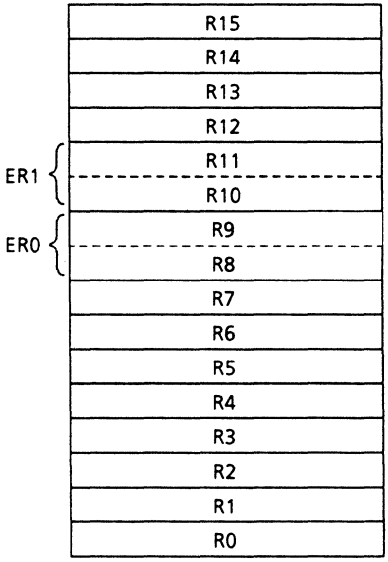


- Carry flag
- Zero flag
- Half-carry flag
- Parity flag
- Master interrupt enable flag
- Do not write "1" here
- Register set select flag
- Register set select flag

Program counter (PC)



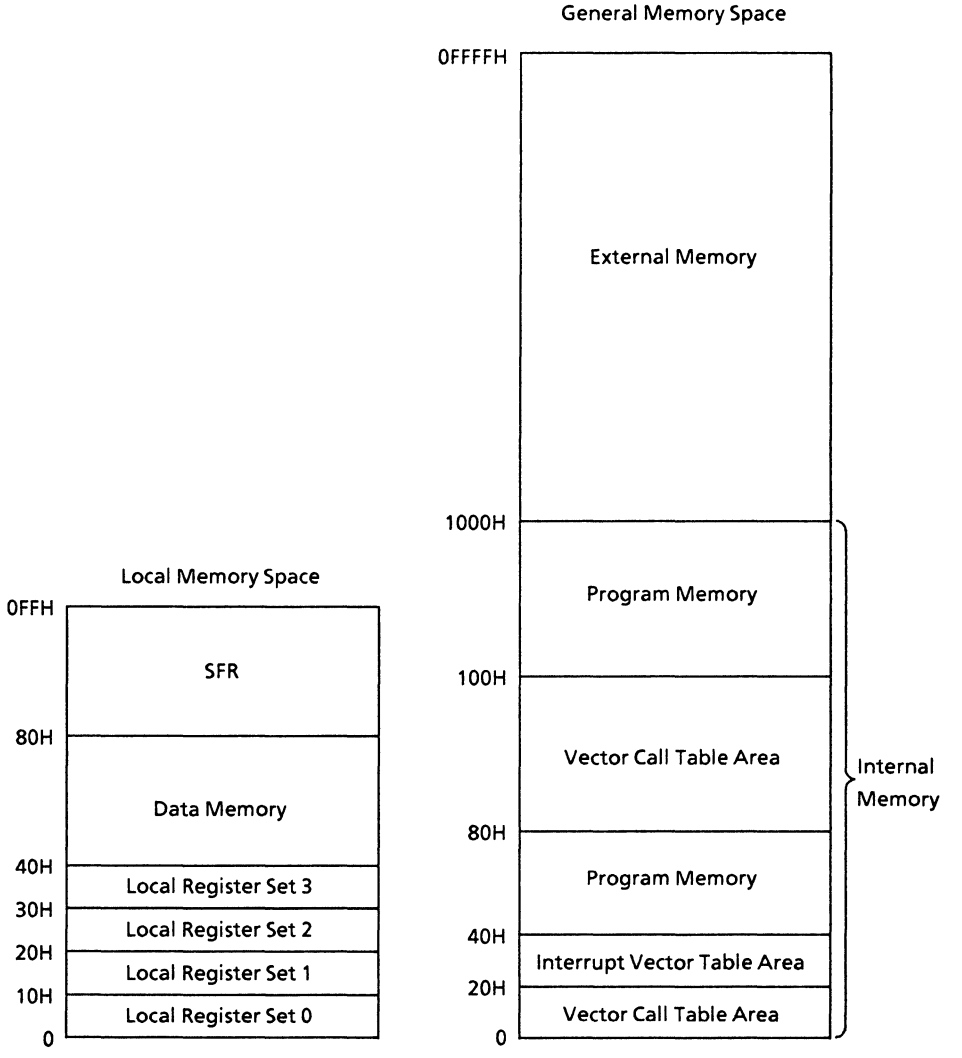
LOCAL REGISTERS



*1 4 banks of local registers are mapped in local memory space data memory.

*2 Registers R8 to R11 can be used as 16-bit registers, ER0 and ER1.

MEMORY MAPS


General Memory Space
0FFFFH
External Memory
1000H
Program Memory
100H
Vector Call Table Area
80H
Program Memory
40H
Interrupt Vector Table Area
20H
Vector Call Table Area
0
Internal Memory
Local Memory Space
0FFH
SFR
80H
Data Memory
40H
Local Register Set 3
30H
Local Register Set 2
20H
Local Register Set 1
10H
Local Register Set 0
0

SFR TABLE

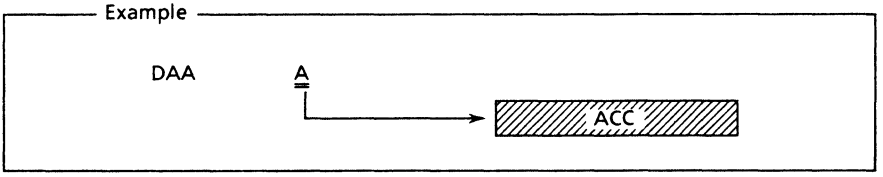
Address (HEX)	Name	Symbol	Symbol	R/W	Reset
0F6	Interrupt enable register		IEL	R/W	00H
0F4	Interrupt request register		IRQL		00H
0F3	External interrupt control register		XICON		0F0H
0F2	Standby control register		SBYCON		0F0H
0F0	Watchdog timer control register		WDTCON	W	—
0EE	Port 3 mode register		P3MOD	R/W	5FH
0ED	Port 3 direction register		P3DIR		00H
0EC	Port 3 data register		P3D		Undefined
0EA	Port 2 mode register		P2MOD		3FH
0E9	Port 2 direction register		P2DIR		00H
0E8	Port 2 data register		P2D		Undefined
0E5	Port 1 direction register		P1DIR		00H
0E4	Port 1 data register		P1D		Undefined
0E3	Port 0 direction register		P0DIR		00H
0E2	Port 0 data register		P0D		Undefined
0E1	Shift register		SFTR		Undefined
0E0	Shift register control register		SFTCON		0E0H
0DF	Timer 1 register	T01R	T1R		Undefined
0DE	Timer 0 register		T0R		Undefined
0DD	Timer 1 counter	T01C	T1C		Undefined
0DC	Timer 0 counter		T0C		Undefined
0DB	Timer 1 control register		T1CON		0F0H
0DA	Timer 0 control register		T0CON		0E0H
0D9	Time base counter control register		TBCON		0E0H

ADDRESSING MODES

MSM65511 has 256 bytes of local memory space and 8 Kbytes of general memory space. A variety of addressing modes are available for accessing these spaces.

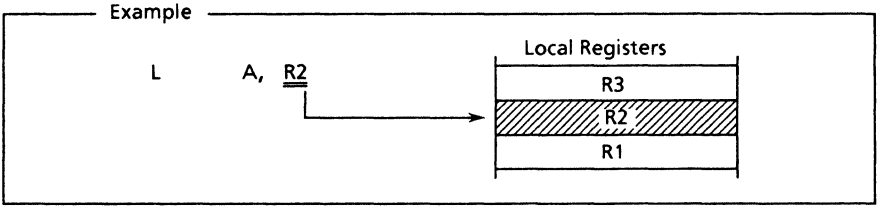
1. Register Direct Addressing

   - A、B、SP、PSW
   - BA



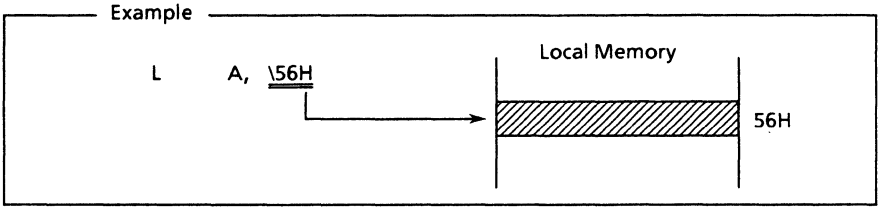
2. Local Register Direct Addressing

   - Rn (n = 0~15)
   - ERn (n = 0, 1)



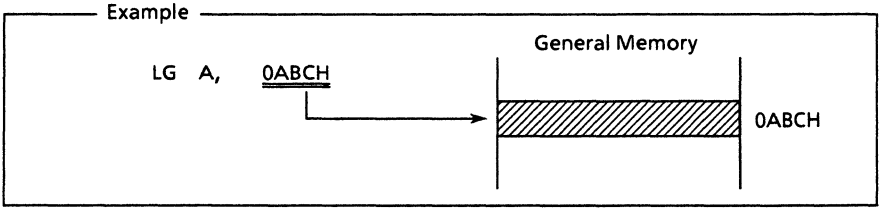
3. Local Memory Direct Addressing

   - \adrs



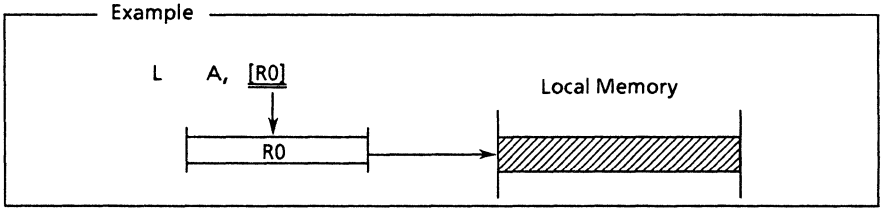
4. General Memory Direct Addressing

   - adrs



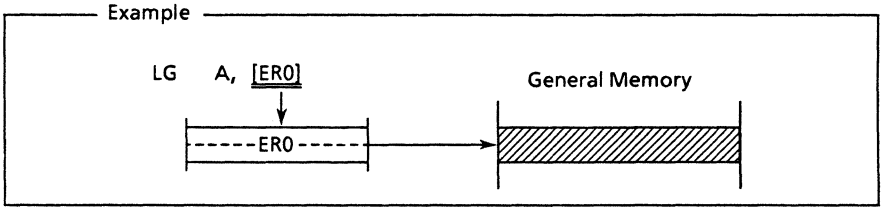
5. Local Memory - Register Indirect Addressing

- [Rn] (n = 0, 1, 8, 9)



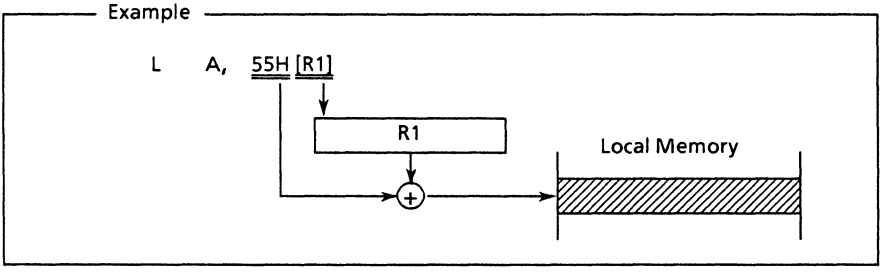
6. General Memory - Register Indirect Addressing

- [ERn] (n = 0, 1)
- [BA]



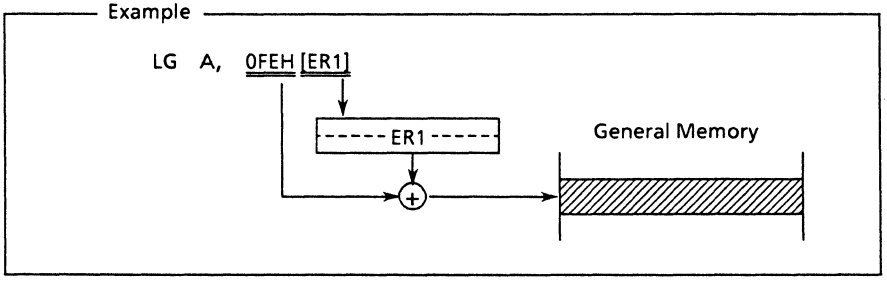
7. Local Memory Index Addressing

- disp [Rn] (n = 1, 9)



8. General Memory Index Addressing

- disp [ER1]



9. Immediate Addressing

- #n

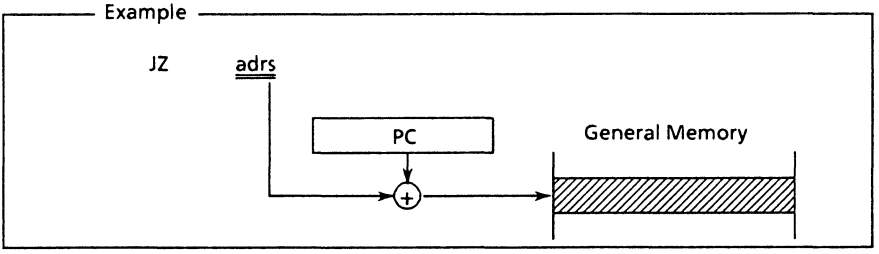
Example

```
L    A, #255
```

10. PC Relative Addressing

- adrs

Example



INSTRUCTION TABLES

• Data Transfer Instructions

Mnemonic		Function
L	obj1, obj2	Local memory load
LG	obj1, obj2	General memory load
ST	obj1, obj2	Store into local memory
STG	obj1, obj2	Store into general memory
MOV	PSW, #n	Immediate data transfer to PSW
MOV	obj1, obj2	Data transfer
MOVG	obj1, obj2	General memory data transfer
MOVW	obj1, obj2	16-bit data transfer
XCH	C, P	Carry and parity exchange
XCH	obj1, obj2	Data exchange
SWAP	obj	Upper nibble and lower nibble swap

• Increment and Decrement

Mnemonic		Function
INC	obj	Data increment
INCG	obj	General memory increment
INCW	obj	16-bit data increment
DEC	obj	Data decrement
DECG	obj	General memory decrement
DECW	obj	16-bit data decrement

• Arithmetic Operations

Mnemonic		Function
ADD	obj1, obj2	Data add
ADDW	obj1, obj2	16-bit data add
ADC	obj1, obj2	Data add with carry
ADCG	obj1, obj2	General memory data add with carry
SUB	obj1, obj2	Data subtract
SUBW	obj1, obj2	16-bit data subtract
SBC	obj1, obj2	Data subtract with carry
SBCG	obj1, obj2	General memory data subtract with carry

• Comparisons

Mnemonic		Function
CMP	obj1, obj2	Data compare
CMPW	obj1, obj2	16-bit data compare

• Logical Operations

Mnemonic		Function
AND	PSW, #n	PSW and immediate data logical AND
AND	obj1, obj2	Data logical AND
OR	PSW, #n	PSW and immediate data logical OR
OR	obj1, obj2	Data logical OR
XOR	obj1, obj2	Data exclusive OR

• Bit Operations

Mnemonic		Function
SB	obj. n	Bit set
SB	obj	PSW bit set
RB	obj. n	Bit reset
RB	obj	PSW bit reset
CPL	C	Carry complement
L	C, obj	Bit transfer to carry
ST	C, obj	Bit transfer from carry

• Rotate and Shift

Mnemonic		Function
ROL	obj	Rotate left
ROR	obj	Rotate right
SLL	obj	Shift left
SRL	obj	Shift right

• Decimal Adjust

Mnemonic		Function
DAA	obj	Decimal adjust after add
DAS	obj	Decimal adjust after subtract

• Conditional Jumps

Mnemonic		Function
JZ	adrs	Jump if zero flag is set
JNZ	adrs	Jump if zero flag is not set
JC	adrs	Jump if carry is set
JNC	adrs	Jump if carry is not set
DJZ	Rn, adrs	Decrement register, and jump if zero
DJNZ	Rn, adrs	Decrement register, and jump if not zero
JBS	obj. n, adrs	Jump if bit is set
JBR	obj. n, adrs	Jump, if bit is reset
JBSC	obj. n, adrs	Jump and clear bit if bit is set
CJE	C, P, adrs	Compare carry and parity; jump if equal
CJNE	C, P, adrs	Compare carry and parity; jump if not equal
CJE	obj1, obj2, adrs	Compare; jump if equal
CJNE	obj1, obj2, adrs	Compare; jump if not equal
CJEG	obj1, obj2, adrs	Compare with general memory data; jump if equal
CJNEG	obj1, obj2, adrs	Compare with general memory data; jump if not equal

• Jumps

Mnemonic		Function
J	adrs	Jump
SJ	adrs	Short jump
J	[BA]	Indirect jump

• Subroutines

Mnemonic		Function
PUSH	obj	Data push
POP	obj	Data pop
CAL	adrs	Subroutine call
CALZ	adrs	Call subroutine if zero flag is set
CALC	adrs	Call subroutine if carry flag is set
VCAL	n	Vector call
VCALZ	n	Vector call if zero flag is set
VCALC	n	Vector call if carry flag is set
RT		Return from subroutine
RTZ		Return from subroutine if zero flag is set
RTC		Return from subroutine if carry flag is set

• Other Instructions

Mnemonic		Function
CLR	obj	Clear
CLRW	BA	16-bit data clear
CPL	obj	Data complement
CPLW	BA	16-bit data complement
NOP		No operation
CHK	obj	Parity check
DLY	n	Program execution delay

Preliminary

OKI semiconductor
MSM65512/65P512

OKI ORIGINAL HIGH PERFORMANCE CMOS 8 BIT SINGLE CHIP MICROCONTROLLER

GENERAL DESCRIPTION

MSM65512 is a high-performance 8-bit single-chip controller that employs Oki's original nX-8/50 CPU core. With a minimum instruction execution time of 400 ns (10MHz clock), the MSM65512 is capable of high-speed processing, and includes 8 Kbytes of program memory, 256 bytes of data memory, timers and serial ports on chip. Also available is the MSM65P512, which replaces the on-chip program memory with one-time PROM.

OPERATING RANGE

- Operating Frequency : DC ~ 10 MHz
- Operating Voltage : 4.5 ~ 5.5 V
- Operating Temperature : -40 ~ 85°C (MSM65512)

FEATURES

- Memory Space : 64 Kbytes

  On-Chip Program Memory : 8 Kbytes
  On-Chip Data Memory : 256 bytes
- Minimum Instruction Execution Cycle:
  400nS @ 10 MHz
- Powerful instruction set:

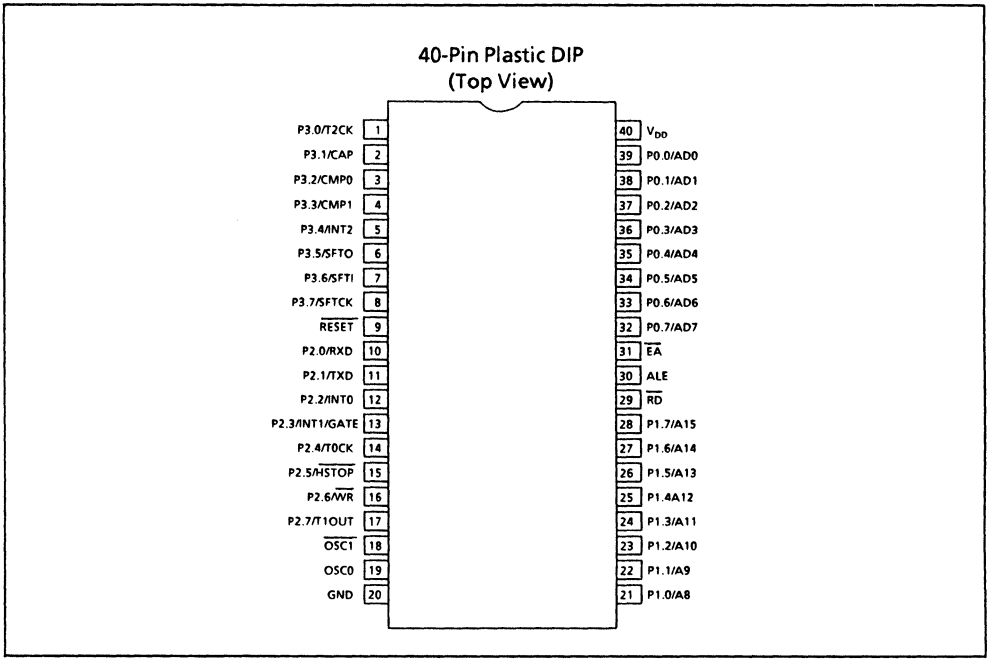
  83 basic instructions
  8/16-bit operation instructions
  Bit manipulation instructions
  Compound function instructions
- Abundant addressing modes
- Multiplication/division operation functions
  16 ← 8 x 8
  16 ← 16/8, 8 ← 16 mod 8
- I/O ports: 8-bit x 4
- Timers
  8-bit auto-reload timer x 2
  16-bit auto-reload timer x 1
  Watchdog timer x 1
- Counters
  Time base counter x 1
  16-bit free-running counter x 1
- Capture input: 1 channel
- Compare output: 2 channels
- Serial ports

  Shift register x 1

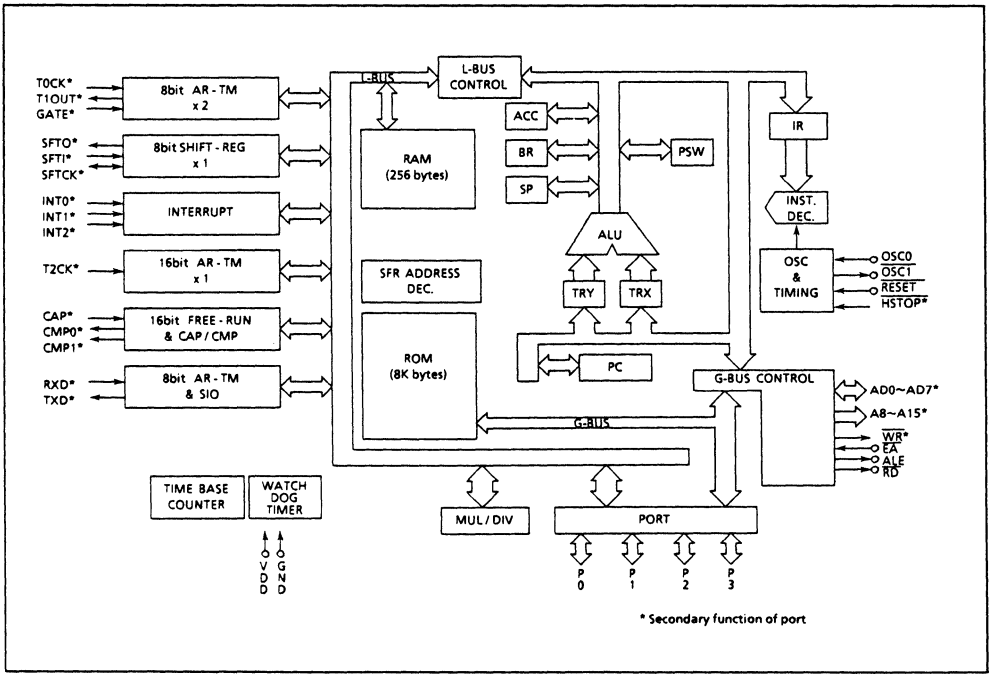
  Serial port with baud rate generator (UART/synchronous) x 1
- External interrupts: 3
- Interrupt factors: 15
- Package:
  40 pin plastic DIP (DIP40-P-600)
  44 pin plastic QFP (T.B.D)
  44 pin PLCC (QFJ44-P-S650)
  64 pin plastic QFP (T.B.D)

* Specifications are subject to change without notice.

PIN CONFIGURATION



FUNCTIONAL BLOCK DIAGRAM



PIN DESCRIPTIONS

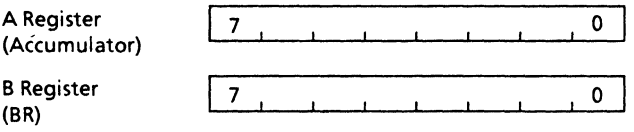
Type	Pin Name	I/O	Description
Power supply	VDD		+ 5 V power supply
	GND		0 V ground
Oscillation	OSC0	Input	System clock input pin. Quartz oscillator or ceramic oscillator is connected between OSC0 and $\overline{OSC1}$. For external clock, input at OSC0, leaving $\overline{OSC1}$ open.
	$\overline{OSC1}$	Output	System clock output pin
Control	$\overline{RESET}$	Input	System reset input (program starts from address 0040H); internal pull-up resistance
	$\overline{EA}$	Input	Program memory select input pin. "L" level input for external program memory; "H" level input for internal program memory.
	$\overline{RD}$	Output	Read strobe signal during external memory access
	ALE	Output	Address latch signal during external memory access
Port	PORT 0	I/O	8-bit I/O port During external memory access, becomes address/data bus for address output, instruction fetch or data read/write along with ALE, $\overline{RD}$ and $\overline{WR}$ pins
	PORT 1	I/O	8-bit I/O port Address bus during external memory access
	PORT 2 PORT 3	I/O	8-bit I/O port x 2. Secondary functions shown in following table are added for ports 2 and 3.

PIN SECONDARY FUNCTIONS

Pin Name	I/O	Description
RXD	I/O	P2.0 secondary functions. UART: Input pin for serial port receive data. Synchronous: Input pin for serial port transmit/receive data.
TXD	Output	P2.1 secondary functions. UART: Output pin for serial port transmit data. Synchronous: Output pin for serial port synchronizing clock.
INT0	Input	P2.2 secondary function. External interrupt 0 input pin.
INT1/GATE	Input	P2.3 secondary functions. External interrupt 1 input pin. Also used as input pin for gate signal for timer 0 count enable /disable.
T0CK	Input	P2.4 secondary function Timer 0 external clock input pin.
$\overline{\text{HSTOP}}$	Input	P2.5 secondary function. Hard stop mode input pin; stops system clock oscillation with "L" level input.
$\overline{\text{WR}}$	Output	P2.6 secondary function. Write strobe signal output pin during external data memory access.
T1OUT	Output	P2.7 secondary function. Output pin for signal that 2-divided timer 1 overflow.
T2CK	Input	P3.0 secondary function. Timer 2 external clock input pin.
CAP	Input	P3.1 secondary function. Capture trigger input pin.
CMP0	Output	P3.2 secondary function. Compare output channel 0 output pin.
CMP1	Output	P3.3 secondary function. Compare output channel 1 output pin.
INT2	Input	P3.4 secondary function. External interrupt 2 input signal.
SFTO	Output	P3.5 secondary function. Shift register data output pin.
SFTI	Input	P3.6 secondary function. Shift register data input signal.
SFTCK	I/O	P3.7 secondary function. Shift register synchronizing clock input/output signal.

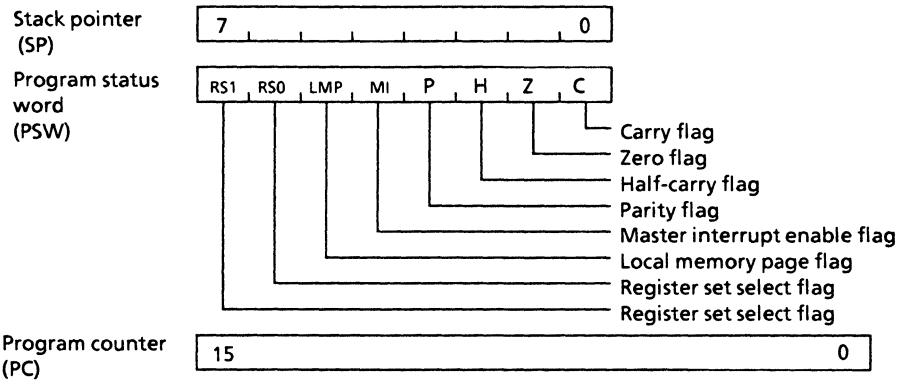
REGISTERS

● Operation Registers

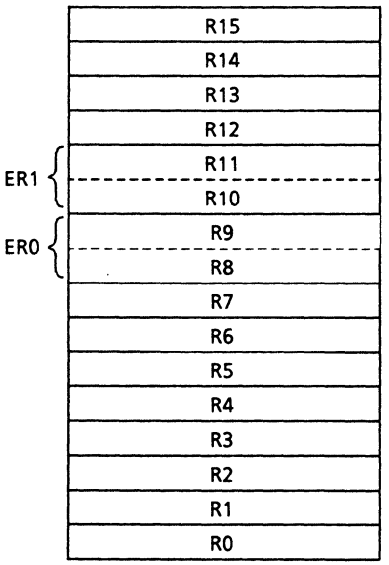


* For 16-bit operation instruction, A register holds low byte data and the B register holds high byte data.

● Special Purpose Registers



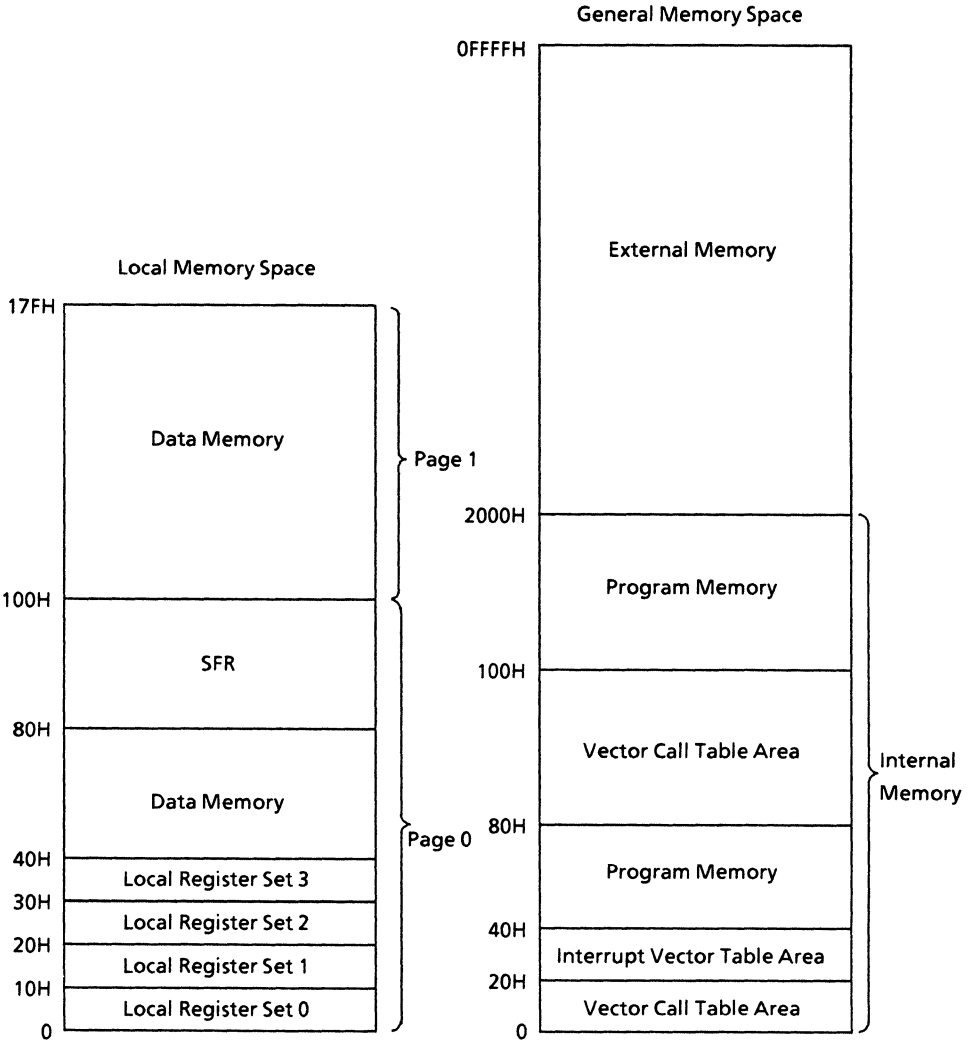
LOCAL REGISTERS



*1 4 banks of local registers are mapped in local memory space data memory.

*2 Registers R8 to R11 can be used as 16-bit registers, ER0 and ER1.

MEMORY MAPS


General Memory Space
0FFFFH
External Memory
2000H
Program Memory
100H
Vector Call Table Area
80H
Program Memory
40H
Interrupt Vector Table Area
20H
Vector Call Table Area
0
Internal Memory
Local Memory Space
17FH
Data Memory
Page 1
100H
SFR
80H
Data Memory
Page 0
40H
Local Register Set 3
30H
Local Register Set 2
20H
Local Register Set 1
10H
Local Register Set 0
0

SFR TABLE

Address (HEX)	Name	Sym-bol	Symbol	R/W	Reset
0FF	F register (multiplication/division operation register)	FER	FR	R/W	00H
0FE	E register (multiplication/division operation register)		ER		00H
0FD	D register (multiplication/division operation register)	DCR	DR		00H
0FC	C register (multiplication/division operation register)		CR		00H
0FB	Multiplication/division condition register		MDCR		0FCH
0F7	Interrupt enable register	IE	IEH	R/W	70H
0F6			IEL		00H
0F5	Interrupt request register	IRQ	IRQH		70H
0F4			IRQL		00H
0F3	External interrupt control register		XICON		0C0H
0F2	Standby control register		SBYCON		0F0H
0F1	Stack page specification register		SPR		0FEH
0F0	Watchdog timer control register		WDTCON	W	—
0EE	Port 3 mode register		P3MOD	R/W	53H
0ED	Port 3 direction register		P3DIR		00H
0EC	Port 3 data register		P3D		Unde-fined
0EA	Port 2 mode register		P2MOD		3CH
0E9	Port 2 direction register		P2DIR		00H
0E8	Port 2 data register		P2D		Unde-fined
0E5	Port 1 direction register		P1DIR		00H
0E4	Port 1 data register		P1D		Unde-fined
0E3	Port 0 direction register		P0DIR		00H
0E2	Port 0 data register		P0D		Unde-fined
0E1	Shift register		SFTR		Unde-fined
0E0	Shift register control register		SFTCON		0E0H

Address (HEX)	Name	Symbol	Symbol	R/W	Reset
0DF	Timer 1 register	T01R	T1R	R/W	Undefined
0DE	Timer 0 register		T0R		Undefined
0DD	Timer 1 counter	T01C	T1C		Undefined
0DC	Timer 0 counter		T0C		Undefined
0DB	Timer 1 control register		T1CON		0F0H
0DA	Timer 0 control register		T0CON		0E0H
0D9	Time base counter control register		TBCON		0E0H
0D7	Timer 2 register	T2R	T2RH		Undefined
0D6			T2RL		Undefined
0D5	Timer 2 counter	T2C	T2CH		Undefined
0D4			T2CL		Undefined
0D3	Timer 2 control register		T2CON		0F4H
0D2	Free-running counter	FRC	FRCH		00H
0D1			FRCL		00H
0D0	Free-running counter control register		FRCON		0F4H
0CF	Compare channel 1 data register	CMP1R	CMP1RH		Undefined
0CE			CMP1RL		Undefined
0CD	Compare channel 0 data register	CMP0R	CMP0RH		Undefined
0CC			CMP0RL		Undefined
0CB	Compare output control register		CMPCON		0F0H
0CA	Compare output data register		CMPOUT		0FCH
0C9	Capture data register	CAPR	CAPRH	R	Undefined
0C8			CAPRL		Undefined
0C7	Capture input control register		CAPCON	R/W	0FCH
0C6	Serial port buffer		SBUF		Undefined
0C5	Serial port control register	SCON	SCONH		0F0H
0C4			SCONL		00H
0C3	Timer 3 register		T3R		Undefined
0C2	Timer 3 counter		T3C		Undefined
0C1	Timer 3 control register		T3CON		0F4H

ADDRESSING MODES

MSM65512 has 384 bytes of local memory space and 64 Kbytes of general memory space. A variety of addressing modes are available for accessing these spaces.

1. Register Direct Addressing

   - A、B、SP、PSW
   - BA

Example

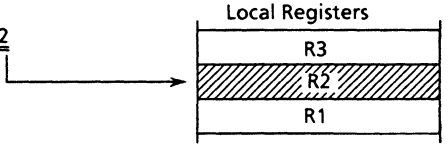
```
DAA    A
```

2. Local Register Direct Addressing

   - Rn (n = 0~15)
   - ERn (n = 0, 1)

Example

```
L      A, R2
```

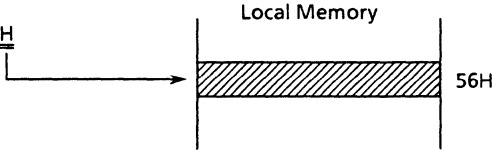


3. Local Memory Direct Addressing

   - \adrs (within 256-byte page; page specified by LMP flag)
   - adrs (384 bytes)

Example

```
L      A, \56H
```

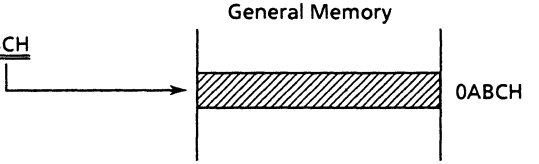


4. General Memory Direct Addressing

   - adrs

Example

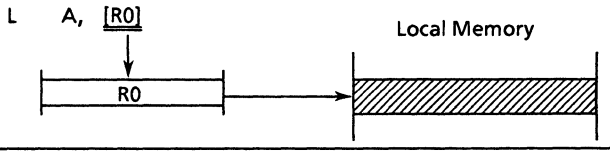
```
LG  A,  0ABCH
```



5. Local Memory - Register Indirect Addressing

- [Rn] (n = 0, 1, 8, 9; within 256-byte page; page specified by LMP flag)
- page: [Rn] (page = 0, 1 n = 0, 1, 8, 9 384 bytes)

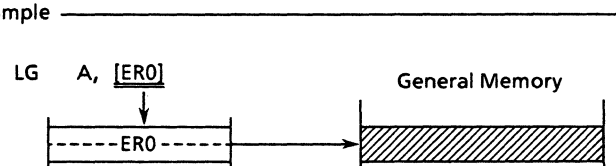
Example



6. General Memory - Register Indirect Addressing

- [ERn] (n = 0, 1)
- [BA]

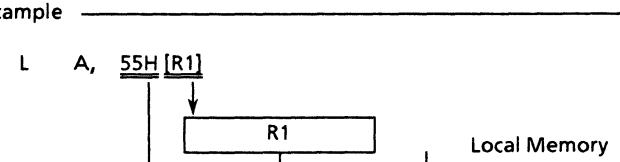
Example



7. Local Memory Index Addressing

- disp [Rn] (n = 1, 9; within 256-byte page; page specified by LMP flag)
- page: disp [Rn] (page = 0, 1 n = 1, 9 384 bytes)

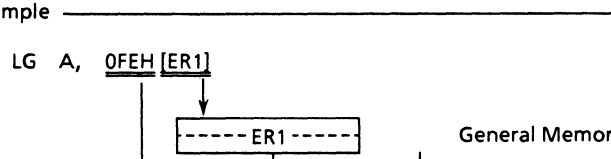
Example



8. General Memory Index Addressing

- disp [ER1]

Example



9. Immediate Addressing

- #n

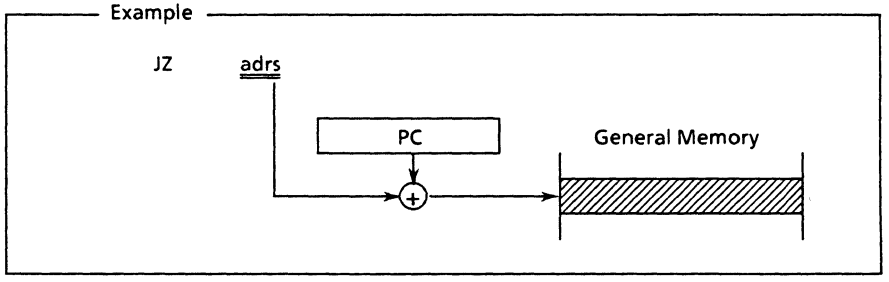
Example

```
L    A, #255
```

10. PC Relative Addressing

- adrs

Example



INSTRUCTION TABLES

● Data Transfer Instructions

Mnemonic		Function
L	obj1, obj2	Local memory load
LG	obj1, obj2	General memory load
ST	obj1, obj2	Store into local memory
STG	obj1, obj2	Store into general memory
MOV	PSW, #n	Immediate data transfer to PSW
MOV	obj1, obj2	Data transfer
MOVG	obj1, obj2	General memory data transfer
MOVW	obj1, obj2	16-bit data transfer
XCH	C, P	Carry and parity exchange
XCH	obj1, obj2	Data exchange
SWAP	obj	Upper nibble and lower nibble swap

● Increment and Decrement

Mnemonic		Function
INC	obj	Data increment
INCG	obj	General memory increment
INCW	obj	16-bit data increment
DEC	obj	Data decrement
DECG	obj	General memory decrement
DECW	obj	16-bit data decrement

• Arithmetic Operations

Mnemonic		Function
ADD	obj1, obj2	Data add
ADDW	obj1, obj2	16-bit data add
ADC	obj1, obj2	Data add with carry
ADCG	obj1, obj2	General memory data add with carry
SUB	obj1, obj2	Data subtract
SUBW	obj1, obj2	16-bit data subtract
SBC	obj1, obj2	Data subtract with carry
SBCG	obj1, obj2	General memory data subtract with carry
MUL		Multiplication 16 ← 8 x 8
DIV		Division 16 ← 16/8, 8 ← 16 mod 8

• Comparisons

Mnemonic		Function
CMP	obj1, obj2	Data compare
CMPW	obj1, obj2	16-bit data compare

• Logical Operations

Mnemonic		Function
AND	PSW, #n	PSW and immediate data logical AND
AND	obj1, obj2	Data logical AND
OR	PSW, #n	PSW and immediate data logical OR
OR	obj1, obj2	Data logical OR
XOR	obj1, obj2	Data exclusive OR

• Bit Operations

Mnemonic		Function
SB	obj. n	Bit set
SB	obj	PSW bit set
RB	obj. n	Bit reset
RB	obj	PSW bit reset
CPL	C	Carry complement
L	C, obj	Bit transfer to carry
ST	C, obj	Bit transfer from carry

● Rotate and Shift

Mnemonic		Function
ROL	obj	Rotate left
ROR	obj	Rotate right
SLL	obj	Shift left
SRL	obj	Shift right

● Decimal Adjust

Mnemonic		Function
DAA	obj	Decimal adjust after add
DAS	obj	Decimal adjust after subtract

● Conditional Jumps

Mnemonic		Function
JZ	adrs	Jump if zero flag is set
JNZ	adrs	Jump if zero flag is not set
JC	adrs	Jump if carry is set
JNC	adrs	Jump if carry is not set
DJZ	Rn, adrs	Decrement register, and jump if zero
DJNZ	Rn, adrs	Decrement register, and jump if not zero
JBS	obj. n, adrs	Jump if bit is set
JBR	obj. n, adrs	Jump, if bit is reset
JBSC	obj. n, adrs	Jump and clear bit if bit is set
CJE	C, P, adrs	Compare carry and parity; jump if equal
CJNE	C, P, adrs	Compare carry and parity; jump if not equal
CJE	obj1, obj2, adrs	Compare; jump if equal
CJNE	obj1, obj2, adrs	Compare; jump if not equal
CJEG	obj1, obj2, adrs	Compare with general memory data; jump if equal
CJNEG	obj1, obj2, adrs	Compare with general memory data; jump if not equal

● Jumps

Mnemonic		Function
J	adrs	Jump
SJ	adrs	Short jump
J	[BA]	Indirect jump

• Subroutines

Mnemonic		Function
PUSH	obj	Data push
POP	obj	Data pop
CAL	adrs	Subroutine call
CALZ	adrs	Call subroutine if zero flag is set
CALC	adrs	Call subroutine if carry flag is set
VCAL	n	Vector call
VCALZ	n	Vector call if zero flag is set
VCALC	n	Vector call if carry flag is set
RT		Return from subroutine
RTZ		Return from subroutine if zero flag is set
RTC		Return from subroutine if carry flag is set

• Other Instructions

Mnemonic		Function
CLR	obj	Clear
CLRW	BA	16-bit data clear
CPL	obj	Data complement
CPLW	BA	16-bit data complement
NOP		No operation
CHK	obj	Parity check
DLY	n	Program execution delay

Preliminary

OKI semiconductor
MSM65513/65P513

OKI ORIGINAL HIGH PERFORMANCE CMOS 8 BIT SINGLE CHIP MICROCONTROLLER

GENERAL DESCRIPTION

MSM65513 is a high-performance 8-bit single-chip controller that employs Oki's original nX-8/50 CPU core. With a minimum instruction execution time of 400 ns (10MHz clock), the MSM65513 is capable of high-speed processing, and includes 8 Kbytes of program memory, 256 bytes of data memory, timers and serial ports on chip. Also available is the MSM65P513, which replaces the on-chip program memory with one-time PROM.

OPERATING RANGE

- Operating Frequency : DC ~ 10 MHz
- Operating Voltage : 4.5 ~ 5.5 V
- Operating Temperature : -40 ~ 85°C (MSM65513)

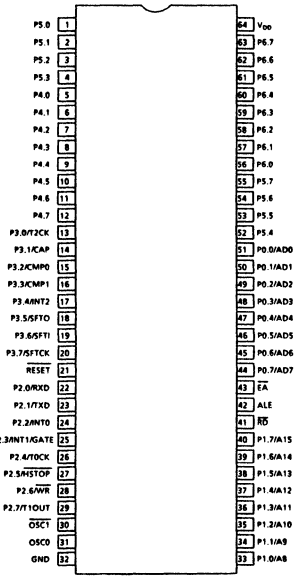
FEATURES

- Memory Space : 64 Kbytes
  On-Chip Program Memory : 8 Kbytes
  On-Chip Data Memory : 256 bytes
- Minimum Instruction Execution Cycle:
  400nS @ 10 MHz
- Powerful instruction set:
  83 basic instructions
  8/16-bit operation instructions
  Bit manipulation instructions
  Compound function instructions
- Abundant addressing modes
- Multiplication/division operation functions
  16 ← 8 x 8
  16 ← 16/8, 8 ← 16 mod 8
- I/O ports: 8-bit x 6
  Input only port: 8-bit x 1
- Timers
  8-bit auto-reload timer x 2
  16-bit auto-reload timer x 1
  Watchdog timer x 1
- Counters
  Time base counter x 1
  16-bit free-running counter x 1
- Capture input: 1 channel
- Compare output: 2 channels
- Serial ports
  Shift register x 1
  Serial port with baud rate generator (UART/synchronous) x 1
- External interrupts: 3
- Interrupt factors: 15
- Package:
  64 pin plastic shrink DIP (SDIP64-P-750)
  64 pin plastic QFP (T.B.D)
  68 pin PLCC (QFJ68-P-S950)

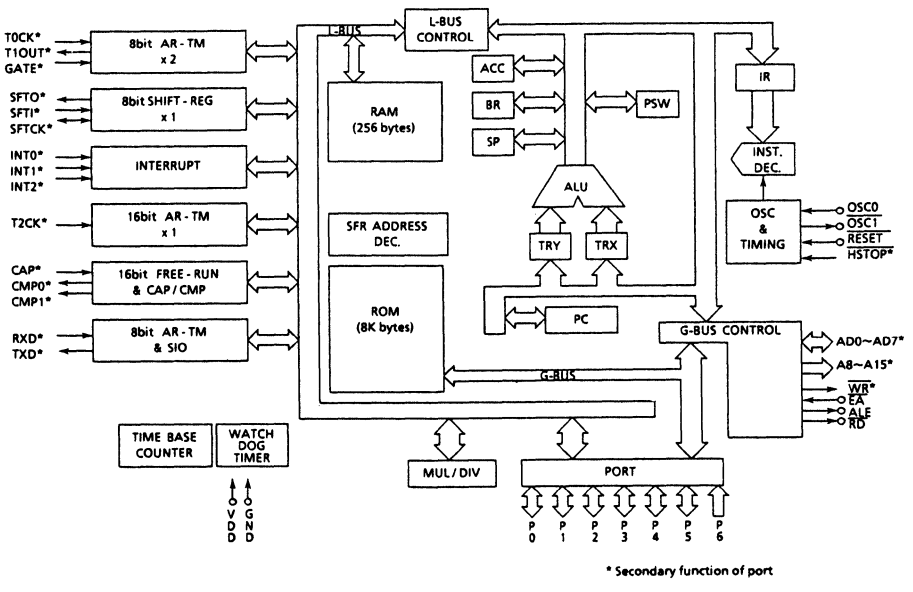
* Specifications are subject to change without notice.

PIN CONFIGURATION

64-Pin Plastic Shrink DIP
(Top View)



FUNCTIONAL BLOCK DIAGRAM



PIN DESCRIPTIONS

Type	Pin Name	I/O	Description
Power supply	VDD		+ 5 V power supply
	GND		0 V ground
Oscillation	OSC0	Input	System clock input pin. Quartz oscillator or ceramic oscillator is connected between OSC0 and $\overline{OSC1}$. For external clock, input at OSC0, leaving $\overline{OSC1}$ open.
	$\overline{OSC1}$	Output	System clock output pin
Control	$\overline{RESET}$	Input	System reset input (program starts from address 0040H); internal pull-up resistance
	$\overline{EA}$	Input	Program memory select input pin. "L" level input for external program memory; "H" level input for internal program memory.
	$\overline{RD}$	Output	Read strobe signal during external memory access
	ALE	Output	Address latch signal during external memory access
Port	PORT 0	I/O	8-bit I/O port During external memory access, becomes address/data bus for address output, instruction fetch or data read/write along with ALE, $\overline{RD}$ and $\overline{WR}$ pins
	PORT 1	I/O	8-bit I/O port Address bus during external memory access
	PORT 2 PORT 3 PORT 4 PORT 5	I/O	8-bit I/O port x 4. Secondary functions shown in following table are added for ports 2 and 3.
	PORT 6	Input	8-bit I/O port.

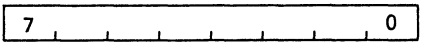
PIN SECONDARY FUNCTIONS

Pin Name	I/O	Description
RXD	I/O	P2.0 secondary functions. UART: Input pin for serial port receive data. Synchronous: Input pin for serial port transmit/receive data.
TXD	Output	P2.1 secondary functions. UART: Output pin for serial port transmit data. Synchronous: Output pin for serial port synchronizing clock.
INT0	Input	P2.2 secondary function. External interrupt 0 input pin.
INT1/GATE	Input	P2.3 secondary functions. External interrupt 1 input pin. Also used as input pin for gate signal for timer 0 count enable /disable.
T0CK	Input	P2.4 secondary function Timer 0 external clock input pin.
$\overline{\text{HSTOP}}$	Input	P2.5 secondary function. Hard stop mode input pin; stops system clock oscillation with "L" level input.
$\overline{\text{WR}}$	Output	P2.6 secondary function. Write strobe signal output pin during external data memory access.
T1OUT	Output	P2.7 secondary function. Output pin for signal that 2-divided timer 1 overflow.
T2CK	Input	P3.0 secondary function. Timer 2 external clock input pin.
CAP	Input	P3.1 secondary function. Capture trigger input pin.
CMP0	Output	P3.2 secondary function. Compare output channel 0 output pin.
CMP1	Output	P3.3 secondary function. Compare output channel 1 output pin.
INT2	Input	P3.4 secondary function. External interrupt 2 input signal.
SFTO	Output	P3.5 secondary function. Shift register data output pin.
SFTI	Input	P3.6 secondary function. Shift register data input signal.
SFTCK	I/O	P3.7 secondary function. Shift register synchronizing clock input/output signal.

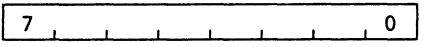
REGISTERS

● Operation Register

A Register (Accumulator)



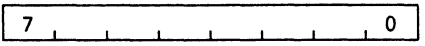
B Register (BR)



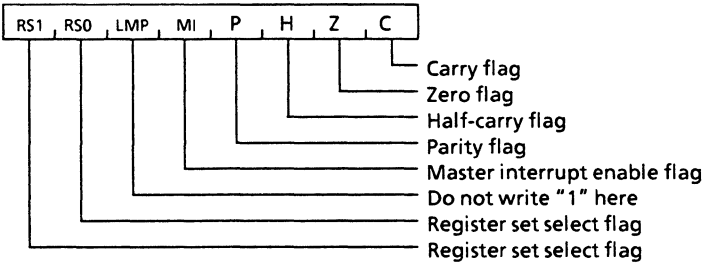
* For 16-bit operation instruction, A register holds low byte data and the B register holds high byte data.

● Special Purpose Registers

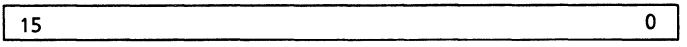
Stack pointer (SP)



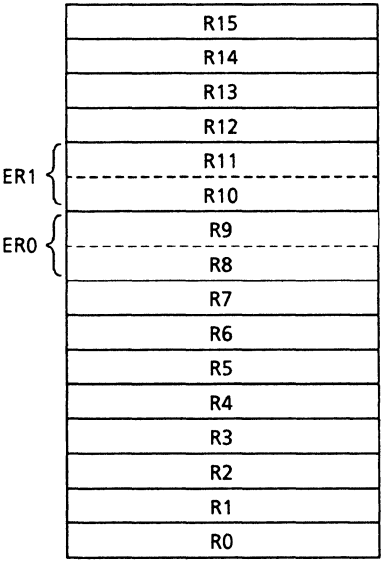
Program status word (PSW)



Program counter (PC)



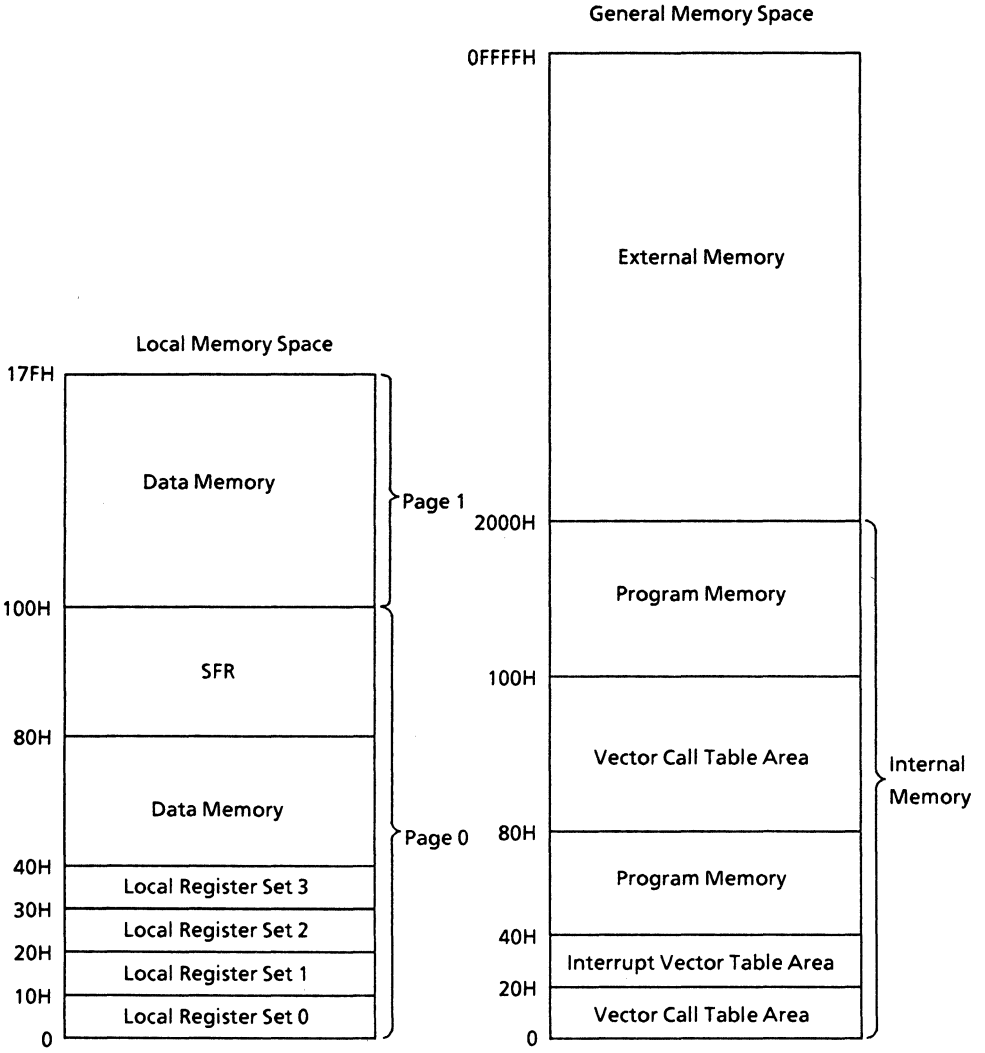
LOCAL REGISTERS



*1 4 banks of local registers are mapped in local memory space data memory.

*2 Registers R8 to R11 can be used as 16-bit registers, ER0 and ER1.

MEMORY MAPS


General Memory Space
0FFFFH
External Memory
2000H
Program Memory
100H
Vector Call Table Area
80H
Program Memory
40H
Interrupt Vector Table Area
20H
Vector Call Table Area
0
Internal Memory
Local Memory Space
17FH
Data Memory
Page 1
100H
SFR
80H
Data Memory
Page 0
40H
Local Register Set 3
30H
Local Register Set 2
20H
Local Register Set 1
10H
Local Register Set 0
0

SFR TABLE

Address (HEX)	Name	Sym-bol	Symbol	R/W	Reset
0FF	F register (multiplication/division operation register)	FER	FR	R/W	00H
0FE	E register (multiplication/division operation register)		ER		00H
0FD	D register (multiplication/division operation register)	DCR	DR		00H
0FC	C register (multiplication/division operation register)		CR		00H
0FB	Multiplication/division condition register		MDCR		0FCH
0F7	Interrupt enable register	IE	IEH	R/W	70H
0F6			IEL		00H
0F5	Interrupt request register	IRQ	IRQH		70H
0F4			IRQL		00H
0F3	External interrupt control register		XICON		0C0H
0F2	Standby control register		SBYCON		0F0H
0F1	Stack page specification register		SPR		0FEH
0F0	Watchdog timer control register		WDTCON	W	—
0EE	Port 3 mode register		P3MOD	R/W	53H
0ED	Port 3 direction register		P3DIR		00H
0EC	Port 3 data register		P3D		Unde-fined
0EA	Port 2 mode register		P2MOD		3CH
0E9	Port 2 direction register		P2DIR		00H
0E8	Port 2 data register		P2D		Unde-fined
0E5	Port 1 direction register		P1DIR		00H
0E4	Port 1 data register		P1D		Unde-fined
0E3	Port 0 direction register		P0DIR		00H
0E2	Port 0 data register		P0D		Unde-fined
0E1	Shift register		SFTR		Unde-fined
0E0	Shift register control register		SFTCON		0E0H

Address (HEX)	Name	Symbol	Symbol	R/W	Reset
0DF	Timer 1 register	T01R	T1R	R/W	Undefined
0DE	Timer 0 register		T0R		Undefined
0DD	Timer 1 counter	T01C	T1C		Undefined
0DC	Timer 0 counter		T0C		Undefined
0DB	Timer 1 control register		T1CON		0F0H
0DA	Timer 0 control register		T0CON		0E0H
0D9	Time base counter control register		TBCON		0E0H
0D7	Timer 2 register	T2R	T2RH		Undefined
0D6			T2RL		Undefined
0D5	Timer 2 counter	T2C	T2CH		Undefined
0D4			T2CL		Undefined
0D3	Timer 2 control register		T2CON		0F4H
0D2	Free-running counter	FRC	FRCH		00H
0D1			FRCL		00H
0D0	Free-running counter control register		FRCON		0F4H
0CF	Compare channel 1 data register	CMP1R	CMP1RH		Undefined
0CE			CMP1RL		Undefined
0CD	Compare channel 0 data register	CMP0R	CMP0RH		Undefined
0CC			CMP0RL		Undefined
0CB	Compare output control register		CMPCON		0F0H
0CA	Compare output data register		CMPOUT		0FCH
0C9	Capture data register	CAPR	CAPRH	R	Undefined
0C8			CAPRL		Undefined
0C7	Capture input control register		CAPCON	R/W	0FCH
0C6	Serial port buffer		SBUF		Undefined
0C5	Serial port control register	SCON	SCONH		0F0H
0C4			SCONL		00H
0C3	Timer 3 register		T3R		Undefined
0C2	Timer 3 counter		T3C		Undefined
0C1	Timer 3 control register		T3CON		0F4H

Address (HEX)	Name	Symbol	R/W	Reset
0BE	Port 6	P6	R	—
0BB	Port 5 direction register	P5DIR	R/W	00H
0BA	Port 5 data register	P5D		Unde-fined
0B9	Port 4 direction register	P4DIR		00H
0B8	Port 4 data register	P4D		Unde-fined

ADDRESSING MODES

MSM65513 has 384 bytes of local memory space and 64 Kbytes of general memory space. A variety of addressing modes are available for accessing these spaces.

1. Register Direct Addressing

   - A、B、SP、PSW
   - BA

Example

```
DAA    A
       ═══►  ACC
```

2. Local Register Direct Addressing

   - Rn (n = 0~15)
   - ERn (n = 0, 1)

Example

```
L      A, R2
       ═══►  Local Registers
                 R3
                 R2
                 R1
```

3. Local Memory Direct Addressing

   - \adrs (within 256-byte page; page specified by LMP flag)
   - adrs (384 bytes)

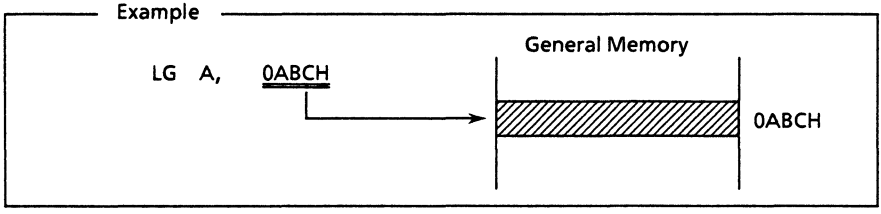
Example

```
L      A, \56H
       ═══►  Local Memory
                 56H
```

4. General Memory Direct Addressing

- adrs

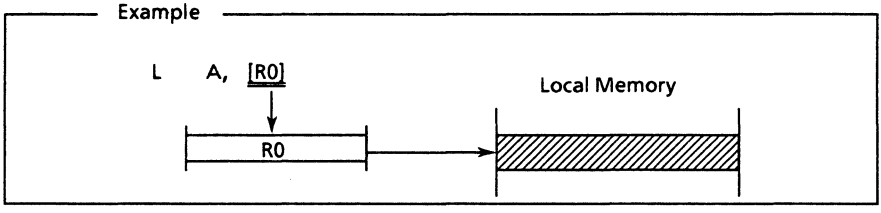
Example



5. Local Memory - Register Indirect Addressing

- [Rn] (n = 0, 1, 8, 9; within 256-byte page; page specified by LMP flag)
- page: [Rn] (page = 0, 1 n = 0, 1, 8, 9 384 bytes)

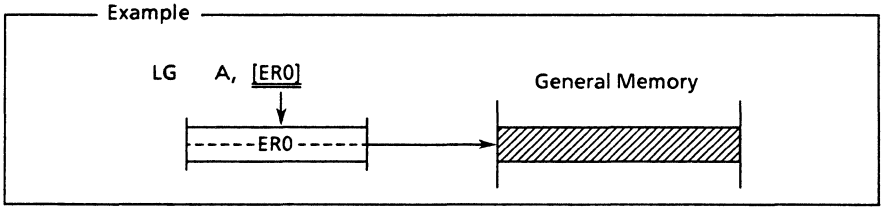
Example



6. General Memory - Register Indirect Addressing

- [ERn] (n = 0, 1)
- [BA]

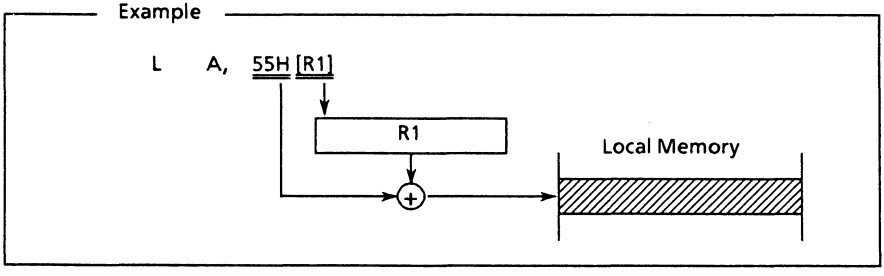
Example



7. Local Memory Index Addressing

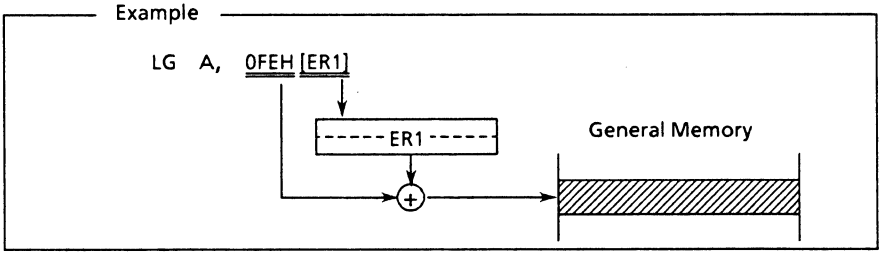
- disp [Rn] (n = 1, 9; within 256-byte page; page specified by LMP flag)
- page: disp [Rn] (page = 0, 1 n = 1, 9 384 bytes)

Example



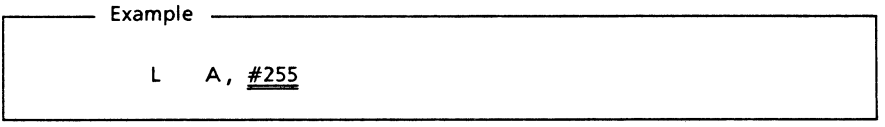
8. General Memory Index Addressing

- disp [ER1]



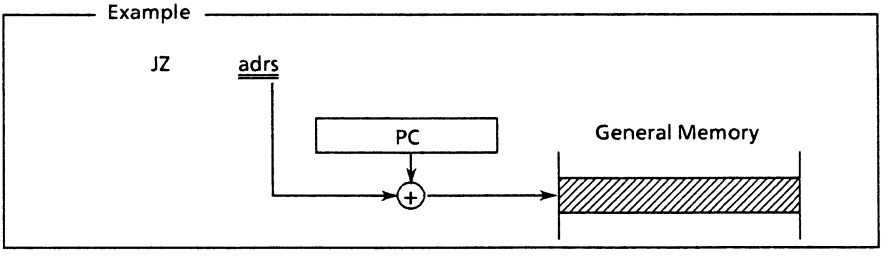
9. Immediate Addressing

- #n



10. PC Relative Addressing

- adrs



INSTRUCTION TABLES

● Data Transfer Instructions

Mnemonic		Function
L	obj1, obj2	Local memory load
LG	obj1, obj2	General memory load
ST	obj1, obj2	Store into local memory
STG	obj1, obj2	Store into general memory
MOV	PSW, #n	Immediate data transfer to PSW
MOV	obj1, obj2	Data transfer
MOVG	obj1, obj2	General memory data transfer
MOVW	obj1, obj2	16-bit data transfer
XCH	C, P	Carry and parity exchange
XCH	obj1, obj2	Data exchange
SWAP	obj	Upper nibble and lower nibble swap

• Increment and Decrement

Mnemonic		Function
INC	obj	Data increment
INCG	obj	General memory increment
INCW	obj	16-bit data increment
DEC	obj	Data decrement
DECG	obj	General memory decrement
DECW	obj	16-bit data decrement

• Arithmetic Operations

Mnemonic		Function
ADD	obj1, obj2	Data add
ADDW	obj1, obj2	16-bit data add
ADC	obj1, obj2	Data add with carry
ADCG	obj1, obj2	General memory data add with carry
SUB	obj1, obj2	Data subtract
SUBW	obj1, obj2	16-bit data subtract
SBC	obj1, obj2	Data subtract with carry
SBCG	obj1, obj2	General memory data subtract with carry
MUL		Multiplication 16 ← 8 x 8
DIV		Division 16 ← 16/8, 8 ← 16 mod 8

• Comparisons

Mnemonic		Function
CMP	obj1, obj2	Data compare
CMPW	obj1, obj2	16-bit data compare

• Logical Operations

Mnemonic		Function
AND	PSW, #n	PSW and immediate data logical AND
AND	obj1, obj2	Data logical AND
OR	PSW, #n	PSW and immediate data logical OR
OR	obj1, obj2	Data logical OR
XOR	obj1, obj2	Data exclusive OR

• Bit Operations

Mnemonic		Function
SB	obj. n	Bit set
SB	obj	PSW bit set
RB	obj. n	Bit reset
RB	obj	PSW bit reset
CPL	C	Carry complement
L	C, obj	Bit transfer to carry
ST	C, obj	Bit transfer from carry

• Rotate and Shift

Mnemonic		Function
ROL	obj	Rotate left
ROR	obj	Rotate right
SLL	obj	Shift left
SRL	obj	Shift right

• Decimal Adjust

Mnemonic		Function
DAA	obj	Decimal adjust after add
DAS	obj	Decimal adjust after subtract

• Conditional Jumps

Mnemonic		Function
JZ	adrs	Jump if zero flag is set
JNZ	adrs	Jump if zero flag is not set
JC	adrs	Jump if carry is set
JNC	adrs	Jump if carry is not set
DJZ	Rn, adrs	Decrement register, and jump if zero
DJNZ	Rn, adrs	Decrement register, and jump if not zero
JBS	obj. n, adrs	Jump if bit is set
JBR	obj. n, adrs	Jump, if bit is reset
JBSC	obj. n, adrs	Jump and clear bit if bit is set
CJE	C, P, adrs	Compare carry and parity; jump if equal
CJNE	C, P, adrs	Compare carry and parity; jump if not equal
CJE	obj1, obj2, adrs	Compare; jump if equal
CJNE	obj1, obj2, adrs	Compare; jump if not equal
CJEG	obj1, obj2, adrs	Compare with general memory data; jump if equal
CJNEG	obj1, obj2, adrs	Compare with general memory data; jump if not equal

• Jumps

Mnemonic		Function
J	adrs	Jump
SJ	adrs	Short jump
J	[BA]	Indirect jump

• Subroutines

Mnemonic		Function
PUSH	obj	Data push
POP	obj	Data pop
CAL	adrs	Subroutine call
CALZ	adrs	Call subroutine if zero flag is set
CALC	adrs	Call subroutine if carry flag is set
VCAL	n	Vector call
VCALZ	n	Vector call if zero flag is set
VCALC	n	Vector call if carry flag is set
RT		Return from subroutine
RTZ		Return from subroutine if zero flag is set
RTC		Return from subroutine if carry flag is set

• Other Instructions

Mnemonic		Function
CLR	obj	Clear
CLRW	BA	16-bit data clear
CPL	obj	Data complement
CPLW	BA	16-bit data complement
NOP		No operation
CHK	obj	Parity check
DLY	n	Program execution delay

Preliminary

OKI semiconductor
MSM65524/65P524

OKI ORIGINAL HIGH PERFORMANCE CMOS 8 BIT SINGLE CHIP MICROCONTROLLER

GENERAL DESCRIPTION

MSM65524 is a high-performance 8-bit single-chip controller that employs Oki's original nX-8/50 CPU core. With a minimum instruction execution time of 400 ns (10MHz clock), the MSM65524 is capable of high-speed processing, and includes 16 Kbytes of program memory, 384 bytes of data memory, timers, serial ports, an A/D converter and PWMs on chip. Also available is the MSM65P524, which replaces the on-chip program memory with one-time PROM.

OPERATING RANGE

- Operating Frequency : DC ~ 10 MHz
- Operating Voltage : 4.5 ~ 5.5 V
- Operating Temperature : -40 ~ 85°C (MSM65524)

FEATURES

- Memory Space : 64 Kbytes

  On-Chip Program Memory : 16 Kbytes
  On-Chip Data Memory : 384 bytes
- Minimum Instruction Execution Cycle:
  400nS @ 10 MHz
- Powerful instruction set:

  83 basic instructions
  8/16-bit operation instructions
  Bit manipulation instructions
  Compound function instructions
- Abundant addressing modes
- Multiplication/division operation functions
  16 ← 8 x 8
  16 ← 16/8, 8 ← 16 mod 8
- I/O ports: 8-bit x 5
  4-bit x 1
  Input only port: 8-bit x 1
- Timers
  8-bit auto-reload timer x 2
  16-bit auto-reload timer x 1
  Watchdog timer x 1
- Counters
  Time base counter x 1
  16-bit free-running counter x 1
- Capture input: 1 channel
- Compare output: 2 channels
- Serial ports

  Shift register x 1

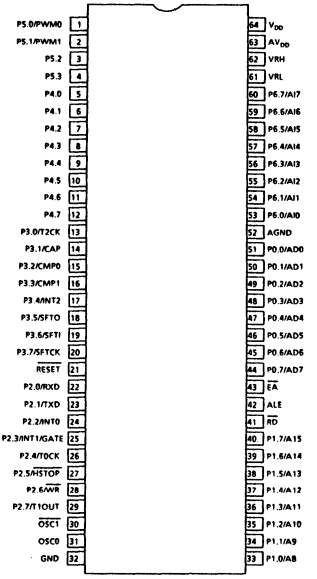
  Serial port with baud rate generator (UART/synchronous) x 1
- A/D converter: 8 bits x 8 channels
- PWM: 8 bits x 2 channels

  PWM with auto-reload timer for period setting
- External interrupts: 3
- Interrupt factors: 19
- Package:
  64 pin plastic shrink DIP (SDIP64-P-750)
  64 pin plastic QFP (T.B.D)
  68 pin PLCC (QFJ68-P-S950)

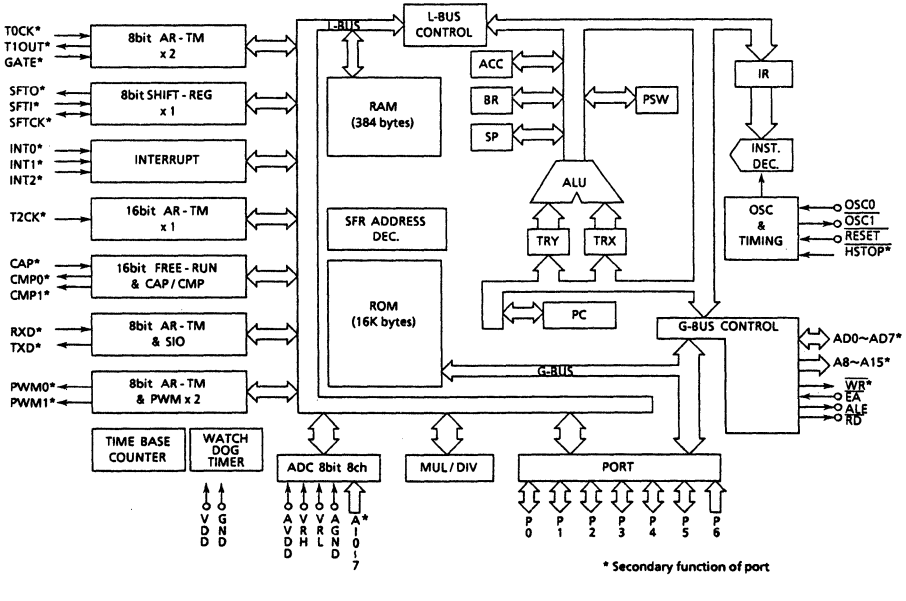
* Specifications are subject to change without notice.

PIN CONFIGURATION

64-Pin Plastic Shrink DIP
(Top View)



FUNCTIONAL BLOCK DIAGRAM



PIN DESCRIPTIONS

Type	Pin Name	I/O	Description
Power supply	VDD		+ 5 V digital power supply
	GND		0 V digital ground
	AVDD		+ 5 V analog power supply
	AGND		0 V analog ground
	VRH		+ 5 V analog reference voltage
	VRL		0 V analog reference voltage
Oscillation	OSC0	Input	System clock input pin. Quartz oscillator or ceramic oscillator is connected between OSC0 and $\overline{OSC1}$. For external clock, input at OSC0, leaving $\overline{OSC1}$ open.
	$\overline{OSC1}$	Output	System clock output pin
Control	$\overline{RESET}$	Input	System reset input (program starts from address 0040H); internal pull-up resistance
	$\overline{EA}$	Input	Program memory select input pin. "L" level input for external program memory; "H" level input for internal program memory.
	$\overline{RD}$	Output	Read strobe signal during external memory access
	ALE	Output	Address latch signal during external memory access
Port	PORT 0	I/O	8-bit I/O port During external memory access, becomes address/data bus for address output, instruction fetch or data read/write along with ALE, $\overline{RD}$ and $\overline{WR}$ pins
	PORT 1	I/O	8-bit I/O port Address bus during external memory access
	PORT 2 PORT 3 PORT 4	I/O	8-bit I/O port x 3. Secondary functions shown in following table are added for ports 2 and 3.
	PORT 5	I/O	4-bit I/O port. Secondary functions shown in following table are added for port 5.
	PORT 6	Input	8-bit I/O port. Functions as analog input channel during A/D conversion.

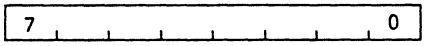
PIN SECONDARY FUNCTIONS

Pin Name	I/O	Description
RXD	I/O	P2.0 secondary functions. UART: Input pin for serial port receive data. Synchronous: Input pin for serial port transmit/receive data.
TXD	Output	P2.1 secondary functions. UART: Output pin for serial port transmit data. Synchronous: Output pin for serial port synchronizing clock.
INT0	Input	P2.2 secondary function. External interrupt 0 input pin.
INT1/GATE	Input	P2.3 secondary functions. External interrupt 1 input pin. Also used as input pin for gate signal for timer 0 count enable /disable.
T0CK	Input	P2.4 secondary function Timer 0 external clock input pin.
$\overline{\text{HSTOP}}$	Input	P2.5 secondary function. Hard stop mode input pin; stops system clock oscillation with "L" level input.
$\overline{\text{WR}}$	Output	P2.6 secondary function. Write strobe signal output pin during external data memory access.
T1OUT	Output	P2.7 secondary function. Output pin for signal that 2-divided timer 1 overflow.
T2CK	Input	P3.0 secondary function. Timer 2 external clock input pin.
CAP	Input	P3.1 secondary function. Capture trigger input pin.
CMP0	Output	P3.2 secondary function. Compare output channel 0 output pin.
CMP1	Output	P3.3 secondary function. Compare output channel 1 output pin.
INT2	Input	P3.4 secondary function. External interrupt 2 input signal.
SFTO	Output	P3.5 secondary function. Shift register data output pin.
SFTI	Input	P3.6 secondary function. Shift register data input signal.
SFTCK	I/O	P3.7 secondary function. Shift register synchronizing clock input/output signal.
PWM0	Output	P5.0 secondary function. PWM channel 0 output signal.
PWM1	Output	P5.1 secondary function. PWM channel 1 output signal.

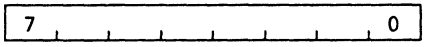
REGISTERS

● Operation Registers

A Register (Accumulator)



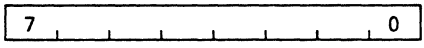
B Register (BR)



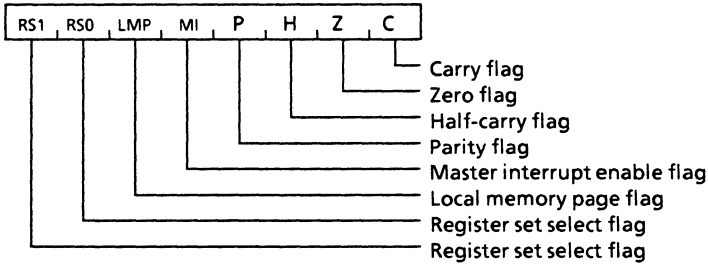
* For 16-bit operation instruction, A register holds low byte data and the B register holds high byte data.

● Special Purpose Registers

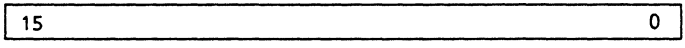
Stack pointer (SP)



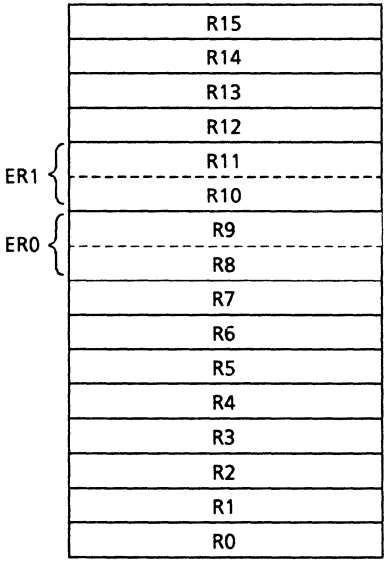
Program status word (PSW)



Program counter (PC)



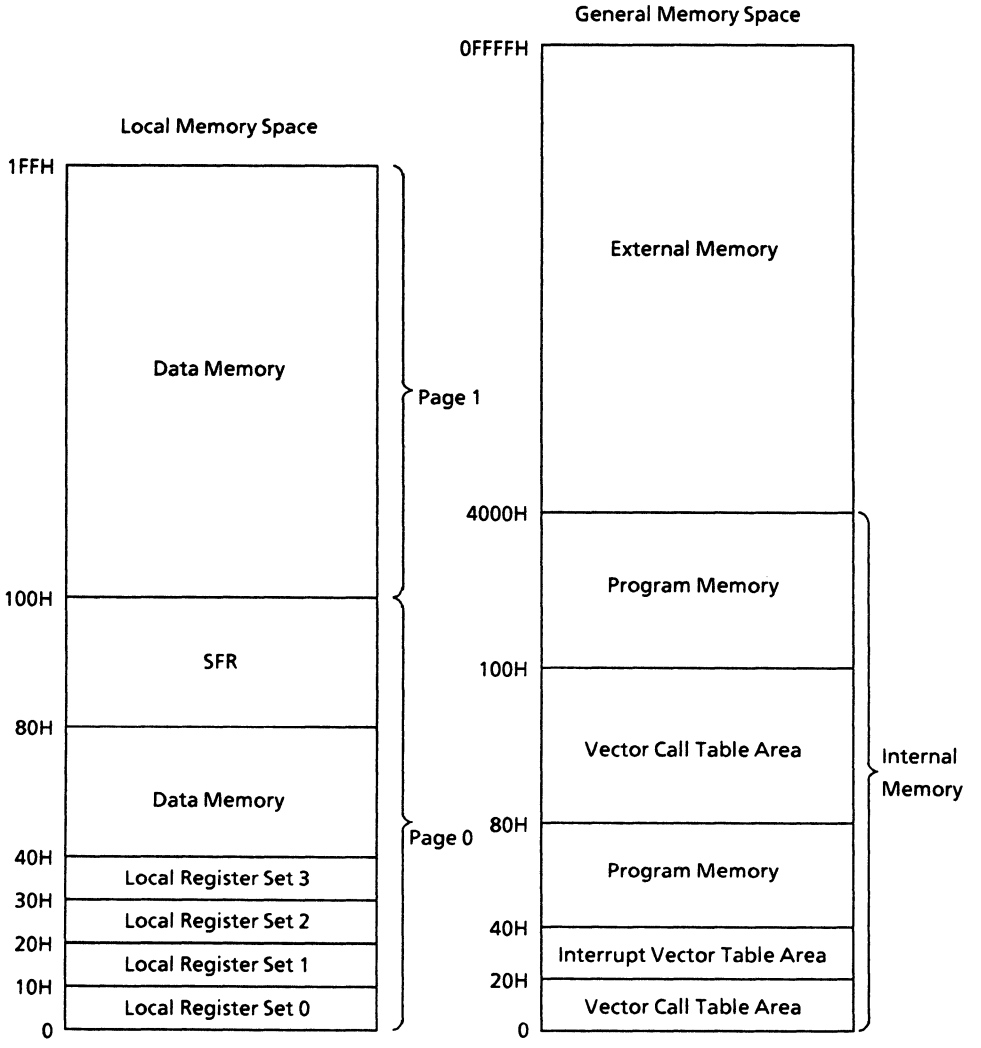
LOCAL REGISTERS



*1 4 banks of local registers are mapped in local memory space data memory.

*2 Registers R8 to R11 can be used as 16-bit registers, ER0 and ER1.

MEMORY MAPS


General Memory Space
0FFFFH
External Memory
4000H
Program Memory
100H
Vector Call Table Area
80H
Program Memory
40H
Interrupt Vector Table Area
20H
Vector Call Table Area
0
Internal Memory
Local Memory Space
1FFH
Data Memory
100H
SFR
80H
Data Memory
40H
Local Register Set 3
30H
Local Register Set 2
20H
Local Register Set 1
10H
Local Register Set 0
0
Page 1
Page 0

SFR TABLE

Address (HEX)	Name	Sym-bol	Symbol	R/W	Reset
0FF	F register (multiplication/division operation register)	FER	FR	R/W	00H
0FE	E register (multiplication/division operation register)		ER		00H
0FD	D register (multiplication/division operation register)	DCR	DR		00H
0FC	C register (multiplication/division operation register)		CR		00H
0FB	Multiplication/division condition register		MDCR		0FCH
0F7	Interrupt enable register	IE	IEH	R/W	00H
0F6			IEL		00H
0F5	Interrupt request register	IRQ	IRQH		00H
0F4			IRQL		00H
0F3	External interrupt control register		XICON		0C0H
0F2	Standby control register		SBYCON		0F0H
0F1	Stack page specification register		SPR		0FEH
0F0	Watchdog timer control register		WDTCON	W	—
0EE	Port 3 mode register		P3MOD	R/W	0D3H
0ED	Port 3 direction register		P3DIR		00H
0EC	Port 3 data register		P3D		Unde-fined
0EA	Port 2 mode register		P2MOD		3CH
0E9	Port 2 direction register		P2DIR		00H
0E8	Port 2 data register		P2D		Unde-fined
0E5	Port 1 direction register		P1DIR		00H
0E4	Port 1 data register		P1D		Unde-fined
0E3	Port 0 direction register		P0DIR		00H
0E2	Port 0 data register		P0D		Unde-fined
0E1	Shift register		SFTR		Unde-fined
0E0	Shift register control register		SFTCON		0E0H

Address (HEX)	Name	Symbol	Symbol	R/W	Reset
0DF	Timer 1 register	T01R	T1R	R/W	Undefined
0DE	Timer 0 register		T0R		Undefined
0DD	Timer 1 counter	T01C	T1C		Undefined
0DC	Timer 0 counter		T0C		Undefined
0DB	Timer 1 control register		T1CON		0F0H
0DA	Timer 0 control register		T0CON		0E0H
0D9	Time base counter control register		TBCON		0E0H
0D7	Timer 2 register	T2R	T2RH		Undefined
0D6			T2RL		Undefined
0D5	Timer 2 counter	T2C	T2CH		Undefined
0D4			T2CL		Undefined
0D3	Timer 2 control register		T2CON		0F4H
0D2	Free-running counter	FRC	FRCH		00H
0D1			FRCL		00H
0D0	Free-running counter control register		FRCON		0F4H
0CF	Compare channel 1 data register	CMP1R	CMP1RH		Undefined
0CE			CMP1RL		Undefined
0CD	Compare channel 0 data register	CMP0R	CMP0RH		Undefined
0CC			CMP0RL		Undefined
0CB	Compare output control register		CMPCON		0F0H
0CA	Compare output data register		CMPOUT		0FCH
0C9	Capture data register	CAPR	CAPRH	R	Undefined
0C8			CAPRL		Undefined
0C7	Capture input control register		CAPCON	R/W	0FCH
0C6	Serial port buffer		SBUF		Undefined
0C5	Serial port control register	SCON	SCONH		0F0H
0C4			SCONL		00H
0C3	Timer 3 register		T3R		Undefined
0C2	Timer 3 counter		T3C		Undefined
0C1	Timer 3 control register		T3CON		0F4H

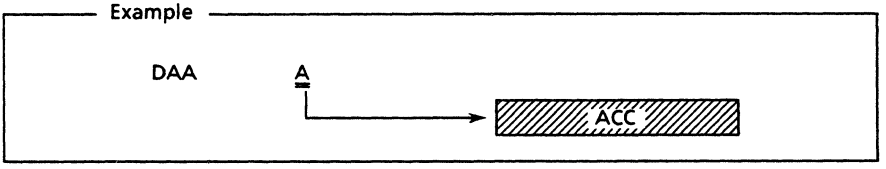
Address (HEX)	Name	Symbol	R/W	Reset
0BE	Port 6	P6	R	—
0BC	Port 5 mode register	P5MOD	R/W	0FCH
0BB	Port 5 direction register	P5DIR		00H
0BA	Port 5 data register	P5D		Undefined
0B9	Port 4 direction register	P4DIR		00H
0B8	Port 4 data register	P4D		Undefined
0B7	PWM1 register	PWM1R		00H
0B6	PWM0 register	PWM0R		00H
0B5	PWM control register	PWMCON		0CH
0B3	Timer 4 register	T4R		Undefined
0B2	Timer 4 counter	T4C		Undefined
0B1	Timer 4 control register	T4CON		0F4H
0AF	A/D converter result register 3	ADR3	R	Undefined
0AE	A/D converter result register 2	ADR2		Undefined
0AD	A/D converter result register 1	ADR1		Undefined
0AC	A/D converter result register 0	ADR0		Undefined
0AB	A/D converter control register	ADCON	R/W	0F0H

ADDRESSING MODES

MSM65524 has 512 bytes of local memory space and 64 Kbytes of general memory space. A variety of addressing modes are available for accessing these spaces.

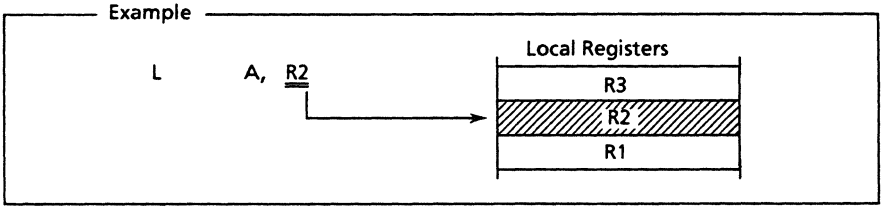
1. Register Direct Addressing

   - A、B、SP、PSW
   - BA



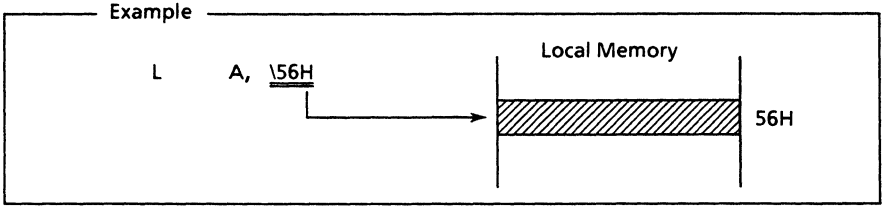
2. Local Register Direct Addressing

   - Rn (n = 0~15)
   - ERn (n = 0, 1)



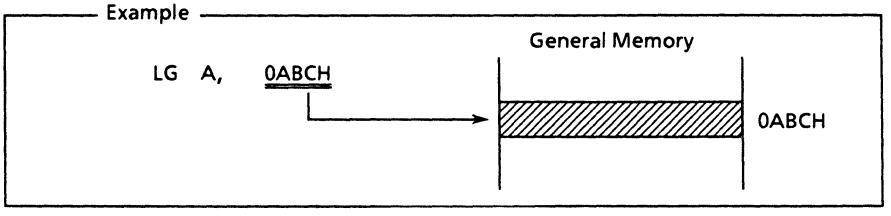
3. Local Memory Direct Addressing

   - \adrs (within 256-byte page; page specified by LMP flag)
   - adrs (512 bytes)



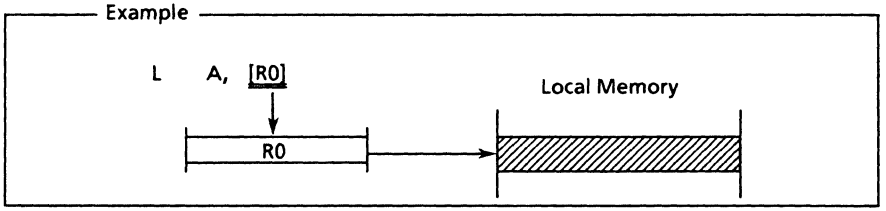
4. General Memory Direct Addressing

   - adrs



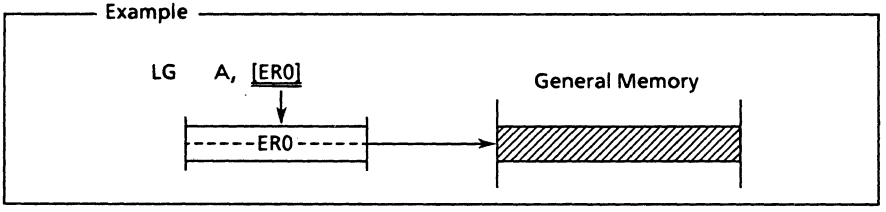
5. Local Memory - Register Indirect Addressing

- [Rn] (n = 0, 1, 8, 9; within 256-byte page; page specified by LMP flag)
- page: [Rn] (page = 0, 1 n = 0, 1, 8, 9 512 bytes)



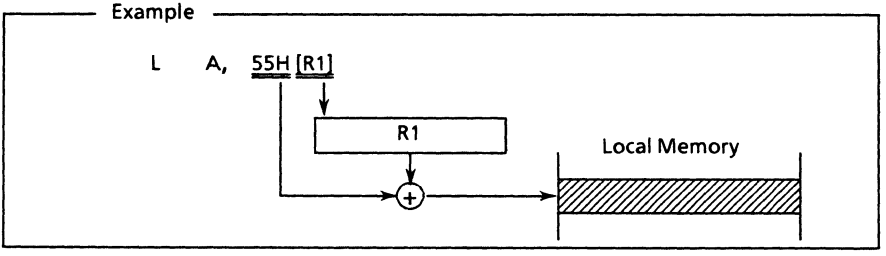
6. General Memory - Register Indirect Addressing

- [ERn] (n = 0, 1)
- [BA]



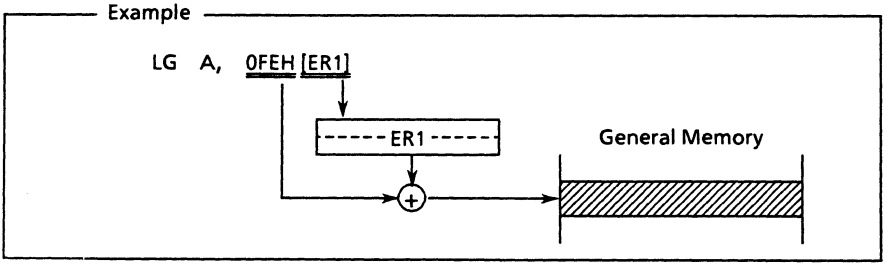
7. Local Memory Index Addressing

- disp [Rn] (n = 1, 9; within 256-byte page; page specified by LMP flag)
- page: disp [Rn] (page = 0, 1 n = 1, 9 512 bytes)



8. General Memory Index Addressing

- disp [ER1]



9. Immediate Addressing

- #n

Example

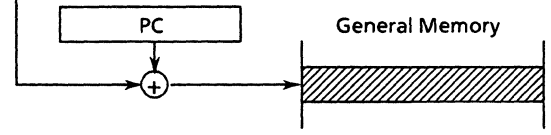
```
L    A, #255
```

10. PC Relative Addressing

- adrs

Example

```
JZ   adrs
```



INSTRUCTION TABLES

• Data Transfer Instructions

Mnemonic		Function
L	obj1, obj2	Local memory load
LG	obj1, obj2	General memory load
ST	obj1, obj2	Store into local memory
STG	obj1, obj2	Store into general memory
MOV	PSW, #n	Immediate data transfer to PSW
MOV	obj1, obj2	Data transfer
MOVG	obj1, obj2	General memory data transfer
MOVW	obj1, obj2	16-bit data transfer
XCH	C, P	Carry and parity exchange
XCH	obj1, obj2	Data exchange
SWAP	obj	Upper nibble and lower nibble swap

• Increment and Decrement

Mnemonic		Function
INC	obj	Data increment
INCG	obj	General memory increment
INCW	obj	16-bit data increment
DEC	obj	Data decrement
DECG	obj	General memory decrement
DECW	obj	16-bit data decrement

● Arithmetic Operations

Mnemonic		Function
ADD	obj1, obj2	Data add
ADDW	obj1, obj2	16-bit data add
ADC	obj1, obj2	Data add with carry
ADCG	obj1, obj2	General memory data add with carry
SUB	obj1, obj2	Data subtract
SUBW	obj1, obj2	16-bit data subtract
SBC	obj1, obj2	Data subtract with carry
SBCG	obj1, obj2	General memory data subtract with carry
MUL		Multiplication 16 ← 8 x 8
DIV		Division 16 ← 16/8, 8 ← 16 mod 8

● Comparisons

Mnemonic		Function
CMP	obj1, obj2	Data compare
CMPW	obj1, obj2	16-bit data compare

● Logical Operations

Mnemonic		Function
AND	PSW, #n	PSW and immediate data logical AND
AND	obj1, obj2	Data logical AND
OR	PSW, #n	PSW and immediate data logical OR
OR	obj1, obj2	Data logical OR
XOR	obj1, obj2	Data exclusive OR

● Bit Operations

Mnemonic		Function
SB	obj. n	Bit set
SB	obj	PSW bit set
RB	obj. n	Bit reset
RB	obj	PSW bit reset
CPL	C	Carry complement
L	C, obj	Bit transfer to carry
ST	C, obj	Bit transfer from carry

• Rotate and Shift

Mnemonic		Function
ROL	obj	Rotate left
ROR	obj	Rotate right
SLL	obj	Shift left
SRL	obj	Shift right

• Decimal Adjust

Mnemonic		Function
DAA	obj	Decimal adjust after add
DAS	obj	Decimal adjust after subtract

• Conditional Jumps

Mnemonic		Function
JZ	adrs	Jump if zero flag is set
JNZ	adrs	Jump if zero flag is not set
JC	adrs	Jump if carry is set
JNC	adrs	Jump if carry is not set
DJZ	Rn, adrs	Decrement register, and jump if zero
DJNZ	Rn, adrs	Decrement register, and jump if not zero
JBS	obj. n, adrs	Jump if bit is set
JBR	obj. n, adrs	Jump, if bit is reset
JBSC	obj. n, adrs	Jump and clear bit if bit is set
CJE	C, P, adrs	Compare carry and parity; jump if equal
CJNE	C, P, adrs	Compare carry and parity; jump if not equal
CJE	obj1, obj2, adrs	Compare; jump if equal
CJNE	obj1, obj2, adrs	Compare; jump if not equal
CJEG	obj1, obj2, adrs	Compare with general memory data; jump if equal
CJNEG	obj1, obj2, adrs	Compare with general memory data; jump if not equal

• Jumps

Mnemonic		Function
J	adrs	Jump
SJ	adrs	Short jump
J	[BA]	Indirect jump

• Subroutines

Mnemonic		Function
PUSH	obj	Data push
POP	obj	Data pop
CAL	adrs	Subroutine call
CALZ	adrs	Call subroutine if zero flag is set
CALC	adrs	Call subroutine if carry flag is set
VCAL	n	Vector call
VCALZ	n	Vector call if zero flag is set
VCALC	n	Vector call if carry flag is set
RT		Return from subroutine
RTZ		Return from subroutine if zero flag is set
RTC		Return from subroutine if carry flag is set

• Other Instructions

Mnemonic		Function
CLR	obj	Clear
CLRW	BA	16-bit data clear
CPL	obj	Data complement
CPLW	BA	16-bit data complement
NOP		No operation
CHK	obj	Parity check
DLY	n	Program execution delay

INTEL COMPATIBLE SERIES

MSM80C35/48
MSM80C39/49
MSM80C40/50

CMOS 8-BIT SINGLE CHIP MICROCONTROLLER

GENERAL DESCRIPTION

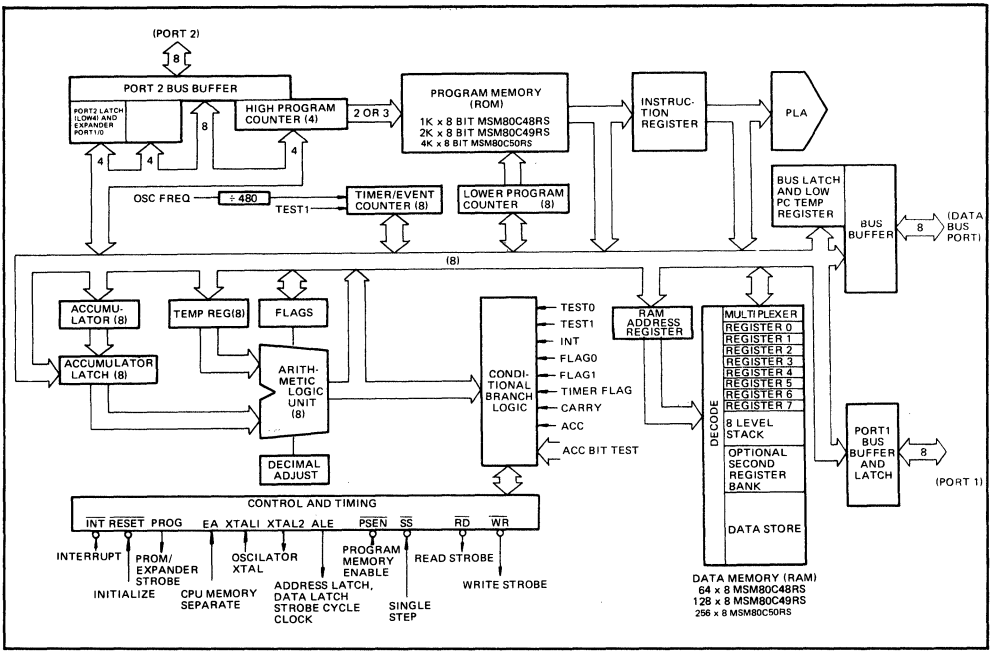
The OKI MSM80C48/MSM80C49/MSM80C50 microcontroller is a low-power, high-performance 8-bit single chip device implemented in silicon gate complementary metal oxide semiconductor technology. Integrated within these chips are 8K/16K/32K bits of mask program ROM, 512/1024/2048 bits of data RAM, 27 I/O lines, built-in 8 bit timer/counter, and oscillator. Program memory and data paths are byte wide. Eleven new instructions have been added to the NMOS version's instruction set, thereby optimizing power down, port data transfer, decrement and port float functions.

Available in 40-pin plastic DIP (RS) or 44-pin plastic flat packages (GSK).

FEATURES

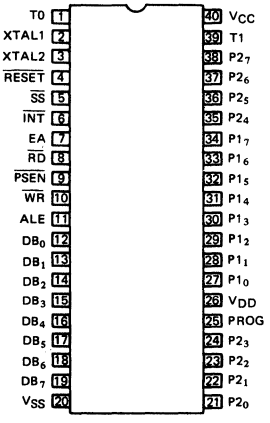
- Lower power consumption enabled by CMOS silicon gate process
- Completely static operation
- Improved power-down feature
- Minimum instruction cycle 1.36 μs (11MHz) @ V_{CC} = +5V ±10%
  11 MHz version of MSM80C40/50 (6 MHz < XTAL1.2 < 11 MHz) is under development.
- Every signal input terminal is provided with a Schmitt circuit, except XTAL1 Pin.
- Every signal output terminal is capable of driving a standard TTL, except XTAL2 Pin.
- 111 instructions
- All instructions are usable even during execution of external ROM instructions.
- Operation facility
  Addition, logical operations, and decimal adjust
- Program memory (ROM) : 1K × 8 bits (MSM80C48)
  : 2K × 8 bits (MSM80C49)
  : 4K × 8 bits (MSM80C50)
- Data memory (RAM) : 64 × 8 bits (MSM80C48)
  : 128 × 8 bits (MSM80C49)
  : 256 × 8 bits (MSM80C50)
- Two sets of working registers
- External and timer interrupts
- Two test inputs
- Built-in 8-bit timer counter
- Extendable external memory and I/O ports
- Input/output ports : Input/output ports – 8 bits × 2
  : Data bus input/output – 8 bits × 1
- Single-step execution function
- Wide range of operating voltage, from +2.5V to +6V of V_{CC}.
- High noise margin action
- Compatible with Intel's 8048, 8049 and 8050
- Package:
  40 pin plastic DIP (DIP40-P-600)
  44 pin plastic QFP (QFP44-P-910-K)
  44 pin plastic QFP (QFP44-P-910-VK)

FUNCTIONAL BLOCK DIAGRAM

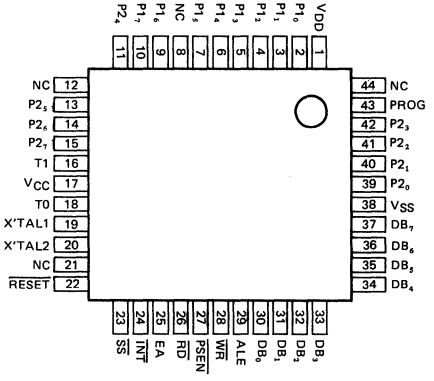


PIN CONFIGURATION

(Top View) 40 Lead Plastic DIP



(Top View) 44 Lead Plastic FLAT



Pin Name

$P1_0 \sim P1_7$	: Input/output port (PORT1)	ALE	: Address Latch Enable
$P2_0 \sim P2_7$	: Input/output port (PORT2)	$\overline{PSEN}$	: Program Store Enable
$DB_0 \sim DB_7$	: Data bus port (BUS PORT)	$\overline{RESET}$	: Reset
T0, T1	: Test	$\overline{SS}$	: Single Step
$\overline{INT}$	: Interrupt	EA	: ROM Mode
$\overline{RD}$	: Read	XTAL 1, 2	: Crystal Controlled Oscillator
$\overline{WR}$	: Write		

PIN DESCRIPTION

Designation	Input/Output	Function
$P1_0 \sim P1_7$ (PORT 1)	Input/Output	8-bit quasi-bidirectional port
$P2_0 \sim P2_7$ (PORT 2)	Input/Output	8-bit quasi-bidirectional port The high-order four bits of external program memory addresses can be output from P20-P23, to which the I/O expander MSM82C43RS may also be connected.
$DB_0 \sim DB_7$ (BUS)	Input/Output	Bidirectional port The low-order eight bits of external program memory address can be output from this port, and the addressed instruction is fetched under the control of $\overline{PSEN}$ signal. Also, the external data memory address is output, and data is read and written synchronously using $\overline{RD}$ and $\overline{WR}$ signals. The port can also serve as either a statically latched output port or a non-latching input port.
T0 (Test 0)	Input/Output	The input can be tested with the conditional jump instructions JT0 and JNT0. The execution of the ENT0 CLK instruction causes a clock output to be generated.
T1	Input	The input can be tested with the conditional jump instructions JT1 and JNT1. The execution of a STRT CNT instruction causes an internal counter input to be activated.
$\overline{INT}$ (Interrupt)	Input	Interrupt input. If interrupt is enabled, $\overline{INT}$ input initiates an interrupt. Interrupt is disabled after a reset. Also testable with a JNI instruction. Can be used to terminate the power-down mode. (Active "0" level)
$\overline{RD}$ (Read)	Output	A signal to read data from external data memory. (Active "0" level)
$\overline{WR}$ (Write)	Output	A signal to write data to external data memory. (Active "0" level)
ALE Address & Data Latch Clock	Output	This signal is generated in each cycle. It may be used as a clock output. External data memory or external program memory is addressed upon the falling edge. For the external ROM, this signal is used to latch the bus port data upon the ALE signal rise-up after the execution of the OUTL BUS, A instruction.
$\overline{PSEN}$ Program Store Enable	Output	A signal to fetch an instruction from external program memory (Active "0" level)
$\overline{RESET}$	Output	$\overline{RESET}$ input initialize the processor. (Active "0" level) Used to terminate the power-down mode.
$\overline{SS}$ (Single Step)	Output	A program is executed step by step. This pin can also be used to control internal oscillation when the power-down mode is reset. (Active "0" level)
EA (External Access)	Input	When held at high level, all instructions are fetched from external memory. (Active "1" level)
$\overline{PROG}$ (Expander Strobe)	Output	This output strobes the MSM82C43RS I/O expander.

PIN DESCRIPTION (CONT.)

Designation	Input/Output	Function
XTAL 1 (Crystal 1)	Input	One side of the internal crystal oscillator. An external clock can also be input.
XTAL 2 (Crystal 2)	Output	Other side of the internal crystal oscillator.
V_{CC}	–	Power supply terminal
V_{DD}	–	Standby control input. Normally, "1" level. When set to "0" level, oscillation is stopped and processor goes into standby mode.
V_{SS}	–	GND

Note: The required $\overline{\text{RESET}}$ pulse duration is at least two machine cycles under the condition that the power supply and the oscillator have been stabilized.

ADDED FUNCTIONS OF MSM80C48, MSM80C49 AND MSM80C50

The MSM80C48, MSM80C49 and MSM80C50 basically incorporate the capabilities of Intel's 8048, 8049, and 8050 plus the following new functions:

1. Power-Down Mode Enhancements

1.1 Power-down by software

(1) Clock (See item 4, "Power-down mode", for details.)
   a. Crystal-controlled oscillator halt (HLTS instruction)
      Power requirements can be minimized.
   b. Clock supply halt (HALT instruction)
      Restart is accomplished without oscillator wait.

(2) I/O ports (See Table 4-1 and 4-2 for details.)
   I/O port floating instructions
   Power consumption resulting from inputs/outputs can be minimized with FLT and FLTT instructions.
   Port floating is cancelled by executing FRES instruction, "0" level at $\overline{INT}$ pin or "0" level at $\overline{RESET}$ pin.

(3) Six types of power-down can be done by a combination of HLTS/HALT and FLT/FLTT instructions.

1.2 Power-down by hardware (See 4.3, Power-down mode by V_{DD} pin utilization for details.)

Crystal-controlled oscillators can be halted by controlling the V_{DD} terminal, thereby floating all I/O ports for minimum power consumption.

2. Additional Instructions (11)

HLTS	MOV A, P2
HALT	MOV P1, @ R3
FLT	MOVP1 P, @ R3
FLTT	DEC @ Rr
FRES	DJNZ @ R, addr
MOV A, P1	

3. Improved Uses of BUS $P_{0 \sim 7}$, $P1_{0 \sim 7}$, $P2_{0 \sim 7}$, and $\overline{SS}$ terminals

3.1 BUS $P_{0 \sim 7}$

The MSM80C48, MSM80C49, and MSM80C50 remove the limitation on the use of OUTL BUS, A instructions during the external ROM access mode by having an independent data latch and external ROM mode address latch in BUS $P_{0 \sim 7}$.

Consequently, there is no need to relocate bus port instructions when in the external ROM access mode.

3.2 $P1_{0 \sim 7}$ and $P2_{0 \sim 7}$

The MSM80C48, MSM80C49 and MSM80C50 are designed to minimize power consumption when $P1_{0 \sim 7}$ and $P2_{0 \sim 7}$ are used as input/output ports, to maximize the performance of CMOS.

When these ports are used as output ports, the acceleration circuit is actuated only when output data changes from "0" to "1", thus speeding up the rise time of the output signals.

When these ports are used as input ports, the internal pullup resistance becomes approximately 9 kΩ when input data is "1".

The internal pullup resistance rises to approximately 100 kΩ when input data is "0". Thus, a high noise margin can be obtained by selecting the impedance and thus the outflow of current is minimized whenever these ports are used as output or input ports.

3.3 Clock generation control via the $\overline{SS}$ terminal

When the crystal-controlled oscillator is halted in the HLTS or hardware power-down mode, the $\overline{SS}$ terminal is pulled down by a resistor of 20 – 50 kΩ, while its internal pullup resistor of 200 – 500 kΩ is isolated from V_{CC}. When the power-down mode is cancelled, the internal resistor of the $\overline{SS}$ terminal is changed from pull-down to pullup. Consequently, the CPU can be halted for any period of time until the crystal-controlled oscillator resumes normal oscillation when a capacitor is connected to the $\overline{SS}$ terminal.

4. Power-Down Mode

The MSM80C48, MSM80C49, and MSM80C50 power-down mode can be enabled in 2 different ways-through software by a combination of clock control and port floating instructions, and through hardware by control of the V_{DD} pin.

4.1 Software power-down mode

Power-down mode can be done by a combination of the following instructions.

(1) HALT (clock supply halt to control circuit)

Instruction code:

0	0	0	0	0	0	0	1

Description: Although crystal-controlled oscillator operation is continued, the clock supply to the CPU control circuit is halted and CPU operations suspended. When cancelling this software mode, restart is accomplished without oscillator wait. Timing charts are outlined in Figs. 4-1 and 4-2.

(2) HLTS (oscillation stop)

Instruction code:

1	0	0	0	0	0	1	0

Description: The oscillator operation is halted and CPU operations suspended. In cancelling this power down mode, connecting a capacitor to the $\overline{SS}$ pin enables a reasonable

wait period to be accomplished before normal operation is resumed. [Except in the case of using the $\overline{\text{RESET}}$ pin]
Timing charts are outlined in Figs. 4-3 and 4-4.

(3) FLT (floating $P1_0 \sim 7$, $P2_0 \sim 7$, and $BP_0 \sim 7$)

Instruction code:

1	0	1	0	0	0	1	0

Description:

	Internal ROM mode	External ROM mode
P1	Floating	Floating
P2	Floating	$P2_0 \sim 3$ operation $P2_4 \sim 7$ floating
BP	Floating	Operation

Details of IC pin status as a result of executing the FLT instruction are shown in Table 4-1.

(4) FLTT (floating of all output pins)

Instruction code:

1	1	0	0	0	0	1	0

Description:

	Internal ROM mode	External ROM mode
ALE	Floating	Operation
$\overline{\text{PSEN}}$	Floating	Operation
PROG	Floating	Floating
$\overline{\text{WR}}$	Floating	Floating
$\overline{\text{RD}}$	Floating	Floating
TO OUT	Floating	Floating
P1	Floating	Floating
P2	Floating	$P2_0 \sim 3$ operation $P2_4 \sim 7$ floating
BP	Floating	Operation
XTAL	Operation	Operation

Details of IC pin status as a result of executing the FLTT instruction are shown in Table 4-2.

Example 1: Power-down mode accomplished by stopping oscillation.
- ○ Setting by execution of HLTS [82H] instruction.

Example 2: Power-down mode accomplished by stopping the clock supply to the CPU control circuit.
- ○ Setting by execution of HALT [01H] instruction.

Example 3: Power-down mode by floating of $P1_0 \sim 7$, $P2_0 \sim 7$ and $BP_0 \sim 7$, and subsequent stopping of CPU oscillation.
- ○ Setting by first executing the FLT[A2H] instruction and then the HLTS[82H] instruction.

Example 4: Power-down mode by floating $P1_0 \sim 7$, $P2_0 \sim 7$ and $BP_0 \sim 7$, and then stopping the clock supply to the CPU control circuit.
- ○ Setting by first executing the FLT[A2H] instruction, and then the HALT[01H] instruction.

Example 5: Power-down mode by floating all output pins, followed by stopping oscillation.
- ○ Setting by first executing the FLTT[C2H] instruction followed by execution of the HLTS[82H] instruction.

Example 6: Power-down mode by floating all output pins, followed by stopping of the clock supply to the CPU control circuit.
- ○ Setting by first executing the FLTT[C2H] instruction, followed by execution of the HALT[01H] instruction.

4.2 Cancellation of software power-down mode

The power-down mode status outlined above in examples 1 to 6 can be cancelled by using either the interrupt pin or the $\overline{\text{RESET}}$ pin.

(1) Use of the $\overline{\text{INT}}$ pin during external interrupt enabled mode (i.e. following execution of EN I instruction).
- ○ The clock generator is activated and the CPU started up when a "0" level is applied to the $\overline{\text{INT}}$ pin. If this "0" level is maintained until at least 2 ALE output signals occur, an external interrupt is generated, and execution proceeds from address 3. If, however, the power-down mode has been done during the interrupt processing routine, execution is resumed just after the power-down instruction.

(2) Use of the $\overline{\text{INT}}$ pin during external interrupt disabled mode (i.e. following execution of DIS I instruction or hardware reset)
- ○ The clock generator is activated and the CPU started up when a "0" level is applied to the $\overline{\text{INT}}$ pin. If this "0" level is maintained until at least 2 ALE output signals occur, execution is resumed just after the power-down instruction.

(3) Use of the $\overline{\text{RESET}}$ pin
- ○ The clock generator is activated and the CPU started up when a "0" level is applied to the $\overline{\text{RESET}}$ pin. If this "0" level is maintained until at least 2 ALE output signals occur, the CPU is reset and execution proceeds from address 0. In case cancellation is done in oscillation stop mode, the "0" level must be input to the $\overline{\text{RESET}}$ PIN until oscillation is stabilized.

Table 4-1 Details of Pin Status Following Execution of FLT Instruction

Pin No.	Pin Name	Internal ROM	External ROM
1P	T0	Active	Active
2P	XTAL1	Active	Active
3P	XTAL2	Active	Active
4P	$\overline{\text{RESET}}$	Active	Active
5P	$\overline{\text{SS}}$	200 ~ 500 kΩ pullup	200 ~ 500 kΩ pullup
6P	$\overline{\text{INT}}$	Active	Active
7P	EA	Active	Active
8P	$\overline{\text{RD}}$	Active	Active
9P	$\overline{\text{PSEN}}$	Active	Active
10P	$\overline{\text{WR}}$	Active	Active
11P	ALE	Active	Active
12P	DB0	Floating	Active
13P	DB1	Floating	Active
14P	DB2	Floating	Active
15P	DB3	Floating	Active
16P	DB4	Floating	Active
17P	DB5	Floating	Active
18P	DB6	Floating	Active
19P	DB7	Floating	Active
20P	V_{SS}	0 [V]	0 [V]
21P	P20	Floating	Active
22P	P21	Floating	Active
23P	P22	Floating	Active
24P	P23	Floating	Active
25P	PROG	Active	Active
26P	V_{DD}	"1" level	"1" level
27P	P10	Floating	Floating
28P	P11	Floating	Floating
29P	P12	Floating	Floating
30P	P13	Floating	Floating
31P	P14	Floating	Floating
32P	P15	Floating	Floating
33P	P16	Floating	Floating
34P	P17	Floating	Floating
35P	P24	Floating	Floating
36P	P25	Floating	Floating
37P	P26	Floating	Floating
38P	P27	Floating	Floating
39P	T1	Active	Active
40P	V_{CC}	+2 to +6 [V]	+2 to +6 [V]

Note: The FLT mode itself is reset by executing the FRES instruction, or supplying "0" level to $\overline{\text{INT}}$ or $\overline{\text{RESET}}$ pin.

Table 4-2 Details of Pin Status Following Execution of FLTT Instruction

Pin No.	Pin Name	Internal ROM	External ROM
1P	T0	Floating if output enabled	Floating if output enabled
2P	XTAL1	Active	Active
3P	XTAL2	Active	Active
4P	$\overline{\text{RESET}}$	Active	Active
5P	SS	200 to 500 kΩ pullup	200 to 500 kΩ pullup
6P	$\overline{\text{INT}}$	Active	Active
7P	EA	Active	Active
8P	$\overline{\text{RD}}$	Floating	Floating
9P	$\overline{\text{PSEN}}$	Floating	Active
10P	$\overline{\text{WR}}$	Floating	Floating
11P	ALE	Floating	Active
12P	DB0	Floating	Active
13P	DB1	Floating	Active
14P	DB2	Floating	Active
15P	DB3	Floating	Active
16P	DB4	Floating	Active
17P	DB5	Floating	Active
18P	DB6	Floating	Active
19P	DB7	Floating	Active
20P	V_{SS}	0 [V]	0 [V]
21P	P20	Floating	Active
22P	P21	Floating	Active
23P	P22	Floating	Active
24P	P23	Floating	Active
25P	PROG	Floating	Floating
26P	V_{DD}	"1" level	"1" level
27P	P10	Floating	Floating
28P	P11	Floating	Floating
29P	P12	Floating	Floating
30P	P13	Floating	Floating
31P	P14	Floating	Floating
32P	P15	Floating	Floating
33P	P16	Floating	Floating
34P	P17	Floating	Floating
35P	P24	Floating	Floating
36P	P25	Floating	Floating
37P	P26	Floating	Floating
38P	P27	Floating	Floating
39P	T1	Active	Active
40P	V_{CC}	+2.5 to +6 [V]	+2.5 to +6 [V]

Note: The FLTT mode itself is reset by executing the FRES instruction, or supplying "0" level to $\overline{\text{INT}}$ or $\overline{\text{RESET}}$ pin.

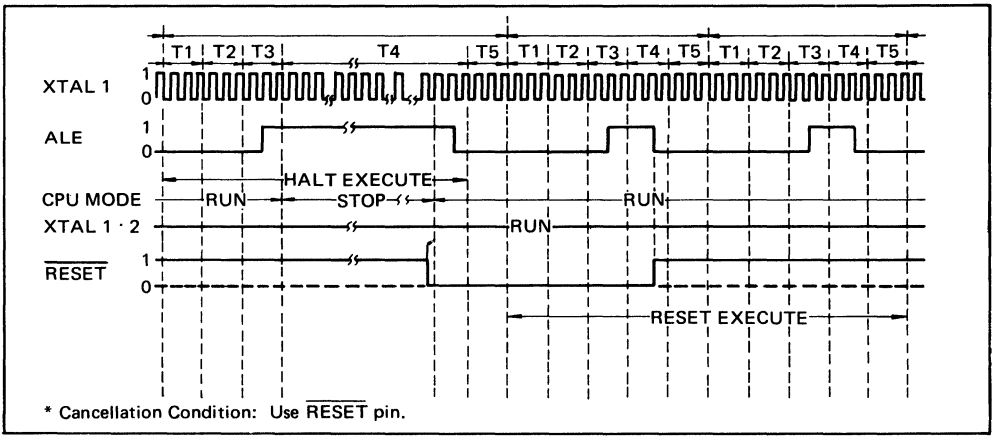


Fig. 4-1 HALT [01H] Instruction Execution Timing Chart

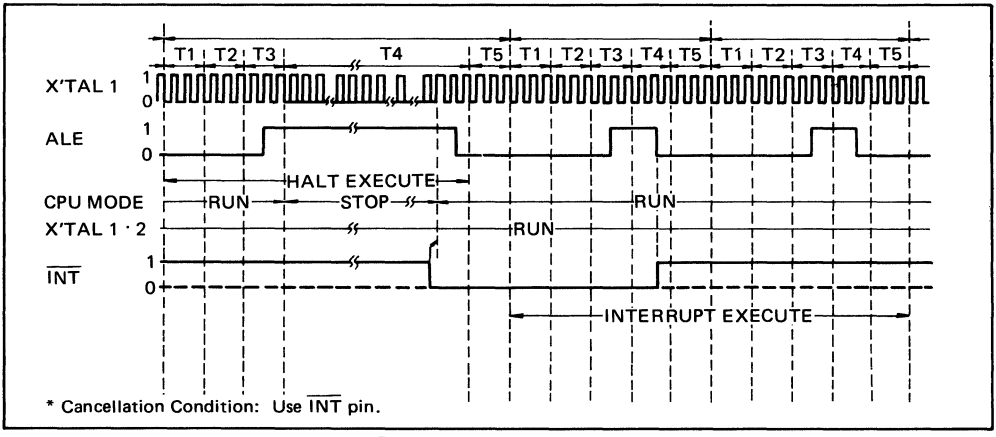


Fig. 4-2 HALT [01H] Instruction Execution Timing Chart

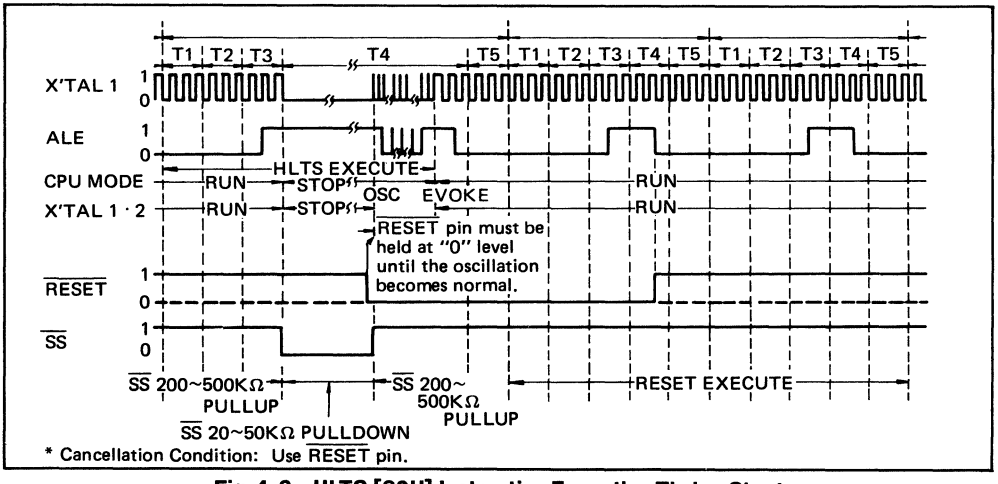


Fig. 4-3 HLTS [82H] Instruction Execution Timing Chart

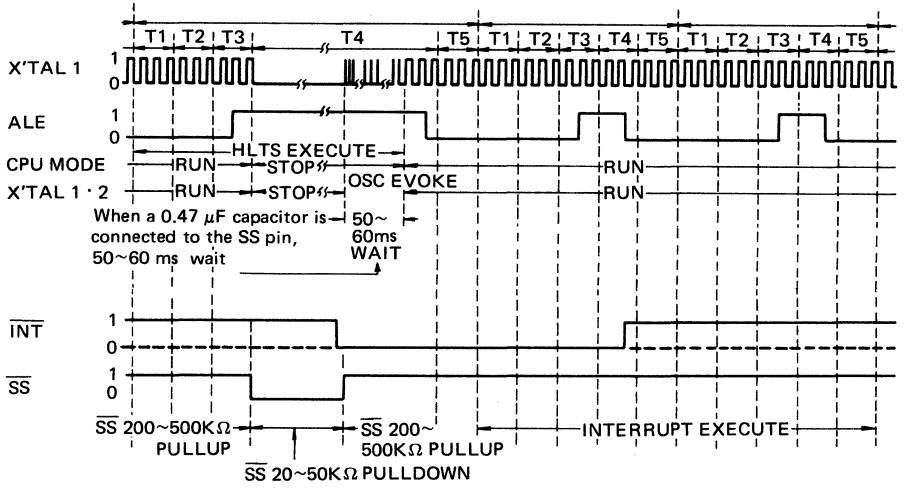


Fig. 4-4 HLTS [82H] Instruction Execution Timing Chart

4.3 Hardware power-down mode

In the MSM80C48, MSM80C49 and MSM80C50, forcing the level at the V_{DD} pin [pin 26] to a "0" during either external ROM or internal ROM mode results in suspension of the oscillator function and subsequent floating (high impedance) of all the I/O pins except the $\overline{RESET}$, $\overline{SS}$ and XTAL 1/2 pins. The CPU is thereby stopped while maintaining internal status. Details of the IC pin status at this time are outlined in Table 4-3.

4.4 Cancellation of hardware power-down mode

(1) Use of $\overline{RESET}$ pin

○ The clock generator is activated and the CPU started up when a "1" level is applied to the V_{DD} pin while a "0" level is input to the $\overline{RESET}$ pin. If this "0" level is kept applied to the $\overline{RESET}$ pin until oscillation become stable, the CPU will be reset and will start executing from address 0. The timing chart is outlined in Fig. 4-5.

(2) Use of the $\overline{INT}$ pin during external interrupt enabled status (i.e. following execution of EN I instruction)

○ The clock generator is activated and the CPU started up when a "1" level is applied to the V_{DD} pin while a "0" level is applied to the $\overline{INT}$ pin.

If this "0" level is maintained until at least 2 ALE output signals occur, an external interrupt is generated, and execution starts from address 3.

However, if the power-down mode is started during an interrupt processing routine, execution will be continued on the next instruction after the present instruction. The timing chart is outlined in Fig. 4-6.

(3) Use of the $\overline{INT}$ pin during external interrupt disabled mode (i.e. following execution of DIS I instruction or hardware reset)

○ The clock generator is activated and the CPU started up when a "1" level is applied to the V_{DD} pin while a "0" level is applied to the $\overline{INT}$ pin. If this "0" level is maintained until at least 2 ALE output signals occur, execution is continued on the next instruction after the present instruction. The timing chart is outlined in Fig. 4-6.

(4) Use of V_{DD} pin only

○ The clock generator is activated and the CPU started up when a "1" level is applied to the V_{DD} pin while a "1" level is also applied to both the $\overline{RESET}$ and $\overline{INT}$ pins. In this case, execution is resumed from the stopped position. The timing chart is outlined in Fig. 4-7.

Table 4-3 Details of Pin Status during Hardware Power-Down Mode

Pin No.	Pin Name	Normal Operation (V_{DD} = "1" level)	Power Down Mode (V_{DD} = "0" level)
1P	T0	Active	Floating if output enabled
2P	XTAL1	Active	Active
3P	XTAL2	Active	Active
4P	$\overline{RESET}$	Active	Active
5P	$\overline{SS}$	200 to 500 kΩ pullup	20 to 50 kΩ pulldown
6P	$\overline{INT}$	Active	Active
7P	EA	Active	Active
8P	$\overline{RD}$	Active	Floating
9P	$\overline{PSEN}$	Active	Floating
10P	$\overline{WR}$	Active	Floating
11P	ALE	Active	Floating
12P	DB0	Active	Floating
13P	DB1	Active	Floating
14P	DB2	Active	Floating
15P	DB3	Active	Floating
16P	DB4	Active	Floating
17P	DB5	Active	Floating
18P	DB6	Active	Floating
19P	DB7	Active	Floating
20P	V_{SS}	0 [V]	0 [V]
21P	P20	Active	Floating
22P	P21	Active	Floating
23P	P22	Active	Floating
24P	P23	Active	Floating
25P	PROG	Active	Floating
26P	V_{DD}	"1" level	"0" level
27P	P10	Active	Floating
28P	P11	Active	Floating
29P	P12	Active	Floating
30P	P13	Active	Floating
31P	P14	Active	Floating
32P	P15	Active	Floating
33P	P16	Active	Floating
34P	P17	Active	Floating
35P	P24	Active	Floating
36P	P25	Active	Floating
37P	P26	Active	Floating
38P	P27	Active	Floating
39P	T1	Active	Active
40P	V_{CC}	+2 to +6 [V]	+2 to +6 [V]

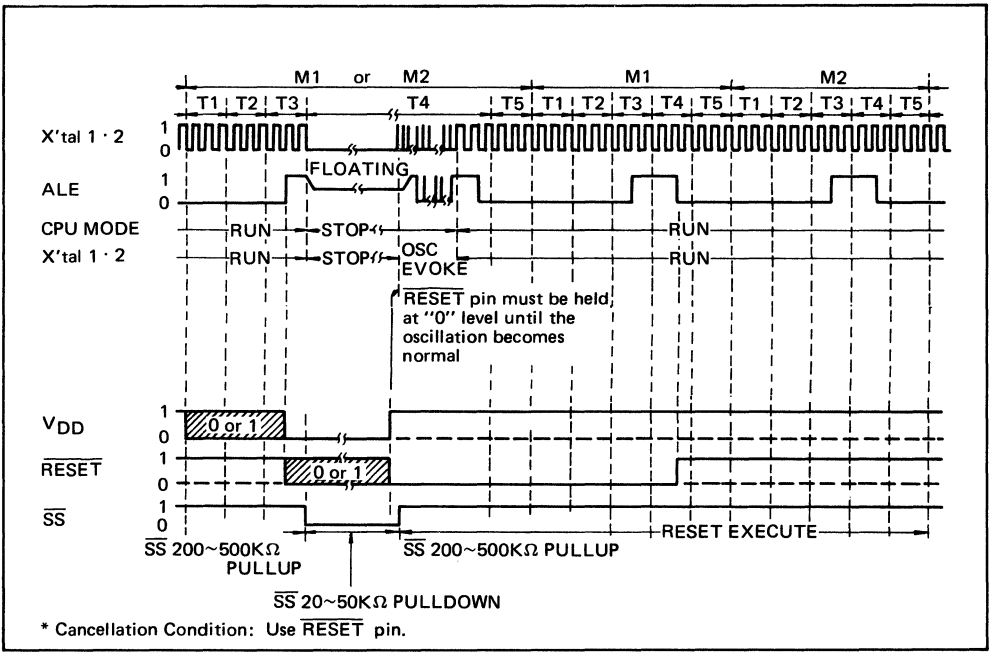


Fig. 4-5 Hardware Power-Down Mode Timing Chart

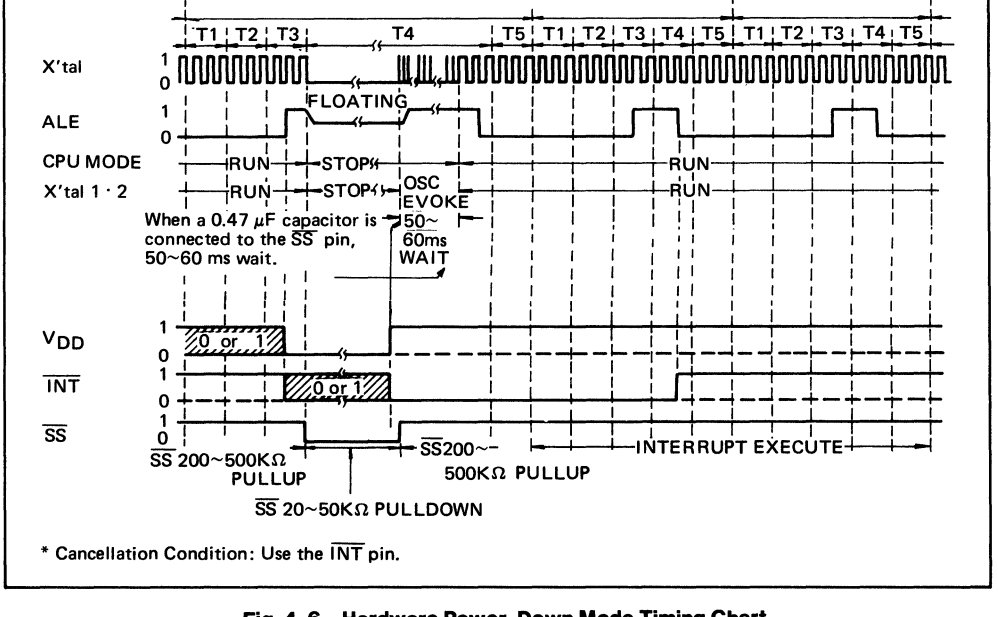


Fig. 4-6 Hardware Power-Down Mode Timing Chart

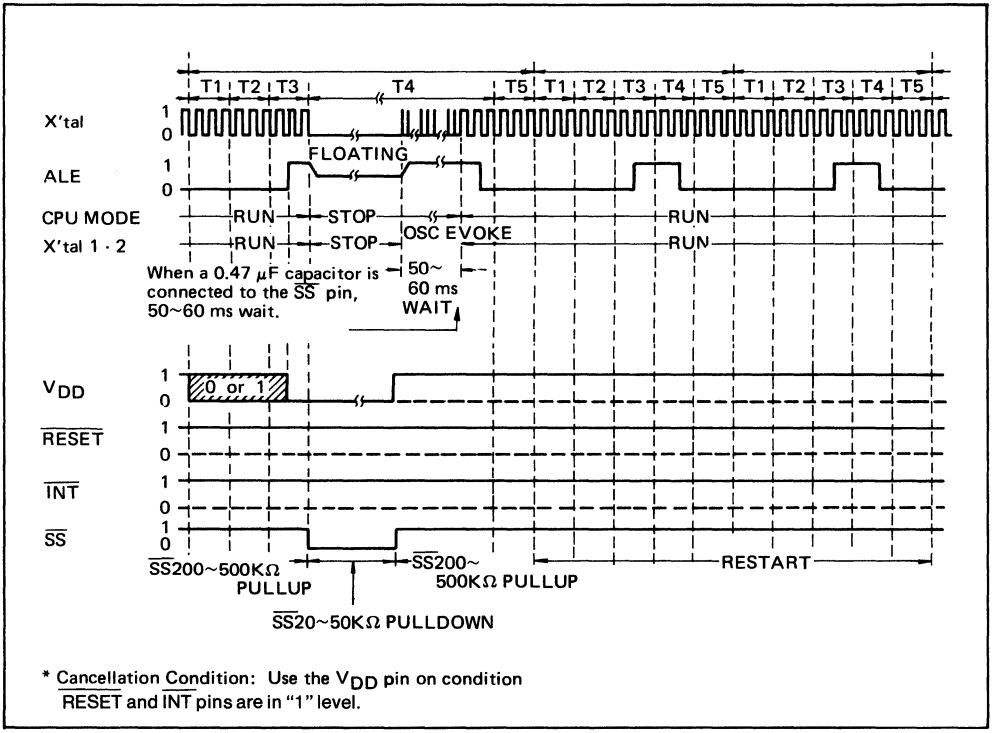


Fig. 4-7 Hardware Power-Down Mode Timing Chart

MSM80C48/MSM80C49/MSM80C50 INSTRUCTION TABLE

H \ L	0 0 0 0 0	1 0 0 0 1	2 0 0 1 0	3 0 0 1 1	4 0 1 0 0	5 0 1 0 1	6 0 1 1 0	7 0 1 1 1	8 1 0 0 0	9 1 0 0 1	A 1 0 1 0	B 1 0 1 1	C 1 1 0 0	D 1 1 0 1	E 1 1 1 0	F 1 1 1 1
0 0 0 0 0	NOP	HALT Added	OUTL BUS, A	ADD A, # data	JMP	EN I		DEC A	INS A, BUS	IN A, P1	INA, P2		MOVD A, Pp			
1 0 0 0 1	INC @, R0	INC @, R1	JB0 addr	ADDC A, # data	CALL	DIS I	JTF addr	INC A	INC Rr							
2 0 0 1 0	XCHA@, R0	XCHA@, R1		MOV A, # data	JMP	EN TCNTI	JNT0 addr	CLR A	XCH A, Rr							
3 0 0 1 1	XCHD A, @R0	XCHD A, @R1	JB1 addr		CALL	DIS TCNTI	JT0 addr	CPL A		OUTL P1, A	OUTL P2, A		MOVD Pp, A			
4 0 1 0 0	ORL A, @R0	ORL A, @R1	MOV A, T	ORL A, # data	JMP	STRT CNT	JNT1 addr	SWAP A	ORLA, Rr							
5 0 1 0 1	ANL A, @R0	ANL A, @R1	JB2 addr	ANL A, # data	CALL	STRT T	JT1 addr	DA A	ANL A, Rr							
6 0 1 1 0	ADD A, @ R0	ADD A, @ R1	MOV T, A	MOV A, P1 Added	JMP	STOP TCNT		RRC A	ADD A, Rr							
7 0 1 1 1	ADDC A, @ R0	ADDC A, @ R1	JB3 addr	MOV A, P2 Added	CALL	ENT0 CLK	JF1 addr	RR A	ADDC A, Rr							
8 1 0 0 0	MOVX A, @ R0	MOVX A, @ R1	HLTS Added	RET	JMP	CLR F0	JN1 addr		ORL BUS, # data	ORL P1, # data	ORL P2, # data		ORLD Pp, A			
9 1 0 0 1	MOVX @ R0, A	MOVX @ R1, A	JB4 addr	RETR	CALL	CLR F0	JNZ addr	CLR C	ANL BUS, # data	ANL P1, # data	ANL P2, # data		ANL D Pp, A			
A 1 0 1 0	MOV @ R0, A	MOV @ R1, A	FLT Added	MOVP A, @A	JMP	CLR F1		CPL C	MOV Rr, A							
B 1 0 1 1	MOV @ R0, #data	MOV @R1, #data	JB5 addr	JMPP @A	CALL	CPL F1	JF0 addr		MOV R, #data							
C 1 1 0 0	DEC @R0 Added	DEC @R1 Added	FLTT Added	MOVP1 P,@ R3 Added	JMP	SEL RB0	JZ addr	MOV A, PSW	DEC Rr							
D 1 1 0 1	XRL A, @R0	XRL A, @R1	JB6 addr	XRLA, #data	CALL	SEL RB1		MOV PSW, A	XRLA, Rr							
E 1 1 1 0	DJNZ @R0 Added	DJNZ @R1 Added	FRES Added	MOVP3 A, @A	JMP	SEL MB0	JNC addr	RL A	DJNZ Rr							
F 1 1 1 1	MOV A, @R0	MOV A, @R1	JB7 addr	MOVP1, @ R3 Added	CALL	SEL MB1	JC addr	RLC A	MOV A, Rr							

EXPLANATION OF INSTRUCTION SYMBOLS

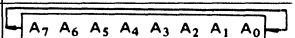
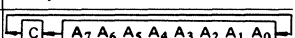
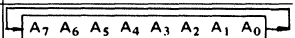
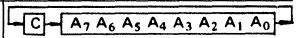
Symbols are listed below.

A	: Accumulator	PC	: Program counter
AC	: Auxiliary carry	Pp	: Port indicator (p = 4 ~ 7)
addr	: 12-bit program memory address or its part	PSW	: Program status word
Bb	: Bit indicator (b = 0 ~ 7)	Rr	: Resister indicator (r = 0 ~ 7)
BS	: Bank switch	SP	: Stack pointer
BUS	: BUS PORT	T	: Timer
C	: Carry	TF	: Timer flag
CLK	: Clock	T0, T1	: Test pins T0 and T1
CNT	: Counter	X	: External RAM
D	: 4-bit data	#	: Symbol denoting immediate data
data	: 8-bit numerical value	@	: Symbol denoting indirect address
DBF	: Memory data bank flip-flop	(X)	: Denotes contents of X
F0, F1	: F0 flag and F1 flag	((X))	: Denotes contents addressed by X
I	: Interrupt	←	: Transference

LIST OF INSTRUCTIONS

Classification	Mnemonic	Instruction Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal	Byte	Cycle	Description
Accumulator operation instructions	ADD A, Rr	0	1	1	0	1	r_2	r_1	r_0	68 ~ 6F	1	1	(AC), (C), (A) ← (A) + (Rr)
	ADD A, @Rr	0	1	1	0	0	0	0	r_0	60~61	1	1	(AC), (C), (A) ← (A) + ((Rr))
	ADD A, #data	0 d_7	0 d_6	0 d_5	0 d_4	0 d_3	0 d_2	1 d_1	1 d_0	03 Byte 2	2	2	(AC), (C), (A) ← (A) + data
	ADDC A, Rr	0	1	1	1	1	r_2	r_1	r_0	78~7F	1	1	(AC), (C), (A) ← (A) + (Rr) + (C)
	ADDC A, @Rr	0	1	1	1	0	0	0	r_0	70~71	1	1	(AC), (C), (A) ← (A) + ((Rr)) + (C)
	ADDC A, #data	0 d_7	0 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	13 Byte 2	2	2	(AC), (C), (A) ← (A) + data + (C)
	ANL A, Rr	0	1	0	1	1	r_2	r_1	r_0	58~5F	1	1	(A) ← (A) AND (Rr)
	ANL A, @Rr	0	1	0	1	0	0	0	r_0	50~51	1	1	(A) ← (A) AND ((Rr))
	ANL A, #data	0 d_7	1 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	53 Byte 2	2	2	(A) ← (A) AND data
	ORL A, Rr	0	1	0	0	1	r_2	r_1	r_0	48~4F	1	1	(A) ← (A) OR (Rr)
	ORL A, @Rr	0	1	0	0	0	0	0	r_0	40~41	1	1	(A) ← (A) OR ((Rr))
	ORL A, #data	0 d_7	1 d_6	0 d_5	0 d_4	0 d_3	0 d_2	1 d_1	1 d_0	43 Byte 2	2	2	(A) ← (A) OR data
	XRLA, Rr	1	1	0	1	1	r_2	r_1	r_0	D8~DF	1	1	(A) ← (A) XOR (Rr)
	XRLA, @Rr	1	1	0	1	0	0	0	r_0	D0~D1	1	1	(A) ← (A) XOR ((Rr))
	XRL A, #data	1 d_7	1 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	D3 Byte 2	2	2	(A) ← (A) XOR data
	INC A	0	0	0	1	0	1	1	1	17	1	1	(A) ← (A) + 1
	DEC A	0	0	0	0	0	1	1	1	07	1	1	(A) ← (A) − 1
	CLR A	0	0	1	0	0	1	1	1	27	1	1	(A) ← 0
	CPL A	0	0	1	1	0	1	1	1	37	1	1	(A) ← ($\bar{A}$)
	DA A	0	1	0	1	0	1	1	1	57	1	1	Add 6 to bits 0 ~ 3 when contents of accumulator bits 0 ~ 3 exceed 9 or when auxiliary carry (AC) is 1. Then add 6 to bits 4 ~ 7 when the result of adding the carry from the lower 0 ~ 3 exceeds 9, or when carry (C) is 1. Set 1 in the carry flag if an overflow is generated in the end result, or when the carry prior to adjustment is 1.

LIST OF INSTRUCTIONS (CONT.)

Classification	Mnemonic	Instruction Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal	Byte	Cycle	Description
Accumulator operation instructions	SWAP A	0	1	0	0	0	1	1	1	47	1	1	$(A_{4\sim7}) \rightleftarrows (A_{0\sim3})$
	RL A	1	1	1	0	0	1	1	1	E7	1	1	 Rotate accumulator contents to the left by 1 bit.
	RLC A	1	1	1	1	0	1	1	1	F7	1	1	 Rotate accumulator contents with carry to the left by 1 bit.
	RR A	0	1	1	1	0	1	1	1	77	1	1	 Rotate accumulator contents to the right by 1 bit.
	RRC A	0	1	1	0	0	1	1	1	67	1	1	 Rotate accumulator contents with carry to the right by 1 bit.
Input/output instructions	IN A, P1	0	0	0	0	1	0	0	1	09	1	2	(A) ← (P1)
	IN A, P2	0	0	0	0	1	0	1	0	0A	1	2	(A) ← (P2)
	OUTL P1, A	0	0	1	1	1	0	0	1	39	1	2	(P1) ← (A)
	OUTL P2, A	0	0	1	1	1	0	1	0	3A	1	2	(P2) ← (A)
	ANL P1, #data	1 d_7	0 d_6	0 d_5	1 d_4	1 d_3	0 d_2	0 d_1	1 d_0	99 Byte 2	2	2	(P1) ← (P1) AND data
	ANL P2, #data	1 d_7	0 d_6	0 d_5	1 d_4	1 d_3	0 d_2	1 d_1	0 d_0	9A Byte 2	2	2	(P2) ← (P2) AND data
	ORL P1, #data	1 d_7	0 d_6	0 d_5	0 d_4	1 d_3	0 d_2	0 d_1	1 d_0	89 Byte 2	2	2	(P1) ← (P1) OR data
	ORL P2, #data	1 d_7	0 d_6	0 d_5	0 d_4	1 d_3	0 d_2	1 d_1	0 d_0	8A Byte 2	2	2	(P2) ← (P2) OR data
	INS A, BUS	0	0	0	0	1	0	0	0	08	1	2	(A) ← (BUS)
	OUTL BUS, A	0	0	0	0	0	0	1	0	02	1	2	(BUS) ← (A)
	ANL BUS, #data	1 d_7	0 d_6	0 d_5	1 d_4	1 d_3	0 d_2	0 d_1	0 d_0	98 Byte 2	2	2	(BUS) ← (BUS) AND data
	ORL BUS, #data	1 d_7	0 d_6	0 d_5	0 d_4	1 d_3	0 d_2	0 d_1	0 d_0	88 Byte 2	2	2	(BUS) ← (BUS) OR data
	MOVD A, Pp	0	0	0	0	1	1	P_1	P_0	0C~0F	1	2	$(A_{0\sim3}) \leftarrow$ (Pp) p=4~7 $(A_{4\sim7}) \leftarrow 0$
	MOVD Pp, A	0	0	1	1	1	1	P_1	P_0	3C~3F	1	2	(Pp) $\leftarrow (A_{0\sim3})$ p=4~7
	ANLD Pp, A	1	0	0	1	1	1	P_1	P_0	9C~9F	1	2	(Pp) ← (Pp) AND $(A_{0\sim3})$ p=4~7
	ORLD Pp, A	1	0	0	0	1	1	P_1	P_0	8C~8F	1	2	(Pp) ← (Pp) OR $(A_{0\sim3})$ p=4~7
Register operation instructions	INC Rr	0	0	0	1	1	r_2	r_1	r_0	18~1F	1	1	(Rr) ← (Rr) + 1
	INC @Rr	0	0	0	1	0	0	0	r_0	10~11	1	1	((Rr)) ← ((Rr)) + 1
	DEC Rr	1	1	0	0	1	r_2	r_1	r_0	C8~CF	1	1	(Rr) ← (Rr) − 1
	DEC @Rr	1	1	0	0	0	0	0	r_0	C0~C1	1	1	((Rr)) ← ((Rr)) − 1
Branching instructions	JMP addr	a_{10} a_7	a_9 a_6	a_8 a_5	0 a_4	0 a_3	1 a_2	0 a_1	0 a_0	ϕ4 ~ E4 Byte2	2	2	$(PC_{8\sim10})$ ← addr 8~10 $(PC_{0\sim7})$ ← addr 0~7 (PC_{11}) ← DBF
	JMPP @A	1	0	1	1	0	0	1	1	B3	1	2	$(PC_{0\sim7}) \leftarrow$ ((A))

LIST OF INSTRUCTIONS (CONT.)

Classification	Mnemonic	Instruction Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal	Byte	Cycle	Description
Branching instructions	DJNZ Rr, addr	1	1	1	0	1	r_2	r_1	r_0	E8~EF	2	2	(Rr) ←(Rr) − 1 ($PC_{0\sim7}$) ←addr if (Rr) = 0 (PC) ←(PC) + 2 if (Rr) = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	DJNZ @Rr, addr	1	1	1	0	0	0	0	r_0	E0~E1	2	2	((Rr)) ←((Rr)) − 1 ($PC_{0\sim7}$) ←addr if ((Rr)) = 0 (PC) ←(PC) + 2 if ((Rr)) = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JC addr	1	1	1	1	0	1	1	0	F6	2	2	($PC_{0\sim7}$) ←addr if C = 1 (PC) ←(PC) + 2 if C = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JNC addr	1	1	1	0	0	1	1	0	E6	2	2	($PC_{0\sim7}$) ←addr if C = 0 (PC) ←(PC) + 2 if C = 1
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JZ addr	1	1	0	0	0	1	1	0	C6	2	2	($PC_{0\sim7}$) ←addr if A = 0 (PC) ←(PC) + 2 if A ≠ 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JNZ addr	1	0	0	1	0	1	1	0	96	2	2	($PC_{0\sim7}$) ←addr if A ≠ 0 (PC) ←(PC) + 2 if A = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JT0 addr	0	0	1	1	0	1	1	0	36	2	2	($PC_{0\sim7}$) ←addr if T0 = 1 (PC) ←(PC) + 2 if T0 = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JNT0 addr	0	0	1	0	0	1	1	0	26	2	2	($PC_{0\sim7}$) ←addr if T0 = 0 (PC) ←(PC) + 2 if T0 = 1
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JT1 addr	0	1	0	1	0	1	1	0	56	2	2	($PC_{0\sim7}$) ←addr if T1 = 1 (PC) ←(PC) + 2 if T1 = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JNT1 addr	0	1	0	0	0	1	1	0	46	2	2	($PC_{0\sim7}$) ←addr if T1 = 0 (PC) ←(PC) + 2 if T1 = 1
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JF0 addr	1	0	1	1	0	1	1	0	B6	2	2	($PC_{0\sim7}$) ←addr if F0 = 1 (PC) ←(PC) + 2 if F0 = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JF1 addr	0	1	1	1	0	1	1	0	76	2	2	($PC_{0\sim7}$) ←addr if F1 = 1 (PC) ←(PC) + 2 if F1 = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JTF addr	0	0	0	1	0	1	1	0	16	2	2	($PC_{0\sim7}$) ←addr TF ←0 } if TF = 1 (PC) ←(PC) + 2 if TF = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JNI addr	1	0	0	0	0	1	1	0	86	2	2	($PC_{0\sim7}$) ←addr if $\overline{INT}$ = 0 (PC) ←(PC) + 2 if $\overline{INT}$ = 1
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	JBb addr	b_2	b_1	b_0	1	0	0	1	0	12 ~ F2	2	2	($PC_{0\sim7}$) ←addr if Bb = 1 (PC) ←(PC) + 2 if Bb = 0
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
Sub-routine instructions	CALL addr	a_{10}	a_9	a_8	1	0	1	0	0	14 ~ F4	2	2	((SP)) ←(PC) + 2, ($PSW_{4\sim7}$) ($PC_{8\sim10}$) ←addr 8 ~ 10 ($PC_{0\sim7}$) ←addr 0 ~ 7 (PC_{11}) ←DBF (SP) ←(SP) + 1
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	RET	1	0	0	0	0	0	1	1	83	1	2	(SP) ←(SP) − 1 (PC) ←((SP))
	RETR	1	0	0	1	0	0	1	1	93	1	2	(SP) ←(SP) − 1 (PC) ←((SP)) ($PSW_{4\sim7}$) ←((SP)) INT END
Flag operation instructions	CLR C	1	0	0	1	0	1	1	1	97	1	1	(C) ←0
	CPL C	1	0	1	0	0	1	1	1	A7	1	1	(C) ←$\overline{(C)}$
	CLR F0	1	0	0	0	0	1	0	1	85	1	1	(F0) ←0
	CPL F0	1	0	0	1	0	1	0	1	95	1	1	(F0) ←$\overline{(F0)}$
	CLR F1	1	0	1	0	0	1	0	1	A5	1	1	(F1) ←0
	CPL F1	1	0	1	1	0	1	0	1	B5	1	1	(F1) ←$\overline{(F1)}$
Data transfer instructions	MOV A, Rr	1	1	1	1	1	r_2	r_1	r_0	F8~FF	1	1	(A) ←(Rr)
	MOV A, @Rr	1	1	1	1	0	0	0	r_0	F0~F1	1	1	(A) ←((Rr))
	MOV A, #data	0	0	1	0	0	0	1	1	23	2	2	(A) ←data
		d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0	Byte 2			

LIST OF INSTRUCTIONS (CONT.)

Classification	Mnemonic	Instruction Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal	Byte	Cycle	Description
Data transfer instructions	MOV Rr, A	1	0	1	0	1	r_2	r_1	r_0	A8~AF	1	1	(Rr) ←(A)
	MOV @Rr, A	1	0	1	0	0	0	0	r_0	A0~A1	1	1	((Rr)) ←(A)
	MOV Rr, #data	1 d_7	0 d_6	1 d_5	1 d_4	1 d_3	r_2 d_2	r_1 d_1	r_0 d_0	B8~BF Byte 2	2	2	(Rr) ←data
	MOV @Rr, #data	1 d_7	0 d_6	1 d_5	1 d_4	0 d_3	0 d_2	0 d_1	r_0 d_0	B0~B1 Byte 2	2	2	((Rr)) ←data
	MOV A, PSW	1	1	0	0	0	1	1	1	C7	1	1	(A) ←(PSW)
	MOV PSW, A	1	1	0	1	0	1	1	1	D7	1	1	(PSW) ←(A)
	XCH A, Rr	0	0	1	0	1	r_2	r_1	r_0	28~2F	1	1	(A) ⇆(Rr)
	XCH A, @Rr	0	0	1	0	0	0	0	r_0	20~21	1	1	(A) ⇆((Rr))
	XCHD A, @Rr	0	0	1	1	0	0	0	r_0	30~31	1	1	$(A_{0\sim3})\leftrightarrows((Rr_{0\sim3}))$
	MOVX A, @Rr	1	0	0	0	0	0	0	r_0	80~81	1	2	(A) ←((Rr)) External RAM
	MOVX @Rr, A	1	0	0	1	0	0	0	r_0	90~91	1	2	((Rr)) ←(A) External RAM
	MOVP A, @A	1	0	1	0	0	0	1	1	A3	1	2	$(A) \leftarrow((PC_{8\sim10}, A))$
	MOVP3 A, @A	1	1	1	0	0	0	1	1	E3	1	2	$(A) \leftarrow((PC_{011}, A))$
	MOVP1 P, @R3	1	1	0	0	0	0	1	1	C3	1	2	$(P1) \leftarrow(((PC_{0\sim7})\leftarrow((R3))))$
	MOV P1, @R3	1	1	1	1	0	0	1	1	F3	1	2	(P1) ←((R3))
	MOV A, P1	0	1	1	0	0	0	1	1	63	1	1	(A) ←(P1) Latch data
	MOV A, P2	0	1	1	1	0	0	1	1	73	1	1	(A) ←(P2) Latch data
Control instructions	EN TCNTI	0	0	1	0	0	1	0	1	25	1	1	TINT Enable F/F ←1
	DIS TCNTI	0	0	1	1	0	1	0	1	35	1	1	TINT Enable F/F ←0
	EN I	0	0	0	0	0	1	0	1	05	1	1	EXINT Enable F/F ←1
	DIS I	0	0	0	1	0	1	0	1	15	1	1	EXINT Enable F/F ←0
	SEL RB0	1	1	0	0	0	1	0	1	C5	1	1	(BS) ←0
	SEL RB1	1	1	0	1	0	1	0	1	D5	1	1	(BS) ←1
	SEL MB0	1	1	1	0	0	1	0	1	E5	1	1	(DBF) ←0
	SEL MB1	1	1	1	1	0	1	0	1	F5	1	1	(DBF) ←1
	ENT0 CLK	0	1	1	1	0	1	0	1	75	1	1	T0 ←1/3 XTAL 1
	FLT	1	0	1	0	0	0	1	0	A2	1	1	P1, P2, BUS Floating
	FLTT	1	1	0	0	0	0	1	0	C2	1	1	CPU Output Signal Floating
	FRES	1	1	1	0	0	0	1	0	E2	1	1	FLT, FLTT RESET
	HLT	0	0	0	0	0	0	0	1	01	1	1	CPU Control Clock Stop
	HALTS	1	0	0	0	0	0	1	0	82	1	1	XTAL 1·2 Stop
Timer/counter instructions	MOV A, T	0	1	0	0	0	0	1	0	42	1	1	(A) ←(T)
	MOV T, A	0	1	1	0	0	0	1	0	62	1	1	(T) ←(A)
	STRT T	0	1	0	1	0	1	0	1	55	1	1	(T) ←÷32←÷15←XTAL
	STRT CNT	0	1	0	0	0	1	0	1	45	1	1	(T) ←T1 Clock
	STOP TCNT	0	1	1	0	0	1	0	1	65	1	1	(T) Count Stop
Other instruction	NOP	0	0	0	0	0	0	0	0	00	1	1	(PC) ←(PC) + 1

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{CC}	Ta = 25°C	−0.3 to 7	V
Input Voltage	V_I	Ta = 25°C	−0.3 to V_{CC}	V
Storage Temperature	T_{stg}		−55 to +150	°C

OPERATING RANGE

- MSM80C35/48, 80C39/49 ... DC to 11 MHz, VCC = 5V±10%
- MSM80C40/50 ... DC to 6 MHz, VCC = 5V±20%

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{CC}	fosc = DC~11 MHz*	+2.5 to +6	V
RAM Retention Voltage	V_{CC}		+2 to +6	V
Ambient Temperature	T_A		−40 to +85	°C
Fan Out	N	MOS load	10	
		TTL load	1	

* 11 MHz version of MSM80C40/50 (6 MHz < XTAL1.2 < 11 MHz) is under development.

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 5V±10%, Ta = −40 to +85°C)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit	Measuring Circuit
"L" Input Voltage	V_{IL}		−0.5		0.13V_{CC}	V	1
"H" Input Voltage (1)	V_{IH}		0.4V_{CC}		V_{CC}	V	
"H" Input Voltage (2)	V_{IH}		0.7V_{CC}		V_{CC}	V	
"L" Output Voltage (3)	V_{OL}	I_{OL} = 2 mA			0.45	V	
"L" Output Voltage (4)	V_{OL}	I_{OL} = 1.6 mA			0.45	V	
"H" Output Voltage (3)	V_{OH}	I_{OH} = −400 μA	0.75V_{CC}			V	
"H" Output Voltage (4)	V_{OH}	I_{OH} = −50 μA	0.75V_{CC}			V	
"H" Output Voltage (3)	V_{OH}	I_{OH} = −20 μA	0.93V_{CC}			V	
"H" Output Voltage (4)	V_{OH}	I_{OH} = −10 μA	0.93V_{CC}			V	
Input Leak Current	I_{IL}	$V_{SS} \leqq V_{IN} \leqq V_{CC}$			±5	μA	2
Output Leak Current (5)	I_{OL}	$V_{SS} \leqq V_O \leqq V_{CC}$			±5	μA	3
$\overline{\text{RESET}}$ Input current	I_R	$V_{IN} = V_{IH}$	−20	−50	−80	μA	2
		$V_{IN} = V_{IL}$	−3	−8	−15	μA	
$\overline{SS}$ Input current (6)	I_{SS}	Pull-up ($V_{IN}=V_{IL}$)	−4	−15	−25	μA	
		Pull-down ($V_{IN}=V_{IH}$)	20	70	125	μA	
P1, P2 Input current	I_{P1}, I_{P2}	$V_{IN} = V_{IH}$	−300	−600	−900	μA	2
		$V_{IN} = V_{IL}$	−10	−40	−80	μA	
Power Down Mode Standby Current	I_{CCS}	At hardware power down (7) Ta = 25°C, V_{CC} = 2.0V	—	—	10	μA	4
		At HLTS execution (7) Ta = 25°C, V_{CC} = 2.0V	—	—	10		
Power Supply Current (Halt Mode)	I_{CC}	V_{CC} = 4.5V, f = 1 MHz	—	—	0.8	mA	
		V_{CC} = 4.5V, f = 6 MHz	—	—	1.5		
		V_{CC} = 4.5V, f = 11 MHz (8)	—	—	2.5		
		V_{CC} = 5V, f = 1 MHz	—	—	1.0		
		V_{CC} = 5V, f = 6 MHz	—	—	2.0		
		V_{CC} = 5V, f = 11 MHz (8)	—	—	3.0		
		V_{CC} = 5.5V, f = 1 MHz	—	—	1.3		
		V_{CC} = 5.5V, f = 6 MHz	—	—	2.5		
		V_{CC} = 5.5V, f = 11 MHz (8)	—	—	4.0		
Power Suppy Current	I_{CC}	V_{CC} = 4.5V, f = 1 MHz	—	—	2.0	mA	
		V_{CC} = 4.5V, f = 6 MHz	—	—	6.5		
		V_{CC} = 4.5V, f = 11 MHz (8)	—	—	13		
		V_{CC} = 5V, f = 1 MHz	—	—	2.5		
		V_{CC} = 5V, f = 6 MHz	—	—	7.5		
		V_{CC} = 5V, f = 11 MHz (8)	—	—	15		
		V_{CC} = 5.5V, f = 1 MHz	—	—	4.0		
		V_{CC} = 5.5V, f = 6 MHz	—	—	9.0		
		V_{CC} = 5.5V, f = 11 MHz (8)	—	—	18		

Notes: (1) This does not apply to $\overline{\text{RESET}}$, XTAL1, XTAL2, and V_{DD}.
(2) $\overline{\text{RESET}}$, XTAL1, XTAL2, V_{DD}
(3) BUS, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, ALE
(4) Other outputs
(5) High-impedance state
(6) This operates as a pull-down resistor when the oscillation is stopped in the HLTS or hardware power-down mode and as a pull-up resistor in other states.
(7) This does not contain flow out current from I/O Ports and Signal pins.
(8) MSM80C35/48, 80C39/49

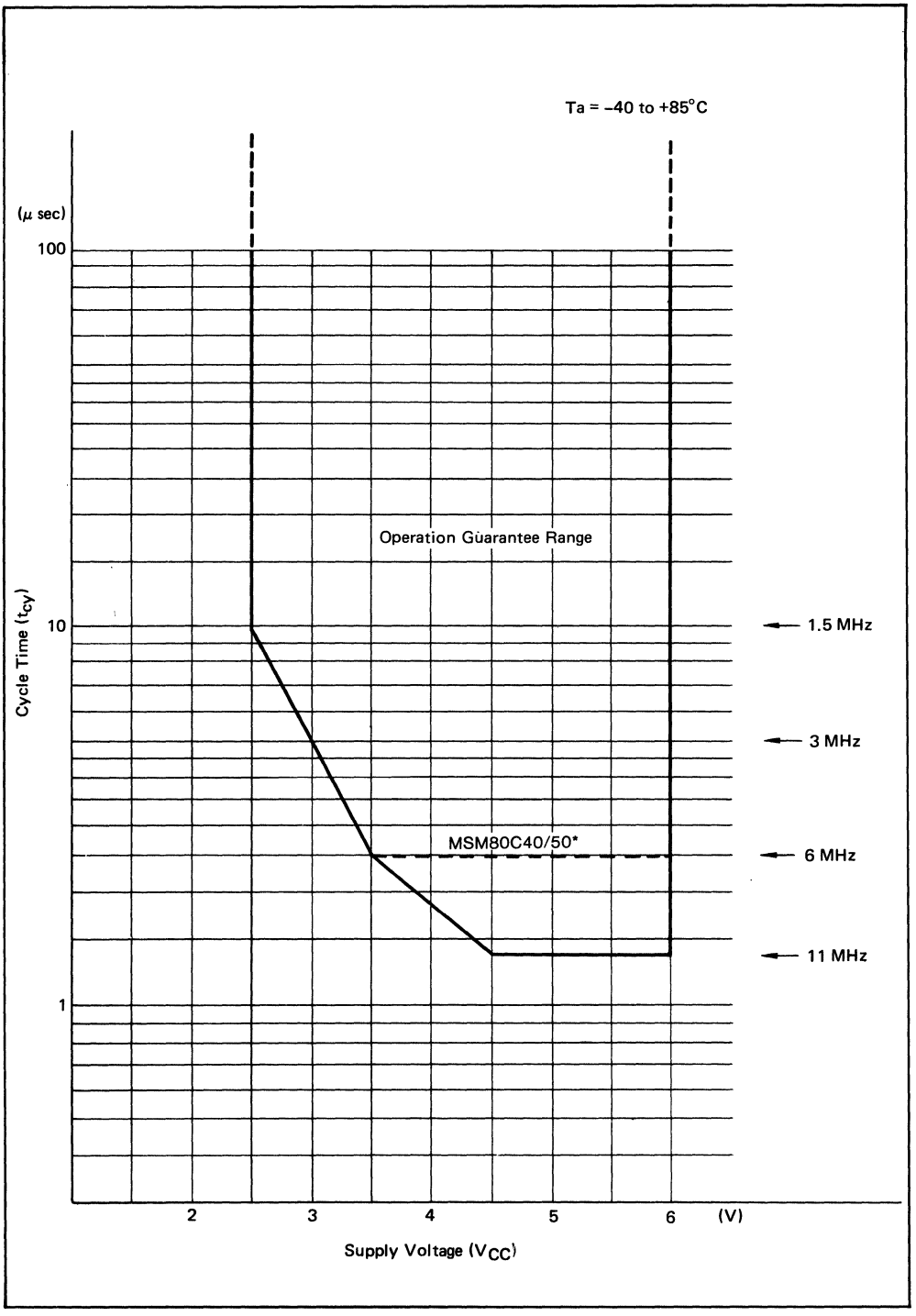
AC CHARACTERISTICS

(V_{CC} = 5V±10%, TA = −40°C to +85°C)

Parameter	Symbol	Limits				Unit
		11 MHz Clock		Variable Clock (0 – 11 MHz)		
		Min.	Max.	Min.	Max.	
Cycle Time	t_{CY}	1.36		1.36		μS
ALE Pulse Width	t_{LL}	150		$7/30t_{CY}-165$		ns
Address Set up ALE	t_{AL}	70		$2/15t_{CY}-110$		ns
Address Hold from ALE	t_{LA}	50		$1/15t_{CY}-40$		ns
Bus Port Latch Data Setup to ALE	t_{BL}	110		$5/30t_{CY}-115$		ns
Bus Port Latch Data Hold from ALE	t_{LB}	90		$3/30t_{CY}-45$		ns
Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, and $\overline{WR}$)	t_{CC}	300		$6/15t_{CY}-245$		ns
Data Setup before $\overline{WR}$	t_{DW}	250		$6/15t_{CY}-295$		ns
Data Hold after $\overline{WR}$	t_{WD}	40		$2/15t_{CY}-140$		ns
Data Hold after $\overline{RD}$	t_{DR}	0	100	0	100	ns
$\overline{PSEN}$, $\overline{RD}$ to Data-in	t_{RD}		200		$5/15t_{CY}-250$	ns
Address Setup to $\overline{WR}$	t_{AW}	200		$6/15t_{CY}-345$		ns
Address Setup to Data-in	t_{AD}		400		$8/15t_{CY}-325$	ns
Address Float to $\overline{RD}$, $\overline{PSEN}$	t_{AFC}	0		0		ns
Port Control Setup to $\overline{PROG}$	t_{CP}	100		$2/15t_{CY}-80$		ns
Port Control Hold from $\overline{PROG}$	t_{PC}	60		$4/15t_{CY}-300$		ns
$\overline{PROG}$ to P2 Input Valid	t_{PR}	–	650		$9/15t_{CY}-165$	ns
Output Data Setup	t_{DP}	200		$6/15t_{CY}-345$		ns
Output Data Hold	t_{PD}	20		$3/15t_{CY}-250$		ns
Input Data Hold from $\overline{PROG}$	t_{PF}	0	150	0	150	ns
$\overline{PROG}$ Pulse Width	t_{PP}	700		$10/15t_{CY}-205$		ns
Port 2 I/O Setup to ALE	t_{PL}	150		$9/30t_{CY}-255$		ns
Port 2 I/O Hold from ALE	t_{LP}	20		$3/30t_{CY}-115$		ns

Note: Control output: C_L = 80 pF
Bus output: C_L = 150 pF [for 20 pF (t_{WD})]

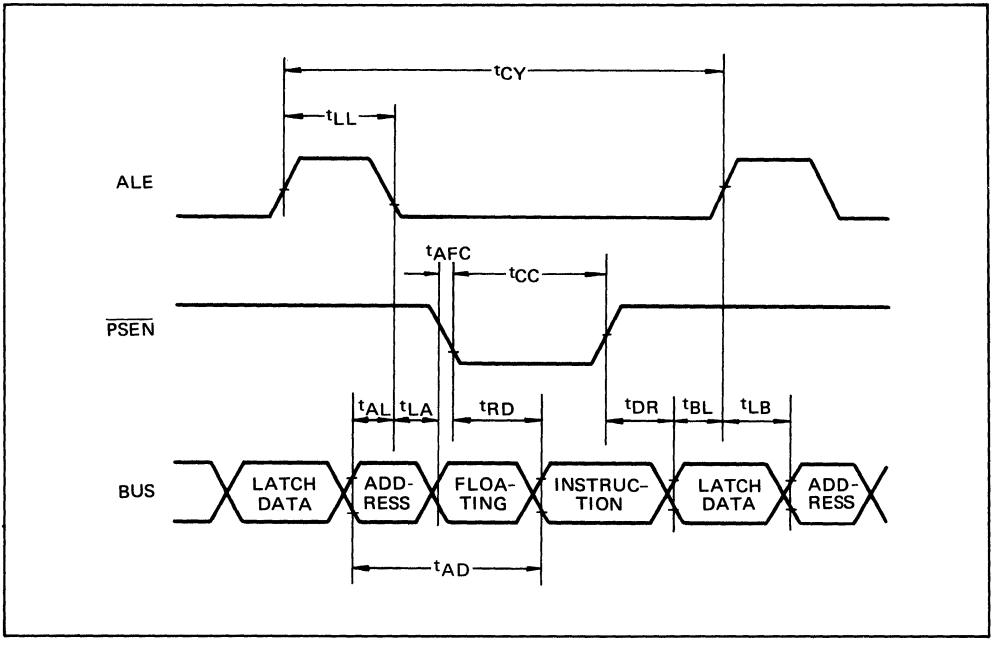
MSM80C49 OPERATION GUARANTEE RANGE



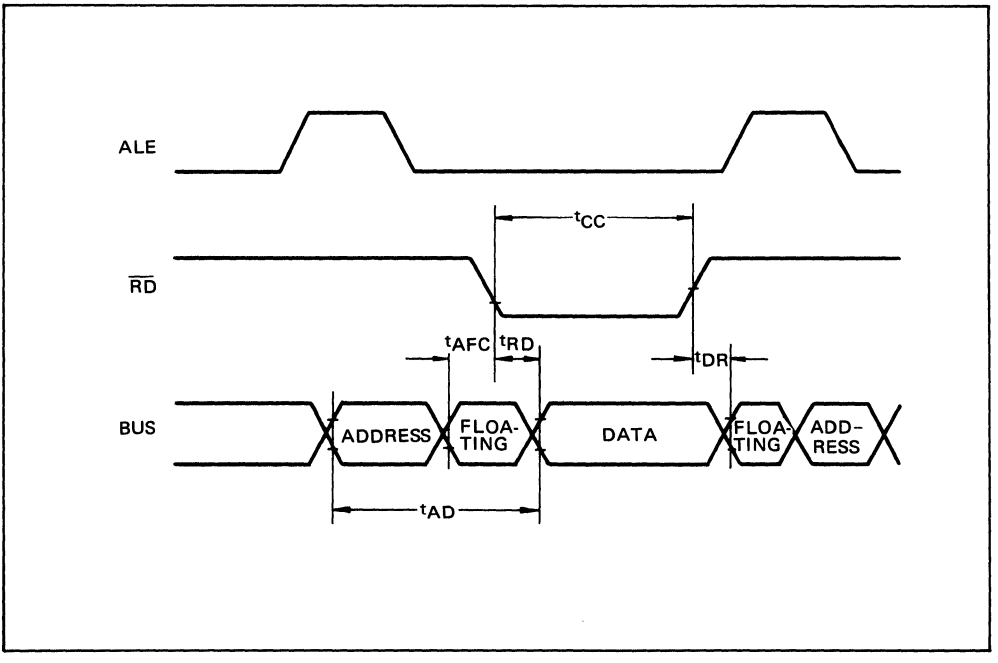
* 11 MHz version of MSM80C40/50 is under development

TIMING CHART

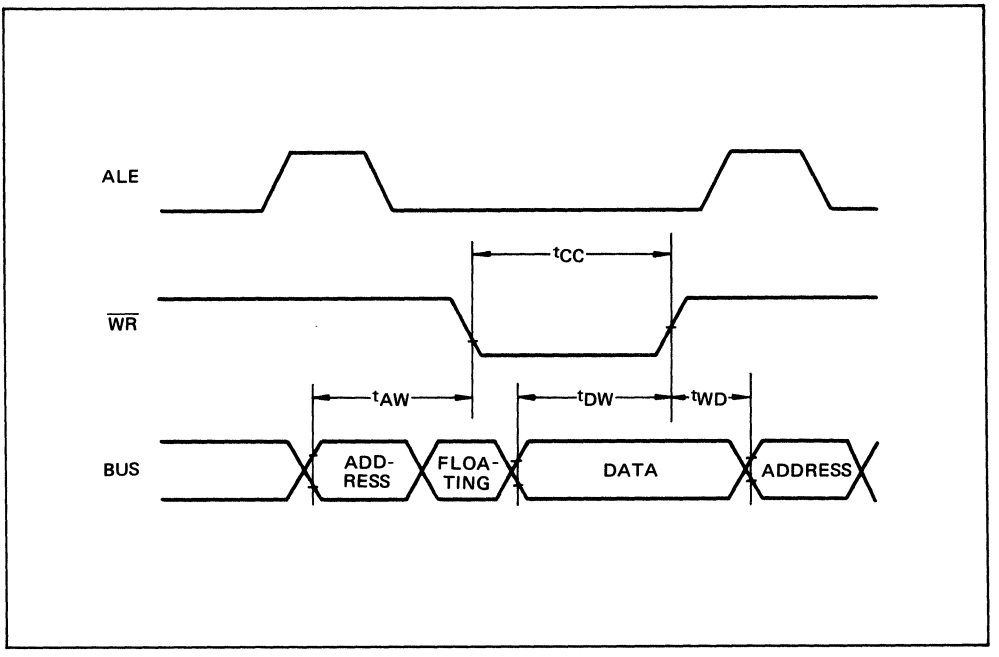
Instruction Fetch (from external program memory)



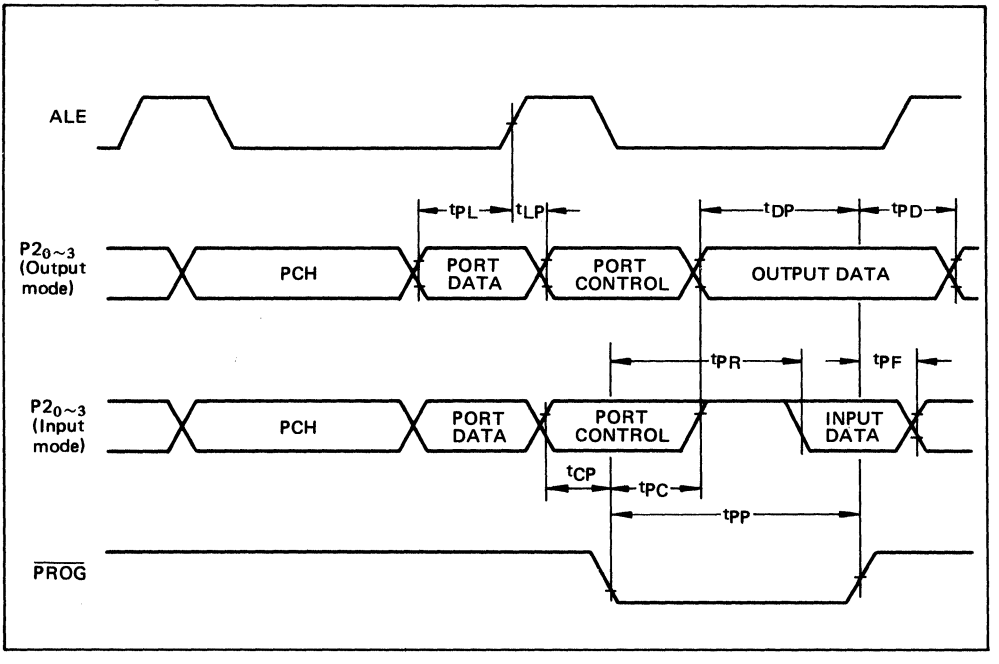
Read (from external data memory)



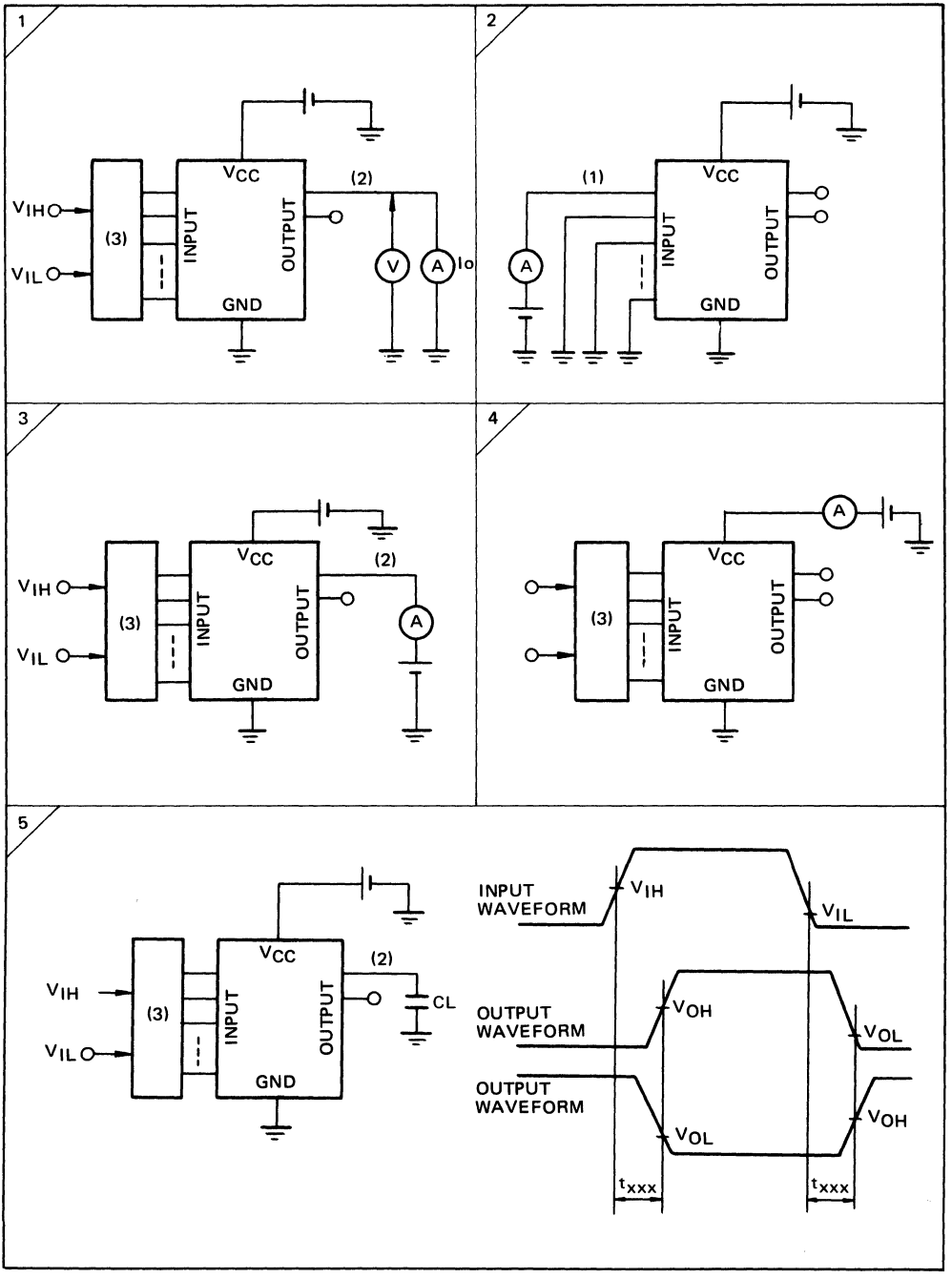
Write (to external memory)



4 low-order bits input/output of port 2 when expanded I/O is used
(in external program memory access mode)



MEASUREMENT CIRCUIT



Notes: (1) This is repeated for each specified input pin.
(2) This is repeated for each specified output pin.
(3) Input logic for setting the specified state.

OKI semiconductor

MSM80C31F/MSM80C51F

CMOS SINGLE-COMPONENT 8-BIT MICROCONTROLLER

GENERAL DESCRIPTION

The OKI MSM80C31F/MSM80C51F microcontroller is a low power, high performance 8-bit single component device implemented in OKI's silicon gate complementary metal oxide semiconductor process technology. Integrated within the device is 4K bytes of mask programmable ROM (MSM80C51F only), 128 bytes of data RAM, 32 I/O lines, two 16-bit timer/counters, a five-source two-level interrupt structure, a full duplex serial port, and an on chip oscillator and clock circuitry. In addition, the device has two software selectable modes for further power reduction – Idle and Power Down. Idle mode freezes the CPU's instruction execution while maintaining RAM and allowing the timers, serial port and interrupt system to continue functioning. Power Down mode saves the RAM contents but freezes the oscillator causing all other device functions to be inoperative.

FEATURES

- Operating temperature:
  $-40 \sim +85°C$ (@ 12 MHz, $V_{CC} = 5V \pm 20\%$)
  $-20 \sim +70°C$ (@ 16 MHz, $V_{CC} = 5V \pm 5\%$)
- Operating frequency: 0.5 ~ 16 MHz
- CMOS technology, 2μm Silicon gate
- Minimum instruction cycle:
  1.0μs (@ 12MHz, Vcc = 5V±20%)
  0.75μs (@ 16MHz, $V_{CC} = 5V \pm 5\%$)
- Low power consumption:

Normal Operation	16 mA @ 5V, 12MHz
Idle Mode	3.7 mA @ 5V, 12MHz
Power Down Mode	50μA @ 2V

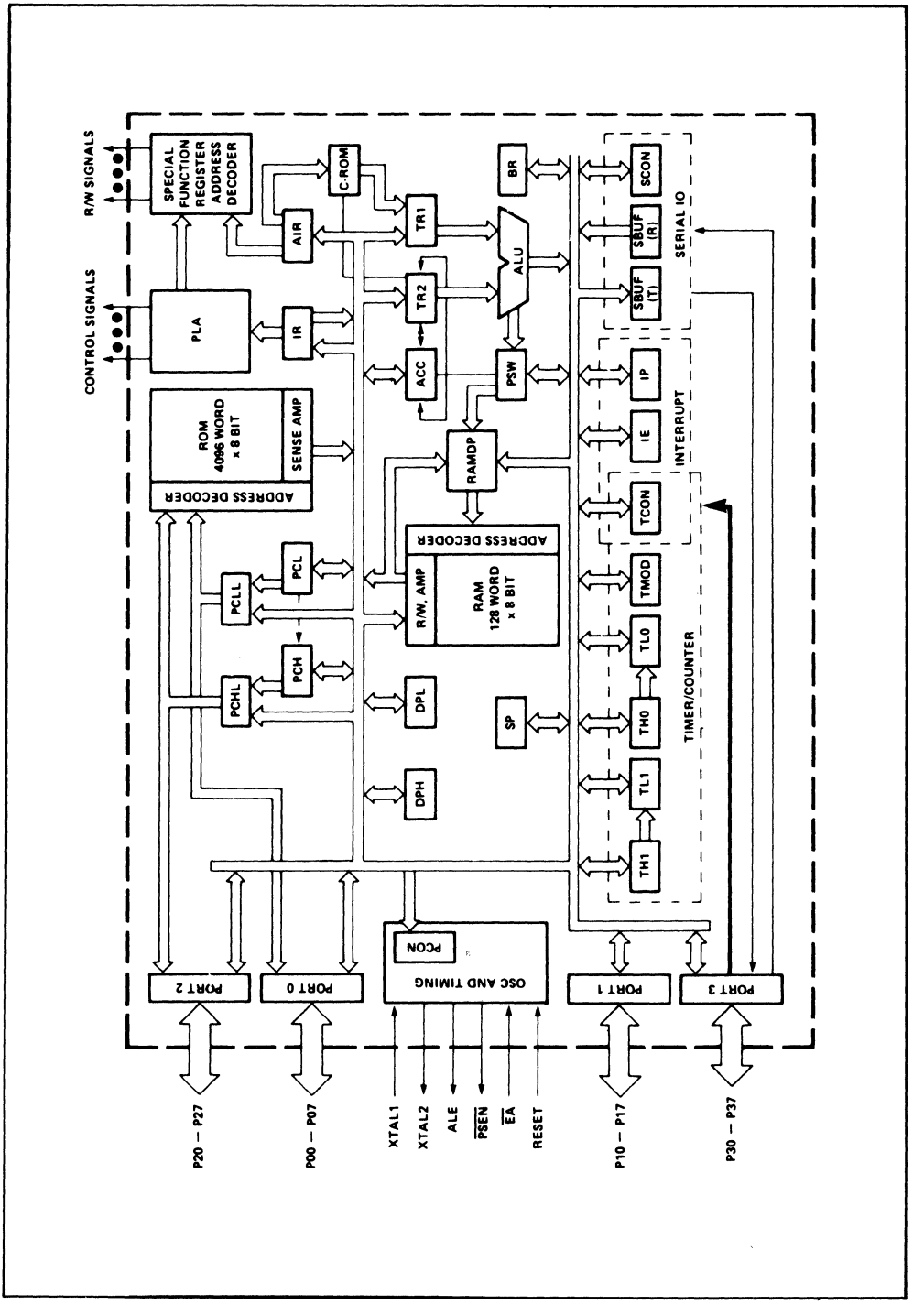
- Instruction set includes 111 instructions
- 8-bit CPU
- On chip oscillator and clock circuitry
- 32 Input/Output lines
- 4069 × 8 bits on chip ROM (MSM80C51F)
- 128 × 8 bits on chip RAM
- 64K address space for program memory
- 64K address space for external data memory
- Two 16-bit timer/counters
- Five source two-priority level interrupt structure
- Full duplex serial port
- Boolean processor
- CMOS and TTL compatible
- CMOS ROM LESS development device (MSM80C31F)
- Package:
  40 pin plastic DIP (DIP40-P-600)
  44 pin plastic QFP (QFP44-P-910-K)
  44 pin plastic QFP (QFP44-P-910-VIK)
  44 pin PLCC QFP (QFJ44-P-S650)

DIFFERENCES BETWEEN MSM80C31F/MSM80C51F AND MSM80C31/MSM80C51

- Operating frequency
  0.5 ~ 16 MHz MSM80C31F-1/MSM80C51F-1
  0.5 ~ 12 MHz MSM80C31/MSM80C51/MSM80C31F/MSM80C51F
- External clock input terminal
  XTAL1 ..... MSM80C31F(-1)/MSM80C51F(-1)
  XTAL2 ..... MSM80C31/MSM80C51
- Emulation mode
  Output impedance of ALE and $\overline{PSEN}$ pins becomes about 20kΩ while CPU is being reset in MSM80C31F/MSM80C51F.

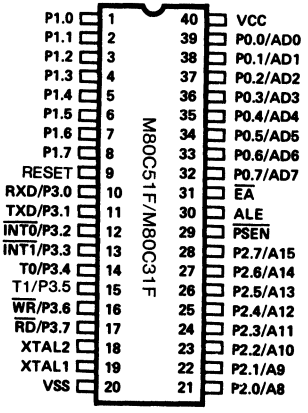
Any other functions and electrical characteristics of MSM80C31F/MSM80C51F except for the above three differences are the same as those of MSM80C31/MSM80C51.

FUNCTIONAL BLOCK DIAGRAM

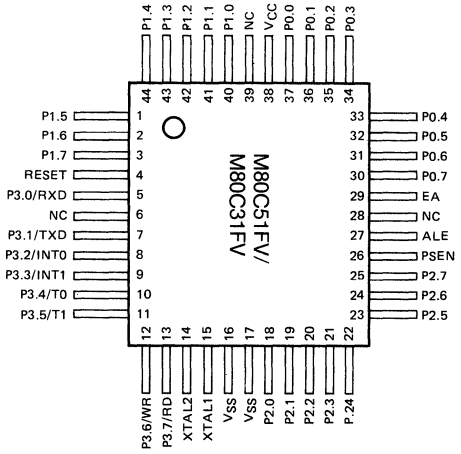


PIN CONFIGURATION

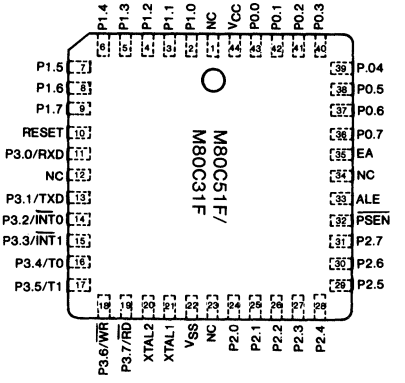
MSM80C51F-XXRS/MSM80C31FRS
(Top View) 40-Lead Plastic DIP



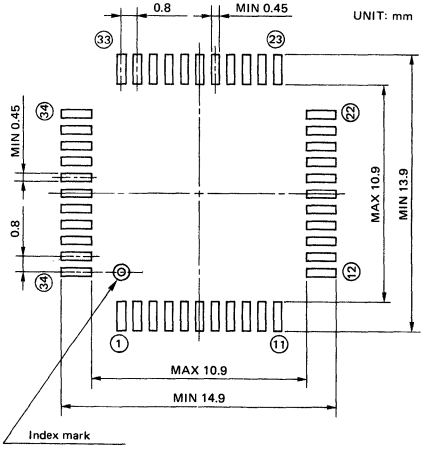
MSM80C51FV-XXGSK/MSM80C31FVGSK
(Top View) 44-Lead Plastic Flat Package



MSM80C51F-XXJS/MSM80C31FJS
(Top View) 44 Lead Plastic Leaded Chip Carrier



RECOMMENDED FOOTPRINT FOR QFD PACKAGE



PIN DESCRIPTIONS

Designation	Description
V_{SS}	Ground potential
V_{CC}	Supply voltage during Normal, Idle and Power Down operation
Port 0	Port 0 is an 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus during accesses to external memory.
Port 1	Port 1 is an 8-bit bidirectional I/O port with internal pullups. It can drive CMOS inputs without external pullups.
Port 2	Port 2 is an 8-bit bidirectional I/O port with internal pullups. It Outputs the high-order address byte during accesses to external memory. It can drive CMOS inputs without external pullups.
Port 3	Port 3 is an 8-bit bidirectional I/O port with internal pullups. It also serves the functions of various special features, as shown below: **Port Pin** **Alternate Fucntion** P3.0 RXD (serial input port) P3.1 TXD (serial output port) P3.2 $\overline{INT0}$ (external interrupt) P3.3 $\overline{INT1}$ (external interrupt) P3.4 T0 (Timer 0 external input) P3.5 T1 (Timer 1 external input) P3.6 $\overline{WR}$ (external data memory write strobe) P3.7 $\overline{RD}$ (external data memory read strobe) It can drive CMOS inputs without external pullups.
RESET	Reset input pin. A reset is accomplished by holding the RESET pin high for at least 1 μ second, even if the oscillator has been stopped. The CPU responds by executing an internal reset. An internal pulldown resistor permits Power-On reset using only a capacitor connected to V_{CC}. This pin does not receive the power down voltage since the function has been transferred to the V_{CC} pin.
ALE	Address Latch Enable output for latching the low byte of the address during accesses to external memory. For this purpose, ALE is activated twice every machine cycle or at a constant rate of 1/6 the oscillator frequency, except during an external memory access at which time one ALE pulse is skipped. It can drive CMOS inputs without an external pullup.
$\overline{PSEN}$	Program Store Enable output is the read strobe to external Program Memory. $\overline{PSEN}$ is activated twice each machine cycle during fetches from external Program Memory. (However, when executing out of external Program Memory, two activations of $\overline{PSEN}$ are skipped during each access to external Data Memory.) $\overline{PSEN}$ is not activated during fetches from internal Program Memory. It can drive CMOS inputs without an external pullup.
$\overline{EA}$	External Access input pin. When $\overline{EA}$ is held high, the CPU executes out of internal Program Memory (unless the Program Counter exceeds 0FFFH). When $\overline{EA}$ is held low, the CPU executes only out of external Program Memory. $\overline{EA}$ must not be floated.
XTAL1	Crystal 1 pin. It is an input to the inverting amplifier which forms the internal oscillator. It also receives the external clock signal when an external oscillator is used. (External clock signal should be at 50% duty and C-MOS level).
XTAL2	Crystal 2 pin. It is an output of the inverting amplifier that forms the internal oscillator.

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS*

Item	Symbol	Conditions	Rating	Units
Supply Voltage	V_{CC}		−0.5 to +7.0	V
Voltage from any Pin to V_{SS}	VAP		−0.5 to V_{CC} +0.5	V
Storage Temperature	TSTG		−55 to +150	°C
Power Dissipation	PD		1.0	W

OPERATING RANGE

- MSM80C31F-1/80C51F-1 ... 0.5 to 16 MHz, VCC = ±5%
- MSM80C31F/80C51F, MSM80C31/80C51 ... 0.5 to 12 MHz, VCC = ±20%

Item	Symbol	Conditions	Rating	Units
Supply Voltage	V_{CC}	** fosc = 0.5 ~ 16 MHz	2.5 to +6.0	V
RAM Retention Voltage	VPD	Power Down	2.0 to +6.0	V
Operating Temperature	TOP		−40 to +85	°C

***NOTICE:** Stresses at or above those listed under 'Absolute Maximum Ratings' may cause permanent damage to the device. This is a stress rating only and functional operation of this device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

** DC and AC characteristics in the range of 12 MHz < f ≤ 16 MHz and 2.5 V ≤ Vcc < 4 V will be specified elsewhere.

DC CHARACTERISTICS

(TA = −40°C to +85°; V_{CC} = 4 to 6V: V_{SS} = 0V)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Input Low Voltage (Except EA)	V_{IL}	–	−0.5	–	$0.2V_{CC}-0.1$	V
Input Low Voltage To EA	V_{IL1}	–	−0.5	–	$0.2V_{CC}-0.3$	V
Input High Voltage (Except XTAL1, RESET)	V_{IH}	–	$0.2V_{CC}+0.9$	–	$V_{CC}+0.5$	V
Input High Voltage (XTAL1 and RESET)	V_{IH}	–	$0.7V_{CC}$	–	$V_{CC}+0.5$	V
Power Down Voltage to V_{CC} in PD mode	V_{PD}	–	2.0	–	6.0	V
Output Low Voltage Ports 1, 2, 3 (Note 1)	V_{OL}	I_{OL} = 1.6mA	–	–	0.45	V
Output Low Voltage Port 0, ALE, $\overline{PSEN}$ (Note 1)	V_{OL1}	I_{OL} = 3.2mA	–	–	0.45	V
Output High Voltage Ports 1, 2, 3	V_{OH}	I_{OH} = −10μA	$0.9V_{CC}$	–	–	V
		I_{OH} = −30μA	$0.75V_{CC}$	–	–	V
		I_{OH} = −60μA	2.4	–	–	V

NOTE 1: V_{OL} is degraded when the 80C31F/80C51F rapidly discharges external capacitance. This AC noise is most pronounced during the emission of address data. When using external memory, locate the latch or buffer as close to the 80C31F/80C51F as possible.

DC CHARACTERISTICS (CONT.)

(TA = -40°C to +85°C, V_{CC} = 4 to 6V, V_{CC} = 0V, fosc = 0.5 to 12 MHz
TA = -20°C to +70°C, V_{CC} = 4.75 to 5.25V, V_{CC} = 0V, fosc = 0.5 to 16 MHz)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Output High Voltage Port 0(in External Bus Mode), ALE, PSEN	V_{OH1}	$I_{OH} = -40\mu A$ (Note 2)	$0.9V_{CC}$	–	–	V
		$I_{OH} = -150\mu A$	$0.75V_{CC}$	–	–	V
		$I_{OH} = -400\mu A$	2.4	–	–	V
Logical 0 Input Current/ Logical 1 Output Current Ports 1, 2, 3	I_{IL}/I_{OH}	V_{in}=0.45V / V_{out}=0.45V	−10	–	−200	μA
Logical 1 To 0 Transition Current-Ports 1, 2, 3	I_{TL}	$V_{in} = 2.0V$	–	–	−500	μA
Input Leakage Current Port 0, EA	I_{LI}	$V_{SS} < V_{in} < V_{CC}$	–	–	±10	μA
Maximum Power Supply Current Normal Operation (Note 3)	I_{CC}	V_{CC} =	4V	5V	6V	
		f_{osc} = 12 MHz	12	16	20	mA
		f_{osc} = 8 MHz	8.3	11	14	mA
		f_{osc} = 3.5 MHz	4.3	5.7	7.5	mA
		f_{osc} = 0.5 MHz	1.6	2.2	3	mA
Maximum Power Supply Current Idle Mode (Note 4)	I_{CC1}	f_{osc} = 12 MHz	2.5	3.7	5	mA
		f_{osc} = 8 MHz	1.8	2.7	3.7	mA
		f_{osc} = 3.5 MHz	1.1	1.6	2.2	mA
		f_{osc} = 0.5 MHz	0.6	0.9	1.2	mA
Maximum Power Supply Current Normal Operation (Note 3)	I_{CC}	V_{CC} =	4.5V	5V	5.5V	
		f_{osc} = 16 MHz	18 (Note 6)	20	23 (Note 6)	mA
		f_{osc} = 12 MHz	14	16	18	mA
		f_{osc} = 8 MHz	10	11	12.5	mA
		f_{osc} = 1.2 MHz	2.0	2.3	2.6	mA
Maximum Power Supply Current Idle Mode (Note 4)	I_{CC1}	f_{osc} = 16 MHz	4.0 (Note 6)	5.0	6.0 (Note 6)	mA
		f_{osc} = 12 MHz	3.0	3.7	5.0	mA
		f_{osc} = 8 MHz	2.3	2.7	3.2	mA
		f_{osc} = 1.2 MHz	1.4	1.5	1.6	mA
Power Supply Current (Power Down Mode)	I_{PD}	V_{CC} = 2.0 ~ 6.0V (Note 5)	–	–	50	μA
RESET Pulldown Resistor	R_{RST}		20	40	125	kΩ
Capacitance of I/O Buffer	C_{IO}	f_C = 1 MHz, T_A = 25°C	–	–	10	pF

NOTE 2: Capacitive loading on Ports 0 and 2 may cause the V_{OH} on ALE and PSEN to momentarily fall below the 0.9 V_{CC} specification when the address bits are stabilizing.

NOTE 3: I_{CC} is measured with all output pins disconnected; XTAL1 driven with TCLCH, TCHCL = 10 ns, $V_{IL} = V_{SS} + 0.5V$, $V_{IH} = V_{CC} - 0.5V$; $\overline{EA}$ = RESET = PORT 0 = V_{CC}. I_{CC} may be higher if a crystal oscillator is used.

NOTE 4: Idle I_{CC} is measured with all output pins disconnected; XTAL1 driven with TCLCH, TCHCL = 10 ns, $V_{IL} = V_{SS} + 0.5V$, $V_{IH} = V_{CC} - 0.5V$; $\overline{EA}$ = PORT 0 = V_{CC}. I_{CC} may be higher if a crystal oscillator is used.

NOTE 5: Power Down I_{PD} is measured with all output pins disconnected; $\overline{EA}$ = PORT 0 = V_{CC}; XTAL2 N.C.; RESET = V_{SS}.

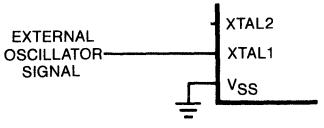
NOTE 6: @ $V_{CC} = 5V \pm 5\%$

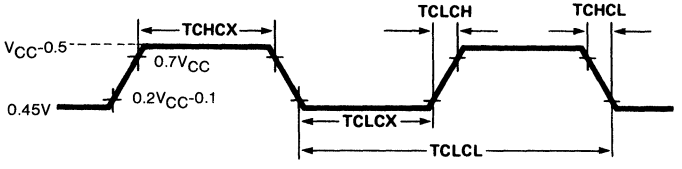
Datum	Emitting Ports	Degraded I/O Lines	V_{OL} (peak/max)
Address	P2, P0	P1, P3	0.8V
Write Data	P0	P1, P2, P3	0.8V

CLOCK PARAMETERS

Item	Symbol	Variable Clock freq = 0.5 MHz to 16 MHz		
		Min.	Max.	Unit
Oscillator Period	T_{CLCL}	62.5	–	ns
High Time	T_{CHCX}	20	–	ns
Low Time	T_{CLCX}	20	–	ns
Rise Time	T_{CLCH}		20	ns
Fall Time	T_{CHCL}	–	20	ns

EXTERNAL CLOCK DRIVE





*XTAL1
Input to the inverting oscillator amplifier and input to the internal clock generator circuits.

*XTAL2
Output from the inverting oscillator amplifier.

AC CHARACTERISTICS

(T_{OP} = −40°C to 85°C; V_{CC} = 4V to 6V; V_{SS} =0V; 0.5 to 12 MHz; Load Capacitance for Port 0, ALE and $\overline{PSEN}$ = 100pF; Load Capacitance for all other outputs = 80 pF)

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

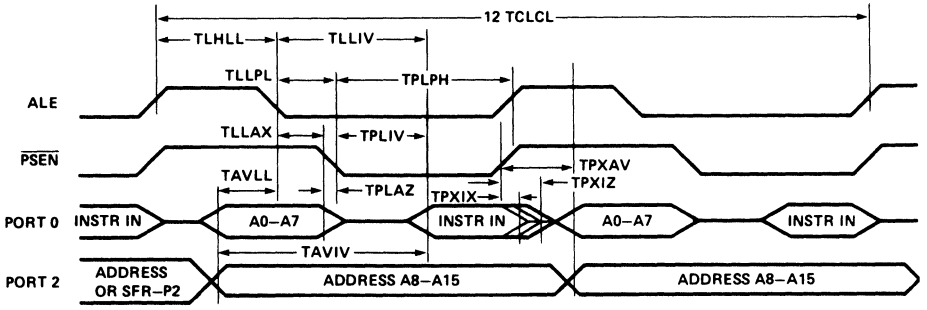
Item	Symbol	Min.	Typ.	Max.	Unit
ALE Pulse Width	TLHLL	2TCLCL-40	–	–	ns
Address Valid to ALE Low	TAVLL	TCLCL-40	–	–	ns
Address Hold after ALE Low	TLLAX	TCLCL-35	–	–	ns
ALE Low to Valid Instr. In	TLLIV	–	–	4TCLCL-100	ns
ALE Low to $\overline{PSEN}$ Low	TLLPL	TCLCL-25	–	–	ns
$\overline{PSEN}$ Pulse Width	TPLPH	3TCLCL-35	–	–	ns
$\overline{PSEN}$ Low to Valid Instr. In	TPLIV	–	–	3TCLCL-105	ns
Input Instr. Hold after $\overline{PSEN}$	TPXIX	0	–	–	ns
Input Instr. Float after $\overline{PSEN}$	TPXIZ	–	–	TCLCL-20	ns
$\overline{PSEN}$ to Address Valid	TPXAV	TCLCL-8	–	–	ns
Address to Valid Instr. In	TAVIV	–	–	5TCLCL-105	ns
$\overline{PSEN}$ Low to Address Float	TPLAZ	–	–	0	ns

EXTERNAL DATA MEMORY CHARACTERISTICS

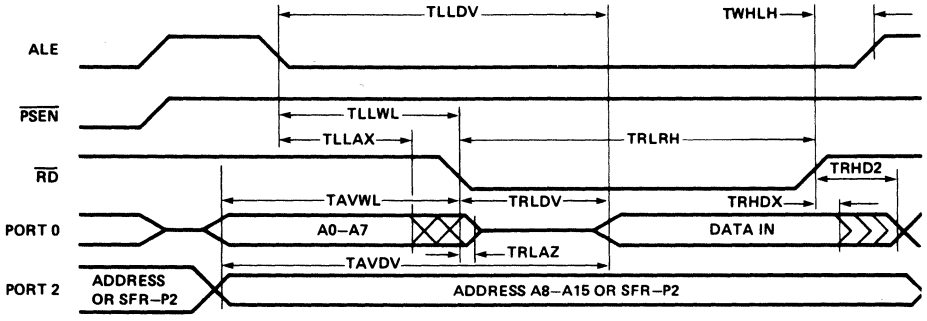
Item	Symbol	Min.	Typ.	Max.	Unit
$\overline{RD}$ Pulse Width	TRLRH	6TCLCL-100	–	–	ns
$\overline{WR}$ Pulse Width	TWLWH	6TCLCL-100	–	–	ns
Data Address Hold after ALE Low	TLLAX	TCLCL-35	–	–	ns
$\overline{RD}$ Low to Valid Data In	TRLDV	–	–	5TCLCL-165	ns
Data Hold after $\overline{RD}$	TRHDX	0	–	–	ns
Data Float after $\overline{RD}$	TRHDZ	–	–	2TCLCL-70	ns
ALE Low to Valid Data In	TLLDV	–	–	8TCLCL-150	ns
Address to Valid Data In	TAVDV	–	–	9TCLCL-165	ns
ALE Low to $\overline{WR}$ or $\overline{RD}$ Low	TLLWL	3TCLCL-50	–	3TCLCL+50	ns
Address to WR or RD Low	TAVWL	4TCLCL-130	–	–	ns
Data Valid to $\overline{WR}$ Transition	TQVWX	TCLCL-60	–	–	ns
Data Valid to $\overline{WR}$ High	TQVWH	7TCLCL-150	–	–	ns
Data Hold after $\overline{WR}$	TWHQX	TCLCL-50	–	–	ns
$\overline{RD}$ Low to Address Float	TRLAZ	–	–	0	ns
$\overline{RD}$ or $\overline{WR}$ High to ALE High	TWHLH	TCLCL-40	–	TCLCL+40	ns

AC TIMING DIAGRAMS

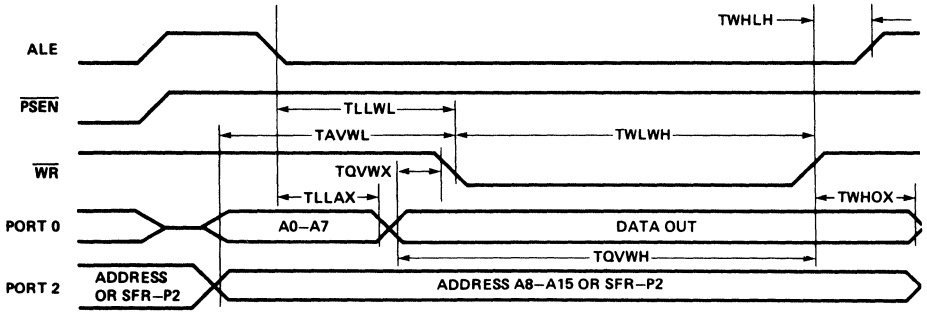
EXTERNAL PROGRAM MEMORY READ CYCLE



EXTERNAL DATA MEMORY READ CYCLE

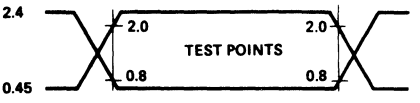


EXTERNAL DATA MEMORY WRITE CYCCLE

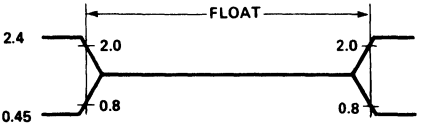


AC TESTING INPUT/OUTPUT, FLOAT WAVEFORMS

INPUT/OUTPUT



FLOAT



AC inputs during testing are driven at 2.4V for a logic "1" and 0.45V for a logic "0". Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0". For timing purposes, the float state is defined as the point at which a P0 pin sinks 3.2 mA or sources 400μA at the voltage test levels.

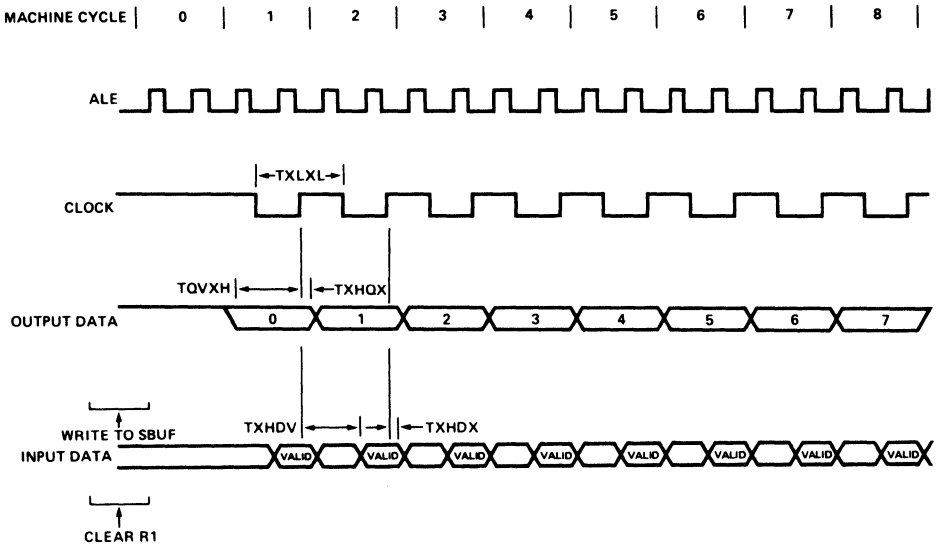
SERIAL PORT TIMING

I/O EXPANSION MODE

(TA = −40°C to +85°C; V_{CC} = 4 to 6V; V_{SS} = 0V, 0.5 to 12 MHz; Load capacitance = 80 pF)

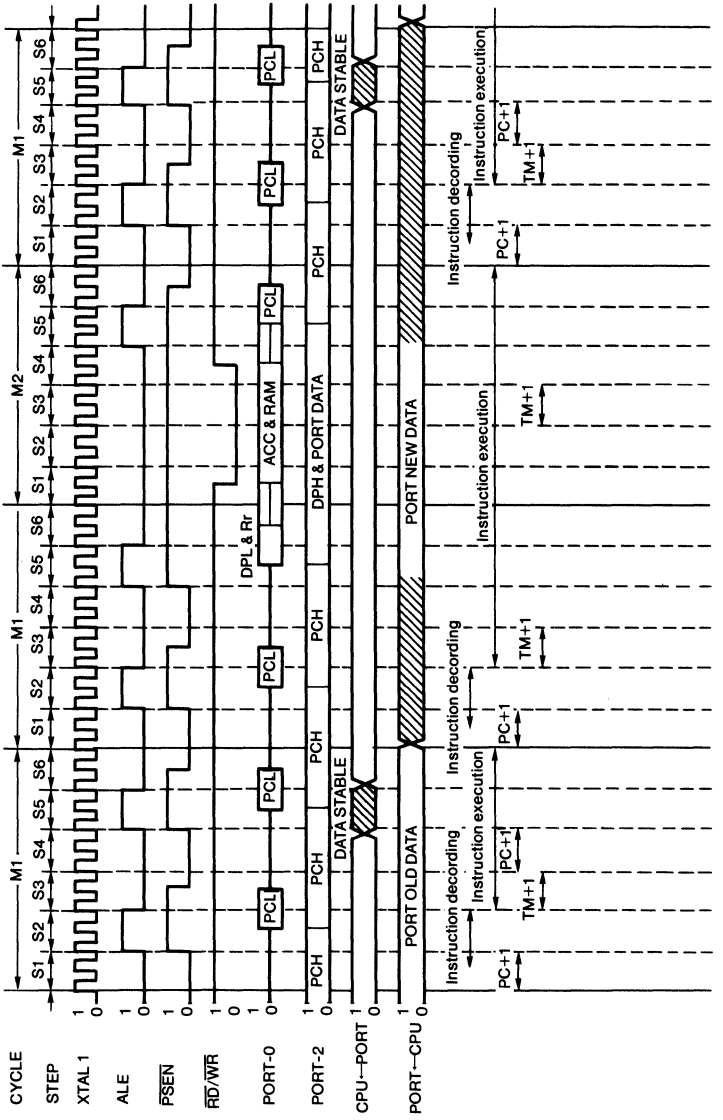
Symbol	Parameter	Min.	Max.	Units
TXLXL	Serial port clock cycle time	12TCLCL	–	μs
TQVXH	Output data setup to clock rising edge	10TCLCL-133	–	ns
TXHQX	Output data hold after clock rising edge	2TCLCL-117	–	ns
TXHDX	Input data hold after clock rising edge	0	–	ns
TXHDV	Clock rising edge to input data valid	–	10TCLCL-133	ns

SERIAL PORT (I/O EXPANSION MODE)

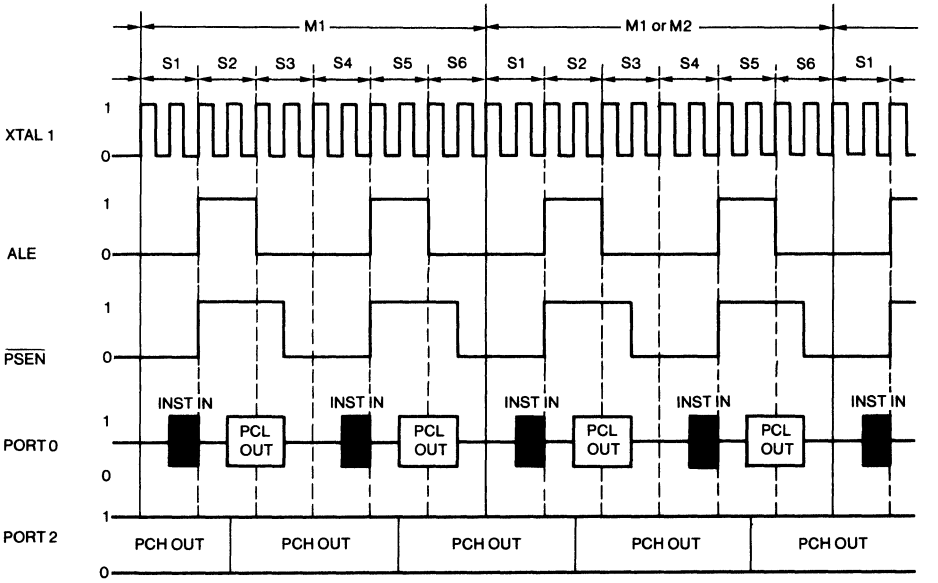


CLOCK WAVEFORMS

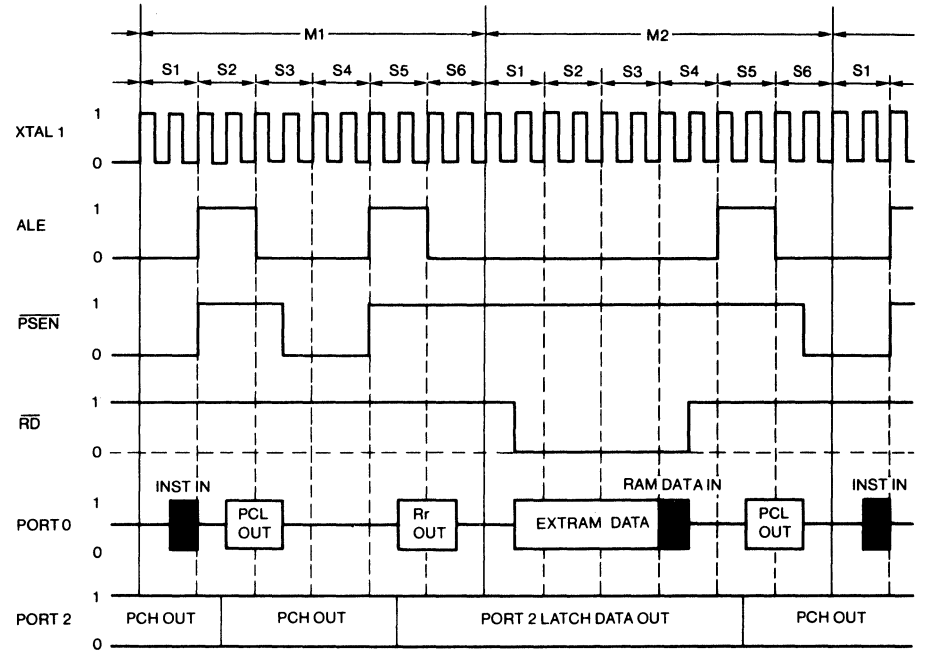
BASIC TIMING CHART



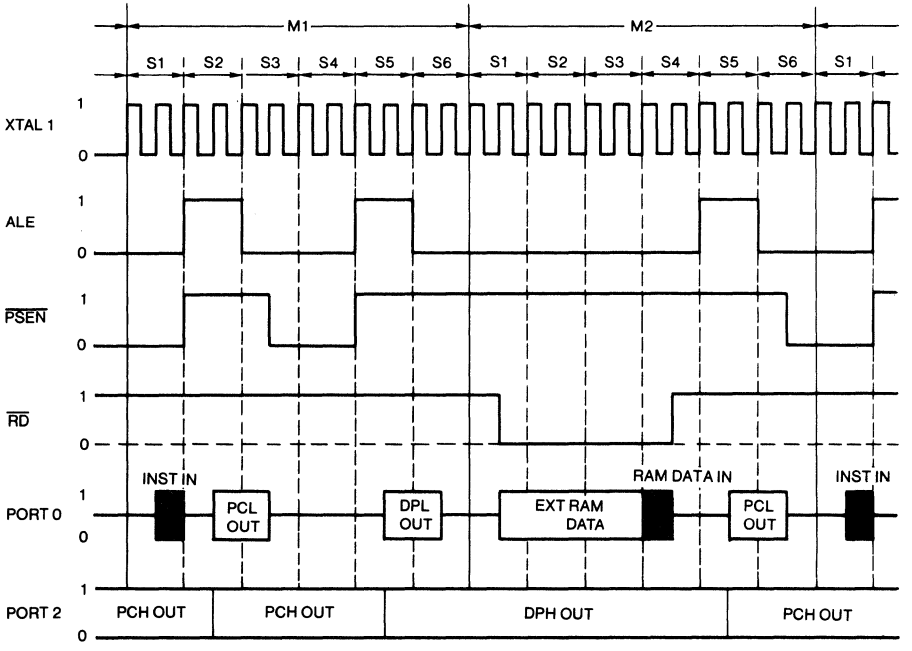
EXTERNAL PROGRAM MEMORY FETCH



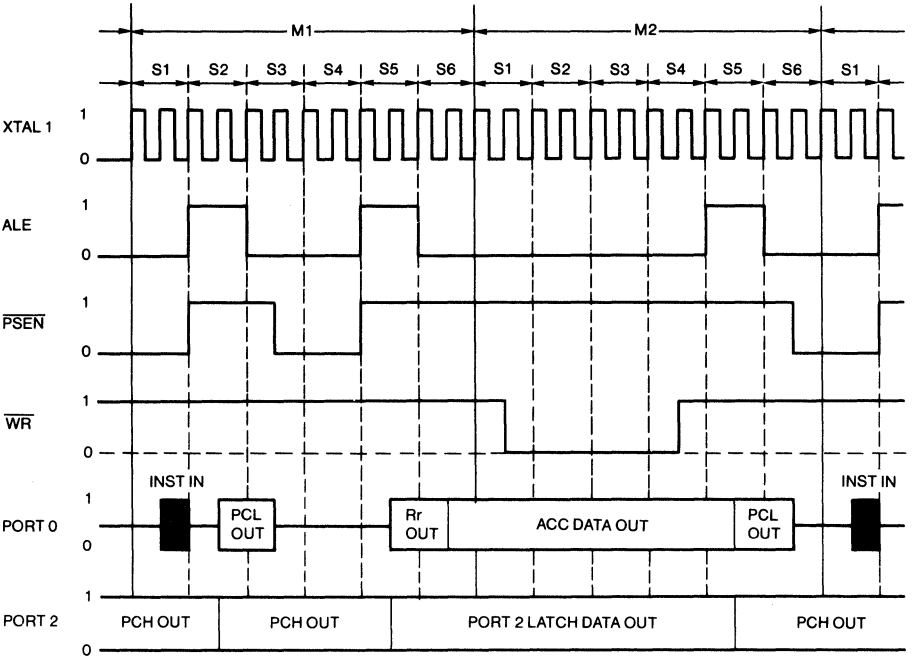
READ CYCLE 1 (MOVX A, @Rr)



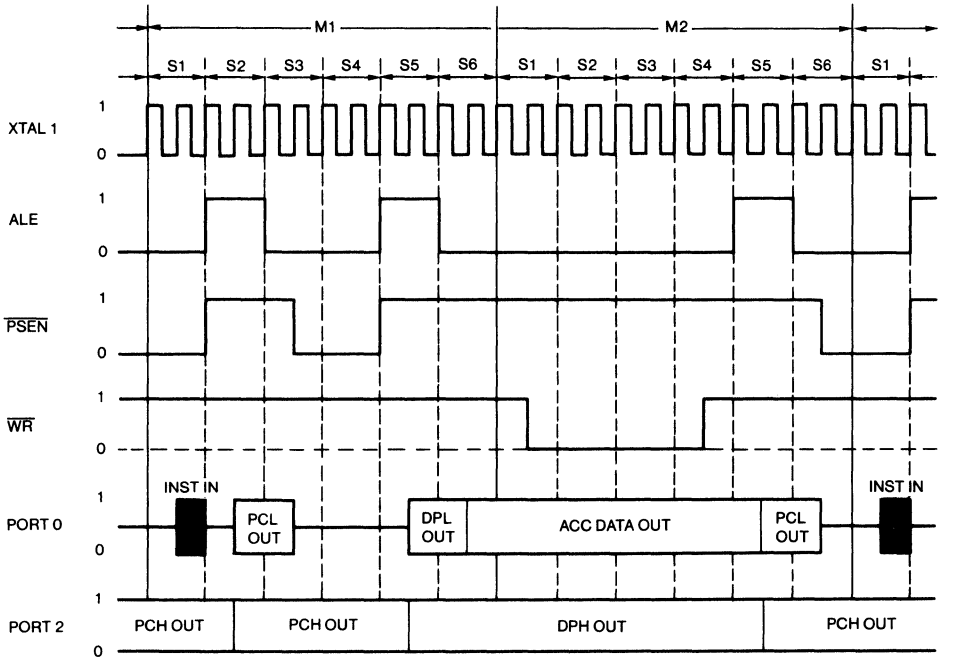
READ CYCLE 2 (MOVX A, @DPTR)



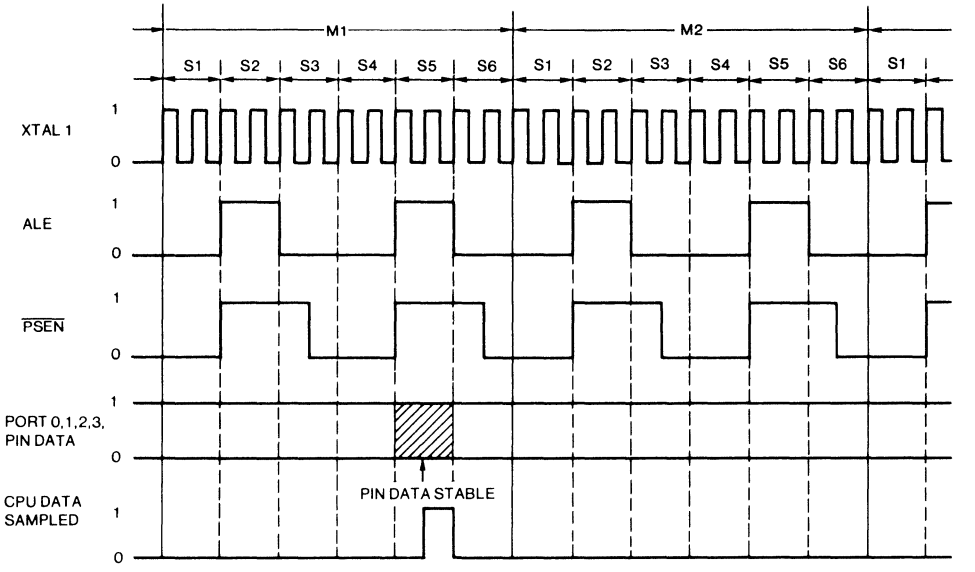
WRITE CYCLE 1 (MOVX, @Rr, A)



WRITE CYCLE 2 (MOVX A, @DPTR, A)



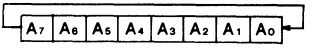
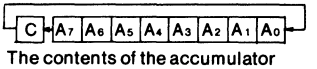
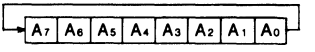
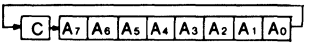
PORT OPERATION



INSTRUCTION SET DETAILS

	Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Hexa-decimal	Byte	Cycle	Explanation
Arithmetic operations	ADD A, Rn	0 0 1 0 1 n_2 n_1 n_0	28 ~ 2F	1	1	(A) ← (A) + (Rn)
	ADD A, direct	0 0 1 0 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	25 Byte 2	2	1	(A) ← (A) + (direct)
	ADD A, @Ri	0 0 1 0 0 1 1 1	26 ~ 27	1	1	(A) ← (A) + ((Ri))
	ADD A, #data	0 0 1 0 0 1 0 0 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	24 Byte 2	2	1	(A) ← (A) + #data
	ADDC A, Rn	0 0 1 1 1 n_2 n_1 n_0	38 ~ 3F	1	1	(A) ← (A) + (C) + (Rn)
	ADDC A, direct	0 0 1 1 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	35 Byte 2	2	1	(A) ← (A) + (C) + (direct)
	ADDC A, @Ri	0 0 1 1 0 1 1 1	36 ~ 37	1	1	(A) ← (A) + (C) + ((Ri))
	ADDC A, #data	0 0 1 1 0 1 0 0 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	34 Byte 2	2	1	(A) ← (A) + (C) + #data
	SUBB A, Rn	1 0 0 1 1 n_2 n_1 n_0	98 ~ 9F	1	1	(A) ← (A) − ((C) + (Rn))
	SUBB A, direct	1 0 0 1 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	95 Byte 2	2	1	(A) ← (A) − ((C) + (direct))
	SUBB A, @Ri	1 0 0 1 0 1 1 1	96 ~ 97	1	1	(A) ← (A) − ((C) + ((Ri)))
	SUBB A, #data	1 0 0 1 0 1 0 0 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	94 Byte 2	2	1	(A) ← (A) − ((C) + #data)
	INC A	0 0 0 0 0 1 0 0	04	1	1	(A) ← (A) + 1
	INC Rn	0 0 0 0 1 n_2 n_1 n_0	08 ~ 0F	1	1	(Rn) ← (Rn) + 1
	INC direct	0 0 0 0 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	05 Byte 2	2	1	(direct) ← (direct) + 1
	INC @Ri	0 0 0 0 0 1 1 1	06 ~ 07	1	1	((Ri)) ← ((Ri)) + 1
	INC DPTR	1 0 1 0 0 0 1 1	A3	1	2	(DPTR) ← (DPTR) + 1
	DEC A	0 0 0 1 0 1 0 0	14	1	1	(A) ← (A) − 1
	DEC Rn	0 0 0 1 1 n_2 n_1 n_0	18 ~ 1F	1	1	(Rn) ← (Rn) − 1
	DEC direct	0 0 0 1 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	15	2	1	(direct) ← (direct) − 1
	DEC @Ri	0 0 0 1 0 1 1 1	16 ~ 17	1	1	((Ri) ← ((Ri)) − 1
	MUL AB	1 0 1 0 0 1 0 0	A4	1	4	$(B_{15} \sim _8)$, $(A_7 \sim _0)$ ← (A) × (B)
	DIV AB	1 0 0 0 0 1 0 0	84	1	4	$(A_{15} \sim _8)$, $(B_7 \sim _0)$ ← (A)/(B)
	DA A	1 1 0 1 0 1 0 0	D4	1	1	Contents of Accumulator are BCD, IF [[$(A_3 \sim _0) > 9$] or [(AC) = 1]] THEN $(A_3 \sim _0)$ ← $(A_3 \sim _0)$ + 6 AND IF [[$(A_7 \sim _4) > 9$] OR [(C) = 1]] THEN $(A_7 \sim _4)$ ← $(A_7 \sim _4)$ + 6
Logical operations	ANL A, Rn	0 1 0 1 1 n_2 n_1 n_0	58 ~ 5F	1	1	(A) ← (A) AND (Rn)
	ANL A, direct	0 1 0 1 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	55 Byte 2	2	1	(A) ← (A) AND (direct)
	ANL A, @Ri	0 1 0 1 0 1 1 1	56 ~ 57	1	1	(A) ~ (A) AND ((Ri))
	ANL A, #data	0 1 0 1 0 1 0 0 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	54 Byte 2	2	1	(A) ← (A) AND #data

INSTRUCTION SET DETAILS (CONT.)

	Mnemonic	Instruction Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexa-decimal	Byte	Cycle	Explanation
Logical operations	ANL direct, A	0	1	0	1	0	0	1	0	52	2	1	(direct) ← (direct) AND (A)
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	ANL direct, #data	0	1	0	1	0	0	1	1	53	3	2	(direct) ← (direct) AND #data
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
		d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0	Byte 3			
	ORL A, Rn	0	1	0	0	1	n_2	n_1	n_0	48 ~ 4F	1	1	(A) ← (A) OR (Rn)
	ORL A, direct	0	1	0	0	0	1	0	1	45	2	1	(A) ← (A) OR (direct)
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	ORL A, @Ri	0	1	0	0	0	1	1	1	46 ~ 47	1	1	(A) ← (A) OR ((Ri))
	ORL A, #data	0	1	0	0	0	1	0	0	44	2	1	(A) ← (A) OR #data
		d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0	Byte 2			
	ORL direct, A	0	1	0	0	0	0	1	0	42	2	1	(direct) ← (direct) OR (A)
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	ORL direct, #data	0	1	0	0	0	0	1	1	43	3	2	(direct) ← (direct) OR #data
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
		d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0	Byte 3			
	XRL A, Rn	0	1	1	0	1	n_2	n_1	n_0	68 ~ 6F	1	1	(A) ← (A) XOR (Rn)
	XRL A, direct	0	1	1	0	0	1	0	1	65	2	1	(A) ← (A) XOR (direct)
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	XRL A, @Ri	0	1	1	0	0	1	1	1	66 ~ 67	1	1	(A) ← (A) XOR ((Ri))
	XRL A, #data	0	1	1	0	0	1	0	0	64	2	1	(A) ← (A) XOR #data
		d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0	Byte 2			
	XRL direct, A	0	1	1	0	0	0	1	0	62	2	1	(direct) ← (direct) XOR (A)
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
	XRL direct, #data	0	1	1	0	0	0	1	1	63	3	2	(direct) ← (direct) XOR #data
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			
		d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0	Byte 3			
	CLR A	1	1	1	0	0	1	0	0	E4	1	1	(A) ← 0
	CPL A	1	1	1	1	0	1	0	0	F4	1	1	(A) ← ($\overline{A}$)
	RL A	0	0	1	0	0	0	1	1	23	1	1	 The contents of the accumulator are rotated left by one bit.
	RLC A	0	0	1	1	0	0	1	1	33	1	1	 The contents of the accumulator and carry are rotated left by one bit.
	RR A	0	0	0	0	0	0	1	1	03	1	1	 The contents of the accumulator are rotated right by one bit.
	RRC A	0	0	0	1	0	0	1	1	13	1	1	 The contents of the accumulator and carry are rotated right by one bit.
	SWAP A	1	1	0	0	0	1	0	0	C4	1	1	($A_{3 \sim 0}$) ⇄ ($A_{7 \sim 4}$)

INSTRUCTION SET DETAILS (CONT.)

	Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Hexa-decimal	Byte	Cycle	Explanation
Data transfer	MOV A, Rn	1 1 1 0 1 n_2 n_1 n_0	E8 ~ EF	1	1	(A) ← (Rn)
	MOV A, direct	1 1 1 0 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	E5 Byte 2	2	1	(A) ← (direct)
	MOV A, @Ri	1 1 1 0 0 1 1 1	E6 ~ E7	1	1	(A) ← ((Ri))
	MOV A, #data	0 1 1 1 0 1 0 0 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	74 Byte 2	2	1	(A) ← #data
	MOV Rn, A	1 1 1 1 1 n_2 n_1 n_0	F8 ~ FF	1	1	(Rn) ← (A)
	MOV Rn, direct	1 0 1 0 1 n_2 n_1 n_0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	A8 ~ AF Byte 2	2	2	(Rn) ← (direct)
	MOV Rn, #data	0 1 1 1 1 n_2 n_1 n_0 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	78 ~ 7F Byte 2	2	1	(Rn) ← #data
	MOV direct, A	1 1 1 1 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	F5 Byte 2	2	1	(direct) ← (A)
	MOV direct, Rn	1 0 0 0 1 n_2 n_1 n_0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	88 ~ 8F Byte 2	2	2	(direct) ← (Rn)
	MOV direct 1, direct 2	1 0 0 0 0 1 0 1 a_8^2 a_8^2 a_5^2 a_4^2 a_3^2 a_2^2 a_1^2 a_0^2 a_7^1 a_6^1 a_5^1 a_4^1 a_3^1 a_2^1 a_1^1 a_0^1	85 Byte 2 Byte 3	3	2	(direct 1) ← (direct 2)
	MOV direct, @Ri	1 0 0 0 0 1 1 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	86 ~ 87 Byte 2	2	2	(direct) ← ((Ri))
	MOV direct, #data	0 1 1 1 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	75 Byte 2 Byte 3	3	2	(direct) ← #data
	MOV @Ri, A	1 1 1 1 0 1 1 1	F6 ~ F7	1	1	((Ri)) ← A
	MOV @Ri, direct	1 0 1 0 0 1 1 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	A6 ~ A7 Byte 2	2	2	((Ri)) ← (direct)
	MOV @Ri, #data	0 1 1 1 0 1 1 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	76 ~ 77 Byte 2	2	1	((Ri)) ← #data
	MOV DPTR, #data 16	1 0 0 1 0 0 0 0 d_{15} d_{14} d_{13} d_{12} d_{11} d_{10} d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	90 Byte 2 Byte 3	3	2	(DPTR) ← #data$_{16}$
	MOVC A, @A+ DPTR	1 0 0 1 0 0 1 1	93	1	2	(A) ← ((A) + (DPTR))
	MOVC A, @A+PC	1 0 0 0 0 0 1 1	83	1	2	(PC) ← (PC) + 1, (A) ← ((A) + (PC))
	MOVX A, @Ri	1 1 1 0 0 0 1 1	E2 ~ E3	1	2	(A) ← ((Ri)) External RAM
	MOVX A, @DPTR	1 1 1 0 0 0 0 0	E0	1	2	(A) ← ((DPTR)) External RAM
	MOVX @Ri, A	1 1 1 1 0 0 1 1	F2 ~ F3	1	2	((Ri)) ← (A) External RAM
	MOVX @DPTR, A	1 1 1 1 0 0 0 0	F0	1	2	((DPTR)) ← (A) External RAM
	PUSH direct	1 1 0 0 0 0 0 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	C0 Byte 2	2	2	(SP) ← (SP) + 1 ((SP)) ← (direct)
	POP direct	1 1 0 1 0 0 0 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	D0 Byte 2	2	2	(direct) ← ((SP)) (SP) ← (SP) − 1
	XCH A, Rn	1 1 0 0 1 n_2 n_1 n_0	C8 ~ CF	1	1	(A) ⇄ (Rn)
	XCH A, direct	1 1 0 0 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	C5 Byte 2	2	1	(A) ⇄ (direct)

INSTRUCTION SET DETAILS (CONT.)

	Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Hexa-decimal	Byte	Cycle	Explanation
Data transfer	XCH A, @Ri	1 1 0 0 0 1 1 1	C6 ~ C7	1	1	(A) $\rightleftarrows$ ((Ri))
	XCHD A, @Ri	1 1 0 1 0 1 1 1	D6 ~ D7	1	1	($A_{3\sim0}$) $\rightleftarrows$ (($Ri_{3\sim0}$))
Boolean variable manipulation	CLR C	1 1 0 0 0 0 1 1	C3	1	1	(C) ← 0
	CLR bit	1 1 0 0 0 0 1 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	C2 Byte 2	2	1	(bit) ← 0
	SETB C	1 1 0 1 0 0 1 1	D3	1	1	(C) ← 1
	SETB bit	1 1 0 1 0 0 1 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	D2 Byte 2	2	1	(bit) ← 1
	CPL C	1 0 1 1 0 0 1 1	B3	1	1	(C) ← ($\overline{C}$)
	CPL bit	1 0 1 1 0 0 1 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	B2 Byte 2	2	1	(bit) ← ($\overline{bit}$)
	ANL C, bit	1 0 0 0 0 0 1 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	82 Byte 2	2	2	(C) ← (C) AND (bit)
	ANL C,/bit	1 0 1 1 0 0 0 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	B0 Byte 2	2	2	(C) ← (C) AND ($\overline{bit}$)
	ORL C, bit	0 1 1 1 0 0 1 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	72 Byte 2	2	2	(C) ← (C) OR (bit)
	ORL C,/bit	1 0 1 0 0 0 0 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	A0 Byte 2	2	2	(C) ← (C) OR ($\overline{bit}$)
	MOV C, bit	1 0 1 0 0 0 1 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	A2 Byte 2	2	1	(C) ← (bit)
	MOV bit, C	1 0 0 1 0 0 1 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	92 Byte 2	2	2	(bit) ← (C)
Program branching	ACALL addr 11	a_{10} a_9 a_8 1 0 0 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	Byte 1 Byte 2	2	2	(PC) ← (PC) + 2 (SP) ← (SP) + 1 ((SP)) ← ($PC_{7\sim0}$) (SP) ← (SP) + 1 ((SP)) ← ($PC_{15\sim8}$) (PC) ← page address
	LCALL addr 16	0 0 0 1 0 0 1 0 a_{15} a_{14} a_{13} a_{12} a_{11} a_{10} a_9 a_8 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	12 Byte 2 Byte 3	3	2	(PC) ← (PC) + 3 (SP) ← (SP) + 1 ((SP)) ← ($PC_{7\sim0}$) (SP) ← (SP) + 1 ((SP)) ← ($PC_{15\sim8}$) (PC) ← $addr_{15\sim0}$
	RET	0 0 1 0 0 0 1 0	22	1	2	($PC_{15\sim8}$) ← ((SP)) (SP) ← (SP) − 1 ($PC_{7\sim0}$) ← ((SP)) (SP) ← (SP) − 1
	RETI	0 0 1 1 0 0 1 0	32	1	2	($PC_{15\sim8}$) ← ((SP)) (SP) ← (SP) − 1 ($PC_{7\sim0}$) ← ((SP)) (SP) ← (SP) − 1
	AJMP addr 11	a_{10} a_9 a_8 0 0 0 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	Byte 1 Byte 2	2	2	(PC) ← (PC) + 2 ($PC_{10\sim0}$) ← page address
	LJMP addr 16	0 0 0 0 0 0 1 0 a_{15} a_{14} a_{13} a_{12} a_{11} a_{10} a_9 a_8 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	02 Byte 2 Byte 3	3	2	(PC) ← $addr_{15\sim0}$

INSTRUCTION SET DETAILS (CONT.)

	Mnemonic	Instruction Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexa-decimal	Byte	Cycle	Explanation
Program branching	SJMP rel	1	0	0	0	0	0	0	0	80	2	2	(PC) ← (PC) + 2
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 2			(PC) ← (PC) + rel
	JMP @A+DPTR	0	1	1	1	0	0	1	1	73	1	2	(PC) ← (A) + (DPTR)
	JZ rel	0	1	1	0	0	0	0	0	60	2	2	(PC) ← (PC) + 2
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte			IF (A) = 0 THEN (PC) ← (PC) + rel
	JNZ rel	0	1	1	1	0	0	0	0	70	2	2	(PC) ← (PC) + 2
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 2			IF (A) ≠ 0 THEN (PC) ← (PC) + rel
	JC rel	0	1	0	0	0	0	0	0	40	2	2	(PC) ← (PC) + 2
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 2			IF (C) ≠ 1 THEN (PC) ← (PC) + rel
	JNC rel	0	1	0	1	0	0	0	0	50	2	2	(PC) ← (PC) + 2
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 2			IF (C) ≠ 0 THEN (PC) ← (PC) + rel
	JB bit, rel	0	0	1	0	0	0	0	0	20	3	2	(PC) ← (PC) + 3
		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Byte 2			IF (bit) = 1 THEN (PC) ← (PC) + rel
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 3			
	JNB bit, rel	0	0	1	1	0	0	0	0	30	3	2	(PC) ← (PC) + 3
		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Byte 2			IF (bit) = 0 THEN (PC) ← (PC) + rel
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 3			
	JBC bit, rel	0	0	0	1	0	0	0	0	10	3	2	(PC) ← (PC) + 3
		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Byte 2			IF (bit) = 1 THEN (bit) ← 0
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 3			(PC) ← (PC) + rel
	CJNE A, direct, rel	1	0	1	1	0	1	0	1	B5	3	2	(PC) ← (PC) + 3
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			IF (direct) < (A) THEN (PC) ← (PC) + rel and (C) ← 0
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 3			IF (direct) > (A) THEN (PC) ← (PC) + rel and (C) ← 1
	CJNE A, #data, rel	1	0	1	1	0	1	0	0	B4	3	2	(PC) ← (PC) + 3
		d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0	Byte 2			IF #data < (A) THEN (PC) ← (PC) + rel and (C) ← 0
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 3			IF #data> (A) THEN (PC) ← (PC) + rel and (C) ← 1
	CJNE Rn, #data, rel	1	0	1	1	1	n_2	n_1	n_0	B8 ~ BF	3	2	(PC) ← (PC) + 3
		d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0	Byte 2			IF #data < (Rn) THEN (PC) ← (PC) + rel and (C) ← 0
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 3			IF #data> (Rn) THEN (PC) ← (PC) + rel and (C) ← 1
	CJNE @Ri, #data, rel	1	0	1	1	0	1	1	1	B6 ~ B7	3	2	(PC) ← (PC) + 3
		d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0	Byte 2			IF #data < ((Ri)) THEN (PC) ← (PC) + rel and (C) ← 0
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 3			IF #data> ((Ri)) THEN (PC) ← (PC) + rel and (C) ← 1
	DJNZ Rn, rel	1	1	0	1	1	n_2	n_1	n_0	D8 ~ DF	2	2	(PC) ← (PC) + 2
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 2			(Rn) ← (Rn) − 1
													IF (Rn) ≠ 0 THEN (PC) ← (PC) + rel
	DJNZ direct, rel	1	1	0	1	0	1	0	1	D5	3	2	(PC) ← (PC) + 3
		a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	Byte 2			(direct) ← (direct) − 1
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 3			IF (direct) ≠ 0 THEN (PC) ← (PC) + rel
	NOP	0	0	0	0	0	0	0	0	00	1	1	(PC) ← (PC) + 1

NOTES ON THE INSTRUCTION SET AND THE ADDRESSING MODES

Rn	– Register R7-R0 of the currently selected Register Bank.
direct	– 8-bit internal data location's address. This could be an Internal Data RAM location (0 – 127) or a SFR [i.e., I/O port, control register, status register, etc. (128 – 255)].
@Ri	– 8-bit internal data RAM location (0 – 255) addressed indirectly through register R1 or R0.
#data	– 8-bit constant included in instruction.
#data 16	– 16-bit constant included in instruction.
addr 16	– 16-bit destination address. Used by LCALL & LJMP. A branch can be anywhere within the 64K-byte Program Memory address space.
addr 11	– 11-bit destination address. Used by ACALL & AJMP. The branch will be within the same 2K-byte page of program memory as the first byte of the following instruction.
rel	– Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is −128 to +127 bytes relative to first byte of the following instruction.
bit	– Direct Addressed bit in Internal Data RAM or Special Function Register.

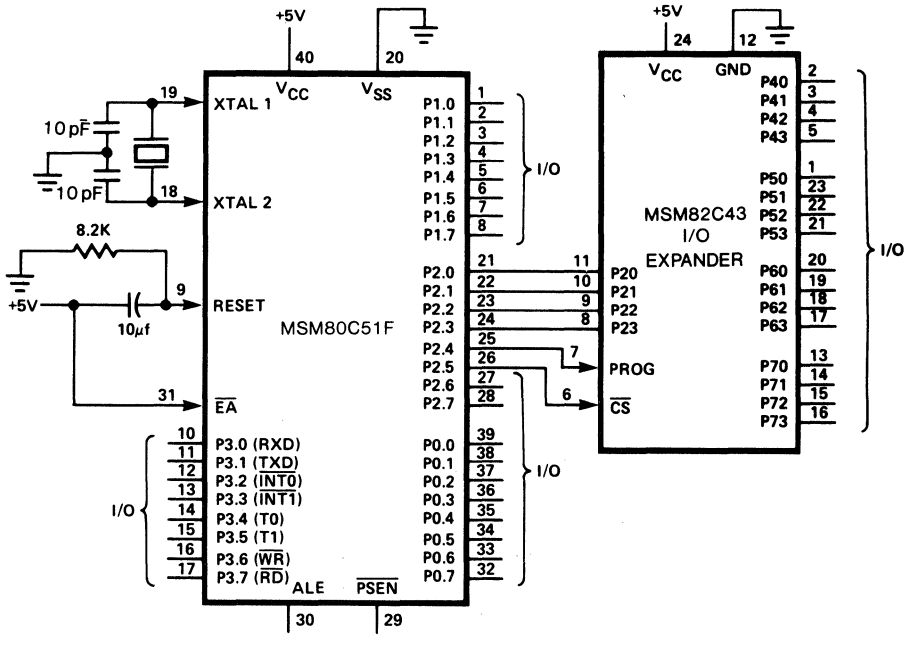
INSTRUCTIONS THAT AFFECT FLAG SETTINGS[1]

INSTRUCTION	FLAG		
	C	OV	AC
ADD	X	X	X
ADDC	X	X	X
SUBB	X	X	X
MUL	O	X	
DIV	O	X	
DA	X		
RRC	X		
RLC	X		
SETB C	I		

INSTRUCTION	FLAG		
	C	OV	AC
CLR C	O		
CPL C	X		
ANL C, bit	X		
ANL C,/bit	X		
ORL C, bit	X		
ORL C, /bit	X		
MOV C, bit	X		
CJNE	X		

[1] Note that operations on SFR byte address 208 or bit addresses 208-215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

APPLICATION EXAMPLES



The following software driver is required to interface to the 82C43
Mixing Parallel Output, Input, and
Control Strobes on Port 2.

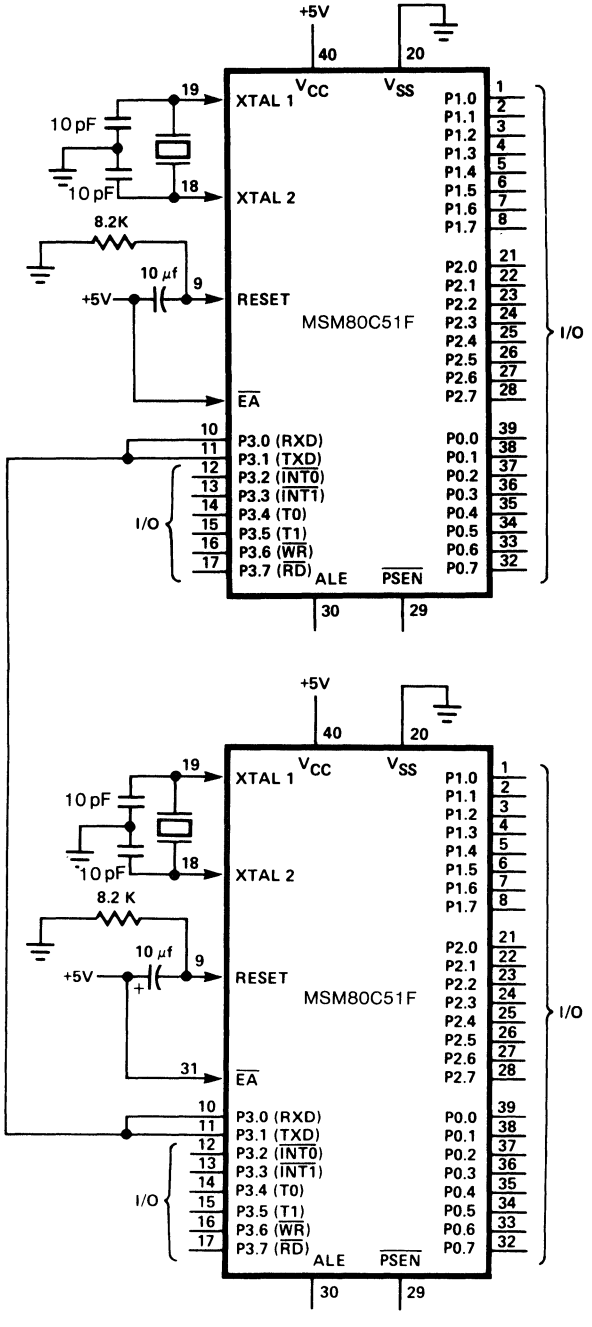
```
IN82C43     INPUT DATA FROM AN 82C43 I/O EXPANDER
            CONNECT TO P23-P20
            P25 & P24 MIMIC CS/ & PROG
            P27-P26 USED AS INPUTS
            PORT TO BE READ IN ACC

IN82C43     MOV     A, #11010000B
            MOV     P2,A          ;OUTPUT INSTRUCTION CODE
            CLR     P2.4          ;FALLING EDGE OF PROG
            ORL     P2,#00001111B ;SET FOR INPUT
            MOV     A,P2          ;READ INPUT DATA
            SETB    P2.4          ;RETURN PROG HIGH
```

P21	P20	Address Code	P23	P22	Instruction Code
0	0	Port 4	0	0	Read
0	1	Port 5	0	1	Write
1	0	Port 6	1	0	ORLD
1	1	Port 7	1	1	ANLD

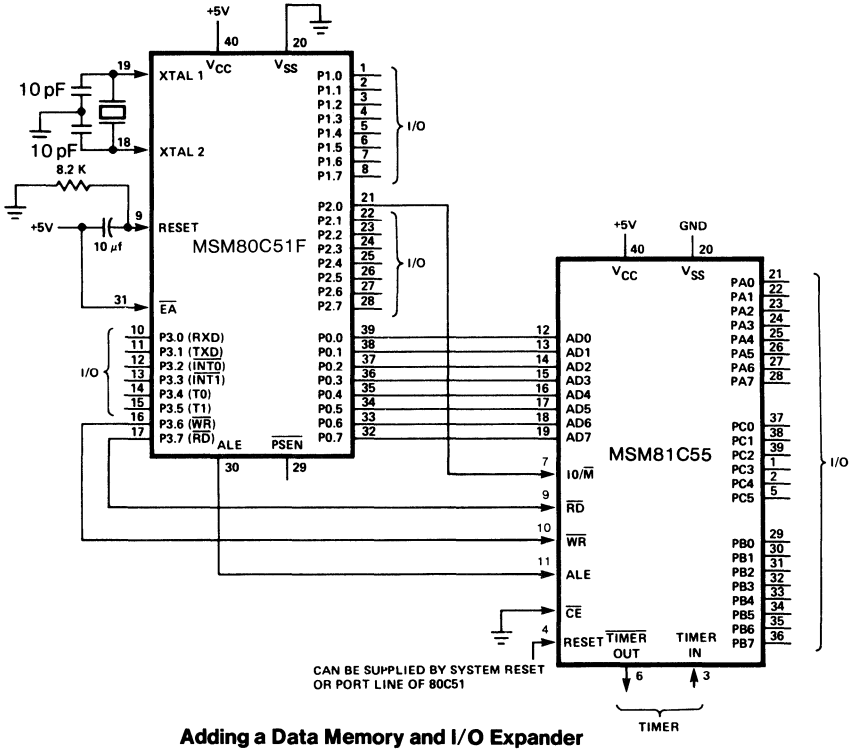
I/O Expansion Using an 82C43

APPLICATION EXAMPLES (CONT.)

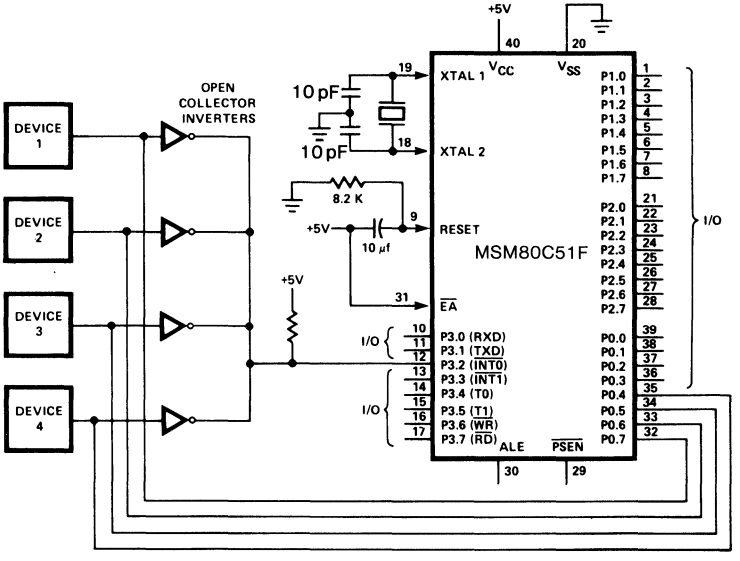


Multiple 80C51F's Using Half-Duplex Serial Communication

APPLICATION EXAMPLES (CONT.)



Adding a Data Memory and I/O Expander



Multiple Interrupt Sources

MSM80C31/MSM80C51 INSTRUCTION CODES

H \ L	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111
0 0000	NOP	AJMP address 11 (Page 0)	LJMP address 16	RR A	INC A	INC direct	INC @R0	INC @R1
1 0001	JBC bit, rel	ACALL address 11 (Page 0)	LCALL address 16	RRC A	DEC A	DEC direct	DEC @R0	DEC @R1
2 0010	JB bit, rel	AJMP address 11 (Page 1)	RET	RL A	ADD A, #data	ADD A, direct	ADD A, @R0	ADD A, @R1
3 0011	JNB bit, rel	ACALL address 11 (Page 1)	RETI	RLC A	ADDC A, #data	ADDC A, direct	ADDC A, @R0	ADDC A, @R1
4 0100	JC bit, rel	AJMP address 11 (Page 2)	ORL direct, A	ORL direct, #data	ORL A, #data	ORL A, direct	ORL A, @R0	ORL A, @R1
5 0101	JNC rel	ACALL address 11 (Page 2)	ANL direct, A	ANL direct, #data	ANL A, #data	ANL A, direct	ANL A, @R0	ANL A, @R1
6 0110	JZ rel	AJMP address 11 (Page 3)	XRL direct, A	XRL direct, #data	XRL A, #data	XRL A, direct	XRL A, @R0	XRL A, @R1
7 0111	JNZ rel	ACALL address 11 (Page 3)	ORL C, bit	JMP @A+DPTR	MOV A, #data	MOV direct, #data	MOV @R0, #data	MOV @R1, #data
8 1000	SJMP rel	AJMP address 11 (Page 4)	ANL C, bit	MOVC A, @A+PC	DIV AB	MOV direct 1, direct 2	MOV direct, @R0	MOV direct, @R1
9 1001	MOV DPTR, #data 16	ACALL address 11 (Page 4)	MOV bit, C	MOVC A, @A+DPTR	SUBB A, #data	SUBB A, direct	SUBB A, @R0	SUBB A, @R1
A 1010	ORL C/bit	AJMP address 11 (Page 5)	MOV C, bit	INC DPTR	MUL AB		MOV @R0, direct	MOV @R1, direct
B 1011	ANL C/bit	ACALL address 11 (Page 5)	CPL bit	CPL C	CJNE A, #data, rel	CJNE A, direct, rel	CJNE @R0, #data, rel	CJNE @R1, #data, rel
C 1100	PUSH direct	AJMP address 11 (Page 6)	CLR bit	CLR C	SWAP A	XCH A, direct	XCH A, @R0	XCH A, @R1
D 1101	POP direct	ACALL address 11 (Page 6)	SETB bit	SETB C	DA A	DJNZ direct, rel	XCHD A, @R0	XCHD A, @R1
E 1110	MOVX A, @DPTR	AJMP address 11 (page 7)	MOVX A, @R0	MOVX A, @R1	CLR A	MOV A, direct	MOV A, @R0	MOV A, @R1
F 1111	MOVX @DPTR, A	ACALL address 11 (page 7)	MOVX @R0, A	MOVX @R1, A	CPL A	MOV direct, A	MOV @R0, A	MOV @R1, A

2 BYTE	MNEMONIC	3 BYTE
2 CYCLE		4 CYCLE

8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
INC R0	INC R1	INC R2	INC R3	INC R4	INC R5	INC R6	INC R7
DEC R0	DEC R1	DEC R2	DEC R3	DEC R4	DEC R5	DEC R6	DEC R7
ADD A, R0	ADD A, R1	ADD A, R2	ADD A, R3	ADD A, R4	ADD A, R5	ADD A, R6	ADD A, R7
ADDC A, R0	ADDC A, R1	ADDC A, R2	ADDC A, R3	ADDC A, R4	ADDC A, R5	ADDC A, R6	ADDC A, R7
ORL A, R0	ORL A, R1	ORL A, R2	ORL A, R3	ORL A, R4	ORL A, R5	ORL A, R6	ORL A, R7
ANL A, R0	ANL A, R1	ANL A, R2	ANL A, R3	ANL A, R4	ANL A, R5	ANL A, R6	ANL A, R7
XRL A, R0	XRL A, R1	XRL A, R2	XRL A, R3	XRL A, R4	XRL A, R5	XRL A, R6	XRL A, R7
MOV R0, #data	MOV R1, #data	MOV R2, #data	MOV R3, #data	MOV R4, #data	MOV R5, #data	MOV R6, #data	MOV R7, #data
MOV direct, R0	MOV direct, R1	MOV direct, R2	MOV direct, R3	MOV direct, R4	MOV direct, R5	MOV direct, R6	MOV direct, R7
SUBB A, R0	SUBB A, R1	SUBB A, R2	SUBB A, R3	SUBB A, R4	SUBB A, R5	SUBB A, R6	SUBB A, R7
MOV R0, direct	MOV R1, direct	MOV R2, direct	MOV R3, direct	MOV R4, direct	MOV R5, direct	MOV R6, direct	MOV R7, direct
CJNE R0, #data, rel	CJNE R1, #data, rel	CJNE R2, #data, rel	CJNE R3, #data, rel	CJNE R4, #data, rel	CJNE R5, #data, rel	CJNE R6, #data, rel	CJNE R7, #data, rel
XCH A, R0	XCH A, R1	XCH A, R2	XCH A, R3	XCH A, R4	XCH A, R5	XCH A, R6	XCH A, R7
DJNZ R0, rel	DJNZ R1, rel	DJNZ R2, rel	DJNZ R3, rel	DJNZ R4, rel	DJNE R5, rel	DJNE R6, rel	DJNE R7, rel
MOV A, R0	MOV A, R1	MOV A, R2	MOV A, R3	MOV A, R4	MOV A, R5	MOV A, R6	MOV A, R7
MOV R0, A	MOV R1, A	MOV R2, A	MOV R3, A	MOV R4, A	MOV R5, A	MOV R6, A	MOV R7, A

OKI semiconductor

MSM80C154/ MSM83C154

CMOS 8-bit One-Chip Microcontroller

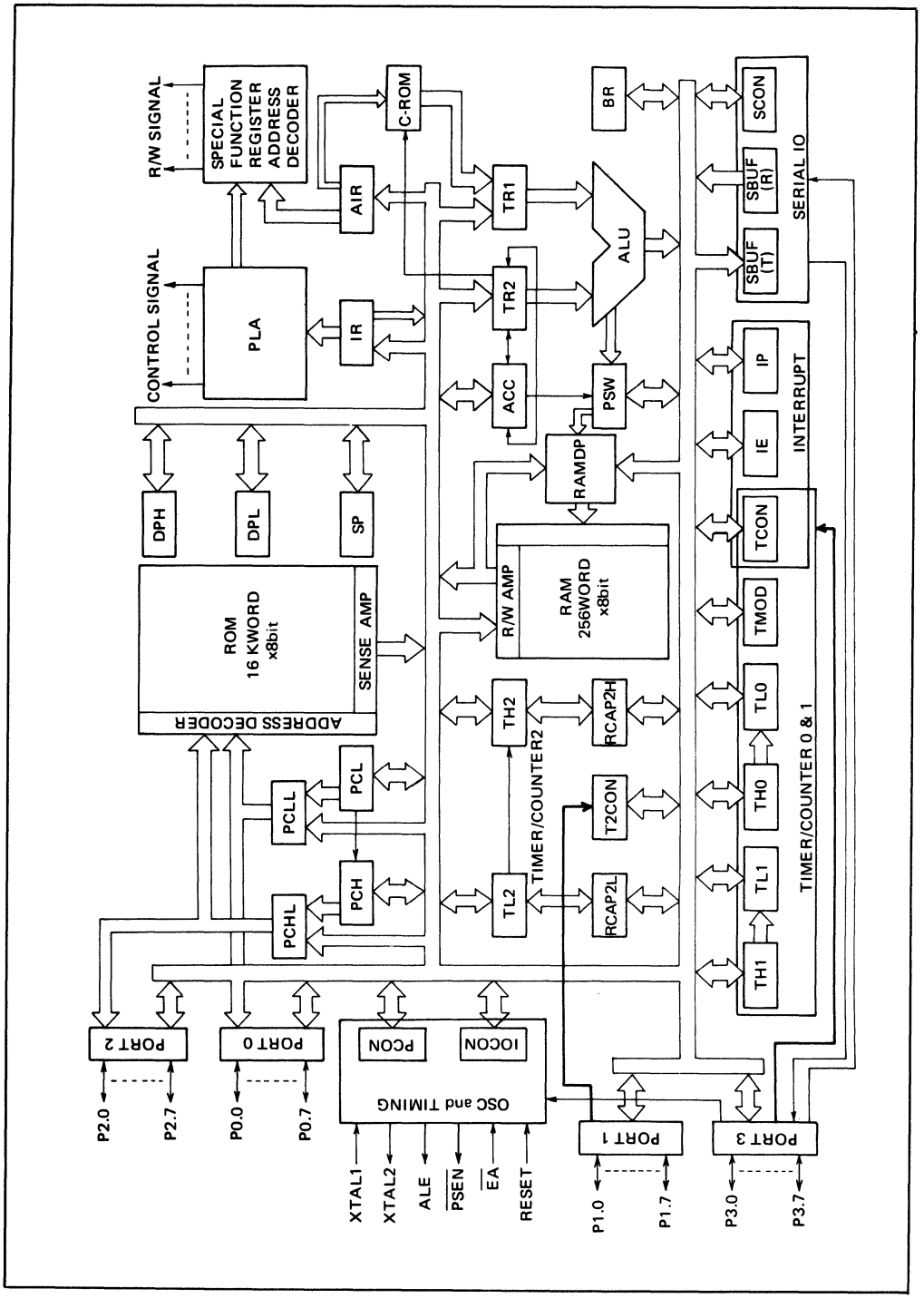
GENERAL DESCRIPTION

The MSM83C154/MSM80C154 is a high performance 8-bit one-chip microcontroller implementing large integration, high speed and low power consumption by 2 μm silicon gate CMOS process technology.
The MSM83C154 features 16K byte ROM, 256 byte RAM, 32 I/O ports, three 16-bit timer/counters, multifunctional serial port and clock generator. In addition, the MSM83C154 has three standby modes enabling further power reduction.
The MSM80C154 is identical to the MSM83C154 except the omission of 16K byte ROM.

FEATURES

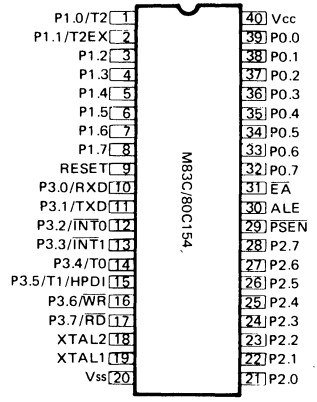
- Fully static circuit
- On-chip program memory : 16K x 8 bit ROM (MSM83C154 only)
- On-chip data memory : 256 x 8 bit RAM
- External program memory address space : 64K bytes
- External data memory address space : 64K bytes
- I/O ports
  (Port 1, 2, 3, impedance programmable) : 32
- 16-bit timer/counters
  (includes watch dog timer & 32 bit timer) : 3
- Multifunctional serial port : I/O Expansion mode
  : UART mode (featuring error detection)
- 6-source 2-priority level
  interrupt and multi-level
  interrupt available by programming IP and IE registers
- Memory-mapped special function registers
- Bit addressable data memory and SFRs
- Minimum instruction cycle : 1.0μs @ 12 MHz operation (MSM80C154/MSM83C154)
  : 0.75μs @ 16 MHz operation (MSM80C154-1/MSM83C154-1)
- "Multiply"/"divide" instruction cycle : 3 μs @16 MHz operation
- Standby functions : Idle mode (CPU halt)
  : Power down mode (Oscillator stop)
  Activated by Software or Hardware; Providing ports with floating or active status
  The software power down mode is terminated by interrupt signal enabling execution from the interrupted address.
- Lower power consumption achieved by 2 μm silicon gate CMOS process
- Upward compatible with MSM80C51/80C31
- Packages : 40 pin plastic DIP (DIP40-P-600)
  44 pin plastic QFP (QFP44-P-910-K)
  44 pin plastic QFP (QFP44-P-910-VIK)
  44 pin PLCC (QFJ44-P-S650)

CIRCUIT BLOCK DIAGRAM

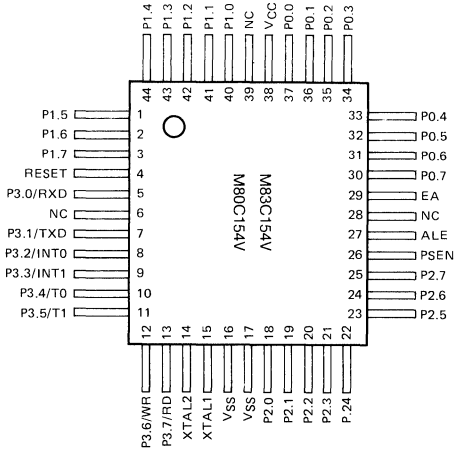


PIN CONFIGURATION

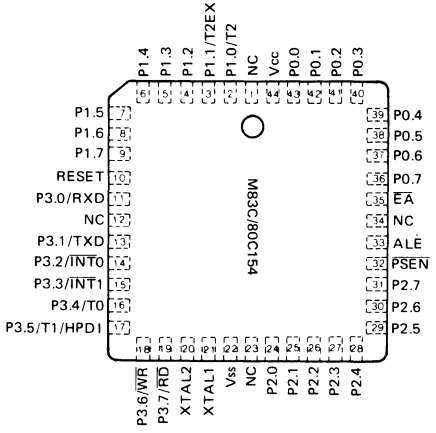
MSM83C154-XXRS/MSM80C154RS
(Top-View) 40 Lead Plastic DIP



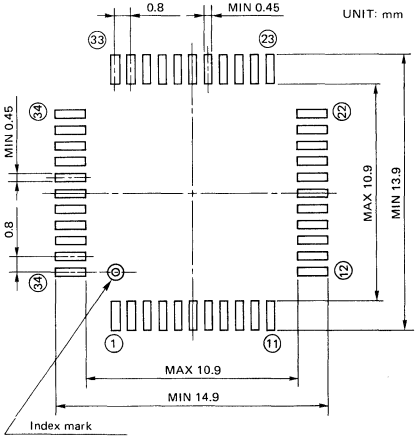
MSM83C154V-XXGS/MSM80C154VGS
(Top-View) 44 Lead Plastic Package



MSM83C154-XXJS/MSM80C154JS *1
(Top-View) 44 Lead Plastic Leaded Chip Carrier



RECOMMENDED FOOTPRINT
FOR QPD PACKAGE



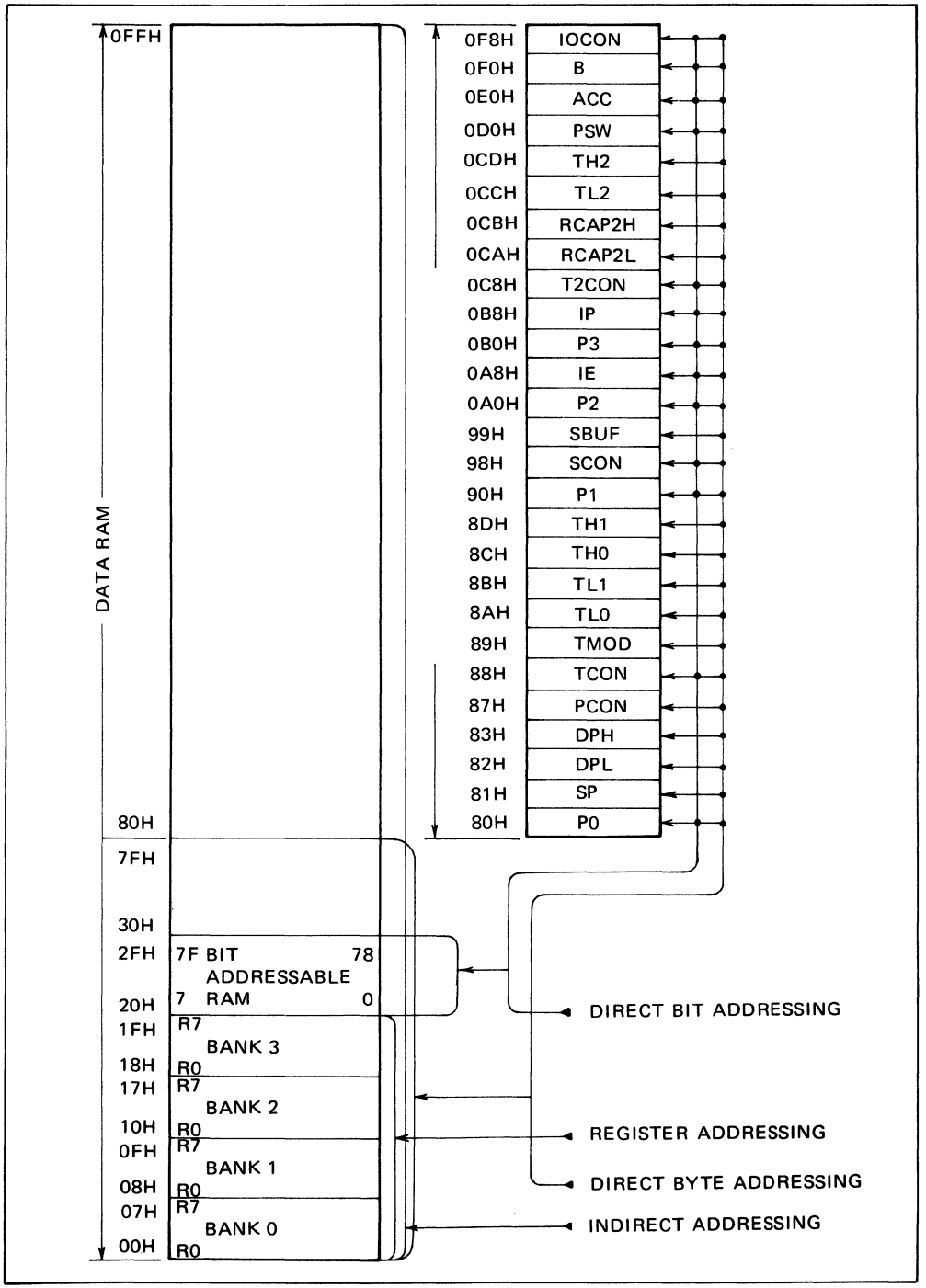
PIN FUNCTIONS

Pin Name	Description
P0.0 ~ P0.7	Bidirectional I/O ports. They are also the data/address bus (input/output of data and output of lower 8-bit address when external memory is accessed). They are open drain output when used as I/O ports, but tri-state output when used as data/address bus.
P1.0 ~ P1.7	P1.0 to P1.7 are quasi-bidirectional I/O ports. They are pulled up internally when used as input ports. Two of them have the following secondary functions: • P1.0 (T2) : Used as external clock input pin for the timer/counter 2. • P1.1 (T2EX) : Used as trigger input for the timer/counter 2 to be reloaded or captured; causing the timer/counter 2 interrput.

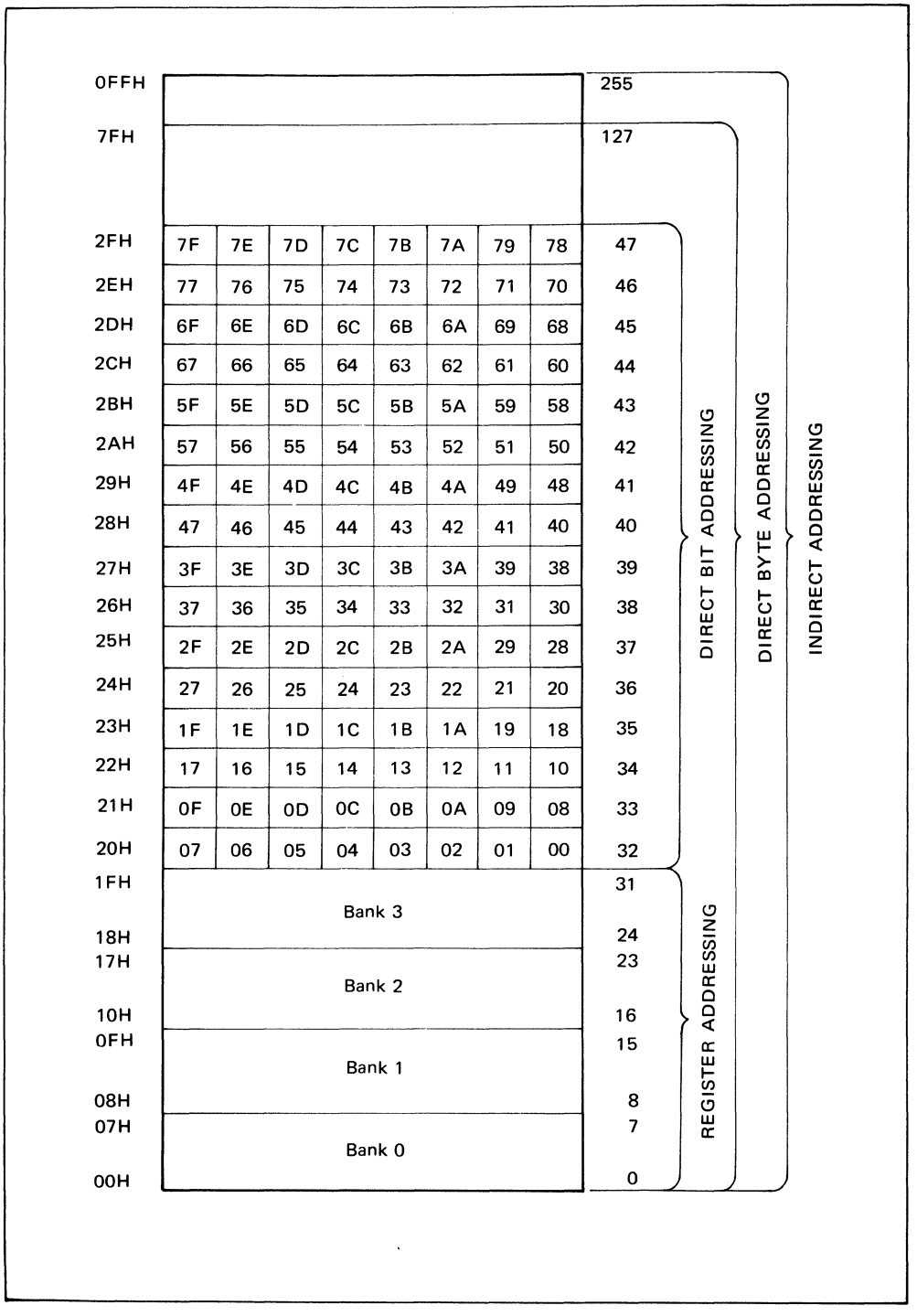
PIN FUNCTIONS (CONT.)

Pin Name	Description
P2.0 ~ P2.7	P2.0 to P2.7 are quasi-bidirectional I/O ports. They also output the higher 8-bit address when an external memory is accessed. They are pulled up internally when used as input ports.
P3.0 ~ P3.7	P3.0 to P3.7 are quasi-bidirectional I/O ports. They are pulled up internally when used as input ports. They also have the following secondary functions: • P3.0 (RXD) Serial data input/output in the I/O expansion mode and serial data input in the UART mode when the serial port is used. • P3.1 (TXD) Synchronous clock output in the I/O expansion mode and serial data output in the UART mode when the serial port is used. • P3.2 ($\overline{INT0}$) Used as input pin for the external interrupt 0, and as count-up control pin for the timer/counter 0. • P3.3 ($\overline{INT1}$) Used as input pin for the external interrupt 1, and as count-up control pin for the timer/counter 1. • P3.4 (T0) Used as external clock input pin for the timer/counter 0. • P3.5 (T1) Used as external clock input pin for the timer/counter 1 and power down mode control input pin. • P3.6 ($\overline{WR}$) Output of the write strobe signal when data is written into external data memory. • P3.7 ($\overline{RD}$) Output of the read strobe signal when data is read from external data memory.
ALE	Address latch enable output for latching the lower 8-bit address during external memory access. Two ALE pulses are activated per machine cycle except during external data memory access at which time one ALE pulse is skipped.
$\overline{PSEN}$	Program store enable output which enable the external memory output to the bus during external program memory access. Two $\overline{PSEN}$ pulses are activated per machine cycle except during external data memory access at which two $\overline{PSEN}$ pulses are skipped.
$\overline{EA}$	When $\overline{EA}$ is held at "H" level, the MSM83C154 executes instructions from internal program memory at address 0000H to 3FFFH, and executes instructions from external program memory above address 3FFFH. When $\overline{EA}$ is held at "L" level, the MSM80C154/MSM83C154 executes instructions from external program memory for all addresses.
RESET	If this pin remains "H" for at least 1 μ second, the MSM80C154/MSM83C154 is reset. Since this pin is pulled down internally, a power-on reset is achieved by simply connecting a capacitor between Vcc and this pin.
XTAL1	Oscillator inverter input pin. External clock is input through XTAL1 pin.
XTAL2	Oscillator inverter output pin.
V_{CC}	Power supply pin during both normal operation and standby operations.
V_{SS}	GND pin.

DATA MEMORY AND SPECIAL FUNCTION REGISTER LAYOUT DIAGRAM



DETAILED DIAGRAM OF DATA MEMORY (RAM)



DETAILED DIAGRAM OF SPECIAL FUNCTION REGISTERS

Direct Byte Address	Bit Address (MSB)							(LSB)	Special Function Register Symbol
	WDT	T32	SERR	IZC	P3HZ	P2HZ	P1HZ	ALF	
0F8H	FF	FE	FD	FC	FB	FA	F9	F8	IOCON
0F0H	F7	F6	F5	F4	F3	F2	F1	F0	B
0E0H	E7	E6	E5	E4	E3	E2	E1	E0	ACC
	CY	AC	F0	RS1	RS0	0V	F1	P	
0D0H	D7	D6	D5	D4	D3	D2	D1	D0	PSW
0CDH	Not Bit Addressable								TH2
0CCH	Not Bit Addressable								TL2
0CBH	Not Bit Addressable								RCAP2H
0CAH	Not Bit Addressable								RCAP2L
	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$	
0C8H	CF	CE	CD	CC	CB	CA	C9	C8	T2CON
	PCT		PT2	PS	PT1	PX1	PT0	PX0	
0B8H	BF	–	BD	BC	BB	BA	B9	B8	IP
0B0H	B7	B6	B5	B4	B3	B2	B1	B0	P3
	EA		ET2	ES	ET1	EX1	ET0	EX0	
0A8H	AF	–	AD	AC	AB	AA	A9	A8	IE
0A0H	A7	A6	A5	A4	A3	A2	A1	A0	P2
99H	Not Bit Addressable								SBUF
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
98H	9F	9E	9D	9C	9B	9A	99	98	SCON
90H	97	96	95	94	93	92	91	90	P1

Direct Byte Address	Bit Address (MSB)							(LSB)	Special Function Register Symbol
8DH	Not Bit Addressable								TH1
8CH	Not Bit Addressable								TH0
8BH	Not Bit Addressable								TL1
8AH	Not Bit Addressable								TL0
89H	Not Bit Addressable								TMOD
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
88H	8F	8E	8D	8C	8B	8A	89	88	TCON
87H	Not Bit Addressable								PCON
83H	Not Bit Addressable								DPH
82H	Not Bit Addressable								DPL
81H	Not Bit Addressable								SP
80H	87	86	85	84	83	82	81	80	P0

SPECIAL FUNCTION REGISTERS

Timer mode register (TMOD)

NAME	ADDRESS	MSB 7	6	5	4	3	2	1	LSB 0
TMOD	89H	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

BIT LOCATION	FLAG	FUNCTION
TMOD.0	M0	M1 = 0, M0 = 0: Timer/counter 0 mode setting — 8-bit timer/counter with 5-bit prescalar. M1 = 0, M0 = 1: 16-bit timer/counter. M1 = 1, M0 = 0: 8-bit timer/counter with 8-bit auto reloading.
TMOD.1	M1	M1 = 1, M0 = 1: Timer/counter 0 separated into TL0 (8-bit) timer/counter and TH0 (8-bit) timer/counter. TF0 is set by TL0 carry, and TF1 is set by TH0 carry.
TMOD.2	C/$\overline{T}$	Timer/counter 0 count clock designation control bit. XTAL1 • 2 divided by 12 clocks is the input applied to timer/counter 0 when C/$\overline{T}$ = "0". The external clock applied to the T0 pin is the input applied to timer/counter 0 when C/$\overline{T}$ = "1".
TMOD.3	GATE	When this bit is "0", the TR0 bit of TCON (timer control register) is used to control the start and stop of timer/counter 0 counting. If this bit is "1", timer/counter 0 starts counting when both the TR0 bit of TCON and $\overline{INT0}$ pin input signal are "1", and stops counting when either is changed to "0".
TMOD.4	M0	M1 = 0, M0 = 0: Timer/counter 1 mode setting — 8-bit timer/counter with 5-bit prescalar. M1 = 0, M0 = 1: 16-bit timer/counter. M1 = 1, M0 = 0: 8-bit timer/counter with 8-bit auto reloading.
TMOD.5	M1	M1 = 1, M0 = 1: Timer/counter 1 operation stopped.
TMOD.6	C/$\overline{T}$	Timer/counter 1 count clock designation control bit. XTAL1 • 2 divided by 12 clocks is the input applied to timer/counter 1 when C/$\overline{T}$ = "0". The external clock applied to the T1 pin is the input applied to timer/counter 1 when C/$\overline{T}$ = "1".
TMOD.7	GATE	When this bit is "0", the TR1 bit of TCON is used to control the start and stop of timer/counter 1 counting. If this bit is "1", timer/counter 1 starts counting when both the TR1 bit of TCON and $\overline{INT1}$ pin input signal are "1", and stops counting when either is changed to "0".

Power control register (PCON)

NAME	ADDRESS	MSB 7	6	5	4	3	2	1	LSB 0
PCON	87H	SMOD	HPD	RPD	–	GF1	GF0	PD	IDL

BIT LOCATION	FLAG	FUNCTION
PCON.0	IDL	IDLE mode set when this bit is set to "1". CPU operations are stopped when IDLE mode is set, but XTAL1•2, timer/counters 0, 1, and 2, the interrupt circuits, and serial port remain active. IDLE mode is cancelled when the CPU is reset or when an interrupt is generated.
PCON.1	PD	PD mode set when this bit is set to "1". CPU operations and XTAL1◦2 are stopped when PD mode is set. PD mode is cancelled when the CPU is reset or when an interrupt is generated.
PCON.2	GF0	General purpose bit. Testing this flag when IDLE mode is cancelled by an interrupt shows whether the interrupt is a normal interrupt or an IDLE mode release interrupt.
PCON.3	GF1	General purpose bit. Testing this flag when PD mode is cancelled by an interrupt shows whether the interrupt is a normal interrupt or a PD mode release interrupt.
PCON.4	–	Reserved bit. The output data is "1" if the bit is read.
PCON.5	RPD	Bit used to specify cancellation of CPU power down mode (IDLE or PD) by interrupt signal. Power down mode cannot be cancelled by interrupt signal if interrupt is not enabled by IE (interrupt enable register) when this bit is "0". If the interrupt flag is set to "1" by an interrupt request signal when this bit is "1" (even if interrupt is disabled), the program is executed from the next address of the power down mode setting instruction. The flag is reset to "0" by software.
PCON.6	HPD	The hard power down setting mode is enabled when this bit is set to "1". If the level of the power failure detect signal applied to the HPDI pin (pin 3.5) is changed from "1" to "0" when this bit is "1", XTAL1•2 oscillation is stopped and the system is put into hard power down mode. HPD mode is cancelled when the CPU is reset.
PCON.7	SMOD	When the timer/counter 1 carry signal is used as a clock in mode 1, 2 or 3 of the serial port, this bit has the following functions. The serial port operation clock is reduced by 1/2 when the bit is "0" for delayed processing. And when the bit is "1", the serial port operation clock is normal for faster processing.

Timer control register (TCON)

NAME	ADDRESS	MSB 7	6	5	4	3	2	1	LSB 0
TCON	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

BIT LOCATION	FLAG	FUNCTION
TCON.0	IT0	External interrupt 0 signal used in level detect mode when this bit is "0", and in trigger detect mode when "1".
TCON.1	IE0	Interrupt request flag for external interrupt 0. Bit is reset automatically when interrupt is serviced. Bit can be set and reset by software when IT0 = "1".
TCON.2	IT1	External interrupt 1 signal used in level detect mode when this bit is "0", and in trigger detect mode when "1".
TCON.3	IE1	Interrupt request flag for external interrupt 1. Bit is reset automatically when interrupt is serviced. Bit can be set and reset by software when IT1 = "1".
TCON.4	TR0	Counting start and stop control bit for timer/counter 0. Timer/counter 0 starts counting when this bit is "1", and stops counting when "0".
TCON.5	TF0	Interrupt request flag for timer interrupt 0. Bit is reset automatically when interrupt is serviced. Bit is set to "1" when carry signal is generated from timer/counter 0.
TCON.6	TR1	Counting start and stop control bit for timer/counter 1. Timer/counter 1 starts counting when this bit is "1", and stops counting when "0".
TCON.7	TF1	Interrupt request flag for timer interrupt 1. Bit is reset automatically when interrupt is serviced. Bit is set to "1" when carry signal is generated from timer/counter 1.

Serial port control register (SCON)

NAME	ADDRESS	MSB 7	6	5	4	3	2	1	LSB 0
SCON	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI

BIT LOCATION	FLAG	FUNCTION
SCON.0	RI	"End of serial port reception" interrupt request flag. This flag must be reset by software during interrupt service routine. This flag is set after the eighth bit of data has been received when in mode 0, or by the STOP bit when in any other mode. In mode 2 or 3, however, RI is not set if the RB8 data is "0" with SM2 = "1". RI is set in mode 1 if STOP bit is received when SM2 = "1".
SCON.1	TI	"End of serial port transmission" interrupt request flag. This flag must be reset by software during interrupt service routine. This flag is set after the eighth bit of data has been sent when in mode 0, or after the last bit of data has been sent when in any other mode.
SCON.2	RB8	The ninth bit of data received in mode 2 or 3 is passed to RB8. The STOP bit is applied to RB8 if SM2 = "0" when in mode 1. RB8 can not be used in mode 0.
SCON.3	TB8	The TB8 data is sent as the ninth data bit when in mode 2 or 3. Any desired data can be set in TB8 by software.
SCON.4	REN	Reception enable control bit No reception when REN = "0". Reception enabled when REN = "1".
SCON.5	SM2	If the ninth bit of received data is "0" with SM2 = "1" in mode 2 or 3, the "end of reception" signal is not set in the RI flag. Nor is the "end of reception" signal set in the RI flag if the STOP bit is not "1" when SM2 = "1" in mode 1.
SCON.6	SM1	
SCON.7	SM0	

SM0	SM1	MODE	
0	0	0	8-bit shift register I/O
0	1	1	8-bit UART variable baud rate
1	0	2	9-bit UART 1/32 XTAL1, 1/64 XTAL1 baud rate
1	1	3	9-bit UART variable baud rate

Interrupt enable register (IE)

NAME	ADDRESS	MSB 7	6	5	4	3	2	1	LSB 0
IE	0A8H	EA	–	ET2	ES	ET1	EX1	ET0	EX0

BIT LOCATION	FLAG	FUNCTION
IE.0	EX0	Interrupt control bit for external interrupt 0. Interrupt disabled when bit is "0". Interrupt enabled when bit is "1".
IE.1	ET0	Interrupt control bit for timer interrupt 0. Interrupt disabled when bit is "0". Interrupt enabled when bit is "1".
IE.2	EX1	Interrupt control bit for external interrupt 1. Interrupt disabled when bit is "0". Interrupt enabled when bit is "1".
IE.3	ET1	Interrupt control bit for timer interrupt 1. Interrupt disabled when bit is "0". Interrupt enabled when bit is "1".
IE.4	ES	Interrupt control bit for serial port. Interrupt disabled when bit is "0". Interrupt enabled when bit is "1".
IE.5	ET2	Interrupt control bit for timer interrupt 2. Interrupt disabled when bit is "0". Interrupt enabled when bit is "1".
IE.6	–	Reserved bit. The output data is "1" if the bit is read.
IE.7	EA	Overall interrupt control bit. All interrupts are disabled when bit is "0". All interrupts are controlled by IE.0 thru IE.5 when bit is "1".

Interrupt priority register (IP)

NAME	ADDRESS	MSB 7	6	5	4	3	2	1	LSB 0
IP	0B8H	PCT	–	PT2	PS	PT1	PX1	PT0	PX0

BIT LOCATION	FLAG	FUNCTION
IP.0	PX0	Interrupt priority bit for external interrupt 0. Priority is assigned when bit is "1".
IP.1	PT0	Interrupt priority bit for timer interrupt 0. Priority is assigned when bit is "1".
IP.2	PX1	Interrupt priority bit for external interrupt 1. Priority is assigned when bit is "1".
IP.3	PT1	Interrupt priority bit for timer interrupt 1. Priority is assigned when bit is "1".
IP.4	PS	Interrupt priority bit for serial port. Priority is assigned when bit is "1".
IP.5	PT2	Interrupt priority bit for timer interrupt 2. Priority is assigned when bit is "1".
IP.6	–	Reserved bit. The output data is "1" if the bit is read.
IP.7	PCT	Priority interrupt circuit control bit. The priority register contents are valid and priority assigned interrupts can be processed when this bit is "0". When the bit is "1", the priority interrupt circuit is stopped, and interrupts can only be controlled by the interrupt enable register (IE).

Program status word register (PSW)

NAME	ADDRESS	MSB 7	6	5	4	3	2	1	LSB 0
PSW	0D0H	CY	AC	F0	RS1	RS0	OV	F1	P

BIT LOCATION	FLAG	FUNCTION
PSW.0	P	Accumulator (ACC) parity indicator. "1" when the "1" bit number in the accumulator is an odd number, and "0" when an even number.
PSW.1	F1	User flag which may be set to "0" or "1" as desired by the user.
PSW.2	OV	Overflow flag which is set if the carry C6 from bit 6 of the ALU or CY is "1" as a result of an arithmetic operation. The flag is also set to "1" if the resultant product of executing a multiplication instruction (MUL AB) is greater than 0FFH, but is reset to "0" if the product is less than or equal to 0FFH.
PSW.3	RS0	RAM register bank switch
PSW.4	RS1	

RS1	RS0	BANK	RAM ADDRESS
0	0	0	00H – 07H
0	1	1	08H – 0FH
1	0	2	10H – 17H
1	1	3	18H – 1FH

PSW.5	F0	User flag which may be set to "0" or "1" as desired by the user.
PSW.6	AC	Auxiliary carry flag. This flag is set to "1" if a carry C_3 is generated from bit 3 of the ALU as a result of executing an arithmetic operation instruction. In all other cases, the flag is reset to "0".
PSW.7	CY	Main carry flag. This flag is set to "1" if a carry C_7 is generated from bit 7 of the ALU as result of executing an arithmetic operation instruction. If a carry C_7 is not generated, the flag is reset to "0".

I/O control register (IOCON)

NAME	ADDRESS	MSB 7	6	5	4	3	2	1	LSB 0
IOCON	0F8H	WDT	T32	SERR	IZC	P3HZ	P2HZ	P1HZ	ALF

BIT LOCATION	FLAG	FUNCTION
IOCON.0	ALF	If CPU power down mode (PD, HPD) is activated with this bit set to "1", the outputs from ports 0, 1, 2, and 3 are switched to floating status. When this bit is "0", ports 0, 1, 2, and 3 are in output mode.
IOCON.1	P1HZ	Port 1 becomes a high impedance input port when this bit is "1".
IOCON.2	P2HZ	Port 2 becomes a high impedance input port when this bit is "1".
IOCON.3	P3HZ	Port 3 becomes a high impedance input port when this bit is "1".
IOCON.4	IZC	The 10 kohm pull-up resistance for ports 1, 2, and 3 is switched off when this bit is "1", leaving only the 100 kohm pull-up resistance.
IOCON.5	SERR	Serial port reception error flag. This flag is set to "1" if an overrun or framing error is generated when data is received at a serial port. The flag is reset by software.
IOCON.6	T32	Timer/counters 0 and 1 are connected serially to form a 32-bit timer/counter when this bit is set to "1". TF1 of TCON is set if a carry is generated in the 32-bit timer/counter.
IOCON.7	WDT	Watchdog timer mode is set when this bit is set to "1". And if TF1 is set to "1" after watchdog timer mode has been set, the CPU is reset and the program is executed from address 0.

Timer 2 control register (T2CON)

NAME	ADDRESS	MSB 7	6	5	4	3	2	1	LSB 0
T2CON	0C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

BIT LOCATION	FLAG	FUNCTION
T2CON.0	CP/$\overline{RL2}$	Capture mode is set when TCLK + RCLK = "0" and CP/$\overline{RL2}$ = "1". 16-bit auto reload mode is set when TCLK + RCLK = "0" and CP/$\overline{RL2}$ = "0". CP/$\overline{RL2}$ is ignored when TCLK + RCLK = "1".
T2CON.1	C/$\overline{T2}$	Timer/counter 2 count clock designation control bit. The internal clocks (XTAL1 • 2 ÷ 12, XTAL1 • 2 ÷ 2) are used when this bit is "0", and the external clock applied to the T2 pin is passed to timer/counter 2 when the bit is "1".
T2CON.2	TR2	Timer/counter 2 counting start and stop control bit. Timer/counter 2 commences counting when this bit is "1" and stops counting when "0".
T2CON.3	EXEN2	T2EX timer/counter 2 external control signal control bit. Input of the T2EX signal is disabled when this bit is "0", and enabled when "1".
T2CON.4	TCLK	Serial port transmit circuit drive clock control bit. Timer/counter 2 is switched to baud rate generator mode when this bit is "1", and the timer/counter 2 carry signal becomes the serial port transmit clock. Note, however, that the serial ports can only use the timer/counter 2 carry signal in serial port modes 1 and 3.
T2CON.5	RCLK	Serial port receive circuit drive clock control bit. Timer/counter 2 is switched to baud rate generator mode when this bit is "1", and the timer/counter 2 carry signal becomes the serial port receive clock. Note, however, that the serial ports can only use the timer/counter 2 carry signal in serial port modes 1 and 3.
T2CON.6	EXF2	Timer/counter 2 external flag. This bit is set to "1" when the T2EX timer/counter 2 external control signal level is changed from "1" to "0" while EXEN2 = "1". This flag serves as the timer interrupt 2 request signal. If an interrupt is generated, EXF2 must be reset to "0" by software.
T2CON.7	TF2	Timer/counter 2 carry flag. This bit is set to "1" by a carry signal when timer/counter 2 is in 16-bit auto reload mode or in capture mode. This flag serves as the timer interrupt 2 request signal. If an interrupt is generated, TF2 must be reset to "0" by software.

LIST OF INSTRUCTIONS

LIST OF INSTRUCTION SYMBOLS

A : Accumulator
AB : Register pair
AC : Auxiliary carry flag
B : Arithmetic operation register
C : Carry flag
DPTR : Data pointer
PC : Program counter
Rr : Register indicator (r = 0 ~ 7)
SP : Stack pointer
AND : Logical product
OR : Logical sum
XOR : Exclusive OR
+ : Addition
– : Subtraction
X : Multiplication
/ : Division
(X) : Denotes the contents of X
((X)) : Denotes the contents of address determined by the contents of X
: Denotes the immediate data
@ : Denotes the indirect address
= : Equality
≠ : Non equality
← : Substitution
→ : Substitution
– : Negation
< : Smaller than
> : Larger than
bit address : RAM and the special function register bit specifier address ($b_0 \sim b_7$)
code address : Absolute address ($A_0 \sim A_{15}$)
data : Immediate data ($I_0 \sim I_7$)
relative offset: Relative jump address offset value ($R_0 \sim R_7$)
direct address: RAM and the special function register byte specifier address ($a_0 \sim a_7$)

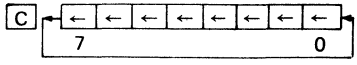
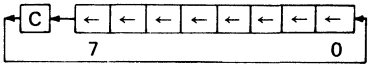
MSM80C154/MSM83C154 INSTRUCTION TABLE

H \ L	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111
0 0000	NOP	AJMP address 11 (Page 0)	LJMP address 16	RR A	INC A	INC direct	INC @R0	INC @R1
1 0001	JBC bit, rel	ACALL address 11 (Page 0)	LCALL address 16	RRC A	DEC A	DEC direct	DEC @R0	DEC @R1
2 0010	JB bit, rel	AJMP address 11 (Page 1)	RET	RL A	ADD A, #data	ADD A, direct	ADD A, @R0	ADD A, @R1
3 0011	JNB bit, rel	ACALL address 11 (Page 1)	RETI	RLC A	ADDC A, #data	ADDC A, direct	ADDC A, @R0	ADDC A, @R1
4 0100	JC bit, rel	AJMP address 11 (Page 2)	ORL direct, A	ORL direct, #data	ORL A, #data	ORL A, direct	ORL A, @R0	ORL A, @R1
5 0101	JNC rel	ACALL address 11 (Page 2)	ANL direct, A	ANL direct, #data	ANL A, #data	ANL A, direct	ANL A, @R0	ANL A, @R1
6 0110	JZ rel	AJMP address 11 (Page 3)	XRL direct, A	XRL direct, #data	XRL A, #data	XRL A, direct	XRL A, @R0	XRL A, @R1
7 0111	JNZ rel	ACALL address 11 (Page 3)	ORL C, bit	JMP @A+DPTR	MOV A, #data	MOV direct, #data	MOV @R0, #data	MOV @R1, #data
8 1000	SJMP rel	AJMP address 11 (Page 4)	ANL C, bit	MOVC A, @A+PC	DIV AB	MOV direct 1, direct 2	MOV direct, @R0	MOV direct, @R1
9 1001	MOV DPTR #data 16	ACALL address 11 (Page 4)	MOV bit, C	MOVC A, @A+DPTR	SUBB A, #data	SUBB A, direct	SUBB A, @R0	SUBB A, @R1
A 1010	ORAL C,bit	AJMP address 11 (Page 5)	MOV C, bit	INC DPTR	MUL AB		MOV @R0, direct	MOV @R1, direct
B 1011	ANL C,bit	ACALL address 11 (Page 5)	CPL bit	CPL C	CJNE A, #data, rel	CJNE A, direct, rel	CJNE @R0, #data, rel	CJNE @R1, #data, rel
C 1100	PUSH direct	AJMP address 11 (Page 6)	CLR bit	CLR C	SWAP A	XCH A, direct	XCH A, @R0	XCH A, @R1
D 1101	POP direct	ACALL address 11 (Page 6)	SETB bit	SETB C	DA A	DJNZ direct, rel	XCHD A, @R0	XCH A, @R1
E 1110	MOVX A, @DPTR	AJMP address 11 (Page 7)	MOVX A, @R0	MOVX A, @R1	CLR A	MOV A, direct	MOV A, @R0	MOV A, @R1
F 1111	MOVX @DPTR, A	ACALL address 11 (Page 7)	MOVX @R0, A	MOVX @R1, A	CPL A	MOV direct, A	MOV @R0, A	MOV @R1, A

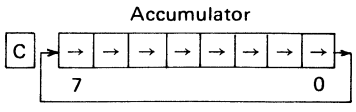
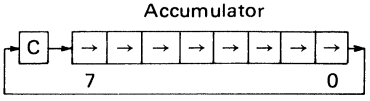
2 BYTE		3 BYTE
	MNEMONIC	
2 CYCLE		4 CYCLE

L / H	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
0 0000	INC R0	INC R1	INC R2	INC R3	INC R4	INC R5	INC R6	INC R7
1 0001	DEC R0	DEC R1	DEC R2	DEC R3	DEC R4	DEC R5	DEC R6	DEC R7
2 0010	ADD A, R0	ADD A, R1	ADD A, R2	ADD A, R3	ADD A, R4	ADD A, R5	ADD A, R6	ADD A, R7
3 0011	ADDC A, R0	ADDC A, R1	ADDC A, R2	ADDC A, R3	ADDC A, R4	ADDC A, R5	ADDC A, R6	ADDC A, R7
4 0100	ORL A, R0	ORL A, R1	ORL A, R2	ORL A, R3	ORL A, R4	ORL A, R5	ORL A, R6	ORL A, R7
5 0101	ANL A, R0	ANL A, R1	ANL A, R2	ANL A, R3	ANL A, R4	ANL A, R5	ANL A, R6	ANL A, R7
6 0110	XRL A, R0	XRL A, R1	XRL A, R2	XRL A, R3	XRL A, R4	XRL A, R5	XRL A, R6	XRL A, R7
7 0111	MOV R0, #data	MOV R1, #data	MOV R2, #data	MOV R3, #data	MOV R4, #data	MOV R5, #data	MOV R6, #data	MOV R7, #data
8 1000	MOV direct, R0	MOV direct, R1	MOV direct, R2	MOV direct, R3	MOV direct, R4	MOV direct, R5	MOV direct, R6	MOV direct, R7
9 1001	SUBB A, R0	SUBB A, R1	SUBB A, R2	SUBB A, R3	SUBB A, R4	SUBB A, R5	SUBB A, R6	SUBB A, R7
A 1010	MOV R0, direct	MOV R1, direct	MOV R2, direct	MOV R3, direct	MOV R4, direct	MOV R5, direct	MOV R6, direct	MOV R7, direct
B 1011	CJNE R0, #data, rel	CJNE R1, #data, rel	CJNE R2, #data, rel	CJNE R3, #data, rel	CJNE R4, #data, rel	CJNE R5, #data, rel	CJNE R6, #data, rel	CJNE R7, #data, rel
C 1100	XCH A, R0	XCH A, R1	XCH A, R2	XCH A, R3	XCH A, R4	XCH A, R5	XCH A, R6	XCH A, R7
D 1101	DJNZ R0 rel	DJNZ R1 rel	DJNZ R2, rel	DJNZ R3, rel	DJNZ R4, rel	DJNE R5, rel	DJNE R6, rel	DJNE R7, rel
E 1110	MOV A, R0	MOV A, R1	MOV A, R2	MOV A, R3	MOV A, R4	MOV A, R5	MOV A, R6	MOV A, R7
F 1111	MOV R0, A	MOV R1, A	MOV R2, A	MOV R3, A	MOV R4, A	MOV R5, A	MOV R6, A	MOV R7, A

INSTRUCTION SET DETAILS

Type	Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Bytes	Cycles	Description
Airthmetic operation instructions	ADD A, Rr	0 0 1 0 1 r_2 r_1 r_0	1	1	(AC), (0V), (C), (A) ← (A)+(Rr)
	ADD A, direct	0 0 1 0 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(AC), (0V), (C), (A) ← (A)+(direct address)
	ADD A, @Rr	0 0 1 0 0 1 1 r_0	1	1	(AC), (0V), (C), (A) ← (A)+((Rr))
	ADD A, #data	0 0 1 0 0 1 0 0 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	2	1	(AC), (0V), (C), (A) ← (A)+#data
	ADDC A, Rr	0 0 1 1 1 r_2 r_1 r_0	1	1	(AC), (0V), (C), (A) ← (A)+(C)+(Rr)
	ADDC A, direct	0 0 1 1 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(AC), (0V), (C), (A) ← (A)+(C)+(direct address)
	ADDC A, @Rr	0 0 1 1 0 1 1 r_0	1	1	(AC), (0V), (C), (A) ← (A)+(C)+((Rr))
	ADDC A, #data	0 0 1 1 0 1 0 0 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	2	1	(AC), (0V), (C), (A) ← (A)+(C)+#data
	SUBB A, Rr	1 0 0 1 1 r_2 r_1 r_0	1	1	(AC), (0V), (C), (A) ← (A)−((C))+((Rr))
	SUBB A, direct	1 0 0 1 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(AC), (0V), (C), (A) ← (A)−((C)+(direct address))
	SUBB A, @Rr	1 0 0 1 0 1 1 r_0	1	1	(AC), (0V), (C), (A) ← (A)−((C)+((Rr))
	SUBB A, #data	1 0 0 1 0 1 0 0 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	2	1	(AC), (0V), (C), (A) ← (A)−((C)+#data)
	MUL AB	1 0 1 0 0 1 0 0	1	4	(AB) ← (A) x (B)
	DIV AB	1 0 0 0 0 1 0 0	1	4	(A) quotient, (B) remainder ← (A)/(B)
	DA A	1 1 0 1 0 1 0 0	1	1	When the contents of accumulator bits 0 thru 3 are greater than 9, or when auxiliary carry (AC) is 1, 6 is added to bits 0 thru 3. Bits 4 thru 7 are then examined, and when bits 4 thru 7 following compensation of lower bits 0 thru 3 is greater than 9, or when carry (C) is 1, 6 is added to bits 4 thru 7. As a result, the carry flag can be set, but cannot be cleared.
Accumulation operation instructions	CLR A	1 1 1 0 0 1 0 0	1	1	(A) ← 0
	CPL A	1 1 1 1 0 1 0 0	1	1	(A) ← $\overline{(A)}$
	RL A	0 0 1 0 0 0 1 1	1	1	Accumulator 
	RLC A	0 0 1 1 0 0 1 1	1	1	Accumulator 

INSTRUCTION SET DETAILS (CONT.)

Type	Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Bytes	Cycles	Description
Accumulator operation instructions	RR A	0 0 0 0 0 0 1 1	1	1	
	RRC A	0 0 0 1 0 0 1 1	1	1	
	SWAP A	1 1 0 0 0 1 0 0	1	1	$(A_4 \sim _7) \rightleftharpoons (A_0 \sim _3)$
Increment/decrement	INC A	0 0 0 0 0 1 0 0	1	1	(A) ← (A)+1
	INC Rr	0 0 0 0 1 r_2 r_1 r_0	1	1	(Rr) ← (Rr)+1
	INC direct	0 0 0 0 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(direct address) ← (direct address)+1
	INC @Rr	0 0 0 0 0 1 1 r_0	1	1	((Rr)) ← ((Rr))+1
	INC DPTR	1 0 1 0 0 0 1 1	1	2	(DPTR) ← (DPTR)+1
	DEC A	0 0 0 1 0 1 0 0	1	1	(A) ← (A)−1
	DEC Rr	0 0 0 1 1 r_2 r_1 r_0	1	1	(Rr) ← (Rr)−1
	DEC direct	0 0 0 1 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(direct address) ← (direct address)−1
	DEC @Rr	0 0 0 1 0 1 1 r_0	1	1	((Rr)) ← ((Rr))−1
Logical operation instructions	ANL A, Rr	0 1 0 1 1 r_2 r_1 r_0	1	1	(A) ← (A) AND (Rr)
	ANL A, direct	0 1 0 1 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(A) ← (A) AND (direct address)
	ANL A, @Rr	0 1 0 1 0 1 1 r_0	1	1	(A) ← (A) AND ((Rr))
	ANL A, #data	0 1 0 1 0 1 0 0 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	2	1	(A) ← (A) AND #data
	ANL direct, A	0 1 0 1 0 0 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(direct address) ← (direct address) AND (A)
	ANL direct, #data	0 1 0 1 0 0 1 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	3	2	(direct address) ← (direct address) AND #data
	ORL A, Rr	0 1 0 0 1 r_2 r_1 r_0	1	1	(A) ← (A) OR (Rr)
	ORL A, direct	0 1 0 0 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(A) ← (A) OR (direct address)
	ORL A, @Rr	0 1 0 0 0 1 1 r_0	1	1	(A) ← (A) OR ((Rr))
	ORL A, #data	0 1 0 0 0 1 0 0 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	2	1	(A) ← (A) OR #data

INSTRUCTION SET DETAILS (CONT.)

Type	Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Bytes	Cycles	Description
Logical operation instructions	ORL direct, A	0 1 0 0 0 0 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(direct address) ← (direct address) OR (A)
	ORL direct, #data	0 1 0 0 0 0 1 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	3	2	(direct address) ← (direct address) OR #data
	XRL A, Rr	0 1 1 0 1 r_2 r_1 r_0	1	1	(A) ← (A) XOR (Rr)
	XRL A, direct	0 1 1 0 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(A) ← (A) XOR (direct address)
	XRL A, @Rr	0 1 1 0 0 1 1 r_0	1	1	(A) ← (A) XOR ((Rr))
	XRL A, #data	0 1 1 0 0 1 0 0 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	2	1	(A) ← (A) XOR #data
	XRL direct, A	0 1 1 0 0 0 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(direct address) ← (direct address) XOR (A)
	XRL direct, #data	0 1 1 0 0 0 1 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	3	2	(direct address) ← (direct address) XOR #data
Immediate data setting instructions	MOV A, #data	0 1 1 1 0 1 0 0 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	2	1	(A) ← #data
	MOV Rr, #data	0 1 1 1 1 r_2 r_1 r_0 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	2	1	(Rr) ← #data
	MOV direct, #data	0 1 1 1 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	3	2	(direct address) ← #data
	MOV @Rr, #data	0 1 1 1 0 1 1 r_0 I_7 I_5 I_5 I_4 I_3 I_2 I_1 I_0	2	1	(Rr)) ← #data
	MOV DPTR, #data 16	1 0 0 1 0 0 0 0 I_{15} I_{14} I_{13} I_{12} I_{11} I_{10} I_9 I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0	3	2	(DPTR) ← #data 16
Carry flag operation instructions	CLR C	1 1 0 0 0 0 1 1	1	1	(C) ← 0
	SETB C	1 1 0 1 0 0 1 1	1	1	(C) ← 1
	CPL C	1 0 1 1 0 0 1 1	1	1	(C) ← $\overline{(C)}$
	ANL C, bit	1 0 0 0 0 0 1 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	2	2	(C) ← (C) AND (bit address)
	ANL C,/bit	1 0 1 1 0 0 0 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	2	2	(C) ← (C) AND $\overline{\text{(bit address)}}$
	ORL C, bit	0 1 1 1 0 0 1 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	2	2	(C) ← (C) OR (bit address)
	ORL C,/bit	1 0 1 0 0 0 0 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	2	2	(C) ← (C) OR $\overline{\text{(bit address)}}$
	MOV C, bit	1 0 1 0 0 0 1 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	2	1	(C) ← (bit address)

INSTRUCTION SET DETAILS (CONT.)

Type	Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Bytes	Cycles	Description
Bit operation instructions	MOV bit, C	1 0 0 1 0 0 1 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	2	2	(bit address) ← (C)
	SETB bit	1 1 0 1 0 0 1 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	2	1	(bit address) ← 1
	CLR bit	1 1 0 0 0 0 1 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	2	1	(bit address) ← 0
	CPL bit	1 0 1 1 0 0 1 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	2	1	(bit address) ← $\overline{\text{(bit address)}}$
Data transfer instructions	MOV A, Rr	1 1 1 0 1 r_2 r_1 r_0	1	1	(A) ← (Rr)
	MOV A, direct	1 1 1 0 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(A) ← (direct address)
	MOV A, @Rr	1 1 1 0 0 1 1 r_0	1	1	(A) ← ((Rr))
	MOV Rr, A	1 1 1 1 1 r_2 r_1 r_0	1	1	(Rr) ← (A)
	MOV Rr, direct	1 0 1 0 1 r_2 r_1 r_0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2	(Rr) ← (direct address)
	MOV direct, A	1 1 1 1 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(direct address) ← (A)
	MOV direct, Rr	1 0 0 0 1 r_2 r_1 r_0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2	(direct address) ← (Rr)
	MOV direct, @Rr	1 0 0 0 0 1 1 r_0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2	(direct address) ← ((Rr))
	MOV @Rr, A	1 1 1 1 0 1 1 r_0	1	1	((Rr)) ← (A)
	MOV @Rr, direct	1 0 1 0 0 1 1 r_0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2	((Rr)) ← (direct address))
Constant code instructions	MOVC A, @A+DPTR	1 0 0 1 0 0 1 1	1	2	(A) ← ((A)+(DPTR))
	MOVC A,@A+PC	1 0 0 0 0 0 1 1	1	2	(PC) ← (PC) + 1 (A) ← ((A) + (PC))
Data exchange instructions	XCH A, Rr	1 1 0 0 1 r_2 r_1 r_0	1	1	(A) ⇌ (Rr)
	XCH A, direct	1 1 0 0 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	1	(A) ⇌ (direct address)
	XCH A, @Rr	1 1 0 0 0 1 1 r_0	1	1	(A) ⇌ ((Rr))
	XCHD A, @Rr	1 1 0 1 0 1 1 r_0	1	1	($A_0 \sim_3$) ⇌ (($Rr_0 \sim_3$))

INSTRUCTION SET DETAILS (CONT.)

Type	Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Bytes	Cycles	Description
Subroutine instructions	PUSH direct	1 1 0 0 0 0 0 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2	(SP) ← (SP)+1 ((SP)) ← (direct address)
	POP direct	1 1 0 1 0 0 0 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2	(direct address) ← ((SP)) (SP) ← (SP)−1
	ACALL addr 11	A_{10} A_9 A_8 1 0 0 0 1 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0	2	2	(PC) ← (PC)+2 (SP) ← (SP)+1 ((SP)) ← (PC_0~7) (SP) ← (SP)+1 ((SP)) ← (PC_8~15) (PC_0~10) ← A_0~10
	LCALL addr 16	0 0 0 1 0 0 1 0 A_{15} A_{14} A_{13} A_{12} A_{11} A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0	3	2	(PC) ← (PC)+3 (SP) ← (SP)+1 ((SP)) ← (PC_0~7) (SP) ← (SP)+1 ((SP)) ← (PC_8~15) (PC_0~15) ← A_0~15
	RET	0 0 1 0 0 0 1 0	1	2	(PC_8~15) ← ((SP)) (SP) ← (SP)−1 (PC_0~7) ← ((SP)) (SP) ← (SP)−1
	RETI	0 0 1 1 0 0 1 0	1	2	(PC_8~15) ← ((SP)) (SP) ← (SP)−1 (PC_0~7) ← ((SP)) (SP) ← (SP)−1
Jump instructions	AJMP addr 11	A_{10} A_9 A_8 0 0 0 0 1 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0	2	2	(PC) ← (PC)+2 (PC_0~10) ← A_0~10
	LJMP addr 16	0 0 0 0 0 0 1 0 A_{15} A_{14} A_{13} A_{12} A_{11} A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0	3	2	(PC_0~15) ← A_0~15
	SJMP rel	1 0 0 0 0 0 0 0 R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0	2	2	(PC) ← (PC)+2 (PC) ← (PC)+relative offset
	JMP @A+DPTR	0 1 1 1 0 0 1 1	1	2	(PC) ← (A)+(DPTR)

Branch instructions	CJNE A, direct, rel	1 0 1 1 0 1 0 1
		a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0
		R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0
			3	2	(PC) ← (PC)+3
					IF (A) ≠ (direct address)
					THEN
					(PC) ← (PC)+relative offset
					IF (A) < (direct address)
					THEN
					(C) ← 1
					ELSE
					(C) ← 0

	CJNE A, #data, rel	1 0 1 1 0 1 0 0
		I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0
		R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0
			3	2	(PC) ← (PC)+3
					IF (A) ≠ #data
					THEN
					(PC) ← (PC)+relative offset
					IF (A) < #data
					THEN
					(C) ← 1
					ELSE
					(C) ← 0

INSTRUCTION SET DETAILS (CONT.)

Type	Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Bytes	Cycle	Description

Branch instructions	CJNE Rr, #data, rel	1 0 1 1 1 r_2 r_1 r_0
		I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0
		R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0
			3	2	(PC) ← (PC)+3
					IF ((Rr)) ≠ #data
					THEN
					(PC) ← (PC)+relative offset
					IF ((Rr)) < #data
					THEN
					(C) ← 1
					ELSE
					(C) ← 0

	CJNE @Rr, #data, rel	1 0 1 1 0 1 1 r_0
		I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0
		R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0
			3	2	(PC) ← (PC)+3
					IF ((Rr)) ≠ #data
					THEN
					(PC) ← (PC)+relative offset
					IF ((Rr)) < #data
					THEN
					(C) ← 1
					ELSE
					(C) ← 0

	DJNZ Rr, rel	1 1 0 1 1 r_2 r_1 r_0 R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0	2	2	(PC) ← (PC)+2 (Rr) ← (Rr)−1 IF (Rr) ≠ 0 THEN (PC) ← (PC)+relative offset
	DJNZ direct, rel	1 1 0 1 0 1 0 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0	3	2	(PC) ← (PC)+3 (direct address) ← (direct address)−1 IF (direct address) ≠ 0 THEN (PC) ← (PC)+relative offset
	JZ rel	0 1 1 0 0 0 0 0 R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0	2	2	(PC) ← (PC)+2 IF (A) = 0 THEN (PC) ← (PC)+relative offset
	JNZ rel	0 1 1 1 0 0 0 0 R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0	2	2	(PC) ← (PC)+2 IF (A) ≠ 0 THEN (PC) ← (PC)+relative offset
	JC rel	0 1 0 0 0 0 0 0 R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0	2	2	(PC) ← (PC)+2 IF (C) = 1 THEN (PC) ← (PC)+relative offset
	JNC rel	0 1 0 1 0 0 0 0 R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0	2	2	(PC) ← (PC)+2 IF (C) = 0 THEN (PC) ← (PC)+relative offset
	JB bit, rel	0 0 1 0 0 0 0 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0 R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0	3	2	(PC) ← (PC)+3 IF (bit address) = 1 THEN (PC) ← (PC)+relative offset
	JNB bit, rel	0 0 1 1 0 0 0 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0 R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0	3	2	(PC) ← (PC)+3 IF (bit address) = 0 THEN (PC) ← (PC)+relative offset

INSTRUCTION SET DETAILS (CONT.)

Type	Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Bytes	Cycle	Description
Branch instructions	JBC bit, rel	0 0 0 1 0 0 0 0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0 R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0	3	2	(PC) ← (PC)+3 IF (bit address) = 1 THEN (bit address) ← 0 (PC) ← (PC)+relative offset
External memory instructions	MOVX A, @Rr	1 1 1 0 0 0 1 r_0	1	2	(A) ← ((Rr)) EXTERNAL RAM
	MOVX A, @DPTR	1 1 1 0 0 0 0 0	1	2	(A) ← ((DPTR)) EXTERNAL RAM
	MOVX @Rr, A	1 1 1 1 0 0 1 r_0	1	2	(Rr) ← (A) EXTERNAL RAM
	MOVX @DPTR, A	1 1 1 1 0 0 0 0	1	2	((DPTP)) ← (A) EXTERNAL RAM
Other instructions	NOP	0 0 0 0 0 0 0 0	1	1	(PC) ← (PC)+1

ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

Parameter	Symbol	Conditions	Rating	Unit
Supply voltage	Vcc	Ta = 25 °C	−0.5 ~ 7	V
Input voltage	V_I	Ta = 25 °C	−0.5 ~ Vcc + 0.5	V
Storage temperature	Tstg		−55 ~ + 150	°C

Operational Range

- MSM80C154/83C154 ...DC to 12 MHz, VCC = ±20%
- MSM80C154-1/83C154-1 ...DC to 16 MHz, VCC = ±5%

Parameter	Symbol	Conditions	Rating	Unit
Supply voltage	Vcc	*1 fosc = DC-16 MHz	2.5 ~ 6	V
Memory hold voltage	Vcc		2 ~ 6	V
Ambient temperature	Ta		−40 ~ +85 (0 ~ 12 MHz) −20 ~ +70 (12 ~ 16 MHz)	°C

*1: 2.5 V ≤ Vcc < 4 V DC characteristics will be specified elsewhere.

DC Characteristics

(V_{CC} = 5V±10%, Ta = −40 to +85°C)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit	Measuring circuit
Input Low Voltage	V_{IL}		−0.5		0.2 V_{CC}−0.1	V	1
Input High Voltage	V_{IH}	Except XTAL1 and RESET	0.2 V_{CC} + 0.9		V_{CC} + 0.5	V	
Input High Voltage	V_{IH1}	XTAL1 and RESET	0.7 V_{CC}		V_{CC} + 0.5	V	
Output Low Voltage (PORT 1, 2, 3)	V_{OL}	I_{OL} = 1.6 mA			0.45	V	
Output Low Voltage (PORT 0, ALE, PSEN)	V_{OL1}	I_{OL} = 3.2 mA			0.45	V	
Output High Voltage (PORT 1, 2, 3)	V_{OH}	I_{OH} = −60 μA V_{CC} = 5 V ± 10%	2.4			V	
		I_{OH} = −30 μA	0.75 V_{CC}			V	
		I_{OH} = −10 μA	0.9 V_{CC}			V	
Output High Voltage (PORT 0, ALE, PSEN)	V_{OH1}	I_{OH} = −400 μA V_{CC} = 5 V ± 10%	2.4			V	
		I_{OH} = −150 μA	0.75 V_{CC}			V	
		I_{OH} = −40 μA	0.9 V_{CC}			V	
Logical 0 Input Current/ Logical 1 Output Current (PORT 1, 2, 3)	I_{IL}/I_{OH}	V_I=0.45V / V_O=0.45V	−10		−200	μA	2
Logical 1 to 0 Transition Current (PORT 1, 2, 3)	I_{TL}	V_I = 2.0 V			−500	μA	
Input Leakage Current (PORT 0 floating, EA)	I_{LI}	$V_{SS} < V_I < V_{CC}$			± 10	μA	3
RESET Pulldown Resistor	R_{RST}		20	40	125	KΩ	2
Pin Capacitance	C_{IO}	T_A = 25°C, f = 1 MHz 5 V (except XTAL1)			10	pF	
Power Down Current	I_{PD}	V_{CC} = 2 ~ 6 V		1	50	μA	4

Maximum Power Supply Current
Normal Operation I_{CC} (mA)

V_{CC}	4 V	5 V	6 V
Freq			
0.5 MHz	1.6	2.2	3
3.5 MHz	4.3	5.7	7.5
8 MHz	8.3	11	14
12 MHz	12	16	20

Maximum Power Supply Current
Idle Mode I_{CC} (mA)

V_{CC}	4 V	5 V	6 V
Freq			
0.5 MHz	0.6	0.9	1.2
3.5 MHz	1.1	1.6	2.2
8 MHz	1.8	2.7	3.7
12 MHz	2.5	3.7	5

*1: **2.5 V** ≤ V_{CC} < 4 V DC characteristics will be specified elsewhere.

Maximum Power Supply Current
Normal Operation I_{CC} (mA)

V_{CC}	4.5 V	5 V	5.5 V
Freq.			
1.2 MHz	2.0	2.3	2.6
8 MHz	10	11	12.5
12 MHz	14	16	18

V_{CC}	4.75 V	5 V	5.25 V
Freq.			
16 MHz	18	20	23

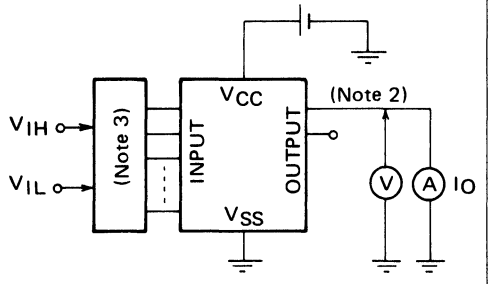
Maximum Power Supply Current
Idle Mode I_{CC} (mA)

V_{CC}	4.5 V	5 V	5.5 V
Freq.			
1.2 MHz	1.4	1.5	1.6
8 MHz	2.3	2.7	3.2
12 MHz	3.0	3.7	5.0

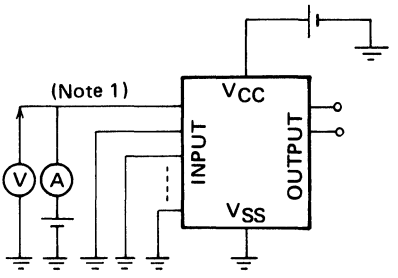
V_{CC}	4.75 V	5 V	5.25 V
Freq.			
16 MHz	4.0	5.0	6.0

Measuring Circuits

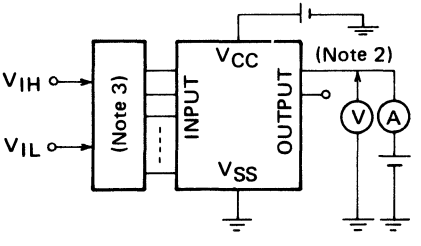
1



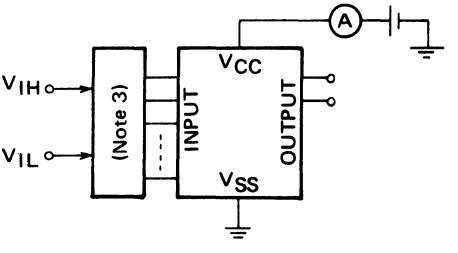
2



3



4



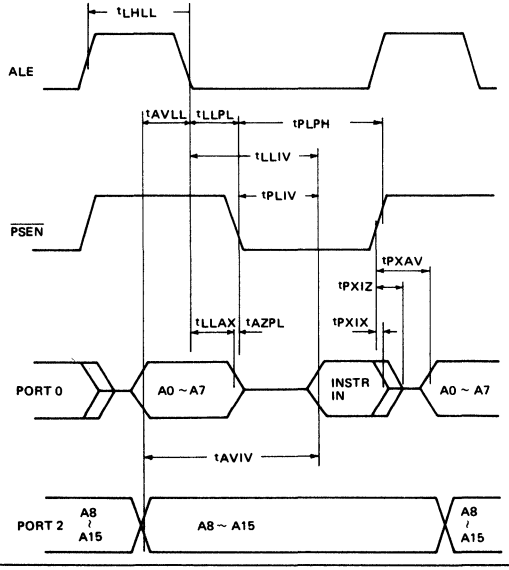
Note 1. Repeated for specified input pins.
2. Repeated for specified output pins.
3. Input logic for specified status.

External Program Memory Access AC Characteristics

(V_{CC} = 5 V ± 20%, V_{SS} = 0 V, XTAL1·2 = 12 MHz, Ta = −40 °C to 85 °C
V_{CC} = 5 V ± 5%, V_{SS} = 0 V, 12 MHz < XTAL1·2 ≤ 16 MHz, Ta = − 20°C to 70°C
PORT 0, ALE, and $\overline{PSEN}$ connected with 100 pF load, other connected with 80 pF load)

Parameter	Symbol	Ratings				Unit
		16 MHz clock		Variable clock from DC to 16 MHz		
		Min.	Max.	Min.	Max.	
XTAL1·2 Oscillator Period	tCLCL	62.5		62.5		ns
ALE Pulse Width	tLHLL	85		2tCLCL-40		ns
Address Valid to ALE Low	tAVLL	18.5		1tCLCL-44		ns
Address Hold After ALE Low	tLLAX	27.5		1tCLCL-35		ns
ALE Low to Valid Instr In	tLLIV		150		4tCLCL-100	ns
ALE Low to $\overline{PSEN}$ Low	tLLPL	32.5		1tCLCL-30		ns
$\overline{PSEN}$ Pulse Width	tPLPH	152.5		3tCLCL-35		ns
$\overline{PSEN}$ Low to Valid Instr In	tPLIV		82.5		3tCLCL-105	ns
Input Instr Hold After $\overline{PSEN}$	tPXIX	0		0		ns
Input Instr Float After $\overline{PSEN}$	tPXIZ		42.5		1tCLCL-20	ns
$\overline{PSEN}$ to Address Valid	tPXAV	42.5		1tCLCL-20		ns
Address to Valid Instr In	tAVIV		207.5		5tCLCL-105	ns
Address Float to $\overline{PSEN}$ Low	tAZPL	0		0		ns

External program memory read cycle



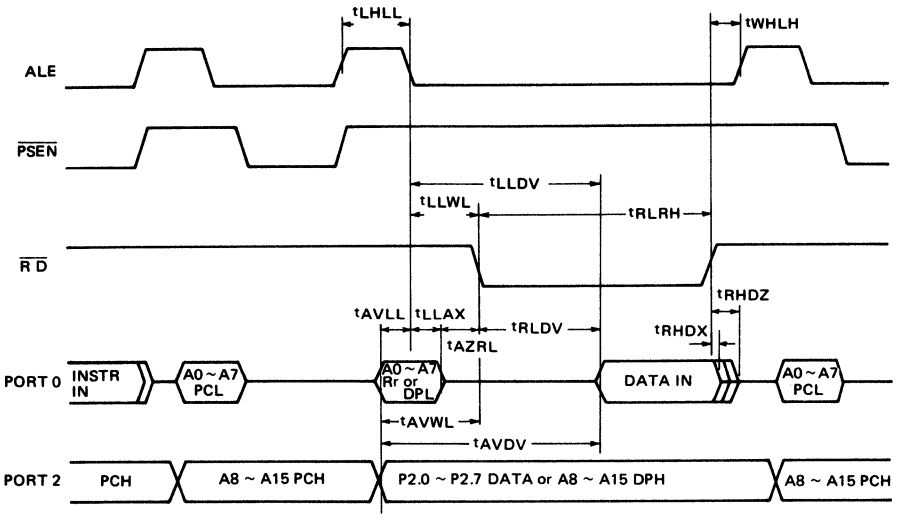
External Program Memory Access AC Characteristics

(V_{CC} = 5 V ± 20%, V_{SS} = 0 V, XTAL1·2 = 12 MHz, Ta = –40 °C to 85 °C
V_{CC} = 5 V ± 5%, V_{SS} = 0 V, 12 MHz < XTAL1·2 ≤ 16 MHz, Ta = – 20°C to 70°C

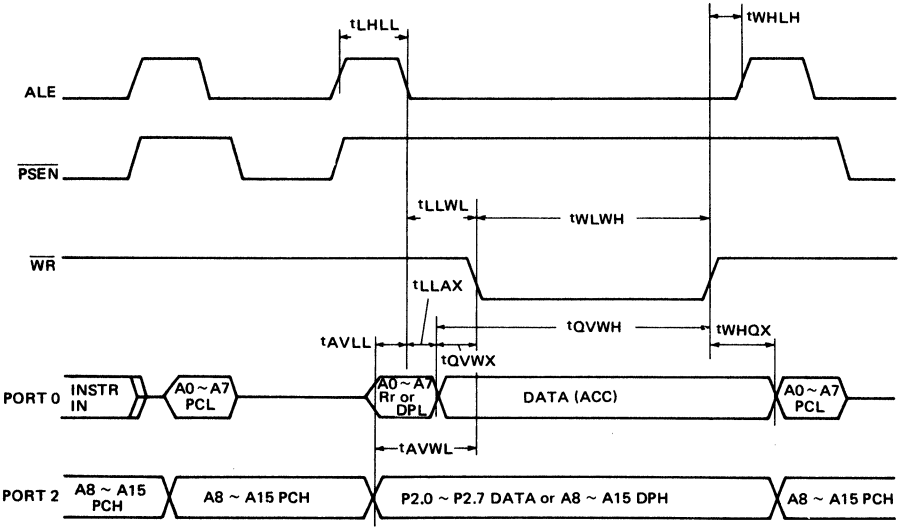
PORT 0, ALE, and $\overline{PSEN}$ connected with 100 pF load, other connected with 80 pF load)

Parameter	Symbol	Ratings				Unit
		16 MHz clock		Variable clock from DC to 16 MHz		
		Min.	Max.	Min.	Max.	
XTAL1·2 Oscillator Period	tCLCL	62.5		62.5		ns
ALE Pulse Width	tLHLL	85		2tCLCL-40		ns
Address Valid to ALE Low	tAVLL	18.5		1tCLCL-44		ns
Address Hold After ALE Low	tLLAX	27.5		1tCLCL-35		ns
$\overline{RD}$ Pulse Width	tRLRH	275		6tCLCL-100		
$\overline{WR}$ Pulse Width	tWLWH	275		6tCLCL-100		ns
$\overline{RD}$ Low to Valid Data In	tRLDV		207.5		5tCLCL-105	ns
Data Hold After $\overline{RD}$	tRHDX	0		0		ns
Data Float After $\overline{RD}$	tRHDZ		55		2tCLCL-70	ns
ALE Low to Valid Data In	tLLDV		400		8tCLCL-100	ns
Address to Valid Data In	tAVDV		457.5		9tCLCL-105	ns
ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	tLLWL	147.5	227.5	3tCLCL-40	3tCLCL+40	ns
Address to $\overline{RD}$ or $\overline{WR}$ Low	tAVWL	180		4tCLCL-70		ns
Data Valid to $\overline{WR}$ Transition	tQVWX	22.5		1tCLCL-40		ns
Data Valid to $\overline{WR}$ High	tQVWH	332.5		7tCLCL-105		ns
Data Hold After $\overline{WR}$	tWHQX	75		2tCLCL-50		ns
Address Float to $\overline{RD}$ Low	tAZRL		0		0	ns
$\overline{RD}$ or $\overline{WR}$ High to ALE High	tWHLH	32.5	102.5	1tCLCL-30	1tCLCL+40	ns

External data memory read cycle



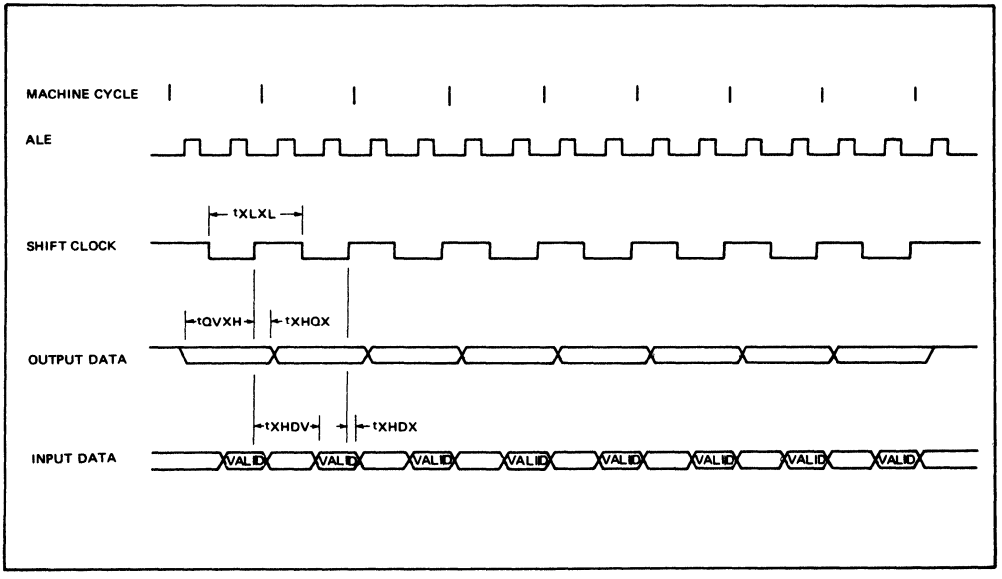
External data memory write cycle



Serial Port (I/O Extension Mode) AC Characteristics

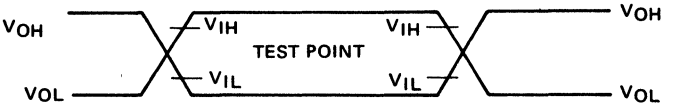
V_{CC} = 5 V ±20%, V_{SS} = 0 V, XTAL1·2 = 12 MHz, Ta = –40 °C to 85 °C
V_{CC} = 5 V ± 5%, V_{SS} = 0 V, 12 MHz < XTAL1·2 ≤ 16 MHz, Ta = – 20°C to 70°C

Parameter	Symbol	Min.	Max.	Unit
Serial Port Clock Cycle Time	tXLXL	12tCLCL		ns
Output Data Setup to Clock Rising Edge	tQVXH	10tCLCL-133		ns
Output Data Hold After Clock Rising Edge	tXHQX	2tCLCL-75		ns
Input Data Hold After Clock Rising Edge	tXHDX	0		ns
Clock Rising Edge to Input Data Valid	tXHDV		10tCLCL-133	ns



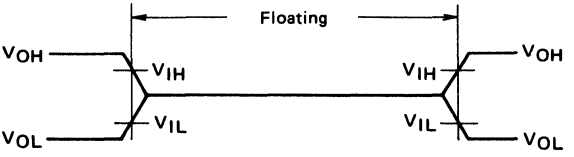
AC Characteristics Measuring Conditions

1. Input/output signal



* The input signals in AC test mode are either V_{OH} (logic "1") or V_{OL} (logic "0") input signals where logic "1" corresponds to a CPU output signal waveform measuring point in excess of V_{IH}, and logic "0" to a point below V_{IL}.

2. Floating

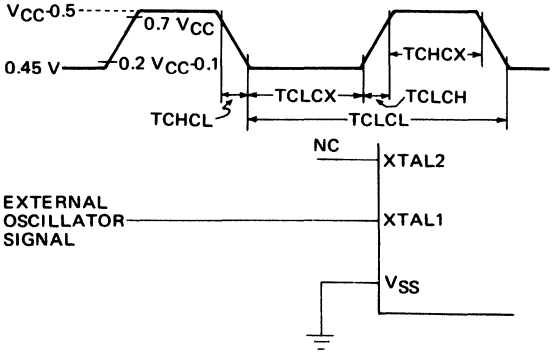


* The port 0 floating interval is measured from the time the port 0 pin voltage drops below V_{IH} after sinking to GND at 2.4 mA when switching to floating status from a "1" output, and from the time the port 0 pin voltage exceeds V_{IL} after connecting to a 400 μA source when switching to floating status from a "0" output.

XTAL1 External Clock Input Waveform Conditions

Parameter	Symbol	Min.	Max.	Units
Oscillator Freq.	1/tCLCL	DC	16	MHz
High Time	tCHCX	20		ns
Low Time	tCLCX	20		ns
Rise Time	tCLCH		20	ns
Fall Time	tCHCL		20	ns

EXTERNAL CLOCK DRIVE WAVEFORM

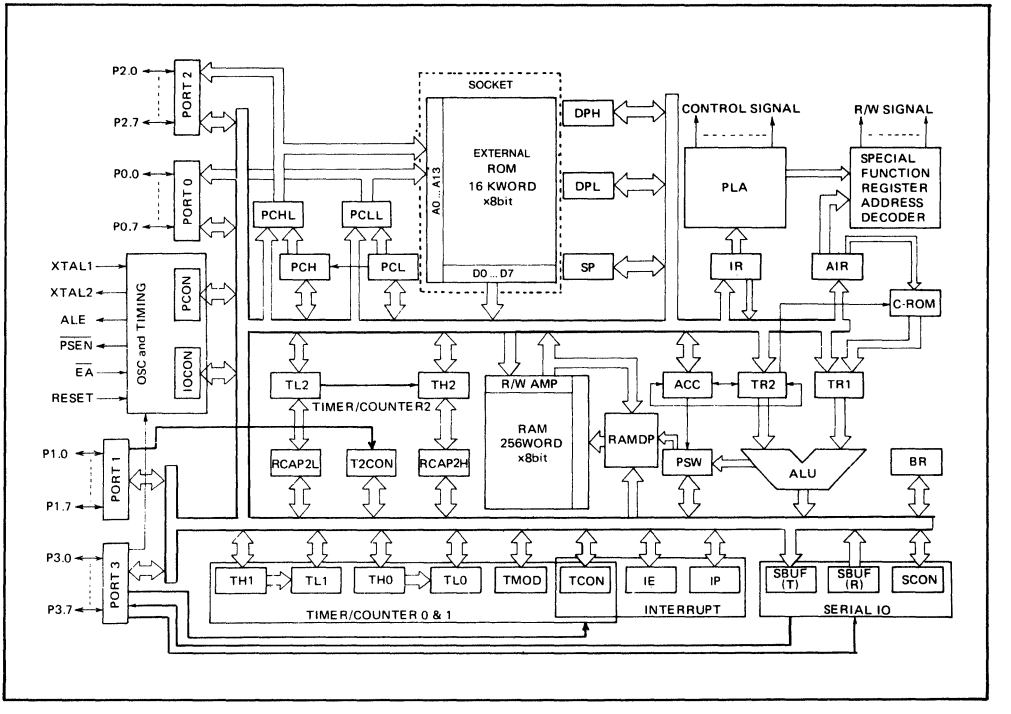


MSM85C154VS

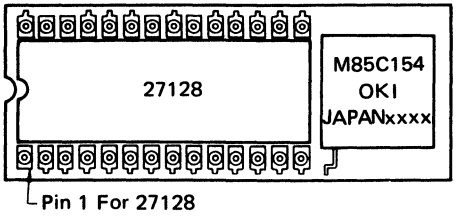
M83C154/M80C51F PIGGY BACK

GENERAL DESCRIPTION

The MSM85C154 is a device whose built-in ROM is replaced by external EPROM using the piggy-back method. External EPROM capacity is up to 16K bytes. It can be used for evaluation of programs for MSM83C154 and MSM80C51F.



INSTALLATION METHOD FOR EXTERNAL ROM



* The MSM85C154VS pin layout of bottom side is the same as the pin layout for MSM83C154-XXRS.

*NOTE

MSM85C154VS piggy back is originally designed for the programming of MSM83C154 and it covers the function as the piggy back for MSM80C51F.
Please be careful not to use additional function which dedicated to MSM83C154 in usig the piggy back for MSM80C51. The function, flag, and register listed below are dedicated to MSM83C154.

- IOCON (0F8H) : I/O CONTROL RESISTOR
- TH2 (0CDH) : TIMER 2. UPPER SIDE REGISTER
- TL2 (0CCH) : TIMER 2. LOWER SIDE REGISTER
- RCAP2H (0CBH) : CAPTURE REGISTER UPPER SIDE
- RCAP2L (0CAH) : CAPTURE REGISTER LOWER SIDE
- T2CON (0C8H) : TIMER CONTROL REGISTER 2
- IP (0B8H) : INTERRUPT PRIORITY REGISTER 2
  bit 5 (Bit address BDH) PT2
  bit 7 (Bit address BFH) PCT
- IE (0A8H) : INTERRUPT ENABLE REGISTER
  bit 5 (Bit address ADH) ET2
- PCON (087H) : POWER CONTROL REGISTER
  bit 5 (Bit address Nil) RPD
  bit 6 (Bit address Nil) HPD

In using this piggy back for MSM80C51F, do not set the above items (Control bit should not be "1").
All bits are set to "0" at initial reset.

In high temperature atmosphere, malfunction may happen (output latches of the Port 0 are set when interrupt occurs) in writing the instruction code of which LSB is "0" at address 0 of the EPROM.
To avoid this problem, please be sure to write AJMP instruction (operational code is X1) at address 0 instead of LJMP instruction (operational code is 02).

Operating frequency is from DC to 12MHz.

The MSM85C154VS has been developed assuming that it is used for evaluation of program. Please use the MSM83C154 (mask ROM version) as the devices installed on a product.

Operating temperature : −20 to 70°C
Operating voltage : 4.75 to 5.25V
Operating frequency : DC to 12 MHz

OLMS-66K SERIES (OKI ORIGINAL)

OKI semiconductor
MSM66201/66P201

OKI ORIGINAL HIGH PERFORMANCE CMOS SINGLE CHIP 8/16 BIT MICROCONTROLLER

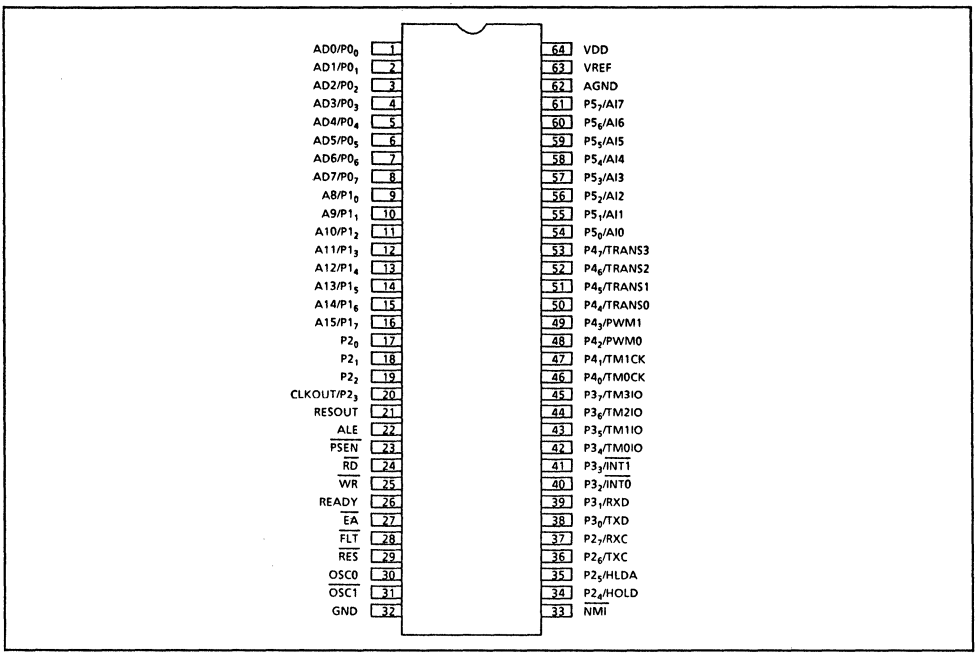
GENERAL DESCRIPTION

The OKI MSM66201 is a new generation, high performance single chip microcontroller implemented in silicon gate complementary metal oxide semiconductor technology (CMOS). Integrated within this chip are a 16-bit ALU, 16K bytes of mask program ROM, 512 bytes of data RAM, 48 I/O lines, built-in 16-bit timers, 10-bit A/D converter, serial I/O port, pulse width modulator (PWM), and an oscillator. Also available is the MSM66P201, which replaces the on-chip program memory with one-time PROM (OTP).

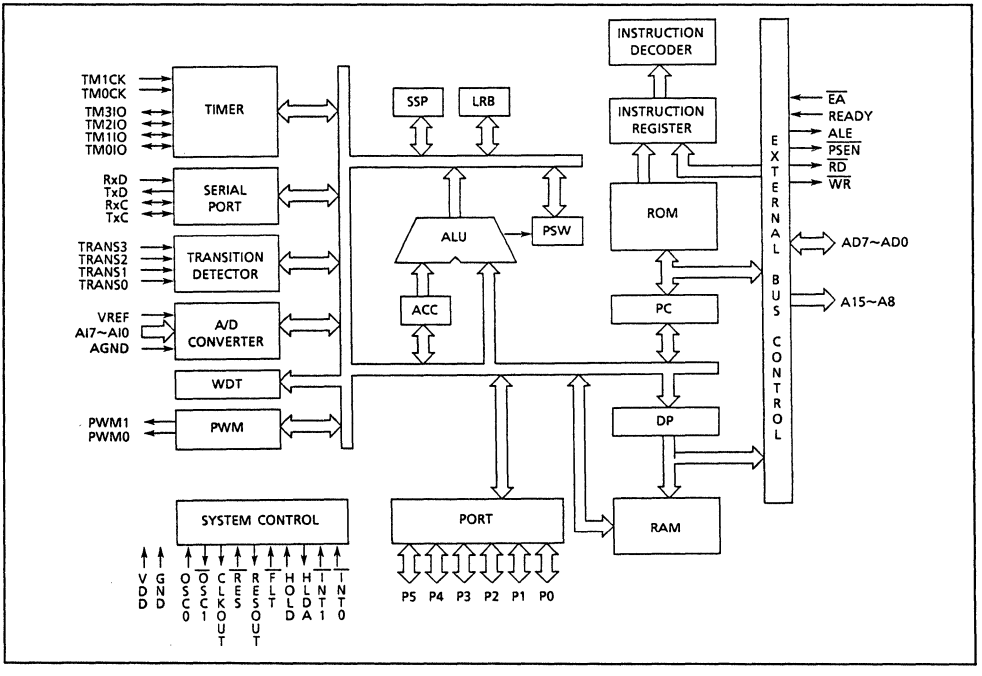
FEATURES

- 8-Bit External Data Bus Interface
- 16-Bit Internal Architecture
- 64K address space for program memory (including 16K bytes on-chip ROM)
- 64K address space for data memory (including 512 bytes on-chip RAM)
- High speed execution
  Minimum Cycle for Instruction:
  400ns (10MHz)
- The Abundance of Powerful Instructions
  8/16 data transfer operation
  8/16 bit arithmetic operation
  16(8) bit x 16(8) bit→32(16) bit
  32(16) bit x 16(8) bit→32(16) bit
  16(8) bit ± 16(8) bit→16(8) bit
  8/16 logic operation
  Bit operation
  ROM table access operation
- The same instruction allows both byte and word width operation according to Data Descriptor.
  That is to say, the same algorithm and the same source program lines are applicable to byte and word width data manipulation with only changing Data Descriptor.
- Many Addressing Modes
- 8 Input lines 40 Input/Output lines
- Built-in 16 bit timer x 4
  Each timer has the following 4 modes.
  Auto reload timer mode
  Clock output mode
  Capture register mode
  Real time output mode
- Serial Port x 1 ch.
  (variable bit length, baud rate generators for transmitter & receiver)
  Asynchronous normal mode
  Asynchronous multi processor communication mode
  Synchronous normal mode
  Synchronous multi processor communication mode
- 16 bit Pulse Width Modulator x 2
- Transition Detector x 4
- 10 bit A/D converter (8 channel)
- 1 non-maskable interrupt, 16 maskable interrupts
- Stand-by Function
  Software Clock stop
  Software CPU stop
  Hardware CPU stop
- Package:
  64 pin plastic shrink DIP (SDIP64-P-750)
  64 pin plastic QFP (QFP64-P-1414-1K)
  68 pin PLCC (QFJ68-P-S950)

PIN CONFIGURATION



FUNCTIONAL BLOCK DIAGRAM



PIN DESCRIPTION

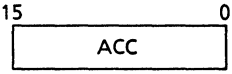
Designation	Input/Output	Function
$P0_0 - P0_7$/		P0: 8-bit I/O port. Each bit can be assigned to input or output.
AD0 – AD7	I/O	AD: Outputs the lower 8 bits of program counter during external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also outputs the address, Outputs or inputs data during an external data memory access instruction, under the control of ALE, $\overline{RD}$, and $\overline{WR}$.
$P1_0 - P1_7$/	I/O	P1: 8-bit I/O port. Each bit can be assigned to input or output.
A8 – A15		A: Outputs the upper 8 bits of program counter (PC_{8-15}) during external program memory fetch. Also outputs the upper 8 bits of address during an external data memory access instructions.
$P2_0 - P2_2$		P2: 8-bit I/O port. Each bit can be assigned to input or output.
$P2_3$/CLKOUT		CLKOUT: Clock output pin. Output frequency range is equal to or twice the system clock.
$P2_4$/HOLD		HOLD: Input pin to request the CPU to enter the hardware power-down state.
$P2_5$/HLDA	I/O	HLDA: HOLD ACKNOWLEDGE: the HLDA signal appears in response to the HOLD signal and indicates that the CPU has entered the power-down state.
$P2_6$/TxC		TxC: Transmitter clock input/output pin.
$P2_7$/RxC		RxC: Receiver clock input/output pin.
$P3_0$/TxD		P3: 8-bit I/O port. Each bit can be assigned to input or output.
$P3_1$/RxD		TxD: Transmitter data output pin.
$P3_2/\overline{INT0}$		RxD: Receiver data input pin.
$P3_3/\overline{INT1}$	I/O	$\overline{INT}$: Interrupt Request Input pin. Falling edge trigger or level trigger is selectable.
$P3_4$/TM0IO		TM0IO~TM3IO: One of the following signals is output or input.
$P3_5$/TM1IO		• clock twice the frequency range of the 16 bit timer overflow
$P3_6$/TM2IO		• load trigger signal to the capture register input
		• setting value output
$P3_7$/TM3IO		Whether the signal is input or output depends on the mode.

PIN DESCRIPTION (Continued)

Designation	Input/Output	Function
$P4_0$/TM0CK	I/O	P4: 8-bit I/O port. Each bit can be assigned to input or output
$P4_1$/TM1CK		TM0CK, TM1CK: Clock input pins of timer 0, timer 1.
P_{42}/PWM0		TRANS: Transition Detector. The input pins which sense the falling edge and set the flag.
$P4_3$/PWM1		PWM: Pulse Wide Modulator output pin.
$P4_4 - P4_7$/ TRANS0 – TRANS3		
$P5_0 - P5_7$/	INPUT	P5: 8-bit input port.
AI0 – AI7		AI: Analog signal input pin for A/D converter.
RESOUT	OUTPUT	Output 'H' level when the CPU is in RESET cycle. Reset to 'L' level by program.
ALE	OUTPUT	Address Latch Enable: The timing pulse to latch the lower 8 bit of the address output from port 0 when the CPU accesses the external memory.
$\overline{PSEN}$	OUTPUT	Program Store Enable: The strobe pulse to fetch to external program memory.
$\overline{RD}$	OUTPUT	Output strobe activated during a bus read cycle. Used to enable data on to the bus from the external data memory.
$\overline{WR}$	OUTPUT	Output strobe during a bus write cycle. Used as write strobe to external data memory.
READY	INPUT	Used when the CPU accesses low speed peripherals.
$\overline{EA}$	INPUT	Normaly set to 'H' level. If set to 'L' level, the CPU fetches the code from external program memory.
$\overline{FLT}$	INPUT	If $\overline{FLT}$ is 'H' level, ALE, $\overline{WR}$, $\overline{RD}$, $\overline{PSEN}$ are set 'H' level when reset. If $\overline{FLT}$ is set to 'L', ALE, $\overline{WR}$, $\overline{RD}$, $\overline{PSEN}$ are set to floating level when reset.
$\overline{RES}$	INPUT	RESET input pin.
OSC_0, $\overline{OSC_1}$		Oscillation circuit input and output.
$\overline{NMI}$	INPUT	Non maskable interrupt input pin (falling edge)
VREF	INPUT	Reference voltage input pin for A/D converter
AGND	INPUT	Ground for A/D converter
VDD		System power supply
GND		Ground

REGISTERS

• ACCUMULATOR



• CONTROL REGISTERS (CR)

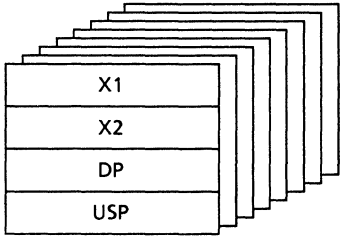
PROGRAM STATUS WORD — PSW

PROGRAM COUNTER — PC

LOCAL REGISTER BASE — LRB

SYSTEM STACK POINTER — SSP

• POINTING REGISTERS (PR)



INDEX REGISTER1

INDEX REGISTER2

DATA POINTER

USER STACK POINTER

• LOCAL REGISTERS

er0	r1	r0
er1	r3	r2
er2	r5	r4
er3	r7	r6

• SPECIAL FUNCTION REGISTERS (SFR)

All of the I/O functions are controlled by SFRs. Also, some of the internal functions (Timer, WDT, etc, ...) are controlled by SFRs. SFRs are located in the top of RAM space (000H~007FH).

MSM66201 SPECIAL FUNCTION REGISTERS

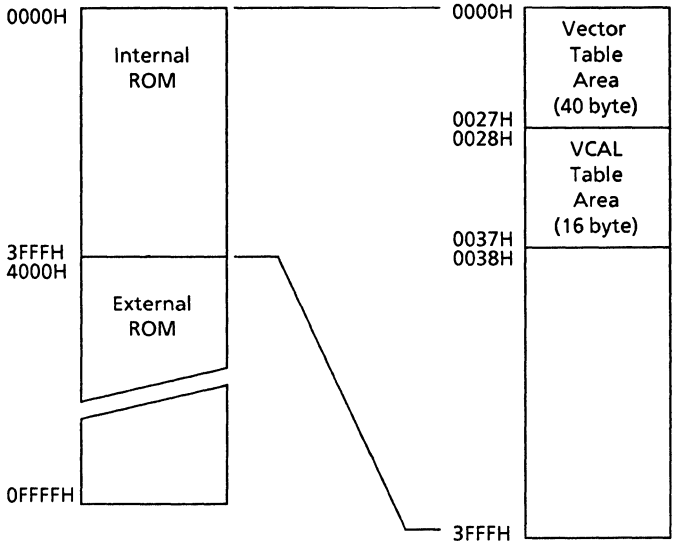
Address (HEX)	SFR Name	Abbre-viated Name	R/W	8/16-bit operation	Reset Value (HEX)
0000	System Stack Printer	SSP (ASSP)	R/W	8/16	FFH
0001					FFH
0002	Local Register Base	LRB (ALRB)			undefined
0003					
0004	Program Status Word	PSWL (APSW)			C8H
0005		PSWH			0CH
0006	Accumulator	ACC			00H
0007					00H
0010	Standby control register	SBYCON		8	F8H or F0H
0011	Watchdog timer	WDT	W		00H/WDT is stopped
0012	Peripheral control register	PRPHF	R/W		FDH
0013	Stop code acceptor	STPACP	W	8	"0"
0018	Interrupt request flag	IRQ	R/W	8/16	00H
0019					00H
001A	Interrupt Enable flag	IE			00H
001B					00H
001C	External interrupt control register	EXICON		8	FCH
0020	Port 0 data register	P0			undefined
0021	Port 0 mode register	P0IO			00H
0022	Port 1 data register	P1			undefined
0023	Port 1 mode register	P1IO			00H
0024	Port 2 data register	P2			undefined
0025	Port 2 mode register	P2IO			00H
0026	Port 2 secondary function control register	P2SF			07H
0028	Port 3 data register	P3			undefined
0029	Port 3 mode register	P3IO			00H
002A	Port 3 secondary function control register	P3SF			00H
002C	Port 4 data register	P4			undefined
002D	Port 4 mode register	P4IO			00H
002E	Port 4 secondary function control register	P4SF			00H

Address (HEX)	SFR Name	Abbre-viated Name	R/W	8/16-bit operation	Reset Value (HEX)
002F	Port 5	P5	R	8	–
0030	Timer 0 counter	TM0	R/W	16	00H
0031					00H
0032	Timer 0 register	TMR0			00H
0033					00H
0034	Timer 1 counter	TM1			00H
0035					00H
0036	Timer 1 register	TMR1			00H
0037					00H
0038	Timer 2 counter	TM2			00H
0039					00H
003A	Timer 2 register	TMR2			00H
003B					00H
003C	Timer 3 counter	TM3			00H
003D					00H
003E	Timer 3 register	TMR3			00H
003F					00H
0040	Timer 0 Control register	TC0N0	R/W	8	00H
0041	Timer 1 Control register	TC0N1			00H
0042	Timer 2 Control register	TC0N2			00H
0043	Timer 3 Control register	TC0N3			00H
0046	Transition detector register	TRNSIT			undefined
0048	Serial port transmission baud rate generator counter	STTM			00H
0049	Serial port transmission baud rate generator register	STTMR			00H
004A	Serial port transmission control register	STTMC			0CH
004C	Serial port receiving baud rate generator counter	SRTM			00H
004D	Serial port receiving baud rate generator register	SRTMR			00H
004E	Serial port receiving control register	SRTMC			0EH
0050	Serial port transmission mode control register	STCON			80H
0051	Serial port transmission data buffer register	STBUF	W		undefined
0054	Serial port receiving mode control register	SRCON	R/W		00H
0055	Serial port receiving data buffer register	SRBUF	R		undefined
0056	Serial port receiving error register	SRSTAT	R/W		F0H
0058	A/D scan mode register	ADSCAN			80H
0059	A/D select mode register	ADSEL			A0H

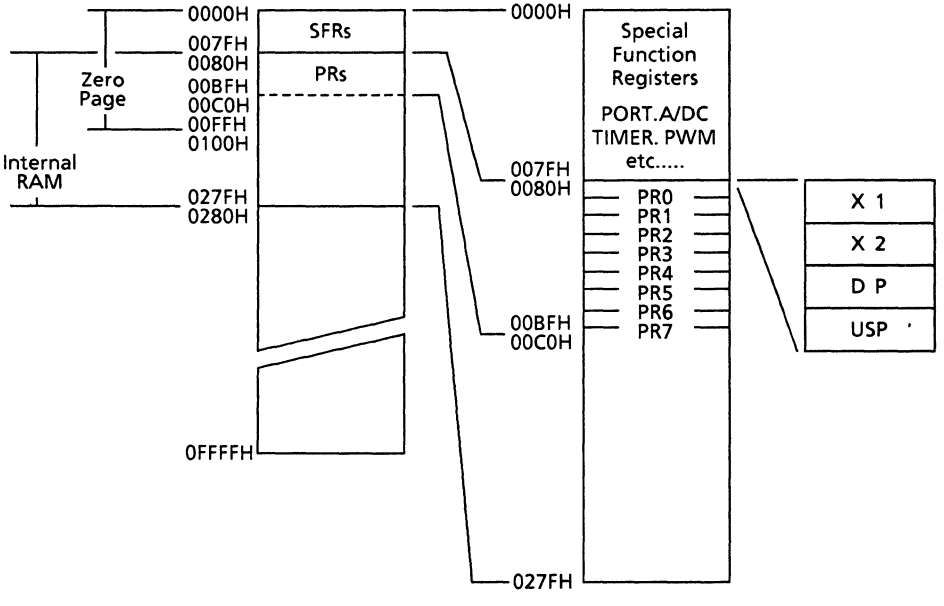
Address (HEX)	SFR Name	Abbre-viated Name	R/W	8/16-bit operation	Reset Value (HEX)
0060	A/D conversion result register 0	ADCR0	R	8/16	undefined
0061					
0062	A/D conversion result register 1	ADCR1			
0063					
0064	A/D conversion result register 2	ADCR2			
0065					
0066	A/D conversion result register 3	ADCR3			
0067					
0068	A/D conversion result register 4	ADCR4			
0069					
006A	A/D conversion result register 5	ADCR5			
006B					
006C	A/D conversion result register 6	ADCR6			
006D					
006E	A/D conversion result register 7	ADCR7			
006F					
0070	PWM 0 counter	PWMC0	R/W		00H
0071					00H
0072	PWM 0 register	PWMR0			00H
0073					00H
0074	PWM 1 counter	PWMC1			00H
0075					00H
0076	PWM 1 register	PWMR1			00H
0077					00H
0078	PWM 0 control register	PWC0N0		8	00H
007A	PWM 1 control register	PWC0N1			00H

MEMORY MAP

- Program Memory Space



- Data Memory Space

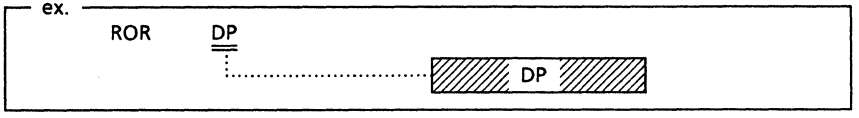


ADDRESSING MODE

The MSM66201 supports 64KB of data space and 64KB of program space with verious types of addressing modes. These modes divide into the following types.

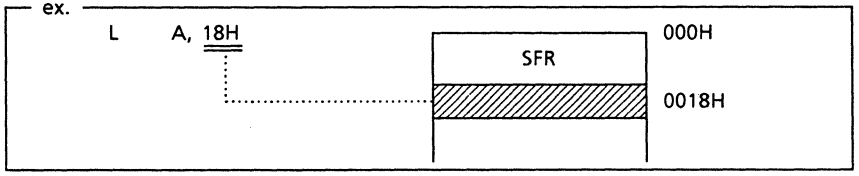
1. RAM ADDRESSING (FOR DATA SPACE)

1.1 Register Direct Addressing

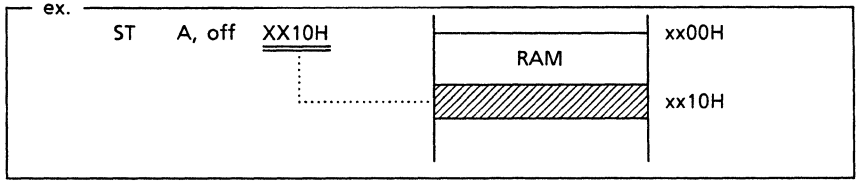


1.2 Displacement Addressing

a) Zero Page

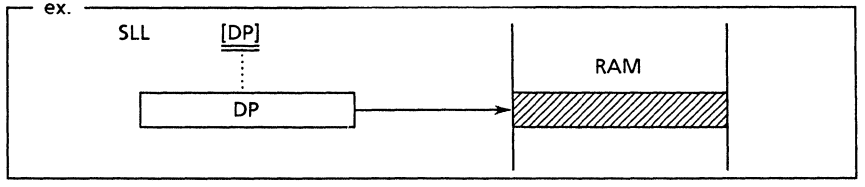


b) Direct Page

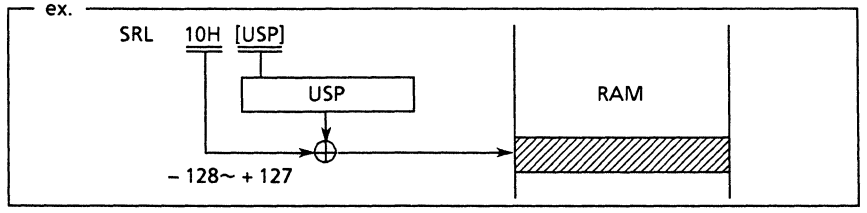


1.3 Pointing Register Indirect Addressing

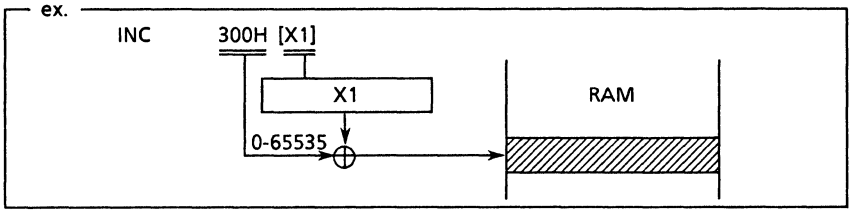
a) Data Pointer (DP) Indirect



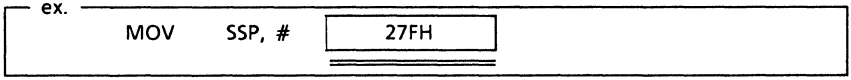
b) User Stack Pointer (USP) Indirect



c) Index register (X1, X2) Indirect

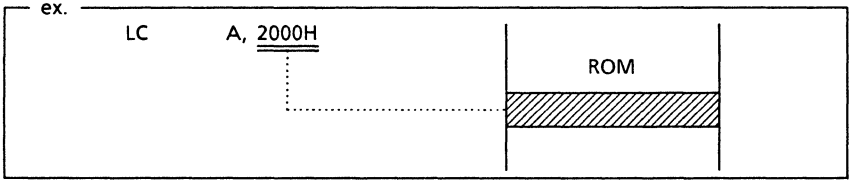


1.4 Immediate Addressing



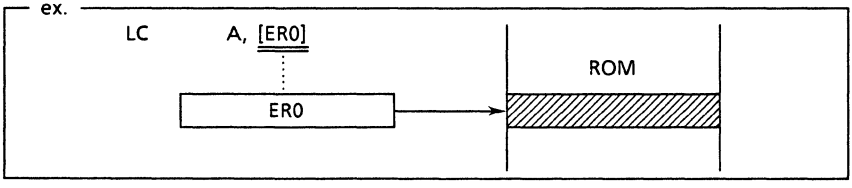
2. ROM ADDRESSING (FOR PROGRAM SPACE)

2.1 Direct Addressing



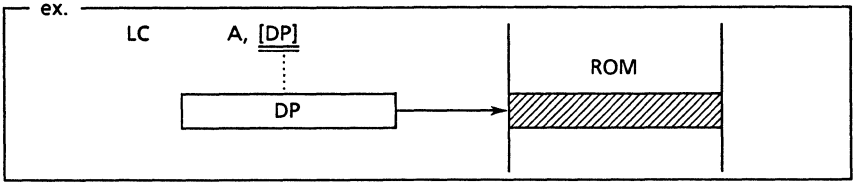
2.2 Simple Indirect Addressing

a) Local Register Indirect

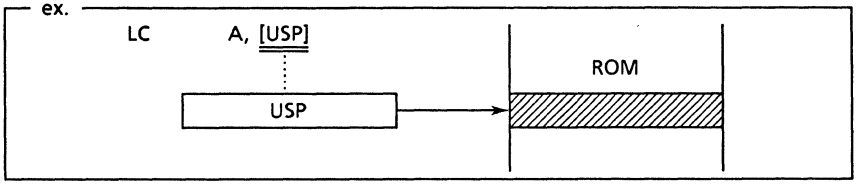


b) Pointing Register Indirect

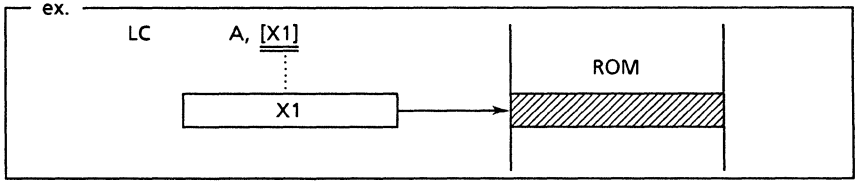
① Data Pointer (DP) Indirect



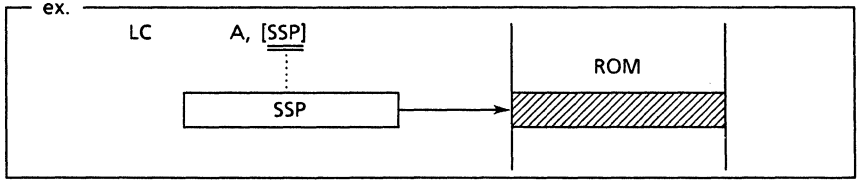
② User Stack Pointer (USP) Indirect



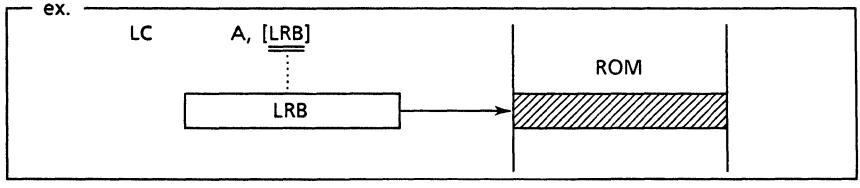
③ Index Register (X1, X2) Indirect



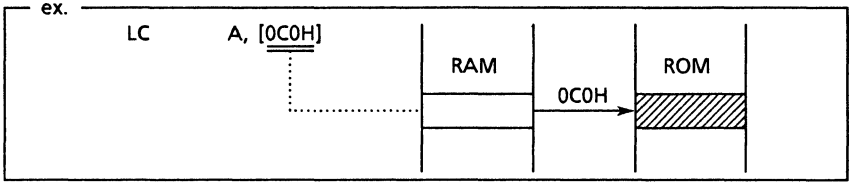
c) System Stack Pointer (SSP) Indirect



d) Local Register Base (LRB) Indirect

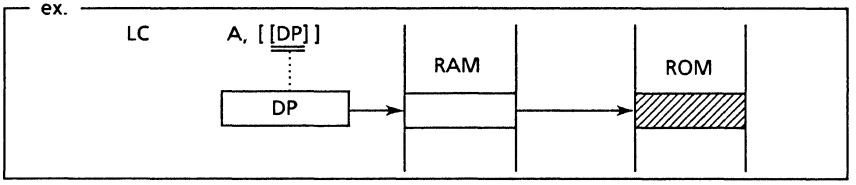


e) RAM Indirect

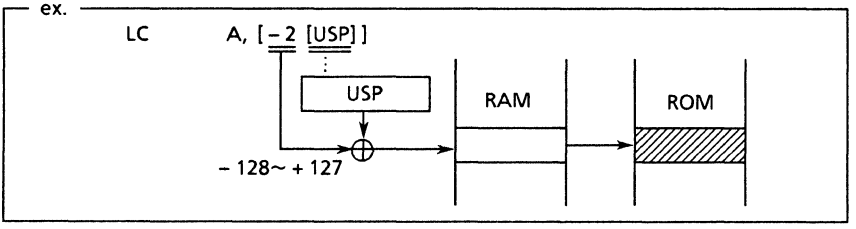


2.3 Double Indirect Addressing

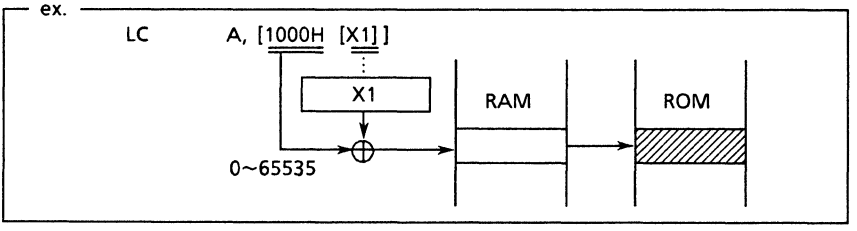
a) Data Pointer (DP) Double Indirect



b) User Stack Pointer (USP) Double Indirect



c) Index Register (X1, X2) Double Indirect



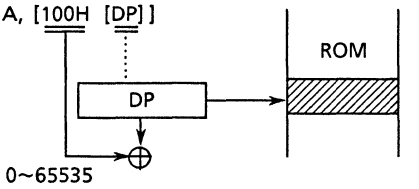
2.4 Indirect Addressing with 16 bit offset

a) Pointing Register Indirect

① Data Pointer (DP) Indirect

ex.

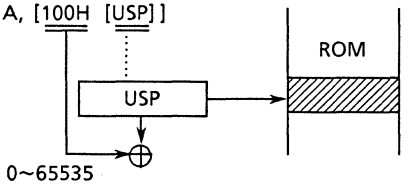
```
LC      A, [100H  [DP] ]
```



② User Stack Pointer (USP) Indirect

ex.

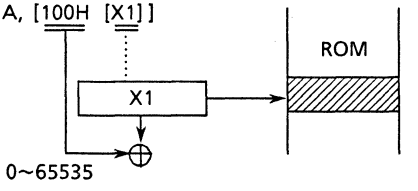
```
LC      A, [100H  [USP] ]
```



③ Index Register (X1, X2) Indirect

ex.

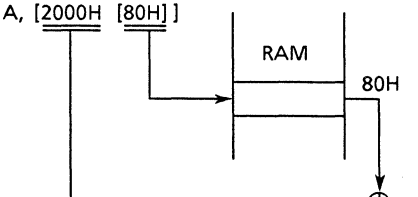
```
LC      A, [100H  [X1] ]
```



b) RAM Indirect

ex.

```
LC      A, [2000H  [80H] ]
```



INSTRUCTIONS

(Note) '*' Addressing mode expression
'#' and 'imme' Immediate value

Data Transfer

Instruction	Function
L A, *	A ← *
ST A, *	A → *
MOV *, A	A ← *
MOV *, #	* → Imme.
MOV er.*	er ← *
MOV d, *	(d) ← *
MOV DP, *	DP ← *
MOV X1, *	X1 ← *
MOV X2, *	X2 ← *
MOV USP, *	USP ← *
MOV PSW, *	PSW ← *
MOV SSP, *	SSP ← *
MOV LRB, *	LRB ← *
CLR *	* ← 0
SWAP	$A_{15-8} \leftrightarrow A_{7-0}$
XCHG A, *	A ↔ *
XNBL A, *	$A_{3-8} \leftrightarrow *_{3-0}$

Instruction	Function
LB A, *	A_L ← *
STB A, *	A_L → *
MOVB A, *	Al ← *
MOVB *, A	* ← AL
MOVB *, #	* ← Imme.
MOVB r, *	r ← *
MOVB d, *	(d) ← *
MOVB PSW0, *	PSW0 ← *
MOVB SCB, *	SCB ← *
CLRB *	* ← 0
SWAPB	$A_{15-8} \leftrightarrow A_{7-0}$
XCHGB A, *	A ↔ *

Push & Pop

Instruction	Function
PUSHS *, A	* → SYSTEM STACK
POPS *, #	* ← SYSTEM STACK

Rotate & Shift

Instruction	Function
ROL *	Rotate left
ROR *	Rotate right
SLL *	Shift left Logical
SRL *	Shift right Logical
SRA *	Shift right Arithmetic

Instruction	Function
ROLB *	Rotate left
RORB *	Rotate right
SLLB *	Shift left Logical
SRLB *	Shift right Logical
SRAB *	Shift right Arithmetic

Increment & Decrement

Instruction	Function
INC *	* ← * + 1
DEC *	* ← * – 1

Instruction	Function
INCB *	* ← * + 1
DECB *	* ← * – 1

ROM Table Reference

Instruction	Function
LC A, *	A ← * (ROM)
CMPC A, *	* – * (ROM)

Instruction	Function
LCB A, *	A_L ← * (ROM)
CMPCB A, *	A_L – * (ROM)

Arithmetic Operation

Instruction		Function
MUL		er1: A ← A x er0
DIV		er0: A ← er0: A/er2
ADD	A, *	A ← A + *
ADD	*, A	* ← * + A
ADD	*, d	* ← * + (d)
ADD	*, #	* ← * + Imme.
ADC	A, *	A ← A + * + C
ADC	*, A	* ← * + A + C
ADC	*, d	* ← * + (d) + C
ADC	*, #	* ← * + Imme. + C
SUB	A, *	A ← A − *
SUB	*, A	* ← * − A
SUB	*, d	* ← * − (d)
SUB	*, #	* ← * − Imme.
SBC	A, *	A ← A − * − C
SBC	*, A	* ← * − A − C
SBC	*, d	* ← * − (d) − C
SBC	*, #	* ← * − Imme. − C

Instruction		Function
MULB		A ← A_L x r0
DIVB		A ← A/r0
ADDB	A, *	A_L ← A_L + *
ADDB	*, A	* ← * + A_L
ADDB	*, d	* ← * + (d)
ADDB	*, #	* ← * + Imme.
ADCB	A, *	A_L ← A_L + * + C
ADCB	*, A	* ← * + A_L + C
ADCB	*, d	* ← * + (d) + C
ADCB	*, #	* ← * + Imme. + C
SUBB	A, *	A_L ← A_L − *
SUBB	*, A	* ← * − A_L
SUBB	*, d	* ← * − (d)
SUBB	*, #	* ← * − Imme.
SBCB	A, *	A_L ← A_L − * − C
SBCB	*, A	* ← * − A_L − C
SBCB	*, d	* ← * − (d) − C
SBCB	*, #	* ← * − Imme. − C

Logical Operation

Instruction		Function
AND	A, *	A ← A and *
AND	*, A	* ← * and A
AND	*, d	* ← * and (d)
AND	*, #	* ← * and Imme.
OR	A, *	A ← A or *
OR	*, A	* ← * or A
OR	*, d	* ← * or (d)
OR	*, #	* ← * or Imme.
XOR	A, *	A ← A xor *
XOR	*, A	* ← xor A
XOR	*, d	* ← xor (d)
XOR	*, #	* ← * xor Imme.

Instruction		Function
ANDB	A, *	A_L ← A_L and *
ANDB	*, A	* ← * and A_L
ANDB	*, d	* ← * and (d)
ANDB	*, #	* ← * and Imme.
ORB	A, *	A_L ← A_L or *
ORB	*, A	* ← * or A_L
ORB	*, d	* ← * or (d)
ORB	*, #	* ← * or Imme.
XORB	A, *	A_L ← A_L xor *
XORB	*, A	* ← * xor A_L
XORB	*, d	* ← * xor (d)
XORB	*, #	* ← * xor Imme.

Comparison

Instruction		Function
CMP	A, *	A − *
CMP	*, A	* − A
CMP	*, d	* − (d)
CMP	*, #	* − Imme.

Instruction		Function
CMPB	A, *	A_L − *
CMPB	*, A	* − A_L
CMPB	*, d	* − (d)
CMPB	*, #	* − Imme.

Decimal Adjust

Instruction	Function
DAA	Decimal Adjust for Addition
DAS	Decimal Adjust for Substruct

Data Type Conversion

Instruction	Function
EXTND	A15 – 8 ← An

Bit Operation

Instruction		Function
SBR	*	bit ← 1
RBR	*	bit ← 0
MBR	C, *	C ← bit
MBR	*, C	bit ← C
TRB	*, C	Z ← bit

Instruction		Function
SBR	*	bit ← 1
RBR	*	bit ← 0
MBR	C, *	C ← bit
MBR	*, C	bit ← C

Jump & Call

Instruction		Function
SJ	adrs	Short Jump
J	*	Jump
JC	EQ, adrs	Jump if '='
JC	LE, adrs	Jump if '<='
JC	GE, adrs	Jump if '>='
JBS	bit, adrs	Jump if bit-on
JRNZ	DP, adrs	Loop Function

Instruction		Function
SCAL	adrs	Short Call
CAL	*	Call Subroutine
JC	NE, adrs	Jump if '<>'
JC	LT, adrs	Jump if '<'
JC	GT, adrs	Jump if '>'
JBR	bit, adrs	Jump if bit-off
VCAL	table:adrs	Vector Call

Return

Instruction	Function
RT	Return from Subroutine
RTI	Return from Interrupt

Others

Instruction	Function
SC	C ← 1
SS	STACK FLAG ← 1
NOP	No Operation

Instruction	Function
RC	C ← 0
BRK	Software Reset

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	Ta = 25°C	− 0.3~7.0	V
Input Voltage	V_I		− 0.3~V_{DD} + 0.3	V
Output Voltage	V_O		− 0.3~V_{DD} + 0.3	V
Analog Ref. Voltage	V_R		− 0.3~V_{DD} + 0.3	V
Analog Input Voltage	V_{AI}		− 0.3~V_R	V
Power Dissipation	P_D	Ta = 25°C (per package)	400 MAX	mW
		Ta = 25°C (per out)	50 MAX	mW
Storage Temperature	T_{STG}	−	− 55~ + 150	°C

OPERATING CONDITIONS

Parameter	Symbol	Conditions		Limits	Unit
Supply Voltage	V_{DD}	$f_{(OSC)} \leqq 10$ MHz		4.5~5.5	V
Memory Hold Voltage	V_{DDH}	$f_{(OSC)} = 0$ Hz		2~5.5	V
Operating Frequency	$f_{(OSC)}$	$V_{DD} = 5V \pm 10\%$		0~10	MHz
Ambient Temperature	Ta	MSM66201		− 40~ + 85	°C
		MSM66P201		0~ + 70	
Fan Out	P_D	MOS load		20	
		TTL load	P0	2	
			P1, P2, P3, P4	1	

OKI semiconductor
MSM66207/66P207

OKI ORIGINAL HIGH PERFORMANCE CMOS SINGLE CHIP 8/16 BIT MICROCONTROLLER

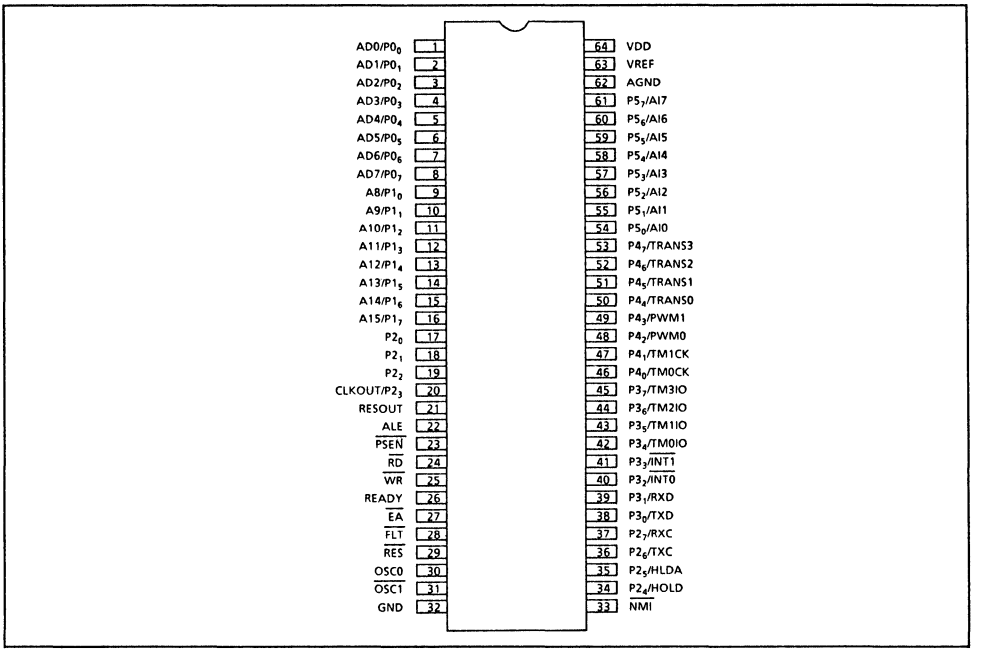
GENERAL DESCRIPTION

The OKI MSM66207 is a new generation, high performance single chip microcontroller implemented in silicon gate complementary metal oxide semiconductor technology (CMOS). Integrated within this chip are a 16-bit ALU, 32K bytes of mask program ROM, 1K bytes of data RAM, 48 I/O lines, built-in 16-bit timers, 10-bit A/D converter, serial I/O port, pulse width modulator (PWM), and an oscillator. Also available is the MSM66P207, which replaces the on-chip program memory with one-time PROM.

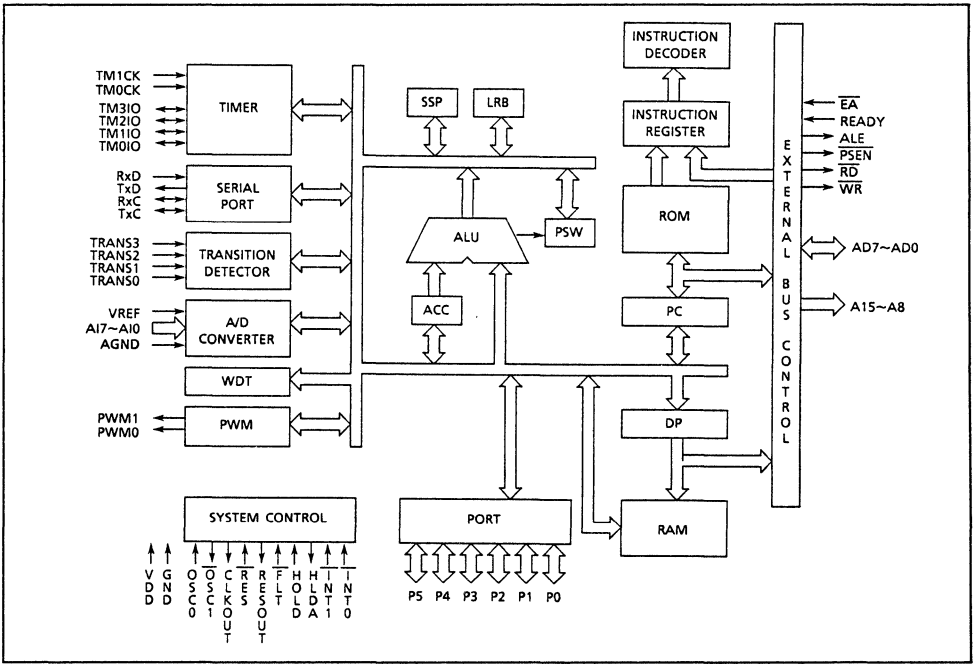
FEATURES

- 8-Bit External Data Bus Interface
- 16-Bit Internal Architecture
- 64K address space for program memory (including 32K bytes on-chip ROM)
- 64K address space for data memory (including 1K bytes on-chip RAM)
- High speed execution
  Minimum Cycle for Instruction:
  400ns (10MHz)
- The Abundance of Powerful Instructions
  8/16 data transfer operation
  8/16 bit arithmetic operation
  16(8) bit x 16(8) bit→32(16) bit
  32(16) bit x 16(8) bit→32(16) bit
  16(8) bit ± 16(8) bit→16(8) bit
  8/16 logic operation
  Bit operation
  ROM table access operation
- The same instruction allows both byte and word width operation according to Data Descriptor.
  That is to say, the same algorithm and the same source program lines are applicable to byte and word width data manipulation with only changing Data Descriptor.
- Many Addressing Modes
- 8 Input lines 40 Input/Output lines
- Built-in 16 bit timer x 4
  Each timer has the following 4 modes.
  Auto reload timer mode
  Clock output mode
  Capture register mode
  Real time output mode
- Serial Port x 1 ch.
  (variable bit length, baud rate generators for transmitter & receiver)
  Asynchronous normal mode
  Asynchronous multi processor communication mode
  Synchronous normal mode
  Synchronous multi processor communication mode
- 16 bit Pulse Width Modulator x 2
- Transition Detector x 4
- 10 bit A/D converter (8 channel)
- 1 non-maskable interrupt, 16 maskable interrupts
- Stand-by Function
  Software Clock stop
  Software CPU stop
  Hardware CPU stop
- Package:
  64 pin plastic shrink DIP (SDIP64-P-750)
  68 pin PLCC (QFJ68-P-S950)

PIN CONFIGURATION



FUNCTIONAL BLOCK DIAGRAM



PIN DESCRIPTION

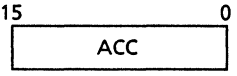
Designation	Input/Output	Function
$P0_0$ – $P0_7$/		P0: 8-bit I/O port. Each bit can be assigned to input or output.
AD0 – AD7	I/O	AD: Outputs the lower 8 bits of program counter during external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also outputs the address, Outputs or inputs data during an external data memory access instruction, under the control of ALE, $\overline{RD}$, and $\overline{WR}$.
$P1_0$ – $P1_7$/	I/O	P1: 8-bit I/O port. Each bit can be assigned to input or output.
A8 – A15		A: Outputs the upper 8 bits of program counter (PC_{8-15}) during external program memory fetch. Also outputs the upper 8 bits of address during an external data memory access instructions.
$P2_0$ – $P2_2$		P2: 8-bit I/O port. Each bit can be assigned to input or output.
$P2_3$/CLKOUT		CLKOUT: Clock output pin. Output frequency range is equal to or twice the system clock.
$P2_4$/HOLD		HOLD: Input pin to request the CPU to enter the hardware power-down state.
$P2_5$/HLDA	I/O	HLDA: HOLD ACKNOWLEDGE: the HLDA signal appears in response to the HOLD signal and indicates that the CPU has entered the power-down state.
$P2_6$/TxC		TxC: Transmitter clock input/output pin.
$P2_7$/RxC		RxC: Receiver clock input/output pin.
$P3_0$/TxD		P3: 8-bit I/O port. Each bit can be assigned to input or output.
$P3_1$/RxD		TxD: Transmitter data output pin.
$P3_2/\overline{INT0}$		RxD: Receiver data input pin.
$P3_3/\overline{INT1}$	I/O	$\overline{INT}$: Interrupt Request Input pin. Falling edge trigger or level trigger is selectable.
$P3_4$/TM0IO		TM0IO~TM3IO: One of the following signals is output or input. • clock twice the frequency range of the 16 bit timer overflow • load trigger signal to the capture register input • setting value output
$P3_5$/TM1IO		
$P3_6$/TM2IO		
$P3_7$/TM3IO		Whether the signal is input or output depends on the mode.

PIN DESCRIPTION (Continued)

Designation	Input/Output	Function
$P4_0$/TM0CK	I/O	P4: 8-bit I/O port. Each bit can be assigned to input or output
$P4_1$/TM1CK		TM0CK, TM1CK: Clock input pins of timer 0, timer 1.
$P4_2$/PWM0		TRANS: Transition Detector. The input pins which sense the falling edge and set the flag.
$P4_3$/PWM1 $P4_4$ – $P4_7$/ TRANS0 – TRANS3		PWM: Pulse Wide Modulator output pin.
$P5_0$ – $P5_7$/	INPUT	P5: 8-bit input port.
AI0 – AI7		AI: Analog signal input pin for A/D converter.
RESOUT	OUTPUT	Output 'H' level when the CPU is in RESET cycle. Reset to 'L' level by program.
ALE	OUTPUT	Address Latch Enable: The timing pulse to latch the lower 8 bit of the address output from port 0 when the CPU accesses the external memory.
$\overline{PSEN}$	OUTPUT	Program Store Enable: The strobe pulse to fetch to external program memory.
$\overline{RD}$	OUTPUT	Output strobe activated during a bus read cycle. Used to enable data on to the bus from the external data memory.
$\overline{WR}$	OUTPUT	Output strobe during a bus write cycle. Used as write strobe to external data memory.
READY	INPUT	Used when the CPU accesses low speed peripherals.
$\overline{EA}$	INPUT	Normaly set to 'H' level. If set to 'L' level, the CPU fetches the code from external program memory.
$\overline{FLT}$	INPUT	If $\overline{FLT}$ is 'H' level, ALE, $\overline{WR}$, $\overline{RD}$, $\overline{PSEN}$ are set 'H' level when reset. If $\overline{FLT}$ is set to 'L', ALE, $\overline{WR}$, $\overline{RD}$, $\overline{PSEN}$ are set to floating level when reset.
$\overline{RES}$	INPUT	RESET input pin.
OSC_0, $\overline{OSC_1}$		Oscillation circuit input and output.
$\overline{NMI}$	INPUT	Non maskable interrupt input pin (falling edge)
VREF	INPUT	Reference voltage input pin for A/D converter
AGND	INPUT	Ground for A/D converter
VDD		System power supply
GND		Ground

REGISTERS

• ACCUMULATOR



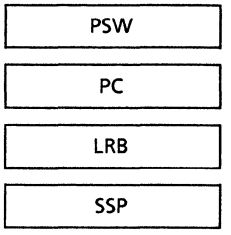
• CONTROL REGISTERS (CR)

PROGRAM STATUS WORD

PROGRAM COUNTER

LOCAL REGISTER BASE

SYSTEM STACK POINTER



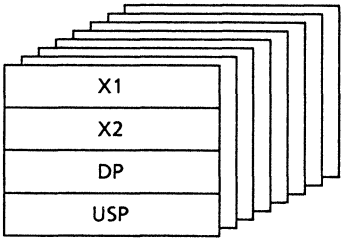
• POINTING REGISTERS (PR)

INDEX REGISTER1

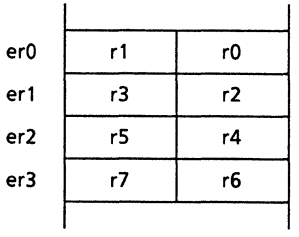
INDEX REGISTER2

DATA POINTER

USER STACK POINTER



• LOCAL REGISTERS



• SPECIAL FUNCTION REGISTERS (SFR)

All of the I/O functions are controlled by SFRs. Also, some of the internal functions (Timer, WDT, etc, ...) are controlled by SFRs. SFRs are located in the top of RAM space (000H~007FH).

MSM66207 SPECIAL FUNCTION REGISTERS

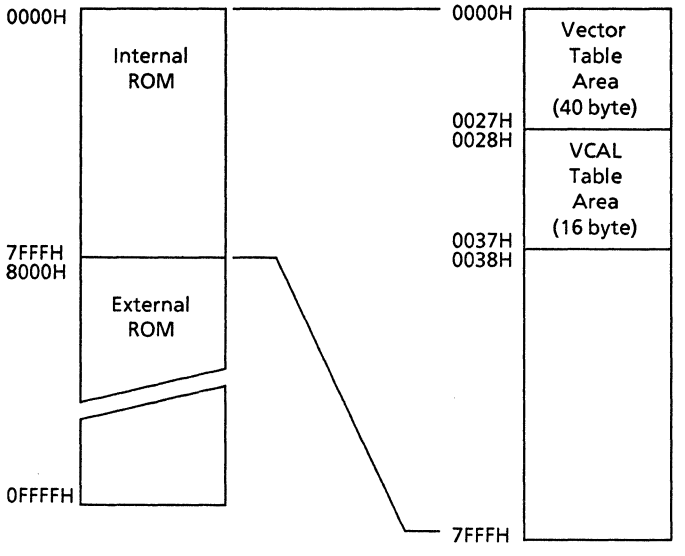
Address (HEX)	SFR Name	Abbre-viated Name	R/W	8/16-bit operation	Reset Value (HEX)
0000	System Stack Printer	SSP (ASSP)	R/W	8/16	FFH
0001					FFH
0002	Local Register Base	LRB (ALRB)			undefined
0003					
0004	Program Status Word	PSWL (APSW)			C8H
0005		PSWH			0CH
0006	Accumulator	ACC			00H
0007					00H
0010	Standby control register	SBYCON		8	F8H or F0H
0011	Watchdog timer	WDT	W		00H/WDT is stopped
0012	Peripheral control register	PRPHF	R/W		FDH
0013	Stop code acceptor	STPACP	W	8	"0"
0018	Interrupt request flag	IRQ	R/W	8/16	00H
0019					00H
001A	Interrupt Enable flag	IE			00H
001B					00H
001C	External interrupt control register	EXICON		8	FCH
0020	Port 0 data register	P0			undefined
0021	Port 0 mode register	P0IO			00H
0022	Port 1 data register	P1			undefined
0023	Port 1 mode register	P1IO			00H
0024	Port 2 data register	P2			undefined
0025	Port 2 mode register	P2IO			00H
0026	Port 2 secondary function control register	P2SF			07H
0028	Port 3 data register	P3			undefined
0029	Port 3 mode register	P3IO			00H
002A	Port 3 secondary function control register	P3SF			00H
002C	Port 4 data register	P4			undefined
002D	Port 4 mode register	P4IO			00H
002E	Port 4 secondary function control register	P4SF			00H

Address (HEX)	SFR Name	Abbre-viated Name	R/W	8/16-bit operation	Reset Value (HEX)
002F	Port 5	P5	R	8	–
0030	Timer 0 counter	TM0	R/W	16	00H
0031					00H
0032	Timer 0 register	TMR0			00H
0033					00H
0034	Timer 1 counter	TM1			00H
0035					00H
0036	Timer 1 register	TMR1			00H
0037					00H
0038	Timer 2 counter	TM2			00H
0039					00H
003A	Timer 2 register	TMR2			00H
003B					00H
003C	Timer 3 counter	TM3			00H
003D					00H
003E	Timer 3 register	TMR3			00H
003F					00H
0040	Timer 0 Control register	TC0N0	R/W	8	00H
0041	Timer 1 Control register	TC0N1			00H
0042	Timer 2 Control register	TC0N2			00H
0043	Timer 3 Control register	TC0N3			00H
0046	Transition detector register	TRNSIT			undefined
0048	Serial port transmission baud rate generator counter	STTM			00H
0049	Serial port transmission baud rate generator register	STTMR			00H
004A	Serial port transmission control register	STTMC			0CH
004C	Serial port receiving baud rate generator counter	SRTM			00H
004D	Serial port receiving baud rate generator register	SRTMR			00H
004E	Serial port receiving control register	SRTMC			0EH
0050	Serial port transmission mode control register	STCON			80H
0051	Serial port transmission data buffer register	STBUF	W		undefined
0054	Serial port receiving mode control register	SRCON	R/W		00H
0055	Serial port receiving data buffer register	SRBUF	R		undefined
0056	Serial port receiving error register	SRSTAT	R/W		F0H
0058	A/D scan mode register	ADSCAN			80H
0059	A/D select mode register	ADSEL			A0H

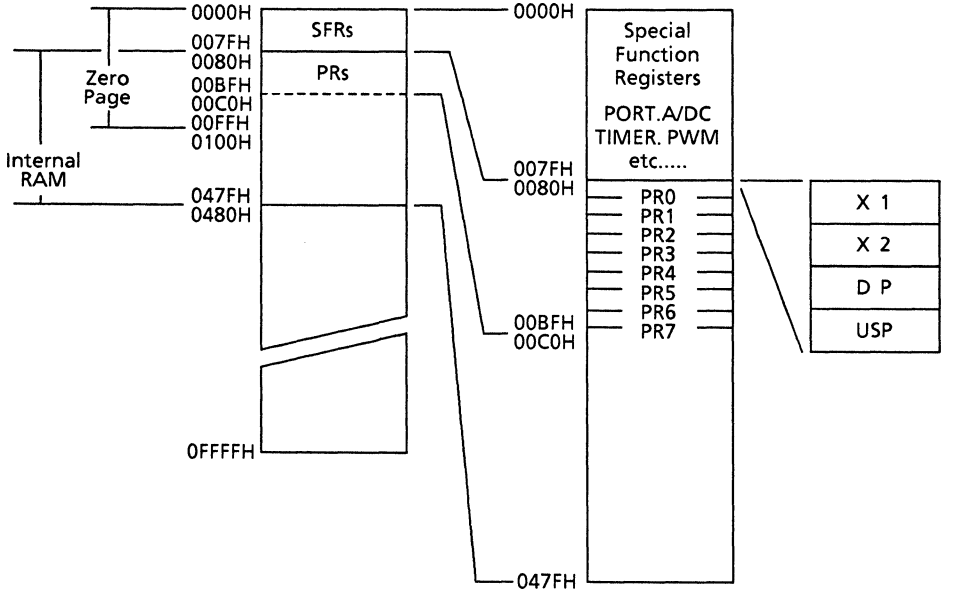
Address (HEX)	SFR Name	Abbre-viated Name	R/W	8/16-bit operation	Reset Value (HEX)
0060	A/D conversion result register 0	ADCR0	R	8/16	undefined
0061					
0062	A/D conversion result register 1	ADCR1			
0063					
0064	A/D conversion result register 2	ADCR2			
0065					
0066	A/D conversion result register 3	ADCR3			
0067					
0068	A/D conversion result register 4	ADCR4			
0069					
006A	A/D conversion result register 5	ADCR5			
006B					
006C	A/D conversion result register 6	ADCR6			
006D					
006E	A/D conversion result register 7	ADCR7			
006F					
0070	PWM 0 counter	PWMC0	R/W		00H
0071					00H
0072	PWM 0 register	PWMR0			00H
0073					00H
0074	PWM 1 counter	PWMC1			00H
0075					00H
0076	PWM 1 register	PWMR1			00H
0077					00H
0078	PWM 0 control register	PWC0N0		8	00H
007A	PWM 1 control register	PWC0N1			00H

MEMORY MAP

- Program Memory Space



- Data Memory Space

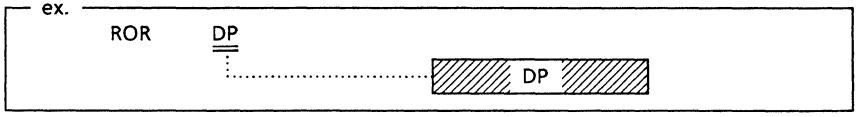


ADDRESSING MODE

The MSM66207 supports 64KB of data space and 64KB of program space with verious types of addressing modes. These modes divide into the following types.

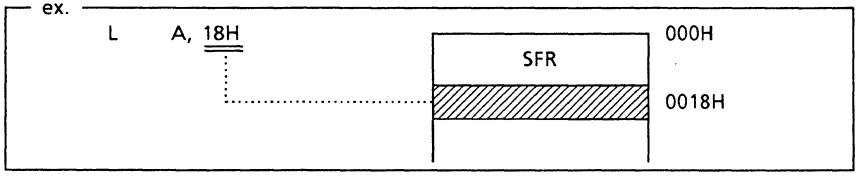
1. RAM ADDRESSING (FOR DATA SPACE)

1.1 Register Direct Addressing

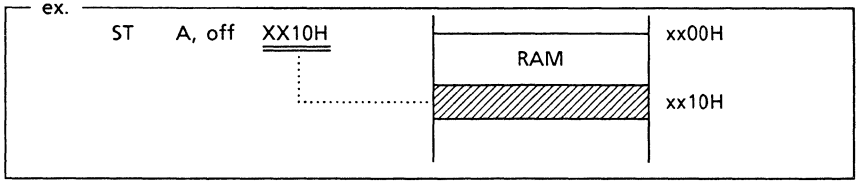


1.2 Displacement Addressing

a) Zero Page

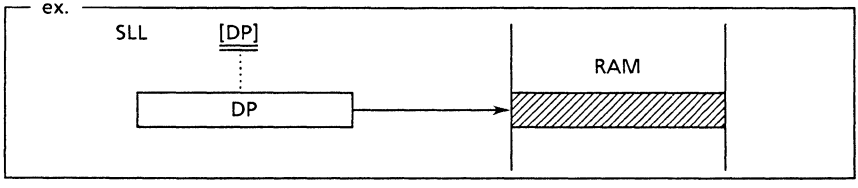


b) Direct Page

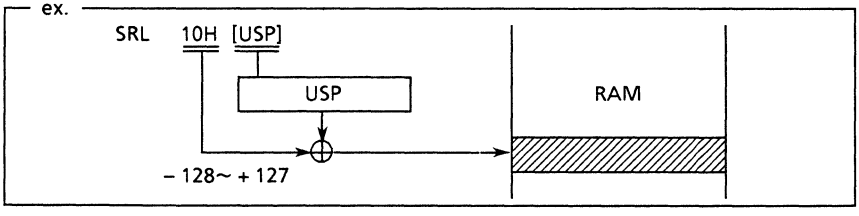


1.3 Pointing Register Indirect Addressing

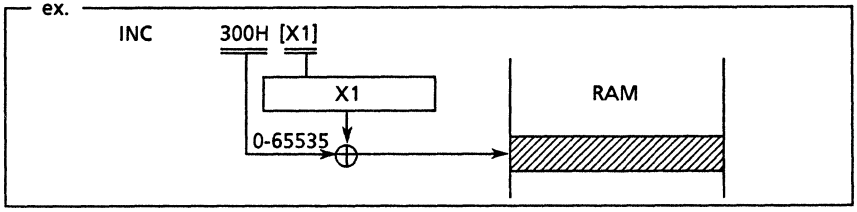
a) Data Pointer (DP) Indirect



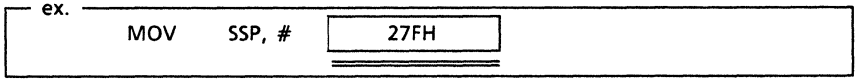
b) User Stack Pointer (USP) Indirect



c) Index register (X1, X2) Indirect

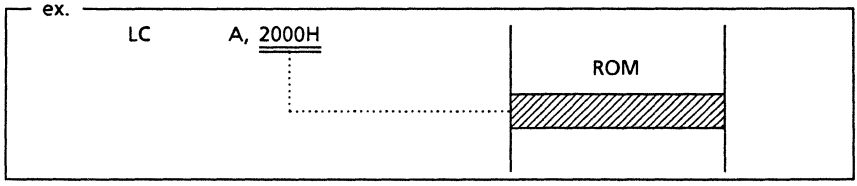


1.4 Immediate Addressing



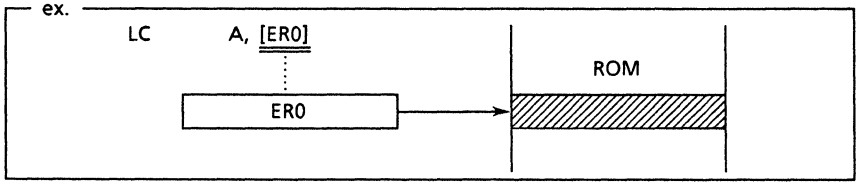
2. ROM ADDRESSING (FOR PROGRAM SPACE)

2.1 Direct Addressing



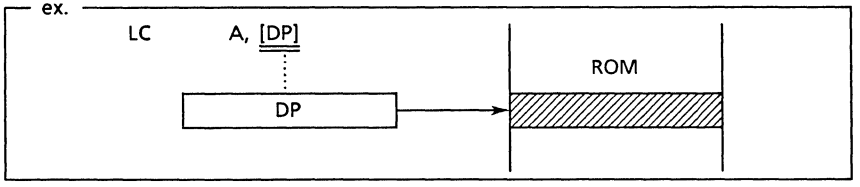
2.2 Simple Indirect Addressing

a) Local Register Indirect

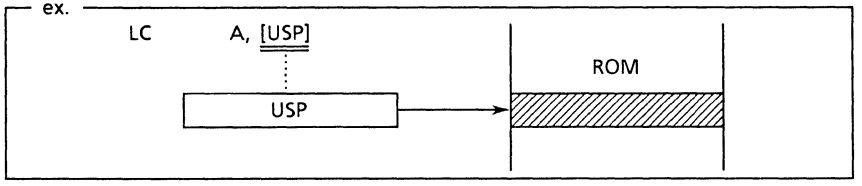


b) Pointing Register Indirect

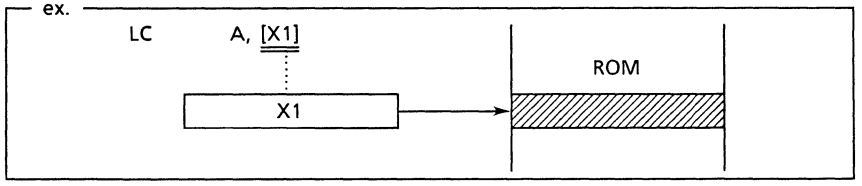
① Data Pointer (DP) Indirect



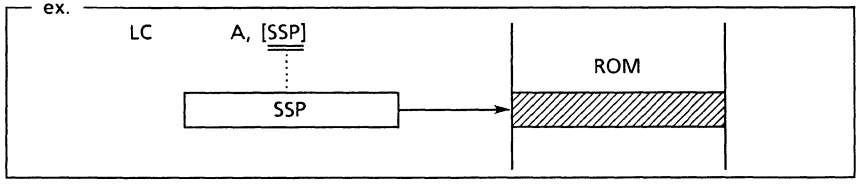
② User Stack Pointer (USP) Indirect



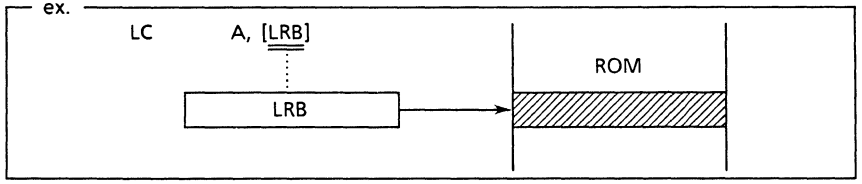
③ Index Register (X1, X2) Indirect



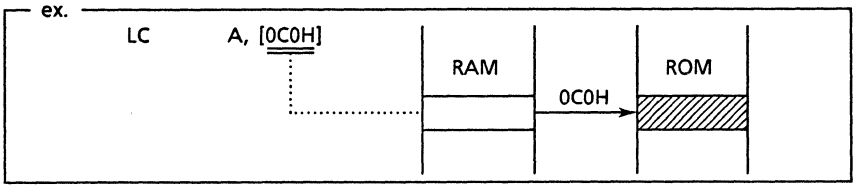
c) System Stack Pointer (SSP) Indirect



d) Local Register Base (LRB) Indirect

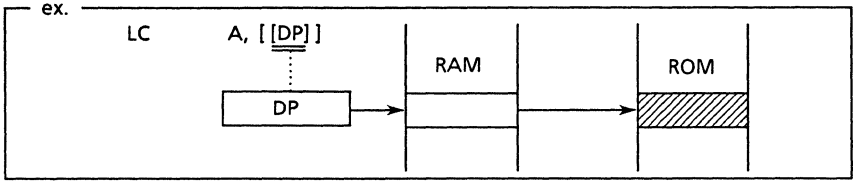


e) RAM Indirect

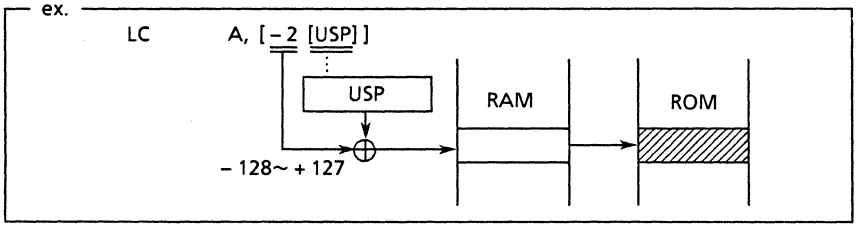


2.3 Double Indirect Addressing

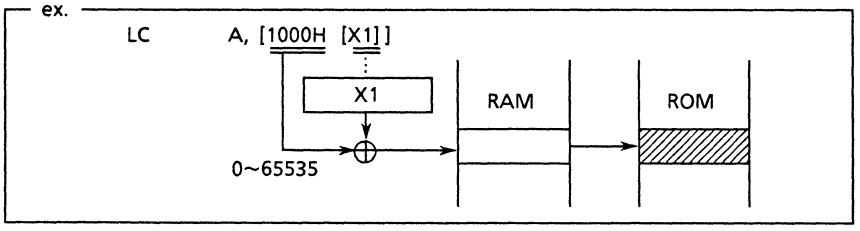
a) Data Pointer (DP) Double Indirect



b) User Stack Pointer (USP) Double Indirect



c) Index Register (X1, X2) Double Indirect



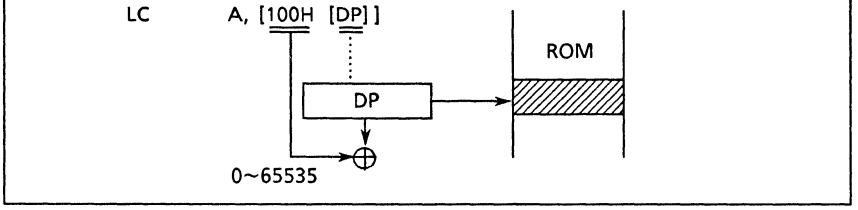
2.4 Indirect Addressing with 16 bit offset

a) Pointing Register Indirect

① Data Pointer (DP) Indirect

ex.

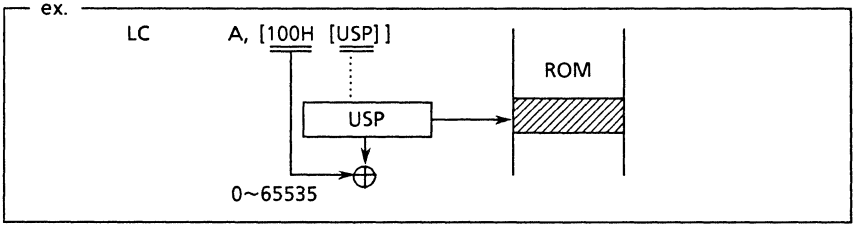
LC A, [100H [DP]]



② User Stack Pointer (USP) Indirect

ex.

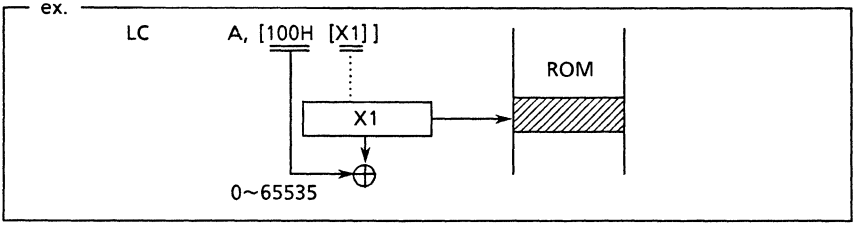
LC A, [100H [USP]]



③ Index Register (X1, X2) Indirect

ex.

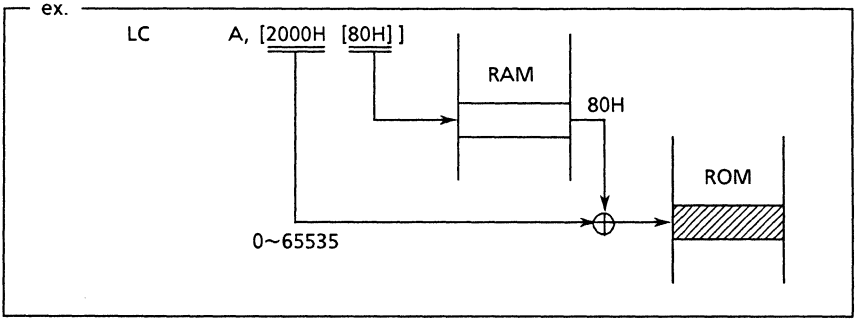
LC A, [100H [X1]]



b) RAM Indirect

ex.

LC A, [2000H [80H]]



INSTRUCTIONS

(Note) '*' Addressing mode expression
'#' and 'imme' Immediate value

Data Transfer

Instruction	Function
L A, *	A ← *
ST A, *	A → *
MOV *, A	A ← *
MOV *, #	* → Imme.
MOV er.*	er ← *
MOV d, *	(d) ← *
MOV DP, *	DP ← *
MOV X1, *	X1 ← *
MOV X2, *	X2 ← *
MOV USP, *	USP ← *
MOV PSW, *	PSW ← *
MOV SSP, *	SSP ← *
MOV LRB, *	LRB ← *
CLR *	* ← 0
SWAP	$A_{15-8} \leftrightarrow A_{7-0}$
XCHG A, *	A ↔ *
XNBL A, *	$A_{3-8} \leftrightarrow *_{3-0}$

Instruction	Function
LB A, *	A_L ← *
STB A, *	A_L → *
MOVB A, *	Al ← *
MOVB *, A	* ← AL
MOVB *, #	* ← Imme.
MOVB r, *	r ← *
MOVB d, *	(d) ← *
MOVB PSW0, *	PSW0 ← *
MOVB SCB, *	SCB ← *
CLRB *	* ← 0
SWAPB	$A_{15-8} \leftrightarrow A_{7-0}$
XCHGB A, *	A ↔ *

Push & Pop

Instruction	Function
PUSHS *, A	* → SYSTEM STACK
POPS *, #	* ← SYSTEM STACK

Rotate & Shift

Instruction	Function
ROL *	Rotate left
ROR *	Rotate right
SLL *	Shift left Logical
SRL *	Shift right Logical
SRA *	Shift right Arithmetic

Instruction	Function
ROLB *	Rotate left
RORB *	Rotate right
SLLB *	Shift left Logical
SRLB *	Shift right Logical
SRAB *	Shift right Arithmetic

Increment & Decrement

Instruction	Function
INC *	* ← * + 1
DEC *	* ← * – 1

Instruction	Function
INCB *	* ← * + 1
DECB *	* ← * – 1

ROM Table Reference

Instruction	Function
LC A, *	A ← * (ROM)
CMPC A, *	* – * (ROM)

Instruction	Function
LCB A, *	A_L ← * (ROM)
CMPCB A, *	A_L – * (ROM)

Arithmetic Operation

Instruction		Function
MUL		er1: A ← A x er0
DIV		er0: A ← er0: A/er2
ADD	A, *	A ← A + *
ADD	*, A	* ← * + A
ADD	*, d	* ← * + (d)
ADD	*, #	* ← * + Imme.
ADC	A, *	A ← A + * + C
ADC	*, A	* ← * + A + C
ADC	*, d	* ← * + (d) + C
ADC	*, #	* ← * + Imme. + C
SUB	A, *	A ← A − *
SUB	*, A	* ← * − A
SUB	*, d	* ← * − (d)
SUB	*, #	* ← * − Imme.
SBC	A, *	A ← A − * − C
SBC	*, A	* ← * − A − C
SBC	*, d	* ← * − (d) − C
SBC	*, #	* ← * − Imme. − C

Instruction		Function
MULB		A ← A_L x r0
DIVB		A ← A/r0
ADDB	A, *	A_L ← A_L + *
ADDB	*, A	* ← * + A_L
ADDB	*, d	* ← * + (d)
ADDB	*, #	* ← * + Imme.
ADCB	A, *	A_L ← A_L + * + C
ADCB	*, A	* ← * + A_L + C
ADCB	*, d	* ← * + (d) + C
ADCB	*, #	* ← * + Imme. + C
SUBB	A, *	A_L ← A_L − *
SUBB	*, A	* ← * − A_L
SUBB	*, d	* ← * − (d)
SUBB	*, #	* ← * − Imme.
SBCB	A, *	A_L ← A_L − * − C
SBCB	*, A	* ← * − A_L − C
SBCB	*, d	* ← * − (d) − C
SBCB	*, #	* ← * − Imme. − C

Logical Operation

Instruction	Function
AND A, *	A ← A and *
AND *, A	* ← * and A
AND *, d	* ← * and (d)
AND *, #	* ← * and Imme.
OR A, *	A ← A or *
OR *, A	* ← * or A
OR *, d	* ← * or (d)
OR *, #	* ← * or Imme.
XOR A, *	A ← A xor *
XOR *, A	* ← xor A
XOR *, d	* ← xor (d)
XOR *, #	* ← * xor Imme.

Instruction	Function
ANDB A, *	A_L ← A_L and *
ANDB *, A	* ← * and A_L
ANDB *, d	* ← * and (d)
ANDB *, #	* ← * and Imme.
ORB A, *	A_L ← A_L or *
ORB *, A	* ← * or A_L
ORB *, d	* ← * or (d)
ORB *, #	* ← * or Imme.
XORB A, *	A_L ← A_L xor *
XORB *, A	* ← * xor A_L
XORB *, d	* ← * xor (d)
XORB *, #	* ← * xor Imme.

Comparison

Instruction	Function
CMP A, *	A − *
CMP *, A	* − A
CMP *, d	* − (d)
CMP *, #	* − Imme.

Instruction	Function
CMPB A, *	A_L − *
CMPB *, A	* − A_L
CMPB *, d	* − (d)
CMPB *, #	* − Imme.

Decimal Adjust

Instruction	Function
DAA	Decimal Adjust for Addition
DAS	Decimal Adjust for Substruct

Data Type Conversion

Instruction	Function
EXTND	A15 – 8 ← An

Bit Operation

Instruction		Function
SBR	*	bit ← 1
RBR	*	bit ← 0
MBR	C, *	C ← bit
MBR	*, C	bit ← C
TRB	*, C	Z ← bit

Instruction		Function
SBR	*	bit ← 1
RBR	*	bit ← 0
MBR	C, *	C ← bit
MBR	*, C	bit ← C

Jump & Call

Instruction		Function
SJ	adrs	Short Jump
J	*	Jump
JC	EQ, adrs	Jump if '='
JC	LE, adrs	Jump if '<='
JC	GE, adrs	Jump if '>='
JBS	bit, adrs	Jump if bit-on
JRNZ	DP, adrs	Loop Function

Instruction		Function
SCAL	adrs	Short Call
CAL	*	Call Subroutine
JC	NE, adrs	Jump if '<>'
JC	LT, adrs	Jump if '<'
JC	GT, adrs	Jump if '>'
JBR	bit, adrs	Jump if bit-off
VCAL	table:adrs	Vector Call

Return

Instruction	Function
RT	Return from Subroutine
RTI	Return from Interrupt

Others

Instruction	Function
SC	C ← 1
SS	STACK FLAG ← 1
NOP	No Operation

Instruction	Function
RC	C ← 0
BRK	Software Reset

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	Ta = 25°C	− 0.3~7.0	V
Input Voltage	V_I		− 0.3~V_{DD} + 0.3	V
Output Voltage	V_O		− 0.3~V_{DD} + 0.3	V
Analog Ref. Voltage	V_R		− 0.3~V_{DD} + 0.3	V
Analog Input Voltage	V_{AI}		− 0.3~V_R	V
Power Dissipation	P_D	Ta = 25°C (per package)	400 MAX	mW
		Ta = 25°C (per out)	50 MAX	mW
Storage Temperature	T_{STG}	−	− 55~ + 150	°C

OPERATING CONDITIONS

Parameter	Symbol	Conditions		Limits	Unit
Supply Voltage	V_{DD}	$f_{(OSC)} \leqq 10$ MHz		4.5~5.5	V
Memory Hold Voltage	V_{DDH}	$f_{(OSC)} = 0$ Hz		2~5.5	V
Operating Frequency	$f_{(OSC)}$	$V_{DD} = 5V \pm 10\%$		0~10	MHz
Ambient Temperature	Ta	MSM66207		− 40~ + 85	°C
		MSM66P207		0~ + 70	
Fan Out	P_D	MOS load		20	
		TTL load	P0	2	
			P1, P2, P3, P4	1	

Preliminary

OKI semiconductor
MSM66301

OKI ORIGINAL HIGH PERFORMANCE CMOS SINGLE CHIP 8/16 BIT MICROCONTROLLER

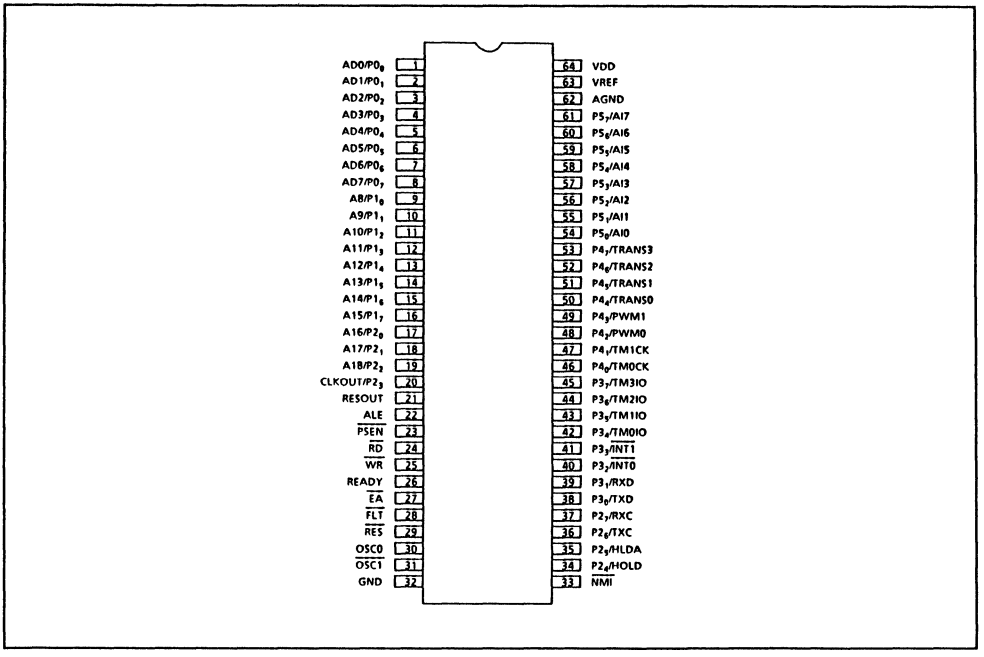
GENERAL DESCRIPTION

The OKI MSM66301 is a new generation, high performance single chip microcontroller implemented in silicon gate complementary metal oxide semiconductor technology (CMOS). Integrated within this chip are a 16-bit ALU, 16K bytes of mask program ROM, 512 bytes of data RAM, 48 I/O lines, built-in 16-bit timers, 10-bit A/D converter, serial I/O port, pulse width modulator (PWM), and an oscillator.

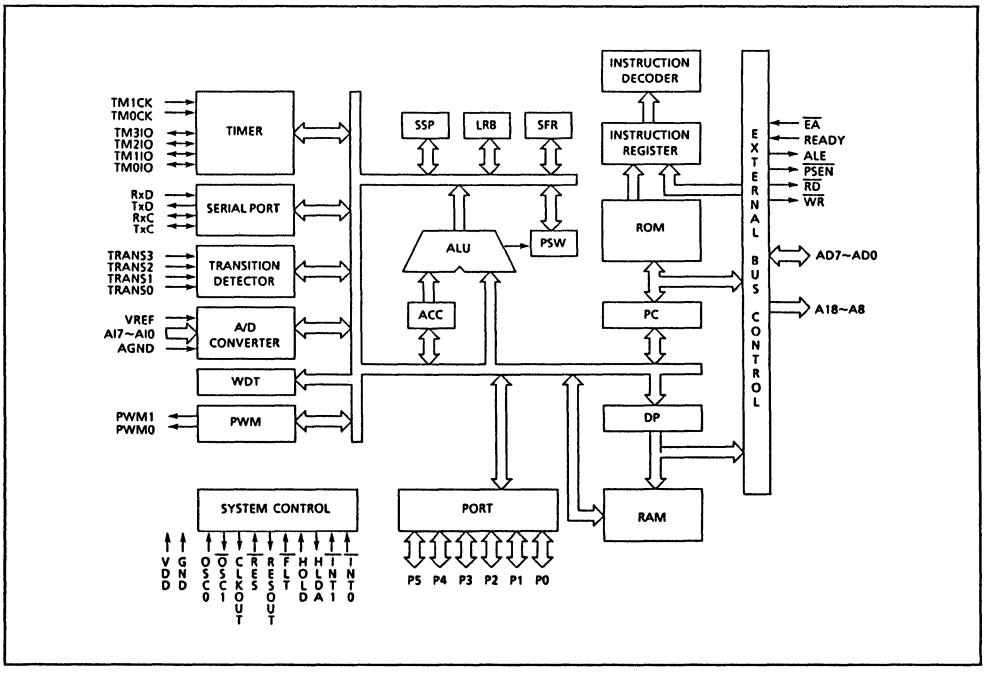
FEATURES

- 8-Bit External Data Bus Interface
- 16-Bit Internal Architecture
- 64K address space for program memory (including 16K bytes on-chip ROM)
- 512K address space for data memory (including 512 bytes on-chip RAM)
- High speed execution
  Minimum Cycle for Instruction:
  400ns (10MHz)
- The Abundance of Powerful Instructions
  8/16 data transfer operation
  8/16 bit arithmetic operation
  16(8) bit x 16(8) bit→32(16) bit
  32(16) bit x 16(8) bit→32(16) bit
  16(8) bit ± 16(8) bit→16(8) bit
  8/16 logic operation
  Bit operation
  String operation
  User stack operation
  ROM table access operation
- The same instruction allows both byte and word width operation according to Data Descriptor.
  That is to say, the same algorithm and the same source program lines are applicable to byte and word width data manipulation with only changing Data Descriptor.
- Many Addressing Modes
- 8 Input lines 40 Input/Output lines
- Built-in 16 bit timer x 4
  Each timer has the following 4 modes.
  Auto reload timer mode
  Clock output mode
  Capture register mode
  Real time output mode
- Serial Port x 1 ch.
  (variable bit length, baud rate generators for transmitter & receiver)
  Asynchronous normal mode
  Asynchronous multi processor communication mode
  Synchronous normal mode
  Synchronous multi processor communication mode
- 16 bit Pulse Width Modulator x 2
- Transition Detector x 4
- 10 bit A/D converter (8 channel)
- 1 non-maskable interrupt, 16 maskable interrupts
- Stand-by Function
  Software Clock stop
  Software CPU stop
  Hardware CPU stop
- Package:
  64 pin shrink DIP (SDIP64-P-750)
  64 pin plastic QFP (QFP64-P-1414-K)
  68 pin PLCC (QFJ68-P-S950)

PIN CONFIGURATION



FUNCTIONAL BLOCK DIAGRAM



PIN DESCRIPTION

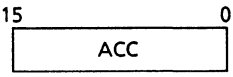
Designation	Input/Output	Function
$P0_0$ – $P0_7$/		P0: 8-bit I/O port. Each bit can be assigned to input or output.
AD0 – AD7	I/O	AD: Outputs the lower 8 bits of program counter during external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also outputs the address, Outputs or inputs data during an external data memory access instruction, under the control of ALE, $\overline{RD}$, and $\overline{WR}$.
$P1_0$ – $P1_7$/	I/O	P1: 8-bit I/O port. Each bit can be assigned to input or output.
A8 – A15		A: Outputs the upper 8 bits of program counter (PC_{8-15}) during external program memory fetch. Also outputs the middle 8 bits of address during an external data memory access instructions.
$P2_0$ – $P2_2$/		P2: 8-bit I/O port. Each bit can be assigned to input or output.
A16 – A18		A: Outputs the upper 3 bits of address during external data memory access instructions.
$P2_3$/CLKOUT		CLKOUT: Clock output pin. Output frequency range is equal to or twice the system clock.
$P2_4$/HOLD	I/O	HOLD: Input pin to request the CPU to enter the hardware power-down state.
$P2_5$/HLDA		HLDA: HOLD ACKNOWLEDGE: the HLDA signal appears in response to the HOLD signal and indicates that the CPU has entered the power-down state.
$P2_6$/TxC		TxC: Transmitter clock input/output pin.
$P2_7$/RxC		RxC: Receiver clock input/output pin.
$P3_0$/TxD		P3: 8-bit I/O port. Each bit can be assigned to input or output.
$P3_1$/RxD		TxD: Transmitter data output pin.
$P3_2$/$\overline{INT0}$		RxD: Receiver data input pin.
$P3_3$/$\overline{INT1}$	I/O	$\overline{INT}$: Interrupt Request Input pin. Falling edge trigger or level trigger is selectable.
$P3_4$/TM0IO		TM0IO~TM3IO: One of the following signals is output or input.
$P3_5$/TM1IO		• clock twice the frequency range of the 16 bit timer overflow
$P3_6$/TM2IO		• load trigger signal to the capture register input
		• setting value output
$P3_7$/TM3IO		Whether the signal is input or output depends on the mode.

PIN DESCRIPTION (Continued)

Designation	Input/Output	Function
$P4_0$/TM0CK $P4_1$/TM1CK $P4_2$/PWM0 $P4_3$/PWM1 $P4_4$ – $P4_7$/ TRANS0 – TRANS3	I/O	P4: 8-bit I/O port. Each bit can be assigned to input or output TM0CK, TM1CK: Clock input pins of timer 0, timer 1. TRANS: Transition Detector. The input pins which sense the falling edge and set the flag. PWM: Pulse Wide Modulator output pin.
$P5_0$ – $P5_7$/ AI0 – AI7	INPUT	P5: 8-bit input port. AI: Analog signal input pin for A/D converter.
RESOUT	OUTPUT	Output 'H' level when the CPU is in RESET cycle. Reset to 'L' level by program.
ALE	OUTPUT	Address Latch Enable: The timing pulse to latch the lower 8 bit of the address output from port 0 when the CPU accesses the external memory.
$\overline{PSEN}$	OUTPUT	Program Store Enable: The strobe pulse to fetch to external program memory.
$\overline{RD}$	OUTPUT	Output strobe activated during a bus read cycle. Used to enable data on to the bus from the external data memory.
$\overline{WR}$	OUTPUT	Output strobe during a bus write cycle. Used as write strobe to external data memory.
READY	INPUT	Used when the CPU accesses low speed peripherals.
$\overline{EA}$	INPUT	Normaly set to 'H' level. If set to 'L' level, the CPU fetches the code from external program memory.
$\overline{FLT}$	INPUT	If $\overline{FLT}$ is 'H' level, ALE, $\overline{WR}$, $\overline{RD}$, $\overline{PSEN}$ are set 'H' level when reset. If $\overline{FLT}$ is set to 'L', ALE, $\overline{WR}$, $\overline{RD}$, $\overline{PSEN}$ are set to floating level when reset.
$\overline{RES}$	INPUT	RESET input pin.
OSC_0, $\overline{OSC_1}$		Oscillation circuit input and output.
$\overline{NMI}$	INPUT	Non maskable interrupt input pin (falling edge)
VREF		Reference voltage input pin for A/D converter
AGND		Ground for A/D converter
VDD		System power supply
GND		Ground

REGISTERS

● ACCUMULATOR



● CONTROL REGISTERS (CR)

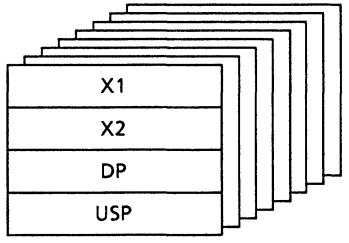
PROGRAM STATUS WORD — PSW

PROGRAM COUNTER — PC

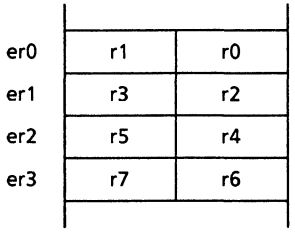
LOCAL REGISTER BASE — LRB

SYSTEM STACK POINTER — SSP

● POINTING REGISTERS (PR)



● LOCAL REGISTERS



● SPECIAL FUNCTION REGISTERS (SFR)

All of the I/O functions are controlled by SFRs. Also, some of the internal functions (Timer, WDT, etc, ...) are controlled by SFRs. SFRs are located in the top of RAM space (000H~007FH).

MSM66301 SPECIAL FUNCTION REGISTERS [1]

Address (HEX)	SFR Name	Abbreviat -ed Name	R/W	8/16-bit operation	Reset Value (HEX)
0000	System Stack Printer	SSP (ASSP)	R/W	8/16	FFH
0001					FFH
0002	Local Register Base	LRB (ALRB)			undefined
0003					
0004	Program Status Word	PSWL (APSW)			C8H
0005		PSWH			0CH
0006	Accumulator	ACC			00H
0007					00H
0008	Source index register (for string operation)	SI		16	undefined
0009					
000A	Destination index register (for string operation)	DI			
000B					
000C	Count register (for string operation)	CX			
000D					
000E	Source/destination bank register (for string operation)	BSDI		8	
0010	Standby control register	SBYCON			F8H or F0H
0011	Watchdog timer	WDT	W		00H/WDT is stopped
0012	Peripheral control register	PRPHF	R/W		FDH
0013	Stop code acceptor	STPACP	W	8	"0"
0018	Interrupt request flag	IRQ	R/W	8/16	00H
0019					00H
001A	Interrupt Enable flag	IE			00H
001B					00H
001C	External interrupt control register	EXICON		8	FCH
0020	Port 0 data register	P0			undefined
0021	Port 0 mode register	P0IO			00H
0022	Port 1 data register	P1			undefined
0023	Port 1 mode register	P1IO			00H
0024	Port 2 data register	P2			undefined
0025	Port 2 mode register	P2IO			00H
0026	Port 2 secondary function control register	P2SF			07H
0028	Port 3 data register	P3			undefined
0029	Port 3 mode register	P3IO			00H
002A	Port 3 secondary function control register	P3SF			00H
002C	Port 4 data register	P4			undefined
002D	Port 4 mode register	P4IO			00H
002E	Port 4 secondary function control register	P4SF			00H

Address (HEX)	SFR Name	Abbreviat -ed Name	R/W	8/16-bit operation	Reset Value (HEX)
002F	Port 5	P5	R	8	–
0030	Timer 0 counter	TM0	R/W	16	00H
0031					00H
0032	Timer 0 register	TMR0			00H
0033					00H
0034	Timer 1 counter	TM1			00H
0035					00H
0036	Timer 1 register	TMR1			00H
0037					00H
0038	Timer 2 counter	TM2			00H
0039					00H
003A	Timer 2 register	TMR2			00H
003B					00H
003C	Timer 3 counter	TM3			00H
003D					00H
003E	Timer 3 register	TMR3			00H
003F					00H
0040	Timer 0 Control register	TC0N0	R/W	8	00H
0041	Timer 1 Control register	TC0N1			00H
0042	Timer 2 Control register	TC0N2			00H
0043	Timer 3 Control register	TC0N3			00H
0046	Transition detector register	TRNSIT			undefined
0048	Serial port transmission baud rate generator counter	STTM			00H
0049	Serial port transmission baud rate generator register	STTMR			00H
004A	Serial port transmission control register	STTMC			0CH
004C	Serial port receiving baud rate generator counter	SRTM			00H
004D	Serial port receiving baud rate generator register	SRTMR			00H
004E	Serial port receiving control register	SRTMC			0EH
0050	Serial port transmission mode control register	STCON			80H
0051	Serial port transmission data buffer register	STBUF	W		undefined
0054	Serial port receiving mode control register	SRCON	R/W		00H
0055	Serial port receiving data buffer register	SRBUF	R		undefined
0056	Serial port receiving error register	SRSTAT	R/W		F0H
0058	A/D scan mode register	ADSCAN			80H
0059	A/D select mode register	ADSEL			A0H

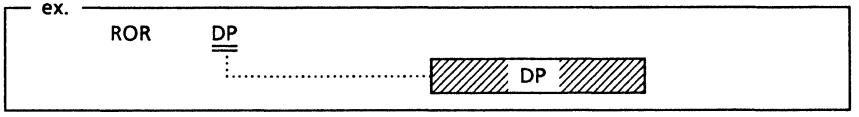
Address (HEX)	SFR Name	Abbreviat -ed Name	R/W	8/16-bit operation	Reset Value (HEX)
0060	A/D conversion result register 0	ADCR0	R	8/16	undefined
0061					
0062	A/D conversion result register 1	ADCR1			
0063					
0064	A/D conversion result register 2	ADCR2			
0065					
0066	A/D conversion result register 3	ADCR3			
0067					
0068	A/D conversion result register 4	ADCR4			
0069					
006A	A/D conversion result register 5	ADCR5			
006B					
006C	A/D conversion result register 6	ADCR6			
006D					
006E	A/D conversion result register 7	ADCR7			
006F					
0070	PWM 0 counter	PWMC0	R/W		00H
0071					00H
0072	PWM 0 register	PWMR0			00H
0073					00H
0074	PWM 1 counter	PWMC1			00H
0075					00H
0076	PWM 1 register	PWMR1			00H
0077					00H
0078	PWM 0 control register	PWC0N0		8	00H
007A	PWM 1 control register	PWC0N1			00H

ADDRESSING MODE

The MSM66301 supports 512KB (64KB x 8 BANKs) of data space and 64KB of program space with verious types of addressing modes. These modes divide into the following types.

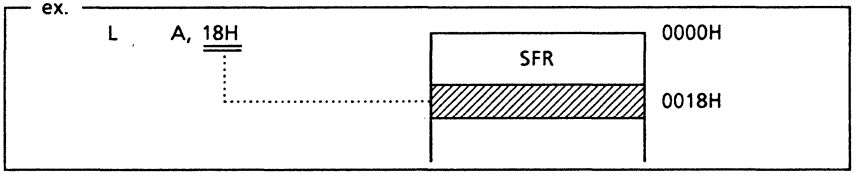
1. RAM ADDRESSING (FOR DATA SPACE)

1.1 Register Direct Addressing

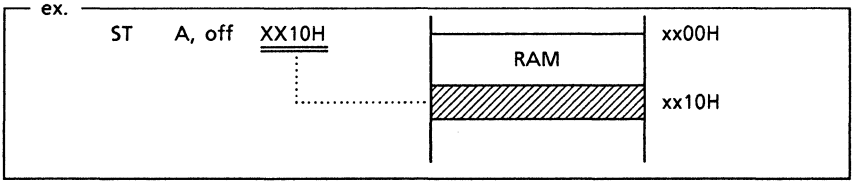


1.2 Displacement Addressing

a) Zero Page

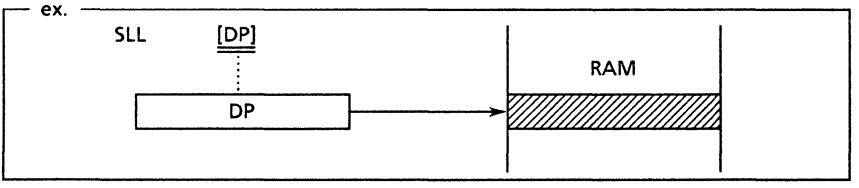


b) Direct Page

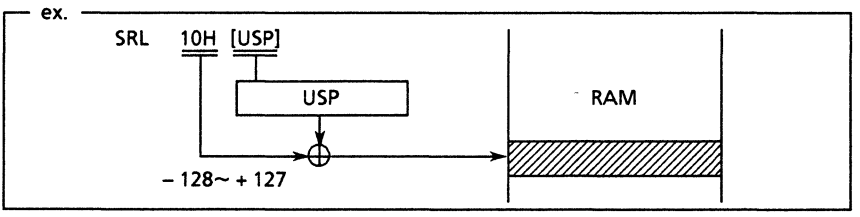


1.3 Pointing Register Indirect Addressing

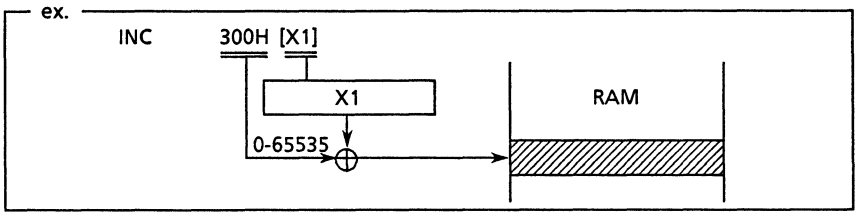
a) Data Pointer (DP) Indirect



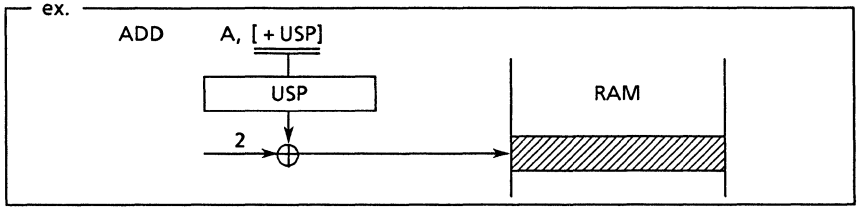
b) User Stack Pointer (USP) Indirect



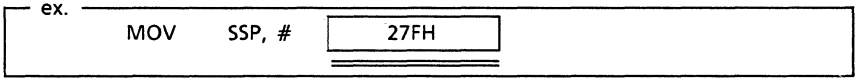
c) Index register (X1, X2) Indirect



d) User Stack Painter (USP) indirect with preincrement

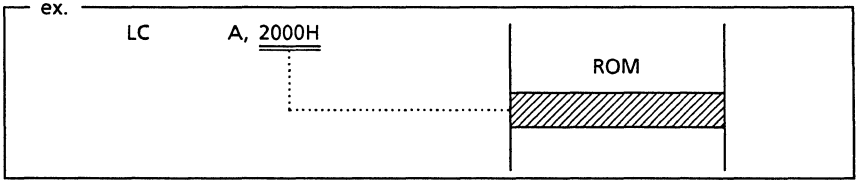


1.4 Immediate Addressing



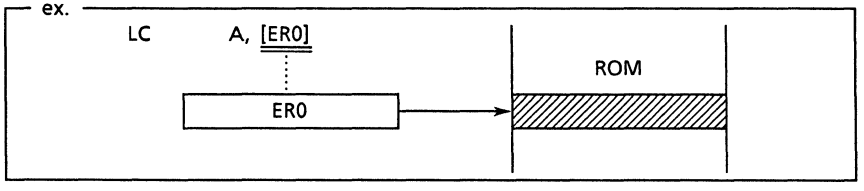
2. ROM ADDRESSING (FOR PROGRAM SPACE)

2.1 Direct Addressing



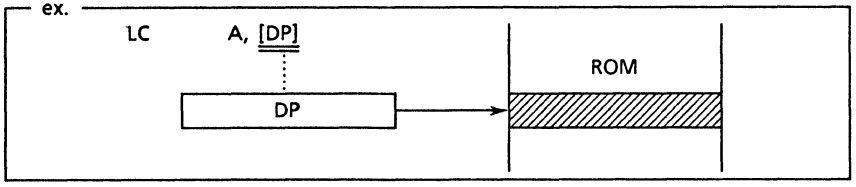
2.2 Simple Indirect Addressing

a) Local Register Indirect

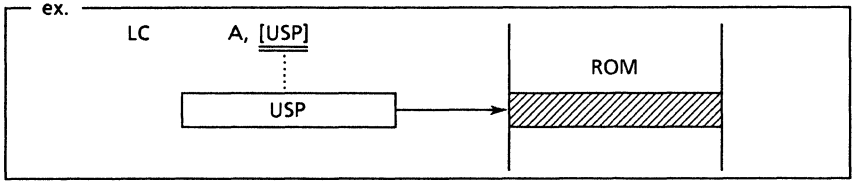


b) Pointing Register Indirect

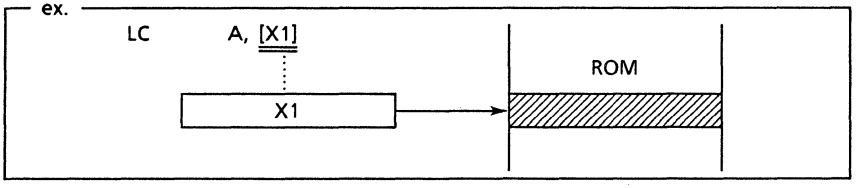
① Data Pointer (DP) Indirect



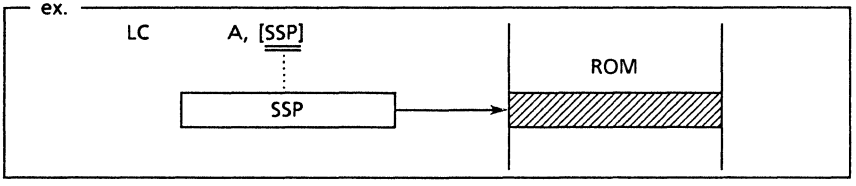
② User Stack Pointer (USP) Indirect



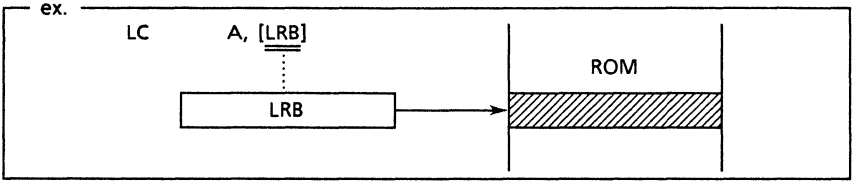
③ Index Register (X1, X2) Indirect



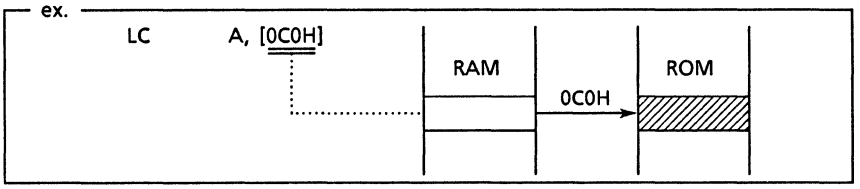
c) System Stack Pointer (SSP) Indirect



d) Local Register Base (LRB) Indirect

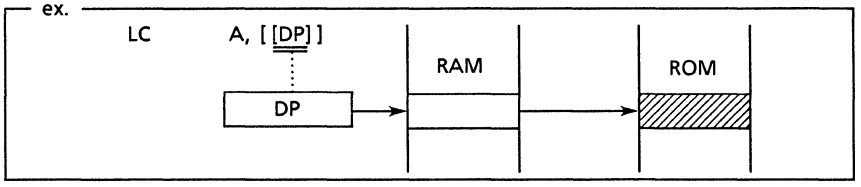


e) RAM Indirect

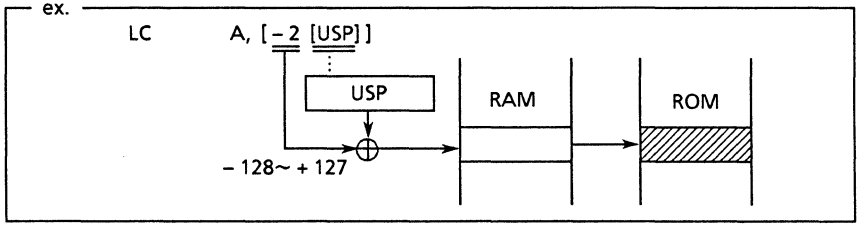


2.3 Double Indirect Addressing

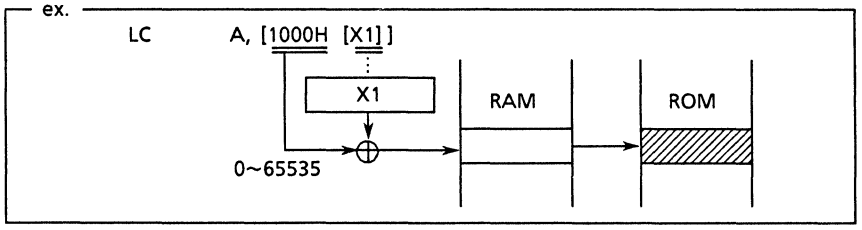
a) Data Pointer (DP) Double Indirect



b) User Stack Pointer (USP) Double Indirect



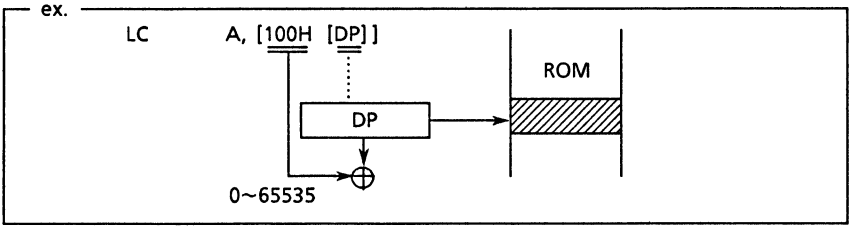
c) Index Register (X1, X2) Double Indirect



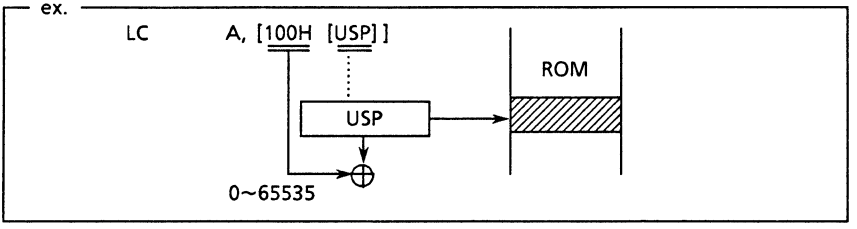
2.4 Indirect Addressing with 16 bit offset

a) Pointing Register Indirect

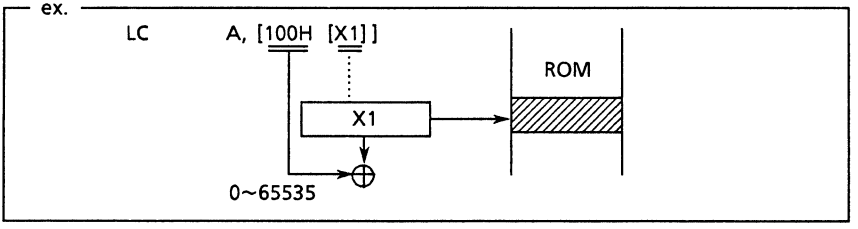
① Data Pointer (DP) Indirect



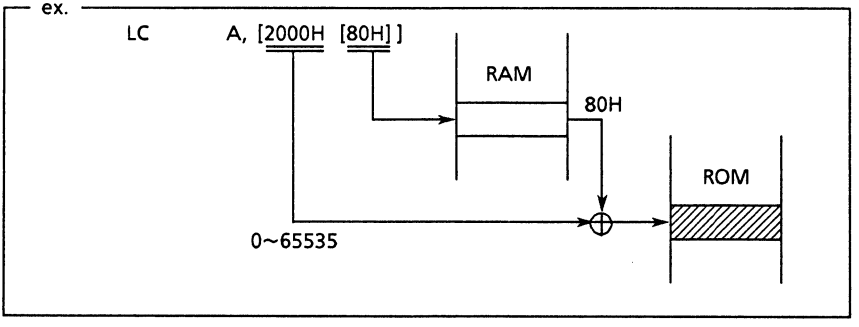
② User Stack Pointer (USP) Indirect



③ Index Register (X1, X2) Indirect



b) RAM Indirect



INSTRUCTIONS

(Note) '*' Addressing mode expression
'#' and 'imme' Immediate value

Data Transfer

Instruction	Function
L A, *	A ← *
ST A, *	A → *
MOV *, A	A ← *
MOV *, #	* → Imme.
MOV er.*	er ← *
MOV d, *	(d) ← *
MOV DP, *	DP ← *
MOV X1, *	X1 ← *
MOV X2, *	X2 ← *
MOV USP, *	USP ← *
MOV PSW, *	PSW ← *
MOV SSP, *	SSP ← *
MOV LRB, *	LRB ← *
CLR *	* ← 0
SWAP	$A_{15-8} \leftrightarrow A_{7-0}$
XCHG A, *	A ↔ *
XNBL A, *	$A_{3-8} \leftrightarrow *_{3-0}$
TRNS USP, LRB	$USP_{15-3} \leftarrow LRB_{12-0}$ $USP_{2-0} \leftarrow 0$
TRNS LRB, USP	$LRB_{12-0} \leftarrow USP_{15-3}$ $LRB_{15-13} \leftarrow 0$

Instruction	Function
LB A, *	$A_L \leftarrow$ *
STB A, *	$A_L \rightarrow$ *
MOVB A, *	Al ← *
MOVB *, A	* ← AL
MOVB *, #	* ← Imme.
MOVB r, *	r ← *
MOVB d, *	(d) ← *
MOVB PSW0, *	PSW0 ← *
MOVB SCB, *	SCB ← *
CLRB *	* ← 0
SWAPB	$A_{15-8} \leftrightarrow A_{7-0}$
XCHGB A, *	A ↔ *

Push & Pop

Instruction	Function
PUSHU *	* → USER STACK
POPU A	A ← USER STACK
PUSHS *, A	* → SYSTEM STACK
POPS *, #	* ← SYSTEM STACK

Instruction	Function
PUSHUB *	* → USER STACK
POPUB A	A_L ← USER STACK

Rotate & Shift

Instruction	Function
ROL *	Rotate left
ROR *	Rotate right
SLL *	Shift left Logical
SRL *	Shift right Logical
SRA *	Shift right Arithmetic

Instruction	Function
ROLB *	Rotate left
RORB *	Rotate right
SLLB *	Shift left Logical
SRLB *	Shift right Logical
SRAB *	Shift right Arithmetic

Increment & Decrement

Instruction	Function
INC *	* ← * + 1
DEC *	* ← * − 1

Instruction	Function
INCB *	* ← * + 1
DECB *	* ← * − 1

ROM Table Reference

Instruction	Function
LC A, *	A ← * (ROM)
CMPC A, *	* − * (ROM)

Instruction	Function
LCB A, *	A_L ← * (ROM)
CMPCB A, *	A_L − * (ROM)

Arithmetic Operation

Instruction		Function
MUL		er1: A ← A x er0
DIV		er0: A ← er0: A/er2
ADD	A, *	A ← A + *
ADD	*, A	* ← * + A
ADD	*, d	* ← * + (d)
ADD	*, #	* ← * + Imme.
ADC	A, *	A ← A + * + C
ADC	*, A	* ← * + A + C
ADC	*, d	* ← * + (d) + C
ADC	*, #	* ← * + Imme. + C
SUB	A, *	A ← A − *
SUB	*, A	* ← * − A
SUB	*, d	* ← * − (d)
SUB	*, #	* ← * − Imme.
SBC	A, *	A ← A − * − C
SBC	*, A	* ← * − A − C
SBC	*, d	* ← * − (d) − C
SBC	*, #	* ← * − Imme. − C

Instruction		Function
MULB		A ← A_L x r0
DIVB		A ← A/r0
ADDB	A, *	A_L ← A_L + *
ADDB	*, A	* ← * + A_L
ADDB	*, d	* ← * + (d)
ADDB	*, #	* ← * + Imme.
ADCB	A, *	A_L ← A_L + * + C
ADCB	*, A	* ← * + A_L + C
ADCB	*, d	* ← * + (d) + C
ADCB	*, #	* ← * + Imme. + C
SUBB	A, *	A_L ← A_L − *
SUBB	*, A	* ← * − A_L
SUBB	*, d	* ← * − (d)
SUBB	*, #	* ← * − Imme.
SBCB	A, *	A_L ← A_L − * − C
SBCB	*, A	* ← * − A_L − C
SBCB	*, d	* ← * − (d) − C
SBCB	*, #	* ← * − Imme. − C

Logical Operation

Instruction		Function
AND	A, *	A ← A and *
AND	*, A	* ← * and A
AND	*, d	* ← * and (d)
AND	*, #	* ← * and Imme.
OR	A, *	A ← A or *
OR	*, A	* ← * or A
OR	*, d	* ← * or (d)
OR	*, #	* ← * or Imme.
XOR	A, *	A ← A xor *
XOR	*, A	* ← xor A
XOR	*, d	* ← xor (d)
XOR	*, #	* ← * xor Imme.

Instruction		Function
ANDB	A, *	A_L ← A_L and *
ANDB	*, A	* ← * and A_L
ANDB	*, d	* ← * and (d)
ANDB	*, #	* ← * and Imme.
ORB	A, *	A_L ← A_L or *
ORB	*, A	* ← * or A_L
ORB	*, d	* ← * or (d)
ORB	*, #	* ← * or Imme.
XORB	A, *	A_L ← A_L xor *
XORB	*, A	* ← * xor A_L
XORB	*, d	* ← * xor (d)
XORB	*, #	* ← * xor Imme.

Comparison

Instruction		Function
CMP	A, *	A − *
CMP	*, A	* − A
CMP	*, d	* − (d)
CMP	*, #	* − Imme.

Instruction		Function
CMPB	A, *	A_L − *
CMPB	*, A	* − A_L
CMPB	*, d	* − (d)
CMPB	*, #	* − Imme.

Stack Operation

Instruction	Function
ADD A, [+ USP]	USP ← USP + 2, A ← A + [USP]
ACD A, [+ USP]	USP ← USP + 2, A ← A + [USP] + C
SUB A, [+ USP]	USP ← USP + 2, A ← A – [USP]
SBC A, [+ USP]	USP ← USP + 2, A ← A + [USP] – C
AND A, [+ USP]	USP ← USP + 2, A ← A and [USP]
OR A, [+ USP]	USP ← USP + 2, A ← A or [USP]
XOR A, [+ USP]	USP ← USP + 2, A ← A_{xor} [USP]
CMP A, [+ USP]	USP ← USP + 2, A – [USP]

Decimal Adjust

Instruction	Function
DAA	Decimal Adjust for Addition
DAS	Decimal Adjust for Substruct

Data Type Conversion

Instruction	Function
EXTND	A15 – 8 ← An

Bit Operation

Instruction	Function
SBR *	bit ← 1
RBR *	bit ← 0
MBR C, *	C ← bit
MBR *, C	bit ← C
TRB *, C	Z ← bit

Instruction	Function
SBR *	bit ← 1
RBR *	bit ← 0
MBR C, *	C ← bit
MBR *, C	bit ← C

Execute

Instruction	Function
EX *	Execute Specified Data as the Instruction

Jump & Call

Instruction		Function
SJ	adrs	Short Jump
J	*	Jump
JC	EQ, adrs	Jump if '='
JC	LE, adrs	Jump if '<='
JC	GE, adrs	Jump if '>='
JBS	bit, adrs	Jump if bit-on
JRNZ	DP, adrs	Loop Function

Instruction		Function
SCAL	adrs	Short Call
CAL	*	Call Subroutine
JC	NE, adrs	Jump if '<>'
JC	LT, adrs	Jump if '<'
JC	GT, adrs	Jump if '>'
JBR	bit, adrs	Jump if bit-off
VCAL	table:adrs	Vector Call

Return

Instruction	Function
RT	Return from Subroutine
RTI	Return from Interrupt

String Operation

Instruction	Function
SMOVI	String Transfer with Increasing Pointers
SMOVD	String Transfer with Decreasing Pointers
SCMP	String Compare

Others

Instruction	Function
SC	C ← 1
SS	STACK FLAG ← 1
NOP	No Operation

Instruction	Function
RC	C ← 0
BRK	Software Reset

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	Ta = 25°C	− 0.3~7.0	V
Input Voltage	V_I		− 0.3~V_{DD} + 0.3	V
Output Voltage	V_O		− 0.3~V_{DD} + 0.3	V
Analog Ref. Voltage	V_R		− 0.3~V_{DD} + 0.3	V
Analog Input Voltage	V_{AI}		− 0.3~V_R	V
Power Dissipation	P_D	Ta = 25°C (per package)	400 MAX	mW
		Ta = 25°C (per out)	50 MAX	mW
Storage Temperature	T_{STG}	−	− 55~ + 150	°C

OPERATING CONDITIONS

Parameter	Symbol	Conditions		Limits	Unit
Supply Voltage	V_{DD}	$f_{(OSC)} \leqq$ 10 MHz		4.5~5.5	V
Memory Hold Voltage	V_{DDH}	$f_{(OSC)}$ = 0 Hz		2~5.5	V
Operating Frequency	$f_{(OSC)}$	V_{DD} = 5V ± 10%		0~10	MHz
Ambient Temperature	Ta	−		− 40~ + 85	°C
Fan Out	P_D	MOS load		20	
		TTL load	P0	2	
			P1, P2, P3, P4	1	

DC CHARACTERISTICS

(V_{DD} = 5V ± 10%, Ta = −40~+85°C)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
"H" Input voltage *1 *2 *3 *6	V_{IH}	–	2.4		V_{DD} + 0.3	V
"H" Input voltage *5 *7	V_{IH}	–	4.0		V_{DD} + 0.3	V
"H" Input voltage *8	V_{IH}	–	4.2		V_{DD} + 0.3	V
"L" Input voltage *1 *2 *3 *6	V_{IL}	–	−0.3		0.8	V
"L" Input voltage *5 *7	V_{IL}	–	−0.3		0.8	V
"L" Input voltage *8	V_{IL}	–	−0.3		0.4	V
"H" Outputput voltage *1 *4	V_{OH}	I_O = −400μA	4.2			V
"H" Output voltage *2	V_{OH}	I_O = −200μA	4.2			V
"L" Outputput voltage *1 *4	V_{OL}	I_O = 3.2mA			0.4	V
"L" Output voltage *2	V_{OL}	I_O = 1.6mA			0.4	V
Input Leak Current *3 *6 *7	I_{IH}/I_{IL}	$V_I = V_{DD}/0V$			1/−1	μA
Input Current *5	I_{IH}/I_{IL}	$V_I = V_{DD}/0V$			1/−20	μA
Input Current *8	I_{IH}/I_{IL}	$V_I = V_{DD}/0V$			10/−10	μA
"H" Output Current *1	I_{OH}	V_O = 2.4V	−2			mA
"H" Output Current *2	I_{OH}	V_O = 2.4V	−1			mA
"L" Output Current *1	I_{OL}	V_O = 2.4V	10			mA
"L" Output Current *2	I_{OL}	V_O = 2.4V	5			mA
Output Leak Current *1 *2 *4	I_{LO}	$V_O = V_{DD}/0V$			±2	μA
Input Capacity	C_I	f = 1MHz, Ta = 25°C		5		pF
Output Capacity	C_O	f = 1MHz, Ta = 25°C		7		pF
Current Consumption (STOP)	I_{DDS}	V_{DD} = 2V, Ta = 25°C **		0.2	10	μA
		**		1	100	μA
Current Consumption (HALT)	I_{DDH}	$f_{(OSC)}$ = 10MHz, No Load		6	10	mA
Current Consumption	I_{DD}	$f_{(OSC)}$ = 10MHz, No Load		20	35	mA

*1 Applied for P0
*2 Applied for P1, P2, P3, P4
*3 Applied for P5
*4 Applied for ALE, $\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$, RESOUT
*5 Applied for $\overline{RES}$, $\overline{NMI}$
*6 Applied for READY, $\overline{EA}$
*7 Applied for $\overline{FLT}$
*8 Applied for OSC_0
** Port for input terminal is V_{DD} or 0V. The other ports are non-loaded.

AC CHARACTERISTICS

External Program Memory Control

(V_{DD} = 5V ± 10%, Ta = −40~ +85°C)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Clock (OSC) Pulse	$t_{\phi W}$	–	50			nS
ALE Pulse Width	t_{AW}	CL = 50pF	$2t_{\phi W} - 20$			nS
$\overline{PSEN}$ Pulse Width	t_{PW}		$5t_{\phi W} - 20$			nS
$\overline{PSEN}$ Pulse Delay Time	t_{PAD}		$t_{\phi W} - 20$		$t_{\phi W} + 20$	nS
Low Address Set Up Time	t_{AAS}		$t_{\phi W} - 35$		$t_{\phi W} + 20$	nS
Low Address Hold Time	t_{AAH}		$t_{\phi W} - 20$		$t_{\phi W} + 40$	nS
High Address Delay Time	t_{AAD}		$t_{\phi W} - 20$		$t_{\phi W} + 40$	nS
High Address Hold Time	t_{APH}		$t_{\phi W} - 20$		$t_{\phi W} + 40$	nS
Instruction Set Up Time	t_{IS}		100			nS
Instruction Hold Time	t_{IH}		0		$t_{\phi W} - 20$	nS

External Data Memory Control

(V_{DD} = 5V ± 10%, Ta = −40~ +85°C)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Clock (OSC) Pulse	$t_{\phi W}$	–	50			nS
ALE Pulse Width	t_{AW}	CL = 50pF	$2t_{\phi W} - 20$			nS
$\overline{RD}$ Pulse Width	t_{RW}		$5t_{\phi W} - 20$			nS
$\overline{WR}$ Pulse Width	t_{WW}		$4t_{\phi W} - 20$			nS
$\overline{RD}$ Pulse Delay Time	t_{RAD}		$t_{\phi W} - 20$		$t_{\phi W} + 20$	nS
$\overline{WR}$ Pulse Delay Time	t_{WAD}		$t_{\phi W} - 20$		$t_{\phi W} + 20$	nS
Low Address Set Up Time	t_{AAS}		$t_{\phi W} - 35$		$t_{\phi W} + 20$	nS
Low Address Hold Time	t_{AAH}		$t_{\phi W} - 20$		$t_{\phi W} + 40$	nS
High Address Delay Time	t_{AAD}		$t_{\phi W} - 20$		$t_{\phi W} + 40$	nS
High Address Hold Time	t_{ARH}		$t_{\phi W} - 20$		$t_{\phi W} + 40$	nS
High Address Hold Time	t_{AWH}		$2t_{\phi W} - 20$		$2t_{\phi W} + 40$	nS
Memory Data Set Up Time	t_{MS}		100			nS
Memory Data Hold Time	t_{MH}		0		$t_{\phi W} - 20$	nS
Data Delay Time	t_{DD}		$t_{\phi W} - 20$		$t_{\phi W} + 40$	nS
Data Retention Time	t_{DH}		$2t_{\phi W} - 20$		$2t_{\phi W} + 40$	nS

OLMS-67K SERIES (OKI ORIGINAL)

Preliminary

OKI semiconductor
MSM67620/67P620

OKI ORIGINAL HIGH PERFORMANCE CMOS 16 BIT SINGLE CHIP MICROCONTROLLER

GENERAL DESCRIPTION

MSM67620 is a high-performance 16-bit single-chip microcontroller that employs Oki's original nX-16 CPU core, and integrates memory as well as peripheral functions on-chip. With a minimum instruction execution time of 200 ns (10MHz clock), the MSM67620 is capable of high-speed processing, and includes 16 K words (16 x 16 K) of program memory, 256 words (16 x 256) of data memory, 8-bit timers, 16-bit timers, capture input/compare output, a serial port, and other functions on chip. Also available is the MSM67P620, which replaces the on-chip program memory with one-time PROM.

FEATURES

CPU

- 16-bit CPU using Oki original architecture
- General Register Machine
  Four banks of eight 16-bit registers
- Data Types
  Bit*, byte (8 bits), word (16 bits), double word** (32 bits)
  * Bit operation instructions
  ** Multiplication/division instructions
- High-Speed Data Processing (10 MHz Clock)
  16-bit addition and subtraction 200 ns
  16-bit multiplication 2.3 µs
  32 ÷ 16 bit division 2.3 µs
- 3-Words Instruction Queue
- Highly Orthogonal Instruction Set
- Bit Operation Instructions
- Easy-To-Use I/O Access Instruction
- Address Space
  Program Memory: 64 K words
  Data Memory: 64 K bytes

Peripheral Functions

- 8-bit auto reload timer x 2
- 16-bit auto reload timer x 3
- 4-channel capture input/compare output
- Serial port with 8-bit baud rate generator
  Asynchronous mode/Synchronous mode
- 8-bit parallel port x 7
  Bit-by-bit Input/Output specification

Interrupts

- 3 external interrupts (NMI, INT0, INT1)
- 10 internal interrupt factors
- 4 priority level settings

On-Chip Memory

- 256 words data memory
- 16K words program memory
  Mask ROM --- MSM67620
  One-time PROM --- MSM67P620

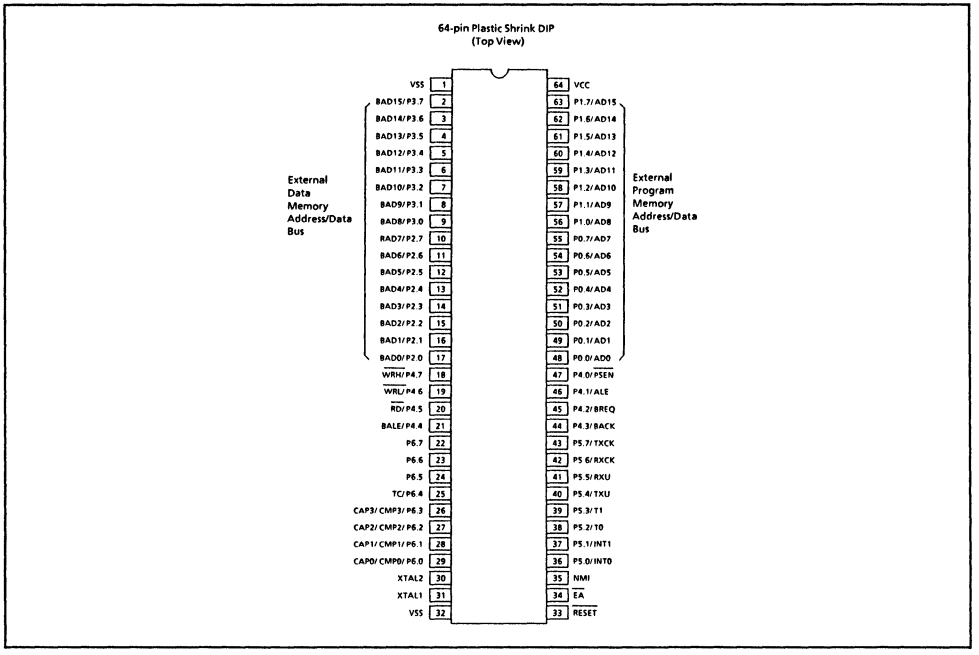
Power Down Modes

- Halt mode
  Terminates instruction execution but peripheral functions keep operating
- Stop mode
  Stop system clock oscillation

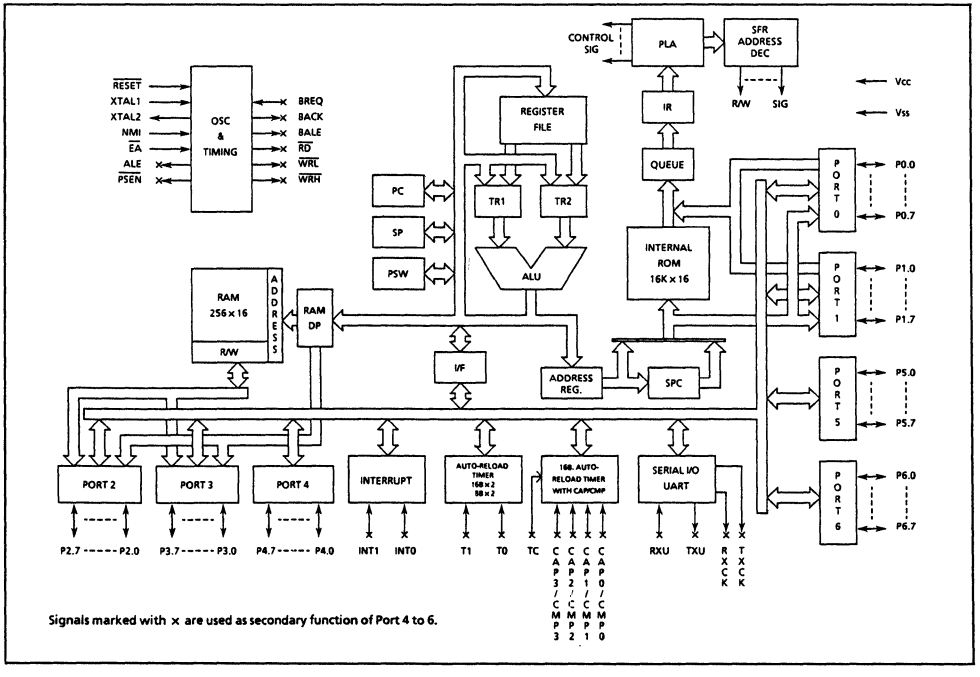
Packages

- 64 pin plastic shrink DIP (SDIP64-P-750)
- 64 pin plastic QFP (T.B.D)
- 68 pin PLCC (QFJ68-P-S950)

PIN CONFIGURATION



FUNCTIONAL BLOCK DIAGRAM



PIN DESCRIPTIONS

Type	Pin Name	I/O	Description
Port	P0.0~P0.7 (port 0)	I/O	8-bit bidirectional port that allows bit-by-bit specification for I/O. Secondary function of port 0 is external program memory address/data but.
	P1.0~P1.7 (port 1)	I/O	8-bit bidirectional port that allows bit-by-bit specification for I/O. Secondary function of port 1 is external program memory address/data bus.
	P2.0~P2.7 (port 2)	I/O	8-bit bidirectional port that allows bit-by-bit specification for I/O. Secondary function of port 2 is external data memory address/data bus.
	P3.0~P3.7 (port 3)	I/O	8-bit bidirectional port that allows bit-by-bit specification for I/O. Secondary function of port 3 is external data memory address/data bus.
	P4.0~P4.7 (port 4)	I/O	8-bit bidirectional port that allows bit-by-bit specification for I/O. Secondary functions of port 4 are external address/data bus control signal I/O pins.
	P5.0~P5.7 (port 5)	I/O	8-bit bidirectional port that allows bit-by-bit specification for I/O. Secondary functions of port 5 are on-chip timer clock input pins, serial port I/O pins; external interrupt input pins.
	P6.0~P6.7 (port 6)	I/O	8-bit bidirectional port that allows bit-by-bit specification for I/O. Secondary function of port 6 are capture input/compare output I/O pins.
System Control	$\overline{\text{RESET}}$	Input	System reset input pin. "L" level input performs system reset; pulled up to V_{CC} by resistance.
	$\overline{\text{EA}}$	Input	Program memory select input pin. "L" level input for external program memory; "H" level input for internal program memory. High impedance input.
	XTAL1	Input	System clock input pin. Quartz oscillator or ceramic oscillator is connected between XTAL1 and XTAL2. For external clock, input at XTAL1, leaving XTAL2 open.
	XTAL2	Output	System clock output pin.
Interrupt	NMI	Input	Non-maskable interrupt input pin. High impedance input.
Power Supply	V_{CC}	Input	+ 5V power supply pin.
	V_{SS}	Input	GND pins. Connect both to power supply (0 V).

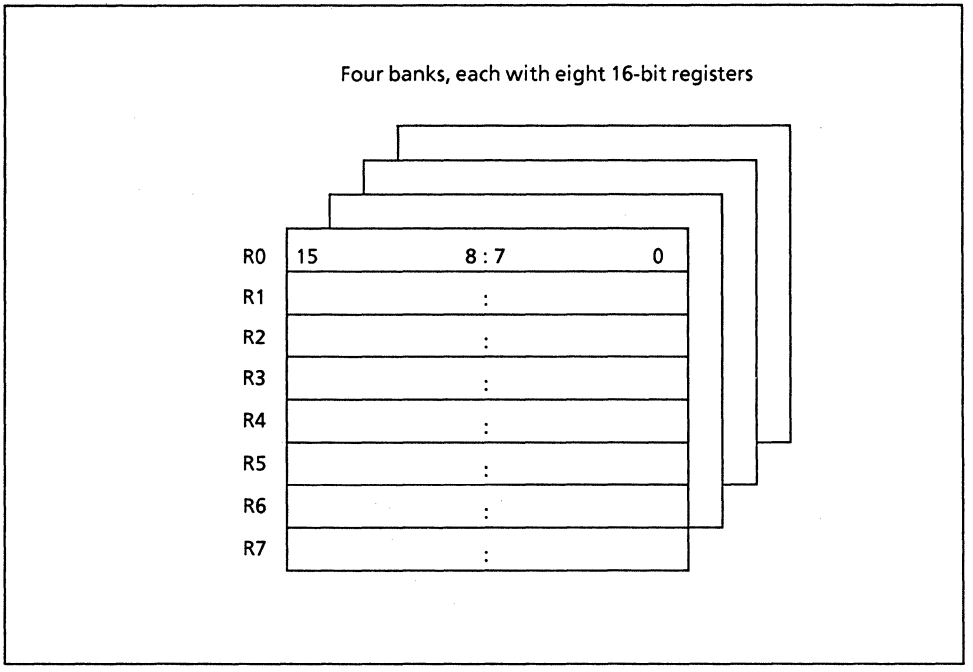
SECONDARY FUNCTIONS OF PORTS

Type	Pin Name	I/O	Description
Bus and Bus Control	AD0~AD7	I/O	Lower 8 bits of external program memory address/data bus. Port 0 secondary function (P0.0 to P0.7).
	AD8~AD15	I/O	Upper 8 bits of external program memory address/data bus. Port 1 secondary function (P1.0 to P1.7).
	$\overline{\text{PSEN}}$	Output	External program memory read strobe output pin; low active output. P4.0 secondary function.
	ALE	Output	External program memory address latch signal output pin. ALE latches address from multiplexed address/data bus AD0 through AD15. P4.1 secondary function.
	BAD0~BAD7	I/O	Lower 8 bits of external data memory address/data bus. Port 2 secondary function (P2.0 to P2.7).
	BAD8~BAD15	I/O	Upper 8 bits of external data memory address/data bus. Port 3 secondary function (3.0 to P3.7).
	BALE	Output	External data memory address latch signal output pin. BALE latches address from multiplexed address/data bus BAD0 through BAD15. P4.4 secondary function.
	$\overline{\text{RD}}$	Output	External data memory read strobe output pin; low active output. P4.5 secondary function.
	$\overline{\text{WRL}}$	Output	External data memory lower byte write strobe output pin; low active output. Secondary function of P4.6.
	$\overline{\text{WRH}}$	Output	External data memory upper byte write strobe output pin; low active output. Secondary function of P4.7.
	BREQ	Input	Bus request input pin. "H" level input opens address/data bus to external device. P4.2 secondary function.
	BACK	Output	Bus acknowledge output pin. High active output that indicates address/data bus is open to external device. P4.3 secondary function.

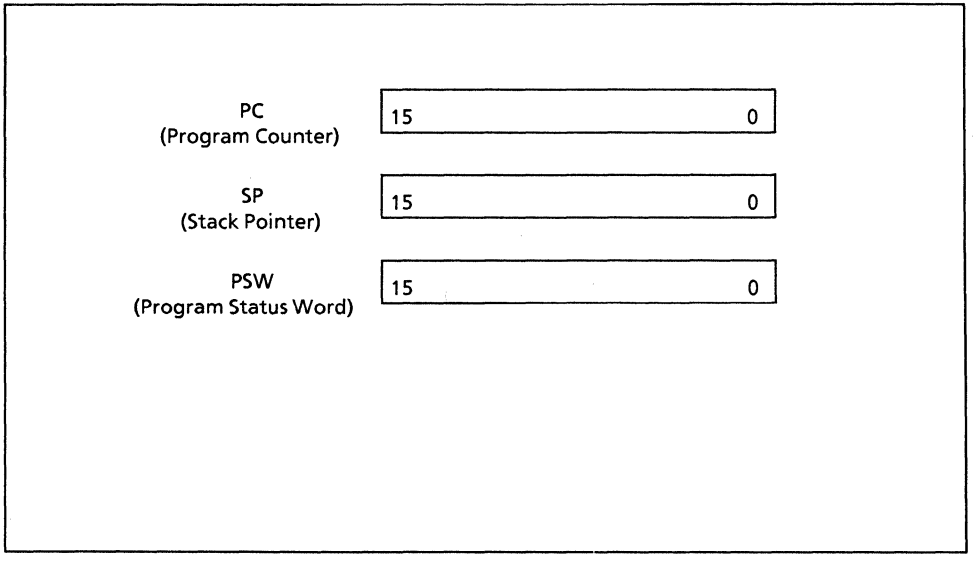
PROGRAM DEVELOPMENT SUPPORT TOOL SYSTEMS

REGISTERS

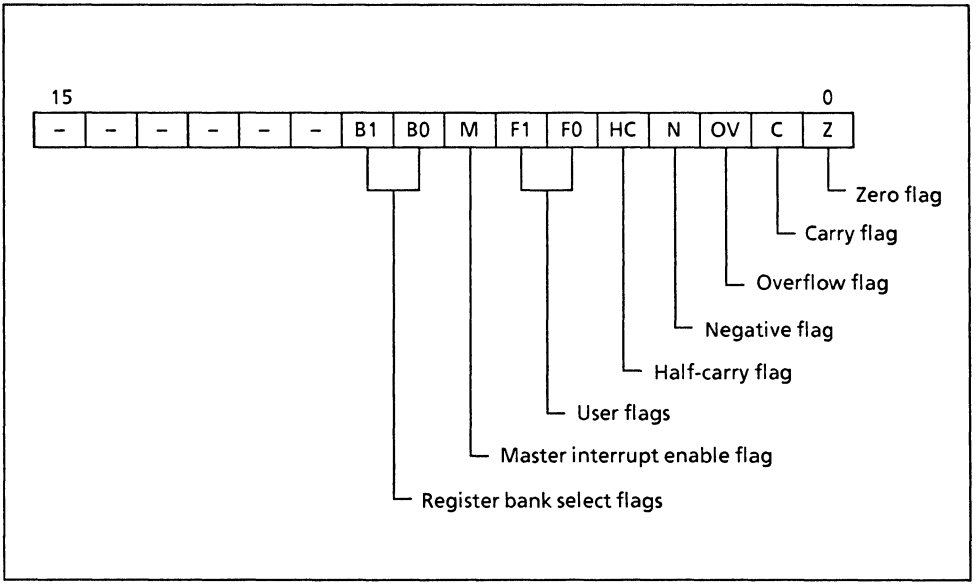
- General Purpose Registers (used as data registers and address registers)



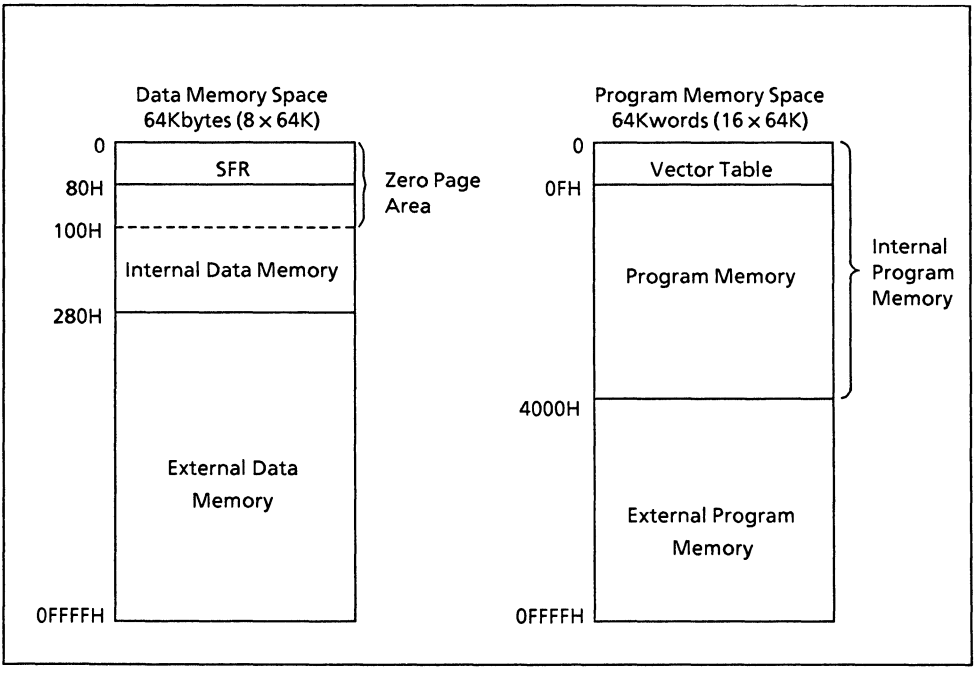
- Special Purpose Registers



● PSW Configuration



MEMORY MAP



SFR TABLE

Address (HEX)	Name	Symbol	Read/ Write	Word/ Byte	Reset
0000	Port 0	P0	R/W	B	undefined
0001	Port 0 direction register	P0DIR			00H
0002	Port 1	P1			undefined
0003	Port 1 direction register	P1DIR			00H
0004	Port 2	P2			undefined
0005	Port 2 direction register	P2DIR			00H
0006	Port 3	P3			undefined
0007	Port 3 direction register	P3DIR			00H
0008	Port 4	P4			0FXH
0009	Port 4 direction register	P4DIR			0F0H
000A	Port 5	P5			undefined
000B	Port 5 direction register	P5DIR			00H
000C	Port 6	P6			undefined
000D	Port 6 direction register	P6DIR			00H
000E	Port 4 mode register	P4MOD			0F3H
000F	Port 5 mode register	P5MOD			00H
0010	Interface control register	IFCON			0BBH
0011	External interrupt control register	XICON			0E0H
0012	Program status word	PSWL		W/B	00H
0013	* PSW during word operation	PSWH			0FCH
0014	Stack pointer	SPL			undefined
0015	* SP during word operation	SPH			undefined
0016	Interrupt priority register	IPL			0FFH
0017	* IP during word operation	IPH			0FFH
0018	Interrupt enable register	IEL			00H
0019	* IE during word operation	IEH			0F0H
001A	In-service priority register	ISPR	R	B	0F0H
001B	Serial port transmit/receive buffer	SBUF	R/W		undefined
001C	Serial port control register	SCON			00H
001D	Serial port status register	SSTAT			0E0H
001E	Baud rate counter	BRC			00H
001F	Baud rate register	BRR			00H
0020	Baud rate control register	BCON			0F0H
0021	Oscillation start control register	OSCON	W		undefined
0022	Timer 0 control register	T0CON	R/W		0E0H
0023	Timer 1 control register	T1CON			0E0H

R/W ; Read/write
R ; Read only
W ; Word only
W/B ; Word/byte access
W ; Write access only
B ; Byte access only

Address (HEX)	Name	Symbol	Read/ Write	Word/ Byte	Reset
0024	Timer 2 control register	T2CON	R/W	B	0E2H
0025	Timer 3 control register	T3CON			0E2H
0026	Capture timer control register	CTCON			0E0H
002A	System control register	SYSCON			0F3H
002B	Port 6 mode register	P6MOD			0E0H
0030	Capture control register	CAPCON			00H
0031	Capture interrupt flag register	CAPINT			0F0H
0032	Compare output 0 buffer	CMP0BF	W		0FEH
0033	Compare output 1 buffer	CMP1BF			0FEH
0034	Compare output 2 buffer	CMP2BF			0FEH
0035	Compare output 3 buffer	CMP3BF			0FEH
0060	Capture register 0	CAPR0	R/W	W	0000H
0062	Capture register 1	CAPR1			0000H
0064	Capture register 2	CAPR2			0000H
0066	Capture register 3	CAPR3			0000H
0070	Timer 0 counter	T0C			0000H
0072	Timer 0 register	T0R			0000H
0074	Timer 1 counter	T1C			0000H
0076	Timer 1 register	T1R			0000H
0078	Timer 2 counter	T2C		B	00H
0079	Timer 2 register	T2R			00H
007A	Timer 3 counter	T3C			00H
007B	Timer 3 register	T3R			00H
007C	Capture timer counter	CTC		W	0000H
007E	Capture timer register	CTR			0000H

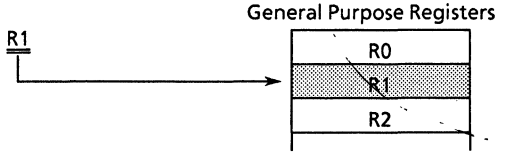
ADDRESSING MODES

MSM67620's instruction set allows use of eight addressing modes.

1. Register Direct Addressing Rm

Example

DAA R1



2. Immediate Addressing #n

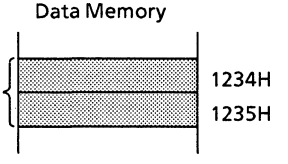
Example

MOV R0, #1234H

3. Direct Addressing adrs

Example

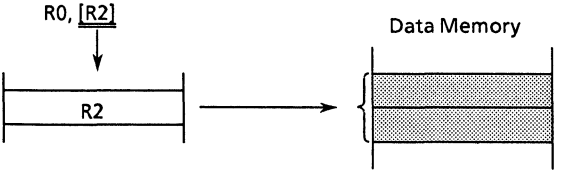
MOV R0,1234H



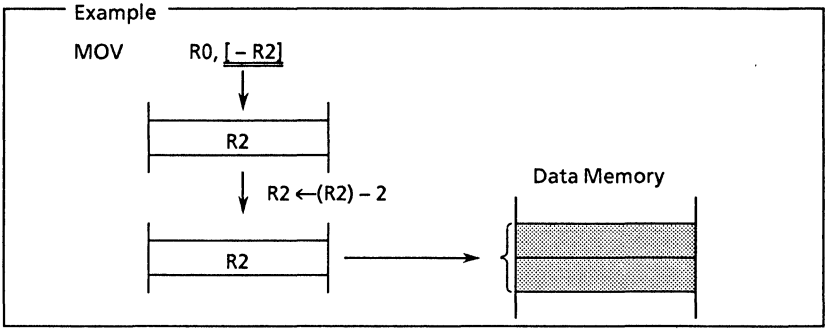
4. Indirect Addressing [Rm]

Example

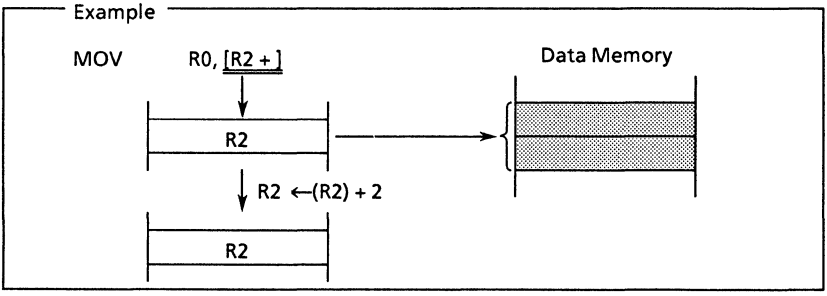
MOV R0, [R2]



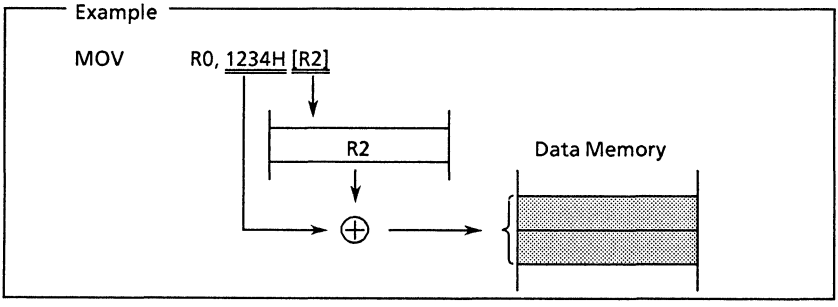
5. Pre-Decrement Addressing [– Rm]



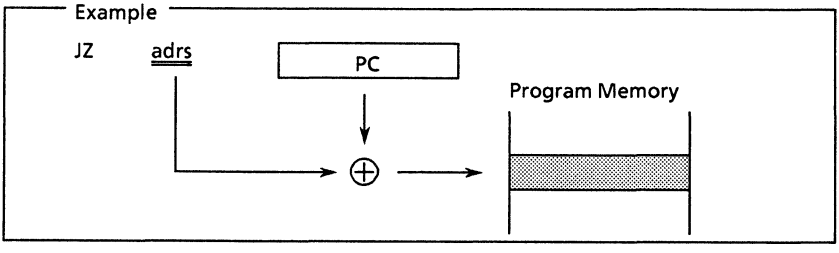
6. Post-Increment Addressing [Rm +]



7. Index Addressing adrs [Rm]



8. PC Relative Addressing adrs



INSTRUCTION TABLE

- Arithmetic Operations

Mnemonic		Function
ADD	ADDB	Data addition
ADDS	ADDSB	Short data addition (#1 to #8 addition)
ADC	ADCB	Data addition with carry
SUB	SUBB	Data subtraction
SUBS	SUBSB	Short data subtraction (#1 to #8 subtraction)
SBC	SBCB	Data subtraction with carry
MUL	MULB	Multiplication
DIV	DIVB	Division
DIVQ		Word quick division (16-bit quotient)
NEG	NEGB	Two's complement
DAA		Register decimal adjust after addition
DAS		Register decimal adjust after subtraction
CMP	CMPB	Comparison
TST	TSTB	Test (comparison with zero)
EXTS		Signed word extension
EXTU		Unsigned word extension

- Logical Operations

Mnemonic		Function
AND	ANDB	Logical AND
ANDZ	ANDZB	Zero page area immediate logical AND
OR	ORB	Logical OR
ORZ	ORZB	Zero page area immediate logical OR
XOR	XORB	Exclusive OR
XORZ	XORZB	Zero page area immediate exclusive OR
NOT	NOTB	Data complement

- Data Transfer Instructions

Mnemonic		Function
MOV	MOVB	Data transfer
MOVS	MOVSB	Short immediate transfer (#1 to #8 transfer)
MOVZ Rm	MOVZB Rm	Zero page area data transfer
MOVZ #n	MOVZB #n	Immediate data transfer to zero page area
MOVR		Data transfer between register banks
MOVC		Code data transfer
XCH		Data swap
SWAP		High byte and low byte swap
CLR	CLRB	Clear

- Rotate and Shift

Mnemonic		Function
ROL	ROLB	Register rotate left
ROR	RORB	Register rotate right
RLC	RLCB	Register and carry flag rotate left
RRC	RRCB	Register and carry flag rotate right
SLL	SLLB	Register shift left
SRL	SRLB	Register shift right
SRA	SRAB	Register shift right arithmetic

- Bit Operations

Mnemonic	Function
BSET	Bit set
BCLR	Bit clear
BNOT	Bit complement
BTST	Bit test
BAND	Carry and bit logical AND
BOR	Carry and bit logical OR
BXOR	Carry and bit exclusive OR
BMOV	Carry and bit transfer
SC	Carry flag set
RC	Carry flag reset
NC	Carry flag complement

- Jumps

Mnemonic	Function
J	Jump
SJ	Short jump
J [Rm]	Indirect jump
JCC	Conditional jump
JB JNB	Zero page area bit condition jump
DJNZ	Decrement register, and jump if not 0

- Subroutines

Mnemonic	Function
PUSH Rm	Register push
PUSH PSW	PSW push
POP Rm	Register pop
POP PSW	PSW pop
CAL	Subroutine call
SCAL	Subroutine short call
CAL [Rm]	Indirect call according to register Rm contents
RET	Return from subroutine
RETI	Return from interrupt
RETNMI	Return from NMI

- CPU Control

Mnemonic	Function
EI	Interrupt enable
DI	Interrupt disable
HALT	Halt (instruction execution halt)
STOP	Stop (clock oscillation stop)
BRK	System reset by software

Type	Pin Name	I/O	Description
Interrupt	INT0	Input	External interrupt 0 input pin. P5.0 secondary function.
	INT1	Input	External interrupt 1 input pin. P5.1 secondary function.
Timer	T0	Input	Timer 0 external clock input pin. P5.2 secondary function.
	T1	Input	Timer 1 external clock input pin. P5.3 secondary function.
	TC	Input	Capture timer external clock input pin. P6.4 secondary function.
Serial Port	TXU	Output	Serial port data transmit pin. P5.4 secondary function.
	RXU	Input	Serial port data receive pin. P5.5 secondary function.
	RXCK	Output	Serial port receive synchronizing clock output pin. P5.6 secondary function.
	TXCK	Output	Serial port transmit synchronizing clock output pin. P5.7 secondary function.
Capture Input / Compare Output	CAP0/ CMP0	I/O	Capture input/compare output channel 0 input/output pin. Capture trigger input pin in capture input mode, output pin when compare match is generated in compare output mode. P6.0 secondary function.
	CAP1/ CMP1	I/O	Capture input/compare output channel 1 input/output pin. Capture trigger input pin in capture input mode, output pin when compare match is generated in compare output mode. P6.1 secondary function.
	CAP2/ CMP2	I/O	Capture input/compare output channel 2 input/output pin. Capture trigger input pin in capture input mode, output pin when compare match is generated in compare output mode. P6.2 secondary function.
	CAP3/CMP3	I/O	Capture input/compare output channel 3 input/output pin. Capture trigger input pin in capture input mode, output pin when compare match is generated in compare output mode. P6.3 secondary function.

MICROCONTROLLER PROGRAM DEVELOPMENT SUPPORT SYSTEM EASE Series

GENERAL

The EASE program development support system has been developed for an exclusive purpose of conducting rapid and efficient program development for Oki's CMOS 4-bit, 8-bit and 16-bit one-chip microcontrollers.

FEATURES

- In-circuit emulation is possible by connecting to a host CP/M. computer having RS 232C interface or such an input/output unit as CRT terminal.
- A user program area of sufficient capacity is available, making possible code transfer between this area and the floppy disk, referencing/modification of the area data and continu-ous execution/stopped execution thereof.
- Multiple break (interruption of emulation) factors of execution address or machine cycle counter can be selected.
- The EPROM writer is available, permitting write/transfer/comparison of the data in the user program area.
- Transmission format
- The debugging operation of the user program can be designated by a debug command from the keyboard of an input/output unit connected.
- The diagnosis program for the system is built in.

CONDITIONS OF HOST COMPUTER

In the case where a computer of 80 system other than the if800, the system can use CP/M-80 (ver 2.0 or more), requiring a memory capacity of 52K bytes or more.

Also , the assignment of CON: = TTY: is possible, so that direct input to or output from the RS232C port must be possible. (Software for PC. DOS is also available)

DATA TRANSMISSION SYSTEM FOR EMULATION KIT IC USED:

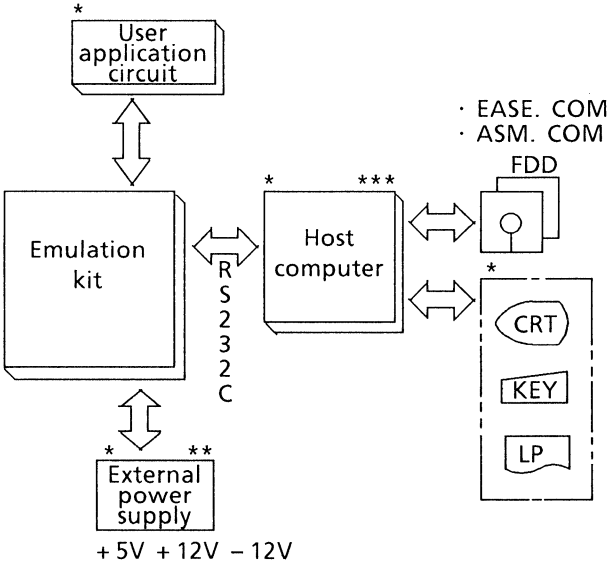
- Communication interface:
  MSM82C51A
- Driver/receiver:
  SN75188N, SN75189N

EASE 6502		· 110 to 9600 bps · 8 bits, 2 stop bits,even-parity 7bits, 2 stop bits, even parity 7bits, 2 stop bits, odd parity (switch operation)
EASE 80C49		· 110 to 2400 bps · 8 bits, 2 stop bits, non-parity
EASE 6458 EASE 62780	EASE 80C51mk II EASE 66301	· 300 to 19200 bps · 8 bits, 2 stop bits, non-parity
EASE 6352 EASE 5054/55 EASE 6351/53	EASE 5052/56 EASE 6400 EASE 6051	· 110 to 9600 bps · 8 bits, 2 stop bits, non-parity

SYSTEM CONFIGURATION

This system is configured of a high-performance evaluation emulator (including an EASE host monitor, emulation kit and host CP/M computer) and a strong cross assembler/ASM cross assembler operated under CP/M.

By connecting the emulation kit and the host CP/M computer (such as the if800) having the RS232C interface, all operations ranging from assembly of a source program to evaluation/ debug of the program can be accomplished.



Notes;

- CP/M is a registered trade mark of U.S. Digital Research, Inc.
- The part indicated by * should be supplied by the user.
- The part indicated by ** should be supplied by the user only for EASE6502 and EASE80C49.
- The part indicated by *** is an emulation kit that can be used with the MS.DOS computer as option.

EASE 5052/56, EASE 5054/55, **EASE 6352**
EASE 6351/53
EASE 6400
EASE 6502
EASE 80C49
EASE 80C51mk II
EASE 62780
EASE 66301

OKI semiconductor
EASE6400

PROGRAM DEVELOPMENT SUPPORT SYSTEM

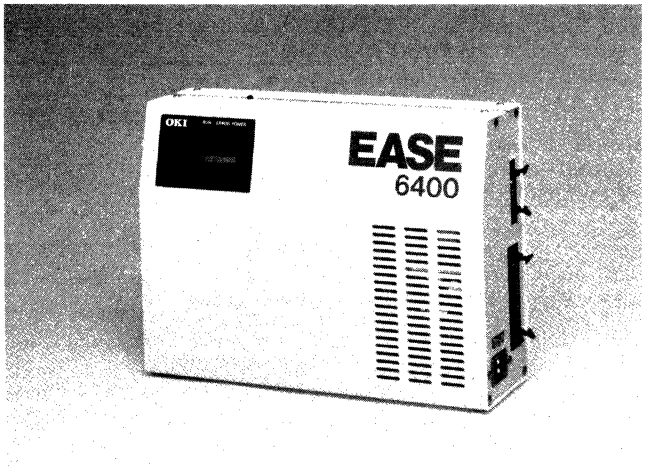
APPLICABLE MODELS

The EASE6400 Program Development Support System is capable of program development and evaluation of OKI's CMOS 4-bit one-chip microcontroller Series OLMS64. The applicable models include MSM6408, MSM6404, MSM6402, 6422, MSM6411 and MSM6442, MSM6434.

SUPPLY FORMAT

System	Class	Model	Name
EASE6400	Hardware	EASE6400 Emulation kit	Emulation kit for evaluation of OLMS-64 Series 4-bit one-chip microcomputer
	Software	EASE	Floppy disk base host monitor
		ASM6400	Floppy disk base cross assembler
	Manual	TM-6400	Program development support system users' manual
	Accessories	TCU-6400	Connection cable for user application system
		TCS-2	Connection cable for host CP/M computer (for if800 Model 20/30)
		TCP-2	AC cable for power supply
		TCX-3	External probe for emulation

*CP/M file 5 1/4 inch double-side double-density, 8-inch single-side single density MS-DOS file (optional software), PC-DOS file (optional software)



GENERAL DESCRIPTION OF CROSS ASSEMBLER

The ASM6400 Cross Assembler is a high-performance unit with a floppy disk base operated under CP/M or MS-DOS. This assembler produces an object file (Intel HEX format), assemble list file and a cross reference file into a designated system.

FEATURES

- Free description style of the source program.
- Ten types of operators can be described in the operand column of the source program.
- The number of characters for each line and the number of lines per page of the assemble list file can be designated.
- Fourteen different types of strong pseudo instructions are available.
- The branch pseudo instruction permits automatic production of an object code assigned with in-page jump instruction or full-page jump instruction.
- The output of the assemble list file can be controlled by the LIST, NLST pseudo instructions.
- List of pseudo instructions.

LIST OF PSEUDO INSTRUCTIONS

Pseudo Instruction	Function
EQU	Assigns the operand value to the name
SET	Similar to EQU pseudo instruction, but redefinable
ORG	Sets starting address of program
END	Means end of program
B	Checks destination of branch and automatically converts into in-page or full-page jump instruction
DB	Definition of 8-bit data or ASCII character
DS	Secures a memory area for as many bytes as designated
NSE	The four least significant bits of the location counter of the assembler are set to zero and 16 is added. In the process, the empty area not assigned with any machine instruction is assigned with NOP instruction.
DATE	Fills the date in the title section of assemble list
PAGE	Changes the page of assemble list
TITL	Gives title to assemble list
LIST	Designates the output of assemble list
NLST	Inhibits output of assemble list
TYPE	Gives microcomputer model number of OLMS64 Series

GENERAL DESCRIPTION OF EMULATOR

The EASE6400 Emulator efficiently supports all debug jobs for development from the debug lacking user application circuits to the debug of a complete system of the OLMS64 Series application program configured of a host computer and the EASE6400 Emulation Kit.

Real-time Trace Function Not Affecting Execution Time

The EASE6400 Emulation Kit has a real-time trace area for 2048 machine cycles, and by setting an address of the user program where trace is to be effected, the condition of an instruction code, board and probe can be traced each time of execution of a designated address.

Execution Time Measuring Function

By use of a cycle counter on the EASE6400 Emulation Kit, the time of execution process of the user program can be measured. (Max. 16777215 cycles)

Also, the counter can be used as a pass counter, making it possible to determine how many times a specific address has been executed. Further, the emulation may be suspended after a specific address has been executed a designated number of times.

Wide User Program Area

The EASE6400 Emulation Kit is provided with 8192 bytes of static RAM area as a user program area, freeing the user from the worry about the shortage of memory capacity at the time of debug. The user program can of course be loaded or saved in this RAM area from the host CP/M computer. Further, equipped with an EPROM writer, this kit is capable of writing the content of RAM area into EPROM or reading the content of EPROM.

Multiple Break Functions

This emulator has the following six methods of interrupting (breaking) the program execution:

a) BREAKPOINT BREAK
This break occurs when a designated address is executed (possible for all addresses).

b) EXTERNAL BREAK
A break due to an external forcible break signal.

c) HALT/STOP INSTRUCTION BREAK
A break occurs upon execution of HALT or STOP instruction.

d) TRACE BUFFER FULL BREAK
Breaks when the trace area overflows.

e) CYCLE COUNTER OVERFLOW BREAK
Breaks when the cycle counter overflows.

f) PROBE MATCH BREAK
Breaks when the probe data coincides with the set data.

Wealth of Debug Commands

Not only the display or modication of the data in each register, port or RAM is effected but also an efficient debug job is performed by the following commands:

a) Execution of user program

Input format:
Number of STP instructions and starting address

By entering a command as mentioned above, the user program

can be executed by the designated number of instructions from a designated address (starting address).

Input format:
G starting address and break address (n)

When the command mentioned above is entered, the user program is executed from a designated address (starting address), and upon execution of a designated address (break address) n times, the emulation is stopped.
Note: The program execution is suspended each time the designated address is executed.

Input format:
G starting address and break address RAM (Address-n)

When this message is entered, the execution of the user program is started from a designated address (starting address), and upon execution of a designated address (break address), the emulation is stopped if the data in the address of the RAM designated is n. Also in this case, the program execution is suspended each time the designated address is executed.

b) Utilization of floppy disk

Input format:
LOD file name

In accordance with this command, the data of the designated file (user program) can be loaded in the code memory on the EASE6400 emulation kit corresponding to the mask ROM area of OLMS64 Series.

Input format:
SAV file name starting address and end address

This command enable the data in the designated area of the code memory to be saved by a designated file name.

c) Instruction executed bit memory

This emulator has an instruction executed bit memory representing the address where the user program has been executed. By checking the data in this memory, therefore, it is possible to know the flow of the program.

Command Format Easy To Remember

The debug command of this emulator is configured of a command mnemonic and the following parameters (address or mnemonic).

Command mnemonic structure:

a) First character
Means the function performed by the emulator

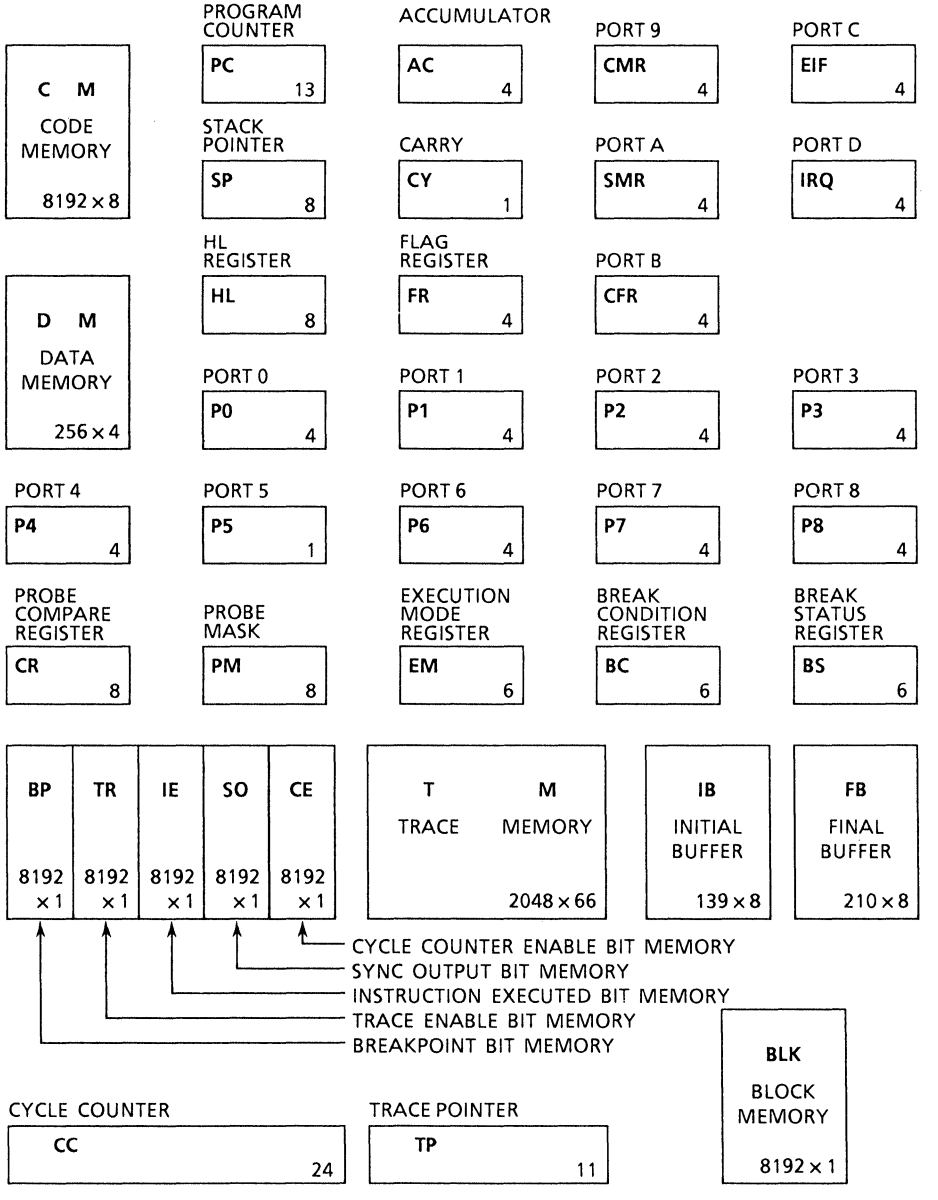
b) Second and subsequent characters
The name of the object of the function performed by the emulator. Evaluation chip of MSM6400 or the register, memory or port name of EASE6400 emulation kit (abbreviated name).

[Example]

a - Command input wait from the emulator is displayed.
b - The display of the object data designated is indicated.
c - Data memory (corresponding to RAM area of MSM6404) is designated.
d - Start address of the data displayed
e - End address of the data displayed
f - Carriage return meaning the end of command entry

MEMORY STRUCTURE OF EMULATOR

The memory area of the EASE6400 Emulator usable by the user is shown below. The portions surrounded by frame in the drawing indicate the register or memory areas accessible by way of keyboard of the host computer, where display or modification is possible by debug command.



OPERATION OF MAIN MEMORIES

Code Memory

The code memory is a RAM area of 8192 x 8 bits and corresponds to the ROM area of OLMS64 Series for storing the user program. (ROM capacity of MSM6404 is 4000 bytes)

Data Memory

The data memory, making up a RAM area of 256 x 4 bits, corresponds to the internal RAM area of OLMS64 Series.

Instruction Executed Bit Memory

This memory, a RAM area of 8192 x 1 bits having the same address as the code memory, and has the bit corresponding to the address thereof set to 1 upon execution of the program in the code memory. By checking the content of this memory, the flow of the program can be determined.

Break Point Bit Memory

The break point bit memory is an RAM area of 8192 x 1 bits having the same address as the code memory. When the bit of this memory is 1, the program is suspended immediately after execution of the content of the code memory corresponding to the address thereof.

Trace Enable Bit Memory

This memory is a RAM area of 8192 x 1 bits having the same address as the code memory. When the content of this memory is set to 1, the contents of the port and register are stored in the trace memory when the content of the code memory corresponding to the address thereof is executed.

Sink Output Bit Memory

This memory is a RAM area of 8192 x 1 bits, and has the same address as the code memory. When the content of the code memory is executed, the corresponding bit to this memory is checked, and if the bit is 1, a sync signal is applied to the probe terminal.

Cycle Counter Enable Bit Memory

This memory is a RAM area of 8192 x 1 bits having the same address as the code memory. When the content of code memory is executed, the content of the corresponding bit of this is checked, and if the bit is 1, the cycle counter is counted up in accordance with the machine cycle thereof.

Probe Compare Register/Probe Mask

By changing these two settings, the conditions for causing a break (probe match break) by the input data from the probe can be changed.

Initial/Final Buffer

These two buffers, which are a RAM area of 210 x 8bits, are for initial setting of MSM6400E for evaluation chip in real-time emulation or for storing the conditions of MSM6400 immediately following a break.

Trace Memory

The trace memory is a RAM area of 2048 x 66 bits for storing the trace data. The trace designation is effected by the trace enable bit.

LIST OF DEBUGGING COMMANDS

Load, Save, and Verify Command		
	LOD (dr :)filename↵ SAV (dr :)filename (address, address)↵ VER (dr :)filename (address, address)↵	Load programs into code memory Save code memory program Verify file with code memory
EPROM Command		
	TYPE mnenonic ↵ PPR address, address (, address) ↵ TPR address, address (, address) ↵ VPR address, address (, address) ↵	EPROM Type set Program Code Memory into EPROM Transfer EPROM into Code Memory Verify EPROM with Code Memory
Display of the Internal Condition of Everchip, and Change Command		
PC	DPC CPC address↵	Display Program Counter Change Program Counter
SP	DSP↵ CSP data↵	Display Stack Pointer Change Stack Pointer
AC	DAC↵ CAC data↵	Display Acc Change Acc
CY	DCY↵ CCY data↵	Display Carry Flag Change Carry Flag
HL	DHL↵ CHL data↵	Display H-L registers Change H-L registers
FR	DFR↵ CFR data↵	Display Flag Register Change Flag Register
P0–P8	DPn↵ (n=0, 1, 2, 3, ···8) CPn data↵ (n=0, 1, 3, ···8)	Display Port n Change Port n
P9	DCMR↵ CCMR data↵	Display Counter Mode Register (port 9) Change Counter Mode Register
PA	DSMR↵ CSMR data↵	Display Shift Mode Register (port A) Change Shift Mode Register
PB	DCFR *Change CFR command is not permitted	Display Control Flag Register (port B)
PC	DEIF↵ CEIF data↵	Display Enable Interrupt Flag (port C) Change Enable Interrupt Flag
PD	DIRQ↵ *Change IRQ command is not permitted	Display Interrupt Request Flag (port D)
MEI	DMEI ↵ CMEI data↵	Display Master Enable Interrupt Flag Change Master Enable Interrupt Flag
	D↵	Display all internal status
Display of Data Memory and Code Memory, Change and Field Command		
DM	DDM address (, address)↵ CDM address↵ data data···↵ FDM address, address, data↵	Display Data Memory Change Data Memory Fill Data Memory
CM	DCM address (, address)↵ CCM address↵ data data···↵ FCM address, address, data↵	Display Code Memory Change Code Memory Fill Code Memory

Display of Attribute Memory, Enable and Reset Command		
BP	DBP address [, address]↵ EBP address [, address]↵ RBP addres [, address]↵	Display Break Point Bits Memory Enable Break Point Bits Memory Reset Break Point Bits Memory
TR	DTR address [, address]↵ ETR address [, address]↵ RTR address [, address]↵	Display Trace Enable Bits Memory Enable Trace Enable Bits Memory Reset Trace Enable Bits Memory
IE	DIE address [, address]↵ RIE address [, address]↵	Display Instruction Executed Bits Memory Reset Instruction Executed Bits Memory
CE	DCE address [, address]↵ ECE address [, address]↵ RCE address [, address]↵	Display Cycle Counter Enable Bits Memory Enable Cycle Counter Enable Bits Memory Reset Cycle Counter Enable Bits Memory
SO	DSO address [, address]↵ ESO address [, address]↵ RSO address [, address]↵	Display Sync Out Enable Bits Memory Enable Sync Out Enable Bits Memory Reset Sync Out Enable Bits Memory
Display Command of Trace Memory		
TM	DTM number, number DTL -number, number	Display Trace Memory Display Trace List
Other Display of Hardware and Change Commnd		
CC	DCC↵ CCC number↵	Display Cycle Counter Change Cycle Counter
BC	DBC↵ SBC ([±]mnemonic, ···)↵	Display Break Condition Register Set Break Condition Register mnemonic means one of following keywords here. BB···Break at Break Point XB···External Break CO···Cycle Counter Over Flow TF···Trace Memory Full PM···Probe Match HS···HALT/STOP Instruction Executed
BS	DBS↵	Display Break Status Register
PM	DPM↵ CPM data↵	Display Probe Mask Register Change Probe Mask Register
CR	DCR↵ CCR data↵	Display Probe Compare Register Change Probe Compare Register
EM	DEM CEM mode-ref↵	Display Execution Mode Register Change Execution Mode Register mode-ref means one of following keywords here IF ···IB TO FB IN ···IB TO NON NN···NON TO NON NF···NON TO FB FF···FB TO FB

BLK	DBLK address [, address]↵ SPB address [, address]↵ SDB address [, address]↵	Display Block Memory Set Block Memory into Program Block Set Block Memory into Data Block
BANK	Bank 1 or 2	Set Attribute Memory Bank register
Initial Buffer, Final Buffer		
	DIB↵ DFB↵ CIkey-word data↵	Display Initial Buffer Display Final Buffer Change elements of Initial Buffer key-word···AC, HL, FR, SMR, CMR, MEI, EIF, CY, Pn (n=0, 1, 3, 4, ···, 8)
IB	CIDM address↵data data···↵	Change Initial Buffer Data Memory
FB	CF key-word data↵	Change elements of Final Buffer key-word···AC, HL, FR, SMR, CMR, MEI, EIF, CY, Pn (n=0, 1, 3, 4, ···, 8)
	CFDM address↵data data···↵ FIDM address, address, data↵ FFDM address, address, data↵	Change Final Buffer Data Memory Fill Initial Buffer Data Memory Fill Final Buffer Data Memory
Assemble or Disassemble Command		
	ASM address↵mnemonic·····↵ DASM address [, address]↵	Assemble to Code Memory Disassemble to Console
Emulation Command		
	G [start-address] [, break-parameter, ··· ·····]↵	Go (Start Real Time Emulation)
	GD [start-address] [, break-parameter, ·····]↵	Go Direct (Start Real Time Emulation) break-parameter format is shown below 1) address 2) address (n) 3) address (n)& address (m) 4) address RAM (address−data) 1≦n, m≦65535
	STP [number] [, address]↵	Start Step Execution
Other Commands		
	@ (−n)↵ (n=1, 2, ·····, 9) HELP↵ EXIT↵ RES↵ RES E↵ SIO {F / H}↵ DIV↵ ST mnemonic FDD [dr·····dr]↵ DIR	Command-Repeat Display Help File (EASE64.HLP) Exit to CP/M Reset EASE6400 system Reset Eva-chip (MSM6400E) Set communication mode CH2 port Oatput Control Family Set Flopy Disk Unit Set Display file directory

MODEL-WISE MODULE (option)

The module-wise module is provided for efficient program development and evaluation of respective microcontroller models of OLMS64 Series.

It is divided into PAM64 Series conveni-ently used for program evaluation in connection with the EASE6400 Emulation Kit and PEM64 Series effective for final evaluation of the program using the program EPROM.

● PAM6411 and PEM6411

PAM6411 and PEM6411 are used for MSM6411.

- PAM6411 has a cable connected to the EASE6400 Emulation Kit and a cable having a 16-pin DIP connector for connection the user application circuit. The name of each signal of the 16-pin DIP connector is arranged the same way as the terminals of the MSM6411, permitting direction insertion into the sockets of the printed board of the user application circuit.
- PEM6411 has a cable with a 16-pin DIP connector connected to the user application circuit and the program EPROM socket and an X'tal board packageable with an external crystal oscillator. The program EPROM usable by being set on the EPROM socket is of four types including 2732, 2732A, 2764 and 27128, and is provided in a form suitable for the final evaluation of the program converted to EPROM. The DC power (+5V) is supplied from the user application circuit through the user cable. The current consumption is the total of current consumed by EPROM, the microcontroller MSM6400E for program evaluation (CMOS' IC) and the oscillation MSM4069 and MSM4052.

● PAM6422 and PEM6422

PAM6422 and PEM6422 are used for MSM6422.

- PAM6422 has a cable connected to the EASE6400 Emulation Kit and a cable with a 24-pin DIP connector connected to the user application circuit. The name of each signal of the 24-pin DIP connector is arranged in the same manner as the MSM6422 terminals, and therefore direct insertion is possible into the sockets of the printed board of the user application circuit.
- PEM6422 has a cable with a 24-pin DIP connector connected to the program EPROM socket and the user application circuit and an X'tal board packageable on an external crystal oscillator. The program EPROM usable by being set on the EPROM socket is of four types including 2732, 2732A, 2764 and 27128, which are presented in a form suitable for final evaluation of the EPROM-converted program. The DC power (+5V) is supplied from the user application circuit through the user cable. The current consumption is a total of current consumed by EPROM, the microcontroller MSM6400E (CMOS' IC) for program evaluation and an oscillation MSM4069 and MSM4052.

● PAM6434 and PEM6434

PAM6434 and PEM6434 are used for program development of MSM6434.

- PAM6434 has a cable connected to the EASE6400 Emulation Kit and a cable with a 30-pin shrink DIP connector connected to the user application circuit. The terminals of the 30-pin shrink DIP connector arranged in the same order as those of MSM6434 making possible direct connection to the user application circuit.
- PEM6434 has a 30-pin shrink DIP connector for connection to the EPROM socket and the user application circuit. The EPROM usable by being set in the EPROM socket is of four types including 2732, 2732A, and 2764, which are given in a form suitable for final evaluation of the EPROM-converted program. The DC power (+5V) is supplied from the user application circuit through the 30-pin shrink DIP connector. The current consumption involved is a total of current consumed by EPROM, the microcontroller MSM6400E for program evaluation, A/D conversion IC (CMOS' IC) and one TTL.

● PAM6442 and PEM6442

PAM6442 and PEM6442 are used for MSM6442.

- PAM6442 has a cable connected to the EASE6400 Emulation Kit and a cable with a connector to the user application circuit. PAM6442 case has packaged therein an LCD driver (CMOS' IC) having the same function as the driver section of MSM6442LCD, making possible efficient program development in connection with an actual equipment (user application unit).
- PEM6442 has a cable with a connector to the program EPROM socket and the user application circuit and an X'tal board packageable with an external crystal oscillator. The program EPROM usable by being set in the EPROM socket is of four types including 2732, 2732A, 2764 and 27128, and are given in a form suitable for final evaluation of the EPROM-converted program. The DC power (+5V) is supplied from the user application circuit through the user cable. The current consumption involved is a sum of those currents consumed by the EPROM, the microcomputer (CMOS' IC) for program evaluation, oscillation MSM4069, LCD drive (CMOS' IC) and several standard CMOS' ICs.

OKI semiconductor
EASE 6458
PROGRAM DEVELOPMENT SUPPORT SYSTEM

APPLICABLE MODELS

The EASE6458 program development support system can perform program development and evaluation of Oki's CMOS 4-bit one-chip microcontroller MSC6458. It is a high-performance system exclusely developed incorporating Oki's own concept and technologies with the intention of efficient and prompt program development for MSC6458.

System	Class	Model	Name
EASE6458	Hardware	EASE6458 Emulation kit	Emulation kit for MSC6458
	Software	EASE	Floppy disk base host monitor*
		ASM6458	Floppy disk base cross assembler*
	Manual	TM-6458	EASE6458 program development support system user's manual
		AM-6458	ASM6458 cross assembler manual
	Accessories	TCU-6458	User application system connection cable
		TCS-2	Host CP/M computer connection cable (for if800 Model 20/30)
		TCP-2	AC power cable (100V/220V)
		TCX-6458	Probe cable

*CP/M file $5\frac{1}{4}$ inch double-side double-density, 8-inch single-side single density.
MS-DOS file (optional software), PC-DOS file (optional software).

GENERAL DESCRIPTION OF CROSS ASSEMBLER

The ASM6458 is a high-performance cross assembler with floppy disk base operated under CP/M or MS-DOS. This assembler generates the object file (Intel HEX format), assemble list file and the cross reference list file into a designated system.

FEATURES

- A source program of free description type
- Ten types of operation can be described in the operand column of the source program.
- The number of characters per line and the number of lines per page of the assemble list file can be designated.
- The cross reference list file can be generated.
- Thirteen types of strong pseudo instructions are available.
- A branch pseudo instruction makes it possible to obtain an object code assigned with an in-page jump instruction or a full-page jump instruction automatically.
- By the LIST or NLST pseudo instruction, the output of the assemble list file can be controlled.
- List of pseudo instructions.

LIST OF PSEUDO INSTRUCTIONS

Pseudo Instruction	Function
EQU	Assigns the operand value to the name
SET	Similar to the EQU pseudo instruction, but redefinable
ORG	Sets the starting address of the program
END	Means the end of a program
B	Checks the destination of a branch, converting to the in-page or full-page jump instruction automatically.
DB	Definition of the 8-bit data or ASII character
DS	Secures a memory area for as many bytes as designated.
NSE	The value of the four least significant bits of the location counter of the assembler is reduced to zero, and 16 is added. The empty area not assigned with a machine language is assigned with an NOP instruction.
DATE	Fills a date in the title section of the assemble list.
PAGE	Changes the page of the assemble list.
TITL	Gives a title to the assemble list.
LIST	Designates the output of the assemble list.
NLST	Inhibits the output of the assemble list

GENERAL DESCRIPTION OF EMULATOR

The EASE6458 Emulator, by being connected with a host computer through RS232C Interface, efficiently supports all types of debug jobs in development work from debug free of user application circuits for MSC6458 application program to the debug for a completed system.

Real-Time Emulation

MSC6458E, an exclusive chip for evalution, is used, permitting real-time emulation without insertion of wait state.

Execution Time Measurment Function

By using the cycle counter on this emulation kit, the execution time of the user program can be measured. (32-bit binary counter)

The start/stop of the cycle counter is effected by setting a start address/stop address of the cycle counter by commands.

Wide User Program Area

This emulator is equipped with a 16K-byte RAM area as a user program area (code memory), and therefore all the user programs on the host CP/M computers can be loaded in the RAM area.

Further, with the EPROM writer provided, the data of the user program area can be written in EPROM or data can be read from the EPROM.
(Applicable models: i2732, 2732A, 2764, 27128, 27256)

Multiple Break Functions

This emulator has the following methods of breaking (suspending) the program execution, each capable of being set or cancelled:

a) BREAK POINT BREAK
A break that occurs when the program of the same address as that where a break point is set is executed. (All address can be designated)

b) ADDRESS BREAK
A break that occures when the address designated at the time of entry of an emulation command is executed. Also, a break may be designated to occur upon n executions of a designated address.

c) BREAK DUE TO POWER-DOWN
A break occurs in power-down mode.

d) BREAK BY EXTERNAL FORCIBLE BREAK SIGNAL
A break can be caused by input of low-level signal from an external source by use of an accessorial probe cable.

e) OVERFLOW OF TRACE MEMORY
A break that occurs when the trace memory is filled up with trace information.

f) OVERFLOW OF CYCLE COUNTER

g) BREAK DUE TO CONTENT OF INTERNAL RAM AREA
A break occurs when the data of an internal RAM of a designated MSC6458E is checked at the time of execution of a program of a designated address and coincides with a designated content a designated number of times.

Improved Function of Real-Time Tracing

This emulator has a real-time trace area that does not affect the following two types of execution time:

- **Trace memory**

  A memory for tracing (storing) the conditions of the port, carry flag and accumulator of MSC6458E when an instruction in the code memory is executed. (for a maximum of 2048 machine cycles) The following three methods are available for trace designation:

a) A method in which trace is effected each time of execution of a designated address.

b) A method in which trace is effected when a special instruction is executed. (CAL, LCAL, RTS, PUSH, etc.)

c) A method in which a trace is conducted by a trace start/stop bit.

- **Flash trace memory**

  A memory for tracing the conditions of all the areas of internal RAM of MSC6458 in executing the instruction of a designated address by an emulation command (debug command).

The data of the internal RAM area can be stored a maximum of 32 times.

Easy-to-Use Emulator Command

The emulator command of this emulator (debug command) consists of a command mnemonic and a following parameter (address or mnemonic).

Construction of command mnemonic

a) **First character**
Means a function performed by the emulator.

b) Second and subsequent characters
These make up the name of an object of the function performed by the emulator. An abbreviated name of memory, port and address of the EASE6458 Emulation Kit are inserted.

[Example]

a - Indication of command input wait from emulator
b - Designation of display of the data of the object designated
c - Designation of data memory (corresponding to internal RAM area of MSC6458E)
d - Starting address of the data displayed.
e - End address of the data displayed.
f - Carriage return meaning the end of command input.

This emulator has a variety of emulator commands not only for display/change of the data of each register and port of the MSC6458E exclusive chip for evaluation but also efficient debugging.

Input format:
STP NUMBER START ADDRESS

By input of a command as mentioned above, the user program can be executed as many as the instructions (commands) from a designated address (start address).

Input format:

G START ADDRESS, PARAMETER

PARAMETER INPUT FORMAT

1. **BREAK ADDRESS,**····· **BREAK ADDRESS**
2. **BREAK ADDRESS (n)**
3. **BREAK ADDRESS** **RAMram ADDRESS (byte-n)**

When a command is entered like (1), the user program is executed from a designated address (start address), and a break occurs when any of the break addresses designated is executed.

Upon entry of a command of (2), the user program is executed from the start address, and a break occurs upon a number n of executions of a designated break address.

Upon entry of a command like (3), the data at a designated address (ram address) of the internal RAM of MSC6458 is checked at the time of execution of the user program of a break address executed from a designated start address and compared with a designated data (byte), with the result that they coincide with each other a number n of times.

Input format:

1. **LOD** **FILE NAME**
2. **SAV** **FILE NAME START ADDRESS END ADDRESS**
3. **VER** **FILE NAME START ADDRESS END ADDRESS**

By entering a command like (1), the data (user program) of a designated file can be loaded in a code memory on the EASE6458 Emulation Kit corresponding to the program memory (user program) area of MSC6458.

By the command of (2), the data in the designated range in the code memory can be saved by the designated file name.

The entry of (3), on the other hand, permits comparison of the data of a designated file with that of the code memory.

Input format:

1. **DIAG FILE NAME**

The command in the file designated by the command of (1) can be executed automatically. Also, if used with the PAUSE command, this command suspends the execution of the command in the file.

Input format:

1. **LIST FILE NAME**
2. **NLST**

By using these two commands, a file with the same content as the console display can be prepared. If it is used with the above-mentioned DIAG command, on the other hand, the operator is set free of the need to attend the host CP/M computer.

Input format:

S **MNEMONTC DATA NUMBER**

This command makes it possible to search the trace data (mnemonic) designated in the trace memory.

COMMAND MEMORY & DATA MEMORY COMMANDS

1.	DCM address [address]	Display Code Memory
2.	DDM address [address]	Display Data Memory
3.	CCM address	Change Code Memory
4.	CDM address	Change Data Memory
5.	FCM address address byte	Fill Code Memory with byte
6.	FDM address address byte	Fill Data Memory with byte

ATTRIBUTE MEMORY COMMANDS

1.	DBP address [address]	Display Break Point Bit Memory
2.	DTR address [address]	Display Trace Enable Bit Memory
3.	DSO address [address]	Display Sync Output Enable Bit Memory
4.	DIE address [address]	Display Instrucion Executed Bit Memory
5.	EBP address. . address	Enable Break Point Bit
6.	ETR address. . address	Enable Trace Enable Bit
7.	ESO address. . address	Enable Sync Output Enable Bit
8.	FBP address address byte	Fill Break Point Bit Memory with byte
9.	FTR address address byte	Fill Trace Enable Bit Memory with byte
10.	FSO address address byte	Fill Sync Output Enable Bit Memory with byte
11.	RBP address. . address	Reset Break Point Bit
12.	BTR address. . address	Reset Trace Enable Bit
13.	RSO address. . address	Reset Sync Output Enable Bit
14.	RIE	Reset Instruction Executed Bit Memory

BUFFER MEMORY COMMANDS

1.	DBUF	Dispiay Buffer Memory
2.	TBUF	Transfer EVA Internal Status into Buffer Memory
3.	LBUF	Load Buffer Memory into EVA

CYCLE COUNTER COMMANDS

1.	DCC	Display Cycle Counter
2.	CCC data	Change Cycle Counter
3.	TIME data	Set 1 cycle time

BREAK CONDITION & STATUS COMMANDS

1.	DBC	Display Break Status
2.	DBS	Display Break Status
3.	SBC (+/−) mnemonic. . (+/−) mnemonic	Set Break

PROBE COMMANDS

1.	DPM	Display Probe Mask
2.	CPM	Change "
3.	DCR	Display Probe Compare Register
4.	CCR	"

MISCELLANEOUS COMMANDS

1.	RES	EASE6458 System initalization
2.	RESE	MSM6458E eva-chip reset
3.	SIO F or H	Set Emulator to I/O terminal mode
4.	PAUSE	Stop command file execution and wait key-in
5.	EXIT	Reture host Computer's OS
6.	DIR [dr:]	Display fill directory
7.	MAC	Define command execution
8.	M	Execute front command
9.	@	Repeat front command
10.	CVFLT	Cange VFLT voltage
11.	SSOP	Set syncout palse
12.	EEXTR	Enable External Reset
13.	REXTR	Reset External Reset

DISK ACCESS COMMANDS

1.	LOD [dr:] filename	Load Program into Code Memory
2.	SAV [dr:] filename [address address]	Save Code Memory Program
3.	VER [dr:] filename [address address]	Verify File with Code Memory
4.	DIAG [dr:] filename	Execute command file
5.	LIST [dr:] filename	List console Output into Disk file
6.	NLST	End listing
7.	HELP	Display HELP File
8.	FDD dr..dr:	Set Disk Drive
9.	LSG [dr:] filename	Load Data into Segmemt Data Memory
10.	SSG [dr:] filename	Save Segement DataMemory Data
11.	VSG [dr:] filename	Verify File with Segment Data Memory

EPROM WRITER COMMANDS

1.	PPR address address [address]	Program Code Memory into EPROM
2.	TPR address address [address]	Transfer EPROM into Code Memory
3.	VPR address address [address]	Verify EPROM with Code Memory
4.	TYPE mnemonic	Set EPROM type

ASSEMBLE COMMANDS

1.	ASM address	Assemble to Code Memory
2.	DASM address	Disassemble to Console
3.	DBLK address [address]	Display Block Memory
4.	SPB address [address]	Set Program Block
5.	SDB address [address]	Set Data Block

TRACE COMMANDS

1.	DTM-number number	Display Trace Memory
2.	DFTM numbre number	Display Flash Trace Memory
3.	DTG mnemonic	Display Trigger Mode
4.	DFA	Display Flash Trace address
5.	S mnemonic data number	Search Trace Memory data
6.	STG mnemonic	Set Trigger Mode
7.	SFA address [..address]	Set Flash Trace address
8.	RFA address [..address]	Reset Flash Trace address
9.	RTG mnemonic	Reset Trigger Mode

EMULATION COMMANDS

1. STP number [st-address]
2. G [st-address] [break-parameter]
   if the optional st-address is give, emulator will begin emulation from st-address.
   And if optional break-parameter is given.
   emulation will break on first break-parameter to be satisfied.

   break-parameter=break-address..break-address (max.10)
   break-address (pass count)
   ram-address (byte-pass count)

EVA CHIP COMMANDS

1.	D		Display Regiater, Flag Port
2.	PC	DPC	Display Progrm Counter
		CPC data	Change "
3.	AC	DAC	Display Accumulator
		CAC data	Change "
4.	CY	DCY	Dispay Carry
		CCY data	Change "
5.	MEIF	DMEI	Display Master Enable Interrupt Flag
		CMEI data	Change "
6.	HL	DHL	Display HL Register
		CHL data	Change "
7.	P0	DP0	Display Port0
		CP0 data	Change " "
8.	P1	DP1	Display Port1
		CP1data	Change "
9.	P2	DP2	Display Port2
10.	P3	DP3	Display Port3
		CP3 data	Change "
11.	P4	DP4	Display Port4
		CP4 data	Change "
12.	P5	DP5	Display Port5
		CP5 data	Change "
13.	P6	DP6	Display Port6
		CP6 data	Change "
14.	P7	DP7	Display Port7
	CP7	data	Change "
15.	P8	DP8	Display Port8
16.	FR	DFR	Display F Register
		CFR data	Change "
17.	SP	DSP	Display Stack Pointer
		CSP data	Change "
18.	TMR	DTMR	Display Programable Timer
		CTMR data	Change "
19.	CT	DCT	Display Timer Event Counter
		CCT data	Change "
20.	SR	DSR	Display Shift Register
		CCMR data	Change "
21.	CMR	DCMR	Display Counter Mode Register
		CCMR data	Change "
22.	SMR	DSMR	Display Shift Mode Register
		CSMR data	Change "
23.	CFR	DCFR	Display Control Flag Register
		CCFR data	Change "
24.	IRQ	DIRQ	Display Interrupt Request Flag
		CIRQ data	Change "
25.	EID	DEID	Display Enable Interrupt Flag D
		CEID data	Change "
26.	EIE	DEIE	Display Enable Interrupt Flag E
		CEIE data	Change "
27.	PF	DPF	Display Port F
		CPF data	Change "

OKI semiconductor

EASE 6502

PROGRAM DEVELOPMENT SUPPORT SYSTEM

APPLICABLE MODELS

The EASE6502 Program Development Support System is capable of developing and evaluating the programs of Oki's CMOS 4-bit one-chip microcontroller OLMS65 Series. The applicable model is MSM6502.

SUPPLY FORMATS

System	Class	Model	Name
EASE6502	Hardware	MPB6502	Board for evaluating OLMS-65 Series 4-bit one-chip microcontroller
	Software	EASE65	Floppy disk base host monitor[1)]
		ASM6502	Floppy disk base cross assembler[1)]
	Manual	TM-6502	EASE6502 Development Support System Users' Manual
	Accessories	CTU-6502	User application circuit connection cable
		TCS-1	Host-CP/M computer connection cable (for if800 Model 20/30)
		TCP-1	Power supply cable (+ 5V, 4.5A) (+ 12V, 0.2A) (−12V, 0.2A)
		TCX-1	External probe cable for emulation
		TCC-1	Board connection cable

1) CP/M non-system disk
- 5¼-inch double-side double density if800 format
- 8-inch single-side single density IBM3740 soft sectored format

GENERAL DESCRIPTION OF CROSS ASSEMBLER

ASM6502 is a high-performance cross-assembler with floppy disk base adapted to be operated under CP/M or MS-DOS[2].

This assembler translates the source file on the disk prepared by use of a commercially-available editor, and generates a designated system from an object file (Intel HEX format), an assemble list file and a cross reference list file.

[2] MS-DOS version and PC-DOS version are optional softwares

FEATURES

- A source file of free description type
- Ten types of operator can be described in the operand column of the source program.
- The number of characters per line and the number of lines per page of the assemble list file can be designated.
- A cross reference list file can be generated.
- Thirteen types of strong pseudo instructions are available.
- An object code assigned with an in-page jump instruction or a full-page jump instruction is automatically obtained by a branch pseudo instruction.
- The output of the assemble list file can be controlled by the LIST of NLIST pseudo instruction.

LIST OF PSEUDO INSTRUCTIONS

Pseudo Instruction	Function
EQU	Assigns the value of an operand to the name
SET	Similar to EQU pseudo instruction but redefinable
ORG	Sets the start address of a program
END	Means the end of a program
B	Checks the destination of a branch converting automatically into an in-page/full-page jump instruction.
DB	Definition of 8-bit data or ASCII character
DS	Secures a memory area for the number of bytes as designated.
NSE	Sets to 0 the values of the four least significant bits of location counter of assembler, and adds 16. The empty area not assigned with a machine instruction is assigned with NOP instruction.
DATE	Fills the date in the title section of the assemble list.
PAGE	Changes the page of assemble list.
TITL	Gives a title to assemble list.
LIST	Designates output of assemble list.
NLST	Prohibits the output of the assemble list

GENERAL DESCRIPTION OF EMULATOR

The EASE6502 Emulator, consisting of a host computer and an MPB6502 program evaluation board, efficiently supports the debug work for development task ranging from the debug job free of the user application circuit of the MSM6502 application program all the way to the debug job on a completed system.

- **Real-Time Trace Function Not Affecting Execution Time**
  The MPB6502 evaluation board, with a real-time trace function for 2048 machine cycles, is capable of tracing the conditions of the port, HL register or MEI flag by setting an address of the user program area the execution of which initiates trace.

- **Execution Time Measurement Function**
  By using a cycle counter on the MPB6502 evaluation board, the user program execution time can be measured. (Max. 16777215 cycles)
  Also, this counter may be used as a pass counter for determining the number of executions conducted for a specific address. Further it is possible to suspend emulation upon execution of a specific address a designated number of times.

- **Wide User Program Area**
  The MPB6502 evaluation board is equipped with a static RAM area of 4096 bytes as a user program area, and therefore there is no worry of shortage of the memory capacity at the time of debugging. The user program can of course by loaded or saved in this RAM area from the host CP/M computer.
  Further, this board is equipped with the EPROM writer permitting the data in the user program area to be written in the EPROM or the data in the EPROM to be read.

- **Multiple Break Functions**
  This emulator has the following six methods of suspending the program execution:

a) BREAKPOINT BREAK
   This break occurs when a designated address is executed. (possible for all addresses)

b) EXTERNAL BREAK
   A break under an external break signal

c) HALT INSTRUCTION BREAK
   A break caused by execution of the HALT instruction

d) TRACE BUFFER FULL BREAK
   A break that occurs by overflow of the trace area

e) CYCLE COUNTER OVER FLOW BREAK
   A break caused by overflow of the cycle counter

f) PROBE MATCH BREAK
   A break caused by coincidence between probe data and set data.

- **Wealth of Debug Commands**
  Not only display or change of the data in each register, port or RAM but efficient debug work are pos-sible by the following commands:

a) Execution of user program

Input format:
STP NUMBER OF INSTRUCTIONS, ADDRESS

By entering this command, the user program can be executed as many instructions as designated from a designated address (start address).

Also, by designating the content of display by the SDF command at the same time, the display format can be made easier to watch.

Input format:
G START ADDRESS, BREAK ADDRESS (n)

If this command is entered, the user program is executed from a designated address (start address), and upon a number n of executions of a designated address (break address), emulation is stopped.
Note: Program execution is suspended each time of execution of a designated address.

Input format:
G START ADDRESS, BREAK ADDRESS RAM (Address-n)

Upon entry of this command, the user program begins to be executed from a designated address (start address), and upon execution of the designated address (break address), emulation is stopped if the designated RAM address (address) is n. The program execution is also suspended each time of execution of the designated address.

b) Utilization of floppy disk

Input format:
LOD FILE NAME

This command permits the content of a designated file (user program) to be loaded in a code memory on the MPB6502 evaluation board corresponding to the mask ROM area of MSM6502.

Input format:
SAV FILE NAME, START ADDRESS, END ADDRESS

This command makes it possible to save the content in a designated range of a code memory by the designated file name.

Input format:
DIAG FILE NAME

This command permits execution of a command in the designated file. At the same time, by use of the PAUSE command, the execution of a command in the file can be suspended.

Input format:
LIST FILE NAME, NLST

The use of these two commands makes it possible to prepare a file having the same content as the display on the console.

If this command is used together with the DIAG command, the operator is not required to attend the host CP/M computer all the time any longer.

c) Instruction executed bit memory

This emulator has an instruction executed bit memory indicating the address for which the user program has been executed. By checking the data in this memory, therefore, program flow can be determined.

- **Easy-to-Remember Command Format**

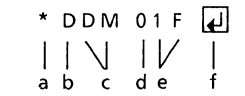
The debug command of this emulator consists of a command mnemonic and an immediately following parameter (address or mnemonic).

Construction of Command Mnemonic:

a) First character
Means the function performed by the emulator.

b) Second and susequent characters
The name of an object of the function performed by the emulator. Names of register, memory and port (abbreviated) of the MPB6502 evaluation board or MSM6502 evaluation chip.

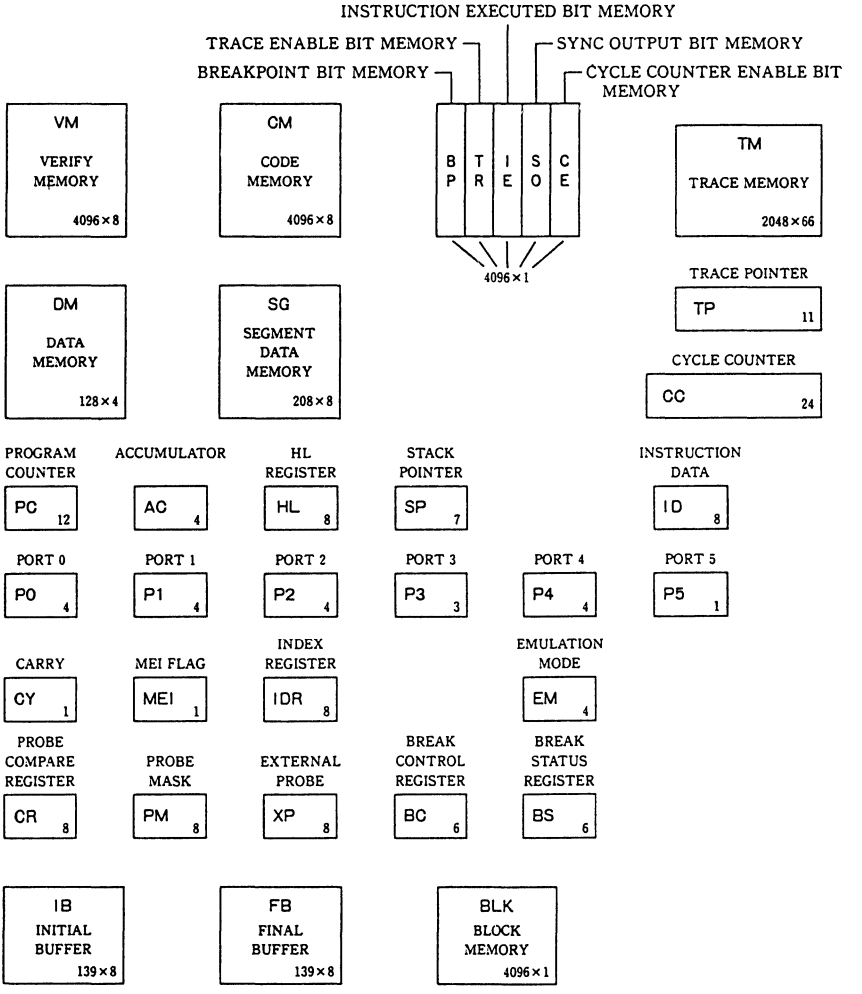
[Example]



a - Indication of command input wait from emulator
b - Designation of display of content of designated object
c - Designation of data memory (corresponding to RAM area of MSM6502)
d - Start address of data to be displayed.
e - End address of data to be displayed.
f - Carriage return meaning command input end.

EMULATOR MEMORY CONSTRUCTION

The memory areas of the EASE6502 that can be used by the user are shown below. The parts surrounded by rectangles indicate register and memory areas accessible by the computer keyboard. The display or change can be made by debug command.



MAIN MEMORY OPERATIONS

- **Code Memory**
  The code memory is a RAM area of 4096 × 8 bits, and corresponds to the mask ROM area of MSM6502 for storing the user program.
  (ROM capacity of MSM6502 is 2000 bytes)

- **Data Memory**
  The data memory is a RAM area of 128 × 4 bits and corresponds to the RAM read of MSM6502.

- **Verify Memory**
  The verify memory, which is a RAM area having the same address as the code memory when the user program is loaded in the code memory.
  By comparing this memory with the code memory, it is possible to know the part changed after the user program is loaded.

- **Instruction Executed Bit Memory**
  This memory is a RAM area of 4096 × 1 bits having the same address as the code memory. When the program in the code memory is executed, a bit corresponding to the particular address is set to 1.
  By checking the content of this memory, it is possible to determine the program flow.

- **Breakpoint Bit Memory**
  The breakpoint bit memory is a RAM area of 4096 × 1 bits with the same address as the code memory. When the bit of this memory is 1, the program execution is stopped immediately after execution of the data in the code memory corresponding to the particular address.

- **Trace Enable Bit Memory**
  This memory is a RAM area of 4096 × 1 bits having the same address as the code memory. When the data in this memory is set to 1, the data in the port and register is stored in the trace memory when the data in the code memory corresponding to the particular address is executed.

- **Sync Output Bit Memory**
  This memory, which a RAM area a RAM area of 4096 × 1 bits, has the same address as the code memory. In executing the data of the code memory, the bit corresponding to the memory is checked, and if the bit is 1, a sync signal (active LOW) is produced at the probe terminal.

- **Cycle Counter Enable Bit**
  This memory is a RAM area of 4096 × 1 bits with the same address as the code memory. When the data in the code memory is executed, the content of the bit corresponding to this memory is checked, and if the bit is 1, the cycle counter is counted up in accordance with the machine cycle.

- **Probe Compare Register/Probe Mask**
  By changing these two settings, it is possible to change the conditions for causing a break (probe match break) due to the input data from the probe.

- **Segment Data Memory**
  The segment data memory, which is a RAM area of 208 × 8 bits, has a coordinate with 0-7 in ordinate and A-Z in abscissa. This memory is for indicating the conditions of each bit in the crystal display of the RAM area of the MSM6502 at a designate coordinate by a designated character. By indicating the condition of this memory, the image under crystal display can be grasped.

- **Initial/Final Buffer**
  These two memories, which are RAM areas of 139 x 8 bits, are for initial setting of the evaluation chip MSM6502 for real-time emulation or storing the condition of MSM6502E immediately after a break.

- **Trace Memory**
  The trace memory, which is a RAM area of 2048 x 66 bits, is for storing trace information. A trace designation is conducted by a trace enable bit.

OKI semiconductor

Low Power 4-bit Microcotroller: OLMS-50/60 Series

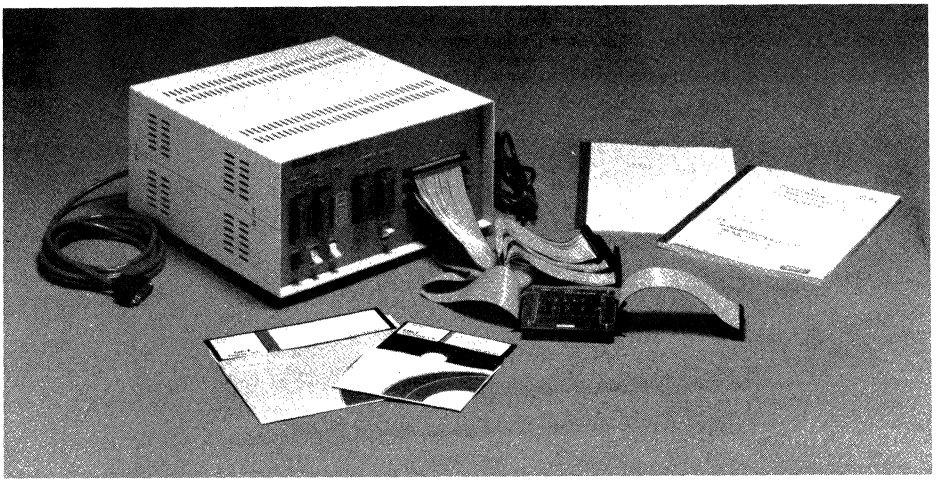
PROGRAM DEVELOPMENT SUPPORT SYSTEM

EASE6352/6052 **EASE6351/53**
EASE5052/56 **EASE6051**
EASE5054/55

APPLICABLE MODELS

Each of the program development support systems EASE6352/6052, EASE5052/56, EASE5054/55, EASE6351/53 and EASE6051 is capable of program development and evaluation of the Oki's CMOS 4-bit low-power one-chip microcontroller OLMS50/60 Series. It is a high-performance system developed exclusively for rapid and efficient program development for OLMS50/60 Series incorporating Oki's own concept and technologies.

System	Applicable microcontroller models
EASE6352/6052 Program Development Support System	MSM6052, MSM6352
EASE5052/56 Program Development Support System	MSM5052, MSM5056
EASE5054/55 Program Development Support System	MSM5054, MSM5055
EASE6351/53 Program Development Support System	MSM6351, MSM6353
EASE6051 Program Development Support System	MSM6051



SUPPLY FORM

System	Class	Model	Name
EASE6352/6052	Hardware	EASE6352/6052 Emulation Kit	Emulation Kit for MSM6352/6052
	Software	EASE	Host monitor for floppy disk base MS-DOS (Note 1)
		ASM-50	Assembler for floppy disk base MS-DOS (Note 1)
	Manual	TM-6352/6052	EASE 6352/6052 Users' Manual
		AM-50	ASM50 Assembler Users' Manual
	Accessories	TCU-6352/6052	Connection cable for user application circuit (Note 2)
		TCS-3S	Host computer connection cable (straight type 3 m)
		TCS-3C	Host computer connection cable (cross type 3 m)
		TCP-2	AC cable for power supply

System	Class	Model	Name
EASE5052/56	Hardware	EASE5052/56 Emulation Kit	Emulation Kit for MSM5052/56
	Software	EASE	Host monitor for floppy disk base CP/M (Note 1)
		ASM-50	Assembler for floppy disk base CP/M (Note 1)
		IF20	Transfer parameter change program for if800 Model 20
		IF30	Transfer parameter change program for if800 Model 30
	Manual	TM-5052/56	EASE5052/56 Users' Manual
		AM-5052/56	ASM50 Assembler Users' Manual
	Accessories	TCU-5052/56	User application circuit connection cable (Note 3)
		TCS-2	Host CP/M computer connection cable (for if800 Model 20/30)
		TCP-2	AC cable for power supply
		QA-11169	TEMP(S) board

System	Class	Model	Name
EASE5054/55	Hardware	EASE5054/55 Emulation Kit	Emulation Kit for MSM5054/55
	Software	EASE	Host monitor for floppy disk base CP/M (Note 1)
		ASM-50	Assembler for floppy disk base CP/M (Note1)
		IF20	Transfer parameter change program for if800 Model 20
		IF30	Transfer parameter change program for if800 Model 30
	Manual	TM-5054	EASE5054/55 Users' Manual
		AM-5054	ASM50 Assembler Users' Manual
	Accessories	TCU-5054	User application circuit connection cable (Note 4)
		TCS-2	Host CP/M computer connection cable (for if800 model 20/30)
		TCP-2	AC cable for power supply

System	Class	Model	Name
EASE6351/53	Hardware	EASE6351/53 Emulation Kit	Emulation Kit for MSM6351/53
	Software	EASE	Host monitor for floppy disk base MS-DOS (Note 1)
		ASM-50	Assembler for floppy disk base MS-DOS (Note 1)
	Manual	TM-6351/53	EASE Users' Manual
		AM-50	ASM50 Assembler Users' Manual
	Accessories	TCU-6351/53	User application circuit connection cable (Note 4)
		TCS-3S	Host computer connection cable (Straight type 3m)
		TCS-3C	Host computer connection cable (Cross type 3m)
		TCP-2	AC cable for power supply

System	Class	Model	Name
EASE6051	Hardware	EASE6051 Emulation Kit	Emulation Kit for MSM6051
	Software	EASE	Host monitor for floppy disk base MS-DOS (Note 1)
		ASM-50	Assembler for floppy disk base MS-DOS (Note 1)
	Manual	TM-6051	EASE6051 Users' Manual
		AM-50	ASM50 Assembler Users' Manual
	Accessories	TCU-6051	User application circuit connection cable (Note 4)
		TCS-3S	Host computer connection cable (Straight type 3m)
		TCS-3C	Host computer connection cable (Cross type 3m)
		TCP-2	AC cable for power supply

Note 1: 5-inch floppy disk for CP/M, if800 format, two-sided double-density
8-inch floppy disk for CP/M, IB3740 soft sectored format, one-side single density
5.25-inch index for MS/DOS, if800 Model 160 format (two-side double-density, 77 tracks, 26 sectors/track, 1024 bytes/sector, 2HD)
8-inch disk for MS/DOS, if800 Model 160 format (one-side single density, 77 tracks, 26 sectors/track, 128 bytes/sector, 1S)
Softwares for PC-DOS are supplied for each systems.
Note 2: One 40-core cable is supplied for connection of user application circuit.
Note 3: Three 60-core and two 40-core cables are supplied for user circuit.
Note 4: One 40-core and five 60-core cables are supplied for user application circuit.

GENERAL DESCRIPTION OF CROSS ASSEMBLER

ASM50 is a high-performance cross assembler of floppy disk base operated under CP/M, PC-DOS and MS-DOS.

This assembler generates an object file (Intel HEX format) and an assemble list file.

FEATURES

- Free style of description of a source program.
- A symbol list prepared in the assemble list.

GENERAL DESCRIPTION OF EMULATOR

The OLMS50/60 Emulator efficiently supports all debug works in the development task ranging from the debug free of a user application circuit for the OLMS50/60 Series program to the debug on a completed system by connecting through a host (CP/M, MS-DOS and PC-DOS) computer and RS232C interface.

REAL-TIME EMULATION

The user of an exclusive evaluation circuit permits real-time emulation without inserting a wait state.

This emulator has two operation modes. EMU mode is a normal program emulation mode making an efficient command available for program debug.

EVA mode, on the other hand, an evalu-ation mode for actuating a program in the form of EPROM and is used effectively for final check of a program almost completed.

USER PROGRAM AREA AND EPROM WRITER

The program to be debugged is stored in the code memory providing a user program area prepared in this emulator. The user program can be loaded or saved in the code memory from the host computer.

Also, this emulator has an EPROM built therein making possible loading/ programming in the code memory of the content of the program EPROM set in the EPROM writer socket.

The EPROMs usable include 2732, 2732A, 2764 and 27128.

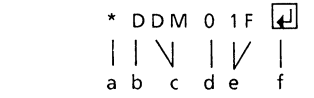
EASY-TO-USE EMULATOR COMMAND

The emulator command of this emulator (debug command) consists of a command mnemonic and following parameter (address or mnemonic).

Construction of Command Mnemonic:

a) First character
Means the function performed by the emulator.

[Example]



a - Indication of command input wait from emulator
b - Designation of display of the data of the object designated
c - Designation of data memory (corresponding to internal RAM area of OLMS50/60 Series)
d - Starting address of the data displayed.
e - End address of the data displayed.
f - Carriage return meaning the end of command input.

b) Second and subsequent characters
Provide the name of an object of the function performed by the emulator. It contains abbreviated names of the port, register and memory of the emulation kit.

Input format:
STP NUMBER OF INSTRUCTIONS, START ADDRESS

By entering the above-mentioned command, the user program can be executed as many times as the instructions designated from a designated address (start address).

Input format:
G START ADDRESS, PARAMETER
PARAMETER INPUT FORMAT
1. **BREAK ADDRESS ····, BREAK ADDRESS**

When this command is entered, the user program is executed from a designated address (Start address), and breaks upon execution of any of the designated break address.

Input format:

1. LOD FILE NAME
2. SAV FILE NAME START ADDRESS
   END ADDRESS
3. VER FILE NAME START ADDRESS
   END ADDRESS

By entering the command shown in (1) above, the content of the designated file (user program) can be loaded on the code memory on the emulation kit corresponding to the program memory (user program area) area of OLMS50/60 Series.

The command of (2), on the other hand, permits the content in the designated range of the code memory to be saved by a designated file name.

By entering the command of (3), the content of a designated file can be compared with the data in the code memory.

Input format:

1. DIAG FILE NAME

This command permits automatic execution of the command in the designated file.

Also, if used with the PAUSE command, it suspends the execution of the command in the file.

Input format:

1. LIST FILE NAME
2. NLST

By use of these two commands, a file with the same content as the console display can be prepared. If used in combination with the DIAG command mentioned above, it eliminates the operator's need to attend the host computer.

EMULATOR COMMANDS FOR EASE6052

DISK ACCESS COMMANDS

	LOD (dr :) filename	Lond Program into Code Memory
	SAV (dr :) filename (address address)	Save Code Memory Program
	VER (dr :) filename (address address)	Verify File With Code Memory
	DIAG (dr :) filename	Execute command file
	LIST (dr :) filename	List console Output into Disk file
	NLST	End listing
	FDD dr : ---dr :	Set Disk drive

EPROM PROGRAMMER COMMANDS

	PPR address address (address)	Program Code Memory into EPROM
	TPR address address (address)	Transfer EPROM into Code Memory
	VPR address address (address)	Verify EPROM with Code Memory
	TYPE mnemonic	Set EPROM type

CODE MEMORY COMMANDS

CM	DCM address (address)	Display Code Memory High & Low
	DCMH address (address)	Display Code Memory High
	DCML address (address)	Display Code Memory Low
	CCM address	Change Code Memory High & Low
	CCMH address	Change Code Memory High
	CCML address	Change Code Memory Low
	FCM address address data	Fill Code Memory High & Low
	FCMH address address data	Fill Code Memory High
	FCML address address data	Fill Code Memory Low

DATA MEMORY COMMANDS

DM	DDM address (address)	Display Data Memory
	CDM address	Change Data Memory
	FDM address data	Fill Data Memory

REGISTER COMMANDS

AC	DAC	Display Accumulator
	CAC data	Change Accumulator
C	DC	Display Carry Flag
	CC data	Chamge Carry Flag
G	DG	Display Greater Flag
	CG data	Change Greater Flag
Z	DZ	Display Zero Flag
	CZ data	Change Zero Flag
B	DB	Display Bank Register
	CB data	Change Bank Register
P	DP	Display Page Register
	CP data	Change Page Register
AR1	DAR1	Display Adderss Register 1
	CAR1 data	Change Adderss Register 1
AR2	DAR2	Display Adderss Register 2
	CAR2 data	Chamge Adderss Register 2
PC	DPC	Display Program Counter
	CPC data	Change Program Counter
SP	DSP	Display Stack Pointer
	CSP data	Change stack pointer
L	DL	Display Loop Counter
	CL data	Change Loop Counter

EMULATOR COMMANDS FOR EASE6052

REGISTER COMMANDS		
PT	DPTC	Display Program Timer Counter
	CPTC data	Change Program timer Counter
	DPTL	Display Program Timer out Latch
	CPTL data	Change program Timer out Latch
	DIC	Display Interrupt Control
	CIC data	Change Interrupt Control
T	DDIV	Display Timer Divider
	RDIV	Reset Timer Divider
	DTAC	Display Timer Active Condition
	CTAC data	Change Timer Active Condition
HS	DHS	Display Hook Switch
R14	DR14	Display Port R1-4
R58	DR58	Display Port R5-8
I14	DI14	Display Port I1-4
O14	DO14	Display Port O1-4
	CO14 data	Change Port O1-4
EO14	DEO14	Display Port EO1-4
	CEO14	Change Port EO1-4
C14	DC14	Display Port C1-4
	CC14 data	Change Port C1-4
	DCC	Display Port C1-4 Condition
	CCC data	Change Port C1-4 Condition
IO14	DCO14	Display Port IO1-4
	CIO14 data	Change Port IO1-4
	DIOC	Display Port IO1-4 Condition
	CIOC data	Change Port IO1-4 Condition
DTMF	DDTMF	Display DTMF Control Regiter
	CDTMF data	Change DTMF Control Regigter
BD	DBD	Display BD Control Register
	CBD data	Change BD Control Regiter
EMULATION COMMANDS		
	STP (number) (address)	Start Step Execution
	G (start-address) (break-address)	Go Start Real time Emulation
OTEHR COMMANDS		
	RES	Reset EASE6052 system
	RESE	Reset Evaluation Board
	SIO ForH	Set Emulater to I/O terminal made
	PAUCE	Stop command file execution
	EXIT	Exit to CP/M
	DIR (dr :)	Display file directory

OKI semiconductor
EASE80C49
PROGRAM DEVELOPMENT SUPPORT SYSTEM

APPLICABLE MODELS

This system provides the capability to develop and evaluate the program of 8-bit single chip CMOS microcontroller, OKI OLMS48 Series; MSM80C48, MS80C49, MSM80C50, MSM80C35, MSM80C39 and MSM80C40.

THE SYSTEM SUPPLIED CONTAINS

System	Class	Model	Name
EASE80C49	Hardware	EASE80C49 Emulation Kit	Emulation kit for single chip 8 bit microcontrollers, OLMS48 series.
	Software	EASE49	Floppy disk base host monitor[1)]
		ASM49	Floppy disk base cross-assembler[1)]
	Manual	TM-80C49	User's manual for program development system
	Accessories	TCU-80C49	Connection cable for user application system
		TCS-8	Connection cable for CP/M based host computer, if800 model 20/30
		TCP-8	Power supply cable; + 5V/3.5A, + 12V/0.2A and −12V/0.2A
		TCX-1	External probe for emulation
		TCC-8	PC board connection cable

[1)] CP/M non-system disk
- $5\frac{1}{4}$" 2DD if800 format
- 8" ISID IBM3740 soft sectored format

OUTLINE OF CROSS-ASSEMBLER

The ASM49 is a high performance cross-assembler that runs under either CP/M or MS-DOS[2]. It translates the source program that a user created on a disk by the use of an editor, generating the object file and the assemble list.

[2] MS-DOS version and PC-DOS version are optional softwares.

FEATURES

- The source program is a free description text.
- 35 pseudo instructions are available for assembler control.
- Macro definition is available.
- Eleven pseudo instructions for conditional assembling.
- Interactive process (loop process) is available in assembling.
- Veriables in the source program can be defined by an input from the console when assembling. Linkage of exsisting source programs is also possible. (INPUT, LINK pseudo instructions)
- Thirteen operators are available in the operand field of the source program.

LIST OF PSEUDO INSTRUCTIONS

Pseudo Instruction	Function
ORG	Origine the value of a location counter to nnn
DS	Reserve n byte area. The values of both starting and ending bytes are changed. When you do not want the change, use ORG $ + n.
DW	Define 16-bit data.
DB	Define 8-bit data or ASCII character strings.
EQU	Allocate an operand value to the name.
SET	The same as EQU except that it is re-assignable.
IF	Skip to the next ENDIF, END or EOF (end of file), if the expression is zero. Assemble the next command, if the expression is not equal to zero.
NIF	Skip to the next ENDIF, END or EOF (end of file), if the expression is not equal to zero.
END	Assemble end.
MACRO	Define macro.
MACND	Macro definition end.
GO TO	Branch to a labeled location.
REPT	Assemble is repeated the number of times specified by the expression value.
REPND	Define repetition block end.
LIST	Invalidate path option and output all the assemble list.
NOLIST	Inhibit output of the assemble list except error messages.
TITLE	Add a title at the top of the assemble list.

EASE80C49 EMULATOR

The EASE80C49 emulator consists of a CP/M based host computer and the EASE80C49 emulation kit, and provides effective supports for debugging work in any development stages of MSM80C49 and RS/MSM80C50RS;[3] from an application program without user application circuit to a complete system.

[3] It is necessary to exchange evaluation chip for MSM80C40, if debug a MSM80C40/MSM80C50RS application program.

Real Time Tracing Capability That Does Not Affect Execution

The EASE80C49 emulation kit has a real time tracing area with the capacity up to 1024 machine cycles. You can set at which address the tracing function is to be invoked. Each time the specified address is executed, status of port, program counter, probe data is traced.

Measurement of Execution Time

The execution time of the user program is measured by the use of the cycle counter of the EASE80C49 emulation kit (up to 16777215 cycles). The counter can be used as a pass counter to know how many times a specific address was executed.

4069 byte User Program Area

A 4096 byte static RAM area is provided on the EASE80C49 emulation kit, and thus user programs of a CP/M based host computer can be loaded or saved there. In addition, with an aid of EPROM writer of this board, the contents in the user program area are written into EPROM or the contents of EPROM are read into the user program area.

Various Break Functions

Eight kinds of means to break the program execution are available:

a) **BB (Breakpoint Break)**
The program stops when executing the address specified.(Breakpoint can be specified in any address).

b) **EB (External Break)**
Break occurs by external break signal.

c) **CO (Cycle Counter Over Flow Break)**
Break occurs after 4097 times of traces.

d) **TF (Trace Full Break)**
Break occurs after 4097 times of traces.

e) **PM (Probe Match Break)**
Break occurs when the probe data match a preset data.

f) **IF (Instruction Fetch)**
Break occurs when a machine code is fetched.

g) **MX (MOVX)**
Break occurs by MOVX command.

h) **ER (Error Code)**
Break occurs when a error machine code is fetched.

Various Debug Commands

In addition to the capabilities to display or change the contents of each register, port or RAM, there are a variety of DEBUG commands, which enable you to perform an effective debugging work:

a) **Execute user program**

Input format:
STP, NUMBER OF INSTRUCTION, START ADDRESS

The specified number of commands are executed starting from the specified address.

SDF command can be combined with this command to specify what to be displayed.

Input format:
G [, START ADDRESS]

The program starts to run from the specified address. Without any specification, the program starts from the address that the program counter indicates.

b) Use of floppy disk

Input format:
LOA, FILE NAME

This command loads the code memory on the EASE80C49 emuluation kit, which is equivalent to the mask ROM area of MSM80C49RS/ MSM80C50RS, with the contents of the specified file (user program.) With the use of LDU command, the same contents can be loaded on the utility buffer.

Input format:
SAV, START ADDRESS, END ADDRESS, FILE NAME

This command saves a specified range of contents of the code memory with the specified file name.

c) Instruction executed bit memory

This emulator has an instruction executed bit memory indicating at which address the user program was executed. You can see the program flow by looking at the contents of the memory.

Mnemonic Command

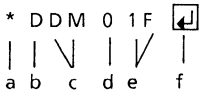
Each debug command of this emulator is mnemonic symbols followed by parameters (address or mnemonic code.)

Construction of Command Mnemonic:

a) The first symbol
Means what function to be performed by the emulator.

b) The second and the latter symbols
Designates the object name of the function to performed by the emulator. The object is specified by abbreviation of register, memory or port name of MSM80C39 evaluation chip or EASE80C49 emulation kit.

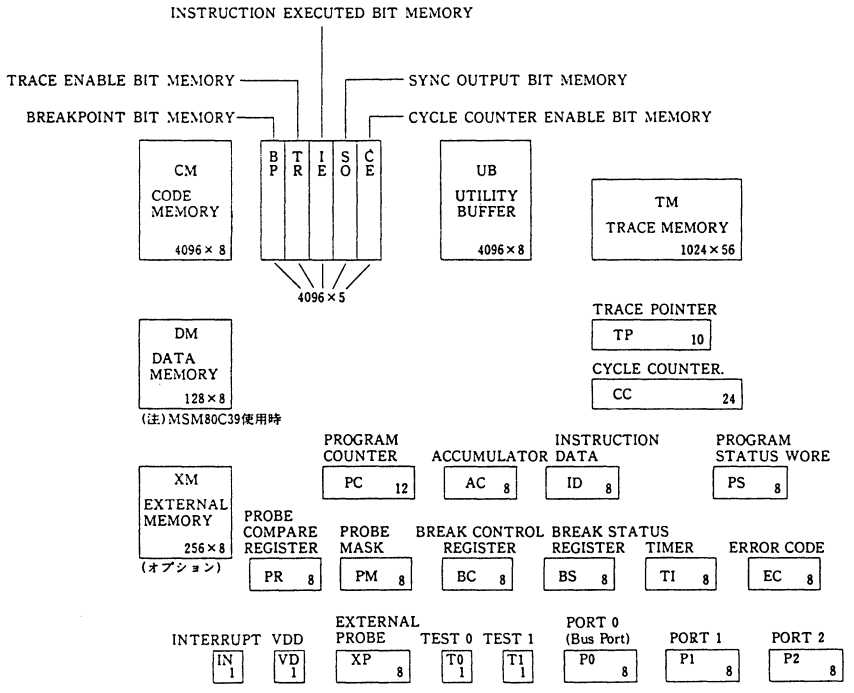
[Example]



a - Prompt for command input of emulator
b - Instruct to display of the contents of the specified object
c - Specify that the object is the data (corresponding to internal RAM area of MSM80C50RS)
d - Starting address of the data displayed.
e - Ending address of the data displayed.
f - Carriage return meaning the end of command input.

MEMORY MAP OF EMULATOR

The EASE80C49 emulator consists of a CP/M based host computer and the EASE80C49 emulation kit, and is designed for an effective debugging work of application programs of MSM80C49RS/MSM80C50RS. The block diagram below illustrates each section of EASE80C49 emulator available for user. The boxed parts are registers and memory areas accessible from the keyboard of the host computer and the contents can be displayed or changed by the commands of the emulator.



FUNCTION OF MAJOR MEMORY

- **Code Memory**

  This memory is a 4096 x 8-bit RAM area and equivalent to he program memory (Mask ROM) of MSM80C495S/MAM80C50RS, where the user program is stored.

- **Data Memory**

  This memory is a 128 x 8-bit RAM area and equivalent to the data memory of MSM80C49RS chip. In the case of MSM80C50RS, it is a 256x8-bit memory size.

- **Utility Buffer**
  This is a 4098 x 8-bit RAM area and used as a temporary saving location by emulator when load, save or transfer command is inputted.

- **Breakpoint Bit Memory**

  This is a 4096 x 1-bit RAM area with the same addresses as the code memory. When the bit is 1, the program stops immediately after executing the contents of the code memory corresponding to the address.

- **Trace Enable Bit Memory**
  This is a 4096 x 1-bit RAM area having the same address as the code memory. If the bit is set 1, probe data are stored in the trace memory when the content of the code memory corresponding to the address is executed.

- **Sync Output Bit Memory**
  This is a 4096 x 1-bit RAM area with the same addresses as the code memory. When the contents of the code memory are executed, each bit of this memory corresponding to the address of the code memory is checked; if the bit is 1, a sync signal (active LOW) is outputted to the probe terminal.

- **Cycle Counter Enable Bit Memory**
  This is a 4096 x 1-bit RAM area with the same addresses as the code memory. When the contents of the code memory are executed, each bit of this memory corresponding to the address of the code memory is checked; if the bit is 1, the cycle counter is incremented in accordance with the machine cycle.

- **Probe Compare Register/Probe Mask**
  Conditions to activate PROBE MATCH BREAK can be changed with settings of these two.

- **Trace Memory**
  This is a 1024 x 56-bit RAM area and trace information is stored here. Trace instruction is done by the trace enable bit.

USERABLE CHARACTER

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz 1234567890. . @ + −
Cntl/C Cntl/P Cntl/Q Cntl/R Cntl/S Cntl/T Escape Space

DISPLAY COMMAND

1.	DAC	Display Accumulator
2.	DBP,000[,FFF]	Display Breakpoint Bit(s) Status
3.	DBS	Display Break Status
4.	DCC	Display Cycle Counter
5.	DCE,000[,FFF]	Display Cycle Counter Enable Bit(s) Status
6.	DCM,000[,FFF]	Display Code Memory
7.	DDM,00[,FF]	Display Data Memory
8.	DIE,000[,FFF]	Display Instruction Executed Bit(s) Status
9.	DP0	Display Port 0 (Bus port)
10.	DP1	Display Port 1
11.	DP2	Display Port 2
12.	DPC	Display Program Counter
13.	DPS	Display Program Status Word, Test 0, Test 1, Interrupt Pin
14.	DRG	Display Registers R0 thru R7
15.	DSK	Display Stack
16.	DSO,000[,FFF]	Display Sync Output Bit Memory
17.	DTI	Display Timer
18.	DTM,XXXX, YYYY	Move Trace Pointer (XXXX) and display Trace Memory (YYYY)
19.	DTP	Display Trace Pointer
20.	DTR,000[,FFF]	Display Trace Enable Bit(s) Status
21.	DVD	Display Vdd Pin
22.	DXM,00[,FF]	Display External Memory
23.	DXP	Display External Probe Byte, Probe Mask and Probe Compare Register

CHANGE COMMAND

1.	CAC,FF	Change Accumulator
2.	CCC,ZZZZZZZZ	Change Cycle Counter (ZZZZZZZZ is a positive or negative decmal number)
3.	CCM,FFF,FF	Change Code Memory
4.	CDM,7F, FF	Change Data Memory
5.	CP0,FF	Change Port 0 (Bus Port)
6.	CP1,FF	Change Port 1
7.	CP2,FF	Change Port 2
8.	CPC,FFF	Change Program Counter
9.	CPM,BBBBBBBB	Change Probe Mask (BBBBBBBB is a binary number)
10.	CPR,BBBBBBBB	Change Probe Compare Register (BBBBBBBBis a binary number)
11.	CPS,FF	Change Program Status Word
12.	CTI,FF	Change Timer
13.	CXM,FF,FF	Change External Memory

FILE COMMAND

1.	FCM,000,FFF,FF	Fill Code Memory
2.	FDM,00,FF,FF	Fill Data Memory
3.	FXM,00,FF,FF	Fill External Memory

ENABLE COMMAND

1.	EBP,000[,FFF]	Enable Breakpoint Bit(s)
2.	ECE,000[,FFF]	Enable Cycle Counter Bit(s)
3.	ESO,000[,FFF]	Enable Sync Output Bit(s)
4.	ETR,000[,FFF]	Enable Trace Bit(s)

RESET COMMAND

1. RBP,000[,FFF]	Reset Breakpoint Bit(s)
2. RCE,000[,FFF]	Reset CyCle Counter Enable Bit(s)
3. RIE,000[,FFF]	Reset Instruction Executed Bit(s)
4. RSO,000[,FFF]	Reset Sync Output Bit(s)
5. RTR,000[,FFF]	Reset Trace Enable Bit(s)
6. RES	Reinitialize MPB800 System

UTILITY/DISK INPUT/OUTPUT COMMAND

1. CUB,FFF,FF	Change Utility Buffer
2. DUB,000[,FFF]	Display Utility Buffer
3. FUB,000,FFF,FF	Fill Utility Buffer
4. TCM,000,FFF	Transfer Code Memory into Utility Buffer
5. TUB,000,FFF	Transfer Utility Buffer into Code Memory
6. TST	Test
7. LOA,filename	Load Program into Code Memory
8. LDU,filename	Load Program into Utility Buffer
9. SAV,000,FFF,filename	Save Code Memory Program
10. SVU,000,FFF,filename	Save Utility Buffer Program

EPROM WRITER COMMAND

1. PPR,000,FFF,FFF	Program Code Memory onto EPROM
2. VPR,000,FFF,FFF	Verify EPROM with Code Memory
3. TPR,000,FFF	Transfer EPROM into Code Memory

EMULATION COMMAND

1. G[,HHH]	Begin real time emulation, HHH is the start address (hex)
2. STP[,EEEE] [,HHH]	Step emulation, where EEEE is the decimal number of instruction to execute, and HHH is the start address (hex).
3. SBC,mnemonic[,mnemonic]	Set/Reset Break Control Bit
4. SDF,mnemonic[,mnemonic]	Set/Reset Dump Format

EDIT OF COMMAND LINE AND KEY BOARD OPERATION

1. Rubout	Delete the last character entered
2. Cntl/C	Return to CP/M
3. Cntl/P Copy all subsequent console output to the currently assigned list device.	Output is sent to both the list device until the next Cntl/P is typed.
4. Cntl/Q	Continue normal display
5. Cntl/R	Echoe current input line
6. Cntl/S	Stop display
7. Escape	Abort any command inprogress
8. @	Repeat last command

OKI semiconductor
EASE80C51mkII

PROGRAM DEVELOPMENT SUPPORT SYSTEM

APPLICABLE MODELS

This system has been designed with OKI's original concept and technology for time-saving and effective program development and evaluation of 8-bit single chip CMOS microcontrollers, OKI80C51 series: MSM80C51/C31, MSM80C154, and MSM83C154.

SUPPLY FORMATS

System	Class	Model	Name
EASE 80C51mkII	Hardware	EASE80C51mkII Emulation Kit	Emulation kit for MSM80C154/MSM80C51
	Software	EASE	Host monitor, floppy disk*
		ASM51	Cross-assembler, floppy disk*
	Manual	TM-80C51mkII	User's manual for EASE80C51mkII program development system
		AM-80C51mkII	User's manual for ASM51 cross-assembler
	Accessories	TCU-80C51	Connection cable for user application system
		TCS-2	Connection cable for CP/M or PC-DOS or IBM PC based host computer, if800 model 20-30 or IBMPC
		TCP-2	AC power supply cable (100V/220V)
		TCX-2	External probe for emulation

* CP/M file 5¼-inch double-side double-density, 8-inch single-side single-density
- MS-DOS file (Optional software)
- PC-DOS 5¼-inch double-side double-density

OUTLINE OF CROSS-ASSEMBLER

The ASM51 is a higher performance cross-assembler that runs on either CP/M or MS-DOS. It translates the source program that a user created on a disk by the use of a commercial available editor, generating the object file (in INTEL HEX format), the assembly list file, the cross reference list file and the symbol list file on a specified device.

OUTPUT FILE

- Object file (in an INTEL HEX format)
- Assembly list file
- Cross-reference list file
- Symbol list file

PART OF PSEUDO COMMANDS

Pseudo Instruction	Function
EQU	Allocate an operand value to the name
SET	The same as EQU except that it is re-assignable.
ORG	Origin the starting addresses of a program.
END	Assemble end.
JMP	Check the destination and automatically convert to relative, 2K-page range or full page jump command.
CALL	Check the destination and automatically convert to relative, 2K-page range or full page jump command.
DB	Define 8-bit data or ASCII characters
DW	Define 16-bit data
DS	Reserve memory area specified in byte

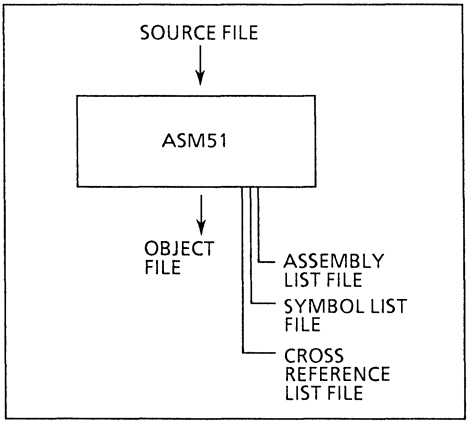
PART OF ASSEMBLER CONTROL STATEMENTS

Pseudo Instruction	Function
DATE	Add a date in the title portion of the assembly listing
EJECT	Eject page
TITLE	Add a title to the assembly listing
LIST	
NOLIST	Inhibit assembly list output
DEBUG	Give debug symbol information to the object file

FEATURES

- The source program is a free description text.
- Thirteen kinds of operators are available for each operand field of the source program.
- The character number per line and the line number per page of the assembly listing can be specified. Twenty-two pseudo commands and six assemble control statements are available.
- LIST and NOLIST commands are capable of controlling output of the assembly list file.
- With the use of DEBUG control statements, the object file can obtain debugging information necessary for symbolic debugging.
- Output file and device can be specified when invoking assembler.

OUTLINE VIEW OF ASM51 ASSEMBLER



OUTLINE OF EMULATOR

The EMSE80C51mkII emulator is connected to a CP/M based host computer through a RS-232C interface, provides support for effective debugging work of application programs with MSS8051 series in any development stage; from an application program without user application circuit to a complete system.

REAL TIME EMULATION

Emulation is performed without having to insert a wait state because of the use of an exclusive evaluation chip of MSM85C154E.

MEASUREMENT OF EXECUTION TIME

The execution time of the user program is measured by the use of the cycle counter of this emulation kit. (Up to 16777215 cycles). Starting and ending addresses of the cycle counter are set by command.

WIDE USER PROGRAM AREA

A 64K byte RAM area (code memory) is implemented on this emulator, and thus all the user programs of the CP/M based host computer can be loaded into the RAM area (See note 1.) Mapping command is capable of assigning the program area in the RAM area on the emulation kit or in the ROM on a user application circuit in a 4K byte unit. EMROM writer is implemented on the emulator so you can write the contents of the user program area in EPROM and read the contents of EPROM.

(Note 1: Available for i2732, 2732A, 2764, 27128)

VARIOUS BREAK FUNCTIONS

Various kinds of means to interrupt or break the program execution are available. Each can be set or removed.

a) **BreakPoint Break**
The breakpoint can be set at any addresses. The program stops when executing the program whose address is the same as where the breakpoint is specified.

b) **Address Break**
The program stops when executing the program at the address specified by the emulation command. You can also set a break so the program may stop after executing a specified address n times.

c) **Power Down Break**
The program stops when power-down status occurs.

d) **Externally Forced Break**
You can stop the program by sending a low level of signal through the attached probe cable.

e) **Break Occurs When the Trace Memory Is Full with Information.**

f) **Break Occurs When the Cycle Memory Overflows.**

g) **Break Depending on the Contents of an Internal RAM/SFR Area.**
When the program at a specified address is run, the contents of the specified internal RAM/SFR area of MSM85C154E are checked; if the contents match the specified number of times, break occurs.

COMPLETE READ TIME TRACING FUNCTION

This emulator has to types of memories to realize the real time tracing capability which do not affect execution time:

- **Trace Memory**
Status information of port, carry flag and accumulator of MSM85C154E when instructions are executed is stored in this memory for tracing up to 3048 machine cycles. Three different tracing functions are available:

a) Tracking is performed each time a specified address is executed.

b) Tracing is performed when a special command such as ACALL, AJMP, LCALL, LJMP, RET, RETI, PUSH, POP, are executed.

c) Tracing is controlled by a start/stop bit.

- **Flash Trace Memory**
This memory is used to trace the status of the internal RAM or a full SFR area of SM85C154E when the instruction at the address specified by emulation command (debug command) is executed.
The contents of the internal RAM or SFR area can be stored up to 16 times of execution.

(Note 1) Standard memory:
64K byte code memory, 16K byte external data memory

MNEMONIC EMULATOR COMMAND

Each emulator command (debug command) of this emulator is mnemonic symbols followed by parameters (address or mnemonic code.)

a) **The First Symbol**
Means what function to be performed.

b) The Second and the Latter Symbols

Designates the object of the function to be performed by the emulator. The object is specified by an abbreviation of the name of register, memory or port of the MSM80C51mk/II emulation kit.

[Example]

a - Prompt for command input of emulator
b - Instruct to display of the contents of the specified object
c - Designation of data memory (corresponding to internal RAM area of MSM80C50RS)
d - Starting address of the data displayed.
e - Ending address of the data displayed.
f - Carriage return meaning the end of command input.

In addition to the capability of displaying or changing the contents of each register, port, etc. of the exclusive evaluation chip MSM85C154E, this emulator provides various emulation commands to achieve an effective debugging work.

Input format:

STP NUMBER START ADDRESS

The user program is executed by the specified number of commands starting from the specified address (starting address.)

SSF command can be combined with this command to specify what to be displayed.

Input format:

G START ADDRESS, PARAMETER

PARAMETER INPUT FORMAT

1. BREAK ADDRESS ····, BREAK ADDRESS
2. BREAK ADDRESS (n)
3. BREAK ADDRESS RAM ram ADDRESS (byte-n)
4. BREAK ADDRESS SFR MNEMONIC (byte-n)

(1) The user program starts from the specified address and stops after executing one of specified addresses.

(2) The user program starts from the specified address and stops after executing the specified address n times.

(3) The program starts to run from the specified address. Without any specification, the program starts at the address that the program counter indicates.

(4) The user program starts from the specified starting address. When the user program at the break address is executed, the content of the specified address on internal RAM of MSM85C154E is checked and compared with the specified content (byte); if coincides happens the number of times (n) specified, break occurs.

Input format:

1. LOD FILE NAME
2. SAV FILE NAME START ADDRESS END ADDRESS
3. VER FILE NAME START ADDRESS END ADDRESS

(1) Commands in the file specified are automatically executed. PAUSE command can pause the execution of commands in the file.

(2) Execute a series of commands defined by MAC command.

(3) Execute a front command.

Input format:

1. **DIAG FILE NAME**
2. **M**
3. **@**

(1) Commands in the file specified are automatically executed. PAUSE command can pause the execution of commands in the file.

(2) Execute a series of commands defined by MAC command.

(3) Execute a front command.

Format:

1. LIST file name
2. NLST

With the use of these two commands, a file with the same contents as in the console display can be generated. Combination with DIAG command can eliminate the necessity of an operator sitting at the CP/M based host computer.

Format:
S tm mnemonic data number of times

Enables to search data of the specified trace information (tm mnemonic) in the trace memory.

DISK ACCESS COMMAND

1.	LOD [dr:] filename	Load Program into Code Memory
2.	SAV [dr:] filename[address address]	Save Code Memory Program
3.	VER [dr:] filename[address address]	Verify File with Code Memory
4.	DIAG [dr:] filename	Execute command file
5.	LIST [dr:] filename	List console Output into Disk file
6.	NLST	End listing
7.	HELP	Display HELP File
8.	FDD dr:_ dr:	Set Disk Drive

EPROM WRITER COMMAND

1.	PPR address address [address]	Program Code Memory into EPROM
2.	TPR address address [address]	Transfer EPROM into Code Memory
3.	VPR address. address[address]	Verify EPROM with Code Memory
4.	TYPE mnemonic	Set EPROM type

ASSEMBLE COMMAND

1.	ASM address	Assemble to Code Memory
2.	DASM address	Disassemble to Console
3.	DBLK address[address]	Display Block Memory
4.	SPB address[address]	Set Program Block
5.	SDB address[address]	Set Data Block

TRACE COMMAND

1.	DTM-number number	Display Trace Memory
2.	DFTM number number	Display Flash Trace Memory
3.	DTG	Display Trigger Mode
4.	DFA	Display Flash Trace address
5.	S mnemonic data number	Search Trace Memory date
6.	SFF mnemonic	Set Flash Trace Display format
7.	STG mnemonic	Set Trigger Mode
8.	SFA address[... address]	Set Flash Trace address
9.	RFA address[... address]	Reset Flash Trace address
10.	RTG	Reset Trigger Mode

EMULATION COMMAND

1. STP number[number]
2. G[st-address] [,break-parameter]
   If the optional st-address is given, emulator will begin emulation from st-address.
   And if optional break-parameter is given,
   emulation will break on the first break-parameter to be satisfied.

   break-parameter = break-address,_ ,break-address (max. 10)
   break-address (pass count)
   break-address RAM ram-address (byte-pass count)
   break-address sfr-mnemonic (byte-pass count)
3. SSF (+/−) mnemonic[... (+/−)mnemonic]
   Set STP Command Display Format

DISPLAY COMMAND

1.	D	Display Register. Flag. Port
2.	Dsfr-mnemonic	Display sfr-mnemonic data
3.	DSFR mnemonic [... mnemonic]	Display mnemonic data
4.	DPC	Display Program Counter
5.	DREG	Display all Register Bank
6.	DSBUF	Display receiver data

CHANGE COMMAND

1.	Csfr-mnemonic data	Change sfr-mnemonic
2.	CSFR [mnemonic... mnmonic]	Change SFR
3.	CPC data	Change Program Counter
4.	CREG data	Change Register bank
5.	CSBUF data	Change Transmitter data

CODE MEMORY & DATA MEMORY COMMAND

1.	DCM address[abbress]	Display Code Memory
2.	DDM address[address]	Display Data Memory
3.	DXDM address[address]	Display External Data Memory
4.	CCM address	Change Code Memory
5.	CDM address	Change Data Memory
6.	CXDM address	Change External Data Memory
7.	FCM address address byte	Fill Code Memory with byte
8.	FDM address address byte	Fill Data Memory with byte
9.	FXDM address address byte	Fill External Data Memory with byte

ATTRIBUTE MEMORY COMMAND

1.	DBP address [address]	Display Break Point Bit Memory
2.	DTR address [address]	Display Trace Enable Bit Memory
3.	DSO address [address]	Display Sync Output Enable Bit Memory
4.	DIE adresss [address]	Display Instruction Executed Bit Memory
5.	EBP address .. address	Enable Break Point Bit
6.	ETR address .. address	Enable Trace Enable Bit
7.	ESO address .. address	Enable Sync Output Enable Bit
8.	FBP address address byte	Fill Break Point Bit Memory with byte
9.	FTR adresss address byte	Fill Trace Enable Bit Memory with byte
10.	FSO address address byte	Fill Sync Output Enable Bit Memory with byte
11.	RBP address .. address	Reset Break Point Bit
12.	RTR address .. address	Reset Trace Enable Bit
13.	RSO address .. address	Reset Sync Output Enable Bit
14.	RIE	Reset Instruction Executed Bit Memory

BUFFER MEMORY COMMAND

1.	DBUF	Display Buffer Memory
2.	TBUF	Transfer Data Memory & SFR Data into Buffer Memory
3.	LBUF	Load Buffer Memory into Data Memory & SFR

CYCLE COUNTER COMMAND

1.	DCC.	Display Cycle Counter
2.	CCC data	Change Cycle Counter
3.	TIME data	Set 1 cycle time

BREAK CONDITION & STATUS COMMAND

1.	DBC	Display Break Condition
2.	DBS	Display Break Status
3.	SBC (+/−)mnemonic.. (+/−)mnemonic	Set Break

OTHER COMMAND

1.	RES	EASE80C51 System initialization
2.	RES E	MSM80C51E eva-chip reset
3.	SIO F or H	Set Emulator to I/O terminal mode
4.	PAUSE	Stop command file execution and wait key-in
5.	EXIT	Return Host Computer's OS
6.	DIR[dr:]	Display file directory
7.	MAP[address]	Mapping
8.	MAC	Define command execution
9.	M	Execute Defined command
10.	@	Repeat front command
11.	M80C51	Set emulation mode to MSM80C51
12.	M83C154	Set emulation mode to MSM83C154

OKI semiconductor

MAC51 SUPPORT SOFTWARE PACKAGE for the

MSM80C51 CMOS 8-BIT MICROCONTROLLER

MAC51 SUPPORT SOFTWARE PACKAGE

The following MAC51 programs are available to develop user programs.

- **MP51 Pre-Processor**
  MP51 is used to expand macro calls, conditional assembly statements, and INCLUDE statements included in user generated source programs, thereby generating expanded source programs.

- **MAC51 Assembler**
  MAC51 converts source programs into relocatable codes to form relocatable object files (OBJ files). Print, symbol, cross reference, and error files are generated as assembly information.

- **LIB51 Librarian**
  The LIB51 program manages OBJ files for each module. Files consisting of a number of relocatable object modules (OBJ modules) generated by LIB51 are called the object library. LIB51 handles object library generation, and OBJ module addition, deletion, and upgrading.

- **RL51 Linker**
  RL51 links and relocates one or more OBJ modules to generate one absolute object module. RL51 also generates a list file consisting of symbol table and link map as link information. OBJ modules which serve as the RL51 input can be OBJ files generated by MAC51, and OBJ modules located within the object library generated by LIB51.

- **SID51 Symbolic Debugger**
  SID51 is used when a symbol using debugger is selected. Absolute object files are converted to Intellec HEX format files.

- **OBJHEX Converter**
  OBJHEX converts Object file into Intellect HEX file.

OUTLINE OF PROGRAM DEVELOPMENT

The procedures involved in the processing from source program generation through ROM loading are described below in the sequence indicated in Figure 1-1.

1. Generation of MAC51 assembly language source program by the editor. Source programs can contain basic and pseudo instructions, and assembler control, macro call, conditional assembly, and INCLUDE statements.
2. When macros are used, macro definitions are generated in macro library files (with MAC extension) by the editor. Macro call statements are described within source programs.
3. Where macro call, conditional assembly statement, and INCLUDE statement descriptions are included in a source program, there are expanded by MP51 to form an expanded source program.

4. Source programs or expanded source programs are assembled by MAC51 to form relocatable object files.
5. A group of relocatable object files can be managed by LIB51 together in a relocatable object library files (LIB extension). When required, object modules can be called from this object library by RL51.
6. Relocatable object files are converted to absolute object files by RL51. At this stage, one or more relocatable object files can be linked to relocatable object modules in the library file.
7. ABS modules are converted by SID51 or OBJHEX to Intellec HEX format files. With EASE80C51mkII, HEX file contents can be written into EPROM devices.

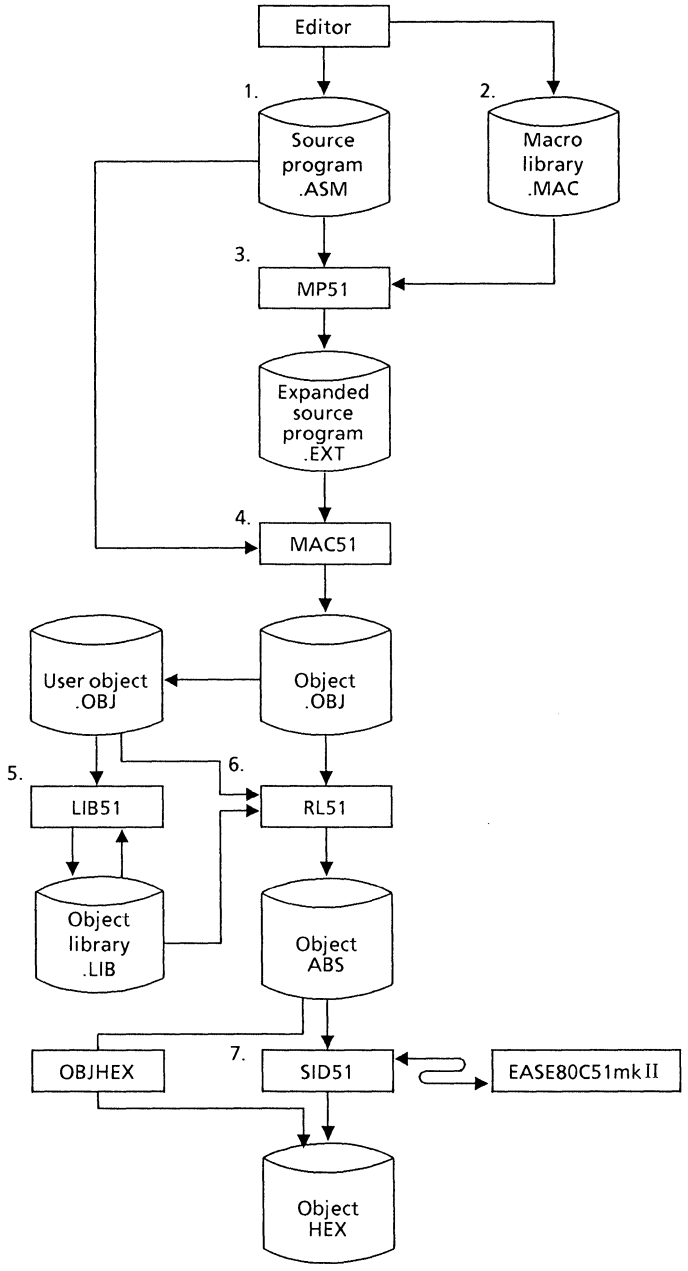


Fig. 1-1 Program Development Flow

OKI semiconductor
EASE62780
PROGRAM DEVELOPMENT SYSTEM

APPLICABLE MODELS

This system has been designed with OKI's original concept and technology for time-saving and effective program development and evaluation of 8-bit single CMOS micro controllers, OKI MSM62780 and MSM62720.

OUTLINE OF CROSS-ASSEMBLER

AS62780 is a higher performance floppy disk assembler that runs on MS-DOS.

It translates the source program that a user created on a disk by the use of a commercially available editor, generating the object file (in INTEL HEX format) and the assembly list file.

OUTPUT FILE

- Object File (in INTEL HEX format)
- Assembly List File

LIST OF PSEUDO INSTRUCTIONS

Pseudo Instruction	Function
EQU	Allocate an operand value to the name
SET	The same as EQU except that it is re-assignable.
ORG	Origin start address of program
END	Assemble end
DB	Define 8-bit data or ASCII characters
DW	Define 16-bit data
DS	Reserve memory area specified in byte
TITLE	Give a title to the assembly listing
DATE	Add a date in the title portion of the assembly listing
PAGE	Specify the number of column of the assembly listing and eject page

OPERATOR

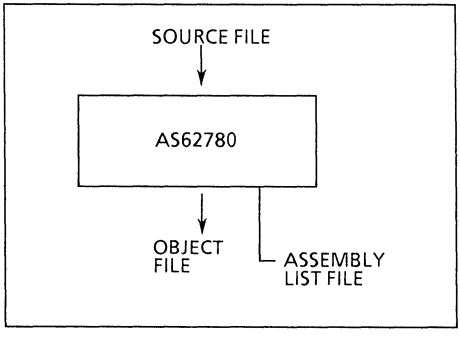
The table below shows the precedence numbers of the operators. One (1) is the highest precedence number and six (6) is the lowest precedence number. Operations with the same precedence number are performed sequentially from left to right.

	Precedence No.	Operator
◦ Parenthesis	1	( )
◦ Bit address operator	2	•
◦ Multiplication/Division/Modulo	3	* / %
◦ Addition/Subtraction	4	+ −
◦ Left shift/Right shift	5	≪ ≪ ≫ ≫
◦ Bit AND	6	&
◦ Bit EOR	7	∧
◦ Bit OR	8	|

FEATURES

- Eleven (11) kinds of operations can be described in the operand field of a source program.
- The character number per line and the line number per page of the assembly list can be specified.
- Sixteen (16) pseudo instructions are available.
- The allocation map of the internal data RAM is outputted.
- The output file and device can be specified when activating assembler.

AS62780 ASSEMBLER FUNCTIONAL OUTLINE



OUTLINE OF EMULATOR

The EASE62780 emulator is connected to a host computer through a RS-232C interface, and provides support for effective debugging work of application programs with MSM62780 and MSM62720 in any development stage; from an application program without user application circuit to a complete system.

REAL TIME EMULATION

Emulation is performed without having to insert a wait state because of the use of an exclusive evaluation chip of MSM62E780.

MEASUREMENT OF EXECUTION TIME

The execution time of the user program is measured by the use of the cycle counter of this emulation kit. (Up to 65536 cycles). Starting and ending addresses of the cycle counter are set by command.

USER PROGRAM AREA

A 8K byte RAM area (code memory) is provided on this emulator, and thus all the user programs of the host computer can be loaded into the RAM area. Furthermore, EPROM writer is implemented on the emulator so you can write the contents of the user program area in EPROM and read the contents of EPROM. (Available for i2732, 2732A, 2764, 27128).

VARIOUS BREAK FUNCTIONS

Various kinds of means to interrupt or break the program execution are available. Each can be either set or removed.

a) **Breakpoint Break**
The breakpoint can be set at any addresses. The program stops when executing the program whose address is the same as where breakpoint is set.

b) **Address Break**
The program stops when executing the program at the address specified by the emulation command. You can also set a break so the program may stop after executing the specified address n times.

c) **Break Occurs when the Trace Memory is Full with Information.**

d) **Break Occurs when the Cycle Memory Overflows.**

e) **Break Depending on the Contents of an Internal RAM/SFR Area**
When the program at a specified address is executed, the specified contents of the internal RAM/ register of MSM62E780 are checked; if the contents match the specified number of times, break occurs.

f) **Break Depending on the Contents of a Serial Data.**

COMPLETE REAL TIME TRACING FUNCTION

This emulator has two types of memories to realize real time tracing capability which do not affect execution time:

- **Trace Memory**
Status information of each register, carry flag and accumulator of MSM62E780 when instructions in the code memory are executed is stored in this memory for tracing up to 2048 machine cycles. Two different tracing functions are available:

a) Tracing is performed each time a specified address is executed.

b) Tracing is controlled by a start/stop bit.

- **Serial I/O Trace Memory**
Input data to Serial I/O port in the range of the trace starting address to

the ending address are changed to 8-bit data and then the 8-bit data are stored in this memory up to 2048 data.

MNEMONIC EMULATOR COMMAND

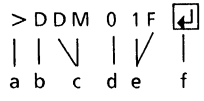
Each emulator command (debug command) of this emulator is mnemonic symbols followed by parameters (address or mnemonic code.)

Construction of Command Mnemonic:

a) First Symbol
Means what function to be performed.

b) Second and the Latter Symbols
Designates the object of the function to be performed by the emulator. The object is specified by an abbreviation of register, memory or port name of the EASE62780 emulation kit.

[Example]



a - Prompt for command input of emulator
b - Instruct to display the contents of the specified object
c - Specify that the object is the data memory (corresponding to internal RAM area of MSM80C50RS)
d - Starting address of the data displayed.
e - Ending address of the data displayed.
f - Carriage return meaning the end of command input.

In addition to the capability of displaying or changing the contents of each register, port, etc. of the exclusive evaluation chip MSM62E780, this emulator provides various emulation commands to achieve an effective debugging work.

Input format:
STP NUMBER START ADDRESS

The user program is executed by the specified number of commands starting from the specified address.

Input format:
G START ADDRESS, PARAMETER
PARAMETER INPUT FORMAT
1. **BREAK ADDRESS ····, BREAK ADDRESS**
2. **BREAK ADDRESS (n)**
3. **BREAK ADDRESS RAMram ADDRESS (byte-n)**
4. **BREAK ADDRESS REGISTER (byte-n)**
5. **SIO (byte)**

(1) The user program starts from the specified address and stops after executing one of specified addresses.

(2) The user program starts from the specified address and stops after executing the specified address n times.

(3) The user program starts from the specified starting address. When the user program at the break address is executed, the content of the specified address on internal RAM of MSM62E780 is checked and compared with the specified content (byte); if coincidence happens the number of times (n) specified, break occurs.

(4) The user program starts from the specified starting address. When the user program at the break address is executed, the specified content of register of MSM62E780 is checked; if the content coincides with the specified value (byte) the number of times (n) specified, break occurs.

(5) Input and output data of serial I/O are converted to 8 bit data, and if the data match a specified value (byte), break occurs.

Input format:

```
1. LOD    FILE NAME
2. SAV    FILE NAME  START ADDRESS
                     END ADDRESS
3. VER    FILE NAME  START ADDRESS
                     END ADDRESS
```

(1) The contents of the file specified are loaded into the code memory on the EASE62780 emulation kit, which is equivalent to the program memory (user program area) of MSM62780.

(2) The contents within the range specified of the code memory are saved with the file name specified.

(3) Compare the contents of the file specified with the code memory content.

Input format:

```
1. DIAG   FILENAME
2. @
```

(1) Commands in the file specified are automatically executed.

(2) Execute a front command.

Input format:

```
1. LIST   FILENAME
2. NLST
```

With the use of these two commands, a file with the same contents as in the console display can be generated. Combination with DIAG command can eliminate the necessity of an operator sitting at the host computer.

Input format:

```
Stm MNEMONIC DATA NUMBER
    OF TIMES
```

Enables to search data of the specified trace information (tm mnemonic) in the trace memory.

DISK ACCESS COMMANDS

1.	LOD	[dr:] filename	Load Program into Code Memory
2.	SAV	[dr:] filename [address address]	Save Code Memory Program
3.	VER	[dr:] filename [address address]	Verify File with Code Memory
4.	DIAG	[dr:] filename	Execute command file
5.	LIST	[dr:] filename	List console Output into Disk file
6.	NLIST		End listing
7.	HELP		Display HELP File
8.	FDD dr:... dr:		Set Disk Drive

EPROM PROGRAMMER COMMANDS

1.	PPR address address [address]	Program Code Memory into EPROM
2.	TPR address address [address]	Transfer EPROM into Code Memory
3.	VPR address address [address]	Verify EPROM with Code Memory
4.	TYPE mnemonic	Set EPROM type

ASSEMBLE COMMANDS

1.	ASM address	Assemble to Code Memory
2.	DASM [address] [address]	Disassemble to Console

TRACE COMMANDS

1.	DTM-number number	Display Trace Memory
2.	DSM number number	Display Serial Trace Memory
3.	DTG mnemonic	Display Trigger Mode
4.	S mnemonic data number	Search Trace Memory data
5.	SST	Set Serial I/O Trace
6.	RST	Reset Serial I/O Trace
7.	STG mnemonic	Set Trigger Mode
8.	RTG mnemonic	Reset Trigger Mode

EMULATION COMMANDS

1. STP number [st-address]
2. G [st-address] [break-parameter]
   if the optional st-address is given, emulator will begin emulation from st-address.
   And if optional break-parameter is given.
   emulation will break on first break-parameter to be satisfied.

break-parameter = break-address ..... break-address (max. 10)
break-address-pass count
[break-addredd] mnemonic (byte)
mnemonic = AR, BR, DR, MD
[break-address] RAM ram-address (byte)
break-address EM address (byte[bit])
SIO (byte)
SIO. bit (0 or 1)

EVA CHIP COMMAND

1.		D	Display Register, Flag
2.	PC	DPC	Display Program Counter
		CPC data	Change Program Counter
3.	AC	DAR	Display Accumulator
		CAR data	Change Accumulator
4.	BR	DBR	Display B Register
		CBR data	Change B Register
5.	DR	DDR	Display D Register
		CDR data	Change D Register
6.	SP	DSP	Display Stack Pointer
		CSP data	Change Stack Pointer
7.	FLAG	DF	Display Flag
		CF data	Change Flag
8.	REG	DRG	Display Register 0 - 7
		CRG	Change Register 0 - 7

CODE MEMORY & DATA MEMORY COMMANDS

1.	DCM address [address]	Display Code Memory
2.	DDM address [address]	Display Data Memory
3.	DEM address [address]	Display EEPROM data
4.	CCM address	Change Code Memory
5.	CDM address	Change Data Memory
6.	CEM address	Change EEPROM data
7.	FCM address address byte	Fill Code Memory with byte
8.	FDM address address byte	Fill Data Memory with byte
9.	CLEM	Clear EEPROM

ATTRUBUTE MEMORY COMMANDS

1.	DBP address [address]	Display Break Point Bit Memory
2.	DTR address [address]	Display Trace Enable Bit Memory
3.	DIE address [address]	Display Instruction Executed Bit Memory
4.	EBP address .. address	Enable Break Point Bit
5.	ETR address .. address	Enable Trace Enable Bit
6.	FBP address address byte	Fill Break Point Bit Memory with byte
7.	FTR address address byte	Fill Trace Enable Bit Memory with byte
8.	RBP address .. address	Reset Break Point Bit
9.	RTR address .. address	Reset Trace Enable Bit
10.	RIE	Reset Instruction Executed Bit Memory

CYCLE COUNTER COMMANDS

1.	DCC	Display Cycle Counter
2.	CCC data	Change Cycle Counter
3.	TIME data	Set 1 cycle time

BREAK CONDITION & STATUS COMMANDS

1.	DBC	Display Break Condition
2.	DBS	Display Break Status
3.	SBC (+/-) mnemonic .. (+/-) mnemonic	Set Break Condition

OTHER COMMANDS

1.	RES	EASE62780 System initialization
2.	RES E	MSM62E780 eva-chip reset.
3.	SIM F or H	Set Emulator to I/O terminal mode
4.	DEV	CH2 Device Configuration
5.	EXIT	Reture host Computer's OS
6.	DIR [dr:]	Display fill directory
7.	M780	Set CPU type to MSM62780
8.	M720	Set CPU type to MSM62720
9.	@	Repeat front command

OKI semiconductor
EASE66301
PROGRAM DEVELOPMENT SYSTEM

APPLICABLE MODELS

This is a high performance system dedicated for OKI's 8/16-bit single chip CMOS microcontroller and has been designed with OKI's concept and technology for time-saving and effective program development and evaluation work with MSM66301.

THE SYSTEM SUPPLIED CONTEAINES

System	Class	Model	Name
EASE66301	Hardware	EASE66301 Emulation Kit	Emulation kit for MSM66301
	Software	EASE66	Host monitor, floppy disk*
		ASM-66301	Assembler, floppy disk*
	Manual	TM-66301	User's manual for EASE66301
		AM-66301	User's manual for ASM66301 assembler
	Accessories	TCU-66301	Connection cable for user application system
		TCS-3S	Connection cable for host computer (straight through connection)
		TCS-3C	Connection cable for host computer (cross-wired connection)
		TCP-2	AC power supply cable
		TCX-2	External probe for emulation

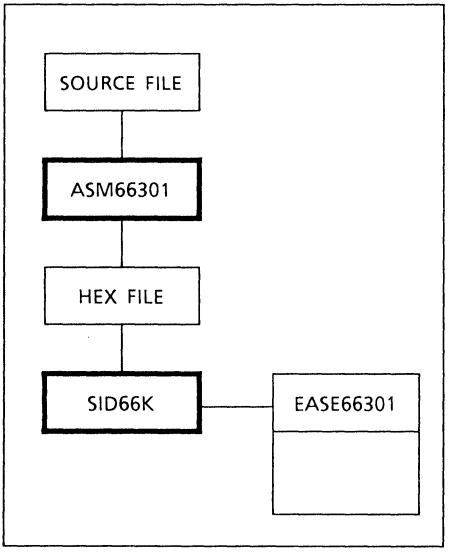
* 5.25" floppy disk for MS-DOS, if800 model 160 format
(77 track, 8 sector/track, 1024 byte/sector, 2HD)
8" floppy disk for MS-DOS, if800 model 160 format
(77 track, 26 sector/track, 128 byte/sector, 1S)
5.25" floppy disk for PC-DOS, IBM JX format
(40 tracks for each side, 8 sectors per track, 512 bytes per sector)

OUTLINE OF PROGRAM DEVELOPMENT SOFTWARE

1. ASM66301

The ASM66301 generates an INTEL HEX format file from an absolute source file written for the ASM66301 single chip micro computer. In the case of relatively small scale of program development, it offers an easy-to-understand advantage because absolute addressing is available for all the locations.

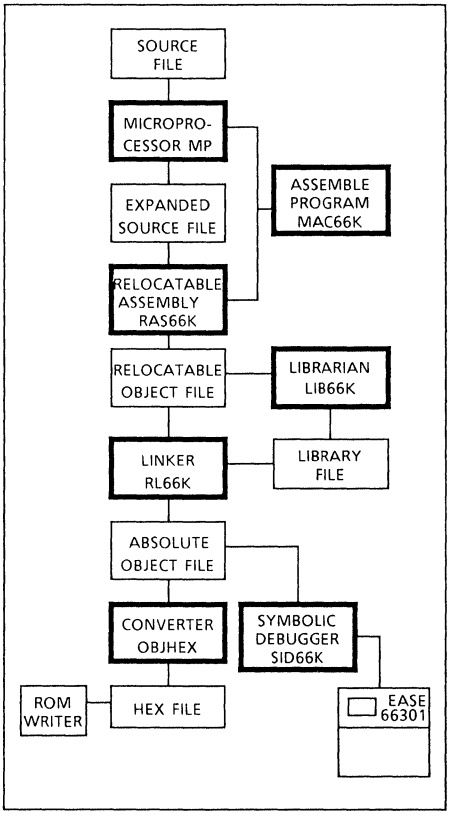
Furthermore, the ASM66301 provides the capability of generating a HEX file at a time.



2. MAC66K Software Package

The use of the MAC66K software package enables you to handle relocatable object modules. It means that you can re-use the modules, each of them has an independent function, as precious resources.

The following shows the development flow by the MAC66K software package. The softwares boxed by " ▬ " are contained in the MAC66K package.



OUTLINE OF EMULATOR

The EASE66301 emulator is connected to a host computer through a RS-232C interface, and provides effective supports for program development work with MSM66301.

1. Real Time Emulation

Emulation is performed without having to insert a wait state because of the use of an exclusive evaluation chip for MSM66301.

2. Wide User Program Area

A 64K byte RAM area (code memory) is implemented on this emulator. You can save the contents of this RAM area to a disk and also write them into EPROM. (This is available for 2764/128/256). Furthermore, the program area can be mapped in a 4K byte unit.

3. Various Break Functions

Various kinds of means to interrupt or break the program execution are available. Each can be either set or removed.

a) Breakpoint break
The breakpoint can be set at any addresses. The program stops when executing the program whose address is the same as where the breakpoints is specified.

b) Address break
The program stops when executing the program at the address specified by the emulation command (G command). You can also set a break so the program may stop after executing a specified address n times.

c) Power down break
The program stops when the evaluation chip becomes power-down status.

d) Break occurs when the trace memory is full with information.

e) Break occurs when the cycle counter overflows. The execution time of MSM66301 varies with a command executed, and therefore this emulator has the cycle counter to measure the execution time by counting the number of state.

f) Accumulator match break
Break occurs when the content of the accumulator becomes a given value.

g) RAM match break
Break occurs when the content at a specified address of RAM becomes a given value.

h) Externally forced break
Break occurs when sending a low level signal through an attached probe cable.

4. Complete Real Time Tracing Function

This emulator has two types of memories to realize the real time tracing capability which do not affect the execution time:

– Trace memory
This memory is used to trace (store) each value of command execution address, accumulator, SSP, USP, PSW, ports 0-2, and the contents of a specified RAM address and the external probe value (Up to 4096 steps)

Two different tracing functions are available:

a) Tracing is performed each time a specified address is executed.
b) Tracing is controlled by a start/stop bit.

– Flash trace memory (see note 1)
This memory is used to trace the status of the internal RAM of the evaluation chip when executing command at a specified address. But all the internal RAM area is traced at a time (up to seven times).
The traced results can be displayed on CRT even during real time emulation like done in the trace memory.

Note 1: This is now in developing stage.

Mnemonic Emulator Command

Each emulator command (debug command) of this emulator is constituted of mnemonic symbols and following parameters (address or mnemonic code).

Construction of Command Mnemonic:

a) The first symbol
Means what function to be performed.

b) The second and the latter symbols
Designates the object of the function to be performed by the emulator. The object is specified by an abbreviation of the name of register, memory or port of the EASE66301 emulation kit.

[Example]

```
* DDM 100  13F ↵
| | |  |    |  |
a b c  d    e  f
```

a - Prompt for command input of emulator
b - Instruct to display of the contents of the specified object
c - Specify Data MEMORY
d - Starting address of the contents to be displayed
e - Ending address of the contents to be displayed
f - Carriage return which means the end of command input

LIST OF DEBUGGING COMMANDS

DISK ACCESS COMMAND

1.	LOD dr:filename	Load Program into Code Memory
2.	SAV dr:filename address address	Save Program on Code Memory
3.	VER dr:filename address address	Verify File with Code Memory
4.	DIAG dr:filename	Execute command file
5.	LIST dr:filename	List console Output into Disk file
6.	NLST	End listing
7.	HELP	Display HELP File
8.	FDD dr ··· dr	Set Disk Drive
9.	DIR	Display File Directory
10.	PAUSE	Stop command file execution

EPROM WRITER COMMAND

1.	TYPE mnemonic	Set EPROM type
2.	PPR address address rom-add	Program Code Memory into EPROM
3.	TPR address address	Transfer EPROM into Code Memory
4.	VPR address address rom-add	Verify EPROM with Code Memory

ASSEMBLE COMMAND

1.	ASM address	Assemble to Code Memory
2.	DASM address address [data, data]	Disassemble to Console
3.	DBLK address address	Display Block
4.	SPB address address	Set Program Block
5.	SDB address address	Set Data Block

TRACE COMMAND

1.	DTM [−] number number	Display Trace Memory
2.	S mnemonic data [−] number	Search trace memory data
3.	DTG	Display Tigger Mode
4.	STG TR	Set trace Trigger Mode
5.	RTG	Reset Trigger Mode

EMULATION COMMAND

1.	G address parameter	real time emulation
	parameter = ,brk-address,·····, brk-address (max. 10)	
	,break-address [pass-count]	
	/ RAM ram-add [data, pass-count] st-address ed-address	
	/ mnem [data, pass-count] st-address ed-address	
	mnem = ACC, ACCH, ACCL	
2.	STP numder address [data, data]	Single Step
3.	SSF [−] mnem ··· [−] mnem	Set Step Format
	mnem = PR, LR, SFR-mnemonic	

DATA MEMORY COMMAND

1.	DDM address address [bank]	Display Data Memory
2.	CDM address [bank] [/W]	Change Data Memory
3.	FDM address address data [bank] [/W]	Fill Data Memory

CODE MEMORY COMMAND

1.	DCM address address	Display Code Memory
2.	CCM address [/W/]	Change Code Memory
3.	FCM address address data [/W]	Fill Code Memory

SFR COMMAND

1.	Dsfr-mnem	Display sfr mnemonic
2.	Csfr-mnem	Cange sfr mnemonic

REGISTER COMMAND

1.	DCR	Display Control Register
2.	CCR	Change Control Register
3.	DPR bank	Display Pointing Register
4.	CPR bank	Change Pointing Register
5.	DX1 bank	Display X1
6.	CX1 bank	Change X1
7.	DX2 bank	Display X2
8.	CX2 bank	Change X2
9.	DDP bank	Display DP
10.	CDP bank	Change DP
11.	DUSP bank	Display USP
12.	CUSP bank	Change USP
13.	DPC	Display Program Counter
14.	CPC data	Change Program Counter
15.	DLR bank	Display Local Register
16.	CLR bank	Change Local Register
17.	DREG	Display Register
18.	CREG element	Change Register

ATTRIBUTE MEMORY COMMAND

1.	DIE address address	Display Instruction Executed Bit Memory
2.	RIE	Reset Instruction Executed Bit Memory
3.	DBP address address	Display Break Point Bit Memory
4.	EBP address··· address	Enable Break Point Bit
5.	RBP address··· address	Reset Break Point Bit
6.	FBP address address data	Fill Break Point Bit Memory
7.	DTR address address	Display Trace Enable Bit Memory
8.	ETR address··· address	Enable Trace Enable Bit
9.	RTR address··· address	Reset Trace Enable Bit
10.	FTR address address data	Fill Trace Enable Bit Memory
11.	DSO address address	Display Sync Out Enable Bit Memory
12.	ESO address··· address	Enable Sync Out Enable Bit
13.	RSO address··· address	Reset Sync Out Enable Bit
14.	FSO address address data	Fill Sync Out Enable Bit Memory
15.	DWB address address	Display Word Byte flag Bit Memory
16.	SWD address··· address	Set Word Data Bit
17.	SBD address··· address	Set Byte Bit
18.	FWB address address data	Fill Word Byte Bit Memory
19.	DSTK address address	Display Stack flag Bit Memory
20.	ESTK address··· address	Enable Stack flag Bit
21.	RSTK address··· address	Reset Stack flag Bit
22.	FSTK address address data	Fill Stack flagBit Memory

SFR BUFFER COMMAND

1.	DBUF	Display Bufer
2.	TBUF	Transefer SFR data into Buffer
3.	LBUF	Load Bussfer data into SFR
4.	RBUF	Reset Buffer

CYCLE COUNTER COMMAND

1.	STG CC	Set CC Triger
2.	DCC	Display Cycle Counter
3.	CCC data	Change Cycle Counter
4.	TIME data	Set 1 cycle time

BREAK CONDITION, STATUS COMMAND

1.	DBC	Display Break Condition
2.	SBC |−| mnemonic··· | -| mnemonic	Set Break Condition

OTHER COMMAND

1.	MAP add	Mapping Eva-chip and memory bus
2.	RES |E|	Reset System OR Eva-chip
3.	EXIT	Return Host Computer's OS
4.	MAC	Define command execution
5.	M	Execution define command
6.	@	Repeat last command
7.	SIO F or H	Set Emulator to I O terminal mode
8.	DEV	CH2 Device Configuration

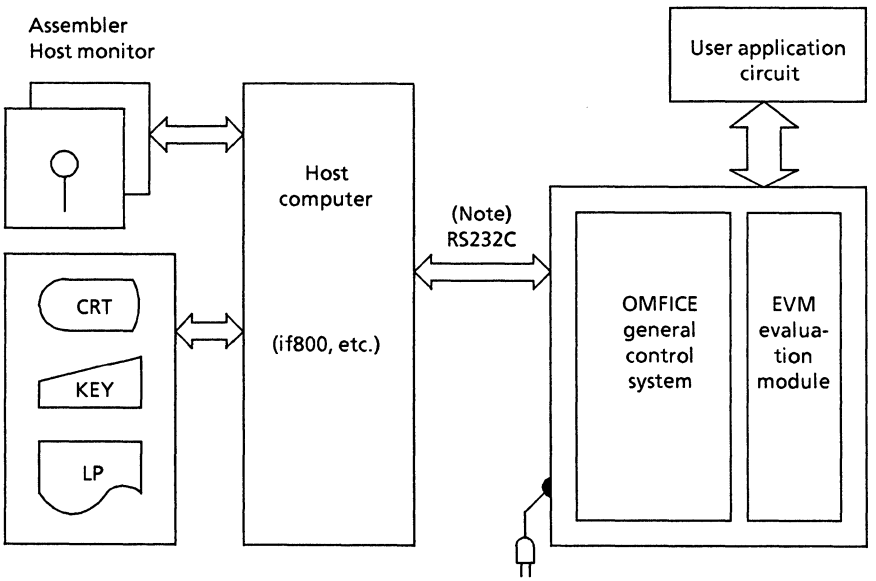
OMFICE Program Development Support System

■ GENERAL

The OMFICE program development support system is a high-performance, general-purpose system that has been created to allows quick and efficient development and evaluation of programs for all types of OKI 1-chip microcontrollers.

■ SYSTEM CONFIGURATION

This system consists of a high-performance program evaluation emulator (OMFICE general control system and interchangeable EVM evaluation module), as well as a host monitor program and assemblers (absolute and relocatable) that run on the host computer (MS-DOS, PC-DOS, UNIX). On-line connection of the program evaluation emulator with a computer equipped with an RS-232C interface provides everything from source program assembly to program evaluation and debugging.



Note: GP-IB interface is under development.

■ AVAILABLE CONFIGURATIONS

This system is available from OKI in one of the following configurations.

1. OMFICE xxxxx (xxxxx stands for the name of the 1-chip microcontroller for each EVM evaluation module) program development support system, including OMFICE general control system and EVM evaluation modules, host monitor program and assembler.
2. EVM evaluation modules, host monitor program and assembler for connection with user's OMFICE general control system to form an OMFICE xxxxx program development support system.
3. OMFICE general control system only.

■ OMFICE PROGRAM DEVELOPMENT SUPPORT SYSTEM INTRODUCTION

1. Overview of OMFICE General Control System

   This system incorporates the portions of a program development support system that are not affected by a change in the type of 1-chip microcontroller for which a program is being evaluated. This means that a single OMFICE general control system makes it possible to perform development for a variety of 1-chip microcontrollers by simply supplying the applicable EVM evaluation module.

2. EVM Evaluation Modules

   EVM evaluation modules are provided for each types of 1-chip microcontrollers. Installation of these modules on the connector of the OMFICE general control system provides program development capabilities for the applicable 1-chip microcontrollers.

The following are the types of EVMs available:

- EVM 67620 Evaluation Module (under development)
- EVM66417 Evaluation Module (under development)
- EVM65524 Evaluation Module (under development)

3. Host Monitor

   This software runs on the host computer to analyze command lines entered by the operator, provide the emulator with the operations required for processing commands, edit the data sent from the emulator as the result of a command, and display it on the CRT.

■ OMFICE PROGRAM DEVELOP-MENT SYSTEM FUNCTIONS

The OMFICE program development system includes the following functions to provide the most efficient development environment for OKI 1-chip microcontrollers.

1. Versatile break functions

   The following methods are available for interrupting (breaking) program execution.

a) Break point break
b) Address break
c) Power down break
d) Trace memory overflow break
e) Cycle counter overflow break
f) Accumulator match break
g) Data match break
h) External forced break
i) 3-point address passing break
j) Flash tarce memory overflow break

2. Versatile trace functions
The following two types of real time trace functions, which have no affect on program execution, are supported.

- Trace Memory
This memory is used during program execution to trace values such as: execution address, accumulator, port, program status word, etc.
a) Up to 8K word step trace
b) Execution address can be specified using trace enable bit
c) Trace execution address range can be specified using trace start/stop bit
d) Trace start/trace end trigger can be specified according to trace contents
e) Delay counts up to 256 steps can be specified for the trace end trigger
f) Trace details can be displayed during real time emulation

- Flash Trace Memory
This memory is used for batch tracing (up to 1K word) of RAM and SFR status during execution of the command at a spepcific address. As with trace memory, flash trace memory contents can be displayed on the CRT during real time emulation.
(Note)The specific operation of the functions described above may differ slightly according to the EVM evaluation module being used.

EVM67620 Evaluation Module

■ APPLICABLE MODEL

Installation of EVM67620 evaluation module on the OMFICE general control system provides development and evaluation capabilities for the OKI CMOS 16-bit, 1-chip microcontroller. The MSM67620 was specially developed to incorporate a wide range of original OKI concepts and technology to make program development quick and efficient.

■ MODULE OVERVIEW

Installation of EVM67620 evaluation module on the OMFICE general control system makes it possible to perform comprehensive debugging operations, from debugging without a complete user circuit to debugging of the complete system.

■ MODULE FUNCTIONS

- Real time emulation
- Measurement functions available during execution
- Versatile breaks
- Real time trace
- Other

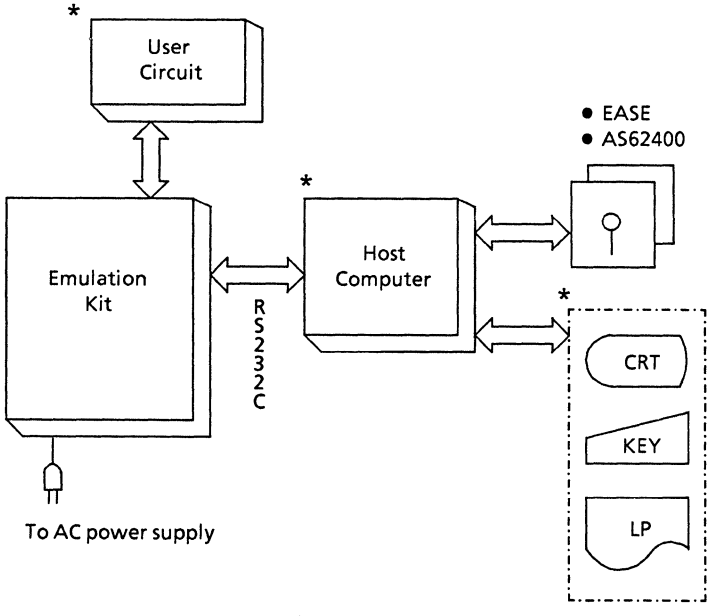
EVM62408 Evaluation Module

■ APPLICABLE MODEL

The OMFICE 62408 program development support system makes it possible to perform program development and evaluation of the OKI MSC62408 CMOS 4-bit, 1-chip microcontroller. The OMFICE62408 was specially developed as a high-performance system that incorporates a wide range of original OKI concepts and technology to make program development quick and efficient.

■ SYSTEM CONFIGURATION

The system consists of the AS62400 cross assembler — a powerful cross assembler based on a high-performance program evaluation emulator (EASE host monitor, emulation kit, and host MS-DOS computer). On-line connection of the emulation kit with an MS-DOS computer (such as an if800) equipped with an RS-232C interface provides comprehensive capabilities, from source program assembly to program evaluation and debugging.



(Note 1) MS-DOS is a registered trademark of the Microsoft Corporation of the United States.
(Note 2) Asterisks indicate user side equipment.

■ OMFICE 62408 EMULATOR FEATURES

Code memory	32,768 x 8 bits
Data memory	368 x 8 bits
Attribute memory	32,768 x 13 bits
Instruction-executed memory	32, 768 x 1 bit
Trace memory	8,192 x 97 bits
Trace types	① Address trace AR, BR, CCR, SP, P0, P1, P2, ② Trigger trace P3, P4, P5, EX (probe)
Cycle counter	32-bit binary counter
User program exeuction procedure	User program loaded into code memory and executed with GO or STP command.
Break types	① Address breaker ② ESC key break ③ Normal trace pointer overflow break ④ Cycle counter overflow break ⑤ RAM data match break ⑥ STP mode break
EPROM programmer	High-speed writing for 2764, 27128, 27256, 27512 and equivalents.
RS-232C port	① CH1 port Connection port for host machine that inputs the emulator command. ② CH2 port Connection port for list device and personal computer.

Oki Program Development Tool Software

■ NEC PC9801

Model	OS	FDD (Inches)	Remarks
EASE80C49-PC9801	MS-DOS Ver.2.11	5.26 (2DD, 2HD) 8 (1S-1D)	EASE49-PC9801/ASM49-PC9801 for EASE80C49
EASE6400-PC9801		"	EASE-PC9801/ASM6400-PC9801 for EASE6400
EASE6458-PC9801		"	EASE-PC9801/ASM6458-PC9801 for EASE6458
EASE6502-PC9801		"	EASE65-PC9801/ASM6502-PC9801 for EASE6502
EASE80C51mKII-PC9801		"	EASE-PC9801/ASM51-PC9801 for EASE80C51
MAC51-PC9801		"	Macro-assembler/symbolic debugger for EASE80C51
EASE5052/56-PC9801		"	EASE65-PC9801/ASM50-PC9801 for EASE5052/56
EASE5054/55-PC9801		"	EASE65-PC9801/ASM50-PC9801 for EASE5054/55
EASE6052-PC9801		"	EASE65-PC9801/ASM50-PC9801 for EASE6052

Notes:
Includes RS-232C cable for the PC9801.
MAC51 forms set with RL51, LIB51, SID51.
* Under development

■ IBM PC-XT and PC-AT

Model	OS	FDD (Inches)	Remarks
EASE80C49-The PC	IBM MS-DOS Ver.2.11	5.25 (2HD)	EASE49-The PC/ASM49-The PC for EASE80C49
EASE6400-The PC		"	EASE6400-The PC/ASM6400-The PC for EASE6400
EASE6458-The PC		"	EASE-The PC/ASM6458-The PC for EASE6458
EASE6502-The PC		"	EASE65-The PC/ASM6502-The PC for EASE6502
EASE80C51mKII-The PC		"	EASE-The PC/ASM51-The PC for EASE80C51
MAC51-The PC		"	Macro-assembler/symbolic debugger for EASE80C51
EASE66301-The PC		"	EASE-The PC/ASM66301-The PC for EASE66301
EASE62580-The PC		"	EASE-The PC/AS62580-The PC for EASE62580
EASE5052/56-The PC		"	EASE-The PC/ASM50-The PC for EASE5052/56
EASE5054/55-The PC		"	EASE-The PC/ASM50-The PC for EASE5054/55
EASE6052-The PC		"	EASE-The PC/ASM50-The PC for EASE6052
EAES6051-The PC		"	EASE-The PC/ASM50-The PC for EASE6051
EASE6352/6052-The PC		"	EASE-The PC/ASM50-The PC for EASE6352/6052
EASE6351/53-The PC		"	EASE-The PC/ASM50-The PC for EASE6351/53

Notes:
Includes RS-232C cable for the PC9801.
MAC51 forms set with RL51, LIB51, SID51.
* Under development

■ if800 model 50

Model	OS	FDD (Inches)	Remarks
EASE80C49-model 50	if800 MS-DOS Ver.2.11	8 (1S-1D)	EASE49-model 50/ASM49-model 50 for EASE80C49
EASE6400-model 50		"	EASE-model 50/ASM6400-model 50 for EASE6400
EASE6458-model 50		"	EASE-model 50/ASM6458-model 50 for EASE6458
EASE6502-model 50		"	EASE65-model 50/ASM6502-model 50 for EASE6502
EASE80C51mKII-model 50		"	EASE-model 50/ASM51-model 50 for EASE50C51
MAC51-model 50		"	Macro-assembler/symbolic debugger for EASE80C51
EASE5052/56-model 50		"	EASE-model 50/ASM50-model 50 for EASE5052/56
EASE5054/55-model 50		"	EASE-model 50/ASM50-model 50 for EASE5054/55
EASE5052-model 50		"	EASE-model 50/ASM50-model 50 for EASE6052

Notes:
Cable for if800 model 30 emulator can be used for RS-232C cable.
MAC51 forms set with RL51 and LIB51.
MS-DOS version included as standard with EASE62580, 63301, 66301, EASE6351/53, EASE6352/6052, EASE6051.
* Under development

■ if800 model 60/RX110, 120/EX110, 120

Model	OS	FDD (Inches)	Remarks
EASE80C49-model 60	if800 MS-DOS Ver.2.11	5.25 (2HD)	EASE49-model 60/ASM49-model 60 for EASE80C49
EASE6400-model 60		"	EASE-model 60/ASM6400-model 60 for EASE6400
EASE6458-model 60		"	EASE-model 60/ASM6458-model 60 for EASE6458
EASE6502-model 60		"	EASE65-model 60/ASM6502-model 60 for EASE6502
EASE80C51mKII-model 60		"	EASE-model 60/ASM51-model 60 for EASE80C51
MAC51-model 60		"	Macro-assembler/symbolic debugger for EASE80C51
MAC66K-model 60		"	Macro-assembler/symbolic debugger for EAES66301
EASE5052/56-model 60		"	EASE-model 60/ASM50-model 60 for EASE5052/56
EASE5054/55-model 60		"	EASE-model 60/ASM50-model 60 for EASE5054/55
EASE6052-model 60		"	EASE-moldel 60/ASM50-model 60 for EASE 6052

Notes:
RS-232C straight cable included.
MAC51 forms set with LIB51.
MS-DOS version included as standard with EASE62580, 63310, 66301, EASE6351/53, EASE6352/6052, EASE6051.
* Under development

MEMO

NOTICE

1. The information contained herein can change without notice owing to product and/or technical improvements. Please make sure before using the product that the information you are referring to is up-to-dated.

2. The outline of action and examples of application circuits described herein have been chosen as an explanation of the standard action and performance of the product. When you actually plan to use the product, please ensure that the outside conditions are reflected in the actual circuit and assembling designs.

3. NO RESPONSIBILITY IS ASSUMED BY US FOR ANY CONSEQUENCE RESULTING FROM ANY WRONG OR IMPROPER USE OR OPERATION, ETC. OF THE PRODUCT.

4. Neither indemnity against nor license of a third party's industrial and intellectual property right, etc. is granted by us in connection with the use of the product and/or the information and drawings contained herein. No responsibility is assumed by us for any infringement of a third party's right which may result from the use thereof.

5. Certain product described herein falls within the category of strategical goods, etc. under the Foreign Exchange and Foreign Trade Control Law. Accordingly, before exporting such product or any part thereof, you are required under the Law to file the application for the export license by the Japanese Government.

6. No part of the contents contained herein may be reprinted or reproduced without our prior permission.

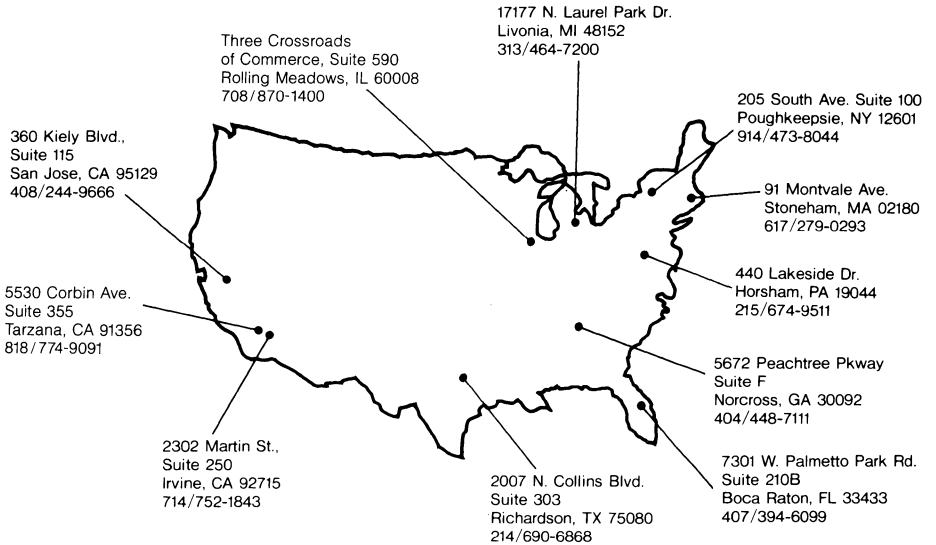
7. MS-DOS is a registered trademark of Microsoft Corporation.
   PC-DOS is a registered trademark of IBM Corporation.
   CP/M-80 is a resistered trademark of Digital Research Corporation.

OKI

Oki Semiconductor

785 North Mary Avenue, Sunnyvale, CA 94086, U.S.A. Tel: 408-720-1900 Fax: 408-720-1918

For further information please contact:



Oki Electric Industry Co., Ltd.
Head Office Annex
7-5-25 Nishishinjuku,
Shinjuku-ku, Tokyo 160, Japan
Tel: 3-5386-8100
Fax: 3-5386-8110
Telex: J27662 OKIDENED
Electronic Devices Group
Overseas Marketing Dept.

Oki Semiconductor
785 North Mary Avenue.
Sunnyvale, CA 94086, U.S.A.
Tel: 408-720-1900
Fax: 408-720-1918
Telex: 296687 OKI SNTA

Oki Electric Europe GmbH
Hellersbergstraße 2.
D-4040 Neuss
Fed. Rep. of Germany
Tel: 2101-15960
Fax: 2101-103539
Telex: 8517427 OKI D

Oki Electronics (Hong Kong) Ltd.
Suite 1801-4, Tower 1,
China Hong Kong City,
33 Canton Road, Tsim Sha Tsui,
Kowloon, Hong Kong
Tel: 3-7362336
Fax: 3-7362395
Telex: 45999 OKIHK HX

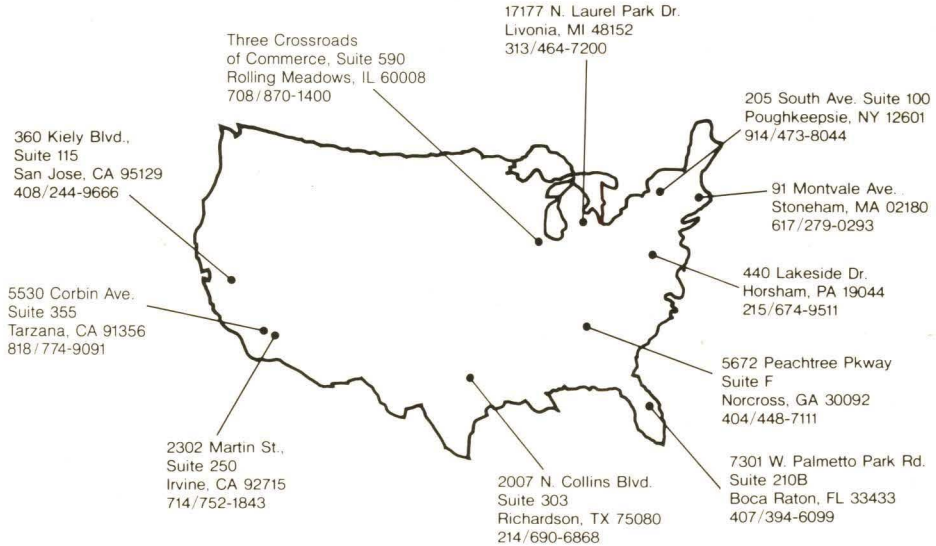
SINCE 1881

REF NO.: E3S010932

Oki Semiconductor

785 North Mary Avenue, Sunnyvale, CA 94086, U.S.A. Tel: 408-720-1900 Fax: 408-720-1918

For further information please contact:



Oki Electric Industry Co., Ltd.
Head Office Annex
7-5-25 Nishishinjuku,
Shinjuku-ku, Tokyo 160, Japan
Tel: 3-5386-8100
Fax: 3-5386-8110
Telex: J27662 OKIDENED
Electronic Devices Group
Overseas Marketing Dept.

Oki Semiconductor
785 North Mary Avenue,
Sunnyvale, CA 94086, U.S.A.
Tel: 408-720-1900
Fax: 408-720-1918
Telex: 296687 OKI SNTA

Oki Electric Europe GmbH
Hellersbergstraße 2
D-4040 Neuss
Fed. Rep. of Germany
Tel: 2101-15960
Fax: 2101-103539
Telex: 8517427 OKI D

Oki Electronics (Hong Kong) Ltd.
Suite 1801-4, Tower 1,
China Hong Kong City,
33 Canton Road, Tsim Sha Tsui,
Kowloon, Hong Kong
Tel: 3-7362336
Fax: 3-7362395
Telex: 45999 OKIHK HX

SINCE 1881

REF NO.: E3S010932