SERVICE and ACCESSORIES

United States customers can obtain service and repair assistance from Intel by contacting the MCD Technical Support Center in Santa Clara, California at one of the following numbers:

Telephone:

From Alaska or Hawaii call —
(408) 987-7187

From locations within California call toll free —
(800) 672-3507

From all other U.S. locations call toll free —
(800) 538-8014

Customers outside of the United States should contact their sales source (Intel Sales Office or Authorized Intel Distributor) for directions on obtaining service or repair assistance.

PROMT-SPP	Connects Prompt to Intellec Microcomputer Development System as Specialized PROM Programming. Permits direct downloading of programs for debugging.
PROMT-SER	Connects Prompt to teletypewriter or serial terminal.
PROMT-475	Adaptor socket allows Prompt 48 to program 8755.

Additional PROMPT 48 Programming Pads (98-401), User's Manuals (98-402) and Reference Cardlets (98-404) may be ordered from the Intel Literature Department.

I/O PORTS and BUS CONNECTOR

Signal Name	Pin No.	Buffer Characteristic
BUS 0	17	3-State Bidirectional
1	21	3-State Bidirectional
2	25	3-State Bidirectional
3	29	3-State Bidirectional
4	31	3-State Bidirectional
5	27	3-State Bidirectional
6	23	3-State Bidirectional
7	19	3-State Bidirectional
PORT 10	18	Chip (No Buffer)
1	20	Chip (No Buffer)
2	22	Chip (No Buffer)
3	24	Chip (No Buffer)
4	26	Chip (No Buffer)
5	28	Chip (No Buffer)
6	30	Chip (No Buffer)
7	32	Chip (No Buffer)
PORT 20	7	3-State Mapped Bidirectional 100Ω Series Resistor
1	5	3-State Mapped Bidirectional 100Ω Series Resistor

Signal Name	Pin No.	Buffer Characteristic
PORT 2	3	3-State Mapped Bidirectional 100Ω Series Resistor
3	1	3-State Mapped Bidirectional 100Ω Series Resistor
4	4	Chip (No Buffer)
5	6	Chip (No Buffer)
6	8	Chip (No Buffer)
7	10	Chip (No Buffer)
ALE	13	TTL Output (10 LS Loads)
T0	14	Chip Input/Output Clock 2.2K Pull-Up
T1	12	Chip Input 2.2K Pull-Up
INT/	49	TTL Schmitt Input, 2.2K Pull-Up, Monitor Gated
PSEN/	15	TTL Output (10 LS Loads)
RD/	9	TTL Output (10 LS Loads)
WR/	11	TTL Output (10 LS Loads)
POWR/	33	TTL Output (5 LS Loads)
PROG/	2	Chip (No Buffer)
RESET/	16	System Reset Overrides Chip Input/System Output 2.2K Pull-Up
GND	45,46 47,48	

Use header puller to remove I/O PORTS and BUS CONNECTOR

SPP 48 and ASM 48

disk-based only:

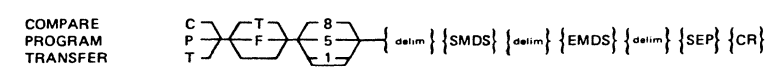
SPP48 (fname) {CR} Downloads ISIS hex file to Prompt

READ

R {space} (fname) {CR} Reads ISIS hex file into development system

R {CR} Reads hex file from system reader into development system

disk or paper tape-based only:



abbreviations:

{CR}	carriage return	(fname)	file name (hex file)
{delim}	delimiters — spaces or comma and spaces	{SEP}	starting EPROM address, hex
{EMDS}	ending MDS address, hex	{SPROMPT}	starting PROMPT address, hex
{EPROMPT}	ending PROMPT address, hex	{SMDS}	starting MDS address, hex

ASM48 OPERATORS

(,)	ORG	Origin
*,/, MOD, SHL, SHR	END	End program
+, –	DB	Define Byte
EQ, LT, LE, GT, GE, NE	DS	Define Storage
NOT	DW	Define Word
AND	IF ELSE	Conditional Assembly
OR, XOR	ENDIF	End Conditional
HIGH, LOW	EQU Set	Set Symbol Value

ASM48 DIRECTIVES

MACRO	Define Macro
ENDM	End Macro
LOCAL	Define Local Symbol
REPT	Define Repeat Block
IRP	Indefinite Repeat
IRPC	Indefinite Repeat Character
EXITM	Alternate Macro Exit

HEX/ASCII

00	NUL		20	SP		40	@		60		
01	SOH		21	!		41	A		61	a	
02	STX		22	"		42	B		62	b	
03	ETX		23	#		43	C		63	c	
04	EOT		24	$		44	D		64	d	
05	ENQ		25	%		45	E		65	e	
06	ACK		26	&		46	F		66	f	
07	BEL		27	'		47	G		67	g	
08	BS		28	(		48	H		68	h	
09	HT		29	)		49	I		69	i	
0A	LF		2A	*		4A	J		6A	j	
0B	VT		2B	+		4B	K		6B	k	
0C	FF		2C	,		4C	L		6C	l	
0D	CR		2D	—		4D	M		6D	m	
0E	SO		2E	.		4E	N		6E	n	
0F	SI		2F	/		4F	O		6F	o	
10	DLE		30	0		50	P		70	p	
11	DC1	(X-ON)	31	1		51	Q		71	q	
12	DC2	(TAPE)	32	2		52	R		72	r	
13	DC3	(X-OFF)	33	3		53	S		73	s	
14	DC4		34	4		54	T		74	t	
15	NAK		35	5		55	U		75	u	
16	SYN		36	6		56	V		76	v	
17	ETB		37	7		57	W		77	w	
18	CAN		38	8		58	X		78	x	
19	EM		39	9		59	Y		79	y	
1A	SUB		3A	:		5A	Z		7A	z	
1B	ESC		3B	;		5B	[		7B	{	
1C	FS		3C	<		5C	\		7C	|	
1D	GS		3D	=		5D	]		7D	}	(ALT MODE)
1E	RS		3E	>		5E	^	(↑)	7E	~	
1F	US		3F	?		5F	—	(←)	7F	DEL	(RUB OUT)

MCS-48 INSTRUCTIONS

Hex	Mnemonic	Description	Symbolic	Flags C	Flags AC	Cycles
D0	XRL A, @R0	Exclusive Or Indirect to A	(A) ← (A) XOR ((R0))	–	–	1
D1	XRL A, @R1		(A) ← (A) XOR ((R1))	–	–	1
D3 □□	XRL A, #data	Exclusive Or immediate to A	(A) ← (A) XOR data	–	–	2

MCS-48 INSTRUCTIONS

Hex	Mnemonic	Description	Symbolic	Flags C	Flags AC	Cycles
67	RRC A	Rotate A right, through carry	$(A_n) \leftarrow (A_{n+1})$ n = 0–6 $(A_7) \leftarrow (C)$; $(C) \leftarrow (A_0)$	X	–	1
E5	SEL MB0	Select Memory Bank 0 PC_{11} is set to 0 on next jump unconditional	$(DBF) \leftarrow 0$	–	–	1
F5	SEL MB1	Select Memory Bank 1 PC_{11} is set to 1 on next jump unconditional	$(DBF) \leftarrow 1$	–	–	1
C5	SEL RB0	Select Register Bank 0	$(PSW_4{=}BS) \leftarrow 0$	–	–	1
D5	SEL RB1	Select Register Bank 1	$(PSW_4{=}BS) \leftarrow 1$	–	–	1
65	STOP TCNT	Stop Timer/Counter		–	–	1
45	STRT CNT	Start Counter		–	–	1
55	STRT T	Start Timer		–	–	1
47	SWAP A	Swap Nibbles Within A	$(A_{7-4}) \leftrightarrow (A_{3-0})$	–	–	1
2_	XCH A, R0–7	Exchange A with R	$(A) \leftrightarrow (R)$	–	–	1
	28 XCH A, R0 29 XCH A, R1	2A XCH A, R2 2B XCH A, R3	2C XCH A, R4 2D XCH A, R5	2E XCH A, R6 2F XCH A, R7		
20	XCH A, @R0	Exchange A with R Indirect	$(A) \leftrightarrow ((R0))$	–	–	1
21	XCH A, @R1		$(A) \leftrightarrow ((R1))$	–	–	1
30	XCHD A, @R0	Exchange Digit (Nibble) with R Indirect	$(A_{3-0}) \leftrightarrow ((R0)_{3-0})$	–	–	1
31	XCHD A, @R1		$(A_{3-0}) \leftrightarrow ((R1)_{3-0})$	–	–	1
D_	XRL A, R0–7	Exclusive Or Register to A	$(A) \leftarrow (A)$ XOR (R)	–	–	1
	D8 XRL A, R0 D9 XRL A, R1	DA XRL A, R2 DB XRL A, R3	DC XRL A, R4 DD XRL A, R5	DE XRL A, R6 DF XRL A, R7		

MCS-48 INSTRUCTIONS

Hex	Menomic	Description	Symbolic	Flags C	Flags AC	Cycles
90	MOVX @R0, A	Move from external data memory to A	((R0)) ← (A)	–	–	2
91	MOVX @R1, A		((R1)) ← (A)	–	–	2
00	NOP	No Operation		–	–	2
4_	ORL A, R0–7	Or register to A	(A) ← (A) OR (R)	–	–	1
	48 ORL A, R0	4A ORL A, R2	4C ORL A, R4	4E ORL A, R6		
	49 ORL A, R1	4B ORL A, R3	4D ORL A, R5	4F ORL A, R7		
40	ORL A, @R0	Or register indirect to A	(A) ← (A) OR ((R0))	–	–	1
41	ORL A, @R1		(A) ← (A) OR ((R1))	–	–	1
43 □□	ORL A, #data	Or immediate to A	(A) ← (A) OR data	–	–	2
8_ □□	ORL P12BUS, #data	Or immediate to P1, P2 or BUS	(P12BUS) ← (P12BUS) OR data	–	–	2
	88 □□ ORL BUS, #data	89 □□ ORL, P1, #data	8A □□ ORL P2, #data			
8_	ORLD P4–7, A	Or A to expanDer port	(P4–7) ← (P4–7) OR (A_{3-0})	–	–	2
	8C ORL P4, A	8D ORL P5, A	8E ORL P6, A	8F ORL P7, A		
__	OUTL BUSP12, A	Output A to BUS, P1 or P2	(BUSP12) ← (A)	–	–	2
	02 OUTL BUS, A	39 OUTL P1, A	3A OUTL P2, A			
83	RET	Return without PSW restore	(SP) ← (SP) – 1; (PC) ← ((SP))	–	–	2
93	RETR	Return and restore PSW	(SP) ← (SP) – 1; (PC) ← ((SP))	–	–	2
E7	RL A	Rotate A left, without carry	(A_{n+1}) ← (A_n) n = 0–6 (A_6) ← (A_7)	–	–	1
F7	RLC A	Rotate A left, through carry	(A_{n+1}) ← (A_n) n = 0–6 (A_0) ← (C); (C) ← (A_7)	X	–	1
77	RR A	Rotate A right, wihtout carry	(A_n) ← (A_{n+1}) n = 0–6 (A_7) ← (A_0)	–	–	1

MCS-48 INSTRUCTIONS

Hex	Mnemonic	Description	Symbolic	Flags C	Flags AC	Cycles
F0	MOV A, @R0	Move indirect to A	(A) ← ((R0))	–	–	1
F1	MOV A, @R1		(A) ← ((R1))	–	–	1
42	MOV A, T	Read Timer/Counter	(A) ← (T)	–	–	1
D7	MOV PSW, A	Move PSW to A	(A) ← (PSW)	X	X	1
A_	MOV R0–7, A	Move A to R	(R) ← (A)	–	–	1
	A8 MOV R0, A A9 MOV R1, A	AA MOV R2, A AB MOV R3, A	AC MOV R4, A AD MOV R5, A	AE MOV R6, A AF MOV R7, A		
B_ □□	MOV R0–7, #data	Move immediate to register	(R) ← #data	–	–	2
	B8 □□ MOV R1, # data B9 □□ MOV R1, #data	BA □□ MOV R2, #data BB □□ MOV R3, #data	BC □□ MOV R4, #data BD □□ MOV R5, #data	BE □□ MOV R6, #data BF □□ MOV R7, #data		
A0	MOV @R0, A	Move A to R indirect	((R0)) ← (A)	–	–	1
A1	MOV @R1, A		((R1)) ← (A)	–	–	1
B0 □□	MOV @R0, #data	Move immediate data to R indirect	((R0)) ← #data	–	–	2
B1 □□	MOV @R1, #data		((R1)) ← #data	–	–	2
62	MOV T, A	Load Timer/Counter	(T) ← (A)	–	–	1
0_	MOVD A, P4–7	Input expanDer Port to A	(A) ← (P4–7)	–	–	2
	0C MOVD A, P4	0D MOVD A, P5	0E MOVD A, P6	0F MOVD A, P7		
3_	MOVD P4–7, A	Output A to expanDer Port	(P4–7) ← (A)	–	–	2
	3C MOVD P4, A	3D MOVD P5, A	3E MOVD P6, A	3F MOVD P7, A		
A3	MOVP A, @A	Move to A from current page	(PC_{0-7}) ← (A); (A) ← ((PC))	–	–	2
E3	MOVP3 A, @A	Move to A from program memory page 3.	(A) ← (3(A))	–	–	2
80	MOVX A, @R0	Move to A from external data memory	(A) ← ((R0))	–	–	2
81	MOVX A, @R1		(A) ← ((R1))	–	–	2

MCS-48 INSTRUCTIONS

Hex	Mnemonic	Description	Symbolic	Flags C	Flags AC	Cycles
_4 □□	JMP _addr	Jump unconditional	$(PC_{10-0}) \leftarrow$ addr; $(PC_{11}) \leftarrow$ (DBF)	–	–	2
	04 □□ JMP0 addr 24 □□ JMP1 addr	44 □□ JMP2 addr 64 □□ JMP3 addr	84 □□ JMP4 addr A4 □□ JMP5 addr	C4 □□ JMP6 addr E4 □□ JMP7 addr		
B3	JMPP @A	Jump Indirect Within Page	$(PC_{7-0}) \leftarrow ((A))$	–	–	2
		Program memory contents addressed by A become PC_{7-10}.				
E6 □□	JNC addr	Jump if Not Carry	IF (C) = 0 THEN $(PC_{7-0}) \leftarrow$ addr	–	–	2
86 □□	JNI addr	Jump if Not Interrupt Input	IF (I) = 0 THEN $(PC_{7-0}) \leftarrow$ addr	–	–	2
		Interrupt input is asserted low, so jumps on interrupt signal.				
26 □□	JNT0 addr	Jump if Not Test 0	IF (T0) = 0 THEN $(PC_{7-0}) \leftarrow$ addr	–	–	2
46 □□	JNT1 addr	Jump if Not Test 1	IF (T1) = 0 THEN $(PC_{7-0}) \leftarrow$ addr	–	–	2
96 □□	JNZ addr	Jump if A Not Zero	IF (A) ≠ 0 THEN $(PC_{7-0}) \leftarrow$ addr	–	–	2
16 □□	JTF addr	Jump if Timer Flag	IF (TF) = 1 THEN $(PC_{7-0}) \leftarrow$ addr	–	–	2
36 □□	JT0 addr	Jump if Test 0	IF (T0) = 1 THEN $(PC_{7-0}) \leftarrow$ addr	–	–	2
56 □□	JT1 addr	Jump if Test 1	IF (T1) = 1 THEN $(PC_{7-0}) \leftarrow$ addr	–	–	2
C6 □□	JZ addr	Jump if A Zero	IF (A) = 0 THEN $(PC_{7-0}) \leftarrow$ addr	–	–	2
23 □□	MOV A, #data	Move immediate to A	(A) ← data	–	–	2
C7	MOV A, PSW	Move PSW to A	(A) ← (PSW)	–	–	1
F_	MOV A, R0–7	Move Register to A	(A) ← (R)	–	–	1
	F8 MOV A, R0 F9 MOV A, R1	FA MOV A, R2 FB MOV A, R3	FC MOV A, R4 FD MOV A, R5	FE MOV A, R6 FF MOV A, R7		

MCS-48 INSTRUCTIONS

Hex	Mnemonic	Description	Symbolic	Flags C	Flags AC	Cycles
E_ □□	DJNZ R0–7, addr	Decrement Register, Jump Not Zero	(R) ← (R) – 1; IF (R) ≠ 0 THEN (PC_{7-0}) ← addr	–	–	2
	E8 □□ DJNZ R0, addr E9 □□ DJNZ R1, addr	EA □□ DJNZ R2, addr EB □□ DJNZ R3, addr	EC □□ DJNZ R4, addr ED □□ DJNZ R5, addr	EE □□ DJNZ R5, addr EF □□ DJNZ R7, addr		
	Note: DJNZ at XFF jumps to X+1 addr (next page) if taken.					
05	EN I	Enable external Interrupt		–	–	1
25	EN TCNTI	Enable Timer/Counter Interrupt		–	–	1
75	ENT0 CLK	Enable Clock Output to T0 pin.	Cleared by SYS RST	–	–	1
0_	IN(S) A, BUSP12	Input Bus, P1 or P2 to A	(A) ← (BUSP12)	–	–	2
	08 INS A, BUS	09 IN A, P1	0A IN A, P2			
17	INC A	Increment A	(A) ← (A) + 1	–	–	1
1_	INC R0–7	Increment register	(R) ← (R) + 1	–	–	1
	18 INC R0 19 INC R1	1A INC R2 1B INC R3	1C INC R4 1D INC R5	1E INC R6 1F INC R7		
10	INC @R0	Increment indirect	((R0)) ← ((R0)) + 1	–	–	1
11	INC @R1		((R1)) ← ((R1)) + 1	–	–	1
_2 □□	JBb addr	Jump if accumulator Bit set	IF (A_b) = 1 THEN (PC_{7-0}) ← addr	–	–	2
	12 □□ JB0 addr 32 □□ JB1 addr	52 □□ JB2 addr 72 □□ JB3 addr	92 □□ JB4 addr B2 □□ JB5 addr	D2 □□ JB6 addr F2 □□ JB7 addr		
F6 □□	JC addr	Jump if Carry	IF (C) = 1 THEN (PC_{7-0}) ← addr	–	–	2
B6 □□	JF0 addr	Jump if Flag 0	IF (F0) = 1 THEN (PC_{7-0}) ← addr	–	–	2
76 □□	JF1 addr	Jump if Flag 1	IF (F1) = 1 THEN (PC_{7-0}) ← addr	–	–	2

MCS-48 INSTRUCTIONS

Hex	Mnemonic	Description	Symbolic	Flags C	Flags AC	Cycles
_4 □□	CALL Xaddr	Subroutine call	((SP)) ← (PC), (PSW_{7-4}); (SP) ← (SP) + 1; (PC_{10-0}) ← addr; (PC_{11}) ← (DBF)	–	–	2
	14 □□ CALL 0addr 34 □□ CALL 1addr	54 □□ CALL 2addr 74 □□ CALL 3addr	94 □□ CALL 4addr B4 □□ CALL 5addr	D4 □□ CALL 6addr F4 □□ CALL 7addr		
27	CLR A	Clear A	(A) ← 0	–	–	1
97	CLR C	Clear Carry	(C) ← 0	0	–	1
85	CLR F0	Clear F0	(F0) ← 0	–	–	1
A5	CLR F1	Clear F1	(F1) ← 0	–	–	1
37	CPL A	Complement A	(A) ← NOT (A)	–	–	1
A7	CPL C	Complement carry	(C) ← NOT (C)	X	–	1
95	CPL F0	Complement F0	(F0) ← NOT (F0)	–	–	1
B5	CPL F1	Complement F1	(F1) ← NOT (F1)	–	–	1
57	DA A	Decimal Adjust Accumulator forming 2 BCD digits after ADD.	IF (AC) OR (A_{3-0} >9) THEN (A) ← (A) + 60 IF (C) OR (A_{7-4} > 9) THEN (A) ← (A) + 60	X	–	1
07	DEC A	Decrement A	(A) ← (A) – 1	–	–	1
C_	DEC R0–7	Decrement Register	(R) ← (R) – 1	–	–	1
	C8 DEC R0 C9 DEC R1	CA DEC R2 CB DEC R3	CC DEC R4 CD DEC R5	CE DEC R6 CF DEC R7		
15	DIS I	Disable external Interrupt		–	–	1
35	DIS TCNTI	Disable Timer/Counter Interrupt		–	–	1

MCS-48 INSTRUCTIONS

Hex	Mnemonic	Description	Symbolic	Flags C	Flags AC	Cycles
6_	ADD A, R0–7	Add register to A	(A) ← (A) + (R)	X	X	1
	68 ADD A, R0	6A ADD A, R2	6C ADD A, R4	6E ADD A, R6		
	69 ADD A, R1	6B ADD A, R3	6D ADD A, R5	6F ADD A, R7		
60	ADD A, @R0	Add indirect to A	(A) ← (A) + ((R0))	X	X	1
61	ADD A, @R1		(A) ← (A) + ((R1))	X	X	1
03 □□	ADD A, #data	Add immediate to A	(A) ← (A) + data	X	X	2
7_	ADDC A, R0–7	Add register with carry	(A) ← (A) + (R) + (C)	X	X	1
	78 ADDC A, R0	7A ADDC A, R2	7C ADDC A, R4	7E ADDC A, R6		
	79 ADDC A, R1	7B ADDC A, R3	7D ADDC A, R5	7F ADDC A, 27		
70	ADDC A, @R0	Add indirect with carry	(A) ← (A) + ((R0)) + (C)	X	X	1
71	ADDC A, @R1		(A) ← (A) + ((R1)) + (C)	X	X	1
13 □□	ADDC A, #data	Add immediate to A	(A) ← (A) + data	X	X	2
5_	ANL A, R0–7	And register to A	(A) ← (A) AND (R)	–	–	1
	58 ANL A, R0	5A ANL A, R2	5C ANL A, R4	5E ANL A, R6		
	59 ANL A, R1	5B ANL A, R3	5D ANL A, R5	5F ANL A, R7		
50	ANL A, @R0	And indirect to A	(A) ← (A) AND ((R0))	–	–	1
51	ANL A, @R1		(A) ← (A) AND ((R1))	–	–	1
53 □□	ANL A, #data	And immediate to A	(A) ← (A) AND data	–	–	2
9_ □□	ANL BUSP12, #data	And immediate to BUS, P1 or P2	(BUSP12) ← (BUSP12) AND data	–	–	2
	98 ANL BUS, #	99 ANL P1, #	9A ANL P2, #			
9_	ANLD P4–7, A	And A to expander P4–7	(P4–7) ← (P4–7) AND (A_{3-0})	–	–	2
	9C ANLD P4, A	9D ANLD P5, A	9E ANLD P6, A	9F ANLD P7, A		

BY HEX

84	□□	JMP, 4addr
85		CLR F0
86	□□	JNI, addr
87		undefined
88	□□	ORL BUS,#data
89	□□	ORL P1,#data
8A	□□	ORL P2,#data
8B		undefined
8C		ORLD P4,A
8D		ORLD P5,A
8E		ORLD P6,A
8F		ORLD P7,A
90		MOVX @R0,A
91		MOVX @R1,A
92	□□	JB4, addr
93		RETR
94	□□	CALL, 4addr
95		CPL F0
96	□□	JNZ, addr
97		CLR C
98	□□	ANL BUS,#data
99	□□	ANL P0,#data
9A	□□	ANL P1,#data
9B		undefined
9C		ANLD P4,A
9D		ANLD P5,A
9E		ANLD P6,A
9F		ANLD P7,A
A0		MOV @R0,A
A1		MOV @R1,A
A2		undefined
A3		MOVP A,@A
A4	□□	JMP, 5addr
A5		CLR F1
A6		undefined
A7		CPL C
A8		MOV R0,A
A9		MOV R1,A
AA		MOV R2,A
AB		MOV R3,A
AC		MOV R4,A
AD		MOV R5,A
AE		MOV R6,A
AF		MOV R7,A
B0	□□	MOV @R0,#data
B1	□□	MOV @R1,#data
B2	□□	JB5, addr
B3		JMMP @A
B4	□□	CALL, 5addr
B5		CPL F1
B6	□□	JF0, addr
B7		undefined
B8	□□	MOV R0,#data
B9	□□	MOV R1,#data
BA	□□	MOV R2,#data
BB	□□	MOV R3,#data
BC	□□	MOV R4,#data
BD	□□	MOV R5,#data
BE	□□	MOV R6,#data
BF	□□	MOV R7,#data
C0		undefined
C1		undefined
C2		undefined
C3		undefined
C4	□□	JMP, 6addr
C5		SEL RB0
C6	□□	JZ, addr
C7		MOV A,PSW
C8		DEC R0
C9		DEC R1
CA		DEC R2
CB		DEC R3
CC		DEC R4
CD		DEC R5
CE		DEC R6
CF		DEC R7
D0		XRL A,@R0
D1		XRL A,@R1
D2	□□	JB6, addr
DR	□□	XRL A,#data
D4	□□	CALL, 6addr
D5		SEL RB1
D6		undefined
D7		MOV PSW,A
D8		XRL A,R0
D9		XRL A,R1
DA		XRL A,R2
DB		XRL A,R3
DC		XRL A,R4
DD		XRL A,R5
DE		XRL A,R6
DF		XRL A,R7
E0		undefined
E1		undefined
E2		undefined
E3		MOVP3 A,@A
E4	□□	JMP, 7addr
E5		SEL MB0
E6		JNC
E7		RL A
E8	□□	DJNZ R0,addr
E9	□□	DJNZ R1,addr
EA	□□	DJNZ R2,addr
EB	□□	DJNZ R3,addr
EC	□□	DJNZ R4,addr
ED	□□	DJNZ R5,addr
EE	□□	DJNZ R6,addr
EF	□□	DJNZ R7,addr
F0		MOV A,@R0
F1		MOV A,@R1
F2	□□	JB7, addr
F3		undefined
F4	□□	CALL, 7addr
F5		SEL MB1
F6	□□	JC, addr
F7		RLC A
F8		MOV A,R0
F9		MOV A,R1
FA		MOV A,R2
FB		MOV A,R3
FC		MOV A,B4
FD		MOV A,R5
FE		MOV A,R6
FF		MOV A,R7

For further information, please refer to MCS-48 Assembly Language Programming Manual, 98-255 and the MCS-48 Microcomputer User's Manual.

BY HEX

00		NOP
01		undefined
02		OUTL BUS, A
03	□□	ADD A,#data
04	□□	JMP, 0addr
05		EN I
06		undefined
07		DEC A
08		INS A, BUS
09		IN A,P1
0A		IN A,P2
0B		undefined
0C		MOVD A,P4
0D		MOVD A,P5
0E		MOVD A,P6
0F		MOVD A,P7
10		INC @R0
11		INC @R1
12	□□	JB0, addr
13	□□	ADDC A,#data
14	□□	CALL, 0addr
15		DIS I
16	□□	JTF, addr
17		INC A
18		INC R0
19		INC R1
1A		INC R2
1B		INC R3
1C		INC R4
1D		INC R5
1E		INC R6
1F		INC R7
20		XCH A,@R0
21		XCH A,@R1
22		undefined
23	□□	MOV A,#data
24	□□	JMP, 1addr
25		EN TCNTI
26	□□	JNT0, addr
27		CLR A
28		XCH A,R0
29		XCH A,R1
2A		XCH A,R2
2B		XCH A,R3
2C		XCH A,R4
2D		XCH A,R5
2E		XCH A,R6
2F		XCH A,R7
30		XCHD A,@R0
31		XCHD A,@R1
32	□□	JB1, addr
33		undefined
34	□□	CALL, 1addr
35		DIS TCNTI
36	□□	JT0, addr
37		CPL A
38		undefined
39		OUTL P1,A
3A		OUTL P2,A
3B		undefined
3C		MOVD P4,A
3D		MOVD P5,A
3E		MOVD P6,A
3F		MOVD P7,A
40		ORL A,@R0
41		ORL A,@R1
42		MOV A, T
43	□□	ORL A,#data
44	□□	JMP, 2addr
45		STRT CNT
46	□□	JNT1, addr
47		SWAP A
48		ORL A,R0
49		ORL A,R1
4A		ORL A,R2
4B		ORL A,R3
4C		ORL A,R4
4D		ORL A,R5
4E		ORL A,R6
4F		ORL A,R7
50		ANL A,@R0
51		ANL A,@R1
52	□□	JB2, addr
53	□□	ANL A,#data
54	□□	CALL, 2addr
55		STRT T
56	□□	JT1, addr
57		DA A
58		ANL A,R0
59		ANL A,R1
5A		ANL A,R2
5B		ANL A,R3
5C		ANL A,R4
5D		ANL A,R5
5E		ANL A,R6
5F		ANL A,R7
60		ADD A,@R0
61		ADD A,@R1
62		MOV T, A
63		undefined
64	□□	JMP, 3addr
65		STOP TCNT
66		undefined
67		RRC A
68		ADD A,R0
69		ADD A,R1
6A		ADD A,R2
6B		ADD A,R3
6C		ADD A,R4
6D		ADD A,R5
6E		ADD A,R6
6F		ADD A,R7
70		ADDC A,@R0
71		ADDC A,@R1
72	□□	JB3, addr
73		undefined
74	□□	CALL, 3addr
75		ENT0 CLK
76	□□	JF1, addr
77		RR A
78		ADDC A,R0
79		ADDC A,R1
7A		ADDC A,R2
7B		ADDC A,R3
7C		ADDC A,R4
7D		ADDC A,R5
7E		ADDC A,R6
7F		ADDC A,R7
80		MOVX A,@R0
81		MOVX A,@R1
82		undefined
83		RET

BY MNEMONIC

MOV A, #data	23	☐☐
MOV A, PSW	C7	
MOV A, R0	F8	
R1	F9	
R2	FA	
R3	FB	
R4	FC	
R5	FD	
R6	FE	
R7	FF	
MOV A, @R0	F0	
@R1	F1	
MOV A, T	42	
• MOV PSW, A	D7	
MOV R0, A	A8	
R1, A	A9	
R2, A	AA	
R3, A	AB	
R4, A	AC	
R5, A	AD	
R6, A	AE	
R7, A	AF	
MOV R0, #data	B8	☐☐
R1, #data	B9	☐☐
R2, #data	BA	☐☐
R3, #data	BB	☐☐
R4, #data	BC	☐☐
R5, #data	BD	☐☐
R6, #data	BE	☐☐
R7, #data	BF	☐☐
MOV @R0, A	A0	
MOV @R1, A	A1	
MOV @R0, #data	B0	☐☐
@R1, #data		☐☐
MOV T, A	62	
MOVD A, P4	0C	
P5	0D	
P6	0E	
P7	0E	
MOVD P4, A	3C	
P5, A	3D	
P6, A	3E	
P7, A	3F	
MOVP A, @A	A3	
MOVP3 A, @A	E3	
MOVX A, @R0	80	
@R1	81	
MOVX @R0, A	90	
@R1, A	91	
NOP	00	
ORL A, R0	48	
R1	49	
R2	4A	
R3	4B	
R4	4C	
R5	4D	
R6	4E	
R7	4F	
ORL BUS, #data	88	☐☐
P1, #data	89	☐☐
P2, #data	8A	☐☐
ORLD P4, A	8C	
P5, A	8D	
P6, A	8E	
ORLD P7, A	8F	
OUTL BUS, A	02	
P1, A	39	
P2, A	3A	
RET	83	
RETR	93	
RL A	E7	
• RLC A	F7	
RR A	77	
• RRC A	67	
SEL MB0	E5	
SEL MB1	F5	
SEL RB0	C5	
SEL RB1	D5	
STOP TCNT	65	
STRT CNT	45	
STRT T	55	
SWAP A	47	
XCH A, R0	28	
R1	29	
R2	2A	
R3	2B	
R4	2C	
R5	2D	
R6	2E	
R7	2F	
XCH A, @R0	20	
XCH A, @R1	21	
XCHD A, @R0	30	
@R1	31	
XRL A, R0	D8	
R1	D9	
R2	DA	
R3	DB	
R4	DC	
R5	DD	
R6	DE	
R7	DF	
XRL A, @R0	D0	
@R1	D1	

• CARRY FLAG AFFECTED

BY MNEMONIC

MOV A, #data	23	☐☐
MOV A, PSW	C7	
MOV A, R0	F8	
R1	F9	
R2	FA	
R3	FB	
R4	FC	
R5	FD	
R6	FE	
R7	FF	
MOV A, @R0	F0	
@R1	F1	
MOV A, T	42	
• MOV PSW, A	D7	
MOV R0, A	A8	
R1, A	A9	
R2, A	AA	
R3, A	AB	
R4, A	AC	
R5, A	AD	
R6, A	AE	
R7, A	AF	
MOV R0, #data	B8	☐☐
R1, #data	B9	☐☐
R2, #data	BA	☐☐
R3, #data	BB	☐☐
R4, #data	BC	☐☐
R5, #data	BD	☐☐
R6, #data	BE	☐☐
R7, #data	BF	☐☐
MOV @R0, A	A0	
MOV @R1, A	A1	
MOV @R0, #data	B0	☐☐
@R1, #data		☐☐
MOV T, A	62	
MOVD A, P4	0C	
P5	0D	
P6	0E	
P7	0E	
MOVD P4, A	3C	
P5, A	3D	
P6, A	3E	
P7, A	3F	
MOVP A, @A	A3	
MOVP3 A, @A	E3	
MOVX A, @R0	80	
@R1	81	
MOVX @R0, A	90	
@R1, A	91	
NOP	00	
ORL A, R0	48	
R1	49	
R2	4A	
R3	4B	
R4	4C	
R5	4D	
R6	4E	
R7	4F	
ORL BUS, #data	88	☐☐
P1, #data	89	☐☐
P2, #data	8A	☐☐
ORLD P4, A	8C	
P5, A	8D	
P6, A	8E	
ORLD P7, A	8F	
OUTL BUS, A	02	
P1, A	39	
P2, A	3A	
RET	83	
RETR	93	
RL A	E7	
• RLC A	F7	
RR A	77	
• RRC A	67	
SEL MB0	E5	
SEL MB1	F5	
SEL RB0	C5	
SEL RB1	D5	
STOP TCNT	65	
STRT CNT	45	
STRT T	55	
SWAP A	47	
XCH A, R0	28	
R1	29	
R2	2A	
R3	2B	
R4	2C	
R5	2D	
R6	2E	
R7	2F	
XCH A, @R0	20	
XCH A, @R1	21	
XCHD A, @R0	30	
@R1	31	
XRL A, R0	D8	
R1	D9	
R2	DA	
R3	DB	
R4	DC	
R5	DD	
R6	DE	
R7	DF	
XRL A, @R0	D0	
@R1	D1	

• CARRY FLAG AFFECTED

BY MNEMONIC

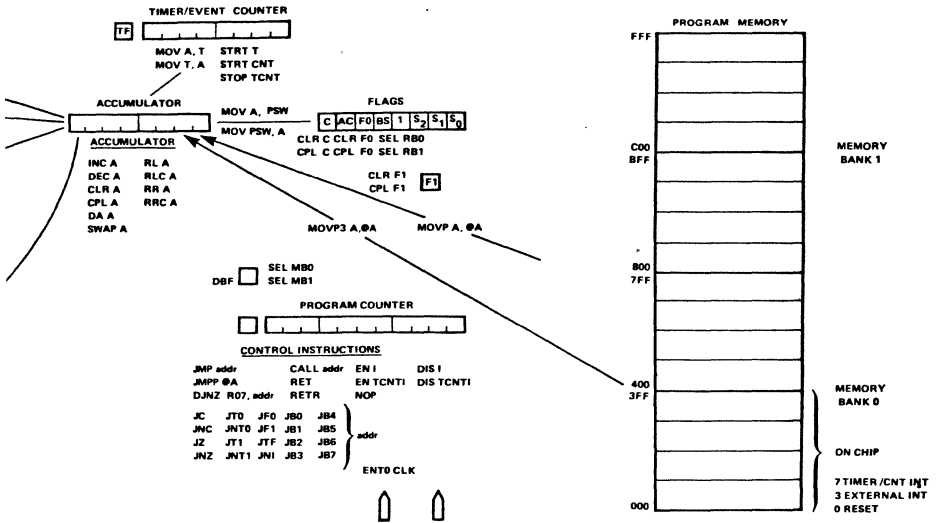
• ADD A, R0	68	
R1	69	
R2	6A	
R3	6B	
R4	6C	
R5	6D	
R6	6E	
R7	6F	
• ADD A, @R0	60	
@R1	61	
• ADD A, #data	03	□□
• ADDC A, R0	78	
R1	79	
R2	7A	
R3	7B	
R4	7C	
R5	7D	
R6	7E	
R7	7F	
• ADDC A, @R0	70	
@R1	71	
• ADDC A, #data	13	□□
ANL A, R0	58	
R1	59	
R2	5A	
R3	5B	
R4	5C	
R5	5D	
R6	5E	
R7	5F	
ANL A, @R0	50	

ANL A, @R1	51	
ANL A, #data	53	□□
ANL BUS, #data	98	□□
P1, #data	99	□□
P2, #data	9A	□□
CALL 0addr	14	□□
1addr	34	□□
2addr	54	□□
3addr	74	□□
4addr	94	□□
5addr	B4	□□
6addr	D4	□□
7addr	F4	□□
CLR A	27	
• CLR C	97	
CLR F0	85	
CLR F1	A5	
CPL A	37	
• CPL C	A7	
CPL F0	95	
CPL F1	B5	
• DA A	57	
DEC A	07	
DEC R0	C8	
R1	C9	
R2	CA	
R3	CB	
R4	CC	
R5	CD	
R6	CE	

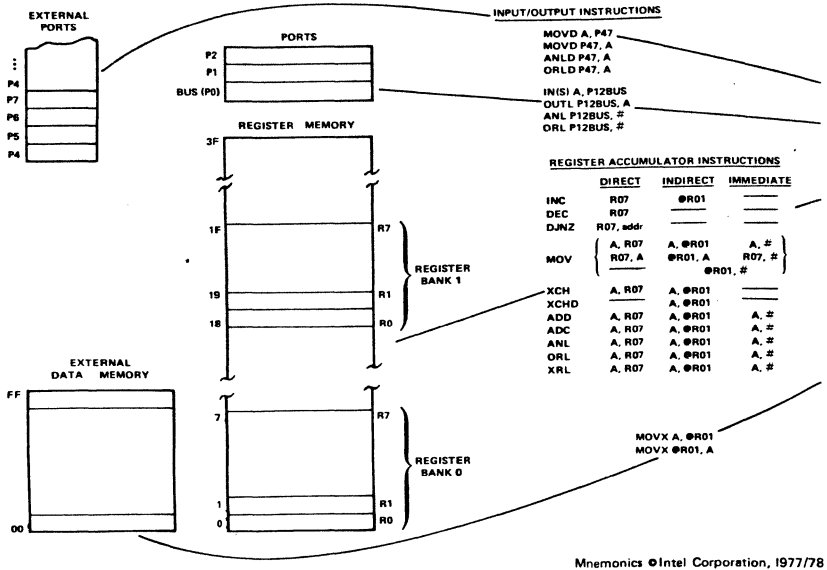
DEC R7	CF	
DIS I	15	
DIS TCNTI	35	
DJNZ R0,addr	E8	□□
R1,addr	E9	□□
R2,addr	EA	□□
R3,addr	EB	□□
R4,addr	EC	□□
R5,addr	ED	□□
R6,addr	EE	□□
R7,addr	EF	□□
EN I	05	
EN TCNTI	25	
ENT0 CLK	75	
INS A, BUS	08	
IN A, P1	09	
IN A, P2	0A	
INC A	17	
INC R0	18	
INC R1	19	
INC R2	1A	
INC R3	1B	
INC R4	1C	
INC R5	1D	
INC R6	1E	
INC R7	1F	
INC @R0	10	
INC @R1	11	

JB0 addr	12	□□
JB1 addr	32	□□
JB2 addr	52	□□
JB3 addr	72	□□
JB4 addr	94	□□
JB5 addr	B2	□□
JB6 addr	D2	□□
JB7 addr	F2	□□
JC addr	F6	□□
JF0 addr	B6	□□
JF1 addr		□□
JMP 0addr	04	□□
1addr	24	□□
2addr	44	□□
3addr	64	□□
4addr	84	□□
5addr	A4	□□
6addr	C4	□□
7addr	E4	□□
JMPP @A	B3	□□
JNC addr	E6	□□
JNI addr	86	□□
JNT0 addr	26	□□
JNT1 addr	46	□□
JNZ addr	96	□□
JTF addr	16	□□
JT0 addr	36	□□
JT1 addr	56	□□
JZ addr	C6	□□

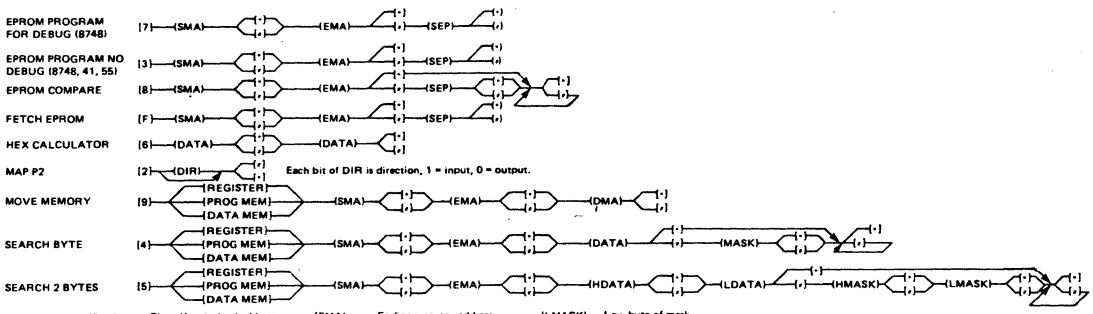
MICROMAP



MICROMAP



FUNCTIONS

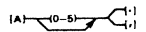


abbreviations:

(BIAS)	Bias offset to load address	(EMA)	Ending memory address	(LMASK)	Low byte of mask
(BKA)	Breakpoint address	(HDATA)	High byte of data	(MASK)	Mask
(DATA)	Data	(HMASK)	High byte of mask	(SEP)	Starting EPROM Address
(DIR)	Direction for lines of Port 2	(LDATA)	Low byte of data	(SMA)	Starting Memory Address
(DMA)	Destination memory address				

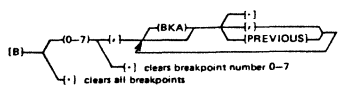
FUNCTIONS

ACCESS

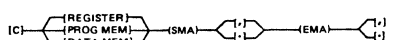


Address Code	Program Memory	System IO and System Calls	Expansion Memory and IO	OUTL Port 0
0	WRITABLE (RAM)	no	no	yes
1	WRITABLE (RAM)	no	yes	no
2	WRITABLE (RAM)	yes	no	no
3	READ ONLY (ON CHIP)	no	no	yes
4	READ ONLY (ON CHIP)	no	yes	no
5	READ ONLY (ON CHIP)	yes	no	no

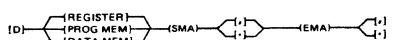
BREAKPOINT



CLEAR

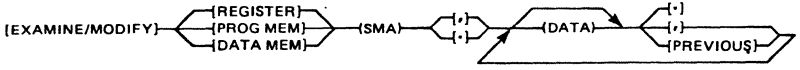


DUMP



ENTER

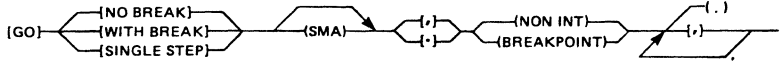
COMMANDS



MCS 48 Processors have 64 bytes register memory, numbered 0–$3F_{16}$. Prompt 48 allows access to other "register" locations via [EXAMINE/MODIFY] [REGISTER].

Number	Location	Format
40	ACCUMULATOR	
41	TIMER	
42	PSW	CY | AC | F0 | RB | F1 | S_2 | S_1 | S_0
43	PCL	
44	PCH	
45	BUS (PORT 0)	READ-ONLY
46	PORT 1	READ-ONLY
47	PORT 2	
48	MISC	Counter Run | Timer Run | Timer Flag | Nested Fr Int | Will En Int | Mem Bank | T1 | T0

Prompt 48 provides 256 bytes data memory, numbered 0–FF_{16}.



Ensure you have selected the correct access code, P2 MAP and LSN P2 contents before running programs.

abbreviations: (DATA) data (SMA) starting memory address

CAUTIONS

MCS-48 Processors are fragile. Dropping, twisting or uneven pressure may break them. Leave them in the protective foam until ready to use. Never press down on the cap of the processor.

Never subject MOS devices to the discharge of static electricity. (Touch the chassis of Prompt 48 before inserting processors in the panel sockets.)

Ensure a processor is properly locked in the EXECUTION SOCKET before inserting a processor in the PROGRAMMING SOCKET. Remove PROGRAMMING SOCKET processor before removing EXECUTION SOCKET processor.

Use cable puller to remove 50-pin cable from I/O PORTS and BUS CONNECTOR.

Appendix A of the Prompt 48 User's Manual contains a familiarization exercise for new comers to Prompt 48.