MSB's / LSB's	0Ø	1Ø	2Ø	3Ø	4Ø	5Ø	6Ø	7Ø	8Ø	9Ø	AØ	BØ	CØ	DØ	EØ	FØ
Ø	NOP					MOV D,B	MOV H,B	MOV M,B	ADD B	SUB B	ANA B	ORA B	RNZ	RNC	RPO	RP Ø
1	LXI B	LXI D	LXI H	LXI SP	MOV B,C	MOV D,C	MOV H,C	MOV M,C	ADD C	SUB C	ANA C	ORA C	POP B	POP D	POP H	POP PSW 1
2	STAX B	STAX D	SHLD	STA	MOV B,D		MOV H,D	MOV M,D	ADD D	SUB D	ANA D	ORA D	JNZ	JNC	JPO	JP 2
3	INX B	INX D	INX H	INX SP	MOV B,E	MOV D,E	MOV H,E	MOV M,E	ADD E	SUB E	ANA E	ORA E	JMP	OUT	XTHL	DI 3
4	INR B	INR D	INR H	INR M	MOV B,H	MOV D,H		MOV M,H	ADD H	SUB H	ANA H	ORA H	CNZ	CNC	CPO	CP 4
5	DCR B	DCR D	DCR H	DCR M	MOV B,L	MOV D,L	MOV H,L	MOV M,L	ADD L	SUB L	ANA L	ORA L	PUSH B	PUSH D	PUSH H	PUSH PSW 5
6	MVI B	MVI D	MVI H	MVI M	MOV B,M	MOV D,M	MOV H,M	STR	ADD M	SUB M	ANA M	ORA M	ADI	SUI	ANI	ORI 6
7	RLC	RAL	DAA	STC	MOV B,A	MOV D,A	MOV H,A	MOV M,A	ADD A	SUB A	ANA A	ORA A	RST Ø	RST 2	RST 4	RST 6 7
8					MOV C,B	MOV E,B	MOV L,B	MOV A,B	ADC B	SBB B	XRA B	CMP B	RZ	RC	RPE	RM 8
9	DAD B	DAD D	DAD H	DAD SP		MOV E,C	MOV L,C	MOV A,C	ADC C	SBB C	XRA C	CMP C	RET		PCHL	SPHL 9
A	LDAX B	LDAX D	LHLD	LDA	MOV C,D	MOV E,D	MOV L,D	MOV A,D	ADC D	SBB D	XRA D	CMP D	JZ	JC	JPE	JM A
B	DCX B	DCX D	DCX H	DCX SP	MOV C,E		MOV L,E	MOV A,E	ADC E	SBB E	XRA E	CMP E		IN	XCHG	EI B
C	INR C	INR E	INR L	INR A	MOV C,H	MOV E,H	MOV L,H	MOV A,H	ADC H	SBB H	XRA H	CMP H	CZ	CC	CPE	CM C
D	DCR C	DCR E	DCR L	DCR A	MOV C,L	MOV E,L		MOV A,L	ADC L	SBB L	XRA L	CMP L	CALL			D
E	MVI C	MVI E	MVI L	MVI A	MOV C,M	MOV E,M	MOV L,M	MOV A,M	ADC M	SBB M	XRA M	CMP M	ACI	SBI	XRI	CPI E
F	RRC	RAR	CMA	CMC	MOV C,A	MOV E,A	MOV L,A		ADC A	SBB A	XRA A	CMP A	RST 1	RST 3	RST 5	RST 7 F

HEX-ASCII TABLE

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

EDIT 80/A COMMANDS

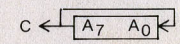
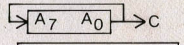
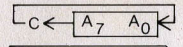
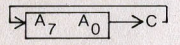
nA$$	APPEND INPUT
B$$	POINT TO BEGINNING
nC$$	MOVE CHARACTER POINTER
nD$$	DELETE CHARACTERS
E$$	END, PUNCH AND CLEAR
F	FIND
I	INSERT
nK$$	KILL LINES
nL$$	MOVE LINE POINTER
N$$	OUTPUT NULLS
(n)S	SEARCH AND SUBSTITUTE
nT$$	LIST LINES
nW$$	PUNCH AND CLEAR LINES
Z$$	POINT TO END OF WORKSPACE

J. L. UNGER

pcs

MicroPac 80/A
Programmers' Card

ACCUMULATOR INSTRUCTIONS

LDA	adr	[adr] → (A)
STA	adr	(A) → [adr]
LDAX	rp	[rp,rp -1] → (A)
STAX	rp	(A) → [rp, rp - 1]
*ADD	r	(A)+(r) → (A)
*ADC	r	(A)+(r)+C → (A)
*SUB	r	(A) - (r) → (A)
*SBB	r	(A) - (r) - C → (A)
*CMP	r	(A) - (r)
*ANA	r	(A)∧(r) → (A)
*ORA	r	(A)∨(r) → (A)
*XRA	r	(A)∀(r) → (A)
*ADI	D8	(A)+D8 → (A)
*ACI	D8	(A)+D8+C → (A)
*SUI	D8	(A) - D8 → (A)
*SBI	D8	(A) - D8 - C → (A)
*CPI	D8	(A) - D8
*ANI	D8	(A)∧D8 → (A)
*ORI	D8	(A)∨D8 → (A)
*XRI	D8	(A)∀D8 → (A)
CMA		(Ā)→(A)
*DAA		Decimal Adjust (A)
†RLC		
†RRC		
†RAL		
†RAR		
†STC		1 → C
†CMC		C → C

REGISTER INSTRUCTIONS

MOV	r1, r2	(r1) ← (r2)
MVI	r, D8	D8 → (r)
**INR	r	(r)+1 → (r)
**DCR	r	(r) - 1 → (r)

H & L INSTRUCTIONS

LHLD	adr	(adr - 1, adr) → (H&L)
SHLD	adr	(H&L) → (adr - 1, adr)
† DAD	rp	(H&L)+(rp) → (H&L)
XCHG		(H&L) ⇄ (D&E)
XTHL		(H&L) ⇄ Top of Stack
PCHL		(H&L) → P.C.
SPHL		(H&L) → Stack Pointer

INDEX REG. INSTRUCTIONS

LXI	rp, D16	D16 → (rp)
INX	rp	(rp)+1 → (rp)
DCX	rp	(rp) - 1 → (rp)
PUSH	rp	(rp) → Top of Stack
POP	rp	Top of Stack → (rp)

Type	MSB's
SIN/COUT	ØFFH
DIN/DOUT	ØFEH

I/O OPERATIONS

EI	Enable Interrupts
DI	Disable Interrupts
IN 1	(I) → (A)
OUT 1	(A) → (0)
IN Ø	Interrupt Acknowledge
OUT Ø	Master Clear

CONTROL INSTRUCTIONS

NOP		
JMP	adr	(adr) → P.C.
Jcc	adr	IF cc TRUE, (adr) → P.C.
CALL	adr	Call Subroutine
Ccc	adr	IF cc TRUE, call subroutine
RET		RETURN FROM SUBROUTINE
Rcc		IF cc TRUE, RETURN
RST	n	RESTART n, n - 0, 1, 2, ... , 7
STR		Ø → P.C., DI

Flag Register

7	6	5	4	3	2	1	0
S	Z	0	A C	0	P	1	C

PSEUDO OPERATIONS

ORG	adr	ORIGIN
END	adr	END ASSEMBLY
EQU	D16	EQUATE SYMBOL
SET	D16	SET SYMBOL
DS	D16	DEFINE STORAGE
DB	D8	DEFINE BYTE(S)
DW	D16	DEFINE WORD(S)
IF	D16	CONDITIONAL ASSEMBLY, Ø - FALSE
ENDIF		END CONDITIONAL ASSM.
MACRO		DEFINE MACRO
ENDM		END MACRO

OPERATORS

( )
* / MOD SHL SHR
+ −
NOT
AND
OR XOR

SYMBOL DEFINITIONS

SYMBOL	DESCRIPTION
A	ACCUMULATOR
adr	ADDRESS
C	CARRY BIT
cc	CONDITION CODE C - Carry, NC - Not Carry P - Positive, M - Minus Z - Zero, NZ - Not Zero
D8	CONSTANT OR ARITHMETIC EXPRESSION FOR 8 BIT DATA
D16	CONSTANT OR ARITHMETIC EXPRESSION FOR 16 BIT DATA
I	MSB INPUT REGISTER
0	MSB OUTPUT REGISTER
P.C.	PROGRAM COUNTER
r	REGISTER (A,B,C,D,H,L or M)
rp	REGISTER PAIR
(SP)	SPACE
*	ALL CONDITION FLAGS AFFECTED
**	ALL FLAGS EXCEPT CARRY AFFECTED
†	CARRY AFFECTED
( )	CONTENTS OF
[ ]	MEMORY CONTENTS AT
→	REPLACES
⇄	INTERCHANGES VALUES
∧	AND
∨	OR
∀	EXCLUSIVE OR

X<A X=A X>A
NZ,NC Z,NC NZ,C

BOS80/A COMMANDS

A	ldev = pdev	ASSIGN DEVICE
D	low, high	DUMP
E	adr	END RECORD
F	low, high, data	FILL MEMORY
G	adr, bk1, bk2	GO
H	data, data (SP)	HEX ADD
In		n - Ø, INTRPS OFF
M	low, high, dest	MOVE MEMORY
N		OUTPUT NULLS
Q		FDOS REQUEST
R	bias	READ HEX OBJECT
S	adr (SP)	SUBSTITUTE MEMORY
V	bias	VERIFY OBJECT
W	low, high	WRITE MEMORY
Xr		EXAMINE REGISTER(S)
Z		ZERO MODEL 733 BUFFER

ADDS

8Ø	ADD	B
81	ADD	C
82	ADD	D
83	ADD	E
84	ADD	H
85	ADD	L
86	ADD	M
87	ADD	A
88	ADC	B
89	ADC	C
8A	ADC	D
8B	ADC	E
8C	ADC	H
8D	ADC	L
8E	ADC	M
8F	ADC	A
C6	ADI	
CE	ACI	

ANDS

A0	ANA	B
A1	ANA	C
A2	ANA	D
A3	ANA	E
A4	ANA	H
A5	ANA	L
A6	ANA	M
A7	ANA	A
E6	ANI	

CALLS

CD	CALL	
C4	CNZ	
CC	CZ	
D4	CNC	
DC	CC	
E4	CPO	
EC	CPE	
F4	CP	
FC	CM	

COMPARES

B8	CMP	B
B9	CMP	C
BA	CMP	D
BB	CMP	E
BC	CMP	H
BD	CMP	L
BE	CMP	M
BF	CMP	A
FE	CPI	

— — — —

2F	CMA
3F	CMC
27	DAA

DECREMENTS

Ø5	DCR	B
ØD	DCR	C
15	DCR	D
1D	DCR	E
25	DCR	H
2D	DCR	L
35	DCR	M
3D	DCR	A
ØB	DCX	B
1B	DCX	D
2B	DCX	H
3B	DCX	SP

— — — —

F3	DI
FB	EI
DB	IN
ØØ	NOP
D3	OUT

DOUBLE ADDS

Ø9	DAD	B
19	DAD	D
29	DAD	H
39	DAD	SP

INCREMENTS

Ø4	INR	B
ØC	INR	C
14	INR	D
1C	INR	E
24	INR	H
2C	INR	L
34	INR	M
3C	INR	A
Ø3	INX	B
13	INX	D
23	INX	H
33	INX	SP

JUMP

C3	JMP
C2	JNZ
CA	JZ
D2	JNC
DA	JC
E2	JPO
EA	JPE
F2	JP
FA	JM

LOADS

3A	LDA	
ØA	LDAX	B
1A	LDAX	D
2A	LHLD	
Ø1	LXI	B
11	LXI	D
21	LXI	H
31	LXI	SP

MOVES

Ø6	MVI	B,
ØE	MVI	C,
16	MVI	D,
1E	MVI	E,
26	MVI	H,
2E	MVI	L,
36	MVI	M,
3E	MVI	A,
41	MOV	B,C
42	MOV	B,D
43	MOV	B,E
44	MOV	B,H
45	MOV	B,L
46	MOV	B,M
47	MOV	B,A
48	MOV	C,B
4A	MOV	C,D
4B	MOV	C,E
4C	MOV	C,H
4D	MOV	C,L
4E	MOV	C,M
4F	MOV	C,A
50	MOV	D,B
51	MOV	D,C
53	MOV	D,E
54	MOV	D,H
55	MOV	D,L
56	MOV	D,M
57	MOV	D,A
58	MOV	E,B
59	MOV	E,C
5A	MOV	E,D
56	MOV	E,H
5D	MOV	E,L
5E	MOV	E,M
5F	MOV	E,A
60	MOV	H,B
61	MOV	H,C
62	MOV	H,D
63	MOV	H,E
65	MOV	H,L
66	MOV	H,M
67	MOV	H,A
68	MOV	L,B
69	MOV	L,C
6A	MOV	L,D
6B	MOV	L,E
6C	MOV	L,H
6E	MOV	L,M
6F	MOV	L,A
70	MOV	M,B
71	MOV	M,C
72	MOV	M,D
73	MOV	M,E
74	MOV	M,H
75	MOV	M,L
77	MOV	M,A
78	MOV	A,B
79	MOV	A,C
7A	MOV	A,D
7B	MOV	A,E
7C	MOV	A,H
7D	MOV	A,L
7E	MOV	A,M

LOGICAL OR's

BØ	ORA	B
B1	ORA	C
B2	ORA	D
B3	ORA	E
B4	ORA	H
B5	ORA	L
B6	ORA	M
B7	ORA	A
F6	ORI	

PUSH's & POP's

C5	PUSH	B
D5	PUSH	D
E5	PUSH	H
F5	PUSH	PSW
C1	POP	B
D1	POP	D
E1	POP	H
F1	POP	PSW

RESTARTS

C7	RST	Ø
CF	RST	1
D7	RST	2
DF	RST	3
E7	RST	4
EF	RST	5
F7	RST	6
FF	RST	7

RETURNS

C9	RET
CØ	RNZ
C8	RZ
DØ	RNC
D8	RC
EØ	RPO
E8	RPE
FØ	RP
F8	RM

ROTATES

Ø7	RLC
ØF	RRC
17	RAL
1F	RAR

STORES

32	STA	
Ø2	STAX	B
12	STAX	D
22	SHLD	

— — — —

F9	SPHL
37	STC
76	STR
EB	XCHG
E3	XTHL

SUBTRACTS

9Ø	SUB	B
91	SUB	C
92	SUB	D
93	SUB	E
94	SUB	H
95	SUB	L
96	SUB	M
97	SUB	A
98	SBB	B
99	SBB	C
9A	SBB	D
9B	SBB	E
9C	SBB	H
9D	SBB	L
9E	SBB	M
9F	SBB	A
D6	SUI	
DE	SBI	

EXCLUSIVE-OR's

A8	XRA	B
A9	XRA	C
AA	XRA	D
AB	XRA	E
AC	XRA	H
AD	XRA	L
AE	XRA	M
AF	XRA	A
EE	XRI	