MICRO LOGIC CORP.

HACKENSACK, NJ

Z80 CPU

MICROPROCESSOR INSTANT REFERENCE CARD

MICRO ® CHART

Single-Byte-Opcode to Instruction Conversion

LSD ➔

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	NOP	LD BC,nn	LD (BC),A	INC BC	INC B	DEC B	LD B,n	RLCA	EX AF,AF'	ADD HL,BC	LD A,(BC)	DEC BC	INC C	DEC C	LD C,n	RRCA	0
1	DJNZ n	LD DE,nn	LD (DE),A	INC DE	INC D	DEC D	LD D,n	RLA	JR n	ADD HL,DE	LD A,(DE)	DEC DE	INC E	DEC E	LD E,n	RRA	1
2	JR NZ,n	LD HL,nn	LD (nn),HL	INC HL	INC H	DEC H	LD H,n	DAA	JR Z,n	ADD HL,HL	LD HL,(nn)	DEC HL	INC L	DEC L	LD L,n	CPL	2
3	JR NC,n	LD SP,nn	LD (nn),A	INC SP	INC (HL)	DEC (HL)	LD (HL),n	SCF	JR C,n	ADD HL,SP	LD A,(nn)	DEC SP	INC A	DEC A	LD A,n	CCF	3
4	LD B,B	LD B,C	LD B,D	LD B,E	LD B,H	LD B,L	LD B,(HL)	LD B,A	LD C,B	LD C,C	LD C,D	LD C,E	LD C,H	LD C,L	LD C,(HL)	LD C,A	4
5	LD D,B	LD D,C	LD D,D	LD D,E	LD D,H	LD D,L	LD D,(HL)	LD D,A	LD E,B	LD E,C	LD E,D	LD E,E	LD E,H	LD E,L	LD E,(HL)	LD E,A	5
6	LD H,B	LD H,C	LD H,D	LD H,E	LD H,H	LD H,L	LD H,(HL)	LD H,A	LD L,B	LD L,C	LD L,D	LD L,E	LD L,H	LD L,L	LD L,(HL)	LD L,A	6
7	LD (HL),B	LD (HL),C	LD (HL),D	LD (HL),E	LD (HL),H	LD (HL),L	HALT	LD (HL),A	LD A,B	LD A,C	LD A,D	LD A,E	LD A,H	LD A,L	LD A,(HL)	LD A,A	7
8	ADD A,B	ADD A,C	ADD A,D	ADD A,E	ADD A,H	ADD A,L	ADD A,(HL)	ADD A,A	ADC A,B	ADC A,C	ADC A,D	ADC A,E	ADC A,H	ADC A,L	ADC A,(HL)	ADC A,A	8
9	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB (HL)	SUB A	SBC A,B	SBC A,C	SBC A,D	SBC A,E	SBC A,H	SBC A,L	SBC A,(HL)	SBC A,A	9
A	AND B	AND C	AND D	AND E	AND H	AND L	AND (HL)	AND A	XOR B	XOR C	XOR D	XOR E	XOR H	XOR L	XOR (HL)	XOR A	A
B	OR B	OR C	OR D	OR E	OR H	OR L	OR (HL)	OR A	CP B	CP C	CP D	CP E	CP H	CP L	CP (HL)	CP A	B
C	RET NZ	POP BC	JP NZ,nn	JP nn	CALL NZ,nn	PUSH BC	ADD A,n	RST 00H	RET Z	RET	JP Z,nn	table	CALL Z,nn	CALL nn	ADC A,n	RST 08H	C
D	RET NC	POP DE	JP NC,nn	OUT (n),A	CALL NC,nn	PUSH DE	SUB n	RST 10H	RET C	EXX	JP C,nn	IN A,(n)	CALL C,nn	table	SBC A,n	RST 18H	D
E	RET PO	POP HL	JP PO,nn	EX (SP),HL	CALL PO,nn	PUSH HL	AND n	RST 20H	RET PE	JP (HL)	JP PE,nn	EX DE,HL	CALL PE,nn	table	XOR n	RST 28H	E
F	RET P	POP AF	JP P,nn	DI	CALL P,nn	PUSH AF	OR n	RST 30H	RET M	LD SP,HL	JP M,nn	EI	CALL M,nn	table	CP n	RST 38H	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Multi-Byte-Opcode to Instruction Conversion

CB00	RLC B
CB01	RLC C
CB02	RLC D
CB03	RLC E
CB04	RLC H
CB05	RLC L
CB06	RLC (HL)
CB07	RLC A
CB08	RRC B
CB08	RRC C
CB0A	RRC D
CB0B	RRC E
CB0C	RRC H
CB0D	RRC L
CB0E	RRC (HL)
CB0F	RRC A
CB10	RL B
CB11	RL C
CB12	RL D
CB13	RL E
CB14	RL H
CB15	RL L
CB16	RL (HL)
CB17	RL A
CB18	RR B
CB19	RR C
CB1A	RR D
CB1B	RR E
CB1C	RR H
CB1D	RR L
CB1E	RR (HL)
CB1F	RR A
CB20	SLA B
CB21	SLA C
CB22	SLA D
CB23	SLA E
CB24	SLA H
CB25	SLA L
CB26	SLA (HL)
CB27	SLA A
CB28	SRA B
CB29	SRA C
CB2A	SRA D
CB2B	SRA E
CB2C	SRA H
CB2D	SRA L
CB2E	SRA (HL)
CB2F	SRA A
CB38	SRL B
CB39	SRL C
CB3A	SRL D
CB3B	SRL E
CB3C	SRL H
CB3D	SRL L
CB3E	SRL (HL)
CB3F	SRL A
CB40	see BIT
...	see RES
CBFF	see SET

ED40	IN B,(C)
ED41	OUT (C),B
ED42	SBC HL,BC
ED43aa	LD (aa),BC
ED44	NEG
ED45	RETN
ED46	IM 0
ED47	LD I,A
ED48	IN C,(C)
ED49	OUT (C),C
ED4A	ADC HL,BC
ED4Baa	LD BC,(aa)
ED4D	RETI
ED4F	LD R,A
ED50	IN D,(C)
ED51	OUT (C),D
ED52	SBC HL,DE
ED53aa	LD (aa),DE
ED56	IM 1
ED57	LD A,I
ED58	IN E,(C)
ED59	OUT (C),E
ED5A	ADC HL,DE
ED5Baa	LD DE,(aa)
ED5E	IM 2
ED5F	LD A,R
ED60	IN H,(C)
ED61	OUT (C),H
ED62	SBC HL,HL
ED67	RRD
ED68	IN L,(C)
ED69	OUT (C),L
ED6A	ADC HL,HL
ED6F	RLD
ED72	SBC HL,SP
ED73aa	LD (aa),SP
ED78	IN A,(C)
ED79	OUT (C),A
ED7A	ADC HL,SP
ED7Baa	LD SP,(aa)
EDA0	LDI
EDA1	CPI
EDA2	INI
EDA3	OUTI
EDA8	LDD
EDA9	CPD
EDAA	IND
EDAB	OUTD
EDB0	LDIR
EDB1	CPIR
EDB2	INIR
EDB3	OTIR
EDB8	LDDR
EDB9	CPDR
EDBA	INDR
EDBB	OTDR

%%09	ADD XY,BC
%%19	ADD XY,DE
%%21aa	LD XY,aa
%%22aa	LD (aa),XY
%%23	INC XY
%%29	ADD XY,XY
%%2Aaa	LD XY,(aa)
%%2B	DEC XY
%%34d	INC (XY+d)
%%35d	DEC (XY+d)
%%36dn	LD (XY+d),n
%%39	ADD XY,SP
%%46d	LD B,(XY+d)
%%4Ed	LD C,(XY+d)
%%56d	LD D,(XY+d)
%%5Ed	LD E,(XY+d)
%%66d	LD H,(XY+d)
%%6Ed	LD L,(XY+d)
%%70d	LD (XY+d),B
%%71d	LD (XY+d),C
%%72d	LD (XY+d),D
%%73d	LD (XY+d),E
%%74d	LD (XY+d),H
%%75d	LD (XY+d),L
%%77d	LD (XY+d),A
%%7Ed	LD A,(XY+d)
%%86d	ADD A,(XY+d)
%%8Ed	ADC A,(XY+d)
%%96d	SUB (XY+d)
%%9Ed	SBC A,(XY+d)
%%A6d	AND (XY+d)
%%AEd	XOR (XY+d)
%%B6d	OR (XY+d)
%%BEd	CP (XY+d)

%%means DD or FD and
for DD, XY means IX
for FD, XY means IY

%%CBd06	RLC (XY+d)
%%CBd0E	RRC (XY+d)
%%CBd16	RL (XY+d)
%%CBd1E	RR (XY+d)
%%CBd26	SLA (XY+d)
%%CBd2E	SRA (XY+d)
%%CBd3E	SRL (XY+d)
%%CBd46	BIT 0,(XY+d)
%%CBd4E	BIT 1,(XY+d)
%%CBd56	BIT 2,(XY+d)
%%CBd5E	BIT 3,(XY+d)
%%CBd66	BIT 4,(XY+d)
%%CBd6E	BIT 5,(XY+d)
%%CBd76	BIT 6,(XY+d)
%%CBd7E	BIT 7,(XY+d)
%%CBd86	RES 0,(XY+d)
%%CBd8E	RES 1,(XY+d)
%%CBd96	RES 2,(XY+d)
%%CBd9E	RES 3,(XY+d)
%%CBdA6	RES 4,(XY+d)
%%CBdAE	RES 5,(XY+d)
%%CBdB6	RES 6,(XY+d)
%%CBdBE	RES 7,(XY+d)
%%CBdC6	SET 0,(XY+d)
%%CBdCE	SET 1,(XY+d)
%%CBdD6	SET 2,(XY+d)
%%CBdDE	SET 3,(XY+d)
%%CBdE6	SET 4,(XY+d)
%%CBdEE	SET 5,(XY+d)
%%CBdF6	SET 6,(XY+d)
%%CBdFE	SET 7,(XY+d)
%%E1	POP XY
%%E3	EX (SP),XY
%%E5	PUSH XY
%%E9	JP (XY)
%%F9	LD SP,XY

Hex and Decimal Conversion

LSD ➔

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	2
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	3
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	4
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	5
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	6
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	7
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	8
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	9
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	A
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	B
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	C
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	D
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	E
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Powers of Two

1	2	9	512
2	4	10	1.024
3	8	11	2.048
4	16	12	4.096
5	32	13	8.192
6	64	14	16.384
7	128	15	32.768
8	256	16	65.536

17	131.072
18	262.144
19	524.285
20	1.048.576
21	2.097.152
22	4.194.304
23	8.388.608
24	16.777.216

Unsigned Comparisons

example: CP B

A < B	JP C,YES
A ⩽ B	JP C,YES JP Z,YES ①
A = B	JP Z,YES
A ≠ B	JP NZ,YES
A ⩾ B	JP NC,YES
A > B	JR C,3 ① JP NZ,YES

YES represents label for code to be executed if condition is true. Internally, A-B is computed to determine flags as for 'SUB B'.

① Requires both instructions.

Status Flags

MSB LSB

S	Z	-	H	-	P/V	N	C

S = Sign (MSB) of result
Z = 1 when result is Zero
H = Half carry from bit 3
P/V = 1 = Parity even for logic op or oVerflow for arithmetic op
N = 1 when last op was substract (0 for add)
C = Carry (CY)

Interrupts and Reset

Falling edge sensitive $\overline{\text{NMI}}$ does a RST 66H regardless of IFF1, 2 (Interrupt Flip Flop)

If interrupts are enabled (IFF1=1), low level sensitive $\overline{\text{INT}}$ depends on mode:

MODE 0: Interripting device puts instruction on bus (e.g. RST or CALL). Takes 2 extra time states.

MODE 1: Does a RST 38H (Z13).

MODE 2: Location pointed to by

15	87	10
I	ivi	0

and next hold vector of service subroutine. ivi (7 bit int vector index) is put on data bus by interrupting device (Z19).

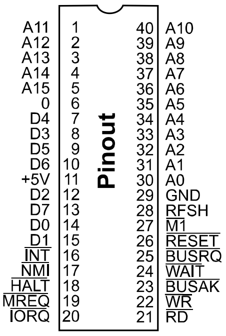
IFF1 and IFF2 are both cleared by $\overline{\text{INT}}$ or DI. Both are set by EI $\overline{\text{NMI}}$ clears IFF1. RETN loads IFF1 from IFF2. LD A,I and LD A,R set P/V flag to IFF2. Reset sets PC=0, IFF1=IFF2=0, I=0, R=0, MODE=0

General Instruction Description (except shifts)

ADC x, y	Add y+CY to x.
ADD x, y	Add y to x.
AND x	AND x to A.
BIT b, x	Test bit b of x.
CALL c, x	If condition c is true call subroutine at x.
CALL x	Call subroutine at x (push PC and jump to x).
CCF	Complement carry flag.
CP x	Compare A with x (see "Unsigned Comparisons").
CPD	Compare A with (HL); DEC HL; DEC BC.
CPDR	Like CPD, but repeat until A=(HL) or BC=0.
CPI	Compare A with (HL); INC HL; DEC BC.
CPIR	Like CPI, but repeat until A=(HL) or BC=0.
CPL	Complement A (1's comp.).
DAA	Decimal adjust A (after add or sub of BCD data).
DEC x	Decrement x by 1.
DI	Disable interrupts.
DJNZ d	Decrement B; jump relative by d if B not zero.
EI	Enable interrupts after next instruction.
EX x, y	Exchange x with y.
EXX	Exchange BC, DE, HL with BC', DE', HL'.
HALT	Halt (wait for interrupt or reset).
IM x	Set interrupt mode to x.
IN A, (n)	Input port n into A (6).
IN r, (C)	Input port (C) into r (7).
INC x	Increment x by 1.
IND	Load (HL) from port (C); DEC B; DEC HL; (7).
INDR	Like IND, but repeat until B=0 (7).
INI	Load (HL) from port (C); DEC B; INC HL; (7).
INIR	Like INI, but repeat until B=0 (7).
JP c, x	If condition c is true jump to location x.
JP x	Jump to location x.
JR c, d	If condition c is true jump relative by d.
JR d	Jump relative by d.
LD x, y	Load x with y (move y to x).
LDD	Load (DE) with (HL); DEC DE; DEC HL; DEC BC.
LDDR	Like LDD, but repeat until BC=0.
LDI	Load (DE) with (HL); INC DE; INC HL; DEC BC.
LDIR	Like LDI, but repeat until BC=0.
NEG	Negate A (2's comp.).
NOP	No operation.
OR x	OR x to A.
OTDR	Like OUTD, but repeat until B=0 (7).
OTIR	Like OUTI, but repeat until B=0 (7).
OUT (C), r	Output r to port (C) (7).
OUT (n), A	Output A to port n (7).
OUTD	Output (HL) to port (C); DEC B; DEC HL; (7).
OUTI	Output (HL) to port (C); DEC B; INC HL; (7).
POP x	Pop x from top of stack updating SP.
PUSH x	Push x onto top of stack updating SP.
RES b, x	Reset bit b of x (to 0).
RET	Return from subroutine (pop PC).
RET c	If condition c is true return from subroutine.
RETI	Return from interrupt.
RETN	Return from NMI (see "Interrupts").
RST x	Call subroutine at x (1 byte inst).
SBC x, y	Subtrack y+CY from x.
SCF	Set carry flag (to 1).
SET b, x	Set bit b of x (to 1).
SUB x	Subtract x from A.
XOR x	XOR x to A.

ASCII Character Set

MSD / LSD		0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	0000	NUL	DLE	SP	0	@	P	`	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	,	<	L	\	l	|
D	1101	CR	GS	-	=	M	]	m	}
E	1110	SO	RS	.	>	N	^	n	~
F	1111	SI	US	/	?	O	_	o	DEL



Registers

main		alternate		special	
A	F	A'	F'	I	R
B	C	B'	C'	INDEX IX	
D	E	D'	E'	INDEX IY	
H	L	H'	L'	STCK PTR SP	
small=8 bit		large=16 bit		PGRM CTR PC	

A=Accumulator
F=Flags
I=Interrupt vector
R=Memory refresh

When AF,BC,DE,HL used as pairs A,B,D,H are high order.

MICRO LOGIC CORP.

HACKENSACK, NJ

Z80 CPU

MICROPROCESSOR INSTANT REFERENCE CARD

MICRO® CHART

Example of reading instruction set tables: ADC A,A ... ADC A,- entry says to see table; table shows opcode 8F; 4 states; and flag code 'A' which is defined under 'Flag Codes'. ADC HL,BC ... 2 byte opcode is ED,4A; flag code is H; takes 15 states. CALL C, address ... opcode is DC followed by 2 byte address; flag code is Z; states are described by note 5.

Instruction Set

A	ADC	A,-	TABLE	A
	ADC	HL,BC	ED4A	H15
	ADC	HL,DE	ED5A	H15
	ADC	HL,HL	ED6A	H15
	ADC	HL,SP	ED7A	H15
	ADD	A,-	TABLE	A
	ADD	HL,BC	09	G11
	ADD	HL,DE	19	G11
	ADD	HL,HL	29	G11
	ADD	HL,SP	39	G11
	ADD	IX,BC	DD09	G15
	ADD	IX,DE	DD19	G15
	ADD	IX,IX	DD29	G15
	ADD	IX,SP	DD39	G15
	ADD	IY,BC	FD09	G15
	ADD	IY,DE	FD19	G15
	ADD	IY,IY	FD29	G15
	ADD	IY,SP	FD39	G15
	AND	-	TABLE	C
	BIT	-	TABLE	V
C	CALL	aa	CDaa	Z17
	CALL	C,aa	DCaa	Z(5)
	CALL	M,aa	FCaa	Z(5)
	CALL	NC,aa	D4aa	Z(5)
	CALL	NZ,aa	C4aa	Z(5)
	CALL	P,aa	F4aa	Z(5)
	CALL	PE,aa	ECaa	Z(5)
	CALL	PO,aa	E4aa	Z(5)
	CALL	Z,aa	CCaa	Z(5)
	CCF		3F	G4
	CP	-	TABLE	B
	CPD		EDA9	T16
	CPDR		EDB9	T(1)
	CPI		EDA1	T16
	CPIR		EDB1	T(1)
	CPL		2F	N4
D	DAA		27	M4
	DEC	(HL)	35	F11
	DEC	(IX+d)	DD35d	F23
	DEC	(IY+d)	FD35d	F23
	DEC	A	3D	F4
	DEC	B	05	F4
	DEC	BC	0B	Z6
	DEC	C	0D	F4
	DEC	D	15	F4
	DEC	DE	1B	Z6
	DEC	E	1D	F4
	DEC	H	25	F4
	DEC	HL	2B	Z6
	DEC	IX	DD2B	Z10
	DEC	IY	FD2B	Z10
	DEC	L	2D	F4
	DEC	SP	3B	Z6
	DI		F3	Z4
	DJNZ	d	10d	Z(2)
E	EI		FB	Z4
	EX	(SP),HL	E3	Z19
	EX	(SP),IX	DDE3	Z23
	EX	(SP),IY	FDE3	Z23
	EX	AF,AF'	08	Z4
	EX	DE,HL	EB	Z4
	EXX		D9	Z4
	HALT		76	Z4
I	IM	0	ED46	Z8
	IM	1	ED56	Z8
	IM	2	ED5E	Z8
	IN	A,(C)	ED78	W12
	IN	A,(n)	DBn	Z11
	IN	B,(C)	ED40	W12
	IN	C,(C)	ED48	W12
	IN	D,(C)	ED50	W12
	IN	E,(C)	ED58	W12
	IN	H,(C)	ED60	W12
	IN	L,(C)	ED68	W12
	INC	(HL)	34	E11
	INC	(IX+d)	DD34d	E23
	INC	(IY+d)	FD34d	E23
	INC	A	3C	E4
	INC	B	04	E4
	INC	BC	03	Z6
	INC	C	0C	E4
	INC	D	14	E4
	INC	DE	13	Z6
	INC	E	1C	E4
	INC	H	24	E4
	INC	HL	23	Z6
	INC	IX	DD23	Z10
	INC	IY	FD23	Z10
	INC	L	2C	E4
	INC	SP	33	Z6
	IND		EDAA	P16
	INDR		EDBA	Q(1)
	INI		EDA2	P16
	INIR		EDB2	Q(1)
J	JP	(HL)	E9	Z4
	JP	(IX)	DDE9	Z8
	JP	(IY)	FDE9	Z8
	JP	aa	C3aa	Z10
	JP	C,aa	DAaa	Z10
	JP	M,aa	FAaa	Z10
	JP	NC,aa	D2aa	Z10
	JP	NZ,aa	C2aa	Z10
	JP	P,aa	F2aa	Z10
	JP	PE,aa	EAaa	Z10
	JP	PO,aa	E2aa	Z10
	JP	Z,aa	CAaa	Z10
	JR	C,d	38d	Z(3)
	JR	d	18d	Z12
	JR	NC,d	30d	Z(3)
	JR	NZ,d	20d	Z(3)
	JR	Z,d	28d	Z(3)
L	LD	(BC),A	02	Z7
	LD	(DE),A	12	Z7
	LD	(HL),A	77	Z7
	LD	(HL),B	70	Z7
	LD	(HL),C	71	Z7
	LD	(HL),D	72	Z7
	LD	(HL),E	73	Z7
	LD	(HL),H	74	Z7
	LD	(HL),L	75	Z7
	LD	(HL),n	36n	Z10
	LD	(IX+d),A	DD77d	Z19
	LD	(IX+d),B	DD70d	Z19

L	LD	(IX+d),C	DD71d	Z19
	LD	(IX+d),D	DD72d	Z19
	LD	(IX+d),E	DD73d	Z19
	LD	(IX+d),H	DD74d	Z19
	LD	(IX+d),L	DD75d	Z19
	LD	(IX+d),n	DD36dn	Z19
	LD	(IY+d),A	FD77d	Z19
	LD	(IY+d),B	FD70d	Z19
	LD	(IY+d),C	FD71d	Z19
	LD	(IY+d),D	FD72d	Z19
	LD	(IY+d),E	FD73d	Z19
	LD	(IY+d),H	FD74d	Z19
	LD	(IY+d),L	FD75d	Z19
	LD	(IY+d),n	FD36dn	Z19
	LD	(aa),A	32aa	Z13
	LD	(aa),BC	ED43aa	Z20
	LD	(aa),DE	ED53aa	Z20
	LD	(aa),HL	22aa	Z16
	LD	(aa),IX	DD22aa	Z20
	LD	(aa),IY	FD22aa	Z20
	LD	(aa),SP	ED73aa	Z20
	LD	A,(BC)	0A	Z7
	LD	A,(DE)	1A	Z7
	LD	A,(aa)	3Aaa	Z13
	LD	A,I	ED57	U9
	LD	A,R	ED5F	U9
	LD	A,-	TABLE	Z
	LD	B,-	TABLE	Z
	LD	BC,(aa)	ED4Baa	Z20
	LD	BC,aa	01aa	Z10
	LD	C,-	TABLE	Z
	LD	D,-	TABLE	Z
	LD	DE,(aa)	ED5Baa	Z20
	LD	DE,aa	11aa	Z10
	LD	E,-	TABLE	Z
	LD	H,-	TABLE	Z
	LD	HL,(aa)	2Aaa	Z16
	LD	HL,aa	21aa	Z10
	LD	I,A	ED47	Z9
	LD	IX,(aa)	DD2Aaa	Z20
	LD	IX,aa	DD21aa	Z14
	LD	IY,(aa)	FD2Aaa	Z20
	LD	IY,aa	FD21aa	Z14
	LD	L,-	TABLE	Z
	LD	R,A	ED4F	Z9
	LD	SP,(aa)	ED7Baa	Z20
	LD	SP,HL	F9	Z6
	LD	SP,IX	DDF9	Z10
	LD	SP,IY	FDF9	Z10
	LD	SP,aa	31aa	Z10
	LDD		EDA8	R16
	LDDR		EDB8	S(1)
	LDI		EDA0	R16
	LDIR		EDB0	S(1)
	NEG		ED44	B8
	NOP		00	Z4
O	OR	-	TABLE	D
	OTDR		EDBB	Q(1)
	OTIR		EDB3	Q(1)
	OUT	(C),A	ED79	Z12
	OUT	(C),B	ED41	Z12
	OUT	(C),C	ED49	Z12
	OUT	(C),D	ED51	Z12
	OUT	(C),E	ED59	Z12
	OUT	(C),H	ED61	Z12
	OUT	(C),L	ED69	Z12
	OUT	(n),A	D3n	Z11
	OUTD		EDAB	P16
	OUTI		EDA3	P16
P	POP	AF	F1	Z10
	POP	BC	C1	Z10
	POP	DE	D1	Z10
	POP	HL	E1	Z10
	POP	IX	DDE1	Z14
	POP	IY	FDE1	Z14
	PUSH	AF	F5	Z11
	PUSH	BC	C5	Z11
	PUSH	DE	D5	Z11
	PUSH	HL	E5	Z11
	PUSH	IX	DDE5	Z15
	PUSH	IY	FDE5	Z15
R	RES	-	TABLE	Z
	RET		C9	Z10
	RET	C	D8	Z(4)
	RET	M	F8	Z(4)
	RET	NC	D0	Z(4)
	RET	NZ	C0	Z(4)
	RET	P	F0	Z(4)
	RET	PE	E8	Z(4)
	RET	PO	E0	Z(4)
	RET	Z	C8	Z(4)
	RETI		ED4D	Z14
	RETN		ED45	Z14
	RL	-	TABLE	K
	RLA		17	J4
	RLC	-	TABLE	K
	RLCA		07	J4
	RLD		ED6F	L18
	RR	-	TABLE	K
	RRA		1F	J4
	RRC	-	TABLE	K
	RRCA		0F	J4
	RRD		ED67	L18
	RST	00H	C7	Z11
	RST	08H	CF	Z11
	RST	10H	D7	Z11
	RST	18H	DF	Z11
	RST	20H	E7	Z11
	RST	28H	EF	Z11
	RST	30H	F7	Z11
	RST	38H	FF	Z11
S	SBC	A,-	TABLE	B
	SBC	HL,BC	ED42	I15
	SBC	HL,DE	ED52	I15
	SBC	HL,HL	ED62	I15
	SBC	HL,SP	ED72	I15
	SCF		37	O4
	SET	-	TABLE	Z
	SLA	-	TABLE	K
	SRA	-	TABLE	K
	SRL	-	TABLE	K
	SUB	-	TABLE	B
	XOR	-	TABLE	D

	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)	
BIT 0,	CB,47	CB,40	CB,41	CB,42	CB,43	CB,44	CB,45	CB,46	DD,CB,d,46	FD,CB,d,46	V
BIT 1,	CB,4F	CB,48	CB,49	CB,4A	CB,4B	CB,4C	CB,4D	CB,4E	DD,CB,d,4E	FD,CB,d,4E	V
BIT 2,	CB,57	CB,50	CB,51	CB,52	CB,53	CB,54	CB,55	CB,56	DD,CB,d,56	FD,CB,d,56	V
BIT 3,	CB,5F	CB,58	CB,59	CB,5A	CB,5B	CB,5C	CB,5D	CB,5E	DD,CB,d,5E	FD,CB,d,5E	V
BIT 4,	CB,67	CB,60	CB,61	CB,62	CB,63	CB,64	CB,65	CB,66	DD,CB,d,66	FD,CB,d,66	V
BIT 5,	CB,6F	CB,68	CB,69	CB,6A	CB,6B	CB,6C	CB,6D	CB,6E	DD,CB,d,6E	FD,CB,d,6E	V
BIT 6,	CB,77	CB,70	CB,71	CB,72	CB,73	CB,74	CB,75	CB,76	DD,CB,d,76	FD,CB,d,76	V
BIT 7,	CB,7F	CB,78	CB,79	CB,7A	CB,7B	CB,7C	CB,7D	CB,7E	DD,CB,d,7E	FD,CB,d,7E	V
STATES:	8							12	20		

	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)	
RES 0,	CB,87	CB,80	CB,81	CB,82	CB,83	CB,84	CB,85	CB,86	DD,CB,d,86	FD,CB,d,86	Z
RES 1,	CB,8F	CB,88	CB,89	CB,8A	CB,8B	CB,8C	CB,8D	CB,8E	DD,CB,d,8E	FD,CB,d,8E	Z
RES 2,	CB,97	CB,90	CB,91	CB,92	CB,93	CB,94	CB,95	CB,96	DD,CB,d,96	FD,CB,d,96	Z
RES 3,	CB,9F	CB,98	CB,99	CB,9A	CB,9B	CB,9C	CB,9D	CB,9E	DD,CB,d,9E	FD,CB,d,9E	Z
RES 4,	CB,A7	CB,A0	CB,A1	CB,A2	CB,A3	CB,A4	CB,A5	CB,A6	DD,CB,d,A6	FD,CB,d,A6	Z
RES 5,	CB,AF	CB,A8	CB,A9	CB,AA	CB,AB	CB,AC	CB,AD	CB,AE	DD,CB,d,AE	FD,CB,d,AE	Z
RES 6,	CB,B7	CB,B0	CB,B1	CB,B2	CB,B3	CB,B4	CB,B5	CB,B6	DD,CB,d,B6	FD,CB,d,B6	Z
RES 7,	CB,BF	CB,B8	CB,B9	CB,BA	CB,BB	CB,BC	CB,BD	CB,BE	DD,CB,d,BE	FD,CB,d,BE	Z
SET 0,	CB,C7	CB,C0	CB,C1	CB,C2	CB,C3	CB,C4	CB,C5	CB,C6	DD,CB,d,C6	FD,CB,d,C6	Z
SET 1,	CB,CF	CB,C8	CB,C9	CB,CA	CB,CB	CB,CC	CB,CD	CB,CE	DD,CB,d,CE	FD,CB,d,CE	Z
SET 2,	CB,D7	CB,D0	CB,D1	CB,D2	CB,D3	CB,D4	CB,D5	CB,D6	DD,CB,d,D6	FD,CB,d,D6	Z
SET 3,	CB,DF	CB,D8	CB,D9	CB,DA	CB,DB	CB,DC	CB,DD	CB,DE	DD,CB,d,DE	FD,CB,d,DE	Z
SET 4,	CB,E7	CB,E0	CB,E1	CB,E2	CB,E3	CB,E4	CB,E5	CB,E6	DD,CB,d,E6	FD,CB,d,E6	V
SET 5,	CB,EF	CB,E8	CB,E9	CB,EA	CB,EB	CB,EC	CB,ED	CB,EE	DD,CB,d,EE	FD,CB,d,EE	V
SET 6,	CB,F7	CB,F0	CB,F1	CB,F2	CB,F3	CB,F4	CB,F5	CB,F6	DD,CB,d,F6	FD,CB,d,F6	V
SET 7,	CB,FF	CB,F8	CB,F9	CB,FA	CB,FB	CB,FC	CB,FD	CB,FE	DD,CB,d,FE	FD,CB,d,FE	V
STATES:	8							15	23		

	A (8)	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)	
RLC	CB,07	CB,00	CB,01	CB,02	CB,03	CB,04	CB,05	CB,06	DD,CB,d,06	FD,CB,d,06	K
RRC	CB,0F	CB,08	CB,09	CB,0A	CB,0B	CB,0C	CB,0D	CB,0E	DD,CB,d,0E	FD,CB,d,0E	K
RL	CB,17	CB,10	CB,11	CB,12	CB,13	CB,14	CB,15	CB,16	DD,CB,d,16	FD,CB,d,16	K
RR	CB,1F	CB,18	CB,19	CB,1A	CB,1B	CB,1C	CB,1D	CB,1E	DD,CB,d,1E	FD,CB,d,1E	K
SLA	CB,27	CB,20	CB,21	CB,22	CB,23	CB,24	CB,25	CB,26	DD,CB,d,26	FD,CB,d,26	K
SRA	CB,2F	CB,28	CB,29	CB,2A	CB,2B	CB,2C	CB,2D	CB,2E	DD,CB,d,2E	FD,CB,d,2E	K
SRL	CB,37	CB,38	CB,39	CB,3A	CB,3B	CB,3C	CB,3D	CB,3E	DD,CB,d,3E	FD,CB,d,3E	K
STATES:	8							15	23		

Flag Codes

	C	Z	P V	S	N	H	
A	C	Z	V	S	0	H	
B	C	Z	V	S	1	H	
C	0	Z	P	S	0	1	
D	0	Z	P	S	0	0	
E	=	Z	V	S	0	H	
F	=	Z	V	S	1	H	
G	C	=	=	=	0	U	
H	C	Z	V	S	0	U	
I	C	Z	V	S	1	U	
J	C	=	=	=	0	0	
K	C	Z	P	S	0	0	
L	=	Z	P	S	0	0	
M	C	Z	P	S	=	H	
N	=	=	=	=	1	1	
O	1	=	=	=	0	0	
P	U	F	U	U	1	U	(1)
Q	U	1	U	U	1	U	
R	=	U	F	U	0	0	(2)
S	=	U	0	U	0	0	
T	=	F	F	S	1	U	(3)
U	=	Z	F	S	0	0	(4)
V	=	Z	U	U	0	1	(5)
W	=	Z	P	S	0	U	
X							
Y							
Z	=	=	=	=	=	=	

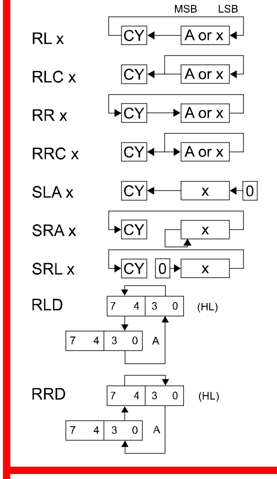
Codes:
0: reset
1: set
C: Carry*
F: Footnote
H: Half carry
N: Add/Sub*
P: Parity*
S: Sign*
U: Undefined
V: oVerflow*
Z: Zero*
=: not affected

* Indicated flag affected by result

(1) Z=1 if B becomes 0
(2) PV=0 if BC becomes 0
(3) PV=0 if BC becomes 0 and Z=1 if A=(HL)
(4) PV=IFF2
(5) Z=$\overline{\text{bit}}$

	A	B	C	D	E	H	L	(HL)	n	(IX+d)	(IY+d)	
ADC A,	8F	88	89	8A	8B	8C	8D	8E	CE,n	DD,8E,d	FD,8E,d	A
ADD A,	87	80	81	82	83	84	85	86	C6,n	DD,86,d	FD,86,d	A
AND	A7	A0	A1	A2	A3	A4	A5	A6	E6,n	DD,A6,d	FD,A6,d	C
CP	BF	B8	B9	BA	BB	BC	BD	BE	FE,n	DD,BE,d	FD,BE,d	B
OR	B7	B0	B1	B2	B3	B4	B5	B6	F6,n	DD,B6,d	FD,B6,d	D
SBC A,	9F	98	99	9A	9B	9C	9D	9E	DE,n	DD,9E,d	FD,9E,d	B
SUB	97	90	91	92	93	94	95	96	D6,n	DD,96,d	FD,96,d	B
XOR	AF	A8	A9	AA	AB	AC	AD	AE	EE,n	DD,AE,d	FD,AE,d	D
LD A,	7F	78	79	7A	7B	7C	7D	7E	3E,n	DD,7E,d	FD,7E,d	Z
LD B,	47	40	41	42	43	44	45	46	06,n	DD,46,d	FD,46,d	Z
LD C,	4F	48	49	4A	4B	4C	4D	4E	0E,n	DD,4E,d	FD,4E,d	Z
LD D,	57	50	51	52	53	54	55	56	16,n	DD,56,d	FD,56,d	Z
LD E,	5F	58	59	5A	5B	5C	5D	5E	1E,n	DD,5E,d	FD,5E,d	Z
LD H,	67	60	61	62	63	64	65	66	26,n	DD,66,d	FD,66,d	Z
LD L,	6F	68	69	6A	6B	6C	6D	6E	2E,n	DD,6E,d	FD,6E,d	Z
STATES:	4								7	19		

Rotates and Shifts



Notes

(1) 21 except 16 at termination
(2) 13 except 8 at termination
(3) 12 for success; 7 for failure
(4) 11 for success; 5 for failure
(5) 17 for success; 10 for failure
(6) A to A15..A8 and n to A7..A0
(7) B to A15..A8 and C to A7..A0
(8) See faster version of 'Rotate A' instructions

Addressing

n	n is immediate 8-bit data.
aa	aa is immediate 16-bit data or address to CALL/to JP to.
(aa)	aa is address of data.
(rr)	16-bit reg rr holds address of data or address to CALL or to JP to.
(n)	n is port number.
(r)	8-bit reg r holds port number.
(IX+d)	IX+d is address of data (d is a 1 byte signed displacement).
d	In relative jumping, address to jump to is d + address of next instruction (d is signed).

Full 2 byte addresses in code, stack, and data areas are stored low byte followed by high byte. Thus JP 1234H is: C3,34,12.

SP points to used byte at top of stack. PUSH decrements SP by 2.

Author: James D. Lewis
Micro Logic Corp (Ⓒ1981)
SVG version by RetroParla (2019)

RETRO PARLA
ASOCIACION DE AFICIONADOS A LA RETROINFORMATICA DE PARLA
www.retroparla.com

MICRO LOGIC CORP.

HACKENSACK, NJ

Z80 CPU

MICROPROCESSOR INSTANT REFERENCE CARD

MICRO® CHART

Single-Byte-Opcode to Instruction Conversion

LSD ⇒

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	NOP	LD BC,nn	LD (BC),A	INC BC	INC B	DEC B	LD B,n	RLCA	EX AF,AF'	ADD HL,BC	LD A,(BC)	DEC BC	INC C	DEC C	LD C,n	RRCA	0
1	DJNZ n	LD DE,nn	LD (DE),A	INC DE	INC D	DEC D	LD D,n	RLA	JR n	ADD HL,DE	LD A,(DE)	DEC DE	INC E	DEC E	LD E,n	RRA	1
2	JR NZ,n	LD HL,nn	LD (nn),HL	INC HL	INC H	DEC H	LD H,n	DAA	JR Z,n	ADD HL,HL	LD HL,(nn)	DEC HL	INC L	DEC L	LD L,n	CPL	2
3	JR NC,n	LD SP,nn	LD (nn),A	INC SP	INC (HL)	DEC (HL)	LD (HL),n	SCF	JR C,n	ADD HL,SP	LD A,(nn)	DEC SP	INC A	DEC A	LD A,n	CCF	3
4	LD B,B	LD B,C	LD B,D	LD B,E	LD B,H	LD B,L	LD B,(HL)	LD B,A	LD C,B	LD C,C	LD C,D	LD C,E	LD C,H	LD C,L	LD C,(HL)	LD C,A	4
5	LD D,B	LD D,C	LD D,D	LD D,E	LD D,H	LD D,L	LD D,(HL)	LD D,A	LD E,B	LD E,C	LD E,D	LD E,E	LD E,H	LD E,L	LD E,(HL)	LD E,A	5
6	LD H,B	LD H,C	LD H,D	LD H,E	LD H,H	LD H,L	LD H,(HL)	LD H,A	LD L,B	LD L,C	LD L,D	LD L,E	LD L,H	LD L,L	LD L,(HL)	LD L,A	6
7	LD (HL),B	LD (HL),C	LD (HL),D	LD (HL),E	LD (HL),H	LD (HL),L	HALT	LD (HL),A	LD A,B	LD A,C	LD A,D	LD A,E	LD A,H	LD A,L	LD A,(HL)	LD A,A	7
8	ADD A,B	ADD A,C	ADD A,D	ADD A,E	ADD A,H	ADD A,L	ADD A,(HL)	ADD A,A	ADC A,B	ADC A,C	ADC A,D	ADC A,E	ADC A,H	ADC A,L	ADC A,(HL)	ADC A,A	8
9	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB (HL)	SUB A	SBC A,B	SBC A,C	SBC A,D	SBC A,E	SBC A,H	SBC A,L	SBC A,(HL)	SBC A,A	9
A	AND B	AND C	AND D	AND E	AND H	AND L	AND (HL)	AND A	XOR B	XOR C	XOR D	XOR E	XOR H	XOR L	XOR (HL)	XOR A	A
B	OR B	OR C	OR D	OR E	OR H	OR L	OR (HL)	OR A	CP B	CP C	CP D	CP E	CP H	CP L	CP (HL)	CP A	B
C	RET NZ	POP BC	JP NZ,nn	JP nn	CALL NZ,nn	PUSH BC	ADD A,n	RST 00H	RET Z	RET	JP Z,nn	table	CALL Z,nn	CALL nn	ADC A,n	RST 08H	C
D	RET NC	POP DE	JP NC,nn	OUT (n), A	CALL NC,nn	PUSH DE	SUB n	RST 10H	RET C	EXX	JP C,nn	IN A,(n)	CALL C,nn	table	SBC A,n	RST 18H	D
E	RET PO	POP HL	JP PO,nn	EX (SP), HL	CALL PO,nn	PUSH HL	AND n	RST 20H	RET PE	JP (HL)	JP PE,nn	EX DE,HL	CALL PE,nn	table	XOR n	RST 28H	E
F	RET P	POP AF	JP P,nn	DI	CALL P,nn	PUSH AF	OR n	RST 30H	RET M	LD SP,HL	JP M,nn	EI	CALL M,nn	table	CP n	RST 38H	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Multi-Byte-Opcode to Instruction Conversion

CB00	RLC B	ED40	IN B,(C)	%%09	ADD XY,BC	%%CBd06	RLC (XY+d)
CB01	RLC C	ED41	OUT (C),B	%%19	ADD XY,DE	%%CBd0E	RRC (XY+d)
CB02	RLC D	ED42	SBC HL,BC	%%21aa	LD XY,aa	%%CBd16	RL (XY+d)
CB03	RLC E	ED43aa	LD (aa),BC	%%22aa	LD (aa),XY	%%CBd1E	RR (XY+d)
CB04	RLC H	ED44	NEG	%%23	INC XY	%%CBd26	SLA (XY+d)
CB05	RLC L	ED45	RETN	%%29	ADD XY,XY	%%CBd2E	SRA (XY+d)
CB06	RLC (HL)	ED46	IM 0	%%2Aaa	LD XY,(aa)	%%CBd3E	SRL (XY+d)
CB07	RLC A	ED47	LD I,A	%%2B	DEC XY	%%CBd46	BIT 0,(XY+d)
CB08	RRC B	ED48	IN C,(C)	%%34d	INC (XY+d)	%%CBd4E	BIT 1,(XY+d)
CB09	RRC C	ED49	OUT (C),C	%%35d	DEC (XY+d)	%%CBd56	BIT 2,(XY+d)
CB0A	RRC D	ED4A	ADC HL,BC	%%36dn	LD (XY+d),n	%%CBd5E	BIT 3,(XY+d)
CB0B	RRC E	ED4Baa	LD BC,(aa)	%%39	ADD XY,SP	%%CBd66	BIT 4,(XY+d)
CB0C	RRC H	ED4D	RETI	%%46d	LD B,(XY+d)	%%CBd6E	BIT 5,(XY+d)
CB0D	RRC L	ED4F	LD R,A	%%4Ed	LD C,(XY+d)	%%CBd76	BIT 6,(XY+d)
CB0E	RRC (HL)	ED50	IN D,(C)	%%56d	LD D,(XY+d)	%%CBd7E	BIT 7,(XY+d)
CB0F	RRC A	ED51	OUT (C),D	%%5Ed	LD E,(XY+d)	%%CBd86	RES 0,(XY+d)
CB10	RL B	ED52	SBC HL,DE	%%66d	LD H,(XY+d)	%%CBd8E	RES 1,(XY+d)
CB11	RL C	ED53aa	LD (aa),DE	%%6Ed	LD L,(XY+d)	%%CBd96	RES 2,(XY+d)
CB12	RL D	ED56	IM 1	%%70d	LD (XY+d),B	%%CBd9E	RES 3,(XY+d)
CB13	RL E	ED57	LD A,I	%%71d	LD (XY+d),C	%%CBdA6	RES 4,(XY+d)
CB14	RL H	ED58	IN E,(C)	%%72d	LD (XY+d),D	%%CBdAE	RES 5,(XY+d)
CB15	RL L	ED59	OUT (C),E	%%73d	LD (XY+d),E	%%CBdB6	RES 6,(XY+d)
CB16	RL (HL)	ED5A	ADC HL,DE	%%74d	LD (XY+d),H	%%CBdBE	RES 7,(XY+d)
CB17	RL A	ED5Baa	LD DE,(aa)	%%75d	LD (XY+d),L	%%CBdC6	SET 0,(XY+d)
CB18	RR B	ED5E	IM 2	%%77d	LD (XY+d),A	%%CBdCE	SET 1,(XY+d)
CB19	RR C	ED5F	LD A,R	%%7Ed	LD A,(XY+d)	%%CBdD6	SET 2,(XY+d)
CB1A	RR D	ED60	IN H,(C)	%%86d	ADD A,(XY+d)	%%CBdDE	SET 3,(XY+d)
CB1B	RR E	ED61	OUT (C),H	%%8Ed	ADC A,(XY+d)	%%CBdE6	SET 4,(XY+d)
CB1C	RR H	ED62	SBC HL,HL	%%96d	SUB (XY+d)	%%CBdEE	SET 5,(XY+d)
CB1D	RR L	ED67	RRD	%%9Ed	SBC A,(XY+d)	%%CBdF6	SET 6,(XY+d)
CB1E	RR (HL)	ED68	IN L,(C)	%%A6d	AND (XY+d)	%%CBdFE	SET 7,(XY+d)
CB1F	RR A	ED69	OUT (C),L	%%AEd	XOR (XY+d)		
CB20	SLA B	ED6A	ADC HL,HL	%%B6d	OR (XY+d)		
CB21	SLA C	ED6F	RLD	%%BEd	CP (XY+d)	%%E1	POP XY
CB22	SLA D	ED72	SBC HL,SP			%%E3	EX (SP),XY
CB23	SLA E	ED73aa	LD (aa),SP			%%E5	PUSH XY
CB24	SLA H	ED78	IN A,(C)			%%E9	JP (XY)
CB25	SLA L	ED79	OUT (C),A			%%F9	LD SP,XY
CB26	SLA (HL)	ED7A	ADC HL,SP				
CB27	SLA A	ED7Baa	LD SP,(aa)				
CB28	SRA B	EDA0	LDI				
CB29	SRA C	EDA1	CPI				
CB2A	SRA D	EDA2	INI				
CB2B	SRA E	EDA3	OUTI				
CB2C	SRA H	EDA8	LDD				
CB2D	SRA L	EDA9	CPD				
CB2E	SRA (HL)	EDAA	IND				
CB2F	SRA A	EDAB	OUTD				
CB38	SRL B	EDB0	LDIR				
CB39	SRL C	EDB1	CPIR				
CB3A	SRL D	EDB2	INIR				
CB3B	SRL E	EDB3	OTIR				
CB3C	SRL H	EDB8	LDDR				
CB3D	SRL L	EDB9	CPDR				
CB3E	SRL (HL)	EDBA	INDR				
CB3F	SRL A	EDBB	OTDR				
CB40	see BIT						
....	see RES						
CBFF	see SET						

%% means DD or FD and
for DD, XY means IX
for FD, XY means IY

Hex and Decimal Conversion

LSD →

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	2
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	3
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	4
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	5
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	6
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	7
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	8
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	9
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	A
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	B
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	C
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	D
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	E
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Powers of Two

1	2	9	512
2	4	10	1,024
3	8	11	2,048
4	16	12	4,096
5	32	13	8,192
6	64	14	16,384
7	128	15	32,768
8	256	16	65,536
17	131,072		
18	262,144		
19	524,288		
20	1,048,576		
21	2,097,152		
22	4,194,304		
23	8,388,608		
24	16,777,216		

Unsigned Comparisons

example: CP B

A < B	JP C,YES
A ⩽ B	JP C,YES ① JP Z,YES
A = B	JP Z,YES
A ≠ B	JP NZ,YES
A ⩾ B	JP NC,YES
A > B	JR C,3 ① JP NZ,YES

YES represents label for code to be executed if condition is true. Internally, A-B is computed to determine flags as for 'SUB B'.

① Requires both instructions.

Status Flags

MSB							LSB
S	Z	-	H	-	P/V	N	C

S = Sign (MSB) of result
Z = 1 when result is Zero
H = Half carry from bit 3
P/V = 1 = Parity even for logic op or oVerflow for arithmetic op
N = 1 when last op was subtract (0 for add)
C = Carry (CY)

Interrupts and Reset

Falling edge sensitive $\overline{NMI}$ does a RST 66H regardless of IFF1, 2 (Interrupt Flip Flop).

If interrupts are enabled (IFF1=1), low level sensitive $\overline{INT}$ depends on mode:
MODE 0: Interrupting device puts instruction on bus (e.g. RST or CALL). Takes 2 extra time states.
MODE 1: Does a RST 38H (Z13).
MODE 2: Location pointed to by

15	87	10
I	ivi	0

and next hold vector of service subroutine. ivi (7 bit int vector index) is put on data bus by interrupting device (Z19).

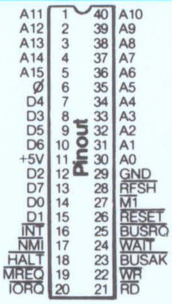
IFF1 and IFF2 are both cleared by $\overline{INT}$ or DI. Both are set by EI. $\overline{NMI}$ clears IFF1. RETN loads IFF1 from IFF2. LD A,I and LD A,R set P/V flag to IFF2. Reset sets PC=0, IFF1=IFF2=0, I=0, R=0, MODE =0.

General Instruction Description (except shifts)

ADC x, y	Add y+CY to x.
ADD x, y	Add y to x.
AND x	AND x to A.
BIT b, x	Test bit b of x.
CALL c, x	If condition c is true call subroutine at x.
CALL x	Call subroutine at x (push PC and jump to x).
CCF	Complement carry flag.
CP x	Compare A with x (see "Unsigned Comparisons")
CPD	Compare A with (HL); DEC HL; DEC BC.
CPDR	Like CPD, but repeat until A=(HL) or BC=0.
CPI	Compare A with (HL); INC HL;DEC BC.
CPIR	Like CPI, but repeat until A=(HL) or BC=0.
CPL	Complement A (1's comp.).
DAA	Decimal adjust A (after add or sub of BCD data).
DEC x	Decrement x by 1.
DI	Disable interrupts.
DJNZ d	Decrement B; jump relative by d if B not zero.
EI	Enable interrupts after next instruction.
EX x,y	Exchange x with y.
EXX	Exchange BC, DE, HL with BC', DE', HL'.
HALT	Halt (wait for interrupt or reset).
IM x	Set interrupt mode to x.
IN A, (n)	Input port n into A (6).
IN r, (C)	Input port (C) into r (7).
INC x	Increment x by 1.
IND	Load (HL) from port (C); DEC B; DEC HL; (7).
INDR	Like IND, but repeat until B=0 (7).
INI	Load (HL) from port (C); DEC B; INC HL; (7).
INIR	Like INI, but repeat until B=0 (7).
JP c, x	If condition c is true jump to location x.
JP x	Jump to location x.
JR c, d	If condition c is true jump relative by d.
JR d	Jump relative by d.
LD x, y	Load x with y (move y to x).
LDD	Load (DE) with (HL); DEC DE; DEC HL; DEC BC.
LDDR	Like LDD, but repeat until BC=0.
LDI	Load (DE) with (HL); INC DE; INC HL; DEC BC.
LDIR	Like LDI, but repeat until BC=0.
NEG	Negate A (2's comp.).
NOP	No operation.
OR x	OR x to A.
OTDR	Like OUTD, but repeat until B=0 (7).
OTIR	Like OUTI, but repeat until B=0 (7).
OUT (C), r	Output r to port (C) (7).
OUT (n), A	Output A to port n (7).
OUTD	Output (HL) to port (C); DEC B; DEC HL; (7).
OUTI	Output (HL) to port (C); DEC B; INC HL; (7).
POP x	Pop x from top of stack updating SP.
PUSH x	Push x onto top of stack updating SP.
RES b, x	Reset bit b of x (to 0).
RET	Return from subroutine (pop PC).
RET c	If condition c is true return from subroutine.
RETI	Return from interrupt.
RETN	Return from NMI (see "Interrupts").
RST x	Call subroutine at x (1 byte inst).
SBC x, y	Subtract y+CY from x.
SCF	Set carry flag (to 1).
SET b, x	Set bit b of x (to 1).
SUB x	Subtract x from A.
XOR x	XOR x to A.

ASCII Character Set

LSD \ MSD		0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	0000	NUL	DLE	SP	0	@	P	'	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	,	<	L	\	l	|
D	1101	CR	GS	-	=	M	]	m	}
E	1110	SO	RS	.	>	N	↑	n	~
F	1111	SI	US	/	?	O	←	o	DEL



Registers

main		alternate		special	
A	F	A'	F'	I	R
B	C	B'	C'	INDEX IX	
D	E	D'	E'	INDEX IY	
H	L	H'	L'	STCK PTR SP	
small=8 Bit		large=16 bit		PGRM CTR PC	

A=Accumulator
F=Flags
I= Interrupt vector
R=Memory refresh

When AF,BC,DE,HL used as pairs A,B,D,H are high order.

100% PLASTIC

MICRO CHARTS: Z80, 6502-65XX, 8080-8085, 8086-8088, 8048 Family, 54/7400 TTL pinouts, BASIC Algorithms, Wordstar, Electronic Components, Sampling Statistics, C Language.

This card is not a promotional item. Please observe our copyright and replace any copies with plastic originals.

INSTANT ACCESS

MICRO LOGIC CORP.
HACKENSACK, NJ

Z80 CPU

MICROPROCESSOR INSTANT REFERENCE CARD

MICRO® CHART

Example of reading instruction set tables: ADC A,A ... ADC A, - entry says to see table; table shows opcode 8F. 4 states; and flag code 'A' which is defined under 'Flag Codes'. ADC HL,BC ... 2 byte opcode is ED,4A; flag code is H; takes 15 states. CALL C, address ... opcode is DC followed by 2 byte address; flag code is Z; states are described by note 5.

Instruction Set

A	ADC	A,—	TABLE	A
	ADC	HL,BC	ED4A	H15
	ADC	HL,DE	ED5A	H15
	ADC	HL,HL	ED6A	H15
	ADC	HL,SP	ED7A	H15
	ADD	A,—	TABLE	A
	ADD	HL,BC	09	G11
	ADD	HL,DE	19	G11
	ADD	HL,HL	29	G11
	ADD	HL,SP	39	G11
	ADD	IX,BC	DD09	G15
	ADD	IX,DE	DD19	G15
	ADD	IX,IX	DD29	G15
	ADD	IX,SP	DD39	G15
	ADD	IY,BC	FD09	G15
	ADD	IY,DE	FD19	G15
	ADD	IY,IY	FD29	G15
	ADD	IY,SP	FD39	G15
	AND	—	TABLE	C
	BIT	—	TABLE	V
C	CALL	aa	CDaa	Z17
	CALL	C,aa	DCaa	Z(5)
	CALL	M,aa	FCaa	Z(5)
	CALL	NC,aa	D4aa	Z(5)
	CALL	NZ,aa	C4aa	Z(5)
	CALL	P,aa	F4aa	Z(5)
	CALL	PE,aa	ECaa	Z(5)
	CALL	PO,aa	E4aa	Z(5)
	CALL	Z,aa	CCaa	Z(5)
	CCF		3F	G4
	CP	—	TABLE	B
	CPD		EDA9	T16
	CPDR		EDB9	T(1)
	CPI		EDA1	T16
	CPIR		EDB1	T(1)
	CPL		2F	N4
D	DAA		27	M4
	DEC	(HL)	35	F11
	DEC	(IX+d)	DD35d	F23
	DEC	(IY+d)	FD35d	F23
	DEC	A	3D	F4
	DEC	B	05	F4
	DEC	BC	0B	Z6
	DEC	C	0D	F4
	DEC	D	15	F4
	DEC	DE	1B	Z6
	DEC	E	1D	F4
	DEC	H	25	F4
	DEC	HL	2B	Z6
	DEC	IX	DD2B	Z10
	DEC	IY	FD2B	Z10
	DEC	L	2D	F4
	DEC	SP	3B	Z6
	DI		F3	Z4
	DJNZ	d	10d	Z(2)
E	EI		FB	Z4
	EX	(SP) ,HL	E3	Z19
	EX	(SP) ,IX	DDE3	Z23
	EX	(SP) ,IY	FDE3	Z23
	EX	AF,AF'	08	Z4
	EX	DE,HL	EB	Z4
	EXX		D9	Z4
	HALT		76	Z4
I	IM	0	ED46	Z8
	IM	1	ED56	Z8
	IM	2	ED5E	Z8
	IN	A,(C)	ED78	W12
	IN	A,(n)	DBn	Z11
	IN	B,(C)	ED40	W 12
	IN	C,(C)	ED48	W 12
	IN	D,(C)	ED50	W 12
	IN	E,(C)	ED58	W 12
	IN	H,(C)	ED60	W 12
	IN	L,(C)	ED68	W 12
	INC	(HL)	34	E11
	INC	(IX+d)	DD34d	E23
	INC	(IY+d)	FD34d	E23
	INC	A	3C	E4
	INC	B	04	E4
	INC	BC	03	Z6
	INC	C	0C	E4
	INC	D	14	E4
	INC	DE	13	Z6
	INC	E	1C	E4
	INC	H	24	E4
	INC	HL	23	Z6
	INC	IX	DD23	Z10
	INC	IY	FD23	Z10
	INC	L	2C	E4
	INC	SP	33	Z6
	IND		EDAA	P16
	INDR		EDBA	Q(1)
	INI		EDA2	P16
	INIR		EDB2	Q(1)
J	JP	(HL)	E9	Z4
	JP	(IX)	DDE9	Z8
	JP	(IY)	FDE9	Z8
	JP	aa	C3aa	Z10
	JP	C,aa	DAaa	Z10
	JP	M,aa	FAaa	Z10
	JP	NC,aa	D2aa	Z10
	JP	NZ,aa	C2aa	Z10
	JP	P,aa	F2aa	Z10
	JP	PE,aa	EAaa	Z10
	JP	PO,aa	E2aa	Z10
	JP	Z,aa	CAaa	Z10
	JR	C,d	38d	Z(3)
	JR	d	18d	Z12
	JR	NC,d	30d	Z(3)
	JR	NZ,d	20d	Z(3)
	JR	Z,d	28d	Z(3)
L	LD	(BC),A	02	Z7
	LD	(DE),A	12	Z7
	LD	(HL),A	77	Z7
	LD	(HL),B	70	Z7
	LD	(HL),C	71	Z7
	LD	(HL),D	72	Z7
	LD	(HL),E	73	Z7
	LD	(HL),H	74	Z7
	LD	(HL),L	75	Z7
	LD	(HL),n	36n	Z10
	LD	(IX+d),A	DD77d	Z19
	LD	(IX+d),B	DD70d	Z19
	LD	(IX+d),C	DD71d	Z19
	LD	(IX+d),D	DD72d	Z19
	LD	(IX+d),E	DD73d	Z19
	LD	(IX+d),H	DD74d	Z19
	LD	(IX+d),L	DD75d	Z19
	LD	(IX+d),n	DD36dn	Z19
	LD	(IY+d),A	FD77d	Z19
	LD	(IY+d),B	FD70d	Z19
	LD	(IY+d),C	FD71d	Z19
	LD	(IY+d),D	FD72d	Z19
	LD	(IY+d),E	FD73d	Z19
	LD	(IY+d),H	FD74d	Z19
	LD	(IY+d),L	FD75d	Z19
	LD	(IY+d),n	FD36dn	Z19
	LD	(aa),A	32aa	Z13
	LD	(aa),BC	ED43aa	Z20
	LD	(aa),DE	ED53aa	Z20
	LD	(aa),HL	22aa	Z16
	LD	(aa),IX	DD22aa	Z20
	LD	(aa),IY	FD22aa	Z20
	LD	(aa) SP	ED73aa	Z20
	LD	A,(BC)	0A	Z7
	LD	A,(DE)	1A	Z7
	LD	A,(aa)	3Aaa	Z13
	LD	A,I	ED57	U9
	LD	A,R	ED5F	U9
	LD	A,—	TABLE	Z
	LD	B,—	TABLE	Z
	LD	BC,(aa)	ED4Baa	Z20
	LD	BC,aa	01aa	Z10
	LD	C,—	TABLE	Z
	LD	D,—	TABLE	Z
	LD	DE,(aa)	ED5Baa	Z20
	LD	DE,aa	11aa	Z10
	LD	E,—	TABLE	Z
	LD	H,—	TABLE	Z
	LD	HL,(aa)	2Aaa	Z16
	LD	HL,aa	21aa	Z10
	LD	I,A	ED47	Z9
	LD	IX,(aa)	DD2Aaa	Z20
	LD	IX,aa	DD21aa	Z14
	LD	IY,(aa)	FD2Aaa	Z20
	LD	IY,aa	FD21aa	Z14
	LD	L,—	TABLE	Z
	LD	R,A	ED4F	Z9
	LD	SP,(aa)	ED7Baa	Z20
	LD	SP,HL	F9	Z6
	LD	SP,IX	DDF9	Z10
	LD	SP,IY	FDF9	Z10
	LD	SP,aa	31aa	Z10
	LDD		EDA8	R16
	LDDR		EDB8	S(1)
	LDI		EDA0	R16
	LDIR		EDB0	S(1)
	NEG		ED44	B8
O	NOP		00	Z4
	OR	—	TABLE	D
	OTDR		EDBB	Q(1)
	OTIR		EDB3	Q(1)
	OUT	(C),A	ED79	Z12
	OUT	(C),B	ED41	Z12
	OUT	(C),C	ED49	Z12
	OUT	(C),D	ED51	Z12
	OUT	(C),E	ED59	Z12
	OUT	(C),H	ED61	Z12
	OUT	(C),L	ED69	Z12
	OUT	(n),A	D3n	Z11
	OUTD		EDAB	P16
	OUT I		EDA3	P16
P	POP	AF	F1	Z10
	POP	BC	C1	Z10
	POP	DE	D1	Z10
	POP	HL	E1	Z10
	POP	IX	DDE1	Z14
	POP	IY	FDE1	Z14
	PUSH	AF	F5	Z11
	PUSH	BC	C5	Z11
	PUSH	DE	D5	Z11
	PUSH	HL	E5	Z11
	PUSH	IX	DDE5	Z15
	PUSH	IY	FDE5	Z15
R	RES	—	TABLE	Z
	RET		C9	Z10
	RET	C	D8	Z(4)
	RET	M	F8	Z(4)
	RET	NC	D0	Z(4)
	RET	NZ	C0	Z(4)
	RET	P	F0	Z(4)
	RET	PE	E8	Z(4)
	RET	PO	E0	Z(4)
	RET	Z	C8	Z(4)
	RETI		ED4D	Z14
	RETN		ED45	Z14
	RL	—	TABLE	K
	RLA		17	J4
	RLC	—	TABLE	K
	RLCA		07	J4
	RLD		ED6F	L18
	RR	—	TABLE	K
	RRA		1F	J4
	RRC	—	TABLE	K
	RRCA		0F	J4
	RRD		ED67	L18
	RST	00H	C7	Z11
	RST	08H	CF	Z11
	RST	10H	D7	Z11
	RST	18H	DF	Z11
	RST	20H	E7	Z11
	RST	28H	EF	Z11
	RST	30H	F7	Z11
	RST	38H	FF	Z11
S	SBC	A,—	TABLE	B
	SBC	HL,BC	ED42	I15
	SBC	HL,DE	ED52	I15
	SBC	HL,HL	ED62	I15
	SBC	HL,SP	ED72	I15
	SCF		37	O4
	SET	—	TABLE	Z
	SLA	—	TABLE	K
	SRA	—	TABLE	K
	SRL	—	TABLE	K
	SUB	—	TABLE	B
	XOR	—	TABLE	D

	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)	
BIT 0,	CB,47	CB,40	CB,41	CB,42	CB,43	CB,44	CB,45	CB,46	DD,CB,d,46	FD,CB,d,46	V
BIT 1,	CB,4F	CB,48	CB,49	CB,4A	CB,4B	CB,4C	CB,4D	CB,4E	DD,CB,d,4E	FD,CB,d,4E	V
BIT 2,	CB,57	CB,50	CB,51	CB,52	CB,53	CB,54	CB,55	CB,56	DD,CB,d,56	FD,CB,d,56	V
BIT 3,	CB,5F	CB,58	CB,59	CB,5A	CB,5B	CB,5C	CB,5D	CB,5E	DD,CB,d,5E	FD,CB,d,5E	V
BIT 4,	CB,67	CB,60	CB,61	CB,62	CB,63	CB,64	CB,65	CB,66	DD,CB,d,66	FD,CB,d,66	V
BIT 5,	CB,6F	CB,68	CB,69	CB,6A	CB,6B	CB,6C	CB,6D	CB,6E	DD,CB,d,6E	FD,CB,d,6E	V
BIT 6,	CB,77	CB,70	CB,71	CB,72	CB,73	CB,74	CB,75	CB,76	DD,CB,d,76	FD,CB,d,76	V
BIT 7,	CB,7F	CB,78	CB,79	CB,7A	CB,7B	CB,7C	CB,7D	CB,7E	DD,CB,d,7E	FD,CB,d,7E	V
	STATES:			8				12	20		

	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)	
RES 0,	CB,87	CB,80	CB,81	CB,82	CB,83	CB,84	CB,85	CB,86	DD,CB,d,86	FD,CB,d,86	Z
RES 1,	CB,8F	CB,88	CB,89	CB,8A	CB,8B	CB,8C	CB,8D	CB,8E	DD,CB,d,8E	FD,CB,d,8E	Z
RES 2,	CB,97	CB,90	CB,91	CB,92	CB,93	CB,94	CB,95	CB,96	DD,CB,d,96	FD,CB,d,96	Z
RES 3,	CB,9F	CB,98	CB,99	CB,9A	CB,9B	CB,9C	CB,9D	CB,9E	DD,CB,d,9E	FD,CB,d,9E	Z
RES 4,	CB,A7	CB,A0	CB,A1	CB,A2	CB,A3	CB,A4	CB,A5	CB,A6	DD,CB,d,A6	FD,CB,d,A6	Z
RES 5,	CB,AF	CB,A8	CB,A9	CB,AA	CB,AB	CB,AC	CB,AD	CB,AE	DD,CB,d,AE	FD,CB,d,AE	Z
RES 6,	CB,B7	CB,B0	CB,B1	CB,B2	CB,B3	CB,B4	CB,B5	CB,B6	DD,CB,d,B6	FD,CB,d,B6	Z
RES 7,	CB,BF	CB,B8	CB,B9	CB,BA	CB,BB	CB,BC	CB,BD	CB,BE	DD,CB,d,BE	FD,CB,d,BE	Z
SET 0,	CB,C7	CB,C0	CB,C1	CB,C2	CB,C3	CB,C4	CB,C5	CB,C6	DD,CB,d,C6	FD,CB,d,C6	Z
SET 1,	CB,CF	CB,C8	CB,C9	CB,CA	CB,CB	CB,CC	CB,CD	CB,CE	DD,CB,d,CE	FD,CB,d,CE	Z
SET 2,	CB,D7	CB,D0	CB,D1	CB,D2	CB,D3	CB,D4	CB,D5	CB,D6	DD,CB,d,D6	FD,CB,d,D6	Z
SET 3,	CB,DF	CB,D8	CB,D9	CB,DA	CB,DB	CB,DC	CB,DD	CB,DE	DD,CB,d,DE	FD,CB,d,DE	Z
SET 4,	CB,E7	CB,E0	CB,E1	CB,E2	CB,E3	CB,E4	CB,E5	CB,E6	DD,CB,d,E6	FD,CB,d,E6	Z
SET 5,	CB,EF	CB,E8	CB,E9	CB,EA	CB,EB	CB,EC	CB,ED	CB,EE	DD,CB,d,EE	FD,CB,d,EE	Z
SET 6,	CB,F7	CB,F0	CB,F1	CB,F2	CB,F3	CB,F4	CB,F5	CB,F6	DD,CB,d,F6	FD,CB,d,F6	Z
SET 7,	CB,FF	CB,F8	CB,F9	CB,FA	CB,FB	CB,FC	CB,FD	CB,FE	DD,CB,d,FE	FD,CB,d,FE	Z
	STATES:			8				15	23		

	A(8)	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)	
RLC	CB,07	CB,00	CB,01	CB,02	CB,03	CB,04	CB,05	CB,06	DD,CB,d,06	FD,CB,d,06	K
RRC	CB,0F	CB,08	CB,09	CB,0A	CB,0B	CB,0C	CB,0D	CB,0E	DD,CB,d,0E	FD,CB,d,0E	K
RL	CB,17	CB,10	CB,11	CB,12	CB,13	CB,14	CB,15	CB,16	DD,CB,d,16	FD,CB,d,16	K
RR	CB,1F	CB,18	CB,19	CB,1A	CB,1B	CB,1C	CB,1D	CB,1E	DD,CB,d,1E	FD,CB,d,1E	K
SLA	CB,27	CB,20	CB,21	CB,22	CB,23	CB,24	CB,25	CB,26	DD,CB,d,26	FD,CB,d,26	K
SRA	CB,2F	CB,28	CB,29	CB,2A	CB,2B	CB,2C	CB,2D	CB,2E	DD,CB,d,2E	FD,CB,d,2E	K
SRL	CB,3F	CB,38	CB,39	CB,3A	CB,3B	CB,3C	CB,3D	CB,3E	DD,CB,d,3E	FD,CB,d,3E	K
	STATES:			8				15	23		

Flag Codes

	C	Z	P V	S	N	H	
A	C	Z	V	S	0	H	
B	C	Z	V	S	1	H	
C	0	Z	P	S	0	1	
D	0	Z	P	S	0	0	
E	=	Z	V	S	0	H	
F	=	Z	V	S	1	H	
G	C	=	=	=	0	U	
H	C	Z	V	S	0	U	
I	C	Z	V	S	1	U	
J	C	=	=	=	0	0	
K	C	Z	P	S	0	0	
L	=	Z	P	S	0	0	
M	C	Z	P	S	=	H	
N	=	=	=	=	1	1	
O	1	=	=	=	0	0	
P	U	F	U	U	1	U	(1)
Q	U	1	U	U	1	U	
R	=	U	F	U	0	0	(2)
S	=	U	0	U	0	0	
T	=	F	F	S	1	U	(3)
U	=	Z	F	S	0	0	(4)
V	=	Z	U	U	0	1	(5)
W	=	Z	P	S	0	U	
X							
Y							
Z	=	=	=	=	=	=	

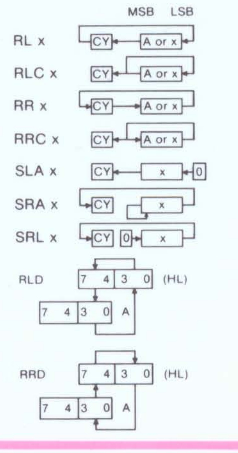
Codes:
0: reset
1: set
C: Carry*
F: Footnote
H: Half carry*
N: Add/Sub*
P: Parity*
S: Sign*
U: Undefined
V: oVerflow*
Z: Zero*
=: not affected

* Indicated flag affected by result

(1)Z=1 iff B becomes 0
(2)PV=0 iff BC becomes 0
(3)PV=0 iff BC becomes 0 and Z=1 iff A=(HL)
(4)PV=IFF2
(5)Z=$\overline{\text{bit}}$

	A	B	C	D	E	H	L	(HL)	n	(IX+d)	(IY+d)	
ADC A,	8F	88	89	8A	8B	8C	8D	8E	CE,n	DD,8E,d	FD,8E,d	A
ADD A,	87	80	81	82	83	84	85	86	C6,n	DD,86,d	FD,86,d	A
AND	A7	A0	A1	A2	A3	A4	A5	A6	E6,n	DD,A6,d	FD,A6,d	C
CP	BF	B8	B9	BA	BB	BC	BD	BE	FE,n	DD,BE,d	FD,BE,d	B
OR	B7	B0	B1	B2	B3	B4	B5	B6	F6,n	DD,B6,d	FD,B6,d	D
SBC A,	9F	98	99	9A	9B	9C	9D	9E	DE,n	DD,9E,d	FD,9E,d	B
SUB	97	90	91	92	93	94	95	96	D6,n	DD,96,d	FD,96,d	B
XOR	AF	A8	A9	AA	AB	AC	AD	AE	EE,n	DD,AE,d	FD,AE,d	D
LD A,	7F	78	79	7A	7B	7C	7D	7E	3E,n	DD,7E,d	FD,7E,d	Z
LD B,	47	40	41	42	43	44	45	46	06,n	DD,46,d	FD,46,d	Z
LD C,	4F	48	49	4A	4B	4C	4D	4E	0E,n	DD,4E,d	FD,4E,d	Z
LD D,	57	50	51	52	53	54	55	56	16,n	DD,56,d	FD,56,d	Z
LD E,	5F	58	59	5A	5B	5C	5D	5E	1E,n	DD,5E,d	FD,5E,d	Z
LD H,	67	60	61	62	63	64	65	66	26,n	DD,66,d	FD,66,d	Z
LD L,	6F	68	69	6A	6B	6C	6D	6E	2E,n	DD,6E,d	FD,6E,d	Z
	STATES: 4							7		19		

Rotates and Shifts



Addressing

n	n is immediate 8-bit data.
aa	aa is immediate 16-bit data or address to CALL/to JP to.
(aa)	aa is address of data.
(rr)	16-bit reg rr holds address of data or address to CALL or to JP to.
(n)	n is port number.
(r)	8-bit reg r holds port number.
(IX+d)	IX+d is address of data (d is a 1 byte signed displacement).
d	In relative jumping, address to jump to is d + address of next instruction (d is signed).

Full 2 byte addresses in code, stack, and data areas are stored low byte followed by high byte. Thus JP 1234H is: C3,34,12.

SP points to used byte at top of stack. PUSH decrements SP by 2.

Notes

(1) 21 except 16 at termination
(2) 13 except 8 at termination
(3) 12 for success; 7 for failure
(4) 11 for success; 5 for failure
(5) 17 for success; 10 for failure
(6) A to A15..A8 and n to A7..A0
(7) B to A15..A8 and C to A7..A0
(8) See faster version of 'Rotate A' instructions

Intentionally Blank

DO NOT PLACE ON HOT SURFACE

Copyrighted and published by Micro Logic Corp. POB 174, Hackensack, NJ 07602. Dealer, school, catalogue, club, premium, and OEM inquiries welcome. End user comments invited. Printed in U.S.A. World copyrighted. All rights reserved.

WHILE PREPARED WITH EXTREME CARE, THERE ARE NO WARRANTIES EXPRESS OR IMPLIED AS TO MERCHANTABILITY OR FITNESS FOR PURPOSE.

AUTHOR: JAMES D LEWIS

#104A

MICRO LOGIC CORP.
HACKENSACK, NJ

6502 (65XX)

MICROPROCESSOR INSTANT REFERENCE CARD

MICRO® CHART

INSTRUCTION SET

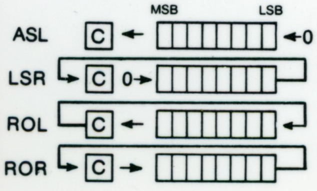
	INSTRUCTION	OP	C	B	DESCRIPTION	ADDRESSING
A	ADC #n	69	2	2	Add with carry to A	Immediate
	ADC nn	6D	4	3	Add with carry to A	Absolute
	ADC n	65	3	2	Add with carry to A	Zero Page
	ADC (n,X)	61	6	2	Add with carry to A	Ind X
	ADC (n),Y	71	5+	2	Add with carry to A	Ind Y
	ADC n,X	75	4	2	Add with carry to A	Zero Page X
	ADC nn,X	7D	4+	3	Add with carry to A	Absolute X
	ADC nn,Y	79	4+	3	Add with carry to A	Absolute Y
	AND #n	29	2	2	AND to A	Immediate
	AND nn	2D	4	3	AND to A	Absolute
	AND n	25	3	2	AND to A	Zero Page
	AND (n,X)	21	6	2	AND to A	Ind X
	AND (n),Y	31	5+	2	AND to A	Ind Y
	AND n,X	35	4	2	AND to A	Zero Page X
	AND nn,X	3D	4+	3	AND to A	Absolute X
	AND nn,Y	39	4+	3	AND to A	Absolute Y
	ASL nn	0E	6	3	Arithmetic shift left	Absolute
	ASL n	06	5	2	Arithmetic shift left	Zero Page
	ASL A	0A	2	1	Arithmetic shift left	Accumulator
	ASL n,X	16	6	2	Arithmetic shift left	Zero Page X
	ASL nn,X	1E	7	3	Arithmetic shift left	Absolute X
B	BCC n	90	2+	2	Branch if carry clear (C=0)	Relative
	BCS n	B0	2+	2	Branch if carry set (C=1)	Relative
	BEQ n	F0	2+	2	Branch if equal (Z=1)	Relative
	BNE n	D0	2+	2	Branch if not equal (Z=0)	Relative
	BMI n	30	2+	2	Branch if minus (N=1)	Relative
	BPL n	10	2+	2	Branch if plus (N=0)	Relative
	BVC n	50	2+	2	Branch if ovfl clear (V=0)	Relative
	BVS n	70	2+	2	Branch if ovfl set (V=1)	Relative
	BIT nn	2C	4	3	AND with A (A unchanged)	Absolute
	BIT n	24	3	2	AND with A (A unchanged)	Zero Page
	BRK	00	7	1	Break (force interrupt)	None
C	CLC	18	2	1	Clear carry	None
	CLD	D8	2	1	Clear decimal mode	None
	CLI	58	2	1	Clear IRQ disable	None
	CLV	B8	2	1	Clear overflow	None
	CMP #n	C9	2	2	Compare with A	Immediate
	CMP nn	CD	4	3	Compare with A	Absolute
	CMP n	C5	3	2	Compare with A	Zero Page
	CMP (n,X)	C1	6	2	Compare with A	Ind X
	CMP (n),Y	D1	5+	2	Compare with A	Ind Y
	CMP n,X	D5	4	2	Compare with A	Zero Page X
	CMP nn,X	DD	4+	3	Compare with A	Absolute X
	CMP nn,Y	D9	4+	3	Compare with A	Absolute Y
	CPX #n	E0	2	2	Compare with X	Immediate
	CPX nn	EC	4	3	Compare with X	Absolute
	CPX n	E4	3	2	Compare with X	Zero Page
	CPY #n	C0	2	2	Compare with Y	Immediate
	CPY nn	CC	4	3	Compare with Y	Absolute
	CPY n	C4	3	2	Compare with Y	Zero Page
D	DEC nn	CE	6	3	Decrement by one	Absolute
	DEC n	C6	5	2	Decrement by one	Zero Page
	DEC n,X	D6	6	2	Decrement by one	Zero Page X
	DEC nn,X	DE	7	3	Decrement by one	Absolute X
	DEX	CA	2	1	Decrement X by one	None
	DEY	88	2	1	Decrement Y by one	None
E	EOR #n	49	2	2	XOR to A	Immediate
	EOR nn	4D	4	3	XOR to A	Absolute
	EOR n	45	3	2	XOR to A	Zero Page
	EOR (n,X)	41	6	2	XOR to A	Ind X
	EOR (n),Y	51	5+	2	XOR to A	Ind Y
	EOR n,X	55	4	2	XOR to A	Zero Page X
	EOR nn,X	5D	4+	3	XOR to A	Absolute X
	EOR nn,Y	59	4+	3	XOR to A	Absolute Y
I	INC nn	EE	6	3	Increment by one	Absolute
	INC n	E6	5	2	Increment by one	Zero Page
	INC n,X	F6	6	2	Increment by one	Zero Page X
	INC nn,X	FE	7	3	Increment by one	Absolute X
	INX	E8	2	1	Increment X by one	None
	INY	C8	2	1	Increment Y by one	None
J	JMP nn	4C	3	3	Jump to new location	Absolute
	JMP (nn)	6C	5	3	Jump to new location	Indirect
	JSR nn	20	6	3	Jump to subroutine	Absolute

	INSTRUCTION	OP	C	B	DESCRIPTION	ADDRESSING
L	LDA #n	A9	2	2	Load A	Immediate
	LDA nn	AD	4	3	Load A	Absolute
	LDA n	A5	3	2	Load A	Zero Page
	LDA (n,X)	A1	6	2	Load A	Ind X
	LDA (n),Y	B1	5+	2	Load A	Ind Y
	LDA n,X	B5	4	2	Load A	Zero Page X
	LDA nn,X	BD	4+	3	Load A	Absolute X
	LDA nn,Y	B9	4+	3	Load A	Absolute Y
	LDX #n	A2	2	2	Load X	Immediate
	LDX nn	AE	4	3	Load X	Absolute
	LDX n	A6	3	2	Load X	Zero Page
	LDX nn,Y	BE	4+	3	Load X	Absolute Y
	LDX n,Y	B6	4	2	Load X	Zero Page Y
	LDY #n	A0	2	2	Load Y	Immediate
	LDY nn	AC	4	3	Load Y	Absolute
	LDY n	A4	3	2	Load Y	Zero Page
	LDY n,X	B4	4	2	Load Y	Zero Page X
	LDY nn,X	BC	4+	3	Load Y	Absolute X
	LSR nn	4E	6	3	Logical shift right	Absolute
	LSR n	46	5	2	Logical shift right	Zero Page
	LSR A	4A	2	1	Logical shift right	Accumulator
	LSR n,X	56	6	2	Logical shift right	Zero Page X
	LSR nn,X	5E	7	3	Logical shift right	Absolute X
N	NOP	EA	2	1	No operation	None
O	ORA #n	09	2	2	OR to A	Immediate
	ORA nn	0D	4	3	OR to A	Absolute
	ORA n	05	3	2	OR to A	Zero Page
	ORA (n,X)	01	6	2	OR to A	Ind X
	ORA (n),Y	11	5+	2	OR to A	Ind Y
	ORA n,X	15	4	2	OR to A	Zero Page X
	ORA nn,X	1D	4+	3	OR to A	Absolute X
	ORA nn,Y	19	4+	3	OR to A	Absolute Y
P	PHA	48	3	1	Push A onto stack	None
	PHP	08	3	1	Push P onto stack	None
	PLA	68	4	1	Pull (pop) A from stack	None
	PLP	28	4	1	Pull (pop) P from stack	None
R	ROL nn	2E	6	3	Rotate left through carry	Absolute
	ROL n	26	5	2	Rotate left through carry	Zero Page
	ROL A	2A	2	1	Rotate left through carry	Accumulator
	ROL n,X	36	6	2	Rotate left through carry	Zero Page X
	ROL nn,X	3E	7	3	Rotate left through carry	Absolute X
	ROR nn	6E	6	3	Rotate right through carry	Absolute
	ROR n	66	5	2	Rotate right through carry	Zero Page
	ROR A	6A	2	1	Rotate right through carry	Accumulator
	ROR n,X	76	6	2	Rotate right through carry	Zero Page X
	ROR nn,X	7E	7	3	Rotate right through carry	Absolute X
	RTI	40	6	1	Return from interrupt	None
	RTS	60	6	1	Return from subroutine	None
S	SBC #n	E9	2	2	Subtract with borrow from A	Immediate
	SBC nn	ED	4	3	Subtract with borrow from A	Absolute
	SBC n	E5	3	2	Subtract with borrow from A	Zero Page
	SBC (n,X)	E1	6	2	Subtract with borrow from A	Ind X
	SBC (n),Y	F1	5+	2	Subtract with borrow from A	Ind Y
	SBC n,X	F5	4	2	Subtract with borrow from A	Zero Page X
	SBC nn,X	FD	4+	3	Subtract with borrow from A	Absolute X
	SBC nn,Y	F9	4+	3	Subtract with borrow from A	Absolute Y
	SEC	38	2	1	Set carry	None
	SED	F8	2	1	Set decimal mode	None
	SEI	78	2	1	Set IRQ disable	None
	STA nn	8D	4	3	Store A	Absolute
	STA n	85	3	2	Store A	Zero Page
	STA (n,X)	81	6	2	Store A	Ind X
	STA (n),Y	91	6	2	Store A	Ind Y
	STA n,X	95	4	2	Store A	Zero Page X
	STA nn,X	9D	5	3	Store A	Absolute X
	STA nn,Y	99	5	3	Store A	Absolute Y
	STX nn	8E	4	3	Store X	Absolute
	STX n	86	3	2	Store X	Zero Page
	STX n,Y	96	4	2	Store X	Zero Page Y
	STY nn	8C	4	3	Store Y	Absolute
	STY n	84	3	2	Store Y	Zero Page
	STY n,X	94	4	2	Store Y	Zero Page X
T	TAX	AA	2	1	Transfer A to X	None
	TAY	A8	2	1	Transfer A to Y	None
	TSX	BA	2	1	Transfer S to X	None
	TXA	8A	2	1	Transfer X to A	None
	TXS	9A	2	1	Transfer X to S	None
	TYA	98	2	1	Transfer Y to A	None

Instruction Notes

ADC	A+DATA+C→A
BRK	Ignore I flag, Set B=1 Push return address+1 Push P Jump to IRQ vector
JSR	Push return address-1 Jump absolute
RTI	Pop P, Pop PC
RTS	Pop PC, Increment PC
SBC	A-DATA-$\overline{C}$ →A

Shift Instructions



Added Cycle Time

A (+) in the (C) column for branch instructions means:
Add 0 if branch not taken.
Add 1 if taken within page.
Add 2 if taken across pages.

A (+) in the (C) column for other instructions means:
Add 1 if indexing across page boundary

Assembler Symbols

.	Assembler directive
#	Immediate addressing
$	Hex number prefix
@	Octal number prefix
%	Binary number prefix
'	ASCII character prefix
()	Indirect addressing
;	In col 1 for comment

Intentionally Blank

DO NOT PLACE ON HOT SURFACES

WHILE PREPARED WITH EXTREME CARE, THERE ARE NO WARRANTIES EXPRESS OR IMPLIED AS TO MERCHANTABILITY OR FITNESS FOR PURPOSE.

MICRO CHART
AUTHOR:
JAMES D. LEWIS

#101B

AVAILABLE CARDS:
Z80, 6502, 8048,
8080/8085, 7400-TTL
BASIC ALGORITHMS

100%
PLASTIC

MICRO LOGIC CORP.
HACKENSACK, NJ

6502 (65XX)

MICROPROCESSOR INSTANT REFERENCE CARD

MICRO® CHART

Hex to Instruction Conversion

LSD→

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0-	BRK	ORA (n,X)	□	□	□	ORA n	ASL n	□	PHP	ORA #n	ASL A	□	□	ORA nn	ASL nn	□	0-
1-	BPL n	ORA (n),Y	□	□	□	ORA n,X	ASL n,X	□	CLC	ORA nn,Y	□	□	□	ORA nn,X	ASL nn,X	□	1-
2-	JSR nn	AND (n,X)	□	□	BIT n	AND n	ROL n	□	PLP	AND #n	ROL A	□	BIT nn	AND nn	ROL nn	□	2-
3-	BMI n	AND (n),Y	□	□	□	AND n,X	ROL n,X	□	SEC	AND nn,Y	□	□	□	AND nn,X	ROL nn,X	□	3-
4-	RTI	EOR (n,X)	□	□	□	EOR n	LSR n	□	PHA	EOR #n	LSR A	□	JMP nn	EOR nn	LSR nn	□	4-
5-	BVC n	EOR (n),Y	□	□	□	EOR n,X	LSR n,X	□	CLI	EOR nn,Y	□	□	□	EOR nn,X	LSR nn,X	□	5-
6-	RTS	ADC (n,X)	□	□	□	ADC n	ROR n	□	PLA	ADC #n	ROR A	□	JMP (nn)	ADC nn	ROR nn	□	6-
7-	BVS n	ADC (n),Y	□	□	□	ADC n,X	ROR n,X	□	SEI	ADC nn,Y	□	□	□	ADC nn,X	ROR nn,X	□	7-
8-	□	STA (n,X)	□	□	STY n	STA n	STX n	□	DEY	□	TXA	□	STY nn	STA nn	STX nn	□	8-
9-	BCC n	STA (n),Y	□	□	STY n,X	STA n,X	STX n,Y	□	TYA	STA nn,Y	TXS	□	□	STA nn,X	□	□	9-
A-	LDY #n	LDA (n,X)	LDX #n	□	LDY n	LDA n	LDX n	□	TAY	LDA #n	TAX	□	LDY nn	LDA nn	LDX nn	□	A-
B-	BCS n	LDA (n),Y	□	□	LDY n,X	LDA n,X	LDX n,Y	□	CLV	LDA nn,Y	TSX	□	LDY nn,X	LDA nn,X	LDX nn,Y	□	B-
C-	CPY #n	CMP (n,X)	□	□	CPY n	CMP n	DEC n	□	INY	CMP #n	DEX	□	CPY nn	CMP nn	DEC nn	□	C-
D-	BNE n	CMP (n),Y	□	□	□	CMP n,X	DEC n,X	□	CLD	CMP nn,Y	□	□	□	CMP nn,X	DEC nn,X	□	D-
E-	CPX #n	SBC (n,X)	□	□	CPX n	SBC n	INC n	□	INX	SBC #n	NOP	□	CPX nn	SBC nn	INC nn	□	E-
F-	BEQ n	SBC (n),Y	□	□	□	SBC n,X	INC n,X	□	SED	SBC nn,Y	□	□	□	SBC nn,X	INC nn,X	□	F-
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Memory Map

ZERO PAGE	0000 00FF
DATA & STACK*	0100 01FF
RAM I/O ROM	0200 FFF9
NMI VECTOR	FFFA&B
RES VECTOR	FFFC&D
IRQ VECTOR	FFFE&F

*In systems with < 512 bytes of RAM the hardware can ignore signal AB8, moving stack into page zero.

Status Flags

MSB LSB

N	V	–	B	D	I	Z	C

N=negative result
V=overflow
B=BRK instruction
D=decimal mode
I=IRQ disable
Z=zero result
C=carry=$\overline{\text{borrow}}$

Note: above is true when flag = 1.

Overflow normally signifies signed arithmetic result is out of range.

When D=1, only ADC and SBC use decimal (BCD) arithmetic.

Effect on Flags

	N	V	-	B	D	I	Z	C	
ADC	N	V	-	-	-	-	Z	C	①
AND	N	-	-	-	-	-	Z	-	
ASL	N	-	-	-	-	-	Z	C	
BIT	N	V	-	-	-	-	Z	-	②
BRK	-	-	-	1	-	1	-	-	
CLC	-	-	-	-	-	-	-	0	
CLD	-	-	-	-	0	-	-	-	
CLI	-	-	-	-	-	0	-	-	
CLV	-	0	-	-	-	-	-	-	
CMP	N	-	-	-	-	-	Z	C	
CPX	N	-	-	-	-	-	Z	C	
CPY	N	-	-	-	-	-	Z	C	
DEC	N	-	-	-	-	-	Z	-	
DEX	N	-	-	-	-	-	Z	-	
DEY	N	-	-	-	-	-	Z	-	
EOR	N	-	-	-	-	-	Z	-	
INC	N	-	-	-	-	-	Z	-	
INX	N	-	-	-	-	-	Z	-	
INY	N	-	-	-	-	-	Z	-	
LDA	N	-	-	-	-	-	Z	-	
LDX	N	-	-	-	-	-	Z	-	
LDY	N	-	-	-	-	-	Z	-	
LSR	0	-	-	-	-	-	Z	C	
ORA	N	-	-	-	-	-	Z	-	
PLA	N	-	-	-	-	-	Z	-	
PLP	N	V	-	B	D	I	Z	C	
ROL	N	-	-	-	-	-	Z	C	
ROR	N	-	-	-	-	-	Z	C	
RTI	N	V	-	B	D	I	Z	C	
SBC	N	V	-	-	-	-	Z	C	③
SEC	-	-	-	-	-	-	-	1	
SED	-	-	-	-	1	-	-	-	
SEI	-	-	-	-	-	1	-	-	
TAX	N	-	-	-	-	-	Z	-	
TAY	N	-	-	-	-	-	Z	-	
TSX	N	-	-	-	-	-	Z	-	
TXA	N	-	-	-	-	-	Z	-	
TYA	N	-	-	-	-	-	Z	-	

① If in decimal mode Z flag is invalid.

② N = data bit 7
V = data bit 6
Z = AND result

③ $\overline{C}$ = borrow

Note: unlisted instructions have no effect on flags.

Addressing Modes

Note: Full 2 byte addresses in code , stack, and data areas are stored low byte followed by high byte. Thus, in hex, JMP $1234 is: 4C 34 12

FORM	ADDRESSING	DESCRIPTION
nn	Absolute	Location nn holds data.
nn,X	Absolute X	Location nn+X holds data.
nn,Y	Absolute Y	Location nn+Y holds data.
A	Accumulator	Accumulator holds data.
#n	Immediate	n is data.
(n,X)	Ind X	Location n+X and next of page 0 hold address of data.*,**
(n), Y	Ind Y	Address of data is Y + address held by location n and next of page 0.**
(nn)	Indirect	Location nn and next hold adddress to jump to.**
n	Relative	Address to jump to is n + address of next instruction, with n treated as a signed number.
n	Zero Page	Location n of page 0 holds data.
n,X	Zero Page X	Location n+X of page 0 holds data.
n,Y	Zero Page Y	Location n+Y of page 0 holds data.

*n+X is computed discarding any carry.
**2 bytes must not cross page boundary.

ASCII Character Set

LSD \ MSD		0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	0000	NUL	DLE	SP	0	@	P	'	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	″	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	,	<	L	\	l	|
D	1101	CR	GS	-	=	M	]	m	}
E	1110	SO	RS	.	>	N	↑	n	~
F	1111	SI	US	/	?	O	←	o	DEL

Interrupts

$\overline{\text{IRQ}}$ is low level sensitive.
$\overline{\text{NMI}}$ is falling edge sensitive.
Reset sets I=1.
Interrupts are processed by:

1. Push PC of unexecuted instruction.
2. Push P.
3. I=1.
4. Jump via appropriate vector.

Miscellaneous

S points to next free byte of stack.

Stack push decrements S.

In pushing PC, high byte is pushed first.

Pre 6/76 chips have no ROR instruction.

65XX is a totally software compatible family.

This card is based on specifications from MOS Technology, Inc.

Registers

A	ACCUMULATOR
Y	Y INDEX REG
X	X INDEX REG
PC	PROGRAM COUNTER
S	STACK PNTR
P	FLAGS

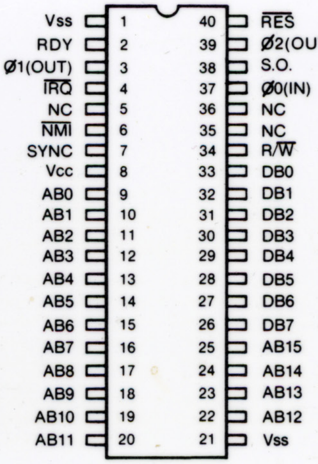
A, Y, X, S, P = 1 byte.
Only PC is 2 bytes.

Hex and Decimal Conversion

LSD→

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	2
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	3
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	4
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	5
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	6
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	7
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	8
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	9
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	A
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	B
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	C
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	D
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	E
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

6502 Pins



Abbreviations

B = number of Bytes
C = number of Cycles. also Carry.

n = 1 byte quantity
nn = 2 byte quantity

IRQ = Interrupt ReQuest
NMI = Non Maskable Interrupt
RES = RESet
XOR = eXclusive OR
(00→0 01→1 10→1 11→0)

A,P,S,X,Y,PC=see "Registers"
N,V,B,D,I,Z,C = see "Status Flags"
.#$@%'(); = see "Assembler Symbols"

Unsigned Comparisons

example: CMP # n

A < n	BCC YES
A = n	BEQ YES
A > n	BCC NO BNE YES
A ⩾ n	BCS YES
A ≠ n	BNE YES
A ⩽ n	BCC YES BEQ YES

YES represents label for code to be executed if condition is true. For > & ⩽, test requires both instructions.

Internally, A–n is computed to determine N,Z,C flags.

INSTANT
ACCESS