



HACKENSACK, NJ

## Z80 CPU

## MICROPROCESSOR INSTANT REFERENCE CARD

MICRO<sup>®</sup>  
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

ADC	A,—	TABLE	A	LD	(IX+d),C	DD71d	Z19	
ADC	HL,BC	ED4A	H15	LD	(IX+d),D	DD72d	Z19	
ADC	HL,DE	ED5A	H15	LD	(IX+d),E	DD73d	Z19	
ADC	HL,HL	ED6A	H15	LD	(IX+d),H	DD74d	Z19	
ADC	HL,SP	ED7A	H15	LD	(IX+d),L	DD75d	Z19	
ADD	A,—	TABLE	A	LD	(IX+d),n	DD36dn	Z19	
ADD	HL,BC	09	G11	LD	(IY+d),A	FD77d	Z19	
ADD	HL,DE	19	G11	LD	(IY+d),B	FD70d	Z19	
ADD	HL,HL	29	G11	LD	(IY+d),C	FD71d	Z19	
ADD	HL,SP	39	G11	LD	(IY+d),D	FD72d	Z19	
ADD	IX,BC	DD09	G15	LD	(IY+d),E	FD73d	Z19	
ADD	IX,DE	DD19	G15	LD	(IY+d),H	FD74d	Z19	
ADD	IX,IX	DD29	G15	LD	(IY+d),L	FD75d	Z19	
ADD	IX,SP	DD39	G15	LD	(IY+d),n	FD36dn	Z19	
ADD	IY,BC	FD09	G15	LD	(aa),A	32aa	Z13	
ADD	IY,DE	FD19	G15	LD	(aa),BC	ED43aa	Z20	
ADD	IY,IY	FD29	G15	LD	(aa),DE	ED53aa	Z20	
ADD	IY,SP	FD39	G15	LD	(aa),HL	22aa	Z16	
AND	—	TABLE	C	LD	(aa),IX	DD22aa	Z20	
BIT	—	TABLE	V	LD	(aa),IY	FD22aa	Z20	
CALL	aa	CDaa	Z17	LD	(aa),SP	ED73aa	Z20	
CALL	C,aa	DCaa	Z(5)	LD	A,(BC)	0A	Z7	
CALL	M,aa	FCaa	Z(5)	LD	A,(DE)	1A	Z7	
CALL	NC,aa	DAaa	Z(5)	LD	A,(aa)	3Aaa	Z13	
CALL	NZ,aa	C4aa	Z(5)	LD	A,I	ED57	U9	
CALL	P,aa	F4aa	Z(5)	LD	A,R	ED5F	U9	
CALL	PE,aa	ECaa	Z(5)	LD	A,—	TABLE	Z	
CALL	PO,aa	E4aa	Z(5)	LD	B,—	TABLE	Z	
CALL	Z,aa	QCAa	Z(5)	LD	BC,(aa)	ED4Baa	Z20	
CFP	—	TABLE	G4	LD	BC,aa	01aa	Z10	
CP	—	TABLE	B	LD	C,—	TABLE	Z	
CPD	—	EDA9	T16	LD	D,—	TABLE	Z	
CPDR	—	EDB9	T(1)	LD	DE,(aa)	ED5Baa	Z20	
CPI	—	EDA1	T16	LD	DE,aa	11aa	Z10	
CPH	—	EDB1	T(1)	LD	E,—	TABLE	Z	
CPL	—	2F	N4	LD	H,—	TABLE	Z	
DAA	—	27	M4	LD	HL,(aa)	2Aaa	Z16	
DEC	(HL)	35	F11	LD	HL,aa	21aa	Z16	
DEC	(IX+d)	DD35d	F23	LD	I,A	ED47	Z9	
DEC	(IY+d)	DD35d	F23	LD	I,(aa)	ED48	Z9	
DEC	A	03	F4	LD	IX,aa	DD21aa	Z14	
DEC	B	05	F4	LD	IY,(aa)	FD2Aaa	Z20	
DEC	BC	0B	Z6	LD	IY,aa	FD21aa	Z14	
DEC	C	0D	F4	LD	L,—	TABLE	Z	
DEC	D	15	F4	LD	R,A	ED4F	Z9	
DEC	DE	1B	Z6	LD	SP,(aa)	ED7Baa	Z20	
DEC	E	1D	F4	LD	SP,HL	F9	Z6	
DEC	H	25	F4	LD	SP,IX	DDF9	Z10	
DEC	HL	2B	Z6	LD	SP,IY	DDF9	Z10	
DEC	IX	DD2B	Z10	LD	SP,aa	31aa	Z10	
DEC	IY	DD2B	Z10	LD	—	—	—	
DEC	L	2D	F4	LD	—	—	—	
DEC	SP	3B	Z6	LD	—	—	—	
DI	—	F3	Z4	LD	LDIR	EDB0	S(1)	
DJNZ	d	10d	Z(2)	LD	LDI	ED4A	B8	
EI	—	FB	Z4	LD	LDIR	EDB0	S(1)	
EX	(SP),HL	E3	Z19	LD	NEG	0A	Z4	
EX	(SP),IX	DDE3	Z23	LD	NOT	D	Z4	
EX	(SP),IY	FDE3	Z23	LD	OTDR	EDBB	Q(1)	
EX	AF,AF	08	Z4	LD	OTIR	EDB3	Q(1)	
EX	DE,HL	EB	Z4	LD	OUT	ED79	Z12	
EXX	—	D9	Z4	LD	OUT	(C),A	ED41	Z12
HALT	—	76	Z4	LD	OUT	(C),B	ED49	Z12
IM	0	ED46	Z8	LD	OUT	(C),C	ED41	Z12
IM	1	ED56	Z8	LD	OUT	(C),D	ED51	Z12
IM	2	ED5E	Z8	LD	OUT	(C),E	ED59	Z12
IN	A,(C)	ED78	W12	LD	OUT	(C),H	ED61	Z12
IN	A,(n)	DBn	Z11	LD	OUT	(C),L	ED69	Z12
IN	B,(C)	ED40	W12	LD	OUTD	(n),A	D39	Z11
IN	C,(C)	ED48	W12	LD	OUT I	—	EDAB	P16
IN	D,(C)	ED50	W12	LD	POP	AF	EDAC	P16
IN	E,(C)	ED58	W12	LD	POP	BC	F1	Z10
IN	H,(C)	ED60	W12	LD	POP	DE	C1	Z10
IN	L,(C)	ED68	W12	LD	POP	HL	D1	Z10
INC	(HL)	34	E11	LD	POP	IX	E1	Z10
INC	(IX+d)	DD34d	E23	LD	POP	IY	DDE1	Z14
INC	(IY+d)	DD34d	E23	LD	PUSH	AF	FDE1	Z14
INC	A	3C	E4	LD	PUSH	BC	F5	Z11
INC	B	04	E4	LD	PUSH	DE	C5	Z11
INC	BC	03	Z6	LD	PUSH	DE	D5	Z11
INC	C	0C	E4	LD	PUSH	HL	E5	Z11
INC	D	14	E4	LD	PUSH	IX	DDE5	Z15
INC	DE	13	Z6	LD	PUSH	IY	FDE5	Z15
INC	E	1C	E4	LD	RES	—	TABLE	Z
INC	H	24	E4	LD	RET	C9	Z10	
INC	HL	23	Z6	LD	RET	M	D8	Z(4)
INC	IX	DD23	Z10	LD	RET	NC	F8	Z(4)
INC	IY	DD23	Z10	LD	RET	NZ	D0	Z(4)
INC	L	2C	E4	LD	RET	P	C0	Z(4)
INC	SP	33	Z6	LD	RET	PE	F0	Z(4)
IND	—	EDAA	P16	LD	RET	PO	E8	Z(4)
INDR	—	EDBA	Q(1)	LD	RET	Z	C8	Z(4)
INI	—	EDBA	P16	LD	RETI	—	ED4D	Z14
INIR	—	EDB2	Q(1)	LD	RETN	—	ED45	Z14
JP	(HL)	E9	Z4	LD	RLA	—	TABLE	K
JP	(IX)	DDE9	Z8	LD	RLC	—	17	K4
JP	(IY)	FDE9	Z8	LD	RLCA	—	07	J4
JP	aa	C3aa	Z10	LD	RLD	—	ED6F	L18
JP	C,aa	DAaa	Z10	LD	RR	—	TABLE	K
JP	M,aa	FAaa	Z10	LD	RRR	—	1F	J4
JP	NC,aa	D2aa	Z10	LD	RRC	—	TABLE	K
JP	NZ,aa	C2aa	Z10	LD	RRD	—	ED67	K4
JP	P,aa	F2aa	Z10	LD	RST	00H	CF	Z11
JP	PE,aa	EAaa	Z10	LD	RST	08H	C7	Z11
JP	PO,aa	E2aa	Z10	LD	RST	10H	DF	Z11
JP	Z,aa	CAaa	Z10	LD	RST	18H	F7	Z11
JR	C,d	38d	Z(3)	LD	RST	20H	EF	Z11
JR	d	18d	Z12	LD	RST	28H	F7	Z11
JR	NC,d	30d	Z(3)	LD	RST	30H	FF	Z11
JR	NZ,d	20d	Z(3)	LD	RST	38H	FF	Z11
JR	Z,d	28d	Z(3)	LD	SBC	A,—	TABLE	B
LD	(BC),A	02	Z7	LD	SBC	HL,BC	ED42	I15
LD	(DE),A	12	Z7	LD	SBC	HL,DE	ED52	I15
LD	(HL),A	77	Z7	LD	SBC	HL,HL	ED62	I15
LD	(HL),B	70	Z7	LD	SBC	HL,SP	ED72	I15
LD	(HL),C	71	Z7	LD	SCF	—	37	O4
LD	(HL),D	72	Z7	LD	SET	—	TABLE	Z
LD	(HL),E	73	Z7	LD	SLA	—	TABLE	K
LD	(HL),H	74	Z7	LD	SRL	—	TABLE	K
LD	(HL),L	75	Z7	LD	SRL	—	TABLE	K
LD	(HL),n	36n	Z10	LD	SUB	—	TABLE	B
LD	(IX+d),A	DD77d	Z19	LD	XOR	—	TABLE	D
LD	(IX+d),B	DD70d	Z19	LD	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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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	V	S	0	
E	=	Z	V	S	1	0	
F	=	=	=	=	0	H	
G	C	Z	V	S	1	U	
H	C	Z	V	S	1	U	
I	J	C	=	=	=	0	
J	C	=	=	=	=	0	
K	C	Z	P	S	0	0	
L	C	Z	P	S	0	H	
M	=	=	=	=	=	1	
N	=	=	=	=	=	1	
O	1	=	=	=	=	0	
P	U	U	U	U	U	U	(1)
Q	U	U	U	U	U	0	(2)
R	=	U	F	F	S	0	(3)
S	=	F	F	S	1	0	(4)
T	=	F	F	S	0	1	(5)
U	=	Z	U	P	0	U	
V	=	Z	U	P	0	U	
W	=	Z	U	P	0	U	
X	=	=	=	=	=	=	
Y	=	=	=	=	=	=	
Z	=	=	=	=	=	=	





# Z80 CPU

## MICROPROCESSOR INSTANT REFERENCE CARD

LSD →

## Single-Byte-Opcode to Instruction Conversion

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
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
2 JR NZ,n	LD HL,nn	LD (nn),A	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
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
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
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
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
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
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
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
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
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
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
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
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
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
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)	%609 ADD XY,BC	%Cbd06 RLC (XY+d)
CB01 RLC C	ED41 OUT (C),B	%619 ADD XY,DE	%Cbd0E RRC (XY+d)
CB02 RLC D	ED42 SBC HL,BC	%621aa LD XY,aa	%Cbd16 RL (XY+d)
CB03 RLC E	ED43aa LD (aa),BC	%622aa LD (aa),XY	%Cbd1E RR (XY+d)
CB04 RLC H	ED44 NEG	%623 INC XY	%Cbd26 SRA (XY+d)
CB05 RLC L	ED45 RETN	%629 ADD XY,XY	%Cbd2E SRA (XY+d)
CB06 RLC (HL)	ED46 IM 0	%62aaa LD XY,(aa)	%Cbd3E SRL (XY+d)
CB07 RLC A	ED47 LD I,A	%62B DEC XY	%Cbd46 BIT 0,(XY+d)
CB08 RRC B	ED48 IN C,(C)	%634d INC (XY+d)	%Cbd4E BIT 1,(XY+d)
CB09 RRC C	ED49 OUT C,(C)	%635d DEC (XY+d)	%Cbd56 BIT 2,(XY+d)
CB0A RRC D	ED4A ADC HL,BC	%636dn LD (XY+d),n	%Cbd5E BIT 3,(XY+d)
CB0B RRC E	ED4Baa LD BC,(aa)	%639 ADD XY,SP	%Cbd66 BIT 4,(XY+d)
CB0C RRC H	ED4D RETI	%646d LD B,(XY+d)	%Cbd6E BIT 5,(XY+d)
CB0D RRC L	ED4F LD R,A	%64Ed LD C,(XY+d)	%Cbd76 BIT 6,(XY+d)
CB0E RRC (HL)	ED50 IN D,(C)	%656d LD D,(XY+d)	%Cbd7E BIT 7,(XY+d)
CB0F RRC A	ED51 OUT (C),D	%65Ed LD E,(XY+d)	%Cbd86 RES 0,(XY+d)
CB10 RLC B	ED52 SBC HL,DE	%666d LD H,(XY+d)	%Cbd8E RES 1,(XY+d)
CB11 RLC C	ED53aa LD (aa),DE	%66Ed LD L,(XY+d)	%Cbd96 RES 2,(XY+d)
CB12 RLC D	ED56 IM 1	%670d LD (XY+d),B	%Cbd9E RES 3,(XY+d)
CB13 RLC E	ED57 LD A,I	%671d LD (XY+d),C	%CbdA6 RES 4,(XY+d)
CB14 RLC H	ED58 IN E,(C)	%672d LD (XY+d),D	%CbdA6 RES 5,(XY+d)
CB15 RLC L	ED59 OUT (C),E	%673d LD (XY+d),E	%CbdB6 RES 6,(XY+d)
CB16 RLC (HL)	ED5A ADC HL,DE	%674d LD (XY+d),H	%CbdB6 RES 7,(XY+d)
CB17 RLC A	ED5Baa LD DE,(aa)	%675d LD (XY+d),L	%CbdC6 SET 0,(XY+d)
CB18 RRC B	ED5E IM 2	%677d LD A,(XY+d)	%CbdC6 SET 1,(XY+d)
CB19 RRC C	ED5F LD A,R	%67Ed LD (XY+d),A	%CbdCE SET 2,(XY+d)
CB1A RRC D	ED60 IN H,(C)	%686d ADD A,(XY+d)	%CbdDE SET 3,(XY+d)
CB1B RRC E	ED61 OUT C,(H)	%68Ed ADD A,(XY+d)	%CbdDE SET 4,(XY+d)
CB1C RRC H	ED62 SBC HL,HL	%696d SUB (XY+d)	%CbdEE SET 5,(XY+d)
CB1D RRC L	ED67 RRD	%69Ed SBC A,(XY+d)	%CbdFE SET 6,(XY+d)
CB1E RRC (HL)	ED68 IN L,(C)	%6A6d AND (XY+d)	%CbdFE SET 7,(XY+d)
CB1F RRC A	ED69 OUT (C),L	%6AEd XOR (XY+d)	
CB20 SLA B	ED6A ADC HL,HL	%6B6d OR (XY+d)	
CB21 SLA C	ED6F RLD	%6BEd CP (XY+d)	
CB22 SLA D	ED72 SBC HL,SP		%E1 POP XY
CB23 SLA E	ED73aa LD (aa),SP	% means DD or FD and for DD, XY means IX for FD, XY means IY	%E5 EX (SP),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	ED80 LDI		
CB29 SRA C	ED81 CPI		
CB2A SRA D	ED82 INI		
CB2B SRA E	ED83 OUTI		
CB2C SRA H	ED88 LDD		
CB2D SRA L	ED89 CPD		
CB2E SRA (HL)	EDAA IND		
CB2F SRA A	EDAB OUTD		
CB30 SRL B	EDB0 LDIR		
CB31 SRL C	EDB1 CPIR		
CB32 SRL D	EDB2 INIR		
CB33 SRL E	EDB3 OTIR		
CB34 SRL H	EDB8 LDDR		
CB35 SRL L	EDB9 INDR		
CB36 SRL (HL)	EDBA CPDR		
CB37 SRL A	EDBB OTDR		
CB3F see BIT			
CB40 see RES			
CBFF see SET			

## ASCII Character Set

MSD	0	1	2	3	4	5	6	7
LSD	000	001	010	011	100	101	110	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	

## 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
A = B	JP Z,YES
A > B	JP NZ,YES
A >= B	JP NC,YES
A > B	JP C,3
A >= B	JP NC,3

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.

A11	1	40	A10
A12	2	39	A9
A13	3	38	A8
A14	4	37	A7
A15	5	36	A6
A16	6	35	A5
A17	7	34	A4
A18	8	33	A3
A19	9	32	A2
A20	10	31	A1
A21	11	30	A0
A22	12	29	GND
A23	13	28	REFRESH
A24	14	27	MT
A25	15	26	RESET
A26	16	25	BUSIO
A27	17	24	WAIT
A28	18	23	BUSAK
A29	19	22	WR
A30	20	21	RD

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	PGMR CTR PC

## Status Flags

MSB	LSB
S	Z
H	P
V	N
C	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 NMI does a RST 66H regardless of IFF1, 2 (Interrupt Flip Flop).

If interrupts are enabled (IFF1=1), low level sensitive 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

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

IFF1 and IFF2 are both cleared by INT or DI. Both are set by EI.

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.

## Registers

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.

## General Instruction Description (except shifts)

ADC x,y Add y+CY to x.  
ADD x,y Add y to x.  
AND x,A AND x to A.  
BIT b,x Test bit b of x.  
CALL x,c 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.  
CPL Complement 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,A 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,r 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 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,A Subtract x from A.  
XOR x,XOR x to A.