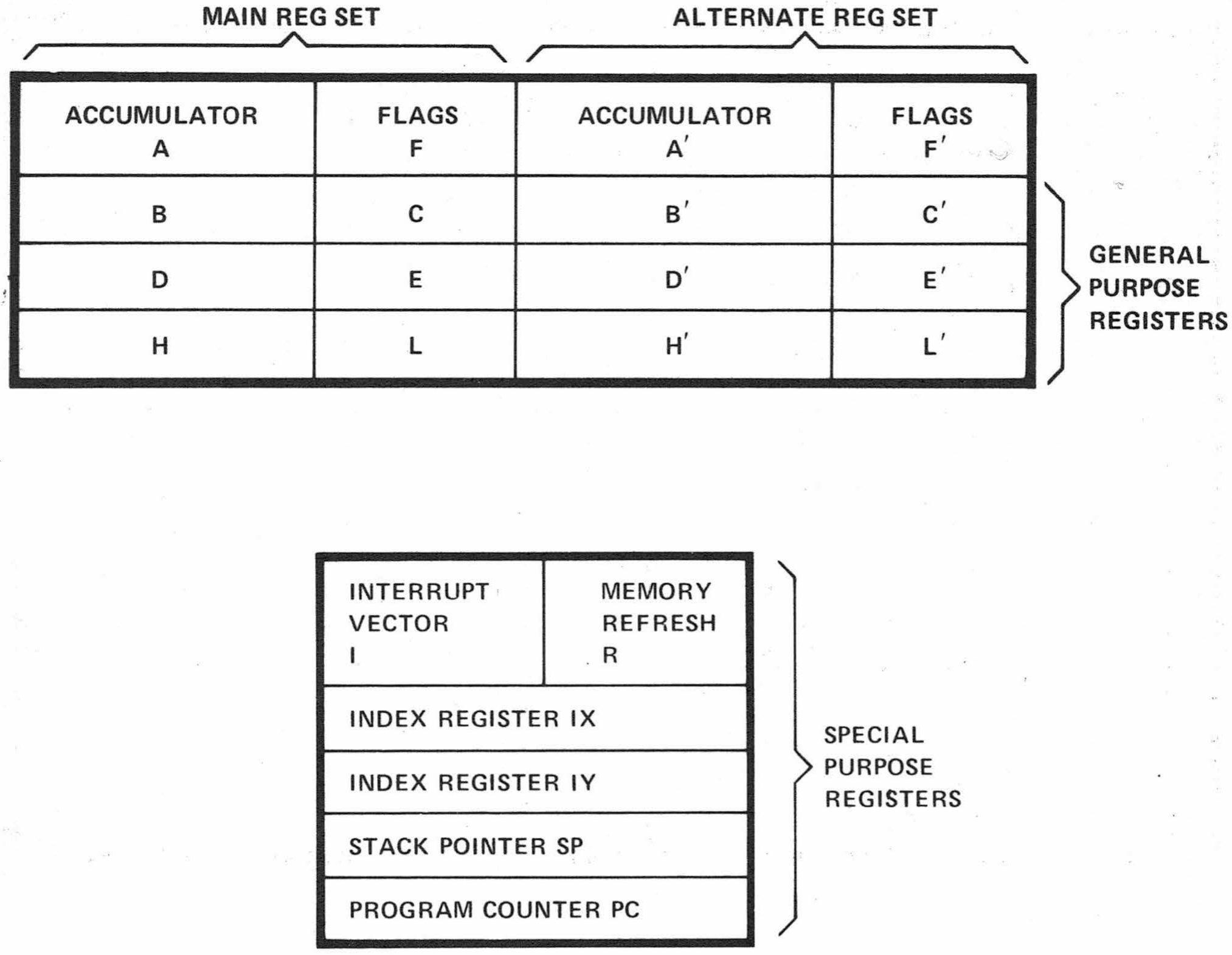
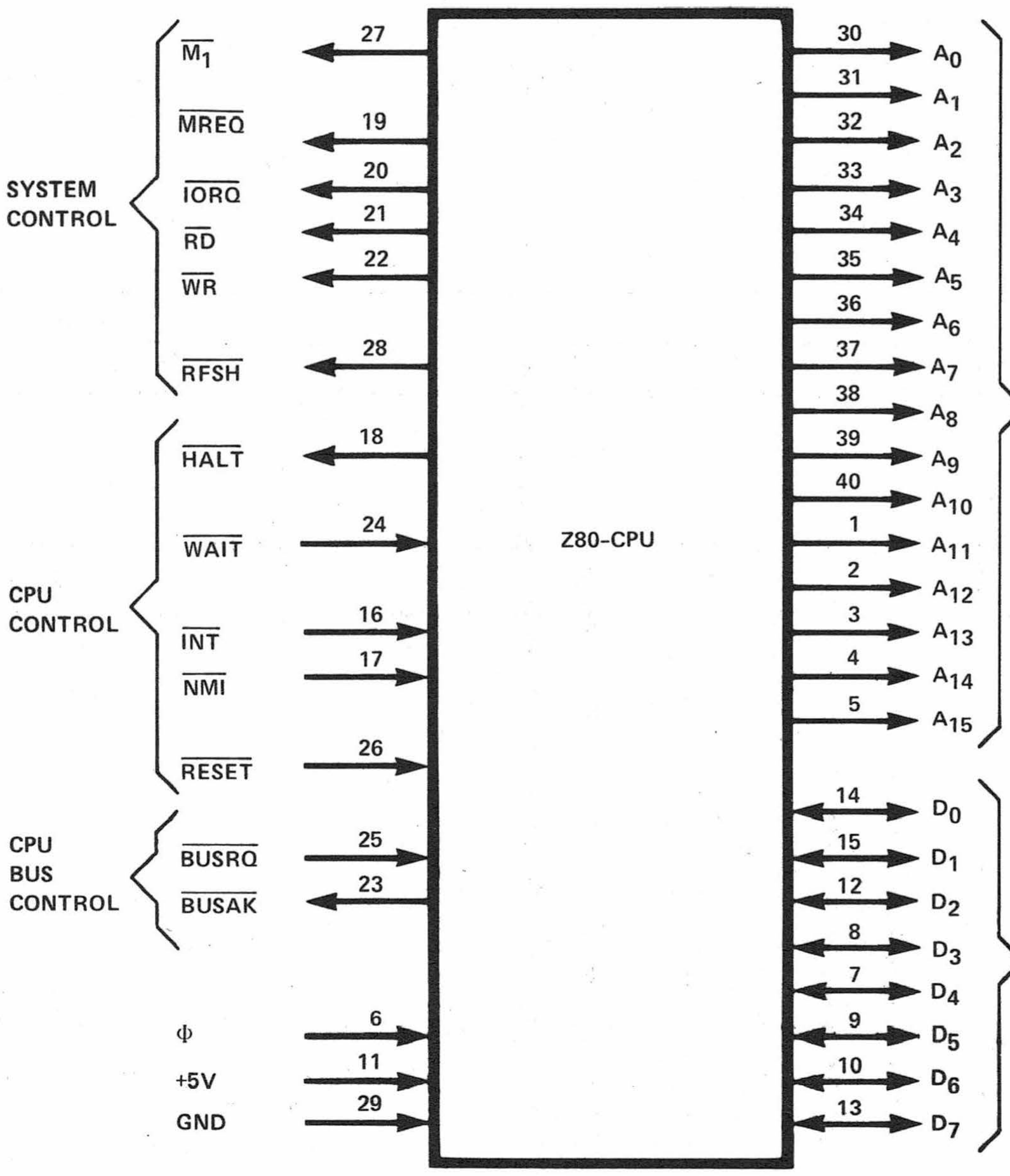
ZILOG Z-80 CPU

Programming Reference Card



Z80-CPU
REGISTER CONFIGURATION



CPU PIN-OUTS

ASCII CHARACTER SET (7-BIT CODE)

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	ENG	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	VS	/	?	O	←	o	DEL

HEXADECIMAL COLUMNS					
6	5	4	3	2	1
HEX = DEC	HEX = DEC	HEX = DEC	HEX = DEC	HEX = DEC	HEX = DEC
0 0	0 0	0 0	0 0	0 0	0 0
1 1,048,576	1 65,536	1 4,096	1 256	1 16	1 1
2 2,097,152	2 131,072	2 8,192	2 512	2 32	2 2
3 3,145,728	3 196,608	3 12,288	3 768	3 48	3 3
4 4,194,304	4 262,144	4 16,384	4 1,024	4 64	4 4
5 5,242,880	5 327,680	5 20,480	5 1,280	5 80	5 5
6 6,291,456	6 393,216	6 24,576	6 1,536	6 96	6 6
7 7,340,032	7 458,752	7 28,672	7 1,792	7 112	7 7
8 8,388,608	8 524,288	8 32,768	8 2,048	8 128	8 8
9 9,437,184	9 589,824	9 36,864	9 2,304	9 144	9 9
A 10,485,760	A 655,360	A 40,960	A 2,560	A 160	A 10
B 11,534,336	B 720,896	B 45,056	B 2,816	B 176	B 11
C 12,582,912	C 786,432	C 49,152	C 3,072	C 192	C 12
D 13,631,488	D 851,968	D 53,248	D 3,328	D 208	D 13
E 14,680,064	E 917,504	E 57,344	E 3,584	E 224	E 14
F 15,728,640	F 983,040	F 61,440	F 3,840	F 240	F 15
0 1 2 3	4 5 6 7	0 1 2 3	4 5 6 7	0 1 2 3	4 5 6 7
BYTE		BYTE		BYTE	

POWERS OF 2

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

$2^0 = 16^0$
$2^4 = 16^1$
$2^8 = 16^2$
$2^{12} = 16^3$
$2^{16} = 16^4$
$2^{20} = 16^5$
$2^{24} = 16^6$
$2^{28} = 16^7$
$2^{32} = 16^8$
$2^{36} = 16^9$
$2^{40} = 16^{10}$
$2^{44} = 16^{11}$
$2^{48} = 16^{12}$
$2^{52} = 16^{13}$
$2^{56} = 16^{14}$
$2^{60} = 16^{15}$

POWERS OF 16

16^n	n
1	0
16	1
256	2
4 096	3
65 536	4
1 048 576	5
16 777 216	6
268 435 456	7
4 294 967 296	8
68 719 476 736	9
1 099 511 627 776	10
17 592 186 044 416	11
281 474 976 710 656	12
4 503 599 627 370 496	13
72 057 594 037 927 936	14
1 152 921 504 606 846 976	15

INSTRUCTION	C	Z	P/V	S	N	H	COMMENTS
ADD A, s; ADC A, s	↕	↕	V	↕	0	↕	8-bit add or add with carry
SUB s; SBC A, s, CP s, NEG	↕	↕	V	↕	1	↕	8-bit subtract, subtract with carry, compare and negate accumulator
AND s	0	↕	P	↕	0	1	Logical operations
OR s; XOR s	0	↕	P	↕	0	0	And sets different flags
INC s	●	↕	V	↕	0	↕	8-bit increment
DEC m	●	↕	V	↕	1	↕	8-bit decrement
ADD DD, ss	↕	●	●	●	0	X	16-bit add
ADC HL, ss	↕	↕	V	↕	0	X	16-bit add with carry
SBC HL, ss	↕	↕	V	↕	1	X	16-bit subtract with carry
RLA; RLCA, RRA, RRCA	↕	●	●	●	0	0	Rotate accumulator
RL m; RLC m; RR m; RRC m SLA m; SRA m; SRL m	↕	↕	P	↕	0	0	Rotate and shift location m
RLD, RRD	●	↕	P	↕	0	0	Rotate digit left and right
DAA	↕	↕	P	↕	●	↕	Decimal adjust accumulator
CPL	●	●	●	●	1	1	Complement accumulator
SCF	1	●	●	●	0	0	Set carry
CCF	↕	●	●	●	0	X	Complement carry
IN r, (C)	●	↕	P	↕	0	0	Input register indirect
INI; IND; OUTI; OUTD	●	↕	X	X	1	X	Block input and output
INIR; INDR; OTIR; OTDR	●	1	X	X	1	X	Z = 0 if B ≠ 0 otherwise Z = 1
LDI, LDD	●	X	↕	X	0	0	Block transfer instructions
LDIR, LDDR	●	X	0	X	0	0	P/V = 1 if BC ≠ 0, otherwise P/V = 0
CPI, CPIR, CPD, CPDR	●	↕	↕	↕	1	X	Block search instructions Z = 1 if A = (HL), otherwise Z = 0 P/V = 1 if BC ≠ 0, otherwise P/V = 0
LD A, I; LD A, R	●	↕	IFF	↕	0	0	The content of the interrupt enable flip-flop (IFF) is copied into the P/V flag
BIT b, s	●	↕	X	X	0	1	The complement of bit b of location is copied into the Z flag
NEG	↕	↕	V	↕	1	↕	Negate accumulator

The following notation is used in this table:

SYMBOL	OPERATION
C	Carry/link flag. C=1 if the operation produced a carry from the MSB of the operand or result.
Z	Zero flag. Z=1 if the result of the operation is zero.
S	Sign flag. S=1 if the MSB of the result is one.
P/V	Parity or overflow flag. Parity (P) and overflow (V) share the same flag. Logical operations affect this flag with the parity of the result while arithmetic operations affect this flag with the overflow of the result. If P/V holds parity, P/V=1 if the result of the operation is even, P/V=0 if result is odd. If P/V holds overflow, P/V=1 if the result of the operation produced an overflow.
H	Half-carry flag. H=1 if the add or subtract operation produced a carry into or borrow from bit 4 of the accumulator.
N	Add/Subtract flag. N=1 if the previous operation was a subtract.
	H and N flags are used in conjunction with the decimal adjust instruction (DAA) to properly correct the result into packed BCD format following addition or subtraction using operands with packed BCD format.
↕	The flag is affected according to the result of the operation.
●	The flag is unchanged by the operation.
0	The flag is reset by the operation.
1	The flag is set by the operation.
X	The flag is a "don't care."
V	P/V flag affected according to the overflow result of the operation.
P	P/V flag affected according to the parity result of the operation.
r	Any one of the CPU registers A, B, C, D, E, H, L.
s	Any 8-bit location for all the addressing modes allowed for the particular instruction.
ss	Any 16-bit location for all the addressing modes allowed for that instruction.
ii	Any one of the two index registers IX or IY.
R	Refresh counter.
n	8-bit value in range <0, 255>.
nn	16-bit value in range <0, 65535>.
m	Any 8-bit location for all the addressing modes allowed for the particular instruction.

SUMMARY OF FLAG OPERATION

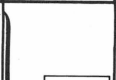
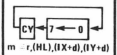
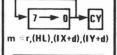
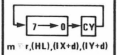
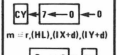
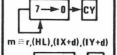
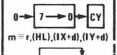
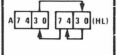
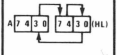
S	Z	X	H	X	P/V	N	C

Sequence of flags in F register

MNEMONIC	SYMBOLIC OPERATION	FLAGS						OP-CODE	NO. OF T CYCLES	COMMENTS
		C		P/V	S	N	H	76 543 210		
LD r, r'	r ← r'	•	•	•	•	•	•	01 r r'	4	r, r' Reg.
LD r, n	r ← n	•	•	•	•	•	•	00 r 110	7	000 B
								← n →		001 C
LD r, (HL)	r ← (HL)	•	•	•	•	•	•	01 r 110	7	010 D
LD r, (IX+d)	r ← (IX+d)	•	•	•	•	•	•	11 011 101	19	011 E
								01 r 110		100 H
								← d →		101 L
LD r, (IY+d)	r ← (IY+d)	•	•	•	•	•	•	11 111 101	19	111 A
								01 r 110		
								← d →		
LD (HL), r	(HL) ← r	•	•	•	•	•	•	01 110 r	7	
LD (IX+d), r	(IX+d) ← r	•	•	•	•	•	•	11 011 101	19	
								01 110 r		
								← d →		
LD (IY+d), r	(IY+d) ← r	•	•	•	•	•	•	11 111 101	19	
								01 110 r		
								← d →		
LD (HL), n	(HL) ← n	•	•	•	•	•	•	00 110 110	10	
								← n →		
LD (IX+d), n	(IX+d) ← n	•	•	•	•	•	•	11 011 101	19	
								00 110 110		
								← d →		
								← n →		
LD (IY+d), n	(IY+d) ← n	•	•	•	•	•	•	11 111 101	19	
								00 110 110		
								← d →		
								← n →		
LD A, (BC)	A ← (BC)	•	•	•	•	•	•	00 001 010	7	
LD A, (DE)	A ← (DE)	•	•	•	•	•	•	00 011 010	7	
LD A, (nn)	A ← (nn)	•	•	•	•	•	•	00 111 010	13	
								← n →		
								← n →		
LD (BC), A	(BC) ← A	•	•	•	•	•	•	00 000 010	7	
LD (DE), A	(DE) ← A	•	•	•	•	•	•	00 010 010	7	
LD (nn), A	(nn) ← A	•	•	•	•	•	•	00 110 010	13	
								← n →		
								← n →		
LD A, I	A ← I	•	↕	IFF	↕	0	0	11 101 101	9	
								01 010 111		
LD A, R	A ← R	•	↕	IFF	↕	0	0	11 101 101	9	
								01 011 111		
LD I, A	I ← A	•	•	•	•	•	•	11 101 101	9	
								01 000 111		
LD R, A	R ← A	•	•	•	•	•	•	11 101 101	9	
								01 001 111		
LD dd, nn	dd ← nn	•	•	•	•	•	•	00 dd0 001	10	dd Pair
								← n →		00 BC
								← n →		01 DE
LD IX, nn	IX ← nn	•	•	•	•	•	•	11 011 101	14	10 HL
								00 100 001		11 SP
								← n →		
								← n →		
LD IY, nn	IY ← nn	•	•	•	•	•	•	11 111 101	14	
								00 100 001		
								← n →		
								← n →		
LD HL, (nn)	H ← (nn+1)	•	•	•	•	•	•	00 101 010	16	
	L ← (nn)							← n →		
								← n →		
LD dd, (nn)	dd_H ← (nn+1)	•	•	•	•	•	•	11 101 101	20	
	dd_L ← (nn)							01 dd1 011		
								← n →		
								← n →		
LD IX, (nn)	IX_H ← (nn+1)	•	•	•	•	•	•	11 011 101	20	
	IX_L ← (nn)							00 101 010		
								← n →		
								← n →		
LD IY, (nn)	IY_H ← (nn+1)	•	•	•	•	•	•	11 111 101	20	
	IY_L ← (nn)							00 101 010		
								← n →		
								← n →		
LD (nn), HL	(nn+1) ← H	•	•	•	•	•	•	00 100 010	16	
	(nn) ← L							← n →		
								← n →		

MNEMONIC	SYMBOLIC OPERATION	FLAGS						OP-CODE	NO. OF T CYCLES	COMMENTS
		C	Z	P/V	S	N	H	76 543 210		
LD (nn), dd	(nn+1) ← dd_H (nn) ← dd_L	•	•	•	•	•	•	11 101 101 01 dd0 011 ← n → ← n →	20	
LD (nn), IX	(nn+1) ← IX_H (nn) ← IX_L	•	•	•	•	•	•	11 011 101 00 100 010 ← n → ← n →	20	
LD (nn), IY	(nn+1) ← IY_H (nn) ← IY_L	•	•	•	•	•	•	11 111 101 00 100 010 ← n → ← n →	20	
LD SP, HL	SP ← HL	•	•	•	•	•	•	11 111 001	6	
LD SP, IX	SP ← IX	•	•	•	•	•	•	11 011 101 11 111 001	10	
LD SP, IY	SP ← IY	•	•	•	•	•	•	11 111 101 11 111 001	10	
PUSH qq	(SP-2) ← qq_L (SP-1) ← qq_H	•	•	•	•	•	•	11 qq0 101	11	qq | Pair 00 | BC 01 | DE 10 | HL 11 | AF
PUSH IX	(SP-2) ← IX_L (SP-1) ← IX_H	•	•	•	•	•	•	11 011 101 11 100 101	15	
PUSH IY	(SP-2) ← IY_L (SP-1) ← IY_H	•	•	•	•	•	•	11 111 101 11 100 101	15	
POP qq	qq_H ← (SP+1) qq_L ← (SP)	•	•	•	•	•	•	11 qq0 001	10	
POP IX	IX_H ← (SP+1) IX_L ← (SP)	•	•	•	•	•	•	11 011 101 11 100 001	14	
POP IY	IY_H ← (SP+1) IY_L ← (SP)	•	•	•	•	•	•	11 111 101 11 100 001	14	
EX DE, HL	DE ↔ HL	•	•	•	•	•	•	11 101 011	4	
EX AF, AF′	AF ↔ AF′	•	•	•	•	•	•	00 001 000	4	
EXX	BC ↔ BC′ DE ↔ DE′ HL ↔ HL′	•	•	•	•	•	•	11 011 001	4	Register bank and auxiliary register bank exchange
EX (SP), HL	H ↔ (SP+1) L ↔ (SP)	•	•	•	•	•	•	11 100 011	19	
EX (SP), IX	IX_H ↔ (SP+1) IX_L ↔ (SP)	•	•	•	•	•	•	11 011 101 11 100 011	23	
EX (SP), IY	IY_H ↔ (SP+1) IY_L ↔ (SP)	•	•	•	•	•	•	11 111 101 11 100 011	23	
LDI	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC-1	•	•	① ↕	•	0	0	11 101 101 10 100 000	16	Load (HL) into (DE), increment the pointers and decrement the byte counter (BC)
LDIR	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC-1 Repeat until BC = 0	•	•	0	•	0	0	11 101 101 10 110 000	21 16	If BC ≠ 0 If BC = 0
LDD	(DE) ← (HL) DE ← DE-1 HL ← HL-1 BC ← BC-1	•	•	① ↕	•	0	0	11 101 101 10 101 000	16	
LDDR	(DE) ← (HL) DE ← DE-1 HL ← HL-1 BC ← BC-1 Repeat until BC = 0	•	•	0	•	0	0	11 101 101 10 111 000	21 16	If BC ≠ 0 If BC = 0
CPI	A - (HL) HL ← HL+1 BC ← BC-1	•	② ↕	① ↕	↕	1	↕	11 101 101 10 100 001	16	
CPIR	A - (HL) HL ← HL+1 BC ← BC-1 Repeat until A = (HL) or BC = 0	•	② ↕	① ↕	↕	1	↕	11 101 101 10 110 001	21 16	If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)
CPD	A - (HL) HL ← HL-1 BC ← BC-1	•	② ↕	① ↕	↕	1	↕	11 101 101 10 101 001	16	

MNEMONIC	SYMBOLIC OPERATION	FLAGS C	Z	P/V	S	N	H	OP-CODE 76 543 210	NO. OF T CYCLES	COMMENTS
			②	①						
CPDR	A - (HL) HL ← HL-1 BC ← BC-1 Repeat until A = (HL) or BC = 0	•	↕	↕	↕	1	↕	11 101 101 10 111 001	21 16	If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)
ADD A, r	A ← A + r	↕	↕	V	↕	0	↕	10 [000] r	4	r | Reg. 000 | B 001 | C 010 | D 011 | E 100 | H 101 | L 111 | A
ADD A, n	A ← A + n	↕	↕	V	↕	0	↕	11 [000] 110 ← n →	7	
ADD A, (HL)	A ← A + (HL)	↕	↕	V	↕	0	↕	10 [000] 110	7	
ADD A, (IX+d)	A ← A + (IX+d)	↕	↕	V	↕	0	↕	11 011 101 10 [000] 110 ← d →	19	
ADD A, (IY+d)	A ← A+(IY+d)	↕	↕	V	↕	0	↕	11 111 101 10 [000] 110 ← d →	19	
ADC A,s	A ← A + s + CY	↕	↕	V	↕	0	↕	[001]		s is any of r, n, (HL), (IX+d), (IY+d) as shown for ADD instruction
SUB s	A ← A - s	↕	↕	V	↕	1	↕	[010]		
SBC A,s	A ← A - s - CY	↕	↕	V	↕	1	↕	[011]		
AND s	A ← A ∧ s	0	↕	P	↕	1	1	[100]		
OR s	A ← A ∨ s	0	↕	P	↕	0	0	[110]		
XOR s	A ← A ⊕ s	0	↕	P	↕	0	0	[101]		The indicated bits replace the 000 in the ADD set above.
CP s	A - s	↕	↕	V	↕	1	↕	[111]		
INC r	r ← r + 1	•	↕	V	↕	0	↕	00 r [100]	4	
INC (HL)	(HL) ← (HL)+1	•	↕	V	↕	0	↕	00 110 [100]	11	
INC (IX+d)	(IX+d) ← (IX+d)+1	•	↕	V	↕	0	↕	11 011 101 00 110 [100] ← d →	23	
INC (IY+d)	(IY+d) ← (IY+d)+1	•	↕	V	↕	0	↕	11 111 101 00 110 [100] ← d →	23	
DEC m	m ← m-1	•	↕	V	↕	1	↕	[101]		m is any of r, (HL), (IX+d), (IY+d) as shown for INC. Same format and states as INC. Replace 100 with 101 in OP code.
ADD HL, ss	HL ← HL+ss	↕	•	•	•	0	X	00 ss1 001	11	ss | Reg. 00 | BC 01 | DE 10 | HL 11 | SP
ADC HL, ss	HL←HL+ss+CY	↕	↕	V	↕	0	X	11 101 101 01 ss1 010	15	
SBC HL, ss	HL←HL-ss-CY	↕	↕	V	↕	1	X	11 101 101 01 ss0 010	15	
ADD IX, pp	IX ← IX + pp	↕	•	•	•	0	X	11 011 101 00 pp1 001	15	pp | Reg. 00 | BC 01 | DE 10 | IX 11 | SP
ADD IY, rr	IY← IY+rr	↕	•	•	•	0	X	11 111 101 00 rr1 001	15	rr | Reg. 00 | BC 01 | DE 10 | IY 11 | SP
INC ss	ss ← ss + 1	•	•	•	•	•	•	00 ss0 011	6	
INC IX	IX ← IX + 1	•	•	•	•	•	•	11 011 101 00 100 011	10	
INC IY	IY ← IY + 1	•	•	•	•	•	•	11 111 101 00 100 011	10	
DEC ss	ss ← ss -1	•	•	•	•	•	•	00 ss1 011	6	
DEC IX	IX ← IX -1	•	•	•	•	•	•	11 011 101 00 101 011	10	
DEC IY	IY ← IY -1	•	•	•	•	•	•	11 111 101 00 101 011	10	
RLCA		↕	•	•	•	0	0	00 000 111	4	Rotate left circular accumulator
RLA		↕	•	•	•	0	0	00 010 111	4	Rotate left accumulator
RRCA		↕	•	•	•	0	0	00 001 111	4	Rotate right circular accumulator
RRA		↕	•	•	•	0	0	00 011 111	4	Rotate right accumulator

MNEMONIC	SYMBOLIC OPERATION	FLAGS						OP-CODE	NO. OF T CYCLES	COMMENTS
		C	Z	P/V	S	N	H	76 543 210		
RLC r	 r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	11 001 011 00 [000] r	8	Rotate left circular register r
RLC (HL)		↕	↕	P	↕	0	0	11 001 011 00 [000] 110	15	r | Reg. 000 | B 001 | C 010 | D 011 | E 100 | H 101 | L 111 | A
RLC (IX+d)		↕	↕	P	↕	0	0	11 011 101 11 001 011 ← d → 00 [000] 110	23	
RLC (IY+d)		↕	↕	P	↕	0	0	11 111 101 11 001 011 ← d → 00 [000] 110	23	
RL m	 m = r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	[010]		Instruction format and states are as shown for RLC,m. To form new OP-code replace 000 of RLC, m with shown code
RRC m	 m = r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	[001]		
RR m	 m = r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	[011]		
SLA m	 m = r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	[100]		
SRA m	 m = r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	[101]		
SRL m	 m = r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	[111]		
RLD		•	↕	P	↕	0	0	11 101 101 01 101 111	18	Rotate digit left and right between the accumulator and location (HL). The content of the upper half of the accumulator is unaffected.
RRD		•	↕	P	↕	0	0	11 101 101 01 100 111	18	
DAA	Converts acc. content into packed BCD following add or subtract with packed BCD operands	↕	↕	P	↕	•	↕	00 100 111	4	Decimal adjust accumulator.
CPL	$A \leftarrow \overline{A}$	•	•	•	•	1	1	00 101 111	4	Complement accumulator (one's complement)
NEG	A ← 0-A	↕	↕	V	↕	1	↕	11 101 101 01 000 100	8	Negate acc. (two's complement)
CCF	$CY \leftarrow \overline{CY}$	↕	•	•	•	0	X	00 111 111	4	Complement carry flag
SCF	CY ← 1	1	•	•	•	0	0	00 110 111	4	Set carry flag
NOP	No operation	•	•	•	•	•	•	00 000 000	4	
HALT	CPU halted	•	•	•	•	•	•	01 110 110	4	
DI	IFF ← 0	•	•	•	•	•	•	11 110 011	4	
EI	IFF ← 1	•	•	•	•	•	•	11 111 011	4	
IM 0	Set interrupt mode 0	•	•	•	•	•	•	11 101 101 01 000 110	8	
IM 1	Set interrupt mode 1	•	•	•	•	•	•	11 101 101 01 010 110	8	
IM 2	Set interrupt mode 2	•	•	•	•	•	•	11 101 101 01 011 110	8	
BIT b, r	$Z \leftarrow \overline{r}_b$	•	↕	X	X	0	1	11 001 011 01 b r	8	r | Reg. 000 | B 001 | C 010 | D 011 | E 100 | H 101 | L 111 | A
BIT b, (HL)	$Z \leftarrow \overline{(HL)}_b$	•	↕	X	X	0	1	11 001 011 01 b 110	12	
BIT b, (IX+d)	$Z \leftarrow \overline{(IX+d)}_b$	•	↕	X	X	0	1	11 011 101 11 001 011 ← d → 01 b 110	20	
BIT b, (IY+d)	$Z \leftarrow \overline{(IY+d)}_b$	•	↕	X	X	0	1	11 111 101 11 001 011 ← d → 01 b 110	20	b | Bit Tested 000 | 0 001 | 1 010 | 2 011 | 3 100 | 4 101 | 5 110 | 6 111 | 7

MNEMONIC	SYMBOLIC OPERATION	FLAGS						OP-CODE	NO. OF T CYCLES	COMMENTS
		C	Z	P/V	S	N	H	76 543 210		
SET b, r	$r_b \leftarrow 1$	•	•	•	•	•	•	11 001 011 11 b r	8	
SET b, (HL)	$(HL)_b \leftarrow 1$	•	•	•	•	•	•	11 001 011 11 b 110	15	
SET b, (IX+d)	$(IX+d)_b \leftarrow 1$	•	•	•	•	•	•	11 011 101 11 001 011 ← d → 11 b 110	23	
SET b, (IY+d)	$(IY+d)_b \leftarrow 1$	•	•	•	•	•	•	11 111 101 11 001 011 ← d → 11 b 110	23	
RES b, m	$m_b \leftarrow 0$ m ≡ r, (HL), (IX+d), (IY+d)							10		To form new OP-code replace 11 of SET b, m with 10 . Flags and time states for SET instruction
JP nn	PC ← nn	•	•	•	•	•	•	11 000 011 ← n → ← n →	10	
JP cc, nn	If condition is true PC ← nn, otherwise continue	•	•	•	•	•	•	11 cc 010 ← n → ← n →	10	cc | Condition 000 | NZ non zero 001 | Z zero 010 | NC non carry 011 | C carry 100 | PO parity odd 101 | PE parity even 110 | P sign positive 111 | M sign negative
JR e	PC ← PC + e	•	•	•	•	•	•	00 011 000 ← e-2 →	12	
JR C, e	If C = 0, continue	•	•	•	•	•	•	00 111 000 ← e-2 →	7	If condition not met
	If C = 1, PC ← PC + e								12	If condition is met
JR NC , e	If C = 1, continue	•	•	•	•	•	•	00 110 000 ← e-2 →	7	If condition not met
	If C = 0, PC ← PC + e								12	If condition is met
JR Z, e	If Z = 0 continue	•	•	•	•	•	•	00 101 000 ← e-2 →	7	If condition not met
	If Z = 1, PC ← PC + e								12	If condition is met
JR NZ, e	If Z = 1, continue	•	•	•	•	•	•	00 100 000 ← e-2 →	7	If condition not met
	If Z = 0, PC ← PC + e								12	If condition met
JP (HL)	PC ← HL	•	•	•	•	•	•	11 101 001	4	
JP (IX)	PC ← IX	•	•	•	•	•	•	11 011 101 11 101 001	8	
JP (IY)	PC ← IY	•	•	•	•	•	•	11 111 101 11 101 001	8	
DJNZ e	B ← B-1 If B = 0, continue	•	•	•	•	•	•	00 010 000 ← e-2 →	8	If B = 0
	If B ≠ 0, PC ← PC + e								13	If B ≠ 0
CALL nn	$(SP-1) \leftarrow PC_H$ $(SP-2) \leftarrow PC_L$ PC ← nn	•	•	•	•	•	•	11 001 101 ← n → ← n →	17	
CALL cc, nn	If condition cc is false continue, otherwise same as CALL nn	•	•	•	•	•	•	11 cc 100 ← n → ← n →	10 17	If cc is false If cc is true
RET	$PC_L \leftarrow (SP)$ $PC_H \leftarrow (SP+1)$	•	•	•	•	•	•	11 001 001	10	
RET cc	If condition cc is false continue, otherwise same as RET	•	•	•	•	•	•	11 cc 000	5 11	If cc is false If cc is true cc | Condition 000 | NZ non zero 001 | Z zero 010 | NC non carry 011 | C carry 100 | PO parity odd 101 | PE parity even 110 | P sign positive 111 | M sign negative
RETI	Return from interrupt	•	•	•	•	•	•	11 101 101 01 001 101	14	
RETN	Return from non maskable interrupt	•	•	•	•	•	•	11 101 101 01 000 101	14	

MNEMONIC	SYMBOLIC OPERATION	FLAGS						OP-CODE	NO. OF T CYCLES	COMMENTS
		C	Z	P/V	S	N	H	76 543 210		
RST p	$(SP-1) \leftarrow PC_H$ $(SP-2) \leftarrow PC_L$ $PC_H \leftarrow 0$ $PC_L \leftarrow P$	•	•	•	•	•	•	11 t 111	11	
IN A, (n)	$A \leftarrow (n)$	•	•	•	•	•	•	11 011 011 ← n →	11	n to $A_0 \sim A_7$ Acc to $A_8 \sim A_{15}$
IN r, (C)	$r \leftarrow (C)$ If r = 110 only the flags will be affected	•	↕	P	↕	0	↕	11 101 101 01 r 000	12	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INI	$(HL) \leftarrow (C)$ $B \leftarrow B-1$ $HL \leftarrow HL + 1$	•	③ ↕	X	X	1	X	11 101 101 10 100 010	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INIR	$(HL) \leftarrow (C)$ $B \leftarrow B-1$ $HL \leftarrow HL + 1$ Repeat until B = 0	•	1	X	X	1	X	11 101 101 10 110 010	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
IND	$(HL) \leftarrow (C)$ $B \leftarrow B-1$ $HL \leftarrow HL-1$	•	③ ↕	X	X	1	X	11 101 101 10 101 010	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INDR	$(HL) \leftarrow (C)$ $B \leftarrow B-1$ $HL \leftarrow HL-1$ Repeat until B = 0	•	1	X	X	1	X	11 101 101 10 111 010	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUT (n), A	$(n) \leftarrow A$	•	•	•	•	•	•	11 010 011 ← n →	11	n to $A_0 \sim A_7$ Acc to $A_8 \sim A_{15}$
OUT (C), r	$(C) \leftarrow r$	•	•	•	•	•	•	11 101 101 01 r 001	12	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUTI	$(C) \leftarrow (HL)$ $B \leftarrow B-1$ $HL \leftarrow HL + 1$	•	③ ↕	X	X	1	X	11 101 101 10 100 011	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OTIR	$(C) \leftarrow (HL)$ $B \leftarrow B-1$ $HL \leftarrow HL + 1$ Repeat until B = 0	•	1	X	X	1	X	11 101 101 10 110 011	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUTD	$(C) \leftarrow (HL)$ $B \leftarrow B-1$ $HL \leftarrow HL-1$	•	③ ↕	X	X	1	X	11 101 101 10 101 011	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OTDR	$(C) \leftarrow (HL)$ $B \leftarrow B-1$ $HL \leftarrow HL-1$ Repeat until B = 0	•	1	X	X	1	X	11 101 101 10 111 011	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$

Notes: r, r' means any of the registers A, B, C, D, E, H, L
ss is any of the register pairs BC, DE, HL, SP
rr is any of the register pairs BC, DE, IY, SP
① P/V flag is 0 if the result of BC–1 = 0, otherwise P/V = 1
② Z flag is 1 if A = (HL), otherwise Z = 0
③ **If the result of B–I = 0, the Z flag is set, otherwise it is reset.**
e represents the extension in the relative addressing mode
e is a signed two's complement number in the range $< -126, 129 >$
e–2 in the op-code provides an effective address of pc+e as PC is incremented by 2 prior to the addition of e.
The notation s_b indicates bit b (0 to 7) of location s.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown
↕ = flag is affected according to the result of the operation.

Z80–CPU INSTRUCTION SET

OBJ CODE	SOURCE STATEMENT		OPERATION
8E	ADC	A,(HL)	Add with Carry Operand to Acc.
DD8E05	ADC	A,(IX+d)	
FD8E05	ADC	A,(IY+d)	
8F	ADC	A,A	
88	ADC	A,B	
89	ADC	A,C	
8A	ADC	A,D	
8B	ADC	A,E	
8C	ADC	A,H	
8D	ADC	A,L	
CE20	ADC	A,n	
ED4A	ADC	HL,BC	Add with Carry Reg. Pair to HL
ED5A	ADC	HL,DE	
ED6A	ADC	HL,HL	
ED7A	ADC	HL,SP	
86	ADD	A,(HL)	Add Operand to Acc.
DD8605	ADD	A,(IX+d)	
FD8605	ADD	A,(IY+d)	
87	ADD	A,A	
80	ADD	A,B	
81	ADD	A,C	
82	ADD	A,D	
83	ADD	A,E	
84	ADD	A,H	
85	ADD	A,L	
C620	ADD	A,n	
09	ADD	HL,BC	Add Reg. Pair to HL
19	ADD	HL,DE	
29	ADD	HL,HL	
39	ADD	HL,SP	
DD09	ADD	IX,BC	Add Reg. Pair to IX
DD19	ADD	IX,DE	
DD29	ADD	IX,IX	
DD39	ADD	IX,SP	
FD09	ADD	IY,BC	Add Reg. Pair to IY
FD19	ADD	IY,DE	
FD29	ADD	IY,IY	
FD39	ADD	IY,SP	
A6	AND	(HL)	Logical 'AND' of Operand and Acc.
DDA605	AND	(IX+d)	
FDA605	AND	(IY+d)	
A7	AND	A	
A0	AND	B	
A1	AND	C	
A2	AND	D	
A3	AND	E	
A4	AND	H	
A5	AND	L	
E620	AND	n	
CB46	BIT	0,(HL)	Test Bit b of Location or Reg.
DDCB0546	BIT	0,(IX+d)	
FDCB0546	BIT	0,(IY+d)	
CB47	BIT	0,A	
CB40	BIT	0,B	
CB41	BIT	0,C	
CB42	BIT	0,D	
CB43	BIT	0,E	
CB44	BIT	0,H	

OBJ CODE	SOURCE STATEMENT		OPERATION
CB45	BIT	0,L	Test Bit b of Location
CB4E	BIT	1,(HL)	or Reg.
DDCB054E	BIT	1,(IX+d)	
FDCB054E	BIT	1,(IY+d)	
CB4F	BIT	1,A	
CB48	BIT	1,B	
CB49	BIT	1,C	
CB4A	BIT	1,D	
CB4B	BIT	1,E	
CB4C	BIT	1,H	
CB4D	BIT	1,L	
CB56	BIT	2,(HL)	
DDCB0556	BIT	2,(IX+d)	
FDCB0556	BIT	2,(IY+d)	
CB57	BIT	2,A	
CB50	BIT	2,B	
CB51	BIT	2,C	
CB52	BIT	2,D	
CB53	BIT	2,E	
CB54	BIT	2,H	
CB55	BIT	2,L	
CB5E	BIT	3,(HL)	
DDCB055E	BIT	3,(IX+d)	
FDCB055E	BIT	3,(IY+d)	
CB5F	BIT	3,A	
CB58	BIT	3,B	
CB59	BIT	3,C	
CB5A	BIT	3,D	
CB5B	BIT	3,E	
CB5C	BIT	3,H	
CB5D	BIT	3,L	
CB66	BIT	4,(HL)	
DDCB0566	BIT	4,(IX+d)	
FDCB0566	BIT	4,(IY+d)	
CB67	BIT	4,A	
CB60	BIT	4,B	
CB61	BIT	4,C	
CB62	BIT	4,D	
CB63	BIT	4,E	
CB64	BIT	4,H	
CB65	BIT	4,L	
CB6E	BIT	5,(HL)	
DDCB056E	BIT	5,(IX+d)	
FDCB056E	BIT	5,(IY+d)	
CB6F	BIT	5,A	
CB68	BIT	5,B	
CB69	BIT	5,C	
C86A	BIT	5,D	
CB6B	BIT	5,E	
CB6C	BIT	5,H	
CB6D	BIT	5,L	
CB76	BIT	6,(HL)	
DDCB0576	BIT	6,(IX+d)	
FDCB0576	BIT	6,(IY+d)	
CB77	BIT	6,A	
CB70	BIT	6,B	
CB71	BIT	6,C	
CB72	BIT	6,D	
CB73	BIT	6,E	
CB74	BIT	6,H	
CB75	BIT	6,L	
CB7E	BIT	7,(HL)	
DDCB057E	BIT	7,(IX+d)	

OBJ CODE	SOURCE STATEMENT		OPERATION
FDCB057E	BIT	7,(IY+d)	Test Bit b Location or Reg.
CB7F	BIT	7,A	
CB78	BIT	7,B	
CB79	BIT	7,C	
CB7A	BIT	7,D	
CB7B	BIT	7,E	
CB7C	BIT	7,H	
CB7D	BIT	7,L	
DC8405	CALL	C,nn	Call Subroutine at Location nn if Condition True
FC8405	CALL	M,nn	
D48405	CALL	NC,nn	
C48405	CALL	NZ,nn	
F48405	CALL	P,nn	
EC8405	CALL	PE,nn	
E48405	CALL	PO,nn	
CC8405	CALL	Z,nn	
CD8405	CALL	nn	Unconditional Call to Subroutine at nn
3F	CCF		Complement Carry Flag
BE	CP	(HL)	Compare Operand with Acc.
DDBE05	CP	(IX+d)	
FDBE05	CP	(IY+d)	
BF	CP	A	
B8	CP	B	
B9	CP	C	
BA	CP	D	
BB	CP	E	
BC	CP	H	
BD	CP	L	
FE20	CP	n	
EDA9	CPD		Compare Location (HL) and Acc. Decrement HL and BC
EDB9	CPDR		Compare Location (HL) and Acc. Decrement HL and BC, Repeat until BC = 0
EDA1	CPI		Compare Location (HL) and Acc., Increment HL and Decrement BC
EDB1	CPIR		Compare Location (HL) and Acc. Increment HL, Decrement BC, Repeat until BC = 0
2F	CPL		Complement Acc. (1's Comp)
27	DAA		Decimal Adjust Acc.
35	DEC	(HL)	Decrement Operand
DD3505	DEC	(IX+d)	
FD3505	DEC	(IY+d)	
3D	DEC	A	
05	DEC	B	
0B	DEC	BC	
0D	DEC	C	
15	DEC	D	
1B	DEC	DE	

OBJ CODE	SOURCE STATEMENT		OPERATION
1D	DEC	E	Decrement Operand
25	DEC	H	
2B	DEC	HL	
DD2B	DEC	IX	
FD2B	DEC	IY	
2D	DEC	L	
3B	DEC	SP	
F3	DI		Disable Interrupts
102E	DJNZ	e	Decrement B and Jump Relative if B = 0
FB	EI		Enable Interrupts
E3	EX	(SP),HL	Exchange Location and (SP)
DDE3	EX	(SP),IX	
FDE3	EX	(SP),IY	
08	EX	AF,AF'	Exchange the Contents of AF and AF'
EB	EX	DE,HL	Exchange the Contents of DE and HL
D9	EXX		Exchange the Contents of BC,DE,HL with Contents of BC',DE',HL' Respectively
76	HALT		HALT (Wait for Interrupt or Reset)
ED46	IM	0	Set Interrupt Mode
ED56	IM	1	
ED5E	IM	2	
ED78	IN	A,(C)	Load Reg. with Input from Device (C)
ED40	IN	B,(C)	
ED48	IN	C,(C)	
ED50	IN	D,(C)	
ED58	IN	E,(C)	
ED60	IN	H,(C)	
ED68	IN	L,(C)	
34	INC	(HL)	Increment Operand
DD3405	INC	**(IX+d)**	
FD3405	INC	**(IY+d)**	
3C	INC	A	
04	INC	B	
03	INC	BC	
0C	INC	C	
14	INC	D	
13	INC	DE	
1C	INC	E	
24	INC	H	
23	INC	HL	
DD23	INC	IX	
FD23	INC	IY	
2C	INC	L	
33	INC	SP	
DB20	IN	A,(n)	Load Acc. with Input from Device n
EDAA	IND		Load Location (HL) with Input from Port (C), Decrement HL and B

OBJ CODE	SOURCE STATEMENT		OPERATION
EDBA	INDR		Load Location (HL) with Input from Port (C), Decrement HL and Decrement B, Repeat until B = 0
EDA2	INI		Load Location (HL) with Input from Port (C); Increment HL and Decrement B
EDB2	INIR		Load Location (HL) with Input from Port (C), Increment HL and Decrement B, Repeat until B = 0
C38405	JP	nn	Unconditional Jump to Location
E9	JP	(HL)	
DDE9	JP	(IX)	
FDE9	JP	(IY)	
DA8405	JP	C,nn	Jump to Location if Condition True
FA8405	JP	M,nn	
D28405	JP	NC,nn	
C28405	JP	NZ,nn	
F28405	JP	P,nn	
EA8405	JP	PE,nn	
E28405	JP	PO,nn	
CA8405	JP	Z,nn	
382E	JR	C,e	Jump Relative to PC+e if Condition True
302E	JR	NC,e	
202E	JR	NZ,e	
282E	JR	Z,e	
182E	JR	e	Unconditional Jump Relative to PC+e
02	LD	(BC),A	Load Source to Destination
12	LD	(DE),A	
77	LD	(HL),A	
70	LD	(HL),B	
71	LD	(HL),C	
72	LD	(HL),D	
73	LD	(HL),E	
74	LD	(HL),H	
75	LD	(HL),L	
3620	LD	(HL),n	
DD7705	LD	(IX+d),A	
DD7005	LD	(IX+d),B	
DD7105	LD	(IX+d),C	
DD7205	LD	(IX+d),D	
DD7305	LD	(IX+d),E	
DD7405	LD	(IX+d),H	
DD7505	LD	(IX+d),L	
DD360520	LD	(IX+d),n	
FD7705	LD	(IY+d),A	
FD7005	LD	(IY+d),B	
FD7105	LD	(IY+d),C	
FD7205	LD	(IY+d),D	
FD7305	LD	(IY+d),E	
FD7405	LD	(IY+d),H	
FD7505	LD	(IY+d),L	
FD360520	LD	(IY+d),n	
328405	LD	(nn),A	
ED438405	LD	(nn),BC	

OBJ CODE	SOURCE STATEMENT		OPERATION
ED538405	LD	(nn),DE	Load Source to Destination
228405	LD	(nn),HL	
DD228405	LD	(nn),IX	
FD228405	LD	(nn),IY	
ED738405	LD	(nn),SP	
0A	LD	A,(BC)	
1A	LD	A,(DE)	
7E	LD	A,(HL)	
DD7E05	LD	A,(IX+d)	
FD7E05	LD	A,(IY+d)	
3A8405	LD	A,(nn)	
7F	LD	A,A	
78	LD	A,B	
79	LD	A,C	
7A	LD	A,D	
7B	LD	A,E	
7C	LD	A,H	
ED57	LD	A,I	
7D	LD	A,L	
3E20	LD	A,n	
ED5F	LD	A,R	
46	LD	B,(HL)	
DD4605	LD	B,(IX+d)	
FD4605	LD	B,(IY+d)	
47	LD	B,A	
40	LD	B,B	
41	LD	B,C	
42	LD	B,D	
43	LD	B,E	
44	LD	B,H	
45	LD	B,L	
0620	LD	B,n	
ED4B8405	LD	BC,(nn)	
018405	LD	BC,nn	
4E	LD	C,(HL)	
DD4E05	LD	C,(IX+d)	
FD4E05	LD	C,(IY+d)	
4F	LD	C,A	
48	LD	C,B	
49	LD	C,C	
4A	LD	C,D	
4B	LD	C,E	
4C	LD	C,H	
4D	LD	C,L	
0E20	LD	C,n	
56	LD	D,(HL)	
DD5605	LD	D,(IX+d)	
FD5605	LD	D,(IY+d)	
57	LD	D,A	
50	LD	D,B	
51	LD	D,C	
52	LD	D,D	
53	LD	D,E	
54	LD	D,H	
55	LD	D,L	
1620	LD	D,n	
ED5B8405	LD	DE,(nn)	
118405	LD	DE,nn	
5E	LD	E,(HL)	
DD5E05	LD	E,(IX+d)	
FD5E05	LD	E,(IY+d)	
5F	LD	E,A	
58	LD	E,B	
59	LD	E,C	

OBJ CODE	SOURCE STATEMENT		OPERATION
5A	LD	E,D	Load Source to Destination
5B	LD	E,E	
5C	LD	E,H	
5D	LD	E,L	
1E20	LD	**E,n**	
66	LD	H,(HL)	
DD6605	LD	**H,(IX+d)**	
FD6605	LD	**H,(IY+d)**	
67	LD	H,A	
60	LD	H,B	
61	LD	H,C	
62	LD	H,D	
63	LD	H,E	
64	LD	H,H	
65	LD	H,L	
2620	LD	**H,n**	
2A8405	LD	**HL,(nn)**	
218405	LD	**HL,nn**	
ED47	LD	I,A	
DD2A8405	LD	**IX,(nn)**	
DD218405	LD	**IX,nn**	
FD2A8405	LD	**IY,(nn)**	
FD218405	LD	**IY,nn**	
6E	LD	L,(HL)	
DD6E05	LD	**L,(IX+d)**	
FD6E05	LD	**L,(IY+d)**	
6F	LD	L,A	
68	LD	L,B	
69	LD	L,C	
6A	LD	L,D	
6B	LD	L,E	
6C	LD	L,H	
6D	LD	L,L	
2E20	LD	**L,n**	
ED4F	**LD**	**R,A**	
ED7B8405	LD	**SP,(nn)**	
F9	LD	SP,HL	
DDF9	LD	SP,IX	
FDF9	LD	SP,IY	
318405	LD	**SP,nn**	
EDA8	LDD		Load Location (DE) with Location (HL), Decrement DE,HL and BC
EDB8	LDDR		Load Location (DE) with Location (HL), Repeat until BC = 0
EDA0	LDI		Load Location (DE) with Location (HL), Increment DE,HL, Decrement BC
EDB0	LDIR		Load Location (DE) with Location (HL), Increment DE,HL, Decrement BC and Repeat until BC = 0
ED44	NEG		Negate Acc. (2's Complement)
00	NOP		No Operation
B6	OR	(HL)	Logical "OR" of Operand and Acc.
DDB605	OR	**(IX+d)**	

OBJ CODE	SOURCE STATEMENT		OPERATION
FDB605	OR	(IY+d)	Logical "OR" of Operand and Acc.
B7	OR	A	
B0	OR	B	
B1	OR	C	
B2	OR	D	
B3	OR	E	
B4	OR	H	
B5	OR	L	
F620	OR	n	
ED8B	OTDR		Load Output Port (C) with Location (HL) Decrement HL and B, Repeat until B = 0
EDB3	OTIR		Load Output Port (C) with Location (HL), Increment HL, Decrement B, Repeat until B = 0
ED79	OUT	(C),A	Load Output Port (C) with Reg.
ED41	OUT	(C),B	
ED49	OUT	(C),C	
ED51	OUT	(C),D	
ED59	OUT	(C),E	
ED61	OUT	(C),H	
ED69	OUT	(C),L	
D320	OUT	(n),A	Load Output Port (n) with Acc.
EDAB	OUTD		Load Output Port (C) with Location (HL), Decrement HL and B
EDA3	OUTI		Load Output Port (C) with Location (HL), Increment HL and Decrement B
F1	POP	AF	Load Destination with Top of Stack
C1	POP	BC	
D1	POP	DE	
E1	POP	HL	
DDE1	POP	IX	
FDE1	POP	IY	
F5	PUSH	AF	Load Source to Stack
C5	PUSH	BC	
D5	PUSH	DE	
E5	PUSH	HL	
DDE5	PUSH	IX	
FDE5	PUSH	IY	
CB86	RES	0,(HL)	Reset Bit b of Operand
DDCB0586	RES	0,(IX+d)	
FDCB0586	RES	0,(IY+d)	
CB87	RES	0,A	
CB80	RES	0,B	
CB81	RES	0,C	
CB82	RES	0,D	
CB83	RES	0,E	
CB84	RES	0,H	
CB85	RES	0,L	
CB8E	RES	1,(HL)	
DDCB058E	RES	1,(IX+d)	
FDCB058E	RES	1,(IY+d)	
CB8F	RES	1,A	

OBJ CODE	SOURCE STATEMENT		OPERATION
CB88	RES	1,B	Reset Bit b of
CB89	RES	1,C	Operand
CB8A	RES	1,D	
CB8B	RES	1,E	
CB8C	RES	1,H	
CB8D	RES	1,L	
CB96	RES	2,(HL)	
DDCB0596	RES	2,(IX+d)	
FDCB0596	RES	2,(IY+d)	
CB97	RES	2,A	
CB90	RES	2,B	
CB91	RES	2,C	
CB92	RES	2,D	
CB93	RES	2,E	
CB94	RES	2,H	
CB95	RES	2,L	
CB9E	RES	3,(HL)	
DDCB059E	RES	3,(IX+d)	
FDCB059E	RES	3,(IY+d)	
CB9F	RES	3,A	
CB98	RES	3,B	
CB99	RES	3,C	
CB9A	RES	3,D	
CB9B	RES	3,E	
CB9C	RES	3,H	
CB9D	RES	3,L	
CBA6	RES	4,(HL)	
DDCB05A6	RES	4,(IX+d)	
FDCB05A6	RES	4,(IY+d)	
CBA7	RES	4,A	
CBA0	RES	4,B	
CBA1	RES	4,C	
CBA2	RES	4,D	
DBA3	RES	4,E	
CBA4	RES	4,H	
CBA5	RES	4,L	
CBAE	RES	5,(HL)	
DDCB05AE	RES	5,(IX+d)	
FDCB05AE	RES	5,(IY+d)	
CBAF	RES	5,A	
CBA8	RES	5,B	
CBA9	RES	5,C	
CBAA	RES	5,D	
CBAB	RES	5,E	
CBAC	RES	5,H	
CBAD	RES	5,L	
CBB6	RES	6,(HL)	
DDCB05B6	RES	6,(IX+d)	
FDCB05B6	RES	6,(IY+d)	
CBB7	RES	6,A	
CBB0	RES	6,B	
CBB1	RES	6,C	
CBB2	RES	6,D	
CBB3	RES	6,E	
CBB4	RES	6,H	
CBB5	RES	6,L	
CBBE	RES	7,(HL)	
DDCB05BE	RES	7,(IX+d)	
FDCB05BE	RES	7,(IY+d)	
CBBF	RES	7,A	
CBB8	RES	7,B	
CBB9	RES	7,C	
CBBA	RES	7,D	

OBJ CODE	SOURCE STATEMENT		OPERATION
CBBB	RES	7,E	Reset Bit b of Operand
CBBC	RES	7,H	
CBBD	RES	7,L	
C9	RET		Return from Subroutine
D8	RET	C	Return from Subroutine if Condition True
F8	RET	M	
D0	RET	NC	
C0	RET	NZ	
F0	RET	P	
E8	RET	PE	
E0	RET	P0	
C8	RET	Z	
ED4D	RETI		Return from Interrupt
ED45	RETN		Return from Non-Maskable Interrupt
CB16	RL	(HL)	Rotate Left Through Carry
DDCB0516	RL	**(IX+d)**	
FDCB0516	RL	**(IY+d)**	
CB17	RL	A	
CB10	RL	B	
CB11	RL	C	
CB12	RL	D	
CB13	RL	E	
CB14	RL	H	
CB15	RL	L	
17	RLA		Rotate Left Acc. Through Carry
CB06	RLC	(HL)	Rotate Left Circular
DDCB0506	RLC	**(IX+d)**	
FDCB0506	RLC	**(IY+d)**	
CB07	RLC	A	
CB00	RLC	B	
CB01	RLC	C	
CB02	RLC	D	
CB03	RLC	E	
CB04	RLC	H	
CB05	RLC	L	
07	RLCA		Rotate Left Circular Acc.
ED6F	RLD		Rotate Digit Left and Right between Acc. and and Location (HL)
CB1E	RR	(HL)	Rotate Right Through Carry
DDCB051E	RR	**(IX+d)**	
FDCB051E	RR	**(IY+d)**	
CB1F	RR	A	
CB18	RR	B	
CB19	RR	C	
CB1A	RR	D	
CB1B	RR	E	
CB1C	RR	H	
CB1D	RR	L	
1F	RRA		Rotate Right Acc. Through Carry
CB0E	RRC	(HL)	Rotate Right Circular
DDCB050E	RRC	**(IX+d)**	
FDCB050E	RRC	**(IY+d)**	
CB0F	RRC	A	

OBJ CODE	SOURCE STATEMENT		OPERATION
CB08	RRC	B	Rotate Right Circular
CB09	RRC	C	
CB0A	RRC	D	
CB0B	RRC	E	
CB0C	RRC	H	
CB0D	RRC	L	
0F	RRCA		Rotate Right Circular Acc.
ED67	RRD		Rotate Digit Right and Left Between Acc. and Location (HL)
C7	RST	00H	Restart to Location
CF	RST	08H	
D7	RST	10H	
DF	RST	18H	
E7	RST	20H	
EF	RST	28H	
F7	RST	30H	
FF	RST	38H	
DE20	SBC	**A,n**	Subtract Operand
9E	SBC	A,(HL)	from Acc. with Carry
DD9E05	SBC	**A,(IX+d)**	
FD9E05	SBC	**A,(IY+d)**	
9F	SBC	A,A	
98	SBC	A,B	
99	SBC	A,C	
9A	SBC	A,D	
9B	SBC	A,E	
9C	SBC	A,H	
9D	SBC	A,L	
ED42	SBC	HL,BC	
ED52	SBC	HL,DE	
ED62	SBC	HL,HL	
ED72	SBC	HL,SP	
37	SCF		Set Carry Flag (C = 1)
CBC6	SET	0,(HL)	Set Bit b of Location
DDCB05C6	SET	**0,(IX+d)**	
FDCB05C6	SET	**0,(IY+d)**	
CBC7	SET	0,A	
CBC0	SET	0,B	
CBC1	SET	0,C	
CBC2	SET	0,D	
CBC3	SET	0,E	
CBC4	SET	0,H	
CBC5	SET	0,L	
CBCE	SET	1,(HL)	
DDCB05CE	SET	**1,(IX+d)**	
FDCB05CE	SET	**1,(IY+d)**	
CBCF	SET	1,A	
CBC8	SET	1,B	
CBC9	SET	1,C	
CBCA	SET	1,D	
CBCB	SET	1,E	
CBCC	SET	1,H	
CBCD	SET	1,L	
CBD6	SET	2,(HL)	
DDCB05D6	SET	**2,(IX+d)**	
FDCB05D6	SET	**2,(IY+d)**	
CBD7	SET	2,A	
CBD0	SET	2,B	
CBD1	SET	2,C	
CBD2	SET	2,D	

OBJ CODE	SOURCE STATEMENT		OPERATION
CBD3	SET	2,E	Set Bit b of Location
CBD4	SET	2,H	
CBD5	SET	2,L	
CBD8	SET	3,B	
CBDE	SET	3,(HL)	
DDCB05DE	SET	3,(IX+d)	
FDCB05DE	SET	3,(IY+d)	
CBDF	SET	3,A	
CBD9	SET	3,C	
CBDA	SET	3,D	
CBDB	SET	3,E	
CBDC	SET	3,H	
CBDD	SET	3,L	
CBE6	SET	4,(HL)	
DDCB05E6	SET	4,(IX+d)	
FDCB05E6	SET	4,(IY+d)	
CBE7	SET	4,A	
CBE0	SET	4,B	
CBE1	SET	4,C	
CBE2	SET	4,D	
CBE3	SET	4,E	
CBE4	SET	4,H	
CBE5	SET	4,L	
CBEE	SET	5,(HL)	
DDCB05EE	SET	5,(IX+d)	
FDCB05EE	SET	5,(IY+d)	
CBEF	SET	5,A	
CBE8	SET	5,B	
CBE9	SET	5,C	
CBEA	SET	5,D	
CBEB	SET	5,E	
CBEC	SET	5,H	
CBED	SET	5,L	
CBF6	SET	6,(HL)	
DDCB05F6	SET	6,(IX+d)	
FDCB05F6	SET	6,(IY+d)	
CBF7	SET	6,A	
CBF0	SET	6,B	
CBF1	SET	6,C	
CBF2	SET	6,D	
CBF3	SET	6,E	
CBF4	SET	6,H	
CBF5	SET	6,L	
CBFE	SET	7,(HL)	
DDCB05FE	SET	7,(IX+d)	
FDCB05FE	SET	7,(IY+d)	
CBFF	SET	7,A	
CBF8	SET	7,B	
CBF9	SET	7,C	
CBFA	SET	7,D	
CBFB	SET	7,E	
CBFC	SET	7,H	
CBFD	SET	7,L	
CB26	SLA	(HL)	Shift Operand Left
DDCB0526	SLA	(IX+d)	Arithmetic
FDCB0526	SLA	(IY+d)	
CB27	SLA	A	
CB20	SLA	B	
CB21	SLA	C	
CB22	SLA	D	
CB23	SLA	E	
CB24	SLA	H	
CB25	SLA	L	

OBJ CODE	SOURCE STATEMENT		OPERATION
CB2E	SRA	(HL)	Shift Operand Right
DDCB052E	SRA	(IX+d)	Arithmetic
FDCB052E	SRA	(IY+d)	
CB2F	SRA	A	
CB28	SRA	B	
CB29	SRA	C	
CB2A	SRA	D	
CB2B	SRA	E	
CB2C	SRA	H	
CB2D	SRA	L	
CB3E	SRL	(HL)	Shift Operand Right
DDCB053E	SRL	(IX+d)	Logical
FDCB053E	SRL	(IY+d)	
CB3F	SRL	A	
CB38	SRL	B	
CB39	SRL	C	
CB3A	SRL	D	
CB3B	SRL	E	
CB3C	SRL	H	
CB3D	SRL	L	
96	SUB	(HL)	Subtract Operand
DD9605	SUB	(IX+d)	from Acc.
FD9605	SUB	(IY+d)	
97	SUB	A	
90	SUB	B	
91	SUB	C	
92	SUB	D	
93	SUB	E	
94	SUB	H	
95	SUB	L	
D620	SUB	n	
AE	XOR	(HL)	Exclusive "OR"
DDAE05	XOR	(IX+d)	Operand and Acc.
FDAE05	XOR	(IY+d)	
AF	XOR	A	
A8	XOR	B	
A9	XOR	C	
AA	XOR	D	
AB	XOR	E	
AC	XOR	H	
AD	XOR	L	
EE20	XOR	n	

Example Values

```
nn      EQU     584H
d       EQU     5
n       EQU     20H
e               30H
```

RIO

RIO I/O REQUEST VECTOR							
LOGICAL UNIT	REQUEST	DATA TRANSFER ADDRESS	DATA LENGTH	COMPLETION RETURN ADDRESS	ERROR RETURN ADDRESS	COMPLETION CODE	SUPPLEMENTAL PARAMETER VECTOR ADDRESS
0	1	2-3	4-5	6-7	8-9	A	B-C

SUPPLEMENTAL PARAMETER VECTOR			
TYPE OPEN OR ASSIGN	DRIVE SPECIFIER	LENGTH OF NAME	NAME
0	1	2	3 . . . .

I/O REQUEST CODE (RETURN WHEN COMPLETE)		I/O REQUEST CODE (RETURN WHEN COMPLETE)	
0	INITIALIZE	1A	ERASE FILE
2	ASSIGN	1C	READ AND DELETE
4	OPEN	1E	READ CURRENT RECORD
6	CLOSE	20	READ PREVIOUS RECORD
8	REWIND	22	READ DIRECT
A	READ BINARY	24	SKIP FORWARD
C	READ ASCII	26	SKIP BACKWARD
E	WRITE BINARY	28	SKIP TO END
10	WRITE ASCII	2A	RENAME
12	WRITE CURRENT RECORD	2C	UPDATE
14	WRITE DIRECT	2E	SET ATTRIBUTES
16	DELETE	30	QUERY ATTRIBUTES
18	DELETE REMAINING		

COMPLETION CODE	MEANING	COMPLETION CODE	MEANING
40	INVALID DRIVE NAME	C6	DATA TRANSFER ERROR
41	INVALID OR INACTIVE DEVICE	C7	FILE NOT FOUND
42	INVALID UNIT	C9	END OF FILE ERROR
43	MEMORY PROTECT VIOLATION	CA	POINTER CHECK ERROR
44	MISSING OR INVALID OPERAND(S)	CB	FILE NOT OPEN
45	SYSTEM ERROR	CC	UNIT ALREADY ACTIVE (OPEN)
46	ILLEGAL FILE NAME	CD	ASSIGN BUFFER FULL
47	NON-EXISTENT COMMAND	CE	INVALID DRIVE SPECIFICATION
48	ILLEGAL FILE TYPE	CF	LOGICAL UNIT TABLE FULL (>16 OPEN)
49	PROGRAM ABORT		
4A	INSUFFICIENT MEMORY		
4B	MISSING OR INVALID FILE PROPERTIES	D0	DUPLICATE FILE
		D1	DISKETTE ID ERROR
80	OPERATION COMPLETE	D2	INVALID ATTRIBUTES
81	DIRECTORY FORMAT ERROR	D3	DISK IS FULL
82	SCRATCH FILE CREATED	D4	FILE NOT IN PROPER DIRECTORY RECORD
83	FILE NAME TRUNCATED		
84	ATTRIBUTE LIST TRUNCATED	D5	BEGINNING OF FILE ERROR
		D6	FILE ALREADY OPEN ON OTHER UNIT
C1	INVALID OPERATION (REQUEST)	D7	INVALID RENAME TO SCRATCH FILE
C2	DEVICE IS NOT READY	D8	FILE LOCKED (ATTEMPT TO CHANGE ATTRIBUTES)
C3	WRITE PROTECTION		
C4	SECTOR ADDRESS ERROR	D9	INVALID OPEN REQUEST
C5	SEEK ERROR		

ZDOS RETURN CODES

DATA LOST	DISK ERROR	
C 1	8 1	INVALID OPERATION
C 2	8 2	DUPLICATE FILE
C 3	8 3	ACTIVE FILE TABLE FULL
C 4	8 4	FILE NOT FOUND
C 5	8 5	DIRECTORY FULL
C 6	8 6	SYSTEM ERROR
C 8	8 8	FILE NOT OPEN
C 9	8 9	END OF FILE
C A	8 A	DISK ERROR
C B	8 B	DISK FULL
C C	8 C	POINTER ERROR
C D	8 D	BEGINNING OF FILE
C E	8 E	FILE ALREADY OPEN
C F	8 F	DISK NOT READY
D 0	9 0	WRONG DISK
D 1	9 1	NONEXISTENT DISK

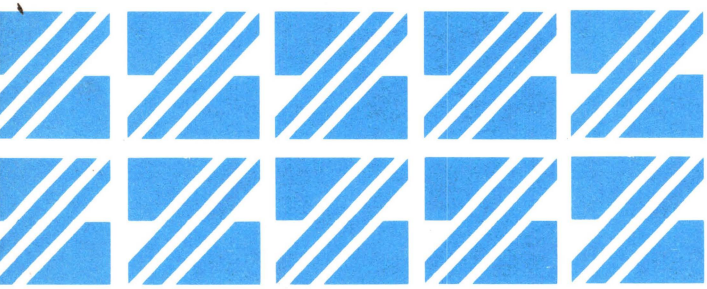
ZDOS REQUEST VECTOR

REQUEST	03	BUFFER ADDR	NOT USED	# RECORDS	fn	ft	UNIT	ERROR CODE
0	1	2-3	4-5	6	7-12	13	14	15

REQUEST CODE	OPERATION
08	INITIALIZE
0C	OPEN
0E	CREATE
10	CLOSE
12	ERASE
14	RENAME
16	REWIND
18	READ n RECORDS
1A	READ CURRENT RECORD
1C	READ PREVIOUS RECORD
1E	SKIP n RECORDS
20	BACK n RECORDS
22	REWRITE CURRENT RECORD
24	INSERT n RECORDS
26	DELETE n RECORDS

Zilog

10460 Bubb Road
Cupertino, California 95014
(408) 446-4666
TWX 910-338-7621



03-0012-01 PRINTED IN USA