

8080 Mnemonic	Z80 Mnemonic	Code	Operation	Forth Assembler
ADC M	ADC A,(HL)	8E	A <- A + (HL) + Carry	M ADC
---	ADC A,(IX+index)	DD8Eindex	A <- A + (IX+index) + Carry	index +IX ADC
---	ADC A,(IY+index)	FD8Eindex	A <- A + (IY+index) + Carry	index +IY ADC
ADC A	ADC A,A	8F	A <- A + A + Carry	A ADC
ADC B	ADC A,B	88	A <- A + B + Carry	B ADC
ACI byte	ADC A,byte	CEbyte	A <- A + byte + Carry	byte ACI
ADC C	ADC A,C	89	A <- A + C + Carry	C ADC
ADC D	ADC A,D	8A	A <- A + D + Carry	D ADC
ADC E	ADC A,E	8B	A <- A + E + Carry	E ADC
ADC H	ADC A,H	8C	A <- A + H + Carry	H ADC
ADC L	ADC A,L	8D	A <- A + L + Carry	L ADC
---	ADC HL,BC	ED4A	HL <- HL + BC + Carry	BC DADC
---	ADC HL,DE	ED5A	HL <- HL + DE + Carry	DE DADC
---	ADC HL,HL	ED6A	HL <- HL + HL + Carry	HL DADC
---	ADC HL,SP	ED7A	HL <- HL + SP + Carry	SP DADC
ADD M	ADD A,(HL)	86	A <- A + (HL)	M ADD
---	ADD A,(IX+index)	DD86index	A <- A + (IX+index)	index +IX ADD
---	ADD A,(IY+index)	FD86index	A <- A + (IY+index)	index +IY ADD
ADD A	ADD A,A	87	A <- A + A	A ADD
ADD B	ADD A,B	80	A <- A + B	B ADD
ADI byte	ADD A,byte	C6byte	A <- A + byte	byte ADI
ADD C	ADD A,C	81	A <- A + C	C ADD
ADD D	ADD A,D	82	A <- A + D	D ADD
ADD E	ADD A,E	83	A <- A + E	E ADD
ADD H	ADD A,H	84	A <- A + H	H ADD
ADD L	ADD A,L	85	A <- A + L	L ADD
DAD B	ADD HL,BC	9	HL <- HL + BC	BC DAD
DAD D	ADD HL,DE	19	HL <- HL + DE	DE DAD
DAD H	ADD HL,HL	29	HL <- HL + HL	HL DAD
DAD SP	ADD HL,SP	39	HL <- HL + SP	SP DAD
---	ADD IX,BC	DD09	IX <- IX + BC	BC IX DAD
---	ADD IX,DE	DD19	IX <- IX + DE	DE IX DAD
---	ADD IX,IX	DD29	IX <- IX + IX	IX IX DAD
---	ADD IX,SP	DD39	IX <- IX + SP	SP IX DAD
---	ADD IY,BC	FD09	IY <- IY + BC	BC IY DAD
---	ADD IY,DE	FD19	IY <- IY + DE	DE IY DAD
---	ADD IY,IY	FD29	IY <- IY + IY	IY IY DAD
---	ADD IY,SP	FD39	IY <- IY + SP	SP IY DAD
ANA A	AND A	A7	A <- A AND A	A AND
ANA B	AND B	A0	A <- A AND B	B AND
ANA C	AND C	A1	A <- A AND C	C AND
ANA D	AND D	A2	A <- A AND D	D AND
ANA E	AND E	A3	A <- A AND E	E AND
ANA H	AND H	A4	A <- A AND H	H AND
ANA L	AND L	A5	A <- A AND L	L AND
ANA M	AND (HL)	A6	A <- A AND (HL)	M AND
---	AND (IX+index)	DDA6index	A <- A AND (IX+index)	index +IX AND
---	AND (IY+index)	FDA6index	A <- A AND (IY+index)	index +IY AND

8080 Mnemonic	Z80 Mnemonic	Code	Operation	Forth Assembler
ANI byte	AND byte	E6byte	A <- A AND byte	byte ANI
---	BIT 0,A	CB47	Z flag <- NOT 0b	A 0 BIT
---	BIT 0,B	CB40	Z flag <- NOT 0b	B 0 BIT
---	BIT 0,C	CB41	Z flag <- NOT 0b	C 0 BIT
---	BIT 0,D	CB42	Z flag <- NOT 0b	D 0 BIT
---	BIT 0,E	CB43	Z flag <- NOT 0b	E 0 BIT
---	BIT 0,H	CB44	Z flag <- NOT 0b	H 0 BIT
---	BIT 0,L	CB45	Z flag <- NOT 0b	L 0 BIT
---	BIT 0,(HL)	CB46	Z flag <- NOT 0b	M 0 BIT
---	BIT 0,(IX+index)	DDCIndex46	Z flag <- NOT 0b	index +IX 0 BIT
---	BIT 0,(IY+index)	FDCIndex46	Z flag <- NOT 0b	index +IY 0 BIT
---	BIT 1,A	CB4F	Z flag <- NOT 1b	A 1 BIT
---	BIT 1,B	CB48	Z flag <- NOT 1b	B 1 BIT
---	BIT 1,C	CB49	Z flag <- NOT 1b	C 1 BIT
---	BIT 1,D	CB4A	Z flag <- NOT 1b	D 1 BIT
---	BIT 1,E	CB4B	Z flag <- NOT 1b	E 1 BIT
---	BIT 1,H	CB4C	Z flag <- NOT 1b	H 1 BIT
---	BIT 1,L	CB4D	Z flag <- NOT 1b	L 1 BIT
---	BIT 1,(HL)	CB4E	Z flag <- NOT 1b	M 1 BIT
---	BIT 1,(IX+index)	DDCIndex4E	Z flag <- NOT 1b	index +IX 1 BIT
---	BIT 1,(IY+index)	FDCIndex4E	Z flag <- NOT 1b	index +IY 1 BIT
---	BIT 2,A	CB57	Z flag <- NOT 2b	A 2 BIT
---	BIT 2,B	CB50	Z flag <- NOT 2b	B 2 BIT
---	BIT 2,C	CB51	Z flag <- NOT 2b	C 2 BIT
---	BIT 2,D	CB52	Z flag <- NOT 2b	D 2 BIT
---	BIT 2,E	CB53	Z flag <- NOT 2b	E 2 BIT
---	BIT 2,H	CB54	Z flag <- NOT 2b	H 2 BIT
---	BIT 2,L	CB55	Z flag <- NOT 2b	L 2 BIT
---	BIT 2,(HL)	CB56	Z flag <- NOT 2b	M 2 BIT
---	BIT 2,(IX+index)	DDCIndex56	Z flag <- NOT 2b	index +IX 2 BIT
---	BIT 2,(IY+index)	FDCIndex56	Z flag <- NOT 2b	index +IY 2 BIT
---	BIT 3,A	CB5F	Z flag <- NOT 3b	A 3 BIT
---	BIT 3,B	CB58	Z flag <- NOT 3b	B 3 BIT
---	BIT 3,C	CB59	Z flag <- NOT 3b	C 3 BIT
---	BIT 3,D	CB5A	Z flag <- NOT 3b	D 3 BIT
---	BIT 3,E	CB5B	Z flag <- NOT 3b	E 3 BIT
---	BIT 3,H	CB5C	Z flag <- NOT 3b	H 3 BIT
---	BIT 3,L	CB5D	Z flag <- NOT 3b	L 3 BIT
---	BIT 3,(HL)	CB5E	Z flag <- NOT 3b	M 3 BIT
---	BIT 3,(IX+index)	DDCIndex5E	Z flag <- NOT 3b	index +IX 3 BIT
---	BIT 3,(IY+index)	FDCIndex5E	Z flag <- NOT 3b	index +IY 3 BIT
---	BIT 4,A	CB67	Z flag <- NOT 4b	A 4 BIT
---	BIT 4,B	CB60	Z flag <- NOT 4b	B 4 BIT
---	BIT 4,C	CB61	Z flag <- NOT 4b	C 4 BIT
---	BIT 4,D	CB62	Z flag <- NOT 4b	D 4 BIT
---	BIT 4,E	CB63	Z flag <- NOT 4b	E 4 BIT
---	BIT 4,H	CB64	Z flag <- NOT 4b	H 4 BIT
---	BIT 4,L	CB65	Z flag <- NOT 4b	L 4 BIT
---	BIT 4,(HL)	CB66	Z flag <- NOT 4b	M 4 BIT

8080 Mnemonic	Z80 Mnemonic	Code	Operation	Forth Assembler
---	BIT 4,(IX+index)	DDCBindex66	Z flag <- NOT 4b	index +IX 4 BIT
---	BIT 4,(IY+index)	FDCBindex66	Z flag <- NOT 4b	index +IY 4 BIT
---	BIT 5,A	CB6F	Z flag <- NOT 5b	A 5 BIT
---	BIT 5,B	CB68	Z flag <- NOT 5b	B 5 BIT
---	BIT 5,C	CB69	Z flag <- NOT 5b	C 5 BIT
---	BIT 5,D	CB6A	Z flag <- NOT 5b	D 5 BIT
---	BIT 5,E	CB6B	Z flag <- NOT 5b	E 5 BIT
---	BIT 5,H	CB6C	Z flag <- NOT 5b	H 5 BIT
---	BIT 5,L	CB6D	Z flag <- NOT 5b	L 5 BIT
---	BIT 5,(HL)	CB6E	Z flag <- NOT 5b	M 5 BIT
---	BIT 5,(IX+index)	DDCBindex6E	Z flag <- NOT 5b	index +IX 5 BIT
---	BIT 5,(IY+index)	FDCBindex6E	Z flag <- NOT 5b	index +IY 5 BIT
---	BIT 6,A	CB77	Z flag <- NOT 6b	A 6 BIT
---	BIT 6,B	CB70	Z flag <- NOT 6b	B 6 BIT
---	BIT 6,C	CB71	Z flag <- NOT 6b	C 6 BIT
---	BIT 6,D	CB72	Z flag <- NOT 6b	D 6 BIT
---	BIT 6,E	CB73	Z flag <- NOT 6b	E 6 BIT
---	BIT 6,H	CB74	Z flag <- NOT 6b	H 6 BIT
---	BIT 6,L	CB75	Z flag <- NOT 6b	L 6 BIT
---	BIT 6,(HL)	CB76	Z flag <- NOT 6b	M 6 BIT
---	BIT 6,(IX+index)	DDCBindex76	Z flag <- NOT 6b	index +IX 6 BIT
---	BIT 6,(IY+index)	FDCBindex76	Z flag <- NOT 6b	index +IY 6 BIT
---	BIT 7,A	CB7F	Z flag <- NOT 7b	A 7 BIT
---	BIT 7,B	CB78	Z flag <- NOT 7b	B 7 BIT
---	BIT 7,C	CB79	Z flag <- NOT 7b	C 7 BIT
---	BIT 7,D	CB7A	Z flag <- NOT 7b	D 7 BIT
---	BIT 7,E	CB7B	Z flag <- NOT 7b	E 7 BIT
---	BIT 7,H	CB7C	Z flag <- NOT 7b	H 7 BIT
---	BIT 7,L	CB7D	Z flag <- NOT 7b	L 7 BIT
---	BIT 7,(HL)	CB7E	Z flag <- NOT 7b	M 7 BIT
---	BIT 7,(IX+index)	DDCBindex7E	Z flag <- NOT 7b	index +IX 7 BIT
---	BIT 7,(IY+index)	FDCBindex7E	Z flag <- NOT 7b	index +IY 7 BIT
CALL address	CALL address	CDaddress	(SP-1) <- PCh;(SP-2) <- PCl	address CALL
CNZ address	CALL NZ,address	C4address	If NZ, CALL address	address CLNZ
CZ address	CALL Z,address	CCaddress	If Z, CALL address	address CLZ
CNC address	CALL NC,address	D4address	If NC, CALL address	address CLNC
CC address	CALL C,address	DCaddress	If C, CALL address	address CLC
CPO address	CALL PO,address	E4address	If PO, CALL address	address CLPO
CPE address	CALL PE,address	ECaddress	If PE, CALL address	address CLPE
CP address	CALL P,address	F4address	If P, CALL address	address CLP
CM address	CALL M,address	FCaddress	If M, CALL address	address CLM
CMC	CCF	3F	CF (Carry Flag) <-NOT CF	CCF
CMP A	CP A	BF	A - A	A CP
CMP B	CP B	B8	A - B	B CP
CMP C	CP C	B9	A - C	C CP
CMP D	CP D	BA	A - D	D CP
CMP E	CP E	BB	A - E	E CP
CMP H	CP H	BC	A - H	H CP
CMP L	CP L	BD	A - L	L CP

8080 Mnemonic	Z80 Mnemonic	Code	Operation	Forth Assembler
CMP M	CP (HL)	BE	A - (HL)	M CP
---	CP (IX+index)	DDBEindex	A - (IX+index)	index +IX CP
---	CP (IY+index)	FDBEindex	A - (IY+index)	index +IY CP
CPI byte	CP byte	FEbyte	A - byte	byte CPBI
---	CPD	EDA9	A - (HL);HL <- HL-1;BC <- BC-1	CPD
---	CPDR	EDB9	A - (HL);HL <- HL-1;BC <- BC-1	CPDR
---	CPI	EDA1	A - (HL);HL <- HL+1;BC <- BC-1	CPI
---	CPIR	EDB1	A - (HL);HL <- HL+1;BC <- BC-1	CPIR
CMA	CPL	2F	A <- NOT A	CPL
DAA	DAA	27	---	DAA
DCR A	DEC A	3D	A <- A - 1	A DEC
DCR B	DEC B	5	B <- B - 1	B DEC
DCR C	DEC C	0D	C <- C - 1	C DEC
DCR D	DEC D	15	D <- D - 1	D DEC
DCR E	DEC E	1D	E <- E - 1	E DEC
DCR H	DEC H	25	H <- H - 1	H DEC
DCR L	DEC L	2D	L <- L - 1	L DEC
DCR M	DEC (HL)	35	(HL) <- (HL) - 1	M DEC
---	DEC (IX+index)	DD35index	(IX+index) <- (IX+index) - 1	index +IX DEC
---	DEC (IY+index)	FD35index	(IY+index) <- (IY+index) - 1	index +IY DEC
DCX B	DEC BC	0B	BC <- BC - 1	BC DCX
DCX D	DEC DE	1B	DE <- DE - 1	DE DCX
DCX H	DEC HL	2B	HL <- HL - 1	HL DCX
DCX SP	DEC SP	3B	SP <- SP - 1	SP DCX
---	DEC IX	DD2B	IX <- IX - 1	IX DCX
---	DEC IY	FD2B	IY <- IY - 1	IY DCX
DI	DI	F3	IFF <- 0	DI
---	DJNZ index	10index	B <- B - 1;	index DJNZ
EI	EI	FB	IFF <- 1	EI
XCHG	EX DE,HL	EB	HL <-> DE	XCHG
XTHL	EX (SP),HL	E3	H <-> (SP+1); L <-> (SP)	XTHL
---	EX (SP),IX	DDE3	IXh <-> (SP+1); IXl <-> (SP)	XTIX
---	EX (SP),IY	FDE3	IYh <-> (SP+1); IYl <-> (SP)	XTIY
---	EX AF,AF'	8	AF <-> AF'	EXAF
---	EXX	D9	BC/DE/HL <-> BC'/DE'/HL'	EXX
HLT	HLT	76	NOP;PC <- PC-1	HALT
---	IM 0	ED46	---	IM0
---	IM 1	ED56	---	IM1
---	IM 2	ED5E	---	IM2
IN byte	IN A,(byte)	DBbyte	A <- [byte]	byte IN
---	IN A,(C)	ED78	A <- [C]	A IN(C)
---	IN B,(C)	ED40	B <- [C]	B IN(C)
---	IN C,(C)	ED48	C <- [C]	C IN(C)
---	IN D,(C)	ED50	D <- [C]	D IN(C)
---	IN E,(C)	ED58	E <- [C]	E IN(C)
---	IN H,(C)	ED60	H <- [C]	H IN(C)

8080 Mnemonic	Z80 Mnemonic	Code	Operation	Forth Assembler
---	IN L,C)	ED68	L <- [C]	L IN(C)
INR A	INC A	3C	A <- A + 1	A INC
INR B	INC B	4	B <- B + 1	B INC
INR C	INC C	0C	C <- C + 1	C INC
INR D	INC D	14	D <- D + 1	D INC
INR E	INC E	1C	E <- E + 1	E INC
INR H	INC H	24	H <- H + 1	H INC
INR L	INC L	2C	L <- L + 1	L INC
INR M	INC (HL)	34	(HL) <- (HL) + 1	M INC
---	INC (IX+index)	DD34index	(IX+index) <- (IX+index) + 1	index +IX INC
---	INC (IY+index)	FD34index	(IY+index) <- (IY+index) + 1	index +IY INC
INX B	INC BC	3	BC <- BC + 1	BC INX
INX D	INC DE	13	DE <- DE + 1	DE INX
INX H	INC HL	23	HL <- HL + 1	HL INX
INX SP	INC SP	33	SP <- SP + 1	SP INX
---	INC IX	DD23	IX <- IX + 1	IX INX
---	INC IY	FD23	IY <- IY + 1	IY INX
---	IND	EDAA	(HL) <- [C];B <- B-1;HL <- HL-1	IND
---	INDR	EDBA	(HL) <- [C];B <- B-1;HL <- HL-1	INDR
---	INI	EDA2	(HL) <- [C];B <- B-1;HL <- HL+1	INI
---	INIR	EDB2	(HL) <- [C];B <- B-1;HL <- HL+1	INIR
JMP address	JP address	C3address	PC <- address	address JP
JNZ address	JP NZ,address	C2address	If NZ, PC <- address	address JNZ
JZ address	JP Z,address	CAaddress	If Z, PC <- address	address JZ
JNC address	JP NC,address	D2address	If NC, PC <- address	address JNC
JC address	JP C,address	DAaddress	If C, PC <- address	address JC
JPO address	JP PO,address	E2address	If PO, PC <- address	address JPO
JPE address	JP PE,address	EAaddress	If PE, PC <- address	address JPE
JP address	JP P,address	F2address	If P, PC <- address	address JPP
JM address	JP M,address	FAaddress	If M, PC <- address	address JPM
PCHL	JP (HL)	E9	PC <- HL	PCHL
---	JP (IX)	DDE9	PC <- IX	PCIX
---	JP (IY)	FDE9	PC <- IY	PCIY
---	JR index	18index	PC <- PC + index	index JR
---	JR NZ,index	20index	If NZ, PC <- PC + index	index JRNZ
---	JR Z,index	28index	If Z, PC <- PC + index	index JRZ
---	JR NC,index	30index	If NC, PC <- PC + index	index JRNC
---	JR C,index	38index	If C, PC <- PC + index	index JRC
STAX B	LD (BC),A	2	(BC) <- A	BC STAX
STAX D	LD (DE),A	12	(DE) <- A	DE STAX
MOV M,A	LD (HL),A	77	(HL) <- A	A M MOV
MOV M,B	LD (HL),B	70	(HL) <- B	B M MOV
MVI M,byte	LD (HL),byte	36byte	(HL) <- byte	byte M MVI
MOV M,C	LD (HL),C	71	(HL) <- C	C M MOV
MOV M,D	LD (HL),D	72	(HL) <- D	D M MOV
MOV M,E	LD (HL),E	73	(HL) <- E	E M MOV
MOV M,H	LD (HL),H	74	(HL) <- H	H M MOV
MOV M,L	LD (HL),L	75	(HL) <- L	L M MOV
---	LD (IX+index),A	DD77index	(IX+index) <- A	A index +IX MOV

8080 Mnemonic	Z80 Mnemonic	Code	Operation	Forth Assembler
---	LD (IX+index),B	DD70index	(IX+index) <- B	B index +IX MOV
---	LD (IX+index),byte	DD76indexbyte????	(IX+index) <- byte	byte index +IX MVI
---	LD (IX+index),byte	DD36index byte	(IX+index) <- byte	byte index +IX MVI
---	LD (IX+index),C	DD71index	(IX+index) <- C	C index +IX MOV
---	LD (IX+index),D	DD72index	(IX+index) <- D	D index +IX MOV
---	LD (IX+index),E	DD73index	(IX+index) <- E	E index +IX MOV
---	LD (IX+index),H	DD74index	(IX+index) <- H	H index +IX MOV
---	LD (IX+index),L	DD75index	(IX+index) <- L	L index +IX MOV
---	LD (IY+index),A	FD77index	(IY+index) <- A	A index +IY MOV
---	LD (IY+index),B	FD70index	(IY+index) <- B	B index +IY MOV
---	LD (IY+index),byte	FD76indexbyte????	(IY+index) <- byte	byte index +IY MVI
---	LD (IY+index),byte	FD36index byte	(IY+index) <- byte	byte index +IY MVI
---	LD (IY+index),C	FD71index	(IY+index) <- C	C index +IY MOV
---	LD (IY+index),D	FD72index	(IY+index) <- D	D index +IY MOV
---	LD (IY+index),E	FD73index	(IY+index) <- E	E index +IY MOV
---	LD (IY+index),H	FD74index	(IY+index) <- H	H index +IY MOV
---	LD (IY+index),L	FD75index	(IY+index) <- L	L index +IY MOV
STA word	LD (word),A	32word	(word) <- A	word STA
---	LD (word),BC	ED43word	(word) <- BC	BC word SXW
---	LD (word),DE	ED53word	(word) <- DE	DE word SXW
SHLD word	LD (word),HL	22word	(word) <- HL	word SHLD
---	LD (word),HL	ED63word	(word) <- HL	HL word SXW
---	LD (word),IX	DD22word	(word) <- IX	IX word SXW
---	LD (word),IY	FD22word	(word) <- IY	IY word SXW
---	LD (word),SP	ED73word	(word) <- SP	SP word SXW
LDAX B	LD A,(BC)	0A	A <- (BC)	BC LDAX
LDAX D	LD A,(DE)	1A	A <- (DE)	DE LDAX
MOV A,M	LD A,(HL)	7E	A <- (HL)	M A MOV
---	LD A,(IX+index)	DD7Eindex	A <- (IX+index)	index +IX A MOV
---	LD A,(IY+index)	FD7Eindex	A <- (IY+index)	index +IY A MOV
LDA word	LD A,(word)	3Aword	A <- (word)	word LDA
MOV A,A	LD A,A	7F	A <- A	A A MOV
MOV A,B	LD A,B	78	A <- B	B A MOV
MVI A,byte	LD A,byte	3Ebyte	A <- byte	byte A MVI
MOV A,C	LD A,C	79	A <- C	C A MOV
MOV A,D	LD A,D	7A	A <- D	D A MOV
MOV A,E	LD A,E	7B	A <- E	E A MOV
MOV A,H	LD A,H	7C	A <- H	H A MOV
---	LD A,I	ED57	A <- Interrupt Page	LDAI
MOV A,L	LD A,L	7D	A <- L	L A MOV
---	LD A,R	ED5F	A <- Refresh Register	LDAR
MOV B,M	LD B,(HL)	46	B <- (HL)	M B MOV
---	LD B,(IX+index)	DD46index	B <- (IX+index)	index +IX B MOV
---	LD B,(IY+index)	FD46index	B <- (IY+index)	index +IY B MOV
MOV B,A	LD B,A	47	B <- A	A B MOV
MOV B,B	LD B,B	40	B <- B	B B MOV
MVI B,byte	LD B,byte	06byte	B <- byte	byte B MVI
MOV B,C	LD B,C	41	B <- C	C B MOV
MOV B,D	LD B,D	42	B <- D	D B MOV

8080 Mnemonic		Z80 Mnemonic		Code	Operation	Forth Assembler
MOV	B,E	LD	B,E	43	B <- E	E B MOV
MOV	B,H	LD	B,H	44	B <- H	H B MOV
MOV	B,L	LD	B,L	45	B <- L	L B MOV
---		LD	BC,(word)	ED4Bword	BC <- (word)	word BC LXW
LXI	B,word	LD	BC,word	01word	BC <- word	word BC LXI
MOV	C,M	LD	C,(HL)	4E	C <- (HL)	M C MOV
---		LD	C,(IX+index)	DD4Eindex	C <- (IX+index)	index +IX C MOV
---		LD	C,(IY+index)	FD4Eindex	C <- (IY+index)	index +IY C MOV
MOV	C,A	LD	C,A	4F	C <- A	A C MOV
MOV	C,B	LD	C,B	48	C <- B	B C MOV
MVI	C,byte	LD	C,byte	0Ebyte	C <- byte	byte C MVI
MOV	C,C	LD	C,C	49	C <- C	C C MOV
MOV	C,D	LD	C,D	4A	C <- D	D C MOV
MOV	C,E	LD	C,E	4B	C <- E	E C MOV
MOV	C,H	LD	C,H	4C	C <- H	H C MOV
MOV	C,L	LD	C,L	4D	C <- L	L C MOV
MOV	D,M	LD	D,(HL)	56	D <- (HL)	M D MOV
---		LD	D,(IX+index)	DD56index	D <- (IX+index)	index +IX D MOV
---		LD	D,(IY+index)	FD56index	D <- (IY+index)	index +IY D MOV
MOV	D,A	LD	D,A	57	D <- A	A D MOV
MOV	D,B	LD	D,B	50	D <- B	B D MOV
MVI	D,byte	LD	D,byte	16byte	D <- byte	byte D MVI
MOV	D,C	LD	D,C	51	D <- C	C D MOV
MOV	D,D	LD	D,D	52	D <- D	D D MOV
MOV	D,E	LD	D,E	53	D <- E	E D MOV
MOV	D,H	LD	D,H	54	D <- H	H D MOV
MOV	D,L	LD	D,L	55	D <- L	L D MOV
---		LD	DE,(word)	ED5Bword	DE <- (word)	word DE LXW
LXI	D,word	LD	DE,word	11word	DE <- word	word DE LXI
MOV	E,M	LD	E,(HL)	5E	E <- (HL)	M E MOV
---		LD	E,(IX+index)	DD5Eindex	E <- (IX+index)	index +IX E MOV
---		LD	E,(IY+index)	FD5Eindex	E <- (IY+index)	index +IY E MOV
MOV	E,A	LD	E,A	5F	E <- A	A E MOV
MOV	E,B	LD	E,B	58	E <- B	B E MOV
MVI	E,byte	LD	E,byte	1Ebyte	E <- byte	byte E MVI
MOV	E,C	LD	E,C	59	E <- C	C E MOV
MOV	E,D	LD	E,D	5A	E <- D	D E MOV
MOV	E,E	LD	E,E	5B	E <- E	E E MOV
MOV	E,H	LD	E,H	5C	E <- H	H E MOV
MOV	E,L	LD	E,L	5D	E <- L	L E MOV
MOV	H,M	LD	H,(HL)	66	H <- (HL)	M H MOV
---		LD	H,(IX+index)	DD66index	H <- (IX+index)	index +IX H MOV
---		LD	H,(IY+index)	FD66index	H <- (IY+index)	index +IY H MOV
MOV	H,A	LD	H,A	67	H <- A	A H MOV
MOV	H,B	LD	H,B	60	H <- B	B H MOV
MVI	H,byte	LD	H,byte	26byte	H <- byte	byte H MVI
MOV	H,C	LD	H,C	61	H <- C	C H MOV
MOV	H,D	LD	H,D	62	H <- D	D H MOV
MOV	H,E	LD	H,E	63	H <- E	E H MOV

8080 Mnemonic	Z80 Mnemonic	Code	Operation	Forth Assembler
MOV H,H	LD H,H	64	H <- H	H H MOV
MOV H,L	LD H,L	65	H <- L	L H MOV
LHLD word	LD HL,(word)	2Aword	HL <- (word)	word LHLD
---	LD HL,(word)	ED6Bword	HL <- (word)	word HL LXW
LXI H,word	LD HL,word	21word	HL <- word	word HL LXI
---	LD I,A	ED47	Interrupt Page <- A	LDIA
---	LD IX,(word)	DD2Aword	IX <- (word)	word IX LXW
---	LD IX,word	DD21 word	IX <- word	word IX LXI
---	LD IY,(word)	FD2Aword	IY <- (word)	word IY LXW
---	LD IY,word	FD21word	IY <- word	word IY LXI
MOV L,M	LD L,(HL)	6E	L <- (HL)	M L MOV
---	LD L,(IX+index)	DD6Eindex	L <- (IX+index)	index +IX L MOV
---	LD L,(IY+index)	FD6Eindex	L <- (IY+index)	index +IY L MOV
MOV L,A	LD L,A	6F	L <- A	A L MOV
MOV L,B	LD L,B	68	L <- B	B L MOV
MVI L,byte	LD L,byte	2Ebyte	L <- byte	byte L MVI
MOV L,C	LD L,C	69	L <- C	C L MOV
MOV L,D	LD L,D	6A	L <- D	D L MOV
MOV L,E	LD L,E	6B	L <- E	E L MOV
MOV L,H	LD L,H	6C	L <- H	H L MOV
MOV L,L	LD L,L	6D	L <- L	L L MOV
---	LD R,A	ED4F	Refresh Register <- A	LDRA
---	LD SP,(word)	ED7Bword	SP <- (word)	word SP LXW
SPHL	LD SP,HL	F9	SP <- HL	SPHL
---	LD SP,IX	DDF9	SP <- IX	SPIX
---	LD SP,IY	FDF9	SP <- IY	SPIY
LXI SP,word	LD SP,word	31word	SP <- word	word SP LXI
---	LDD	EDA8	(DE) <- (HL);HL <- HL-1	LDD
---	LDDR	EDB8	(DE) <- (HL);HL <- HL-1	LDDR
---	LDI	EDA0	(DE) <- (HL);HL <- HL+1	LDI
---	LDIR	EDB0	(DE) <- (HL);HL <- HL+1	LDIR
---	NEG	ED44	A <- 0-A	NEG
NOP	NOP	0	No Operation	NOP
ORA A	OR A	B7	A <- A OR A	A OR
ORA B	OR B	B0	A <- A OR B	B OR
ORA C	OR C	B1	A <- A OR C	C OR
ORA D	OR D	B2	A <- A OR D	D OR
ORA E	OR E	B3	A <- A OR E	E OR
ORA H	OR H	B4	A <- A OR H	H OR
ORA L	OR L	B5	A <- A OR L	L OR
ORA M	OR (HL)	B6	A <- A OR (HL)	M OR
---	OR (IX+index)	DDB6index	A <- A OR (IX+index)	index +IX OR
---	OR (IY+index)	FDB6index	A <- A OR (IY+index)	index +IY OR
ORI byte	OR byte	F6byte	A <- A OR byte	byte ORI
---	OTDR	EDBB	[C] <- (HL);B <- B-1;HL <- HL-1	OTDR
---	OTIR	EDB3	[C] <- (HL);B <- B-1;HL <- HL+1	OTIR
OUT byte	OUT (byte),A	D3(byte)	[byte] <- A	byte OUT
---	OUT (C),A	ED79	[C] <- A	A OUT(C)
---	OUT (C),B	ED41	[C] <- B	B OUT(C)

8080 Mnemonic	Z80 Mnemonic	Code	Operation	Forth Assembler
---	OUT (C),C	ED49	[C] <- C	C OUT(C)
---	OUT (C),D	ED51	[C] <- D	D OUT(C)
---	OUT (C),E	ED59	[C] <- E	E OUT(C)
---	OUT (C),H	ED61	[C] <- H	H OUT(C)
---	OUT (C),L	ED69	[C] <- L	L OUT(C)
---	OUTD	EDAB	[C] <- (HL); B <- B-1; HL <- HL-1	OUTD
---	OUTI	EDA3	[C] <- (HL); B <- B-1; HL <- HL+1	OUTI
POP B	POP BC	C1	B <- (SP+1); C <- (SP);	BC POP
POP D	POP DE	D1	D <- (SP+1); E <- (SP);	DE POP
POP H	POP HL	E1	H <- (SP+1); L <- (SP);	HL POP
POP PSW	POP AF	F1	A <- (SP+1); Flags <- (SP);	F POP
---	POP IX	DDE1	IXh <- (SP+1); IXl <- (SP);	IX POP
---	POP IY	FDE1	IYh <- (SP+1); IYl <- (SP);	IY POP
PUSH B	PUSH BC	C5	(SP-2) <- C; (SP-1) <- B;	BC PUSH
PUSH D	PUSH DE	D5	(SP-2) <- E; (SP-1) <- D;	DE PUSH
PUSH H	PUSH HL	E5	(SP-2) <- L; (SP-1) <- H;	HL PUSH
PUSH PSW	PUSH AF	F5	(SP-2) <- Flags; (SP-1) <- A;	F PUSH
---	PUSH IX	DDE5	(SP-2) <- IXl; (SP-1) <- IXh	IX PUSH
---	PUSH IY	FDE5	(SP-2) <- IYl; (SP-1) <- IYh	IY PUSH
---	RES 0,A	CB87	0b <- 0	A 0 RES
---	RES 0,B	CB80	0b <- 0	B 0 RES
---	RES 0,C	CB81	0b <- 0	C 0 RES
---	RES 0,D	CB82	0b <- 0	D 0 RES
---	RES 0,E	CB83	0b <- 0	E 0 RES
---	RES 0,H	CB84	0b <- 0	H 0 RES
---	RES 0,L	CB85	0b <- 0	L 0 RES
---	RES 0,(HL)	CB86	0b <- 0	M 0 RES
---	RES 0,(IX+index)	DDCBindex86	0b <- 0	index +IX 0 RES
---	RES 0,(IY+index)	FDCBindex86	0b <- 0	index +IY 0 RES
---	RES 1,A	CB8F	1b <- 0	A 1 RES
---	RES 1,B	CB88	1b <- 0	B 1 RES
---	RES 1,C	CB89	1b <- 0	C 1 RES
---	RES 1,D	CB8A	1b <- 0	D 1 RES
---	RES 1,E	CB8B	1b <- 0	E 1 RES
---	RES 1,H	CB8C	1b <- 0	H 1 RES
---	RES 1,L	CB8D	1b <- 0	L 1 RES
---	RES 1,(HL)	CB8E	1b <- 0	M 1 RES
---	RES 1,(IX+index)	DDCBindex8E	1b <- 0	index +IX 1 RES
---	RES 1,(IY+index)	FDCBindex8E	1b <- 0	index +IY 1 RES
---	RES 2,A	CB97	2b <- 0	A 2 RES
---	RES 2,B	CB90	2b <- 0	B 2 RES
---	RES 2,C	CB91	2b <- 0	C 2 RES
---	RES 2,D	CB92	2b <- 0	D 2 RES
---	RES 2,E	CB93	2b <- 0	E 2 RES
---	RES 2,H	CB94	2b <- 0	H 2 RES
---	RES 2,L	CB95	2b <- 0	L 2 RES
---	RES 2,(HL)	CB96	2b <- 0	M 2 RES
---	RES 2,(IX+index)	DDCBindex96	2b <- 0	index +IX 2 RES
---	RES 2,(IY+index)	FDCBindex96	2b <- 0	index +IY 2 RES

8080 Mnemonic	Z80 Mnemonic	Code	Operation	Forth Assembler
---	RES 3,A	CB9F	3b <- 0	A 3 RES
---	RES 3,B	CB98	3b <- 0	B 3 RES
---	RES 3,C	CB99	3b <- 0	C 3 RES
---	RES 3,D	CB9A	3b <- 0	D 3 RES
---	RES 3,E	CB9B	3b <- 0	E 3 RES
---	RES 3,H	CB9C	3b <- 0	H 3 RES
---	RES 3,L	CB9D	3b <- 0	L 3 RES
---	RES 3,(HL)	CB9E	3b <- 0	M 3 RES
---	RES 3,(IX+index)	DDCBindex9E	3b <- 0	index +IX 3 RES
---	RES 3,(IY+index)	FDCBindex9E	3b <- 0	index +IY 3 RES
---	RES 4,A	CBA7	4b <- 0	A 4 RES
---	RES 4,B	CBA0	4b <- 0	B 4 RES
---	RES 4,C	CBA1	4b <- 0	C 4 RES
---	RES 4,D	CBA2	4b <- 0	D 4 RES
---	RES 4,E	CBA3	4b <- 0	E 4 RES
---	RES 4,H	CBA4	4b <- 0	H 4 RES
---	RES 4,L	CBA5	4b <- 0	L 4 RES
---	RES 4,(HL)	CBA6	4b <- 0	M 4 RES
---	RES 4,(IX+index)	DDCBindexA6	4b <- 0	index +IX 4 RES
---	RES 4,(IY+index)	FDCBindexA6	4b <- 0	index +IY 4 RES
---	RES 5,A	CBAF	5b <- 0	A 5 RES
---	RES 5,B	CBA8	5b <- 0	B 5 RES
---	RES 5,C	CBA9	5b <- 0	C 5 RES
---	RES 5,D	CBAA	5b <- 0	D 5 RES
---	RES 5,E	CBAB	5b <- 0	E 5 RES
---	RES 5,H	CBAC	5b <- 0	H 5 RES
---	RES 5,L	CBAD	5b <- 0	L 5 RES
---	RES 5,(HL)	CBAE	5b <- 0	M 5 RES
---	RES 5,(IX+index)	DDCBindexAE	5b <- 0	index +IX 5 RES
---	RES 5,(IY+index)	FDCBindexAE	5b <- 0	index +IY 5 RES
---	RES 6,A	CBB7	6b <- 0	A 6 RES
---	RES 6,B	CBB0	6b <- 0	B 6 RES
---	RES 6,C	CBB1	6b <- 0	C 6 RES
---	RES 6,D	CBB2	6b <- 0	D 6 RES
---	RES 6,E	CBB3	6b <- 0	E 6 RES
---	RES 6,H	CBB4	6b <- 0	H 6 RES
---	RES 6,L	CBB5	6b <- 0	L 6 RES
---	RES 6,(HL)	CBB6	6b <- 0	M 6 RES
---	RES 6,(IX+index)	DDCBindexB6	6b <- 0	index +IX 6 RES
---	RES 6,(IY+index)	FDCBindexB6	6b <- 0	index +IY 6 RES
---	RES 7,A	CBBF	7b <- 0	A 7 RES
---	RES 7,B	CBB8	7b <- 0	B 7 RES
---	RES 7,C	CBB9	7b <- 0	C 7 RES
---	RES 7,D	CBBA	7b <- 0	D 7 RES
---	RES 7,E	CBBB	7b <- 0	E 7 RES
---	RES 7,H	CBBC	7b <- 0	H 7 RES
---	RES 7,L	CBBD	7b <- 0	L 7 RES
---	RES 7,(HL)	CBBE	7b <- 0	M 7 RES
---	RES 7,(IX+index)	DDCBindexBE	7b <- 0	index +IX 7 RES

8080 Mnemonic	Z80 Mnemonic	Code	Operation	Forth Assembler
---	RES 7,(IY+index)	FDCBIndexBE	7b <- 0	index +IY 7 RES
RET	RET	C9	PCI <- (SP);PCh <- (SP+1)	RET
RNZ	RET NZ	C0	If NZ, RET	RET-NZ
RZ	RET Z	C8	If Z, RET	RET-Z
RNC	RET NC	D0	If NC, RET	RET-NC
RC	RET C	D8	If C, RET	RET-C
RPO	RET PO	E0	If PO, RET	RET-PO
RPE	RET PE	E8	If PE, RET	RET-PE
RP	RET P	F0	If P, RET	RET-P
RM	RET M	F8	If M, RET	RET-M
---	RETI	ED4D	Return from Interrupt	RETI
---	RETN	ED45	IFF1 <- IFF2;RETI	RETN
---	RL A	CB17	---	A RL
---	RL B	CB10	---	B RL
---	RL C	CB11	---	C RL
---	RL D	CB12	---	D RL
---	RL E	CB13	---	E RL
---	RL H	CB14	---	H RL
---	RL L	CB15	---	L RL
---	RL (HL)	CB16	---	M RL
---	RL (IX+index)	DDCBIndex16	---	index +IX RL
---	RL (IY+index)	FDCBIndex16	---	index +IY RL
---	RL (IX+index)	DDCBIndex1E	---	index +IX RR
---	RL (IY+index)	FDCBIndex1E	---	index +IY RR
RAL	RLA	17	---	RLA
---	RLC A	CB07	---	A RLC
---	RLC B	CB00	---	B RLC
---	RLC C	CB01	---	C RLC
---	RLC D	CB02	---	D RLC
---	RLC E	CB03	---	E RLC
---	RLC H	CB04	---	H RLC
---	RLC L	CB05	---	L RLC
---	RLC (HL)	CB06	---	M RLC
---	RLC (IX+index)	DDCBIndex06	---	index +IX RLC
---	RLC (IY+index)	FDCBIndex06	---	index +IY RLC
RLC	RLCA	7	---	RLCA
---	RLD	ED6F	---	RLD
---	RR A	CB1F	---	A RR
---	RR B	CB18	---	B RR
---	RR C	CB19	---	C RR
---	RR D	CB1A	---	D RR
---	RR E	CB1B	---	E RR
---	RR H	CB1C	---	H RR
---	RR L	CB1D	---	L RR
---	RR (HL)	CB1E	---	M RR
RAR	RRA	1F	---	RRA
---	RRC A	CB0F	---	A RRC
---	RRC B	CB08	---	B RRC
---	RRC C	CB09	---	C RRC

8080 Mnemonic	Z80 Mnemonic	Code	Operation	Forth Assembler
---	RRC D	CB0A	---	D RRC
---	RRC E	CB0B	---	E RRC
---	RRC H	CB0C	---	H RRC
---	RRC L	CB0D	---	L RRC
---	RRC (HL)	CB0E	---	M RRC
---	RRC (IX+index)	DDCBindex0E	---	index +IX RRC
---	RRC (IY+index)	FDCBindex0E	---	index +IY RRC
RRC	RRCA	0F	---	RRCA
---	RRD	ED67	---	RRD
RST	0 RST 0h	C7	CALL 0h	0 RST
RST	1 RST 8h	CF	CALL 8h	8 RST
RST	2 RST 10H	D7	CALL 10h	10 RST
RST	3 RST 18H	DF	CALL 18h	18 RST
RST	4 RST 20H	E7	CALL 20h	20 RST
RST	5 RST 28H	EF	CALL 28h	28 RST
RST	6 RST 30H	F7	CALL 30h	30 RST
RST	7 RST 38H	FF	CALL 38h	38 RST
SBB A	SBC A	9F	A <- A - A - Carry	A SBC
SBB B	SBC B	98	A <- A - B - Carry	B SBC
SBB C	SBC C	99	A <- A - C - Carry	C SBC
SBB D	SBC D	9A	A <- A - D - Carry	D SBC
SBB E	SBC E	9B	A <- A - E - Carry	E SBC
SBB H	SBC H	9C	A <- A - H - Carry	H SBC
SBB L	SBC L	9D	A <- A - L - Carry	L SBC
SBB M	SBC (HL)	9E	A <- A - (HL) - Carry	M SBC
---	SBC (IX+index)	DD9Eindex	A <- A - (IX+index) - Carry	index +IX SBC
---	SBC (IY+index)	FD9Eindex	A <- A - (IY+index) - Carry	index +IY SBC
SBI byte	SBC byte	DEbyte	A <- A - byte - Carry	byte SBI
---	SBC HL,BC	ED42	HL <- HL - BC - Carry	BC DSBC
---	SBC HL,DE	ED52	HL <- HL - DE - Carry	DE DSBC
---	SBC HL,HL	ED62	HL <- HL - HL - Carry	HL DSBC
---	SBC HL,SP	ED72	HL <- HL - SP - Carry	SP DSBC
STC	SCF	37	CF (Carry Flag) <- 1	SCF
---	SET 0,A	CBC7	0b <- 1	A 0 SET
---	SET 0,B	CBC0	0b <- 1	B 0 SET
---	SET 0,C	CBC1	0b <- 1	C 0 SET
---	SET 0,D	CBC2	0b <- 1	D 0 SET
---	SET 0,E	CBC3	0b <- 1	E 0 SET
---	SET 0,H	CBC4	0b <- 1	H 0 SET
---	SET 0,L	CBC5	0b <- 1	L 0 SET
---	SET 0,(HL)	CBC6	0b <- 1	M 0 SET
---	SET 0,(IX+index)	DDCBindexC6	0b <- 1	index +IX 0 SET
---	SET 0,(IY+index)	FDCBindexC6	0b <- 1	index +IY 0 SET
---	SET 1,A	CBCF	1b <- 1	A 1 SET
---	SET 1,B	CBC8	1b <- 1	B 1 SET
---	SET 1,C	CBC9	1b <- 1	C 1 SET
---	SET 1,D	CBCA	1b <- 1	D 1 SET
---	SET 1,E	CBCB	1b <- 1	E 1 SET
---	SET 1,H	CBCC	1b <- 1	H 1 SET

8080 Mnemonic	Z80 Mnemonic	Code	Operation	Forth Assembler
---	SET 1,L	CBCD	1b <- 1	L 1 SET
---	SET 1,(HL)	CBCE	1b <- 1	M 1 SET
---	SET 1,(IX+index)	DDCBIindexCE	1b <- 1	index +IX 1 SET
---	SET 1,(IY+index)	FDCBIindexCE	1b <- 1	index +IY 1 SET
---	SET 2,A	CBD7	2b <- 1	A 2 SET
---	SET 2,B	CBD0	2b <- 1	B 2 SET
---	SET 2,C	CBD1	2b <- 1	C 2 SET
---	SET 2,D	CBD2	2b <- 1	D 2 SET
---	SET 2,E	CBD3	2b <- 1	E 2 SET
---	SET 2,H	CBD4	2b <- 1	H 2 SET
---	SET 2,L	CBD5	2b <- 1	L 2 SET
---	SET 2,(HL)	CBD6	2b <- 1	M 2 SET
---	SET 2,(IX+index)	DDCBIindexD6	2b <- 1	index +IX 2 SET
---	SET 2,(IY+index)	FDCBIindexD6	2b <- 1	index +IY 2 SET
---	SET 3,A	CBDF	3b <- 1	A 3 SET
---	SET 3,B	CBD8	3b <- 1	B 3 SET
---	SET 3,C	CBD9	3b <- 1	C 3 SET
---	SET 3,D	CBDA	3b <- 1	D 3 SET
---	SET 3,E	CBDB	3b <- 1	E 3 SET
---	SET 3,H	CBDC	3b <- 1	H 3 SET
---	SET 3,L	CBDD	3b <- 1	L 3 SET
---	SET 3,(HL)	CBDE	3b <- 1	M 3 SET
---	SET 3,(IX+index)	DDCBIindexDE	3b <- 1	index +IX 3 SET
---	SET 3,(IY+index)	FDCBIindexDE	3b <- 1	index +IY 3 SET
---	SET 4,A	CBE7	4b <- 1	A 4 SET
---	SET 4,B	CBE0	4b <- 1	B 4 SET
---	SET 4,C	CBE1	4b <- 1	C 4 SET
---	SET 4,D	CBE2	4b <- 1	D 4 SET
---	SET 4,E	CBE3	4b <- 1	E 4 SET
---	SET 4,H	CBE4	4b <- 1	H 4 SET
---	SET 4,L	CBE5	4b <- 1	L 4 SET
---	SET 4,(HL)	CBE6	4b <- 1	M 4 SET
---	SET 4,(IX+index)	DDCBIindexE6	4b <- 1	index +IX 4 SET
---	SET 4,(IY+index)	FDCBIindexE6	4b <- 1	index +IY 4 SET
---	SET 5,A	CBEF	5b <- 1	A 5 SET
---	SET 5,B	CBE8	5b <- 1	B 5 SET
---	SET 5,C	CBE9	5b <- 1	C 5 SET
---	SET 5,D	CBEA	5b <- 1	D 5 SET
---	SET 5,E	CBEB	5b <- 1	E 5 SET
---	SET 5,H	CBEC	5b <- 1	H 5 SET
---	SET 5,L	CBED	5b <- 1	L 5 SET
---	SET 5,(HL)	CBEE	5b <- 1	M 5 SET
---	SET 5,(IX+index)	DDCBIindexEE	5b <- 1	index +IX 5 SET
---	SET 5,(IY+index)	FDCBIindexEE	5b <- 1	index +IY 5 SET
---	SET 6,A	CBF7	6b <- 1	A 6 SET
---	SET 6,B	CBF0	6b <- 1	B 6 SET
---	SET 6,C	CBF1	6b <- 1	C 6 SET
---	SET 6,D	CBF2	6b <- 1	D 6 SET
---	SET 6,E	CBF3	6b <- 1	E 6 SET

8080 Mnemonic	Z80 Mnemonic	Code	Operation	Forth Assembler
---	SET 6,H	CBF4	6b <- 1	H 6 SET
---	SET 6,L	CBF5	6b <- 1	L 6 SET
---	SET 6,(HL)	CBF6	6b <- 1	M 6 SET
---	SET 6,(IX+index)	DDCBIndexF6	6b <- 1	index +IX 6 SET
---	SET 6,(IY+index)	FDCBIndexF6	6b <- 1	index +IY 6 SET
---	SET 7,A	CBFF	7b <- 1	A 7 SET
---	SET 7,B	CBF8	7b <- 1	B 7 SET
---	SET 7,C	CBF9	7b <- 1	C 7 SET
---	SET 7,D	CBFA	7b <- 1	D 7 SET
---	SET 7,E	CBFB	7b <- 1	E 7 SET
---	SET 7,H	CBFC	7b <- 1	H 7 SET
---	SET 7,L	CBFD	7b <- 1	L 7 SET
---	SET 7,(HL)	CBFE	7b <- 1	M 7 SET
---	SET 7,(IX+index)	DDCBIndexFE	7b <- 1	index +IX 7 SET
---	SET 7,(IY+index)	FDCBIndexFE	7b <- 1	index +IY 7 SET
---	SLA A	CB27	---	A SLA
---	SLA B	CB20	---	B SLA
---	SLA C	CB21	---	C SLA
---	SLA D	CB22	---	D SLA
---	SLA E	CB23	---	E SLA
---	SLA H	CB24	---	H SLA
---	SLA L	CB25	---	L SLA
---	SLA (HL)	CB26	---	M SLA
---	SLA (IX+index)	DDCBIndex26	---	index +IX SLA
---	SLA (IY+index)	FDCBIndex26	---	index +IY SLA
---	SRA A	CB2F	---	A SRA
---	SRA B	CB28	---	B SRA
---	SRA C	CB29	---	C SRA
---	SRA D	CB2A	---	D SRA
---	SRA E	CB2B	---	E SRA
---	SRA H	CB2C	---	H SRA
---	SRA L	CB2D	---	L SRA
---	SRA (HL)	CB2E	---	M SRA
---	SRA (IX+index)	DDCBIndex2E	---	index +IX SRA
---	SRA (IY+index)	FDCBIndex2E	---	index +IY SRA
---	SRL A	CB3F	---	A SRL
---	SRL B	CB38	---	B SRL
---	SRL C	CB39	---	C SRL
---	SRL D	CB3A	---	D SRL
---	SRL E	CB3B	---	E SRL
---	SRL H	CB3C	---	H SRL
---	SRL L	CB3D	---	L SRL
---	SRL (HL)	CB3E	---	M SRL
---	SRL (IX+index)	DDCBIndex3E	---	index +IX SRL
---	SRL (IY+index)	FDCBIndex3E	---	index +IY SRL
SUB A	SUB A	97	A <- A - A	A SUB
SUB B	SUB B	90	A <- A - B	B SUB
SUB C	SUB C	91	A <- A - C	C SUB
SUB D	SUB D	92	A <- A - D	D SUB

8080 Mnemonic		Z80 Mnemonic		Code	Operation	Forth Assembler
SUB	E	SUB	E	93	A <- A - E	E SUB
SUB	H	SUB	H	94	A <- A - H	H SUB
SUB	L	SUB	L	95	A <- A - L	L SUB
SUB	M	SUB	(HL)	96	A <- A - (HL)	M SUB
---		SUB	(IX+index)	DD96index	A <- A - (IX+index)	index +IX SUB
---		SUB	(IY+index)	FD96index	A <- A - (IY+index)	index +IY SUB
SUI	byte	SUB	byte	D6byte	A <- A - byte	byte SUI
XRA	A	XOR	A	AF	A <- A XOR A	A XOR
XRA	B	XOR	B	A8	A <- A XOR B	B XOR
XRA	C	XOR	C	A9	A <- A XOR C	C XOR
XRA	D	XOR	D	AA	A <- A XOR D	D XOR
XRA	E	XOR	E	AB	A <- A XOR E	E XOR
XRA	H	XOR	H	AC	A <- A XOR H	H XOR
XRA	L	XOR	L	AD	A <- A XOR L	L XOR
XRA	M	XOR	(HL)	AE	A <- A XOR (HL)	M XOR
---		XOR	(IX+index)	DDAEindex	A <- A XOR (IX+index)	index +IX XOR
---		XOR	(IY+index)	FDAEindex	A <- A XOR (IY+index)	index +IY XOR
XRI	byte	XOR	byte	EEbyte	A <- A XOR byte	byte XRI