8080/Z80 Instruction Set

8 Bit Transfer Instructions

8080 Mnemonic		Z80 Mnemonic		Machine Code	Operation
MOV	A,A	LD	A,A	7F	A <- A
MOV	A,B	LD	A,B	78	A <- B
MOV	A,C	LD	A,C	79	A <- C
MOV	A,D	LD	A,D	7A	A <- D
MOV	A,E	LD	A,E	7B	A <- E
MOV	A,H	LD	A,H	7C	A <- H
MOV	A,L	LD	A,L	7D	A <- L
MOV	A,M	LD	A,(HL)	7E	A <- (HL)
LDAX	B	LD	A,(BC)	0A	A <- (BC)
LDAX	D	LD	A,(DE)	1A	A <- (DE)
LDA	**word**	LD	A,(**word**)	3A**word**	A <- (**word**)
---		LD	A,(IX+**index**)	DD7E**index**	A <- (IX+**index**)
---		LD	A,(IY+**index**)	FD7E**index**	A <- (IY+**index**)
MOV	B,A	LD	B,A	47	B <- A
MOV	B,B	LD	B,B	40	B <- B
MOV	B,C	LD	B,C	41	B <- C
MOV	B,D	LD	B,D	42	B <- D
MOV	B,E	LD	B,E	43	B <- E
MOV	B,H	LD	B,H	44	B <- H
MOV	B,L	LD	B,L	45	B <- L
MOV	B,M	LD	B,(HL)	46	B <- (HL)
---		LD	B,(IX+**index**)	DD46**index**	B <- (IX+**index**)
---		LD	B,(IY+**index**)	FD46**index**	B <- (IY+**index**)
MOV	C,A	LD	C,A	4F	C <- A
MOV	C,B	LD	C,B	48	C <- B
MOV	C,C	LD	C,C	49	C <- C
MOV	C,D	LD	C,D	4A	C <- D
MOV	C,E	LD	C,E	4B	C <- E
MOV	C,H	LD	C,H	4C	C <- H
MOV	C,L	LD	C,L	4D	C <- L
MOV	C,M	LD	C,(HL)	4E	C <- (HL)
---		LD	C,(IX+**index**)	DD4E**index**	C <- (IX+**index**)
---		LD	C,(IY+**index**)	FD4E**index**	C <- (IY+**index**)

MOV	D,A	LD	D,A	57	D <- A
MOV	D,B	LD	D,B	50	D <- B
MOV	D,C	LD	D,C	51	D <- C
MOV	D,D	LD	D,D	52	D <- D
MOV	D,E	LD	D,E	53	D <- E
MOV	D,H	LD	D,H	54	D <- H
MOV	D,L	LD	D,L	55	D <- L
MOV	D,M	LD	D,(HL)	56	D <- (HL)
---		LD	D,(IX+**index**)	DD56**index**	D <- (IX+**index**)
---		LD	D,(IY+**index**)	FD56**index**	D <- (IY+**index**)
MOV	E,A	LD	E,A	5F	E <- A
MOV	E,B	LD	E,B	58	E <- B
MOV	E,C	LD	E,C	59	E <- C
MOV	E,D	LD	E,D	5A	E <- D
MOV	E,E	LD	E,E	5B	E <- E
MOV	E,H	LD	E,H	5C	E <- H
MOV	E,L	LD	E,L	5D	E <- L
MOV	E,M	LD	E,(HL)	5E	E <- (HL)
---		LD	E,(IX+**index**)	DD5E**index**	E <- (IX+**index**)
---		LD	E,(IY+**index**)	FD5E**index**	E <- (IY+**index**)
MOV	H,A	LD	H,A	67	H <- A
MOV	H,B	LD	H,B	60	H <- B
MOV	H,C	LD	H,C	61	H <- C
MOV	H,D	LD	H,D	62	H <- D
MOV	H,E	LD	H,E	63	H <- E
MOV	H,H	LD	H,H	64	H <- H
MOV	H,L	LD	H,L	65	H <- L
MOV	H,M	LD	H,(HL)	66	H <- (HL)
---		LD	H,(IX+**index**)	DD66**index**	H <- (IX+**index**)
---		LD	H,(IY+**index**)	FD66**index**	H <- (IY+**index**)
MOV	L,A	LD	L,A	6F	L <- A
MOV	L,B	LD	L,B	68	L <- B
MOV	L,C	LD	L,C	69	L <- C
MOV	L,D	LD	L,D	6A	L <- D
MOV	L,E	LD	L,E	6B	L <- E
MOV	L,H	LD	L,H	6C	L <- H
MOV	L,L	LD	L,L	6D	L <- L
MOV	L,M	LD	L,(HL)	6E	L <- (HL)
---		LD	L,(IX+**index**)	DD6E**index**	L <- (IX+**index**)
---		LD	L,(IY+**index**)	FD6E**index**	L <- (IY+**index**)
MOV	M,A	LD	(HL),A	77	(HL) <- A
MOV	M,B	LD	(HL),B	70	(HL) <- B
MOV	M,C	LD	(HL),C	71	(HL) <- C
MOV	M,D	LD	(HL),D	72	(HL) <- D

MOV	M,E	LD	(HL),E	73	(HL) <- E
MOV	M,H	LD	(HL),H	74	(HL) <- H
MOV	M,L	LD	(HL),L	75	(HL) <- L
---		LD	(IX+**index**),A	DD77**index**	(IX+**index**) <- A
---		LD	(IX+**index**),B	DD70**index**	(IX+**index**) <- B
---		LD	(IX+**index**),C	DD71**index**	(IX+**index**) <- C
---		LD	(IX+**index**),D	DD72**index**	(IX+**index**) <- D
---		LD	(IX+**index**),E	DD73**index**	(IX+**index**) <- E
---		LD	(IX+**index**),H	DD74**index**	(IX+**index**) <- H
---		LD	(IX+**index**),L	DD75**index**	(IX+**index**) <- L
---		LD	(IX+**index**),**byte**	DD76**indexbyte**	(IX+**index**) <- **byte**
---		LD	(IY+**index**),A	FD77**index**	(IY+**index**) <- A
---		LD	(IY+**index**),B	FD70**index**	(IY+**index**) <- B
---		LD	(IY+**index**),C	FD71**index**	(IY+**index**) <- C
---		LD	(IY+**index**),D	FD72**index**	(IY+**index**) <- D
---		LD	(IY+**index**),E	FD73**index**	(IY+**index**) <- E
---		LD	(IY+**index**),H	FD74**index**	(IY+**index**) <- H
---		LD	(IY+**index**),L	FD75**index**	(IY+**index**) <- L
---		LD	(IY+**index**),**byte**	FD76**indexbyte**	(IY+**index**) <- **byte**
MVI	A,**byte**	LD	A,**byte**	3E**byte**	A <- **byte**
MVI	B,**byte**	LD	B,**byte**	06**byte**	B <- **byte**
MVI	C,**byte**	LD	C,**byte**	0E**byte**	C <- **byte**
MVI	D,**byte**	LD	D,**byte**	16**byte**	D <- **byte**
MVI	E,**byte**	LD	E,**byte**	1E**byte**	E <- **byte**
MVI	H,**byte**	LD	H,**byte**	26**byte**	H <- **byte**
MVI	L,**byte**	LD	L,**byte**	2E**byte**	L <- **byte**
MVI	M,**byte**	LD	(HL),**byte**	36**byte**	(HL) <- **byte**
---		LD	(IX+**index**),**byte**	DD36**index** **byte**	(IX+**index**) <- **byte**
---		LD	(IY+**index**),**byte**	FD36**index** **byte**	(IY+**index**) <- **byte**
STAX	B	LD	(BC),A	02	(BC) <- A
STAX	D	LD	(DE),A	12	(DE) <- A
STA	**word**	LD	(**word**),A	32**word**	(**word**) <- A

---

16 Bit Transfer Instructions

8080 Mnemonic		Z80 Mnemonic		Machine Code	Operation
LXI	B,**word**	LD	BC,**word**	01**word**	BC <- **word**
LXI	D,**word**	LD	DE,**word**	11**word**	DE <- **word**
LXI	H,**word**	LD	HL,**word**	21**word**	HL <- **word**
LXI	SP,**word**	LD	SP,**word**	31**word**	SP <- **word**
---		LD	IX,**word**	DD21**word**	IX <- **word**
---		LD	IY,**word**	FD21**word**	IY <- **word**
LHLD	**word**	LD	HL,(**word**)	2A**word**	HL <- (**word**)

---	LD	BC,(**word**)	ED4B**word**	BC <- (**word**)
---	LD	DE,(**word**)	ED5B**word**	DE <- (**word**)
---	LD	HL,(**word**)	ED6B**word**	HL <- (**word**)
---	LD	SP,(**word**)	ED7B**word**	SP <- (**word**)
---	LD	IX,(**word**)	DD2A**word**	IX <- (**word**)
---	LD	IY,(**word**)	FD2A**word**	IY <- (**word**)
SHLD **word**	LD	(**word**),HL	22**word**	(**word**) <- HL
---	LD	(**word**),BC	ED43**word**	(**word**) <- BC
---	LD	(**word**),DE	ED53**word**	(**word**) <- DE
---	LD	(**word**),HL	ED6B**word**	(**word**) <- HL
---	LD	(**word**),IX	DD22**word**	(**word**) <- IX
---	LD	(**word**),IY	DD22**word**	(**word**) <- IY
---	LD	(**word**),SP	ED73**word**	(**word**) <- SP
SPHL	LD	SP,HL	F9	SP <- HL
---	LD	SP,IX	DDF9	SP <- IX
---	LD	SP,IY	FDF9	SP <- IY

Register Exchange Instructions

8080 Mnemonic	Z80 Mnemonic		Machine Code	Operation
XCHG	EX	DE,HL	EB	HL <-> DE
XTHL	EX	(SP),HL	E3	H <-> (SP+1); L <-> (SP)
---	EX	(SP),IX	DDE3	IXh <-> (SP+1); IXl <-> (SP)
---	EX	(SP),IY	FDE3	IYh <-> (SP+1); IYl <-> (SP)
---	EX	AF,AF'	08	AF <-> AF'
---	EXX		D9	BC/DE/HL <-> BC'/DE'/HL'

Add Byte Instructions

8080 Mnemonic	Z80 Mnemonic		Machine Code	Operation
ADD A	ADD	A,A	87	A <- A + A
ADD B	ADD	A,B	80	A <- A + B
ADD C	ADD	A,C	81	A <- A + C
ADD D	ADD	A,D	82	A <- A + D
ADD E	ADD	A,E	83	A <- A + E
ADD H	ADD	A,H	84	A <- A + H
ADD L	ADD	A,L	85	A <- A + L
ADD M	ADD	A,(HL)	86	A <- A + (HL)
---	ADD	A,(IX+**index**)	DD86**index**	A <- A + (IX+**index**)
---	ADD	A,(IY+**index**)	FD86**index**	A <- A + (IY+**index**)
ADI **byte**	ADD	A,**byte**	C6**byte**	A <- A + **byte**

Add Byte with Carry-In Instructions

8080	Z80 Mnemonic		Machine Code	Operation

8080 Mnemonic		Z80 Mnemonic		Machine Code	Operation
ADC	A	ADC	A,A	8F	A <- A + A + Carry
ADC	B	ADC	A,B	88	A <- A + B + Carry
ADC	C	ADC	A,C	89	A <- A + C + Carry
ADC	D	ADC	A,D	8A	A <- A + D + Carry
ADC	E	ADC	A,E	8B	A <- A + E + Carry
ADC	H	ADC	A,H	8C	A <- A + H + Carry
ADC	L	ADC	A,L	8D	A <- A + L + Carry
ADC	M	ADC	A,(HL)	8E	A <- A + (HL) + Carry
---		ADC	A,(IX+**index**)	DD8E**index**	A <- A + (IX+**index**) + Carry
---		ADC	A,(IY+**index**)	FD8E**index**	A <- A + (IY+**index**) + Carry
ACI	**byte**	ADC	A,**byte**	CE**byte**	A <- A + **byte** + Carry

---

Subtract Byte Instructions

8080 Mnemonic		Z80 Mnemonic		Machine Code	Operation
SUB	A	SUB	A	97	A <- A - A
SUB	B	SUB	B	90	A <- A - B
SUB	C	SUB	C	91	A <- A - C
SUB	D	SUB	D	92	A <- A - D
SUB	E	SUB	E	93	A <- A - E
SUB	H	SUB	H	94	A <- A - H
SUB	L	SUB	L	95	A <- A - L
SUB	M	SUB	(HL)	96	A <- A - (HL)
---		SUB	(IX+**index**)	DD96**index**	A <- A - (IX+**index**)
---		SUB	(IY+**index**)	FD96**index**	A <- A - (IY+**index**)
SUI	**byte**	SUB	**byte**	D6**byte**	A <- A - **byte**

Subtract Byte With Borrow-In Instructions

8080 Mnemonic		Z80 Mnemonic		Machine Code	Operation
SBB	A	SBC	A	9F	A <- A - A - Carry
SBB	B	SBC	B	98	A <- A - B - Carry
SBB	C	SBC	C	99	A <- A - C - Carry
SBB	D	SBC	D	9A	A <- A - D - Carry
SBB	E	SBC	E	9B	A <- A - E - Carry
SBB	H	SBC	H	9C	A <- A - H - Carry
SBB	L	SBC	L	9D	A <- A - L - Carry
SBB	M	SBC	(HL)	9E	A <- A - (HL) - Carry
---		SBC	(IX+**index**)	DD9E**index**	A <- A - (IX+**index**) - Carry
---		SBC	(IY+**index**)	FD9E**index**	A <- A - (IY+**index**) - Carry
SBI	**byte**	SBC	**byte**	DE**byte**	A <- A - **byte** - Carry

---

Double Byte Add Instructions

8080 Mnemonic	Z80 Mnemonic		Machine Code	Operation
DAD B	ADD	HL,BC	09	HL <- HL + BC
DAD D	ADD	HL,DE	19	HL <- HL + DE
DAD H	ADD	HL,HL	29	HL <- HL + HL
DAD SP	ADD	HL,SP	39	HL <- HL + SP
---	ADD	IX,BC	DD09	IX <- IX + BC
---	ADD	IX,DE	DD19	IX <- IX + DE
---	ADD	IX,IX	DD29	IX <- IX + IX
---	ADD	IX,SP	DD39	IX <- IX + SP
---	ADD	IY,BC	FD09	IY <- IY + BC
---	ADD	IY,DE	FD19	IY <- IY + DE
---	ADD	IY,IY	FD29	IY <- IY + IY
---	ADD	IY,SP	FD39	IY <- IY + SP

Double Byte Add With Carry-In Instructions

8080 Mnemonic	Z80 Mnemonic		Machine Code	Operation
---	ADC	HL,BC	ED4A	HL <- HL + BC + Carry
---	ADC	HL,DE	ED5A	HL <- HL + DE + Carry
---	ADC	HL,HL	ED6A	HL <- HL + HL + Carry
---	ADC	HL,SP	ED7A	HL <- HL + SP + Carry

Double Byte Subtract With Borrow-In Instructions

8080 Mnemonic	Z80 Mnemonic		Machine Code	Operation
---	SBC	HL,BC	ED42	HL <- HL - BC - Carry
---	SBC	HL,DE	ED52	HL <- HL - DE - Carry
---	SBC	HL,HL	ED62	HL <- HL - HL - Carry
---	SBC	HL,SP	ED72	HL <- HL - SP - Carry

Control Instructions

8080 Mnemonic	Z80 Mnemonic		Machine Code	Operation
DI	DI		F3	IFF <- 0
EI	EI		FB	IFF <- 1
---	IM	0	ED46	---
---	IM	1	ED56	---
---	IM	2	ED5E	---
---	LD	A,I	ED57	A <- Interrupt Page
---	LD	I,A	ED47	Interrupt Page <- A
---	LD	A,R	ED5F	A <- Refresh Register
---	LD	R,A	ED4F	Refresh Register <- A
NOP	NOP		00	No Operation

HLT	HLT	76	NOP;PC <- PC-1

Increment Byte Instructions

8080 Mnemonic	Z80 Mnemonic	Machine Code	Operation
INR A	INC A	3C	A <- A + 1
INR B	INC B	04	B <- B + 1
INR C	INC C	0C	C <- C + 1
INR D	INC D	14	D <- D + 1
INR E	INC E	1C	E <- E + 1
INR H	INC H	24	H <- H + 1
INR L	INC L	2C	L <- L + 1
INR M	INC (HL)	34	(HL) <- (HL) + 1
---	INC (IX+**index**)	DD34**index**	(IX+**index**) <- (IX+**index**) + 1
---	INC (IY+**index**)	FD34**index**	(IY+**index**) <- (IY+**index**) + 1

Decrement Byte Instructions

8080 Mnemonic	Z80 Mnemonic	Machine Code	Operation
DCR A	DEC A	3D	A <- A - 1
DCR B	DEC B	05	B <- B - 1
DCR C	DEC C	0D	C <- C - 1
DCR D	DEC D	15	D <- D - 1
DCR E	DEC E	1D	E <- E - 1
DCR H	DEC H	25	H <- H - 1
DCR L	DEC L	2D	L <- L - 1
DCR M	DEC (HL)	35	(HL) <- (HL) - 1
---	DEC (IX+**index**)	DD35**index**	(IX+**index**) <- (IX+**index**) - 1
---	DEC (IY+**index**)	FD35**index**	(IY+**index**) <- (IY+**index**) - 1

Increment Register Pair Instructions

8080 Mnemonic	Z80 Mnemonic	Machine Code	Operation
INX B	INC BC	03	BC <- BC + 1
INX D	INC DE	13	DE <- DE + 1
INX H	INC HL	23	HL <- HL + 1
INX SP	INC SP	33	SP <- SP + 1
---	INC IX	DD23	IX <- IX + 1
---	INC IY	FD23	IY <- IY + 1

Decrement Register Pair Instructions

8080 Mnemonic	Z80 Mnemonic	Machine Code	Operation
DCX B	DEC BC	0B	BC <- BC - 1

DCX	D	DEC	DE	1B	DE <- DE - 1
DCX	H	DEC	HL	2B	HL <- HL - 1
DCX	SP	DEC	SP	3B	SP <- SP - 1
---		DEC	IX	DD2B	IX <- IX - 1
---		DEC	IY	FD2B	IY <- IY - 1

Special Accumulator and Flag Instructions

8080 Mnemonic	Z80 Mnemonic	Machine Code	Operation
DAA	DAA	27	---
CMA	CPL	2F	A <- NOT A
STC	SCF	37	CF (Carry Flag) <- 1
CMC	CCF	3F	CF (Carry Flag) <- NOT CF
---	NEG	ED44	A <- 0-A

Rotate Instructions

8080 Mnemonic	Z80 Mnemonic		Machine Code	Operation
RLC	RLCA		07	---
RRC	RRCA		0F	---
RAL	RLA		17	---
RAR	RRA		1F	---
---	RLD		ED6F	---
---	RRD		ED67	---
---	RLC	A	CB07	---
---	RLC	B	CB00	---
---	RLC	C	CB01	---
---	RLC	D	CB02	---
---	RLC	E	CB03	---
---	RLC	H	CB04	---
---	RLC	L	CB05	---
---	RLC	(HL)	CB06	---
---	RLC	(IX+**index**)	DDCB**index**06	---
---	RLC	(IY+**index**)	FDCB**index**06	---
---	RL	A	CB17	---
---	RL	B	CB10	---
---	RL	C	CB11	---
---	RL	D	CB12	---
---	RL	E	CB13	---
---	RL	H	CB14	---
---	RL	L	CB15	---
---	RL	(HL)	CB16	---
---	RL	(IX+**index**)	DDCB**index**16	---
---	RL	(IY+**index**)	FDCB**index**16	---

---	RRC	A	CB0F	---
---	RRC	B	CB08	---
---	RRC	C	CB09	---
---	RRC	D	CB0A	---
---	RRC	E	CB0B	---
---	RRC	H	CB0C	---
---	RRC	L	CB0D	---
---	RRC	(HL)	CB0E	---
---	RRC	(IX+**index**)	DDCB**index**0E	---
---	RRC	(IY+**index**)	FDCB**index**0E	---
---	RL	A	CB1F	---
---	RL	B	CB18	---
---	RL	C	CB19	---
---	RL	D	CB1A	---
---	RL	E	CB1B	---
---	RL	H	CB1C	---
---	RL	L	CB1D	---
---	RL	(HL)	CB1E	---
---	RL	(IX+**index**)	DDCB**index**1E	---
---	RL	(IY+**index**)	FDCB**index**1E	---

Logical Byte Instructions

8080 Mnemonic		Z80 Mnemonic		Machine Code	Operation
ANA	A	AND	A	A7	A <- A AND A
ANA	B	AND	B	A0	A <- A AND B
ANA	C	AND	C	A1	A <- A AND C
ANA	D	AND	D	A2	A <- A AND D
ANA	E	AND	E	A3	A <- A AND E
ANA	H	AND	H	A4	A <- A AND H
ANA	L	AND	L	A5	A <- A AND L
ANA	M	AND	(HL)	A6	A <- A AND (HL)
---		AND	(IX+**index**)	DDA6**index**	A <- A AND (IX+**index**)
---		AND	(IY+**index**)	FDA6**index**	A <- A AND (IY+**index**)
ANI	**byte**	AND	**byte**	E6**byte**	A <- A AND **byte**
XRA	A	XOR	A	AF	A <- A XOR A
XRA	B	XOR	B	A8	A <- A XOR B
XRA	C	XOR	C	A9	A <- A XOR C
XRA	D	XOR	D	AA	A <- A XOR D
XRA	E	XOR	E	AB	A <- A XOR E
XRA	H	XOR	H	AC	A <- A XOR H
XRA	L	XOR	L	AD	A <- A XOR L
XRA	M	XOR	(HL)	AE	A <- A XOR (HL)
---		XOR	(IX+**index**)	DDAE**index**	A <- A XOR (IX+**index**)

---		XOR	(IY+**index**)	FDAE**index**	A <- A XOR (IY+**index**)
XRI	**byte**	XOR	**byte**	EE**byte**	A <- A XOR **byte**
ORA	A	OR	A	B7	A <- A OR A
ORA	B	OR	B	B0	A <- A OR B
ORA	C	OR	C	B1	A <- A OR C
ORA	D	OR	D	B2	A <- A OR D
ORA	E	OR	E	B3	A <- A OR E
ORA	H	OR	H	B4	A <- A OR H
ORA	L	OR	L	B5	A <- A OR L
ORA	M	OR	(HL)	B6	A <- A OR (HL)
---		OR	(IX+**index**)	DDB6**index**	A <- A OR (IX+**index**)
---		OR	(IY+**index**)	FDB6**index**	A <- A OR (IY+**index**)
ORI	**byte**	OR	**byte**	F6**byte**	A <- A OR **byte**
CMP	A	CP	A	BF	A - A
CMP	B	CP	B	B8	A - B
CMP	C	CP	C	B9	A - C
CMP	D	CP	D	BA	A - D
CMP	E	CP	E	BB	A - E
CMP	H	CP	H	BC	A - H
CMP	L	CP	L	BD	A - L
CMP	M	CP	(HL)	BE	A - (HL)
---		CP	(IX+**index**)	DDBE**index**	A - (IX+**index**)
---		CP	(IY+**index**)	FDBE**index**	A - (IY+**index**)
CPI	**byte**	CP	**byte**	FE**byte**	A - **byte**
---		CPI		EDA1	A - (HL);HL <- HL+1;BC <- BC-1
---		CPIR		EDB1	A - (HL);HL <- HL+1;BC <- BC-1
---		CPD		EDA9	A - (HL);HL <- HL-1;BC <- BC-1
---		CPDR		EDB9	A - (HL);HL <- HL-1;BC <- BC-1

Branch Control/Program Counter Load Instructions

8080 Mnemonic		Z80 Mnemonic		Machine Code	Operation
JMP	address	JP	address	C3address	PC <- address
JNZ	address	JP	NZ,address	C2address	If NZ, PC <- address
JZ	address	JP	Z,address	CAaddress	If Z, PC <- address
JNC	address	JP	NC,address	D2address	If NC, PC <- address
JC	address	JP	C,address	DAaddress	If C, PC <- address
JPO	address	JP	PO,address	E2address	If PO, PC <- address
JPE	address	JP	PE,address	EAaddress	If PE, PC <- address
JP	address	JP	P,address	F2address	If P, PC <- address
JM	address	JP	M,address	FAaddress	If M, PC <- address
PCHL		JP	(HL)	E9	PC <- HL
---		JP	(IX)	DDE9	PC <- IX
---		JP	(IY)	FDE9	PC <- IY

---		JR	**index**	18**index**	PC <- PC + **index**
---		JR	NZ,**index**	20**index**	If NZ, PC <- PC + **index**
---		JR	Z,**index**	28**index**	If Z, PC <- PC + **index**
---		JR	NC,**index**	30**index**	If NC, PC <- PC + **index**
---		JR	C,**index**	38**index**	If C, PC <- PC + **index**
---		DJNZ	**index**	10**index**	B <- B - 1; while B > 0, PC <- PC + **index**
CALL	address	CALL	address	CDaddress	(SP-1) <- PCh;(SP-2) <- PCl; SP <- SP - 2;PC <- address
CNZ	address	CALL	NZ,address	C4address	If NZ, CALL address
CZ	address	CALL	Z,address	CCaddress	If Z, CALL address
CNC	address	CALL	NC,address	D4address	If NC, CALL address
CC	address	CALL	C,address	DCaddress	If C, CALL address
CPO	address	CALL	PO,address	E4address	If PO, CALL address
CPE	address	CALL	PE,address	ECaddress	If PE, CALL address
CP	address	CALL	P,address	F4address	If P, CALL address
CM	address	CALL	M,address	FCaddress	If M, CALL address
RET		RET		C9	PCl <- (SP);PCh <- (SP+1); SP <- (SP+2)
RNZ		RET	NZ	C0	If NZ, RET
RZ		RET	Z	C8	If Z, RET
RNC		RET	NC	D0	If NC, RET
RC		RET	C	D8	If C, RET
RPO		RET	PO	E0	If PO, RET
RPE		RET	PE	E8	If PE, RET
RP		RET	P	F0	If P, RET
RM		RET	M	F8	If M, RET
---		RETI		ED4D	Return from Interrupt
---		RETN		ED45	IFF1 <- IFF2;RETI
RST	0	RST	0	C7	CALL 0
RST	1	RST	8	CF	CALL 8
RST	2	RST	10H	D7	CALL 10H
RST	3	RST	18H	DF	CALL 18H
RST	4	RST	20H	E7	CALL 20H
RST	5	RST	28H	EF	CALL 28H
RST	6	RST	30H	F7	CALL 30H
RST	7	RST	38H	FF	CALL 38H

Stack Operation Instructions

8080 Mnemonic		Z80 Mnemonic		Machine Code	Operation
PUSH	B	PUSH	BC	C5	(SP-2) <- C; (SP-1) <- B; SP <- SP - 2
PUSH	D	PUSH	DE	D5	(SP-2) <- E; (SP-1) <- D; SP <- SP - 2
PUSH	H	PUSH	HL	E5	(SP-2) <- L; (SP-1) <- H; SP <- SP - 2
PUSH	PSW	PUSH	AF	F5	(SP-2) <- Flags; (SP-1) <- A; SP <- SP - 2
---		PUSH	IX	DDE5	(SP-2) <- IXl; (SP-1) <- IXh; SP <- SP - 2

---		PUSH	IY	FDE5	(SP-2) <- IYl; (SP-1) <- IYh; SP <- SP - 2
POP	B	POP	BC	C1	B <- (SP+1); C <- (SP); SP <- SP + 2
POP	D	POP	DE	D1	D <- (SP+1); E <- (SP); SP <- SP + 2
POP	H	POP	HL	E1	H <- (SP+1); L <- (SP); SP <- SP + 2
POP	PSW	POP	AF	F1	A <- (SP+1); Flags <- (SP); SP <- SP + 2
---		POP	IX	DDE1	IXh <- (SP+1); IXl <- (SP); SP <- (SP+2)
---		POP	IY	FDE1	IYh <- (SP+1); IYl <- (SP); SP <= (SP+2)

Input/Output Instructions

8080 Mnemonic		Z80 Mnemonic		Machine Code	Operation
IN	**byte**	IN	A,(**byte**)	DB**byte**	A <- [**byte**]
---		IN	A,(C)	ED78	A <- [C]
---		IN	B,(C)	ED40	B <- [C]
---		IN	C,(C)	ED48	C <- [C]
---		IN	D,(C)	ED50	D <- [C]
---		IN	E,(C)	ED58	E <- [C]
---		IN	H,(C)	ED60	H <- [C]
---		IN	L,(C)	ED68	L <- [C]
---		INI		EDA2	(HL) <- [C];B <- B-1;HL <- HL+1
---		INIR		EDB2	(HL) <- [C];B <- B-1;HL <- HL+1; Repeat while B>0
---		IND		EDAA	(HL) <- [C];B <- B-1;HL <- HL-1
---		INDR		EDBA	(HL) <- [C];B <- B-1;HL <- HL-1; Repeat while B>0
OUT	**byte**	OUT	(**byte**),A	D320	[**byte**] <- A
---		OUT	(C),A	ED79	[C] <- A
---		OUT	(C),B	ED41	[C] <- B
---		OUT	(C),C	ED49	[C] <- C
---		OUT	(C),D	ED51	[C] <- D
---		OUT	(C),E	ED59	[C] <- E
---		OUT	(C),H	ED61	[C] <- H
---		OUT	(C),L	ED69	[C] <- L
---		OUTI		EDA3	[C] <- (HL);B <- B-1;HL <- HL+1
---		OTIR		EDB3	[C] <- (HL);B <- B-1;HL <- HL+1; Repeat while B>0
---		OUTD		EDAB	[C] <- (HL);B <- B-1;HL <- HL-1
---		OTDR		EDBB	[C] <- (HL);B <- B-1;HL <- HL-1; Repeat while B>0

Data Transfer Instructions (Z80 Only)

8080 Mnemonic	Z80 Mnemonic	Machine Code	Operation
---	LDI	EDA0	(DE) <- (HL);HL <- HL+1; DE <- DE+1; BC <- BC-1

---	LDIR	EDB0	(DE) <- (HL);HL <- HL+1; DE <- DE+1; BC <- BC-1; repeat while BC<>-1
---	LDD	EDA8	(DE) <- (HL);HL <- HL-1; DE <- DE-1; BC <- BC-1
---	LDDR	EDB8	(DE) <- (HL);HL <- HL-1; DE <- DE-1; BC <- BC-1; repeat while BC<>-1

Bit Manipulation Instructions (Z80 Only)

8080 Mnemonic	Z80 Mnemonic		Machine Code	Operation
---	BIT	0,A	CB47	Z flag <- NOT Bit 0
---	BIT	0,B	CB40	Z flag <- NOT Bit 0
---	BIT	0,C	CB41	Z flag <- NOT Bit 0
---	BIT	0,D	CB42	Z flag <- NOT Bit 0
---	BIT	0,E	CB43	Z flag <- NOT Bit 0
---	BIT	0,H	CB44	Z flag <- NOT Bit 0
---	BIT	0,L	CB45	Z flag <- NOT Bit 0
---	BIT	0,(HL)	CB46	Z flag <- NOT Bit 0
---	BIT	0,(IX+**index**)	DDCB**index**46	Z flag <- NOT Bit 0
---	BIT	0,(IY+**index**)	FDCB**index**46	Z flag <- NOT Bit 0
---	BIT	1,A	CB4F	Z flag <- NOT Bit 1
---	BIT	1,B	CB48	Z flag <- NOT Bit 1
---	BIT	1,C	CB49	Z flag <- NOT Bit 1
---	BIT	1,D	CB4A	Z flag <- NOT Bit 1
---	BIT	1,E	CB4B	Z flag <- NOT Bit 1
---	BIT	1,H	CB4C	Z flag <- NOT Bit 1
---	BIT	1,L	CB4D	Z flag <- NOT Bit 1
---	BIT	1,(HL)	CB4E	Z flag <- NOT Bit 1
---	BIT	1,(IX+**index**)	DDCB**index**4E	Z flag <- NOT Bit 1
---	BIT	1,(IY+**index**)	FDCB**index**4E	Z flag <- NOT Bit 1
---	BIT	2,A	CB57	Z flag <- NOT Bit 2
---	BIT	2,B	CB50	Z flag <- NOT Bit 2
---	BIT	2,C	CB51	Z flag <- NOT Bit 2
---	BIT	2,D	CB52	Z flag <- NOT Bit 2
---	BIT	2,E	CB53	Z flag <- NOT Bit 2
---	BIT	2,H	CB54	Z flag <- NOT Bit 2
---	BIT	2,L	CB55	Z flag <- NOT Bit 2
---	BIT	2,(HL)	CB56	Z flag <- NOT Bit 2
---	BIT	2,(IX+**index**)	DDCB**index**56	Z flag <- NOT Bit 2
---	BIT	2,(IY+**index**)	FDCB**index**56	Z flag <- NOT Bit 2
---	BIT	3,A	CB5F	Z flag <- NOT Bit 3
---	BIT	3,B	CB58	Z flag <- NOT Bit 3
---	BIT	3,C	CB59	Z flag <- NOT Bit 3
---	BIT	3,D	CB5A	Z flag <- NOT Bit 3
---	BIT	3,E	CB5B	Z flag <- NOT Bit 3

---	BIT	3,H	CB5C	Z flag <- NOT Bit 3
---	BIT	3,L	CB5D	Z flag <- NOT Bit 3
---	BIT	3,(HL)	CB5E	Z flag <- NOT Bit 3
---	BIT	3,(IX+**index**)	DDCB**index**5E	Z flag <- NOT Bit 3
---	BIT	3,(IY+**index**)	FDCB**index**5E	Z flag <- NOT Bit 3
---	BIT	4,A	CB67	Z flag <- NOT Bit 4
---	BIT	4,B	CB60	Z flag <- NOT Bit 4
---	BIT	4,C	CB61	Z flag <- NOT Bit 4
---	BIT	4,D	CB62	Z flag <- NOT Bit 4
---	BIT	4,E	CB63	Z flag <- NOT Bit 4
---	BIT	4,H	CB64	Z flag <- NOT Bit 4
---	BIT	4,L	CB65	Z flag <- NOT Bit 4
---	BIT	4,(HL)	CB66	Z flag <- NOT Bit 4
---	BIT	4,(IX+**index**)	DDCB**index**66	Z flag <- NOT Bit 4
---	BIT	4,(IY+**index**)	FDCB**index**66	Z flag <- NOT Bit 4
---	BIT	5,A	CB6F	Z flag <- NOT Bit 5
---	BIT	5,B	CB68	Z flag <- NOT Bit 5
---	BIT	5,C	CB69	Z flag <- NOT Bit 5
---	BIT	5,D	CB6A	Z flag <- NOT Bit 5
---	BIT	5,E	CB6B	Z flag <- NOT Bit 5
---	BIT	5,H	CB6C	Z flag <- NOT Bit 5
---	BIT	5,L	CB6D	Z flag <- NOT Bit 5
---	BIT	5,(HL)	CB6E	Z flag <- NOT Bit 5
---	BIT	5,(IX+**index**)	DDCB**index**6E	Z flag <- NOT Bit 5
---	BIT	5,(IY+**index**)	FDCB**index**6E	Z flag <- NOT Bit 5
---	BIT	6,A	CB77	Z flag <- NOT Bit 6
---	BIT	6,B	CB70	Z flag <- NOT Bit 6
---	BIT	6,C	CB71	Z flag <- NOT Bit 6
---	BIT	6,D	CB72	Z flag <- NOT Bit 6
---	BIT	6,E	CB73	Z flag <- NOT Bit 6
---	BIT	6,H	CB74	Z flag <- NOT Bit 6
---	BIT	6,L	CB75	Z flag <- NOT Bit 6
---	BIT	6,(HL)	CB76	Z flag <- NOT Bit 6
---	BIT	6,(IX+**index**)	DDCB**index**76	Z flag <- NOT Bit 6
---	BIT	6,(IY+**index**)	FDCB**index**76	Z flag <- NOT Bit 6
---	BIT	7,A	CB7F	Z flag <- NOT Bit 7
---	BIT	7,B	CB78	Z flag <- NOT Bit 7
---	BIT	7,C	CB79	Z flag <- NOT Bit 7
---	BIT	7,D	CB7A	Z flag <- NOT Bit 7
---	BIT	7,E	CB7B	Z flag <- NOT Bit 7
---	BIT	7,H	CB7C	Z flag <- NOT Bit 7
---	BIT	7,L	CB7D	Z flag <- NOT Bit 7
---	BIT	7,(HL)	CB7E	Z flag <- NOT Bit 7
---	BIT	7,(IX+**index**)	DDCB**index**7E	Z flag <- NOT Bit 7

---	BIT	7,(IY+**index**)	FDCB**index**7E	Z flag <- NOT Bit 7
---	RES	0,A	CB87	Bit 0 <- 0
---	RES	0,B	CB80	Bit 0 <- 0
---	RES	0,C	CB81	Bit 0 <- 0
---	RES	0,D	CB82	Bit 0 <- 0
---	RES	0,E	CB83	Bit 0 <- 0
---	RES	0,H	CB84	Bit 0 <- 0
---	RES	0,L	CB85	Bit 0 <- 0
---	RES	0,(HL)	CB86	Bit 0 <- 0
---	RES	0,(IX+**index**)	DDCB**index**86	Bit 0 <- 0
---	RES	0,(IY+**index**)	FDCB**index**86	Bit 0 <- 0
---	RES	1,A	CB8F	Bit 1 <- 0
---	RES	1,B	CB88	Bit 1 <- 0
---	RES	1,C	CB89	Bit 1 <- 0
---	RES	1,D	CB8A	Bit 1 <- 0
---	RES	1,E	CB8B	Bit 1 <- 0
---	RES	1,H	CB8C	Bit 1 <- 0
---	RES	1,L	CB8D	Bit 1 <- 0
---	RES	1,(HL)	CB8E	Bit 1 <- 0
---	RES	1,(IX+**index**)	DDCB**index**8E	Bit 1 <- 0
---	RES	1,(IY+**index**)	FDCB**index**8E	Bit 1 <- 0
---	RES	2,A	CB97	Bit 2 <- 0
---	RES	2,B	CB90	Bit 2 <- 0
---	RES	2,C	CB91	Bit 2 <- 0
---	RES	2,D	CB92	Bit 2 <- 0
---	RES	2,E	CB93	Bit 2 <- 0
---	RES	2,H	CB94	Bit 2 <- 0
---	RES	2,L	CB95	Bit 2 <- 0
---	RES	2,(HL)	CB96	Bit 2 <- 0
---	RES	2,(IX+**index**)	DDCB**index**96	Bit 2 <- 0
---	RES	2,(IY+**index**)	FDCB**index**96	Bit 2 <- 0
---	RES	3,A	CB9F	Bit 3 <- 0
---	RES	3,B	CB98	Bit 3 <- 0
---	RES	3,C	CB99	Bit 3 <- 0
---	RES	3,D	CB9A	Bit 3 <- 0
---	RES	3,E	CB9B	Bit 3 <- 0
---	RES	3,H	CB9C	Bit 3 <- 0
---	RES	3,L	CB9D	Bit 3 <- 0
---	RES	3,(HL)	CB9E	Bit 3 <- 0
---	RES	3,(IX+**index**)	DDCB**index**9E	Bit 3 <- 0
---	RES	3,(IY+**index**)	FDCB**index**9E	Bit 3 <- 0
---	RES	4,A	CBA7	Bit 4 <- 0
---	RES	4,B	CBA0	Bit 4 <- 0
---	RES	4,C	CBA1	Bit 4 <- 0

---	RES	4,D	CBA2	Bit 4 <- 0
---	RES	4,E	CBA3	Bit 4 <- 0
---	RES	4,H	CBA4	Bit 4 <- 0
---	RES	4,L	CBA5	Bit 4 <- 0
---	RES	4,(HL)	CBA6	Bit 4 <- 0
---	RES	4,(IX+**index**)	DDCB**index**A6	Bit 4 <- 0
---	RES	4,(IY+**index**)	FDCB**index**A6	Bit 4 <- 0
---	RES	5,A	CBAF	Bit 5 <- 0
---	RES	5,B	CBA8	Bit 5 <- 0
---	RES	5,C	CBA9	Bit 5 <- 0
---	RES	5,D	CBAA	Bit 5 <- 0
---	RES	5,E	CBAB	Bit 5 <- 0
---	RES	5,H	CBAC	Bit 5 <- 0
---	RES	5,L	CBAD	Bit 5 <- 0
---	RES	5,(HL)	CBAE	Bit 5 <- 0
---	RES	5,(IX+**index**)	DDCB**index**AE	Bit 5 <- 0
---	RES	5,(IY+**index**)	FDCB**index**AE	Bit 5 <- 0
---	RES	6,A	CBB7	Bit 6 <- 0
---	RES	6,B	CBB0	Bit 6 <- 0
---	RES	6,C	CBB1	Bit 6 <- 0
---	RES	6,D	CBB2	Bit 6 <- 0
---	RES	6,E	CBB3	Bit 6 <- 0
---	RES	6,H	CBB4	Bit 6 <- 0
---	RES	6,L	CBB5	Bit 6 <- 0
---	RES	6,(HL)	CBB6	Bit 6 <- 0
---	RES	6,(IX+**index**)	DDCB**index**B6	Bit 6 <- 0
---	RES	6,(IY+**index**)	FDCB**index**B6	Bit 6 <- 0
---	RES	7,A	CBBF	Bit 7 <- 0
---	RES	7,B	CBB8	Bit 7 <- 0
---	RES	7,C	CBB9	Bit 7 <- 0
---	RES	7,D	CBBA	Bit 7 <- 0
---	RES	7,E	CBBB	Bit 7 <- 0
---	RES	7,H	CBBC	Bit 7 <- 0
---	RES	7,L	CBBD	Bit 7 <- 0
---	RES	7,(HL)	CBBE	Bit 7 <- 0
---	RES	7,(IX+**index**)	DDCB**index**BE	Bit 7 <- 0
---	RES	7,(IY+**index**)	FDCB**index**BE	Bit 7 <- 0
---	SET	0,A	CBC7	Bit 0 <- 1
---	SET	0,B	CBC0	Bit 0 <- 1
---	SET	0,C	CBC1	Bit 0 <- 1
---	SET	0,D	CBC2	Bit 0 <- 1
---	SET	0,E	CBC3	Bit 0 <- 1
---	SET	0,H	CBC4	Bit 0 <- 1
---	SET	0,L	CBC5	Bit 0 <- 1

---	SET	0,(HL)	CBC6	Bit 0 <- 1
---	SET	0,(IX+**index**)	DDCB**index**C6	Bit 0 <- 1
---	SET	0,(IY+**index**)	FDCB**index**C6	Bit 0 <- 1
---	SET	1,A	CBCF	Bit 1 <- 1
---	SET	1,B	CBC8	Bit 1 <- 1
---	SET	1,C	CBC9	Bit 1 <- 1
---	SET	1,D	CBCA	Bit 1 <- 1
---	SET	1,E	CBCB	Bit 1 <- 1
---	SET	1,H	CBCC	Bit 1 <- 1
---	SET	1,L	CBCD	Bit 1 <- 1
---	SET	1,(HL)	CBCE	Bit 1 <- 1
---	SET	1,(IX+**index**)	DDCB**index**CE	Bit 1 <- 1
---	SET	1,(IY+**index**)	FDCB**index**CE	Bit 1 <- 1
---	SET	2,A	CBD7	Bit 2 <- 1
---	SET	2,B	CBD0	Bit 2 <- 1
---	SET	2,C	CBD1	Bit 2 <- 1
---	SET	2,D	CBD2	Bit 2 <- 1
---	SET	2,E	CBD3	Bit 2 <- 1
---	SET	2,H	CBD4	Bit 2 <- 1
---	SET	2,L	CBD5	Bit 2 <- 1
---	SET	2,(HL)	CBD6	Bit 2 <- 1
---	SET	2,(IX+**index**)	DDCB**index**D6	Bit 2 <- 1
---	SET	2,(IY+**index**)	FDCB**index**D6	Bit 2 <- 1
---	SET	3,A	CBDF	Bit 3 <- 1
---	SET	3,B	CBD8	Bit 3 <- 1
---	SET	3,C	CBD9	Bit 3 <- 1
---	SET	3,D	CBDA	Bit 3 <- 1
---	SET	3,E	CBDB	Bit 3 <- 1
---	SET	3,H	CBDC	Bit 3 <- 1
---	SET	3,L	CBDD	Bit 3 <- 1
---	SET	3,(HL)	CBDE	Bit 3 <- 1
---	SET	3,(IX+**index**)	DDCB**index**DE	Bit 3 <- 1
---	SET	3,(IY+**index**)	FDCB**index**DE	Bit 3 <- 1
---	SET	4,A	CBE7	Bit 4 <- 1
---	SET	4,B	CBE0	Bit 4 <- 1
---	SET	4,C	CBE1	Bit 4 <- 1
---	SET	4,D	CBE2	Bit 4 <- 1
---	SET	4,E	CBE3	Bit 4 <- 1
---	SET	4,H	CBE4	Bit 4 <- 1
---	SET	4,L	CBE5	Bit 4 <- 1
---	SET	4,(HL)	CBE6	Bit 4 <- 1
---	SET	4,(IX+**index**)	DDCB**index**E6	Bit 4 <- 1
---	SET	4,(IY+**index**)	FDCB**index**E6	Bit 4 <- 1
---	SET	5,A	CBEF	Bit 5 <- 1

---	SET	5,B	CBE8	Bit 5 <- 1
---	SET	5,C	CBE9	Bit 5 <- 1
---	SET	5,D	CBEA	Bit 5 <- 1
---	SET	5,E	CBEB	Bit 5 <- 1
---	SET	5,H	CBEC	Bit 5 <- 1
---	SET	5,L	CBED	Bit 5 <- 1
---	SET	5,(HL)	CBEE	Bit 5 <- 1
---	SET	5,(IX+**index**)	DDCB**index**EE	Bit 5 <- 1
---	SET	5,(IY+**index**)	FDCB**index**EE	Bit 5 <- 1
---	SET	6,A	CBF7	Bit 6 <- 1
---	SET	6,B	CBF0	Bit 6 <- 1
---	SET	6,C	CBF1	Bit 6 <- 1
---	SET	6,D	CBF2	Bit 6 <- 1
---	SET	6,E	CBF3	Bit 6 <- 1
---	SET	6,H	CBF4	Bit 6 <- 1
---	SET	6,L	CBF5	Bit 6 <- 1
---	SET	6,(HL)	CBF6	Bit 6 <- 1
---	SET	6,(IX+**index**)	DDCB**index**F6	Bit 6 <- 1
---	SET	6,(IY+**index**)	FDCB**index**F6	Bit 6 <- 1
---	SET	7,A	CBFF	Bit 7 <- 1
---	SET	7,B	CBF8	Bit 7 <- 1
---	SET	7,C	CBF9	Bit 7 <- 1
---	SET	7,D	CBFA	Bit 7 <- 1
---	SET	7,E	CBFB	Bit 7 <- 1
---	SET	7,H	CBFC	Bit 7 <- 1
---	SET	7,L	CBFD	Bit 7 <- 1
---	SET	7,(HL)	CBFE	Bit 7 <- 1
---	SET	7,(IX+**index**)	DDCB**index**FE	Bit 7 <- 1
---	SET	7,(IY+**index**)	FDCB**index**FE	Bit 7 <- 1

Bit Shift Instructions (Z80 Only)

8080 Mnemonic	Z80 Mnemonic		Machine Code	Operation
---	SLA	A	CB27	---
---	SLA	B	CB20	---
---	SLA	C	CB21	---
---	SLA	D	CB22	---
---	SLA	E	CB23	---
---	SLA	H	CB24	---
---	SLA	L	CB25	---
---	SLA	(HL)	CB26	---
---	SLA	(IX+**index**)	DDCB**index**26	---
---	SLA	(IY+**index**)	FDCB**index**26	---
---	SRA	A	CB2F	---

---	SRA	B	CB28	---
---	SRA	C	CB29	---
---	SRA	D	CB2A	---
---	SRA	E	CB2B	---
---	SRA	H	CB2C	---
---	SRA	L	CB2D	---
---	SRA	(HL)	CB2E	---
---	SRA	(IX+**index**)	DDCB**index**2E	---
---	SRA	(IY+**index**)	FDCB**index**2E	---
---	SRL	A	CB3F	---
---	SRL	B	CB38	---
---	SRL	C	CB39	---
---	SRL	D	CB3A	---
---	SRL	E	CB3B	---
---	SRL	H	CB3C	---
---	SRL	L	CB3D	---
---	SRL	(HL)	CB3E	---
---	SRL	(IX+**index**)	DDCB**index**3E	---
---	SRL	(IY+**index**)	FDCB**index**3E	---

Related Topics

A flat-ASCII version of this 8080/Z80 Instruction Set table (ASCII)

Return to The QD Assembler Index (HTML)

Visit the nemesis.lonestar.org home page and index