Intel 8080 instruction set

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NOP 1 4 - - - - -	LXI B,d16 3 10 - - - - -	STAX B 1 7 - - - - -	INX B 1 5 - - - - -	INR B 1 5 S Z A P -	DCR B 1 5 S Z A P -	MVI B,d8 2 7 - - - - -	RLC 1 4 - - - - C	*NOP 1 4 - - - - -	DAD B 1 10 - - - - C	LDAX B 1 7 - - - - -	DCX B 1 5 - - - - -	INR C 1 5 S Z A P -	DCR C 1 5 S Z A P -	MVI C,d8 2 7 - - - - -	RRC 1 4 - - - - C
1x	*NOP 1 4 - - - - -	LXI D,d16 3 10 - - - - -	STAX D 1 7 - - - - -	INX D 1 5 - - - - -	INR D 1 5 S Z A P -	DCR D 1 5 S Z A P -	MVI D,d8 2 7 - - - - -	RAL 1 4 - - - - C	*NOP 1 4 - - - - -	DAD D 1 10 - - - - C	LDAX D 1 7 - - - - -	DCX D 1 5 - - - - -	INR E 1 5 S Z A P -	DCR E 1 5 S Z A P -	MVI E,d8 2 7 - - - - -	RAR 1 4 - - - - C
2x	*NOP 1 4 - - - - -	LXI H,d16 3 10 - - - - -	SHLD a16 3 16 - - - - -	INX H 1 5 - - - - -	INR H 1 5 S Z A P -	DCR H 1 5 S Z A P -	MVI H,d8 2 7 - - - - -	DAA 1 4 S Z A P C	*NOP 1 4 - - - - -	DAD H 1 10 - - - - C	LHLD a16 3 16 - - - - -	DCX H 1 5 - - - - -	INR L 1 5 S Z A P -	DCR L 1 5 S Z A P -	MVI L,d8 2 7 - - - - -	CMA 1 4 - - - - -
3x	*NOP 1 4 - - - - -	LXI SP,d16 3 10 - - - - -	STA a16 3 13 - - - - -	INX SP 1 5 - - - - -	INR M 1 10 S Z A P -	DCR M 1 10 S Z A P -	MVI M,d8 2 10 - - - - -	STC 1 4 - - - - C	*NOP 1 4 - - - - -	DAD SP 1 10 - - - - C	LDA a16 3 13 - - - - -	DCX SP 1 5 - - - - -	INR A 1 5 S Z A P -	DCR A 1 5 S Z A P -	MVI A,d8 2 7 - - - - -	CMC 1 4 - - - - C
4x	MOV B,B 1 5 - - - - -	MOV B,C 1 5 - - - - -	MOV B,D 1 5 - - - - -	MOV B,E 1 5 - - - - -	MOV B,H 1 5 - - - - -	MOV B,L 1 5 - - - - -	MOV B,M 1 7 - - - - -	MOV B,A 1 5 - - - - -	MOV C,B 1 5 - - - - -	MOV C,C 1 5 - - - - -	MOV C,D 1 5 - - - - -	MOV C,E 1 5 - - - - -	MOV C,H 1 5 - - - - -	MOV C,L 1 5 - - - - -	MOV C,M 1 7 - - - - -	MOV C,A 1 5 - - - - -
5x	MOV D,B 1 5 - - - - -	MOV D,C 1 5 - - - - -	MOV D,D 1 5 - - - - -	MOV D,E 1 5 - - - - -	MOV D,H 1 5 - - - - -	MOV D,L 1 5 - - - - -	MOV D,M 1 7 - - - - -	MOV D,A 1 5 - - - - -	MOV E,B 1 5 - - - - -	MOV E,C 1 5 - - - - -	MOV E,D 1 5 - - - - -	MOV E,E 1 5 - - - - -	MOV E,H 1 5 - - - - -	MOV E,L 1 5 - - - - -	MOV E,M 1 7 - - - - -	MOV E,A 1 5 - - - - -
6x	MOV H,B 1 5 - - - - -	MOV H,C 1 5 - - - - -	MOV H,D 1 5 - - - - -	MOV H,E 1 5 - - - - -	MOV H,H 1 5 - - - - -	MOV H,L 1 5 - - - - -	MOV H,M 1 7 - - - - -	MOV H,A 1 5 - - - - -	MOV L,B 1 5 - - - - -	MOV L,C 1 5 - - - - -	MOV L,D 1 5 - - - - -	MOV L,E 1 5 - - - - -	MOV L,H 1 5 - - - - -	MOV L,L 1 5 - - - - -	MOV L,M 1 7 - - - - -	MOV L,A 1 5 - - - - -
7x	MOV M,B 1 7 - - - - -	MOV M,C 1 7 - - - - -	MOV M,D 1 7 - - - - -	MOV M,E 1 7 - - - - -	MOV M,H 1 7 - - - - -	MOV M,L 1 7 - - - - -	HLT 1 7 - - - - -	MOV M,A 1 7 - - - - -	MOV A,B 1 5 - - - - -	MOV A,C 1 5 - - - - -	MOV A,D 1 5 - - - - -	MOV A,E 1 5 - - - - -	MOV A,H 1 5 - - - - -	MOV A,L 1 5 - - - - -	MOV A,M 1 7 - - - - -	MOV A,A 1 5 - - - - -
8x	ADD B 1 4 S Z A P C	ADD C 1 4 S Z A P C	ADD D 1 4 S Z A P C	ADD E 1 4 S Z A P C	ADD H 1 4 S Z A P C	ADD L 1 4 S Z A P C	ADD M 1 7 S Z A P C	ADD A 1 4 S Z A P C	ADC B 1 4 S Z A P C	ADC C 1 4 S Z A P C	ADC D 1 4 S Z A P C	ADC E 1 4 S Z A P C	ADC H 1 4 S Z A P C	ADC L 1 4 S Z A P C	ADC M 1 7 S Z A P C	ADC A 1 4 S Z A P C
9x	SUB B 1 4 S Z A P C	SUB C 1 4 S Z A P C	SUB D 1 4 S Z A P C	SUB E 1 4 S Z A P C	SUB H 1 4 S Z A P C	SUB L 1 4 S Z A P C	SUB M 1 7 S Z A P C	SUB A 1 4 S Z A P C	SBB B 1 4 S Z A P C	SBB C 1 4 S Z A P C	SBB D 1 4 S Z A P C	SBB E 1 4 S Z A P C	SBB H 1 4 S Z A P C	SBB L 1 4 S Z A P C	SBB M 1 7 S Z A P C	SBB A 1 4 S Z A P C
Ax	ANA B 1 4 S Z A P C	ANA C 1 4 S Z A P C	ANA D 1 4 S Z A P C	ANA E 1 4 S Z A P C	ANA H 1 4 S Z A P C	ANA L 1 4 S Z A P C	ANA M 1 7 S Z A P C	ANA A 1 4 S Z A P C	XRA B 1 4 S Z A P C	XRA C 1 4 S Z A P C	XRA D 1 4 S Z A P C	XRA E 1 4 S Z A P C	XRA H 1 4 S Z A P C	XRA L 1 4 S Z A P C	XRA M 1 7 S Z A P C	XRA A 1 4 S Z A P C
Bx	ORA B 1 4 S Z A P C	ORA C 1 4 S Z A P C	ORA D 1 4 S Z A P C	ORA E 1 4 S Z A P C	ORA H 1 4 S Z A P C	ORA L 1 4 S Z A P C	ORA M 1 7 S Z A P C	ORA A 1 4 S Z A P C	CMP B 1 4 S Z A P C	CMP C 1 4 S Z A P C	CMP D 1 4 S Z A P C	CMP E 1 4 S Z A P C	CMP H 1 4 S Z A P C	CMP L 1 4 S Z A P C	CMP M 1 7 S Z A P C	CMP A 1 4 S Z A P C
Cx	RNZ 1 11/5 - - - - -	POP B 1 10 - - - - -	JNZ a16 3 10 - - - - -	JMP a16 3 10 - - - - -	CNZ a16 3 17/11 - - - - -	PUSH B 1 11 - - - - -	ADI d8 2 7 S Z A P C	RST 0 1 11 - - - - -	RZ 1 11/5 - - - - -	RET 1 10 - - - - -	JZ a16 3 10 - - - - -	*JMP a16 3 10 - - - - -	CZ a16 3 17/11 - - - - -	CALL a16 3 17 - - - - -	ACI d8 2 7 S Z A P C	RST 1 1 11 - - - - -
Dx	RNC 1 11/5 - - - - -	POP D 1 10 - - - - -	JNC a16 3 10 - - - - -	OUT d8 2 10 - - - - -	CNC a16 3 17/11 - - - - -	PUSH D 1 11 - - - - -	SUI d8 2 7 S Z A P C	RST 2 1 11 - - - - -	RC 1 11/5 - - - - -	*RET 1 10 - - - - -	JC a16 3 10 - - - - -	IN d8 2 10 - - - - -	CC a16 3 17/11 - - - - -	*CALL a16 3 17 - - - - -	SBI d8 2 7 S Z A P C	RST 3 1 11 - - - - -
Ex	RPO 1 11/5 - - - - -	POP H 1 10 - - - - -	JPO a16 3 10 - - - - -	XTHL 1 18 - - - - -	CPO a16 3 17/11 - - - - -	PUSH H 1 11 - - - - -	ANI d8 2 7 S Z A P C	RST 4 1 11 - - - - -	RPE 1 11/5 - - - - -	PCHL 1 5 - - - - -	JPE a16 3 10 - - - - -	XCHG 1 5 - - - - -	CPE a16 3 17/11 - - - - -	*CALL a16 3 17 - - - - -	XRI d8 2 7 S Z A P C	RST 5 1 11 - - - - -
Fx	RP 1 11/5 - - - - -	POP PSW 1 10 S Z A P C	JP a16 3 10 - - - - -	DI 1 4 - - - - -	CP a16 3 17/11 - - - - -	PUSH PSW 1 11 - - - - -	ORI d8 2 7 S Z A P C	RST 6 1 11 - - - - -	RM 1 11/5 - - - - -	SPHL 1 5 - - - - -	JM a16 3 10 - - - - -	EI 1 4 - - - - -	CM a16 3 17/11 - - - - -	*CALL a16 3 17 - - - - -	CPI d8 2 7 S Z A P C	RST 7 1 11 - - - - -

Misc/control instructions
Jumps/calls
8bit load/store/move instructions
16bit load/store/move instructions
8bit arithmetic/logical instructions
16bit arithmetic/logical instructions

Length in bytes →

INS reg	← Instruction mnemonic
2 7	← Duration in cycles
S Z A P C	← Flags affected

Duration of conditional calls and returns is different when action is taken or not. This is indicated by two numbers separated by "/". The higher number (on the left side of "/") means duration of instruction when action is taken, the lower number (on the right side of "/") means duration of instruction when action is not taken.

All instructions marked by "*" are only alternative opcodes for existing instructions. Those alternative opcodes should not be used.

Registers

15 . . . 8	7 . . . 0	
A (accumulator)	F (flags)	← PSW
B	C	← B
D	E	← D
H	L	← H

15 . . . 0
SP (stack pointer)
PC (program counter)

Flag register (F) bits:

7	6	5	4	3	2	1	0
S	Z	0	A	0	P	1	C

`S` - Sign Flag
`Z` - Zero Flag
`0` - Not used, always zero
`A` - also called AC, Auxiliary Carry Flag
`0` - Not used, always zero
`P` - Parity Flag
`1` - Not used, always one
`C` - Carry Flag