Emulator 101

8080 reference 6502 reference

8080 By Opcode

I constructed this table specifically for examining raw code and to aid in writing a disassembler.

Opcode	Instruction	size	flags	function
0x00	NOP	1		
0x01	LXI B,D16	3		B <- byte 3, C <- byte 2
0x02	STAX B	1		(BC) <- A
0x03	INX B	1		BC <- BC+1
0x04	INR B	1	Z, S, P, AC	B <- B+1
0x05	DCR B	1	Z, S, P, AC	B <- B-1
0x06	MVI B, D8	2		B <- byte 2
0x07	RLC	1	CY	A = A << 1; bit 0 = prev bit 7; CY = prev bit 7
0x08	-			
0x09	DAD B	1	CY	HL = HL + BC
0x0a	LDAX B	1		A <- (BC)
0x0b	DCX B	1		BC = BC-1
0x0c	INR C	1	Z, S, P, AC	C <- C+1
0x0d	DCR C	1	Z, S, P, AC	C <-C-1
0x0e	MVI C,D8	2		C <- byte 2
0x0f	RRC	1	CY	A = A >> 1; bit 7 = prev bit 0; CY = prev bit 0
0x10	-			
0x11	LXI D,D16	3		D <- byte 3, E <- byte 2
0x12	STAX D	1		(DE) <- A
0x13	INX D	1		DE <- DE + 1
0x14	INR D	1	Z, S, P, AC	D <- D+1
0x15	DCR D	1	Z, S, P, AC	D <- D-1
0x16	MVI D, D8	2		D <- byte 2
0x17	RAL	1	CY	A = A << 1; bit 0 = prev CY; CY = prev bit 7
0x18	-			
0x19	DAD D	1	CY	HL = HL + DE
0x1a	LDAX D	1		A <- (DE)
0x1b	DCX D	1		DE = DE-1
0x1c	INR E	1	Z, S, P, AC	E <-E+1
0x1d	DCR E	1	Z, S, P, AC	E <- E-1
0x1e	MVI E,D8	2		E <- byte 2
0x1f	RAR	1	CY	A = A >> 1; bit 7 = prev bit 7; CY = prev bit 0
0x20	-			

0x21	LXI H,D16	3		H <- byte 3, L <- byte 2
0x22	SHLD adr	3		(adr) <-L; (adr+1)<-H
0x23	INX H	1		HL <- HL + 1
0x24	INR H	1	Z, S, P, AC	H <- H+1
0x25	DCR H	1	Z, S, P, AC	H <- H-1
0x26	MVI H,D8	2		H <- byte 2
0x27	DAA	1		special
0x28	-			
0x29	DAD H	1	CY	HL = HL + HI
0x2a	LHLD adr	3		L <- (adr); H<-(adr+1)
0x2b	DCX H	1		HL = HL-1
0x2c	INR L	1	Z, S, P, AC	L <- L+1
0x2d	DCR L	1	Z, S, P, AC	L <- L-1
0x2e	MVI L, D8	2		L <- byte 2
0x2f	CMA	1		A <- !A
0x30	-			
0x31	LXI SP, D16	3		SP.hi <- byte 3, SP.lo <- byte 2
0x32	STA adr	3		(adr) <- A
0x33	INX SP	1		SP = SP + 1
0x34	INR M	1	Z, S, P, AC	(HL) <- (HL)+1
0x35	DCR M	1	Z, S, P, AC	(HL) <- (HL)-1
0x36	MVI M,D8	2		(HL) <- byte 2
0x37	STC	1	CY	CY = 1
0x38	-			
0x39	DAD SP	1	CY	HL = HL + SP
0x3a	LDA adr	3		A <- (adr)
0x3b	DCX SP	1		SP = SP-1
0x3c	INR A	1	Z, S, P, AC	A <- A+1
0x3d	DCR A	1	Z, S, P, AC	A <- A-1
0x3e	MVI A,D8	2		A <- byte 2
0x3f	CMC	1	CY	CY=!CY
0x40	MOV B,B	1		B <- B
0x41	MOV B,C	1		B <- C
0x42	MOV B,D	1		B <- D
0x43	MOV B,E	1		B <- E
0x44	MOV B,H	1		B <- H
0x45	MOV B,L	1		B <- L
0x46	MOV B,M	1		B <- (HL)
0x47	MOV B,A	1		B <- A
0x48	MOV C,B	1		C <- B
0x49	MOV C,C	1		C <- C
0x4a	MOV C,D	1		C <- D
0x4b	MOV C,E	1		C <- E
0x4c	MOV C,H	1		C <- H
0x4d	MOV C,L	1		C <- L
0x4e	MOV C,M	1		C <- (HL)
0x4f	MOV C,A	1		C <- A

0x50	MOV D,B	1		D <- B
0x51	MOV D,C	1		D <- C
0x52	MOV D,D	1		D <- D
0x53	MOV D,E	1		D <- E
0x54	MOV D,H	1		D <- H
0x55	MOV D,L	1		D <- L
0x56	MOV D,M	1		D <- (HL)
0x57	MOV D,A	1		D <- A
0x58	MOV E,B	1		E <- B
0x59	MOV E,C	1		E <- C
0x5a	MOV E,D	1		E <- D
0x5b	MOV E,E	1		E <- E
0x5c	MOV E,H	1		E <- H
0x5d	MOV E,L	1		E <- L
0x5e	MOV E,M	1		E <- (HL)
0x5f	MOV E,A	1		E <- A
0x60	MOV H,B	1		H <- B
0x61	MOV H,C	1		H <- C
0x62	MOV H,D	1		H <- D
0x63	MOV H,E	1		H <- E
0x64	MOV H,H	1		H <- H
0x65	MOV H,L	1		H <- L
0x66	MOV H,M	1		H <- (HL)
0x67	MOV H,A	1		H <- A
0x68	MOV L,B	1		L <- B
0x69	MOV L,C	1		L <- C
0x6a	MOV L,D	1		L <- D
0x6b	MOV L,E	1		L <- E
0x6c	MOV L,H	1		L <- H
0x6d	MOV L,L	1		L <- L
0x6e	MOV L,M	1		L <- (HL)
0x6f	MOV L,A	1		L <- A
0x70	MOV M,B	1		(HL) <- B
0x71	MOV M,C	1		(HL) <- C
0x72	MOV M,D	1		(HL) <- D
0x73	MOV M,E	1		(HL) <- E
0x74	MOV M,H	1		(HL) <- H
0x75	MOV M,L	1		(HL) <- L
0x76	HLT	1		special
0x77	MOV M,A	1		(HL) <- A
0x78	MOV A,B	1		A <- B
0x79	MOV A,C	1		A <- C
0x7a	MOV A,D	1		A <- D
0x7b	MOV A,E	1		A <- E
0x7c	MOV A,H	1		A <- H
0x7d	MOV A,L	1		A <- L
0x7e	MOV A,M	1		A <- (HL)

0x7f	MOV A,A	1		A <- A
0x80	ADD B	1	Z, S, P, CY, AC	A <- A + B
0x81	ADD C	1	Z, S, P, CY, AC	A <- A + C
0x82	ADD D	1	Z, S, P, CY, AC	A <- A + D
0x83	ADD E	1	Z, S, P, CY, AC	A <- A + E
0x84	ADD H	1	Z, S, P, CY, AC	A <- A + H
0x85	ADD L	1	Z, S, P, CY, AC	A <- A + L
0x86	ADD M	1	Z, S, P, CY, AC	A <- A + (HL)
0x87	ADD A	1	Z, S, P, CY, AC	A <- A + A
0x88	ADC B	1	Z, S, P, CY, AC	A <- A + B + CY
0x89	ADC C	1	Z, S, P, CY, AC	A <- A + C + CY
0x8a	ADC D	1	Z, S, P, CY, AC	A <- A + D + CY
0x8b	ADC E	1	Z, S, P, CY, AC	A <- A + E + CY
0x8c	ADC H	1	Z, S, P, CY, AC	A <- A + H + CY
0x8d	ADC L	1	Z, S, P, CY, AC	A <- A + L + CY
0x8e	ADC M	1	Z, S, P, CY, AC	A <- A + (HL) + CY
0x8f	ADC A	1	Z, S, P, CY, AC	A <- A + A + CY
0x90	SUB B	1	Z, S, P, CY, AC	A <- A - B
0x91	SUB C	1	Z, S, P, CY, AC	A <- A - C
0x92	SUB D	1	Z, S, P, CY, AC	A <- A + D
0x93	SUB E	1	Z, S, P, CY, AC	A <- A - E
0x94	SUB H	1	Z, S, P, CY, AC	A <- A + H
0x95	SUB L	1	Z, S, P, CY, AC	A <- A - L
0x96	SUB M	1	Z, S, P, CY, AC	A <- A + (HL)
0x97	SUB A	1	Z, S, P, CY, AC	A <- A - A
0x98	SBB B	1	Z, S, P, CY, AC	A <- A - B - CY
0x99	SBB C	1	Z, S, P, CY, AC	A <- A - C - CY
0x9a	SBB D	1	Z, S, P, CY, AC	A <- A - D - CY
0x9b	SBB E	1	Z, S, P, CY, AC	A <- A - E - CY
0x9c	SBB H	1	Z, S, P, CY, AC	A <- A - H - CY
0x9d	SBB L	1	Z, S, P, CY, AC	A <- A - L - CY
0x9e	SBB M	1	Z, S, P, CY, AC	A <- A - (HL) - CY
0x9f	SBB A	1	Z, S, P, CY, AC	A <- A - A - CY
0xa0	ANA B	1	Z, S, P, CY, AC	A <- A & B
0xa1	ANA C	1	Z, S, P, CY, AC	A <- A & C
0xa2	ANA D	1	Z, S, P, CY, AC	A <- A & D
0xa3	ANA E	1	Z, S, P, CY, AC	A <- A & E
0xa4	ANA H	1	Z, S, P, CY, AC	A <- A & H
0xa5	ANA L	1	Z, S, P, CY, AC	A <- A & L
0xa6	ANA M	1	Z, S, P, CY, AC	A <- A & (HL)
0xa7	ANA A	1	Z, S, P, CY, AC	A <- A & A
0xa8	XRA B	1	Z, S, P, CY, AC	A <- A ^ B
0xa9	XRA C	1	Z, S, P, CY, AC	A <- A ^ C
0xaa	XRA D	1	Z, S, P, CY, AC	A <- A ^ D
0xab	XRA E	1	Z, S, P, CY, AC	A <- A ^ E
0xac	XRA H	1	Z, S, P, CY, AC	A <- A ^ H
0xad	XRA L	1	Z, S, P, CY, AC	A <- A ^ L

0xae	XRA M	1	Z, S, P, CY, AC	A <- A ^ (HL)
0xaf	XRA A	1	Z, S, P, CY, AC	A <- A ^ A
0xb0	ORA B	1	Z, S, P, CY, AC	A <- A | B
0xb1	ORA C	1	Z, S, P, CY, AC	A <- A | C
0xb2	ORA D	1	Z, S, P, CY, AC	A <- A | D
0xb3	ORA E	1	Z, S, P, CY, AC	A <- A | E
0xb4	ORA H	1	Z, S, P, CY, AC	A <- A | H
0xb5	ORA L	1	Z, S, P, CY, AC	A <- A | L
0xb6	ORA M	1	Z, S, P, CY, AC	A <- A | (HL)
0xb7	ORA A	1	Z, S, P, CY, AC	A <- A | A
0xb8	CMP B	1	Z, S, P, CY, AC	A - B
0xb9	CMP C	1	Z, S, P, CY, AC	A - C
0xba	CMP D	1	Z, S, P, CY, AC	A - D
0xbb	CMP E	1	Z, S, P, CY, AC	A - E
0xbc	CMP H	1	Z, S, P, CY, AC	A - H
0xbd	CMP L	1	Z, S, P, CY, AC	A - L
0xbe	CMP M	1	Z, S, P, CY, AC	A - (HL)
0xbf	CMP A	1	Z, S, P, CY, AC	A - A
0xc0	RNZ	1		if NZ, RET
0xc1	POP B	1		C <- (sp); B <- (sp+1); sp <- sp+2
0xc2	JNZ adr	3		if NZ, PC <- adr
0xc3	JMP adr	3		PC <= adr
0xc4	CNZ adr	3		if NZ, CALL adr
0xc5	PUSH B	1		(sp-2)<-C; (sp-1)<-B; sp <- sp - 2
0xc6	ADI D8	2	Z, S, P, CY, AC	A <- A + byte
0xc7	RST 0	1		CALL $0
0xc8	RZ	1		if Z, RET
0xc9	RET	1		PC.lo <- (sp); PC.hi<-(sp+1); SP <- SP+2
0xca	JZ adr	3		if Z, PC <- adr
0xcb	-			
0xcc	CZ adr	3		if Z, CALL adr
0xcd	CALL adr	3		(SP-1)<-PC.hi;(SP-2)<-PC.lo;SP<-SP-2;PC=adr
0xce	ACI D8	2	Z, S, P, CY, AC	A <- A + data + CY
0xcf	RST 1	1		CALL $8
0xd0	RNC	1		if NCY, RET
0xd1	POP D	1		E <- (sp); D <- (sp+1); sp <- sp+2
0xd2	JNC adr	3		if NCY, PC<-adr
0xd3	OUT D8	2		special
0xd4	CNC adr	3		if NCY, CALL adr
0xd5	PUSH D	1		(sp-2)<-E; (sp-1)<-D; sp <- sp - 2
0xd6	SUI D8	2	Z, S, P, CY, AC	A <- A - data
0xd7	RST 2	1		CALL $10
0xd8	RC	1		if CY, RET
0xd9	-			
0xda	JC adr	3		if CY, PC<-adr

0xdb	IN D8	2		special
0xdc	CC adr	3		if CY, CALL adr
0xdd	-			
0xde	SBI D8	2	Z, S, P, CY, AC	A <- A - data - CY
0xdf	RST 3	1		CALL $18
0xe0	RPO	1		if PO, RET
0xe1	POP H	1		L <- (sp); H <- (sp+1); sp <- sp+2
0xe2	JPO adr	3		if PO, PC <- adr
0xe3	XTHL	1		L <-> (SP); H <-> (SP+1)
0xe4	CPO adr	3		if PO, CALL adr
0xe5	PUSH H	1		(sp-2)<-L; (sp-1)<-H; sp <- sp - 2
0xe6	ANI D8	2	Z, S, P, CY, AC	A <- A & data
0xe7	RST 4	1		CALL $20
0xe8	RPE	1		if PE, RET
0xe9	PCHL	1		PC.hi <- H; PC.lo <- L
0xea	JPE adr	3		if PE, PC <- adr
0xeb	XCHG	1		H <-> D; L <-> E
0xec	CPE adr	3		if PE, CALL adr
0xed	-			
0xee	XRI D8	2	Z, S, P, CY, AC	A <- A ^ data
0xef	RST 5	1		CALL $28
0xf0	RP	1		if P, RET
0xf1	POP PSW	1		flags <- (sp); A <- (sp+1); sp <- sp+2
0xf2	JP adr	3		if P=1 PC <- adr
0xf3	DI	1		special
0xf4	CP adr	3		if P, PC <- adr
0xf5	PUSH PSW	1		(sp-2)<-flags; (sp-1)<-A; sp <- sp - 2
0xf6	ORI D8	2	Z, S, P, CY, AC	A <- A | data
0xf7	RST 6	1		CALL $30
0xf8	RM	1		if M, RET
0xf9	SPHL	1		SP=HL
0xfa	JM adr	3		if M, PC <- adr
0xfb	EI	1		special
0xfc	CM adr	3		if M, CALL adr
0xfd	-			
0xfe	CPI D8	2	Z, S, P, CY, AC	A - data
0xff	RST 7	1		CALL $38

Post questions or comments on Twitter @realemulator101, or if you find issues in the code, file them on the github repository.