Emulator 101

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
				A = A >> 1; bit 7 = prev bit 7; CY = prev bit