Instruction Set

hex mnemonic	B/C	hex mnemonic	B/C	hex mnemonic	B/C	hex mnemonic	B/C
00 = NOP	1/1	40 = JC code addr	2/2*				
01 = AJMP code addr	2/3	41 = AJMP code addr	2/3				
02 = LJMP code addr	3/3	42 = ORL dir, A	2/2				
03 = RR A	1/1	43 = ORL dir, #data	3/3				
04 = INC A	1/1	44 = ORL A, #data	2/2				
05 = INC dir	2/2	45 = ORL A, dir	2/2				
06 = INC @R0	1/1	46 = ORL A, @R0	1/1				
07 = INC @R1	1/1	47 = ORL A, @R1	1/1				
08 = INC R0	1/1	48 = ORL A, R0	1/1				
09 = INC R1	1/1	49 = ORL A, R1	1/1				
0A = INC R2	1/1	4A = ORL A, R2	1/1				
0B = INC R3	1/1	4B = ORL A, R3	1/1				
0C = INC R4	1/1	4C = ORL A, R4	1/1				
0D = INC R5	1/1	4D = ORL A, R5	1/1				
0E = INC R6	1/1	4E = ORL A, R6	1/1				
0F = INC R7	1/1	4F = ORL A, R7	1/1				
10 = JBC bit addr, code	3/3	50 = JNC code addr	2/2*				
11 = ACALL code addr	2/3	51 = ACALL code addr	2/3				
12 = LCALL code addr	3/3	52 = ANL dir, A	2/1				
13 = RRC A	1/1	53 = ANL dir, #data	3/3				
14 = DEC A	1/1	54 = ANL A, #data	2/2				
15 = DEC dir	2/2	55 = ANL A, dir	2/2				
16 = DEC @R0	1/1	56 = ANL A, @R0	1/1				
17 = DEC @R1	1/1	57 = ANL A, @R1	1/1				
		58 = ANL A, R0	1/1				
		59 = ANL A, R1	1/1				
		5A = ANL A, R2	1/1				
		5B = ANL A, R3	1/1				
		5C = ANL A, R4	1/1				
		5D = ANL A, R5	1/1				
		5E = ANL A, R6	1/1				
		5F = ANL A, R7	1/1				
		60 = JZ code addr	2/2*				
		61 = AJMP code addr	2/3				
		62 = XRL dir, A	2/2				
		63 = XRL dir, #data	3/3				
		64 = XRL A, #data	2/2				
		65 = XRL A, dir	2/2	A5 = n/a (reserved)	1/1		
		66 = XRL A, @R0	1/1	A6 = MOV @R0, dir	2/2		
		67 = XRL A, @R1	1/1				
		68 = XRL A, R0	1/1				
		69 = XRL A, R1	1/1				
						EA = MOV A, R2	1/1
						EB = MOV A, R3	1/1
						EC = MOV A, R4	1/1
						ED = MOV A, R5	1/1
						EE = MOV A, R6	1/1
						EF = MOV A, R7	1/1
						F0 = MOVX @DPTR, A	1/1
						F1 = ACALL code addr	2/3
						F2 = MOVX @R0, A	1/1
						F3 = MOVX @R1, A	1/1
						F4 = CPL A	1/1
						F5 = MOV dir, A	2/2
						F6 = MOV @R0, A	1/1
						F7 = MOV @R1, A	1/1
						F8 = MOV R0, A	1/1
						F9 = MOV R1, A	1/1
						FA = MOV R2, A	1/1
						FB = MOV R3, A	1/1
3C = ADDC A, R4	1/1					FC = MOV R4, A	1/1
3D = ADDC A, R5	1/1					FD = MOV R5, A	1/1
3E = ADDC A, R6	1/1					FE = MOV R6, A	1/1
3F = ADDC A, R7	1/1					FF = MOV R7, A	1/1

Figure 1: *Instruction Set of 8051*. 'B' ... number of bytes, 'C' ... number of instruction cycles.

* All conditional jump instructions take one clock cycle more if jump is taken.

e.g. JZ takes 3 clock cycles if jump is taken, 2 clock cycles if not.

** These instructions take one clock cycle more with Generic "fast_cpu" set to 0.