8051 Instruction hex code

MOVE with immediate data

Hex	Bytes	Instruction
74	2	MOV A, #immediate
75	3	MOV direct, #immediate
76	2	MOV @R0, #immediate
77	2	MOV @R1, #immediate
78	2	MOV R0, #immediate
79	2	MOV R1, #immediate
7A	2	MOV R2, #immediate
7B	2	MOV R3, #immediate
7C	2	MOV R4, #immediate
7D	2	MOV R5, #immediate
7E	2	MOV R6, #immediate
7F	2	MOV R7, #immediate
90	3	MOV DPTR, #immediate

MOVE with direct memory location

85	3	MOV direct, direct
88	2	MOV direct, R0
89	2	MOV direct, R1
8A	2	MOV direct, R2
8B	2	MOV direct, R3
8C	2	MOV direct, R4
8D	2	MOV direct, R5
8E	2	MOV direct, R6
8F	2	MOV direct, R7
A8	2	MOV R0, direct
A9	2	MOV R1, direct
AA	2	MOV R2, direct
AB	2	MOV R3, direct
AC	2	MOV R4, direct
AD	2	MOV R5, direct
AE	2	MOV R6, direct
AF	2	MOV R7, direct

MOVE with Accumulator

E5	2	MOV A, direct
E6	1	MOV A, @R0
E7	1	MOV A, @R1
E8	1	MOV A, R0
E9	1	MOV A, R1
EA	1	MOV A, R2
EB	1	MOV A, R3
EC	1	MOV A, R4
ED	1	MOV A, R5
EE	1	MOV A, R6
EF	1	MOV A, R7
F5	2	MOV direct, A
F6	1	MOV @R0, A
F7	1	MOV @R1, A
F8	1	MOV R0, A
F9	1	MOV R1, A
FA	1	MOV R2, A
FB	1	MOV R3, A
FC	1	MOV R4, A
FD	1	MOV R5, A
FE	1	MOV R6, A
FF	1	MOV R7, A

MOVE with external memory

E0	1	MOVX A, @DPTR
E2	1	MOVX A, @R0
E3	1	MOVX A, @R1
F0	1	MOVX @DPTR, A
F2	1	MOVX @R0, A
F3	1	MOVX @R1, A

MOVE with code memory

83	1	MOVC A, @A+PC
93	1	MOVC A, @A+DPTR

Bit MOVE

92	2	MOV bit, C
A2	2	MOV C, bit

Indirect MOVE with R0 and R1

86	2	MOV direct, @R0
87	2	MOV direct, @R1
A6	2	MOV @R0, direct
A7	2	MOV @R1, direct

ADD with Accumulator

24	2	ADD A, #immediate
25	2	ADD A, direct
26	1	ADD A, @R0
27	1	ADD A, @R1
28	1	ADD A, R0
29	1	ADD A, R1
2A	1	ADD A, R2
2B	1	ADD A, R3
2C	1	ADD A, R4
2D	1	ADD A, R5
2E	1	ADD A, R6
2F	1	ADD A, R7

ADD with Accumulator & carry flag

34	2	ADDC A, #immediate
35	2	ADDC A, direct
36	1	ADDC A, @R0
37	1	ADDC A, @R1
38	1	ADDC A, R0
39	1	ADDC A, R1

3A	1	ADDC	A, R2
3B	1	ADDC	A, R3
3C	1	ADDC	A, R4
3D	1	ADDC	A, R5
3E	1	ADDC	A, R6
3F	1	ADDC	A, R7

Subtract with borrow

94	2	SUBB	A, #immediate
95	2	SUBB	A, direct
96	1	SUBB	A, @R0
97	1	SUBB	A, @R1
98	1	SUBB	A, R0
99	1	SUBB	A, R1
9A	1	SUBB	A, R2
9B	1	SUBB	A, R3
9C	1	SUBB	A, R4
9D	1	SUBB	A, R5
9E	1	SUBB	A, R6
9F	1	SUBB	A, R7

INC and DEC

04	1	INC	A
05	2	INC	direct
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
A3	1	INC	DPTR
14	1	DEC	A
15	2	DEC	direct
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7

COMPARE with JUMP

B4	3	CJNE	A, #immediate, offset
B5	3	CJNE	A, direct, offset
B6	3	CJNE	@R0, #immediate, offset
B7	3	CJNE	@R1, #immediate, offset
B8	3	CJNE	R0, #immediate, offset
B9	3	CJNE	R1, #immediate, offset
BA	3	CJNE	R2, #immediate, offset
BB	3	CJNE	R3, #immediate, offset
BC	3	CJNE	R4, #immediate, offset
BD	3	CJNE	R5, #immediate, offset
BE	3	CJNE	R6, #immediate, offset
BF	3	CJNE	R7, #immediate, offset

D8	2	DJNZ	R0, offset
D9	2	DJNZ	R1, offset
DA	2	DJNZ	R2, offset
DB	2	DJNZ	R3, offset
DC	2	DJNZ	R4, offset
DD	2	DJNZ	R5, offset
DE	2	DJNZ	R6, offset
DF	2	DJNZ	R7, offset
D5	3	DJNZ	direct, offset

JUMP

01	2	AJMP	addr11
21	2	AJMP	addr11
41	2	AJMP	addr11
61	2	AJMP	addr11
81	2	AJMP	addr11
A1	2	AJMP	addr11
C1	2	AJMP	addr11
E1	2	AJMP	addr11
02	3	LJMP	addr16
80	2	SJMP	offset

JUMP with flag

10	3	JBC	bit, offset
20	3	JB	bit, offset
30	3	JNB	bit, offset
40	2	JC	offset
50	2	JNC	offset
60	2	JZ	offset
70	2	JNZ	offset

JUMP indirect

73	1	JMP	@A+DPTR

CALL subroutine and Return

11	2	ACALL	addr11
31	2	ACALL	addr11
51	2	ACALL	addr11
71	2	ACALL	addr11
91	2	ACALL	addr11
B1	2	ACALL	addr11

D1	2	ACALL	addr11
F1	2	ACALL	addr11
12	3	LCALL	addr16
22	1	RET	
32	1	RETI	

ROTATE

03	1	RR	A
13	1	RRC	A
23	1	RL	A
33	1	RLC	A
C4	1	SWAP	A

LOGICAL

52	2	ANL	direct, A
53	3	ANL	direct, #immediate
54	2	ANL	A, #immediate
55	2	ANL	A, direct
56	1	ANL	A, @R0
57	1	ANL	A, @R1
58	1	ANL	A, R0
59	1	ANL	A, R1
5A	1	ANL	A, R2
5B	1	ANL	A, R3
5C	1	ANL	A, R4
5D	1	ANL	A, R5
5E	1	ANL	A, R6
5F	1	ANL	A, R7
42	2	ORL	direct, A
43	3	ORL	direct, #immediate
44	2	ORL	A, #immediate
45	2	ORL	A, direct
46	1	ORL	A, @R0
47	1	ORL	A, @R1
48	1	ORL	A, R0
49	1	ORL	A, R1
4A	1	ORL	A, R2
4B	1	ORL	A, R3
4C	1	ORL	A, R4
4D	1	ORL	A, R5
4E	1	ORL	A, R6
4F	1	ORL	A, R7
62	2	XRL	direct, A
63	3	XRL	direct, #immediate
64	2	XRL	A, #immediate
65	2	XRL	A, direct
66	1	XRL	A, @R0
67	1	XRL	A, @R1
68	1	XRL	A, R0
69	1	XRL	A, R1
6A	1	XRL	A, R2
6B	1	XRL	A, R3
6C	1	XRL	A, R4
6D	1	XRL	A, R5
6E	1	XRL	A, R6
6F	1	XRL	A, R7

BIT logical

72	2	ORL	C, bit
82	2	ANL	C, bit
A0	2	ORL	C, /bit
B0	2	ANL	C, /bit
B2	2	CPL	bit
B3	1	CPL	C
C2	2	CLR	bit
C3	1	CLR	C
D2	2	SETB	bit
D3	1	SETB	C

Exchange

C5	2	XCH	A, direct
C6	1	XCH	A, @R0
C7	1	XCH	A, @R1
C8	1	XCH	A, R0
C9	1	XCH	A, R1
CA	1	XCH	A, R2
CB	1	XCH	A, R3
CC	1	XCH	A, R4
CD	1	XCH	A, R5
CE	1	XCH	A, R6
CF	1	XCH	A, R71
D6	1	XCHD	A, @R0
D7	1	XCHD	A, @R1

PUSH & POP

C0	2	PUSH	direct
D0	2	POP	direct

Accumulator

A4	1	MUL	AB
84	1	DIV	AB
D4	1	DA	A
E4	1	CLR	A
F4	1	CPL	A

No operation

00	1	NOP

8051 Instruction hex code

Hex	Bytes	Instruction
00	1	NOP
01	2	AJMP addr11
02	3	LJMP addr16
03	1	RR A
04	1	INC A
05	2	INC direct
06	1	INC @R0
07	1	INC @R1
08	1	INC R0
09	1	INC R1
0A	1	INC R2
0B	1	INC R3
0C	1	INC R4
0D	1	INC R5
0E	1	INC R6
0F	1	INC R7
10	3	JBC bit, offset
11	2	ACALL addr11
12	3	LCALL addr16
13	1	RRC A
14	1	DEC A
15	2	DEC direct
16	1	DEC @R0
17	1	DEC @R1
18	1	DEC R0
19	1	DEC R1
1A	1	DEC R2
1B	1	DEC R3
1C	1	DEC R4
1D	1	DEC R5
1E	1	DEC R6
1F	1	DEC R7
20	3	JB bit, offset
21	2	AJMP addr11
22	1	RET
23	1	RL A
24	2	ADD A, #immediate
25	2	ADD A, direct
26	1	ADD A, @R0
27	1	ADD A, @R1
28	1	ADD A, R0
29	1	ADD A, R1
2A	1	ADD A, R2
2B	1	ADD A, R3
2C	1	ADD A, R4
2D	1	ADD A, R5
2E	1	ADD A, R6
2F	1	ADD A, R7
30	3	JNB bit, offset
31	2	ACALL addr11
32	1	RETI
33	1	RLC A
34	2	ADDC A, #immediate
35	2	ADDC A, direct
36	1	ADDC A, @R0
37	1	ADDC A, @R1
38	1	ADDC A, R0
39	1	ADDC A, R1
3A	1	ADDC A, R2
3B	1	ADDC A, R3
3C	1	ADDC A, R4
3D	1	ADDC A, R5
3E	1	ADDC A, R6
3F	1	ADDC A, R7
40	2	JC offset
41	2	AJMP addr11
42	2	ORL direct, A
43	3	ORL direct, #immediate
44	2	ORL A, #immediate
45	2	ORL A, direct
4 6	1	ORL A, @R0
47	1	ORL A, @R1
48	1	ORL A, R0
49	1	ORL A, R1
4A	1	ORL A, R2
4B	1	ORL A, R3
4C	1	ORL A, R4
4D	1	ORL A, R5
4E	1	ORL A, R6
4F	1	ORL A, R7
50	2	JNC offset
51	2	ACALL addr11
52	2	ANL direct, A
53	3	ANL direct, #immediate
54	2	ANL A, #immediate
55	2	ANL A, direct
56	1	ANL A, @R0
57	1	ANL A, @R1
58	1	ANL A, R0
59	1	ANL A, R1
5A	1	ANL A, R2
5B	1	ANL A, R3
5C	1	ANL A, R4
5D	1	ANL A, R5
5E	1	ANL A, R6
5F	1	ANL A, R7
60	2	JZ offset
61	2	AJMP addr11
62	2	XRL direct, A
63	3	XRL direct, #immediate
64	2	XRL A, #immediate
65	2	XRL A, direct
66	1	XRL A, @R0
67	1	XRL A, @R1
68	1	XRL A, R0
69	1	XRL A, R1
6A	1	XRL A, R2
6B	1	XRL A, R3
6C	1	XRL A, R4
6D	1	XRL A, R5

6E	1	XRL A, R6
6F	1	XRL A, R7
70	2	JNZ offset
71	2	ACALL addr11
72	2	ORL C, bit
73	1	JMP @A+DPTR
74	2	MOV A, #immediate
75	3	MOV direct, #immediate
76	2	MOV @R0, #immediate
77	2	MOV @R1, #immediate
78	2	MOV R0, #immediate
79	2	MOV R1, #immediate
7A	2	MOV R2, #immediate
7B	2	MOV R3, #immediate
7C	2	MOV R4, #immediate
7D	2	MOV R5, #immediate
7E	2	MOV R6, #immediate
7F	2	MOV R7, #immediate
80	2	SJMP offset
81	2	AJMP addr11
82	2	ANL C, bit
83	1	MOVC A, @A+PC
84	1	DIV AB
85	3	MOV direct, direct
86	2	MOV direct, @R0
87	2	MOV direct, @R1
88	2	MOV direct, R0
89	2	MOV direct, R1
8A	2	MOV direct, R2
8B	2	MOV direct, R3
8C	2	MOV direct, R4
8D	2	MOV direct, R5
8E	2	MOV direct, R6
8F	2	MOV direct, R7
90	3	MOV DPTR, #immediate
91	2	ACALL addr11
92	2	MOV bit, C
93	1	MOVC A, @A+DPTR
94	2	SUBB A, #immediate
95	2	SUBB A, direct
96	1	SUBB A, @R0
97	1	SUBB A, @R1
98	1	SUBB A, R0
99	1	SUBB A, R1
9A	1	SUBB A, R2
9B	1	SUBB A, R3
9C	1	SUBB A, R4
9D	1	SUBB A, R5
9E	1	SUBB A, R6
9F	1	SUBB A, R7
A0	2	ORL C, /bit
A1	2	AJMP addr11
A2	2	MOV C, bit
A3	1	INC DPTR
A4	1	MUL AB
A5		undefined
A6	2	MOV @R0, direct
A7	2	MOV @R1, direct
A8	2	MOV R0, direct
A9	2	MOV R1, direct
AA	2	MOV R2, direct
AB	2	MOV R3, direct
AC	2	MOV R4, direct
AD	2	MOV R5, direct
AE	2	MOV R6, direct
AF	2	MOV R7, direct
B0	2	ANL C, /bit
B1	2	ACALL addr11
B2	2	CPL bit
B3	1	CPL C
B4	3	CJNE A, #immediate, offset
B5	3	CJNE A, direct, offset
B6	3	CJNE @R0, #immediate, offset
B7	3	CJNE @R1, #immediate, offset
B8	3	CJNE R0, #immediate, offset
B9	3	CJNE R1, #immediate, offset
BA	3	CJNE R2, #immediate, offset
BB	3	CJNE R3, #immediate, offset
BC	3	CJNE R4, #immediate, offset
BD	3	CJNE R5, #immediate, offset
BE	3	CJNE R6, #immediate, offset
BF	3	CJNE R7, #immediate, offset
C0	2	PUSH direct
C1	2	AJMP addr11
C2	2	CLR bit
C3	1	CLR C
C4	1	SWAP A
C5	2	XCH A, direct
C6	1	XCH A, @R0
C7	1	XCH A, @R1
C8	1	XCH A, R0
C9	1	XCH A, R1
CA	1	XCH A, R2
CB	1	XCH A, R3
CC	1	XCH A, R4
CD	1	XCH A, R5
CE	1	XCH A, R6
CF	1	XCH A, R7
D0	2	POP direct
D1	2	ACALL addr11
D2	2	SETB bit
D3	1	SETB C
D4	1	DA A
D5	3	DJNZ direct, offset
D6	1	XCHD A, @R0
D7	1	XCHD A, @R1
D8	2	DJNZ R0, offset
D9	2	DJNZ R1, offset
DA	2	DJNZ R2, offset
DB	2	DJNZ R3, offset
DC	2	DJNZ R4, offset
DD	2	DJNZ R5, offset
DE	2	DJNZ R6, offset

DF	2	DJNZ R7, offset
E0	1	MOVX A, @DPTR
E1	2	AJMP addr11
E2	1	MOVX A, @R0
E3	1	MOVX A, @R1
E4	1	CLR A
E5	2	MOV A, direct
E6	1	MOV A, @R0
E7	1	MOV A, @R1
E8	1	MOV A, R0
E9	1	MOV A, R1
EA	1	MOV A, R2
EB	1	MOV A, R3
EC	1	MOV A, R4
ED	1	MOV A, R5
EE	1	MOV A, R6
EF	1	MOV A, R7
F0	1	MOVX @DPTR, A
F1	2	ACALL addr11
F2	1	MOVX @R0, A
F3	1	MOVX @R1, A
F4	1	CPL A
F5	2	MOV direct, A
F6	1	MOV @R0, A
F7	1	MOV @R1, A
F8	1	MOV R0, A
F9	1	MOV R1, A
FA	1	MOV R2, A
FB	1	MOV R3, A
FC	1	MOV R4, A
FD	1	MOV R5, A
FE	1	MOV R6, A
FF	1	MOV R7, A

NOTES

1. Direct address is RAM location 00-7F and SFR 80-FF

Hex code	Instruction
75 30 55	MOV 30H, #55H

75 90 1F MOV 90H,#1FH

90H is PORT1, Special Function Register (SFR).

2. Indirect address is RAM location 00-FF

78 90	MOV R0,#90H
76 20	MOV @R0,#20H

90H is not PORT1. It is general RAM space in the upper page 80-FF. The upper page must be accessed with indirect addressing

3. Immediate data for DPTR will be 16-bit

90 01 00	MOV DPTR,#0100H

4. Bit is location of bit address.

A2 B2	MOV C, P3.2

5. OFFSET is the byte distant between current Program counter and the destination.

Finding the OFFSET byte can be done with ALT, OFFSET press.

6. Addr11 is 11 bits of destination address.

Finding the hex code for AJMP or ACALL can be done with ALT, AJMP or ACALL press.

7. Long CALL or Long JMP uses 16-bit address.

02 91 00	LJMP 9100H
12 00 0B	LCALL 000B