8051 Instructions in numerical Order

Abbreviations:

direct	=	8-bit DATA address in internal memory
const8	=	8-bit constant in CODE memory
const16	=	16-bit constant in CODE memory
addr16	=	16-bit long CODE address
addr11	=	11-bit absolute CODE address
rel	=	signed 8-bit relative CODE address
bit	=	8-bit BIT address in internal memory

Opcode	Mnemonic	Operands	Bytes	Flags	Cycles
00	NOP		1		1
01	AJMP	addr11	2		2
02	LJMP	addr16	3		2
03	RR	A	1		1
04	INC	A	1	P	1
05	INC	direct	2		1
06	INC	@R0	1		1
07	INC	@R1	1		1
08	INC	R0	1		1
09	INC	R1	1		1
0A	INC	R2	1		1
0B	INC	R3	1		1
0C	INC	R4	1		1
0D	INC	R5	1		1
0E	INC	R6	1		1
0F	INC	R7	1		1
10	JBC	bit, rel	3		2
11	ACALL	addr11	2		2
12	LCALL	addr16	3		2
13	RRC	A	1	CY P	1
14	DEC	A	1	P	1
15	DEC	direct	2		1
16	DEC	@R0	1		1
17	DEC	@R1	1		1
18	DEC	R0	1		1
19	DEC	R1	1		1
1A	DEC	R2	1		1
1B	DEC	R3	1		1
1C	DEC	R4	1		1
1D	DEC	R5	1		1
1E	DEC	R6	1		1
1F	DEC	R7	1		1
20	JB	bit, rel	3		2
21	AJMP	addr11	2		2
22	RET		1		2
23	RL	A	1		1
24	ADD	A, #const8	2	CY AC OV P	1
25	ADD	A, direct	2	CY AC OV P	1
26	ADD	A, @R0	1	CY AC OV P	1
27	ADD	A, @R1	1	CY AC OV P	1

| Opcode | Mnemonic | Operands | Bytes | Flags | Cycles |

28	ADD	A, R0	1	CY	AC	OV	P	1
29	ADD	A, R1	1	CY	AC	OV	P	1
2A	ADD	A, R2	1	CY	AC	OV	P	1
2B	ADD	A, R3	1	CY	AC	OV	P	1
2C	ADD	A, R4	1	CY	AC	OV	P	1
2D	ADD	A, R5	1	CY	AC	OV	P	1
2E	ADD	A, R6	1	CY	AC	OV	P	1
2F	ADD	A, R7	1	CY	AC	OV	P	1
30	JNB	bit, rel	3					2
31	ACALL	addr11	2					2
32	RETI		1					2
33	RLC	A	1	CY			P	1
34	ADDC	A, #const8	2	CY	AC	OV	P	1
35	ADDC	A, direct	2	CY	AC	OV	P	1
36	ADDC	A, @R0	1	CY	AC	OV	P	1
37	ADDC	A, @R1	1	CY	AC	OV	P	1
38	ADDC	A, R0	1	CY	AC	OV	P	1
39	ADDC	A, R1	1	CY	AC	OV	P	1
3A	ADDC	A, R2	1	CY	AC	OV	P	1
3B	ADDC	A, R3	1	CY	AC	OV	P	1
3C	ADDC	A, R4	1	CY	AC	OV	P	1
3D	ADDC	A, R5	1	CY	AC	OV	P	1
3E	ADDC	A, R6	1	CY	AC	OV	P	1
3F	ADDC	A, R7	1	CY	AC	OV	P	1
40	JC	rel	2					2
41	AJMP	addr11	2					2
42	ORL	direct, A	2					1
43	ORL	direct, #const8	3					2
44	ORL	A, #const8	2				P	1
45	ORL	A, direct	2				P	1
46	ORL	A, @R0	1				P	1
47	ORL	A, @R1	1				P	1
48	ORL	A, R0	1				P	1
49	ORL	A, R1	1				P	1
4A	ORL	A, R2	1				P	1
4B	ORL	A, R3	1				P	1
4C	ORL	A, R4	1				P	1
4D	ORL	A, R5	1				P	1
4E	ORL	A, R6	1				P	1
4F	ORL	A, R7	1				P	1
50	JNC	rel	2					2
51	ACALL	addr11	2					2
52	ANL	direct, A	2					1
53	ANL	direct, #const8	3					2
54	ANL	A, #const8	2				P	1
55	ANL	A, direct	2				P	1
56	ANL	A, @R0	1				P	1
57	ANL	A, @R1	1				P	1
58	ANL	A, R0	1				P	1
59	ANL	A, R1	1				P	1
5A	ANL	A, R2	1				P	1
5B	ANL	A, R3	1				P	1
5C	ANL	A, R4	1				P	1
5D	ANL	A, R5	1				P	1
5E	ANL	A, R6	1				P	1
5F	ANL	A, R7	1				P	1

Opcode	Mnemonic	Operands	Bytes	Flags	Cycles
60	JZ	rel	2		2
61	AJMP	addr11	2		2
62	XRL	direct, A	2		1
63	XRL	direct, #const8	3		2
64	XRL	A, #const8	2	P	1
65	XRL	A, direct	2	P	1
66	XRL	A, @R0	1	P	1
67	XRL	A, @R1	1	P	1
68	XRL	A, R0	1	P	1
69	XRL	A, R1	1	P	1
6A	XRL	A, R2	1	P	1
6B	XRL	A, R3	1	P	1
6C	XRL	A, R4	1	P	1
6D	XRL	A, R5	1	P	1
6E	XRL	A, R6	1	P	1
6F	XRL	A, R7	1	P	1
70	JNZ	rel	2		2
71	ACALL	addr11	2		2
72	ORL	C, bit	2	CY	2
73	JMP	@A+DPTR	1		2
74	MOV	A, #const8	2	P	1
75	MOV	direct, #const8	3		2
76	MOV	@R0, #const8	2		1
77	MOV	@R1, #const8	2		1
78	MOV	R0, #const8	2		1
79	MOV	R1, #const8	2		1
7A	MOV	R2, #const8	2		1
7B	MOV	R3, #const8	2		1
7C	MOV	R4, #const8	2		1
7D	MOV	R5, #const8	2		1
7E	MOV	R6, #const8	2		1
7F	MOV	R7, #const8	2		1
80	SJMP	rel	2		2
81	AJMP	addr11	2		2
82	ANL	C, bit	2	CY	2
83	MOVC	A, @A+PC	1	P	2
84	DIV	AB	1	CY OV P	4
85	MOV	direct, direct	3		2
86	MOV	direct, @R0	2		2
87	MOV	direct, @R1	2		2
88	MOV	direct, R0	2		2
89	MOV	direct, R1	2		2
8A	MOV	direct, R2	2		2
8B	MOV	direct, R3	2		2
8C	MOV	direct, R4	2		2
8D	MOV	direct, R5	2		2
8E	MOV	direct, R6	2		2
8F	MOV	direct, R7	2		2
90	MOV	DPTR, #const16	3		2
91	ACALL	addr11	2		2
92	MOV	bit, C	2		2
93	MOVC	A, @A+DPTR	1	P	2
94	SUBB	A, #const8	2	CY AC OV P	1

95	SUBB	A, direct	2	CY	AC	OV	P	1
96	SUBB	A, @R0	1	CY	AC	OV	P	1
97	SUBB	A, @R1	1	CY	AC	OV	P	1

Opcode	Mnemonic	Operands	Bytes	Flags				Cycles
98	SUBB	A, R0	1	CY	AC	OV	P	1
99	SUBB	A, R1	1	CY	AC	OV	P	1
9A	SUBB	A, R2	1	CY	AC	OV	P	1
9B	SUBB	A, R3	1	CY	AC	OV	P	1
9C	SUBB	A, R4	1	CY	AC	OV	P	1
9D	SUBB	A, R5	1	CY	AC	OV	P	1
9E	SUBB	A, R6	1	CY	AC	OV	P	1
9F	SUBB	A, R7	1	CY	AC	OV	P	1
A0	ORL	C, /bit	2	CY				2
A1	AJMP	addr11	2					2
A2	MOV	C, bit	2	CY				1
A3	INC	DPTR	1					2
A4	MUL	AB	1	CY		OV	P	4
A5	illegal	opcode						
A6	MOV	@R0, direct	2					2
A7	MOV	@R1, direct	2					2
A8	MOV	R0, direct	2					2
A9	MOV	R1, direct	2					2
AA	MOV	R2, direct	2					2
AB	MOV	R3, direct	2					2
AC	MOV	R4, direct	2					2
AD	MOV	R5, direct	2					2
AE	MOV	R6, direct	2					2
AF	MOV	R7, direct	2					2
B0	ANL	C, /bit	2	CY				2
B1	ACALL	addr11	2					2
B2	CPL	bit	2					1
B3	CPL	C	1	CY				1
B4	CJNE	A, #const8, rel	3	CY				2
B5	CJNE	A, direct, rel	3	CY				2
B6	CJNE	@R0, #const8, rel	3	CY				2
B7	CJNE	@R1, #const8, rel	3	CY				2
B8	CJNE	R0, #const8, rel	3	CY				2
B9	CJNE	R1, #const8, rel	3	CY				2
BA	CJNE	R2, #const8, rel	3	CY				2
BB	CJNE	R3, #const8, rel	3	CY				2
BC	CJNE	R4, #const8, rel	3	CY				2
BD	CJNE	R5, #const8, rel	3	CY				2
BE	CJNE	R6, #const8, rel	3	CY				2
BF	CJNE	R7, #const8, rel	3	CY				2
C0	PUSH	direct	2					2
C1	AJMP	addr11	2					2
C2	CLR	bit	2					1
C3	CLR	C	1	CY				1
C4	SWAP	A	1					1
C5	XCH	A, direct	2				P	1
C6	XCH	A, @R0	1				P	1
C7	XCH	A, @R1	1				P	1
C8	XCH	A, R0	1				P	1
C9	XCH	A, R1	1				P	1

CA	XCH	A, R2	1	P	1
CB	XCH	A, R3	1	P	1
CC	XCH	A, R4	1	P	1
CD	XCH	A, R5	1	P	1
CE	XCH	A, R6	1	P	1
CF	XCH	A, R7	1	P	1

Opcode	Mnemonic	Operands	Bytes	Flags	Cycles
D0	POP	direct	2		2
D1	ACALL	addr11	2		2
D2	SETB	bit	2		1
D3	SETB	C	1	CY	1
D4	DA	A	1	CY P	1
D5	DJNZ	direct, rel	3		2
D6	XCHD	A, @R0	1	P	1
D7	XCHD	A, @R1	1	P	1
D8	DJNZ	R0, rel	2		2
D9	DJNZ	R1, rel	2		2
DA	DJNZ	R2, rel	2		2
DB	DJNZ	R3, rel	2		2
DC	DJNZ	R4, rel	2		2
DD	DJNZ	R5, rel	2		2
DE	DJNZ	R6, rel	2		2
DF	DJNZ	R7, rel	2		2
E0	MOVX	A, @DPTR	1	P	2
E1	AJMP	addr11	2		2
E2	MOVX	A, @R0	1	P	2
E3	MOVX	A, @R1	1	P	2
E4	CLR	A	1	P	1
E5	MOV	A, direct	2	P	1
E6	MOV	A, @R0	1	P	1
E7	MOV	A, @R1	1	P	1
E8	MOV	A, R0	1	P	1
E9	MOV	A, R1	1	P	1
EA	MOV	A, R2	1	P	1
EB	MOV	A, R3	1	P	1
EC	MOV	A, R4	1	P	1
ED	MOV	A, R5	1	P	1
EE	MOV	A, R6	1	P	1
EF	MOV	A, R7	1	P	1
F0	MOVX	@DPTR, A	1		2
F1	ACALL	addr11	2		2
F2	MOVX	@R0, A	1		2
F3	MOVX	@R1, A	1		2
F4	CPL	A	1	P	1
F5	MOV	direct, A	2		1
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
FC	MOV	R4, A	1		1
FD	MOV	R5, A	1		1
FE	MOV	R6, A	1		1

FF	MOV	R7, A	1		1

8051 Instructions in lexical Order

Abbreviations:

direct	=	8-bit DATA address in internal memory
const8	=	8-bit constant in CODE memory
const16	=	16-bit constant in CODE memory
addr16	=	16-bit long CODE address
addr11	=	11-bit absolute CODE address
rel	=	signed 8-bit relative CODE address
bit	=	8-bit BIT address in internal memory

i = register numbers 0 or 1
n = register numbers 0 thru 7
a = 32 * m
m = the 3 most significant bits of an absolute address

Opcode	**Mnemonic**	**Operands**	**Bytes**	**Flags**	**Cycles**
11+a	ACALL	addr11	2		2
24	ADD	A, #const8	2	CY AC OV P	1
26+i	ADD	A, @Ri	1	CY AC OV P	1
25	ADD	A, direct	2	CY AC OV P	1
28+n	ADD	A, Rn	1	CY AC OV P	1
34	ADDC	A, #const8	2	CY AC OV P	1
36+i	ADDC	A, @Ri	1	CY AC OV P	1
35	ADDC	A, direct	2	CY AC OV P	1
38+n	ADDC	A, Rn	1	CY AC OV P	1
01+a	AJMP	addr11	2		2
54	ANL	A, #const8	2	P	1
56+i	ANL	A, @Ri	1	P	1
55	ANL	A, direct	2	P	1
58+n	ANL	A, Rn	1	P	1
B0	ANL	C, /bit	2	CY	2
82	ANL	C, bit	2	CY	2
53	ANL	direct, #const8	3		2
52	ANL	direct, A	2		1
B6+i	CJNE	@Ri, #const8, rel	3	CY	2
B4	CJNE	A, #const8, rel	3	CY	2
B5	CJNE	A, direct, rel	3	CY	2
B8+n	CJNE	Rn, #const8, rel	3	CY	2
E4	CLR	A	1	P	1
C2	CLR	bit	2		1
C3	CLR	C	1	CY	1
F4	CPL	A	1	P	1
B2	CPL	bit	2		1
B3	CPL	C	1	CY	1
D4	DA	A	1	CY P	1
16+i	DEC	@Ri	1		1
14	DEC	A	1	P	1

15	DEC	direct	2				1
18+n	DEC	Rn	1				1
84	DIV	AB	1	CY	OV	P	4
D5	DJNZ	direct, rel	3				2

Opcode	Mnemonic	Operands	Bytes	Flags			Cycles
D8+n	DJNZ	Rn, rel	2				2
06+i	INC	@Ri	1				1
04	INC	A	1			P	1
05	INC	direct	2				1
A3	INC	DPTR	1				2
08+n	INC	Rn	1				1
20	JB	bit, rel	3				2
10	JBC	bit, rel	3				2
40	JC	rel	2				2
73	JMP	@A+DPTR	1				2
30	JNB	bit, rel	3				2
50	JNC	rel	2				2
70	JNZ	rel	2				2
60	JZ	rel	2				2
12	LCALL	addr16	3				2
02	LJMP	addr16	3				2
76+i	MOV	@Ri, #const8	2				1
F6+i	MOV	@Ri, A	1				1
A6+i	MOV	@Ri, direct	2				2
74	MOV	A, #const8	2			P	1
E6+i	MOV	A, @Ri	1			P	1
E5	MOV	A, direct	2			P	1
E8+n	MOV	A, Rn	1			P	1
92	MOV	bit, C	2				2
A2	MOV	C, bit	2	CY			1
75	MOV	direct, #const8	3				2
86+i	MOV	direct, @Ri	2				2
F5	MOV	direct, A	2				1
85	MOV	direct, direct	3				2
88+n	MOV	direct, Rn	2				2
90	MOV	DPTR, #const16	3				2
78+n	MOV	Rn, #const8	2				1
F8+n	MOV	Rn, A	1				1
A8+n	MOV	Rn, direct	2				2
93	MOVC	A, @A+DPTR	1			P	2
83	MOVC	A, @A+PC	1			P	2
F0	MOVX	@DPTR, A	1				2
F2+i	MOVX	@Ri, A	1				2
E0	MOVX	A, @DPTR	1			P	2
E2+i	MOVX	A, @Ri	1			P	2
A4	MUL	AB	1	CY	OV	P	4
00	NOP		1				1
44	ORL	A, #const8	2			P	1
46+i	ORL	A, @Ri	1			P	1
45	ORL	A, direct	2			P	1
48+n	ORL	A, Rn	1			P	1
A0	ORL	C, /bit	2	CY			2
72	ORL	C, bit	2	CY			2
43	ORL	direct, #const8	3				2

42	ORL	direct, A	2			1
D0	POP	direct	2			2
C0	PUSH	direct	2			2
22	RET		1			2
32	RETI		1			2
23	RL	A	1			1
33	RLC	A	1	CY	P	1

Opcode	**Mnemonic**	**Operands**	**Bytes**	**Flags**		**Cycles**
03	RR	A	1			1
13	RRC	A	1	CY	P	1
D2	SETB	bit	2			1
D3	SETB	C	1	CY		1
80	SJMP	rel	2			2
94	SUBB	A, #const8	2	CY AC OV	P	1
96+i	SUBB	A, @Ri	1	CY AC OV	P	1
95	SUBB	A, direct	2	CY AC OV	P	1
98+n	SUBB	A, Rn	1	CY AC OV	P	1
C4	SWAP	A	1			1
C6+i	XCH	A, @Ri	1		P	1
C5	XCH	A, direct	2		P	1
C8+n	XCH	A, Rn	1		P	1
D6+i	XCHD	A, @Ri	1		P	1
64	XRL	A, #const8	2		P	1
66+i	XRL	A, @Ri	1		P	1
65	XRL	A, direct	2		P	1
68+n	XRL	A, Rn	1		P	1
63	XRL	direct, #const8	3			2
62	XRL	direct, A	2			1