A.0 Introduction

Appendix A lists two arrangements of mnemonics for the 8051: by function, and alphabetically. The mnemonic definitions here differ from those of the original manufacturer (Intel Corporation) in the names used for addresses or data; for example, add is used to represent an address in internal RAM, whereas Intel uses the name direct. The author believes that the names used here are clearer than those used by Intel. This appendix also includes an alphabetical listing of the mnemonics using Intel names. There is **no** difference between the mnemonics when real numbers replace the names. For example: MOV add,#n and MOV direct,#data become MOV 10h,#40h when the number 10h replaces the internal RAM address (add/direct) and 40h replaces the number (#n/# data).

8051 INSTRUCTION SET

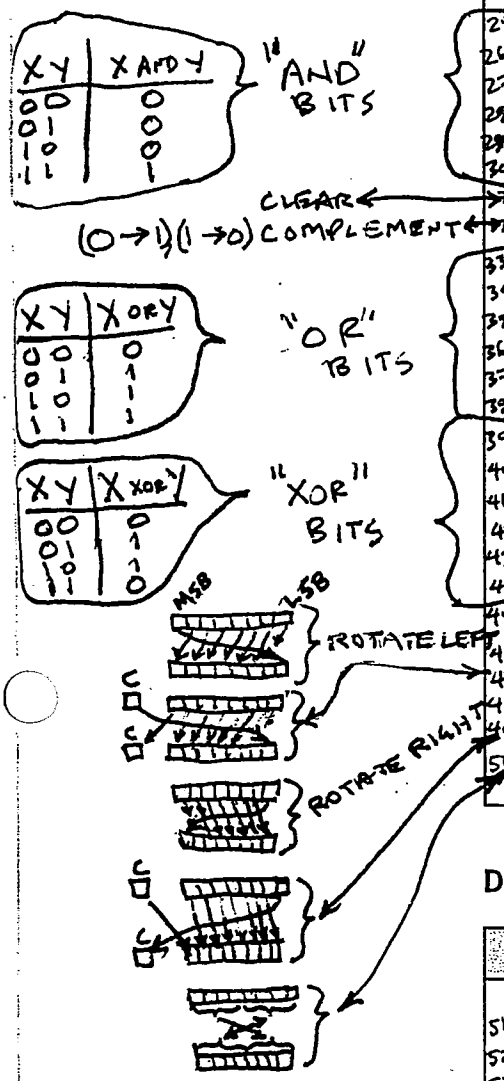
A.1 Mnemonics, Arranged by Function

Arithmetic

BAD! NEED A FOR EVERYTHING.

ADJUST SUM OF TWO BCD NUMBERS

	Mnemonic	Description	Bytes	Cycles	Flags
1	ADD A,Rr	A+Rr → A	1	1	C OV AC
2	ADD A,add	A+(add) → A	2	1	C OV AC
3	ADD A,@Rp	A+(Rp) → A	1	1	C OV AC
4	ADD A,#n	A+n → A	2	1	C OV AC
5	ADDC A,Rr	A+Rr+C → A	1	1	C OV AC
6	ADDC A,add	A+(add)+C → A	2	1	C OV AC
7	ADDC A,@Rp	A+(Rp)+C → A	1	1	C OV AC
8	ADDC A,#n	A+n+C → A	2	1	C OV AC
9	DA A	Abin → Adec ONLY WORKS FOR ADDITION	1	1	C
10	DEC A	A−1 → A	1	1	
11	DEC Rr	Rr−1 → Rr	1	1	
12	DEC add	(add)−1 → (add)	2	1	
13	DEC @Rp	(Rp)−1 → (Rp)	1	1	
14	DIV AB	A/B → AB	1	4	0 OV
15	INC A	A+1 → A	1	1	
16	INC Rr	Rr+1 → Rr	1	1	
17	INC add	(add)+1 → (add)	2	1	
18	INC @Rp	(Rp)+1 → (Rp)	1	1	
19	INC DPTR	DPTR+1 → DPTR	1	2	
20	MUL AB	A×B → AB	1	4	0 OV
21	SUBB A,Rr	A−Rr−C → A	1	1	C OV AC
22	SUBB A,add	A−(add)−C → A	2	1	C OV AC
23	SUBB A,@Rp	A−(Rp)−C → A	1	1	C OV AC
24	SUBB A,#n	A−n−C → A	2	1	C OV AC



Mnemonic	Description	Bytes	Cycles	Flags
25 ANL A,Rr	A AND Rr → A	1	1	
26 ANL A,add	A AND (add) → A	2	1	
27 ANL A,@Rp	A AND (Rp) → A	1	1	
28 ANL A,#n	A AND n → A	2	1	
29 ANL add,A	(add) AND A → (add)	2	1	
30 ANL add,#n	(add) AND n → (add)	3	2	
31 CLR A	00 → A	1	1	
32 CPL A	Ā → A	1	1	
33 ORL A,Rr	A OR Rr → A	1	1	
34 ORL A,add	A OR (add) → A	2	1	
35 ORL A,@Rp	A OR (Rp) → A	1	1	
36 ORL A,#n	A OR n → A	2	1	
37 ORL add,A	(add) OR A → (add)	2	1	
38 ORL add,#n	(add) OR n → (add)	3	2	
39 XRL A,Rr	A XOR Rr → A	1	1	
40 XRL A,add	A XOR (add) → A	2	1	
41 XRL A,@Rp	A XOR (Rp) → A	1	1	
42 XRL A,#n	A XOR n → A	2	1	
43 XRL add,A	(add) XOR A → (add)	2	1	
44 XRL add,#n	(add) XOR n → (add)	3	2	
45 NOP	PC+1 → PC	1	1	
46 RL A	A0←A7←A6..←A1←A0	1	1	
47 RLC A	C←A7←A6..←A0←C	1	1	C
48 RR A	A0→A7→A6..→A1→A0	1	1	
49 RRC A	C→A7→A6..→A0→C	1	1	C
50 SWAP A	Alsn ↔ Amsn	1	1	

Data Moves

Mnemonic	Description	Bytes	Cycles	Flags
51 MOV A,Rr	Rr → A	1	1	
52 MOV A,add	(add) → A	2	1	
53 MOV A,@Rp	(Rp) → A	1	1	
54 MOV A,#n	n → A	2	1	
55 MOV Rr,A	A → Rr	1	1	
56 MOV Rr,add	(add) → Rr	2	2	
57 MOV Rr,#n	n → Rr	2	1	
58 MOV add,A	A → (add)	2	1	
59 MOV add,Rr	Rr → (add)	2	2	
60 MOV add1,add2	(add2) → (add1)	3	2	
61 MOV add,@Rp	(Rp) → (add)	2	2	
62 MOV add,#n	n → (add)	3	2	
63 MOV @Rp,A	A → (Rp)	1	1	
64 MOV @Rp,add	(add) → (Rp)	2	2	
65 MOV @Rp,#n	n → (Rp)	2	1	

Mnemonic	Description	Bytes	Cycles	Flags
MOV DPTR,#nn	nn → DPTR	3	2	
MOVC A,@A+DPTR	(A+DPTR) → A	1	2	
MOVC A,@A+PC	(A+PC) → A	1	2	
MOVX A,@DPTR	(DPTR)^ → A	1	2	
MOVX A,@Rp	(Rp)^ → A	1	2	
MOVX @Rp,A	A → (Rp)^	1	2	
MOVX @DPTR,A	A → (DPTR)^	1	2	
POP add	(SP) → (add)	2	2	
PUSH add	(add) → (SP)	2	2	
XCH A,Rr	A ↔ Rr	1	1	
XCH A,add	A ↔ (add)	2	1	
XCH A,@Rp	A ↔ (Rp)	1	1	
XCHD A,@Rp	AIsn ↔ (Rp)Isn	1	1	

Calls and Jumps

Mnemonic	Description	Bytes	Cycles	Flags
ACALL sadd	PC+2 → (SP); sadd → PC	2	2	
CJNE A,add,radd	[A<>(add)]: PC+3+radd → PC	3	2	C
CJNE A,#n,radd	[A<>n]: PC+3+radd → PC	3	2	C
CJNE Rr,#n,radd	[Rr<>n]: PC+3+radd → PC	3	2	C
CJNE @Rp,#n,radd	[(Rp)<>n]: PC+3+radd → PC	3	2	C
DJNZ Rr,radd	[Rr−1<>00]: PC+2+radd → PC	2	2	
DJNZ add,radd	[(add)−1<>00]: PC+3+radd → PC	3	2	
LCALL ladd	PC+3 → (SP); ladd → PC	3	2	
AJMP sadd	sadd → PC	2	2	
LJMP ladd	ladd → PC	3	2	
SJMP radd	PC+2+radd → PC	2	2	
JMP @A+DPTR	DPTR+A → PC	1	2	
JC radd	[C=1]: PC+2+radd → PC	2	2	
JNC radd	[C=0]: PC+2+radd → PC	2	2	
JB b,radd	[b=1]: PC+3+radd → PC	3	2	
JNB b,radd	[b=0]: PC+3+radd → PC	3	2	
JBC b,radd	[b=1]: PC+3+radd → PC; 0 → b	3	2	
JZ radd	[A=00]: PC+2+radd → PC	2	2	
JNZ radd	[A>00]: PC+2+radd → PC	2	2	
RET	(SP) → PC	1	2	
RETI	(SP) → PC; EI	1	2	

Boolean

Mnemonic	Description	Bytes	Cycles	Flags
ANL C,b	C AND b → C	2	2	C
ANL C,/b	C AND $\overline{b}$ → C	2	2	C

Mnemonic	Description	Bytes	Cycles	Flags
CLR C	$0 \rightarrow C$	1	1	C=0
CLR b	$0 \rightarrow b$	2	1	
CPL C	$\overline{C} \rightarrow C$	1	1	C
CPL b	$\overline{b} \rightarrow b$	2	1	
ORL C,b	C OR $b \rightarrow C$	2	2	C
ORL C,/b	C OR $\overline{b} \rightarrow C$	2	2	C
MOV C,b	$b \rightarrow C$	2	1	C
MOV b,C	$C \rightarrow b$	2	2	
SETB C	$1 \rightarrow C$	1	1	C=1
SETB b	$1 \rightarrow b$	2	1	

A.2 Mnemonics, Arranged Alphabetically

Mnemonic	Description	Bytes	Cycles	Flags
ACALL sadd	PC+2 → (SP); sadd → PC	2	2	
ADD A,add	A+(add) → A	2	1	C OV AC
ADD A,@Rp	A+(Rp) → A	1	1	C OV AC
ADD A,#n	A+n → A	2	1	C OV AC
ADD A,Rr	A+Rr → A	1	1	C OV AC
ADDC A,add	A+(add)+C → A	2	1	C OV AC
ADDC A,@Rp	A+(Rp)+C → A	1	1	C OV AC
ADDC A,#n	A+n+C → A	2	1	C OV AC
ADDC A,Rr	A+Rr+C → A	1	1	C OV AC
AJMP sadd	sadd → PC	2	2	
ANL A,add	A AND (add) → A	2	1	
ANL A,@Rp	A AND (Rp) → A	1	1	
ANL A,#n	A AND n → A	2	1	
ANL A,Rr	A AND Rr → A	1	1	
ANL add,A	(add) AND A → (add)	2	1	
ANL add,#n	(add) AND n → (add)	3	2	
ANL C,b	C AND $b \rightarrow C$	2	2	C
ANL C,/b	C AND $\overline{b} \rightarrow C$	2	2	C
CJNE A,add,radd	[A<>(add)]: PC+3+radd → PC	3	2	C
CJNE A,#n,radd	[A<>n]: PC+3+radd → PC	3	2	C
CJNE @Rp,#n,radd	[(Rp)<>n]: PC+3+radd → PC	3	2	C
CJNE Rr,#n,radd	[Rr<>n]: PC+3+radd → PC	3	2	C
CLR A	$0 \rightarrow A$	1	1	
CLR b	$0 \rightarrow b$	2	1	
CLR C	$0 \rightarrow C$	1	1	C=0
CPL A	$\overline{A} \rightarrow A$	1	1	
CPL b	$\overline{b} \rightarrow b$	2	1	
CPL C	$\overline{C} \rightarrow C$	1	1	C
DA A	Abin → Adec	1	1	C
DEC A	$A-1 \rightarrow A$	1	1	
DEC add	(add)−1 → (add)	2	1	
DEC @Rp	(Rp)−1 → (Rp)	1	1	

Mnemonic	Description	Bytes	Cycles	Flags
DEC Rr	Rr−1 → Rr	1	1	
DIV AB	A/B → AB	1	4	0 OV
DJNZ add,radd	[(add)−1<>00]: PC+3+radd → PC	3	2	
DJNZ Rr,radd	[Rr−1<>00]: PC+2+radd → PC	2	2	
INC A	A+1 → A	1	1	
INC add	(add)+1 → (add)	2	1	
INC DPTR	DPTR+1 → DPTR	1	2	
INC @Rp	(Rp)+1 → (Rp)	1	1	
INC Rr	Rr+1 → Rr	1	1	
JB b,radd	[b=1]: PC+3+radd → PC	3	2	
JBC b,radd	[b=1]: PC+3+radd → PC; 0 → b	3	2	
JC radd	[C=1]: PC+2+radd → PC	2	2	
JMP @A+DPTR	DPTR+A → PC	1	2	
JNB b,radd	[b=0]: PC+3+radd → PC	3	2	
JNC radd	[C=0]: PC+2+radd → PC	2	2	
JNZ radd	[A>00]: PC+2+radd → PC	2	2	
JZ radd	[A=00]: PC+2+radd → PC	2	2	
LCALL ladd	PC+3 → (SP); ladd → PC	3	2	
LJMP ladd	ladd → PC	3	2	
MOV A,add	(add) → A	2	1	
MOV A,@Rp	(Rp) → A	1	1	
MOV A,#n	n → A	2	1	
MOV A,Rr	Rr → A	1	1	
MOV add,A	A → (add)	2	1	
MOV add1,add2	(add2) → (add1)	3	2	
MOV add,@Rp	(Rp) → (add)	2	2	
MOV add,#n	n → (add)	3	2	
MOV add,Rr	Rr → (add)	2	2	
MOV b,C	C → b	2	2	
MOV C,b	b → C	2	1	C
MOV @Rp,A	A → (Rp)	1	1	
MOV @Rp,add	(add) → (Rp)	2	2	
MOV @Rp,#n	n → (Rp)	2	1	
MOV DPTR,#nn	nn → DPTR	3	2	
MOV Rr,A	A → Rr	1	1	
MOV Rr,add	(add) → Rr	2	2	
MOV Rr,#n	n → Rr	2	1	
MOVC A,@A+DPTR	(A+DPTR) → A	1	2	
MOVC A,@A+PC	(A+PC) → A	1	2	
MOVX A,@DPTR	(DPTR)^ → A	1	2	
MOVX A,@Rp	(Rp)^ → A	1	2	
MOVX @DPTR,A	A → (DPTR)^	1	2	
MOVX @Rp,A	A → (Rp)^	1	2	
MUL AB	A×B → AB	1	4	0 OV
NOP	PC+1 → PC	1	1	
ORL A,add	A OR (add) → A	2	1	
ORL A,@Rp	A OR (Rp) → A	1	1	
ORL A,#n	A OR n → A	2	1	
ORL A,Rr	A OR Rr → A	1	1	
ORL add,A	(add) OR A → (add)	2	1	
ORL add,#n	(add) OR n → (add)	3	2	

Mnemonic	Description	Bytes	Cycles	Flags
ORL C,b	C OR b → C	2	2	C
ORL C,/b	C OR /b → C	2	2	C
POP add	(SP) → (add)	2	2	
PUSH add	(add) → (SP)	2	2	
RET	(SP) → PC	1	2	
RETI	(SP) → PC; EI	1	2	
RL A	A0←A7←A6..←A1←A0	1	1	
RLC A	C←A7←A6..←A0←C	1	1	C
RR A	A0→A7→A6..→A1→A0	1	1	
RRC A	C→A7→A6..→A0→C	1	1	C
SETB b	1 → b	2	1	
SETB C	1 → C	1	1	C=1
SJMP radd	PC+2+radd → PC	2	2	
SUBB A,add	A−(add)−C → A	2	1	C OV AC
SUBB A,@Rp	A−(Rp)−C → A	1	1	C OV AC
SUBB A,#n	A−n−C → A	2	1	C OV AC
SUBB A,Rr	A−Rr−C → A	1	1	C OV AC
SWAP A	Alsn ↔ Amsn	1	1	
XCH A,add	A ↔ (add)	2	1	
XCH A,@Rp	A ↔ (Rp)	1	1	
XCH A,Rr	A ↔ Rr	1	1	
XCHD A,@Rp	Alsn ↔ (Rp)lsn	1	1	
XRL A,add	A XOR (add) → A	2	1	
XRL A,@Rp	A XOR (Rp) → A	1	1	
XRL A,#n	A XOR n → A	2	1	
XRL A,Rr	A XOR Rr → A	1	1	
XRL add,A	(add) XOR A → (add)	2	1	
XRL add,#n	(add) XOR n → (add)	3	2	

Mnemonic Acronyms

add	Address of the internal RAM from 00h to FFh.
ladd	Long address of 16 bits from 0000h to FFFFh.
radd	Relative address, a signed number from −128d to +127d.
sadd	Short address of 11 bits; complete address = PC11–PC15 and sadd.
b	Addressable bit in internal RAM or a SFR.
C	The Carry flag.
lsn	Least significant nibble.
msn	Most significant nibble.
n	Any immediate 8-bit number from 00h to FFh.
Rr	Any of the eight registers, R0 to R7, in the selected bank.
Rp	Either of the pointing registers R0 or R1 in the selected bank.
[ ]:	IF the condition inside the brackets is *true*, THEN the action listed will occur; ELSE go to the next instruction.
^	External memory location.
( )	Contents of the location inside the parentheses.

Note that flags affected by each instruction are shown where appropriate; any operations that affect the PSW address may also affect the flags.