

8051 – Instruction set

ARITHMETIC INSTRUCTIONS

ADD	A,#byte	;A ← A + byte
ADD	A,@Rn	;A ← A + DM(Rn)
ADD	A,Ri	;A ← A + Ri
ADD	A,byte	;A ← A + DM(byte)
ADDC	A,#byte	;A ← A + byte + CY
ADDC	A,@Rn	;A ← A + DM(Rn) + CY
ADDC	A,Ri	;A ← A + Ri + CY
ADDC	A,byte	;A ← A + DM(byte) + CY

LOGICAL INSTRUCTIONS

ANL	A,#byte	;A \leftarrow A \wedge byte
ANL	A,@Rn	;A \leftarrow A \wedge DM(Rn)
ANL	A,Ri	;A \leftarrow A \wedge Ri
ANL	A,byte	;A \leftarrow A \wedge DM(byte)
ANL	dbyte,#byte	;DM(dbyte) \leftarrow DM(dbyte) \wedge byte
ANL	byte,A	;DM(byte) \leftarrow A \wedge DM(byte)
CLR	A	;A \leftarrow 0
CPL	A	;A \leftarrow \overline{A}
RL	A	;A \leftarrow A(6..0) A(7)
RLC	A	;CY A \leftarrow A CY
RR	A	;A \leftarrow A(0) A(7..1)
RRC	A	;A CY \leftarrow CY A
SWAP	A	;A \leftarrow A(3..0) A(7..4)

DATA MOVE INSTRUCTIONS

MOV	A,#byte	;A ← byte
MOV	A,@Rn	;A ← DM(Rn)
MOV	A,Ri	;A ← Ri
MOV	A,byte	;A ← DM(byte)
MOV	@Rn,A	;DM(Rn) ← A
MOV	@Rn,#byte	;DM(Rn) ← byte
MOV	@Rn,byte	;DM(Rn) ← DM(byte)
MOV	Ri,A	;Ri ← A
MOV	Ri,#byte	;Ri ← byte
MOV	Ri,byte	;Ri ← DM(byte)
MOV	byte,A	;DM(byte) ← A

MOV	dbyte,#byte	;DM(dbyte) ← byte
MOV	byte,@Rn	;DM(byte) ← DM(Rn)
MOV	byte,Ri	;DM(byte) ← Ri
MOV	dbyte,byte	;DM(dbyte) ← DM(byte)
MOV	DPTR,#addr	;DPTR ← addr
MOVC	A,@A + DPTR	;A ← CM(A+DPTR)
MOVC	A,@A + PC	;A ← CM(A+PC)
MOVX	A,@DPTR	;A ← XM(DPTR)
MOVX	A,@Rn	;A ← XM(P2 Rn)
MOVX	@DPTR,A	;XM(DPTR) ← A
MOVX	@Rn,A	;XM(P2 Rn) ← A

Push, Pop, and exchange

PUSH	byte	;DM(SP+1) \leftarrow DM(byte), SP \leftarrow SP + 1
POP	byte	;DM(byte) \leftarrow DM(SP), SP \leftarrow SP - 1
XCH	A,Ri	;A \leftrightarrow Ri
XCH	A,byte	;A \leftrightarrow DM(byte)
XCH	A,@Rn	;A \leftrightarrow DM(Ri)
XCHD	A,@Rn	;A(3..0) \leftrightarrow DM(Rn;3..0)

PROGRAM AND MACHINE CONTROL

CALL	Note: Assembler translates CALL to ACALL or LCALL	
ACALL	short	;DM(SP+2) DM(SP+1) ← PC+2, ;PC(10..0) ← short, SP ← SP + 2
LCALL	addr	;DM(SP+2) DM(SP+1) ← PC+3, ;PC(15..0) ← addr, SP ← SP + 2
RET		;PC ← DM(SP) DM(SP-1), SP ← SP-2
RETI		;PC ← DM(SP) DM(SP-1), SP ← SP-2 ;Reenable equal or lower priority INT
JMP	Note: JMP is translated to AJMP, LJMP, or SJMP.	
AJMP	short	;PC(10..0) ← short
LJMP	addr	;PC(15..0) ← addr
SJMP	byte	;PC ← PC + 2 + byte(7)..byte(7) byte
JMP	@A+DPTR	;PC ← DPTR + A
JZ	byte	;IF A = 0 THEN ;PC ← PC + 2 + byte(7)..byte(7) byte ;ELSE PC ← PC+2
JNZ	byte	;IF A ≠ 0 THEN ;PC ← PC + 2 + byte(7)..byte(7) byte ;ELSE PC ← PC+2

CJNE	A,dbyte,byte	<pre> ;IF A ≠ DM(dbyte) THEN ;PC ← PC + 2 + byte(7)..byte(7) byte ;IF A < DM(dbyte) THEN CY ← 1 ;ELSE CY ← 0 </pre>
CJNE	A,#dbyte,byte	<pre> ;IF A ≠ dbyte THEN ;PC ← PC + 2 + byte(7)..byte(7) byte ;IF A < dbyte THEN CY ← 1 ;ELSE CY ← 0 </pre>
CJNE	Rn,#dbyte,byte	<pre> ;IF Rn ≠ dbyte THEN ;PC ← PC + 2 + byte(7)..byte(7) byte ;IF A < DM(dbyte) THEN CY ← 1 ;ELSE CY ← 0 </pre>
CJNE	@Rn,#dbyte,byte	<pre> ;IF DM(Rn) ≠ dbyte THEN ;PC ← PC + 2 + byte(7)..byte(7) byte ;IF A < dbyte THEN CY ← 1 ;ELSE CY ← 0 </pre>

DJNZ	Rn,byte	<pre> ;Rn ← Rn-1, IF (Rn-1) ≠ 0 THEN ;PC ← PC + 2 + byte(7)..byte(7) byte ;ELSE PC ← PC + 2 </pre>
DJNZ	dbyte,byte	<pre> ;DM(dbyte) ← DM(dbyte)-1, ;IF (DM(dbyte)-1) ≠ 0 THEN ;PC ← PC + 3 + byte(7)..byte(7) byte ;ELSE PC ← PC + 3 </pre>
NOP		<pre> ;PC ← PC + 1 </pre>

BIT MANIPULATION INSTRUCTIONS

CLR	C	;CY \leftarrow 0
CLR	byte	;BADM(byte) \leftarrow 0
SETB	C	;CY \leftarrow 1
SETB	byte	;BADM(byte) \leftarrow 1
CPL	C	;CY \leftarrow $\bar{C}Y$
CPL	byte	;BADM(byte) \leftarrow $\overline{\text{BADM(byte)}}$
ANL	C,byte	;CY \leftarrow CY \wedge $\overline{\text{BADM(byte)}}$
ANL	C,/byte	;CY \leftarrow CY \wedge $\overline{\text{BADM(byte)}}$
ANL	byte,bit	;BADM(byte) \leftarrow BADM(byte) \wedge bit
ORL	C,byte	;CY \leftarrow CY \vee $\overline{\text{BADM(byte)}}$
ORL	C,/byte	;CY \leftarrow CY \vee $\overline{\text{BADM(byte)}}$
ORL	byte,bit	;BADM(byte) \leftarrow BADM(byte) \vee bit
MOV	C,byte	;CY \leftarrow BADM(byte)
MOV	byte,C	;BADM(byte) \leftarrow CY

BIT JUMP INSTRUCTIONS

JB	dbyte,byte	;IF BADM(dbyte) = 1 THEN ;PC \leftarrow PC + 3 + byte(7)..byte(7) byte ;ELSE PC \leftarrow PC+3
JNB	dbyte,byte	;IF BADM(dbyte) = 0 THEN ;PC \leftarrow PC + 3 + byte(7)..byte(7) byte ;ELSE PC \leftarrow PC+3
JBC	dbyte,byte	;IF BADM(dbyte) = 1 THEN ; BADM(dbyte) \leftarrow 0 ; PC \leftarrow PC + 3 + byte(7)..byte(7) byt ;ELSE PC \leftarrow PC+3
JC	byte	;IF CY = 1 THEN ;PC \leftarrow PC + 3 + byte(7)..byte(7) byte ;ELSE PC \leftarrow PC+3
JNC	byte	;IF CY = 0 THEN ;PC \leftarrow PC + 3 + byte(7)..byte(7) byte ;ELSE PC \leftarrow PC+3

INSTRUCTIONS THAT AFFECT FLAGS (i)

Instruction	CY	OV	AC	Instruction	CY	OV	AC
ADD	X	X	X	CLR C	0	-	-
ADDC	X	X	X	CPL C	X	-	-
SUBB	X	X	X	ANL C,bit	X	-	-
MUL	0	X	-	ANL C,/bit	X	-	-
DIV	0	X	-	ORL C,bit	X	-	-
DA	X	-	-	ORL C,bit	X	-	-
RRC	X	-	-	MOV C,bit	X	-	-
RLC	X	-	-	CJNE	X	-	-
SETB C	1	-	-				