

Chapter 3

Instruction Set Summary

(I. Scott Mackenzie)

8051 Addressing Modes

There are basically **5 ways of specifying source/destination** operand addresses:

1. Particular On-chip Resources:

This includes the Accumulator (A), the Stack Pointer (SP), the Data Pointer (DP), the Program Counter (PC), and the Carry (C). Other On-chip Registers are Memory-mapped while these have special Op-codes.

2. Immediate operands:

The # sign is the designator. These are 8-bits except for DPTR contents (16-bits).

3. Register operands:

Designated as R_n, where n is 0..7. One of the four Register Banks is used (selected by RS0 and RS1 in PSW).

4. Direct Operands:

From 00 to FF Hex, specifies one of the internal data addresses.

5. Indirect Address:

Designated as @R_i, where i is 0 or 1, uses the contents of R0 or R1 in the selected Register Bank to specify the address. Other form is @A, using Accumulator contents.

8051 Addressing Modes

Addressing modes are an integral part of each computers instruction set. They allow different ways of specifying source/destination operand addresses depending on the programming situation. There are **8 modes of addressing**:

1. **Immediate**
2. **Register**
3. **Direct**
4. **Indirect**
5. **Relative**
6. **Absolute**
7. **Long**
8. **Indexed**

General Format of Instruction:

Label: OpCode Destination, source ; Comments

Instruction Set : Arithmetic

Mnemonics **Operands** **Bytes/Cycles**

ADD	-----	A, Rn	1/1
ADDC		A, direct	2/1
SUBB		A, @Ri	1/1
		A, #data	2/1
INC	-----	A	1/1
DEC		Rn	1/1
		direct	2/1
		@Ri	1/1
INC		DPTR	1/2
MUL	-----	AB	1/4
DIV		AB	1/4
DA		A	1/1

In Rn, n is 0..7. One of the four Register Banks is used (selected by RS0 and RS1 in PSW)

MOV PSW, #00011000B ; Select Register Bank 3

ADD A, R7 ; Add the contents of Register 7 to the Acc.

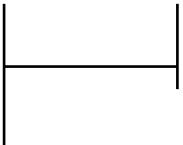
In Ri, i is 0 or 1.

By: Masud-ul-Hasan

Instruction Set : Logical

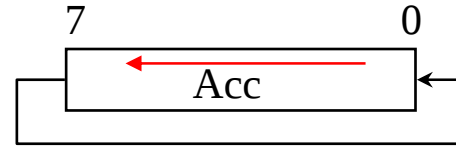
Mnemonic		Operands	Bytes/Cycles
ANL	-----	A, Rn	1/1
ORL		A, direct	2/1
XRL		A, @Ri	1/1
		A, #data	2/1
		direct, A	2/1
		direct, #data	3/2
		C, bit	2/2
CLR	-----	A	1/1
CPL		C	1/1
		bit	2/1

Instruction Set : Logical (cont'd)

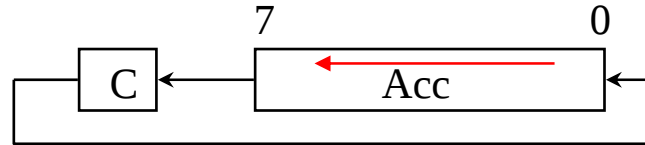
Mnemonic	Operands	Bytes/Cycles
RL	A	1/1
RLC	A	1/1
RR	A	1/1
RRC	A	1/1
SWAP	A	1/1
SETB		1/1
CLR		2/1
CPL		

Instruction Set : Logical (cont'd)

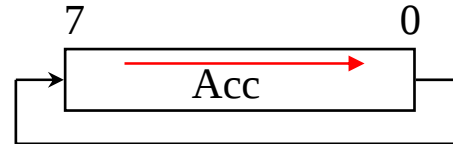
RL A



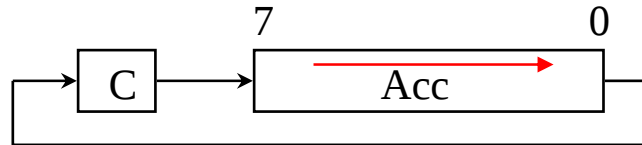
RLC A



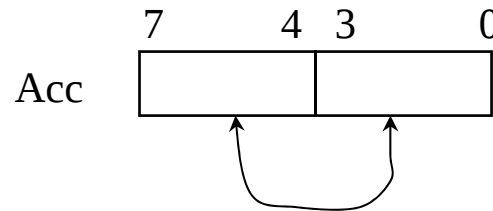
RR A



RRC A



SWAP A



Instruction Set : Data Transfer

Mnemonic	Operands	Bytes/Cycles
MOV	A, Rn	1/1
	A, direct	2/1
	A, @Ri	1/1
	A, #data	2/1
	Rn, A	1/1
	Rn , direct	2/2
	Rn, #data	2/1
	direct, A	2/1
	direct, Rn	2/2
	direct, direct	3/2
	direct, @Ri	2/2
	direct, #data	3/2

Instruction Set : Data Transfer (cont'd)

Mnemonic	Operands	Bytes/Cycles
MOV	@Ri, A	1/1
	@Ri, direct	2/2
	@Ri, #data	2/1
	DPTR, #data16	3/2
	C, bit	2/1
	bit, C	2/2
MOVB	A, @DPTR	1/2
	@DPTR, A	1/2
	A, @Ri	1/2
	@Ri, A	1/2

Instruction Set: Data Transfer (cont'd)

Mnemonic	Operands	Bytes/Cycles
MOVC	A, @A+DPTR	1/2
	A, @A+PC	1/2
PUSH	direct	2/2
POP	direct	2/2
XCH	A, Rn	1/1
	A, direct	2/1
	A, @Ri	1/1
XCHD	A, @Ri	1/1

Instruction Set : Branching

Mnemonic	Operands	Bytes/Cycles
LCALL	addr16	3/2
ACALL	addr11	2/2
RET	-	1/2
RETI	-	1/2
LJMP	addr16	3/2
AJMP	addr11	2/2
SJMP	rel	2/2
JMP	@A+DPTR	1/2
JZ	rel	2/2
JNZ	rel	2/2

Instruction Set : Branching (cont'd)

Mnemonic	Operands	Bytes/Cycles
CJNE	A, direct, rel	3/2
	A, #data, rel	3/2
	Rn, #data, rel	3/2
	@Ri, #data, rel	3/2
DJNZ	Rn, rel	2/2
	direct, rel	3/2
NOP	-	1/1
JC	rel	2/2
JNC	rel	2/2
JB	bit, rel	3/2
JNB	bit, rel	3/2
JBC	bit, rel	3/2