## Addressing modes:

Addressing mode specify the way in which operands are accessed by CPU. There are 5 types.

1) Immediate addressing mode
2) Register addressing mode.
3) Direct addressing mode.
4) Register indirect addressing
5) Indexed addressing mode.

### Immediate addressing mode:

The source operand is constant & it is specified immediately after the opcode. That is data is specified in instruction itself. Immediate data defined by using # sign.

```
Eg: MOV A, #52H      ; load 52H into A.
    MOV DPTR, #52H   ; load 52H into DPTR.
                       DPTR = 52H.
```

### Register addressing mode:

The data is placed in one of CPU registers.
We can move data between accumulator & Rn (n=0 to 7) but movement of data between Rn registers is not permitted. for eg the instruction MOV R2, R6 is not valid.

```
Eg: MOV A, R0    ; copy contents of R0 into A.
    MOV R7, A    ; save accumulator in R7.
```

### Direct addressing mode:

8 bit address given in instruction itself. The data is available in internal RAM. The RAM has been assigned

# not used in this since it uses only RAM location addresses.

Eg: MOV R0, 60H ; save content of RAM location 60H into R0

MOV 35H, A ; save content of A in RAM location 35H.

## Register indirect addressing mode:

The register is used as pointer to the data. The addres register for 8 bit addres can be R0(@r) R1 of register bank. When R0 & R1 are used as pointers that is when they hold addres of RAM location it must be defined by @ sign.

Eg: MOV A, @R0 ; Move contents of RAM location whose address is held by R0 into A.

## Indexed addressing mode:

Widely used in accessing data elements of lookup table entries located in program ROM space of 8051.

The instruction used for this purpose is `MOVC A, @A+DPTR`

16 bit register DPTR + register A

MOVC is used instead of MOV with C denoting program code.

Content of A added with 16 bit register to form 16 bit addres