# *ADDRESSING MODES*

*The way in which an operand is specified in the instruction.*

## *TYPES:*

- *Immediate addressing mode*
- *Register addressing mode*
- *Direct addressing mode*
- *Register indirect addressing mode*
- *Implied addressing mode*
- *Relative addressing mode*

## *Immediate addressing mode:*

- *In this mode 8/16 bit data is specified in the instruction itself.*
- *Sign'#' indicates it is immediate addressing mode.*

*(eg):*

*MOV A, #2Ch* ***//move 2CH to accumulator***

## *Register addressing mode:*

*In this mode, the instruction will specify the name of the register in which the data is available.*

*(eg):*

*MOV R1,A* ***//accumulator content is moved to register R1***

*MOV A,R2* ***// content R3 is moved to the accumulator***

## *Direct addressing mode:*

*In this mode, the address of the data is directly specified in the instruction.*

*(eg):*

*MOV A,05H* *//move the content of R5 to the accumulator*

## *Register indirect addressing mode:*

- *In this mode, the instruction specifies the name of the register in which the address of the data is available.*
- *Sign'@' indicates indirect addressing*

*(eg):*

*MOV A,@R1* *//move the content of the address in R1 to A*

## *Implied addressing mode:*

*In this mode, operand is specified in the instruction.*

*(eg):*

*CPL C* *//complement carry flag*

*CLR A* *//clear accumulator*

## *Relative addressing mode:*

*In this mode, the instruction specifies the address relative to program counter.*

*The instruction will carry an offset whose value varies from -128 to +127. This offset value is added to the program counter to generate the 16 bit physical address.*

## *Applications:*

*Immediate Addressing mode: It is ideal for loading constant.*

*Register Addressing mode: It is used for frequently accessed variables in loops.*

*Direct Addressing mode: It is used for accessing fixed memory locations.*

*Register Indirect Addressing mode: It is used to hold a memory location's address.*

*Relative Addressing mode: It is used to relocate the program at run time.*