**ADDRESSING MODES** :-

* An instruction performs an operation on the specified data is known as **operand**.
* Addressing modes indicates a way of locating data or operand which is accessed by an instruction that to be executed.

**categories of addressing modes:**

* + - 1. sequential control flow
      2. control transfer instruction

**1. Sequential control flow instruction:-**

which offers execution, control transfer to the next instruction appearing immediately after it in the program.

Eg:- arithmetic, logical , data transfer, processor control

**2. Control transfer instruction**

Instructions transfer control to the some predefined address or the address some how specified in the instructions ,after their execution.

Eg:- INT,CALL,RET,& JUMP

**Addressing modes in sequential control flow**

There are 8 addressing modes for sequential control flow

* In this 2-addressing modes are provided for instructions which operate on Registers (or) immediate operands
* The reaming 6- addressing modes are for specifying an operand stored in the memory

**Types of Addressing modes for Register (or) operand instructions**

* 1. Immediate Addressing Mode
  2. Direct Addressing Mode
  3. Register Addressing Mode
  4. Register Indirect Addressing Mode
  5. Indexed Addressing Mode
  6. Register Relative Addressing Mode
  7. Based Indexed Addressing Mode
  8. Relative Based Indexed Addressing Modes

**Register addressing mode :**

The operand to be accessed is specified as residing in an internal register of 8086.

Fig. below shows internal registers, any one can be used as a source or destination operand, however only the data registers can be accessed as either a byte or word.

|  |  |  |
| --- | --- | --- |
| REGISTER | OPERAND | SIZES |
| BYTE(REG-8) | WORD (reg-16) |
| ACCUMULATOR | AL, AH | AX |
| BASE | BL,BH | BX |
| COUNT | CL,CH | CX |
| DATA | DL,DH | DX |
| STACK POINTER | - | SP |
| BASE POINTER | - | BP |
| SOURCE INDEX | - | SI |
| DESTINATION INDEX | - | SI |
| CODE SEGMENT | - | CS |
| DATA SEGMENT | - | DS |
| STACK SEGMENT | - | SS |
| EXTRA SEGMENT | - | ES |

* 1. **Immediate Addressing Mode**

The data is part of instruction itself and is available in successive bytes of instruction code.

**Example: -**

MOV AL,26H //\*\*\*\*\* copies 8-bit data 26H in to AL register

(since AL is 8-bit register)

**Example:-**

MOV AX, 4653H ///\*\*\*\*\*copies 16-bit data 4653H in to ax register pair

( Since AX is pair of two 8-bit AL & AH registers)

**2. Direct Addressing Mode**

The instruction operands specifies in memory address (offset) where data is located.

**Example:-**

MOV AX, [5000] ; 5000 is a Effective Address [offset] directly written in the address.

Data resides in a memory location in data segment(DS), whose Effective Address is computed using **5000H** as the **offset address &** content **DS** as segment address

**Effective Address {EA}** here is = **10H \* DS+ 5000H**

**3. Register Addressing Mode**

Refers the data in a register (or) in a register pair . All the registers, except IP, may be used in this mode.

**Eg: -**

MOV AX, CX //\*\*\*\*copies 16-bit content of CX into AX register pair

MOV AL, CL //\*\*\*\* copies 8-bit content of CL into AL register

**4.Register Indirect Addressing Mode**

The address of the memory location which contain data or operand is determined as indirect way. Using the offset register, this mode of addressing is known as register indirect mode.

In this mode the offset address is either BX (or) SI (or) DI register.

The default segment is either DS (or) ES .

**Eg:**

**MOV** AX, [BX] //\*\* data present in memory location in DS whose offset address is in BX.

Effective address = **10H\*DS+[BX]**

**5. Indexed Addressing Mode**

* In this addressing mode, offset of the operand is stores in one of the index registers. DS is the default segment for index register SI and DI . in this case DS and ES are default segment for SI and DI respectively .
* This mode is special case of the above discussed register indirect addressing mode.

**Eg:-**

**MOV AX, [SI]**

Here data is available at an offset address stored in SI in DS. The Effective Address, in this case is computed as

**EA=10H\*DS+[SI]**

**6. Register Relative Addressing Mode**

* In this addressing mode , data is available at EA formed by adding 8-bit (or) 16-bit displacement With content of any of registers BX,BP,SI and DI in the default (either DS or ES) segment.

**Eg:-**

**MOV AX, 50H[BX]**

Here **Effective Address = 10H\*DS+50H+[BX]**

**7.Based Indexed Addressing Mode**

* **Based Addressing Mode:**

The effective address will be the contents of base register (or) base pointer register.

* **Based Indexed Addressing Mode:**
* This is same as register indirect
* The effective address of the operand is formed by adding contents of base register to the index registers.
* DS and ES are default segments,
* SI and DI are used as index registers and BX and BP are used as base registers.

**Example:** MOV AX, [BX][SI]

**8. Relative Based Indexed Addressing Modes**

The Effective address of the operand is formed by adding 8-bit or 16-bit displacement with the contents of any one of the base registers( i.e. BX/BP) and index register (i.e. SI/DI) in the default segments.

**Example: MOV AX, 50H [BX][SI]**

Here 50H is immediate displacement

**2. Control transfer instruction**

* The Program memory access addressing modes explains about the transfer of control to some other location which is not in sequence hence is applicable to control transfer instructions and is sometimes known as Control Transfer Addressing Modes.
* The control transfer instructions, the addressing modes depend upon whether the destination location is within the same segment or in different segment. It also depends upon the method of passing the destination address to the processor .
* The various control transfer addressing modes are:
  1. Inter segment addressing modes
  2. Intra segment addressing modes
* If the location to which the control is to be transferred lies in the same segment as the current location then it is called Intra-segment mode .
* If the destination lies in a different segment then it is known as Inter-segment mode.
* **Intra-Segment Direct Addressing Mode:**

The address to which the control is to be transferred lies in the same segment in which the control transfer instruction lies and appears directly in the instruction as an immediate displacement value .

* The displacement is computed relative to the contents of IP. The EA is sum of 8-bit or 16-bit displacement and current contents of the IP.

The effective address to which the control will be transferred is given by the sum of 8 or 16-bit displacement and current contents of IP.In case of jump instruction, if the signed displacement (d) is of 8-bits(i.e.,-128<d<+127), we term it as *Short jump* and if it is of 16-bits(i.e., -32768<d<+32767), it is termed as *Long jump.*

**Example : JMP SHORT *Label***

* **Intra-Segment Indirect Addressing Mode:**

The displacement to which the control is to be transferred is in the same segment in which the control is transfer instruction lies but it is passed to the instruction indirectly. The branch address is found as the content of a register or a memory location.This addressing mode is used in unconditional branch instructions.

**Example: JMP [BX]**

* **Inter-segment Direct Addressing Mode:**

In this mode, the address to which the control is to be transferred is in a different segment. This addressing mode Provides a means of branching from one code segment to another code segment.Here the CS and IP of the destination address are specified directly in the instruction

**Example:**

JMP 5000H:2000H

[CS]:5000H,[IP]:2000H

* **Inter-segment Indirect Addressing Mode:**

In this mode, the address to which the control is to be transferred lies in a different segment and it is passed to the instruction indirectly,i.e.,contents of a memory block containing four bytes, i.e.,IP(LSB),IP(MSB),CS(LSB), CS(MSB) sequentially. The starting address of the memory block may be referred using any of the addressing modes,except immediate addressing mode.

**Example:**

JMP [2000H]

[IP]=[DS:2001H,DS:2000H],[CS]=[DS:2004H,DS:2003H]