Addressing Modes Assembly Language Instructions:

- In assembly language program the instruction is consists of two parts:
    - Opcode
    - Operand
- When the microprocessor executes an instruction, it performs the specific function on data.
- The opcode specifies the function to be performed on the data.
- The data which is the operand may be part of the instruction, may reside in one of the internal registers, may be stores at a memory location, or may be held at an I/O port.
- To access these types of operands, the 8086 microprocessor provides the following addressing Modes.

1. Immediate Addressing Mode:

- The addressing mode in which the data operand (source operand) is a part of the instruction itself is known as immediate addressing mode.
- Destination operand can never be immediate data.
- For Example:

```
MOV AX, 2000
MOV CL, 0A
ADD AL, 45
AND AX, 0000
```

- Note that to initialize the value of segment register another register is required.
- For Example:

```
MOV AX, 2000
MOV CS, AX
```

2. Register Addressing Mode:

- In this type of addressing mode both the operands are registers.
- For Example:

```
MOV AX, BX
XOR AX, DX
ADD AL, BL
```

3. Displacement or Direct Addressing Mode:

- Displacement is an 8-bits or 16-bits immediate value given in the instruction.
- In this addressing mode the effective address of the memory location is written directly in the instruction as displacement.
- For Example

```
MOV AX, [DISP]
MOV AX, [0500]
```

4. Register Indirect Addressing Mode:

- This addressing mode allows data to be addressed at any memory location through an offset address held in base or index registers i.e. BP, BX, DI, & SI.
- For Example:

```
MOV AX, [DI]
ADD AL, [BX]
MOV AX, [SI]
```

5. Based Addressing Mode:

- In this addressing mode, the offset address of the operand is given by the sum of contents of the base registers (BX or BP) and 8-bit/16-bit displacement.

- For Example:

```
MOV AL, [BP + 0100]
ADD AX, [BX + 0500]
```

6. Indexed Addressing Mode:

- In this addressing mode, the operands offset address is found by adding the contents of the index registers (SI or DI) and 8-bit/16-bit displacements.
- For Example:

```
MOV AX, [SI + 2000]
SUB AH, [DI + 3000]
```

7: Based Indexed Addressing Mode:

- In this the effective address is sum of the contents of base register (BX or BP) and index register (SI or DI).
- For Example:

```
MOV CL, [BP + SI]
ADD AX, [BX + DI]
```

8. Based Indexed Displacement Addressing Mode:

- In this addressing mode, the operands offset is computed by adding the contents Base register (BX, BP), Index register (SI or DI) and 8 or 16-bit displacement.
- For Example:

```
MOV AL, [BP + SI + 2000]
ADD CX, [BX + DI + 0016]
```

9. String Addressing Mode

- This addressing mode is related to string instructions. In this the value of SI and DI are auto incremented and decremented depending upon the value of directional flag.
- For Example:

```
MOVS B
MOVS W
```

10. Input/Output Addressing Mode:

- This addressing mode is related with input output operations.
- For Example:

```
IN A, 45
OUT A, 50
```