Internal Architecture and Features of 8086 Microprocessor

- **Execution Unit (EU)**
  - **Stack Pointer (SP) and Base Pointer (BP)**

➢SP is a 16-bit register is used to hold the offset address for stack segment.

➢BP is a 16-bit register is used to hold the offset address for stack segment. BP register is usually used for based, based indexed or register indirect addressing.

➢The difference between SP and BP is that the SP is used internally to store the address in case of interrupt and CALL instruction

  - **Source Index (SI) and Destination Index (DI) Registers**

➢SI is used for indexed, based indexed and register indirect addressing, as well as a source data addresses in string manipulation instructions

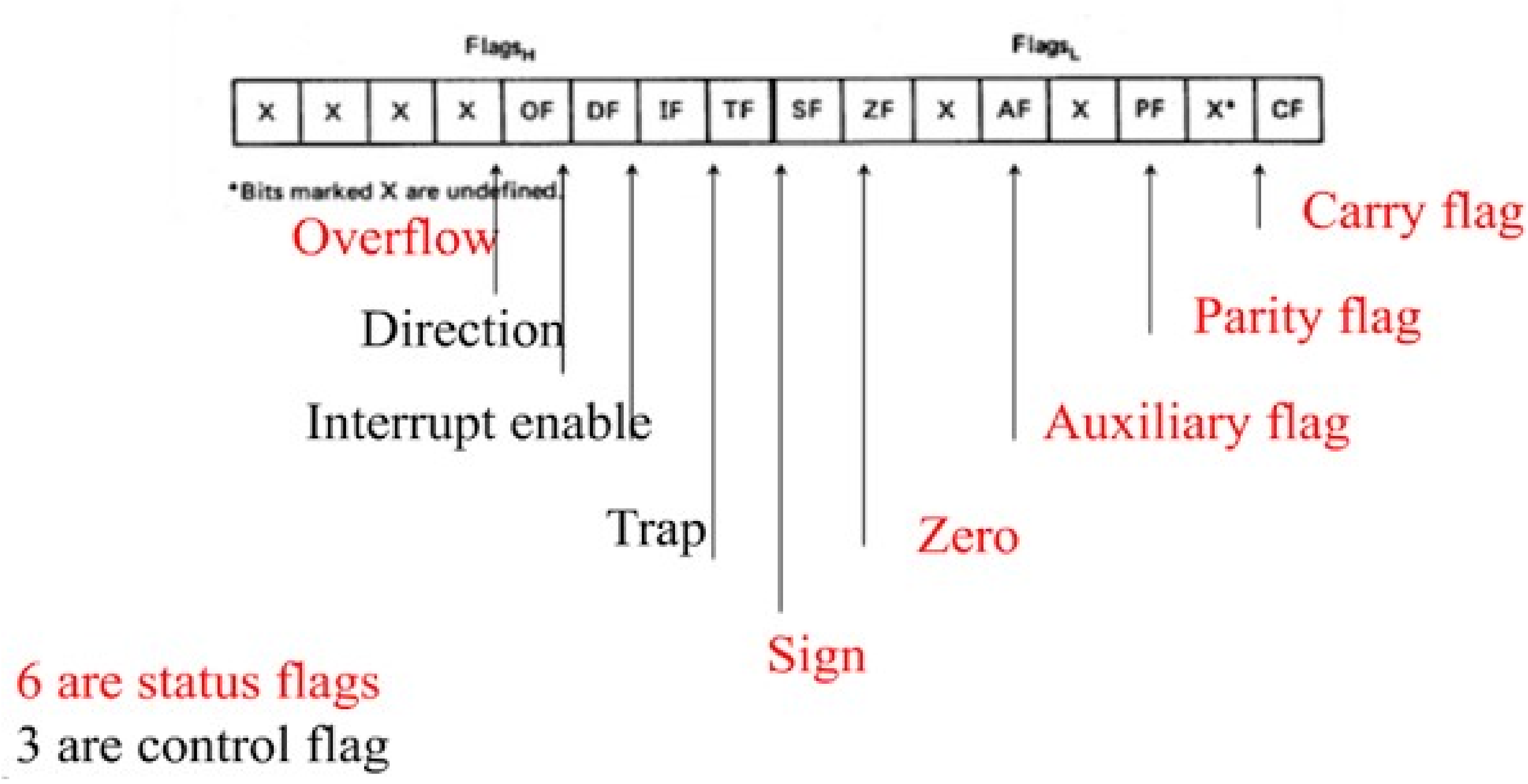
➢DI is used for indexed, based indexed and register indirect addressing, as well as a destination data addresses in string manipulation instructions.

Internal Architecture and Features of 8086 Microprocessor

- **Execution Unit (EU)**
  - **Arithmetic and Logical Unit (ALU)**
  - ➢Performs arithmetic and logical operations
  - ➢8-bit or 16-bit mathematical operations such as addition, subtraction, multiplication, division, data conversion, logical operation like AND, OR, NOT
  - ➢Also performs increment, decrement and shift operations
  - **Flag Register**
  - ➢It is a 16–bits register containing nine active flags.
  - ➢A flag is a flip–flop that typically holds a particular status of last ALU operation or controls the certain operation of the execution unit (EU).
  - ➢The nine 8086 flags are classified into two categories: 6 Status flags and 3 control flags .

Internal Architecture and Features of 8086 Microprocessor

- **Execution Unit (EU)**
  - **Flag Register**



6 are status flags
3 are control flag

Internal Architecture and Features of 8086 Microprocessor

- **Execution Unit (EU)**
  - **Flag Register**

➢ ***Carry (CY):*** is set if the MSB of 8–bits or 16–bits ALU operation generates a carry.

➢ ***Parity (P):*** is set if the 8–bits or lower 8–bits of a 16–bits result has even number of 1’s.

➢ ***Auxiliary Carry (AC):*** is set if the D_3 bits of two numbers generate a carry in 8–bits or 16–bits ALU operation.

➢ ***Zero (z):*** is set if the result of 8–bits or 16–bits ALU operation zero.

➢ ***Sign (s):*** is set if the MSB of 8–bits or 16–bits result of an ALU operation is 1.

➢ ***Overflow (O):*** is set if there is a carry in 8–bits or 16–bits signed ALU operation.

➢ ***Trap (T):*** If T is set, it puts the processor in single step mode to encourage step-wise program execution. Otherwise, processor will be in complete execution mode

➢ ***Interrupt (I):*** If I is set, the INTR pin is enabled thereby allowing the devices to interrupt microprocessor. It I is reset, the microprocessor does not recognize the interrupts generated by other devices (disables INTR pin). This flag can be or reset by using STI or CLI instructions respectively.

➢ ***Direction (D) :*** If D is set, then it automatically decrements the source and destination string indices SI & DI respectively during string instruction if D is reset, they are automatically incremented. The STD and CLD instruction are used to set and reset the direction flag respectively.

Addressing Modes of 8086

- Following are the addressing modes of 8086
  - Implied addressing mode
  - Immediate addressing mode
  - Register direct addressing mode
  - Register indirect addressing mode
  - Absolute addressing mode
  - Displacement addressing mode
    - Based addressing mode
    - Indexed addressing mode
    - Based indexed addressing mode

Addressing Modes of 8086

- **Implied addressing mode**
  - In this mode, the operands are implied and are hence not specified in the instruction

    Example: STC, CMC,CLC,STD
- **Immediate addressing mode**
  - In this mode, the operand is specified in the instruction itself
  - 8-bit or 16-bit data can be specified as a part of instruction

    Example:
    ```
    MOV CL, 12H;      CL <- 12H
    MOV AX, 6547H;    AX <- 6547H
    ```
- **Register Direct addressing mode**
  - This mode specifies the source operand, destination operand, or both to be contained in an 8086 register

    Example:
    ```
    MOV AL, BL
    MOV BX, DX
    ```

Addressing Modes of 8086

- **Register Indirect addressing mode**
  - In this mode, the effective address of the memory may be taken directly from one of the base register or index register specified by instruction. If register is SI, DI and BX then DS is by default segment register
  - If BP is used, then SS is by default segment register

    Example:

    MOV CX, [BX]

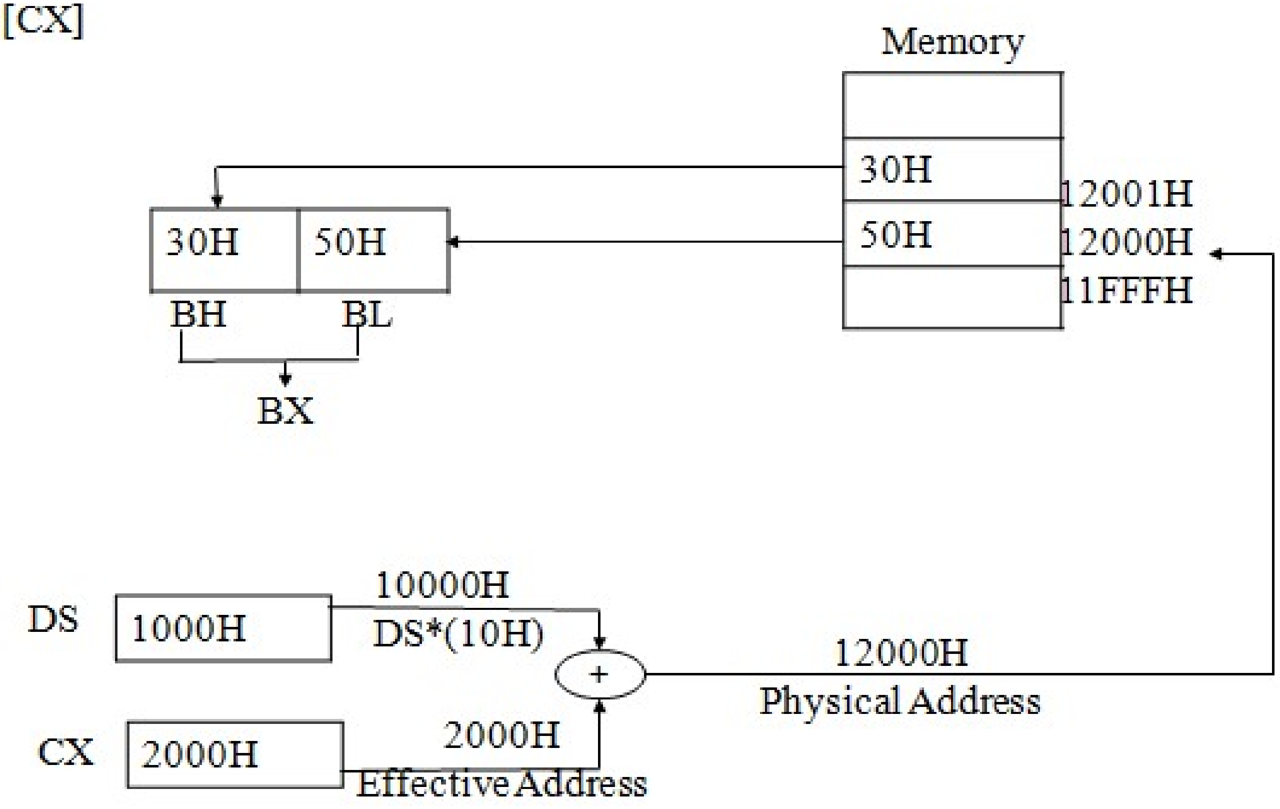
    This instruction moves a word from the address pointed by BX and BX + 1 in data segment into CL and CH respectively.

    CL ← DS: [BX] and CH ← DS: [BX + 1]

Addressing Modes of 8086

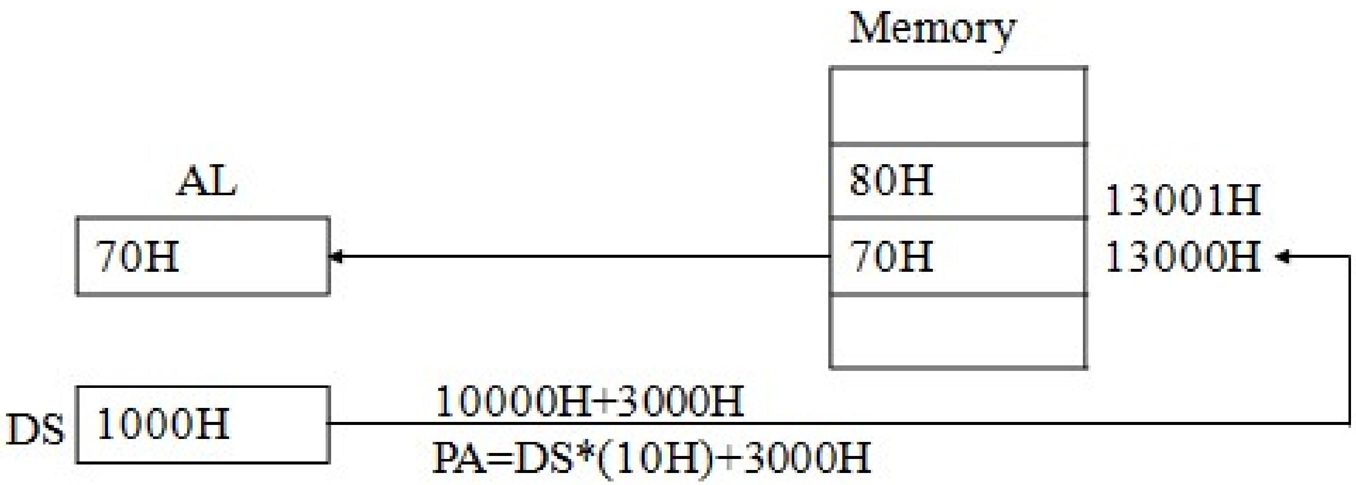
- **Register Indirect addressing mode**

Example:
MOV BX, [CX]



Addressing Modes of 8086

- **Direct (Absolute) addressing mode**
  - In this mode, address of the operand is directly specified in the instruction. Here only the offset address is specified, the segment being indicated by the instruction.
  - In this mode, the 16-bit effective address (EA) is taken directly from the displacement field of the instruction
  - Example: MOV AL, [3000H]

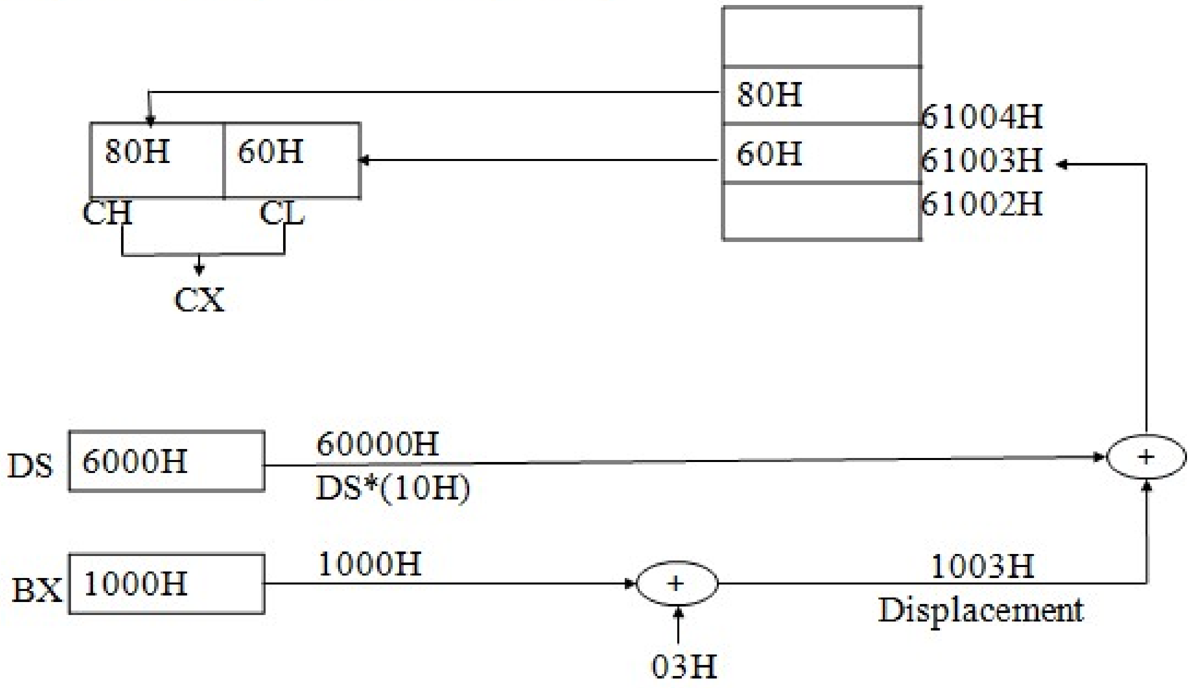


Addressing Modes of 8086

- **Based addressing mode**
    - In this mode, the operand address is calculated using one of the base registers and an 8 bit or a 16 bit displacement.

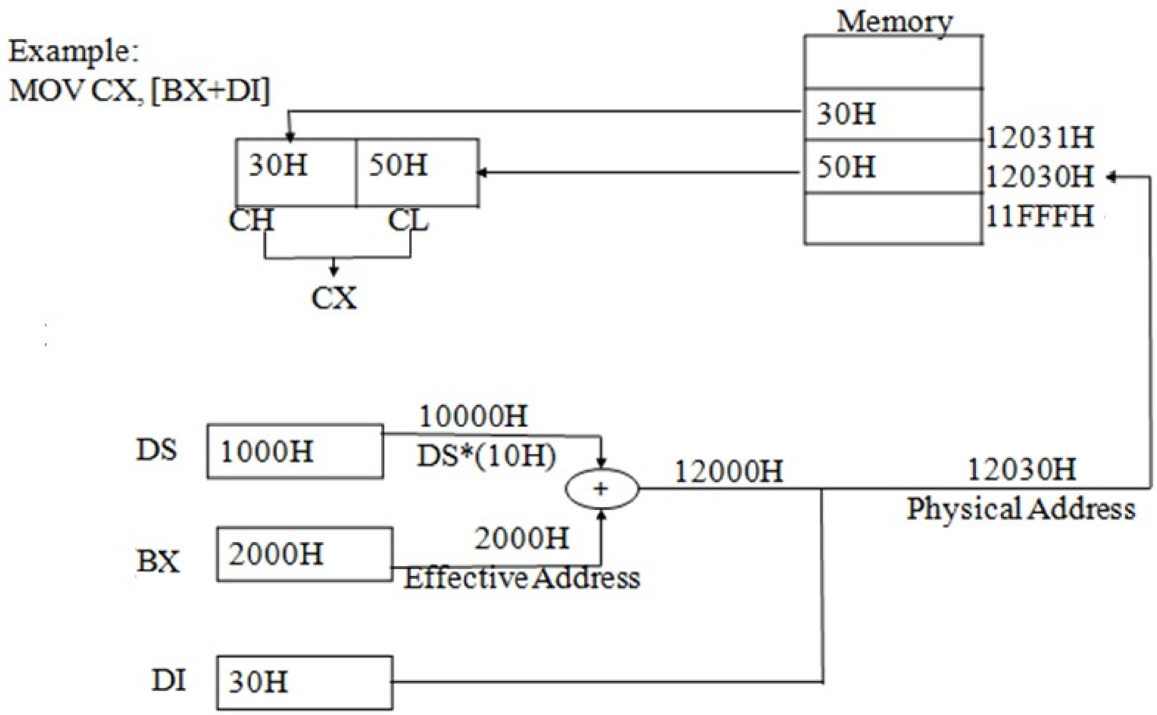
Example:

MOV CX,[BX+0003H] or MOV CX,[BX+3]



Addressing Modes of 8086

- **Indexed addressing mode**
  - Here, operand address is calculated as base register plus an index register .

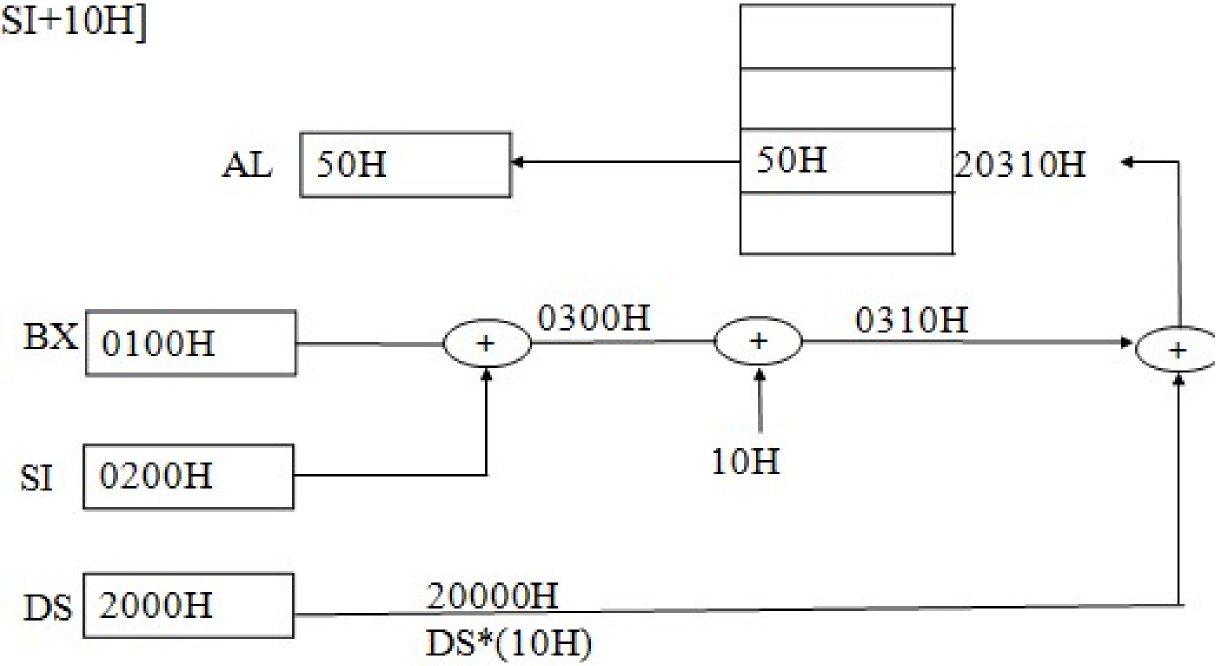


Addressing Modes of 8086

- **Based Indexed addressing mode**
  - In this mode, the address of the operand is calculated as the sum of base register, index register and 8 bit or 16 bit displacement..

Example:

MOV AL,[BX+SI+10H]



AL
50H
50H
20310H
BX
0100H
+
0300H
+
0310H
+
10H
SI
0200H
DS
2000H
20000H
DS*(10H)