Processor –

- **General Registers**
- **Control Registers**
- **Segment Registers**

General Registers

- **Data Registers**
- **Pointer Registers**
- **Index Registers**

General Purpose

- **Ax**(Accumulator): It is used in arithmetic, logic and data Transfer instructions in 8086 microprocessors. In manipulation and division, one of the numbers involved must be in AX or AL.
- **BX**(Base): It usually contains a data pointer used for based, based indexed or register indirect addressing.
- **CX**(Count):Program loop counter,string manipulation counter, shift/Rotate instruction counter
- **DX**(data): Data register can be used as a port number in I/O operations. It is also used in multiplication and division.

Pointer registers:

- **IP**(Instruction Pointer)-The 16-bit IP register stores the offset address Of the next instruction to be executed.IP in association with the CS register Gives the complete address of the current instruction in the code segment
- **SP**(stack pointer): . It points to the topmost item of the stack. If the stack is Empty the stack pointer will be (FFFE)H. Its offset address is relative to the stack segment.
- **BP**(Base Pointer): This is base pointer register pointing to data In stack segment It is primarily used in accessing parameters Passed by the stack. Its offset address Is relative to the stack segment.

Index registers:

- **SI**(source index): This is source index register which is used To point to memory locations in the data segment addressed by DS Its offset is relative to the data segment.
- **DI**(Destination Index): This is destination index register performs the same function as SI.There is a class of instructions called string operations, that use DI to access the memory locations addressed by ES.

Segment Registers:

- **Code Segment :** It contains all the instruction to be executed. A 16 bit code segment Register for CS register stores the starting Address of the code segment.
- **Data Segment :** It contains data, constants and work areas. A 16-bit data segment .Register or DS register stores the starting address of the data segment.
- **Stack Segment:** It contains data and return addresses of procedures or subroutines. It is implemented as stack data structure.

Flag/control register:

- **Cary Flag :** this flag is set to **1** when there is an **unsigned overflow**. For example when you add bytes **255 + 1** When there is no overflow this flag is set to **0**.
- **Parity flag:** this flag is set to **1** when there is even number of one bits in result, and to **0** when there is odd number of one bits.
- **Auxiliary Flag: 1** when there is an **unsigned overflow** for low nibble
- **Zero flag**: **1** when result is **zero**.For none zero result this flag is set to **0**.
- **Sign flag**: **1** when result is **negative**.When result is **positive** it is set to **0**. Actually this flag take the value of the most significant bit.
- **Trap flag**: Used for on-chip debugging.
- **Interrupt enable flag**- when this flag is **1** CPU reacts to interrupts from external devices.
- **Direction flag**: this flag is used by some instructions to process data chains, When this flag is set to **0** - the processing is done forward, when this flag is set to **1** the processing is done backward.
- **Overflow flag** : set to **1** when there is a **signed overflow**. For example, when you add bytes **100 + 50**