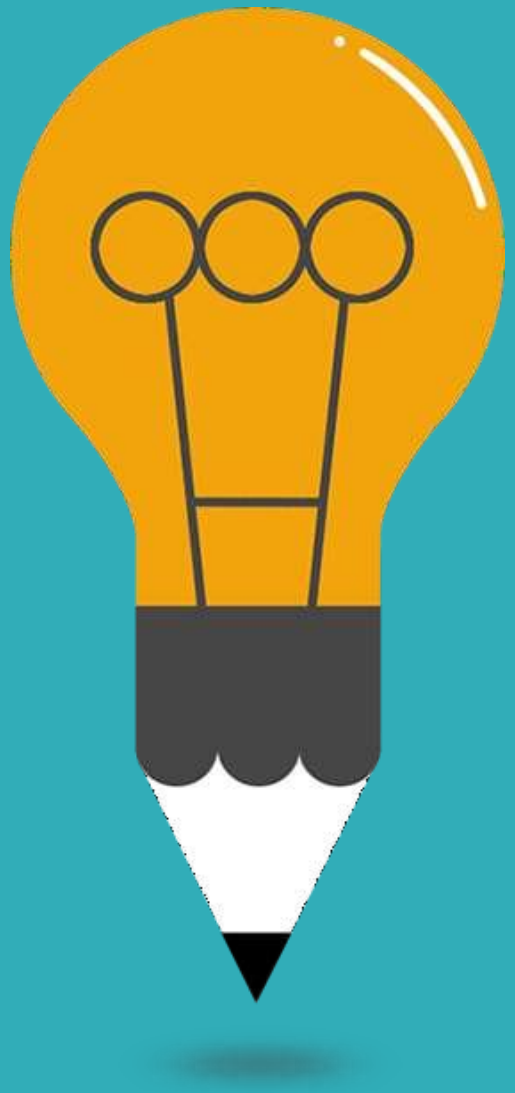
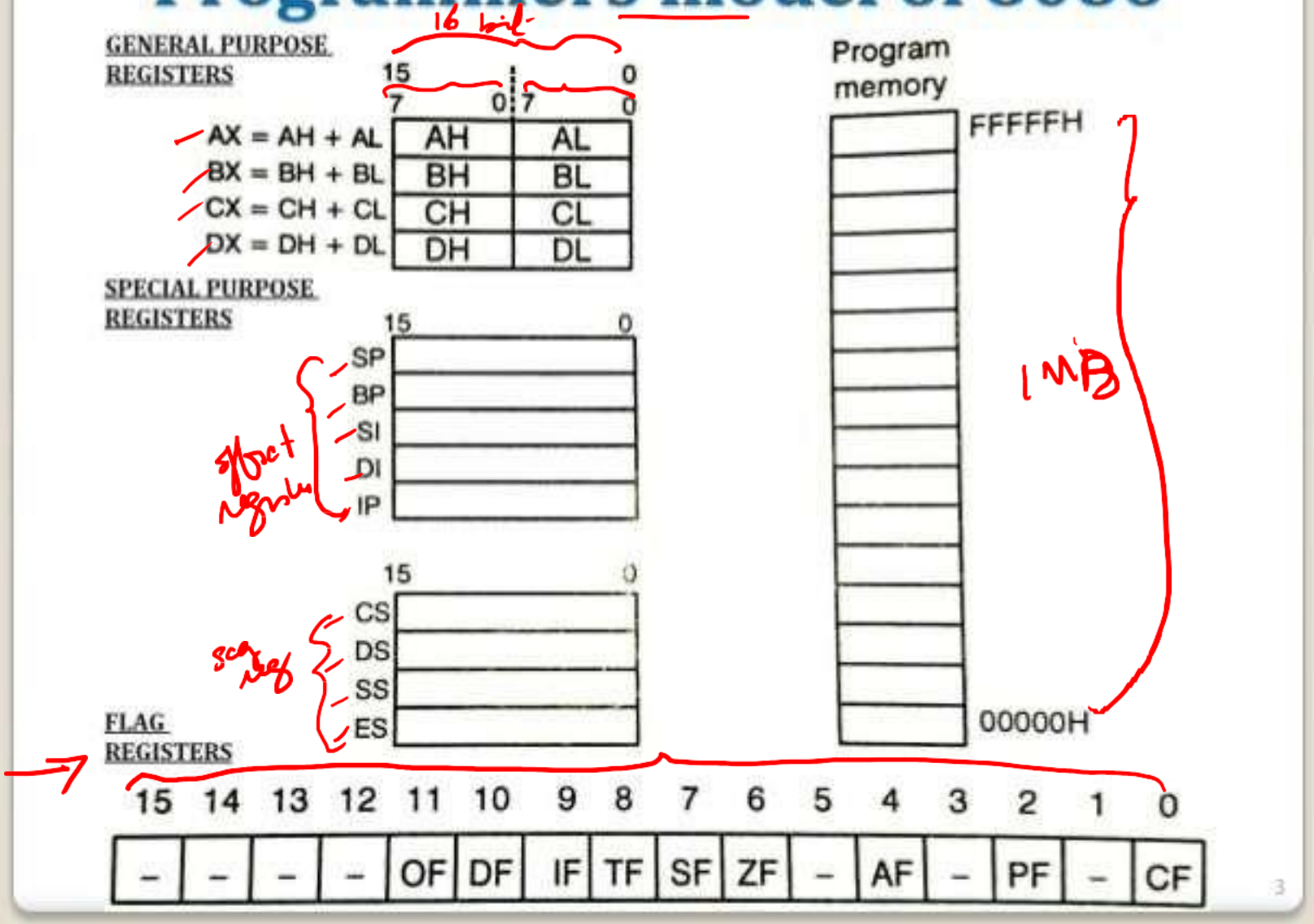


MICROPROCESSORS

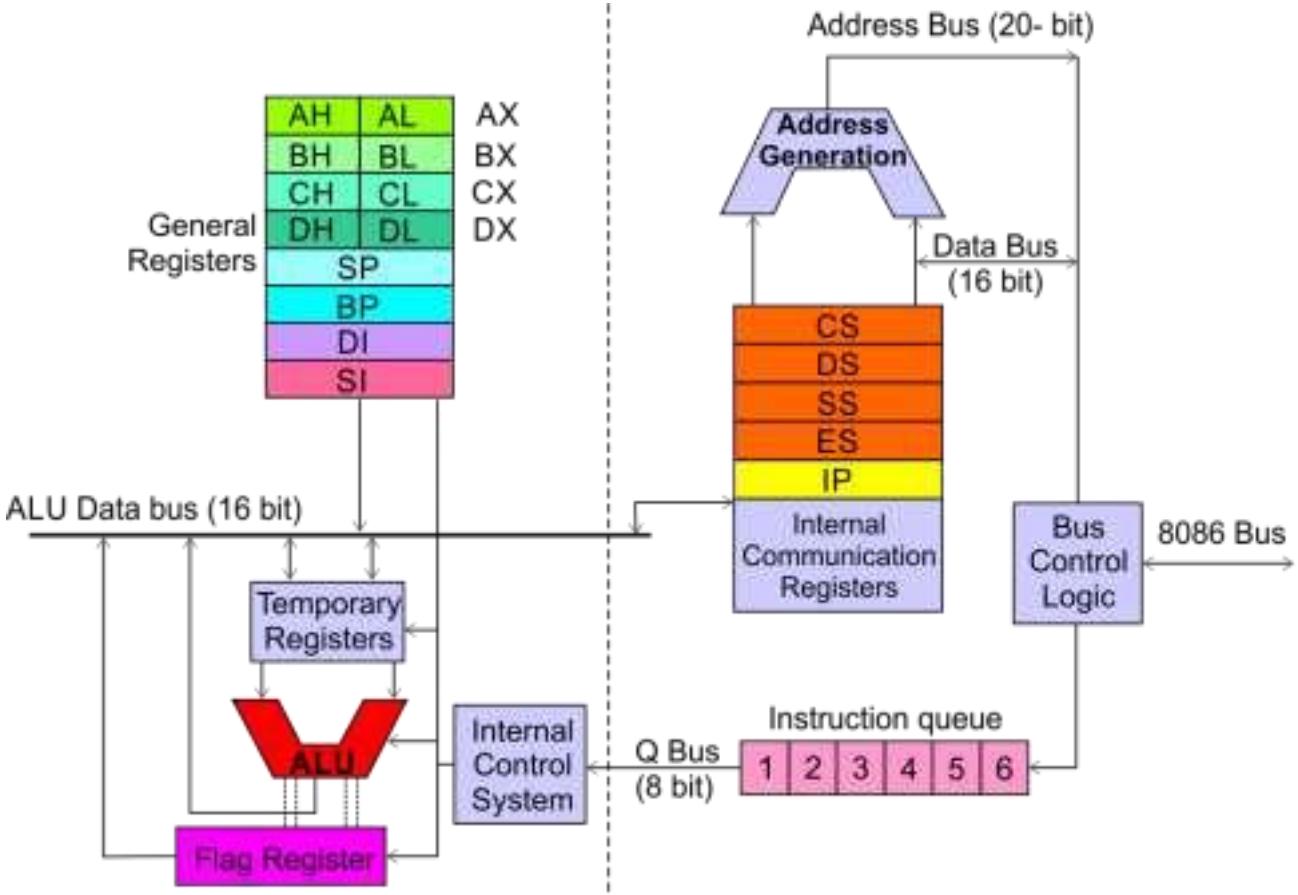
8086 programmers model

PROF.POONAM PANGARKAR, ASSISTANT PROFESSOR , CSE-DATA SCIENCE ,APST THANE

Programmers model of 8086



Architecture



Execution Unit (EU)

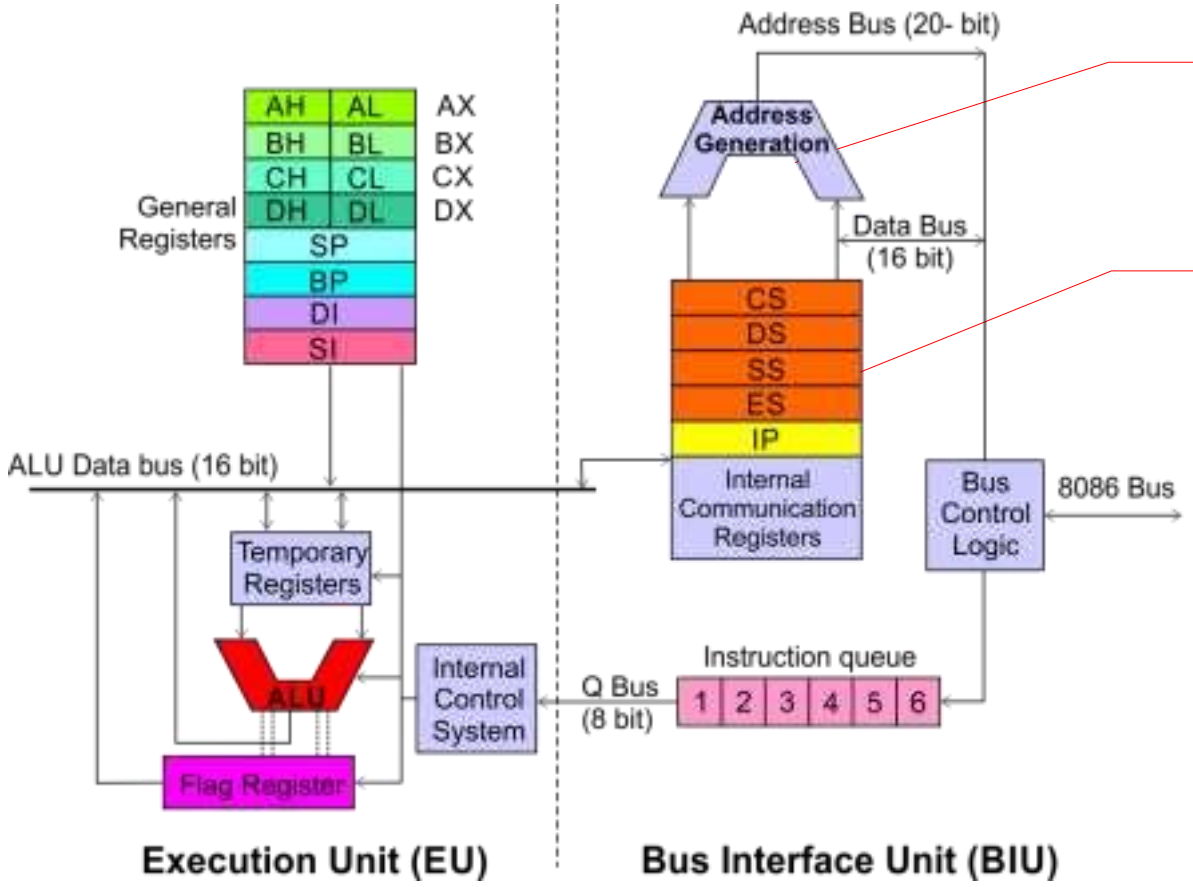
EU executes instructions that have already been fetched by the BIU.

BIU and EU functions separately.

Bus Interface Unit (BIU)

BIU fetches instructions, reads data from memory and I/O ports, writes data to memory and I/ O ports.

Architecture



Bus Interface Unit (BIU)

Dedicated Adder to generate 20 bit address

Four 16-bit segment registers

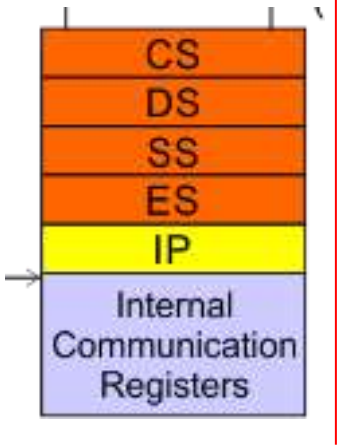
Code Segment (CS)
Data Segment (DS)
Stack Segment (SS)
Extra Segment (ES)

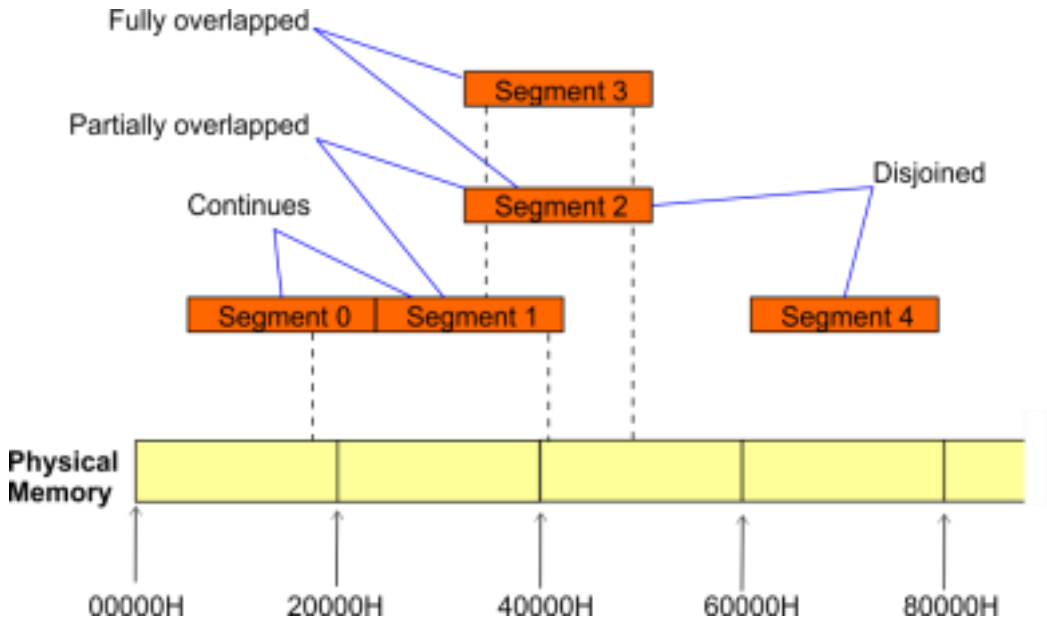
Segment Registers >>

Architecture

Bus Interface Unit (BIU)

Segment Registers





- **8086's 1-megabyte memory is divided into segments of up to 64K bytes each.**
- **The 8086 can directly address four segments (256 K bytes within the 1 M byte of memory) at a particular time.**
- **Programs obtain access to code and data in the segments by changing the segment register content to point to the desired segments.**

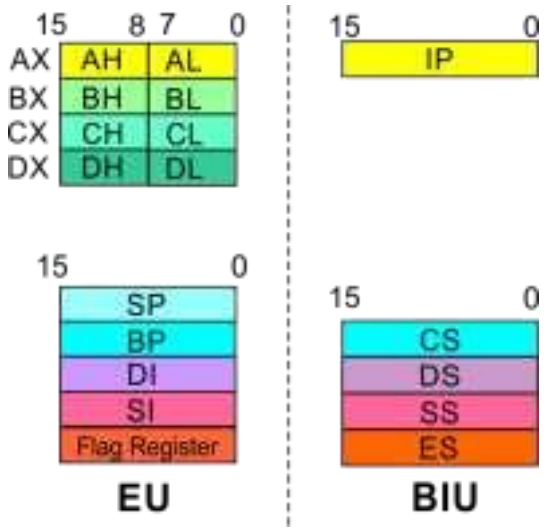
Architecture

Bus Interface Unit (BIU)

Segment Registers

Code Segment Register

- **16-bit**
- **CS contains the base or start of the current code segment; IP contains the distance or offset from this address to the next instruction byte to be fetched.**
- **BIU computes the 20-bit physical address by logically shifting the contents of CS 4-bits to the left and then adding the 16-bit contents of IP.**
- **That is, all instructions of a program are relative to the co ntents of the CS register multiplied by 16 and then offset i s added provided by the IP.**



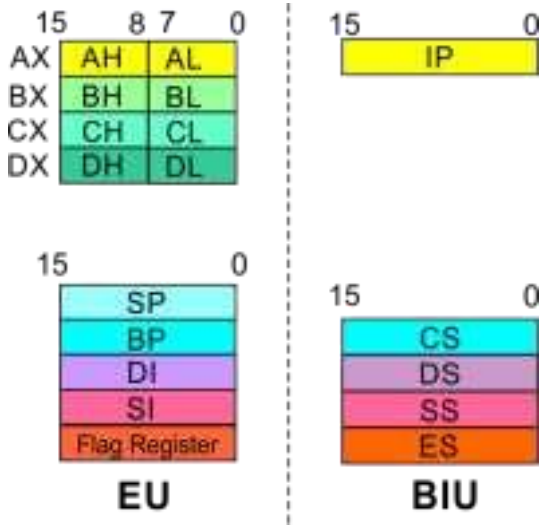
Architecture

Bus Interface Unit (BIU)

Segment Registers

Data Segment Register

- **16-bit**
- **Points to the current data segment; operands for most instructions are fetched from this segment.**
- **The 16-bit contents of the Source Index (SI) or Destination Index (DI) or a 16-bit displacement are used as offset for computing the 20-bit physical address.**

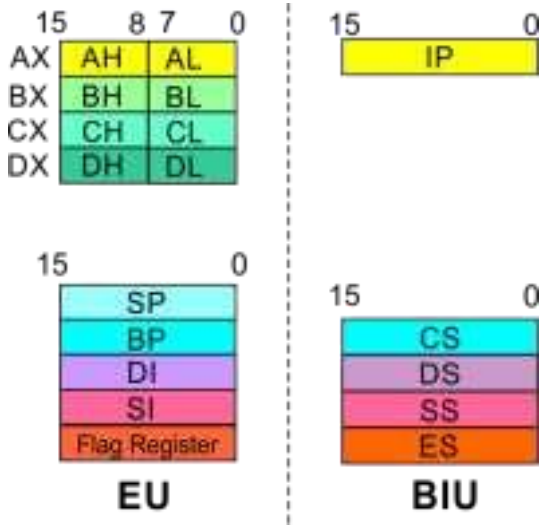


Architecture

Segment Registers

Stack Segment Register

- **16-bit**
- **Points to the current stack.**
- **The 20-bit physical stack address is calculated from the Stack Segment (SS) and the Stack Pointer (SP) for stack instructions such as PUSH and POP.**
- **In based addressing mode, the 20-bit physical stack address is calculated from the Stack segment (SS) and the Base Pointer (BP).**

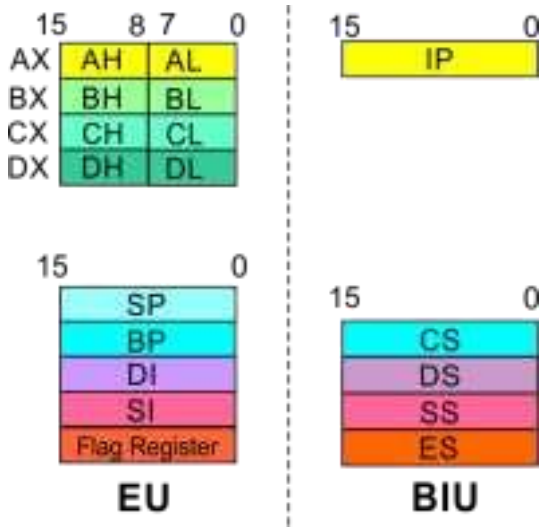


Architecture

Segment Registers

Extra Segment Register

- **16-bit**
- **Points to the extra segment in which data (in excess of 64 K pointed to by the DS) is stored.**
- **String instructions use the ES and DI to determine the 20-bit physical address for the destination.**



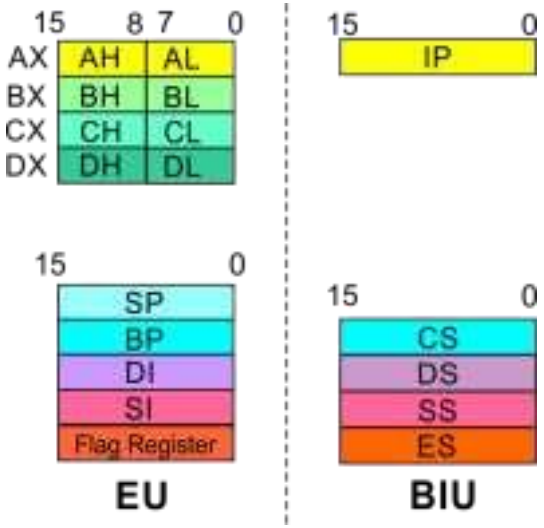
Architecture

Bus Interface Unit (BIU)

Segment Registers

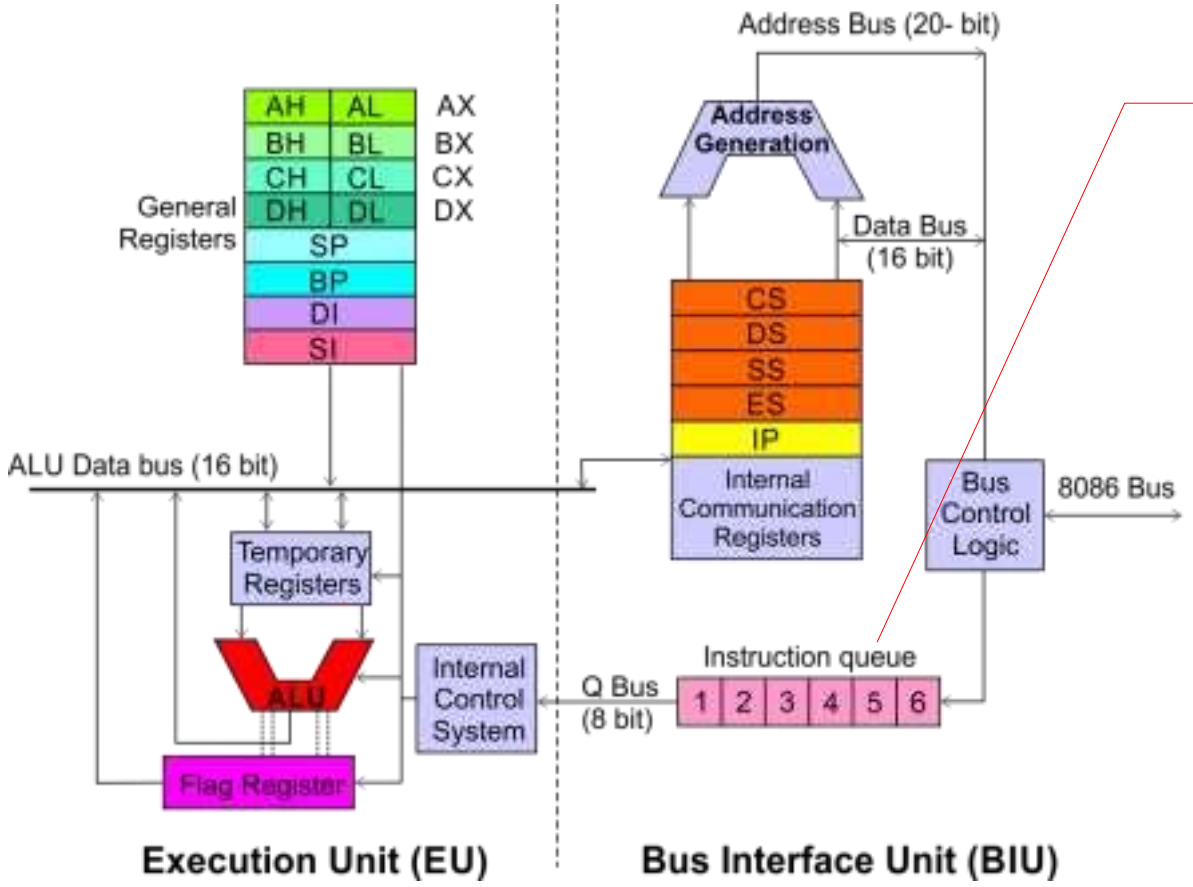
Instruction Pointer

- **16-bit**
- **Always points to the next instruction to be executed within the currently executing code segment.**
- **So, this register contains the 16-bit offset address pointing to the next instruction code within the 64Kb of the code segment area.**
- **Its content is automatically incremented as the execution of the next instruction takes place.**



Architecture

Bus Interface Unit (BIU)



Instruction queue

- **A group of First-In-First-Out (FIFO) in which up to 6 bytes of instruction code are pre fetched from the memory ahead of time.**
- **This is done in order to speed up the execution by overlapping instruction fetch with execution.**
- **This mechanism is known as pipelining.**

Architecture

EU decodes and executes instructions.

A decoder in the EU control system translates instructions.

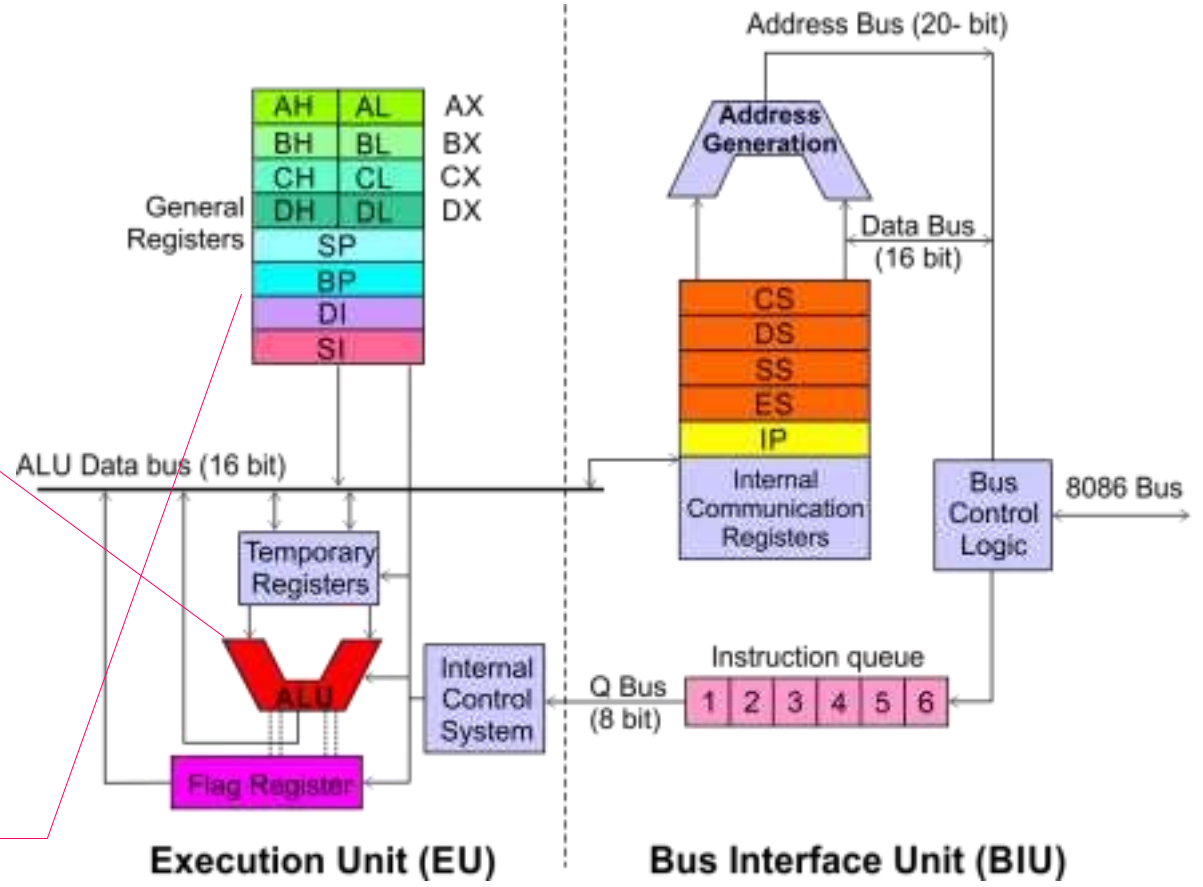
16-bit ALU for performing arithmetic and logic operation

Four general purpose registers(AX, BX, CX, DX);

Pointer registers (Stack Pointer, Base Pointer);

and

Index registers (Source Index, Destination Index) each of 16-bits



Some of the 16 bit registers can be used as two 8 bit registers as :

AX can be used as AH and AL
BX can be used as BH and BL
CX can be used as CH and CL
DX can be used as DH and DL