University of Diyala
College of engineering
Department of computer Engineering
Second class

microprocessor Programming

Lecture 2

8086 microprocessor

Presented by

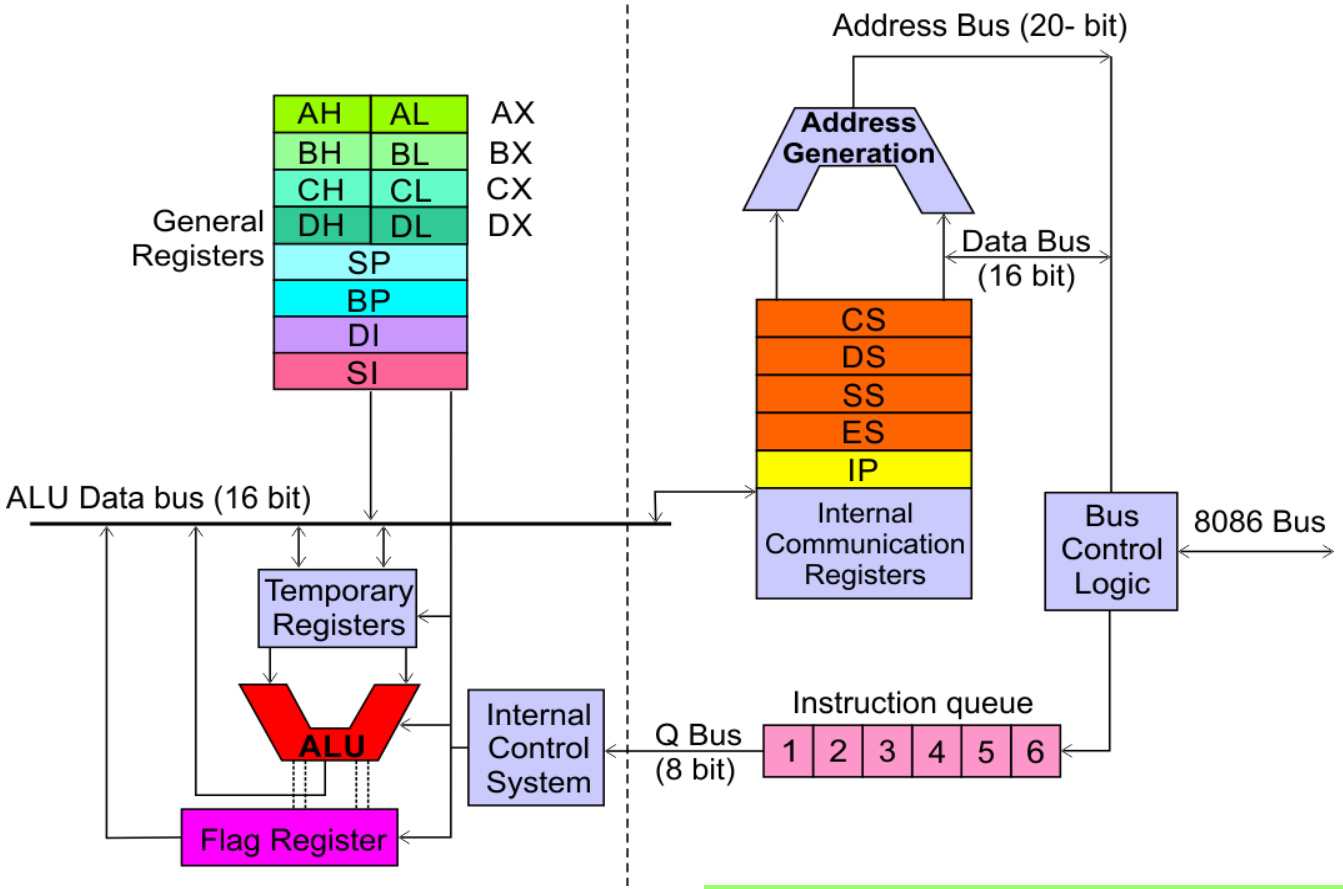
Lecturer :Abdullah Thair Abdalsatir
Department of computer Engineering
University of Diyala

Lecture 2

8086 microprocessor

- ❖ Internal Architecture
- ❖ BUS interface unit BIU
- Instruction Pointer(IP)
- Code Segment Register (CS)
- Data Segment Register (DS)
- Stack Segment Register(SS)
- Extra Segment Register(ES)

Internal Architecture



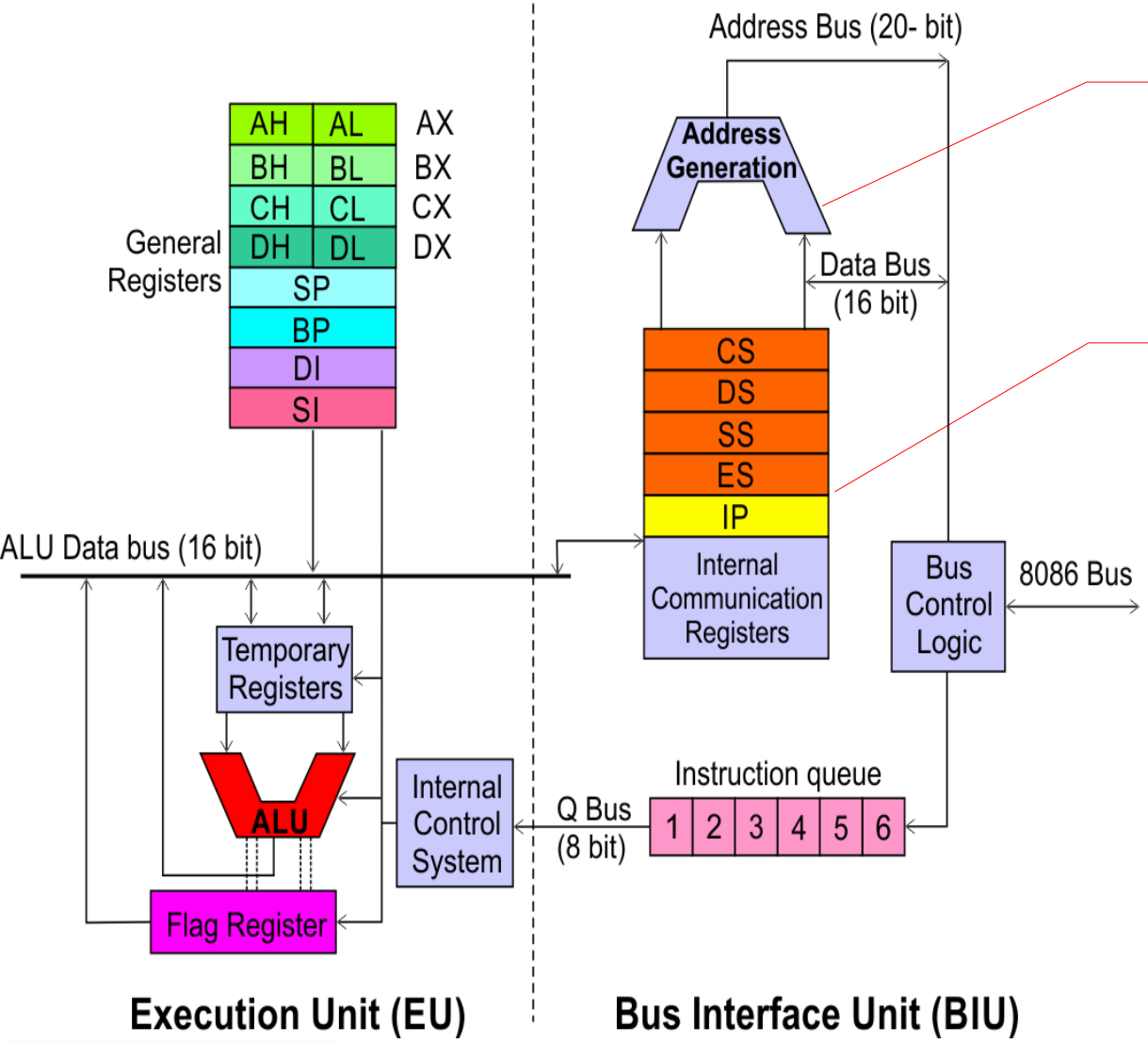
Execution Unit (EU)

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)

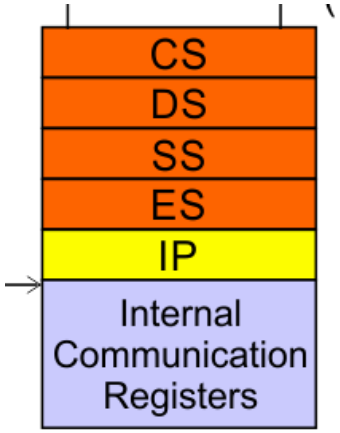


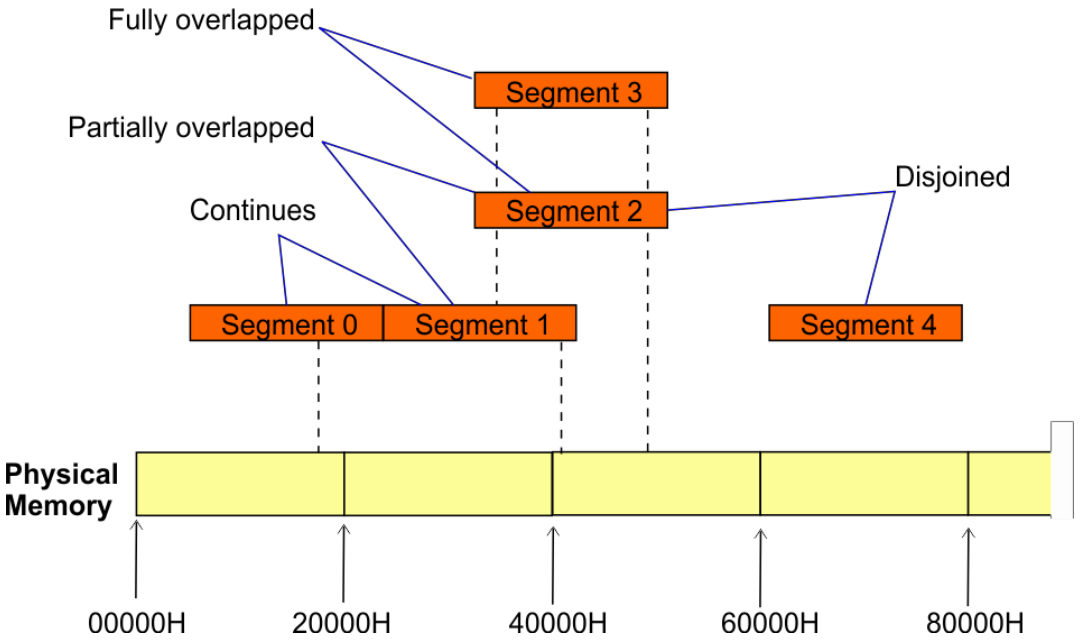
Dedicated Adder to generate 20-bit address

Four 16-bit segment registers

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

Segment Registers



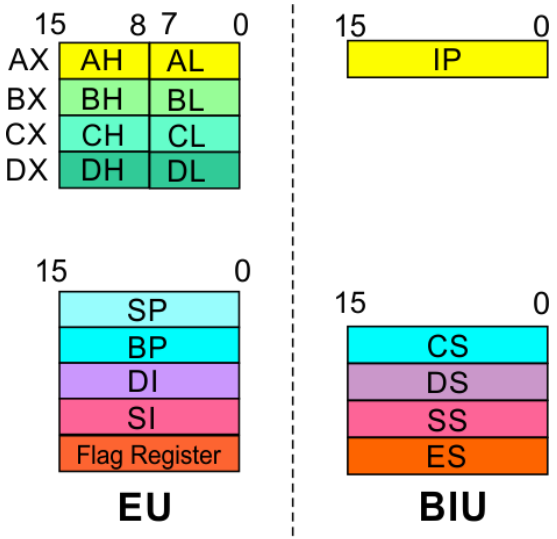


- 8086's 1-megabyte memory is divided into segments of up to 64K bytes each.
- The 8086 can directly address four segments (256 K bytes within the 1 M byte of memory) at a particular time.
- Programs obtain access to code and data in the segments by changing the segment register content to point to the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

Segment Registers

Instruction Pointer (IP)

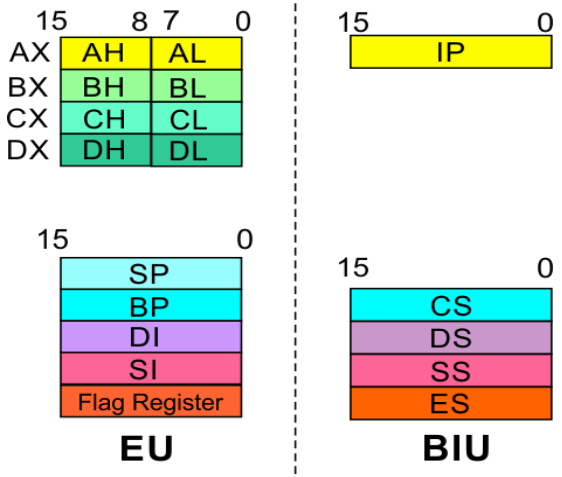
- 16-bit
- Always points to the next instruction to be executed within the currently executing code segment.
- So, this register contains the 16-bit offset address pointing to the next instruction code within the 64Kb of the code segment area.
- Its content is automatically incremented as the execution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



Segment Registers

Code Segment Register (CS)

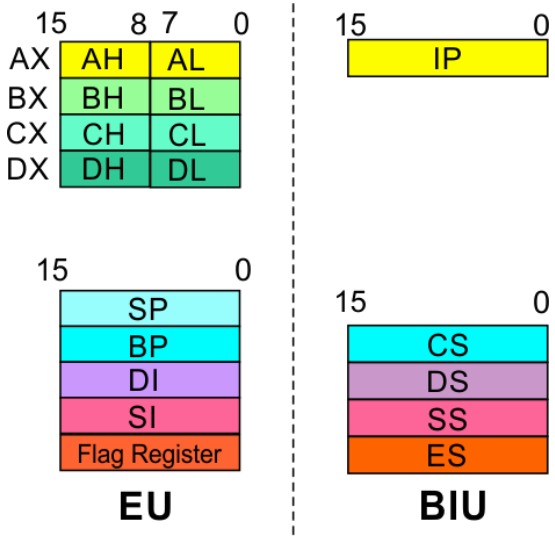
- 16-bit
- CS contains the base or start of the current code segment; IP contains the distance or offset from this address to the next instruction byte to be fetched.
- BIU computes the 20-bit physical address by logically shifting the contents of CS 4-bits to the left and then adding the 16-bit contents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



Segment Registers

Data Segment Register (DS)

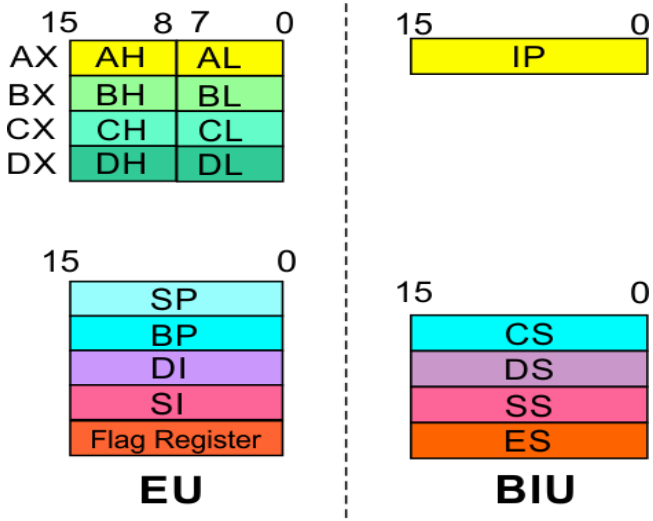
- 16-bit
- Points to the current data segment; operands for most instructions are fetched from this segment.
- The 16-bit contents of the Source Index (SI) or Destination Index (DI) or a 16-bit displacement are used as offset for computing the 20-bit physical address.



Segment Registers

Stack Segment Register (SS)

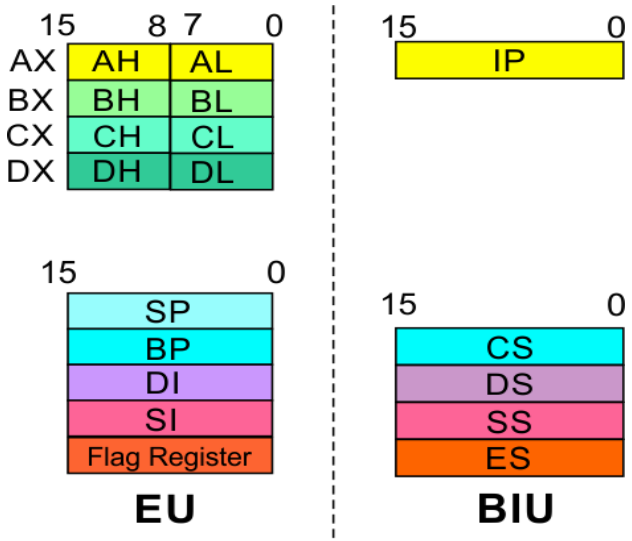
- 16-bit
- Points to the current stack.
- The 20-bit physical stack address is calculated from the Stack Segment (SS) and the Stack Pointer (SP) for stack instructions such as PUSH and POP.
- In based addressing mode, the 20-bit physical stack address is calculated from the Stack segment (SS) and the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

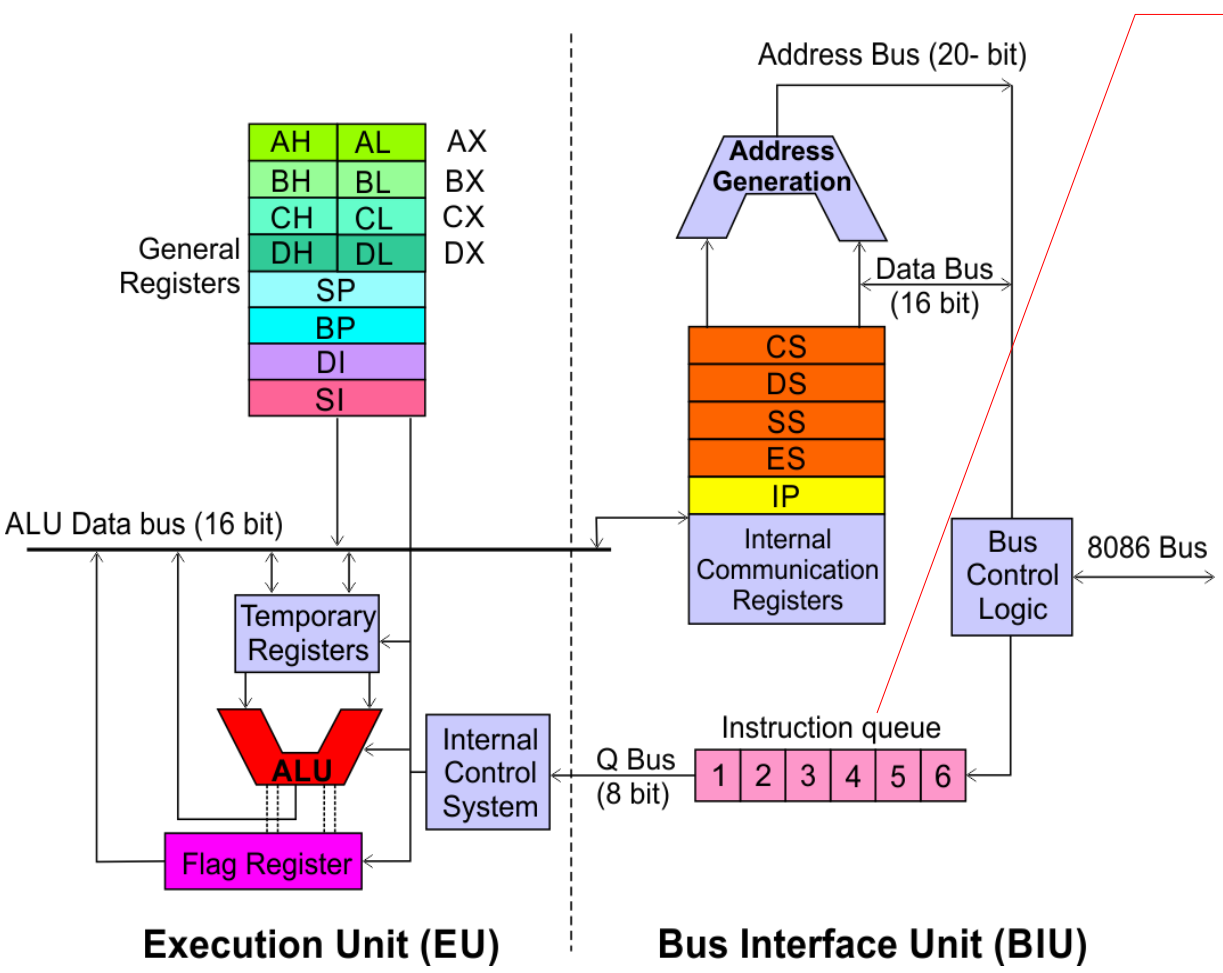


Segment Registers

Extra Segment Register (ES)

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





Instruction

- **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.**

Thank you so much

Any questions ?