8086 CPU ARCHITECTURE:

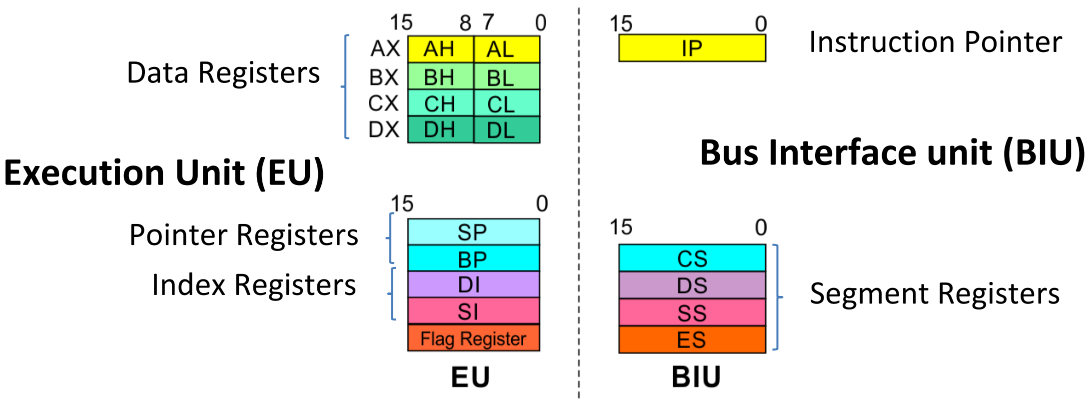
A Microprocessor is an Integrated Circuit with all the functions of a CPU however, it cannot be used stand-alone *because* it has no memory or peripherals. 8086 does not have a RAM or ROM inside it. However, it has internal registers for storing intermediate and final results and interfaces with memory located outside it through the System Bus.

In case of 8086, it is a 16-bit Integer processor in a 40 pin, Dual Inline Packaged IC. The size of the internal registers (present within the chip) indicates how much information the processor can operate on at a time (in this case 16-bit registers) and how it moves data around internally within the chip, sometimes also referred to as the internal data bus.

The microprocessors functions as the CPU in the stored program model of the digital computer. Its job is to generate all system timing signals and synchronize the transfer of data between memory, I/O, and itself. It accomplishes this task via the three-bus system architecture (*Address bus, Data bus, and Control bus*).

The 8086 CPU is organized as two separate processors (or units), called the Bus Interface Unit (BIU) and the Execution Unit (EU). The BIU provides H/W functions, including generation of the memory and I/O addresses for the transfer of data between the outside world - outside the CPU, that is- and the EU.

The EU receives program instruction codes and data from the BIU, executes these instructions, and store the results in the general registers. By passing the data back to the BIU, data can also be stored in a memory location or written to an output device. Note that the EU has no connection to the system buses. It receives and outputs all its data thru the BIU. The basic architecture of 8086 is shown below.



The 8086 Microprocessor has two sections that are EU & BIU

1. The Execution Unit (EU):

The main components of the EU are:

- Control Circuit
- Instruction Register and Instruction Decoder
- ALU
- General Purpose Registers ▪ Special Purpose Registers ▪ Flag/Status Register.

The Control Circuit controls all the operations & flow the data inside the Microprocessor. The Instruction Decoder works to translate or decode the instructions which are fetched from the memory (pre-fetch Stack). After translation it place the instructions in a series to perform the required task.

EU unit performs the following functions:

- It Fetches instructions from the Stack in BIU, and decodes it.
- It performs the logic & arithmetic operation on memory or register, using the ALU.
- It stores the information temporary in the register array.
- Sends request signals to the BIU to access the external module.

General Purpose Registers

Registers are a type of computer memory used to quickly accept, store, and transfer data and instructions that are being used immediately by the CPU. The registers used by the CPU are often termed as Processor registers. The computer needs processor registers *(why)* for processing data and a register for holding a memory address.

The register holding the memory location is used to calculate the address of the next instruction after the execution of the current instruction is completed. Following is the list of some of the most common registers used in a basic computer:

A. **Data Registers**
B. **Pointer Registers**
C. **Index Registers**

A- Data Registers include:

I. Accumulator Register (AX):

Consists of two 8-bit registers AL and AH, which can be combined together and used as a 16-bit register AX. AL in this case contains the low order byte of the word, and AH contains the high-order byte. The I/O instructions use the AX or AL for inputting / outputting 16 or 8-bit data to or from an I/O port. It holds operands and results during multiplication, division operations and logical (AND, OR, NOT, compare, etc.) functions. Also, an accumulator during String operations. AL register is also called accumulator because it has some characteristics different from other general-purpose registers.

II. Base Register (BX):

Consists of two 8-bit registers BL and BH, which can be combined together and used as a 16-bit register BX. BL in this case contains the low-order byte of the word, and BH contains the high-order byte. This is the only general purpose register whose contents can be used for addressing the 8086 memory. All memory references utilizing this register content for addressing use DS as the default segment register.

III. Counter Register (CX):

Consists of two 8-bit registers CL and CH, which can be combined together and used as a 16-bit register CX. When combined, CL register contains the low order byte of the word, and CH contains the high-order byte. Instructions such as SHIFT, ROTATE and LOOP use the contents of CX as a counter.

IV. Data Register (DX):

Consists of two 8-bit registers DL and DH, which can be combined together and used as a 16-bit register DX. When combined, DL register contains the low order byte of the word, and DH contains the high-order byte. Used to hold the high 16-bit result (data) in 16 X 16 multiplication or the high 16-bit dividend (data) before a 32 ÷ 16 division and the 16-bit reminder after division.

B- Pointer Registers include:

The 8086 has four other general-purpose registers, two-pointer registers SP and BP, and two index registers DI and SI. These are used to store what is called offset addresses. These registers are unlike the general-purpose data registers, the pointer and index registers are only accessed as words (16 bits). *Pointer Registers include:*

I. SP (Stack Pointer):

Points to Stack top. Stack is in Stack Segment, used during instructions like PUSH, POP, CALL, RET etc. It is used with SS to access the stack segment.

II. BP (Base Pointer):

This is base pointer register pointing to data in Stack Segment. Unlike SP, we can use BP to access data in the other segments. BP can hold offset address of any location in the stack segment. It is used to access random locations of the stack.

C- Index Registers include:

These types of registers useful for doing *vector/array operations*. These also used to reduce the amount of memory used and increased execution speed.

I. SI (Source Index):

This is source index register which is used to point to memory locations in the Data Segment addressed by DS. Thus, when we increment the contents of SI, we can easily access consecutive memory locations. It holds offset address in Data Segment during string operations.

II. DI (Destination Index):

This is destination index register performs the same function as SI but in *Extra Segment ES* not in Data Segment DS. It holds offset address in Data Segment during string operations.

A special purpose register:

Some registers serve specific functions within the CPU. Several of the more important of these registers are Instruction Register and Program Counter:

Instruction Register

The instruction register (IR) or current instruction register (CIR) is used to store the *current instruction* to be executed or decoded, that having been fetched from memory (pre-fetch). IR or CIR is the part of a *CPU's control circuit*.

Decoding the op-code in the instruction register *includes:* determining the instruction, determining where its operands are in memory, retrieving the operands from memory, allocating the processor to execute the command (in superscalar processors), etc.

Program Counter

The program counter also *called* the instruction pointer (IP), which holds the address of the *following instruction* to be executed. Thus, the program counter is responsible for the following process, for the purpose of saving time.