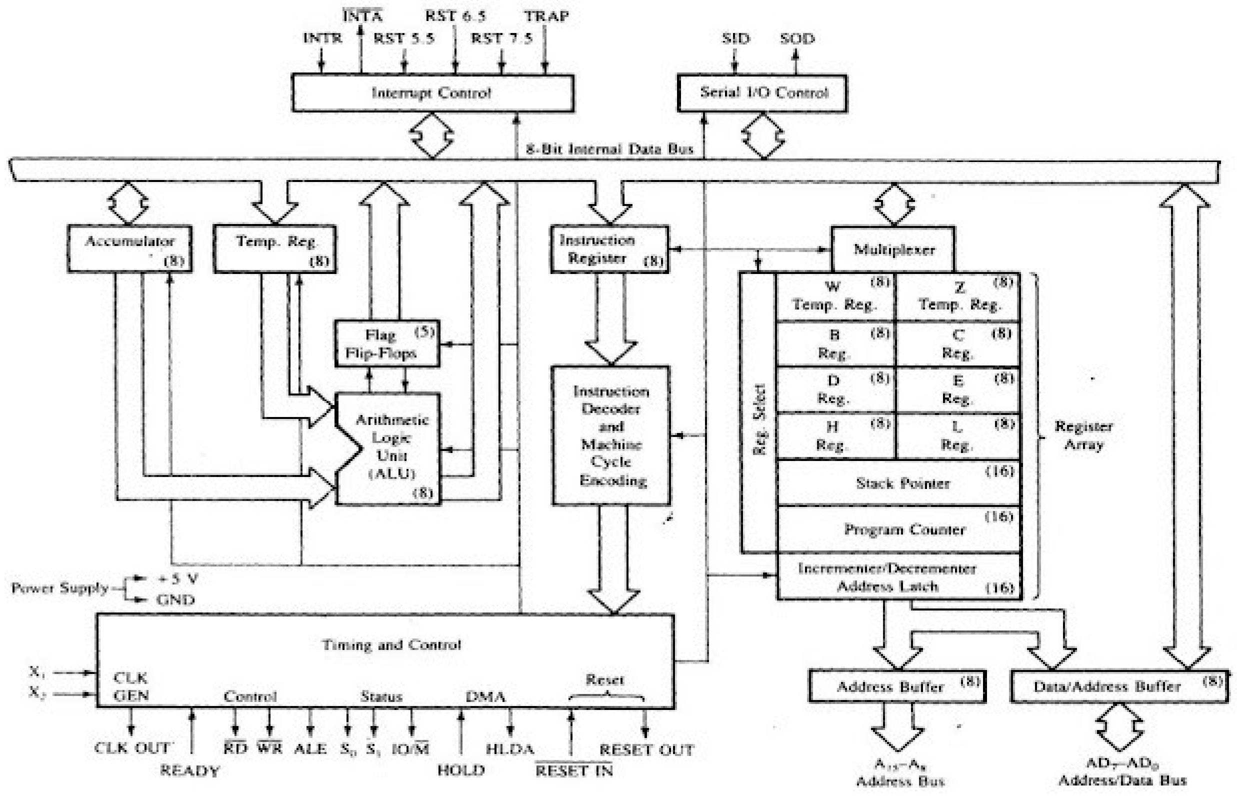


6/

Fig: Functional Block Diagram of 8085 Microprocessor

1. ALU

- The ALU performs the actual numerical and logic operation such as ‘add’, ‘subtract’, ‘AND’, ‘OR’ etc.
- Uses data from memory and from Accumulator to perform arithmetic operation and always stores result of operation in Accumulator.
- The ALU consists of accumulator, flag register and temporary register.

a. Accumulator

- The accumulator is an 8-bit register that is a part of arithmetic/logic unit (ALU). This register is used to store 8-bit data and to perform arithmetic and logical operations. The result of an operation is stored in the accumulator.
- The accumulator is also identified as register A.

b. Flag register

- 8085 has 8-bit flag register. There are only 5 active flags.

S	Z		AC		P		CY

Fig: 8085 flag register

- Flags are flip-flops which are used to indicate the status of the accumulator and other register after the completion of operation.
- These flip-flops are set or reset according to the data condition of the result in the accumulator and other registers.

i. Sign flag(S):

- Sign flag indicates whether the result of a mathematical or logical operation is negative or positive.
- If the result is negative, this flag will be set (i.e. S=1) and if the result is positive, the flag will be reset (i.e. S=0).

ii. Zero flag (Z):

- Zero flag indicates whether the result of a mathematical or logical operation is zero or not.
- If the result of current operation is zero, the flag will be set (i.e. Z=1) otherwise the flag will be reset (Z=0).
- This flag will be modified by the result in the accumulator as well as in the other register.

iii. Auxiliary carry flag (AC):

- In operation when a carry is generated by bit D3 and passes on to bit D4, the AC flag will be set otherwise AC flag will be reset.
- This flag is used only internally for BCD operation and is not available for the programmer to change the sequence of program with the jump instruction.

iv. Parity flag (P):

- This flag indicates whether the current result is of even parity (no. of 1's is even) or odd parity (no. of 1's is odd).
- If even parity, P flag will be set otherwise reset.

v. Carry flag (CY):

- This flag indicates whether during an addition or subtraction operation carry or borrow is generated or not.
- If carry or borrow is generated, the flag will be set otherwise reset.

2. Timing and control unit

- This unit produces all the timing and control signal for all the operation.
- This unit synchronizes all the MP operations with the clock and generates the control signals necessary for communication between the MP and peripherals.

3. Instruction register and decoder

- The instruction register and decoder are part of ALU. When an instruction is fetched from memory, it is loaded in the instruction register.
- The decoder decodes the instruction and establishes the sequence of events to follow.
- The IR is not programmable and cannot be accessed through any instruction.

4. Register array

- The register unit of 8085 consists of

-Six general-purpose data registers B,C,D,E,H,L
-Two internal registers W and Z
-Two 16-bit address registers PC (program counter) and SP (stack pointer)
-One increment/decrement counter register
-And, one multiplexer (MUX)

- The six general-purpose registers are used to store 8-bit data. They can be combined as register pairs BC, DE, and HL to perform some 16-bit operations.
- The two internal registers W and Z are used to hold 8-bit data during the execution of some instructions, CALL and XCHG instructions.
- SP is 16-bit registers used to point the address of data stored in the stack memory. It always indicates the top of the stack.
- PC is 16-bit register used to point the address of the next instruction to be fetched and executed stored in the memory.

5. System bus
   a. Data bus

- It carries 'data', in binary form, between MP and other external units, such as memory.
- Typical size is 8 or 16 bits.

b. Address bus

- It carries 'address' of operand in binary form.
- Typical size is 16-bit.

c. Control Bus

- Control Bus are various lines which have specific functions for coordinating and controlling MP operations.
- E.g.: Read/Write control line.

6. Interrupt Control

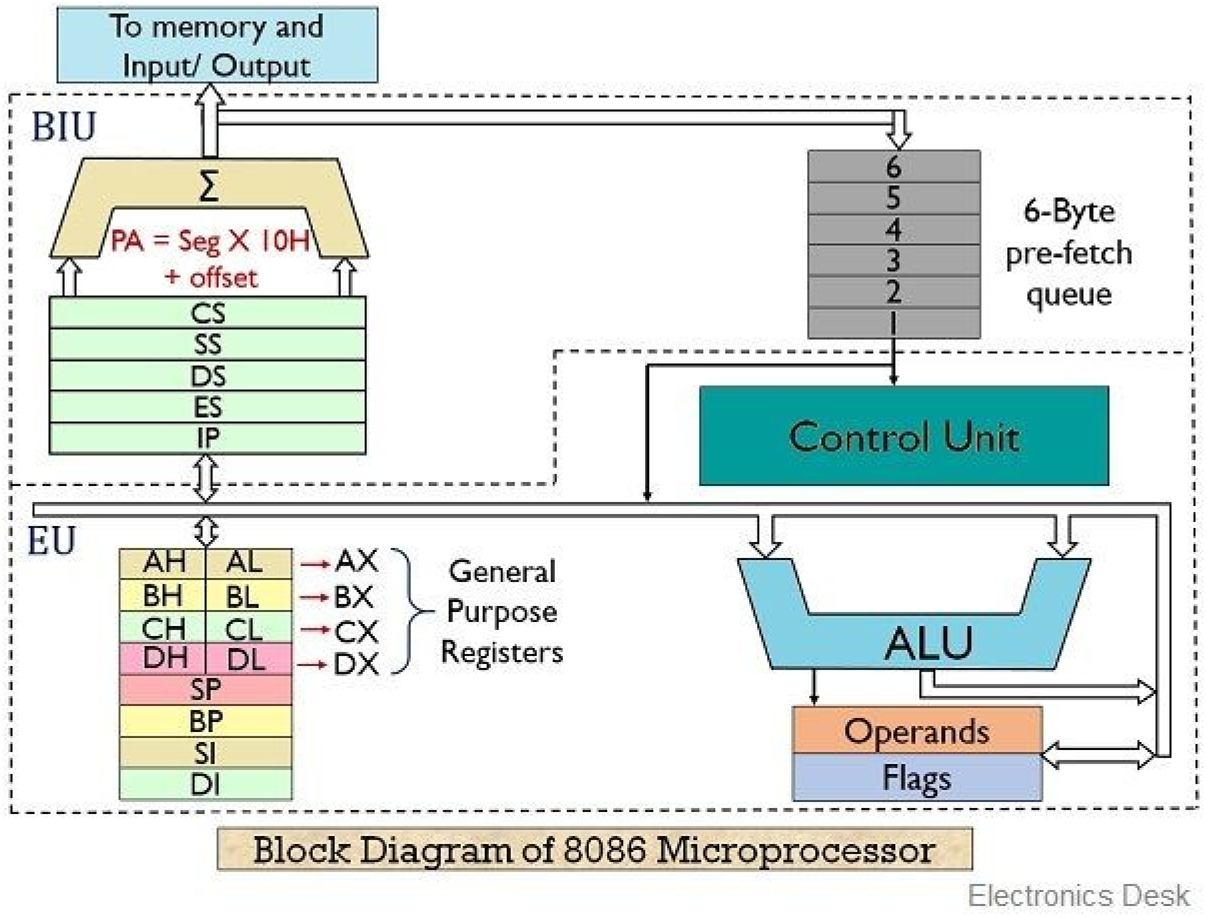
- Interrupt is a signal, which suspends the routine what the MP is doing, brings the control to perform the subroutine, completes it and returns to main routine.
- May be hardware or software interrupts. Some interrupts may be ignored (maskable), some cannot (non-maskable).
- E.g. INTR, TRAP, RST 7.5, RST 6.5, RST 5.5

7. Serial I/O Control

- The MP performs serial data input or output (one bit at a time). In serial transmission, data bits are sent over a single line, one bit at a time.
- The 8085 has two signals to implement the serial transmission: SID (serial input data) and SOD (serial output data).

Block Diagram of 8086 Microprocessor

The architecture of 8086 microprocessor is composed of 2 major units, the BIU i.e., Bus Interface Unit and EU i.e., Execution Unit. The figure below shows the block diagram of the architectural representation of the 8086 microprocessor:



Bus Interface Unit (BIU)

The Bus Interface Unit (BIU) manages the data, address and control buses.
The BIU functions in such a way that it:
●Fetches the sequenced instruction from the memory,
●Finds the physical address of that location in the memory where the instruction is stored and
●Manages the 6-byte pre-fetch queue where the pipelined instructions are stored.
An 8086 microprocessor exhibits the property of pipelining the instructions in a queue while performing decoding and execution of the previous instruction. This saves the processor time of operation by a large amount. This pipelining is done in a **6-byte queue**. Also, the BIU contains **4 segment registers**. Each segment register is

16-bit. The segments are present in the memory and these registers hold the address of all the segments. These registers are as follows:
1.Code segment register: It is a 16-bit register and holds the address of the instruction or program stored in the code segment of the memory.
Also, the IP in the block diagram is the instruction pointer which is a default register that is used by the processor in order to get the desired instruction. The **IP contains the offset address** of the next byte that is to be taken from the code segment.
2. Stack segment register: The stack segment register provides the starting address of the stack segment in the memory. Like in stack pointer, PUSH and POP operations are used in this segment to give and take the data to/from it.
3. Data segment register: It holds the address of the data segment. The data segment stores the data in the memory whose address is present in this 16-bit register.
4. Extra segment register: Here the starting address of the extra segment is present. This register basically contains the address of the string data.
It is to be noteworthy that the physical address of the instruction is achieved by combining the segment address with that of the offset address.
6-byte pre-fetch queue: This queue is used in 8086 in order to perform pipelining. As at the time of decoding and execution of the instruction in EU, the BIU fetches the sequential upcoming instructions and stores it in this queue.
mory location.
Control Unit:
Like the timing and control unit in 8085 microprocessor, the control unit in 8086 microprocessor produces control signal after decoding the opcode to inform the general purpose register to release the value stored in it. And it also signals the ALU to perform the desired operation.
ALU:
The arithmetic and logic unit carries out the logical tasks according to the signal generated by the CU. The result of the operation is stored in the desired register.
Flag:
Like in 8085, here also the flag register holds the status of the result generated by the ALU. It has several flags that show the different conditions of the result.
Operand:
It is a temporary register and is used by the processor to hold the temporary values at the time of operation.

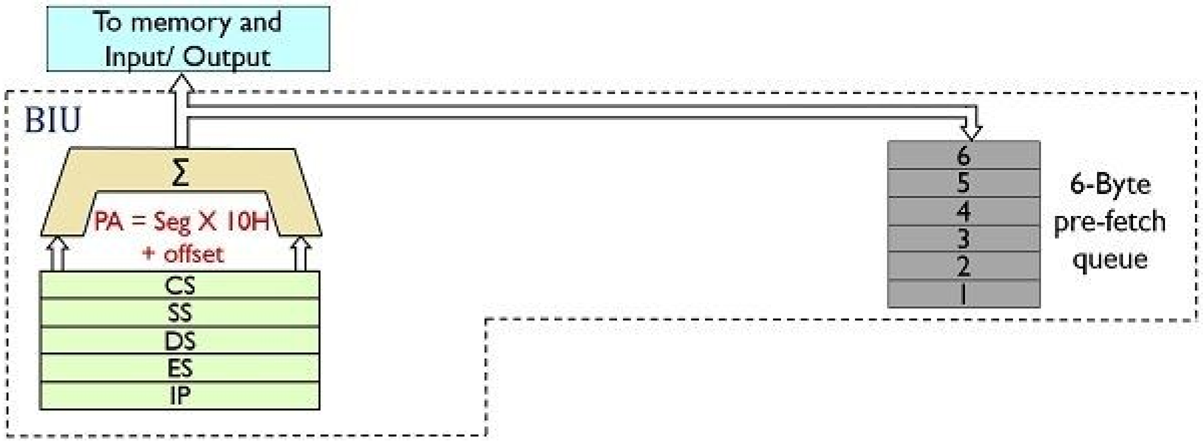
The reason behind two separate sections for BIU and EU in the architecture of 8086 is to perform fetching and decoding-executing simultaneously.

Working of 8086 Microprocessor

In the previous section, we have discussed the operation of various sections of the BIU and EU. Now in this section, we will have a look at the overall processing cycle of the 8086 microprocessors. So, basically, when an instruction is to be fetched from the memory, then firstly its physical address must be calculated and this is done at the BIU. The physical address of an instruction is given as:

PA = Segment address X 10 + Offset

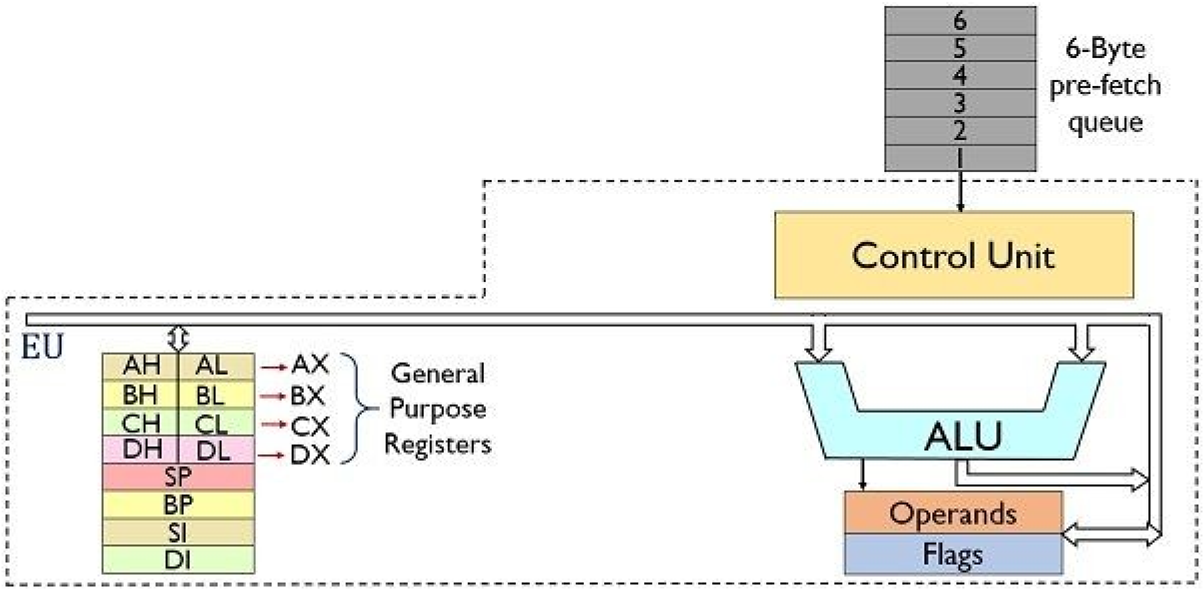
For example: Suppose the segment address is 2000 H and the offset address is 4356 H. So, the generated physical address is **24356 H**. Here, the code segment register provides the base address of the code segment which is combined with the offset address.



The code segment contains the instructions. Each time an instruction is fetched the offset address inside the code segment gets incremented. So, once the physical address of an instruction is calculated by the BIU of the processor, it sends the memory location by the address bus to the memory. Further, the desired instruction at that memory location which is present in the form of the opcode is fetched by the microprocessor through the data bus.

Suppose the instruction is **ADD BL, CL**. But, inside the memory, it will be in the form of an opcode. So, this opcode is sent to the control unit.

The control unit decodes the opcode and generates control signals that inform the BL and CL register to release the value stored in it. Also, it signals the ALU to perform the ADD operation on that particular data.



It is noteworthy that in any instruction, like ADD BL, CL. BL denotes the destination of the result of the add operation. This clearly shows that whatever, the operation is performed its result must be stored in the first register i.e., BL for this particular example.

Let us take another example: Consider an instruction, **ADD CL, 05H**.

This means that the operand which is 05H is to be added with the data present in the CL register and is stored in that particular register i.e., CL. In such conditions, the operand is not provided to the control unit as only the opcode is required to be decoded by the CU. Hence the operand is directly provided to the ALU. Also, the status of this result is stored in the flag register. So, whenever, ALU carries out an operation, it simultaneously generates the result as well as its status.

It is to be noteworthy that in BIU, pipelining fails whenever there is branching in the instruction. This is because generally instructions are present in a sequential manner. But, sometimes the instructions are required to be executed unsequentially. However, in the queue, the instructions are stored sequentially. So, in case there exist a need for any random instruction to be decoded. The opcode stored in the queue will become invalid and must be cleared at that particular time.