Block Diagram of Intel 8086

The 8086 CPU is divided into two independent functional units:

1. Bus Interface Unit (BIU)
2. Execution Unit (EU)

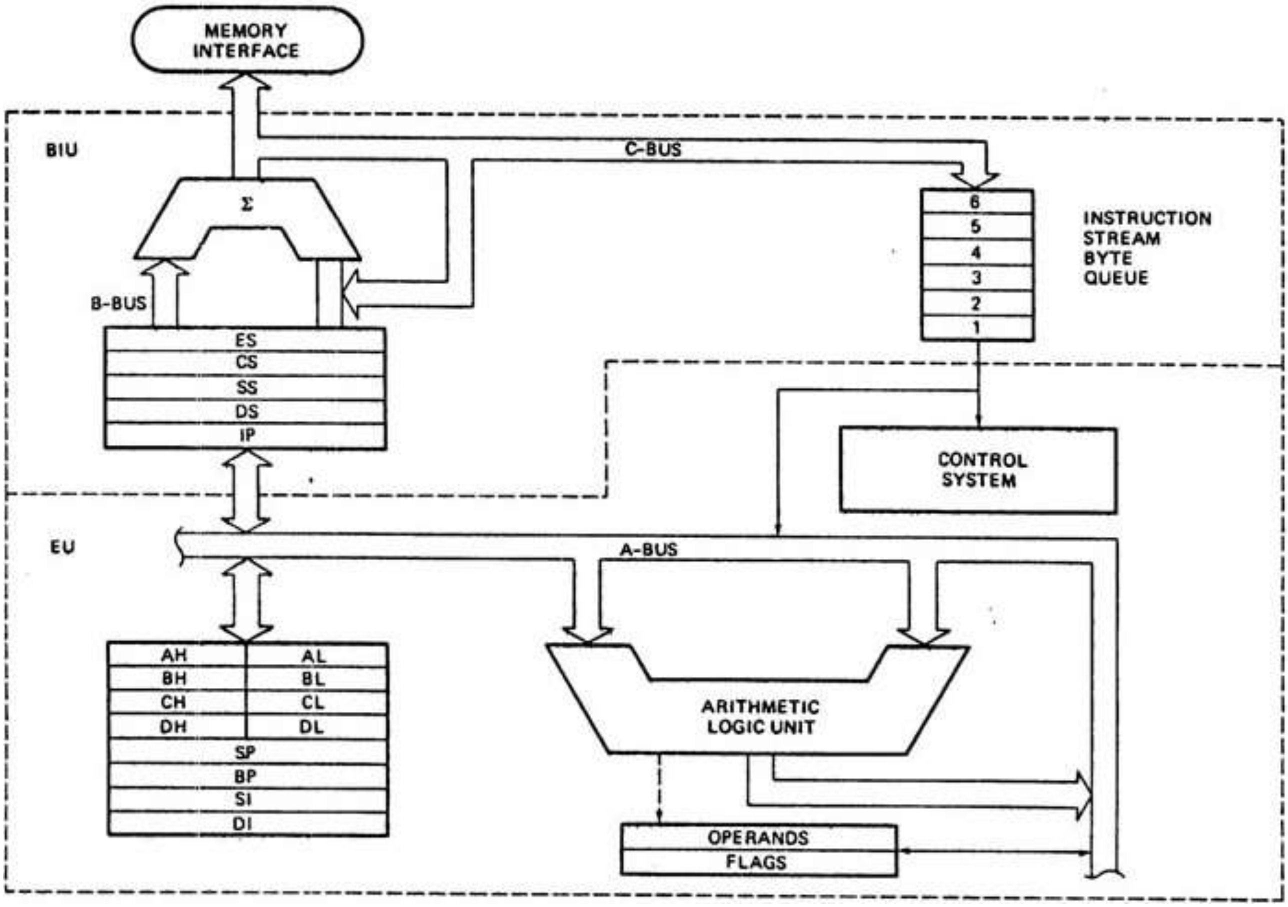


Fig. 1: Block Diagram of Intel 8086

Features of 8086 Microprocessor:

1. Intel 8086 was launched in 1978.
2. It was the first 16-bit microprocessor.
3. This microprocessor had major improvement over the execution speed of 8085.
4. It is available as 40-pin Dual-Inline-Package (DIP).
5. It is available in three versions:
   - **a.** 8086 (5 MHz)
   - **b.** 8086-2 (8 MHz)

   c. 8086-1 (10 MHz)

6. It consists of 29,000 transistors.

Bus Interface Unit (BIU)

The function of BIU is to:

- Fetch the instruction or data from memory.
- Write the data to memory.
- Write the data to the port.
- Read data from the port.

Instruction Queue

1. To increase the execution speed, BIU fetches as many as six instruction bytes ahead to time from memory.
2. All six bytes are then held in first in first out 6 byte register called instruction queue.
3. Then all bytes have to be given to EU one by one.
4. This pre fetching operation of BIU may be in parallel with execution operation of EU, which improves the speed execution of the instruction.

Execution Unit (EU)

The functions of execution unit are:

- To tell BIU where to fetch the instructions or data from.
- To decode the instructions.
- To execute the instructions.

The EU contains the control circuitry to perform various internal operations. A decoder in EU decodes the instruction fetched memory to generate different internal or external control signals required to perform the operation. EU has 16-bit ALU, which can perform arithmetic and logical operations on 8-bit as well as 16-bit.

General Purpose Registers of 8086

These registers can be used as 8-bit registers individually or can be used as 16-bit in pair to have AX, BX, CX, and DX.

1. **AX Register:** AX register is also known as accumulator register that stores operands for arithmetic operation like divided, rotate.
2. **BX Register:** This register is mainly used as a base register. It holds the starting base location of a memory region within a data segment.
3. **CX Register:** It is defined as a counter. It is primarily used in loop instruction to store loop counter.
4. **DX Register:** DX register is used to contain I/O port address for I/O instruction.

Segment Registers

Additional registers called segment registers generate memory address when combined with other in the microprocessor. In 8086 microprocessor, memory is divided into 4 segments as follow:

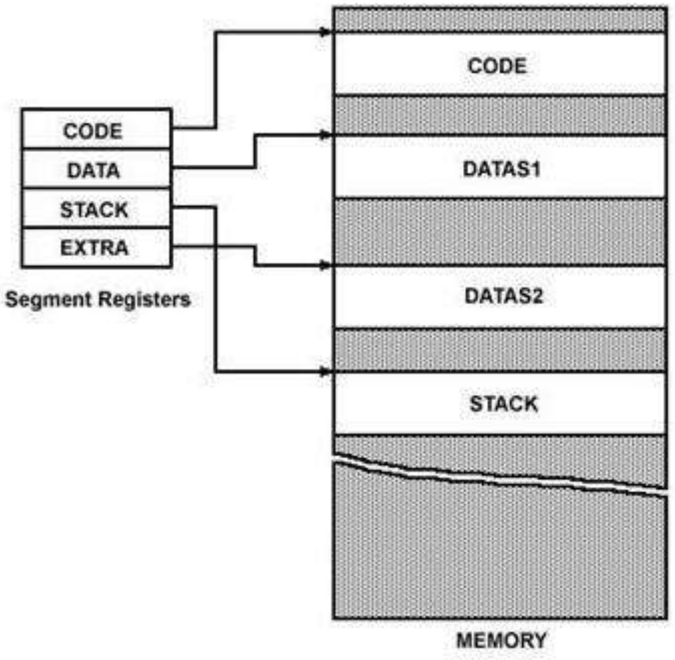


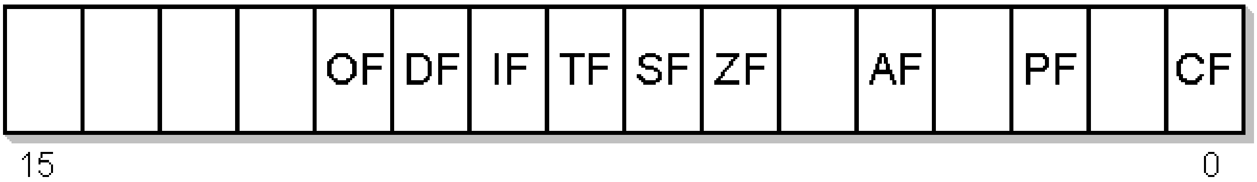
Fig. 2: Memory Segments of 8086

1. **Code Segment (CS):** The CS register is used for addressing a memory location in the Code Segment of the memory, where the executable program is stored.
2. **Data Segment (DS):** The DS contains most data used by program. Data are accessed in the Data Segment by an offset address or the content of other register that holds the offset address.
3. **Stack Segment (SS):** SS defined the area of memory used for the stack.

4. **Extra Segment (ES):** ES is additional data segment that is used by some of the string to hold the destination data.

Flag Registers of 8086

Flag register in EU is of 16-bit and is shown in fig. 3:



				OF	DF	IF	TF	SF	ZF		AF		PF		CF
15															0

Fig. 3: Flag Register of 8086

Flags Register determines the current state of the processor. They are modified automatically by CPU after mathematical operations, this allows to determine the type of the result, and to determine conditions to transfer control to other parts of the program. 8086 has 9 flags and they are divided into two categories:

1. Conditional Flags
2. Control Flags

Conditional Flags

Conditional flags represent result of last arithmetic or logical instruction executed. Conditional flags are as follows:

- **Carry Flag (CF):** This flag indicates an overflow condition for unsigned integer arithmetic. It is also used in multiple-precision arithmetic.
- **Auxiliary Flag (AF):** If an operation performed in ALU generates a carry/barrow from lower nibble (i.e. $D_0 – D_3$) to upper nibble (i.e. $D_4 – D_7$), the AF flag is set i.e. carry given by D_3 bit to D_4 is AF flag. This is not a general-purpose flag, it is used internally by the processor to perform Binary to BCD conversion.
- **Parity Flag (PF):** This flag is used to indicate the parity of result. If lower order 8-bits of the result contains even number of 1’s, the Parity Flag is set and for odd number of 1’s, the Parity Flag is reset.
- **Zero Flag (ZF):** It is set; if the result of arithmetic or logical operation is zero else it is reset.
- **Sign Flag (SF):** In sign magnitude format the sign of number is indicated by MSB bit. If the result of operation is negative, sign flag is set.

- **Overflow Flag (OF):** It occurs when signed numbers are added or subtracted. An OF indicates that the result has exceeded the capacity of machine.

Control Flags

Control flags are set or reset deliberately to control the operations of the execution unit. Control flags are as follows:

1. **Trap Flag (TP):**
   - **a.** It is used for single step control.
   - **b.** It allows user to execute one instruction of a program at a time for debugging.
   - **c.** When trap flag is set, program can be run in single step mode.
2. **Interrupt Flag (IF):**
   - **a.** It is an interrupt enable/disable flag.
   - **b.** If it is set, the maskable interrupt of 8086 is enabled and if it is reset, the interrupt is disabled.
   - **c.** It can be set by executing instruction sit and can be cleared by executing CLI instruction.
3. **Direction Flag (DF):**
   - **a.** It is used in string operation.
   - **b.** If it is set, string bytes are accessed from higher memory address to lower memory address.
   - **c.** When it is reset, the string bytes are accessed from lower memory address to higher memory address.