

Block Diagram of Intel 8086

The 8086 CPU is divided into two independent functional units:

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

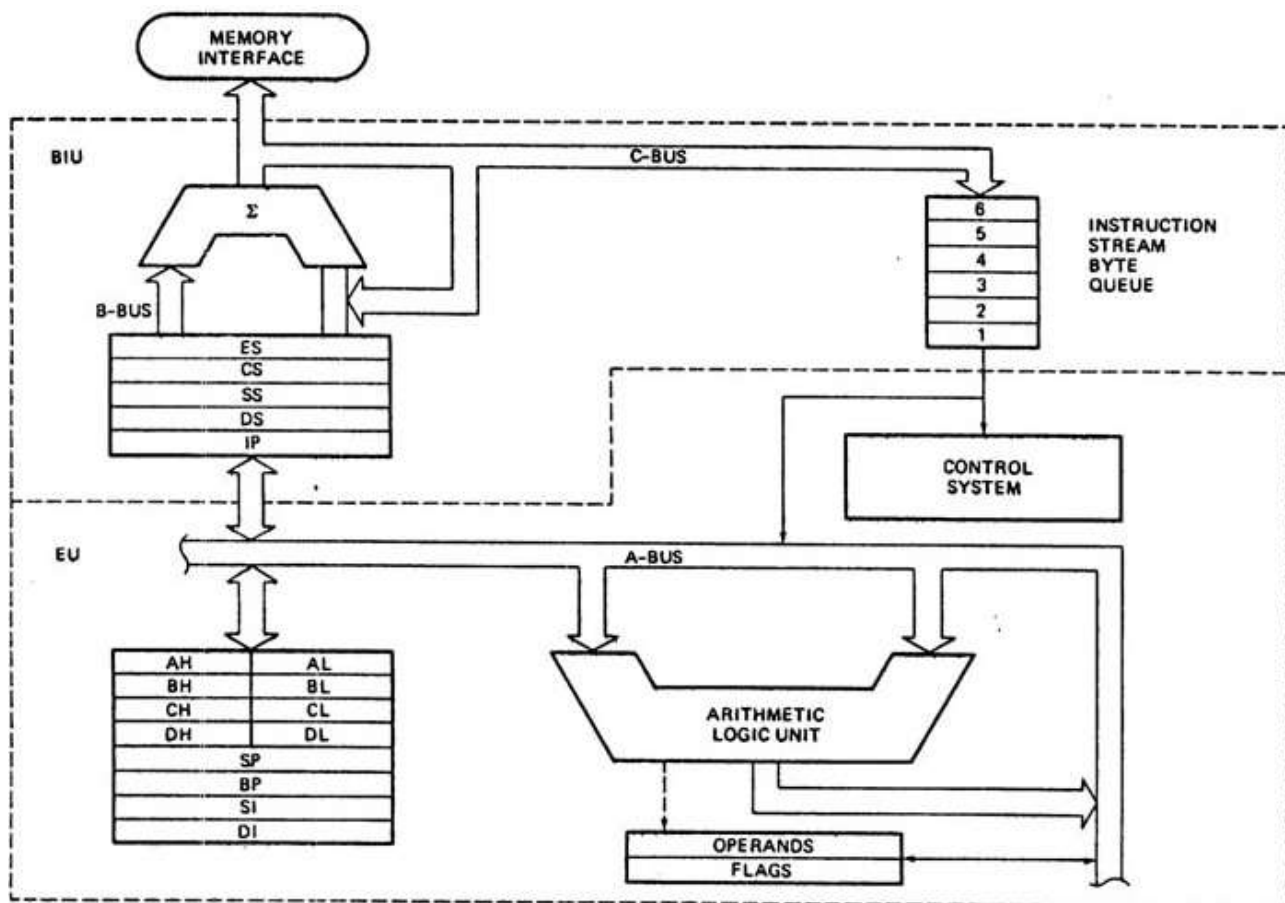


Fig. 1: Block Diagram of Intel 8086

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:

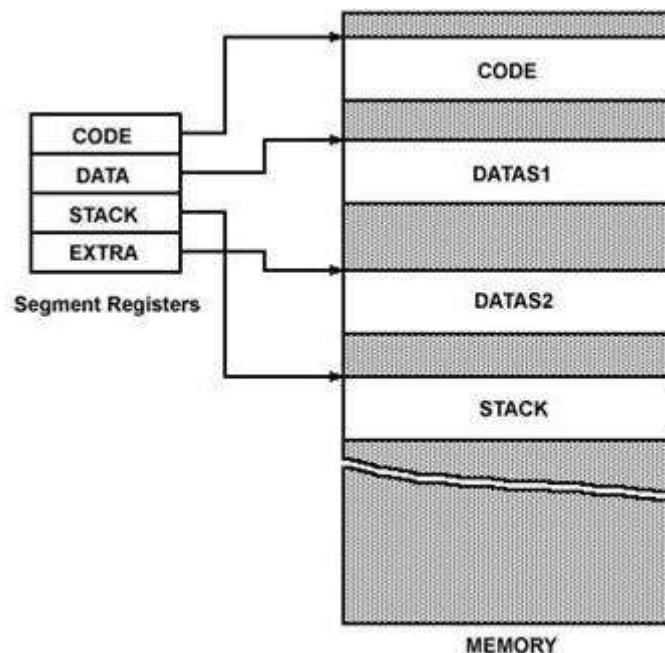


Fig. 2: Memory Segments of 8086

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:

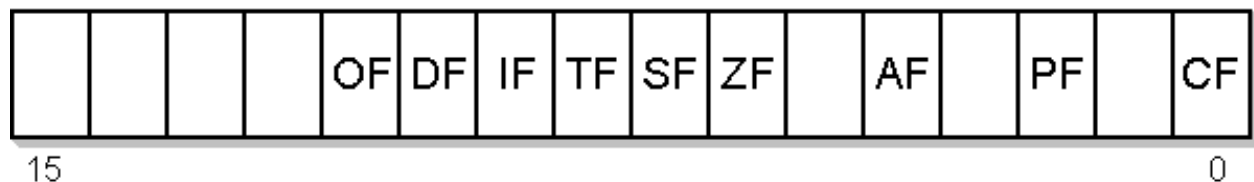


Fig. 3: Flag Register of 8086

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/borrow 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 (TF):

- 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 `sti` 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.

16Bit Microprocessor : 8086

Features of 8086

- 8086 is a 16bit processor. It's ALU, internal registers works with 16bit binary word
- 8086 has a 16bit data bus. It can read or write data to a memory/port either 16bits or 8 bit at a time
- 8086 has a 20bit address bus which means, it can address upto $2^{20} = 1\text{MB}$ memory location
- Frequency range of 8086 is 6-10 MHz

Data Read/Write process from /To Memory

Word Read

- Each of 1 MB memory address of 8086 represents a byte wide location
- 16bit words will be stored in two consecutive Memory location
- If first byte of the data is stored at an **even address** , 8086 can read the entire word in one operation.
 - o For example if the 16 bit data is stored at even address 00520_H is 2607

MOV BX, [00520]
 8086 reads the first byte and stores the data in BL and reads the 2nd byte and stores the data in BH

BL ← (00520)
 BH ← (00521)
- If the first byte of the data is stored at an ODD address, 8086 needs two operation to read the 16 bit data
 - o For example if the 16 bit data is stored at even address 00521_H is F520

MOV BX, [00521]
 In first operation , 8086 reads the 16 bit data from the 00520 location and stores the data of 00521 location in register BL and discards the data of 00520 location

In 2nd operation, 8086 reads the 16 bit data from the 00522 location and stores the data of 00522 location in register BH and discards the data of 00523 location

BL ← (00521)

BH ← (00522)

Byte Read:

MOV BH, [Addr]

For Even Address:

Ex: MOV BH, [00520]

8086 reads the first byte from 00520 location and stores the data in BH and reads the 2nd byte from the 00521 location and ignores it

BH ← [00520]

For Odd Address

MOV BH, [Addr]

Ex: MOV BH, [00521]

8086 reads the first byte from 00520 location and ignores it and reads the 2nd byte from the 00521 location and stores the data in BH

BH ← [00521]

Registers of 8086

Category	Bits	Register Names
General	16	AX,BX,CX,DX
	8	AH,AL,BH,BL,CH,CL,DH,DL
Pointer	16	SP (Stack Pointer), Base Pointer (BP)
Index	16	SI (Source Index), DI (Destination Index)
Segment	16	CS(Code Segment) DS (Data Segment) SS (Stack Segment) ES (Extra Segment)
Instruction	16	IP (Instruction Pointer)
Flag	16	FR (Flag Register)

General Purpose Registers

15	H	8	7	L	0
AX (Accumulator)					
AH			AL		
BX (Base Register)					
BH			BL		
CX (Used as a counter)					
CH			CL		
DX (Used to point to data in I/O operations)					
DH			DL		

- **Data Registers** are normally used for storing temporary results that will be acted upon by subsequent instructions
- Each of the registers is 16 bits wide (AX, BX, CX, DX)
- General purpose registers can be accessed as either 16 or 8 bits
e.g., AH: upper half of AX, AL: lower half of AX

Pointer and Index Registers

SP	Stack Pointer
BP	Base Pointer
SI	Source Index
DI	Destination Index
IP	Instruction Pointer

The registers in this group are all 16 bits wide
Low and high bytes are not accessible

These registers are used as memory pointers

- Example: MOV AH, [SI]

*Move the byte stored in memory location
whose address is contained in register SI to register
AH*

IP is not under direct control of the programmer

Important 8086 Pin Diagram/Description

AD15±AD0

ADDRESS DATA BUS: These lines constitute the time multiplexed memory/I/O address and data bus.

ALE

Address Latch Enable. A HIGH on this line causes the lower order 16bit address bus to be latched that stores the addresses and then, the lower order 16bit of the address bus can be used as data bus.

READY

READY is the acknowledgement from the addressed memory or I/O device that it will complete the data transfer.

INTR

INTERRUPT REQUEST: is a level triggered input which is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt acknowledge operation. A subroutine is vectored to via an interrupt vector lookup table located in system memory. It can be internally masked by software resetting the interrupt enable bit. INTR is internally synchronized. This signal is active HIGH.

$\overline{\text{INTA}}$

Interrupt Acknowledge from the MP

NMI

NON-MASKABLE INTERRUPT: an edge triggered input which causes an interrupt request to the MP. A subroutine is vectored to via an interrupt vector lookup table located in system memory. NMI is not maskable internally by software.

RESET: causes the processor to immediately terminate its present activity. The signal must be active HIGH for at least four clock cycles. It restarts execution

$\overline{\text{MN/MX}}$

MINIMUM/MAXIMUM: indicates what mode the processor is to operate in. The two modes are discussed in the following sections.

$\overline{\text{M/IO}}$: Differentiate between the Memory and I/O operation. A LOW on this pin indicated I/O operation and a HIGH indicated a Memory Operation

HOLD : The 8086 has a pin called HOLD. This pin is used by external devices to gain control of the busses.

HLDA :

When the HOLD signal is activated by an external device, the 8086 stops executing instructions and stops using the busses. This would allow external devices to control the information on the

8086 MINIMUM AND MAXIMUM MODES of operation

$\overline{\text{MN/MX}}$

- **Minimum mode** *The 8086 processor works in a single processor environment. All control signals for memory and I/O are generated by the microprocessor.*
- **Maximum mode** *is designed to be used when a coprocessor exists in the system.*
- *8086 works in a multiprocessor environment. Control signals for memory and I/O are generated by an external BUS Controller.*

Data Transfer Between CPU and the Memory

Memory Write:

Byte Transfer: `move BYTEPTR    ds : [SI], 37H`

Word Transfer: `move WORDPTR   ds : [SI], 1237H`

Memory Read:

Byte Transfer: `move al, BYTEPTR    ds : [SI]`

Transfers data from the physical memory address calculated using ds and [SI] to register AL (Lower byte of AX Register)

Word Transfer: `move ax, WORDPTR   ds : [SI]`

Transfers data from the physical memory address calculated using ds and [SI] to register AL (Lower byte of AX Register) and the next byte from the next memory location calculated as ds:[SI+1] is transferred to AH (Higher byte of AX Register)

Memory operation through ax Register

Write:

`MOV AX , 1234H`

`MOV WORDPTR ds: [SI], ax`

Ds: 0000H

SI: 0500H

Physical Address: 00000+0500= 00500 H

The instruction transfers

34 → 00500H

12 → 00501H

Read:

`MOV ax, WORDPTR ds: [SI]`

Ds: 0000H

SI: 0500H

Physical Address: 00000+0500= 00500 H

The instruction transfers

AL $\leftarrow$ (00500)

AH $\leftarrow$ (00501)

Data Transfer Between CPU and the Port

Port addresses in 8086 are assigned either 8bit port address or 16 bit address

For a Port with 8bit port address:

Read Operation:

IN Pdr where Pdr is the 8bit Port address

Ex: IN 20H

The instruction transfers data byte from the 8bit port address 20H to register AL

Write Operation:

OUT Pdr where Pdr is the 8bit Port address

Ex: OUT 20H

The instruction transfers data byte from AL to the 8bit port address 20H .

For a Port with 16bit port address:

DX register is used to hold the Port address

Read Operation:

Example:

Mov DX, 4000H

IN al, DX

The instruction transfers data byte from 16bit port address 4000H contained in DX register to AL.

Write Operation:

Example:

```
MOV AL, 10H  
MOV DX, 4000H  
OUT DX, al
```

The instruction transfers data byte 10H from register AL to 16bit port address 4000H contained in DX