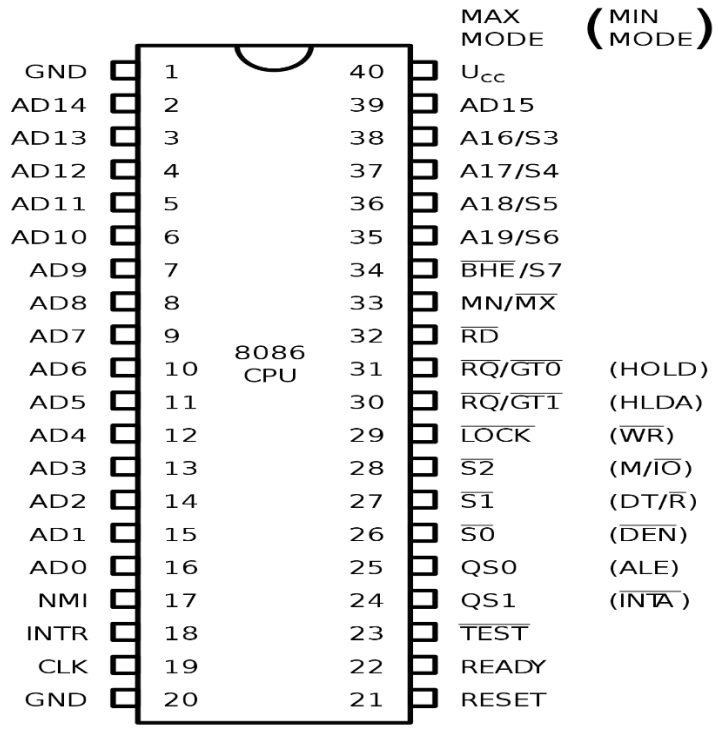
UNIT II 8086 Microprocessor (Notes)

Features of 8086 Microprocessor

- It has an instruction queue, which is capable of storing six instruction bytes from the memory resulting in faster processing.
- It was the first 16-bit processor having 16-bit ALU, 16-bit registers, internal data bus, and 16-bit external data bus resulting in faster processing.
- It is available in 3 versions based on the frequency of operation – 5MHz,8MHz and 10 MHz
- It uses two stages of pipelining, i.e. Fetch Stage and Execute Stage, which improves performance.
- Fetch stage can prefetch up to 6 bytes of instructions and stores them in the queue.
- Execute stage executes these instructions.
- It has 256 vectored interrupts.
- It consists of 29,000 transistors.

Pin Diagram & Description



1) Power supply and frequency signals

It uses 5V DC supply at V_{CC} **pin 40**, and uses ground at V_{SS} **pin 1 and 20** for its operation.

2) Clock signal

Clock signal is provided through **Pin-19**. It provides timing to the processor for operations. Its frequency is different for different versions, i.e. 5MHz, 8MHz and 10MHz.

3) Address/data bus

AD0-AD15. These are 16 address/data bus. AD0-AD7 carries low order byte data and AD8AD15 carries higher order byte data. During the first clock cycle, it carries 16-bit address and after that it carries 16-bit data.

4) Address/status bus

A16-A19/S3-S6. These are the 4 address/status buses. During the first clock cycle, it carries 4-bit address and later it carries status signals.

5) S7/BHE

BHE stands for Bus High Enable. It is available at **pin 34** and used to indicate the transfer of data using data bus D8-D15. This signal is low during the first clock cycle, thereafter it is active.

6) Read (RD')

It is available at **pin 32** and is used to read signal for Read operation.

7) Ready

It is available at **pin 22**. It is an acknowledgement signal from I/O devices that data is transferred. It is an active high signal. When it is high, it indicates that the device is ready to transfer data. When it is low, it indicates wait state.

8) RESET

It is available at **pin 21** and is used to restart the execution. It causes the processor to immediately terminate its present activity. This signal is active high for the first 4 clock cycles to RESET the microprocessor.

9) INTR

It is available at **pin 18**. It is an interrupt request signal, which is sampled during the last clock cycle of each instruction to determine if the processor considered this as an interrupt or not.

10) NMI

It stands for non-maskable interrupt and is available at **pin 17**. It is an edge triggered input, which causes an interrupt request to the microprocessor.

11) (TEST')

This signal is like wait state and is available at **pin 23**. When this signal is high, then the processor has to wait for IDLE state, else the execution continues.

12) MN/*MX'*

It stands for Minimum/Maximum and is available at **pin 33**. It indicates what mode the processor is to operate in; when it is high, it works in the minimum mode and vice-versa.

- **Minimum Mode Signal Pins:**

13) INTA

It is an interrupt acknowledgement signal and id available at **pin 24**. When the microprocessor receives this signal, it acknowledges the interrupt.

14) ALE

It stands for address enable latch and is available at **pin 25**. A positive pulse is generated each time the processor begins any operation. This signal indicates the availability of a valid address on the address/data lines.

15) DEN

It stands for Data Enable and is available at **pin 26**. It is used to enable Transreceiver 8286. The transreceiver is a device used to separate data from the address/data bus.

16) DT/R

It stands for Data Transmit/Receive signal and is available at **pin 27**. It decides the direction of data flow through the transreceiver. When it is high, data is transmitted out and vice-a-versa.

17) M/IO

This signal is used to distinguish between memory and I/O operations. When it is high, it indicates I/O operation and when it is low indicates the memory operation. It is available at **pin 28.**

18) WR

It stands for write signal and is available at **pin 29**. It is used to write the data into the memory or the output device depending on the status of M/IO signal.

19) HLDA

It stands for Hold Acknowledgement signal and is available at **pin 30**. This signal acknowledges the HOLD signal.

20) HOLD

This signal indicates to the processor that external devices are requesting to access the address/data buses. It is available at **pin 31**.

- **Maximum Mode Signal Pins:**

21) QS_1 and QS_0

These are queue status signals and are available at **pin 24 and 25**. These signals provide the status of instruction queue. Their conditions are shown in the following table –

QS_0	QS_1	Status
0	0	No operation
0	1	First byte of opcode from the queue
1	0	Empty the queue
1	1	Subsequent byte from the queue

22) S_0, S_1, S_2

These are the status signals that provide the status of operation, which is used by the Bus Controller 8288 to generate memory & I/O control signals. These are available at **pin 26, 27, and 28.** Following is the table showing their status −

S_2	S_1	S_0	Status
0	0	0	Interrupt acknowledgement
0	0	1	I/O Read
0	1	0	I/O Write
0	1	1	Halt
1	0	0	Opcode fetch
1	0	1	Memory read
1	1	0	Memory write
1	1	1	Passive

23) LOCK’

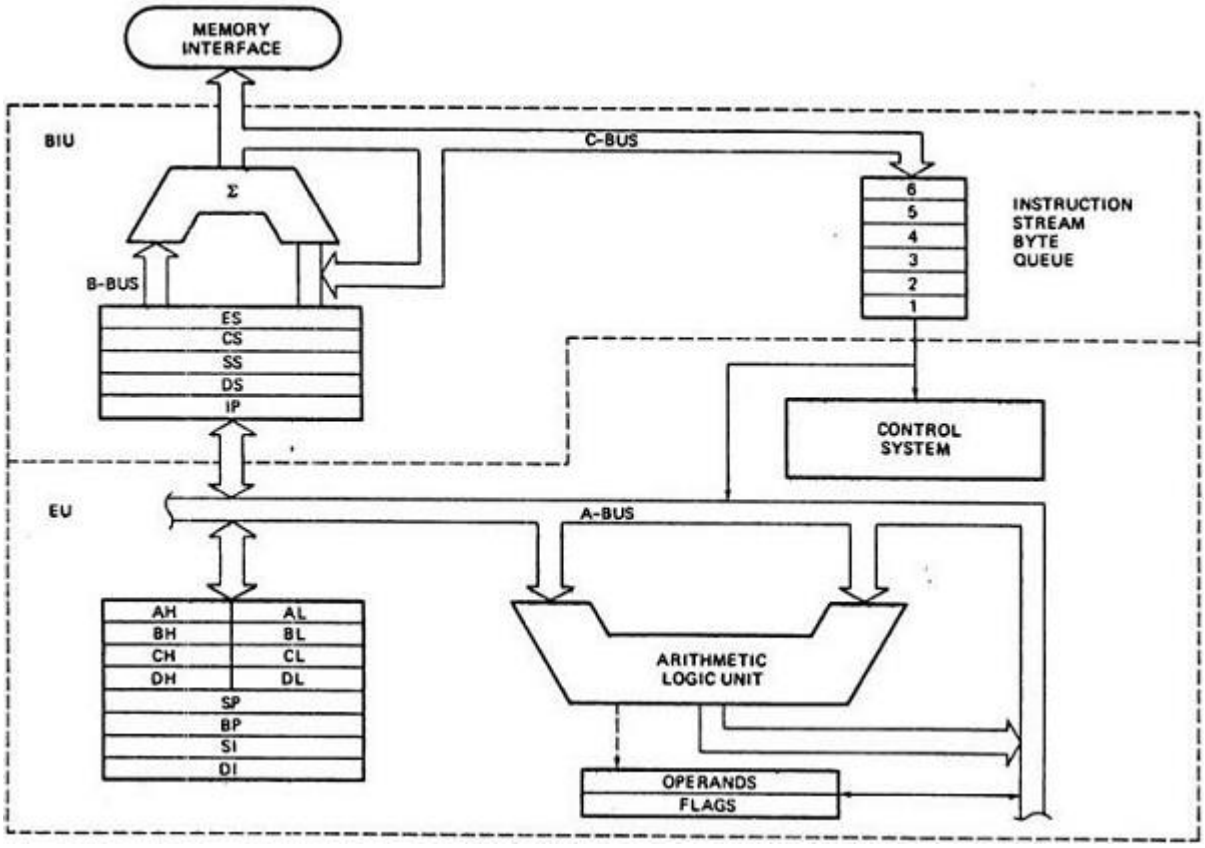
When this signal is active, it indicates to the other processors not to ask the CPU to leave the system bus. It is activated using the LOCK prefix on any instruction and is available at **pin 29.**

24) RQ/GT_1 and RQ/GT_0

These are the Request/Grant signals used by the other processors requesting the CPU to release the system bus. When the signal is received by CPU, then it sends

acknowledgment. RQ/GT_0 has a higher priority than RQ/GT_1 and it is available at **pin 30 and 31.**

Architecture of 8086-Block diagram and description



8086 Microprocessor is divided into two functional units, i.e., **EU** (Execution Unit) and **BIU** (Bus Interface Unit).

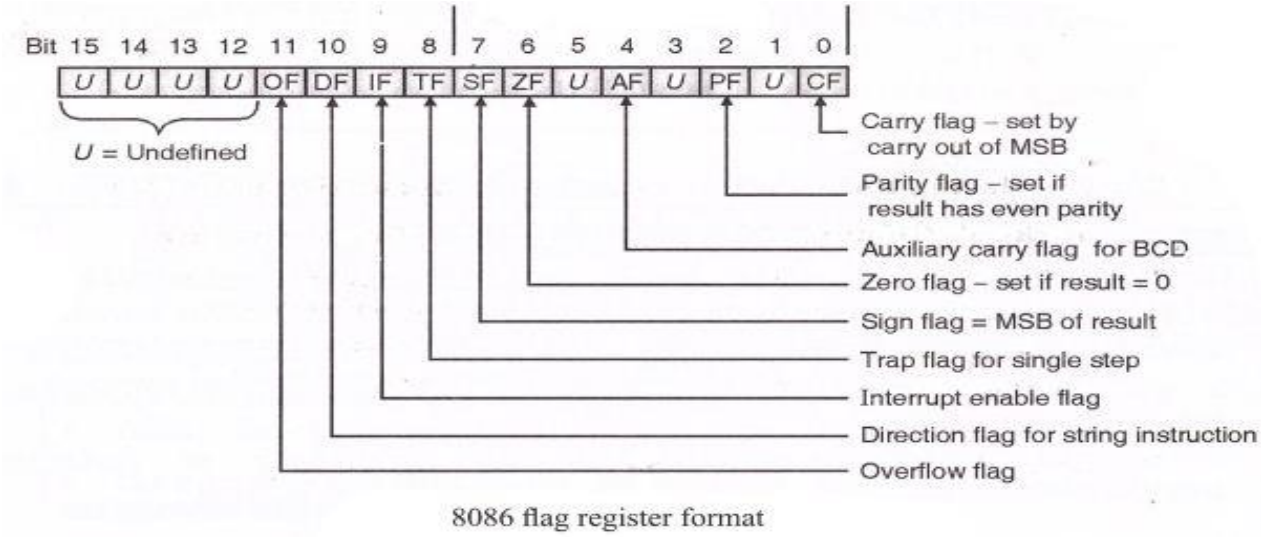
EU (Execution Unit)

Execution unit gives instructions to BIU stating from where to fetch the data and then decode and execute those instructions. Its function is to control operations on data using the instruction decoder & ALU. EU has no direct connection with system buses as shown in the above figure, it performs operations over data through BIU.

ALU

It handles all arithmetic and logical operations, like +, −, ×, /, OR, AND, NOT operations.

Flag Register



8086 flag register format

It is a 16-bit register that behaves like a flip-flop, i.e. it changes its status according to the result stored in the accumulator. It has 9 flags and they are divided into 2 groups − Conditional Flags and Control Flags.

Conditional Flags

It represents the result of the last arithmetic or logical instruction executed. Following is the list of conditional flags −

- **Carry flag** − This flag indicates an overflow condition for arithmetic operations.
- **Auxiliary flag** − When an operation is performed at ALU, it results in a carry/barrow from lower nibble (i.e. D0 – D3) to upper nibble (i.e. D4 – D7), then this flag is set, i.e. carry given by D3 bit to D4 is AF flag. The processor uses this flag to perform binary to BCD conversion.
- **Parity flag** − This flag is used to indicate the parity of the result, i.e. when the lower order 8-bits of the result contains even number of 1’s, then the Parity Flag is set. For odd number of 1’s, the Parity Flag is reset.
- **Zero flag** − This flag is set to 1 when the result of arithmetic or logical operation is zero else it is set to 0.

- **Sign flag** − This flag holds the sign of the result, i.e. when the result of the operation is negative, then the sign flag is set to 1 else set to 0.
- **Overflow flag** − This flag represents the result when the system capacity is exceeded.

Control Flags

Control flags controls the operations of the execution unit. Following is the list of control flags −

- **Trap flag** − It is used for single step control and allows the user to execute one instruction at a time for debugging. If it is set, then the program can be run in a single step mode.
- **Interrupt flag** − It is an interrupt enable/disable flag, i.e. used to allow/prohibit the interruption of a program. It is set to 1 for interrupt enabled condition and set to 0 for interrupt disabled condition.
- **Direction flag** − It is used in string operation. As the name suggests when it is set then string bytes are accessed from the higher memory address to the lower memory address and vice-a-versa.

General purpose register

There are 8 general purpose registers, i.e., AH, AL, BH, BL, CH, CL, DH, and DL. These registers can be used individually to store 8-bit data and can be used in pairs to store 16bit data. The valid register pairs are AH and AL, BH and BL, CH and CL, and DH and DL. It is referred to the AX, BX, CX, and DX respectively.

- **AX register** − It is also known as accumulator register. It is used to store operands for arithmetic operations.
- **BX register** − It is used as a base register. It is used to store the starting base address of the memory area within the data segment.
- **CX register** − It is referred to as counter. It is used in loop instruction to store the loop counter.
- **DX register** − This register is used to hold I/O port address for I/O instruction.

Stack pointer register

It is a 16-bit register, which holds the address from the start of the segment to the memory location, where a word was most recently stored on the stack.

BIU (Bus Interface Unit)

BIU takes care of all data and addresses transfers on the buses for the EU like sending addresses, fetching instructions from the memory, reading data from the ports and the memory as well as writing data to the ports and the memory. EU has no direction connection with System Buses so this is possible with the BIU. EU and BIU are connected with the Internal Bus.

It has the following functional parts −

- **Instruction queue** − BIU contains the instruction queue. BIU gets upto 6 bytes of next instructions and stores them in the instruction queue. When EU executes instructions and is ready for its next instruction, then it simply reads the instruction from this instruction queue resulting in increased execution speed.
- Fetching the next instruction while the current instruction executes is called **pipelining**.
- **Segment register** − BIU has 4 segment buses, i.e. CS, DS, SS& ES. It holds the addresses of instructions and data in memory, which are used by the processor to access memory locations. It also contains 1 pointer register IP, which holds the address of the next instruction to executed by the EU.
    - **CS** − It stands for Code Segment. It is used for addressing a memory location in the code segment of the memory, where the executable program is stored.
    - **DS** − It stands for Data Segment. It consists of data used by the program and is accessed in the data segment by an offset address or the content of other register that holds the offset address.
    - **SS** − It stands for Stack Segment. It handles memory to store data and addresses during execution.
    - **ES** − It stands for Extra Segment. ES is additional data segment, which is used by the string to hold the extra destination data.

- **Instruction pointer** − It is a 16-bit register used to hold the address of the next instruction to be executed.

Concept of Pipelining:

The process of fetching the next instruction when the present instruction is being executed is called pipelining.
Pipeline system is like modern day assembly line setup in factories.
Example:
In a car manufacturing industry, huge assembly lines are setup & at each point, there are robotic arms to perform a certain task & car moves on ahead next arm.

- **Process of Pipelining**:

Pre-fetched 6 bytes which are stored in first in first out (FIFO) register set is called queue.
EU gets opcode of an instruction from instruction queue. Then the EU decodes It & executes it.
At this time BIU fetches instruction codes from the memory &store them in the queue.

Example:

➢ While EU is busy in decoding instruction to memory location 100F0H, BIU fetches next 6 instruction bytes from 100F1 H to 100F6 H as 1 to 6.
➢ While EU completes the execution of existing instruction &become ready for next instruction, it simply reads instruction bytes in sequence 1, 2, 3,….
from queue.

- **Advantage of pipelining:**

The cycle time of the processor is reduced.
It increases the throughput of the system.
It makes the system reliable.

- **Disadvantages of pipelining:**

The design of pipelined processor is complex & costly to manufacture.
The instruction latency is more.

Concept of Memory Segmentation:

Segmentation is the process in which the main memory of the computer is divided into different segments and each segment has its own base address. It is basically used to enhance the speed of execution of the computer system, so that processor is able to fetch and execute the data from the memory easily and fast.

Need for Segmentation –

The Bus Interface Unit (BIU) contains four 16 bit special purpose registers (mentioned below) called as Segment Registers.

- **Code segment register (CS):** is used for addressing memory location in the code segment of the memory, where the executable program is stored.
- **Data segment register (DS):** points to the data segment of the memory where the data is stored.
- **Extra Segment Register (ES):** also refers to a segment in the memory which is another data segment in the memory.
- **Stack Segment Register (SS):** is used for addressing stack segment of the memory. The stack segment is that segment of memory which is used to store stack data.

The number of address lines in 8086 is 20, 8086 BIU will send 20bit address, so as to access one of the 1MB memory locations. The four segment registers actually contain the upper 16 bits of the starting addresses of the four memory segments of 64 KB each with which the 8086 is working at that instant of time. A segment is a logical unit of memory that may be up to 64 kilobytes long. Each segment is made up of contiguous memory locations. It is independent, separately addressable unit. Starting address will always be changing. It will not be fixed.

Note that the 8086 does not work the whole 1MB memory at any given time. However it works only with four 64KB segments within the whole 1MB memory.

Advantages of the Segmentation

- It provides a powerful memory management mechanism.
- Data related or stack related operations can be performed in different segments.
- Code related operation can be done in separate code segments.

- It allows to processes to easily share data.
- It allows to extend the address ability of the processor, i.e. segmentation allows the use of 16 bit registers to give an addressing capability of 1 Megabytes. Without segmentation, it would require 20 bit registers.
- It is possible to enhance the memory size of code data or stack segments beyond 64 KB by allotting more than one segment for each area.

Minimum and Maximum Mode Signals:

8086 has two operating Modes:

1. Minimum mode
2. Maximum mode

Minimum mode:

- In this 8086 is the only processor in the system. In a minimum mode 8086 system.
- 8086 is operated in minimum mode when MN/MX' pin to logic 1.
- In this mode, all the control signals are given out by the 8086 itself.

Maximum mode:

- In this we can connect more processors to 8086 (8087/8089).
- 8086 max mode is basically for implementation of allocation of global resources and passing bus control to other coprocessor (i.e. second processor in the system), because two processors cannot access system bus at same instant.
- All processors execute their own program.
- The resources which are common to all processors are known as global resources.
- The resources which are allocated to a particular processor are known as local or private resources.