विशाल जाधव सरांचे

VJTech Academy

Inspiring Your Success...

UNIT-I 8086 16-bit Microprocessor

Syllabus:

Total Marks:14

1.1 8086 Microprocessor: salient features, pin description

1.2 Architecture of 8086: functional block diagram, register organization.

1.3 Concept of pipelining 1.4 Memory segmentation, physical memory address generation

Introduction to Microprocessor:

● A processor is the logic circuitry that responds to and processes the basic instructions that drives a computer.

● The term processor has generally replaced the term central processing unit (CPU).

● The processor in a personal computer or embedded in small devices is often called a microprocessor.

● The processor (CPU, for Central Processing Unit) is the computer's brain. It allows the processing of numeric data, meaning information entered in binary form, and the execution of instructions stored in memory.

Evolution of Microprocessor:

- A microprocessor is used as the CPU in a microcomputer. There are now many different microprocessors available.
- Microprocessor is a program-controlled device, which fetches the instructions from memory, decodes and executes the instructions.
- Most Microprocessor are single- chip devices. Microprocessor is a backbone of computer system. which is called CPU Microprocessor speed depends on the processing speed depends on DATA BUS WIDTH.
- A common way of categorizing microprocessors is by the no. of bits that their ALU can Work with at a time.
- The address bus is unidirectional because the address information is always given by the Microprocessor to address a memory location of an input / output devices.
- The data bus is Bi-directional because the same bus is used for transfer of data between Micro Processor and memory or input / output devices in both the direction. It has limitations on the size of data.
- Microprocessor contain ROM chip because it contains instructions to execute data.

1. **4-bit Microprocessor:** The first microprocessor (Intel 4004) was invented in 1971. It was a 4-bit calculation device with a speed of 108 kHz. It has 3200 PMOS transistors. It is a 4-bit device used in calculator.
2. **8-Bit microprocessor:** In 1972, Intel came out with the 8008 which is 8-bit. In 1974, Intel announced the 8080 followed by 8085 is a 8-bit processor Because 8085 processor has 8 bit ALU. The 8080 is referred to as a "Second generation Microprocessor".

   Limitations of 8 Bit microprocessor:

   - Low speed of execution
   - Low memory addressing capability
   - Limited number of general-purpose registers
   - Less power full instruction set 4.
3. **16-bit Microprocessor:** Similarly, 8086 processor has 16-bit ALU. This had a larger instruction set then 8080. used NMOS transistors, so it operated much faster than the 8008.

Examples for 4/ 8 / 16 / 32-bit Microprocessors:

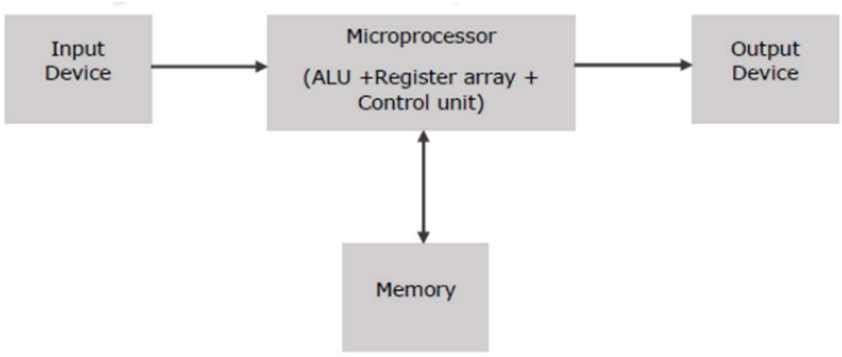
- 4-Bit processor – 4004/4040
- 8-bit Processor - 8085 / Z80 / 6800
- 16-bit Processor - 8086 / 68000 / Z8000
- 32-bit Processor - 80386 / 80486

What are 1st / 2nd / 3rd / 4th generation processor?

- The processor made of PMOS technology is called 1st generation processor, and it is made up of 4 bits
- The processor made of NMOS technology is called 2nd generation processor, and it is made up of 8 bits
- The processor made of CMOS technology is called 3rd generation processor, and it is made up of 16 bits

- The processor made of HCMOS technology is called 4th generation processor, and it is made up of 32 bits (HCMOS : High-density n- type Complementary Metal Oxide Silicon field effect transistor)

Block diagram of microprocessor:



How does a Microprocessor Work?

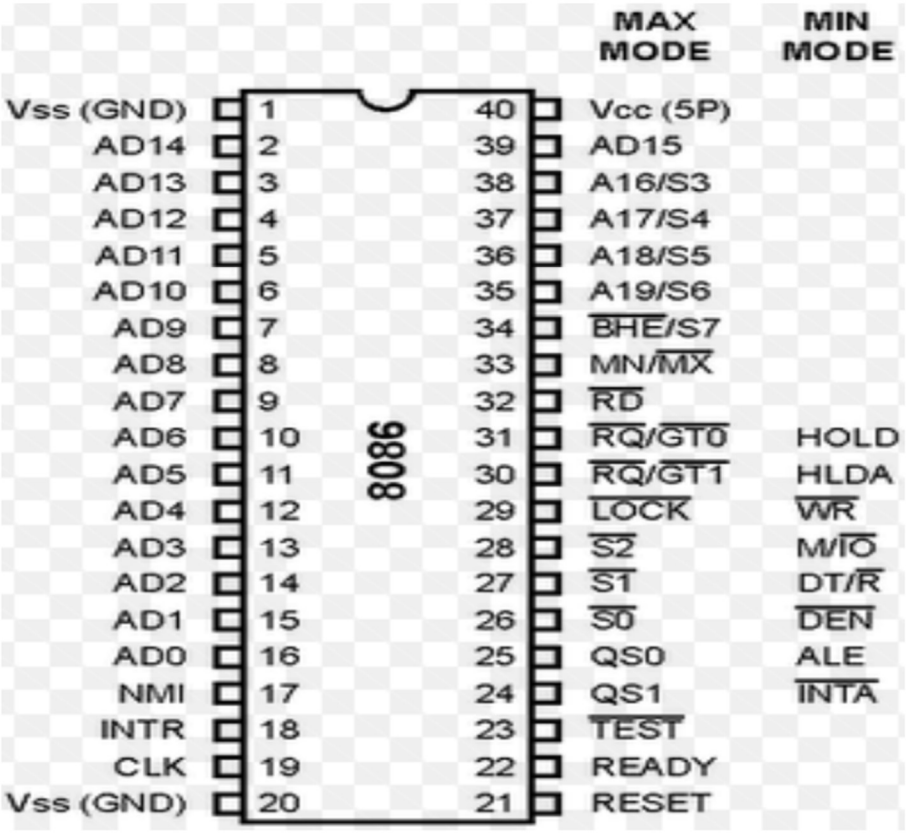
- The microprocessor follows a sequence: Fetch, Decode, and then Execute. Initially, the instructions are stored in the memory in a sequential order.
- The microprocessor fetches those instructions from the memory, then decodes it and executes those instructions till STOP instruction is reached.
- Later, it sends the result in binary to the output port. Between these processes, the register stores the temporarily data and ALU performs the computing functions.

8086 Microprocessor: Pin descriptions

- 8086 Microprocessor is an enhanced version of 8085 Microprocessor that was designed by Intel in 1976.
- It is a 16-bit Microprocessor having 20 address lines and 16 data lines that provides up to 1MB storage.
- It consists of powerful instruction set, which provides operations like multiplication and division easily.
- It supports two modes of operation, i.e. Maximum mode and Minimum mode. Maximum mode is suitable for system having multiple processors and Minimum mode is suitable for system having a single processor.

8086 Pin Diagram:

- 8086 was the first 16-bit microprocessor available in 40-pin DIP (Dual Inline Package) chip. Here is the pin diagram of 8086 microprocessor.



Power supply and frequency signals:

- It uses 5V DC supply at VCC pin 40, and uses ground at VSS pin 1 and 20 for its operation

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.

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.

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.

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.

Read($\overline{RD}$):

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

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

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.

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.

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.

$\overline{TEST}$:

- 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.

MN/$\overline{MX}$:

- 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-aversa.

INTA:

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

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.

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.

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.

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.

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.

HLDA:

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

HOLD:

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

QS1 and QS0:

- 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

S0, S1, S2:

- 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

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.

RQ/GT1 and RQ/GT0:

- 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/GT0 has a higher priority than RQ/GT1

Silent features of 8086.

1) It is a 16-bit Microprocessor.

2) 8086 has a 20-bit address bus that can access upto 220 memory locations (1 MB).

3) It has two blocks: BIU and EU.

4) It provides 16-bit registers. AX, BX, CX, DX, CS, SS, DS, ES, BP, SP, SI, DI, IP & FLAG REGISTER.

5) It has multiplexed address and data bus i.e. AD0- AD15 and A16 – A19.

6) It works in a multiprocessor environment. Control signals are generated by an external BUS Controller.

7) 8086 is designed to operate in two modes, Minimum and Maximum.

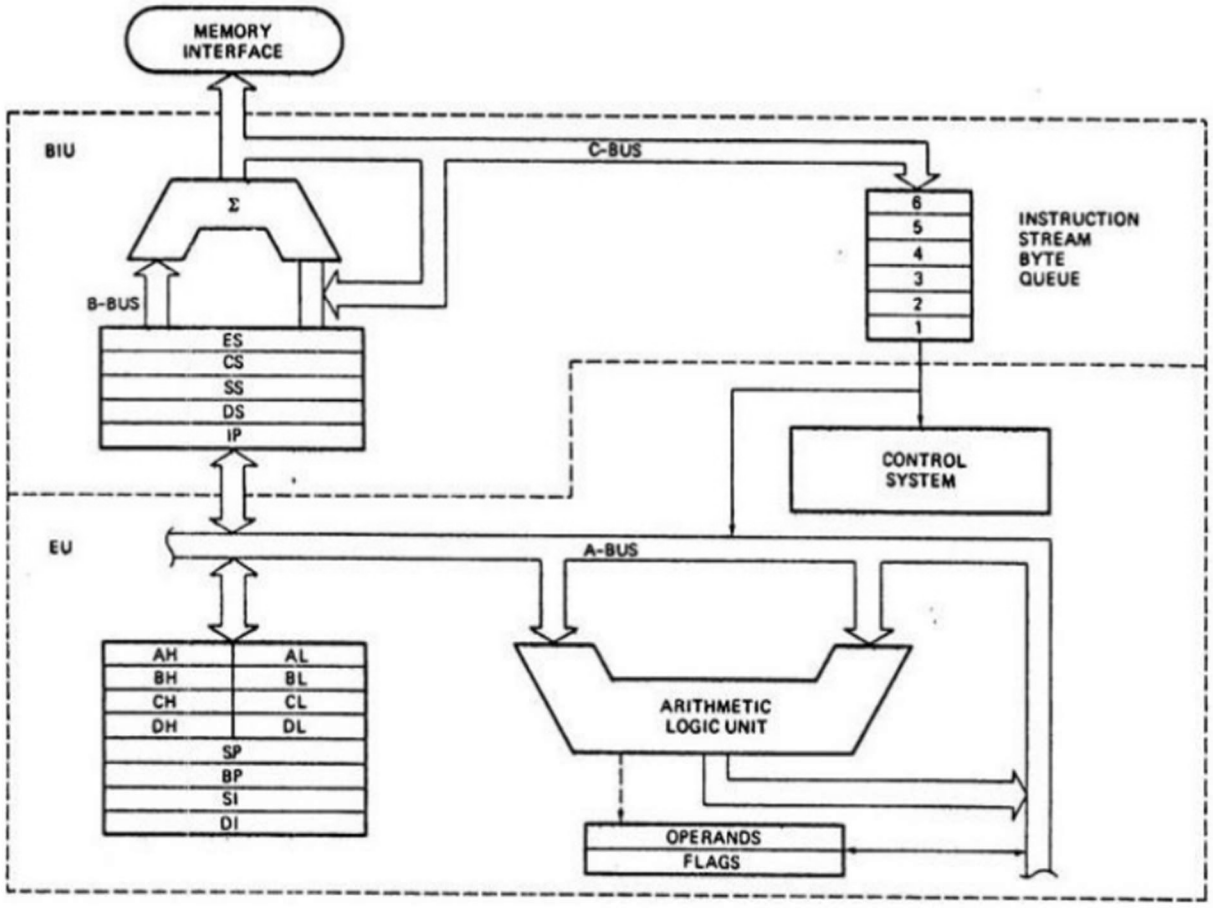
8) It can prefetches up to 6 instruction bytes from memory and queues them in order to speed up instruction execution.

9) Interrupts: - 8086 has 256 vectored interrupts.

10) Provides separate instructions for string manipulation.

11) Operating frequency range is 6-10MHz.

Architecture of 8086.



- 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 starting 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

- 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

1. **Carry flag** − This flag indicates an overflow condition for arithmetic operations
2. **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.
3. **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.
4. **Zero flag** − This flag is set to 1 when the result of arithmetic or logical operation is zero else it is set to 0
5. **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
6. **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 flag.

1. **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.
2. **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

3. **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 16 bit 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.

Pointers and Index Registers:

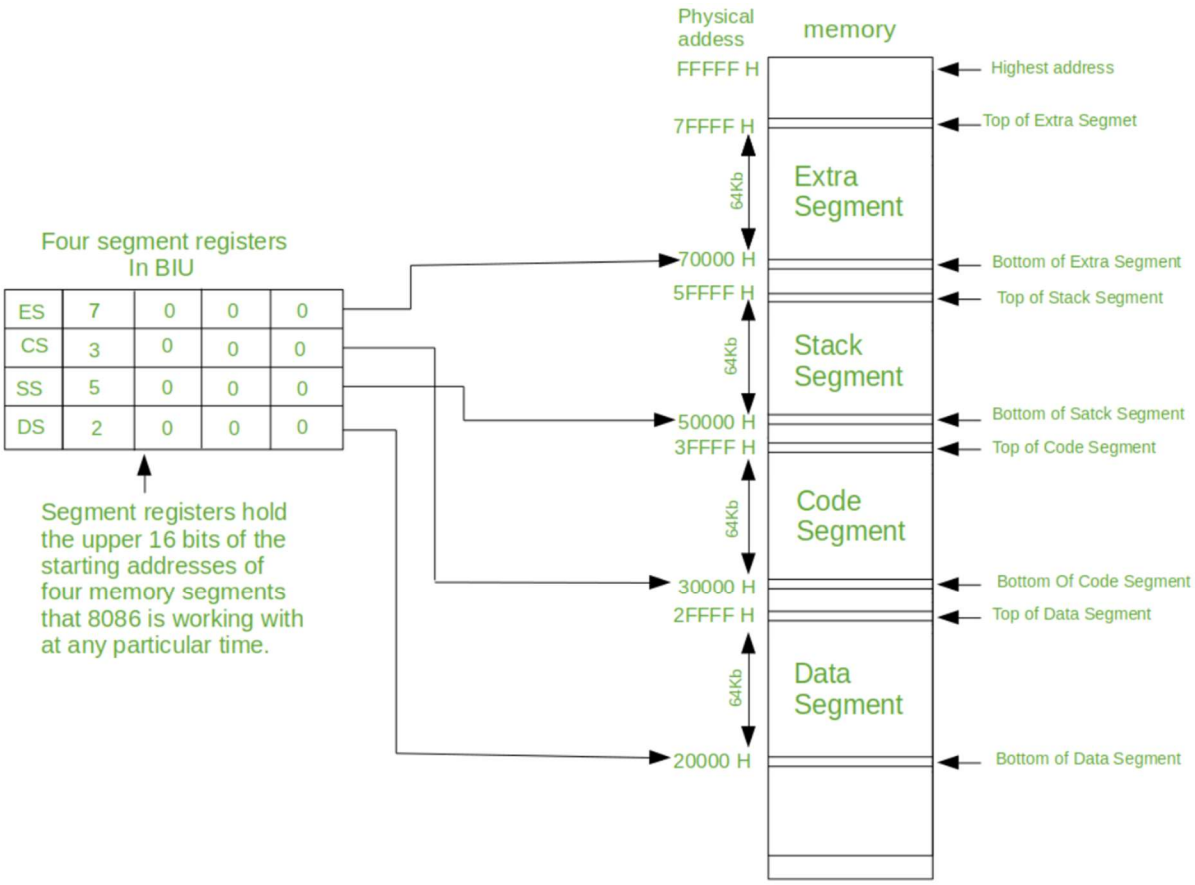
- SP and BP are pointer registers which holds the 16 bit offset address within the particular segment.
- SI and DI are the index registers.
- During the execution of string related instructions SI is used to store the offset of source data in data segment and DI is used to store the offset of destination data in data or extra segment.
- **Instruction pointer –** It is a 16-bit register used to hold the address of the next instruction to be executed.

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.
- **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 andis 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.

Memory segmentation

- Memory segmentation: Segmentation is the process in which the main memory of a computer is logically divided into the different segments and each segment has its own base address.
- It is basically used to increase the speed of the execution of the computer system
- Processor is able to fetch and execute the data from the memory easily and fast.
- Memory in 8086 is segmented in different segments and this memory management techniques are called as segmentation.
- The complete physical memory is divided into the number of logical segments. Size of each segment is 64Kbyte.
- These segments are addressed by one of the segment register that are DS,CS,SS and ES.
- These segment registers holds the starting address of a particular segments.
- The CPU of 8086 is able to address the 1Mbyte of physical memory
- The complete 1 Mbyte memory can be divided into the 16 segments each of 64 Kbyte in size.



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.

Concepts of pipelining in 8086:

- Pipelining is simply pre-fetching instruction and lining up them in queue.
- The process of fetching the next instruction when the present instruction is being executed is called as pipelining.
- Pipelining has become possible due to the use of 6 byte queue.
- BIU (Bus Interfacing Unit) fills in the queue until the entire queue is full. BIU restarts filling in the queue when at least two locations of queue are vacant.
- The Technique used to enable an instruction to complete with each clock cycle is called as pipelining.
- In Pipelined processor, the fetch, decode and execute operation are performed in parallel.
- So, pipelining improve the execution speed of microprocessor.

	cycle 1	cycle 2	cycle 3	cycle 4	cycle 5	cycle 6
Instruction 1	Fetch	Decode	Execute			
Instruction 2		Fetch	Decode	Execute		
Instruction 3			Fetch	Decode	Execute	
Instruction 4				Fetch	Decode	Execute

Advantages of Pipelining

- Pipelining enables many instructions to be executed at the same time.
- It allows execution to be done in fewer cycles.
- Speed up the execution speed of the processor.
- More efficient use of processor.
- The execution unit always reads the next instruction byte from the queue in BIU. This is faster than sending out an address to the memory and waiting for the next instruction byte to come.
- Pipelining removes the waiting time of EU and speeds up the processing. The 8086 BIU will not initiate a fetch unless and until there are two empty bytes in its queue. 8086 BIU normally obtains two instruction bytes per fetch.

Interrupts in 8086

- An interrupt is a condition that halts the microprocessor temporarily to work on a different task and then return to its previous task.
- Interrupt is an event or signal that request to attention of CPU. This halt allows peripheral devices to access the microprocessor.
- Whenever an interrupt occurs the processor completes the execution of the current instruction and starts the execution of an Interrupt Service Routine (ISR) or Interrupt Handler.
- ISR is a program that tells the processor what to do when the interrupt occurs. After the execution of ISR, control returns back to the main routine where it was interrupted.
- The different types of interrupts present in 8086 microprocessor are given by

1) Hardware Interrupts

- Hardware interrupts are those interrupts which are caused by any peripheral device by sending a signal through a specified pin to the microprocessor.
- There are two hardware interrupts in 8086 microprocessor.
- **NMI (Non Mask able Interrupt)** –
    - It is a single pin non mask able hardware interrupt which cannot be disabled. It is the highest priority interrupt in 8086 microprocessor.
    - After its execution, this interrupt generates a TYPE 2 interrupt. IP is loaded from word location 00008 H and CS is loaded from the word location 0000A H
- **INTR (Interrupt Request)** –
    - It provides a single interrupt request and is activated by I/O port.
    - This interrupt can be masked or delayed.
    - It is a level triggered interrupt.
    - It can receive any interrupt type, so the value of IP and CS will change on the interrupt type received.

2) Software Interrupts

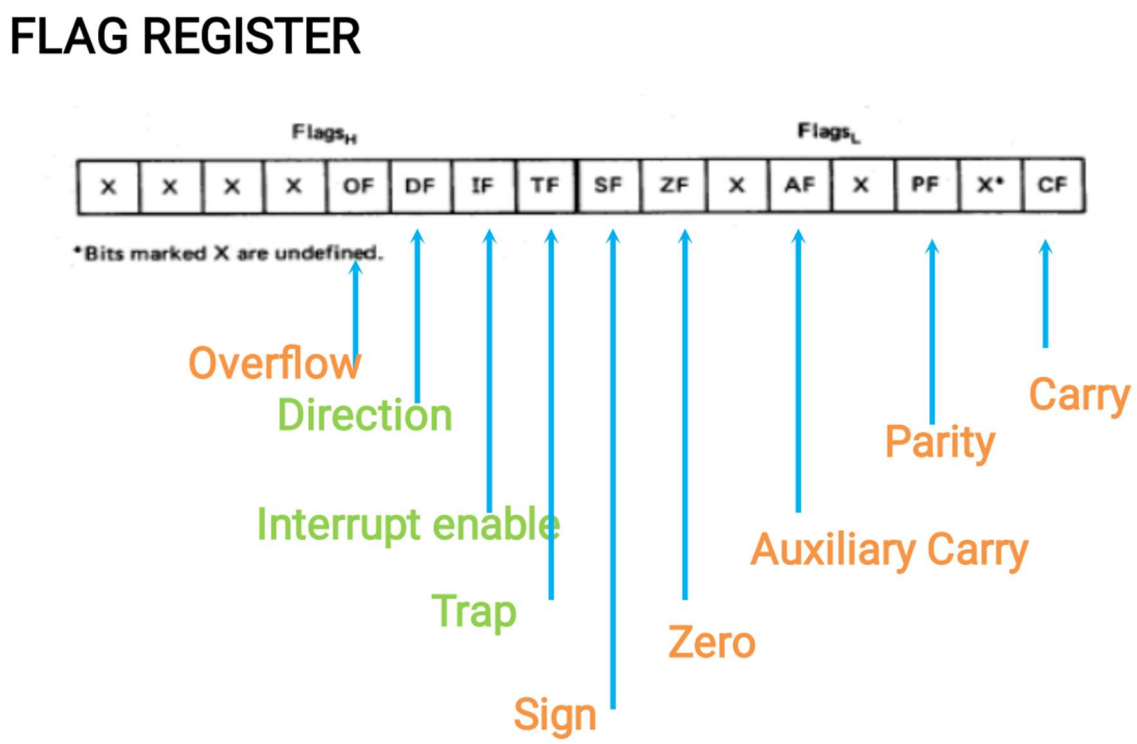
- These are instructions that are inserted within the program to generate interrupts.
- There are 256 software interrupts in 8086 microprocessor.
- Some important software interrupts are:
    - **TYPE 0** corresponds to division by zero(0).
    - **TYPE 1** is used for single step execution for debugging of program

- **TYPE 2** represents NMI and is used in power failure conditions.
- **TYPE 3** represents a break-point interrupt.
- **TYPE 4** is the overflow interrupt.

Difference between minimum and maximum mode:

Sr. No.	Minimum mode	Maximum mode
1.	MN/$\overline{MX}$ pin is connected to VCC i.e. MN/$\overline{MX}$ = 1.	MN/$\overline{MX}$ pin is connected to ground i.e. MN/$\overline{MX}$ = 0.
2.	Control system M/$\overline{IO}$, $\overline{RD}$, $\overline{WR}$ is available on 8086 directly.	Control system M/$\overline{IO}$, $\overline{RD}$, $\overline{WR}$ is not available directly in 8086.
3.	Single processor in the minimum mode system.	Multiprocessor configuration in maximum mode system.
4.	In this mode, no separate bus controller is required.	Separate bus controller (8288) is required in maximum mode.
5.	Control signals such as $\overline{IOR}$, $\overline{IOW}$, $\overline{MEMW}$, $\overline{MEMR}$ can be generated using control signals M/$\overline{IO}$, $\overline{RD}$, $\overline{WR}$ which are available on 8086 directly.	Control signals such as $\overline{MRDC}$, $\overline{MWTC}$, $\overline{AMWC}$, $\overline{IORC}$, $\overline{IOWC}$ and $\overline{AIOWC}$ are generated by bus controller 8288.
6.	ALE, $\overline{DEN}$, DT/$\overline{R}$ and $\overline{INTA}$ signals are directly available.	ALE, $\overline{DEN}$, DT/$\overline{R}$ and $\overline{INTA}$ signals are not directly available and are generated by bus controller 8288.
7.	HOLD and HLDA signals are available to interface another master in system such as DMA controller.	$\overline{RQ}$/$\overline{GT0}$ and $\overline{RQ}$/$\overline{GT1}$ signals are available to interface another master in system such as DMA controller and coprocessor 8087.
8.	Status of the instruction queue is not available.	Status of the instruction queue is available on pins QS0 and QS1 .

Flag Registers:



- 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 flag

- **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.