Module_1

1. Describe the Block Level Description of the Functional Units.
Block Level Description of the Functional Units

A computer is not a single unit but it consists of many functional units:

- Input unit
- Central Processing Unit(ALU and Control Unit),
- Storage (Memory) Unit
- Output Unit

The CPU is the brain of the computer. Controls the operation of the computer and performs its data processing functions; often simply referred to as processor. The CPU is made up of these main parts:

1. Instruction register – It holds the instruction to be executed.

2. Decoder – It decodes (converts to machine level language) the instruction and sends to the ALU (Arithmetic Logic Unit).

3. ALU – It has necessary circuits to perform arithmetic,logical, memory, register and program sequencing operations.

---

4. Register – It holds intermediate results obtained during program processing.

Registers are used for holding such results rather than RAM because accessing registers is almost 10 times faster than accessing RAM.

Memory

Microprocessor has two types of memory
RAM – Random Access Memory is volatile memory that gets erased when power is switched off. All data and instructions are stored in RAM.

ROM – Read Only Memory is non-volatile memory whose data remains intact even after power is switched off. Microprocessor can read from it any time it wants but cannot write to it. It is pre programmed with most

essential data like booting sequence by the manufacturer.

RAM is used for storage of data while ROM is used for storage of programs especially to the start-up program that runs when the microprocessor is powered on.

Input/output (I/O) devices

• I/O devices allow the computer to communicate with the outside world. • Input devices allow the user to enter data into the computer, such as a keyboard, mouse, and scanner.
• Its aim is to supply data (Alphanumeric, image , audio, video, etc.) to the computer for processing. The Input devices are keyboard, mouse, scanner, mic, camera. Output devices allow the computer to display information to the user, such
as a monitor and printer.
• I/O devices are connected to the computer through the system bus.

2. Explain The Von Neumann Model in detail.

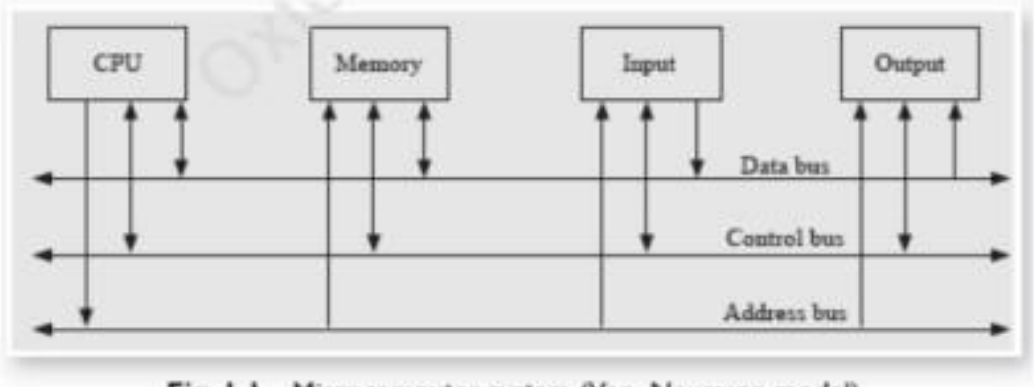
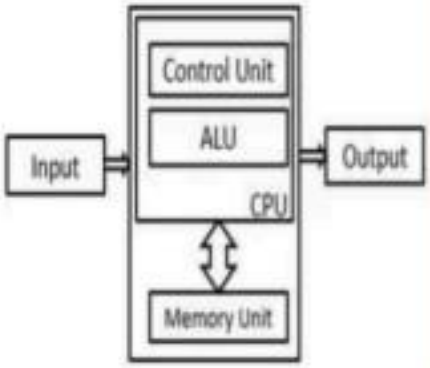


Fig. 1.1 Microcomputer system (Von–Neumann model)

Von Neumann Model

Neumann architecture is a theoretical computer design based on the concept of stored- program where programs and data are stored in the same.

The concept was designed by a mathematician John Von Neumann in 1945 and currently serves as the foundation of almost all modern computers. Neumann machine consists of a arithmetic/logic unit and a control unit, a memory, mass storage and input and output.



Von Neumann Model

Key Components of the Von Neumann Model:

Central Processing Unit (CPU):

- Control Unit (CU): Responsible for fetching, decoding, and executing instructions. It controls the operation of all other components.
- Arithmetic Logic Unit (ALU): Handles all mathematical calculations and logical operations.
- Memory Unit:The main memory stores both data and instructions in a single, unified addressable memory space. This is known as thestored-program concept, where programs and data share the same memory.
- Input/Output (I/O) System:Handles interaction with external devices like keyboards, monitors, printers, etc., allowing the computer to receive input from and send output to the user
- System Bus:A communication system that transfers data between the CPU, memory, and I/O devices. It typically consists of three types of buses:
- Data Bus: Carries data between components.
- Address Bus: Carries memory addresses to identify where data is located.
- Control Bus: Sends control signals for coordination among components.

How the Von Neumann Model Works:

- Fetch: The CPU retrieves an instruction from memory using the program counter (PC).
- Decode: The control unit interprets the fetched instruction.

- Execute: The ALU performs the required operation (e.g., arithmetic or logic).
- Store: The result is written back to memory if needed.

3. Compare the Computer organization and architecture.

S.No.	Computer Architecture	Computer Organization
1.	Architecture describes what the computer does.	The Organization describes how it does it.
2.	Computer Architecture deals with the functional behavior of computer systems.	Computer Organization deals with a structural relationship.
3.	In the above figure, it's clear that it deals with high-level design issues.	In the above figure, it's also clear that it deals with low-level design issues.
4.	Architecture indicates its hardware.	Whereas Organization indicates its performance.
5.	As a programmer, you can view architecture as a series of instructions, addressing modes, and registers.	The implementation of the architecture is called organization.
6.	For designing a computer, its architecture is fixed first.	For designing a computer, an organization is decided after its architecture.

7.	Computer Architecture is also called Instruction Set Architecture (ISA).	Computer Organization is frequently called microarchitecture.

8.	Computer Architecture comprises logical functions such as instruction sets, registers, data types, and addressing modes.	Computer Organization consists of physical units like circuit designs, peripherals, and adders.
9.	The different architectural categories found in our computer systems are as follows: 1. Von-Neumann Architecture 2. Harvard Architecture 3. Instruction Set Architecture 4. Micro-architecture 5. System Design	CPU organization is classified into three categories based on the number of address fields: 1. Organization of a single Accumulator. 2. Organization of general registers 3. Stack organization

10.	It makes the computer's hardware visible.	It offers details on how well the computer performs.
11.	Architecture coordinates the hardware and software of the system.	Computer Organization handles the segments of the network in a system.
12.	The software developer is aware of it.	It escapes the software programmer's detection.
13.	Examples- Intel and AMD created the x86 processor. Sun Microsystems and others created the SPARC processor. Apple, IBM, and Motorola created the PowerPC.	Organizational qualities include hardware elements that are invisible to the programmer, such as interfacing of computer and peripherals, memory technologies, and control signals.

4. Performance Measure of Computer Architecture.

In computer organization, performance refers to the speed and efficiency at which a computer system can execute tasks and process data. A high-performing computer system is one that can perform tasks quickly and efficiently while minimizing the amount of time and resources required to complete these tasks.

Performance of a computer depends on the constituent subsystems of the system including software.

Each of the subsystems can be measured and tuned for performance. Thus performance can be measured for:

- CPU performance for Scientific application, Vector processing, Business application, etc – Instructions per Second
- Graphics performance – Rendering - Pixels per second
- I/O Performance – Transactions Per Second
- Internet performance and more – bandwidth utilization in Mbps or Gbps

There are three metrics for any system performance measure and these are

- Performance or Response Time
- Execution Time
- Throughput.

Response time is the time from the start to completion of a task.

This also includes:

- Operating system overhead.
- Waiting for I/O and other processes
- Accessing disk and memory
- Time spent executing on the CPU or execution time.

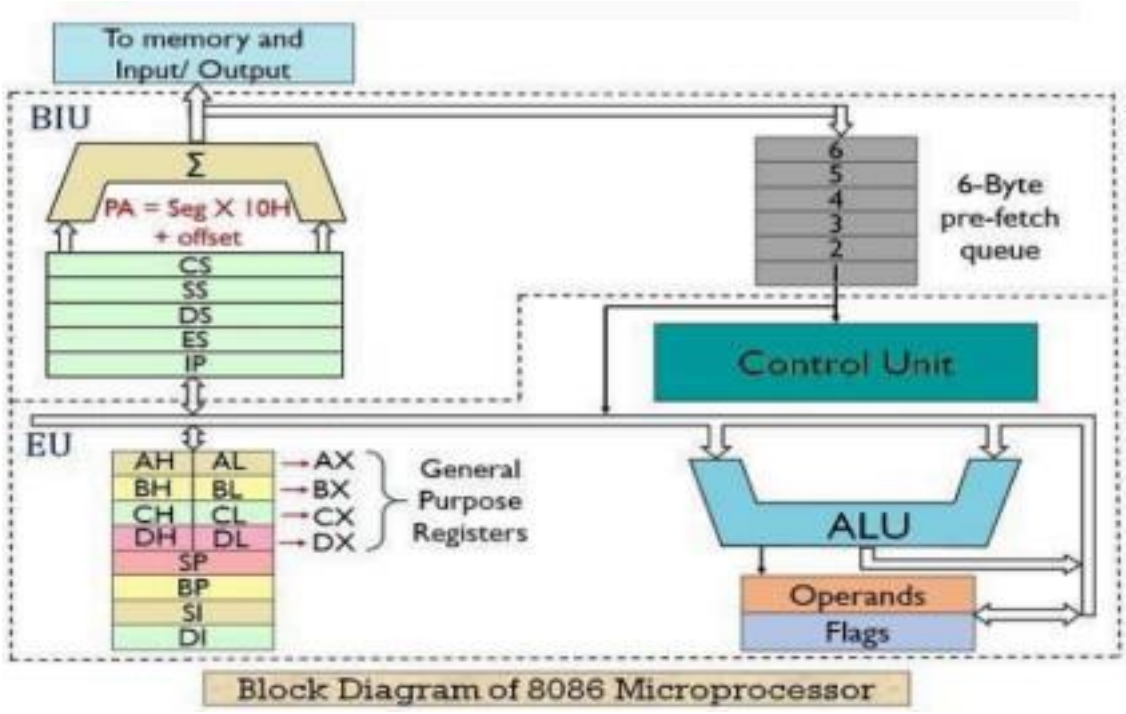
Throughput is the total amount of work done in a given time. Or the measure of work done in a unit of time.

CPU execution time is the total time a CPU spends computing on a given task.

5. Explain the Von Neumann Architecture in details, and list down the performance Measure of Computer Architecture.

Refer above questions.

6. Draw and Explain 8086 Microprocessor Architecture in detail.



Block Diagram of 8086 Microprocessor

Explain in detail

7. Explain Instruction Set for 8086 Microprocessor.

8086 Instruction Set

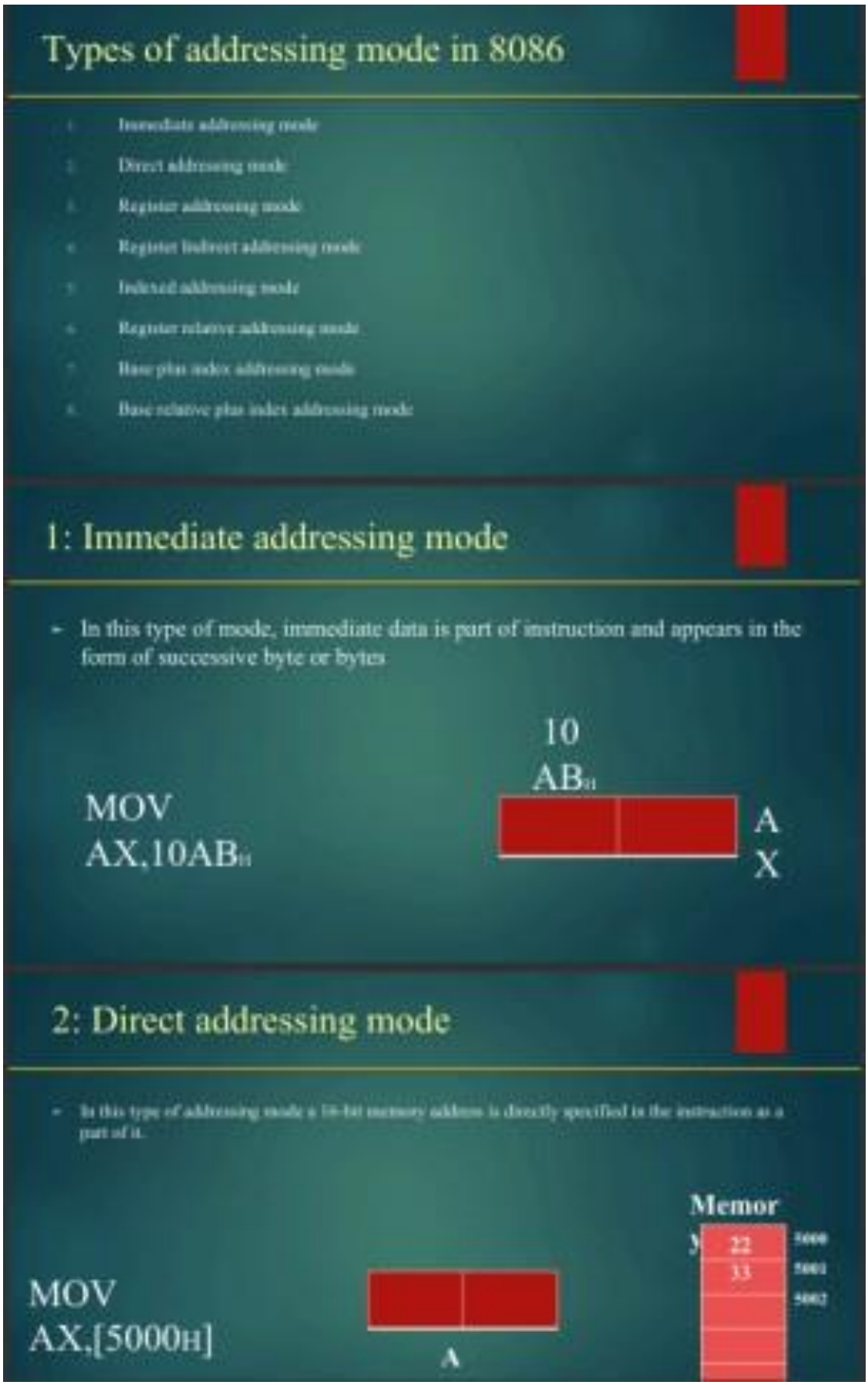
The Intel 8086 is a 16-bit microprocessor that was introduced in 1978. It is the first processor of the x86 family. The instruction set architecture of the 8086 CPU consists of instructions that a processor can execute. The 8086 instruction set is characterized by its versatility and efficiency, allowing programmers to write code for a wide range of applications. **Instructions are encoded in binary format and organized into different categories based on their functionality.** These instructions encompass various operations, including data movement, arithmetic and logic operations, control flow instructions, and input/output operations.

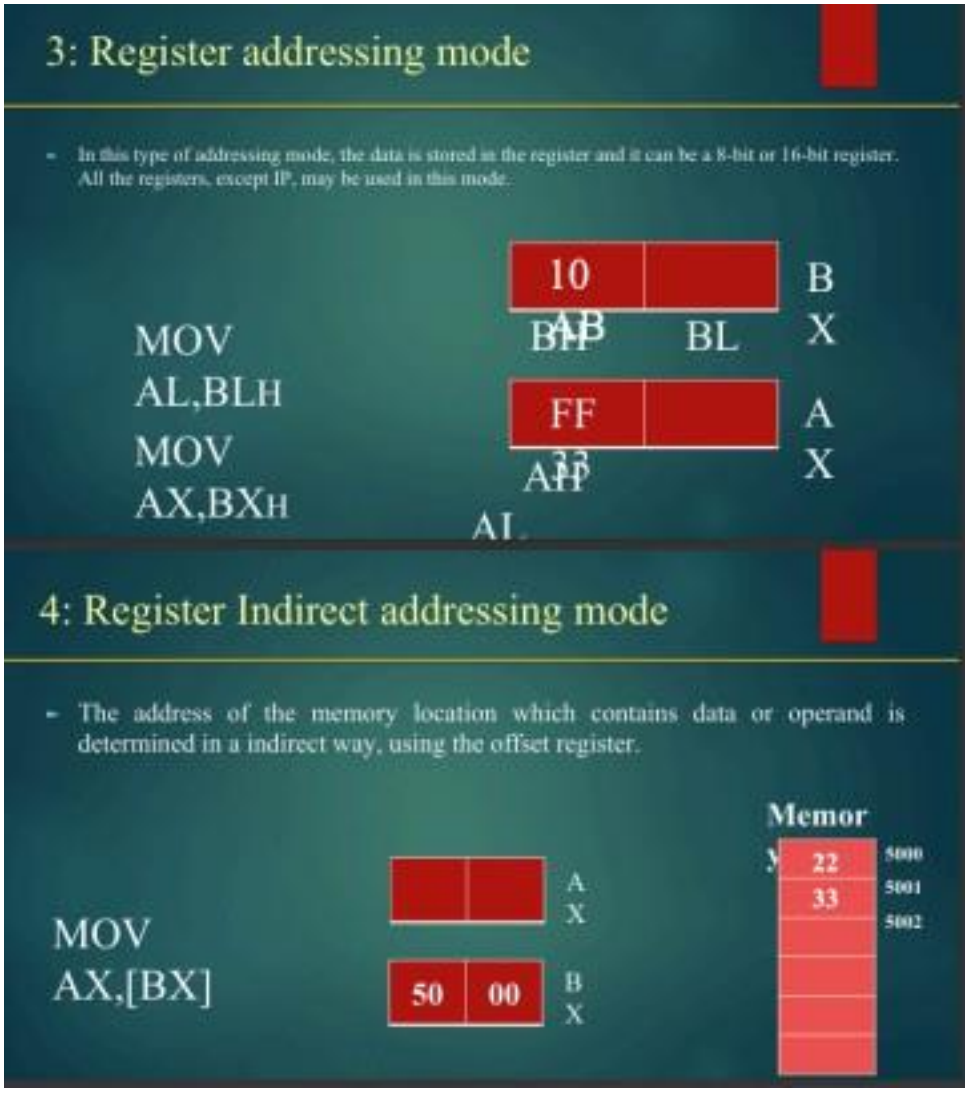
Below is an overview of the 8086 instruction set.

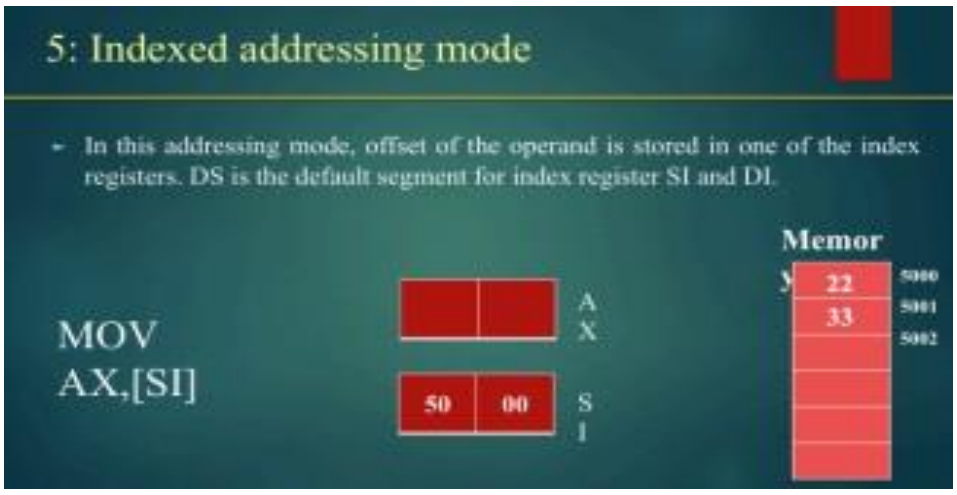
Data Movement Instructions

Instructions	Definition/Meaning
MOV	Transfer data from source to destination.
XCHG	Swap the contents of two registers or a register and a memory location.
PUSH	Push data onto the stack.
POP	Pop the data from the stack.
LEA	Load Effective Address (loads the address of a memory operand into a register).

8. Describe any 5 Addressing Mode.







9. Explain features of 8086microprocessor.

Features of 8086

1. It is a 16-bit Microprocessor (µp). It's ALU, internal registers works with 16bit binary word.

2. A 40 pin dual in line package.Have 29000Transistors.
3. 8086 has a 20 bit address bus can access up to 2^20= 1 MB memory locations. It segments it into 16, 64kB segments.
4. 8086 has a 16bit data bus. It can read or write data to a memory/port either 16bits or 8 bit at a time.
5. It can support up to 64K I/O ports.
6. It provides 14, 16 -bit registers.
7. Frequency range of 8086 is 6-10 MHz .(The Intel 8086 microprocessor operates at a frequency of 5 MHz.)

Features of 8086

1. It has multiplexed address and data bus AD0- AD15 and A16 – A19. 2. It requires single phase clock with 33% duty cycle to provide internal timing.
3. It can prefetch upto 6 instruction bytes from memory and queues them in order to speedup instruction execution.
4. It requires +5V power supply.
5. 8086 is designed to operate in two modes, Minimum mode and Maximum mode.
6. Address ranges from 00000H to FFFFFH

10. Explain Bus Interface unit.

The 8086 CPU is divided into two independent functional parts

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

1. The Bus Interface Unit (BIU):

It provides the **interface** of 8086 to **external memory and I/O devices via the System Bus**. It performs various machine cycles such as m**emory read, I/O read**, etc. to **transfer data between memory and I/O devices.**
BIU performs the following functions are as follows:

- It generates the 20-bit physical address for memory access.
- It fetches instructions from the memory.
- It transfers data to and from the memory and I/O.
- Maintains the 6-byte pre-fetch instruction queue(supports pipelining).

BIU mainly contains the

1. **4 Segment registers,**
2. **the Instruction Pointer,**
3. **a pre-fetch queue,**
4. **and an Address Generation Circuit.**

- The BIU sends out addresses, fetches instructions from memory, reads data from ports and memory, and writes data to ports and memory. Or we can say it reads the operational codes from memory and stores them in the instruction queue registers
- While the EU is decoding an instruction or executing an instruction, which does not require use of the buses,the BIU fetches up to six instruction bytes.
- The BIU stores these pre-fetched bytes in a first-in-first-out register set called a queue.
- Except in the case of JMP and CALL instructions, where the queue must be dumped and then reloaded Starting from a new address, this pre-fetch and queue scheme greatly speeds up processing.
- Fetching the next instruction while the current instruction executes is called pipelining.
- BIU contains Instruction queue, Segment registers, Instruction pointer, and Address adder.
- It provides a full 16 bit bidirectional data bus and 20 bit address bus. The bus interface unit is responsible for performing all external bus operations.
- In simple words, the BIU handles all transfers of data and addresses on the buses for the execution unit.

11. Explain Register organization in 8086 Microprocessor.

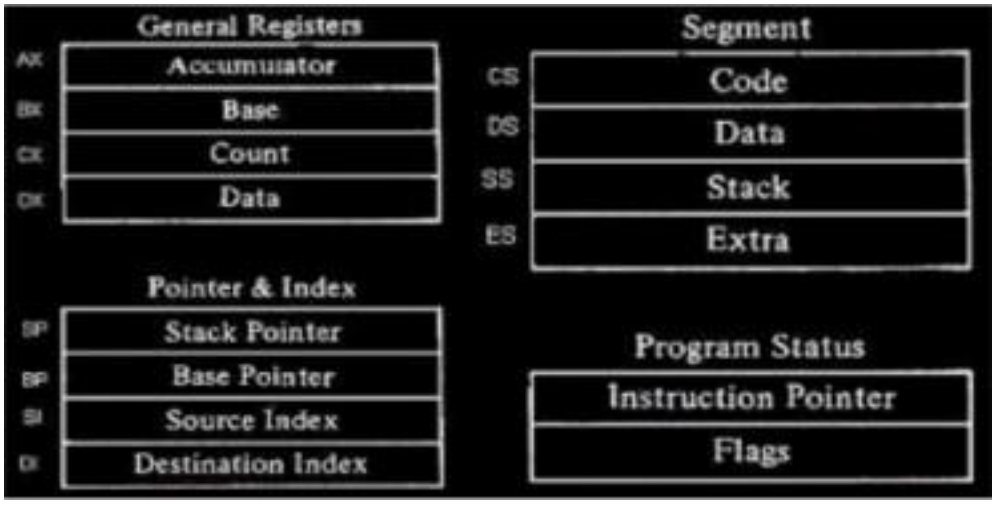
Register Organization:

The 8086 microprocessor has a total of fourteen registers that are accessible to the programmer. All of them are 16-bit registers. It is divided into two groups. They are:

1. General purpose registers:
These registers can be used as either 8-bit registers or 16-bit registers. They may be either used for holding data, variables and intermediate results temporarily or for other purposes like a counter or for storing offset address for some particular addressing modes etc.

2. Special purpose registers:
These registers are used as segment registers, pointers, index registers or as offset storage registers for particular addressing modes. The 8086 registers are classified into the following types:

1. General Purpose Registers
2. Segment Registers
3. Pointers and Index Registers
4. Flag Register



12. Explain Execution Unit.

EXECUTION UNIT :

- The Execution unit is responsible for decoding and executing all instructions.
- The EU extracts instructions from the top of the queue in the BIU, decodes them, generates operands if necessary, passes them to the BIU and requests it to perform the read or write cycles to memory or I/O and perform the operation specified by the instruction on the operands.
- During the execution of the instruction, the EU tests the status and control flags and updates them based on the results of executing the instruction.
- If the queue is empty, the EU waits for the next instruction byte to

be fetched and shifted to top of the queue.

- When the EU executes a branch or jump instruction, it transfers control to a location corresponding to another set of sequential instructions. Whenever this happens, the BIU automatically resets the queue and then begins to fetch instructions from this new location to refill the queue.

EXECUTION UNIT :

- The execution unit of the 8086 tells the BIU where to fetch instructions or data from, decodes instructions,and executes instructions.
- The EU contains control circuitry, which directs internal operations.
- The EU has a 16-bit arithmetic logic unit (ALU) which can add, subtract, AND, OR, XOR, increment, decrement,complement or shift binary numbers.

13. Explain Flag Register of 8086 Microprocessor.

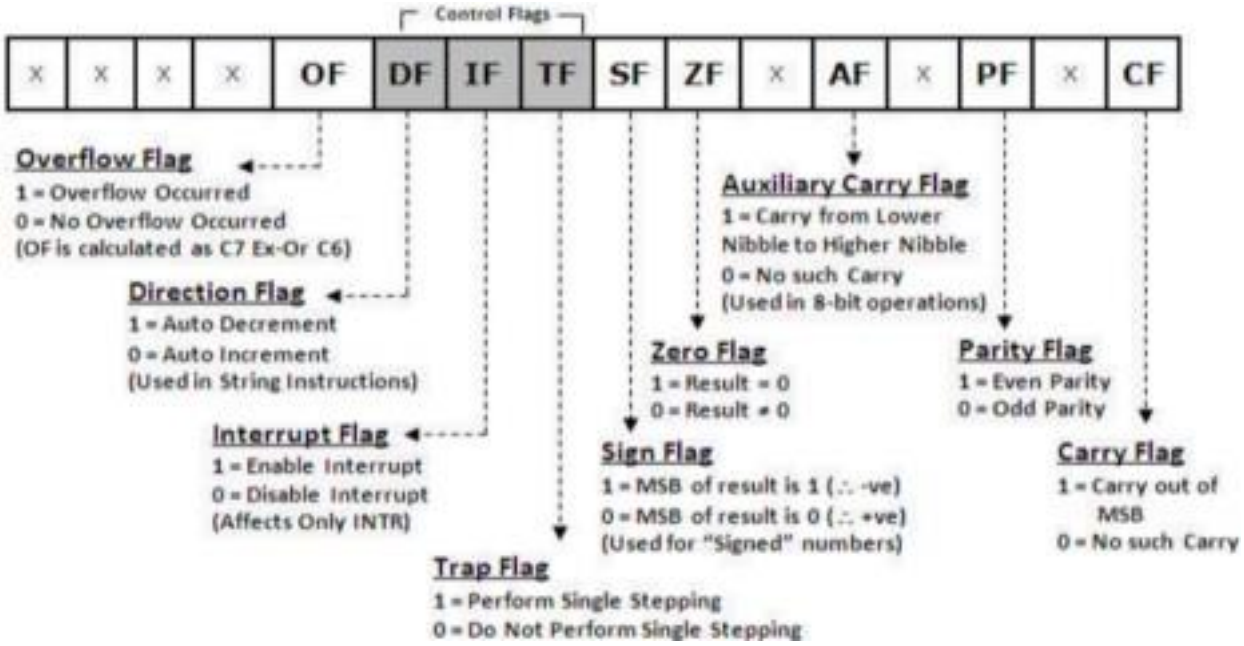
Flag Register contains a group of status bits called flags that indicate the status of the
CPU or the result of arithmetic operations. There are two types of flags:

1. **The status flags** which reflect the **result of executing an instruction**. The programmer cannot set/reset these flags directly.
2. **The control flags enable or disable certain CPU operations**. The programmer can set/reset these bits to control the CPU's operation.

Nine individual bits of the status register are used as **control flags (3 of them)** and **status flags (6 of them)**.The remaining **7 are not used.**

A flag can only take on the values **0 and 1**. We say a **flag is set if it has the value 1.**

The **status flags** are used to **record specific characteristics of arithmetic and of logical instructions.**



Module_2

1. Explain various Instruction Formats.

The forms for the instructions are a series of bits (0 and 1). When these pieces are combined, they form fields. The CPU receives precise information about the operation and location of the data from each field of the machine.

The bit configuration for an instruction is also specified by the instruction format. It may have a variety of locations and be of varying lengths. The address elements in the instruction format change depending on how the CPU's registers are set up. The CPU's supported file formats rely on the Instructions Set Architecture the processor has put in place. Depending on the multiple address fields, the instruction is categorized as follows:

- Zero address instruction
- One address instructions
- Two address instruction
- Three address instruction

Explain about all 4 in short.

2. Draw and Explain Basic Instruction Cycle with Interrupt Processing. **Instruction Cycle**

A program residing in the memory unit of the computer consists of a sequence of instructions.

The program is executed in the computer by going through a cycle for each

instruction.
Each instruction cycle in turn is subdivided into a sequence of sub-cycles or phases.

Instruction Cycle:

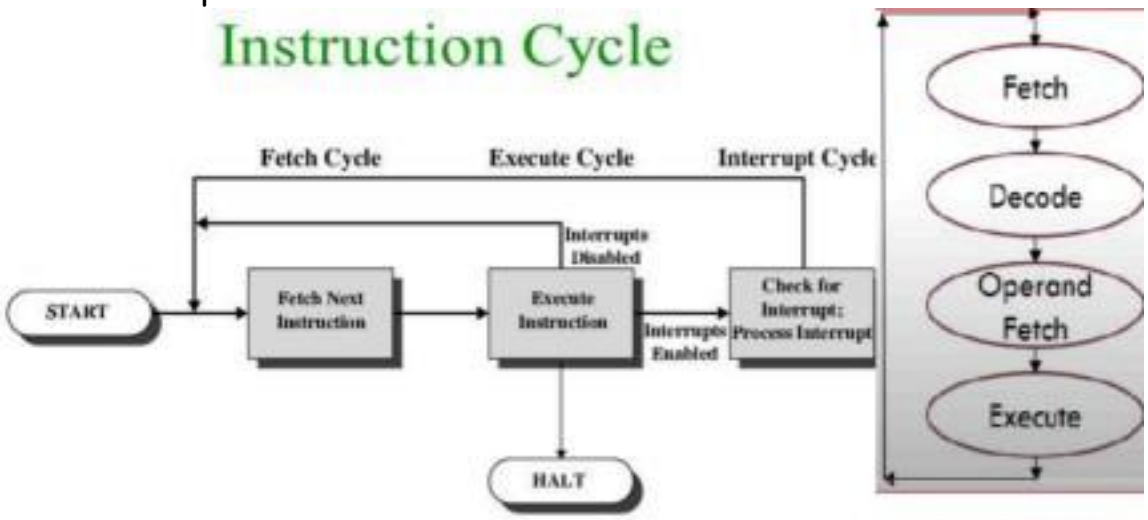
In the basic computer each instruction cycle consists of the following phases:

Fetch an instruction from memory

2. **Decode:** Decode the instruction Decode
3. **Operand Fetch:** Read the effective address from memory if the instruction has a indirect address Operand
4. **Execute:** Execute the instruction

Upon the completion of step 4, the control goes back to step I to fetch, decode and execute the next instruction. This process continues indefinitely unless a HALT instruction is encountered.

To perform fetch, decode and execute cycles the processor has to perform set of microperations.



Instuction cycle with Interrupt

Basic Instuction cycle

3. Given X =A [B+C(D+E)]/ F(G+H) Write the program to evaluate the expression , using the one address instructions.

Program using one address instructions

LOAD H ; AC← M[H]

ADD G ; AC←AC + M[G]

MUL F ; AC←AC*M[F]

STORE T ; M[T]←AC

LOAD D ; AC← M[D]

ADD E ; AC←AC + M[E]

ADD B ; AC←AC + M[B]

MUL A ; AC←AC* M[A]

DIV T ; AC←AC/M[T]

STORE X ; M[X] ← AC

4. Example :X = A×B+C×C

Explain how the above expression will be executed in one address, two address and three address processors in an accumulator organization.

Example 3 X = A×B+C×C

Explain how the above expression will be executed in one address, two address and three address processors in an accumulator organization.

Solution :

Using one address instruction			Using two address instructions			Using three address instructions		
LOAD A	;	AC← M[A]	MOV A, R1	;	R1←M[A]	MUL A, B, R1	;	R1←M[A]*M[B]
MUL B	;	AC←AC* M[B]	MUL B, R1	;	R1←R1*M[B]	MUL C, C, R2	;	R2←M[C]+M[C]
STORE T	;	M[T] ←AC	MOV C, R2	;	R2←M[C]			
LOAD C	;	AC← M[C]	MUL C, R2	;	R2←R2*M[C]	Add R1, R1, X	;	M[X] ←R1*R2
MUL C	;	AC←AC* M[C]	ADD R2, R1	;	R1←R1+R2			
ADD T	;	AC←AC + M[T]	MOV R1, X	;	M[T]←R1			
STORE X	;	M[X] ← AC						

5. What are the stages involved in the Basic Instruction Cycle explain in detail with diagram.

Refer to 2nd question

6. Briefly illustrate the instruction cycle state diagram with interrupts.

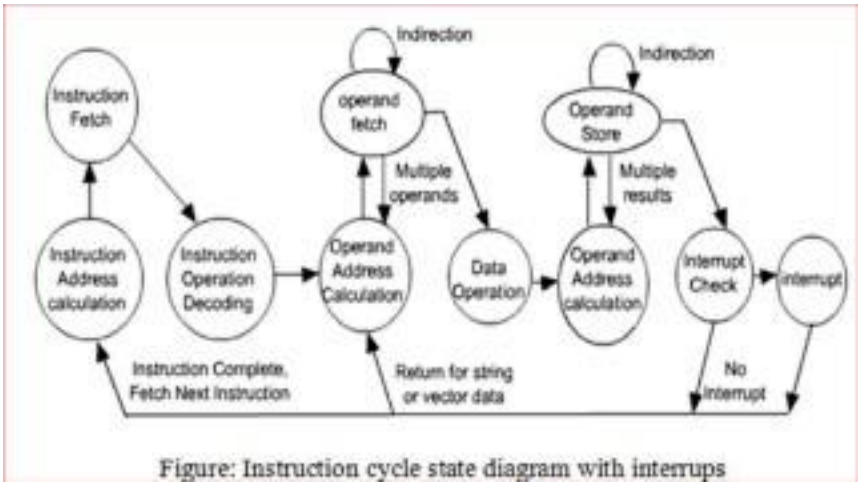


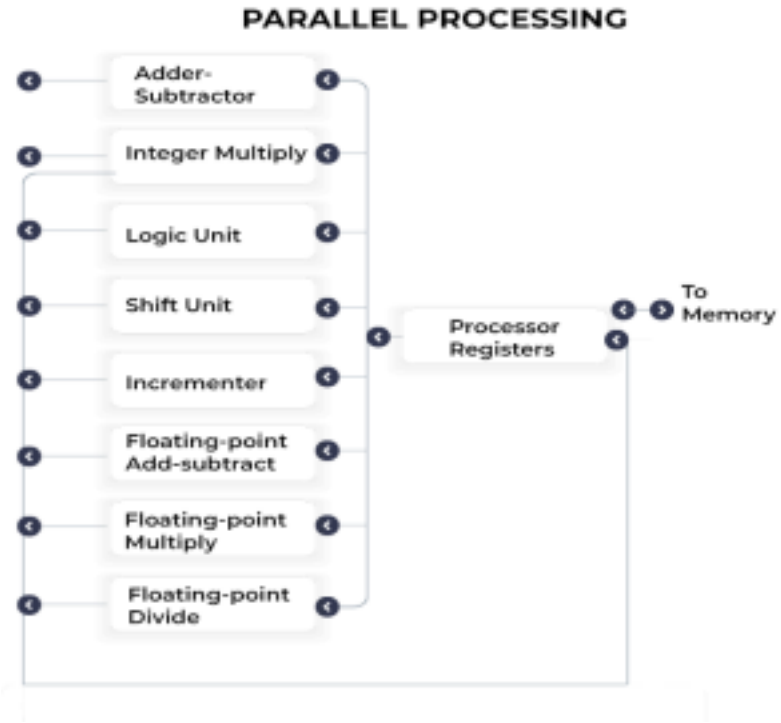
Figure: Instruction cycle state diagram with interrups

7. Explain Flynn's Classifications, in detail.

Flynn's Taxonomy

- Michael Flynn classified computers according to their type of parallelism:
  - **SISD** – Single Instruction Single Data – simple computers that are essentially devoid of parallelism
  - **SIMD** – Single Instruction Multiple Data – processors capable of performing the same operation on multiple pairs of operands
  - **MISD** – Multiple Instruction Single Data – performing several operations on the same set of data – only of theoretical interest
  - **MIMD** - Multiple Instruction Multiple Data – capable of processing several programs simultaneously on different sets of data

8. Explain the concept of data parallelism and its relationship to Flynn's classifications.



Parallel processing is a computing technique when multiple streams of calculations or data processing tasks co-occur through numerous central processing units (CPUs) working concurrently.

Parallel Processing

- Parallel processing – denotes the use of techniques designed to perform various data processing tasks simultaneously to increase a computer's overall speed.
- These techniques can include:
  - performing arithmetic or logical operations while fetching the next instruction
  - executing several instructions at the same time
  - performing arithmetic or logical operations on multiple sets of operands.
- While parallel processing can be more expensive, technological advances have dropped to overall cost of processor design enough to make it financially feasible.

9. Illustrate the 6 stage Pipelining.

Pipelining

- Pipelining is a technique where sequential processes are broken down into separate suboperations, each of which being performed by its own hardware.
- Each computation is passed along to the next segment in the pipeline, with the processes are carried in a manner analogous to an assembly line.
- The fact that each suboperation is performed by different hardware allows different stages of the overall operation to be performed in parallel.

Pipelining is a technique where multiple instruction stages (or segments) are overlapped during execution. This is similar to an assembly line in a factory where different stages of production occur simultaneously for different items.

+ Additional Stages

- Fetch instruction (FI)
  - Read the next expected instruction into a buffer
- Decode instruction (DI)
  - Determine the opcode and the operand specifiers
- Calculate operands (CO)
  - Calculate the effective address of each source operand
  - This may involve displacement, register indirect, indirect, or other forms of address calculation
- Fetch operands (FO)
  - Fetch each operand from memory
  - Operands in registers need not be fetched
- Execute instruction (EI)
  - Perform the indicated operation and store the result, if any, in the specified destination operand location
- Write operand (WO)
  - Store the result in memory

10. Explain the Control Unit: Soft Wired (Micro programmed).
Refer notes

11. Explain the Control Unit: Hardwired Control Unit.
Refer notes

12. State the difference between Soft wired and Hardwired Control units.

	Hardwired Control Unit	**Micro-programmed Control Unit**
Implementation	Fixed set of logic gates and circuits	Microcode stored in memory
Flexibility	Less flexible, difficult to modify	More flexible, easier to modify
Instruction Set	Supports limited instruction sets	Supports complex instruction sets
Complexity of Design	Simple design, easy to implement	Complex design, more difficult to implement
Speed	Fast operation	Slower operation due to microcode decoding
Debugging and Testing	Difficult to debug and test	Easier to debug and test
Size and Cost	Smaller size, lower cost	Larger size, higher cost

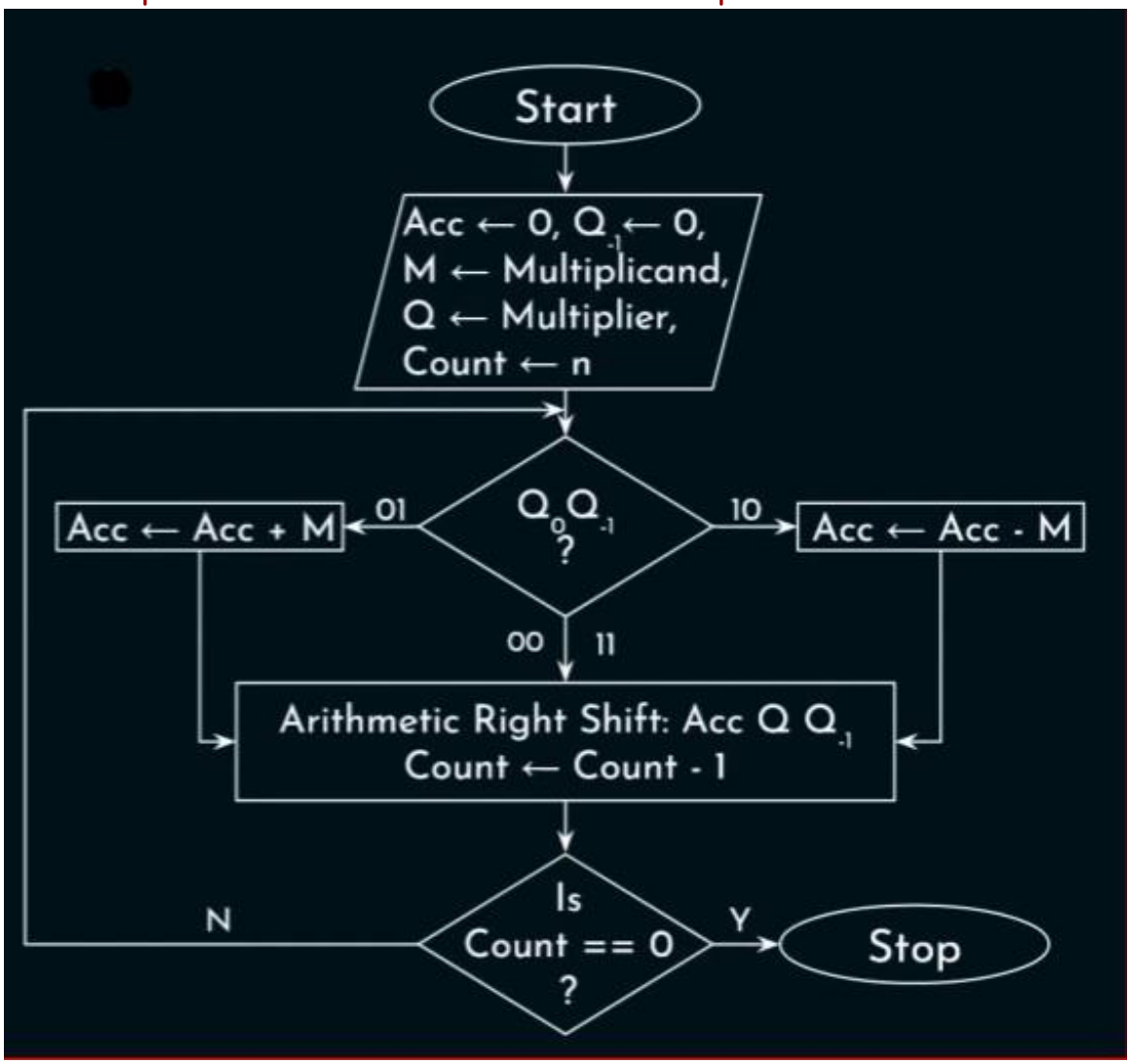
Maintenance and Upgradability	Difficult to upgrade and maintain	Easier to upgrade and maintain

Module_3

1. Explain Digital Circuits: NOT, AND, OR, NAND, NOR, EX-OR, EX-NOR Gates.

Refer notes

2. Explain the flowchart for Booth's multiplication.



Step-by-Step Explanation of Booth's Algorithm

1. Initialization:

- Acc (Accumulator)** is initialized to `0`.
- Q (Multiplier)** is set to the binary value of the multiplier. - M (Multiplicand)** is set to the binary value of the multiplicand. - Q-1 (Previous Bit)** is initialized to `0`.
- Count is initialized to the number of bits in the multiplier (`n`).

2. Evaluate the Current Pair of Bits (Q0 and Q-1):

- Q0 (Least Significant Bit of Q)** and **Q-1** are the bits that determine the operation:
- If Q0 and Q-1 = 01:** Perform `Acc = Acc + M` (Add the multiplicand to the accumulator).
- If Q0 and Q-1 = 10:** Perform `Acc = Acc - M` (Subtract the multiplicand from the accumulator).
- If Q0 and Q-1 = 00 or 11:** No addition or subtraction is done.

3. Arithmetic Right Shift:

- After the addition or subtraction operation (if any), perform an **Arithmetic Right Shift** on the concatenated value of `Acc`, `Q`, and `Q-1`. - This operation shifts all bits to the right, preserving the sign bit of `Acc`.
- The count is decremented by 1 after the shift.

4. Check the Count:

- f Count = 0: The algorithm stops. The final result of the multiplication is stored in the combined register `[Acc Q]`.
- If Count ≠ 0: The algorithm goes back to step 2, continuing the process until the count reaches 0.

5. End of the Algorithm:

- The loop continues until all the bits of the multiplier have been

processed (i.e.,
`Count` becomes 0).
- The final product is found in the combined content of the `Acc` and `Q` registers.

3.Questions on Number system conversions. Various numerical can be asked.
Refer notes