QB.COA-M1

Q1>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.

Q2>2. Explain The Von Neumann Model in detail.

Neumann architecture is a theoretical computer design based on the

concept of stored- program where
programs and data are stored in the same. memory.
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 central processor with an 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 the stored-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.

Advantages of the Von Neumann Model:

Simplicity: Its straightforward design makes it easier to build and understand.

Flexibility: Can run various programs by changing the stored instructions.

Efficiency: Sharing the same memory for instructions and data reduces hardware complexity.

Limitations:

Von Neumann Bottleneck: The CPU and memory share the same bus, causing delays because only one piece of data or instruction can be accessed at a time. This limits the system's speed.

Shared Memory: Both data and instructions can be accidentally modified, leading to errors or potential security issues.

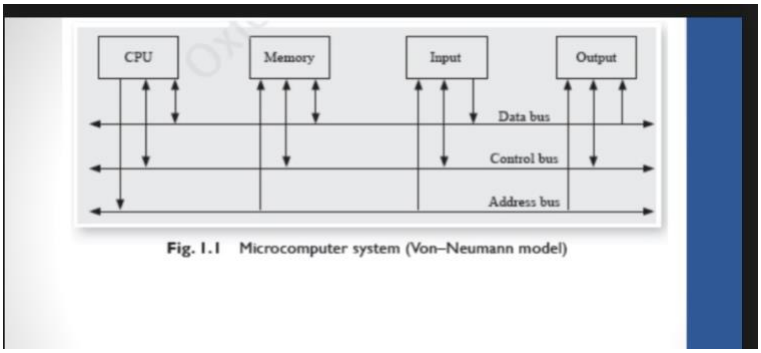
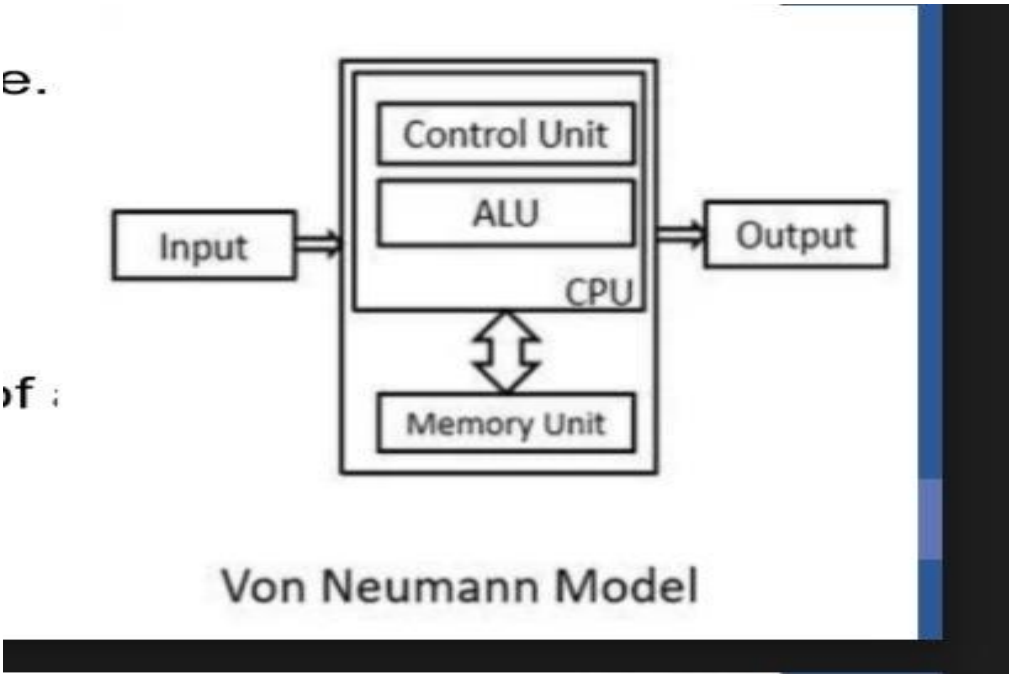
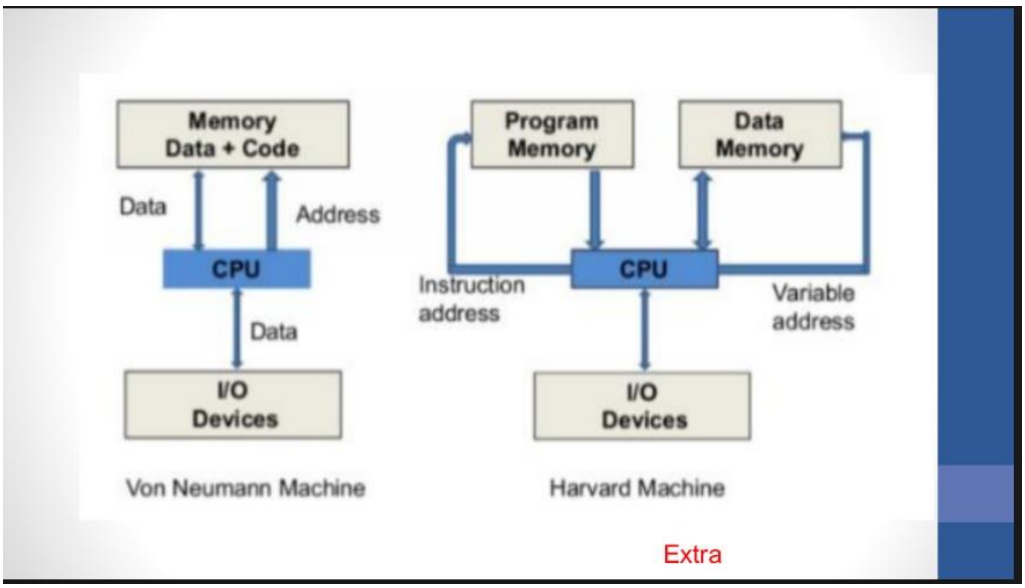
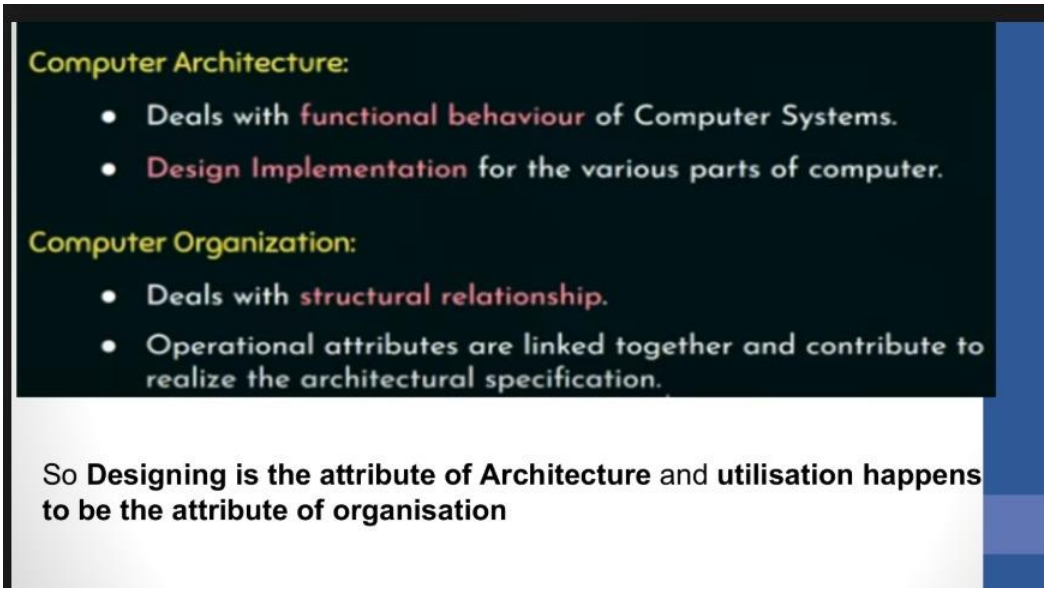


Fig. 1.1 Microcomputer system (Von–Neumann model)





Q3>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.

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.

Q4>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

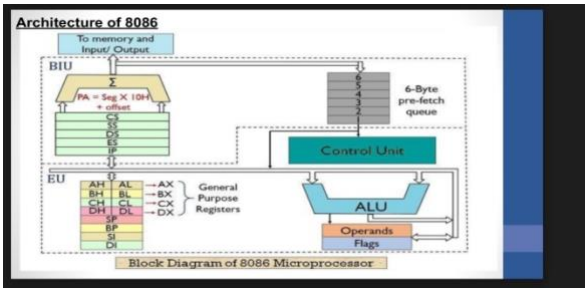
Performance or Response Time

Execution Time

Throughput.

Q5--similar to 2 and 45.-- Explain the Von Neumann Architecture in details, and list down the performance

Measure of Computer Architecture.



6. Draw and Explain 8086 Microprocessor Architecture in detail.

As shown in the below figure, 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 memory 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

4 Segment registers,

the Instruction Pointer,

a pre-fetch queue,

and an Address Generation Circuit.

EXECUTION UNIT :

The main parts are:

Control Circuitry

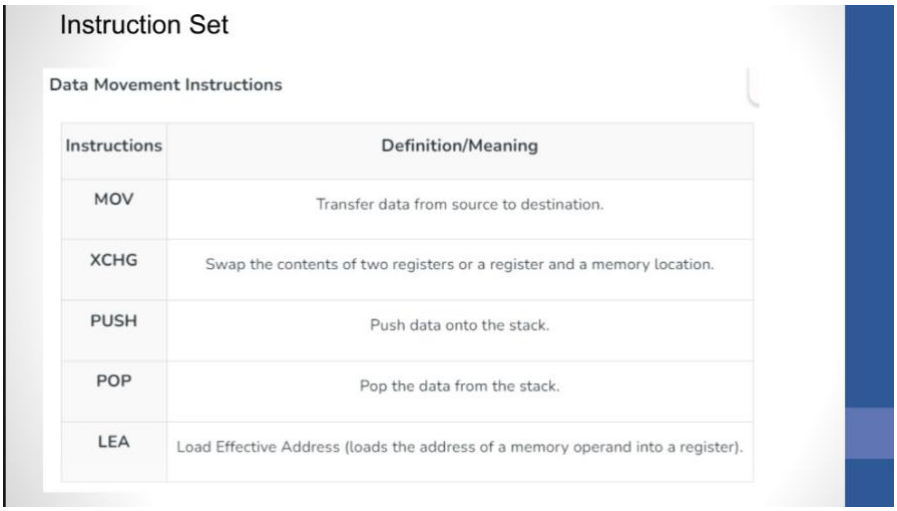
Instruction decoder

ALU

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.

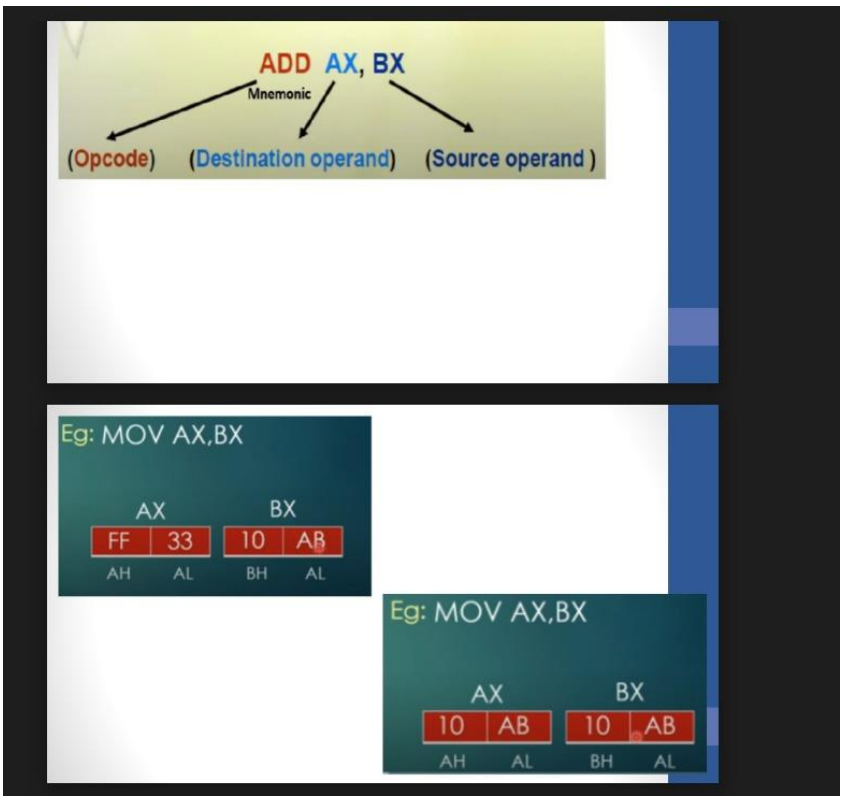


Q7>7. Explain Instruction Set for 8086 Microprocessor.

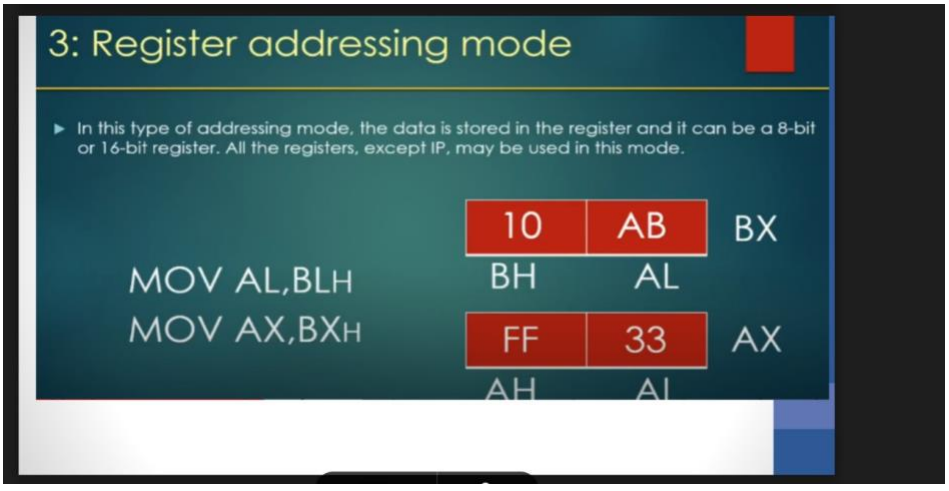
8086 Instruction Set

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.



Q8>8. Describe any 5 Addressing Mode.

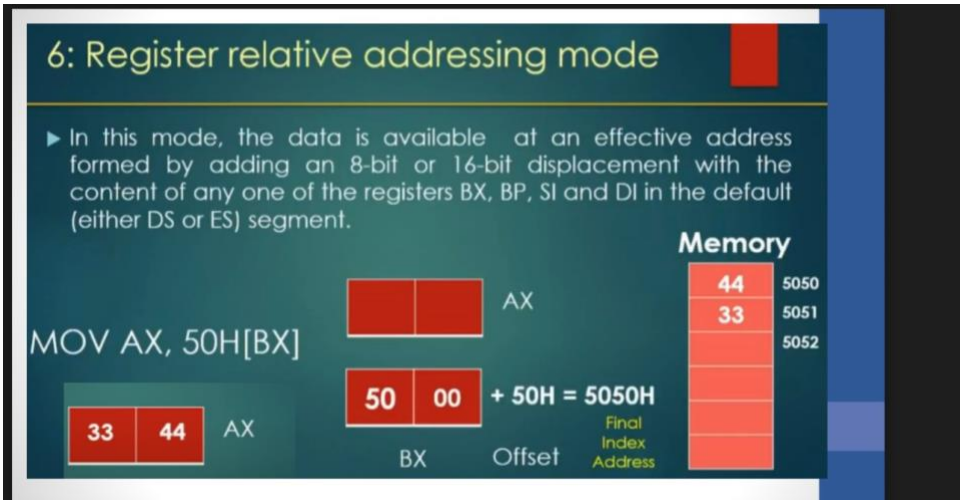


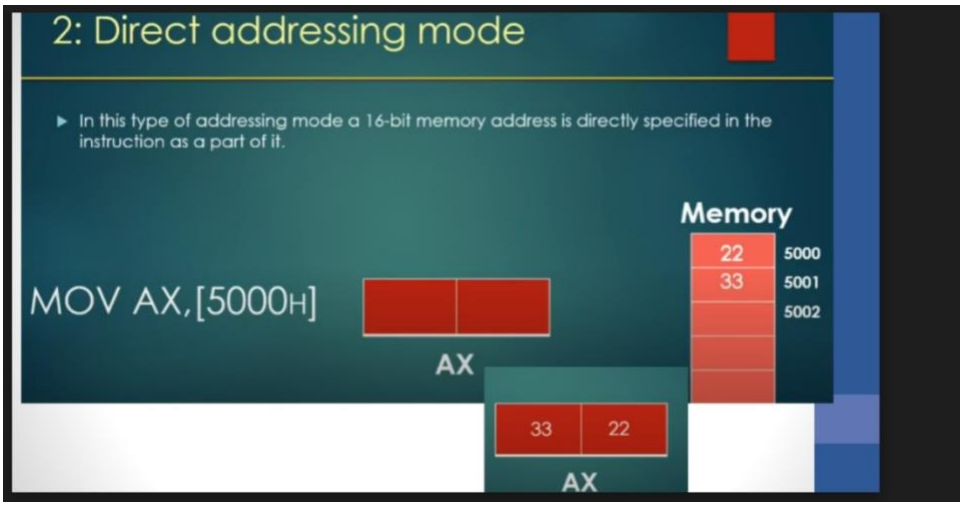
Register addressing modes:

- Here data is stored in a registeres and referred using the particular register.
- In this type of addressing mode both the operands are registers.

- **Example:**
- MOV AL, CL

AH	9B H	AL	FF H
BH	F0 H	BL	F0 H
CH	3E H	CL	25 H
DH	AB H	DL	A5 H

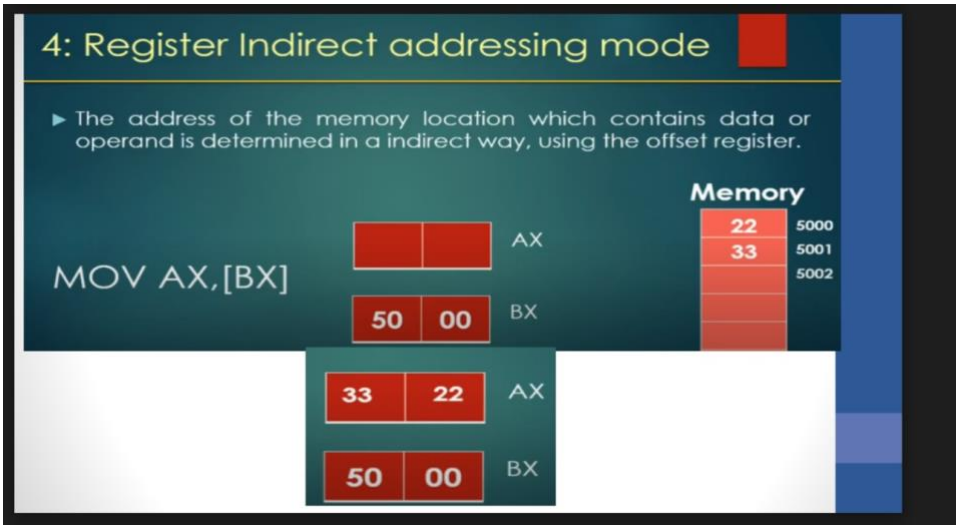




4: Register Indirect addressing mode

- The address of the memory location which contains data or operand is determined in a indirect way, using the offset register.

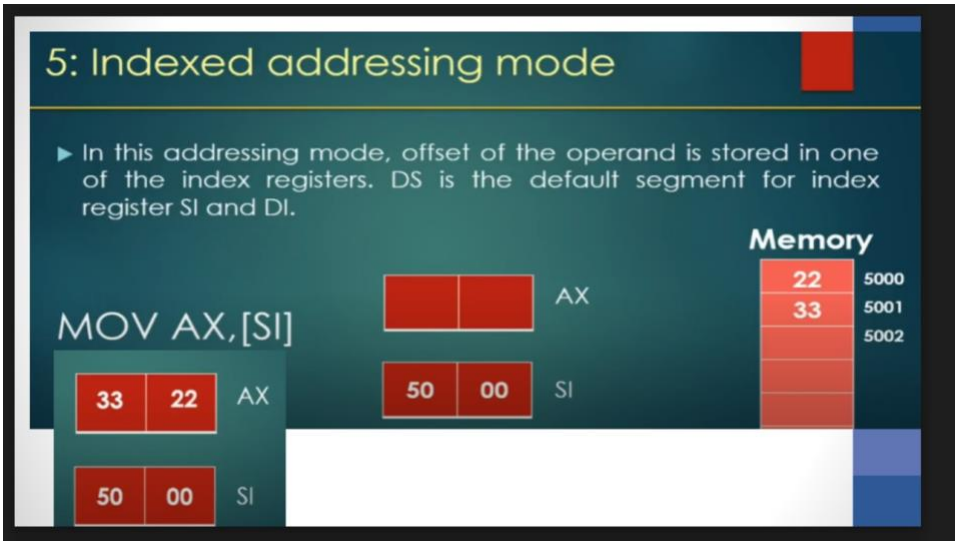
MOV AX,[BX]



5: Indexed addressing mode

- In this addressing mode, offset of the operand is stored in one of the index registers. DS is the default segment for index register SI and DI.

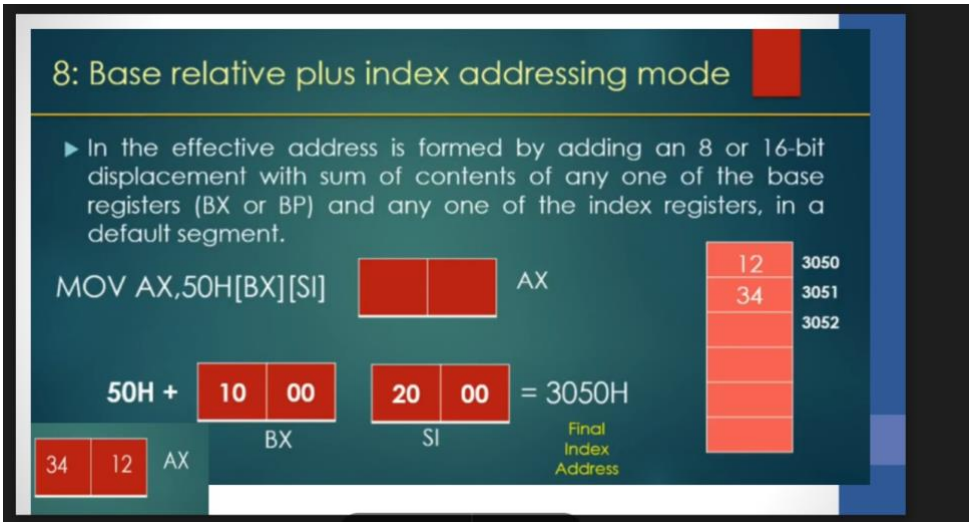
MOV AX,[SI]

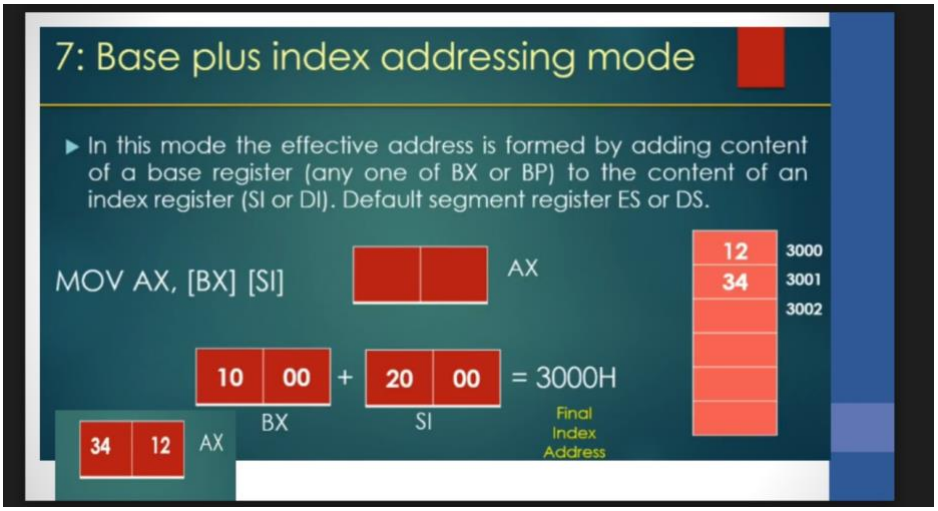


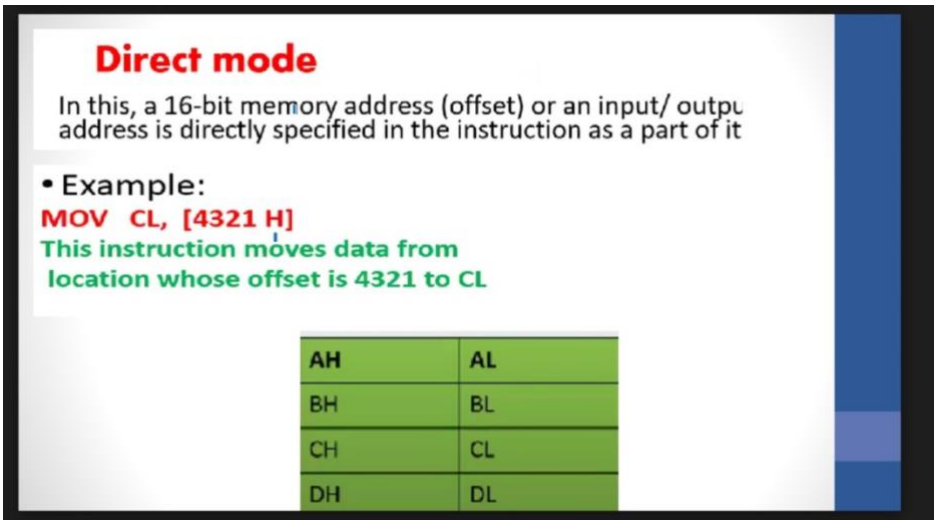
8: Base relative plus index addressing mode

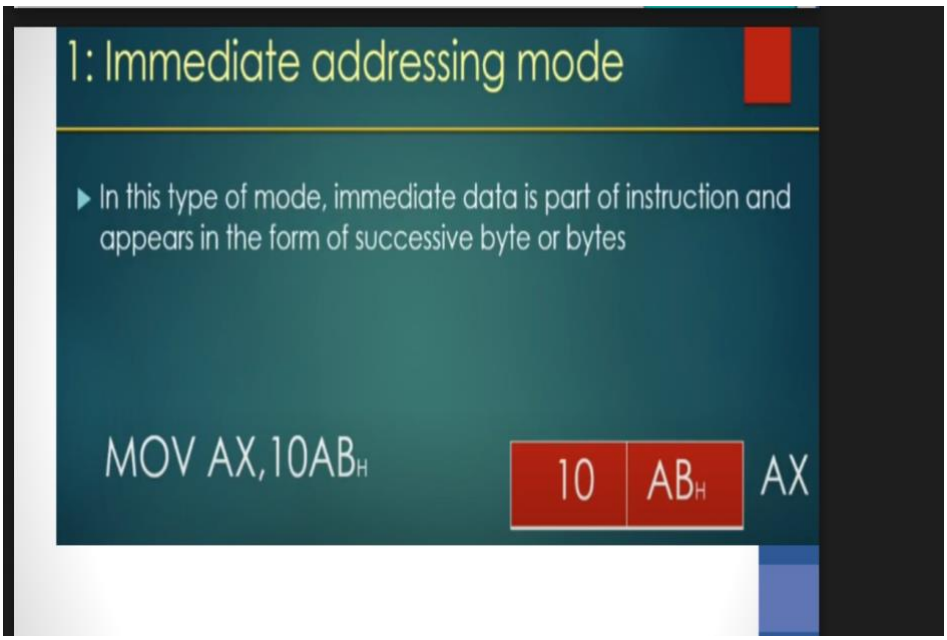
- In the effective address is formed by adding an 8 or 16-bit displacement with sum of contents of any one of the base registers (BX or BP) and any one of the index registers, in a default segment.

MOV AX,50H[BX][SI]









Q9>9. Explain features of 8086microprocessor.

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

A 40 pin dual in line package.Have 29000Transistors.

8086 has a 20 bit address bus can access up to 2^20= 1 MB

memory locations. It segments it into 16, 64kB segments.

8086 has a 16bit data bus. It can read or write data to a

memory/port either 16bits or 8 bit at a time.

It can support up to 64K I/O ports.

It provides 14, 16 -bit registers.

Frequency range of 8086 is 6-10 MHz .(The Intel 8086

microprocessor operates at a frequency of 5 MHz.

10.Explain Bus Interface unit.

Bus Interface Unit (BIU) of 8086

The Bus Interface Unit (BIU) is responsible for interfacing the 8086 microprocessor with memory and I/O devices using the system bus.

It takes care of all external bus operations, so the Execution Unit (EU) can focus only on instruction decoding and execution.

Functions of BIU:

Generates 20-bit physical address for accessing memory locations (max 1 MB memory).

Fetches instructions from memory in advance and places them into a 6-byte instruction queue (supports pipelining, i.e., fetching next instructions while the current one executes).

Transfers data between CPU, memory, and I/O devices.

Provides a 16-bit bidirectional data bus and a 20-bit address bus.

Components of BIU:

4 Segment Registers (CS, DS, ES, SS).

Instruction Pointer (IP) → holds offset of the next instruction.

Instruction Queue → 6-byte FIFO queue for prefetching instructions.

Address Generation Circuit → calculates physical addresses.

Segment Registers:

8086 uses segmented memory (1 MB divided into 16 segments of 64 KB). At a time, 4 segments can be used:

Code Segment Register (CS):

Holds base address of program instructions.

Used with IP to fetch instructions.

Data Segment Register (DS):

Points to the data segment where program data is stored.

General registers (AX, BX, CX, DX, SI, DI) access data from DS by default.

Extra Segment Register (ES):

Provides an additional data segment.

Commonly used in string and memory operations.

Stack Segment Register (SS):

Points to stack segment, used for storing temporary data, return addresses, etc.

Works with SP (Stack Pointer) and BP (Base Pointer) registers.

The address of the next instruction is calculated by using the formula

PA= CS x 10H + IP.

Assume Cs=1000h

IP=2345h

PA= 1000 X 10h + 2345= 12345H

It has multiplexed address and data bus AD0- AD15 and

A16 – A19.

It requires single phase clock with 33% duty cycle to

provide internal timing.

It can prefetch upto 6 instruction bytes from memory

and queues them in order to speedup instruction

execution.

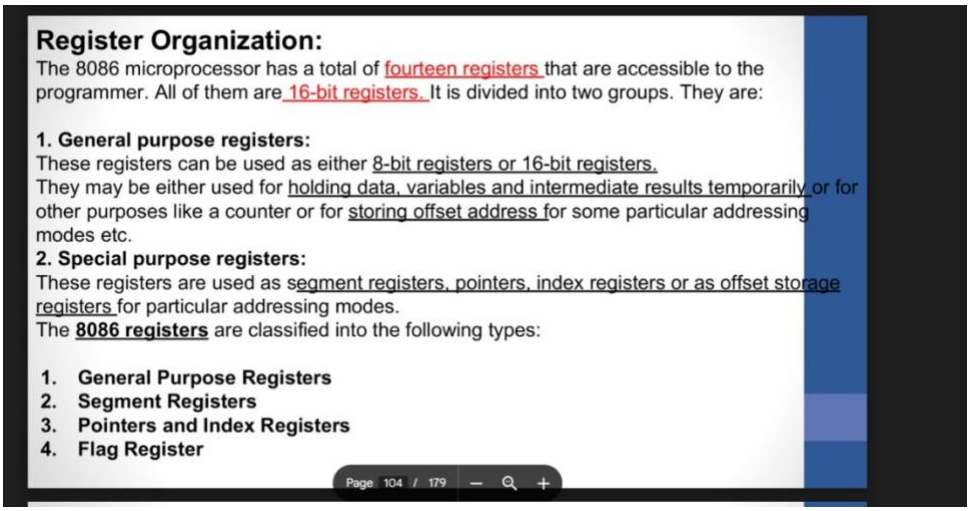
It requires +5V power supply.

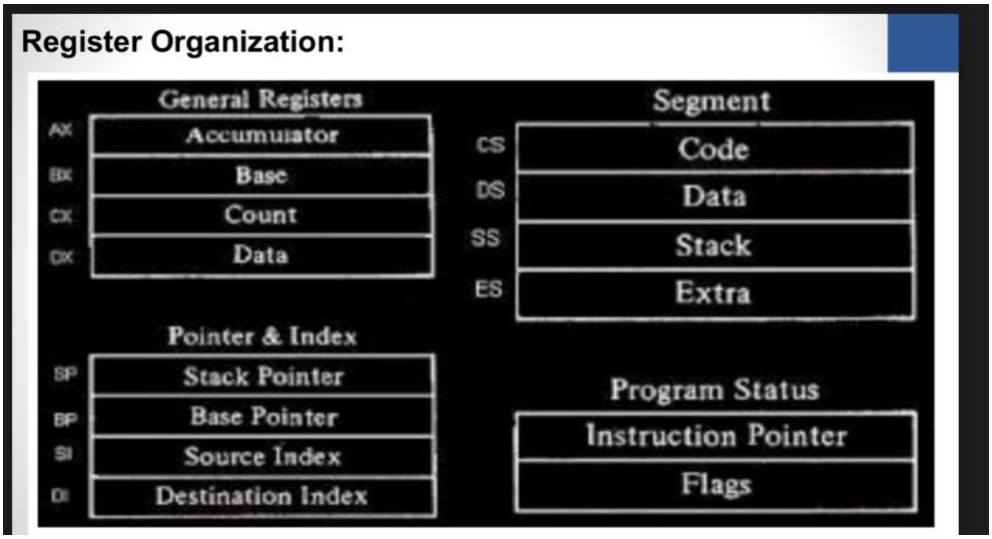
8086 is designed to operate in two modes, Minimum

mode and Maximum mode.

Address ranges from 00000H to FFFFFH

11.Explain Register organization in 8086 Microprocessor.





12.Explain 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.

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.

StepS

EU extracts instructions from top of queue in BIU

Decode the instructions

Generates operands if necessary

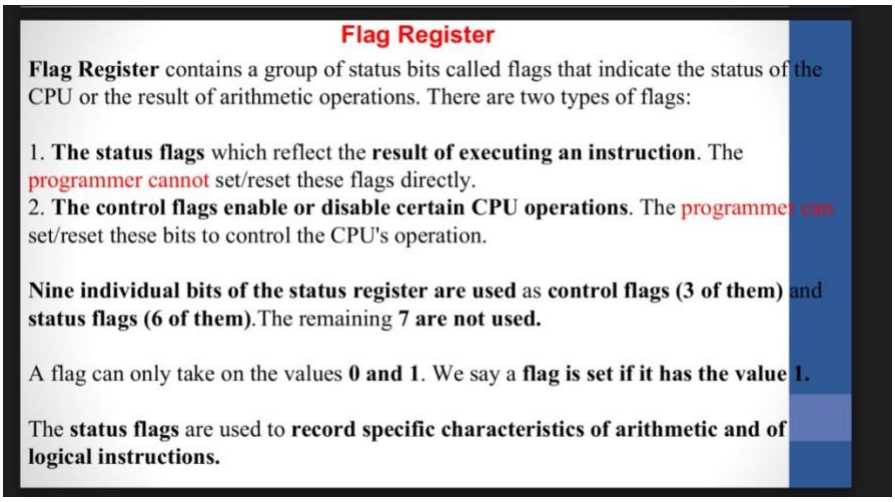
Passes operands to BIU & requests it to perform read or write

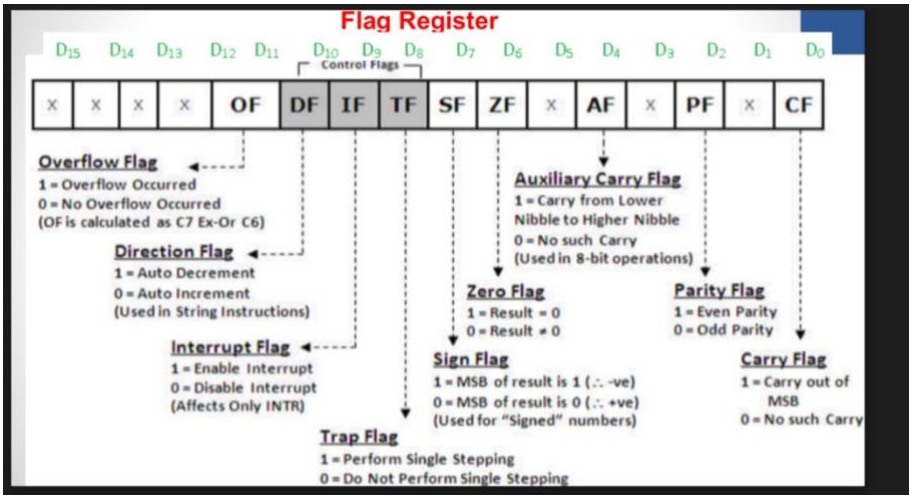
bus cycles to memory or I/O

Perform the operation specified by the instruction on

Operands

13.Explain Flag Register of 8086 Microprocessor.





GPT-

Flag Register of 8086 Microprocessor

The flag register in the 8086 microprocessor is a 16-bit register that indicates the status of the CPU or the result of arithmetic/logic operations. It contains status flags (reflect result of instructions) and control flags (control CPU operations).

Out of 16 bits, 9 bits are used as flags (6 status + 3 control).

A flag can have value 0 (reset) or 1 (set).

Types of Flags

Status Flags → Show the result of arithmetic/logic operations.

Carry Flag (CF): Set if there is a carry out of MSB (for addition) or borrow (for subtraction).

Parity Flag (PF): Set if result has even number of 1s, otherwise cleared.

Auxiliary Carry Flag (AF): Set if there is a carry from D3 to D4 (used in BCD operations).

Zero Flag (ZF): Set if result = 0, otherwise cleared.

Sign Flag (SF): Set if MSB of result is 1 (negative result), cleared if MSB = 0.

Overflow Flag (OF): Set if signed operation result is out of range.

Control Flags → Control processor operations, can be set/reset by programmer.

Trap Flag (TF): Enables single-step execution (debugging).

Interrupt Flag (IF): Enables/disables external interrupts.

Direction Flag (DF): Controls string operations (0 = auto-increment, 1 = auto-decrement).

Structure

16-bit register: D0–D15.

Only 9 bits are used (CF, PF, AF, ZF, SF, OF, TF, IF, DF).

Example of Flag Operation

If we add 7FH + FEH,

Carry FlAg (CF) = 1

Zero Flag (ZF) = 0 (result ≠ 0)

Sign Flag (SF) = 0 (result positive)

Auxiliary Carry (AF) = 1

Parity Flag (PF) = 1 (even number of 1's)

Overflow Flag (OF) = 0
