Unit 1:

1.1)Introduction to basic concept of computer:

A **computer** is an electronic device that accepts data from the user, processes it, produces results, displays them to the users, and stores the results for future usage.

Data is a collection of unorganized facts & figures and does not provide any further information regarding patterns, context, etc. Hence data means "unstructured facts and figures".

Information is a structured data i.e. organized meaningful and processed data. To process the data and convert into information, a computer is used.

Functions of Computers:

Receiving Input

Data is fed into computer through various input devices like keyboard, mouse, digital pens, etc. Input can also be fed through devices like CD-ROM, pen drive, scanner, etc.

Processing the information

Operations on the input data are carried out based on the instructions provided in the programs.

Storing the information

After processing, the information gets stored in the primary or secondary storage area.

Producing output

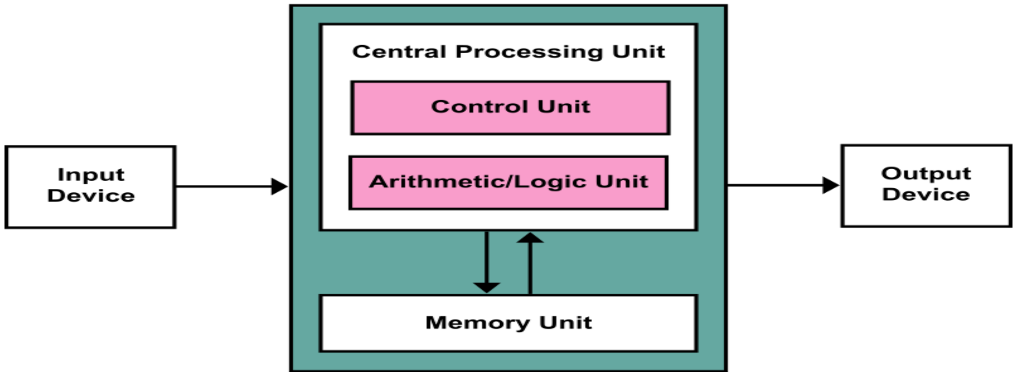
The processed information and other details are communicated to the outside world through output devices like monitor, printer, etc.

1.2)Computer organization(Block diagram of Von Neumann and Hazard Architecture):

Computer Organization is realization of what is specified by the computer architecture .It deals with how operational attributes are linked together to meet the requirements specified by computer architecture. Some organizational attributes are hardware details, control signals, peripherals.

Harvard and Von Neumann Architecture with diagram explanation

Harvard architecture is developed to overcome the bottleneck of Von-Neumann Architecture. In Harvard architecture, it contains separate buses and storages for instructions and data.
The major advantages of having separate busses for data and instructions is that the **central procession unit** can access instructions and write and read data at the same time.
It stores machine instructions and data separately that are connected to totally different busses so that both tasks can be executed simultaneously and execution speed will be increased.



Let's talk about the structure of Harvard architecture.

1. Buses: It is used as a signal pathway for instructions and data. in this architecture, there are separate buses for both data and instruction. There are different types of buses they are as follows: **Data Bus**, **Data Address Bus**, **Instruction bus**, **Instruction Address bus**.

2. Operational Registers: In Harvard architecture different types of registers used which are used for storing addresses of different types of instructions. some registers are **memory data register**, **memory address registers.**

3. **Program Counter**: program counter has the location of the next instruction to be executed. And program counter passes the next address to the memory address register.

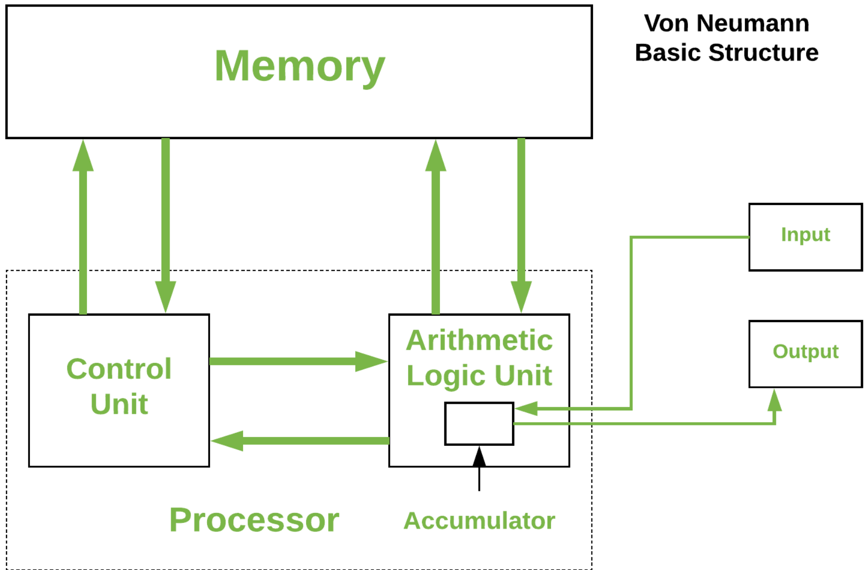
4. **ALU:** Arithmetic Logic Unit is responsible for all the calculations required such as Multiplication, Addition, Subtraction, comparison, Logical operators and other arithmetic operators.

5. **Control Unit (CU):** CU is also a part of the central processing unit (CPU). CU operates all the processor control signals, Input/output devices, movement of data and instructions within the systems.

6. **Input/Output Stream:** I/P devices are used to read data into the main memory with the help of central processing unit (CUP) input instructions. It gives the information from the system as output are displayed through output devices. In simple word, computer systems give the result of processing (computation) with the help of O/P devices.

Von Neumann Architecture

This architecture is published by Jhon von Neumann in 1945. This architecture contains major components like control unit (CU), memory Unit, ALU, inputs/outputs, and registers. This concept is based on the stored-program computer concept. Where program data an instruction data are stored in the same memory. Von Neumann design still used in most computers produces to date.



1. CPU: CPU is an electronic circuit unit which is responsible for the execution of the instruction of a computer program. The central processing unit is the primary component of a computer system that processes instructions. It runs the OS and applications, constantly receiving input from the user or active software programs.

2. Registers: It is a kind of memory wont to quickly accept, store, and transfer data and directions that are getting used immediately by the central processing unit.

MAR	Memory Address Register	Holds the memory location of data that needs to be accessed
MDR	Memory Data Register	Holds data that is being transferred to or from memory
AC	Accumulator	Where intermediate arithmetic and logic results are stored
PC	Program Counter	Contains the address of the next instruction to be executed
CIR	Current Instruction Register	Contains the current instruction during processing

3. ALU: Arithmetic Logic Unit is responsible for all the calculations required such as Multiplication, Addition, Subtraction, comparison, Logical operators and other arithmetic operators.

4. **Control Unit (CU):** CU is also a part of the central processing unit (CPU). CU operates all the processor control signals, Input/output devices, movement of data and instructions within the systems.

5. Buses: It is used as a signal pathway for instructions and data. in this architecture, It is used to connect all the components of the computer's CPU and Memory unit.

6. **Memory Unit**: The memory unit consists of Random-access memory (RAM), sometimes mentioned as primary or main memory. Unlike a tough drive (secondary memory), this memory is fast and also directly accessible by the CPU. RAM is split into partitions. Each partition of it consists of an address and its contents. The address will uniquely identify every location within the memory. Loading data from permanent memory (hard drive), into the faster and directly accessible temporary memory (RAM), allows the CPU to work much quicker.

1.3)Introduction to basic hardware components:

Hardware is the physical equipment. It includes the case, keyboard, monitor, cables, storage drives, motherboard, and power supply. Software includes the operating system and programs. The operating system, or OS, manages computer operations such as identifying, accessing, and processing information.

1. Central processing unit (CPU)

The CPU controls the computer processes and communicates with the other components of a personal computer. A computer's CPU may be one of the most complicated parts of the computer because of its intricacies. If a computer is experiencing CPU issues, a computer technician may start by checking the fan and cleaning out any dust or debris inside the machine. Another step a computer technician may complete is verifying that power supply cables work. A CPU will not function properly unless it receives a signal from the power supply.

2. Motherboard

The motherboard provides the structure for all other components and connects them, while also providing a way to distribute power, deliver information and connect to devices such as a printer or mouse. It controls how data transfers and what type of monitor or screen device to use, for example. It houses the CPU, memory and secondary storage devices such as hard drives.

The first thing that a computer technician may do to troubleshoot any motherboard issues is to take the PC apart and inspect all the connections for corrosion. They may also check the power supply and make sure the computer is receiving electricity.

3. Random Access Memory (RAM)

RAM is where data lives temporarily while it's being actively used by programs, such as when a user launches a computer application. A technician may know how to identify the type of RAM in a computer, how to replace it if it's defective and how to diagnose problems with copying data from one location in memory to another.

A technician should have knowledge about the different RAMs that are available and they may also know what types of errors might affect a computer's RAM operation. A technician may backup all computer files before fixing RAM to avoid losing critical programs and documents.

4. Video graphics array port

A video graphics array (VGA) port is a video input that is primarily used on PC monitors. Troubleshooting a VGA port could include verifying there isn't a loose connection, faulty cable or a broken monitor. Another task a computer technician may do is use compressed air to spray inside the VGA port to ensure it's free of dust.

5. Power supply

A power supply provides electricity to all components of a computer system. Typically, it's a power cord connected from the back of a PC tower into an electricity socket. A technician may troubleshoot the power supply by turning off the computer, unplugging and detaching the power supply cord or trying a new cord or outlet.

6. Cooling fan

Cooling fans are a computer's system to decrease overheating. Many computers have more than one cooling fan to help users who run their computer heavily, such as video streaming or gaming. A computer technician may need to fix a computer's cooling fan if a user notices their computer overheating. They may check for any damage to the blades and make sure that they are free from debris. Replacing computer fans can be a common troubleshooting solution for a technician.

7. Hard drive

Hard drives are data storage devices used to store files, programs or other information on a computer system. They use magnetically coated discs called hard disks that store digital representations of information. If a hard drive fails, a computer technician may suspect a corrupt hard drive. They may use data recovery software to repair the computer or may replace the hard drive.

8. Monitor

A computer monitor is an electronic device that displays what's what programs are running on your computer so a user can see. Some computer technicians may wear anti-static gloves when handling computer monitors to avoid static electricity. They may also troubleshoot monitor issues by disconnecting it from the computer and trying a new power cord.

9. Printer

This is a machine that produces copies of text or images on paper using ink. Popular printers include laser or inkjet and computer technicians may be skilled in troubleshooting issues across multiple brands and varieties. Computer technicians may service printers including verifying power is running to the machine. They may also check to see if the paper tray is full and ready for printing. Technicians might replace or refill any ink cartridges and toner.

If the physical printer components are operating efficiently but the machine still isn't working, a technician will then troubleshoot software malfunctions that may occur in the communication between the computer and the printer. This requires more expertise to fix since it involves how the systems interact with each other rather than physical component failures on the printer itself.

10. Scanner

A scanner is a device that digitally copies an image or and makes it available as a file for access on a computer. If a scanner malfunctions, a computer technician may remove the cover and carefully check if it has any damage. If there are no visible issues, then they may check the power connection cable. Similar to a printer, there is computer software that connects the scanner to the computer, so a technician will also verify that the software is communicating to the scanner.

11. Computer mouse

A computer mouse can have a wire or is wireless and is an input device used to control a cursor on the computer monitor. A wired computer mouse has a cord that connects to a USB port on a computer while a wireless mouse has no physical connection with a computer system. Instead, a wireless mouse communicates using an adapter that is usually plugged into an available USB port and runs off its own batteries using Bluetooth technology.

1.4)Power supply,Casing,motherboard,CPU,Chipset,real-time clock,BIOS:

a)Power supply :A power supply is an electronic circuit designed to provide various ac and dc voltages for equipment operation. A power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a a different form of energy – such as solar, mechanical, or chemical - into electrical energy.

Parts found inside a power supply

Below is a list of parts inside a power supply.

- A rectifier that converts AC (alternating current) into DC.
- A filter that smooths out the DC (direct current) coming from a rectifier.
- A transformer that controls the incoming voltage by stepping it up or down.
- A voltage regulator that controls the DC output, allowing the correct amount of power, volts or watts, to be supplied to the computer hardware.

The order that these internal power supply components function is as follows.

1. Transformer
2. Rectifier
3. Filter
4. Voltage Regulator

b)Casing:A **computer case**, also known as a **computer chassis**, is the enclosure that contains most of the components of a personal computer . Cases are usually constructed from steel,aluminium and plastic. A **computer case** safeguards all of the important—and expensive—parts of a PC or other computing device. Internal components include things like the motherboard, chassis, drives, CPU and more for small desktop units, all the way through whole networks and even up to advanced blade servers.

The desktop computer case helps protect the components from electrical interference, physical damage, and intrusive foreign objects. The design of the case is important, too, as different shapes can affect, for better or worse, the airflow through the unit—which is integral to keeping it cool and operational. The cases also keep the internal parts separate from each other while creating space for expansion slots, warning lights, graphics cards, RAM, and other server applications.

c)Motherboard:A motherboard is the main printed circuit board (PCB) in a computer. The motherboard is a computer's central communications backbone connectivity point, through which all components and external peripherals connect.

Features of Motherboard

A motherboard comes with following features −

- Motherboard varies greatly in supporting various types of components.
- Motherboard supports a single type of CPU and few types of memories.
- Video cards, hard disks, sound cards have to be compatible with the motherboard to function properly.
- Motherboards, cases, and power supplies must be compatible to work properly together.

d)**CPU**:The computer's central processing unit (CPU) is the portion of a computer that retrieves and executes instructions. The CPU is essentially the brain of a CAD system. It consists of an arithmetic and logic unit (ALU), a control unit, and various registers.

Central Processing Unit (CPU) consists of the following features −

- CPU is considered as the brain of the computer.
- CPU performs all types of data processing operations.
- It stores data, intermediate results, and instructions (program).
- It controls the operation of all parts of the computer.

e)Chipset:A chipset is a group of interdependent motherboard chips or integrated circuits that control the flow of data and instructions between the central processing unit (CPU) or microprocessor and external devices. A chipset controls external buses, memory cache and some peripherals. A CPU is unable to function without impeccable chipset timing.

A chipset includes the circuit board layout/functionality and circuit mechanisms. Varieties include microprocessors and modem card chipsets. In addition, a CPU has several different chipsets that vary according to architecture.

A chipset has two sections – southbridge and northbridge – with specific sets of functions that communicate between the CPU and external devices.

The southbridge, which is not directly connected to the CPU, is also known as the input/output controller hub. Southbridge handles the motherboard's slower connections, including input/output (I/O) devices and computer peripherals like expansion slots and hard disk drives.

The northbridge connects the southbridge to the CPU and is commonly known as the memory controller hub. The northbridge handles a computer's faster interaction requirements and controls communication between the CPU, RAM, ROM, the basic input/output system (BIOS), the accelerated graphics port (AGP) and the southbridge chip. The northbridge links I/O signals directly to the CPU. The CPU uses the northbridge frequency as a baseline for determining its operating frequency.

f)Real-time clock:A real-time clock (RTC) is a computer clock, usually in the form of an integrated circuit that is solely built for keeping time. Naturally, it counts hours, minutes, seconds, months, days and even years. RTCs can be found running in personal computers, embedded systems and servers, and are present in any electronic device that may require accurate time keeping. Being able to still function even when the computer is powered down through a battery or independently from the system's main power is fundamental.

BIOS: BIOS (basic input/output system) is **the program a computer's microprocessor uses to start the computer system after it is powered on**. It also manages data flow between the computer's operating system (OS) and attached devices, such as the hard disk, video adapter, keyboard, mouse and printer.

Functions of BIOS

The BIOS has different instructions that are required to load the hardware, and it is responsible for loading the operating system. The major functions of a Basic input/output system (BIOS) are discussed below:

- **BIOS Power on Self-Test (POST):** It ensures the proper functioning of the computer hardware as it is a built-in diagnostic program that. In the system, it verifies the computer meets the necessary parts and functionality. POST does this function efficiently. It ensures that the computer is loading tasks successfully, such as the use of memory, a keyboard, and other parts when it starts up. If the POST test fails at the time of test, the computer provides a combination of beeps to display the error type, and the system continues to boot when the POST test is passed completely.
- **Bootstrap Loader:** The BIOS recognizes and locates the operating system when the POST running successfully. The program bootstrap loader is contained by BIOS, which searches and starts the OS boot program. When BIOS detects one, it transfers access to Operating System that is known as booting.
- **BIOS drivers:** BIOS drivers are stored in the non-volatile memory, whose primary function is to supply basic computer hardware information.

- **BIOS Setup Utility Program:** It is a configuration software, also known as CMOS setup, that allows users to configure hardware settings as well as device settings, time and date, computer password. The NVRAM, non-volatile memory, is used to store settings of memory, disk types, and information about the computer system; this information is not stored in the BIOS chip

1.5)Parallel ports,Serial ports,interfacing(IDE,SATA,PATA,ATAPC):

a)Parallel ports:

A parallel port is an interface allowing a personal computer (PC) to transmit or receive data down multiple bundled cables to a peripheral device such as a printer. The most common parallel port is a printer port known as the Centronics port. A parallel port has multiple connectors and in theory allows data to be sent simultaneously down several cables at once. Later versions allow bi-directional communications. This technology is still used today for low-data-rate communications such as dot-matrix printing. The **parallel port** is found on the back of IBM compatible computers and is a 25-pin (type **DB-25**) computer interface commonly used to connect printers to the computer.

b)Serial ports:

A serial port is an interface that allows a PC to transmit or receive data one bit at a time. It is one of the oldest types of interfaces and at one time was commonly used to connect printers and external modems to a PC. Modern serial ports are used in scientific instruments, shop till systems such as cash registers and applications like industrial machinery systems.
Compared to a parallel port, the data transfer rate of a serial port is slower.
Serial ports are usually identified on IBM compatible computers as COM (communications) ports. For example, a mouse might connect to COM1 and a modem to COM2. The picture shows the **DB9** serial connector on a cable.

Sr. No.	Key	Serial Ports	Parallel Ports
1	Purpose	Serial Port is used for serial data transmission.	Parallel Port is used for parallel data transmission.
2	Transmission Speed	Transmission speed of a serial port is slow as compared to a parallel port.	Transmission speed of a parallel port is quiet high as compared to a serial port.
3	Redundancy	Bottom-Up model is better suited as it ensures minimum data redundancy and focus is on re-usability.	Top-down model has high ratio of redundancy as the size of project increases.
4	No. Of Wires	Wire connections to serial port are quiet less as compared to parallel port.	No. of wires that are connected to parallel port are quiet high as compared to serial port.
5	Capability	A serial port is able to transmit a single stream of data at a time.	A parallel port is able to transmit multiple data streams at a time.

Sr. No.	Key	Serial Ports	Parallel Ports
6	Data Sending Mechanism	A serial port sends data bit by bit after sending a bit at a time.	A parallel port sends data by sending multiple bits in parallel fashion.
7	Port Type	A serial port uses Male ports.	A parallel port uses Female ports.
8	Applications	Modems, security cameras, device controllers use serial ports.	Printers, Hard Drives, CD drives use parallel ports.

c)Interfacing(IDE,SATA,PATA,ATAPI):

Interface may refer to a hardware interface, a software interface, or sometimes it can refer to the interaction between the user and the computer via an input device such as the keyboard. An **interface** is a shared boundary across which two or more separate components of a computer system exchange information. The exchange can be between software, computer hardware, peripheral devices, humans, and combinations of these.

a)IDE:IDE (Integrated Drive Electronics) is **an electronic interface standard that defines the connection between a bus on a computer's motherboard and the computer's disk storage devices**.Integrated Drive Electronics (IDE) is a standard interface for connecting a motherboard to storage devices such as hard drives and CD-ROM/DVD drives. The original IDE had a 16-bit interface that connected two devices to a single-ribbon cable. This cost-effective IDE device carried its own circuitry and included an integrated disk drive controller. Prior to IDE, controllers were separate external devices.

b)SATA:Stands for "Serial Advanced Technology Attachment," or "Serial ATA." It is an interface used to connect ATA hard drives to a computer's motherboard. SATA transfer rates start at 150MBps, which is significantly faster than even the fastest 100MBps ATA/100 drives. Serial ATA (Serial Advanced Technology Attachment or SATA) is a command and transport protocol that defines how data is transferred between a computer's motherboard and mass storage devices, such as hard disk drives (HDDs), optical drives and solid-state drives (SSDs). As its name implies, SATA is based on serial signaling technology, where data is transferred as a sequence of individual bits.

c)PATA:**PATA** stands for **Parallel Advanced Technology Attachment**.A motherboard has many components in it. Each component has its own functionality. There are cables used for different purposes in a motherboard and PATA is one of them.Parallel Advanced Technology Attachment(PATA) also called Advanced Technology(AT) Attachment is a bus interface used for connecting secondary storage devices like hard disks, optical drives to the computer system. The functionality of PATA is to transfer the data.

d)ATAPI: ATAPI (AT Attachment Packet Interface) is an interface between your computer and attached CD-ROM drives and tape backup drives. Most of today's PC computers use the standard Integrated Drive Electronics (IDE) interface to address hard disk drives. ATAPI provides the additional commands needed for controlling a CD-ROM player or tape backup so that your computer can use the IDE interface and controllers to control these relatively newer device types.

Unit 2: Data Representation

1) Introduction to data representation:

The data can be represented in the following ways:

Data

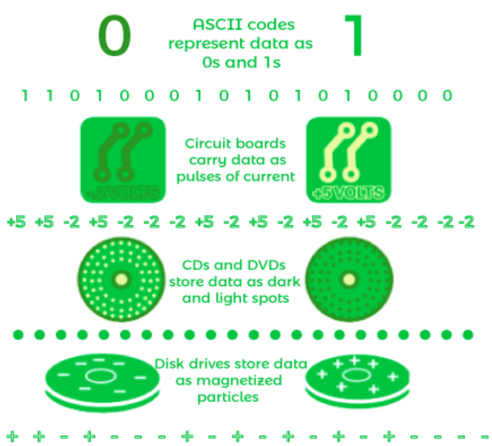
Data can be anything like a number, a name, notes in a musical composition, or the color in a photograph. Data representation can be referred to as the form in which we stored the data, processed it and transmitted it. In order to store the data in digital format, we can use any device like computers, smartphones, and iPads. Electronic circuitry is used to handle the stored data.

Digitization

Digitization is a type of process in which we convert information like photos, music, number, text into digital data. Electronic devices are used to manipulate these types of data. The digital revolution has evolved with the help of 4 phases, starting with the big, expensive standalone computers and progressing to today's digital world. All around the world, small and inexpensive devices are spreading everywhere.

Binary Digits

The **binary digits** or bits are used to show the digital data, which is represented by 0 and 1. The binary digits can be called the smallest unit of information in a computer. The main use of binary digit is that it can store the information or data in the form of 0s and 1s. It contains a value that can be on/off or true/false. On or true will be represented by the 1, and off or false will be represented by the 0. The digital file is a simple file, which is used to collect data contained by the storage medium like the flash drive, CD, hard disk, or DVD.



Representing Numbers

The number can be represented in the following way:

Numeric Data

Numeric data is used to contain numbers, which helps us to perform arithmetic operations. The digital devices use a binary number system so that they can represent numeric data. The binary number system can only be represented by two digits 0 and 1. There can't be any other digits like 2 in the system. If we want to represent number 2 in binary, then we will write it as 10.

Representing Numbers

DECIMAL (BASE 10)	BINARY (BASE 2)
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
1000	1111101000

Representing Text

The text can be represented in the following ways:

Character Data

Character data can be formed with the help of symbols, letters, and numerals, but they can?t be used in calculations. Using the character data, we can form our address, hair colour, name, etc. Character data normally takes the data in the form of text. With the help of the text, we can describe many things like our father name, mother name, etc.

Digital Devices

Several types of codes are employed by the **digital devices** to represent character data, including Unicode, ASCII, and other types of variants. The full form of ASCII is American Standard Code for Information Interchange. It is a type of character encoding standard, which is used for electronic communication. With the help of telecommunication equipment, computers and many other devices, ASCII code can represent the text. The ASCII code needs 7 bits for each character, where the unique character is represented by every single bit. For the uppercase letter A, the ASCII code is represented as 1000001.

Extended ASCII

Extended ASCII can be described as a superset of ASCII. The ASCII set uses 7 bits to represent every character, but the Extended ASCII uses 8 bits to represent each character. The extended ASCII contains 7 bits of ASCII characters and 1 bit for additional characters. Using the 7 bits, the ASCII code provides code for 128 unique symbols or characters, but Extended ASCII provides code for 256 unique symbols or characters. For the uppercase letter A, the Extended ASCII code is represented as 01000001.

2)State units of measurement(Bits,Bytes):

Memory unit is the amount of data that can be stored in the storage unit. This storage capacity is expressed in terms of Bytes.

The following table explains the main memory storage units −

S.No.	Unit & Description
1	**Bit (Binary Digit)** A binary digit is logical 0 and 1 representing a passive or an active state of a component in an electric circuit.
2	**Nibble** A group of 4 bits is called nibble.
3	**Byte**

	A group of 8 bits is called byte. A byte is the smallest unit, which can represent a data item or a character.
4	**Word** A computer word, like a byte, is a group of fixed number of bits processed as a unit, which varies from computer to computer but is fixed for each computer. The length of a computer word is called word-size or word length. It may be as small as 8 bits or may be as long as 96 bits. A computer stores the information in the form of computer words.

The following table lists some higher storage units −

S.No.	Unit & Description
1	**Kilobyte (KB)** 1 KB = 1024 Bytes
2	**Megabyte (MB)** 1 MB = 1024 KB
3	**GigaByte (GB)** 1 GB = 1024 MB
4	**TeraByte (TB)** 1 TB = 1024 GB
5	**PetaByte (PB)** 1 PB = 1024 TB

3)Describe Signed number representation:

Variables such as integers can be represent in two ways, i.e., signed and unsigned. Signed numbers use sign flag or can be distinguish between negative values and positive values. Whereas unsigned numbers stored only positive numbers but not negative numbers.

Number representation techniques like: Binary, Octal, Decimal and Hexadecimal number representation techniques can represent numbers in both signed and unsigned ways. Binary Number System is one the type of Number Representation techniques. It is most popular and used in digital systems. Binary system is used for representing binary quantities which can be represented by any device that has only two operating states or possible conditions.

4)Explain instruction set completeness:

A set of instructions is said to be complete if the computer includes a sufficient number of instructions in each of the following categories:

Arithmetic, logical and shift instructions

A set of instructions for moving information to and from memory and processor registers.

Instructions which controls the program together with instructions that check status conditions.

Input and Output instructions

Arithmetic, logic and shift instructions provide computational capabilities for processing the type of data the user may wish to employ.

A huge amount of binary information is stored in the memory unit, but all computations are done in processor registers. Therefore, one must possess the capability of moving information between these two units.Program control instructions such as branch instructions are used change the sequence in which the program is executed.Input and Output instructions act as an interface between the computer and the user. Programs and data must be transferred into memory, and the results of computations must be transferred back to the user.

5)Introduce BCD:

A binary-coded decimal (BCD) is a type of binary representation for decimal values where each digit is represented by a fixed number of binary bits, usually between four and eight.
The norm is four bits, which effectively represent decimal values 0 to 9. This writing format system is used because there is no limit to the size of a number. Four bits can simply be added as another decimal digit, versus real binary representation, which is limited to the usual powers of two, such as 16, 32 or 64 bits.

Explains Binary-Coded Decimal (BCD)

Binary-coded decimals are an easy way to represent decimal values, as each digit is represented by its own 4-bit binary sequence which only has 10 different combinations. By comparison, converting real binary representation to decimal requires arithmetic operations like multiplication and addition.

It is easier for conversion to decimal digits for display or printing, but the resulting circuit required to implement this system is more complex.For example, the binary coded decimal "1001 0101 0110," which has three groups of 4 bits, means there are three decimal digits. In order, from left to right, the resulting decimal value is 956.

The following are the 4-bit binary representation of decimal values:

0 = 0000
1 = 0001
2 = 0010
3 = 0011
4 = 0100
5 = 0101
6 = 0110
7 = 0111
8 = 1000
9 = 1001

Unit 3:Instruction format

1)Introduce instruction format:

Instruction includes a set of operation codes and operands that manage with the operation codes. Instruction format supports the design of bits in an instruction. It contains fields including opcode, operands, and addressing mode.

The instruction length is generally preserved in multiples of the character length, which is 8 bits. When the instruction length is permanent, several bits are assigned to opcode, operands, and addressing modes.

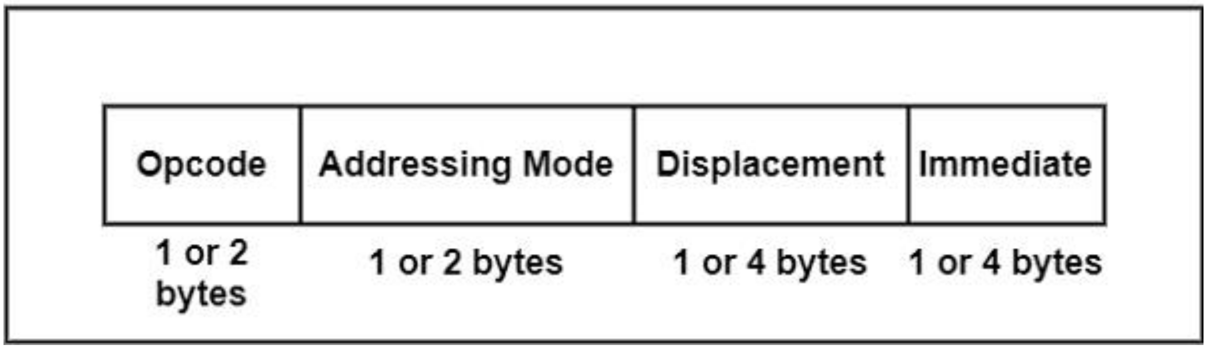
The function of allocating bits in the instruction can be interpreted by considering the following elements −

- Number of addressing modes
- Number of operands

- Number of CPU registers
- Number of register sets
- Number of address lines

The figure displayed the general IA-32 (Intel Architecture- 32 bits) instruction format. IA-32 is the instruction format that can Intel's most outstanding microprocessors. This instruction format includes four fields, such as opcode field, addressing mode field, displacement field, and immediate field.

IA-32 Instruction Format



The opcode field has 1 or 2 bytes. The addressing mode field also includes 1 or 2 bytes. In the addressing mode field, an instruction needs only one byte if it uses only one register to generate the effective address of an operand.

The field that directly follows the addressing mode field is the displacement field. If an effective address for a memory operand is computed using the displacement value, then it uses either one or four bytes to encode. If an operand is an immediate value, then it is located in the immediate field and it appears either one or four bytes.

2)Explain instruction types:

Depending on operation they perform, all instructions are divided in several groups:

- Arithmetic Instructions
- Branch Instructions
- Data Transfer Instructions
- Logic Instructions
- Bit-oriented Instructions
  The first part of each instruction, called MNEMONIC refers to the operation an instruction performs (copy, addition, logic operation etc.). Mnemonics are abbreviations of the name of operation being executed. For example:

- INC R1 - Means: Increment register R1 (increment register R1);
- LJMP LAB5 - Means: Long Jump LAB5 (long jump to the address marked as LAB5);
- JNZ LOOP - Means: Jump if Not Zero LOOP (if the number in the accumulator is not 0, jump to the address marked as LOOP);
  The other part of instruction, called OPERAND is separated from mnemonic by at least one whitespace and defines data being processed by instructions. Some of the instructions have no operand, while some of them have one, two or three. If there is more than one operand in an instruction, they are separated by a comma. For example:

- RET - return from a subroutine;
- JZ TEMP - if the number in the accumulator is not 0, jump to the address marked as TEMP;
- ADD A,R3 - add R3 and accumulator;

- CJNE A,#20,LOOP - compare accumulator with 20. If they are not equal, jump to the address marked as LOOP;

Arithmetic instructions

Arithmetic instructions perform several basic operations such as addition, subtraction, division, multiplication etc. After execution, the result is stored in the first operand. For example: ADD A,R1 - The result of addition (A+R1) will be stored in the accumulator.

ARITHMETIC INSTRUCTIONS			
Mnemonic	**Description**	**Byte**	**Cycle**
ADD A,Rn	Adds the register to the accumulator	1	1
ADD A,direct	Adds the direct byte to the accumulator	2	2
ADD A,@Ri	Adds the indirect RAM to the accumulator	1	2
ADD A,#data	Adds the immediate data to the accumulator	2	2
ADDC A,Rn	Adds the register to the accumulator with a carry flag	1	1
ADDC A,direct	Adds the direct byte to the accumulator with a carry flag	2	2
ADDC A,@Ri	Adds the indirect RAM to the accumulator with a carry flag	1	2
ADDC A,#data	Adds the immediate data to the accumulator with a carry flag	2	2
SUBB A,Rn	Subtracts the register from the accumulator with a borrow	1	1

SUBB A,direct	Subtracts the direct byte from the accumulator with a borrow	2	2
SUBB A,@Ri	Subtracts the indirect RAM from the accumulator with a borrow	1	2
SUBB A,#data	Subtracts the immediate data from the accumulator with a borrow	2	2
INC A	Increments the accumulator by 1	1	1
INC Rn	Increments the register by 1	1	2
INC Rx	Increments the direct byte by 1	2	3
INC @Ri	Increments the indirect RAM by 1	1	3
DEC A	Decrements the accumulator by 1	1	1
DEC Rn	Decrements the register by 1	1	1
DEC Rx	Decrements the direct byte by 1	1	2
DEC @Ri	Decrements the indirect RAM by 1	2	3
INC DPTR	Increments the Data Pointer by 1	1	3
MUL AB	Multiplies A and B	1	5
DIV AB	Divides A by B	1	5
DA A	Decimal adjustment of the accumulator	1	1

	according to BCD code		

Branch Instructions

There are two kinds of branch instructions: Unconditional jump instructions: upon their execution a jump to a new location from where the program continues execution is executed. Conditional jump instructions: a jump to a new program location is executed only if a specified condition is met. Otherwise, the program normally proceeds with the next instruction.

BRANCH INSTRUCTIONS			
Mnemonic	**Description**	**Byte**	**Cycle**
ACALL addr11	Absolute subroutine call	2	6
LCALL addr16	Long subroutine call	3	6
RET	Returns from subroutine	1	4
RETI	Returns from interrupt subroutine	1	4
AJMP addr11	Absolute jump	2	3
LJMP addr16	Long jump	3	4
SJMP rel	Short jump (from –128 to +127 locations relative to the following instruction)	2	3
JC rel	Jump if carry flag is set. Short jump.	2	3
JNC rel	Jump if carry flag is not set. Short jump.	2	3
JB bit,rel	Jump if direct bit is set. Short jump.	3	4
JBC bit,rel	Jump if direct bit is set and clears bit. Short jump.	3	4

JMP @A+DPTR	Jump indirect relative to the DPTR	1	2
JZ rel	Jump if the accumulator is zero. Short jump.	2	3
JNZ rel	Jump if the accumulator is not zero. Short jump.	2	3
CJNE A,direct,rel	Compares direct byte to the accumulator and jumps if not equal. Short jump.	3	4
CJNE A,#data,rel	Compares immediate data to the accumulator and jumps if not equal. Short jump.	3	4
CJNE Rn,#data,rel	Compares immediate data to the register and jumps if not equal. Short jump.	3	4
CJNE @Ri,#data,rel	Compares immediate data to indirect register and jumps if not equal. Short jump.	3	4
DJNZ Rn,rel	Decrements register and jumps if not 0. Short jump.	2	3
DJNZ Rx,rel	Decrements direct byte and jump if not 0. Short jump.	3	4
NOP	No operation	1	1

Data Transfer Instructions

Data transfer instructions move the content of one register to another. The register the content of which is moved remains unchanged. If they have the suffix “X” (MOVX), the data is exchanged with external memory.

DATA TRANSFER INSTRUCTIONS			
Mnemonic	**Description**	**Byte**	**Cycle**
MOV A,Rn	Moves the register to the accumulator	1	1
MOV A,direct	Moves the direct byte to the accumulator	2	2
MOV A,@Ri	Moves the indirect RAM to the accumulator	1	2
MOV A,#data	Moves the immediate data to the accumulator	2	2
MOV Rn,A	Moves the accumulator to the register	1	2
MOV Rn,direct	Moves the direct byte to the register	2	4
MOV Rn,#data	Moves the immediate data to the register	2	2
MOV direct,A	Moves the accumulator to the direct byte	2	3
MOV direct,Rn	Moves the register to the direct byte	2	3
MOV direct,direct	Moves the direct byte to the direct byte	3	4
MOV direct,@Ri	Moves the indirect RAM to the direct byte	2	4
MOV direct,#data	Moves the immediate data to the direct byte	3	3
MOV @Ri,A	Moves the accumulator to the indirect RAM	1	3
MOV @Ri,direct	Moves the direct byte to the indirect RAM	2	5

MOV @Ri,#data	Moves the immediate data to the indirect RAM	2	3
MOV DPTR,#data	Moves a 16-bit data to the data pointer	3	3
MOVC A,@A+DPTR	Moves the code byte relative to the DPTR to the accumulator (address=A+DPTR)	1	3
MOVC A,@A+PC	Moves the code byte relative to the PC to the accumulator (address=A+PC)	1	3
MOVX A,@Ri	Moves the external RAM (8-bit address) to the accumulator	1	3-10
MOVX A,@DPTR	Moves the external RAM (16-bit address) to the accumulator	1	3-10
MOVX @Ri,A	Moves the accumulator to the external RAM (8-bit address)	1	4-11
MOVX @DPTR,A	Moves the accumulator to the external RAM (16-bit address)	1	4-11
PUSH direct	Pushes the direct byte onto the stack	2	4
POP direct	Pops the direct byte from the stack/td>	2	3
XCH A,Rn	Exchanges the register with the accumulator	1	2
XCH A,direct	Exchanges the direct byte with the accumulator	2	3

XCH A,@Ri	Exchanges the indirect RAM with the accumulator	1	3
XCHD A,@Ri	Exchanges the low-order nibble indirect RAM with the accumulator	1	3

Logic Instructions

Logic instructions perform logic operations upon corresponding bits of two registers. After execution, the result is stored in the first operand.

LOGIC INSTRUCTIONS			
Mnemonic	**Description**	**Byte**	**Cycle**
ANL A,Rn	AND register to accumulator	1	1
ANL A,direct	AND direct byte to accumulator	2	2
ANL A,@Ri	AND indirect RAM to accumulator	1	2
ANL A,#data	AND immediate data to accumulator	2	2
ANL direct,A	AND accumulator to direct byte	2	3
ANL direct,#data	AND immediae data to direct register	3	4
ORL A,Rn	OR register to accumulator	1	1
ORL A,direct	OR direct byte to accumulator	2	2
ORL A,@Ri	OR indirect RAM to accumulator	1	2
ORL direct,A	OR accumulator to direct byte	2	3

ORL direct,#data	OR immediate data to direct byte	3	4
XRL A,Rn	Exclusive OR register to accumulator	1	1
XRL A,direct	Exclusive OR direct byte to accumulator	2	2
XRL A,@Ri	Exclusive OR indirect RAM to accumulator	1	2
XRL A,#data	Exclusive OR immediate data to accumulator	2	2
XRL direct,A	Exclusive OR accumulator to direct byte	2	3
XORL direct,#data	Exclusive OR immediate data to direct byte	3	4
CLR A	Clears the accumulator	1	1
CPL A	Complements the accumulator (1=0, 0=1)	1	1
SWAP A	Swaps nibbles within the accumulator	1	1
RL A	Rotates bits in the accumulator left	1	1
RLC A	Rotates bits in the accumulator left through carry	1	1
RR A	Rotates bits in the accumulator right	1	1

RRC A	Rotates bits in the accumulator right through carry	1	1

Bit-oriented Instructions

Similar to logic instructions, bit-oriented instructions perform logic operations. The difference is that these are performed upon single bits.

BIT-ORIENTED INSTRUCTIONS			
Mnemonic	**Description**	**Byte**	**Cycle**
CLR C	Clears the carry flag	1	1
CLR bit	Clears the direct bit	2	3
SETB C	Sets the carry flag	1	1
SETB bit	Sets the direct bit	2	3
CPL C	Complements the carry flag	1	1
CPL bit	Complements the direct bit	2	3
ANL C,bit	AND direct bit to the carry flag	2	2
ANL C,/bit	AND complements of direct bit to the carry flag	2	2
ORL C,bit	OR direct bit to the carry flag	2	2
ORL C,/bit	OR complements of direct bit to the carry flag	2	2
MOV C,bit	Moves the direct bit to the carry flag	2	2
MOV bit,C	Moves the carry flag to the direct bit	2	3

3)Explain instruction set completeness

A set of instructions is said to be complete if the computer includes a sufficient number of instructions in each of the following categories:

- Arithmetic, logical and shift instructions
- A set of instructions for moving information to and from memory and processor registers.
- Instructions which controls the program together with instructions that check status conditions.
- Input and Output instructions

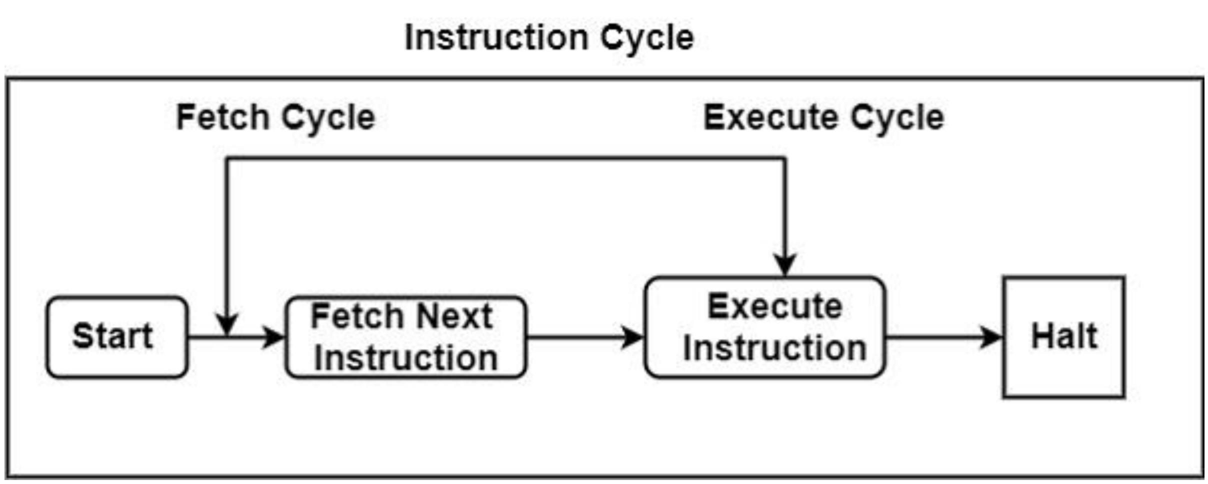
Arithmetic, logic and shift instructions provide computational capabilities for processing the type of data the user may wish to employ.
A huge amount of binary information is stored in the memory unit, but all computations are done in processor registers. Therefore, one must possess the capability of moving information between these two units.
Program control instructions such as branch instructions are used change the sequence in which the program is executed.
Input and Output instructions act as an interface between the computer and the user. Programs and data must be transferred into memory, and the results of computations must be transferred back to the user.

4)Describe instruction cycle:

A program consisting of the memory unit of the computer includes a series of instructions. The program is implemented on the computer by going through a cycle for each instruction.
In the basic computer, each instruction cycle includes the following procedures –

- It can fetch instruction from memory.
- It is used to decode the instruction.
- It can read the effective address from memory if the instruction has an indirect address.
- It can execute the instruction.

After the following four procedures are done, the control switches back to the first step and repeats the similar process for the next instruction. Therefore, the cycle continues until a **Halt** condition is met. The figure shows the phases contained in the instruction cycle.



As display in the figure, the halt condition appears when the device receive turned off, on the circumstance of unrecoverable errors, etc.

Fetch Cycle

The address instruction to be implemented is held at the program counter. The processor fetches the instruction from the memory that is pointed by the PC.

Next, the PC is incremented to display the address of the next instruction. This instruction is loaded onto the instruction register. The processor reads the instruction and executes the important procedures.

Execute Cycle

The data transfer for implementation takes place in two methods are as follows −

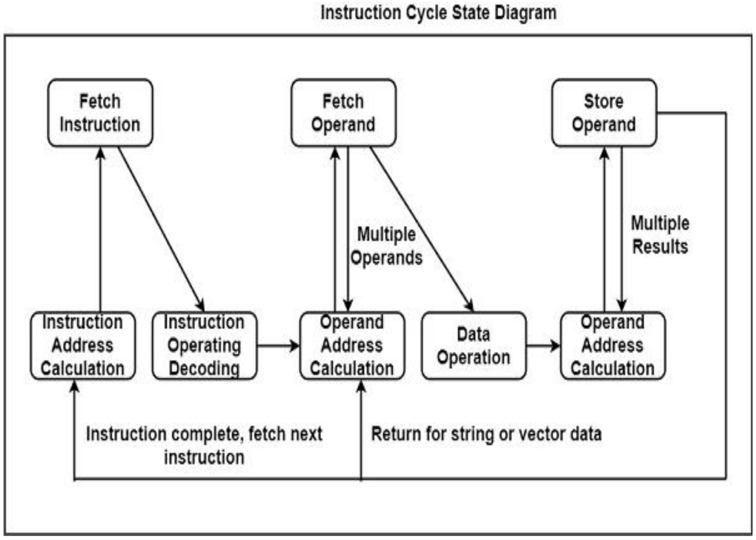
- **Processor-memory** − The data sent from the processor to memory or from memory to processor.
- **Processor-Input/Output** − The data can be transferred to or from a peripheral device by the transfer between a processor and an I/O device.

In the execute cycle, the processor implements the important operations on the information, and consistently the control calls for the modification in the sequence of data implementation. These two methods associate and complete the execute cycle.

State Diagram for Instruction Cycle

The figure provides a large aspect of the instruction cycle of a basic computer, which is in the design of a state diagram. For an instruction cycle, various states can be null, while others can be visited more than once.

Instruction Cycle State Diagram



- **Instruction Address Calculation** − The address of the next instruction is computed. A permanent number is inserted to the address of the earlier instruction.
- **Instruction Fetch** − The instruction is read from its specific memory location to the processor.
- **Instruction Operation Decoding** − The instruction is interpreted and the type of operation to be implemented and the operand(s) to be used are decided.
- **Operand Address Calculation** − The address of the operand is evaluated if it has a reference to an operand in memory or is applicable through the Input/Output.
- **Operand Fetch** − The operand is read from the memory or the I/O.
- **Data Operation** − The actual operation that the instruction contains is executed.
- **Store Operands** − It can store the result acquired in the memory or transfer it to the I/O.

Unit 4:Memory

4.1)Introduce memory and its types:

A memory is just like a human brain. It is used to store data and instructions. Computer memory is the storage space in the computer, where data is to be processed and instructions required for processing are stored. The memory is divided into large number of small parts called cells. Each location or cell has a unique address, which varies from zero to memory size minus one. For example, if the computer has 64k words, then this memory unit has 64 * 1024 = 65536 memory locations. The address of these locations varies from 0 to 65535.

Memory is primarily of three types −

- Cache Memory
- Primary Memory/Main Memory
- Secondary Memory

4.1.1)Cache memory: Cache memory is a very high speed semiconductor memory which can speed up the CPU. It acts as a buffer between the CPU and the main memory. It is used to hold those parts of data and program which are most frequently used by the CPU. The parts of data and programs are transferred from the disk to cache memory by the operating system, from where the CPU can access them.

Advantages

The advantages of cache memory are as follows −

- Cache memory is faster than main memory.
- It consumes less access time as compared to main memory.
- It stores the program that can be executed within a short period of time.
- It stores data for temporary use.

Disadvantages

The disadvantages of cache memory are as follows −

- Cache memory has limited capacity.
- It is very expensive.

4.1.2)Primary memory: Primary memory holds only those data and instructions on which the computer is currently working. It has a limited capacity and data is lost when power is switched off. It is generally made up of semiconductor device. These memories are not as fast as registers. The data and instruction required to be processed resides in the main memory. It is divided into two subcategories RAM and ROM.

Characteristics of Main Memory

- These are semiconductor memories.
- It is known as the main memory.
- Usually volatile memory.
- Data is lost in case power is switched off.
- It is the working memory of the computer.
- Faster than secondary memories.
- A computer cannot run without the primary memory.

4.1.3)Secondary memory: This type of memory is also known as external memory or non-volatile. It is slower than the main memory. These are used for storing data/information permanently. CPU directly does not access these memories, instead they are accessed via input-output routines. The contents of secondary memories are first transferred to the main memory, and then the CPU can access it. For example, disk, CD-ROM, DVD, etc.

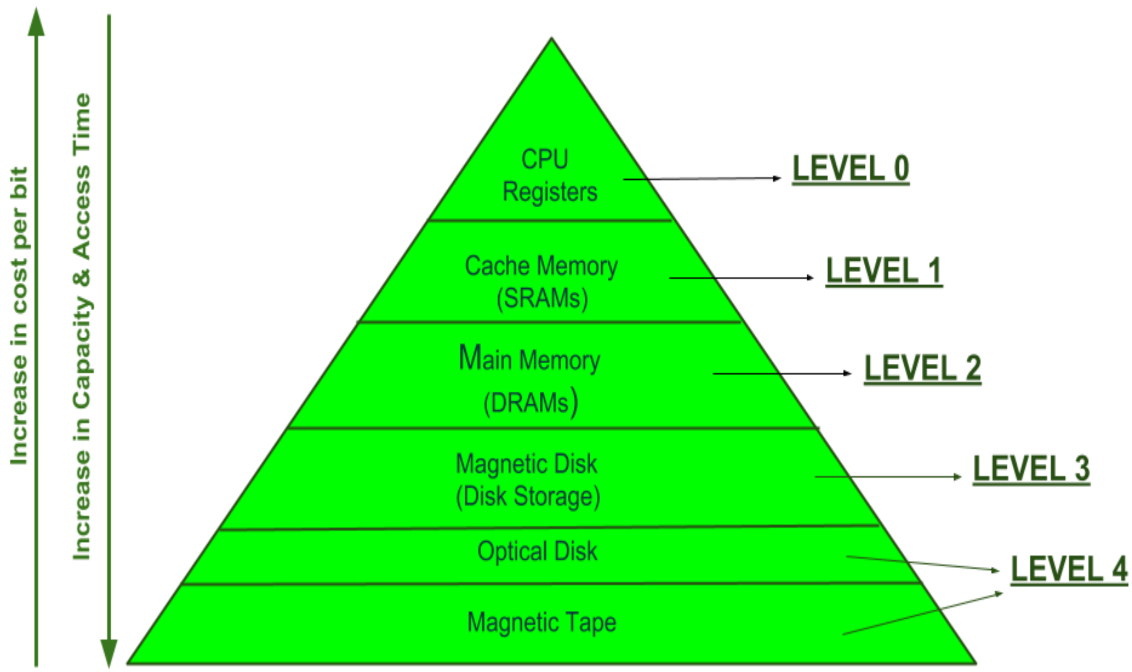
Characteristics of Secondary Memory

- These are magnetic and optical memories.

- It is known as the backup memory.
- It is a non-volatile memory.
- Data is permanently stored even if power is switched off.
- It is used for storage of data in a computer.
- Computer may run without the secondary memory.
- Slower than primary memories.

4.2)Describe memory hierarchy:

Memory Hierarchy is an enhancement to organize the memory such that it can minimize the access time. The Memory Hierarchy was developed based on a program behavior known as locality of references.The figure below clearly demonstrates the different levels of memory hierarchy :



MEMORY HIERARCHY DESIGN

This Memory Hierarchy Design is divided into 2 main types:

1. **External Memory or Secondary Memory –**
   Comprising of Magnetic Disk, Optical Disk, Magnetic Tape i.e. peripheral storage devices which are accessible by the processor via I/O Module.
2. **Internal Memory or Primary Memory –**
   Comprising of Main Memory, Cache Memory & CPU registers. This is directly accessible by the processor.

We can infer the following characteristics of Memory Hierarchy Design from above figure:

1. **Capacity:**
   It is the global volume of information the memory can store. As we move from top to bottom in the Hierarchy, the capacity increases.
2. **Access Time:**
   It is the time interval between the read/write request and the availability of the data. As we move from top to bottom in the Hierarchy, the access time increases.

3. **Performance:**
   Earlier when the computer system was designed without Memory Hierarchy design, the speed gap increases between the CPU registers and Main Memory due to large difference in access time. This results in lower performance of the system and thus, enhancement was required. This enhancement was made in the form of Memory Hierarchy Design because of which the performance of the system increases. One of the most significant ways to increase system performance is minimizing how far down the memory hierarchy one has to go to manipulate data.
4. **Cost per bit:**
   As we move from bottom to top in the Hierarchy, the cost per bit increases i.e. Internal Memory is costlier than External Memory.

4.3) Describe characteristics of Memory system:

The key characteristics of memory devices or memory system are as follows:

1. Location
2. Capacity
3. Unit of Transfer
4. Access Method
5. Performance
6. Physical type
7. Physical characteristics
8. Organization

1. Location:

It deals with the location of the memory device in the computer system. There are three possible locations:

- CPU : This is often in the form of CPU registers and small amount of cache
- Internal or main: This is the main memory like RAM or ROM. The CPU can directly access the main memory.
- External or secondary: It comprises of secondary storage devices like hard disks, magnetic tapes. The CPU doesn't access these devices directly. It uses device controllers to access secondary storage devices.

2. Capacity:

The capacity of any memory device is expressed in terms of: i)word size ii)Number of words

- **Word size:** Words are expressed in bytes (8 bits). A word can however mean any number of bytes. Commonly used word sizes are 1 byte (8 bits), 2bytes (16 bits) and 4 bytes (32 bits).

- **Number of words:** This specifies the number of words available in the particular memory device. For example, if a memory device is given as 4K x 16.This means the device has a word size of 16 bits and a total of 4096(4K) words in memory.

3. Unit of Transfer:

It is the maximum number of bits that can be read or written into the memory at a time. In case of main memory, it is mostly equal to word size. In case of external memory, unit of transfer is not limited to the word size; it is often larger and is referred to as blocks.

4. Access Methods:

It is a fundamental characteristic of memory devices. It is the sequence or order in which memory can be accessed. There are three types of access methods:

- **Random Access:** If storage locations in a particular memory device can be accessed in any order and access time is independent of the memory location being accessed. Such memory devices are said to have a random access mechanism. RAM (Random Access Memory) IC's use this access method.

- **Serial Access:** If memory locations can be accessed only in a certain predetermined sequence, this access method is called serial access. Magnetic Tapes, CD-ROMs employ serial access methods.

- **Semi random Access:** Memory devices such as Magnetic Hard disks use this access method. Here each track has a read/write head thus each track can be accessed randomly but access within each track is restricted to a serial access.

5. Performance: The performance of the memory system is determined using three parameters

- **Access Time:** In random access memories, it is the time taken by memory to complete the read/write operation from the instant that an address is sent to the memory. For non-random access memories, it is the time taken to position the read write head at the desired location. Access time is widely used to measure performance of memory devices.

- **Memory cycle time:** It is defined only for Random Access Memories and is the sum of the access time and the additional time required before the second access can commence.
- **Transfer rate:** It is defined as the rate at which data can be transferred into or out of a memory unit.

6. Physical type: Memory devices can be either semiconductor memory (like RAM) or magnetic surface memory (like Hard disks).

7.Physical Characteristics:

- **Volatile/Non- Volatile:** If a memory devices continues hold data even if power is turned off. The memory device is non-volatile else it is volatile.

8. Organization:

- **Erasable/Non-erasable:** The memories in which data once programmed cannot be erased are called Non-erasable memories. Memory devices in which data in the memory can be erased is called erasable memory.

  E.g. RAM(erasable), ROM(non-erasable).

4.4) Elaborate Memory Measurement Unit:

Memory unit is the amount of data that can be stored in the storage unit. This storage capacity is expressed in terms of Bytes.

S.No.	Unit & Description
1	**Kilobyte (KB)** 1 KB = 1024 Bytes
2	**Megabyte (MB)** 1 MB = 1024 KB
3	**GigaByte (GB)** 1 GB = 1024 MB
4	**TeraByte (TB)** 1 TB = 1024 GB
5	**PetaByte (PB)** 1 PB = 1024 TB

4.5) Introduce Memory address:

A memory address is a unique identifier used by a device or CPU for data tracking. This binary address is defined by an ordered and finite sequence allowing the CPU to track the location of each memory byte.
Modern computers are addressed by bytes which are assigned to memory addresses – binary numbers assigned to a random access memory (RAM) cell that holds up to one byte. Data greater than one byte is consecutively segmented into multiple bytes with a series of corresponding addresses.
Hardware devices and CPUs track stored data by accessing memory addresses via data buses.
Before CPU processing, data and programs must be stored in unique memory address locations.
The bus determines a fixed number of CPU memory addresses assigned according to CPU requirements.
The CPU then processes physical memory in individual segments.

Memory addresses are usually allocated during the boot process. This initiates the startup BIOS on the ROM BIOS chip, which becomes the assigned address. To enable immediate video capability, the first memory addresses are assigned to video ROM and RAM, followed by the following assigned memory addresses:

- Expansion card ROM and RAM chips
- Motherboard dual inline memory modules, single inline memory modules or Rambus inline memory modules
- Other devices

4.6)Describe Error-Correcting Codes:

Error-correcting codes (ECC) are a sequence of numbers generated by specific algorithms for detecting and removing errors in data that has been transmitted over noisy channels.

When bits are transmitted over the computer network, they are subject to get corrupted due to interference and network problems. The corrupted bits leads to spurious data being received by the receiver and are called errors.

Error correcting codes ascertain the exact number of bits that has been corrupted and the location of the corrupted bits, within the limitations in algorithm. This method of correcting errors at the receiver's end is called forward error correction.

Types of Error Correcting Codes

ECCs can be broadly categorized into two types, block codes and convolution codes.

- **Block codes** − The message is divided into fixed-sized blocks of bits, to which redundant bits are added for error detection or correction.
- **Convolutional codes** − The message comprises of data streams of arbitrary length and parity symbols are generated by the sliding application of a Boolean function to the data stream.

Common Error Correcting Codes

There are four popularly used error correction codes.

- **Hamming Codes** − It is a block code that is capable of detecting up to two simultaneous bit errors and correcting single-bit errors.
- **Binary Convolution Code** − Here, an encoder processes an input sequence of bits of arbitrary length and generates a sequence of output bits.
- **Reed - Solomon Code** − They are block codes that are capable of correcting burst errors in the received data block.
- **Low-Density Parity Check Code** − It is a block code specified by a parity-check matrix containing a low density of 1s. They are suitable for large block sizes in very noisy channels.

Hamming Code

Hamming code is a block code that is capable of detecting up to two simultaneous bit errors and correcting single-bit errors. It was developed by R.W. Hamming for error correction.

In this coding method, the source encodes the message by inserting redundant bits within the message. These redundant bits are extra bits that are generated and inserted at specific positions in the message itself to enable error detection and correction. When the destination receives this message, it performs recalculations to detect errors and find the bit position that has error.

Encoding a message by Hamming Code

The procedure used by the sender to encode the message encompasses the following steps −

- **Step 1** − Calculation of the number of redundant bits.
- **Step 2** − Positioning the redundant bits.
- **Step 3** − Calculating the values of each redundant bit.

Unit 5: Processor

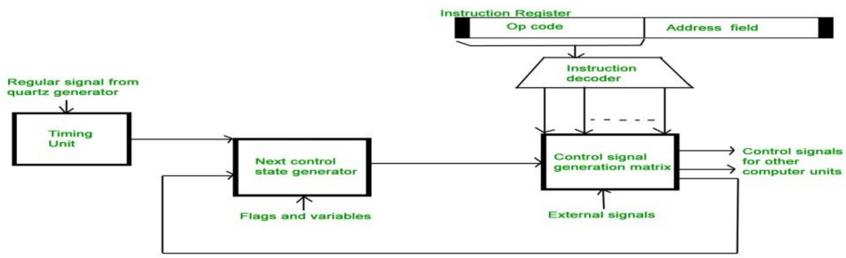
1)Introduce Hardwired and Micro Programmed:

To execute an instruction, the control unit of the CPU must generate the required control signal in the proper sequence. There are two approaches used for generating the control signals in proper sequence as Hardwired Control unit and Micro-programmed control unit.

Hardwired Control Unit –

The control hardware can be viewed as a state machine that changes from one state to another in every clock cycle, depending on the contents of the instruction register, the condition codes and the external inputs. The outputs of the state machine are the control signals. The sequence of the operation carried out by this machine is determined by the wiring of the logic elements and hence named as "hardwired".

- Fixed logic circuits that correspond directly to the Boolean expressions are used to generate the control signals.
- Hardwired control is faster than micro-programmed control.
- A controller that uses this approach can operate at high speed.
- RISC architecture is based on hardwired control unit

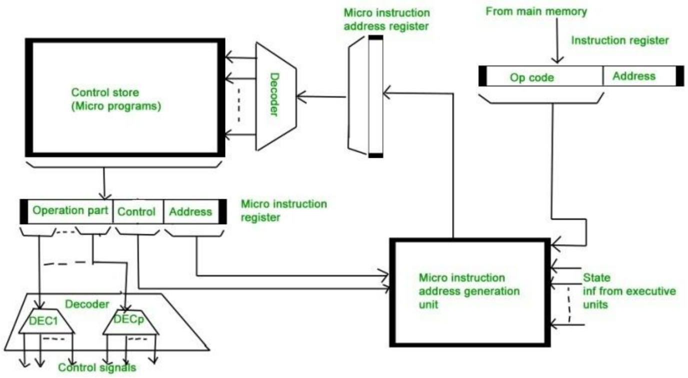


Micro-programmed Control Unit –

- The control signals associated with operations are stored in special memory units inaccessible by the programmer as Control Words.
- Control signals are generated by a program are similar to machine language programs.
- Micro-programmed control unit is slower in speed because of the time it takes to fetch microinstructions from the control memory.

Some Important Terms –

1. **Control Word :** A control word is a word whose individual bits represent various control signals.
2. **Micro-routine :** A sequence of control words corresponding to the control sequence of a machine instruction constitutes the micro-routine for that instruction.
3. **Micro-instruction :** Individual control words in this micro-routine are referred to as microinstructions.
4. **Micro-program :** A sequence of micro-instructions is called a micro-program, which is stored in a ROM or RAM called a Control Memory (CM).
5. **Control Store :** the micro-routines for all instructions in the instruction set of a computer are stored in a special memory called the Control Store.



Types of Micro-programmed Control Unit – Based on the type of Control Word stored in the Control Memory (CM), it is classified into two types :

1. Horizontal Micro-programmed control Unit :

The control signals are represented in the decoded binary format that is 1 bit/CS. Example: If 53 Control signals are present in the processor than 53 bits are required. More than 1 control signal can be enabled at a time.

- It supports longer control word.
- It is used in parallel processing applications.
- It allows higher degree of parallelism. If degree is n, n CS are enabled at a time.
- It requires no additional hardware(decoders). It means it is faster than Vertical Microprogrammed.
- It is more flexible than vertical microprogrammed

2. Vertical Micro-programmed control Unit :

The control signals re represented in the encoded binary format. For N control signals- Log2(N) bits are required.

- It supports shorter control words.
- It supports easy implementation of new control signals therefore it is more flexible.
- It allows low degree of parallelism i.e., degree of parallelism is either 0 or 1.
- Requires an additional hardware (decoders) to generate control signals, it implies it is slower than horizontal microprogrammed.
- It is less flexible than horizontal but more flexible than that of hardwired control unit.

2)Explain Arithmetic and logical Unit:

An arithmetic logic unit (ALU) is a major component of the central processing unit of a computer system. It does all processes related to arithmetic and logic operations that need to be done on instruction words. In some microprocessor architectures, the ALU is divided into the arithmetic unit (AU) and the logic unit (LU).

An ALU can be designed by engineers to calculate any operation. As the operations become more complex, the ALU also becomes more expensive, takes up more space in the CPU and dissipates more heat. That is why engineers make the ALU powerful enough to ensure that the CPU is also powerful and fast, but not so complex as to become prohibitive in terms of cost and other disadvantages.An arithmetic logic unit is also known as an integer unit (IU).

The arithmetic logic unit is that part of the CPU that handles all the calculations the CPU may need. Most of these operations are logical in nature. Depending on how the ALU is designed, it can make the CPU more powerful, but it also consumes more energy and creates more heat. Therefore, there must be a balance between how powerful and complex the ALU is and how expensive the whole unit becomes. This is why faster CPUs are more expensive, consume more power and dissipate more heat.

The main functions of the ALU are to do arithmetic and logic operations, including bit shifting operations. These are essential processes that need to be done on almost any data that is being processed by the CPU.

ALUs routinely perform the following operations:

- Logical Operations: These include AND, OR, NOT, XOR, NOR, NAND, etc.
- Bit-Shifting Operations: This pertains to shifting the positions of the bits by a certain number of places to the right or left, which is considered a multiplication operation.
- Arithmetic Operations: This refers to bit addition and subtraction. Although multiplication and division are sometimes used, these operations are more expensive to make. Addition can be used to substitute for multiplication and subtraction for division.

3)Describe the types of registers:

Registers are a type of computer memory used to quickly accept, store, and transfer data and instructions that are being used immediately by the CPU. The registers used by the CPU are often termed as Processor registers.

A processor register may hold an instruction, a storage address, or any data (such as bit sequence or individual characters).

The computer needs processor registers for manipulating data and a register for holding a memory address. The register holding the memory location is used to calculate the address of the next instruction after the execution of the current instruction is completed.

Following is the list of some of the most common registers used in a basic computer:

Register	Symbol	Number of bits	Function
Data register	DR	16	Holds memory operand
Address register	AR	12	Holds address for the memory
Accumulator	AC	16	Processor register
Instruction register	IR	16	Holds instruction code
Program counter	PC	12	Holds address of the instruction
Temporary register	TR	16	Holds temporary data

Input register	INPR	8	Carries input character
Output register	OUTR	8	Carries output character

These are classified as given below.

- **Accumulator:**
  This is the most frequently used register used to store data taken from memory. It is in different numbers in different microprocessors.

- **Memory Address Registers (MAR):**
  It holds the address of the location to be accessed from memory. MAR and MDR (Memory Data Register) together facilitate the communication of the CPU and the main memory.

- **Memory Data Registers (MDR):**
  It contains data to be written into or to be read out from the addressed location.

- **General Purpose Registers:**
  These are numbered as R0, R1, R2….Rn-1, and used to store temporary data during any ongoing operation. Its content can be accessed by assembly programming. Modern CPU architectures tends to use more GPR so that register-to-register addressing can be used more, which is comparatively faster than other addressing modes.

- **Program Counter (PC):**
  Program Counter (PC) is used to keep the track of execution of the program. It contains the memory address of the next instruction to be fetched. PC points to the address of the next instruction to be fetched from the main memory when the previous instruction has been successfully completed. Program Counter (PC) also functions to count the number of instructions. The incrementation of PC depends on the type of architecture being used. If we are using 32-bit architecture, the PC gets incremented by 4 every time to fetch the next instruction.

- **Instruction Register (IR):**
  The IR holds the instruction which is just about to be executed. The instruction from PC is fetched and stored in IR. As soon as the instruction in placed in IR, the CPU starts executing the instruction and the PC points to the next instruction to be executed.

- **Condition code register (CCR) :**
  Condition code registers contain different flags that indicate the status of any operation.for instance lets suppose an operation caused creation of a negative result or zero, then these flags are set high accordingly.and the flags are

1. Carry C: Set to 1 if an add operation produces a carry or a subtract operation produces a borrow; otherwise cleared to 0.
2. Overflow V: Useful only during operations on signed integers.
3. Zero Z: Set to 1 if the result is 0, otherwise cleared to 0.
4. Negate N: Meaningful only in signed number operations. Set to 1 if a negative result is produced.
5. Extend X: Functions as a carry for multiple precision arithmetic operations.

   These are generally decided by ALU.

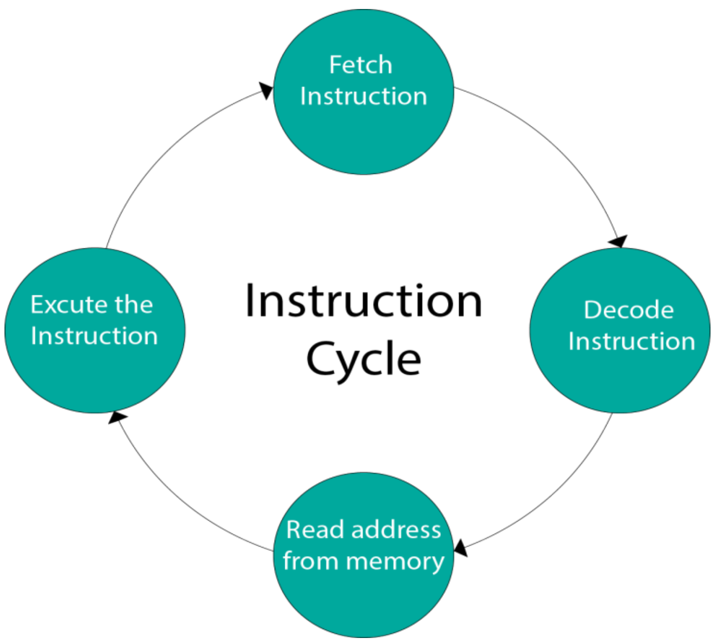
So, these are the different registers which are operating for a specific purpose.

4)Elaborate instruction Execution:

A program residing in the memory unit of a computer consists of a sequence of instructions. These instructions are executed by the processor by going through a cycle for each instruction.

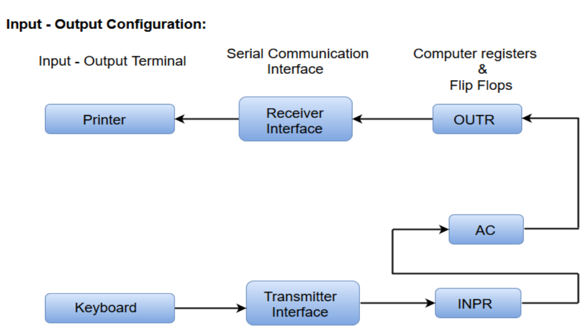
In a basic computer, each instruction cycle consists of the following phases:

1. Fetch instruction from memory.
2. Decode the instruction.
3. Read the effective address from memory.
4. Execute the instruction.



Input-Output Configuration

In computer architecture, input-output devices act as an interface between the machine and the user. Instructions and data stored in the memory must come from some input device. The results are displayed to the user
through some output device.



The following block diagram shows the input-output configuration for a basic computer.

- The input-output terminals send and receive information.
- The amount of information transferred will always have eight bits of an alphanumeric code.
- The information generated through the keyboard is shifted into an input register 'INPR'.
- The information for the printer is stored in the output register 'OUTR'.
- Registers INPR and OUTR communicate with a communication interface serially and with the AC in parallel.
- The transmitter interface receives information from the keyboard and transmits it to INPR.
- The receiver interface receives information from OUTR and sends it to the printer serially.

5)Describe bus architecture:

Bus is a group of wires that connects different components of the computer. It is used for transmitting data, control signal and memory address from one component to another. A bus can be 8 bit, 16 bit, 32 bit and 64 bit. A 32 bit bus can transmit 32 bit information at a time. A bus can be internal or external.
Types of bus:

a)Data bus
Data bus carries data from on component to another. It is uni-directional for input and output devices and bi-directional for memory and CPU.

b)Control bus
Control bus carries control signal. CU of CPU uses control signal for controlling all the components. It is uni-directional from CPU to all other components.

c)Address bus
Address bus carries memory address. A memory address is a numerical value used for identifying a memory location. Computer performs all its task through the memory address. CU of CPU sends memory address to all the components. So, address bus is also uni-directional from CPU to all other components.

6)Introduce the addressing modes:

Addressing Modes– The term addressing modes refers to the way in which the operand of an instruction is specified. The addressing mode specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually executed.

Addressing modes for 8086 instructions are divided into two categories:

1) Addressing modes for data
2) Addressing modes for branch
The 8086 memory addressing modes provide flexible access to memory, allowing you to easily access variables, arrays, records, pointers, and other complex data types. The key to good assembly language programming is the proper use of memory addressing modes.
An assembly language program instruction consists of two parts

The memory address of an operand consists of two components:

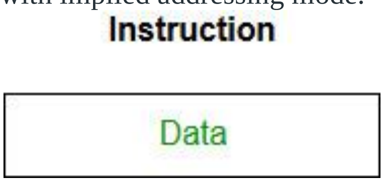
IMPORTANT TERMS

- **Starting address** of memory segment.
- **Effective address or Offset**: An offset is determined by adding any combination of three address elements: **displacement, base and index.**
    - **Displacement:** It is an 8 bit or 16 bit immediate value given in the instruction.
    - **Base**: Contents of base register, BX or BP.
    - **Index**: Content of index register SI or DI.

According to different ways of specifying an operand by 8086 microprocessor, different addressing modes are used by 8086.

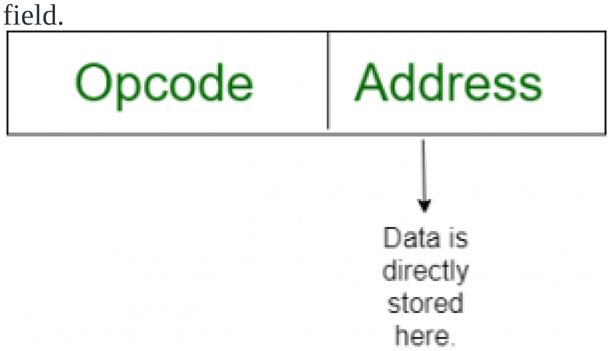
Addressing modes used by 8086 microprocessor are discussed below:

- **Implied mode:**: In implied addressing the operand is specified in the instruction itself. In this mode the data is 8 bits or 16 bits long and data is the part of instruction.Zero address instruction are designed with implied addressing mode.

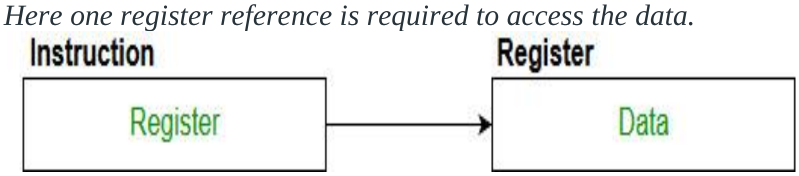
  Example: CLC (used to reset Carry flag to 0)

- **Immediate addressing mode (symbol #):**In this mode data is present in address field of instruction .Designed like one address instruction format.
  Note: Limitation in the immediate mode is that the range of constants are restricted by size of address field.

  Example: MOV AL, 35H (move the data 35H into AL register)

- **Register mode:** In register addressing the operand is placed in one of 8 bit or 16 bit general purpose registers. The data is in the register that is specified by the instruction.
  Here one register reference is required to access the data.

  Example: MOV AX,CX (move the contents of CX register to AX register)

- **Register Indirect mode**: In this addressing the operand's offset is placed in any one of the registers BX,BP,SI,DI as specified in the instruction. The effective address of the data is in the base register or an index register that is specified by the instruction.

Here two register reference is required to access the data.

The 8086 CPUs let you access memory indirectly through a register using the register indirect addressing modes.

- MOV AX, [BX](move the contents of memory location s

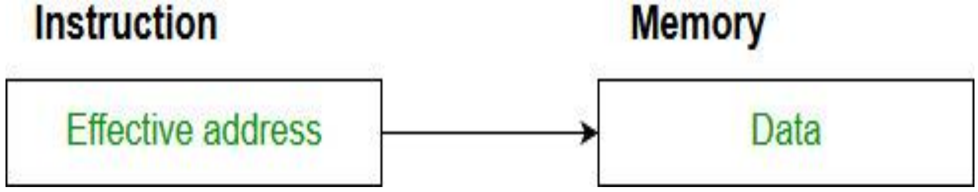
  addressed by the register BX to the register AX)

- **Auto Indexed (increment mode)**: Effective address of the operand is the contents of a register specified in the instruction. After accessing the operand, the contents of this register are automatically incremented to point to the next consecutive memory location.**(R1)+**.
  Here one register reference,one memory reference and one ALU operation is required to access the data.
  Example:
- Add R1, (R2)+ // OR
- R1 = R1 +M[R2]

  R2 = R2 + d

  Useful for stepping through arrays in a loop. R2 – start of array d – size of an element
- **Auto indexed (decrement mode)**: Effective address of the operand is the contents of a register specified in the instruction. Before accessing the operand, the contents of this register are automatically decremented to point to the previous consecutive memory location. –**(R1)**
  Here one register reference,one memory reference and one ALU operation is required to access the data.
  Example:
  Add R1,-(R2) //OR

  R2 = R2-*d*
  R1 = R1 + M[R2]
  Auto decrement mode is same as auto increment mode. Both can also be used to implement a stack as push and pop . Auto increment and Auto decrement modes are useful for implementing "Last-In-First-Out" data structures.
- **Direct addressing/ Absolute addressing Mode (symbol [ ]):** The operand's offset is given in the instruction as an 8 bit or 16 bit displacement element. In this addressing mode the 16 bit effective address of the data is the part of the instruction.
  Here only one memory reference operation is required to access the data.



Example:ADD AL,[0301] //add the contents of offset address 0301 to AL

- **Indirect addressing Mode (symbol @ or ()):**In this mode address field of instruction contains the address of effective address.Here two references are required.
  1st reference to get effective address.
  2nd reference to access the data.
  Based on the availability of Effective address, Indirect mode is of two kind:

1. Register Indirect:In this mode effective address is in the register, and corresponding register name will be maintained in the address field of an instruction.
   Here one register reference,one memory reference is required to access the data.
2. Memory Indirect:In this mode effective address is in the memory, and corresponding memory address will be maintained in the address field of an instruction.
   Here two memory reference is required to access the data.

- **Indexed addressing mode**: The operand's offset is the sum of the content of an index register SI or DI and an 8 bit or 16 bit displacement.
  Example:MOV AX, [SI +05]
- **Based Indexed Addressing:** The operand's offset is sum of the content of a base register BX or BP and an index register SI or DI.
  Example: ADD AX, [BX+SI]

Based on Transfer of control, addressing modes are:

- **PC relative addressing mode:** PC relative addressing mode is used to implement intra segment transfer of control, In this mode effective address is obtained by adding displacement to PC.
- EA= PC + Address field value
  PC= PC + Relative value.
- **Base register addressing mode:**Base register addressing mode is used to implement inter segment transfer of control.In this mode effective address is obtained by adding base register value to address field value.
- EA= Base register + Address field value.
- PC= Base register + Relative value.

Note:

1. PC relative nad based register both addressing modes are suitable for program relocation at runtime.
2. Based register addressing mode is best suitable to write position independent codes.

Advantages of Addressing Modes

6. To give programmers to facilities such as Pointers, counters for loop controls, indexing of data and program relocation.
7. To reduce the number bits in the addressing field of the Instruction.

7)Explain about types of processor:

The following article provides an outline for Types of Processor. The processor is defined as a logic circuit or simple chip which reacts to fundamental instructions and input processes to operate the computer. The important purposes of a processor are getting, decoding, processing, executing and writing back as feedback to the instructions of the chip. The processor is termed as the brain of any electronic systems that incorporate into a laptop, computers, smartphones, and embedded systems. The control unit and arithmetic logic unit are the two significant components of the processors.

Different Types of Processor

The different types of processors are microprocessor, microcontroller, embedded processor, digital signal processor and the processors can be varied according to the devices. The important elements of the CPU are labelled as heart elements of the processor and system. The control unit activates, fetches, and execute the input instructions. The processor can be embedded in a microprocessor and comprise of unit IC chip. But some devices are based on multi-core processors. It comprises one or more CPU. It is a typical tiny component with pins embedded on the motherboard. It can also be linked to motherboard with fan and heat sink to disperse the produced heat.

1. Microprocessor

The fundamental process of the system is denoted by a microprocessor incorporated in the embedded systems. There are various types of microprocessors in the market implemented by different enterprises. The microprocessor is a standard processor which comprises of ALU, control unit and club of registers known as control registers, status registers, and scratchpad registers.

2. Microcontroller

The microcontroller is standard which is available in different size and packages. The input reading and reacting to its corresponding output is the fundamental function of the basic microcontroller and so it is called as general-purpose input and output processors (GPIO). Few of the microcontrollers are Microchip P1C16F877A, Microchip Atmega328, Microchip P1C18F45K22, Microchip P1C16F671, and Microchip P1C16F1503.

3. Embedded Processor

The embedded processor is structured to control the electrical and mechanical functions. It comprises of numerous blocks like timer, program memory, data memory, reset, power supply, data memory, interrupt controller, clock oscillator systems, interfacing circuits, specific circuits and system application ports and circuits.

4. Digital Signal Processor

The digital signal processor is used for filtering, measuring, compressing analogue and digital signals. The processing of signal means that manipulation and analysis of digital signals. This process can be made using application specified integration circuits, digital signal processor, field-programmable gate array or it can be a computer to achieve a distinct signal. The processors in DSP are used for barcode scanners, oscilloscope, printers, mobile phones. These processors are used for rapid and implied for real-time applications.

Components of Processor

The fundamental parts of the processor are control units, arithmetic logic unit, registers, floating points, L1 and L2 cache memory.

- The arithmetic logic unit is comprised of logical and arithmetic functions on the operands in instructions.
- The unit of floating-point is called as numeric coprocessor or math coprocessor. It is a specialized operator which manipulates the numbers in rapid when compared to the operation of basic microprocessor circuits.
- The registers are used to save the instructions and other data to feed the operands to ALU and store the operation result.
- The L2 and L1 cache memory saves the time of CPU to fetch the data from RAM.

8)Illustrate RISC and CISC:

a)Reduced Instruction Set Architecture (RISC) –

The main idea behind is to make hardware simpler by using an instruction set composed of a few basic steps for loading, evaluating, and storing operations just like a load command will load data, store command will store the data.

b)Complex Instruction Set Architecture (CISC) –

The main idea is that a single instruction will do all loading, evaluating, and storing operations just like a multiplication command will do stuff like loading data, evaluating, and storing it, hence it’s complex.

Both approaches try to increase the CPU performance

- **RISC:** Reduce the cycles per instruction at the cost of the number of instructions per program.

- **CISC:** The CISC approach attempts to minimize the number of instructions per program but at the cost of increase in number of cycles per instruction.

$$CPU\ Time = \frac{Seconds}{Program} = \frac{Instructions}{Program} X \frac{Cycles}{Instructions} X \frac{Seconds}{Cycle}$$

Earlier when programming was done using assembly language, a need was felt to make instruction do more tasks because programming in assembly was tedious and error-prone due to which CISC architecture evolved but with the uprise of high-level language dependency on assembly reduced RISC architecture prevailed.

Characteristic of RISC –

1. Simpler instruction, hence simple instruction decoding.
2. Instruction comes undersize of one word.
3. Instruction takes a single clock cycle to get executed.
4. More general-purpose registers.
5. Simple Addressing Modes.
6. Less Data types.
7. Pipeline can be achieved.

Characteristic of CISC –

1. Complex instruction, hence complex instruction decoding.
2. Instructions are larger than one-word size.
3. Instruction may take more than a single clock cycle to get executed.
4. Less number of general-purpose registers as operation get performed in memory itself.
5. Complex Addressing Modes.
6. More Data types.

Difference –

RISC	CISC
Focus on software	Focus on hardware
Uses only Hardwired control unit	Uses both hardwired and micro programmed control unit
Transistors are used for more registers	Transistors are used for storing complex Instructions
Fixed sized instructions	Variable sized instructions
Can perform only Register to Register Arithmetic operations	Can perform REG to REG or REG to MEM or MEM to MEM
Requires more number of registers	Requires less number of registers

RISC	CISC
Code size is large	Code size is small
An instruction execute in a single clock cycle	Instruction takes more than one clock cycle
An instruction fit in one word	Instructions are larger than the size of one word

Unit 6:Input/ Output Organization

6.1 Peripheral devices: Peripheral devices are those devices that are linked either internally or externally to a computer. These devices are commonly used to transfer data. The most common processes that are carried out in a computer are entering data and displaying processed data. Several devices can be used to receive data and display processed data. The devices used to perform these functions are called peripherals or I/O devices.

The following are some of the commonly used peripherals −

Keyboard

The keyboard is the most commonly used input device. It is used to provide commands to the computer. The commands are usually in the form of text. The keyboard consists of many keys such as function keys, numeric keypad, character keys, and various symbols.

Monitor

The most commonly used output device is the monitor. A cable connects the monitor to the video adapters in the computer's motherboard. These video adapters convert the electrical signals to the text and images that are displayed.

Printer

Printers provide a permanent record of computer data or text on paper. We can classify printers as impact and non-impact printers. Impact printers print characters due to the physical contact of the print head with the paper. In non-impact printers, there is no physical contact.

Magnetic Tape

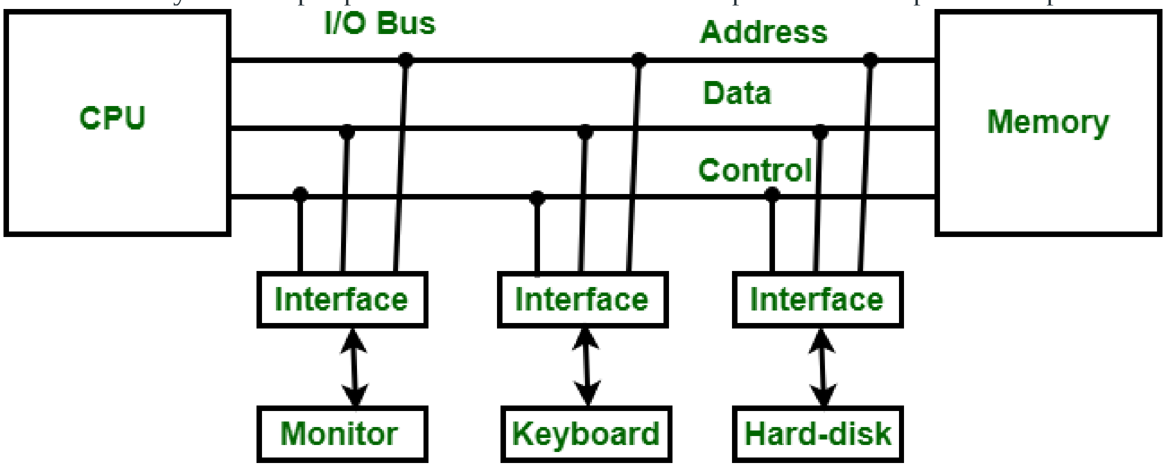
Magnetic tapes are used in most companies to store data files. Magnetic tapes use a read-write mechanism. The read-write mechanism refers to writing data on or reading data from a magnetic tape. The tapes sequentially store the data manner.

Magnetic Disk

There is another medium for storing data is magnetic disks. Magnetic disks have high-speed rotational surfaces coated with magnetic material. A read-write mechanism is used to achieve access to write on or read from the magnetic disk.

6.2 Basic I/O Interfaces:

Input-Output Interface is used as an method which helps in transferring of information between the internal storage devices i.e. memory and the external peripheral device . A peripheral device is that which provide input and output for the computer, it is also called Input-Output devices. For Example: A keyboard and mouse provide Input to the computer are called input devices while a monitor and printer that provide output to the computer are called output devices. Just like the external hard-drives, there is also availability of some peripheral devices which are able to provide both input and output.



Input-Output Interface

In micro-computer base system, the only purpose of peripheral devices is just to provide **special communication links** for the interfacing them with the CPU. To resolve the differences between peripheral devices and CPU, there is a special need for communication links.
The major differences are as follows:

1. The nature of peripheral devices is electromagnetic and electro-mechanical. The nature of the CPU is electronic. There is a lot of difference in the mode of operation of both peripheral devices and CPU.
2. There is also a synchronization mechanism because the data transfer rate of peripheral devices are slow than CPU.
3. In peripheral devices, data code and formats are differ from the format in the CPU and memory.
4. The operating mode of peripheral devices are different and each may be controlled so as not to disturb the operation of other peripheral devices connected to CPU.

Functions of Input-Output Interface:

1. It is used to synchronize the operating speed of CPU with respect to input-output devices.
2. It selects the input-output device which is appropriate for the interpretation of the input-output device.
3. It is capable of providing signals like control and timing signals.
4. In this data buffering can be possible through data bus.
5. There are various error detectors.
6. It converts serial data into parallel data and vice-versa.
7. It also convert digital data into analog signal and vice-versa.

6.3 I/O Technique :

The method that is used to transfer information between internal storage and external I/O devices is known as I/O interface. The CPU is interfaced using special communication links by the peripherals connected to any computer system. These communication links are used to resolve the differences between CPU and peripheral. There exists special hardware components between CPU and peripherals to supervise and synchronize all the input and output transfers that are called interface units.

Mode of Transfer:
The binary information that is received from an external device is usually stored in the memory unit. The information that is transferred from the CPU to the external device is originated from the memory unit. CPU merely processes the information but the source and target is always the memory unit. Data transfer between CPU and the I/O devices may be done in different modes.

Data transfer to and from the peripherals may be done in any of the three possible ways

1. Programmed I/O.
2. Interrupt- initiated I/O.
3. Direct memory access(DMA).

Now let's discuss each mode one by one.

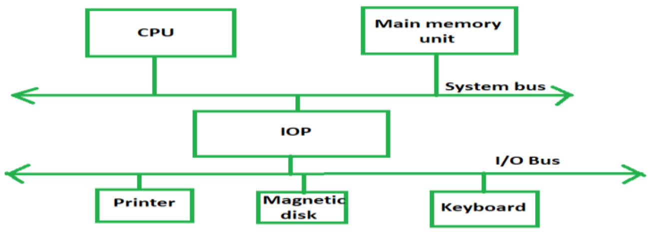
1. **Programmed I/O:** It is due to the result of the I/O instructions that are written in the computer program. Each data item transfer is initiated by an instruction in the program. Usually the transfer is from a CPU register and memory. In this case it requires constant monitoring by the CPU of the peripheral devices.

2. **Interrupt- initiated I/O:** Since in the above case we saw the CPU is kept busy unnecessarily. This situation can very well be avoided by using an interrupt driven method for data transfer. By using interrupt facility and special commands to inform the interface to issue an interrupt request signal whenever data is available from any device. In the meantime the CPU can proceed for any other program execution. The interface meanwhile keeps monitoring the device. Whenever it is determined that the device is ready for data transfer it initiates an interrupt request signal to the computer.
3. **Direct Memory Access**: The data transfer between a fast storage media such as magnetic disk and memory unit is limited by the speed of the CPU. Thus we can allow the peripherals directly communicate with each other using the memory buses, removing the intervention of the CPU. This type of data transfer technique is known as DMA or direct memory access. During DMA the CPU is idle and it has no control over the memory buses. The DMA controller takes over the buses to manage the transfer directly between the I/O devices and the memory unit.

6.4 I/O Processor:

The **DMA mode** of data transfer reduces CPU's overhead in handling I/O operations. It also allows parallelism in CPU and I/O operations. Such parallelism is necessary to avoid wastage of valuable CPU time while handling I/O devices whose speeds are much slower as compared to CPU. The concept of DMA operation can be extended to relieve the CPU further from getting involved with the execution of I/O operations. This gives rises to the development of special purpose processor called **Input-Output Processor (IOP) or IO channel**.
The Input Output Processor (IOP) is just like a CPU that handles the details of I/O operations. It is more equipped with facilities than those are available in typical DMA controller. The IOP can fetch and execute its own instructions that are specifically designed to characterize I/O transfers. In addition to the I/O – related tasks, it can perform other processing tasks like

arithmetic, logic, branching and code translation. The main memory unit takes the pivotal role. It communicates with processor by the means of DMA.



Input-Output Processor

The Input Output Processor is a specialized processor which loads and stores data into memory along with the execution of I/O instructions. It acts as an interface between system and devices. It involves a sequence of events to executing I/O operations and then store the results into the memory.

Advantages –

- The I/O devices can directly access the main memory without the intervention by the processor in I/O processor based systems.
- It is used to address the problems that are arises in Direct memory access method.

6.5 I/O Ports (Serial, Parallel, HDMI, VGA, USB 2.0 and 3.0):

A connection point that acts as interface between the computer and external devices like mouse, printer, modem, etc. is called **port**. Ports are of two types −

- **Internal port** − It connects the motherboard to internal devices like hard disk drive, CD drive, internal modem, etc.
- **External port** − It connects the motherboard to external devices like modem, mouse, printer, flash drives, etc.

a)Serial Port:

A serial port is basically a serial communication interface through which information transforms one bit at a time. It is one of the oldest type of interfaces.

- These are basically used for external modems.
- These are basically available in two versions in market these are 9 pins, 25 pin models.
- Data travels at a speed of 115 kilo-bits per second.

Serial ports transmit data sequentially one bit at a time. So they need only one wire to transmit 8 bits. However it also makes them slower. Serial ports are usually 9-pin or 25-pin male connectors. They are also known as COM (communication) ports or RS323C ports.

b) Parallel Port:

A parallel port is basically a parallel communication interface through which information transforms multiple bits at a time.

- These are basically used to connect peripherals such as scanners or printers.
- These are also known as printer ports.
- These are available in a 25 pin model.
- Data travels at a speed of 150 kilo bits per second.

Parallel ports can send or receive 8 bits or 1 byte at a time. Parallel ports come in form of 25-pin female pins and are used to connect printer, scanner, external hard disk drive, etc.

c) USB Port:

It is basically a standard cable connection interface between computer and external device. USB is an industrial standard for short-distance digital data communication.

- Basically it can connect all types of external devices to the computer such as mouse, keyboard, printers, speakers etc.
- These ports were introduced in 1997.
- Minimum 2 ports are there in every computer system.
- Data basically travels at a speed of 14mb/s which is much faster than serial port.
- The devices that uses USB port gets power from a USB port.

USB stands for Universal Serial Bus. It is the industry standard for short distance digital data connection. USB port is a standardized port to connect a variety of devices like printer, camera, keyboard, speaker, etc.

d) HDMI: Short for High Definition Multimedia Interface, HDMI is a connector and cable capable of transmitting high-quality and high-bandwidth streams of audio and video between devices. The HDMI technology is used with devices such as an HDTV, Projector, DVD player, or Blu-ray player. The picture is an example of an HDMI cable from Mediabridge.

e) VGA: VGA connector stands for Video Graphic Array connector, these are basically 15 pin connector available in many video-cards, computer, projectors etc.

- It is used to connect monitor to computer's video card.
- It is 15 pin connector.
- These were introduced by IBM in 1987.
- VGA basically utilizes analog signal hence it can only be used to lower resolution or we can say VGA is only capable of lowering the resolution.

These are some of the common ports available in computer system. Except these there are many more ports available in computer. These are as follows:

1. **Modem Port:**
   These are basically used to connect PC's modem to telephone networks.
2. **Ethernet Port:**
   These are basically used to connect Ethernet cables to the computer. In this data may travel with a speed of 10mb/s to 100 mb/s based on the network bandwidth.

f**) USB 2.0**: It is also known as **hi-speed USB**. It was introduced in 2000. It is an updated version of USB 1.1, which provides improved functionalities and better speed. It is capable to deliver the maximum transfer speed of 480 Megabits per second. However, practically it is approximately 280 Mbps.

g) USB 3.0: It is also known as **SuperSpeed USB** and was first made available in November 2009. It is a much-improved version of USB 2.0. It supports the data transfer rate of 5 Gigabits per second, which is much much faster than the speed provided by USB 2.0.

S.NO	USB 2.0	USB 3.0
1.	Data transmission rate of USB 2.0 is 480 Mbps.	While data transmission rate of USB 3.0 is 4800 Mbps.
2.	USB 2.0 has four number of wires.	While USB 3.0 has nine number of wires.
3.	USB 2.0 is a one-way communication.	While USB 3.0 is a two-way communication.
4.	USB 2.0 is a less costly.	While It is costlier than USB 2.0.
5.	USB 2.0 cable length can be up to five meters.	While USB 3.0 cable length can be up to three meters.
6.	USB 2.0 delivers 500 mA amount of power.	While it delivers 900 mA amount of power.
7.	Black block occurs within USB 2.0.	Whereas blue block occurs within USB 3.0.