Microprocessors

Introduction to Microprocessors

A microprocessor is a computer processor which incorporates the functions of a computer's central processing unit (CPU) on a single Integrated Circuit (IC), or at most a few Integrated Circuits. The microprocessor is a multipurpose, clock-driven, register-based, digital-integrated circuit which accepts binary data as input, processes it according to instructions stored in its memory, and provides results as output. Microprocessors contain both combinational logic and sequential digital logic. Microprocessors operate on numbers and symbols represented in the binary numeral system.

A microprocessor is a computer processor where the data processing logic and control is included on a single integrated circuit (IC), or a small number of ICs. The microprocessor contains the arithmetic, logic, and control circuitry required to perform the functions of a computer's central processing unit (CPU). The IC is capable of interpreting and executing program instructions and performing arithmetic operations. The microprocessor is a multipurpose, clock-driven, register-based, digital integrated circuit that accepts binary data as input, processes it according to instructions stored in its memory, and provides results (also in binary form) as output. Microprocessors contain both combinational logic and sequential digital logic and operate on numbers and symbols represented in the binary number system.

Features of Microprocessor or Why Microprocessor?

Some of the features of any microprocessor are-

Cost effective: The microprocessor chips are available at low prices and results its low cost.

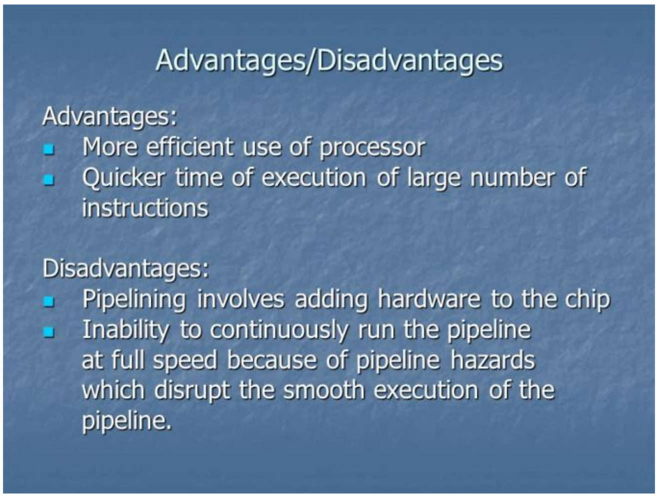
Size: The microprocessor is of small size chip; hence it is portable.

Low power consumption: Microprocessors are manufactured by using metal oxide semiconductor technology, which has low power consumption.

Versatility: The microprocessors are versatile as the same chips in number of applications can be used by configuring the
software program.

Reality: The failure rate of an IC in microprocessors is very low, hence it is reliable.

Disadvantages of more number of pipeline phases



Advantages:

1. **Improved Performance:** A deeper pipeline can result in better performance by allowing multiple instructions to be in various stages of execution simultaneously, leading to higher throughput.
2. **Higher Clock Frequencies:** Deeper pipelines can potentially allow for higher clock frequencies, as the work is divided into smaller, more manageable stages. This can result in faster execution of instructions.
3. **Better Resource Utilization:** With more stages, resources like execution units and registers can be better utilized, reducing idle time and improving overall efficiency.
4. **Reduced Cycle Time:** Deeper pipelines can reduce the cycle time, allowing instructions to be processed more quickly.
5. **Simpler Control Logic:** In some cases, deeper pipelines can lead to simpler control logic as instructions are broken down into smaller, more manageable stages.

Disadvantages:

1. **Increased Complexity:** A higher number of pipeline stages can lead to increased complexity in the design, making it more challenging to implement, debug, and maintain.
2. **Higher Latency:** Deeper pipelines introduce more stages, each adding a certain amount of latency. While throughput may increase, individual instructions may take longer to complete, which can be problematic for latency-sensitive tasks.
3. **Branch Misprediction Penalties:** Deeper pipelines are more susceptible to branch mispredictions, as they can result in a more significant number of instructions needing to be flushed from the pipeline when a misprediction occurs. This can negatively impact performance.
4. **Resource Contention:** More pipeline stages can lead to increased contention for critical resources like registers, cache, and execution units. This contention can cause inefficiencies and bottlenecks.
5. **Power Consumption:** Deeper pipelines often consume more power due to the need for additional latches, buffers, and logic. This can be a concern in mobile and battery-powered devices.
6. **Complex Hazard Handling:** With more pipeline stages, the processor needs more complex hazard detection and resolution mechanisms to deal with data hazards, control hazards, and structural hazards, increasing design complexity.
7. **Diminishing Returns:** Beyond a certain point, adding more pipeline stages may not yield significant performance improvements, and it can lead to diminishing returns, along with increased design complexity.
8. **Code Size Impact:** In some cases, deeper pipelines can result in larger code sizes due to the need for additional pipeline-related instructions, affecting memory requirements and cache performance.

Difference between pipelining and VLIW

Illustrate Pipelining

What is a 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.

Concept of Hardware and Software

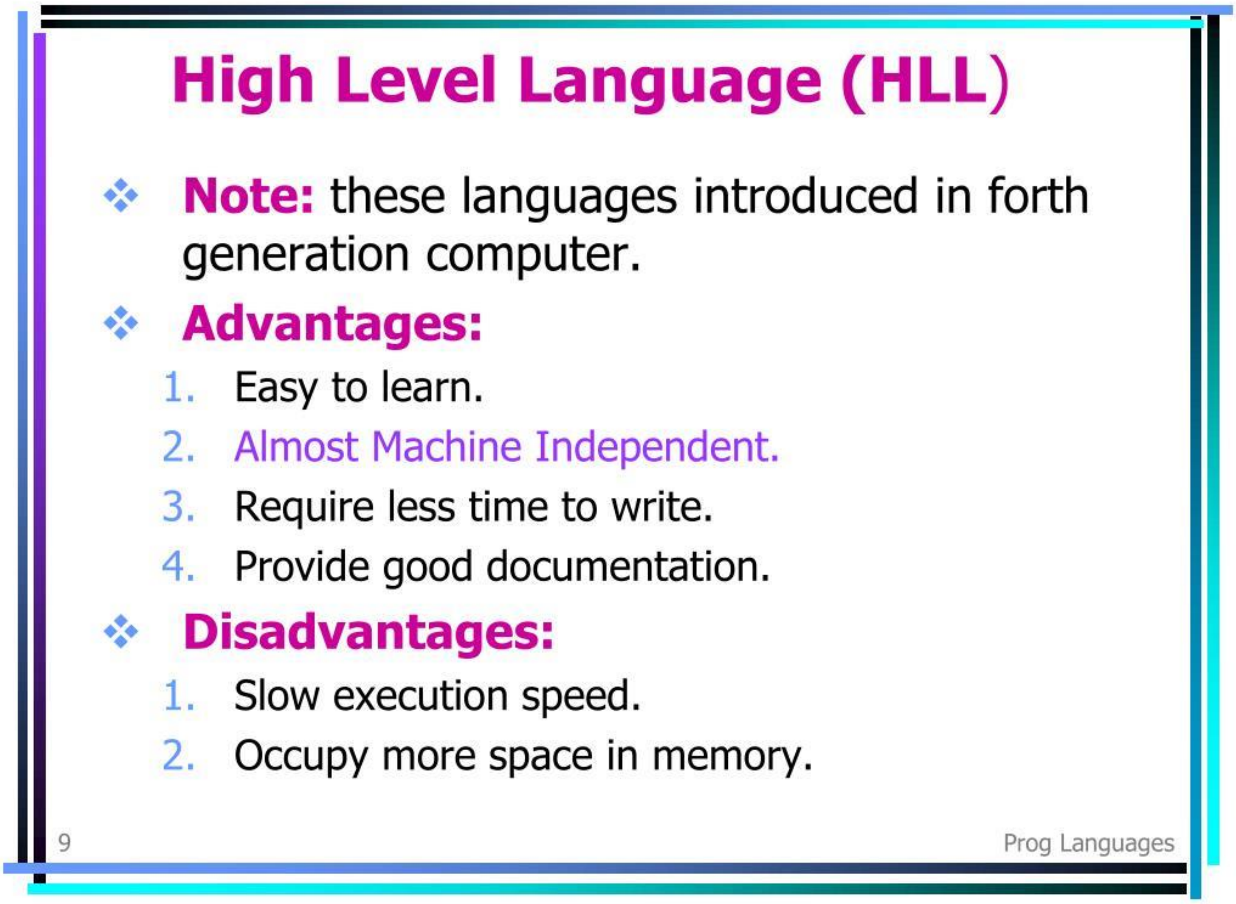
The term hardware refers to a mechanical device that makes up a computer. Software can be categorized into two types software and application software

Programming Languages

The languages that are used to write a program or set of instructions are called "Programming languages". Programming languages are broadly categorized into three types:

- High-level language
- Assembly-level language
- Machine-level or low level language

Advantages & Disadvantages of High-level language



Q1. What are some common applications of high-level programming languages?
Ans: High-level programming languages are used for a wide variety of applications, including web development, scientific computing, data analysis, artificial intelligence, and video game development.

Q2. How do high-level programming languages compare to low-level programming languages?
Ans: High-level programming languages are easier to learn and use than low-level languages, but they generally have reduced performance and increased memory usage compared to low-level languages. Low-level languages provide more control over hardware and can be more efficient in certain applications.

Q3. What resources are available for learning high-level programming languages?
Ans: There are many resources available for learning high-level programming languages, including online courses, tutorials, documentation, and online communities such as forums and chat groups.

Q4. Can high-level programming languages be used for system programming?
Ans: High-level programming languages can be used for system programming, but they may not be the best choice for low-level tasks such as writing device drivers or operating

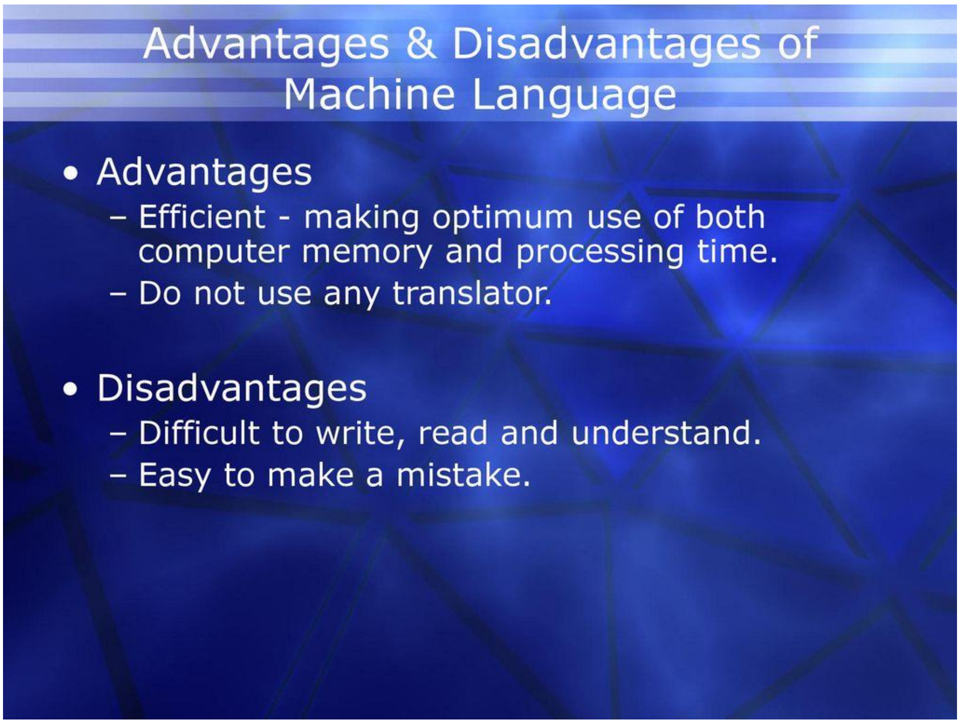
system kernels. However, some high-level languages, such as C and C++, are commonly used for system programming.

Q5. What is the difference between a high-level language and a low-level language?
Ans: High-level languages are designed to be easier to read and write, with a syntax that is closer to human language. They generally provide more abstraction and automation, allowing programmers to focus on solving problems rather than low-level details. Low-level languages, on the other hand, are designed to be closer to machine language, providing more control over hardware and allowing for greater performance and efficiency.

Advantages & Disadvantages of Assembly-level language

Advantages	Disadvantages
Assembly language is easier to understand and use as compared to **machine language**.	Like **machine language**, it is also **machine** dependent/specific.
It is easy to locate and correct errors.	Since it is **machine** dependent, the programmer also needs to understand the hardware.

Advantages & Disadvantages of Machine-level or low-level language



What is Moore's Law?

Moore's Law is an observation that the number of transistors in a computer chip doubles every two years or so. As the number of transistors increases, so does processing power. The law also states that, as the number of transistors increases, the cost per transistor falls.

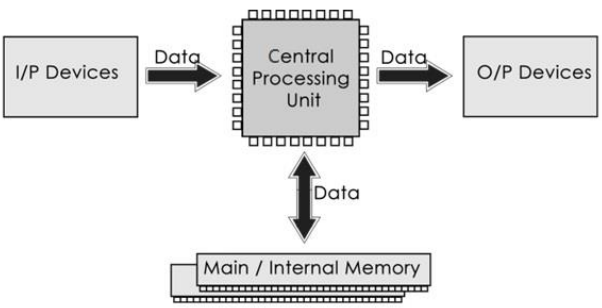
Who invented 1st computer?

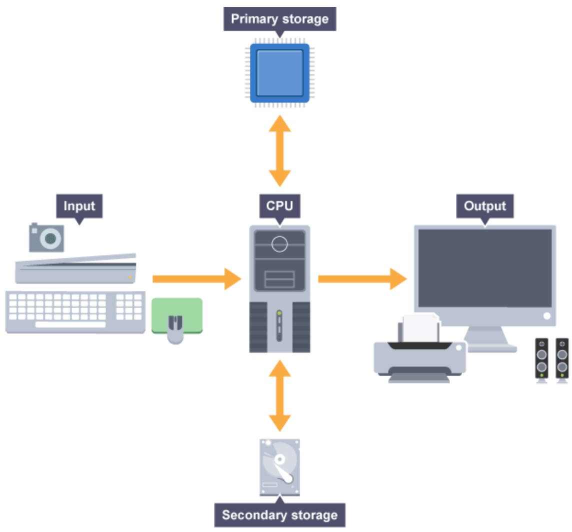
Charles Babbage

The first computer was invented by Charles Babbage (1822) but was not built until 1991! Alan Turing invented computer science. The ENIAC (1945) was the first electronic general-purpose digital computer; it filled a room. The Micral N was the world's first "personal computer" (1973).

Components of Computer System

Computer systems consist of four components as shown in the below image: **Central Processing Unit (CPU), Input devices** & **Output devices** and **Memory**. All general-purpose computers follow the same basic model. This diagram illustrates the flow of data within a computer.





Input Devices: Input is received from an input device such as a keyboard, mouse, digital pen, etc. Input can also be fed through devices like CD-ROM, pen drive, scanner, etc.

Central Processing Unit (CPU)
The Central Processing Unit (CPU) is called "the brain of the computer" as it controls the operation of all parts of the computer. The CPU receives instructions and data from an input or memory. The instructions and data are processed by the CPU and the results are either sent to an output or transferred to secondary storage. Data is held in primary storage while it is being processed.

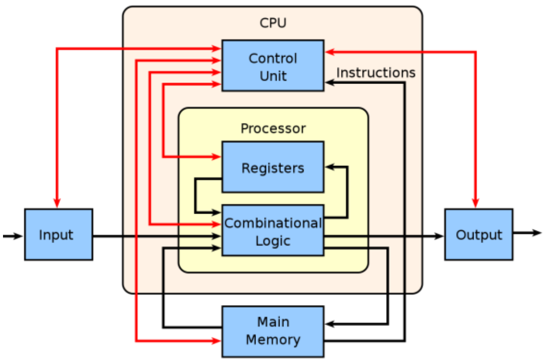
Processing the Information: Operations on the input data are carried out based on the instructions provided in the programs.

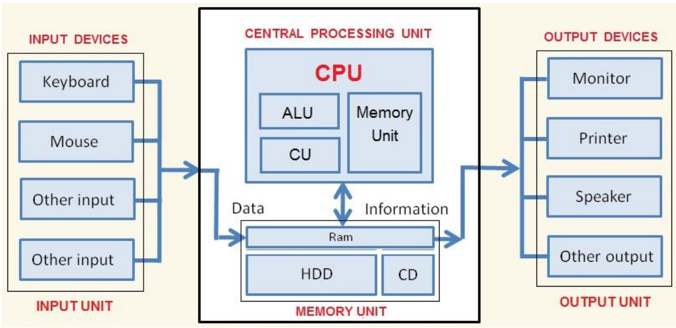
Memory: After processing, the information gets stored in the primary or secondary storage area. This is stored in the computer's memory. The unit of memory is "Byte".
1 Byte = 8 Bits

Output Devices: The processed information and other details are communicated to the outside world through output devices like monitors, printers and speakers etc.

The purpose of the CPU
The central processing unit (CPU) is the most important hardware component in a computer. It has two main functions:
- To process data and instructions
- To control the rest of the computer system

Common CPU components
The central processing unit (CPU) is the brain of a computer. The central processing unit (CPU) consists of six main components. All the components work together to carry out the instructions and system control. The main components of a CPU are:
- Arithmetic Logic Unit (ALU)
- Control Unit (CU)
- Registers
- Cache Memory
- Buses
- Clock





Arithmetic Logic Unit (ALU)
Data entered into the computer is sent to RAM, from where it is then sent to ALU, where the rest of the data processing takes place. All types of processing, such as comparisons, decision-making and processing of non-numeric information take place here and once again data is moved to RAM. The ALU has two main functions:
- It performs arithmetic and logical operations (Decisions). This includes addition, subtraction, multiplication, division, and logical operations such as AND, OR, and NOT.
- It acts as a gateway between primary storage and secondary storage - data transferred between them passes through the ALU.

Control Unit (CU)
The CU controls the flow of instructions through the CPU. It determines which instruction to execute next, and it also coordinates the activities of the other components of the CPU. The control unit performs the following functions:

- It controls all activities of the computer
- Transfers data and instruction to the Arithmetic and Logic Unit
- Transfers results to memory
- Fetches results from memory to output devices

Registers
Registers are a type of small and high-speed computer memory built directly into the processor or CPU. It is used to store and manipulate data during the execution of instructions. A register may hold an instruction, a storage address, or any kind of data (such as a bit sequence or individual characters).

A register is composed of multiple flip-flops, which are electronic circuits capable of storing a single bit of information, which is represented through data - either a 0 or a 1. By combining multiple flip-flops, registers can store larger binary values, such as bytes or words.

Registers also contain control logic circuitry, which allows it to coordinate the flow of data and instructions within the CPU. This can include operations such as decoding control signals, performing data manipulation like data loading, storing, or arithmetic operations, and using multiplexers to route data to a specific location within the register.

Together, flip-flops and control logic work in partnership within registers. Flip-flops provide the storage capacity, while control logic facilitates the coordination of data transfer, manipulation, and synchronization with other components of the CPU. This enables registers to store and process data efficiently during the execution of instructions.

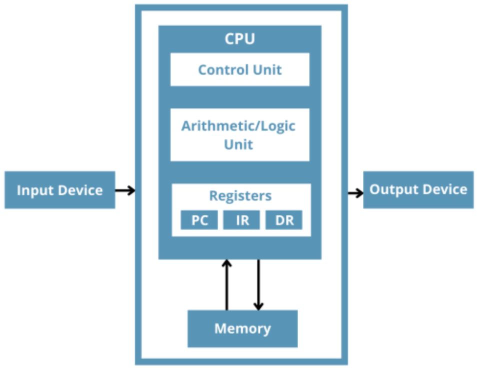
Sizes of CPU Registers
The number and size of registers in a CPU are determined by the processor design and can have a significant impact on its performance and capabilities. Most modern computer processers include:

- **8-bit registers:** These registers can store 8 bits of data (1 byte). They are often used for basic arithmetic operations and data manipulation.
- **16-bit registers:** These registers can store 16 bits of data (2 bytes). They are commonly found in older processors or in specific architectures that require 16-bit operations.
- **32-bit registers:** These registers can store 32 bits of data (4 bytes). They are widely used in many processors and are capable of handling larger data sizes and more complex calculations.
- **64-bit registers:** These registers can store 64 bits of data (8 bytes). They are prevalent in modern processors and offer increased computational power and memory-addressing capabilities.

Modern PCs today most often have 32-bit or 64-bit registers and are referred to as the 32-bit processors and 64-bit processors we often hear about. This indicates the size or width of the processor's registers and the amount of data the processor can handle in a single operation.

In some specialized processors or architectures, you may also find larger register sizes, such as 128-bit, 256-bit, or even larger registers. These larger registers are often used for specific purposes like vector processing or cryptographic operations, where parallelism and large data sets are involved.

How Do Registers Operate with Other CPU Components?
CPUs are made up of various components that when used together, allow it to process data and perform calculations. The major components include the Arithmetic Logic Unit (ALU), Control Unit (CU), Registers, Clock, Cache, and Buses.



The ALU is a fundamental component of the CPU responsible for performing arithmetic and logical operations. It can execute operations like addition, subtraction, AND, OR, and others. The ALU takes input from registers, performs the desired operation, and stores the result back into a register.
The CU directs and coordinates the operations of various components within the CPU. It interprets instructions and generates control signals to manage the flow of data between registers, the ALU, memory, and input/output devices.
The interaction between registers, ALU, and CU can be summarized in the following steps:

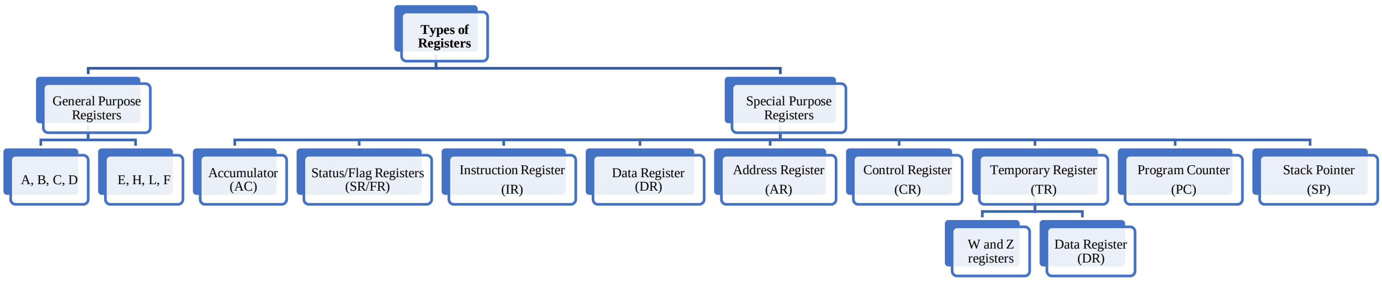
- The CU fetches an instruction from memory and places it into the instruction register.
- The CU decodes the instruction to determine the operation to be performed and identifies the registers involved.
- The CU issues control signals to select the appropriate registers and routes the data to the ALU.
- The ALU performs the arithmetic or logical operation on the data from the selected registers.
- The result of the operation is stored back into a register, based on the CU's control signals.

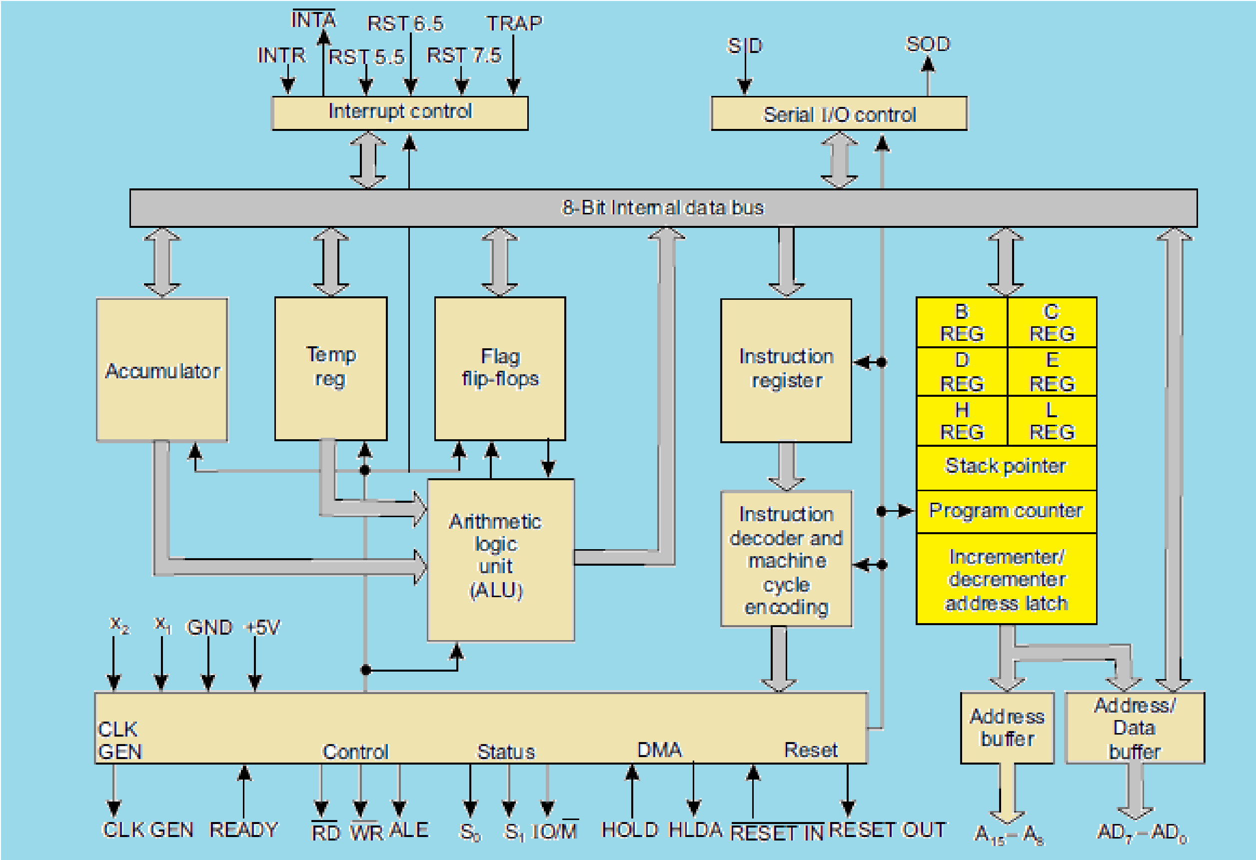
What is the Purpose of Registers?

Registers are used by computers for various purposes, including storing program instructions before they're executed or holding intermediate results from calculations so that their values can be retrieved later on if needed. They also help speed up processes by allowing processors to access frequently used values without having to retrieve them from main memory every time they need them.

Types of CPU Registers

Depending on the CPU's architecture and design, the type and number of registers for different purposes. It has eight addressable 8-bit registers: A, B, C, D, E, H, L, F, and two 16-bit registers. These registers can be classified as:





Architecture of 8085

Following is the list of some of the most common registers in computers:

Register	Symbol	Number of Bits	Function
Accumulator	AC	16	It's a processor register.
Program counter	PC	12	It stores the address of the instruction.
Address Register	AR	12	It is used for storing memory addresses.
Data Register	DR	16	It is a general-purpose register used for storing data during calculations.
Instruction Register	IR	16	It stores the current instruction being executed.
Temporary Register	TR	16	It holds the temporary data.
Input Register	INPR	8	It carries the input character.
Output Register	OUTR	8	It carries the output character

General Purpose Registers

Registers B, C, D, E, H, and L are general-purpose registers in the 8085 Microprocessor. All these GPRS are 8-bits wide. They are less important than the accumulator. They are used to store data temporarily during the execution of the program.

For example, there is no instruction to add the contents of Band E registers. At least one of the operands has to be in A. Thus, to add Band E registers, and to store the result in B register, the following have to be done.

- Move to A register the contents of B register.
- Then add A and E registers. The result will be in A.
- Move this result from the A register to the B register.

It is possible to use these registers as pairs to store 16-bit information. Only BC, DE, and HL can form register pairs. When they are used as register pairs in an instruction, the left register is understood to have the MS byte and the right register the LS byte.

For example, in the DE register pair, the content of the D register is treated as the MS byte, and the content of the E register is treated as the LS byte.

Special Purpose Registers

Accumulator Register(AC): The Accumulator is an 8-bit general-purpose register used for arithmetic and logical operations. It stores intermediate results during calculations. If no specific address for the result operation is specified, the result of arithmetic operations is transferred to AC. The number of bits in the accumulator register equals the number of bits per word.

Status Register/Flags Register (SR/FR): The Status Register or Flags Register contains individual bits that indicate the outcome of operations. These flags help in making decisions and controlling program flow based on the results of previous operations.

It is a 3-bit register, in which five of the bits carry significant information in the form of flags: S (Sign flag), Z (Zero flag), AC (Auxiliary carry flag), P (Parity flag), and CY (carry flag); as shown in Fig.

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	X	AC	X	P	X	CY

Flag Register of 8085

- **S-Sign flag -** After the execution of arithmetic or logical operations, if bit D7 of the result is 1, the sign flag is set. In a given byte if D7 is1, the number will be viewed as a negative number. If D7 is U, the number will be considered as a positive number.
- **Z-Zero flag -**The zero flag sets if the result of the operation in ALU is zero and flag resets if the result is non zero. The zero flags are also set if a certain register content becomes zero following an increment or decrement operation of that register.
- **AC-auxiliary Carry flag -** This flag is set if there is an overflow out of bit 3 i.e., carry from lower nibble to higher nibble (D3 bit to D4 bit). This flag is used for BCD operations and it is not available for the programmer.
- **P-Parity flag** - Parity is defined by the number of ones present in the accumulator. After arithmetic or logical operation, if the result has an even number of ones, i.e., Even parity, the flag is set. If the parity is odd, the flag is reset.
- **CY-Carry flag -** This flag is set if there is an overflow out of bit 7. The carry flag also serves as a borrow flag for subtraction. In both the examples shown below, the carry flag is set.

Addition

```
      9BH    ----->   1001 1011
     +75 H   -----> + 0111 0101
    ---------       -----------
Carry   1 10 H  -----> 1 0001 0000
```

Subtraction

```
      89H    ----->   1000 1001
     -AB H   -----> - 1010 1011
    ---------       -----------
```

Borrow1 10 H -----> 1 1101 1110

Data Register (DR): Data Registers store data fetched from memory or obtained from input/output operations. When a direct or indirect addressing operand is found, it is placed in the Data Register. This value was then used as data by the processor during its operation. It's about the same size as a word in memory

Address Registers (AR): Address Registers store memory addresses for data access or for transferring data between different memory locations. The size of the Address Register is equal to the width of the memory address is directly related to the size of the memory because it contains an address. If the memory has a size of 2n*m, then the address is specified using n bits.

Control Registers (CR): Control Registers manage various control settings and parameters related to the CPU's operation, such as interrupt handling, memory management, and system configuration.

Temporary Register (TR): The Temporary Register is used to hold data while it is being processed. As Temporary Register stores data, the number of bits it contains is the same as the number of bits in the data word.

Temporary Data Register - The ALU has two inputs. One input is supplied by the accumulator and other from the temporary data register. The programmer cannot access this temporary data register. However, it is internally used for execution of most of the arithmetic and logical instructions.

W and Z registers - Wand Z registers are temporary registers. These registers are used to hold 8-bit data during the execution of some instructions. These registers are not available for the programmer since 8085Microprocessor Architecture uses them internally.
Special Purpose Registers

Instruction Register (IR): The instruction is stored in the Instruction Register. The instruction register contains the currently executed instruction. Because it includes instructions, the number of bits in the Instruction Register equals the number of bits in the instruction, which is n bits for an n-bit CPU

Program counter (PC): Program is a sequence of instructions. As mentioned earlier, microprocessor fetches these instructions from the memory and executes them the program counter is a special purpose register which, at a given time, stores the address of the next instruction to be fetched. Program Counter acts as a pointer to the next instruction. How processor increments program counter depends on the nature of the instruction; for one-byte instruction it increments program counter by one, for two-byte instruction it increments program counter by two and for three-byte instruction it increments program counter by three such that program counter always points to the address of the next instruction.

The size of the PC is equal to the width of the memory address, and the number of bits in the PC is equal to the number of bits in the PC.

In case of JUMP and CALL instructions, address followed by JUMP and CALL instructions is placed in the program counter. The processor then fetches the next instruction from the new address specified by JUMP or CALL instruction. In conditional JUMP and conditional CALL instructions, if the condition is not satisfied, the processor increments program counter by three so that it points the instruction followed by conditional JUMP or CALL instruction; otherwise, processor fetches the next instruction from the new address specified by JUMP or CALL instruction is stored.

Stack Pointer (SP): The stack is a reserved area of the memory in the RAM where temporary information may be stored. A 16-bit stack pointer is used to hold the address of the most recent stack entry.
In the following figure depicts the architecture of 8085 microprocessor and with its register organization –

Stack Pointer (SP): The Stack Pointer points to the top of the stack, which is a region of memory used for temporary storage during function calls and other operations.

Input Register (INPR): Input Register is a register that stores the data from an input device. The computer's alphanumeric code determines the size of the input register.

Output Register (OUTR): The data that needs to be sent to an output device is stored in the Output Register. Its size is determined by the alphanumeric code used by the computer

What are Computer Registers?
Computer registers are memory-storing units that operate at high speed. It's a component of a computer's processor. It can hold any type of data, including a bit sequence or a single piece of data.

What is the purpose of a computer register?
Registers are a type of computer memory used to accept, store, and transfer data and instructions used by the CPU right away.

What are the 3 types of registers?
Accumulator: It is a processor register. The symbol for the accumulator is AC.
Address register: It stores an address for memory. The symbol for the address register is AR.
Program Counter: It stores the address of the instruction. The symbol for the Program counter is PC.

What is a register and its function?
Computer Registers are a type of computer memory used to quickly accept, store, and transfer data and instructions that the CPU needs immediately. The registers used by the CPU are called processor registers. The function of a register is to store, accept, transfer, and manipulate the data.

What are the benefits of registers?
Some of the basic benefits of registers are reduced bus traffic, and speed, reduced code size, and increased flexibility. Registers can help a program run more quickly by minimizing the number of times data must be moved between the CPU and main memory.

What is Computer Memory?
Computer memory is just like the human brain. It is used to store data/information and instructions. Computer memory is a data storage unit/space or a data storage device where data is to be processed and instructions required for processing are stored. It can store both the input and output can be stored here.

The memory is divided into a 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, this memory unit has 64*1024 = 65536 memory locations. The address of these locations varies from 0 to 65535.

Characteristics of Computer Memory
- It is faster computer memory as compared to secondary memory.
- It is semiconductor memories.
- It is usually a volatile memory, and the main memory of the computer.
- A computer system cannot run without primary memory.

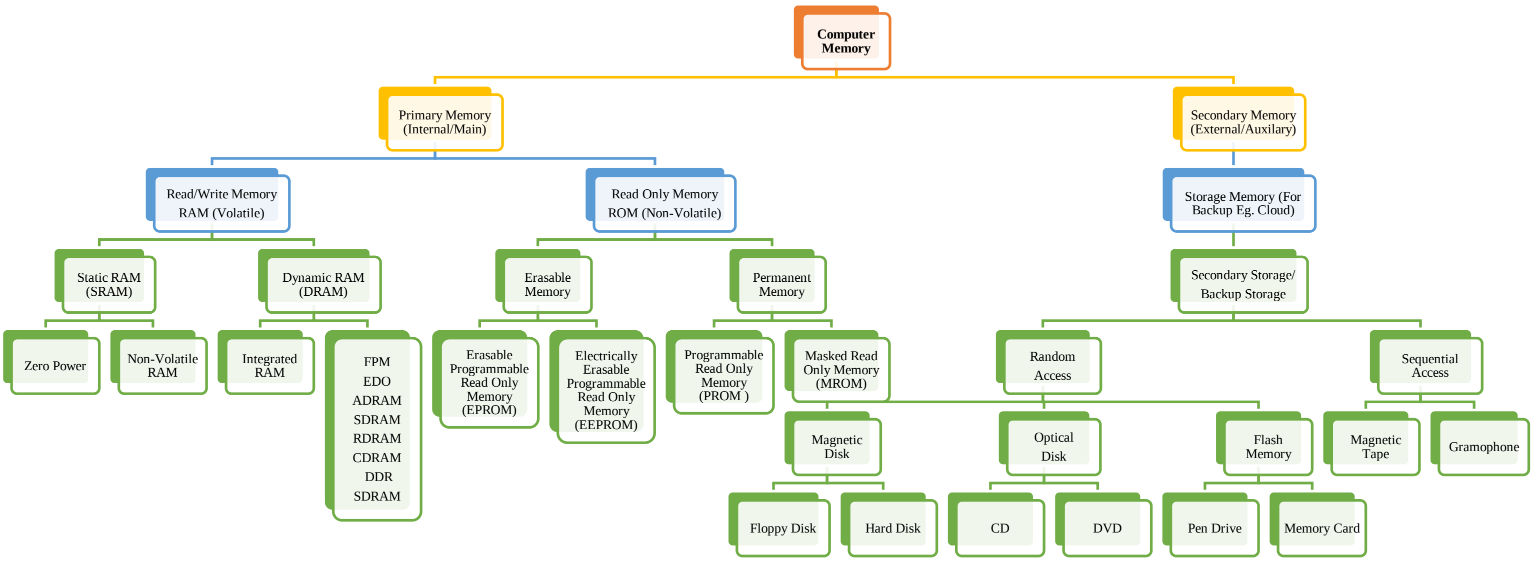
How Does Computer Memory Work?
When you open a program, it is loaded from secondary memory into primary memory. Because there are various types of memory and storage, an example would be moving a program from a Solid-State Drive (SSD) to RAM. Because primary storage is accessed more quickly, the opened software can connect with the computer’s processor more quickly. The primary memory is readily accessible from temporary memory slots or other storage sites.

Memory is volatile, which means that data is only kept temporarily in memory. Data saved in volatile memory is automatically destroyed when a computing device is turned off. When you save a file, it is sent to secondary memory for storage.

There are various kinds of memory accessible. Its operation will depend upon the type of primary memory used. but normally, semiconductor-based memory is more related to memory. Semiconductor memory is made up of IC (integrated circuits) with silicon-based metal-oxide-semiconductor (MOS) transistors.

Types of Memory in Computer

Memory is a very important part of a computer system because without memory a computer could not perform simple tasks. There are two types of memory in a computer: Primary memory (RAM and ROM) and secondary memory (Hard Drive, CD, etc.). Random Access Memory (RAM) is the main variable memory and Read-Only Memory (ROM) is the main fixed memory and glimpse so the same can be seen in the image given below:



Primary Memory

The memory that is an in-built part of a computer is considered to be the main memory of the computer and is also known as internal memory or internal storage. Primary memory is faster than secondary memory when it comes to accessing data either for volatile or non-volatile memory.

Primary memory holds only those data and instructions on which the computer is currently working. It has a limited capacity when comparing it with secondary memory and data is lost when power is switched off. It is generally made up of semiconductor devices. These memories are not as fast as registers. It is divided into two subcategories RAM and ROM.

Characteristics of Primary Memory

- A computer’s primary memory is also known as the main memory, a temporary memory, or a prime memory.
- It is considered volatile memory.
- Semiconductor technology is used in the construction of this memory
- Data is lost in case power is switched off.

- It is the working memory of the computer.
- A computer cannot run without the primary memory.
- The processing speed of secondary memory is slower than primary memory

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.

Random Access Memory (RAM)

Random Access Memory (RAM) is the internal memory of the CPU for storing data, programs and program results. It is a read/write memory that stores data until the machine is working. As soon as the machine is switched off, data is erased.

RAM is volatile, i.e., data stored in it is lost when we switch off the computer or if there is a power failure. Hence, a backup Uninterruptible Power System (UPS) is often used with computers. RAM is small, both in terms of its physical size and in the amount of data it can hold. RAM is divided into two subcategories Static RAM (SRAM) and Dynamic RAM (DRAM).

Characteristics of RAM

- Writing and erasing both types of tasks can be performed by RAM memory.
- Adding up RAM will help boost the computer system's speed and performance.
- RAM memory permits the central processing unit to access data quickly so that execution by the system can be done faster.
- The cost of RAM is much less than the cost of a Solid-State Drive commonly known as SSD, but if we carry out the comparison between the two, then RAM will execute any instruction more quickly than SSD.

Advantages of RAM

- RAM has the capability to increase the speed of the system, and the higher the RAM, the greater the speed.
- RAM reads data slower than the central processing unit.
- It consumes very less power
- It has the ability to write and delete programs.

Disadvantages of RAM

- If the CPU wants to read the data from the RAM, then the data access from the registers and the cache is slow in comparison
- RAM is volatile, i.e., it is difficult to store data forever.
- It is expensive.
- It has limited space.

Static RAM (SRAM)

SRAM falls under the category of semiconductor memory and bistable latching circuitry is being used to store each bit, which makes it very fast and thus named cache memory. However, it is more expensive than DRAM and it takes up a lot more space which leads to less memory on a chip.

However, data is lost when the power goes down due to its volatile nature. SRAM chips use a matrix of 6 transistors and no capacitors. Transistors do not require power to prevent leakage, so SRAM need not be refreshed on a regular basis.

It is also widely used as a cache in a CPU and is usually installed in L2 or L3. But as discussed earlier, because it is very expensive, the price of L2 and L3 is generally in the range of 1MB to 16MB only.

Characteristics of Static RAM

- It also has a memory cell, like all memory components, it consists of 6 transistors.
- It is used as a CPU cache memory.
- Access time is lower, which makes it faster than DRAM.
- SRAM requires a continuous power supply to store information and thus uses additional power (High power consumption).
- Contains flip-flop circuitry to store pieces of information.
- Expensive
- Long life

Advantages of SRAM

- It is preferred because of its access speed, despite being very expensive.
- During the speed-sensitive cache it is very helpful.
- SRAM is simple and easy to manage.
- It is very reliable and therefore is used for cache memory.

Disadvantages of SRAM

- Its speed does not justify its price.
- It is a volatile memory, so all data is lost when power is cut off.
- It also has a small storage capacity and takes up a lot of space.
- The design is complex and not easy to build or understand.

Types of Static RAM

SRAM can be further classified into the following types:

- Non-volatile SRAM
- Pseudo SRAM
- Asynchronous
- Synchronous

Dynamic Random Access Memory (DRAM)
DRAM is also a different type of RAM in which every piece of data is stored within an integrated circuit in a separate capacitor. In dynamic cell holding one bit of data and it is made up of transistors and capacitors. Data is lost due to self-discharge. So, we have to give periodically charge to it. That's why it is called dynamics. In this memory controller is entitled to read the data and rewrite it afterwards, refreshing it constantly. This is a lengthy process which makes DRAM slower than SRAM.

However, DRAM is more cost-effective than SRAM, which makes it usable as the main memory in the CPU, although it is slower than SRAM, it is still fast considering its price and ability to connect directly to the CPU bus. Unlike expensive SRAM, DRAM is typically 4GB to 16 GB for laptops and 1GB to 2GB for smaller devices.

Characteristics of Dynamic RAM

- Short data lifetime
- Needs to be refreshed continuously
- Slower as compared to SRAM
- Used as RAM
- Smaller in size
- Less expensive
- Less power consumption

Advantages of DRAM

- DRAM has a simple design, as it consists of only a single transistor.
- It has high congestion rates.
- It is less costly than SRAM.
- Memory space is large.

Disadvantages of DRAM

- This memory is volatile, as it is continuously refreshing.
- It has a complex manufacturing process.
- Refreshing is required Continuously.
- It is slower than SRAM.

Difference Between SRAM and DRAM

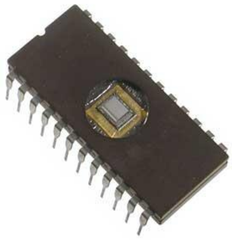
Static RAM	Dynamic RAM
The construction of its circuits is similar to D flip-flops	Tiny capacitors are used in its construction that leak electricity
It holds its information as long as power is available	It needs a recharge after a few milliseconds to maintain its data
Faster than DRAM	Slower than SRAM
More costly	Less costly
Power Hungry	Power-efficient
Generates more heat.	Generates less heat
Used as cache memory	Used as the main memory

Read Only Memory (ROM)

The memory from which we can only read but cannot write on it. This type of memory is non-volatile. The information is stored permanently in such memories during manufacture. A ROM stores such instructions that are required to start a computer. This operation is referred to as **bootstrap**. ROM chips are not only used in the computer but also in other electronic items like washing machines and microwave ovens. It is used in embedded systems or where the programming needs no change. It is also used in calculators and peripheral devices. There are further four types of ROM:

- EPROM

- EEPROM
- PROM
- MROM



Characteristics of ROM

- It consumes less power.
- It is less expensive than RAM.
- ROM has a simple interface in comparison with RAM.
- ROM Memory has a very simple testing method.
- IT is a non-volatile memory, which means that data can be stored permanently.
- The data of the read-only memory does not get deleted even if there is a sudden power cut-off.
- ROM is static in nature and does not need to be refreshed continuously.
- Circuits used in building ROM are very simple which makes it a very reliable memory.

Advantages of ROM

- Non-volatile in nature
- Cannot be accidentally changed
- Cheaper than RAMs
- Easy to test
- Circuits of ROM are simple which makes them more reliable than RAMs
- It can store data permanently
- Static and do not require refreshing
- Contents are always known and can be verified
- It helps in the completion of the bootstrap process which starts the computer and loads the OS.

Disadvantages of ROM

- It is not the quickest type of memory.
- ROM Memory has the capacity to read the data only, it cannot change the ROM data.
- Unlike RAM memory, if ROM Memory is erased incorrectly, its contents will brick the memory.
- Users can only rewrite contents in some special types of ROM Memories.

Why the ROM is called Read Only Memory?

Erasable Programmable read-only memory (EPROM)

It can be reprogrammed, but to erase data from it, one has to expose it to ultraviolet light and to reprogram it.

Advantages of EPROM

- It is a non-volatile memory in nature.
- It has the capacity to erase and rewrite the program.
- It is more cost-effective than PROM.

Disadvantages of EPROM

- It consumes more static power
- In the EPROM Memory Chip, it needs more time to delete the data

- It is not capable of deleting any specific information type stored on it, as if one byte of EPROM is deleted then rest all the bytes of EPROM are also deleted together.

Electrically erasable programmable read-only memory (EEPROM)
EEPROM is a memory in which the data can be erased by applying an electric field, there is no need for ultraviolet light. In this type of memory, data can be erased in portions of the chip also.

Advantages of EEPROM
- In EEPROM data is deleted and uploaded with the help of an electric current.
- An infinite number of times the data can be deleted and rewritten on EEPROM.
- The data stored on an EEPROM chip can be deleted in parts also.

Disadvantages of EEPROM
- EEPROM has limited data retention time.
- EEPROM is more costly than other ROM chips.
- To perform tasks like deleting and rewriting data on the EEPROM memory chip different voltages are required.

Programmable read-only memory (PROM)
This memory can be programmed by the user and once the user has programmed it, the data and programmed instructions cannot be changed.

Advantages of PROM
Hardwiring PROM is not required now as there is a lot of software available in the market for programming PROM today.

Disadvantages of PROM
The biggest disadvantage of PROM is that data cannot be modified or rewritten if any type of error has prevailed.

Difference Between EPROM and EEPROM and PROM

EPROM	EEPROM	PROM
A programmable ROM that can be crushed and reused	A user-modifiable ROM that can be erased and reprogrammed repeatedly through a normal electrical voltage	A Read-Only Memory (ROM) that can be modified only once by users
Stands for Erasable Programmable Read-Only Memory	Stands for Electrically Erasable Programmable Read-Only Memory	Stands for Programmable Read-Only Memory
Developed by Dov Frohman in 1971	Developed by George Perlegos in 1978	Developed by Wen Tsing Chow in 1956
Can be reprogrammed using ultraviolet light	Can be reprogrammed using electrical charge	Reprogrammable only once

Mask Read-Only Memory (MROM)
It is a type of read-only memory that’s pre-programmed with data. These are the original ROMs, which were hard-wired devices that contained a pre-programmed set of data or instructions

Advantages of MROM
- It has a low cost of production.
- It has a smaller storage capacity.
- It is less expensive than any other sort of secondary memory when large quantities of the same ROM are created.

Disadvantages of MROM

Design flaws are particularly expensive because if a defect occurs in the code, MROM becomes unusable and must be replaced in order to change its coding.

Difference Between RAM and ROM

RAM	ROM
It is referred to as volatile memory.	It is referred to as non-volatile memory.
It is considered temporary storage.	It is considered permanent storage.
Writing data in this type of memory is fast.	Writing data in this type of memory is slow.
This memory is used in normal operations.	This memory is used for the startup process of the computer.
In this, data is stored in MBs.	In this, data is stored in GBs.

Secondary Memory is non-volatile and long-term storage. Without it, all programs and data would be lost the moment the computer is turned off. However, it is compulsory that all computers have secondary storage. For instance, embedded computers, computers that are found in a washing machine, do not need to store data when the power is switched off. The set of information that is required to run them is stored in primary memory only. Mainly there are three types of secondary storage in a computer system:

- Solid-state storage devices (like USB memory sticks)
- Optical storage devices (like CD, DVD and Blu-ray discs)
- Magnetic storage devices (like hard disk drives)

Solid State

Solid-state is a type of storage that uses silicon microchips to store data. Like RAM, it can be written to and overwritten. Solid-state storage, unlike RAM, is non-volatile, meaning it retains its contents after the computer is turned off.

Magnetic Devices

Magnetic fields are being used to magnetise tiny individual sections of a metal rotating disc in magnetic devices like hard disc drives. A binary '1' is represented by a magnetised portion, while a binary '0' is represented by a demagnetised section. Disks can hold terabytes (TB) of data because these parts are so small.

Optical Devices

Optical devices scan the surface of a rotating plastic or metal disc with the help of a laser. The disc's surface is divided into tracks, each of which has numerous flat sections and hollows. The flat portions are referred to as lands, while the hollows are referred to as pits.

Computer Memory

What is Memory Hierarchy?

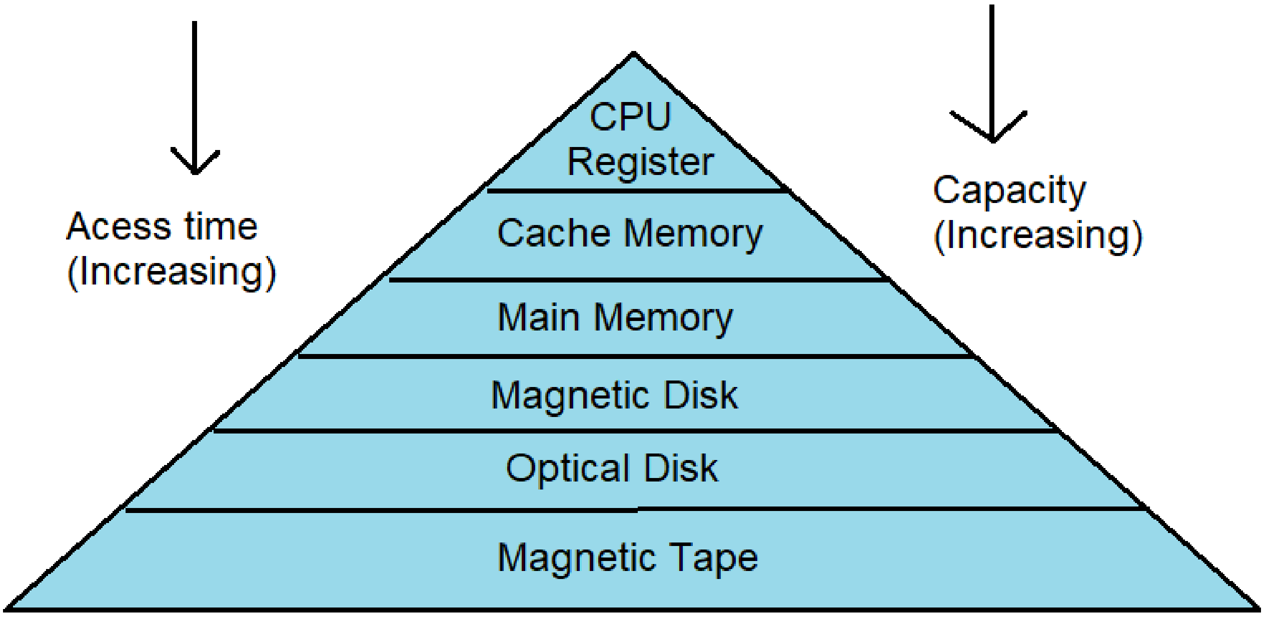


Fig:- Memory Hierarchy

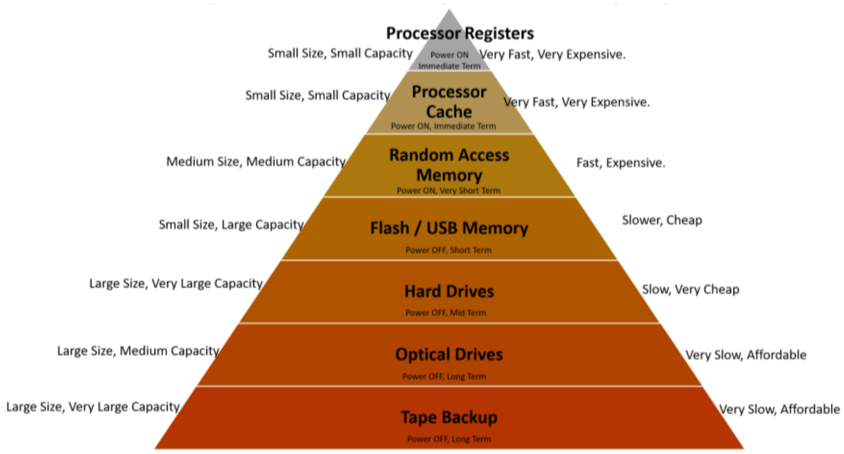
The memory hierarchy is the arrangement of various types of storage on a computing system based on access speed. It organizes computer storage according to response time. Since response time, complexity, and capacity are all connected, the levels can also be distinguished by their performance and controlling technologies.
As shown in the above picture, the computer memory has a pyramid-like structure. It is used to describe the different levels of memory. It separates the computer storage based on hierarchy.
As you can see, capacity is increasing with time. This Memory Hierarchy Design is divided into 2 types:

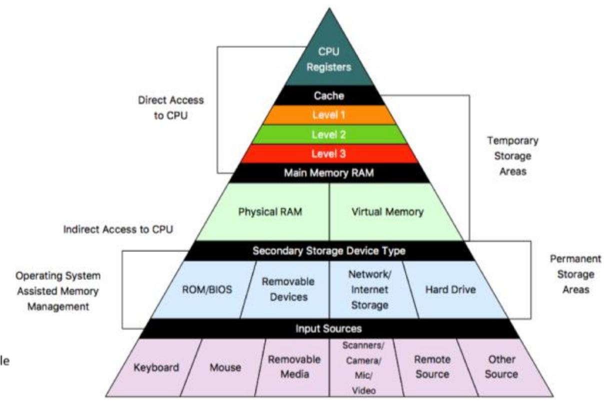
Primary or internal memory

It consists of CPU registers, Cache Memory and Main Memory, and these are directly accessible by the processor.

Secondary or external memory

It consists of a Magnetic Disk, Optical Disk and Magnetic Tape, which are accessible by processor via I/O Module.





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 temporarily hold data and instructions that the processor is likely to reuse. This allows for faster processing, as the processor does not have to wait for the data and instructions to be fetched from the RAM.

Advantages

- 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

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

System-Supported Memory Standards

According to the memory Hierarchy Design, the system supports memory standards are shown below in the table:

Level	**1**	**2**	**3**	**4**
Name	Register	Cache	Main Memory	Secondary Memory
Size	< 1KB	< 16 MB	< 16 GB	> 100 GB
Implementation	Multi ports	On-chip/SRAM	DRAM	Magnetic
Access Time	0.25-0.5ns	0.5-25ns	80ns-250ns	50 lakh ns
Bandwidth	20k to 1lakh MB	5k to 15k	1k to 5k	20 to 150
Managed by	Compiler	Hardware	OS	OS
Backing Mechanism	From cache	From main memory	from secondary memory	from ie

Characteristics of Memory Hierarchy

The memory characteristics mainly include:

- **Access Time**: The interval between data availability and a read or write request.
- **Capacity**: The amount of information that can be stored increasing as we move from top to bottom in the hierarchy.
- **Performance**: Historically, designing a computer system without a memory hierarchy resulted in a significant speed gap between main memory and CPU registers, leading to lower system performance. The memory hierarchy model was introduced to address this issue and improve system performance.
- **Cost Per Bit**: Moving from bottom to top in the system's hierarchy, the cost per bit increases, meaning that internal memory is costlier than external memory.

Why is Memory Hierarchy used in systems?

The memory hierarchy is used in computer systems to optimize the usage of available memory resources. The hierarchy is composed of different levels of memory, each with varying speed, size, and cost. The lower levels, such as registers and caches, have faster access times but are limited in capacity and more expensive, while the higher levels, such as main memory and secondary storage, have larger capacities but are slower and less expensive. By storing frequently accessed data in faster and smaller memory levels and less frequently accessed data in slower but larger memory levels, memory hierarchy ensures efficient data access, reduces processing time, and enhances overall system performance.

Advantages of Memory Hierarchy

- **Faster access:** Due to multiple levels in the memory hierarchy, users get faster access to frequently used data. Cache memory, the fastest memory, stores the most frequently used data.
- **Cost-effective:** Using the most expensive memory only where it is needed can help us reduce the cost. Cache memory is more expensive than main memory or secondary storage. With the help of memory hierarchy, it can be used only in necessary areas rather than using it everywhere.
- **Efficient use of resources:** Computer resources can be used effectively by memory hierarchy, ensuring no waste of resources. For example, high-speed access to a particular data set that is not frequently used results in a waste of resources.
- **Increased capacity:** A system can store large data in secondary storage devices such as Hard drives and SSD (solid-state drives) through memory hierarchy. On the other hand, the most frequently used data is stored inside the cache.
- **Increased processing speed:** By memory hierarchy, operations can be performed much faster as the fastest memory can be used for the most frequently used data. The CPU can access the data in a faster manner, which in turn, increases the processing speed.

What is memory hierarchy in computers?

The memory hierarchy is the arrangement of various types of storage on a computing system based on access speed. It organizes computer storage according to response time. The memory hierarchy levels are CPU registers, cache memory, main memory or primary memory, magnetic disks or secondary memory, and Optical disks or magnetic types or tertiary memory.

What is memory hierarchy and its types?

The memory hierarchy is the management and organization of a computer's memory based on the speed, characteristics, and capacities. The primary types of memory hierarchy include registers, cache, main memory (RAM), and secondary storage (hard drives, SSDs, etc.), tertiary storage.

What is memory hierarchy in ARM processors?

ARM (Advanced RISC Machines) processors are the processors that are used in smartphones, servers, and embedded systems. In them, the smaller and faster memory is close to the core on the other hand, larger and slower memory is much away.

What are the 7 levels of computer hierarchy?

The computer hierarchy has seven levels. These seven layers are high-level language, system software, machine, assembly language, digital logic, control and user.

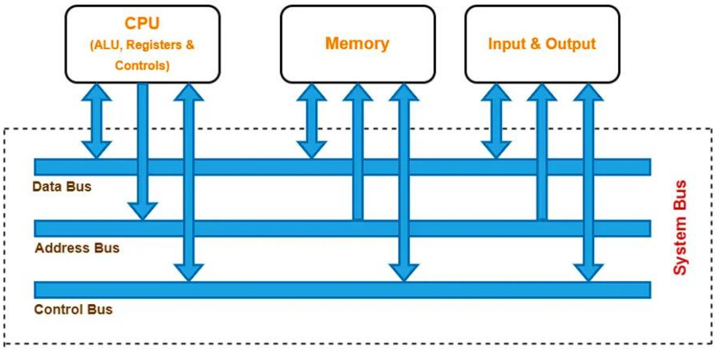
Buses

In computer architecture, a bus is a high-speed internal connection. A bus is a communication system that transfers data between the processor and other components.

What is a System Bus?

A bus that connects major components (CPU, memory and I/O devices) of a computer system is called as a System Bus.

- A bus is a set of electrical wires (lines) that connects the various hardware components of a computer system.
- It works as a communication pathway through which information flows from one hardware component to the other hardware component.



Why Do We Need Bus?

A computer system is made of different components such as ALU, CU, Registers, Memory, etc. Each component should be able to communicate with others for proper execution of instructions and information flow. If we try to implement a mesh topology among different components, it would be really expensive. So, we use a common component to connect each necessary component i.e., BUS.

Components of a Buses

In computer architecture, the system bus consists of three major components:

- Data Bus
- Address Bus
- Control Bus

Data Bus

- The data bus is used for transmitting the data or instruction from CPU to memory and I/O and vice-versa.
- It is bi-directional.

Data Bus Width

- The width of a data bus refers to the number of bits (electrical wires) that the bus can carry at a time.
- Each line carries 1 bit at a time. So, the number of lines in the data bus determines how many bits can be transferred parallel.
- The width of the data bus is an important parameter because it determines how much data can be transmitted at one time.
- The wider the bus width, the faster the data flow on the data bus and thus better the system performance.

Examples:

- A 32-bit bus has thirty-two (32) wires and thus can transmit 32 bits of data at a time.
- A 64-bit bus has sixty-four (64) wires and thus can transmit 64 bits of data at a time.

Address Bus

- The address bus is used to carry the address from the CPU to memory or I/O devices.

- It is used to identify a particular location in memory.
- It carries the source or destination address of data i.e., where to store or from where to retrieve the data.
- It is uni-directional.

Example:
When the CPU wants to read or write data, it sends the memory read or memory write control signal on the control bus to perform the memory read or write operation from the main memory and the address of the memory location is sent on the address bus.

If the CPU wants to read data stored at the memory location (address) 4, the CPU sends the value 4 in binary on the address bus.

Address Bus Width
- The width of the address bus determines the amount of physical memory addressable by the processor.
- In other words, it determines the size of the memory that the computer can use.
- The wider is the address bus, the more memory a computer will be able to use.
- The addressing capacity of the system can be increased by adding more address lines.

Examples:
- An address bus that consists of 16 wires can convey 216 (= 64K) different addresses.
- An address bus that consists of 32 wires can convey 232 (= 4G) different addresses.

Control Bus
- The control bus is used to transfer the control and timing signals from one component to the other component. The control bus also carries the clock's pulses.
- The CPU uses a control bus to communicate with the devices that are connected to the computer system.
- The CPU transmits different types of control signals to the system components.
- It is bi-directional.

What Is Control & Timing Signals?
- Control signals are generated in the control unit of CPU.
- Timing signals are used to synchronize the memory and I/O operations with a CPU clock.

Typical control signals held by control bus:
- **Memory read:** Data from memory address location to be placed on data bus.
- **Memory writes:** Data from data bus to be placed on memory address location.
- **I/O Read:** Data from I/O address location to be placed on the data bus.
- **I/O Write:** Data from the data bus is to be placed on the I/O address location.

Other control signals hold by the control bus are interrupt, interrupt acknowledge, bus request, bus grant and several others. The type of action taking place on the system bus is indicated by these control signals.

Example:
When the CPU wants to read or write data, it sends the memory read or memory write control signal on the control bus to perform the memory read or write operation from the main memory. Similarly, when the processor wants to read from an I/O device, it generates the I/O read signal.

Clock
The clock is an electrical pulse (Signal) that synchronizes the activities of the CPU. The frequency of the pulses is known as clock speed. Clock speed is measured in hertz (Hz). It determines how often instructions are executed.

In the 1980s, processors commonly ran at a rate of between 3 megahertz (MHz) and 5 MHz, which is 3 million to 5 million pulses or cycles per second. Today, processors commonly run at a rate of between 3 gigahertz (GHz) and 5 GHz, which is 3 billion to 5 billion pulses or cycles per second.

Practice Problems Based on System Bus

Problem-01:

Which of the following system bus is used to designate the source or destination of the data on the bus itself?

a) Control bus
b) Data bus
c) Address bus
d) System bus

Solution

The correct option is (C) Address bus.

Address bus carries the source or destination address of data i.e., where to store or from where to retrieve the data.

Problem-02:

The bus which is used to transfer data from main memory to peripheral device is-

a) Data bus
b) Input bus
c) DMA bus
d) Output bus

Solution

The correct option is (A) Data bus.

Data bus carries data / instruction from CPU to memory/IO and vice-versa.

Problem-03:

How many memory locations a system with a 32-bit address bus can address?

a) 2^{8}
b) 2^{16}
c) 2^{32}
d) 2^{64}

Solution

The correct option is (C) 2^{32}.

2^{32} memory locations can be addressed by a 32-bit address bus.

Problem-04:

How many bits can be transmitted at a time using a bus with 32 data lines?

a) 8 bits
b) 16 bits
c) 32 bits
d) 1024 bits

Solution

Each line carries one bit. So, a bus with 32 data lines can transmit 32 bits at a time.

Problem-05:

A microprocessor has a data bus with 64 lines and an address bus with 32 lines. The maximum number of bits that can be stored in memory is-

a) 32×2^{12}
b) 32×2^{64}
c) 64×2^{32}
d) 64×2^{64}

Solution

The correct option is (C) 64 x 2^{32}.
The amount of blocks that could be located is 2^{32}. Now, since data bus has 64 lines, so each block is 64 bits. Thus, maximum number of bits stored in memory is 2^{32} x 64 bits.

Problem-06:
The address bus with a ROM of size 1024 x 8 bits is-
a) 8 bits
b) 10 bits
c) 12 bits
d) 16 bits

Solution
The correct option is (B) 10 bits.
The size of the ROM is 1024 x 8 = 2^{10} x 8. Here, 10 indicates the address bus and 8 indicates the data bus width.

Problem-07:
The data bus width of a ROM of size 2048 x 8 bits is-
a) 8
b) 10
c) 12
d) 16

Solution
The correct option is (A) 8.
The size of the ROM is 2048 x 8 = 2^{11} x 8. Here, 11 indicates the address bus and 8 indicates the data bus width.

Motherboard
The motherboard serves as a single platform to connect all of the parts of a computer together. It connects the CPU, memory, hard drives, optical drives, video card, sound card, and other ports and expansion cards directly or via cables. It can be considered as the backbone of a computer.

Features of Motherboard
- 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.

Popular Manufacturers
Following are the popular manufacturers of the motherboard:
- Intel
- ASUS
- AOpen
- ABIT
- Biostar
- Gigabyte

- MSI

Description of Motherboard
The motherboard is mounted inside the case and is securely attached via small screws through pre-drilled holes. Motherboard contains ports to connect all of the internal components. It provides a single socket for CPU, whereas for memory, normally one or more slots are available. Motherboards provide ports to attach the floppy drive, hard drive, and optical drives via ribbon cables. Motherboard carries fans and a special port designed for power supply.

There is a peripheral card slot in front of the motherboard using which video cards, sound cards, and other expansion cards can be connected to the motherboard.
On the left side, motherboards carry a number of ports to connect the monitor, printer, mouse, keyboard, speaker, and network cables. Motherboards also provide USB ports, which allow compatible devices to be connected in plug-in/plug-out fashion. For example, pen drive, digital cameras, etc.

Memory Units

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:

SL. 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

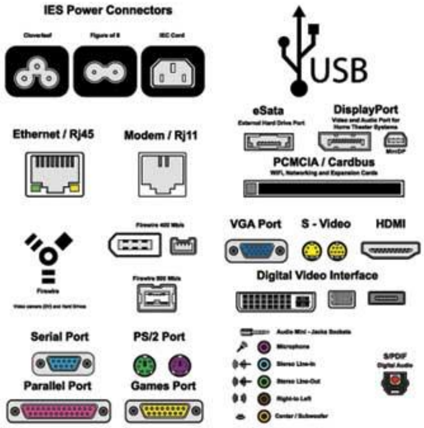
SL. No.	Unit & Description
1	**Kilobyte (KB)** 1 KB = 1024 Bytes
2	**Megabyte (MB)** 1 MB = 1024 KB
3	**GigaByte (GB)** 1 GB = 1024 MB
4	**TeraByte (TB)** 1 TB = 1024 GB
5	**PetaByte (PB)** 1 PB = 1024 TB

Ports

A port is a physical docking point using which an external device can be connected to the computer. It can also be programmatic docking point through which information flows from a program to the computer or over the Internet.

Characteristics of Ports

- External devices are connected to a computer using cables and ports.
- Ports are slots on the motherboard into which a cable of external device is plugged in.
- Examples of external devices attached via ports are the mouse, keyboard, monitor, microphone, speakers, etc.



Serial Port

- Used for external modems and older computer mouse
- Two versions: 9 pins, 25 pin model
- Data travels at 115 kilobits per second

Parallel Port

- Used for scanners and printers
- Also called printer port
- 25 pin model
- IEEE 1284-compliant Centronics port

PS/2 Port

- Used for old computer keyboard and mouse
- Also called mouse port
- Most of the old computers provide two PS/2 port, each for the mouse and keyboard
- IEEE 1284-compliant Centronics port

Universal Serial Bus (or USB) Port

- It can connect all kinds of external USB devices such as external hard disk, printer, scanner, mouse, keyboard, etc.
- It was introduced in 1997.
- Most of the computers provide two USB ports as minimum.
- Data travels at 12 megabits per seconds.
- USB compliant devices can get power from a USB port.

VGA Port

- Connects monitor to a computer's video card.
- It has 15 holes.
- Similar to the serial port connector. However, serial port connector has pins, VGA port has holes.

Power Connector

- Three-pronged plug.
- Connects to the computer's power cable that plugs into a power bar or wall socket.

Firewire Port
- Transfers large amount of data at very fast speed.
- Connects camcorders and video equipment to the computer.
- Data travels at 400 to 800 megabits per seconds.
- Invented by Apple.
- It has three variants: 4-Pin FireWire 400 connector, 6-Pin FireWire 400 connector, and 9-Pin FireWire 800 connector.

Modem Port
- Connects a PC's modem to the telephone network.

Ethernet Port
- Connects to a network and high-speed Internet.
- Connects the network cable to a computer.
- This port resides on an Ethernet Card.
- Data travels at 10 megabits to 1000 megabits per seconds depending upon the network bandwidth.

Game Port
- Connect a joystick to a PC
- Now replaced by USB

Digital Video Interface, DVI port
- Connects Flat panel LCD monitor to the computer's high-end video graphic cards.
- Very popular among video card manufacturers.

Sockets
- Sockets connect the microphone and speakers to the sound card of the computer.

Computer Hardware
Hardware represents the physical and tangible components of a computer, i.e., the components that can be seen and touched. Examples of Hardware are the following:
- **Input devices:** Keyboard, mouse, etc.
- **Output devices:** Printer, monitor, etc.
- **Secondary storage devices:** Hard disk, CD, DVD, etc.
- **Internal components:** CPU, motherboard, RAM, etc.

Relationship between Hardware and Software
- Hardware and software are mutually dependent on each other. Both of them must work together to make a computer produce a useful output.
- Software cannot be utilized without supporting hardware.
- Hardware without a set of programs to operate upon cannot be utilized and is useless.
- To get a particular job done on the computer, relevant software should be loaded into the hardware.
- Hardware is a one-time expense.

- Software development is very expensive and is a continuing expense.
- Different software applications can be loaded on a hardware to run different jobs.
- A software acts as an interface between the user and the hardware.
- If the hardware is the 'heart' of a computer system, then the software is its 'soul'. Both are complementary to each other.

Computer Software

Software is a set of programs, which is designed to perform a well-defined function. A program is a sequence of instructions written to solve a particular problem.

There are two types of software −

- System Software
- Application Software

System Software

The system software is a collection of programs designed to operate, control, and extend the processing capabilities of the computer itself. System software is generally prepared by the computer manufacturers. These software products comprise of programs written in low-level languages, which interact with the hardware at a very basic level. System software serves as the interface between the hardware and the end users. Some examples of system software are Operating System, Compilers, Interpreter, Assemblers, etc.

Here is a list of some of the most prominent features of a system software:

- Close to the system
- Fast in speed
- Difficult to design
- Difficult to understand
- Less interactive
- Smaller in size
- Difficult to manipulate
- Generally written in low-level language

Application Software

Application software products are designed to satisfy a particular need of a particular environment. All software applications prepared in the computer lab can come under the category of Application software. Application software may consist of a single program, such as Microsoft's notepad for writing and editing a simple text. It may also consist of a collection of programs, often called a software package, which work together to accomplish a task, such as a spreadsheet package. Examples of Application software are the following:

- Payroll Software
- Student Record Software
- Inventory Management Software
- Income Tax Software
- Railways Reservation Software
- Microsoft Office Suite Software
- Microsoft Word
- Microsoft Excel
- Microsoft PowerPoint

Features of application software are as follows:

- Close to the user
- Easy to design
- More interactive
- Slow in speed
- Generally written in high-level language
- Easy to understand
- Easy to manipulate and use
- Bigger in size and requires large storage space

Computer Number System

When we type some letters or words, the computer translates them in numbers as computers can understand only numbers. A computer can understand the positional number system where there are only a few symbols called digits and these symbols represent different values depending on the position they occupy in the number. The value of each digit in a number can be determined using:

- The digit
- The position of the digit in the number
- The base of the number system (where the base is defined as the total number of digits available in the number system)

Decimal Number System

The number system that we use in our day-to-day life is the decimal number system. Decimal number system has base 10 as it uses 10 digits from 0 to 9. In decimal number system, the successive positions to the left of the decimal point represent units, tens, hundreds, thousands, and so on.

Each position represents a specific power of the base (10). For example, the decimal number 1234 consists of the digit 4 in the unit's position, 3 in the tens position, 2 in the hundreds position, and 1 in the thousands position. Its value can be written as

(1 x 1000)+ (2 x 100)+ (3 x 10)+ (4 x l)
$(1 \times 10^3)+ (2 \times 10^2)+ (3 \times 10^1)+ (4 \times 10^0)$
1000 + 200 + 30 + 4
1234

As a computer programmer or an IT professional, you should understand the following number systems which are frequently used in computers.

SL. No.	Number System and Description
1	**Binary Number System** Base 2. Digits used : 0, 1
2	**Octal Number System** Base 8. Digits used : 0 to 7
3	**Hexa Decimal Number System** Base 16. Digits used: 0 to 9, Letters used : A- F

Binary Number System

Characteristics of the binary number system are as follows:

- Uses two digits, 0 and 1
- Also called as base 2 number system

- Each position in a binary number represents a **0** power of the base (2). Example 2^0
- Last position in a binary number represents a **x** power of the base (2). Example 2^x where **x** represents the last position - 1.

Example

Binary Number: 10101_2

Calculating Decimal Equivalent

Step	Binary Number	Decimal Number
Step 1	10101_2	$((1 \text{ x } 2^4) + (0 \text{ x } 2^3) + (1 \text{ x } 2^2) + (0 \text{ x } 2^1) + (1 \text{ x } 2^0))_{10}$
Step 2	10101_2	$(16 + 0 + 4 + 0 + 1)_{10}$
Step 3	10101_2	21_{10}

Note: 10101_2 is normally written as 10101.

Octal Number System

Characteristics of the octal number system are as follows:

- Uses eight digits, 0,1,2,3,4,5,6,7
- Also called as base 8 number system
- Each position in an octal number represents a **0** power of the base (8). Example 8^0
- Last position in an octal number represents a **x** power of the base (8). Example 8^x where **x** represents the last position - 1

Example

Octal Number: 12570_8

Calculating Decimal Equivalent

Step	Octal Number	Decimal Number
Step 1	12570_8	$((1 \text{ x } 8^4) + (2 \text{ x } 8^3) + (5 \text{ x } 8^2) + (7 \text{ x } 8^1) + (0 \text{ x } 8^0))_{10}$
Step 2	12570_8	$(4096 + 1024 + 320 + 56 + 0)_{10}$
Step 3	12570_8	5496_{10}

Note: 12570_8 is normally written as 12570.

Hexadecimal Number System

Characteristics of hexadecimal number system are as follows:

- Uses 10 digits and 6 letters, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- Letters represent the numbers starting from 10. A = 10. B = 11, C = 12, D = 13, E = 14, F = 15
- Also called as base 16 number system
- Each position in a hexadecimal number represents a **0** power of the base (16). Example, 16^0
- Last position in a hexadecimal number represents a **x** power of the base (16). Example 16^x where **x** represents the last position - 1

Example

Hexadecimal Number: $19FDE_{16}$

Calculating Decimal Equivalent

Step	Binary Number	Decimal Number
Step 1	$19FDE_{16}$	$((1 \text{ x } 16^4) + (9 \text{ x } 16^3) + (F \text{ x } 16^2) + (D \text{ x } 16^1) + (E \text{ x } 16^0))_{10}$
Step 2	$19FDE_{16}$	$((1 \text{ x } 16^4) + (9 \text{ x } 16^3) + (15 \text{ x } 16^2) + (13 \text{ x } 16^1) + (14 \text{ x } 16^0))_{10}$

Step 3	$19FDE_{16}$	$(65536+ 36864 + 3840 + 208 + 14)_{10}$
Step 4	$19FDE_{16}$	106462_{10}

Note: $19FDE_{16}$ is normally written as 19FDE.

Computer - Number Conversion

There are many methods or techniques which can be used to convert numbers from one base to another. In this chapter, we'll demonstrate the following:

- Decimal to Other Base System
- Other Base System to Decimal
- Other Base System to Non-Decimal
- Shortcut method - Binary to Octal
- Shortcut method - Octal to Binary
- Shortcut method - Binary to Hexadecimal
- Shortcut method - Hexadecimal to Binary

Decimal to Other Base System

Step 1: Divide the decimal number to be converted by the value of the new base.
Step 2: Get the remainder from Step 1 as the rightmost digit (least significant digit) of the new base number.
Step 3: Divide the quotient of the previous divide by the new base.
Step 4: Record the remainder from Step 3 as the next digit (to the left) of the new base number.
Repeat Steps 3 and 4, getting remainders from right to left, until the quotient becomes zero in Step 3.
The last remainder thus obtained will be the Most Significant Digit (MSD) of the new base number.

Example

Decimal Number: 29_{10}
Calculating Binary Equivalent

Step	**Operation**	**Result**	**Remainder**
Step 1	29 / 2	14	1
Step 2	14 / 2	7	0
Step 3	7 / 2	3	1
Step 4	3 / 2	1	1
Step 5	1 / 2	0	1

As mentioned in Steps 2 and 4, the remainders have to be arranged in the reverse order so that the first remainder becomes the Least Significant Digit (LSD) and the last remainder becomes the Most Significant Digit (MSD).

Decimal Number : 29_{10} = Binary Number : $11101_{2.}$

Other Base System to Decimal System

Step 1: Determine the column (positional) value of each digit (this depends on the position of the digit and the base of the number system).
Step 2: Multiply the obtained column values (in Step 1) by the digits in the corresponding columns.
Step 3: Sum the products calculated in Step 2. The total is the equivalent value in decimal.

Example

Binary Number: 11101_2
Calculating Decimal Equivalent

Step	**Binary Number**	**Decimal Number**

Step 1	11101_2	$((1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0))_{10}$
Step 2	11101_2	$(16 + 8 + 4 + 0 + 1)_{10}$
Step 3	11101_2	29_{10}

Binary Number : 11101_2 = Decimal Number : 29_{10}

Other Base System to Non-Decimal System
Step 1: Convert the original number to a decimal number (base 10).
Step 2: Convert the decimal number so obtained to the new base number.
Example
Octal Number : 25_8
Calculating Binary Equivalent
Step 1**:** Convert to Decimal

Step	Octal Number	Decimal Number
Step 1	25_8	$((2 \times 8^1) + (5 \times 8^0))_{10}$
Step 2	25_8	$(16 + 5)_{10}$
Step 3	25_8	21_{10}

Octal Number : 25_8 = Decimal Number : 21_{10}

Step 2**:** Convert Decimal to Binary

Step	Operation	Result	Remainder
Step 1	21 / 2	10	1
Step 2	10 / 2	5	0
Step 3	5 / 2	2	1
Step 4	2 / 2	1	0
Step 5	1 / 2	0	1

Decimal Number : 21_{10} = Binary Number : 10101_2
Octal Number : 25_8 = Binary Number : 10101_2

Shortcut Method - Binary to Octal
Step 1: Divide the binary digits into groups of three (starting from the right).
Step 2: Convert each group of three binary digits to one octal digit.
Example
Binary Number : 10101_2
Calculating Octal Equivalent

Step	Binary Number	Octal Number
Step 1	10101_2	010 101
Step 2	10101_2	2_8 5_8
Step 3	10101_2	25_8

Binary Number : 10101_2 = Octal Number : 25_8
Shortcut Method **-** Octal to Binary

Step 1: Convert each octal digit to a 3-digit binary number (the octal digits may be treated as decimal for this conversion).
Step 2: Combine all the resulting binary groups (of 3 digits each) into a single binary number.
Example
Octal Number : 25_8
Calculating Binary Equivalent

Step	Octal Number	Binary Number
Step 1	25_8	2_{10} 5_{10}
Step 2	25_8	010_2 101_2
Step 3	25_8	010101_2

Octal Number : 25_8 = Binary Number : 10101_2
Shortcut Method - Binary to Hexadecimal
Step 1: Divide the binary digits into groups of four (starting from the right).
Step 2: Convert each group of four binary digits to one hexadecimal symbol.
Example
Binary Number : 10101_2
Calculating hexadecimal Equivalent

Step	Binary Number	Hexadecimal Number
Step 1	10101_2	0001 0101
Step 2	10101_2	1_{10} 5_{10}
Step 3	10101_2	15_{16}

Binary Number : 10101_2 = Hexadecimal Number : 15_{16}
Shortcut Method - Hexadecimal to Binary
Step 1: Convert each hexadecimal digit to a 4-digit binary number (the hexadecimal digits may be treated as decimal for this conversion).
Step 2: Combine all the resulting binary groups (of 4 digits each) into a single binary number.
Example
Hexadecimal Number : 15_{16}
Calculating Binary Equivalent

Step	Hexadecimal Number	Binary Number
Step 1	15_{16}	1_{10} 5_{10}
Step 2	15_{16}	0001_2 0101_2
Step 3	15_{16}	00010101_2

Hexadecimal Number : 15_{16} = Binary Number : 10101_2

Endianness

Different languages read their text in different orders. English reads from left to right, for example, while Arabic is read from right to left.

This is exactly what **endianness** is for computers. If my computer reads **bytes** from left to right, and your computer reads from right to left, we're going to have issues when we need to communicate.

What is the meaning of the byte-ordering address or byte-ordering sequence?

Suppose, we have a data and we have to save or store this data in the computer memory, which sequence we will flow, to store the data in the computer memory this process is called Endianness.

Definition: Endianness means byte-ordering address or byte-ordering sequence. Endianness is the ordering of sequencing of bytes of a digital data to store into the computer memory storage during the transmission. Endianness is primarily expressed as Big-Endian (BE) or Little-Endian (LE).

What are Big-Endian and Little-Endian?

Big-endian is an order in which the "big end" (most significant value in the sequence) is stored first, at the lowest storage address. Little-endian is an order in which the "little end" (least significant value in the sequence) is stored first.

All computers do not store the bytes that comprise a multi-byte value in the same order. Consider a 16-bit internet that is made up of 2 bytes. Two ways to store this value:

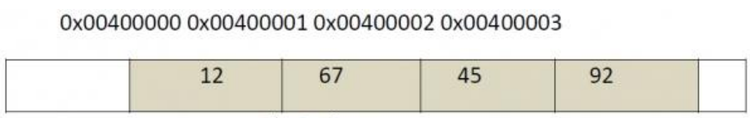
- **Little Endian:** In this scheme, a low-order byte is stored on the starting address (A) and a high-order byte is stored on the next address (A + 1).
- **Big Endian:** In this scheme, a high-order byte is stored on the starting address (A) and a low-order byte is stored on the next address (A + 1).

To allow machines with different byte order conventions to communicate with each other, the Internet protocols specify a canonical byte order convention for data transmitted over the network. This is known as Network Byte Order.

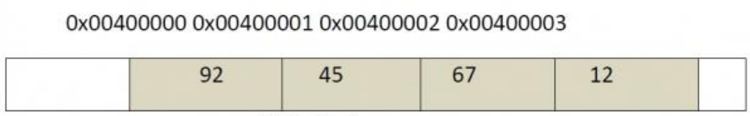
By these definitions, a 32-bit data pattern, which is regarded as a 32-bit unsigned integer. The "high-order" byte is the one for the largest powers of 2: 2^{31}, ..., 2^{24}. The "low-order" byte is the one for the smallest powers of 2: 2^7, ..., 2^0.

Example

0×12674592 in 32-bit representation can be stored as:



Big Endian



Little Endian

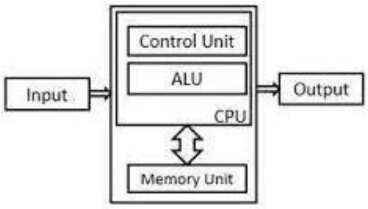
Von Neumann and Harvard Architecture

Von Neumann and Harvard architecture both are computer architectures. They describe a design architecture for an electronic digital computer.

Von Neumann Architecture

The Von Neumann architecture is a type of computer architecture in which the Central Processing Unit (CPU), memory, and Input/Output (I/O) devices all use a single shared bus for communication.

It was named after John von Neumann, a pioneer in the development of electronic computers. The Von Neumann architecture is also known as the Von Neumann model or the Princeton architecture.



Von Neumann Model

In a Von Neumann architecture, instructions and data are stored in a single memory system. This memory is usually implemented as random access memory (RAM) and is connected to the CPU and I/O devices through a common bus. The CPU executes instructions by fetching them from memory, decoding them, and then executing them.

The Von Neumann architecture consists of four main components

- **CPU:** The central processing unit is the brain of the computer. It performs all the calculations and operations required to execute instructions.
- **Memory:** The memory holds both instructions and data that the CPU needs to execute. The memory is organized as a linear sequence of addressable cells, each containing a fixed number of bits.
- **System Bus:** The system bus is a collection of wires that connect the CPU, memory, and I/O devices. It is used to transmit data, instructions, and control signals between these components.
- **Input/Output (I/O) Devices:** These devices are used to communicate with the outside world. Examples: Include keyboards, displays, and printers.

Advantages of Von Neumann Architecture

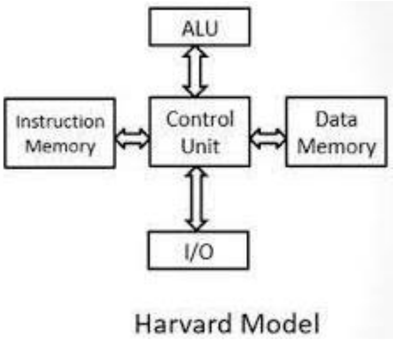
- The Von Neumann architecture has several advantages, including simplicity, flexibility, and efficiency.
- Because instructions and data are stored in the same memory system, the CPU can easily access both of them without having to switch between different memory systems. This makes it easier to write programs and reduces the complexity of the system as a whole.

Disadvantages of Von Neumann Architecture

- One of the main limitations is that the shared bus can become a bottleneck if too many devices are connected to it. This can lead to slow performance and reduced scalability.
- Additionally, the CPU can only execute one instruction at a time, which can limit the overall speed of the system.

Harvard Architecture

Harvard architecture is a type of computer architecture that has separate memory spaces for instructions and data. It was developed at Harvard University in the 1930s, and it is named after this institution. In a Harvard architecture system, the CPU accesses instruction and data memory spaces separately, which can lead to improved performance.



Harvard Model

The Harvard architecture consists of the following main components:

- **CPU:** The central processing unit performs all the calculations and operations required to execute instructions.
- **Instruction Memory:** This memory holds instructions that the CPU needs to execute. It is typically implemented as read-only memory (ROM) or flash memory.
- **Data memory:** This memory holds data that the CPU needs to perform computations. It is typically implemented as random access memory (RAM).
- **System Bus:** The system bus is a collection of wires that connect the CPU, instruction memory, data memory, and I/O devices. It is used to transmit data, instructions, and control signals between these components.
- **Input/Output (I/O) Devices:** These devices are used to communicate with the outside world. Examples include keyboards, displays, and printers.

Advantages of Harvard Architecture

- The CPU can access both instruction and data memory simultaneously.
- This can lead to improved performance because the CPU does not have to switch between memory spaces as often as in a Von Neumann architecture.
- Additionally, because the instruction memory is typically implemented as ROM or flash memory, it is non-volatile, meaning that it does not lose its contents when power is turned off.
- This makes it well-suited for embedded systems that need to operate without a constant power source.

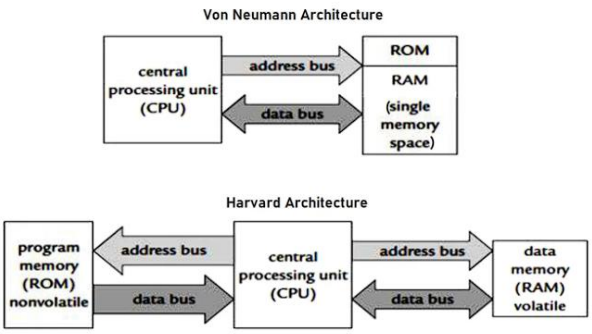
Disadvantages of Harvard Architecture

- As the CPU accesses instruction and data memory separately, it can be more difficult to write programs that require the CPU to modify its own code.
- Additionally, because the instruction and data memories are separate, it can be more difficult to share data between different parts of a program.

Why Harvard Architecture is Effective?

After coming across all, the fact definitely one question will definitely strike to your mind that why Harvard architecture is so effective? The answer is quite clear and simple that the architecture is able to read an instruction and it can also perform data memory access simultaneously at a fast speed. Hence the Harvard architecture is being widely accepted.

Difference Between Von-Neumann and Harvard Architecture



Here is a tabular comparison between Von Neumann architecture and Harvard architecture:

Aspect	Von Neumann Architecture	Harvard Architecture
Memory	Single memory for instructions and data	Separate memory for instructions and data
Access	CPU accesses instructions and data through a shared bus	CPU accesses instruction and data memory spaces separately
Performance	Can become a bottleneck if too many devices are connected to the shared bus	Improved performance because the CPU can access both instruction and data memory simultaneously
Modification	Easier to modify programs as instructions and data are stored in the same memory	More difficult to modify programs that require the CPU to modify its own code
Data Sharing	Easy to share data between different parts of a program	More difficult to share data between different parts of a program
Applications	Suitable for general-purpose computing where flexibility is required	Suitable for embedded systems where performance is critical and code is not frequently modified

Overall, Von Neumann architecture is more flexible and easier to program, whereas Harvard architecture is more efficient and better suited for embedded systems that require high performance and reliability.

What is Von Neumann bottleneck problem? How can this be reduced or resolved?

The Von Neumann bottleneck, also known as the Von Neumann bottleneck problem, refers to a limitation in the processing speed of a computer caused by the traditional von Neumann architecture, which separates the processing unit (CPU) and memory.

How can this CPU-memory disparity be resolved?

This problem of *Von Neumann bottleneck* can be solved in two ways:

1. Use of cache memory between CPU and main memory
2. Using RISC computers

This performance problem can be reduced by introducing a **cache memory** (special type of fast memory) in between the CPU and the main memory.
This is because the speed of the cache memory is almost same as that of the CPU. So there is no waiting time for CPU and data-word to come to it for processing.

Another way of solving the problem is by using special type of computer known as **Reduced Instruction Set Computers** (RISC).
The main intention of the RISC is to reduce the total number of memory references made by the CPU; instead, it uses large number of registers for the same purpose.