UNIT I

Syllabus: Basic Structure Of Computers: Functional unit, Basic Operational concepts, Bus structures, System Software, Performance, The history of computer development.

Functional Unit (Or) Structure of a Computer System:

EveryDigital computersystems consistof fivedistinct functional units. Theseunits areas follows:

1. **Input unit**
2. **Memory unit**
3. **Arithmetic and logic unit**

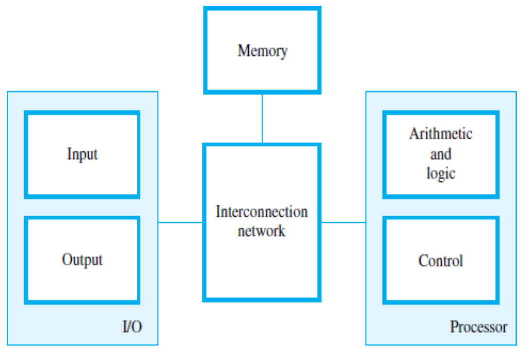


Figure 1.1 Basic functional units of a computer.

4. **Output unit**
5. **Control Unit**

These units are interconnected by electrical cables to permit communication between them. A computer must receive both data and program statements to function properly and be able to solve problems. The method of feeding data and programs to a computer is accomplished by an input device. **Computer input devices read data from a source, such as magnetic disks, and translate that data into electronic impulses for transfer into the CPU**. **Example for input devices are a keyboard, a mouse, or a scanner**.**Central Processing Unit The brain of a computer system is the central processing unit (CPU).The CPU processes data transferred to it from one of the various input devices. It then transfers either an intermediate or final result of the CPU to one or more output devices. A central control section and work areas are required to perform calculations or manipulate data.** The CPU is the computing center of the system**. It consists of a control section, an arithmetic-logic section, and an internal storage section(main memory).** Each section within the CPU serves a specific function and has a particular relationship with the other sections within the CPU.

Input Unit: An input device is usually a **keyboard or mouse**, the input device is the device through which data and instructions enter a computer.

1. **The most common input device is the *keyboard*, which accepts letters, numbers, and commandsfrom the user.**

2. **Another important type of input device is *the mouse*, which lets you select options from on-screen menus**. You use a mouse by moving it across a flat surface and pressing its buttons. A variety of other input devices work with personal computers, too:
3. **The trackball and touchpad are variations of the mouse and enable you to draw or point on the screen.**
   The joystick is a swiveling lever mounted on a stationary base that is well suited for playingvideo games

Memory unit: memory is used to store programs and data. **There are two classes of storage, called primary and secondary.**

Primary storage: It is a fast memory that operates at electronic speeds. Programs must stay in memory while they are being executed.**The memory contains a large number of semiconductor storage cells, eachcapable of storing one bit of information.** To provide easy access to any word in the memory, a distinctaddress is associated with each word location. Addresses are numbers that identify successive locations. Agivenword is accessed by specifying its addressand issuing a control command.

The number of bits in each word is referred as the word length of the computer.

Typical wordlength range from 16 to 64 bits.

Programs must reside in the memory during execution. Instructions and data can be written into the memory or read out under thecontrol of theprocessor.

1. **Memory in which anylocation can be reached in a short and fixed amount of time after specifying its address is called *randomaccess memory* (RAM).**
2. **The time required to access oneword is called the *memoryaccesstime*.**
3. **The small, fast, Ram units are *called caches*. They are tightly coupled with the processor and are often contained on the same integrated circuitchip to achieve high performance**.
4. **The largest and slowest units are referred to as the *main memory*.**

Secondary storage: Secondary storage is used when large amounts of data and many programs have to be stored, particularly for information that is accessed in frequently.
Examples for secondary storage devices are *MagneticDisks, TapeandOptical disks.*

Arithmetic-Logic Unit:-The arithmetic-logic section performs **arithmetic operations, such as addition, subtraction, multiplication, and division.**

Arithmetic-Logic Unit usually called the ALU is a digital circuit that performs two types ofoperations—arithmeticand logical.

Arithmetic operations are the fundamental mathematical operations consisting of addition, subtraction, multiplication and division.

Logical operations consist of comparisons. That is, two pieces of data are compared to see whetherone is equal to, less than, or greater than the other. The ALU is a fundamental building block of the centralprocessingunit of acomputer.

Out put Unit:-An **output device** is any piece of computer hardware equipment used to communicate theresults of data processing carried out by an information processing system (such as a computer) to the outsideworld.

In computing, input/output, or I/O, refers to the communication between an information processingsystem (such as a computer), and the outside world. Inputs are the signals or data sent to the system, and outputs are the signals or data sent by the system to the outside.
Examples of output devices:

- Speaker
- Headphones
- Screen
- Printer

\

Control Unit: All activities inside the machine are directed and controlled by the control unit. **Control Unit** is the part of the computer's central processing **unit** (CPU),which directs the **operation** of the processor. A **control unit** works by receiving input information to which it converts into **control** signals,whicharethen sent to the central processor

BASIC OPERATIONAL CONCEPTS OF COMPUTER

To perform a given task an appropriate program consisting of a list of instructions is stored in the memory. Individual instructions are brought from the memory into the processor, which executes the specified operations. Data to be stored are also stored in the memory.

Examples: - Add LOCA, R0

This instruction adds the operand at memory location LOCA, to operand in register R0 & places the sum into register. This instruction requires the performance of several steps,

1. First the instruction is fetched from the memory into the processor.
2. The operand at LOCA is fetched and added to the contents of R0.
3. Finally the resulting sum is stored in the register R0

The preceding add instruction combines a memory access operation with an ALU Operations. In some other type of computers, these two types of operations are performed by separate instructions for performance reasons.

Load LOCA, R1

Add R1, R0

Transfers between the memory and the processor are started by sending the address of the memory location to be accessed to the memory unit and issuing the appropriate control signals. The data are then transferred to or from the memory

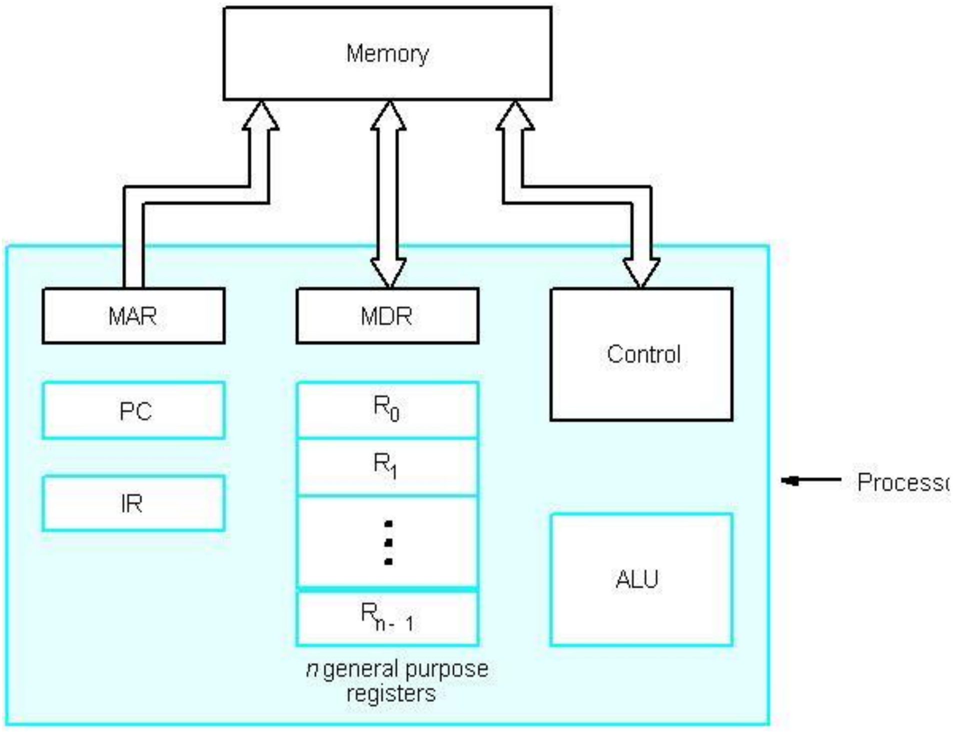


Fig 1.2 Connections between the processor and the memory

The fig shows how memory & the processor can be connected. In addition to the ALU & the control circuitry, the processor contains a number of registers used for several different purposes.

The instruction register (IR):- Holds the instructions that is currently being executed. Its output is available for the control circuits which generates the timing signals that control the various processing elements in one execution of instruction.

The program counter (PC):- This is another specialized register that keeps track of execution of a program. It contains the memory address of the next instruction to be fetched and executed.

Besides IR and PC, there are n-general purpose registers R0 through Rn-1.

The other two registers which facilitate communication with memory are: -

1. MAR – (Memory Address Register):- It holds the address of the location to be accessed.
2. MDR – (Memory Data Register):- It contains the data to be written into or read out of the address location.

Operating steps are:-

1. Programs reside in the memory & usually get these through theInput unit.
2. Execution of the program starts when the PC is set to point at the first instruction of the program.
3. Contents of PC are transferred to MAR and a Read Control Signal is sent to the memory.
4. After the time required to access the memory elapses, the address word is read out of the memory and loaded into the MDR.
5. Now contents of MDR are transferred to the IR & now the instruction is ready to be decoded and executed.
6. If the instruction involves an operation by the ALU, it is necessary to obtain the required operands.
7. An operand in the memory is fetched by sending its address to MAR & Initiating a read cycle.
8. When the operand has been read from the memory to the MDR, it is transferred from MDR to the ALU.
9. After one or two such repeated cycles, the ALU can perform the desired operation.
10. If the result of this operation is to be stored in the memory, the result is sent to MDR.
11. Address of location where the result is stored is sent to MAR & a write cycle is initiated. 12. The contents of PC are incremented so that PC points to the next instruction that is to be executed.

Normal execution of a program may be preempted (temporarily interrupted) if some devices require urgent servicing, to do this one device raises an Interrupt signal.

An interrupt is a request signal from an I/O device for service by the processor. The processor provides the requested service by executing an appropriate interrupt service routine.

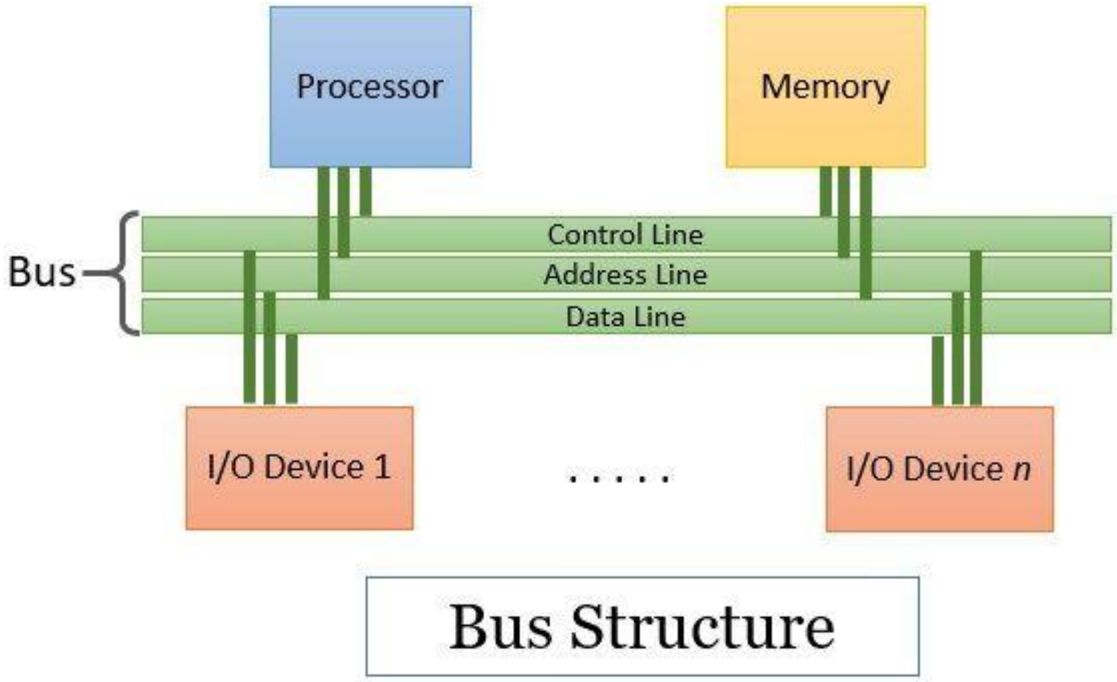
The Diversion may change the internal stage of the processor its state must be saved in the memory location before interruption. When the interrupt-routine service is completed the state of the processor is restored so that the interrupted program may continue.

Bus Structure

Bus structures in computer plays important role in connecting the internal components of the computer. The bus in the computer is the *shared transmission medium*. This means multiple components or devices use the same bus structure to transmit the information signals to each other.

At a time, only one pair of devices can use this bus to communicate with each other successfully. If multiple devices transmit the information signal over the bus at the same time, the signals overlap each other and get jumbled

A system bus has typically from fifty to hundreds of distinct lines where each line is meant for a certain function. These lines can be categorised into three functional groups i.e., data lines, address lines, and control lines. Let us discuss them one by one each.



1. Data Lines

Data lines coordinate in transferring the data among the system components. The data lines are collectively called data bus. A data bus may have 32 lines, 64 lines, 128 lines, or even more lines. The number of lines present in the data bus defines the *width* of the data bus.

Each data line is able to transfer only one bit at a time. So the number of data lines in a data bus determines how many bits it can transfer at a time. The performance of the system also depends on the width of the data bus.

2. Address Lines

The content of the address lines of the bus determines the source or destination of the data present on the data bus. The number of address lines together is referred to as the address bus. The number of address lines in the address bus determines its *width*.

The width of the address bus determines the memory capacity of the system. The content of address lines is also used for addressing I/O ports. The higher-order bits determine the bus module, and the lower-ordered bits determine the address of memory locations or I/O ports.

Whenever the processor has to read a word from memory, it simply places the address of the corresponding word on the address line.

3. Control Lines

The address lines and data lines are shared by all the components of the system, so there must be some means to control the use and access of data and address lines. The control signals placed on the control lines control the use and access to the address and data lines of the bus. The control signal consists of the *command* and *timing information.* Here the command in the control signal specifies the *operation* that has to be performed. And the timing information over the control signals specifies when the data and address information is valid.

The control lines include the lines for:

- **Memory Write:** This command causes the data on the data bus to be placed over the addressed memory location.
- **Memory Read:** This command causes the data on the addressed memory location to be placed on the data bus.
- **I/O Write:** The command over this control line causes the data on the data bus to be placed over the addressed I/O port.
- **I/O Read:** The command over this control line causes the data from the addressed I/O port to be placed over the data bus.
- **Transfer ACK:** This control line indicates the data has been received from the data bus or is placed over the data bus.
- **Bus Request:** This control line indicates that the component has requested control over the bus.
- **Bus Grant:** This control line indicates that the bus has been granted to the requesting component.
- **Interrupt Request:** This control line indicates that interrupts are pending.
- **Interrupt ACK:** This control line acknowledges when the pending interrupt is serviced.
- **Clock:** This control line is used to synchronize the operations.
- **Reset:** The bit information over this control line initializes all the modules.

Suppose a component connected to the bus wishes to send data to another connected component. In that case, it first has to acquire control over the bus, and then it can transfer the data to another component over the bus. The same happens when a component request data from another component.

During data transfer between two components, one component act as a master and the other act as a slave. The device initiating the data transfer is referred to as the *master,* and usually, it is a processor, or sometimes it may be some other device or component. The component addressed by the master component is referred to as a *slave*.

Timing in Bus

As we have seen, the bus’s control lines also provide timing information along with the command. Well, the way of deriving the timing information over the control line can be categorized in two ways:

1. Synchronous Bus

With the synchronous bus scheme, all the devices or components connected to the bus derive timing information over the control line referred to as the *bus clock.* Over the bus clock line, the clock transmits an alternating sequence of 1s and 0s at regular intervals. A single 1-0 transmission is considered a clock or bus cycle.

All the devices or components connected to the bus can read this bus clock line, and all the events start at the starting the clock cycle. Here the transmitting component and the receiving component are synchronized using the clock. The data is sent or received constantly and, therefore used for high-speed transmission.

2. Asynchronous Bus

The clock does not synchronise the transmitter and receiver components in this asynchronous bus scheme. Instead, the data transfer is controlled using a handshake protocol between the master component and the slave component.

Here, the component initiating the data transfer i.e. master component then gets ready for data transfer, and indicates this by activating its master-ready line and placing the address and command information over the bus.

Then all the connected component decodes the address on the address line to recognize which component is being addressed by the master component.

Now the addressed component performs the required operation and notifies the processor by activating its slave ready-line. Once the master recognizes the activated slave ready-line, it removes its control over the bus.

In this way, the occurrence of one event on the bus is followed by and depends on the occurrence of a previous event.

Software:- A total computer system includes both software and Hardware.

1. **Hardware consists of physical components** and all associated equipment.
2. **Software refers to the collection programs that are written for the computer** and writing a program for a computer consists of specifying, directly or indirectly a sequence of machine instructions.
3. The computer software consists of the instructions and data that the computer manipulates to perform various data processing tasks.

Types:

1. Application software,
2. System software

System software: System software is used to run application software.

System software is a collection of programs that are executed as needed to perform functions such as

1. **Receiving and interpreting user commands.**
2. **Entering and editing application programs and sorting them as files in secondary storage devices.(Editor)**
3. **Managing the storage and retrieval of files in secondary storage devices.**
4. **Running standard application programs such as wordprocessors, spreadsheets, or games, with data supplied by the user.**
5. **Controlling I/O units to receive input information and produce output results.**
6. **Translating programs from high level language to low level language.(Assemblers)**
7. **Linking and running user-written application program with existing standard libraryroutines, such as numerical computation packages.(Linker)**

Application software: Application software allows end users to accomplish one or more specific (notdirectly computer development related) tasks. Its usually written in high level languages, such as c, c++, java. Typical applications include:

- **Wordprocessing**
- **spreadsheet**
- **computer games**
- **databases**
- **industrial automation**
- **business software**
- quantum chemistry and solid state physics software
- telecommunications(i.e.,theinternetand everythingthatflowsonit)
- educational software
- medical software
- military software
- molecular modelling software
- imageediting
- simulationsoftware
- Decisionmakingsoftware

Compiler:- A **compiler** is a computer program (or set of programs) that **transforms source code written ina computer language (the *source language*) into another computer language (the *target language*, oftenhaving a binary form known as *object code*).** The most common reason for wanting to transform source code is to create an executable program. The name "compiler" is primarily used for programs that translate source code from a high-level programming language to a lower level language (e.g., assembly language ormachine code). A program that translates from a low level language to a higher level one is a *decompiler*. A program that translates between high-level languages is usually called a *language translator, source to source translator*

Linker:- Linker is a program in a system which helps to link a object modules of program into a single object file. It performs the process of linking. Linker are also called link editors. Linking is process ofcollecting and maintaining piece of code and data into a single file. Linker also link a particular module intosystem library. It takes object modules from assembler as input and forms an executable file as output forloader. Linking is performed at both compile time, when the source code is translated into machine code andload time, when the program is loaded into memory by the loader. Linking is performed at the last step incompilingaprogram.

Assembler: - An assembler is a program that converts assembly language into machine code. **It takes the basic commands and operations from assembly code and converts them into binary code that can be recognized by a specific type of processor**. Assemblers are similar to compilers in that they produceexecutable code. However, assemblers are more simplistic since they only convert low-level code (assembly language) to machine code. Since each assembly language is designed for a specific processor, assembling a program is performed using a simple one-to-one mapping from assembly code to machine code.

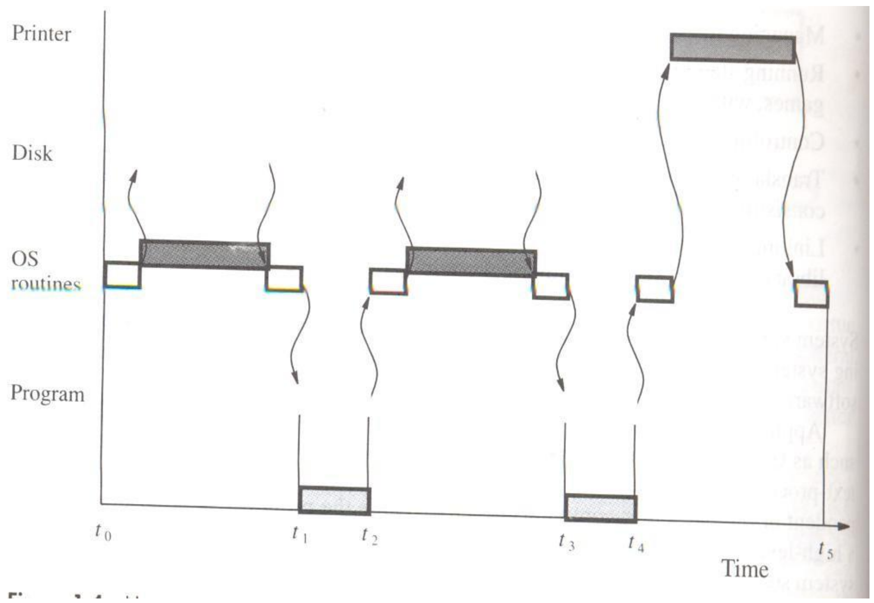
Loader:-A loader is a major component of an operating system that ensures all necessary programs andlibraries are loaded, which is essential during the startup phase of running a program. It places the librariesandprograms into themain memory in order to preparethem forexecution.

System software Component->OS (OPERATING SYSTEM)

Operating System: It is a large program or a collection of routines that is used to control the sharing of and interaction among various computer units.

Functionsof OS:

- Assign resources to individual application program.
- Assign memory and magnetic disk space to program and data files.
- ♦ Move the data between memory and disk units. Handles I/O operations.



User program and os routine sharing of the processor

Steps:-

1) The first step is to transfer the file into memory.

2) When the transfer is completed, the execution of the program starts.

3) During time period t0 to t1, an OS routine initiates loading the application program from disk to memory, wait until the transfer is complete and then passes theexecutioncontrol to the application program & print the results.
4) Similar action takes place during„t2‟to„t3‟ and „t4‟to„t5‟.
5) At „t5‟,Operating System may load and execute another application program.
6) Similarly during„ t0‟to„t1‟, the Operating System can arrange to print the previous program's results while the current program is beingexecuted.
7) The pattern of managing the concurrent execution of the several applicationprograms to make the best possible use of computer resources is called the multi-programming or multi-tasking.

Performance

<u>Performance: -</u>The most important measure of the performance of a computer is how quickly it cancompute programs. The speed with which a computer executes programs is affected by the design of its hardware and its machine language instructions. To represent the performance of a processor, we should consider only the periods during which the processor is active.

At the start of execution, all program instructions and the required **data are stored in the memory as shown below.As execution proceeds, instructions are fetched one by one over the bus into the processor, and a copy is placed in the cache**. **When the execution of instruction calls for data located in the main memory, the data are fetched and a copy is placed in the cache**. Later, if the same instruction or data item is needed a second time, it is read directly from the cache.

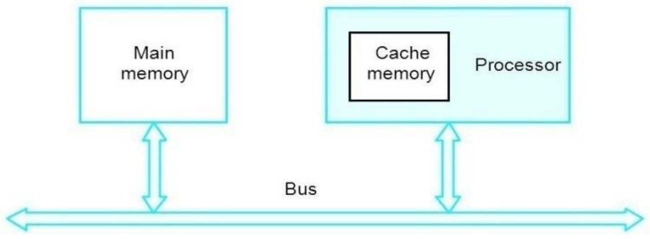


Figure 1.5. The processor cache.

Computer performance is often described in terms of clock speed (usually in MHz or GHz). This refers to the cycles per second of the main clock of the CPU. Performance of a computer depends on the following factors.

a) Processorclock:-

1. Processor circuits are controlled by a timing signal called a clock.A clock is a microchip thatregulatesspeed and timing ofall computer functions.
2. **Clock Cycle** is the speed of a computer processor, or CPU, which is the amount of time between twopulses of an oscillator. Generally speaking, the higher number of pulses per second, the faster the computer processor will be able to process information
3. CPU **clockspeed**, or **clockrate**, is measured in Hertz—generally in gigahertz, or GHz. A CPU's **clock speed rate** is a measure of how many **clock** cycles a CPU can perform per second
4. To execute a machine instruction, the processor divides the action to be performed into a sequence of basic steps, such that each step can be completed in one clock cycle.
5. The length P of one clock cycle is an important parameter that affects processor performance.
6. Itsinverseis theclock rate, R=1/P, whichis measuredincycles persecond.
7. If the clock rate is 500(MHz) million cycles per second, then the corresponding clock period is 2nanoseconds.

b) Basic performance equation:-The **Performance Equation** is a term used in computer science. It refers to the calculation of the performance or speed of a centralprocessing unit (CPU).

Basically the *Basic Performance Equation[BPE]* is an equation with 3 parameters which are required for the calculation of "Basic Performance" of a given system. It is given by

T=(N*S)/R

Where **'T'** is the ***processor time*** [Program Execution Time] required to execute a given program written in some high level language. The compiler generates a machine language object program corresponding to the source program.

'N' is the **total number of steps required to complete program execution**. 'N' is the actual number of instruction executions, not necessarily equal to the total number of machine language instructions in theobject program. Some instructions are executed more than others (loops) and some are not executed at all(conditions).

'S' is the **average number of basic steps each instruction execution requires**, where each basic step is completed in one clock cycle. We say average as each instruction contains a variable number of stepsdependingon theinstruction.

'R'is the**clock rate**[In cyclespersecond]

c) <u>PipeliningandSuperscalaroperation:-</u>

1. A substantial improvement in performance can be achieved by overlapping the execution of successive instructions, using a technique called pipelining.
2. Considertheinstruction
3. Add R1, R2, R3
4. Which adds the contents of registers R1 and R2, and places the sum intoR3
5. The contents of R1 and R2 are first transferred to the inputs of the ALU.
6. After the add operation is performed, the sum is transferred toR3.
7. Processor can read the next instructionfromthememorywhiletheadditionoperationisbeingperformed.
8. Then, if that instruction also uses the ALU, its operands can be transferred to the ALU inputs at the same time that the result of add instruction is being transferred to R3.
9. Thus, pipelining increases the rate of executing instructions significantly.

d) <u>Superscalaroperation:-</u>

1. A higher degree of concurrency can be achieved if multiple instruction pipelines are implemented in the processor.
2. This means that multiple function units are used, creating parallel paths through which different instructions can be executed in parallel.
3. With such an arrangement, it becomes possible to start the execution of several instructions in every clock cycle.
4. This mode of execution is called super scalar operation.

e)<u>Clockrate:-</u>

1. There are two possibilities for increasing the clock rate, R.
2. First, improving the Integrated Circuittechnology makes logic circuit faster, which reduces theneeded to complete a basic step.This allows the clock period, P, to be reduced and the clock rate, R, to be increased.
3. Second, reducing the amount of processing done in one basic step also makes it possible to reduce theclockperiod, P.

f)<u>Instructionset: CISCand RISC:-</u>

1. The terms CISC and RISC refer to design principles and techniques.
2. RISC: Reduced instruction set computers.
3. Simple instructions require a small number of basic steps to execute.
4. For a processor that has only simple instructions, a large number of instructions may be need to perform a given programming task. This could lead to a large value of N and a small value for S.
5. It is much easier to implement efficient pipelining in processors with simple instruction sets.
6. CISC: Complex instruction set computers.
7. Complex instructions involve a large number of steps.
8. If individual instructions perform more complex operations, fewer instructions will be needed, leading to a lower value of N and a larger value of S.
9. Complex instructions combined with pipelining would achieve good performance.

g)<u>OptimizingCompiler:-</u>

1. A compiler translates a high-level language program into a sequence of machine instructions.
2. To reduce N, we need to have a suitable machine instruction set and a compiler that makes good use of it.
3. An optimizing compiler takes advantage of various features of the target processor to reduce the product N* S.
4. The compiler may rearrange program instructions to achieve better performance.

h)<u>Performancemeasurement:-</u>

1. SPECrating.
2. A non profit organization called "System Performance Evaluation Corporation"(SPEC) selects and publishes representative application programs for different application domains.
3. The SPEC rating is computed as follows.
4. $$\text{SPEC rating}=\frac{\text{Running time on the reference computer}}{\text{Running time on the computer under test.}}$$
5. Thus SPEC rating of 50 means that the computer under test is 50times faster than the reference computer for this particular benchmarks.
6. The test is repeated for all the programs in the SPEC suite, and the geometric mean of the results is computed.
7. Let SPEC, be the rating for program 'i' in the suite. The overall SPEC rating for the computer is given by

   $$\text{SPEC rating} = \prod_{i=1}^{n} (\text{SPEC}_i)$$

   Where n is the number of programs in the suite.

The history of computer development

Computer Types: Basing capacity, technology used and performance of computer, they are classified into two types

→According to computational ability
→According to generation

According to computational ability (Based on Size, cost and performance): There are mainly 4 types of computers. These include:

a) Micro computers
b) Mainframe computers
c) Mini computers
d) Super computer

a) Micro computers: - Micro computers are the most common type of computers in existence today, whether at work in school or on the desk at home. These computers include:

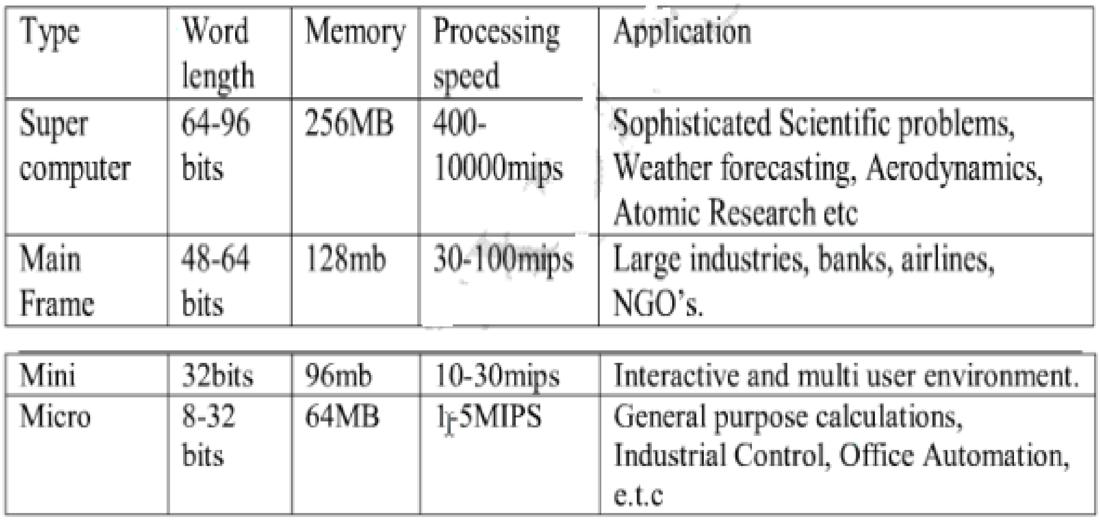
1. Desktop computer
2. Personal digital assistants (more commonly known as PDA's)
3. Palmtop computers
4. Laptop and notebook computers

Micro computers were the smallest, least powerful and least expensive of the computers of the time. The first Micro computers could only perform one task at a time, while bigger computers ran multi-tasking operating systems, and served multiple users. Referred to as a personal computer or "desktop computer", Micro computers are generally meant to service one user (person) at a time. By the late 1990s, all personal computers run a multi-tasking operating system, but are still intended for a single user.

b) Mainframe Computers :- The term Mainframe computer was created to distinguish the traditional, large, institutional computer intended to service multiple users from the smaller, single user machines. These computers are capable of handling and processing very large amounts of data easily and quickly. A mainframe speed is so fast that it is measured in millions of tasks per milliseconds (MTM). While other computers became smaller, Mainframe computers stayed large to maintain the ever growing memory capacity and speed.
Mainframe computers are used in large institutions such as government, banks and large corporations. These institutions were early adopters of computer use, long before personal computers were available to individuals. "Mainframe" often refers to computers compatible with the computer architectures established in the 1960's. Thus, the origin of the architecture also affects the classification, not just processing power.

c) Mini Computers / Workstation :- Mini computers, or Workstations, were computers that are one step above the micro or personal computers and a step below mainframe computers. They are intended to serve one user, but contain special hardware enhancements not found on a personal computer. They run operating systems that are normally associated with mainframe computers, usually one of the variants of the UNIX operating system.

d) Super Computer:- A Super computer is a specialized variation of the mainframe. Where a mainframe is intended to perform many tasks, a Super computer tends to focus on performing a single program of intense numerical calculations. Weather forecasting systems, Automobile design systems, extreme graphic generator for example, are usually based on super computers.



Type	Word length	Memory	Processing speed	Application
Super computer	64-96 bits	256MB	400-10000mips	Sophisticated Scientific problems, Weather forecasting, Aerodynamics, Atomic Research etc
Main Frame	48-64 bits	128mb	30-100mips	Large industries, banks, airlines, NGO's.
Mini	32bits	96mb	10-30mips	Interactive and multi user environment.
Micro	8-32 bits	64MB	1-5MIPS	General purpose calculations, Industrial Control, Office Automation, e.t.c

According to Generations of Computers:
The history of computer development is often referred to in reference to the different generations of computing devices. Each generation of computer is characterized by a major technological development that fundamentally changed the way computers operate, resulting in increasingly smaller, cheaper, more powerful and more efficient and reliable devices.

a) **First Generation (1940-1956): Vacuum Tubes: The first computers used vacuum tubes for circuitry and magnetic drums for memory, and were often enormous, taking up entire rooms.** They were very expensive to operate and in addition to using a great deal of electricity, generated a lot of heat, which was often the cause of malfunctions. First generation computers relied on machine language, the lowest-level programming language understood by computers, to perform operations, and they could only solve one problem at a time.

   Input was based on punched cards and paper tape, and output was displayed on printouts. Example: The UNIVAC and ENIAC computers are examples of first-generation computing devices. The UNIVAC was the first commercial computer delivered to a business client, the U.S. Census Bureau in 1951.

b) **Second Generation (1956-1963): Transistors:- Transistors replaced vacuum tubes and ushered in the second generation of computers**. The transistor was invented in 1947 but did not see widespread use in computers until the late 1950s. The transistor was far superior to the vacuum tube, allowing computers to become smaller, faster, cheaper, more energy-efficient and more reliable than their first-generation predecessors. Though the transistor still generated a great deal of heat that subjected the computer to damage, it was a vast improvement over the vacuum tube. Second-generation computers still relied on punched cards for input and printouts for output.
Second-generation computers moved from cryptic binary machine language to symbolic, or assembly, languages, which allowed programmers to specify instructions in words. High-level programming languages were also being developed at this time, such as early versions of COBOL and FORTRAN. These were also the first computers that stored their instructions in their memory, which moved from a magnetic drum to magnetic core technology. The first computers of this generation were developed for the atomic energy industry

c) **Third Generation(1964-1971):IntegratedCircuits: The development of the integrated circuit** was the hallmark of the third generation of computers.Transistors were miniaturized and placed on silicon chips, called semiconductors, which drastically increased the speed and efficiency of computers. Instead of punched cards and printouts, users interacted with third generation computers through keyboards and monitors and interfaced with an operating system, which allowed the device to run many different applications at one time with a central program that monitored the memory. Computers for the first time became accessible to a mass audience because they were smaller and cheaper than the earlier.

d) **Fourth Generation (1971-Present):Microprocessors:** The microprocessor brought the fourth generation of computers, as thousands **of integrated circuitswere built onto a single silicon chip**. What in the first generation filled an entire room could now fit in the palm of the hand. **The Intel 4004 chip**, developed in 1971, located all the components of the computer—fromthecentral processing unit and memory to input/output controls—onasinglechip.
In 1981 IBM introduced its first computer for the home user, and in 1984 Apple introduced the Macintosh. Microprocessors also moved out of the realm of desktop computers and into many areas of life as more and more everyday products began to usemicroprocessors.
As these small computers became more powerful, they could be linked together to form networks, which eventually led to the development of the Internet. Fourth generation computers also saw the development of GUIs, the mouse and handheld devices.

e) **FifthGeneration(PresentandBeyond):ArtificialIntelligence):**Fifth generation computing devices, based on artificial intelligence, are still in development, though there are some applications, such as voice recognition, that are being used today.

The use of parallel processing and superconductors is helping to make artificial intelligence a reality. Quantum computation andmolecular and nanotechnology will radically change the face of computers in years to come. The goal offifth-generation computing is to develop devices that respond to natural language input and are capable of learning and self-organization

.