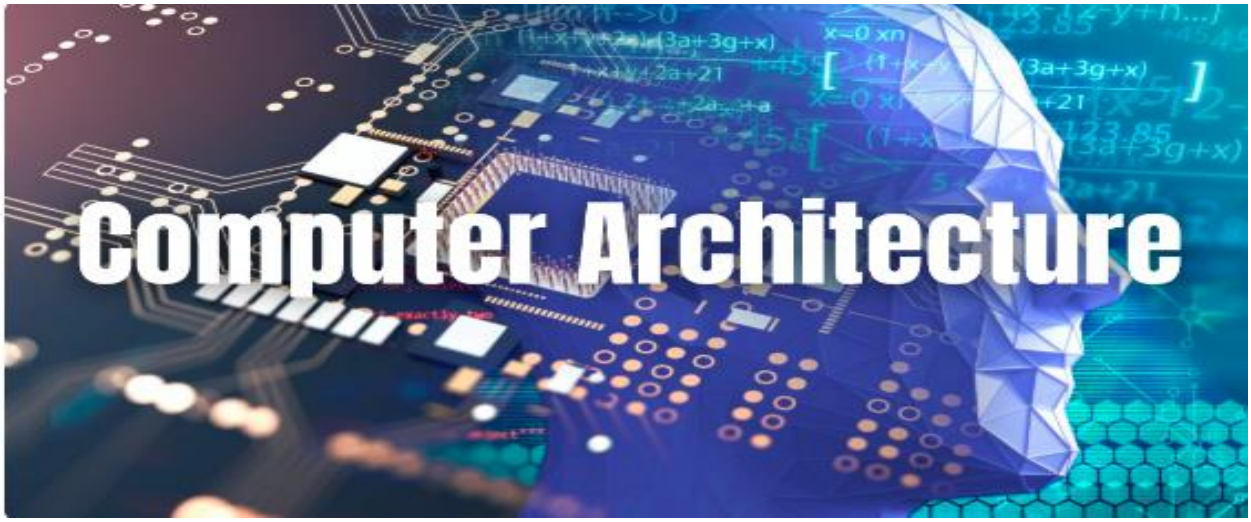
COMPUTER ARCHITECTURE

COM 300



Intro

What is computer Architecture?

Computer architecture is the study of how computers are designed and built, including their components and how they work together.

Types of Computers

Computers can be categorized on the basis of: size and data handling capabilities.

- Further, on the basis of data handling capabilities, computers can be divided into three types:

1. Analogue Computer
2. Digital Computer
3. Hybrid Computer

1. Analogue computer

They are designed so that they can process the analogue data. The data that can change continuously and do not have discrete values such as temperature, current, speed and pressure is known as Analogue data.

The continuous changes that physical quantity goes through are measured by analogue computers. The output rendered by them is generally in the form of a reading on a dial or scale.

Analogue computers don't wait for the data to get converted into codes and numbers and rather accept the data from the measuring device directly.

Examples

Mercury meter and speedometer are the examples of analogue computer.

2. Digital Computers

They are designed to perform logical operations and calculations at a high speed. A digital computer accepts raw data as numbers or digits and then, to produce output, it processes it with the programs stored in its memory.

Examples

The modern computers like desktops and laptops fall under the category of digital computers.

3. Hybrid Computer

It contains the features of both Digital and Analogue computers. It has accuracy and memory like digital computers and is fast like Analogue computers. It can process both discrete and continuous data and hence it is widely used in specialised applications where both digital and analogue data is processed,

Example

Petrol pump where a processor is used to convert the measurement of fuel flow into price and quantity.

Types of computers based on sizes

- Supercomputers
- Mainframe computer
- Miniframe computer
- Workstation
- Microcomputer

1. Supercomputers

They are the fastest in speed and the biggest in size and specialise in processing a huge amount of data.

A supercomputer contains thousands of interconnected processors that help it process trillions of instructions in just a second.

Roger Cray developed the first Supercomputer in 1976. Supercomputers are specifically used in engineering and scientific applications such as nuclear energy research, scientific simulations, and weather forecasting.

2. Mainframe computer

They have the capacity of supporting hundreds or thousands of users simultaneously and they can also support multiple programs at the same time. This means that different processes can be executed simultaneously by a Mainframe Computer. Hence, organisations that need to process and manage high volume of data find mainframe computers ideal for them. Telecom and banking sectors are examples of such organisations.

3. Miniframe computer

It is a multiprocessing computer of midsize. It has the capacity of entertaining 4 to 200 users at one given time and consists of 2 or more processors. Miniframe computers are mostly used in departments and institutes for the tasks like inventory management, billing and accounting.

4. **Workstation**

It is a single user computer designed for scientific or technical applications. It flaunts high speed graphic adapters, faster microprocessor and a large amount of RAM. It can perform specific jobs with great expertise. There are different types of workstation computers like engineering design workstation, graphic workstation and music workstation.

5. **Microcomputer**

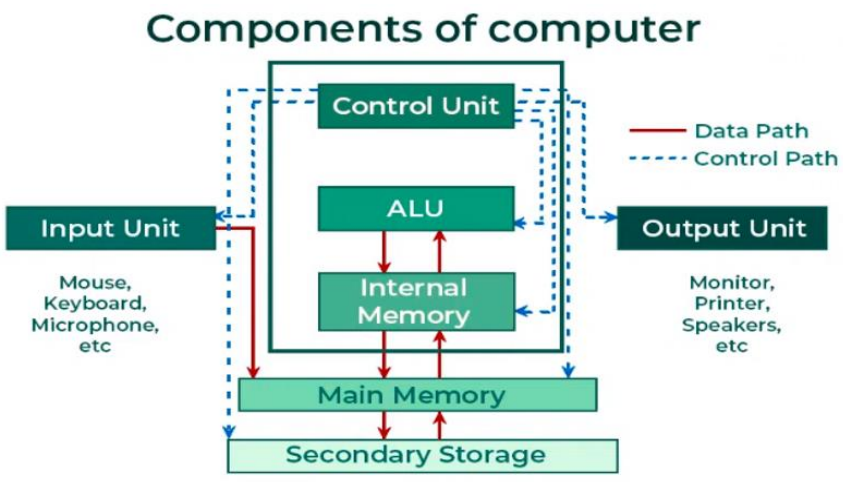
It is also known as Personal Computer. This general purpose computer is mainly designed for individual use. It consists of a microprocessor, which works as the Central Processing Unit, storage area, memory, and input and output unit. The most famous

Examples of Microcomputers are Desktop computers and Laptops.

Computer Components

A computer has 5 main components.

- Input devices
- CPU
- Output devices
- Primary memory
- Secondary memory



Input Unit

- The input unit consists of input devices that are attached to the computer. These devices take input and convert it into binary language that the computer understands. Some of the common input devices are keyboard, mouse, joystick, scanner etc.

The Central Processing Unit (CPU)

- Once the information is entered into the computer by the input device, the processor processes it. The CPU is called the brain of the computer because it is the control centre of the computer. It first fetches instructions from memory and then interprets them so as to know what is to be done. If required, data is fetched from memory or input device.

- Thereafter CPU executes or performs the required computation, and then either stores the output or displays it on the output device. The CPU has three main components, which are responsible for different functions: Arithmetic Logic Unit (ALU), Control Unit (CU) and Memory registers

A. Arithmetic and Logic Unit (ALU)

The ALU, as its name suggests performs mathematical calculations and takes logical decisions.

Arithmetic calculations include addition, subtraction, multiplication and division.

Logical decisions involve the comparison of two data items to see which one is larger or smaller or equal.

B. Control Unit

The Control unit coordinates and controls the data flow in and out of the CPU, and also controls all the operations of ALU, memory registers and also input/output units.

It is also responsible for carrying out all the instructions stored in the program. It decodes the fetched instruction, interprets it and sends control signals to input/output devices until the required operation is done properly by ALU and memory.

C. Memory Registers

A register is a temporary unit of memory in the CPU. These are used to store the data, which is directly used by the processor. Registers can be of different sizes(16 bit, 32 bit, 64 bit and so on) and each register inside the CPU has a specific function, like storing data, storing an instruction, storing address of a location in memory etc.

Output Unit

- The output unit consists of output devices that are attached to the computer. It converts the binary data coming from the CPU to human understandable form.
- The common output devices are monitor, printer, plotter, etc.

Computer – Memory

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

Memory is primarily of three types –

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

Cache Memory

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

The advantages of cache memory are as follows –

- Cache memory is faster than main memory.
- It consumes less access time as compared to main memory.

- It stores the program that can be executed within a short period of time.
- It stores data for temporary use.

The disadvantages of cache memory are as follows –

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

Primary Memory (Main Memory)

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

Characteristics of Main Memory

- These are semiconductor memories.
- It is known as the main memory.
- Usually volatile memory.
- Data is lost in case power is switched off.
- It is the working memory of the computer.

- Faster than secondary memories.
- A computer cannot run without the primary memory.

Random Access Memory

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

- Access time in RAM is independent of the address, that is, each storage location inside the memory is as easy to reach as other locations and takes the same amount of time. Data in the RAM can be accessed randomly but it is very expensive.
- 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 of two types –

- Static RAM (SRAM)
- Dynamic RAM (DRAM)

Static RAM (SRAM)

- The word **static** indicates that the memory retains its contents as long as power is being supplied. However, data is lost when the power gets down due to 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.
- There is extra space in the matrix, hence SRAM uses more chips than DRAM for the same amount of storage space, making the manufacturing costs higher. SRAM is thus used as cache memory and has very fast access.

Characteristic of Static RAM

- Long life
- No need to refresh
- Faster

- Used as cache memory
- Large size
- Expensive
- High power consumption

Dynamic RAM (DRAM)

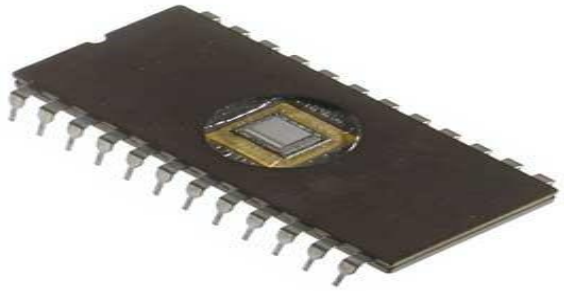
- DRAM, unlike SRAM, must be continually **refreshed** in order to maintain the data. This is done by placing the memory on a refresh circuit that rewrites the data several hundred times per second. DRAM is used for most system memory as it is cheap and small. All DRAMs are made up of memory cells, which are composed of one capacitor and one transistor.

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

Computer - Read Only Memory

- ROM stands for **Read Only Memory**. 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 machine and microwave oven.



Various types of ROMs

- MROM (Masked ROM)
- PROM (Programmable Read Only Memory)
- EPROM (Erasable and Programmable Read Only Memory)
- EEPROM (Electrically Erasable and Programmable Read Only Memory)

MROM (Masked ROM)

- The very first ROMs were hard-wired devices that contained a pre-programmed set of data or instructions. These kind of ROMs are known as masked ROMs, which are inexpensive.

PROM (Programmable Read Only Memory)

- PROM is read-only memory that can be modified only once by a user. The user buys a blank PROM and enters the desired contents using a PROM program. Inside the PROM chip, there are small fuses which are burnt open during programming. It can be programmed only once and is not erasable.

EPROM (Erasable and Programmable Read Only Memory)

- EPROM can be erased by exposing it to ultra-violet light for a duration of up to 40 minutes. Usually, an EPROM eraser achieves this function. During programming, an electrical charge is trapped in an insulated gate region. The charge is retained for more than 10 years because the charge has no leakage path. For erasing this charge, ultra-violet light is passed through a quartz crystal window (lid). This exposure to ultra-violet light dissipates the charge. During normal use, the quartz lid is sealed with a sticker.

EEPROM (Electrically Erasable and Programmable Read Only Memory)

- EEPROM is programmed and erased electrically. It can be erased and reprogrammed about ten thousand times. Both erasing and programming take about 4 to 10 ms (millisecond). In EEPROM, any location can be selectively erased and programmed. EEPROMs can be erased one byte at a time, rather than erasing the entire chip. Hence, the process of reprogramming is flexible but slow.

The advantages of ROM are as follows –

- Non-volatile in nature
- Cannot be accidentally changed
- Cheaper than RAMs
- Easy to test
- More reliable than RAMs
- Static and do not require refreshing
- Contents are always known and can be verified
-

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.

Parallel Computing :
It is the use of multiple processing elements simultaneously for solving any problem. Problems are broken down into instructions and are solved concurrently as each resource that has been applied to work is working at the same time.
Advantages of Parallel Computing over Serial Computing are as follows:

1. It saves time and money as many resources working together will reduce the time and cut potential costs.
2. It can be impractical to solve larger problems on Serial Computing.

3. It can take advantage of non-local resources when the local resources are finite.
4. Serial Computing ‘wastes’ the potential computing power, thus Parallel Computing makes better work of the hardware.

Types of Parallelism:

1. **Bit-level parallelism –**
   It is the form of parallel computing which is based on the increasing processor’s size. It reduces the number of instructions that the system must execute in order to perform a task on large-sized data.
   Example: Consider a scenario where an 8-bit processor must compute the sum of two 16-bit integers. It must first sum up the 8 lower-order bits, then add the 8 higher-order bits, thus requiring two instructions to perform the operation. A 16-bit processor can perform the operation with just one instruction.
2. **Instruction-level parallelism –**
   A processor can only address less than one instruction for each clock cycle phase. These instructions can be re-ordered and grouped which are later on executed concurrently without affecting the result of the program. This is called instruction-level parallelism.
3. **Task Parallelism –**
   Task parallelism employs the decomposition of a task into subtasks and then allocating each of the subtasks for execution. The processors perform the execution of sub-tasks concurrently.
4. **Data-level parallelism (DLP) –**

Instructions from a single stream operate concurrently on several data – Limited by non-regular data manipulation patterns and by memory bandwidth

Why parallel computing?

- The whole real-world runs in dynamic nature i.e. many things happen at a certain time but at different places concurrently. This data is extensively huge to manage.
- Real-world data needs more dynamic simulation and modeling, and for achieving the same, parallel computing is the key.
- Parallel computing provides concurrency and saves time and money.
- Complex, large datasets, and their management can be organized only and only using parallel computing’s approach.
- Ensures the effective utilization of the resources. The hardware is guaranteed to be used effectively whereas in serial computation only some part of the hardware was used and the rest rendered idle.
- Also, it is impractical to implement real-time systems using serial computing.

Applications of Parallel Computing:

- Databases and Data mining.
- Real-time simulation of systems.

- Science and Engineering.
- Advanced graphics, augmented reality, and virtual reality.

Limitations of Parallel Computing:

- It addresses such as communication and synchronization between multiple sub-tasks and processes which is difficult to achieve.
- The algorithms must be managed in such a way that they can be handled in a parallel mechanism.
- The algorithms or programs must have low coupling and high cohesion. But it's difficult to create such programs.
- More technically skilled and expert programmers can code a parallelism-based program well.

Pipelining

Pipelining is a computer architecture technique that allows multiple instructions to be processed simultaneously. This is done by dividing the processor into stages, and assigning an instruction to each stage.

Principles of pipelining

- **Stages**: The stages of pipelining are connected in a pipe-like structure.
- **Instruction flow**: Instructions enter the pipeline from one end and exit from the other.
- **Instruction throughput**: Pipelining increases the overall number of instructions that can be processed.
- **Idle time**: Pipelining reduces idle time for the CPU.
- **Data dependencies**: When an instruction in one stage depends on the results of a previous instruction, the pipeline can stall.
  Examples of pipelining stages: Instruction fetch (IF), Instruction decoding (ID), Execute (EX), Memory access (MA), and Write back (WB).

Benefits of pipelining

Pipelining is important for understanding how modern processors can achieve high-speed performance and multitasking capabilities.

Multithreading

In the context of computer architecture, "interleaved multithreading" refers to the technique of switching between instructions from different threads within a single processor pipeline cycle, while "simultaneous multithreading (SMT)" allows multiple

instructions from different threads to be issued and executed concurrently in the same cycle, essentially running multiple threads seemingly simultaneously on a single core; the key difference being that interleaved multithreading switches between threads more frequently, while SMT can execute instructions from multiple threads in parallel within a single cycle, requiring more advanced hardware capabilities.

Key points about interleaved multithreading:

- **Instruction switching:**

  Instructions from different threads are issued one after another in the pipeline, essentially "interleaving" their execution.

- **Fine-grained vs. coarse-grained:**

  Can be implemented with varying levels of granularity, where fine-grained interleaving switches between threads very frequently, while coarse-grained only switches when there are long latency events like cache misses.

- **Lower complexity:**
  Generally considered a simpler implementation compared to SMT, requiring less hardware overhead.
  Key points about simultaneous multithreading (SMT):

- **Parallel instruction execution:**

  Multiple instructions from different threads can be issued and executed in the same processor cycle.

- **Requires superscalar architecture:**

  To effectively utilize SMT, a processor needs to be capable of handling multiple instructions per cycle.

- **Higher performance potential:**
  Can achieve significantly better performance than interleaved multithreading by exploiting more parallel processing opportunities.

Multithreading

What is a thread?

A thread is a single sequence stream within a process. Threads are also called **lightweight processes** as they possess some of the properties of processes. Each thread belongs to exactly one process.

The process of executing multiple tasks (also called threads) simultaneously is called multithreading. The primary purpose of multithreading is to provide simultaneous execution of two or more parts of a program to make maximum use of CPU time. A multithreaded program contains two or more parts that can run concurrently. It enables programmers to write in a way where multiple activities can proceed simultaneously within a single application.

In the context of computer architecture, "interleaved multithreading" refers to switching between different threads' instructions rapidly, executing a few instructions from one thread then switching to another, while "simultaneous multithreading" allows multiple threads to execute instructions concurrently, essentially running parts of different threads at the same time on a single core, utilizing the processor's resources more efficiently; essentially, interleaved is like taking turns between threads while simultaneous is like running multiple threads partially in parallel.

Key points to remember:

- **Interleaved multithreading:**
- Instructions from different threads are issued one after another, in a "round-robin" fashion.
- Considered a simpler approach, often used in processors with only one execution pipeline per core.
- May experience performance dips due to context switching overhead.
- **Simultaneous multithreading (SMT):**
- Multiple threads can issue instructions concurrently, potentially executing parts of different threads at the same time.
- Requires more complex hardware design to manage multiple thread contexts and ensure proper resource allocation.
- Can significantly improve overall CPU utilization by hiding latency and filling idle execution units.

  Example: Imagine a single processor with two logical cores enabled by SMT. With interleaved multithreading, the processor would execute a few instructions from Thread A, then switch to Thread B, and so on. With simultaneous multithreading,

it could execute some instructions from Thread A and some from Thread B at the same time, depending on available execution units

Superscalar Architecture

What is Superscalar Architecture?

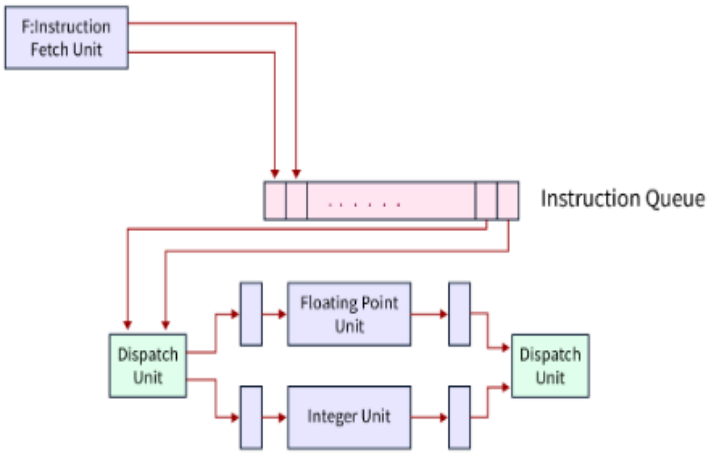
Superscalar architecture is an advanced design concept in modern microprocessor technology that aims to significantly enhance the processing power and efficiency of a CPU (Central Processing Unit). Unlike earlier scalar architectures, which execute one instruction per clock cycle, superscalar architectures can execute multiple instructions simultaneously, effectively achieving parallelism within a single processor.

The key feature of a superscalar processor is its ability to analyze and dispatch multiple instructions from a program in a single clock cycle, provided that these instructions are independent and can be executed concurrently. This is made possible through the inclusion of multiple execution units within the CPU, each responsible for handling specific types of instructions (e.g., arithmetic operations, memory access, branch instructions, etc.).

To efficiently manage the execution of instructions, superscalar processors also incorporate a sophisticated instruction scheduler or dispatcher. This scheduler is responsible for examining the incoming stream of instructions, identifying those that can be executed in parallel, and dispatching them to the available execution units.

Superscalar architectures have become vital in meeting the ever-increasing demands of modern computing tasks, such as multimedia processing, scientific simulations, and complex software applications. They offer the advantage of dramatically improved instruction throughput and overall performance, enabling processors to execute more instructions in a given period and achieve higher levels of computational efficiency.

Processor with two execution units



In the above diagram, there is a processor with two execution units; one for integer and one for floating point operations. The instruction fetch unit is capable of reading the instructions at a time and storing them in the instruction queue. In each cycle, the dispatch unit retrieves and decodes up to two instructions from the front of the queue. If there is one integer, one floating point instruction and no hazards, both the instructions are dispatched in the same clock cycle.

Advantages of Superscalar Architecture

Some of the key advantages offered by superscalar architecture are:

Increased Instruction Throughput: Its ability to execute multiple instructions concurrently, often in a single clock cycle. This results in a substantial increase in instruction throughput compared to scalar processors, which execute one instruction at a time.

Improved Performance: Superscalar processors excel in handling a wide range of tasks, from basic arithmetic operations to complex calculations and data manipulations. This improved performance is especially beneficial for applications that require extensive computational power, such as scientific simulations, 3D rendering, video encoding/decoding, and artificial intelligence tasks like deep learning.

Efficient Resource Utilization: Superscalar processors feature multiple execution units, each specialized in executing specific types of instructions (e.g., arithmetic, memory access, control flow, etc.). This allows for efficient resource utilization, as the CPU can allocate and execute instructions optimizing the use of available hardware resources.

Parallelism Exploitation: Superscalar architectures leverage instruction-level parallelism (ILP), which enables the concurrent execution of independent instructions. The CPU's instruction scheduler identifies and dispatches these independent instructions to different

execution units simultaneously. This parallelism significantly reduces the time needed to complete a task and maximizes CPU utilization.

Out-of-Order Execution: Many superscalar processors incorporate out-of-order execution, a feature that further enhances performance. In out-of-order execution, instructions are executed as soon as their dependencies are satisfied, rather than strictly following the sequential order of the program. This reduces pipeline stalls and keeps the CPU's execution units busy.

Flexibility and Compatibility: Superscalar processors are highly versatile and compatible with a wide range of software applications and programming languages. They can execute both legacy single-threaded programs and modern multi-threaded applications efficiently.

Scalability: Superscalar architecture can be scaled to accommodate different levels of complexity and performance requirements. Chip designers can add more execution units or improve existing ones to create processors tailored to specific needs, from low-power mobile devices to high-performance server CPUs.

Energy Efficiency: While superscalar processors are known for their performance, they have also made strides in energy efficiency. Some superscalar architectures incorporate power-saving features like dynamic voltage and frequency scaling (DVFS) and clock gating, which help reduce power consumption during periods of low computational demand.

Support for Advanced Compiler Techniques: Superscalar processors work hand in hand with advanced compiler techniques that can identify and schedule instructions for parallel execution. Compiler optimizations like loop unrolling, software pipelining, and instruction scheduling can further enhance the performance of superscalar processors.

Handling Complex Branching: Superscalar architectures often include advanced branch prediction mechanisms to minimize the performance impact of conditional branches. Efficient handling of branch instructions is crucial because incorrect branch predictions can lead to pipeline stalls and reduced throughput. By predicting branches accurately, superscalar processors maintain a high instruction throughput even in the presence of branching code.

Disadvantages of Superscalar Architecture:

- In a Superscalar Processor, the detrimental effect on performance of various hazards becomes even more pronounced.
- Due to this type of architecture, problem in scheduling can occur.

Processor interconnection

A "processor interconnection" refers to the system of physical connections that allows multiple processors within a computer system to communicate with each other and access shared memory, essentially enabling data transfer between different processing units by utilizing dedicated pathways or network structures like buses, crossbar switches, or dedicated network links depending on the architecture.

The principles of processor interconnection primarily focus on designing a network that efficiently facilitates communication between multiple processors by considering factors like topology (connection pattern), routing algorithms, bandwidth, latency, and switching mechanisms, aiming to optimize data transfer speed while balancing cost and power consumption across the system; key principles include: choosing an appropriate topology based on performance requirements, managing traffic flow with efficient routing algorithms, minimizing latency through optimized switching techniques, and considering scalability for large systems; all while balancing these factors to achieve the best possible performance for the specific application.

Key aspects of processor interconnection principles:

- **Topology:**

  The physical layout of connections between processors, including options like mesh, hypercube, tree, bus, ring, or crossbar, each with its own advantages and drawbacks based on factors like distance, bandwidth, and scalability.

- **Routing algorithms:**

  The strategy used to determine the path data packets take to reach their destination, with considerations for minimizing hops, avoiding congestion, and ensuring deadlock-free routing.

- **Switching mechanisms:**

  How data is transferred between different links within the network, including options like circuit switching (dedicated path), packet switching (data divided into packets), and wormhole routing (data forwarded as it arrives).

- **Bandwidth:**

  The maximum data transfer rate across the network, crucial for high-performance computing applications.

- **Latency:**

  The time it takes for a data packet to travel from one processor to another, a significant factor in determining overall system responsiveness.

- **Scalability:**

The ability to add more processors to the network without significantly impacting performance, important for large-scale systems.
Important considerations when designing processor interconnects:

- **Application requirements:**

  Understanding the specific needs of the application, such as the expected communication patterns, data size, and performance demands.

- **Cost-performance trade-offs:**

  Balancing the cost of the interconnect hardware with the desired performance level.

- **Power consumption:**

  Minimizing power usage while maintaining adequate performance, especially in high-density computing systems.

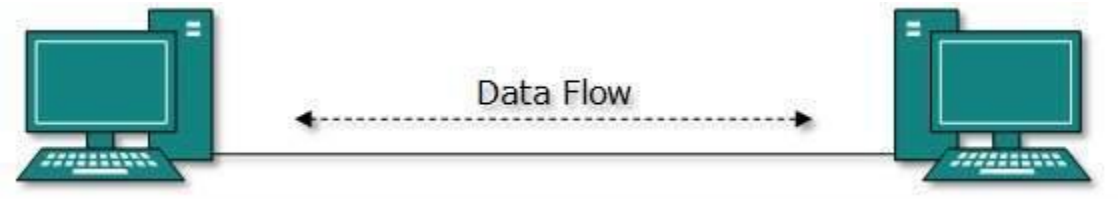
- **Reliability and fault tolerance:**
  Designing mechanisms to handle potential failures and ensure system robustness.

Computer Network Topologies

A Network Topology is the arrangement with which computer systems or network devices are connected to each other. Topologies may define both physical and logical aspect of the network. Both logical and physical topologies could be same or different in a same network.

Point-to-Point

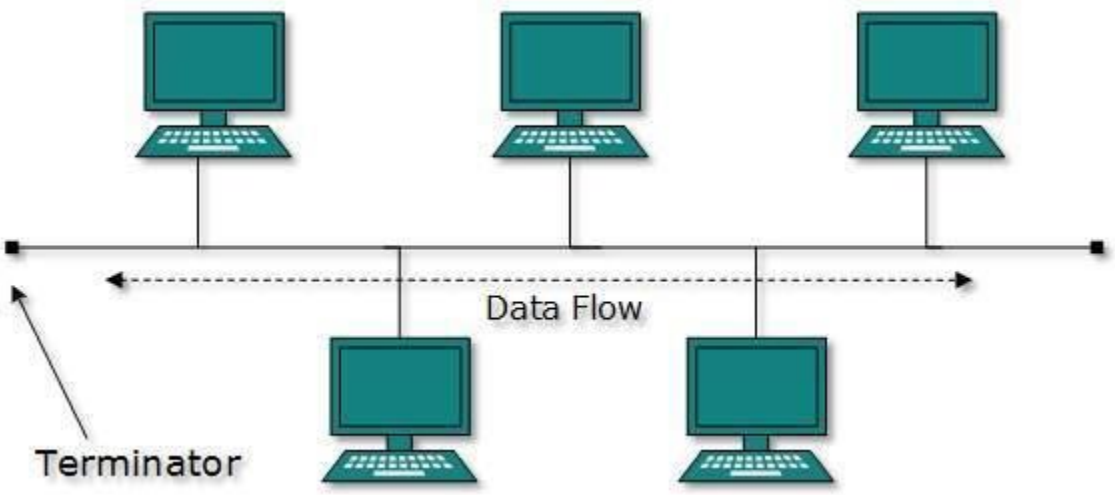
Point-to-point networks contains exactly two hosts such as computer, switches or routers, servers connected back to back using a single piece of cable. Often, the receiving end of one host is connected to sending end of the other and vice-versa.



If the hosts are connected point-to-point logically, then may have multiple intermediate devices. But the end hosts are unaware of underlying network and see each other as if they are connected directly.

Bus Topology

In case of Bus topology, all devices share single communication line or cable.Bus topology may have problem while multiple hosts sending data at the same time. Therefore, Bus topology either uses CSMA/CD technology or recognizes one host as Bus Master to solve the issue. It is one of the simple forms of networking where a failure of a device does not affect the other devices. But failure of the shared communication line can make all other devices stop functioning.

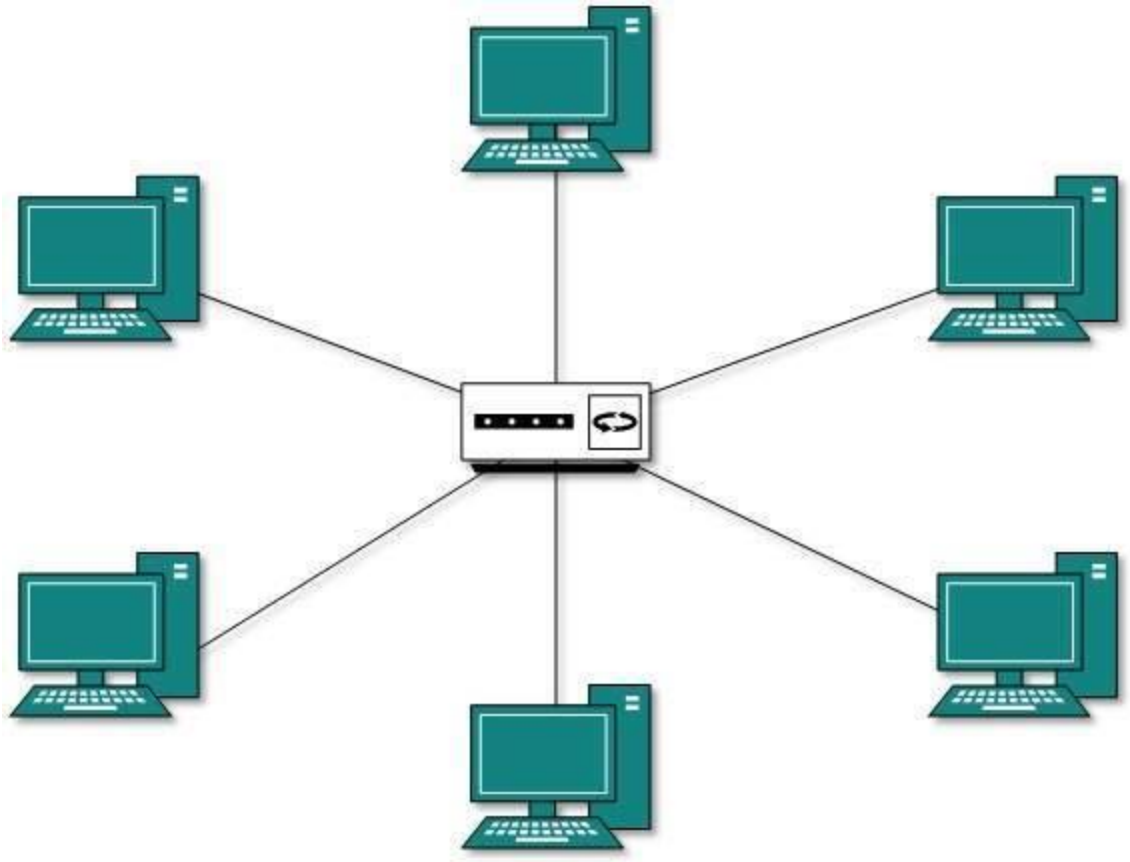


Both ends of the shared channel have line terminator. The data is sent in only one direction and as soon as it reaches the extreme end, the terminator removes the data from the line.

Star Topology

All hosts in Star topology are connected to a central device, known as hub device, using a point-to-point connection. That is, there exists a point to point connection between hosts and hub. The hub device can be any of the following:

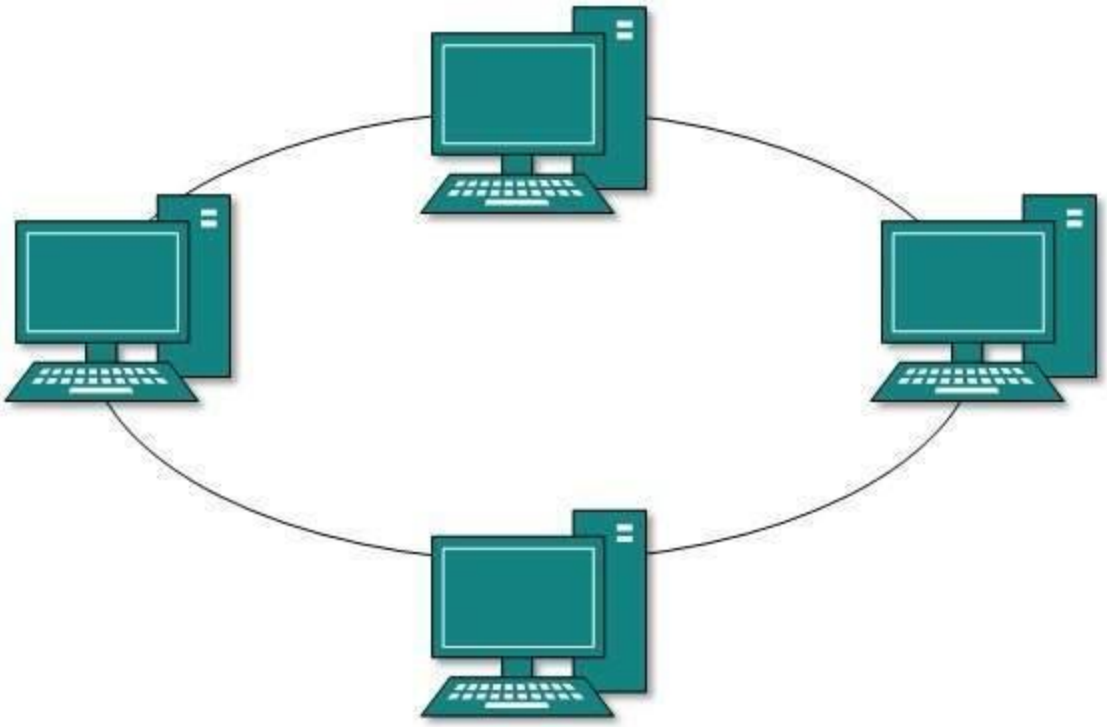
- Layer-1 device such as hub or repeater
- Layer-2 device such as switch or bridge
- Layer-3 device such as router or gateway



As in Bus topology, hub acts as single point of failure. If hub fails, connectivity of all hosts to all other hosts fails. Every communication between hosts, takes place through only the hub.Star topology is not expensive as to connect one more host, only one cable is required and configuration is simple.

Ring Topology

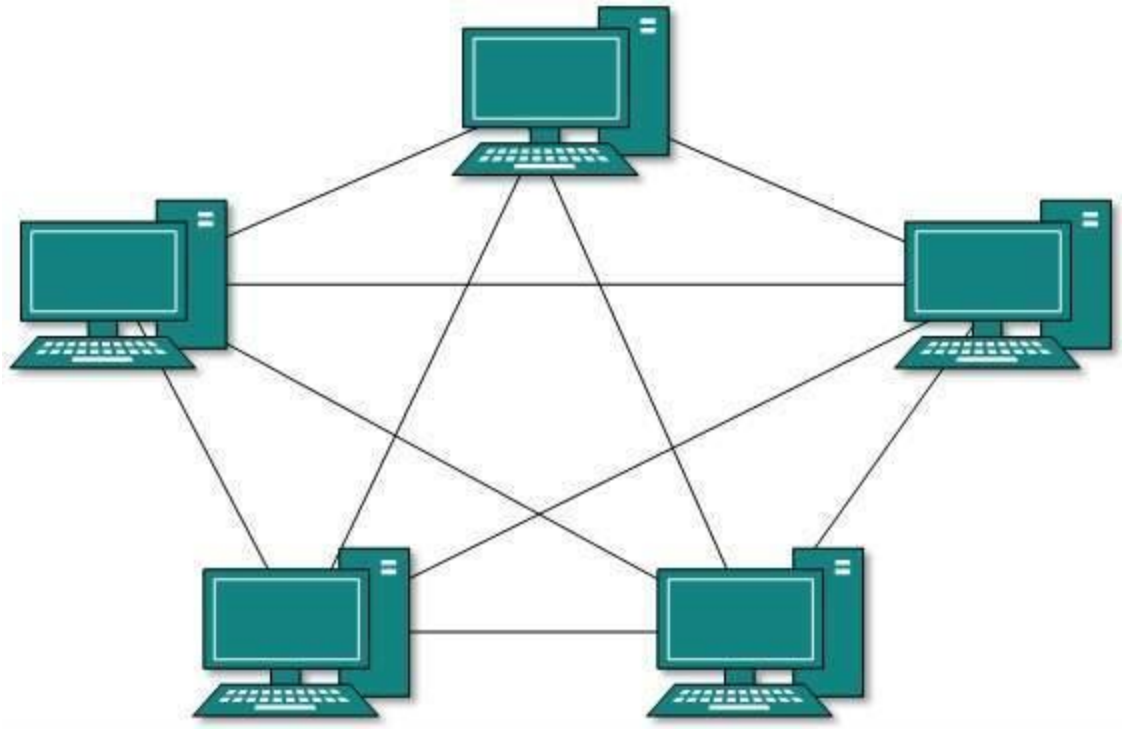
In ring topology, each host machine connects to exactly two other machines, creating a circular network structure. When one host tries to communicate or send message to a host which is not adjacent to it, the data travels through all intermediate hosts. To connect one more host in the existing structure, the administrator may need only one more extra cable.



Failure of any host results in failure of the whole ring.Thus, every connection in the ring is a point of failure. There are methods which employ one more backup ring.

Mesh Topology

In this type of topology, a host is connected to one or multiple hosts.This topology has hosts in point-to-point connection with every other host or may also have hosts which are in point-to-point connection to few hosts only.



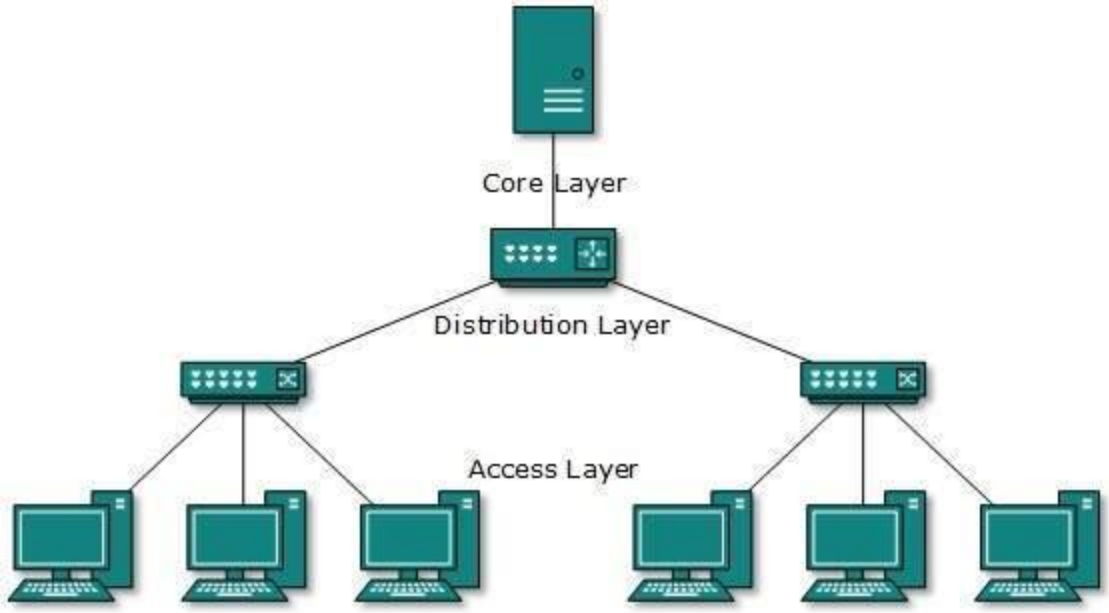
Hosts in Mesh topology also work as relay for other hosts which do not have direct point-to-point links. Mesh technology comes into two types:

- **Full Mesh**: All hosts have a point-to-point connection to every other host in the network. Thus for every new host n(n-1)/2 connections are required. It provides the most reliable network structure among all network topologies.
- **Partially Mesh**: Not all hosts have point-to-point connection to every other host. Hosts connect to each other in some arbitrarily fashion. This topology exists where we need to provide reliability to some hosts out of all.

Tree Topology

Also known as Hierarchical Topology, this is the most common form of network topology in use presently.This topology imitates as extended Star topology and inherits properties of bus topology.

This topology divides the network in to multiple levels/layers of network. Mainly in LANs, a network is bifurcated into three types of network devices. The lowermost is access-layer where computers are attached. The middle layer is known as distribution layer, which works as mediator between upper layer and lower layer. The highest layer is known as core layer, and is central point of the network, i.e. root of the tree from which all nodes fork.



All neighboring hosts have point-to-point connection between them.Similar to the Bus topology, if the root goes down, then the entire network suffers even.though it is not the single point of failure. Every connection serves as point of failure, failing of which divides the network into unreachable segment.

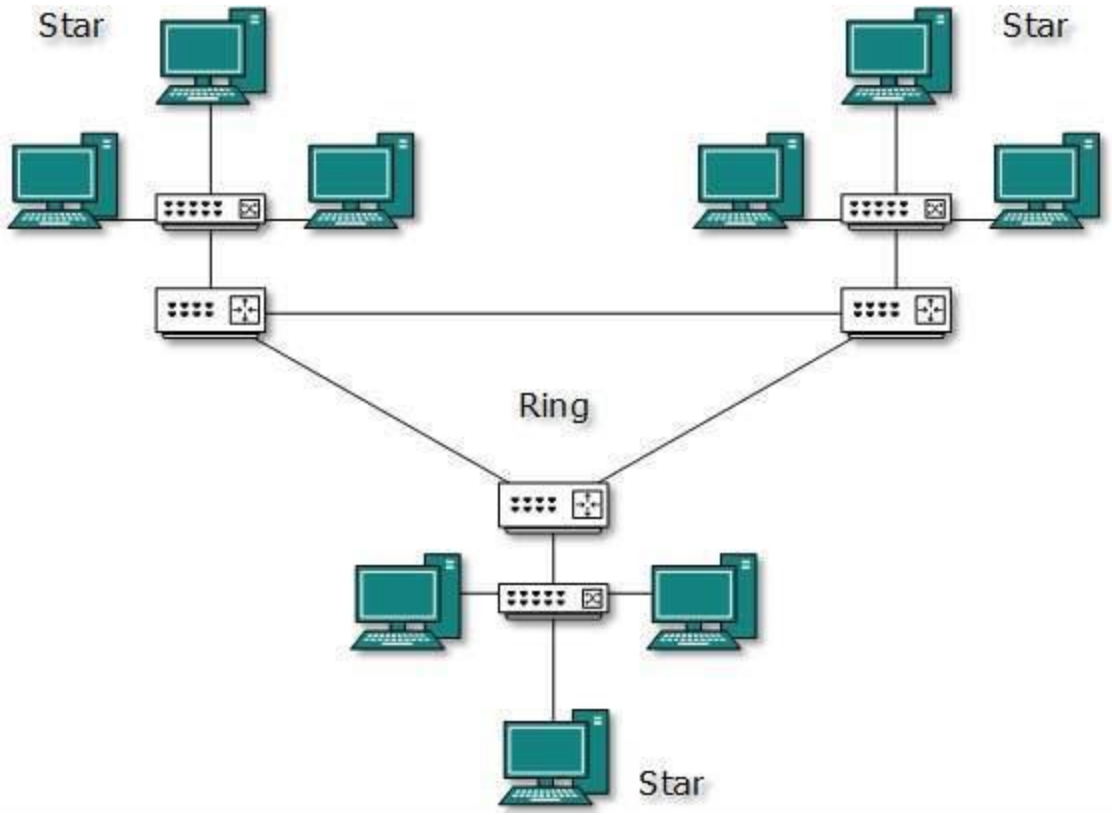
Daisy Chain

This topology connects all the hosts in a linear fashion. Similar to Ring topology, all hosts are connected to two hosts only, except the end hosts. Means, if the end hosts in daisy chain are connected then it represents Ring topology.

Each link in daisy chain topology represents single point of failure. Every link failure splits the network into two segments. Every intermediate host works as relay for its immediate hosts.

Hybrid Topology

A network structure whose design contains more than one topology is said to be hybrid topology. Hybrid topology inherits merits and demerits of all the incorporating topologies.



The above picture represents an arbitrarily hybrid topology. The combining topologies may contain attributes of Star, Ring, Bus, and Daisy-chain topologies. Most WANs are connected by means of Dual-Ring topology and networks connected to them are mostly Star topology networks. Internet is the best example of largest Hybrid topology

Multicore and Multiprocessor Systems

In today's tech world, multi-core and multi-processor systems have become essential for boosting computing power and efficiency. A multicore system packs several processing units or cores into a single chip allowing it to tackle multiple tasks at once. Multiprocessor system uses two or more separate processors that share resources enabling them to work together seamlessly. This means that if one processor runs into trouble the others can keep things running smoothly. Both systems are designed to meet the increasing demands of modern applications with multicore systems focusing on maximizing performance within a single chip and multiprocessor systems expanding capabilities through multiple CPUs.

What is a Multicore System?

A processor that has more than one core is called a Multicore Processor while one with a single core is called Unicore Processor or Uniprocessor. Nowadays, most systems have four cores (Quad-core) or eight cores (Octa-core). These cores can

individually read and execute program instructions, giving feel like a computer system has several processors but in reality, they are cores and not processors. Instructions can be calculation, data transferring instruction, branch instruction, etc. Processors can run instructions on separate cores at the same time. This increases the overall speed of program execution in the system. Thus heat generated by the processor gets reduced and increases the overall speed of execution.

Multicore systems support **Multithreading** and **Parallel Computing.** Multicore processors are widely used across many application domains, including general-purpose, embedded, network, digital signal processing (DSP), and graphics (GPU). Efficient software algorithms should be used for the implementation of cores to achieve higher performance. Software that can run parallel is preferred because we want to achieve parallel execution with the help of multiple cores.

Advantages of Multicore System

- These cores are usually integrated into single IC (integrated circuit) die, or onto multiple dies but in single chip package. Thus allowing higher Cache Coherency.
- These systems are energy efficient since they allow higher performance at lower energy. A challenge in this, however, is additional overhead of writing parallel code.
- It will have less traffic(cores integrated into single chip and will require less time).

Disadvantages of Multicore System

- Dual-core processor do not work at twice speed of single processor. They get only 60-80% more speed.
- Some Operating systems are still using single core processor.
- OS compiled for multi-core processor will run slightly slower on single-core processor.

What is MultiProcessor System?

Two or more processors or CPUs present in same computer, sharing system bus, memory and I/O is called MultiProcessing System. It allows parallel execution of different processors. These systems are reliable since failure of any single processor does not affect other processors. A quad-processor system can execute four processes at a time while an octa-processor can execute eight processes at a time. The memory and other resources may be shared or distributed among processes.

Advantages of MultiProcessor System

- Since more than one processor are working at the same time, throughput will get increased.
- More reliable since failure in one CPU does not affect other.
- It needs little complex configuration.
- Parallel processing (more than one process executing at same time) is achieved through MultiProcessing.

Disadvantages of Multiprocessor System

- It will have more traffic (distances between two will require longer time).
- Throughput may get reduced in shared resources system where one processor using some I/O then another processor has to wait for its turn.
- As more than processors are working at particular instant of time. So, coordination between these is very complex.

Difference between Multicore and Multiprocessor System

MultiCore	MultiProcessor
A single CPU or processor with two or more independent processing units called cores that are capable of reading and executing program instructions.	A system with two or more CPU's that allows simultaneous processing of programs.
It executes single program faster.	It executes multiple programs Faster.
Not as reliable as multiprocessor.	More reliable since failure in one CPU will not affect other.
It has less traffic.	It has more traffic.
It does not need to be configured.	It needs little complex configuration.
It's very cheaper (single CPU that does not require multiple CPU support system).	It is Expensive (Multiple separate CPU's that require system that supports multiple processors) as compared to MultiCore.

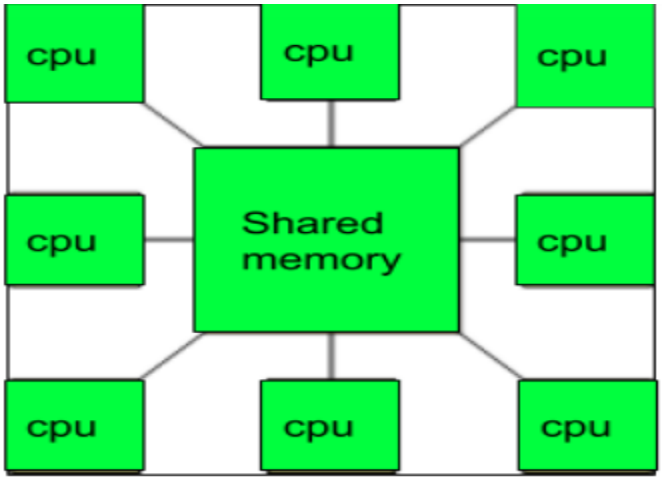
Conclusion

The primary goal of both **multicore** and **multiprocessor systems** is to enhance processing speed but they achieve this in different ways. Multicore systems are generally more costeffective because they integrate multiple cores within a single

processor whereas multiprocessor systems require several physical processors driving up costs. If you're running a single program a multicore system will typically perform faster due to its efficient architecture. However if you need to run multiple programs simultaneously multiprocessor system shines by distributing the workload across several processors. In modern computing it's common to find computers equipped with multiple **CPUs** each containing several cores effectively combining the strengths of both architectures to meet diverse processing needs.

Multiprocessor systems

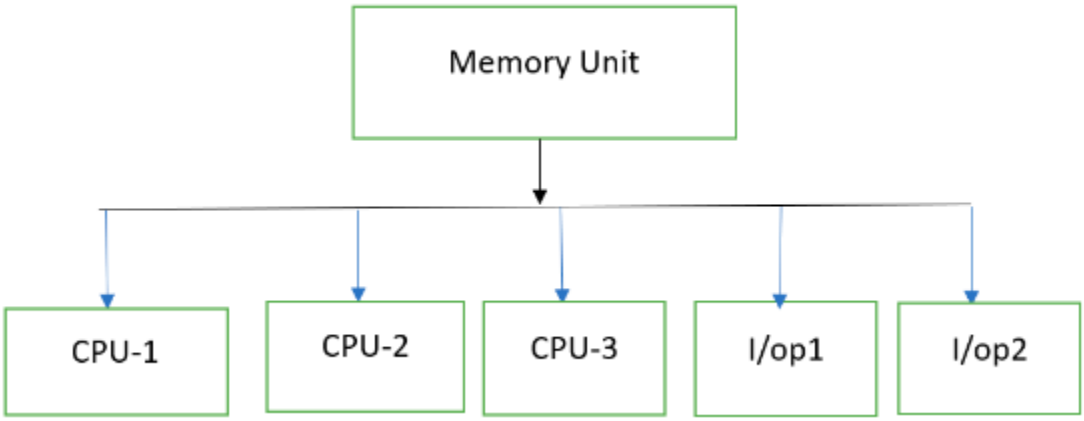
Multiprocessor: A Multiprocessor is a computer system with two or more central processing units (CPUs) share full access to a common RAM. The main objective of using a multiprocessor is to boost the system's execution speed, with other objectives being fault tolerance and application matching. There are two types of multiprocessors, one is called shared memory multiprocessor and another is distributed memory multiprocessor. In shared memory multiprocessors, all the CPUs shares the common memory but in a distributed memory multiprocessor, every CPU has its own private memory.



The interconnection among two or more processor and shared memory is done with three methods

1) Time shared common bus
2) Multiport memories
3) Crossbar switch network

1) Time shared common bus



As the name itself indicates, int this method is contains a single shared bus through which all processor & memory unit can be communicated.
Consider CPU-1 is interacting with memory unit using common shared bud in that case all other processor must be idle as we have only one bus to communicate.

Advantage:

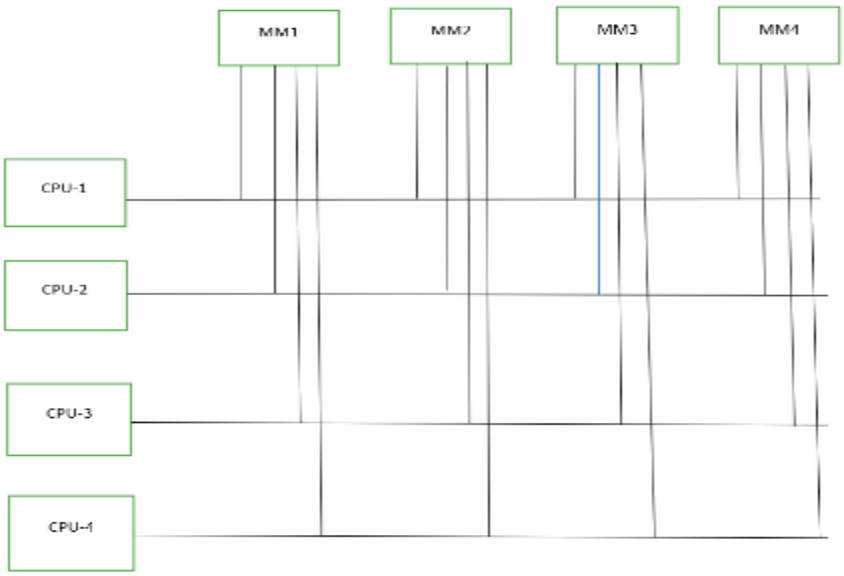
- Simple to implement.
- Due to single common bus cost to implement is very less.
-

Disadvantage:

- Data transfer rate is slow**.**

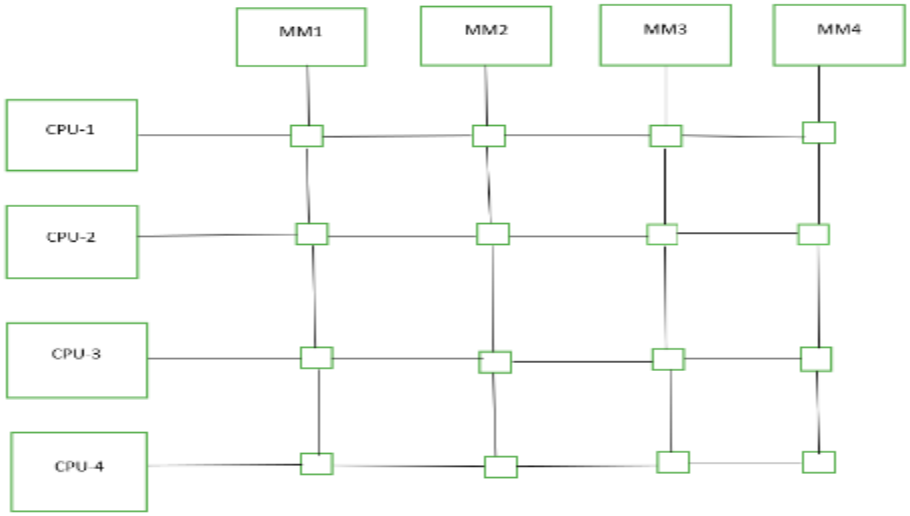
2) Multiport memories

Unlike in the shared common bus method, hence it contains separate bus for each processor to communicate with the memory module.
Suppose CPU-1 wants to interact with memory module 1 then port mm1 is enabled. Similarly CPU-4 wants to
to interact with memory module 4 then port mm4 is enabled. Hence all the process can be communicated parallelly. If more than one CPU request for same time memory module, priority will be given in the order of CPU-1,CPU-2,CPU-3,CPU-4.



3) Crossbar switch network

Here instead multiport unlike in multiport memories, a switch will be installed between memory unit and CPU. Switch is responsible for whether to pass the request to a particular memory module or not based on the request made for.



Advantage:

- High data through rate.

Disadvantage:

- **Complex to implement as more switches involved.**
- Costlier to implement.

Applications of Multiprocessor –

1. As a uniprocessor, such as single instruction, single data stream (SISD).
2. As a multiprocessor, such as single instruction, multiple data stream (SIMD), which is usually used for vector processing.
3. Multiple series of instructions in a single perspective, such as multiple instruction, single data stream (MISD), which is used for describing hyper-threading or pipelined processors.
4. Inside a single system for executing multiple, individual series of instructions in multiple perspectives, such as multiple instruction, multiple data stream (MIMD).

Benefits of using a Multiprocessor –

- Enhanced performance.
- Multiple applications.
- Multi-tasking inside an application.
- High throughput and responsiveness.
- Hardware sharing among CPUs.

Advantages:

Improved performance: Multiprocessor systems can execute tasks faster than single-processor systems, as the workload can be distributed across multiple processors.

Better scalability: Multiprocessor systems can be scaled more easily than single-processor systems, as additional processors can be added to the system to handle increased workloads.

Increased reliability: Multiprocessor systems can continue to operate even if one processor fails, as the remaining processors can continue to execute tasks.

Reduced cost: Multiprocessor systems can be more cost-effective than building multiple single-processor systems to handle the same workload.

Enhanced parallelism: Multiprocessor systems allow for greater parallelism, as different processors can execute different tasks simultaneously.

Disadvantages:

Increased complexity: Multiprocessor systems are more complex than single-processor systems, and they require additional hardware, software, and management resources.

Higher power consumption: Multiprocessor systems require more power to operate than single-processor systems, which can increase the cost of operating and maintaining the system.

Difficult programming: Developing software that can effectively utilize multiple processors can be challenging, and it requires specialized programming skills.

Synchronization issues: Multiprocessor systems require synchronization between processors to ensure that tasks are executed correctly and efficiently, which can add complexity and overhead to the system.
Limited performance gains: Not all applications can benefit from multiprocessor systems, and some applications may only see limited performance gains when running on a multiprocessor system.

Graphics Processing Unit (GPUs)

A GPU, or Graphics Processing Unit, is a circuit that processes graphics-related tasks like video editing, 3D rendering, and machine learning. GPUs are built on silicon wafers and use thousands of tiny transistors to perform calculations in parallel.

Graphics processing technology has evolved to deliver unique benefits in the world of computing. The latest graphics processing units (GPUs) unlock new possibilities in gaming, content creation, machine learning, and more.

What Does a GPU Do?

The graphics processing unit, or GPU, has become one of the most important types of computing technology, both for personal and business computing. Designed for parallel processing, the GPU is used in a wide range of applications, including graphics and video rendering. Although they're best known for their capabilities in gaming, GPUs are becoming more popular for use in creative production and artificial intelligence (AI).

GPUs were originally designed to accelerate the rendering of 3D graphics. Over time, they became more flexible and programmable, enhancing their capabilities. This allowed graphics programmers to create more interesting visual effects and realistic scenes with advanced lighting and shadowing techniques. Other developers also began to tap the power of GPUs to dramatically accelerate additional workloads in high performance computing (HPC), deep learning, and more.

GPU and CPU: Working Together

The GPU evolved as a complement to its close cousin, the CPU (central processing unit). While CPUs have continued to deliver performance increases through architectural innovations, faster clock speeds, and the addition of cores, GPUs are specifically designed to accelerate computer graphics workloads. When shopping for a system, it can be helpful to know the role of the CPU vs. GPU so you can make the most of both.

GPU vs. Graphics Card: What's the Difference?

While the terms GPU and graphics card (or video card) are often used interchangeably, there is a subtle distinction between these terms. Much like a motherboard contains a

CPU, a graphics card refers to an add-in board that incorporates the GPU. This board also includes the raft of components required to both allow the GPU to function and connect to the rest of the system.

GPUs come in two basic types: integrated and discrete. An integrated GPU does not come on its own separate card at all and is instead embedded alongside the CPU. A discrete GPU is a distinct chip that is mounted on its own circuit board and is typically attached to a PCI Express slot.

Integrated Graphics Processing Unit

The majority of GPUs on the market are actually integrated graphics. So, what are integrated graphics and how does it work in your computer? A CPU that comes with a fully integrated GPU on its motherboard allows for thinner and lighter systems, reduced power consumption, and lower system costs.

Intel® Graphics Technology, which includes Intel® Iris® Xᵉ graphics at the forefront of integrated graphics technology. With Intel® Graphics, users can experience immersive graphics in systems that run cooler and deliver long battery life.

Discrete Graphics Processing Unit

Many computing applications can run well with integrated GPUs. However, for more resource-intensive applications with extensive performance demands, a discrete GPU (sometimes called a dedicated graphics card) is better suited to the job.

These GPUs add processing power at the cost of additional energy consumption and heat creation. Discrete GPUs generally require dedicated cooling for maximum performance.

What Are GPUs Used For?

Two decades ago, GPUs were used primarily to accelerate real-time 3D graphics applications, such as games. However, as the 21st century began, computer scientists realized that GPUs had the potential to solve some of the world's most difficult computing problems.

This realization gave rise to the general purpose GPU era. Now, graphics technology is applied more extensively to an increasingly wide set of problems. Today's GPUs are more programmable than ever before, affording them the flexibility to accelerate a broad range of applications that go well beyond traditional graphics rendering.

GPUs for Gaming

Video games have become more computationally intensive, with hyperrealistic graphics and vast, complicated in-game worlds. With advanced display technologies, such as 4K

screens and high refresh rates, along with the rise of virtual reality gaming, demands on graphics processing are growing fast. GPUs are capable of rendering graphics in both 2D and 3D. With better graphics performance, games can be played at higher resolution, at faster frame rates, or both.

GPUs for Video Editing and Content Creation

For years, video editors, graphic designers, and other creative professionals have struggled with long rendering times that tied up computing resources and stifled creative flow. Now, the parallel processing offered by GPUs—along with built-in AI capabilities and advanced acceleration— makes it faster and easier to render video and graphics in higher-definition formats.

Uniting fluid gaming experiences with the latest in visual technologies, Intel® Arc™ graphics for desktops enables captivating content immersion. Intel Arc graphics cards include built-in machine learning, graphics acceleration, and ray-tracing hardware with scalable performance options for laptops, desktops, and professional workstations.

Create rich digital content augmented by AI and accelerated by Intel® Deep Link Technology. Take advantage of Intel® Arc™ Control's virtual streaming technologies to entertain your viewers and enhance your livestream. Or take your gaming experience to the next level with Intel Xe Super Sampling's AI-enhanced upscaling. Intel® Arc™ A-series graphics offers these advanced graphics technologies to power a premium laptop experience for portable immersive gaming and content creation.

For professional content creation, Intel® Arc™ Pro A-series graphics for mobile, desktop, and workstations improves on the performance and memory bandwidth of Intel Arc graphics cards with single-slot, dual-slot, and mobile workstation form-factors. These cards accelerate the creation of beautiful graphics with ray tracing hardware technology and support multiple large displays with Ultra High Definition (UHD), ultrawide UHD, and high dynamic range (HDR).

GPU for Machine Learning

Some of the most exciting applications for GPU technology involve AI and machine learning. Because GPUs incorporate an extraordinary amount of computational capability, they can deliver incredible acceleration in workloads that take advantage of the highly parallel nature of GPUs, such as image recognition. Many of today's deep learning technologies rely on GPUs working with CPUs.

FPGA vs. GPU for Deep Learning ›

Intel® GPU Technologies

Intel has long been a leader in graphics processing technology, especially when it comes to PCs. Most recently, the Intel® Arc™ A-series graphics bring a new level of

advanced technologies to desktops and laptops, with built-in machine learning, graphics acceleration, and ray tracing.

The Intel® Iris® Xᵉ graphics and Intel® UHD Graphics that are integrated into our 13th Gen Intel® Core™ processors support 4K HDR, 1080p gaming, and other rich visual experiences for desktops. For laptop users, Intel also offers the Intel® Iris® Xᵉ MAX graphics.

GPUs in the Data Center

In the data center, Intel supports amazing visual experiences with integrated graphics in Intel® Xeon® processors.

For today's most complex data center problems, the Intel® Data Center GPU Flex Series and Intel® Data Center GPU Max Series enable powerful and efficient GPU performance. Data center GPUs can offer better support for parallel operations, AI, media, media analytics, and 3D rendering solutions. This makes GPUs essential for advanced use cases such as machine learning, modeling, or 3D rendering for cloud gaming and other content offerings.

RAID (Redundant Arrays of Independent Disks)

RAID (Redundant Arrays of Independent Disks) is a technique that makes use of a combination of multiple disks for storing the data instead of using a single disk for increased performance, data redundancy, or to protect data in the case of a drive failure. The term was defined by David Patterson, Garth A. Gibson, and Randy Katz at the University of California, Berkeley in 1987. In this article, we are going to discuss RAID and types of RAID their Advantages and disadvantages in detail.

What is RAID?

RAID (Redundant Array of Independent Disks) is like having backup copies of your important files stored in different places on several hard drives or solid-state drives (SSDs). If one drive stops working, your data is still safe because you have other copies stored on the other drives. It's like having a safety net to protect your files from being lost if one of your drives breaks down.

RAID (Redundant Array of Independent Disks) in a Database Management System (DBMS) is a technology that combines multiple physical disk drives into a single logical unit for data storage. The main purpose of RAID is to improve data reliability, availability, and performance. There are different levels of RAID, each offering a balance of these benefits.

How RAID Works?

Let's understand how RAID (Redundant Array of Independent Disks) works through a relatable example:

Imagine you have a favorite book that you want to keep safe. Instead of giving the entire book to just one friend for safekeeping, you take a smart approach:

1. **Splitting the Book:** You divide the book into smaller pieces (like chapters or sections) and give each piece to a different friend.
2. **Making Copies:** For extra security, you might also create duplicate pieces and give them to multiple friends.

Now, if one friend misplaces their piece, you can still recreate the entire book using the pieces held by the others. This way, your book is safe even if someone loses their portion.

This is exactly how RAID works with hard drives! RAID splits your data across multiple drives (similar to dividing the book into pieces). Depending on the RAID configuration, it may also create duplicates (like making extra copies). If one drive fails, the remaining drives can help reconstruct the lost data.

What is a RAID Controller?

A RAID controller is like a boss for your hard drives in a big storage system. It works between your computer's operating system and the actual hard drives, organizing them into groups to make them easier to manage. This helps speed up how fast your computer can read and write data, and it also adds a layer of protection in case one of your hard drives breaks down. So, it's like having a smart helper that makes your hard drives work better and keeps your important data safer.

Types of RAID Controller

There are three types of RAID controller:

Hardware Based: In hardware-based RAID, there's a physical controller that manages the whole array. This controller can handle the whole group of hard drives together. It's designed to work with different types of hard drives, like SATA (Serial Advanced Technology Attachment) or SCSI (Small Computer System Interface). Sometimes, this controller is built right into the computer's main board, making it easier to set up and manage your RAID system. It's like having a captain for your team of hard drives, making sure they work together smoothly**.**

Software Based: In software-based RAID, the controller doesn't have its own special hardware. So it use computer's main processor and memory to do its job. It perform the same function as a hardware-based RAID controller, like managing the hard drives and keeping your data safe. But because it's sharing resources with other programs on your computer, it might not make things run as fast. So, while it's still helpful, it might not give you as big of a speed boost as a hardware-based RAID system

Firmware Based: Firmware-based RAID controllers are like helpers built into the computer's main board. They work with the main processor, just like software-based RAID. But they only implement when the computer starts up. Once the operating system is running, a special driver takes over the RAID job. These controllers aren't as expensive as hardware ones, but they make the computer's main processor work

harder. People also call them hardware-assisted software RAID, hybrid model RAID, or fake RAID.

Why Data Redundancy?

Data redundancy, although taking up extra space, adds to disk reliability. This means, that in case of disk failure, if the same data is also backed up onto another disk, we can retrieve the data and go on with the operation. On the other hand, if the data is spread across multiple disks without the RAID technique, the loss of a single disk can affect the entire data.

Key Evaluation Points for a RAID System

When evaluating a RAID system, the following critical aspects should be considered:

1. Reliability

Definition: Refers to the system's ability to tolerate disk faults and prevent data loss.

Example:

- RAID 0 offers no fault tolerance; if one disk fails all data is lost.
- RAID 5 can tolerate one disk failure due to parity data.
- RAID 6 can handle two simultaneous disk failures.

2. Availability

Definition: The fraction of time the RAID system is operational and available for use.

Example:

- RAID 1 (Mirroring) allows immediate data access even during a single disk failure.
- RAID 5 and 6 may degrade performance during a rebuild, but data remains available.

3. Performance

Definition: Measures how efficiently the RAID system handles data processing tasks. This includes:

- **Response Time:** How quickly the system responds to data requests.
- **Throughput:** The rate at which the system processes data (e.g., MB/s or IOPS).

Key Factors:

- RAID levels affect performance differently:
- RAID 0 offers high throughput but no redundancy.
- RAID 1 improves read performance by serving data from either mirrored disk but may not improve write performance significantly.
- RAID 5/6 introduces overhead for parity calculations, affecting write speeds.
- Workload type (e.g., sequential vs. random read/write operations).

Performance Trade-offs: Higher redundancy often comes at the cost of slower writes (due to parity calculations).

4. Capacity

Definition: The amount of usable storage available to the user after accounting for redundancy mechanisms.

Key Calculation: For a set of N disks, each with B blocks, the available capacity depends on the RAID level:

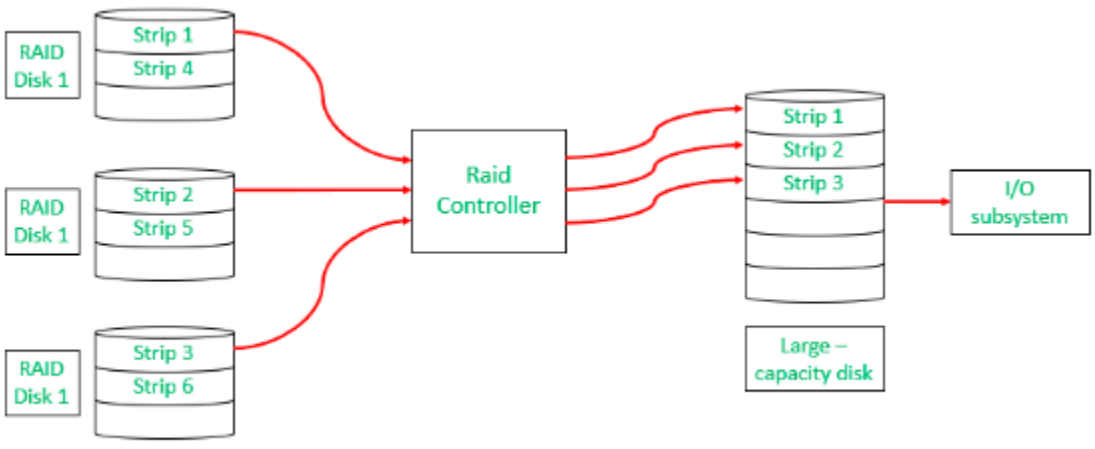
- **RAID 0:** All N×B blocks are usable (no redundancy).
- **RAID 1:** Usable capacity is B (only one disk's capacity due to mirroring).
- **RAID 5:** Usable capacity is (N−1)×B (one disk's worth of capacity used for parity).
- **RAID 6:** Usable capacity is (N−2)×B (two disks' worth used for parity).

Trade-offs: Higher redundancy (RAID 5/6) reduces available capacity compared to non-redundant setups (RAID 0).

RAID is very transparent to the underlying system. This means, that to the host system, it appears as a single big disk presenting itself as a linear array of blocks. This allows older technologies to be replaced by RAID without making too many changes to the existing code.

Different RAID Levels

- RAID-0 (Stripping)
- RAID-1 (Mirroring)
- RAID-2 (Bit-Level Stripping with Dedicated Parity)
- RAID-3 (Byte-Level Stripping with Dedicated Parity)
- RAID-4 (Block-Level Stripping with Dedicated Parity)
- RAID-5 (Block-Level Stripping with Distributed Parity)
- RAID-6 (Block-Level Stripping with two Parity Bits)



Raid Controller

1. RAID-0 (Stripping)

- RAID-0 improves system performance by splitting data into smaller “blocks” and spreading them across multiple disks. This process is called “striping.” It enhances data access speed by enabling parallel read/write operations but provides no redundancy or fault tolerance.

RAID-0

- A set of blocks distributed across the disks forms a “stripe.” For instance, “0, 1, 2, 3” is one stripe.
- Instead of placing just one block into a disk at a time, we can work with two (or more) blocks placed into a disk before moving on to the next one.

Evaluation

- **Reliability:** 0
  There is no duplication of data. Hence, a block once lost cannot be recovered.
- **Capacity:** N*B
  The entire space is being used to store data. Since there is no duplication, N disks each having B blocks are fully utilized.

Advantages

- It is easy to implement.
- It utilizes the storage capacity in a better way.

Disadvantages

- A single drive loss can result in the complete failure of the system.
- It’s not a good choice for a critical system.

2. RAID-1 (Mirroring)

- RAID-1 enhances reliability by creating an identical copy (mirror) of each data block on separate disks. This ensures that even if one disk fails, the data remains accessible from its duplicate. While this configuration is highly reliable, it requires significant storage overhead.

Mirroring:
Each block of data is written to two (or more) disks.
For example:
- Block 0 is stored on Disk 0 and its duplicate on Disk 1.
- Block 1 is stored on Disk 2 and its duplicate on Disk 3.

Mirroring Level 2:
In the figure, “mirroring level 2” indicates that there are two copies of each block, distributed across different disks.
RAID 0 was unable to tolerate any disk failure. But RAID 1 is capable of reliability.
Read about Difference Between RAID 0 and RAID 1.
Evaluation
Assume a RAID system with mirroring level 2.
- **Reliability:** 1 to N/2
  1 disk failure can be handled for certain because blocks of that disk would have duplicates on some other disk. If we are lucky enough and disks 0 and 2 fail, then again this can be handled as the blocks of these disks have duplicates on disks 1 and 3. So, in the best case, N/2 disk failures can be handled.
- **Capacity:** N*B/2
  Only half the space is being used to store data. The other half is just a mirror of the already stored data.

Advantages
- It covers complete redundancy.
- It can increase data security and speed.

Disadvantages
- It is highly expensive.
- Storage capacity is less.

3. RAID-2 (Bit-Level Stripping with Dedicated Parity)
- RAID-2 is a specialized RAID level that uses bit-level striping combined with error correction using Hamming Code. In this configuration, data is distributed at the bit level across multiple drives, and a dedicated parity drive is used for error detection and correction. While it offers strong fault tolerance, its complexity and cost make it rarely used in practice.

Advantages
- In case of Error Correction, it uses hamming code.
- It Uses one designated drive to store parity.

Disadvantages
- It has a complex structure and high cost due to extra drive.
- It requires an extra drive for error detection.

4. RAID-3 (Byte-Level Stripping with Dedicated Parity)

- RAID-3 enhances fault tolerance by employing byte-level striping across multiple drives and storing parity information on a dedicated parity drive. The dedicated parity drive allows for the reconstruction of lost data if a single drive fails. This configuration is suitable for workloads requiring high throughput for sequential data but is less efficient for random I/O operations.

- Here Disk 3 contains the Parity bits for Disk 0, Disk 1, and Disk 2. If data loss occurs, we can construct it with Disk 3.

Evaluation

Reliability: RAID-3 can tolerate the failure of one disk. The lost data can be reconstructed using the parity drive and the remaining data drives.

Capacity: Usable Capacity=$(N-1)\times B$ where N is the total number of drives, and B is the number of blocks per drive. The capacity of one drive is reserved for storing parity information.

Advantages

- Data can be transferred in bulk.
- Data can be accessed in parallel.

Disadvantages

- It requires an additional drive for parity.
- In the case of small-size files, it performs slowly.

Read about Difference Between RAID 2 and RAID 3.

5. RAID-4 (Block-Level Stripping with Dedicated Parity)

- RAID-4 introduces block-level striping across multiple disks, combined with a dedicated parity disk to provide fault tolerance. Data is written in blocks, and a separate disk stores parity information calculated using the XOR function. This setup allows for data recovery in case of a single disk failure, making RAID-4 more reliable than RAID-0 but less efficient in write-intensive scenarios due to reliance on a dedicated parity disk.

- In the figure, we can observe one column (disk) dedicated to parity.
- Parity is calculated using a simple XOR function. If the data bits are 0,0,0,1 the parity bit is XOR(0,0,0,1) = 1. If the data bits are 0,1,1,0 the parity bit is XOR(0,1,1,0) = 0. A simple approach is that an even number of ones results in parity 0, and an odd number of ones results in parity 1.

- Assume that in the above figure, C3 is lost due to some disk failure. Then, we can recompute the data bit stored in C3 by looking at the values of all the other columns and the parity bit. This allows us to recover lost data.

Read about Difference Between RAID 3 and RAID 4.

Evaluation

- **Reliability:** 1
  RAID-4 allows recovery of at most 1 disk failure (because of the way parity works). If more than one disk fails, there is no way to recover the data.
- **Capacity:** (N-1)*B
  One disk in the system is reserved for storing the parity. Hence, (N-1) disks are made available for data storage, each disk having B blocks.

Advantages

- It helps in reconstructing the data if at most one data is lost.

Disadvantages

- It can’t help reconstructing data when more than one is lost.

6. RAID-5 (Block-Level Stripping with Distributed Parity)

- RAID-5 builds on RAID-4 by distributing parity information across all disks instead of storing it on a dedicated parity drive. This distributed parity significantly improves write performance, especially for random write operations, while maintaining fault tolerance for single disk failures. RAID-5 is one of the most commonly used RAID configurations due to its balance between reliability, performance, and storage efficiency.

How RAID-5 Works?

Block-Level Striping: Data is divided into blocks and striped across all drives in the array.

Distributed Parity: Parity bits, calculated using the XOR function, are distributed across all drives in a rotating pattern.

For example:

- Disk 0: Data (D1, D2), Parity (P1)
- Disk 1: Data (D3, D4), Parity (P2)
- Disk 2: Data (D5, D6), Parity (P3)

This rotation ensures no single disk is burdened with all parity operations, reducing bottlenecks.

Data Recovery: In case of a single disk failure, the missing data can be reconstructed by XOR-ing the remaining data blocks and parity information.

Evaluation

- **Reliability:** 1
  RAID-5 allows recovery of at most 1 disk failure (because of the way parity works). If more than one disk fails, there is no way to recover the data. This is identical to RAID-4.
- **Capacity:** (N-1)*B
  Overall, space equivalent to one disk is utilized in storing the parity. Hence, (N-1) disks are made available for data storage, each disk having B blocks.

Advantages

- Data can be reconstructed using parity bits.
- It makes the performance better.

Disadvantages

- Its technology is complex and extra space is required.
- If both discs get damaged, data will be lost forever.

7. RAID-6 (Block-Level Stripping with two Parity Bits)

- RAID-6 is an advanced version of RAID-5 that provides enhanced fault tolerance by introducing double distributed parity. This allows RAID-6 to recover from the failure of up to two disks simultaneously, making it more reliable for critical systems with larger arrays. However, the added parity calculations can impact write performance.

How RAID-6 Works ?

Block-Level Striping: Data is divided into blocks and striped across all disks in the array.

Double Distributed Parity: Two sets of parity information are calculated for every block and distributed across all disks in the array in a rotating pattern.

Example:

- Disk 0: Data (D1, D2), Parity (P1)
- Disk 1: Data (D3, D4), Parity (P2)
- Disk 2: Data (D5, P1), Parity (P3)
- Disk 3: Parity (P2, P3), Data (D6)

Data Recovery: If one or two disks fail, the missing data can be reconstructed using the remaining data and parity information.

Evaluation

1. Reliability: RAID-6 can tolerate the simultaneous failure of two disks, providing greater fault tolerance than RAID-5.

2. Capacity: Usable Capacity=$(N-2)\times B$ where N is the total number of disks and B is the number of blocks per disk.

Advantages

- Very high data Accessibility.
- Fast read data transactions.

Disadvantages

- Due to double parity, it has slow write data transactions.
- Extra space is required.

Advantages of RAID

- **Data redundancy:** By keeping numerous copies of the data on many disks, RAID can shield data from disk failures.
- **Performance enhancement:** RAID can enhance performance by distributing data over several drives, enabling the simultaneous execution of several read/write operations.
- **Scalability:** RAID is scalable, therefore by adding more disks to the array, the storage capacity may be expanded.
- **Versatility:** RAID is applicable to a wide range of devices, such as workstations, servers, and personal PCs

Disadvantages of RAID

- **Cost:** RAID implementation can be costly, particularly for arrays with large capacities.
- **Complexity:** The setup and management of RAID might be challenging.
- **Decreased performance:** The parity calculations necessary for some RAID configurations, including RAID 5 and RAID 6, may result in a decrease in speed.

- **Single point of failure:** RAID is not a comprehensive backup solution while offering data redundancy. The array's whole contents could be lost if the RAID controller malfunctions.

Conclusion

In Conclusion, RAID technology in database management systems distributes and replicates data across several drives to improve data performance and reliability. It is a useful tool in contemporary database setups since it is essential to preserving system availability and protecting sensitive data.