

Day 1

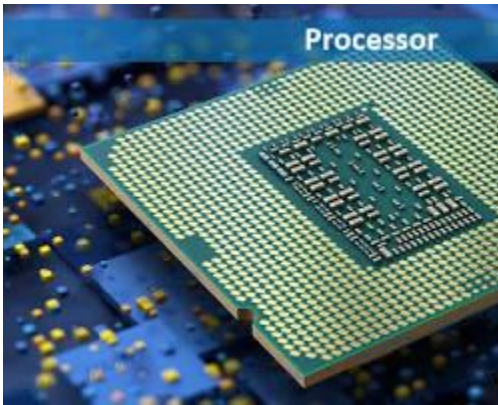
Morning Session

1. What is Computer Architecture?

Computer architecture is the blueprint that defines how a computer system's hardware and software work together. It outlines the structure, organization, and functionality of the computer, ensuring efficient and reliable operation.

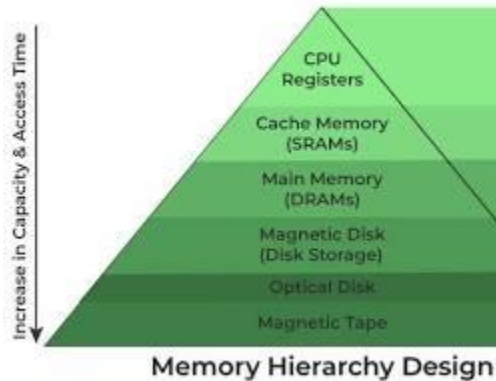
Key Components of Computer Architecture

1. Central Processing Unit (CPU):



Central Processing Unit (CPU)

- The "brain" of the computer, responsible for executing instructions and performing calculations.¹
 - Contains the Control Unit (CU) which fetches and decodes instructions, and the Arithmetic Logic Unit (ALU) which performs arithmetic and logical operations.
- #### 2. Memory:



Memory in computer architecture

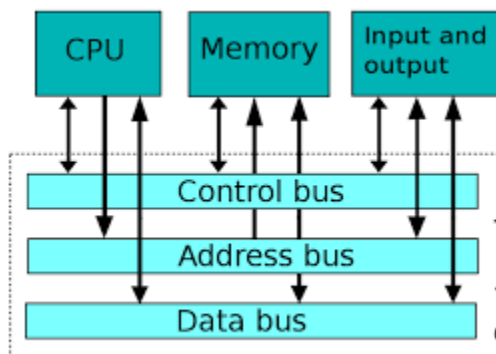
- Stores data and instructions temporarily while the computer is running.
- Divided into:
 - **Main Memory (RAM):** Volatile, fast memory for active data and programs.
 - **Secondary Storage:** Non-volatile, slower storage for long-term data (e.g., hard drives, SSDs).

3. Input/Output (I/O) Devices:

Input/Output (I/O) Devices in computer architecture

- Allow communication with the outside world.
- Examples include:
 - **Input Devices:** Keyboard, mouse, scanner, microphone
 - **Output Devices:** Monitor, printer, speakers

4. System Bus:



System Bus in computer architecture

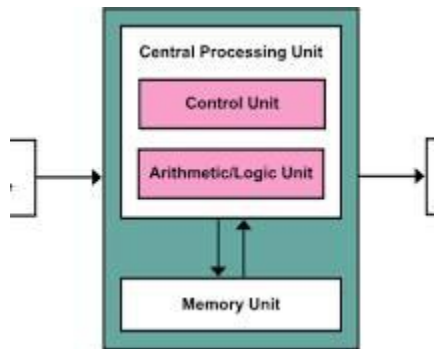
- A collection of wires that connect the CPU, memory, and I/O devices, allowing them to communicate and transfer data.

How They Work Together

1. **Instruction Fetch:** The CPU fetches an instruction from memory.
2. **Instruction Decode:** The CPU decodes the instruction to understand what needs to be done.
3. **Data Fetch:** If the instruction requires data, the CPU fetches it from memory.
4. **Execution:** The CPU executes the instruction, performing calculations or manipulating data.
5. **Store Results:** The results of the operation are stored back in memory.

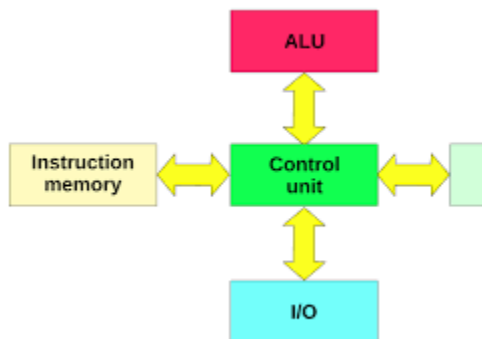
Types of Computer Architecture

- **Von Neumann Architecture:** The most common type, where instructions and data share the same memory.



Von Neumann Architecture

- **Harvard Architecture:** Uses separate memory for instructions and data, improving performance.



Harvard Architecture

- **Modified Harvard Architecture:** A hybrid of Von Neumann and Harvard, using separate memory for instructions and data, but allowing some overlap.

2. Memory Hierarchy

A computer system's memory hierarchy is a strategic arrangement of different types of storage, each with varying speeds, capacities, and costs. This hierarchy is designed to optimize data access and overall system performance by exploiting the principle of locality of reference.

Key Components of Memory Hierarchy

1. Registers:

- Fastest and most expensive storage.
- Located directly within the CPU.
- Used to store temporary data during program execution.
- Smallest capacity.

2. Cache Memory:

- High-speed memory that stores frequently accessed data.
- Located closer to the CPU than main memory.
- Reduces average memory access time.
- Divided into levels (L1, L2, L3) with increasing size and decreasing speed.

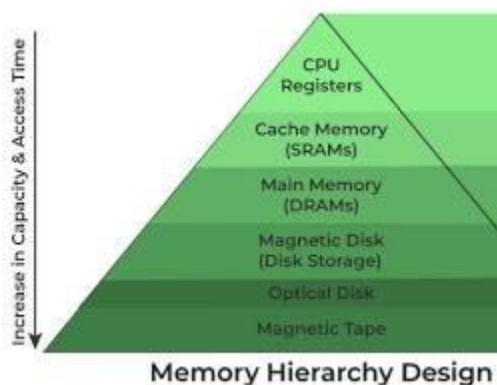
3. Main Memory (RAM):

- Primary storage for data and instructions.
- Slower than cache but faster than secondary storage.
- Volatile, meaning data is lost when power is turned off.

4. Secondary Storage:

- Non-volatile storage for long-term data.
- Slower than main memory but has a larger capacity.
- Examples include hard disk drives (HDDs), solid-state drives (SSDs), and optical discs.

Visual Representation of Memory Hierarchy



Memory Hierarchy

How Memory Hierarchy Works

When the CPU needs to access data, it first checks the fastest level of the hierarchy, the registers. If the data is not found there, it moves to the cache. If still not found, the CPU accesses the main memory. Only if the data is not in main memory is it fetched from secondary storage.

Benefits of Memory Hierarchy

- **Improved Performance:** By storing frequently accessed data in faster levels of the hierarchy, the average memory access time is significantly reduced.
- **Cost-Effective:** Using a combination of different storage technologies allows for a balance between performance and cost.
- **Efficient Resource Utilization:** By organizing memory in a hierarchical manner, resources are used more efficiently.

Key Concepts

- **Locality of Reference:** The tendency of programs to access data and instructions that are located near each other in memory.
- **Cache Hit:** When the requested data is found in the cache.
- **Cache Miss:** When the requested data is not found in the cache and must be fetched from a slower level of the hierarchy.

By understanding the memory hierarchy, system designers can optimize performance and efficiency in computer systems

3. The Need for Software: A Digital World's Enabler

Software, the intangible instructions that guide a computer's hardware, is the lifeblood of the modern digital world. It empowers us to perform tasks, access information, and connect with others in ways that were unimaginable just a few decades ago.

Here are some of the key reasons why software is indispensable:

1. Automation and Efficiency:

- **Streamlining Tasks:** Software automates repetitive tasks, saving time and effort.
- **Increasing Productivity:** By automating processes, businesses can increase productivity and reduce costs.
- **Error Reduction:** Software can minimize human error, leading to more accurate and reliable results.

2. Information Access and Management:

- **Knowledge at Your Fingertips:** Software allows us to access and process vast amounts of information quickly and easily.
- **Data Organization:** It helps organize and manage data efficiently.
- **Data Security:** Software provides tools to protect sensitive information from unauthorized access.

3. Communication and Collaboration:

- **Connecting People:** Software enables communication and collaboration across distances.
- **Remote Work:** It facilitates remote work and virtual teams.
- **Social Networking:** It connects people globally through social media platforms.

4. Entertainment and Creativity:

- **Digital Entertainment:** Software provides a wide range of entertainment options, from video games to streaming services.
- **Creative Tools:** It empowers individuals to create art, music, and other forms of media.
- **Virtual Reality:** It offers immersive experiences and simulations.

5. Scientific Research and Innovation:

- **Data Analysis:** Software tools analyze large datasets to uncover patterns and insights.
- **Simulation and Modeling:** It simulates complex systems and predicts outcomes.
- **AI and Machine Learning:** It powers artificial intelligence and machine learning applications.

6. Business Operations:

- **Customer Relationship Management (CRM):** Software helps manage customer relationships and track sales.
- **Enterprise Resource Planning (ERP):** It integrates various business functions, such as finance, HR, and supply chain.
- **E-commerce:** It enables online shopping and digital commerce.

In essence, software has become an integral part of our lives, driving innovation, improving efficiency, and shaping the future. From the simple tasks of word processing and email to complex scientific simulations and artificial intelligence, software is the engine that powers the digital age.