Computer System Architecture

🎯 Lesson Objectives

By the end of this lesson, students should be able to:

- Describe the **IPOS model** (Input–Processing–Output–Storage).
- Explain the **Von Neumann Architecture** and its components.
- Identify the main components of the **CPU** and their functions.
- Set up a **GitHub learning repository** to document personal work.

1️⃣ Introduction to Computer System Architecture

Computer System Architecture is the conceptual design, fundamental operational structure, and functional organization of a computer system. It defines how the various hardware and software components are integrated and interact to execute programs and process data, with the ultimate goal of delivering performance, efficiency, and reliability.

It serves as the **blueprint** for a computer system, detailing the specifications that guide its implementation. Unlike computer *organization*, which deals with the lower-level details of hardware components (e.g., circuit design, memory technology), architecture focuses on the **structure and behavior** of the system as seen by the programmer or the system designer. It is about the attributes that have a direct impact on the logical execution of a program.

2️⃣ The IPOS Model

The **IPOS cycle** describes the four major operations in every computer system:

1. The IPOS Cycle (Input, Processing, Output, Storage)

The IPOS cycle is a fundamental model that describes the four essential functions of every computer system. It represents the continuous sequence of events that allows a computer to process data into useful information.

Cycle Breakdown:

1. **Input**
   - **Purpose:** To get data and instructions from the outside world into the computer.
   - **Process:** Input devices convert human-understandable data (like keystrokes, mouse clicks, or voice) into binary digital signals (0s and 1s) that the computer can understand.
   - **Examples:** Keyboard, Mouse, Microphone, Scanner, Touchscreen, Sensors.
2. **Processing**
   - **Purpose:** To manipulate and transform the input data according to a set of instructions (a program).
   - **Process:** The Central Processing Unit (CPU) is the primary component responsible for processing. It performs arithmetic calculations, makes logical comparisons, and moves data around.
   - **Key Actions:**
     - **Arithmetic Operations:** Addition, Subtraction, etc.
     - **Logical Operations:** Comparisons (e.g., is A > B?).
     - **Control Operations:** Deciding which instruction to execute next.
   - **Example:** Calculating the total of a bill, applying a filter to a photo, searching for a word in a document.
3. **Output**
   - **Purpose:** To present the processed data (information) back to the user in a human-understandable form.
   - **Process:** Output devices convert the computer's digital signals back into a perceivable form.
   - **Examples:** Monitor/Display, Printer, Speakers, Actuators (e.g., a robotic arm).
4. **Storage**

    - **Purpose:** To hold data, instructions, and information for future use. Storage is non-volatile, meaning it retains data even when the computer is powered off.
    - **Process:** Saving the results of processing, or storing the initial programs and data needed for processing.
    - **Types:**
        - **Primary Storage (Main Memory/RAM):** Fast, volatile memory used to hold data and instructions *currently* being processed by the CPU.
        - **Secondary Storage (Permanent):** Non-volatile storage for long-term data retention (e.g., Hard Disk Drive (HDD), Solid-State Drive (SSD), USB drives).

The Cycle in Action (Example: Writing a Document):

1. **Input:** You press keys on the keyboard and click the mouse.
2. **Processing:** The CPU takes your keystrokes, converts them into characters, and formats them according to the word processor's instructions.
3. **Output:** The monitor displays the characters and formatting in real-time.
4. **Storage:** You save the document, writing the data to your SSD or HDD for later use.

🧠 Simple Example:

When you type an essay:

- **Input:** You press keys on the keyboard
- **Processing:** The CPU interprets the keystrokes
- **Output:** Text appears on the screen
- **Storage:** You save your file on disk

🧩 IPOS Model

```
Input → Processing → Output
      ↓
    Storage
```

This model forms the **foundation of computer operations**.

3 The Von Neumann Architecture

This is the foundational design model for virtually all modern computers. Proposed by John von Neumann in 1945, it is also known as the "Stored-Program Computer" architecture.

Core Principles:

1. **Stored-Program Concept:**
    - Both **data** and **instructions** (the program) are stored in the same main memory (RAM).
    - This was a revolutionary idea, as earlier computers had to be physically rewired to run different programs.
2. **Sequential Execution:**
    - The CPU fetches, decodes, and executes one instruction at a time, in a strict sequence. This is managed by the **Program Counter (PC)** register, which points to the next instruction to be executed.

The Four Main Subsystems:

1. **Central Processing Unit (CPU):** The "brain" that processes data and controls other components.
2. **Memory Unit (RAM):** Stores the program instructions and the data being used.
3. **Input/Output (I/O) Systems:** Handle communication with the external world.
4. **System Buses:** The communication pathways that connect all the components.

The "Von Neumann Bottleneck":

- **Problem:** Since both instructions and data share the same memory and bus, the CPU can only perform one operation (either a fetch or a data transfer) at a time. This limits the processing speed, as the CPU is often left waiting for data or instructions to be fetched from memory.

- **Modern Solutions:** Techniques like **caching** (storing frequently used data in fast memory inside the CPU), **pipelining** (overlapping the stages of instruction execution), and **multi-core processors** help mitigate this bottleneck.

4️⃣ Central Processing Unit (CPU)

The CPU is the primary component for processing data and executing instructions. It is a highly complex integrated circuit, often called a microprocessor.

Core Components of the CPU:

A. Control Unit (CU)

- **Function:** The "nervous system" or "traffic cop" of the CPU. It does not process data itself but **directs and coordinates** all operations of the computer.
- **Key Responsibilities:**
  - **Manages the Fetch-Decode-Execute Cycle:**
    1. **Fetch:** Instructs the memory unit to read an instruction from the address held in the Program Counter (PC).
    2. **Decode:** Interprets the instruction to determine what action is required (e.g., add, load, store).
    3. **Execute:** Sends control signals to the relevant components (ALU, registers, memory) to carry out the instruction.
  - **Controls the flow of data** between the CPU, memory, and I/O devices by activating the appropriate buses and components.

B. Arithmetic Logic Unit (ALU)

- **Function:** The "calculator" of the CPU. It performs all arithmetic and logical operations on data.
- **Key Responsibilities:**
  - **Arithmetic Operations:** Addition, Subtraction, Multiplication, Division.
  - **Logical Operations:** Comparisons (AND, OR, NOT, XOR) and relational tests (e.g., equals, less than, greater than).
  - **Bit-Shifting Operations:** Shifting the bits of a binary number left or right.

- The ALU takes its inputs from registers and sends its output back to a register. The specific operation is determined by control signals from the Control Unit.

C. Registers

- **Function:** Small, extremely high-speed memory locations *inside the CPU*. They are used to store temporary data, instructions, and memory addresses that the CPU is actively using. Accessing a register is thousands of times faster than accessing RAM.
- **Key Types of Registers:**
    - **Program Counter (PC):** Holds the memory address of the **next** instruction to be executed.
    - **Memory Address Register (MAR):** Holds the address of a memory location that is to be read from or written to.
    - **Memory Data Register (MDR) / Memory Buffer Register (MBR):** Holds the actual **data** that has been fetched from or is to be written to memory.
    - **Instruction Register (IR):** Holds the **current instruction** being decoded and executed.
    - **Accumulator (ACC):** A general-purpose register that stores the results of calculations performed by the ALU.
    - **Status Register (Flag Register):** Contains individual bits (flags) that indicate specific conditions, such as whether the last result was zero (Zero Flag), negative (Sign Flag), or caused an overflow (Overflow Flag).

D. Buses

- **Function:** A set of parallel wires (conductors) that act as a shared communication channel, transferring data and signals between the CPU, memory, and I/O controllers.
- **Three Primary Types of Buses:**
    1. **Data Bus:**
        - **Purpose:** Carries the actual **data** between components.
        - **Bidirectional:** Data can flow in both directions (to and from the CPU).

    - **Width:** The number of lines (e.g., 32-bit, 64-bit) determines how much data can be transferred at once, which is a key factor in performance.

2. **Address Bus:**
    - **Purpose:** Carries the **memory addresses** from the CPU to the memory unit. The CPU uses this bus to specify *where* it wants to read or write data.
    - **Unidirectional:** Addresses only flow from the CPU to other components.
    - **Width:** Determines the maximum amount of memory the CPU can address (e.g., a 32-bit address bus can address 2^32 = 4 GB of memory).
3. **Control Bus:**
    - **Purpose:** Carries **control and timing signals** from the Control Unit to coordinate the activities of all the other components.
    - **Bidirectional:** Includes lines for signals like **Memory Read**, **Memory Write**, **I/O Read**, **I/O Write**, **Clock**, and **Interrupt** signals.

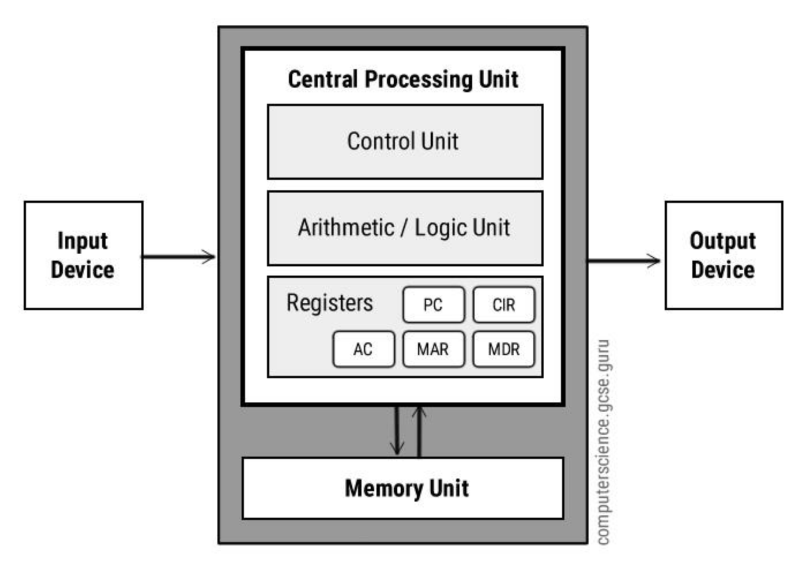
🧱 Main Components of the CPU:

Component	Function
Control Unit (CU)	Directs the flow of data between components.
Arithmetic Logic Unit (ALU)	Handles arithmetic (+, −, ×, ÷) and logic (AND, OR, NOT) operations.
Registers	Small, high-speed memory locations for temporary data storage.
Buses	Pathways that carry data, addresses, and control signals between CPU components.

⚡ Types of Buses

Bus Type	Purpose
Data Bus	Carries actual data.
Address Bus	Specifies memory locations.
Control Bus	Sends control signals to coordinate components.

5 Illustration: CPU Architecture (Text Diagram)



6 Summary of Key Points

Concept	Description
IPOS Model	Sequence of input, processing, output, and storage.
Von Neumann Architecture	Model where data and instructions share memory.
CPU	Core unit containing CU, ALU, registers, and buses.
Fetch-Decode-Execute Cycle	Basic working cycle of modern processors.