academizz

Fundamentals of Computer Design

Course: Computer Architecture and Design
Lecture Notes & Study Module

Prepared by Acadmizz

November 27, 2025

Contents

This is an exclusive content of Academizz. To get access to more resources, subscribe to Academizz (app.academizz.com).

1 Module 1: The Foundations of Computer Systems

1.1 Understanding Architecture vs. Organization

To design or understand a computer, we must distinguish between the ”planning” phase and the ”building” phase.

- **Computer Architecture (The Programmer's View):** This is the conceptual design of the computer. It defines the system's logical behavior. For a programmer, architecture answers the question: *What can this system do?* It includes the Instruction Set Architecture (ISA), data types, and addressing modes.
- **Computer Organization (The Engineer's View):** This is the physical implementation of the architecture. It answers the question: *How does the system do it?* It involves the hardware details—control signals, memory technology, and how the ALU connects to registers.

  Real-World Analogy: Think of a car.

  - **Architecture:** The steering wheel, pedals, and gear shift. Every driver knows how to use these interfaces regardless of the car brand.
  - **Organization:** The engine type (V6 vs. V8), the fuel injection system, and the mechanics of the transmission. These are hidden from the driver but crucial for the mechanic.

1.2 Structural and Functional Design

Computer design occurs in layers. At each layer, we look at the system in two ways:

1. **Functional:** What does each part do? (e.g., The ALU performs math).
2. **Structural:** How are the parts connected? (e.g., The ALU is connected to the internal bus).

The Four Main Functions of a Computer:

- **Data Processing:** Manipulating data (arithmetic/logic).
- **Data Storage:** Holding data temporarily (RAM) or permanently (HDD/SSD).
- **Data Movement:** Transferring data via Input/Output (I/O) or over networks.
- **Control:** Managing the other three functions to ensure instructions are followed correctly.

2 Module 2: The Core Components (The CPU)

The Central Processing Unit (CPU) is the brain of the computer. It contains four primary structural components:

1. **Arithmetic Logic Unit (ALU):** The ”calculator” of the CPU. It performs arithmetic (add, subtract) and logical operations (AND, OR, NOT).
2. **Control Unit (CU):** The ”traffic cop.” It interprets instructions and generates signals to tell other components what to do.
3. **Registers:** Extremely fast, small memory locations inside the CPU used to hold data currently being processed.
4. **Internal Bus:** The data highway connecting the ALU, CU, and registers.

This is an exclusive content of Academizz. To get access to more resources, subscribe to Academizz (app.academizz.com).

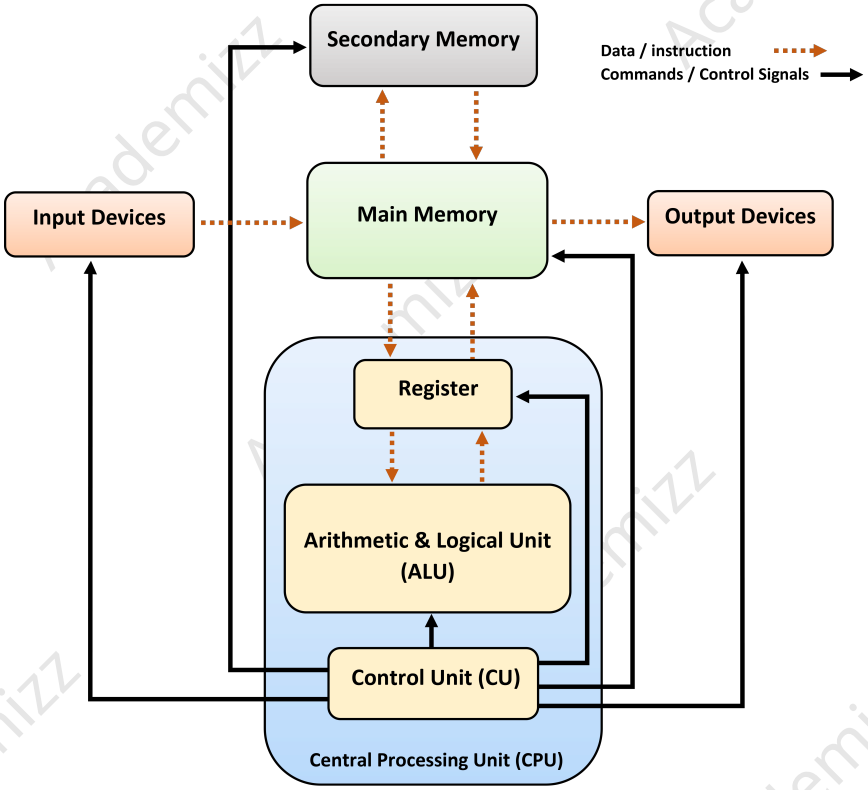

Figure 1: Functional Block Diagram of a Computer System

2.1 The Instruction Cycle

How does a CPU actually run a program? It follows a continuous loop:

1. **Fetch:** Get the next instruction from memory.
2. **Decode:** Figure out what the instruction means.
3. **Execute:** Perform the operation.
4. **Store:** Save the result.

3 Module 3: Architectural Models

3.1 Von Neumann Architecture

This is the classic design used in most PCs and laptops today.

- **Key Concept:** Instructions (code) and Data are stored in the **same** memory and share the **same** bus.
- **The "Bottleneck":** Because they share a bus, the CPU cannot fetch an instruction and read data at the exact same time. This speed limit is called the *Von Neumann Bottleneck*.

3.2 Harvard Architecture

- **Key Concept:** There is separate memory and separate buses for Instructions and Data.

This is an exclusive content of Academizz. To get access to more resources, subscribe to Academizz (app.academizz.com).

- **Advantage:** The CPU can read an instruction and access data simultaneously, leading to higher performance.
- **Use Case:** Common in microcontrollers (like Arduino) and Digital Signal Processors (DSPs).

Table 1: Comparison of Architectural Models

Feature	Von Neumann	Harvard
Memory	Unified (Code + Data)	Separate (Code / Data)
Buses	Shared	Separate
Speed	Slower (Bottleneck)	Faster (Parallel access)
Usage	General Purpose PCs	Microcontrollers

4 Module 4: Memory Hierarchy

Computer memory is arranged in a pyramid structure. The goal is to simulate a memory that is both fast (like a register) and large (like a hard drive).

1. **Registers (Top):** Inside the CPU. Fastest access, smallest capacity.
2. **Cache Memory (L1, L2, L3):** Fast memory located on or near the CPU to hold frequently used data.
3. **Main Memory (RAM):** Primary storage for active programs. It is volatile (loses data when power is off).
4. **Secondary Storage (Bottom):** HDDs or SSDs. Non-volatile, massive capacity, but much slower.

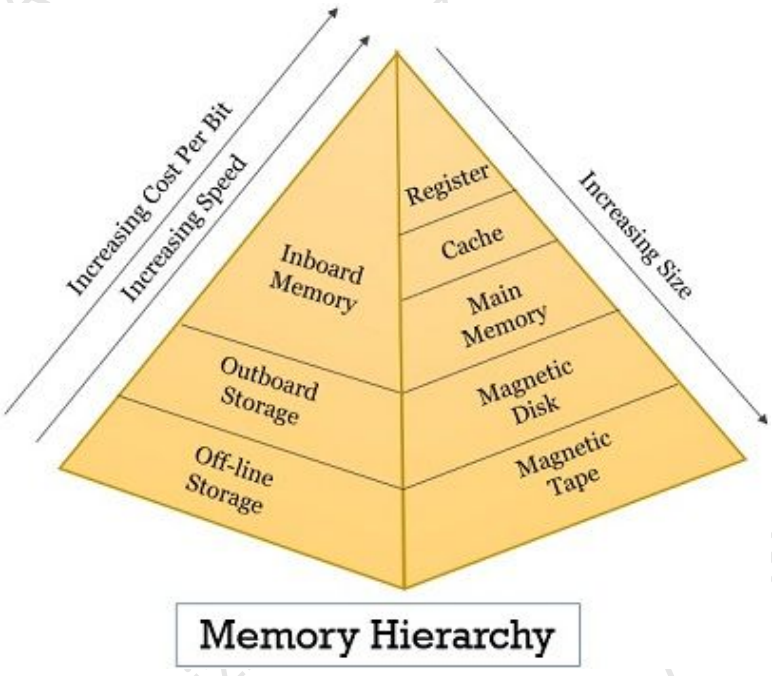

Figure 2: The Memory Hierarchy Pyramid

This is an exclusive content of Academizz. To get access to more resources, subscribe to Academizz (app.academizz.com).

5 Module 5: History and Evolution

5.1 The Generations

- **1st Gen (1940s-50s): Vacuum Tubes.** Huge, hot, and unreliable. Example: The IAS Computer.
- **2nd Gen (1950s-60s): Transistors.** Replaced tubes to make computers smaller, faster, and cooler. Introduction of FORTRAN.
- **3rd Gen (1965-71): Integrated Circuits (ICs).** Multiple transistors packed onto a single chip. Keyboards and monitors became common.

5.2 Moore's Law

In 1965, Gordon Moore observed that the number of transistors on a chip doubles approximately every two years (later revised to 18 months).

$$N(t) = N_0 \times 2^{(t-t_0)/\tau} \tag{1}$$

Where:

- $N(t)$: Number of transistors at time t.
- N_0: Initial number of transistors.
- τ: Doubling period (approx. 2 years).

Current Status: We are hitting physical limits (heat generation and quantum tunneling), so the industry has shifted focus from raw clock speed to parallel processing (multicore).

6 Module 6: Case Study – The Intel 8086

The Intel 8086 (released in 1978) is the ancestor of modern x86 computing. It introduced a clever way to speed up processing by splitting the CPU into two independent units.

6.1 Bus Interface Unit (BIU)

The BIU handles the "logistics." It fetches instructions from memory and stores them in a queue.

- **Pipelining:** Because the BIU has an instruction queue (6 bytes), it can fetch the *next* instruction while the execution unit is still working on the *current* one. This overlapping increases speed.

6.2 Execution Unit (EU)

The EU does the actual work. It takes instructions from the BIU's queue and executes them.

- **Registers:** It contains the General Purpose Registers (AX, BX, CX, DX) and Index Pointers (SI, DI).

This is an exclusive content of Academizz. To get access to more resources, subscribe to Academizz (app.academizz.com).

7 Module 7: Modern Computing Paradigms

7.1 Embedded Systems

These are computers hidden inside other devices (e.g., washing machines, cars, medical devices).

- **Characteristics:** They often have real-time constraints (must react instantly) and are tightly coupled to their environment via sensors.
- **ARM Architecture:** The most popular architecture for embedded systems because of its small size and low power consumption.

7.2 Cloud Computing

Instead of owning powerful servers, users access shared resources over the internet.

- **IaaS (Infrastructure as a Service):** You rent the hardware (storage, networking). You manage the OS and apps.
- **PaaS (Platform as a Service):** You rent the hardware *and* the OS/Runtime. You only manage your applications.
- **SaaS (Software as a Service):** You use the software directly (e.g., Gmail, Google Docs). The provider manages everything.

This is an exclusive content of Academizz. To get access to more resources, subscribe to Academizz (app.academizz.com).