Computer Organization & Architecture

Computer organization and architecture

- Computer organization and architecture are two closely related fields that deal with the design and implementation of computer systems.

- While they are often used interchangeably, there is a subtle difference between the two:

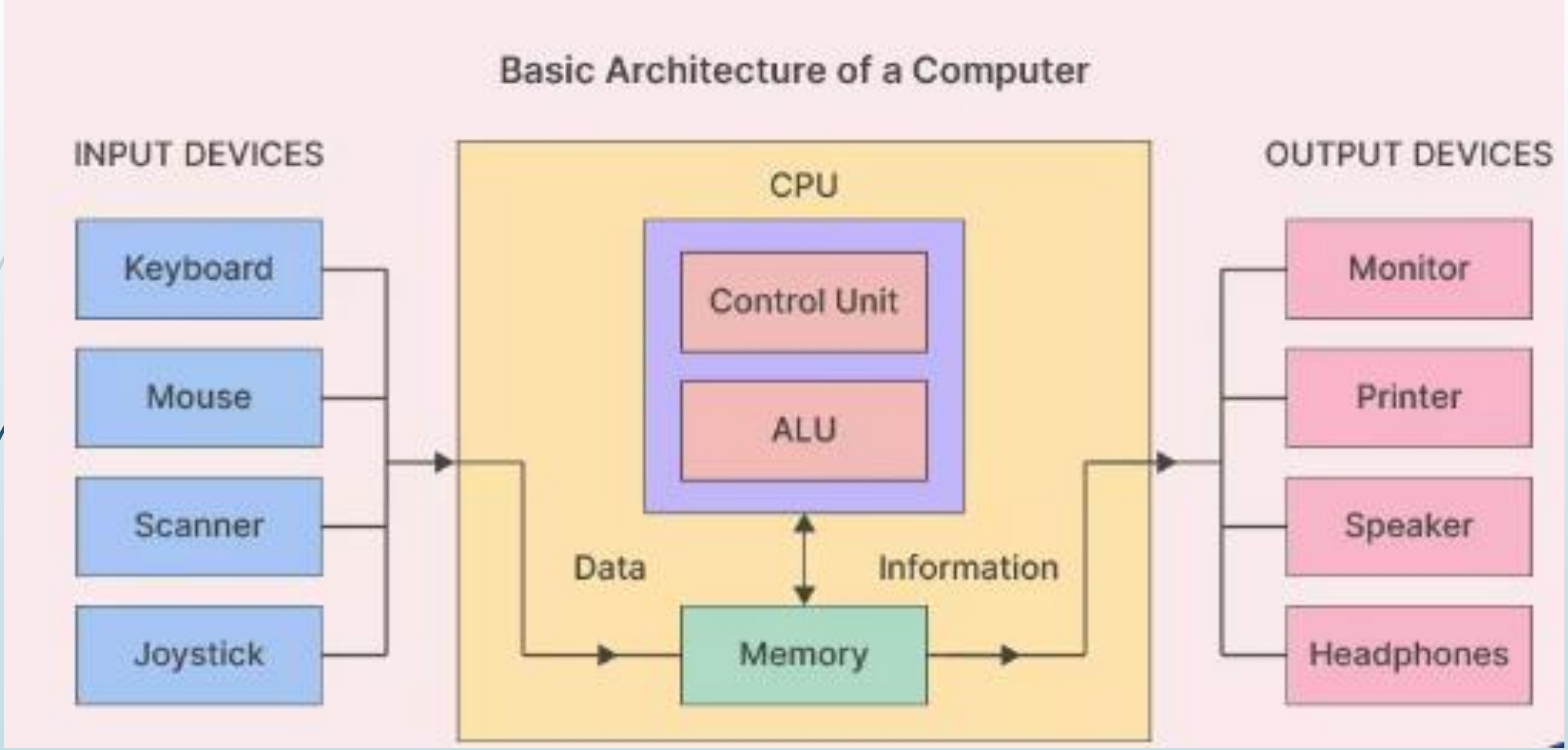
Computer architecture

- Focuses on the **"what":** It defines the high-level **design** of a computer system

- It's like the **blueprint**: It defines the overall structure and capabilities of the system, without going into the nitty-gritty details of how things are implemented.

- Computer architecture sets the foundation for building efficient, reliable, and high-performance computer systems.

Computer architecture (cont.)



Computer architecture (cont.)

- Computer architecture refers to those **attributes** of the system that are visible to the **software programmer** and have a direct impact on the **logical** execution of a program

- Includes:
  - **System Design** (contains hardware components that are used for building the system)
  - **Instruction Set Architecture** (The types of instructions the processor can understand (add, subtract, etc.).)
  - **Interconnection of components** (How the processor, memory, and other units communicate).

Computer organization

- Focuses on the **"how":** It deals with the **physical implementation** of the computer system.

- It's like the **building process**: It takes the blueprint from architecture and figures out how to actually make it work with real hardware.

- Explains how the components are interconnected and work together to realize the desired functionality.

Computer organization (cont.)

- Includes:
  - **<u>Circuit design:</u>** How the processor, memory, and other components are physically built (transistors, etc.).
  - **<u>Bus structure:</u>** How data travels between components (width, speed, etc.).
  - **<u>Control signals</u>**: How components are synchronized and instructed to work together.
  - **<u>Specific hardware choices:</u>** Types of memory chips, processor cores, etc.

To understand the difference

- Think of it like building a house.
  - **Computer Architecture:** This defines the number of rooms, their sizes, types (bedroom, kitchen, etc.), and how they are connected (doors, hallways).

  - **Computer Organization:** This specifies the materials used (bricks, wood), construction techniques, and the actual layout of the rooms and their connections.

To understand the difference (cont.)

- **Computer Architecture** is the plan, **Compute organization** is the construction.

- **Compute Architecture** defines **what** the computer can do, **Compute organization** defines **how** it does it.

Computer Architecture vs. Computer Organization

Computer Architecture	Computer Organization
Concerned with - **What to do**? (Instruction Set)	Concerned with - **How to do**? (implementation of the architecture)
It describes how a computer system is **designed**.	It describes how a computer system **works**.
Refer to **how hardware components connect** to form a computer system (**Functional** behavior of the system).	Refer to the **structure** and **behavior** of the computer system as seen by user (**Structural** relationship of the components)
It is a **blueprint** for **design**. So, while designing a computer system, architecture is decided first.	It's decided after computer architecture (**Implementation of the architecture**)
It involves **logical** components such as Instruction Set, Addressing Modes etc.	It involves **physical** units such as circuit design, adders, signals, peripherals, etc.

Computer Architecture	Computer Organization
Computer architecture deals with **high-level design** issues.	Computer organization deals with **low-level design** issues.
It acts as an interface between hardware and software	It deals with the components of a computer and their interconnection.
It is also called an **instruction set architecture.**	It is also called **microarchitecture**.
Example: Instruction set architecture (ISA)	Example: Memory hierarchy, cache, pipelining

What is the use of studying computer architecture and computer organization?

- Computer architecture and organization provide insights into the fundamental principles underlying the design and operation of computer systems.

- Knowing how instructions are executed, memory is accessed, and data is stored allows programmers to write efficient codes.

- Computer architecture knowledge is crucial for optimizing the performance of applications.

- Studying computer organization helps bridge the gap between hardware and software and provides insights into how the software interacts with the hardware components.

Computer Structure & Computer Function

Computer Structure:

- Think of it as the **"what"** and **"how"** of the **physical components**.

- It refers to the **arrangement and organization** of **hardware components** like CPU, memory, storage, input/output devices, and **their connections**.

- It's like the building blocks and layout of the machine. Think of it as the **hardware**

Computer Structure:

- Example:
  - **Hardware components:** Processor, memory, storage, input/output devices, etc.
  - **Interconnections:** How these components are physically connected (e.g., wires, buses).
  - **Control mechanisms**: How data flows and instructions are carried out between components.

Computer Function:

- Think of it as the **"what it does"** or the **tasks it performs**.
- It describes the **operations** and **processes** that a computer performs.
- It consists of the **logic** and **algorithms** used to execute programs and manipulate data.
- It's like the work the machine does with its structure. Think of it as the **software**:

Computer Function:

- Example
  - **Instructions**: The set of commands that tell the computer what to do.
  - **Data processing**: How the computer manipulates and transforms information.
  - **Output**: The results of the computer's operations, like displaying images, playing music, running calculations.

To understand the difference

Imagine a kitchen.

- **Structure:** The stove, oven, refrigerator, pots, pans, and utensils.
- **Function:** using these components to cook, bake, and prepare meals.

Imagine a car:

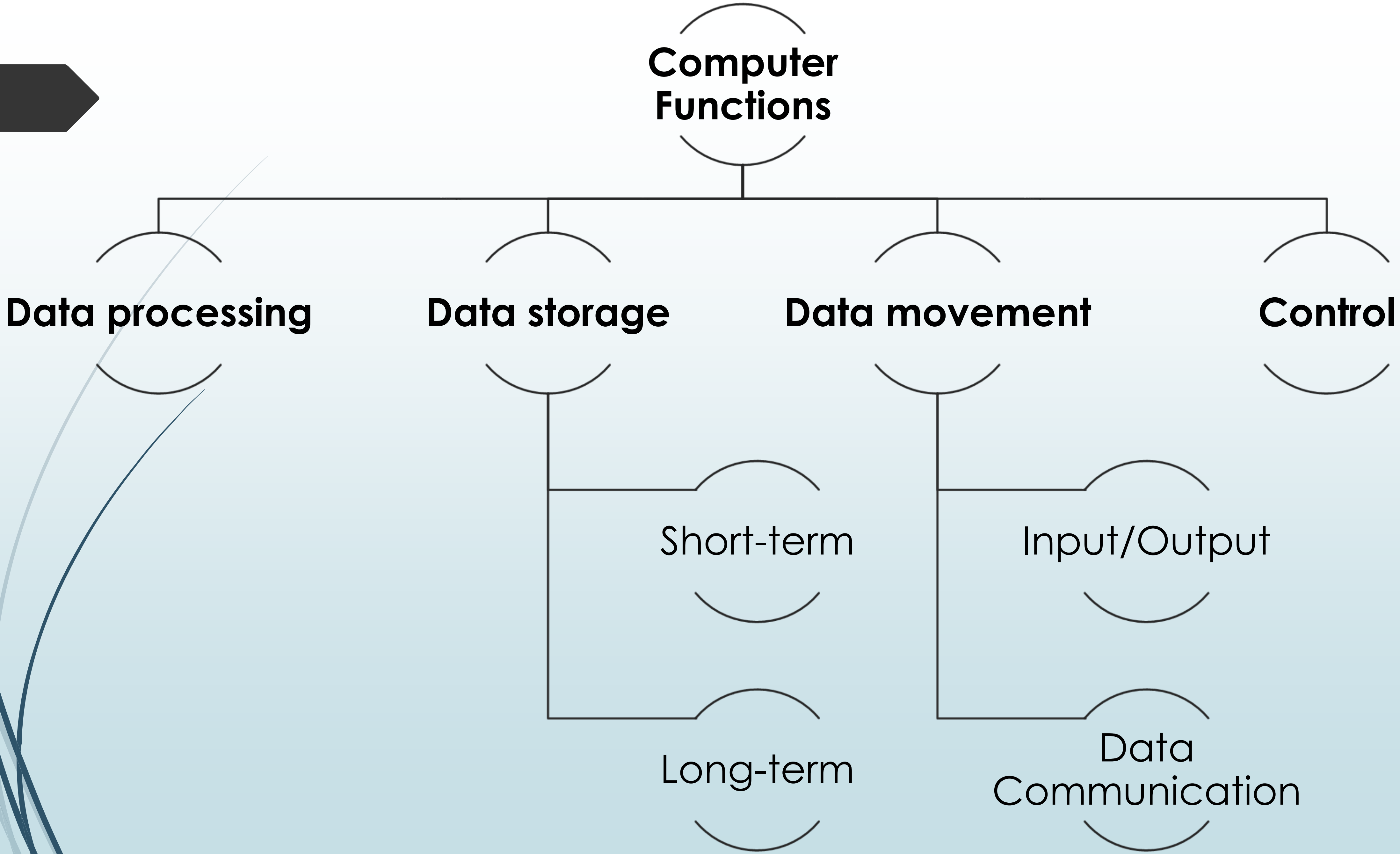
- **Structure:** Engine, wheels, seats, steering wheel, etc., their arrangement and connections.
- **Function:** Driving, transporting people, carrying cargo, etc., the actions it performs.

Summarizing the key differences:

Feature	Computer Structure	Computer Function
Focus	Physical components and organization	Tasks performed and processes followed
Analogy	Building blocks and layout	Work done by the machine
Examples	Processor, memory, storage	Instructions, data processing, output

Computer Function

These simple yet essential gears working in harmony enable computers to handle a vast array of tasks, from basic calculations to complex tasks like artificial intelligence.



Computer Functions
Data processing
Data storage
Data movement
Control
Short-term
Long-term
Input/Output
Data Communication

The 4 basic computer functions:

1) **Data Processing:** Transforming raw data (text, numbers, images) using calculations, comparisons, and logic to get an information.

2) **Data Storage:**

   a) Short-term data storage : Holding information temporarily while working on it (RAM)

   b) Long-term data storage: Holding information permanently for later use (hard drives)

The 4 basic computer functions (cont.):

3) **Data Movement:**
   a) **Input-output (I/O):** Receiving information from input devices (keyboard, mouse) and sending it out output devices (display, printer)that are directly connected to the computer
   b) **Data communications:** when data are transferring over networks.

4) **Control:** Managing the computer's resources and ensuring all functions work together smoothly, like a conductor directing an orchestra.

Computer Structure

Let's explore the internal structure of computers! We'll start with traditional computers with single processor that employs a microprogrammed control unit

A computer consists of four main structural components:

1) **CPU (Processor):** The brain, controlling operations and performing data processing.

2) **Memory:** Stores data for immediate use.

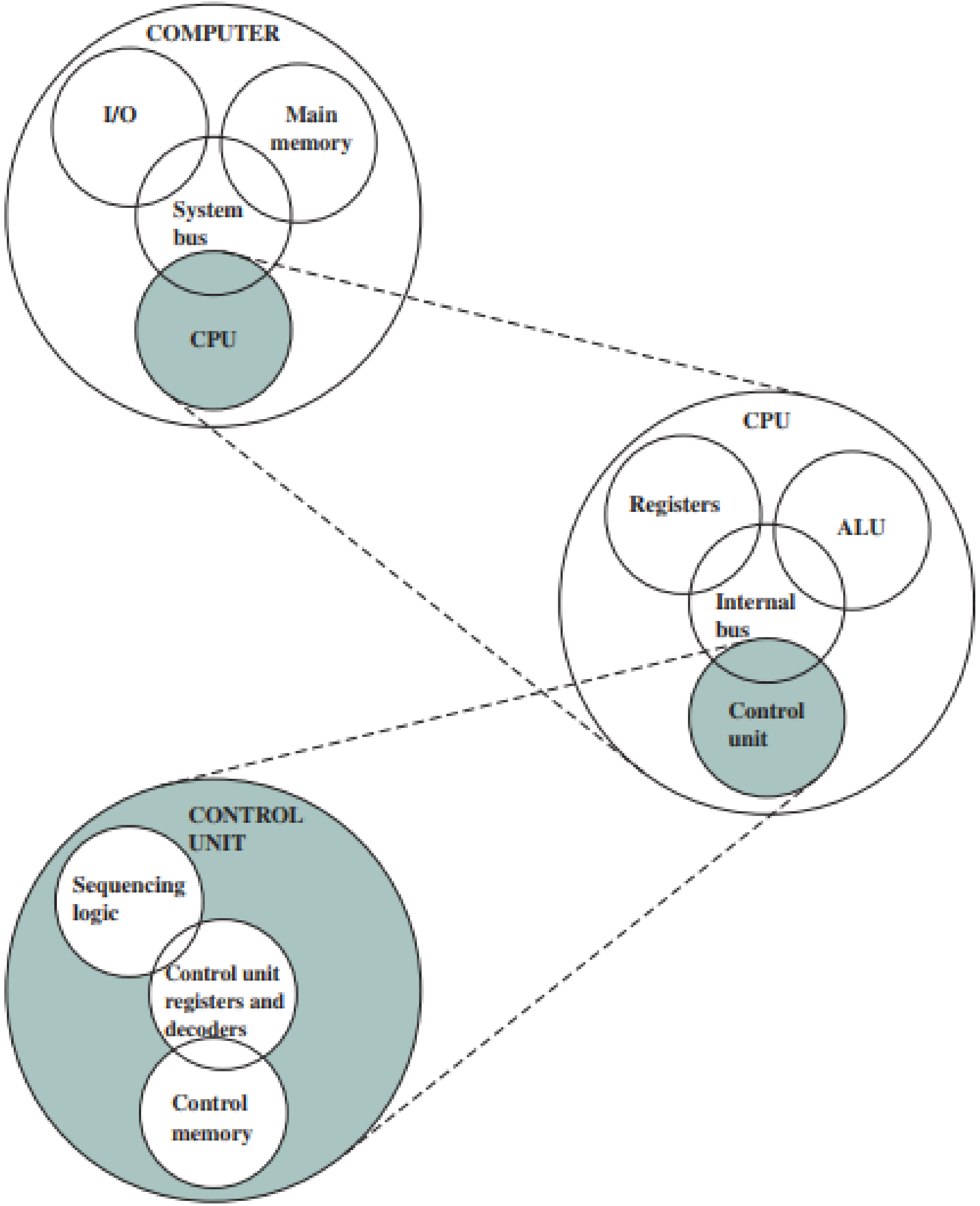
3) **I/O:** Handles data transfer between the computer and the outside world.

4) **System Interconnection:** Connects everything (CPU, memory, I/O), often through a **system bus**.

system bus: consisting of a number of conducting wires to which all the other components attach.

Simple single-processor computer

- The figure provides a hierarchical view of the internal structure of a traditional single-processor computer



The major structural components of CPU

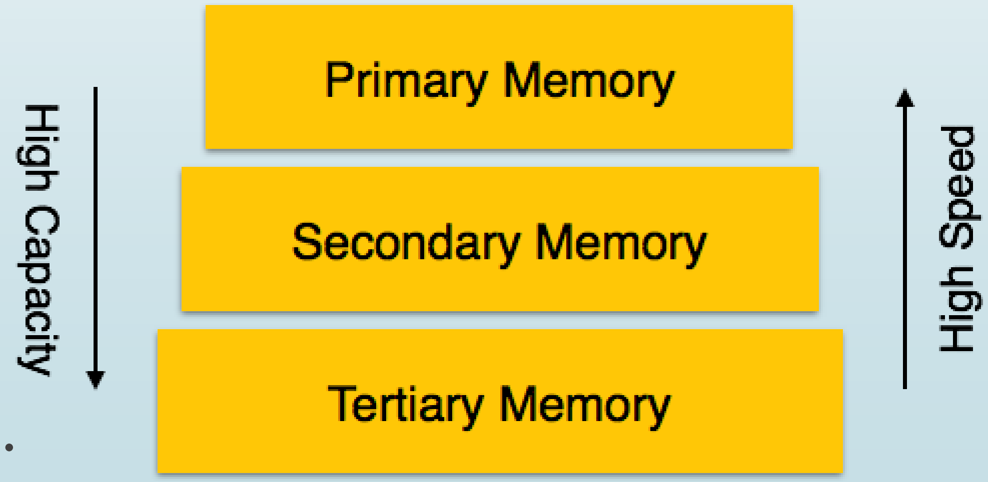
- **Control unit:** The brain of the CPU, it directs the entire operation and coordinates with other components.
- **Arithmetic and logic unit (ALU):** The workhorse, performing calculations and comparisons on data.
- **Registers:** High-speed internal memory, holding data and instructions for immediate use.
- **CPU interconnection:** The internal wiring that enables communication between these components.

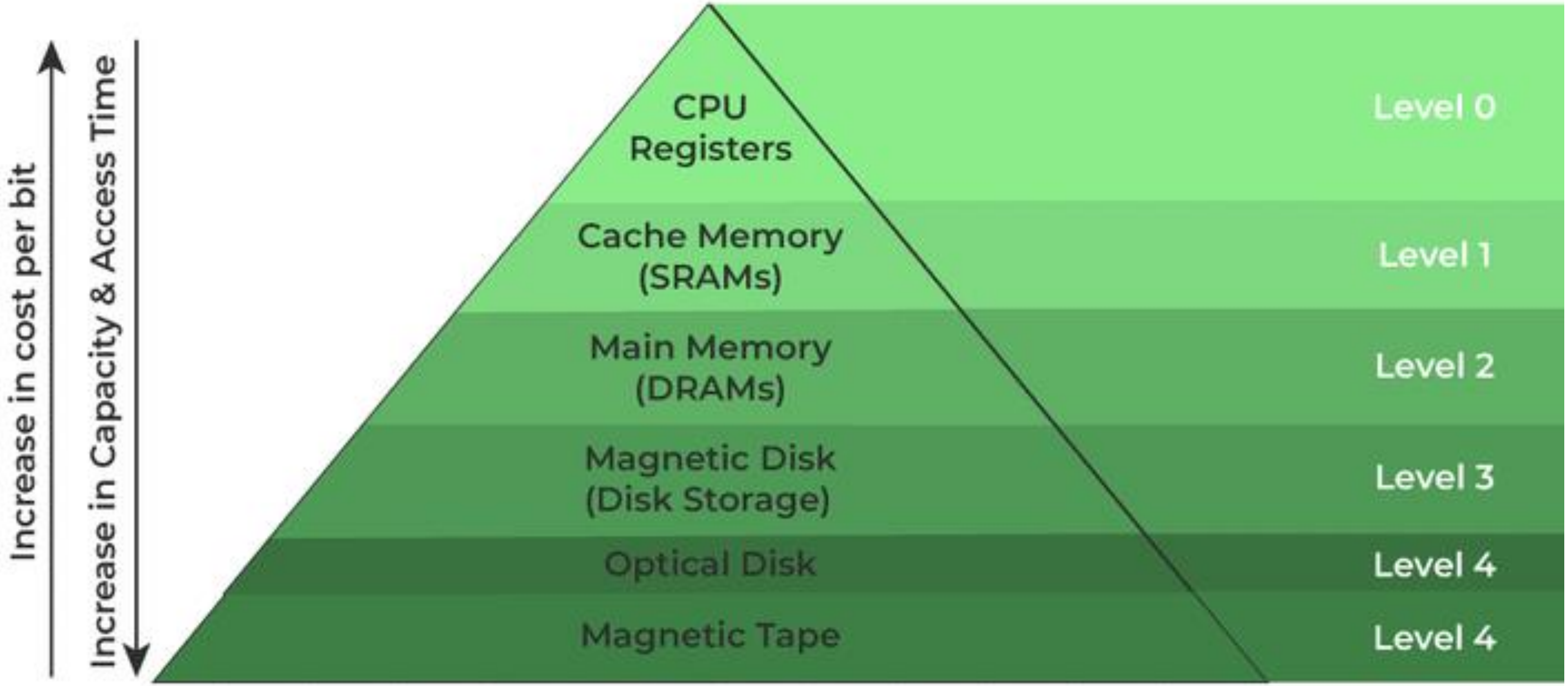
Different Types of Computer Memory (Memory Hierarchy)

Memory hierarchy organizes memory components based on their size, cost, access speed and response times.

Memory Types

- Memory is a hardware device used to store computer programs, instructions and data.
- **Internal Memory (Primary Memory):**
  - Registers, main memory, and cache.
  - Internal memory is directly accessible by the processor.
- **External Memory :**
  - **Secondary storage** (HDDs, SSDs)
  - **Tertiary storage** (magnetic disk, magnetic tape, and optical disk).
  - This is peripheral storage accessible by the processor via an I/O Module.

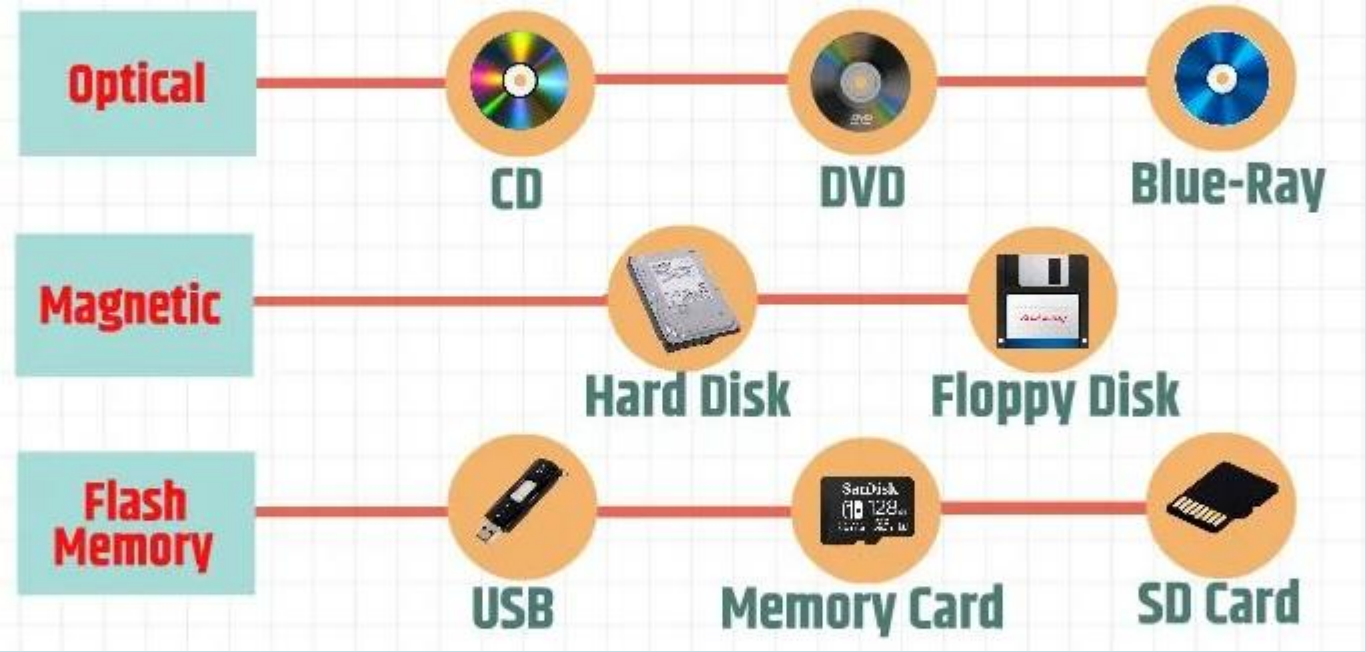




Memory Hierarchy Design

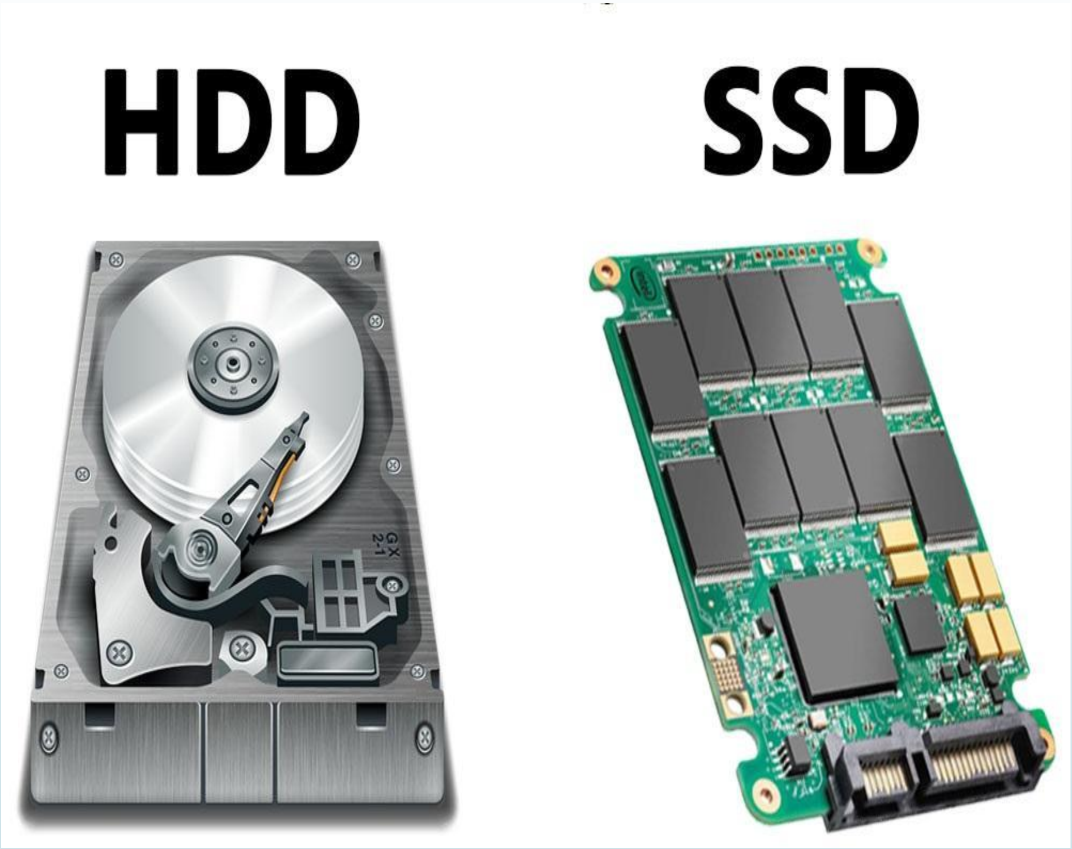
Tertiary Memory

- Tertiary memory refers to storage technologies that offer capacities larger than primary and secondary memory but with significantly slower access times. Tertiary memory examples are:
  - **Magnetic disks.**
  - **Optical disk (CDs, DVDs, Blu-ray Discs).**
  - **Flash Memory (USB , memory Card , SD card)**



Secondary Storage

- Secondary storage is a **non-volatile (permanent) memory** with **greater** storage capacity than main memory.
- Secondary storage has the **slowest access time**
- Common examples of secondary storage include **hard disk drives (HDDs)** and **solid-state drives (SSDs).**



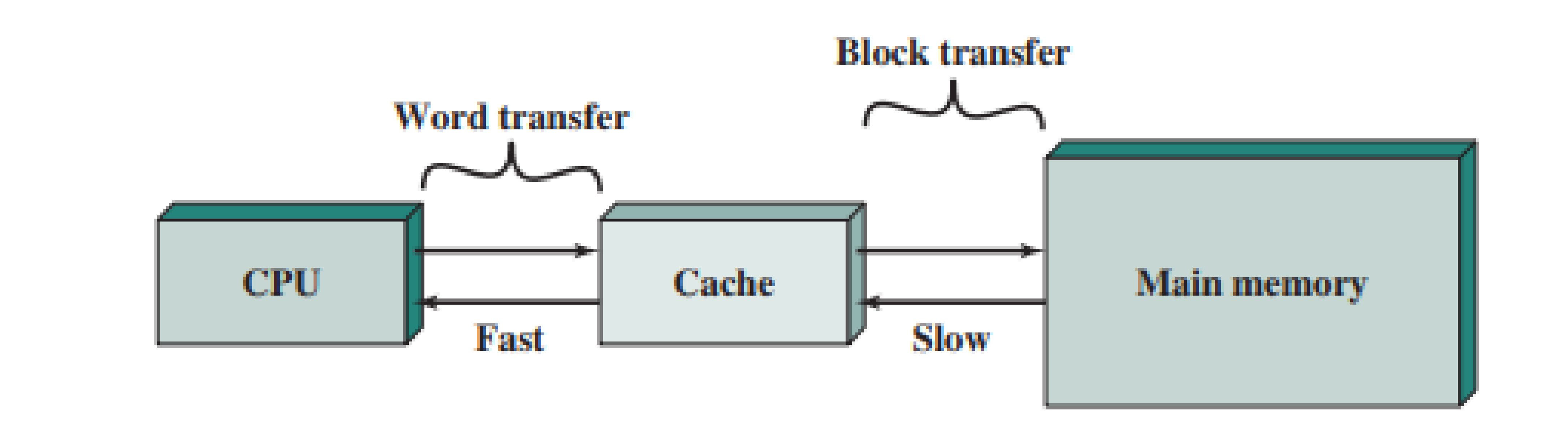
Main Memory

- Main memory, or **RAM** (Random Access Memory), is the primary memory of a computer system.
- It keeps data and instructions the CPU currently uses.
- However, while it has a large storage capacity, RAM is slower than registers and cache memory.
- It's volatile, meaning data gets lost when the power is off.

Cache Memory

- The cache is a **smaller and fast memory.**
- Cache memory is a **separate unit** of memory which is inserted between the CPU and the main memory.
- It allows the processor to quickly access frequently used data by temporarily holding a copy of the information from the main memory.
- The cache memory improves overall system performance and reduces access times.

Cache Memory

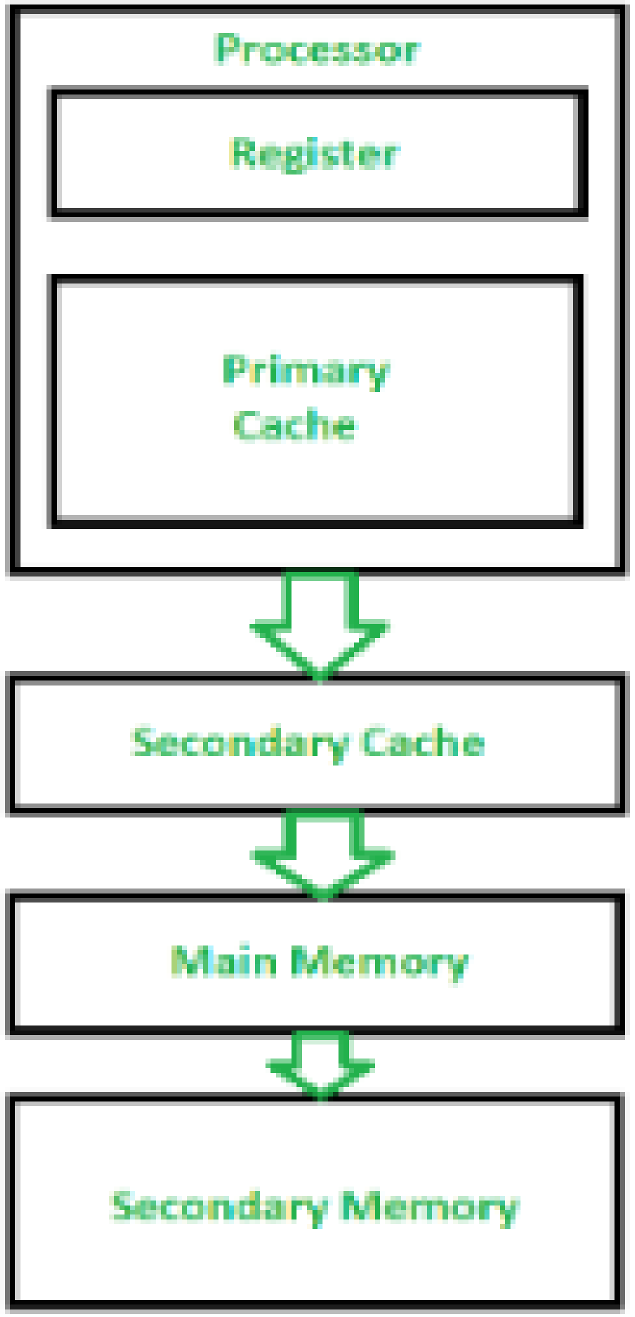


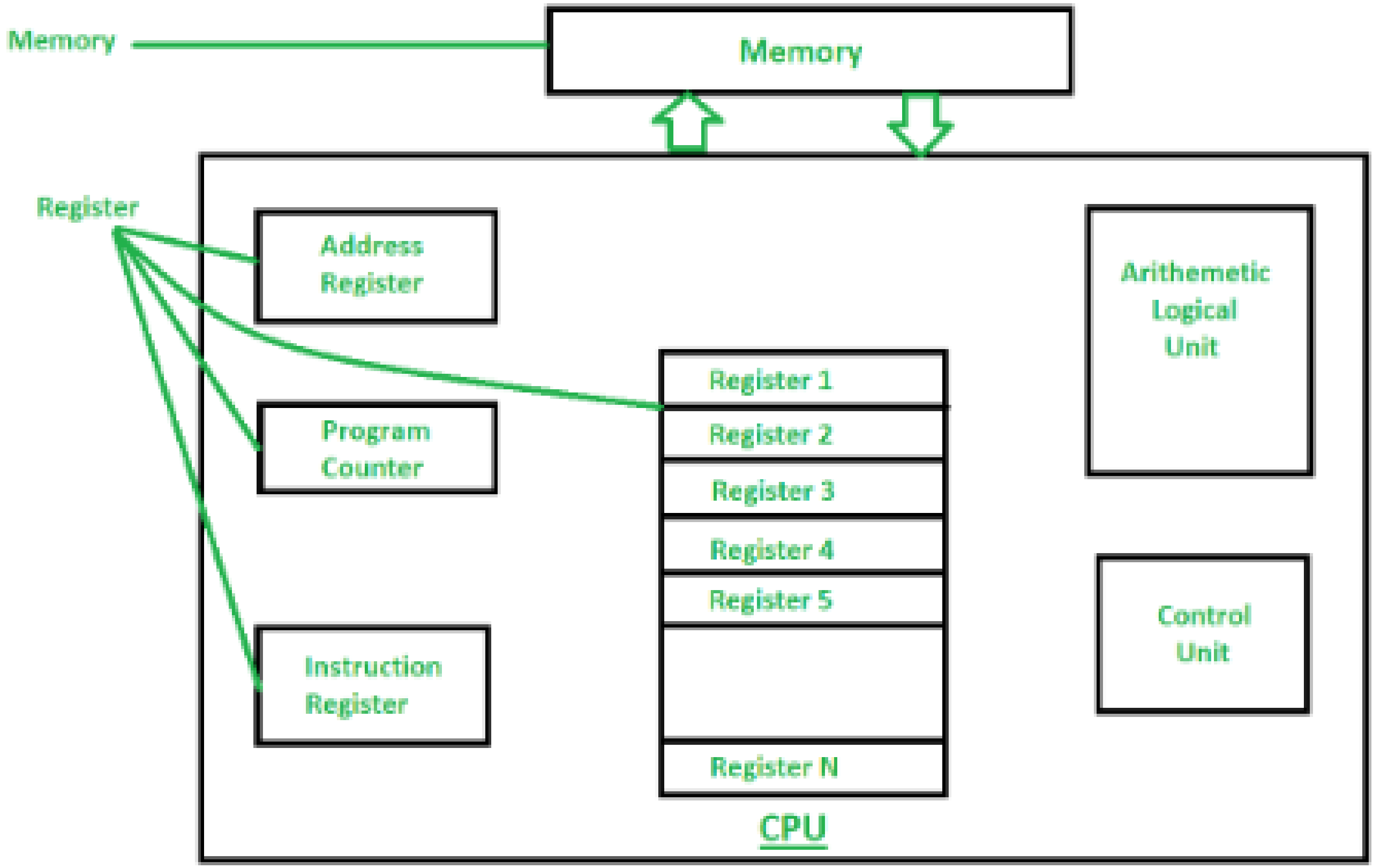
Registers

- Registers are **temporary**, **small, high-speed** and the **most expensive** memory.
- Registers have the **fastest access time** as they are **built into** the processor
- Registers are used to store the data which is directly used by the processor.
- To speed up processing, the CPU copies recently used data from the primary memory into the cache memory. On the other hand, when the CPU needs to execute an operation on specific data, it saves this data into registers.
- Registers can be of different sizes(16 bit, 32 bit, 64 bit and so on)
- However, registers have the smallest storage capacity it can process limited information at a time.

Registers

- Registers loads the resulting data to the main memory and contains the address of the memory location.
- Each register inside the CPU has a specific function like storing an instruction, storing address of a location in memory or storing data such as a bit sequence or individual characters. For example, an instruction may specify that the contents of two defined registers be multiplied together and then placed in a specific register.
- Example: Accumulator register, Program counter, Instruction register, Address register, etc.


Processor
Register
Primary
Cache
Secondary Cache
Main Memory
Secondary Memory


Memory
Memory
Register
Address
Register
Program
Counter
Instruction
Register
Register 1
Register 2
Register 3
Register 4
Register 5
Register N
Arithemetic
Logical
Unit
Control
Unit
CPU

THANK YOU!