جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم

قسم الذكاء الاصطناعي

Lecture: (3)

Central processing units [CPU] , CPU components [ALU,RS,CU], CPU operations
Main memory, Primary storage, Type of main memory [RAM,ROM] , Instruction format with memory
Secondary storage , Type of secondary storage

Subject: COMPUTER ORGANIZATION AND LOGIC DESIGN

Class: First

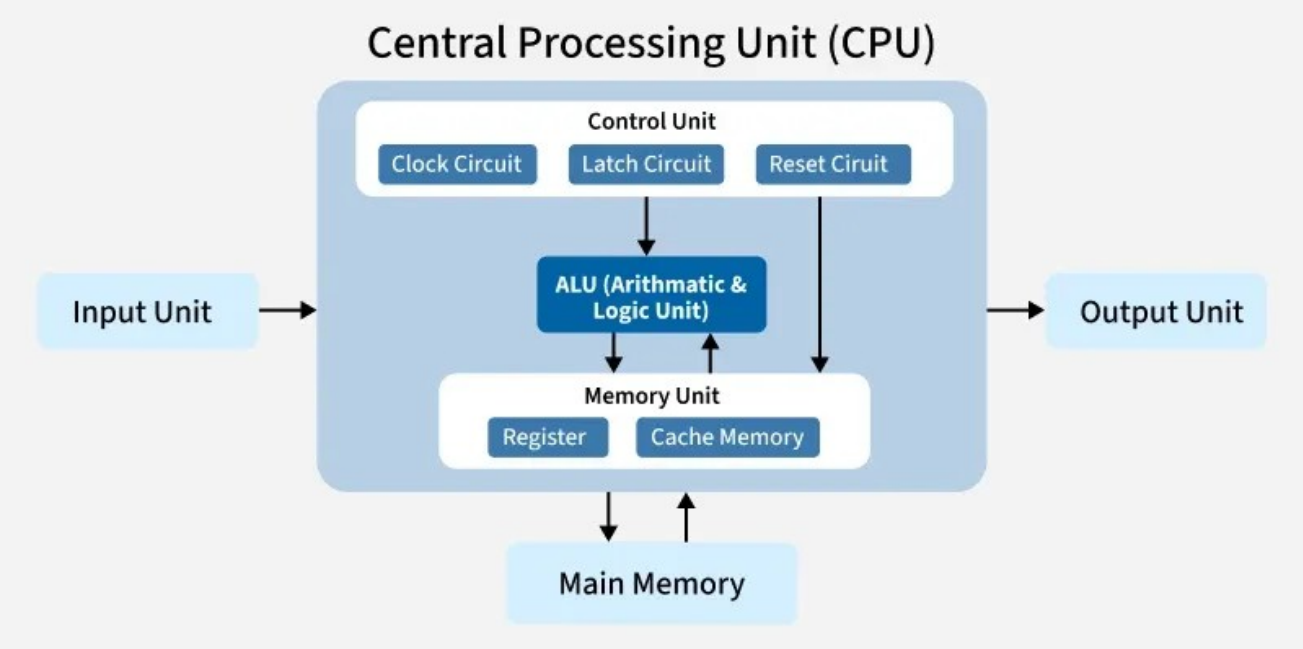
Lecturer: Dr. Maytham N. Meqdad

Central processing units [CPU]

The **Central Processing Unit (CPU)** is like the brain of a computer. It's the part that does most of the thinking, calculating, and decision-making to make your computer work. Whether you're playing a game, typing a school assignment, or watching a video, the CPU is busy handling all the instructions to get the job done.

The CPU is usually placed in a special slot called a **socket** on the computer's **motherboard**, which is like the main circuit board that connects all the parts of a computer. The CPU handles tasks like:

- Doing math calculations (like adding or multiplying numbers).
- Running apps or games.
- Input/Output (I/O) operations: Communicate with memory and peripherals.
- Storing and retrieving information during tasks.



CPU components [ALU, RS, CU]

The main components of the **Central Processing Unit (CPU)** include:

---

1. Arithmetic Logic Unit (ALU)

- Responsible for performing **mathematical operations**:
  - Addition, subtraction, multiplication, division
- Performs **logical operations**:
  - Comparisons (>, <, =)
  - AND, OR, NOT
- It is the part where actual **data processing** happens

---

2. Registers (RS)

- Small, **high-speed storage locations** inside the CPU
- Temporarily hold:
  - Data being processed
  - Instructions
  - Addresses
- Examples:
  - Accumulator
  - Program Counter (PC)
  - Instruction Register (IR)

---

3. Control Unit (CU)

- Acts as the **manager of the CPU**
- Controls and coordinates all operations
- Functions:

- Fetches instructions from memory
- Decodes instructions
- Sends control signals to ALU and registers

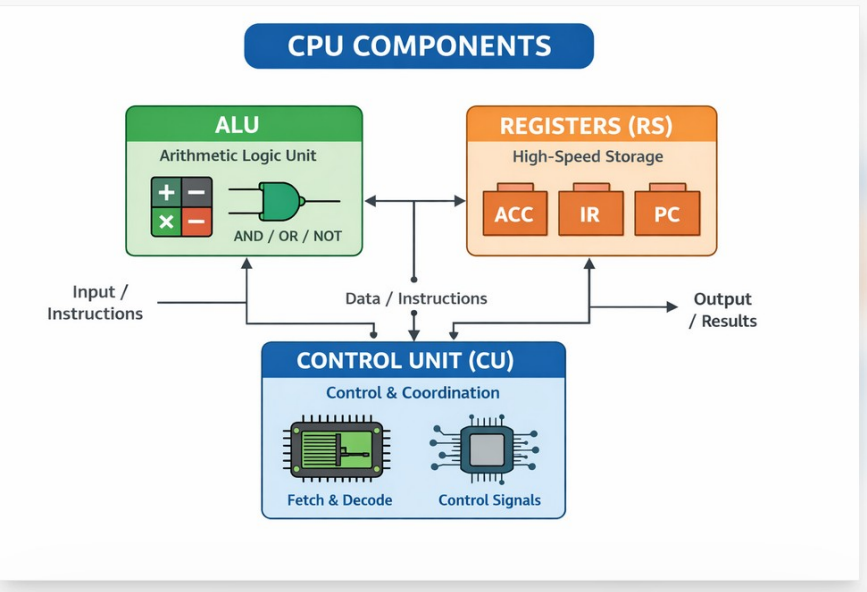
🔹 Simple Explanation

- **ALU** → Calculates and processes data
- **Registers (RS)** → Store temporary data
- **CU** → Controls and organizes operations

🔹 Quick Analogy

Think of the CPU like a factory:

- **CU** = Manager (gives instructions)
- **ALU** = Worker (does calculations)
- **Registers** = Workbench (holds tools and materials)



CPU Operations

CPU Operations refer to the basic actions the **Central Processing Unit (CPU)** performs to execute programs and process data.

---

1. Instruction Cycle (Main CPU Operation)

The CPU works in a continuous cycle called:

Fetch → Decode → Execute → Store

- **Fetch:**
  The CPU retrieves an instruction from memory
- **Decode:**
  The Control Unit interprets the instruction
- **Execute:**
  The Arithmetic Logic Unit performs the operation
- **Store:**
  The result is saved back into Registers or memory

---

2. Types of CPU Operations

1. Arithmetic Operations

- Addition, subtraction, multiplication, division
- Example: 5 + 3

2. Logical Operations

- Comparisons (>, <, =)
- Boolean operations (AND, OR, NOT)

3. Data Transfer Operations

- Moving data between:
  - Registers

- Memory
- Input/Output devices

4. Control Operations

- Direct the flow of execution
- Examples:
  - Jump (branching)
  - Loop control
  - Interrupt handling

---

3. Example of CPU Operation

If the program says:
C = A + B

The CPU will:

1. Fetch the instruction
2. Decode it
3. Load A and B into registers
4. Use ALU to add them
5. Store result in C

Simple Summary

CPU operations are the steps and actions that allow the computer to:

- Process data
- Make decisions
- Execute programs

Main Memory

Main Memory (also called **Primary Memory**) is the part of a computer that stores data and instructions that the **CPU** needs **immediately** while processing.

◆ What is Main Memory?

It is a **fast, directly accessible memory** used by the CPU to:

- Store programs currently running
- Hold data being processed
- Keep intermediate results

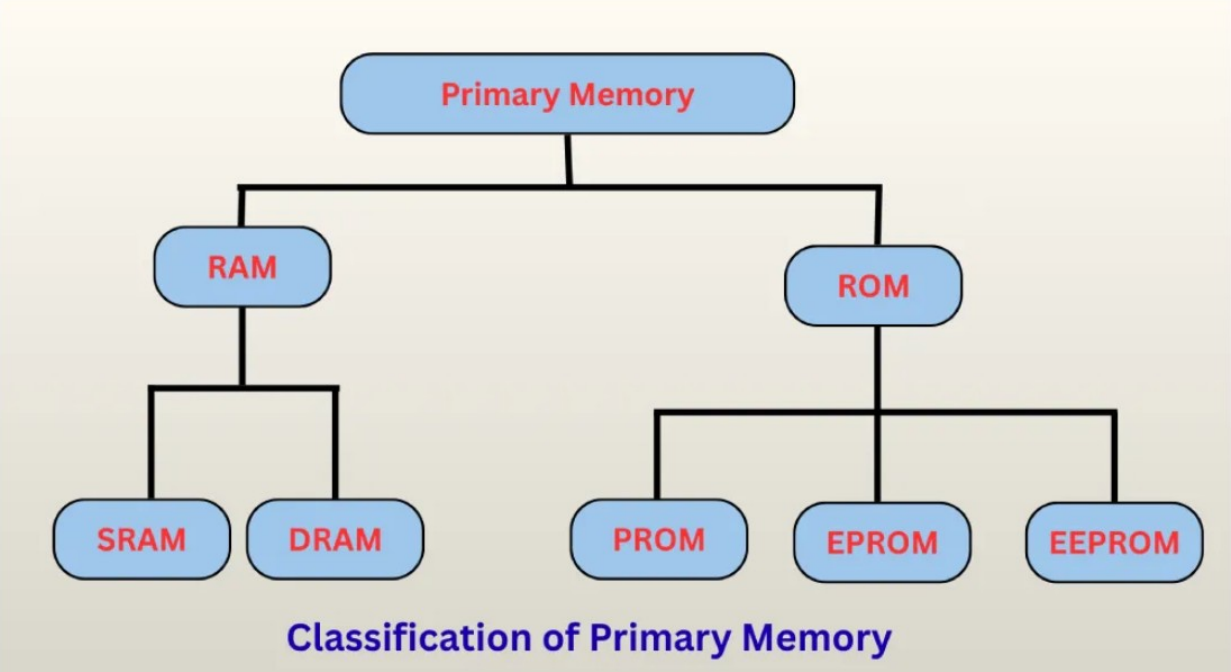
◆ Types of Main Memory

1. RAM (Random Access Memory)

- **Volatile** (data is lost when power is off)
- Used for temporary storage during operation
- Types:
  - DRAM
  - SRAM

2. ROM (Read Only Memory)

- **Non-volatile** (data is permanent)
- Stores firmware (basic startup instructions)



Characteristics of Main Memory

- Fast access speed
- Limited capacity compared to secondary storage
- Directly connected to the CPU
- Expensive compared to hard drives

Functions of Main Memory

- Stores instructions before execution
- Supplies data to the CPU
- Stores results after processing

Simple Example

When you open a program:

- It is loaded from hard disk → into **main memory (RAM)**
- Then the CPU processes it from there

Primary vs Secondary Storage

Feature	Primary Storage	Secondary Storage
Speed	Fast	Slower
Capacity	Smaller	Larger
Volatility	Mostly volatile	Non-volatile
Access	Direct by CPU	Indirect
Example	RAM, Cache, ROM	Hard disk, USB

- **Comparison between RAM and ROM**:

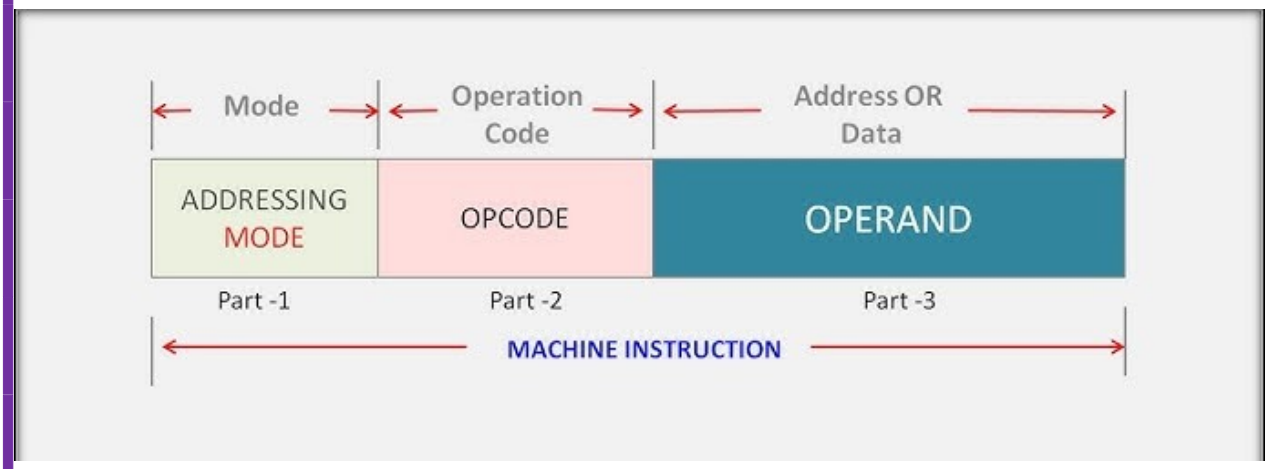
RAM vs ROM

Feature	RAM (Random Access Memory)	ROM (Read Only Memory)
Definition	Temporary working memory	Permanent storage memory
Volatility	Volatile (data lost when power off)	Non-volatile (data retained)
Function	Stores running programs & data	Stores firmware (boot instructions)
Read/Write	Read and Write	Mostly Read Only
Speed	Faster	Slower
Data Storage	Temporary	Permanent
Examples	DRAM, SRAM	PROM, EPROM, EEPROM
Usage	Used during program execution	Used during system startup

What is an Instruction Format?

It is the **layout of bits** in an instruction that tells the CPU:

- What operation to perform
- Where to get the data (memory or registers)



◆ Basic Instruction Format

A typical instruction consists of:

[Opcode | Address / Operand]

- **Opcode** → Specifies the operation (ADD, SUB, LOAD, etc.)
- **Address (Memory Address)** → Specifies the location of data in memory

---

◆ Types of Instruction Format (Based on Memory Usage)

1. One-Address Instruction

- Uses **one memory address**
- Example:
  `ADD X` → AC = AC + M[X]
- Uses an implicit register (Accumulator)

---

2. Two-Address Instruction

- Uses **two addresses**
- Example:
  `ADD A, B` → A = A + B
- One operand acts as both input and result

---

3. Three-Address Instruction

- Uses **three addresses**
- Example:
  `ADD A, B, C` → A = B + C
- Clear and efficient but longer instruction

◆ Example of Instruction with Memory

Instruction:
LOAD A, 500

- **Opcode:** LOAD
- **Operand:** Address 500 (in memory)
- Meaning: Load data from memory location 500 into register A

Memory Addressing

The address field may refer to:

- Direct memory location
- Indirect address
- Register containing address

==========

Secondary Storage

Secondary Storage (also called **Auxiliary Storage**) is used to store data and programs **permanently** outside the main memory.

- It is **non-volatile** (data is not lost when power is off)
- Has **large storage capacity**
- Slower than primary memory
- Not directly accessed by the CPU (data must be loaded into main memory first)

Types of Secondary Storage

1. Magnetic Storage

- Uses magnetic fields to store data
- Examples:
  - Hard Disk Drive (HDD)
  - Magnetic Tape

2. Optical Storage

- Uses laser light to read/write data
- Examples:

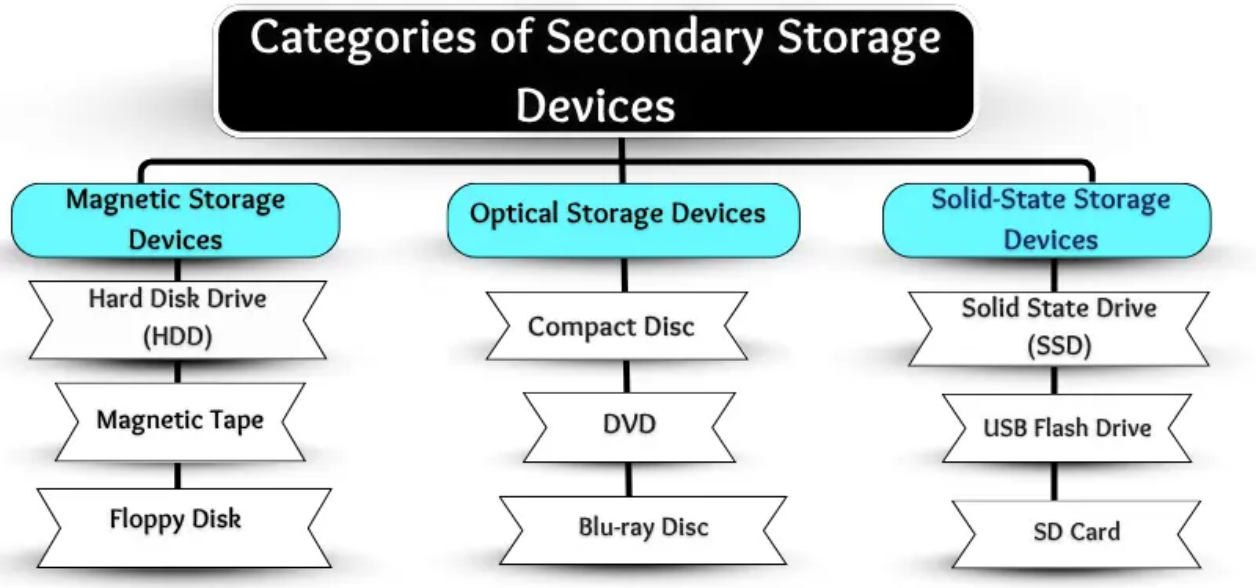
- CD (Compact Disc)
- DVD
- Blu-ray Disc

3. Solid-State Storage

- Uses flash memory (no moving parts)
- Faster and more reliable
- Examples:
  - SSD (Solid State Drive)
  - USB Flash Drive
  - Memory Cards

🔹 Comparison of Types

Type	Speed	Cost	Reliability
Magnetic	Medium	Low	Moderate
Optical	Slow	Low	Moderate
Solid-State	Fast	Higher	High



References

[1] David A. Patterson and John L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, 5th ed. Burlington, MA, USA: Morgan Kaufmann, 2014.

[2] William Stallings, *Computer Organization and Architecture: Designing for Performance*, 10th ed. Boston, MA, USA: Pearson, 2016.

[3] Carl Hamacher, Zvonko Vranesic, and Safwat Zaky, *Computer Organization*, 6th ed. New York, NY, USA: McGraw-Hill, 2012.

[4] Andrew S. Tanenbaum and Todd Austin, *Structured Computer Organization*, 6th ed. Upper Saddle River, NJ, USA: Pearson, 2013.

[5] Morris Mano and Michael D. Ciletti, *Digital Design*, 5th ed. Boston, MA, USA: Pearson, 2013.

[6] Encyclopaedia Britannica, "Computer," Available: https://www.britannica.com/technology/computer.

[7] IBM, "History of Computers," Available: https://www.ibm.com/history.

[8] https://www.geeksforgeeks.org