

1.1 - Basic Computer Organisation

Created By [\(JUMESH KUMAR MEGHWAL\)](#) _____

[Our Website Tap this](#)

Understanding the fundamentals of computers is essential for effectively using modern technology. This foundational knowledge is the basis for developing advanced skills and applications in computing.

Fundamentals of Computing

Computers are complex systems composed of both hardware and software. Hardware includes physical components like the central processing unit (CPU), memory, storage devices, and input/output devices. Software consists of programs that give these hardware components functionality.

Key Hardware Components:

- **Central Processing Unit (CPU):** The brain of the computer that performs computations and manages tasks.
- **Memory:** Short-term storage (RAM) used for current tasks, and long-term storage (HDD/SSD) for saving data.
- **Input Devices:** Devices like keyboards and mice used to input data into the computer.
- **Output Devices:** Devices like monitors and printers that display or produce the results of computations.

Key Software Components:

- **Application Software:** Programs that help users perform specific tasks (e.g., word processors, web browsers).
- **System Software:** Software that manages the hardware and provides a platform for applications (e.g., operating systems, compilers).

Introduction to Computers

Computers have become ubiquitous in our lives, from managing emails to performing complex calculations. They are integral to various aspects of daily life, including work, education, and entertainment.

Definition of a Computer: A computer is an electronic device that processes information based on predefined rules and stores data for future reference. It consists of multiple components working together to perform various tasks. **Computer Architecture and Organization**

Computer Architecture:

- **Architecture** refers to the design and layout of the computer's components. It outlines the system's capabilities and requirements without delving into detailed hardware implementation.
- **Example:** Designing a system to perform binary addition using NAND gates.

Computer Organization:

- **Organization** focuses on the actual implementation and utilization of components, such as how they are connected and communicate with each other.
- **Components:** Includes memory address registers (MAR), memory buffer registers (MBR), program counters (PC), and instruction registers (IR).

Instructions and Instruction Cycle

Instructions are fundamental units of machine language programs that guide the CPU's operations. Each instruction includes:

- **Opcode:** The operation to be performed.
- **Address Field:** The location where the operation will take place.
- **Mode Field:** The method of handling operands.

Instruction Cycle:

1. **Fetch:** Retrieve the instruction from memory.
2. **Decode:** Determine what the instruction will do.
3. **Read:** Access data needed for the operation.
4. **Execute:** Perform the operation.

Types of Instructions

1. **Memory Reference Instructions:** Operate on data in memory.
2. **Input-Output Instructions:** Manage data transfer between the CPU and external devices.
3. **Register Reference Instructions:** Use data in CPU registers.

Addressing Modes

Addressing modes determine how the CPU locates the operands for instructions:

- **Implied Addressing Mode:** The operation is defined within the instruction itself. ○
Example: INC AL (increments the value in register AL).
- **Immediate Addressing Mode:** The operand is included directly in the instruction. ○
Example: MOV AL, 5 (moves the value 5 into register AL).
- **Direct Addressing Mode:** The instruction specifies the memory address directly.

- *Example:* MOV AL, [0400] (moves data from memory address 0400 to register AL).
- **Indirect Addressing Mode:** The instruction specifies a memory address that contains the address of the actual data.
 - *Example:* LOAD AL, @400 (loads data from the address found at location 400 into register AL).
- **Register Addressing Mode:** The operand is located in a register.
 - *Example:* LD R1 (loads data from register R1).
- **Indexed Addressing Mode:** Uses an index register to find the operand's address.
 - *Example:* A = A + [X] (adds the value at memory address indicated by X to A).
- **Auto Increment/Decrement Addressing Modes:** Automatically increments or decrements the address after each operation.
 - *Example:* Fetching rows from a database, where each row's address is incremented automatically.

Bus System

A **system bus** connects the CPU to other components within the computer and allows data transfer between them.

Types of Buses:

1. **Data Bus:** Transfers actual data between components.
2. **Address Bus:** Provides information about where data should be read from or written to.
3. **Control Bus:** Manages the timing and control signals needed for data transfer.

Bus Types:

- **Synchronous Bus:** Operates in sync with the computer's clock, allowing high-speed data transfer.
- **Asynchronous Bus:** Operates independently of a clock, providing flexibility in connecting devices.