भारतीय संस्कृति, शिक्षा और दर्शन

विभिन्न संकल्पनाएं

सम्पादक : डॉ. दिनेश चन्द्र शर्मा

भारतीय संस्कृति, शिक्षा और दर्शन

विभिन्न संकल्पनाएँ

सम्पादक :

डॉ. दिनेश चंद्र शर्मा

Ph.D., D.Lit. (Sub.)

निदेशक, कॉम्पिटिशन प्लस ग्रुप ऑफ इन्स्टीट्यूशन,
अजमेर

Books n Books

51/1509, आचमन, वेदविहार,
लोहाखान, अजमेर-305001

प्रकाशक :
पुष्पा शर्मा
Books n Books
51/1509, आचमन, वेदविहार,
लोहाखान, अजमेर-305001
ई-मेल : bnbajmer@gmail.com

संस्करण
प्रथम 2025

ISBN : 978 - 81 - 993854 - 7 - 4

मूल्य - 300/-

रचना -
भारतीय संस्कृति, शिक्षा और दर्शन विभिन्न संकल्पनाएँ

सम्पादक -
डॉ. दिनेश चंद्र शर्मा

आवरण -
श्रीनिवास प्रिन्टोग्राफिक्स, अजमेर

मुद्रक -
श्री श्याम ऑफसेट, किशनगढ़

Computer Architecture

Radhika Sen
Assistant Professor (Deptt. of Science and Technology)
Hukumchand Noble Institute of Science and Technology, Ajmer

1. Introduction

Computer architecture is the conceptual framework that defines the functionality, structure, and implementation of a computer system. It explains how software interacts with hardware and how different components—such as the processor, memory, and input/output devices—work together to execute instructions.

The field can be divided into three primary levels:

1. **Instruction Set Architecture (ISA):** The interface between software and hardware, defining the set of machine instructions, registers, and addressing modes available to programmers.
2. **Micro architecture:** The internal organization of a processor that implements the ISA, including features such as pipelines, execution units, and scheduling methods.
3. **System Design:** Encompasses all other hardware components such as the memory hierarchy, input/output subsystems, and multiprocessor structures.

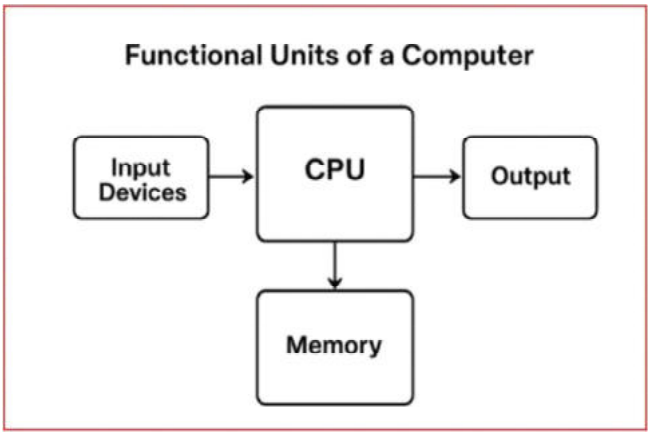
By studying computer architecture, we gain insight into how systems achieve efficiency, scalability, and reliability.

2. The Basic Organization of a Computer System

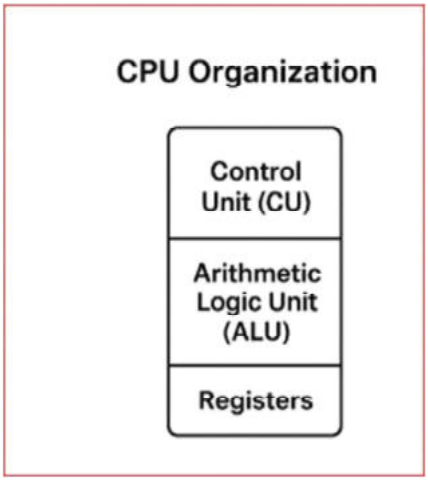
2.1 Functional Components

Every general-purpose computer system consists of the following major units:

- **Input Unit:** Collects raw data and user instructions from devices such as keyboards, scanners, or sensors and passes them into the system.
- **Output Unit:** Presents the processed results through devices like monitors, projecters , printers, speakers etc.



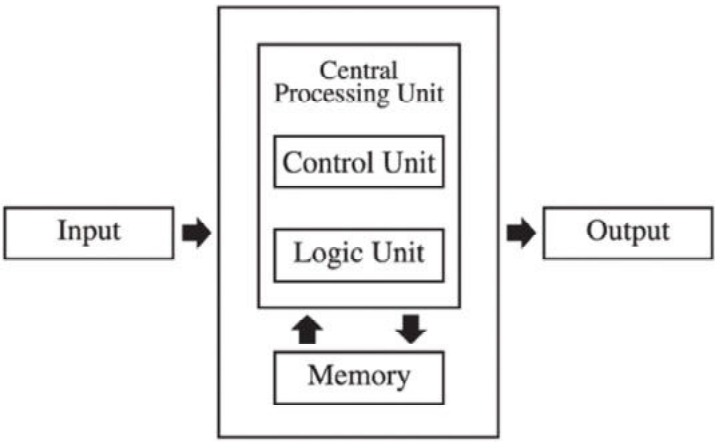
- **Memory Unit:** Stores both data and instructions. It includes fast temporary storage (RAM) and slower, permanent storage (secondary devices such as hard drives).
- **Arithmetic and Logic Unit (ALU):** Handles mathematical calculations and logical comparisons, forming the "computational core" of the processor.
- **Control Unit (CU):** Directs the sequence of operations, interprets instructions, and coordinates the movement of data between components.



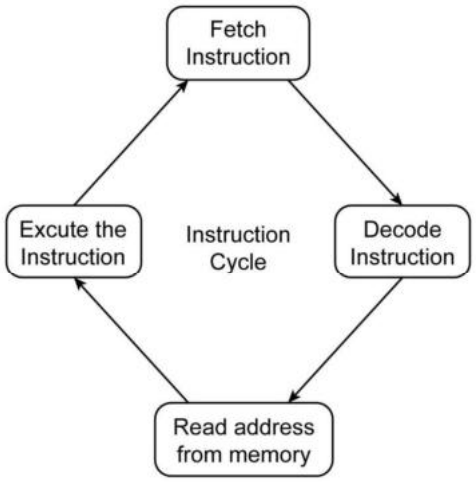
2.2 The Von Neumann Model

Most modern computers follow the **von Neumann architecture**,

which was developed in the mid of 20th century. Its key principles include:



- A single memory holds both data and instructions.
- Instructions are executed sequentially unless control instructions dictate otherwise.
- The processor fetches an instruction from memory, decodes it, executes it, and repeats this cycle.



3. Instruction Set Architecture (ISA)

The ISA defines the rules that a processor understands. It determines

how software is written and how hardware must behave.

3.1 Types of Instructions

1. **Data Movement:** Transfer of data between memory, registers, or I/O (e.g., LOAD, STORE).
2. **Arithmetic Operations:** Addition, subtraction, multiplication, and division.
3. **Logical Operations:** AND, OR, XOR, and NOT.
4. **Control Flow:** Branching and looping through instructions such as jumps, calls, and returns.

3.2 Addressing Modes

Addressing modes describe how the location of data is specified in an instruction. Common types include:

- **Immediate:** Data is embedded directly in the instruction.
- **Direct:** The instruction specifies the exact memory location.
- **Indirect:** The memory address is stored in another register or memory location.
- **Register:** Operands are located in processor registers.
- **Indexed:** Address is generated using a base value and an offset.

3.3 RISC vs. CISC

Instruction sets generally fall into two broad categories:

- **Reduced Instruction Set Computer (RISC):** Emphasizes a small, highly optimized instruction set. Instructions are uniform in length and designed for fast execution (examples: ARM, RISC-V).
- **Complex Instruction Set Computer (CISC):** Provides a wide variety of complex instructions, allowing programs to use fewer lines of code (example: Intel x86).

4. Micro architecture

4.1 Pipelining

Pipelining increases CPU performance by dividing instruction execution into stages such as fetch, decode, execute, memory access, and write-back. Multiple instructions can be in different stages at once, similar to an assembly line.

Challenges in pipelining include:

- **Data hazards** when instructions depend on results of previous ones.
- **Control hazards** caused by branches and jumps.

- **Structural hazards** when two instructions compete for the same hardware resources.

4.2 Superscalar Design

A superscalar processor can issue and execute more than one instruction per clock cycle by using multiple parallel pipelines.

4.3 Out-of-Order Execution

Modern CPUs often execute instructions out of sequence to minimize idle time when some instructions are delayed by data dependencies or resource conflicts.

4.4 Parallelism

- **Instruction-level parallelism (ILP):** Running several instructions concurrently.
- **Thread-level parallelism (TLP):** Executing multiple program threads simultaneously.
- **Data-level parallelism (DLP):** Performing operations on entire arrays or vectors at once (e.g., SIMD instructions).

5. Memory Organization

5.1 Memory Hierarchy

Memory is structured in levels to balance speed, size, and cost:

- **Registers:** Extremely fast storage inside the CPU.
- **Cache Memory (L1, L2, L3):** Small, high-speed memory located close to the CPU.
- **Main Memory (RAM):** Larger but slower than cache.
- **Secondary Storage:** Non-volatile storage such as SSDs or HDDs.
- **Tertiary Storage:** Used for backup or archival purposes, such as tapes or cloud systems.

5.2 Cache Operation

Cache memory temporarily stores frequently accessed data to reduce average memory access time. Its effectiveness is determined by mapping strategies such as:

- Direct mapping
- Fully associative mapping
- Set-associative mapping

5.3 Virtual Memory

Virtual memory extends the apparent size of main memory by using disk space. It provides each process with a logical address space and ensures protection and isolation between processes. Techniques in-

clude **paging** and **segmentation**.

6. Input/Output (I/O) Organization

6.1 I/O Techniques

- **Programmed I/O:** The processor actively controls all data transfers.
- **Interrupt-driven I/O:** Devices send signals to the CPU when they are ready, reducing idle waiting.
- **Direct Memory Access (DMA):** Enables devices to communicate directly with memory, bypassing the CPU for efficiency.

6.2 System Buses

A **system bus** interconnects the CPU, memory, and I/O devices. It consists of:

- **Data bus** for transferring actual data.
- **Address bus** for specifying memory or device locations.
- **Control bus** for coordinating operations.

7. Parallel and Distributed Architectures

To meet growing computational demands, systems often employ parallel or distributed models:

- **SISD (Single Instruction, Single Data):** Traditional uniprocessor system.
- **SIMD (Single Instruction, Multiple Data):** Applies one instruction to multiple data elements simultaneously, as in GPUs.
- **MIMD (Multiple Instruction, Multiple Data):** Multiple processors execute independent instructions on different data streams.
- **Clusters and Clouds:** Networks of interconnected systems that work together to solve large-scale problems.

8. Modern Trends

The evolution of computer architecture has led to several innovations:

- **Multicore processors** that integrate multiple processing units on a single chip.
- **Graphics Processing Units (GPUs)** optimized for large-scale parallel computations.
- **RISC-V**, an open-source ISA offering flexibility and customization.
- **Quantum computing architectures**, which leverage quantum mechanics to solve problems beyond the reach of classical machines.

- **Neuromorphic computing**, inspired by the structure of the human brain, focusing on energy-efficient pattern recognition and learning.

9. Performance Evaluation

Performance is measured using several metrics:

- **Clock speed (GHz):** Number of cycles per second.
- **CPI (Cycles Per Instruction):** Average number of cycles needed to execute an instruction.
- **MIPS (Million Instructions Per Second):** Rate of instruction execution.
- **FLOPS (Floating Point Operations Per Second):** Indicator of performance in scientific applications.
- **Latency and Throughput:** Delay in completing a task versus the number of tasks completed per unit time.

10. Conclusion

Computer architecture forms the foundation of computing systems. From the von Neumann model to advanced multicore processors and distributed networks, it dictates how efficiently computers perform tasks. As the demand for high performance, energy efficiency, and scalability grows, architects increasingly turn to parallel processing, specialized accelerators, and innovative paradigms such as quantum and neuromorphic systems. Understanding computer architecture is therefore essential for both computer scientists and engineers who aim to shape the future of computing technology.

❑❑❑