
UNIT–1: Functional Blocks of a Computer (7 Hours)

1. Functional Blocks of a Computer

A computer system is divided into four main functional blocks:

(a) Central Processing Unit (CPU)

The CPU is the **brain of the computer**. It performs arithmetic, logic, control, and decision-making operations.

Components of CPU:

- **Arithmetic Logic Unit (ALU):** Performs arithmetic (add, subtract, multiply) and logical operations (AND, OR, NOT).
- **Registers:** Small, fast storage locations inside CPU.
- **Control Unit (CU):** Directs operations of the processor.

(b) Memory

Memory stores **data, instructions, and results**.

- **Primary memory:** RAM, ROM
- **Secondary memory:** Hard disk, SSD

(c) Input Subsystem

Used to provide data and instructions to the computer. Examples: Keyboard, Mouse, Scanner

(d) Output Subsystem

Used to display results. Examples: Monitor, Printer

---

2. Control Unit

The Control Unit:

- Fetches instructions
- Decodes instructions
- Generates control signals
- Coordinates data flow between CPU, memory, and I/O

---

3. Instruction Set Architecture (ISA)

ISA acts as an **interface between hardware and software**.

(a) Registers

Registers are high-speed memory elements. Examples:

- **PC (Program Counter)**
- **IR (Instruction Register)**
- **ACC (Accumulator)**
- **MAR, MDR**

(b) Instruction Execution Cycle

Steps:

1. Fetch instruction
2. Decode instruction
3. Execute instruction
4. Write back result

---

4. RTL (Register Transfer Language)

RTL describes **data movement between registers**.

Example:

```
R1 ← R2 + R3
```

This means contents of R2 and R3 are added and stored in R1.

---

5. Addressing Modes

Addressing modes define **how operands are accessed**.

Types:

- Immediate
- Direct
- Indirect
- Register
- Register Indirect
- Indexed
- Relative

---

6. Instruction Set

An instruction set includes:

- Data transfer instructions
- Arithmetic instructions
- Logical instructions
- Control instructions

---

7. Case Study: Instruction Sets of Common CPUs

Examples:

- **x86:** Complex Instruction Set (CISC)
- **ARM:** Reduced Instruction Set (RISC)
- **MIPS:** RISC-based architecture

UNIT–2: Data Representation & Computer Arithmetic (7 Hours)

1. Signed Number Representation

Methods:

- Sign-magnitude
- 1's complement
- 2's complement (most widely used)

---

2. Fixed Point Representation

- Binary point is fixed
- Simple but limited range

---

3. Floating Point Representation

Uses scientific notation:

```
(-1)^sign × mantissa × 2^exponent
```

IEEE 754 standard:

- Single precision (32-bit)
- Double precision (64-bit)

---

4. Character Representation

- ASCII (7-bit)
- Extended ASCII
- Unicode (UTF-8)

---

5. Integer Arithmetic

(a) Addition & Subtraction

Using 2's complement representation.

(b) Ripple Carry Adder

- Carry propagates from LSB to MSB
- Slow for large bit-widths

(c) Carry Look-Ahead Adder

- Reduces delay by predicting carry
- Faster than ripple carry adder

---

6. Multiplication Techniques

(a) Shift-and-Add Method

- Repeated shifting and addition

(b) Booth Multiplier

- Efficient for signed numbers
- Reduces number of additions

(c) Carry Save Multiplier

- Uses carry-save adders
- Faster multiplication

---

7. Division Techniques

(a) Restoring Division

- Partial remainder restored if negative

(b) Non-Restoring Division

- Faster than restoring method

---

8. Floating Point Arithmetic

Operations:

- Addition
- Subtraction
- Normalization
- Rounding

UNIT–3: CPU & Memory System Design (7 Hours)

1. Introduction to x86 Architecture

- CISC architecture
- Variable-length instructions
- Large instruction set
- Multiple addressing modes

---

2. CPU Control Unit Design

(a) Hardwired Control Unit

- Uses logic gates
- Faster
- Difficult to modify

(b) Microprogrammed Control Unit

- Uses control memory
- Slower but flexible

---

3. Case Study: Simple Hypothetical CPU

Includes:

- Minimal registers
- Simple instruction format
- Basic control signals

---

4. Memory System Design

Semiconductor Memory Technologies

- SRAM
- DRAM

- ROM
- Flash Memory

---

5. Memory Organization

- Memory modules
- Address decoding
- Word and byte addressing

UNIT–4: I/O & Interrupts (7 Hours)

1. Peripheral Devices

Characteristics:

- Speed
- Data transfer rate
- Reliability

---

2. Input-Output Subsystem

Provides communication between CPU and peripherals.

---

3. I/O Transfers

(a) Program Controlled I/O

- CPU continuously checks device status

(b) Interrupt Driven I/O

- Device interrupts CPU

(c) DMA (Direct Memory Access)

- Data transfer without CPU involvement

4. Privileged vs Non-Privileged Instructions

- Privileged: Executed in kernel mode
- Non-privileged: Executed in user mode

---

5. Software Interrupts & Exceptions

Used for:

- System calls
- Error handling

---

6. Programs & Processes

Interrupts cause **process state transitions**:

- Running → Waiting
- Waiting → Ready

---

7. I/O Interfaces

- **SCSI**
- **USB**

UNIT–5: Pipelining & Parallel Processing (7 Hours)

1. Pipelining

Pipelining divides instruction execution into stages.

Basic Stages:

- Fetch
- Decode

- Execute
- Memory
- Write back

---

2. Throughput & Speedup

- Throughput increases
- Execution time per instruction decreases

---

3. Pipeline Hazards

Types:

- Structural hazards
- Data hazards
- Control hazards

---

4. Parallel Processors

Multiple processors working simultaneously.

Types:

- SISD
- SIMD
- MISD
- MIMD

---

5. Concurrent Access to Memory

Issues:

- Race condition
- Memory consistency

---

6. Cache Coherency

Ensures all processors see consistent memory data. Protocols:

- MESI
- MOESI

UNIT–6: Memory Organization & Cache (7 Hours)

1. Memory Interleaving

- Divides memory into modules
- Allows parallel access

---

2. Hierarchical Memory Organization

Levels:

- Registers
- Cache
- Main memory
- Secondary storage

---

3. Cache Memory

Small, fast memory between CPU and RAM.

---

4. Cache Size vs Block Size

- Larger block size → fewer misses
- Larger cache → higher hit ratio

---

5. Cache Mapping Techniques

- Direct mapping
- Associative mapping
- Set-associative mapping

---

6. Replacement Algorithms

- LRU
- FIFO
- Random

---

7. Write Policies

- Write-through
- Write-back
