Computer Architecture – Notes

Unit 1: Introduction to Computer Architecture

1. Computer Architecture

Detailed Definition: Computer architecture refers to the design and organization of the various components of a computer system and how they work together to execute instructions. It's like the "blueprint" of a computer that defines what it can do and how it performs tasks.

Key Components:

- **CPU (Central Processing Unit):** The brain of the computer
- **Memory:** Where data and instructions are stored
- **Input/Output Devices:** Keyboard, mouse, monitor, etc.
- **Storage:** Hard drives, SSDs for long-term data storage

2. Functional Units of a Computer

Input Unit

- Receives data and instructions from external devices
- Examples: Keyboard, mouse, scanner
- Converts human-readable data to computer-readable format

Output Unit

- Sends processed data to external devices
- Examples: Monitor, printer, speakers
- Converts computer data to human-readable format

Memory Unit

- Stores data and instructions
- **Primary Memory:** RAM (temporary), ROM (permanent)
- **Secondary Memory:** Hard disk, SSD (long-term)

Arithmetic Logic Unit (ALU)

- Performs mathematical calculations (+, -, ×, ÷)
- Performs logical operations (AND, OR, NOT)
- The "calculator" of the computer

Control Unit (CU)

- Coordinates all computer operations
- Manages data flow between components
- The "traffic controller" of the computer

Central Processing Unit (CPU)

- Combination of ALU and Control Unit
- Executes program instructions
- The "brain" of the computer

3. Von Neumann Architecture

Detailed Definition: A computer design model where both data and programs are stored in the same memory. This is the foundation of most modern computers.

Key Features:

- Single memory for both data and instructions
- Sequential instruction processing
- Stored-program concept
- Four main components: Memory, ALU, Control Unit, Input/Output

How it Works:

1. Fetch instruction from memory
2. Decode the instruction
3. Execute the instruction
4. Store results

4. Generations of Computers

First Generation (1940-1956)

- Used vacuum tubes
- Very large and expensive
- Example: ENIAC

Second Generation (1956-1963)

- Used transistors
- Smaller and more reliable
- Example: IBM 1401

Third Generation (1964-1971)

- Used integrated circuits (ICs)
- Even smaller and faster
- Example: IBM System/360

Fourth Generation (1971-Present)

- Used microprocessors
- Personal computers emerged

- Example: Modern PCs, laptops

Fifth Generation (Present and Beyond)

- Artificial Intelligence
- Parallel processing
- Quantum computing

5. Performance Factors

Clock Speed

- Measured in Hertz (Hz)
- How many cycles per second
- Higher clock speed = faster processing

Word Size

- Number of bits processed at once
- 32-bit vs 64-bit processors
- Larger word size = better performance

Cache Memory

- Small, fast memory near CPU
- Stores frequently used data
- Reduces time to access main memory

Number of Cores

- Multiple processors on one chip
- Dual-core, quad-core, octa-core
- More cores = better multitasking

Video Links:

- **English:** [Computer Architecture Basics]
- **Hindi:** [Computer Architecture in Hindi]

Unit 2: Digital Logic and Circuits

1. Logic Gates

Detailed Definition: Basic building blocks of digital circuits that perform logical operations on binary inputs (0 and 1) to produce a single binary output.

Basic Gates:

- **AND Gate:** Output 1 only if all inputs are 1
- **OR Gate:** Output 1 if at least one input is 1

- **NOT Gate:** Output is opposite of input (inverter)

Universal Gates:

- **NAND Gate:** AND followed by NOT
- **NOR Gate:** OR followed by NOT

Other Gates:

- **XOR Gate:** Output 1 if inputs are different
- **XNOR Gate:** Output 1 if inputs are same

2. Boolean Algebra

Detailed Definition: A mathematical system for logical operations using binary variables (TRUE/FALSE, 1/0) and logical operators (AND, OR, NOT).

Basic Laws:

- **Commutative Law:** $A+B = B+A$, $A \cdot B = B \cdot A$
- **Associative Law:** $(A+B)+C = A+(B+C)$
- **Distributive Law:** $A \cdot (B+C) = A \cdot B + A \cdot C$
- **Identity Law:** $A+0 = A$, $A \cdot 1 = A$

3. Combinational Logic Circuits

Detailed Definition: Digital circuits where output depends only on current inputs (no memory).

Examples:

- **Adders:** Half adder, Full adder
- **Multiplexers:** Selects one of many inputs
- **Decoders:** Converts coded inputs to outputs

4. Sequential Logic Circuits

Detailed Definition: Digital circuits where output depends on both current inputs and previous states (have memory).

Examples:

- **Flip-flops:** Basic memory elements
- **Registers:** Store multiple bits
- **Counters:** Count clock pulses

5. Flip-Flops

Detailed Definition: Basic memory elements that can store one bit of information.

Types:

- **SR Flip-flop:** Set-Reset

- **JK Flip-flop:** Improved version of SR
- **D Flip-flop:** Delay (stores input value)
- **T Flip-flop:** Toggle

6. Registers

Detailed Definition: Groups of flip-flops that can store multiple bits of data.

Types:

- **Shift Register:** Shifts bits left or right
- **Parallel Register:** Loads all bits at once

7. Karnaugh Maps (K-Maps)

Detailed Definition: A graphical method used to simplify Boolean algebra expressions.

Benefits:

- Visual representation
- Easy minimization
- Reduces circuit complexity

Video Links:

- **English:** Digital Logic Circuits
- **Hindi:** Logic Gates in Hindi

Unit 3: Data Representation and Arithmetic

1. Number Systems

Binary System (Base 2)

- Uses only 0 and 1
- Fundamental computer language
- Example: $1011_2 = 11_{10}$

Octal System (Base 8)

- Uses digits 0-7
- Shorter representation of binary
- Example: $75_8 = 61_{10}$

Decimal System (Base 10)

- Uses digits 0-9
- Everyday number system
- Example: 123_{10}

Hexadecimal System (Base 16)

- Uses digits 0-9 and A-F
- Compact representation of binary
- Example: $A5F_{16} = 2655_{10}$

2. Data Representation

Integer Representation

- **Signed Magnitude:** Leftmost bit for sign
- **1's Complement:** Flip all bits
- **2's Complement:** Flip bits and add 1 (most common)

Floating Point Representation

- For real numbers (decimals)
- **Format:** Sign + Exponent + Mantissa
- Example: IEEE 754 standard

Character Representation

- **ASCII:** 7-bit code for English characters
- **Unicode:** Supports multiple languages

3. Computer Arithmetic

Binary Addition

```
   1011  (11)
+  0110  (6)
   -----
  10001  (17)
```

Binary Subtraction

- Using 2's complement method

Binary Multiplication

- Similar to decimal multiplication
- Uses shifting and adding

Binary Division

- Similar to decimal division
- Uses shifting and subtracting

4. Complements

1's Complement

- Flip all bits (0→1, 1→0)

- Example: 1010 → 0101

2's Complement

- 1's complement + 1
- Used for signed numbers
- Example: 1010 → 0101 + 1 = 0110

Video Links:

- **English:** Number Systems and Data Representation
- **Hindi:** Data Representation in Hindi

Unit 4: Basic Computer Organization and Design

1. Bus Organization

Detailed Definition: A communication system that transfers data between computer components using shared pathways.

Types of Buses:

- **Data Bus:** Carries data between components
- **Address Bus:** Carries memory addresses
- **Control Bus:** Carries control signals

2. Micro-operations

Detailed Definition: Elementary operations performed on data stored in registers.

Types:

- **Register Transfer:** Move data between registers
- **Arithmetic:** Add, subtract, increment
- **Logic:** AND, OR, NOT operations
- **Shift:** Shift bits left or right

3. Instruction Codes and Formats

Detailed Definition: Binary codes that tell the computer what operation to perform.

Instruction Parts:

- **Operation Code (Opcode):** What to do
- **Operand:** Data to operate on
- **Addressing Mode:** How to find operand

4. Hardwired vs Microprogrammed Control

Hardwired Control

- Control signals generated by logic circuits

- Fast but inflexible
- Used in RISC processors

Microprogrammed Control

- Control signals stored in memory
- Slower but flexible
- Used in CISC processors

5. Instruction Cycle

Detailed Definition: The sequence of steps a CPU follows to execute an instruction.

Steps:

1. **Fetch:** Get instruction from memory
2. **Decode:** Understand what to do
3. **Execute:** Perform the operation
4. **Store:** Save the result

6. Pipelining

Detailed Definition: A technique where multiple instructions are overlapped in execution to improve performance.

Stages:

- Instruction Fetch (IF)
- Instruction Decode (ID)
- Execute (EX)
- Memory Access (MEM)
- Write Back (WB)

7. Interrupts

Detailed Definition: Signals that temporarily halt normal program execution to handle urgent events.

Types:

- **Hardware Interrupts:** From external devices
- **Software Interrupts:** From programs
- **Exceptions:** Error conditions

Video Links:

- **English:** Computer Organization
- **Hindi:** CPU Organization in Hindi

Unit 5: Processors

1. General Register Organization

Detailed Definition: CPU design that uses multiple general-purpose registers for temporary data storage.

Benefits:

- Faster data access
- Reduced memory access
- Better performance

2. Stack Organization

Detailed Definition: A LIFO (Last-In-First-Out) data structure used for temporary storage.

Operations:

- **PUSH:** Add item to stack
- **POP:** Remove item from stack

3. Addressing Modes

Detailed Definition: Different ways of specifying the location of operands in instructions.

Types:

- **Immediate:** Operand in instruction itself
- **Direct:** Address in instruction
- **Indirect:** Address of address in instruction
- **Register:** Operand in register
- **Indexed:** Address + index register

4. RISC vs CISC Architectures

RISC (Reduced Instruction Set Computer)

- Fewer, simpler instructions
- Fixed instruction length
- Hardwired control
- Example: ARM processors

CISC (Complex Instruction Set Computer)

- Many, complex instructions
- Variable instruction length
- Microprogrammed control

- Example: Intel x86 processors

5. Multicore Processors

Detailed Definition: A single computing component with two or more independent processing units (cores).

Benefits:

- Parallel processing
- Better multitasking
- Improved performance

6. Graphics Processing Units (GPUs)

Detailed Definition: Specialized processors designed for rendering graphics and parallel computations.

Features:

- Thousands of simple cores
- Massive parallelism
- Used in gaming, AI, scientific computing

Video Links:

- **English:** Processor Architecture
- **Hindi:** RISC vs CISC in Hindi

Unit 6: Memory and Input/Output Organization

1. Memory Hierarchy

Detailed Definition: The arrangement of different types of memory in levels based on speed, size, and cost.

Levels (Fastest to Slowest):

1. **Registers:** Inside CPU, very fast, very small
2. **Cache:** Fast, small, expensive
3. **Main Memory (RAM):** Medium speed, medium size
4. **Secondary Storage:** Slow, large, cheap (HDD, SSD)

2. Types of Memory

RAM (Random Access Memory)

- Volatile (loses data when power off)
- Read and write capability
- **Types:** SRAM (faster), DRAM (cheaper)

ROM (Read Only Memory)

- Non-volatile (retains data)
- Read-only normally
- **Types:** PROM, EPROM, EEPROM

3. Cache Memory

Detailed Definition: Small, fast memory between CPU and main memory that stores frequently used data.

Levels:

- **L1 Cache:** Inside CPU, fastest
- **L2 Cache:** Slower than L1, larger
- **L3 Cache:** Shared between cores

4. Input-Output Interface

Detailed Definition: Hardware and software that enables communication between CPU and I/O devices.

Functions:

- Data buffering
- Error detection
- Protocol conversion

5. Modes of Data Transfer

Programmed I/O

- CPU directly controls I/O
- Simple but inefficient
- CPU waits for I/O completion

Interrupt-driven I/O

- I/O device interrupts CPU when ready
- Better CPU utilization
- Still involves CPU for data transfer

Direct Memory Access (DMA)

- Special controller transfers data directly to memory
- CPU free during transfer
- Most efficient method

Video Links:

- **English:** Memory Hierarchy and I/O

- **Hindi:** Computer Memory in Hindi

Unit 7: Advanced Topics in Computer Architecture

1. Performance Measurement

Clock Rate

- Cycles per second (Hz)
- Higher = faster, but not always better

Instructions Per Second

- MIPS (Million Instructions Per Second)
- FLOPS (Floating Point Operations Per Second)

Benchmarks

- Standard tests to compare performance
- Example: SPEC, Geekbench

2. Parallel Architectures

Detailed Definition: Computer systems with multiple processing elements that work simultaneously to solve problems.

Types:

- **SISD:** Single Instruction, Single Data
- **SIMD:** Single Instruction, Multiple Data
- **MISD:** Multiple Instruction, Single Data
- **MIMD:** Multiple Instruction, Multiple Data

3. Distributed Architectures

Detailed Definition: Multiple computers connected via network working together as a single system.

Examples:

- **Cluster Computing:** Group of connected computers
- **Grid Computing:** Geographically distributed resources
- **Cloud Computing:** Internet-based computing services

4. Modern Design Trends

Multi-core Processors

- Multiple processors on single chip
- Standard in modern computers

Quantum Computing

- Uses quantum bits (qubits)
- Can solve complex problems faster

Neuromorphic Computing

- Mimics human brain structure
- Efficient for AI applications

3D Chip Stacking

- Multiple layers of circuits
- Saves space, improves performance

5. Emerging Technologies

- **Optical Computing:** Uses light instead of electricity
- **DNA Computing:** Uses biological molecules
- **Reversible Computing:** Reduces energy consumption

Video Links:

- **English:** Advanced Computer Architecture
- **Hindi:** Parallel Processing in Hindi

Important Concepts Summary

Memory Types Comparison:

Type	Speed	Volatility	Cost	Use
Registers	Fastest	Volatile	Highest	CPU internal
Cache	Very Fast	Volatile	High	CPU-Memory bridge
RAM	Fast	Volatile	Medium	Main memory
ROM	Slow	Non-volatile	Low	Firmware
SSD	Medium	Non-volatile	Medium	Storage
HDD	Slowest	Non-volatile	Lowest	Mass storage

Processor Performance Factors:

1. **Clock Speed:** GHz rating
2. **Cache Size:** L1, L2, L3 cache

3. **Number of Cores:** Parallel processing capability
4. **Architecture:** RISC vs CISC
5. **Instruction Set:** Complexity and efficiency

Data Transfer Methods:

1. **Programmed I/O:** Simplest, CPU-intensive
2. **Interrupt-driven I/O:** Better CPU utilization
3. **DMA:** Most efficient, uses separate controller

Career Applications:

- CPU Design Engineer
- Computer Architect
- Embedded Systems Engineer
- Hardware Developer
- Systems Analyst

Final English Video: Computer Architecture Full Course
Final Hindi Video: Computer Architecture Complete Course