Understanding the CPU (Central Processing Unit)

1 What is a CPU?

The CPU (Central Processing Unit) is considered the **brain** of the computer. It is responsible for executing instructions from programs and coordinating the activities of all hardware components.

2 Main Components of a CPU

1. Control Unit (CU)

- Directs the operation of the processor.
- Fetches instructions from memory.
- Decodes instructions and coordinates their execution.
- Manages data flow between CPU and other components.

2. Arithmetic Logic Unit (ALU)

- Performs arithmetic operations (add, subtract, multiply, divide).
- Executes logical operations (AND, OR, NOT, comparisons).

3. Registers

- Small, fast memory units inside the CPU.
- Temporarily store instructions, data, and addresses.
- Types include:
  - **Accumulator (ACC):** Stores intermediate results.
  - **Program Counter (PC):** Holds the address of the next instruction.
  - **Instruction Register (IR):** Holds the current instruction.
  - **General Purpose Registers (R0, R1, ...):** Store operands or results.

3 Basic CPU Operations: The Instruction Cycle

1. **Fetch** – Retrieve the instruction from memory.
2. **Decode** – Understand the instruction.
3. **Execute** – Carry out the operation.
4. **Store (Write-back)** – Save the result if required.

4 CPU Performance Factors

- **Clock Speed (GHz):** Determines how many cycles are executed per second.
- **Instruction Set Architecture (ISA):** Defines the structure of CPU instructions.
- **Pipelining:** Splits execution into stages to enhance performance.
- **Cache Memory:** Small, high-speed memory near CPU for frequently accessed data.
- **Number of Cores:** More cores enable better parallel task execution.

5 Fun Fact

Modern CPUs can contain **billions of transistors** and multiple cores (e.g., dual-core, quad-core, octa-core), allowing them to process many tasks in parallel.

6 Instruction Cycle: Step-by-Step

1. Fetch

- The Program Counter (PC) holds the address of the next instruction.
- This address is sent to memory; the instruction is retrieved and stored in the Instruction Register (IR).
- The PC is incremented to point to the next instruction.

2. Decode

- The Control Unit decodes the instruction in the IR.
- It identifies:
  - **Opcode:** Operation to perform.
  - **Operands:** Data or memory addresses involved.

3. Execute

- ALU or another unit performs the operation (e.g., ADD, SUB).

4. Memory Access (if needed)

- If instruction involves memory (e.g., LOAD/STORE), the appropriate data is accessed.

5. Write-back (optional)

- Results are stored in a register or memory location.

7 The Continuous Cycle

The CPU returns to the **Fetch** step after each instruction is executed, creating a continuous loop called the **Instruction Cycle**.

8 Instruction Cycle Example: `ADD R1, R2, R3`

Assuming a RISC-style format where `R1 = R2 + R3`:

Step	**Action**
Fetch	CPU fetches the instruction `ADD R1, R2, R3` from memory using the PC.
Decode	CU decodes the instruction: opcode = ADD, operands = R2, R3, R1.
Execute	ALU adds the contents of R2 and R3.
Write-back	Result is stored in R1.
PC Update	PC is incremented to point to the next instruction.

Table 1: Instruction Cycle Example: ADD Operation