## **Core CPU Operations**

### **1. Fetch**

* The **Control Unit** fetches the instruction from **memory (RAM or cache)**.
* The **Program Counter (PC)** holds the address of the next instruction.

### **2. Decode**

* The **Instruction Decoder** interprets the fetched instruction.
* It determines **what operation** is to be performed and **which operands** are needed.

### **3. Execute**

* The **ALU (Arithmetic Logic Unit)** or **FPU (Floating Point Unit)** performs the required operation (add, subtract, AND, OR, etc.).
* May involve accessing memory or registers.

### **4. Memory Access (if needed)**

* If the instruction involves data fetch/store, the memory unit is accessed.

### **5. Write-back**

* The result is written back to a **register** or **memory**.

## **🛠️ CPU Components Involved**

| **Component** | **Role** |
| --- | --- |
| **Program Counter (PC)** | Holds address of the next instruction |
| **Instruction Register (IR)** | Holds current instruction |
| **Control Unit (CU)** | Directs all operations |
| **ALU** | Performs arithmetic/logic |
| **Registers** | Fast local storage |
| **Bus Interface** | Connects CPU to memory and I/O |

## **🔁 Repeated Cycle: The Instruction Cycle**

The CPU keeps repeating the **fetch-decode-execute** cycle millions/billions of times per second (controlled by the clock).

## **Example:**

For the instruction: ADD R1, R2, R3  
 ➤ CPU fetches this instruction from memory  
 ➤ Decodes it (add values in R2 and R3)  
 ➤ Executes it using ALU  
 ➤ Stores result in R1