Architecture vs. Microarchitecture

What is Architecture (Instruction Set Architecture — ISA)?

Architecture refers to what a programmer sees. It defines what the CPU can do, not how it is built inside.

Architecture includes:

- Programmer-visible state: Memory, registers

- Operations / Instructions: ADD, SUB, LOAD, STORE, JUMP, etc.

- Execution semantics: How instructions behave, interrupts

- Input/Output: How the CPU communicates with external devices

- Data Types/Sizes: Integer size, float size, pointer size

What is Microarchitecture (Organization)?

Microarchitecture describes how the CPU is internally designed to implement the ISA.

It includes:

- Trade-offs in implementing the ISA based on speed, energy, cost

- Pipeline depth

- Number of pipelines

- Cache size

- Silicon area

- Power usage

- Execution ordering (in-order / out-of-order)

- Bus widths

- ALU widths

Analogy:

Architecture = Language rules (grammar)

Microarchitecture = How you speak (fast, slow, accent)

Both follow the same rules, but the implementation can vary.