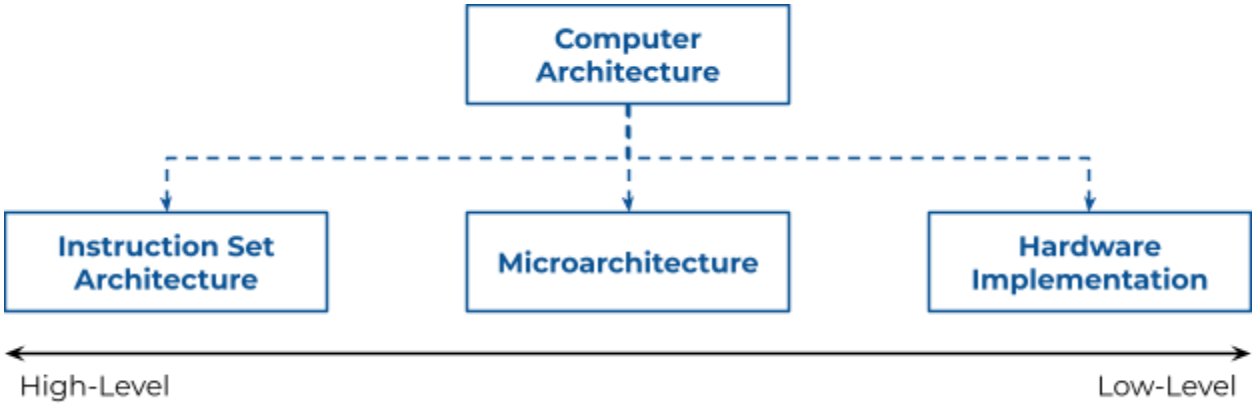
CSARCH1 Introduction to Computer Organization and Architecture

Computer Organization and Architecture



Instruction Set Architecture (ISA)

The part of computer architecture that is visible to the programmer. It **defines the instructions a processor can execute** and the way a programmer interacts with the system.

- **Instruction Set**: The set of commands a processor understands, such as ADD, SUB, MUL, or LOAD.
    - Determines whether the system uses:
        - **RISC (Reduced Instruction Set Computing)**: Simple and fixed-length instructions.
        - **CISC (Complex Instruction Set Computing)**: Complex and variable-length instructions.
- **Data Types**: Specifies how data is represented (e.g., integers, floating-point numbers, characters).
    - Example: Is an integer 32-bit or 64-bit?
- **Registers**: Defines the number, size, and usage of registers available for computations.
    - Example: Does the ISA have general-purpose registers or specialized ones?
- **Memory Addressing**: Specifies how memory locations are accessed.
    - Example: Supports byte or word addressing, endianness (big-endian vs. little-endian).
- **Addressing Modes**: Determines how operands are specified in instructions.
    - Example: Direct addressing, indirect addressing, or auto-increment.

Computer Organization or Microarchitecture

Refers to the implementation details of the ISA. It determines **how the ISA is realized in hardware**, including the design of components like the processor, memory hierarchy, and control logic.

- **Execution Units**: Design of components like the Arithmetic Logic Unit (ALU) or Floating-Point Unit (FPU) to perform operations.
- **Pipelining**: Uses techniques to break down instructions into sub-instructions. These processes are then further optimized by scheduling (a.k.a. load balancing).
- **Memory Hierarchy**: Organizes memory into levels (registers, cache, RAM, disk) to balance speed and cost.
- **Interconnects**: Determines how components like processors and memory are connected (e.g., buses, crossbars).

Hardware Implementation

Refers to the **physical realization of the architecture**. It involves designing and fabricating the actual circuits, chips, and boards that make up the computer.

- **Logic Design**: Implementation of combinational (e.g., adders, multiplexers) and sequential circuits (e.g., registers, counters).
- **Fabrication**: Manufacturing of components using technologies like CMOS, VLSI, or custom ASICs.
- **Packaging**: How chips are assembled, cooled, and connected on a motherboard or within a data center.
- **Physical Constraints**: Balancing power consumption, heat dissipation, and physical space.