COAHub

Computer Architecture Cheat Sheet

Basic Components of a Computer System

- **CPU (Central Processing Unit):**
  - **Arithmetic Logic Unit (ALU):** Performs arithmetic and logical operations.
  - **Control Unit (CU):** Directs operations, fetches instructions, and controls data flow .
  - **Registers:** Small, fast storage locations within the CPU. Key registers include:
    - **Accumulator (AC):** Stores results of operations .
    - **Program Counter (PC):** Points to the next instruction to execute .
    - **Memory Address Register (MAR):** Holds memory addresses for data/instructions .
    - **Memory Data Register (MDR):** Temporarily holds data fetched from or to be written to memory .
    - **Current Instruction Register (CIR):** Holds the current instruction being decoded/executed .
    - **Status/Flag Register:** Indicates conditions like zero, carry, overflow.
- **Memory Hierarchy:**
  - **Cache:** Small, fast memory close to the CPU for frequently accessed data.
  - **Main Memory (RAM):** Stores data and instructions for currently running programs.
  - **Secondary Storage:** Large, slower storage (HDD, SSD, optical disks) .
- **Input/Output (I/O) Devices:** Interfaces for user interaction and external communication (keyboard, mouse, monitor, printer).
- **Bus System:**
  - **Data Bus:** Transfers actual data.
  - **Address Bus:** Transfers memory addresses.
  - **Control Bus:** Transfers control signals and status information.

Key Computer Architectures

Architecture	Memory for Data & Instructions	Speed & Usage
Von Neumann	Shared	Simpler, traditional computers[2]
Harvard	Separate	Faster, used in embedded systems[2]

Instruction Cycle

1. **Fetch:** Get instruction from memory (PC → MAR, fetch to MDR, then CIR)[1].
2. **Decode:** CU interprets the instruction.
3. **Execute:** ALU performs operation, result stored in register or memory.
4. **Store:** Result written back to memory or register if needed.

Types of Instructions

- **Data Transfer:** MOV, LOAD, STORE
- **Arithmetic:** ADD, SUB, MUL, DIV
- **Logical:** AND, OR, NOT
- **Control:** JMP, CALL, RET

Addressing Modes

- **Immediate:** Operand is part of instruction
- **Direct:** Address of operand given directly
- **Indirect:** Address field points to a memory location containing the address of the operand
- **Indexed:** Uses base address plus offset

Memory Organization

- **Cache:** Closest to CPU, fastest access.
- **RAM:** Volatile, used for active processes.
- **Secondary Storage:** Non-volatile, large capacity.

Performance Concepts

- **Clock Speed:** Determines how many instructions per second CPU can execute.
- **Pipelining:** Overlapping instruction execution to improve throughput.
- **RISC vs. CISC:**
    - **RISC:** Simple instructions, faster, more registers
    - **CISC:** Complex instructions, fewer registers, more addressing modes

The Von Neumann Model

- **Main Memory:** Stores both data and instructions.
- **ALU:** Performs binary operations.
- **Control Unit:** Executes instructions from memory.
- **I/O Devices:** Managed by CU.

Quick Reference Table

Component	Function
ALU	Arithmetic/logic operations
CU	Directs data flow, fetch/decode/execute
Registers	Fast, small storage for instructions/data
Cache	High-speed memory for frequent access
RAM	Main memory for running programs
Secondary	Long-term storage
Data Bus	Transfers data
Address Bus	Transfers addresses
Control Bus	Transfers control signals