

## **Arquitetura de Von Neumann - Conceito**

Permite que programas e dados sejam armazenados na mesma memória.

Facilitou que uma mesma máquina execute diferentes programas.

### **Propôs que (PROGRAMA ARMAZENADO):**

- As instruções (programas) são codificadas em **binário (0 e 1)** e armazenadas na memória, junto com os dados.
- Ao processarem o programa, o computador busca as instruções diretamente da memória.

**MEMÓRIA:** Armazena dados e programas, temporariamente ou permanentemente.

### **Unidade Lógica e Aritmética (ULA):**

Executa operações matemáticas (soma, subtração) e lógicas (AND, OR, etc.).

### **Unidade de Controle (UC):**

Controla todas as operações: qual dado será buscado, qual operação a ULA irá executar e quando interagir com dispositivos de entrada/saída.

### **Registrador:**

Memória interna à CPU, onde os dados que foram buscados na memória são armazenados. Armazena resultados intermediários rapidamente.

### **Unidades de Entrada e Saída (E/S):**

Permitem a comunicação do computador com o exterior.

# Fluxo de Execução de um Programa

1. **Busca da Instrução:**  
A **Unidade de Controle (UC)** solicita à memória a próxima instrução.
2. **Decodificação:**  
A UC recebe e **decodifica** a instrução, identificando qual operação realizar.
3. **Execução:**  
A UC envia a instrução para a **ULA** executar.
4. **Busca de Dados:**  
Se necessário, a UC solicita dados à memória ou aos dispositivos de entrada.
5. **Saída:**  
Caso necessário, a UC envia o resultado para um dispositivo de saída.

## Gargalo de Von Neumann:

Diz respeito ao fato do acesso à memória ser o principal fator limitante no desempenho de um computador que segue a arquitetura de Von Neumann, pois os dados e instruções usam o mesmo barramento para se comunicar com a CPU.

## Lei de Moore:

A velocidade de um chip de computador dobraria a cada 18 meses mantendo o mesmo custo.

## Arquitetura de Harvard:

Há 2 memórias:

- Memória de Instruções: onde ficam os programas.
- Memória de Dados: onde ficam os dados manipulados.

Tem barramentos separados para dados e instruções

## Unidade Central de Processamento (UCP):

A UCP, também conhecida como CPU, é o "cérebro" do computador.

## Ciclo de Execução da CPU (Ciclo de Máquina):

- **Busca (Fetch):** A CPU busca a próxima instrução na memória.
- **Decodificação (Decode):** A CPU interpreta a instrução, identificando a operação a ser realizada.
- **Execução (Execute):** A CPU realiza a operação, que pode ser um cálculo, movimentação de dados, ou controle.

## Principais Componentes da UCP e Funções:

- **ULA (Unidade Lógica e Aritmética):** Realiza operações matemáticas e lógicas.
- **Unidade de Controle:** Gerencia e coordena as operações da CPU e dos demais componentes.
- **Registradores:** Pequenas memórias internas usadas para armazenar dados temporários e resultados intermediários.
- **Barramentos:** Canais que fazem a comunicação entre CPU, memória e dispositivos.

**Conjunto de Instruções:** É o conjunto de comandos que o processador entende e executa, escritos em linguagem de máquina (binário).

**Instrução de Máquina:** é a unidade básica de instruções que um computador pode executar diretamente

## Tipos de Registradores

- **Acumulador:** Armazena resultados temporários de operações.
- **Registradores de Dados:** Guardam dados usados nas operações.
- **Registradores de Endereço:** Guardam endereços da memória a serem acessados.

- **Contador de Programa (PC):** Guarda o endereço da próxima instrução a ser executada.
- **Registrador de Instrução (IR):** Guarda a instrução atual enquanto é decodificada e executada.
- **Registrador de Status (Flags):** Indica o estado da CPU (se houve erro, resultado zero, negativo, overflow, etc.).

## Unidade de Controle : Microprogramada vs. Hardwired

- **Microprogramada:**  
Lógica feita em software (microcódigo), armazenada numa memória de controle. Baseada em uma memória de controle (ROM ou RAM) que contém microinstruções.  
Mais flexível, fácil de atualizar, mas mais lenta.
- **Hardwired:**  
Lógica feita em hardware (circuitos físicos). Baseada em portas lógicas, flip-flops e outros componentes de hardware fixo.  
Mais rápida, porém difícil de modificar.

### Microprogramada:

Para arquiteturas **CISC**, com muitas instruções complexas.

### Hardwired:

Para arquiteturas **RISC**, que priorizam velocidade.

## UNIDADE DE CONTROLE: FUNCIONAMENTO BÁSICO:

- Recebe uma instrução em linguagem de máquina.
- Traduz essa instrução em microoperações.
- Gera os sinais de controle para execução.

## CPU: Etapas principais do ciclo de instrução

- Busca de Instrução (BI)
- Decodificação da Instrução (DI)
- Cálculo de Operandos (CO)
- Busca de Operandos (BO)
- Execução da Instrução (EI)
- Escrita de Operando (EO)

## Gerenciamento da CPU:

- A **CPU** gerencia o processamento, transferência, armazenamento e controle de dados e recursos do computador.
- O **Sistema Operacional (SO)** gerencia esses recursos.
- O **SO** gerencia os processos a serem executados e colabora no gerenciamento de energia da CPU.
- **First Come First Served (FCFS)**: É um algoritmo de escalonamento de processos que prioriza a execução do primeiro processo que chegar;
- **Shorter Job First (SJF)**: É um algoritmo de escalonamento de processos que prioriza o processo com o menor tempo de execução;
- **Multiple Level Queue (MLQ)**: É um algoritmo de agendamento de CPU que organiza processos em diferentes níveis de prioridade.
- **Power-Saving State Control**: É o processo de gerenciar o consumo de energia de um sistema ou dispositivo, ativando o baixo consumo de energia quando não está sendo utilizado.
- **Dynamic Voltage Scalling**: É uma técnica de ajustar a voltagem fornecida a um processador de acordo com suas necessidades de processamento, buscando economizar energia.

## Unidade Lógica e Aritmética:

A ULA (Unidade Lógica e Aritmética) é um circuito capaz de realizar operações.

A ULA realiza operações lógicas e aritméticas sobre valores inteiros. Por isso, também é conhecida como unidade de inteiros.

As operações executadas em uma ULA podem ser operações aritméticas e lógicas.

Os valores inteiros e reais representados nas máquinas atuais são em complemento 2.

**Tamanho da palavra:**

É o número de bits que um processador consegue manipular de uma vez só em uma operação.

### **Pipeline:**

Pipeline é uma técnica usada em processadores para executar(sobrepor) várias instruções ao mesmo tempo, dividindo a execução em etapas.

## **RISC X CISC:**

Linguagens de programação evoluíram, aumentando a complexidade das instruções e criando um "gap semântico" entre elas e o hardware.

**GAP SEMÂNTICO:** Refere-se à diferença ou distância entre as instruções usadas em linguagens de programação de alto nível e as instruções de máquina de baixo nível que o hardware do computador realmente entende.

**CISC** (Complex Instruction Set Computer) surgiu para reduzir esse gap com instruções mais complexas.

**CISC** mostrou-se ineficiente, pois a maioria dos comandos de baixo nível eram simples e compiladores não otimizavam instruções complexas.

**RISC** (Reduced Instruction Set Computer) prioriza simplicidade e velocidade com menos instruções e modos de endereçamento.

- **RISC** usa LOAD/STORE para acessar a memória, operando principalmente em registradores.
- **RISC** utiliza "pipelining" para processar instruções mais rápido, com atenção às dependências de dados.
- **RISC** tem unidade de controle por hardware para execução direta de instruções, aumentando a velocidade.
- **RISC** usa instruções de formato fixo e um conjunto reduzido de instruções e endereçamentos para simplificar o decodificador.
- **RISC** emprega múltiplos registradores para reduzir acessos à memória e requer compiladores complexos para otimizar o código.



