

Nome: Barbara Letícia da Silva.

Prontuário: CJ3029921.

1º Semestre.

Professor: Josivan Pereira da Silva.

**Resumo sobre o vídeo: Projeto de Sistemas Computacionais – Aula 01 –
Arquitetura e Organização de Computadores.**

Sumário:

1. Introdução

2. Arquitetura vs. Organização de computadores

2.1 Definições e distinções

2.2 Compatibilidade e famílias de computadores

3. Estrutura e funcionalidade básica

3.1 As quatro funções primárias

3.2 Componentes principais

4. Evolução histórica e Marcos tecnológicos

4.1 Gerações de computadores

4.2 Lei de *Moore* e seu impacto

5. Desafios contemporâneos em arquitetura

5.1 O problema do Gargalo de memória

5.2 Limites térmicos e energéticos

5.3 Paralelismo e *Multicore*

6. Tendências emergentes

6.1 Sistemas Embarcados e *IoT*

6.2 Computação heterogênea

6.3 Arquiteturas Pós-*Moore*

7. Conclusão e perspectivas

1. Introdução

Esta aula inicial do curso de **Projeto de Sistemas Computacionais** estabelece os alicerces conceituais para a compreensão da arquitetura e organização de computadores. Partindo de definições fundamentais, o conteúdo traça um panorama histórico da evolução tecnológica, explora a estrutura funcional dos sistemas computacionais e discute os desafios contemporâneos no campo do *design* de *hardware*. O objetivo é proporcionar aos estudantes uma base sólida para entender tanto os princípios teóricos quanto as implementações práticas que caracterizam os computadores modernos, preparando o terreno para tópicos mais avançados que serão abordados posteriormente no curso.

2. Arquitetura vs. Organização de computadores

2.1 Definições e distinções

- **Arquitetura do computador:** Refere-se aos atributos lógicos e funcionais visíveis ao programador, que determinam como o *software* interage com o *hardware*. Isso engloba:
 - Conjunto de instruções (*ISA - Instruction Set Architecture*)
 - Modos de endereçamento
 - Representação de dados (tamanho de palavras, formatos)

- Mecanismos de entrada/saída
- **Exemplo:** A decisão de incluir uma instrução específica (como multiplicação em *hardware*) é uma característica arquitetural.
- **Organização do computador:** Trata da implementação física da arquitetura, incluindo:
 - Unidades operacionais (*ALU*, *UC*, registradores)
 - Tecnologia de interconexão (barramentos, hierarquia de memória)
 - Sinais de controle e temporização
 - **Exemplo:** Dois computadores podem implementar a mesma arquitetura *x86* mas diferir na organização (**ex.:** *pipeline* mais profundo, *cache* maior).

2.2 Compatibilidade e famílias de computadores

A separação entre arquitetura e organização permite a **compatibilidade binária** dentro de uma mesma família de processadores. Por **exemplo:**

- A série **Intel x86** mantém compatibilidade desde os processadores 8086 até os *Core i9* modernos, apesar de mudanças radicais na organização interna.
- O **IBM System/370** demonstrou essa flexibilidade, permitindo que *software* escrito para gerações antigas funcionasse em máquinas novas com organizações distintas.

3. Estrutura e funcionalidade básica

3.1 As quatro funções primárias

1. Processamento de dados:

- Operações aritméticas (soma, multiplicação) e lógicas (*AND*, *OR*).
- Implementado pela *ULA* (Unidade Lógica Aritmética).

2. Armazenamento de dados:

- **Memória primária (*RAM*):** Volátil, de acesso rápido.
- **Memória secundária (*SSD/HDD*):** Não volátil, alta capacidade.

- **Registradores:** Armazenamento temporário dentro da *CPU*.

3. Movimentação de dados:

- Comunicação entre *CPU*, memória e periféricos via **barramentos** (dados, endereços, controle).
- Dispositivos de E/S (teclado, rede, *GPU*).

4. Controle:

- Coordenação das operações através da **Unidade de Controle (UC)**.
- Geração de sinais de temporização e controle de fluxo de instruções.

3.2 Componentes principais

- **CPU:**
 - **Unidade de Controle:** Decodifica instruções e gerencia execução.
 - **ULA:** Executa cálculos aritméticos e operações lógicas.
 - **Registradores:** Armazenam dados temporários (ex.: *PC*, *AC*).
 - **Memória hierárquica:** *Cache L1/L2*, *RAM*, armazenamento secundário.
 - **Barramentos:** Endereço, dados e controle (ex.: *PCIe*, *USB*).

4. Evolução histórica e Marcos tecnológicos

4.1 Gerações de computadores

- **1ª Geração (1940-1956):** Válvulas termiônicas (ex.: *ENIAC*). **Problemas:** tamanho, consumo energético, falhas frequentes.
- **2ª Geração (1956-1963):** *Transistores* discretos. Redução de tamanho e maior confiabilidade (ex.: *IBM 1401*).
- **3ª Geração (1964-1971):** Circuitos integrados (*SSI/MSI*). Integração de múltiplos transistores em um *chip* (ex.: *IBM System/360*).
- **4ª Geração (1971-presente):** *LSI/VLSI/ULSI*. Microprocessadores, computadores pessoais (ex.: *Intel 4004*).

4.2 Lei de Moore e seu impacto

Formulada por *Gordon Moore* (1965), observa que o número de *transistores* em um *chip* dobra a cada 18-24 meses. **Consequências:**

- **Aumento exponencial de desempenho:** Maior paralelismo, *clocks* mais altos.
- **Redução de custos:** Miniaturização permite mais funcionalidades por área.
- **Novos desafios:** Dissipação térmica, vazamento de corrente, efeitos quânticos em escala nanométrica.

5. Desafios contemporâneos em arquitetura

5.1 O problema do Gargalo de memória

- *CPUs* modernas podem executar operações em nanossegundos, enquanto acessos à *RAM* levam centenas de ciclos.
- **Soluções:**
 - **Caches hierárquicos** (*L1*, *L2*, *L3*): Reduzem latência média.
 - **Prefetching:** Antecipa dados necessários.
 - **Memórias não voláteis rápidas** (*Optane*, *MRAM*).

5.2 Limites térmicos e energéticos

- Aumento da densidade de *transistores* eleva **densidade de potência** (ex.: $>100\text{W}/\text{cm}^2$ em *GPUs*).
- **Técnicas de mitigação:**
 - **Dynamic Voltage and Frequency Scaling (DVFS):** Ajusta *clock* conforme carga.
 - **Cooling avançado:** Líquido, termoelétrico.
 - **Dark silicon:** Parte do *chip* desativada para reduzir calor.

5.3 Paralelismo e *Multicore*

- Limites do paralelismo em nível de instrução (*ILP*) levaram à adoção de processadores *multicore*.
- **Vantagens:**
 - Desempenho escalável para cargas paralelizáveis.
 - Menor consumo por operação (comparado a *single-core* de alta frequência).
- **Desafios:**
 - **Escalabilidade de *software*:** Necessidade de programas *multithread*.
 - **Contenção de recursos:** Acesso concorrente à memória compartilhada.

6. Tendências emergentes

6.1 Sistemas Embarcados e *IoT*

- **Características:**
 - **Arquiteturas *RISC* (*ARM*, *RISC-V*):** Conjunto reduzido de instruções para eficiência energética.
 - **Integração *hardware/software*:** Sistemas dedicados (**ex.:** sensores, microcontroladores).
 - **Aplicações:** Automotivo, *wearables*, automação industrial.

6.2 Computação heterogênea

- Combinação de *CPUs*, *GPUs*, *TPUs* e aceleradores especializados.
- **Exemplo: *SoCs* (*System-on-Chip*)** como *Apple M1*, que integram *CPU*, *GPU* e *Neural Engine*.

6.3 Arquiteturas Pós-Moore

- **Alternativas à *CMOS*:** Transistores de *nanotubos*, spintrônica.
- **Computação quântica:** *Qubits* para resolver problemas intratáveis classicamente.

7. Conclusão e perspectivas

Esta introdução delineou os pilares da arquitetura de computadores, desde conceitos fundamentais até fronteiras tecnológicas. A evolução histórica mostra um campo em constante transformação, impulsionado pela Lei de *Moore* mas agora enfrentando limites físicos que demandam inovações radicais. O futuro do *design* de sistemas computacionais provavelmente envolverá:

- Maior especialização de *hardware* (*accelerators*).
- Novos paradigmas (neuromórfico, quântico).
- Ênfase em eficiência energética e segurança.

O próximo tópico do curso explorará **métricas quantitativas de desempenho** (*CPI, MIPS, benchmarks*), essenciais para avaliar e comparar arquiteturas.

Referências Bibliográficas

Projeto de Sistemas Computacionais – Aula 01 – Arquitetura e organização de computadores. Disponível em: <https://www.youtube.com/watch?v=zStcn5aMLug> .

Acesso em 15 de Agosto de 2025.