



# Infraestrutura de TI para Eng. Software

## Engenharia de Software

### Aula 2

---

Prof. Alvaro Leiroz

## QUAL A DIFERENÇA ENTRE BIT E BYTE

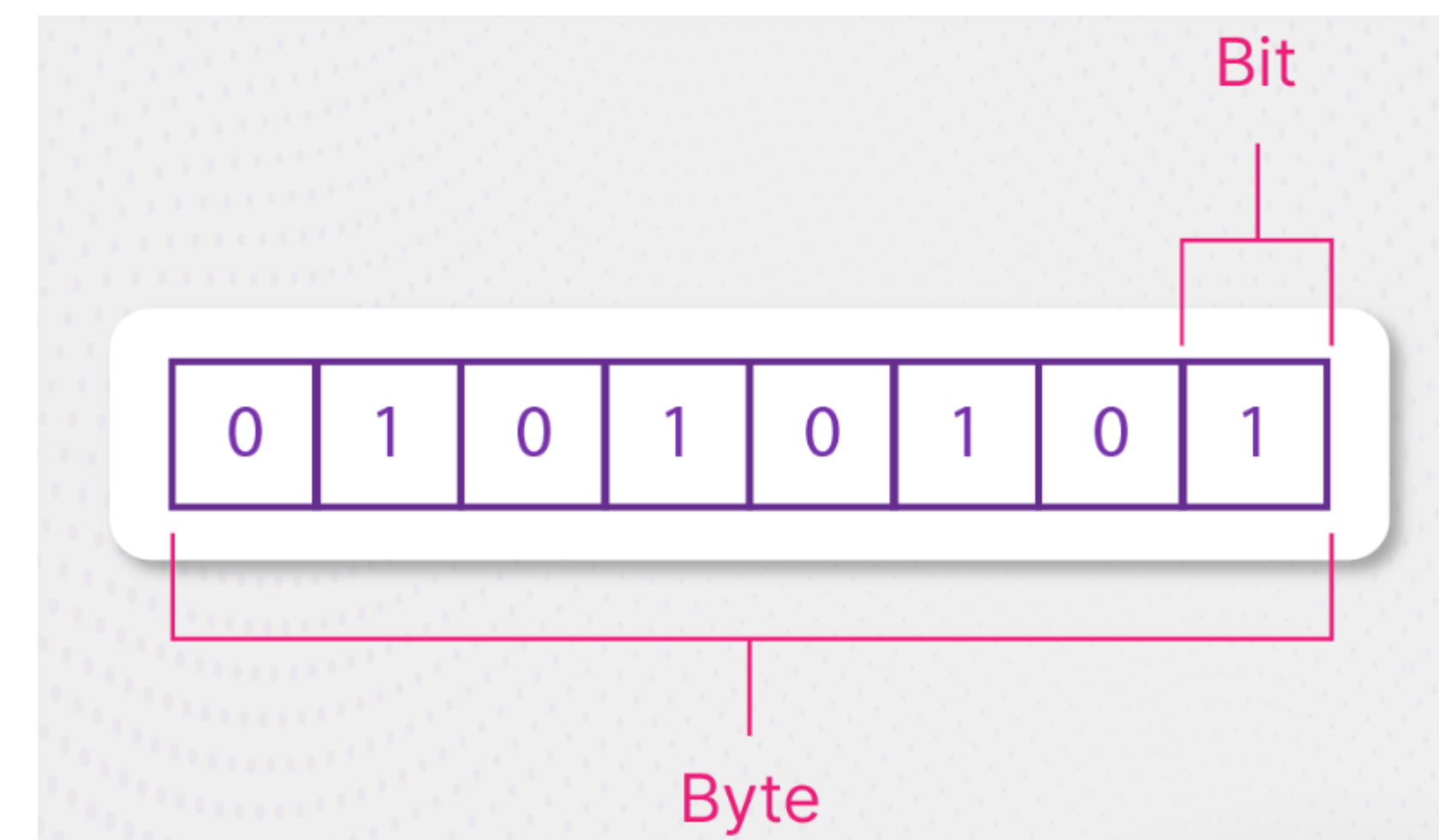
## QUAL A DIFERENÇA ENTRE BIT E BYTE

**Bit** é uma abreviação de binary digit (dígito binário). É a menor unidade de informação em sistemas digitais e pode ter apenas dois valores possíveis (0 ou 1). Sua sigla é o “b” minúsculo.

**Byte** é um conjunto de 8 bits de dados. É a unidade básica para representar caracteres, como letras e números. Como um único byte conta com  $2^8$  combinações de bits, pode representar até 256 símbolos diferentes. Sua sigla é o “B” maiúsculo.

**Bits (b)** são usados em transmissão de dados, algoritmos de criptografia e arquitetura de processadores.

Já **bytes (B)** são o padrão adotado em armazenamento de dados.



## **PORQUE ESTUDAR ARQUITETURA DE COMPUTADORES**

- ALGORITMO
- LINGUAGEM, COMPILADOR E ARQUITETURA
- PROCESSADOR E SISTEMA DE MEMÓRIA
- SISTEMA DE E/S (INCLUSIVE SO)

## ARQUITETURA DE COMPUTADORES

A **Arquitetura de Computadores** é o ramo da computação que define como o hardware e o software interagem. Em termos simples, é o “projeto lógico” de um computador: define quais instruções podem ser executadas, como os dados são armazenados e movimentados, e de que forma a CPU, memória e periféricos se comunicam.

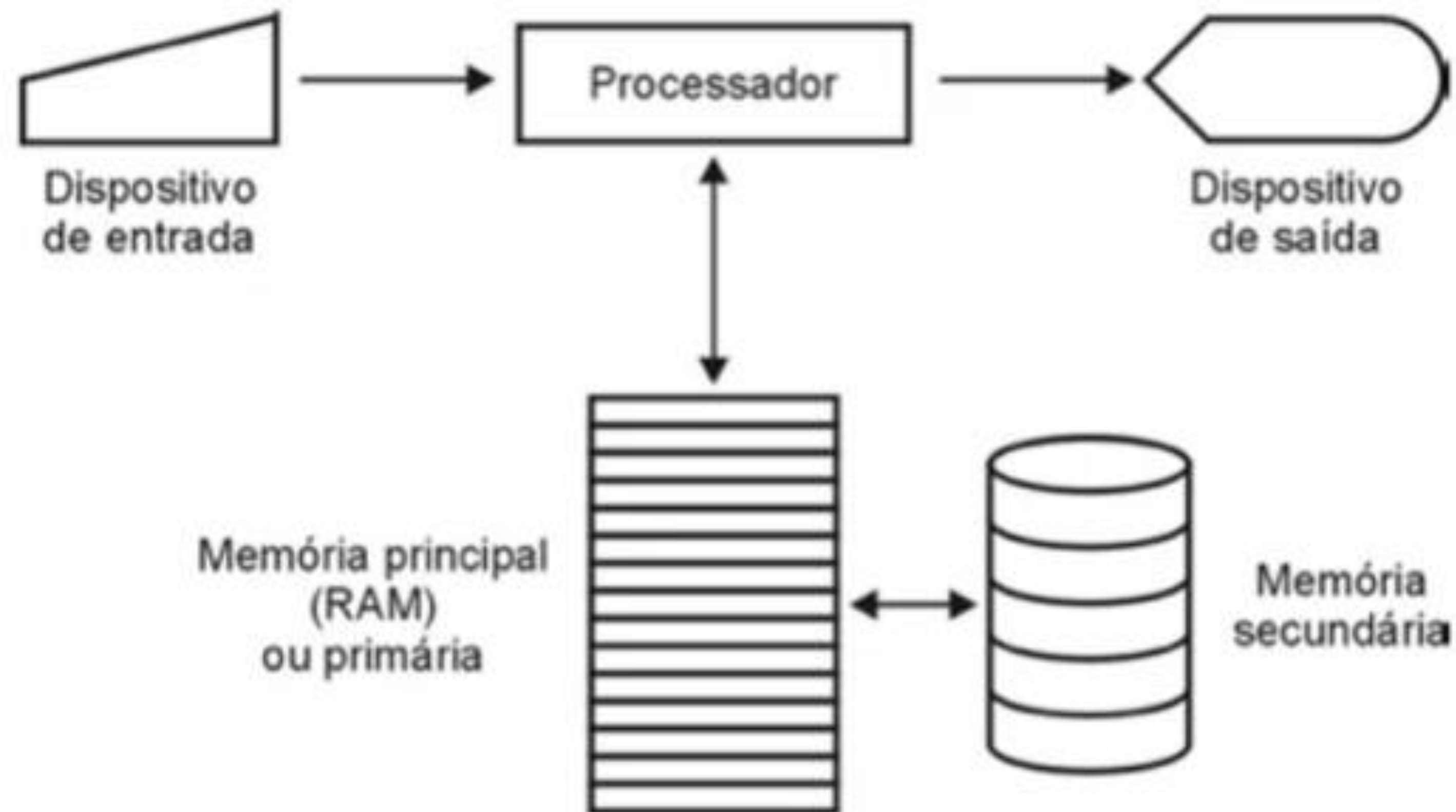
Ela estabelece a interface entre hardware e software (**Instruction Set Architecture – ISA**) e impacta diretamente no desempenho, consumo de energia, custo e aplicabilidade de um sistema computacional.

## ARQUITETURA DE PROCESSADORES

A Arquitetura de processadores é um **subconjunto da arquitetura de computadores**, voltado exclusivamente para o **processador (CPU)**.

**Abrangência:** especifica como o processador funciona, como interpreta instruções e como organiza seus componentes internos (UC, ULA, registradores, pipeline).

## COMPONENTES BÁSICOS



## O QUE SIGNIFICA “ARQUITETURA” X “ORGANIZAÇÃO”

- **Arquitetura** (o “o quê”): aspectos visíveis ao programador (conjunto de instruções, registradores, modos de endereçamento).
- **Organização** (o “como”): como a arquitetura é implementada (pipeline, caches, interconexões, microarquitetura).



## ESPECIFICAÇÕES

- **Conjunto de Instruções (ISA):** instruções que a CPU reconhece (ADD, LOAD, STORE, JMP).
- **Largura de palavra:** número de bits processados simultaneamente (ex.: 32 bits, 64 bits).
- **Modos de endereçamento:** como acessar dados (imediato, direto, indireto, relativo).
- **Número de registradores:** impacta em desempenho (arquiteturas RISC usam muitos registradores).
- **Organização da memória:** endereçamento por byte ou por palavra.
- **Pipeline e paralelismo:** execução de múltiplas instruções ao mesmo tempo.
- **Hierarquia de memória:** uso de caches, RAM e armazenamento secundário.

## TIPOS ARQUITETURAS

### **Computadores:**

- Von Neumann
- Harvard

### **Processadores:**

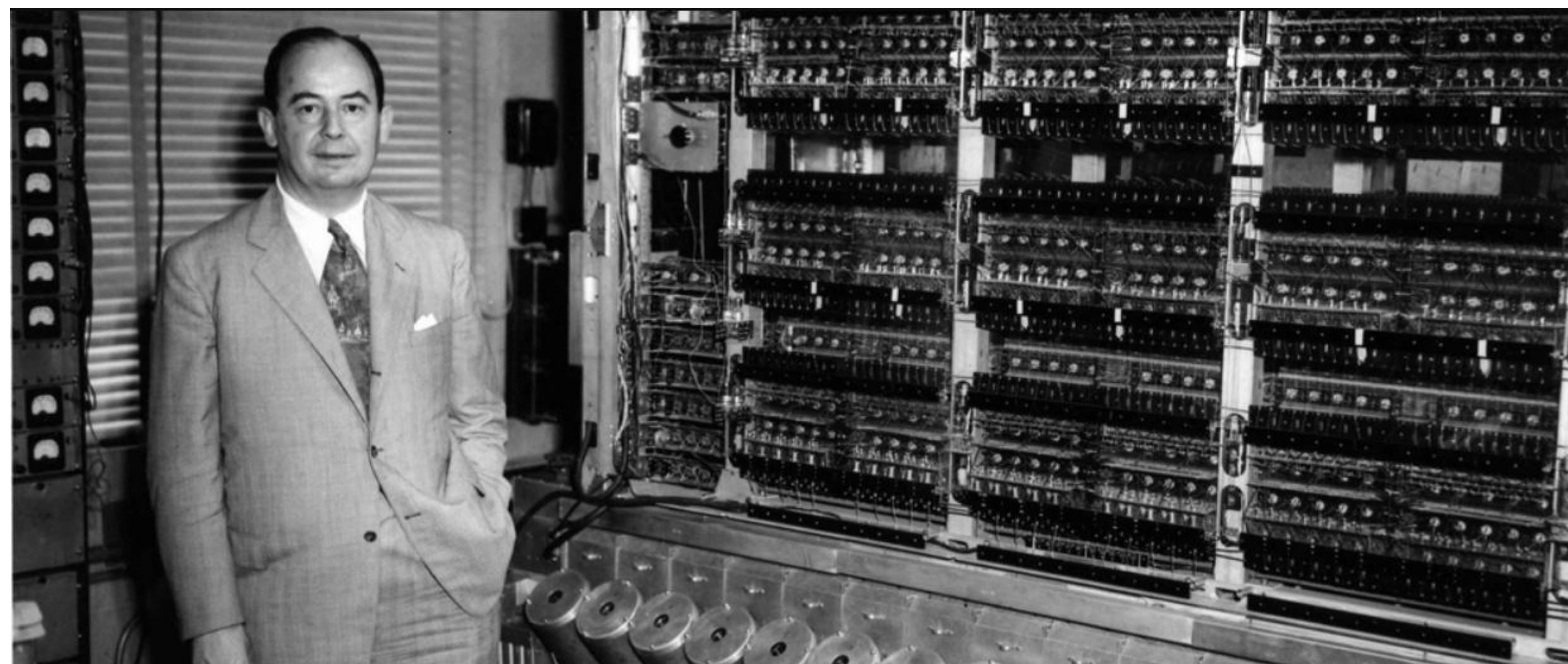
- RISC (Reduced Instruction Set Computer)
- CISC (Complex Instruction Set Computer)
- MIMD / SIMD / VLIW

### **Arquiteturas Especiais**

- **GPU (Graphics Processing Unit):** especializada em paralelismo massivo.
- **Arquiteturas Quânticas:** emergentes, baseadas em qubits.
- **Arquiteturas Neuromórficas:** inspiradas no cérebro humano.

## MODELO VON NEUMANN

É uma arquitetura que se caracteriza pela possibilidade de armazenar seus programas no mesmo espaço de memória que os dados, podendo assim manipular tais programas. Utilizando processamento (CPU) e armazenamento (“memória”) para comportar instruções e dados.

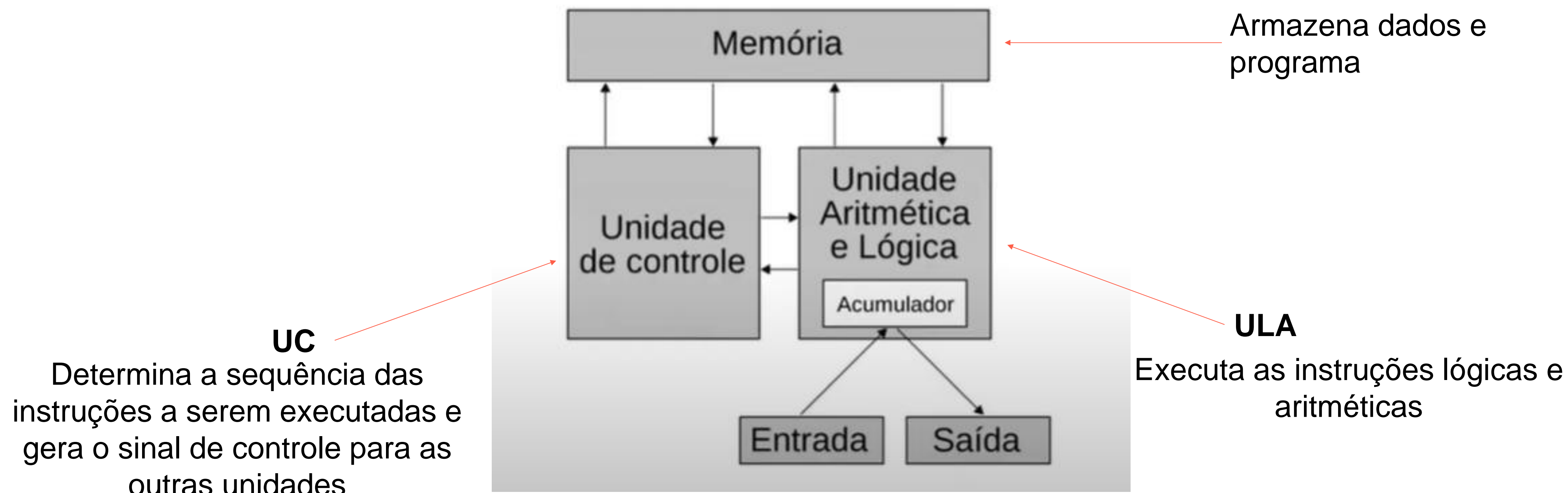


John Von Neumann



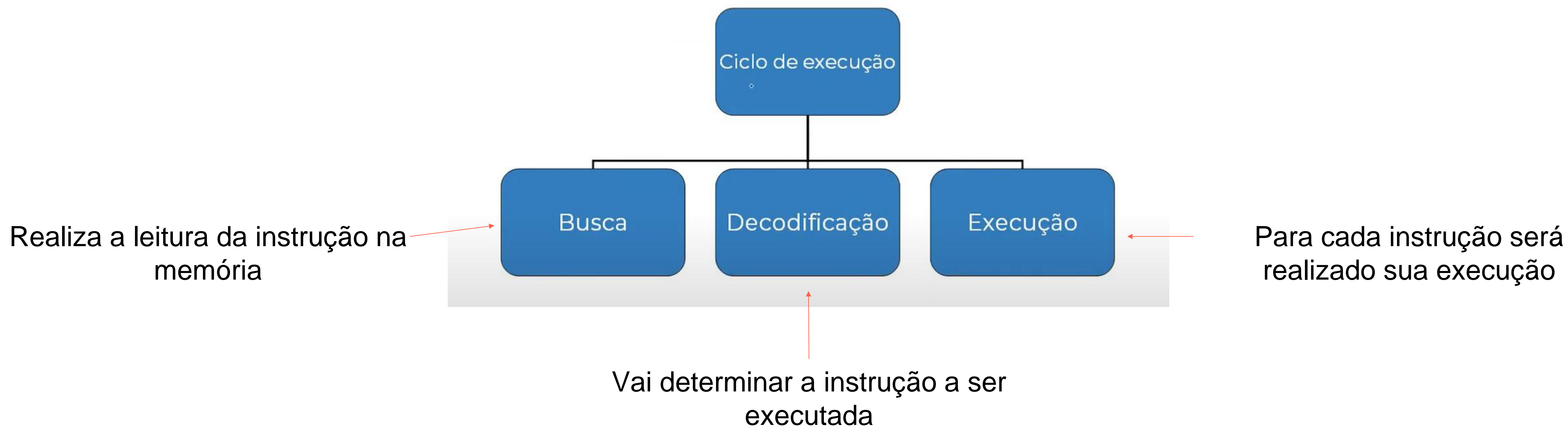
## MODELO VON NEUMANN

BASE DE PRATICAMENTE TODAS AS MÁQUINAS ATUAIS



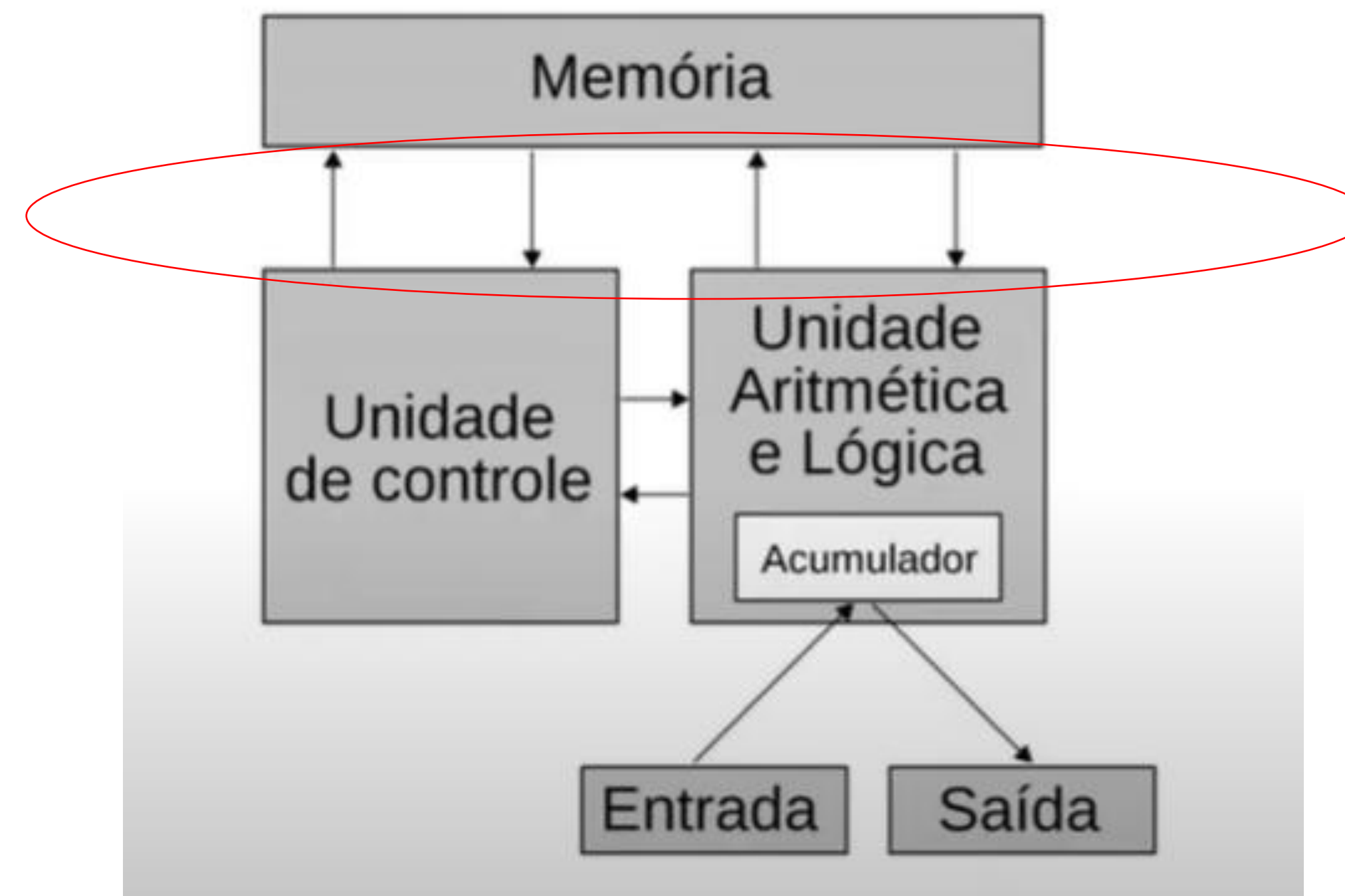
## MODELO VON NEUMANN

### CICLO DE EXECUÇÃO



## MODELO VON NEUMANN

Gargalo de Von Neumann é a limitação da taxa de transferência entre CPU e a memória em comparação com a quantidade de memória. Esta transferência é menor do que a taxa com que o processador consegue trabalhar e menor do que a quantidade de memória em geral disponível. Faz com que a CPU seja forçada a esperar por dados que precisam ser transferidos para ou a partir da memória. Gera desperdício de tempo



## MODELO VON NEUMANN

Onde estão os componentes de Von Neumann na arquitetura moderna

CPU	Barramento	Memória	Dispositivos de E/S
Processador	Placa mãe	RAM Cache Registradores Disco	Mouse Teclado Impressora Monitor



## MODELO HARVARD

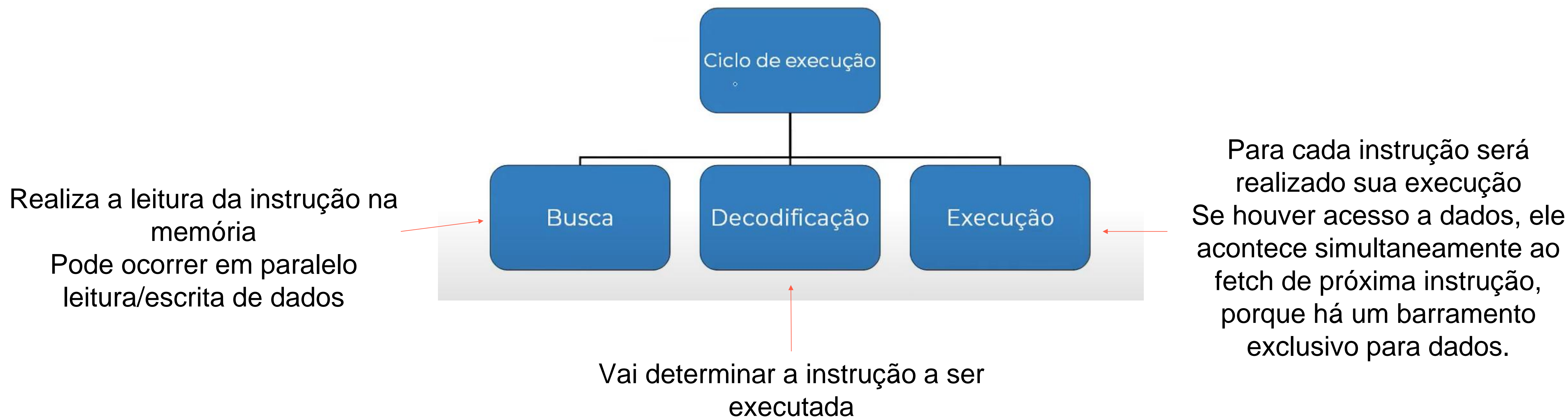
**Semelhante a arquitetura Von Neumann**, aqui também temos um processador que executa instruções subsequentes e dispositivos de entrada/saída. A principal diferença é a memória. Na arquitetura de Harvard, está dividida em dois elementos separados, o primeiro armazena dados e o segundo armazena o programa.





## MODELO HARVARD

### Ciclo de execução



## MODELO HARVARD

Onde estão os componentes de Harvard na arquitetura moderna

CPU	Barramento	Memória	Dispositivos de E/S
Processador	Placa mãe	RAM Cache Registradores Disco	Mouse Teclado Impressora Monitor

## RESUMINDO

### Von Neumann

**Dados e programas na mesma memória**

**Barramento único** entre CPU e memória

**Não permite pipeline** (dados e instruções disputam o mesmo caminho)

**Um ciclo completo leva dois ciclos de clock**

**Mais simples** (menor custo e implementação mais direta)

**Mais indicada/Usada em:** PCs, notebooks, servidores, sistemas de propósito geral

### Harvard

**Dados e programas em memórias separadas**

**Um barramento distinto** entre CPU e cada memória

**Permite pipeline** (acesso simultâneo a dados e instruções)

**Um ciclo completo em um único ciclo de clock**

**Mais complexa** (maior custo e design mais elaborado)

**Mais indicada/Usada em:** Microcontroladores, DSPs (processadores de sinais digitais), sistemas embarcados, aplicações em tempo real

## PROCESSADORES

- É a **Unidade Central de Processamento** – considerada o “cérebro do computador”.
- Responsável por **interpretar instruções** e **executar operações** (cálculos, lógica, controle).
- Opera em **ciclos de clock**, sincronizados por um sinal oscilador.



## PROCESSADORES

- **Unidade de Controle (UC)**

- Decodifica instruções e gera sinais para os outros blocos.
- Controla o fluxo de dados entre CPU, memória e E/S.

- **Unidade Lógica e Aritmética (ULA)**

- Realiza operações matemáticas (soma, subtração, multiplicação) e lógicas (AND, OR, NOT).

- **Registradores**

- Memórias ultrarrápidas internas à CPU.
- Exemplos:
  - **PC (Program Counter)**: aponta a próxima instrução.
  - **IR (Instruction Register)**: guarda a instrução atual.
  - **ACC/AX/EAX/RAX**: acumuladores.
  - **SP (Stack Pointer)**: aponta para o topo da pilha.
  - **Flags**: condições (zero, negativo, carry, overflow).

- **Clock**

- Frequência que dita quantos ciclos por segundo a CPU executa (GHz).
- Mais clock  $\neq$  mais desempenho sempre (depende de arquitetura e IPC).

## PROCESSADORES

### Termos Importantes

- **CPI (Cycles per Instruction):** ciclos necessários para executar uma instrução.
- **IPC (Instructions per Cycle):** instruções executadas por ciclo.
- **Pipeline:** divide a execução em estágios (Fetch, Decode, Execute, Memory, Writeback).
- **Superscalar:** permite múltiplas instruções por ciclo.
- **Out-of-order Execution:** CPU reorganiza a execução para não “ficar parada”.
- **RISC x CISC:**
  - RISC → poucas instruções simples (ARM, RISC-V).
  - CISC → muitas instruções complexas (x86).

## PROCESSADORES

### Termos Importantes

- **CPI (Cycles per Instruction):** ciclos necessários para executar uma instrução.
- **IPC (Instructions per Cycle):** instruções executadas por ciclo.
- **Pipeline:** divide a execução em estágios (Fetch, Decode, Execute, Memory, Writeback).
- **Superscalar:** permite múltiplas instruções por ciclo.
- **Out-of-order Execution:** CPU reorganiza a execução para não “ficar parada”.
- **RISC x CISC:**
  - RISC → poucas instruções simples (ARM, RISC-V).
  - CISC → muitas instruções complexas (x86).

## ESTRUTURA

### O que é x86

- É o nome da arquitetura de conjunto de instruções (ISA) criada pela Intel.
- O nome vem do Intel 8086 (de 1978) e seus sucessores (80186, 80286, 80386, 80486).
- Por isso, x86 se tornou sinônimo de processadores compatíveis com essa família de instruções.
- Originalmente, era uma arquitetura de 16 bits (8086/8088).
- Evoluiu para 32 bits (80386) e depois para 64 bits (x86-64 ou AMD64).



## ESTRUTURA

### O que é x32 (32 bits)

- Refere-se a processadores e sistemas operacionais que usam endereços de 32 bits.
- Significa que podem endereçar até 4 GB de memória RAM diretamente ( $2^{32}$  endereços = 4.294.967.296 bytes).
- Baseada nos processadores Intel 80386 em diante.
- Hoje, já é considerada limitada, pois não aproveita memórias maiores.

## ESTRUTURA

### O que é x64 (64 bits)

- Refere-se a processadores e sistemas operacionais que usam endereços de 64 bits.
- Teoricamente, podem endereçar 16 exabytes de memória ( $2^{64}$ ), mas na prática os processadores limitam a valores menores (ex.: 256 TB).
- Introduzida pela AMD como AMD64 e adotada pela Intel como Intel 64.
- É a arquitetura padrão atual em PCs, notebooks e servidores.
- Permite:
  - Mais registradores.
  - Endereçamento de memórias maiores.
  - Melhor desempenho em algumas aplicações.

## PROCESSADORES

<https://www.youtube.com/watch?v=o9lcBPZMiJQ>

## INTERESSANTE PARA DESENVOLVEDORES



Classe: *register-memory* (80x86) / *load-store* (MIPS)

Endereçamento de Memória: *bytes* (alinhados ou não)

Modos de Endereçamento: como acessar conteúdo da memória

Tipos e Tamanhos dos Operandos

Operações: transferência de dados, aritméticas, flops, ...

Controle de Fluxo das Instruções

Codificação do *ISA*



## EXERCÍCIOS

**1. Qual a principal diferença entre a arquitetura de Von Neumann e a arquitetura de Harvard?**

- A) Von Neumann usa dois barramentos; Harvard, um barramento único.
- B) Von Neumann utiliza memórias separadas; Harvard, memória única.
- C) Von Neumann utiliza memória única para dados e instruções; Harvard, memórias separadas.
- D) Ambas utilizam memória única, mas Von Neumann é mais rápida.

**2. Qual unidade da CPU é responsável por interpretar instruções e gerar sinais de controle?**

- A) ULA
- B) Registradores
- C) Unidade de Controle (UC)
- D) Memória cache

**3. A limitação conhecida como “gargalo de Von Neumann” ocorre porque:**

- A) A CPU é mais lenta que a memória.
- B) O mesmo barramento é usado para buscar instruções e dados.
- C) A memória principal não consegue armazenar programas.
- D) O pipeline trava quando há muitas instruções.

## EXERCÍCIOS

### 4. Qual das opções caracteriza a arquitetura RISC?

- A) Muitas instruções complexas, cada uma realizando várias operações.
- B) Poucas instruções simples, otimizadas para execução rápida.
- C) Executa apenas operações gráficas.
- D) Utiliza apenas memória Harvard.

### 5. Em relação a x32 e x64, podemos afirmar que:

- A) x32 suporta até 8 GB de RAM e x64 até 128 GB.
- B) x32 é limitado a 4 GB de RAM e x64 pode acessar muito mais memória.
- C) x64 foi criado pela Intel em 1985.
- D) x32 é o padrão atual em servidores modernos.

## EXERCÍCIOS

### 6. Cenário – Desempenho em Aplicações de Tempo Real

Uma empresa está desenvolvendo um sistema embarcado para carros autônomos, que precisa processar dados de sensores em tempo real. O time de engenheiros está em dúvida se deve adotar Von Neumann ou Harvard. **Explique qual arquitetura seria mais adequada e justifique sua resposta.**

### 7. Cenário – Processador Multicore em Servidores

Um banco deseja modernizar sua infraestrutura e está avaliando processadores multicore baseados em RISC ou CISC. **Como engenheiro de software, explique as diferenças e aponte qual abordagem pode oferecer melhor escalabilidade para sistemas de alta carga.**

### 8. Cenário – Limitação de Memória em Sistemas Antigos

Uma empresa ainda utiliza máquinas com arquitetura x32 (32 bits) para rodar seu sistema de gestão. O time de TI percebe que, mesmo instalando 8 GB de RAM, o sistema só reconhece cerca de 3,5 GB. **Explique por que isso acontece e qual seria a solução para aproveitar mais memória.**

## REFERÊNCIAS

- **TANENBAUM, A. S.; AUSTIN, T.** *Organização Estruturada de Computadores*. (ed. mais recente). Aborda ISA x microarquitetura, Von Neumann/Harvard, cache e E/S de forma didática.
- **PATTERSON, D. A.; HENNESSY, J. L.** *Computer Organization and Design* (e o *Computer Architecture: A Quantitative Approach*). Base para pipeline, CPI/IPC, paralelismo.
- **STALLINGS, W.** *Computer Organization and Architecture*. Excelente para caminho de dados, controle (hardwired x microprogramado), memória e E/S.
- **MONTEIRO, M. A.** *Introdução à Organização de Computadores* (5ª ed.). Ótimo apoio em português para CPU, memória, barramentos e E/S.
- **DELGADO, J.; RIBEIRO, C.** *Arquitetura de Computadores* (5ª ed.). Complementa com foco em desempenho, paralelismo e arquiteturas modernas.
- **PAIXÃO, R.** *Arquitetura de Computadores – PCs*. Enfoque prático de PCs (componentes reais, barramentos, controladoras).