

INSTITUTO FEDERAL
Sul-rio-grandense

Câmpus
Venâncio Aires

Arquitetura e Organização de Computadores

Professor: Fernando Luís Herrmann

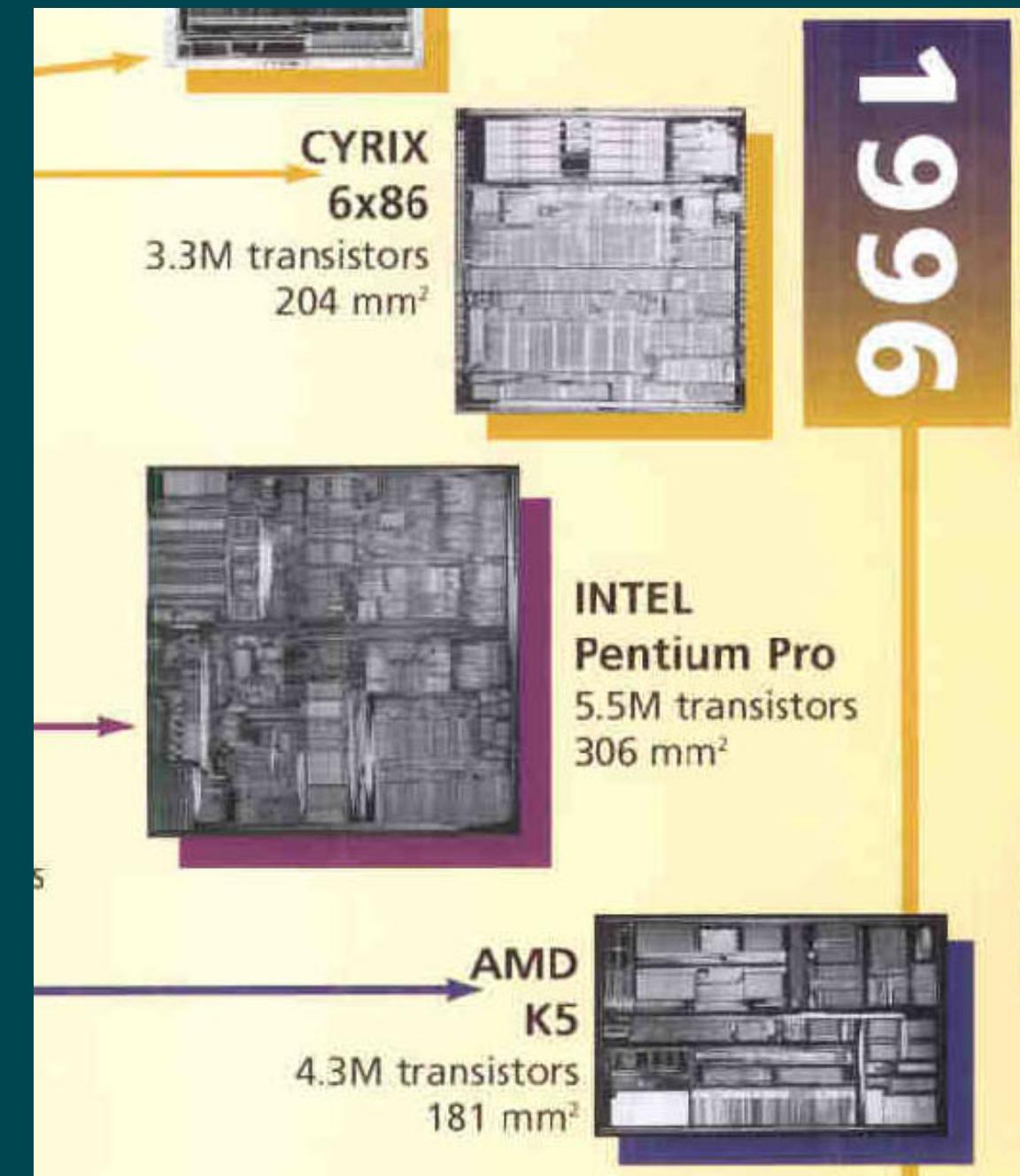
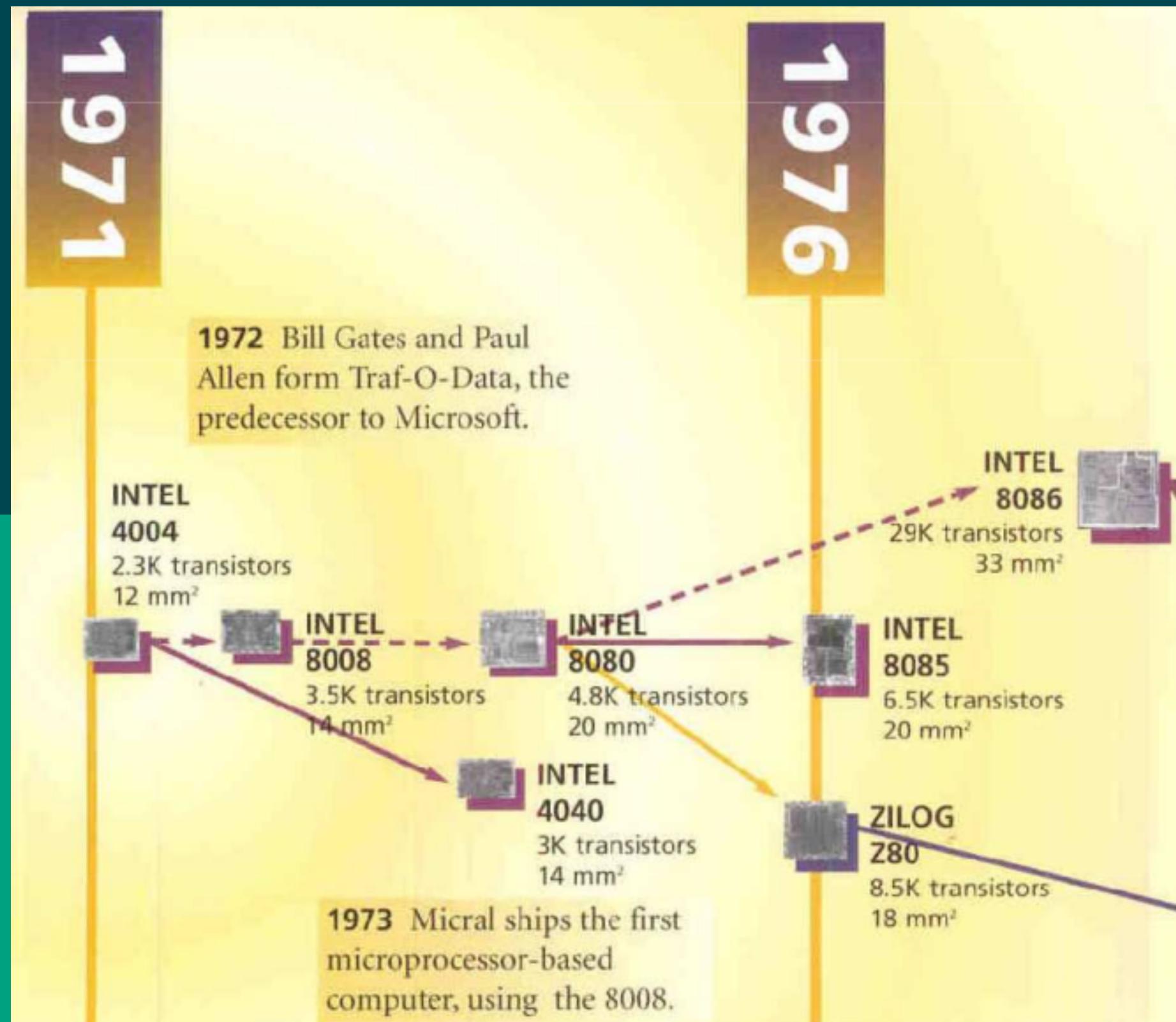
E-mail: fernando.herrmann@ifsul.edu.br

Introdução à Arquitetura de Computadores

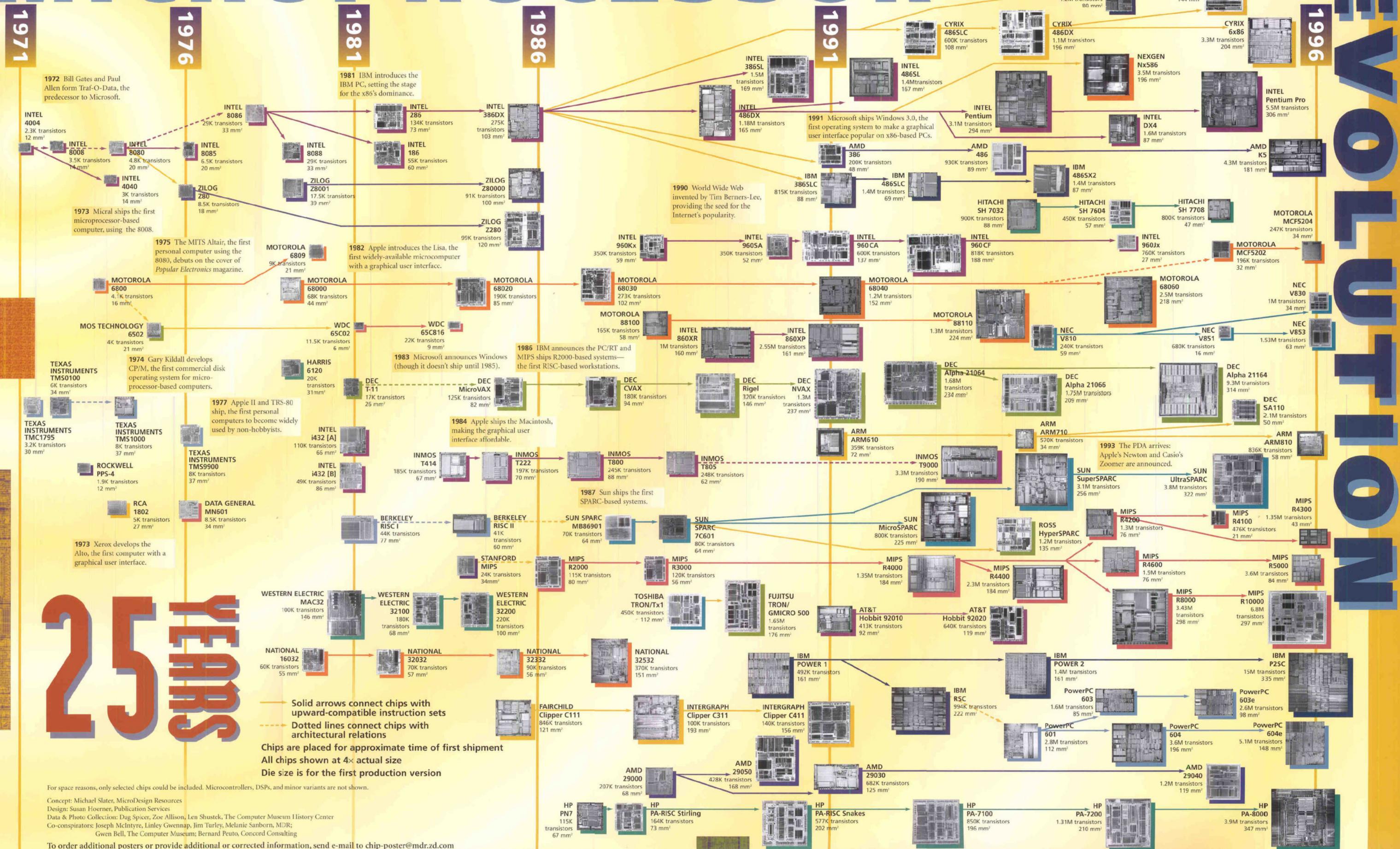
Tópicos

- Origem, Evolução
- Breve Histórico
- Arquitetura Von Neumann
- Arquitetura Harvard
- CPU simples
- Processador Neander

Origem, evolução, ...



MICROPROCESSOR



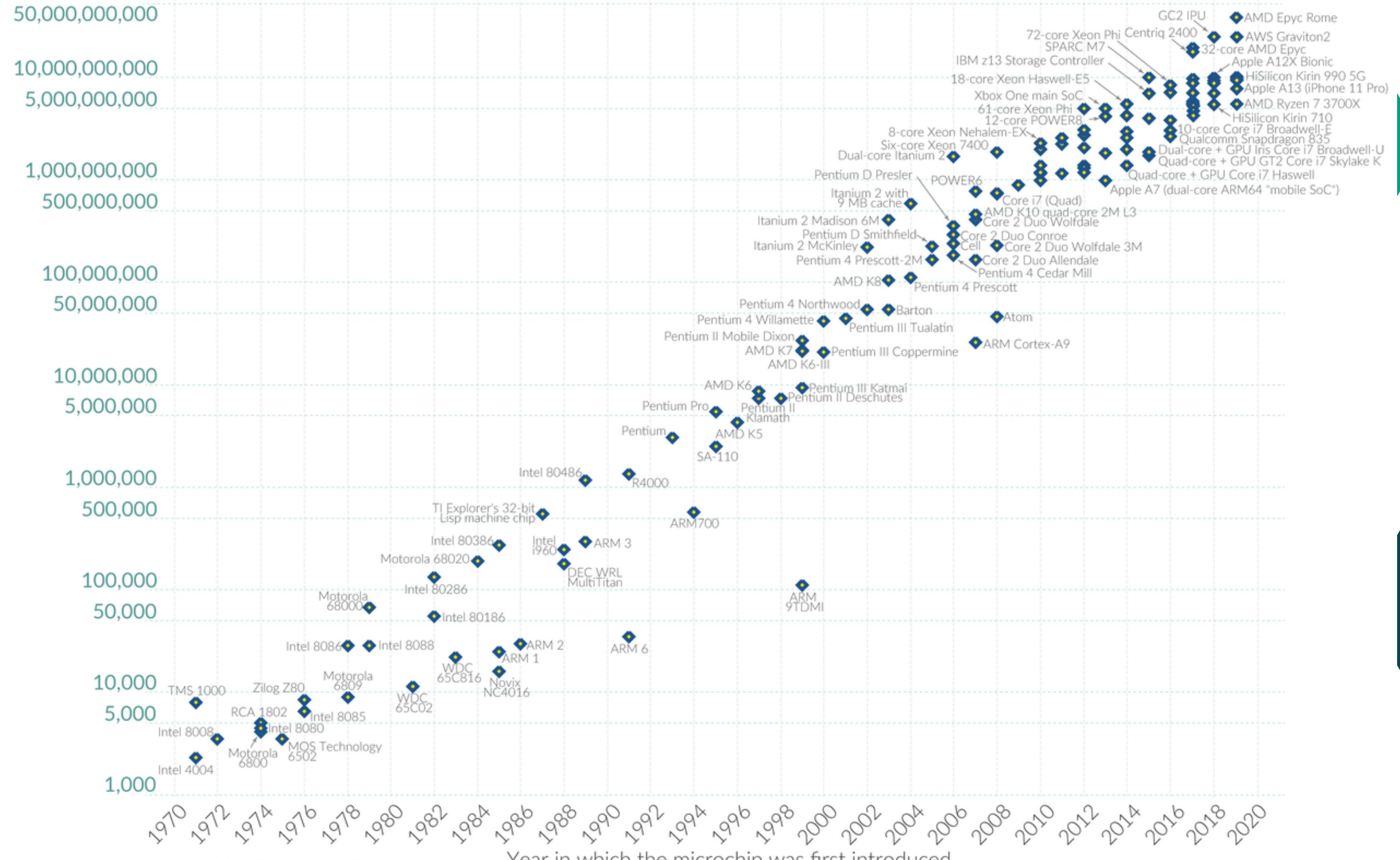
Moore's Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years.

This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Our World
in Data

Transistor count

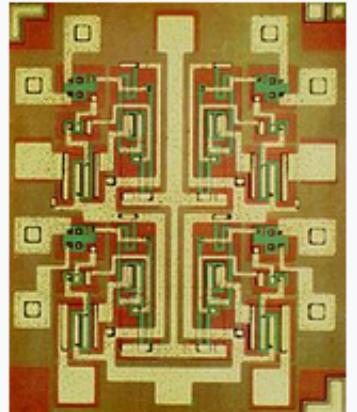


Data source: Wikipedia ([wikipedia.org/wiki/Transistor_count](https://en.wikipedia.org/wiki/Transistor_count))

OurWorldInData.org - Research and data to make progress against the world's largest problems.

Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

Semiconductor
device
fabrication



MOSFET scaling
(process nodes)

10 µm - 1971

6 µm - 1974

3 µm - 1977

1.5 µm - 1981

1 µm - 1984

800 nm - 1987

600 nm - 1990

350 nm - 1993

250 nm - 1996

180 nm - 1999

130 nm - 2001

90 nm - 2003

65 nm - 2005

45 nm - 2007

32 nm - 2009

22 nm - 2012

14 nm - 2014

10 nm - 2016

7 nm - 2018

5 nm - 2020

3 nm - 2022

Future

2 nm ~ 2024

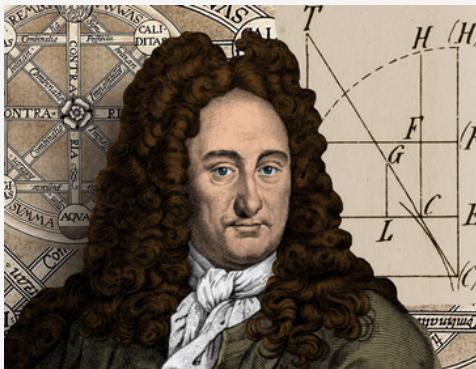
Breve histórico



1642

Blaise Pascal

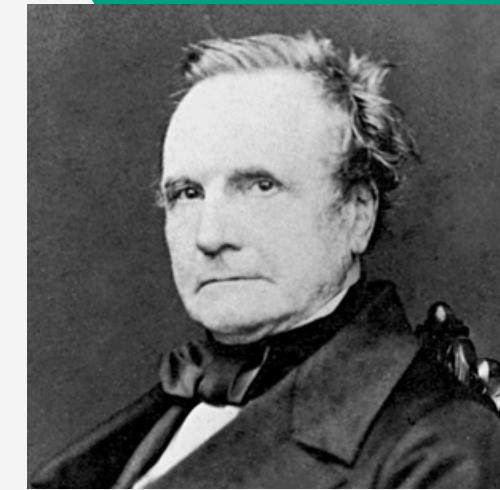
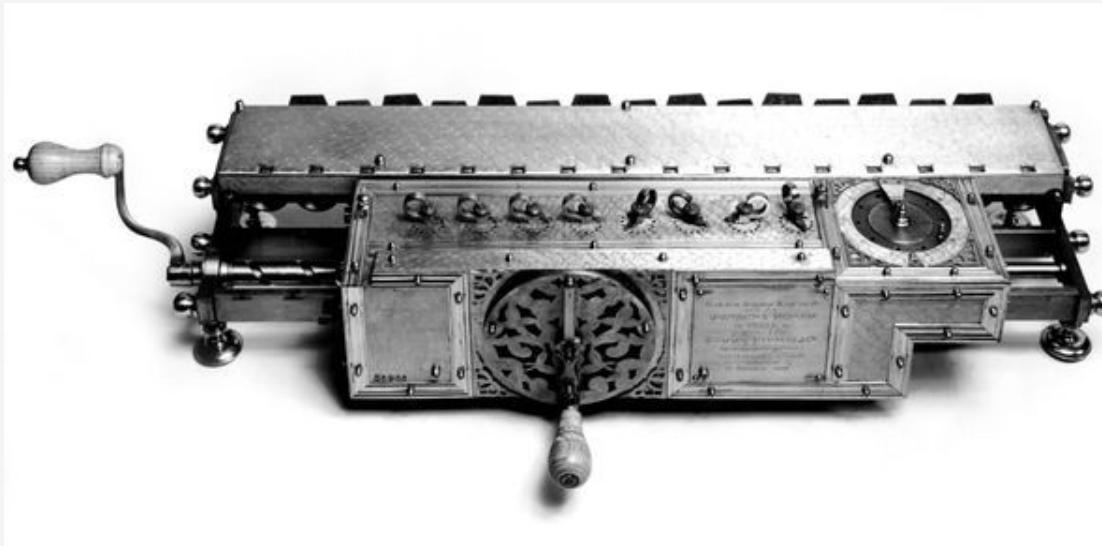
Máquina de calcular mecânica (engrenagens e alavancas). Permitia fazer adições e subtrações - **Inovação:** vai-um, repr. numérica



1671

Leibniz

Máquina de calcular mecânica. Permitia fazer as 4 operações - **Inovação:** multiplicação e divisão

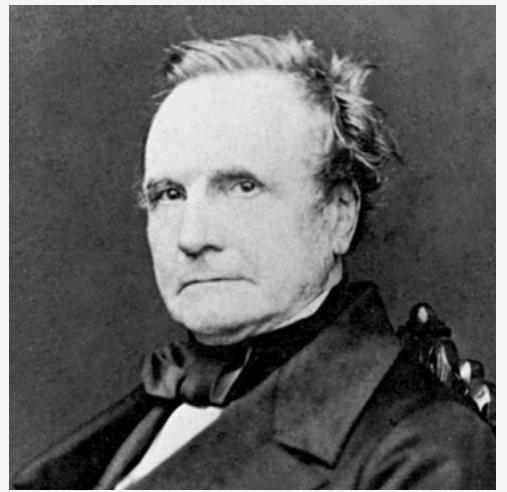


1827

Charles Babbage

Máquina Diferencial: implementava o método de diferenças finitas. **Inovação:** Cálculos com polinômios

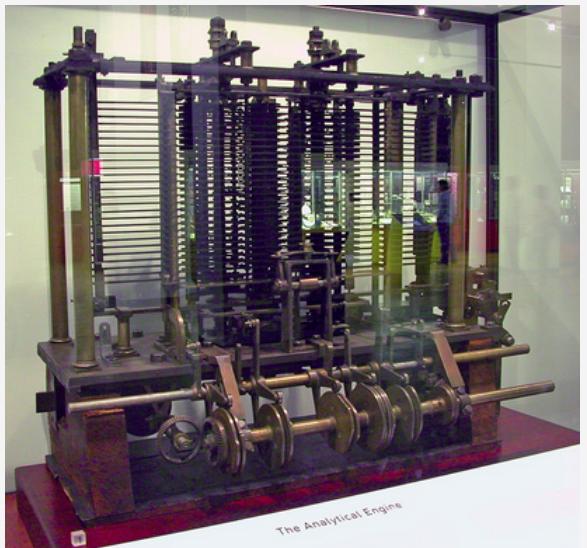




1834

Charles Babbage

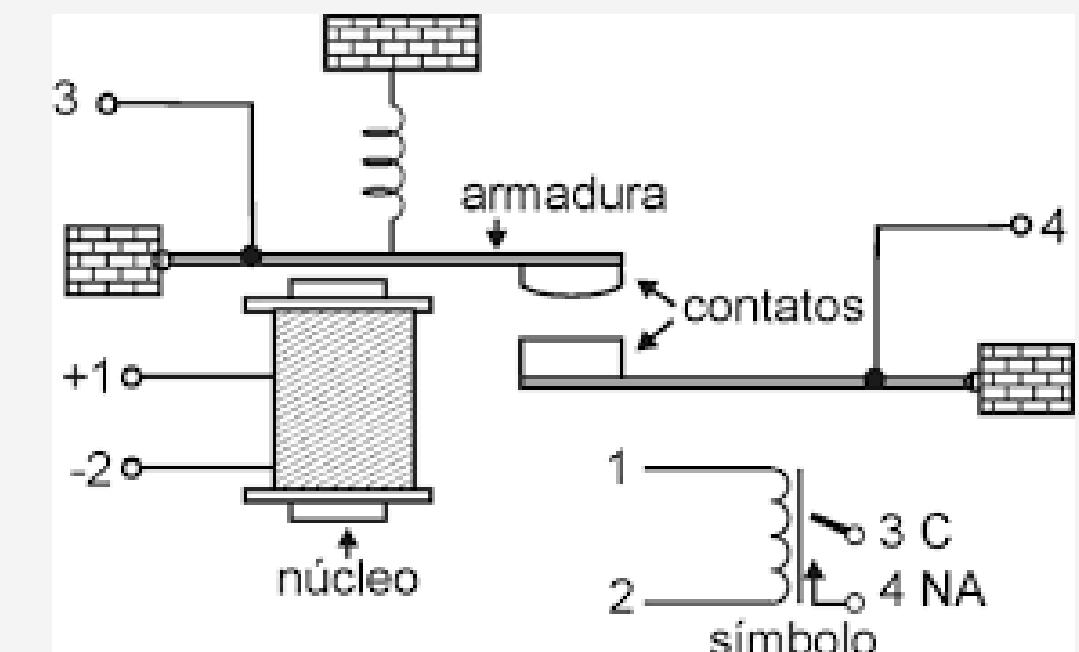
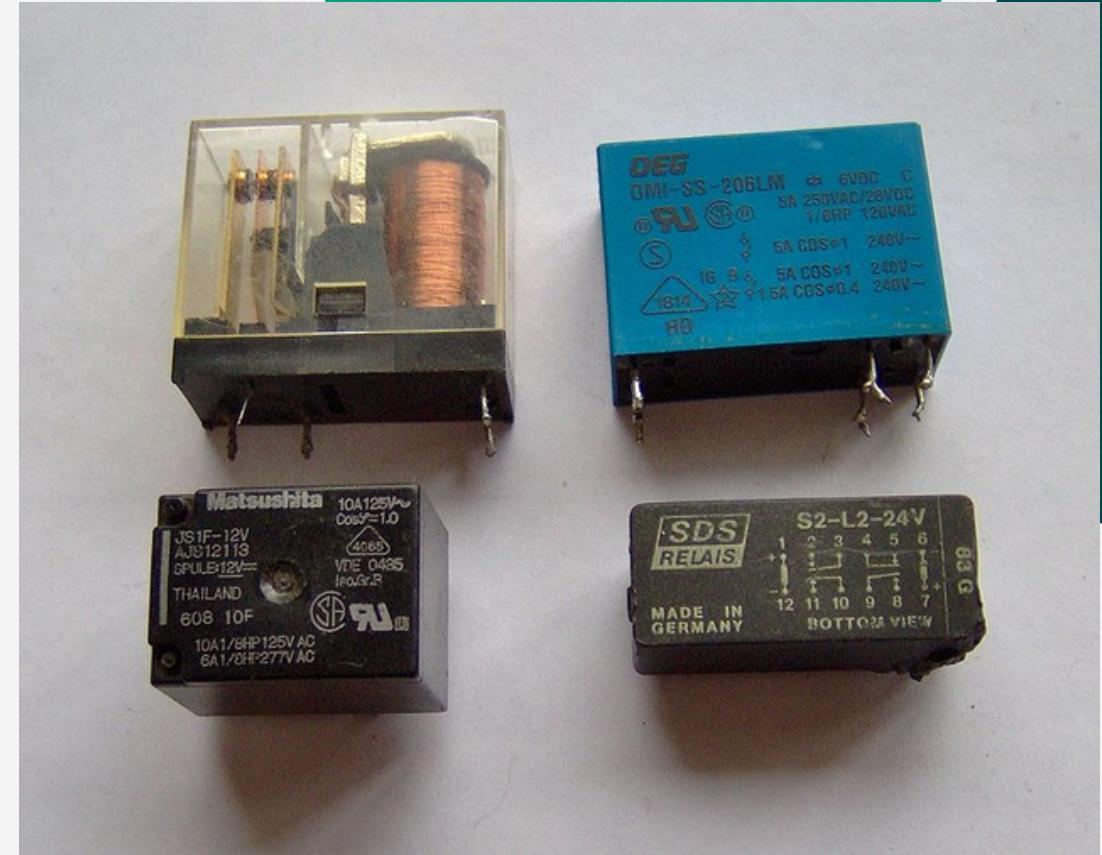
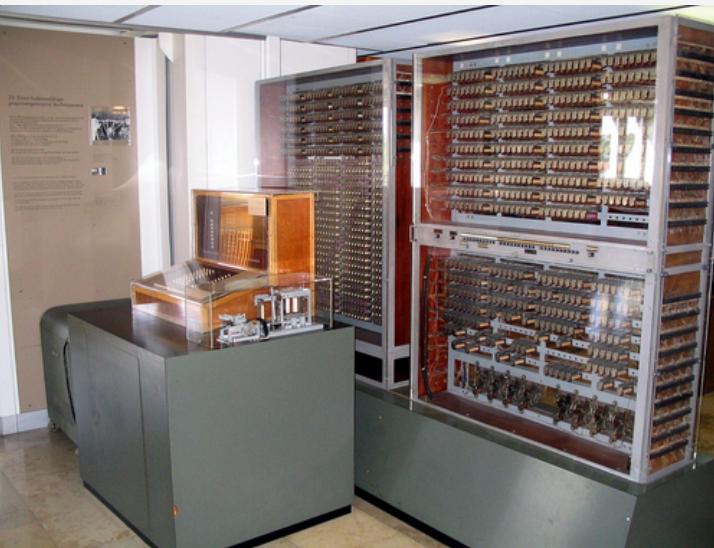
Máquina Analítica: proposta de uma máquina de propósito geral. Babbage não concluiu a construção da máquina devido a financiamento inadequado. **Inovação:** Mecanismo automático de controle de sequência de programa. ULA.

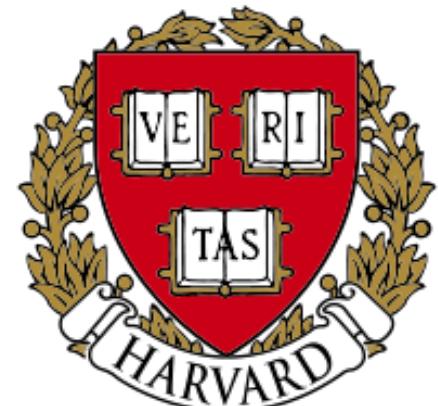


1941

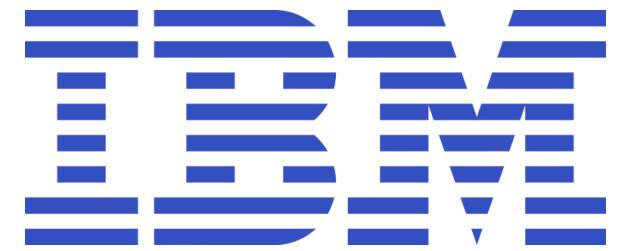
Konrad Zuse

Z3 - Computadores com circuitos eletromecânicos. Utilizava Relés eletromecânicos





1944



Universidade de Harvard e IBM

Harvard Mark I - Calculadora Automática de Sequência Controlada", primeira calculadora eletro-mecânica automática produzida em larga escala. Construído em 1944, em uma parceria com a empresa IBM, durante a Segunda Guerra Mundial.

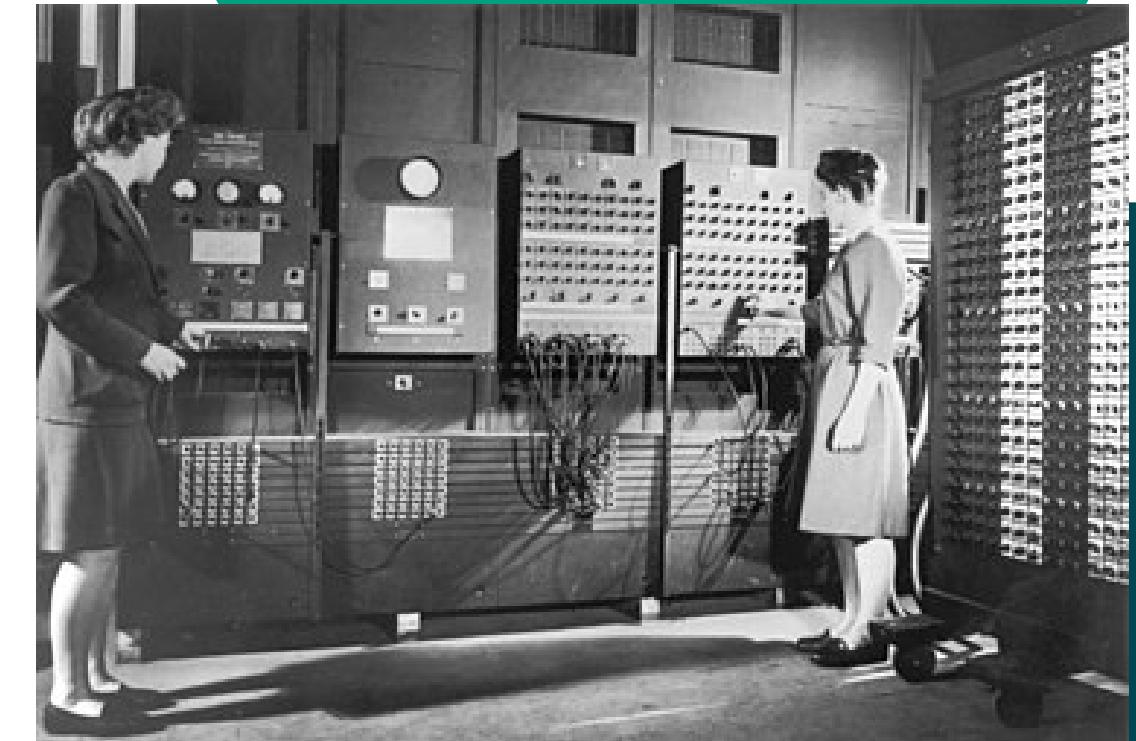


1945

ENIAC

Electronic Numerical Integrator and Computer.

Primeiro computador majoritariamente eletrônico. Criado especificamente para realizar cálculos balísticos ainda na Segunda Guerra Mundial, o ENIAC podia ser usado para outras funções, já que podia receber instruções por algoritmos. O ENIAC era composto por milhares de capacitores, relés, interruptores e tubos à vácuo. ENIAC tinha 18.000 válvulas, consumia 200 KW, ocupava 180m² e pesava 30 toneladas. ENIAC foi desativado em 1955.



1949

EDVAC

Electronic Discrete Variable Automatic Computer.

Foi um dos primeiros computadores eletrônicos, ao contrário do ENIAC que operava com base em codificação decimal, o EDVAC foi projetado para utilizar códigos binários e manter os programas armazenados na memória, respeitando a arquitetura de von Neumann. EDVAC tinha 5937 válvulas, consumia 56 KW, ocupava 45,5m² e pesava 7,8 toneladas.

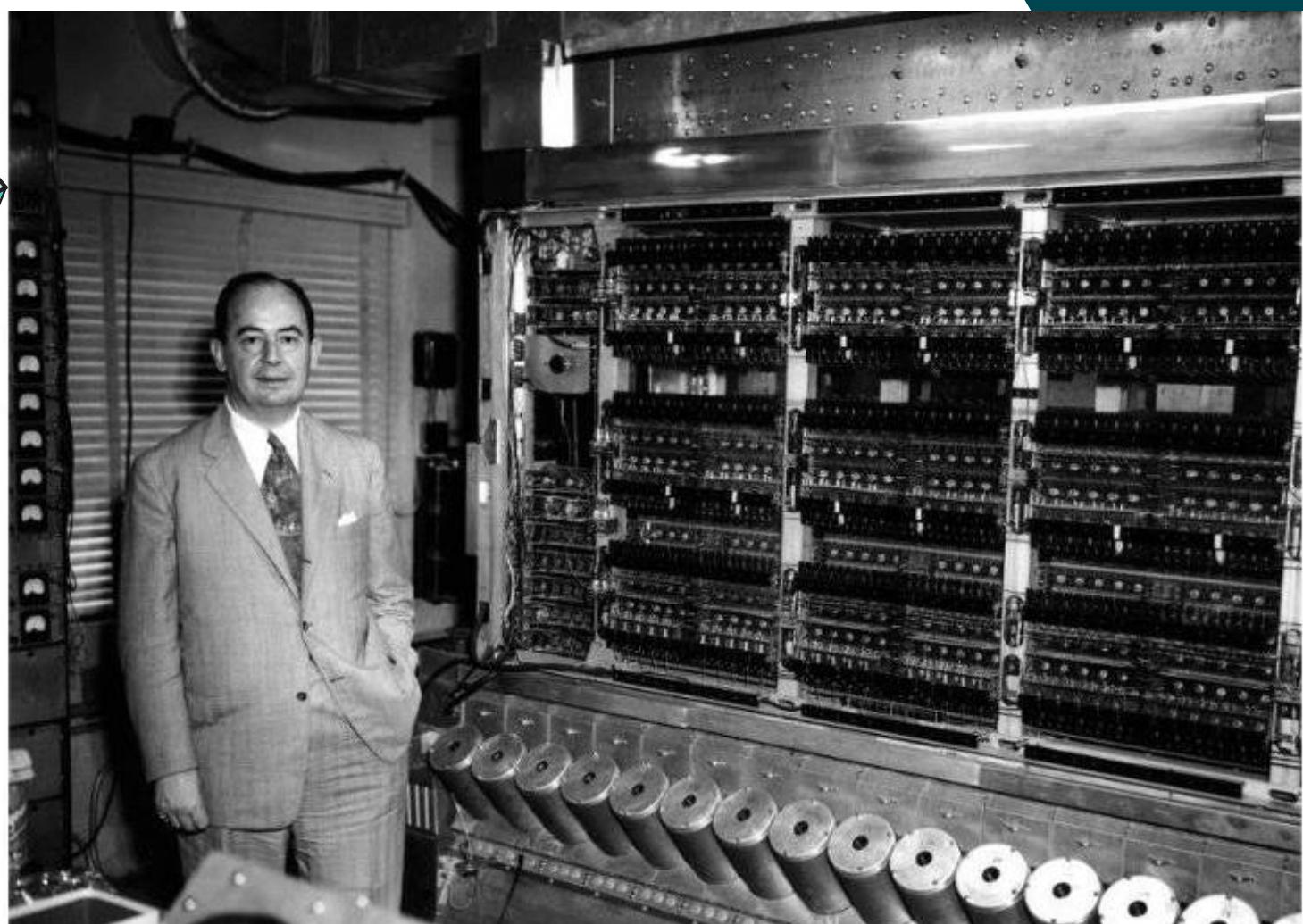
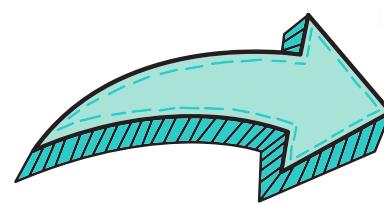
EDVAC recebeu várias atualizações, incluindo um dispositivo de entrada / saída de cartões perfurados em 1953, a memória adicional em um cilindro magnético em 1954 e uma unidade de aritmética de ponto flutuante em 1958. Permaneceu funcional até 1961.



1952

IAS

Jon Von Neumann construiu em 1952 o computador IAS (Institute for Advanced Study - Princeton - USA)



Arquitetura Von Neumann

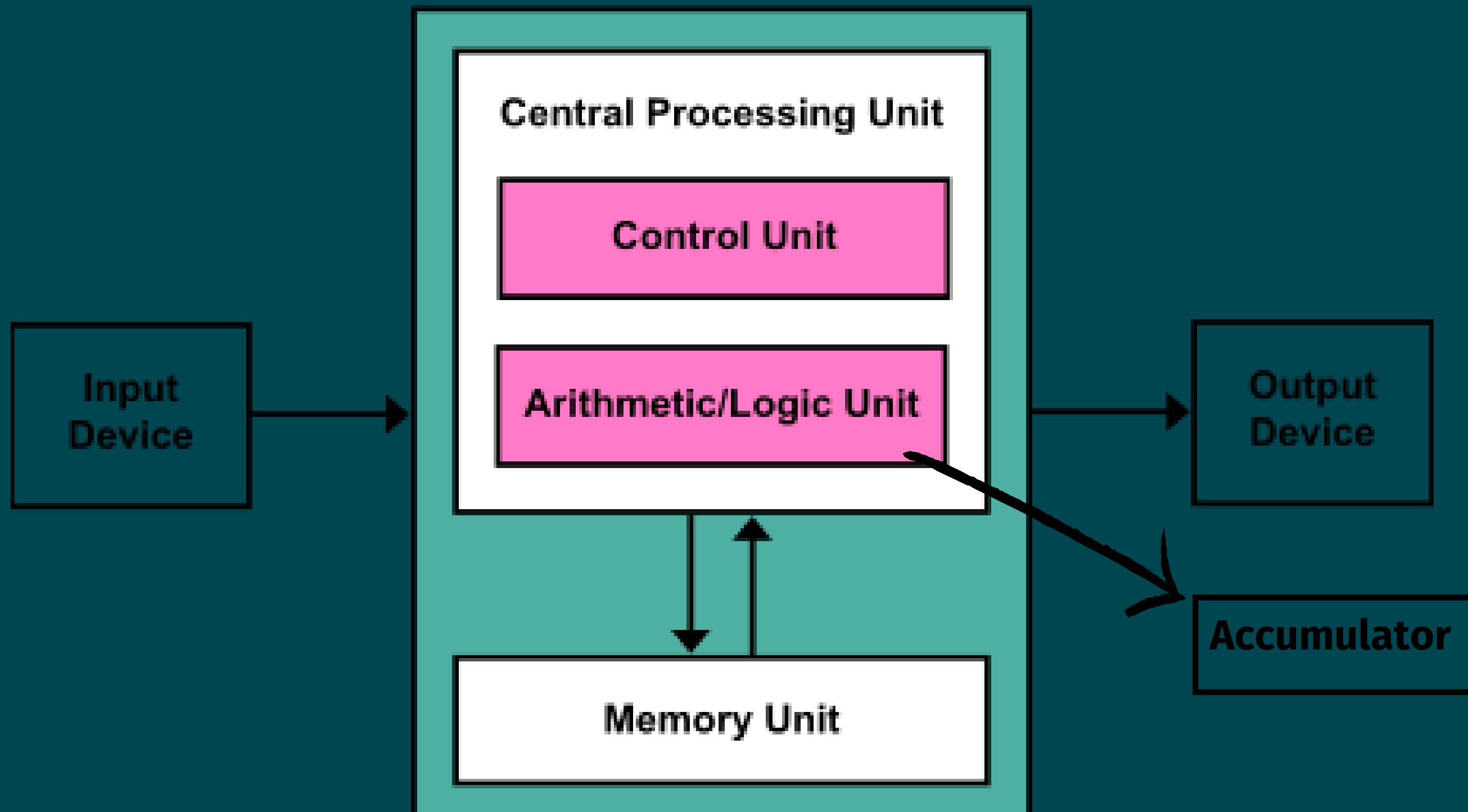
Componentes

- Memória
- Unidade Lógica e Aritmética (ULA)
- Unidade de Controle
- Dispositivos de entrada/saída

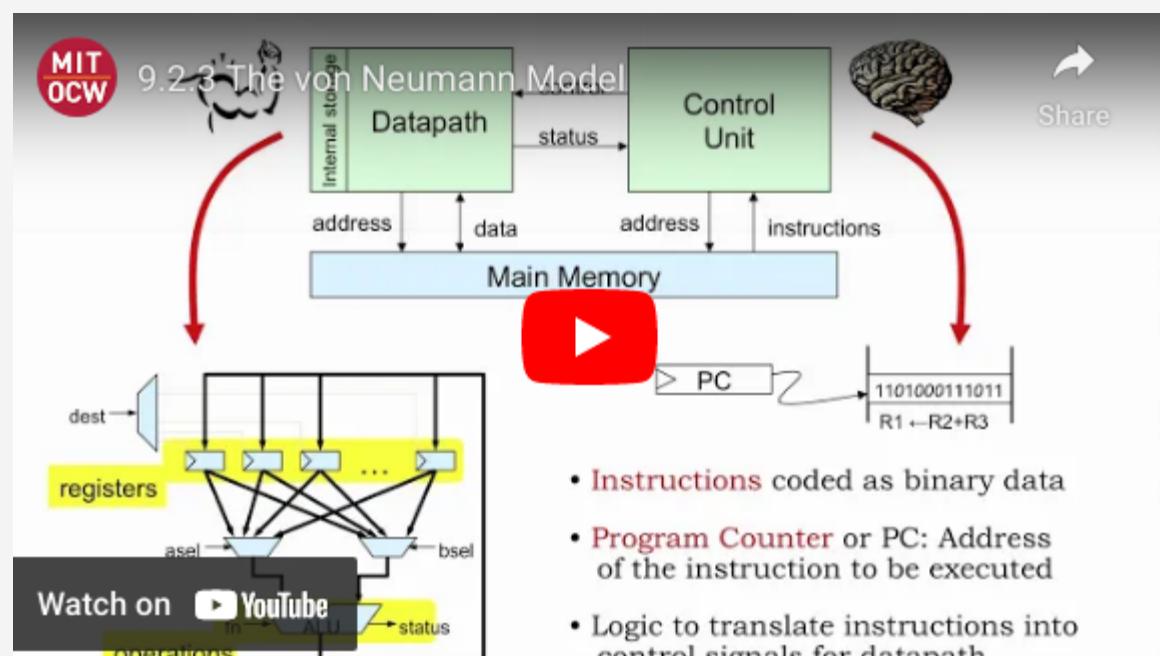
Computador IAS

- Memória: 4096 palavras de 40 bits (2 instruções de 20 bits (inteiro com sinal)).
- Instrução: 8 bits para indicar a instrução e 20 bits para endereçar a memória.
- Acumulador: registrador especial de 40 bits. Tem por função armazenar um operando e/ou resultado fornecido pela ULA.

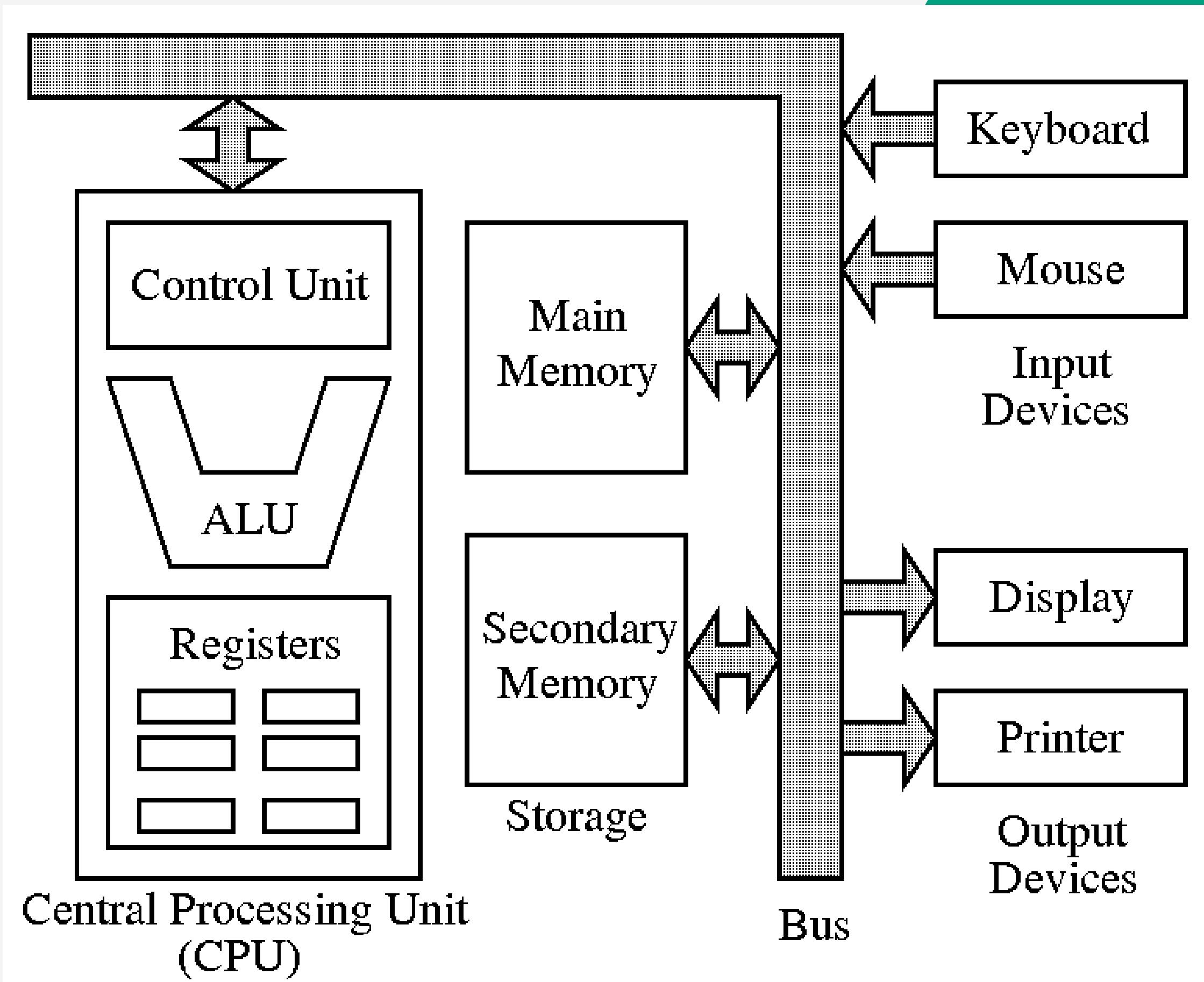
Arquitetura Von Neumann



Máquina de Von Neumann

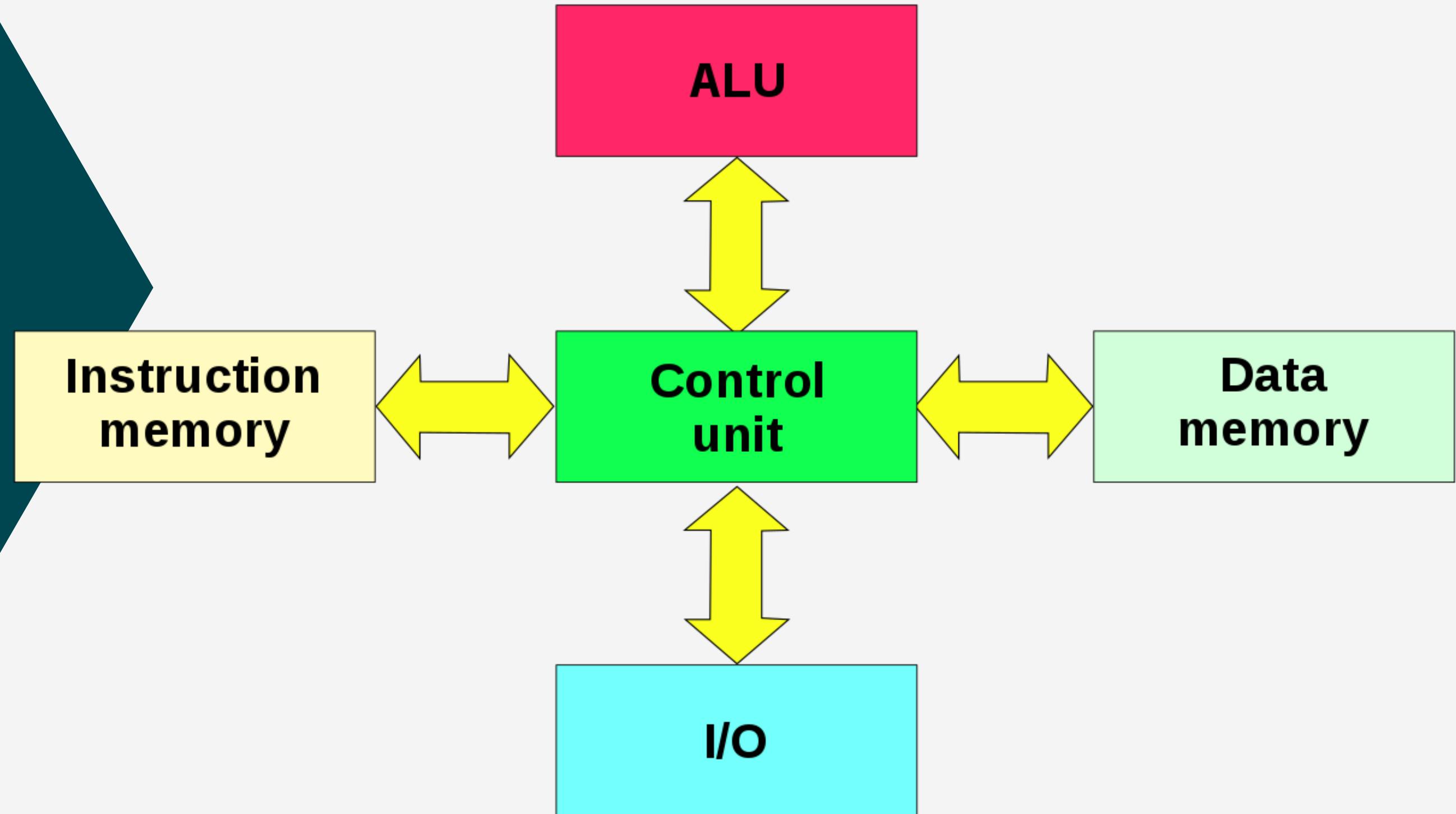


<https://www.youtube.com/watch?v=H0xGKKpKaRE>



Arquitetura de Harvard

Alternativa?



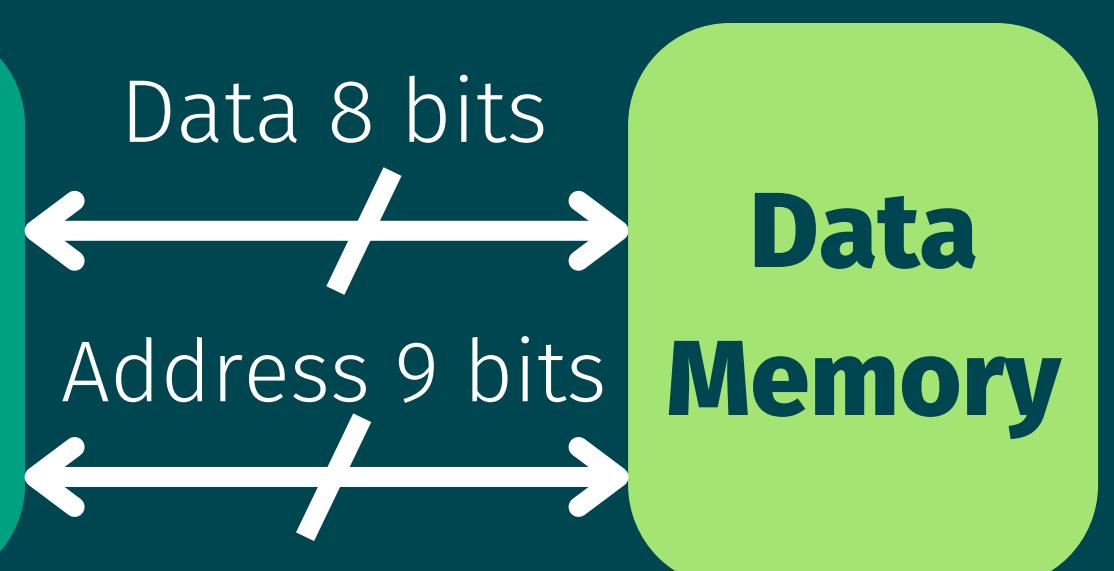
Von Neumann

Instruções:

- Operador + Operando

IAS

- 8 bits - Operação (opcode)
- 12 bits - Operando (endereço)



Harvard

- Separação entre memória de programa e memória de dados.
- Busca de instruções e operandos pode ser simultânea

Arquitetura Von Neumann x Harvard

Arquitetura mais simples

Não permite acesso simultâneo a memória de programa e a memória de dados. Com isso, tende a ser mais lento

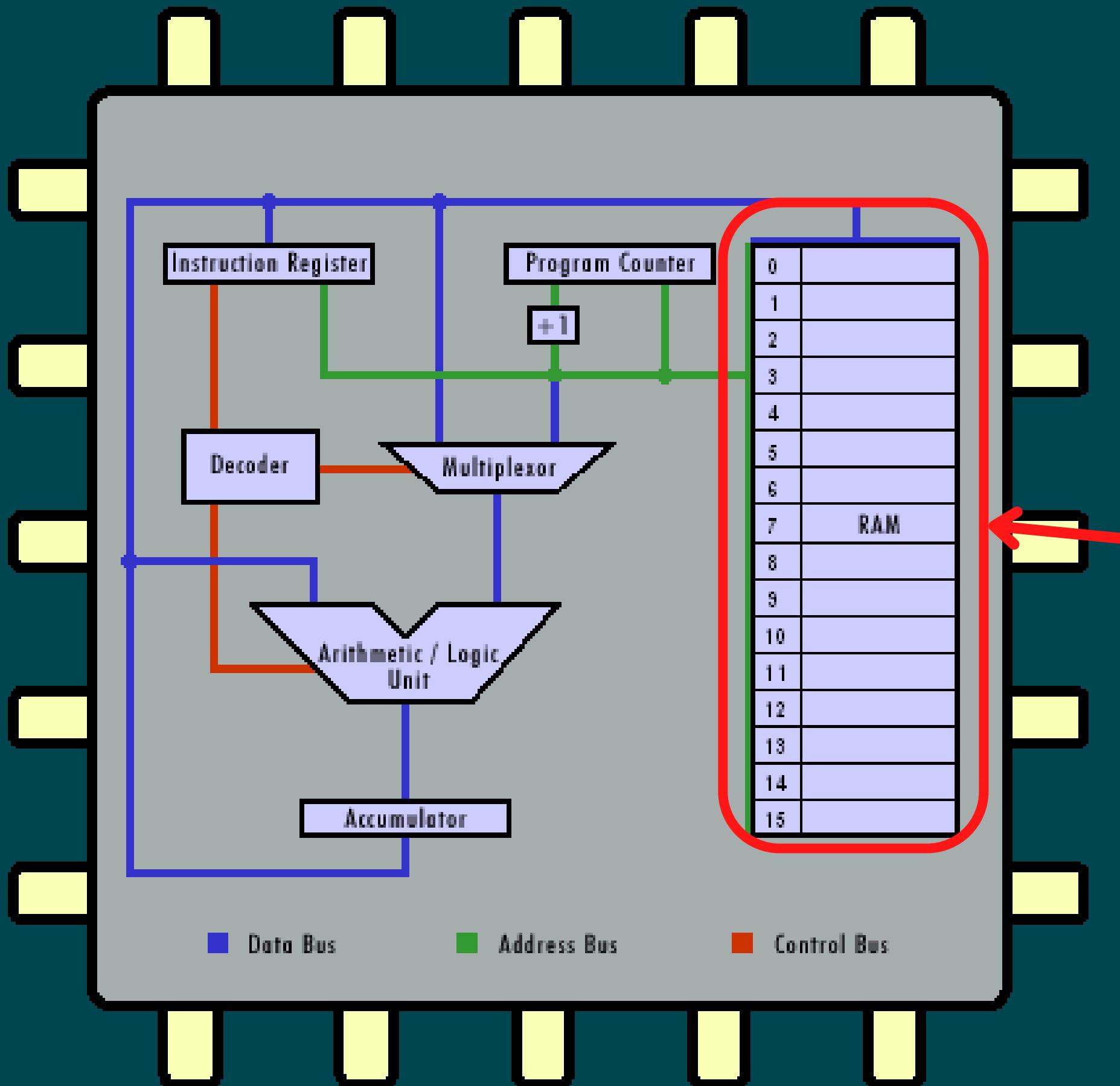
Geralmente CISC (Complex Instruction Set Computer)

Arquitetura mais complexa

Permite acesso simultâneo a memória de programa e a memória de dados. Com isso, tende a ser mais rápido

Geralmente RISC (Reduced Instruction Set Computer)

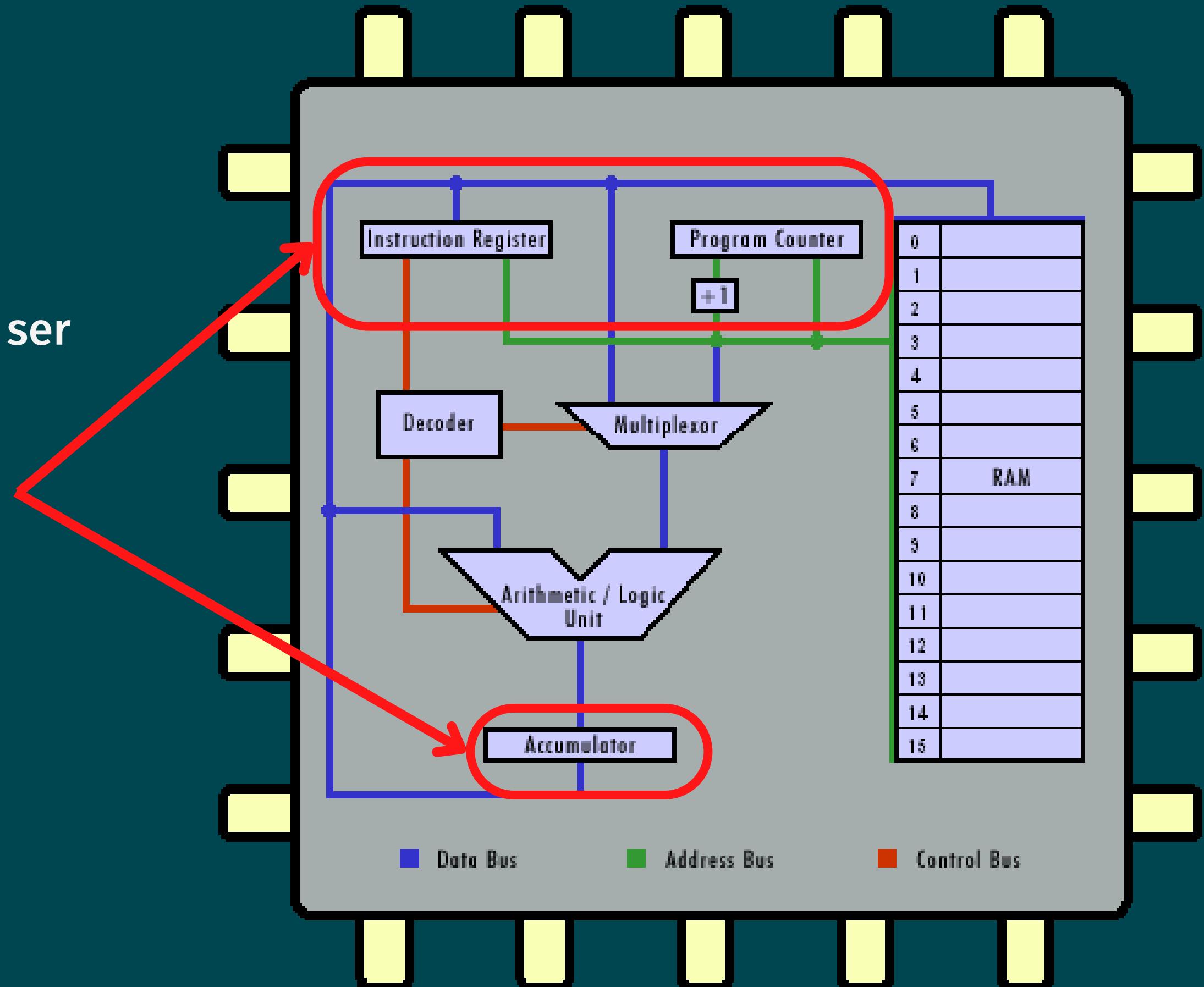
Componentes de uma arquitetura Von Neumann



- **RAM**: Memória de programa e de dados

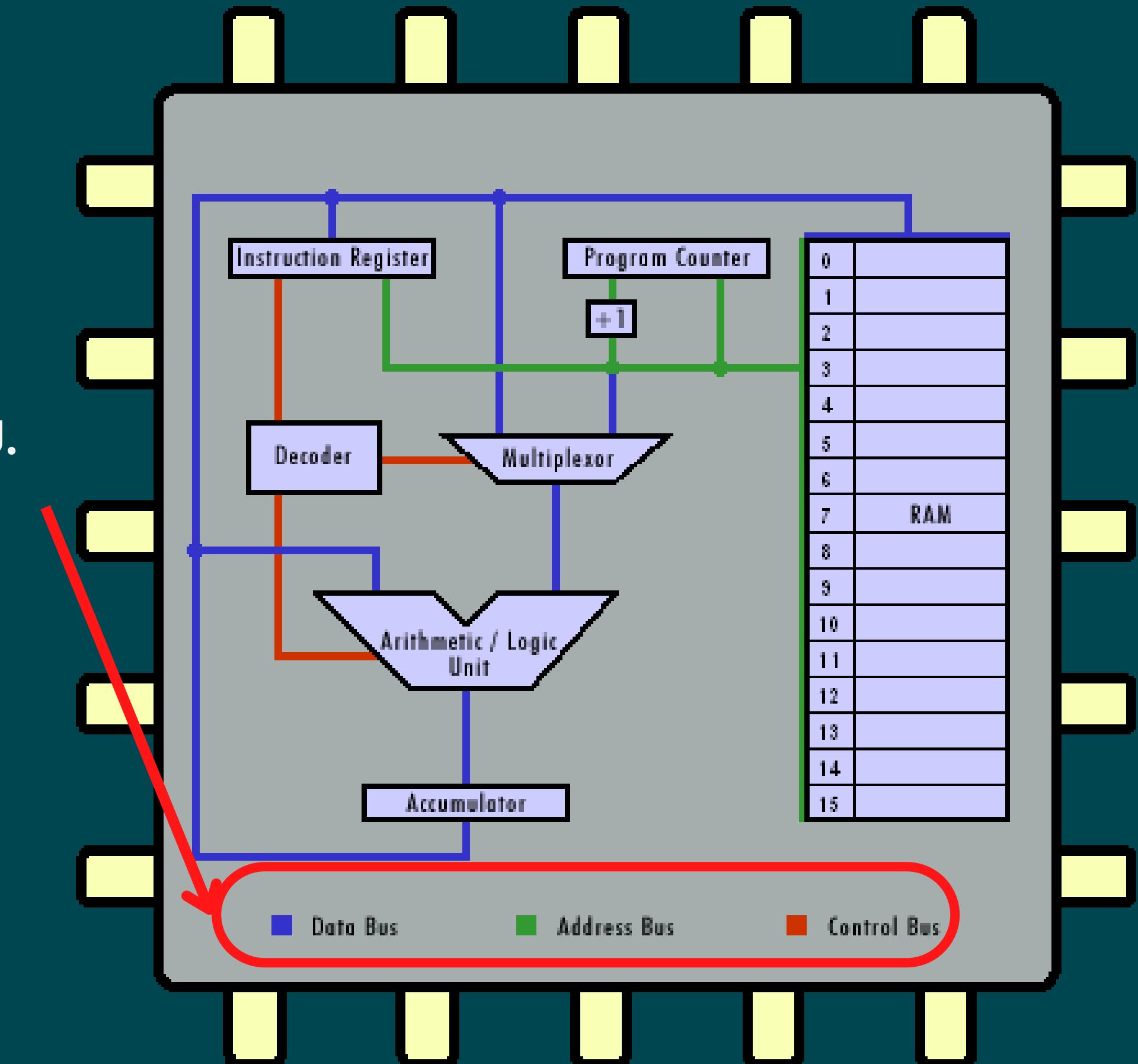
Componentes de uma arquitetura Von Neumann

- **Registradores:** Memórias que podem ser acessadas muito rapidamente.
 - Instruction Register (IR)
 - Program Counter (PC)
 - Accumulator (ACC)
 - Status Register - Flags: Flags

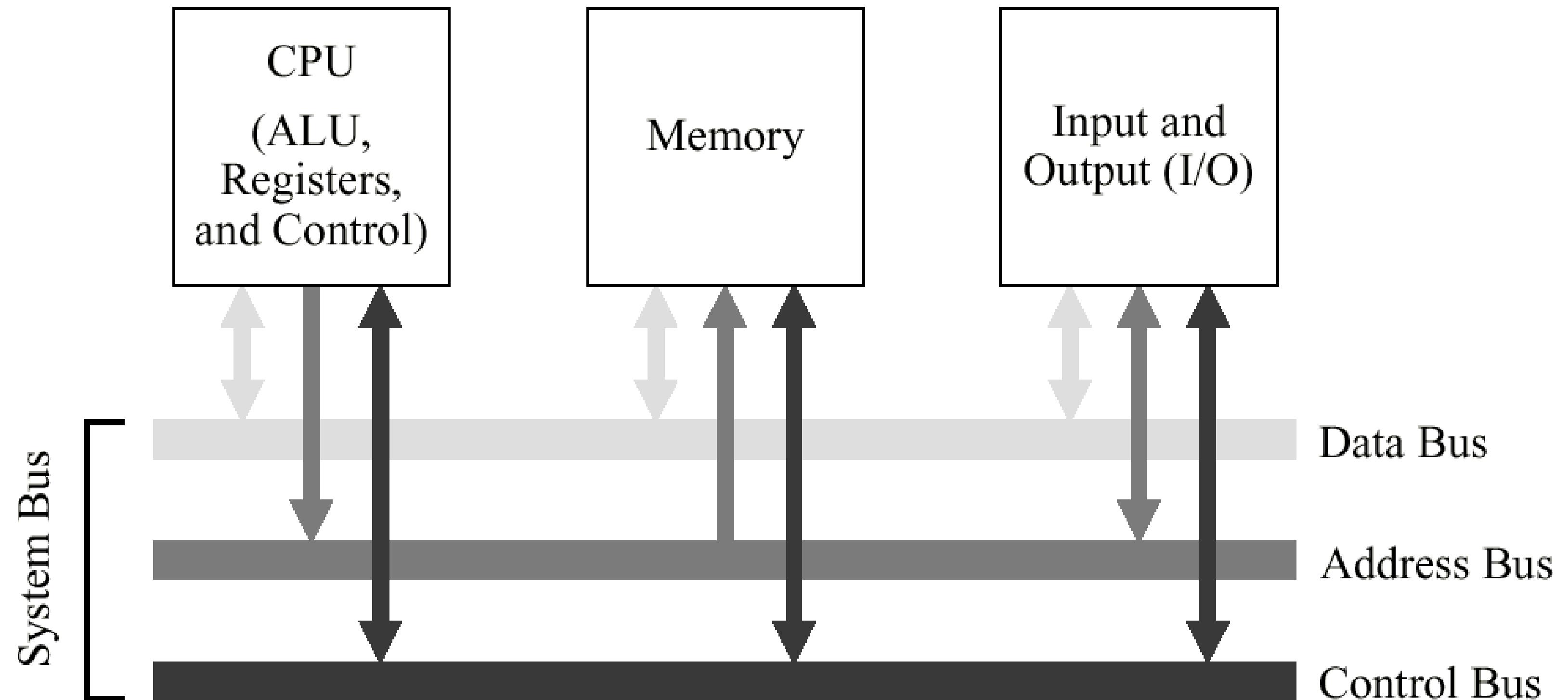


Componentes de uma arquitetura Von Neumann

- **Barramentos:** Ligam os componentes da CPU.
Os mais importantes são: Endereço, dados e controle

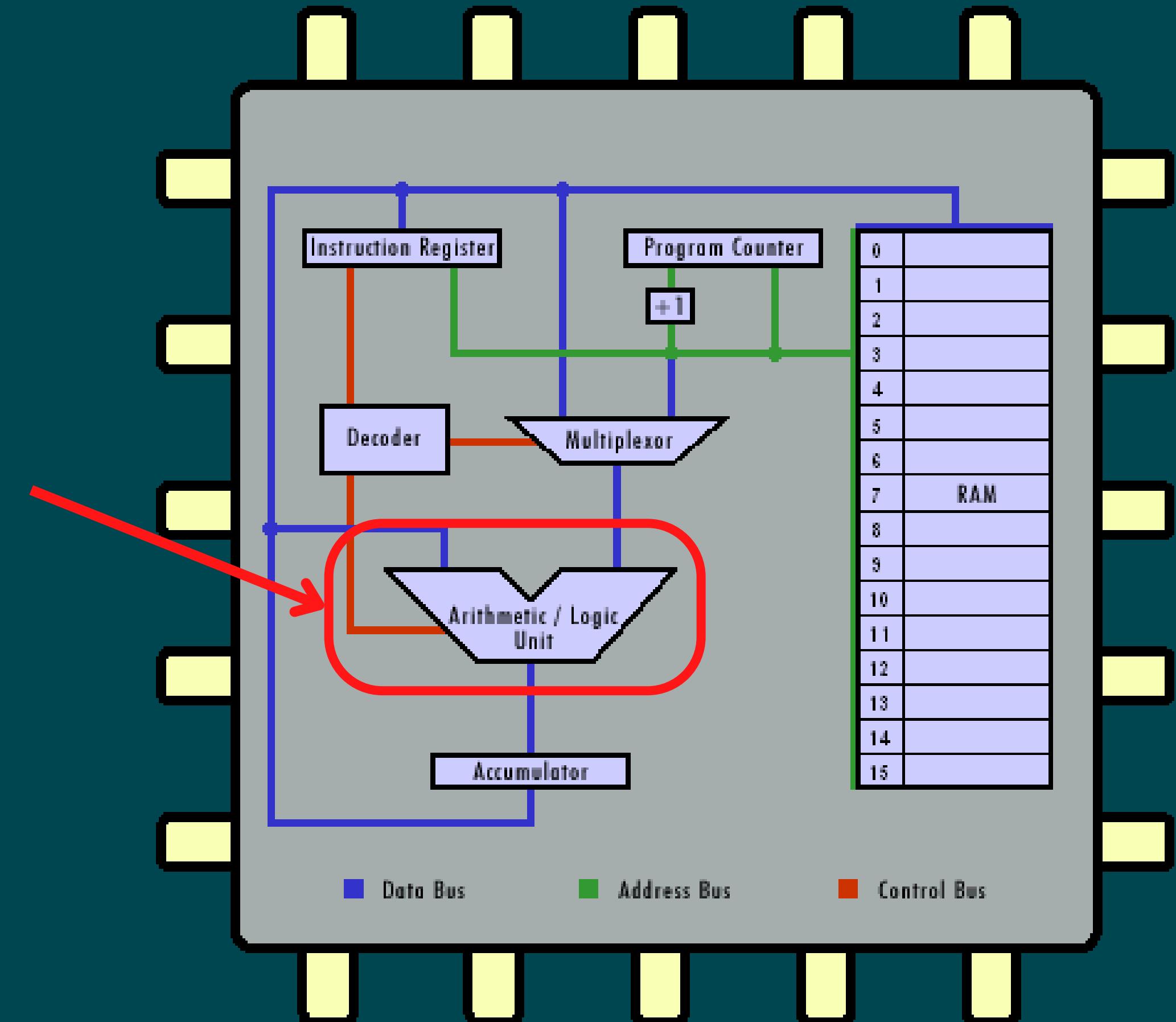


Barramentos



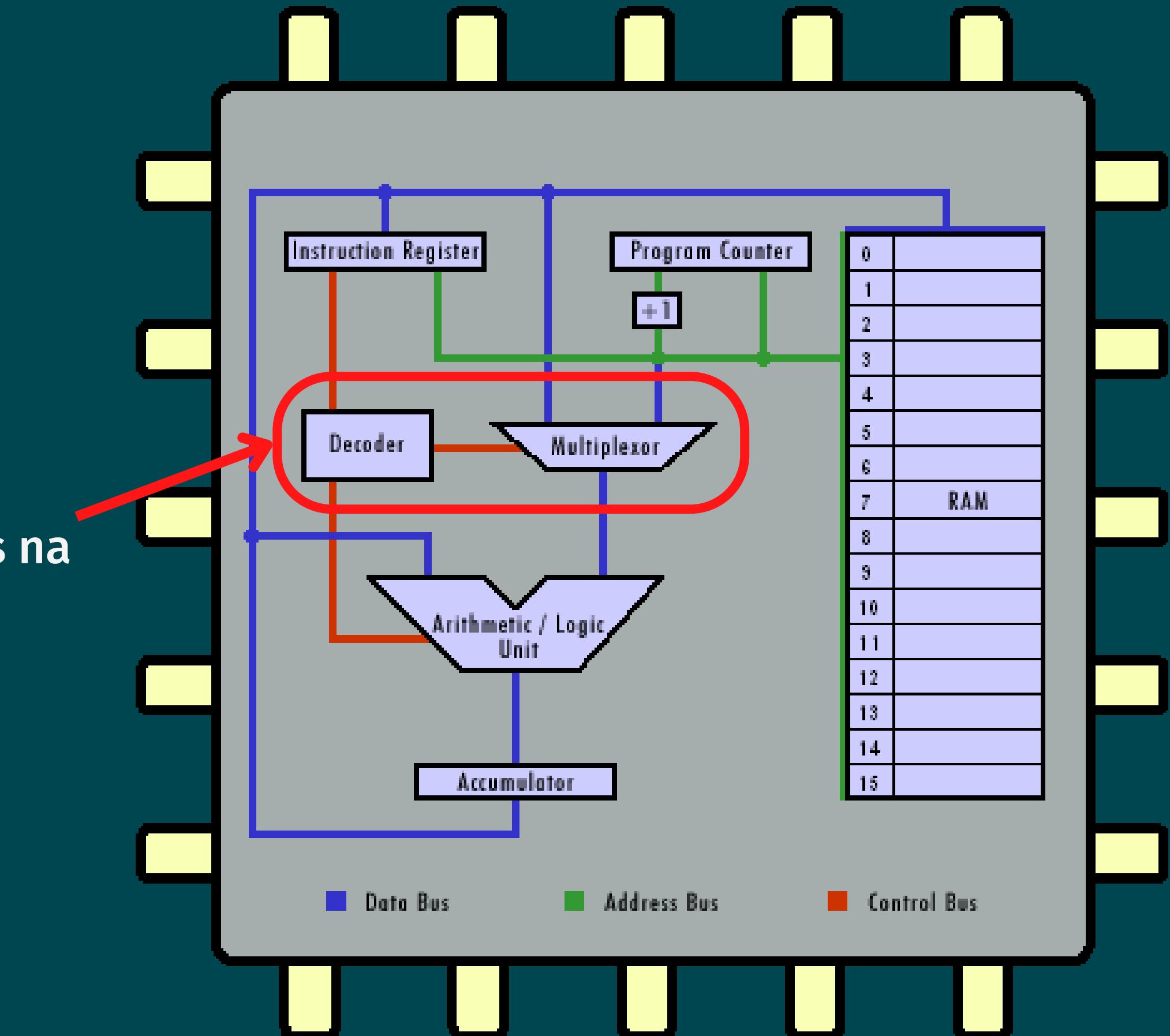
Componentes de uma arquitetura Von Neumann

- **ULA:** Unidade Lógica Aritmética. Ou ALU (Arithmetic Logic Unit). Realiza todas as operações matemáticas da CPU.



Componentes de uma arquitetura Von Neumann

- **Unidade de controle:** Responsável por garantir o fluxo de instruções e dados da memória para dentro da CPU.. Compostas na imagem pela Decoder e pelo Multiplexor



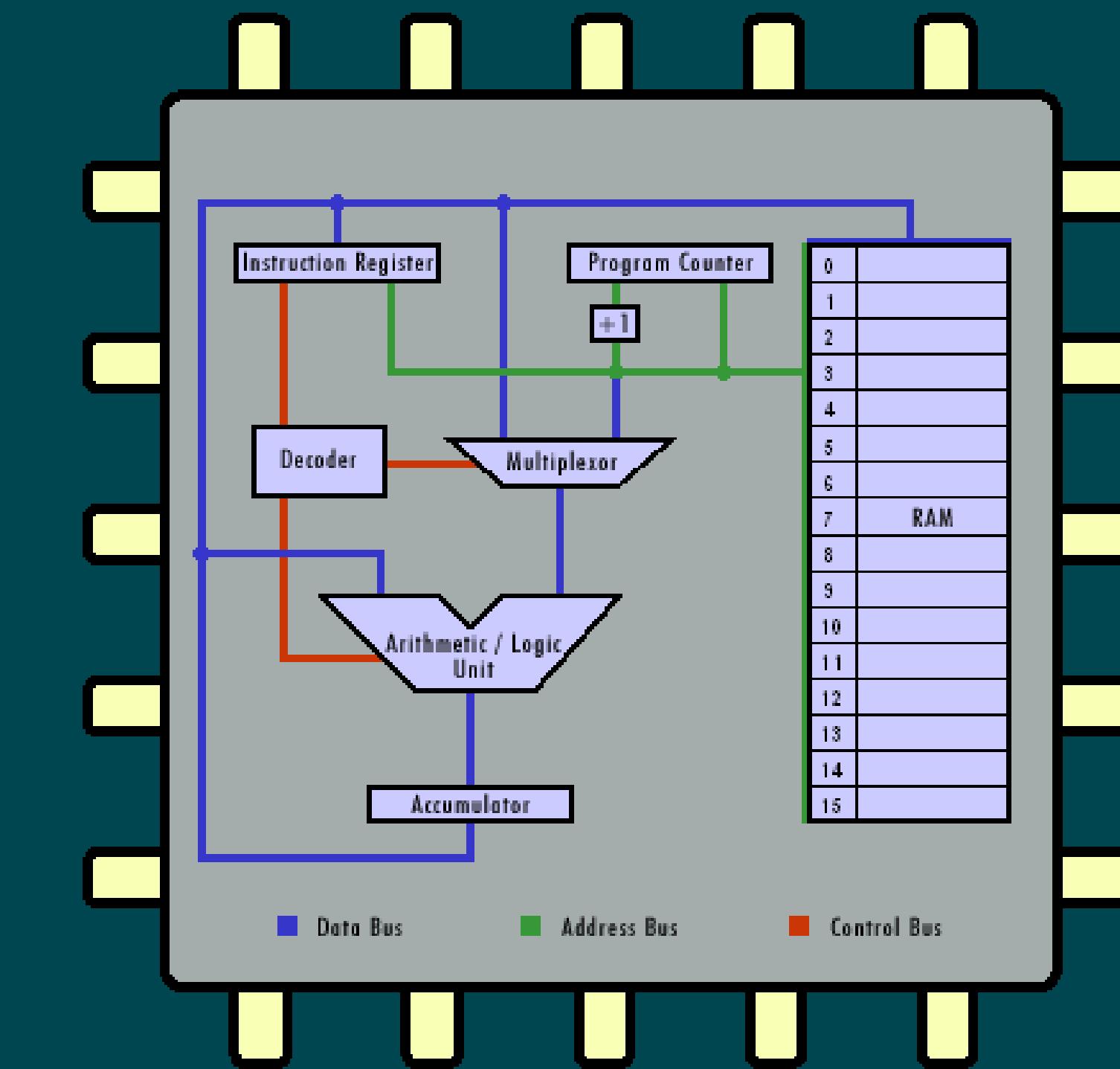
Arquitetura Von Neumann

Unidade de Controle

- Responsável por gerar todos os sinais que controla as operações no exterior da CPU, e ainda por dar todas as instruções para o correto funcionamento interno da CPU;
- Temos um decodificador de instruções, que analisa e decodifica o operador da instrução (OPcode);
- A unidade de controle executa **três ações básicas** intrínsecas e pré-programadas pelo próprio fabricante do processador:
 - **busca (fetch)**
 - **decodificação**
 - **execução**
- Todo processador, ao iniciar sua operação, realiza uma operação cíclica, tendo como base essas três ações. A unidade de controle usualmente é fixa, onde esta define o Conjunto de Instruções do Processador: **CPU Instruction Set**.

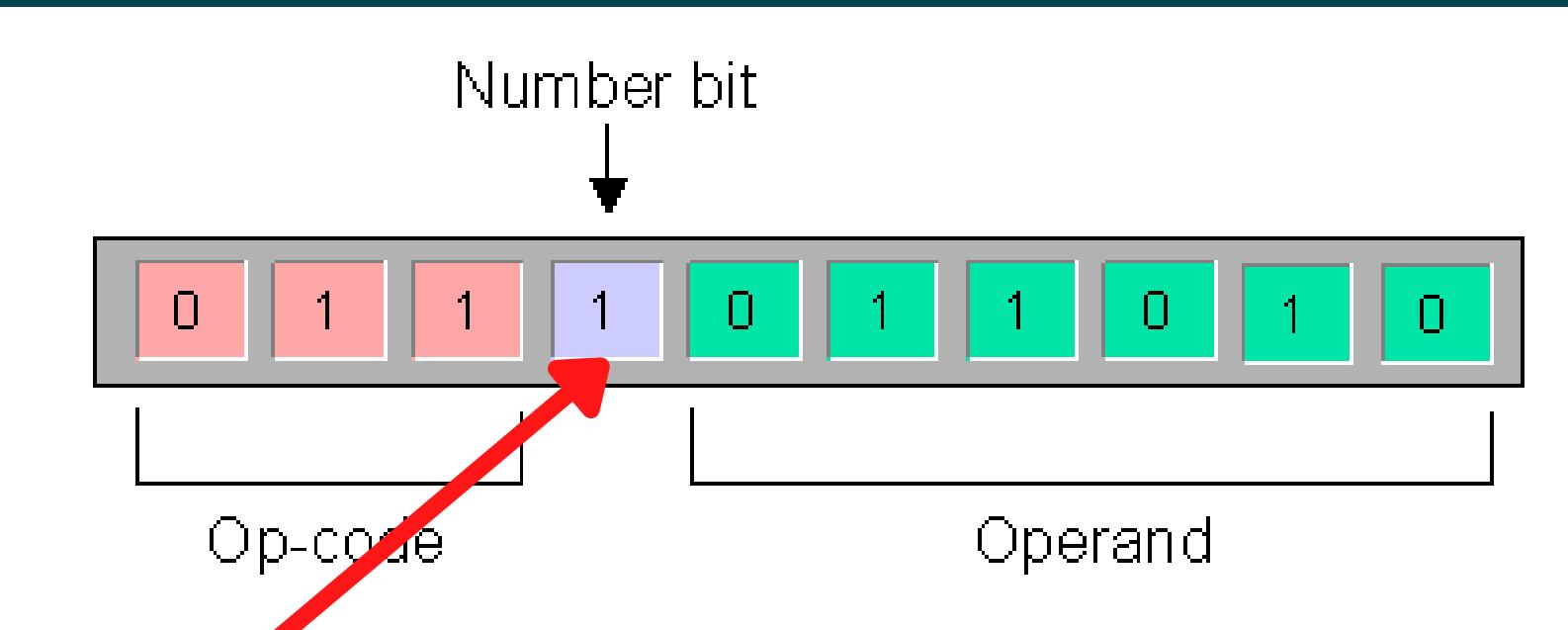
CPU Simples

Funcionamento....



Qualquer instrução de máquina é composta de duas partes:

- **OPcode**: Código da operação a ser realizada
- **Operand**: Fornece mais detalhes sobre o que deve ser realizada pela operação



Para a CPU simples apresentada temos também um **bit do meio** que distingue entre operandos que estão **endereçando memória (0)** e operandos que **são números (1)**. Quando este bit está em '**1**' significa que o operando é um **número**.

Código de Máquina da CPU simples

- Observe que todos os OPcodes possuem um mnemônico correspondente para facilitar a programação.
- Esses mnemônicos juntos são chamados de **Linguagem de Programação Assembly**.
- Programas escritos em assembly devem ser **convertidos** para o binário correspondente para que a CPU entenda. Isso é por um programa chamado de **Assembler**.

OP code	Mnemônico	Função	Exemplo
001	LOAD	Carrega o valor do operando no Acumulador	LOAD 10
010	STORE	Armazena o valor do acumulador no endereço especificado pelo operando	STORE 8
011	ADD	Adiciona o valor do operando ao Acumulador	ADD #5
100	SUB	Subtrai o valor do operando do Acumulador	SUB #1
101	EQUAL	Se o valor do operando for igual ao valor do Acumulador então ignore a instrução seguinte	EQUAL #20
110	JUMP	Atualize o contador de programa para o valor do operando e pule para a instrução desse endereço	JUMP 6
111	HALT	Pare a execução	HALT