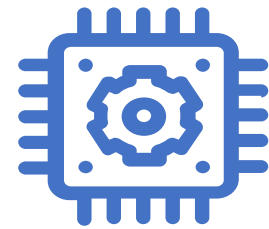
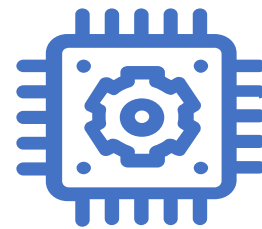
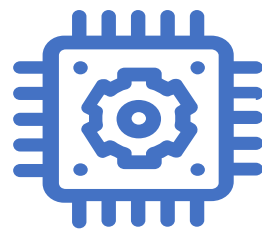
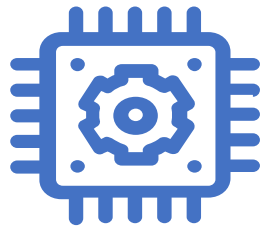
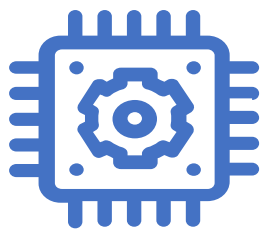




Aula 2 – Microprocessadores: Arquitetura e instruções

Disciplina: Microprocessadores e Microcontroladores
Professor: Daniel Gueter





Cronograma

29/04 – Aula 1 - Introdução da disciplina, revisão de conceitos e histórico da área

06/05 – Aula 2 - Microprocessadores: Arquitetura e instruções

13/05 – Aula 3

20/05 – Aula 4

27/05 – Aula 5

03/06 – Aula 6

10/06 – Aula 7

17/06 – Prova

24/06 – Exame (Prova substitutiva)



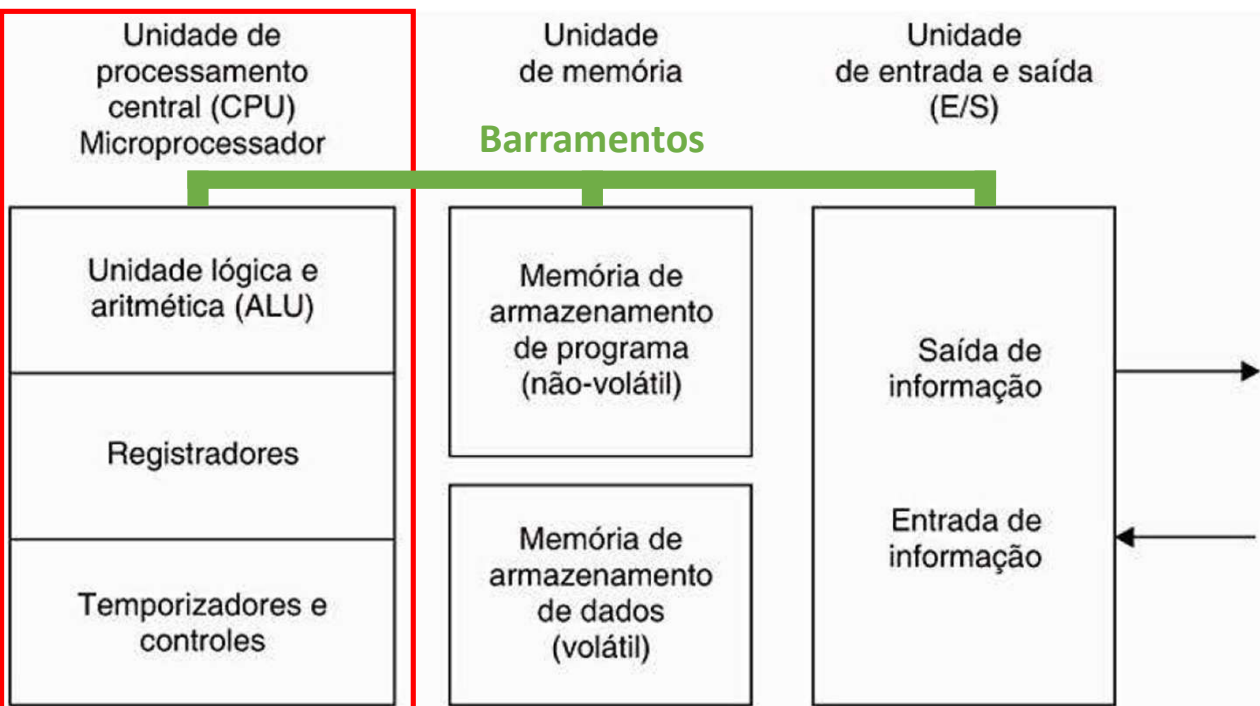
Microprocessadores e microcontroladores – O que são?

Microprocessador

- Também é denominado como unidade central de processamento (CPU).
- Fazendo analogia com o sistema de um ser humano, o microprocessador, ou CPU, de um microcomputador corresponde ao cérebro de um ser humano. Ele é **responsável pela busca de um programa na memória e por sua execução**.
- Fisicamente, é um circuito integrado (CI) constituído por **milhões de transistores que implementam uma variedade de circuitos**, como os vistos anteriormente (somadores, registradores, etc).
- Precisam de mais componentes para formar um microcomputador.



Arquitetura básica de um Microcomputador

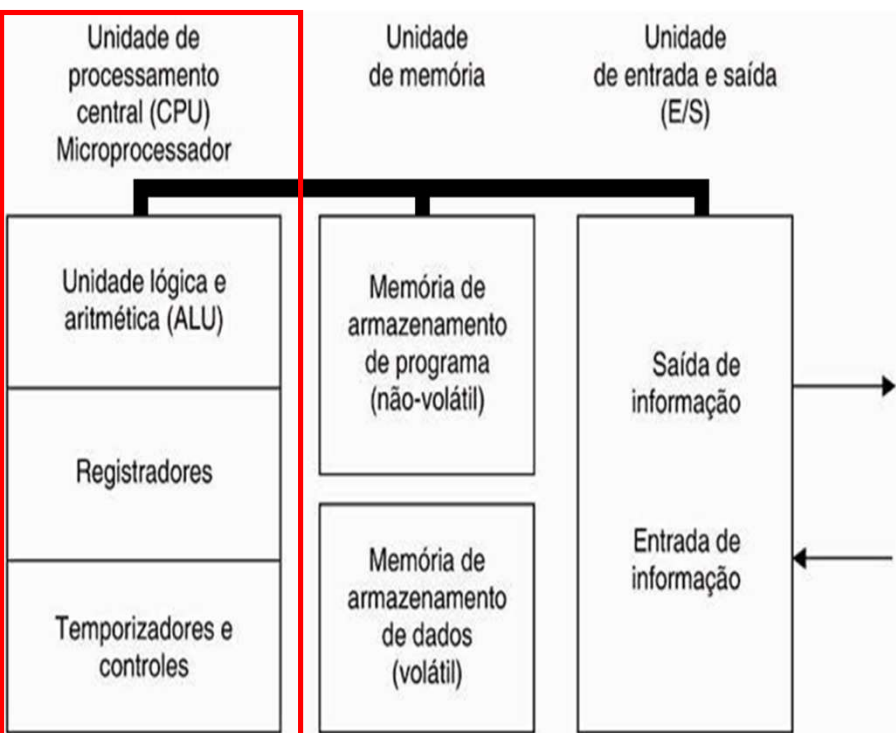


Blocos básicos de um microcomputador

- A arquitetura básica de um Microcomputador se baseia em três blocos: um **Microprocessador**, uma Unidade de memória e uma Unidade de entradas e saídas de informações (I/O).
- A comunicação entre esses três blocos é realizada a partir de **Barramentos**.



Arquitetura básica de um Microcomputador



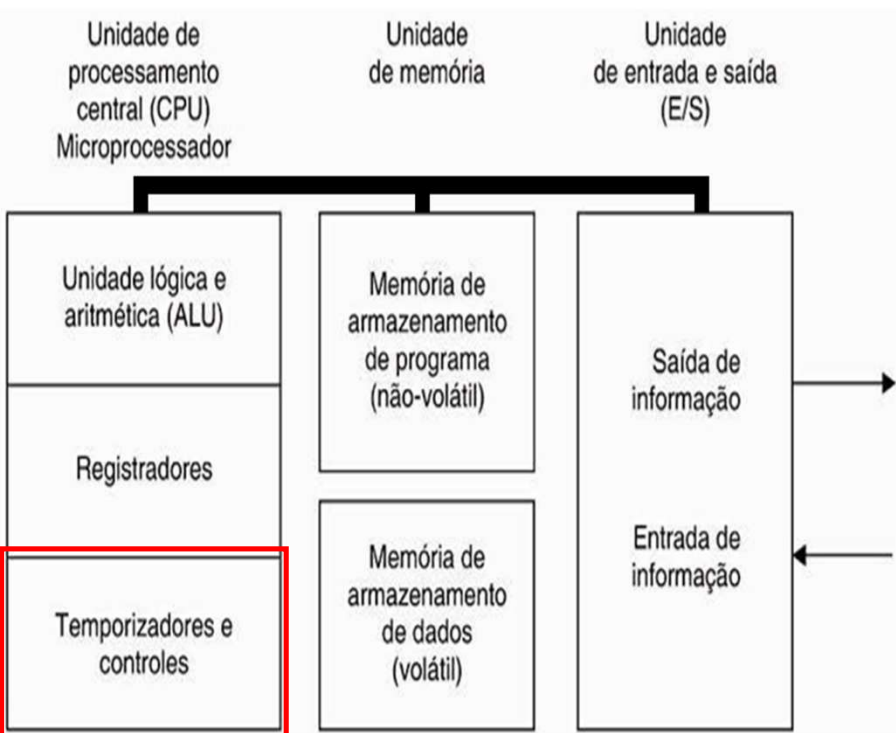
Blocos básicos de um microcomputador

Unidade de processamento central (CPU) ou Microprocessador

- Fazendo analogia com o sistema de um ser humano, o microprocessador, ou CPU, de um microcomputador corresponde ao cérebro de um ser humano. Ele é **responsável pela busca de um programa na memória e por sua execução**.
- Ao executar o programa, ela é responsável por **obter informações dos dispositivos de entrada** (Ex: Teclado), pelo **processamento dessas informações** (Ex: Cálculos lógicos), e pelo **resultado de um programa executado por meio dos dispositivos de saída** (Ex: Envio de dados para outro microcomputador).
- De maneira resumida, a CPU realiza duas funções básicas:
 - Busca e interpretação do programa alocado na memória por meio de **instruções**
 - Execução do programa, por meio de **instruções**



Arquitetura básica de um Microcomputador



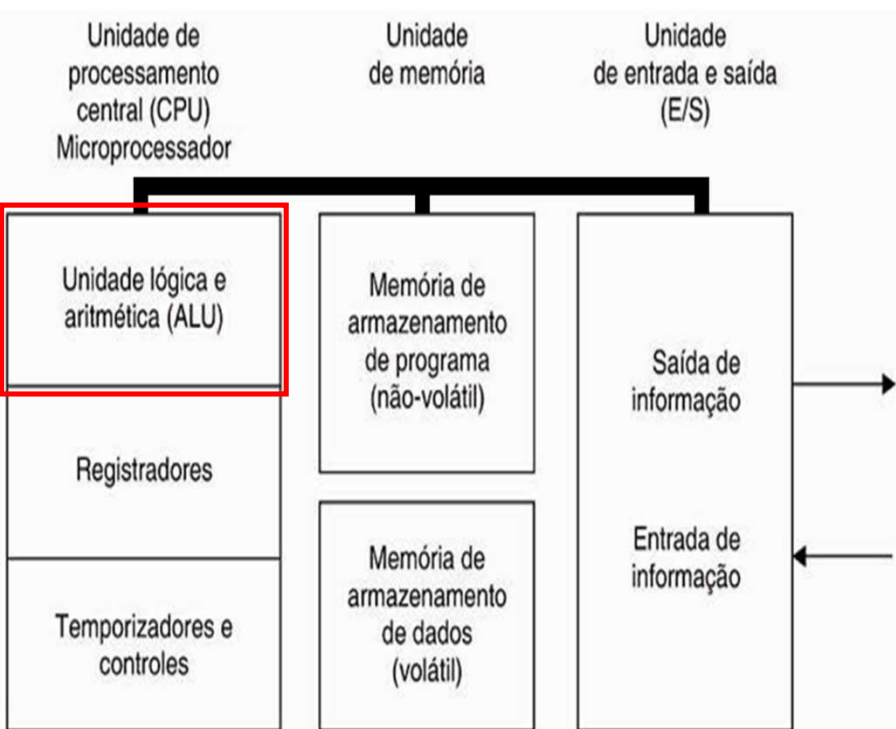
Temporizadores e controles (Unidade de controle)

- Busca instruções na memória principal para que seja executada.
- Controla o fluxo de informações da CPU para as unidades de memória e para as unidades de entrada e saída.
- Garante o sincronismo das informações
- Ex: Liberação de informações da ULA para os registradores internos, memórias externas e I/Os.

Blocos básicos de um microcomputador



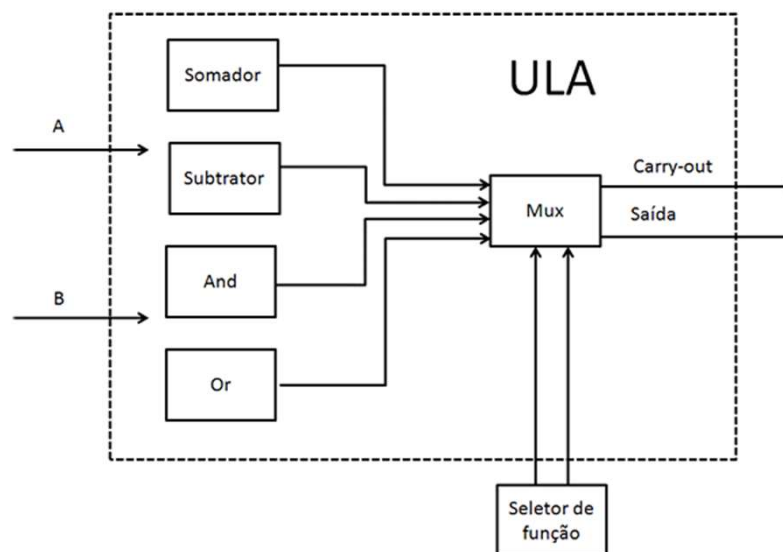
Arquitetura básica de um Microcomputador



Blocos básicos de um microcomputador

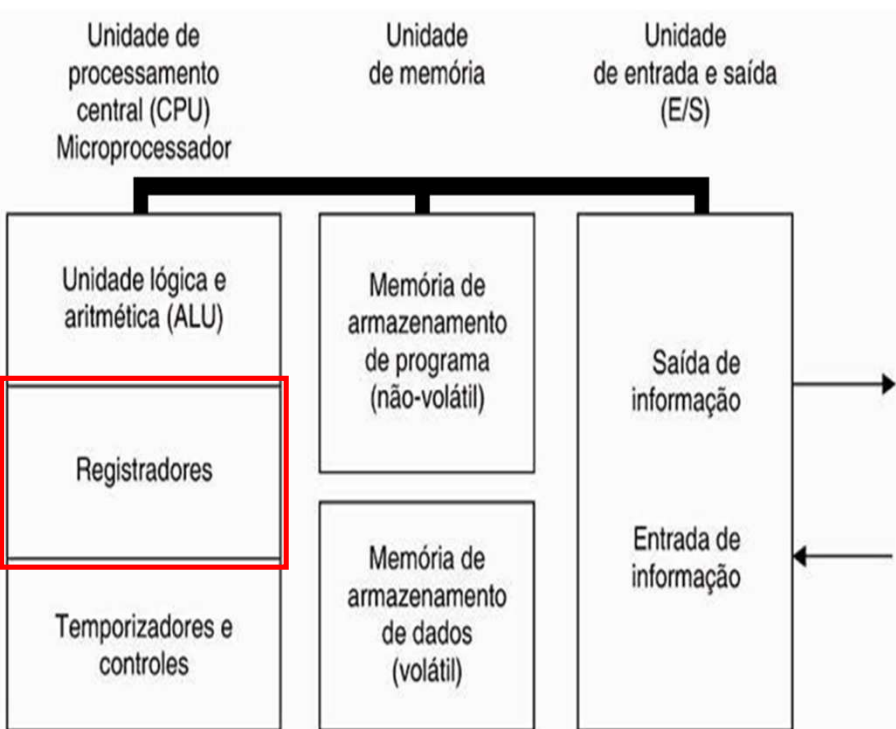
Unidade Lógica e Aritmética (ULA)

- Efetua operações aritméticas (matemáticas) e lógicas a partir de instruções buscadas na memória pela Unidade de Controle.
- Exemplo de uma ULA de 2 entradas de 4 bits e 4 operações:





Arquitetura básica de um Microcomputador



Blocos básicos de um microcomputador

Registradores

- Os registradores são memórias de alta velocidade situadas dentro da CPU.
- São formados por um determinado número de registradores de X bits em paralelo. (Ex: 32 registradores de 8 bits).
- São memórias voláteis que armazenam informações de processamento da CPU temporariamente, e são perdidas quando a CPU é desenergizada (desligada).
- Um microcomputador é chamado de microcomputador de 8 bits se ele for capaz de processar informações de 8 em 8 bits por vez (operações de movimentação de informações e operações aritméticas e lógicas).



Arquitetura básica de um Microcomputador

- A unidade para mediar a capacidade de processamento é MIPS (Milhões de informações por segundo), e ela depende da quantidade de bits processados em paralelo por ciclo e da velocidade de clock.

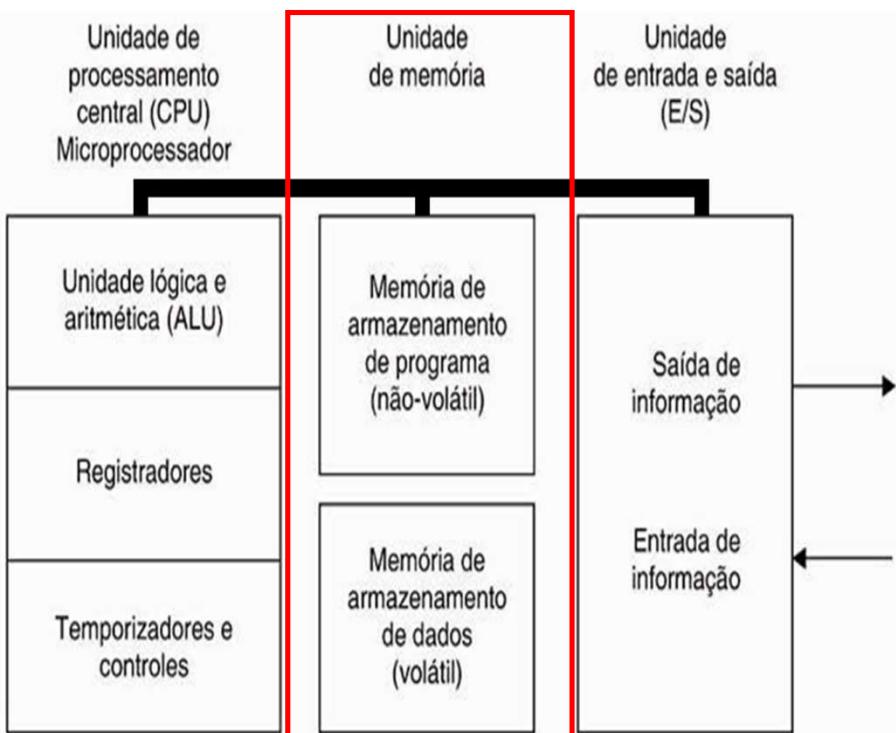
Nome	Data	Transistores	Microns	Velocidade do clock	Largura de dados	MIPS
8080	1974	6.000	6	2 MHz	8 bits	0,64
8088	1979	29.000	3	5 MHz	16 bits 8 bits	0,33
80286	1982	134.000	1,5	6 MHz	16 bits	1
80386	1985	275.000	1,5	16 MHz	32 bits	5
80486	1989	1.200.000	1	25 MHz	32 bits	20
Pentium	1993	3.100.000	0,8	60 MHz	32 bits 64 bits	100
Pentium II	1997	7.500.000	0,35	233 MHz	32 bits 64 bits	300

Nome	Data	Transistores	Microns	Velocidade do clock	Largura de dados	MIPS
Pentium III	1999	9.500.000	0,25	450 MHz	32 bits 64 bits	510
Pentium 4	2000	42.000.000	0,18	1,5 GHz	32 bits 64 bits	1,700
Pentium 4 "Prescott"	2004	125.000.000	0,09	3,6 GHz	32 bits 64 bits	7,000
Pentium D	2005	230.000.000	90nm	2,8 GHz 3,2 GHz	32 bits	
Core2	2006	152.000.000	65nm	1,33 2,33 GHz	32 bits	26,000
Core 2 Duo	2007	820.000.000	45nm	3 GHz	64 bits	53,000
Core i7	2008	731.000.000	45nm	2,66 GHz 3,2 GHz	64 bits	76,000

Evolução dos Microprocessadores



Arquitetura básica de um Microcomputador



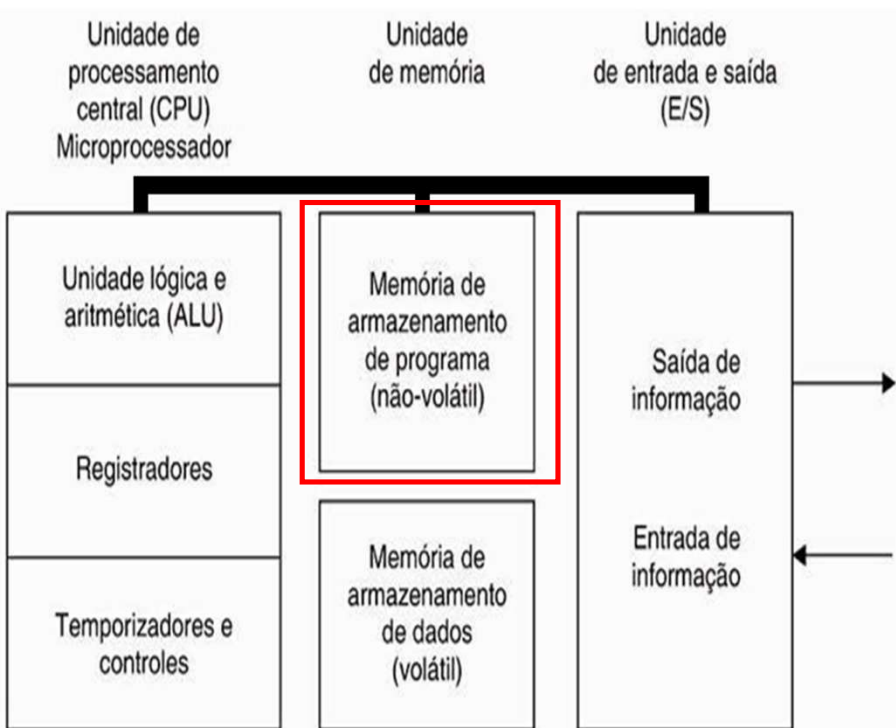
Unidade de memória

- Todo microcomputador possui uma Unidade de memória que é dividida em duas partes:
 - Memória de armazenamento de programa (não-volátil)
 - Memória de armazenamento de dados (volátil)

Blocos básicos de um microcomputador



Arquitetura básica de um Microcomputador



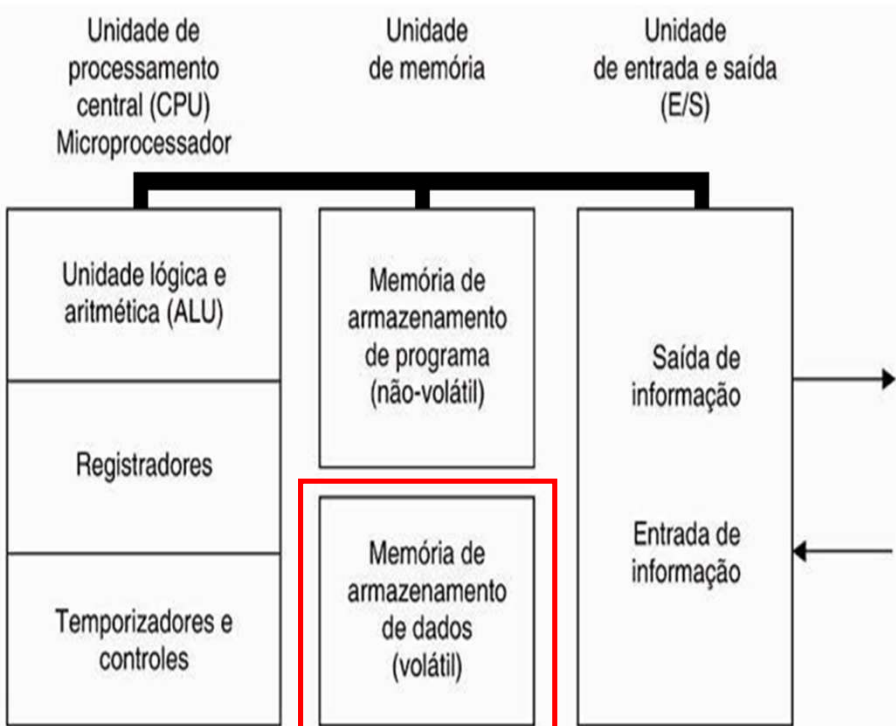
Blocos básicos de um microcomputador

Memória de armazenamento de programa (não-volátil)

- Memória em que estará armazenado o programa implementado que a CPU irá consultar para executar instrução por instrução.
- O microcomputador não funcionará se não existir um programa armazenado em sua memória de programa, sendo assim, esse programa não pode ser perdido ao desligar o microcomputador. Para isso, utiliza-se memórias do tipo **não-volátil**, às quais **mantem as informações mesmo após desenergizar e energizar o microcomputador**.
- Tipos de memórias não-voláteis: **ROM, PROM/OTP, EPROM, EEPROM**, HDs, SSDs, Memórias flash (Pen-drives). Geralmente, esse tipo de memória impossibilita a gravação de novos programas, só a leitura dos existentes (Exceção HDs, SSDs e memórias flash).



Arquitetura básica de um Microcomputador



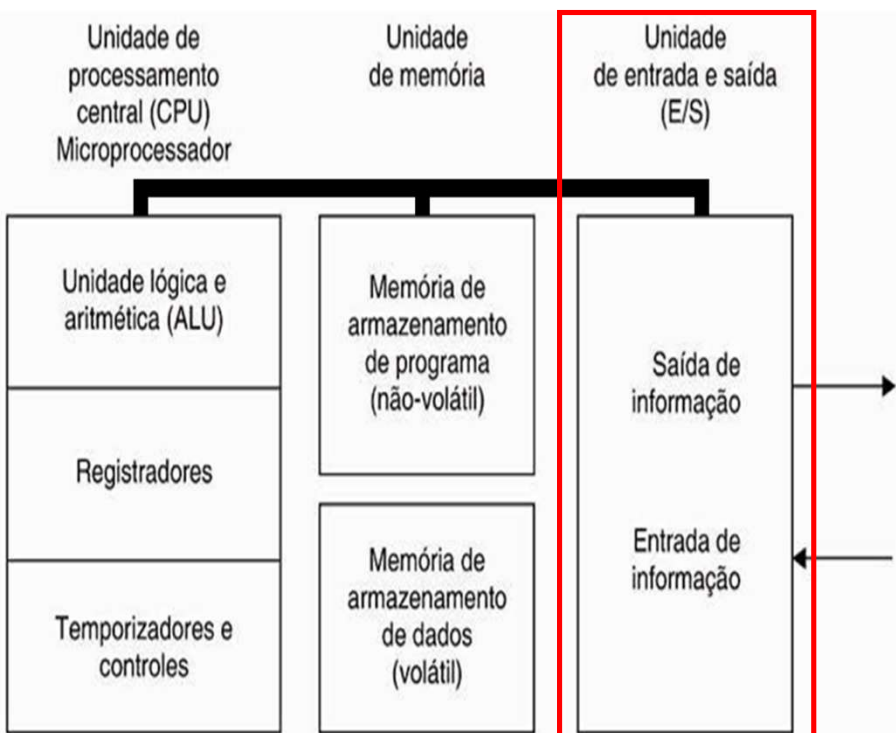
Blocos básicos de um microcomputador

Memória de armazenamento de dados (volátil)

- Memória onde são armazenadas de maneira veloz informações temporárias. São utilizadas para a escrita e leitura de informações temporárias das inúmeras operações de processamento.
- Exemplo: quando uma tecla é pressionada no teclado, a CPU lê o sinal gerado pelo hardware e armazena-o na memória, para depois interpretar qual tecla foi acionada e exibir o resultado no monitor.
- Essas memórias são chamadas do tipo **volátil** pois ao **desenergizar o microcomputador, as informações são perdidas.**
- Exemplo de memória volátil: **RAM.**



Arquitetura básica de um Microcomputador



Unidade de entradas e saídas (I/O)

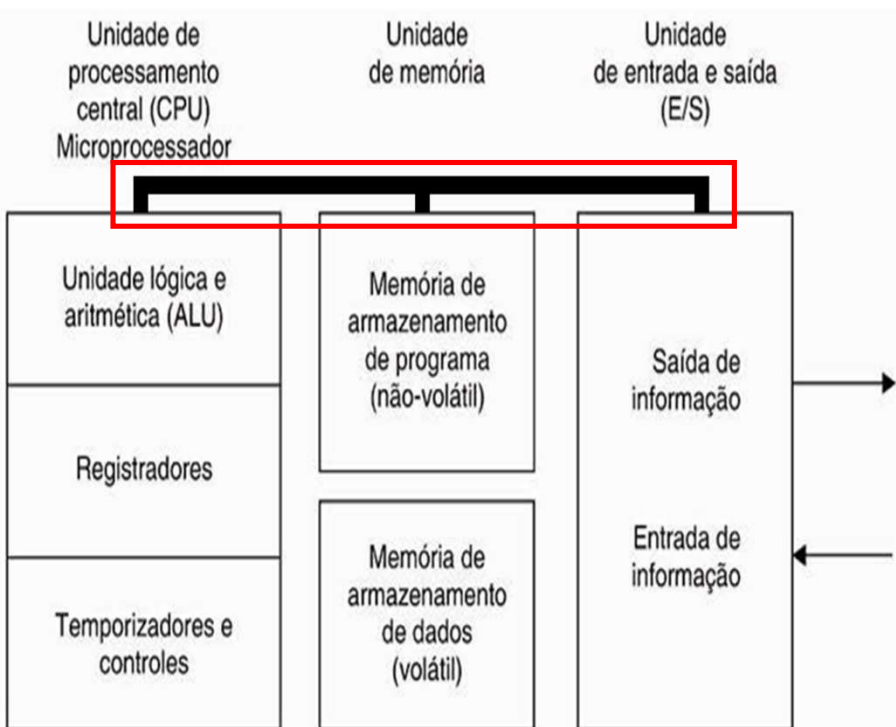
- Também conhecida como unidade de I/O (*Inputs/Outputs*)
- Tem o objetivo de fazer a interface com o mundo externo ao microcomputador, por meio de CIs capazes de ler e armazenar informações de entradas e saídas.
- Exemplo de entradas (*Inputs*): informações do teclado, de um sensor, ou de um canal de comunicação serial.
- Exemplos de saídas (*Outputs*): Alto-falantes, LEDs, telas, interfaces de comunicação.

Blocos básicos de um microcomputador



Arquitetura básica de um Microcomputador

Barramentos



Blocos básicos de um microcomputador

- Vias de comunicação entre os blocos, constituídas por trilhas de cobre (Ex: cabos), nas quais fluem informações em paralelo. Temos 3 principais tipos de barramentos:
- **Barramento de endereços (unidirecional):** por onde são definidos os endereços das posições de memória do programa que a CPU vai buscar instruções e também definidos endereços de memória de dados ou dos dispositivos de I/O.
- **Barramento de temporização e controle (unidirecional):** por onde a CPU defini os sinais de temporização e controle para gerenciar o fluxo de informações.
- **Barramento de dados (bidirecional):** por onde a CPU envia/recebe dados da memória ou dos I/Os.



Como ocorre o
funcionamento interno de
um microcomputador?

Por meio de instruções!

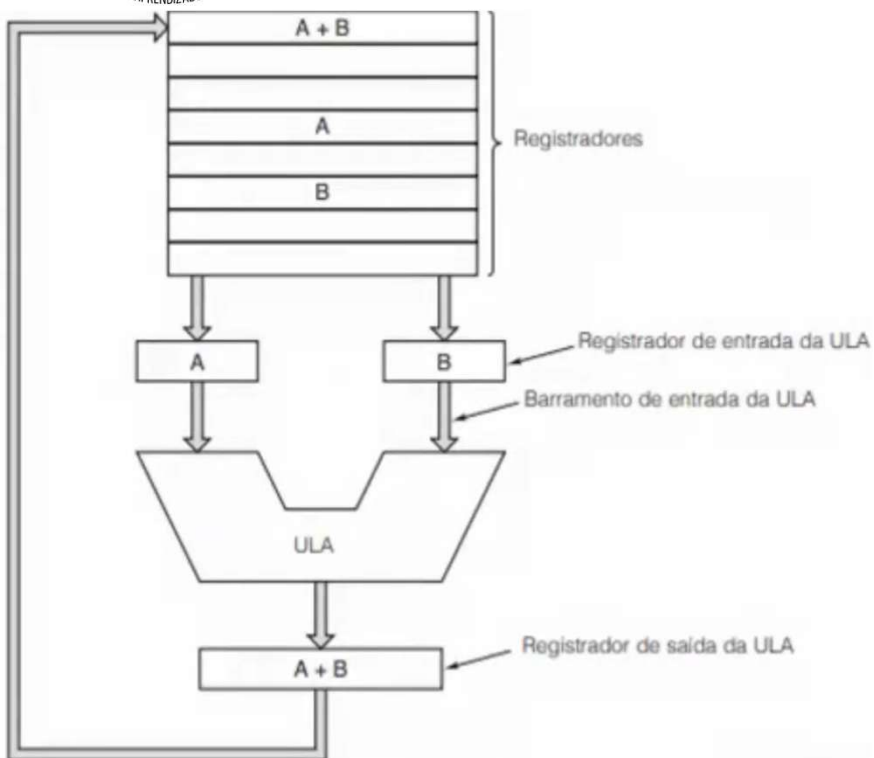


Como começam as instruções

1. Ao ligar um microcomputador, a primeira coisa que acontece é a inicialização de um registrador interno da CPU chamado de **Contador de Programa**. Por definição, ele sempre **contém o endereço e indica a próxima instrução a ser buscada na memória de programa** e executada.
2. Uma vez que a instrução é executada, o **Contador de Programa é ajustado para apontar para a próxima instrução** do programa.
3. Paralelamente, atualiza-se um registrador da CPU chamado de **Registrador de instrução**, o qual sempre **mantém a instrução que está sendo executada no momento**.



Exemplo de operação e Caminho dos dados



Operação de soma com arquitetura de Von-Neumann

Operação: **Soma de dois dados**

Ordem de instruções:

1. CPU busca na memória valores que querem ser somados (A e B), e grava nos registradores internos.
2. Esses valores passam para registradores de entrada da ULA.
3. A Unidade de Controle da a instrução para a ULA de realizar uma soma.
4. A ULA realiza a soma e armazena em um registrador de saída da própria ULA, que faz parte dos registradores internos.



Ciclo de instruções – Ciclo Buscar-Decodificar-Executar

Ordem de instruções da CPU:

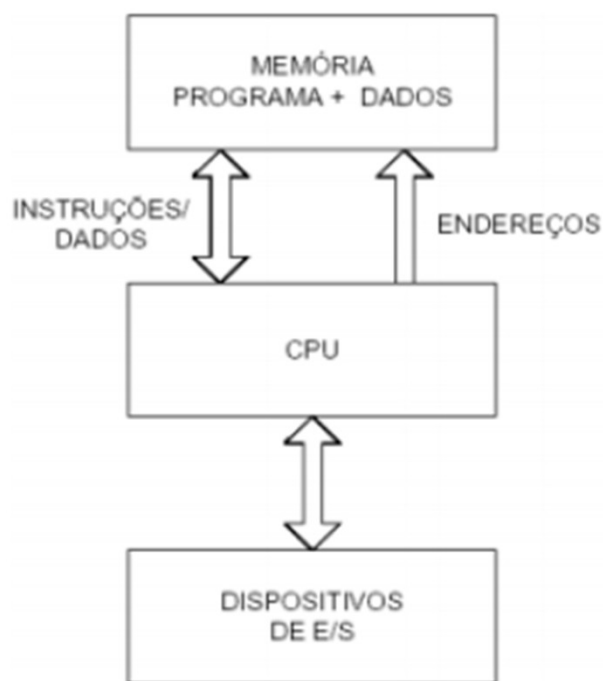
1. A CPU busca a próxima instrução da memória de programa até o Registrador de instrução.
2. É alterado o Contador de programa para ser apontada a próxima instrução na memória.
3. É determinada o tipo de instrução a ser executada.
4. Se a instrução precisar de alguma informação da memória, é determinada onde essa informação está.
5. Se for necessário, é gravada a informação em um dos registradores internos da CPU.
6. A instrução é executada (Ex: ULA realizando soma).
7. Volta para a primeira instrução da ordem para realizar o ciclo novamente executando uma nova instrução.



Arquitetura de memória: Von Neumann e Harvard

Von-Neumann

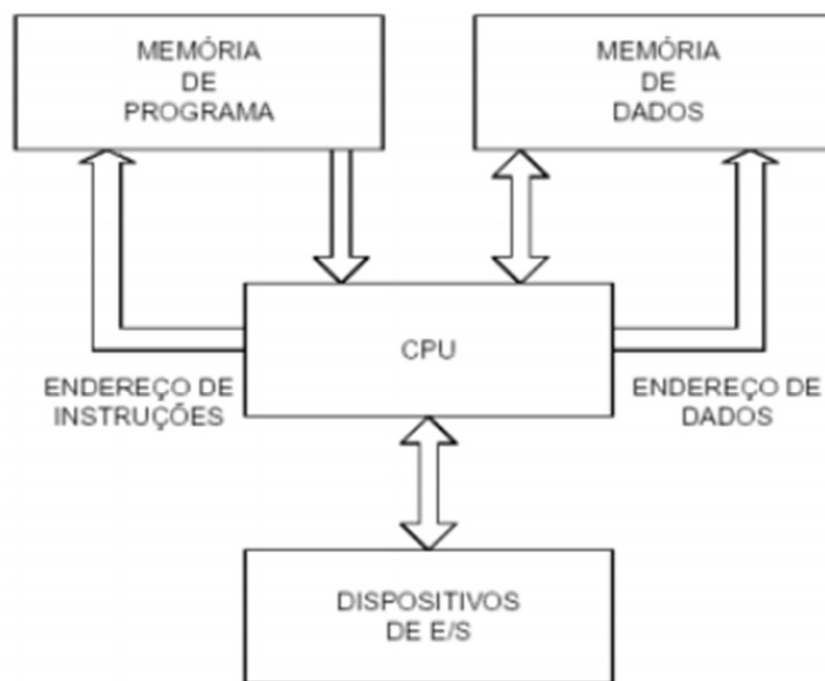
Fisicamente uma memória só, com posições dedicadas para programas e dados.



Arquitetura
Von-Neumann

Harvard

Memória de dados e de programa separadas.



Arquitetura
Harvard

Diagrama de blocos das arquiteturas de Von-Neumann e de Harvard



Arquitetura de memória: Von Neumann e Harvard

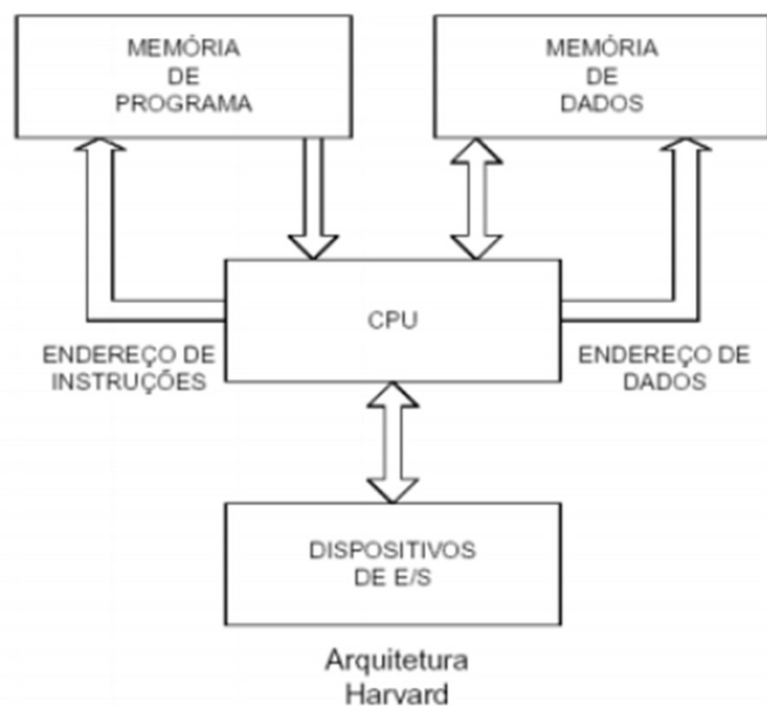


Von-Neumann

- Arquitetura mais simples do que a de Harvard
- Possui um único barramento para programa (instruções) e dados.
- Geralmente utiliza uma arquitetura CISC (Próximos slides!).
- Permite produzir um conjunto complexo de código de instruções para o microcomputador, possuindo muito mais tipos de instruções.



Arquitetura de memória: Von Neumann e Harvard



Harvard

- Arquitetura mais rápida do que a de Von-Neumann
- Possui barramentos separados para programa (instruções) e dados, podendo usá-los paralelamente.
- Geralmente utiliza uma arquitetura RISC (Próximos slides!).
- Produz um conjunto de códigos de instrução mais simples, necessitando mais linhas de código do que a arquitetura de Von-Neumann.



Arquitetura ISA (Instruction Set Architecture): CISC e RISC

CISC: Complex Instruction Set Computer

- Utiliza conjunto de instruções complexas
- Mais operações com menos linhas de códigos
- Necessita de vários ciclos de clock para processar instruções
- Geralmente, processadores com maior desempenho e consequentemente maior consumo de energia.
- Exemplos equipamentos que usam processadores CISC: Computadores pessoais e servidores.

RISC: Reduced Instruction Set Computer

- Utiliza conjunto de instruções simples
- Necessita de mais linhas de código
- Executa cada instrução em um único ciclo de clock
- Geralmente, processadores com menor desempenho e consequentemente menor consumo de energia.
- Exemplos equipamentos que usam processadores RISC: Arduino, Fones de ouvidos.

Nota: Atualmente, existem computadores híbridos, utilizando internamente tanto arquiteturas CISC quanto RISC.



Arquitetura ISA (Instruction Set Architecture): CISC e RISC

Exemplo: Código em Assembly (Programação de baixo nível) para realizar a operação de multiplicar dois números contidos nos endereços 0 e 3 da memória e armazenar o resultado de volta na posição 0.

Em microprocessadores CISC, o objetivo é executar a tarefa com o menor número de códigos possíveis (*assembly*). Assim, um microprocessador CISC hipotético poderia ter a seguinte instrução:

```
MULT 0,3 //multiplica o conteúdo do endereço 0 com o do endereço 3;  
        //armazena o resultado no endereço 0.
```

← CISC

Para um microprocessador RISC, a resolução do problema seria feita por algo como:

```
LOAD A,0 //carrega o registrador A com o conteúdo do endereço 0;  
LOAD B,3 //carrega o registrador B com o conteúdo do endereço 3;  
MULT A,B //multiplica o conteúdo de A com o de B. Resultado fica em A;  
STORE 0,A //armazena o valor de A no endereço 0.
```

← RISC

Código em Assembly para uma arquitetura CISC e RISC



Analizando o processador Intel Pentium (1993)

Intel Pentium (1993)

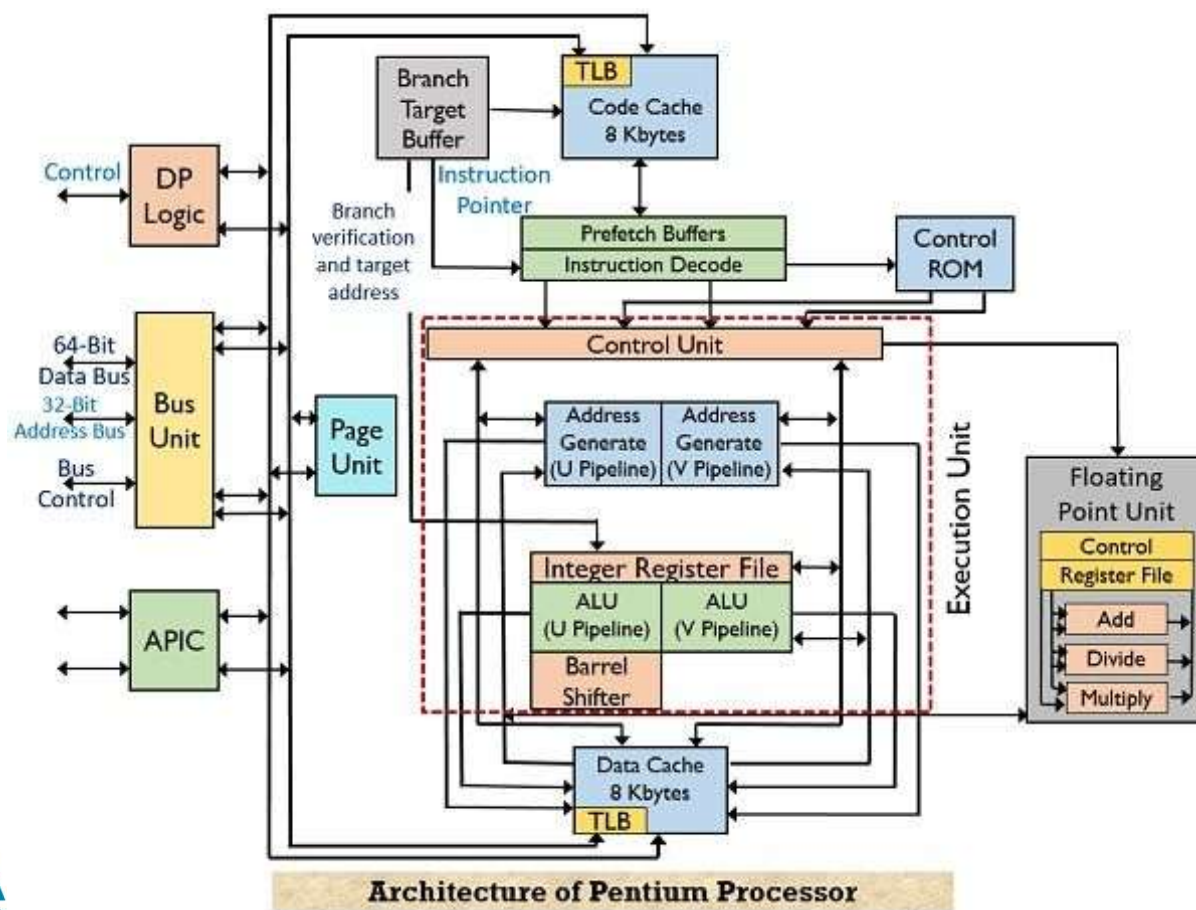


Microprocessador Intel Pentium de 1993

- Aproximadamente 3 milhões de transistores com uma litografia de 0,35 μm
- Processamento de 32 ou 64 bits
- Clock de 60 MHz
- 100 MIPS (Milhões de instruções por segundo)
- Arquitetura de memória: Von-Neumann
- Arquitetura ISA: CISC



Analizando o processador Intel Pentium (1993)



Arquitetura interna do microprocessador Intel Pentium de 1993

Aula 2 - Microprocessadores: Arquitetura e instruções