

SEL-0415 **Introdução à Organização de Computadores**

Estrutura Básica de um Computador

Aula 2

Prof. Dr. Marcelo Andrade da Costa Vieira

INTRODUÇÃO

- **Organização** → implementação do hardware, componentes, construção dos dispositivos → Pouco importante ao programador;
- **Arquitetura** → Tamanho das memórias e barramentos, conjunto de instruções, modos de endereçamentos → Muito importante ao programador;
 - **Ex.** O fabricante define elementos da arquitetura de uma família de processadores, cada um com uma diferente organização, que afeta seu desempenho e custo: Família Intel MCS-51

MODELO DE VON NEUMANN

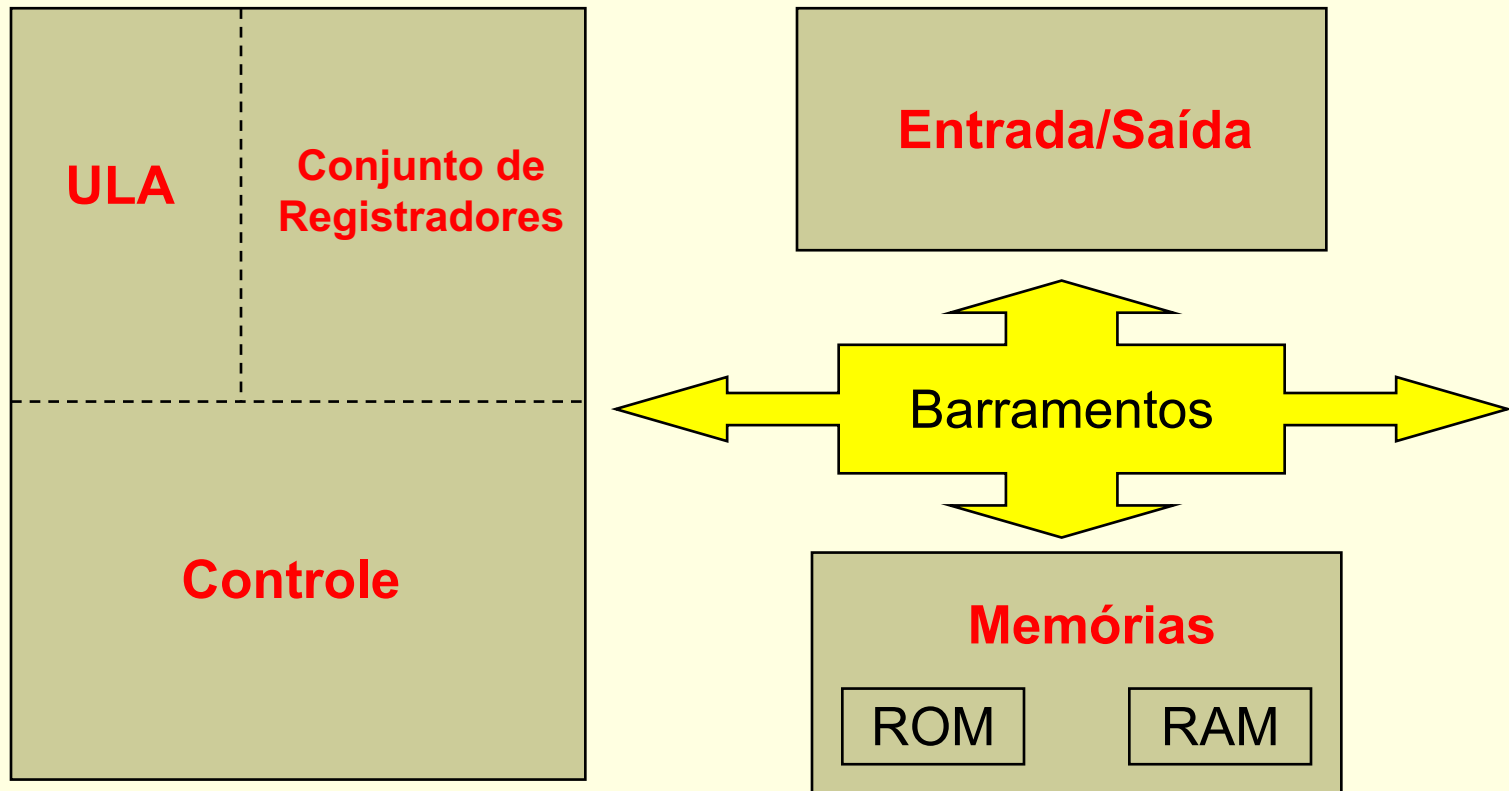
PRINCÍPIOS

A arquitetura básica de um computador consiste de 4 partes principais:

- *Unidade Central de Processamento (CPU)*
- *Memória*
- *Dispositivos de entrada/saída.*
- *Dispositivos de conexão (barramentos)*

MODELO DE VON NEUMANN

Unidade Central de Processamento (CPU)



MODELO DE VON NEUMANN

Função de cada bloco:

▪ **μP (ou CPU)**

- executa instruções lidas da memória de Programa (ROM)
- controla todo o fluxo de informação no duto de dados (gera sinais de /RD e /RW)
- Monitora os demais blocos do sistema

▪ **Memória**

Há dois tipos principais :

- **Memória ROM** - é do tipo não volátil, somente de leitura e contém o **conjunto de instruções (programa)** do sistema.
- **Memória RAM:** é do tipo volátil, de leitura e gravação, é usada para **armazenamento dos dados** gerados durante a execução do programa

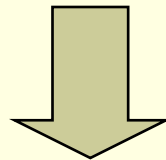
MODELO DE VON NEUMANN

Função de cada bloco:

- **Dispositivos de E/S:**
 - Permitem a comunicação do uP com o meio externo
- **Barramentos**
 - Permitem o tráfego de dados e instruções entre os diversos componentes do computador

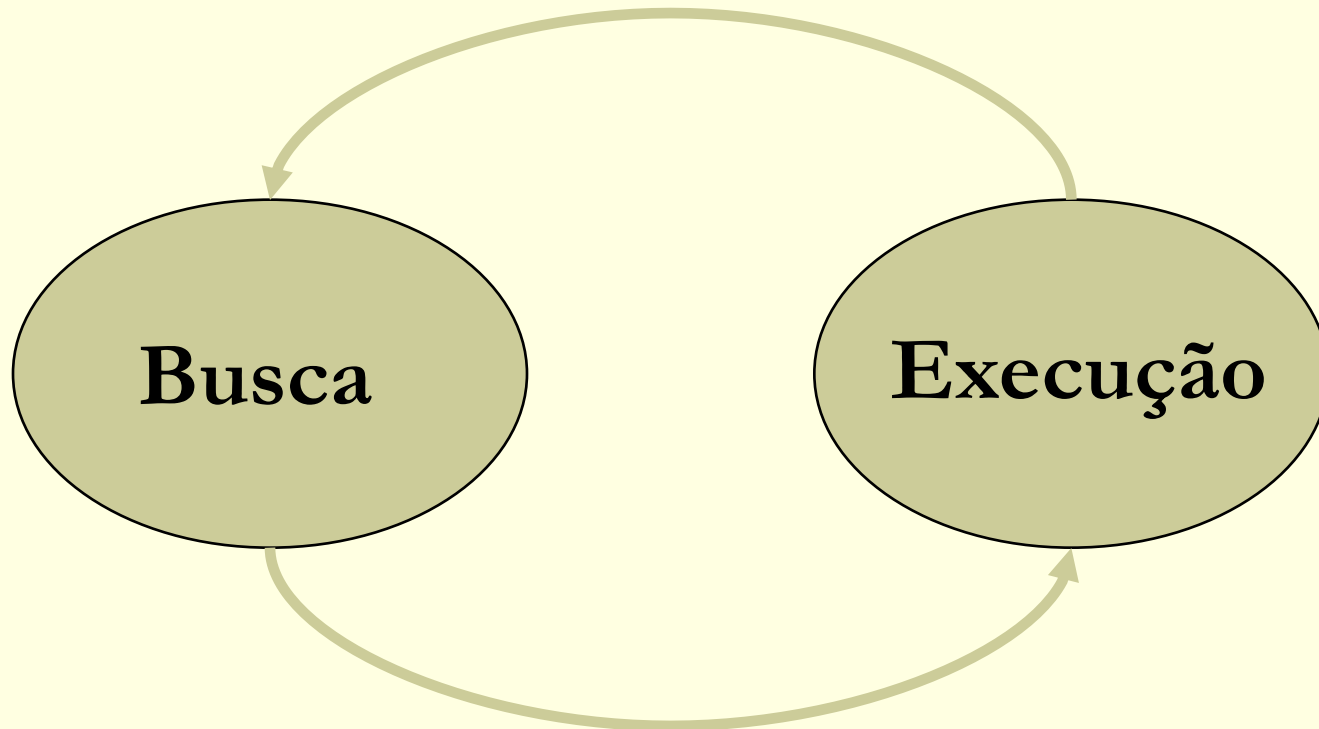
MODELO DE VON NEUMANN

“O programa que direciona as atividades da CPU é armazenado na mesma memória em que estão os dados, que devem ser manipulados pelo programa”



o computador é uma máquina de programas armazenados sequencialmente executados

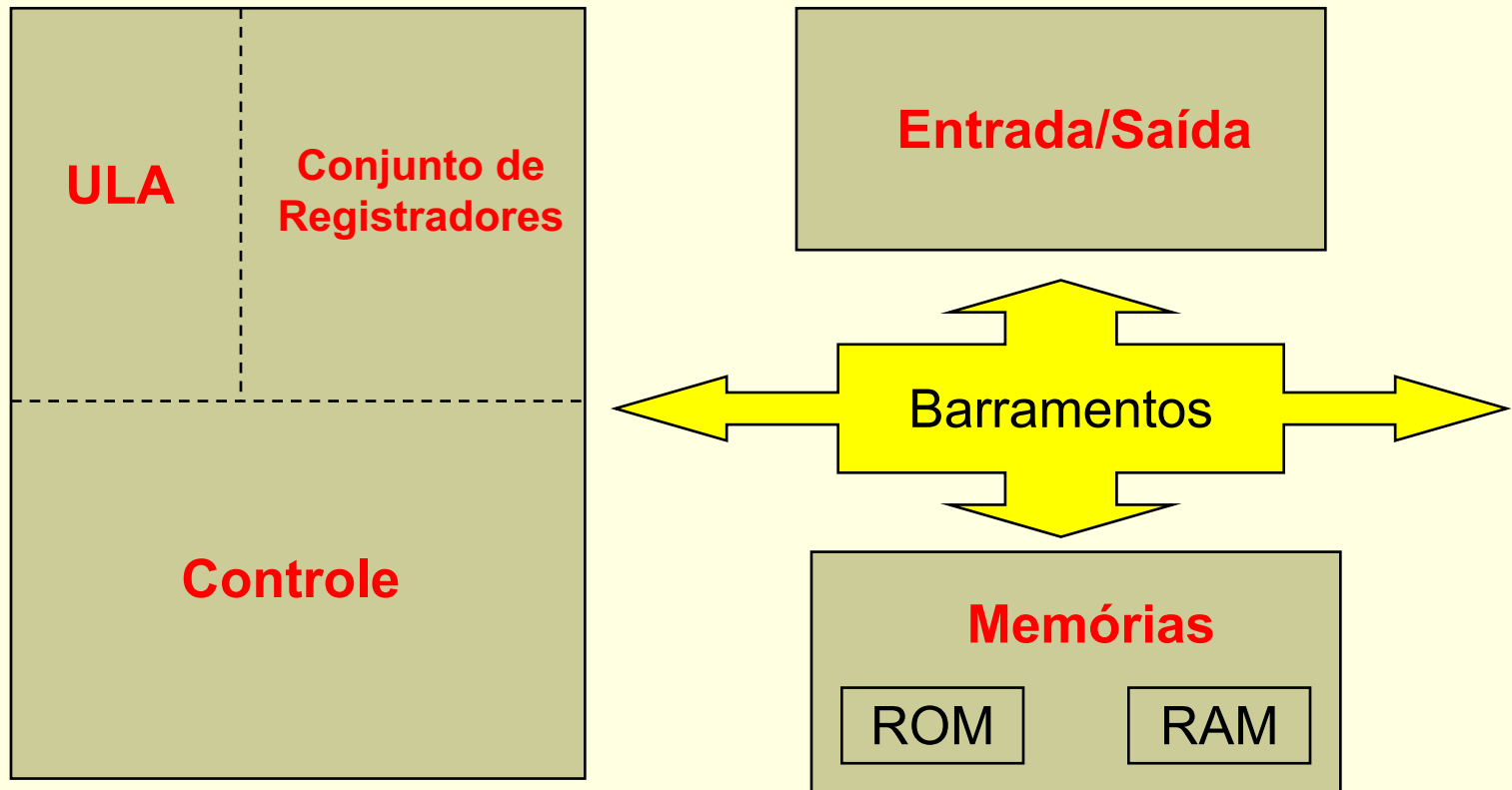
MODELO DE VON NEUMANN



Ciclo de Máquina

MODELO DE VON NEUMANN

Unidade Central de Processamento (CPU)



1. Memórias

MEMÓRIAS

- **Memória de Programa (Tipo ROM)**
 - Instruções
 - Dados não-voláteis
- **Memória de Dados (Tipo RAM)**
 - Registradores Especiais (SFR – Special Function Registers)
 - Dados temporários (GPR – General Purpose Registers)

MEMÓRIAS

Categorias de memória

- **Registrador**

- ➔ interno ao microprocessador
- ➔ Alta velocidade e baixa capacidade de armazenamento

- **Memória *cache***

- ➔ pequena porção de memória com alta velocidade de resposta
- ➔ muito próxima ou até integrada aos microprocessadores
- ➔ melhora de desempenho durante a execução de um programa.

- **Memória principal (memória de trabalho)**

- ➔ onde devem estar armazenados os dados a serem manipulados pelo processador

- **Memória secundária**

- ➔ armazena maior quantidade de dados e instruções por um período de tempo mais longo (Ex.: disco rígido, fitas magnéticas)

Hierarquia de memória

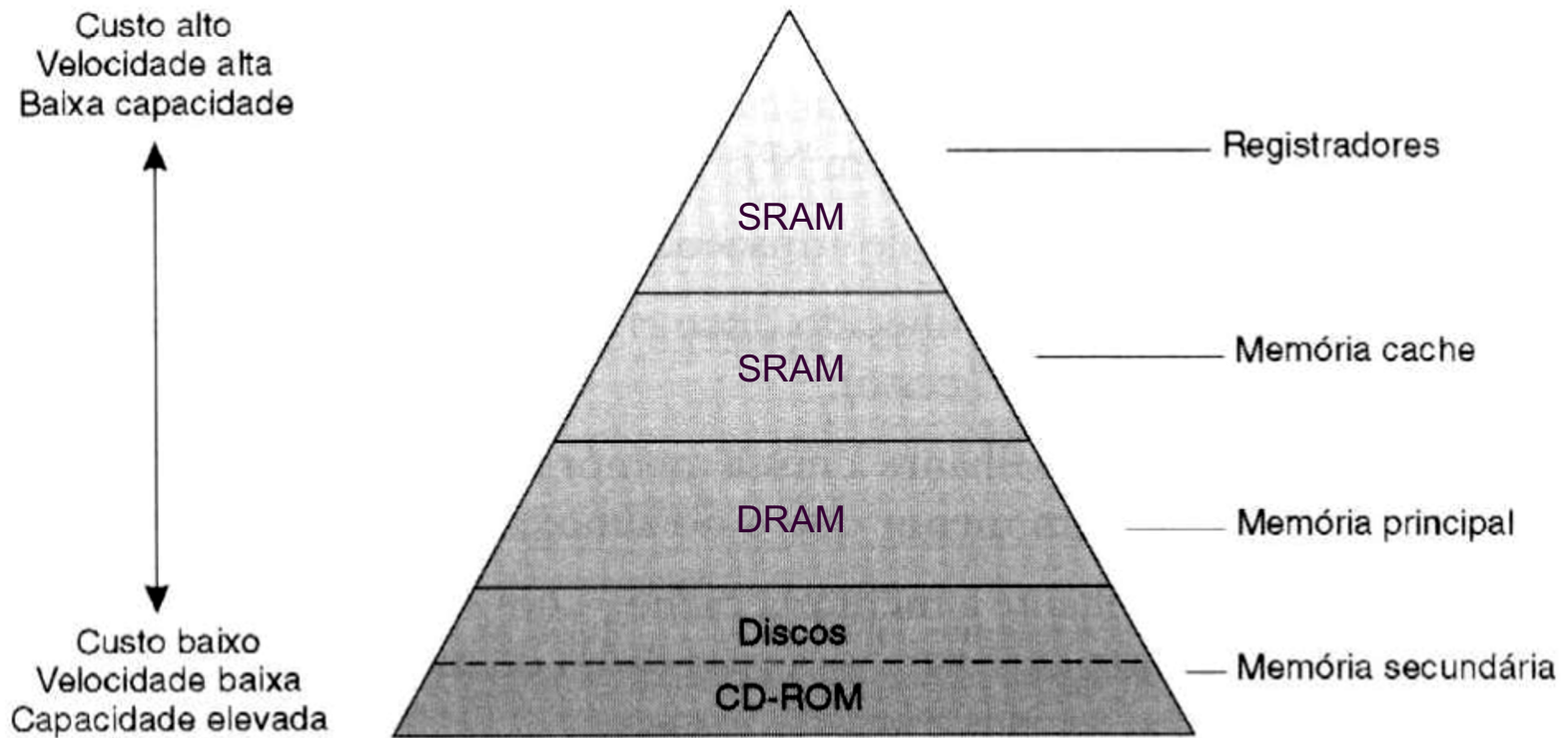
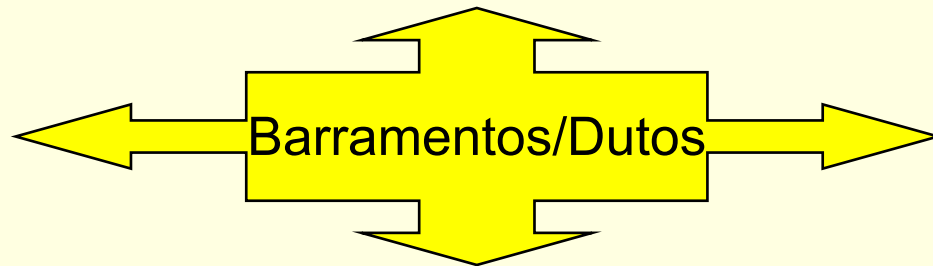


Figura 4.4 Hierarquia de memória.

2. Barramentos

BARRAMENTOS



- Canal de comunicação entre o microprocessador e os periféricos e memórias
- Todos periféricos e memória compartilham o mesmo canal de comunicação
- μ P comunica-se apenas com um por vez
- Tamanho \rightarrow determina quantos bits podem ser transmitidos por vez (ex.: barramento de 16 bits, de 32 bits...)
- Controle: temporizador interno à CPU

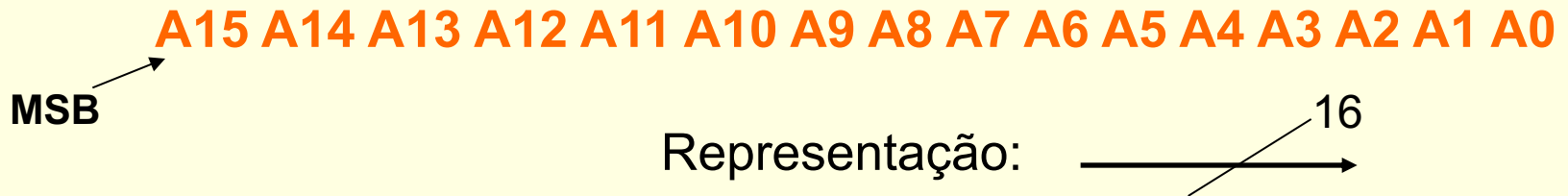
OBS: Todos os barramentos têm três partes: **barramento de dados**, **barramento de endereçamento** e **barramento de controle**

BARRAMENTOS

• Duto de Endereços

é gerado pelo microprocessador

constituído por N_e bits de endereço. Exemplo para $N_e = 16$ bits:



define a máxima capacidade de endereçamento do μP

(Espaço de Endereçamento):

$2^{16} = 64$ Kbytes, onde 1 Kbytes = 1024 bytes

destina-se ao endereçamento e seleção de memórias e dispositivos de E/S

BARRAMENTOS

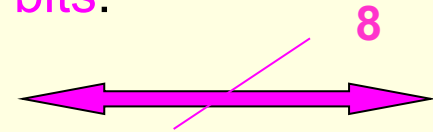
- **Duto de Dados**

É bidirecional : o microprocessador pode gravar ou ler dados de memórias ou dispositivos de I/O

Define o tamanho da palavra de memória **Nd** a ser usada

Exemplo para **Nd = 8 bits**:

D7 D6 D5 D4 D3 D2 D1 D0
MSB → (podem trafegar valores entre **00H** e **FFH**)



Tipo de Informação que trafega nesse duto:

- **Instrução** (código binário do programa – ling. de máquina)
- **Dados** (temporários)

BARRAMENTOS

• Duto de Controle

Contém sinais diversos:

- Controle de leitura e escrita
- entrada para solicitar estado de espera (aumentar duração de ciclos de leitura e escrita)
- entradas para solicitação de interrupção
- entradas para solicitação de DMA (Direct Memory Access)

Cada microprocessador pode ter parte desse conjunto de sinais no duto de controle, todos eles, ou ainda algum outro sinal específico.

3. CPU / Microprocessador

CPU / Microprocessador

- Uma CPU deve conter 3 partes principais: **ULA**, conjunto de **registradores**, unidade de **controle**;
- O primeiro dispositivo semicondutor onde foi encapsulado uma CPU completa em um único chip foi o **Intel 4004** (4 bits) em 1971. Ele continha 2.300 transístores e passou a ser chamado de **microprocessador**;



- Os microprocessadores *Intel Core i7 Quad* possuem aproximadamente 700 milhões de transístores encapsulados em um único chip;
- Os microcontroladores são dispositivos que possuem em um único chip: microprocessador, memórias, barramentos e periféricos (dispositivos de E/S);

Microprocessador X Microcontrolador

- **Microprocessador** ➔ é um dispositivo lógico programável em um único chip de silício, concebido sob a tecnologia VLSI. capacidade de executar operações lógicas, aritméticas, e de controle (CPU).
CPU + encapsulamento
- **Microcontrolador** ➔ é um circuito integrado que possui internamente um microprocessador e todos os periféricos essenciais ao seu funcionamento, como:
 - **Memória de programa** – geralmente uma memória do tipo ROM onde serão armazenadas as informações de programa,
 - **Memória de dados** – geralmente uma memória do tipo RAM, onde ficarão armazenadas as informações de dados que o programa irá utilizar.
 - **Dispositivo de seleção de entrada e saída**
 - **Temporizadores**
 - **Conversores A/D e D/A**
 - **Lógica para controle de interrupção**
 - **Comunicação serial**

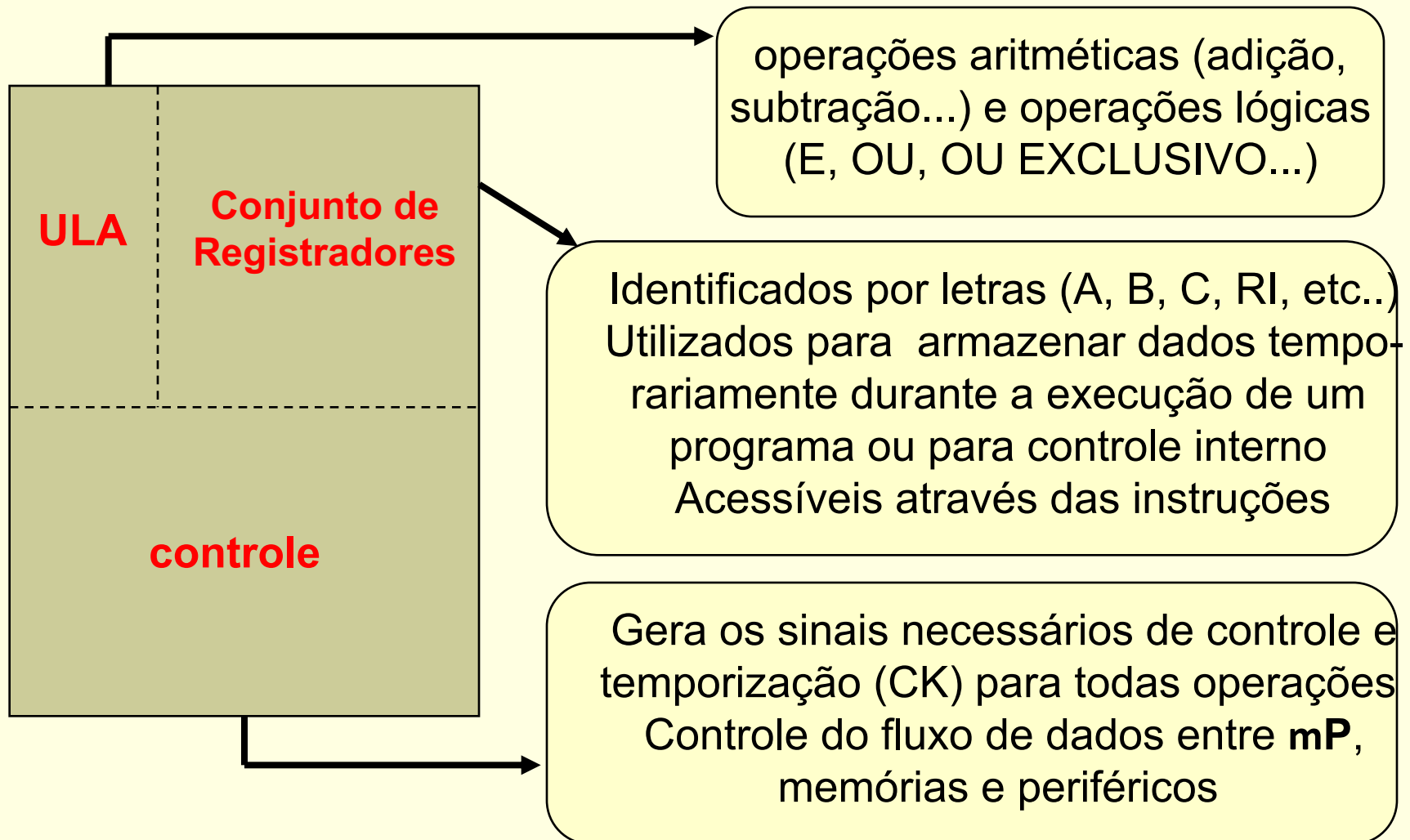
Microprocessador X Microcontrolador

- **Microprocessadores** precisam ser interligados com memória do tipo ROM e RAM, além dos dispositivos de E/S, para se tornarem operacionais
- Os **microcontroladores** são dispositivos que possuem **em um único chip**: microprocessador, memórias, barramentos, dispositivos de E/S e interfaces (para interligar periféricos);

CPU / Microprocessador

- Dispositivo de lógica programável usado para:
 - Controlar processos
 - Ligar/desligar dispositivos
- Opera com 0s e 1s, controlado por CK
- O μP executa um programa que se encontra em memória do tipo **ROM**
- Programa (**armazenado em memória**) ➔ contém conjunto de **instruções** em padrão binário ➔ **Linguagem de máquina**
- A execução é sequencial: uma única instrução por vez é executada.
- Cada μP tem seu próprio conjunto de instruções.

CPU / MICROPROCESSADOR



CPU / Microprocessador

■ ULA:

- **Operações lógicas e aritméticas:** soma, subtração, AND, OR, NAND, NOR, XOR, CMA, CMP;
- **Flags:** bits que sinalizam os resultados de operações lógicas e aritméticas.

CPU / Microprocessador

■ Clock:

- Gera sinais de sincronismo interno;
- Permite sequência ordenada de eventos;
- Um **ciclo de máquina** tem a duração de vários períodos de **CK**. (Ex. 8051 = 12 pulsos; PIC = 4 pulsos)

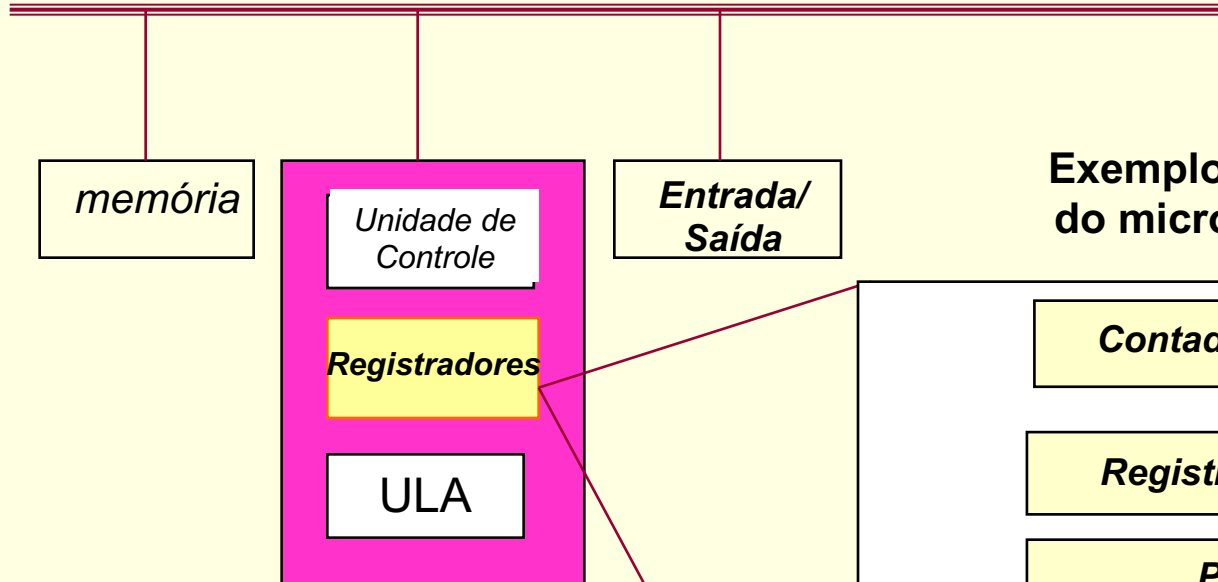
CPU / Microprocessador

■ Registadores

- Normalmente são internos à CPU, alta velocidade
- permitem o armazenamento de valores temporários, intermediários ou informações de comando
- Cada um tem uma função própria

CPU / Microprocessador

Barramento



Exemplo de registradores do microcontrolador 8051

<i>Contador de programa</i>	<i>PC</i>
<i>Registrador de Instrução</i>	<i>RI</i>
<i>Ponteiro</i>	<i>DPTR</i>
<i>Acumulador</i>	<i>A</i>
<i>Timers</i>	<i>TMR</i>
<i>Ponteiro de Pilha</i>	<i>SP</i>

nos **microprocessadores** os registradores são internos à CPU, e nos **microcontroladores** parte deles podem estar mapeados em memória RAM, dedicada a esses registradores.

Microprocessador / CPU

Diferença entre registrador e memória principal

- Registradores se localizam no interior de um microprocessador, enquanto a memória principal é externa à CPU;
- Um registrador armazena um número limitado de bits, geralmente uma palavra de memória;
- Em algumas arquiteturas, alguns “registradores” têm funções específicas, geralmente de configuração e operação do microprocessador, que são chamados de SFRs (SFR – Special Function Register);

Microprocessador / CPU

- **Ponteiros**: registradores que contêm informação de endereço
 - Ponteiro de Programa : **PC**
 - Ponteiro de Dados : **DPTR** (microcontrolador 8051)
 - Ponteiro de Pilha: **SP**
- **Pilha**: Área de Memória **RAM** para armazenamento de **endereço de retorno de subrotina ou interrupção**. Há também instruções que permitem o seu uso pelo programador
- O ponteiro **SP** indica qual a última posição em que foi armazenado um dado na pilha

Microprocessador / CPU

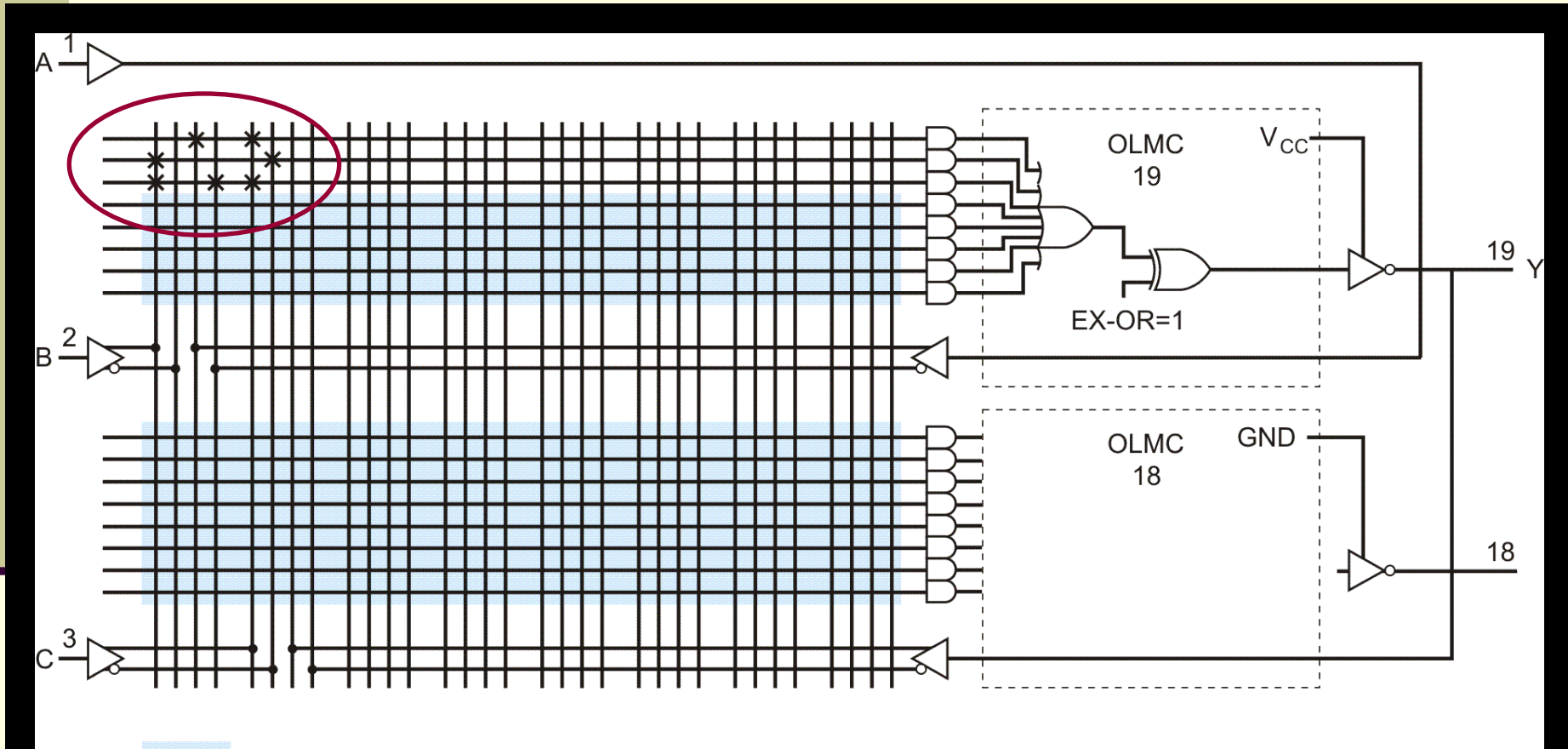
■ UC: Unidade de Controle

- Lê o opcode, que foi armazenado no **IR** (registrador de instruções);
- Elemento que garante a correta execução dos programas e a utilização dos dados corretos nas operações;
- Decodifica a instrução correspondente e gera os sinais para o processamento da mesma;
- Controla o acesso aos barramentos;
- Controla a execução de todas as operações no μP .

PLD x Microprocessador

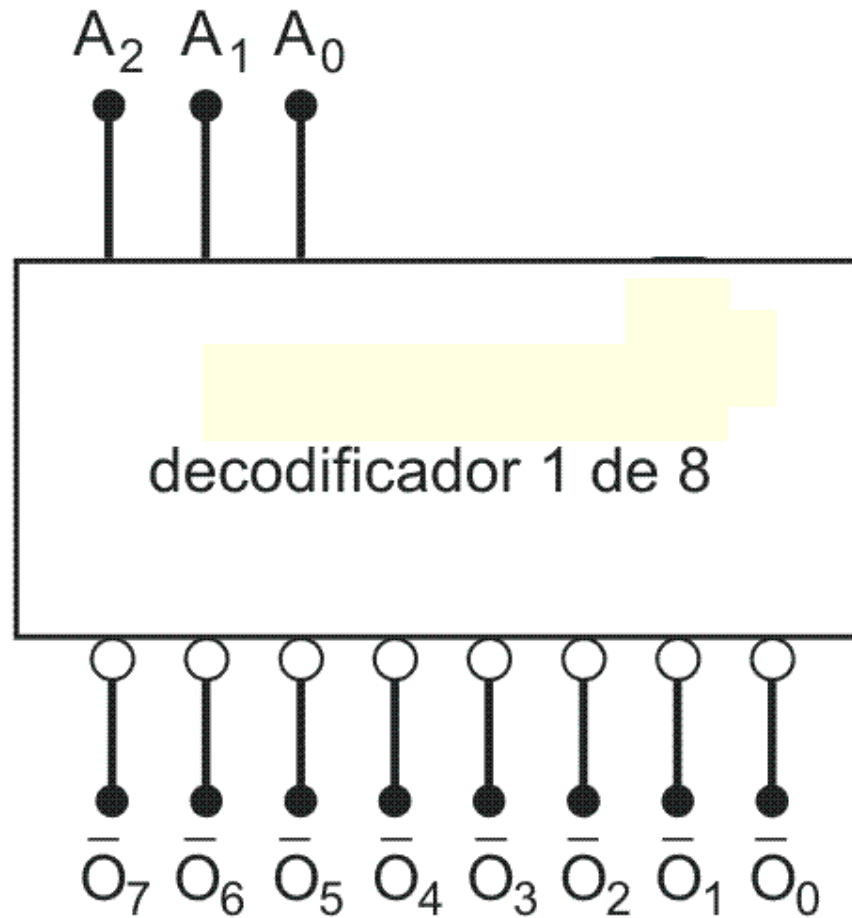
VHDL x Assembly

PLD - GAL16V8



$$Y = AC + B\bar{C} + \bar{A}BC$$

PLD – Decodificador 3 x 8



```
/*74LS138 1 of 8 decoder functional equivalent circuit
```

```
*/
```

```
/*      Inputs      */
```

```
pin 1 =  Azero  ;
```

```
pin 2 =  Aone   ;
```

```
pin 3 =  Atwo   ;
```

```
/*      Outputs     */
```

```
pin [19..12] = ![S7..0] ;
```

```
/*      SET DEFINITIONS      */
```

```
field inputs = [Atwo, Aone, Azero];
```

```
field outputs = [S7..0]
```

```
/*      Hardware Description      */
```

table	inputs	=>	outputs	
{	[0,0,0]	=>	1;	/* output 0 active */
	[0,0,1]	=>	2;	/* output 1 active */
	[0,1,0]	=>	4;	/* output 2 active */
	[0,1,1]	=>	8;	/* output 3 active */
	[1,0,0]	=>	10;	/* output 4 active */
	[1,0,1]	=>	20;	/* output 5 active */
	[1,1,0]	=>	40;	/* output 6 active */
	[1,1,1]	=>	80;}	/* output 7 active */

Decodificador 3x8 no 8051

Porta P0 como entrada (0-2)

Porta P1 como saída (0-7)

```
LOOP:
MOV A,P0
ANL A, #00000111B
XRL A, #00000000B
JZ ZERO
XRL A, #00000001B
JZ UM
XRL A, #00000010B
JZ DOIS
XRL A, #00000011B
JZ TRES
XRL A, #00000100B
JZ QUATRO
XRL A, #00000101B
JZ CINCO
XRL A, #00000110B
JZ SEIS
XRL A, #00000111B
JZ SETE
SJMP LOOP
```

```
ZERO:
MOV P1, #00000001B
SJMP LOOP
UM:
MOV P1, #00000010B
SJMP LOOP
DOIS:
MOV P1, #00000100B
SJMP LOOP
TRES:
MOV P1, #00001000B
SJMP LOOP
QUATRO:
MOV P1, #00010000B
SJMP LOOP
CINCO:
MOV P1, #00100000B
SJMP LOOP
SEIS:
MOV P1, #01000000B
SJMP LOOP
SETE:
MOV P1, #10000000B
SJMP LOOP
```

FIM