

# Arquitetura de Computadores

## Aula 10 - Processadores

**Prof. Dr. Eng. Fred Sauer**

<http://www.fredsauer.com.br>

[fsauer@gmail.com](mailto:fsauer@gmail.com)

## PROCESSADORES

# TÓPICOS

- **ORGANIZAÇÃO**
- **COMPONENTES BÁSICOS**
- **INSTRUÇÃO DE MÁQUINA**
- **CICLO DE INSTRUÇÃO**
- **EXECUÇÃO DE PROGRAMAS**

## PROCESSADORES

**Todo processador pode ser analisado como sendo organizado em duas áreas funcionais:**

**Área de controle**

**Área de execução (ou processamento)**

**Cada uma possuindo componentes específicos.**

## PROCESSADORES

### Função de um Processador

Realizar, etapa por etapa, a execução do ciclo de uma instrução, permitindo que elas sejam executadas de forma automática. O ciclo pode ser compreendido com atividades em duas categorias: as cerebrais (de interpretação e emissão dos sinais de controle apropriados para execução) e as de execução de uma operação propriamente dita.



## PROCESSADORES

### ARQUITETURA DE UM PROCESSADOR

A Arquitetura de qualquer Processador é projetada para executar Instruções de Máquina.

Na fábrica, os engenheiros projetistas definem quais operações serão realizadas diretamente pelo Hardware. As operações são implementadas por sequências binárias denominadas

#### INSTRUÇÕES DA MÁQUINA.

10010101011101010101010110

## PROCESSADORES

# ARQUITETURA DE UM PROCESSADOR

Todo o conjunto de dispositivos existente em um processador (decodificadores, unidades de cálculo, registradores, etc) servem para executar as instruções. Na realidade, servem para realizar as etapas de um ciclo de instrução.

Principalmente o conjunto de instruções e os registradores de dados formam a **ARQUITETURA DO PROCESSADOR** (podem servir a mais de um modelo de processador).

## PROCESSADORES

# EXEMPLOS DE ARQUITETURAS DE PROCESSADORES

**Arquitetura X86 e IA 64 (da INTEL)**

**Arquitetura ARM (da ARM – tipo RISC)**

**Arquitetura POWER (da IBM, usada nos processadores CELL).**

## PROCESSADORES

### Instrução de Máquina

Conjunto de bits que identifica uma determinada operação primitiva a ser realizada diretamente pelo HW.

Usualmente, uma Instrução de Máquina deve conter um grupo de bits que identifica a operação (denomina-se **CÓDIGO DA OPERAÇÃO**) e outro grupo de bits que identifica a localização dos dados que serão manipulados na referida operação. (denomina-se **OPERANDO**)

**110101** 101010101010101010101010

Cód.  
operação

Operando (s)

## PROCESSADORES

### OPERAÇÕES PRIMITIVAS - EXEMPLOS

- Operações aritméticas – somar, subtrair, multiplicar e dividir
- Operações lógicas – AND, OR, XOR
- Operações de entrada e saída de dados
- Operações de desvio de controle
- Operações de movimentação de dados

## PROCESSADORES

### INSTRUÇÃO DE MÁQUINA – EXEMPLOS

- \* Transferir uma palavra de dados de uma célula para outra.
- \* Efetuar a soma entre dois operandos, guardando o resultado em um deles ou em um terceiro operando.
- \* Desviar incondicionalmente para outro endereço fora da sequência.
- \* Testar uma condição. Se teste verdadeiro, então desviar para outro endereço fora da sequência.
- \* Realizar uma operação lógica AND entre dois valores.
- \* Parar a execução de um programa.
- \* Adicionar 1 ao valor de um operando.
- \* Transferir um byte de dados de uma porta de E/S para a MP.
- \* Transferir um byte de dados da MP para uma porta de E/S.
- \* Substituir o operando por seu valor absoluto.

### PARTES DE UMA INSTRUÇÃO DE MÁQUINA



**Código de Operação – C.Op.** – campo da instrução que indica qual a operação a ser realizada e sua decodificação inicia a execução da referida operação.

**Operando(s) – Op** – campo da instrução, cujos bits representam o(s) dado(s) que será(ão) manipulado(s) na operação. A instrução pode conter um ou mais campos operando, bem como pode representar o endereço do dado (caso de uma variável) ou o próprio valor do dado (caso de uma constante do programa).

### INSTRUÇÃO DE MÁQUINA

-Formato Básico: 2 partes:

- (C.Op) - Código de Operação
- (Op) - Operando (s). Pode ter 1, 2 ou 3.

C. Op	Operando		ADD A	
C. Op	Oper 1	Oper 2	ADD A, B	
C. Op	Oper 1	Oper 2	Oper 3	ADD A, B, C

### EXEMPLOS DE INSTRUÇÕES DE MÁQUINA

**ADD Op1, Op2** – soma o valor cujo endereço é Op1 com o valor cujo end. é Op2 e armazena o resultado no endereço indicado por Op1.

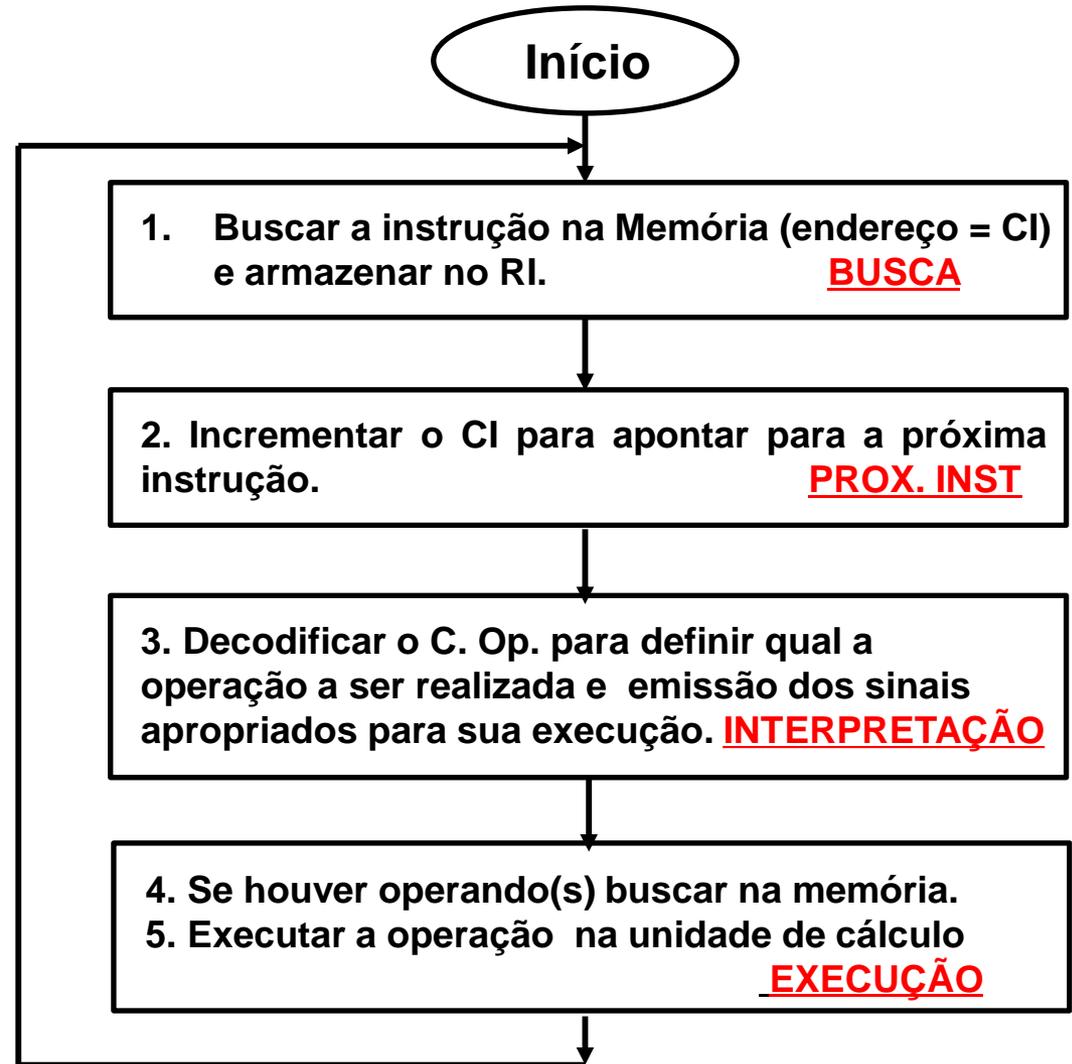
Os valores Op1 e Op2 podem ser endereço de memória ou endereço de registradores de dados. Nesse caso, a sintaxe da instrução costuma ser: **ADD R1, R2**

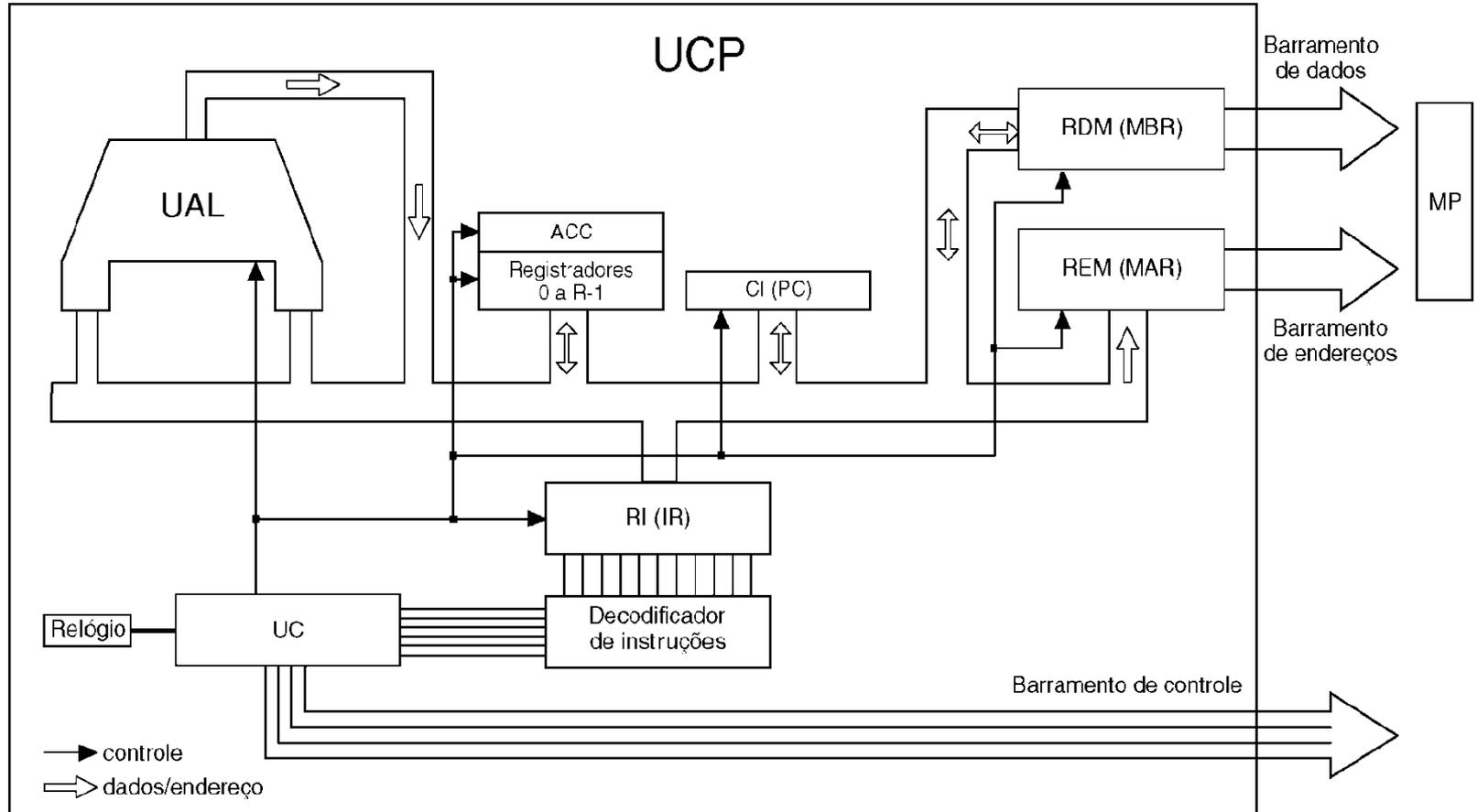
**ADD Nota A, Nota B**

<b>ADD</b>	<b>Nota A</b>	<b>Nota B</b>
<b>C.Op</b>	<b>Op. 1</b>	<b>Op. 2</b>

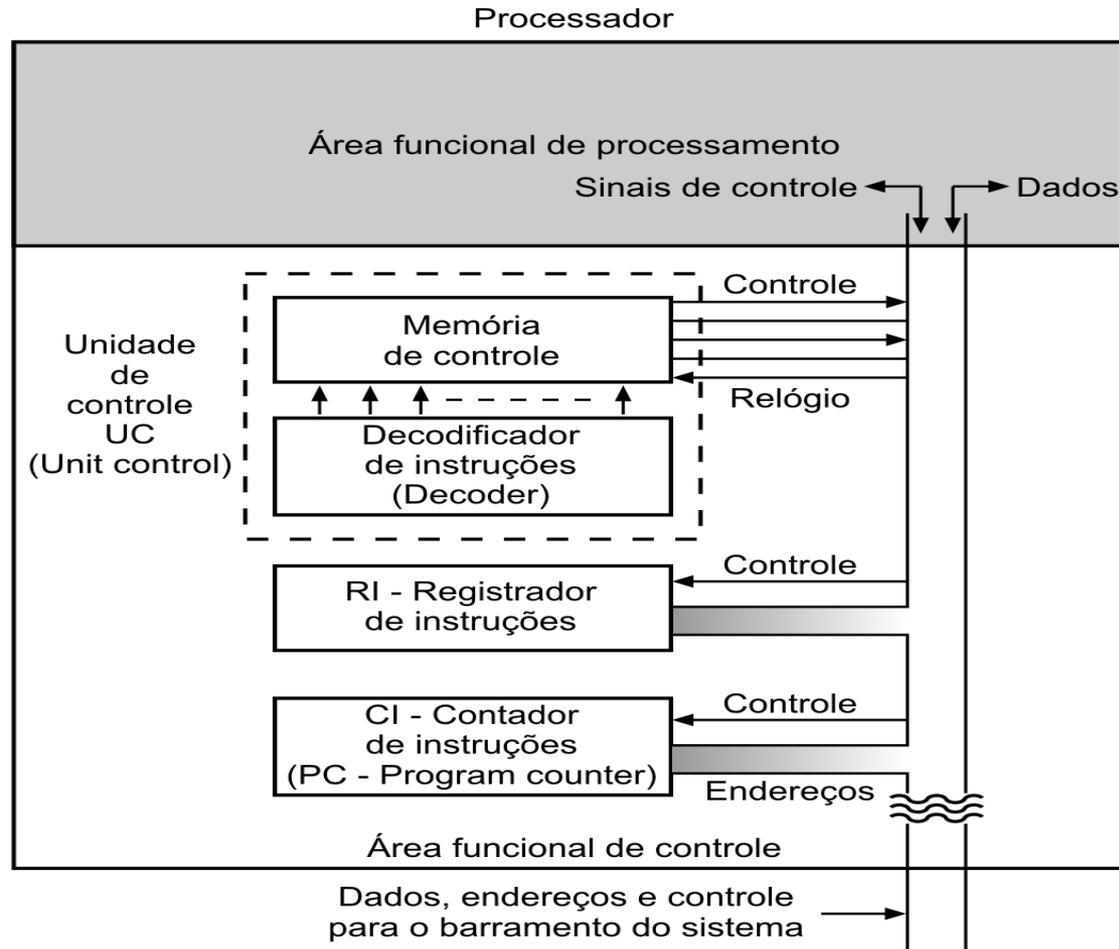
**MOV Op1, Op2** – significa: mover o valor do endereço Op2 para Op1.

**CICLO DE INSTRUÇÃO**





## Componentes da área de controle



### COMPONENTES PARA A ÁREA DE CONTROLE

**Registrador CI (etapa 1 do Ciclo de Instrução)**

**CI (PC ou *program counter*)** – armazena o endereço da próxima instrução

**Registrador RI (etapa 1 do Ciclo de Instrução)**

**RI (IR ou *instruction register*)** – armazena a instrução a ser executada (ou seu C. Op.)

**Decodificador de Instrução (etapa 3 do Ciclo de Instrução)**

atende à etapa 3 do Ciclo de Instrução

Possui  $n$  bits de entrada ( $n =$  largura do C.Op) e  $2^n$  fios de saída, sendo UMA única saída verdadeira (bit 1) para cada entrada.

## Decodificador com 4 entradas e 16 saídas



(a) Diagrama em bloco da decodificação em uma UCP

E <sub>0</sub>	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

(b) Linhas de Saída

### COMPONENTES PARA A ÁREA DE CONTROLE

**Unidade de Controle – UC** – contém a programação para execução do ciclo básico de qualquer instrução, bem como as sequências de microoperações para execução de cada instrução existente.

**Relógio** - dispositivo que produz uma sequência de pulsos em uma frequência determinada, os quais servem para sincronizar e cadenciar as diversas microoperações do processador.

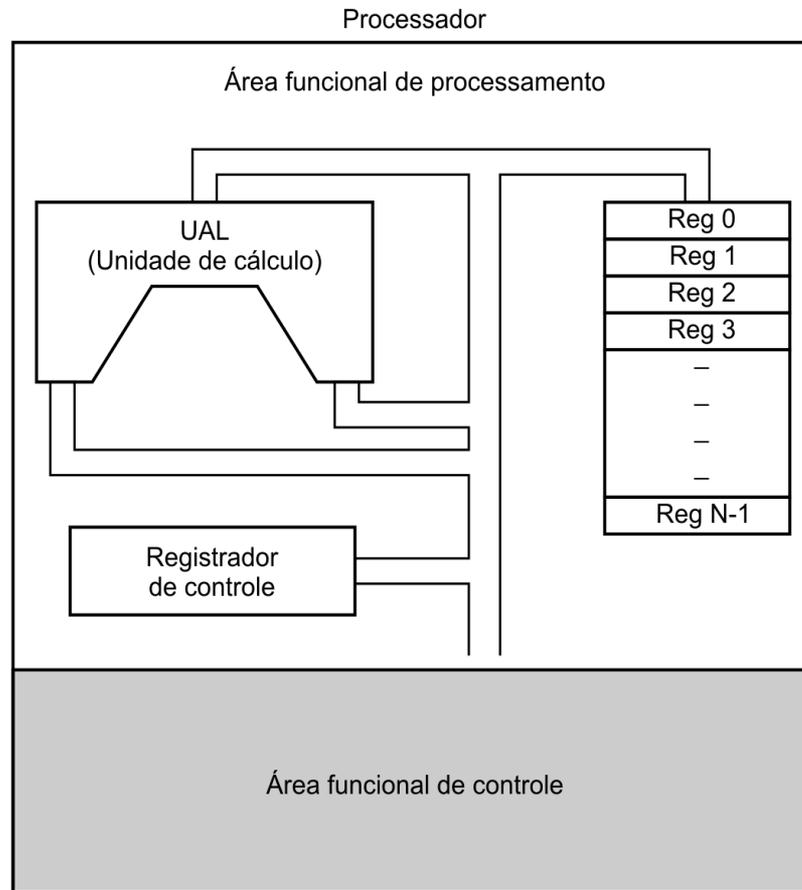
Há um único relógio físico no sistema, que passa por uma distribuidor (multiplica ou divide a quantidade de pulsos) e os envia, na sequência requerida para: processador (maior velocidade); barramentos (menor).

**Velocidade** = 1MHz =  $1 * 10^6$  pulsos.

Intervalo entre pulsos (ciclo do relógio) =  $1s / 1 * 10^6 = 1$  microseg ou  $10^{-6}$  seg

# Componentes da área de processamento

## PROCESSADORES



### PROCESSADORES

**Unidades de cálculo (ULA)** – unidade lógica e aritmética – realiza as operações aritméticas e lógicas existentes no conjunto de instruções do processador.

**Unidade de Ponto Flutuante – FP** –Executa operações aritméticas com números fracionários e números muito grandes ou muito pequenos.

**Registradores de Dados** - armazenam os dados a serem manipulados pelas unidades de cálculo.

**Registrador de Controle** – armazena elementos de ajuda na interpretação de ocorrências em operações (bit de overflow, bit de underflow, bit de “vai 1” em operações de soma, bit indicador de resultado zero, etc)