



# Arquitetura e Organização de Computadores

## Aula 9

Centro Universitário 7  
Setembro - Uni7  
**Sistemas de Informação**

Prof. MSc Manoel Ribeiro

[manoel@opencare.com.br](mailto:manoel@opencare.com.br)

# Barramentos

$$\Delta x = v t$$
$$\Delta x = v_0 t + \frac{a t^2}{2}$$
$$v = v_0 + a t$$
$$v^2 = v_0^2 + 2 a \Delta x$$

$$\nabla \cdot \vec{E} = \frac{1}{\epsilon_0} \rho$$
$$\nabla \cdot \vec{B} = 0$$
$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$

$$\nabla \times \vec{B} = \mu_0 \vec{J} + \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t}$$



$$v_m = \frac{v + v_0}{2}$$

$$h = \frac{v^2 - v_0^2}{2g}$$

$$r = r_x \hat{i} + r_y \hat{j}$$
$$\vec{r} = (r_x, r_y)$$
$$z = \int_1^x \int_1^y \int_1^z dx dy dz$$
$$\vec{g} = g_x \hat{i} + g_y \hat{j} + g_z \hat{k}$$
$$r = \sqrt{r_x^2 + r_y^2}$$
$$r_{gx} = \frac{r_x}{r}$$

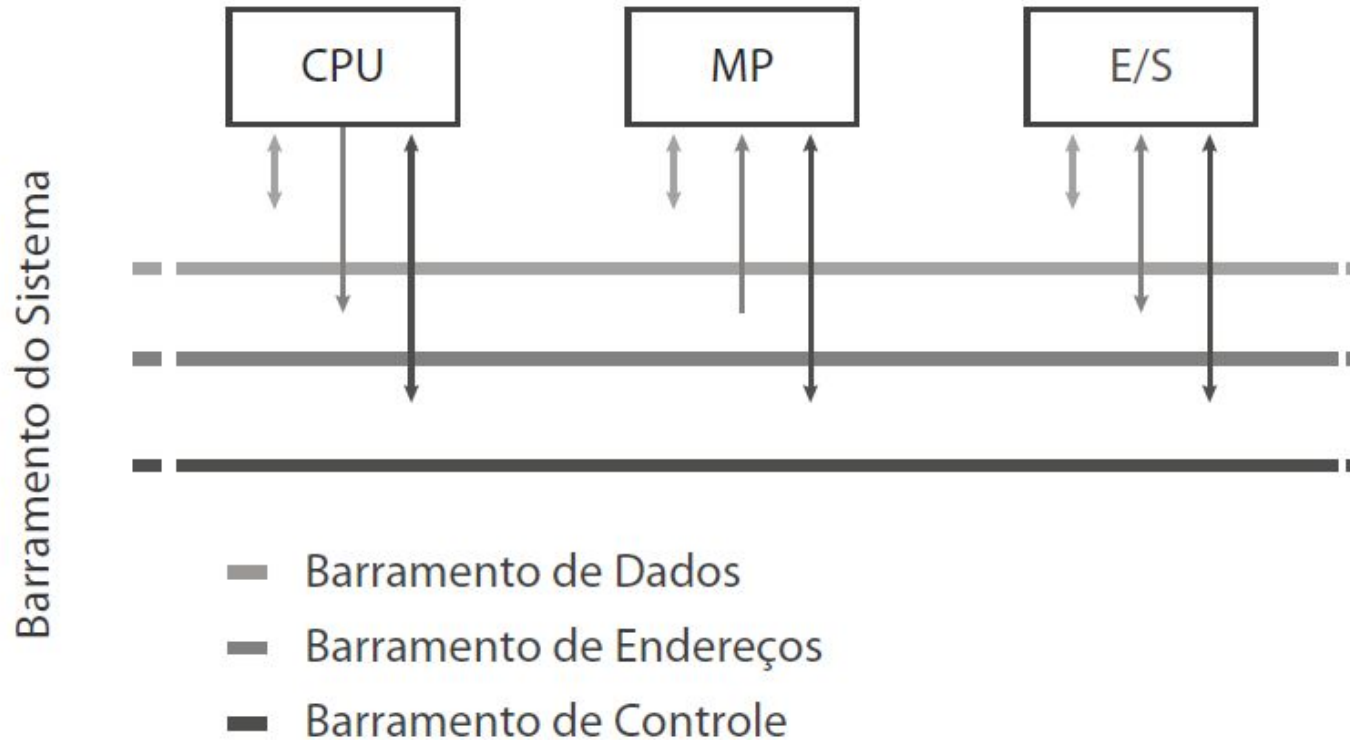


$$v = \frac{\Delta s}{\Delta t} = \frac{s - s_0}{t}$$

# Barramentos

- O barramento conduz de modo sincronizado o fluxo de informações (dados e instruções, endereços e controles) de um componente para outro ao longo da placa-mãe
- O barramento organiza o tráfego de informações observando as necessidades de recursos e as limitações de tempo de cada componente, de forma que não ocorram colisões

# Barramentos



# Barramento de Dados

- Este barramento interliga o RDM (localizado na UCP) à memória principal, para transferência de instruções ou dados a serem executados.
- É bidirecional
- Possui influência direta no desempenho do sistema

# Barramento de Endereço

- Interliga o REM (localizado na UCP) à memória principal, para transferência dos bits que representam um determinado endereço de memória onde se localiza uma instrução ou dado a ser executado
- É unidirecional
- O tamanho do barramento de endereços determina a quantidade máxima de armazenamento de dados que a memória principal pode dispor

# Barramento de Controle

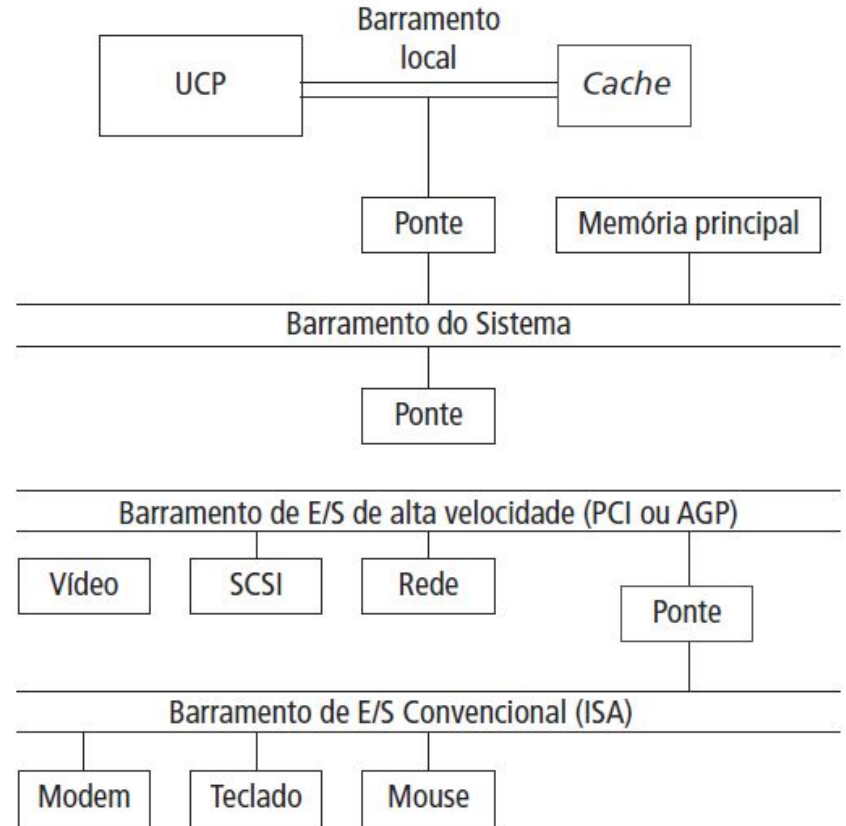
- Interliga a UCP, mais especificamente a Unidade de Controle (UC), aos demais componentes do sistema computacional (memória principal, componentes de entrada e de saída) para passagem de sinais de controle gerados pelo sistema
- É Bidirecional
  - Controle: UCP sinaliza operação para outro componente
  - Interrupção: dispositivo sinaliza algo a UCP

# Barramento de Controle

- Subdivide-se em:
  - Local, liga os componentes do processador
  - Sistema, liga os componentes da placa-mãe
  - Externo, liga componentes conectados externamente à placa-mãe, como placa de expansão ou portas de entrada e saída

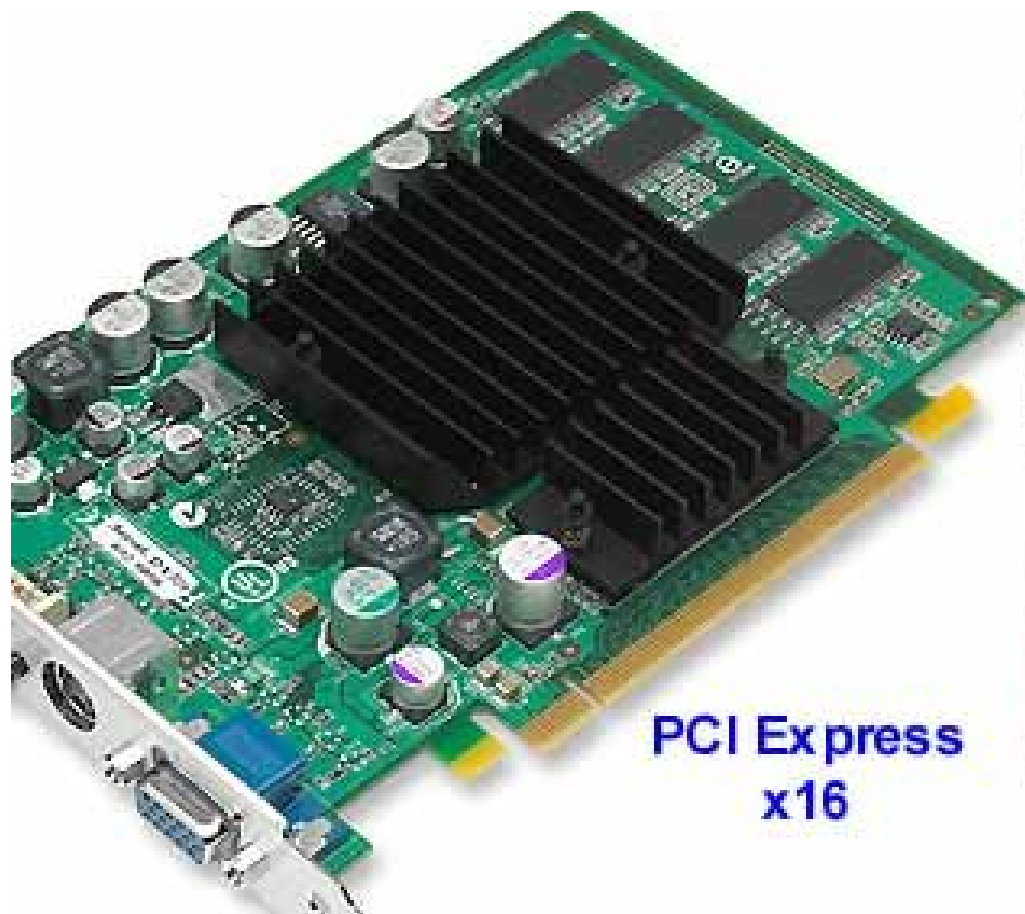


# Barramento de Controle



# Barramentos externos

- Modelos:
  - ISA (Industry Standard Adapter)
  - PCI (Peripheral Component Interconnect)
  - USB (Universal Serial Bus)
  - AGP (Accelerated Graphics Port)
  - PCI Express (Peripheral Component Interconnect Express)



PCI Express  
x16



AGP

# Instruções de máquina

- instrução de máquina é a codificação de uma operação básica que o hardware pode executar diretamente
- Para exemplificar, vamos considerar um processador com uma ULA capaz de executar a soma ou a multiplicação de dois números (operações básicas), mas não as duas coisas ao mesmo tempo. Agora imaginem que esse processador precisa executar:  $X = A + B * C$  de uma só vez. Isso irá gerar a necessidade de transformar essa instrução, considerada complexa para esse processador, em uma sequência de instruções mais básicas, da seguinte forma:
  - Executar primeiro:  $T = B * C$  (sendo que  $T$  é um registrador)
  - Em seguida realizar a operação:  $X = A + T$

# Instruções de máquina

- Um processador precisa dispor de um conjunto de instruções básicas para: movimentação de dados; aritméticas; lógicas; edição; deslocamento; manipulação de registros de índice; desvio; modificação de memória; formais de ligação à sub-rotina; manipulação de pilha; entrada e saída e de controle.

# Instruções de máquina

- Atualmente, há duas tecnologias de projeto de processadores empregadas pelos fabricantes de computadores (MONTEIRO, 2007):
  - Sistemas com conjunto de instruções complexo (Complex Instruction Set Computers – CISC)
    - INTEL, AMD,
  - Sistemas com conjunto de instruções reduzido (Reduced Instruction Set Computers – RISC).
    - ARM, DEC Alpha, SPARC, MIPS, e PowerPC

# RISC x CISC

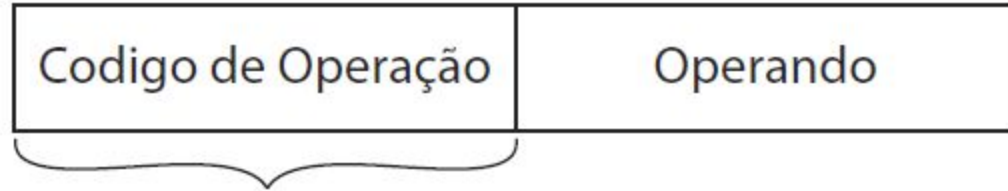
RISC	CISC
Múltiplos conjuntos de registradores, muitas vezes superando 256	Único conjunto de registradores, tipicamente entre 6 e 16 registradores
Três operandos de registradores permitidos por instrução (por ex., add R1, R2, R3)	Um ou dois operandos de registradores permitidos por instrução (por ex., add R1, R2)
Passagem eficiente de parâmetros por registradores no chip (processador)	Passagem de parâmetros ineficiente através da memória
Instruções de um único ciclo (ex. load e store)	Instruções de múltiplos ciclos
Controle hardwired (embutido no hardware)	Controle microprogramado
Altamente paralelizado (pipelined)	Fracamente paralelizado
Instruções simples e em número reduzido	Muitas instruções complexas
Instruções de tamanho fixo	Instruções de tamanho variável
Complexidade no compilador	Complexidade no código
Apenas instruções load e store podem acessar a memória	Muitas instruções podem acessar a memória
Poucos modos de endereçamento	Muitos modos de endereçamento

# Formato das instruções

- Código de operação (Opcode)
  - Possui um subgrupo de bits que identifica a operação a ser realizada pelo processador.
  - É o campo da instrução cujo valor binário identifica a operação a ser realizada
  - Esse valor é a entrada no Decodificador de Instruções na Unidade de Controle. Cada instrução deverá ter um código único que a identifique.



# Opcode



ADD  
01011

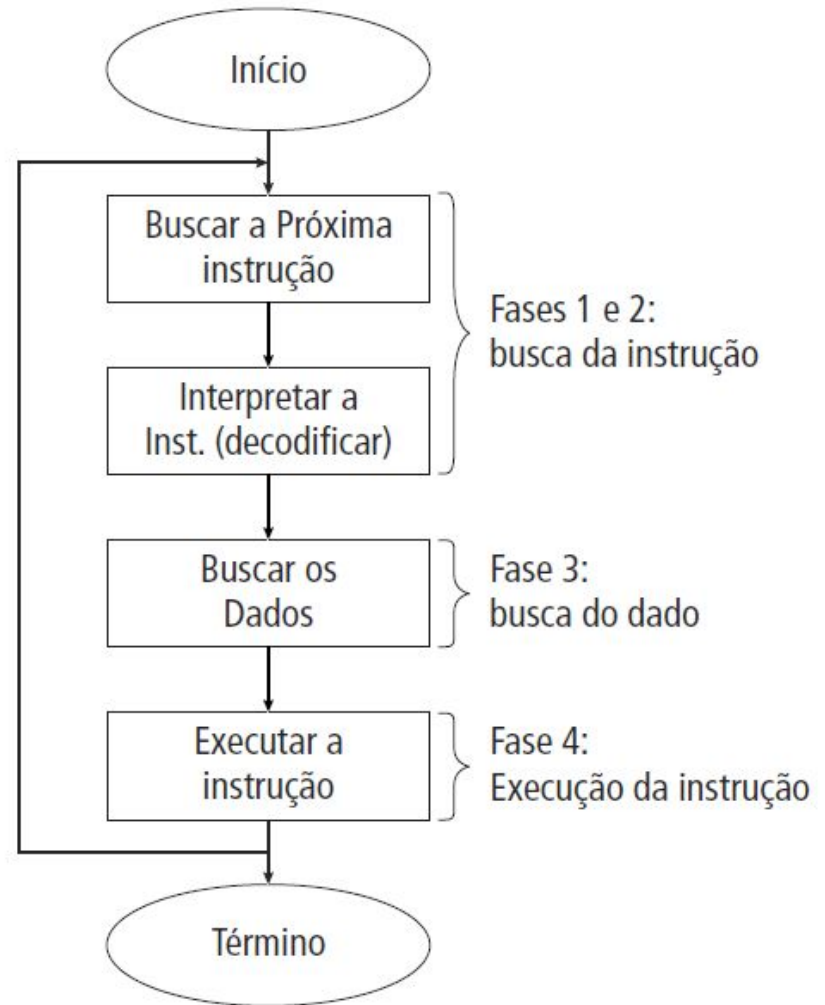


Decodificador

# Ciclo de instrução

- A partir da proposta da arquitetura de Von Neumann, da qual os conceitos básicos ainda são válidos, propunha-se que as instruções fossem executadas sequencialmente (a não ser pela ocorrência de um desvio), uma a uma.
- O contador de instruções indica a sequência de execução, isto é, o CI controla o fluxo de execução das instruções.

# Ciclo de instrução



# Ciclo de instrução - Fase 1

- Micro-operações:

- a UC lê o conteúdo do CI (endereço da próxima instrução) e coloca o endereço no REM
- a UC envia um sinal – via barramento de controle – à controladora da memória principal para que realize uma operação de leitura
- a memória principal lê o endereço que está no REM – via barramento de endereço – e busca o conteúdo da célula referenciada;
- a memória principal coloca no RDM – via barramento de dados – o conteúdo da célula;
- a controladora da memória principal envia à UC – via barramento de controle – o sinal de leitura concluída;
- a UC transfere o código de operação (conteúdo que está no RDM) ao RI.

## Ciclo de instrução - Fase 2

- Micro-operações:
  - o RI envia para o decodificador de instrução os bits correspondentes ao Opcode;
  - o Decodificador de Instruções determina quantas células a instrução ocupa e identifica a operação a ser realizada
  - a UC envia um sinal de controle à ULA informando a operação a ser realizada e incrementa o CI para apontar para a próxima instrução:  $CI \rightarrow (CI+N)$ , onde,  $N = n^{\circ}$  de células que a próxima instrução ocupa.

# Ciclo de instrução - Fase 3

- Micro-operações:

- a UC envia um sinal – via barramento de controle – à controladora da memória principal para que realize uma operação de leitura
- a memória principal lê o endereço que está no REM – via barramento de endereços – e busca o conteúdo da célula referenciada
- a memória coloca no RDM – via barramento de dados – o conteúdo da célula lida;
- a memória principal envia à UC – via barramento de controle – um sinal de leitura concluída;
- a UC transfere o operando (conteúdo do RDM) ao RI (se for um código de operação) ou a um dos registradores internos da UCP

# Ciclo de instrução - Fase 4

- Micro-operações:

- a ULA executa a instrução sobre os dados disponíveis nos registradores;
- ao concluir a operação, a ULA envia um sinal para a UC informando que a execução terminou;
- a UC identifica o endereço de memória para onde deve ser enviado o resultado da operação e o armazena no REM
- a UC autoriza o envio do resultado da operação para o RDM
- a UC autoriza a controladora de memória a realizar uma operação de leitura no REM para obter o endereço de memória onde deverá ser escrito o resultado e uma leitura no RDM para obter o resultado a ser escrito na memória.

# Resumo

- Barramento podem ser de dados, endereço ou controle
- Os barramentos de controle podem ser local, sistema e externo
- Os barramentos externos AGP e PCI Express são considerados os mais rápidos, portanto os mais utilizados atualmente
- O código da operação é chamado Opcode, este é decodificado e executado em 4 fases.



Fim