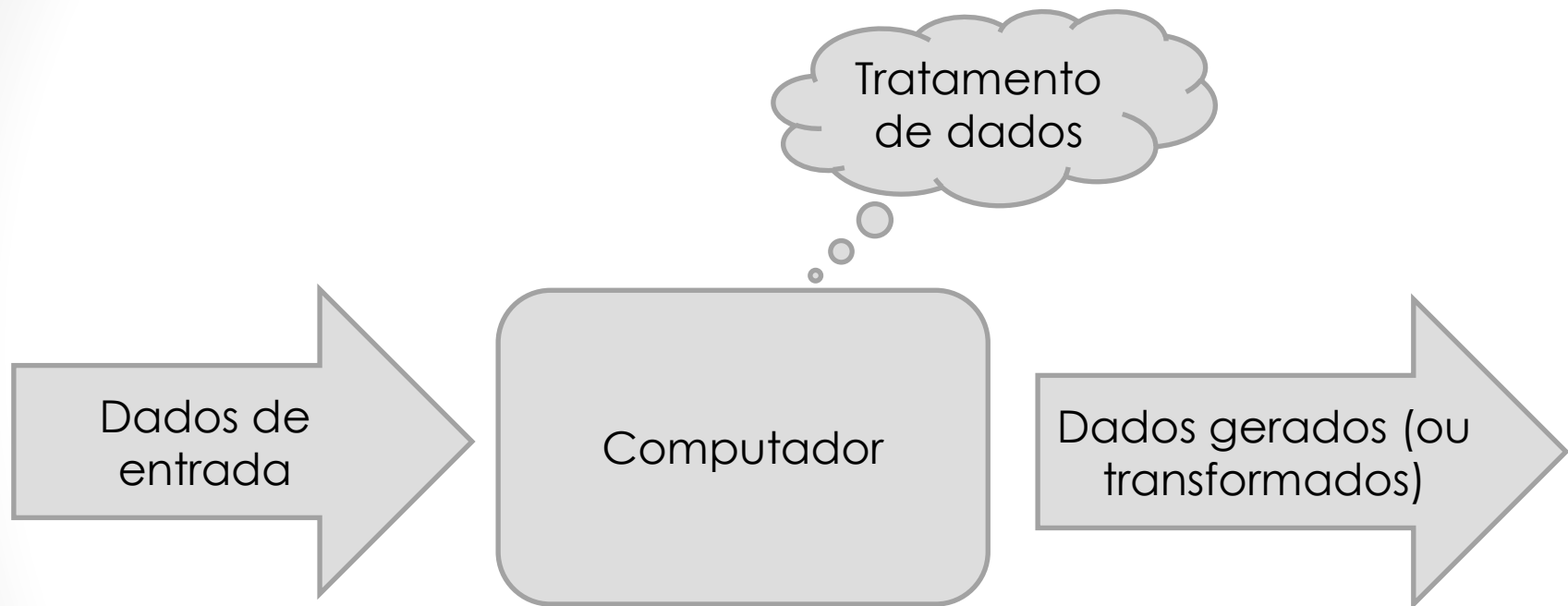


Princípios de funcionamento de computadores

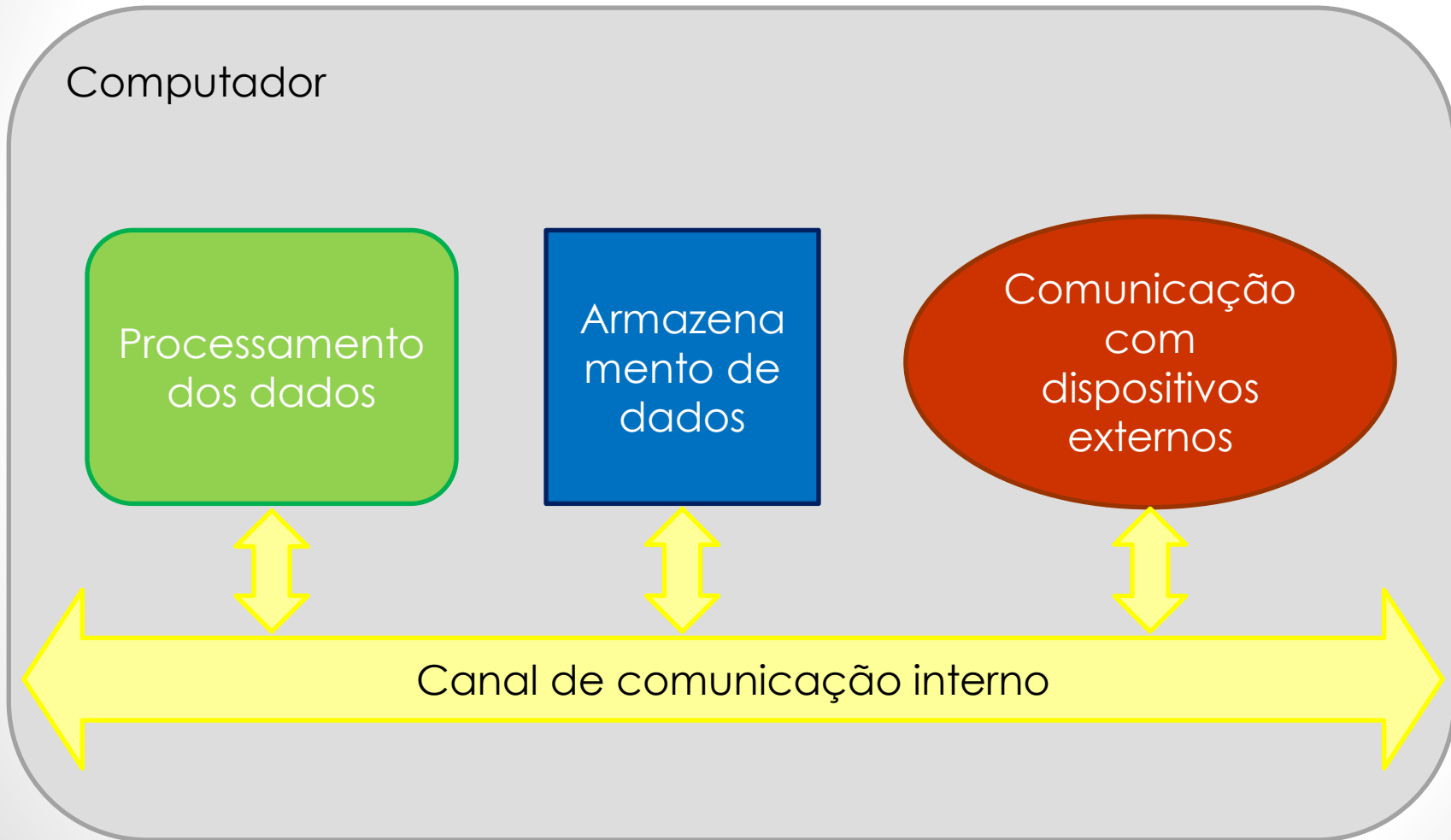
Prof. Gustavo Girão

Princípios de funcionamento

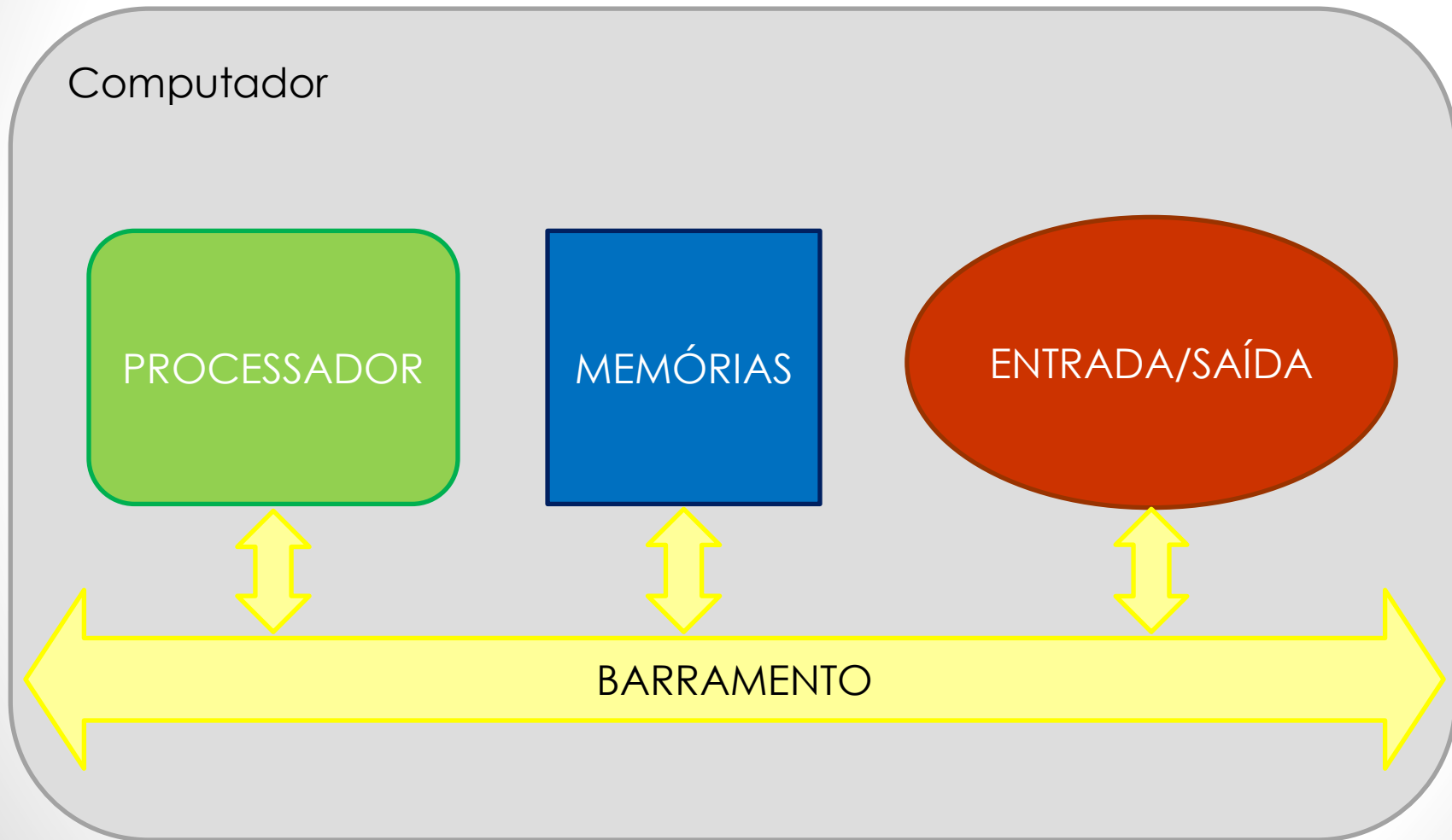
O que existe em um computador?



O que existe em um computador?



O que existe em um computador?



Como funciona?

- Dividido em 4 partes
 - **Processador(CPU - Central Processing Unit)**
 - ✧ Executa as operações de acordo com um programa (instruções)
 - ✧ Altas frequências de operação (hoje ~4 GHz)
 - **Entrada de dados:** teclado, mouse, etc
 - ✧ Permite a interação com um usuário
 - **Saída de dados:** monitor, impressora, etc
 - ✧ Permite a comunicação com o usuário
 - **Memória:**
 - ✧ Armazenamento de dados e de programas

O Processador (ou CPU)

- **Carrega** programa (conjunto de instruções binária) na memória
 - Quanto mais próximas do processador estas instruções estiverem, melhor!
- **Busca** os dados necessários para a execução das instruções
 - De novo: quanto mais perto, melhor!
- **Executa** cada instrução do programa
 - Processadores simples executam **sequencialmente**.
 - Ao longo dos anos percebeu-se que, se possível, executar **fora de ordem** acelera a execução
- **Salva** o(s) resultado(s) de volta na memória

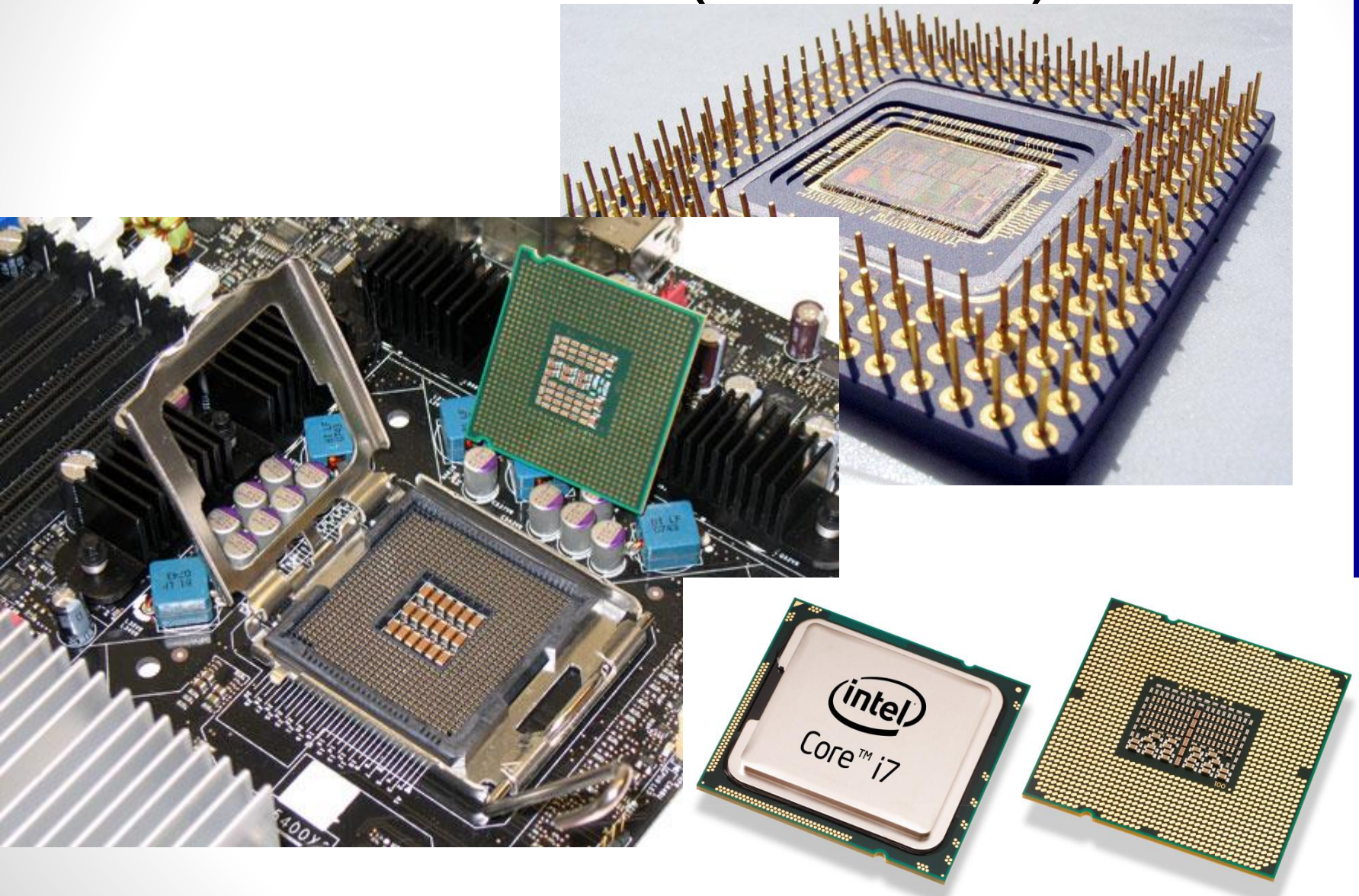
O Processador (ou CPU)

- O modo como o processador se organiza internamente para realizar estas tarefas é chamado de **Caminho de Dados**.
- Existem alguns paradigmas e eles diferem entre si principalmente em **quando** a instrução seguinte deve ser executada.
- De maneira geral, quanto mais **elaborada** é a solução, mais **controle** ela necessita
- Vamos estudar:
 - Vantagens e Desvantagens de cada paradigma
 - Como se deu a evolução das soluções de desempenho

O Processador (ou CPU)

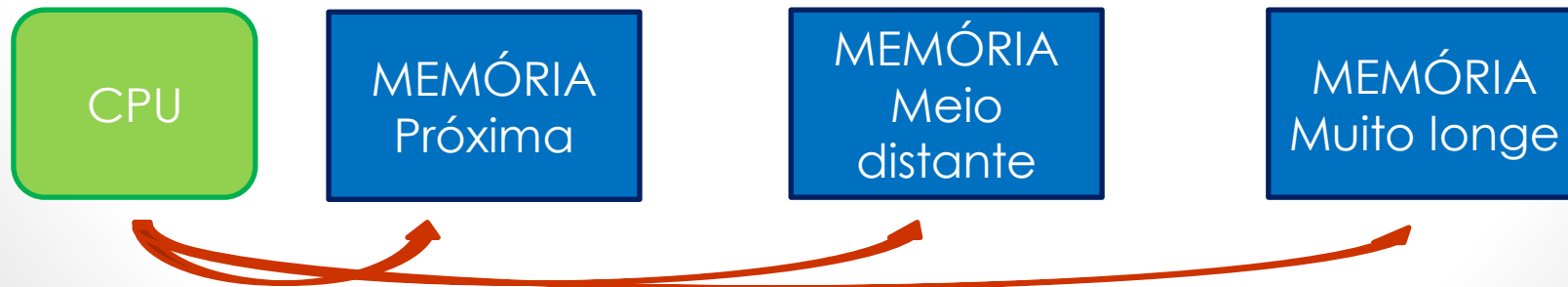
- Além da sua estrutura interna (**Organização**), o processador tem uma **Arquitetura** bem definida
 - **Formato** de uma instrução
 - Como dizer onde estão os dados a serem manipulados (**modo de endereçamento**)
- Estudo de caso: MIPS
 - Uso do Simulador MARS

O Processador (ou CPU)



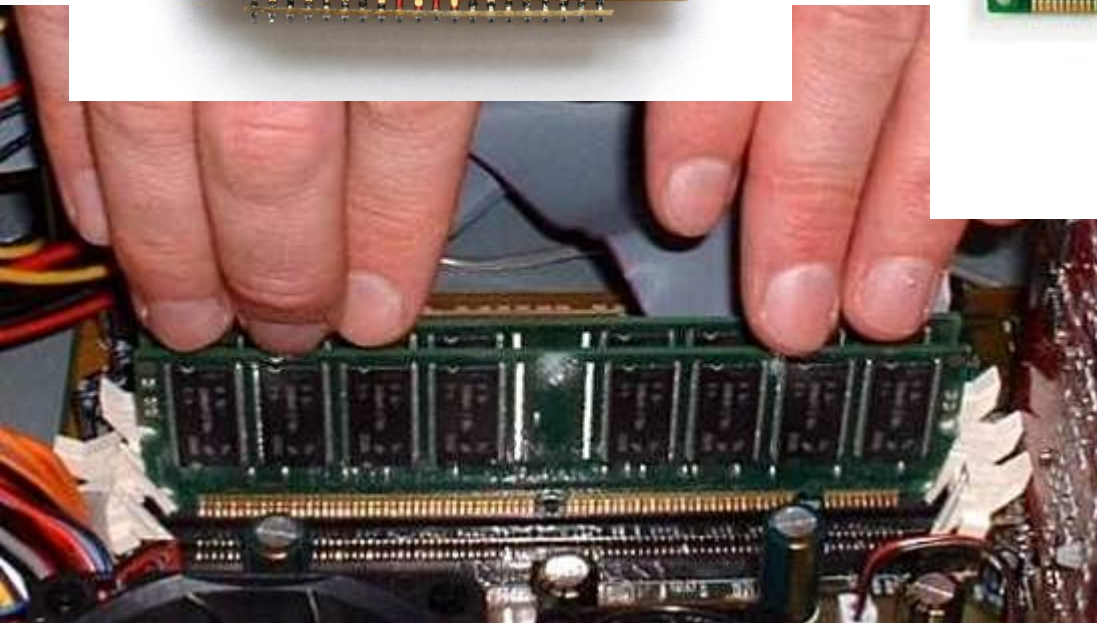
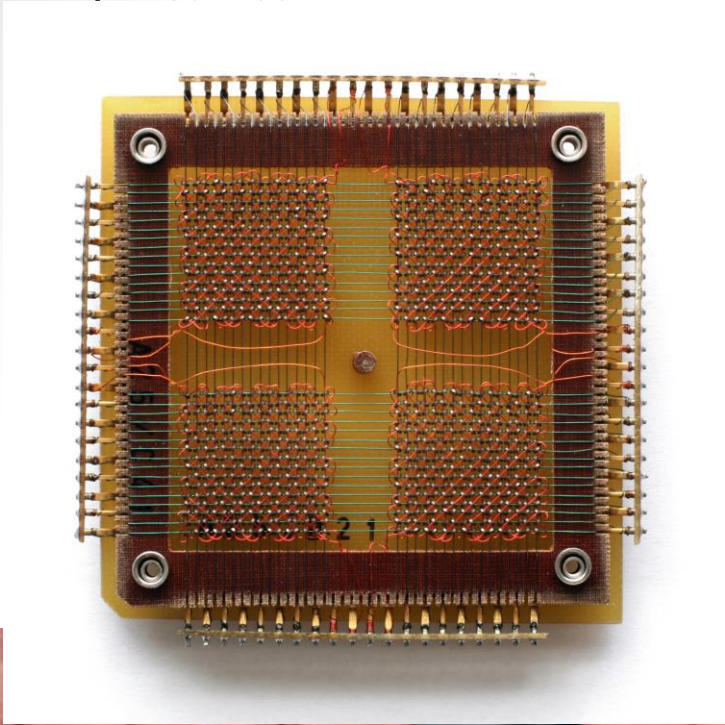
A Memória

- Armazena todas as instruções e dados necessários
- É bem mais lenta do que os processadores
 - Isso causa muitos problemas...
- Frequentemente é utilizada e por isso, seu tempo de resposta (**latência**) tem um enorme impacto no desempenho
- Memórias são dispostas em uma sequência ou **hierarquia**.
 - Tudo na esperança de diminuir a **latência**



A Memória

- Existem memórias **muito** rápidas que são mantidas **dentro** do processador
 - **Registradores**
 - Mas são **muito** caras (nada é de graça nessa vida...)
- Outras memórias são rápidas (não tanto quanto registradores)
 - Memórias **cache**
 - Não são **tão caras** quanto registradores
- Se o dado não está em nenhuma dessas, busca-se na memória **principal** (maior e mais lenta)
- Em ultimo caso acessa-se a memória **secundária**
 - Discos (HDs), fitas magnéticas, etc.



Entrada e Saída

- Toda computação realizada precisa de uma entrada (dados) a serem manipulados para gerar uma saída
- Existe em todo computador
- Equipamentos eletrônicos que se comunicam com um computador são chamados de **periféricos**.
- Periféricos podem ter diferentes características elétricas ou mecânicas.
 - A natureza da sua atuação também pode ser bem diferente
- Cada periférico pode ter uma maneira própria de enviar (ou receber) dados do computador

Entrada e Saída

- Existem vários paradigmas que definem:
 - De que maneira o processador envia (ou recebe dados)
 - Qual o grau de envolvimento do processador
 - Qual a prioridade de cada periférico
- Vamos estudar várias alternativas de implementação de controladores de entradas/saída

Barramentos

- Todos os componentes internos de um computador precisam se comunicar de alguma maneira.
 - Barramentos são **fiações** que realizam esta conexão
- Determinam a velocidade da comunicação interna do computador e impactam em ações simples como buscar dados da memória principal.
- **Árbitro** do barramento
 - Essencial para manter a prioridade das comunicações

Barramentos

- Existem diversas implementações de barramento
 - Prioridades de uso do barramento
 - Dedicado vs. Multiplexado
 - Assíncrono vs. Síncrono
 - Paralelo vs. Serial
- Estudaremos os aspectos a serem levados em consideração na hora de projetar um barramento
 - Vantagens e desvantagens
 - Custo/Benefício

Como tudo começou?

Arquitetura de Von Neumann (1945)

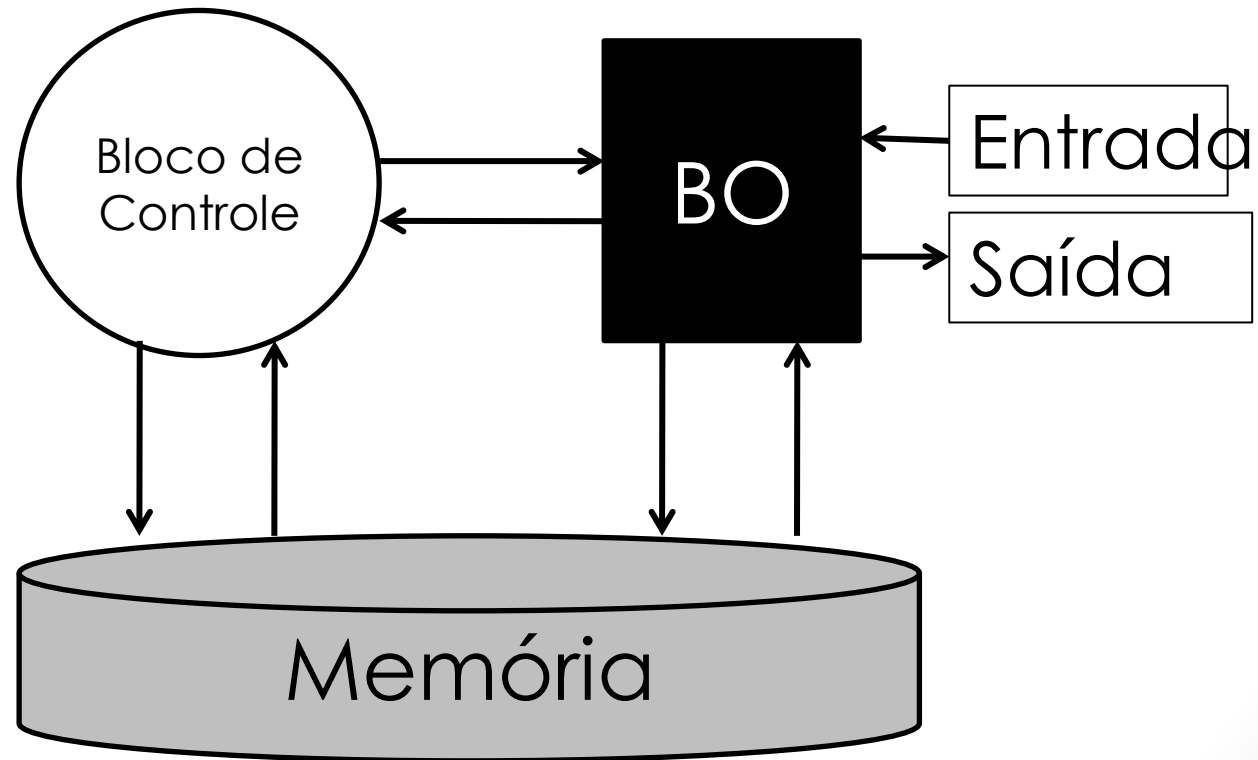


- Com raras exceções, todos os computadores atuais possuem essas mesmas funções e estrutura geral e assim são conhecidos como máquinas com **arquitetura de Von Neumann** (ou simplesmente máquinas de Von Neumann).

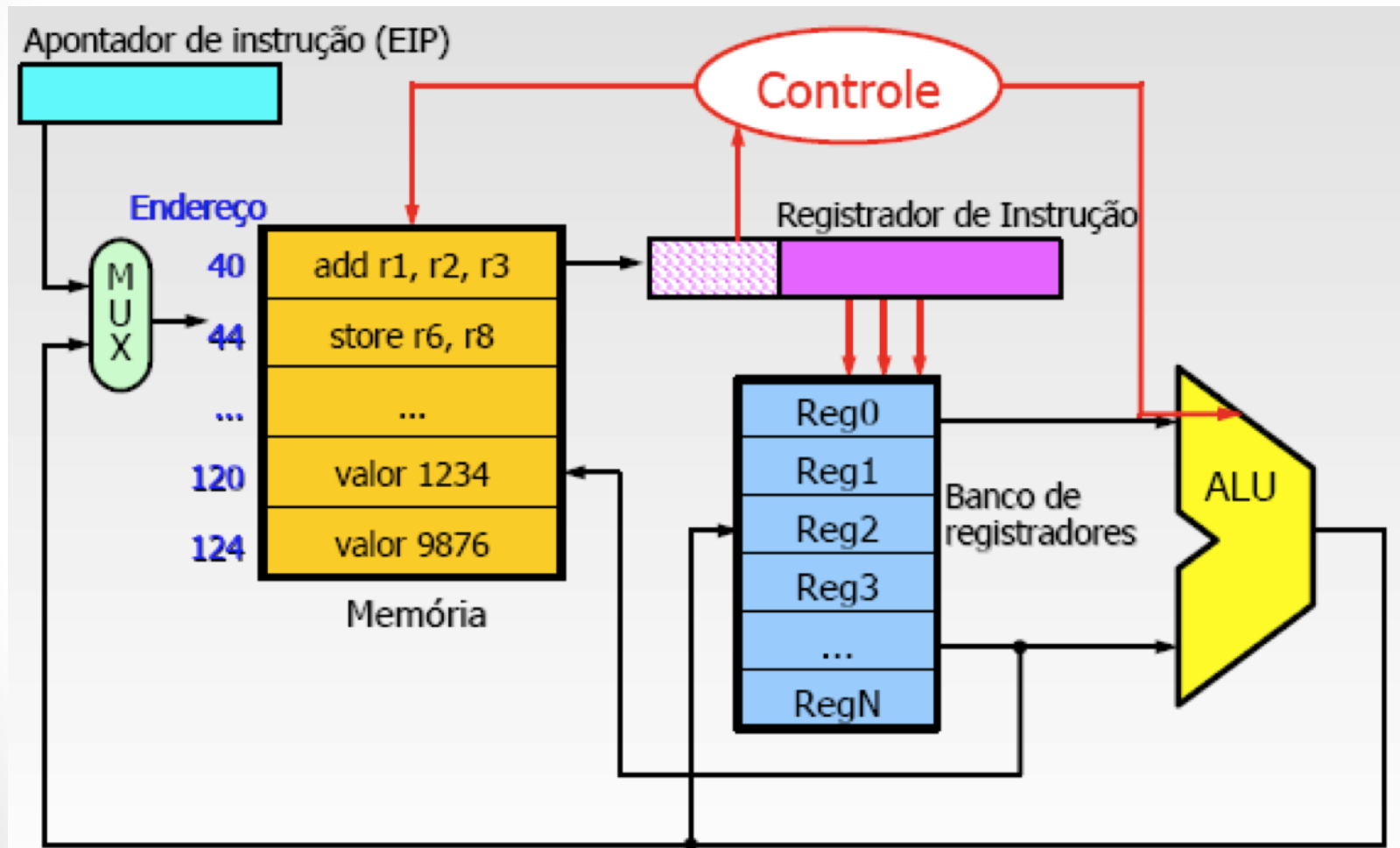
INICIAREMOS POR ESSE MODELO

Modelo de Von Neumann (Organização)

- Bloco Operativo
- Bloco de Controle
- Memória
- E/S



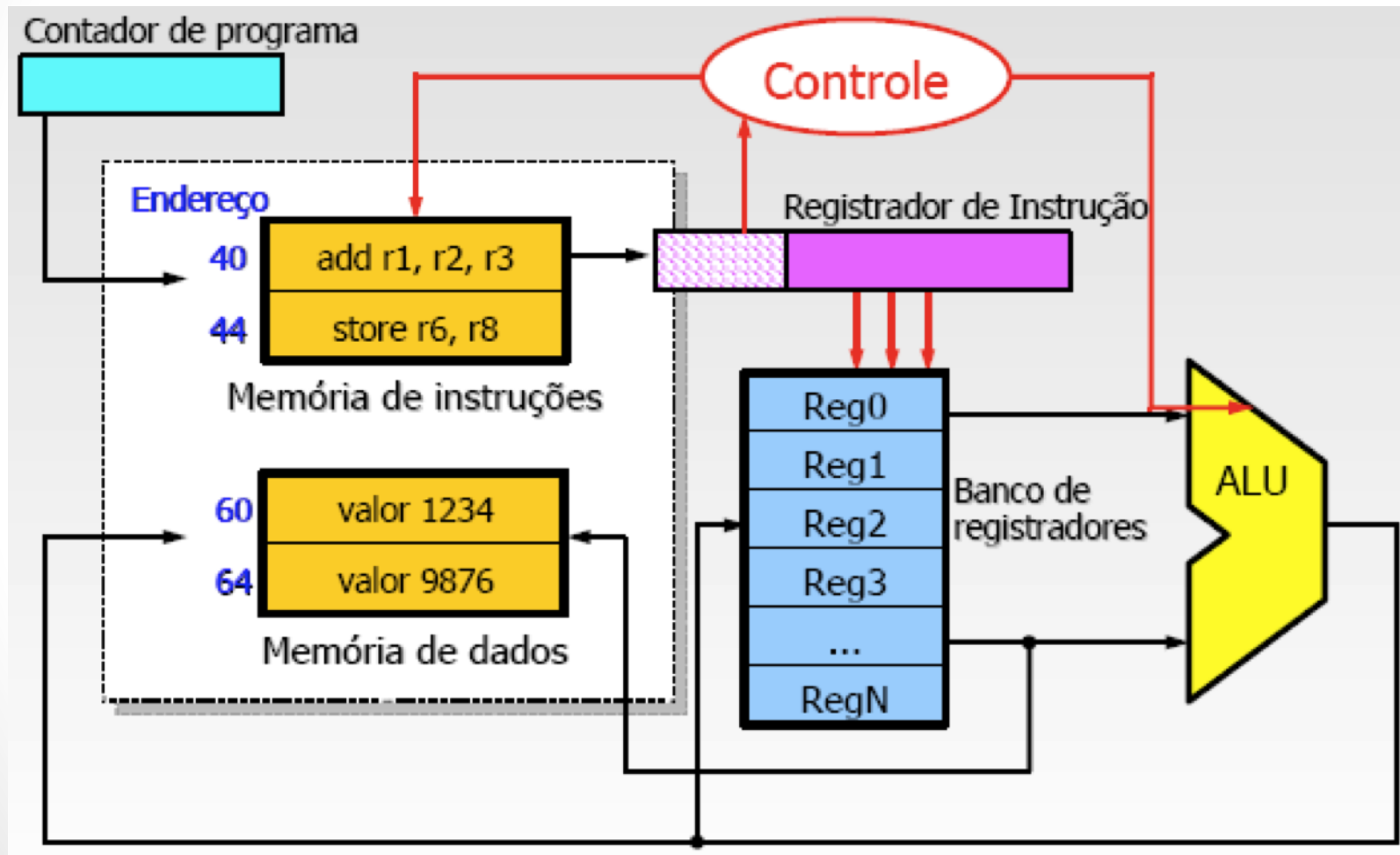
Modelo de Von Neumman (Organização)



Arquitetura de Harvard

- Utiliza memórias separadas:
 - Instruções
 - Dados
- O acesso a estas memórias é feito por barramentos distintos
- Vantagem
 - Possível ler/escrever instruções e dados concorrentemente (ao mesmo tempo)

Modelo de Harvard



Máquinas Não-Von Neumann

- Máquinas paralelas: várias unidades de processamento executando programas de forma cooperativa, com controle centralizado (SIMD ou MIMD)
 - Em partes... Mesmo este modelo, também é fortemente inspirado no modelo de Von Neumann
- Redes neurais artificiais: não executam instruções de um programa; resultados são gerados a partir de respostas a estímulos

Modelo de Von Neumann

:: Características

- **Memória** é dividida em palavras e contém dados e instruções, indistintamente.
- **Palavra** é a unidade básica de transferência de / para memória.
- Palavras são localizadas através de um **endereço**.
- **Programa** é uma sequência de instruções, colocadas numa sequência de endereços.
- A execução de um programa corresponde à execução **sequencial** de suas instruções.
- Dados, instruções e endereços são codificados em **binário**.

Modelo de Von Neumann

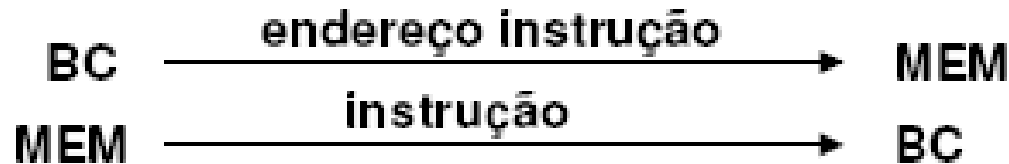
:: Execução de Instruções

- **Busca** da **próxima** instrução na memória
 - Manda endereço, volta instrução
- **Decodificação** da instrução
 - Interpreta código da instrução
- Se a instrução precisa de dados (na memória)
 - Manda endereço, busca dado
- **Execução** da instrução
 - Executa ações específicas para cada instrução
- **Grava** o resultado da operação
 - Manda endereço, manda dado

Modelo de Von Neumann

:: Exemplo de Execução de Instruções

✧ Instrução de soma de dois operandos:



Modelo de Von Neumann

:: Aspectos Temporais

- Uma **instrução**:
 - Um ou vários acessos à memória
- Tempo de execução de uma instrução:
 - Praticamente igual à soma dos tempos de acesso à memória.
- Tempo de acesso à memória principal:
 - Da ordem de 60 a 70 ns
- **Tempos de micro-operações**
(somas, transferências entre registradores, etc.)
 - São bem menores
 - Clock de 1 GHz representa ciclo de micro-operações de 1 ns

Modelo de Von Neumann

:: Gargalo

- **Tráfego de informações** (endereços, dados, instruções) entre CPU e memória
 - Vai endereço da instrução
 - Volta instrução, que contém código da operação e endereços dos operandos
 - Vão endereços dos operandos
 - Vão e voltam operandos
 - Instruções precisam especificar endereços dos dados e podem ocupar 2 a 3 palavras
 - 2 a 3 acessos à memória na busca da instrução



Modelo de Von Neumann

:: Problemas e Soluções

- Tempo de execução da instrução fica comprometido pelo (a)...
 - Sequencialidade das operações
 - Excesso de informações transferidas entre processador e memória
 - Tempo de acesso à memória
- Soluções:
 - Diminuir quantidade de informações a serem transferidas entre processador e memória.
 - Diminuir tempo aparente de acesso à memória.
 - ✦ Realizar operações em paralelo.
- Dimensões de projeto que podem ser otimizadas:
 - Tecnologia
 - Organização
 - Arquitetura

Modelo de Von Neumann

:: Diminuindo o gargalo

- Inclusão de memória cache:
 - Tempo de acesso compatível c/ tempo de execução das micro-operações
 - Tamanho bem menor do que a memória principal
- Introdução de registradores:
 - Dados utilizados frequentemente não precisam ser trazidos/levados de/para memória a cada utilização.
- Tecnologias mais avançadas de memória cache:
 - Aumentar o número de dados que serão utilizados no futuro

Modelo de Von Neumann

:: Diminuindo o gargalo

- **Tecnologias de memória principal mais eficientes**
 - Memórias DDR (Double Data Rate)
 - ✦ Transmitem dados na metade do tempo
 - ✦ Atualmente já existe a quinta geração deste tipo (DDR5)
- **Memórias secundárias mais rápidas**
 - Tecnologia SSD (Solid State Drives)
 - ✦ Semelhante à memória flash
 - ✦ Muito mais rápida que Discos Rígidos (HDs) por não utilizar componentes eletromecânicos

Aumentando o desempenho

- Introdução de **paralelismo**:
 - Na **busca** de instruções:
 - ✧ Pré-**fetch**, buffer de instruções
 - Na **execução** de instruções:
 - ✧ **Pipeline**
 - ✧ **Superescalaridade**
- Arquitetura **Harvard**: memórias separadas para dados e instruções
 - **Acessos paralelos**
- Processadores **RISC**
 - Muitos registradores
 - Instruções mais simples e mais rápidas

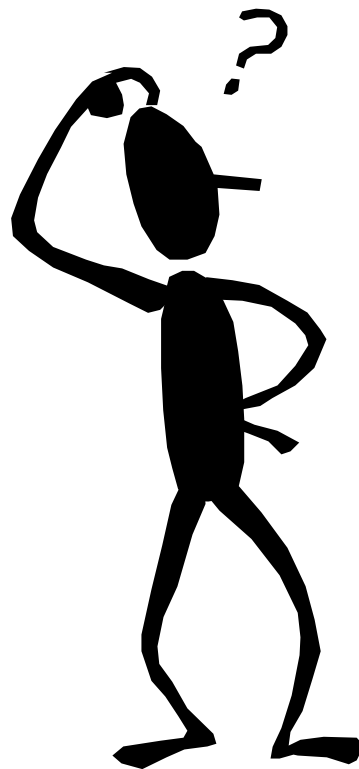
Aumentando o desempenho

- **Multithreading**

- Suporte dentro do processador para a execução de mais de uma thread (fluxo de execução)
- Controle específico de quais instruções pertencem a que threads
- A esperança é de que o processador fique ocioso por menos tempo

- **Múltiplos cores**

- Execução em paralelo de um mesmo algoritmo ou de mais de um programa



Para saber mais ...

- STALLINGS, William. Arquitetura e organização de computadores. 8. ed. São Paulo: Pearson, 2010. **Capítulo 2**
- PATTERSON, D.A. & HENNESSY, J. L. **Organização e Projeto de Computadores - A Interface Hardware/Software**. 3ª ed. Campus, 2005. **Capítulo 1**