Visão de Alto Nível e Barramentos

Conceitos Iniciais

- Um sistema de computação, em seu nível mais alto, é composto pela UCP, memória e dispositivos de E/S;

- Esses componentes devem ser conectados de alguma maneira;

- Analisar o comportamento externo de cada componente e a interconexão entre eles permite que mostrar o funcionamento de um sistema de computação.

Componentes de um computador

- Quase todos os projetos de computadores de arquiteturas convencionais hoje se baseiam na arquitetura *von Neumann*;

- A arquitetura *von Neumann* se baseia em três conceitos básicos:

  ✓ Os dados e as instruções são armazenados em uma única memória de leitura e escrita;

Componentes de um computador

✓ O conteúdo da memória é endereçado pela posição, independente do tipo de dados nela contidos;

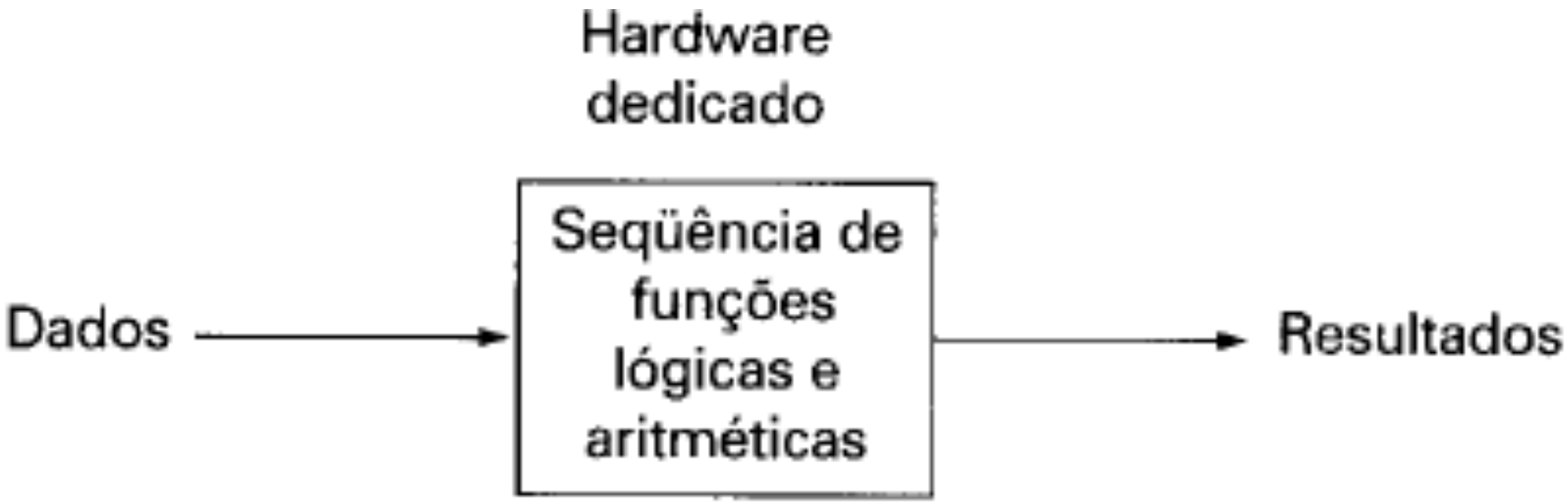
✓ A execução de instruções ocorre de modo sequencial.

4

Componentes *hardwired*

- Faz-se uso de um *hardware* específico para a execução de uma aplicação em particular;

- Para cada nova aplicação é preciso de um novo *hardware;*

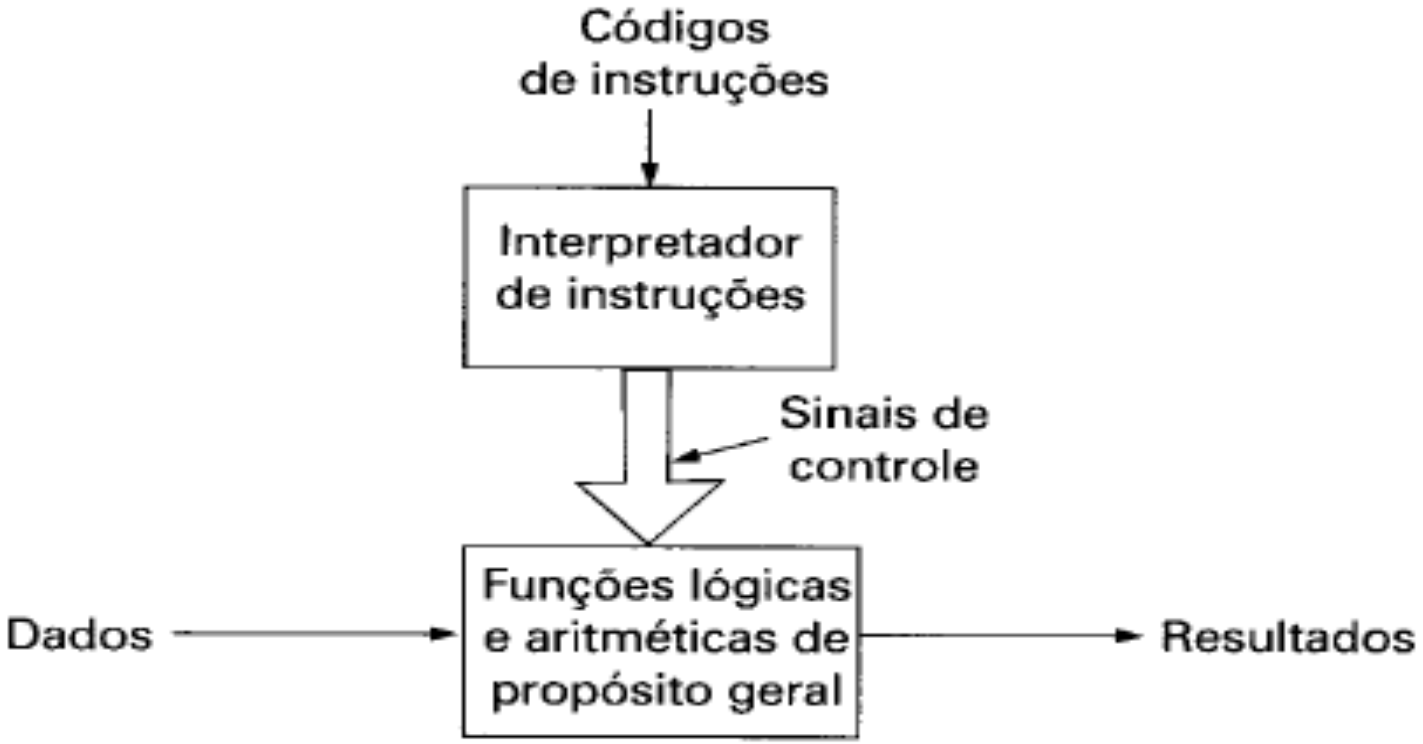


Fonte: Arquitetura e Organização de Computadores – William Stallings

Componentes de *software*

- Produz-se um *hardware* com propósito geral;

- Uso de códigos de controle para as novas instruções.



Fonte: Arquitetura e Organização de Computadores – William Stallings

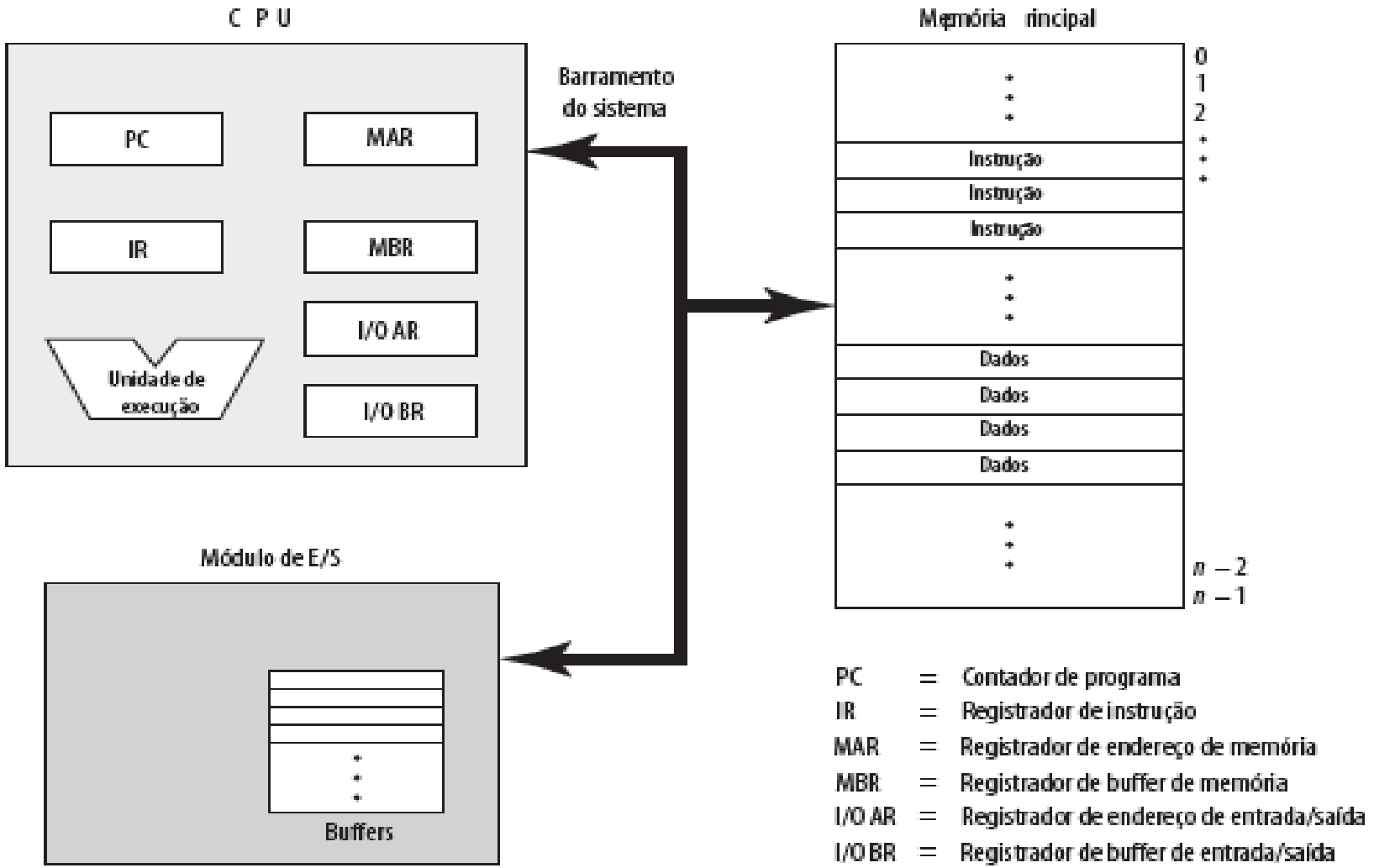
Um programa

- Uma sequência de etapas;
- Para cada etapa, é feita uma operação aritmética ou lógica;
- Para cada operação, é necessário um conjunto diferente de sinais de controle.

Unidade de controle - Necessidade

- Para cada operação, um código exclusivo é fornecido;
  - ✓ P.e. ADD, MOVE;
- Um segmento de hardware aceita o código e emite os sinais de controle;
- Ao sistema computacional passa a operar.

Componentes de um sistema computacional



Fonte: Arquitetura e Organização de Computadores – William Stallings

Ciclo de instrução

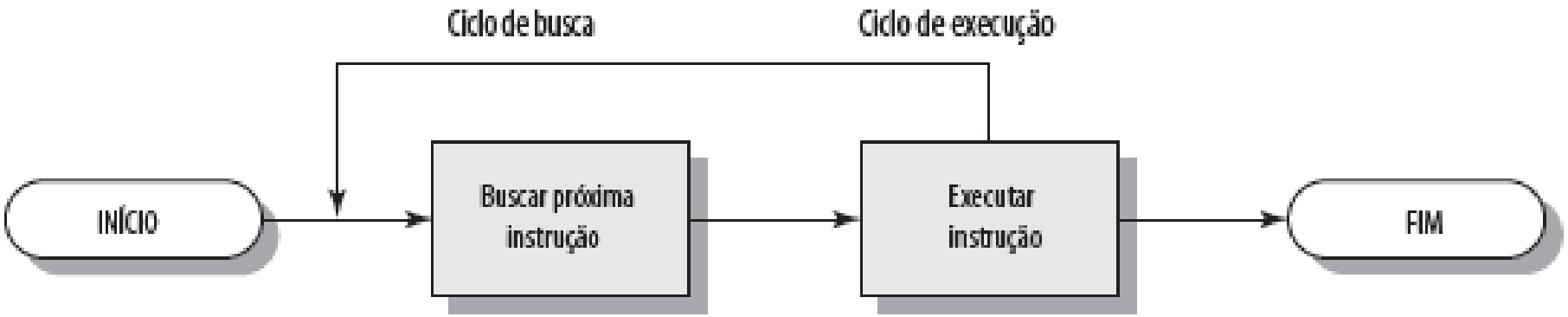
- Para que uma tarefa seja cumprida, instruções devem ser passadas;

- A utilização de cada um dos componentes, dependera da fase do ciclo de instrução;

- Assim, a definição de um ciclo de instrução básico deve ser definido e cumprido pelo programa.

Ciclo de instrução

- Duas etapas (ciclo básico):
  - ✓ Busca;
  - ✓ Execução;



Fonte: Arquitetura e Organização de Computadores – William Stallings

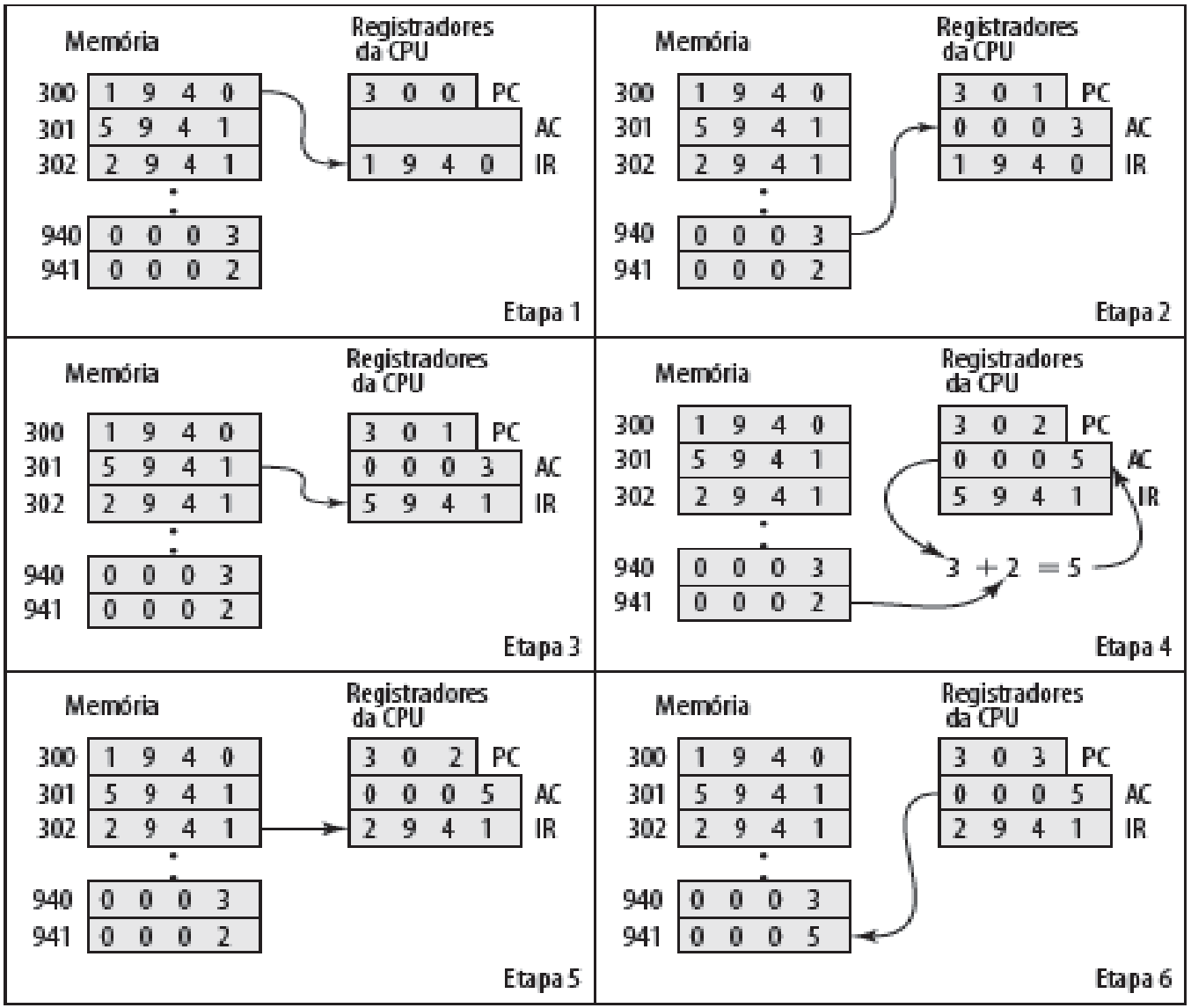
Ciclo de busca – Modelo Básico

- Contador de Programa (PC) mantém endereço da próxima instrução a buscar;
- Processador busca instrução do local de memória apontado pelo PC;
- Incrementar PC:
  - ✓ A menos que seja informado de outra forma;
- Instrução carregada no Registrador de Instrução (IR);
- Processador interpreta instrução e realiza ações exigidas;

Ciclo de execução – Modelo Básico

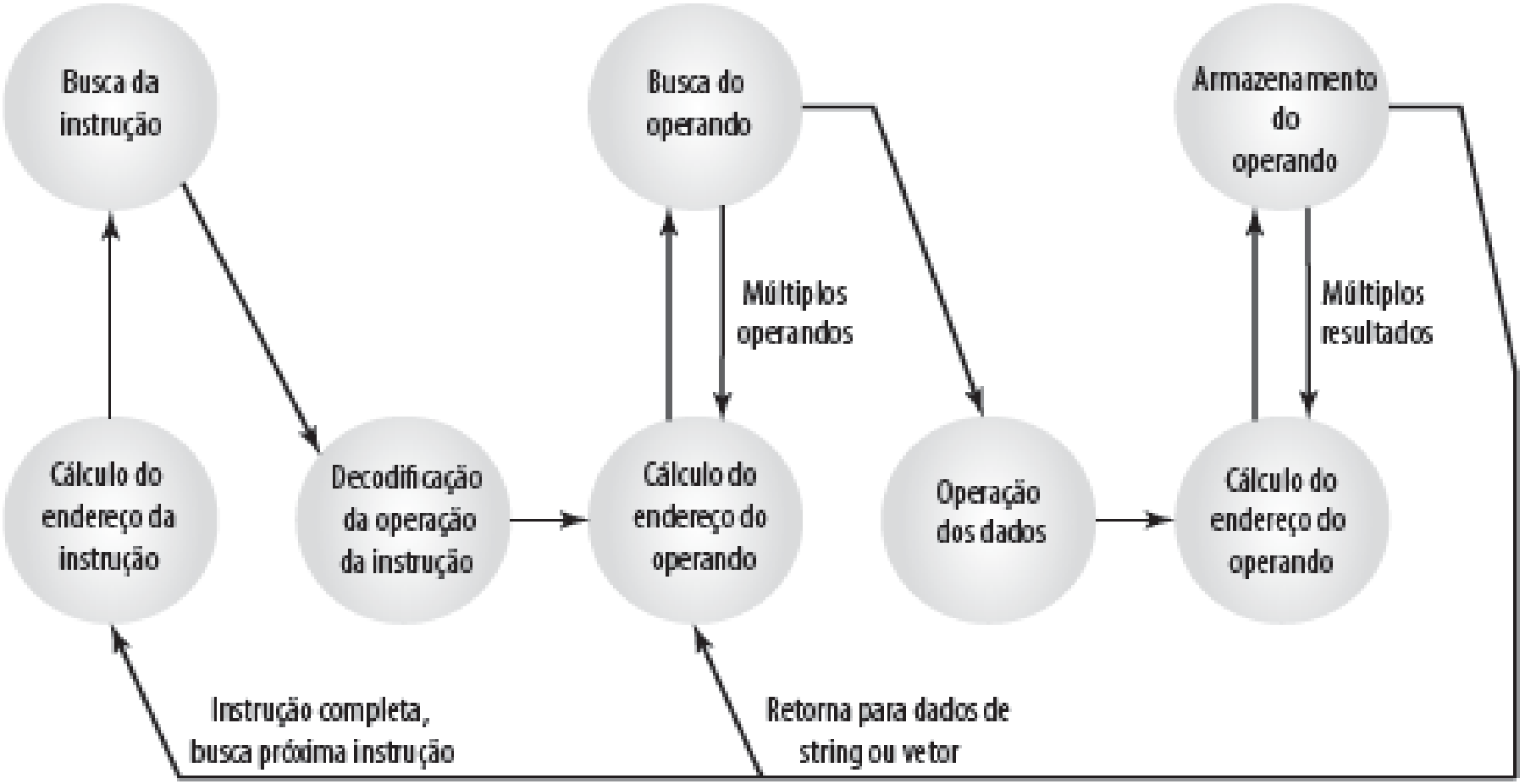
- Processador-memória:
  - ✓ Transferência de dados entre CPU e memória principal.
- E/S do processador:
  - ✓ Transferência de dados entre CPU e módulo de E/S.
- Processamento de dados:
  - ✓ Alguma operação aritmética ou lógica sobre dados.
- Controle:
  - ✓ Alteração da sequência de operações.

Execução de um programa - Exemplo



Fonte: Arquitetura e Organização de Computadores – William Stallings

Ciclo de instrução completo – Diagrama de Estados



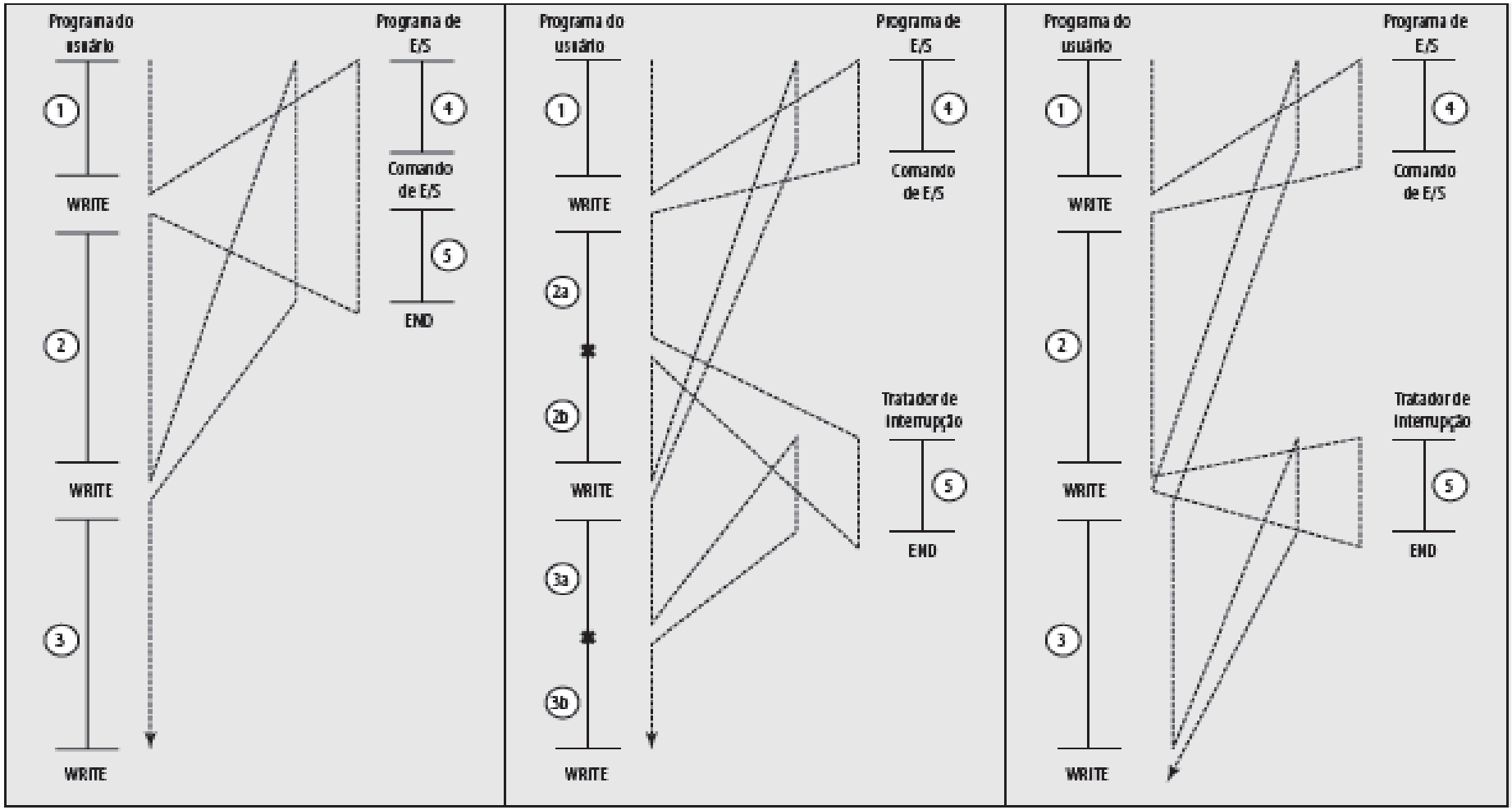
Fonte: Arquitetura e Organização de Computadores – William Stallings

Fluxo do Programa – Interrupções

- Mecanismo pelo qual outros módulos (p.e. E/S) podem interromper a sequência de processamento normal;

- Exemplos de interrupção:
  - ✓ Programa: estouro, divisão por zero;
  - ✓ Timer: Timer dentro do processo;
  - ✓ E/S: Do controlador de E/S;
  - ✓ Falha de hardware: erro de paridade de memória;

Controle de fluxo de programa



(a) Sem Interrupções (b) Interrupções; curta espera pela E/S (c) Interrupções; longa espera pela E/S

Fonte: Arquitetura e Organização de Computadores – William Stallings

Interrupção – Ciclo

- Adicionado ao ciclo de instrução;

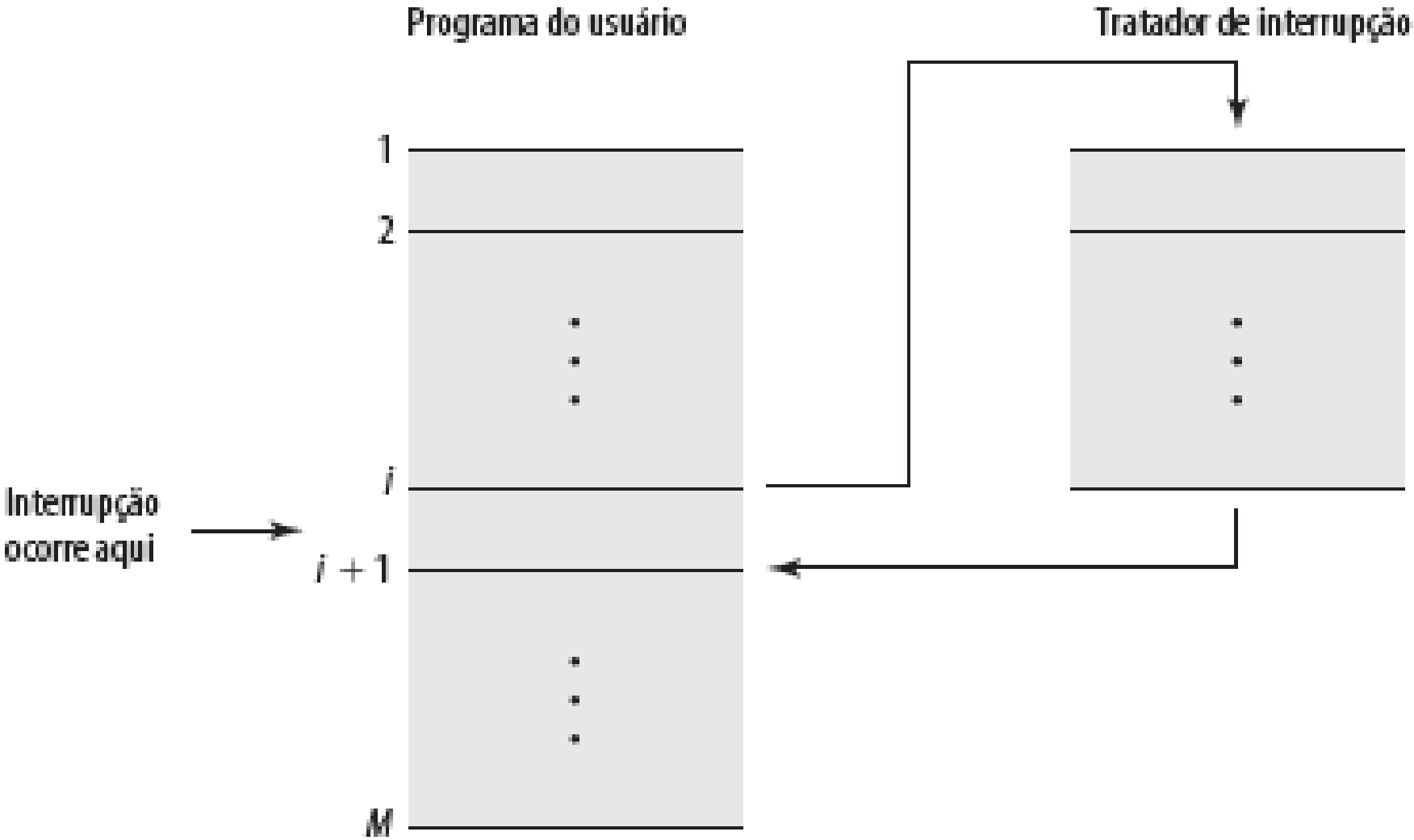
- Processador verifica interrupção;
  - ✓ Indicado por um sinal de interrupção;

- Se não houver interrupção, busca próxima instrução;

Interrupção – Ciclo

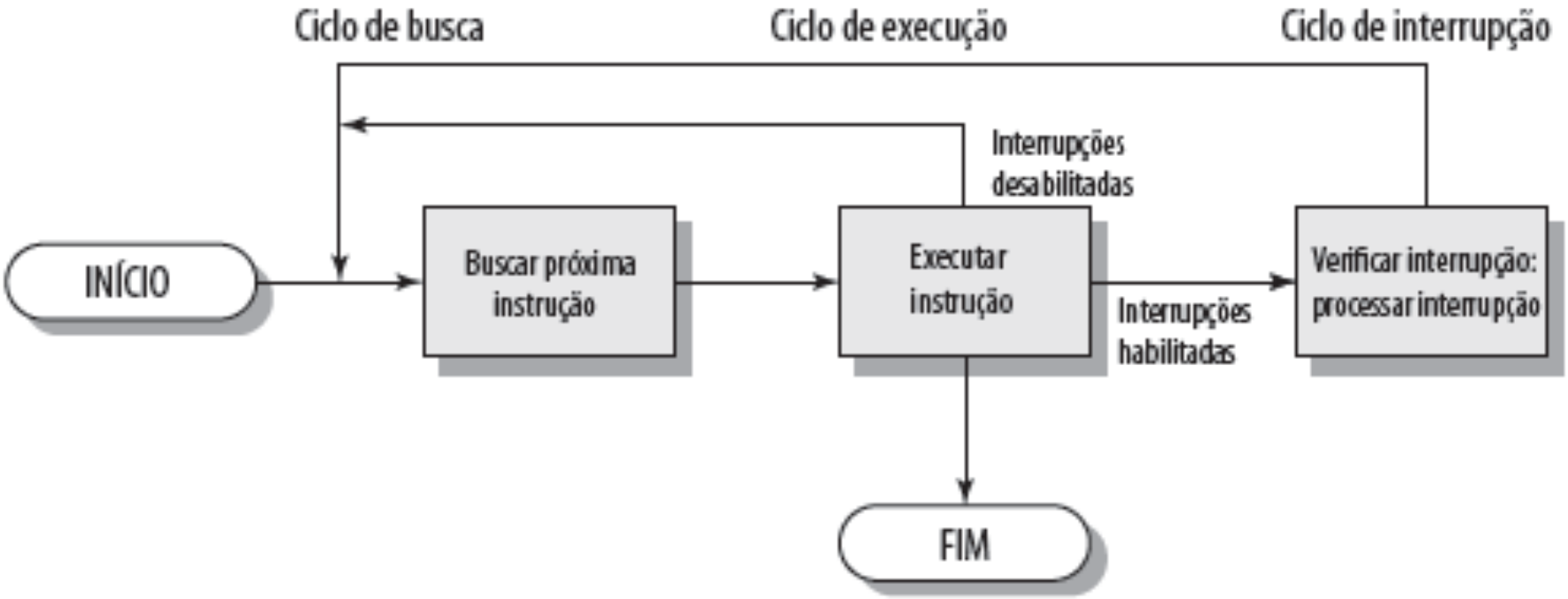
- Se houver interrupção pendente:
  - ✓ Suspende execução do programa atual;
  - ✓ Salva contexto;
  - ✓ Define PC para endereço inicial da rotina de tratamento de interrupção;
  - ✓ Interrupção de processo;
  - ✓ Restaura contexto e continua programa interrompido.

Na prática



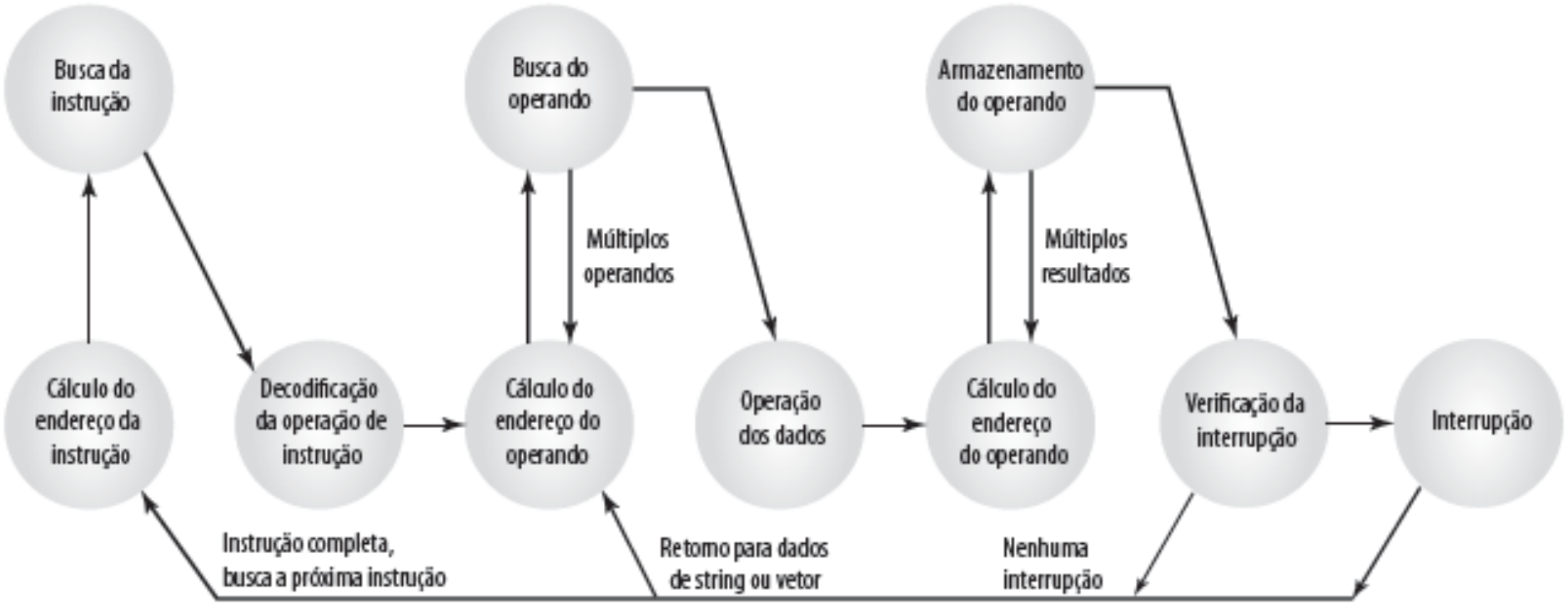
Fonte: Arquitetura e Organização de Computadores – William Stallings

Ciclo de instrução com interrupções



Fonte: Arquitetura e Organização de Computadores – William Stallings

Ciclo de instrução (com interrupções) – diagrama de estado



Fonte: Arquitetura e Organização de Computadores – William Stallings

Múltiplas interrupções

- Diferentemente do que se pensa, pode não haver apenas **uma** em um momento do ciclo;

- A possiblidade de haver mais de uma interrupção pode gerar um efeito cascata;

- Para o sistema computacional é algo bastante crítico;

Múltiplas interrupções

- O processador deve estar apto a tratar múltiplos casos;
- Para isso, alguma restrições devem ser criadas;
- Desativar interrupções:
  - ✓ Processador ignorará outras interrupções enquanto processa uma interrupção;
  - ✓ Interrupções permanecem pendentes e são verificadas após primeira interrupção ter sido processada;
  - ✓ Interrupções tratadas em sequência enquanto ocorrem.

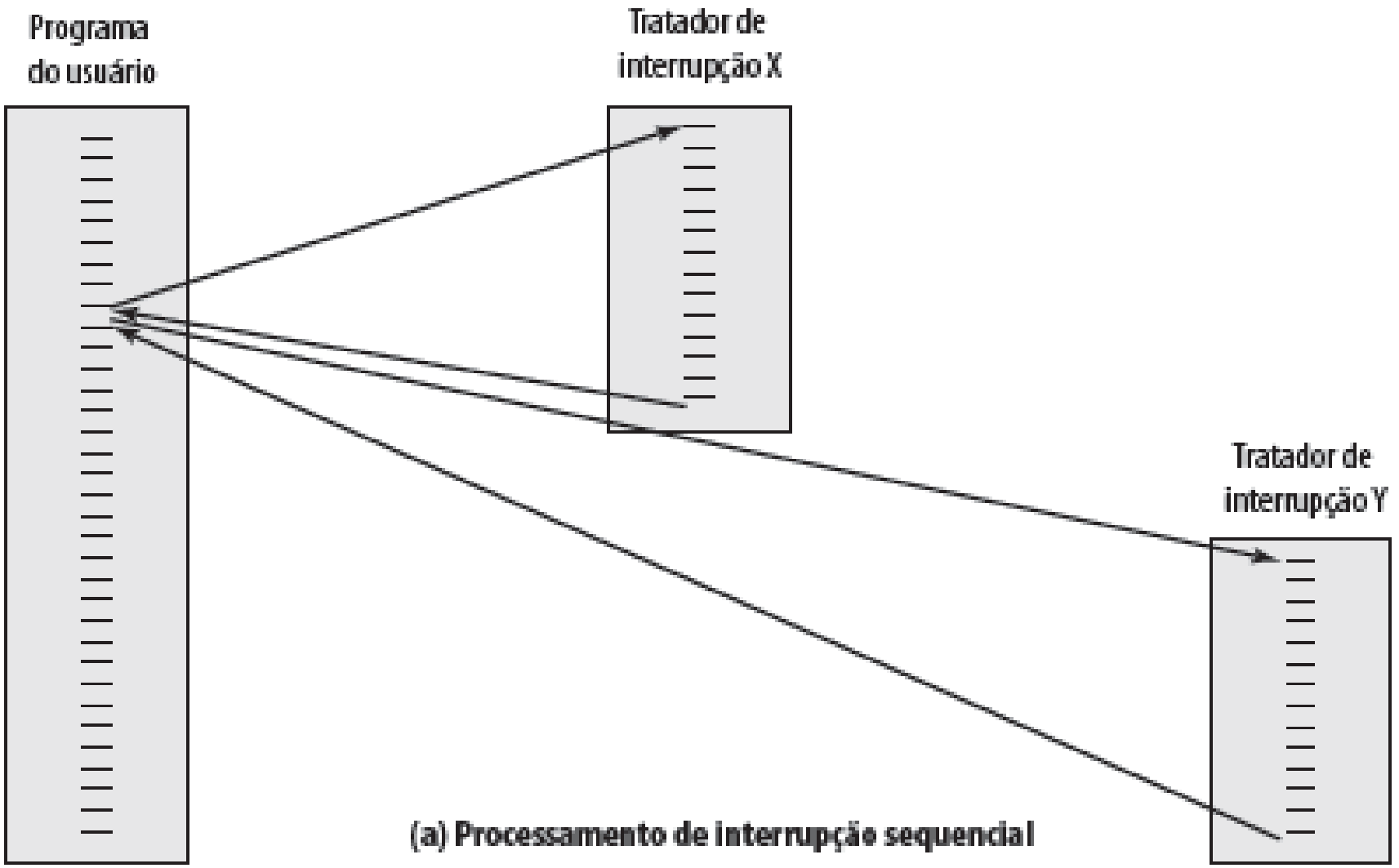
Múltiplas interrupções

- Definir prioridades:
  - ✓ Interrupções de baixa prioridade podem ser interrompidas por interrupções de prioridade mais alta;
  - ✓ Quando interrupção de maior prioridade tiver sido processada, processador retorna à interrupção anterior.

Múltiplas interrupções

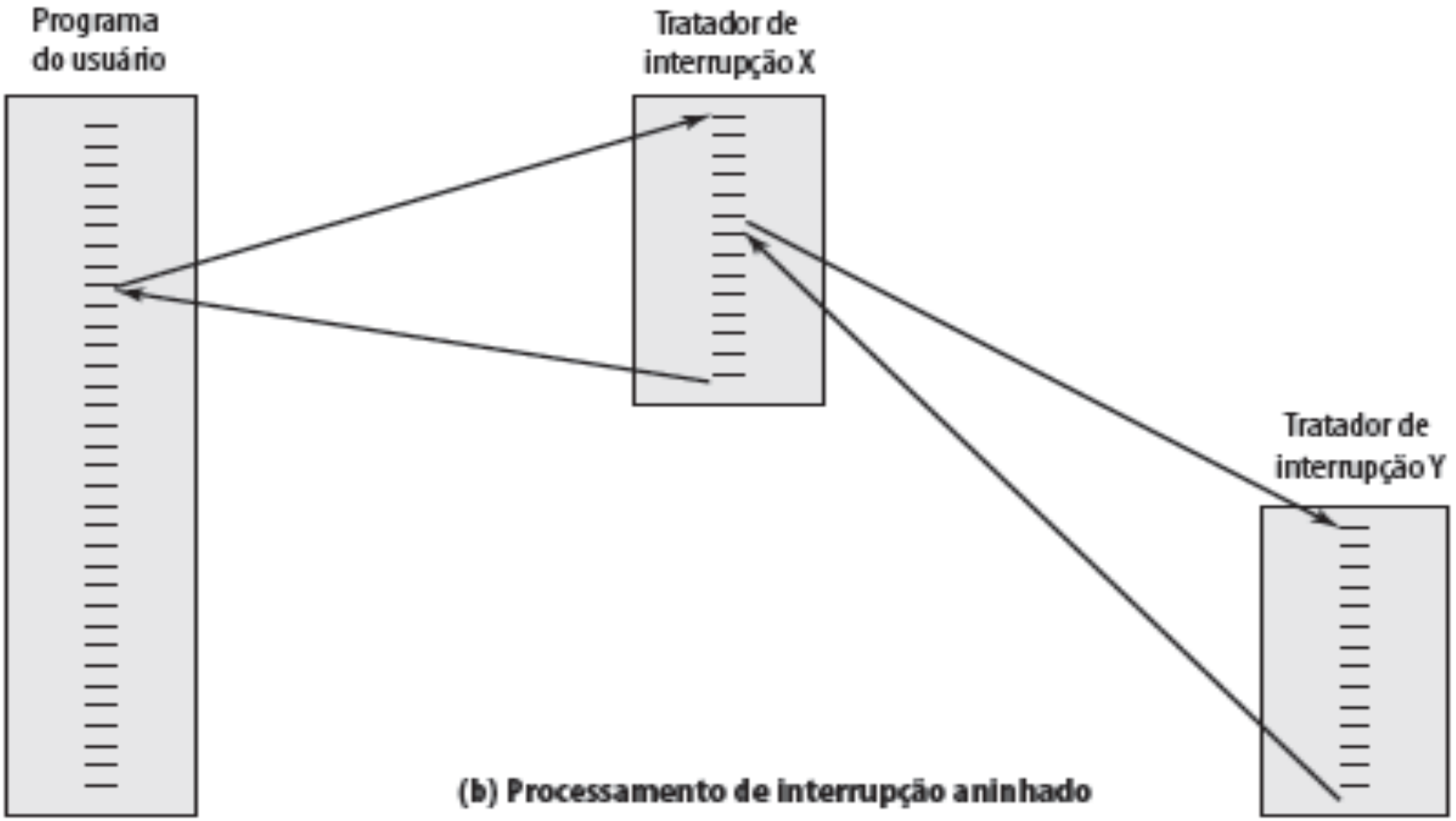
- A seguir, encontram-se alguns casos de múltiplas interrupções

Múltiplas interrupções – sequenciais



Fonte: Arquitetura e Organização de Computadores – William Stallings

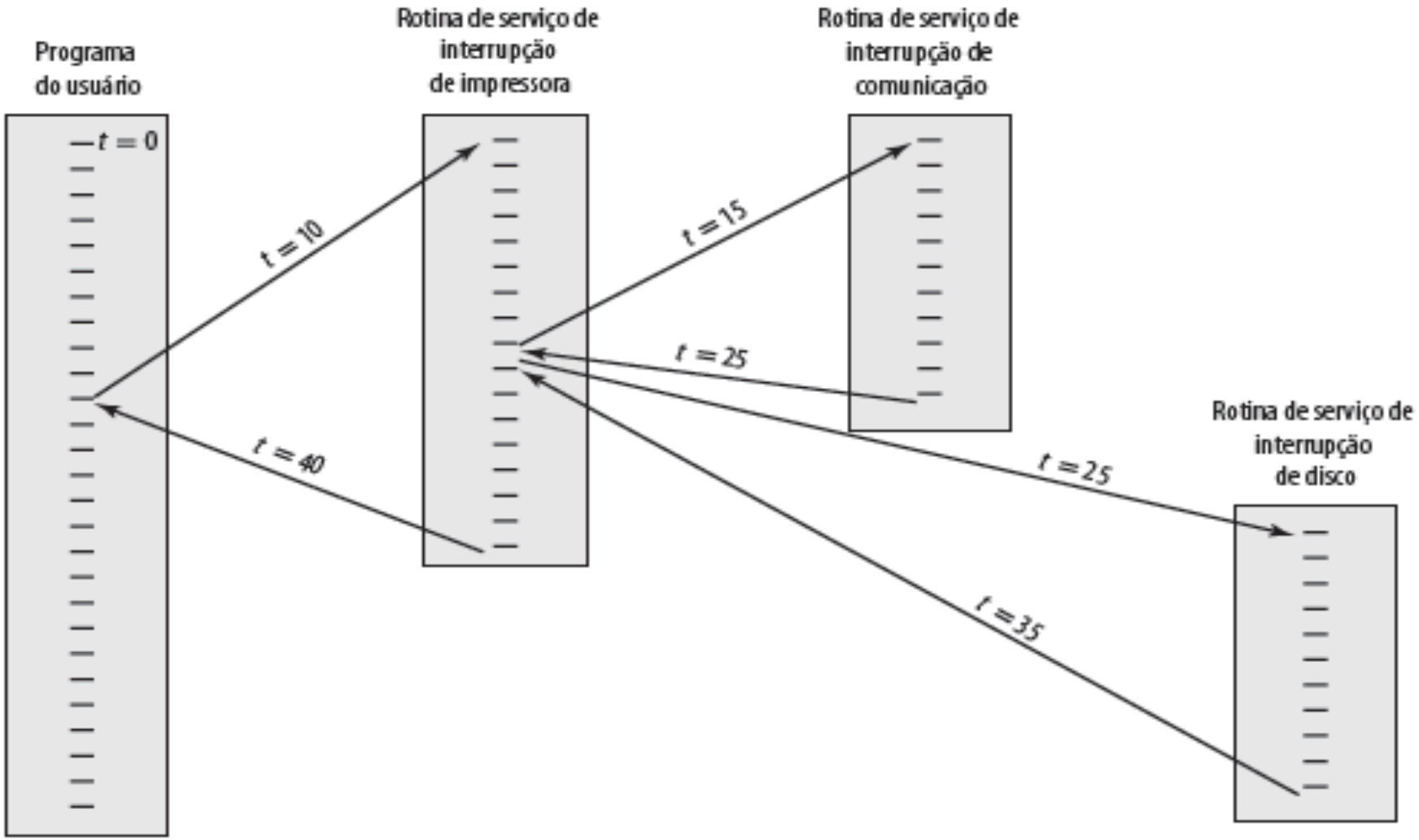
Múltiplas interrupções – aninhadas



(b) Processamento de interrupção aninhado

Fonte: Arquitetura e Organização de Computadores – William Stallings

Múltiplas interrupções controladas por tempo



Fonte: Arquitetura e Organização de Computadores – William Stallings

Comunicação

- Tudo o que foi apresentado anteriormente só opera se puder se comunicar;

- A garantia de que o comando chegará é fundamental para o funcionamento do sistema computacional;

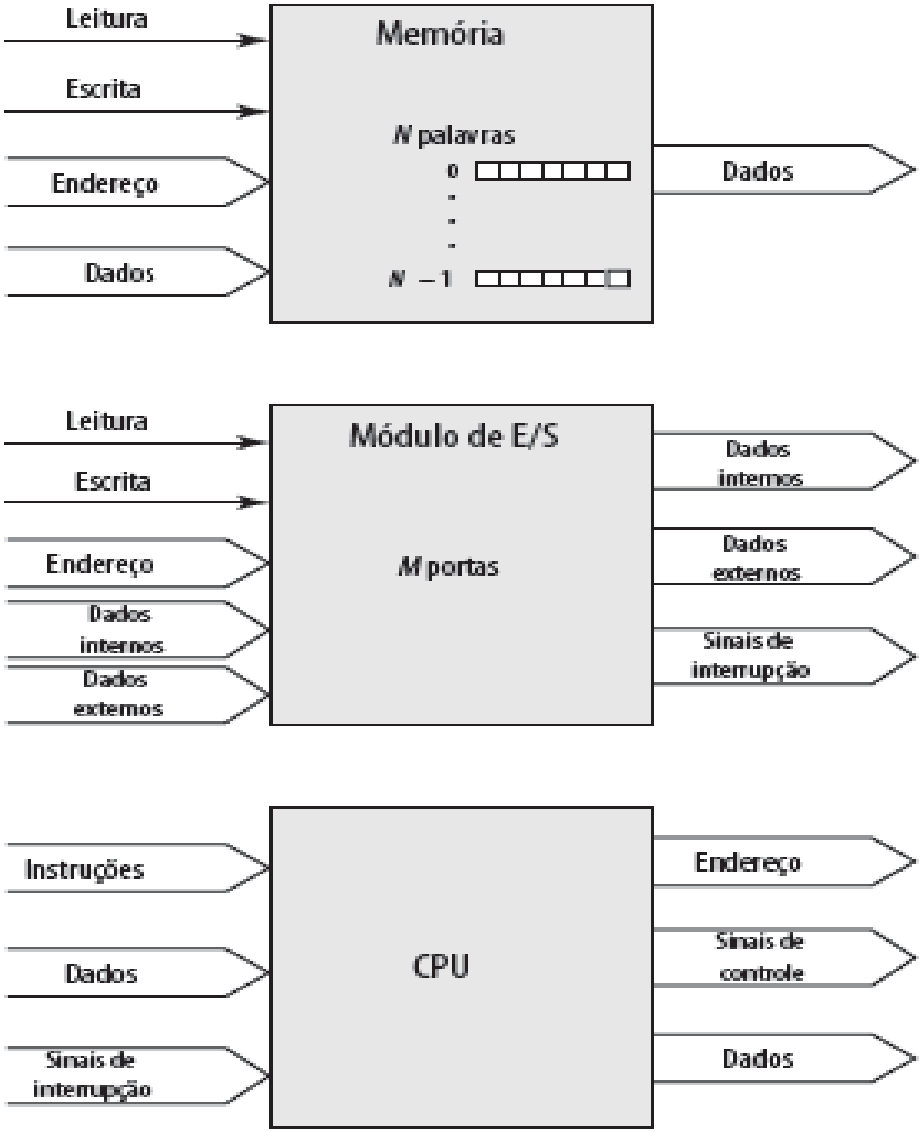
- Todas as unidades devem ser conectadas;

Comunicação

- A comunicação não deve ter obstáculos;

- Tipo de conexão diferente para tipo de unidade diferente.
  - ✓ Memória;
  - ✓ Entrada/saída;
  - ✓ CPU.

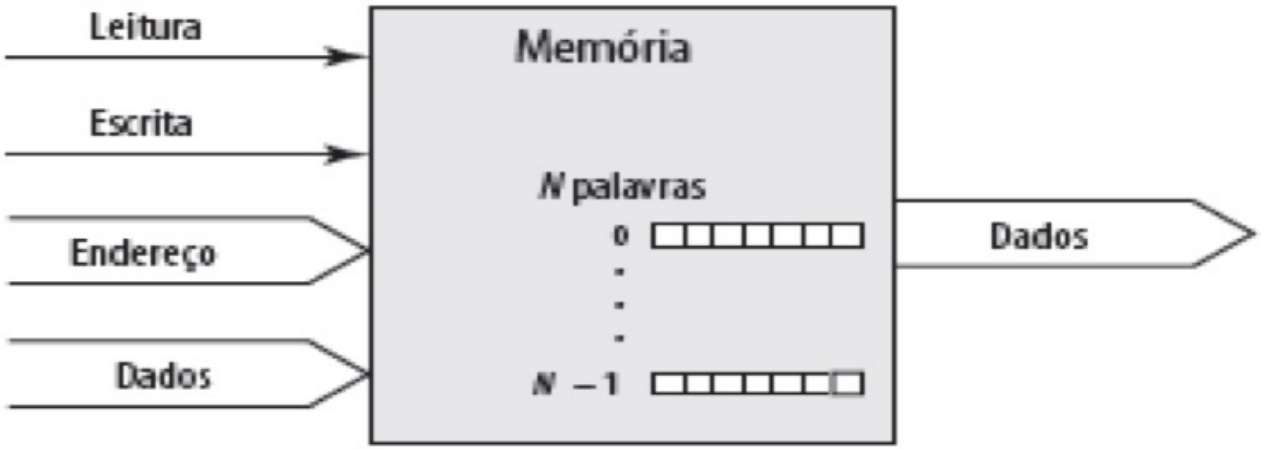
Linhas de comunicação - Módulos



Fonte: Arquitetura e Organização de Computadores – William Stallings

Conexão de memória

- Recebe e envia dados;
- Recebe endereços (de locais);
- Recebe sinais de controle:
  - ✓ Leitura;
  - ✓ Escrita;
  - ✓ Temporização;

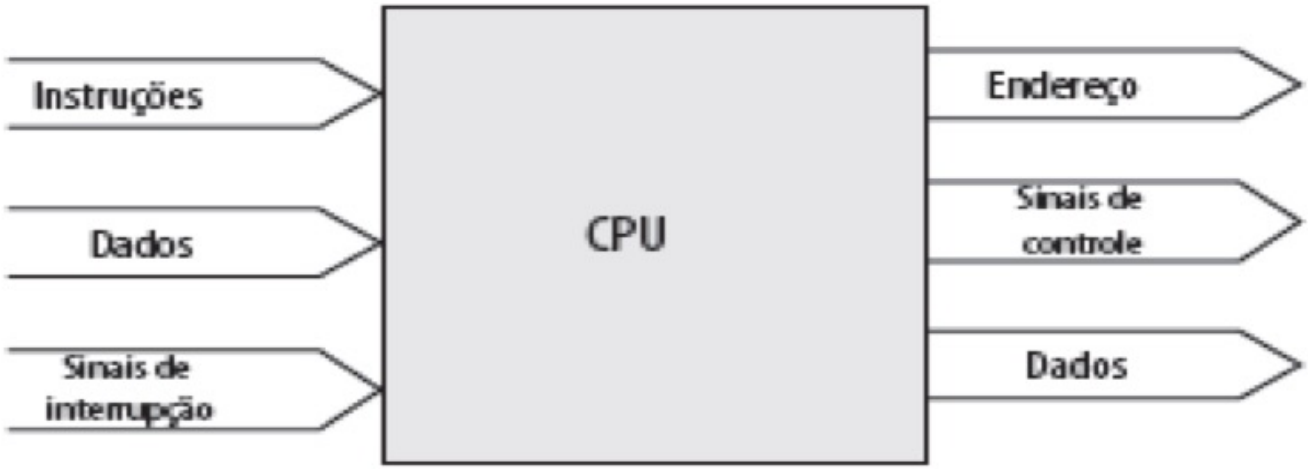


Conexão de entrada/saída

- Saída:
  - ✓ Recebe dados do computador.
  - ✓ Envia dados a periféricos.
- Entrada:
  - ✓ Recebe dados de periféricos.
  - ✓ Envia dados ao computador.

Conexão da UCP

- Lê instruções e dados;
- Escreve dados após processamento;
- Envia sinais de controle a outras unidades;
- Recebe e atua sobre interrupções.



Barramentos

- Para que todos os elementos estabeleçam comunicação são necessárias as **vias;**

- Este modelo de via é descrito pelo **barramento**;

- Existem diversos sistemas de interconexão possíveis;

- Há estruturas de barramento **único** e **múltiplo**;

Descrevendo um barramento

- Um caminho de comunicação conectando dois ou mais dispositivos;
- Normalmente, broadcast;
- Frequentemente agrupado;
  - ✓ Uma série de canais em um barramento;
  - ✓ Exemplo: barramento de dados de 32 bits são 32 canais de bits separados;
- Linhas de potência (energia) podem não ser mostradas.

Barramento de dados

- Transporta dados;

- Lembre-se de que não existe diferença entre **dados** e **instruções** neste nível;

- Largura é determinante para o desempenho.
  - ✓ 8, 16, 32, 64 bits.

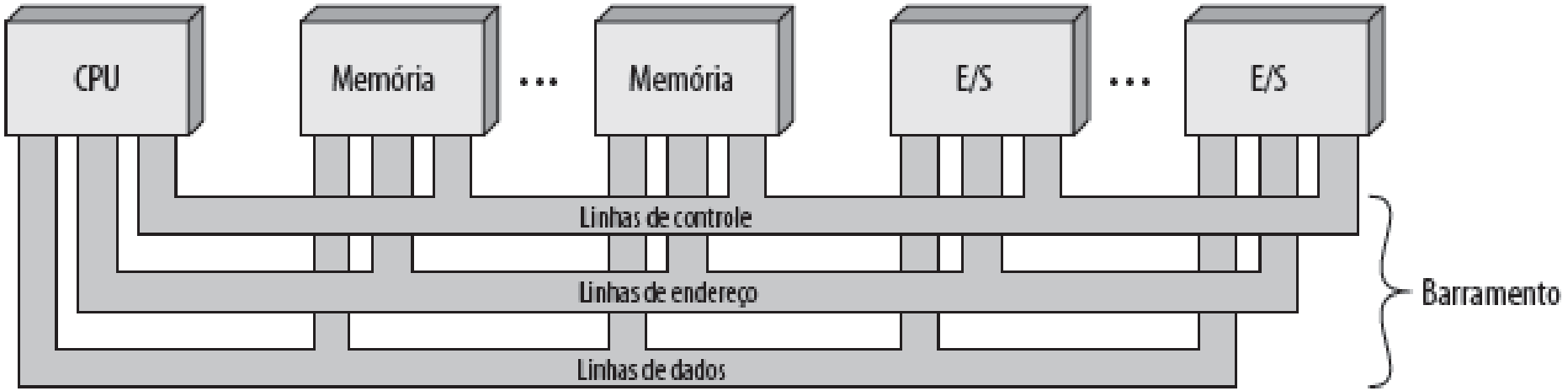
Barramento de endereço

- Identifica origem ou destino dos dados;
- Exemplo: UCP precisa ler uma instrução (dados) de determinado local na memória;
- Largura do barramento determina capacidade máxima da memória do sistema.

Barramento de controle

- Informações de controle e temporização:
  - ✓ Sinal de leitura/escrita de memória;
  - ✓ Solicitação de interrupção;
  - ✓ Sinais de clock;

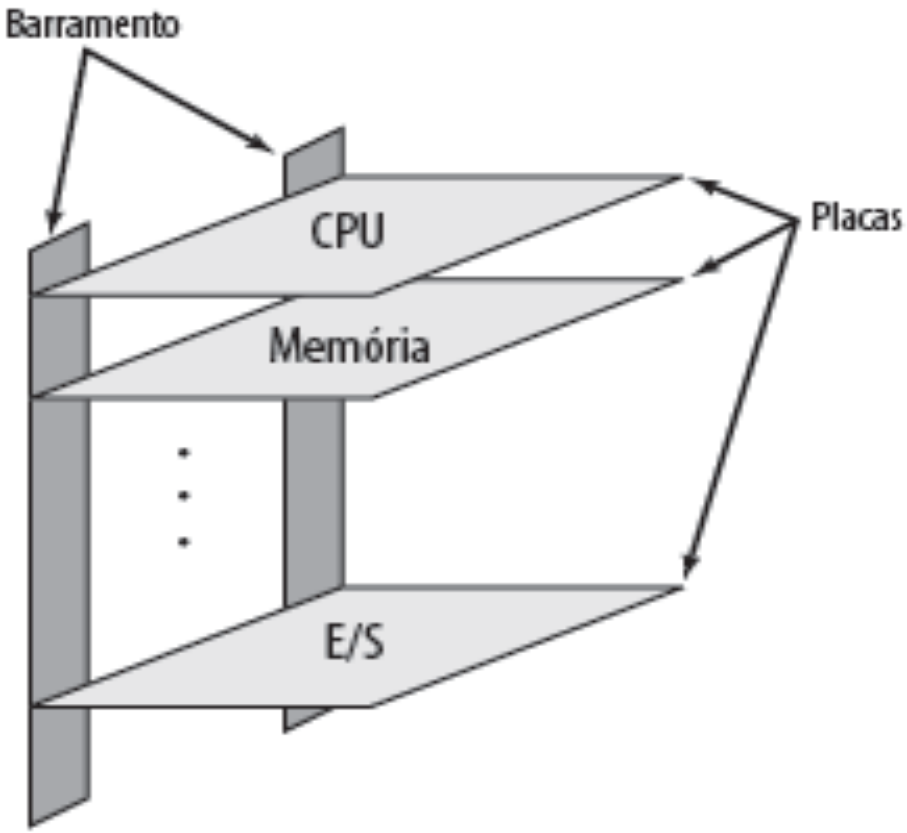
Interconexão de barramento - Modelo



Fonte: Arquitetura e Organização de Computadores – William Stallings

Arquitetura de barramento - fisicamente

- Linhas paralelas em placas de circuito;
- Cabos de fita;
- Conectores em tira nas placas mãe;
- Conjuntos de fios.



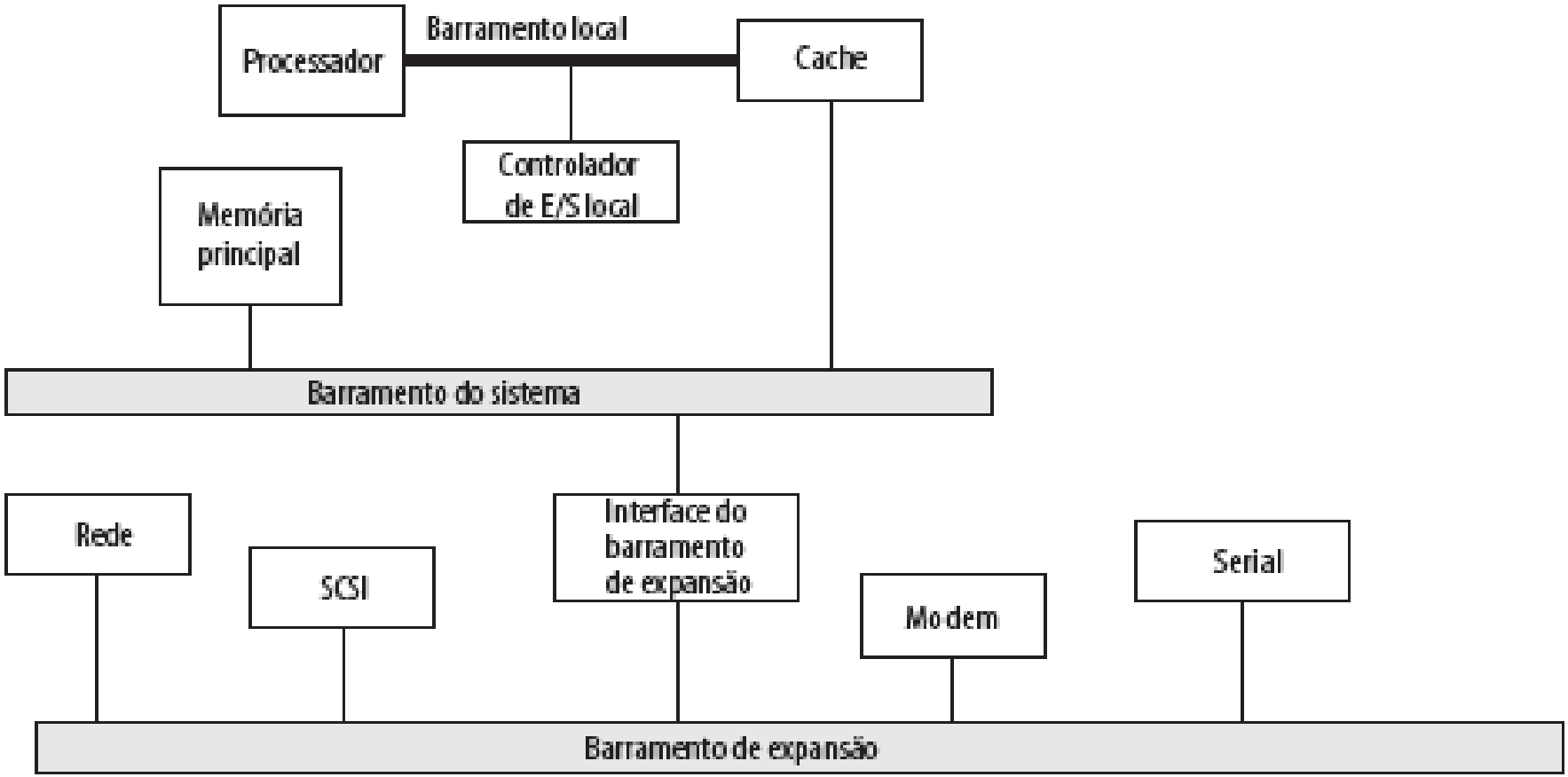
Fonte: Arquitetura e Organização de Computadores – William Stallings

Barramento único - Problema

- Muitos dispositivos em um barramento levam a:
  - ✓ Atrasos de propagação
    - ➢ Longos caminhos de dados significa que a coordenação do uso do barramento pode afetar contrariamente o desempenho;
    - ➢ Se a demanda de transferência de dados agregada se aproxima da capacidade do barramento;

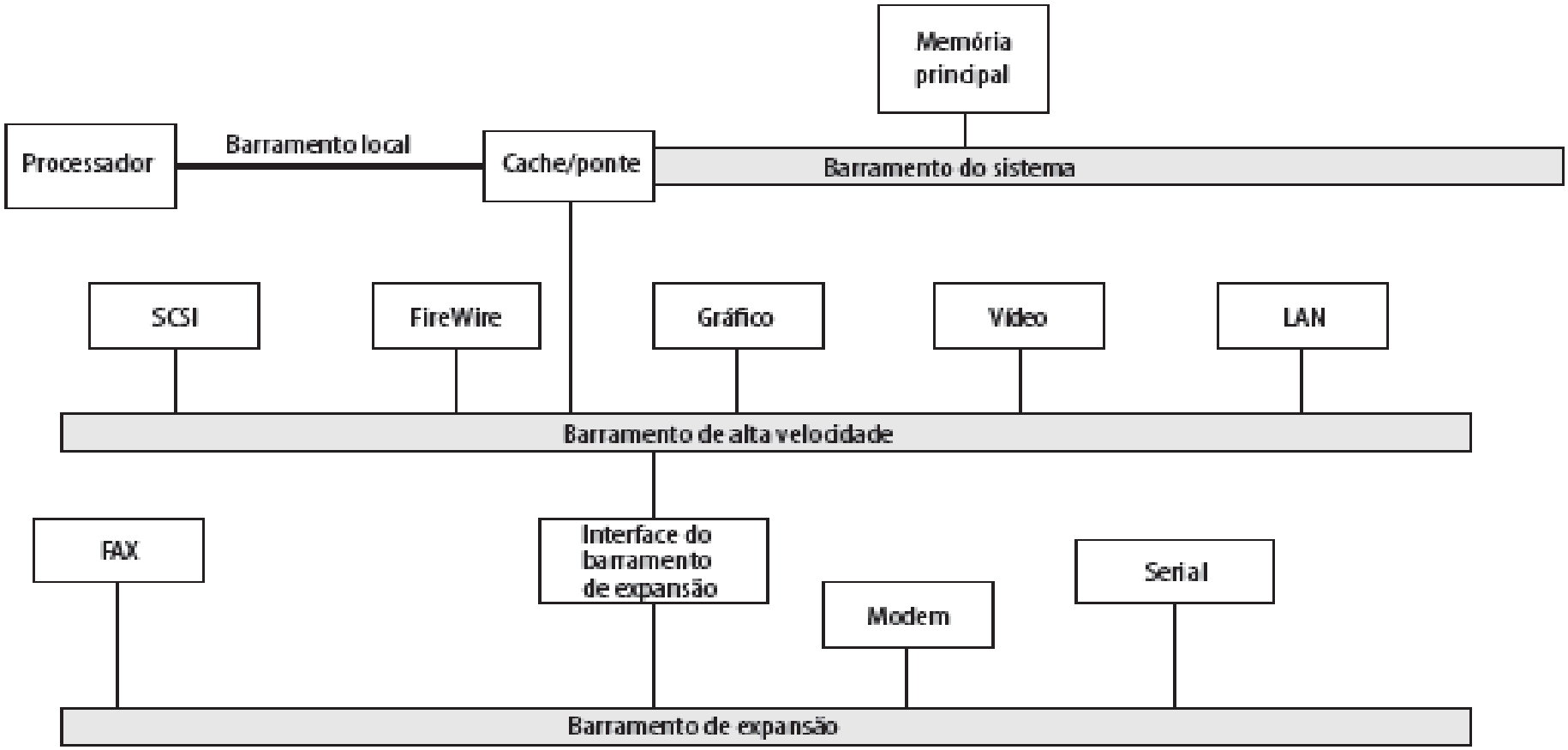
- A maioria dos sistemas utiliza **múltiplos barramentos** para contornar esses problemas.

Estrutura de barramento Convencional (com cache)



Fonte: Arquitetura e Organização de Computadores – William Stallings

Estrutura de Barramento – Alto desempenho



Fonte: Arquitetura e Organização de Computadores – William Stallings

Projeto de Barramentos

- Tipos de barramentos;
- Métodos de arbitração;
- Temporização;
- Largura do barramento.

Tipos de Barramentos

- Podem ser classificados em dois tipos: Dedicado e Multiplexado;

- **Dedicado**: Possui função fixa; Ex: Uso de linhas distintas para dados e para endereço;

- **Multiplexado**: Utiliza-se uma linha de controle de *Endereço Válido*; Cada módulo tem um tempo para copiar o endereço e analisar se os dados vão ser endereçados a ele.

Controle de Arbitração

- Seleciona qual módulo terá o controle do barramento;

- **Controle Centralizado:** um único dispositivo de *hardware* conhecido como ***controlador de barramento ou árbitro*** é responsável por alocar a cada módulo o tempo de utilização do barramento;

- **Controle Distribuído:** não existe árbitro; cada módulo possui uma lógica de controle de acesso nele mesmo.

Temporização

- Como os eventos em determinado barramento são coordenados;

- Para isso existe a transmissão **síncrona** e **assíncrona**.

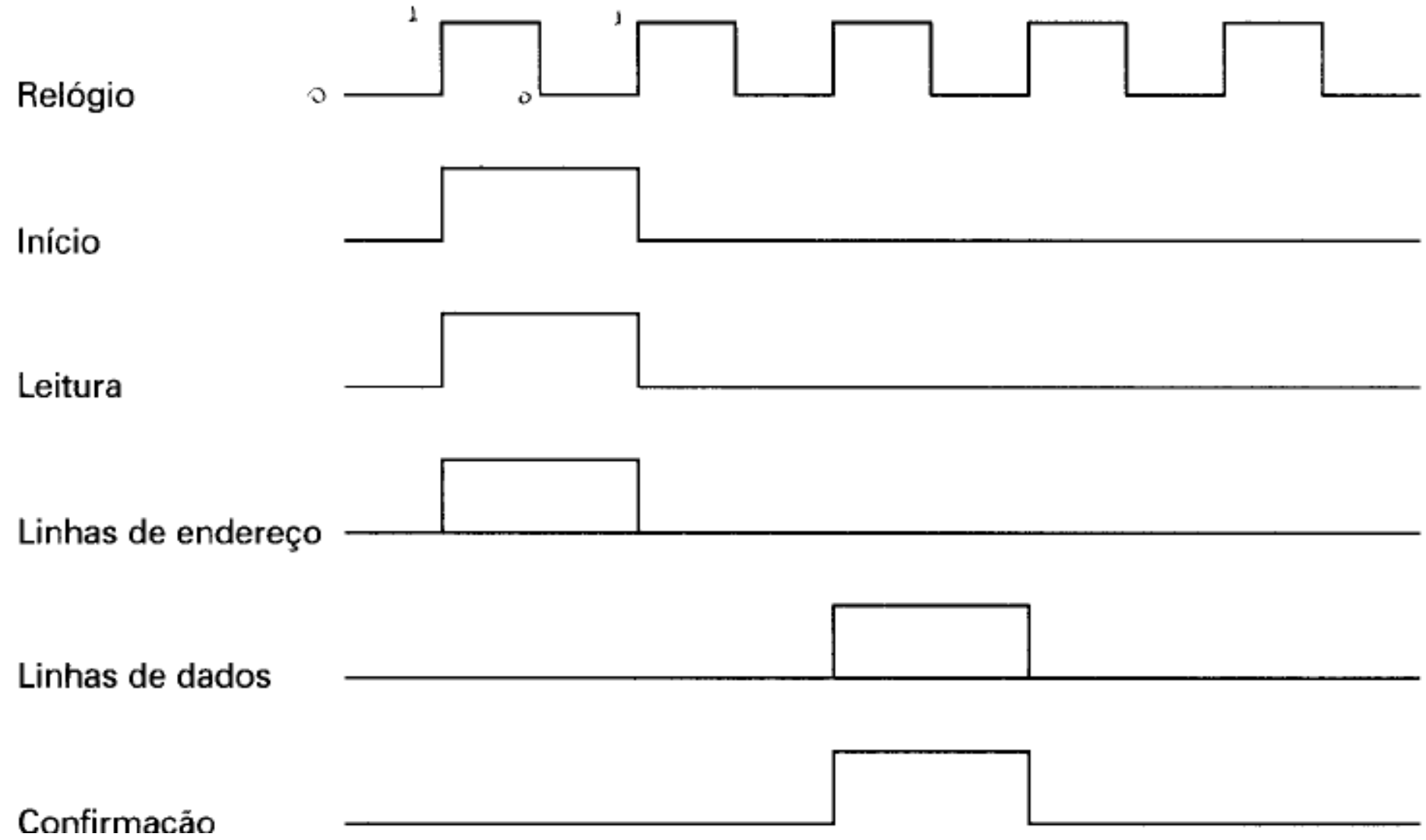
Transmissão Síncrona

- O barramento inclui uma linha de relógio que transmite sequências alternadas de 1s e 0s de igual duração;

- Cada transmissão dessa é denominada *Ciclo de relógio ou ciclo de barramento;*

- Todos os elementos conectados no barramento podem ler a linha do relógio e todos os eventos começam no início de um ciclo.

Transmissão Síncrona



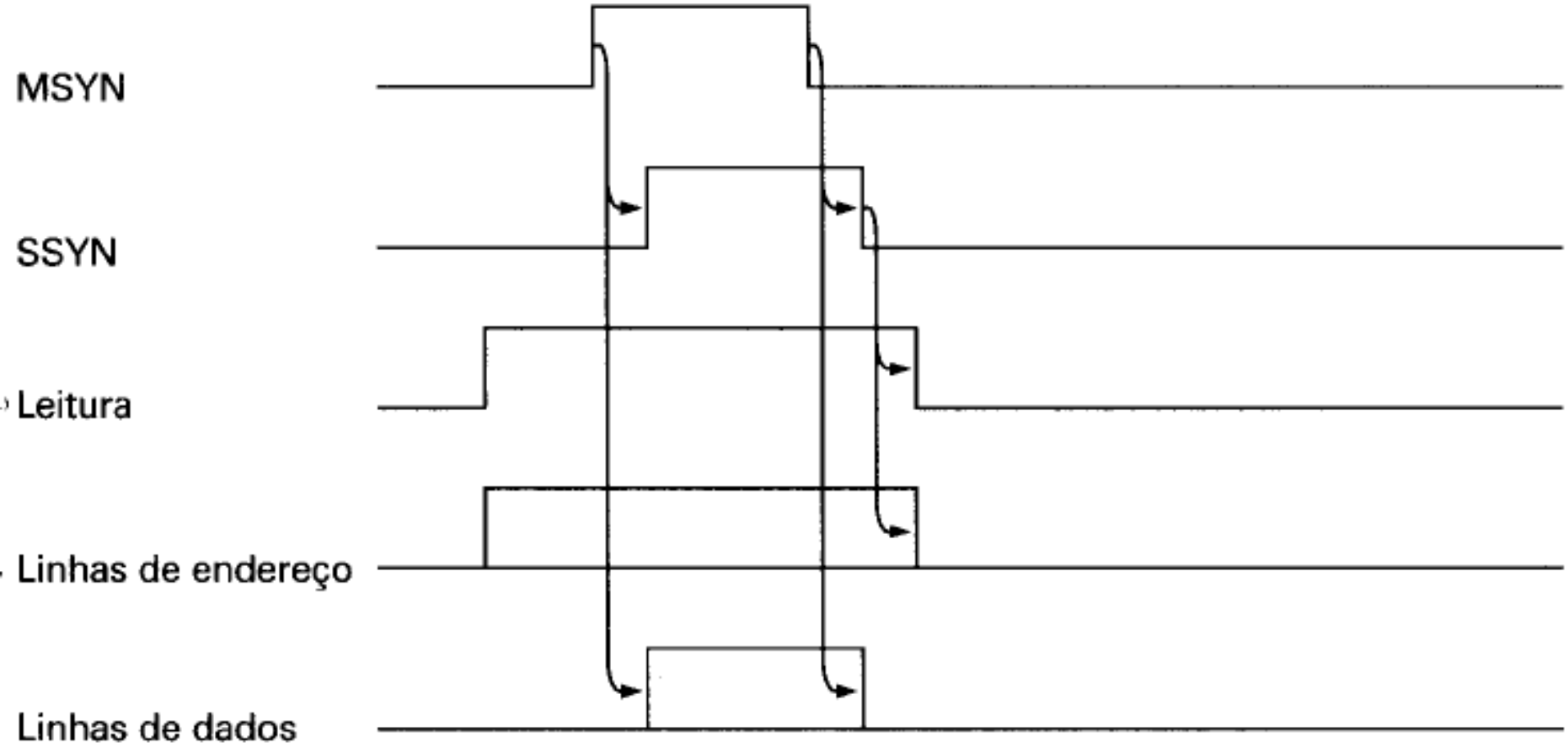
Transmissão Assíncrona

- O ocorrência de um evento depende do evento acontecido anteriormente;

- Mais simples e mais fácil de implementar e testar;

- Qualquer dispositivo, independente da sua velocidade, pode acessar o barramento.

Transmissão Assíncrona



Largura do Barramento

- **Largura do barramento de dados**: Quanto maior a largura do barramento de dados, maior o número de *bits* **transferidos** de cada vez (desempenho do sistema);

- **Largura do barramento de endereço**: Quanto maior a largura do barramento de endereço, maior o número de **posições** que podem ser endereçadas (capacidade do sistema).