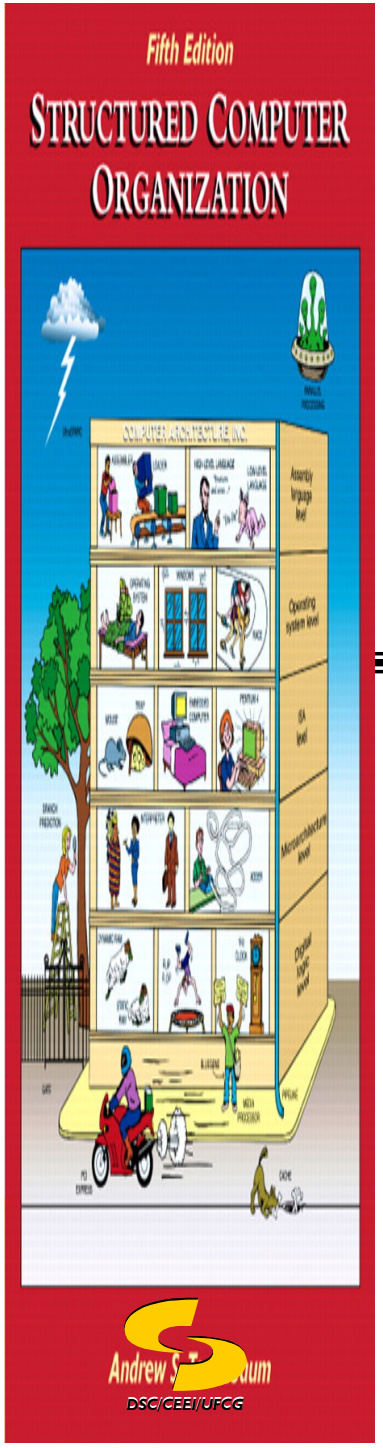


Universidade Federal de Campina Grande

Departamento de Sistemas e Computação

Curso de Bacharelado em Ciência da Computação

Organização e Arquitetura de Computadores I

Organização Básica de Computadores (Parte V, Complementar)

Profª Joseana Macêdo Fechine Régis de Araújo
joseana@computacao.ufcg.edu.br

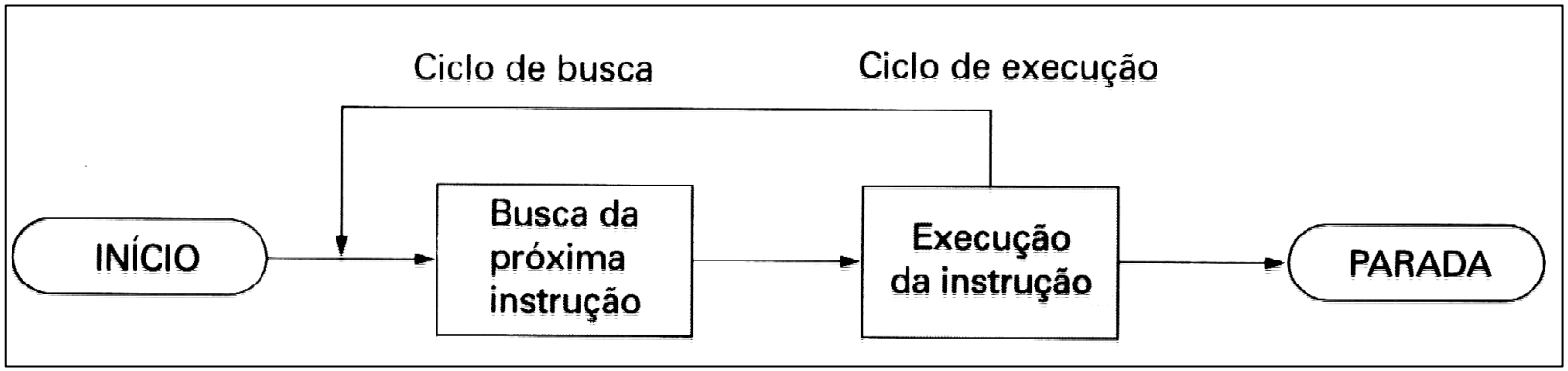
Carga Horária: 60 horas

Tópicos

- Organização Básica de Computadores
  - Conceitos Básicos Adicionais (Processadores)

Organização Básica de Computadores

Ciclo de Busca-Decodificação-Execução de Instrução:



Organização Básica de Computadores

Ciclo de Busca-Decodificação-Execução de Instrução:

1. Busca próxima instrução na memória e armazena no IR.
2. Atualiza o Contador de instrução, PC, para apontar para a próxima instrução
3. Determina tipo de instrução armazenada no IR
4. Determina endereço dos dados na memória, se a instrução requer dados adicionais.
5. Busca palavras (dados) na memória, caso a instrução precise, e as armazena em outros registradores.
6. Executa instrução.
7. Retorna ao passo 1.

Organização Básica de Computadores

Execução de Instruções

Decisão importante:

- Após especificar a linguagem de máquina L de um novo processador, a equipe de projetistas precisa decidir se devem construir um processador real, em hardware, para executar diretamente programas escritos em L, ou se deve escrever um interpretador para interpretar os programas escritos em L.

Organização Básica de Computadores

- **Problema 1**: Como construir computadores de baixo custo capazes de executar todas as instruções complexas de máquinas de alto desempenho, muito mais caras?

- **Problema 2**: O uso da interpretação permitiu a *criação de um conjunto grande de instruções* de importância discutível e que eram difíceis e caras para serem implementadas diretamente por hardware (circuitos muito complexos).

Organização Básica de Computadores

RISC *versus* CISC

CISC - *Complex Instruction Set Computer*

- Tecnologia mais antiga e usada para famílias de computadores compatíveis no nível do software.
- Número maior de instruções (~200 a 300 instruções).
- Uso extensivo de interpretação (principalmente para modelos mais baratos).

Organização Básica de Computadores

RISC *versus* CISC

RISC - *Reduced Instruction Set Computer*

- Processador com pequeno número de instruções muito simples. Primeiro: RISC I.
- Instruções capazes de serem executadas em um único ciclo do caminho de dados.

Organização Básica de Computadores

- ***Questão*: Porque então a tecnologia RISC não suplantou a CISC?**

- Problemas de compatibilidade com máquinas antigas com software já desenvolvido.

- Aparecimento de soluções híbridas: Por exemplo, a INTEL usa RISC para instruções de uso mais frequente (*Núcleo RISC*) e interpretação para instruções mais complexas e de uso menos frequente.

Organização Básica de Computadores

Princípios de Projeto para Computadores Modernos

- **Princípios do projeto RISC que os arquitetos de processadores de propósito geral devem seguir**:
  - Todas as instruções são diretamente executadas por hardware
  - Maximizar a Taxa à qual as instruções são executadas
  - As instruções precisam ser facilmente decodificadas
  - Somente as Instruções de *Load* e *Store* devem referenciar a Memória
  - Projetar uma máquina com muitos registradores (>= 32)

Organização Básica de Computadores

Princípios de Projeto para Computadores Modernos

- **Princípios do projeto RISC que os arquitetos de processadores de propósito geral devem seguir**:
  - Todas as instruções são diretamente executadas por hardware
  - Não existe o nível de microinstrução
  - Para máquina com filosofia CISC as instruções, em geral menos frequentes, que não existem em hardware são interpretadas

Organização Básica de Computadores

Princípios de Projeto para Computadores Modernos

- Princípios do projeto RISC que os arquitetos de processadores de propósito geral devem seguir:
  - Maximizar a Taxa à qual as instruções são executadas
    - **Uso de paralelismo**: execução de várias instruções lentas ao mesmo tempo
    - Execução de instruções não precisa seguir a lógica da programação

Organização Básica de Computadores

Princípios de Projeto para Computadores Modernos

- **Princípios do projeto RISC que os arquitetos de processadores de propósito geral devem seguir**:
  - As instruções precisam ser facilmente decodificadas
    - decodificação influencia na velocidade de execução das instruções
    - decodificação determina os recursos a serem usados na execução das instruções
    - quanto menor o número de formatos, mais fácil a decodificação

Organização Básica de Computadores

Princípios de Projeto para Computadores Modernos

- **Princípios do projeto RISC que os arquitetos de processadores de propósito geral devem seguir**:
  - Somente as Instruções de *Load* e *Store* devem referenciar a Memória
    - Acesso à memória é mais lento
    - Instruções que acessam a memória podem ser intercaladas com outras instruções

Organização Básica de Computadores

Princípios de Projeto para Computadores Modernos

- **Princípios do projeto RISC que os arquitetos de processadores de propósito geral devem seguir**:
  - Projetar uma máquina com muitos registradores (>= 32)
    - Palavras de memória devem permanecer nos registradores o maior tempo possível
    - Falta de registradores pode obrigar a buscar várias vezes a mesma palavra da memória

Organização Básica de Computadores

Princípios de Projeto para Computadores Modernos

- **Princípios do projeto RISC que os arquitetos de processadores de propósito geral devem seguir**:
  - Solução para aumentar a velocidade do processador: Uso de **paralelismo**.
    - **no nível das instruções**: um único processador deve executar mais instruções por segundo
    - **no nível do processador**: vários processadores trabalhando juntos na solução do mesmo problema

Organização Básica de Computadores

Princípios de Projeto para Computadores Modernos

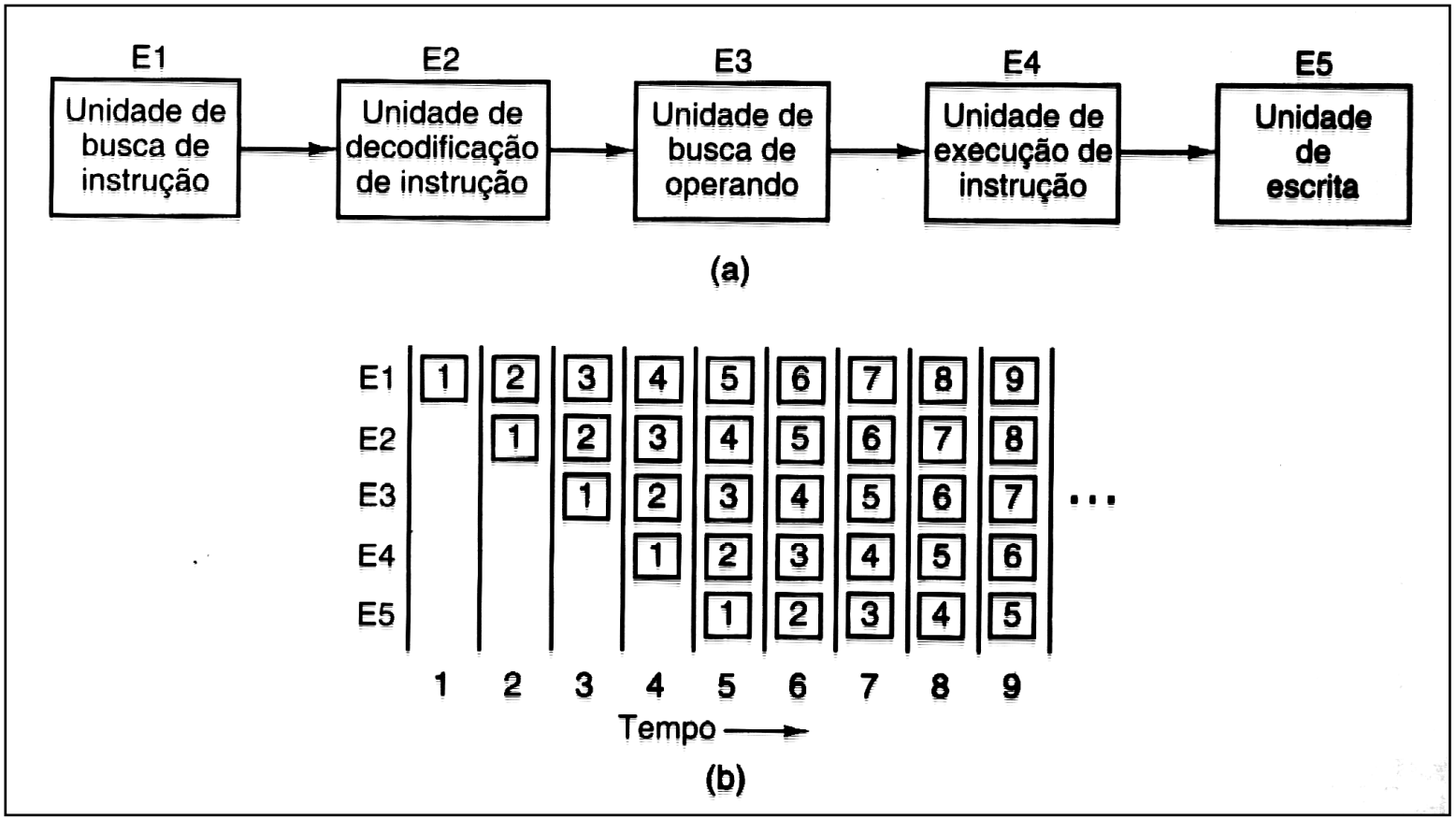
Paralelismo no Nível das Instruções

- Maior gargalo para a velocidade de execução de instruções é o acesso à memória

- **Execução em Pipeline**
  - O processamento em pipeline divide a execução de instruções em várias partes, cada uma das quais tratada por um hardware dedicado exclusivamente a ela.

Organização Básica de Computadores

(a) Pipeline de 5 estágios. (b) Estado de cada um dos estágios em função do tempo (estão ilustrados 9 períodos do clock).



Organização Básica de Computadores

Paralelismo no Nível das Instruções

- Funcionamento de um pipeline de 5 estágios
  - O estágio 1 busca a instrução da memória e armazena num buffer até chegar a hora de executá-la
  - No estágio 2 ocorre a decodificação da instrução, determinando tipo e operandos
  - No estágio 3 ocorre a busca dos operandos na memória ou nos registradores
  - No estágio 4 tem-se a execução - passagem pelo caminho de dados
  - No estágio 5 o resultado do processamento é escrito num registrador

Organização Básica de Computadores

Paralelismo no Nível das Instruções

- A idéia básica do pipeline é a mesma de uma linha de produção em série. Vários processamentos estão sendo executados ao mesmo tempo.
- A figura mostra o funcionamento do pipeline, mostrando que os estágios de cada processamento são aplicados a várias instruções ao mesmo tempo.
  - Exemplo: no tempo 1 a instrução 1 está sendo lida, no tempo 2 a instrução 1 está sendo decodificada enquanto que a instrução 2 está sendo lida, no tempo 3 a instrução 1 está buscando dados, a instrução 2 está sendo decodificada e a instrução 3 está sendo lida, e assim por diante.

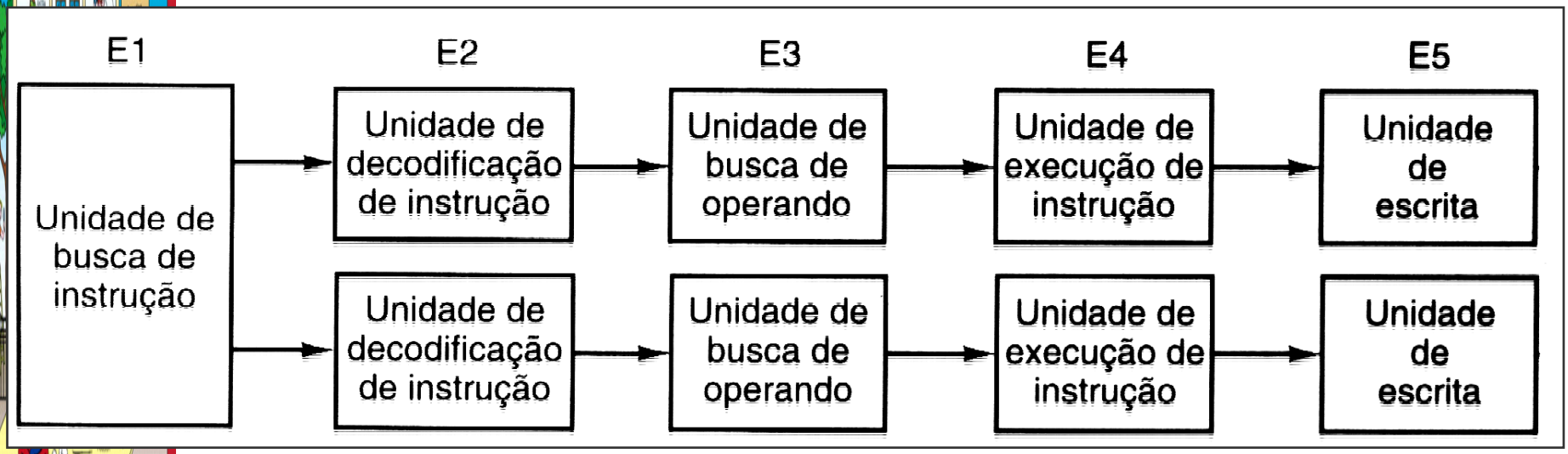
Organização Básica de Computadores

Paralelismo no Nível das Instruções

- **Arquiteturas Superescalares**
  - Se um pipeline é bom, com certeza dois serão ainda melhor.
  - Neste caso, uma única unidade de busca de instruções lê 2 instruções e coloca cada uma em 1 pipeline.

Organização Básica de Computadores

Dois pipelines de 5 estágios com uma unidade de busca de instruções comum a ambos.



Organização Básica de Computadores

Paralelismo no Nível das Instruções

- **Arquiteturas Superescalares**

- A execução das instruções é feita em paralelo e:
  - não pode haver conflitos pelo uso de recursos (mesmo registro, por exemplo)
  - o resultado de uma instrução não pode depender do resultado da outra
  - pode se pensar em pipelines com leitura inicial de 3 ou mais instruções, porém o hardware fica complexo.

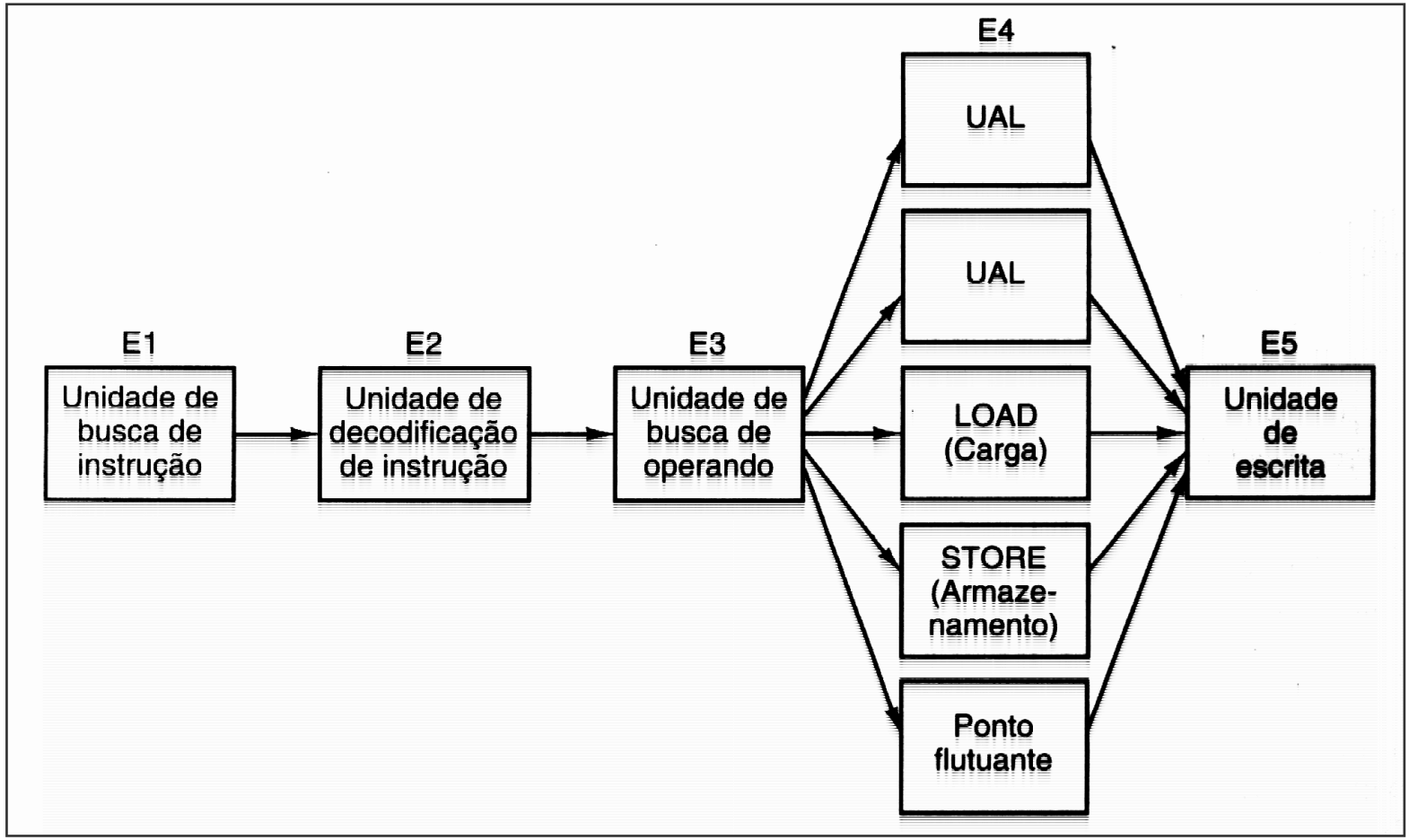
Organização Básica de Computadores

Paralelismo no Nível das Instruções

- **Arquiteturas Superescalares**
- Máquinas de alto desempenho usam outra metodologia:
  - A idéia básica é ter um único pipeline, com diversas unidades funcionais
  - O estágio 3 pode distribuir instruções a uma velocidade consideravelmente mais alta do que o estágio 4 pode executá-las. Este estágio usa vários dispositivos de hardware (inclusive mais do que uma UAL) para acelerar o processamento neste estágio.

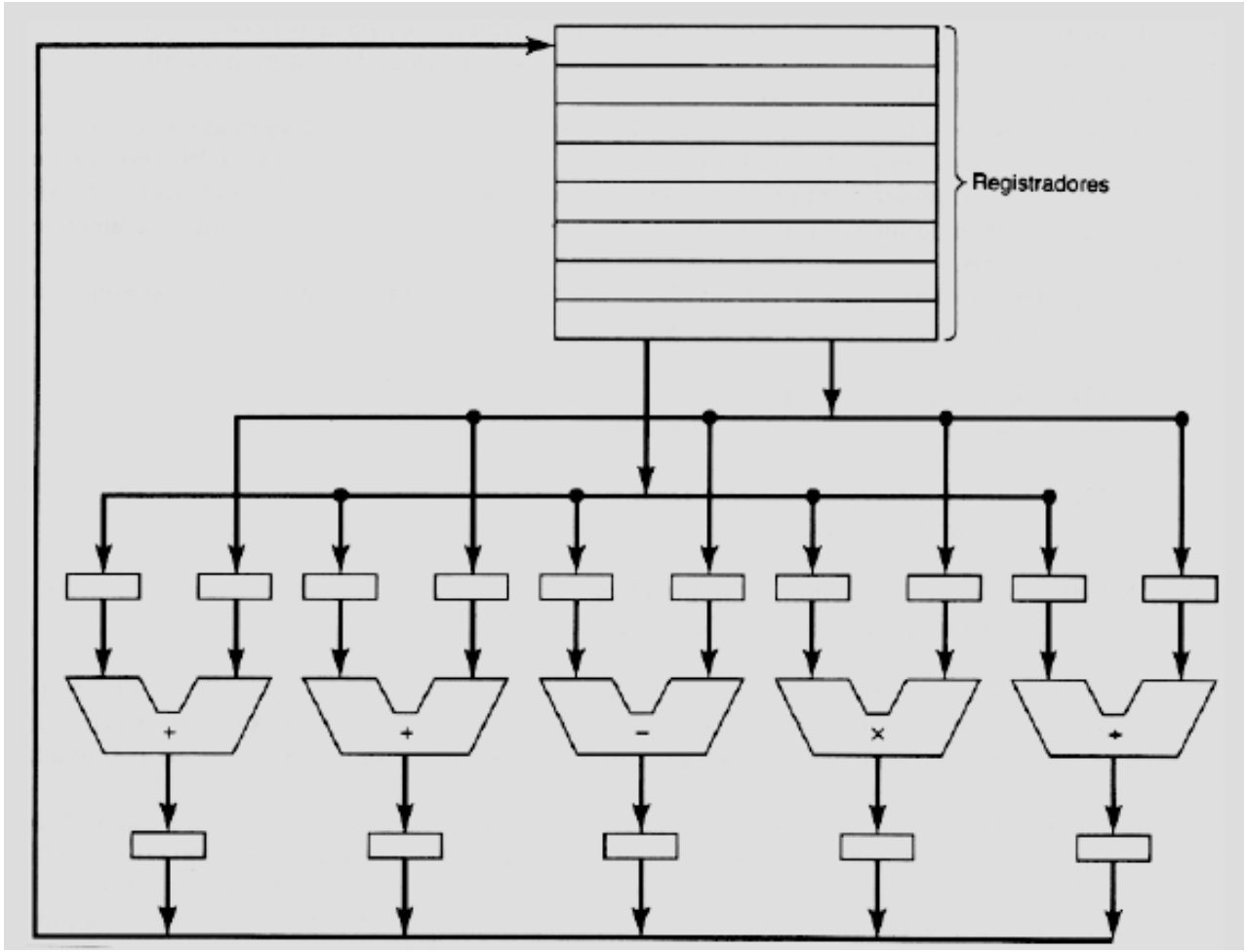
Organização Básica de Computadores

Processador superescalar com 5 unidades funcionais



Organização Básica de Computadores

Uma CPU com três unidades funcionais que podem trabalhar em paralelo.



Organização Básica de Computadores

- **Informações adicionais**:

Processador	Número de pipelines
486	5
Pentium II	10
Athlon Thunderbird	12
Pentium 4	>20

Organização Básica de Computadores

Paralelismo no Nível do Processador

- A medida que os processadores vão ficando mais rápidos:
  - aparecem limitações de ordem física (velocidade da luz em fios de cobre ou fibras ópticas)
  - maior produção de calor pelo chip (problema para dissipar essa energia)
- Operação do processador em pipeline ou em superescalar possibilita ganhos de 5 a ~10 vezes.
- Para ganhos maiores, 50-100 ou mais vezes, deve-se projetar computador com mais de 1 processador.

Organização Básica de Computadores

Computadores Matriciais (2 implementações)

- **Processador matricial**
  - Composto de grande número de processadores idênticos
  - Cada processador executa a mesma sequência de instruções sobre diferentes conjuntos de dados
  - Tem uma única unidade de controle
  - Tem uma ULA para cada processador

Problema: Os processadores matriciais não são independentes pois compartilham uma única UC.

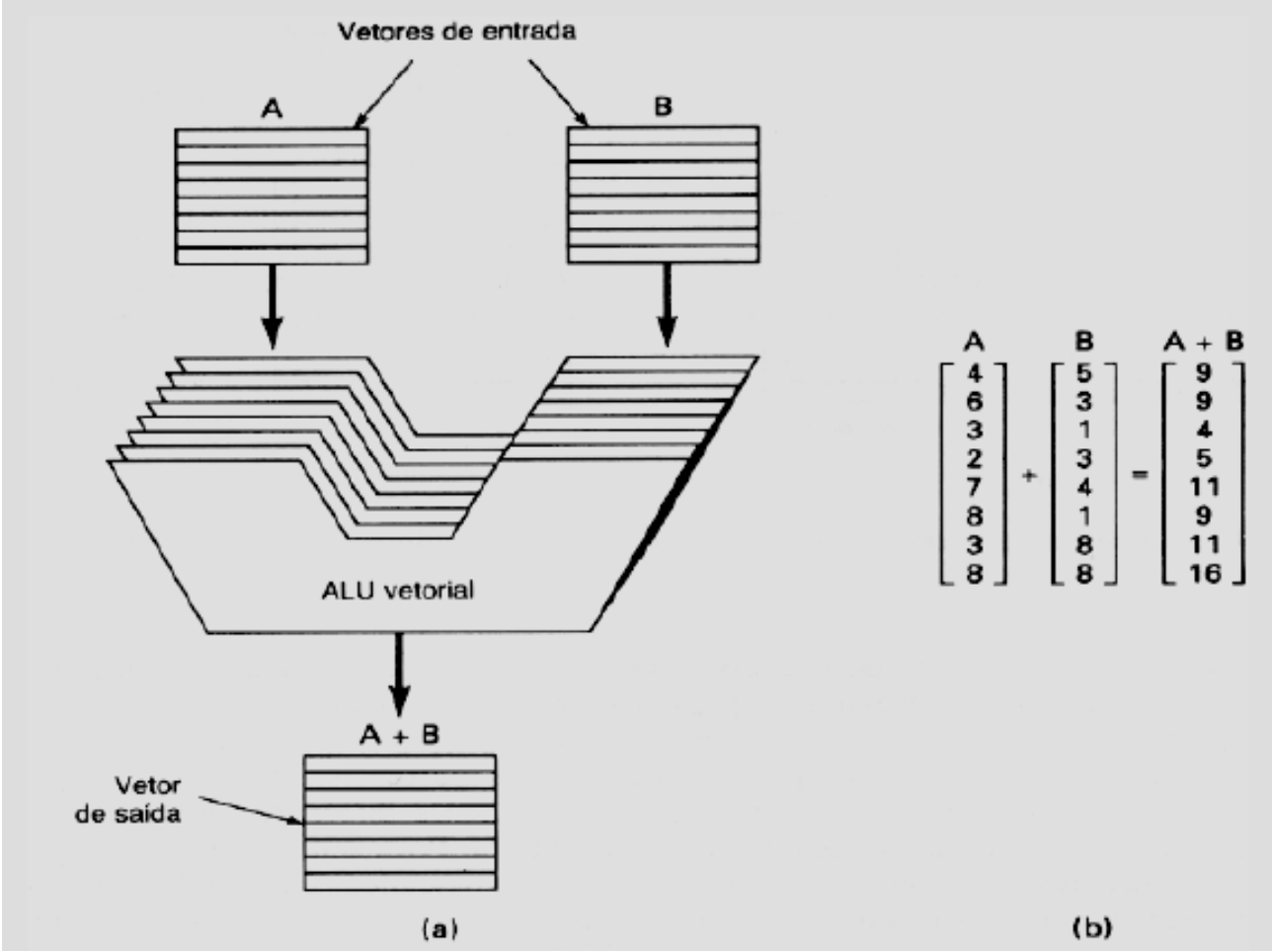
Organização Básica de Computadores

Computadores Matriciais (2 implementações)

- **Processador vetorial**
  - muito parecido com processador matricial
  - operações aritméticas são executadas numa única UAL, que opera em pipeline
  - operandos são colocados em um *registro vetorial* para serem processados na ULA

Organização Básica de Computadores

(a) Uma ULA vetorial. (b) Um exemplo de soma de vetores.



Organização Básica de Computadores

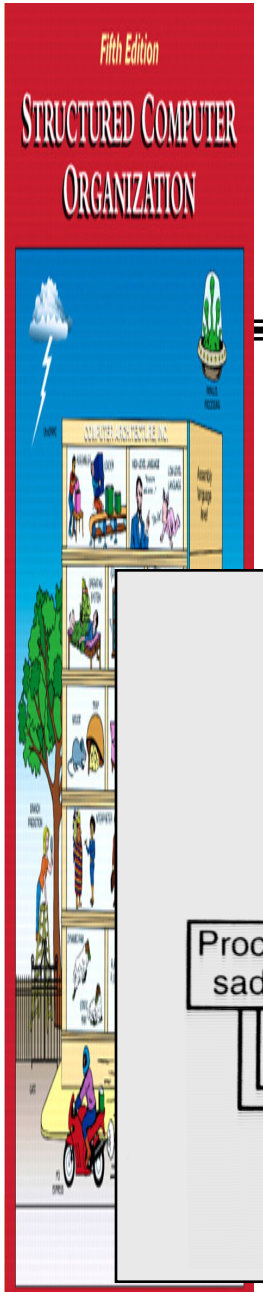
Computadores Matriciais (2 implementações)

- **Processador matricial x vetorial**
  - programação para o matricial voltada ao paralelismo (mais difícil)
  - processador matricial é, em geral, mais rápido principalmente para repetição de um mesmo processamento em vários "pedaços" dos dados
  - processador vetorial se adapta a processamentos paralelos e não paralelos
  - hardware do matricial é mais caro (muitas ULA)

Organização Básica de Computadores

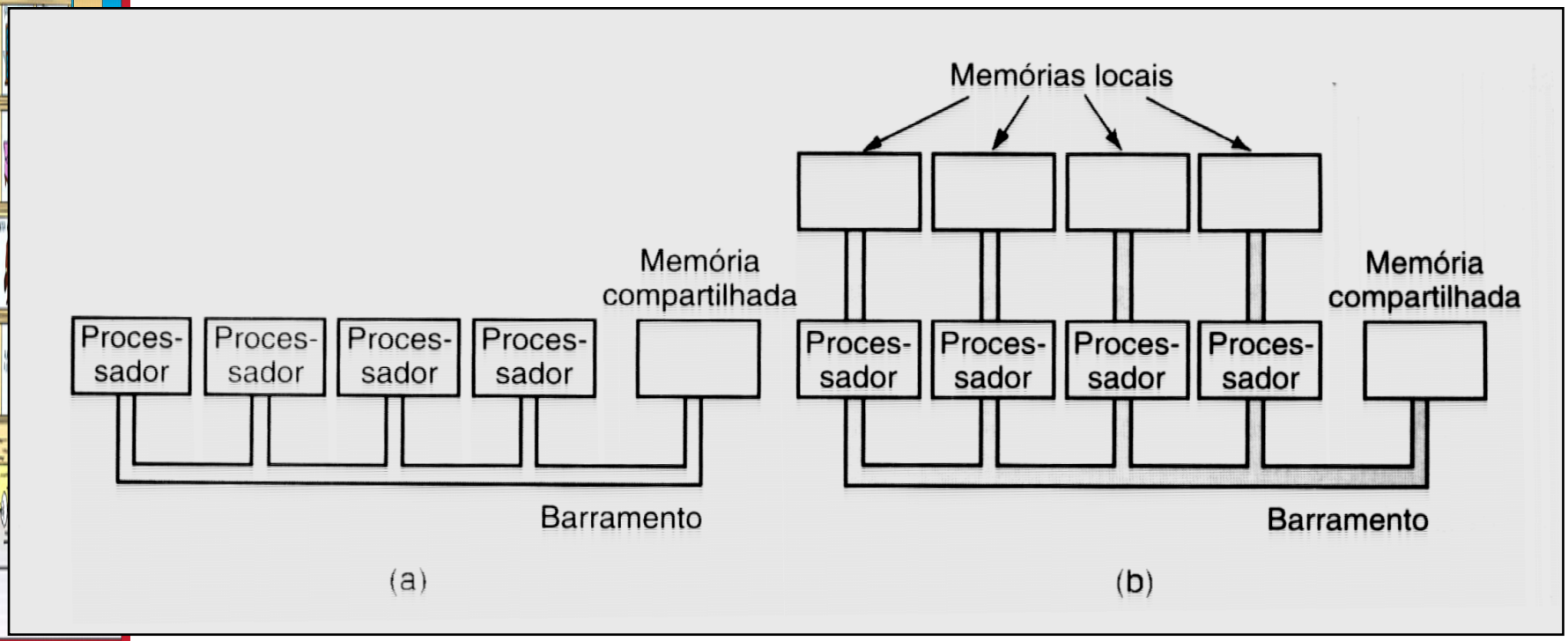
Multiprocessadores

- é composto de vários processadores independentes
- compartilham uma mesma memória por um barramento principal
- ou compartilham uma memória e tem memórias locais
- executam processamentos locais
- liberam tráfego do barramento principal
- é necessário gerenciar conflitos



Organização Básica de Computadores

(a) Multiprocessadores organizados em torno de um único barramento. (b) Multiprocessadores com memórias locais.



Organização Básica de Computadores

Multicomputadores

- Sistemas com um grande número de computadores interconectados
- Não existe nenhum tipo de memória comum sendo compartilhada
- Comunicação entre computadores é feita por meio de troca de mensagens a uma velocidade bem alta
- Computador não precisa estar ligado diretamente com todos os outros (uso de topologias em árvore, anéis, etc..)
- Mensagens são roteadas do computador fonte para o destino (usando computadores intermediários)
- Existem em operação sistemas multicomputadores com mais de 10000 computadores

Organização Básica de Computadores

Conceitos Relevantes

- ***Single Instruction Single Data (SISD) stream***: um único fluxo de instruções opera sobre um único fluxo de dados.
  - Processamento sequencial
  - Apesar dos programas estarem organizados a partir de instruções sequenciais, elas podem ser executadas de forma sobreposta em diferentes estágios (*pipelining*).
- ***Single Instruction Multiple Data (SIMD) stream***: corresponde ao processamento de vários dados sob o comando de apenas uma instrução.
  - O programa ainda segue uma organização sequencial.
  - Nesta classe estão os processadores vetoriais e matriciais.

Organização Básica de Computadores

Conceitos Relevantes

- ***Multiple Instruction Single Data (MISD) stream***: múltiplas unidades de controle executando instruções distintas que operam sobre o mesmo dado.
  - Não representa nenhum paradigma de programação existente e é impraticável tecnologicamente
- ***Multiple Instruction Multiple Data (MIMD) stream***: esta classe é bastante genérica envolvendo o processamento de múltiplos dados por parte de múltiplas instruções.
  - Qualquer grupo de máquinas operando como uma unidade (deve haver um certo grau de interação entre as máquinas) enquadra-se como MIMD.
  - Representantes desta categoria: servidores multiprocessados, as redes de estações e as arquiteturas massivamente paralelas.

Organização Básica de Computadores

Importante:

Como sistemas multiprocessadores são mais fáceis de programar e sistemas multicomputadores são mais fáceis de construir, existem *sistemas híbridos*. Tais computadores dão a ilusão de compartilhamento de memória, sem arcar com o ônus de implementá-lo diretamente.

Problema: Sistemas com muitos processadores (>64) são de difícil implementação. Dificuldade está na conexão dos processadores à memória.