

Aula 10

Professores:

Lúcia M. A. Drummond
Simone de Lima Martins

Conteúdo:

Arquiteturas Avançadas

- Arquiteturas RISC
- Processamento Paralelo

Arquiteturas RISC

- Reduced Instruction Set Computer se contrapõe à arquiteturas até então predominantes - CISC - Complex Instruction Set Computer

Arquiteturas RISC

- Desenvolvimento por três caminhos:
 - Projeto da IBM, desenvolvido em meados da década de 1970, sem sucesso comercial. A IBM ganhou mercado por volta de 1990 com os processadores RS/6000 e posteriormente com a família POWER PC, desenvolvida em conjunto com a Motorola e com a Apple.
 - Estudos em Stanford, por John Hennessy, que redundaram nos processadores Mips.
 - Estudos em Berkley, por David Patterson, que redundaram em processadores desenvolvidos pela Sun

Arquiteturas RISC

- Alguns estudos verificaram:
 - Linguagem de alto nível com comandos poderosos para facilitar a vida dos programadores
 - Gap semântico: separação acentuada entre operações de linguagem de alto nível e em linguagem de máquina
 - Compiladores complexos
- Soluções:
 - Aumentar a quantidade de instruções
 - Incluir mais modos de endereçamento
 - Utilizar mais microprogramação

CISC

Arquiteturas RISC

- Vários pesquisadores realizaram estudos sobre comportamento do programa. Em 1982, Patterson fez uma pesquisa analisando programas científicos, de emprego geral e de editoração.

Comando	Ocorrência		Peso nas inst. máq.		Peso em ref. à MP	
	Pascal	C	Pascal	C	Pascal	C
Assign	45%	38%	23%	13%	14%	15%
Loop	5%	3%	42%	32%	33%	26%
Call	15%	12%	31%	33%	44%	45%
IF	29%	43%	11%	21%	7%	13%
Goto	—	3%	—	—	—	—
Outros	6%	1%	3%	1%	2%	1%

(Fig. 11.2 do livro texto)

Arquiteturas RISC

- Outros estudos chegaram a mesma conclusão:
 - Necessidade de aperfeiçoar o Hw para atender à demanda de recursos
 - Não eram necessárias tantas instruções de máquina, se apenas algumas delas eram utilizadas na maioria dos programas
 - Muitas instruções significa muitos bits em cada código de operação - instrução com maior comprimento, mais tempo de interpretação
 - Desenvolvimento de arquiteturas **RISC**

Arquiteturas RISC

Sistemas	Tipo	Ano	Qtde. inst.	Qtde. reg.	Tamanho inst.
IBM /370-168	CISC	1973	208	16	16-48 bits
Intel 80486	CISC	1989	147	8	1-17 bits
Intel Pentium	CISC	1993	150	8	1-17 bits
Power PC 601	RISC	1993	184	32-I 32-PF	32 bits
Sparc 10	RISC	1987	52	até 528	32 bits
Apha 21064	RISC	1992	125	32-I 32-PF	32 bits

(Fig. 11.1 do livro texto)

Arquiteturas RISC - Características

- Menor quantidade de instruções e tamanho fixo
- Execução otimizada, o sistema deve produzir resultados com melhor desempenho, mesmo considerando que uma menor quantidade de instruções vá conduzir a programas mais longos
- Facilidade na busca e incremento do CI

Arquiteturas RISC - Características

- Execução otimizada de chamada de funções
- Utilização de registradores para passagem de parâmetros e recuperação dos dados
- Enquanto em arquiteturas CISC eram necessárias leitura e escrita na memória principal
- RISC: Mais registradores já que ocorre a redução de circuitos para decodificação e execução de instruções

Arquiteturas RISC - Características

- Menor quantidade de modos de endereçamento
- Apenas dois tipos de instrução para acesso à memória: LOAD/STORE (utilizando modo direto)
- Redução de ciclos de relógio para execução das demais instruções

Arquiteturas RISC - Características

- Modo de execução com *Pipelining*
 - Uso altamente produtivo do pipeline
 - Instruções de formatos simples e únicos tiram maior proveito do pipeline - estágios consomem o mesmo tempo

Arquiteturas RISC - Características

Característica	Considerações
Menor quantidade de instruções que as máquinas CISC	<ul style="list-style-type: none"> • Simplifica o processamento de cada instrução e torna este item mais eficaz. • Embora o processador RS/600 possua 184 instruções, ainda assim é bem menos que as 303 instruções dos sistemas VAX-11. Além disso, a maioria das instruções é realizada em 1 ciclo de relógio, o que é considerado o objetivo maior dessa arquitetura.
Execução otimizada de chamada de funções	<ul style="list-style-type: none"> • As máquinas RISC utilizam os registradores da UCP (em maior quantidade que os processadores CISC) para armazenar parâmetros e variáveis em chamadas de rotinas e funções. Os processadores CISC usam mais a memória para a tarefa.
Menor quantidade de modos de endereçamento	<ul style="list-style-type: none"> • As instruções de processadores RISC são basicamente do tipo Load/Store, desvio e de operações aritméticas e lógicas, reduzindo com isso seu tamanho. • A grande quantidade de modos de endereçamento das instruções de processadores CISC aumenta o tempo de execução das mesmas.
Utilização em larga escala de <i>pipelining</i>	<ul style="list-style-type: none"> • Um dos fatores principais que permite aos processadores RISC atingir seu objetivo de completar a execução de uma instrução pelo menos a cada ciclo de relógio é o emprego de <i>pipelining</i> em larga escala.

(Fig. 11.3 do livro texto)

Arquiteturas RISC - Medidas de Desempenho

- MIPS - milhões de instruções por segundo. Não é uma boa medida para comparar RISC com CISC. RISC mais instruções simples.
- MFLOPS - milhões de operações de ponto-flutuante por segundo. Unidade mais apropriada para medir a velocidade com cálculos matemáticos: programas científicos, cálculos meteorológicos. Os programas usados para teste são escritos em FORTRAN.
- SPECmark - (system performance evaluation committee) - 10 programas - 6 com cálculos matemáticos e 4 com operações de inteiros e caracteres (escritos em C e Fortran). Medidas em SPECmark, composição dos resultados.

Arquiteturas RISC - Observações RISC x CISC

- CISC:
 - Menos instruções produz código-objeto menor?
Não, necessariamente, as instruções podem ser maiores
 - Com menos instruções o programa executa mais rapidamente?
Não, necessariamente, o tempo de execução de cada instrução pode ser grande

Arquiteturas RISC - Observações RISC x CISC

- RISC:
 - Instruções possuem código de operação com menor quantidade de bits - tempo de decodificação menor
 - Instruções executadas diretamente pelo hardware e não por um microprograma - execução mais rápida

Processamento Paralelo

Computador: máquina seqüencial

- Algoritmo: seqüência de instruções
- Processadores: executam instruções seqüencialmente
- Instruções: executada por seqüência de operações (busca, execução, armazenamento)

Visão não totalmente verdadeira!

Processamento Paralelo

- Nível de microoperações: vários sinais de controle gerados ao mesmo tempo
- Técnica de pipeline: sobreposição de etapas de execução
- Máquinas superescalares: diversas unidades de execução em um mesmo processador, que podem executar várias instruções de um mesmo programa em paralelo

Processamento Paralelo

Outras oportunidades de exploração de paralelismo para melhorar o desempenho:

- Multiprocessadores simétricos: múltiplos processadores compartilhando memória
- Clusters: diversos computadores independentes, conectados entre si, organizados de forma cooperativa
- Máquinas de acesso não uniforme à memória (NUMA): multiprocessador no qual o tempo gasto para acesso à memória varia de acordo com a posição da palavra na memória
- Máquinas vetoriais: otimizam a ULA para processamento de vetores de ponto-flutuante

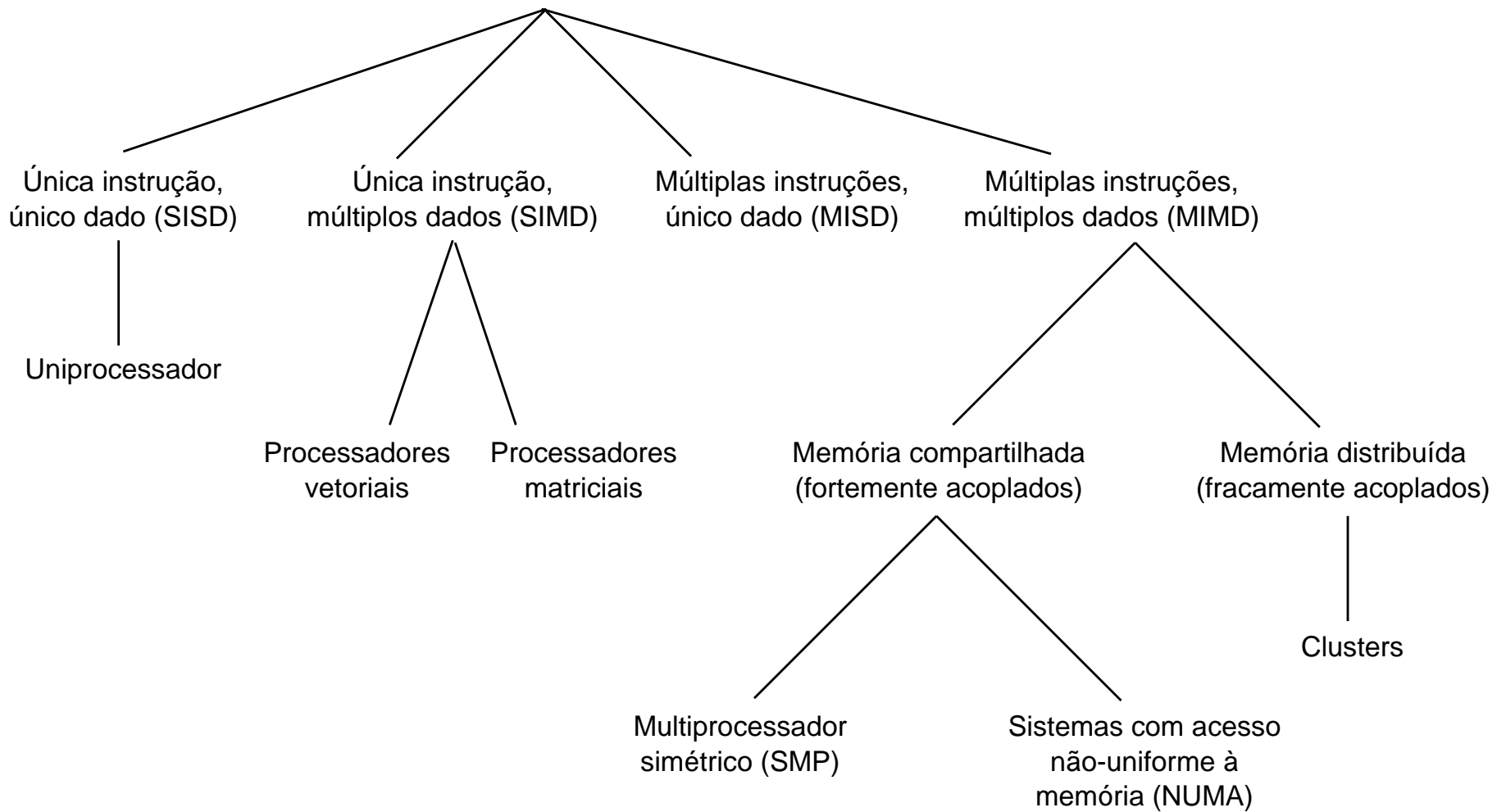
Processamento Paralelo

Taxonomia de Flynn (1972)

- SISD - único processador executa uma única seqüência de instrução sobre dados armazenados em uma única memória.
- SIMD - vários elementos de processamento. Cada um tem uma memória de dados. Cada instrução é executada sobre um conjunto de dados diferente. Processadores vetoriais e matriciais.
- MISD - seqüência de dados é transmitida para um conjunto de processadores, cada um dos quais executa uma seqüência de instruções diferente. Nunca foi implementado.
- MIMD - conjunto de processadores executa simultaneamente seqüências diferentes de instruções. SMPs, clusters, sistemas NUMA.

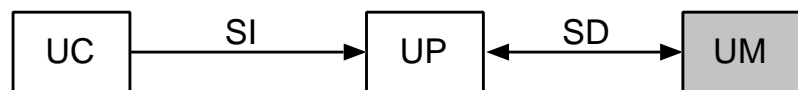
Processamento Paralelo

Organizações de processadores

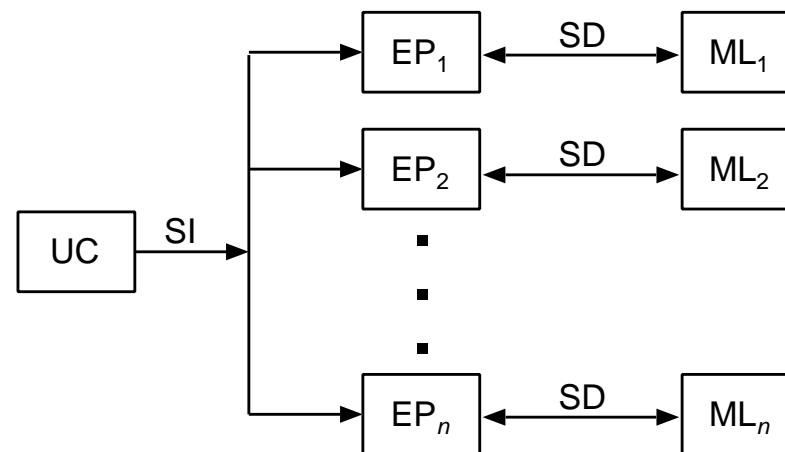


(Fig. 16.1 do livro de Arquitetura e Organização de Computadores" , William Stalling)

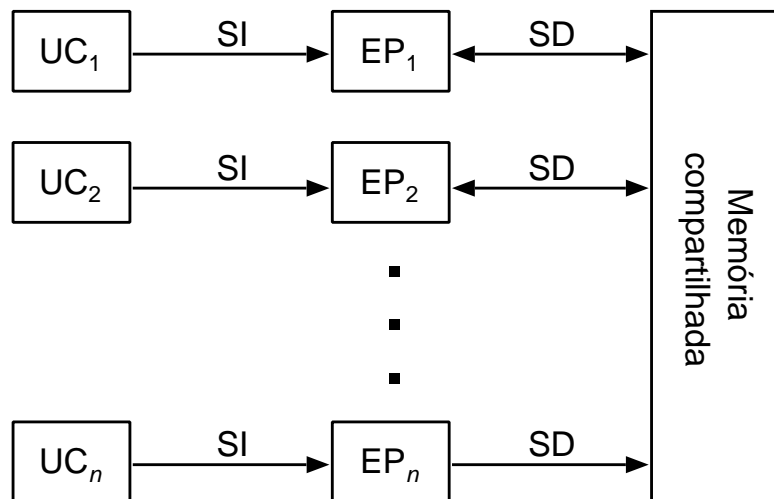
Processamento Paralelo



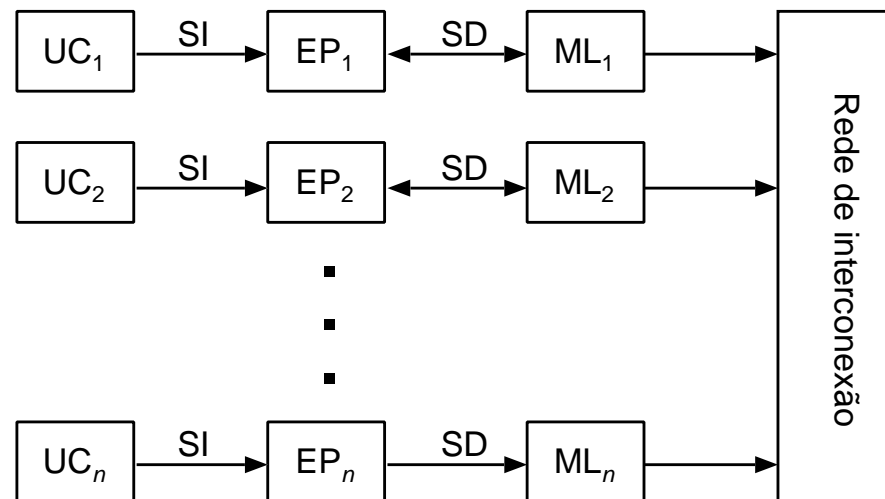
(a) SISD



(b) SIMD (com memória distribuída)



(c) MIMD (com memória compartilhada)



(d) MIMD (com memória distribuída)

Processamento Paralelo - Multiprocessadores Simétricos

- Características:
 - Existem 2 ou mais processadores similares
 - Processadores compartilham memória
 - Compartilham acesso aos dispositivos de E/S
 - Todos os processadores podem desempenhar a mesma função
 - Sistema controlado por SO integrado
 - Exemplo: IBM S/390, até 10 processadores.

Processamento Paralelo - Multiprocessadores Simétricos

- Vantagens em relação a uniprocessadores:
 - Desempenho: se o trabalho efetuado pelo computador pode ser organizado de forma que algumas porções possam ser feitas em paralelo, então um sistema com múltiplos processadores resulta em maior desempenho.



(a) Tempo compartilhado (multiprogramação)



(b) Tempo compartilhado e sobreposição (multiprocessamento)

 Bloqueado  Em execução

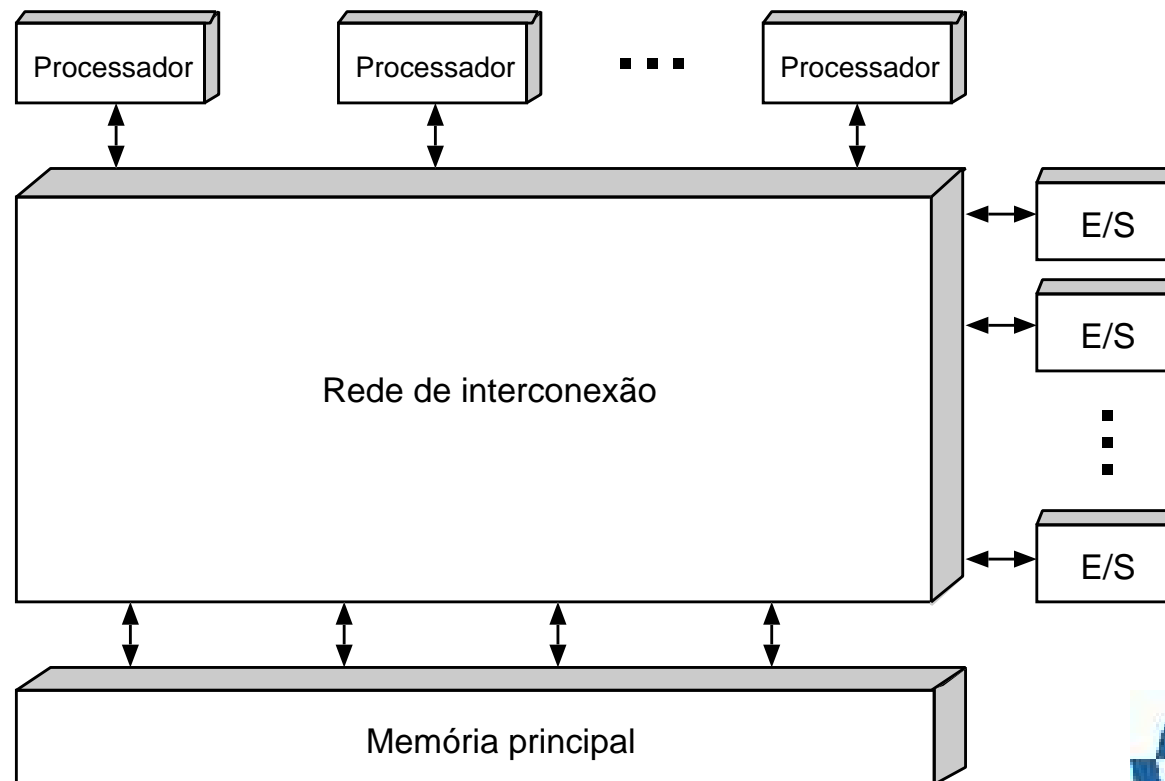
(Fig. 16.3 do livro de Arquitetura e Organização de Computadores" , William Stalling)

Processamento Paralelo - Multiprocessadores Simétricos

- Vantagens em relação a uniprocessadores:
 - Disponibilidade: a falha de um processador não causa a parada do sistema. O sistema pode continuar a funcionar com desempenho reduzido.
 - Crescimento incremental: para aumentar o desempenho pode se adicionar processador
 - Escalabilidade: fabricantes podem oferecer uma larga faixa de produtos com características de desempenho e custo diferentes, com base no número de processadores

Processamento Paralelo - Multiprocessadores Simétricos

- Organização:
 - Dois ou mais processadores (ULA, UC, registradores, cache)
 - Memória e dispositivos de E/S compartilhados
 - Processadores podem se comunicar por meio da memória (mensagens e informações armazenadas em áreas comuns)



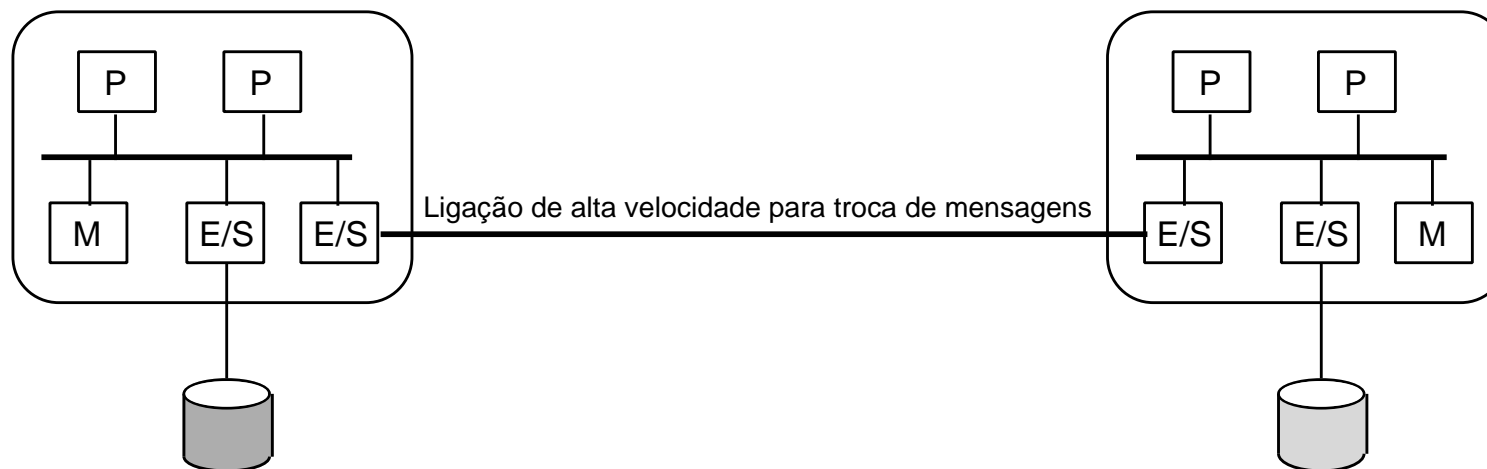
(Fig. 16.4 do livro de Arquitetura e Organização de Computadores", William Stalling)

Processamento Paralelo - Clusters

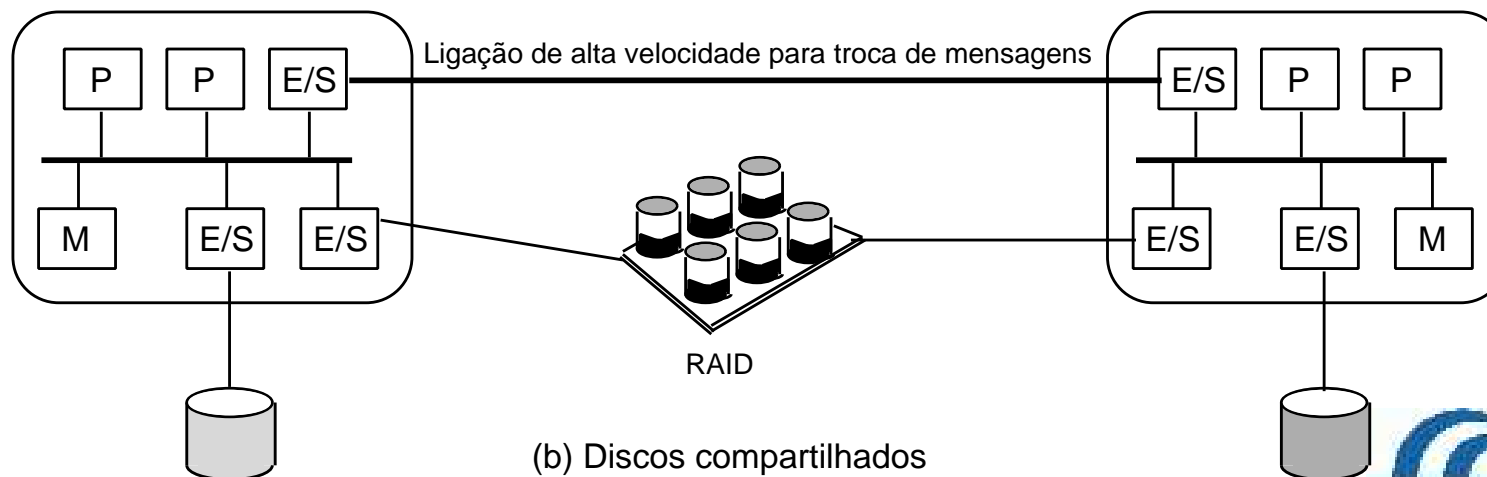
- Grupo de computadores completos interconectados, trabalhando juntos, como um recurso computacional unificado
- Benefícios:
 - Escalabilidade absoluta: é possível criar clusters muito grandes, dezenas de máquinas (cada máquina pode ser um multiprocessador)
 - Escalabilidade incremental: é possível expandi-lo de forma incremental
 - Alta disponibilidade: falha de um nó do cluster não significa perda total do serviço. Tolerância a falhas.
 - Melhor relação custo/ desempenho: facilidade de construir o sistema a partir de nós básicos comercialmente disponíveis

Processamento Paralelo - Clusters

- Configurações: classificação baseada na forma como os computadores do cluster compartilham acesso aos discos.



(a) Servidor independente, sem compartilhamento de discos



(b) Discos compartilhados

(Fig. 16.9 do livro de Arquitetura e Organização de Computadores" , William Stalling)

Processamento Paralelo - Clusters x SMP

SMP:

- Mais próximo do modelo de um único processador para o qual a maior parte das aplicações foi escrita
- Requer menos espaço físico e suprimento de energia que um cluster comparável

Clusters:

- Muito superiores em relação à escalabilidade absoluta e incremental
- Superiores em relação à disponibilidade, todos os componentes do sistema são altamente redundantes

Processamento Paralelo - NUMA

Definições:

- Acesso uniforme à memória (UMA): todos os processadores têm acesso a todas as partes da memória principal. O tempo de acesso é o mesmo - SMP.
- Acesso não uniforme à memória (NUMA): Todos os processadores têm acesso a todas as partes da memória principal. O tempo de acesso difere em relação à posição e processador.
- NUMA com coerência de cache (CC-NUMA): um sistema NUMA no qual é mantida coerência de cache entre as memórias cache dos vários processadores. Ex: Origin da Silicon Graphics, até 1024 processadores

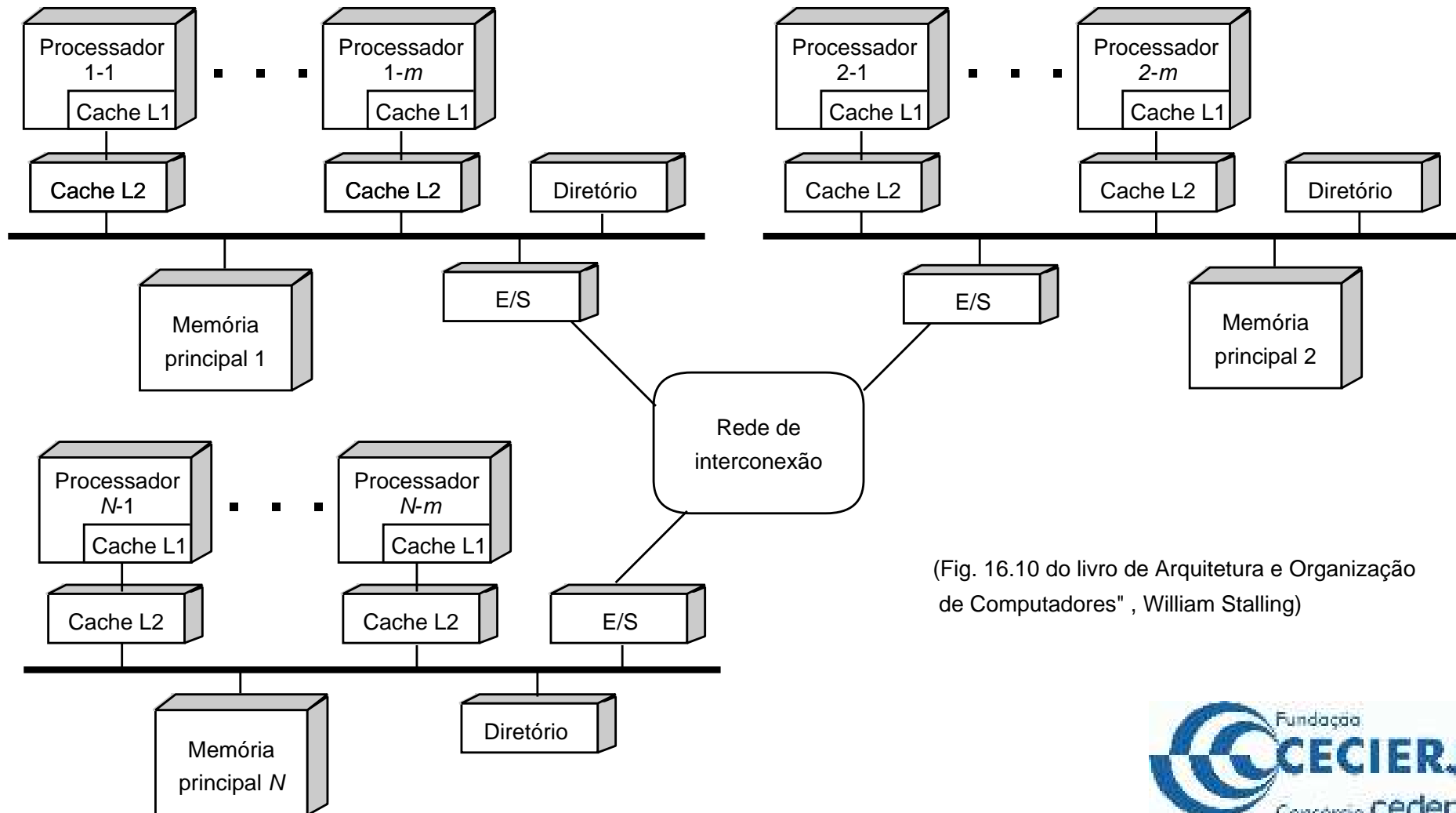
Processamento Paralelo - NUMA

Motivação:

- Em sistemas SMP existe um limite prático para o número de processadores que podem ser usados
- Um esquema de cache efetivo reduz o tráfego no barramento entre qualquer processador e a memória principal. A medida que aumenta o tráfego no barramento, este passa a constituir um gargalo de desempenho do sistema.
- A degradação limita o número de processadores
- Cluster: resolve o problema, mas a aplicação deve ser adaptada ao ambiente
- CC-NUMA grande área de memória no sistema, permitindo vários nós multiprocessadores, cada qual com o seu próprio barramento

Organização:

- Cada nó do sistema inclui uma memória principal, mas do ponto de vista dos processadores existe uma única memória endereçável, com cada posição de memória tendo um endereço único em todo o sistema.



(Fig. 16.10 do livro de Arquitetura e Organização de Computadores", William Stalling)

Processamento Paralelo - CC-NUMA

Consideração:

- Disponibiliza desempenho efetivo em níveis de paralelismo mais altos que o fornecido por sistemas SMP, sem requerer mudanças substanciais no software.

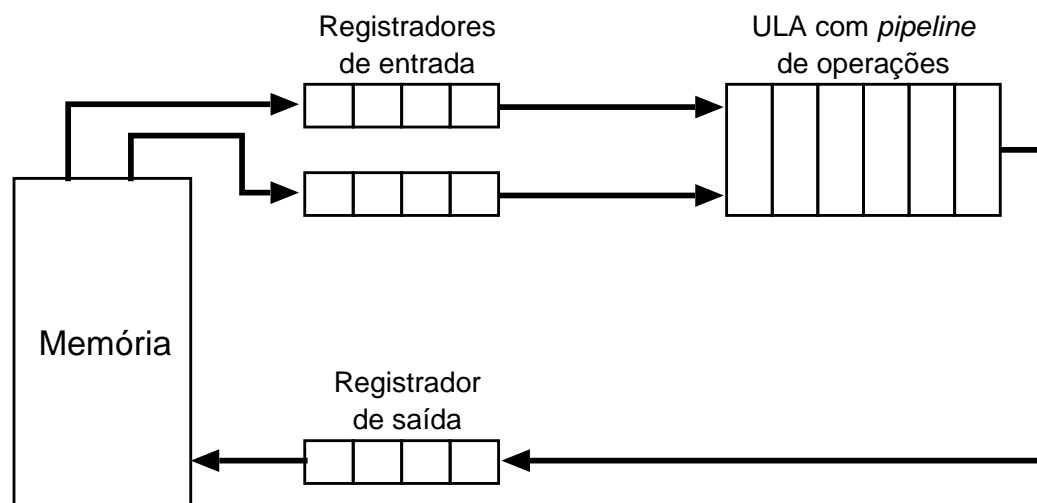
Processamento Paralelo - Computação Vetorial

- Computadores capazes de resolver problemas matemáticos relativos a processos reais em: aerodinâmica, sismologia, meteorologia e física atômica, nuclear e de plasma.
- Problemas efetuam repetidas operações aritméticas de ponto-flutuante em grandes vetores de números
- Supercomputadores muito custosos, otimizados para computação vetorial, projetados para efetuar centenas de milhões de operações de ponto flutuante por segundo

Processamento Paralelo - Computação Vetorial

Abordagens:

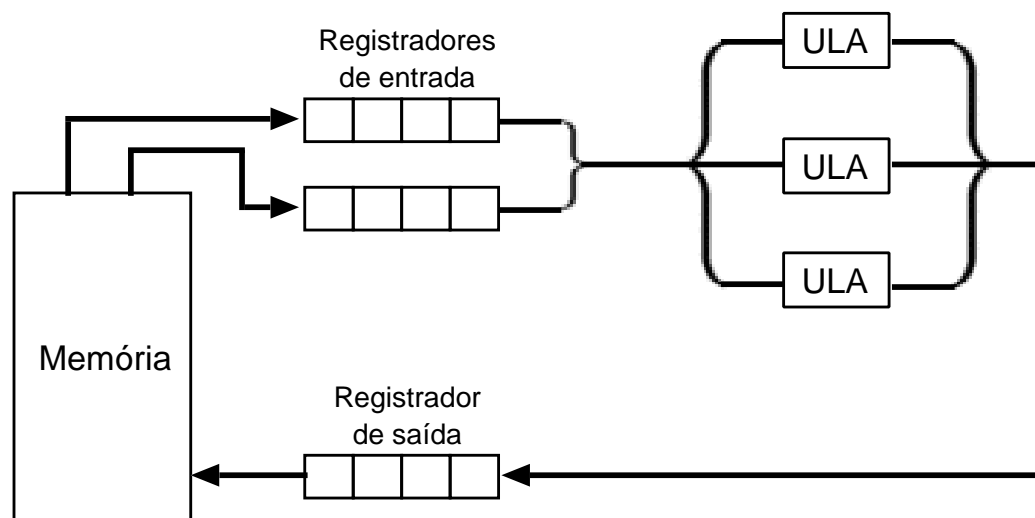
- ULA com pipeline de operações (vector processor)



(Fig. 16.13(a) do livro de Arquitetura e Organização de Computadores" , William Stalling)

Processamento Paralelo - Computação Vetorial

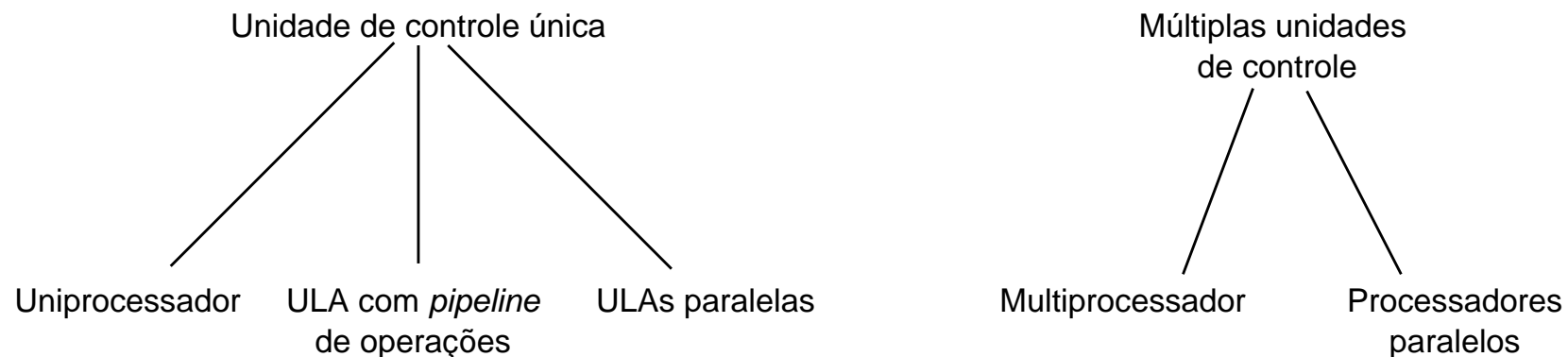
- ULAs paralelas (array processor)



(Fig. 16.13(b) do livro de Arquitetura e Organização de Computadores" , William Stalling)

Processamento Paralelo - Computação Vetorial

- Processadores Paralelos: múltiplos processadores trabalhando de forma cooperativa sobre uma dada tarefa



(Fig. 16.15 do livro de Arquitetura e Organização de Computadores" , William Stalling)

Bibliografia

*Arquiteturas RISC: Introdução à Organização de Computadores -
Mário A. Monteiro - LTC- Capítulo 11*

*Processamento Paralelo: Arquitetura e Organização de
Computadores - William Stalling - Prentice-Hall - Capítulo 16*

Exercícios

*Arquiteturas RISC: Introdução à Organização de Computadores -
Mário A. Monteiro - LTC- Capítulo 11*

Exercícios: 1, 2 ,3, 4 e 6

*Processamento Paralelo: Arquitetura e Organização de
Computadores - William Stalling - Prentice-Hall - Capítulo 16*

Exercício: 16.8