Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Unidade Acadêmica de Sistemas e Computação
Curso de Bacharelado em Ciência da Computação

Organização e Arquitetura de Computadores

(Processador, Parte V - Paralelismo)

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

Carga Horária: 60 horas

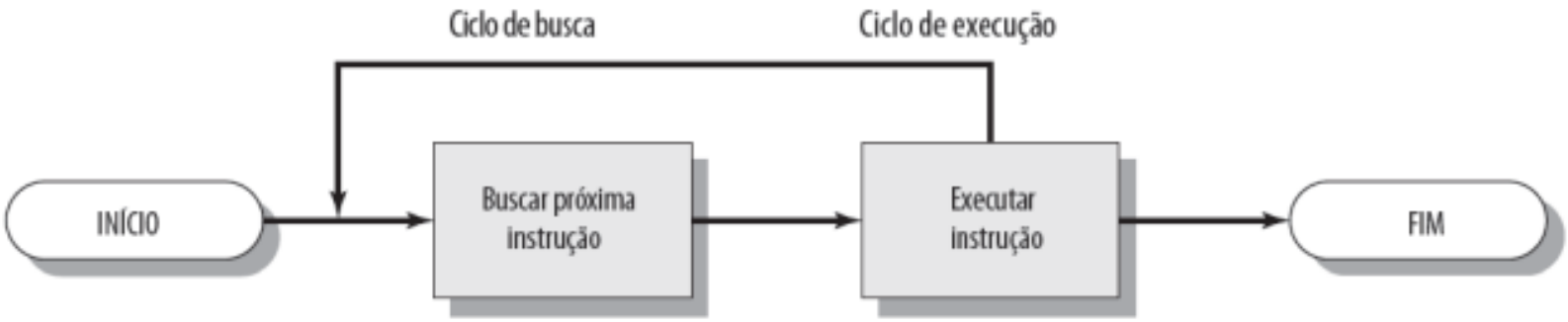
Tópicos

Organização Básica de Computadores (Processadores)

- Arquitetura do Conjunto de Instruções - Paralelismo

Processador - Paralelismo

- **Ciclo de Busca-Decodificação-Execução de Instrução**



Fonte: https://edisciplinas.usp.br/pluginfile.php/2868462/mod_resource/content/4/3aula%20-%20Historia_Visao_Geral.pdf

Processador - Paralelismo

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.

Processador - Paralelismo

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.

Processador - Paralelismo

Execução de Instruções

- **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).

- **RISC, CISC ?????**

Processador - Paralelismo

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.

Processador - Paralelismo

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.

Processador - Paralelismo

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.

Processador - Paralelismo

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.

Processador - Paralelismo

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.

Processador - Paralelismo

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.

Processador - Paralelismo

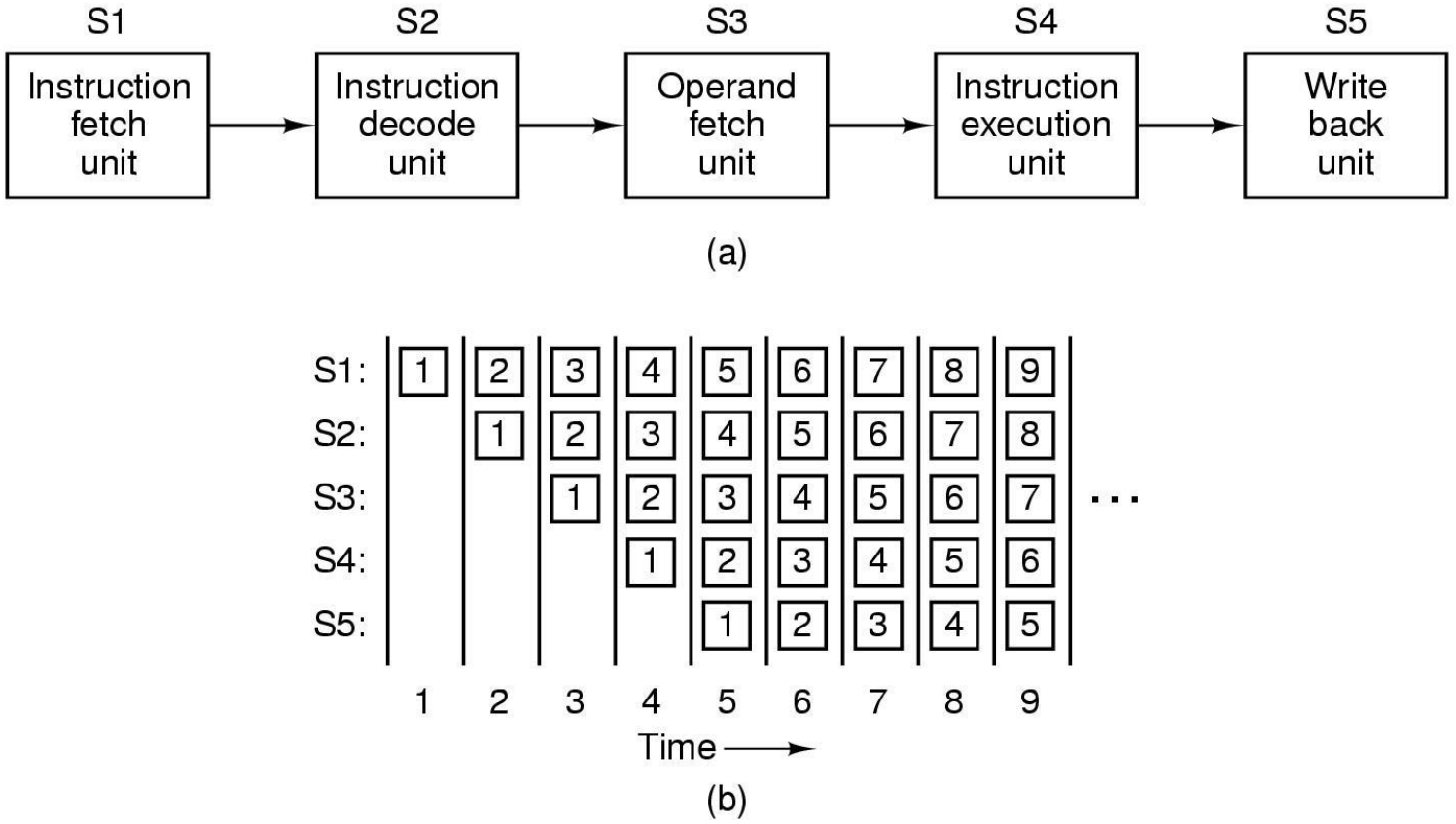
Princípios de Projeto para Computadores Modernos

- **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.

Processador - Paralelismo

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



Processador - Paralelismo

Paralelismo no Nível das Instruções

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

Processador - Paralelismo

Paralelismo no Nível das Instruções

- A ideia 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.

Processador - Paralelismo

Paralelismo no Nível das Instruções

- A ideia 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.

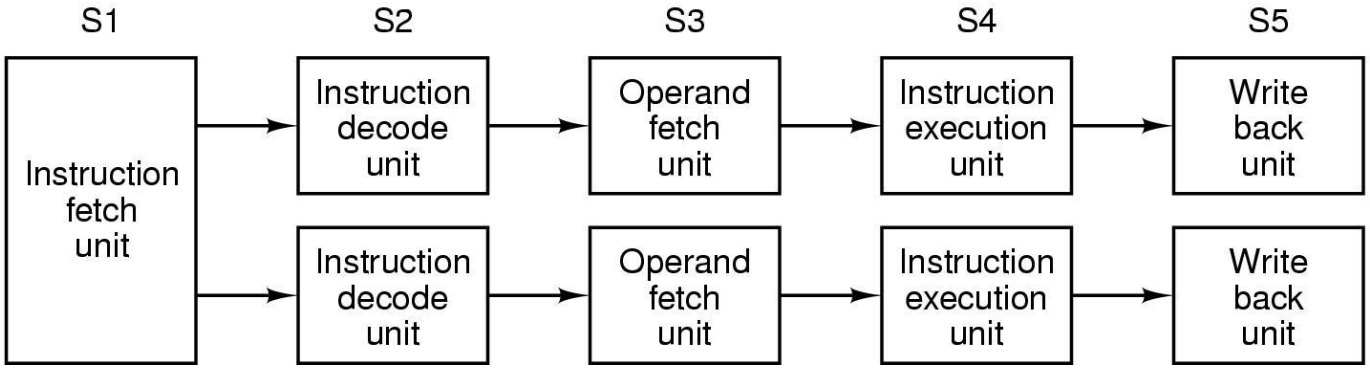
- **Exemplo**: no tempo 1 a instrução 1 está sendo lida, no tempo 2 a instrução 1 está sendo decodificada enquanto que a instrução 2 está sendo lida, no tempo 3 a instrução 1 está buscando dados, a instrução 2 está sendo decodificada e a instrução 3 está sendo lida, e assim por diante.

Processador - Paralelismo

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.

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



Processador - Paralelismo

Paralelismo no Nível das Instruções

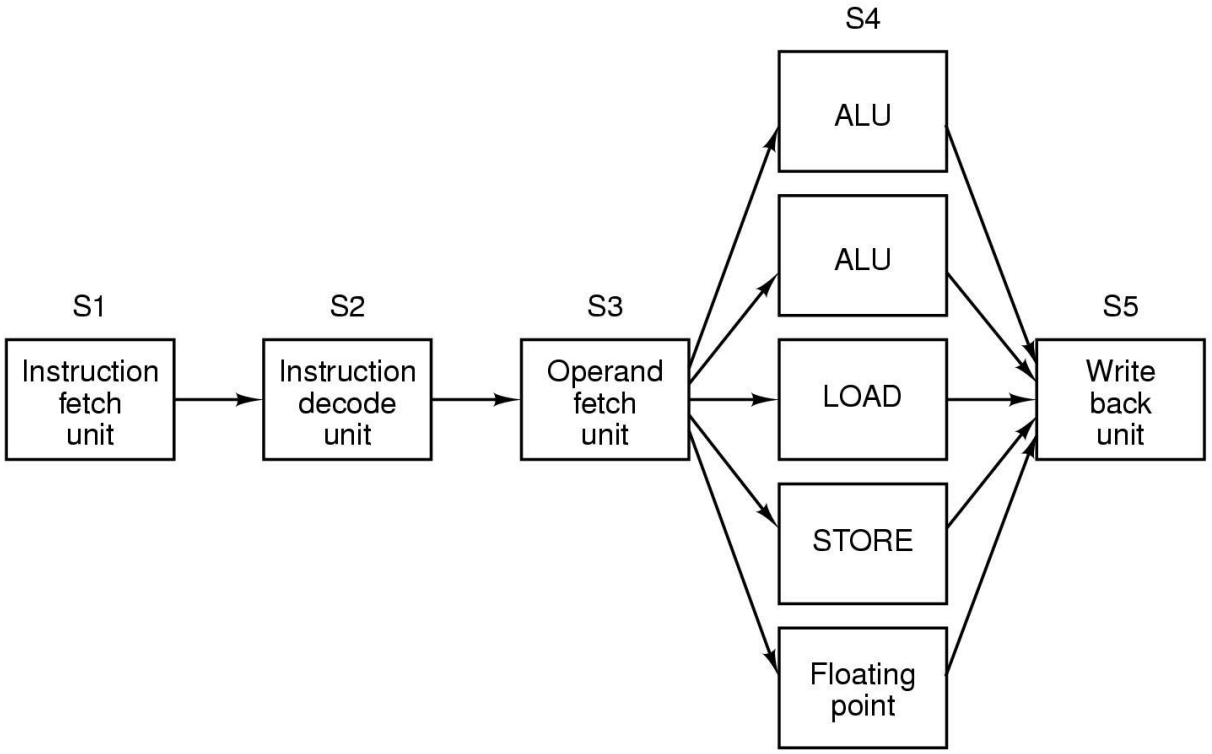
- **Arquiteturas Superescalares**
  - A execução das instruções é feita em paralelo e
    - não pode haver conflitos pelo uso de recursos (mesmo registro, por exemplo)
    - o resultado de uma instrução não pode depender do resultado da outra
    - pode se pensar em pipelines com leitura inicial de 3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

Processador - Paralelismo

Paralelismo no Nível das Instruções

- **Arquiteturas Superescalares (outra metodologia)**

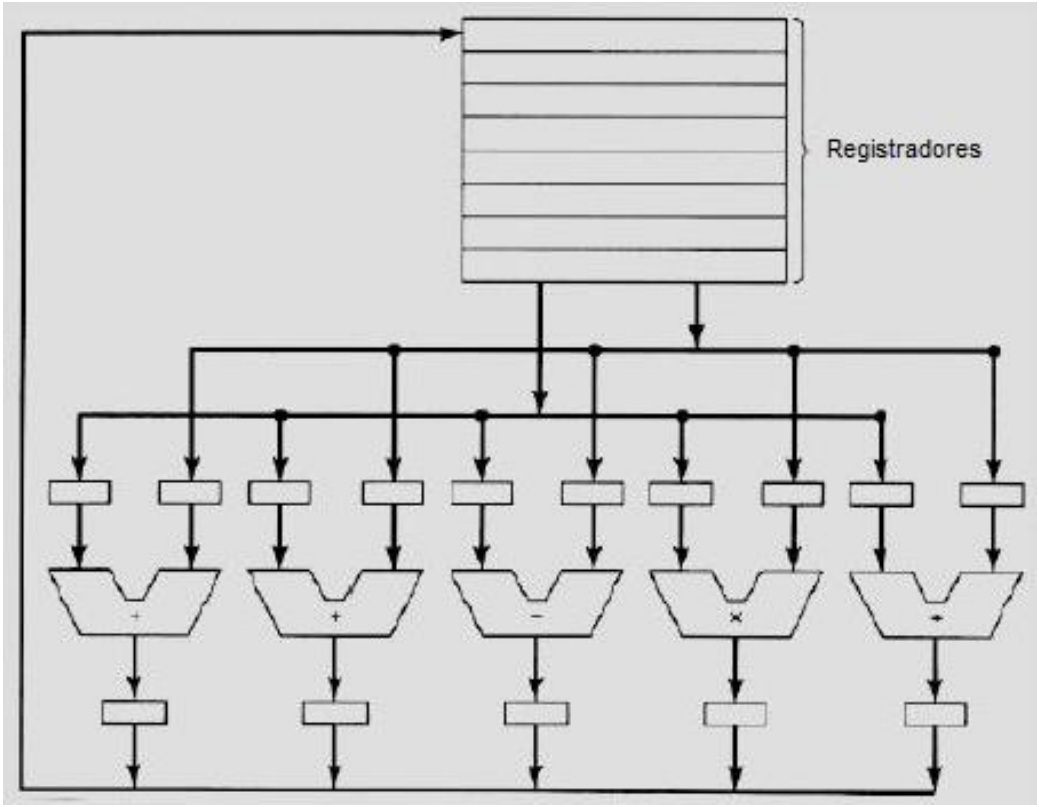
Processador superescalar com 5 unidades funcionais



Processador - Paralelismo

Paralelismo no Nível das Instruções

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



Processador - Paralelismo

Processadores: Estágios de Pipeline

Microarchitecture	Pipeline stages
P5 (Pentium)	5
P6 (Pentium Pro)	14
P6 (Pentium 3)	10
NetBurst (Willamette)	20
NetBurst (Northwood)	20
NetBurst (Prescott)	31
NetBurst (Cedar Mill)	31
Core/NHM/SNB/HSW	14
Atom Bonnell	16
Silvermont/Airmont	

Processador - Paralelismo

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.

Processador - Paralelismo

Computadores Matriciais (2 implementações)

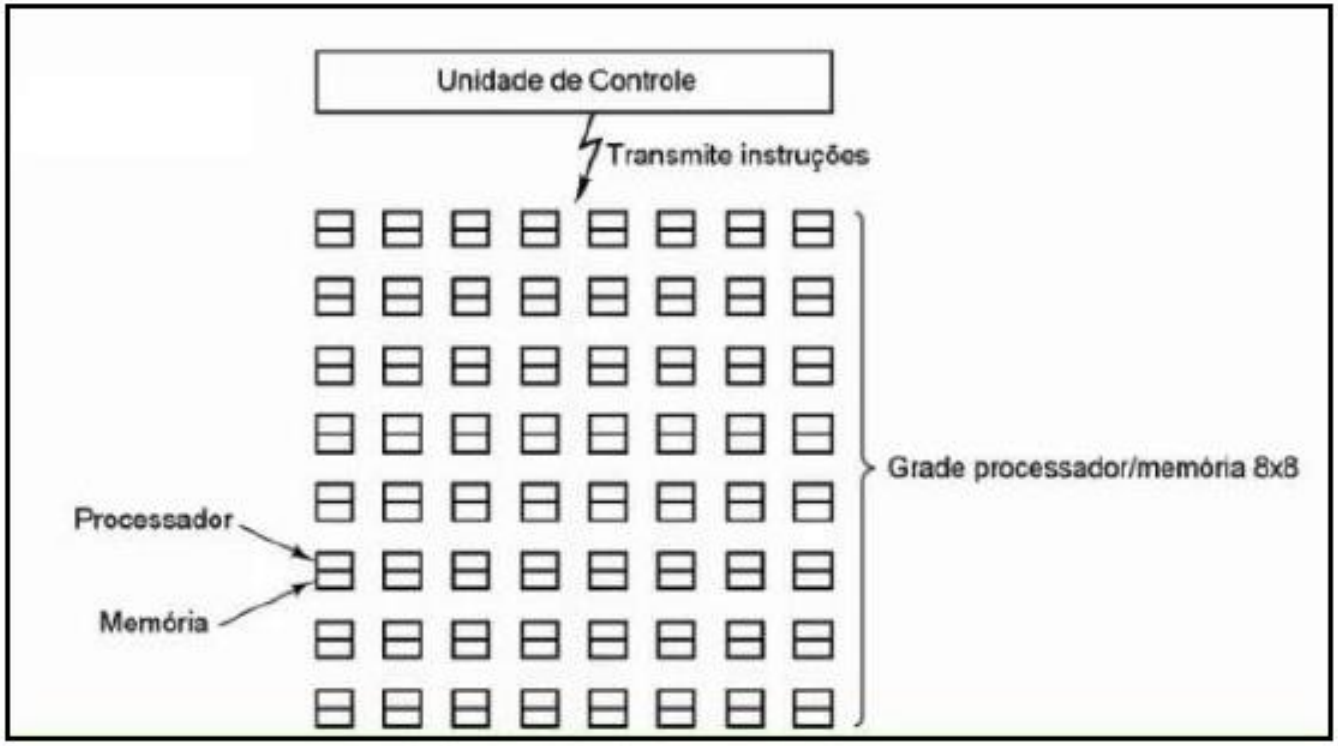
- **Processador Matricial**
  - Composto de grande número de processadores idênticos
  - Cada processador executa a mesma sequência de instruções sobre diferentes conjuntos de dados
  - Tem uma única unidade de controle
  - Tem uma ULA para cada processador
  - Custo elevado

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

Processador - Paralelismo

Computadores Matriciais (2 implementações)

- **Processador Matricial**



Processador - Paralelismo

Computadores Matriciais (2 implementações)

Processador Vetorial

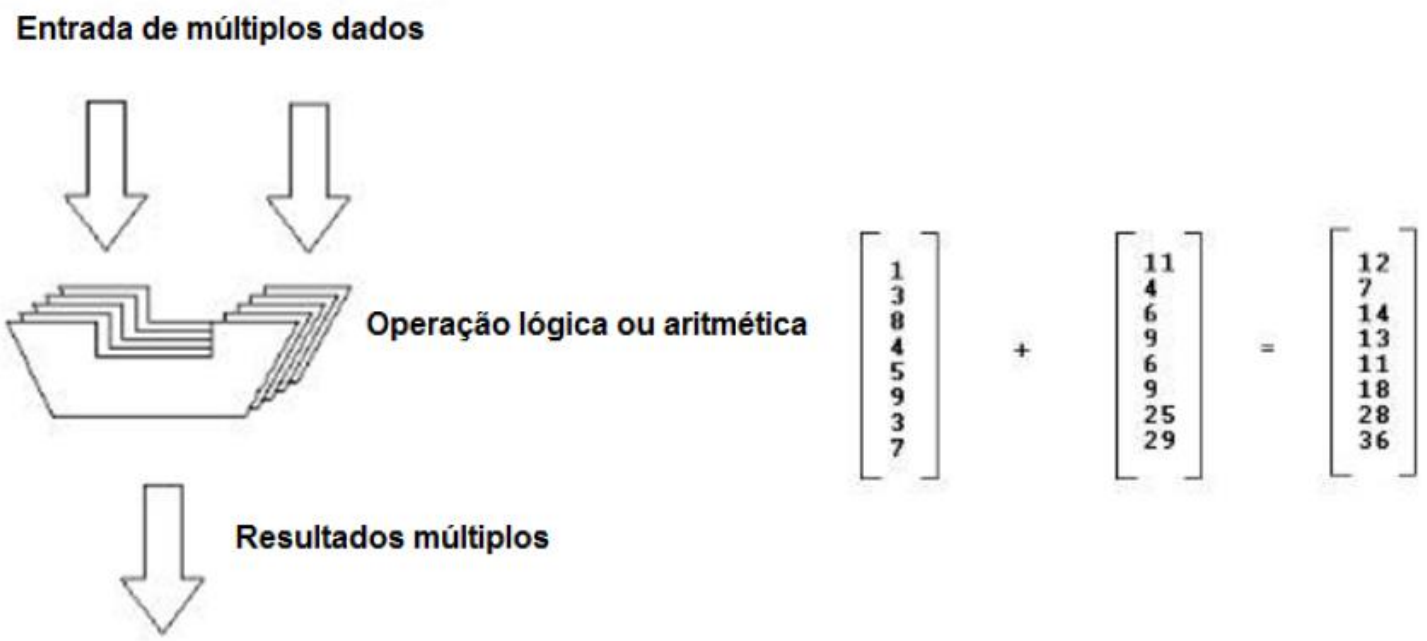
- muito parecido com processador matricial
- operações aritméticas são executadas numa única UAL, que opera em pipeline
- operandos são colocados em um registro vetorial para serem processados na ULA

Processador - Paralelismo

Computadores Matriciais (2 implementações)

- **Processador Vetorial**

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



Processador - Paralelismo

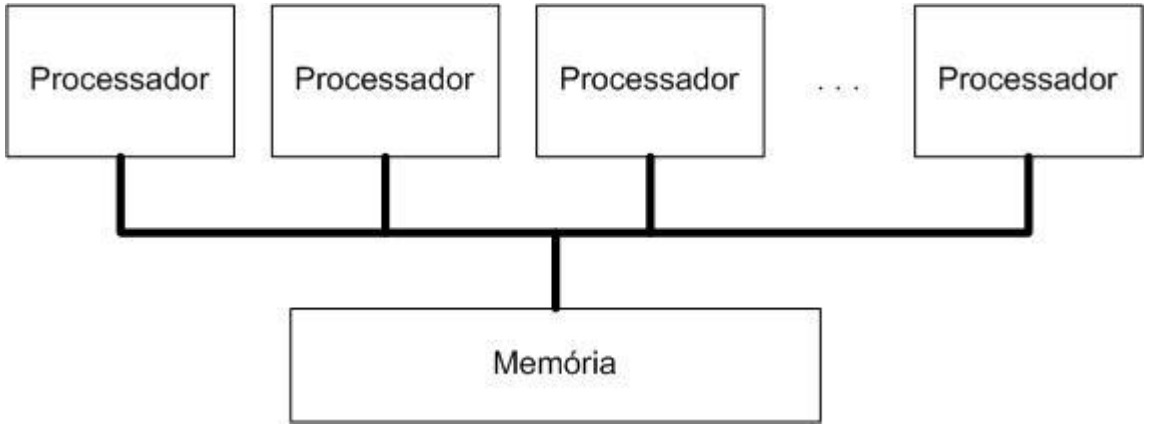
Multiprocessadores

- 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.

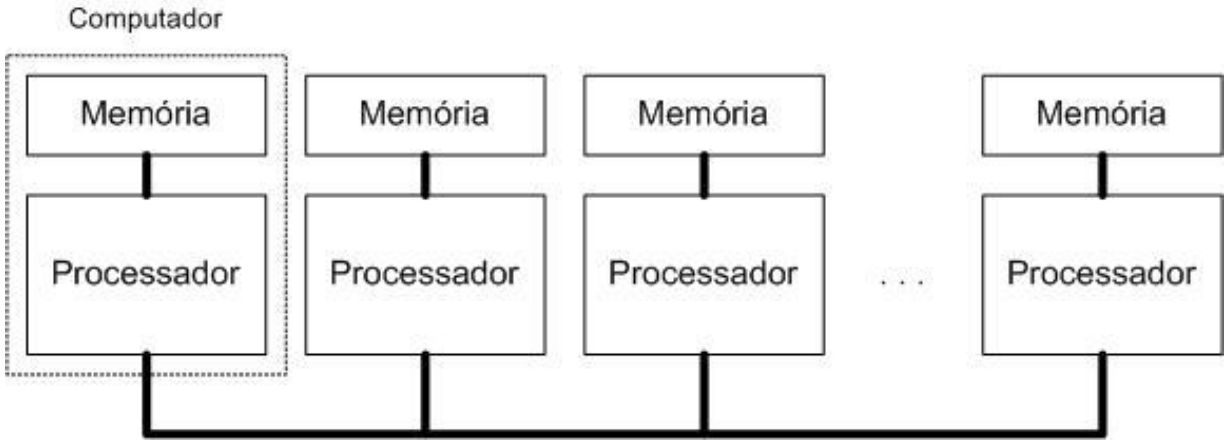
Processador - Paralelismo

Exemplos:

Multiprocessadores organizados em torno de um único barramento.



Multiprocessadores com memórias locais.



Processador - Paralelismo

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 10.000 computadores.

Processador - Paralelismo

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.

Processador - Paralelismo

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.

Processador - Paralelismo