

SSC0611

Arquitetura de Computadores

18ª Aula – Arquiteturas Paralelas

Profa. Sarita Mazzini Bruschi

sarita@icmc.usp.br

Arquiteturas Paralelas

- Níveis de paralelismo
 - Instrução (granulosidade fina)
 - Paralelismo entre as instruções
 - Arquiteturas Pipeline, Superescalar, VLIW
 - Tarefas (granulosidade média)
 - Paralelismo entre as *threads*
 - Arquiteturas SMT (*Simultaneous MultiThreading*)
 - Processos (granulosidade grossa)
 - Paralelismo entre os processos
 - Computação Paralela
 - Arquiteturas multiprocessadores e multicomputadores

Arquiteturas Paralelas

- **Computação Paralela – Conceitos**
 - Permite a execução das tarefas em menor tempo, através da execução em paralelo de diversas tarefas.
 - O paralelismo pode ser obtido em diversos níveis, com ou sem o uso de linguagens de programação paralela.
 - Arquiteturas de diversos tipos, elaboradas para aplicações específicas, podem ser utilizadas para acelerar a execução dessas aplicações.

Arquiteturas Paralelas

- Computação Paralela – Conceitos
 - Programação Sequencial
 - Programação Concorrente
 - Um servidor, atendendo vários clientes através de uma política de escalonamento no tempo
 - Programação Paralela
 - Vários servidores, atendendo vários clientes simultaneamente no tempo

Arquiteturas Paralelas

- Computação Paralela – Aplicações
 - Genoma Humano
 - Turbulência dos Fluidos
 - Dinâmica de Veículos
 - Circulação de Oceanos
 - Dinâmica de Fluidos Viscosos
 - Modelagem de Supercondutores
 - Cromodinâmica Quântica
 - Visão por Computador
 - Farmacêutica
 - Biologia Estrutural
 - Previsão do Tempo (+ 72 hs)

Arquiteturas Paralelas

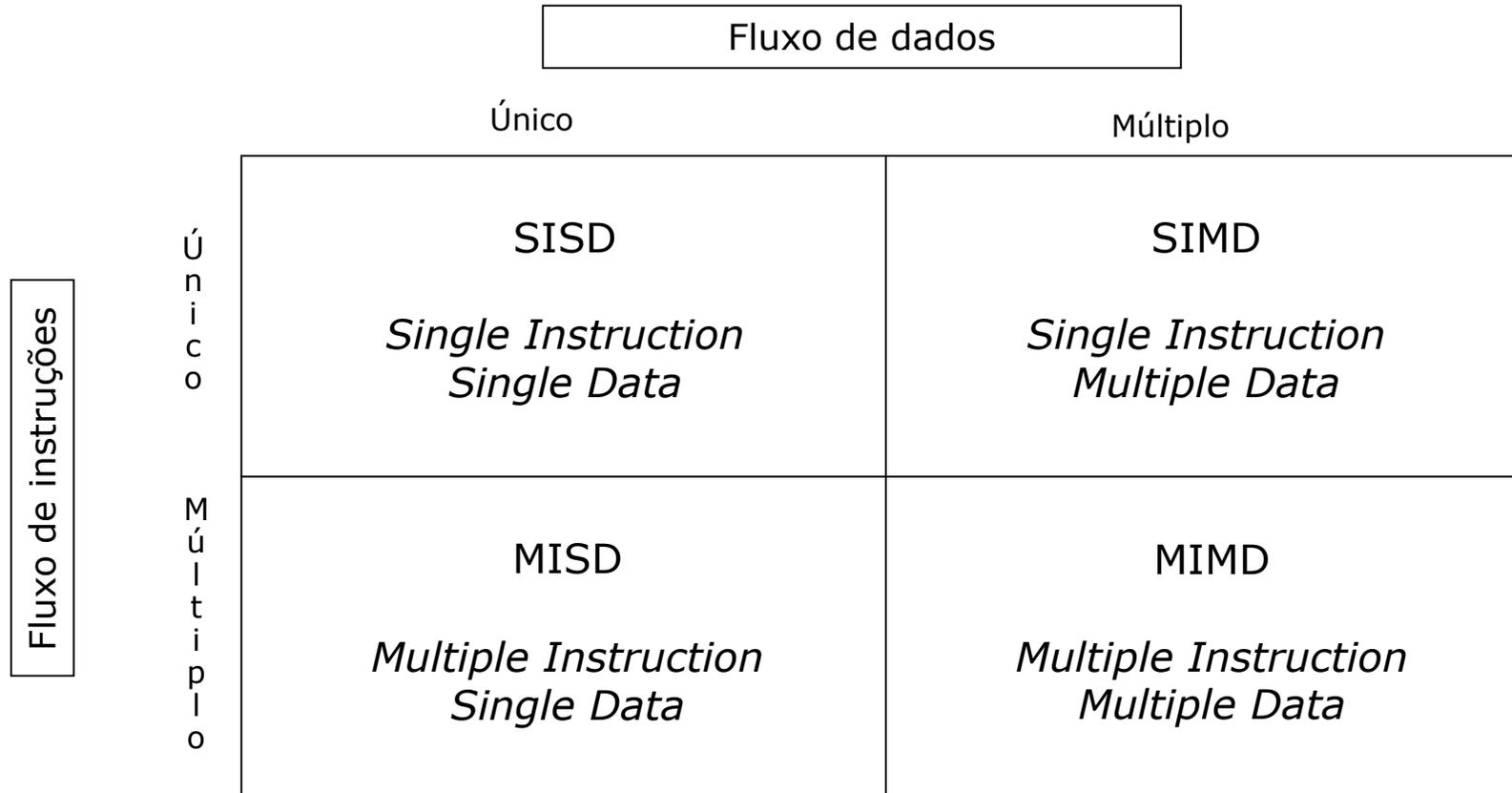
- Existem diversas classificações para as arquiteturas paralelas
- Devido a constante evolução, nenhuma classificação consegue abranger todas as arquiteturas existentes
- Classificação de Flynn (1972):
 - “Some computer organizations and their effectiveness”, *IEEE Transactions on Computers*, vol. C-21, pp. 948-960, 1972.
 - Mais conhecida
 - Baseia-se na unicidade e multiplicidade do fluxo de dados e instruções

Arquiteturas Paralelas

- Classificação de Duncan (1990)
 - “A survey of parallel computer architectures”, *IEEE Computer*, pp. 5-16, Fevereiro, 1990
 - Classificação mais recente e abrangente
 - Menos conhecida

Arquiteturas Paralelas

Classificação de Flynn



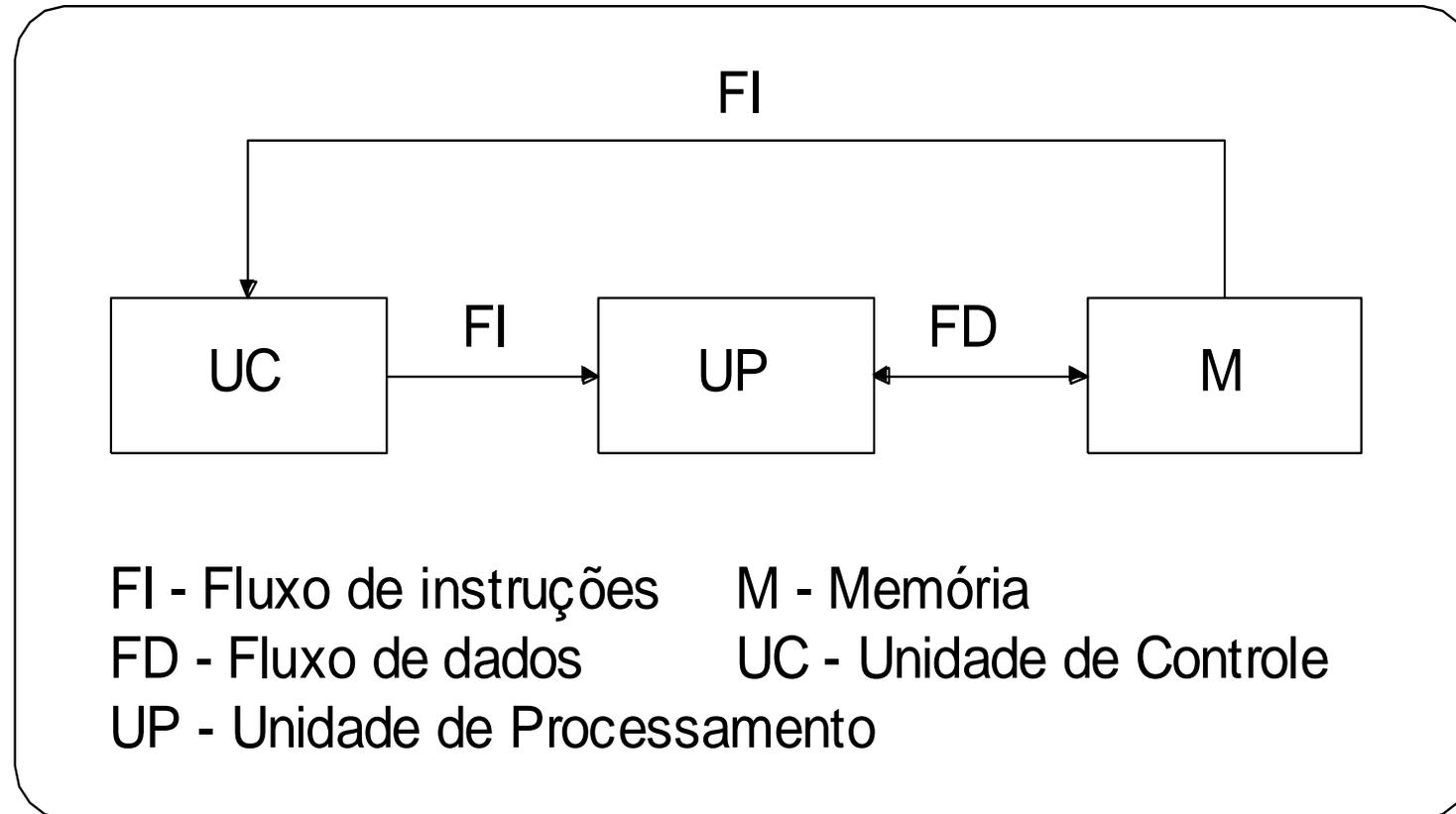
Arquiteturas Paralelas

Classificação de Flynn

- **SISD**: único fluxo de instrução, único fluxo de dados
 - classe que representa os computadores convencionais (seriais)
 - as instruções são executadas serialmente, porém os estágios (busca da instrução, decodificação, busca do operando e execução) podem ser sobrepostos (*pipeline*)
 - Pode-se saber o que está ocorrendo exatamente em cada instante de tempo e reproduzir o processo passo a passo mais tarde

Arquiteturas Paralelas

Classificação de Flynn – SISD



Arquiteturas Paralelas

Classificação de Flynn – SISD

- **Exemplo (SISD)**

LOAD A(I)
B(I)=A(I)*4 => MULT 4
STORE B(I)

TEMPO:

t₁

Processador P

LOAD A(1)

t₂

Processador P

MULT 4

t₃

Processador P

STORE B(1)

t₄

Processador P

LOAD A(2)

t₅

Processador P

MULT 4

t₆

Processador P

STORE B(2)

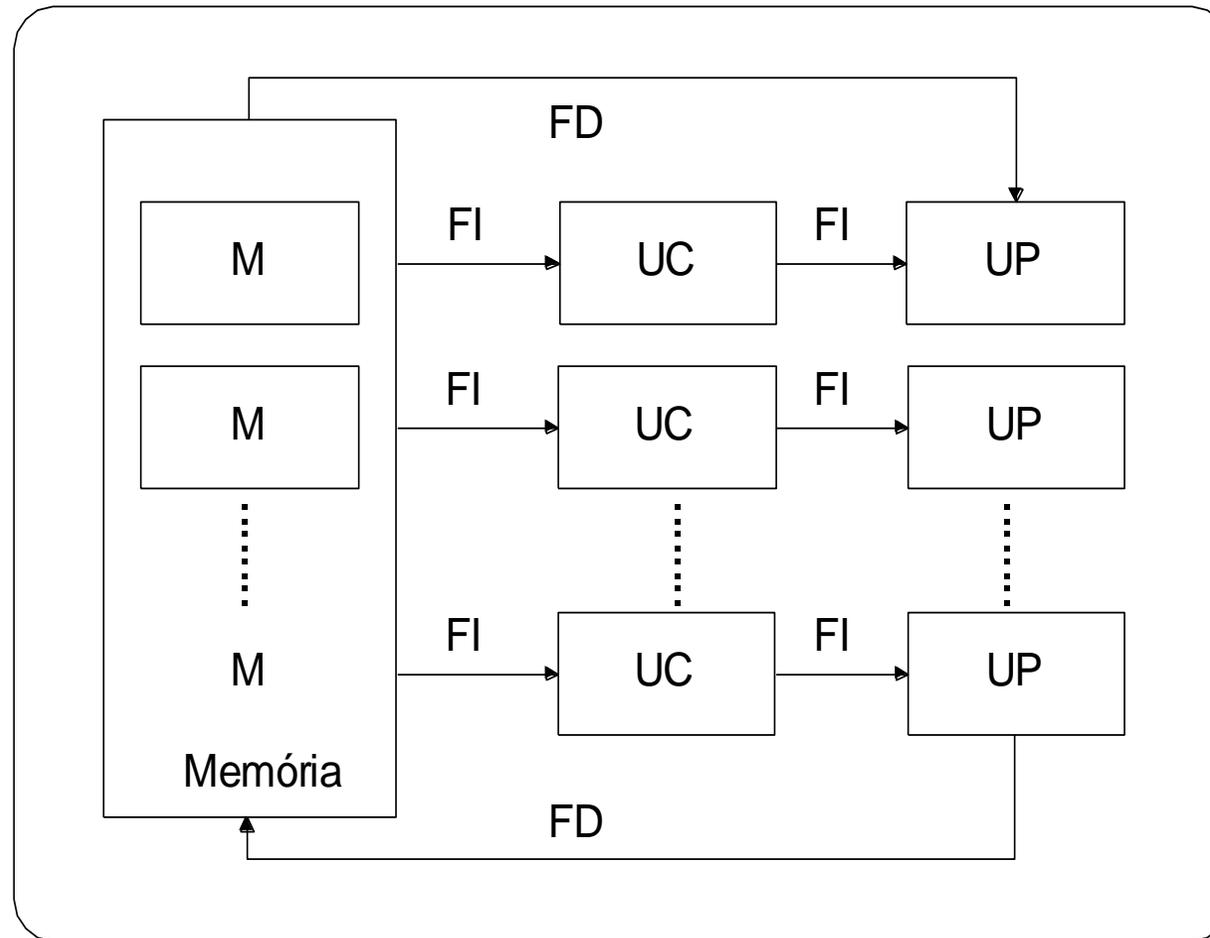
Arquiteturas Paralelas

Classificação de Flynn – MISD

- **MISD**: múltiplo fluxo de instruções, único fluxo de dados
 - vários processadores, onde cada um recebe instruções distintas mas operam sobre o mesmo conjunto de dados
 - poucos exemplos
 - Múltiplos filtros de frequência operando sobre um único fluxo de sinal
 - Múltiplos algoritmos de criptografia para decodificar uma mensagem

Arquiteturas Paralelas

Classificação de Flynn – MISD



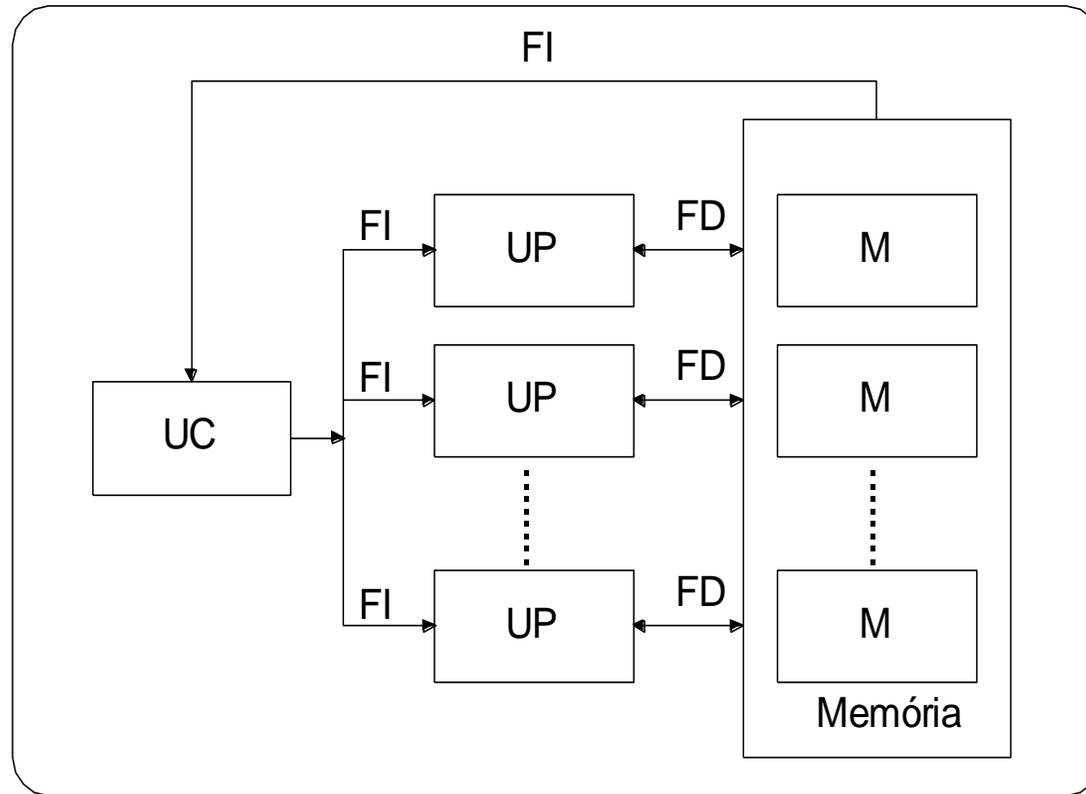
Arquiteturas Paralelas

Classificação de Flynn – SIMD

- **SIMD**: único fluxo de instruções, múltiplo fluxo de dados
 - classe que representa os processadores matriciais, paralelos e associativos
 - uma única unidade de controle que envia um fluxo de instruções para vários processadores
 - os processadores recebem a mesma instrução ao mesmo tempo e atuam sobre diferentes fluxos de dados
 - Cambridge Parallel Processing Gamma II Plus

Arquiteturas Paralelas

Classificação de Flynn – SIMD



- 1024 processadores em um array 32x32, ou 4096 em 64x64
- processador de controle armazena instruções e dados são armazenados na memória de cada processador

Arquiteturas Paralelas

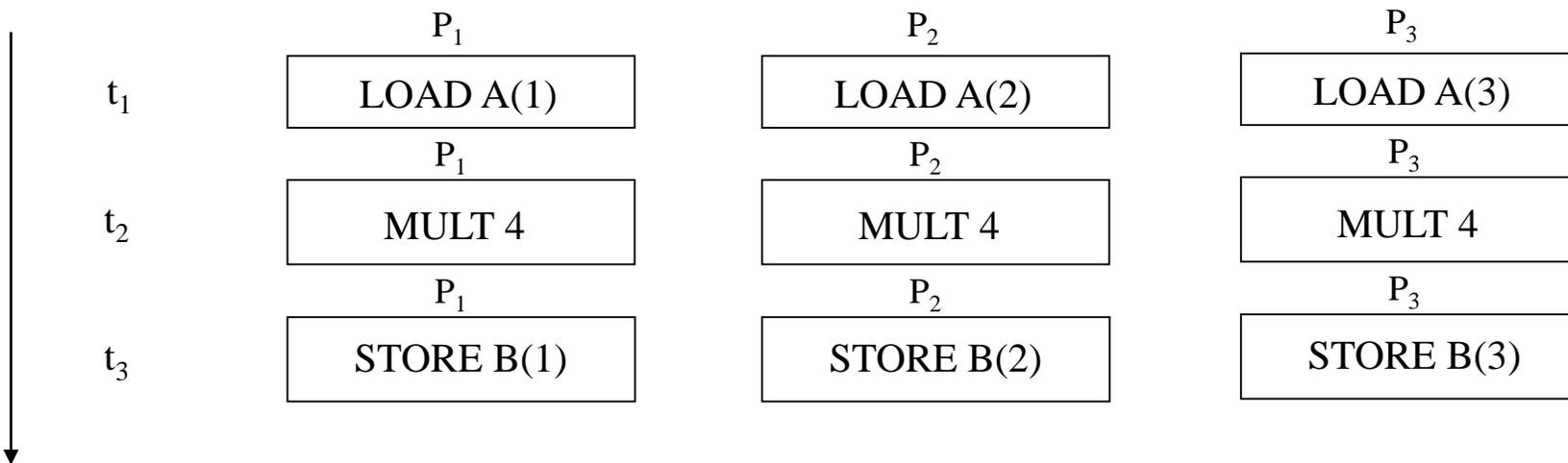
Classificação de Flynn – SIMD

- **Exemplo (SIMD)**

$B(I) = A(I) * 4 \Rightarrow$

LOAD A(I)
MULT 4
STORE B(I)

TEMPO:



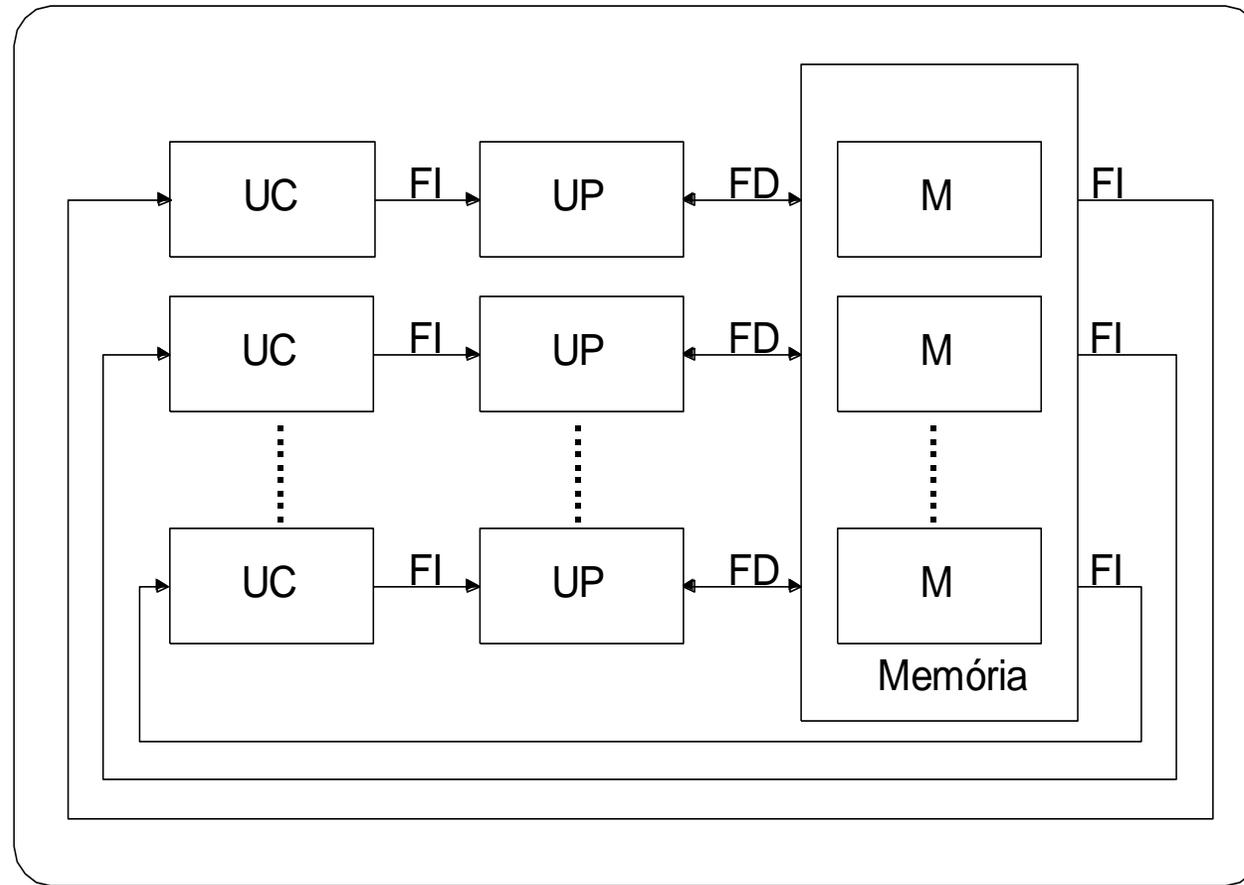
Arquiteturas Paralelas

Classificação de Flynn – MIMD

- **MIMD**: múltiplo fluxo de instruções, múltiplo fluxo de dados
 - vários processadores, cada um controlado por uma unidade de controle
 - processadores recebem instruções diferentes e operam sob fluxo de dados diferentes
 - podem ser síncronos ou assíncronos

Arquiteturas Paralelas

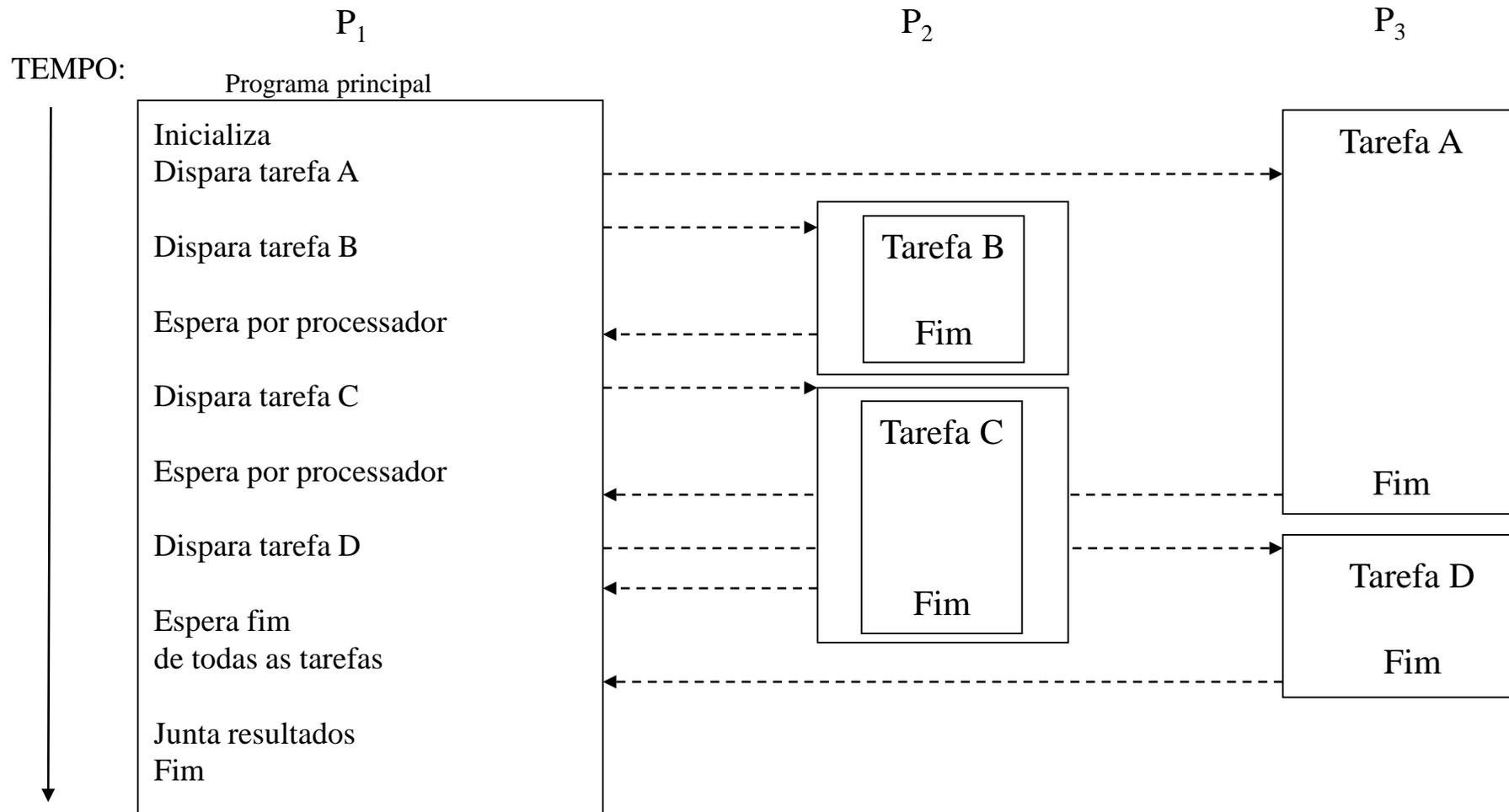
Classificação de Flynn – MIMD



Arquiteturas Paralelas

Classificação de Flynn – MIMD

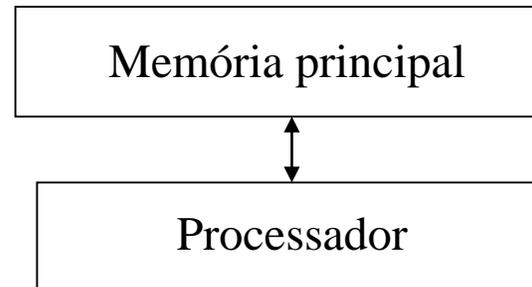
- **Exemplo (MIMD)**



Modelos de acesso à memória

- Um computador convencional consiste de um processador executando um programa armazenado na memória:

Instruções para o processador
Dados para ou do processador



Modelos de acesso à memória

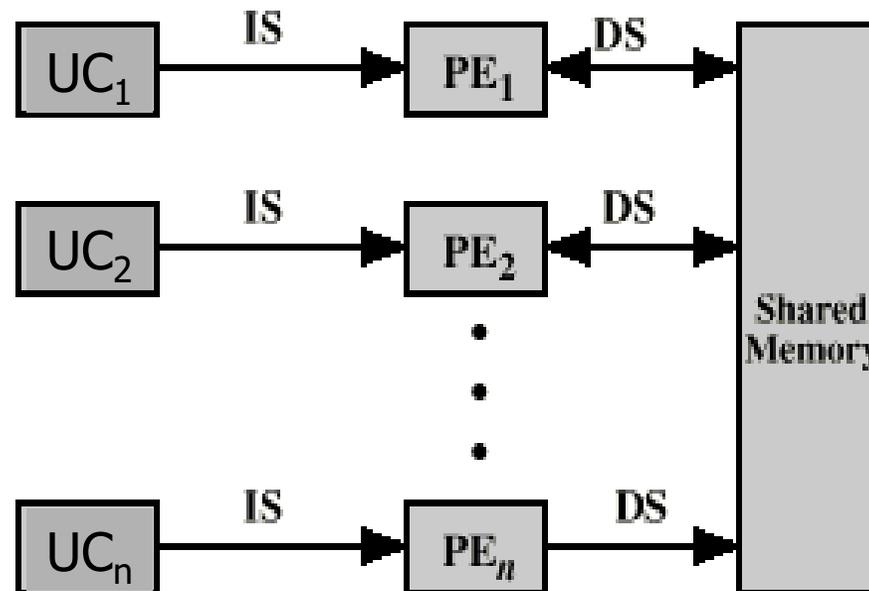
- Cada lugar da memória possui um endereço que inicia em 0 e vai até $2^n - 1$, onde n é o número de bits do endereço
- Em computação paralela, pode-se ter:
 - memória compartilhada (multiprocessadores)
 - memória distribuída (multicomputadores)

MIMD com Memória compartilhada

- A mesma memória é acessada pelos múltiplos processadores
- Sincronização entre tarefas é feita por escrita/leitura na/da memória compartilhada e usuário é responsável por sua especificação
- Um lugar da memória não pode ser modificado por uma tarefa enquanto outra o estiver acessando

MIMD com Memória compartilhada

- Comunicação entre tarefas é rápida
- Escalabilidade limitada pelo número de caminhos entre memória e processadores



MIMD com Memória compartilhada

- Usuário é responsável pela sincronização
- Programação:
 - Linguagens de programação paralela
 - Construções e instruções paralelas permitem declarações de variáveis compartilhadas e seções paralelas de código
 - Compilador responsável pela geração do código final executável
 - Exemplo: OpenMP
 - Threads
 - Sequências de código escritas em alto nível para processadores individuais que podem acessar localidades compartilhadas
 - Exemplo: pThreads

MIMD com Memória compartilhada

- Exemplos:
 - SMP (*Symmetric MultiProcessors*)
 - SGI Power Challenge
 - Máquinas da Sun Microsystems
 - Silicon Graphics
 - NUMA (*NonUniform Memory Access*)

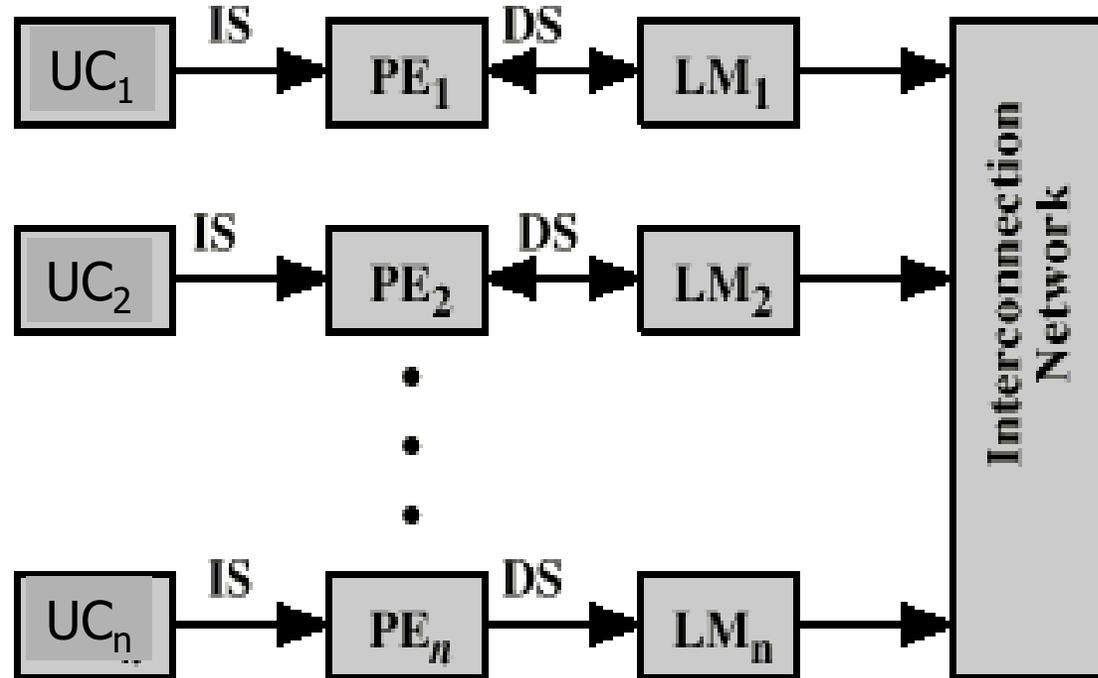
MIMD com Memória distribuída

- Memória fisicamente distribuída entre os processadores e cada memória local só pode ser acessada pelo seu processador
- Tarefas se comunicam através de troca de mensagens e a sincronização entre as tarefas é feita através dessa troca

MIMD com Memória distribuída

- Programação:
 - Bibliotecas com rotinas para passagem de mensagens que são ligadas a programas sequenciais convencionais são bastante utilizadas
 - Exemplo: MPI
 - Problema dividido em um número de tarefas que se comunicam

MIMD com Memória distribuída



MIMD com Memória distribuída

- MPP (*Massively Parallel Processors*)
 - interconectadas por rede de alta velocidade
 - boa escalabilidade (podem ser formadas por uma grande quantidade de máquinas)
 - complicadas de programar
 - alto custo (em torno de US\$ 1.000.000)
 - Exemplos:
 - Intel Paragon
 - Cray T3E
 - Thinking Machines CM-5

MIMD com Memória distribuída

- *COW (Cluster of Workstations)*
 - utilização de estações de trabalho em uma rede local de processamento
 - baixo custo
 - boa relação custo/benefício
- Grades Computacionais

Arquiteturas Paralelas

