

① 2ª Aula: (1ª Aula: Apenas apresentação da disciplina)

Processamento Paralelo e Distribuído



Processamento de Alto Desempenho (PAD)

Paralelismo: Quebra do paradigma de execução sequencial (Von Neumann)

Histórico do paralelismo:

- Descrito em 1842 em máquinas analíticas (Babbage)
- Idealizado na década de 50 por Von Neumann
- Tornou-se possível na década de 70 (ILLIAC-IV com 64 processadores)

- ②
- Anos 80: Computadores vetoriais
 - Anos 90: Computação distribuída
 - Anos 2000: Computação largamente distribuída

Por que PAD?

- Redução de tempo de processamento
- Aumento de poder computacional: processamento e memória
- Resolver problemas "pesados"

Estilos de programação:

- Sequencial
- Paralelo
- Concorrente (pseudo-paralelismo)

3)

Exploração do paralelismo em diversos níveis:

Processos leves (threads) Invocação de Métodos Remotos	Aplicação
Multiprocessamento Multiprogramação	Sistema Operacional
Vários processadores Hierarquia de memória e barramento	Arquitetura
Múltiplas unidades funcionais Pipeline de instrução	Processador

4)

3 maneiras de executar uma tarefa mais rápida:

- Trabalhar mais rápido (Processador)
- Trabalhar de maneira mais eficiente (Algoritmo)
- Pedir ajuda (Paralelismo)

Maior dificuldade: Identificar o paralelismo

- Quais trechos de código consomem mais computação?
- Quais loops podem ser executados de forma independente?
- Quais rotinas podem ser executadas em paralelo?

⑤ Exemplo:

1 pedreiro faz um muro em 3 horas

2 pedreiros fazem em 2 horas

3 pedreiros fazem em 1,5 horas

4 pedreiros fazem em 2 horas (aumentou?!)

Complicadores:

→ Muro só pode ser feito de baixo para cima (dependência de dados)

→ Os tijolos tem que ser distribuídos entre os pedreiros (distribuição de dados)

→ Um pedreiro não pode levantar seu lado do muro muito mais rápido que os outros (sincronização)

⑥ → Se existe apenas um canteiro de cimento ele será disputado por todos os pedreiros (área crítica)

Taxonomia/Classificação de Flynn (1972)

		Fluxo de Dados	
		Único	Múltiplo
Fluxo de Instruções	Único	SISD	SIMD
	Múltiplo	MISD	MI MD

⑦

SISD: Single Instruction Single Data

SIMD: Single Instruction Multiple Data

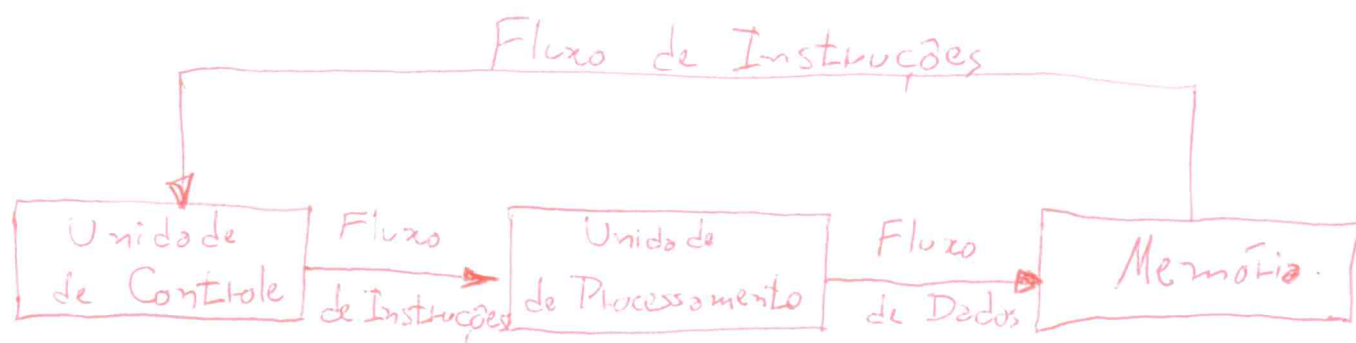
MISD: Multiple Instruction Single Data

MIMD: Multiple Instruction Multiple Data

⑧

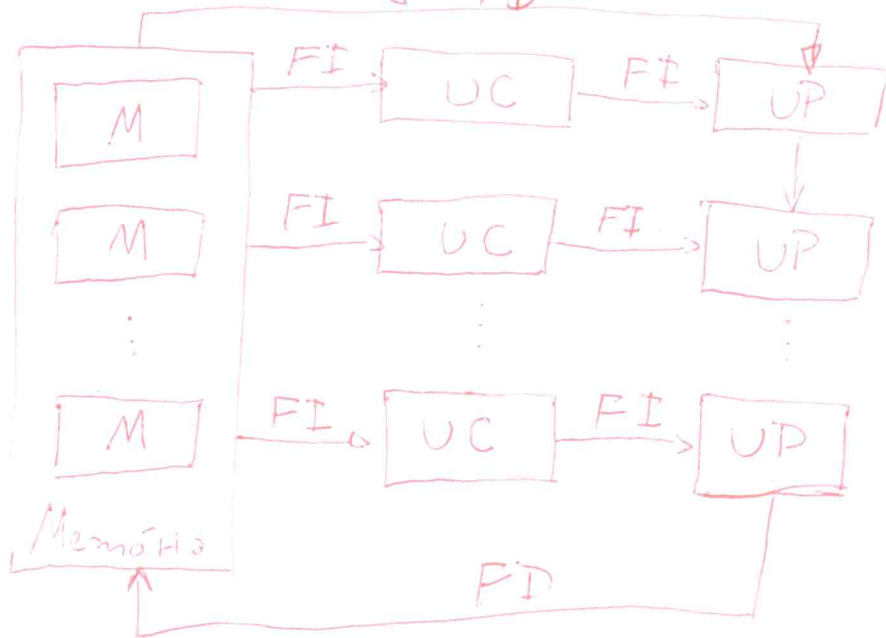
SISD

Computadores convencionais (seriais)



9) MI SD ~~(~~Exemplo: LIAC-III~~)~~

Vários processadores, onde cada um recebe instruções distintas mas operam sobre o mesmo conjunto de dados.

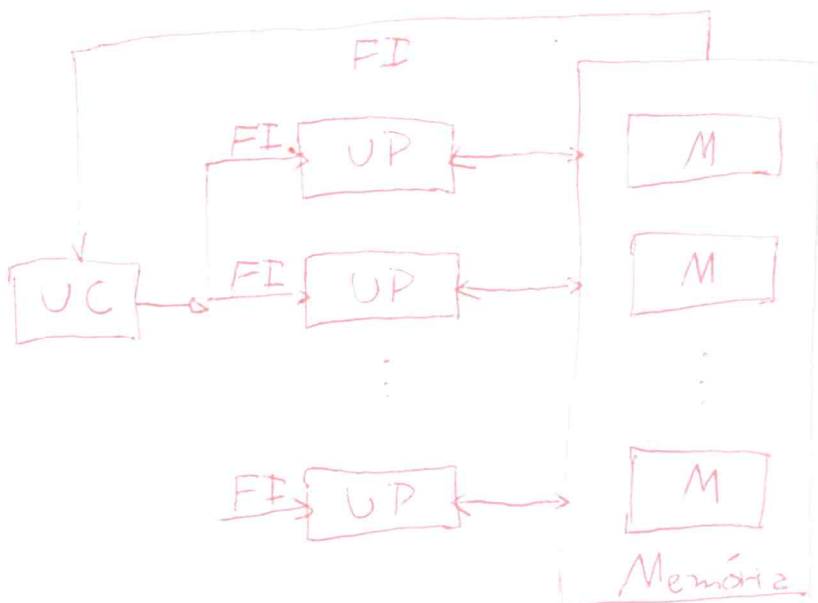


Exemplo:

Warp (Carnegie Mellon University)

10) SIMD

Categoria que representa os processadores matriciais/vetoriais.

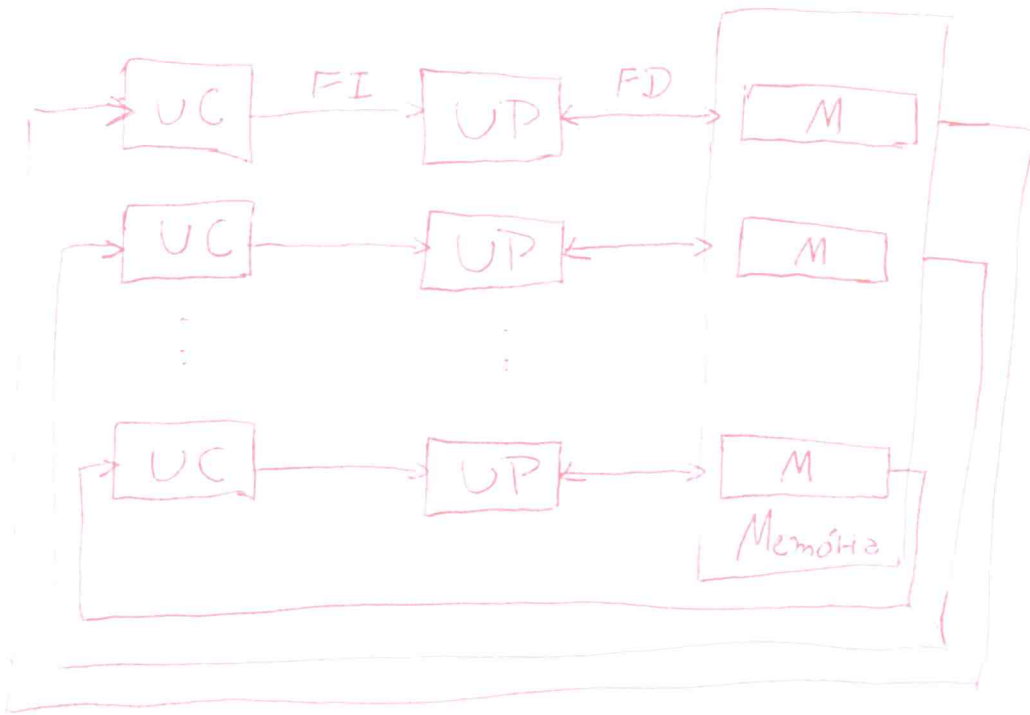


Exemplo:

ILIAC-IV

⑪ MI MD

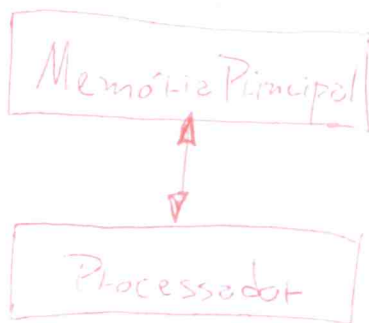
Engloba a maioria dos computadores paralelos.



Exemplo:
Tipo (INPE)

⑫ Modelos de acesso à memória

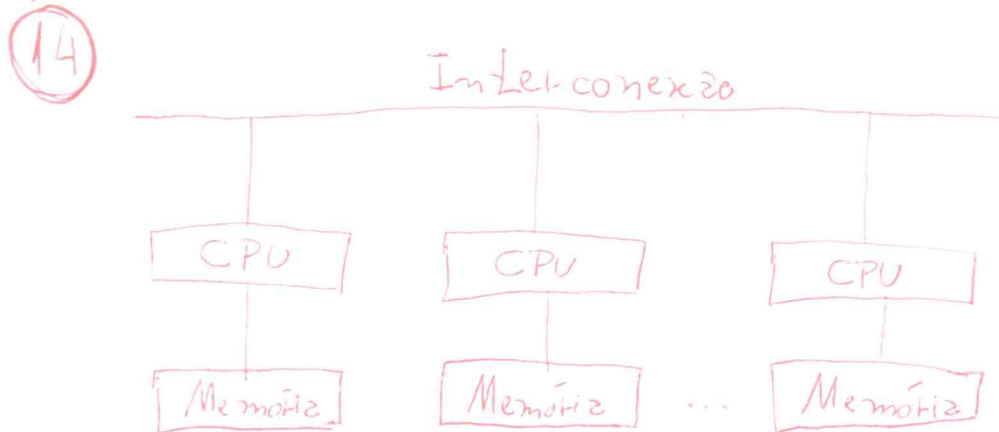
Computador convencional



Instruções para o processador

Dados para/do processador

~~14~~ Memória distribuída (Multi computadores)



Exemplos:

- Cluster: Memória compartilhada através de uma rede local
- Grid: Memória compartilhada através de computadores distantes geograficamente, e além disso as mensagens podem ser trocadas pela web.

~~14~~ Memória compartilhada (Multiprocessadores)



Exemplos:

- Multi-Core: Múltiplos processadores dentro de uma única pastilha compartilhando cache
- Multi-Processadores: Vários processadores em uma mesma placa, cada um com seu cache, mas a troca de informação é feita por um barramento

