

Arquitetura de Computadores

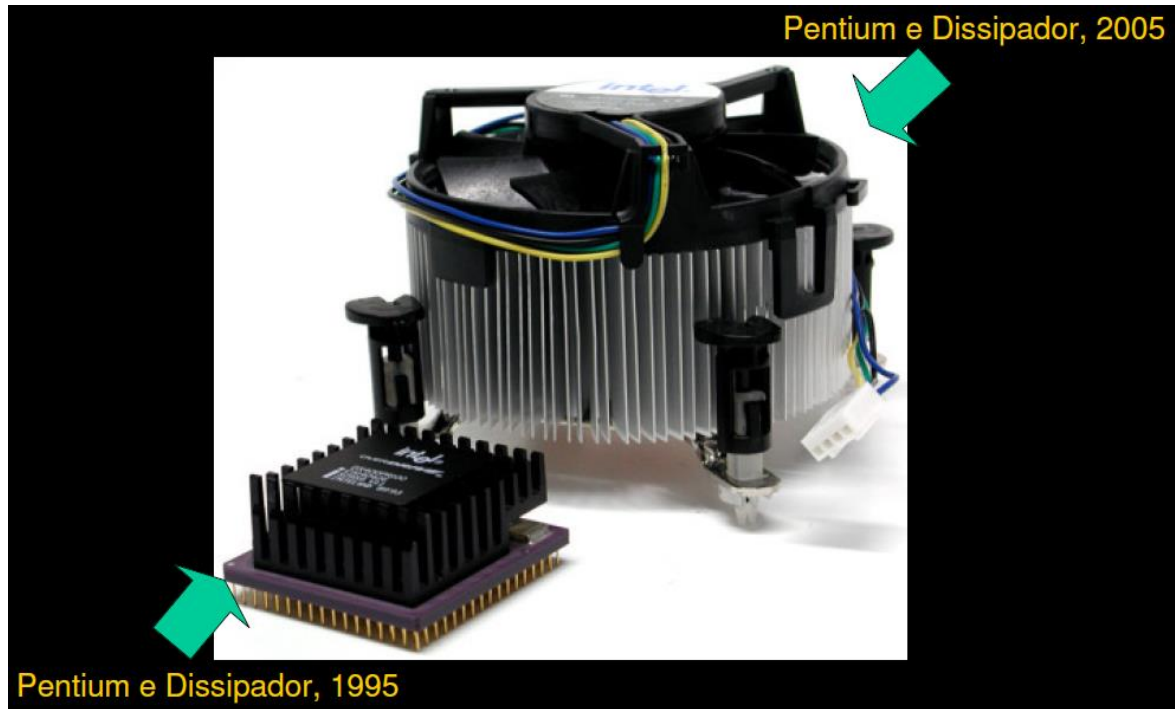
PROF. ISAAC

Paralelismo

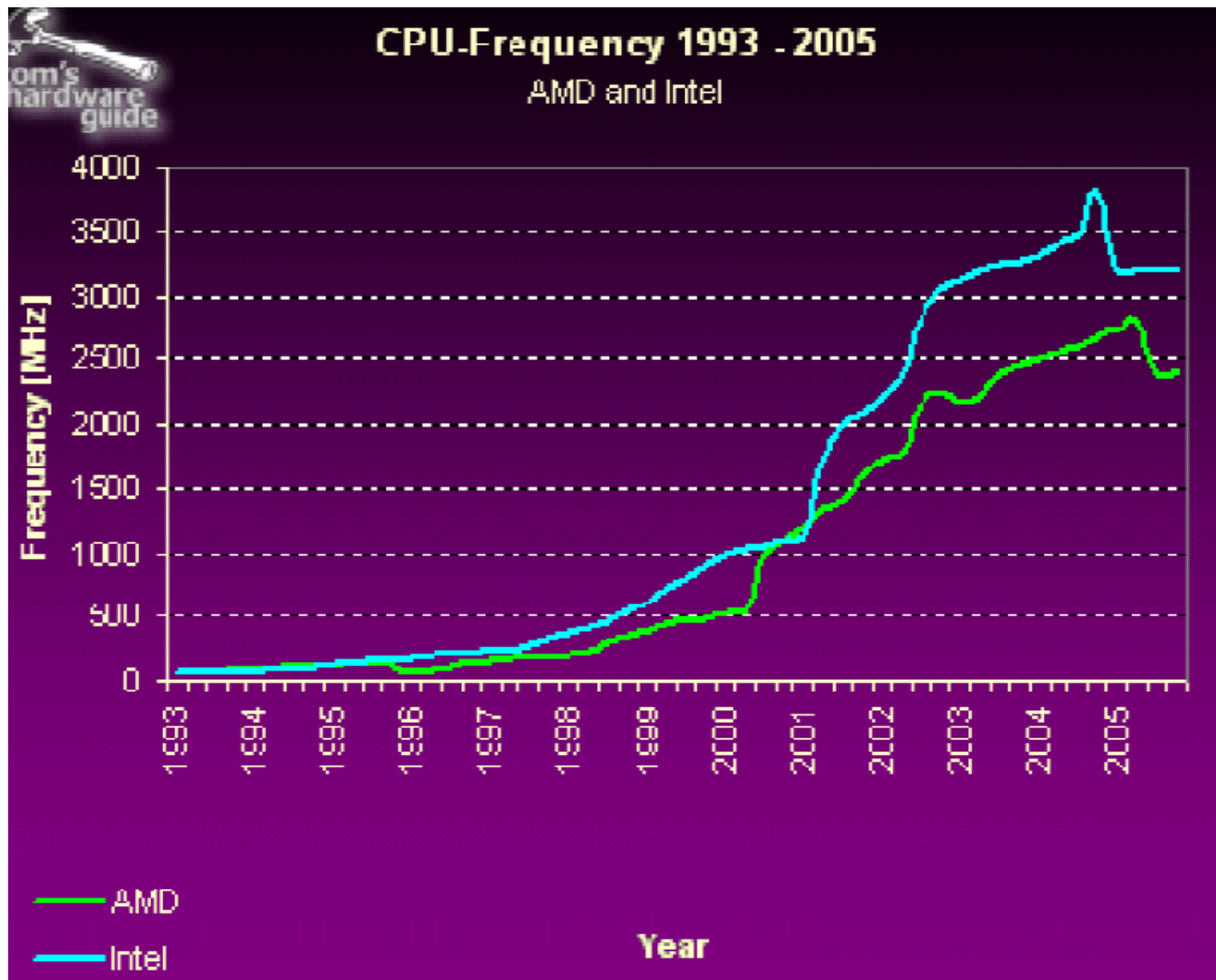
Paralelismo

Limitações

Com o desenvolvimento tecnológico a dissipação atingiu níveis intoleráveis impedindo novo aumento de frequência



Paralelismo



Formas de Paralelismo

- Paralelismo de instrução
 - Pipeline
 - Arquiteturas superescalares
- Multithread
- Adição de CPUs

Revisão: Paralelismo de Instrução

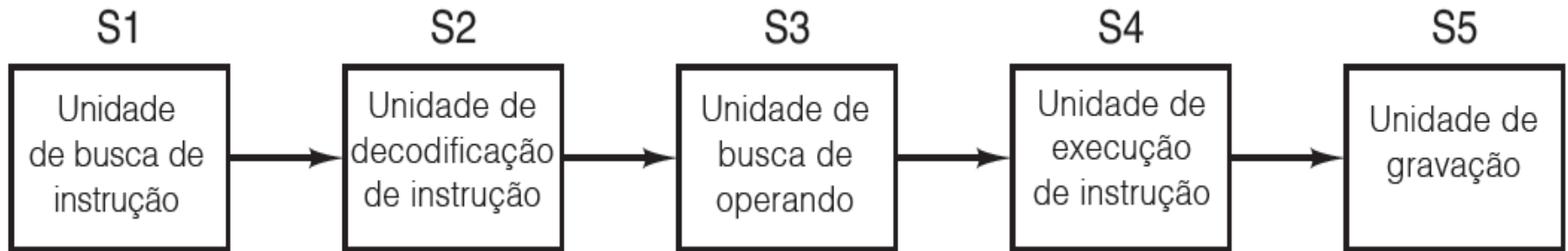
- Feito em hardware
- As instruções já são formadas para serem executadas ao mesmo tempo.
- Exemplos: Pipelines, Superescalares e Very Long Instruction Word (VLIW)

Revisão: Pipeline

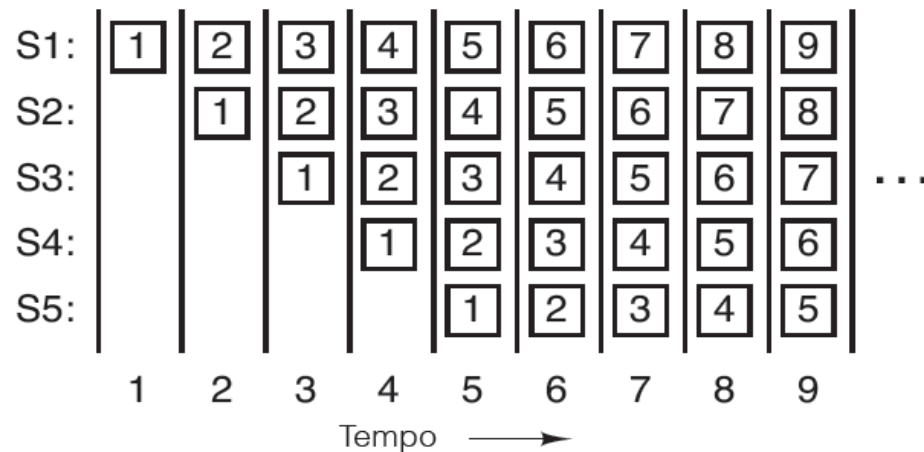
- Técnica onde várias instruções são sobrepostas na execução do programa.
- Uma forma de tentar aproveitar recursos livres em uma CPU.
- Executa partes do programa em paralelo.
- É utilizada para tornar as CPUs mais rápidas.

Revisão: Pipeline

Pipeline de cinco estágios.

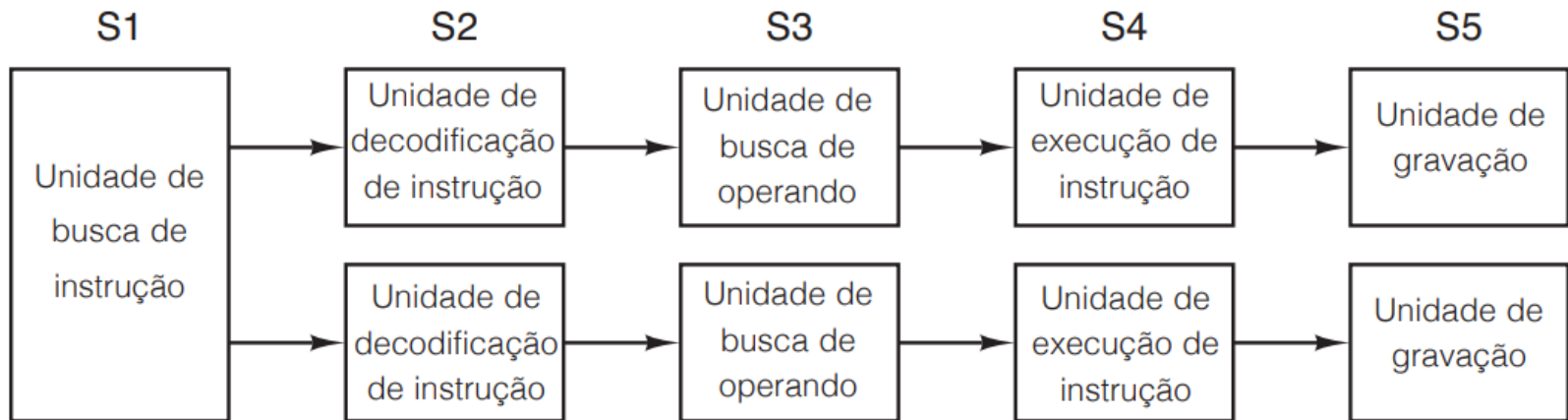


Estado de cada estágio como uma função do tempo.



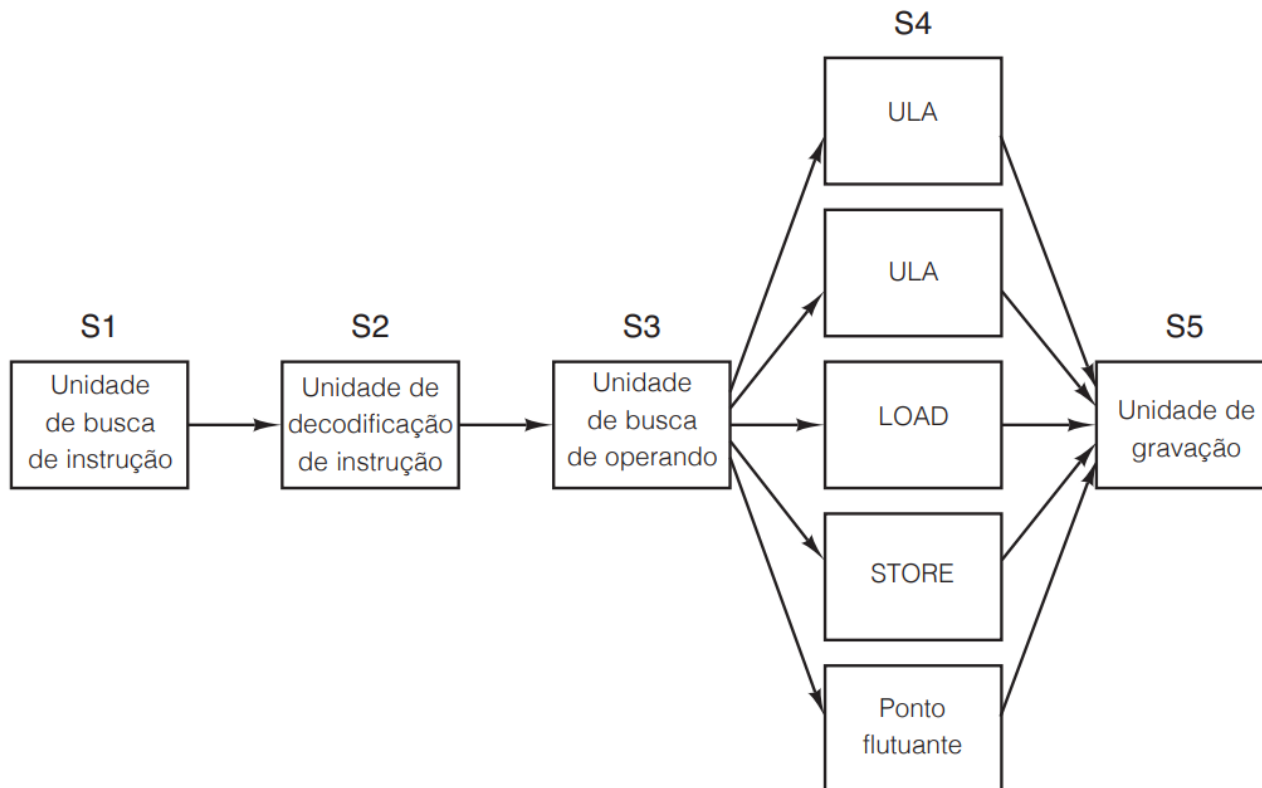
Arquiteturas Superescalar

- Se um pipeline é bom, então certamente dois pipelines são ainda melhores. A figura abaixo apresenta uma CPU com dois pipelines.



Arquiteturas Superescalar

- Uma outra abordagem também é utilizada em CPUs. A ideia básica é ter apenas um único pipeline, mas lhe dar várias unidades funcionais, conforme mostra a Figura:



Revisão: Multithreads

Duas formas de troca de threads:

1. Granulação fina: troca de thread a cada nova instrução.
2. Granulação grossa: troca entre as threads quando é necessário parar para esperar uma informação.

Exemplo

<u>T1</u>	<u>T2</u>
T1.1 MOV A,#10	T2.1 MOV A,#10
T1.2 MOV B,#20	T2.2 MOV B,#20
T1.3 MUL AB	T2.3 DIV AB
T1.4 MOV R0,A	T2.4 MOV R0,A
T1.5 MOV R1,B	T2.5 MOV R1,B

Tempo	T+0	T+1	T+2	T+3	T+4	T+5	T+6	T+7	T+8	T+9
Serial	T1.1	T1.2	T1.3	T1.4	T1.5	T2.1	T2.2	T2.3	T2.4	T2.5
MT G Fina	T1.1	T2.1	T1.2	T2.2	T1.3	T2.3	T1.4	T2.4	T1.5	T2.5
MT G Grossa	T1.1	T1.2	T1.3	T2.1	T2.2	T2.3	T1.4	T1.5	T2.4	T2.5

Formas de Paralelismo

- Paralelismo de instrução



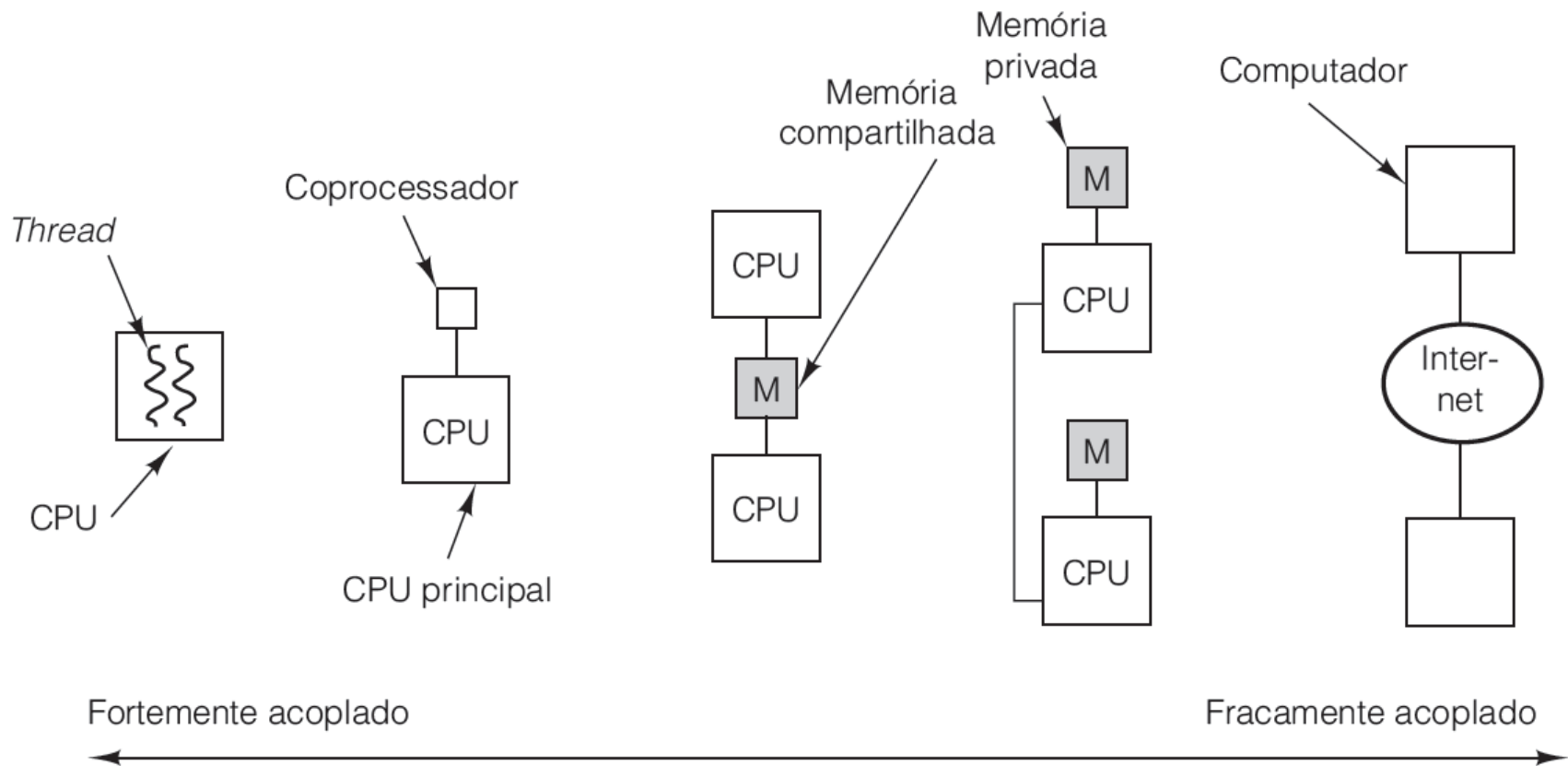
- Multithread



- Adição de CPUs

Arquiteturas de computadores paralelos

- Paralelismo no chip. Coprocessador. Multiprocessador. Multicomputador.

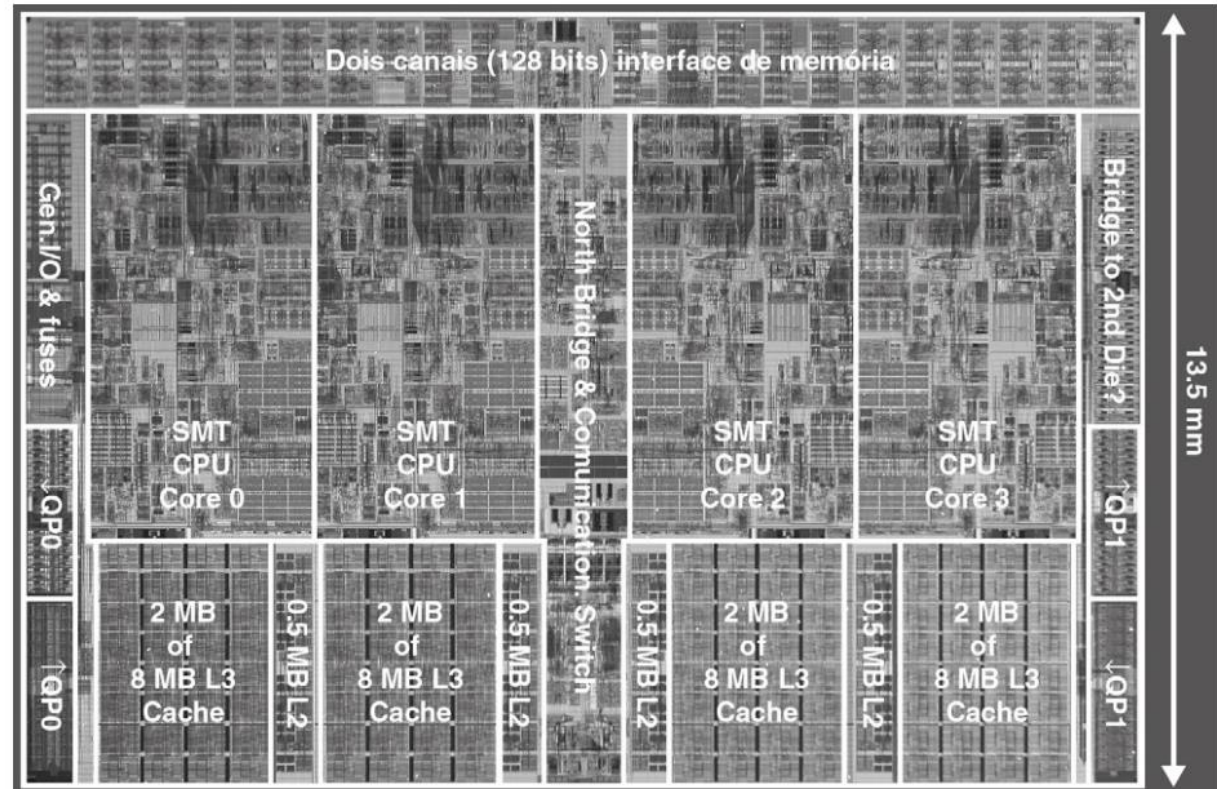
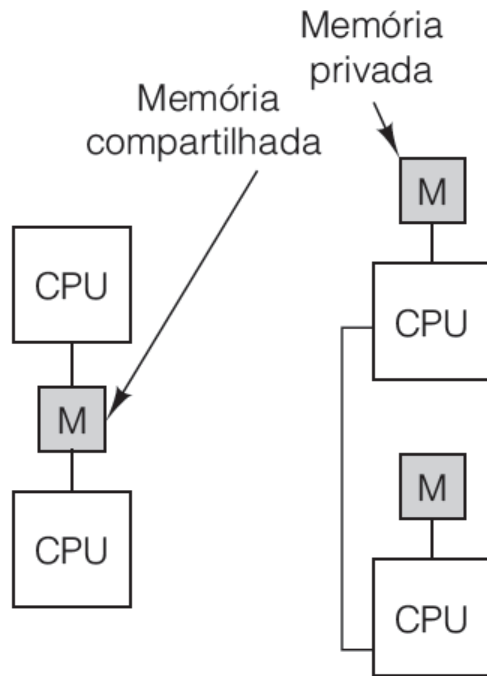


Arquiteturas de computadores paralelos

A Figura mostra a fotografia do die do Intel Nehalem (Intel Core i7 de 2008).

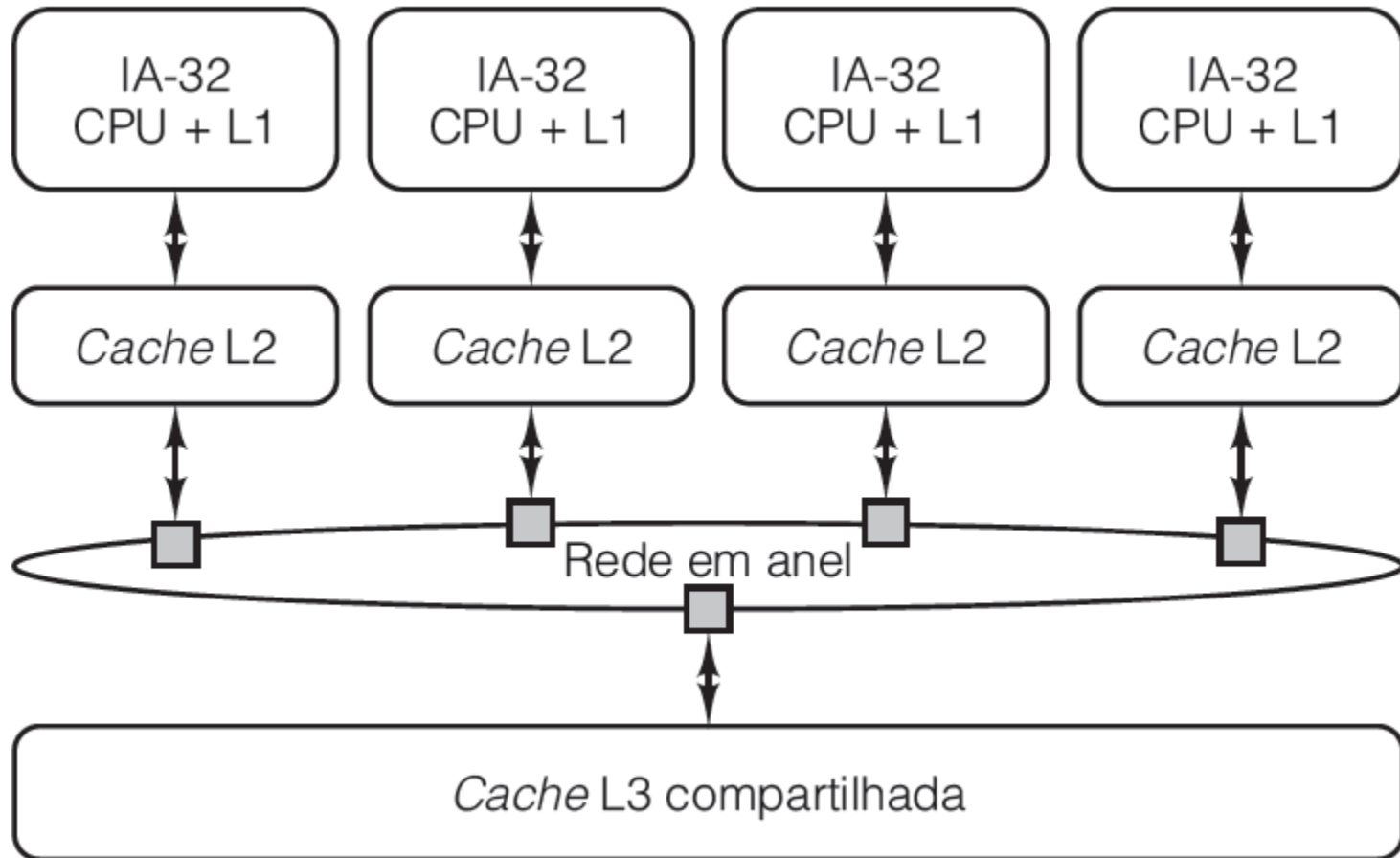
Ele contém 4 processadores (cores) que possuem, cada um, caches de instrução privados de 32 KB, e uma cache L2 de 512 KB.

Os 4 cores compartilham uma cache L3 de 8 MB. Os dois canais de memória de 128 bits são para a DRAM DDR3



Paralelismo no chip

- ARQUITETURA DO MULTIPROCESSADOR EM UM ÚNICO CHIP DO CORE I7.

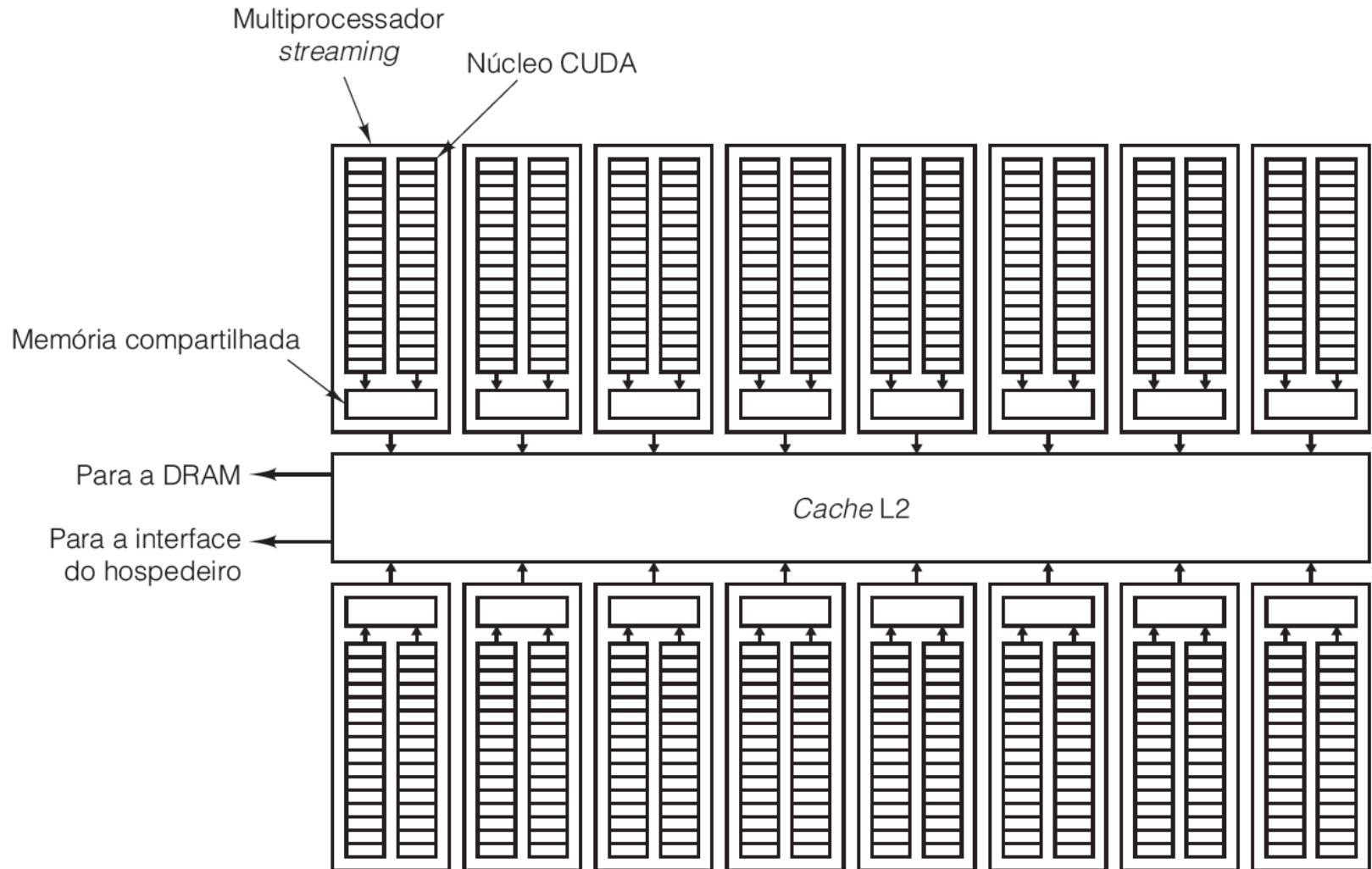


Processadores gráficos - GPU

Unidade de Processamento Gráfico

- 1 operação realizada em N dados.
- Paralelismo com vários núcleos e dentro do núcleo são diferentes.
- Bom para operações em vetores ou matrizes (imagens, redes neurais, etc).

Processadores gráficos - GPU

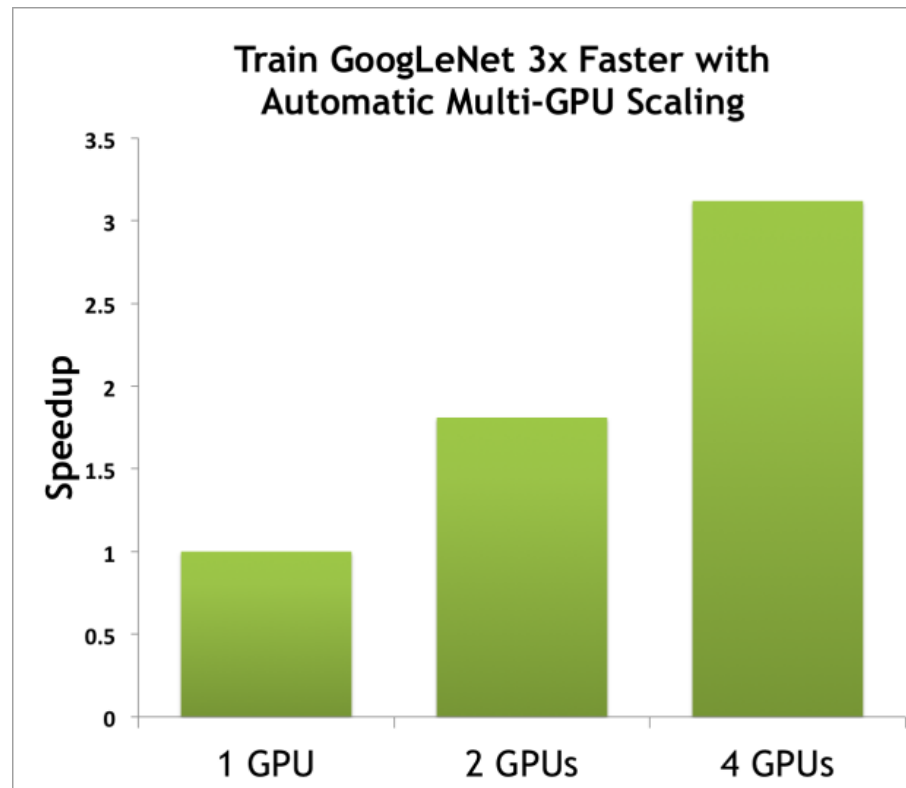


Arquiteturas de computadores paralelos

- Para enfrentar problemas cada vez maiores, os arquitetos de computadores estão recorrendo cada vez mais a computadores paralelos.
- Apesar de talvez não ser possível construir uma máquina com uma única CPU e um tempo de ciclo de 0,001 ns, pode ser perfeitamente viável produzir uma com 1.000 CPUs com um tempo de ciclo de 1 ns cada.
- Embora esse último projeto use CPUs mais lentas do que o primeiro, sua capacidade total de computação é teoricamente a mesma.

Arquiteturas de computadores paralelos

- Exemplo de treinamento de um algoritmo de *Deep Learning* usando múltiplas GPUs



Classificação dos Computadores Paralelos

A classificação de **Flynn** (1972) é a mais utilizada. Ele se baseia nas possíveis unicidade e multiplicidade dos fluxos de instruções e de dados para definir quatro tipos de arquiteturas:

- **SISD** (Single Instruction Stream, Single Data Stream):
Computadores sequenciais um único processador
- **SIMD** (Single Instruction Stream, Multiple Data Streams):
Computadores vetoriais e matriciais
- **MISD** (Multiple Instruction Streams, Single Data Stream):
Não há muitos exemplos da existência desta arquitetura.
- **MIMD** (Multiple Instruction Streams, Multiple Data Streams):
Arquiteturas com múltiplos processadores independentes

Taxonomia de Flynn

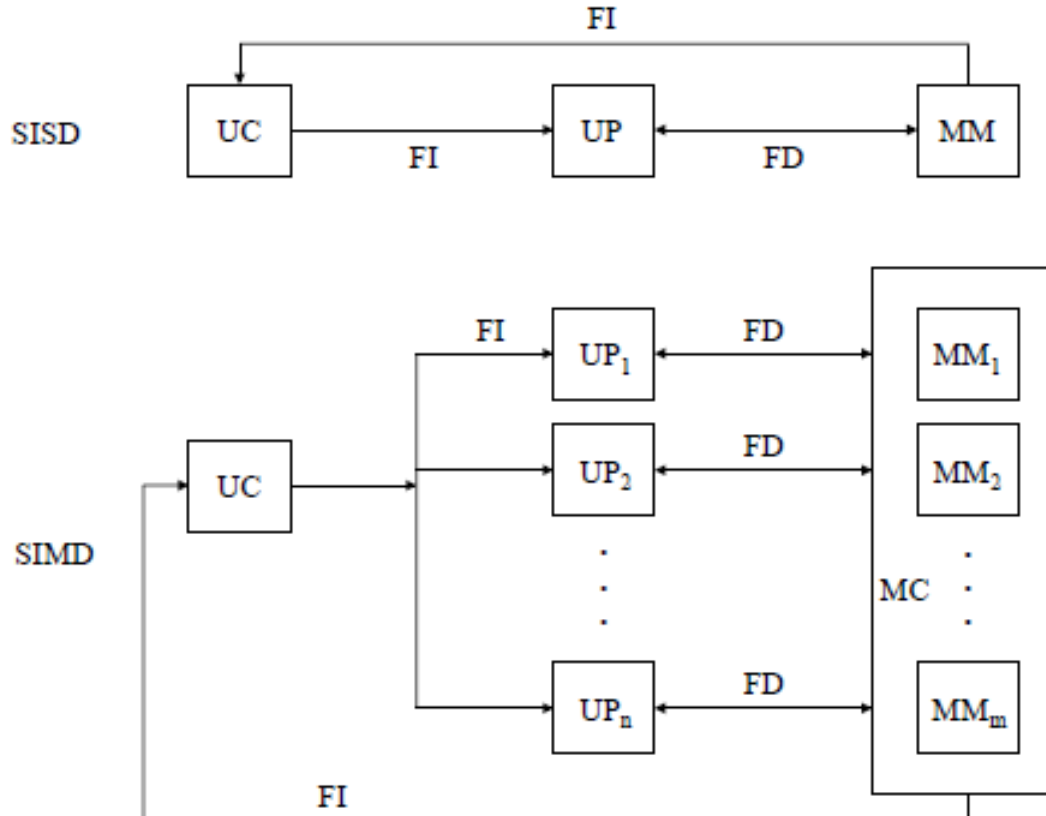
A classificação de **Flynn** é baseada na multiplicidade do fluxo de instruções e de dados num computador, em que o principal processo computacional é a execução de uma sequência de instruções sobre um conjunto de dados.

Flynn definiu quatro organizações de máquina:

- Fluxo único de instrução-fluxo único de dados (**SISD**);
- Fluxo único de instrução-fluxo múltiplo de dados (**SIMD**);
- Fluxo múltiplo de instrução-fluxo único de dados (**MISD**);
- Fluxo múltiplo de instrução-fluxo múltiplo de dados (**MIMD**).

Taxonomia de Flynn

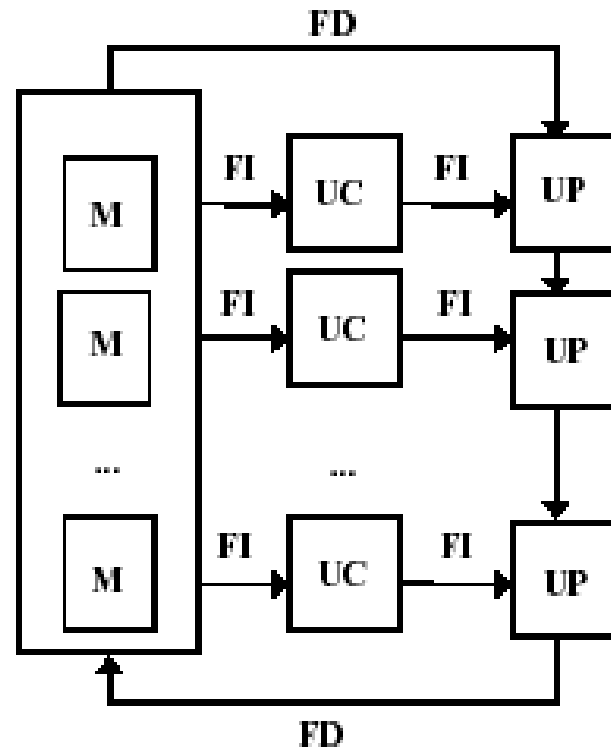
- Fluxo único de instrução-fluxo único de dados (**SISD**);
- Fluxo único de instrução-fluxo múltiplo de dados (**SIMD**);



Taxonomia de Flynn MISD

- Fluxo múltiplo de instrução-fluxo único de dados (MISD);

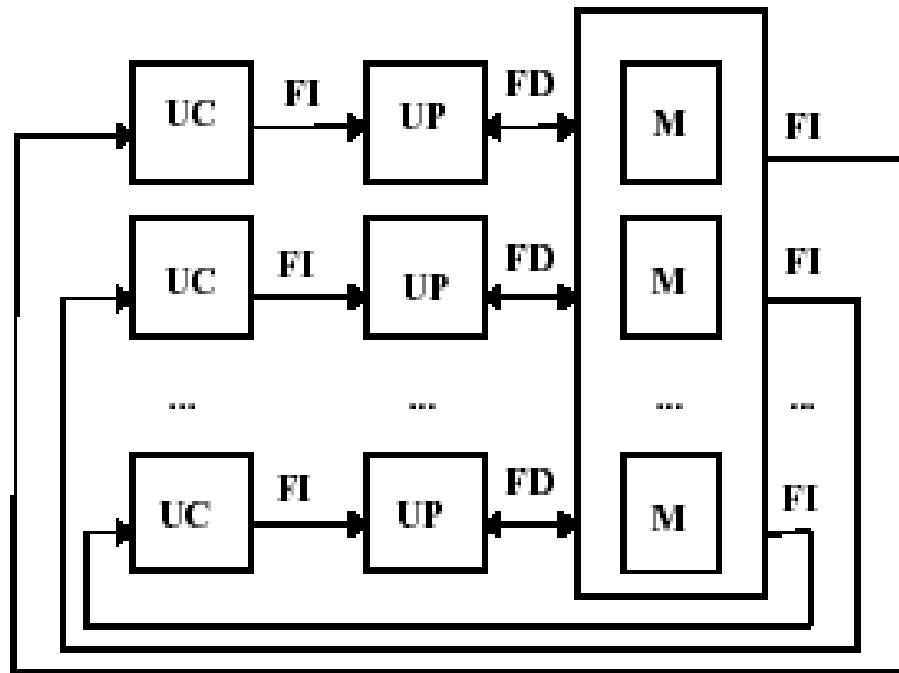
UP = Unidade de Processamento
FD = Fluxo de Dados
M = Memória
UC = Unidade de Controle
FI = Fluxo de Instruções



Taxonomia de Flynn MIMD

- Fluxo múltiplo de instrução-fluxo múltiplo de dados (MIMD).
-

UP = Unidade de Processamento
FD = Fluxo de Dados
M = Memória
UC = Unidade de Controle
FI = Fluxo de Instruções

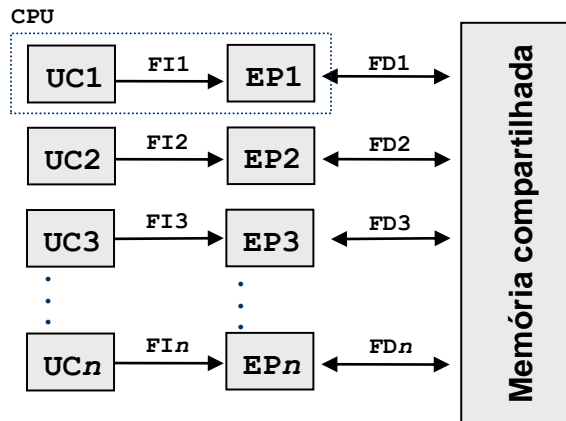


Taxonomia de Flynn MIMD

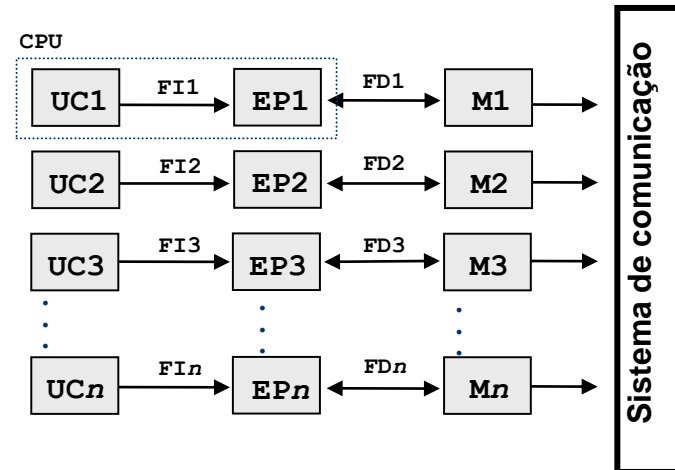
Fluxo múltiplo de instrução-fluxo múltiplo de dados (**MIMD**).

Na classe **MIMD** está a maioria das arquiteturas paralelas da atualidade. De acordo com o método de comunicação entre os processadores, a classe **MIMD** pode ser dividida em:

MIMD compartilhada



MIMD distribuída



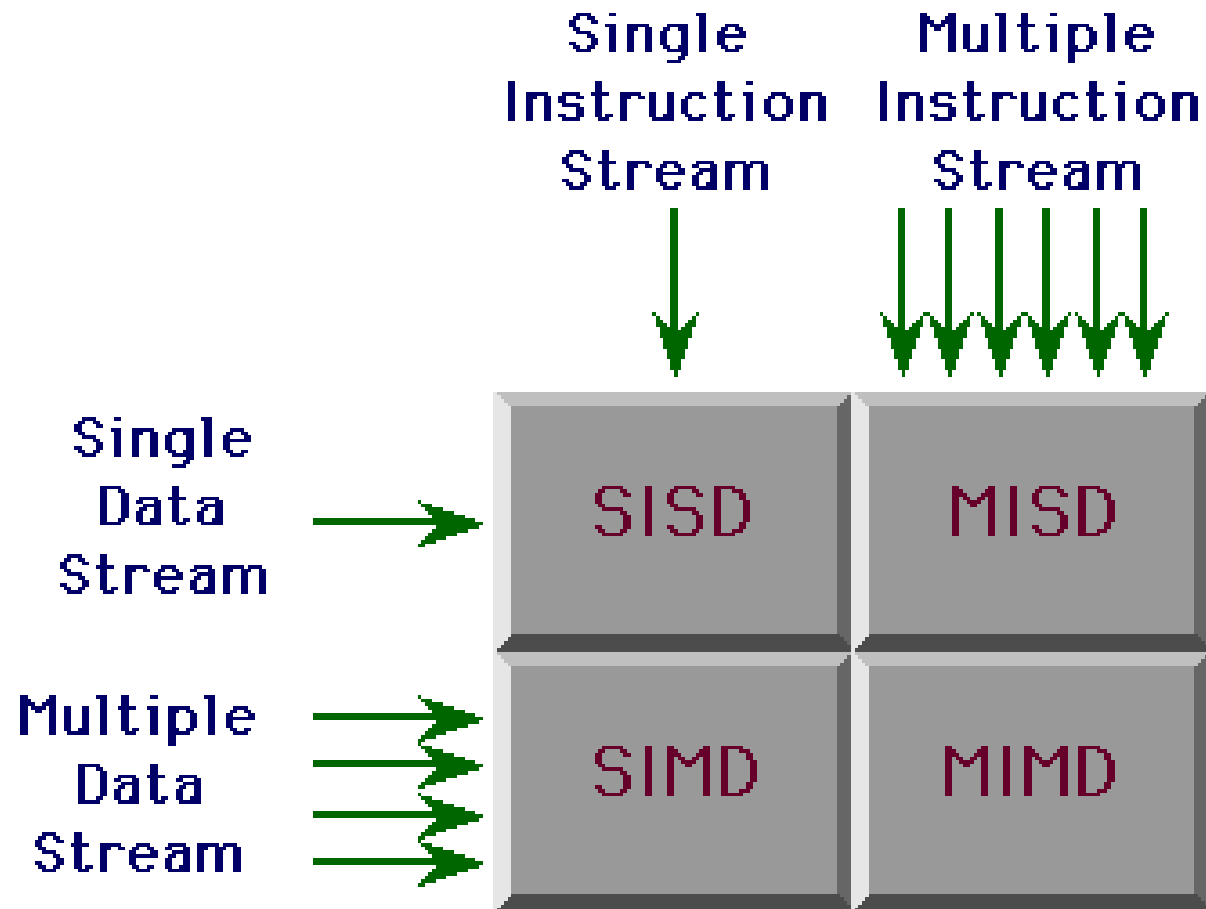
SISD é encontrada na maioria dos computadores sequenciais disponíveis. As instruções são executadas sequencialmente, mas podem ser superpostas (*pipeline*). Pode apresentar mais de uma unidade funcional, mas todas sob a supervisão de uma única unidade de controle.

SIMD corresponde ao arranjo de processadores, onde há múltiplos elementos processadores supervisionados pela mesma unidade de controle. Todas as unidades processadoras recebem a mesma instrução distribuída pela unidade de controle, mas operam sobre diferentes conjuntos de dados. A memória compartilhada pode conter múltiplos módulos.

MISD apresenta n unidades processadoras, cada uma recebendo instruções distintas, operando sobre o mesmo conjunto de dados. Não há implementação deste tipo de arquitetura.

MIMD corresponde a sistemas multiprocessadores. Diz-se que um sistema é fortemente acoplado se o grau de interações entre os processadores é muito alto. Caso contrário, é dito fracamente acoplado, sendo este o mais comum.

Taxonomia de Flynn



Exercícios

1. As GPUs para aumentar seu desempenho executam uma mesma instrução em um conjunto de dados em paralelo. Por exemplo, uma única instrução de soma pode ser usada para somar duas matrizes (a soma será executada em paralelo para cada posição das matrizes).

Esse tipo de processamento paralelo é descrito, na classificação de Flynn, como.

- a) SDMI
- b) SIMD
- c) SISD
- d) MIMD
- e) MISD

Exercícios

2. Considerando os diferentes tipos de organização utilizadas para implementar processamento paralelo, qual afirmação está correta:
- a) Uma das desvantagens dos Clusters é a dificuldade de implementação, custo por utilizar somente supercomputadores, e se um nó falha o sistema todo falha.
 - b) O processamento paralelo com SMP requer computador com mais de um processador com as mesmas características, sendo que os processadores compartilham o barramento e a memória. Os programas podem ser desenvolvidos com o uso de multithreading ou múltiplos processos.
 - c) Na arquitetura paralela baseada em MISD, um único fluxo de instruções opera sobre um único fluxo de dados. Apesar de os programas serem organizados através de instruções sequenciais, elas podem ser executadas em pipelining, de forma sobreposta em diferentes estágios.
 - d) A arquitetura paralela baseada em SIMD envolve o processamento de múltiplos dados por parte de múltiplas instruções. Várias unidades de controle comandam suas unidades funcionais que têm acesso a vários módulos de memória, caracterizando as arquiteturas massivamente paralelas.

Exercícios

3. Considerado a taxonomia clássica de Flynn, os processadores de vários núcleos podem ser classificados como.
- a) SISD
 - b) SIMD
 - c) MIMD
 - d) MISD

HPC - High Performance Computing

O que é High Performance Computing?

Computação de alto desempenho

HPC — High Performance Computing

A **HPC** lida com supercomputadores e seus softwares, especialmente para aplicações científicas, que envolvem grandes quantidades de dados, cálculos de vetores e matrizes e uso de algoritmos paralelos.

HPC - High Performance Computing

O que é High Performance Computing?

- Uso de hardware de alto desempenho.
- Supercomputing.
- Clusters.
- GPUs.
- Clusters de GPUs.

HPC - High Performance Computing

Uso de HPC para solucionar problemas – qualquer problema.

HPC = High
Performance
Computing

HPC = computação paralela, distribuída, GPU, clusters, ...

Abordagens para HPC

A maioria das abordagens para a HPC utiliza Supercomputadores ou Clusters.

Supercomputadores:

- Um computador na fronteira da capacidade de processamento e desempenho, geralmente com memória compartilhada.

Clusters:

- Conjunto de computadores conectados por rede de alta velocidade.

Programando em HPC

Como usar os supercomputadores ou os clusters?

- Dividindo o trabalho em diversos núcleos de processamento.

Necessidade de programação paralela ou distribuída.

Clusters

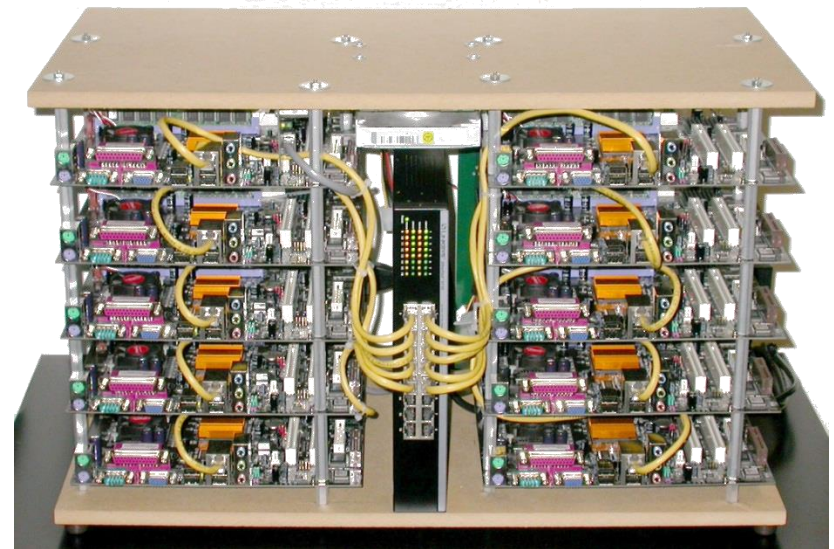
É um conjunto de computadores, os quais são ligados em rede e trabalham como se fossem uma única máquina de grande porte.

Os clusters se popularizaram nos anos 1980 por 3 motivos:

- microprocessadores de alta performance, redes de alta velocidade, e ferramentas para computação distribuída.

Clusters

É um grupo de computadores completos/autônomos (usualmente chamados nós) interconectados, que podem trabalhar juntos, como um recurso de computação unificado, criando a ilusão de uma máquina única. Os nós são geralmente conectados através de uma porta de E/S (geralmente interfaces de rede) de alto desempenho. Atualmente eles são utilizados com sistemas gerenciadores de bancos de dados, com servidores WEB e, principalmente, para proc. paralelo.



Cluster - Tianhe-1A



7.168 GPUs Nvidia Tesla

14.336 processadores Intel Xeon

atinge um desempenho de 2,507
petaflops no benchmark Linpack

a performance do equipamento
equivale, em média, à de 175 mil
laptops ou 50 mil CPUs

Localizado em Tianjin - China

Cluster - Tianhe-2



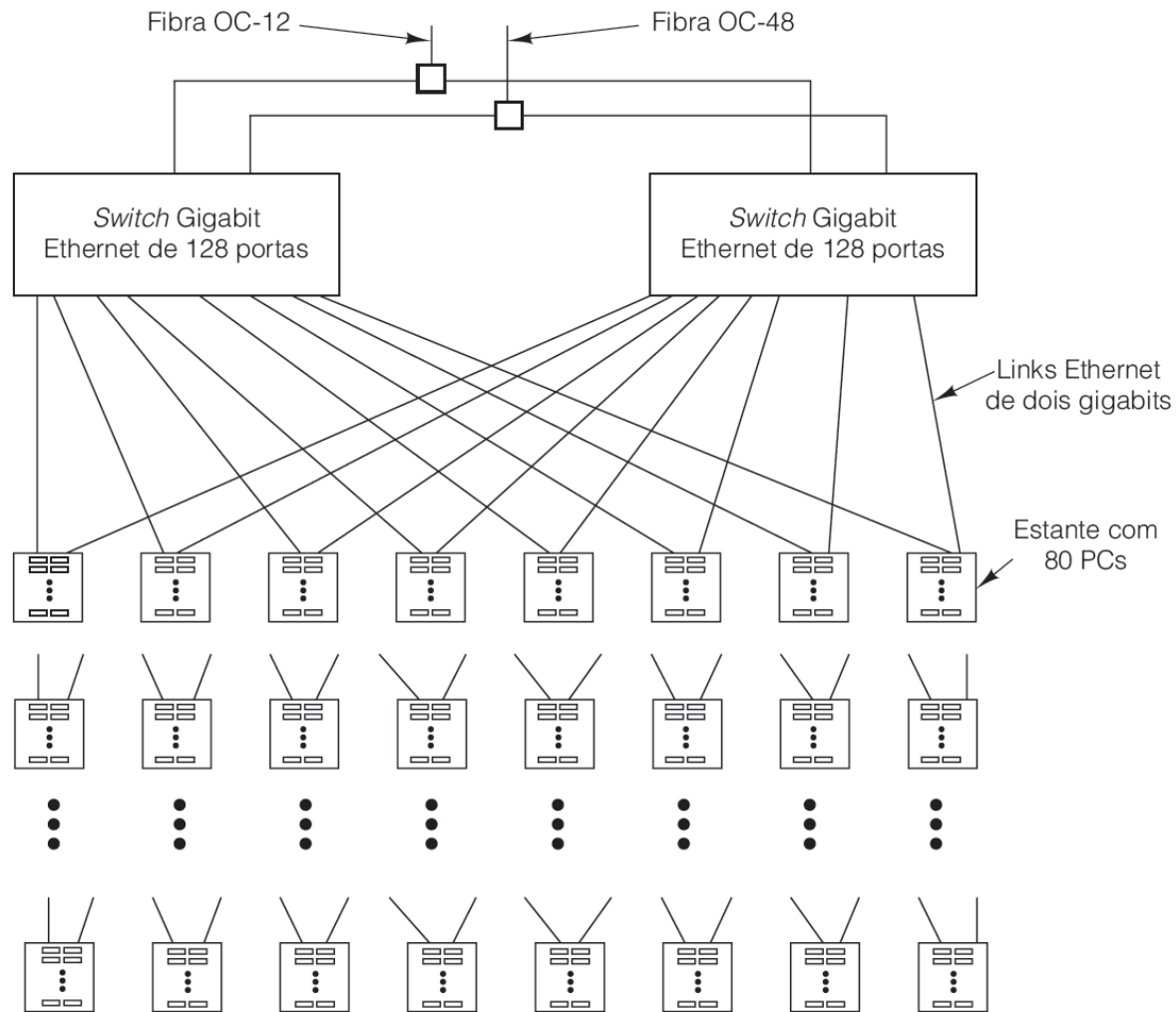
32,000 Intel Xeon E5-2692

48,000 Xeon Phi 31S1P

atinge um desempenho de 33.86
petaflops

Localizado em Guangzhou - China

Cluster Google Típico



Clusters

- ❑ São utilizados quando os conteúdos são críticos, ou seja, eles devem estar sempre disponíveis para o usuário ou sistema.
- ❑ Quando se necessita de uma velocidade maior de processamento para a realização de um trabalho.
- ❑ Confiabilidade.
- ❑ Facilidade de implementação.
- ❑ Baixo custo em relação aos supercomputadores.
- ❑ O cluster pode aumentar de tamanho, bastando adicionar mais máquinas.
- ❑ A velocidade total dependerá da velocidade individual de cada nó e da rede que estão interligados.
- ❑ Utiliza ao máximo o poder de processamento de cada nó.

Tipos de Cluster

De alto desempenho: permite uma grande carga de processamento, geralmente na casa dos gigaflops.

De alta disponibilidade: resistente a falhas de hardware, software e energia, tem o objetivo de manter os serviços disponibilizados o máximo de tempo possível.

Para balanceamento de carga: esse tipo de cluster tem como função controlar a distribuição equilibrada do processamento.

Tipos de cluster

Alta disponibilidade (**High Availability - HA**)

Caso um nó deixe de funcionar (failover), os outros redistribuem o trabalho entre eles;

Trabalham com redundância de informações;

Muito utilizado em bancos de dados críticos, correio e servidores de arquivos

Tipos de cluster

Balanceamento de carga (Load Balancing - LB)

Monitora e distribui o tráfego entrante entre os nós

Ganho de desempenho

Ao receber os pedidos, o cluster transfere a tarefa para os nós

Existem 3 tipos de algoritmos para criar esse balanceamento

Least connections

- Redireciona o pedido de entrada para o nó que estiver processando menos pedidos

Round Robin

- Redireciona requisições sempre para o próximo nó, de forma circular

Weighted Fair

- Redireciona pedidos dependendo do poder de processamento de cada um dos nós

Cluster Beowulf

O tipo mais simples:

- Beowulf, constituído por diversos nós escravos gerenciados por um só computador, geralmente PCs comuns.
- Utiliza rede ethernet or gigabit ethernet.
- Off-the-shelf components.

O maior problema:

- Comunicação entre os nós.

Supercomputadores x Clusters

Supercomputadores

- Maior custo de implementação e pesquisa de tecnologia
- Utilizado para cálculos complexos e únicos. Ex: Previsão do tempo e simulações nucleares

Clusters

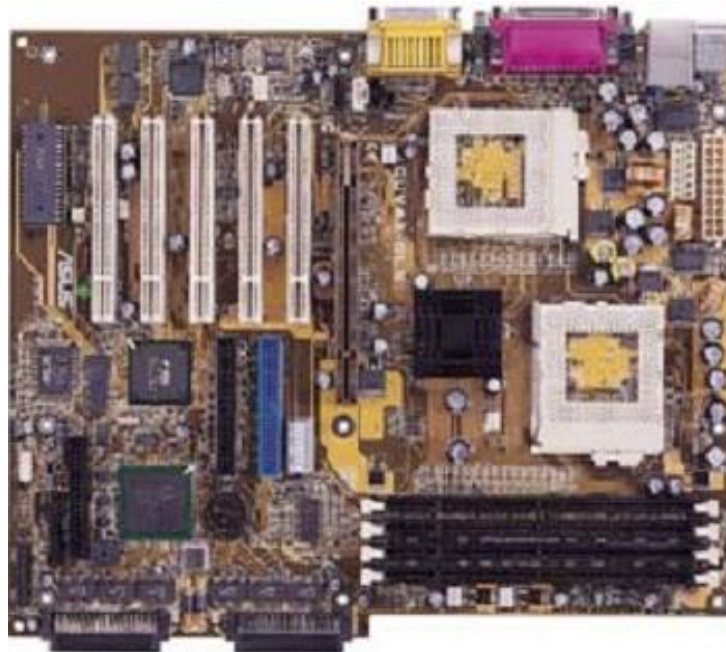
- Baixo Custo
- Utilizado na realização de trabalhos que podem ser divididos em pequenas tarefas independentes

SMP (Symmetric Multi Processor)

São computadores com mais de um processador com as mesmas características.

Os computadores compartilham o mesmo BUS e a mesma memória

- Vantagem: Fácil de programar
- Desvantagem: Requer máquinas com dois ou mais processadores(são máquinas mais caras).



Cluster x SMP

CLUSTER:

- Maior divisibilidade de processamento
- Maior eficiência
- Se uma CPU falhar, o nó é desabilitado

SMP (Symmetric Multiprocessor):

- Maior volume de processamento
- Maior velocidade de comunicação
- Se uma CPU falhar, o sistema sai do ar

Respostas

Resposta dos exercícios:

1 - b

2 - b

3 - c

Bibliografia

Stallings, Willian. Arquitetura e Organização de Computadores. 10ª Ed, Pearson, 2017.

Murdocca, Miles J., and Vincent P. Heuring. Introdução à arquitetura de computadores. Elsevier, 2001.

David A.Patterson & John Hennessy. Organização e projeto de computadores: A interface de Hardware e Software. 4ª Ed. Elsevier. 2014.