Arquitetura de Computadores

Uma Introdução

Gabriel P. Silva – José Antonio Borges

Arquiteturas Avançadas

Capítulo 8

8.1 Pipeline

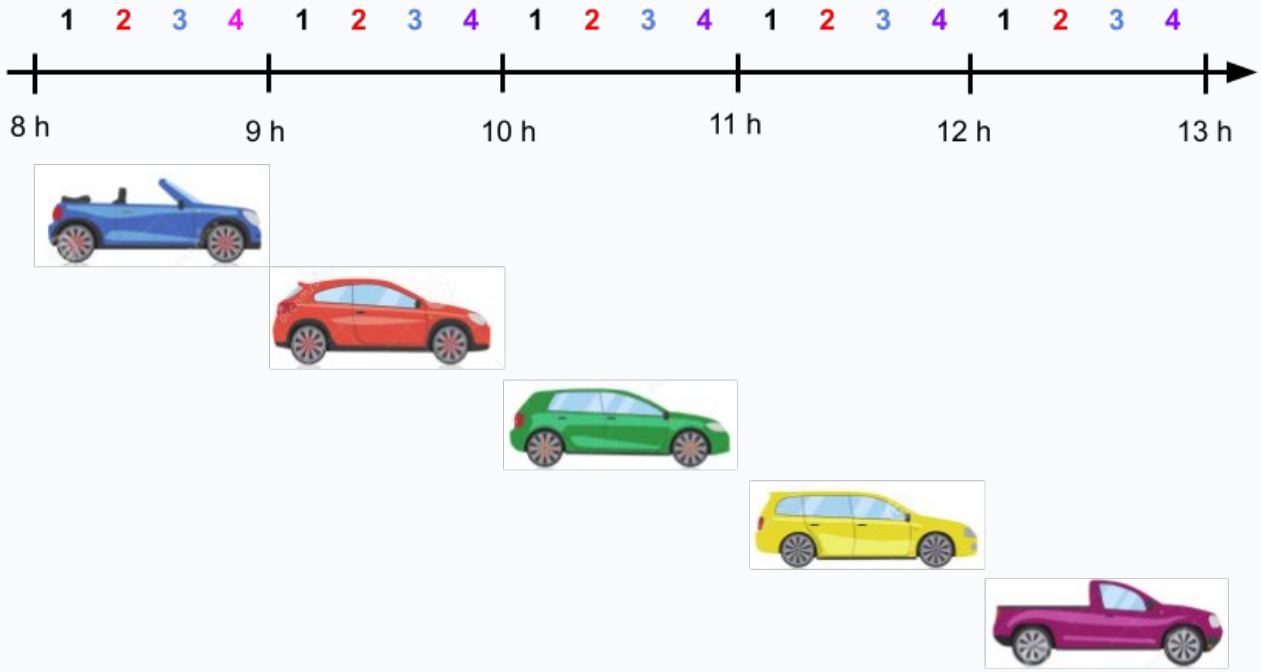
Pipeline

- O pipeline é uma técnica de implementação de processadores que permite a sobreposição temporal das diversas fases de execução das instruções.
- Essa técnica aumenta o número de instruções executadas simultaneamente e a taxa de instruções iniciadas e terminadas por unidade de tempo.
- Contudo, o pipeline não reduz o tempo gasto para completar cada instrução individualmente, podendo até mesmo aumentá-lo em alguns casos.

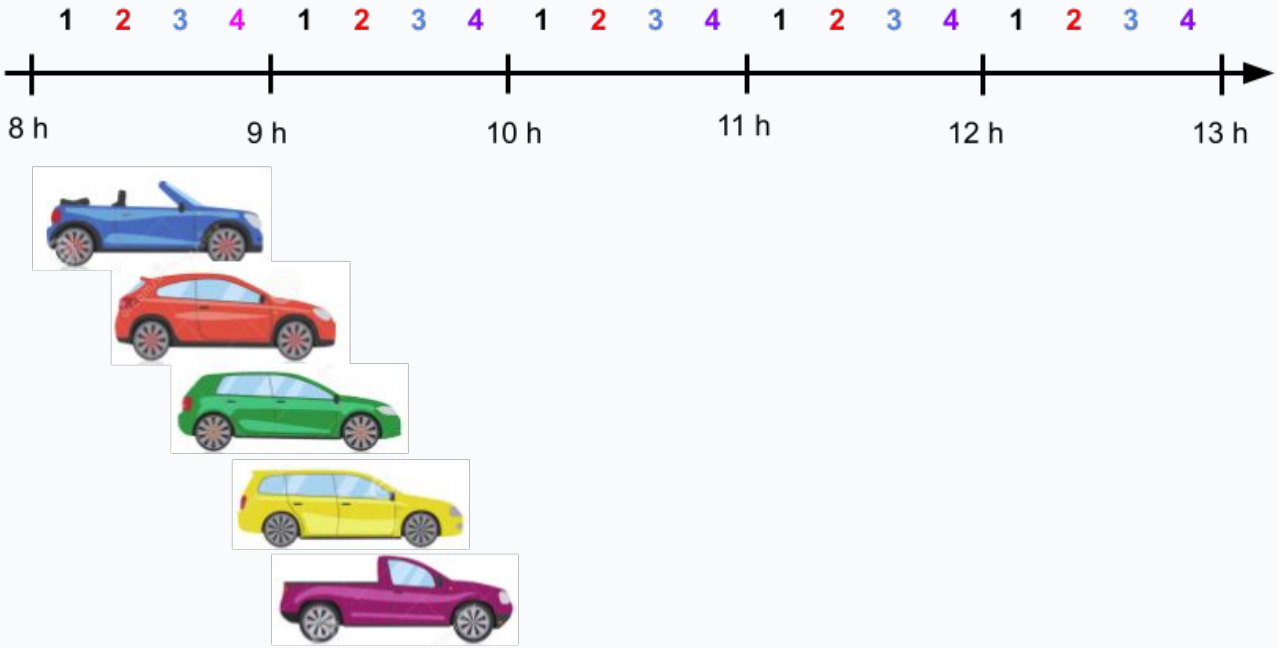
Etapas de um Lava Jato

- A lavagem de um carro pode ser dividida em quatro etapas:
  - Aplicação de desengordurante na lataria, desengraxante nas rodas, enxágue.
  - Aplicação de xampu, enxágue, secagem.
  - Aspiração do interior, limpeza dos vidros.
  - Lavagem dos tapetes, aplicação de silicone nas partes emborrachadas e pretinho nos pneus, troca da bolsinha de lixo.

Sem Pipeline



Com Pipeline



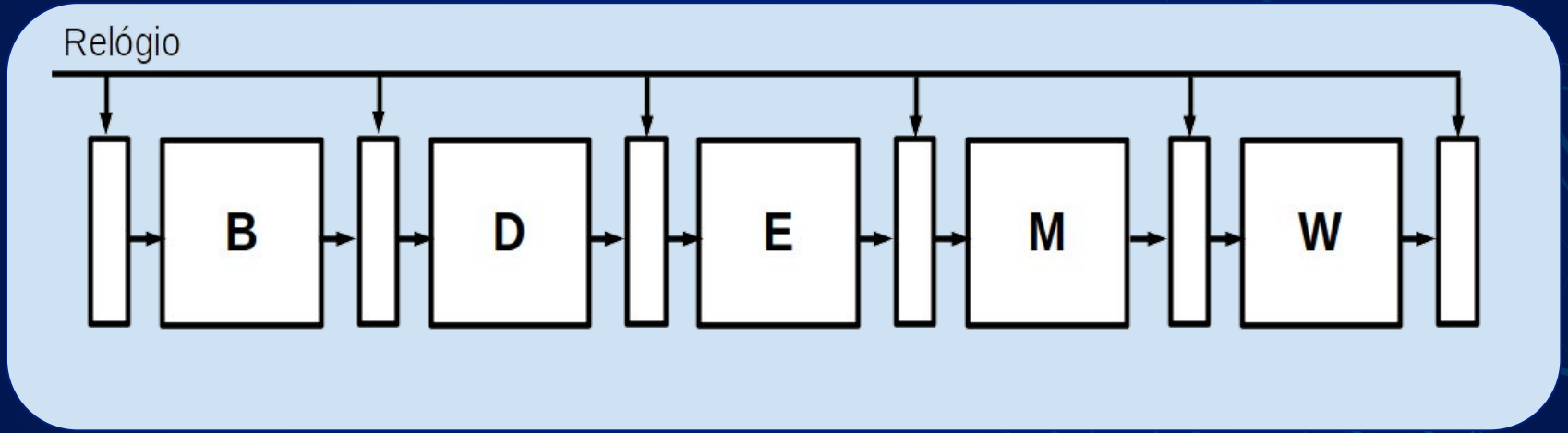
Características do Pipeline

- Com o uso da técnica de pipeline, a lavagem de um carro, individualmente, continuará levando uma hora para ser realizada.
- Ou seja, a técnica de pipeline não melhora o tempo de execução de cada tarefa individualmente, mas melhora o rendimento ou a produtividade de todo o sistema.
- Várias tarefas podem ser executadas simultaneamente desde que utilizem recursos diferentes.
- A aceleração potencial máxima, ou seja, o ganho de tempo que podemos ter, é igual ao número de estágios do pipeline.

Pipeline de Instruções

- Busca da instrução na memória (B).
- Leitura dos registradores e decodificação da instrução (D).
- Execução da instrução / cálculo do endereço de memória (E).
- Acesso a um operando na memória (M).
- Escrita do resultado em um registrador (W).

Pipeline de Instruções



Ciclo de Relógio

- A cada novo ciclo de relógio uma instrução é passada de um estágio para outro do pipeline.
- O relógio, por ser periódico, possui uma frequência e tempo de ciclo definidos.
- Quanto maior a frequência (f), menor o tempo de ciclo (T) do relógio ($T = 1/f$).
- Nos processadores modernos a frequência é expressa em MHz (10^6 Hz) ou GHz (10^9 Hz) e o tempo de ciclo de relógio em ns (10^{-9} s) ou ps (10^{-12}s).

Pipeline de Instruções

Estágio	Ciclo de Clock									
Pipeline	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
B	Inst.1	Inst.2	Inst.3	Inst.4	Inst.5	Inst.6	Inst.7	Inst.8	Inst.9	Inst.10
D		Inst.1	Inst.2	Inst.3	Inst.4	Inst.5	Inst.6	Inst.7	Inst.8	Inst.9
E			Inst.1	Inst.2	Inst.3	Inst.4	Inst.5	Inst.6	Inst.7	Inst.8
M				Inst.1	Inst.2	Inst.3	Inst.4	Inst.5	Inst.6	Inst.7
W					Inst.1	Inst.2	Inst.3	Inst.4	Inst.5	Inst.6

Características do Pipeline

- A duração do ciclo de relógio de um processador deve ser maior ou igual ao tempo que o estágio mais lento do pipeline leva para realizar suas operações.
- Deve-se procurar dividir a execução da instrução em estágios que tenham o mesmo tempo de execução, para otimizar o desempenho do pipeline.

Características do Pipeline

- O pipeline deve ser mantido sempre ‘’cheio’’, ou seja, com instruções úteis em todos os seus estágios, para que o seu desempenho máximo seja alcançado.
- Cada instrução, individualmente, gasta um tempo igual ou maior para ser executada em um processador com pipeline quando comparado à execução sem pipeline.

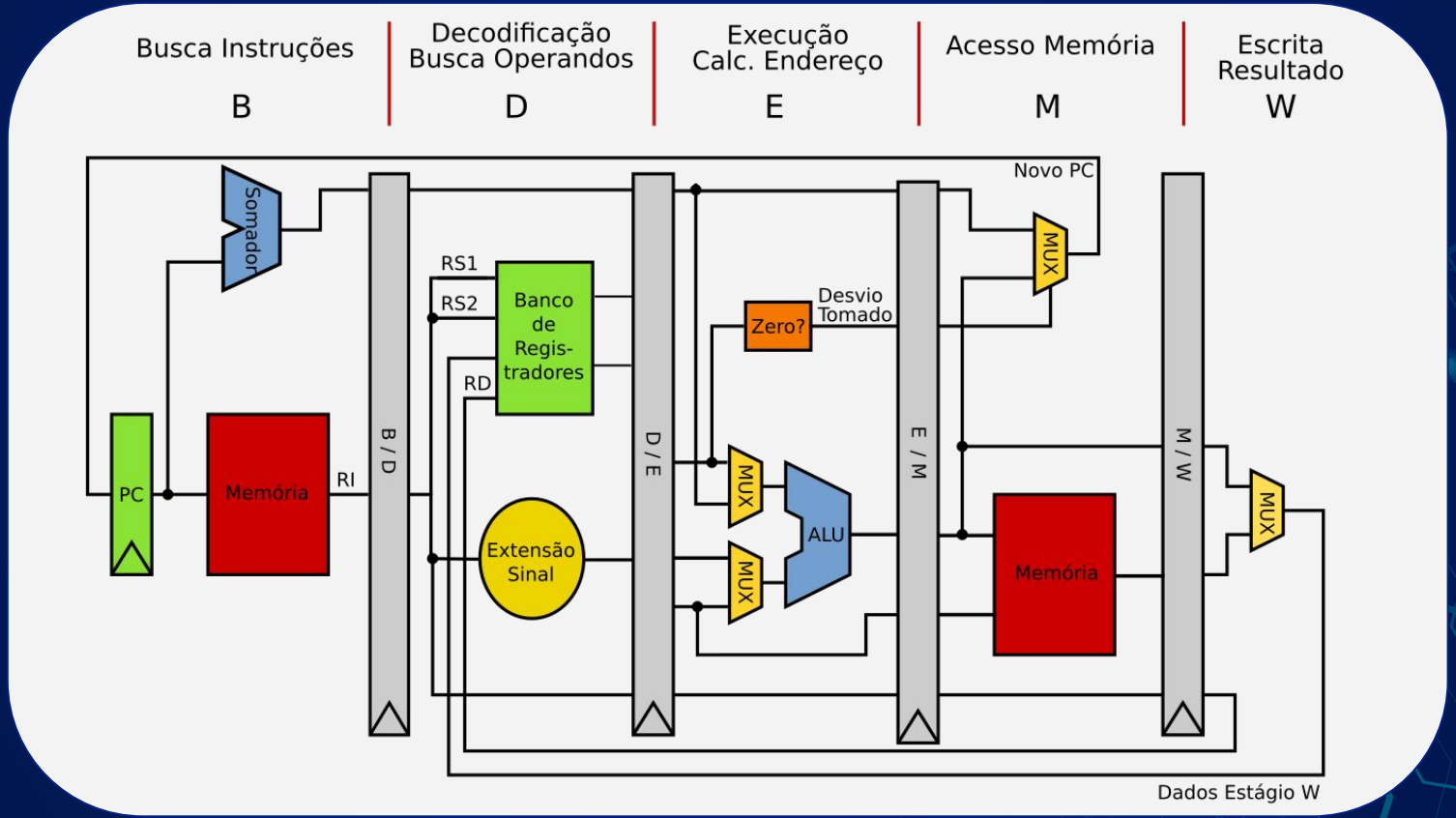
Processador MIPS R2000

- O MIPS R2000, lançado em 1986, foi a primeira implementação comercial de um processador RISC, utilizando a arquitetura MIPS, contendo trinta e dois registradores inteiros de 32 bits e todas as instruções e endereços também tinham 32 bits.
- Tinha um pipeline de 5 estágios com taxa de execução próxima de uma instrução por ciclo, um marco para os processadores da época, onde as paradas do pipeline e eventos de exceção eram tratados com precisão e eficiência.

Pipeline MIPS R2000

- B ➔ busca da instrução (I-Cache)
- D ➔ leitura dos operando necessários dos registradores do processador enquanto faz a decodificação da instrução
- E ➔ realiza a operação requerida nos operandos da instrução
- M ➔ acesso à memória (D-Cache)
- W ➔ escreve de volta o resultado no banco de registradores

Pipeline Processador MIPS

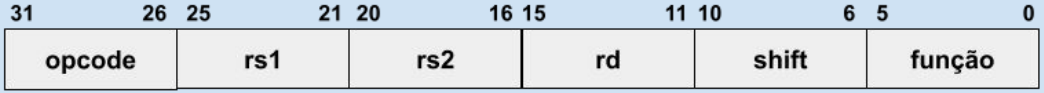


Instruções MIPS R2000

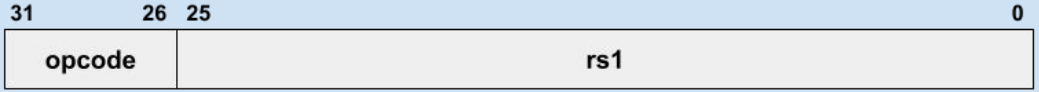
- Cada instrução do R2000 consistia de uma única palavra de 32 bits em um endereço alinhado (múltiplo de 4) na memória.
- Havia três tipos básicos de instrução, R, J e I.
- Esta abordagem simplifica bastante a decodificação das instruções, sendo que as operações e modos de endereçamento mais complicados (e menos frequentes) podiam ser sintetizados pelo compilador.

Formato da Instrução MIPS

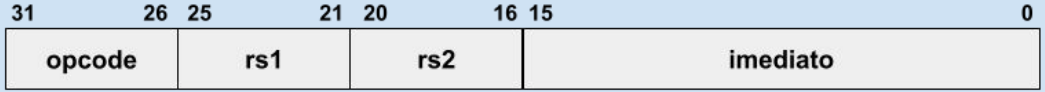
TIPO R (REGISTRADOR)



TIPO J (DESVIOS)



TIPO I (IMEDIATO)



A instrução tem um tamanho de 4 bytes de comprimento, e três formatos diferentes.

Instruções MIPS R2000

- É um processador com uma arquitetura do tipo RISC.
- Apenas instruções de load e store fazem acesso à memória. Todas as demais instruções tem operandos em registrador.
- Instruções de 32 bits com operandos de meia palavra (16 bits) ou um byte (8 bits) de comprimento.
- A ordenação dos bytes é configurável (a configuração ocorre durante o reinício do hardware) em big-endian ou little-endian.

8.2 Superpipeline

Superpipeline

- Lava-Jato sob nova direção! Agora temos:
    1. Aplicação de desengordurante na lataria e desengraxante nas rodas;
    2. Enxágue;
    3. Aplicação de xampu, enxágue;
    4. Secagem;
    5. Aspiração do interior;
    6. Limpeza dos vidros;
    7. Lavagem dos tapetes;
    8. Aplicação de silicone nas partes emborrachadas e pretinho nos pneus, troca da bolsinha de lixo.

Superpipeline

- Agora, o invés de quatro etapas de 15 minutos, temos 8 etapas, com 7,5 minutos cada.
- Conseguimos produzir um novo carro lavado a cada 7,5 minutos, dobrando assim a produtividade do nosso lava a jato!
- Notem que o tempo para a lavagem de cada carro, individualmente, continua sendo igual a uma hora.
- Se sua carreira na computação não der certo, você já pode abrir um lava-jato de sucesso! :-)

Superpipeline

- O grau de superpipelining da arquitetura é determinado pelo número de sub-estágios no qual é dividido cada estágio do pipeline original, supondo-se que essa divisão pode ser feita de forma regular.
- O relógio utilizado no superpipeline pode possuir, em princípio, frequência igual a **N** vezes a frequência do relógio do pipeline original, onde **N** é o número de vezes em cada estágio original foi dividido.

Superpipeline

- De forma geral podemos dizer que a aplicação da técnica de superpipelining com grau **G** a um pipeline convencional com relógio de período igual a **t** e número de estágios igual a **K**, resultará em um superpipeline com **K** x **G** estágios e relógio de período igual a **t/G**.
- Idealmente, o tempo total **T** gasto neste superpipeline para executar **M** instruções é determinado pela seguinte equação:

Superpipeline

$$T = (K \times G)t/G + (M - 1)t/G \tag{8.3}$$

Ou seja,

$$T = Kt + (M - 1)t/G \tag{8.4}$$

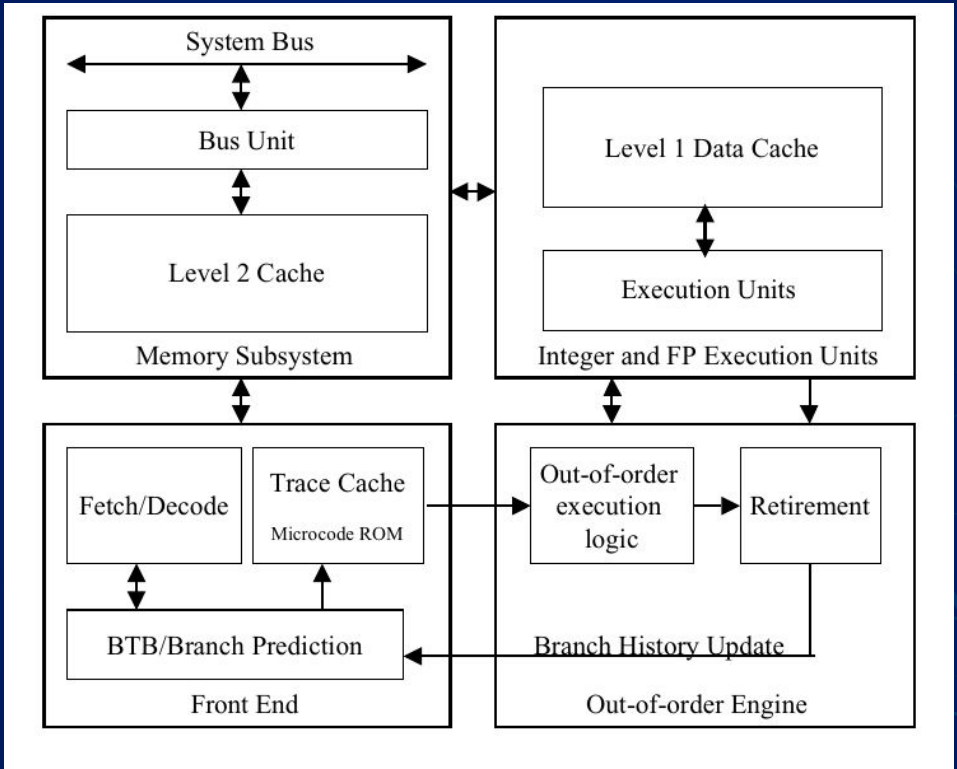
Superpipeline

- O aumento da profundidade do pipeline na técnica de superpipelining tende, no entanto, a ter seu desempenho prejudicado por situações que causam o seu “esvaziamento”.
- Tais situações podem ser decorrentes, por exemplo, de falhas no acesso à cache de instruções ou quando houver predição incorreta dos desvios condicionais.

Pentium 4

- O processador Pentium 4, na sua primeira implementação com 20 estágios, era organizado em quatro seções principais:
  - **Front-end** de execução em ordem.
  - Mecanismo de execução fora de ordem.
  - Unidades de execução inteiras e de ponto flutuante.
  - Subsistema de memória.

Pentium 4


System Bus
Bus Unit
Level 2 Cache
Memory Subsystem
Level 1 Data Cache
Execution Units
Integer and FP Execution Units
Fetch/Decode
Trace Cache
Microcode ROM
BTB/Branch Prediction
Front End
Out-of-order execution logic
Retirement
Branch History Update
Out-of-order Engine

Pentium 4

- O conjunto de instruções da arquitetura original dos processadores Intel, chamada de x86 ou IA32, é uma arquitetura do tipo CISC, que não permite uma implementação adequada de um pipeline.
- Assim, a solução adotada foi realizar a tradução dessas instruções IA-32, em tempo de execução, para o conjunto de instruções de uma arquitetura RISC interna, chamada de µops (micro-operações) pela Intel.
- Essa arquitetura RISC interna pode então executar essas instruções com uso de pipeline e outras facilidades características das arquiteturas RISC.

Pentium 4

- O front-end tem uma lógica de predição de desvio altamente precisa, que usa o histórico de execução dos desvios do programa para especular onde o programa será executado em seguida, fornecendo o endereço usado para buscar novas instrução da cache de nível 2 (L2).
- A microarquitetura NetBurst tem uma forma avançada de cache de instruções de Nível 1 (L1) chamada de Cache de Rastreamento de Execução (Execution Trace Cache).

Pentium 4

- A execução fora de ordem permite que algumas instruções sejam executadas mais cedo, caso estejam após instruções que não possam prosseguir, desde que não dependam de resultados dessas instruções atrasadas.
- A lógica de retirada termina na ordem original do programa as instruções executadas fora de ordem, podendo aposentar até três uops por ciclo de relógio.
- Essa lógica de retirada garante que as exceções ocorram somente se a operação que está causando a exceção for a operação mais antiga no pipeline.

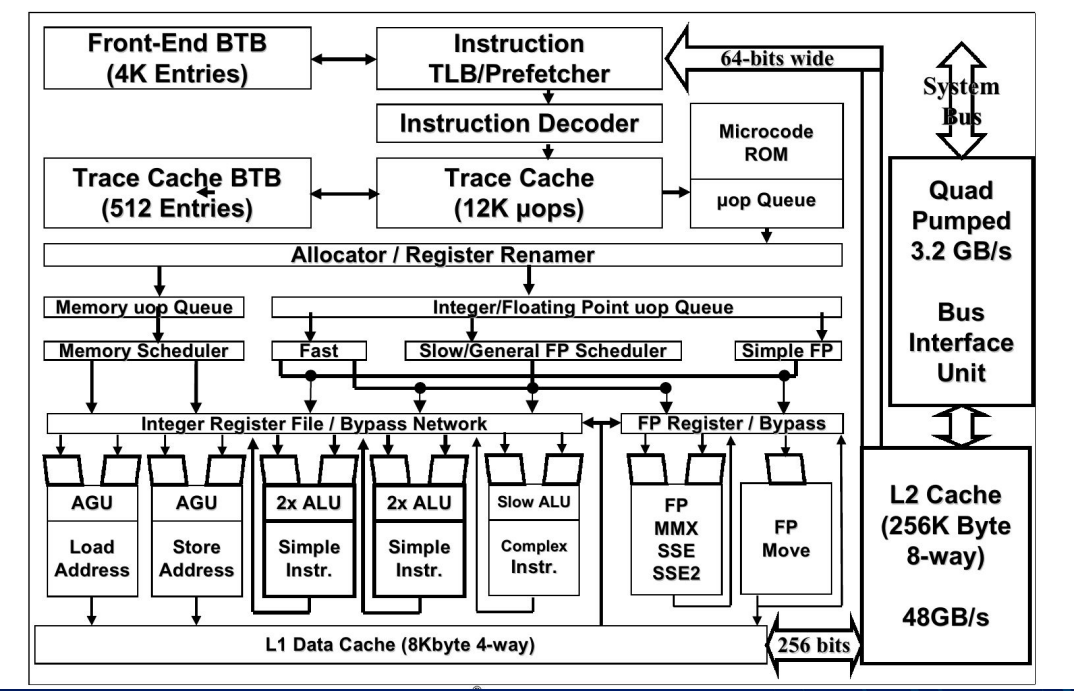
Pentium 4

- O front-end consiste de várias unidades:
  - TLB de instruções (ITLB) para a tradução de endereços virtuais em físicos pela gerência de memória.
  - Preditor de desvios (Front-End BTB) com 4 K entradas.
  - Decodificador de instruções IA-32
  - Trace Cache, com uma capacidade para armazenar até 12K uops.
  - ROM de microcódigo, para decodificar as instruções IA-32 mais complexas.

Pentium 4

- O bancos de registradores inteiro e de ponto flutuante estão situados entre os escalonadores de instrução e as unidades de execução.
- O Pentium 4 possui um total de 7 unidades funcionais:
  - Uma de cálculo dos endereços das instruções de leitura em memória e outra para as de escrita.
  - Duas unidades inteiras que operam no dobro da velocidade do relógio principal.
  - As operações inteiras que são mais complexas vão para uma unidade funcional separada.
  - As operações de ponto flutuante são executadas em duas unidades funcionais.

Pentium 4

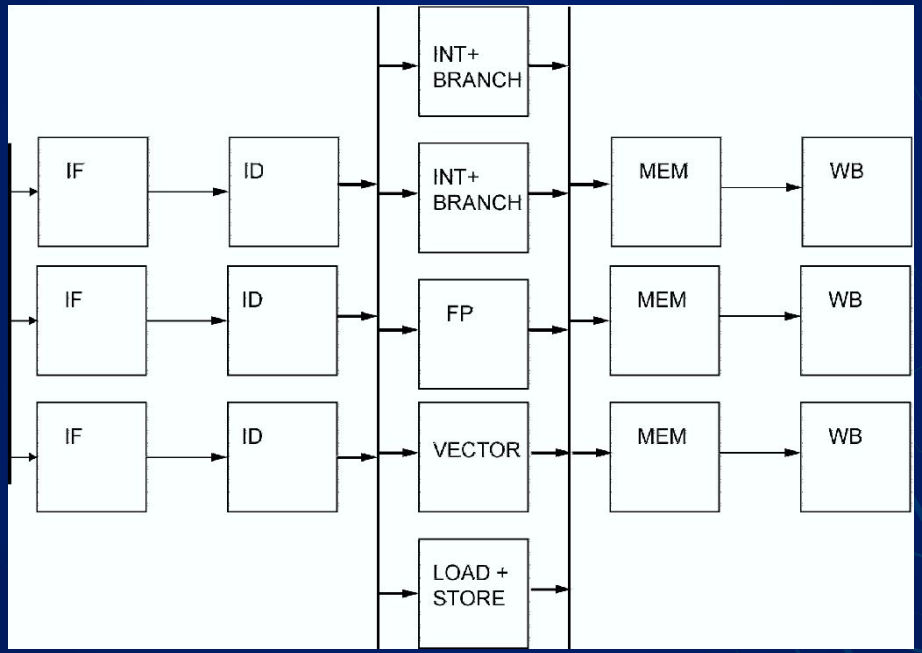

Front-End BTB
(4K Entries)
Instruction
TLB/Prefetcher
64-bits wide
System
Bus
Instruction Decoder
Microcode
ROM
Trace Cache BTB
(512 Entries)
Trace Cache
(12K µops)
µop Queue
Quad
Pumped
3.2 GB/s
Bus
Interface
Unit
Allocator / Register Renamer
Memory uop Queue
Integer/Floating Point uop Queue
Memory Scheduler
Fast
Slow/General FP Scheduler
Simple FP
Integer Register File / Bypass Network
FP Register / Bypass
AGU
Load
Address
AGU
Store
Address
2x ALU
Simple
Instr.
2x ALU
Simple
Instr.
Slow ALU
Complex
Instr.
FP
MMX
SSE
SSE2
FP
Move
L2 Cache
(256K Byte
8-way)
48GB/s
L1 Data Cache (8Kbyte 4-way)
256 bits

8.3 Arquiteturas Superescalares

Arquiteturas Superescalares

- Os processadores superescalares são capazes de executar mais de uma instrução por ciclo, sendo organizados internamente com múltiplos pipelines.
- Esses pipelines fazem uso de várias unidades funcionais, onde diversas instruções são iniciadas e terminadas a cada ciclo.
- O escalonamento das instruções é feito diretamente pelo hardware, com apoio do compilador para reordenar as instruções e otimizar a execução do programa.

Pipeline Superescalar



Condições para execução

- As instruções somente são enviadas para execução nas unidades funcionais quando atendem pelo menos duas condições:
  - Primeiro, não violam as regras de dependência de dados, ou seja, todos os seus operandos devem estar prontos e disponíveis.
  - Segundo, não há conflito estrutural, devendo existir pelo menos uma unidade funcional disponível que possa executar essa instrução.

Execução especulativa e fora de ordem

- Instruções mais recentes podem ser executadas antes de instruções mais antigas, ou seja, as instruções podem ser executadas fora da ordem especificada no código original.
- Os desvios condicionais limitam o número de instruções que podem ser executadas simultaneamente e alguma forma de predição de desvio deve ser empregada.
- Isso faz com que as instruções sejam executadas especulativamente, ou seja, antes de sabermos o resultado dos desvios dos quais elas são dependentes.

Busca das Instruções

- Para realizar a busca e decodificação de múltiplas instruções por ciclo, faz o uso de memória cache de instruções, tem a largura do barramento de busca de instruções aumentada proporcionalmente ao número de instruções a serem executadas em paralelo.
- Pode-se fazer uso da Trace Cache, que constrói sequências de instruções, ordenadas pela execução do programa, chamadas de traces.
- A Trace Cache foi uma solução adotada durante muito tempo pelos processadores da Intel.

Predição de Desvio

- O uso de esquemas eficientes de predição dinâmica de desvios, de modo a aumentar a quantidade de instruções úteis que podem ser buscadas a cada ciclo é uma constante neste tipo de processador.
- Normalmente esses esquemas utilizam o histórico recente dos últimos desvios executados, para predizer o resultado e do endereço de desvio a ser tomado.
- Um mecanismo de retirada garante que essas instruções não alterem o estado arquitetural, ou seja, registradores ou memória, antes de termos certeza do resultado dos desvios executados especulativamente.

Janela de Instruções

- A janela de instruções, onde as instruções aguardam pelos seus operandos e por unidades funcionais disponíveis, isola os estágios de busca e decodificação dos estágios de execução.
- Essa janela podem ser implementada de forma centralizada ou distribuída, ou seja, pode ser uma única fila comum a todas unidades funcionais ou existirem diversas filas, uma para cada unidade funcional do processador.

Reorder Buffer

- É uma fila onde as instruções aguardam a sua retirada na ordem especificada no código original do programa.
- Assim, as exceções e a verificação do resultado dos desvios preditos só é feita quando as instruções chegam na primeira posição da fila.
- Ou seja, quando todas as instruções anteriores a ela na ordem do programa terminaram a sua execução sem problemas.
- Se for constatada uma exceção ou predição de desvio incorreta, todas as demais instruções no reorder buffer são descartadas.

Renomeação de Registradores

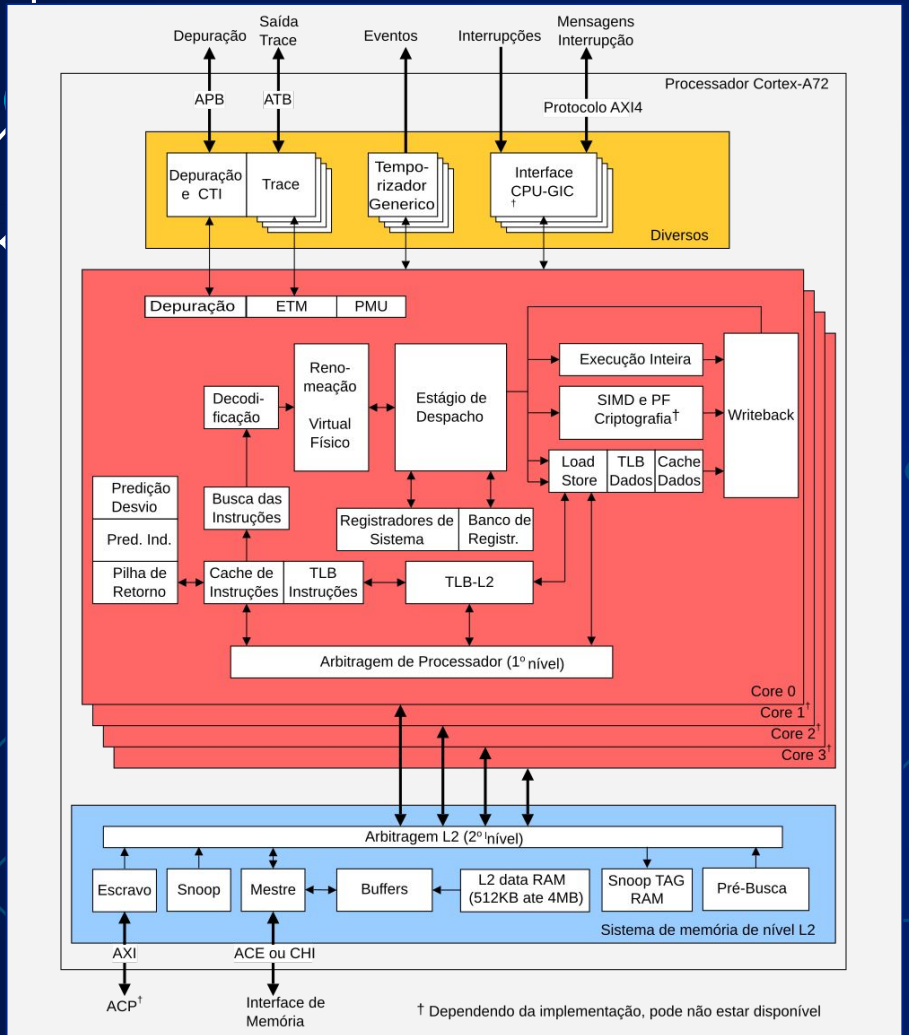
- A remoção das dependências falsas de dados, para permite o escalonamento das instruções com mais liberdade.
- Isso é feito com a renomeação de registradores, que consiste basicamente em alocar um registrador não utilizado para substituir o registrador arquitetural nas instruções.
- Essa técnica pode ser realizada por software ou por mecanismos de hardware, como a tabela de renomeação.

Fila de Instruções de load/store

- Fila de acesso à memória com suporte para tratamento de dependências de dados entre as instruções de leitura (load) e escrita (store) em memória, de modo a otimizar os acessos à memória.
- Assim, as operações de leitura podem ser adiantadas em relação às de escrita, caso não sejam para o mesmo endereço. E, caso haja alguma coincidência de endereços, os dados correspondentes podem ser adiantados da instrução de store para a instrução de load dependente.

Outras características

- Múltiplos barramentos de dados para comunicação de operandos e resultados das unidades funcionais para as instruções aguardando na janela de instruções.
- Banco de registradores deve ter múltiplas portas de leitura e escrita, para fornecer e receber os operandos necessários das diversas instruções em execução.
- Múltiplas unidades funcionais para execução das instruções, em um total que deve ser maior que o número de instruções que se deseja executar em paralelo a cada ciclo.



ARM Cortex-A72

O ARM Cortex-A72 é um processador que implementa o conjunto de instruções ARMv8-A, uma arquitetura de 64 bits, com um pipeline superescalar fora de ordem capaz de despachar até três instruções simultaneamente..

Adaptado de https://developer.arm.com/documentation/100095/0003/

ARM Cortex-A72

- O processador Cortex-A72 implementa a arquitetura ARMv8-A, incluindo suporte para o conjunto de instruções A32, para o conjunto de instruções T32 e para o conjunto de instruções A64.
- É um processador voltado para uso em computação móvel, com alto desempenho e baixo consumo, sendo utilizado, por exemplo, no nanocomputador Raspberry Pi 4, modelo B, trabalhando com uma frequência de 1,5 GHz.

Caches e TLB

- Cada núcleo possui uma cache de instruções L1 com 48 Kibytes, associatividade em 3 vias, linhas com 64 bytes e uma TLB de instruções associada com 48 entradas.
- A cache de dados de nível L1 possui capacidade de 32 Kibytes, com associatividade em 2 vias, linhas de cache de 64 bytes e uma TLB de dados associada também com 48 entradas.
- Cada núcleo possui ainda uma TLB unificada de nível 2 com 1024 entradas para acessar uma cache interna compartilhada de nível 2 com tamanho configurável entre 512 KiB e 4 MiB.

Pipeline de Instruções

- Predição de desvios, com BTB de 4096 posições e uma pilha de endereços de retorno.
- Decodificação, onde são gastos 3 ciclos de relógio para decodificar até 3 instruções ao mesmo tempo.
- Renomeação que mapeia os registradores arquiteturais para 128 registradores físicos.
- Despacho das instruções para oito janelas de instrução com 8 entradas por unidade funcional.
- Write back e de retirada, para escrita dos resultados no banco dos registradores e tratamento das exceções.

Unidades Funcionais

- Possui oito unidades funcionais, ou oito pipelines onde podem ser executadas simultaneamente:
  - Duas instruções inteiras simples.
  - Uma instrução de desvio.
  - Uma instrução inteira de múltiplos ciclos.
  - Duas instruções de ponto flutuante.
  - Uma instrução de load.
  - Uma instrução de store.
- Além disso, possui uma lógica de comparação e adiantamento de resultados para as janelas de instrução.

Unidades Funcionais

- As unidades funcionais possuem entre 1 e 6 estágios, dependendo do tipo.
- Os desvios e stores levam um ciclo para serem executados, as instruções de load tem latência de 4 ciclos, mas podem produzir um resultado novo a cada ciclo.
- A unidade funcional de inteiros de múltiplos ciclos executa instruções com 2 ou mais ciclos.
- A multiplicação de inteiros tem uma latência de 3 a 5 ciclos. A divisão de inteiros é relativamente lenta, levando de 4 a 20 ciclos.

Tipos de Instrução

- As instruções no modo A64 (arquitetura de 64 bits) e A32 (arquitetura de 32 bits) são todas com 32 bits de largura.
- No modo T32, que é utilizado quando há necessidade de um código mais compacto, variam entre 16 e 32 bits.
- As instruções são sempre acessadas no modo little-endian, já os operandos podem ser armazenados na memória no modo big-endian ou little-endian, sendo portanto considerada uma arquitetura bi-endian.

8.4 PROCESSADORES MULTICORE e MANYCORE

Processadores Multicore

- Os processadores multicore, também conhecidos como chip multiprocessor (CMP), têm com principal característica é o encapsulamento de diversos processadores superescalares, ou núcleos (cores), em uma única pastilha (chip).
- Historicamente, com o aumento da escala de integração, os processadores passaram a utilizar mais transistores e a trabalhar com maior frequência, o que por sua vez exigiu um maior consumo de energia, resultando em maior aquecimento, muito além da capacidade de refrigeração convencional (por ventilação).

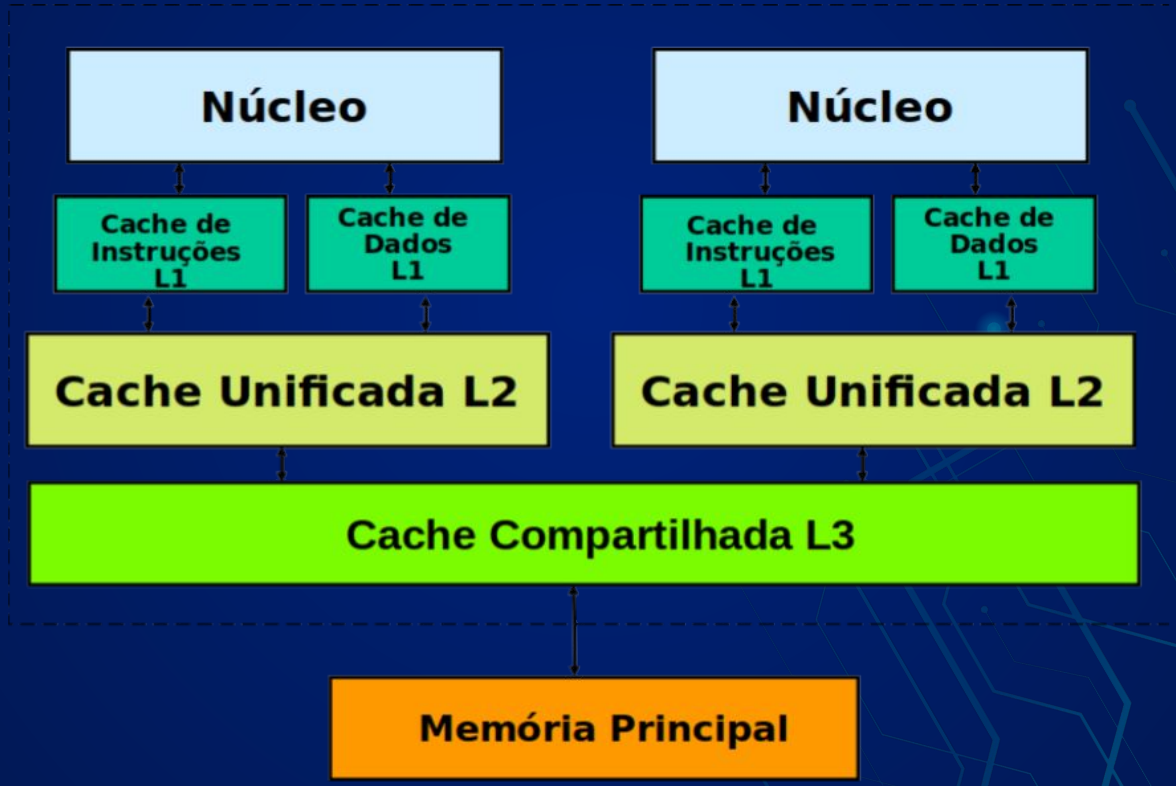
Processadores Multicore

- Com a mudança da carga de trabalho dos computadores, sejam eles servidores ou computadores pessoais, tornou-se interessante ter vários núcleos em um único encapsulamento.
- Desse modo, diversos núcleos podem trabalhar com frequências de relógio menores, ajustadas à carga de trabalho, consumindo menor potência, mas com mais capacidade de processamento do que só um grande processador de maior frequência.

Processadores Multicore

- Para as cargas de trabalho orientadas para a vazão, pode-se conseguir mais desempenho/potência e desempenho/área do chip, levando-se ao extremo o conceito que a latência, mas sim o throughput, é o mais importante.
- O uso de múltiplos núcleos facilitou a tarefa de programação com múltiplas threads, já que agora, a comunicação podia ser feita na memória cache compartilhada de nível 3 (L3), onde cada evento de comunicação toma poucos ciclos do processador, facilitando a divisão das tarefas em diversas threads.

Processador Multicore



Processadores Multicore

- Os processadores multicores também não necessitam de um esforço muito grande de engenharia para cada geração dos processadores, já que não é necessário redesenhar todo o projeto do núcleo dos processadores.
- O projeto das placas do sistema necessita de uma menor mudança em relação às gerações dos multicores, sendo que a única real diferença é que as placas necessitam tratar da maior largura de banda de E/S que é exigida pelos processadores com um número maior de núcleos.

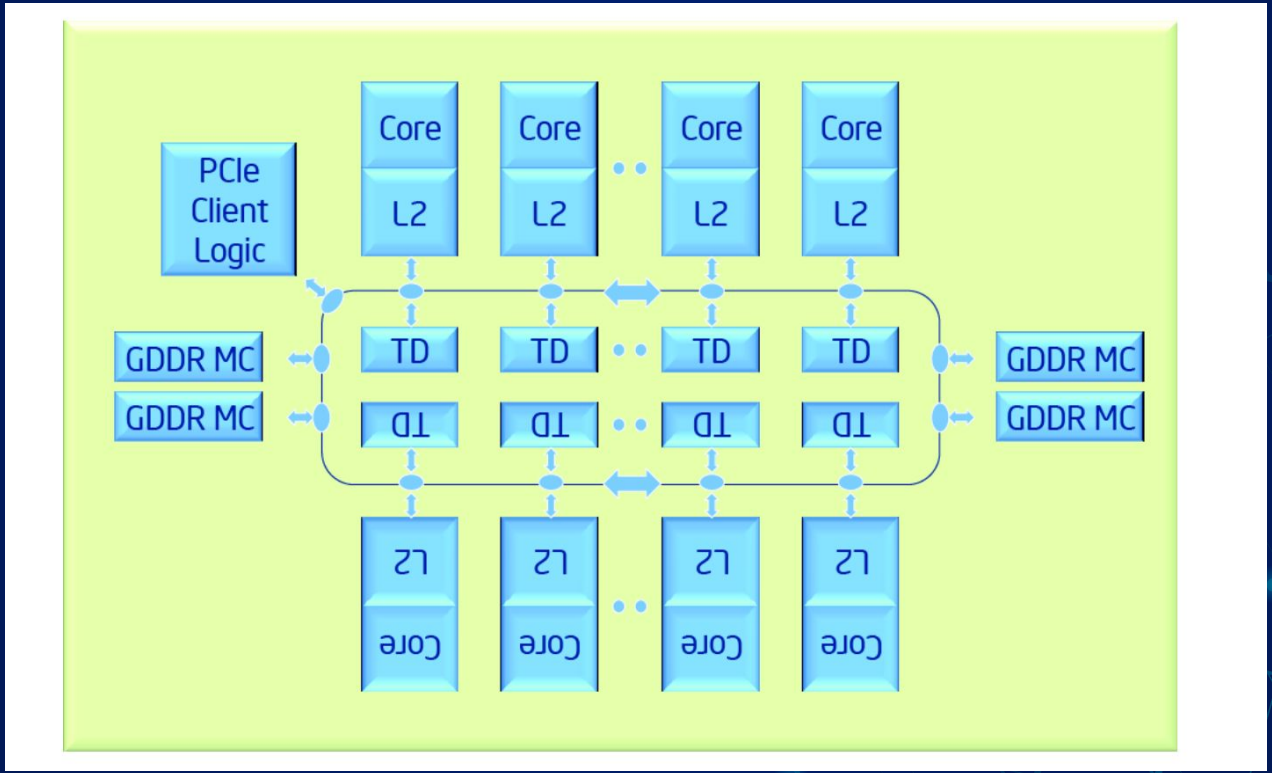
Intel Xeon Phi

- O processador Intel Xeon Phi, foi lançado em 2010 com até 61 núcleos conectados por uma interconexão bidirecional interna ao chip de alto desempenho.
- Inicialmente, cada um dos núcleos de processamento (unidade escalar) era uma arquitetura de execução em ordem, baseada na família de processadores Intel Pentium, com um conjunto de novas instruções vetoriais, que utilizavam uma unidade vetorial de ponto flutuante (VPU) com largura de 512 bits.

Intel Xeon Phi

- Os núcleos eram conectados por uma rede de interconexão em anel por meio da Interface de Anel Central (CRI), que em cada núcleo hospedava a cache L2 e o diretório de tags (TD), conectando cada núcleo a um coprocessador Intel Xeon Phi Ring Stop (RS).
- O chip do Xeon Phi possuía ainda uma lógica de interface do sistema para um processador hospedeiro ou para um barramento PCI Express, além de um mecanismo de DMA, e controlador de memória.

Intel Xeon Phi



Intel Xeon Phi

- Cada núcleo podia executar 2 instruções por ciclo, em dois pipelines diferentes, mas nem todos os tipos de instrução podiam ser executadas nos dois pipelines, como por exemplo, as instruções vetoriais que só podiam ser executadas em um deles.
- Cada núcleo tinha uma cache L1 separada para dados e instruções, com 32 KiB cada uma. A cache L2 era unificada com 512 KiB, contribuindo para o armazenamento total em cache L2, compartilhado globalmente. A latência da cache L1 era de um ciclo de relógio e da cache L2 de 11 ciclos de relógio.

Intel Xeon Phi

- As caches L2 eram mantidas totalmente coerentes pelo hardware, usando DTDs (diretórios de tags distribuídos), que são referenciados após uma falha na cache L2.
- O diretório de tags não é centralizado, mas dividido em 64 DTDs, cada um recebendo uma porção igual do espaço de endereço e sendo responsável por mantê-lo globalmente coerente.
- A capacidade total de armazenamento era maior ou menor em função da quantidade de dados compartilhados entre os diversos núcleos.

8.5 Arquiteturas VLIW

Arquiteturas VLIW

- Os processadores com arquitetura do tipo VLIW (Very Long Instruction Word) também são capazes de executar mais de uma instrução por ciclo de relógio.
- Ao contrário dos processadores superescalares, buscam transferir a responsabilidade da identificação das instruções que podem ser executadas em paralelo do hardware para o software.
- Todas as operações que compõem a palavra longa podem, a princípio, ser executadas em paralelo e a posição da operação na palavra longa define em qual unidade funcional ela será executada.

Arquiteturas VLIW

- As arquiteturas VLIW são organizadas com múltiplas unidades funcionais, que operam em paralelo e realizam acessos concorrentes ao banco de registradores.
- Para suportar essa concorrência de acessos, o banco de registradores deve ter múltiplas portas de leitura e de escrita.
- O escalonamento de instruções realizado pelo compilador tem como vantagens principais o aumento do paralelismo e a simplificação do hardware, quando comparado com processadores superescalares.

Arquiteturas VLIW

- Uma das limitações das arquiteturas VLIW é a impossibilidade de se conhecer, em tempo de compilação, exatamente quais instruções serão executadas em um trecho de código com desvios condicionais.
- Um segundo fator negativo é a incapacidade do compilador de, em muitos casos, identificar se duas operações distintas de acesso à memória se referem ou não a uma mesma posição de memória, uma vez que os endereços de memória só são conhecidos em tempo de execução.

Processador Itanium

- Um dos exemplos mais famosos de implementação de arquitetura VLIW é o processador Itanium.
- Também conhecida como EPIC (Explicit Parallel Instruction Computing), essa arquitetura foi desenvolvida em conjunto pela Intel e pela Hewlett-Packard, e procurava remover algumas das limitações tradicionais das arquiteturas VLIW com a incorporação de algumas técnicas bastante agressivas na arquitetura para atacar os problemas decorrentes de desvios condicionais e de acesso à memória.

Processador Itanium

- Alguns exemplos que podemos citar são:
  - Uso de instruções predicadas (ou condicionais) para lidar com os desvios condicionais.
  - Predição, em tempo de compilação, do nível de memória cache, dentro do sistema de hierarquia de memória, onde se encontram os dados a serem acessados por instruções de load e store.
  - Especulação de dados, permitindo que uma instrução de load seja realizada antes de uma instrução de store que a precede.

Processador Itanium

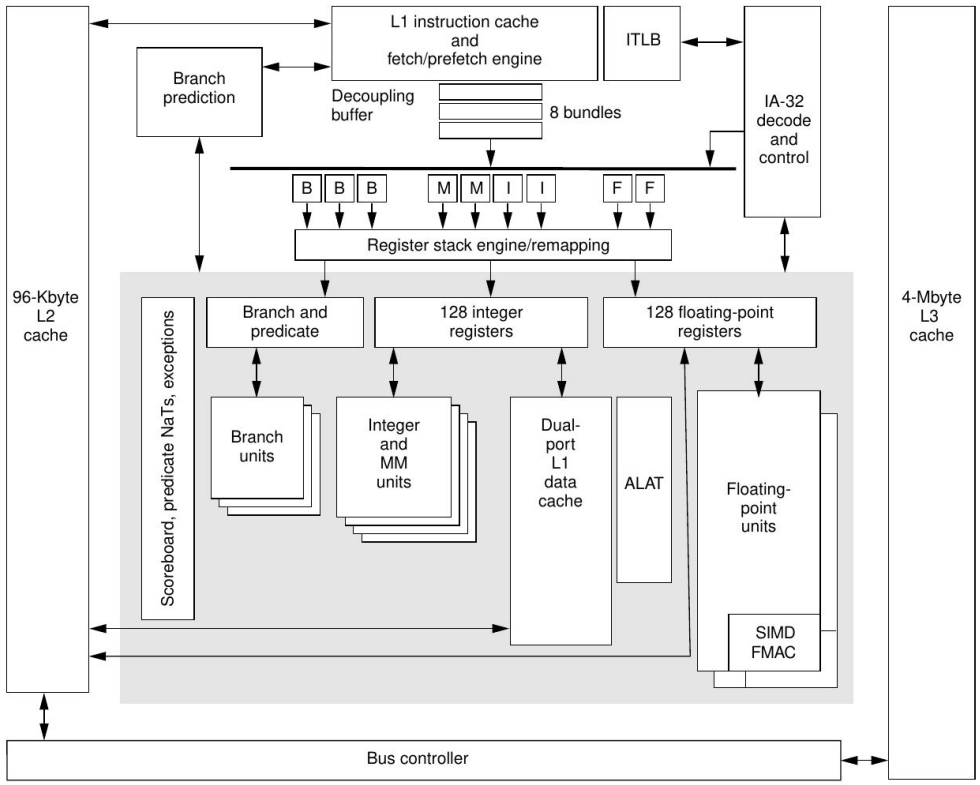


Figure 4. Itanium processor block diagram.

Processador Itanium

- O processador Itanium uma arquitetura VLIW de 64 bits, capaz de executar até 6 operações/ciclo, possuindo 4 unidades inteiras, 4 de multimídia, 3 unidades de desvio, 2 de load/store, 2 de ponto flutuante com precisão estendida e 2 de ponto flutuante com precisão simples.
- Possuía uma cache de instruções de nível 1 (L1) de 16 KiB, capaz de fornecer dois pacotes de instruções a cada ciclo de relógio. Além disso, tinha um cache L1 de 16 KiB e cache L2 unificada de 96 KiB e uma cache de nível (L3) com 4 Mibytes.

Processador Itanium

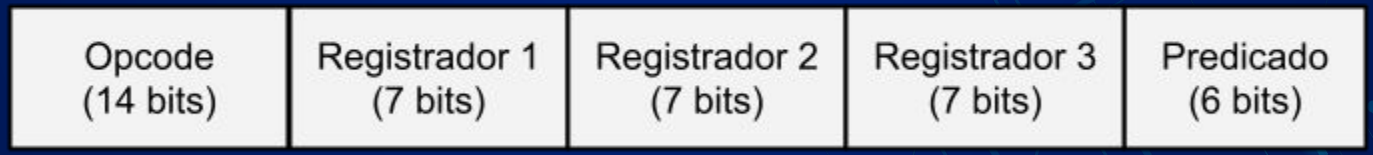
- Cada palavra VLIW consistia de um ou mais pacotes de 128 bits, sendo que cada pacote tinha até 3 operações e um template.
- Duas instruções VLIW eram buscadas a cada ciclo, totalizando até 6 operações buscadas, despachadas e executadas por ciclo de relógio do processador.
- A arquitetura IA-64 não insere operações de NOP para preencher posições ou campos vazios no pacote.

Processador Itanium

- O template indica explicitamente o paralelismo, com a seguir.
  - Se as instruções no pacote podem ser executadas em paralelo.
  - Se uma ou mais delas devem ser executadas serialmente.
  - Se o pacote pode ser executado em paralelo com os pacotes vizinhos.

Formato das Instruções Itanium

(a) Agrupamento de 3 instruções com 128 bits



(b) Formato da instrução

Processador Itanium

- Na sua arquitetura estão previstos 128 registradores de uso geral de 64 bits e outro conjunto de 128 registradores de ponto flutuante de 82 bits, além de 64 registradores de predicado.
- O banco de registradores inteiros possui oito portas de leitura e seis portas de escrita para atender à alta demanda por operandos.
- Os registradores de uso geral de 0 a 31 e os registradores de predicado são fixos.
- Os registradores de uso geral 32 a 127 e os registradores de predicado de 16 a 63 podem ser renomeados.

Processador Itanium

- O compilador pode escalonar os laços de código segundo a técnica de software pipeline.
- O IA-64 suporta especulação de dados e controle controlada pelo software, assim como a predição estática e dinâmica para os desvios.
- Para manter compatibilidade com a família x86, uma unidade de controle e decodificação especial para instruções do IA-32 estava presente no Itanium.

Processador Itanium

- Na tentativa de aumentar o desempenho, o processador IA-64 incluiu diversas facilidades, tornando-se assim o processador VLIW mais complexo já projetado.
- Isso é uma contradição, já que a arquitetura VLIW tem como objetivo simplificar o hardware transferindo complexidade para o compilador.
- Em 2019, a Intel anunciou que a produção da família de processadores Itanium terminaria em janeiro de 2020 e os encomendas poderiam ser feitas até julho de 2021.

8.6 Arquiteturas Multithreading

Arquiteturas Multithreading

- As arquiteturas multithreading procuram esconder ou reduzir o efeito negativo da operações de longa latência realizadas pelo processador, diminuindo o tempo gasto na troca de contexto entre as threads.
- Vários contextos são replicados hardware, caracterizados tipicamente por uma cópia, para cada thread, do banco de registradores, apontador de instruções, apontador de pilha e palavra de status do processador, se for o caso.

Arquiteturas Multithreading

- As operações de longa latência podem ser de diversas origens: uma falha na memória cache; o acesso a dados e instruções em uma memória remota; a espera pelas operações de sincronização no acesso aos dados compartilhados, entre outras.
- Os modelos de arquitetura baseados nesta técnica podem ser dos seguintes tipos:
  - multithreading de granularidade fina.
  - multithreading de granularidade grossa.
  - multithreading simultâneo.

Multithreading de Granularidade Fina

- O número de contextos suportados em hardware deve ser, no mínimo, igual ao número de estágios do pipeline.
- A principal desvantagem desta abordagem é que o desempenho de código sequencial (código com uma única thread) pode ser bastante ruim, piorando conforme aumenta o número de estágios do pipeline.
- Neste modelo o processador realiza uma troca de contexto a cada ciclo de relógio, de modo que apenas uma instrução de cada thread esteja presente no pipeline em um determinado instante de tempo.

Multithreading de Granularidade Fina

- A lógica de controle do pipeline é bastante simplificada, pois não existem dependências de dados e de controle entre as instruções no pipeline. Além disso, a sobrecarga para troca de contexto é nula, já que o processador sempre sabe antecipadamente de qual thread será a próxima instrução a ser executada.
- A arquitetura MTA da Tera Computer suportava 128 contextos, cada um com 32 registradores de uso geral, 8 registradores de endereço, uma palavra de status, todos com 64 bits, e utilizava um pipeline com 21 estágios.

Multithreading de Granularidade Grossa

- As arquiteturas dessa classe se dividem em estáticas e dinâmicas.
  - Estáticas: a troca de contexto pode ser implícita, que é feita quando certas instruções (load, store, branch}, etc.) são encontradas, ou explícita, que faz a troca de contexto quando instruções explícitas de troca de contexto são executadas.
  - Dinâmicas: a troca de contexto acontece quando ocorre uma operação de longa latência (falha na cache, sincronização, etc.).

Multithreading de Granularidade Grossa

- Tipicamente um número não muito grande de contextos (4 a 32) é suportado por esse tipo de arquitetura, como exemplos temos o SPARCLE (Máquina Alewife do MIT) e NCE SPARC+ (Multiplus do NCE/UFRJ), ambos baseados na arquitetura SPARC.
- Essa técnica só é efetiva se a troca de contexto for efetuada em um número menor de ciclos do que o gasto nas operações cuja latência deva ser ocultada.
- A eficiência é determinada por fatores como a frequência operações de alta latência ocorrem.

Multithreading Simultâneo

- O multithreading simultâneo (SMT) é uma técnica que permite múltiplas threads despacharem múltiplas instruções a cada ciclo para unidades funcionais de um processador superescalar.
- O SMT combina a capacidade de despacho de múltiplas instruções das arquiteturas superescalares, com a habilidade de esconder latência das arquiteturas multithreading de granularidade fina ou grossa.
- A cada instante de tempo instruções de diferentes threads podem estar sendo executadas simultaneamente no pipeline.

Modificações Arquiteturais

- Uso de múltiplos apontadores de instrução;
- Um grande banco de registradores, com registradores para as threads e registradores adicionais para renomeação;
- Dois estágios no pipeline para acesso aos registradores;
- Várias pilhas para predição do endereço de retorno das rotinas, uma para cada thread;
- Tabelas de renomeação individualizadas para cada uma das threads;
- A identificação de cada thread nas TLBs, nos mecanismos de predição de desvio e janelas de instrução.

Multithreading Simultâneo

- O uso de SMT resulta em uma maior utilização da memória, na diminuição na taxa de acerto da cache de instruções e do mecanismo de predição de desvios, devido a interferência entre as várias threads/processos em execução.
- Mesmo assim, tem sido uma opção arquitetural adotada em diversos processadores comerciais, como nas últimas versões do Intel Pentium 4 e em toda a linha Intel Core, além dos processadores da IBM da linha Power, a partir do Power5, com SMT de duas threads, até Power10, com SMT de até 8 threads.

Multithreading Simultâneo

- A Intel utiliza a arquitetura SMT em seus processadores com o nome de hyperthreading, onde cada processador lógico mantém uma cópia completa do estado arquitetural, sendo que do ponto de vista do software um processador físico aparece para o programador como se fossem dois processadores lógicos, podendo até passar despercebido para os mais desatentos o fato de que não são processadores reais.

Arquitetura Power5

- O processador IBM Power5 suporta a arquitetura de 64 bits PowerPC, tem dois núcleos, sendo que cada um deles é um processador capaz de executar duas threads usando SMT.
- Essa arquitetura faz com que o chip apareça como um multiprocessador simétrico de quatro núcleos para o sistema operacional.
- Poucos elementos precisaram ser replicados, tais como os apontadores de instrução e o banco de registradores. O processador faz a leitura de até oito instruções por vez, alternando entre as duas threads.

Arquitetura Power5

- A cache de instruções L1 tem 64 Kibytes e a de dados possui 32 Kibytes,.
- Os dois núcleos compartilham uma cache L2 de 1,875 Mibyte.
- Qualquer núcleo do processador pode acessar independentemente cada controlador da cache de nível L2, que está dividida em três partes.
- O diretório para uma cache L3 de 36 Mibytes está também integrado no chip do Power5, mas a cache propriamente dita está fora do chip.

Arquitetura Power 5

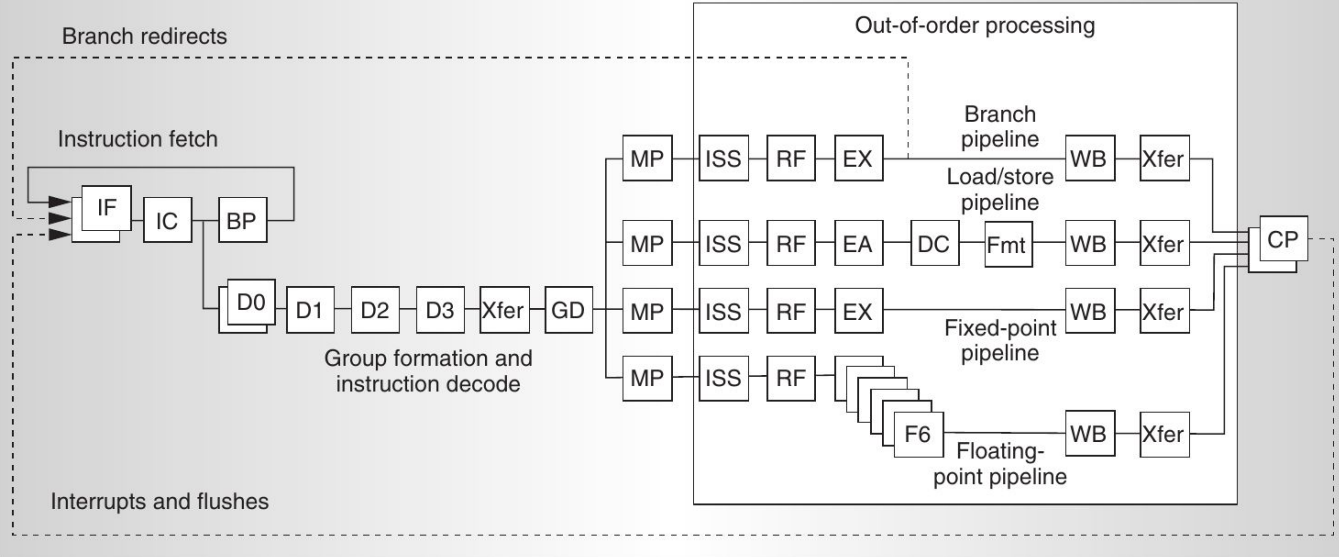


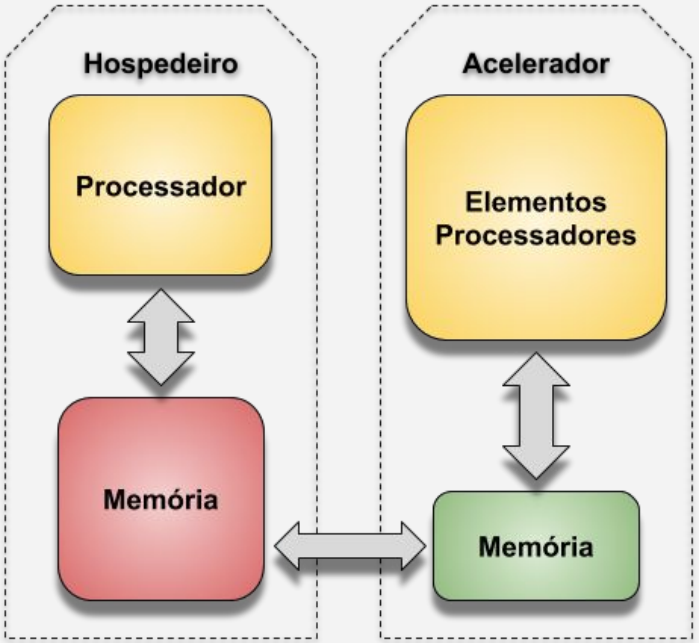
Figure 3. Power5 instruction pipeline (IF = instruction fetch, IC = instruction cache, BP = branch predict, D0 = decode stage 0, Xfer = transfer, GD = group dispatch, MP = mapping, ISS = instruction issue, RF = register file read, EX = execute, EA = compute address, DC = data caches, F6 = six-cycle floating-point execution pipe, Fmt = data format, WB = write back, and CP = group commit).

"IBM Power5 Chip: A Dual-Core Multithreaded Processor". IEEE Micro.

8.7 Aceleradores

Aceleradores

- Aceleradores são dispositivos especiais de hardware que trabalham em conjunto com processadores convencionais, executando trechos intensivos de código, com alto potencial de paralelismo, chamados de kernels.
- Entre os dispositivos desse tipo podemos destacar: Graphics Processing Units (GPUs) e Field Programmable Gate Arrays (FPGAs).



Acelerador

Na arquitetura Harvard as instruções e dados do programa são armazenados em memórias distintas.

FPGAs

- As FPGAs são dispositivos de hardware cuja funcionalidade é programável por software, que podem ser personalizados para executar com eficiência determinados trechos de código com uso intensivo de dados e/ou alta demanda computacional.
- Assim, esses dispositivos podem ser programados e reprogramados de acordo com o tipo de aplicação executada, se adaptando à solução de diversos tipos de problemas.

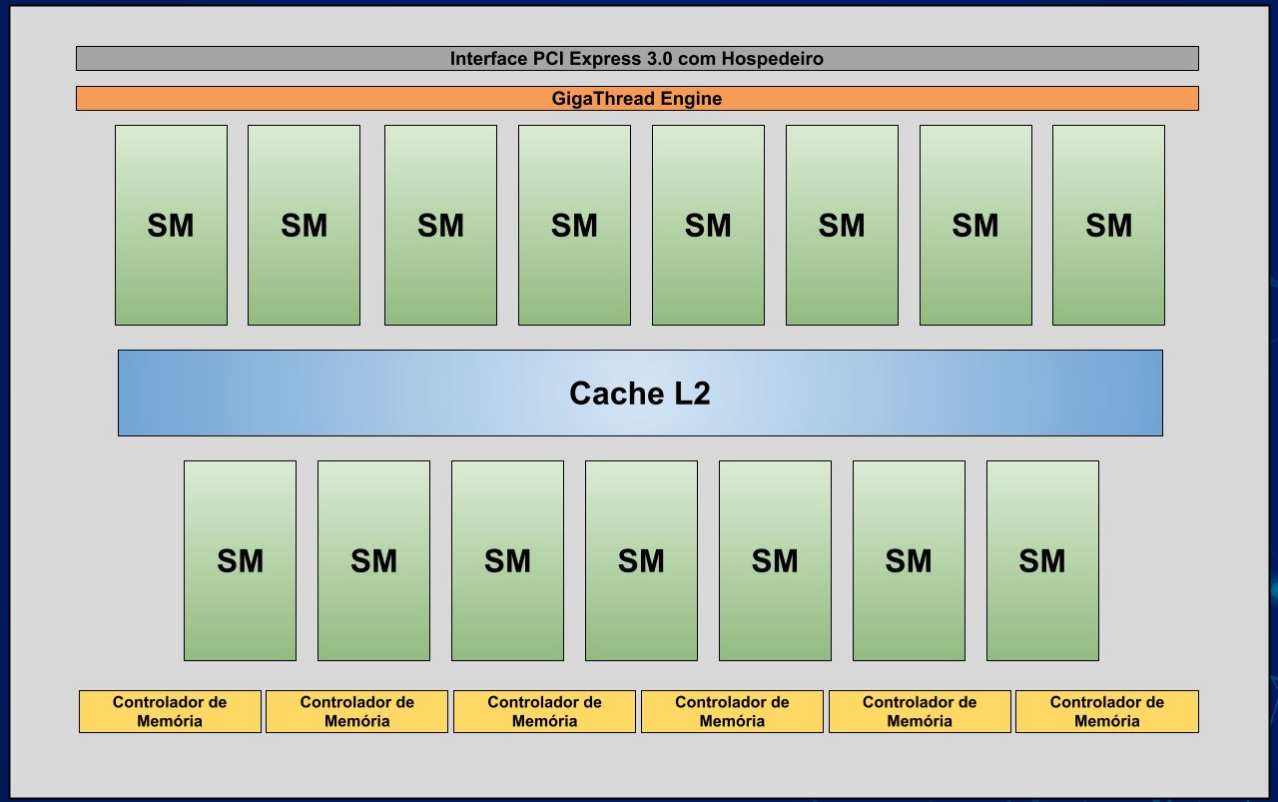
GPUs

- A GPU é um tipo de acelerador com um grande número de núcleos de processamento paralelo maciço com foco na eficiência energética e para aplicações com demandas que melhorem o throughput.
- Inicialmente desenvolvidos com o objetivo de atender a área de jogos, rapidamente mostrou-se muito eficiente na execução de vários tipos de aplicação científica.

GPU Kepler

- Na arquitetura da GPU Kepler, cada unidade de multiprocessador de fluxo (SM) possui 192 núcleos de precisão simples e 64 de precisão dupla.
- As 32 unidades de função especial (SFU) dentro de cada SM são utilizadas para aproximar operações transcendentais como raiz quadrada, seno, cosseno e recíproco (1/x).
- O escalonador do SM dispara grupos de 32 threads chamados de warps. Cada SM permite um máximo de quatro warps disparados e executados concorrentemente.

GPU Kepler


Interface PCI Express 3.0 com Hospedeiro
GigaThread Engine
SM
SM
SM
SM
SM
SM
SM
SM
Cache L2
SM
SM
SM
SM
SM
SM
SM
Controlador de Memória
Controlador de Memória
Controlador de Memória
Controlador de Memória
Controlador de Memória
Controlador de Memória

GPU Kepler

- A memória local (64 ou 128 KiB) de cada SM pode ser dividida em várias proporções entre uma memória compartilhada e uma cache L1.
- Além da cache L1, a arquitetura Kepler introduz uma cache apenas de leitura de 48 KiB (Textura).
- Essa arquitetura possui também uma cache de nível 2 (L2) com 1,5 MiB de capacidade.
- A memória externa é uma DDR5 configurável de acordo com o modelo da placa de vídeo.

Hierarquia de Memória GPU

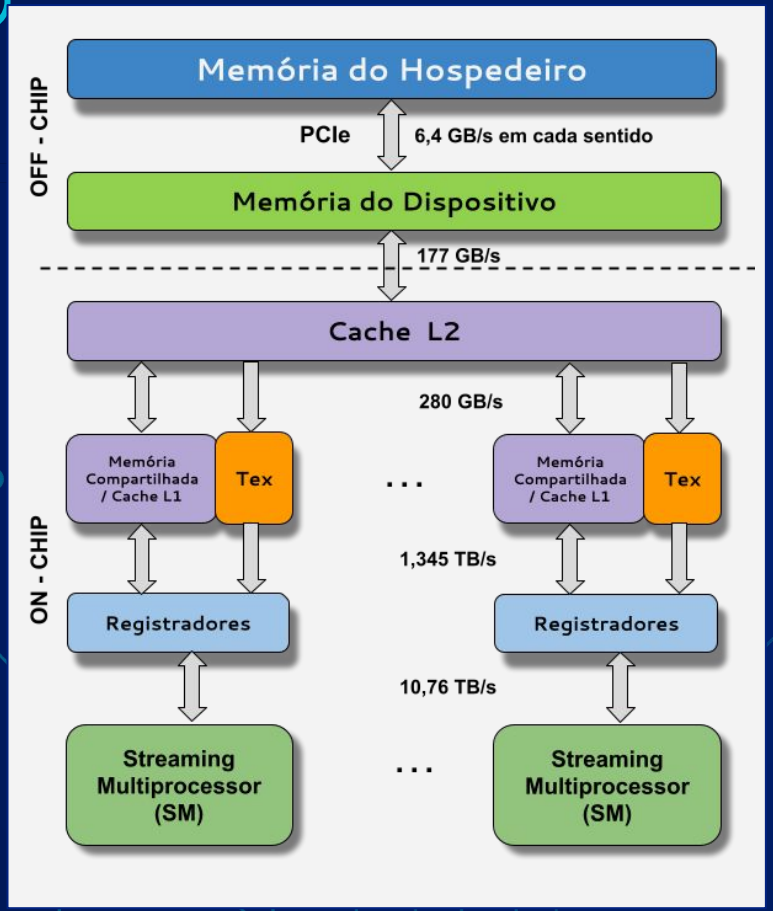


Tabela Evolução GPU

Evolução				
Características da GPU	NVIDIA Tesla P100	NVIDIA Tesla V100	NVIDIA Ampere A100	NVIDIA Kepler
Versão	GP100	GV100	GA100	GK110
Arquitetura GPU	Pascal	Volta	Ampere	Kepler
Capac. Computacional (CC)	6.0	7.0	8.0	3.5
Threads / Warp	32	32	32	32
Máx. Warps / SM	64	64	64	64
Máx. Threads / SM	2048	2048	2048	2048
Máx. Blocos de Thread / SM	32	32	32	16
Máx. Registradores 32-bit / SM	65536	65536	65536	65536
Máx. Registradores / Block	65536	65536	65536	65536
Máx. Registradores / Thread	255	255	255	255
Máx. Tam. Bloco de Threads	1024	1024	1024	1024
Núcleos FP32 / SM	64	64	64	192
Registradores SM / Núcleos FP32	1024	1024	1024	341
Tam. da Memória Compart. / SM	64 KiB	até 96 KiB	até 48 KiB	até 48 KiB

8.8 Arquiteturas Paralelas

Classificação de Flynn

- SISD: Single Instruction Stream Single Data Stream -- (p.ex. processadores convencionais, pipelined ou superescalares)
- SIMD: Single Instruction Stream Multiple Data Streams -- (p.ex. processadores vetoriais, unidades funcionais vetoriais, arquiteturas SIMD)
- MIMD: Multiple Instruction Streams Multiple Data Streams -- (p.ex. Multiprocessadores, multicomputadores)

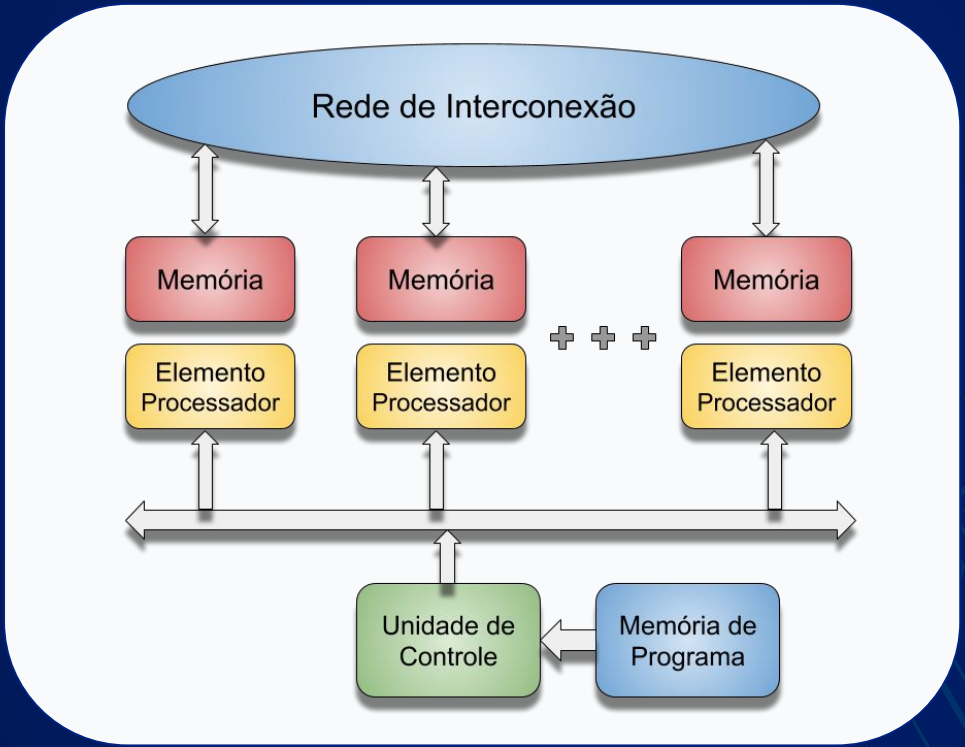
Arquiteturas SISD

- São os processadores convencionais, onde apenas um processo ou thread é executado por vez, podendo fazer uso de técnicas de exploração de paralelismo temporal (pipeline, superpipelined) ou espacial (superescalares) no nível de instrução.
- Nesses casos, o paralelismo é explorado de forma transparente ao usuário, no nível de instrução, com um suporte mínimo do compilador, por exemplo, no escalonamento mais adequado das instruções em linguagem de máquina para a execução em várias unidades funcionais.

Arquiteturas SIMD

- As arquiteturas SIMD são aquelas em que uma única instrução opera sobre um conjunto de dados distintos.
- Os seus principais tipos são:
  - Processadores vetoriais
  - Arquiteturas SIMD convencionais
  - Arquiteturas sistólicas.
- São utilizadas em aplicações como processamento de imagens; computação científica; compressão; criptografia; e aprendizado de máquina, entre outras.

SIMD


Rede de Interconexão
Memória
Memória
Memória
Elemento Processador
Elemento Processador
Elemento Processador
Unidade de Controle
Memória de Programa

Arquiteturas SIMD Convencionais

- É uma classe importante de processadores, devido a fatores como:
  - Simplicidade de conceitos e programação;
  - Regularidade da estrutura;
  - Facilidade de escalabilidade em tamanho e desempenho;
  - Aplicação direta em uma série de aplicações paralelas para obter o desempenho necessário.
- Os processadores executam sincronizadamente a mesma instrução sobre dados diferentes, fazendo uso de vários processadores especiais muito mais simples, geralmente organizados de forma matricial.

Processadores Vetoriais

- Memória-Memória:
  - Arquiteturas mais antigas, com instruções vetoriais que manipulam operandos diretamente na memória. Tem maiores limitações de desempenho e flexibilidade.
- Registrador-Registrador:
  - Arquiteturas de segunda geração. As instruções vetoriais, exceto load/store, trabalham com registradores. As unidades funcionais organizadas em pipelines profundos. Tem maior desempenho e flexibilidade.
- Instruções Vetoriais em Processadores SISD:
  - Processadores comerciais modernos incorporam instruções vetoriais. Realizam a mesma operação em um conjunto de elementos. Destaque para o conjunto AVX-512 da Intel. Tem maior flexibilidade e compatibilidade com software existente.

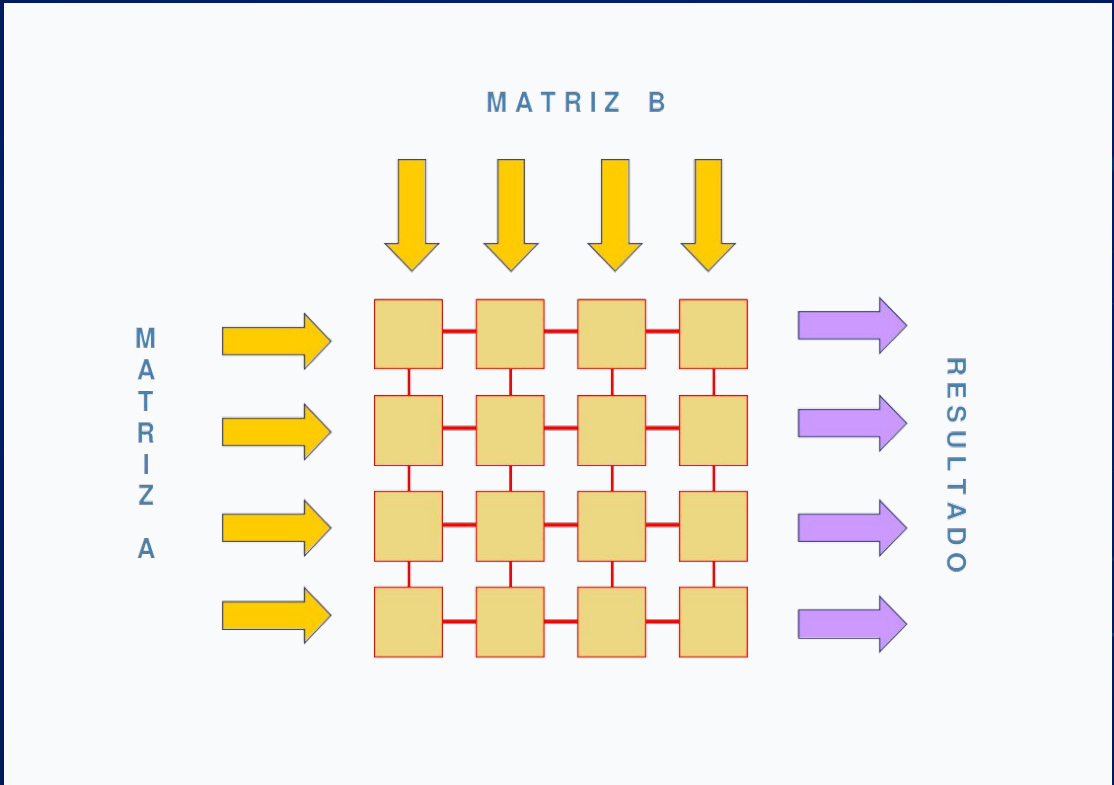
Processadores Vetoriais

- As instruções vetoriais possuem as seguintes características:
  - Cada instrução equivale a um laço.
  - O cálculo de cada resultado não depende de resultados anteriores, assim é possível haver pipelines profundos sem a ocorrência de dependências de dados.
  - O padrão de acesso à memória para a busca dos operandos é conhecido e regular, se beneficiando da utilização de memória com entrelaçamento (interleaving) ou mesmo memórias caches.

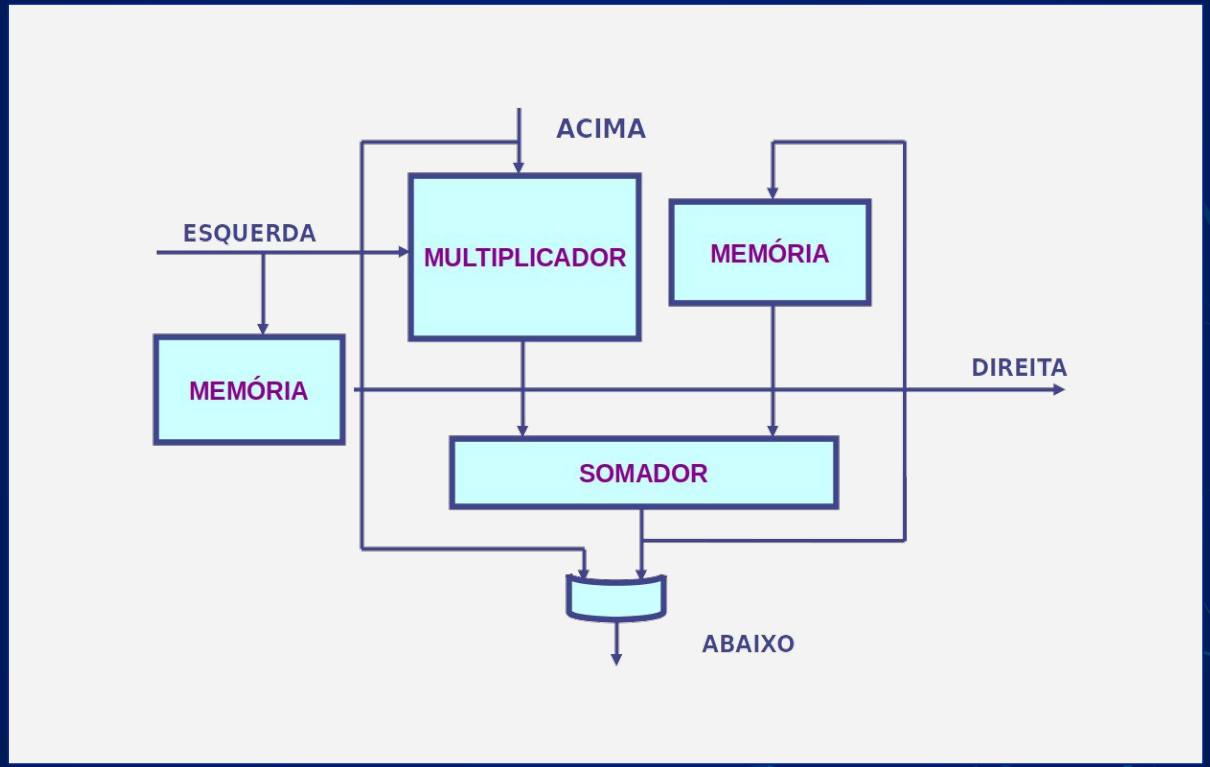
Arquiteturas Sistólicas

- Um arquitetura sistólica consiste em um conjunto de células interconectadas, cada uma delas capaz de realizar uma operação simples.
- As células em uma arquitetura sistólica são normalmente interconectadas em um arranjo ou árvore bi-dimensional.
- Atualmente as arquiteturas sistólicas são utilizadas nas Tensor Processing Unit no aprendizado de máquina para a emulação de redes neuronais em aplicações como inteligência artificial, tradução, reconhecimento de voz e imagens, entre outras.

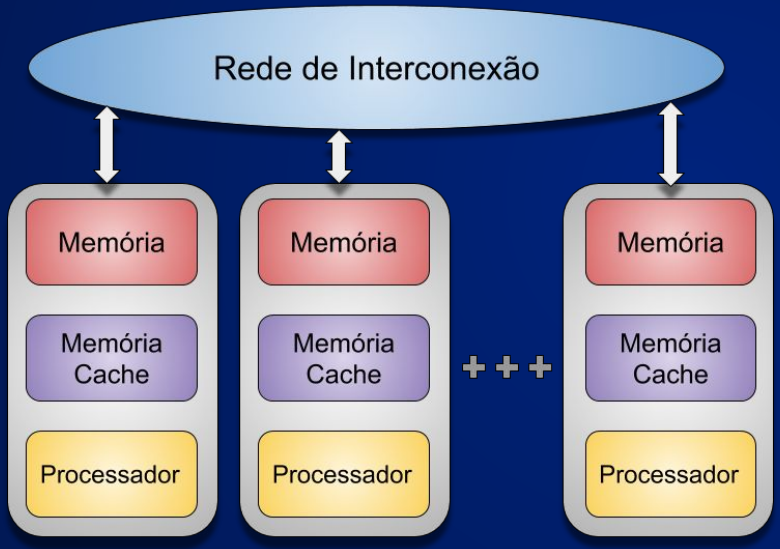
Arquiteturas Sistólicas



Arquiteturas Sistólicas



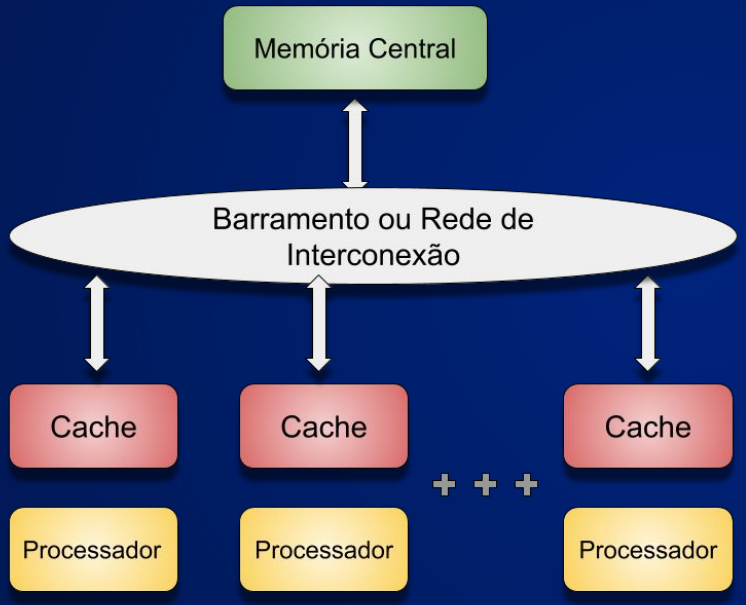
Arquiteturas MIMD Memória Distribuída



Cada processador possui um espaço de endereçamento próprio que não é compartilhado com os demais processadores.

A comunicação é feita pela troca de mensagens transmitidas através de uma rede de interconexão.

Arquiteturas MIMD Memória Compartilhada



As arquiteturas de memória compartilhada têm como característica principal o compartilhamento de um único espaço de endereçamento, permitindo a comunicação através de escritas e leituras em variáveis na memória compartilhada.

Arquiteturas MIMD Memória Compartilhada

- Vantagens:
  - Não necessitam fazer o particionamento de código ou dados, logo técnicas de programação sequenciais podem ser facilmente adaptadas.
  - Não há também necessidade da movimentação física dos dados, quando dois ou mais processadores se comunicam, resultando em uma comunicação entre processos ou threads bastante eficiente.

Arquiteturas MIMD Memória Compartilhada

- As suas desvantagens são:
  - Necessidade do uso de primitivas especiais de sincronização quando do acesso a regiões compartilhadas de memória.
  - Falta de escalabilidade devido ao problema de contenção de memória.
  - Depois de um determinado número de processadores a adição de mais processadores não aumenta o desempenho.

Arquiteturas MIMD Memória Compartilhada

- Principais tipos:
  - Arquiteturas UMA
    - São arquiteturas com memória única global.
    - O tempo de acesso à memória é uniforme para todos os nós de processamento
  - Arquiteturas NUMA
    - A memória está dividida entre local e remota.
    - O acesso à memória local é muito mais rápido do que o acesso à memória remota.

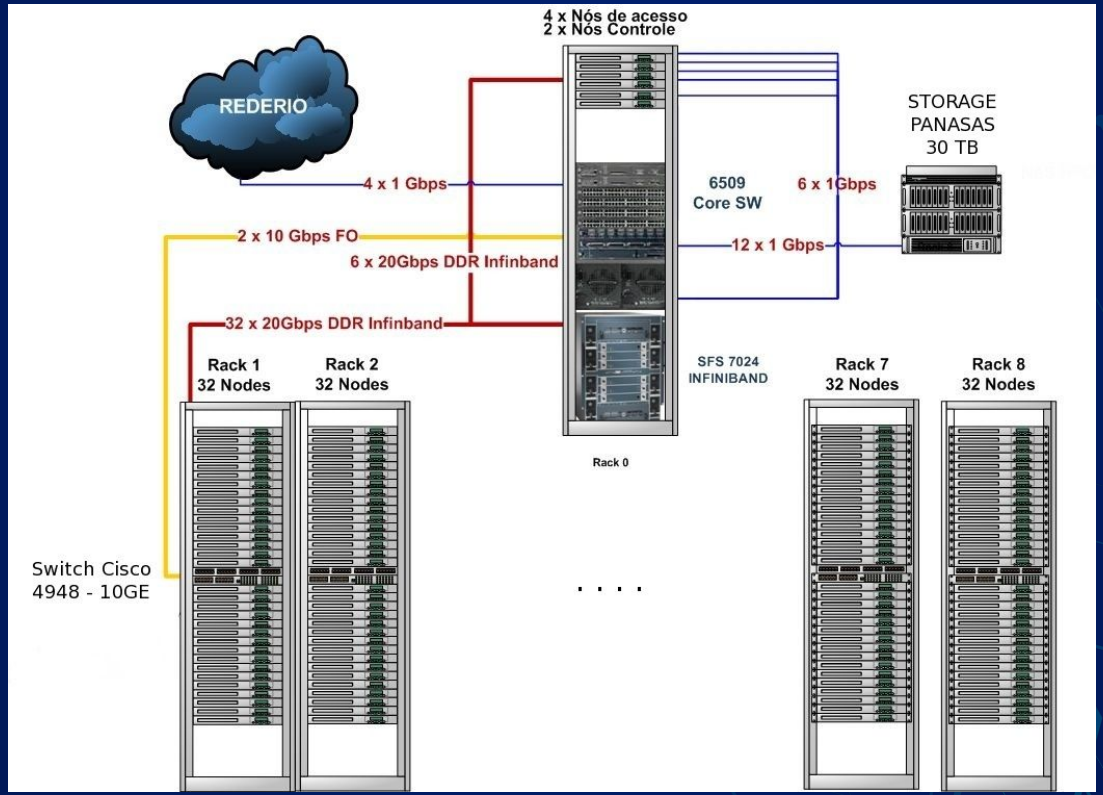
Cluster Netuno

- Supercomputador com uma arquitetura do tipo cluster (MIMD memória distribuída), instalado em 2008 na Universidade Federal do Rio de Janeiro, sendo classificado como o 138º computador mais rápido do mundo e o mais rápido da América Latina na lista Top500 (https://www.top500.org).
- Tinha 10 gabinetes, oito gabinetes com 32 servidores, de computação, um dos gabinetes com um switch Infiniband (20 Gbps) e outro Ethernet (1 Gbps) e um gabinete com 30 Tbytes para um sistema de armazenamento paralelo de alto desempenho e 100 Tbytes para armazenamento com NFS.

Cluster Netuno

- Tinha 256 nós de computação e 4 acessos nós, todos conectados por uma rede Infiniband.
- Cada nó de computação e acesso contém 2 processadores Intel E-5430 de 64 bits com 4 núcleos que compartilham 16 GiB de memória, disco rígido de 160 GB para o sistema operacional, além de interface Ethernet e Infiniband.
- Sistema operacional Linux CentOS, com biblioteca OpenMPI, e drivers para Infiniband OpenFabrics Enterprise Drivers (OFED).
- Desempenho sustentado de 16,2 TFlops.

Cluster Netuno



Obrigado !

Arquitetura de Computadores
Uma Introdução

Mais recursos em:
https://simulador-simus.github.io