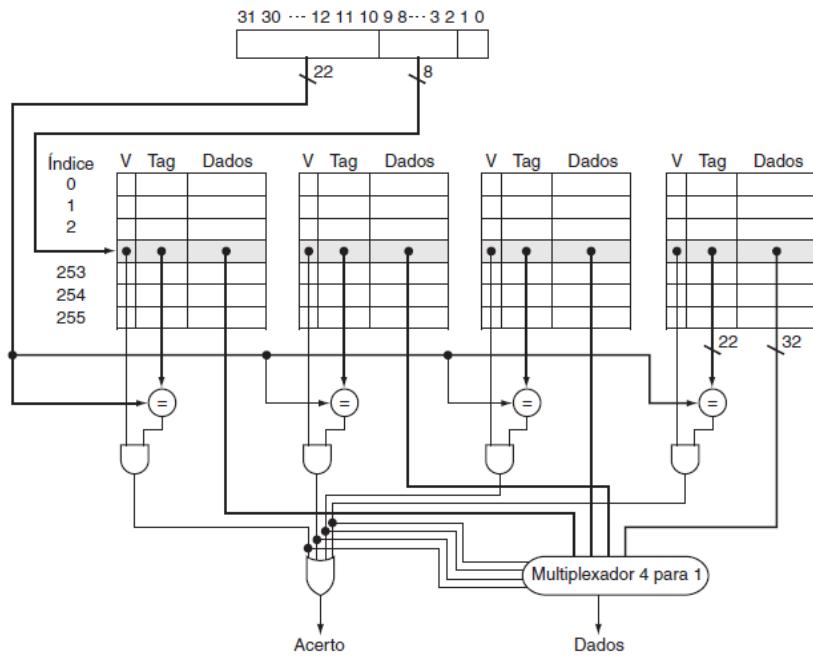


Lista de Exercícios

Arquitetura de Computadores III

1) Explique o funcionamento da arquitetura da figura abaixo:



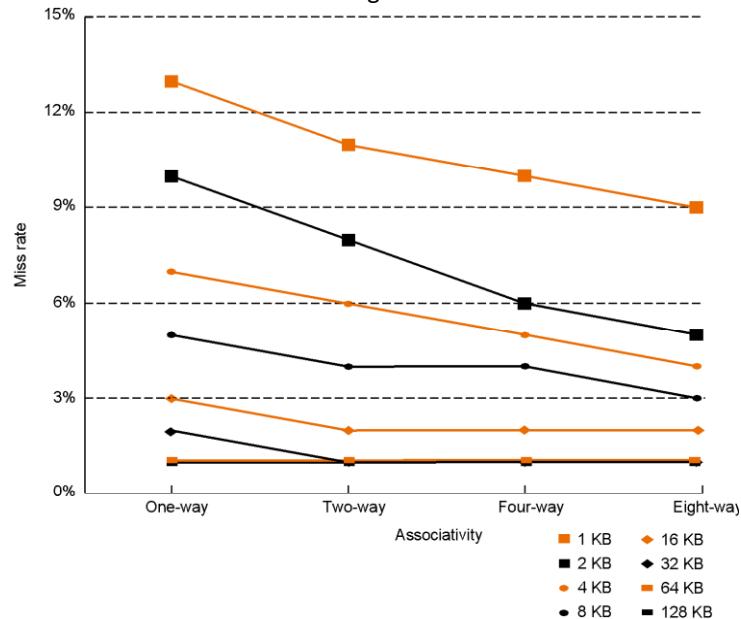
2) Calcule o tamanho em kbytes da memória da questão 1.

3) Supondo cache com mapeamento direto, com 64 kB de dados, linha com uma palavra de 32 bits (4 bytes), e endereços de 32 bits. Quantos bits tem a cache no total?

4) Supondo cache com mapeamento direto, com 16 kB de dados, blocos de 4 palavras, sendo cada palavra de 32 bits e endereços de 32 bits. Quantos bits tem a cache no total?

5) Supondo uma cache com mapeamento conjunto associativo de 2 vias, com 1024kB de dados, palavra de dados de 64 bits e endereço de 64 bits. Cada via possui um bloco de 8 palavras de dados. Quantos bits tem a cache no total?

6) Explique o impacto da associatividade conforme figura abaixo:



Lista de Exercícios

Arquitetura de Computadores III

7) Explique a afirmação seguinte: "Existem vantagens em separar a cache L1 em dados e instruções". Correlacione conceitos.

8) Explique a afirmação seguinte: "A TLB acelera o desempenho no acesso aos dados em memória". Correlacione conceitos. Por exemplo, enumere e explique.

9) Explique de maneira quantitativa, o efeito do miss penalty no acesso a memória principal.

10) Supondo um processador que executa um programa com: CPI = 1.1, 50% aritm/lógica, 30% load/store, 20% desvios. Supondo que 10% das operações de acesso a dados na memória sejam misses. Cada miss resulta numa penalidade de 50 ciclos. Qual o CPI final? E se tiver 1% de miss ratio no fetch de instruções?

11) Suponha que o processador tenha um CPI de 1,0 e que todas as referências acertem na cache primária a uma velocidade de clock de 5GHz (0,2ns). O tempo de acesso à memória principal é de 100ns com todos os tratamentos de faltas. Taxa de falhas por instrução na cache primária é de 2%. O quanto mais rápido será o processador se acrescentarmos uma cache secundária com tempo de acesso de 5ns para um acerto ou uma falha e que seja grande o suficiente para que a taxa de falhas na L2 seja de 0,5%?

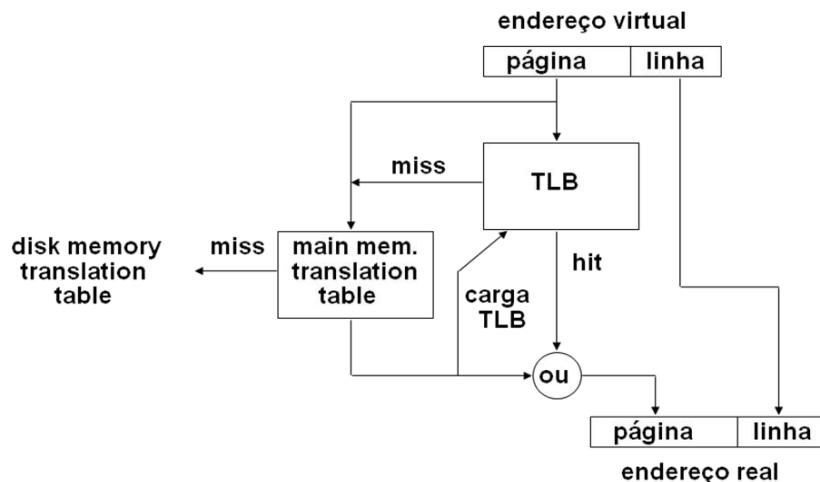
12) Explique os três gráficos abaixo (uma explicação que reúne os três gráficos em um só contexto):



13) Do ponto de vista de desempenho e custo, explique vantagens dos mapeamentos: i) direto, ii) completamente associativo, iii) associativo por conjunto.

14) Explique como transformar endereçamento original do programa no endereçamento real.

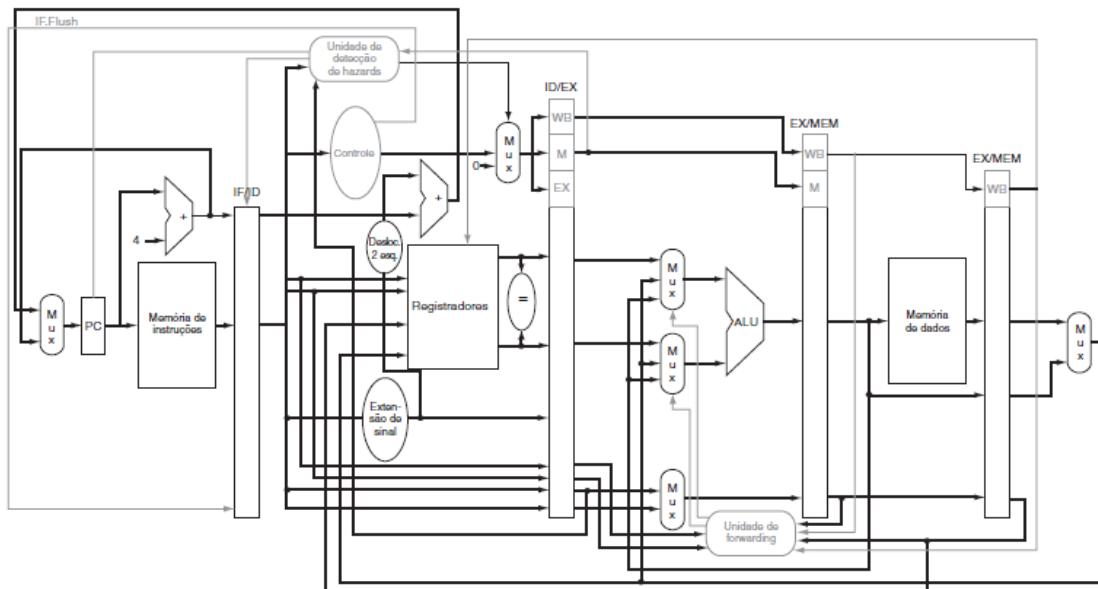
15) Explique a figura abaixo:



Lista de Exercícios

Arquitetura de Computadores III

- 16) Explique memória virtual como solução de um problema em uma arquitetura de computadores.
- 17) Relacione algumas métricas para explicar as vantagens de caches multinível.
- 18) Explique como se dá o acesso à cache e memória principal, considerando a geração de um endereço virtual, a tradução para o endereço real com utilização da TLB, e possível miss na TLB e/ou na cache.
- 19) Um projetista de processador com 5 estágios de pipeline optou pela escrita em registradores em dois estágios. No quarto estágio para instruções lógicas e aritméticas e no quinto estágio para instruções de acesso à memória. No entanto, a equipe de projeto rejeitou a proposta. Você faz parte da equipe. Explique por que o projeto foi rejeitado com base nos conceitos discutidos na disciplina.
- 20) Um projetista de processador com 5 estágios de pipeline fez uso de uma tabela associativa para predição dinâmica de desvios. No entanto, esta abordagem não foi aprovada pela equipe de projeto. Sendo você desta equipe, explique por que o projeto foi recusado. Use uma abordagem quantitativa complementar a sua explicação.
- 21) Para a figura abaixo explicar o que acontece na execução das instruções abaixo (fluxo 1 e fluxo 2) em ordem de entrada no *pipeline* do processador. Apresente o cálculo do CPI para cada fluxo de instruções.



Instruções do fluxo 1:

1. `lw $t0, 0($t0)`
2. `lw $t1, 0($t0)`
3. `beq $t0, $t1, Exit` # considere existência do desvio
4. `sw $t1, 0($t0)`
5. `lw $t3, 0($t1)`
6. `Exit:`
 `sub $t2, $t3, $s4`
7. `slt $s5, $t2, $t3`
8. `sw $t5, 0($t0)`
9. `lw $t2, 0($s2)`
10. `add $s4, $s2, $s1`

Instruções do fluxo 2:

1. `lw $t0, 0($t0)`
2. `lw $t1, 0($t0)`
3. `sw $t1, 0($t0)`

Lista de Exercícios

Arquitetura de Computadores III

```

4.      lw $t3, 0($t1)
5.      slt $s5, $t0, $t3
6.      beq $t0, $s5, Exit # considere existência do desvio
7.      sub $s2, $s3, $s1
8.      lw $t2, 0($s2)
9. Exit:    sub $t2, $t3, $s4

```

22) Utilize o exercício anterior para explicar o que aconteceria para cada fluxo de instruções, em uma figura sem o bloco de adiantamento de dados. Apresente o cálculo do CPI.

23) Com base no que foi estudado na disciplina e discutido em sala de aula, aponte e explique modificações na arquitetura do MIPS da figura acima para que este suporte 4 threads por IMT.

24) Relacione: banco de registradores, janela de instruções distribuídas, renomeação de registradores, buffer de reordenamento.

25) Supondo um processador superescalar com a seguinte configuração:

4 unidades funcionais - 2 somadores, 1 multiplicador, 1 load/store.

Execução fora de ordem com suporte a renomeação de registradores.

Pode executar 4 instruções por ciclo em cada estágio do pipeline.

Latências

- somador (realiza subtração por complemento de 2) - 1 ciclo
- multiplicador - 2 ciclos
- load/store - 2 ciclos

Deve ser executado o seguinte programa:

ADD R4, R3, R4

MUL R4, R7, R3

SW R11, 100 (R5)

SUB R6, R11, R5

ADD R11, R7, R3

SUB R7, R4, R10

ADD R8, R5, R3

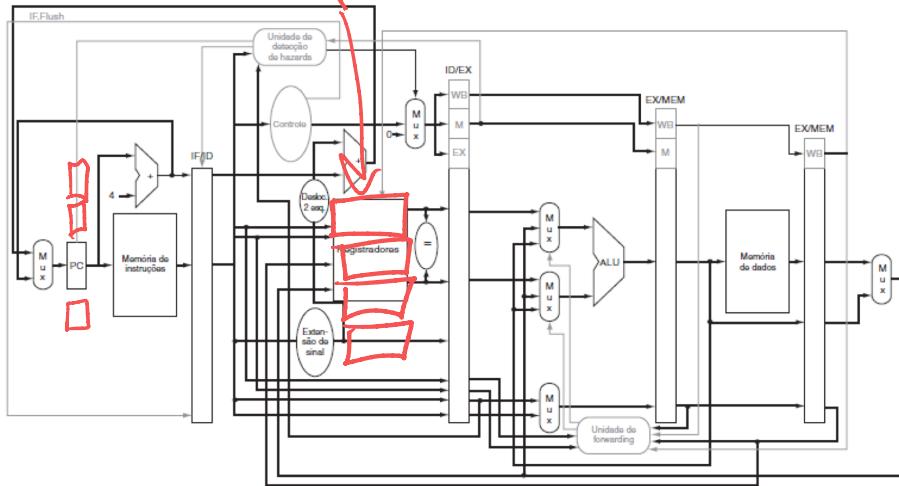
Resolva: Preencha o quadro de execução abaixo e coloque setas no quadro (informe os registradores) que indiquem dependências falsas (tracejadas) e verdadeiras (contínuas). Se for necessário, aumente qtd. de linhas de execução ou não preencha todas.

ciclo	somador 1	somador 2	multiplicador	load/store
1				
2				
3				

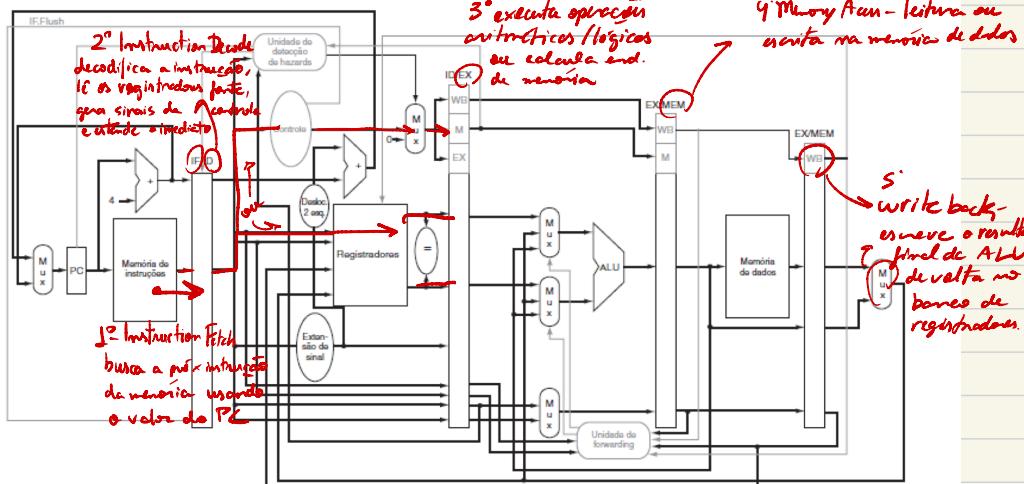
23) Com base no que foi estudado na disciplina e discutido em sala de aula, aponte e explique modificações na arquitetura do MIPS da figura acima para que este suporte 4 threads por IMT.

23) IMT - Interleaved Multithreading → alternância entre os threads de grão fino (de instruções por instruções) para suportar 4 threads, devemos treinar o banco de registradores central por 4 bancos independentes, entendo que o contexto de uma thread não envolve o de outra.

Além disso, cada thread deve ter o seu próprio PC, pra que cada um mantenha seu fluxo de execução independente por si só, também deve ser implementada a lógica de seleção e execução de threads, pq realizar o interleaving efetivamente.



Funcionamento Normal:



24) Relacione: banco de registradores, janela de instruções distribuídas, renomeação de registradores, buffer de reordenamento.

Em uma arquitetura superscalar, podemos implementar a execução de instruções, o despacho e rematracção dentro das instruções fora de ordem. Nela, por sua vez, implementamos uma janela de instruções distribuída, em que agrupamos instruções entre a etapa de decodificação e a de execução, pré-decodificadas, para verificar a melhor execução paralela (a de maior desempenho).

Além disso, também é implementado um buffer de reordenamento, pois algumas instruções não podem terminar fora de ordem, então de uma FIFO que atua em garantir que as instruções não executam em uma ordem predefinida.

Por fim, também implementamos a renomeação de registradores, p/ que dependências falsas (como WAR e WAW) sejam previnidas ao renomear temporariamente os registradores em conflito.

25) Supondo um processador superescalar com a seguinte configuração:

4 unidades funcionais - 2 somadores, 1 multiplicador, 1 load/store.

Execução fora de ordem com suporte a renomeação de registradores.

Pode executar 4 instruções por ciclo em cada estágio do pipeline.

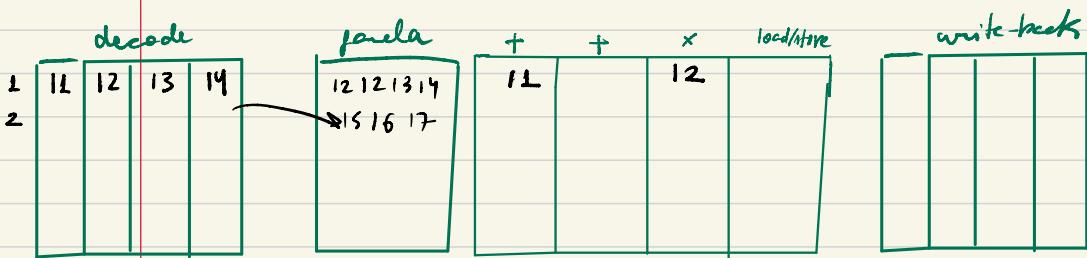
Latências

- somador (realiza subtração por complemento de 2) - 1 ciclo
- multiplicador - 2 ciclos
- load/store - 2 ciclos

Deve ser executado o seguinte programa:

- 11- ADD R4, R3, R4 + *(R4 é resultado)*
12- MUL R4, R7, R3 × *(R4 = RA)*
13- SW R11, 100 (R5) *(leitura)*
14- SUB R6, R11, R5 + *(RAK)*
15- ADD R1, R7, R3 + *(R1 é resultado)*
16- SUB R7, R4, R10 +
17- ADD R8, R5, R3 +

25)



Resolução: Preencha o quadro de execução abaixo e coloque setas no quadro (informe os registradores) que indiquem dependências falsas (tracejadas) e verdadeiras (contínuas). Se for necessário, aumente qtd. de linhas de execução ou não preencha todas.

ciclo	somador 1	somador 2	multiplicador	load/store
1	ADD R4, R3, R4	SUB R6, R11, R5	MUL RA, R7, R3	SW R11, 100(R5)
2	ADD RB, R7, R3	ADD R8, RS, R3	MUL RA, R7, R3	SW R11, 100 (RS)
3	SUB RC, R4, R10			

*Se R11 preservar
seus valores na
memória (estão
é leitura-escrita
entre R6 e R11)*

Lista de Exercícios **Arquitetura de Computadores III**

- 26) Explique: “Um processador many-core executa 32 threads simultaneamente e suporta 128 threads. Ele não possui núcleos em quantidade igual ao número de threads executadas ou suportadas”. Faça considerações sobre o tipo de pipeline e características arquiteturais que suportam as afirmações acima.
- 27) Um consultor, ao analisar a execução de uma aplicação, sugeriu ao analista de suporte de uma empresa desativar o SMT do processador single-core (máquina com um processador que possui um núcleo). Explique por que desta sugestão usando fundamentos discutidos em sala de aula.
- 28) Um cientista/engenheiro da computação optou por usar um processador com dois núcleos Xeon em detrimento do uso de um processador com um núcleo Xeon SMT de duas threads. Todas as demais características dos processadores são iguais (e.g. frequência, tamanho de cache, etc...). Explique esta decisão do cientista da computação.
- 29) Qual a vantagem de: i) executar instruções fora de ordem se há buffer de reordenamento e ii) múltiplas janelas de instruções se IPC não é muito maior do que 2?
- 30) Explique, conte uma história, que justifique o motivo para o desenvolvimento dos processadores multicore.
- 31) Justifique o uso da rede-em-chip em relação ao barramento, com base em três argumentos, exceto escalabilidade.
- 32) Diferencie/Explique: Arquiteturas UMA e Arquiteturas NUMA.
- 33) Diferencie: Escalabilidade da Rede versus Escalabilidade da Aplicação.
- 34) Explique os tipos de paralelismo segundo classificação de Flynn.

26) Explique: "Um processador many-core executa 32 threads simultaneamente e suporta 128 threads. Ele não possui núcleos em quantidade igual ao número de threads executadas ou suportadas". Faça considerações sobre o tipo de pipeline e características arquiteturais que suportam as afirmações acima.

32 threads simultâneos = 32 núcleos
suportar 128 threads $\Rightarrow 128/32 = 4$ threads por núcleo ^{2 threads possíveis}

Uma arquitetura etada é uma arquitetura superscalar, tem múltiplos núcleos e memória distribuída.

Na máquina possui núcleos em qtd igual ao n. de threads executados, pois ela pode suportar 1 thread por núcleo, podendo implementar SMT, IMT ou BMT. Dessa forma, a arquitetura Many-core aproveita de técnicas multi-threading p/ executar + threads do que o número de núcleos disponíveis.

27) Um consultor, ao analisar a execução de uma aplicação, sugeriu ao analista de suporte de uma empresa desativar o SMT do processador single-core (máquina com um processador que possui um núcleo). Explique por que desta sugestão usando fundamentos discutidos em sala de aula.

→ Foi pedido p/ desativar o SMT de processador single-core, pois não faz sentido implementar paralelização de instruções em um único núcleo, pois os threads competem pelos recursos do único núcleo, causando contention e levando à diminuição de desempenho. → (pequena unidade funcional, cache e barramento).

28) Um cientista/engenheiro da computação optou por usar um processador com dois núcleos Xeon em detrimento do uso de um processador com um núcleo Xeon SMT de duas threads. Todas as demais características dos processadores são iguais (e.g. frequência, tamanho de cache, etc...). Explique esta decisão do cientista da computação.

→ escolhendo um processador com 2 núcleos, é possível ter + recursos para os threads de execução (+ unidades funcionais, cache, barramento, etc), enquanto um núcleo de duas threads faria com que uma thread acometesse todos os recursos de 1 só núcleo, causando baixa de desempenho.

29) Qual a vantagem de: i) executar instruções fora de ordem se há buffer de reordenamento e ii) múltiplas janelas de instruções se IPC não é muito maior do que 2?

i) Ao executar instruções fora de ordem, permitimos que o processador execute algumas instruções antes que outras tenham, melhorando a velocidade e desempenho. O buffer de reordenamento, não vai garantir que instruções com dependências e conflitos não sejam executadas na ordem errada, causando resultados errados / inválidos. Além, se tem um reordenamento só para instruções conflitantes, os demais que não possuem conflito ainda podem acharne de serem "otimizados".

ii) IPC - Instruções Por Círculo

A janela de instruções é o conjunto de instruções que o processador analisa simultaneamente p/ identificar oportunidades de execução paralela.

Então, mesmo que a qtd. de instruções por ciclo seja baixa, não elimina a necessidade de múltiplas janelas que garantem que as instruções (que já devem ser executadas) executem de forma + paralela e reduzindo o tempo ocioso das unidades funcionais.

30) Explique, conte uma história, que justifique o motivo para o desenvolvimento dos processadores multicore.

→ Os desenvolvedores de hardware já estavam no limite de aumento da frequência dos processadores e do aumento de paralelismo das instruções. Então, a solução de uma delas passou na Lei de Moore: se diminuirmos + ainda o tamanho dos transistores, e aumentar a qtd de processadores no chip de processadores, e os chamarmos de nucleos, temos os processadores MULTICORE!

31) Justifique o uso da rede-em-chip em relação ao barramento, com base em três argumentos, exceto escalabilidade.

A largura de banda é limitada e compartilhada por todos os nós no barramento, enquanto a rede-em-chip não é afetada.
→ apenas um núcleo pode transmitir dados por vez, criando gargalos
→ permite comunicação simultânea

A latência também é afetada pelo fio no barramento, enquanto na rede em chip ela é afetada somente por latências nos roteadores.

No barramento, os nós conectados a múltiplos núcleos em um único caminho, aumenta a congestão de trabalho e o consumo de energia. Já os NoCs usam fios ponto-a-ponto (somente entre roteadores) e rotacionais, reduzindo a congestão elétrica total em cada fio e consequentemente o consumo.

No barramento, uma falha no caminho pode comprometer toda a comunicação do sistema, e NoCs são modulares e reutilizáveis p/ modificar ou substituir qualquer bloco.

32) Diferencie/Explique: Arquiteturas UMA e Arquiteturas NUMA.

UMA

- Uniform Memory Access
- memória compartilhada centralizada
- Problemas de coerência de cache
- Escalabilidade limitada
- Existe um único bloco de memória física compartilhada entre todos os processadores

↳ (paralelamente

conectadas por

um barramento)

NUMA

- Non-Uniform Memory Access
- memória compartilhada distribuída
- Cada nó da rede possui processadores e um pedaço da memória total.
- escalabilidade superior
- ex: supercomputadores
- NoCs não muito implementados



Ambas são arquiteturas de multiprocessadores em rede, memória compartilhada

33) Diferencie: Escalabilidade da Rede versus Escalabilidade da Aplicação.

da Rede:

capacidade da Infraestrutura de rede de suportar um aumento do nr. de usuários

- desempenho
- disponibilidade
- segurança

da Aplicação

capacidade de um software em execução de lidar com o aumento de usuários

- performance
- disponibilidade
- qualidade do serviço

34) Explique os tipos de paralelismo segundo classificação de Flynn.

SD (Single Data)

MD (Multiple Data)

SI (Single Instruction)

SISD von Neumann

SIMD maquinas paralelas

MI (Multiple Instructions)

MISD
máis eficiente

MIMD
multiprocessoadores e multicore