Iniciado em	sexta, 3 set 2021, 10:45
Estado	Finalizada
Concluída em	sexta, 3 set 2021, 12:37
Tempo empregado	1 hora 52 minutos
Avaliar	**13,20** de um máximo de 20,00(**66**%)

Questão **1**

Completo

Atingiu 3,50 de 5,00

Para cada parte deste exercício, **considere a sequência de atualizações realizadas na cache, por exemplo, a operação da letra b) considera as alterações realizadas na letra a) do exercício. Consulte a Figura 1 para este exercício.**

Cada parte deste exercício especifica uma sequência de uma ou mais operações da CPU na forma:

P# : <op> <endereço> [<valor>]

Onde P# designa a CPU (por exemplo, P0), <op> é a operação da CPU <endereço> indica o endereço da memória e <valor> indica a nova palavra a ser atribuída em uma operação de escrita.

Para cada letra abaixo **APRESENTE**:

- **Estado resultante** (ou seja, estado de coerência, tags e dados) da **cache e da memória** após a ação indicada. Apresente apenas os blocos que mudam, exemplo, P0.B0: (I, 120, 00 01) indica que após a ação o bloco B0 da CPU 0 tem o estado final I, tag de 120 e palavras de dados 00 e 01.
- **Mensagens** enviadas no barramento.
- **Quando não ocorrem mudanças, diga o que aconteceu** (ex: um acerto de leitura ou escrita).
- **Quantos ciclos de stall** são gerados por cada item, dadas as informações na **Tabela 1**.

Então para cada item serão avaliados: o estado atual das caches e da memória (o que foi alterado), mensagens enviadas no barramento, informações adicionais (caso necessário), e os ciclos de stall do processador.

Lembre-se:

- **Acertos de leitura e escrita** de CPU não geram ciclos de stall.
- **Falhas de leitura e escrita** de CPU geram ciclos de stall $N_{memória}$ e N_{cache}, se satisfeitas pela memória e pela cache, respectivamente.
- **Acertos de escrita** de CPU que geram uma invalidação incorrem em $N_{invalidação}$ de ciclos de stall.
- Um **write-back** de um bloco, seja devido a um conflito, seja devido à solicitação de outro processador a um bloco exclusivo, incorre em $N_{write\text{-}back}$ ciclos de stall adicionais.

Parâmetro	Implementação
$N_{memória}$	150
N_{cache}	100
$N_{invalidação}$	50
$N_{write\text{-}back}$	10

Mapeamento: **B1**: tags 108, 128 **B2**: tag 110 **B3**: tag 118

a) P15: read 118

b) P15: write 118 <- 80

c) P15: write 110 <-- 40

d) P1: read 108

e) P15: write 128 <-- 70

a) • Read miss (B3 no estado I);

• Satisfeito pela cache : 100 stalls ;

• P15 - B3: S , 118, 00 18

Total stalls : 100.

b) • Write hit (B3 no estado S);

• P15 - B3 : M, 118, 00 80;

• Invalidação : P1 - B3 : I, 118, 00 18 : 50 stalls;

Total stalls : 50.

c) • Write miss (B2 no estado I) ;

• Satisfeito pela cache : 100 stalls;

• P15 - B2 : M, 110, 00 40;

• Invalidação : P0 - B2 : I, 110, 00 30 : 50 stalls;

• write-back P0 - B2 : 10 stalls;

Total stalls : 160.

d) • Read miss (bloco não encontrado);

• Satisfeita pela cache : 100 stalls;

• P1 - B1 : S, 108, 008;

• Write-back P1 - B1 : 10 stalls.

Total stalls : 110.

e) • Write miss (bloco não encontrado);

• Satisfeita pela memória : 150 stalls;

• P15 - B0 : M, 128, 00 70;

Total stalls : 150.

Comentário:
a) Read miss, satisfeito pela memória -> 150 ciclos de stall (-0,5)
c) Nwriteback + Ncache = 110 (-0,5)
Mem(110, 00 30)
d) Mem (128, 00 68)
Nmem + Nwriteback = 160 ciclos de stall (-0,5)

Questão **2**

Completo

Atingiu 4,70 de 5,00

Os estados da cache L1 são indicados com M, S e I, de Modificado, *Shared* e Inválido. As caches L2 e as memórias têm diretórios (**FIGURA 2**). Os estados do diretório são indicados por DM, DS e DI, de Diretório Modificado, Diretório *Shared* e Diretório Inválido (*uncached*). **Para cada parte do exercício considere a sequência de atualizações realizadas na cache e memória, por exemplo, a operação da letra b) considera as alterações realizadas na letra a) do exercício.**

Cada parte especifica uma sequência de uma ou mais operações de CPU na forma: P# : <op> <endereço> [<-- <valor>]. Onde P# designa a CPU (p.ex, P0,0), <op> operação da CPU (p. ex, read, write), <endereço> indica o endereço da memória e <valor> indica uma nova palavra a ser atribuída em uma operação de escrita. **DETALHE para cada item: (i) se ocorreu hit ou miss; (ii) as mudanças que ocorrem nas memórias e nas caches; e (iii) se ocorre write-back.**

Mapeamento L1: **B0**: tags 100,130 **B1**: tags 108, 118

a) P0,1: read 100

b) P0,0: write 108 <- - 38

c) P0,1: write 118 <-- 48

d) P3,1: write 130 <- - 48

e) P3,1: read 108

a) • Read miss - L1

- P0,1 - B0 : S, 100, 00 10;
- P0,0 - B0 : S, 100, 00 10;
- wite-back (130, 00 68 e 100, 00 10);
- L2$,0 - B0 : DS, P0,1; P0,0, 100, 00 10);
- L2$,0 - B2 : DI, - , 130, 00 68);
- M0 - 100 : DS, C0, 00 10;
- M1 - 130 : DI, - , 00 68.

b) • Write hit - L1

- P0,0 - B1 : M, 108, 00 38;
- Invalidate : P3,1 - B1 : I, 108, 00 08;
- L2$,0 - B1 : DM, P0,0, 108, 00 38);
- L2$,1 - B1 : DI, - , 108, 00 08);
- M0 - 108 : DM, C0 , 00 38;

c) • Write hit - L1

- P0,1 - B1 : M, 118, 00 48
- L2$,0 - B1 : DM, P0,1, 118, 00 48);
- M0 - 118 : DM, C0 , 00 48;

d) • Write miss- L2 e L1

- P3,1 - B0 : M, 130, 00 48
- L2$,0 - B2 : DM, E, 130, 00 48);
- L2$,1 - B0 : DI, -, 120, 00 20);
- M1 - 130 : DM, C1 , 00 48;
- M1 - 120 : DI, - , 00 20;

e) • Read miss - L2 e L1

- P3,1 - B1 : S, 108, 00 38

- P0,0 - B1 : S, 108, 00 38;
- write back (108, 00 38);
- L2$,1 - B1 : DS, P3,1; E, 108, 00 38);
- L2$,0 - B1 : DS, P0,0; E, 108, 00 38);
- M0 - 108 : DS, C0;C1 , 00 38;

Comentário:
a) Write back somente do bloco 130

d) L2$,0 - B2 : DM, E, 130, 00 48); DM não pode ser externo

Questão **3**

Completo

Atingiu 2,00 de 2,00

Implemente a instrução clássica de *test-and-set* usando o par de instruções *Load linked/store-conditional*.

```
foo: MOV R3, #1
     LL R2, 0(R1)
     SC R3, 0(R1)
     BNE R3, #1, foo
```

Comentário:

Questão **4**

Não respondido

Vale 5,00 ponto(s).

Considere a seguinte situação: três processadores P0,0, P0,1 e P3,1 estão tentando obter o *lock* em R1 (Endereço 0x100 que é mapedo para o bloco B0 – Figura 2). 0 (Zero) indica *lock* liberado e 1 (*lock)* não liberado. Suponha que o bloco esteja no processador **P0,0** com a seguinte situação (M, 100, 00 01) e com o mesmo valor no diretório L2,$0 e Memória. **MOSTRE**, com uma tabela, o funcionamento da sincronização entre eles, utilizando o protocolo de sincronização **Diretório** - MSI. Apresente em cada passo da execução o (i) estado do bloco nas caches dos processadores, nos diretórios e memória (ii) as mensagens enviadas; e (iii) a operação executada.

Questão **5**

Completo

Atingiu 3,00 de 3,00

Um dos facilitadores do uso de WSC é o amplo paralelismo no nível de requisição, em contraste ao paralelismo em nível de instrução ou thread. **Liste 2** (duas) potenciais **DESVANTAGENS** de se aumentar o paralelismo em nível de requisição em um WSC. **JUSTIFIQUE A SUA RESPOSTA!**

Ao aumentar o paralelismo em nível de requisição, a comunicação e o estado são compartilhados em centros de processamento de dados ao redor do mundo, o que pode fazer com que ocorra problemas na consistência dos dados. Além disso, são necessárias mais máquinas para aumentar o throughput, no caso de compra de máquinas mais baratas, podem ocorrer falhas mais frequentemente, fazendo com que ocorra a necessidade da implementação de mecanismos para evitar a perda de dados.

Comentário:

◄ Aula_18

Seguir para...

Prova_Suplementar ►