

- 1) O multiprocessador multicore ilustrado na Figura 1 representa uma arquitetura de memória compartilhada simétrica comumente implementada. Cada processador possui uma cache única e privada com coerência mantida usando o protocolo de coerência snooping visto em sala de aula. Cada cache é mapeada diretamente, com quatro blocos cada um contendo duas palavras. Para simplificar a ilustração, a tag de endereço de cache contém o endereço completo e cada palavra mostra apenas dois caracteres hexadecimais (8 bits), com a palavra menos significativa à direita. Os estados de coerência são designados por M, S e I (Modificado, Compartilhado e Inválido).

Para cada item deste exercício, assuma o estado inicial da cache e da memória, conforme ilustrado na Figura 1. Cada item deste exercício especifica uma sequência de uma ou mais operações de CPU da forma:

P #: <op> <endereço> [<valor>]

onde P # designa a CPU (por exemplo, P0), <op> é a operação da CPU (por exemplo, leitura ou escrita), <endereço> indica o endereço da memória e <valor> indica o novo valor a ser atribuído em uma operação de escrita na memória. Sempre trate cada ação dos itens como aplicada ao estado inicial mostrado na Figura 1, ou seja, as ações dos itens não são aplicadas em sequência. Qual é o estado resultante (ou seja, estado de coerência, tags e dados) das caches e memória após cada ação especificada nos itens? Mostre apenas os blocos que mudam; por exemplo, P0.B0: (I, 120, 00 01) indica que o bloco B0 da CPU P0 tem o estado final de I, tag de 120 e as palavras de dados 00 e 01. Além disso, qual valor é retornado por cada operação de leitura?

- i) P0: lê 120
- ii) P0: escreve 120 <-- 80
- iii) P3: escreve 120 <-- 80
- iv) P1: lê 110
- v) P0: escreve 108 <-- 48
- vi) P0: escreve 130 <-- 78
- vii) P3: escreve 130 <-- 78

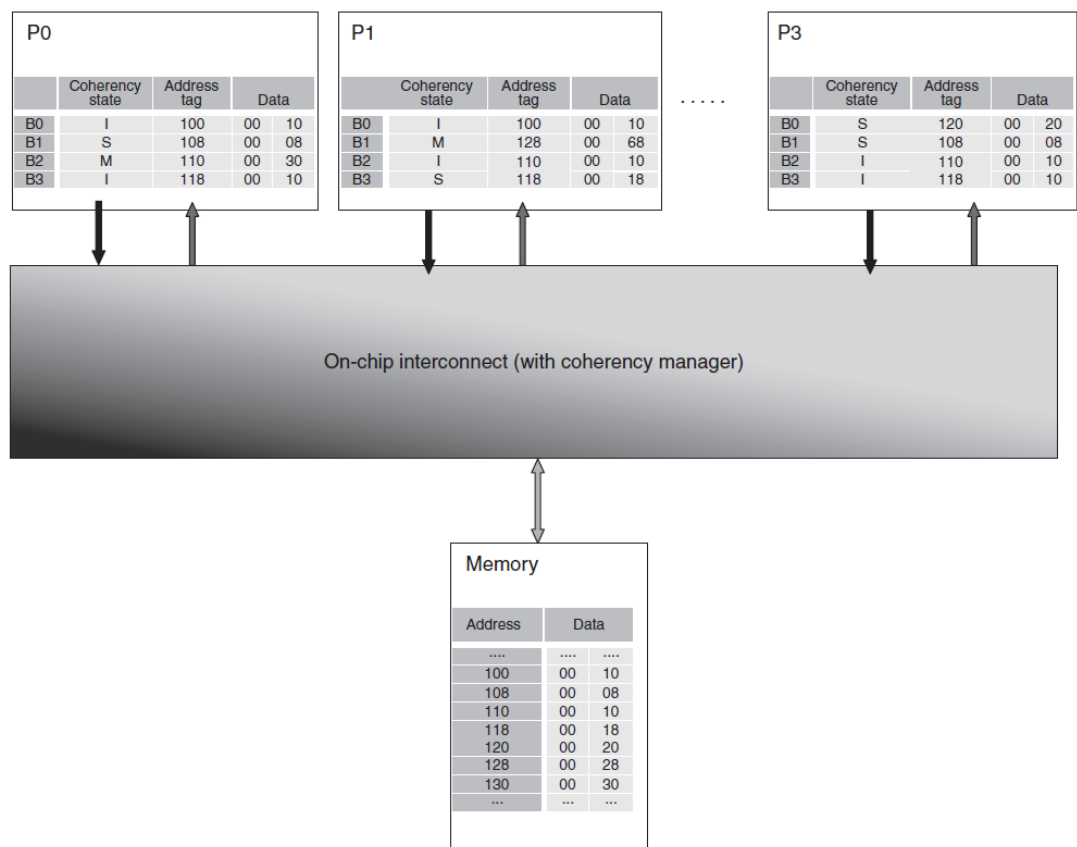


Figura 1. Arquitetura de memória compartilhada simétrica

- 2) O desempenho de um multiprocessador que utiliza o protocolo snooping para coerência da cache depende de muitos problemas específicos de implementação que determinam a rapidez com que uma cache recebe dados de um bloco que está em estado exclusivo. Em algumas implementações, em uma falta de leitura da CPU em um bloco de cache exclusivo na cache de outro processador, os dados são fornecidos mais rapidamente do que em uma falta em um bloco na memória. Isso ocorre porque as caches são menores e, portanto, mais rápidas que a memória principal. Por outro lado, em algumas implementações, nas faltas de blocos que estão na memória, os dados são fornecidos mais rapidamente que blocos que estão nas caches. Isso ocorre porque as caches geralmente são otimizadas para referências da CPU que ela está conectada, em vez de acessos devidos ao protocolo snooping. Para o multiprocessador ilustrado na Figura 1, considere a execução de uma sequência de operações em uma única CPU onde
- As ocorrências de leitura e escrita da CPU não acarretam em ciclos de travamento da CPU
 - As faltas de leitura e escrita da CPU acarretam em ciclos de travamento da CPU denominados $N_{\text{memória}}$ e N_{cache} se o bloco estiver na memória ou na cache, respectivamente.
 - Acertos de escrita na CPU que geram invalidação de outra(s) cache(s) geram $N_{\text{invalidação}}$ ciclos de travamento da CPU.

- A escrita de um bloco na memória, devido a um conflito ou à solicitação de outro processador para um bloco exclusivo, acarreta em $N_{\text{writeback}}$ ciclos adicionais de travamento da CPU.

Considere duas implementações com diferentes características de desempenho resumidas na Figura 2. Considere a seguinte sequência de operações assumindo o estado inicial da cache na Figura 1. Por uma questão de simplicidade, suponha que a segunda operação comece após a primeira ser concluída (mesmo que elas estejam em processadores diferentes):

P1: leia 110

P3: leia 110

Para a Implementação 1, a primeira leitura gera 50 ciclos de travamento da CPU porque a leitura é satisfeita pela cache de P0. P1 para durante 40 ciclos enquanto aguarda o bloco e P0 para durante 10 ciclos enquanto escreve o bloco de volta na memória em resposta à solicitação de P1. A segunda leitura de P3 gera 100 ciclos de travamento porque sua falta é tratada com acesso à memória, e a sequência das duas operações de leituras gera um total de 150 ciclos de travamento.

Parâmetros	Implementação 1	Implementação 2
$N_{\text{memória}}$	100	100
N_{cache}	40	130
$N_{\text{invalidação}}$	15	15
$N_{\text{writeback}}$	10	10

Figura 2. Latências devido ao protocolo snooping para coerência.

Para as seguintes sequências de operações, quantos ciclos de travamento são gerados por cada implementação?

- (a) P0: lê 120
P0: lê 128
P0: lê 130
- (b) P0: lê 100
P0: escreve 108 <-- 48
P0: escreve 130 <-- 78
- (c) P1: lê 120
P1: lê 128
P1: lê 130
- (d) P1: lê 100
P1: escreve 108 <-- 48
P1: escreve 130 <-- 78

- 3) Considere o sistema de memória compartilhada distribuída ilustrado na Figura 3 que consiste em dois chips de quatro núcleos. O processador em cada chip compartilha uma cache L2 (L2\$) e os dois chips são conectados por meio de uma interconexão ponto a ponto. A memória do sistema é distribuída pelos dois chips. A Figura 4 detalha uma parte deste sistema. $P_{i,j}$ indica o processador i no chip j . Cada processador possui uma única cache L1

com mapeamento direto que contém dois blocos, cada um contendo duas palavras (8 bits cada palavra). Cada chip possui uma única cache L2 com mapeamento direto que contém dois blocos, cada um com duas palavras. Para simplificar a ilustração, as tags de endereço de cache contêm o endereço completo e o conteúdo de cada palavra é mostra do por dois caracteres hexadecimais, com a palavra menos significativa à direita.

Os estados da cache L1 são indicados por M, S e I para Modificado, Compartilhado e Inválido. As caches e memórias L2 têm diretórios. Os estados do diretório são indicados como DM, DS e DI para Diretório Modificado, Diretório Compartilhado e Diretório Inválido. O protocolo de diretório simples utilizado é aquele mostrado em sala de aula. O diretório L2 lista os compartilhadores/proprietários locais e registra adicionalmente se uma linha é compartilhada externamente em outro chip; por exemplo, P1, 0; E indica que uma linha é compartilhada pelo processador local P1, 0 e é compartilhada externamente em algum outro chip. O diretório de memória tem uma lista dos compartilhadores/proprietários de uma linha; por exemplo, C0, C1 indica que uma linha é compartilhada nos chips 0 e 1.

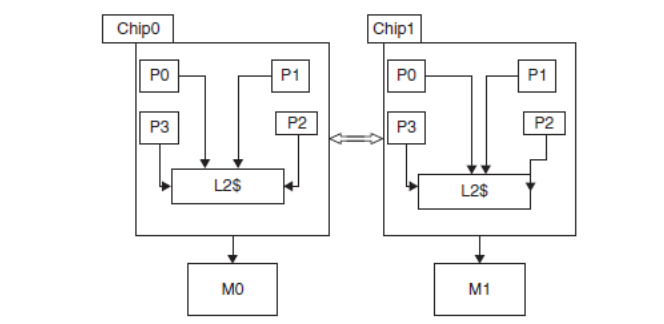


Figura 3. Sistema multiprocessador DSM

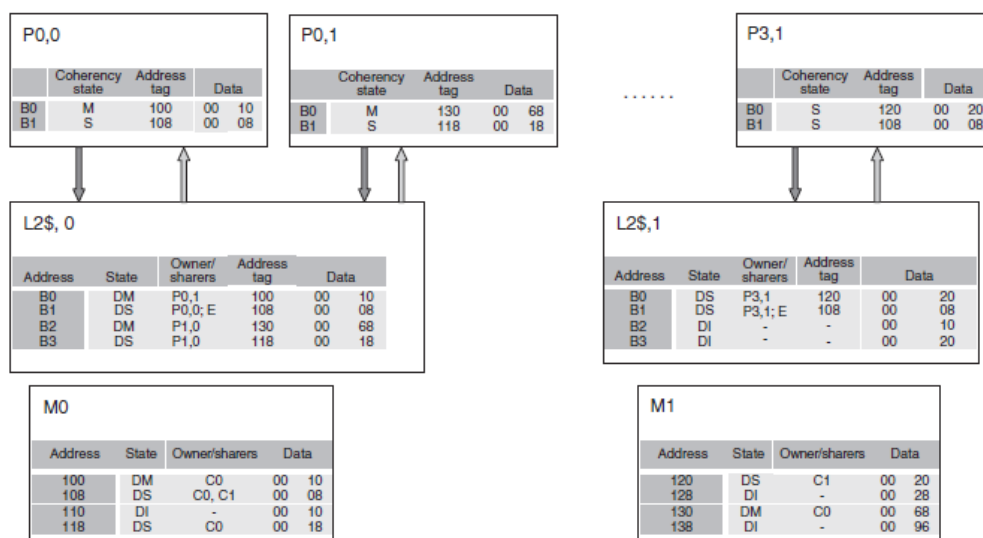


Figura 4. Detalhamento do sistema

Para cada item deste exercício, assuma o estado inicial da cache e da memória conforme ilustrado na Figura 4. Cada item deste exercício especifica uma sequência de uma ou mais operações de CPU da forma:

P #: <op> <endereço> [<valor>]

onde P# designa a CPU (por exemplo, P0,0), <op> é a operação da CPU (por exemplo, leitura ou escrita), <endereço> indica o endereço da memória e <valor> indica o novo valor a ser atribuído em uma operação de escrita na memória. Sempre trate cada ação dos itens como aplicada ao estado inicial mostrado na Figura 4, ou seja, as ações dos itens não são aplicadas em sequência. Qual é o estado resultante (ou seja, estado de coerência, tags compartilhadores/proprietários e dados) das caches e memória após cada ação especificada nos itens? Além disso, qual valor é retornado por cada operação de leitura?

- i) P0,0: lê 100
- ii) P0,0: lê 128
- iii) P0,0: escreve 128 <-- 78
- iv) P0,0: lê 120
- v) P0,0: lê 120
P1,0: lê 120
- vi) P0,0: lê 120
P1,0: escreve 120 <-- 80
- vii) P0,0: escreve 120 <-- 80
P1,0: lê 120
- viii) P0,0: escreve 120 <-- 80
P1,0: escreve 120 <-- 90

- 4) Os protocolos de diretório são mais escaláveis que os protocolos snooping, porque enviam solicitações explícitas e invalidam mensagens somente para os nós que têm cópias de um bloco, enquanto os protocolos snooping transmitem todas as solicitações e invalidações para todos os nós. Considere o sistema de oito processadores ilustrado na Figura 3 e suponha que todas as caches não mostradas na Figura 4 tenham blocos inválidos. Para cada uma das sequências abaixo, identifique quais nós (chip / processador) recebem cada solicitação e invalidação.

- i) P0,0: escreve 100 <-- 80
- ii) P0,0: escreve 108 <-- 88
- iii) P0,0: escreve 118 <-- 90
- iv) P1,0: escreve 128 <-- 98

- 5) Mostre como o protocolo snooping básico mostrado em sala de aula pode ser alterado para uma cache write-through. Qual é a principal funcionalidade de hardware que não é necessária com uma cache write-through comparado com uma cache write-back?