Arquitetura e Organização de Computadores II

Protocolos de Coerência de Cache

Prof. Nilton Luiz Queiroz Jr.

Protocolos de invalidação

- Protocolos de coerência de cache que fazem uso de invalidação tem menor consumo na largura de banda
  - Eles apenas invalidam os dados das caches privadas de outros processadores;
- Quando um processador tenta fazer a leitura de um dado ques está em sua cache, porém está inválido, ocorre um miss na cache e deve ser feita uma nova busca;
- A implementação de um protocolo requer comunicação entre as caches privadas;

Técnicas básicas de implementação

- Em um protocolo de invalidação é necessário usar um barramento para comunicar todas as ações a serem feitas;
  - Todo processador que for fazer uma leitura em memória deve comunicar;
    - Pode ser que o dado esteja na cache, porém inválido!
  - Sempre que um processador for alterar um dado em sua cache ele deve comunicar que está aquele endereço;
    - Desse modo os valores da cache podem ser invalidados;
- Em processadores multicore esse barramento pode ser a conexão entre as caches privadas e a cache compartilhada;

Técnicas básicas de implementação

- Para invalidar um valor:
  - O processador transmite no barramento que está escrevendo no endereço;
  - Todos processadores “escutam” essa transmissão;
  - Os processadores que tem o aquele endereço em sua cache invalidam aquele dado;
- Quando dois processadores tentam escrever ao mesmo tempo as operações de invalidar serão serializadas;
- Para que qualquer escrita seja concluída é necessário que o processador obtenha acesso ao barramento;

Técnicas básicas de implementação

- Quando ocorre um miss na cache é necessário localizar o dado;
- A localização do dado muda conforme a política de escrita na cache:
  - Write-through;
    - O dado estará na memória compartilhada, pois quando foi alterado a escrita na memória compartilhada foi feita;
  - Write-back;
    - O dado pode estar em alguma cache privada, pois pode ser que o bloco ainda não foi substituído, e a escrita na memória primária não foi feita

Técnicas básicas de implementação

- Com uma política write-back, antes de toda leitura que gera um miss em cache, deve-se usar o barramento para avisar que é necessária a leitura naquele endereço;
- Quando é descoberto que o endereço foi alterado, a leitura na memória é abortada e o processador que possui o valor original o transfere para que solicitou;

Protocolo MSI

- Incorpora-se um controlador de estados finitos em cada núcleo;
  - Cada bloco da cache tem possíveis estados;
- Esse controlador responde às solicitações do processador e do barramento alterando o estado do bloco da cache, acessar dados e invalidá-los;
  - Pode ser pensado como um controlador associado aos blocos da cache;
- Esse protocolo incorpora 3 estados:
  - Modificado (Modified)
  - Compartilhado (shared);
  - Invalido (invalid);

Protocolo MSI

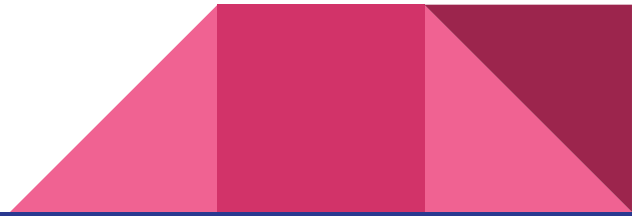
- O estado Modificado indica que o bloco foi alterado na cache privada e está diferente da memória compartilhada;
  - O bloco é “exclusivo”;
- O estado Compartilhado indica que o bloco na cache privada é potencialmente compartilhado;
  - Outros blocos podem ou não ter esse mesmo dado;
- O estado inválido indica que o conteúdo do bloco está desatualizado;
  - Caso seja requisitado o conteúdo daquele bloco irá ocorrer um miss na cache;

Protocolo MSI

- Quando um sinal de invalidação, ou um sinal de miss para escrita é colocado no barramento, todos as outras caches que tem cópias valor armazenado o invalidam;
- Quando um bloco está no estado Modificado e recebe um sinal de miss de write em cache, ele devolve a cópia desse bloco modificado para a memória compartilhada e fornece o bloco com o endereço desejado;

Protocolo MSI

- Como são 3 estados é possível representá-los por 2 bits em cada bloco da cache;
- Somente um determinado endereço só pode estar no estado modificado em uma cache;
  - A cache do processador que fez a última escrita é a única que estará no estado modificado;
    - Podemos também chamar o estado modificado de exclusivo por essa razão;
- Vários blocos podem estar no estado compartilhado;



Protocolo MSI

- Possível estado para as caches de dois processadores de um multiprocessador

Cache do Processador 1

Estado	Tag	Dado	Dado
S	100	123	1080
S	220	700	22
S	300	63	354
S	480	37	3737
S	280	22	23
S	500	60	37773

Cache do Processador 2

Estado	Tag	Dado	Dado
S	500	60	37773
S	220	700	22
S	300	63	354
S	480	37	3737
S	280	22	23
S	100	123	1080

Protocolo MSI

- Caso o processador 1 escrevesse em um dos endereços da tag 500 e o processador 2 escrevesse em um dos endereços da tag 100, o novo estado das caches seria:
  - Essas escritas não poderiam causar miss

Cache do Processador 1

Estado	Tag	Dado	Dado
I	100	123	1080
S	220	700	22
S	300	63	354
S	480	37	3737
S	280	22	23
M	500	73	37773

Cache do Processador 2

Estado	Tag	Dado	Dado
I	500	60	37773
S	220	700	22
S	300	63	354
S	480	37	3737
S	280	22	23
M	100	222	1080

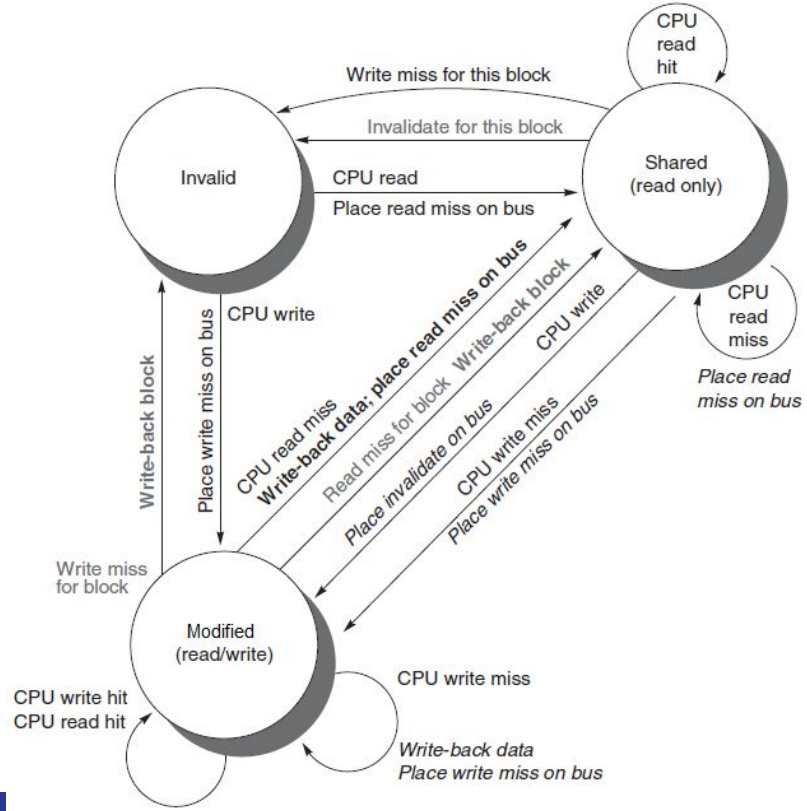
Protocolo MSI

- Qualquer bloco que está no estado compartilhado está com o valor compartilhado terá o mesmo valor da memória compartilhada;
    - Caso ele faça uma leitura de um valor que está marcado como modificado em alguma outra cache privada, cache que contém o valor atual irá “aproveitar” que deve passar o valor e fazer a escrita na memória compartilhada;

Protocolo MSI

- Todos blocos começam no estado inválido;
- De acordo com ações da CPU ou do barramento os estados são alterados;
  - Ações devem ser tomadas para manter a cache coerente;
  - Essas ações podem ocasionar em transições nos estados;

Protocolo MSI



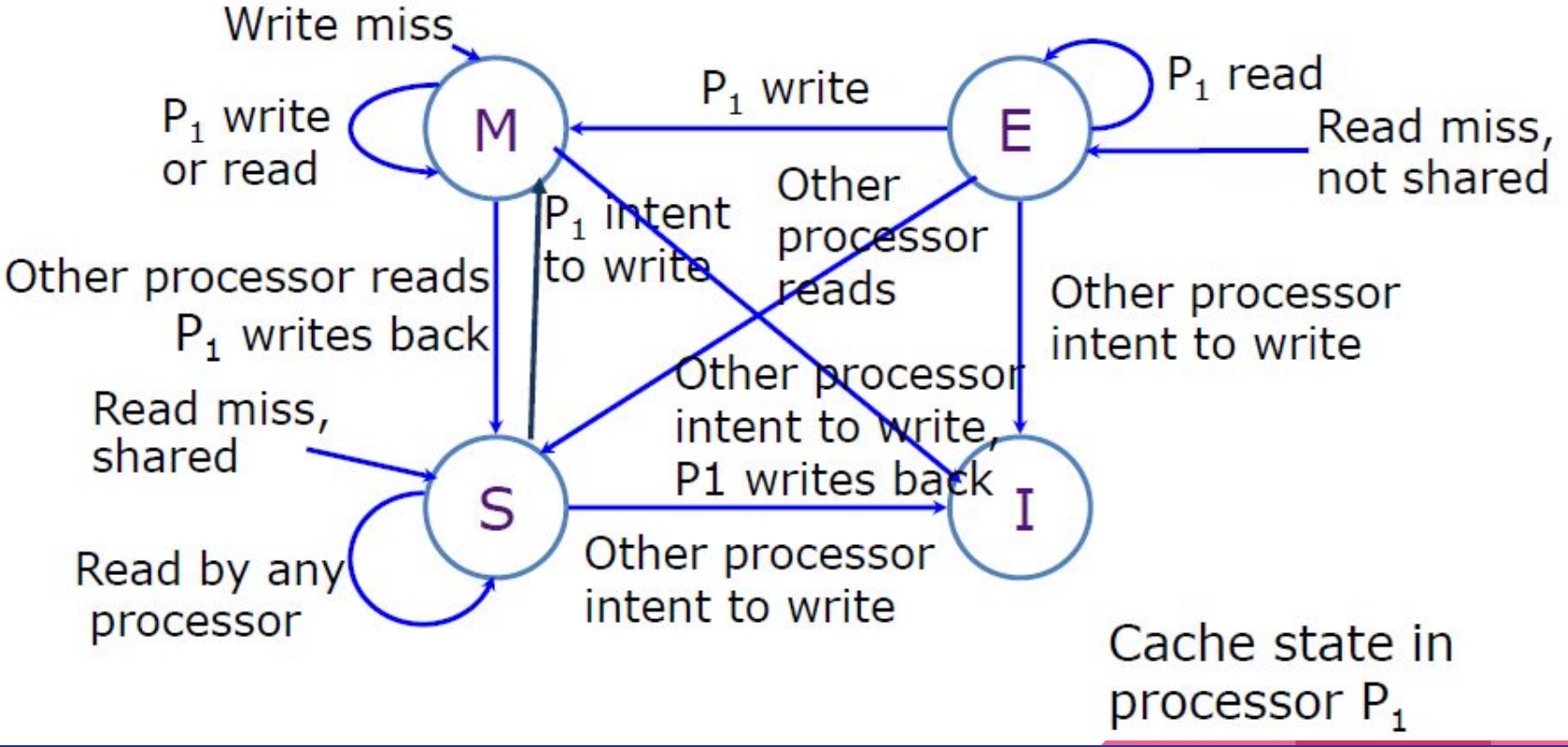
Protocolo MESI

- O protocolo MESI é composto por 4 estados:
  - Modified (Modificado);
  - Exclusive (Exclusivo - não modificado);
  - Shared (Compartilhado);
  - Invalid (Invalido);
- É um aperfeiçoamento do protocolo MSI para reduzir a comunicação;
  - Indica que um determinado bloco está presente somente em uma das caches privadas;
    - Porém ele é somente leitura;

Protocolo MESI

- Quando um bloco está no estado Exclusivo ele pode ser escrito sem invalidar nenhum outro dado;
    - Quando escrito vai para o estado modificado;

Protocolo MESI



Protocolo MOESI

- O protocolo MOESI tem um quinto estado indicando que o bloco é possuído pela cache que se encontra;
  - Modified (Modificado);
  - Owned (Possuido)
  - Exclusive (Exclusivo - não modificado);
  - Shared (Compartilhado);
  - Invalid (Invalido);

Protocolo MOESI

- O estágio “Owned” indica que o bloco está correto somente na cache privada e desatualizado na memória compartilhada;
- No protocolo MOESI, quando um bloco Modificado compartilha seus dados ele passa para o estado “Owned”;
  - Não é feita a escrita em memória;
- A cache que possui o bloco “Owned” é a responsável por atender qualquer miss na cache;
- Só é escrito na memória principal quando é substituído;
- O protocolo MOESI é implementado pelo AMD Opteron;

Protocolo MESIF

- O protocolo MESIF usa um estado para indicar qual a cópia compartilhada responsável por atender miss que ocorrem nas caches;
  - Modified (Modificado);
  - Exclusive (Exclusivo - não modificado);
  - Shared (Compartilhado);
  - Invalid (Invalido);
  - Forward;

Protocolo MESIF

- O bloco que está no estado “Forward” deve avisar quando escreve;
  - As outras cópias devem ser invalidadas;
- Só existe um bloco no estado “Forward”;
- Quando for necessário que o bloco no estado “Forward” volte a memória pode-se optar por:
  - Ficar sem um bloco Forward;
    - O bloco deverá ser buscado na compartilhada;
  - Escolher um outro bloco para assumir o papel de Forward;
    - Mais eficiente, porém mais complexo;

Referências

HENNESSY, John L.;PATTERSON, David A. Computer architecture: a quantitative approach. Elsevier, 2011.

Wentzlaff,D. Computer Architecture ELE 475 / COS 475 Slide Deck 13: Parallel programming and Small Multiprocessors