

Arquitetura de Computadores II

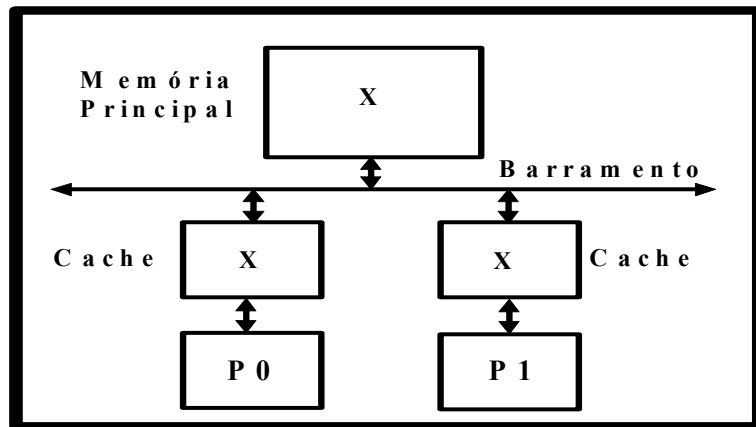
Protocolos de Coerência de Cache

Gabriel P. Silva

Introdução

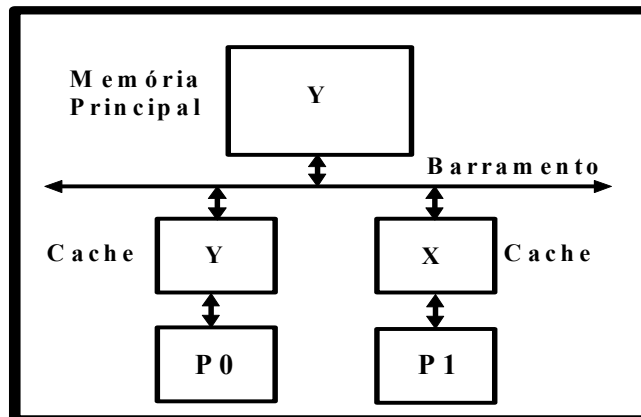
- ◆ **As memórias caches possuem dois modos básicos para trabalhar em relação à atualização dos dados na memória principal durante uma escrita:**
 - **Write-through** → Os dados são atualizados tanto na memória cache como na memória principal.
 - **Write-back** → Os dados são atualizados apenas na memória cache, e copiados para a memória principal, apenas quando da substituição do bloco/linha modificado na cache.

Introdução



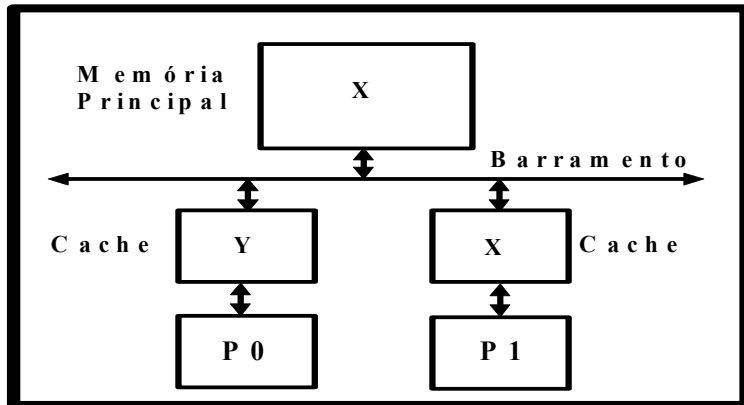
(a) Situação Inicial

Coerência de Cache



(b) Escrita no Modo "Write-Through"

Coerência de Cache

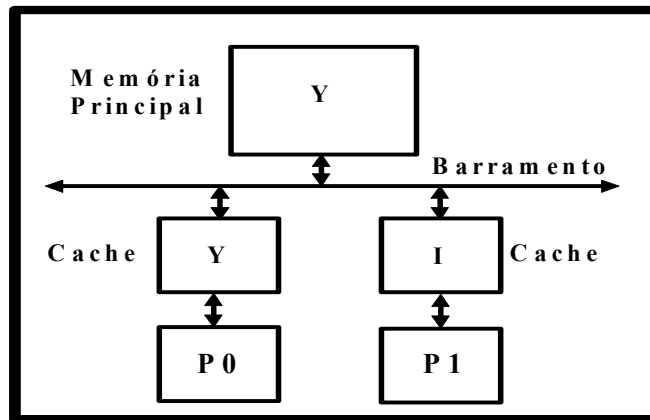


(c) Escrita no Modo "Write-Back"

Coerência de Cache

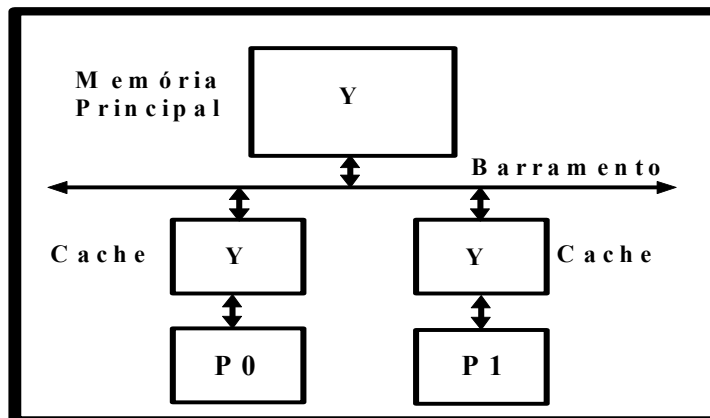
- ◆ A técnica de "*snooping*" para manutenção da coerência das memórias cache se baseia na existência de circuitos em cada nó de processamento que monitoram permanentemente o barramento de modo a invalidarem ou a atualizarem a cópia na memória cache local de um dado que esteja sendo alterado através de uma operação de escrita no barramento.
- ◆ Esta premissa é válida quando os caches operam no modo "*write through*".

Coerência das Memórias Cache Mantida pela Técnica de "Snooping"



(a) Escrita com Invalidação

Coerência das Memórias Cache Mantida pela Técnica de "Snooping"



(b) Escrita com Atualização

Coerência das Memórias Cache Mantida pela Técnica de "Snooping"

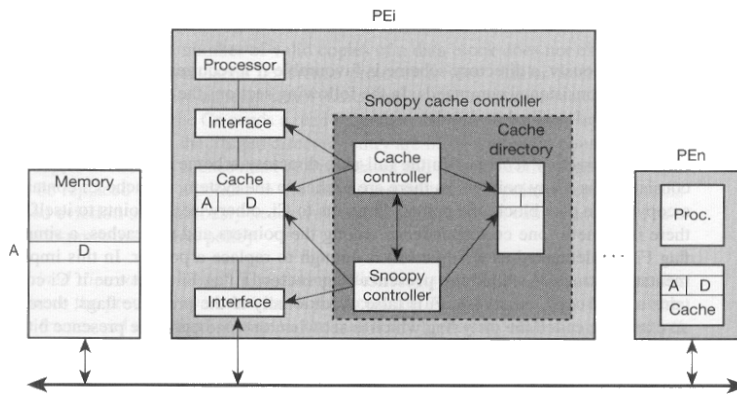


Figure 18.26 Structure of the snoopy cache controller.

Coerência de Cache

- ◆ Duas políticas podem ser adotadas em caso de falha no cache em operações de escrita:
 - **"write allocate"** → o bloco onde vai ser feita a operação de escrita é trazido primeiramente para a memória cache e a operação de escrita é então realizada na memória cache apenas, no modo *write back*, e também na memória principal, no modo *write through*.
 - **"no write allocate"** → o bloco a ser escrito não é trazido para a memória cache e, portanto, a operação de escrita sempre se realiza apenas na memória principal.

Coerência de Cache

- ◆ Para o uso da técnica de "*snooping*" com o esquema "*write back*", há a necessidade de se adotar protocolos mais complexos para a manutenção da coerência dos caches.
- ◆ Estes protocolos visam forçar a colocação no barramento de informações sobre operações de escrita sempre que necessário e a definição sobre quem deve fornecer um bloco de dados quando ocorre uma operação de leitura com falhas, já que no modo "*write back*", a memória principal nem sempre possui uma cópia válida do bloco que está sendo solicitado.

Coerência de Cache

- ◆ Caso uma das caches possua a informação atualizada ("*owner*"), enviará o dado solicitado, bloqueando a resposta da memória principal com a ativação de um sinal no barramento, já que o dado na memória pode estar desatualizado.
- ◆ Usualmente a política "*write allocate*" é adotada com o modo "*write back*" e a política "*no write allocate*" com o modo *write through*.
- ◆ O esquema mais utilizado baseia-se numa combinação de "*write back*" com "*write invalidate*" e com a política "*write allocate*".

Protocolo *MSI*

◆ Nesse protocolo, cada bloco na memória cache pode estar em um de 3 estados:

- **Inválido (I):** bloco inválido na memória cache
- **Shared (S) ou Compartilhado:** bloco só foi lido e pode haver cópias em outras memórias cache
- **Modificado (M):** apenas uma memória cache possui cópia do bloco e a memória principal **não** está atualizada

◆ **Pressuposto:** política de "write-allocate"

Protocolo *MSI*

Estado Atual	Tipo de Acesso	Próx. Estado	Ações no Barramento
Inválido (I)	Leitura Processador	Shared (S)	Leitura c/ falha
Shared (S)	Leitura Processador	Shared (S)	---
Shared (S)	Escrita Processador	Modificado(M)	Invalidação
Shared (S)	Leit. c/falha no barramento	Shared (S)	---
Shared (S)	Invalidação no Barramento	Inválido (I)	---
Modificado (M)	Leitura/Escr. Processador	Modificado(M)	---
Modificado (M)	Leitura c/ Falha no Barramento	Shared (S)	Aciona inibe Envia bloco Atualiza mem.

Protocolo *MESI*

- ◆ Nesse protocolo, cada bloco na memória cache pode estar em um de 4 estados:
 - **Inválido (I):** bloco inválido na memória cache
 - **Shared (S) ou Compartilhado:** bloco só foi lido e pode haver cópias em outras memórias cache
 - **Modificado (M):** apenas uma memória cache possui cópia do bloco e a memória principal **não** está atualizada
 - **Exclusivo (E):** apenas uma memória cache possui cópia do bloco e a memória principal está atualizada.
- ◆ Pressuposto: política de "write-allocate"

Estado Atual	Tipo de Acesso	Próx. Estado	Ações no Barramento
Inválido (I)	Leit. process. <i>(shared desativado no barramento)</i>	Exclusivo (E)	Leitura c/ falha
Inválido (I)	Leit. process. <i>(shared ativado no barramento)</i>	Shared (S)	Leitura c/ falha
Shared (S)	Leit. process.	Shared (S)	---
Shared (S)	escr. Process.	Modificado (M)	Invalidação
Shared (S)	Invalidação no barramento	Inválido (I)	---
Shared (S)	Leitura com falha no barramento	Shared (S)	Aciona <i>shared</i>
Exclusivo (E)	Leit. process.	Exclusivo (E)	---
Exclusivo (E)	Escr. process.	Modificado (M)	---
Exclusivo (E)	Leitura c/ falha no barramento	Shared (S)	Aciona <i>shared</i>
Modificado (M)	Leitura process.	Modificado (M)	---
Modificado (M)	Leitura c/ falha no barramento	Shared (S)	Aciona inibe Aciona <i>shared</i> Envia bloco Atualiza mem.

Protocolo *MOESI*

◆ Nesse protocolo, cada bloco na memória cache pode estar em um de 5 estados:

- **Inválido (I):** bloco inválido na memória cache.
- **Shared (S) ou Compartilhado:** bloco só foi lido e pode haver cópias em outras memórias cache.
- **Modificado (M):** apenas esse cache possui cópia do bloco e a memória principal **não** está atualizada
- **Exclusivo (E):** Apenas esse cache possui cópia do bloco e a memória principal está atualizada.
- **Owner (O) :** Esse cache supre o dado em caso de leitura com falha no barramento uma vez que a memória **não** está atualizada. Existem cópias do bloco em outras caches.

Protocolo *MOESI*

Estado Atual	Tipo de Acesso	Próx. Estado	Ações no Barramento
Inválido (I)	leit. process. (<i>shared</i> desat. no barramento)	Exclusivo (E)	leitura c/ falha
Inválido (I)	leit. process. (<i>shared</i> ativado)	Shared (S)	leitura c/ falha
Shared (S)	leit. process.	Shared (S)	---
Shared (S)	escr. process.	Modificado (M)	invalidação
Shared (S)	invalidação no barramento	Inválido (I)	---
Shared (S)	leitura c/ falha no barramento	Shared (S)	aciona <i>shared</i>
Exclusivo (E)	leit. process.	Exclusivo (E)	---
Exclusivo (E)	escr. process.	Modificado (M)	---
Exclusivo (E)	leitura c/ falha no barramento	Shared (S)	aciona <i>shared</i>

Protocolo *MOESI*

Estado Atual	Tipo de Acesso	Próx. Estado	Ações no Barramento
Owner (O)	invalidação no barramento	Inválido (I)	---
Owner (O)	leitura c/ falha no barramento	Owner (O)	envia bloco aciona inibe aciona shared
Owner (O)	leit. process.	Owner (O)	
Owner (O)	escr. process.	Modificado (M)	invalidação
Modificado (M)	leitura process.	Modificado (M)	---
Modificado (M)	leitura c/ falha no barramento	Owner (O)	<i>envia bloco</i> aciona inibe aciona <i>shared</i>

Protocolos de Coerência de Cache

- ◆ Nos três protocolos quando um bloco da memória cache no estado *Modificado* é selecionado para ser substituído para dar lugar a um bloco novo que está sendo lido da memória, o bloco antes de ser descartado deve ser atualizado na memória.
- ◆ No protocolo *MOESI*, este procedimento também deve ser adotado quando um bloco no estado *Owner* é selecionado para ser substituído.