

Arquiteturas de Computadores

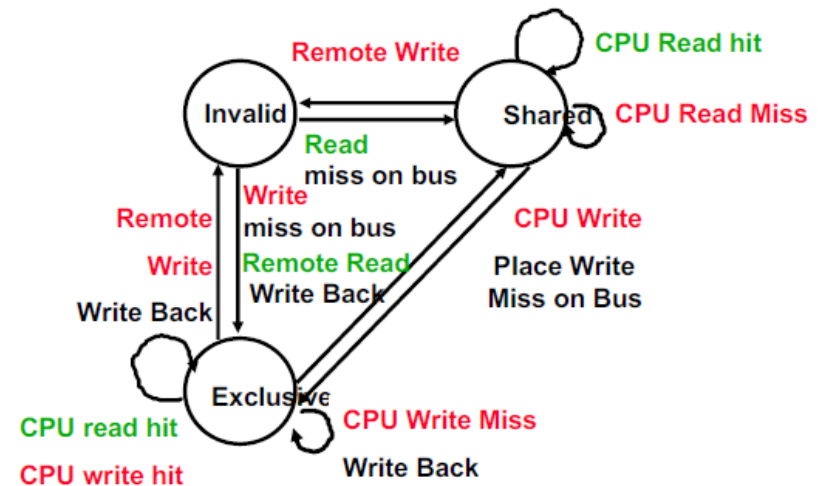
Paralelismo no nível de threads

Fontes dos slides: Livro Patterson e Hennessy, Quantitative Approach e nas notas de aula do Prof. Yonghong Yan do curso CSCE 513 Computer Architecture

Exemplo

	<i>P1</i>			<i>P2</i>			<i>Bus</i>			<i>Memory</i>		
<i>step</i>	<i>State</i>	<i>Addr</i>	<i>Value</i>	<i>State</i>	<i>Addr</i>	<i>Value</i>	<i>Action</i>	<i>Proc.</i>	<i>Addr</i>	<i>Value</i>	<i>Addr</i>	<i>Value</i>
P1: Write 10 to A1												
P1: Read A1												
P2: Read A1												
P2: Write 20 to A1												
P2: Write 40 to A2												

Assuma estado inicial da cache é inválido e A1 e A2 mapeiam para o mesmo bloco de cache, mas $A1 \neq A2$

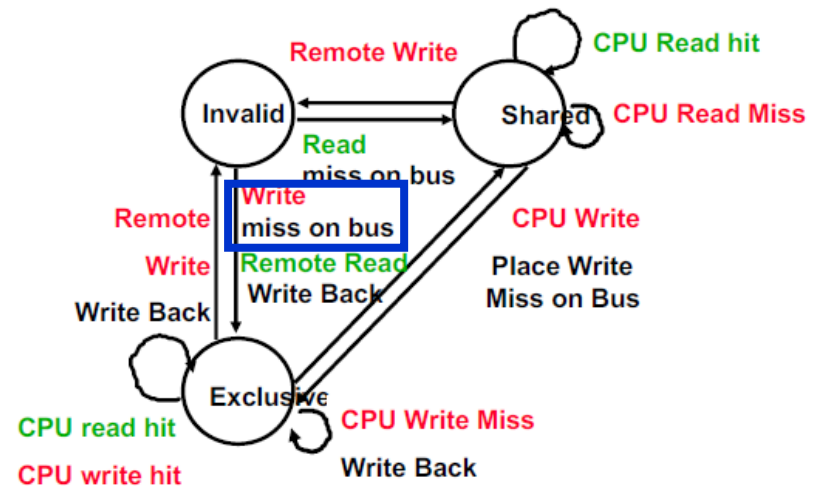


Exemplo: Passo 1

	<i>P1</i>			<i>P2</i>			<i>Bus</i>			<i>Memory</i>		
<i>step</i>	<i>State</i>	<i>Addr</i>	<i>Value</i>	<i>State</i>	<i>Addr</i>	<i>Value</i>	<i>Action</i>	<i>Proc.</i>	<i>Addr</i>	<i>Value</i>	<i>Addr</i>	<i>Value</i>
P1: Write 10 to A1	<u>Excl.</u>	<u>A1</u>	<u>10</u>				<u>WrMs</u>	P1	A1			
P1: Read A1												
P2: Read A1												
P2: Write 20 to A1												
P2: Write 40 to A2												

Assuma estado inicial da cache é inválido e A1 e A2 mapeiam para o mesmo bloco de cache, mas $A1 \neq A2$

Transição ativa :

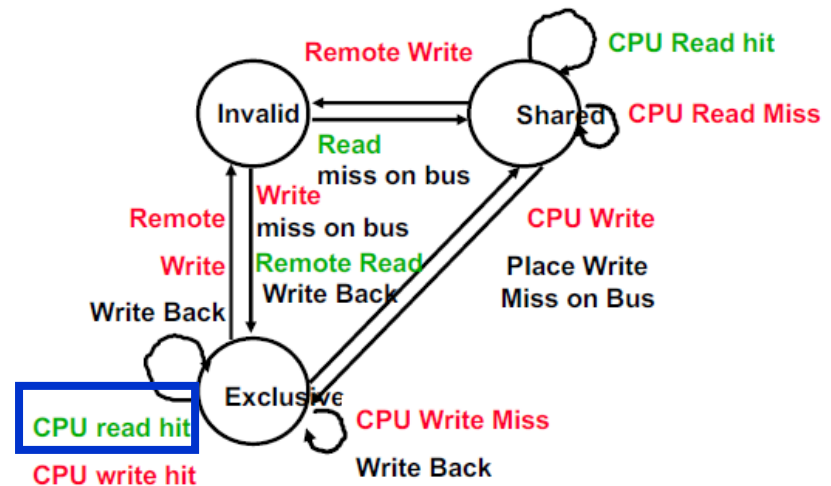


Exemplo: Passo 2

	<i>P1</i>			<i>P2</i>			<i>Bus</i>				<i>Memory</i>	
<i>step</i>	<i>State</i>	<i>Addr</i>	<i>Value</i>	<i>State</i>	<i>Addr</i>	<i>Value</i>	<i>Action</i>	<i>Proc.</i>	<i>Addr</i>	<i>Value</i>	<i>Addr</i>	<i>Value</i>
P1: Write 10 to A1	<u>Excl.</u>	<u>A1</u>	<u>10</u>				<u>WrMs</u>	P1	A1			
P1: Read A1	Excl.	A1	10									
P2: Read A1												
P2: Write 20 to A1												
P2: Write 40 to A2												

Assuma estado inicial da cache é inválido e A1 e A2 mapeiam para o mesmo bloco de cache, mas A1 != A2

Transição ativa :

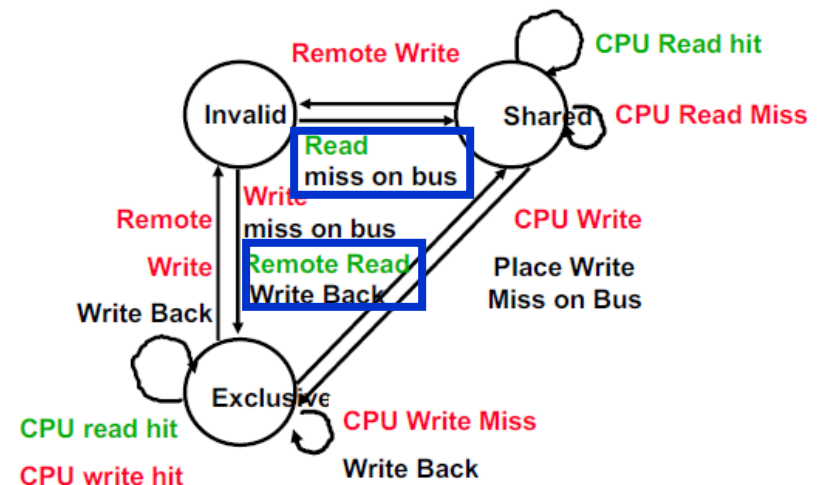


Exemplo: Passo 3

	<i>P1</i>			<i>P2</i>			<i>Bus</i>			<i>Memory</i>		
<i>step</i>	<i>State</i>	<i>Addr</i>	<i>Value</i>	<i>State</i>	<i>Addr</i>	<i>Value</i>	<i>Action</i>	<i>Proc.</i>	<i>Addr</i>	<i>Value</i>	<i>Addr</i>	<i>Value</i>
P1: Write 10 to A1	<i>Excl.</i>	<i>A1</i>	<i>10</i>				<i>WrMs</i>	P1	A1			
P1: Read A1	Excl.	A1	10									
P2: Read A1				<i>Shar.</i>	<i>A1</i>		<i>RdMs</i>	P2	A1			
	<i>Shar.</i>	A1	10				<i>WrBk</i>	P1	A1	10	<i>A1</i>	<i>10</i>
				Shar.	A1	<i>10</i>	<i>RdDa</i>	P2	A1	10	<i>A1</i>	10
P2: Write 20 to A1												
P2: Write 40 to A2												

Assuma estado inicial da cache é inválido e A1 e A2 mapeiam para o mesmo bloco de cache, mas $A1 \neq A2$

Transição ativa :

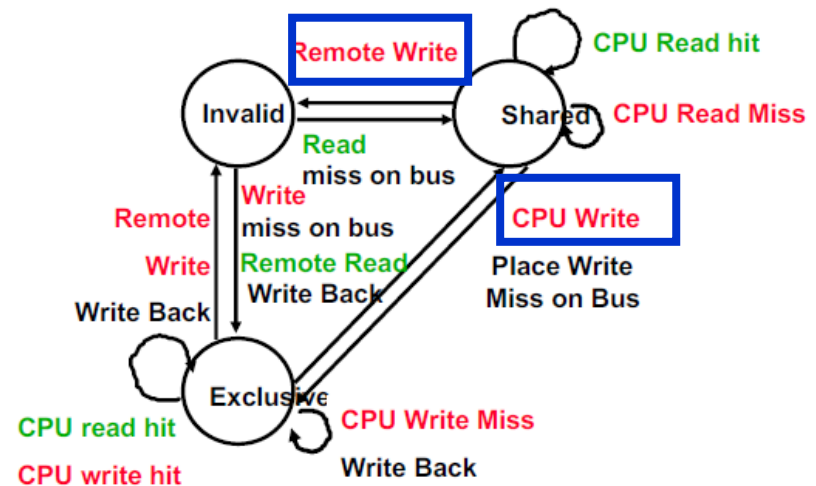


Exemplo: Passo 4

	<i>P1</i>			<i>P2</i>			<i>Bus</i>				<i>Memory</i>	
<i>step</i>	<i>State</i>	<i>Addr</i>	<i>Value</i>	<i>State</i>	<i>Addr</i>	<i>Value</i>	<i>Action</i>	<i>Proc.</i>	<i>Addr</i>	<i>Value</i>	<i>Addr</i>	<i>Value</i>
P1: Write 10 to A1	<u>Excl.</u>	<u>A1</u>	<u>10</u>				<u>WrMs</u>	P1	A1			
P1: Read A1	Excl.	A1	10									
P2: Read A1				<u>Shar.</u>	<u>A1</u>		<u>RdMs</u>	P2	A1			
	<u>Shar.</u>	A1	10				<u>WrBk</u>	P1	A1	10	A1	<u>10</u>
				Shar.	A1	<u>10</u>	<u>RdDa</u>	P2	A1	10	A1	10
P2: Write 20 to A1	<u>Inv.</u>			<u>Excl.</u>	A1	<u>20</u>	<u>Inv</u>	P2	A1		A1	10
P2: Write 40 to A2												-

Assuma estado inicial da cache é inválido e A1 e A2 mapeiam para o mesmo bloco de cache, mas $A1 \neq A2$

Transição ativa :

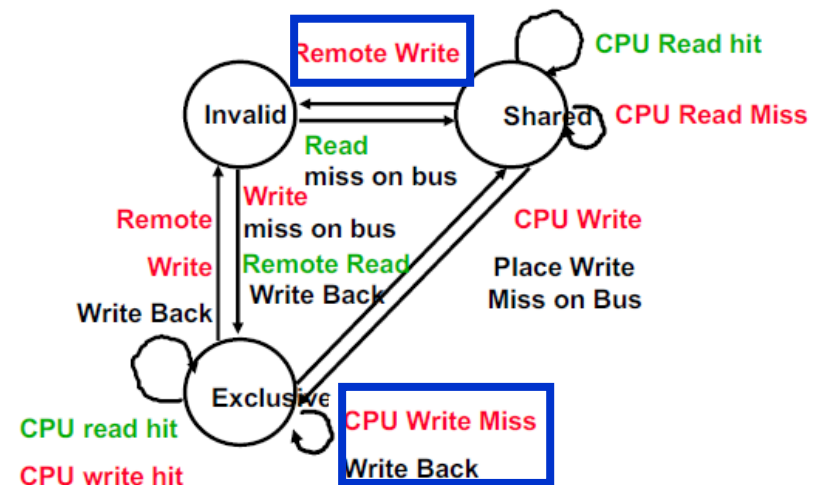


Exemplo: Passo 5

	P1			P2			Bus			Memory		
step	State	Addr	Value	State	Addr	Value	Action	Proc.	Addr	Value	Addr	Value
P1: Write 10 to A1	<u>Excl.</u>	<u>A1</u>	<u>10</u>				<u>WrMs</u>	P1	A1			
P1: Read A1	Excl.	A1	10									
P2: Read A1				<u>Shar.</u>	<u>A1</u>		<u>RdMs</u>	P2	A1			
	<u>Shar.</u>	A1	10				<u>WrBk</u>	P1	A1	10	<u>A1</u>	<u>10</u>
				Shar.	A1	<u>10</u>	<u>RdDa</u>	P2	A1	10	A1	10
P2: Write 20 to A1	<u>Inv.</u>			<u>Excl.</u>	A1	<u>20</u>	<u>Inv</u>	P2	A1		A1	10
P2: Write 40 to A2							<u>WrMs</u>	P2	A2		A1	10
				Excl.	<u>A2</u>	<u>40</u>	<u>WrBk</u>	P2	A1	20	<u>A1</u>	<u>20</u>

Assuma estado inicial da cache é inválido e A1 e A2 mapeiam para o mesmo bloco de cache, mas A1 != A2

Transição ativa :



Categorias de falta na memória

- Em um processador:
 - Faltas compulsórias: o primeiro acesso sempre ocasiona falta porque os blocos não estão na cache
 - Falta por capacidade: a cache não tem capacidade de armazenamento para conter todos os blocos necessários para a execução
 - Faltas por conflito: o bloco de cache deve ser descartado devido à estratégia de substituição de blocos
- Em sistemas com vários processadores:
 - Faltas de coerência: o bloco de cache deve ser descartado porque outro processador modificou o conteúdo
 - Verdadeira falta de compartilhamento: outro processador modificou o conteúdo do elemento de solicitação
 - Falta de compartilhamento falso: outro processador invalidou o bloco, embora o item de interesse real não seja alterado

Categorias de falta na memória

state	line addr	data0	data1	...	dataN
-------	-----------	-------	-------	-----	-------

- Uma linha da cache contém mais de uma palavra
 - Coerência de cache é realizada no nível de linha e não de palavra
 - Suponha M1 escreve na palavra 0 e M2 escreve na palavra 1
 - ambas tem o mesmo endereço de linha
 - O que pode acontecer?