

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA
DE MINAS GERAIS

LABORATÓRIO DE ARQUITETURA E ORGANIZAÇÃO DE
COMPUTADORES 2

Prática 04 – Implementação do Snooping MSI

Renan Siman Claudino
Rubio Torres Castro Viana

Orientado por
Prof. Ms. POLIANA APARECIDA CORRÊA DE OLIVEIRA

Belo Horizonte - MG
2018

Conteúdo

1	Introdução	2
2	Detalhes do projeto	2
2.1	Decisões de implementação	2
2.2	Módulos	2
3	Funcionamento do processador	3
4	Funcionamento do emissor	3
5	Funcionamento do receptor	3
6	Testes	3
7	Dificuldades	4

1 Introdução

Problemas de coerência sempre existiram em sistemas com mais de um processador.[1]. O protocolo snooping msi nos permite resolver esse problema de coêrencia fazendo a manutenção das alterações de dados em cada um dos processadores. Foi aplicado neste projeto a máquina de estados do snooping msi que contém três estados: modificado, compartilhado e inválido.

2 Detalhes do projeto

2.1 Decisões de implementação

Para este projeto optamos por usar as variáveis:

Alterna	receptor = 1 , emissor = 0
Estado	Exclusive = 10 , Shared = 01 , Invalid = 00
Operação	Read = 0 e Write = 1
Acerto	Miss = 0 e Hit = 1
Bus	Read Miss = 01, Write Miss = 10, Invalidate = 11
Estado	Estado resultante
WB	Ativa o write back
ocorrencias	[1]Opcode, [0]acerto
Cache	[5:4]estado:E(10), S(01), I(00),[3]Tag, [2:0]Dado
AbortaAcessoMemoria	Aborta o acesso a memória: Hit = 1 e Miss = 0
Receptor estado saida	Determina o estado saida do receptor
Barramento(bar)	[10]alternacpu,[9:8]escolhe,[7]wb,[6]TAGwb, [5:4]Mensagem, [3]Tag, [2:0]Dado
Inst	Instrução:[6:5]escolhe,[4]operacao,[3]Tag,[2:0]Dado
clk	Clock

2.2 Módulos

Foram implementados 7 módulos principais, parte deles para o funcionamento do snoop, parte para as simulações e parte para a apresentação em sala:

- Snoop: É responsável pela apresentação na placa e definição das variáveis.
- Snoopinstance: Instancia 3 processadores, uma memoria e um barramento comum.
- Bus: Barramento comum.
- Mem: Memoria inicializada.
- processador: É responsável pela execução da instrução.
- Emissor: Máquina de estados que emite a mensagem ao sistema. É mudado diretamente de acordo com a instrução dada.

- Receptor: Máquina de estados que recebe a mensagem do sistema. É mudado de acordo com a mensagem dada pelo emissor.

3 Funcionamento do processador

Verifica o writeback, executa as instruções, inicializa e executa as maquinas de estado.

4 Funcionamento do emissor

Verificamos o estado atual do bloco que está sendo chamado. Após isso, é verificado a operação e se houve um acerto temos a nossa tabela de estados mostrada nos testes.

5 Funcionamento do receptor

Verificamos o estado atual do bloco que esta "escutando a mensagem". Após isso, é verificado a mensagem que o emissor enviou e assim temos a nossa tabela de estados mostrada nos testes.

6 Testes

Foi realizado testes para verificar o load e o store e os processadores: O primeiro código testamos todos os casos do emissor:

Instrução	cache1	cache2	cache3
1001000	000000	010000	000000
1111011	000000	000000	101011

E foi obtido o seguinte resultado:

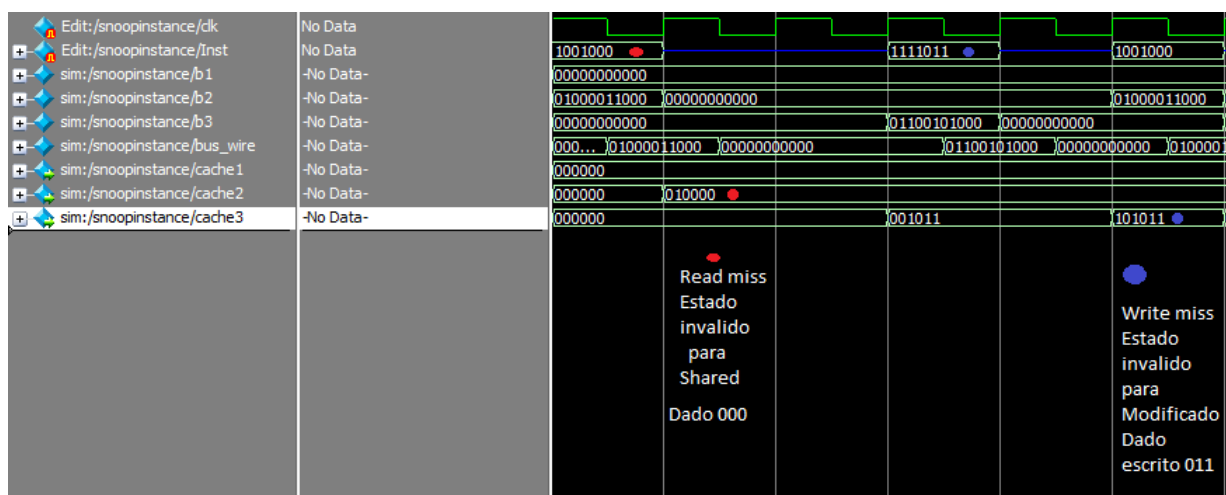


Figura 1: Teste snoop

É possível ver, a cada ciclo, a mudança de estados, a mensagem sendo enviada e o write back sendo ativado nos momentos certos.

7 Dificuldades

A aplicação na placa gerou problemas, além disso a simulação também gerou com a dificuldade de acertar o tempo de término das instruções.

Referências

- [1] John L Hennessy and David A Patterson. *Computer architecture: a quantitative approach*. Elsevier, 2011.