

AOC II

Snooping

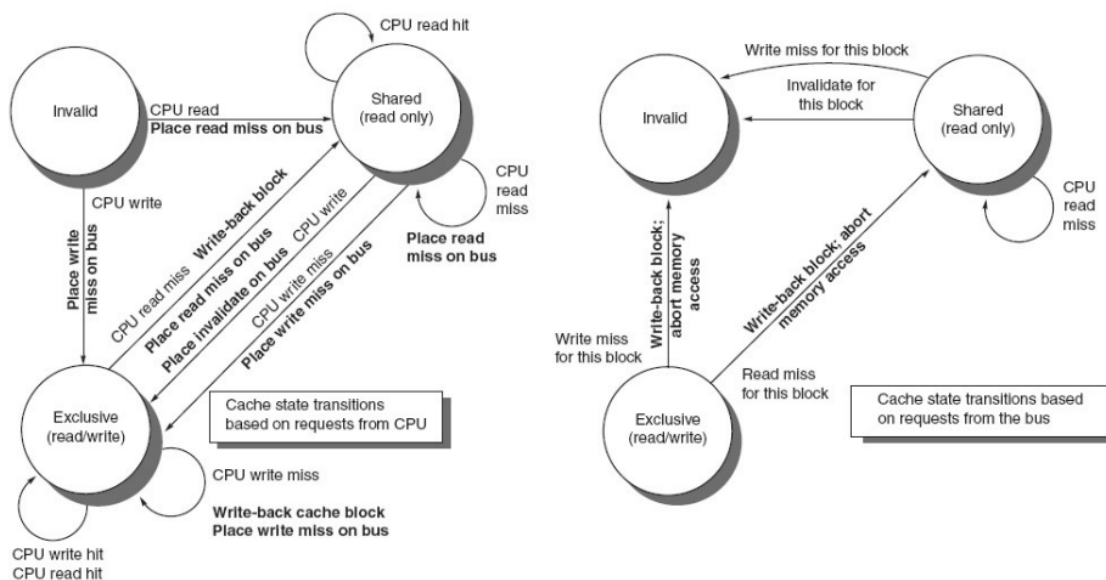
Bruna Gomes Camilo e Lucas Cota Dornelas

9 de setembro de 2021

Introdução

O Snooping consiste em um protocolo, utilizado em máquinas de estados, que tem por intuito realizar a coerência de cache, isto é, manter a consistência dos dados armazenados. Para isso, é implementado um barramento no processo de comunicação. Tal protocolo é usado, de forma recorrente, em multiprocessadores e sistemas de memória compartilhada distribuída. Dessa forma, ao se usar o protocolo de Snooping podemos garantir a persistência e integridade dos dados.

O Snooping implementado utiliza o Verilog HDL e realiza as simulações de seu funcionamento. Com esse intuito, os softwares Quartus II e ModelSim da Altera foram, respectivamente, utilizados para a construção do código e simulação. Nesse presente relatório, será apresentado o mapeamento do Snooping implementado e os resultados alcançados na simulação.



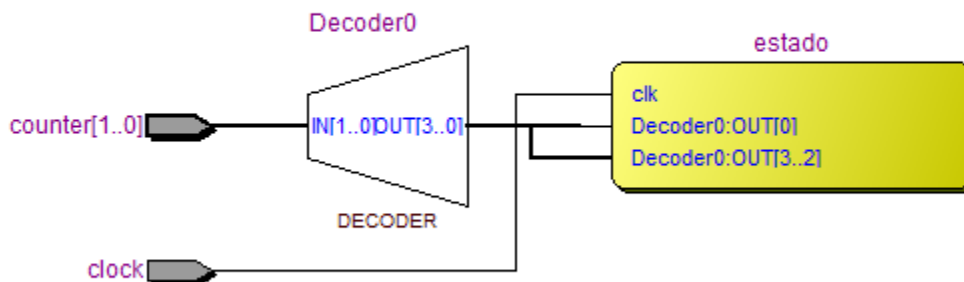
Implementação

A primeira máquina de estado indica a máquina emissora, isto é, a máquina que emitirá os sinais no momento que algum dado for modificado na memória. Já a segunda máquina representa a máquina receptora, ou seja, a máquina que tem por função receber os sinais

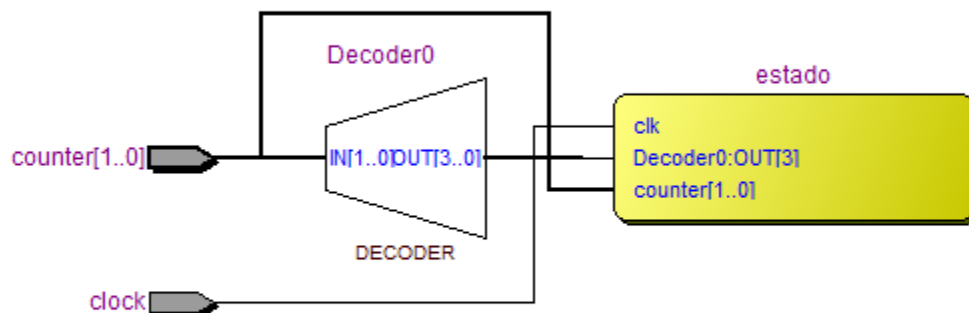
enviados pela emissora e, com eles, realizar as operações necessárias com o intuito de garantir a consistência de dados.

Em cada uma das máquinas foram implementados três estados, sendo eles o invalid, exclusive e shared. O estado invalidado corresponde aos casos nos quais algum dado não pode ser lido, isto é, quando algum dado inválido está salvo na posição de memória. O estado shared corresponde aos casos onde os dados podem ser compartilhados entre caches, seja uma ou mais. Por fim, o estado exclusivo que designa os casos em que um valor em posição de memória é exclusivo da memória cache que a detém.

- **Máquina Emissora**



- **Máquina Receptora**



Simulação

A simulação foi pautada na verificação das transições das máquinas de estados emissora e receptora. Para isso, foi utilizado os seguintes parâmetros de teste

- **Parâmetros da máquina emissora**

```
module partelmaql(clock, counter);  
    input clock;  
    input [1:0] counter;  
  
    // Possiveis Operação  
    parameter readMiss = 2'b00;  
    parameter readHit = 2'b01;  
    parameter writeMiss = 2'b10;  
    parameter writeHit = 2'b11;  
  
    // Estados do Processador  
    parameter invalid = 2'b00;  
    parameter exclusive = 2'b01;  
    parameter shared = 2'b10;  
  
    // Mensagens do Processador  
    parameter emptyMessage = 2'b00;  
    parameter placeReadMissOnBus = 2'b01;  
    parameter placeInvalidateOnBus = 2'b10;  
    parameter placeWriteMissOnBus = 2'b11;  
  
    // Ações do Processador  
    parameter emptyAction = 2'b00;  
    parameter writeBackBlock = 2'b01;  
    parameter writeBackCacheBlock = 2'b10;
```

- **Parâmetros máquina receptora**

```

module partelmaq2 (clock, counter);
    input clock;
    input [1:0] counter;

    //Possíveis operações
    parameter readMiss = 2'b00;
    parameter invalidate = 2'b01;
    parameter writeMiss = 2'b10;

    //Estados do Processador
    parameter invalid = 2'b00;
    parameter exclusive = 2'b01;
    parameter shared = 2'b10;

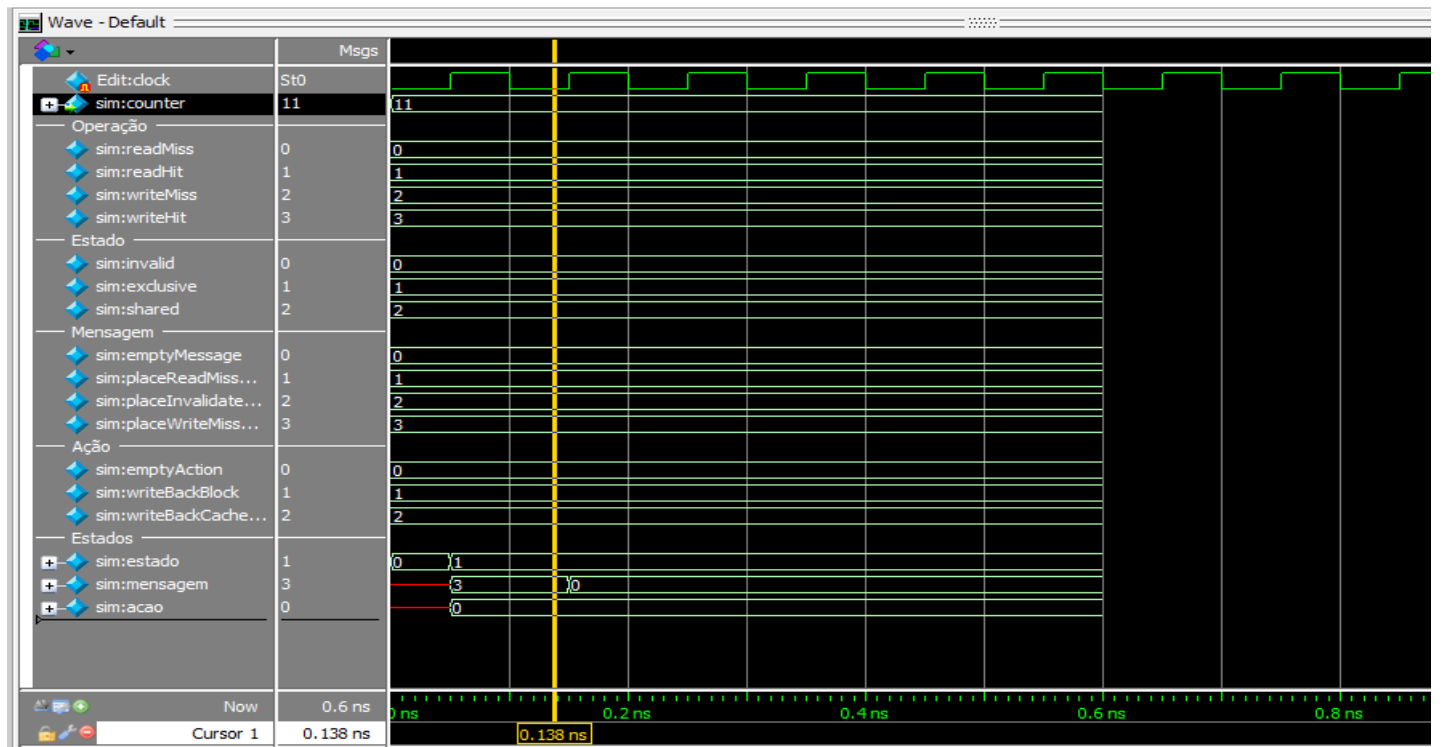
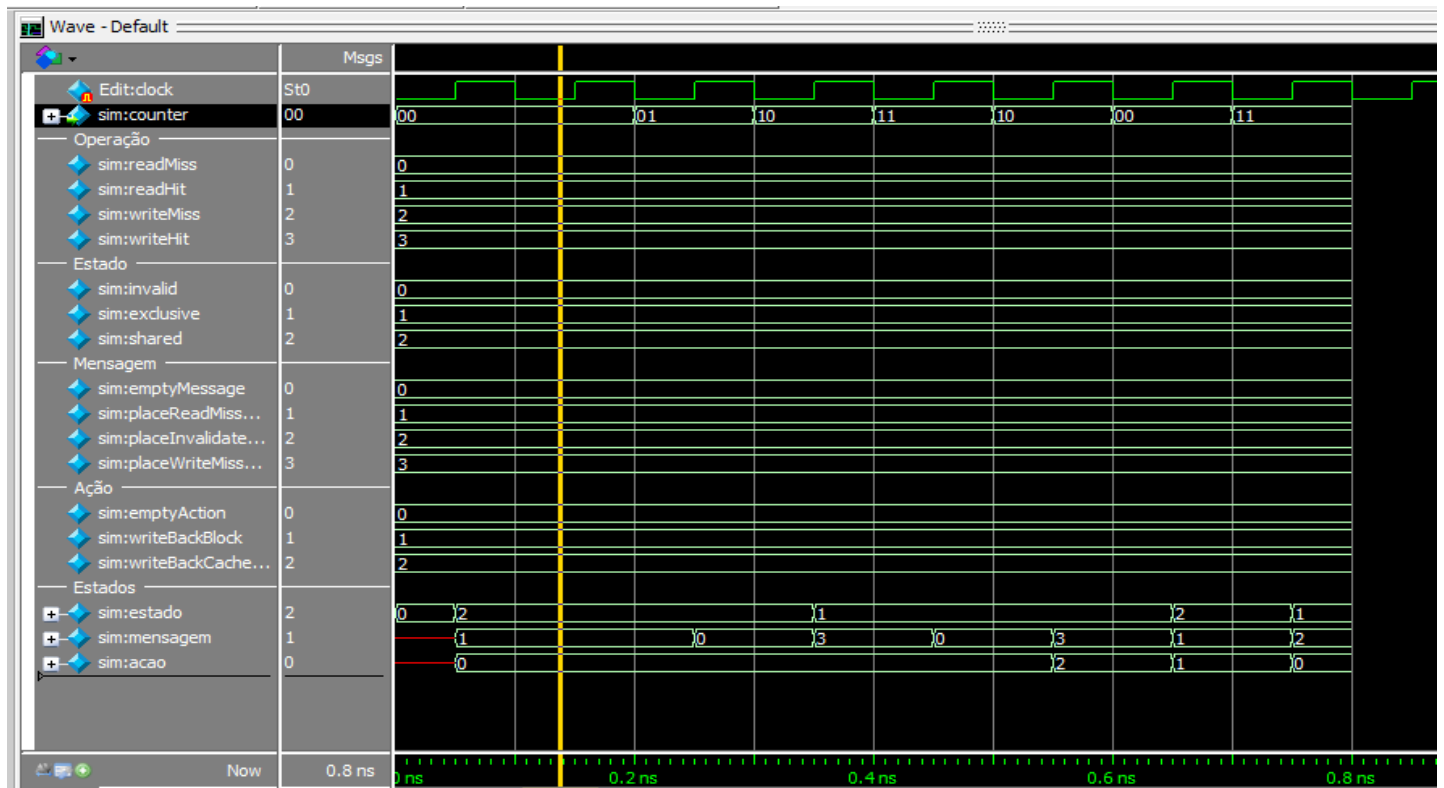
    //Ações do Processador
    parameter empty = 1'b0;
    parameter writeBackBlock = 1'b1;

    reg [1:0] estado;
    reg action;

```

Inicialmente faremos a análise dos movimentos ocorridos em cada estado da máquina emissora.

- **Simulação máquina emissora**



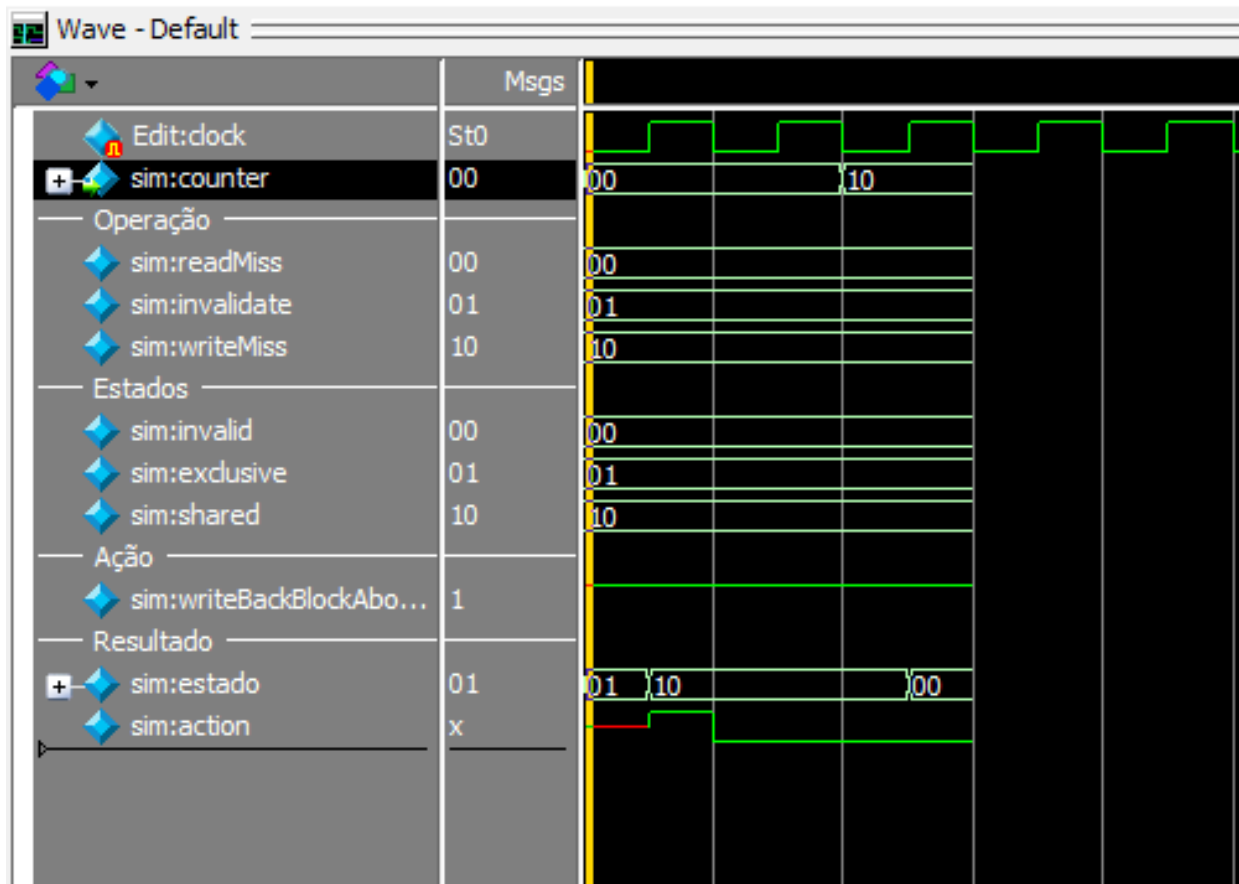
Primeira simulação

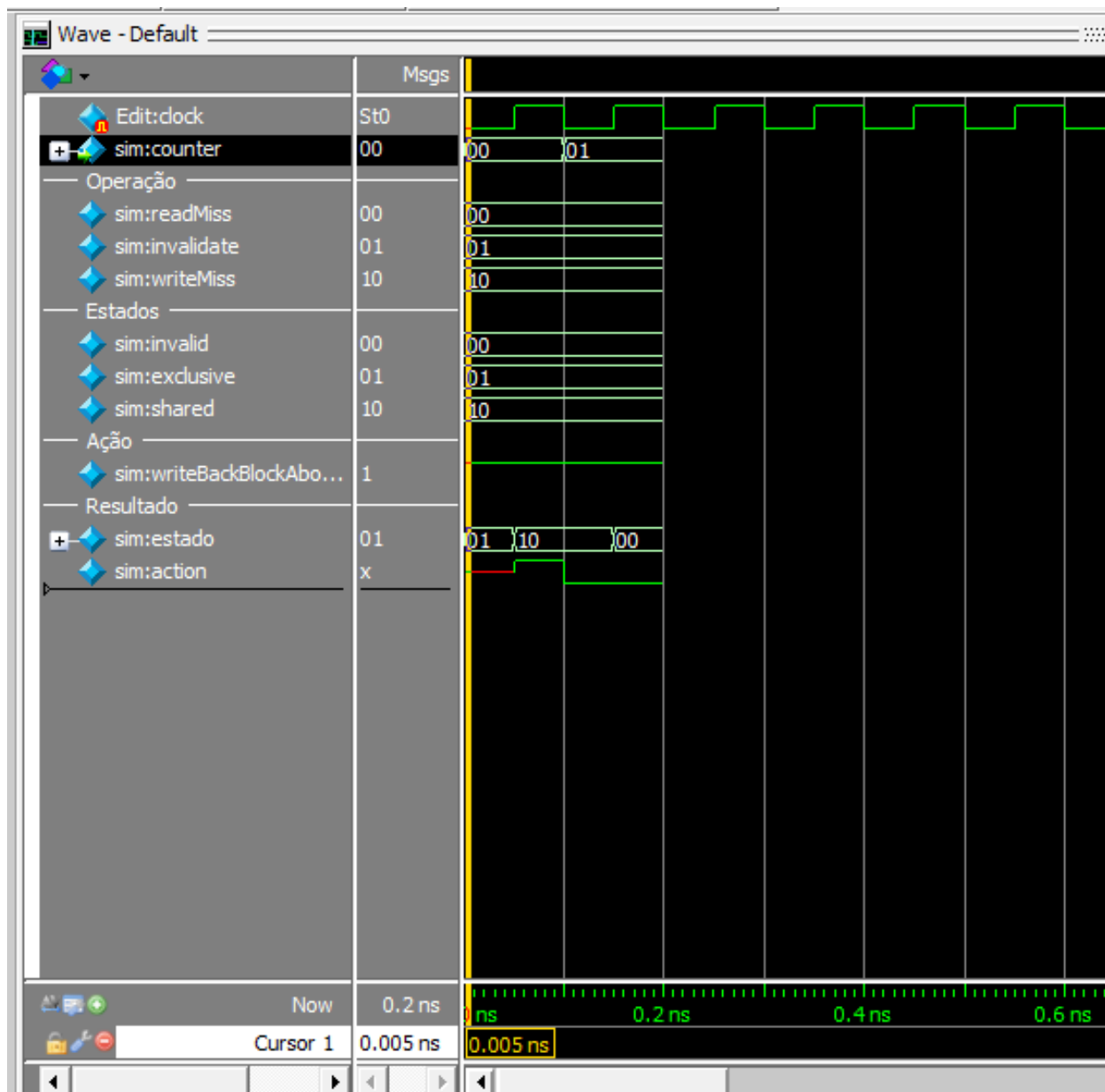
- Em seu estado *invalid*, ao ocorrer uma ação *readMiss*, representado pelo counter *b'00*, é emitido uma mensagem "*place read miss*", representado pelo binário *b'01*, bem como o próximo estado passa a ser o *shared*, representado pelo binário *b'10*.
- Em seu estado *shared*, ao ocorrer uma ação *readMiss*, representado pelo counter *b'00*, não é emitido uma mensagem "*place read miss*", bem como se mantém no estado *shared*, representado pelo binário *b'10*.
- Em seu estado *shared*, ao ocorrer uma ação *readHit*, representado pelo counter *b'01*, não é emitido nenhuma mensagem, representado pelo binário *b'00*, bem como se mantém no estado *shared*, representado pelo binário *b'10*.
- Em seu estado *shared*, ao ocorrer uma ação *writeMiss*, representado pelo counter *b'10*, é emitido uma mensagem "*place write miss*", bem como o próximo estado passa a ser o *exclusive*, representado pelo binário *b'01*.
- Em seu estado *exclusive*, ao ocorrer uma ação *whiteHit*, representado pelo counter *b'11*, ou ao ocorrer uma ação *readHit*, representado pelo counter *b'01*, não é emitido nenhuma mensagem, representado pelo binário *b'00*, bem como se permanece no mesmo estado *exclusive*, representado pelo binário *b'01*.
- Em seu estado *exclusive*, ao ocorrer uma ação *whiteMiss*, representado pelo counter *b'10*, é emitido uma mensagem "*place write miss*", representado pelo binário *b'11*, bem como executado um *write back* e um *cache block*. Por fim, se mantém no mesmo estado *exclusive*, representado pelo binário *b'01*.
- Em seu estado *exclusive*, ao ocorrer uma ação *readMiss*, representado pelo counter *b'00*, não é emitido nenhuma mensagem, representado pelo binário *b'00*, bem como se altera para o estado *shared*, representado pelo binário *b'10*. Além disso, ele emite um *write back* e um *cache block* e passa para o estado *shared*.
- Em seu estado *shared*, ao ocorrer uma ação *writeHit*, representado pelo counter *b'11*, é emitido as seguintes mensagens "*place read miss*" e "*place invalidate*", representado respectivamente pelos binários *b'01* e *b'10*, bem como se altera para o estado *exclusive*, representado pelo binário *b'01*.

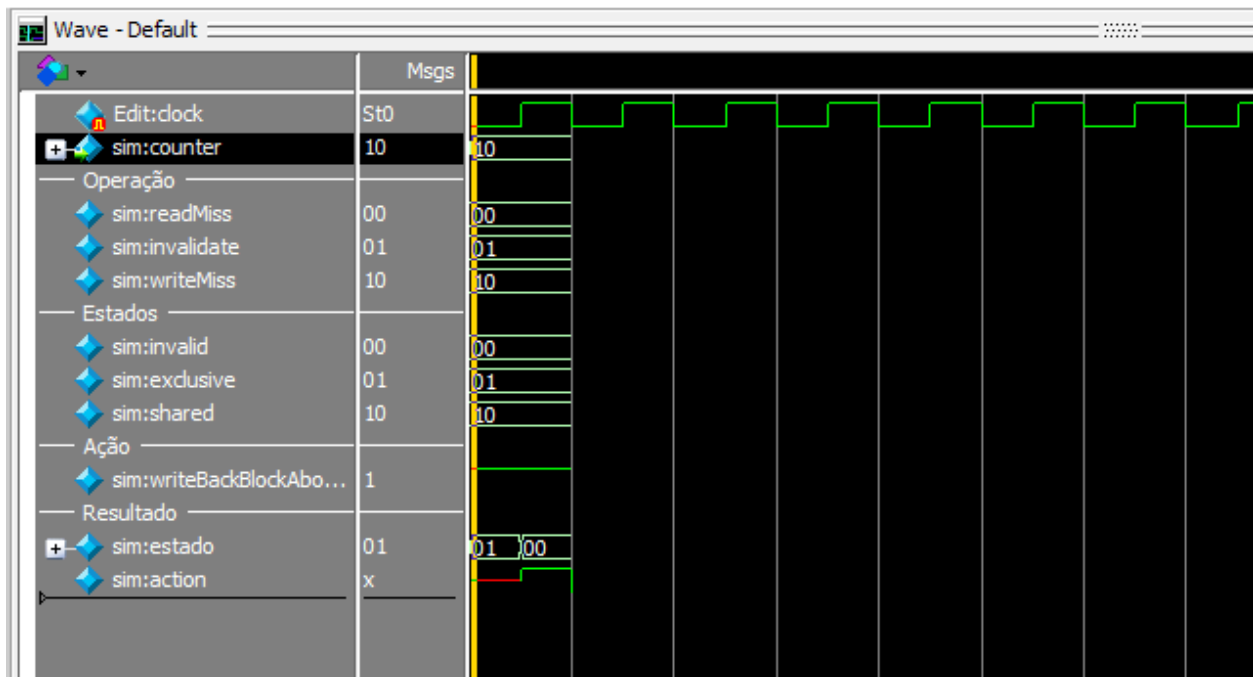
Segunda simulação

- A partir do estado *invalid* um *writeHit*, representado pelo counter *b'11*, é emitido a mensagem "*place write miss*", representado pelo binário *b'11*, bem como o estado passa a ser *exclusive*, representado pelo binário *b'01*.

- Simulação máquina receptora







A seguir faremos a análise dos movimentos ocorridos em cada estado da máquina receptora.

Primeira simulação

- Em seu estado *exclusive*, ao ocorrer uma ação *readMiss*, representado pelo counter *b'00*, bem como o próximo estado passa a ser o *shared*, representado pelo binário *b'10*. Além disso, é disparada uma ação “*write back block: Abort Memory Access*”, representado pelo binário *b'01*.
- Em seu estado *shared*, ao ocorrer uma ação *readMiss*, representado pelo counter *b'00*, bem como o próximo estado passa a ser o *shared*, representado pelo binário *b'10*. Além disso, não é disparada nenhuma ação.
- Em seu estado *shared*, ao ocorrer uma ação *writeMiss*, representado pelo counter *b'10*, bem como o próximo estado passa a ser o *invalid*, representado pelo binário *b'00*. Além disso, não é disparada nenhuma ação.

Segunda simulação

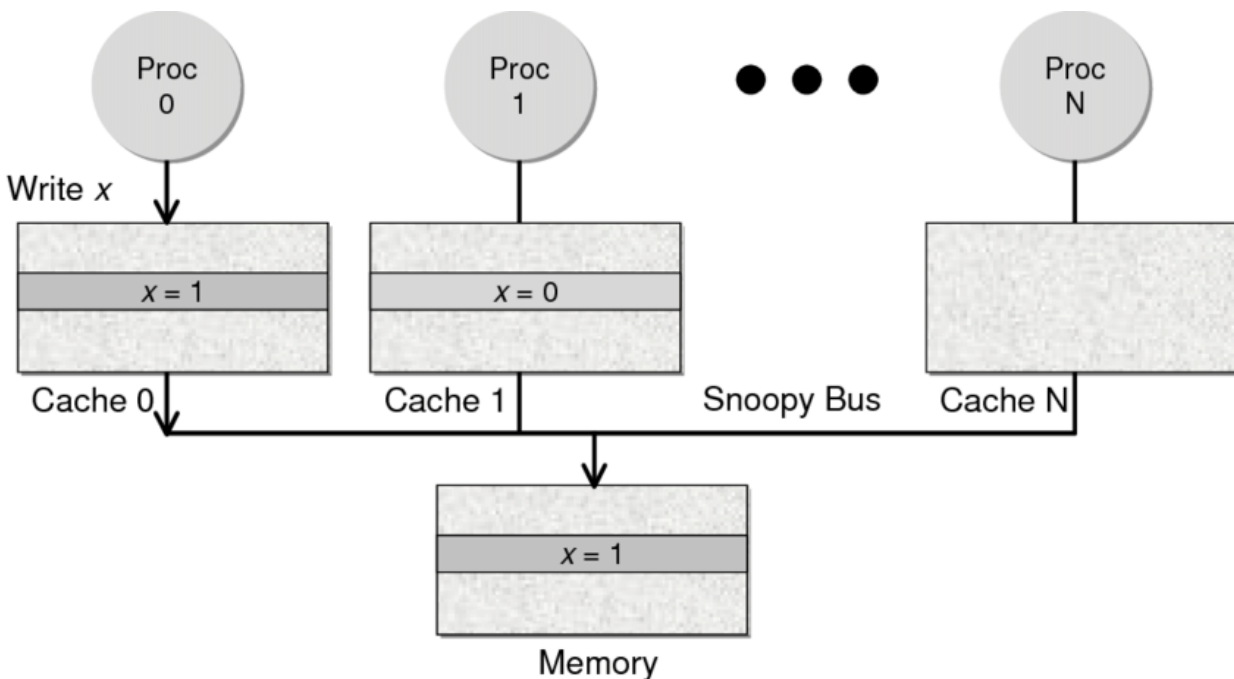
- Em seu estado *exclusive*, ao ocorrer uma ação *readMiss*, representado pelo counter *b'00*, bem como o próximo estado passa a ser o *shared*, representado pelo binário *b'10*. Além disso, é disparada uma ação “*write back block: Abort Memory Access*”, representado pelo binário *b'01*.

- Em seu estado *shared*, ao ocorrer uma ação *invalidate*, representado pelo counter *b'01*, bem como o próximo estado passa a ser o *invalid*, representado pelo binário *b'00*. Além disso, não é disparada nenhuma ação.

Terceira simulação

- Em seu estado *exclusive*, ao ocorrer uma ação *writeMiss*, representado pelo counter *b'10*, bem como o próximo estado passa a ser o *invalidate*, representado pelo binário *b'00*. Além disso, é disparada uma ação “*write back block: Abort Memory Access*”, representado pelo binário *b'01*.

Parte 2



Com o desenvolvimento dos microprocessadores o uso de vários núcleos se tornou comum e com isso surgiu a necessidade de melhorar o sistema de coerência entre as caches. Nessa prática desenvolvemos um processador de três núcleos que implementa protocolo Snooping. O processador acompanha três caches L1 de dois blocos de 8 bits e uma memória principal de 8 blocos.

O módulo principal implementa o processador de três núcleos que utiliza o protocolo Snooping nas caches L1. Assim, possui três instâncias do módulo *cache* e cada uma dessas

possui 2 blocos de 8 bits. O módulo principal lê uma instrução de um núcleo do processador a cada ciclo de clock e executa a instrução em dois passos. O primeiro passo verifica a mudança de estado do bloco solicitado e faz o tratamento necessário da mensagem transmitida pelo barramento e o segundo passo acessa a memória para receber dados e fazer write-back. O diagrama do processador é ilustrado na figura.

Os dados de entrada para o programa são a instrução enviada pelo processador, o núcleo solicitante e o clock. A instrução possui 12 bits em que os 3 bits mais significativos são o endereço acessado, o oitavo bit informa a instrução e os 8 bits menos significativos contém o dado da instrução.

O barramento foi dividido em três seções, uma que envia somente dados, uma que envia somente o endereço e outra que envia somente a mensagem.

Dado	Instrução	Processador	Estado	Mensagem
0	Read	P1	Invalidate	Nada
1	Write	P2	Compartilhado	Write-Miss
2	-	P3	Modificado	Read-Miss
3	-	-	-	Invalidate



Sugestões e dificuldades

Como dificuldades, notamos que os conceitos são um pouco abstrato para a realização de um desenvolvimento conciso com apenas as informações fornecidas para a implementação do algoritmo do snooping, e isso faz com que o aluno acabe tendo que inferir ou pesquisar além do material fornecido, o que pode não atender o desejado tendo-se em vista a proposta da prática.

Além disso, a situação atual de final de período somado com o pouco tempo empregado para nos dedicar ao trabalho nos prejudicou em realizar totalmente a proposta do trabalho.

Sugerimos que desde o início fiquem definidos todos os detalhes relacionados à prática, como a totalidade do enunciado e o código esqueleto para servir de base inicial para o desenvolvimento da prática.

Conclusão

A implementação do snooping, que possui como característica o desenvolvimento dos microprocessadores com o uso de vários núcleos, nos permitiu entender como coordenar todos a fim de realizar operações em conjunto. Somado a isso, por meio de seu barramento, verificamos como pode-se comunicar com vários processadores por meio dos protocolos snooping e diretório.