Protocolo Snooping

Relatório de Laboratório de Arquitetura e Organização de Computadores II

Autores

Daniel Santana
Isaque Fernando

Centro Federal de Educação Tecnológica de Minas Gerais

2018

Sumário

1 Introdução

Este relatório surgiu com a necessidade de se documentar os detalhes de projeto da parte I da atividade prática 4 da disciplina de Laboratório de Arquitetura e Organização de Computadores 2. A parte em questão demandava uma implementação, em linguagem *Verilog*, da máquina de estados do protocolo Snooping, usado para solucionar o problema de coerência de *caches* que acontece atualmente.

Nos processadores atuais, o pararelismo tem sido notavelmente aplicado, e ele é o responsável pelo poder de processamento que temos hoje. Quanto melhor um processador consegue paralelizar suas instruções, mais rápido seu usuário final terá seus resultados esperados. Porém, nesse contexto de paralelismo, e compartilhamento de memórias cache (especialmente comum em nível L2) para obtenção mais rápida de dados, existe um grande problema a ser enfrentado, que é a coerência de dados entre as memórias cache. Um dado, quando requisitado ou alterado em uma cache, não pode estar inconsistente em outra, já que isso geraria problemas de confiabilidade, por causa do paralelismo explorado entre os processadores de uma máquina.

Para solucionar este problema, o protocolo de Snooping sugere o bloqueio de escrita, garantindo que um processador tenha acesso exclusivo a um item de dados antes que escreva neste item. Neste protocolo, o barramento é utilizado como uma forma de controle: Uma escrita não pode ser completada até se ter o acesso ao barramento. Enquanto isso, todos os outros processadores ficam "bisbilhotando"este barramento, ou seja, ficam em modo de escuta, para avaliar se elas possuem cópia do bloco solicitado. Para a serialização, o processador envia ao *bus* o endereço a ser invalidado, e todos os processadores que estão "bisbilhotando"o bus que têm este endereço, logo o invalidam.

2 Implementação

Foi utilizado o softwate *Quartus 2*, 64-bit, ver.13.0sp1 para a escrita do código do projeto. Para as imagens de simulação geradas e apresentadas abaixo, foi usado o software *ModelSim*, uma vez que os alunos possuem familiaridade com o uso destes dois programas, e já que estes possuem uma integração nativa e de fácil utilização.

2.1 Decisões de projeto

Este relatório visa apenas descrever a máquina de estados implementada. As decisões de projeto que concernem à máquina de estados são as seguintes.

- O sinal da CPU possui 4 bits. o MSB indica se a máquina em questão é a de escuta (0), ou a de escrita (1). O LSB indica se o bloco é compartilhado ou não por outras caches;

- Os significados dos sinais da CPU, excluindo o MSB e o LSB (portanto, os dois bits do meio), são: read miss - 00, read hit - 01, write miss - 10, e write hit - 11;
- O sinal do BUS possui dois bits, e os seus significados são: nada - 00, read miss - 01, write miss - 10, invalidate - 11.
- O sinal que vai para a memória possui 2 bits, sendo eles: nada - 00, abort access - 01, e write back - 10;
- Os estados das máquinas são representados, respectivamente, pelos bits: 00 - Invalid, 01 - Shared, 10 - Exclusive, 11 - Modified.

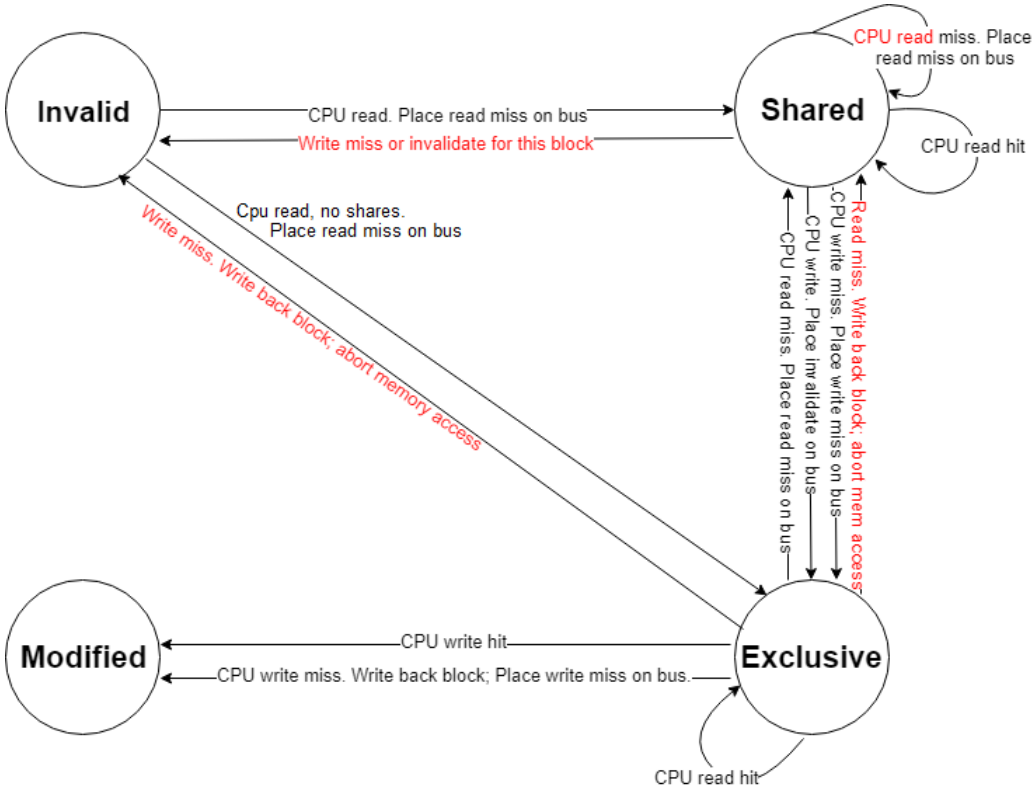

Figura 1: Máquina de estados do protocolo MESI de coerência de dados.

O protocolo MESI possui duas máquinas de estados: uma baseada nas informações enviadas pela CPU (descrita pelas setas com legenda preta), e outra baseada nas informações do *BUS* (descritas pelas setas com legenda vermelha). Para fins de praticidade, o grupo uniu as duas máquinas em uma única imagem. A descrição de cada transição se encontra a seguir.

2.2 Máquina de Escrita

Esta é a máquina de estados que funciona com base nas requisições da CPU. Seus estados, e respectivas transições são as seguintes.

2.2.1 Estado Invalid

Este estado indica que quaisquer dados contidos nesta cache estão inválidos

Caso a CPU indique a requisição de leitura, e o bloco requisitado exista em outra cache, o estado futuro desta máquina sera Shared, e uma mensagem de read miss será enviada ao bus.

No mesmo caso de requisição de leitura, se o bloco desejado não existir em outra cache, o estado futuro da máquina sera Exclusive, e um read miss é colocado no bus.

Caso a requisição seja de escrita, o estado futuro será Exclusive, um write miss será enviado ao barramento.

2.2.2 Estado Shared

Este estado indica que os dados contidos nesta cache existem também em outras tags.

Caso a CPU indique um read, o estado futuro será shared, mas o barramento só receberá algum sinal caso ocorra um miss. Neste caso, o sinal será um read miss.

Caso a CPU indique um write, o estado futuro da máquina será Exclusive. Além disso, caso seja um hit, o bus receberá um sinal de invalidade, e caso seja um miss, ele receberá um sinal de write miss.

2.2.3 Estado Exclusive

Este estado indica que os dados contidos nesta cache não estão válidos em nenhuma outra.

Caso a CPU indique um read hit, o estado futuro não será alterado, e o barramento não receberá sinais.

Caso haja um write hit, o estado futuro será Modified, e o barramento não receberá sinais.

Caso haja um write miss, o estado futuro será Modified, e o sinal de write miss será enviado ao bus.

De forma similar, caso ocorra um read miss, o estado futuro sera Shared, e um read miss será enviado ao bus.

2.2.4 Estado Modified

Este estado indica que, além de o dado buscado ser válido somente nesta cache, ele foi sobrescrito. Na máquina de Escrita, este estado não possui

transições para nenhum outro.

2.3 Máquina de Escuta

Esta é a máquina de estados que funciona com base nas informações do barramento. Seus estados, e respectivas transições são as seguintes.

2.3.1 Estado Invalid

Este estado não possui transições para nenhum outro nesta máquina.

2.3.2 Estado Shared

Caso haja um write miss, ou um invalidate para este bloco no barramento, o estado futuro será Invalid. Caso haja um read miss, o estado futuro será inalterado. Nenhum sinal para a memória é emitido.

2.3.3 Estado Exclusive

Caso "escute"um read miss neste bloco, o estado futuro será Shared. Se o sinal "escutado"for um invalidade, ou um write miss, o sinal futuro será Invalid. São enviados para a memória sinais para abortar o acesso de outros blocos à memória, para que ocorra o write-back deste bloco.

2.3.4 Estado Modified

Este estado não possui transições para nenhum outro nesta máquina.

3 Simulações

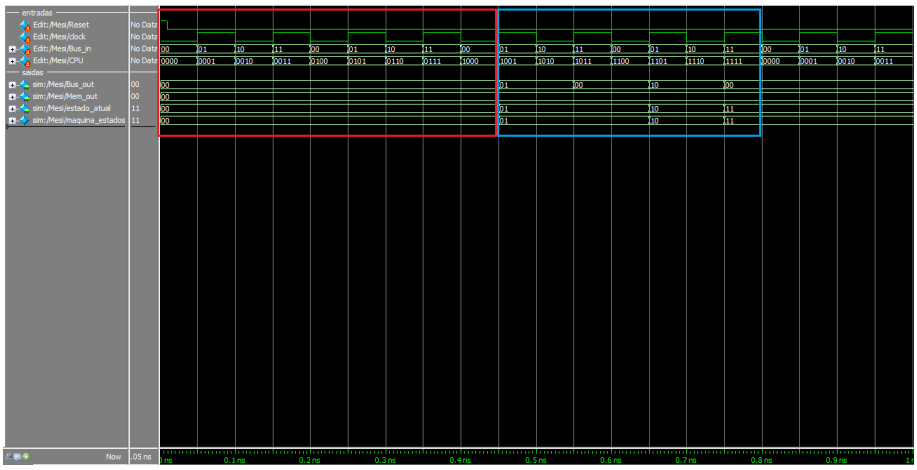

Figura 2: Simulação inicial

Por uma decisão de projeto do grupo, que será descrita e explicada em relatório futuro, a partir do momento do primeiro reset, a máquina de estados permanece inalterada durante toda a parte destacada de vermelho. A partir do momento em que a CPU comeca a enviar sinais com o bit mais significativo sendo 1, a máquina começa a trabalhar.

Nesta simulação, pode-se observar que, durante toda a marcação em azul, as informações enviadas pela CPU governam a máquina. O sinal estado_atual indica o estado em que está a máquina, enquanto que o sinal maquina_estados indica o estado futuro. A primeira mudança mostra que, ao acontecer um read hit na CPU (sinal 1001) enquanto a máquina está no estado Invalid (00 até a borda de subida do clock), o bus recebe um read miss (01), e o estado vai para Shared (01). Na segunda borda de clock, o bus recebe um sinal vazio, porque recebe da CPU um sinal de read hit (1011), o que faz com que a máquina permaneca no estado shared, e não envie nada ao bus.

Para as próximas simulações, vale ressaltar que foi feita uma alteração no módulo. Para a simulação anterior, o módulo não recebia o estado atual da máquina. Para facilitar as simulações, a partir da próxima imagem, isto foi modificado, de forma que, o estado atual das máquinas é enviado como entrada, através do sinal est_at. O código enviado na submissão é o código já com esta alteração.

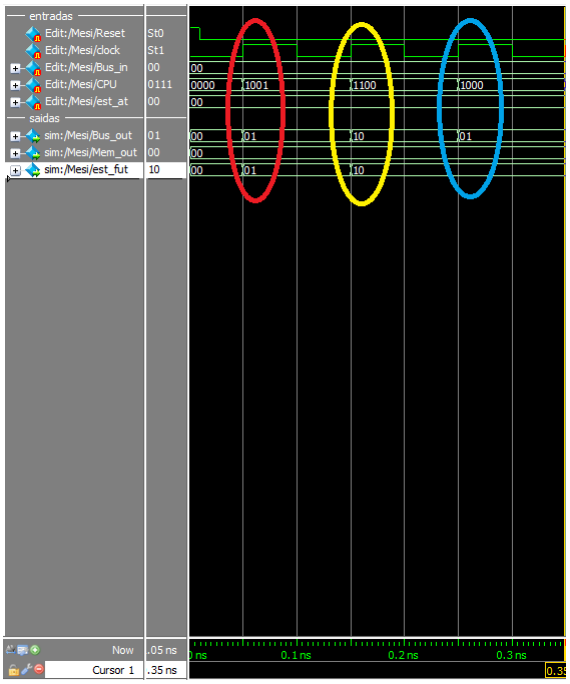

Figura 3: Simulação 1

Nesta simulação são exibidas todas as transições a partir do estado Invalid na máquina de escrita. Na marcação em vermelho, a CPU envia um read miss com bloco compartilhado (1001), o que produz um sinal read miss no bus (01), e o estado futuro é Shared (01). Na marcação amarela, a CPU envia um sinal de write miss(110x), o que produz um sinal de write miss no bus (10), e o estado futuro é Exclusive (10). Na última marcação, a CPU envia um read miss com bloco não compartilhado (1000), o que envia um read miss para o barramento (01), e o estado futuro é Exclusive (10).

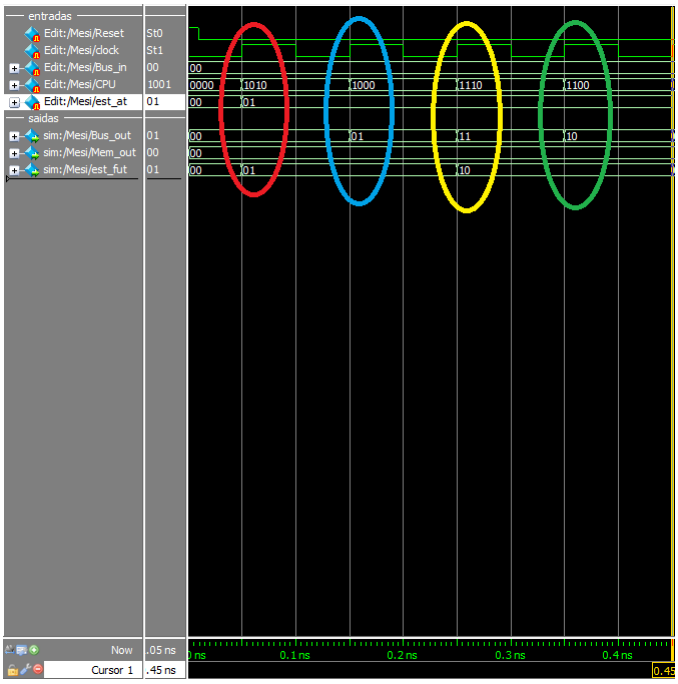

Figura 4: Simulação 2

Nesta simulação, pode-se observar todas as transições que partem do estado Shared da máquina de escrita. Respectivamente, os sinais gerados pela CPU são 1010 - read hit, 1000 - read miss, 1110 - write hit, e 1100 - write miss. Respectivamente, os estados futuros são Shared(01), Shared, Exclusive(10) e Exclusive, e a mensagem no barramento é, em cada ciclo respectivamente: nada(00), read miss(01), invalidate(11), e write miss(10), de acordo com a máquina de estados da figura 1.

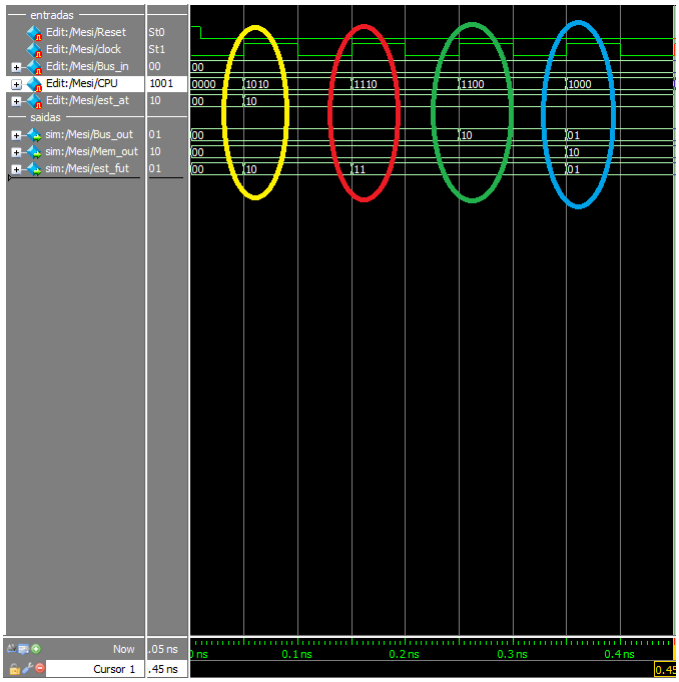

Figura 5: Simulação 3

Nas quatro marcações desta simulação, pode-se observar, respectivamente: o sinal read hit (1010) enviando 00 para o bus (nada) e produzindo Exclusive no estado futuro (10); o sinal write hit (1110) enviando nada para o bus, e produzindo o estado futuro Modified (11); o sinal write miss (1100) enviando write miss para o bus (10), e produzindo o estado futuro Modified (11); e por último, em azul, o sinal read miss (1000) enviando um read miss para o bus (01), um write back para a memória (10), e o estado futuro passa a ser Shared (01).

Para as próximas simulações, será apresentado apenas a máquina em estado de escuta (snoop). Portanto, o sinal enviado pela CPU tem o bit mais significativo 0 (decisão de projeto que indica que é esta máquina a operar).

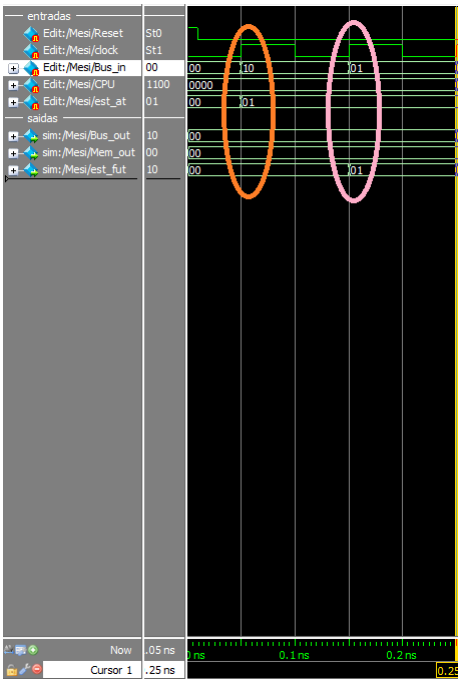

Figura 6: Simulação 4

Nesta simulação, é apresentada a máquina em estado de snoop. Portanto, ela ouvirá as mensagens provindas do barramento. Em alaranjado, pode-se observar que, no estado Shared (01) ao "escutar"o sinal write miss (10), a máquina passa para o estado Invalid (00), e envia nada (00) para o barramento e memória. Já ao "escutar"um read miss (01), o estado futuro é Shared (01), e também não envia nada para a memória ou bus.

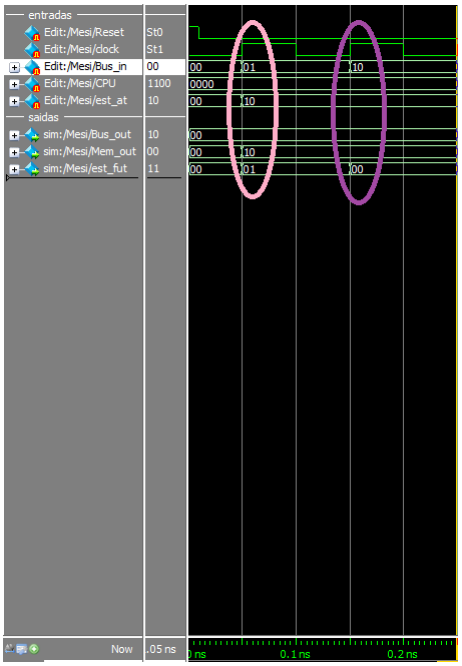

Figura 7: Simulação 5

Nesta simulação, podemos observar todas as transições da máquina de escuta a partir do estado Exclusive. Estas transições são, respectivamente: "escuta"um read miss do bus (01), envia nada para o bus (00) e um write back para a memória(10), e o estado futuro é Shared (01); e de roxo, "escuta"um write miss (10), envia nada para o bus (00), e um write back para a memória (10), e o estado futuro é Invalid (00).

4 Considerações Finais

O presente trabalho, em comparação com os demais, se mostrou um pouco mais simples, porém, exibiu vários detalhes para os quais devemos nos atentar para o prosseguimento da parte II. O grupo conseguiu se reunir em um fim de semana para realizar esta parte, o que facilitou muito a execução. A execução desta parte da prática foi proveitosa, no que possibilitou um entendimento maior da parte teórica da matéria. Durante a implementação, ficou um pouco mais claro o protocolo, mostrando que, realmente, o contato prático facilita o entendimento da disciplina.

Obs: As simulações podem ser vistas a parte caso não estejam visíveis no relatório.