CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS

TRABALHO PRÁTICO IV - PARTE I

A máquina de estados do protocolo snooping

Arthur Estevão de Souza Machado
Marcelo Lopes de Macedo Ferreira Cândido

Engenharia de Computação
Laboratório de Arquitetura e Organização de Computadores II

23 de novembro de 2018

1 INTRODUÇÃO

A busca constante por maior capacidade de processamento sempre foi algo presente desde o surgimento do primeiro computador e com isso, surgiram técnicas tanto em hardware quanto em software a fim de alavancar este propósito. O surgimento de multiprocessadores certamente proporcionou grande desenvolvimento no mundo da tecnologia, mas com isso, a consistência de dados entre os processadores se tornou um novo problema a ser contornado.

Diante disso, surgiram protocolos de coerência de cache que visam manter a coerência para múltiplos processadores, de forma que os dados armazenados estejam em harmonia com as atualizações proporcionadas, oferecendo a possibilidade de replicação de dados que são lidos, reduzindo a latência de acesso e a disputa pela informação.

O protocolo *Snooping* é uma solução eficiente, simples e muito utilizado para o problema de coerência. Tal protocolo consiste em verificar as alterações enviadas ao barramento a fim de tomar as devidas ações para manter a coerência, alterando assim, o estado de compartilhamento quando necessário.

As alterações de estado ocorrem através da utilização de uma máquina de estados. Tendo em vista este fato, as modificações de estado dependem diretamente de um estado atual e de sua condição de modificação sendo necessário, muitas vezes, realizar ações complementares como por exemplo a política de escrita *Write-Back*, a fim de atualizar blocos que necessitam de um dado desatualizado, ou a Invalidação, que existe para garantir que exista apenas uma cópia de um bloco em específico modificada.

O presente relatório, tem como objetivo apresentar um projeto de implementação da máquina de estados do protocolo MESI *Snooping*, uma otimização do protocolo MSI que elimina a necessidade de realizar atualizações dos blocos que são lidos e mais tarde escritos por um único processador.

2 FUNCIONAMENTO E DETALHES DA IMPLEMENTAÇÃO

O protocolo MESI conta com quatro estados fundamentais em sua máquina de estados, são eles:

- Inválido: o conteúdo ou não está contido na cache ou não está atualizado.

- Compartilhado: o conteúdo presente na cache também está presente em outras caches e encontra-se atualizado.
- Exclusivo: a cache e a memória principal são as únicas que possuem este dado, que está atualizado.
- Modificado: somente a cache possui o dado mais recente, ou seja, a memória principal ainda não está atualizada.

A máquina de estados implementada possui o seguinte padrão apresentado nas figuras 1 e 2.

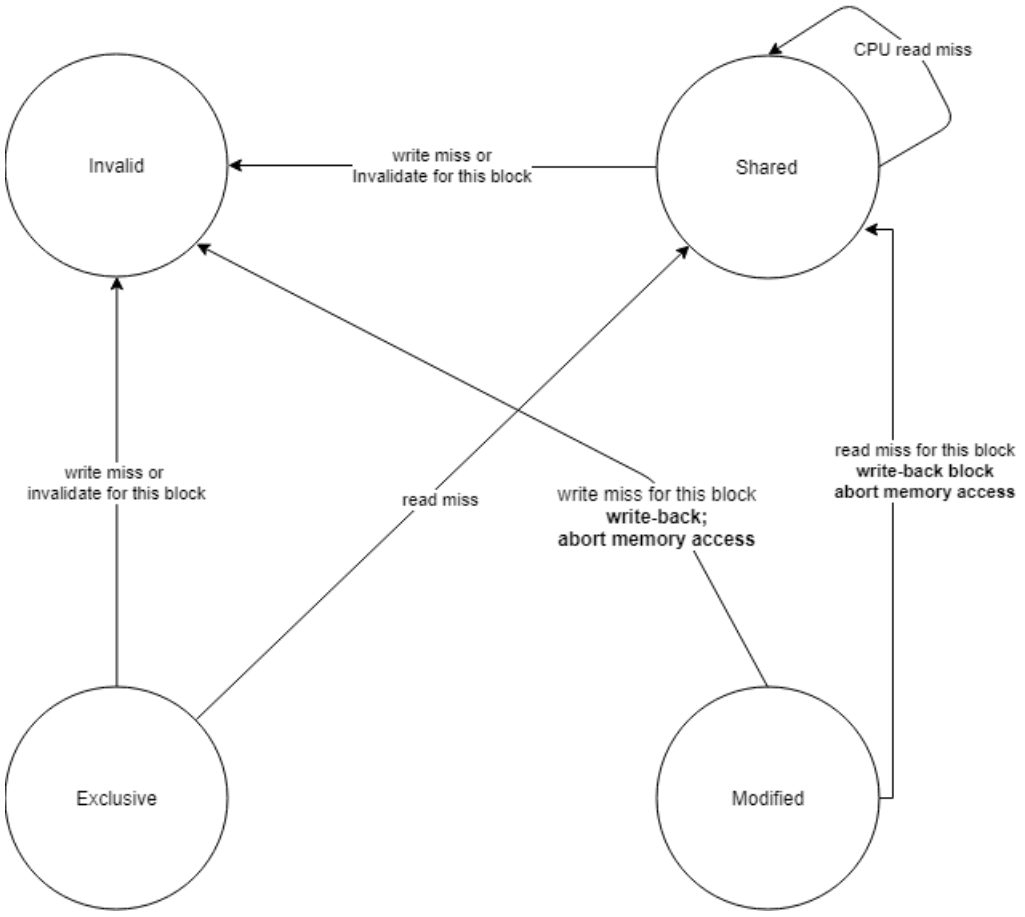

Figura 1: Máquina de estados para o ouvinte de operação

Diante disso, as mudanças de estado irão depender diretamente do estado atual da máquina e do evento que ela recebe. As decisões de implementação foram:

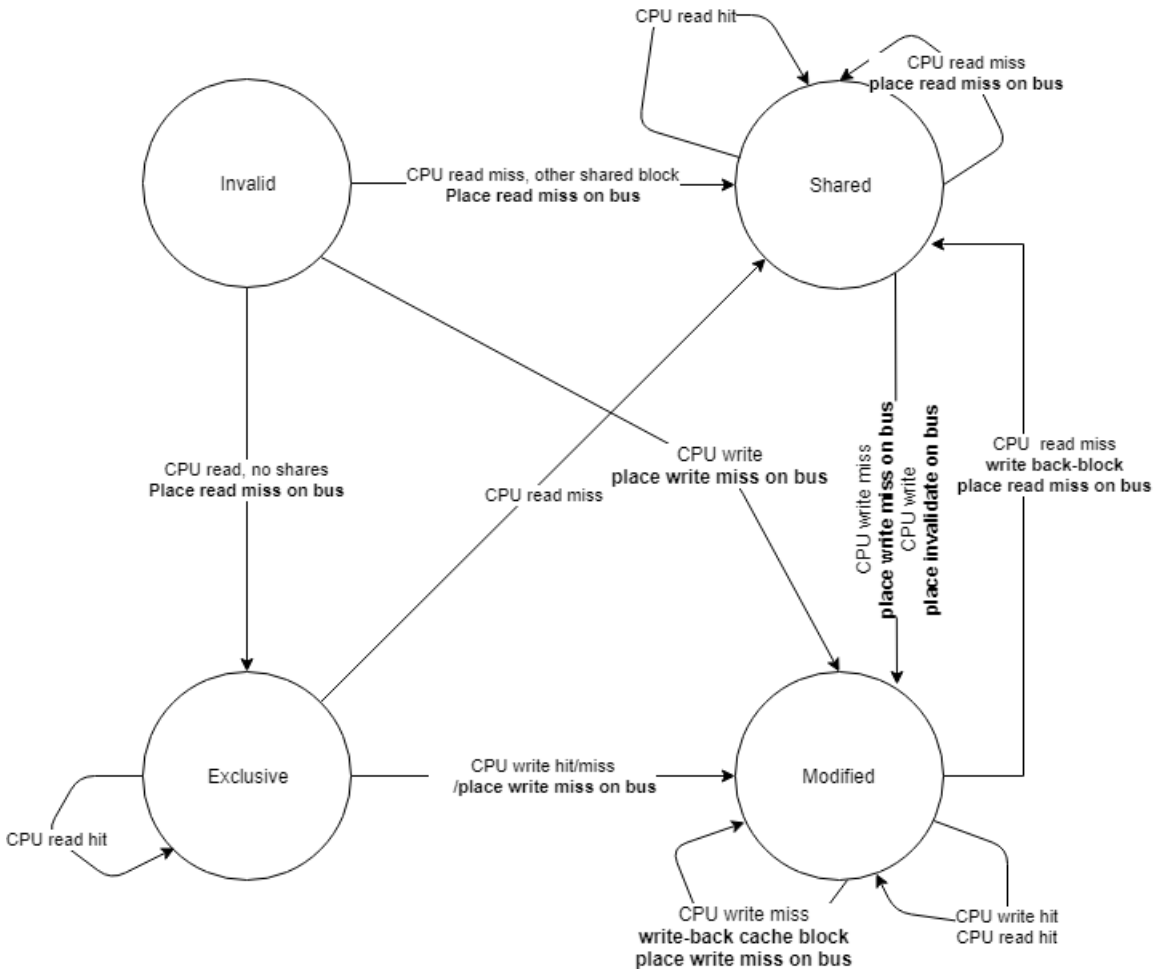

Figura 2: Máquina de estados para o emissor de operação

- Evento corrente da CPU (CPU event) = 5 bits.
- Estado da cache = 3 bits.
- Saída para o barramento = 6 bits

A escolha do número de bits para o CPU Event foi baseada nos eventos possíveis; são eles: Invalidação (10000), Write Hit (01000), Write Miss (00100), Read Hit (00010), Read Miss (00001) e Read Miss, other shared block (10001). Os estados são representados como inválido (001), compartilhado (010), exclusivo (011), modificado (100).Já o barramento pode indicar até duas informações, com 3 bits para cada, como apresentado nos diagramas das Figuras 1 e 2, não

possuindo nessa implementação o endereço, com o objetivo de simplificar o projeto para este caso específico.

3 SIMULAÇÕES

Para testar a funcionalidade da máquina de estados projetada, serão realizadas simulações no software ModelSim 10.1.d. Os testes vão passar por todos os casos possíveis e serão realizados variando o estado da máquina conforme o CPU Event e alternando os modos de uso, entre ouvinte e emissora.

As legendas utilizadas para os eventos encaminhados ao bloco estão decodificadas a seguir:

- Em vermelho, temos read miss.
- Em rosa, temos read hit.
- Em laranja, temos write miss.
- Em verde, temos write hit.
- Em amarelo, temos read miss, *other shared block*.
- Em roxo, temos *invalidate*.

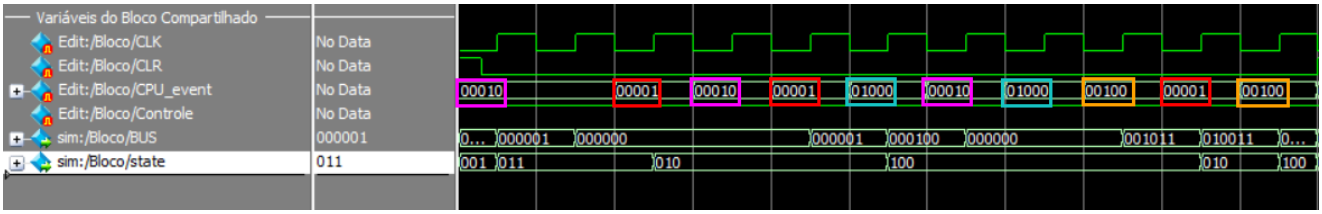

Figura 3: Controle sempre ligado, ou seja, a máquina está ativa como emissora.

As alterações estão descritas em ordem:

- A máquina no estado inválido (001) emite read miss e vai para exclusivo.
- A máquina no estado exclusivo (011)emite read miss e vai para compartilhado.

- A máquina no estado compartilhado (010) emite read hit e não é modificada.
- A máquina no estado compartilhado (010) emite read miss não é modificada.
- A máquina no estado compartilhado (010) emite write hit e vai para modificado.
- A máquina no estado modificado (100) emite read hit e não é modificada.
- A máquina no estado modificado (100) emite write hit e não é modificada.
- A máquina no estado modificado (100) emite write miss e não é modificada.
- A máquina no estado modificado (100) emite read miss e vai para compartilhado.
- A máquina no estado compartilhado (010) emite write miss e foi para modificado.

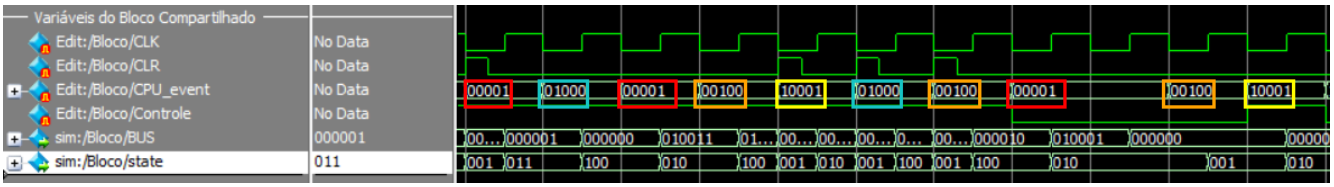

Figura 4: Controle ligado, máquina emissora e ouvinte caso contrário.

- Clear.
- A máquina no estado inválido (001) emite read miss e vai para exclusivo.
- A máquina no estado exclusivo (011) emite write hit e vai para modificado.
- A máquina no estado modificado (100) emite read miss e vai para compartilhado.
- A máquina no estado compartilhado (010) emite write miss e vai para modificado.
- Clear.
- A máquina no estado inválido (001) emite read miss e vai para exclusivo.
- A máquina no estado exclusivo (011) emite write miss e vai para modificado.
- Clear.

- A máquina no estado inválido (001) emite read miss, other shared block e vai para compartilhado.
- Clear.
- A máquina no estado inválido (001) emite write miss e vai para modificado.
- Controle vai para 0, ativando a máquina ouvinte.
- A máquina no estado modificado (100) emite read miss e vai para compartilhado.
- A máquina no estado compartilhado (010) emite write miss e vai para inválido.
- A máquina no estado inválido (001) emite write miss e vai para compartilhado.

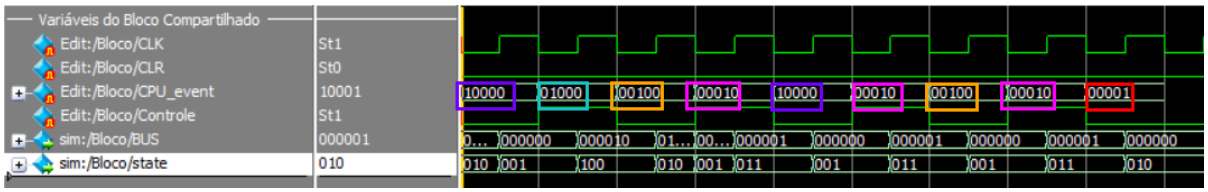

Figura 5: Controle sempre ligado, ou seja, a máquina está ativa como emissora.

- Em vermelho, temos read miss.
- Em rosa, temos read hit.
- Em laranja, temos write miss.
- Em verde, temos write hit.
- Em amarelo, temos read miss, *other shared block*.
- Em roxo, temos *invalidate*.
- A máquina no estado compartilhado (010) emite invalidate e vai para inválido.
- Controle vai para 1, ativando a máquina emissora.
- A máquina no estado inválido (001) emite write hit e vai para modificado.

- Controle vai para 0, ativando a máquina ouvinte.
- A máquina no estado modificado (100) emite write miss e vai para compartilhado.
- Controle vai para 1, ativando a máquina emissora.
- A máquina no estado inválido (001) emite read hit e vai para exclusivo.
- Controle vai para 0, ativando a máquina ouvinte.
- A máquina no estado exclusivo (011) emite invalidate e vai para inválido.
- Controle vai para 1, ativando a máquina emissora.
- A máquina no estado inválido (001) emite read hit e vai para exclusivo.
- Controle vai para 0, ativando a máquina ouvinte.
- A máquina no estado exclusivo (011) emite write miss e vai para inválido.
- Controle vai para 1, ativando a máquina emissora.
- A máquina no estado inválido (001) emite read hit e vai para exclusivo.
- Controle vai para 0, ativando a máquina ouvinte.
- A máquina no estado exclusivo (011) emite read miss e vai para compartilhado.

4 DIFICULDADES, SUGESTÕES E CONSIDERAÇÕES FINAIS

Na prática em questão foi possível trabalhar com a máquina de estados de um protocolo de coerência de cache, propiciando conhecimentos sobre um autômato que segue um modelo, dependente diretamente do estado atual e das informações encaminhadas.

Como dificuldade, existe a compreensão sobre o assunto coerência de cache e suas funcionalidades. De fato, tal tema tem grande importância para os processadores utilizados atualmente e contribui diretamente com paralelismo em nível de thread, proporcionando ao aluno um tema atual e extremamente útil.