Bruno Henrique Amaral Fonseca, Heitor Amaral Santos, Yan Victor Sirianni Azevedo

Trabalho Prático I - Laboratório de Arquitetura e Organização de Computadores II

Trabalho Prático I da Disciplina de Laboratório de Arquitetura e Organização de Computadores II ministrada pelo professor Jeferson Figueiredo Chaves.

Centro Federal de Educação Tecnológica de Minas Gerais – CEFET-MG

Departamento de Computação

Curso de Engenharia da Computação

Belo Horizonte

2019

Resumo

Este trabalho contempla a implementação de uma memória RAM utilizando a biblioteca LPM, do Quartus II. A leitura foi realizada utilizando o display de 7-segmentos e a escrita utilizando os switchs da placa.

Palavras-chave: Memória RAM. Quartus II. Altera.

Sumário

1 Introdução

O conceito de Hierarquia de Memória diz respeito à multi-níveis de memória com tamanhos e velocidades diferentes. As mais rápidas são as que possuem maior custo de armazenamento por bit, e portanto são menores. Existem princípios que precisam ser considerados durante a implementação de um sistema com hierarquia de memória principio da localidade temporal, ou seja, se algum item é referenciado, ele tenderá a ser referenciado novamente. Principio da localidade Espacial, se um item é referenciado, itens cujos endereços são próximos a este, tenderam a ser referenciados também. Neste projeto, será abordado o principio de localidade temporal, e o seu principal objetivo é a aplicação dos conceitos de hierarquia de memória para a implementação de um sistema de memorias cache e RAM. Cada parte contemplada por este relatório referencia uma etapa de implementação do sistema.[3]

2 Detalhes do projeto

2.1 Verilog HDL

Verilog é uma linguagem de descrição de alto nível utilizada tanto para simulação quanto para síntese. Neste projeto será utilizada a Linguagem de Descrição de Hardware, HDL - *Hardware Description Language* que consiste em uma linguagem de programação de software utilizada para modelar um hardware ou parte dele.

2.1.1 FPGA

Para simulação do sistema em hardware, será utilizada uma placa de desenvolvimento FPGA - *field programmable gate array*. É uma placa para aprendizagem, e faz uso do ambiente de desenvolvimento da própria ALTERA – Quartus II/Prime.[2]

2.1.2 Decisões em relação à pinagem

A escrita na memória é realizada através do uso das switchs da FPGA para endereçamento, dados e habilitação de escrita.

2.1.2.1 Escrita na memória

Para a passagem dos dados, foram escolhidas seguintes switchs SW17,SW16,SW15, SW14,SW13,SW12,SW11,SW10. A switch para habilitação de escrita foi definida como SW7 e, para o endereçamento da RAM, foram selecionadas as switchs SW4,SW3,SW2,SW1,SW0 sendo esta ultima o bit menos significativo.

2.1.2.1.1 Arquivo de inicialização de memória - MIF

O arquivo .mif é um arquivo de texto ASCII (com a extensão mif.), que especifica o conteúdo inicial de um bloco de memória (CAM, RAM, ou ROM), isto é, os valores iniciais para cada endereço. Este arquivo é usado neste projeto principalmente para testar a leitura de dados nas posições da memória.[1]

2.1.2.2 Display de 7 Segmentos

Neste projeto serão usados 8 displays 7-segmentos para controle e exibição do endereço e dos dados para leitura e escrita. Para a operação de leitura foi usados os displays HEX7 e HEX6, para a escrita os displays HEX5 e HEX4 e para o endereçamento foram usados HEX3, HEX2, HEX1, HEX0.

2.2 Etapas de implementação

2.2.1 Partes 1 e 2

A primeira parte consiste na implementação de uma memória RAM utilizando a biblioteca LPM, os procedimentos de leitura e escrita serão realizados utilizando o display 7-segmentos. A segunda etapa diz respeito à inicialização da memória implementada na primeira parte, utilizando o *MIF - memory inicialization file*.

2.2.2 Parte 3

A terceira parte deste projeto, consiste na implementação de uma cache assossiativa por conjunto de 2 vias, utilizando o MIF e o display de 7-segmentos para as operações de leitura e escrita. Na terceira parte, será demonstrado o que ocorre no sistema em caso de acerto ou falha de leitura ou escrita na cache, situações que modificam os bits "Dirty", "LRU"e Válido.

2.2.3 Diagrama de Projeto

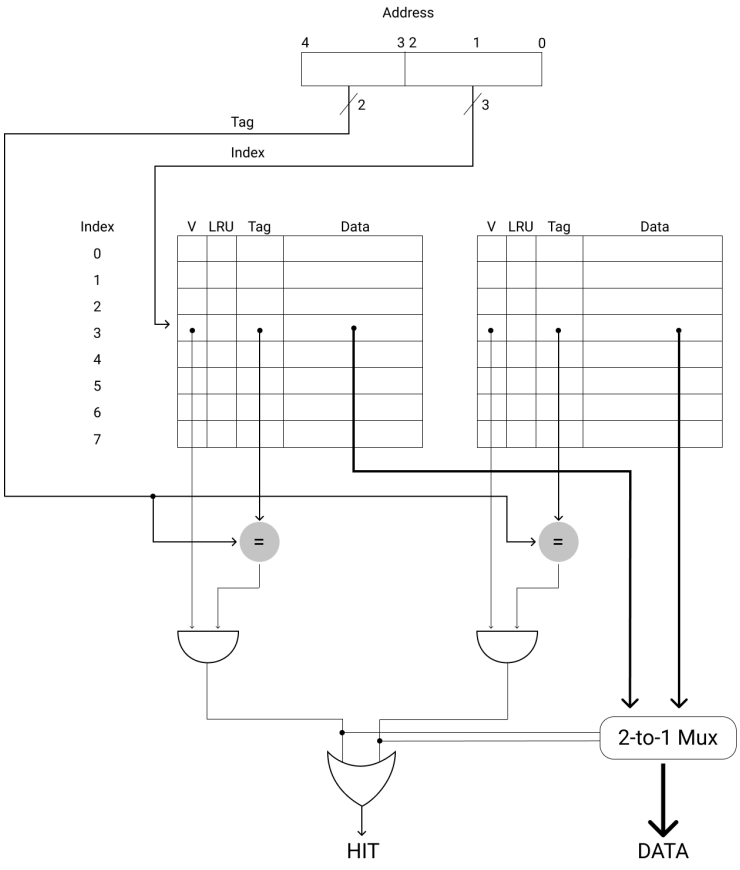

Figura 1 – Modelo referência para implementação

3 Simulações

3.1 Partes 1 e 2

3.1.1 ModelSim Altera

3.1.1.0.1 Detalhes da Simulação da Parte 1

Relação dos displays de 7 segmentos:

- HEX7 e HEX6 são os displays que mostram o dado que está sendo lido da RAM (em hexadecimal).
- HEX5 e HEX4 são os displays que mostram o dado que será escrito na RAM (em hexadecimal).
- HEX1 e HEX0 são os displays que mostram o endereço da RAM que será lido ou escrito (em hexadecimal).

Do instante 0ps até o 200ps:

- SW = 8'b0 5'b0 5'b0. Nada foi escrito e somente foi lido do endereço 0 da RAM. Nesse momento todos os displays estão marcando 0.

Do instante 200ps até o 400ps:

- SW = 8'b00001000 5'b00100 5'b0 Como o writeEnable está habilitado, será escrito o numero 8 no endereço 0 da RAM.

Nesse momento é mostrado:

- 08 nos displays HEX7 e HEX6.
- 08 nos displays HEX5 e HEX4.
- 00 nos displays HEX1 e HEX0.

Do instante 400ps até o 600ps:

- SW = 8'b00001100 5'b00100 5'b00001 Como o writeEnable está habilitado, será escrito o numero 12 no endereço 1 da RAM.

Nesse momento é mostrado:

- 0C nos displays HEX7 e HEX6.
- 0C nos displays HEX5 e HEX4.
- 01 nos displays HEX1 e HEX0.

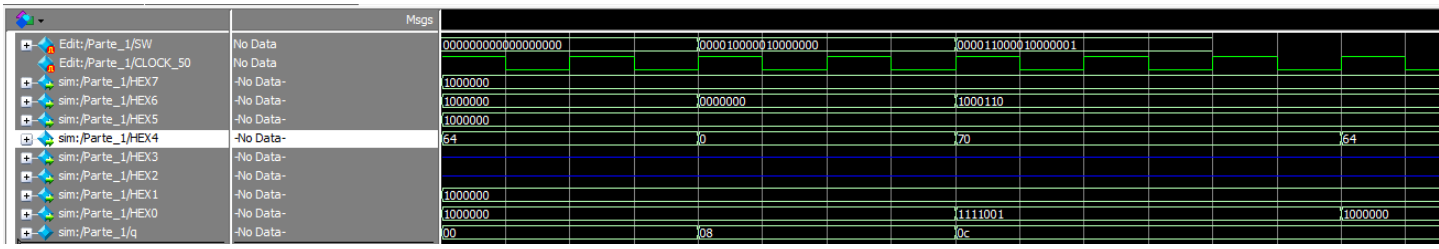

Figura 2 – Simulação parte 1

3.1.1.0.2 Detalhes da Simulação da Parte 2

Relação dos displays de 7 segmentos:

- HEX7 e HEX6 são os displays que mostram o dado que está sendo lido da RAM (em hexadecimal).
- HEX5 e HEX4 são os displays que mostram o dado que será escrito na RAM (em hexadecimal).
- HEX1 e HEX0 são os displays que mostram o endereço da RAM que será lido ou escrito (em hexadecimal).

Foi utilizado nessa prática um arquivo .mif de inicialização da memória RAM. **Do instante 0ps até o 200ps:**

- SW = 8'b0 5'b0 5'b0

Como o writeEnable NÂO está habilitado, será mostrado o numero 1 no endereço 0 da RAM, pois já foi declarado no arquivo de inicialização da memória. Nesse momento é mostrado:

- 01 nos displays HEX7 e HEX6.
- 00 nos displays HEX5 e HEX4.
- 00 nos displays HEX1 e HEX0.

Do instante 200ps até o 400ps:

- SW = 8'b0 5'b0 5'b01

Como o writeEnable NÂO está habilitado, será mostrado o numero 2 no endereço 1 da RAM, pois já foi declarado no arquivo de inicialização da memória. **Nesse momento é mostrado:**

- 02 nos displays HEX7 e HEX6.
- 00 nos displays HEX5 e HEX4.
- 01 nos displays HEX1 e HEX0.

Do instante 400ps até o 600ps:

- SW = 8'b0 5'b0 5'b000010

Como o writeEnable NÂO está habilitado, será mostrado o numero 3 no endereço 2 da RAM, pois já foi declarado no arquivo de inicialização da memória.
Nesse momento é mostrado:

- 03 nos displays HEX7 e HEX6.
- 00 nos displays HEX5 e HEX4.
- 10 nos displays HEX1 e HEX0.

Do instante 600ps até o 800ps:

- SW = 8'b0 5'b0 5'b000011

Como o writeEnable NÂO está habilitado, será mostrado o numero 4 no endereço 3 da RAM, pois já foi declarado no arquivo de inicialização da memória.
Nesse momento é mostrado:

- 04 nos displays HEX7 e HEX6.
- 00 nos displays HEX5 e HEX4.
- 11 nos displays HEX1 e HEX0.

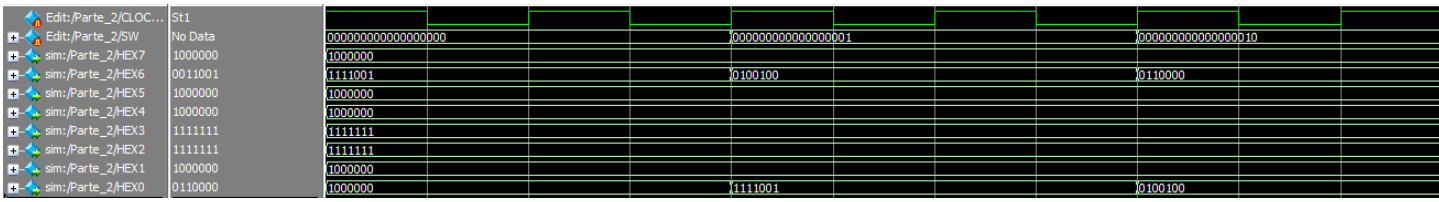

Figura 3 – Simulação parte 2

3.1.2 Parte 3

3.1.2.0.1 Detalhes da Simulação da Parte 3

As escolhas de pinagem se mantém as mesmas das Partes 1 e 2.

Para a implementação desta parte, foi adicionado três mux, dois para selecionar o dado de escrita na cache, um para cada via, e outro para definir a que saída de qual via deverá ser enviada para a placa. Para esta simulação foram adicionadas as waves para visualização de Hit, Miss, bit de LRU das vias 1 e 2, junto com as saídas dos mux de cada via e a saída das vias com os dados. A simulação mostra, até a primeira borda de descida do clock, um Miss em uma tentativa de acesso na cache. No intervalo entre a primeira e a segunda borda de descida, as ondas simulam uma em nível inferior da hierarquia, a memoria RAM. A terceira borda de descida do clock sinaliza um HIT, o que pode ser observado também pela mudança nos valores de lru_via1 e lru_via2 e também pela saída do dado no d_out_mux1 e na data_out_via1 da cache. A partir daí as ondas retornam ao comportamento padrão esperado, e passam a exibir o dado do arquivo .mif.

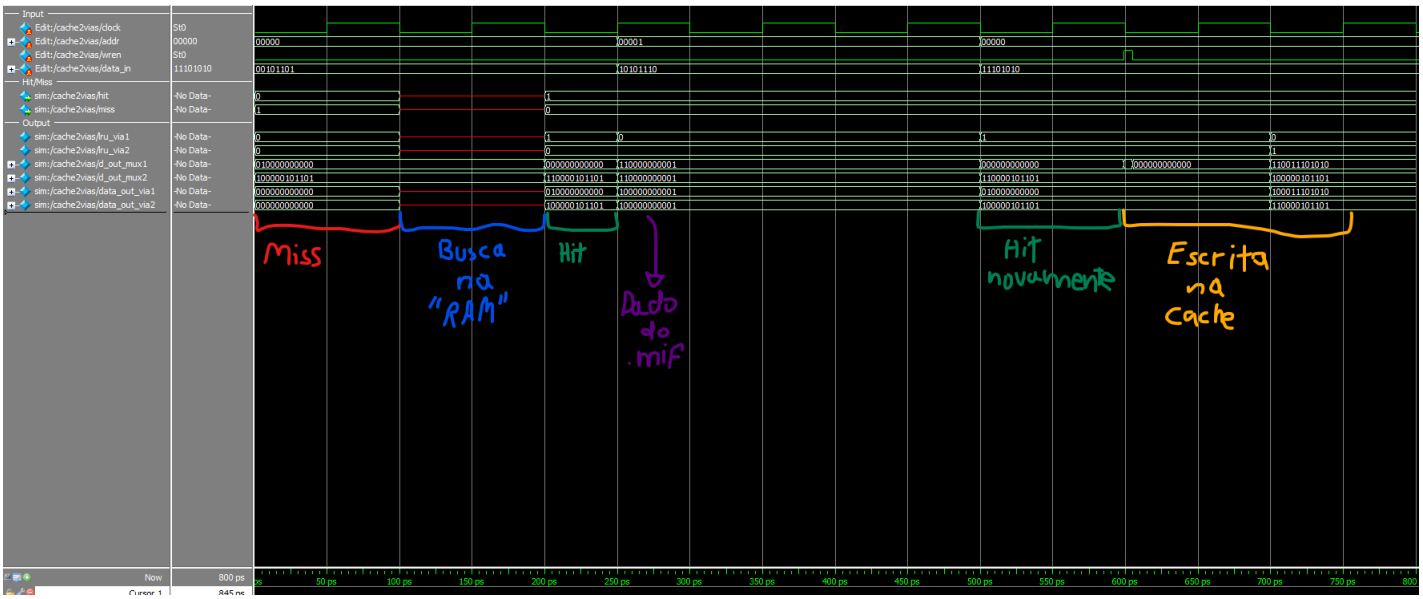

Figura 4 – Simulação parte 3

4 Dificuldades encontradas

A principal dificuldade encontrada no desenvolvimento deste projeto foi a modelagem e planejamento de uma solução que segui-se as regras impostas pelo professor. O que foi causada pelo desajuste na compreensão das informações acerca da implementação e quais seriam os impactos de quaisquer alterações não previstas para o fluxo do código. Em nível de implementação, o maior desafio foi definir a forma de abordagem para selecionar o que deveria ser escrito na cache, e após debate foi decidido que seriam usados multiplexadores para esta função e também para selecionar a saída da via correta para enviar para a placa.

5 Conclusão

Para se desenvolver este projeto, que possui certo nível de complexidade, é necessário que se faça um estudo de caso para compreensão do problema e então se construa um planejamento antes de que qualquer implementação seja feita. A opção por usar os três multiplexadores foi o que tornou possível a conclusão deste projeto, pois posteriormente foi informado que não seria permitido o uso de condicionais "if/else"e nem o @always no código de verilog. Para a implementação de um sistema de hierarquia de memória é recomendado que se siga fielmente os princípios téoricos, principalmente os princípios de localidade, para que seja possível a implementação de uma solução relativamente simples para o problema de velocidade de acesso. [1]

Referências

[1] IFSC.edu.br,. Inicialização de memória com arquivo .MIF e .HEX. Disponível em: <https://wiki.sj.ifsc.edu.br/wiki/index.php/inicializacao_de_memoria_com_arquivo_.MIF_e_.HEX> Acesso em 28 de março de 2019.

[2] INTEL.com,. FPGA. Disponível em: <https://www.intel.com/content/www/us/en/products/programmable/fpga/new-to-fpgas/resource-center/overview.html> Acesso em 2 de abril de 2019.

[3] IC.unicamp,. Sistemas de Memória – Hierarquia de Memória. Disponível em: <http://www.ic.unicamp.br/~ducatte/mc542/Arquitetura/arq_hp7.pdf> Acesso em 2 de abril de 2019.