Centro Federal de Educação Tecnológica de Minas Gerais - CEFET-MG Curso de Engenharia de Computação Laboratório de Arquitetura e Organização de Computadores II - 2017/1 Trabalho Prático 1: Hierarquia de Memória

Victor Balbo de Oliveira Vinícius Henrique Silveira

Parte 1

O objetivo dessa parte da prática foi testar uma memória RAM que pode ser criada automaticamente no software Quartus II. A memória tem 32 palavras de 8 bits e sendo assim endereçada por 5 bits. Há um sinal de controle que habilita leitura em 0 e escrita em 1 e um dado de saída de 8 bits. A memória foi simulada utilizado-se o software Modelsin e obteve-se a forma de onda abaixo como produto da simulação.

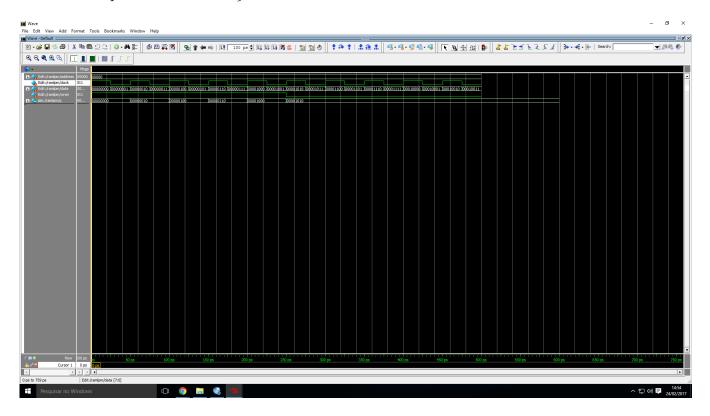


Figura 1: Simulação da RAM importada do Quartus II

Parte 2

Utilizou-se a mesma memória RAM construída na Parte 1, porém, ela foi carregada com valores iniciais a partir de um arquivo .mif (memory initialization file). O resultado da simulação pode ser visto na forma de onda abaixo:

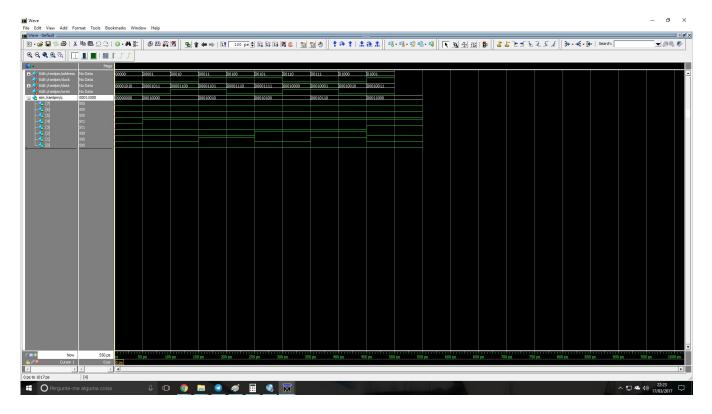
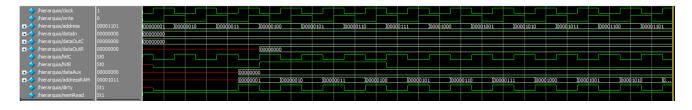


Figura 2: Simulação da RAM importada do Quartus II utilizando-se um arquivo .mif

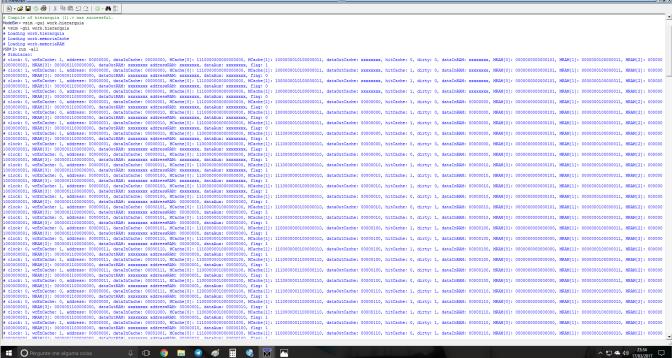
Parte 3

O objetivo dessa parte da prática foi implementar uma hierarquia de memória com dois níveis (uma cache L1 e uma memória RAM). A cache L1 é totalmente associativa e a memória RAM é diretamente mapeada. Criou-se uma cache de duas linhas de dados e uma memória RAM com quatro linhas de dados. Como a cache é totalmente associativa, o dado é substituído na cache em caso de miss utilizado-se a lógica LRU (Least Recently Used). Utilizou-se um bit adicional na linha de cache para tratar dessa lógica (1 indica dado recentemente acessado e 0 que foi acessado a algum tempo). Há também um bit de validade que indica que o dado naquela posição da cache é válido e um bit de sujeira que indica que o dado na cache é inconsistente com o dado presente na memória RAM. Utilizou-se a politica de escrita write-back para atualizar o dado na RAM, ou seja, sempre que um bloco tiver que ser substituído na cache, verificamos seu bit de sujeira, se ele estiver ativo indica que o dado precisa ser salvo na memória RAM antes de ser substutuido. A hierarquia de memória foi implementada em Verilog HDL e simulada no software Modelsin. O resultado da simulação pode ser visto na forma de onda e no display de dados abaixo:



Transcript
File Edit View Bookmarks Window Help
Transcript

Transc



Transcript - 0 × View Bookmarks Window Help File Edit + # × B-2-3-3-3-1-3-8-8-12-1-0-4-\$ 00110, dataIr dataOutRAM: 000110, dataIr dataOutRAM: e: 0, address: 00 Cache: 00000000, MCac 00000000 addressRAM: the[0]: 11 he[0]: 111 1 [1]: 110000011010000000, dataCutCache: 00000000, hitCache: 1, dirty: 0, dataInRAM: 00000000, MRAM[0]: 000000000000101, MRAM[1]: 00000010000000, MRAM[2]: 000000 e: 1, address: 00 00111, dataIr dataOutRAM: 000111, dataIr dataOutRAM: flag: 0

flacting: 0

flag: 0 clock: 0 the[0]: 11 he[0]: 11 0 [1]: 1110000011100000000, dataOutCache: 00000000, hitCache: 0, dirty: 1, dataInEAN: 00000000, MRAM[0]: 000000000000101, MRAM[1]: 00000010000000, MRAM[2]: 000000 Michefel; 111000011100000000, dataOutCache: 00000000, nitGache: 0, dirty: 1, dataInEAM: 00000000, MSAM[0]: 000000000000101, MSAM[1]: 000000100000000, MSAM[1]: 0000001
Michefel: 111000011100000000, dataOutCache: 00000000, hitGache: 0, dirty: 1, dataInEAM: 00000000, MSAM[0]: 000000000000101, MSAM[1]: 000000100000000, MSAM[2]: 000000
Michefel: 111000011100000000, dataOutCache: 000000000, hitGache: 1, dirty: 0, dataInEAM: 00000000, MSAM[0]: 000000000000101, MSAM[1]: 000000100000000, MSAM[2]: 000000

ヘ 炉 ▲ ⑷ 20:56 □