

Prática 1 - parte 1 e 2

Erick Henrique
Marina Bernardes

06/2021

—

Laboratório de Arquitetura e
Organização de Computadores II

2021.1

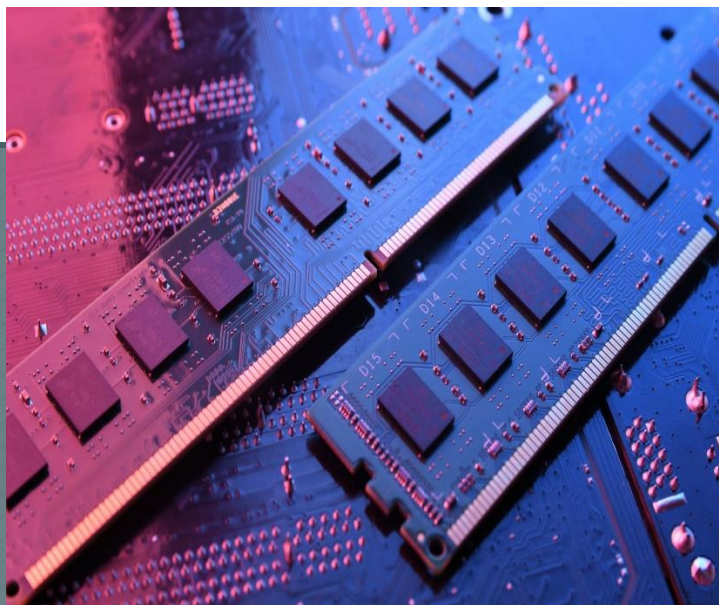
— —

Daniela Cristina Cascini Kupsch

Centro Federal de Educação Tecnológica de Minas
Gerais

OBJETIVOS

Esta prática tem a finalidade de exercitar os conceitos relacionados à hierarquia de memória.





O PROCESSO

Parte 1

Utilizando os softwares Quartus e ModelSim, implementamos uma memória RAM utilizando a biblioteca LPM.

Essa memória possui 32 palavras de 8 bits, que são acessadas por uma porta com endereçamento de cinco bits.

Para montar a biblioteca LPM utilizamos os passos contidos no arquivo “*Laboratory Exercise 8, Memory Blocks*”, disponibilizado pela professora. Criamos um arquivo em Verilog HDL para a chamada dessa biblioteca. O teste foi realizado com o número de ordem de chamada da dupla (Erick “3” e Marina “8”). Esses números foram escritos em posições distintas da memória (primeira e segunda posição), em seguida ocorreu a leitura destas posições.

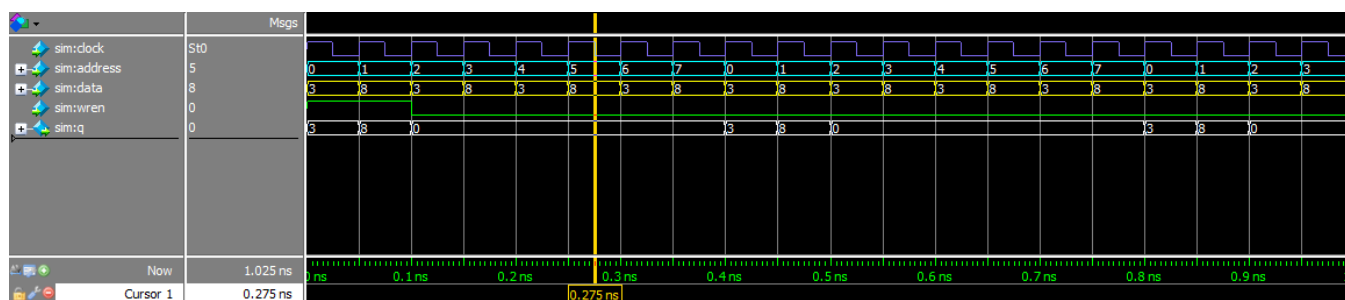
Para a simulação, geramos ondas para cada input:

- Clock: 50 ps;
- Wren: repeater 1-1-1-0-0-0-0-0-0-0;
- Address: counter;
- Data: 3 e 8.

Algoritmo implementando em Verilog HDL

```
module partel (address, data, clock, wren, q);  
  
    input    [4:0]  address;  
    input    clock, wren;  
    input [7:0]  data;  
    output [7:0]  q;  
  
    Ram MemoriaRam(address, clock, data, wren, q);  
  
Endmodule
```

Simulação no ModelSim



Como podemos perceber pela simulação, nas duas primeiras posições da memória é feita a escrita das posições da chamada, enquanto wren está habilitado para a escrita e salvo na saída. Ao decorrer da simulação, é feita a leitura destes dados.



Parte 2

Utilizando os softwares Quartus e ModelSim, inicializamos a memória RAM utilizando um arquivo MIF (*“Memory Initialization File”*).

Para montar esse arquivo seguimos os passos contidos no arquivo *“Laboratory Exercise 8, Memory Blocks”*, disponibilizado pela professora. Registramos as duas primeiras posições da memória com o número da nossa chamada e as demais posições com números sequenciais ao maior número (8). Linkamos na biblioteca LMP, o ***ramlpm.mif*** gerado. Por fim, criamos um arquivo em Verilog HDL para a chamada dessa biblioteca.

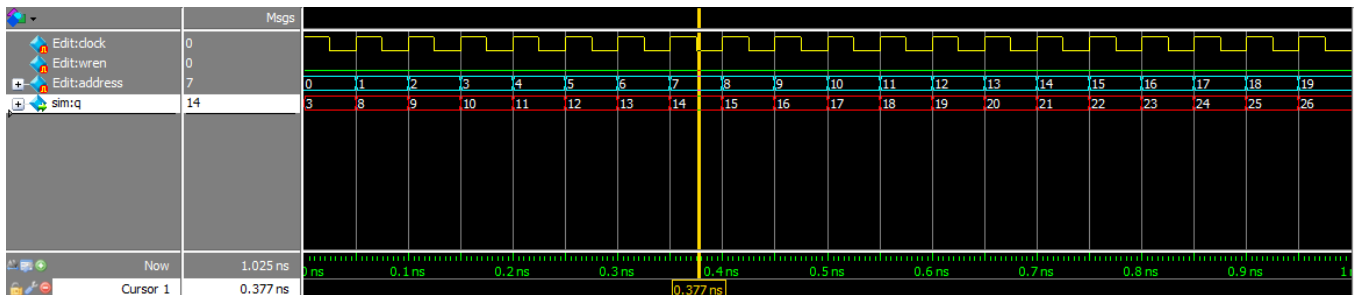
Para a simulação, geramos ondas para cada input

- Clock: 100 ps;
- Wren: constant 0;
- Address: counter;

Algoritmo implementando em Verilog HDL

```
module parte2 (address, data, clock, wren, q);  
  
    input    [4:0]  address;  
    input    clock, wren;  
    input [7:0]  data;  
    output [7:0]  q;  
  
    Ramlpm RAMramlpm(address, clock, data, wren, q);  
  
endmodule
```

Simulação no ModelSim



Como podemos perceber pela simulação, nas duas primeiras posições da memória é feita a escrita das posições da chamada, e nas outras posições, é salvo valores maiores que 8.