



Universidade Estadual de Campinas

## **LABORATÓRIO 04 – Uso da memória RAM com LPM**

Isabela Caroline de Sousa	218071
José Victor Machuca	219118
Matheus Santos Sano	222370

Para esse laboratório utilizamos a memória RAM M9K 32 x 8 bits com um único clock, contida no IP Catalog. Importada a memória, criamos um arquivo contendo dados, no formato hexadecimal, a serem armazenados nas 32 posições da memória, de modo a inicializá-la. A seguir são mostrados os dados que inserimos na memória:

```
:0100000000FF
:0100010001FD
:0100020002FB
:0100030003F9
:0100040004F7
:0100050005F5
:0100060006F3
:0100070007F1
:0100080008EF
:0100090009ED
:01000A000AEB
:01000B000BE9
:01000C000CE7
:01000D000DE5
:01000E000EE3
:01000F000FE1
:0100100010DF
:0100110011DD
:0100120012DB
:0100130013D9
:0100140014D7
:0100150015D5
:0100160016D3
:0100170017D1
:0100180018CF
:0100190019CD
:01001A001ACB
:01001B001BC9
:01001C001CC7
:01001D001DC5
:01001E001EC3
:01001F001FC1
:00000001FF
```

Para verificar o funcionamento da memória, implementamos um testbench. O primeiro teste consiste em fazer a leitura de todas as posições inicializadas, validando se estão corretas, em seguida fizemos um segundo teste que consiste em reescrever os conteúdos da memória, de modo que os valores das últimas posições sejam inseridos nas primeiras posições e vice-versa, e reler os novos valores.

A seguir, o código do testbench na íntegra:

```

1  library ieee;
2  USE ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4  use ieee.math_real.all;
5
6  entity test_memoria is end;
7
8  architecture dut of test_memoria is
9
10     CONSTANT half_period : time := 10 ps;
11
12     COMPONENT ram1pm IS
13     PORT (
14         address : IN STD_LOGIC_VECTOR(4 downto 0);
15         clock    : IN STD_LOGIC;
16         data     : IN STD_LOGIC_VECTOR(7 downto 0);
17         wren     : IN STD_LOGIC;
18         q        : OUT STD_LOGIC_VECTOR(7 downto 0)
19     );
20     END COMPONENT;
21
22     signal clk, simulando : STD_LOGIC := '0';
23     signal wren : STD_LOGIC := '0';
24     signal address : STD_LOGIC_VECTOR(4 downto 0);
25     signal data, q : STD_LOGIC_VECTOR(7 downto 0);
26
27 BEGIN
28     clk <= (simulando and not clk) after half_period;
29
30     DUT: ram1pm port map (address, clk, data, wren, q);
31
32     st: process is
33     variable Q_RTL : integer := 0;
34
35     BEGIN
36         assert false report "BOT" severity note; --inicio do teste
37         simulando <= '1'; -- simulacao começou
38         wait until rising_edge(clk);
39
40
41         -- TESTE LEITURA
42
43         wren <= '0';
44
45         FOR I IN 0 TO 31 LOOP
46             wait until falling_edge(clk);
47
48             address <= std_logic_vector(to_unsigned(I,5));
49             Q_RTL := I;
50
51             wait until rising_edge(clk);
52             wait until falling_edge(clk);
53
54             wait until rising_edge(clk);
55             assert Q_RTL = to_integer(signed(q)) report "Erro na leitura" severity failure;
56
57         END LOOP;
58
59         -- TESTE ESCRITA
60
61         wait until rising_edge(clk);
62
63         wren <= '1';
64
65         wait until falling_edge(clk);
66
67         FOR I IN 0 TO 31 LOOP
68             wait until falling_edge(clk);
69
70             address <= std_logic_vector(to_unsigned(I,5));
71             data <= std_logic_vector(to_unsigned(31-I,8));
72             Q_RTL := 31-I;
73
74             wait until rising_edge(clk);
75             wait until falling_edge(clk);
76
77             wait until rising_edge(clk);
78             assert Q_RTL = to_integer(signed(q)) report "Erro na escrita" severity failure;
79
80         END LOOP;
81
82         assert false report "EOT" severity note; --fim do teste
83         simulando <= '0'; --simulação parou
84         wait;
85     END process;
86 END DUT;

```

## ERROS E CORREÇÕES

Um dos erros neste laboratório foi na sintaxe do registro de fim de arquivo, ou seja, no último dos registros de dados que foram inseridos na memória (:00000001FF). O erro era bem simples, foi apenas a existência de um zero a mais na instrução.

O outro erro foi no código do testbench no momento de inverter os valores das últimas posições para as primeiras. Na execução do código, o testbench não fazia a inversão de forma adequada. Sendo assim, era apresentado o erro “Erro na escrita” no terminal. Para solucionar esse problema invertemos os dados dentro do loop (linha 69) pela subtração “31 - I”, sendo I identificando o endereço e a subtração 31 - I o dado. Dessa forma a lógica de inverter os dados funcionou corretamente.

