



Laboratório de Arquitetura e Organização de computadores II

Prática 1

Alunos: Anélio Gonçalves Caldas e Ryan Eduardo Mansur Vasconcelos

1. Introdução

A prática I introduz o uso da biblioteca LPM, uma biblioteca bastante útil existente no Quartus. Para esse exercício foi utilizado um módulo de memória RAM com 1 porta, configurada com 32 de espaço com 8 bits cada.

2. Objetivos

O objetivo desta prática é entender melhor sobre o uso da biblioteca LPM, com aplicações de conteúdos vistos em aula.

3. Desenvolvimento

Para parte 1 e 2, foi utilizado o código abaixo para simulação do display de 7-segmentos da FPGA EP2C35F672C6.

```
always@(*)
begin
    case(in)
        4'b0000: S = ~(7'b11111110);
        4'b0001: S = ~(7'b01100000);
        4'b0010: S = ~(7'b1101101);
        4'b0011: S = ~(7'b11111001);
        4'b0100: S = ~(7'b0110011);
        4'b0101: S = ~(7'b1011011);
        4'b0110: S = ~(7'b1011111);
        4'b0111: S = ~(7'b11100000);
        4'b1000: S = ~(7'b11111111);
        4'b1001: S = ~(7'b11111011);

        default: S = ~(7'b00000000);
    endcase
end
```

Ambas as partes também utilizaram o módulo principal (top level) e a biblioteca LPM(ramlpm) parecidas, como segue abaixo.

```
1 module toplevel(SW,KEY,LEDR,LEDG,HEX0, HEX1, HEX2, HEX3, HEX4, HEX5, HEX6, HEX7);
2
3     input [17:0]SW;
4     output reg [17:0]LEDR;
5     output reg [7:0]LEDG;
6     input [3:0]KEY;
7     output [0:6] HEX0, HEX1, HEX2, HEX3, HEX4, HEX5, HEX6, HEX7;
8     wire [7:0]q;
9
10    integer i;
11
12    always @(SW or KEY)
13    begin
14        for(i=0;i<12;i = i+1)
15        begin
16            if(SW[i])
17                LEDR[i] <= 1'b1;
18            end
19            case(KEY)
20                4'b0001: LEDG[0] <= 1'b1;
21                default: LEDG[0] <= 1'b0;
22            endcase;
23        end
24    end
```

```

25 ram_lpm ram(
26   SW[12:8], //address,
27   SW[17], //clock,
28   SW[7:0], //data,
29   KEY[0], //wren,
30   q) //q
31
32 //Mostrando dados de entrada
33 decodificador_c h_address(SW[12], HEX7[0:6]);
34 decodificador_c h2_address(SW[11:8], HEX6[0:6]);
35 decodificador_c h_data(SW[7:4], HEX5[0:6]);
36 decodificador_c h2_data(SW[3:0], HEX4[0:6]);
37
38 //Mostrando saída
39 decodificador_c h_q(q[7:4], HEX1[0:6]);
40 decodificador_c h2_q(q[3:0], HEX0[0:6]);
41
42
43 endmodule

```

- Parte 1

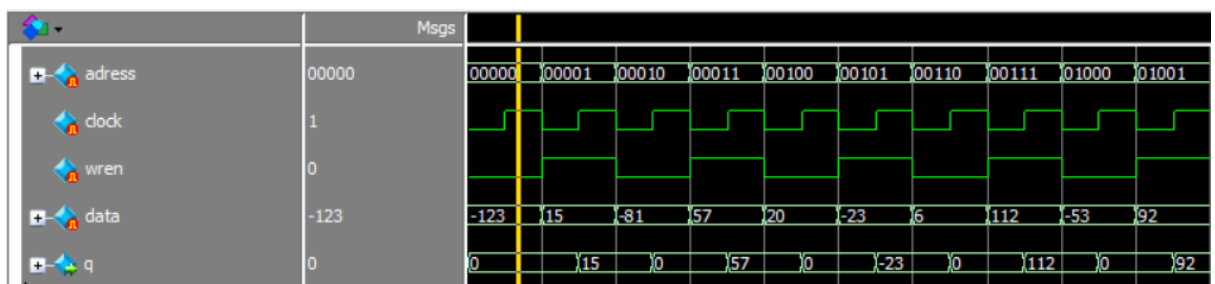


Imagem 1: Endereço 0 com wren desativado.

Como podemos ver na imagem 1 da simulação, o wren (write enable) está desativado durante o tempo que estávamos no endereço 00000, sendo assim nada foi escrito, pois para realizar a escrita, primeiramente é necessário que o wren esteja ativado e após isso precisamos de um clock de borda de subida, com isso o valor contido em data seria escrito no endereço desejado.

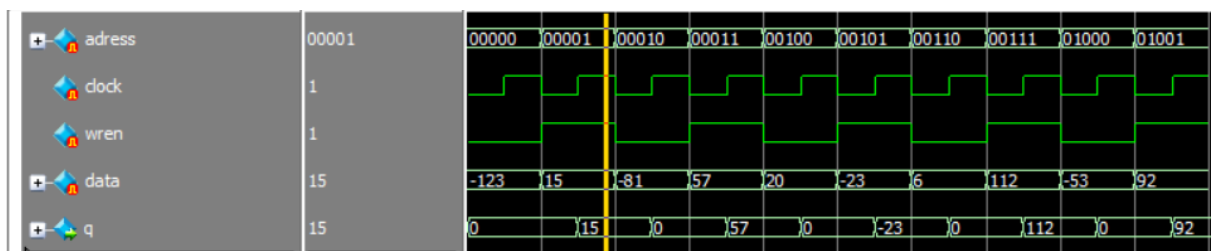


Imagem 2: Primeira escrita é realizada.

Já na imagem 2, podemos ver que quando estamos no endereço 00001, o wren está ativado, sendo assim, podemos realizar de fato a escrita do conteúdo de data no endereço. Sendo assim, quando houve o primeiro clock de borda de subida, o valor 15 foi escrito no endereço 00001, como podemos ver pela imagem. Além disso, é possível notar também, que a leitura do conteúdo que está no endereço 00010 só ocorre quando houve um clock de borda de subida também.

• Parte 2

Para a parte II, a memória foi inicializada com um arquivo mif, de acordo com a tabela abaixo.

Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	1	17	18	19	20	21	22	23
8	24	25	26	27	28	29	30	31
16	32	33	34	35	36	37	38	39
24	40	41	42	43	44	45	46	47

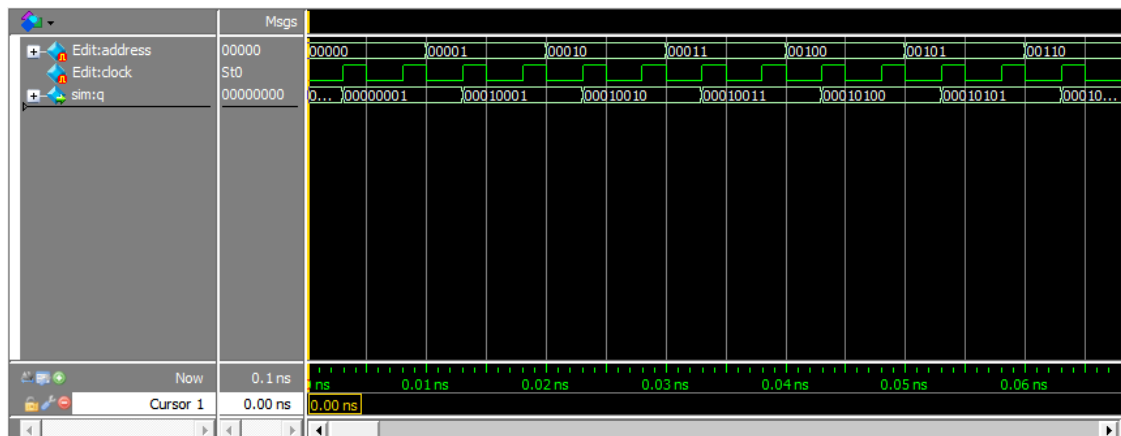


Imagem 3: Leitura dos dados presentes na memória.

Podemos ver na imagem, que no endereço 1, o dado presente nele é o 01, número de chamada do primeiro integrante do grupo, e quando está no endereço 2, o dado presente nele é o 10010 (17) que é o número de chamada do segundo integrante do grupo. Nos próximos endereços, estão presentes os números sequenciais após o 17.

4. Conclusão

Após a execução desta prática, podemos assimilar melhor o como funciona a biblioteca LPM, assim como, aprimorar conhecimentos em verilog, e na utilização da FPGA EP2C35F672C6.