

Memórias ROM e RAM em Verilog

Aula de Exercícios

Bruno Albertini, Edson Gomi e Ricardo Lera

Considere as seguintes descrições de memórias ROM e RAM:

ROM de 16×16 :

```
1 module rom_16(out, addr, CS);
2   input [3:0] addr;
3   input CS;
4   output reg [15:0] out;
5   reg [15:0] data[15:0];
6   integer i;
7   initial
8   begin
9     for (i = 0; i < 15; i++)
10       data[i] = i[15:0];
11     data[15] = 16'h69;
12   end
13   always @(addr, CS)
14     out = data[addr];
15 endmodule
```

RAM de 4×16 :

```
1 module ram_4(out, in, addr, RW, CS);
2   input [15:0] in;
3   input [1:0] addr;
4   input RW, CS;
5   output reg [15:0] out;
6   reg [15:0] data[3:0];
7   integer i;
8   initial
9   for (i = 0; i < 4; i++)
10     data[i] = 16'b0;
11   always @(addr, CS, RW)
12     if (RW == 1'b0)
13       out = data[addr];
14     else
15       if (RW == 1'b1)
16         data[addr] = in;
17       else
18         out = 16'bz;
19 endmodule
```

Parte I

Projete um circuito chamado `romram1.v` que realiza as seguintes operações:

- Leia o valor 0x69 da ROM;
- Escreva e leia o valor lido no endereço 0x3 da RAM.

O módulo `romram1` deverá ser

```
module romram1 (out, done)
```

Onde `out` é o valor lido da RAM e `done` indica que o seu circuito realizou as operações solicitadas.

Parte II

Projete um circuito chamado `romram2.v` que realize as seguintes operações:

- Escolha um único endereço da RAM;
- Coloque neste endereço a somatória dos valores dos endereços 0 a 14 da ROM;
- Compare o valor resultante com o valor do endereço 15 da ROM.

O módulo romram2 deverá ser:

romram2 (soma, comp, done)

Onde soma é o valor da somatória, com é o resultado da comparação (1 se for igual e 0 se for diferente) e done indica que o seu circuito executou as operações solicitadas.