

Endereçamento de Memórias

Aula de Exercícios

Bruno Albertini, Edson Gomi e Ricardo Lera

Considere os seguintes módulos ROM e RAM (disponíveis também no edisciplinas):

ROM de 16 endereços:

```

1 module rom_16(addr, CS, OE, out);
2   input [3:0] addr;
3   input CS, OE;
4   output reg [15:0] out;
5   reg [15:0] data[15:0];
6   initial
7     for (integer i = 0; i <
16; i++)
8       data[i] = ~i[15:0];
9   always @(addr, CS, OE)
10    if (OE == 1'b1)
11      out = data[addr];
12    else
13      out = 16'bz;
14 endmodule

```

RAM de 4 endereços:

```

1 module ram_4(in, addr, RW, CS, OE, out);
2   input [15:0] in;
3   input [1:0] addr;
4   input RW, CS, OE;
5   output reg [15:0] out;
6   reg [15:0] data[3:0];
7   always @(addr, CS, OE, RW)
8   begin
9     if (RW == 1'b0 & OE == 1'b1)
10      out = data[addr];
11    else
12      out = 16'bz;
13    if (RW == 1'b1)
14      data[addr] = in;
15  end
16 endmodule

```

Utilizando dois módulos ram_4, projete o seguinte módulo:

```
module ram_8 (in, addr, RW, CS, OE, out);
```

Utilize alta impedância (16'bz) nas saídas dos módulos que não estão sendo selecionados.

Então, utilizando os módulos anteriores, projete uma memória de 64 endereços de acordo com as seguintes especificações:

```
module memchip_64 (in, addr, RW, out);
```

addr	Bloco	Tamanho (bits)
0x00 a 0x0F	rom_16	16 x 16
0x10 a 0x17	ram_8	08 x 16
0x17 a 0x27	vazio	16 x 16
0x28 a 0x2F	ram_8	08 x 16
0x30 a 0x3F	vazio	16 x 16

Se o módulo receber um valor de addr correspondente a um bloco vazio, a saída (out) deve receber **alta impedância** (16'bz).

Envie somente um arquivo contendo todos os módulos, inclusive as memórias fornecidas.