基本数字逻辑单元HDL描述

主讲：何宾

Email：hebin@mail.buct.edu.cn

2018.08

组合逻辑电路的HDL描述

任何复杂的数字系统可以用若干基本组合逻辑单元和时序逻辑单元组合来实现。

基本逻辑单元一般分为组合逻辑电路和时序逻辑电路两大类。这两类基本逻辑电路构成和复杂数字系统设计的基石。

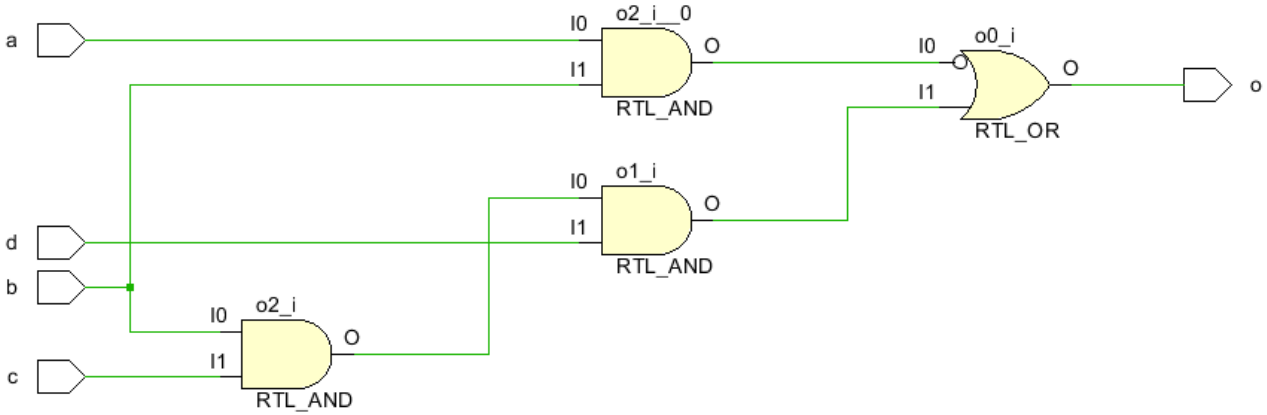
组合逻辑电路的HDL描述
--内容

组合逻辑电路是指输出状态只决定于同一时刻各个输入状态的组合，而与先前状态无关的逻辑电路称为组合逻辑电路。组合逻辑电路主要包括:

- 基本逻辑门
- 编码器
- 译码器
- 数据选择器
- 数据比较器
- 总线缓冲器

逻辑门的HDL描述
--基本门电路过程分配描述

```
module g1(o,a,b,c,d);
input a,b,c,d;
output reg o;
always @(a or b or c or d)
begin
      o=(~(a&b))|(b&c&d);
end
endmodule
```



本设计保存在本书配套资源**eda_verilog\example6_1**目录下

逻辑门的HDL描述
--基本门电路连续分配描述

```
module g2(o,a,b,c,d);
input a,b,c,d;
output o;
    assign o=(~(a&b))|(b&c&d);
endmodule
```

逻辑门的HDL描述
--基本门电路门调用描述

```
module g3(o,a,b,c,d);
input a,b,c,d;
output o;
    nand(o1,a,b);
    and(o2,b,c,d);
    or(o,o1,o2);
endmodule
```

组合逻辑电路的HDL描述
--编码器HDL描述

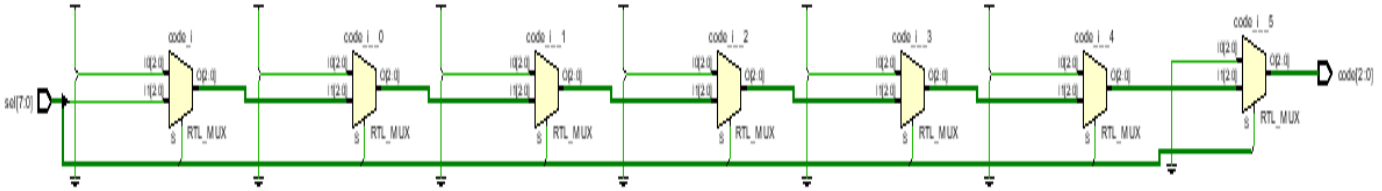
将某一信息用一组按一定规律排列的二进制代码描述称为编码。

- 典型的有8421码、BCD码等。

在使用HDL语言描述编码器时，通过使用CASE和IF语句实现对编码器的描述。

编码器HDL描述
--8/3优先编码器描述的例子

```
module v_priority_encoder_1(sel,code);
input [7:0] sel;
output [2:0] code;
reg [2:0] code;
always @(sel)
begin
    if (sel[0]) code = 3'b000;
    else if (sel[1]) code = 3'b001;
    else if (sel[2]) code = 3'b010;
    else if (sel[3]) code = 3'b011;
    else if (sel[4]) code = 3‘b100;
    else if (sel[5]) code = 3'b101;
    else if (sel[6]) code = 3'b110;
    else if (sel[7]) code = 3'b111;
    else code = 3'bxxx;
  end
endmodule
```



思考与练习1：优先级的含义？

思考与练习2：查看详细描述后的结果，分析

（提示：长链路带来延迟，性能降低）

本设计保存在本书配套资源eda_verilog\example6_2目录下

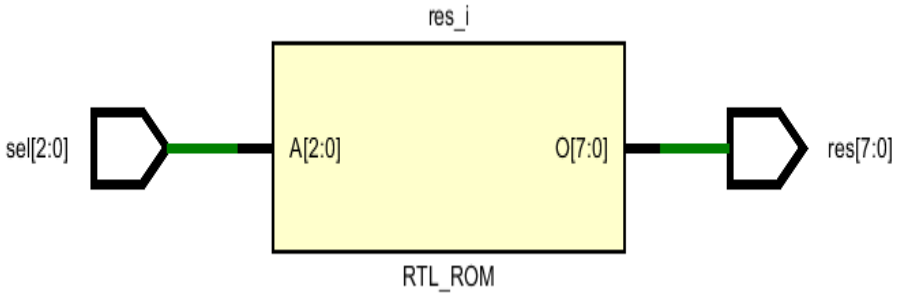
组合逻辑电路的HDL描述
--编码器HDL描述

译码器设计

- 译码的过程实际上就是编码过程的逆过程,即将一组按一定规律排列的二进制数还原为原始的信息。

译码器HDL描述
--3:8译码器描述的例子

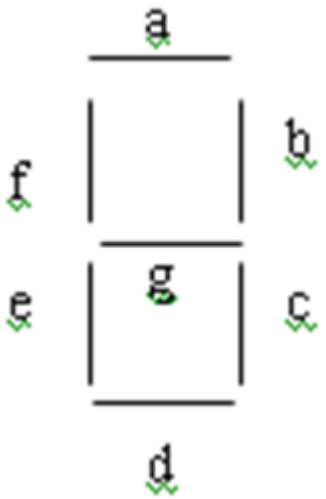
```
module v_decoders_1 (sel, res);
input [2:0] sel;
output [7:0] res;
reg [7:0] res;
always @(sel or res)
begin
   case (sel)
      3'b000 : res = 8'b00000001;
      3'b001 : res = 8'b00000010;
      3'b010 : res = 8'b00000100;
      3'b011 : res = 8'b00001000;
      3'b100 : res = 8'b00010000;
      3'b101 : res = 8'b00100000;
      3'b110 : res = 8'b01000000;
     default : res = 8'b10000000;
   endcase
end
endmodule
```

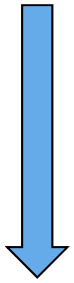


本设计保存在本书配套资源**eda_verilog\example6_3**目录下

译码器HDL描述
--十六进制共阳极7段数码管描述的例子

```
module seven_segment_led(o,i);
input[3:0] i; output reg[6:0] o;
always @(i)
begin
    case (i)
        4'b0001 : o = 7'b1111001;   // 1
        4'b0010 : o = 7'b0100100;   // 2
        4'b0011 : o = 7'b0110000;   // 3
        4'b0100 : o = 7'b0011001;   // 4
        4'b0101 : o = 7'b0010010;   // 5
        4'b0110 : o = 7'b0000010;   // 6
```

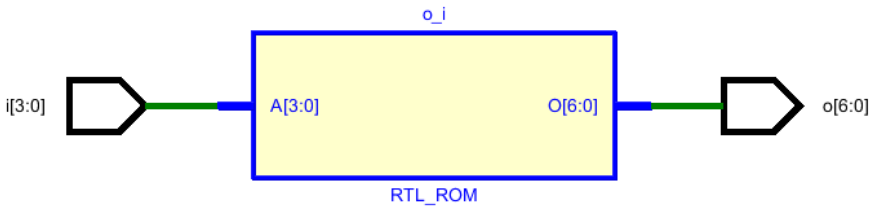




（下页继续）

译码器HDL描述
--十六进制共阳极7段数码管描述的例子

```
        4'b0111 : o = 7'b1111000;   // 7
        4'b1000 : o = 7'b0000000;   // 8
        4'b1001 : o = 7'b0010000;   // 9
        4'b1010 : o = 7'b0001000;   // A
        4'b1011 : o = 7'b0000011;   // b
        4'b1100 : o = 7'b1000110;   // C
        4'b1101 : o = 7'b0100001;   // d
        4'b1110 : o = 7'b0000110;   // E
        4'b1111 : o = 7'b0001110;   // F
        default : o = 7'b1000000;   // 0
    endcase
end
endmodule
```



Cell Properties

o_i

INIT	Value
INIT_DEFAULT	7'b1000000
INIT_1	7'b1111001
INIT_2	7'b0100100
INIT_3	7'b0110000
INIT_4	7'b0011001
INIT_5	7'b0010010
INIT_6	7'b0000010
INIT_7	7'b1111000
INIT_8	7'b0000000
INIT_9	7'b0010000
INIT_10	7'b0001000
INIT_11	7'b0000011
INIT_12	7'b1000110
INIT_13	7'b0100001
INIT_14	7'b0000110
INIT_15	7'b0001110

本设计保存在本书配套资源**eda_verilog\example6_4**目录下

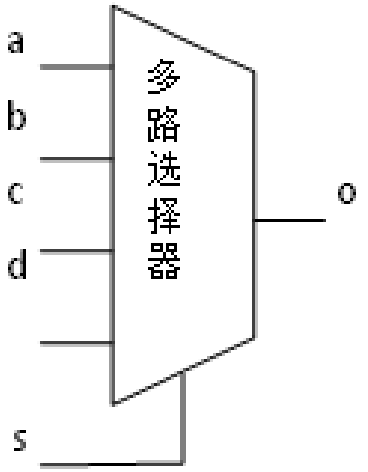
组合逻辑电路的HDL描述
--数据选择器HDL描述

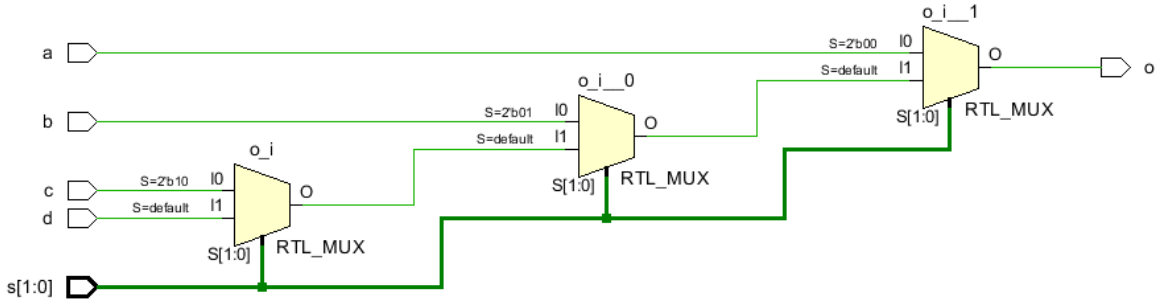
在数字系统中，经常需要把多个不同通道的信号发送到公共的信号通道上，通过数据选择器可以完成这一功能。

- 在数字系统设计中，常使用CASE和IF-ELSE语句描述数据选择器。

数据选择器HDL描述
--使用if-else语句描述4：1多路选择器

```
module v_multiplexers_1 (a, b, c, d, s, o);
input a,b,c,d;
input [1:0] s;
output reg o;
always @(a or b or c or d or s)
begin
    if (s == 2'b00) o = a;
    else if (s == 2'b01) o = b;
    else if (s == 2'b10) o = c;
    else o = d;
end
endmodule
```





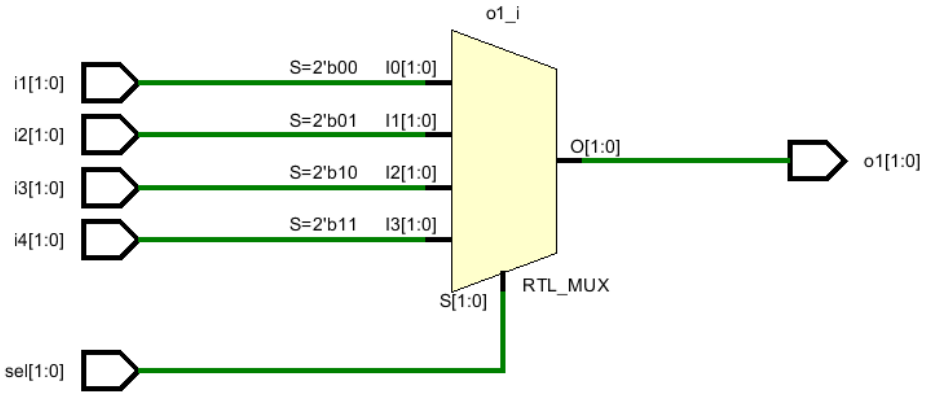
思考与练习1：这是一种优先级结构,为什么？

思考与练习2：这种描述产生长延迟路径。

本设计保存在本书配套资源**eda_verilog\example6_5_1**目录下

数据选择器HDL描述
--使用case语句描述4：1多路选择器

```
module full_mux (sel, i1, i2, i3, i4, o1);
input [1:0] sel;
input [1:0] i1, i2, i3, i4;
output [1:0] o1;
reg [1:0] o1;
always @(sel or i1 or i2 or i3 or i4)
begin
  case (sel)
      2'b00: o1 = i1;
      2'b01: o1 = i2;
      2'b10: o1 = i3;
      2'b11: o1 = i4;
  endcase
end
endmodule
```

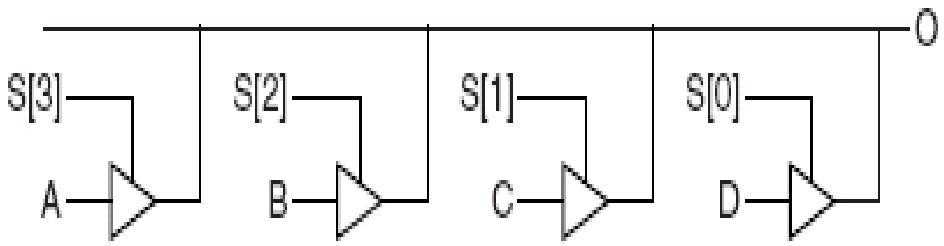


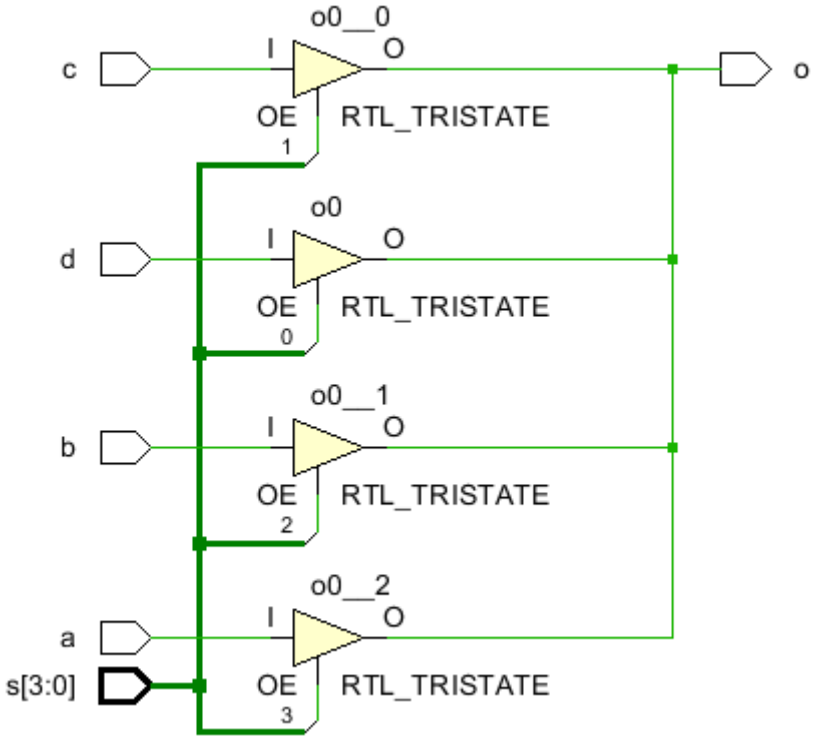
思考与练习3：比较if和case描述生成结构的区别

本设计保存在本书配套资源eda_verilog\example6_5_2目录下

数据选择器HDL描述
--使用三态缓冲实现4：1多路选择

```
module v_multiplexers_3 (a, b, c, d, s, o);
input a,b,c,d;
input [3:0] s;
output o;
    assign o = s[3] ? a :1'bz;
    assign o = s[2] ? b :1'bz;
    assign o = s[1] ? c :1'bz;
    assign o = s[0] ? d :1'bz;
endmodule
```





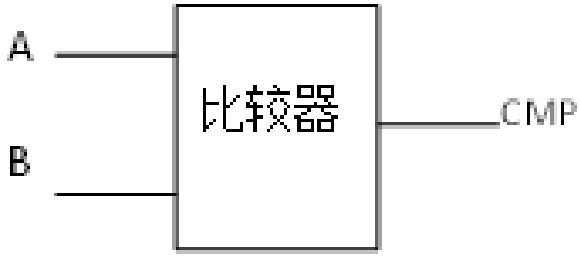
本设计保存在本书配套资源**eda_verilog\example6_6**目录下

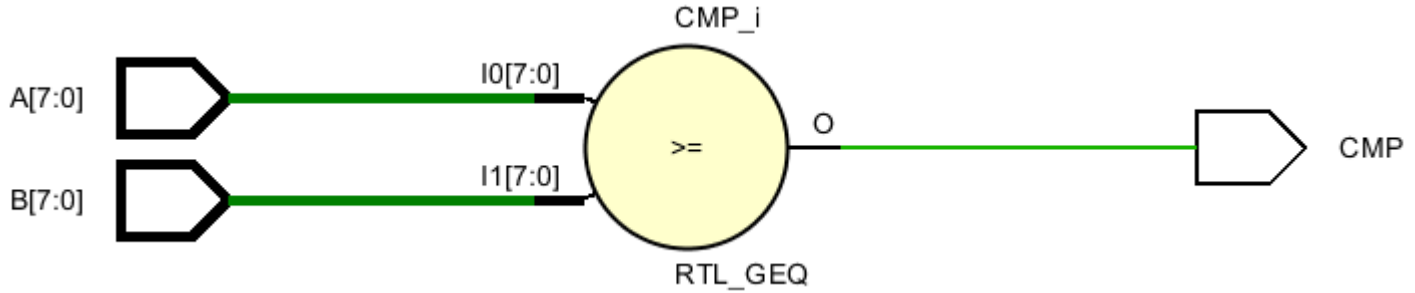
组合逻辑电路的HDL描述
--数字比较器HDL描述

比较器就是对输入数据进行比较，并判断其大小的逻辑电路。在数字系统中，比较器是基本的组合逻辑单元之一，比较器主要是使用关系运算符实现的。

组合逻辑电路的HDL描述
--数字比较器HDL描述的例子

```
module v_comparator_1 (A, B, CMP);

input [7:0] A;

input [7:0] B;

output CMP;

    assign CMP = (A >= B) ? 1'b1 : 1'b0;

endmodule
```





本设计保存在本书配套资源**eda_verilog\example6_7**目录下

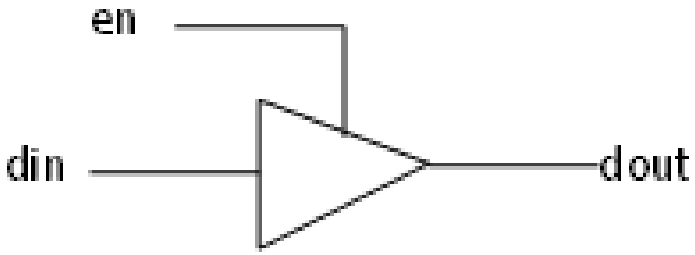
组合逻辑电路的HDL描述
--总线缓冲器HDL描述

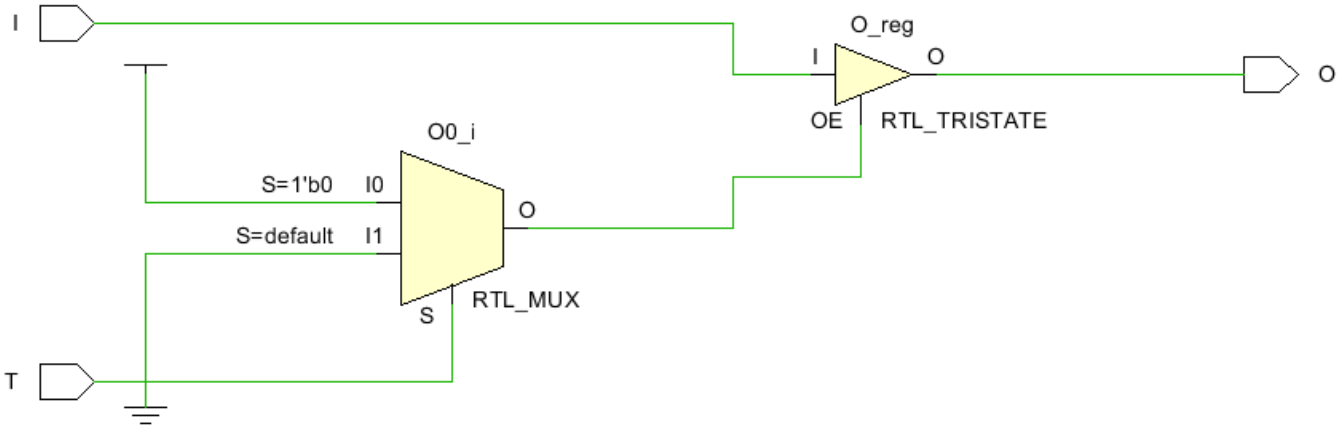
总线是一组相关信号的集合。

- **在计算机系统常用的总线有数据总线、地址总线和控制总线。**
- **由于总线上经常需要连接很多的设备，因此必须正确的控制总线的输入和输出，这样才不会产生总线访问的冲突。**

总线缓冲器HDL描述
--三态缓冲器过程分配描述的例子

```
module v_three_st_1 (T, I, O);

input T, I;

output reg O;

always @(T or I)

begin

    if (~T) O = I;

    else O = 1'bZ;

end

endmodule
```

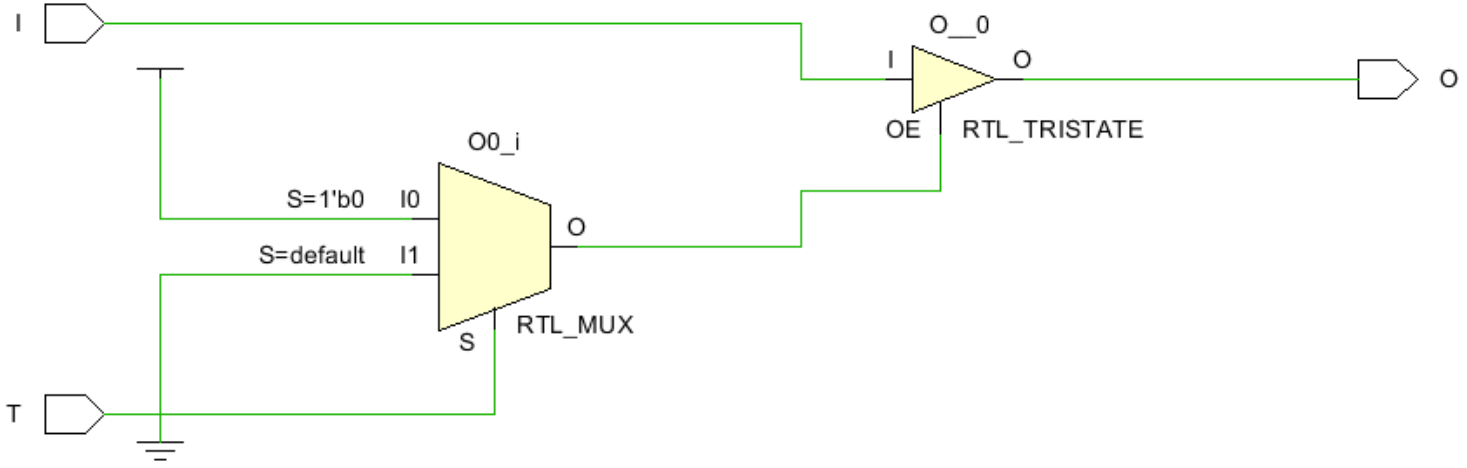




本设计保存在本书配套资源**eda_verilog\example6_8_1**目录下

总线缓冲器HDL描述
--三态缓冲器连续分配描述的例子

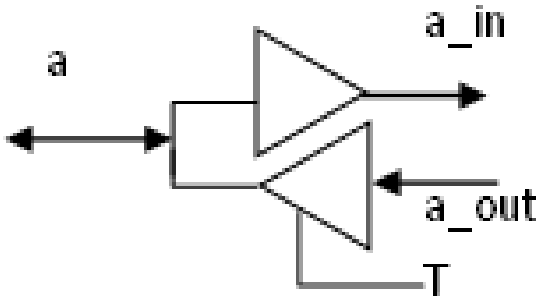
```
module v_three_st_2 (T, I, O);

input T, I;

output O;

      assign O = (~T) ? I: 1'bZ;

endmodule
```

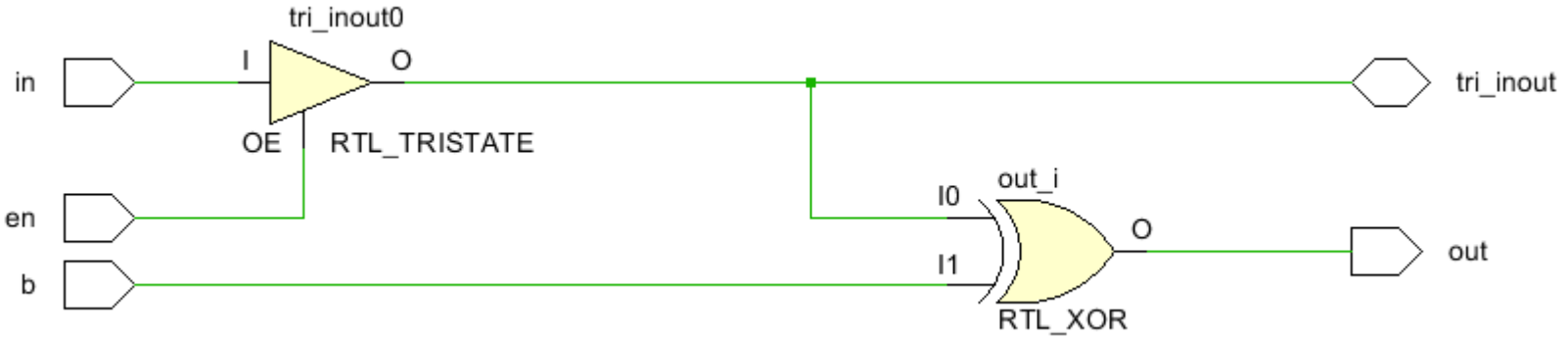


本设计保存在本书配套资源**eda_verilog\example6_8_2**目录下

总线缓冲器HDL描述
--双向总线缓冲区描述的例子

```
module bidir(tri_inout,out,in,en,b);
inout tri_inout;
output out;
input in,en,b;
    assign tri_inout = en ? in : 'bz;
    assign out = tri_inout ^ b;
endmodule
```





本设计保存在本书配套资源**eda_verilog\example6_9**目录下