

## 2023FPGA 国赛能力测试题

注：题目二选一即可，所有题目只验收仿真波形。

### 一、电梯控制电路

设计一个电梯控制电路，要求：

1) 大楼总共有 40 层，初始楼层为 1 层，当目标楼层符合条件，且与当前楼层不同时，电梯向目标楼层行进。为方便仿真，每个经过一个 clk，电梯经过一个楼层；

2) 电梯只有目标楼层按键和紧急按键。不考虑同时按下多个目标楼层按键的情况。

3) 2-9 层为未开发区域，正常情况下不能设定为目标楼层，但是在紧急情况下，按下 2-9 的楼层号后，如果检测到紧急按键 key 的上升沿，此时可以到达这些楼层；

4) 电梯限定只能承载 13 人、且不超过 1000Kg，人数或重量超标时均发出警报信号。

- 输入信号：目标楼层 H[7:0]、紧急按键 key、当前人数 num[3:0]、重量[9:0](KG)；
- 输出信号：当前楼层 N[7:0]，警报信号 alert。

程序头：

```
module elevator#(parameter floor = 40)(
    input clk, rst_n,
    input key, [7:0]H,[3:0]num, [9:0]weight,
    output reg [7:0]N,
    output alert
);
endmodule
```

#### ● Benchmark

```
module elevator_tb(
);
reg clk, rst_n, key;
reg [7:0]H;
reg [3:0]num;
reg [9:0]weight;
wire [7:0]N;
wire alert;

elevator test(
    .clk(clk), .rst_n(rst_n),
    .key(key), .H(H), .num(num), .weight(weight),
    .N(N),
    .alert(alert)
);
initial begin
    clk = 0;
    rst_n = 0;
    #15
```

```
        rst_n = 1;
        forever
            #5 clk = ~clk;
    end
    initial begin
        key = 0;
        H = 0;
        num = 0;
        weight = 0;
        #20
        H = 30;
        num = 1;
        weight = 50;
        #400
        H = 8;
        #10
        key = 1;
        @(negedge clk);key = 0;
        #300
        H = 12;
        num = 14;
        #20
        num = 10;
        weight = 1001;
        #50
        weight = 900;
    end
endmodule
```

## 二、非重叠序列检测

设计一个串行序列检测器，用来检测序列 011010，要求：

- 1、以每六个输入为一组，不检测重复序列，例如第一位数据不符合，则不考虑后五位。一直到第七位数据即下一组信号的第一位开始检测。
- 2、六位数据输入完成后，当信号满足该序列，在最后一位数据输入的下一拍指示信号 `match` 拉高一个周期，当不满足时，指示信号 `not_match` 拉高一个周期。

程序头：

```
module sequence_detect(  
    input clk,  
    input rst_n,  
    input data,  
    output reg match,  
    output reg not_match  
);  
endmodule
```

### ● Benchmark

```
module testbench();  
    reg clk, rst_n, D_in;  
    wire match, not_match;  
    reg [23:0] data;  
    initial begin  
        clk=0;  
        rst_n=0;  
        D_in=0;  
        data=24'b011100_011010_011110_011101;  
        #5;  
        rst_n=1;  
    end  
    always #10 clk<=~clk;  
    always@(posedge clk)begin  
        D_in <= data[23];  
        data <= {data[22:0],data[23]};  
    end  
    sequence_detect test(  
        .clk(clk),  
        .rst_n(rst_n),  
        .data(D_in),  
        .match(match),  
        .not_match(not_match)  
    );  
endmodule
```