数字电路实验 Lab 2 实验报告

奚项正 PB23000020

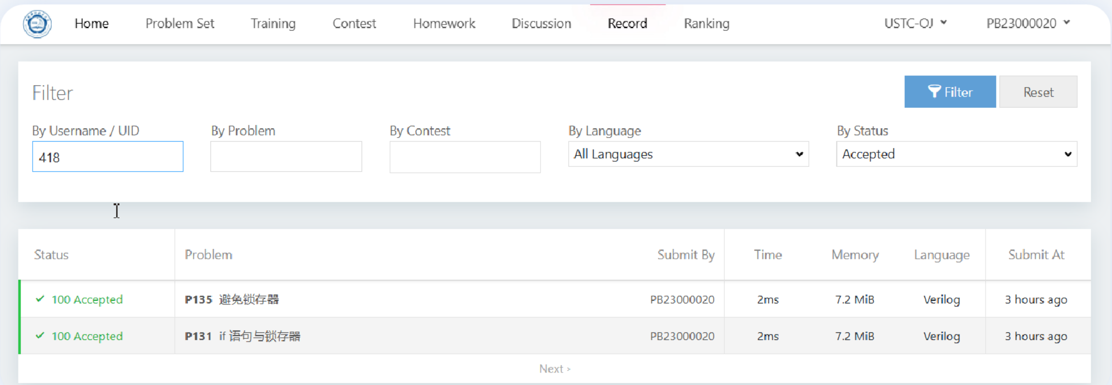
必做内容

题目 1：if 语句与锁存器（1 分）

模块代码：

```
module top_module (
    input                       cpu_overheated,
    output  reg                 shut_off_computer,
    input                       arrived,
    input                       gas_tank_empty,
    output  reg                 keep_driving
);
    // Edit the code below
    always @(*) begin
        if (cpu_overheated)
            shut_off_computer = 1'b1;
        else
            shut_off_computer = 1'b0;   //增加else语句，消除锁存器
    end

    always @(*) begin
        if (~arrived)
            keep_driving = ~gas_tank_empty;
        else
            keep_driving = 1'b0;    //增加else语句，消除锁存器
    end
endmodule
```

> **注：** 本题已在 OJ 系统上提交并通过



Filter

By Username / UID: 418

By Problem:

By Contest:

By Language: All Languages

By Status: Accepted

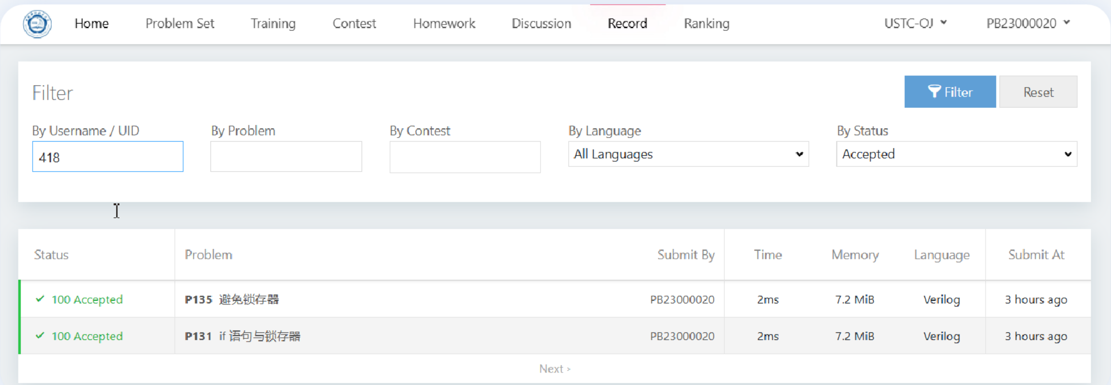
Status	Problem	Submit By	Time	Memory	Language	Submit At
✓ 100 Accepted	**P135** 避免锁存器	PB23000020	2ms	7.2 MiB	Verilog	3 hours ago
✓ 100 Accepted	**P131** if 语句与锁存器	PB23000020	2ms	7.2 MiB	Verilog	3 hours ago

题目 2：case 语句与锁存器（1 分）

模块代码：

```
module top_module (
    input       [15:0]                  scancode,
    output reg                          left,
    output reg                          down,
    output reg                          right,
    output reg                          up
);
// Write your codes here.
    always @(*) begin
        up = 1'b0;
        down = 1'b0;
        left = 1'b0;
        right = 1'b0;   //赋初值，避免锁存器

        case (scancode) //根据识别码建立映射
            16'hE06B:   left = 1'b1;
            16'hE072:   down = 1'b1;
            16'hE074:   right = 1'b1;
            16'hE075:   up = 1'b1;
            default:    ;
        endcase
    end
endmodule
```

> **注：**本题已在 OJ 系统上提交并通过



Filter

By Username / UID: 418

By Problem:

By Contest:

By Language: All Languages

By Status: Accepted

Status	Problem	Submit By	Time	Memory	Language	Submit At
✓ 100 Accepted	**P135** 避免锁存器	PB23000020	2ms	7.2 MiB	Verilog	3 hours ago
✓ 100 Accepted	**P131** if 语句与锁存器	PB23000020	2ms	7.2 MiB	Verilog	3 hours ago

题目 3：计数器 Pro（3 分）

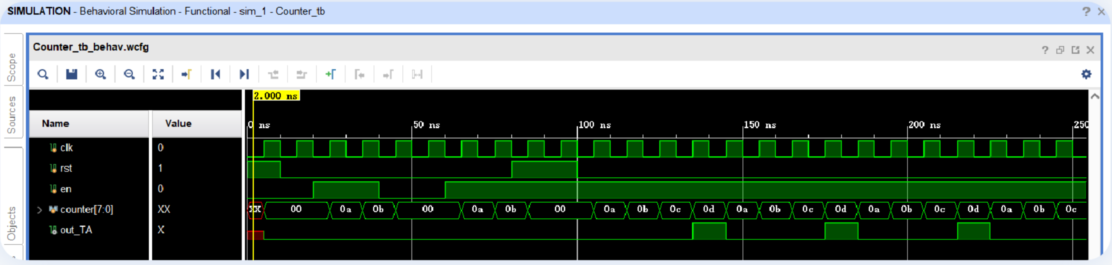
模块代码：

```
module Counter #(
    parameter               MAX_VALUE = 8'd13,
    parameter               MIN_VALUE = 8'd10
)(
    input                   clk,
    input                   rst,
    input                   enable, //增加enable信号
    output                  out
);

reg [7:0] counter;
always @(posedge clk) begin
    if (rst)
        counter ⩽ 0;   //此markdown主题自动将 < = 显示为 ⩽ ，下同
    else begin  //rst信号优先级高于enable
        if (enable) begin
            if (counter ⩾ MAX_VALUE)
                counter ⩽ MIN_VALUE;   //达到MAX_VALUE后跳转回MIN_VALUE
            else if(counter ⩾ MIN_VALUE)
                counter ⩽ counter + 1; //在范围内正常工作时每次+1
            else counter ⩽ MIN_VALUE;  //避免锁存器，确保在指定范围内工作
        end
        else
            counter ⩽ 0;   //enable低电平时counter复位
    end
end

assign out = (counter == MAX_VALUE);
endmodule
```

仿真文件：

```
module Counter_tb();     //此仿真文件为实验文档提供，不予注释
reg clk, rst, en;
wire out_TA;
initial begin
    clk = 0; rst = 1; en = 0;
    #10;
    rst = 0;
    #10;
    en = 1;
    #20;
    en = 0;
    #20;
    en = 1;
    #20;
    rst = 1;
    #20;
    rst = 0;
    #200;
    en = 0;
end
always #5 clk = ~clk;
Counter #(
    .MIN_VALUE(8'd10),
    .MAX_VALUE(8'd13)
) counter (
    .clk(clk),
    .rst(rst),
    .enable(en),
    .out(out_TA)
);
endmodule
```

附：仿真结果截图



SIMULATION - Behavioral Simulation - Functional - sim_1 - Counter_tb

Counter_tb_behav.wcfg

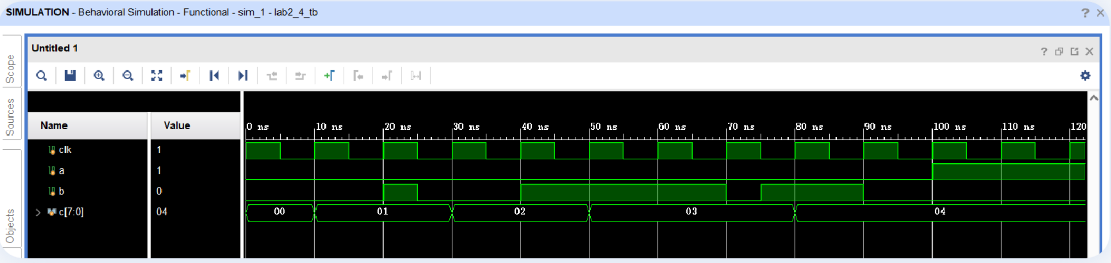
Name	Value
clk	0
rst	1
en	0
counter[7:0]	XX
out_TA	X

题目 4：生成波形（2 分）

仿真文件：

```
module lab2_4_tb();
reg clk,a,b;
reg [7:0] c;

initial begin
    c = 8'b0000_0000;
    clk = 1'b1;
    a = 1'b0;
    b = 1'b0;
    #10;

    c = 8'b0000_0001;
    #10;

    b = ~b;
    #5;

    b = ~b;
    #5;

    c = 8'b0000_0010;
    #10;

    b = ~b;
    #10;

    c = 8'b0000_0011;
    #20;

    b = ~b;
    #5;

    b=~b;
    #5;

    c = 8'b0000_0100;
    #10;

    b = ~b;
    #10;

    a = ~a;
end

always #5 clk = ~clk;   //时钟周期
endmodule
```

> **附：**仿真结果截图



选择性必做内容

题目 3：『众数』统计（3 分）

模块代码：

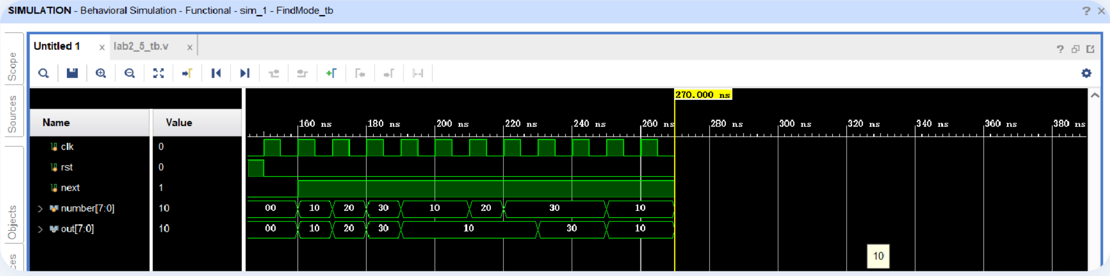
```
module FindMode (    //由于出现次数超过一半的数必定是出现次数最多的，而其他情形可以输出序列中任意一个数，从而本题只需找当前出现次数最多的数
    input                           clk,
    input                           rst,
    input                           next,
    input       [7:0]               number,
    output reg  [7:0]               out
);
// Your codes here.

reg [7:0] cnt[255:0];    //保存0-255每个数的出现次数
reg [7:0] tmp;  //存储出现最多次的数
integer i;

always @(posedge clk) begin
    if (rst) begin  //清除先前的记录
        tmp <= 0;
        for (i = 0; i <= 255 ; i = i + 1)
            cnt[i] <= 0;
    end
    else if (next) begin
        cnt[number] <= cnt[number] + 1; //该数出现次数+1
        if (cnt[number] >= cnt[tmp])    //判断其是否出现次数最多
            tmp <= number;
        else
            tmp <= tmp;
    end
end

always @(*) begin
    out = tmp;  //当前结果输出
end
```

```
endmodule
```

仿真文件：

```
module FindMode_tb; //本文件内容是平凡的，可参见仿
reg clk;
reg rst;
reg next;
reg [7:0] number;
wire [7:0] out;

FindMode dut (
    .clk(clk),
    .rst(rst),
    .next(next),
    .number(number),
    .out(out)
);

initial begin
    clk = 1;
    forever #5 clk = ~clk;
end

initial begin
    rst = 1;
    next = 0;
    number = 0;
    #10 rst = 0;
    #10 next = 1;
    number = 8'h10;
    #10 next = 1;
    number = 8'h20;
    #10 next = 1;
    number = 8'h30;
    #10 next = 1;
    number = 8'h10;
    #10 next = 1;
    number = 8'h10;
    #10 next = 1;
    number = 8'h20;
    #10 next = 1;
    number = 8'h30;
    #10 next = 1;
    number = 8'h30;
    #10 next = 1;
    number = 8'h30;
    #10 next = 1;
    number = 8'h10;
    #10 next = 1;
    number = 8'h10;
    #10 next = 1;
    number = 8'h10;
    #10
    rst = 1;
    next = 0;
    number = 0;
    #10 rst = 0;
    #10 next = 1;
    number = 8'h10;
    #10 next = 1;
    number = 8'h20;
    #10 next = 1;
    number = 8'h30;
    #10 next = 1;
    number = 8'h10;
    #10 next = 1;
    number = 8'h10;
    #10 next = 1;
    number = 8'h20;
    #10 next = 1;
    number = 8'h30;
    #10 next = 1;
    number = 8'h30;
    #10 next = 1;
    number = 8'h30;
    #10 next = 1;
    number = 8'h10;
    #10 next = 1;
    number = 8'h10;
    #10 next = 1;
    number = 8'h10;
    $finish;
end
endmodule
```

附：仿真结果截图



SIMULATION - Behavioral Simulation - Functional - sim_1 - FindMode_tb

Name	Value
clk	0
rst	0
next	1
number[7:0]	10
out[7:0]	10