

实验08

信号处理及有限状态机

2022/11/24

实验安排

数字电路实验关键时间节点

时间	实验发布（23：59前）	实验讲解（上课时间）	检查截止（21:30前）	报告提交截止（23:59前）
第四周 (09. 22)	实验01_Logisim入门			
第五周 (09. 29)	实验02_简单组合逻辑电路	实验01_Logisim入门		
第七周 (10. 13)	实验03_简单时序逻辑电路	实验02_简单组合逻辑电路	实验01_Logisim入门	
第八周 (10. 20)	实验04_Verilog硬件描述语言	实验03_简单时序逻辑电路	实验02_简单组合逻辑电路	实验01_Logisim入门
第九周 (10. 27)	实验05_使用Vivado进行仿真	实验04_Verilog硬件描述语言	实验03_简单时序逻辑电路	实验02_简单组合逻辑电路
第十周 (11. 03)	实验06_FPGA原理及Vivado综合	实验05_使用Vivado进行仿真	实验04_Verilog硬件描述语言	实验03_简单时序逻辑电路
十一周 (11. 10)	实验07_FPGA实验平台及IP核	实验06_FPGA原理及Vivado综合	实验05_使用Vivado进行仿真	实验04_Verilog硬件描述语言
十二周 (11. 17)	实验08_信号处理及有限状态机	实验07_FPGA实验平台及IP核	实验06_FPGA原理及Vivado综合	实验05_使用Vivado进行仿真
十三周 (11. 24)	实验09_竞争冒险及流水线技术	实验08_信号处理及有限状态机	实验07_FPGA实验平台及IP核	实验06_FPGA原理及Vivado综合
十四周 (12. 01)	实验10_综合实验	实验09_竞争冒险及流水线技术	实验08_信号处理及有限状态机	实验07_FPGA实验平台及IP核
十五周 (12. 08)		实验10_综合实验	实验09_竞争冒险及流水线技术	实验08_信号处理及有限状态机
十六周 (12. 15)			实验10_综合实验	实验09_竞争冒险及流水线技术
十七周 (12. 22)				实验10_综合实验

实验目的

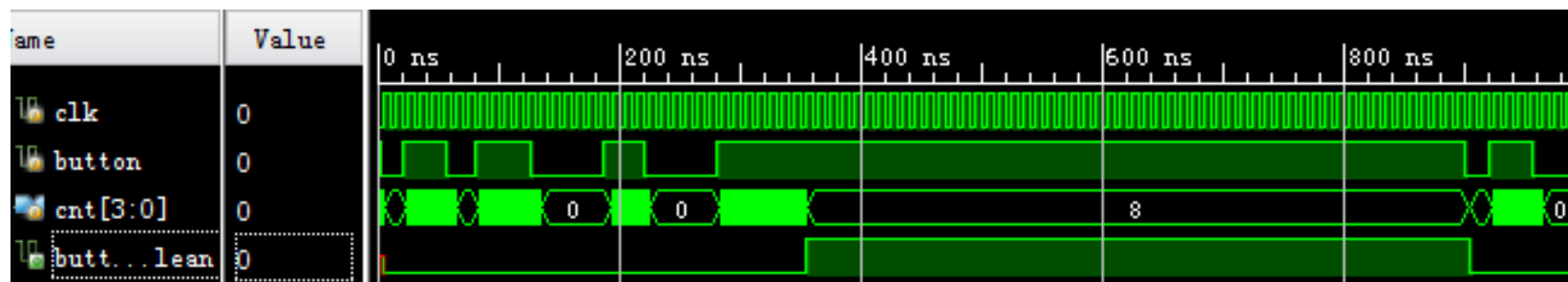
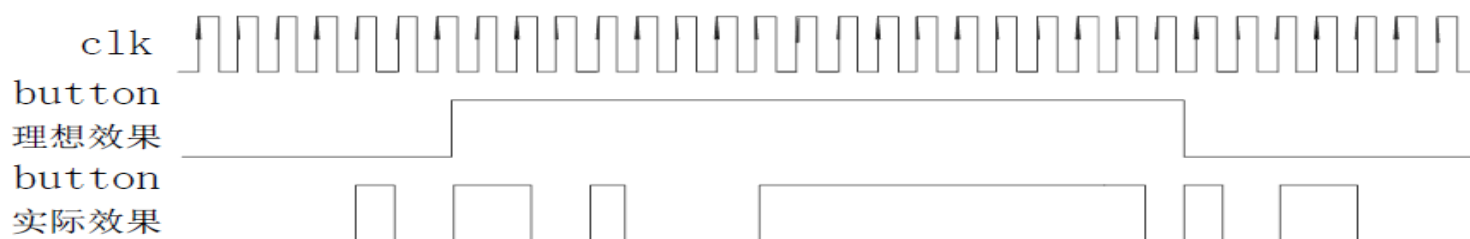
- 进一步熟悉FPGA 开发的整体流程
- 掌握几种常见的信号处理技巧
- 掌握有限状态机的设计方法
- 能够使用有限状态机设计功能电路

实验环境

- VLAB
- FPGAOL
- VIVADO
- Logisim

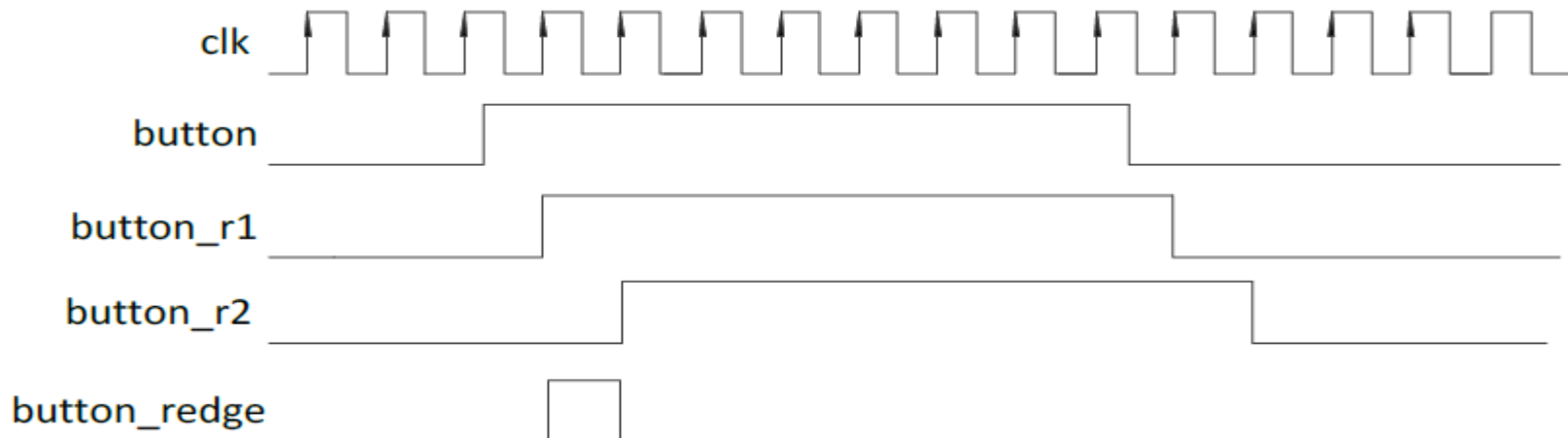
实验步骤：信号整形及去抖动

- 使用按键、开关作为输入时，由于其机械特性，在电平转换瞬间会产生信号抖动
- 持续时间很短，但在高频时钟看来则不然
- 用计数器记录高电平持续时间，以消除“毛刺”

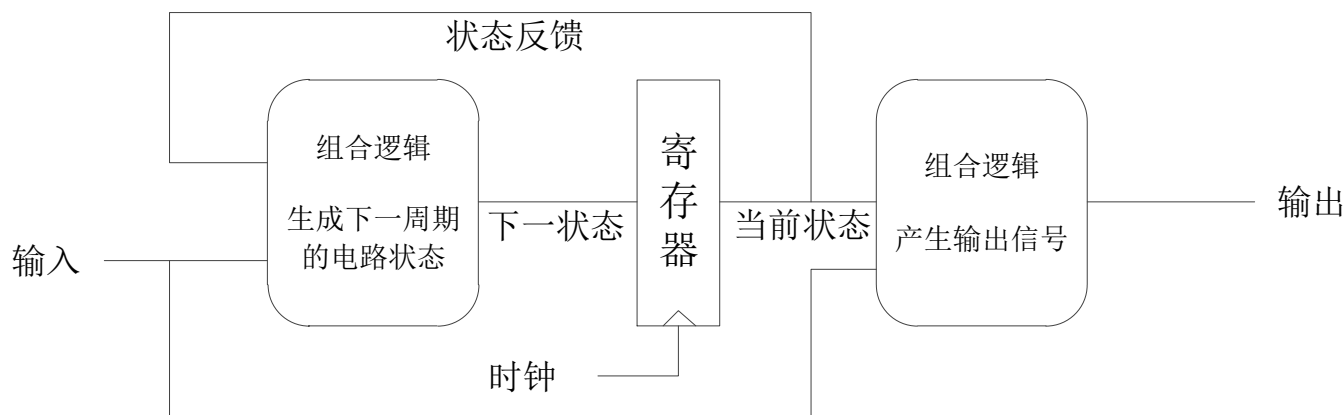
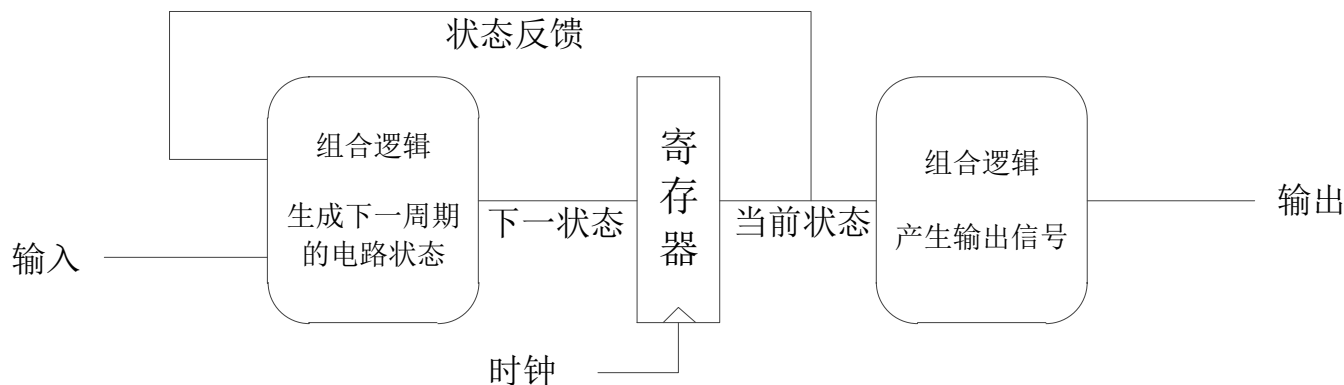


实验步骤：正确使用信号边沿

- 要求电路对某信号变化时进行相应
- 对于单bit信号来说，就是在信号边沿时刻
- 代码示范：
 - 错误： `always@(posedge button)`
 - 正确： `always@(posedge clk)`
`if(button_edge)...`



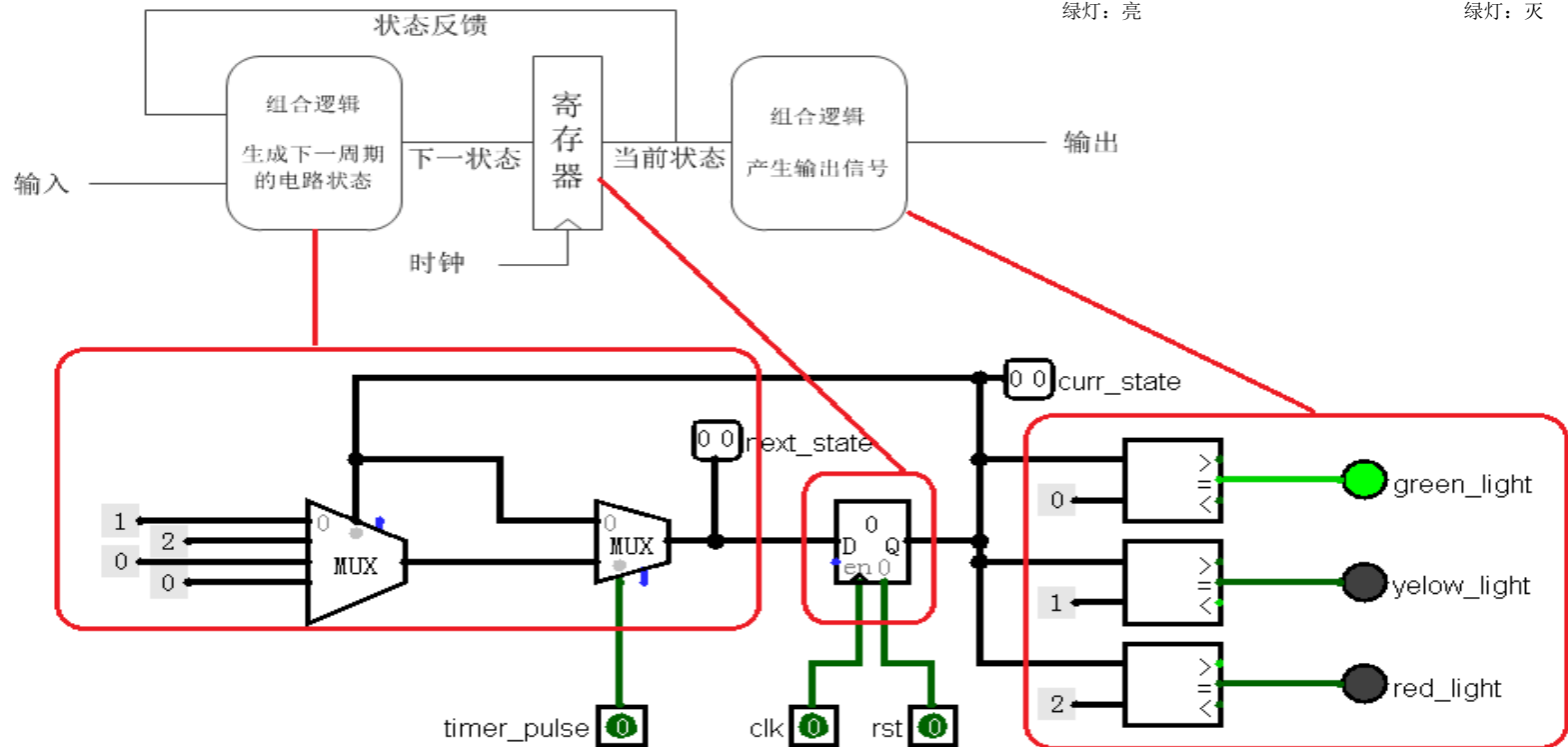
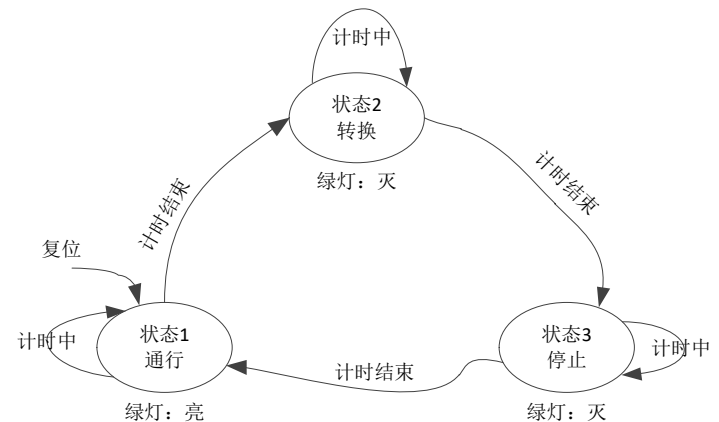
实验步骤：有限状态机（FSM）



对于上述电路结构，设寄存器位宽为 n ，则该电路的状态数量不会超过 2^n ，即其状态数量是有限的，因此这种电路结构称为有限状态机。

实验步骤：有限状态机

- 有限状态机分类：摩尔型、米莉型
- 以交通灯为例了解摩尔型有限状态机原理



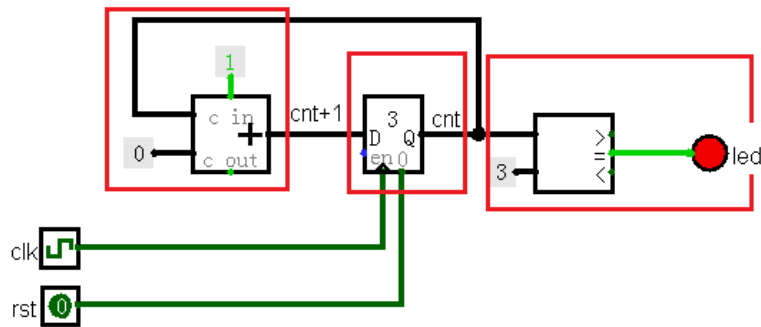
实验步骤：有限状态机代码

```
module traffic_ctrl(  
    input  clk,  
    input  rst,  
    input  timer_pulse,  
    output green_light  
);  
parameter C_PASS = 2'b00;  
parameter C_TRANS = 2'b01;  
parameter C_STOP = 2'b10;  
reg [1:0] curr_state;  
reg [1:0] next_state;  
//有限状态机第一部分  
always@(*)  
begin  
    if(timer_pulse)  
        begin  
            case(curr_state)  
                C_PASS: next_state = C_TRANS;  
                C_TRANS: next_state = C_STOP;
```

```
                C_STOP: next_state = C_PASS;  
                default: next_state = C_PASS;  
            endcase  
        end  
    else  
        next_state = curr_state;  
    end  
//有限状态机第二部分  
always@(posedge clk or posedge rst)  
begin  
    if(rst)  
        curr_state <= C_PASS;  
    else  
        curr_state <= next_state;  
    end  
//第三部分,各输出信号的赋值都应放在此部分  
assign green_light = (curr_state==C_PASS)? 1'b1 : 1'b0;  
//...  
endmodule
```

实验步骤：用计数器实现有限状态机

用计数器作为控制信号实现电路功能，从本质上来说，也是一种有限状态机



```
module test(  
    input clk,rst,  
    output led);  
    reg [1:0] cnt;  
    always@(posedge clk or posedge rst_n)  
    begin  
        if(rst)  
            cnt <= 2'b0;  
        else  
            cnt <= cnt + 1'b1;  
        end  
    assign led = (cnt==2'b11) ? 1'b1 : 1'b0;  
endmodule
```

讲解结束， 请开始实验