《数字逻辑》实验报告

姓名		**年级**	2022 级
学号		**专业、班级**	计算机科学与技术卓越 01 班

实验名称	**实验十七 交通灯信号控制器设计**		
实验时间	2023.11.23	**实验地点**	DS1410
实验成绩		**实验性质**	**□验证性 ☑设计性 □综合性**

教师评价：

□算法/实验过程正确； □源程序/实验内容提交 □程序结构/实验步骤合理；

□实验结果正确； □语法、语义正确； □报告规范；

评语：

评价教师签名（电子签名）：

一、实验目的

1、 巩固有限状态机设计方法。

2、 掌握交通灯控制原理。

3、 学会利用 FPGA 设计交通灯控制系统。

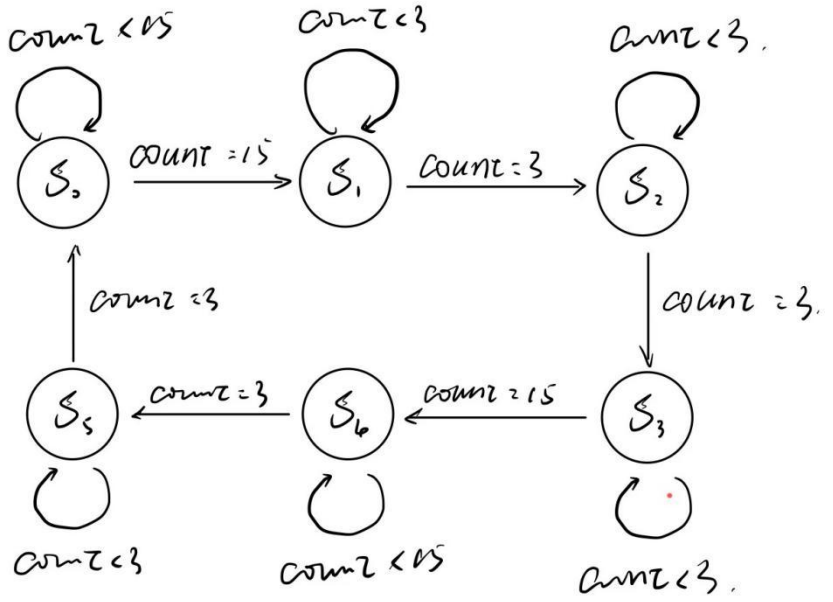
二、实验项目内容

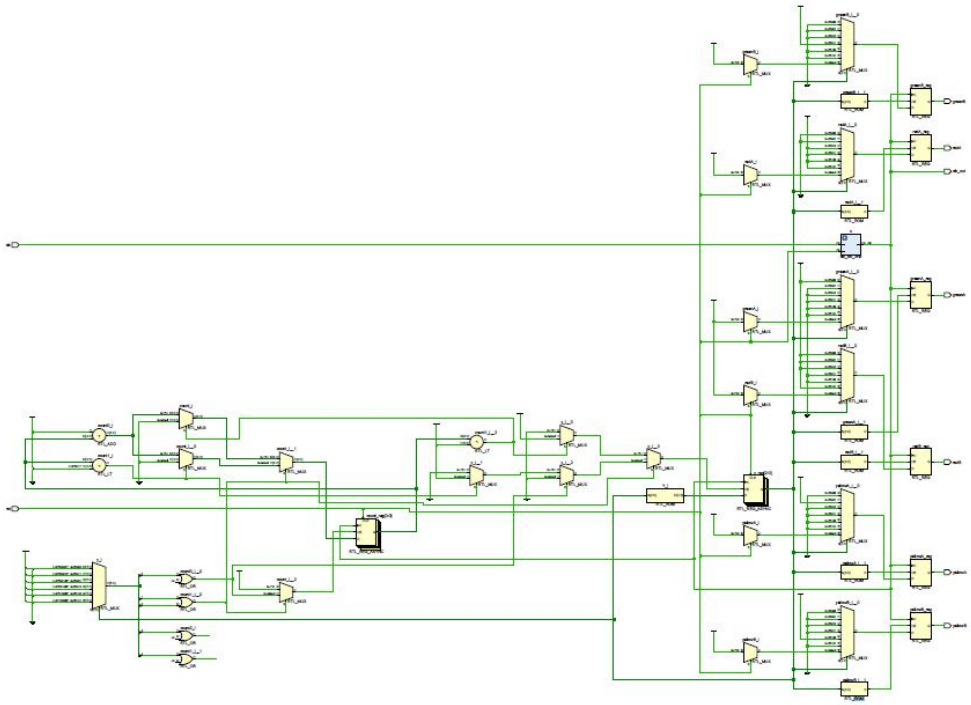
实现一个十字路口（南北和东西方向）交通灯信号控制器。该十字路口的南北和东西方向都有红、黄、绿三种颜色的信号灯。请根据交通规则，给出交通灯的状态表和状态转换图，延迟时间以秒计算。

三、实验设计

交通灯状态表和状态转移图如下图所示，如果使用频率为 3Hz 的时钟来驱动电路，那么延迟 1s 可以用三个时钟得到，类似的，用 15 个时钟可以得到 5 秒的延迟，count 用于延迟计数，在状态转移时归零，并重新开始计数。

状态	南北	东西	延迟（s）
0	绿	红	5
1	黄	红	1
2	红	红	1
3	红	绿	5
4	红	黄	1
5	红	红	1





四、实验过程或算法

1. 设计文件模块如下：

(1)traffic_light 模块代码：

```
`timescale 1ns / 1ps
module traffic_light(clk,rst,redA,redB,yellowA,yellowB,greenA,greenB,clk_out);
    input clk,rst;
    output reg redA,redB,yellowA,yellowB,greenA,greenB; //两个方向的红绿灯，共六个
    output clk_out;  //输出一个脉冲信号表示该时钟正在运行
    reg [2:0]y,Y;     //y 表示现态，Y 表示此态
    parameter S0 = 3'b000,S1=3'b001,S2=3'b010,S3=3'b011,S4=3'b100,S5=3'b101;
    wire clk_div;
    reg [3:0] count;
```

```
//调用 clk_div_3Hz 模块，对 clk 分频
clk_div_3Hz c(.clk(clk),.rst(rst),.clk_div(clk_div));


//计数，根据不同的状态（即不同的红绿灯）对分频后信号作不同计数处理，以实现等待效果；当达到一定要求时切换到次态
always @(posedge clk_div,negedge rst) begin
   if(!rst) begin
      count<=0;
      y<=S0;
   end
   else
   begin
      if(y==S0|y==S3)
      begin
               if(count<15)
            count<=count+1;
          else begin
            y<=Y;
            count<=0;
           end
       end
      else if(y==S1|y==S2|y==S4|y==S5)
      begin
         if(count<3)
            count<=count+1;
         else begin
            y<=Y;
            count<=0;
         end
      end
   end

end
```

```
//根据现态的状态对次态赋值
  always @(count,y,Y) begin
    case(y)
      S0: Y<=S1;
      S1: Y<=S2;
      S2: Y<=S3;
      S3: Y<=S4;
      S4: Y<=S5;
      S5: Y<=S0;
      default: Y<=S0;
    endcase
  end

//根据现态的状态对应到相应的红绿灯上
  always @(posedge clk_div or negedge rst) begin
    if(!rst) begin
        redA=1;yellowA=1;greenA=1;
        redB=1;yellowB=1;greenB=1;
      end
    case(y)
      S0: begin
        redA=0;yellowA=0;greenA=1;
        redB=1;yellowB=0;greenB=0;
      end
      S1: begin
        redA=0;yellowA=1;greenA=0;
        redB=1;yellowB=0;greenB=0;
      end
      S2: begin
        redA=1;yellowA=0;greenA=0;
        redB=1;yellowB=0;greenB=0;
      end
      S3: begin
        redA=1;yellowA=0;greenA=0;
```

```
            redB=0;yellowB=0;greenB=1;
        end
        S4: begin
            redA=1;yellowA=0;greenA=0;
            redB=0;yellowB=1;greenB=0;
        end
        S5: begin
            redA=1;yellowA=0;greenA=0;
            redB=1;yellowB=0;greenB=0;
        end
        endcase
    end

    assign clk_out=clk_div;

endmodule
```

2. clock_3Hz 代码如下：

```
`timescale 1ns / 1ps
module clk_div_3Hz(clk,rst,clk_div);
    input clk;
    input rst;
    output clk_div;
    reg [30:0]count;
    always @(posedge clk, negedge rst) begin
        if(!rst)begin
            count<=0;
        end
        else begin
            count<=count+1;
```

```
        end
    end
    assign clk_div=count[24];

endmodule
```

仿真文件编写如下：

```
`timescale 1ns / 1ps
module traffic_light_sim(

    );
    reg clk;
    reg rst;
    wire redA,redB,yellowA,yellowB,greenA,greenB;

    traffic_light t(
                .clk(clk),
                .rst(rst),
                .redA(redA),
                .redB(redB),
                .yellowA(yellowA),
                .yellowB(yellowB),
                .greenA(greenA),
                .greenB(greenB)
    );

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

```
    initial begin
        rst=0;
        #5 rst=1;
    end

endmodule
```

3. 综合、绑定引脚、实现

约束文件如下：

```
set_property IOSTANDARD LVCMOS18 [get_ports yellowB]
set_property IOSTANDARD LVCMOS18 [get_ports yellowA]
set_property IOSTANDARD LVCMOS18 [get_ports rst]
set_property IOSTANDARD LVCMOS18 [get_ports redB]
set_property IOSTANDARD LVCMOS18 [get_ports redA]
set_property IOSTANDARD LVCMOS18 [get_ports greenB]
set_property IOSTANDARD LVCMOS18 [get_ports greenA]
set_property IOSTANDARD LVCMOS18 [get_ports clk_out]
set_property IOSTANDARD LVCMOS18 [get_ports clk]
set_property PACKAGE_PIN W5 [get_ports clk]
set_property PACKAGE_PIN L1 [get_ports clk_out]
set_property PACKAGE_PIN U15 [get_ports greenA]
set_property PACKAGE_PIN U19 [get_ports greenB]
set_property PACKAGE_PIN V19 [get_ports redA]
set_property PACKAGE_PIN U16 [get_ports redB]
set_property PACKAGE_PIN R2 [get_ports rst]
set_property PACKAGE_PIN W18 [get_ports yellowA]
set_property PACKAGE_PIN E19 [get_ports yellowB]
```

五、实验过程中遇到的问题及解决情况

在实验过程中，当编写到有限状态机的状态转移关系的时候，误将次态当成了现态，导致进行仿真文件的测试时结果不对。

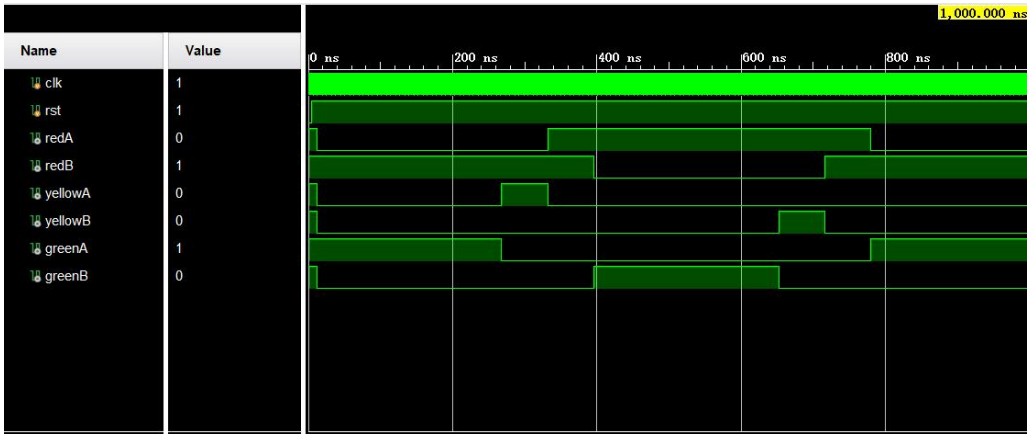
解决办法：通过观察仿真文件发现了问题，并通过添加注释的方式

区分现态和次态，以免搞错。

六、实验结果及分析和（或）源程序调试过程

1.仿真结果

仿真时调整了状态转换的延时，以便可以在较短的时间内完成仿真。



从仿真结果可知，状态转换是可以正常进行的。

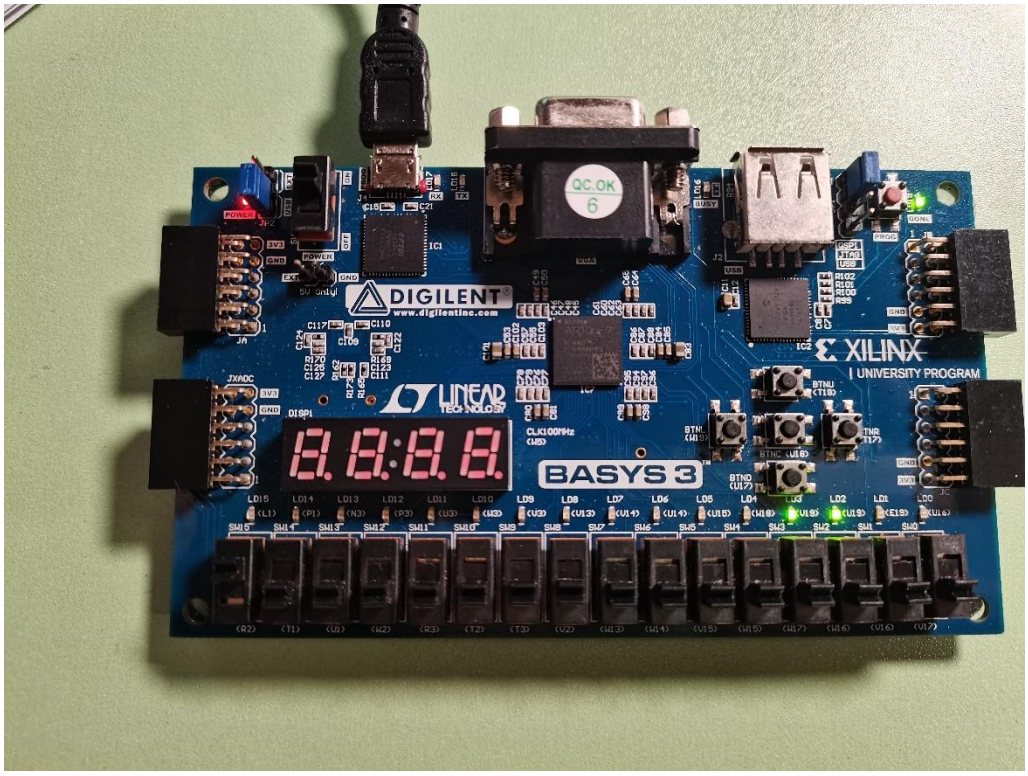
2.上板验证结果

最右边六盏灯分别是：

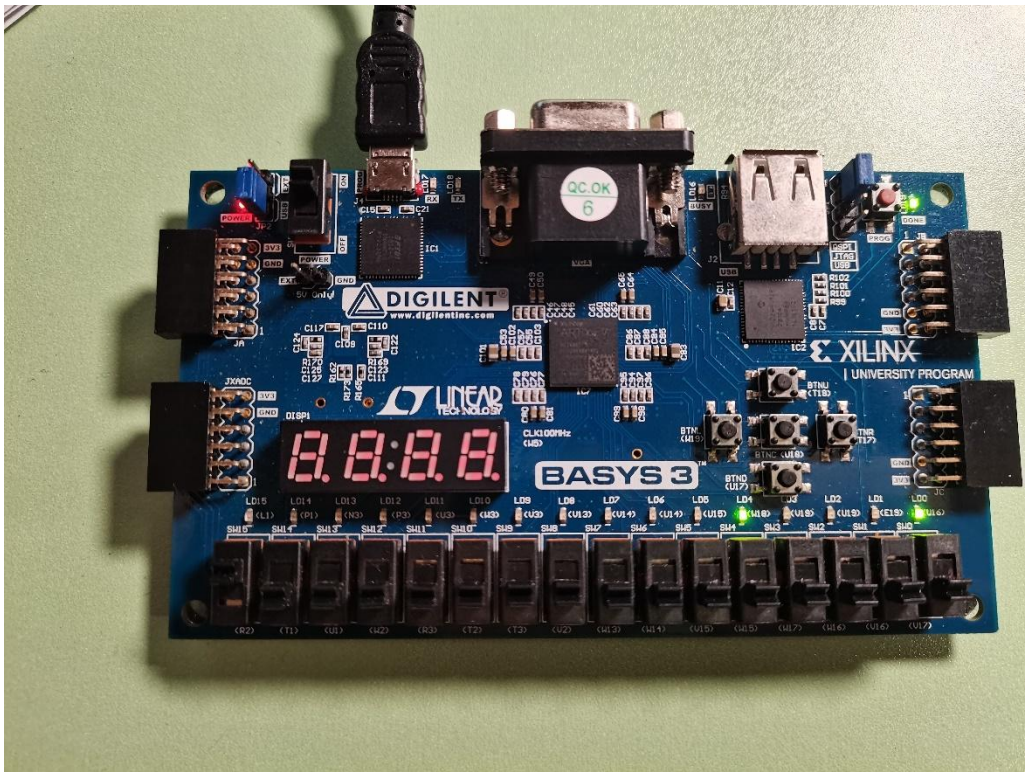
南北红、南北黄、南北绿、东西红、东西黄、东西绿。

最左边的灯显示 3Hz 时钟脉冲

最左边拨码开关表示重置信号。

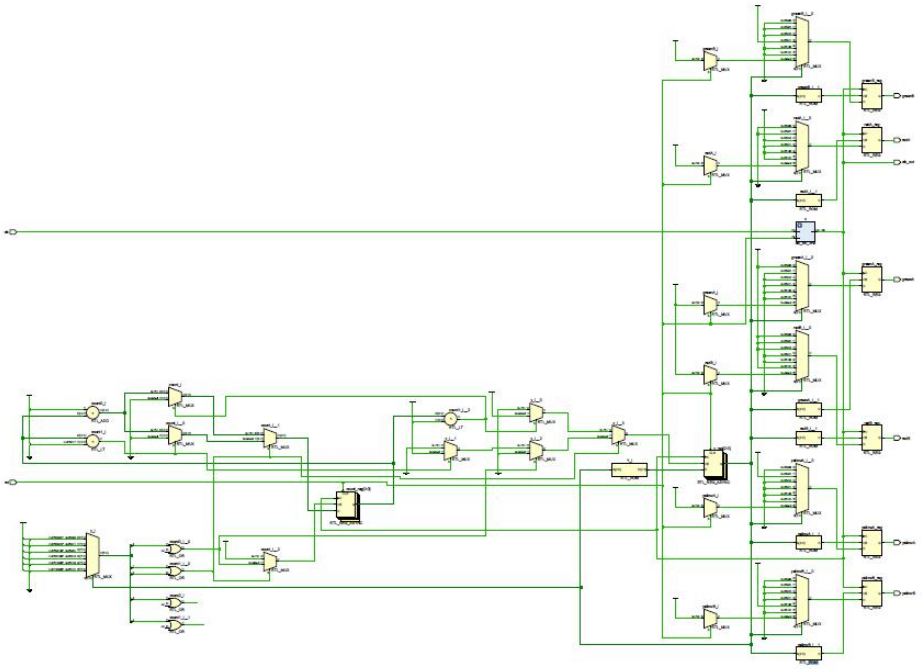


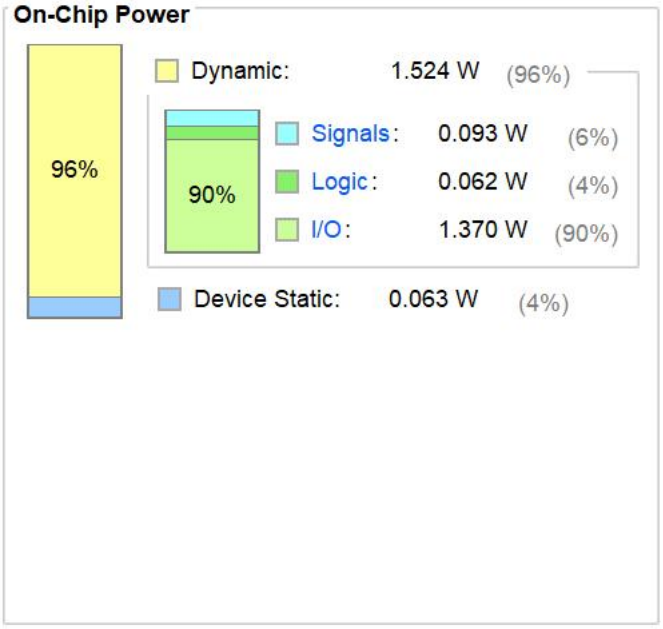
（南北绿灯，东西红灯）



（南北黄灯，东西绿灯）

3. 电路和功耗分析





七、小组分工情况说明

：编写交通信号控制器的代码。共同分析在 RTL 分析、逻辑资源占用、功耗上的问题。

：在仿真时对于设计代码进行适当修改，编写仿真代码，并进行仿真验证。共同分析在 RTL 分析、逻辑资源占用、功耗上的问题。

：完成综合分析、引脚绑定和上板验证。共同分析在 RTL 分析
、逻辑资源占用、功耗上的问题。