|  |  |  |  |
| --- | --- | --- | --- |
| **数字逻辑设计**  **实验报告** | | | |
| 红绿灯数字系统的分析与设计 | | | |
| 图片包含 标牌  已生成极高可信度的说明 | | | |
| **组长** |  |  |  |
| **组员1** |  |  |  |
| **组员2** |  |  |  |
| **电话** |  | | |
| **日期** | 2023-07-23 | | |

# 项目目标

本项目的目标是实现一个红绿灯数字系统，模拟道路交叉路口的红灯、黄灯、绿灯的交替。

# 项目介绍

项目将实现一个简单的红绿灯系统。红绿灯具有红、黄、绿三色，在两条道路的交叉路口，红绿灯可能会存在四种状态，第一种是道路1红灯、道路2绿灯，第二种是道路2通行时间结束后，转为黄灯，第三种是道路1绿灯，道路2红灯，第四种是道路1通行时间结束后，转为黄灯。第四种状态结束后，应转换为第一种状态。我们将使用状态机实现该系统。

项目开发使用的硬件是FuDan Micro新型数字电路开发板，其包括一系列按键和LED灯。项目开发使用的软件是Procise，我们将先进行分析设计、代码编写、调试和仿真，再使用软件进行编译和下载。

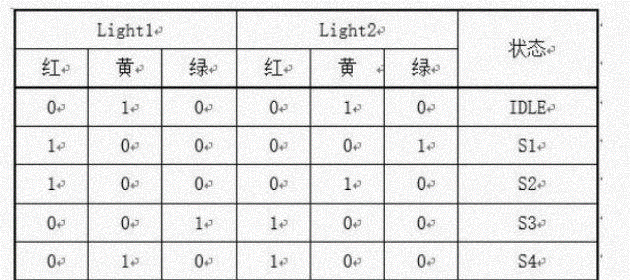
项目开发的分工如下：谢志成同学负责实验项目的确定、状态机逻辑的确定和verilog代码的编写，张胜同学负责电路板连接和操作验证，张子杰同学辅助进行代码调试修改、故障排查并撰写实验报告。小组每位成员工作量的占比均等，约33%。

# 项目设计

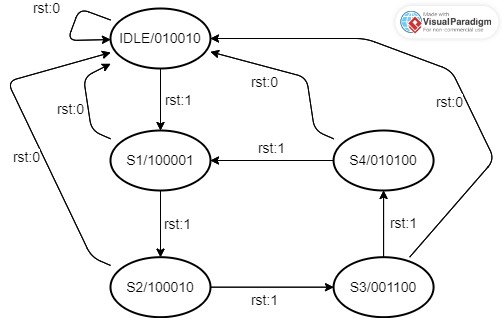
该数字系统需要两个输入信号，包括重置信号reset和时钟信号clk，其中时钟信号使用开发板相关接口，重置信号由按键控制；该系统还应具有六个输出信号lt1[3]和lt2[3]，分别对应一侧路口的红、黄、绿灯和另一侧路口的红、黄、绿灯。

该数字系统采用状态机的方式实现，拥有一个初始状态和四个交替切换的状态，分别是IDLE：（黄，黄），S1：（红，绿），S2：（红，黄），S3：（绿，红），S4：（黄，红）。系统开启后，将自动进入IDLE状态，并立即切换为状态S1，进入状态循环。重置按键按下时，系统将切换为初始状态，并在按键释放后切换为状态S1，并进入状态循环。

状态表如下图所示：



状态图如下图所示：



实现该数字系统的verilog代码如下：

always @ (posedge clk or negedge reset)

begin

if(~reset)

begin

state<=IDLE;

lt1\_s<=3'b010;

lt2\_s<=3'b010;

end

else

case(state)

IDLE:

begin

lt1\_s <= 3'b100;

lt2\_s <= 3'b001;

state <= S1;

end

S1: if(count==T-1)

begin

lt1\_s <= 3'b100;

lt2\_s <= 3'b010;

state <= S2;

end

S2: if(count==T-1)

begin

lt1\_s <= 3'b001;

lt2\_s <= 3'b100;

state <= S3;

end

S3: if(count==T-1)

begin

lt1\_s <= 3'b010;

lt2\_s <= 3'b100;

state <= S4;

end

S4: if(count==T-1)

begin

lt1\_s <= 3'b100;

lt2\_s <= 3'b001;

state <= S1;

end

default:

begin

state<=IDLE;

end

endcase

end

以上代码实现了IDLE、S1、S2、S3、S4五种状态之间的转换。当reset信号为0时，状态机无论处于何种状态均转换至IDLE状态；否则，IDLE和S4状态将转换至S1状态，S1状态转换至S2状态，S2状态转换至S3状态，S3状态转换至S4状态。由此状态机的各状态之间可以循环转换。

always @ (posedge clk or negedge reset)

begin

if(~reset)count<='d0;

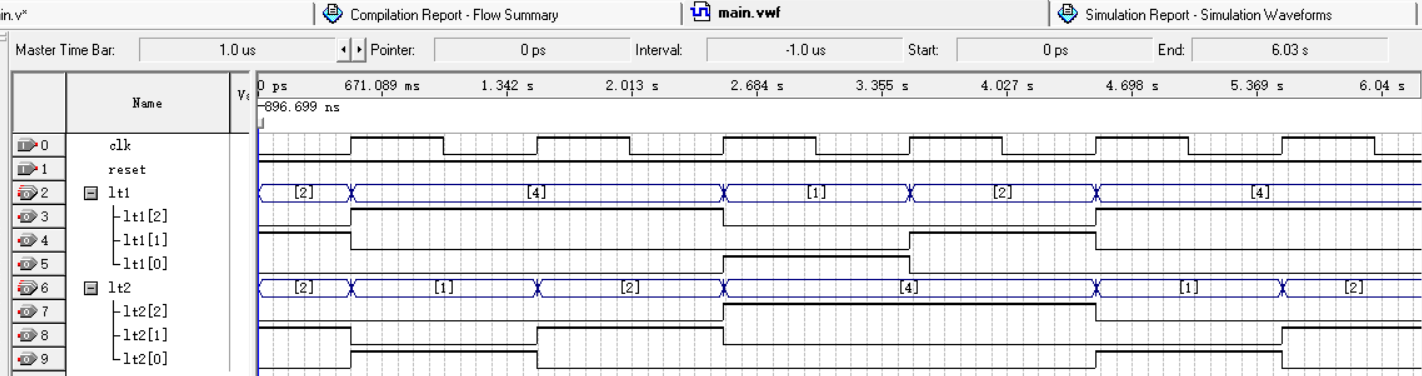
else if(count==T-1)count<='d0;

else count<=count+1;

end

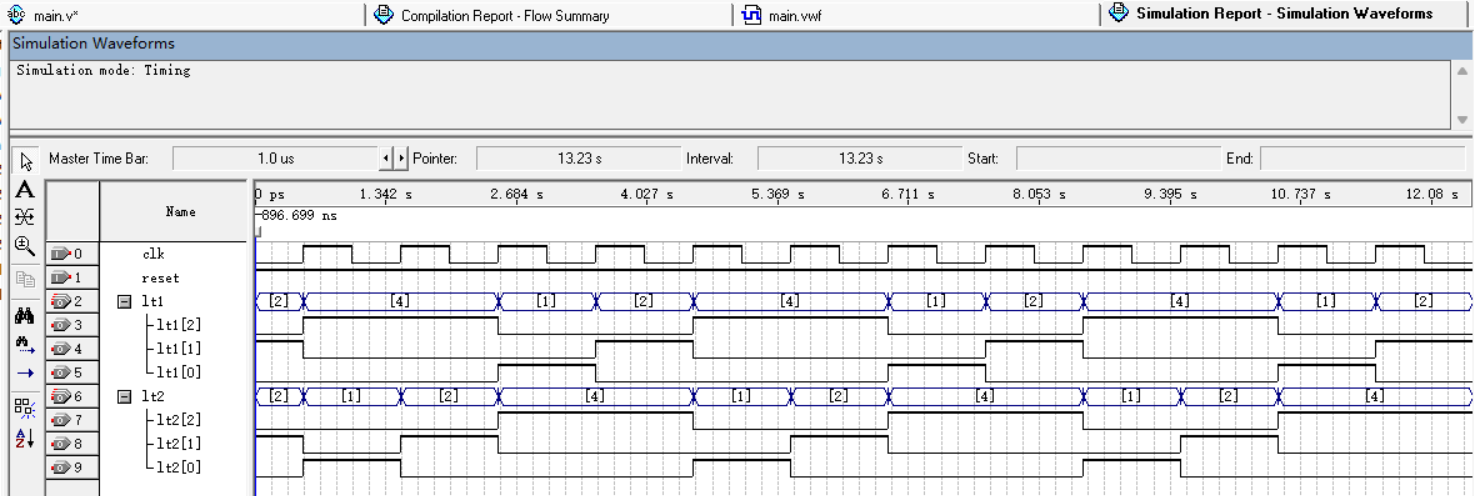
这段代码对时钟沿的数量进行计数。根据输入信号clk接入的时钟信号间隔不同，调整局部变量T的值，变量count将在每次时钟上升沿时加一，直至遇到重置信号或加至T-1。状态切换时，判断count的值是否等于T-1，由此实现状态切换的时间间隔控制。

预期仿真结果如下：



# 结果分析与讨论

对项目代码进行的仿真结果时序图如下：



可以看出，仿真得到的时序图与预期结果一致，编写的verilog代码可以实现项目所需的功能。

依据相关说明，编写约束文件。输入、输出管脚约束及时钟约束提供了输入reset、时钟clk、输出lt1和lt2的相关布线信息。

NET "clk" LOC = W17;

NET "reset" LOC = U19;

NET "lt1[0]" LOC = M17;

NET "lt1[1]" LOC = N17;

NET "lt1[2]" LOC = P18;

NET "lt2[0]" LOC = P19;

NET "lt2[1]" LOC = N19;

NET "lt2[2]" LOC = N18;

NET "clk" TNM\_NET=clk;

TIMESPEC TS\_clk = PERIOD "clk" 10 ns HIGH 50%;

完成编译综合和编译下载，在开发板上观察到的实验结果如下：

从上至下的六个灯分别代表道路1和道路2的绿、黄、红灯。从左至右的四张图片分别为状态S1、S2、S3、S4，在未按下重置按钮时，系统在四种状态之间循环。按下重置按钮时，两个黄灯点亮，系统处于状态IDLE。

实验结果与预期结果相符，开发板上的LED灯根据预期点亮方式点亮，重置按键可以正常工作，红绿灯系统得到实现。

# 项目总结

该项目已完全实现项目目标中所需的功能，可以实现三色红绿灯的交替，适用于交叉路口的简单红绿灯。然而，该项目仍存在许多不足之处。现实中所需的红绿灯一般具有更复杂的功能，包括各路口通行时间的控制、左转和右转红绿灯、红绿灯剩余时间的显示等，该项目并没有实现这些更高级实用的功能。我们可以通过修改项目代码中切换状态的部分、为该状态机增加状态等方式，尝试实现路口通行时间的控制、左转红绿灯等功能。