



课 程 设 计 报 告

**课程名称 EDA技术**

**课题名称 交通灯控制电路设计**

**专 业 通信工程**

**班 级 1601**

**学 号 13**

**姓 名 李云亮**

**指导教师 乔汇东、胡瑛**

**2018年 7月15日**

湖南工程学院

课 程 设 计 任 务 书

课程名称 EDA技术与应用

课 题 交通灯控制电路设计

专业班级 通信工程1601

学生姓名 李云亮

学 号 13

指导老师 乔汇东 胡瑛

审 批 乔汇东

任务书下达日期 2018 年 7月 8日

任务完成日期 2018 年 7月 15日

《EDA技术与应用》课程设计任务书

**一 、设计目的**

全面熟悉、掌握VHDL语言基本知识，掌握利用VHDL语言进行常用的组合逻辑电路和时序逻辑电路编辑，把电路EDA设计和实际结合起来，熟悉编制和调试电路代码的技巧，掌握分析结果的若干有效方法，进一步提高上机动手能力，培养综合电路的设计能力，初步养成规范代码的习惯和撰写规范文档资料的习惯。

**二、设计要求**

1、设计正确，方案合理。

2、程序精炼，结构清晰。

3、设计报告5000字以上，含程序设计说明，用户使用说明，源程序清单及程序框图。

4、上机演示。

5、有详细的文档。文档中包括设计思路、设计仿真程序、仿真结果及相应的分析与结论。

**三、进度安排**

第十九周 星期一14：30-18：30： 课题讲解，查阅资料

　 星期四8：30-12：30： 总体设计，详细设计

星期四14：30-18：30： 编程，上机调试、修改程序

星期五8：30-12：30： 上机调试、完善程序

星期五14：30-18：30： 答辩

星期六-星期天：撰写课程设计报告

附：

课程设计报告装订顺序：封面、任务书、目录、正文、评分、附件（A4大小的图纸及程序清单）。

正文的格式:一级标题用3号黑体,二级标题用四号宋体加粗,正文用小四号宋体;段落设置行距为22磅，首行缩进2字符。

正文的内容:一、课题的主要功能；二、主要功能的实现（要求包含总体的功能模块图，包含各功能模块的详细解释）；三、电路调试与仿真（含仿真图和图的基本解释，运行效果图和图的基本解释）；四、总结与体会；五、评分表

**目 录**

[一、 交通灯控制电路的总体设计](#_Toc16339_WPSOffice_Level1) **[1](#_Toc16339_WPSOffice_Level1)**

**[1.1交通灯控制电路实现目标与要求](#_Toc23416_WPSOffice_Level2)** [1](#_Toc23416_WPSOffice_Level2)

**[1.2基本设计思想](#_Toc9876_WPSOffice_Level2)** [1](#_Toc9876_WPSOffice_Level2)

[二、 交通灯控制电路的功能实现](#_Toc23416_WPSOffice_Level1) **[2](#_Toc23416_WPSOffice_Level1)**

**[2.1系统的总体框图](#_Toc15261_WPSOffice_Level2)** [2](#_Toc15261_WPSOffice_Level2)

**[2.2系统各模块详细设计](#_Toc5784_WPSOffice_Level2)** [2](#_Toc5784_WPSOffice_Level2)

**[2.3系统各功能模块的实现](#_Toc32721_WPSOffice_Level2)** [5](#_Toc32721_WPSOffice_Level2)

[三、电路调试与仿真](#_Toc9876_WPSOffice_Level1) **[8](#_Toc9876_WPSOffice_Level1)**

[四、总结与体会](#_Toc15261_WPSOffice_Level1) **[1](#_Toc15261_WPSOffice_Level1)0**

[五、评分表](#_Toc5784_WPSOffice_Level1) **[1](#_Toc5784_WPSOffice_Level1)2**

# 交通灯控制电路的总体设计

## 1.1交通灯控制电路实现目标与要求

**（1）问题描述：**

交通灯的显示有很多方式，如十字路口、丁字路口等，而对于同一个路口又有很多不同的显示要求，比如十字路口，车辆如果只要东西和南北方向通行就很简单，而如果车子可以左右转弯的通行就比较复杂，本实验仅针对最简单的南北和东西直行的情况。

**（2）功能要求：**

要完成本实验，首先必须了解交通路灯的燃灭规律。依人们的交通常规，“红灯停，绿灯行，黄灯提醒”。其交通灯的燃灭规律为：初始态是两个路口的红灯全亮，之后，东西路口的绿灯亮，南北路口的红灯亮，东西方向通车，延时一段时间后，东西路口绿灯灭，黄灯开始闪烁。闪烁若干次后，东西路口红灯亮，而同时南北路口的绿灯亮，南北方向开始通车，延时一段时间后，南北路口的绿灯灭，黄灯开始闪烁。闪烁若干次后，再切换到东西路口方向，重复上述过程。

系统要求实现对2组交通灯控制信号的模拟输出，其控制规律为如上所述，同时要完成等待时间模拟的输出，即红绿灯的持续时间倒数输出。系统时钟输入为一个1KHz的系统时钟。

**（2）小题目：**

黄灯闪烁时间从3s增至5s,且第1s闪烁1次，第2s闪烁2次，以此类推到第5s闪烁5次。

## 1.2基本设计思想

（1）系统时钟为1KHz，设计一个五百分频器，得到周期为1s的时间脉冲。

（2）设计一个状态机，总共有s0,s1,s2,s3,s4五种状态，初始状态s0为南北路口与东西路口红灯全亮，s1为东西路口绿灯亮，倒计时为9s，南北路口红灯亮，倒计时为14s(多出的5s为黄灯闪烁时间)，s2为东西路口黄灯闪烁，南北路口红灯亮，倒计时为5s，s3为东西路口红灯亮，倒计时为14s，南北路口为绿灯亮，倒计时为9s，s3为东西路口红灯亮，南北路口黄灯闪烁，倒计时为5s。

（3）通过倒计数器进行计数，cnt为0时跳转至下一状态。

（4）用4个数码管显示模块显示当前南北、东西路口的等待或通行时间。

# 交通灯控制电路的功能实现

## 2.1系统的总体框图

交通灯

状态机

控制模块

倒计时

计数模块

数码管显示

模块

## 2.2系统各模块详细设计

1. 分频器设计

500分频器：将1KHz的系统时钟分频，得到周期为1s的时间脉冲。

代码如下：

process(clk)

begin

if clk'event and clk = '1' then

if cnt1<999 then

if cnt1<499 then

clk1<='0';

else

clk1<='1';

end if;

cnt1<=cnt1+1;

else

cnt1<=0;

end if;

end if;

end process;

m分频器：根据不同的m值产生不同的周期，达到黄灯闪烁次数逐渐增加的效果。

代码如下：

process(clk)

begin

if clk'event and clk='1' then

if count<m then tmp<=(m-1)/2;

if count<tmp then

m1<=1;

else

m1<=0;

end if;

count<=count+1;

else

count<=0;

end if;

end if;

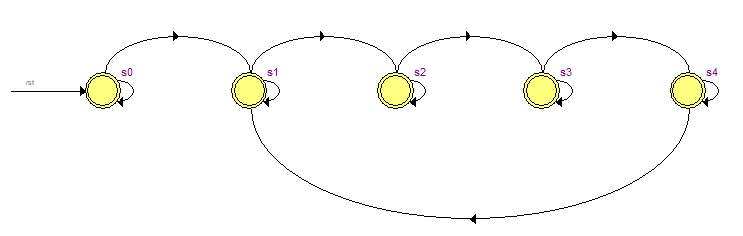
end process;

1. 状态机设计

交通灯状态转换表：

|  |  |  |
| --- | --- | --- |
|  | 南北路口 | 东西路口 |
| S0 | 红灯 | 红灯 |
| S1 | 红灯 | 绿灯 |
| S2 | 红灯 | 黄灯闪烁5s |
| S3 | 绿灯 | 红灯 |
| S4 | 黄灯闪烁5s | 红灯 |

状态机转换图：



1. 倒计时设计

当clk1时钟上升沿来临且等于1（clk1为分出的周期为1s的时间脉冲），先判断lock是否为1，仅当lock=1时，将初始时间init1赋给cnt，否则cnt自减1。

1. 数码管显示设计

定义一个sel信号，在0到3之间循环，利用人眼的视觉残留和数码管的余辉效应，使人感觉四个数码管同时都在显示，定义t1作为南北路口的等待或通行时间，t2作为东西路口的等待或通行时间，将cnt的值赋给t1，t2，将t1，t2除以10，得到十位，取余得到个位，送给段选，即数码管的内容显示控制。

## 2.3系统各功能模块的实现

1. 状态机控制模块

该程序是通过状态机的设置来实现控制模块的，它的作用是根据计数器的计数值来判断等待或通行时间是否结束，当计数值为0时，意味着等待或通行时间结束，进入下一个状态，当rst=’0’时(即按下复位键)，回到初始状态，南北路口与东西路口红灯全亮，计数值置0。

其进程代码如下：

lock--控制信号，倒计时为0时，给高电平，转换状态

c\_state--表示当前状态

n\_state--表示下一状态

init1--寄存时间的信号

state\_com:process(c\_state,lock,count)

begin

case c\_state is

when s0 => r2<=1;r4<=1;r1<=1;r3<=1;

g1<=0;g2<=0;g3<=0;g4<=0;

init1<=12;t1<=cnt;t2<=cnt;lock<='1';

if cnt=0 then n\_state<=s1;

else n\_state<=s0;lock<='0';

end if;

when s1 => r2<=1;r4<=1;g1<=1;g3<=1;

r1<=0;r3<=0;g2<=0;g4<=0;

init1<=5;t1<=cnt+5;t2<=cnt;lock<='1';init2<=0;

if cnt=0 then n\_state<=s2;

else n\_state<=s1;lock<='0';

end if;

when s2 => r2<=1;r4<=1;g2<=0;g4<=0;

r1<=1;g1<=1;r3<=1;g3<=1;

init1<=12;t1<=cnt;t2<=cnt;lock<='1';

if cnt>11 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>10 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>9 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>8 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>7 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>6 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>5 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>4 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>3 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>2 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>1 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

else

r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

end if;

if cnt<12 and cnt>0 then n\_state<=s2;lock<='0';

else n\_state<=s3;

end if;

when s3 => g2<=1;g4<=1;r1<=1;r3<=1;

r2<=0;r4<=0;g1<=0;g3<=0;

init1<=5;t1<=cnt;t2<=cnt+5;lock<='1';init2<=0;

if cnt=0 then n\_state<=s4;

else n\_state<=s3;lock<='0';

end if;

when s4 => r1<=1;r3<=1;g1<=0;g3<=0;

r2<=1;g2<=1;r4<=1;g4<=1;

init1<=12;t1<=cnt;t2<=cnt;lock<='1';

if cnt>11 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>10 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>9 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>8 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>7 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>6 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>5 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>4 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>3 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>2 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>1 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

else

r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

end if;

if cnt<12 and cnt>0 then n\_state<=s4;lock<='0';

else n\_state<=s1;

end if;

when others => n\_state<=s0;

end case;

end process;

1. 倒计时模块

该程序通过设计倒计数器来实现时间倒计时输出，当按下复位键rst=’0’时，cnt置0。

其进程代码如下：

process(clk1)

begin

if clk1'event and clk1='1' then

if lock='1' then

cnt<=init1;cnt0<=init2;

else

cnt<=cnt-1;cnt0<=cnt0+1;

end if;

if cnt0=1 then m<=999;

elsif cnt0=2 then m<=499;

elsif cnt0=3 then m<=249;

elsif cnt0=4 then m<=124;

else m<=61;

end if;

if rst='0' then cnt<=0;

end if;

end if;

temp0<=t1/10;

temp1<=t1 rem 10;

temp2<=t2/10;

temp3<=t2 rem 10;

end process;

1. 数码管显示模块

该程序是通过显示模块将分位模块的数据用七段数码管显示出来，共用了4个显示模块来显示南北、东西路口的等待或通行时间。

其进程代码如下：

process(clk)

begin

if clk'event and clk='1' then

sel<=sel+1;

end if;

end process;

process(sel)

begin

case sel is

when 0 => we<="00000001";num<=temp0;

when 1 => we<="00000010";num<=temp1;

when 2 => we<="00000100";num<=temp2;

when 3 => we<="00001000";num<=temp3;

when others => we<="00000000";

end case;

end process;

process(num)

begin

case num is

when 0 => du<="11111100";

when 1 => du <="01100000";

when 2 => du <="11011010";

when 3 => du <="11110010";

when 4 => du <="01100110";

when 5 => du <="10110110";

when 6 => du <="10111110";

when 7 => du <="11100000";

when 8 => du <="11111110";

when 9 => du <="11110110";

when others =>du<="00000000";

end case;

end process;

# 电路调试与仿真

1. 调试中遇到的问题

首先结束一个进程或者实体描述必须加上结束语句，否则会出现错误。如图1，图2所示：

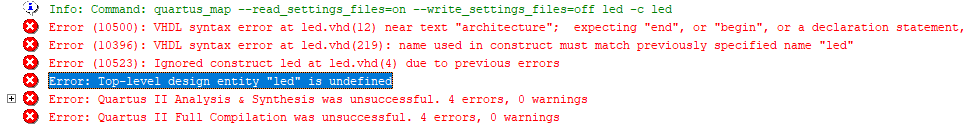


图1 未加上实体描述的结束语句

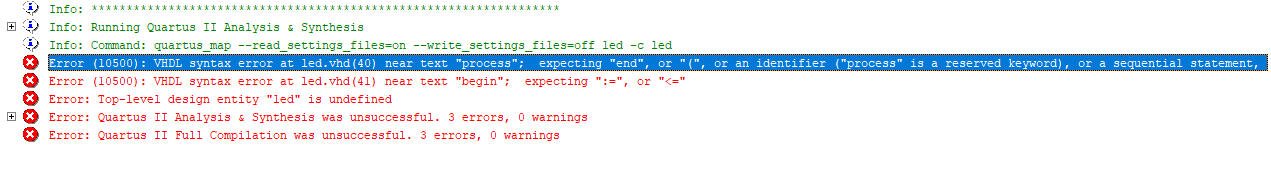


图2 未加上进程的结束语句

如果是赋值语句，那么两边的数据类型必须一致，否则也会出现错误。如图3所示：

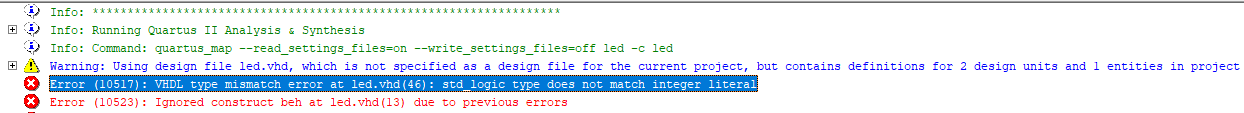


图3 将integer型赋给std\_logic型

1. 仿真图形

程序时序仿真

用状态机控制交通灯转换的仿真，如图4所示：

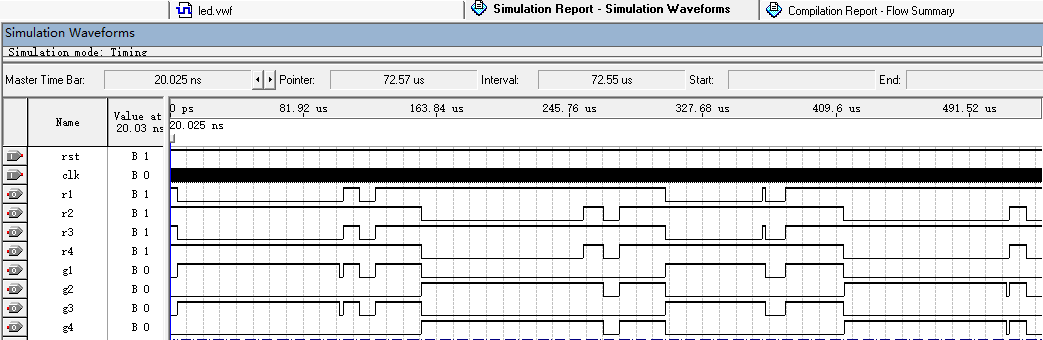


图4交通灯状态转换仿真

用倒计数器设计的倒计时仿真如图5所示：

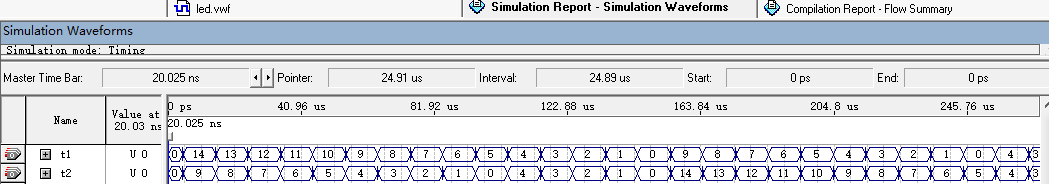


图5倒计时仿真

总体时序仿真如图6所示：

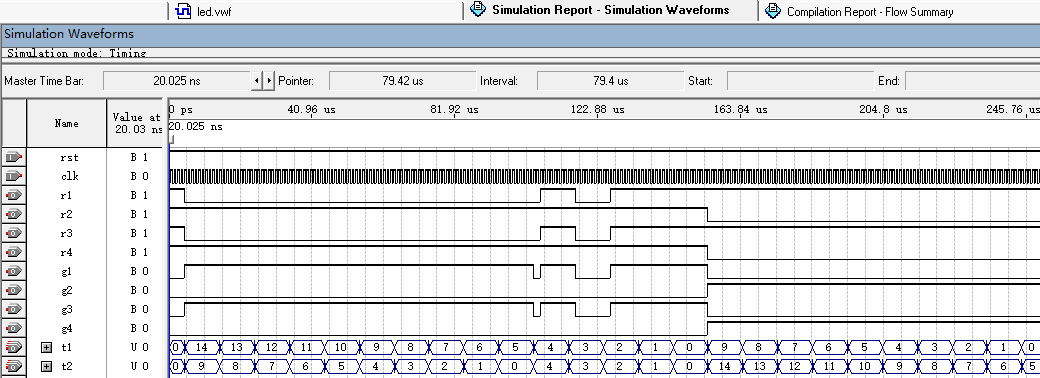


图6 总体时序仿真

# 四、总结与体会

EDA的课程设计大概是我目前遇到的最有意思，也最有挑战性的课程设计了，而乔老师也是声名在外，我也多次从学长那听说过他与众不同的要求，五次上机实验必须有一次通过他的检查才能参加课程设计，课程设计的内容必须独立完成。从这里可以看出乔老师的认真负责，他在课堂上讲解上机实验时也曾多次跟我们说，希望我们不要死搬硬套，要活学活用，代码逻辑要清晰，只有自己动手写代码才能在程序编译中发现问题，从而解决问题，达到学习的目的。我分到的题目是交通灯的控制电路设计，听上去很难，可乔老师说，交通灯挺简单的，就是倒计时数码管显示那里有点麻烦。因为平时也听了课，上机实验也有认真完成，我很快搞清了题目的要求，设计好了程序的总体框架，分为控制模块，计数模块和数码管显示模块三大块。但是在程序编译的过程中，我却遇到了各种各样的错误，有给信号多次赋值的错误，也有忘记加end if结束语句造成的错误，真应了那句“书上得来终觉浅，绝知此事要躬行”。在排查了种种错误后，程序终于编译成功，我开始对电路进行仿真，其实主要是看灯的转换状态是否有误。完成仿真后当然是配置引脚，将程序下载到实验箱上，看最终的效果如何，刚开始的效果总是不尽如人意，这时只有根据效果来反推是程序的哪里出现了逻辑上的错误，最终导致了错误的结果。尽管调试与改错的过程是枯燥乏味的，但是当你每解决掉一个错误，那种发自内心的喜悦是很难用文字表达出来的，每一个错误都代表了你的一个知识盲点，整个课程设计，在我看来，就是一个发现问题，解决问题，收获知识的过程，当然前提是得自己动手写代码，才能深有体会。本来我想星期四上完复习课就答辩，没想到突然接到电话，被告知要去做创新创业项目的PPT汇报，四下一看，乔老师不在，只能让同学跟老师说一下，等我汇报完赶来，发现答辩已经开始了，有不少人已经通过了答辩，其中不乏有和我做同一道题的同学，这意味着如果我现在去答辩，乔老师肯定得给我加个新功能，以此证明代码是我自己写的。虽然我对自己的程序有信心，但是如果要加新功能的话，心里不免还是有点打鼓。果不其然，乔老师看了我的程序后，想了会，让我加一个功能，实现黄灯闪烁5s，听到这里，我心里有点窃喜，这也太简单了，不过改几个数字的事情，分分钟搞定，没想到他接着说，第1s黄灯闪烁1次，第2s闪烁2次，以此类推，到第5s黄灯闪烁5次，听到这里我当时就有点懵了，难道今天要栽在这里吗？冷静下来后，仔细想想，这个问题未必没有办法解决，我最开始想到的是改变占空比，设计一个240分频器，在第1s送一个值使占空比变为50%，第2s送一个值使占空比变为25%......这样一来就能达到乔老师所说的要求，当我改好程序，编译成功下载到实验箱后，发现效果的确挺像那么回事，黄灯闪烁的频率似乎的确越来越快，等到第二天上午，乔老师过来看我的程序时，他指出了我的问题所在，原来我只是改变了黄灯亮灭的占空比，在1s内，黄灯还是只闪烁一次，只是亮灭时间不同而已，效果只是黄灯看上去越来越亮，并没有实现要求。到这里我都有点想要放弃了，好不容易想出的解题思路却是错误的，这下我真感觉自己有点“黔驴技穷”了，但是心里总是有不甘心的念头，难道就这样算了？仔细回想乔老师说的话，占空比不是周期，改变占空比也并没有改变它闪烁的周期，我突然有点明白老师的意思了，要想改变1s内闪烁的次数，就要改变它的周期，这样看来，先前的代码并非没有可取之处，我只需要，在不同的时间送不同的值来得到想要的周期就可以了，于是我故技重施，设计了一个m分频器，m就是那个不同的值，然后将脉冲的状态赋给黄灯就行了，这里要注意的是信号类型必须一致才行。历经几番波折，总归是有惊无险地通过了，短短几天的课程设计给我的收获却是难以估量的，在这里，真心感谢我们认真负责的乔老师！

# 五、评分表

计算机与通信学院课程设计评分表

|  |  |
| --- | --- |
| **项 目** | **评 价** |
| 设计方案的合理性与创造性 |  |
| 设计与调试结果 |  |
| 设计说明书的质量 |  |
| 答辩陈述与回答问题情况 |  |
| 课程设计周表现情况 |  |
| 综合成绩 |  |

课题名称：

|  |  |
| --- | --- |
| **项 目** | **评 价** |
| 设计方案的合理性与创造性 |  |
| 设计与调试结果 |  |
| 设计说明书的质量 |  |
| 答辩陈述与回答问题情况 |  |
| 课程设计周表现情况 |  |
| 综合成绩 |  |

教师签名：

日 期：

# 附录

程序清单：

library ieee;

use ieee.std\_logic\_1164.all;

entity led is

port(clk,rst:in std\_logic;

t1,t2:buffer integer range 0 to 14;

r1,r2,r3,r4,g1,g2,g3,g4,m1:buffer integer range 0 to 1;

we,du:out std\_logic\_vector(7 downto 0)

);

end led;

architecture beh of led is

type state is(s0,s1,s2,s3,s4);

signal n\_state,c\_state:state;

signal cnt,init1,init2:integer range 0 to 14;

signal cnt0:integer range 0 to 4;

signal sel:integer range 0 to 3;

signal cnt1,m,count,tmp:integer range 0 to 999;

signal clk1,lock:bit;

signal num,temp0,temp1,temp2,temp3:integer range 0 to 9;

begin

process(clk)

begin

if clk'event and clk = '1' then

if cnt1<999 then

if cnt1<499 then

clk1<='0';

else

clk1<='1';

end if;

cnt1<=cnt1+1;

else

cnt1<=0;

end if;

end if;

end process;

process(clk)

begin

if clk'event and clk='1' then

if count<m then tmp<=(m-1)/2;

if count<tmp then

m1<=0;

else

m1<=1;

end if;

count<=count+1;

else

count<=0;

end if;

end if;

end process;

state\_reg:process(clk1,rst)

begin

if rst='0' then c\_state<=s0;

elsif clk1'event and clk1='1' then

c\_state<=n\_state;

end if;

end process;

state\_com:process(c\_state,lock,count)

begin

case c\_state is

when s0 => r2<=1;r4<=1;r1<=1;r3<=1;

g1<=0;g2<=0;g3<=0;g4<=0;

init1<=12;t1<=cnt;t2<=cnt;lock<='1';

if cnt=0 then n\_state<=s1;

else n\_state<=s0;lock<='0';

end if;

when s1 => r2<=1;r4<=1;g1<=1;g3<=1;

r1<=0;r3<=0;g2<=0;g4<=0;

init1<=5;t1<=cnt+5;t2<=cnt;lock<='1';init2<=0;

if cnt=0 then n\_state<=s2;

else n\_state<=s1;lock<='0';

end if;

when s2 => r2<=1;r4<=1;g2<=0;g4<=0;

r1<=1;g1<=1;r3<=1;g3<=1;

init1<=12;t1<=cnt;t2<=cnt;lock<='1';

if cnt>11 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>10 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>9 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>8 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>7 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>6 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>5 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>4 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>3 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>2 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

elsif cnt>1 then r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

else

r2<=1;r4<=1;g2<=0;g4<=0;

r1<=m1;g1<=m1;r3<=m1;g3<=m1;

end if;

if cnt<12 and cnt>0 then n\_state<=s2;lock<='0';

else n\_state<=s3;

end if;

when s3 => g2<=1;g4<=1;r1<=1;r3<=1;

r2<=0;r4<=0;g1<=0;g3<=0;

init1<=5;t1<=cnt;t2<=cnt+5;lock<='1';init2<=0;

if cnt=0 then n\_state<=s4;

else n\_state<=s3;lock<='0';

end if;

when s4 => r1<=1;r3<=1;g1<=0;g3<=0;

r2<=1;g2<=1;r4<=1;g4<=1;

init1<=12;t1<=cnt;t2<=cnt;lock<='1';

if cnt>11 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>10 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>9 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>8 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>7 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>6 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>5 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>4 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>3 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>2 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

elsif cnt>1 then r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

else

r1<=1;r3<=1;g1<=0;g3<=0;

r2<=m1;g2<=m1;r4<=m1;g4<=m1;

end if;

if cnt<12 and cnt>0 then n\_state<=s4;lock<='0';

else n\_state<=s1;

end if;

when others => n\_state<=s0;

end case;

end process;

process(clk1)

begin

if clk1'event and clk1='1' then

if lock='1' then

cnt<=init1;cnt0<=init2;

else

cnt<=cnt-1;cnt0<=cnt0+1;

end if;

if cnt0=1 then m<=999;

elsif cnt0=2 then m<=499;

elsif cnt0=3 then m<=249;

elsif cnt0=4 then m<=124;

else m<=61;

end if;

if rst='0' then cnt<=0;

end if;

end if;

temp0<=t1/10;

temp1<=t1 rem 10;

temp2<=t2/10;

temp3<=t2 rem 10;

end process;

process(clk)

begin

if clk'event and clk='1' then

sel<=sel+1;

end if;

end process;

process(sel)

begin

case sel is

when 0 => we<="00000001";num<=temp0;

when 1 => we<="00000010";num<=temp1;

when 2 => we<="00000100";num<=temp2;

when 3 => we<="00001000";num<=temp3;

when others => we<="00000000";

end case;

end process;

process(num)

begin

case num is

when 0 => du<="11111100";

when 1 => du <="01100000";

when 2 => du <="11011010";

when 3 => du <="11110010";

when 4 => du <="01100110";

when 5 => du <="10110110";

when 6 => du <="10111110";

when 7 => du <="11100000";

when 8 => du <="11111110";

when 9 => du <="11110110";

when others =>du<="00000000";

end case;

end process;

end beh;