交通信号灯
设计报告

课程名称： 数字电子技术课程设计

指导教师： _____

小组成员： _____

专　　业： _____

2023年12月28日

目　录

1. 小组分工情况

负责项 / 小组成员	任务需求分析	代码设计	仿真测试	实验报告
	√	√	√	√
	√	√		√

2. 需求分析和系统设计

2.1 问题描述

使用quartus II设计基于VHDL语言的交通信号灯。

2.2 基本要求

- 设计一个十字路口的交通灯控制器，控制A,B两条交叉道路上的车辆通行，东西方向为主干道A，南北方向为副干道B；
- 每条道路设一组信号灯，每组信号灯有红、黄、绿3个灯组成，绿灯表示允许通过，红灯表示禁止通行，黄灯表示该车道上已过停车线的车辆继续通行，未过停车线的车辆停止通行；
- 主干道通行40秒，南北通行时间为20秒；
- 每次变换通行车道之前，要求黄灯先亮5s，才能变换通行车道。黄灯亮时，要求每秒闪烁一次。

2.3 需求分析

2.3.1 设计思路

- 设计两个计时器，分别表示主干道和副干道的信号灯倒计时；

- 设计一个总计时器，表示一整个信号灯周期的时间；
- 设置十字路口信号灯周期为110s；
- 一个周期：主干道直行绿灯（40s）→ 主干道直行黄灯（5s）→ 主干道左转绿灯（20s）→主干道左转黄灯（5s）→ 副干道直行绿灯（20s）→ 副干道直行黄灯（5s）→副干道左转绿灯（10s）→ 副干道左转黄灯（5s）。

2.3.2 端口说明

输入\输出端口	命名	功能介绍
输入端	clk	时钟
输入端	clrn	置零
输出端	port_red_main_straight	主干道直行红灯
输出端	port_green_main_straight	主干道直行绿灯
输出端	port_yellow_main_straight	主干道直行黄灯
输出端	port_red_branch_straight	副干道直行红灯
输出端	port_green_branch_straight	副干道直行绿灯
输出端	port_yellow_branch_straight	副干道直行黄灯
输出端	port_red_main_left	主干道左转红灯
输出端	port_green_main_left	主干道左转绿灯

输出端	port_yellow_main_left	主干道左转黄灯
输出端	port_red_branch_left	副干道左转红灯
输出端	port_green_branch_left	副干道左转绿灯
输出端	port_yellow_branch_left	副干道左转黄灯

3. 详细设计

3.1 数据输入转化实体

```
entity TrafficLight is
port(
    clk,clrn:in std_logic;
    port_cnt:out std_logic_vector(6 downto 0);

    --port_branch_cnt:out std_logic_vector(6 downto 0);--主干倒计时
    --port_main_cnt:out std_logic_vector(6 downto 0);--支干倒计时

    port_red_main_straight:out std_logic;
    port_green_main_straight:out std_logic;
    port_yellow_main_straight:out std_logic;

    port_red_branch_straight:out std_logic;
    port_green_branch_straight:out std_logic;
    port_yellow_branch_straight:out std_logic;

    port_red_main_left:out std_logic;
    port_green_main_left:out std_logic;
    port_yellow_main_left:out std_logic;

    port_red_branch_left:out std_logic;
    port_green_branch_left:out std_logic;
    port_yellow_branch_left:out std_logic
);
end TrafficLight;
```

这里对于左转/直行，红/绿/黄各个状态信号灯只会有0和1两个状态，因此声明为std_logic类型，在该灯应该亮的状态置为1，在不该亮的状态置为0。对于计数器由于一共存在110个周期，计数器的最大值为109，因此计数器的类型应至少为7位二进制信号。

3.2 内部结构的定义

在内部结构中定义相应的信号进行操作，在每个周期后将信号赋值给对应端口。为了更方便地进行数值操作，所以内部的计数器使用integer类型，在赋值时使用

port_cnt<=conv_std_logic_vector(cnt1,7)

函数将整型转化为7位二进制类型。

```
architecture bhv of TrafficLight is
signal cnt1,cnt2,cnt3,cnt4:integer;

--signal main_cnt:integer;
--signal branch_cnt:integer;

signal red_main_straight:std_logic;
signal green_main_straight:std_logic;
signal yellow_main_straight:std_logic;

signal red_branch_straight:std_logic;
signal green_branch_straight:std_logic;
signal yellow_branch_straight:std_logic;

signal red_main_left:std_logic;
signal green_main_left:std_logic;
signal yellow_main_left:std_logic;

signal red_branch_left:std_logic;
signal green_branch_left:std_logic;
signal yellow_branch_left:std_logic;
```

同时定义两个进程(process)

Branch_Light:process(clk,clrn,cnt2)

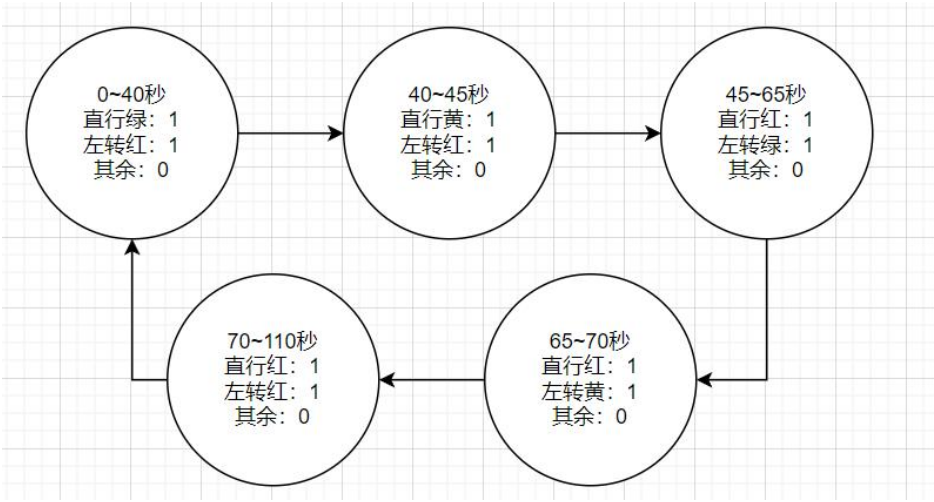
Main_Light:process(clk,clrn,cnt1)

分别使用cnt1和cnt2来计数，用于操作主干道灯的状态和副干道灯的状态。

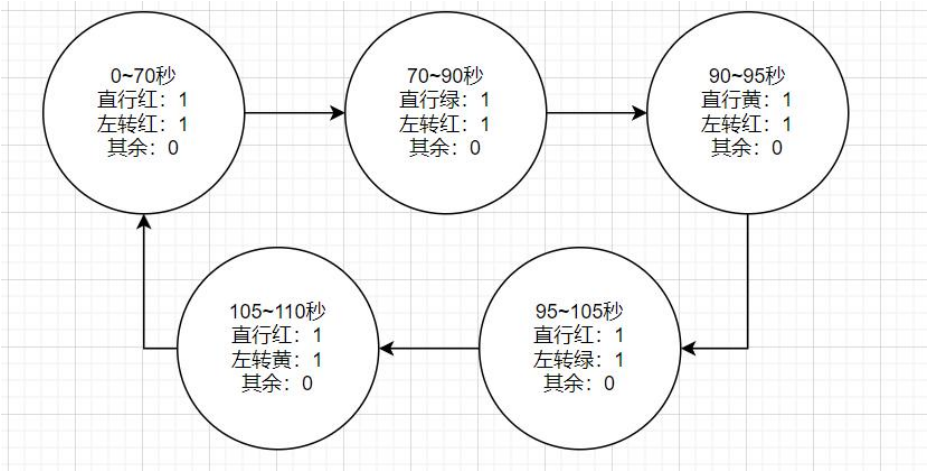
4. 系统实现

4.1 信号灯的状态转化

4.1.1 主干道状态图：



4.1.2 副干道状态图：



根据状态图使用if elsif then语句编写出对应的状态代码。

4.2 黄灯闪烁

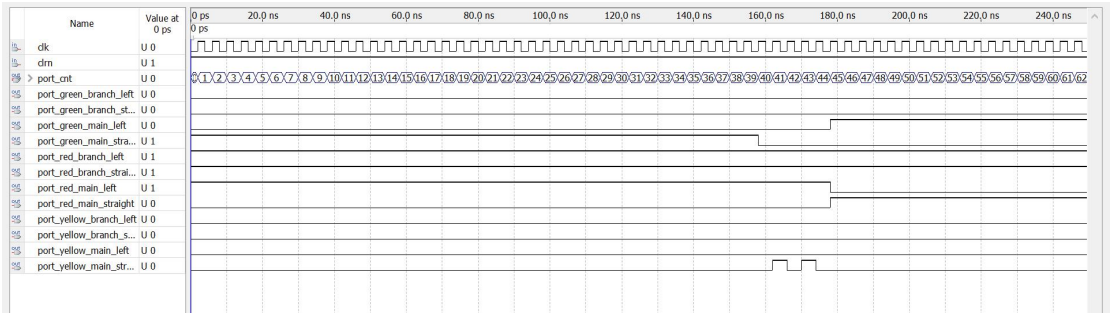
黄灯闪烁采用如下方式模拟：

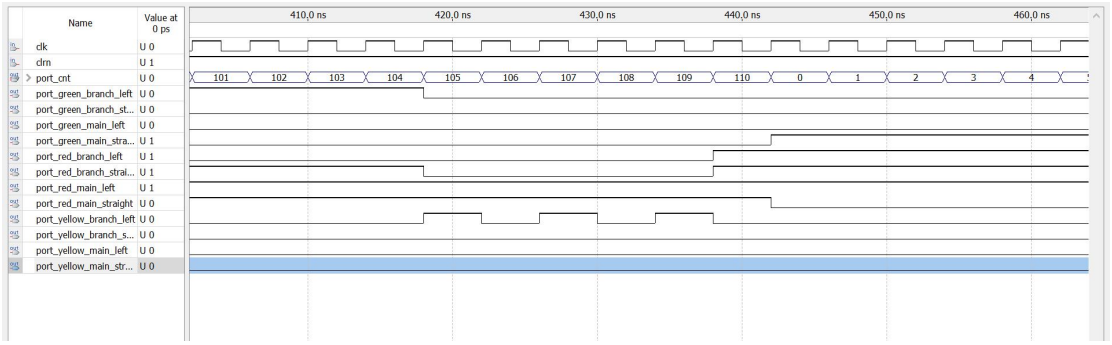
```
if cnt1 MOD 2 = 1 then
    yellow_main_left<='1';
else
    yellow_main_left<='0';
end if;
```

让黄灯在计数器为奇数时亮，在偶数时灭，以实现两秒一次的闪烁。

4.3调试分析

我们设置clk时钟周期为2ns，将接地端clrn设置为恒1，将整数进制显示设置为十进制无符号后进行仿真操作。





可以看到在在40秒，65秒，70秒，90秒，95秒，105秒，110秒时按需求进行了状态转换。在110秒之后计数器归零，进行下一个循环的操作。

5. 总结和展望

5.1 每个模块设计和调试时存在的问题和思考

原本想在程序中加入显示倒计时的功能，但每当加入这样一个端口，无论是否对它进行操作，在仿真的时候程序都会卡死，无任何反应。我们小组也试了很多种方法，但都没有任何效果。最后也很遗憾没有将此功能实现。

5.2 课程设计过程的收获

此次课程设计使用的是VHDL语言，这是一门从未接触过的语言，我们小组从零开始学习，并通过设计交通信号灯，掌握了它的基本应用方法，也同时提升了语言编程的能力。由于设计的复杂性，这个项目需要分工合作。我们小组通过一

人编写c++程序并把逻辑梳理清楚，另一人学习了解VHDL语言，再将c++的逻辑转译成VHDL实现了该程序。这也提升了我们的团队合作能力。

5.3 对数字电子技术这门课程的思考

- 基础性和重要性：数字电子技术是现代电子信息领域的基础，它涉及数字电路的设计、实现和应用。对于计算机、通信、嵌入式系统等方向的学生来说，这门课程是非常重要的基础。
- 硬件描述语言的应用：课程通常会涉及硬件描述语言（如VHDL、Verilog）的使用，这对于理解数字电路的结构和功能、进行数字系统的设计和仿真是至关重要的。
- 时序和同步：课程中通常涉及时序和同步问题，这对于理解数字系统的时钟控制、同步与异步电路设计等方面具有重要启发作用。
- 数字信号处理的基础：数字电子技术提供了理解数字信号处理的基础。在后续的专业课程或实际工作中，这些知识将为我们更深层次的学习和工作提供支持。

5.4 未来展望

我们想在未来的学习过程中解决本次课程设计中的遗憾，并把该程序完善得更加完美，把所有想要的功能全部实现。