



上海大学

SHANGHAI UNIVERSITY

2022-2023 学年夏季学期

课程报告

《计算机硬件综合大型作业报告》

小组序号： 2-2

项目名称： 交通灯控制器的设计

指导老师： 张云华

组员学号姓名：

组长： 李睿

组员 张琨璞

文信伶

贺新宇

许文博

计算机工程与科学学院

报告日期 2023 年 6 月 29 日

# 交通灯控制逻辑的设计

作者

21121082 张焜璞/ 21121083 李睿/ 21121131 文信伶/ 21121135 贺新宇/ 21121136 许文博

日期：2023 年 6 月 29 日

## 摘 要

本课程论文旨在设计一种基于 Quartus II 的交通灯控制逻辑。本论文首先介绍了交通灯控制系统的背景和重要性，接下来介绍了项目设计的思路。然后将整个电路分成计时模块、置数模块、显示模块、主控模块共四个模块进行介绍，最后是每个人的个人感悟。

关键词：Quartus II，交通灯，电路

## 目录

1 实践项目意义	3
2 设计任务及要求	3
3 项目设计思路	3
4 全局符号说明	5
5 各部分电路设计及参数分析	5
5.1 计时模块	5
5.1.1 计数器 74LS190 介绍	5
5.1.2 计时器的设计	6
5.1.3 红灯计时设计	7
5.2 置数模块	10
5.2.1 介绍	10
5.2.2 置数选择	11
5.3 显示模块	12
5.3.1 功能说明	12
5.3.2 符号解释	13
5.3.3 强制显示电路	13
5.3.4 choseR 芯片——8 位 2 选 1 芯片	13
5.3.5 超过 99s 的显示	15
5.4 主控模块	16
5.4.1 符号解释	16
5.4.2 实现内容概述	16
5.4.3 TL 芯片——译码器	17
5.4.4 双闪功能	18

5.4.5	双向红灯 . . . . .	18
5.4.6	单向通行 . . . . .	18
<b>6</b>	<b>工作过程分析</b>	<b>18</b>
<b>7</b>	<b>项目调试过程</b>	<b>18</b>
<b>8</b>	<b>个人感悟</b>	<b>18</b>
8.1	张焜璞 . . . . .	18
8.2	李睿 . . . . .	19
8.3	文信伶 . . . . .	20
8.4	贺新宇 . . . . .	21
8.5	许文博 . . . . .	21
<b>9</b>	<b>参考文献</b>	<b>22</b>

## 1 实践项目意义

随着人口和车辆的日益增长，市区交通也日益拥挤。人们的安全问题也日益重要。交通信号灯的出现有效的解决了道路安全和拥挤问题，成为交管部门管理交通的重要工具之一。交通信号灯常用于十字路口。通过合理的设计和使用，交通信号灯为道路使用者提供了明确的指示，降低了交通事故的发生风险，减少了交通拥堵，提高了交通运行效率，促进了交通系统的可持续性，创造了更安全、便捷和环保的交通环境。大部分城市中十字路口交通灯的控制普遍采用固定转换时间间隔的控制方法，由于十字路口不同时刻车流量是复杂的、随机的，采用固定时间的控制方法，经常会造成道路有效时间的浪费，出现空等现象，影响了道路的畅通。本实践项目所设计的交通信号灯具有时间可变的功能，能够根据不同道路情况调节红绿灯的时间，进而改善交通通行的效率，并为根据实时流量变化情况设计自适应控制的交通灯提供基础。

## 2 设计任务及要求

由一条主干道和一条支干道的汇合点形成十字交叉路口，为确保车辆安全、迅速地通行，在交叉路口的每个入口处设置了红、绿、黄三色信号灯。该交通灯利用 QuartusII 完成原理图设计和电路仿真、调试，并在 xx 实验箱上进行实验验证及调试。

### 1. 设计任务

- 1) 十字路口，交通灯由绿、黄、红灯组成一组灯，南北和东西各一组；
- 2) 二位数码管倒计时显示。

### 2. 设计要求

- 1) 可设定亮灯时间
- 2) 可设定不同的亮灯模式：正常方式、单向绿灯（分为南北绿灯和东西绿灯两种）、双向黄灯闪烁、双向红灯
- 3) 注意倒计时超过 100 秒情况下数码管的显示

## 3 项目设计思路

经查阅相关资料，并结合设计要求，交通信号灯的设计按照功能可分为四个模块，分别是：置数系统、定时系统、主控系统和显示系统，其中本项目的关键在于四种状态的正确转换及对应的倒计时显示实现。该模型系统框架图如图 1 所示：

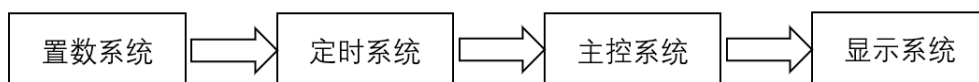


图 1: 交通信号灯系统模型框架图

1. 置数系统：实现各方向信号灯亮灯时间的任意设定由 4 片 74LS373 锁存器芯片存储两组方向上所设定的黄绿灯时间，并对各芯片进行编码，通过 3-8 译码器实现芯片的选择，即可实现对指定方向信号灯时间的输入。
2. 定时系统：实现各方向信号灯的倒计时使用两片可逆十进制计数器 74LS190 芯片级联实现 0 99 内的减法计数，并利用 7476 芯片的输出信号选择对应方向信号灯的倒计时显示转换。判断时间是否

减为 00，若减为 00 则通过输出信号传入 7476 芯片，再由 7476 芯片输出结果作为使能信号，选择 4 片时间锁存器中的一片，将设定的时间打入计时器中实现下一状态的倒计时计数。

3. 主控系统：通过接收来自定时系统的信号作为时钟信号，实现正常情况下状态信号灯的转换。由设计要求可知，主控系统需要控制南北方向红绿黄灯和东西方向红绿黄灯的亮灭，一共需要六种状态。但实际上，可以通过对立方向上黄绿灯的控制信号相或取反来控制红灯（即一个方向上黄绿灯都熄灭时，另一个方向上红灯一定处于亮的状态），故可简化为四种状态：南北方向红灯亮，东西方向绿灯亮（ $t_1$  秒） 南北方向红灯亮，东西方向黄灯亮（ $t_2$  秒） 南北方向绿灯亮，东西方向红灯亮（ $t_1$  秒） 南北方向黄灯亮，东西方向红灯亮（ $t_2$  秒）。使用两个 JK 触发器级联即可实现四种状态的控制转换，通过译码器实现状态译码。而 3 种特殊功能（双向红灯、单向绿灯、双向黄灯闪烁）则可通过逻辑门电路组合设计实现。

Q1Q0 是对四种状态的编码，对应关系如表 xx 所示：

Q1Q0	南北	东西
00	红灯亮	绿灯亮
01	红灯亮	黄灯亮
10	绿灯亮	红灯亮
11	黄灯亮	红灯亮

各个状态的转换关系如图 2 所示，圈中数字为四种状态对应的编码，箭头上方的数字表示各状态转换的条件 T2 代表  $t_2$  秒计时结束的信号、T1 代表  $t_1$  秒计时结束的信号，'1' 代表条件成立，'0' 代表条件不成立，X 代表条件成立与否随机（状态 '0' 或状态 '1' 皆可）

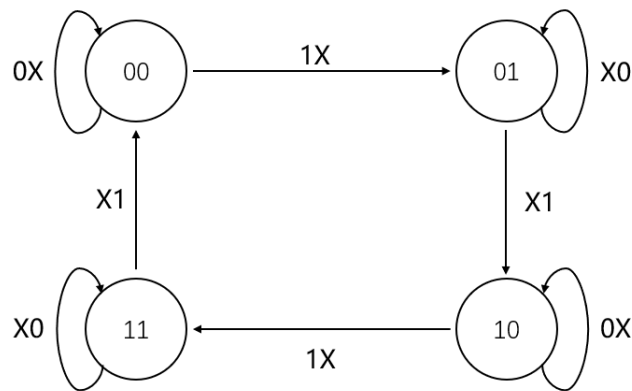


图 2: 状态转换图

由状态转换图画出自状态转换表，由状态转换表得到状态转换方程，再通过与 JK 触发器的特性方程比较，将状态转换方程化为 JK 触发器的标准特性方程，即可设计出状态转换控制器。

根据南北向红灯倒计时时间等于东西向黄灯和绿灯时间之和，东西向红灯时间等于南北向黄灯和绿灯时间之和，设计时只需输入两组方向黄灯和绿灯的时间。红灯倒计时的显示首先需要判断当前对立方向亮黄灯还是绿灯，再根据八位加法计数器的输出作为红灯时间。若当前计时器中进行绿灯计时，则加法器的两组输入分别为计时器中的时间和锁存器中对立方向的黄灯时间；若为黄灯计时，红灯计时等于黄灯计时，即加法计数器的两组输入分别为 0 和计数器中的时间。

## 4 全局符号说明

符号	功能
CLK	时钟输入端（上升沿有效）
CLRN	强制显示信号
GND	接地
VCC	高电平信号

## 5 各部分电路设计及参数分析

### 5.1 计时模块

#### 5.1.1 计数器 74LS190 介绍

引脚及功能

引脚	功能说明
LDN	异步并行置数输入控制端（低电平有效）
A D	并行数据输入端
GN	计数控制端（低电平有效）
DNUP	加/减计数方式控制端
CLK	时钟输入端（上升沿有效）
QA QD	输出端
MXMN	最大/最小计数输出端
RCON	行波时钟输出端（低电平有效）

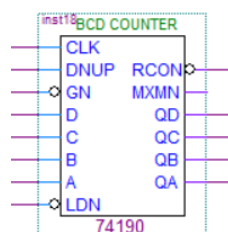


图 3: 74LS190 芯片引脚图

**工作原理** 74190 芯片的预置为异步的。当置入控制端（LDN）为低电平时，无论时钟 CLK 的状态如何，输出端 (QA QD) 即可预置成与数据输入端 (A D) 相一致的状态。

74190 的计数是同步的，依靠 CLK 加在 4 个触发器上而实现。当计数控制端 (GN) 位低电平时，在 CLK 上升沿作用下 QA QD 同时变化，从而消除了异步计数器中出现的计数尖峰。当计数方式控制 (DNUP) 为高电平时进行减计数。只有当 CLK 为高电平时 GN 和 DNUP 才可以跳变。

74190 有超前进位功能。当计数溢出时，进位/错位输出端输出一共低电平脉冲，其宽度为 CLK 脉冲周期的高电平脉冲；行波时钟输出端 (RCON) 输出一个宽度等于 CLK 低电平部分的低电平脉冲。利用 RCON 端，可以级联成 N 位同步计数器。当采用 CLK 控制时，则将 RCON 接到后一级 GN；当采用并行 GN 控制时，则将 RCON 接到后一级 CLK。

## 5.1.2 计时器的设计

### 计时器电路图

根据 74190 芯片的工作原理，可由两个 74LS190 级联构成两位十进制计数器，如图 4 所示：

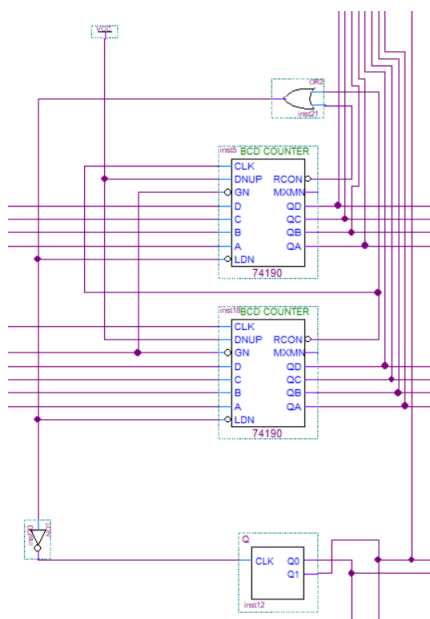


图 4: 计时器电路设计图

### 原理解释

给两片 74190 芯片的加/减计数控制信号端 DNUP 加高电平，使计数器为减计数状态。两个计数器的使能输入端由外部输入信号与低电平信号相或控制，可以实现手动控制计数器是否开始工作。由于当计数器中的内容减为 0 时，RCON 端将发出低电平信号，通过将高位计数器的 CLK 端与低位计数器的 RCON 端相连，每当低位计数器减为 0 时高位计数器才工作减 1，低位没有减为 0 时，高位计数器被“锁”，即可实现减法借位。计数器置位端 LDN 都由低位和高位计数器的行波时钟输出端 RCON 的信号相或进行控制，即只有当两个计数器都减到零时才可以对计时器进行重新置数，而当低位计时器减为零时不进行置数，从而实现信号灯的转换计时。每次置数成功后，RCON 信号将变为高电平，因此将停止计数和状态变换，保证计时器能够顺利工作。

### Q 芯片——状态转换器

#### 激励方程

$$J_0 = K_0 = 1$$

$$J_1 = K_1 = Q_0$$

#### 实现原理

利用 7476 芯片级联的两个 JK 触发器的输出作为选择信号将相应信号灯显示时间打入计时器中（初态为 00）。由于 JK 触发器为上升沿有效，所以当两块计数器都减为 0 时，RCON 信号相或产生低电平信号，再将信号取反后与两个 JK 触发器的时钟 CLK 相连，实现每当某信号灯计时为 0 后将输出 Q1Q0 信号，对 Q1Q0 进行编码，Q1Q0 代表四个锁存器芯片，从而将选定芯片中锁存的信号灯时间置入计时器中，实现信号灯计时的循环切换，具体电路图如图 5 所示：

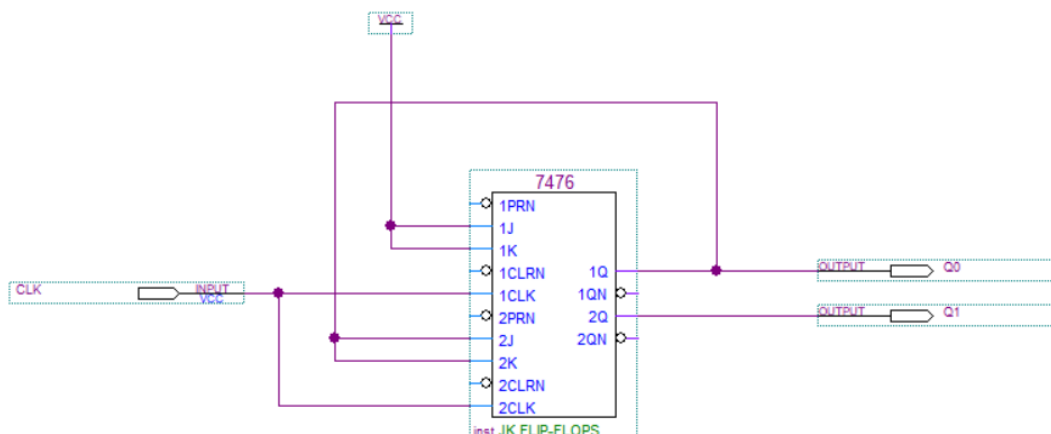


图 5: Q 芯片内部电路图

### 5.1.3 红灯计时设计

黄灯和绿灯的计时可直接通过计时系统实现。由于置数系统的设计中，没有设计红灯显示时间的置数操作，故红灯计时利用了红灯时间与对立方向黄灯和绿灯时间之间的关系所实现。

$$\text{红灯亮灯时间} = \text{黄灯亮灯时间} + \text{绿灯亮灯时间}$$

根据四种状态的转换过程可知，红灯计时时间有两种情况，以南北向亮红灯为例。当东西向为红灯时，计算同理：

1. 当东西向为绿灯，南北向为红灯时：红灯计时时间 = 绿灯计时时间 + 黄灯设定时间；
2. 当东西向为黄灯，南北向为红灯时：红灯计时时间 = 黄灯计时时间；

通过上述分析可知，红灯计时时间始终包含黄灯/绿灯计时时间，因此加法器（加法器的原理图如图 7 所示）的其中一组八位输入端与高低位计数器的输出端依次相连。是否要加上黄灯的设定时间则由当前信号灯的状态有关，因此另一组八位输入端与选择芯片 choseGY 的输出相连。choseGY 芯片原理图如图 xx 所示，choseGY 是一个 4 选 1 选择器。choseGY 的其中两组输入与低电平相连，代表时间为 0，另外两组输入分别与两组方向的黄灯时间锁存芯片的输出端相连，将 JK 触发器的输出端 Q1Q0 与 choseGY 芯片的状态输入端相连，对 Q1Q0 进行编码，分别表示计时器中为南北向绿灯、南北向黄灯、东西向绿灯、东西向黄灯。当 Q1Q0 代表南北向绿灯时，choseGY 输出为东西向黄灯设定时间；当 Q1Q0 代表东西向绿灯时，输出为南北向黄灯设定时间；当 Q1Q0 代表南北向黄灯或东西向黄灯时，输出皆为 0。红灯计时设计原理图如图 6 所示：



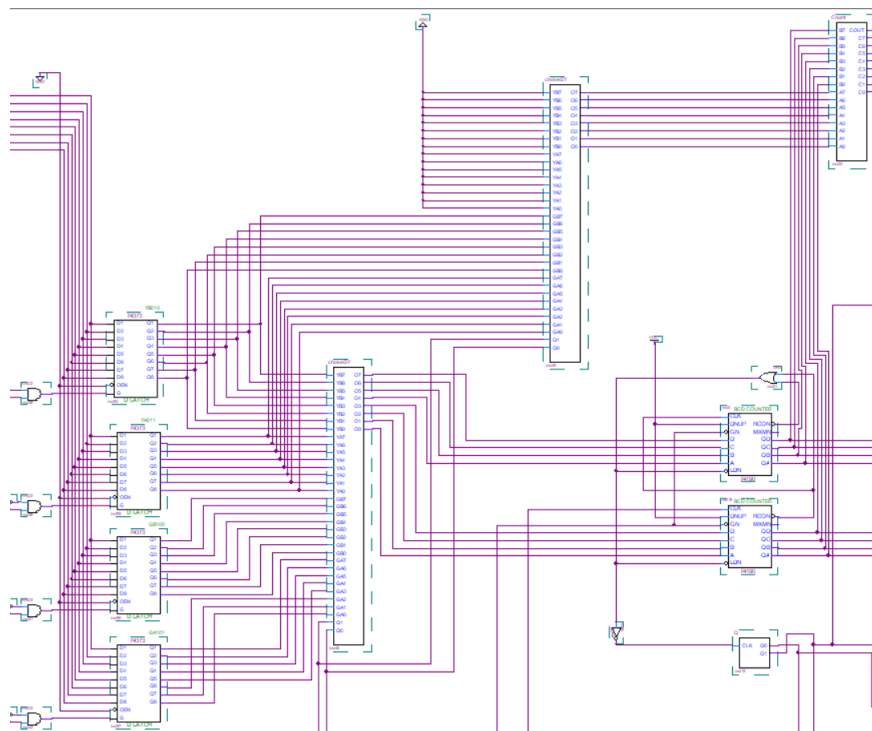


图 6: 红灯计时设计电路图

### Count 芯片——两位十进制加法器

提供两个 BCD 码，计算他两加起来的实际值。输出进位符以及八位 BCD 码。根据加六校正的方法

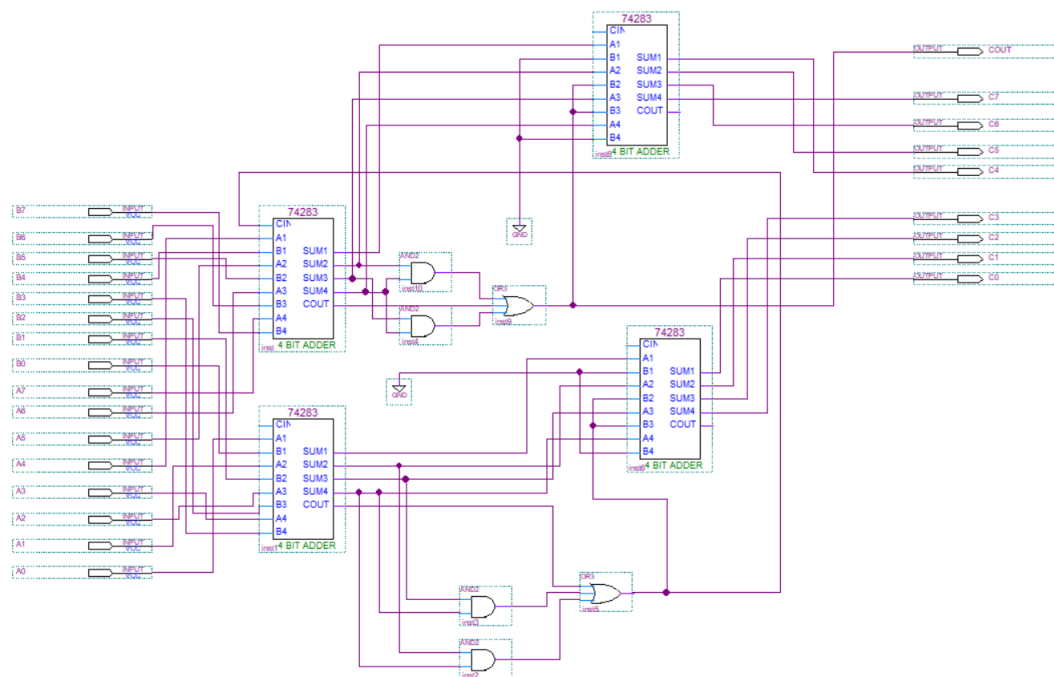


图 7: 两位十进制加法器电路设计图

### choseGY 芯片——8 位四选一

choseGY 芯片内部由八块 4 选 1 芯片 (choseGY) 构成

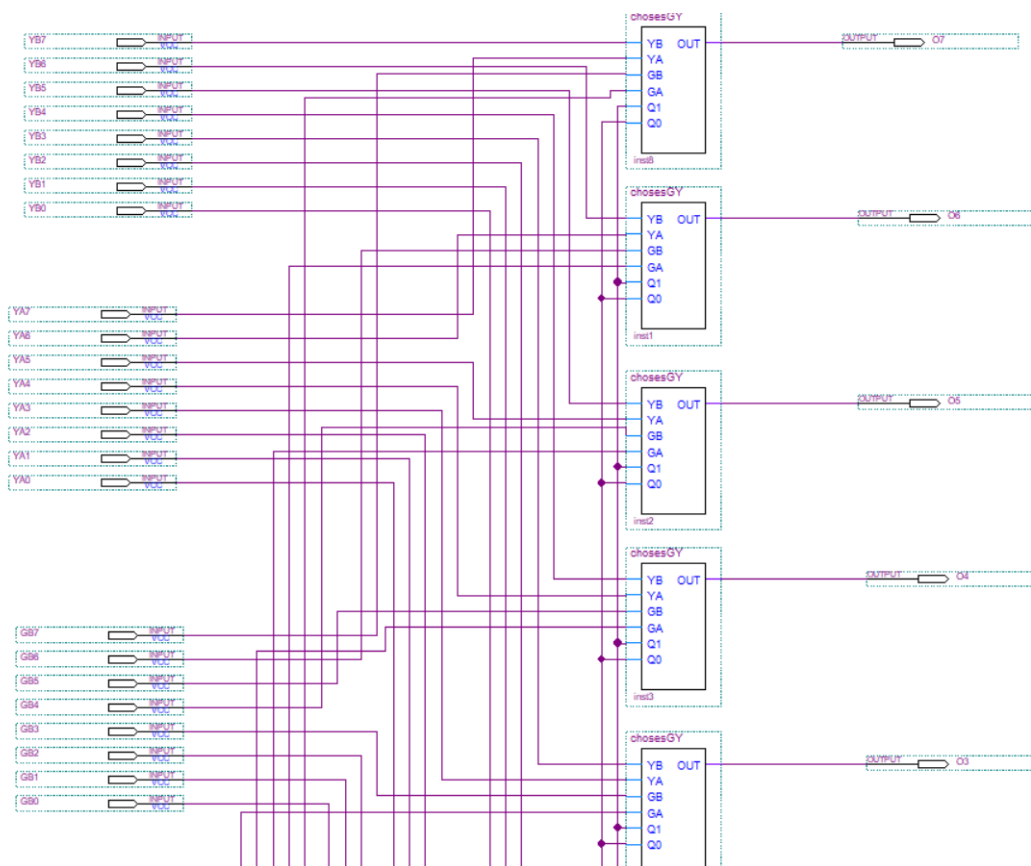


图 8: choseGY 芯片

choseGY——4 选 1 芯片

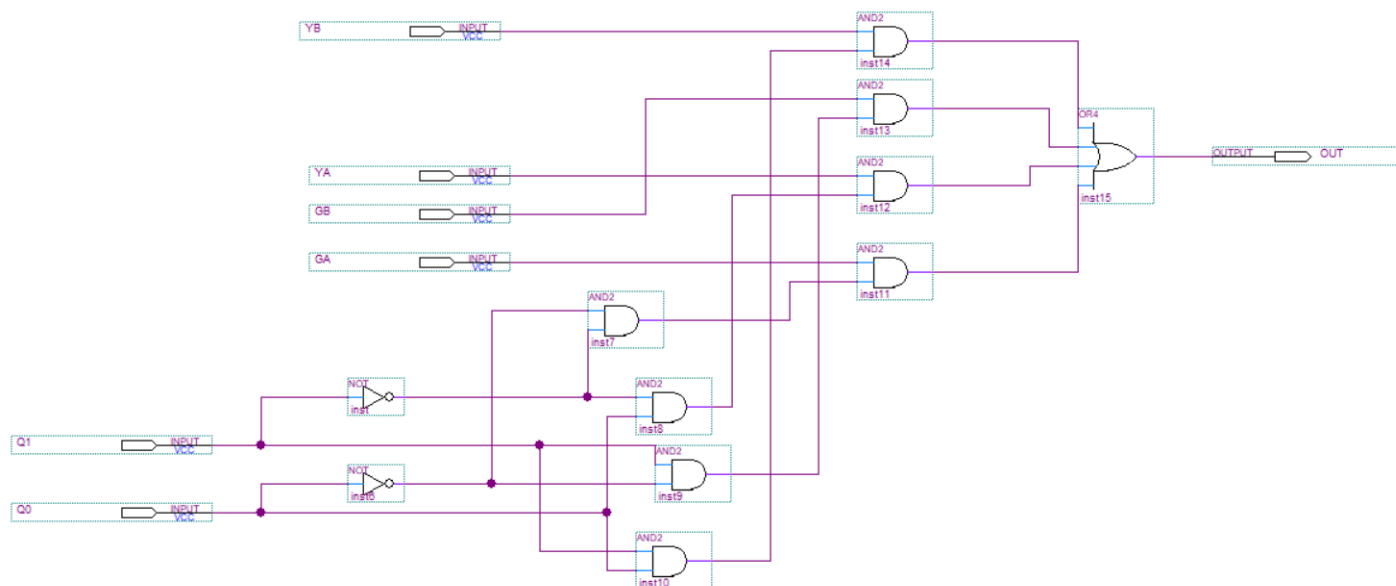


图 9: 4 选 1 芯片

功能表

Q1	Q0	选择
0	0	GA
0	1	YA
1	0	GB
1	1	YB

## 5.2 置数模块

### 5.2.1 介绍

首先是对于我们整个电路中存储单元的介绍，这一部分电路的思路是将东西南北四个方向的红黄绿灯的倒计时时间存储在一个锁存器里，整个电路由 12 个 input 输入端，一个 3-8 译码器，和四块 74373 锁存器以及若干门电路组成。

其中 12 个输入端中，有 8 位是作为倒计时数据输入端，还有 3 位用来作为 3-8 译码器的输入端，剩余的 1 位作为确认信号，后续会用来作为存储器的辅助选通信号。因为设计交通灯时，涉及到多个方向和每个方向的三种颜色的交通灯，所以在仔细思考后，我们决定选用四块 74373 锁存器来存放交通灯的倒计时时间。与此同时，因为有多块存储器，在数据打入的这个阶段，我们就要对存储器进行选择，对此我们采用了 74138 这块 3-8 译码器器件，由图 1 可以看到我们设计的 3-8 译码器是 A、B、C 作为输入端，Y2N、Y3N、Y4N、Y5N 作为选择存储器的输出。

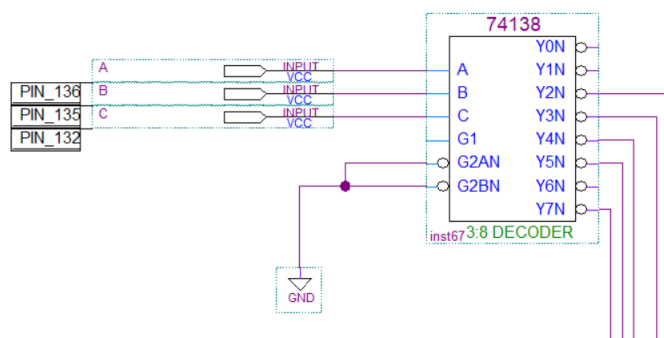


图 10: 译码器各引脚图例

对于存储器我们选择了 74373 这样一块锁存器，74273 具有 8 个独立的锁存器，每个锁存器可以存储一个 8 位的二进制数据。锁存器可以通过输入信号进行锁存或释放操作。当锁存器被锁存时，它会保持存储的数据不变，即使输入信号发生变化也不会影响锁存器的输出。74273 还具有锁存控制端，可以通过该端口控制锁存器的锁存和释放操作。当锁存控制端为高电平时，锁存器被锁存，即锁定存储的数据。当锁存控制端为低电平时，锁存器被释放，即允许输入信号改变锁存器的输出。由图 2 可知，D1-D8 作为数据输入端，当时间数据被打入到锁存器中就可以将数据保存在里面，当电路中的其余部件需要这个数据时，在通过 Q1-Q8 将数据输出。其中我们将 G 接口作为锁存芯片使能端，将 OEN 接口设置为接地端，令其与 GND 相连。

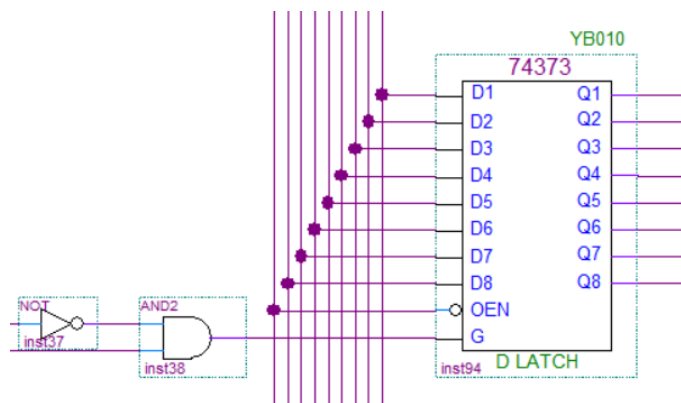


图 11: 锁存器各引脚图例

### 5.2.2 置数选择

那么既然有四块存储器芯片，并且这四块存储器并不能同时使用，所以我们就要对使用的芯片进行选择，这就是选择 74138 这一 3-8 译码器的原因。而我们只使用了 3-8 译码器的四个输出端对四块存储器进行选择，但是最初是设计了六块存储器，在后来的更新迭代后，最终只保留下了四块存储器并沿用原本的设计思路，因此在存储器的编码部分的顺序上会乱一些。如图 3 所示，对每个存储器进行编码，依靠对 3-8 译码器的输入译码后选择相应存储器。

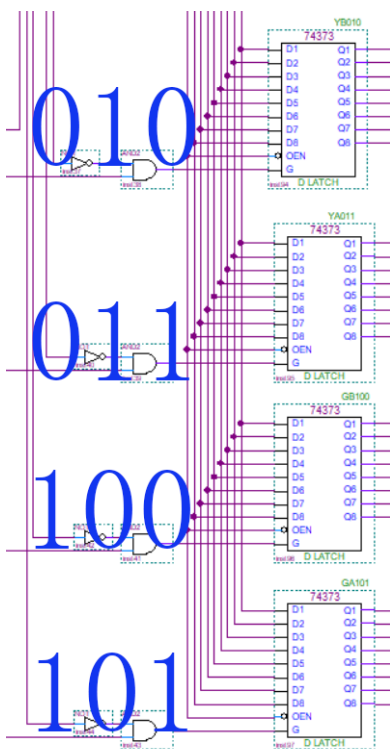


图 12: 存储器的编码

为了进一步保证数据打入的稳定性与正确性，我们还额外设计了一个 choose 使能端，从图 4 中的门电路部分可以看出，存储器的 G 使能端信号来自一个与门给出的信号，这就是我们设计的独特之处，存储器的正常使用不仅仅需要 3-8 译码器的选择信号，还需要人为输入一个使能确认信号，也就是 choose 输入的信号，两个信号同时有效时，对应的存储器被选中并将数据打入到该存储器中。

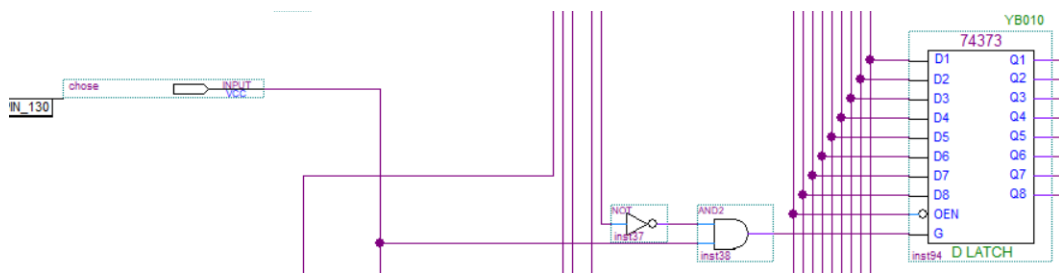


图 13: choose 使能端以及门电路设计

因此这一部分电路的主要工作步骤就是，在预置好时间数据之后，通过在译码器输入端输入各个存储器对应的二进制序列号，译码后给出选择信号，再通过手动给 choose 端有效输入，两个信号做与运算后，打入选中存储器的使能端，该存储器被选中后，将时间数据存储到该位置中。

### 5.3 显示模块

显示模块的电路图如图 12

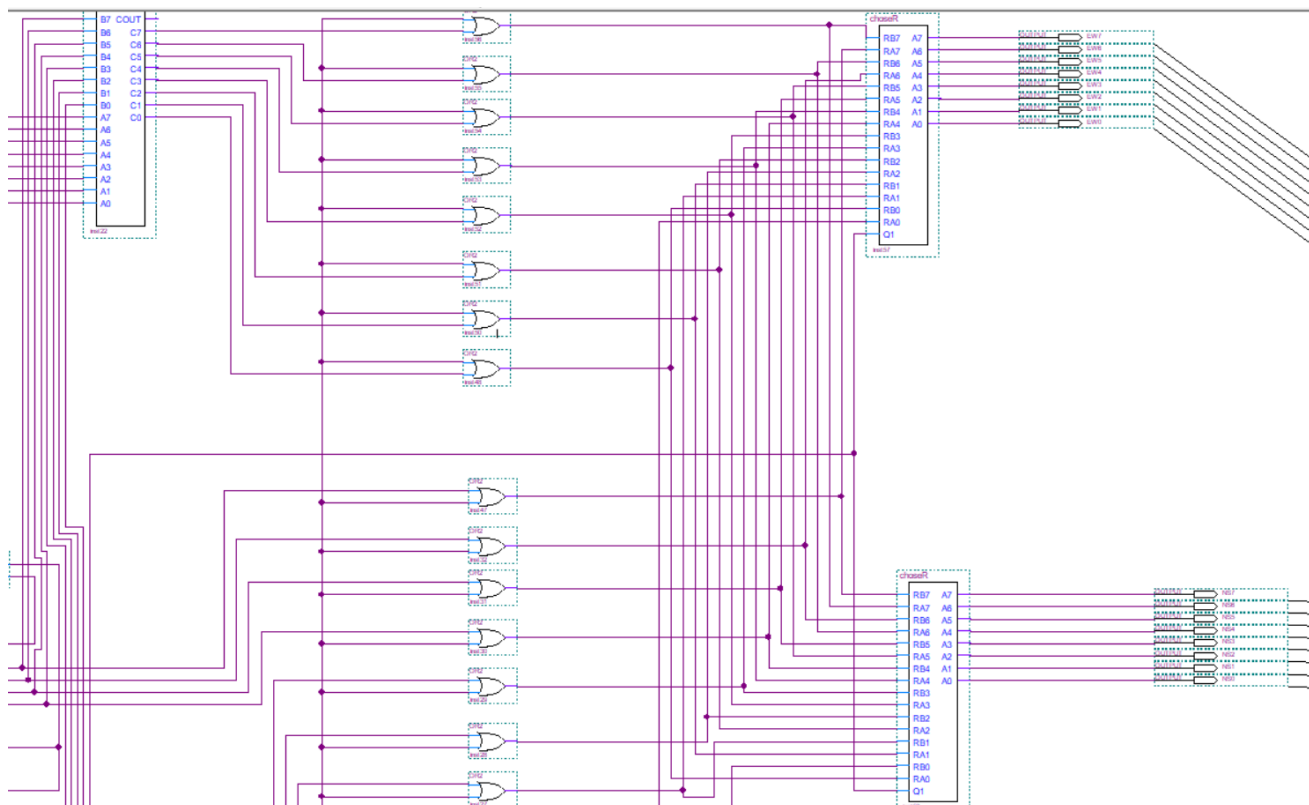


图 14: 显示模块电路图

#### 5.3.1 功能说明

显示模块用来显示信号灯的倒计时

显示模块接收计数模块的数据——红灯倒计时和黄绿灯倒计时，分别为 8 位二进制。然后通过两个 choseR 芯片完成二选一，选择是显示红灯倒计时还是黄绿灯倒计时，左边一竖列的或门是用来决定是否强制显示的。

5.3.2 符号解释

符号	功能说明
EW7-EW0	东西方向的倒计时 (8 位二进制)
NS7-NS0	南北方向的倒计时 (8 位二进制)
RB7-RB0	计数模块的红灯倒计时
RA7-RA0	计数模块的黄绿灯倒计时

5.3.3 强制显示电路

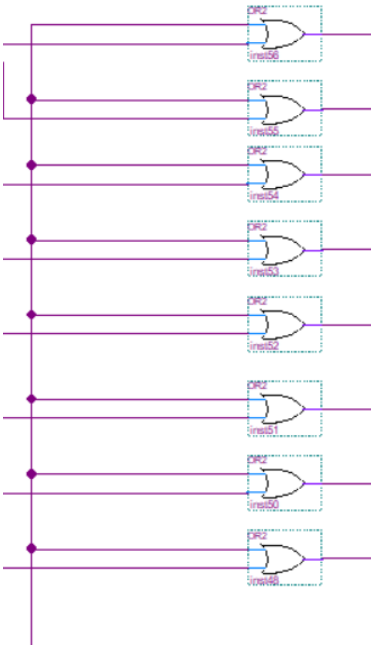


图 15: 强制显示电路

这里左边接在了 CLRN 上，当 CLRN 被激活，那么两边的信号通过或门之后都为高电平，达到屏蔽倒计时的效果

5.3.4 choseR 芯片——8 位 2 选 1 芯片

choseR 芯片可以根据 Q1 的状态，输出两种 BCD 码中的一种。

引脚图

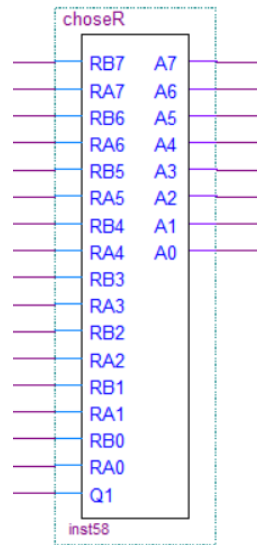


图 16: 2 选 1 芯片 (8 位)

### 内部电路

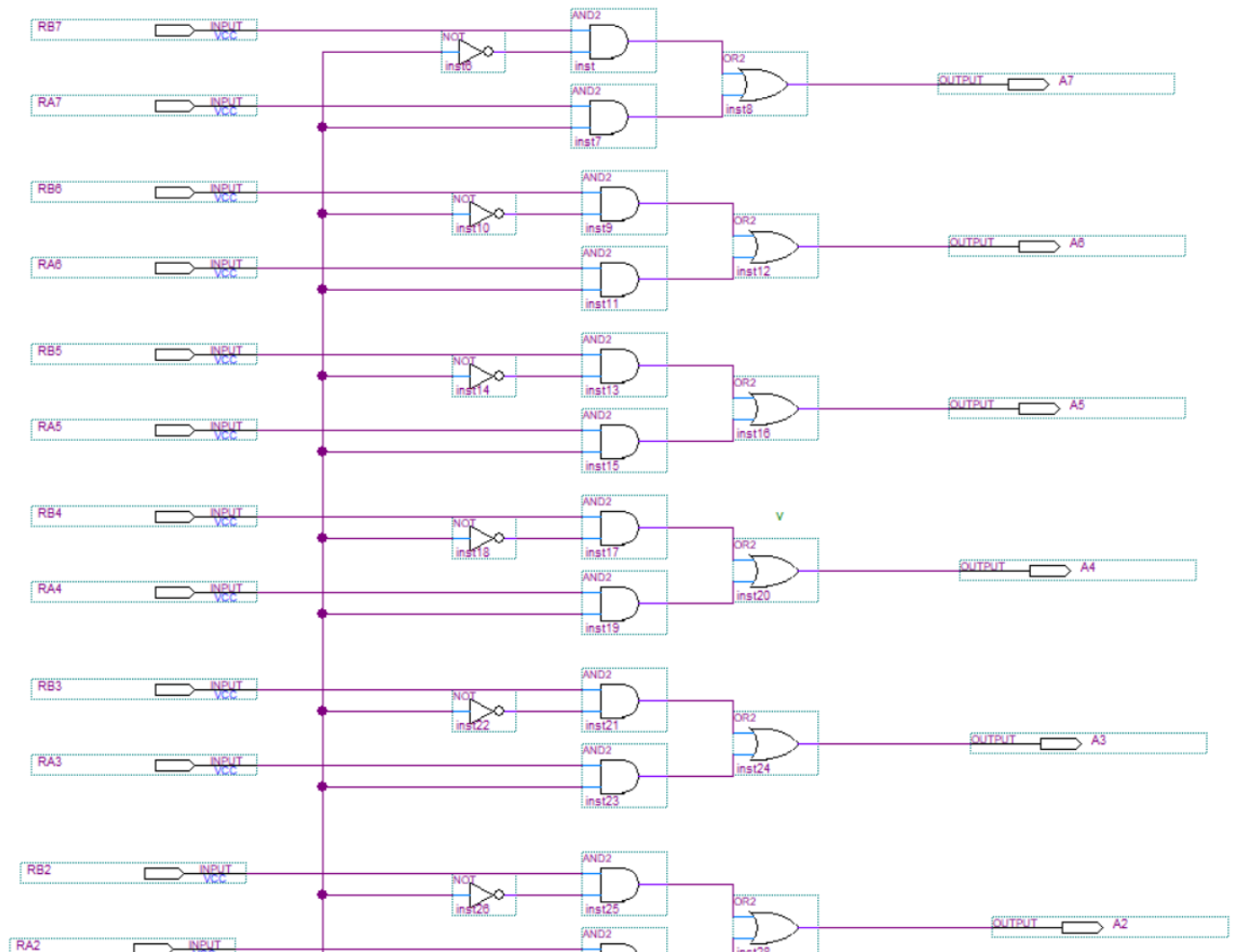


图 17: 2 选 1 芯片 (8 位) 内部结构

功能

- Q1 为 0 时，选择 RB
- Q1 为 1 时，选择 RA

### 5.3.5 超过 99s 的显示

在第一次设计时没有考虑倒计时超过 99s 的情况，加上这个功能只需要对之前的电路做很小的改动。

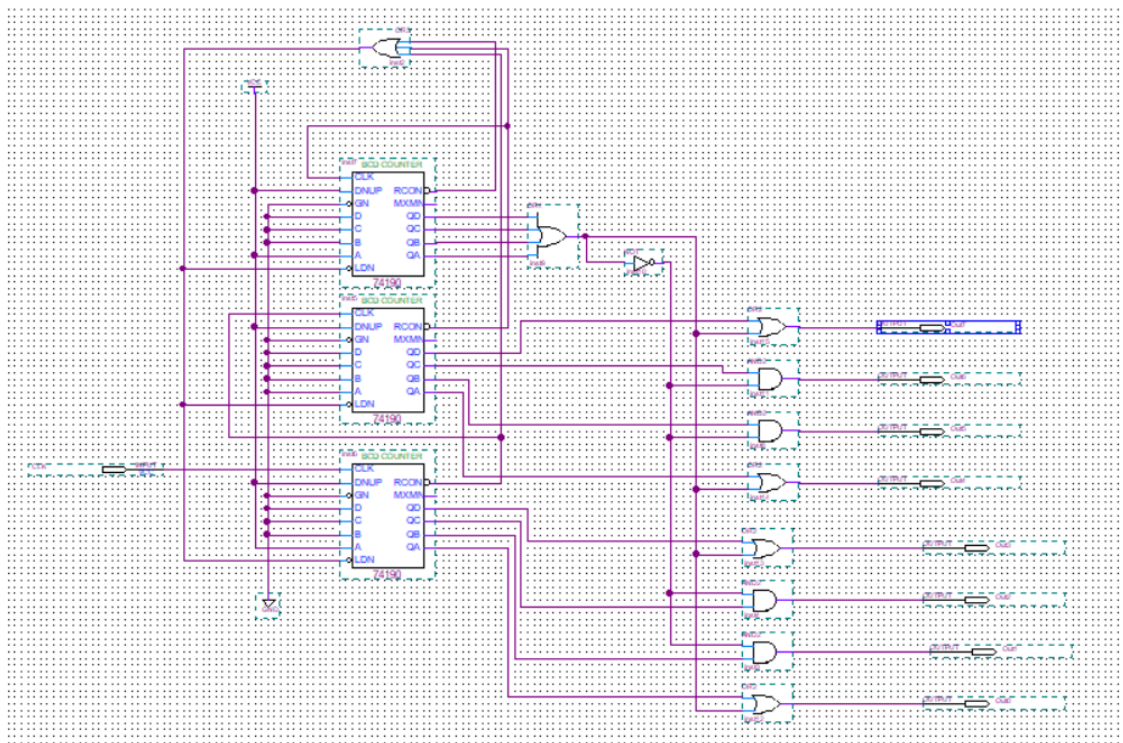


图 18: 显示扩充

将计数器进行如下的更改即可。为方便演示，这里先设置初值为 101。其波形图如下：

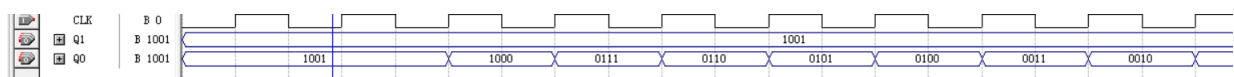


图 19: 扩充后的波形图

如图所示，在 101 与 100s 时，显示 99s，之后计数器正常显示。实现了超过 99s 的显示。



## 5.4 主控模块

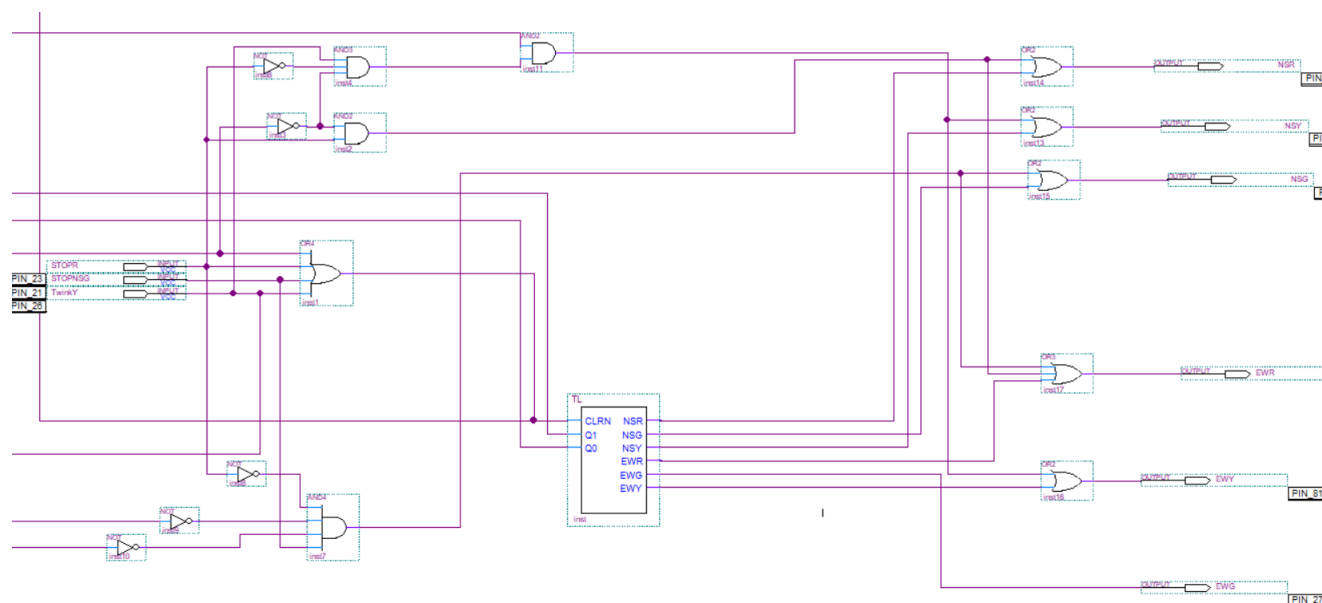


图 20: 主控模块电路图

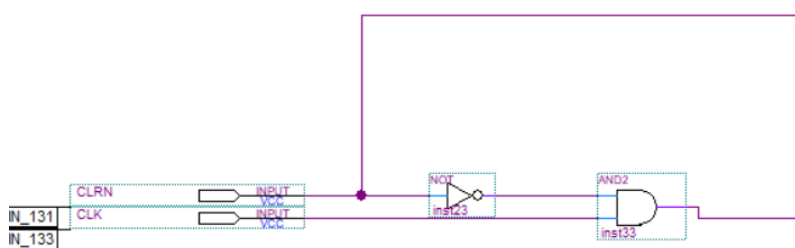


图 21: 时钟信号部分

### 5.4.1 符号解释

符号	含义
STOPR	双向红灯
STOPNSG	单向红灯单向绿灯
TwinkY	双向黄灯双闪
CLK	时钟信号
CLR	用于截断时钟信号，进行强制操作

### 5.4.2 实现内容概述

通过一些简单的门电路来控制哪一盏灯亮，而显示的时间由计时模块完成。  
 通过 **TL** 芯片，根据 Q1Q0 的状态，为红黄绿灯提供显示信号。  
 当 **CLR** 信号输入 **TL** 芯片时，**TL** 芯片的所有输出都被关掉，所有灯都取消显示。  
 对于双闪功能，是通过时钟信号和黄灯信号进行与运算实现的。

### 5.4.3 TL 芯片——译码器

根据 Q1Q0 的状态，为红黄绿灯提供状态：

引脚图

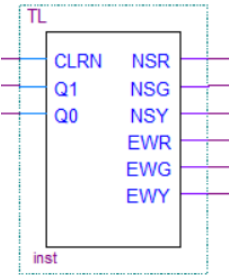


图 22: TL 芯片

引脚符号解释

符号	功能说明
CLRn	清零端，当为高电平时输出全为低电平
Q0/Q1	接在状态转换器上，控制状态转换
NSR	南北方向绿灯
NSG	南北方向红灯
NSY	南北方向黄灯
EWR	东西方向红灯
EWG	东西方向绿灯
EWY	东西方向黄灯

内部电路图

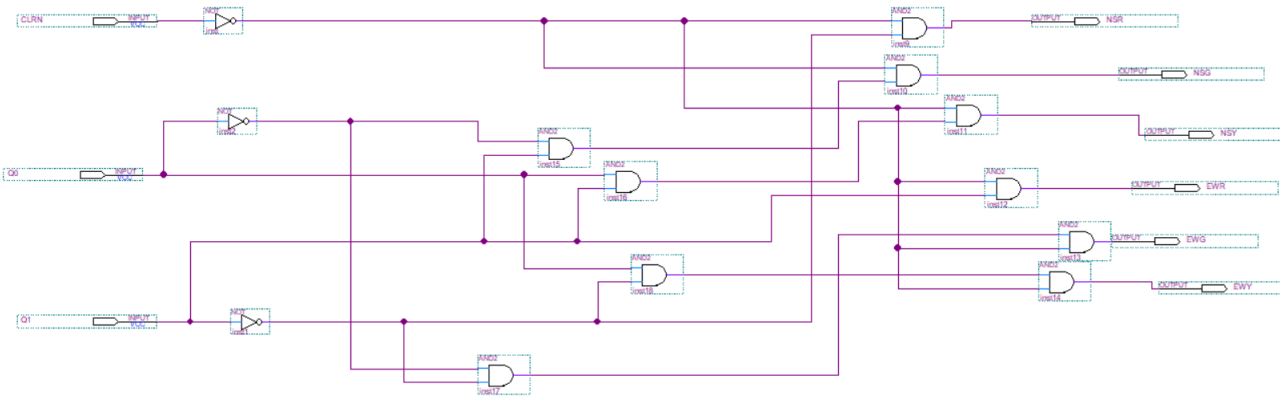


图 23: TL 芯片内部结构

## 功能表

CLRN	Q1	Q0	NSR	NSG	NSY	EWR	EWG	EWY
1	x	x	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0
0	0	1	1	0	0	0	0	1
0	1	0	0	1	0	1	0	0
0	1	1	0	0	1	1	0	0

### 5.4.4 双闪功能

图 16 中，时钟信号接在了与门 inst11 的左边。

当 TwinkY 为高电平时，此时 TL 芯片的 CLRN 输入端为高电平，TL 芯片的输出全为低电平。

当时钟信号也为高电平时，与门 inst11 打通，然后或门 inst16 打通，黄灯亮；当时钟信号为低电平时，与门 inst11 关闭，或门 inst16 关闭，黄灯灭。

这样就完成了黄灯闪烁功能的设计。

### 5.4.5 双向红灯

当 STOPR 置为高电平时，或门 inst1 打通，TL 芯片的 CLRN 输入端为高电平，TL 芯片的输出全为低电平。同时非门 inst3 打通，与门 inst2 打通，或门 inst14 和 inst17 打通，南北方向和东西方向红灯亮起。

这样就完成了双向红灯的设计。

### 5.4.6 单向通行

因为实验箱的控制拉杆不够用了，所以只设计了一个方向的，另一个方向的同理。

当 STOPNSG 置为高电平时，或门 inst1 打通，TL 芯片的 CLRN 输入端为高电平，TL 芯片的输出全为低电平。同时与门 inst7 打通，或门 inst15 和 inst17 打通，南北方向绿灯亮起，东西方向红灯亮起。

这样就完成了单向通行的设计。

## 6 个人感悟

### 6.1 张焜璞

在这两周的实训中，我有幸参与了一项硬件大型作业实训项目，我更进一步的理解 quartus 软件的使用。更进一步的理解了数字逻辑。还让我意识到了团队合作和解决问题的重要性。在这段时间里，我积累了丰富的经验，获得了许多宝贵的教训。首先，这个实训项目让我意识到了硬件设计的复杂性。在实际应用中，硬件的设计和开发需要考虑到许多因素，如性能要求、功耗、成本和可靠性等。通过实际操作，我深刻体会到了这些因素之间的平衡和权衡。我学会了通过合理的架构设计和优化来满足不同的需求，同时确保系统的稳定性和可靠性。在实训项目中，我与团队成员、指导教师以及其他相关人员保持着紧密的沟通。良好的沟通可以帮助我们更好地理解需求、协调工作、解决问题。我学会了如何清晰地表达自己的观点和想法，同时也学会了倾听和理解他人的需求和意见。这个实训项目让我体会到了从理论到实践的转变。在课堂上，我们学到了许多理论知识，但只有通过实际操作和实践才能真正理解和应用这些知识。通过实训项目，我有机会将所学的理论知识应用到实际项目中，不断调整和改进，从而提高自己的技能和能力。本次实训我们前四天都去实验室进行了工作，我们实训题目难度较大，这使得我们不得不多花点时间以及精力去对待这一次的实训。虽然说在这四天当中，我们废寝忘食（指中午觉和

中午饭没去吃留在实验室进行测试),不断的去推断,不断的去寻找错误并订正错误。一遍又一遍推翻自己的已有的思路,一遍又一遍的重建新的思路。最终在第四天的中午我们实现了实训题目,那一刻我们十分的激动,因为四天的努力在这个时候终于得到了收获。在不在实验室的时候,我们在寝室又会对自己为什么还没成功实现改题目进行一定的探讨,然后重新整理自己的思路,从中去寻找解决问题的方法。或者说,重新观察我们所画的图,有没有什么问题,或者是是否要进行更改,甚至是推翻重做,然后又花了一晚上的时间去绘制图形,进行波形图的模拟。在课下的时候也是过于劳累。这次实训的成功,首先第一点,离不开我的组员与我团结奋战在第一条线,每一天都准时与我一起进行我们项目的测试。第二点,也离不开老师的悉心指导,在我们对项目中的难点有疑问或者是很好的思路进行解决时,老师会在一定的程度上为我们进行一定的提示,让我们顺利的去解决我们在其中遇到的问题。第三点,也离不开我对这次实训的认真执行,以及劳苦劳累的对项目进行测试。若没有这三点,我们不可能实现这一次实训项目,我对我的组员,以及指导老师,还有自己进行由衷的感谢。总结而言,这个硬件大型作业实训项目是我宝贵的经历,让我在硬件设计和开发方面取得了长足的进步。通过实践,我深刻认识到了硬件设计的复杂性和挑战性,也意识到了团队合作和解决问题的重要性。这个项目让我不仅学到了专业知识,还培养了我的沟通能力、问题解决能力和团队合作精神。我相信,这些经验和教训将对我未来的职业发展产生积极的影响。我将继续学习和提升自己。以上便是我对这次实训的所有感想。

## 6.2 李睿

本次设计实训的过程中,我们最开始设想的红绿灯版本就是利用基本芯片和简单逻辑门电路去实现,并且我也非常高兴我们将这一想法贯彻到底并且实现了,让我们没有半途而废前功尽弃。我们的流程主要是分设计模块来进行的,从需要实现可以人工控制秒数,想到需要设计存储模;从需要每次每边亮不同组的灯,想到需要有一个状态编码来翻译为灯的亮暗控制,在后续中也设计为了一个主控芯片的模块;从直观上,既然要倒计时,也能直接看出我们需要设计减法计数的模块。这些部分实际上都比较简单,锁存器存储数据,利用状态控制信号和状态编码之间的真值表对应的逻辑表达式,就可以直接利用简单逻辑门来实现主控芯片,一个级联的十进制加减同步计数器就可以实现两位二进制数的减法,只要给一个稳定的电子脉冲就可以正常工作。但是真正的难度在于,这些部件(或者说小模块)到底要几个,怎么实现同一模块中不同芯片的选择,以及他们之间依靠什么信号来互相传递要求对方需要工作,或者让某模块取数据,又或者不许取数据。模块间的通信与时序有关,也许某一个信号的变化过快,来一个高电平就又立刻变回低电平,但是另一模块的 clk 信号端又很敏感,导致虽然信号给到了,但是另一模块并没有工作这样的情况。又比如说某一芯片是否工作的信号需要另外两个信号的逻辑合成来共同控制,如果两个信号不能同时变化,就完全可能出现一个模块要等另一个模块工作完才能执行工作,但是该模块此时才工作已经完全影响了另一个模块的工作这样的时序问题,非常麻烦。所以我们在设计的过程中反复 debug 其中的原因之一就是在谨慎挑选、尝试、调试用合适的信号来控制某个芯片的工作,比如级联的 74190 计数器,怎么给另一块 74190 传一个借位信号作为该芯片的 clk,我们有用过 MXMN 端取反,也有用过 RCON 信号直接控制,虽然芯片介绍的时候说的是两个信号就是单纯的去翻关系,但是最后在试验箱上操作的时候发现两个信号给的居然不是同步的,用波形模拟图也产生如此问题,还只有用 RCON 信号才能正常让另一块 74190 做减一操作。再比如锁存器部分我们先后使用过 74273 和 74373,区别在于 74273 需要一个 clk 信号去进行读入,而 74373 不需要。就是这一个小细节导致了开机是锁存器读入后读出与计数器的读入出现了通信的时间误差,有时候根本读不进去,有时候又读进去了,这就说明这种设计是不牢靠不鲁棒的,同时由于我们只用一组计数器,所以当然涉及到具体把哪一块锁存器的数据放到减法计数器里来的问题,这就需要选择器来解决。我们一开始使用的 74138 芯片居然不能够实现正确的芯片选择,导致选择的锁存器数据直接错乱,但是改用自己使用逻辑门设计的选择器,问题就直接解决了。这些问题其实都是在不断的 debug 的过程中,甚至走了相当多的弯路,才发掘出来的,所以完全可以说我最大的感受就是逻辑理论与实践操作有着挺大的预期误差,遇到上述的问题,我们一开始的纠错思

路完全不一定是对的，因此也只能一个一个去尝试。举例上述说的选择器导致的问题，它实际上也导致了一块自己设计的加法器去实现读取绿灯在减法器中的数据与黄灯数据做加和，一旦绿灯数据变 0 则继续从减法器中选择黄灯数据与 0 做加和，这样就实现了临边红灯的同步显示功能，永远等于黄灯和绿灯时间综合并且能够依赖我们已有的一块计数器去做显示，没必要再弄一个计时模块，这个办法本来很巧妙很好用。就是由于选择器出问题，我们错误的一度怀疑存储器、加法器可能有问题，因此把加法器拆掉再多弄一个计数模块，多加入两个锁存器的操作我们都有尝试，并且最终结果也很明显，失败了。一把选择器搞定也就都搞定了。也有出现过好不容易减法计数器能正常倒计时，但是脉冲给不稳（实际上并不完全是），导致一会儿能正常计时，一会儿又开始抽搐的情况发生，最后我们高度怀疑就是某些芯片内部使用了具有延时的，需要来考虑时序的，比如触发器之类的元件，直接导致计数器的读使能和 `clk` 在不该工作的时候突然工作，根本不同步，导致这也成了一个不鲁棒的设计。好就好在最后都依靠执行力把问题定位好了，各种尝试最终起了作用，成功实现了稳定可靠的红绿灯系统。最后，非常感谢每个组员的协助、出谋划策，每次上机大家都非常认真的去看图、绑定引脚、连线、提出建议，尤其感谢张焜璞同学的主控芯片设计、选择器设计、加法器设计，都起到了解决 `bug` 的重大现实意义，直接搞定了我们 `debug` 一直走歪路的问题。总之结果好的，一切都是好的。

### 6.3 文信伶

本次硬件大作业是基于数字逻辑理论知识的实践，由于距数字逻辑课程的学习已经过了大概一年，我对数字逻辑相关知识已经记忆模糊，因此在项目设计初期我的进展非常缓慢，面对要求也毫无头绪。好在队友们都十分耐心，在他们的帮助与讲解下，在自己一步步查阅相关资料、分析学习已有模型的基础上，我对项目的设计有了逐渐清晰的认识，也逐渐建立起了信心，在大家的共同努力下顺利完成了这个项目。

虽然我们组在项目进行中遇到了不少困难，电路图的设计一改再改，我们有过困惑不解之时，有过“争执”（对设计的激烈讨论），但大家也都没有放弃，而是互相鼓励，分析查找错误，错误的想象也为设计的成功不断地提供了新的思路，不断发散着我们的思维，在失败了数次之后终于实现了所有功能。因此，通过本次项目的合作，我也感受到了团队合作的重要意义，团队合作不仅能够将每个人的长处凝聚在一起更好的完成工作，也能弥补每个人的短板，在团队合作中不断学习，共同进步。同时也让我深深认识到，在学习过程中出现争执是必须的并且多多益善，争执有益于我们去互相学习彼此的想法，发现自身的问题，有助于我们快速提高自我。

通过对十字路口交通信号灯的设计，我对数字逻辑的理论知识得到了进一步的理解和深化，并更加提升了我的实际动手操作能力，进一步丰富和完善了对数字逻辑的认识，同时，在实践中积累的经验和教训将使我在今后的学习和工作上得到有力的帮助。

在这次对交通信号灯的设计与制作过程中，我了解了一些设计电路的步骤，同时也了解了关于交通信号灯的设计原理和设计思想，进一步熟悉了计数器 74190 芯片、JK 触发器等芯片的结构及其工作原理和具体使用方法，熟悉了 Quartus 软件的功能及其仿真环境。

通过本次课程设计，我也意识到课本上所学的知识只有通过实际应用才能够真正的融会贯通，成为活的知识。例如，课本上讲 JK 触发器为下降沿触发的边沿触发器，但当我们按照下降沿触发的方式去设计时却发现电路无法正常工作，最后将触发器单独拿出来测试才发现软件中的 JK 触发器为上升沿触发；而在触发器触发信号的选择上，从理论上讲计数器 8 为输出相或取反是可行的，然而在实际应用中由于竞争一冒险现象，这种设计方案是不可行的。因此，实践是检验真理的唯一标准，理论上可行的方案在实际中不一定也可行。作为一名理工科学生，我们应该要更加避免“读死书”，在平时的学习生活中要将所学知识多加加以实际应用，才能更好地掌握和理解这些知识。

通过本次项目设计，我也认识到了自身的短板以及知识的欠缺，对知识的掌握不够扎实，将知识应用于实践的能力还不足，在接下来的学习生活中，我将不断努力提高自我。

## 6.4 贺新宇

夏季实训的这段时间，我有幸参与了一个非常有趣和富有挑战性的项目，即设计一个交通灯的控制电路。这个实训经历不仅让我学到了许多专业知识和技能，还深刻地影响了我的思考方式和解决问题的能力。通过与团队成员的合作以及克服各种困难和挑战，我获得了很多宝贵的经验和成长。

首先，这个实训项目让我意识到了团队合作的重要性。在整个过程中，我与我的团队成员紧密合作，分享想法、讨论解决方案，并共同努力解决出现的问题。通过相互合作，我们能够充分利用每个人的专长和才能，将我们的想法付诸实践。团队合作不仅仅是完成任务，更是培养了我们的沟通能力、协作能力和解决问题的能力。在未来的工作和生活中，这些技能将非常重要。

其次，这个实训项目锻炼了我的技术能力和创新思维。设计一个交通灯的控制电路需要对电子元件和电路原理有深入的理解。通过实际操作和调试，我学到了很多关于电路设计和电子元件的知识。我学会了如何选择合适的元件，设计有效的电路，并进行测试和修复。在这个过程中，我也遇到了许多技术上的问题和挑战，但通过不断的尝试和探索，我学会了解决问题的方法和技巧。此外，我也发现创新思维在电路设计中的重要性。通过思考和尝试新的想法和方法，我们能够设计出更加高效和可靠的控制电路。

另外，这个实训项目培养了我的解决问题的能力 and 坚持不懈的精神。在实践过程中，我们面临了许多技术和时间上的限制。有时候，我们的设计不够理想或者遇到了困难，但是我们从未放弃。我们持续地努力改进和优化我们的设计，直到最终达到我们的目标。这个过程让我体会到了坚持不懈的重要性。在以后的工作和生活中，面对各种困难和挑战，我将时刻牢记这个经历，坚持不懈地追求目标，并寻找解决问题的方法。

最后，这个实训项目也让我深刻认识到了自己的潜力和成长空间。在开始实训时，我对电路设计和控制电路的知识相对陌生，对自己的能力也有所怀疑。然而，通过实践和学习，我逐渐掌握了必要的知识和技能，并成功地完成了这个项目。这次经历让我明白，只要勇敢面对挑战并全力以赴，我们都能够超越自己，实现更大的成就。

综上所述，夏季实训设计交通灯控制电路的项目对我来说是一次宝贵的经历和成长机会。通过与团队合作、锻炼技术能力和创新思维、培养解决问题的能力以及发现自身潜力，我受益匪浅。这次实训不仅仅是对知识的应用，更是对自己思维和能力的挑战。我相信，在未来的学习和职业生涯中，我会将这些宝贵的经验和教训运用到实际中，不断追求进步和成长。

## 6.5 许文博

在这个夏天，我参加了为期四周的夏季硬件大作业实习项目。这个实习经历不仅让我更深入地了解了硬件开发的流程和技术，也让我体会到了团队合作的重要性和自身的成长。在这段时间里，我获得了许多宝贵的经验和教训。

首先，我要感谢硬件大作业项目给我提供的机会。作为一名大学生，能够亲身参与实际的硬件开发项目，对于我来说是一种巨大的荣幸和机遇。这个项目涉及到了从硬件原型设计到测试和优化的全过程，我在其中扮演了一个团队成员的角色，与其他学生一起合作完成了任务。

在实习期间，我学到了很多关于硬件设计和开发的知识。我熟悉了各种硬件元件的功能和特性，并学会了如何使用 **Quartus II** 软件进行电路设计。我还学会了进行电路板布线，以及如何进行电路分析和测试。通过实践，我深刻理解了理论知识与实际应用之间的联系，也更加熟悉了硬件开发过程中的常见挑战和解决方法。

与此同时，实习也让我认识到了团队合作的重要性。在项目中，我与其他小组成员密切合作，共同解决问题和完成任务。我们进行了频繁的交流和讨论，分享了自己的经验和想法。通过团队合作，我们能够共同攻克难题，提高工作效率，并且相互学习和促进。这个经验让我明白，无论是在硬件开发还是其他领域，团队合作是取得成功的关键。

在实习过程中，我也遇到了一些挑战和困难。有时候，我会遇到复杂的技术问题或者工作安排的压力，但是我从中学会了如何保持冷静和解决问题的能力。我学会了寻求帮助和支持，并且善于与他人沟通和协调。这些经历让我变得更加坚韧和自信，也培养了我解决问题的能力。

总的来说，夏季硬件实习作业给我带来了宝贵的学习和成长机会。通过实践，我不仅学到了硬件开发的技术知识，还提高了团队合作和解决问题的能力。这个实习经历让我更加确定了自己对硬件领域的兴趣，并且激发了我继续深入学习和探索的热情。我相信，这段宝贵的经历将对我未来的职业发展产生积极的影响。我感谢这次实习机会，也感谢团队中每一位给予我指导和支持的人。我会继续努力学习和提升自己，为将来的工作做好充分的准备。

## 7 参考文献

- [1] 靖川川. 电子技术课程设计—交通灯控制系统设计. [EB/OL]. CSDN, 2022. [2023-07-06]. [https://blog.csdn.net/qq\\_60078560/article/details/125580914](https://blog.csdn.net/qq_60078560/article/details/125580914)
- [2] AI 小小白. FPGA 开发基础——基于 multisim 以及 Quartus 实现的交通灯电路. [EB/OL]. CSDN, 2020. [2023-07-06]. <http://blog.csdn.net/PRINCE2327/article/details/105379855>