***2024***



**数字电路与逻辑设计**

**实验报告**

|  |  |
| --- | --- |
| 专 业： | 计算机科学与技术 |
| 班 级： |  |
| 学 号： |  |
| 姓 名： | losyi |
| 电 话： |  |
| 邮 件： |  |
| 完成日期： |  |

**实验报告及电路设计评分细则**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 评 分 项 目 | 满分 | 得分 | 备注 | |  |
| 文档格式（段落、行间距、缩进、图表、编号等） | 15 |  |  | | 实验报告总分 |
| 实验总体设计 | 10 |  |  | |
| 实验过程 | 50 |  |  | |
| 遇到的问题及处理 | 10 |  |  | |
| 设计方案存在的不足 | 5 |  |  | |
| 心得（含思政） | 5 |  |  | |
| 意见和建议 | 5 |  |  | |
| 电路（头歌） | 100 |  |  | |  |
| 教师签名 |  | | 日 期 |  | |

备注：实验过程将从电路的复杂度、是否考虑竞争和险象、电路的美观等方面进行评分。

实验课程总分=电路（头歌）\*0.4+实验报告\*0.6目 录

[1 实验概述 1](#_Toc184022990)

[1.1 实验名称 1](#_Toc184022991)

[1.2 实验目的 1](#_Toc184022992)

[1.3 实验环境 1](#_Toc184022993)

[1.4 实验内容 1](#_Toc184022994)

[1.5 实验要求 3](#_Toc184022995)

[2 实验总体设计 4](#_Toc184022996)

[2.1 实验总体设计思路 4](#_Toc184022997)

[2.2 实验总体设计框架 5](#_Toc184022998)

[3 实验过程 7](#_Toc184022999)

[3.1 7段显示驱动电路设计 7](#_Toc184023000)

[3.2 无符号比较器（4位、8位） 9](#_Toc184023001)

[3.3 2选1选择器设计（1位、8位） 13](#_Toc184023002)

[3.4 模十可逆计数器（包含状态机、输出函数及整体电路） 15](#_Toc184023003)

[3.5 两位十进制可逆计数器 20](#_Toc184023004)

[3.6 交通灯状态机 23](#_Toc184023005)

[3.7 交通灯输出函数设计 28](#_Toc184023006)

[3.8 交通灯控制系统 30](#_Toc184023007)

[4 设计总结与心得 36](#_Toc184023008)

[4.1 实验总结 36](#_Toc184023009)

[4.1.1遇到的问题及处理 37](#_Toc184023010)

[4.1.2设计方案存在的不足 37](#_Toc184023011)

[4.2 实验心得 39](#_Toc184023012)

[4.3 意见与建议 40](#_Toc184023013)

# **实验概述**

## 实验名称

交通灯控制系统设计。

## 实验目的

本实训将提供一个完整的数字逻辑实验包，从真值表方式构建7段数码管驱动电路，到逻辑表达式方式构建比较器，多路选择器，利用同步时序逻辑构建BCD计数器，最终集成实现为交通灯系统，实验由简到难，层次递进，从器件到部件，从部件到系统，帮助同学们了解数字逻辑设计的全过程。

## 实验环境

软件：logisim软件一套。

平台：https://www.educoder.net/shixuns/shplc3jv/challenges

## 实验内容

某个主干道与次干道公路十字交叉路口，为确保人员及车辆安全、迅速地通过，在交叉路口分别设置了两组红、绿、黄三色信号灯。红灯禁止通行；绿灯允许通行；黄灯亮提醒行驶中的车辆减速通行。交通灯控制系统示意图如图1-1所示。

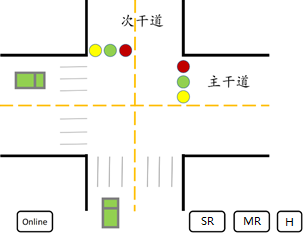


图1-1 交通灯控制系统示意图

设计一个交通灯控制系统，具体功能要求如下：

（1）电路有4个输入，分别为高峰期信号H、主干道通行请求MR、次干道通行请求SR和紧急状态控制信号（Online），其中，主干道通行请求（MR）包括主干道方向有车辆信号和次干道有行人通过信号，次干道通行请求（SR）包括次干道方向有车辆信号和主干道有行人通过信号。 电路输出为红灯、绿灯和黄灯的剩余时间以及主干道和次干道的红灯、绿灯和黄灯的状态。可用2个七段数码管和6个Led灯显示。

（2）任何时刻，主干道绿灯、黄灯和红灯有且仅有一个灯亮，次干道绿灯、黄灯和红灯有且仅有一个灯亮；

（3）主干道绿灯指主干道绿灯亮，主干道黄灯和红灯熄灭，次干道红灯亮，次干道黄灯和绿灯熄灭；主干道黄灯指主干道黄灯闪烁，主干道绿灯和红灯熄灭，次干道红灯亮，次干道黄灯和绿灯熄灭；次干道绿灯指次干道绿灯亮，次干道黄灯和红灯熄灭，主干道红灯亮，主干道黄灯和绿灯熄灭；次干道黄灯指次干道黄灯闪烁，次干道绿灯和红灯熄灭，主干道红灯亮，主干道黄灯和绿灯熄灭；

（4）主干道通行指主干道绿灯或主干道黄灯。高峰期，主干道绿灯倒计时27s（30-04），黄灯倒计时3s（03-01）；非高峰期，主干道绿灯倒计时12s（15-04），黄灯倒计时3s（03-01）；

（5）次干道通行指次干道绿灯或次干道黄灯。次干道绿灯倒计时12s（15-04），黄灯倒计时3s（03-01）；

（6）初始状态，为主次干道均黄灯闪烁，显示0；

（7）紧急状态时，主干道绿灯常亮，显示99；

（8）非紧急状态时（Online=0），若主干道有通行请求，次干道无通行请求，初始状态下直接进入主干道通行，非初始状态下，当前通行干道黄灯倒计时结束后，为主干道通行；

（9）非紧急状态时（Online=0），若主干道无通行请求，次干道有通行请求，初始状态下直接进入次干道通行，非初始状态下，当前通行干道黄灯倒计时结束后，为次干道通行；

（10）非紧急状态时（Online=0），主次干道都有通行请求时，初始状态下直接进入主干道通行，非初始状态时，当前通行干道黄灯倒计时结束后，两干道交替通行，即主干道通行变为次干道通行，次干道通行变为主干道通行；

（11）非紧急状态时（Online=0），若主干道、次干道均无通行请求，则当前通行干道黄灯倒计时结束后，进入初始状态；

（12）当Online=1时，若次干道为通行状态，需次干道黄灯倒计时结束才能进入紧急状态；当Online=1时，若主干道为通行状态，直接进入紧急状态；

（13）紧急状态结束，高峰期时，进入高峰期主干道绿灯状态；紧急状态结束，非高峰期时，进入非高峰期主干道绿灯状态。

## 实验要求

（1）根据给定的实验包，将交通灯控制系统切分为一个个实验单元；

（2）对每一个实验单元，按要求设计电路并使用Logisim软件进行虚拟仿真；

（3）设计好的电路在educoder平台上提交并进行评测，直到通过全部关卡。

# **实验总体设计**

## 实验总体设计思路

### 明确需求，确定状态

实验要求设计一个交通信号灯控制系统，除了红绿灯的转换外，还需要考虑初始状态，紧急状态下的状态切换和优先级确认。根据需求描述，我们首先需要确定电路中存在的状态以及状态间的转换关系（做出状态图）。具体而言，交通信号灯系统需要包括以下几种状态：

1. 初始状态：主次干道均黄灯闪烁，显示0；
2. 非高峰期主干道绿灯：倒计时12s
3. 高峰期主干道绿灯：倒计时27s
4. 主干道黄灯：倒计时3s
5. 次干道绿灯：倒计时12s
6. 次干道黄灯：倒计时3s
7. 紧急状况：主干道绿灯，次干道红灯，显示99s

### 根据需求划分功能模块

1. **显示模块：**实验要求实时显示交通信号灯的剩余时间，由于最大显示99s，我们可以用两个七段显示驱动电路完成需求，将计数器的输出转换为七段数码管的驱动信号，实现倒计时数字的可视化。

* 输入：计数器输出的Timer信号
* 输出：译码器的控制信号(S1T、S1B、S2T、S2B)

1. **倒计时选择模块：**由于存在多种状态主干道，次干道倒计时，初始状态和紧急状态等多种状态，七段数码管应根据电路输入确定接收不同的Timer信号（比如初始状态下为0，紧急状态下为99），我们可以选用8位2路选择器，选择合适的计数器或逻辑输入用于控制信号。

* 输入：通行状态信号(Pass)、主干道和次干道的倒计时信号(Timer1、Timer2)
* 输出：即将用于显示的Timer信号

1. **主干道通行倒计时模块：**主要负责提供主干道红灯和黄灯的倒计时信号，主要功能由可异步置为模十可逆计数器实现，根据目前的状态决定是否进行计数和置位操作。

配合多路选择器，可根据高峰期信号选择不同的计时长度，同时还需要比较器，来确认主干道绿灯/黄灯是否计时完成。最后，配置计数器的输出为两个七段数码管驱动模块提供倒计时信号Timer1。

* 输入：高峰期信号(H)、时钟信号(Clk)、通行状态信号(Pass)、倒计时结束信号(T4、T5)
* 输出：倒计时信号(Timer1)和倒计时结束信号(T3、T4、T6)。

1. **次干道通行倒计时模块：**功能和实现过程类似于主干道通行倒计时模块，由于其不受高峰期和紧急状态等影响，实现逻辑要更加简单。主要使用模十可逆计数器，根据次干道的通行请求信号启动倒计时。配置计数器输出Timer2到七段数码管驱动模块，用于实时显示倒计时

* 输入：次干道通行请求信号(SR)、时钟信号(Clk)。
* 输出：倒计时信号(Timer2)和倒计时结束信号(T3、T4、T6)。

1. **状态机模块（激励和输出函数）：**状态转换模块是交通灯控制系统的核心和设计过程中的重难点，用于管理系统状态的切换，并根据当前状态生成输出信号。其使用同步时序逻辑(如D触发器)实现状态存储。根据状态转移表设计状态激励逻辑。使用逻辑门和多路选择器生成红、黄、绿灯的输出信号。

* 输入：当前状态、输入信号（H、MR、SR、Online、T1、T2、T3、T4）
* 输出：状态机下一状态(Status)、通行信号(Pass)、红绿灯颜色控制信号(RGB)

## 实验总体设计框架

根据2.1中描述的设计思路，将所有功能需求划分为多个模块，通过模块化设计的方式逐步实现并最终集成，同时，还需要实现各个模块中需要用到的多路选择器，比较器等复用器。最终框架示意图如图2-1



图2-1 实验总体设计框架示意图

通过上述框架设计，本实验确保了系统的模块化结构、清晰的数据流和控制流，从而实现高效、可靠的交通灯控制系统。

为实现整个系统，我们需要依次完成以下子电路，将在下一章中具体介绍

1. 七段显示驱动电路
2. 4位无符号数比较器
3. 8位无符号数比较器
4. 1位2路选择器
5. 8位2路选择器
6. 模十可逆计数器激励函数
7. 模十可逆计数器输出函数
8. 模十可逆计数器（可异步置位）
9. 2位十进制可逆计数器设计

(10)交通灯控制器激励函数

(11)交通灯控制器输出函数

# **实验过程**

## 7段显示驱动电路设计

1. 设计思路及设计过程

七段显示译码器通过接受BCD码，把BCD码转换成 7 段字型数码管所要求的代码在数码管上显示十进制数，电路的输入输出如下：

* 输入：4位BCD码，用x3~x0表示
* 输出：7位数码管驱动信号Seg1~Seg7

为了设计七段显示驱动电路，首先需要根据输入的BCD码来生成对应的七段显示信号，每个数字对应七段显示器的特定亮灯组合。由于电路状态较少，我们可以直接利用真值表完成电路设计。

表3-1 7段显示驱动电路设计的真值表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x3 | x2 | x1 | x0 | Seg1 | Seg2 | Seg3 | Seg4 | Seg5 | Seg6 | Seg7 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

1. 电路图

通过logisim的自动生成功能，可得电路图如下：

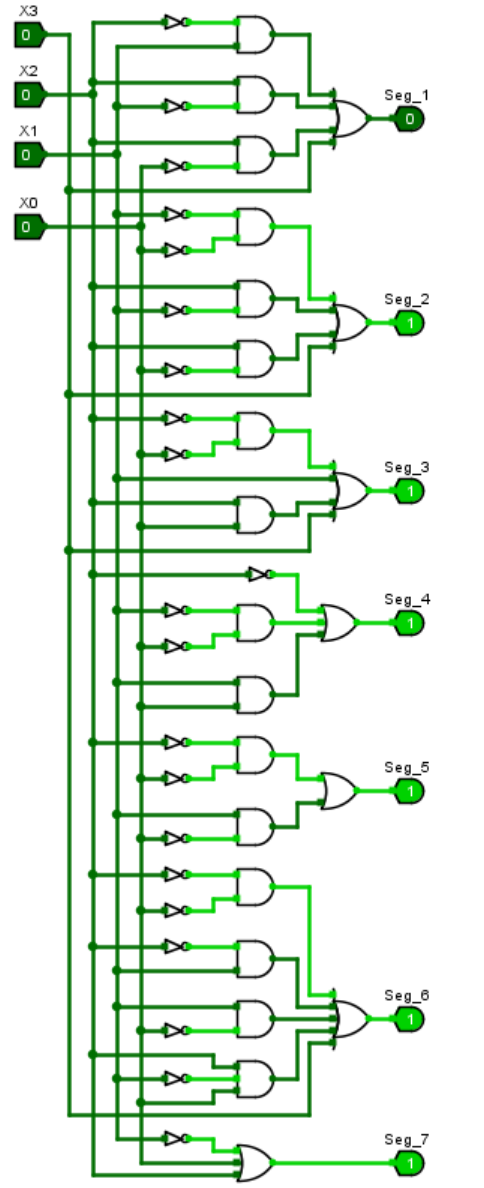


图3-1 七段显示译码器电路图

1. 测试图

利用驱动测试电路结合Educoder平台的自动测试功能，完成对电路的测试工作

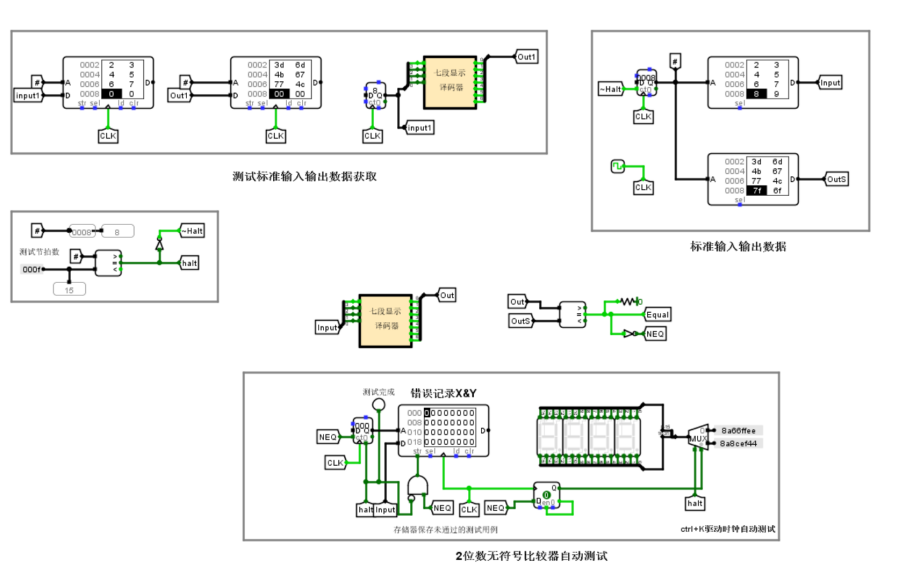


图3-2 七段显示译码器驱动测试电路图

1. 测试分析

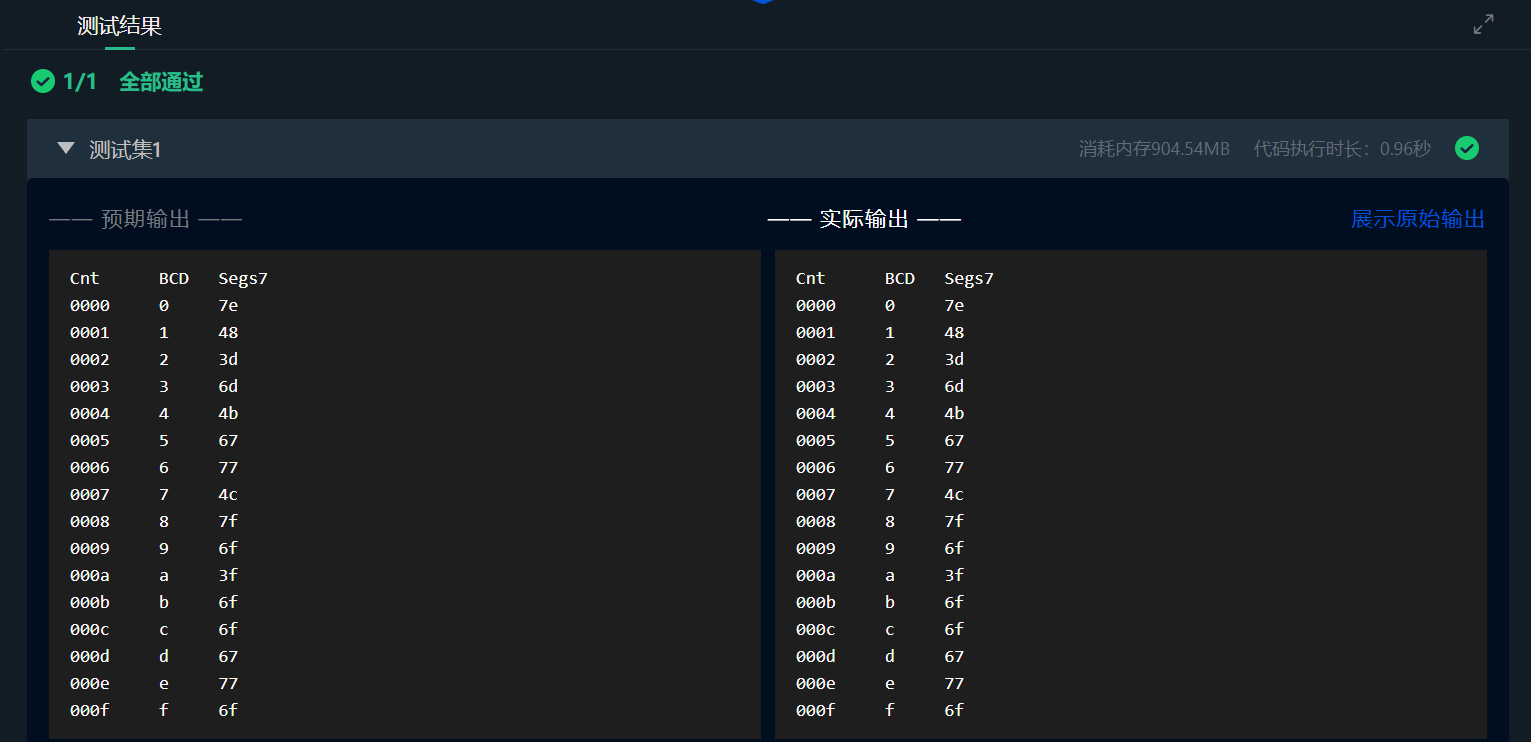


图3-3 七段显示译码器测试结果图

测试结果表明，电路对于所有输入都能输出理论的匹配结果，说明电路设计是正确的。

## 无符号比较器（4位、8位）

1. 设计思路及设计过程

无符号比较器主要用于判断当前倒计时与3s和0s的大小关系，由于实验中倒计时为两位十进制数，每一位需要用4位2进制BCD码表示，因此我们需要8位无符号比较器，直接设计8位无符号比较器状态数多，设计较为复杂，我们可以利用电路级联扩展的基本原理，将两个4位比较器级联。将输入X和Y分别分为高4位和低4位，首先比较高4位，若相等，再比较低4位。利用4位无符号比较器作为子模块，可以减少设计复杂度，提高模块复用性。对此，我们首先需要考虑4位无符号比较器的设计，分析可知电路的输入输出如下

* 输入：4位输入X和Y(X3~X0和Y3~Y0)
* 输出：XY的大小关系G、E、L

该电路有8个输入，真值表表项256项，使用真值表过于复杂，可以使用逻辑表达式来简化设计过程。关键的思路是逐位比较输入的两个4位数（X和Y），并利用比较规则通过布尔表达式描述它们之间的大小关系。不难想到，我们可以通过从最高位开始逐位比较来确定XY的大小关系，具体逻辑表达式的完成过程如下

1. X大于Y(G): X > Y时，首先需要从高位开始逐位比较：

如果X的最高位大于Y的最高位（X3 > Y3），则X > Y；

如果X3 == Y3，继续比较X2与Y2，依此类推。

最终表达式为：G=X0 ~Y3 ~Y2 ~Y1 ~Y0 + X1 ~Y3 ~Y2 ~Y1 + X1 X0 ~Y3 ~Y2 ~Y0 + X2 ~Y3 ~Y2 + X2 X0 ~Y3 ~Y1 ~Y0 + X2 X1 ~Y3 ~Y1 + X2 X1 X0 ~Y3 ~Y0 + X3 ~Y3 + X3 X0 ~Y2 ~Y1 ~Y0 + X3 X1 ~Y2 ~Y1 + X3 X1 X0 ~Y2 ~Y0 + X3 X2 ~Y2 + X3 X2 X0 ~Y1 ~Y0 + X3 X2 X1 ~Y1 + X3 X2 X1 X0 ~Y0

1. X等于Y(E): X = Y时，要求所有对应位都相等，可以直接写出表达式：E=(X3=Y3)&(X2=Y2)&(X1=Y1)&(X0=Y0)
2. X小于Y时(L)，X < Y时，逻辑与X > Y相对，可以直接为G取反(这点无法通过logisim自动生成电路，需要手动完成电路)得到结果，也可以改变1式中非符号的位置来实现，逻辑表达式为L=G&E

通过上述逻辑表达式设计，我们能够通过逐位比较推导出X和Y之间的大小关系，并通过逻辑表达式简化设计。对于4位无符号数比较器，逐位比较是核心方法，而逻辑表达式提供了更高效、易于实现的方式，而在实现了4位无符号数比较器后，以相同的思想设计8位无符号数比较器就更为简单了，将8位输入分为两部分(每部分4位，通过4位无符号比较器进行比较),比较过程如下：

1. X > Y（G）：如果高4位中的X大于Y或者如果高4位相等且低4位中X大于Y，则X > Y。
2. X = Y（E）：如果高4位和低4位都相等，则X = Y。
3. X < Y（L）：如果高4位中的X小于Y或者如果高4位相等且低4位中X小于Y，则X < Y。

通过上述思想，8位无符号数比较器的设计变得更加简洁，我们可以开始完成具体电路了。

1. 电路图

由于电路过长，重复性高，这里4位无符号比较器只展示部分电路

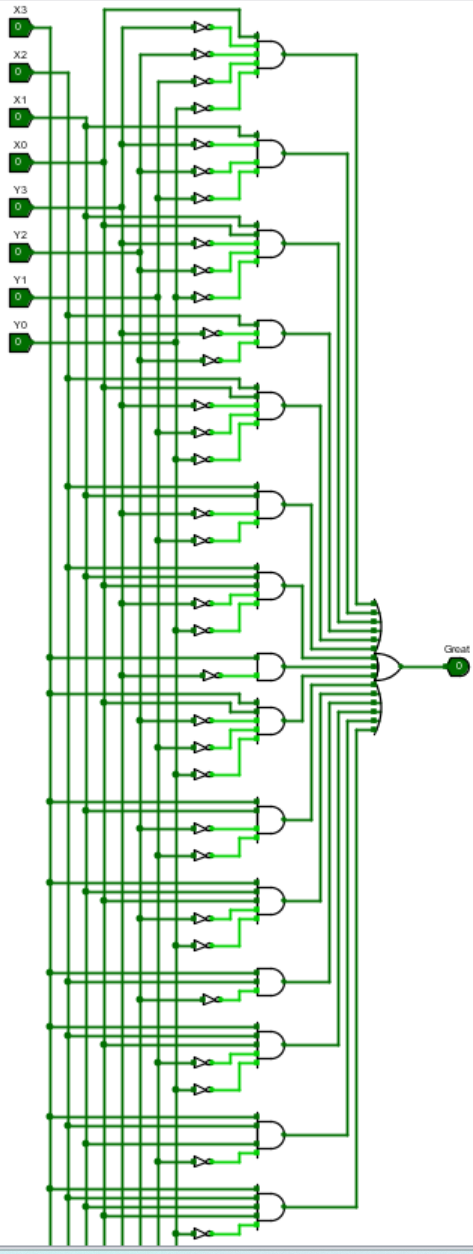


图3-4 4位无符号比较器电路图

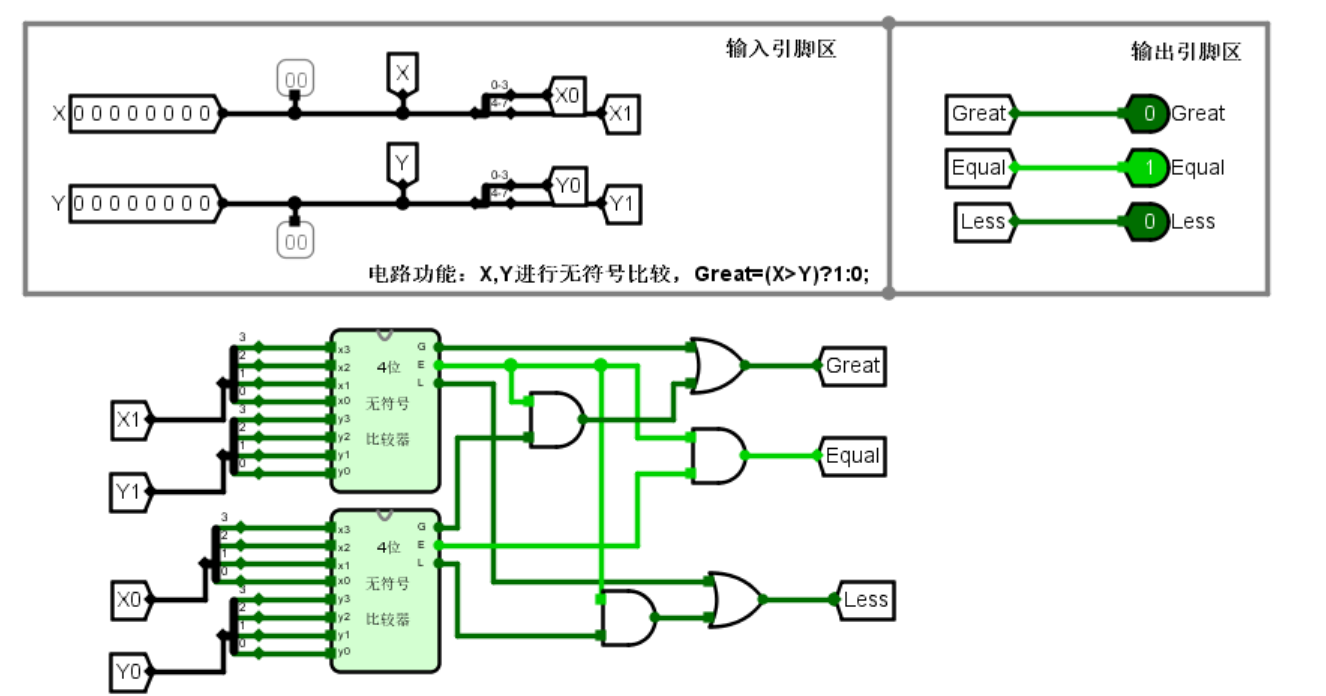


图3-5 8位无符号比较器电路图

1. 测试图

利用驱动测试电路结合Educoder平台的自动测试功能，完成对电路的测试工作

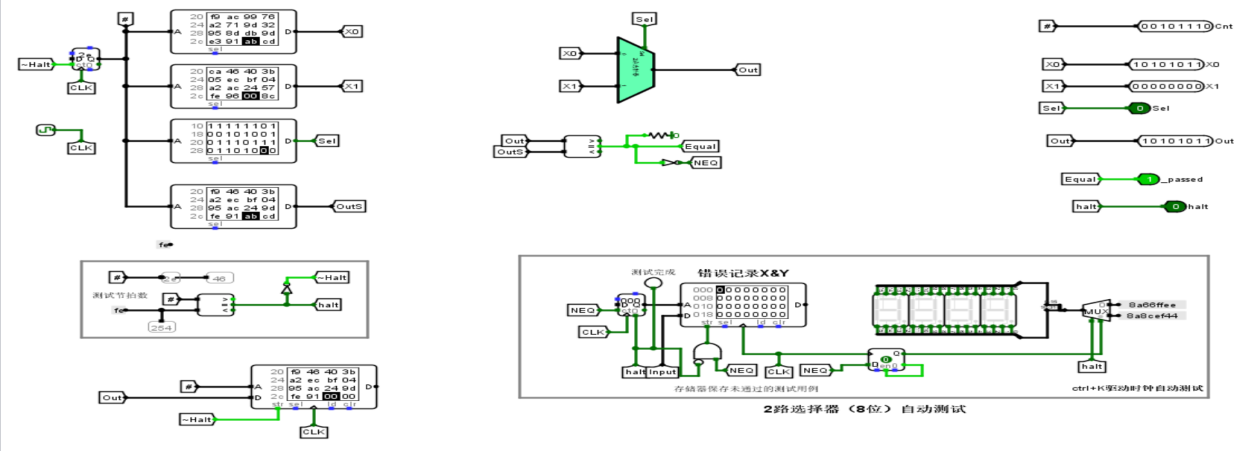


图3-6 8位无符号比较器驱动测试电路图

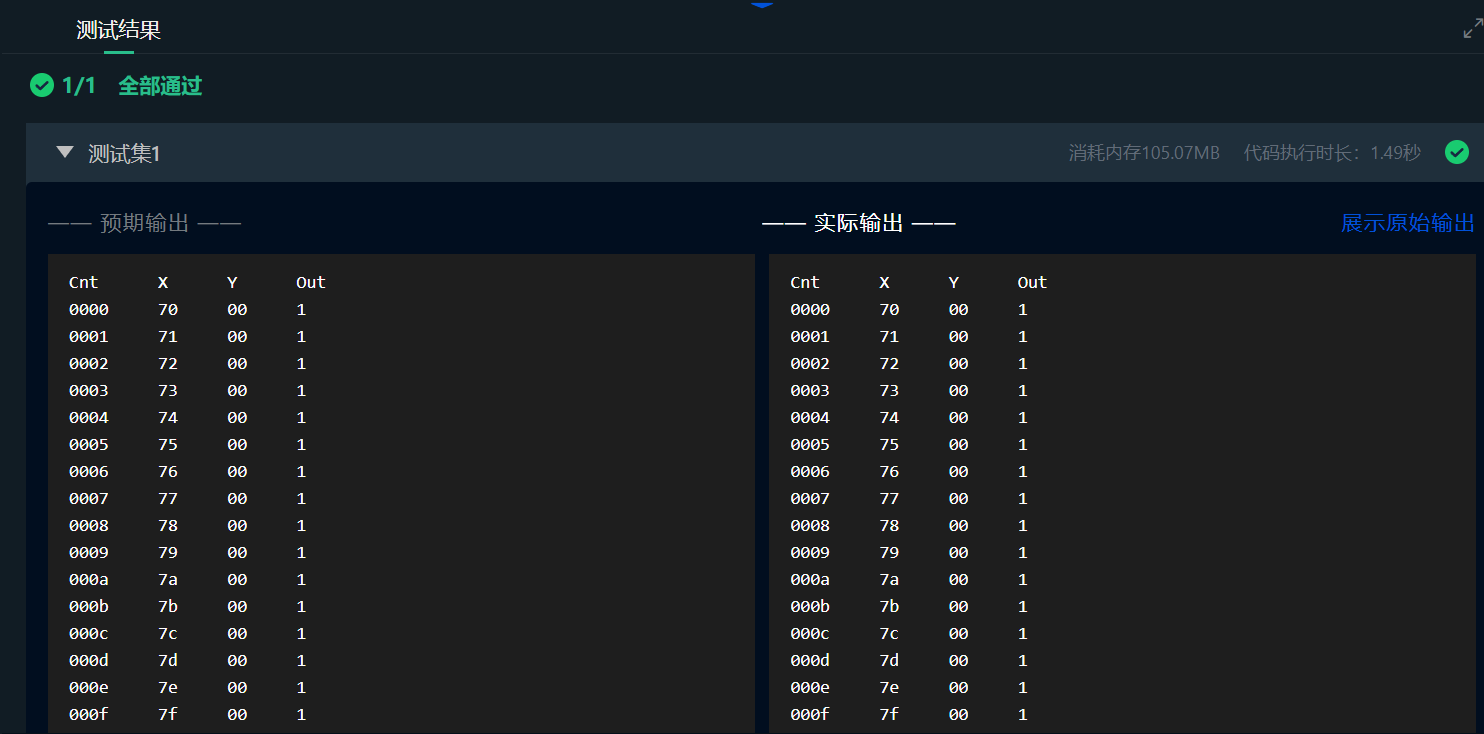


图3-7 8位无符号比较器测试结果图

1. 测试分析

测试结果表明，电路对于所有输入都能输出理论的匹配结果，说明电路设计是正确的。

## 2选1选择器设计（1位、8位）

1. 设计思路及设计过程

多路选择器(Multiplexer)又称数据选择器或者多路开关,是一种多路输入、单路输出的组合逻辑电路,本实验需要实现的2路选择器是最简单的结构，8位和1位的2路选择器在本质上没有区别，只需通过分线器保持输入位宽的一致即可。设计思路比较简单，通过一个与门，一个或门，一个非门即可实现。1位2路选择器需要根据选择信号Sel来选择两个输入之一。Sel信号控制一个与门，当Sel为0时，X通过该与门传递；当Sel为1时，Y通过另一个与门传递，最后通过或门将两个信号结合输出到Out。

* 输入：1/8位输入X和Y，选择控制端Sel
* 输出：1/8位输出Out(Out=（Sel==0）?X:Y)

1. 电路图

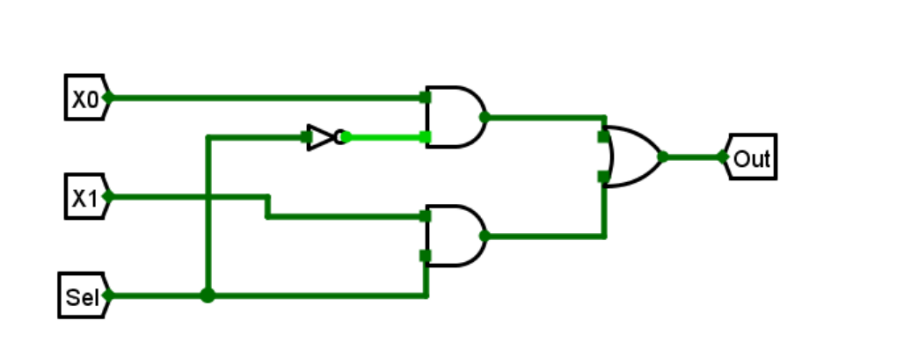


图3-8 1位2路选择器电路图

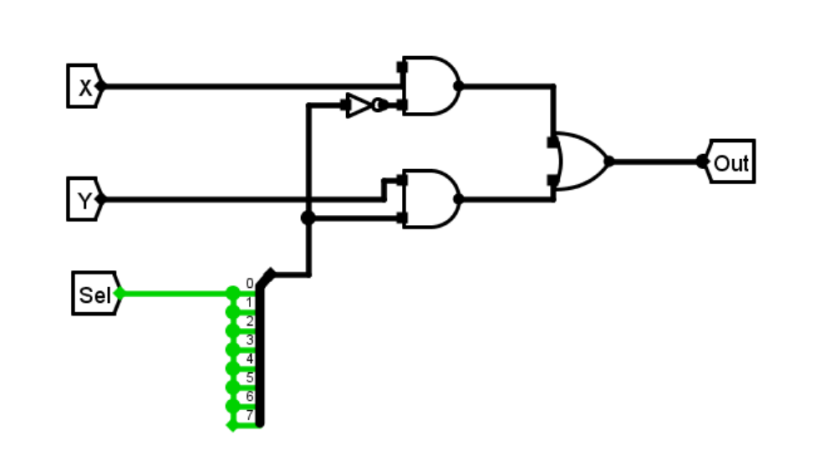


图3-9 8位2路选择器电路图

1. 测试图

利用驱动测试电路结合Educoder平台的自动测试功能，完成对电路的测试工作，由于逻辑基本一致，这里仅展示8位2路选择器的测试图。

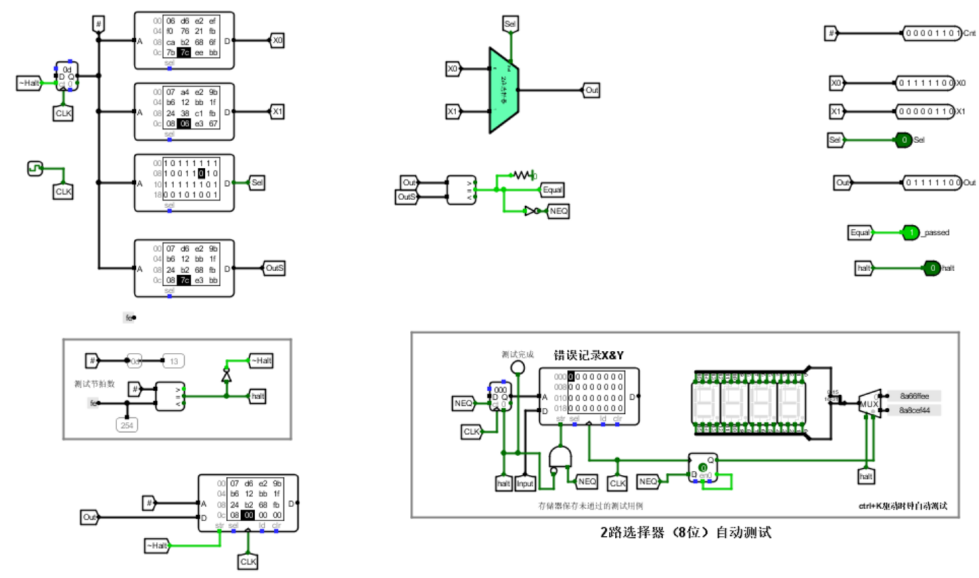


图3-10 8位2路选择器驱动测试电路图



图3-11 8位2路选择器测试结果图

1. 测试分析

测试了255种不同的输入组合，确保每一位的选择都能正确输出，从而验证整个8位选择器电路的功能。测试结果表明，电路对于所有输入都能输出理论的匹配结果，说明电路设计是正确的。

## 模十可逆计数器（包含状态机、输出函数及整体电路）

1. 设计思路及设计过程

本实验要求实现的模十可逆计数器是一个4位的二进制计数器，能够在正向和反向两种模式下工作。该计数器的状态范围为0到9，支持正向计数和反向计数，并且支持异步置位功能，模10可逆计数器的核心是通过时钟触发以及控制输入（设置计数方向和异步置位）来控制计数过程。具体设计流程如下：

(a)设计模十可逆计数器的状态图和状态表，得到D触发器的激励函数的真值表，利用“分析组合逻辑电路”，输入激励函数的表达式，直接生成电路。

确保在Mode控制位为0时，计数器从0到9正向计数；当Mode为1时，计数器从9到0反向计数。

* 输入：当前状态y(y3~y0),Mode(Mode=0正向计数，Mode=1，反向计数)
* 输出：D3~D0,对应的D触发器的激励

表3-2 模十可逆计数器状态表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 现态y | | | | 输入信号 | 次态 | | | |
| y3 | y2 | y1 | y0 | Mode | D3 | D2 | D1 | D0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |

通过状态表，易得真值表和激励函数的逻辑表达式：

D3=~y3 ~y2 ~y1 ~y0 Mode + ~y3 y2 y1 y0 ~Mode + y3 ~y2 ~y1 ~y0 ~Mode + y3 ~y2 ~y1 y0 Mode

D2=~y3 ~y2 y1 y0 ~Mode + ~y3 y2 ~y1 ~Mode + ~y3 y2 y0 Mode + ~y3 y2 y1 ~y0 + y3 ~y2 ~y1 ~y0 Mode

D1=~y3 ~y1 y0 ~Mode + ~y3 y1 ~y0 ~Mode + ~y3 y1 y0 Mode + ~y3 y2 ~y1 ~y0 Mode + y3 ~y2 ~y1 ~y0 Mode

D0=~y3 ~y0 + ~y2 ~y1 ~y0

最后通过logisim自动生成激励函数电路。

(b)设计模十可逆计数器的输出函数，生成计数器的进位/借位信号，该输出信号与状态和输入信号有关（Mealy型）。

* 输入：当前状态y(y3~y0),Mode(Mode=0正向计数，Mode=1，反向计数)
* 输出：Cout(进位或借位信号)

根据实验要求，计数器需要输出进位（或借位）信号Cout，当正向计数器从9跳回0时，或者反向计数器从0跳回9时，输出Cout为1。输出函数根据当前状态和Mode信号计算Cout，并根据状态改变给出正确的进位或借位信号。

不难发现，只有两种输入状态可以使得Cout输出为1,其它状态均输出0因此，我们可以通过构建真值表自动生成逻辑电路，且电路结构较简单。

表3-3 模10可逆计数器输出函数的真值表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| y3 | y2 | y1 | y0 | Mode | Cout |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| x | x | x | x | x | 0 |

并可得Cout的逻辑函数表达式为~S3 ~S2 ~S1 ~S0 Mode + S3 ~S2 ~S1 S0 ~Mode

(c)构建模十可逆计数器

实验要求， 利用已经设计完成的模十可逆计数器激励函数、模十可逆计数器输出函数，采用D触发器构建模十可逆计数器，该计数器支持异步预置功能，当预置控制位为1，直接将Din数据写入触发器中。

首先，我们需要明确电路的输入输出信号，如表3-4。

表3-4 模10可逆计数器的引脚表

|  |  |  |  |
| --- | --- | --- | --- |
| 信号 | 输入/输出 | 位宽 | 说明 |
| CLK | 输入 | 1 位 | 时钟输入 |
| En | 输入 | 1 位 | 使能信号，为1时据Mode位进行计数 |
| Mode | 输入 | 1位 | Mode=0正向计数Mode=反向计数 |
| PreSet | 输入 | 1位 | 预置控制端，为1时异步写入Din |
| Din | 输入 | 4位 | 计数器预置数据 |
| Q | 输出 | 4位 | 计数器计数输出 |
| Cout | 输出 | 1位 | 进位借位输出，正向数到9，反向计数到0时输出为1 |

我们使用典型维持阻塞的D触发器来作为实现计数器的基本存储单元。D触发器能够根据时钟信号和使能信号更新输出状态，并且具有直接置位端，因此非常适合用作计数器的基本单元。通过集成D触发器和激励函数、输出函数的组合逻辑，不难组合出出完整的模十可逆计数器。各个D触发器通过时钟信号同步工作，通过使能信号决定是否工作。由于电路涉及1位和4位位宽，需要在适当的地方加入分线器进行转换。

为实现置位功能，我们设置了PreSet和Din输入，当PreSet为1时，将Din的值传给D3-D0，D3-D0分别与对应D触发器的置位端相连，最终电路输出Cout和Q，Q同时作为激励函数电路的输入信号实现状态转换。

1. 电路图

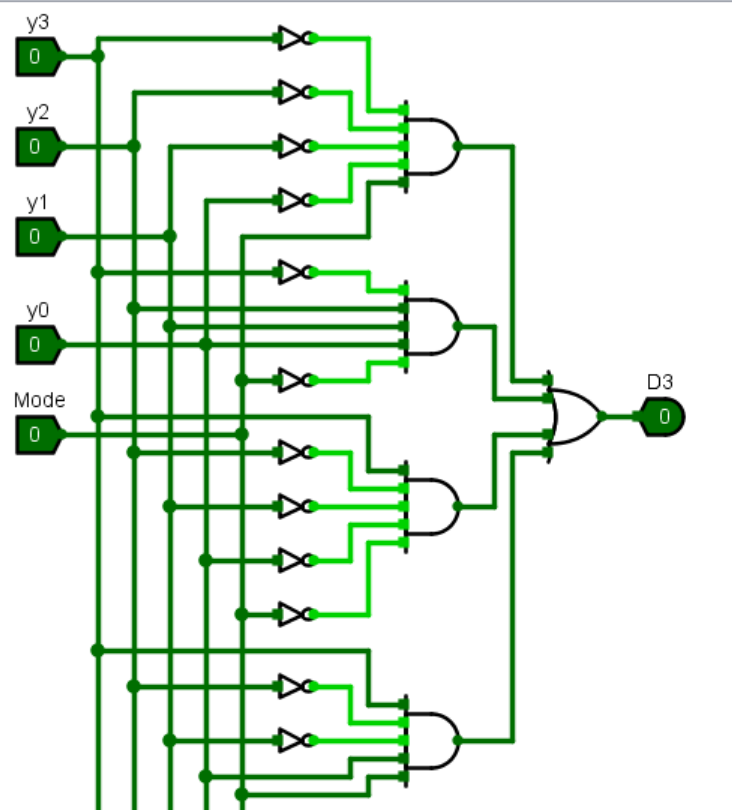


图3-12 模10可逆计数器激励函数电路(部分)

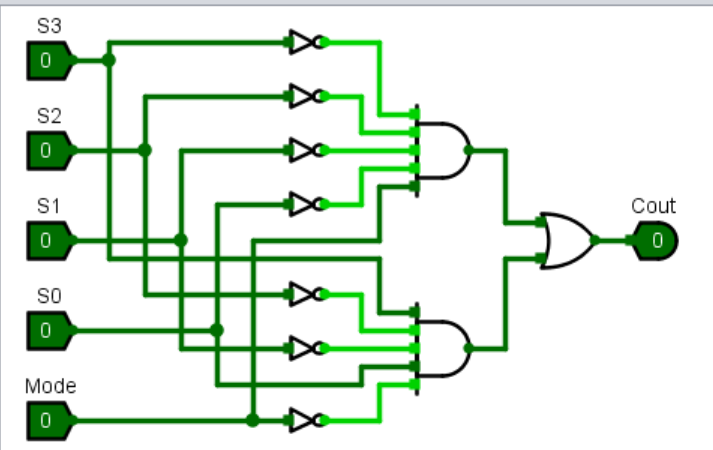


图3-13 模10可逆计数器输出函数电路

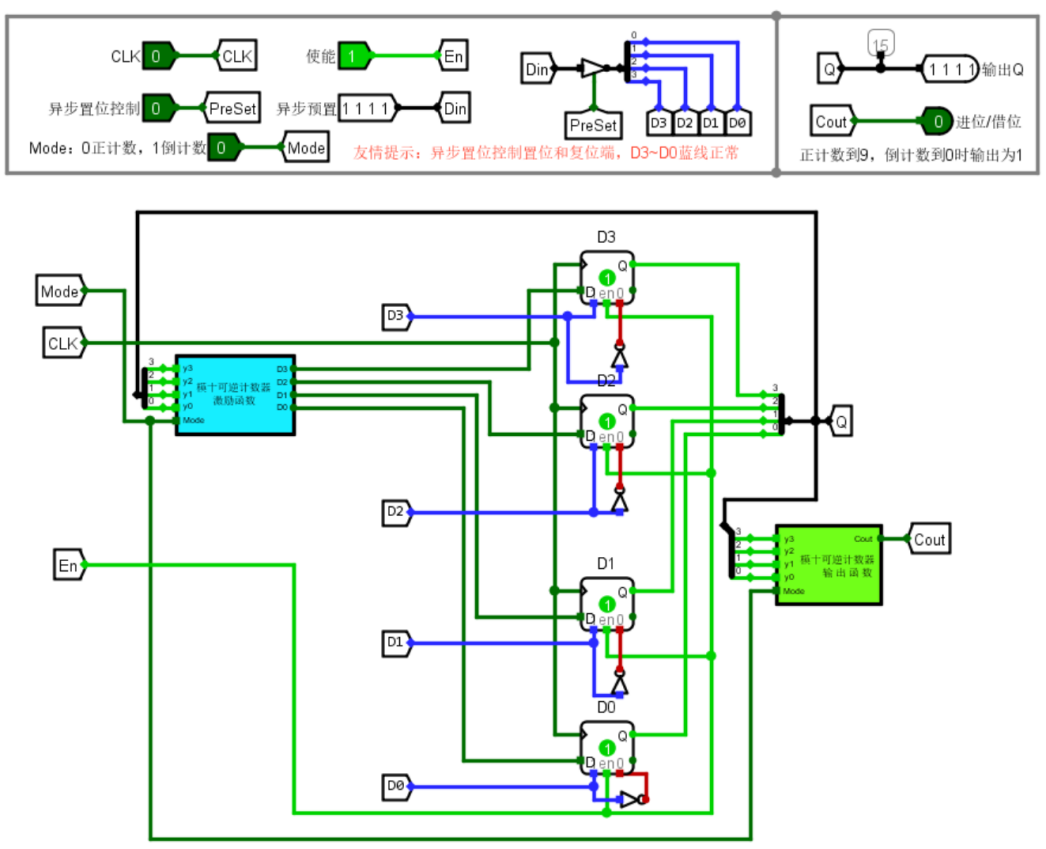


图3-14 模10可逆计数器(可异步置位)电路

1. 测试图

利用驱动测试电路结合Educoder平台的自动测试功能，完成对电路的测试工作。

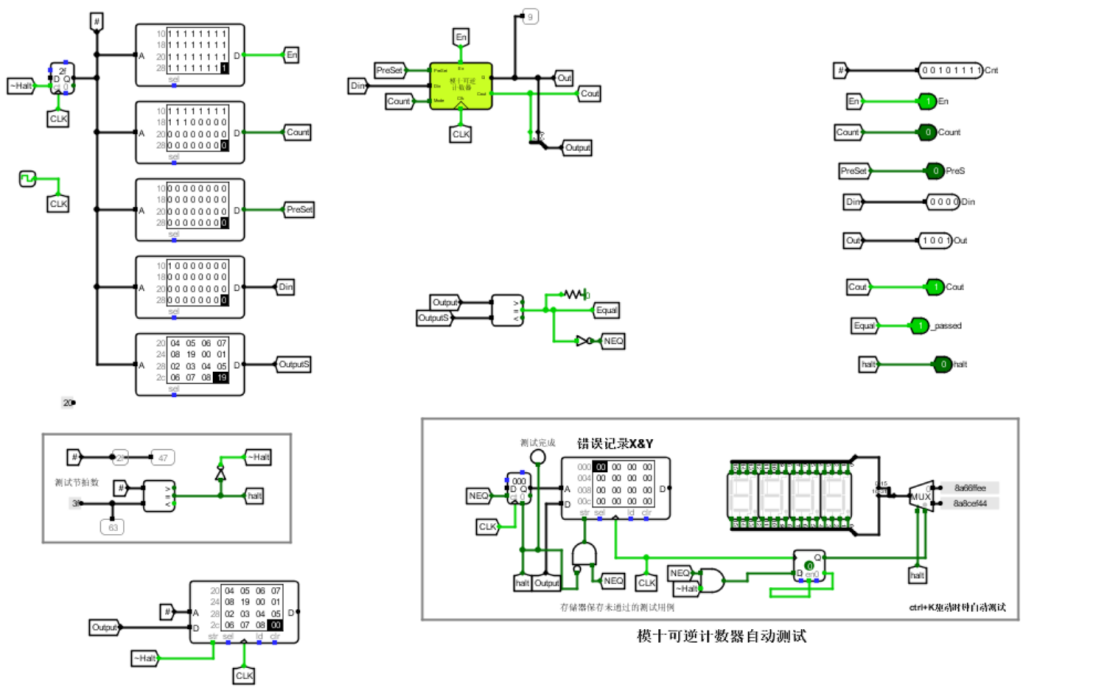


图3-15 模10可逆计数器(可异步置位)测试电路

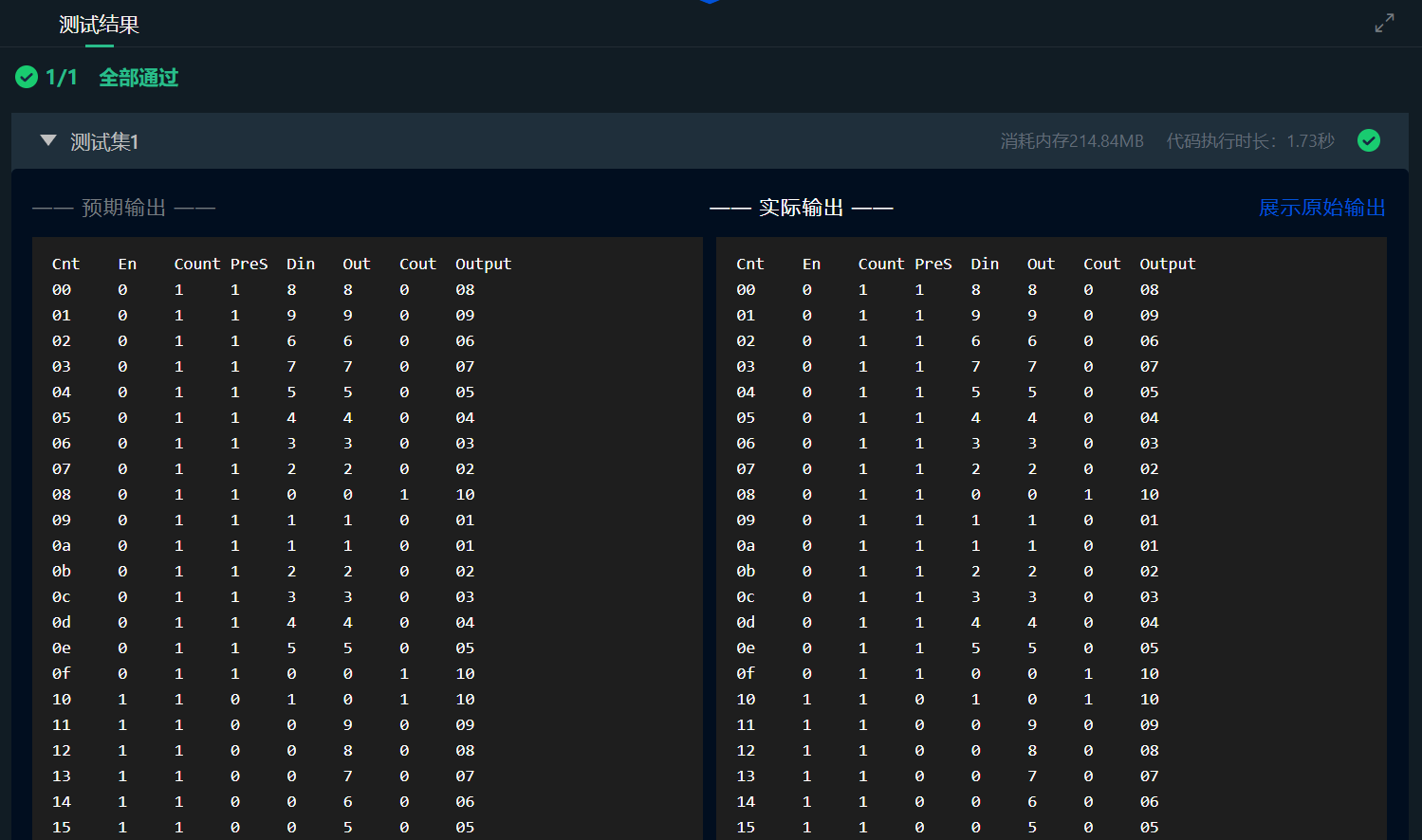


图3-16 模10可逆计数器(可异步置位)测试结果图

1. 测试分析

测试过程中，我们将测试不同的输入组合（如Mode、En、PreSet、Din），并观察计数器的状态输出（Q0-Q3）和进位/借位输出（Cout），对于Educoder平台上的64个测试用例，都正常通过，计数功能和置位功能都符合要求，验证了电路的正确性。

## 两位十进制可逆计数器

1. 设计思路及设计过程

与之前8位无符号比较器的设计思路类似，两位十进制可逆计数器可以由两个模十可逆计数器（4位二进制计数器）级联而成的。每个模十可逆计数器负责一位数的计数，整个两位计数器负责两个十进制数字的计数。通过级联设计，计数器能够进行两位十进制的正向和反向计数。

在输入输出方面，与模10可逆计数器类似，只是输入输出位数发生了改变，具体设计过程如下。

由于每个计数器只能表示0到9之间的数，因此我们使用已经设计完成的模十可逆计数器（4位二进制计数器）作为每一位数字的计数器。低位计数器负责从0到9的单位数计数，而高位计数器则负责十位数的计数。为了实现两位十进制计数，首先设计一个模十可逆计数器用于低位（个位）。当低位计数器从9回到0时，它会产生一个进位信号Cout。高位计数器通过将低位计数器的Cout连接到其使能端，实现从0到9的十位计数。

为了实现置位功能，我们将Dim分解为高四位和低四位，分别接入两个模10可逆计数器的Din端口，实现置位。

对于CLK,Preset,Mode信号，两个计数器同时接受相同输入，而高位计数器的En端应接受En信号和低位计数器Cout的与(原因在上文谈过)。现在我们可以开始连接整个电路了。

1. 电路图

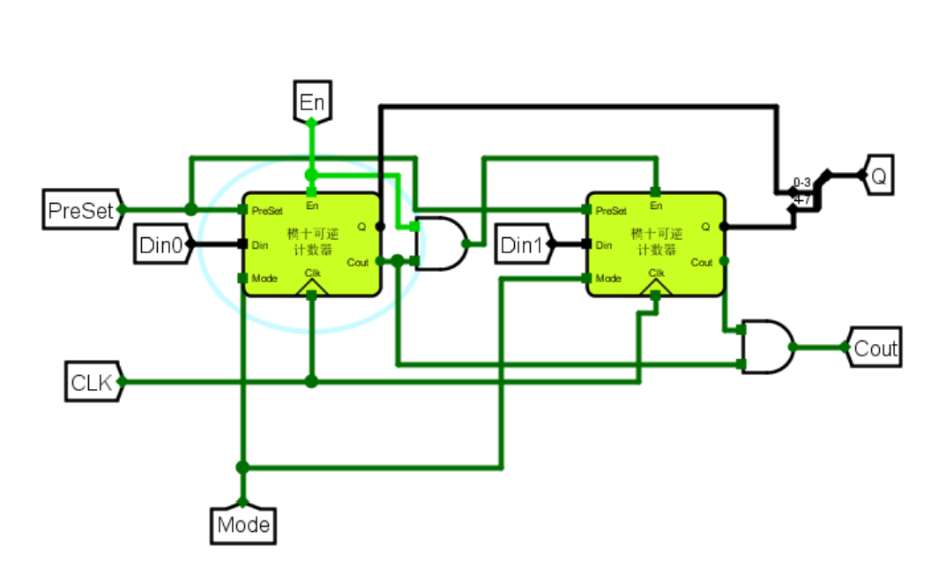


图3-17 两位十进制可逆计数器电路图

1. 测试图

利用测试电路结合Educoder平台的自动测试功能，完成对电路的测试工作。

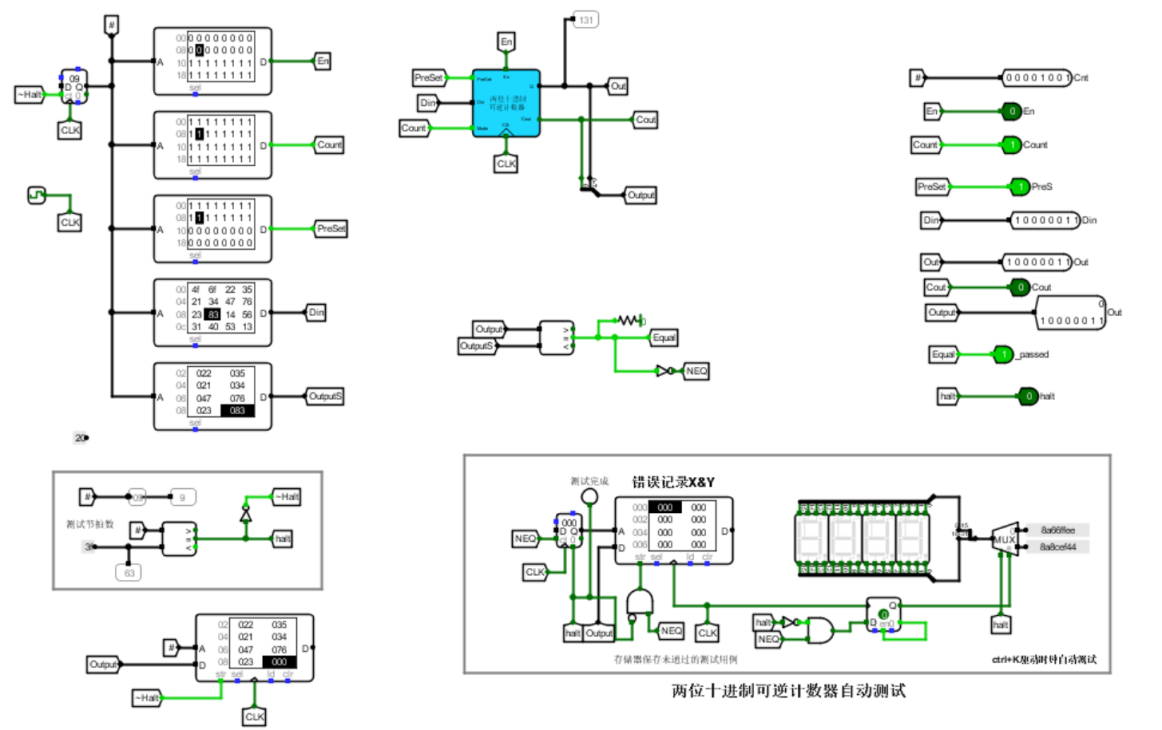


图3-18 两位十进制可逆计数器测试电路图

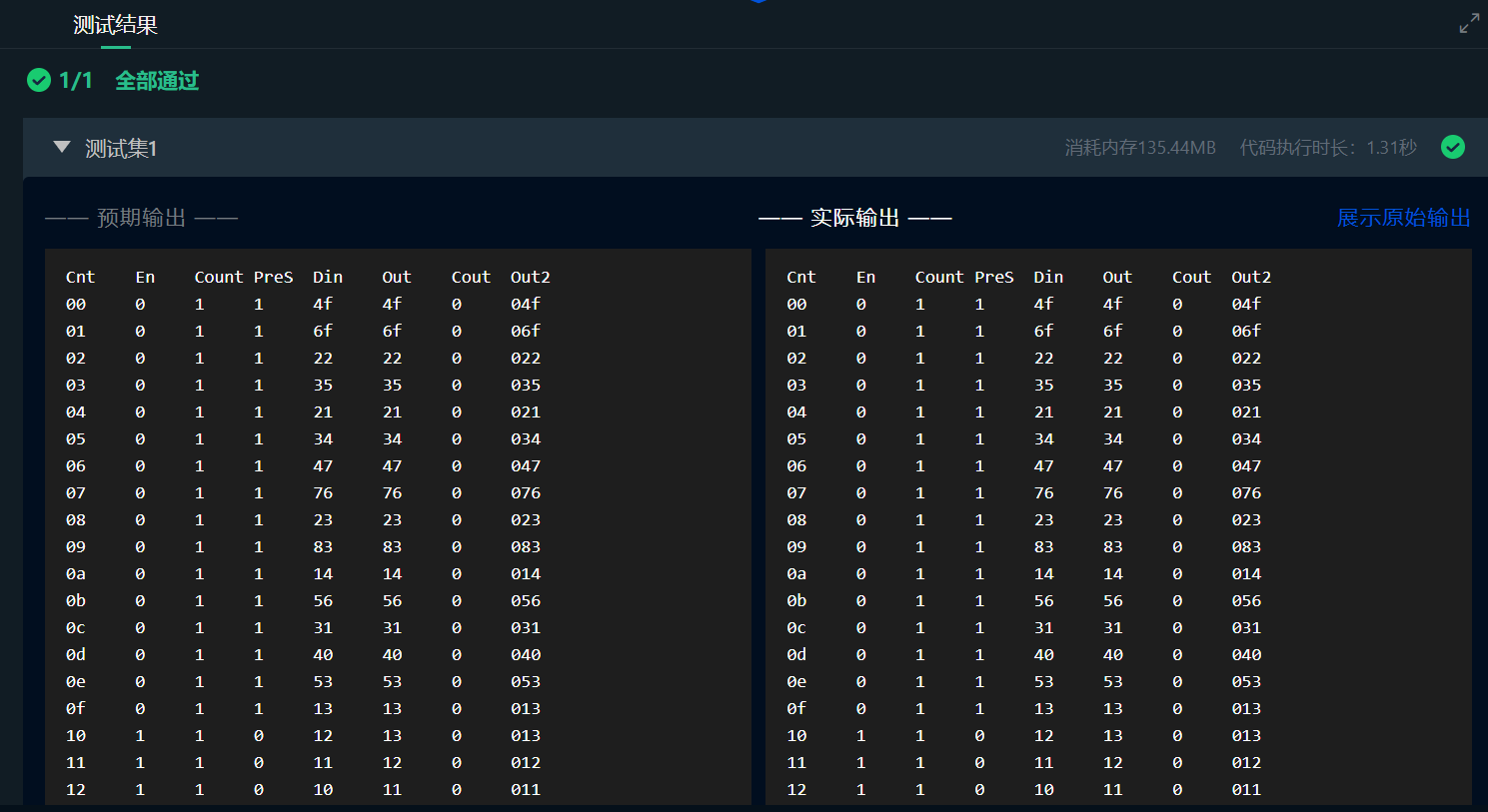


图3-19两位十进制可逆计数器测试结果图

1. 测试分析

本地利用测试电路测试没有错误记录，头歌的64个测试用例也全部通过，计数，进位和置位功能都正常运转，验证了电路的正确性。

## 交通灯状态机

1. 设计思路及设计过程

本部分的设计是实验的重难点，关键是做出正确的状态图，明确状态间的转换关系，从而得出激励函数表达式。根据需求描述，我们首先可以确定以下七个状态和引脚表(电路的输入输出)

表3-5 交通灯控制器状态描述表

|  |  |  |  |
| --- | --- | --- | --- |
| 状态编号 | 状态描述 | 状态编码 | 说明 |
| S0 | 主、次干道均为黄灯闪烁 | 000 | 无倒计时 |
| S1 | 非高峰期主干道绿灯 | 001 | 倒计时12s（计时器从15-04） |
| S2 | 高峰期主干道绿灯 | 010 | 倒计时27s（计时器从30-04） |
| S3 | 主干道黄灯 | 011 | 倒计时3s（计时器从03-00） |
| S4 | 次干道绿灯 | 100 | 倒计时12s（计时器从15-04） |
| S5 | 次干道黄灯 | 101 | 倒计时3s（计时器从03-00） |
| S6 | 紧急状况 | 110 | 主干道绿灯，次干道红灯，显示99s |

表3-6 交通灯状态转换引脚表

|  |  |  |  |
| --- | --- | --- | --- |
| 信号 | 输入/输出 | 位宽 | 说明 |
| y2~y0 | 输入 | 3 位 | 当前状态y |
| T1 | 输入 | 1 位 | 主干道绿灯结束 |
| T2 | 输入 | 1位 | 主干道黄灯结束 |
| T3 | 输入 | 1位 | 次干道绿灯结束 |
| T4 | 输入 | 1位 | 次干道黄灯结束 |
| Online | 输入 | 1位 | 紧急状况信号 |
| H | 输入 | 1位 | 高峰期信号 |
| MR | 输入 | 1位 | 主干道通行请求信号 |
| SR | 输入 | 1位 | 次干道通行请求信号 |
| D2~D0 | 输出 | 3位 | 对于D触发器的激励 |

接下来，我们需要从系统需求中提取出状态转换的时机，做出状态图，具体而言，状态的转换关系如下。

S0：初始状态，当有主干道通行请求时，次态为S1；当有主干道通行请求和高峰期时，次态为S2；当只有次干道通行请求时，次态为S2;当有紧急状态信号时，次态为S6；

S1：非高峰期主干道绿灯状态，当有主干道绿灯结束信号时，次态为S3；当有紧急状态信号时，次态为S6；

S2：高峰期主干道绿灯状态，当有主干道绿灯结束信号时，次态为S3；当有紧急状态信号时，次态为S6；

S3：主干道黄灯：黄灯结束时(T2)，当有次干道通行请求时，次态为次干道绿灯S4；当只有主干道通行请求时，次态为S1(无高峰期信号)，S2(有高峰期信号);当有紧急状态信号时，次态为S6；当无通行请求时，次态为S0；

S4：次干道绿灯：绿灯结束后，次态为S5

S5：次干道黄灯：黄灯结束后，当有主干道通行请求时，次态为S1(无高峰期信号)，S2(有高峰期信号);当有紧急状态信号时，次态为S6；当无通行请求时，次态为S0；

当只有次干道通行请求时，次态为S4   
 S6：紧急状态：结束后，高峰期时，进入高峰期主干道绿灯状态S2；紧急状态结束，非高峰期时，进入非高峰期主干道绿灯状态S1,没有通行请求时，回到初始状态S0。

需要注意的是紧急状态的优先级问题以及在何种输入情况下状态会保持不变，由此得状态图和状态表如下(由于条件较多，状态图仅供理解设计所用，未列出完整得输入条件和对应输出)。

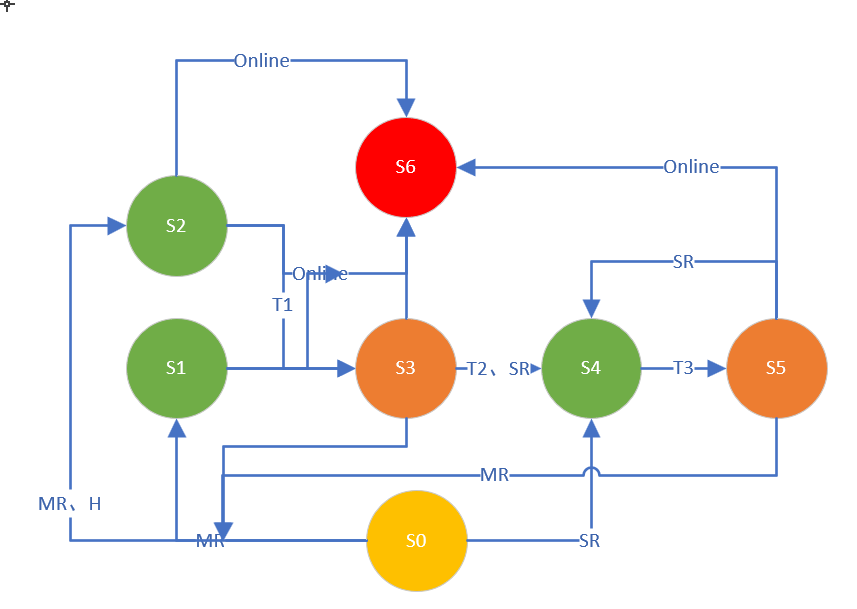
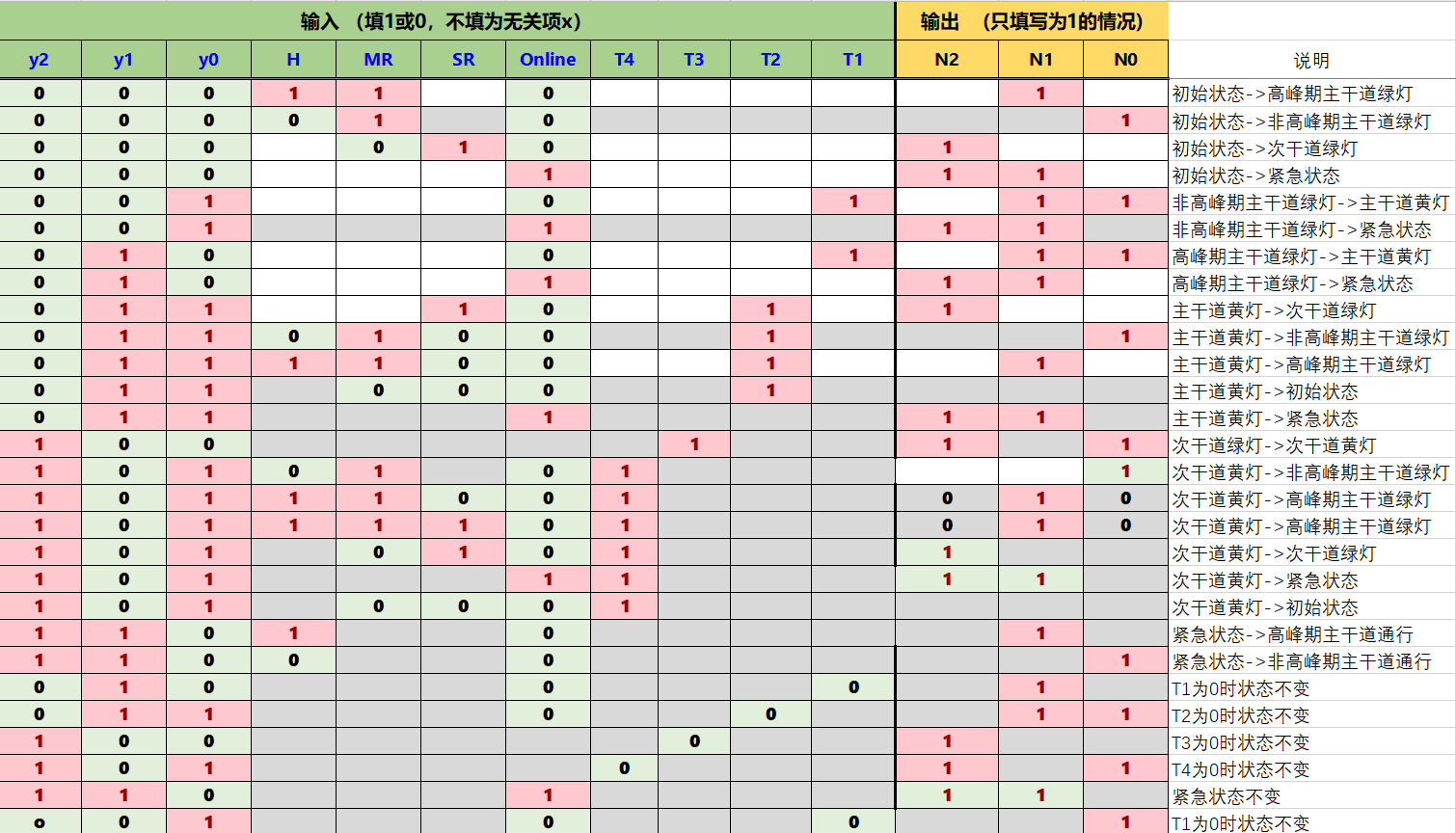


图3-20 交通灯状态转换图

表3-7 交通灯状态转换表



通过状态表，可得对应输出得表达式：

D2=~y2 Online + ~y1 Online + ~y0 Online + ~y1 ~y0 ~MR SR + ~y2 y1 y0 SR T2 + y2 ~y1 ~y0 + y2 ~y1 ~T4 + y2 ~y1 ~MR SR

D1=~y2 Online + ~y2 ~y0 H MR + ~y2 ~y1 y0 T1 + ~y1 y0 Online T4 + ~y2 y1 ~y0 + ~y2 y1 ~T2 + y1 ~y0 Online + y1 ~y0 H + ~y2 y1 H MR ~SR + y2 ~y1 y0 H MR T4

D0=~y2 ~y1 ~H MR ~Online + ~y2 ~y1 y0 ~Online + ~y2 y0 ~Online ~T2 + ~y2 y0 ~H MR ~SR ~Online + ~y1 y0 ~H MR ~Online + ~y1 y0 H ~MR ~Online ~T1 + ~y1 y0 H ~SR ~Online ~T1 + ~y2 y1 ~y0 ~Online T1 + y2 ~y1 ~y0 T3 + y2 ~y1 y0 ~T4 + y2 y1 ~y0 ~H ~Online

由此可由logisim自动生成电路如下，由于图片过长，这里只截出D2的部分

1. 电路图

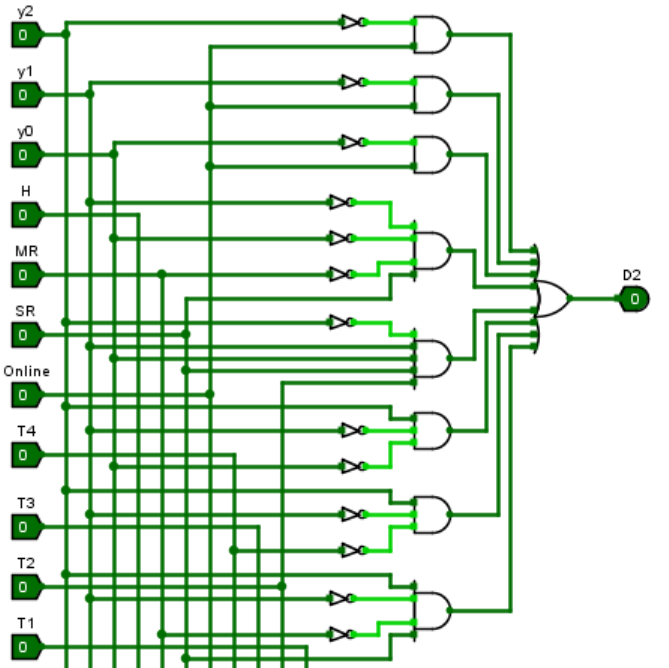


图3-20 交通灯激励函数电路图

1. 测试图

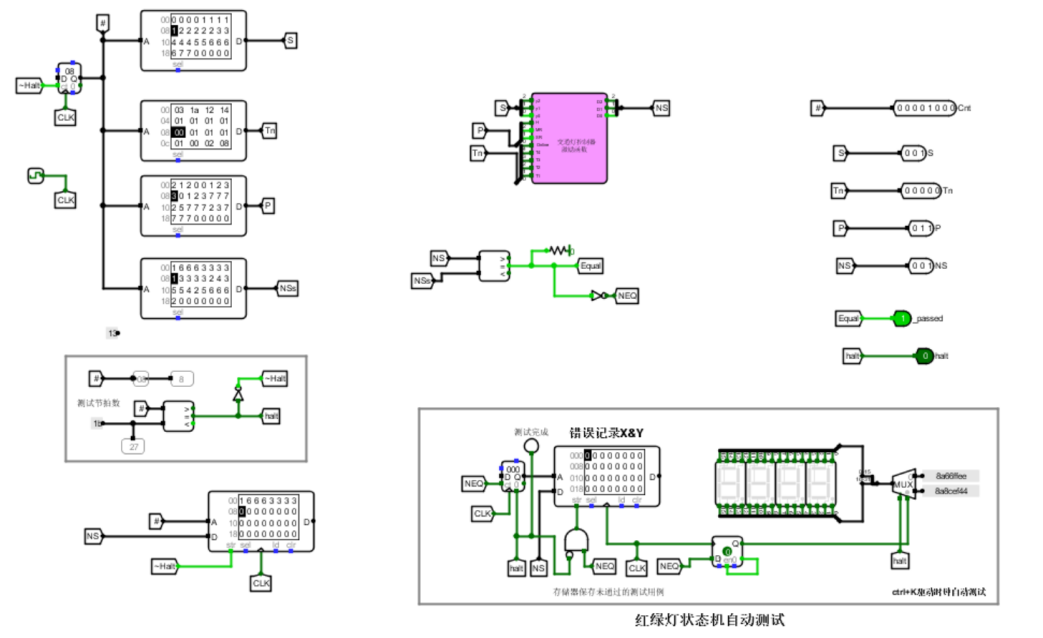


图3-21 交通灯激励函数测试电路图

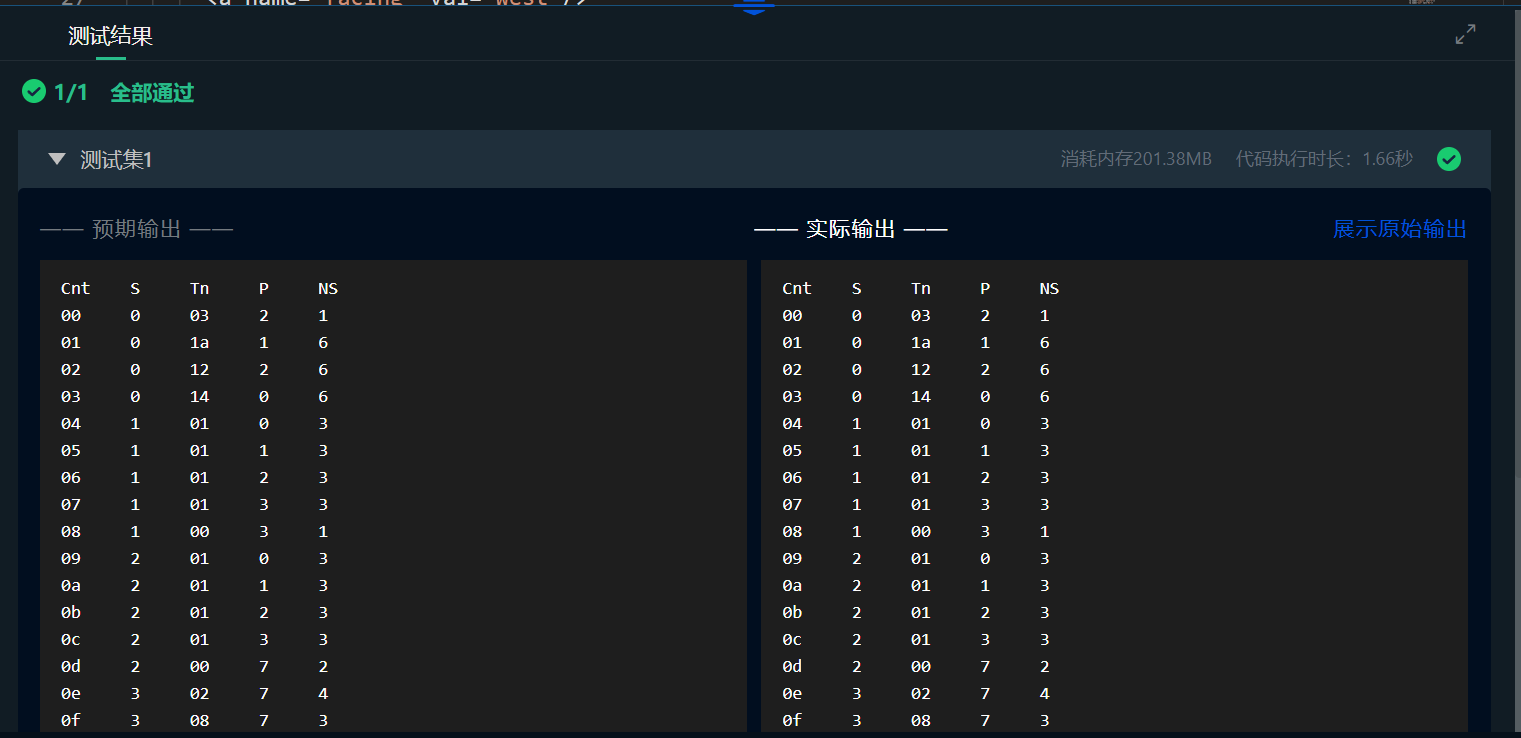


图3-22 交通灯激励函数测试结果图

1. 测试分析

本部分需要处理的输入情况较为复杂，即使完全通过头歌提供的22个测试用例，也可能在后续系统中出现问题,对此还需要结合测试电路和手动测试，对以下情况进行测试：

(a)测试输入信号（高峰期信号H、主干道请求MR、次干道请求SR、紧急状态信号Online）在不同情况下的状态转换是否符合设计要求。例如，当H=1，MR=1时，状态机应从初始状态S0转到S2状态（高峰期主干道绿灯）。类似地，测试各种输入信号组合，确保状态机的转换正确。

(b)倒计时信号测试：测试倒计时信号T1~T4的工作是否正常，确保倒计时结束后，状态机正确地从一个状态转移到下一个状态。例如，在S1状态下，T1为0时，系统应转换到S3状态（主干道黄灯）；T1信号结束后，若没有其它的请求，系统应返回到S0状态。

(c)紧急状态测试：测试紧急状态的切换是否立即生效。例如，当Online=1时，如果是主干道通行状态，系统应立即切换到紧急状态，如果是次干道通行状态，则应等待黄灯结束后再切换到紧急状态

(d)状态保持测试(容易遗漏的点)：测试每个状态在倒计时为0时的保持功能，确保系统在无输入信号变化时保持当前状态。例如，当处于S3状态时，如果T2信号为0，系统应保持在S3状态，直到条件满足下一个状态转换的要求。

测试完成后，无错误产生，验证了最终电路的正确性。

## 交通灯输出函数设计

1. 设计思路及设计过程

不难发现，交通灯控制系统的输出只和当前电路状态，因此有关函数设计应该基于Moore型状态机的原则，输出信号只与当前状态相关，而不依赖于输入信号。这意味着每个状态有一组固定的输出信号，控制交通灯的红、黄、绿灯以及通行信号。根据交通灯的状态机（如S0、S1、S2等），我们为每个状态分配对应的输出信号，确保交通灯的显示与状态保持一致。由于电路只有3位输入，较为简单，我们可以直接利用真值表构建电路。

表3-8 交通灯控制器输出函数引脚表

|  |  |  |  |
| --- | --- | --- | --- |
| 信号 | 输入/输出 | 位宽 | 说明 |
| R1 | 输出 | 1 位 | 主干道红灯控制信号 |
| Y1 | 输出 | 1位 | 主干道黄灯控制信号 |
| G1 | 输出 | 1位 | 主干道绿灯控制信号 |
| R2 | 输出 | 1位 | 次干道红灯控制信号 |
| Y2 | 输出 | 1位 | 次干道黄灯控制信号 |
| G2 | 输出 | 1位 | 次干道绿灯控制信号 |
| PASS0 | 输出 | 1位 | 初始状态通行信号 |
| PASS1 | 输出 | 1位 | 主干道通行信号 |
| PASS2 | 输出 | 1位 | 次干道通行信号 |
| PASS3 | 输出 | 1位 | 紧急状况通行信号 |

根据输出引脚表，构建真值表如下

表3-9 交通灯控制器输出函数真值表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| y2 | y1 | y0 | R1 | Y1 | G1 | R2 | Y2 | G2 | PASS0 | PASS1 | PASS2 | PASS3 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |

通过真值表，可得逻辑函数表达式

R1=y2 ~y1 + y2 y0 R2=~y2 y0 + y1

Y1=~y2 ~y1 ~y0 + y1 y0 Y2=~y2 ~y1 ~y0 + y2 y0

G1=~y2 ~y1 y0 + y1 ~y0 G2=y2 ~y1 ~y0

PASS0=~y2 ~y1 ~y0 PASS2=y2 ~y1

PASS1=~y2 y0 + ~y2 y1 PASS3=y2 y1

由此构建电路图如下：

1. 电路图

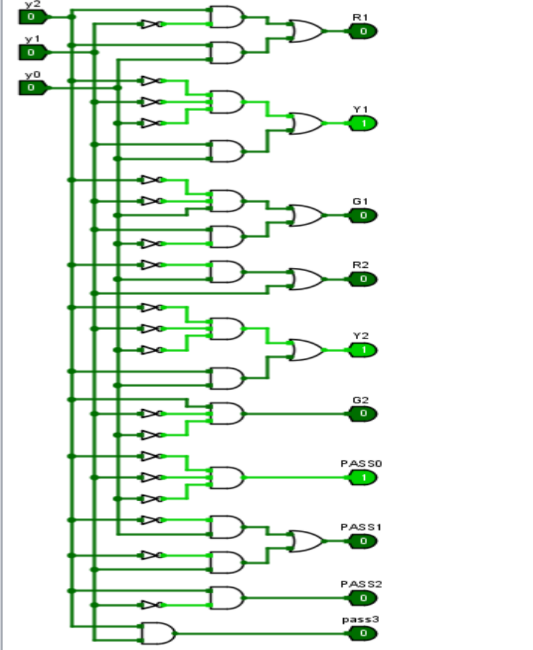


图3-23 交通灯输出函数电路图

1. 测试图

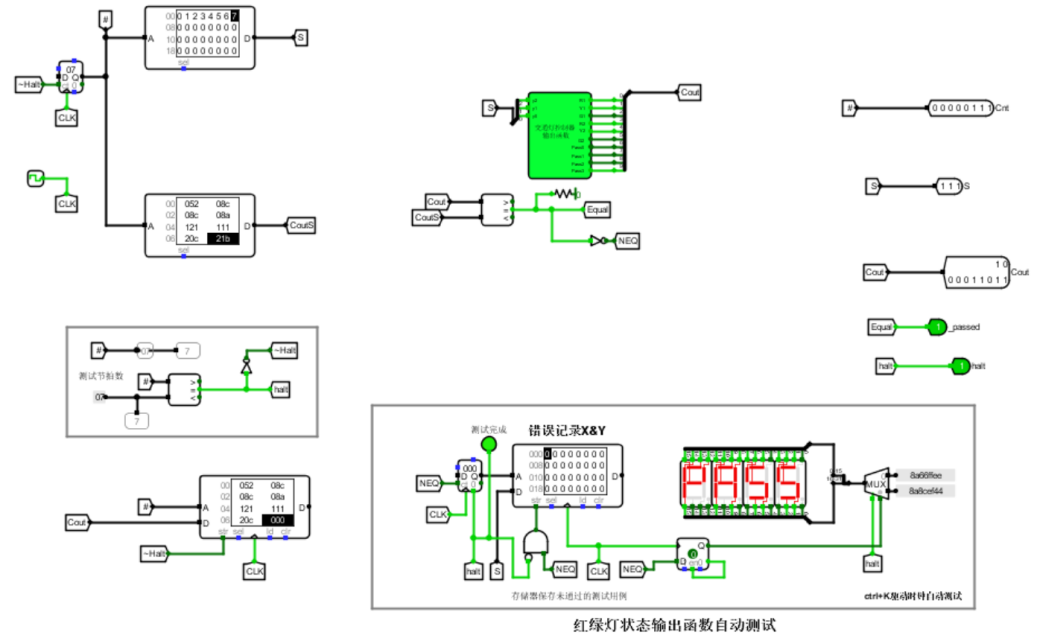


图3-24 交通灯输出测试电路图

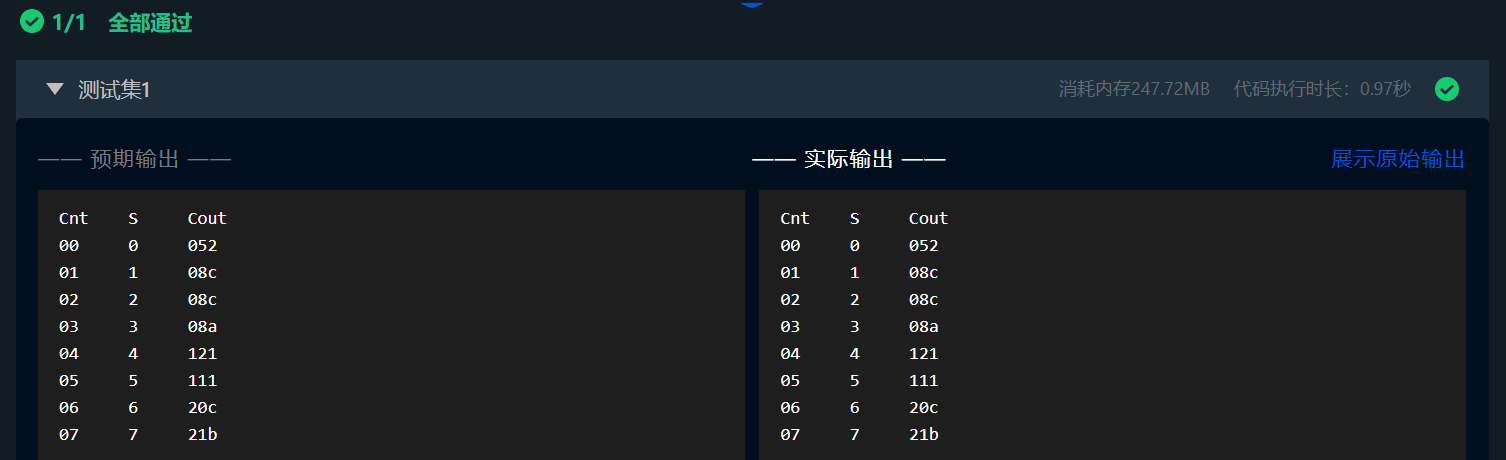


图3-25 交通灯输出测试结果图

1. 测试分析

对于八种可能的输入，都得到的与理论匹配的输出，验证了程序的正确性。

## 交通灯控制系统

1. 设计思路及设计过程

现在，我们以及完成了所有模块，子电路的设计工作，并通过测试电路验证了正确性，为了完成整个控制系统，我们需要准确处理各模块输入和输出的关系，现在，让我们再整理一遍整个系统所涉及的输入输出信号以及内部信号。

表3-10 交通灯控制器的输入及输出信号

|  |  |  |  |
| --- | --- | --- | --- |
| 信号 | 输入/输出 | 位宽 | 说明 |
| H | 输入 | 1 位 | 高峰期信号 | |
| MR | 输入 | 1 位 | 主干道通行请求信号 | |
| SR | 输入 | 1 位 | 次干道通行请求信号 | |
| Online | 输入 | 1 位 | 紧急状况 | |
| R1 | 输出 | 1 位 | 主道红灯控制信号 | |
| Y1 | 输出 | 1 位 | 主道黄灯控制信号 | |
| G1 | 输出 | 1 位 | 主道绿灯控制信号 | |
| R2 | 输出 | 1 位 | 次干道红灯控制信号 | |
| Y2 | 输出 | 1 位 | 次干道黄灯控制信号 | |
| G2 | 输出 | 1 位 | 次干道绿灯控制信号 | |

表3-11 交通灯控制系统内部信号说明

|  |  |  |
| --- | --- | --- |
| 信号 | 输入信号描述 | 说明 |
| T1 | 主干道绿灯结束信号 | 主干道倒计时为04 |
| T2 | 主干道黄灯结束信号 | 主干道倒计时为01 |
| T3 | 次干道绿灯结束信号 | 次干道倒计时为04 |
| T4 | 次干道黄灯结束信号 | 次干道倒计时为01 |
| T5 | 主干道通行状态结束信号 | 主干道倒计时为00 |
| T6 | 次干道通行状态结束信号 | 次干道倒计时为00 |
| PASS0 | 初始状态通行信号 | 初始信号时PASS0=1 |
| PASS1 | 主干道通行信号 | 主干道绿灯或主干道黄灯时PASS1=1 |
| PASS2 | 次干道通行信号 | 次干道绿灯或主干道黄灯时PASS2=1 |
| PASS3 | 紧急状况通行信号 | 紧急状况下PASS3=1 |

在本实验中，最终的交通灯控制系统设计是基于之前设计完成的各个子电路和模块的组合。核心目标是将所有模块、信号和功能整合起来，确保交通灯能够根据不同的输入信号（如高峰期信号、主干道通行请求信号、次干道通行请求信号和紧急状态信号）自动调整工作状态，控制交通灯的红、黄、绿灯信号，并根据倒计时进行切换。系统通过状态机管理状态变化，倒计时电路控制时间逻辑，显示电路展示倒计时信息。

（a)对于状态机模块，需要负责根据交通信号输入（如主干道请求、次干道请求、高峰期信号、紧急状态信号等）进行状态转换，决定交通灯的状态（红、黄、绿灯）和当前通行的道路（主干道或次干道）。这里我们只需要将对应的信号与引脚相连即可，将现态输出到Status，通过激励函数，输出函数电路和一个D触发器，实现状态的转换功能。

(b)对于倒计时选择模块，我们需要通过三个2路选择器选择合适的Timer信号传递给七段译码器。分析可知：

当PASS0=1时为初始状态，第一个二路选择器输出00，作为第二个二路选择器的输入，PASS2和PASS3均为0，所以连接到0端，最终将0传递给七段显示译码器，显示00；

当PASS0=0时，第一个2路选择器输出Timer1作为第二个2路选择器0端口的输出。

再判断PASS2(这里不需要再判断PASS1了，因为第二个二路选择器的输入分别为Timer1和Timer2)。若PASS2=1，则输出Timer2信号给第三个2路选择器，此时判断PASS3(紧急状态)，当PASS3=1时，直接输出99，否则输出Timer2信号，最终显示次干道通行倒计时；若PASS2=0，则此时PASS1应等于1，第二个二路选择器输出Timer1信号给第三个二路选择器，类似地，再判断PASS3是否等于1，因为紧急状态的优先级要更高。最终，显示模块接受到七段显示译码器的信号和输出函数输出的RGB信号(控制红绿灯颜色)完成红绿灯显示工作。

(c)次干道通行倒计时模块

次干道通行倒计时模块相对比较简单，主要有两位十进制可逆计数器和8位无符号比较器组成，根据需求和两位十进制可逆计数的引脚，不难想到将15作为置位输入Din，PASS2(次干道通行)作为En使能端输入，Mode设置为1(因为是倒计时，所以-1计数),时钟端接入CLK信号，那么什么时候进行置为(Preset)操作呢？显然是本轮15秒倒计时结束，即T6为1时。那么后面无符号比较器的功能也比较清楚了,分别与4，1，0比较，输出E给相应的倒计时结束信号T3(次干道绿灯结束信号)、T4(次干道黄灯结束信号)、T6(次干道通行状态结束信号)即可，至此完成了次干道通行模块的设计操作。

(d)主干道通行倒计时模块

主干道通行模块的设计理念与流程与次干道通行模块保持一致，然而，必须考虑高峰期及紧急状态下的交通控制（次干道通行模块则无需考虑紧急状态，因其通行需等待黄灯周期结束方可切换至紧急状态）。首先，输入信号Din的设定需依据高峰期信号(H)的不同取值（30或15）进行选择，该功能可通过二选一选择器实现；时钟输入端同样与CLK信号1相连；三个8位无符号比较器负责输出倒计时结束信号T1（主干道绿灯结束信号）、T2（主干道黄灯结束信号）以及T5（主干道通行状态结束信号）。在此设计基础上，主干道通行模块的可逆计数器的使能端的输入应满足PASS1=1且PASS0=PASS3=0，我们用与门实现，置为端Preset的输入应满足主干道同行状态结束或次干道通行状态结束或紧急状态结束，由此可完成主干道通行倒计时模块的设计。

综上所述，设计完成后的电路如下

1. 电路图

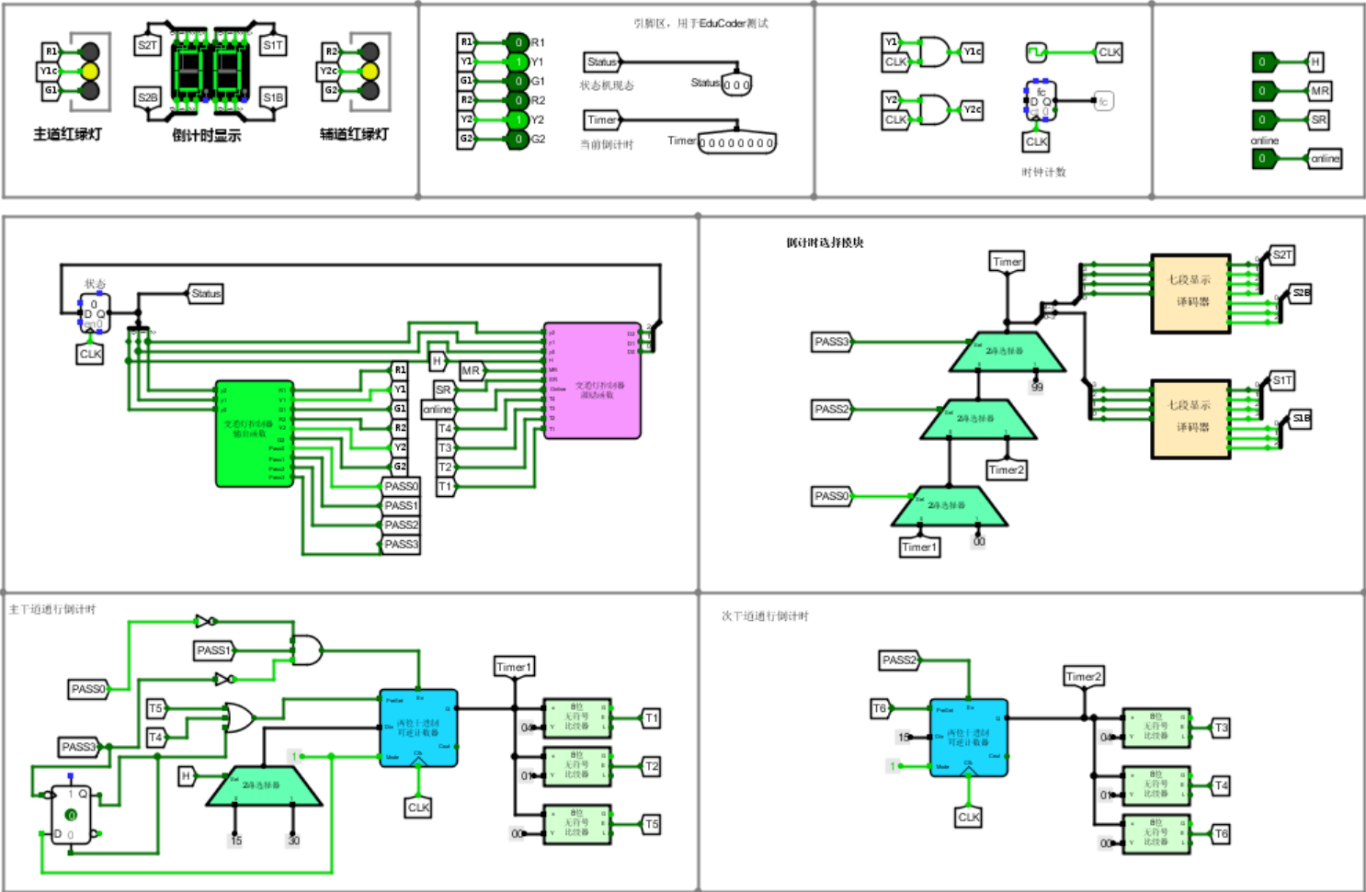


图3-26 交通灯控制系统电路图

1. 测试图

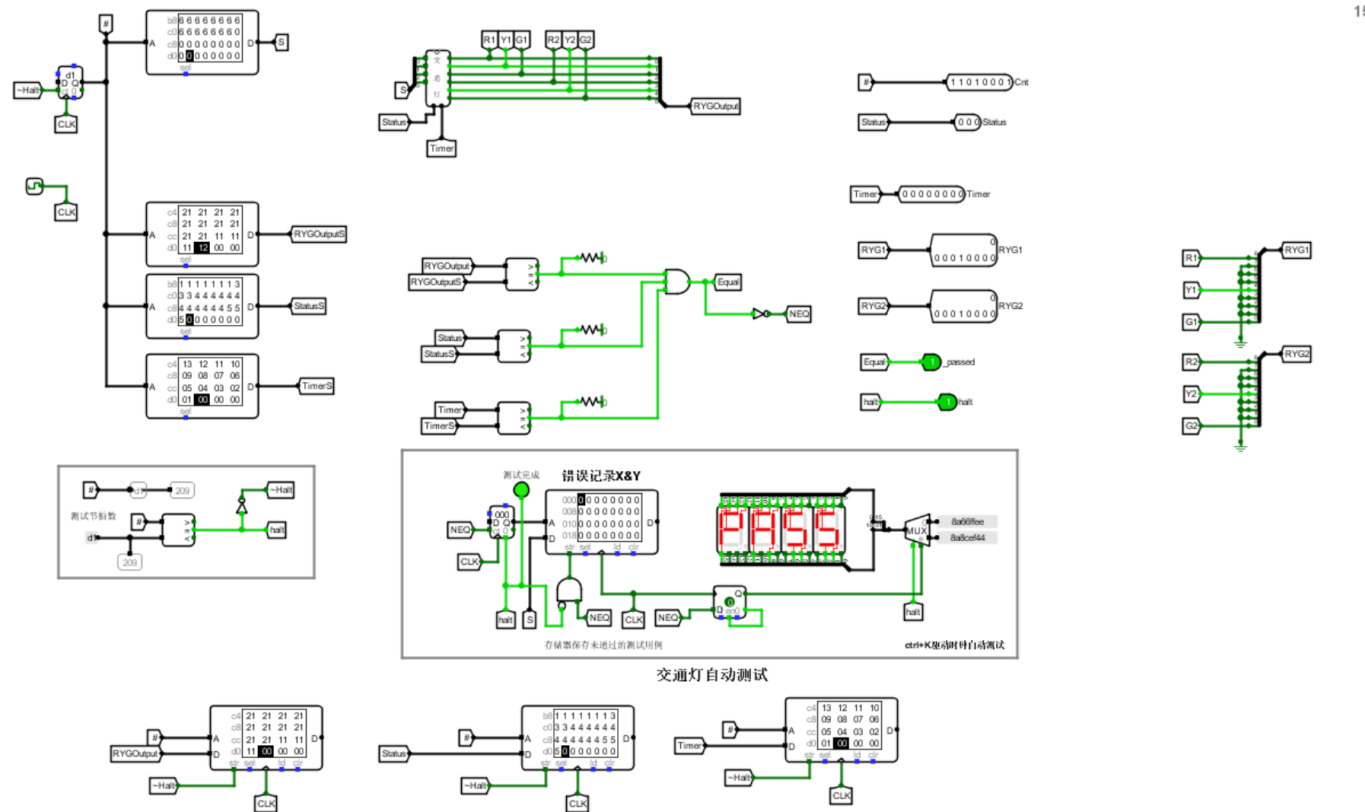


图3-27 交通灯控制系统测试电路图

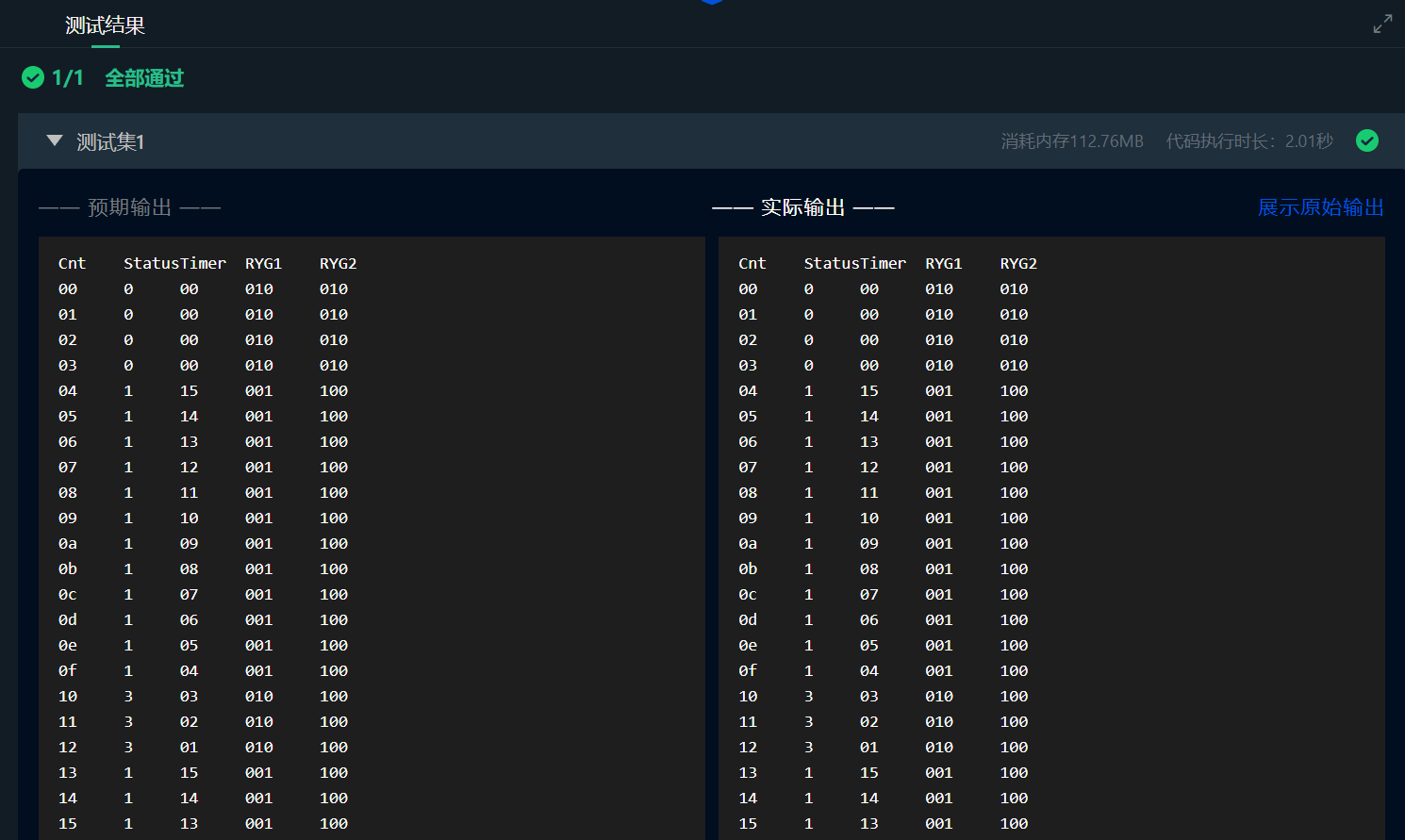


图3-28 交通灯控制系统测试结果图

1. 测试分析

在完成整个交通灯控制系统的设计后，我们对系统进行了详尽的测试分析。在本实验中，我们主要对系统的状态转换、倒计时功能、交通灯状态输出、显示模块以及紧急状态响应等方面进行详细测试。通过一系列输入条件的模拟，确保系统在不同的场景下都能够稳定运行。经过对电路的反复修正，最终系统在状态转换，倒计时功能，状态保持，红绿灯显示，紧急状态和初始状态等方面都实现了预期的效果。

电路顺利通过了头歌提供的210个测试样例，本地利用测试电路测试同样无错误产生，验证了系统的正确性。

# **设计总结与心得**

## 实验总结

在本次数字电路与逻辑设计课程的实验中，我们设计并实现了一个完整的交通灯控制系统。通过整个实验过程，我对数字电路的设计与实现过程有了更深刻的理解，尤其是在设计与调试过程中，我充分体会到了如何将理论知识转化为实际可操作的电路，如何根据不同的需求从器件到部件，从部件到系统逐步完成一个复杂系统的设计。

本实验涵盖了多个重要模块的设计与实现，具体包括交通灯状态机的设计、倒计时模块的设计、显示模块的实现、输入输出信号的控制、以及最终的交通灯控制系统集成。在设计过程中，有不少基础器件是理论课程中学习过的(如七段显示译码器，无符号比较器，可逆计数器等等),通过自己动手实现这些基本器件，我对这使得我对它们的工作原理有了更加直观的认识，对理论也更加熟悉。

同时，通过实验，我能更加熟练地使用Logisim软件进行数字电路设计，学习使用自动仿真功能，使得整个设计过程更为高效，使用真值表和逻辑表达式自动生成电路，简化了电路设计过程。同时，通过Excel输入状态表自动生成逻辑表达式，也让我看到了现代工具的强大之处，这为电路设计简化了很多步骤(不再需要人为画卡诺图来得到逻辑函数表达式)。

交通灯控制系统的难点和核心是状态机的设计。通过本实验，我学到了如何通过状态机来描述系统的行为，如何根据输入信号和当前状态来确定下一状态的转换。通过画状态表和状态图，我们能够清晰地描绘出交通灯控制系统的每个状态和状态之间的转换关系，这让我深刻认识到状态机在实际系统中的重要性。在设计过程中，我们通过逐步细化状态转换规则，确保状态机能够按预期工作，并能在不同条件下做出正确响应。

总之，这次实验不仅让我深入了解了数字电路与逻辑设计的基本原理，还提高了我在系统设计、调试、优化等方面的能力。通过亲自设计并实现一个完整的交通灯控制系统，我对数字电路的实际应用有了更直观的理解，也为未来更复杂系统的设计打下了坚实的基础。

### 遇到的问题及处理

整个交通灯控制系统的设计过程中，我遇到了不少问题，通过分析测试失败的原因，利用头歌平台给出的框架，我逐渐理解了实验要求和问题原因，以下是我遇到的主要问题及处理方法：

**(a)导线重叠：**在添加导线过程中，有可能因为意外放置了了多余导线，由于与存在和其它导线或器件的重叠，肉眼难以发现，导致实验过程中尽管逻辑正确，也经常出现未知错误。经过排查和老师指导，我明白了问题的原因和解决方法，只需要用鼠标框选住存在问题的部分，便可发现多余的导线，然后按下delete键删除即可。

**(b)状态转换缺失：**在设计激励函数时，由于对实现需求和电路设计的理解不足，容易遗漏某些状态转换关系，比如在没有相应T信号时，电路状态应保持不变，还有通行状态和紧急状态切换的优先级等问题。通过检查头歌平台测试未通过的样例，我明白了问题原因，并在状态表中补充相应的状态转换规则。

**(c)自动生成逻辑函数表达式错误：**在设计交通灯激励函数时，老师为我们提供了一个EXCEL转换表，可以根据我们输入的状态表自动生成输出D对应的表达式。但是在后面的实验过程中我发现尽管我在状态表加入了特定规则，生成后的逻辑函数表达式对应的真值表仍然错在些许错误，通过手动对真值表进行修改，我=成功修正了错误

**(d)没有理解倒计时表示方法：**本实验中倒计时并不是直接用以8位二进制数表示，而是将其看成两个四位二进制数，分别表示高位和低位。起初我没有理解这一点，导致在两位十进制可逆计数器和后续交通灯控制系统的设计中存在错误，发现倒计时并不是连续变化而是发生了跳变。通过分析倒计时的2进制表示和变化，我理解了这一点并对可逆计数器的设计和通行模块的Din输入进行了修正，成功达到了理想结果。

### 设计方案存在的不足

尽管顺利完成了实验，但是我也认识到自己很大程度上依赖于头歌平台给出的框架和指导，也大量使用了logisim的自动生成功能，设计时十分方便，但最终设计的电路也存在一些问题。

**(a)部分电路过于复杂：**利用Logisim软件，通过逻辑表达式或真值表自动生成电路的过程，虽然在一定程度上实现了电路设计的自动化，但生成的电路设计往往并非最优解。这些电路在结构复杂度和效率方面存在明显的不足。具体而言，自动生成的电路可能过于繁琐，包含不必要的冗余连接和多余的逻辑门。为了进一步提升电路系统的效率，本研究计划在未来的设计中采取手动优化电路的策略。该策略将着重于减少逻辑元件的数量，同时强化对电路精简性和操作性的关注，以期达到提升电路性能和降低资源消耗的目标。

**(b)紧急状态结束后的处理：**当在紧急状态解除后，恢复主干道通行状态的过程中，本研究发现主干道通行倒计时的重置至15秒或30秒是必要的。初始阶段，由于未考虑此因素，导致了测试的不通过。经过深入分析，本研究通过引入D触发器解决了该问题，并成功通过了后续测试。然而，进一步的评估揭示了使用D触发器虽然解决了问题，但带来了较高的切换延迟，这可能影响状态转换的及时性。为了提升系统的响应速度和时效性，未来研究应着重于优化状态机逻辑和信号传递机制，以减少状态切换的延迟。

**(c)系统响应速度问题：**本实验中的交通灯控制系统针对一般的交通情况进行设计，但在高负荷情况下，可能会遇到多个路口信号冲突、频繁状态转换等问题，这对系统的响应速度和稳定性提出了更高要求。未来可以通过优化系统的设计，提升系统在高负荷条件下的响应能力，例如通过提高信号处理的优先级，减少系统中断和延迟，确保系统在高负荷情况下仍然能保持稳定和准确的操作。

**(d)系统扩展性不足：**当前开发的交通信号控制系统针对双路口（包括主干道与次干道）的信号控制进行了优化。然而，若未来需求扩展至更多路口或更复杂的交通流量场景（如四路交叉口或多车道控制需求），现行设计可能缺乏足够的灵活性。系统的状态机、倒计时模块以及显示模块均是针对双路口场景进行优化设计的，当扩展至更多路口时，可能需要对现有结构进行重新设计或大规模修改。因此，在未来的设计中，应考虑采用更加模块化的系统架构，以实现系统的便捷扩展与调整，从而避免引入额外的复杂性。

## 实验心得

在本次交通灯控制系统的设计与实现过程中，我深入探究了数字电路的基本原理，并显著提升了在实际应用中解决技术问题的能力。通过将理论知识应用于实际系统的设计，我经历了从基础元件到系统集成的完整过程，并在实验中不断修正错误、优化设计，最终构建了一个功能完备的交通灯控制系统。这一过程不仅要求我对数字电路的各个组成部分有深刻的理解，还要求能够将这些知识综合运用，以解决实际问题。例如，在设计交通灯的逻辑控制模块时，我必须考虑到交通流量的动态变化，以及如何通过时序逻辑来实现交通灯的合理切换，确保交通的流畅与安全。

本次实验的最大收获在于实现了从理论到实践的转变。在课程中学习的时序逻辑、状态机、计数器、触发器和组合逻辑等基础知识，在实验中得到了全面的应用。这加深了我对这些基础理论在实际数字电路设计中重要性的理解。通过设计交通灯控制系统，我将课程中学到的知识与实际需求相结合，完成了一个相对复杂的数字系统设计。在这一过程中，我不仅复习和巩固了课堂上所学的理论知识，还学会了如何将这些理论知识转化为解决实际问题的工具。例如，在设计状态机时，我必须确保状态转换的准确性和及时性，以避免交通灯出现混乱的信号，这要求我对状态机的设计原则有深入的掌握。

在实验过程中，我的调试与优化能力得到了显著提升。尽管在系统设计阶段遇到了诸多挑战，但通过持续的调试，我掌握了如何通过测试快速定位问题，并通过逐步排查找到解决方案。整个实验过程中，调试不仅提升了我的问题解决能力，还使我对数字电路设计的细节有了更深入的了解，培养了处理复杂问题的能力。在调试过程中，我经常需要与同学进行讨论，共同分析问题所在，并提出创新的解决方案。这种团队合作的经验，不仅锻炼了我的沟通能力，也增强了团队协作精神。

模块化设计的重要性是我从本次实验中获得的宝贵经验。通过将交通灯控制系统划分为多个模块(例如状态机、倒计时模块、显示模块等)，每个模块均可以独立进行设计、调试和优化。这种模块化的设计方法不仅提升了系统的可维护性，也使得设计过程更为高效。在面对问题时，我能够迅速定位到特定模块中的问题，并进行针对性的优化。例如，在倒计时模块的设计中，我发现了一个计时跳变的问题，通过调试，我能够快速地对计数器的逻辑进行调整，而不影响其他模块的正常运行。这种模块化的设计策略，不仅提高了系统的稳定性和可靠性，也为后续的升级和维护提供了便利。

## 意见与建议

在本次实验完成后，结合个人体验，本人对课程提出以下建议：

**(a)关于实验需求描述与讲解：**在初次课程中，通过小组合作探讨交通灯系统状态图的设计，我们发现同学普遍对系统需求存在理解偏差，同时在状态图设计方面也表现出理论知识的不足。建议实验在这些方面进行优化，对需求描述进行更详尽的阐述，并补充相关的前置理论知识。例如，可以引入状态机理论中的关键概念，如状态转换、事件触发和条件判断等，以加深我们对系统需求的全面理解。

**(b)关于Logisim软件的使用：**鉴于大多数同学之前未接触过该软件，导致在实际操作中遇到困难，例如分析组合逻辑功能的使用、导线颜色（蓝色与红色）的含义、快捷键操作等。建议课程组制作教学视频，针对学生普遍存在的疑惑和常见错误进行详细解释。此外，软件本身亦存在一些bug，例如误添加组件后无法删除导致ERROR状态，只能通过编辑源文件来操作电路代码，而理解代码含义及定位错误对学生们而言颇具挑战，且耗时较多。因此，建议实验引入更先进的仿真软件，或对现有解决方案进行额外的讲解。

**(c)关于头歌平台的指导与测试：**头歌平台的测试未提供相应的解释和说明，使得理解各种字母对应的取值增加了学生们的理解与调试难度。建议对头歌平台上的教程进行改进，以便于学生更有效地对电路进行修正。例如，可以增加交互式教学元素，如即时反馈和逐步指导，帮助学生在遇到问题时能够迅速定位并解决问题，从而提升学习效率和理解深度。

|  |
| --- |
| 原创性声明 |
| 本人郑重声明本报告内容，是由作者本人独立完成的。有关观点、方法、数据和文献等的引用已在文中指出。除文中已注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品成果，不存在剽窃、抄袭行为。  已阅读并同意以下内容。  判定为不合格的一些情形：  （1） 请人代做或冒名顶替者；  （2） 替人做且不听劝告者；  （3） 实验报告内容抄袭或雷同者；  （4） 实验报告内容与实际实验内容不一致者；  （5） 实验电路抄袭者。  **作者签名：** |