華中科技大學

2024

数字电路与逻辑设计
实验报告

专　　业：网络空间安全

班　　级：

学　　号：

姓　　名：

电　　话：

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	15			
实验过程	50			
遇到的问题及处理	5			
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

1.1 实验名称

交通灯控制系统设计。

1.2 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控制系统。

实验由简到难，层次递进，从器件到部件，从部件到系统，通过本实验的设计、仿真、验证 3 个训练过程使同学们掌握小型数字电路系统的设计、仿真、调试方法以及电路模块封装的方法。

1.3 实验环境

软件：logisim-hust-20200118.exe 软件一套。

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

1.4 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。

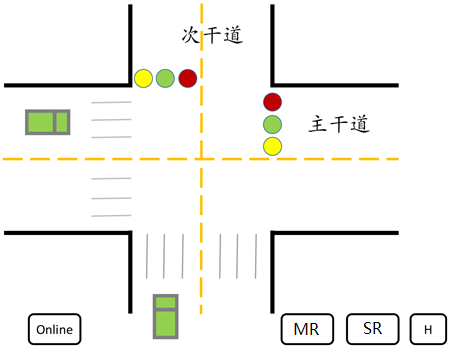


图 1-1 交通灯控制系统示意图

设计一个交通灯控制系统，具体内容及要求如下：

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（30~4），黄灯倒计时 3s；非高峰期，主干道绿灯倒计时 12s（15~4），黄灯倒计时 3s；

（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，黄灯倒计时 3s；

（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。

1.5 实验要求

（1） 根据给定的实验包，将运动码表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。

2　实验总体设计

2.1 实验初始设计思路及框架

实验要求是设计一个交通灯控制系统，具体功能包括：系统有 4 个输入信号（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），输出红灯、绿灯和黄灯的剩余时间以及主干道和次干道的红灯、绿灯和黄灯状态，使用 2 个七段数码管和 6 个 LED 灯显示；初始状态为主次干道均黄灯闪烁显示 0；紧急状态下主干道绿灯常亮显示 99;主干道通行按照绿灯或黄灯,高峰期主干道绿灯倒计时 27s，黄灯 3s；非高峰期主干道绿灯倒计时 15s，黄灯 3s；次干道通行按照绿灯或黄灯，次干道绿灯倒计时 12s，黄灯 3s；根据优先级规则控制交通灯状态转换，包括处理单个或双方通行请求、紧急状态切换和结束后回归正常流程等。

根据实验要求，通过与同学进行讨论，我的初始设计思路是将交通灯控制系统分解成五个关键模块，分别是：输入模块、逻辑模块、计数器模块、显示模块以及控制模块，每个模块负责特定的功能以确保系统的高效运行和易于维护。接下来将对初步设计思路的各模块进行详细解释。

输入模块：该模块负责接收并处理来自外部环境的信号，包括高峰期信号 H、主干道请求信号 MR、次干道请求信号 SR 和紧急控制信号 Online。通过输入模块的处理，系统可以及时获取交通路况信息并做出相应调控。

逻辑模块：逻辑模块对输入信号进行逻辑处理和判断，生成相应的控制信号以驱动交通灯的运行。该模块负责实现交通灯控制系统的核心逻辑，确保交通灯状态的准确切换和遵循交通规则。

计数器模块：在交通灯控制系统中，计时是至关重要的功能。计数器模块实现了 BCD 计数器，用于倒计时并显示每个交通灯亮灭的剩余时间。通过计数器模块，系统可以精准控制交通灯状态的变化。

显示模块：显示模块承担着展示系统状态的任务。通过两个 7 段数码管和六个 LED 灯，系统可以直观地展示交通灯的当前状态、剩余时间以及主次干道的通行情况，方便行人和车辆了解道路状况。

控制模块：控制模块是系统的核心，它将逻辑模块、计数器模块和显示模块有机地结合在一起。控制模块负责整合各个模块的功能，实现对交通灯状态的全面控制和管

理。通过控制模块的协调作用，系统可以高效稳定地运行。

2.2 实验最终设计思路及框架

在综合实验任务书要求的基础上，经过深入分析和细致思考后，我决定采用一个更加高效和模块化的设计框架以实现交通灯控制系统的功能。最终设计将由以下模块组成，并分成四个部分来更好地展现系统的功能和结构：

（1） 交通灯系统状态转移和输出模块：

输入模块：接收高峰期信号 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 等输入信号。

优先级编码器模块：对输入信号进行编码，确定不同请求间的优先级。

交通灯控制模块：根据编码结果生成相应的控制信号，控制交通灯的状态变化。

显示模块：利用两个 7 段数码管和六个 LED 灯显示交通灯状态和剩余时间。

（2） 倒计时选择模块：

计数器模块：实现 BCD 计数器功能，用于倒计时并显示每个交通灯的剩余时间。

（3） 主干道通行倒计时模块：

主干道通行指示模块：根据信号控制主干道绿灯、黄灯和红灯的状态。

倒计时模块：在高峰期和非高峰期分别倒计时主干道绿灯及黄灯时间。

（4） 次干道通行倒计时模块：

次干道通行指示模块：控制次干道绿灯和红灯的状态。

倒计时模块：倒计时次干道绿灯和黄灯时间。

通过上述模块的分工和相互串联，可以实现交通灯控制系统的各项功能。模块框架图设计如下图 2-1 所示。

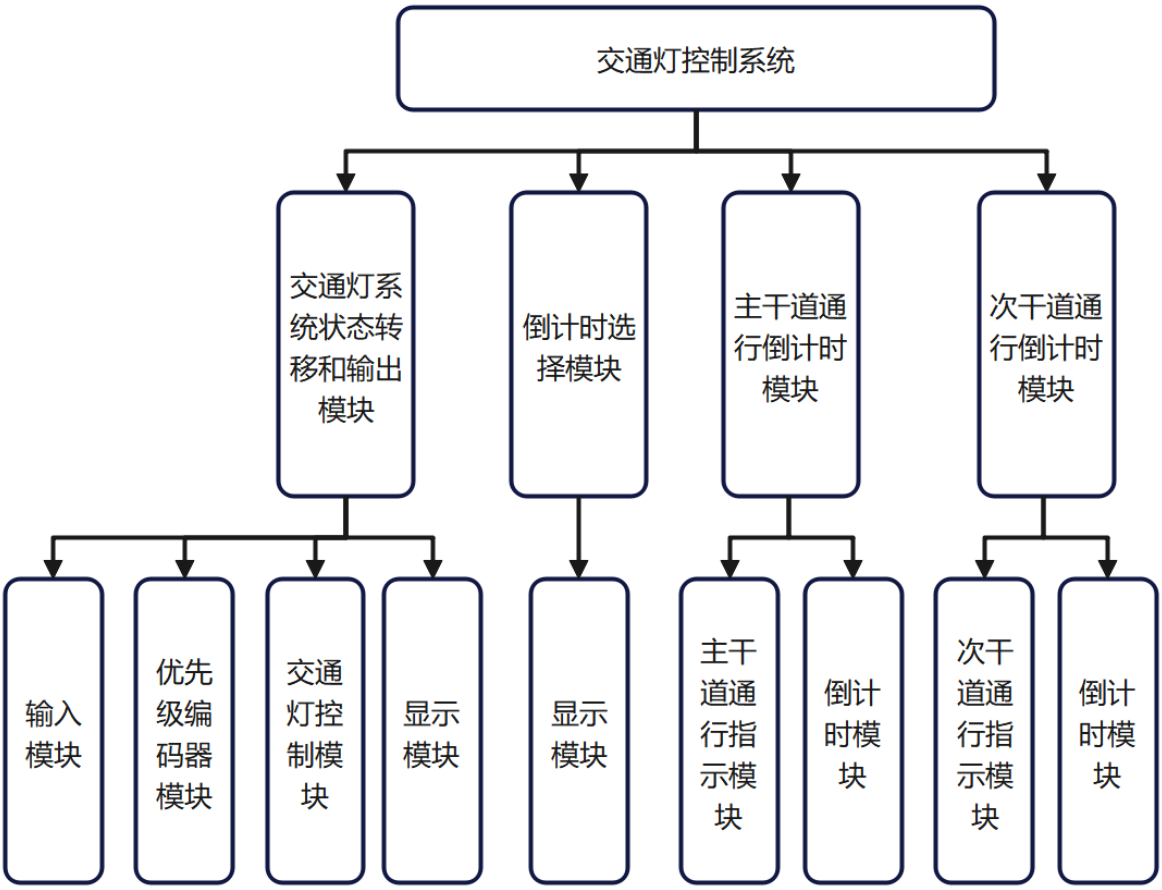


图 2-1 模块框架图设计图

在设计交通灯控制系统的最终框架中，注重了模块化和可扩展性，确保每个模块都有明确定义的接口与其他模块通信，从而实现系统功能的清晰划分和模块间的高效协作。相较于初始设计，最终设计框架更加高效和模块化，并引入了优先级编码器模块和状态机模块，这些改进确保了系统严格遵循指定的规则并正确进行状态转换。

优先级编码器模块的引入使系统能够合理分配不同请求的优先级，从而确保交通灯控制的精准性和有效性。状态机模块的实现使得系统具备了有限状态机的能力，能够更灵活地管理交通灯的状态和转换，确保系统运行符合预期。

通过优先级编码器和状态机的引入，最终设计成功简化了逻辑模块的复杂性，使系统更易于实现和调试。这种精简化的设计不仅提高了系统的可维护性，还减少了潜在的错误和故障的可能性，提高了系统的稳定性和可靠性。这样的设计框架不仅满足了实验任务书的要求，还为未来对系统功能的扩展和改进提供了良好的基础。

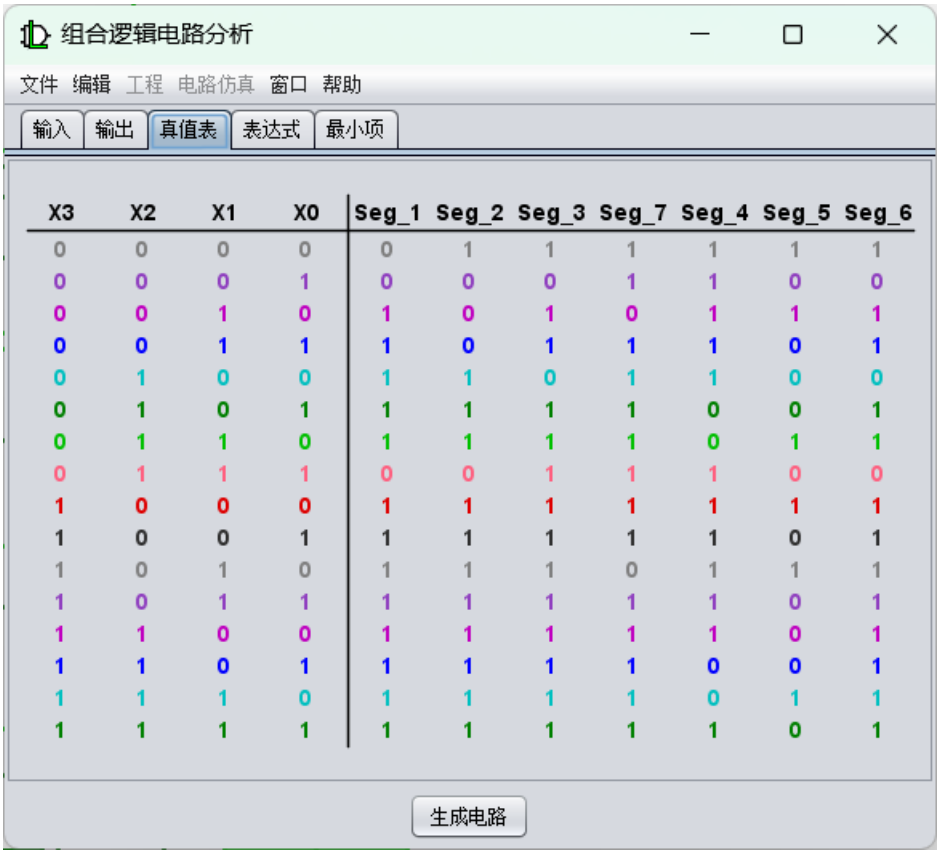
3　实验过程

3.1 7 段数码管驱动电路设计

（1）设计思路及设计过程

设计本实验的 7 段数码管驱动电路的首要任务是根据七段显示译码器组合逻辑电路指示图，在 Logisim 中实现一个能够将输入的 4 位 8421 码转换为 7 位数码管驱动信号的电路。

首先，我们需要理解七段显示译码器的工作原理。七段显示器的每个段（a，b，c，d，e，f，g）能够根据输入的编码点亮不同的组合，形成数字 0 到 9 以及一些字母。然后设计完成逻辑电路的真值表，数码管驱动真值表设计指示图如图 3-1 所示。



X3	X2	X1	X0	Seg_1	Seg_2	Seg_3	Seg_7	Seg_4	Seg_5	Seg_6
0	0	0	0	0	1	1	1	1	1	1
0	0	0	1	0	0	0	1	1	0	0
0	0	1	0	1	0	1	0	1	1	1
0	0	1	1	1	0	1	1	1	0	1
0	1	0	0	1	1	0	1	1	0	0
0	1	0	1	1	1	1	1	0	0	1
0	1	1	0	1	1	1	1	0	1	1
0	1	1	1	0	0	1	1	1	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	1	0	1
1	0	1	0	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	1	0	1
1	1	0	0	1	1	1	1	1	0	1
1	1	0	1	1	1	1	1	0	0	1
1	1	1	0	1	1	1	1	0	1	1
1	1	1	1	1	1	1	1	1	0	1

图 3-1 数码管驱动真值表设计指示图

本关的设计关键便在于数码管驱动真值表的设计，此后的操作便是对 Logisim 软件部件和指示的了解。打开 Logisim 软件，并载入实验资料包中提供的 05_TrafficLight.circ 文件。在 Logisim 中，定位到七段显示驱动电路子电路，该子电路负责将 4 位 8421 码输入转换为对应的七段数码管驱动信号。根据七段显示译码器

组合逻辑电路指示图，设置适当的逻辑门，确保每个数码管段（a 到 g）能够正确显示输入的数字。选择 Logisim 菜单中的“工程”下的“分析组合逻辑电路”，以便填写真值表并生成电路。

（2）电路图

电路图中包括从 4 位 8421 码输入到七段数码管驱动信号输出的详细连接，以下是设计的 7 段数码管驱动电路的示意图：

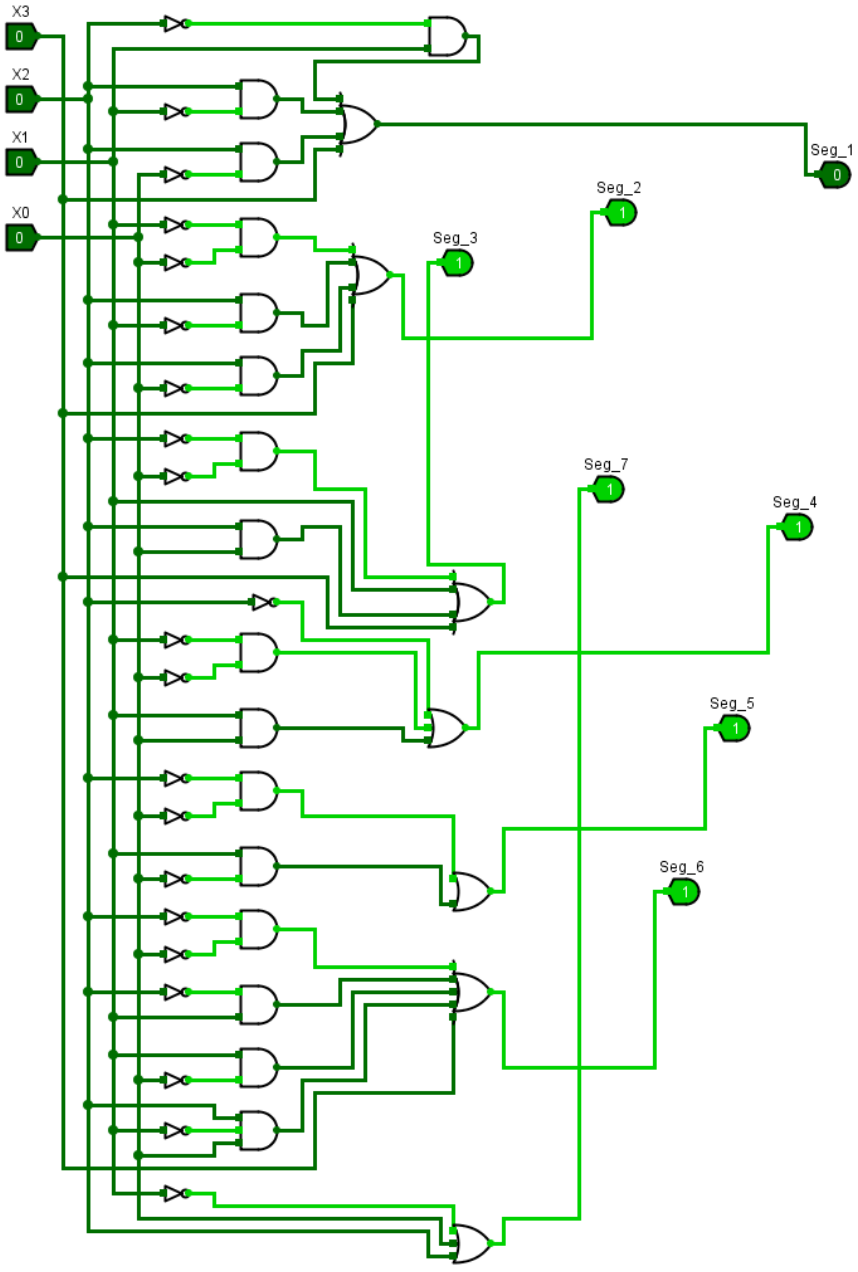


图 3-2 7 段数码管驱动电路示意图

（3）测试图

设计完成后，对设计电路进行测试且显示通过，初始默认状态运行较慢，可通过“时针滴答频率”调整测试电路运行频率，logisim 和头歌的测试图分别如下所示，部分测试电路头歌的测试集数据过多，报告篇幅有限便只会截取部分数据进行展示，后面也不再特意注明，如若需要查看整体通过情况可以在头歌平台进行进一步检查。

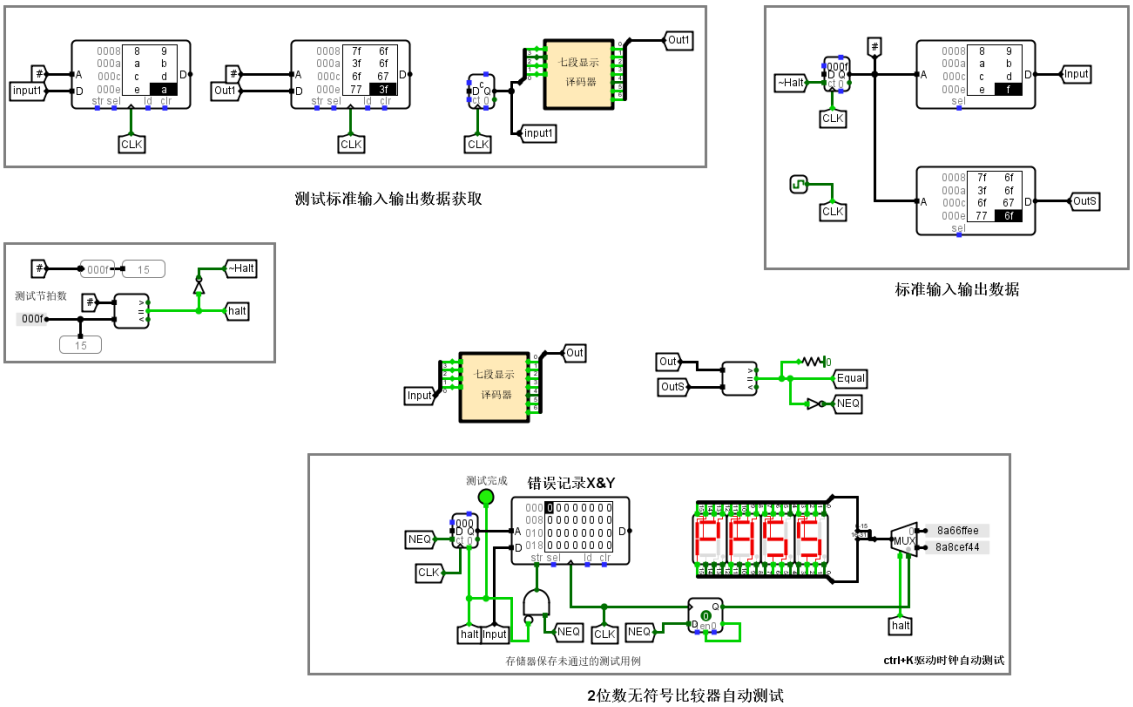
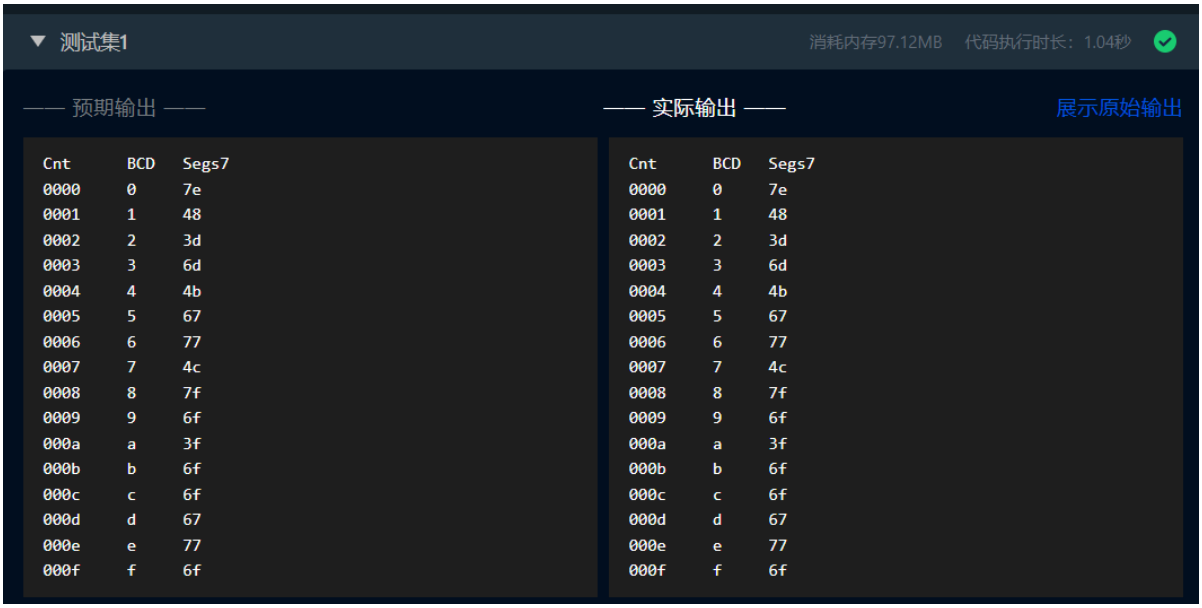


图 3-3 7 段数码管驱动电路测试图(logisim)



Cnt	BCD	Segs7
0000	0	7e
0001	1	48
0002	2	3d
0003	3	6d
0004	4	4b
0005	5	67
0006	6	77
0007	7	4c
0008	8	7f
0009	9	6f
000a	a	3f
000b	b	6f
000c	c	6f
000d	d	67
000e	e	77
000f	f	6f

图 3-4 7 段数码管驱动电路测试图(头歌)

（4）测试分析

经过测试和分析，我们确认电路功能良好，能够稳定可靠地运行并输出预期的结果，且无报错或险象，符合设计要求。这验证了我们设计的 7 段数码管驱动电路在实验目的和内容上的正确实现。

3.2 无符号比较器（4 位、8 位）

（1）设计思路及设计过程

设计实现 4 位无符号数比较器，比较 X3X2X1X0 ： Y3Y2Y1Y0，比较顺序从高位到低位，当高位大、小关系确定时则无需看低位，当高位相等时再看相邻低位的关系。而该电路有 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可以完成。在子电路“4 位无符号数比较器”中，利用“分析组合逻辑电路”，填写输出函数表达式，直接生成电路。具体 Great，Equal 以及 Less 的表达式如下：

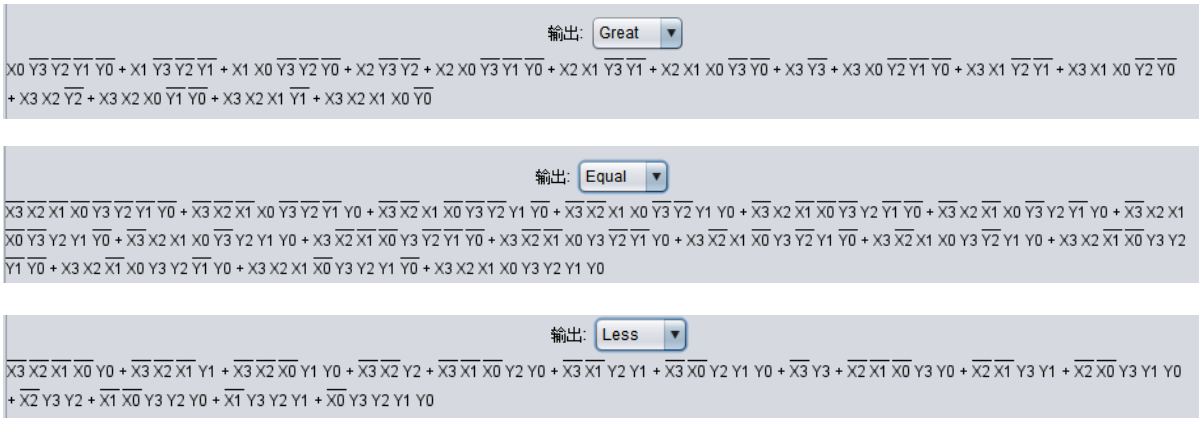


图 3-5 4 位无符号比较器逻辑表达式图

可见表达式是非常复杂的，而函数表达式与电路的复杂程度直接相关，输入上述逻辑表达式生成的电路可见下图所示。

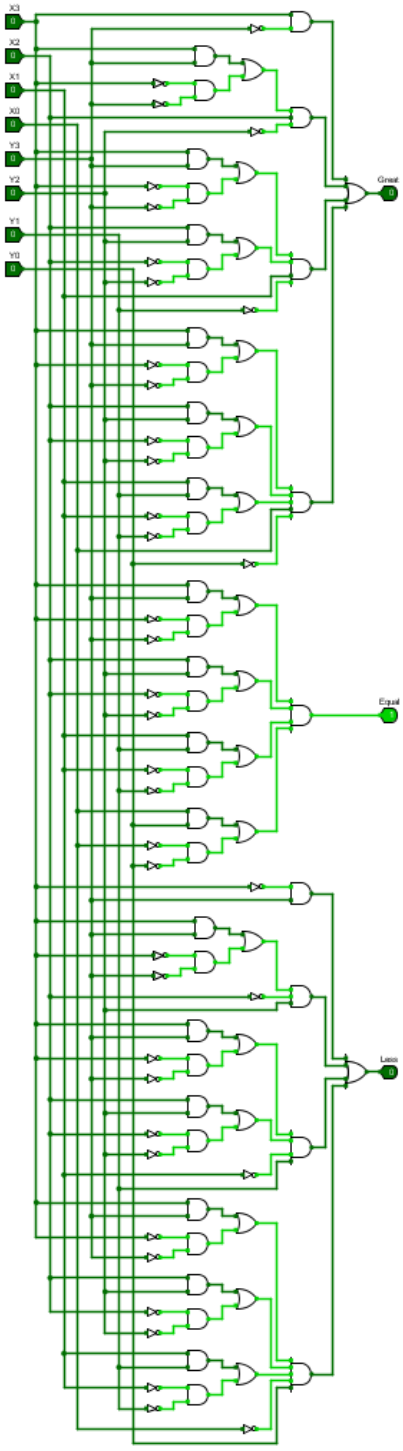


图 3-6 4 位无符号比较器电路图（ I ）

电路是十分冗杂的，因此我们需要对逻辑函数进行简化和修改，易知对于三个比较结果，已知其中任意两个，可以用或非门求得第三个。我们可以通过修改整逻辑门的布局和连接方式，以及优化电路的延迟和功耗，进一步提升了电路的性能和效率。4 位无符号比较器电路进一步修改如下图所示。

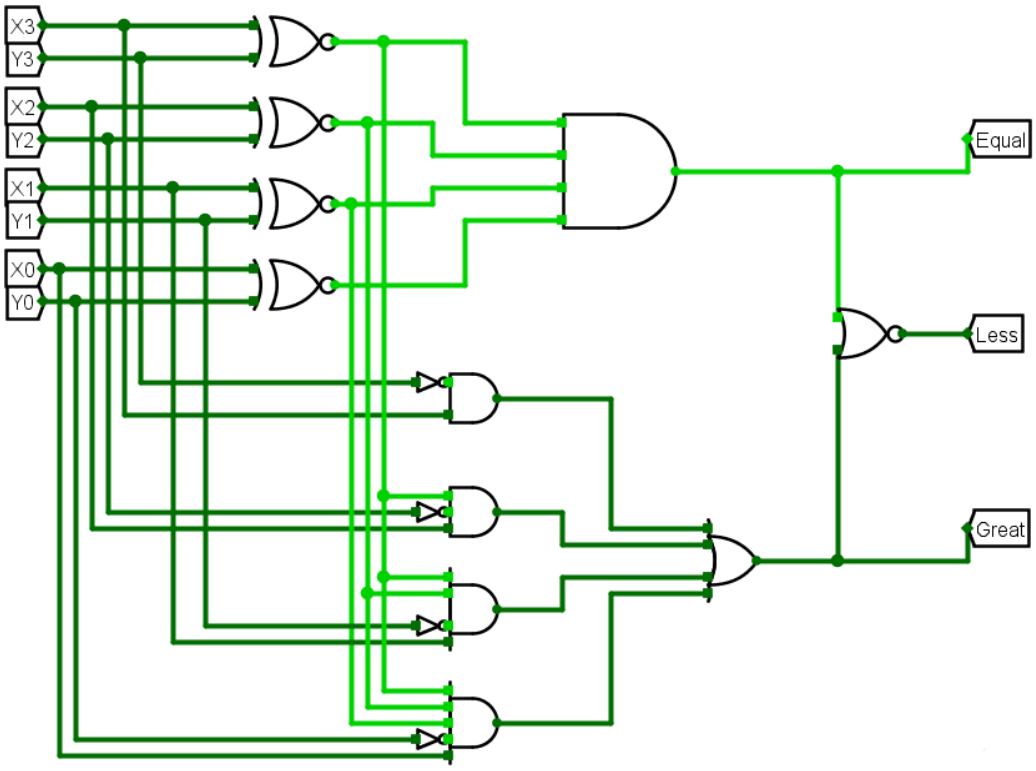


图 3-7 4 位无符号比较器电路图（II）

尽管这个电路的逻辑函数做到了最简，但是连线逻辑并没有那么直接。除了从简化逻辑函数入手，其实我们可以从真值表下手，4 位的无符号比较器，写真值表不太可能，所以可以先设计一个 1 位的无符号比较器，利用真值表自动生成电路。

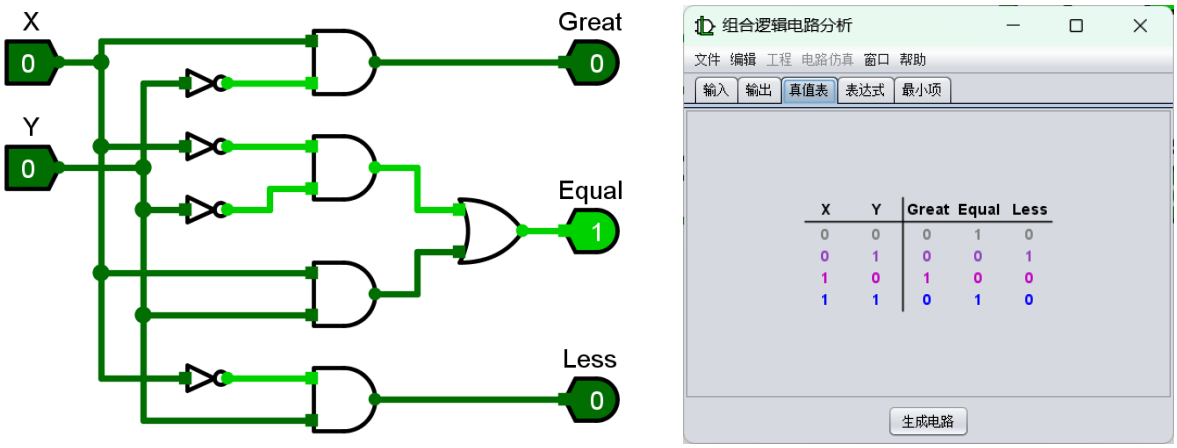


图 3-8 1 位无符号比较器电路图和真值表

然后封装 1 位无符号比较器电路，并编辑电路封装结构使得接口使用更加明晰，然后导入 4 位无符号比较器电路。综合前几步的电路特征，可以形成下图所示的电路。

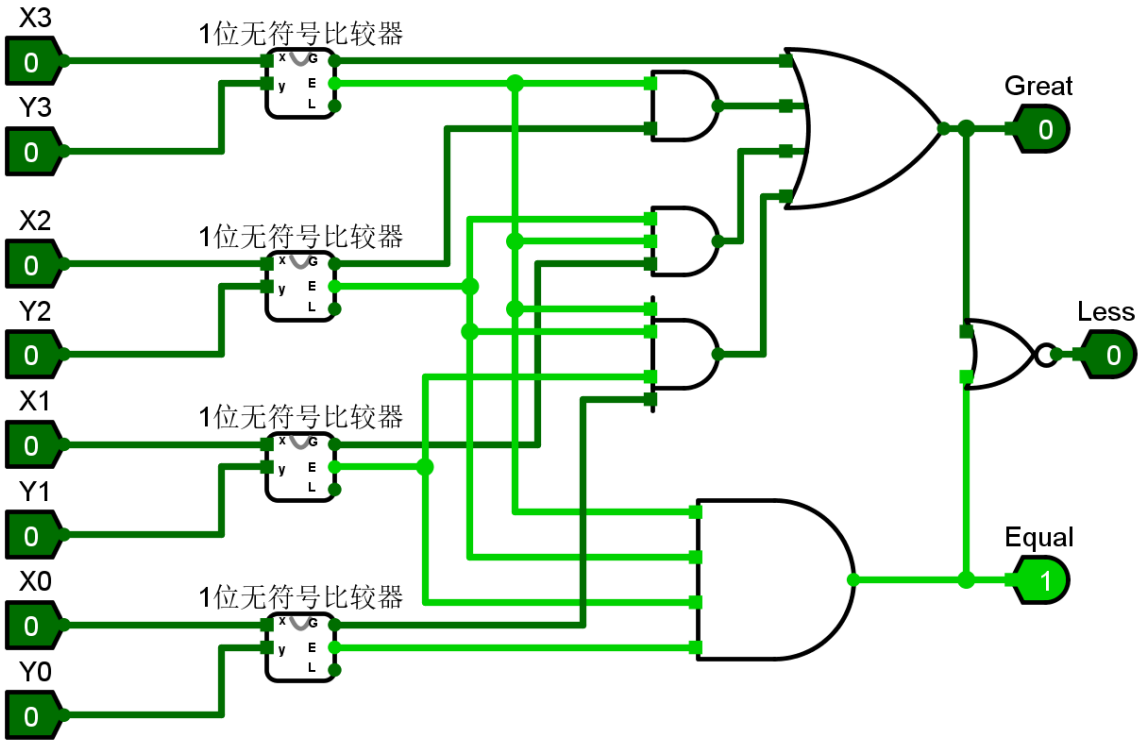


图 3-9 4 位无符号比较器电路图（Ⅲ）

而设计一个 8 位无符号数比较器的关键在于将 4 位无符号数比较器进行级联扩展。首先，我们已经设计完成了 4 位无符号数比较器，现在的任务是将其扩展为一个 8 位的比较器。在 4 位比较器的基础上，我们需要增加额外的电路以处理更高位数的输入信号。将输入的 X 和 Y 分别拆分为高 4 位和低 4 位，需要使用分线器（Splitter）将其分别输入两个 4 位比较器。同时，将两个 4 位比较器的输出进行级联，以确定整个 8 位数的大小关系。设计额外的逻辑电路，将两个 4 位比较器的输出进行合并，得出整个 8 位数的大小关系。整体的连线思路还是很清晰的，电路图将在下面进行展示。

（2）电路图

4 位无符号比较器电路的优化过程和电路图在上部分已经进行了展示，便不再赘述，以下是设计的 8 位无符号比较器电路的示意图：

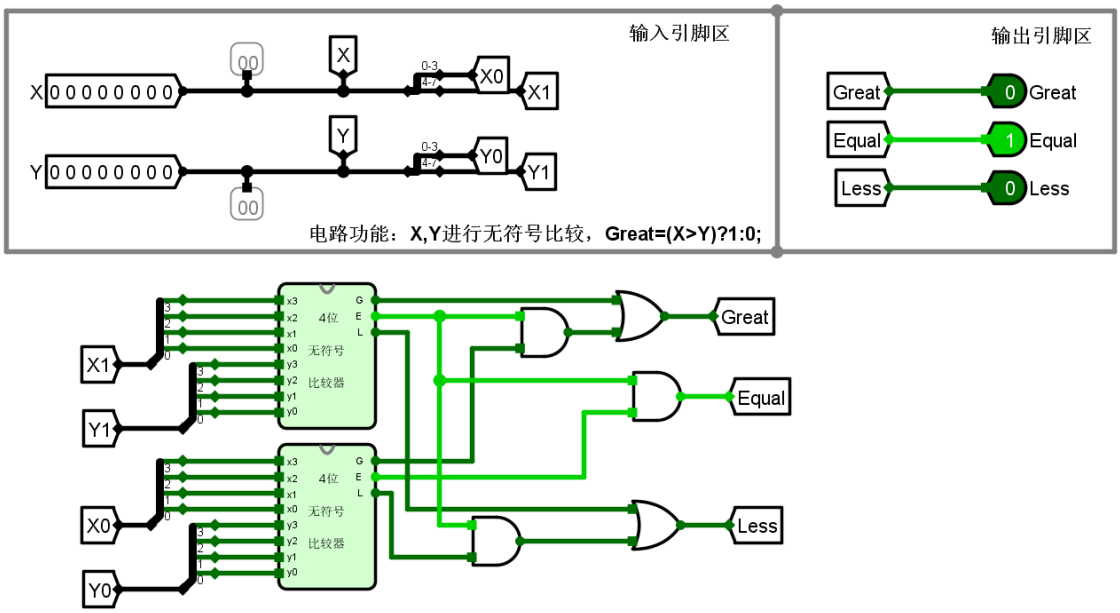


图 3-10 8 位无符号比较器电路图

（3）测试图

设计完成后，对 4 位和 8 位无符号比较器设计电路进行测试且均显示通过，8 位通过即说明 4 位也执行正确，以下仅附上 8 位的 logisim 和头歌测试图。

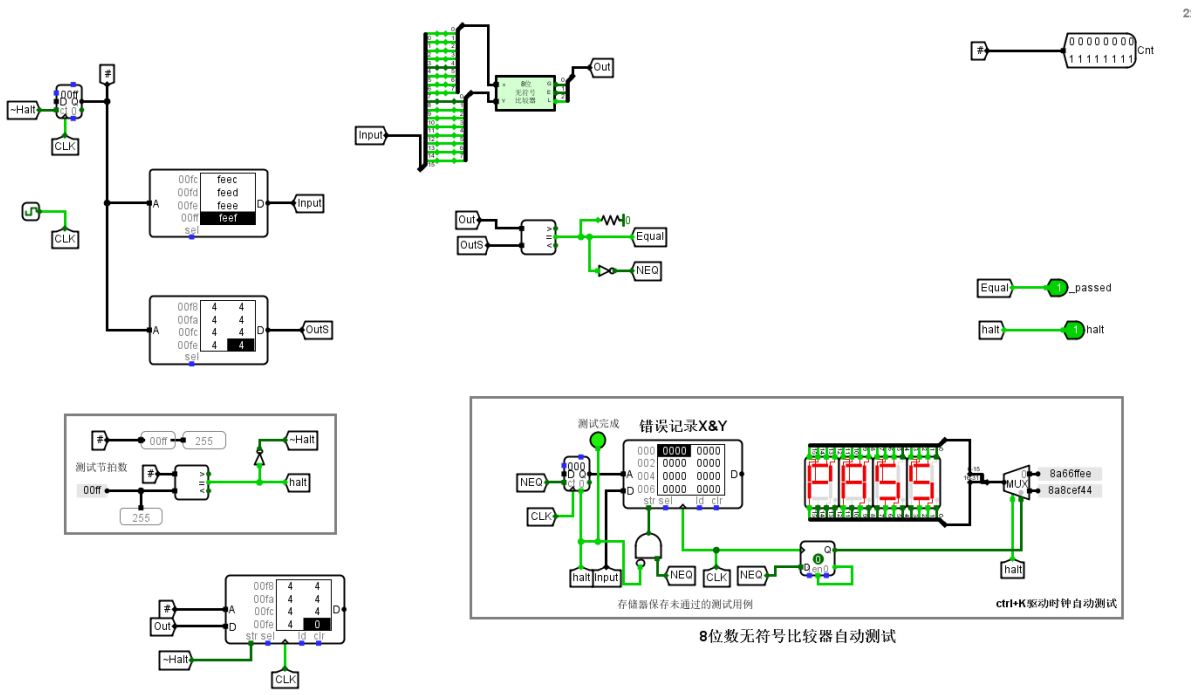


图 3-11 8 位无符号比较器测试图(logisim)

图 3-12 8 位无符号比较器测试图(头歌)

（4）测试分析

经过测试分析以及对电路性能的优化，我们确认电路功能良好，能够稳定可靠地运行并输出预期的结果，且无报错或险象，符合设计要求。这验证了我们设计的 4 位和 8 位无符号比较器电路在实验目的和内容上的正确实现。

3.3 2 选 1 选择器设计（2 位、8 位）

（1）设计思路及设计过程

2 选 1 选择器是一种基本的数字电路元件，用于在两个输入信号中选择一个输出。在本实验中，我们需要设计一个 2 选 1 选择器，其中包括 2 位和 8 位两种情况。

对于 2 位 2 选 1 选择器，我们可以利用基本的逻辑门构建，根据选择控制端的值决定输出是 X 还是 Y。首先，根据输入输出要求，设计一个选择器，其中包括两个输入端（X 和 Y）、一个选择控制端（Sel）和一个输出端（Out）。然后根据选择控制端的值，通过逻辑电路判断选择输出 X 还是 Y。

对于 8 位 2 选 1 选择器，我们可以利用多个 1 位 2 选 1 选择器构建，以实现 8 位输入的选择功能。首先，将 8 位输入信号 X 和 Y 分别拆分为单个位，然后将它们分别输入到对应的 1 位 2 选 1 选择器中。通过设置相同的选择控制端（Sel），选择器将根据控制端的值选择输出 X 还是 Y。该设计情况下电路图如下。

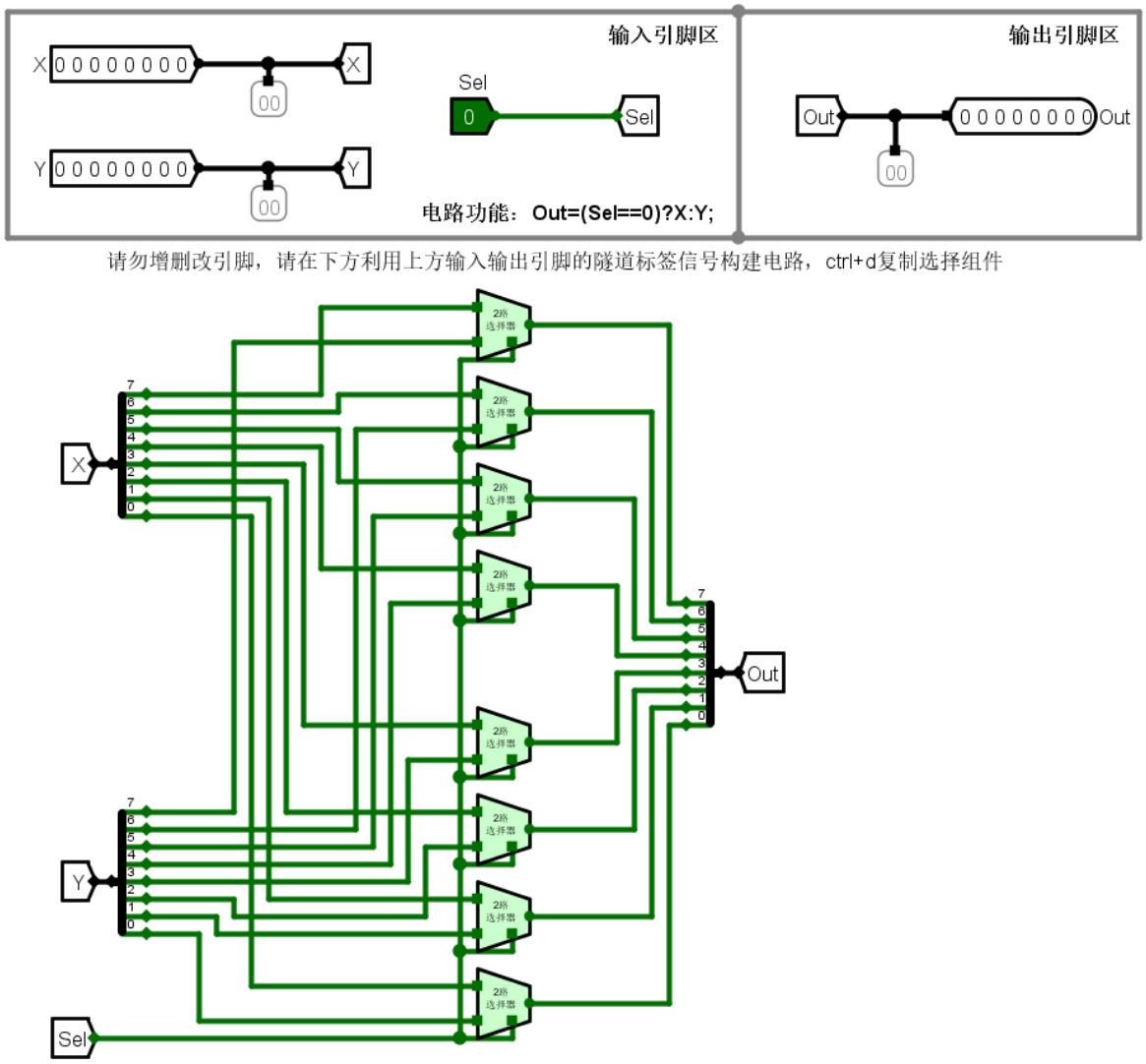
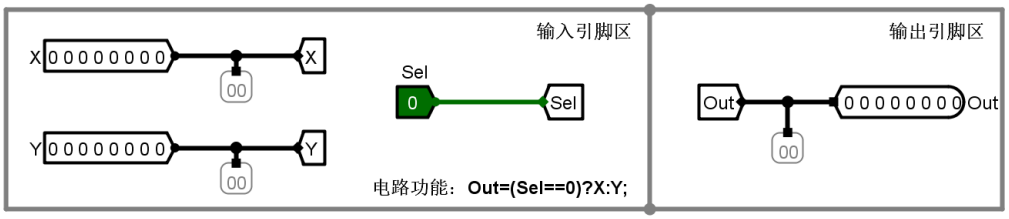


图 3-13 8 位 2 选 1 选择器的电路图（ｌ）

可见这个电路是比较复杂的，8 位则需要用到 8 个 2 位 2 选 1 选择器，但是其实我们可以进一步思考 8 位 2 选 1 选择器的逻辑和特点，2 选 1 选择器的特点是 X 和 Y 中一定会选择其中之一，而不论是 2 位还是 8 位都只是选择其中的一个，且 Sel 的值非零即一。故可以将 Sel 的位宽通过分线器与输入值进行统一，便可以直接进行相与来进行比较，再将两者的输出相或即为结果。电路图可进一步简化如下所示。



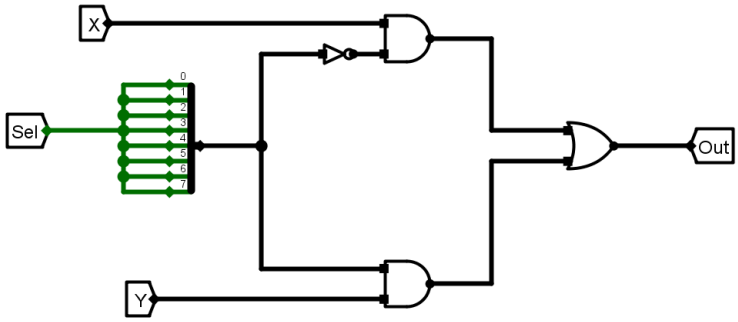
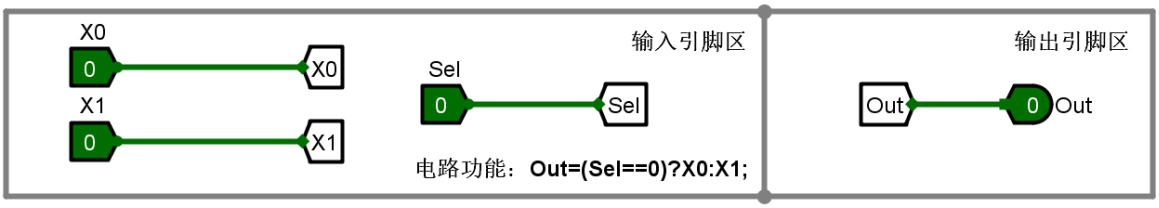


图 3-14 8 位 2 选 1 选择器的电路图（II）

（2）电路图

2 位 2 选 1 选择器的电路图将包括逻辑门的组合，以实现选择功能。8 位 2 选 1 选择器的电路图在上面已经进行展示，便不再进行赘述。以下是 2 位 2 选 1 选择器设计电路的示意图：



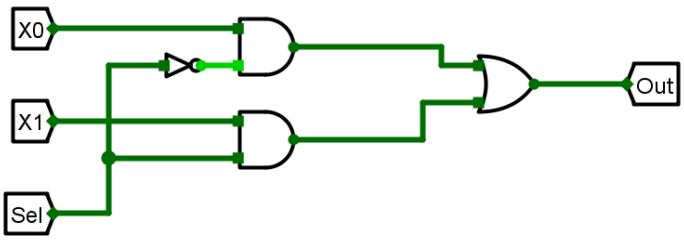


图 3-15 2 位 2 选 1 选择器的电路图

（3）测试图

设计完成后，对 2 位和 8 位 2 选 1 选择器设计电路进行测试且均显示通过，两个

比较器的测试图如下所示：

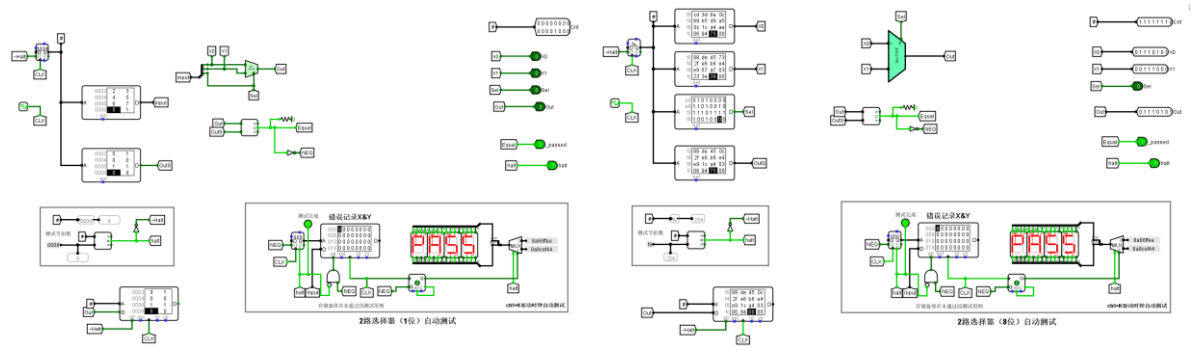


图 3-16 2 选 1 选择器测试图(logisim)

测试集1　消耗内存84.9MB　代码执行时长：0.96秒

预期输出　实际输出　展示原始输出

Cnt	X0	X1	Sel	Out
0000	0	0	0	0
0001	1	0	0	1
0002	0	1	0	0
0003	1	1	0	1
0004	0	0	1	0
0005	1	0	1	0
0006	0	1	1	1
0007	1	1	1	1
0008	0	0	0	0

图 3-17 2 位 2 选 1 选择器测试图(头歌)

测试集1　消耗内存87.23MB　代码执行时长：1.33秒

预期输出　实际输出　展示原始输出

Cnt	X0	X1	Sel	Out
00	06	07	1	07
01	d6	a4	0	d6
02	e2	e2	1	e2
03	ef	9b	1	9b
04	f0	b6	1	b6
05	76	12	1	12
06	21	bb	1	bb
07	fb	1f	1	1f
08	ca	24	1	24

图 3-18 8 位 2 选 1 选择器测试图(头歌)

（4）测试分析

经过测试和分析，我们确认电路功能良好，能够稳定可靠地运行并输出预期的结果，且无报错或险象，符合设计要求。这验证了我们设计的 2 位和 8 位 2 选 1 选择器电路在实验目的和内容上的正确实现。

3.4 十进制可逆计数器

（1）设计思路及设计过程

设计十进制可逆计数器的关键在于构建合适的状态图和状态转换逻辑，以及输出函数。首先，十进制可逆计数器设计采用了 4 个 D 触发器以控制状态。每个状态由四位表示，接收一个四位二进制数 y3y2y1y0（范围为 0000 到 1001），其中，正逆计数由控制变量 Mode 决定：当 Mode=1 时为逆向计数，Mode=0 时为正向计数。而输出函数 Cout 输出为 1 的条件为：在正向计数时，当前状态为 1001（9），在反向计数时，当前状态为 0000（0）。该电路的激励函数和输出函数的真值表如下图所示。

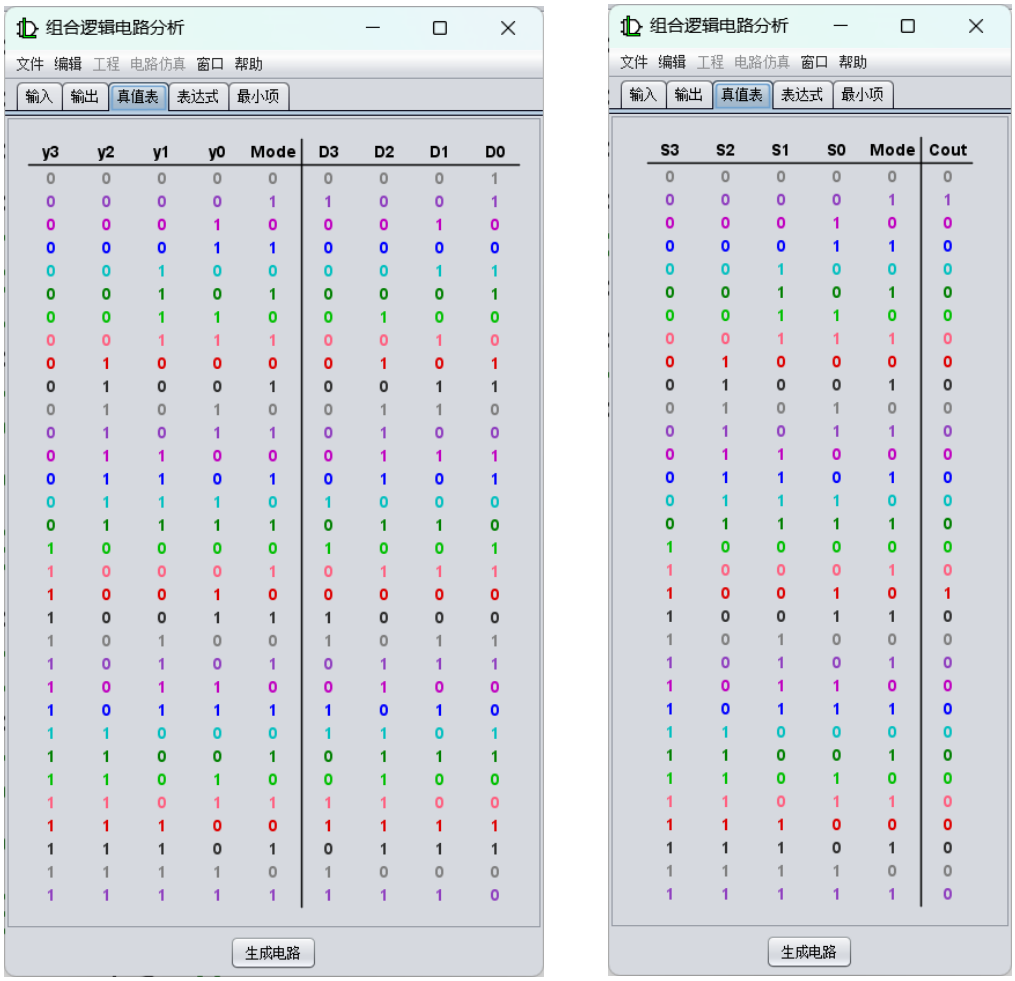


图 3-19 模十可逆计数器激励(左)和输出(右)函数真值表

由此根据状态集合和状态转移条件，可以确定这是一个 Mealy 型电路，因为输出信号 Cout（进位/借位）是与状态和输入信号（包括当前状态 y 和 Mode）有关的。Mealy 型电路的输出取决于输入信号以及当前状态，而不仅仅取决于当前状态，这与 Moore 型电路有所不同，后者的输出仅仅取决于当前状态。可以绘制出状态图和状态表如下，该

电路实现了无挂起（可自启动）且无错误输出的功能。

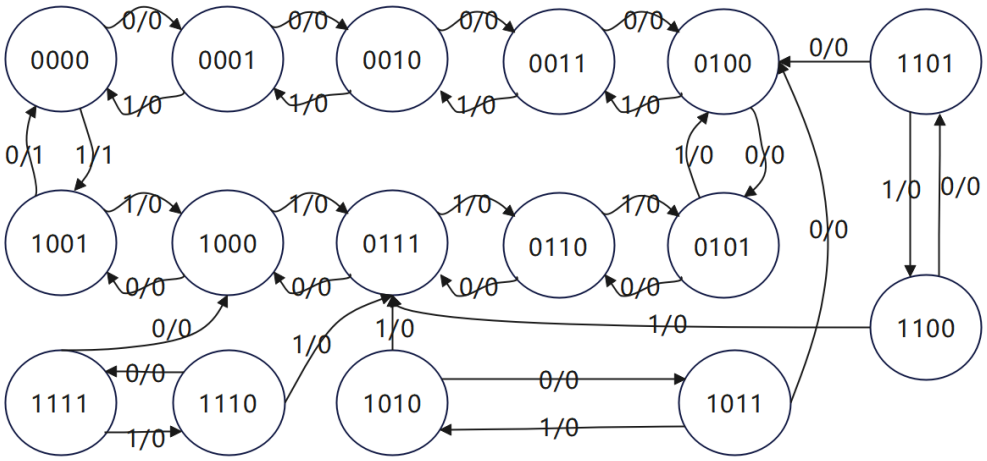


图 3-20 模十可逆计数器状态图

表 3-1 模十可逆计数器状态表

现态 y3y2y1y0	次态 $y3^{n+1}y2^{n+1}y1^{n+1}y0^{n+1}$/输出 Cout	
	Mode=0	Mode=1
0000	0001/0	1001/1
0001	0010/0	0000/0
0010	0011/0	0001/0
0011	0100/0	0010/0
0100	0101/0	0011/0
0101	0110/0	0100/0
0110	0111/0	0101/0
0111	1000/0	0110/0
1000	1001/1	0111/0
1001	0000/0	1000/0
1010	1011/0	0111/0
1011	0100/0	1010/0
1100	1101/0	0100/0
1101	0100/0	1100/0
1110	1111/0	0111/0
1111	1000/0	1110/0

根据实验要求，我们需要设计模十可逆计数器的激励函数和输出函数。激励函数确定了每个状态下各个 D 触发器的激励值，而输出函数则负责生成进位/借位信号。激

励函数设计过程中，需要考虑正向计数和反向计数两种模式下的状态转换规则，确保在每次时钟脉冲到来时，状态能够正确地转移至下一个状态。同时，输出函数的设计需要根据当前状态和计数模式来确定是否产生进位/借位信号。

激励函数输出为 D3D2D1D0，故其表达式分别如下：

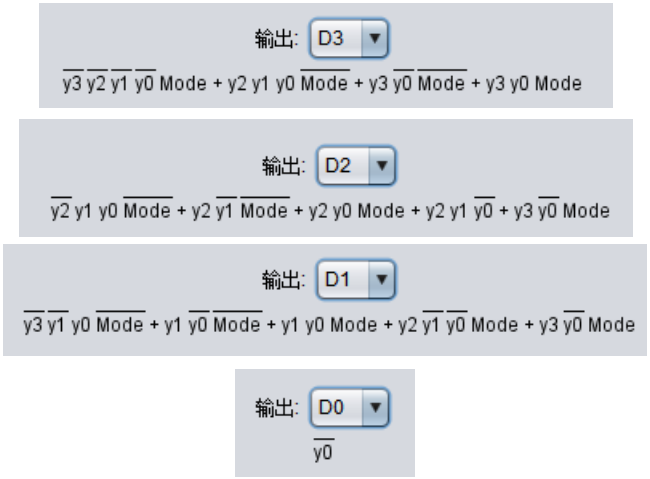


图 3-21 模十可逆计数器激励函数表达式

输出函数输出为 Cout，其表达式如下：

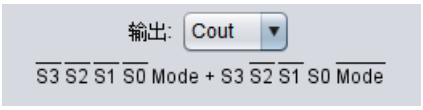


图 3-22 模十可逆计数器输出函数表达式

综合激励函数和输出函数的设计，可以利用 D 触发器构建十进制可逆计数器电路。在电路框架设计中，需要考虑时钟信号、使能信号、计数模式选择信号以及预置控制信号等输入，以及计数器的输出信号和进位/借位输出信号。同时需要注意触发器的状态，如果 CLK 触发了一个上升沿，则状态变化，而**本地 logisim 和实验测试平台的并不一致，需要根据实际测试电路对触发器的上升/下降沿进行调整**，否则可能存在无法评测通过的情况。

（2）电路图

电路图包括了 D 触发器的连接方式以及激励函数和输出函数的逻辑电路设计。其中，D 触发器的输入由激励函数确定，而输出则由输出函数决定。整个电路以时钟信号驱动，根据使能信号和计数模式选择信号进行计数。同时，这里有个细节可以注意到在 Din 端使用了三态门(Controlled Buffer)进行异步置位控制。以下是设计的模十可逆计数器激励函数，输出函数以及综合电路的示意图：

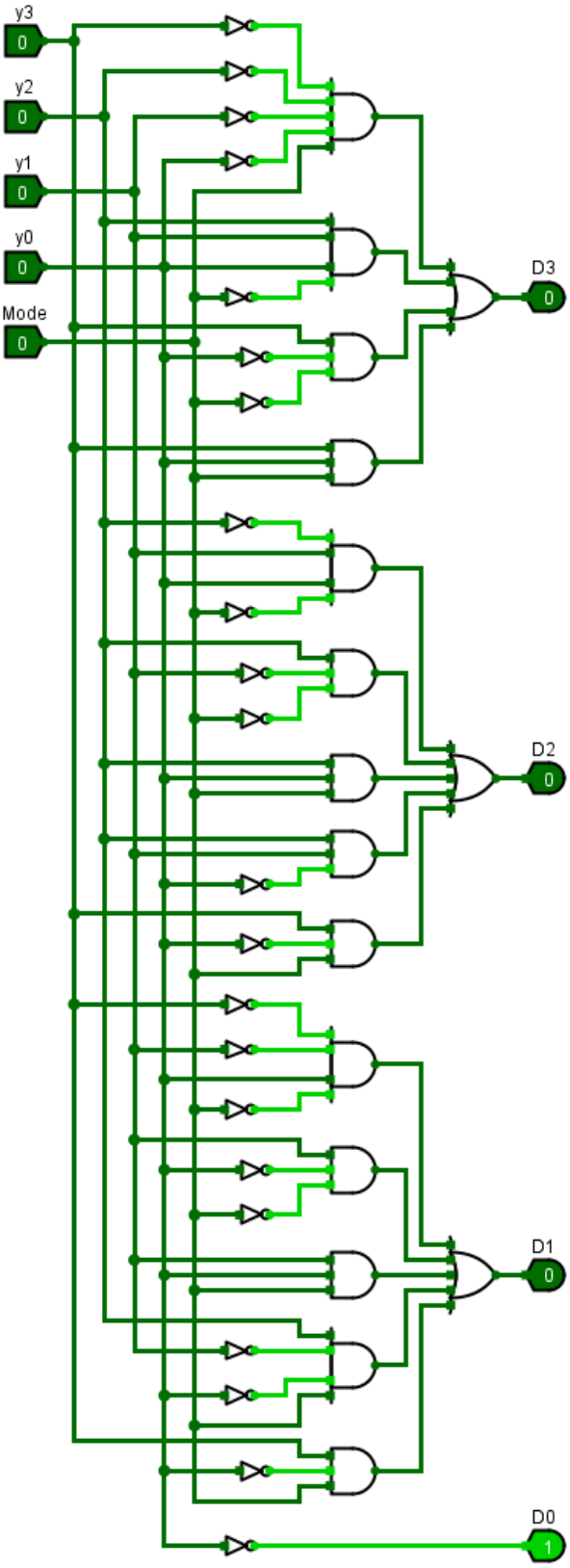


图 3-23 模十可逆计数器激励函数电路图

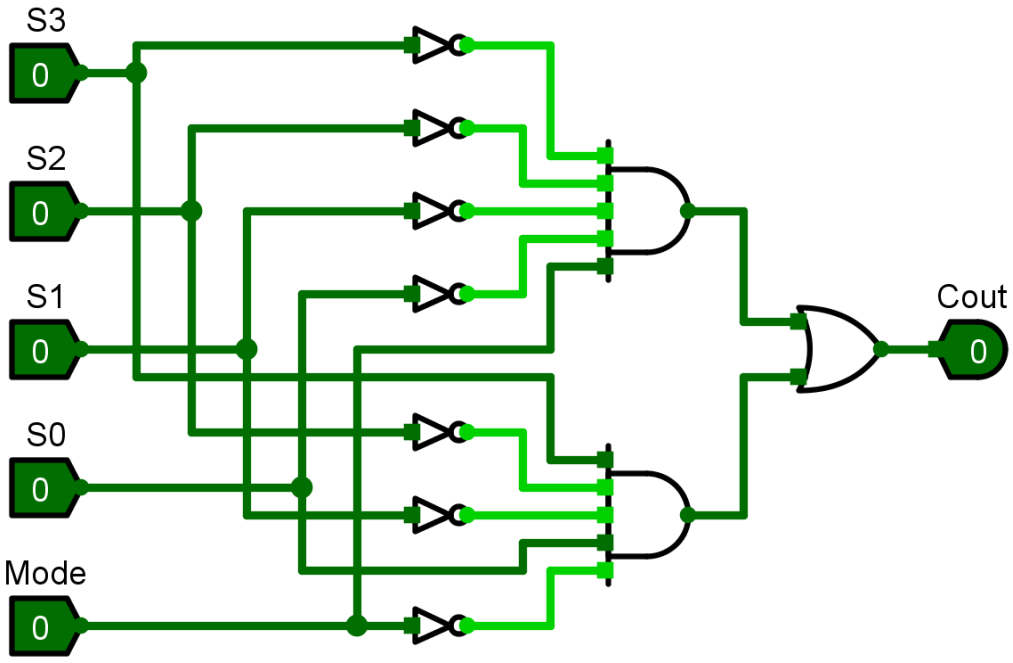


图 3-24 模十可逆计数器输出函数电路图

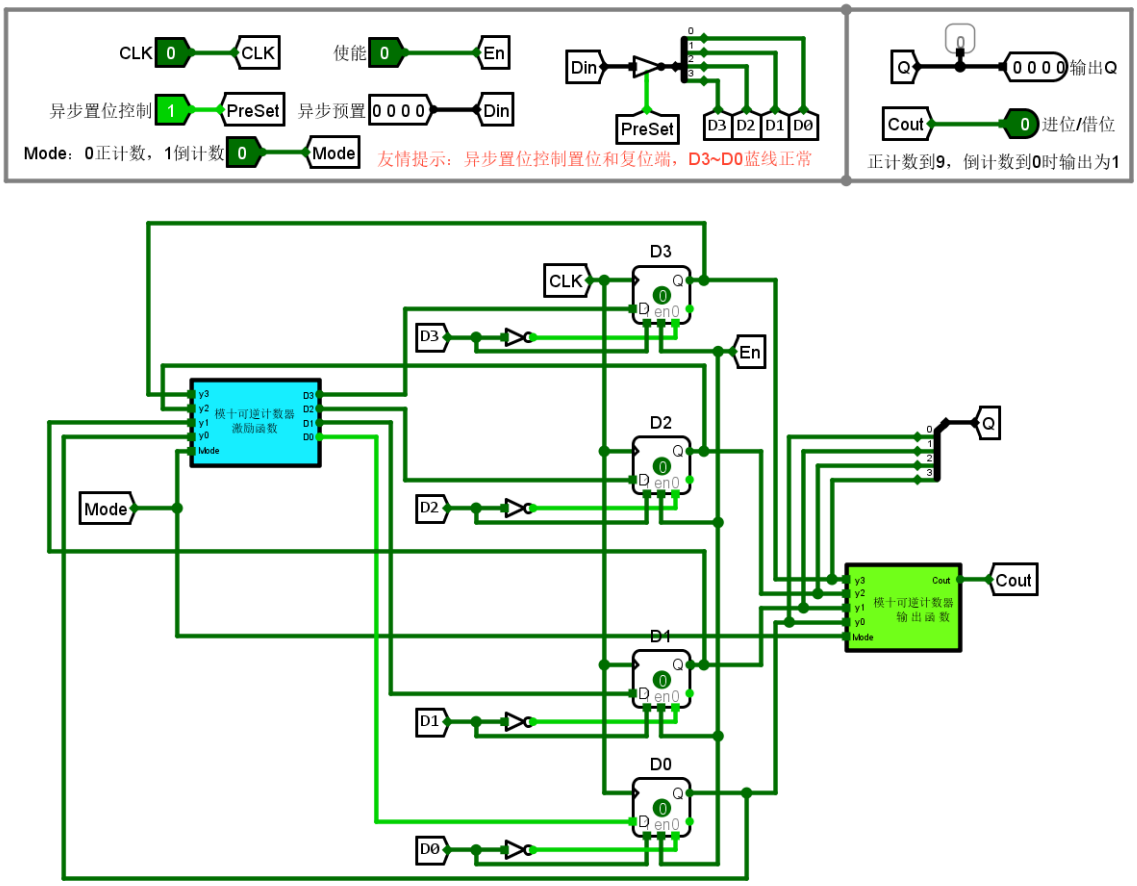


图 3-25 模十可逆计数器电路图

（3）测试图

设计完成后，对模十可逆计数器激励函数，输出函数以及综合设计电路进行测试且均显示通过，测试图如下：

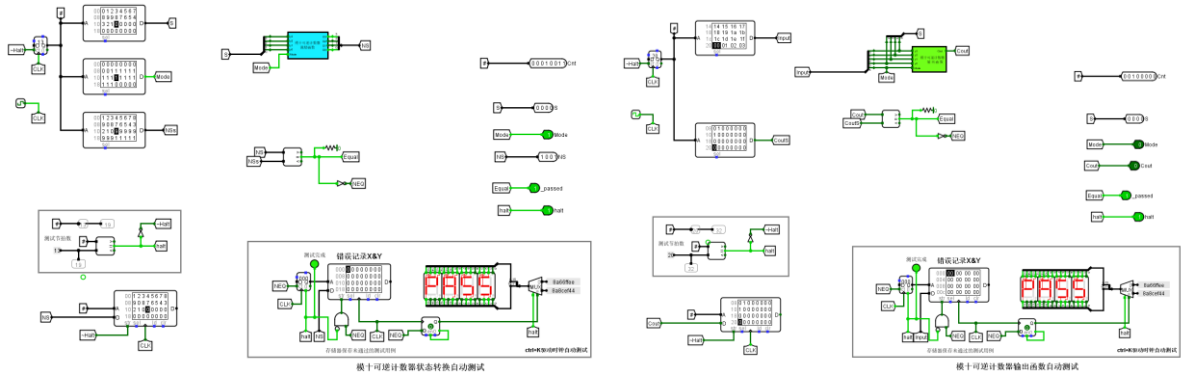


图 3-26 7 模十可逆计数器激励和输出函数测试图(logisim)

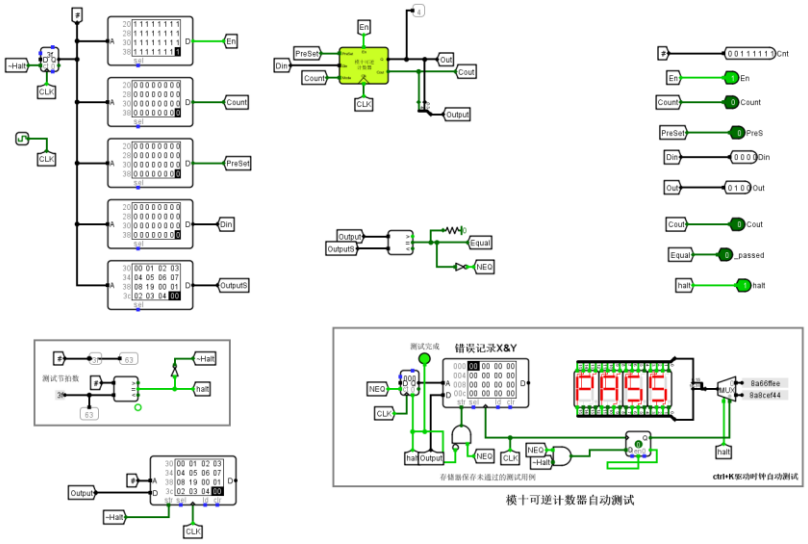


图 3-27 模十可逆计数器测试图(logisim)

测试集1　　消耗内存88.82MB　代码执行时长：1.02秒

预期输出　　实际输出　　展示原始输出

Cnt	En	Count	PreS	Din	Out	Cout	Output
00	0	1	1	8	8	0	08
01	0	1	1	9	9	0	09
02	0	1	1	6	6	0	06
03	0	1	1	7	7	0	07
04	0	1	1	5	5	0	05
05	0	1	1	4	4	0	04
06	0	1	1	3	3	0	03
07	0	1	1	2	2	0	02
08	0	1	1	0	0	1	10

Cnt	En	Count	PreS	Din	Out	Cout	Output
00	0	1	1	8	8	0	08
01	0	1	1	9	9	0	09
02	0	1	1	6	6	0	06
03	0	1	1	7	7	0	07
04	0	1	1	5	5	0	05
05	0	1	1	4	4	0	04
06	0	1	1	3	3	0	03
07	0	1	1	2	2	0	02
08	0	1	1	0	0	1	10

图 3-28 模十可逆计数器测试图(头歌)

（4）测试分析

经过测试和分析，我们确认电路功能良好，能够稳定可靠地运行并输出预期的结果，且无报错或险象，符合设计要求。这验证了我们设计的模十可逆计数器在实验目的和内容上的正确实现。

3.5 两位十进制可逆计数器

（1）设计思路及设计过程

为了实现两位十进制可逆计数器，我们将两个模十可逆计数器进行级联连接。具体来说，我们将第一个计数器的进位/借位输出连接到第二个计数器的预置输入端。这样，当第一个计数器计数到 9 或反向计数到 0 时，会触发进位/借位信号，从而将第二个计数器的初值设定为 1 或者 0，实现两位十进制的计数。同时我们需要实现：正向计数功能，确认在 Mode=0 时，两位十进制可逆计数器能够按照正向顺序正确计数；反向计数功能，确认在 Mode=1 时，两位十进制可逆计数器能够按照反向顺序正确计数；预置功能，验证预置功能是否正常，即在 PreSet 信号为 1 时，能否正确预置计数器的值；进位/借位输出，验证进位/借位输出是否正确，即在正向计数到 9 或反向计数到 0 时，Cout 信号是否正确输出为 1 等。

而对于时序电路，我们可以选择采用异步或者同步时序电路来进行设计。这两种设计方法在时序控制方面有所不同，下面将对它们进行更详细的介绍，并说明如何设计这两种类型的电路。

对于**同步时序电路**，所有触发器都在统一的时钟信号的控制下进行更新。在设计两位十进制可逆计数器时，我们可以采用同步时序设计的方法，确保计数器的状态变化是同步的，通过使能端 En 来对十位模十可逆计数器的工作情况进行控制，减少了电路中可能出现的时序逻辑问题。

对于**异步时序电路**，各个触发器的更新不受统一的时钟信号控制，而是根据特定的输入条件进行更新。在设计两位十进制可逆计数器时，异步时序电路设计方法也可以实现，将个位的输出直接与十位的时钟端相连，但是要注意触发器为上升沿触发器，所以需要将 Cout 信号通过非门进行转换才可以正确对十位的模十可逆计数器进行操作。

（2）电路图

电路图中包括同步时序逻辑电路设计和异步时序电路设计两种情况，以下是设计的两位十进制可逆计数器电路的示意图：

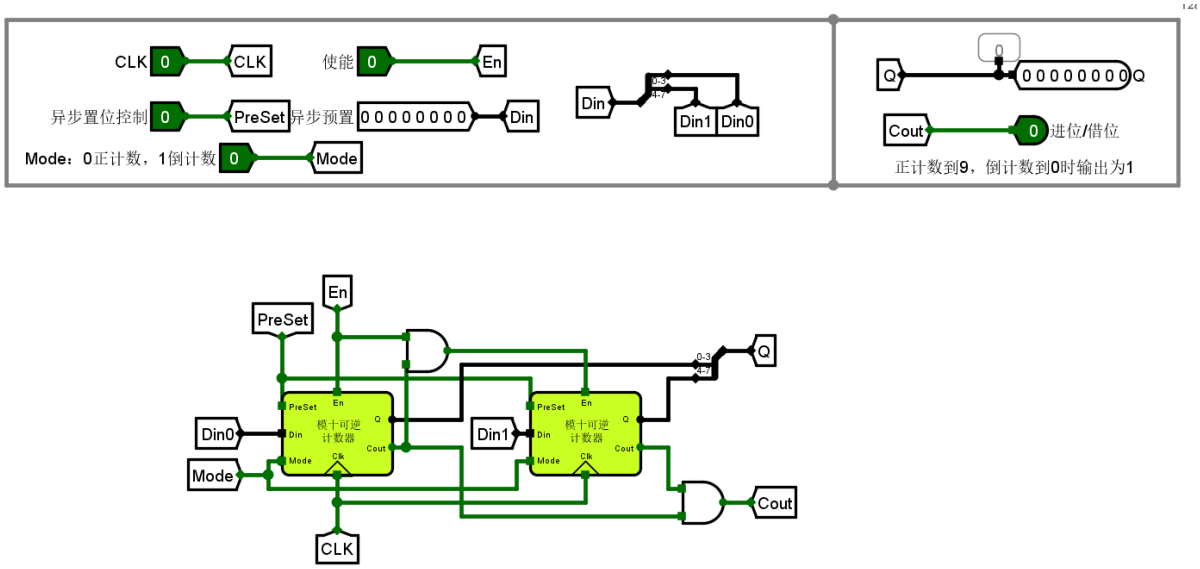


图 3-29 两位十进制可逆计数器电路示意图（同步时序）

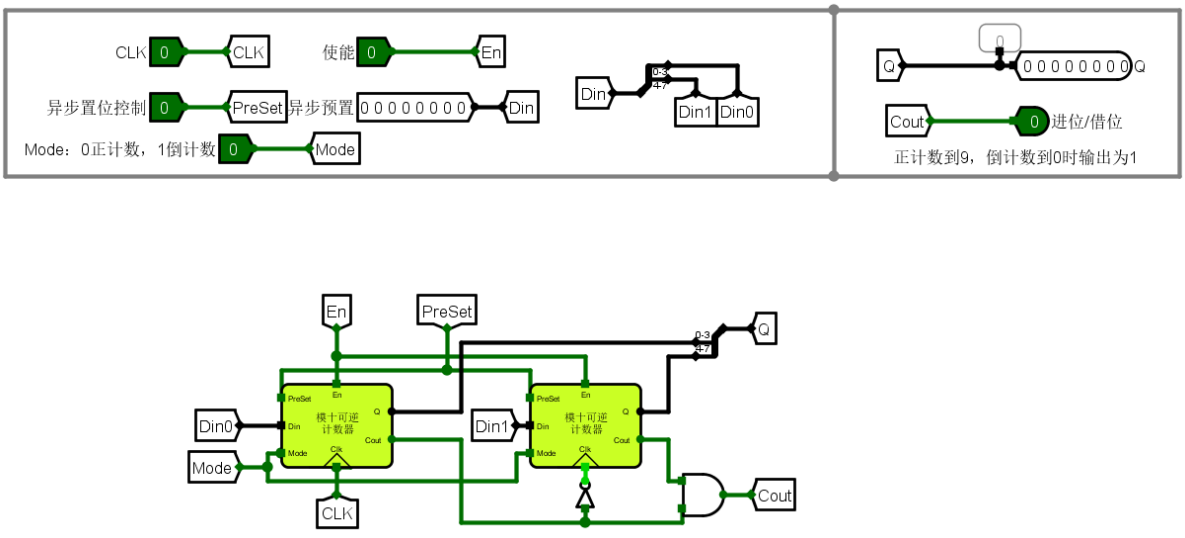


图 3-30 两位十进制可逆计数器电路示意图（异步时序）

（3）测试图

设计完成后，对设计电路进行测试且显示通过，测试图如下：

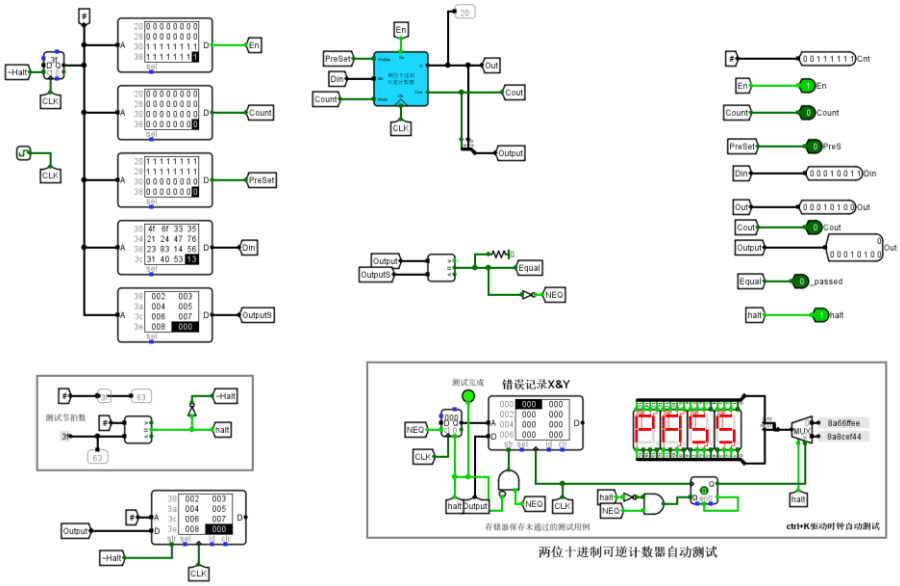


图 3-31 两位十进制可逆计数器电路测试图(logisim)

测试集1　　消耗内存89.82MB　代码执行时长：0.96秒

预期输出

Cnt	En	Count	PreS	Din	Out	Cout	Out2
00	0	1	1	4f	4f	0	04f
01	0	1	1	6f	6f	0	06f
02	0	1	1	22	22	0	022
03	0	1	1	35	35	0	035
04	0	1	1	21	21	0	021
05	0	1	1	34	34	0	034
06	0	1	1	47	47	0	047
07	0	1	1	76	76	0	076
08	0	1	1	23	23	0	023

实际输出　展示原始输出

Cnt	En	Count	PreS	Din	Out	Cout	Out2
00	0	1	1	4f	4f	0	04f
01	0	1	1	6f	6f	0	06f
02	0	1	1	22	22	0	022
03	0	1	1	35	35	0	035
04	0	1	1	21	21	0	021
05	0	1	1	34	34	0	034
06	0	1	1	47	47	0	047
07	0	1	1	76	76	0	076
08	0	1	1	23	23	0	023

图 3-32 两位十进制可逆计数器电路测试图(头歌)

（4）测试分析

经过测试和分析，我们确认电路功能良好，能够稳定可靠地运行并输出预期的结果，且无报错或险象，符合设计要求。这验证了我们设计的两位十进制可逆计数器电路在实验目的和内容上的正确实现。

3.6 交通灯状态机

（1）设计思路及设计过程

设计交通灯控制器的状态机是为了实现交通灯在不同情景下的状态切换，从而实现交通流的控制。根据实验要求，我们需要设计一个具有 7 个状态的状态机，每个状态对应着不同的交通灯状态及其倒计时。

首先，根据题目给出的要求，我们定义了 7 个状态，并为每个状态分配了状态编号、状态描述和状态编码，以及相应的说明。其次，在设计状态转移时，我们考虑了交通灯在不同输入信号下的状态切换规则，如高峰期信号、主干道和次干道的通行请求信号、紧急状况信号等。根据这些规则，我们确定了状态之间的转移关系。其状态转移的真值表如下所示。

输入 （填1或0，不填为无关项x）											输出 （只填写为1的情况）			
y2	y1	y0	H	MR	SR	Online	T4	T3	T2	T1	N2	N1	N0	
0	0	0				1					1	1	0	
0	0	0		0	0	0					0	0	0	
0	0	0	0	1		0					0	0	1	
0	0	0	1	1		0					0	1	0	初始状态（主次干道黄灯）
0	0	0		0	1	0					1	0	0	
0	0	1				1					1	1	0	主干道绿灯（非高峰期）
0	0	1				0				1	0	1	1	
0	0	1				0				0	0	0	1	
0	1	0				1					1	1	0	
0	1	0				0				1	0	1	1	高峰期主干道绿灯
0	1	0				0				0	0	1	0	
0	1	1				1					1	1	0	
0	1	1				0			0		0	1	1	主干道黄灯
0	1	1		0	0	0			1		0	0	0	
0	1	1	0	1	0	0			1		0	0	1	
0	1	1	1	1	0	0			1		0	1	0	
0	1	1			1	0			1		1	0	0	
1	0	0						1			1	0	1	
1	0	0						0			1	0	0	次干道绿灯
1	0	1					0				1	0	1	
1	0	1				1	1				1	1	0	次干道黄灯
1	0	1		0	0	0	1				0	0	0	
1	0	1	0	1		0	1				0	0	1	
1	0	1	1	1		0	1				0	1	0	
1	0	1		0	1	0	1				1	0	0	
1	1	0				1					1	1	0	
1	1	0	0			0					0	0	1	紧急情况
1	1	0	1			0					0	1	0	

图 3-33 状态转移真值表示意图

然后激励函数用于确定下一个状态的逻辑条件。我们根据状态转移关系，设计了相应的激励函数表达式，以实现状态的自动切换。该过程利用已有程序可以自动生成相关激励信号表达式，将激励函数表达式复制到 Logisim 中，即可自动生成激励函数电路。具体激励函数表达式如下：

输出: D2

$\overline{y2}$ Online + $\overline{y1}$ Online + $\overline{y0}$ Online + $\overline{y1}$ $\overline{y0}$ $\overline{MR}$ SR + $\overline{y2}$ y1 y0 SR T2 + y2 $\overline{y1}$ $\overline{y0}$ + y2 $\overline{y1}$ $\overline{T4}$ + y2 $\overline{y1}$ $\overline{MR}$ SR

输出: D1

$\overline{y2}$ Online + $\overline{y2}$ $\overline{y0}$ H MR + $\overline{y2}$ $\overline{y1}$ y0 T1 + $\overline{y1}$ y0 Online T4 + $\overline{y2}$ y1 $\overline{y0}$ + $\overline{y2}$ y1 $\overline{T2}$ + y1 $\overline{y0}$ Online + y1 $\overline{y0}$ H + $\overline{y2}$ y1 H MR $\overline{SR}$ + y2 $\overline{y1}$ y0 H MR T4

输出: D0

$\overline{y2}$ $\overline{y1}$ $\overline{H}$ MR $\overline{Online}$ + $\overline{y2}$ y1 y0 $\overline{Online}$ + y2 y0 $\overline{Online}$ $\overline{T2}$ + $\overline{y2}$ y0 $\overline{H}$ MR $\overline{SR}$ $\overline{Online}$ + $\overline{y1}$ y0 $\overline{H}$ MR $\overline{Online}$ + $\overline{y2}$ y1 $\overline{y0}$ $\overline{Online}$ T1 + y2 $\overline{y1}$ $\overline{y0}$ T3 + y2 $\overline{y1}$ y0 $\overline{T4}$ + y2 y1 $\overline{y0}$ $\overline{H}$ $\overline{Online}$

图 3-34 交通灯状态机激励函数表达式

（2）电路图

以下是设计的交通灯状态机电路的示意图：

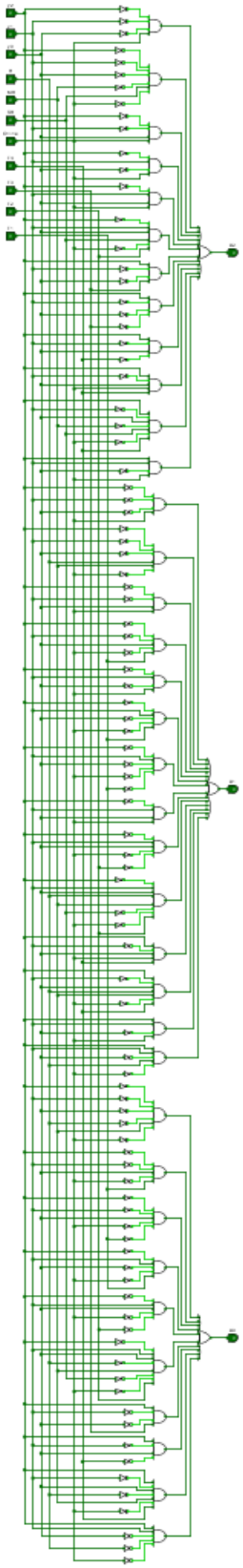


图 3-35 交通灯状态机电路示意图

（3）测试图

设计完成后，对设计电路进行测试且显示通过，测试图如下：

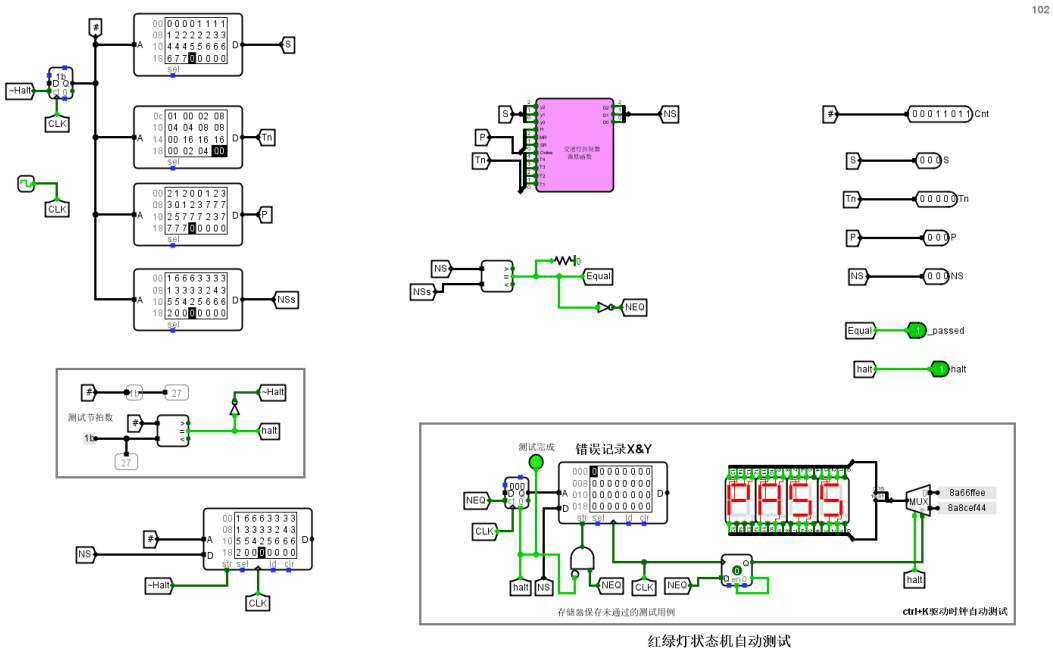


图 3-36 交通灯状态机电路测试图(logisim)

Cnt	S	Tn	P	NS
00	0	03	2	1
01	0	1a	1	6
02	0	12	2	6
03	0	14	0	6
04	1	01	0	3
05	1	01	1	3
06	1	01	2	3
07	1	01	3	3
08	1	00	3	1

图 3-37 交通灯状态机电路测试图(头歌)

（4）测试分析

经过测试和分析，我们确认电路功能良好，能够稳定可靠地运行并输出预期的结果，且无报错或险象，符合设计要求。这验证了我们设计的交通灯状态机电路在实验目的和内容上的正确实现。

3.7 交通灯输出函数设计

（1）设计思路及设计过程

交通灯输出函数的设计旨在根据当前状态信号 y，生成对应的交通灯控制信号。

根据实验内容提供的状态描述和状态编码表，我们可以根据当前状态 y 来确定各个交通灯的控制信号。充分阅读题目要求后，下面将对具体的设计思路进行介绍。

状态解码： 首先，根据输入的当前状态信号（y），使用状态解码器将其解码为各个控制信号的状态。根据状态编码表，我们可以确定主干道和次干道的灯光状态，以及是否处于紧急状况下。

倒计时功能： 根据当前状态（y），确定倒计时的值。在非紧急状态下，根据状态编码表中给出的倒计时信息，确定倒计时的初始值和计时范围。对于不同状态下的倒计时值，需要使用计时器进行倒计时，并在倒计时结束时进行状态转换。

高峰期判断： 根据当前状态（y），判断是否处于高峰期。如果是高峰期，则根据状态编码表中给出的信息，设置相应的倒计时值；否则，使用非高峰期的倒计时值。

紧急状况处理： 如果当前状态（y）表示紧急状况，需要设置主干道绿灯，次干道红灯，并显示紧急通行信号，不进行倒计时操作。

输出控制信号： 根据解码后的状态和倒计时结果，生成相应的输出控制信号。这些控制信号包括主干道和次干道的红、黄、绿灯控制信号，以及紧急状况下的通行信号。

将分析结果形成真值表，状态 111 示意紧急状态并非无效状态，将输出函数真值表填入 Logisim 中，即可自动生成交通灯输出函数电路。

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

图 3-38 交通灯输出函数真值表设计指示图

（2）电路图

以下是设计的交通灯输出函数电路的示意图：

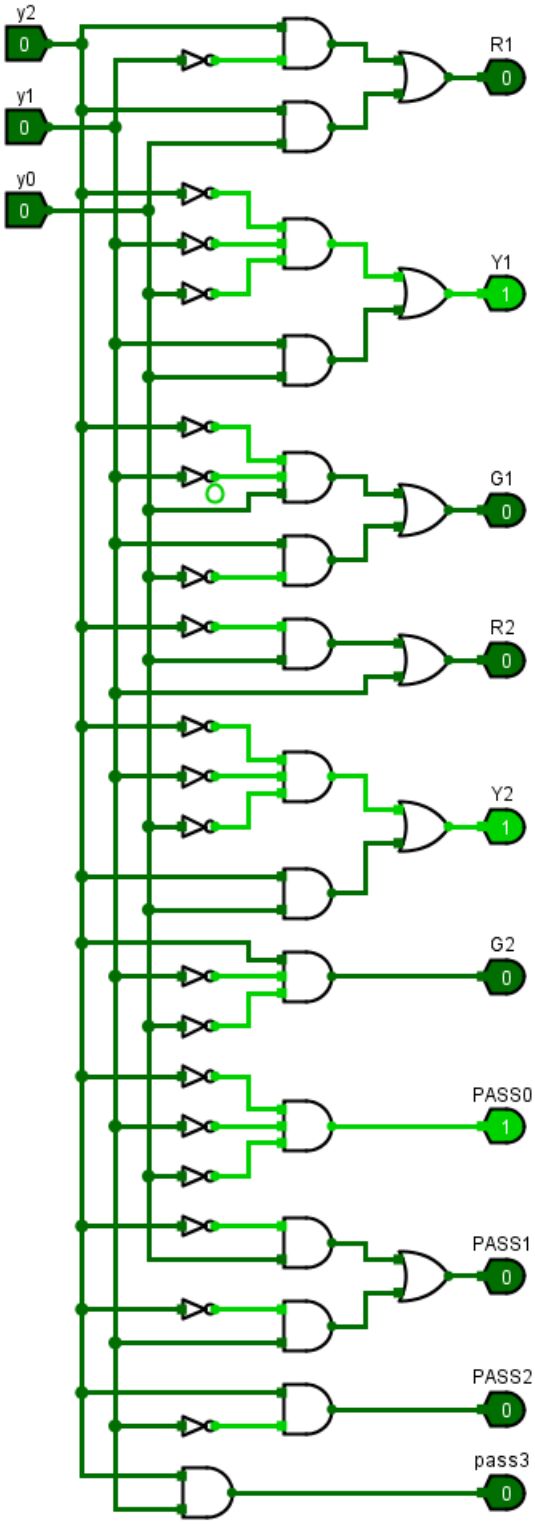


图 3-39 交通灯输出函数电路图

（3）测试图

设计完成后，对设计电路进行测试且显示通过，测试图如下：

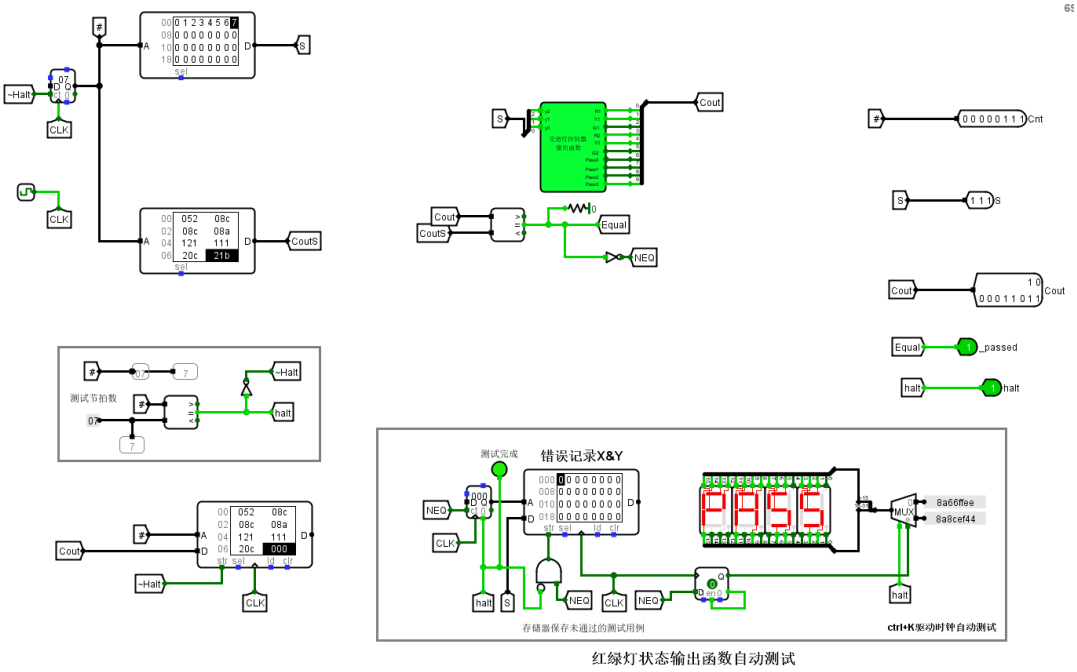


图 3-40 交通灯输出函数电路测试图(logisim)

图 3-41 交通灯输出函数电路测试图(头歌)

（4）测试分析

经过测试和分析，我们确认电路功能良好，能够稳定可靠地运行并输出预期的结果，且无报错或险象，符合设计要求。这验证了我们设计的交通灯输出函数电路在实验目的和内容上的正确实现。

3.8 交通灯控制系统

（1）设计思路及设计过程

交通灯控制系统的设计目的是实现一个能够根据输入信号的变化，控制交通信号灯状态的系统。在设计过程中，我们需要考虑系统的功能需求以及如何将这些需求转化为电路设计的具体步骤。

首先，我们需要明确系统的功能需求，包括系统的输入信号、输出信号以及内部信号。根据实验内容提供的要求，我们已经列出了输入信号、输出信号以及内部信号的详细说明，这些信号将作为系统设计的基础。

其次，根据功能需求，我们需要设计核心控制器系统，这包括状态机、状态寄存器和输出函数等组件的设计。状态机将根据输入信号的变化，控制输出信号的状态转移，从而实现交通灯的控制逻辑。状态寄存器用于存储系统的当前状态，以便状态机能够根据当前状态进行状态转移。输出函数则根据当前状态和输入信号，生成对应的输出信号。同时要注意 D 触发器的使能端 CLK 上升沿/下降沿的使用。通灯系统状态转移和输出模块电路示意图如下图所示。

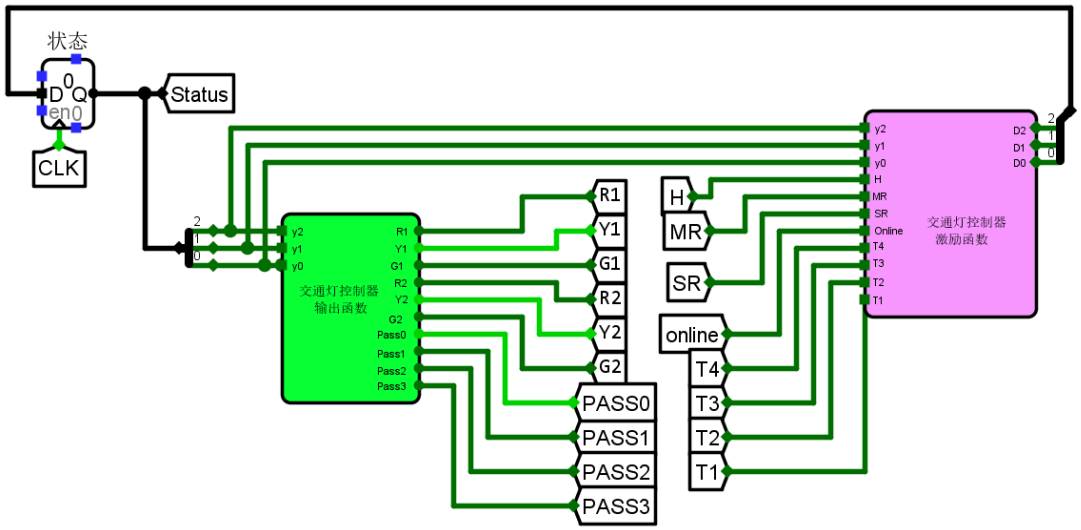


图 3-42 交通灯系统状态转移和输出模块电路示意图

接下来，我们需要设计倒计时电路，用于控制交通灯的倒计时功能。这包括主干道倒计时和次干道倒计时两部分。倒计时电路将根据系统的工作模式和当前状态，预置倒计时时间，并生成计时完成信号。倒计时完成信号将触发状态机进行状态转移，从而控制交通灯的状态变化。在设计倒计时电路的过程中，我们需要利用已经设计好的 2 位十进制可逆计数器构建倒计时电路，并利用 8 位无符号比较器比较计数器输出值，以生成计时完成信号。根据不同的工作模式和当前状态，我们需要预置不同的倒计时时间，如高峰期和非高峰期的主干道绿灯时间、次干道绿灯时间以及黄灯时间等。主次干道通信倒计时电路示意图如下图所示。

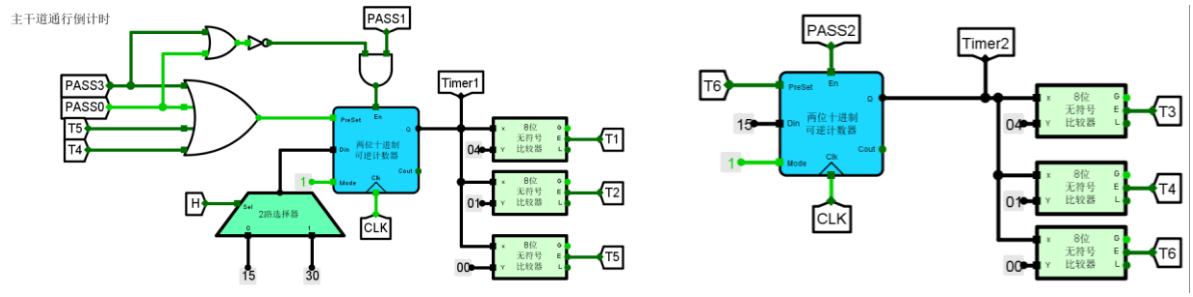


图 3-43 主(左)和次(右)干道通信倒计时电路示意图

最后，我们需要设计倒计时电路显示模块，用于显示交通灯的倒计时时间。由于主干道和次干道的显示时间一致，我们只需要设计一个显示模块，但是显示模块需要根据当前通行道路的不同，显示对应的倒计时时间。这需要利用多路选择器和七段显示驱动电路来实现。

而由于紧急情况的优先级最高，故 PASS3 最先进行处理，而由于电路输入的限制，PASS2 和 PASS0 的顺序亦不可更改，从而确保时钟信号的传输和时序约束的满足，提高电路的稳定性和可靠性。倒计时选择模块电路如下图所示。

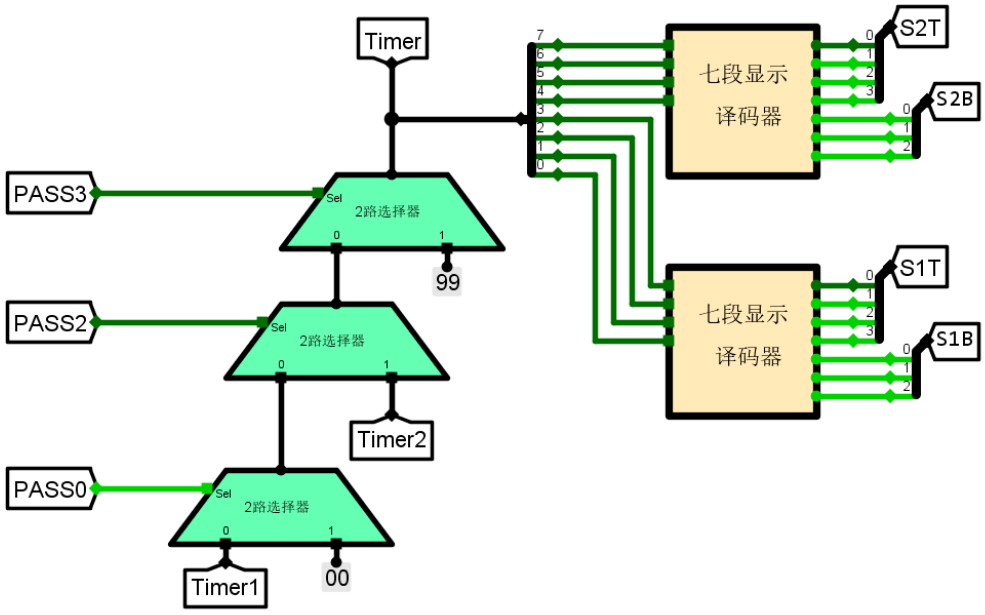


图 3-44 倒计时选择模块电路示意图

整个系统设计完成后，我们需要进行系统联调和测试，确保系统能够正常工作。在测试过程中，我们需要调整时钟频率，以满足系统的工作要求，并使用模拟输入信号来验证系统的功能是否符合设计要求。

（2）电路图

以下是整体设计的交通灯控制系统电路的示意图：

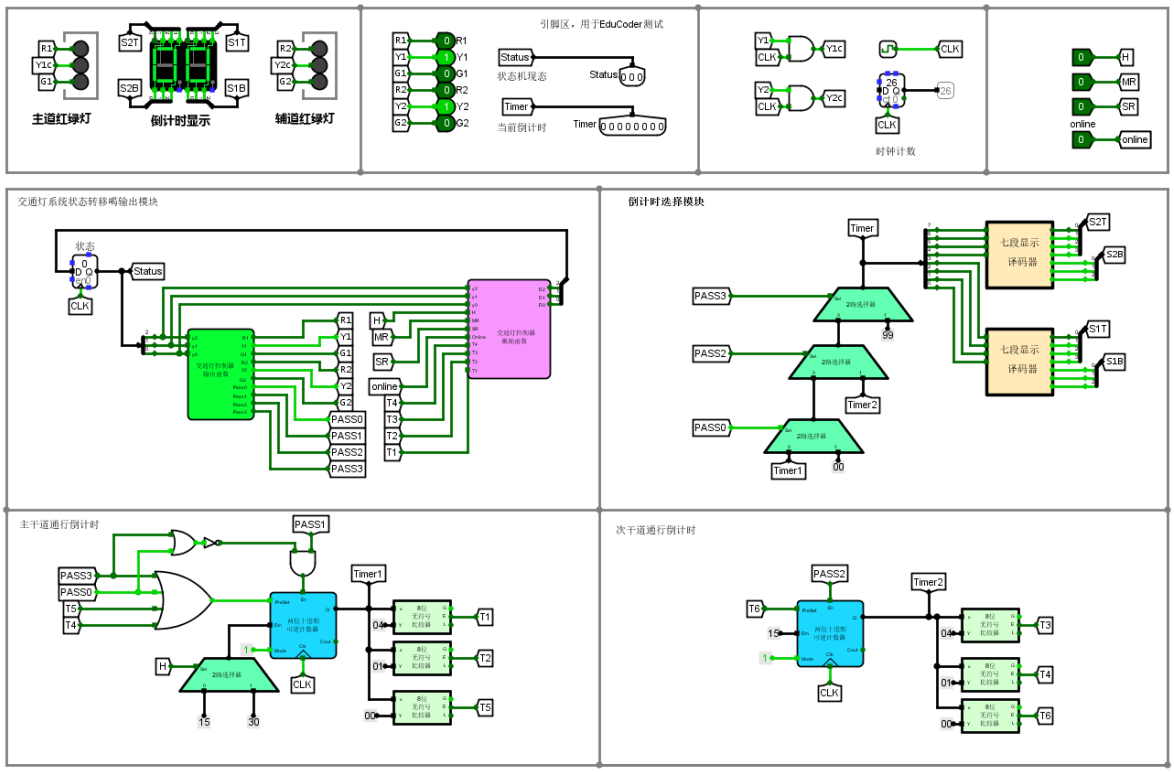


图 3-45 交通灯控制系统电路示意图

（3）测试图

设计完成后，对设计电路进行测试且显示通过，测试图如下：

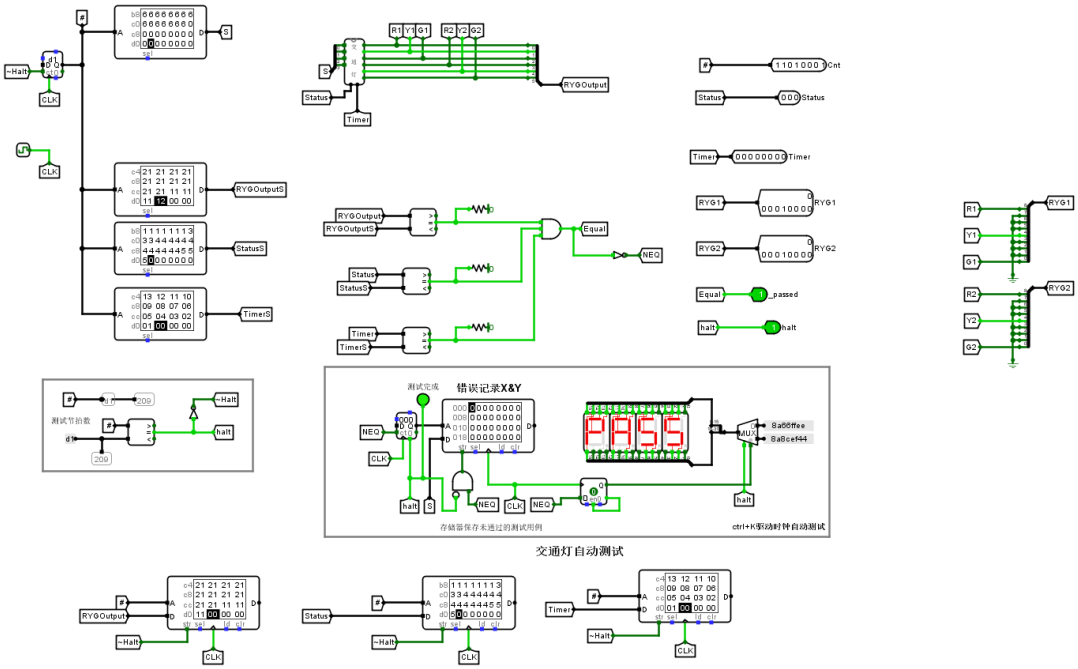


图 3-46 交通灯控制系统电路测试图(logisim)

图 3-47 交通灯控制系统电路测试图(头歌)

（4）测试分析

通过对交通灯输出函数进行测试，我们可以验证其在不同状态下是否能够正确生成相应的交通灯控制信号。测试分析应该包括以下内容：针对每个状态，验证输出信号是否符合预期，包括主干道和次干道的红、黄、绿灯控制信号，以及紧急状况下的通行信号。测试不同状态之间的状态转换，确保状态转换时输出信号的正确性和时序性。对高峰期和非高峰期进行测试，验证倒计时功能和状态转换的正确性。对紧急状况进行测试，确保在紧急情况下交通灯控制器能够正确响应并生成相应的控制信号。其次我们可以通过“时针滴答频率”调整测试电路运行频率，便于观看测试电路的状态。下面将对测试结果进行展示，因为实验验收时进行了全程的录制可对照查看，这里只根据下面交通灯控制系统状态编码表进行部分内容的静态呈现。

表 3-2 交通灯控制系统状态编码表

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

从 000 开始，为初始状态，此时主次干道均黄灯闪烁，显示 0。

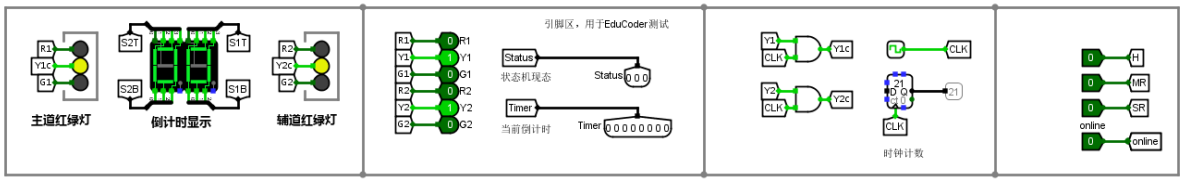


图 3-48 初始状态测试图

然后是 010，非高峰期主干道绿灯倒计时 12s，黄灯倒计时 3s，次干道红灯常亮。

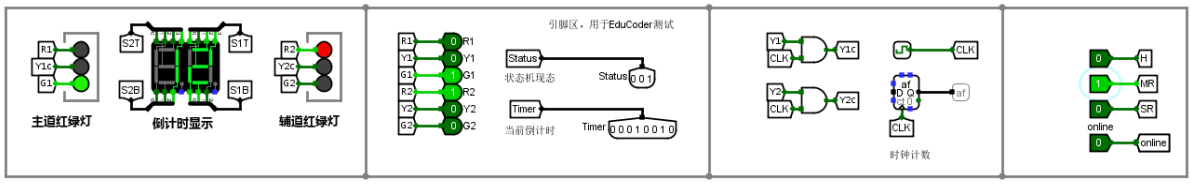


图 3-49 非高峰期测试图

接着是 011，主次干道都有通行请求时，主次干道绿灯交替进行。

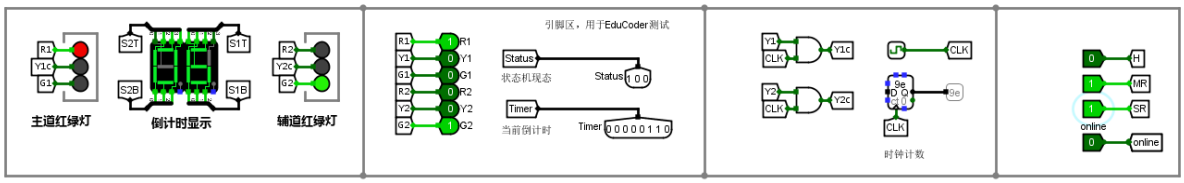


图 3-50 主次干道交替测试图

其次是 110，高峰期主干道绿灯倒计时 27s，黄灯倒计时 3s，次干道红灯常亮。

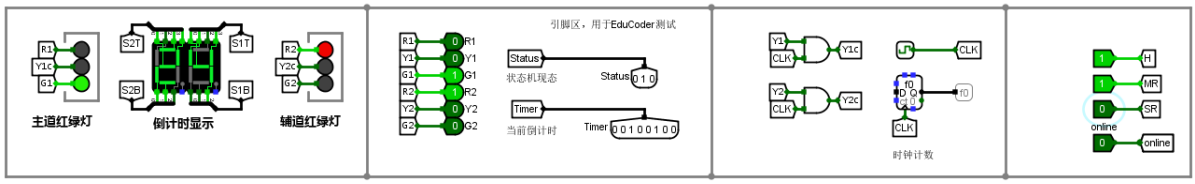


图 3-51 高峰期测试图

最后测试一下紧急状态 Online=1，主干道绿灯常亮，显示 99。

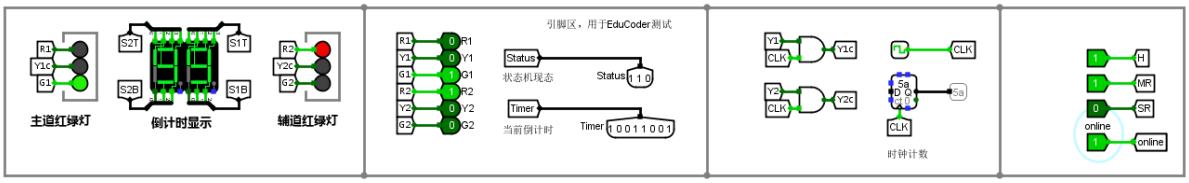


图 3-52 紧急状态测试图

经过测试和分析，我们确认电路功能良好，能够稳定可靠地运行并输出预期的结果，且无报错或险象，符合设计要求。这验证了我们设计的交通灯控制系统电路在实验目的和内容上的正确实现。

4　设计总结与心得

4.1 实验总结

首先完成对七段显示译码器、4 位和 8 位无符号比较器、2 位和 8 位 2 路选择器、模十可逆计数器以及二位十进制可逆计数器电路设计，最终再进行交通灯控制系统的整体设计，逐步从基础门电路向组合逻辑电路再到时序逻辑电路，将所学的内容进行实践，同时难度逐步递增。如若只是为了通过测试电路只需进行简单的填写和连线即可，但是实验的目的绝不仅是为了通关，而是让我们对数字逻辑电路设计进行更加深入地了解并且提高自主思考的能力，如何提高电路的性能和降低复杂程度都是可以不断地思考的问题。报告中呈现的可能并非最简但是如果可以给大家在思路上提供一定的参考，也是万般荣幸。

4.1.1 遇到的问题及处理

首先，由于 Logisim 软件的部件和布线与课堂所学的不太相同，需要提前了解不同符号形式，同时也需要掌握一些线路的实际使用，例如隧道(Tunnel)，分线器(Splitter)和三态门(Controlled Buffer)。而且在设计倒计时电路时，我们发现时序约束对于电路的稳定性和正确性至关重要，由于时钟信号的传输和逻辑电路的延迟，我们需要仔细考虑时序问题，并采取相应的措施来满足时序约束，同时尽量采用同步时序电路，我在报告中也对不同设计进行了阐述。然后是逻辑设计问题，在设计状态机和输出函数时，我们需要确保逻辑电路的正确性和稳定性，在初期设计中，可能会出现逻辑错误或者未考虑到的特殊情况，需要通过仿真和调试来发现和解决这些问题，后期该如何进行性能的优化也需要不断地思考和调试，在报告中也已呈现这里不再赘述。

4.1.2 设计方案存在的不足

尽管我的设计方案取得了一定的成果，但还存在一些不足之处：目前的设计方案主要针对了基本的交通灯控制功能，对于一些特殊情况或者功能扩展性方面考虑不够充分。未来可以进一步优化设计方案，增加功能扩展性，满足更多场景的需求；同时在当前设计中，可能存在一些性能优化的空间未被充分利用，例如，在电路设计和布线方面可能存在一些优化的余地，可以进一步提高电路的性能和稳定性。

4.2 实验心得

在完成本次实验的过程中，我深切体会到了数字逻辑设计的重要性和挑战性。通过设计和实现交通灯控制系统，我更深入地理解了逻辑门、数字电路和状态机的应用。在解决问题和实现功能的过程中，我学会了分析电路、优化设计、合理搭配元件以及保证整体系统的稳定性和正确性。通过理论知识的学习，我们知道实际运用中电路设计需要兼顾有效性和经济性，所以在本次实验中我对电路设计优化部分进行了更加细致地思考，同时也充分考虑了不同的设计情况，例如同步和异步时序电路等，这些部分也在实验报告进行了详细地展示。

实验的过程不仅提高了我的技能水平，还培养了我的逻辑思维和动手能力。通过实际操作，我进一步加深了对数字电路设计的理解，培养了我的实际动手能力，并锻炼了解决问题的能力，深刻认识到了设计理论与实际操作的结合的重要性。同时，在实验过程中，与同学之间密切合作和有效沟通起到了至关重要的作用。课程开始，高老师让我们组队进行讨论而非急于开始动手实验，我们也受益于此，通过对问题不断深入地讨论，不同的思路相互碰撞产生火花，让我们能够更好地解决问题、优化设计，并最终完成了设计任务。非常感谢老师的耐心指导和同学们的热情帮助，才得以在实验报告中呈现丰富的内容，在这次实验中收获颇丰！

透过数字电路设计的过程，我学会了不断思考、分析和改进的能力，这种思维方式将对我的专业学习和未来职业发展产生积极影响。我将继续保持学习和探索的热情，注重细节，培养团队协作精神，不断提升自己的综合素养和专业技能，努力成为一名优秀的网络空间安全专业人才，争取为国家网安安全建设做出一份自己的贡献。这次实验为我打下了良好的基础，也启迪了我在学术研究和工程实践上的思考和进步。

4.3 意见与建议

针对本次实验，我们建议整体实验课程可以早于理论课程考试安排，可以让同学们有更多的时间对实验内容进行深入探索，便于在理论考试涉及的实验部分能充分反映学生对实验的完成和思考程度，同时也可以加强对理论课程的理解。

非常感谢老师们对课程的辛勤付出和准备，在过程中还在不断完善评议程序和实验内容，整体实验体验感非常好收获颇丰！希望通过不断地改进和优化，相信本实验课程能够为学生提供更好的学习体验和成长空间。

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：