

华中科技大学

2023

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:	CS2208
学 号:	U202215628
姓 名:	方子豪
电 话:	19546890096
邮 件:	3299488768@qq.com
完成日期:	2023. 12. 4

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
1.1	实验名称	1
1.2	实验目的	1
1.3	实验环境	1
1.4	实验内容	1
1.5	实验要求	3
2	实验总体设计	4
2.1	实验总体设计思路	4
2.2	实验总体设计框架	4
3	实验过程	7
3.1	7 段数码管驱动电路设计	7
3.2	无符号比较器（2 位、4 位、8 位）	9
3.3	2 选 1 选择器设计（2 位、8 位）	15
3.4	十进制可逆计数器（包含状态机、输出函数及整体电路）	17
3.5	两位十进制可逆计数器	24
3.6	交通灯状态机	26
3.7	交通灯输出函数设计	31
3.8	交通灯控制系统	34
4	设计总结与心得	37
4.1	实验总结	38
4.1.1	遇到的问题及处理	38
4.1.2	设计方案存在的不足	38
4.2	实验心得	38
4.3	意见与建议	39

1 实验概述

1.1 实验名称

交通灯系统设计。

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/g8vqp5xw/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、车辆安全、迅速地通过，在交叉路口的每个入口处设置了红、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。

设计一个交通灯控制系统，具体内容及要求如下：

（1）输入信号

输入信号包括高峰期信号 H，主干道通行请求 PCM，次干道通行请求 PCC 和总控制台控制信号 Online。

（2）输出信号

输出信号包括 1 个 7 段数码管显示数字，用于显示红灯、绿灯和黄灯的剩余时间；6 个 Led 灯，用于显示主干道和次干道的红灯、绿灯和黄灯。

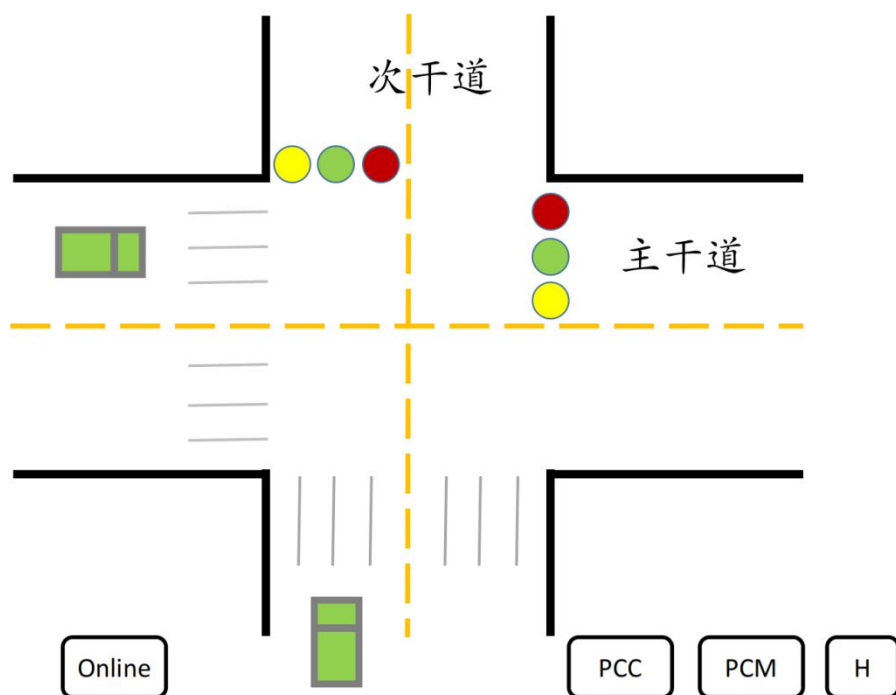


图 1-1 交通灯控制系统示意图

(3) 具体功能

- 路口指示灯规则为：“红--绿--黄”循环；
- 控制参数假设：红灯 15 秒，绿灯 12 秒，黄灯 3 秒；
- 通行请求定义：主干道通行请求（PCM）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（PCC）包括：次干道方向有车辆信号和主干道有行人通过信号。
- 通行规则 1：主干道和次干道均无通行请求，主、次干道两边黄灯“闪亮”。
提示：“通过时要注意观察”；
- 通行规则 2：主、次干道一边有通行请求，一边无通行请求，有通行请求一边绿灯亮，它的倒计时时间为 16s，归 0 后重新开始倒计时。
- 通行规则 3：只有主干道有通行请求 PCM，此时接收到次干道通行请求 PCC，则在绿灯倒计时为 0 时，考虑次干道方向的车辆或行人通行；只有次干道有通行请求的情况类似。
- 通行规则 4：非高峰时期，主、次干道均有通行请求时，主、次干道交替通行。
- 通行规则 5：高峰时期，主、次干道均有通行请求时，主、次干道交替通行，主干道放行时间（绿灯时间+黄灯时间）加倍。

i. 通行规则 6: 由交通控制中心发出的总控制台控制信号 (Online), 当 Online=1, 本地交通灯控制器控制权“失效”, 且主干道放行, 次干道禁止通行、当 Online=0 本地交通灯控制器恢复控制权 (接着原来的状态进行运行)。

1.5 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。

2 实验总体设计

2.1 实验总体设计思路

(1) 首先明确目的,通过分析实验要求我们发现我们需要设计的交通灯是有着四种不同持续时间的,分别是 3s, 12s, 15s 和 30s, 通过分析我们可以发现只需要设计三种灯的持续时间就行。

(2) 写出所有情况的状态并化简,因为只有两条路,而我们发现一般情况下两条路的灯都是同时黄色或者一个红色一个绿色或者黄色,所以依照此类情况,我们可以将不同状态合并化简,最后得出所需最少得状态。

(3) 写出状态表,画出状态图,明确状态转移情况。

(4) 明确状态之间转换和维持当前状态以及输出所需要的不同模块,比如两位十进制可逆计数器用来输出计数倒计时。

(5) 分别实现每一个需求的部分,最后再对需求部分进行组装连线从而实现交通灯电路实验。

2.2 实验总体设计框架

将交通灯系统分为六个模块,分别是:红绿灯状态转移模块,红绿灯状态输出模块,倒计时选择模块,主干道计时模块,次干道计时模块,单侧同行倒计时模块。

(1) 红绿灯状态转移模块:将红绿灯的状态设置成 8 种,通过 Excel 表格将状态转移自动化为输出函数再输入 logisim 中实现,实现主干道红绿灯状态间的切换功能。

(2) 红绿灯输出状态模块:对于二位数输出显示,选择使用两个七段数码管来显示当前时间,数码管的设计使用真值表输入完成。

(3) 倒计时选择模块:红绿灯状态倒计时总共最多只有两位数,选择两位十进制可逆计数器进行计数,通过预制等手段控制倒计时时间。

(4) 主干道计时模块:与次干道计时模块相限制,两者分别使用一套两位十进制可逆计数器计数

(5) 次干道计时器模块:与主干道计时模块相限制,两者分别使用一套两位十进制可逆计数器计数

（6）单侧同行倒计时模块：分别用来处理单侧同行情况和特殊情况，对于单侧同行选择与主干道和次干道计时相同的方案，对于紧急情况选择特殊预制的两位十进制可逆计数器实现。

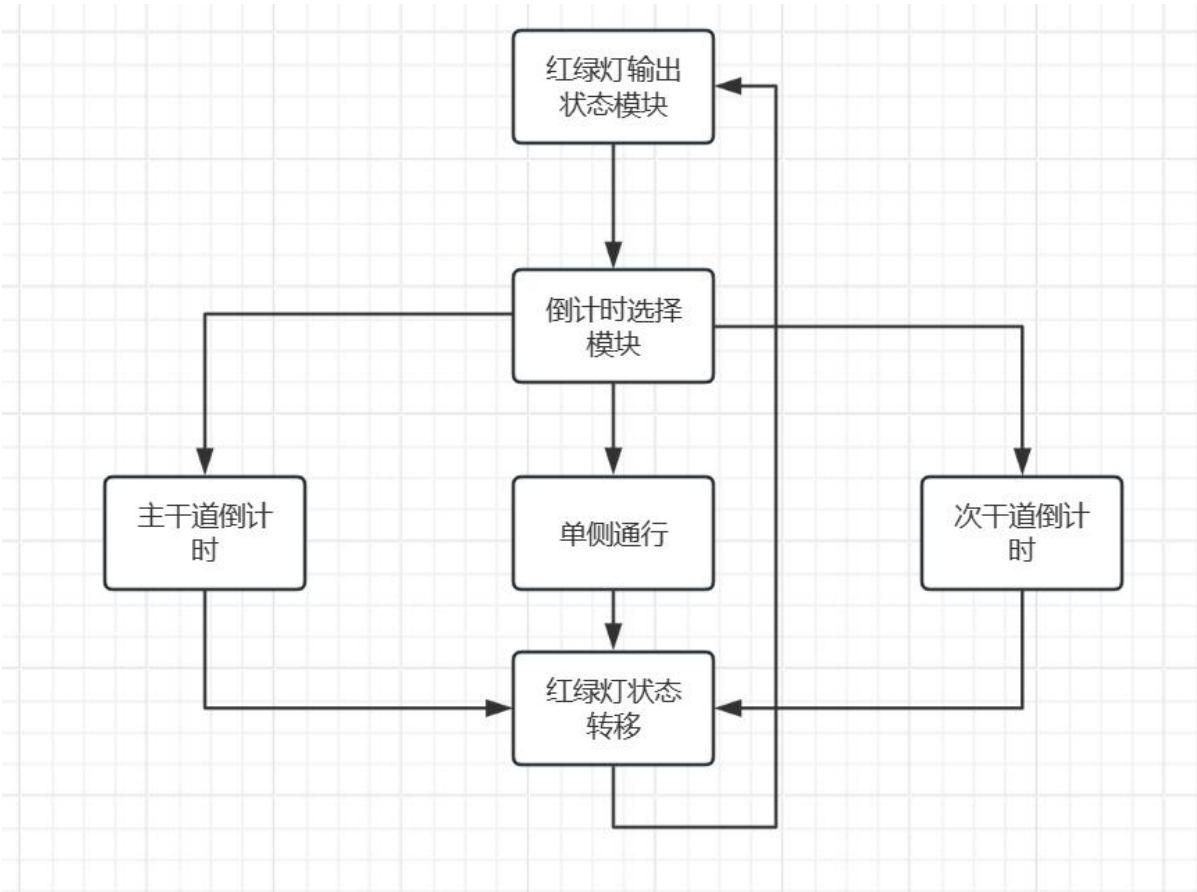


图 2-1 交通灯设计框架

对应其中的模块我们可以将电路的状态细分成八种，而状态的改变随着输入的改变进行。比如我将 S0 设计为没有通行请求，那么我一旦输入 PCM 就会进入主干道通行状态，以此类推，我们可以得出对应的状态间的转换图，如图。

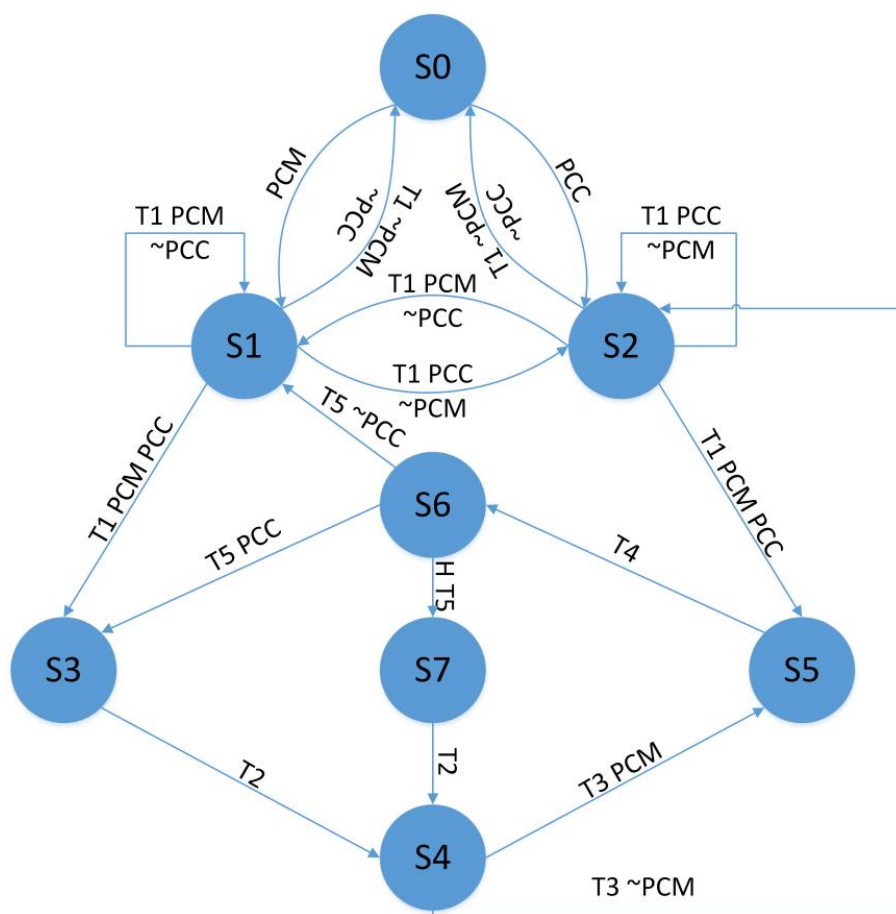


图 2-2 交通灯状态转换图

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

因为个位数只有 0~9 十个，对应的每一个数字的形状都可以直接得知，所以可以选择用七段数码管来实现。用编号为 1~7 的七个数码管分别放置形成“日”字形，对应输出每一个数时选择对应的编号的数码管亮，比如 0001 代表数字“1”那么我就让对应的“1”即 seg_7 亮起来即可，将数码管对应的亮与不亮作为状态输入真值表，并最后用 logisim 自动生成对应电路。

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

图 3-1 数码管真值表

(2) 电路图

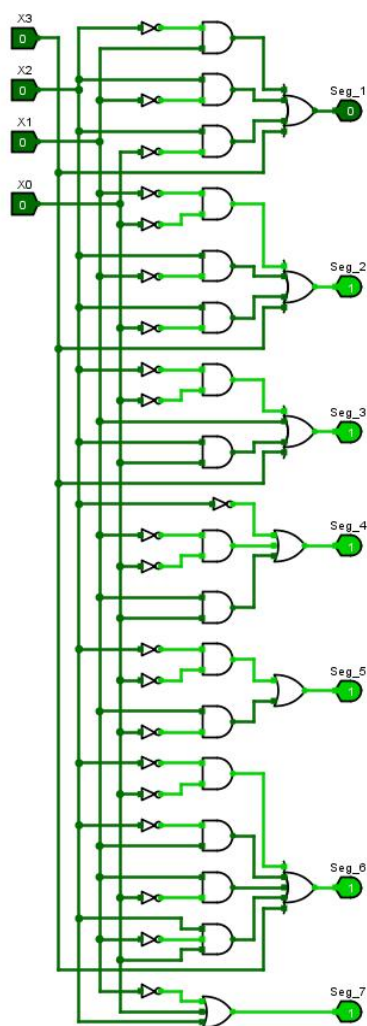


图 3-2 七段数码管电路图

(3) 测试图

仅粘贴 3, 4 的测试图

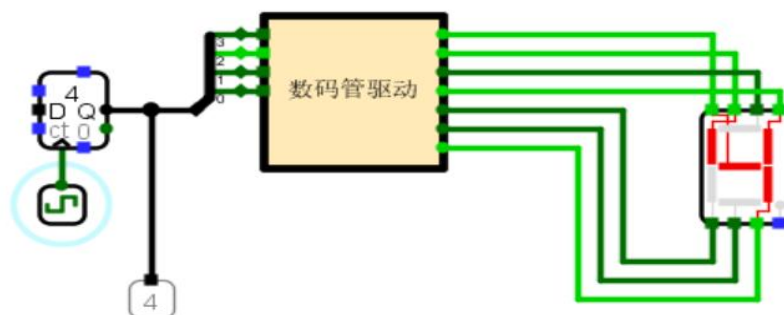


图 3-3 数码管测试（数字 4）

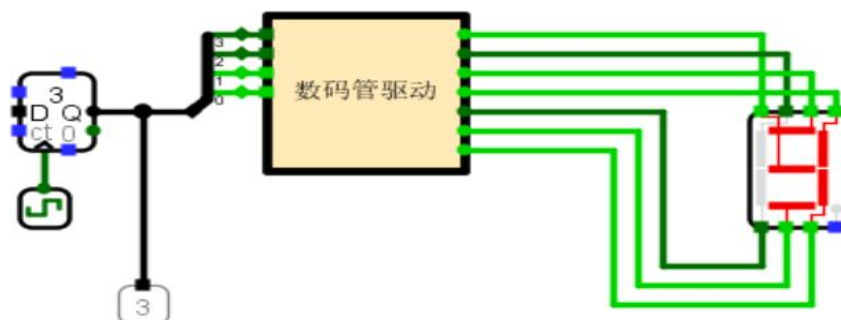


图 3-4 数码管测试（数字 3）

（4）测试分析

当 $x_3x_2x_1x_0=0011$ 时，对应十进制数为 3，对应引脚 13476 均发光。当 $x_3x_2x_1x_0=0100$ 时，对应十进制数为 4，对应引脚 1247 均发光。由实验测试结果可以看出实验电路设计正确。

3.2 无符号比较器（2 位、4 位、8 位）

（1）设计思路及设计过程

对于偶数位无符号比较器，我们设置三个档位，分别是 great, equal 和 less，假设有 $2*n$ 位，那么可以通过拆分，将该比较器分成高 n 位和低 n 位，先比较高位，高位更大的必然整个数字更大们可以直接输出 great, 更低同理，直接输出 less。对于高位相等的情况，那么直接比较低位就行，低位是什么情况就输出什么情况即可。对于两位无符号比较器总共也只有少数情况，所以我们可以通过真值表进行快速设置。对于四位无符号比较器就可以通过上面的思路和刚刚实现的 2 位来进行组装，对于 8 位的同理，使用四位的即可。

(2) 电路图

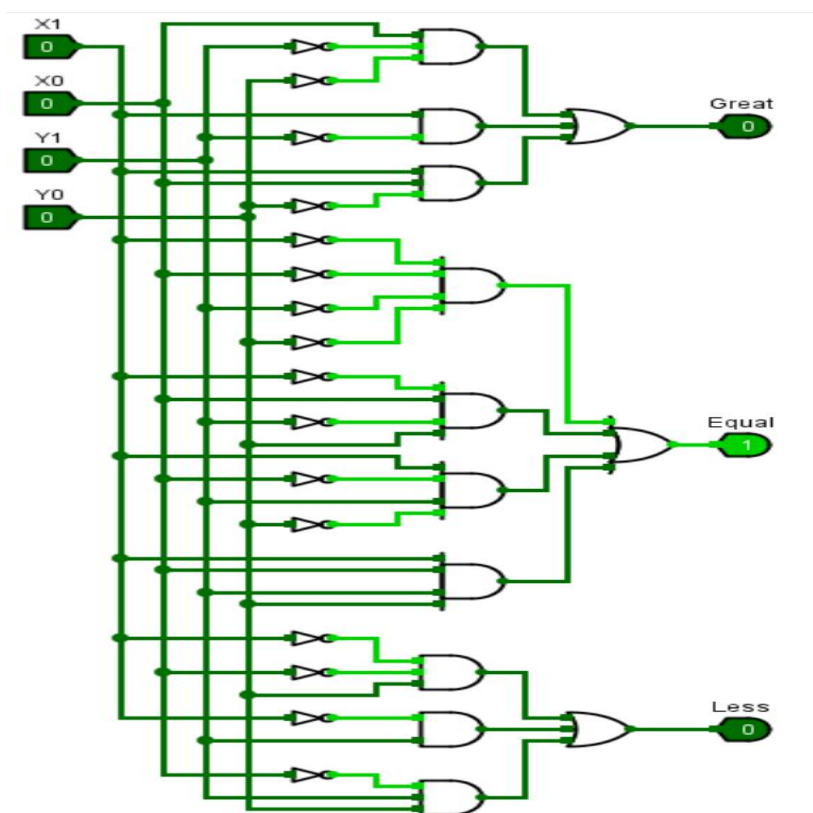


图 3-5 两位无符号比较器电路图

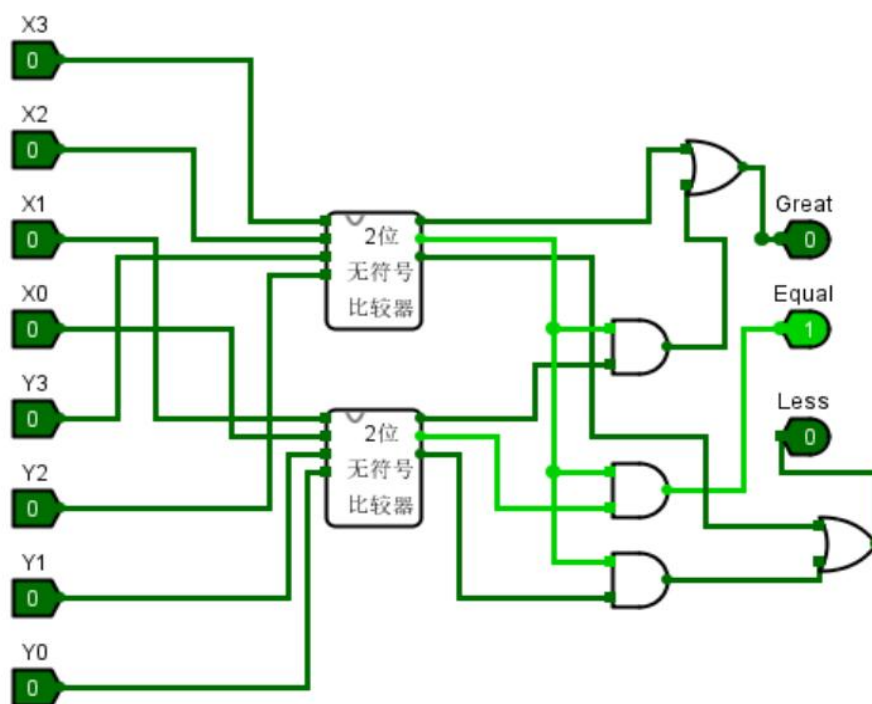


图 3-6 四位无符号比较器电路图

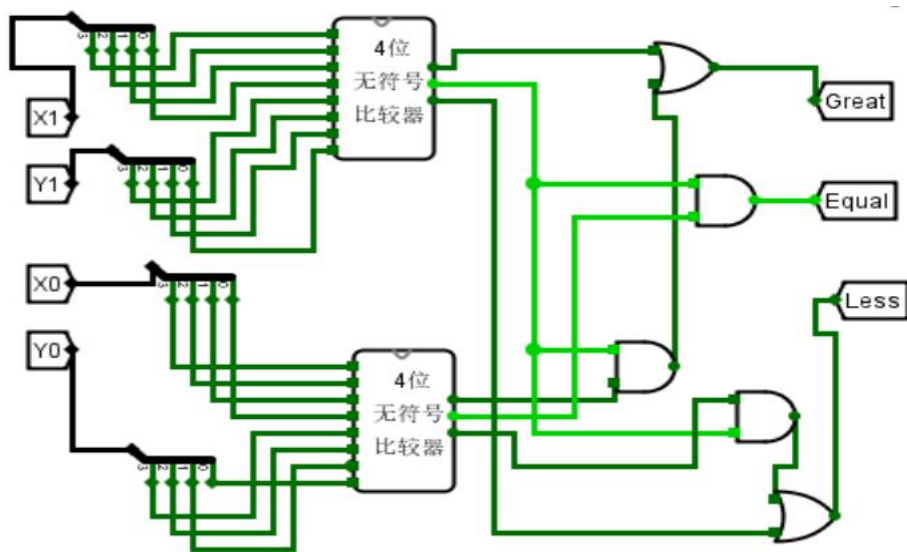


图 3-7 八位无符号比较器电路图

(3) 测试图

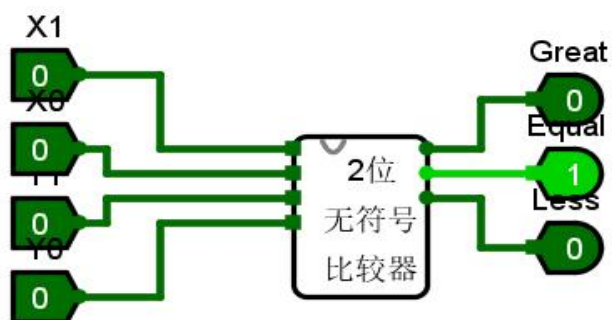


图 3-8 两位无符号比较器测试图（相等）

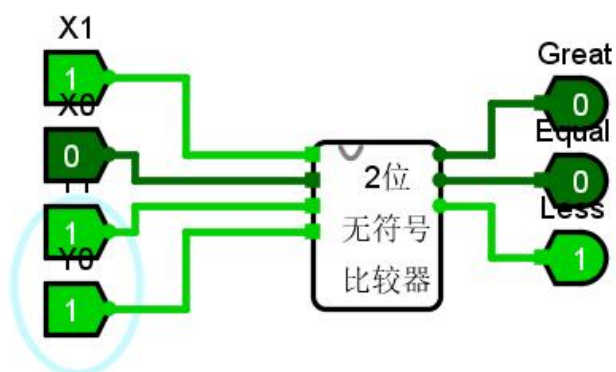


图 3-9 两位无符号比较器测试图（小于）

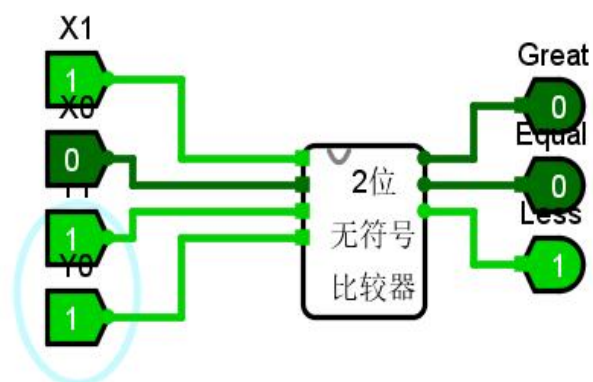


图 3-10 两位无符号比较器测试图（大于）

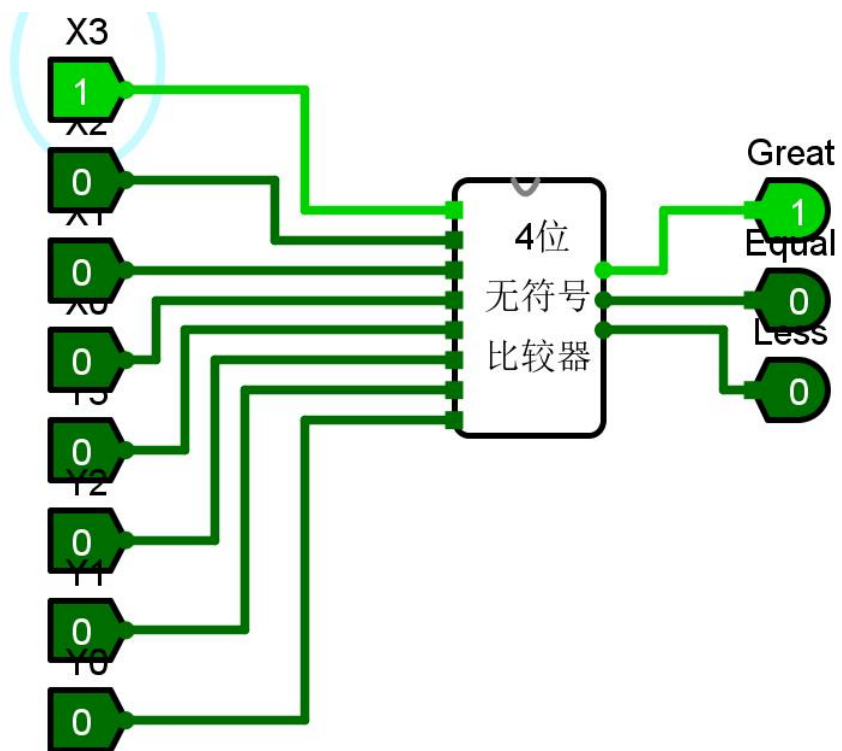


图 3-11 四位无符号比较器测试图（大于）

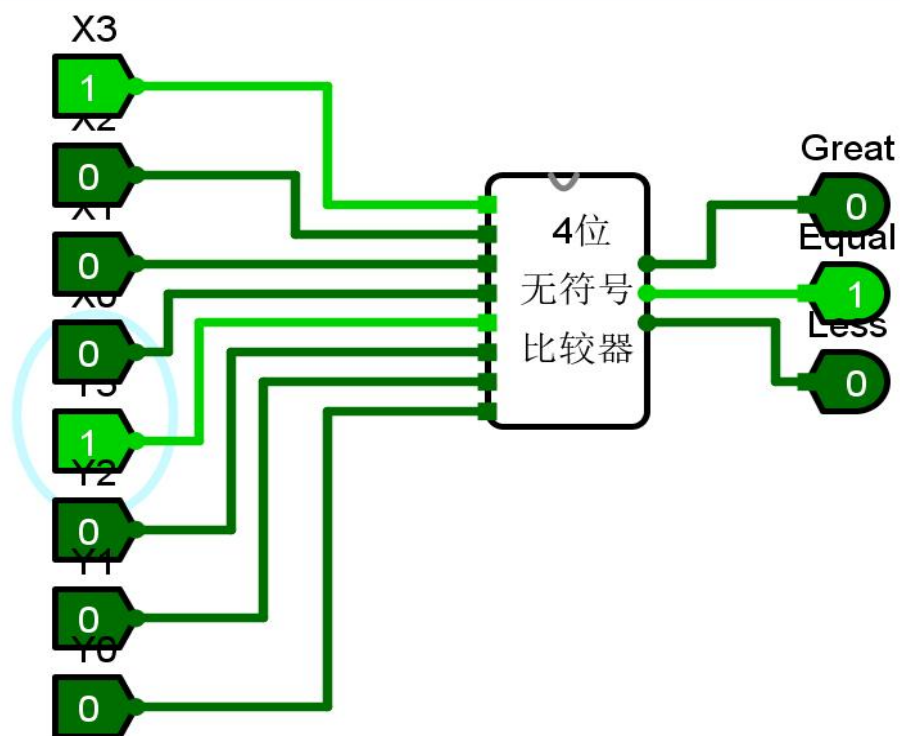


图 3-12 四位无符号比较器测试图（等于）

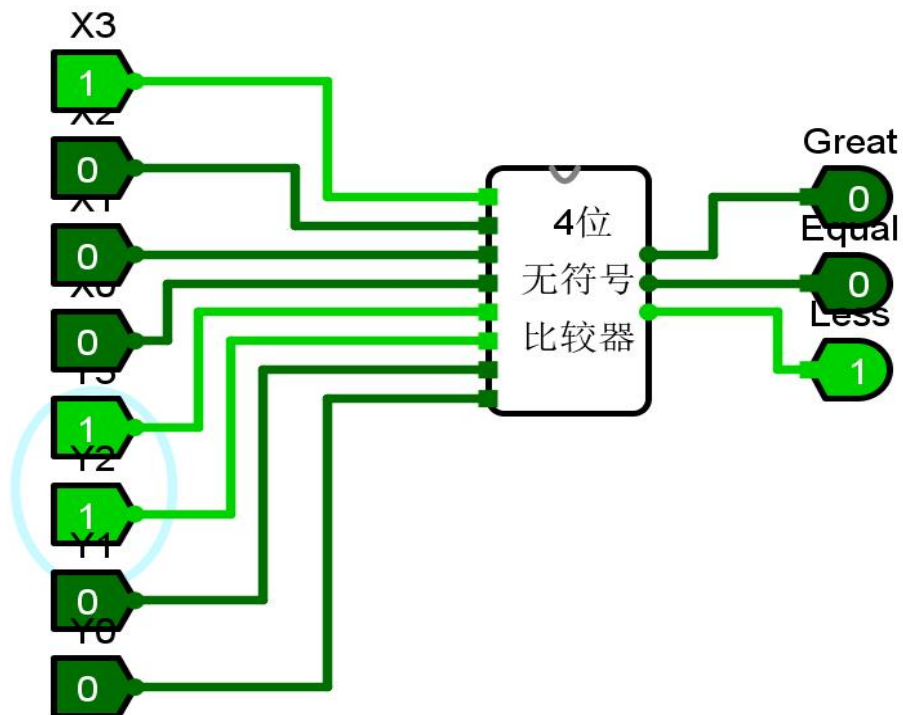


图 3-13 四位无符号比较器测试图（小于）

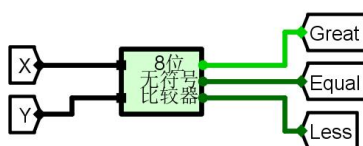
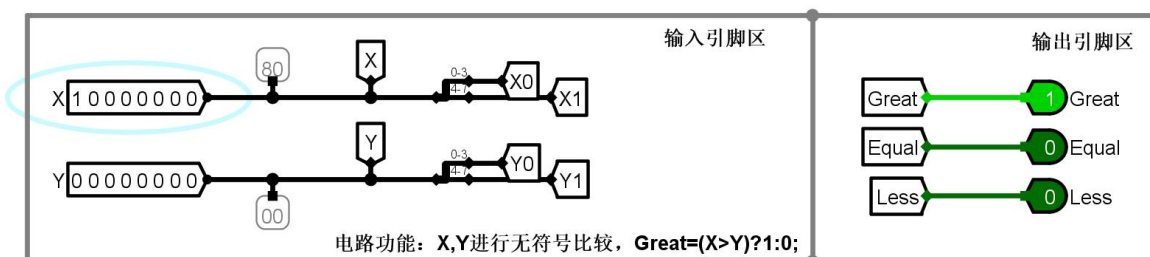


图 3-14 八位无符号比较器测试图（大于）

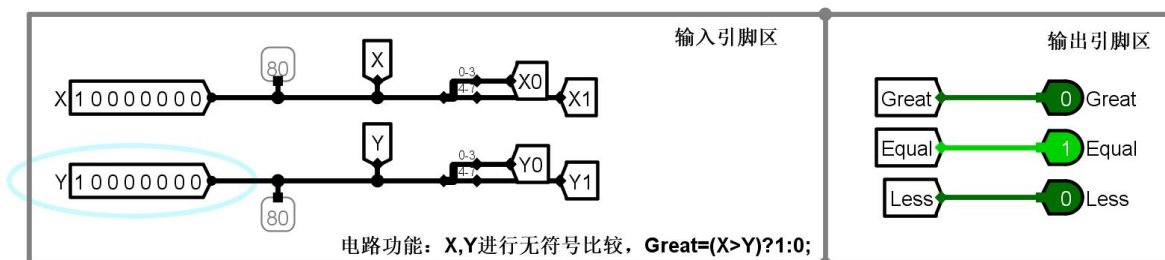


图 3-15 八位无符号比较器测试图（等于）

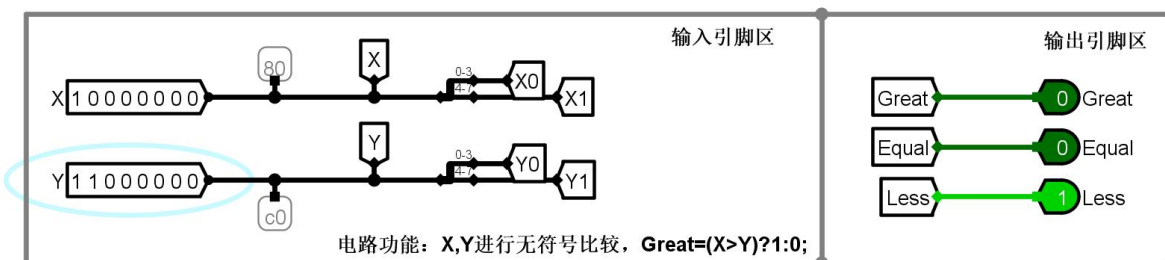


图 3-16 八位无符号比较器测试图（小于）

(4) 测试分析

- 2 位无符号比较器, 当 $x_1x_0=00, y_1y_0=00$ 时, 比较器结果为 Equal; 当 $x_1x_0=10, y_1y_0=00$ 时, 比较器结果为 Great; 当 $x_1x_0=10, y_1y_0=11$ 时, 比较器结果为 Less。
- 4 位无符号比较器, 当 $x_3x_2x_1x_0=1000, y_3y_2y_1y_0=1000$ 时, 比较器结果为 Equal; 当 $x_3x_2x_1x_0=1000, y_3y_2y_1y_0=0000$ 时, 比较器结果为 Great; 当 $x_3x_2x_1x_0=1000, y_3y_2y_1y_0=1100$ 时, 比较器结果为 Less。

c. 8 位无符号比较器，当 $X=10000000$ ， $Y=10000000$ 时，比较器结果为 Equal；当 $X=10000000$ ， $Y=00000000$ 时，比较器结果为 Great；当 $X=10000000$ ， $Y=11000000$ 时，比较器结果为 Less。

由实验测试结果可以看出实验电路设计正确。

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 选择器的实现较为简单，自己通过写输出函数即可发现只需要一路与上选择输入 sel ，另一路与上选择输入 sel 的非即可实现 2 路选择。对于八位选择器，一中方便的方法是可以将八位数分别拆成八个一位数，然后将这八个一位数分别用二路选择器选择，再进行合并就行。

（2）电路图

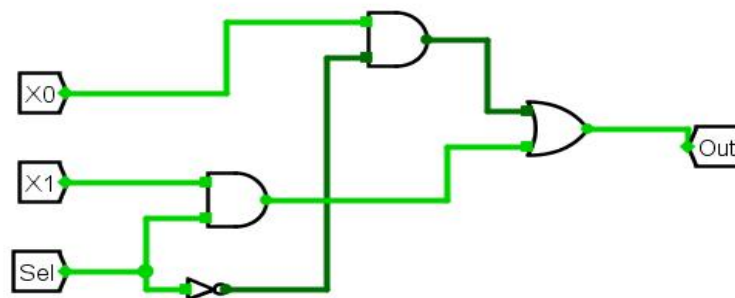


图 3-17 2 路选择器（1 位）电路图

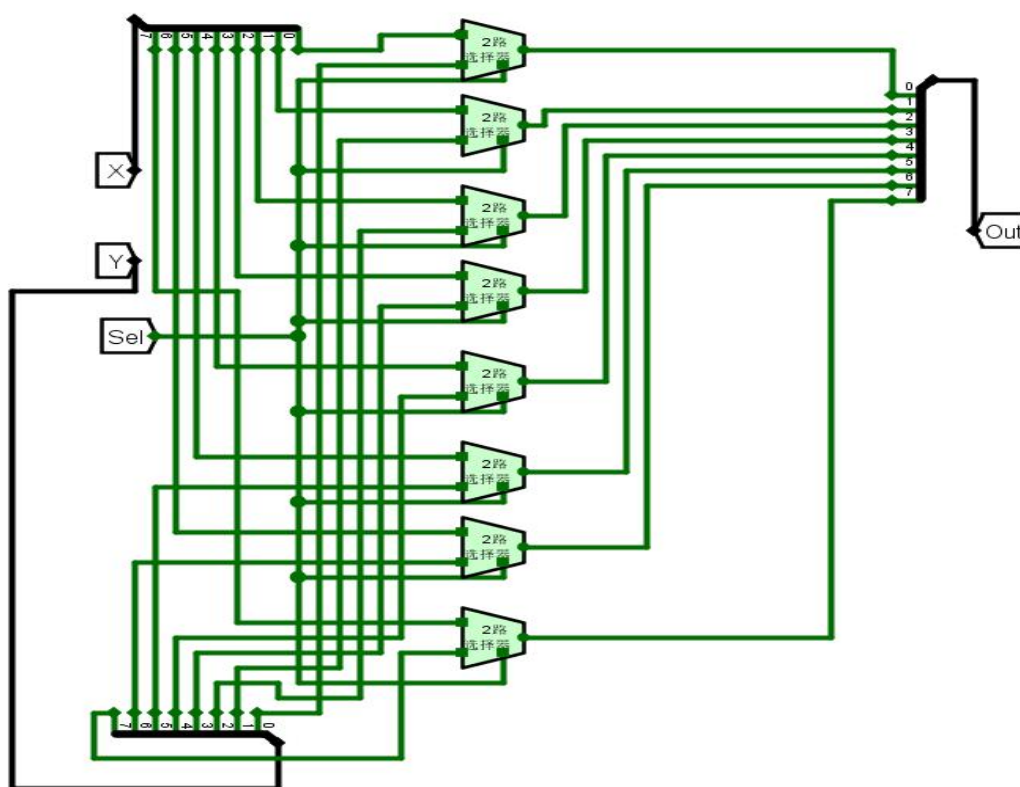


图 3-18 2 路选择器（8 位）电路图

(3) 测试图

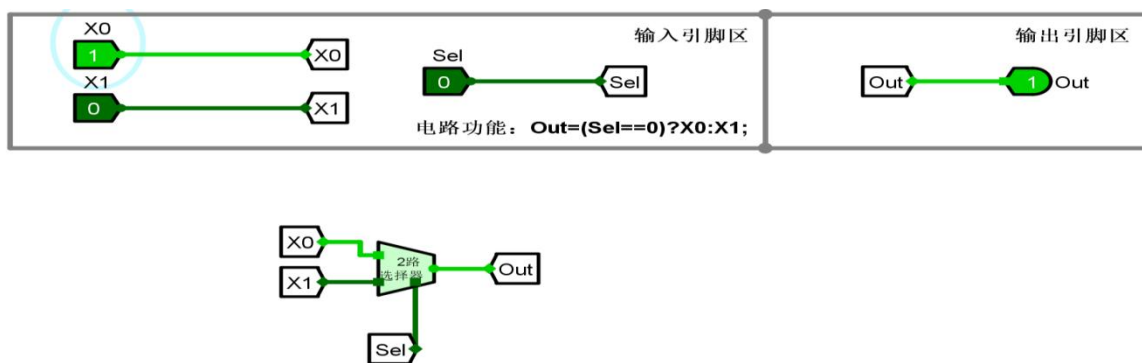


图 3-19 2 路选择器（1 位）测试图 1

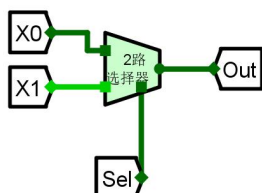
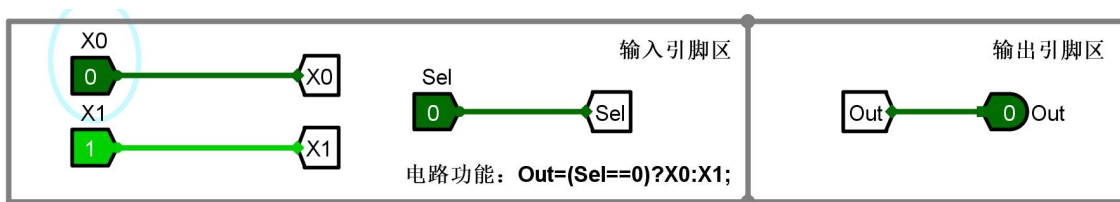


图 3-20 2 路选择器（1 位）测试图 2

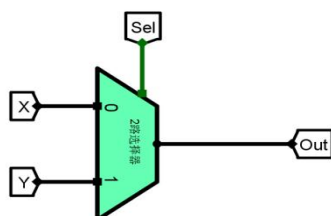
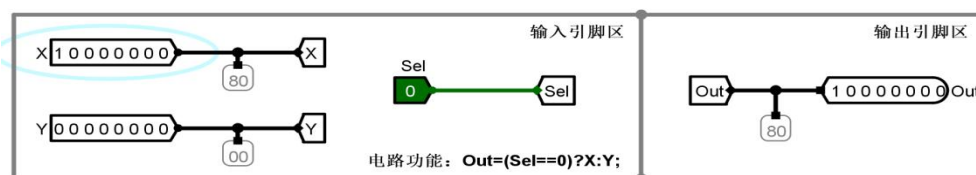


图 3-21 2 路选择器（8 位）测试图 1

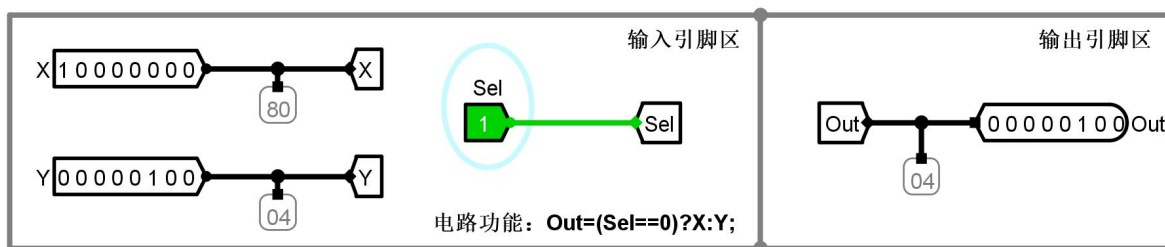


图 3-22 2 路选择器（8 位）测试图 2

(4) 测试分析

- 1 位 2 路选择器，当 $x_0=0$, $x_1=1$ 时， $Sel=0$ 输出 x_0 的值 0； $Sel=1$ 输出 x_1 的值 1。

b. 8 位 2 路选择器, 当 $x=10000000$, $y=00000100$ 时, $Sel=0$ 输出 $x0$ 的值 10000000 ; $Sel=1$ 输出 $x1$ 的值 00000100 。

由实验测试结果可以看出实验电路设计正确。

3.4 十进制可逆计数器（包含状态机、输出函数及整体电路）

（1）设计思路及设计过程

状态机：容易知道十进制其实本质上就是十个数不断循环，比如当正计数时，遇到 9 就会变成 0，倒计数时，遇到 0 就会变成 9。因为状态转换较为简单，状态较少，所以可以通过状态函数实现。刚好利用 Excel 中自动生成状态函数的功能，直接输入对应状态转换获得状态函数即可。

输出函数：当 $mode$ 等于 0 时正向计数，此时当计数到 9 时 $cout$ 有输出，

当 $mode$ 等于 1 时逆向计数，当计数到 0 时 $cout$ 输出，因为只有这两个状态输出 1，所以可以用真值表或者用输出函数来让 logsim 自动生成。

(2) 电路图

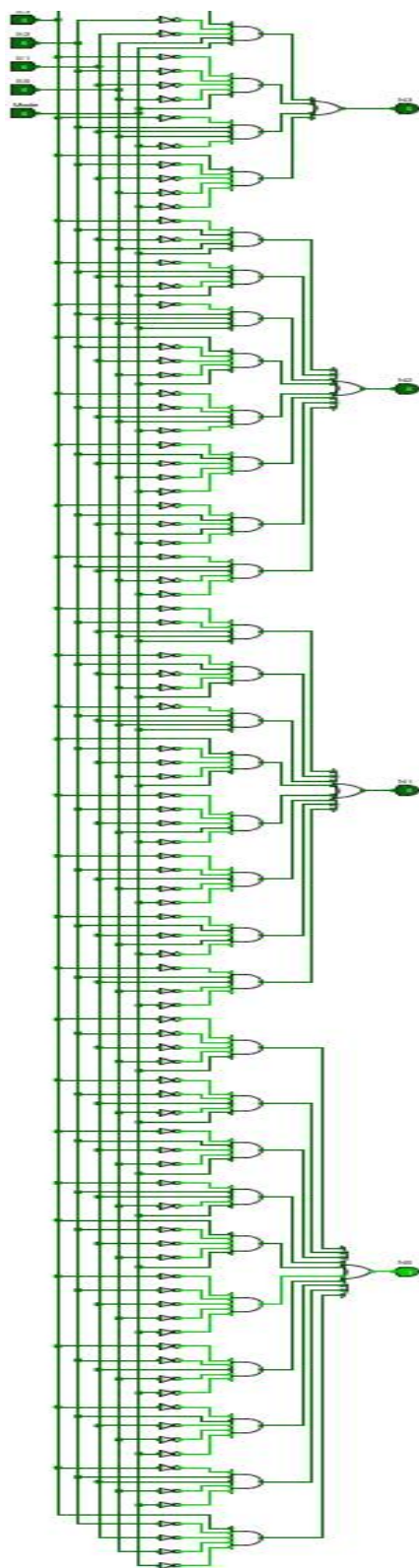


图 3-23 十进制可逆计数器状态机电路图

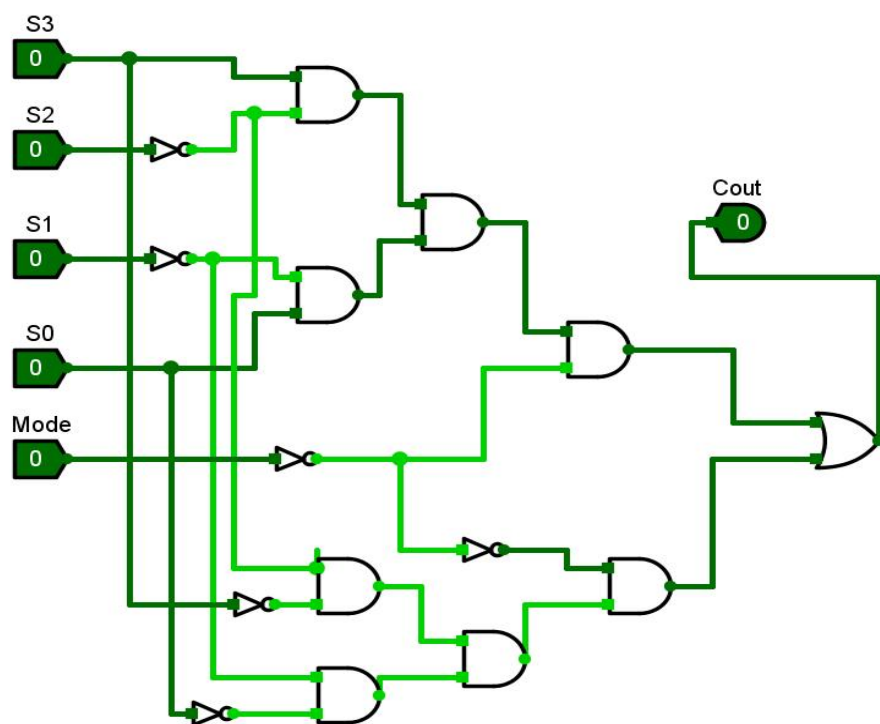


图 3-24 十进制可逆计数器输出函数电路图

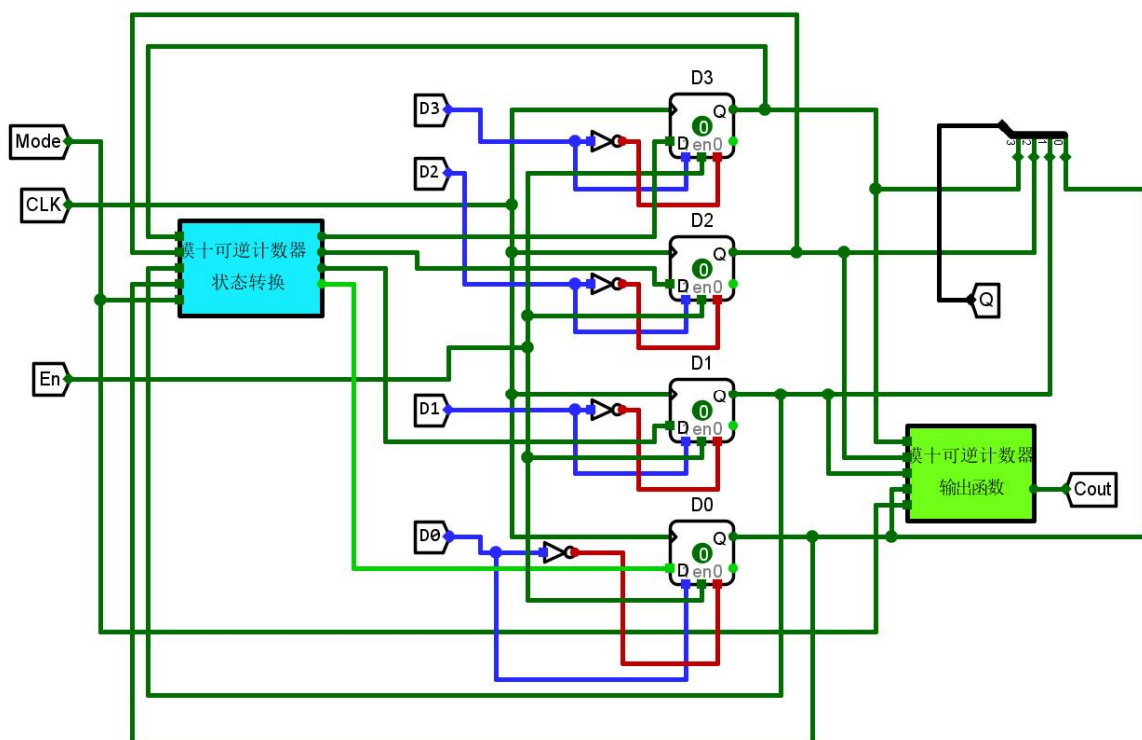


图 3-25 十进制可逆计数器电路图

(3) 测试图

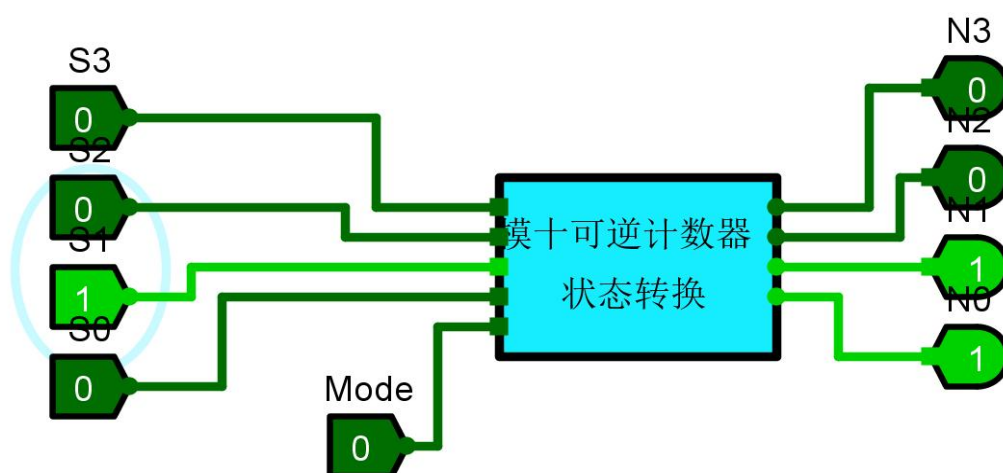


图 3-26 模十可逆计数器状态转换测试图 1

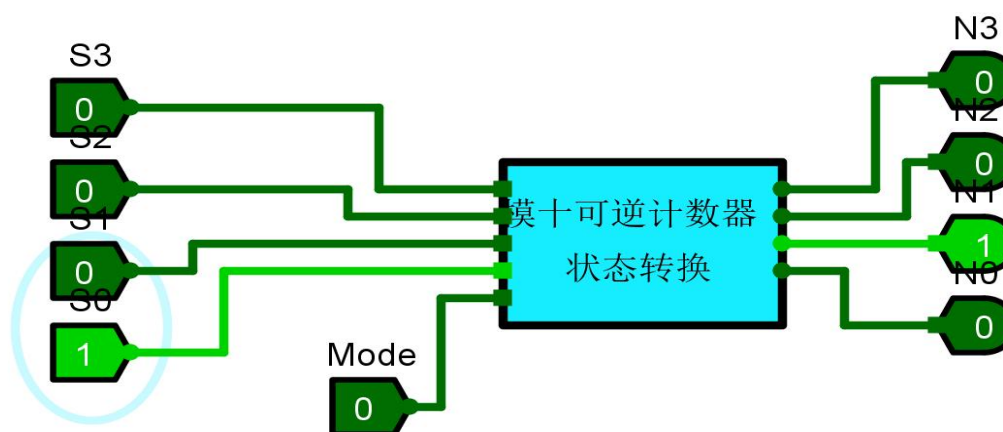


图 3-27 模十可逆计数器状态转换测试图 2

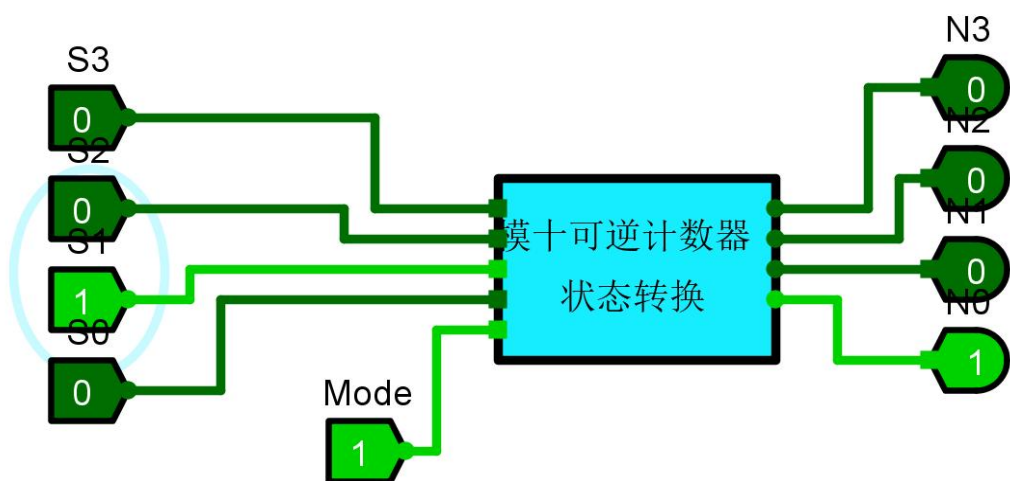


图 3-28 模十可逆计数器状态转换测试图 3

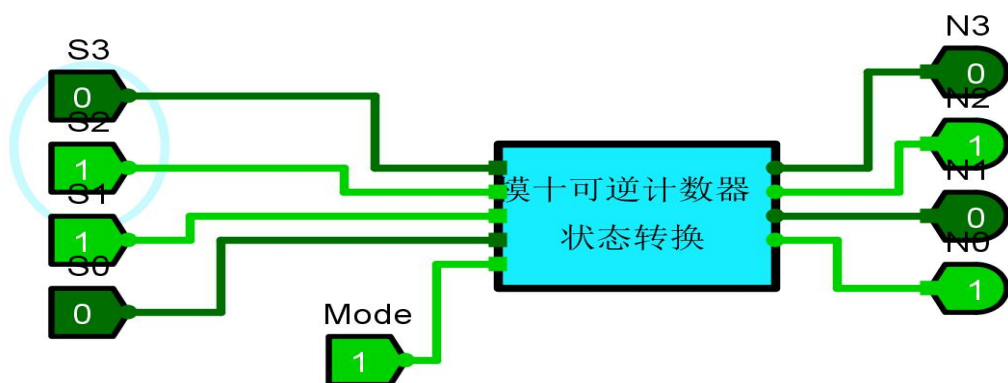


图 3-29 模十可逆计数器状态转换测试图 4

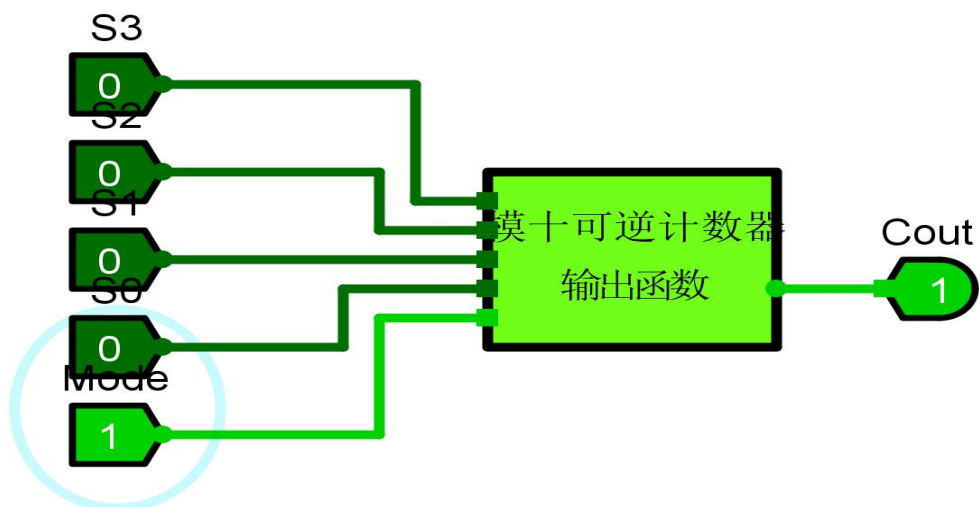


图 3-30 模十可逆计数器输出函数测试（借位）

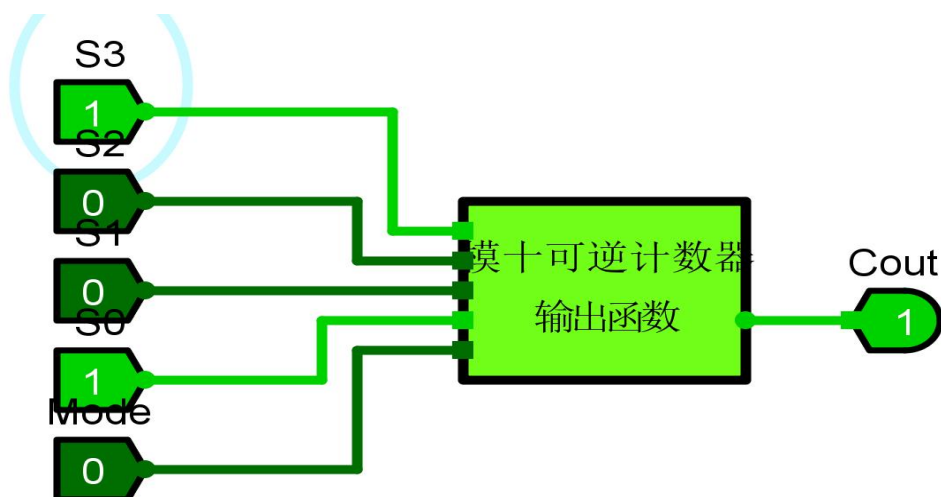


图 3-31 模十可逆计数器输出函数测试（进位）

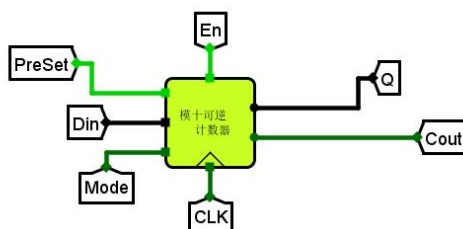
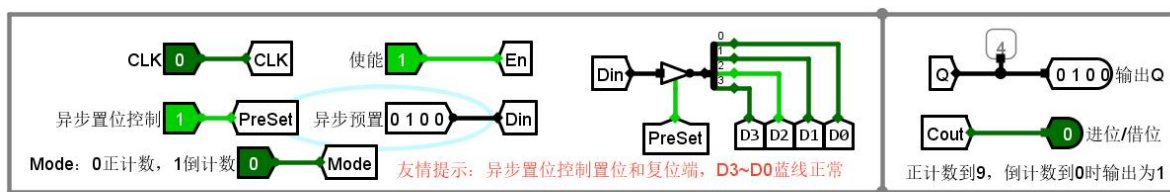


图 3-32 模十可逆计数器测试图

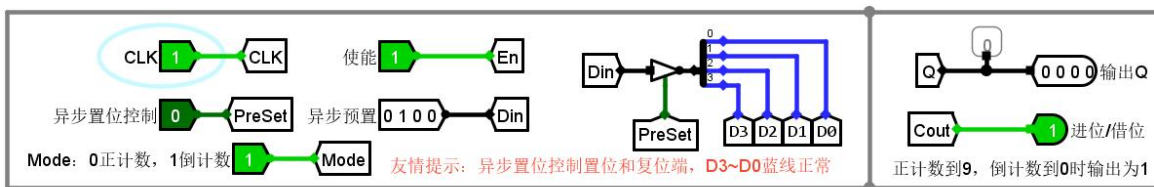


图 3-33 模十可逆计数器测试图（借位）

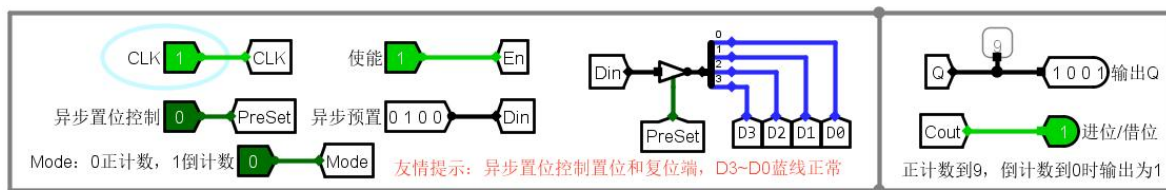


图 3-34 模十可逆计数器测试图（进位）

（4）测试分析

a. 异步预置: 当 PreSet=1 时, 此时会将 Din 端的数据直接给到输出 Q。当 Din=0100 时, 输出 Q=0100, 和 Din 相同, 且此时 En=1, CLK 输入了正脉冲（从 0 到 1）, Q 不变, 说明异步预置的优先级最高。

b. 计数: 当 PreSet=0, En=1 时, 此时为正常的计数功能。当 CLK 从 0 变 1 时, 便进行一次计数。Mode=0 时, 此时正向计数, CLK 输入正脉冲; Mode=1, 此时反向计数, CLK 输入正脉冲。

c. 进位/借位: 当 PreSet=0, En=1 时。Mode=0, 计数器在正向计数到状态 Q 为 1001 时, 会产生进位信号; Mode=1, 计数器在反向计数到状态 Q 为 0000 时, 会产生借位信号。

由实验测试结果可以看出实验电路设计正确。

3.5 两位十进制可逆计数器

（1）设计思路及设计过程

想要记录十位数就需要将至少两个十进制的计数器来组成两位十进制计数器, 第一位作为个位数来计数, 第二位作为十位数来计数, 每当第一位有进位或者借位的时候（即第一个计数器 cout 为 1 的时候）, 第二个计数器才会有计数, 这样便可以做到计数到二位数的范围。又因为第一个计数器动了十次第二个计数器才动, 所以将第一个计数器的 cout 输出与第二个计数器的使能端绑在一起从而实现每十个数十位数计数一次, 最终总的 cout 就是当我为 99 增加或者 00 减少时, 两个 cout 同时输出为 1, 这样会代表我这个两位数迭代了, 从 99 变成 00 或者 00 变成 99。

（2）电路图

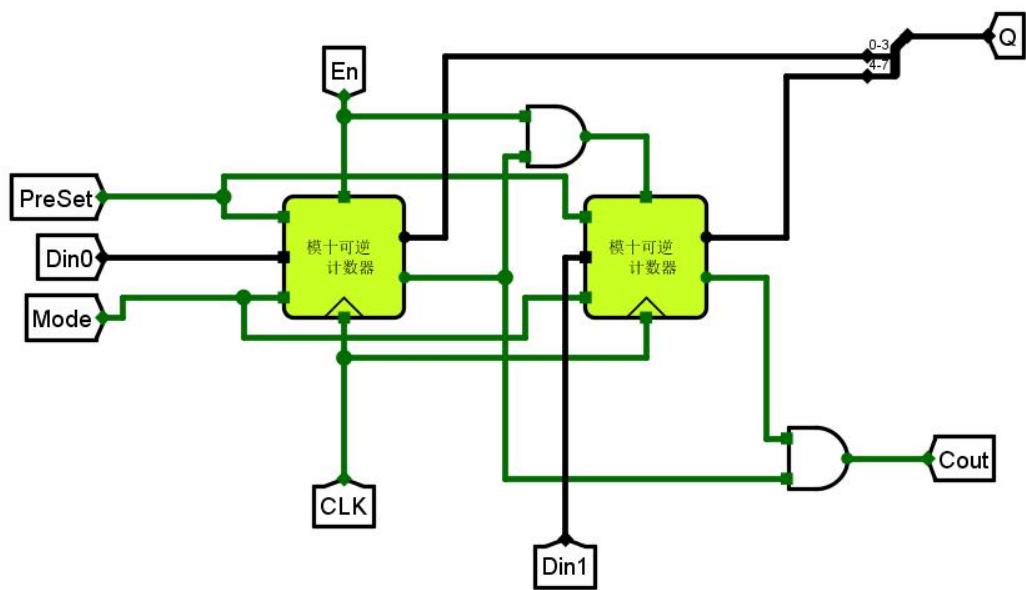


图 3-35 两位十进制可逆计数器电路图

(3) 测试图

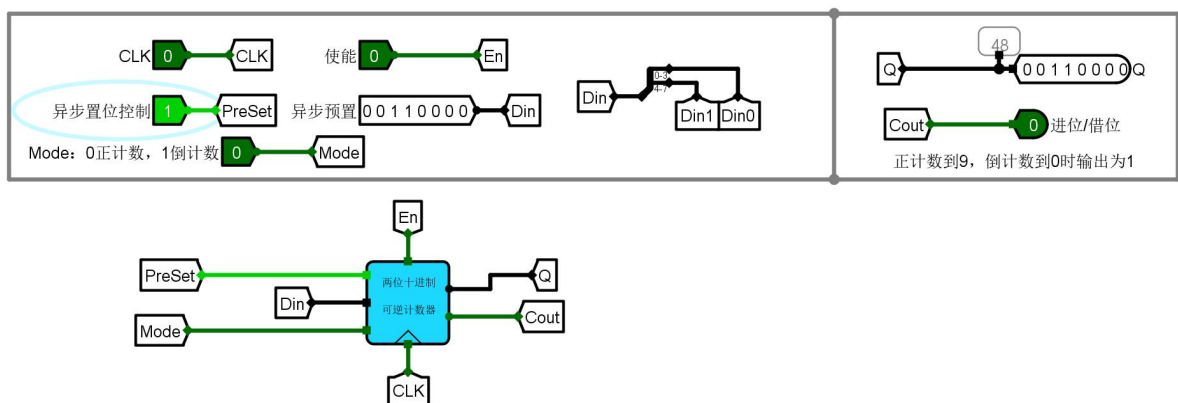


图 3-36 两位十进制可逆计数器测试图

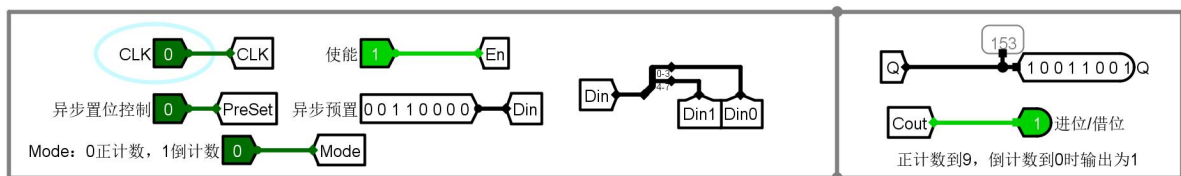


图 3-37 两位十进制可逆计数器测试图（进位）

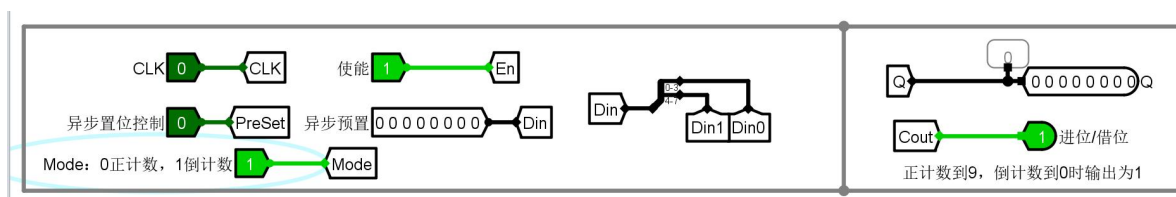


图 3-38 两位十进制可逆计数器测试图（借位）

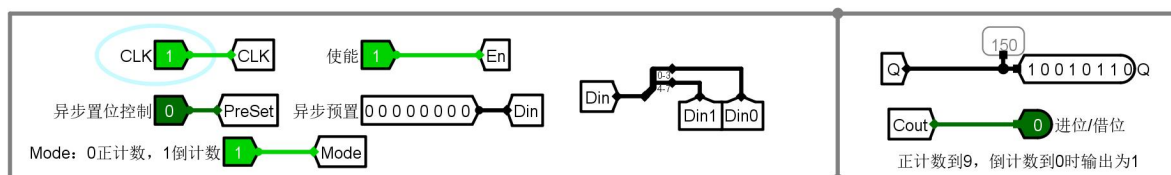


图 3-39 两位十进制可逆计数器测试图

（4）测试分析

- 异步预置：**当 $\text{PreSet}=1$ 时，此时会将 Din 端的数据直接给到输出 Q 。当 $\text{Din}=00110000$ 时，输出 $Q=00110000$ ，和 Din 相同，且此时 $\text{En}=1$ ， CLK 输入了正脉冲（从 0 到 1）， Q 不变，说明异步预置的优先级最高。
- 计数：**当 $\text{PreSet}=0$ ， $\text{En}=1$ 时，此时为正常的计数功能。当 CLK 从 0 变 1 时，便进行一次计数。 $\text{Mode}=0$ 时，此时正向计数， CLK 输入正脉冲； $\text{Mode}=1$ ，此时反向计数， CLK 输入正脉冲。
- 进位/借位：**当 $\text{PreSet}=0$ ， $\text{En}=1$ 时。 $\text{Mode}=0$ ，计数器在正向计数到状态 Q 为 10011001 时，会产生进位信号； $\text{Mode}=1$ ，计数器在反向计数到状态 Q 为 00000000 时，会产生借位信号。

由实验测试结果可以看出实验电路设计正确。

3.6 交通灯状态机

(1) 设计思路及设计过程

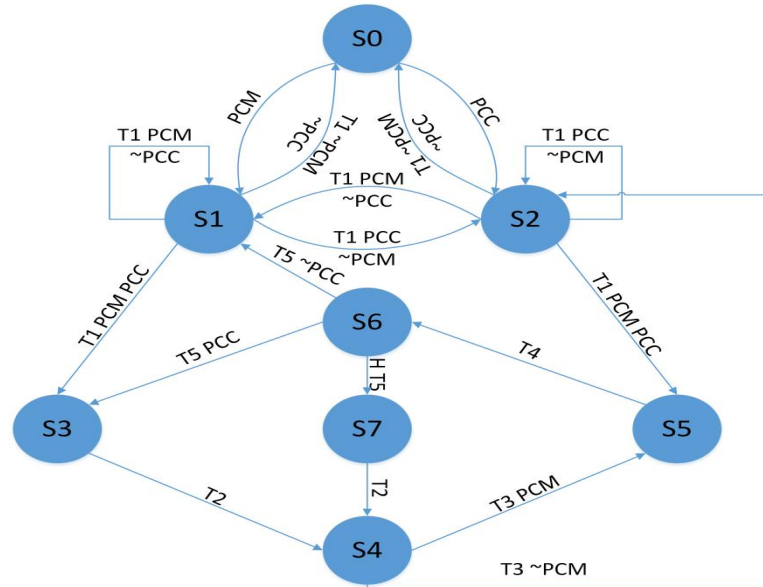


图 3-40 交通灯状态设计图

总体将交通灯的状态设计成 8 个大的状态，S0 代表此时没有通行请求，设置为黄灯闪烁。S1 代表非高峰期主干道单侧通行，绿灯。，S2 代表当前非高峰期次干道单侧通行，绿灯。S3 设置为非高峰期主干道通行，绿灯。S4 代表非高峰期主干道通行，黄灯。S5 代表主次干道交替通行，次干道绿灯状态。S5 代表非高峰期次干道通行，绿灯。S6 代表非高峰期次干道通行，黄灯。S7 代表高峰期主干道通行，绿灯。通过在 Excel 表格中画出状态之间的转换，可以求出对应的状态函数，从而生成对应的电路。

1	输入 (填1或0, 不填为无关项x)											输出 (只填写为1的情况)		
2	S2	S1	S0	H	PCM	PCC	T5	T4	T3	T2	T1	N2	N1	N0
18	1	1	0	0		1	1						1	1
19	1	1	0	1			1					1	1	1
20	1	1	1							1		1		
21	0	0	0		0	0								
22	0	0	0		1	1							1	1
23	0	0	1								0			1
24	0	1	0								0		1	
25	0	1	1							0			1	1
26	1	0	0						0			1		
27	1	1	0				0					1	1	
28	1	1	1							0		1	1	1
29	1	0	1					0				1		1
30														

图 3-41 交通灯状态转移表

输出: ▼

$$\overline{S3} \overline{S1} \overline{S0} \overline{Mode} + \overline{S3} \overline{S1} \overline{S0} Mode + \overline{S3} \overline{S1} S0 \overline{Mode} + \overline{S3} \overline{S2} \overline{S1} \overline{S0} Mode + \overline{S3} \overline{S2} \overline{S1} S0 Mode$$

图 3-42 N1 表达式

输出: ▼

$$\overline{S3} \overline{S2} \overline{S1} \overline{S0} \overline{Mode} + \overline{S3} \overline{S2} \overline{S1} Mode + \overline{S3} \overline{S2} S0 \overline{Mode} + \overline{S3} \overline{S2} S1 \overline{S0} + \overline{S3} \overline{S2} S1 S0 Mode$$

图 3-43 N2 表达式

输出: ▼

$$\overline{S3} \overline{S2} \overline{S1} \overline{S0} \overline{Mode} + \overline{S3} \overline{S2} \overline{S1} S0 \overline{Mode} + \overline{S3} \overline{S2} \overline{S1} S0 Mode + \overline{S3} \overline{S2} \overline{S1} S0 Mode$$

图 3-44 N3 表达式

(2) 电路图

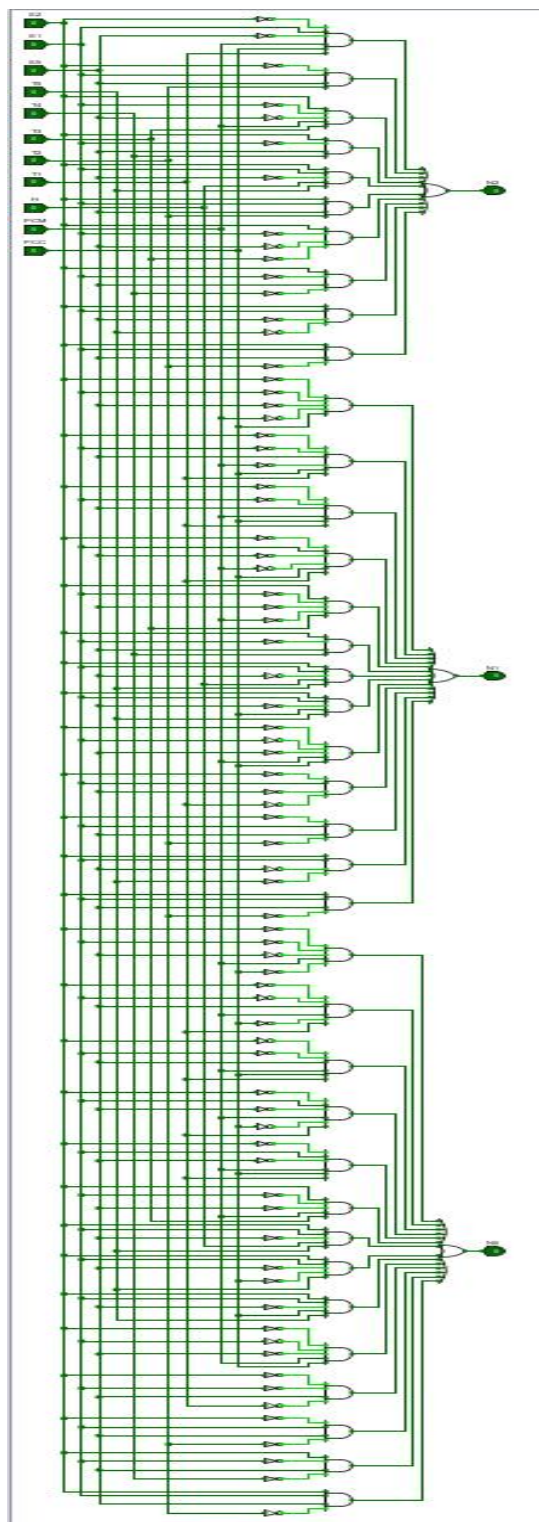


图 3-45 交通灯状态机电路图

(3) 测试图

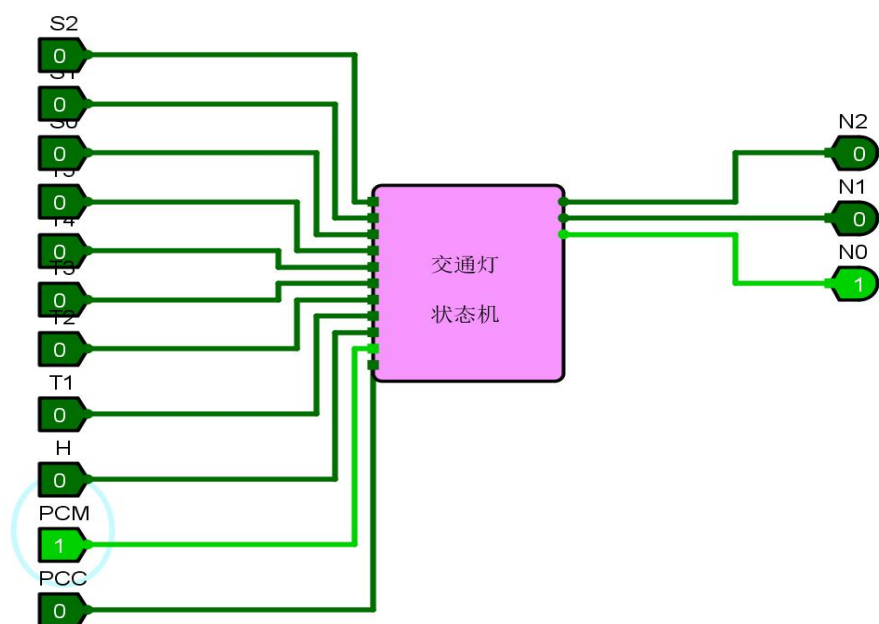


图 3-46 交通灯状态机测试图 (PCM)

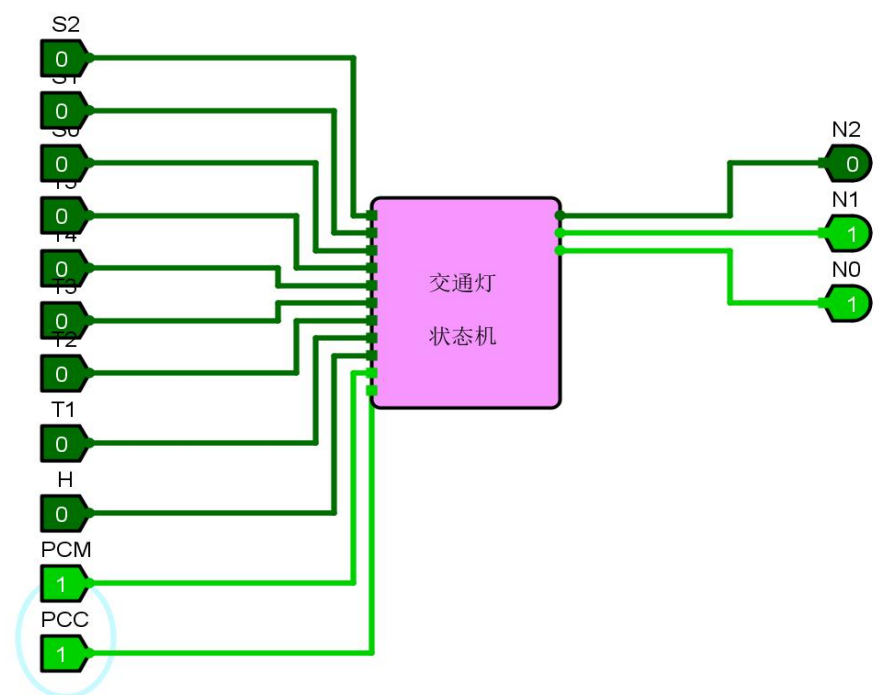


图 3-47 交通灯状态机测试图 (PCM+PCC)

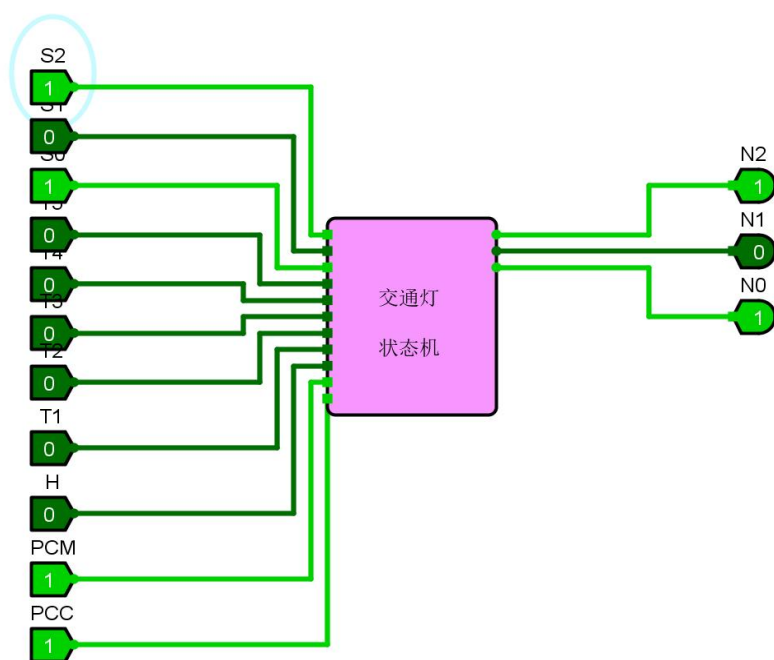


图 3-48 交通灯状态机测试图

(4) 测试分析

测试分析：当 $S_2S_1S_0=000$ 时，代表 S_0 状态，此时 $PCM=1$ 时，状态转移到 S_1 状态，则 $N_2N_1N_0=001$ ；当 $S_2S_1S_0=000$ 时，此时 PCM 和 $PCC=1$ ，状态转移为 011 ；当 $S_2S_1S_0=101$ 时，此时代表 S_5 状态，此时 PCM 和 $PCC=1$ ，状态不变还是 101 ；

由实验测试结果可以看出实验电路设计正确。

3.7 交通灯输出函数设计

(1) 设计思路及设计过程

因为我们在交通灯状态设计中总共设计了八中状态，所以我们只需要三位二进制数就可以表示当前所处的状态了。而对应的每一种状态，我们都知道主干道的灯的情况，由于状态数不多，所以我们可以通过真值表的方式来将状态一一列举出来。

S2	S1	S0	R1	Y1	G1	R2	Y2	G2	PASS1	PASS2	PASS3
0	0	0	0	1	0	0	1	0	0	0	0
0	0	1	0	0	1	1	0	0	0	0	1
0	1	0	1	0	0	0	0	1	0	0	1
0	1	1	0	0	1	1	0	0	1	0	0
1	0	0	0	1	0	1	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	1	0
1	1	0	1	0	0	0	1	0	0	1	0
1	1	1	0	0	1	1	0	0	1	0	0

图 3-49 交通灯状态真值表

(2) 电路图

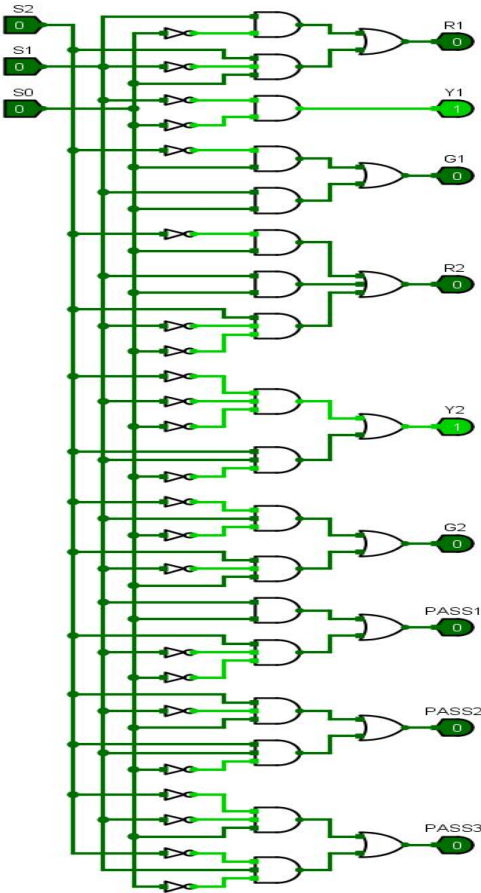


图 3-50 交通灯输出函数电路图

(3) 测试图

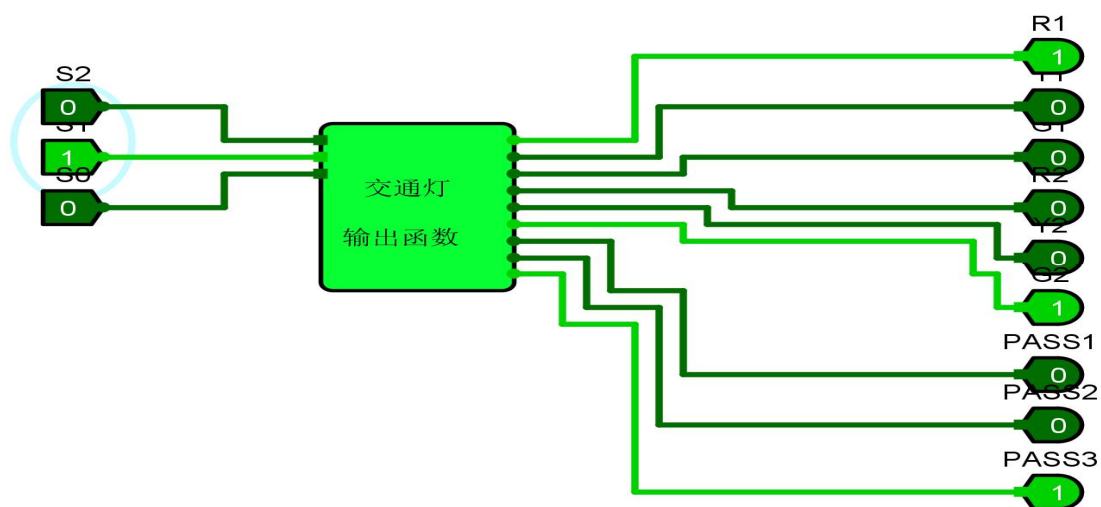


图 3-51 交通灯输出函数测试图 (s1)

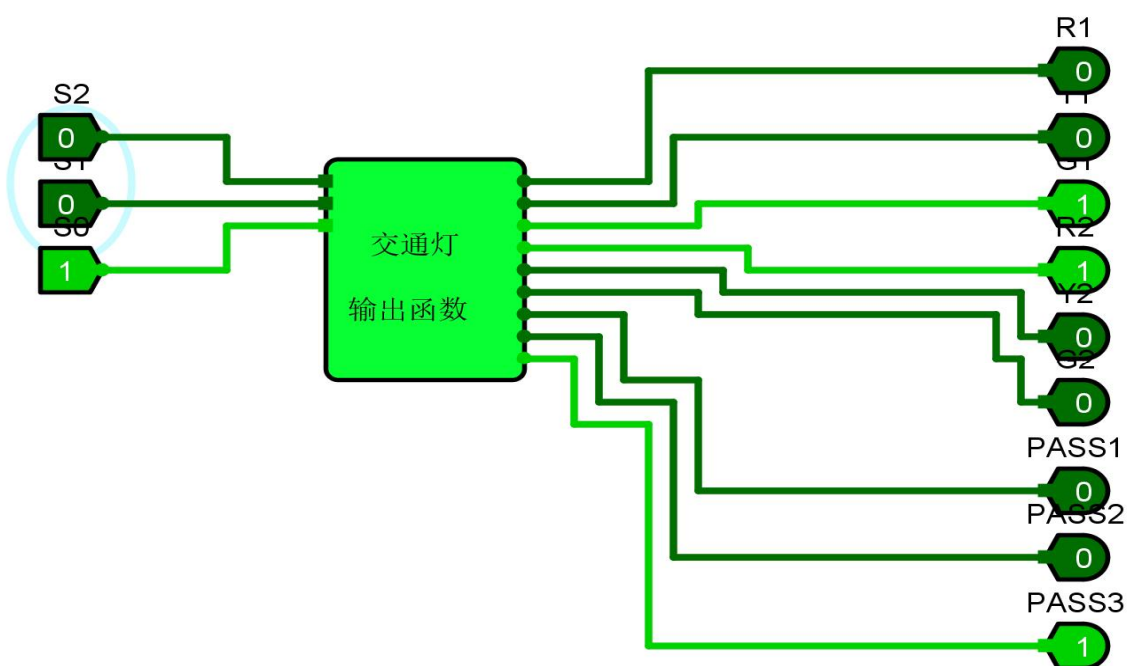


图 3-52 交通灯输出函数测试图 (s0)

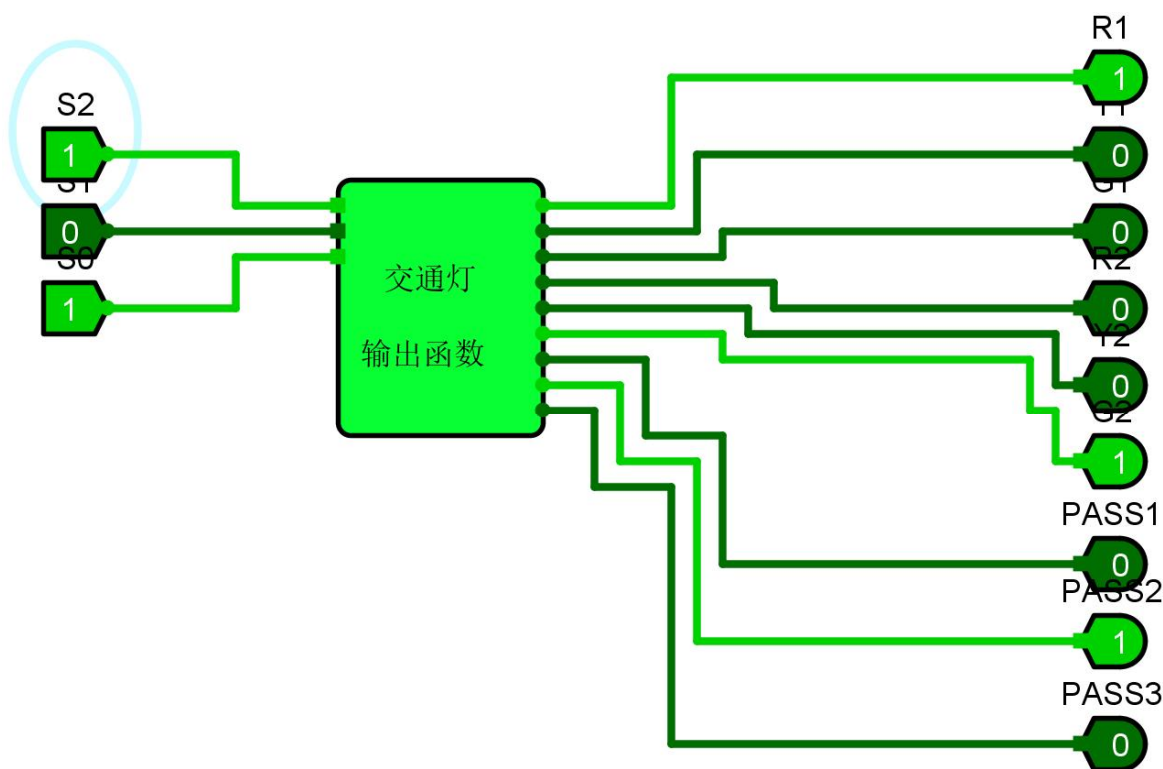


图 3-53 交通灯输出函数测试图

(4) 测试分析

测试分析：当 $S_2S_1S_0=010$ 时，为 S_2 状态，此时次干单独同行， R_1 和 G_2 亮， $PASS_3$ 亮；当 $S_2S_1S_0=001$ 时，为 S_1 状态，由状态表得，此时主干道绿灯，次干道为红灯，且为单侧通行信号。故 G_1 亮， R_2 亮， $PASS_3$ 亮；当 $S_2S_1S_0=101$ 时，为 S_5 状态，此时次干道通行，主干道为红灯，次干道为绿灯。故 R_1 亮， G_2 亮， $PASS_2$ 亮。

由实验测试结果可以看出实验电路设计正确。

3.8 交通灯控制系统

(1) 设计思路及设计过程

通过刚刚设计的交通灯状态机来控制总体的状态变化规律，通过上一步设计的交通的输出函数来选择对应的板块来控制其输出。其中对应每一个板块都有一个独立的两位十进制计数器来对应计数。同时也有 $T_1\sim T_5$ 的输入信号来控制状态变化。并且，设计 $T_6\sim T_8$ 三个特殊的信号，用于初始化两位十进制计数器，作为初

始使能和每次结束时的刷新使能使用，使其能在开始时和一次计数结束后能重新回到对应的初始状态即 16s 计时或者 99s 紧急状态。并且利用之前设计的二路选择器，来应对突发情况。二路选择器可以对应当前状态选择对应状态的计数器来使我们设计的红绿灯输出状态机输出，同时运用两个二路选择器可以做到在三个状态间切换输出，分别是主干道输出，次干道输出，紧急情况。

(2) 电路图

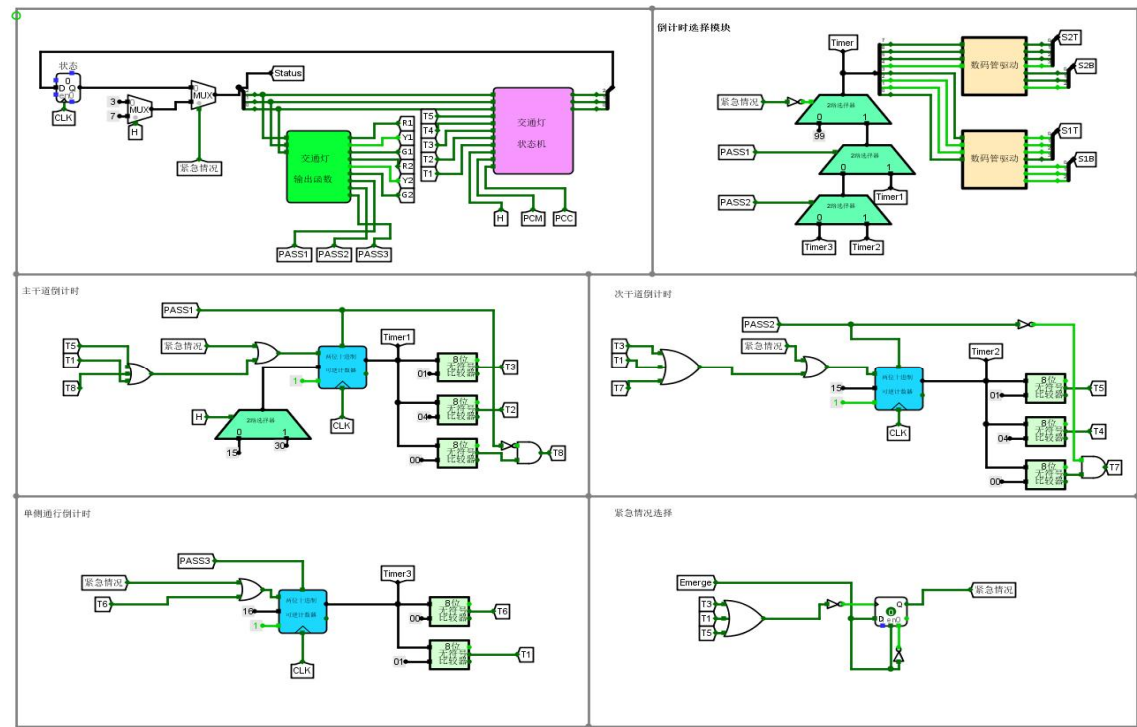


图 3-54 交通灯控制系统电路图

(3) 测试图

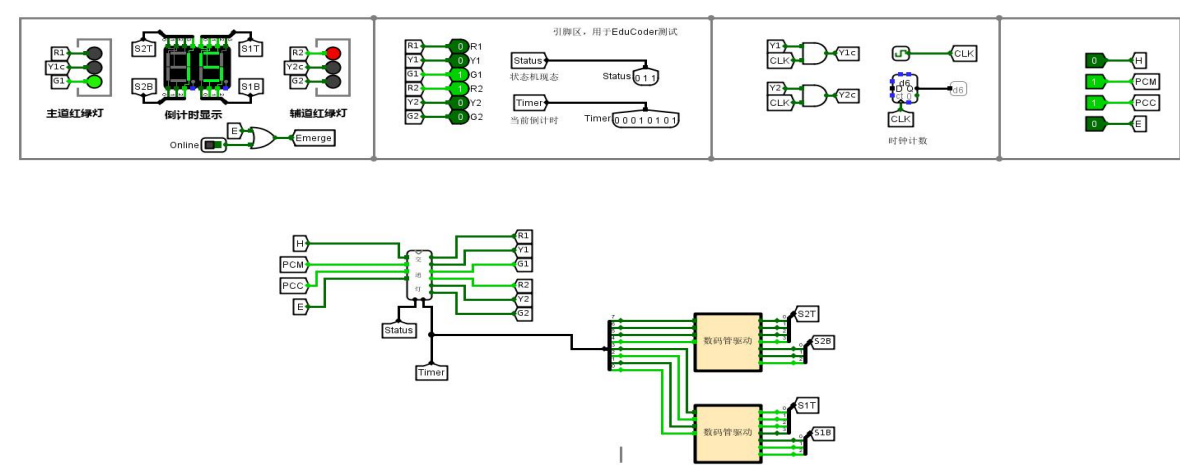


图 3-55 交通灯控制系统测试图（交替通行，主干道绿灯）

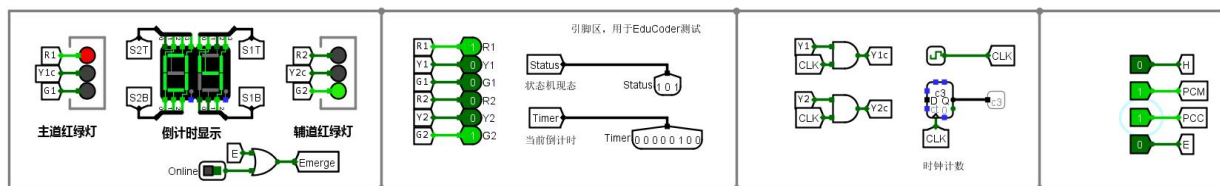


图 3-56 交通灯控制系统测试图（交替通行，次干道绿灯）

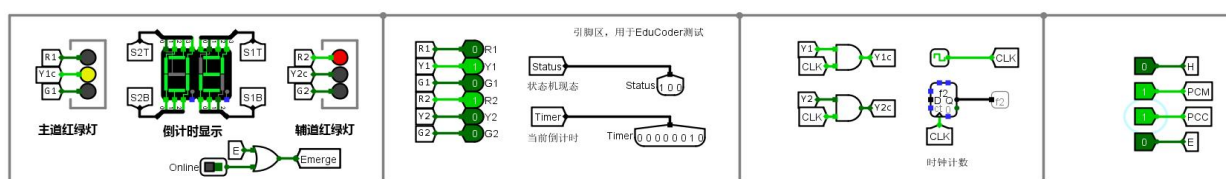


图 3-57 交通灯控制系统测试图（交替通行，主干道黄灯）

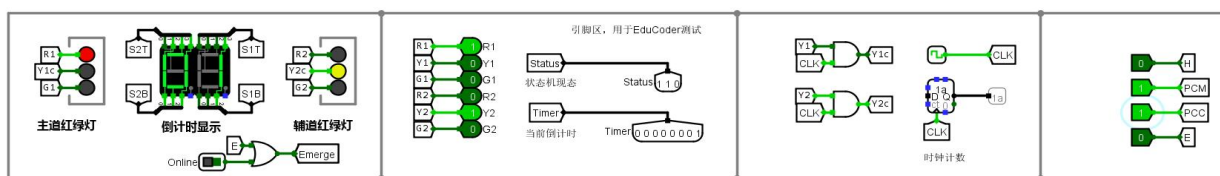


图 3-58 交通灯控制系统测试图（交替通行，次干道黄灯）

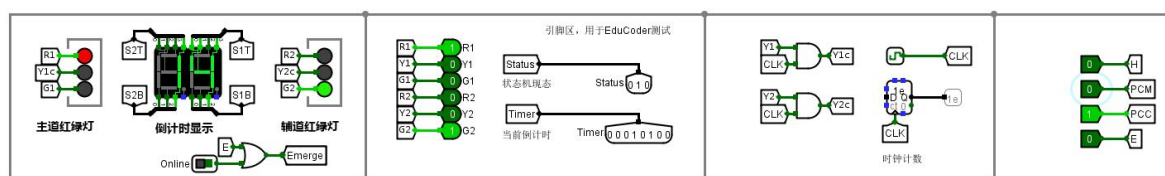


图 3-59 交通灯控制系统测试图（次干道通行）

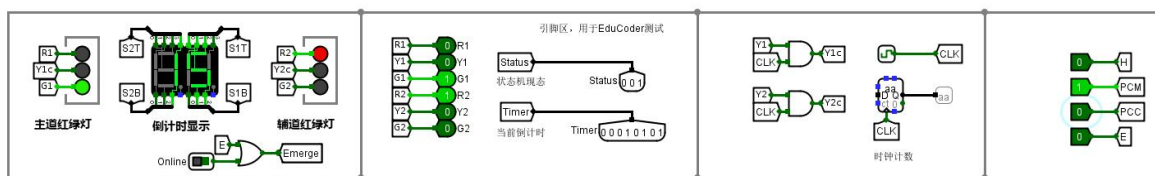


图 3-60 交通灯控制系统测试图（主干道通行）

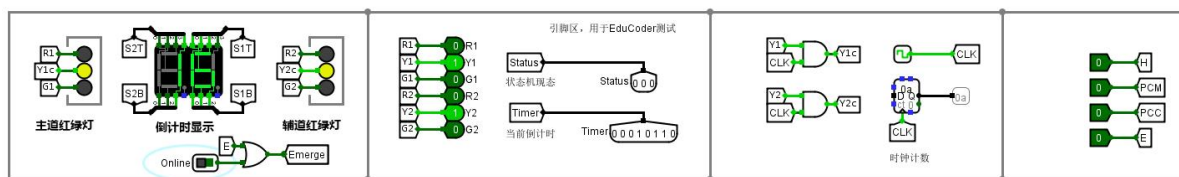


图 3-61 交通灯控制系统测试图（黄灯闪烁）

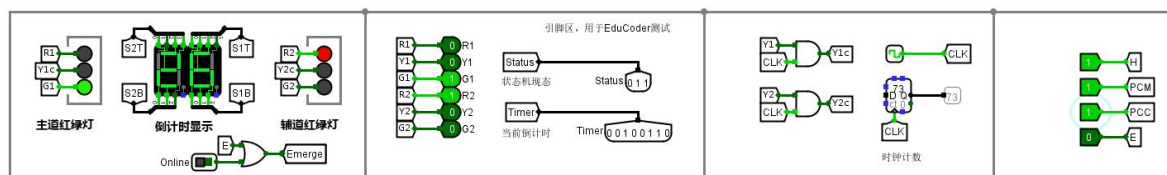


图 3-62 交通灯控制系统测试图（主干道状态）

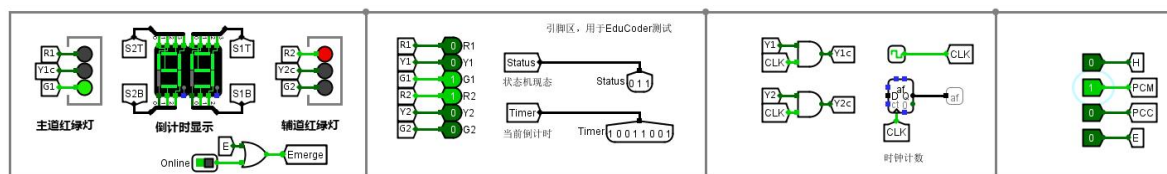


图 3-63 交通灯控制系统测试图（紧急情况）

（4）测试分析

测试分析：当没有任何输入信号时，根据通行规则，此时主干道与次干道黄灯均闪烁，且倒计时恒为 16 秒。当 $H=1$ ， $PCM=1$ ， $PCC=1$ 时，此时主干道次干道交替通行，且主干道通行时间加倍。当 $E=1$ 时，此时为紧急状况，主干道放行，次干道禁止通行，且计时器显示 99。

经过对每一个状态的测试，得出结论电路设计正确。

4 设计总结与心得

4.1 实验总结

4.1.1 遇到的问题及处理

在设计交通灯时我曾经一度以为进入紧急状态就是指直接进入紧急状态而不考虑当前状态是否进行完毕，于是这样就导致我设计交通灯时考虑了许多复杂的地方，思路走偏了，但是在与同学的交流当中我发现了灯当前状态进行完毕再进入紧急状态的话更加合理并且对于我们来说更加容易实现，只需考虑正常的 15s 和 30s 即可。

4.1.2 设计方案存在的不足

(1) 在用真值表设计电路的部分，可能 logisim 自动生成的电路并不是最简单的电路，此时我们其实如果稍加修改可以大大减少电路的复杂度，比如二位无符号比较器之类的器件。

(2) 在考虑可能出现的情况时，在没有通行请求时出现突发状况，红绿灯状态是继续闪烁黄灯，但是当我有通行请求时发现突发状况，红绿灯状态是 99s，当我通行请求结束时，理论上应该回归闪烁黄灯状态，但是实际上还是会保持 99s 的状态不变。相同情况下有两种输出，这是一个不足，其次，由于连线比较复杂，可能有冗余，在应对复杂情况时可能也不一定能及时处理。

4.2 实验心得

首先就是这个实验教会了我使用 logisim，我觉得这是一个很好玩且很实用的工具。其次我觉得做课程设计同时也是对课本知识的巩固和加强，由于课本上的知识太多，平时课间的学习并不能很好的理解和运用各个元件的功能，而且在课本上能学习到的实用的功能有限，所以在这次课程设计过程中，我们了解了很多有用的原件和其设计的方法，也明白了原来无论是多复杂的系统最终都可以划分为数个小的简单的原件，并且对于其在电路中的使用有了更多的认识，就比如看起来复杂的交通灯，最后划分来看也就是几个小的模块而已。认识来源于实践，实践是认识的动力和最终目的，实践是检验真理的唯一标准。

4.3 意见与建议

首先希望对于问题的描述更为清晰，比如对于 15s 和 16s 的区别，这些方面可能有的同学就会较真。其次，头哥上设计的状态图还是略微有点不够完全，比如在次干道通行时有主干道请求，在次干道通行结束后还会再进行一次次干道通行。并且在次干道通行时，有主干道通行请求然后消失，其实实际情况是没有主干道通行请求，但是对应状态图还是会进行一次主干道通行，希望在给以后得同学们设计实验的时候能够尽量减少让人产生歧义的地方，总体看来本实验非常优秀，让我提升了自己的动手能力的同时也对数电理解的更加深刻了。最后还希望能够给学生提供一个交流问题的平台，如 Discuss 区，让同学们能够分享自己的观点和看法。

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：方子豪

最终提交的文件

- (1) 实验电路[电子版];
- (2) 实验报告 [电子版];
- (3) 实验报告[纸质版]。

提交的电子版文件无需压缩，每个学生放在一个文件夹，文件夹及文件命名方式：

班级-学号-姓名。如：信安 2001-U20010101-张三-交通灯实验报告

全班收齐后统一打包压缩交给老师。