

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:	U202215642
姓 名:	田清林
电 话:	15603480922
邮 件:	1240067633@qq.com
完成日期:	2023 年 12 月 3 日

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（1 位、8 位）.....	16
3.4	十进制可逆计数器（包含状态机、输出函数及整体电路）.....	19
3.5	两位十进制可逆计数器.....	26
	图 3.40 两位十进制可逆计数器电路图.....	26
3.6	交通灯状态机.....	28
3.7	交通灯输出函数设计.....	33
3.8	交通灯控制系统.....	37
4	设计总结与心得	41
4.1	实验总结.....	41
4.1.1	遇到的问题及处理.....	41
4.1.2	设计方案存在的不足.....	41
4.2	实验心得.....	42
4.3	意见与建议.....	42

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。

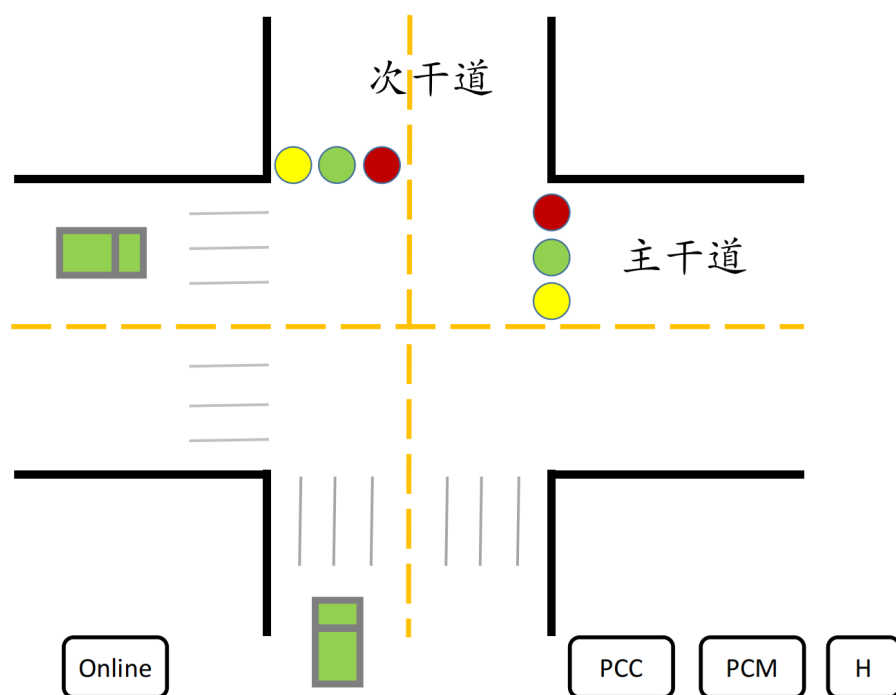


图 1-1 交通灯控制系统示意图

(3) 具体功能

- a. 路口指示灯规则为：“红--绿--黄”循环；
- b. 控制参数假设：红灯 15 秒，绿灯 12 秒，黄灯 3 秒；
- c. 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。
- d. 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”；
- e. 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。
- f. 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。
- g. 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。
- h. 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）确定基本参数和功能需求：

确定交通灯的基本控制参数，如红灯、绿灯、黄灯的时间；

在功能方面，交通灯系统需要有红黄绿三色的红绿灯（分别为主次干道），需要有显示当前倒计时数码管，需要根据不同情况进行正确的红绿灯切换。

（2）分析所需子电路和器件：

采用模块化设计，将系统分为一个个子电路，便于调试和后期维护。分析系统需要用到的子电路，如 8 位无符号比较器、8 位二路选择器、双位 BCD 双向计数器等。了解这些子电路的功能和特性，以便合理运用。

（3）使用状态机描述系统状态：

根据交通灯的工作状态，设计一个状态机，明确不同状态之间的转换关系。可根据通行规则 1 到 6 设计状态（状态至少 6 个）根据规则进行状态转换。

使用状态输出函数描述每个状态下的输出逻辑，例如灯光的控制信号、倒计时控制与结束信号。

（4）引入异常处理和紧急情况处理：

考虑系统可能出现的故障或异常情况，引入相应的异常处理机制；

设计处理紧急信号的方式，确保系统在紧急情况下能够迅速做出响应。

（5）性能测试和调试：

在 logisim 中对系统进行性能测试，模拟不同的输入情况，验证系统的稳定性和可靠性；

2.2 实验总体设计框架

交通灯系统主要分为六个模块：红绿灯状态转移、红绿灯控制、主干道倒计时、次干道倒计时、单侧通行倒计时模块、倒计时选择与显示。各模块介绍如下：

（1）红绿灯状态转移模块：这里根据各输入信号（如主次干道通行请求、高峰期信号等）、电路内反馈信号（如主干道黄灯倒计时结束信号）进行状态的转移，并输出控制红绿灯亮灯颜色的信号。

(2) 红绿灯控制模块：根据红绿灯状态转移模块输出的状态控制两个红绿灯各颜色灯的亮灭。

(3) 主干道倒计时模块：在主次干道交替通行时，负责主干道的倒计时，根据是否高峰期设置倒计时从 15 或 30 开始，每一秒（即一个时钟周期）倒计时减 1，计时到最后三秒和计时结束时分别输出信号，作为红绿灯状态转移和输出模块的输入。

(4) 次干道倒计时模块：在主次干道交替通行时，负责次干道的倒计时，倒计时从 15 开始，每一秒（即一个时钟周期）倒计时减 1，计时到最后三秒和计时结束时分别输出信号，作为红绿灯状态转移和输出模块的输入。

(5) 单侧通行倒计时模块：在主次干道交替通行（主干道或次干道只有一方有同行请求）时，负责通行路段的倒计时，倒计时从 16 开始，每一秒（即一个时钟周期）倒计时减 1，计时结束时分别输出信号，作为红绿灯状态转移和输出模块的输入。

(6) 倒计时显示模块：由于在单侧通行、主次干道交替通行、高峰期和紧急状况时的倒计时不同，这个模块接受以上三个倒计时模块的计时值，在不同的情况下应该显示的倒计时值传给数码管驱动。要注意在紧急状况下显示的应为 99 秒不变。

各模块之间关系如图 2.1 所示。

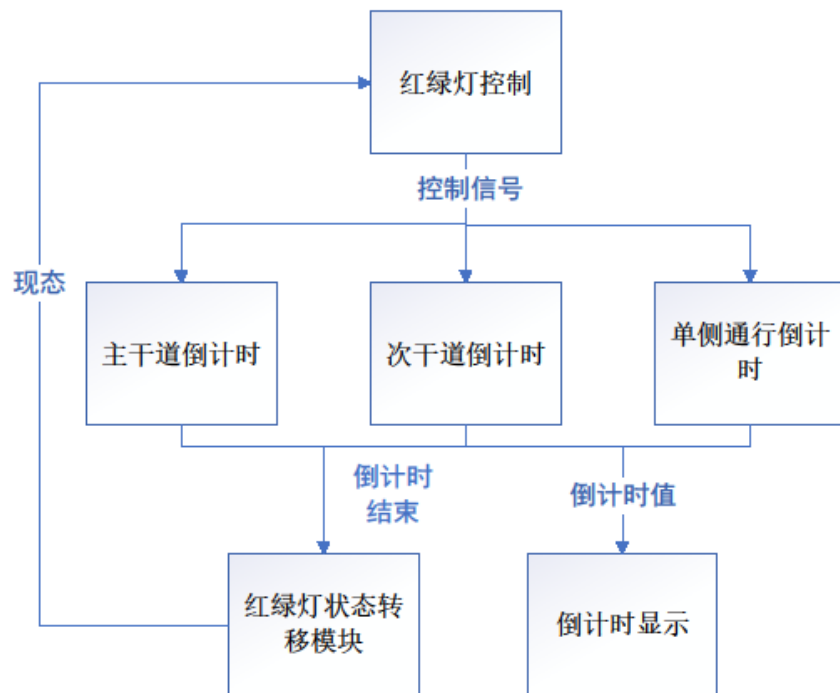


图 2.1 交通灯各模块间关系

交通灯状态设计如图 2.2，状态转换图如图 2.3 所示。

状态编号	状态描述
S0	主、次干道均为黄灯闪烁
S1	非高峰期主干道单侧通行，绿灯。
S2	非高峰期次干道单侧通行，绿灯。
S3	非高峰期主干道通行，绿灯。
S4	非高峰期主干道通行，黄灯。
S5	非高峰期次干道通行，绿灯。
S6	非高峰期次干道通行，黄灯。
S7	高峰期主干道通行，绿灯。

图 2.2 交通灯状态设计

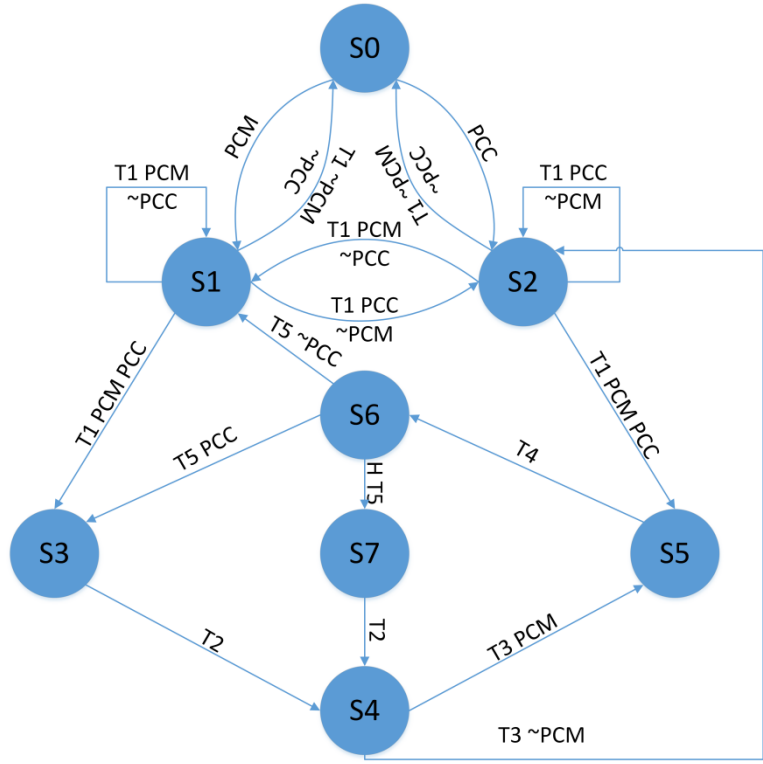


图 2.3 交通灯状态转换图

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

根据七段数码管引脚根据十进制与七段数码管的对应关系（如图 3.1 所示）对真值表进行编辑，最后使用真值表生成电路，真值表如图 3.2 所示。

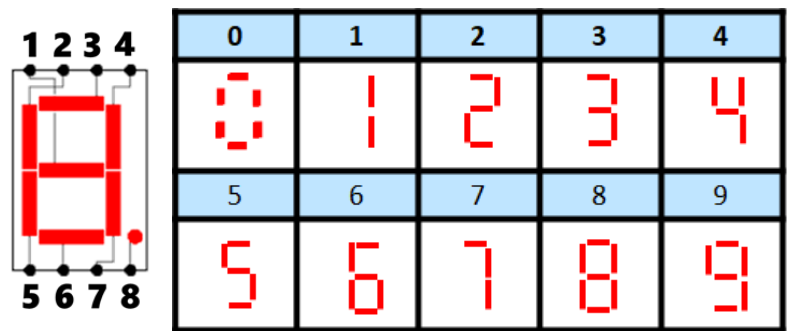


图 3.1 十进制与数码管的对应关系

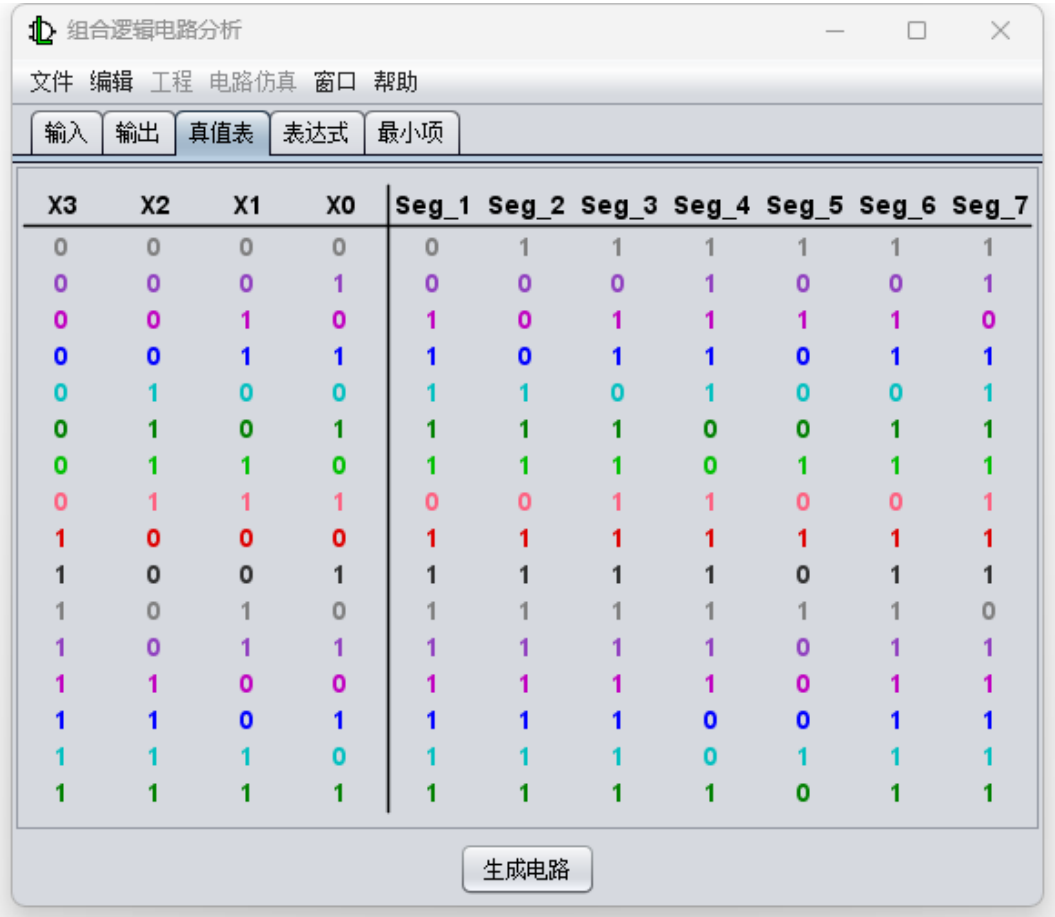


图 3.2 七位数码管真值表

(2) 电路图

如图 3.3 所示。

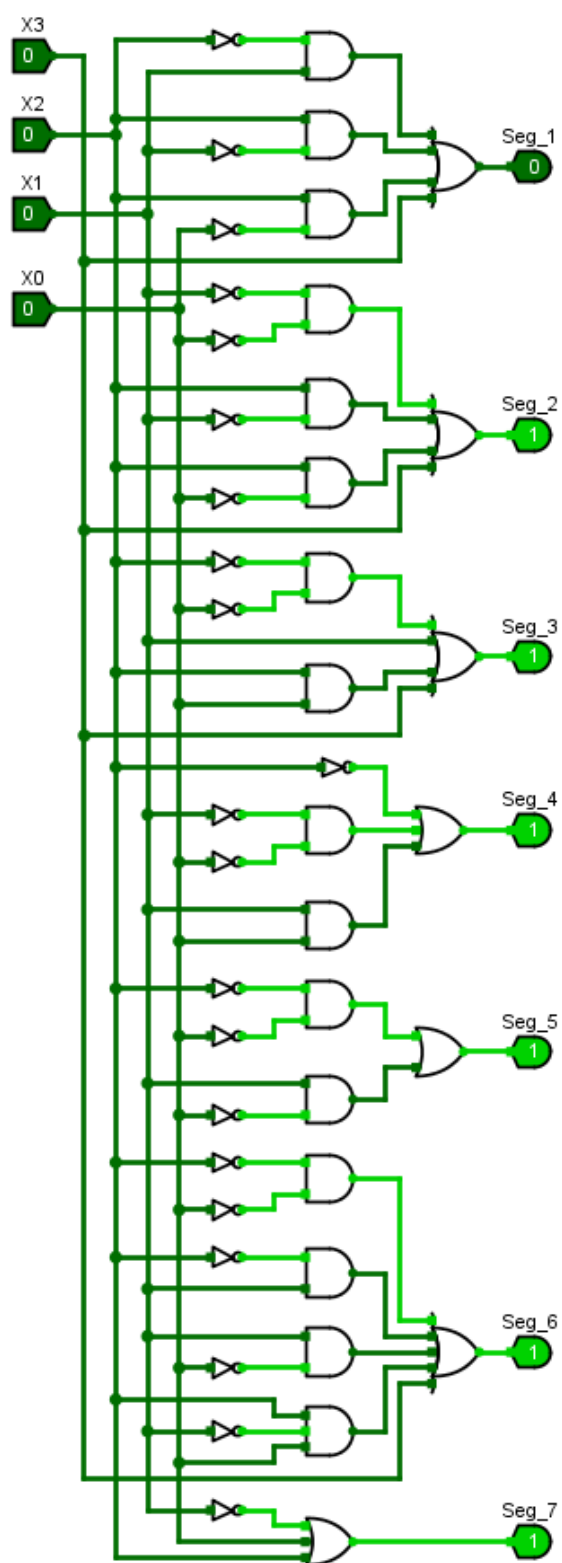


图 3.3 七位数码管驱动电路图

(3) 测试图

如图 3.4、图 3.5、图 3.6。

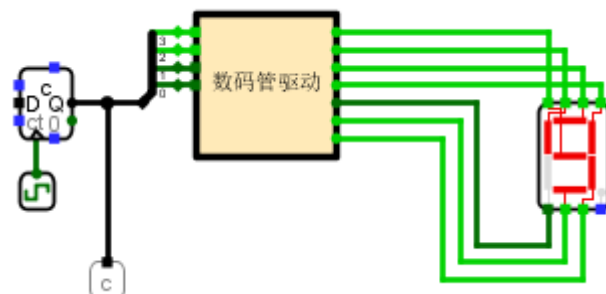


图 3.4 七位数码管驱动电路测试

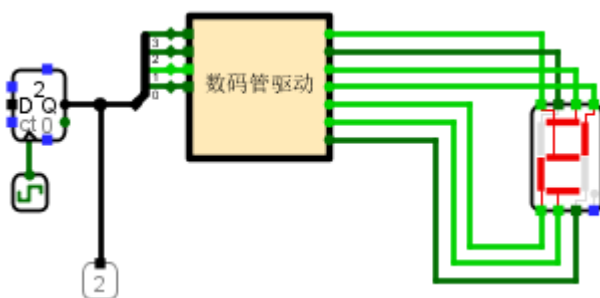


图 3.5 七位数码管驱动电路测试

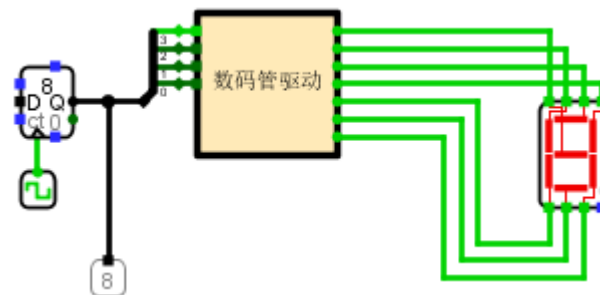


图 3.6 七位数码管驱动电路测试

(4) 测试分析

电路可以正确地显示 0 到 9 的数字，手动测试无误，提交平台测试通过。

3.2 无符号比较器（2 位、4 位、8 位）

(1) 设计思路及设计过程

2 位无符号比较器可以直接写出真值表，如图 3.7 所示。

X1	X0	Y1	Y0	Great	Equal	Less
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

图 3.7 2 位无符号比较器真值表

利用两个 2 位比较器设计四位比较器。分别用两个 2 位比较器比较高位和低位，并进行分析，当 X 的高位大于 Y 的高位或 X 高位等于 Y 的高位且 X 低位大于 Y 的高位时有 $X > Y$ ，当 X 和 Y 的高低位分别对应相等时有 $X == Y$ ，当 X 的高位小于 Y 的高位或 X 高位等于 Y 的高位且 X 低位小于 Y 的高位时有 $X < Y$ 。根据以上分析使用逻辑门并进行连线。

8 位比较器与 4 位比较器的设计类似，利用两个 4 位比较器实现，只是要注意这里要利用分线将两个四位的输入转化为八个一位输入。

(2) 电路图

2 位、4 位、8 位无符号比较器分别如图 3.8、图 3.9、图 3.10 所示。

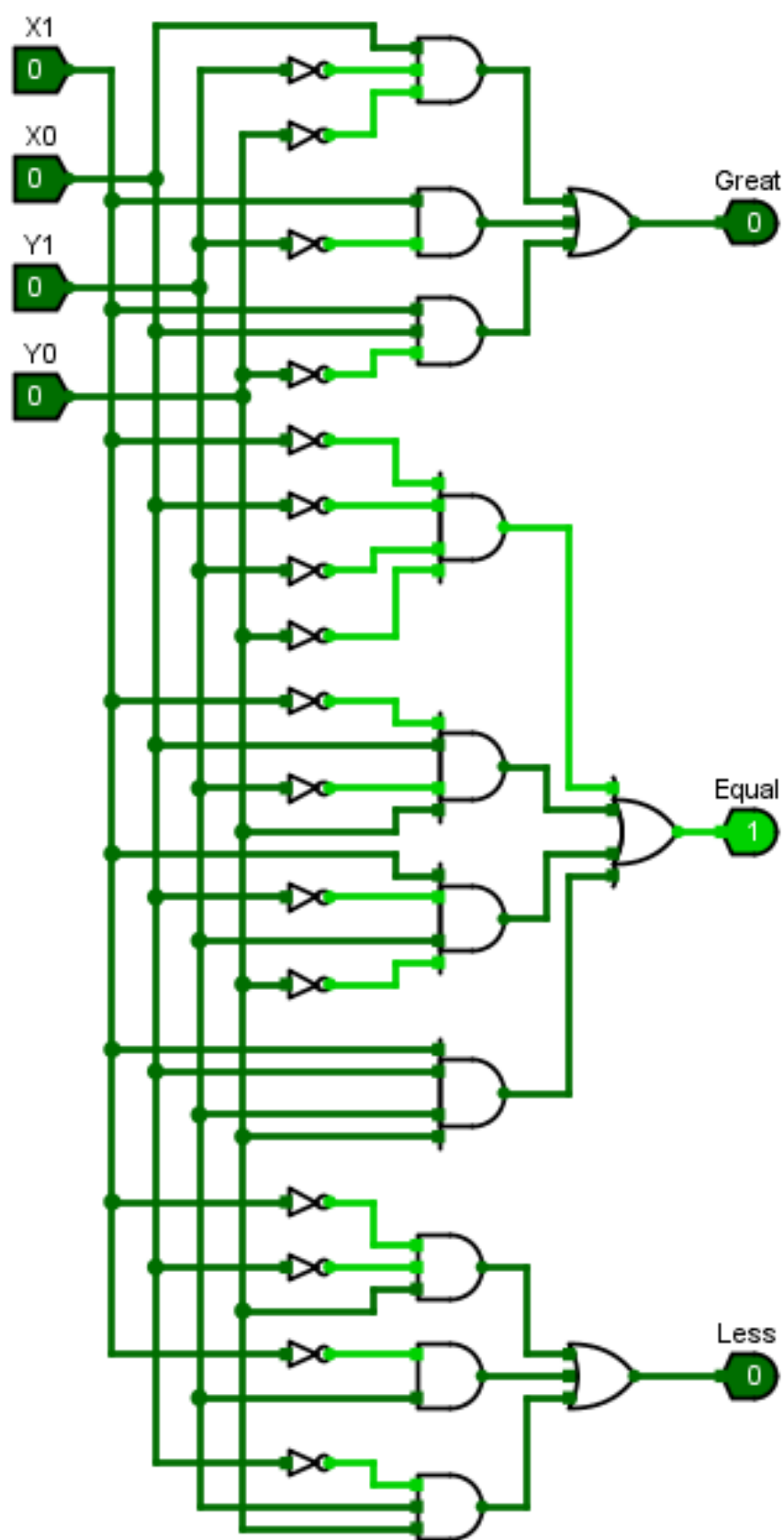


图 3.8 2 位无符号比较器电路图

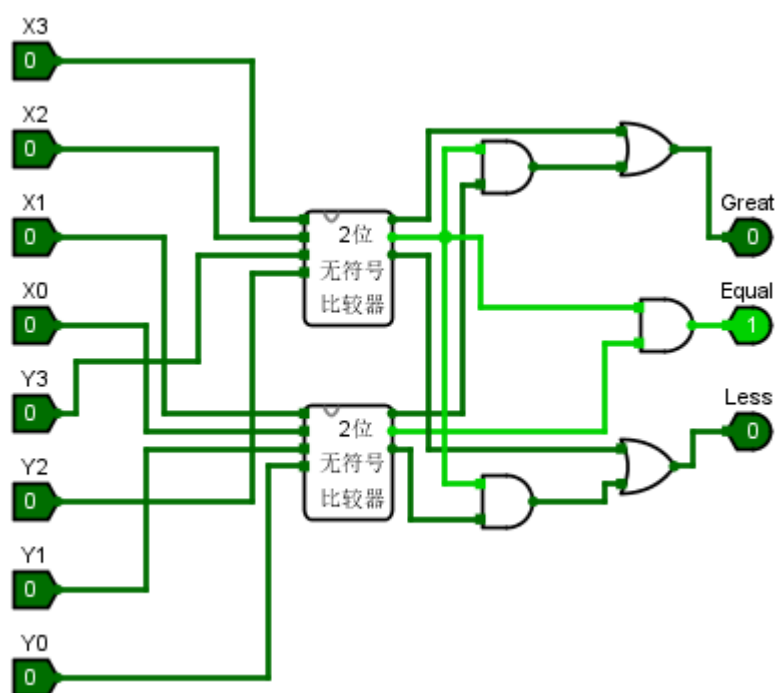


图 3.9 4 位无符号比较器电路图

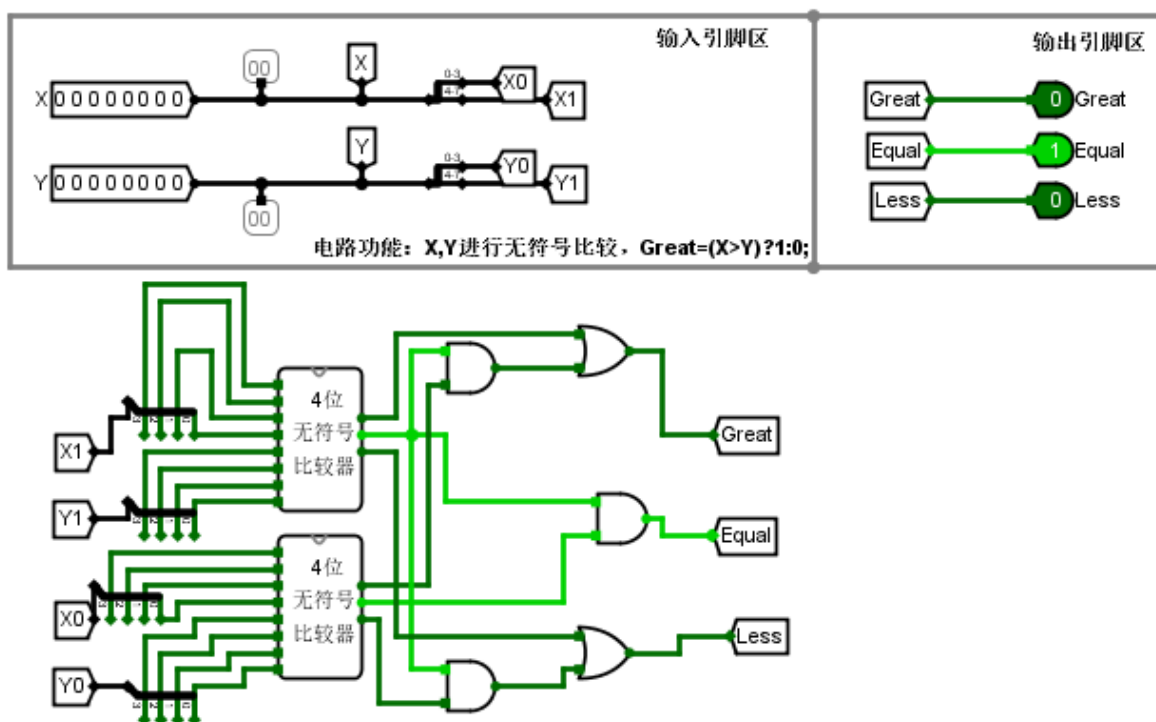


图 3.10 8 位无符号比较器电路图

(3) 测试图

对 2 位、4 位、8 位无符号比较器分别选取 $X>Y$ 、 $X==Y$ 、 $X<Y$ 的三个样例。2 位无符号比较器测试图如图 3.11、图 3.12、图 3.13 所示，4 位无符号比较器测试图如图

3.14、图 3.15、图 3.16 所示，8 位无符号比较器测试图如图 3.17、图 3.18、图 3.19 所示。

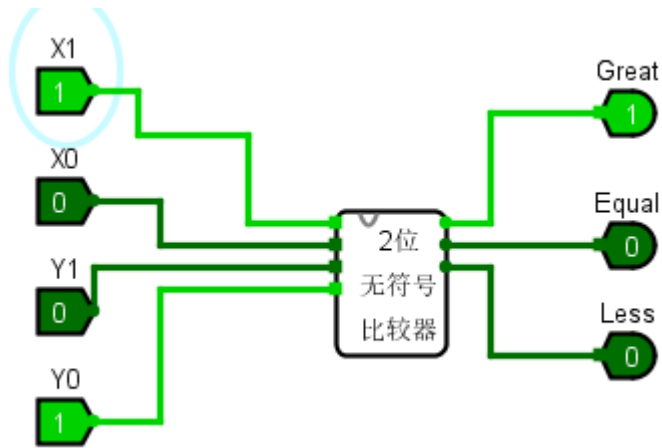


图 3.11 2 位无符号比较器测试图 ($X > Y$)

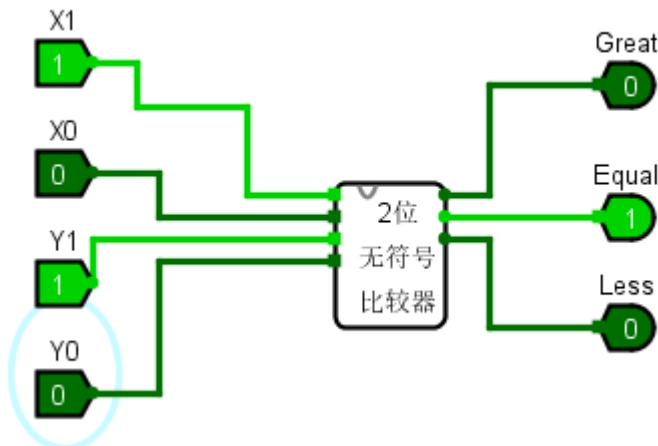


图 3.12 2 位无符号比较器测试图 ($X == Y$)

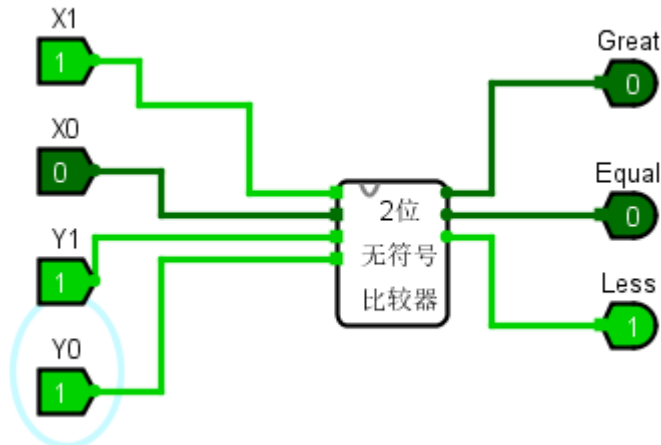


图 3.13 2 位无符号比较器测试图 ($X < Y$)

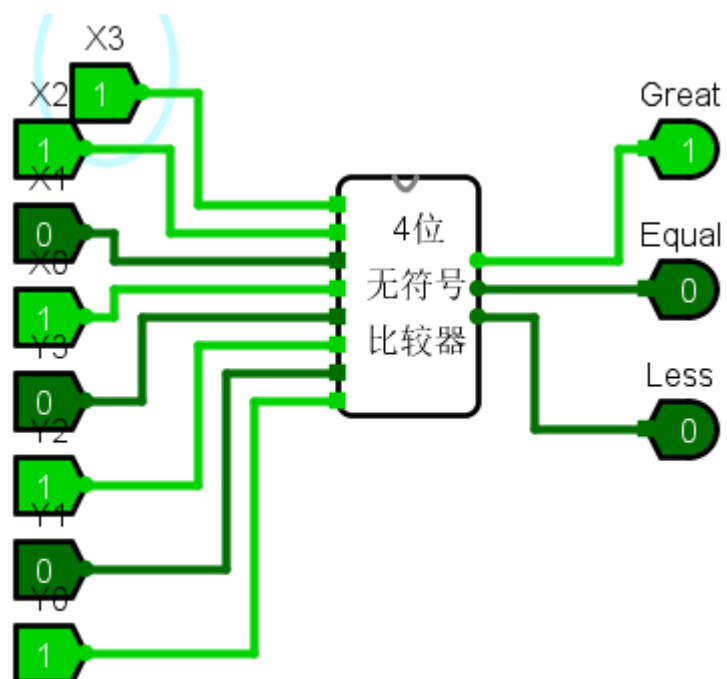


图 3.14 4 位无符号比较器测试图 ($X > Y$)

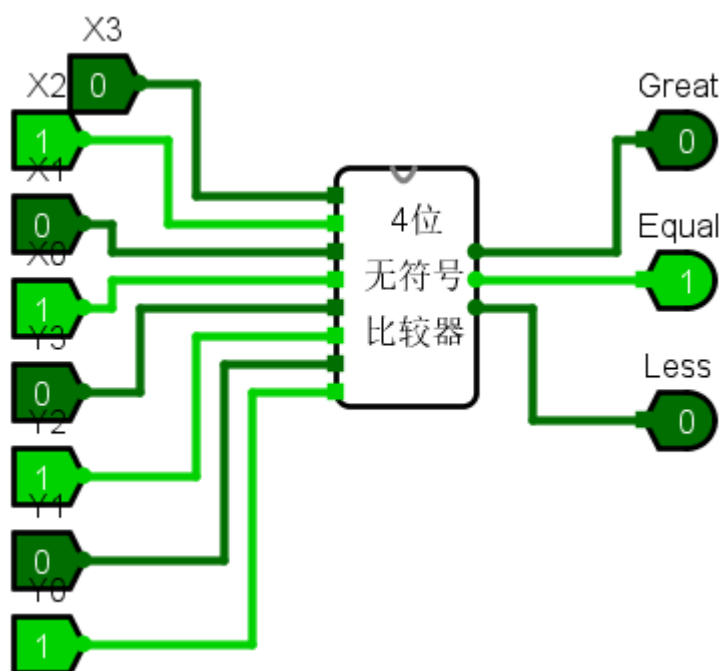


图 3.15 4 位无符号比较器测试图 ($X == Y$)

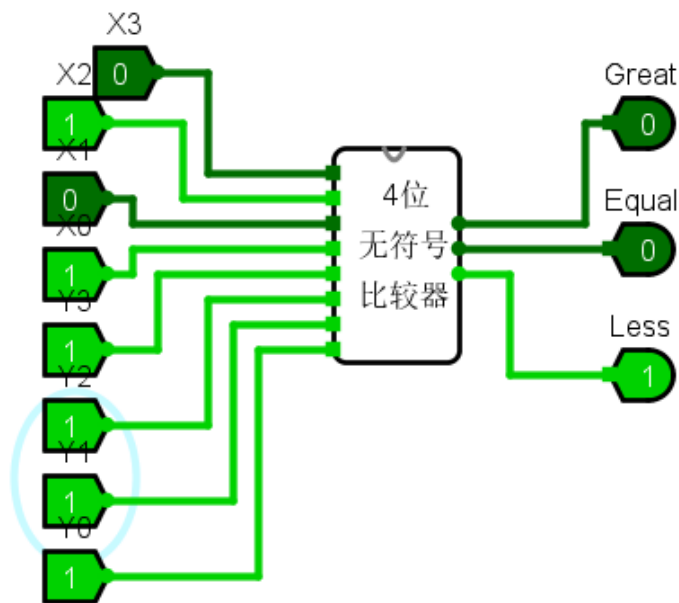


图 3.16 4 位无符号比较器测试图 ($X < Y$)

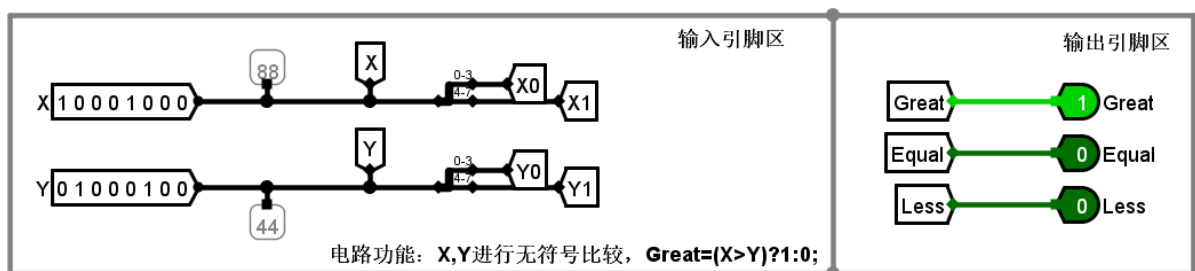


图 3.17 8 位无符号比较器测试图 ($X > Y$)

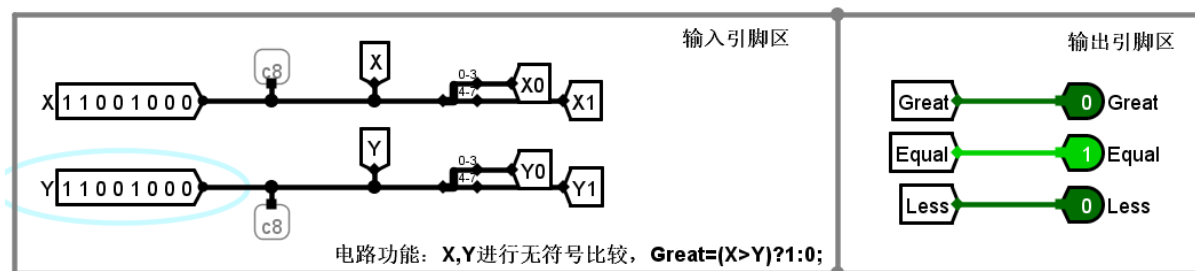


图 3.18 8 位无符号比较器测试图 ($X == Y$)

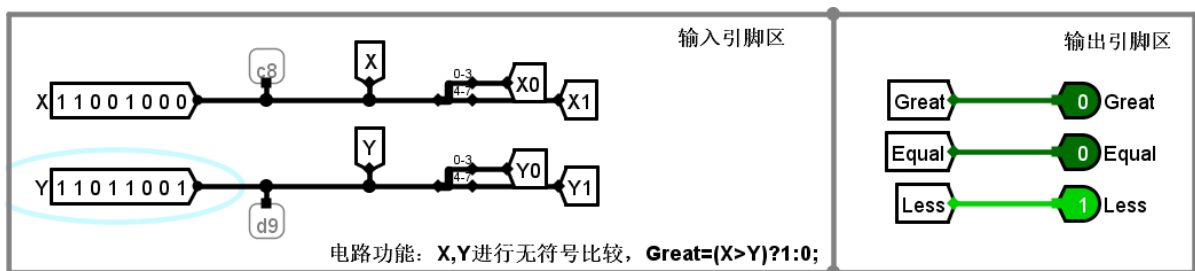


图 3.19 8 位无符号比较器测试图 ($X < Y$)

(4) 测试分析

2 位无符号比较器、4 位无符号比较器、8 位无符号比较器分别在 $X>Y$, $X==Y$, $X<Y$ 三种情况下给出正确输出 (Great、Equal、Less 分别对应为 1)，符合设计要求。

3.3 2 选 1 选择器设计 (1 位、8 位)

(1) 设计思路及设计过程

由设计要求当 $Sel==0$ 时把 $X0$ 输出给 Out ， $Sel==1$ 时把 $X1$ 输出给 Out 得到逻辑表达式 $Sel X1 + X0 \sim Sel$ ，考虑到有险象，所以通过增加同余项 $X1 X0$ ，得到最终逻辑表达式 $Sel X1 + X0 \sim Sel + X1 X0$ ，根据逻辑表达式进行电路连接。

使用 8 个 1 位 2 选 1 选择器，对输入的 8 位的 X 和 Y 使用分线器进行 8 个分线，每一个分线对应一个 2 选 1 选择器， Sel 接分线器的选择端，当 $Sel==0$ 时送出 X ，否则送出 Y ，送出的 8 位二进制再使用分线器进行汇聚给 Out 输出。正确连接电路。

(2) 电路图

1 位、8 位 2 选 1 选择器电路图分别如图 3.20、图 3.21 所示。

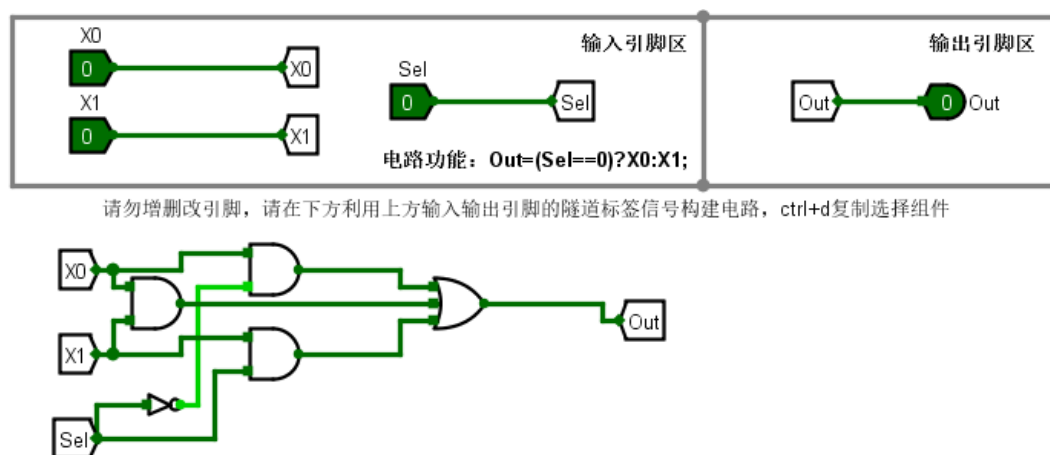
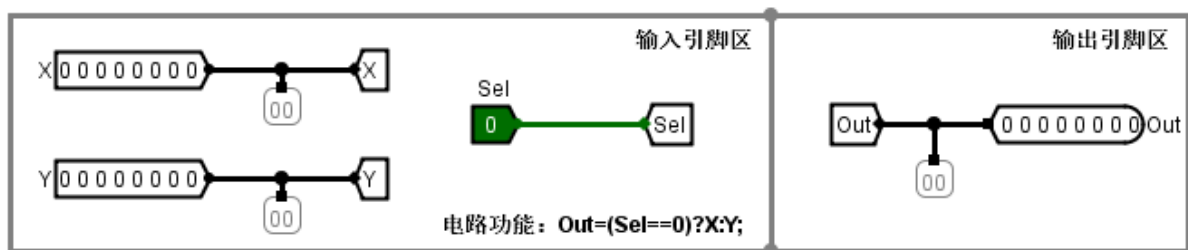


图 3.20 1 位 2 选 1 选择器设计电路



请勿增删引脚，请在下方利用上方输入输出引脚的隧道标签信号构建电路，ctrl+d复制选择组件

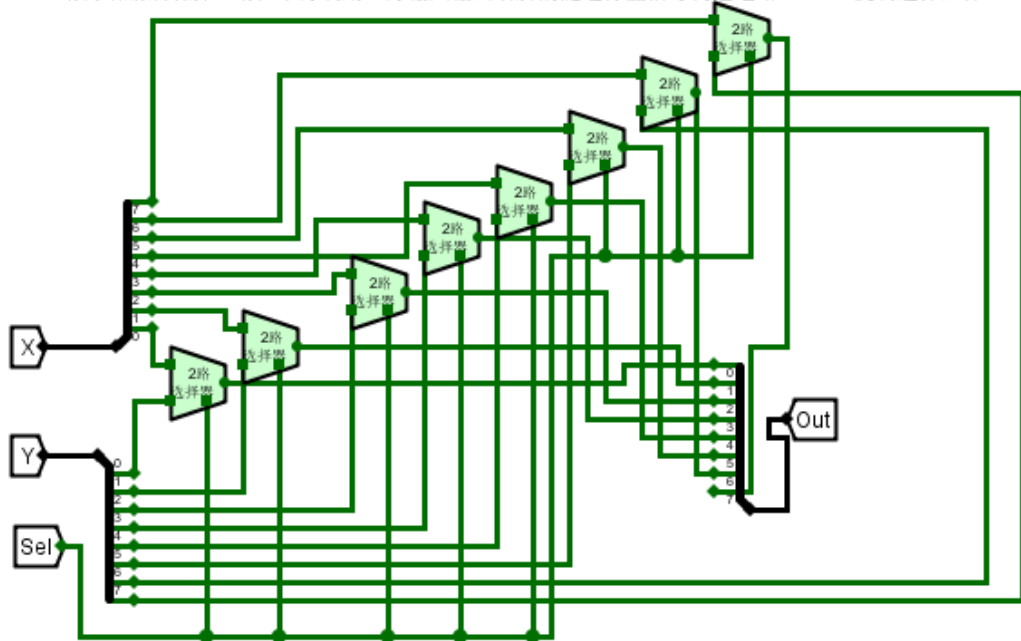


图 3.21 8 位 2 选 1 选择器设计电路

(3) 测试图

1 位 2 选 1 测试如图 3.22、图 3.23 所示，8 位测试如图 3.24、图 3.25 所示。

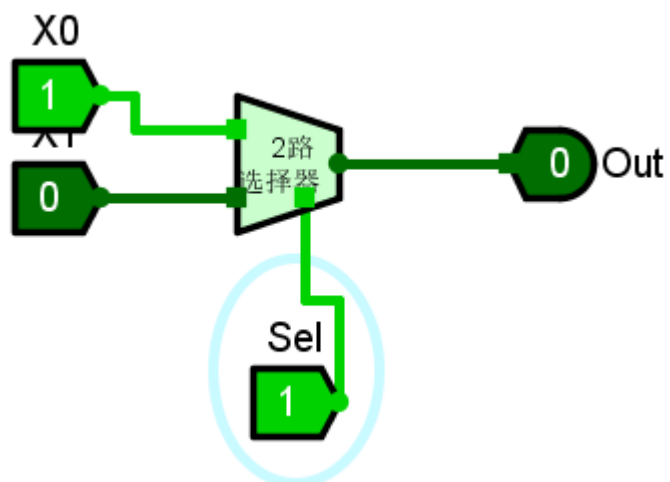


图 3.22 1 位 2 选 1 选择器测试 ($\text{Sel} == 1$)

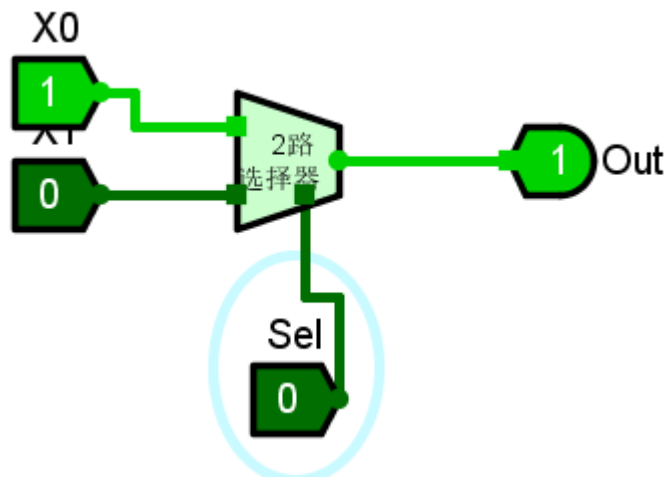
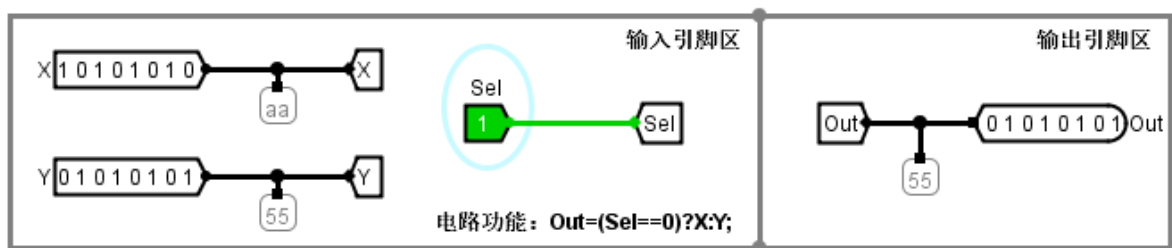


图 3.23 1 位 2 选 1 选择器测试 (Sel==0)



请勿增删改引脚, 请在下方利用上方输入输出引脚的隧道标签信号构建电路, ctrl+d 复制选择组件

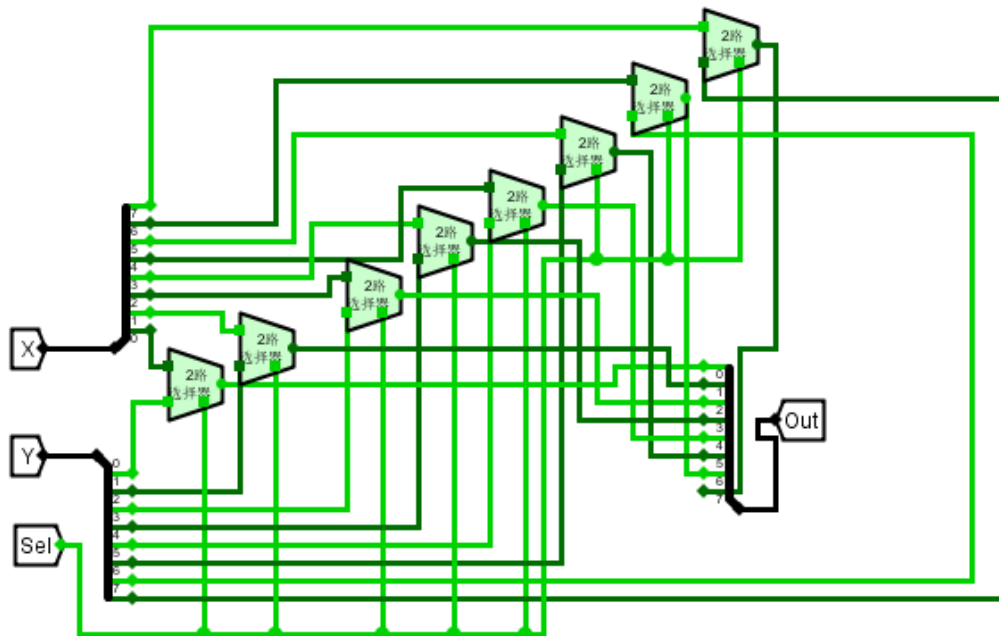
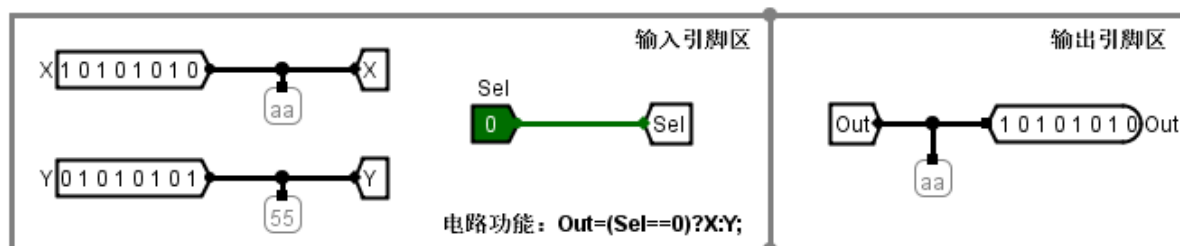


图 3.24 8 位 2 选 1 选择器测试 (Sel==1)



请勿增删改引脚，请在下方利用上方输入输出引脚的隧道标签信号构建电路，ctrl+d复制选择组件

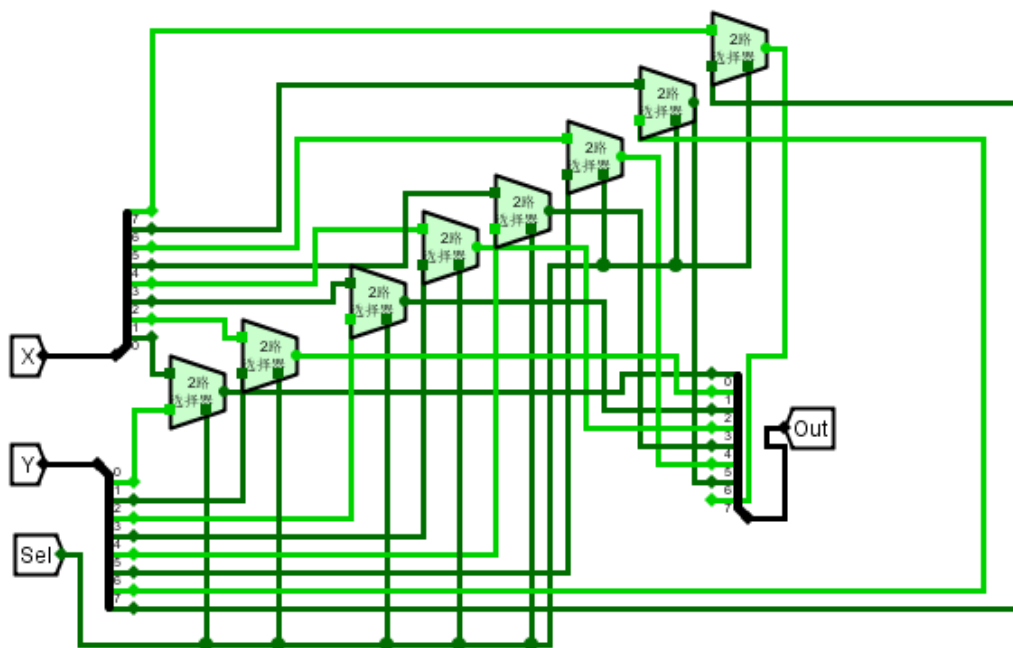


图 3.25 8 位 2 选 1 选择器测试 (Sel==0)

(4) 测试分析

当 Sel=1 时输出 X1，Sel=1 时输出 X0，2 位和 8 位 2 选 1 选择器都准确的执行了选择功能。

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

状态机电路：S3,S2,S1,S0 为 8421 码表示当前计数器数值，N3,N2,N1,N0 表示次态计数器数值，Mode=0 正向计数，Mode =1 反向计数，注意正向计数时 9->0，反向计数时 0->9，填写状态转换表，并生成函数。状态转换表如下。

当前状态(现态)					输入信号							下一状态 (次态)				
S3	S2	S1	S0	现态 10进制	Mode							次态 10进制	N3	N2	N1	N0
0	0	0	0	0	0							1	0	0	0	1
0	0	0	1	1	0							2	0	0	1	0
0	0	1	0	2	0							3	0	0	1	1
0	0	1	1	3	0							4	0	1	0	0
0	1	0	0	4	0							5	0	1	0	1
0	1	0	1	5	0							6	0	1	1	0
0	1	1	0	6	0							7	0	1	1	1
0	1	1	1	7	0							8	1	0	0	0
1	0	0	0	8	0							9	1	0	0	1
1	0	0	1	9	0							0	0	0	0	0
0	0	0	0	0	1							9	1	0	0	1
0	0	0	1	1	1							0	0	0	0	0
0	0	1	0	2	1							1	0	0	0	1
0	0	1	1	3	1							2	0	0	1	0
0	1	0	0	4	1							3	0	0	1	1
0	1	0	1	5	1							4	0	1	0	0
0	1	1	0	6	1							5	0	1	0	1
0	1	1	1	7	1							6	0	1	1	0
1	0	0	0	8	1							7	0	1	1	1
1	0	0	1	9	1							8	1	0	0	0

图 3.26 十进制可逆计数器状态机真值表

输出函数：根据当前状态 S3,S2,S1,S0 和 Mode,产生输出 Cout。当正向计数到 9 或反向计数到 1 时输出 1，填写真值表并自动生成电路。

整体电路：各部件已设计完毕，利用状态转换、输出函数的电路和 4 个边缘 D 触发器实现整体的电路，只需将对应输入输出接好。将 Mode 接到状态转换和输出函数对应位置；D3~D0 输入为异步置位端，将它们连接到对应 D 触发器的置 1 端和置 0 端（置零端连接前需要加上非门）；时钟信号 CLK 连接 D 触发器对应位置，使能端 En 连接至元件对应位置；D 触发器的输出 Q 分别连接到状态转换和输出函数，输出函数的 cout 端与 Cout 连接。

Rst 异步复位直接接 D 触发器的置 0 端，CLK 接触发器的时钟端，EN 接触发器的使能端，4 个触发器现态 Q 输出给封装好的输出函数与输出的 Q，输出函数接 Cout，同时反馈给封装好的状态转换，通过状态转换输出的 4 位数据分别接触发器的 D 端。

（2）电路图

状态机、输出函数、整体电路图依次分别如图 3.27、图 3.28、图 3.29 所示。

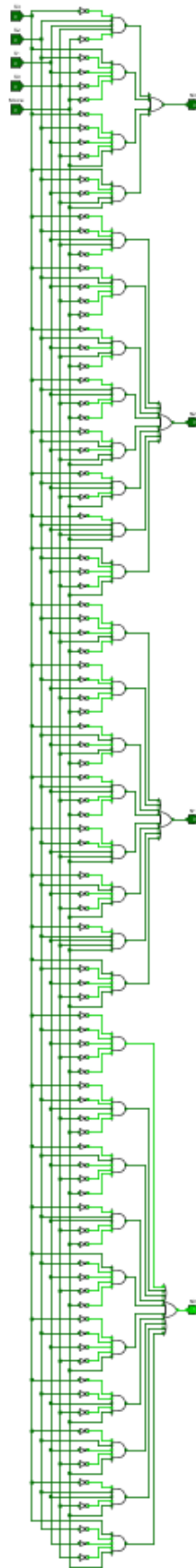


图 3.27 十进制可逆计数器状态机电路图

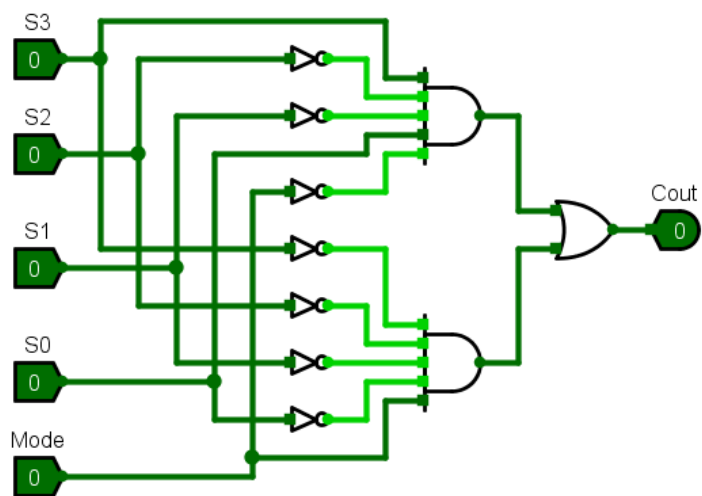


图 3.28 十进制可逆计数器输出函数电路图

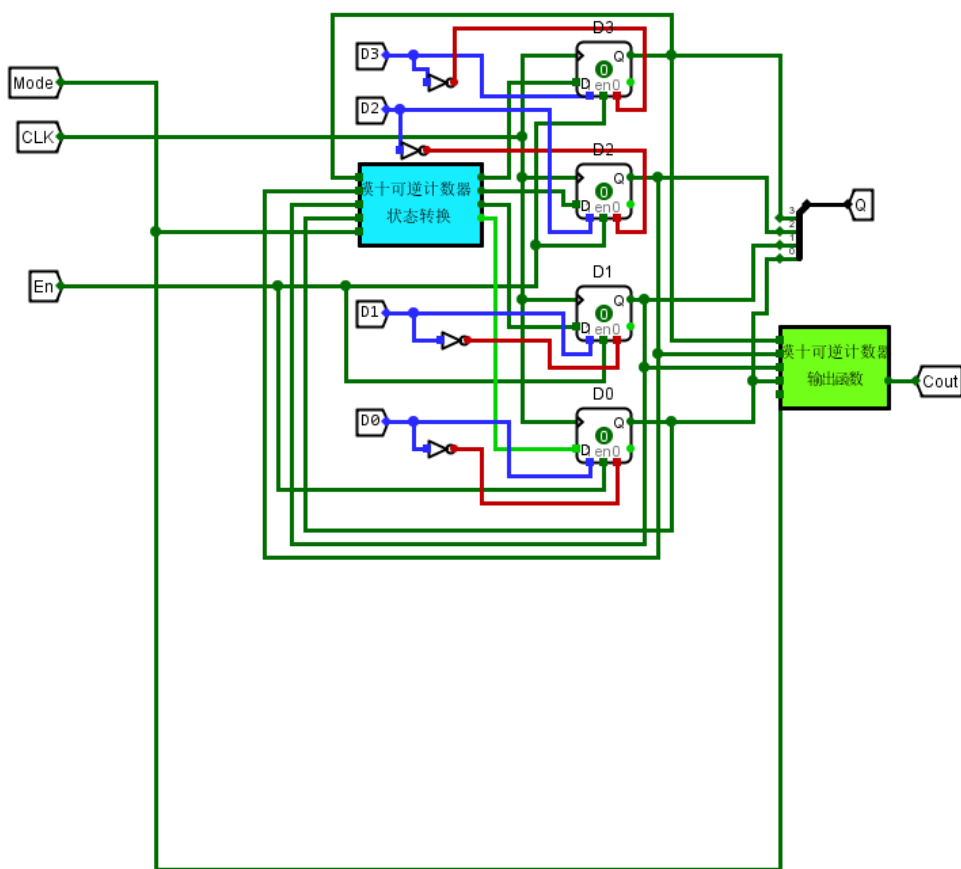
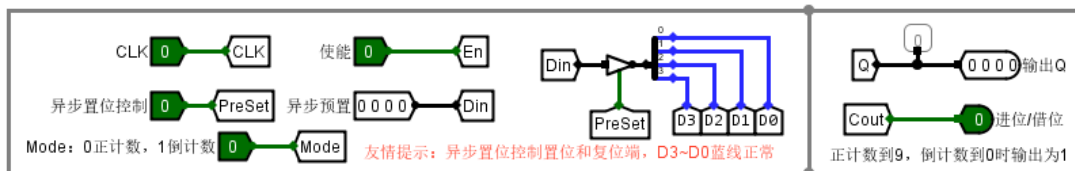


图 3.29 十进制可逆计数器整体电路图

(3) 测试图

状态转换测试：在分别对正向计数和反向计数（Mode=0 与 Mode=1）进行测试，如图 3.30、图 3.31 所示。

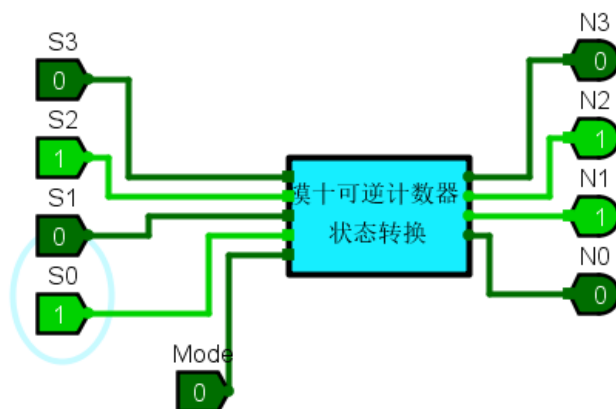


图 3.30 十进制可逆计数器状态机测试 (Mode=0, 即正向计数)

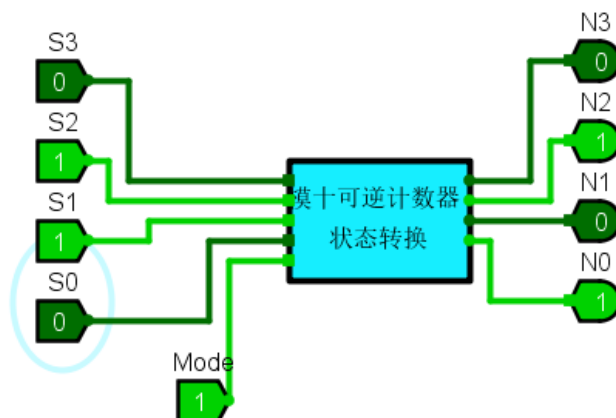


图 3.31 十进制可逆计数器状态机测试 (Mode=1, 即反向计数)

输出函数测试：在分别对正向计数和反向计数（Mode=0 与 Mode=1）需要进位、借位时进行测试，如图 3.32、图 3.33 所示。

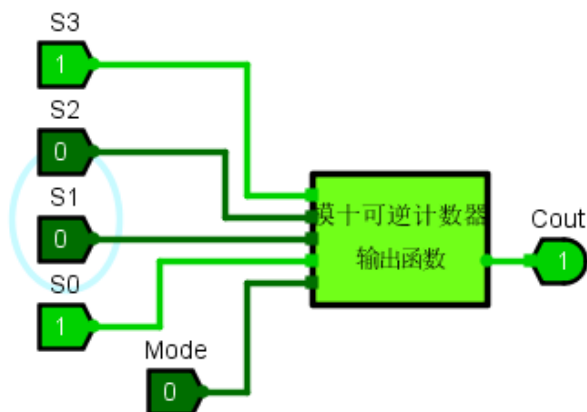


图 3.32 十进制可逆计数器输出函数测试 (Mode=0, 即正向计数到 9 输出 1)

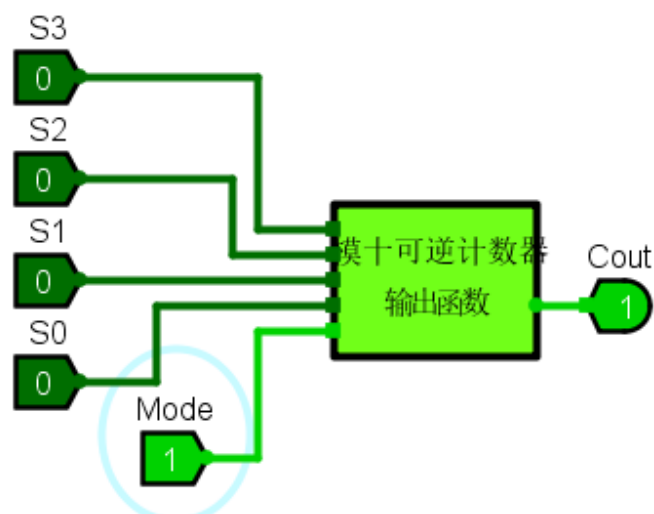


图 3.33 十进制可逆计数器输出函数测试 (Mode=1, 即反向计数到 0 输出 1)

整体测试：测试电路图如图 3.34 所示，之后给出测试的输出输出信号图：

- a. 异步置位控制为 5 (0101)，测试结果如图 3.35 所示。
- b. Mode=0, CLK 上升沿正向计数，测试结果如图 3.36 所示。
- c. 正向计数至 9 (1001) 时输出进位信号 1，测试结果如图 3.37 所示。
- d. Mode=1, CLK 上升沿反向计数，测试结果如图 3.38 所示。
- e. 反向计数至 0 (0000) 时输出借位信号 1，测试结果如图 3.39 所示。

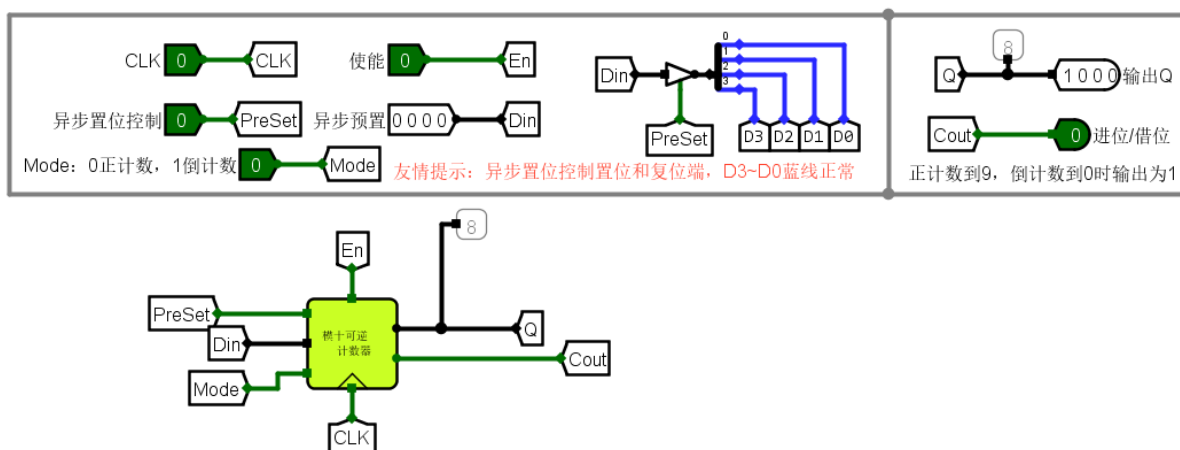


图 3.34 十进制可逆计数器整体测试电路图

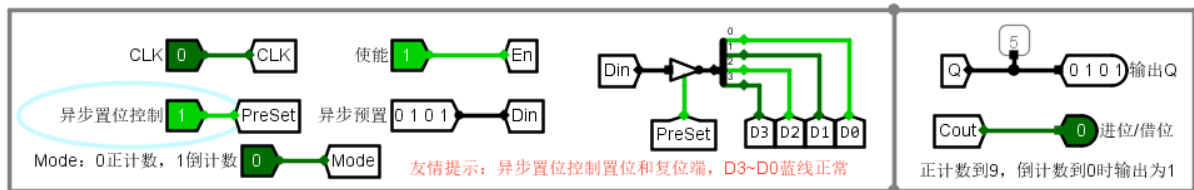


图 3.35 异步置位控制为 5 (0101)

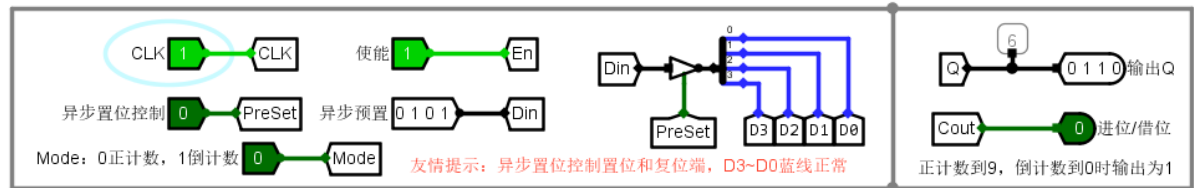


图 3.36 Mode=0, CLK 上升沿正向计数

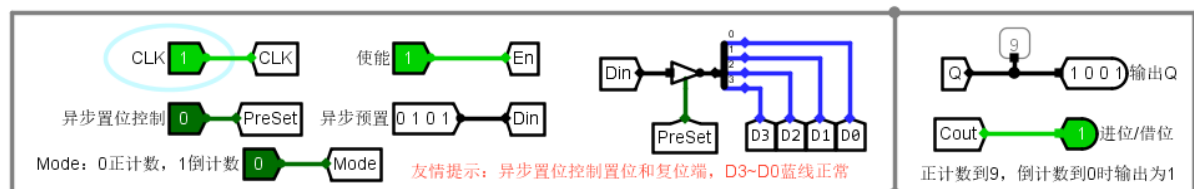


图 3.37 正向计数至 9 (1001) 时输出进位信号 1

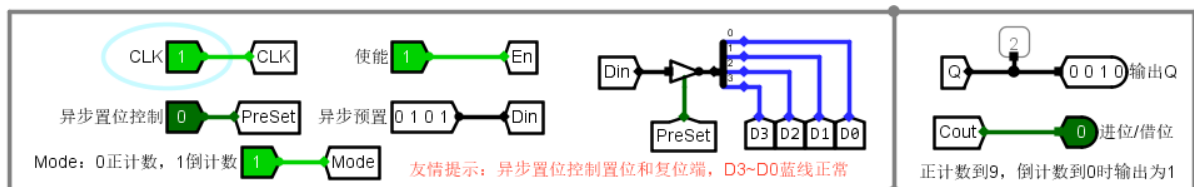


图 3.38 Mode=1, CLK 上升沿反向计数

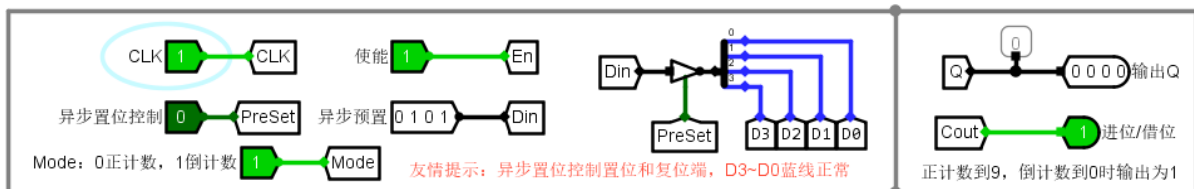


图 3.39 反向计数至 0 (0000) 时输出借位信号 1

(4) 测试分析

状态转换：正向计数和反向计数可正常进行，适时置位。

输出函数：可在正向计数到 9 和反向计数到 0 时输出进位/借位信号。

整体电路：异步置位、正向计数和反向计数都可以正常进行，进位信号与借位信号

也可在正确的状态和输入下输出，通过测试。

3.5 两位十进制可逆计数器

(1) 设计思路及设计过程

利用两个一位模式可逆计数器，一个代表低位，一个代表高位。低位计数器每个时钟周期进行计数，高位计数器仅在低位计数器有进位信号时计数，这里我采用将进位信号与使能通过与门后连接到高位计数器的使能端。输出的 Cout 仅在低位计数器和高位计数器均发出进位信号时才为 1（比如计数到 99 时）。其他的输入（时钟信号 CLK，Mode，En，异步预置信号 Din0 和 Din1）连接到元件对应位置即可。

(2) 电路图

如图 3.40 所示。

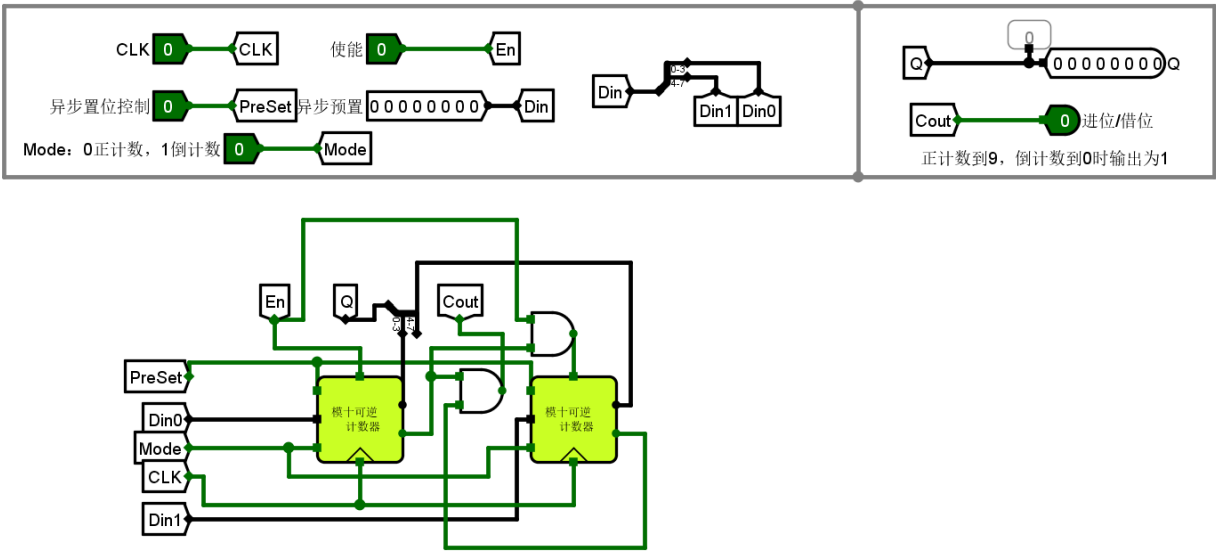


图 3.40 两位十进制可逆计数器电路图

(3) 测试图

测试电路图如图 3.41 所示，之后给出测试的输出输出信号图：

- 异步置位控制为 55（0101 0101），测试结果如图 3.42 所示。
- Mode=0，CLK 上升沿正向计数，测试结果如图 3.43 所示。
- 正向计数至 99（1001 1001）时输出进位信号 1，测试结果如图 3.44 所示。
- Mode=1，CLK 上升沿反向计数，测试结果如图 3.45 所示。
- 反向计数至 00（0000 0000）时输出借位信号 1，测试结果如图 3.46 所示。

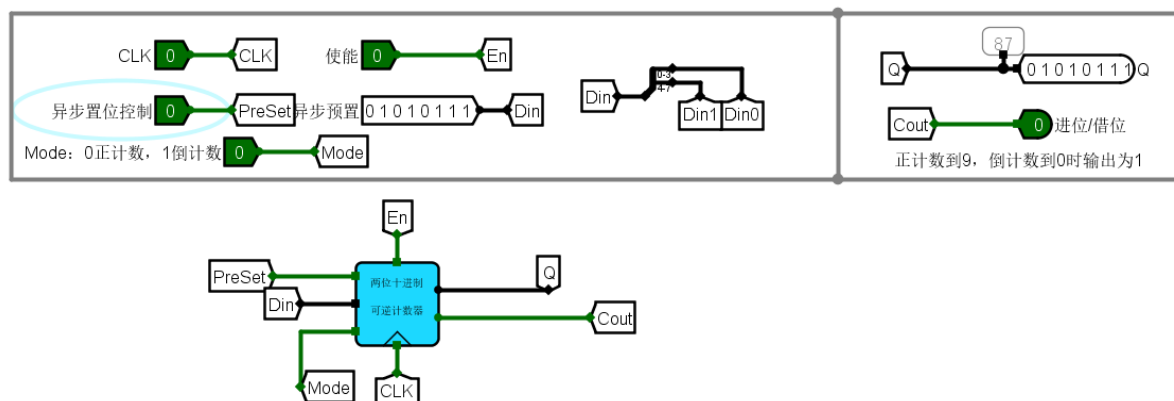


图 3.41 两位十进制可逆计数器测试电路图

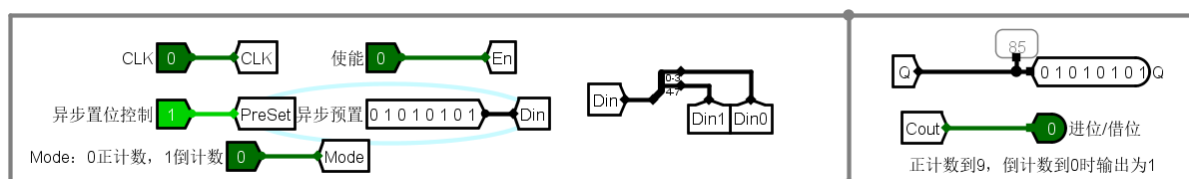


图 3.42 异步置位控制为 55 (0101 0101)

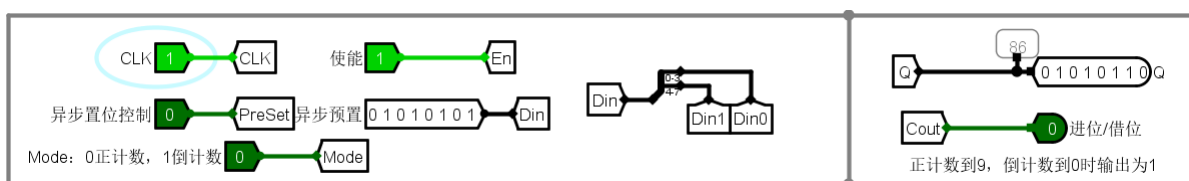


图 3.43 Mode=0, CLK 上升沿正向计数

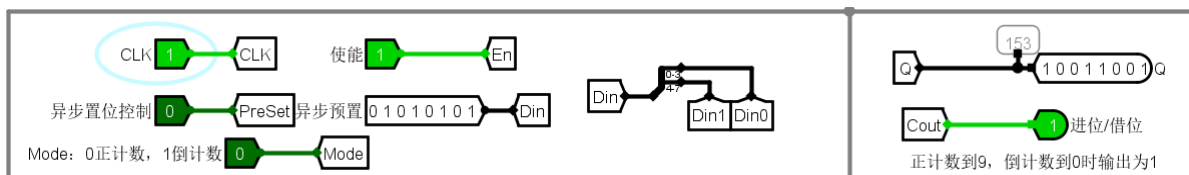


图 3.44 正向计数至 99 (1001 1001) 时输出进位信号 1

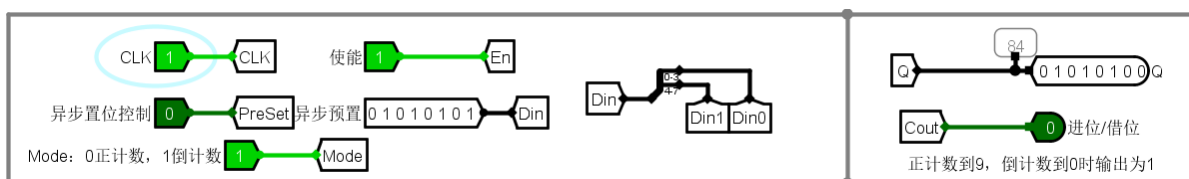


图 3.45 Mode=1, CLK 上升沿反向计数

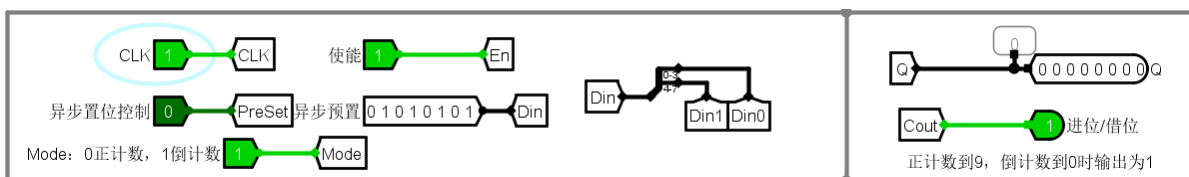


图 3.46 反向计数至 00 (0000 0000) 时输出借位信号 1

(4) 测试分析

异步置位、正向计数和反向计数都可以正常进行，进位信号与借位信号也可在正确的状态和输入下输出，通过测试。

3.6 交通灯状态机

(1) 设计思路及设计过程

根据交通灯设计中的 S0~S7 状态（如表格 3.1 所示）、各输入输出描述（如表格 3.2 所示）、以及交通灯状态转换图（如图 3.47 所示），可以写出交通灯状态机真值表（如图 3.48 所示），并根据真值表生成电路。

表格 3.1 交通灯状态设计

状态编号	状态描述
S0	主、次干道均为黄灯闪烁
S1	非高峰期主干道单侧通行，绿灯。
S2	非高峰期次干道单侧通行，绿灯。
S3	非高峰期主干道通行，绿灯。
S4	非高峰期主干道通行，黄灯。
S5	非高峰期次干道通行，绿灯。
S6	非高峰期次干道通行，黄灯。

表格 3.2 交通灯状态机各输入输出描述

信号	输入/输出
S2~S0	输入
R1	输出
Y1	输出
G1	输出
R2	输出
Y2	输出
G2	输出
PASS1	输出
PASS2	输出
PASS3	输出

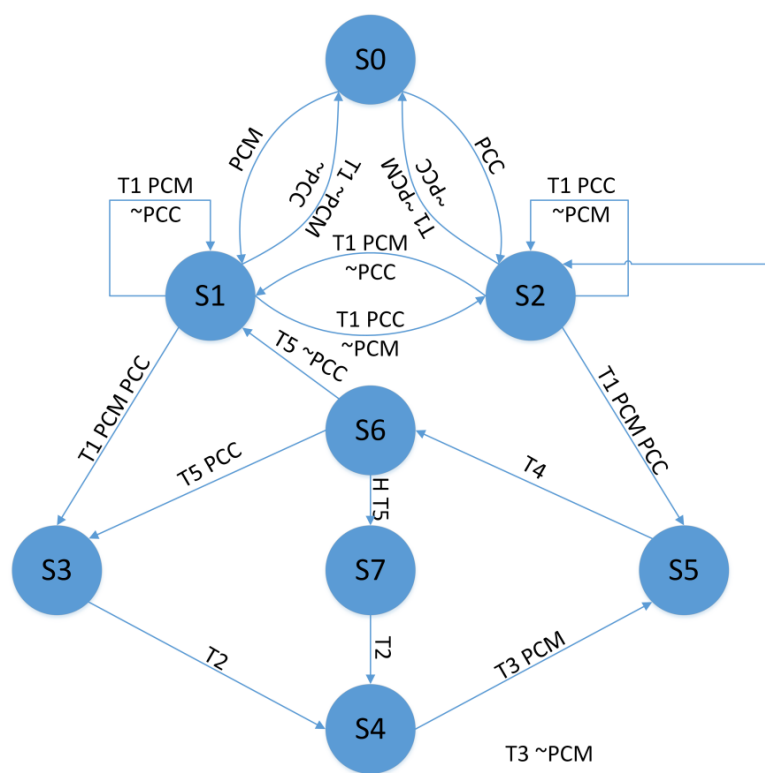


图 3.47 交通灯状态转换图

输入 (填1或0, 不填为无关项x)											输出 (只填写为1的情况)		
S2	S1	S0	H	PCM	PCC	T5	T4	T3	T2	T1	N2	N1	N0
0	0	0		1	0								1
0	0	0		0	1							1	
0	0	1		0	0					1			
0	0	1		1	0					1			1
0	0	1		0	1					1		1	
0	1	0		1	0					1			1
0	1	0		0	0					1			
0	1	0		0	1					1		1	
0	0	1		1	1					1		1	1
0	1	0		1	1					1	1		1
0	1	1							1		1		
1	0	0		1				1			1		1
1	0	0		0				1				1	
1	0	1					1				1	1	
1	1	0	0		0	1							1
1	1	0	0		1	1						1	1
1	1	0	1			1					1	1	1
1	1	1							1		1		
0	0	0		0	0								
0	0	0		1	1							1	1
0	0	1								0			1
0	1	0								0		1	
0	1	1							0			1	1
1	0	0						0			1		
1	1	0				0					1	1	
1	1	1							0		1	1	1
1	0	1					0				1		1

图 3.48 交通灯状态机真值表

(2) 电路图

如图 3.49 所示。

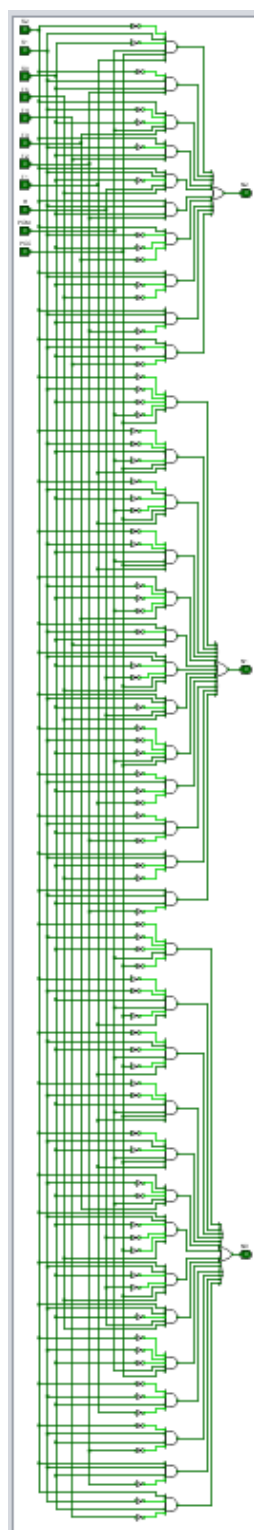


图 3.49 交通灯状态机电路图

(3) 测试图

选取其在 S0 在主干道有通行请求时变为 S1（测试结果如图 3.50 所示）、S1 在主干道次干道均有同行请求转变为 S2（测试结果如图 3.51 所示）、S3 在主干道次干道均有通行请求且主干道绿灯计时结束转变到 S4（测试结果如图 3.52 所示）、S6 在高峰期次车道黄灯倒计时结束后转换到 S7（测试结果如图 3.53 所示），共四个测试样例。

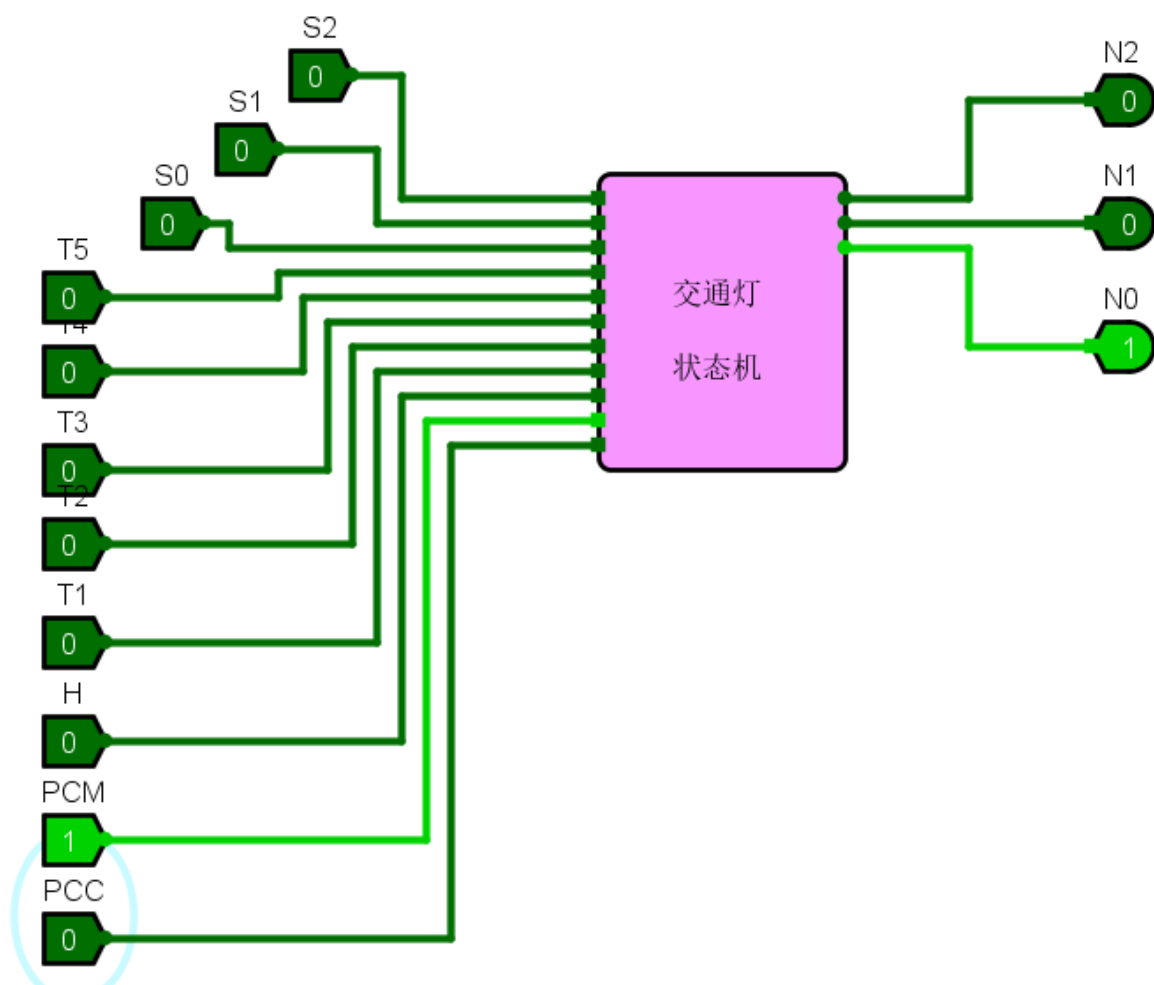


图 3.50 S0 在主干道有通行请求时变为 S1

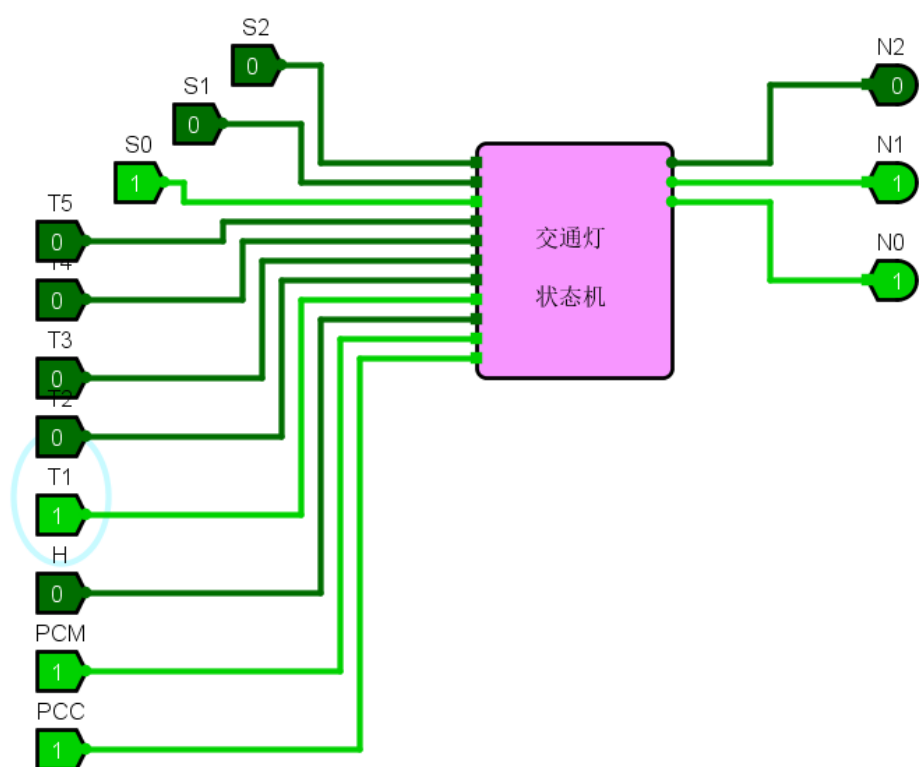


图 3.51 S1 在主干道次干道均有同行请求转变为 S2

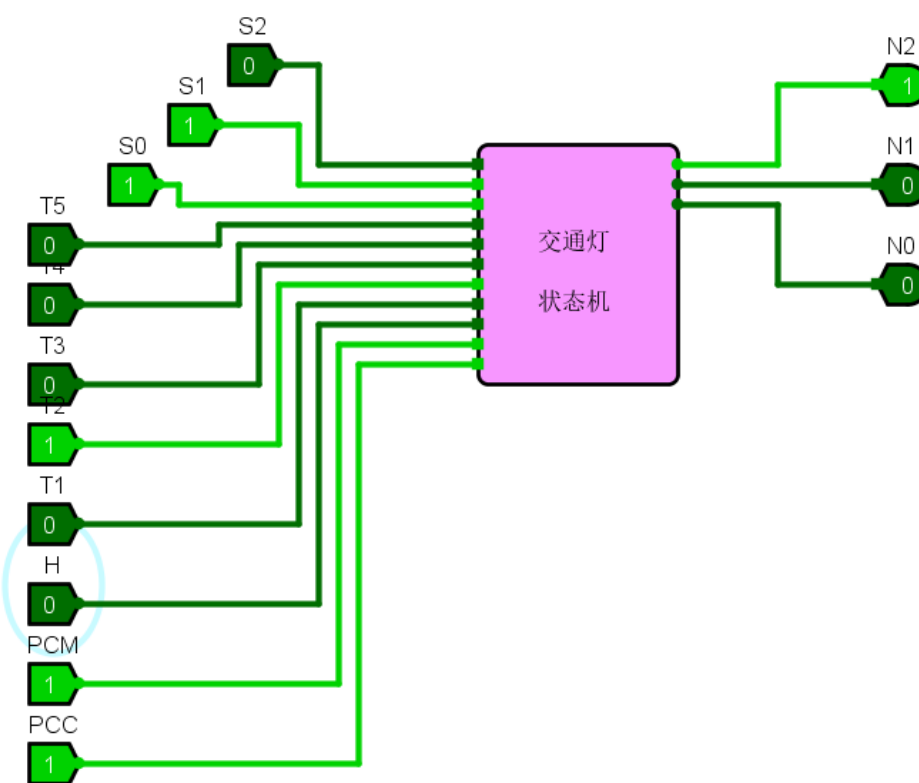


图 3.52 S3 转变到 S4

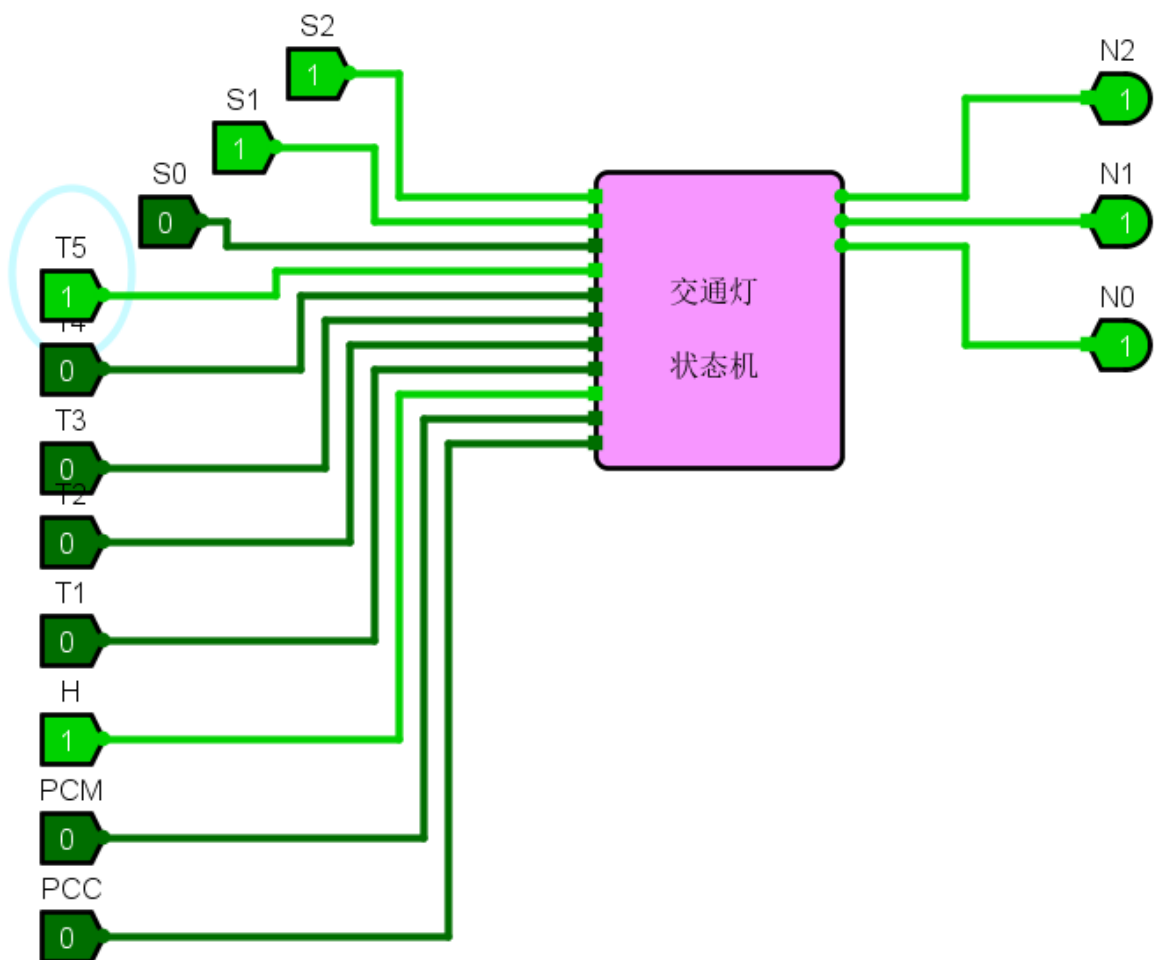


图 3.53 S6 在高峰期转换到 S7

(4) 测试分析

生成的电路满足状态转移图上的状态转移关系，通过测试。

3.7 交通灯输出函数设计

(1) 设计思路及设计过程

根据状态表各状态描述（如表格 3.1 所示），在当前状态为 S0~S7 时输出对应的灯控制信号分别使得对应颜色的灯闪亮，并输出正确的 PASS1~PASS3 信号。各输入输出信号的含义如图 3.54 所示，真值表如图 3.55 所示。

表2 交通灯状态输出函数引脚表

信号	输入/输出	位宽	说明
S2~S0	输入	3 位	当前状态S
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	次道红灯控制信号
Y2	输出	1位	次道黄灯控制信号
G2	输出	1位	次道绿灯控制信号
PASS1	输出	1位	主道允许通行信号
PASS2	输出	1位	次道允许通行信号
PASS3	输出	1位	单侧通行信号

图 3.54 交通灯状态输出函数引脚

组合逻辑电路分析

文件 编辑 工程 电路仿真 窗口 帮助

输入 输出 真值表 表达式 最小项

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

生成电路

图 3.55 交通灯输出函数真值表

(2) 电路图

真值表生成电路如图 3.56 所示。

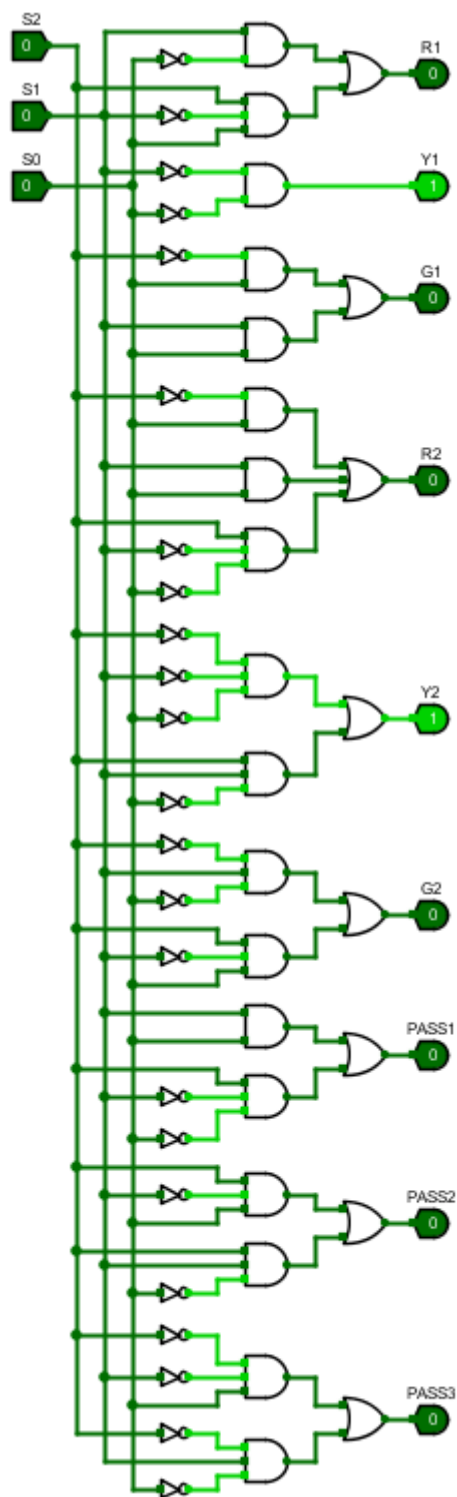


图 3.56 交通灯输出函数电路图

(3) 测试图

选取了 S0,S2,S7 状态下的输出作为测试，如图 3.57、图 3.58、图 3.59 所示。

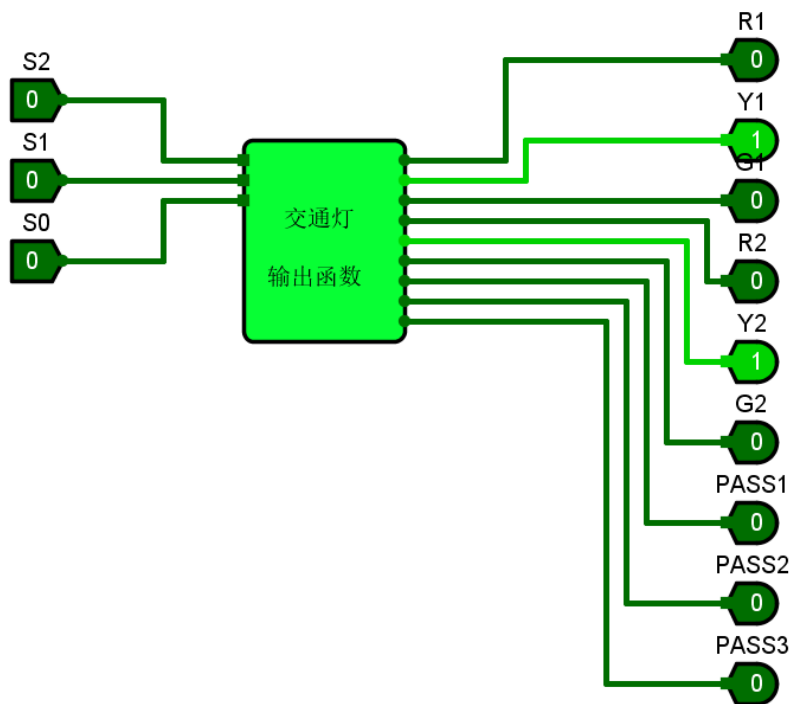


图 3.57 S0 状态下测试电路

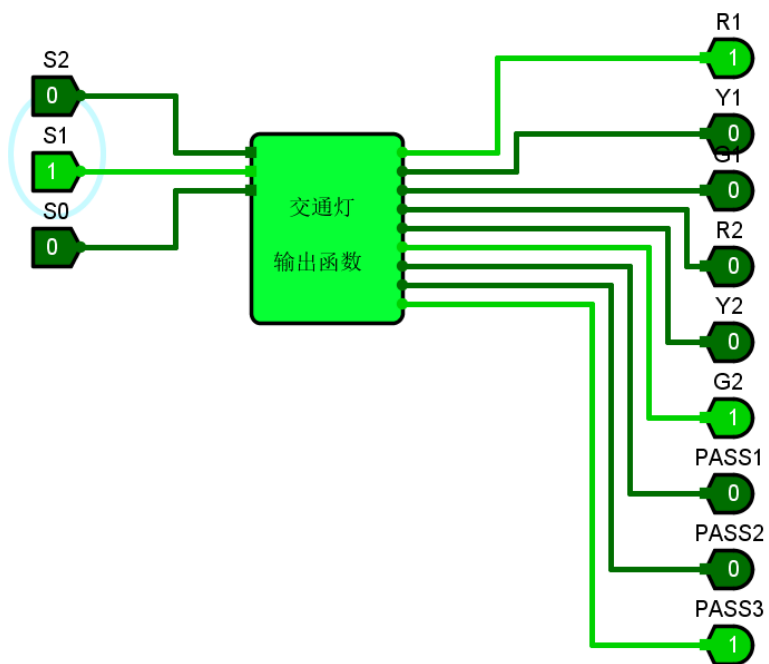


图 3.58 S2 状态下测试电路

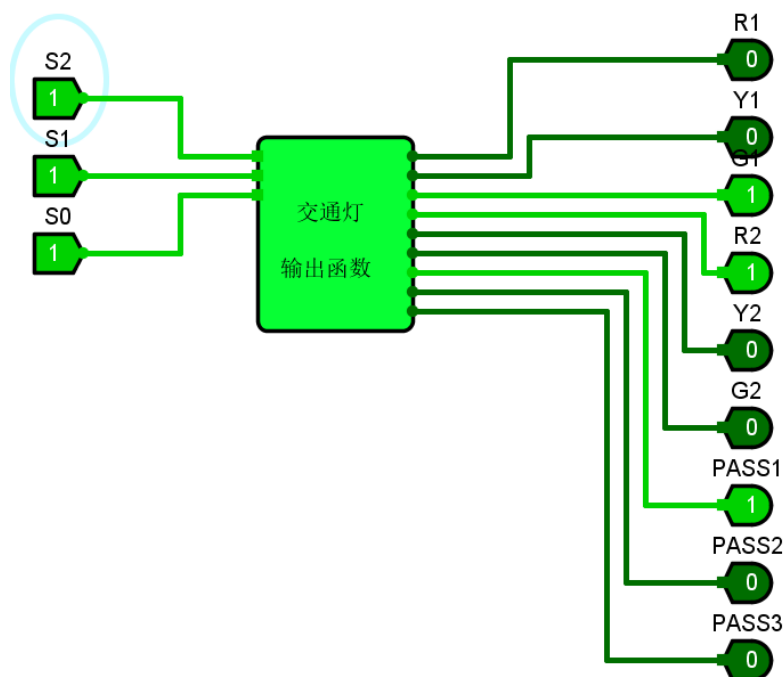


图 3.59 S7 状态下测试电路

(4) 测试分析

输出函数可以在 S0~S7 状态下有对应的正确输出，通过测试。

3.8 交通灯控制系统

(1) 设计思路及设计过程

进行交通灯系统的设计时，首先根据实验设计要求，需要确定基本参数和功能需求。这包括设定交通灯的基本控制参数，如红灯、绿灯、黄灯的时间。在功能方面，交通灯系统应包括红黄绿三色的红绿灯，分别用于主次干道。此外，系统需要显示当前倒计时数码管，并能根据不同情况进行正确的红绿灯切换。

本人采用模块化设计方法，将系统分为各个子电路，便于调试和后期维护。先前已经将交通灯系统所需的各部件设计完毕（如交通灯输出函数、交通灯状态机、双位 BCD 双向计数器等），此处只需根据交通灯通行规则把前面的各部件连接成最终的电路。

这里交通灯控制系统分为六个模块，红绿灯输入输出模块、倒计时选择主干道倒计时、次干道倒计时、单侧通行倒计时、紧急情况选择。

红绿灯输入输出模块、和倒计时选择模块只需把交通灯输出函数、交通灯状态机对应引脚接好即可。

在三个倒计时模块中加入了 T6、T7、T8 三个额外的信号进行倒计时的启动，即在电路复位、计时器全部为 0 时强制开始倒计时，否则电路无法启动。

紧急情况选择模块设计时，按照规则如果收到总控制台的信号，在倒计时结束时才会进入紧急状态，而不再接受到总控制台的信号时要立即变回原状态，所以在将紧急状况信号置 1 时要等待 T3、T1、T5 的信号，而置 0 时直接强制置零。

(2) 电路图

如图 3.60。

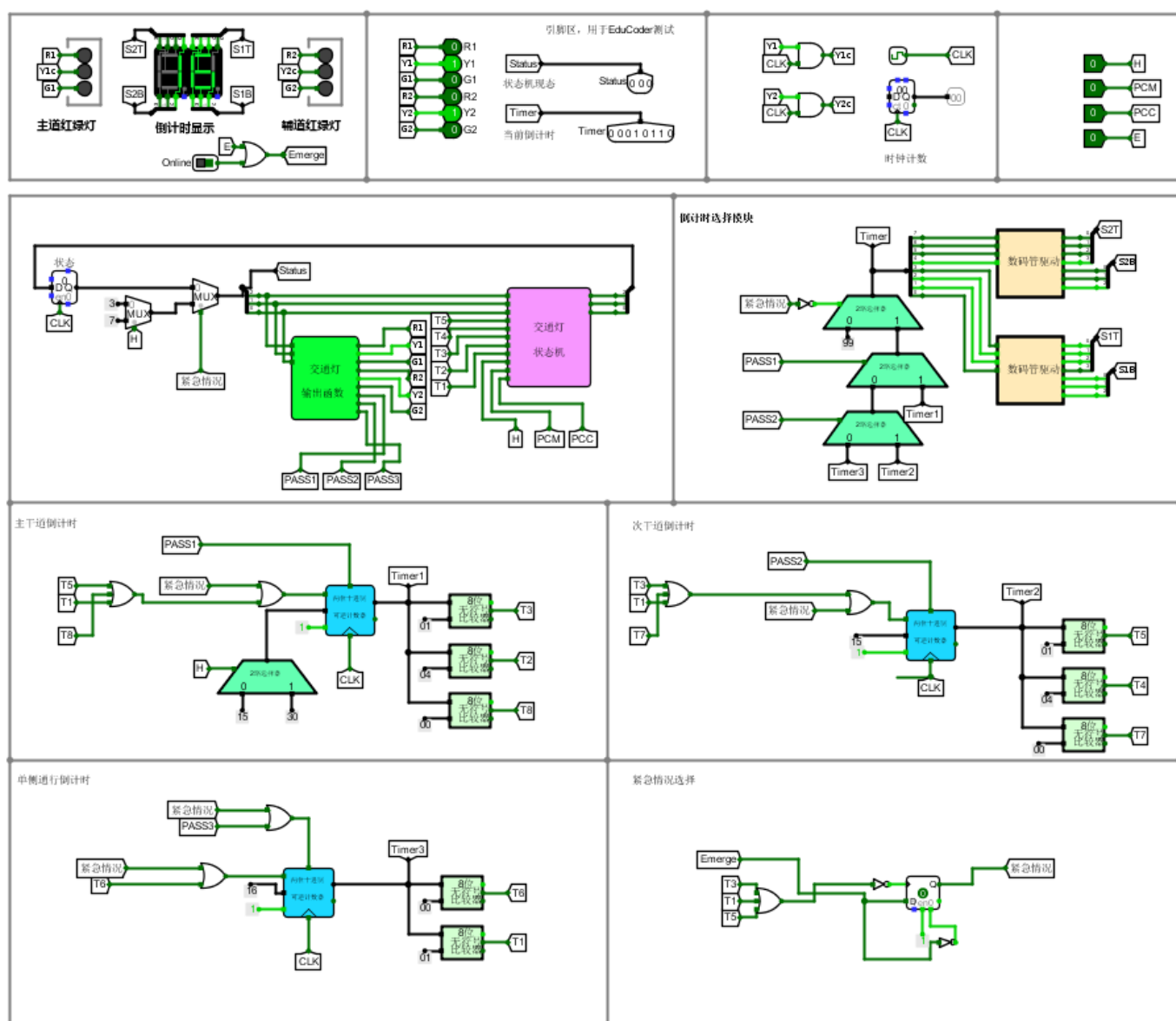


图 3.60 交通灯电路图

(3) 测试图

测试电路图如图 3.60 交通灯电路图所示。

电路复位后，回到初始状态。

首先不进行任何输入，黄灯闪烁，如图 3.61 所示。

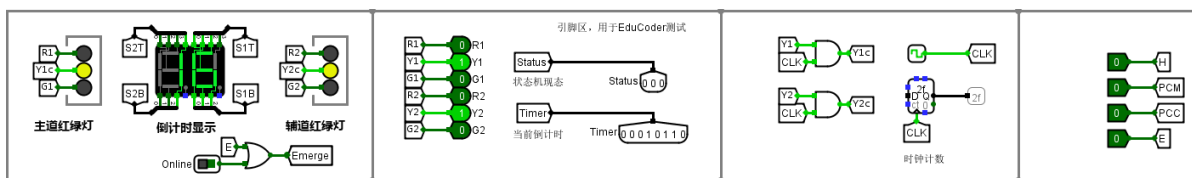


图 3.61 无同行请求，黄灯闪烁

之后收到主干道同行请求，开始单侧通行，从 16 开始倒计时，如图 3.62 所示。

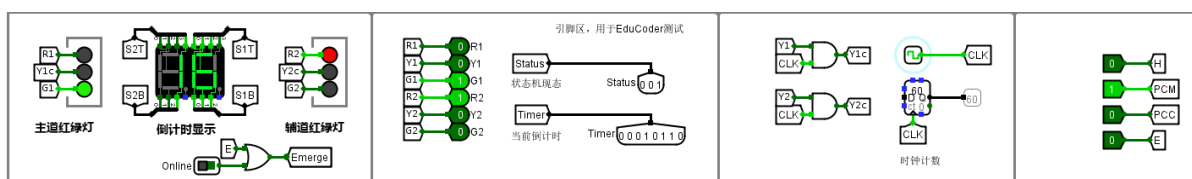


图 3.62 单侧通行

之后又收到了次干道的通行请求，开始主次干道交替通行，如图 3.63 所示。

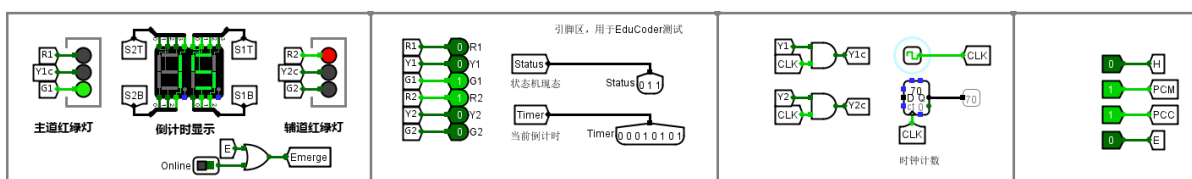


图 3.63 主次干道交替通行，主干道绿灯

主干道交通灯在倒计时最后三秒时变为黄灯，如图 3.64 所示。

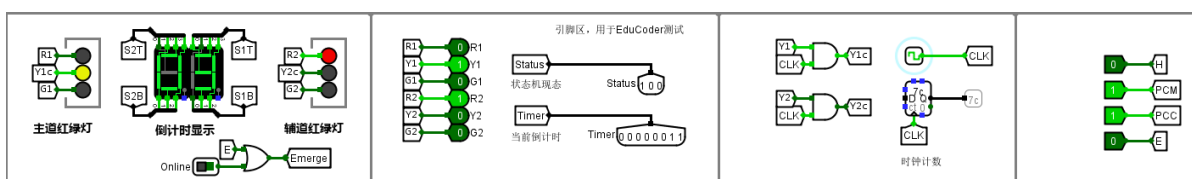


图 3.64 主次干道交替通行，主干道黄灯

倒计时结束后正常切换至次干道通行，次干道绿灯，主干道红灯，从 15 开始倒计时，如图 3.65 所示。

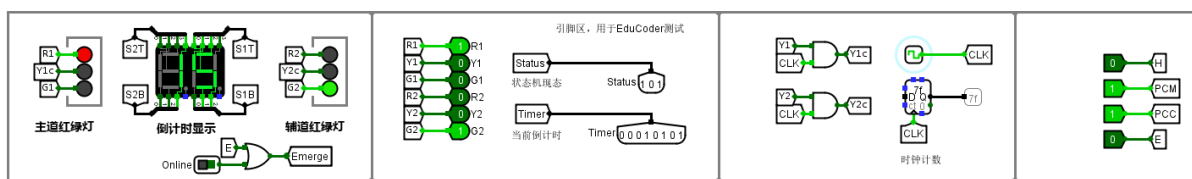


图 3.65 主次干道交替通行，切换至次干道绿灯

次干道交通灯在倒计时最后三秒时变为黄灯，如图 3.66 所示。

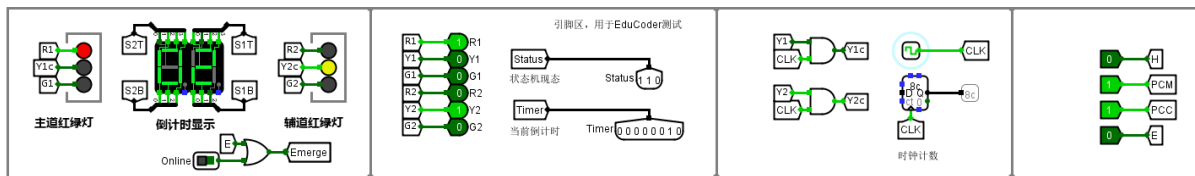


图 3.66 主次干道交替通行，次干道黄灯

输入 H，进入高峰期，主干道通行时间翻倍，从 30 开始计时，如图 3.67 所示。

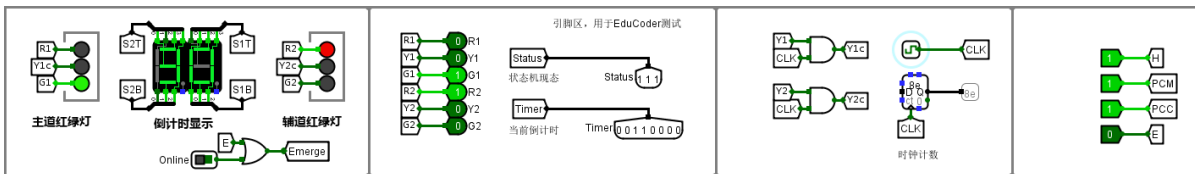


图 3.67 高峰期，主干道通行

之后收到紧急状况信号 E，代表此时遇到紧急状况，在倒计时结束后会进入紧急状况，如图 3.68 所示。

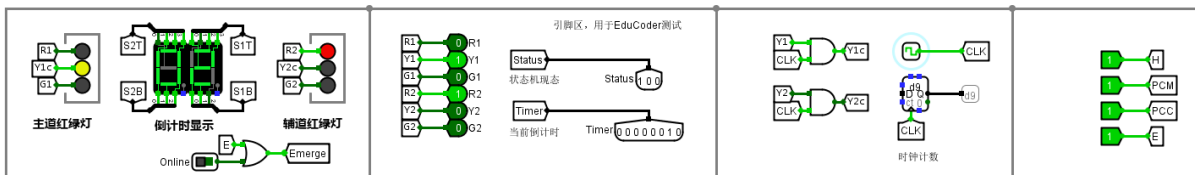


图 3.68 等待计时结束进入紧急状态

倒计时结束后，进入紧急状态，主干道通行，次干道禁止通行，倒计时显示 99 不变，如图 3.69 所示。

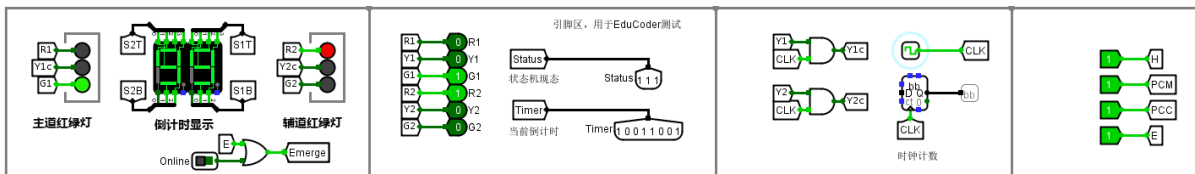


图 3.69 紧急状态

当紧急状态信号 E 变为 0 时，立刻退出紧急状态，回到进入紧急状态之前的状态（这里轮到次干道通行），如图 3.70 所示。

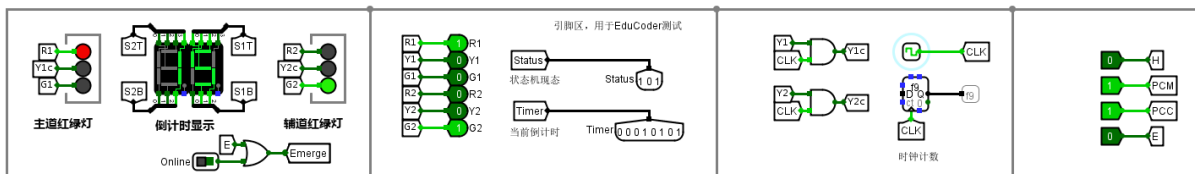


图 3.70 退出紧急状态

(4) 测试分析

交通灯系统的行为符合题中所给出的规则，测试通过。

4 设计总结与心得

4.1 实验总结

在本次交通灯控制系统设计实验中，我利用 logisim，运用课上学到的数字电路设计的知识，成功搭建了一个基本的交通灯控制系统。在实验过程中，我对数字电路设计有了更深刻的理解，并学到了许多关于交通流控制的实际应用知识。

通过运用状态机的概念，我们能够清晰地描述不同交通灯状态之间的转换关系，并通过状态输出函数实现相应的控制逻辑。这使得系统的设计更加模块化和可维护。

在设计中，我合理地把电路划分为多个子电路，构建了数码管驱动、8 位无符号比较器、8 位二路选择器和双位 BCD 双向计数器等子电路，并在最后将子电路组装起来成为交通灯系统。使得整个系统的设计更为简洁和高效，而且具有较强的可维护性。

4.1.1 遇到的问题及处理

1.交通灯控制系统的设计时，我开始对 T6,T7,T8 这三个新出现的信号一无所知，不懂它在电路中起到的作用，之后在设计过程中做出了一些猜测，最终自己分析出这三个信号应该是在电路复位后强制开始倒计时的，和后来老师讲到的一样，很有成就感。

2.交通灯系统的设计中的紧急状况选择部分，花了我很长时间，每次我觉得它应该能正确地完成任务时就会得到“FAIL”的反馈，后来发现我理解错了题意，当紧急状况信号输入 1 时，他要等计时结束才进入紧急状况，而当他输入 0 时要立刻退出紧急状况并开始正常的工作，并更改了电路，最后成功得到了“PASS”。

4.1.2 设计方案存在的不足

1.在状态机的实现中，状态转移可能存在一些细节上的问题。本人并没有考虑所有可能的输入与状态条件下的状态转移，在非正常输入输出情况下，可能发现一些不符合常理的状态转换，甚至存在死锁或不稳定的情况。这些都需要仔细检查和验证。

2. 时序问题：在设计中，需要考虑各种情况下的时序问题，包括计时器的准确计时和状态切换的时机。如果时序控制不精准，时钟信号不符合要求，可能导致交通灯系统的混乱或不符合实际交通需求。

3. 未考虑到实际物理操作延迟：在实际的交通灯系统中，控制信号的切换和灯光状态的变化都需要一定的时间，这一点在设计中没有被充分考虑。因此，在实际应用中可能需要引入一些缓冲和延迟机制，以模拟实际物理操作。

4. 系统简易，不够智能化。系统只是基本完成了作为交通灯的职责，而现实情况更加复杂。比如现实中交通流量可能会不断变化，应当根据实时交通流量变化进行的动态调整，以提高交通效率。

4.2 实验心得

最初拿到这个实验时，我是个数电小白，看到这样的设计要求根本无从下手。但在老师的引导和同学们的帮助下，我以闯关的形式分步骤完成各个部件，最终级联成一个交通灯控制系统。细细想来，每一次过关我都有很强满足感，并热切地想要继续闯下一关。大概世间万事都没有我们想象的那么难，把大目标分解成小目标，一步一个脚印地向前，都能取得不错的成果。

这次也是我第一次接触 logisim 并使用这个软件。通过 logisim 的实际操作，我更深入地理解了数字电路设计的原理和流程，对数字电路在实际场景中的工程应用有了更直观的认识。从把通行规则翻译为电路，到临界条件的考虑，这种实际应用的经验对今后的工程设计工作具有参考价值。

总体而言，这次实验不仅提高了我对数字电路设计的理解，也培养了我的逻辑思维和问题解决能力。通过动手实践，我更深入地学习了交通灯控制系统的原理和设计方法，为我今后的学习和工作奠定了坚实的基础。

4.3 意见与建议

我认为头歌的评测平台相当不智能，只有按照头歌的意思进行设计、完成电路才能过关。这虽然给予了我们指引，一定程度上减轻了学生负担，但是却使得我在第一节时对电路总体的设计变成了一张废纸。

头歌的标准答案也没有完全反映题目的要求，在某些方甚至与设计题目的要求相悖。比如在从单侧主干道同行的状态下加入了次干道的通行请求时，按照题目要求和生活中的经验，接下来应该变为次干道通行之后交替通行，而最终的答案却是主干道计时结束后重新计时 15 秒，之后才改为次干道并交替通行。显然是不满足实际情况的。

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：田清林