数字码表实验报告

一、 实验目的

掌握数字系统的特点
掌握数字系统的设计方法：模块化、层次化
掌握组合逻辑的设计流程与设计思想
掌握同步时序电路的设计流程与设计思想
掌握系统集成联调的方法

二、 实验任务说明/分析

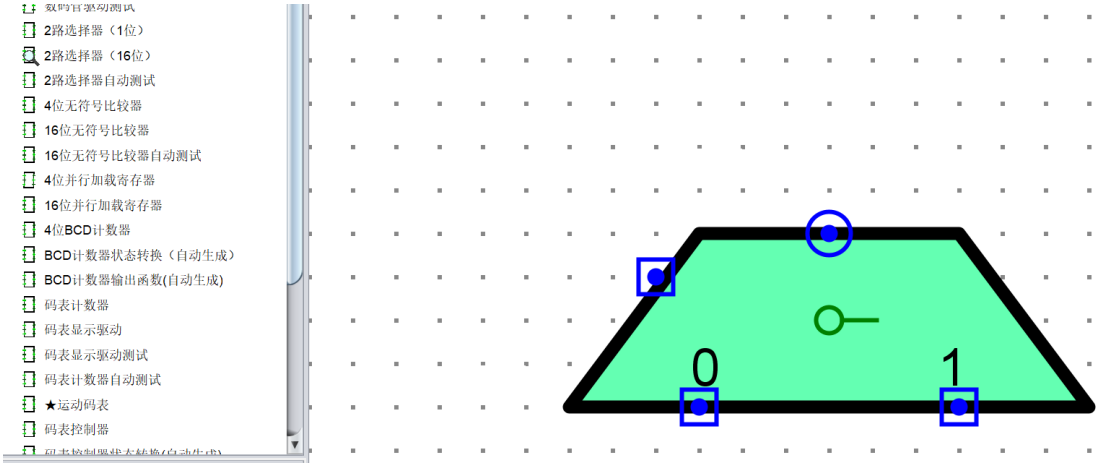
运动码表的功能
输入：4 个按钮
输出：4 个 7 段数码管
Start：计时器归 0，重新开始计时
Stop：停止计时，显示当前计时数据
Store：尝试更新系统记录的计时数据（满足条件：当前计时数据 < 系统记录 ）
Reset：复位，使得计时器 = 00.00， 系统记录 = 99.99

三、 实验过程

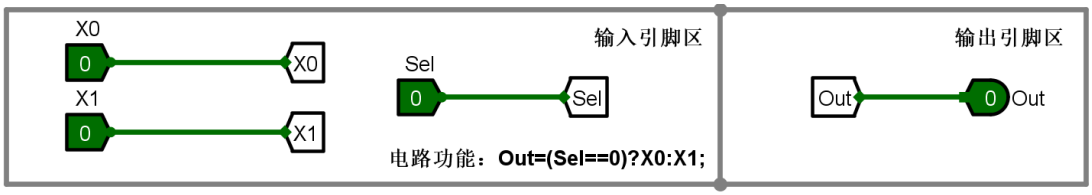
运动码表的设计分为三部分
一、组合逻辑电路设计
　　2 路选择器
　　16 位无符号比较
　　数码管显示
二、同步时序逻辑设计
　　16 位寄存器
　　4 位 BDC 计数器
　　码表计数器
三、数字系统综合设计
　　运动码表数据通路建设
　　运动码表控制单元设计
　　系统集成联调

1. 功能部件设计

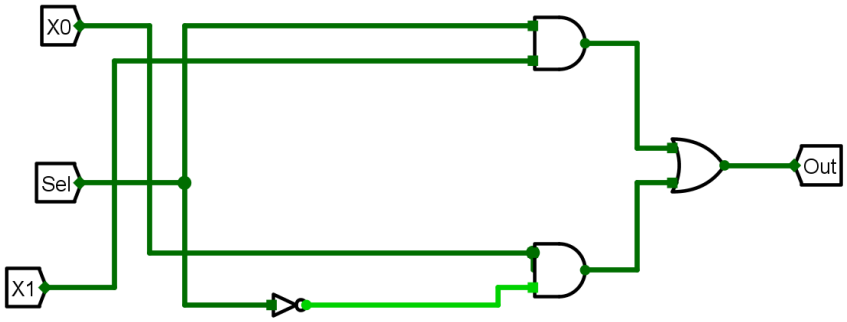
2 路选择器



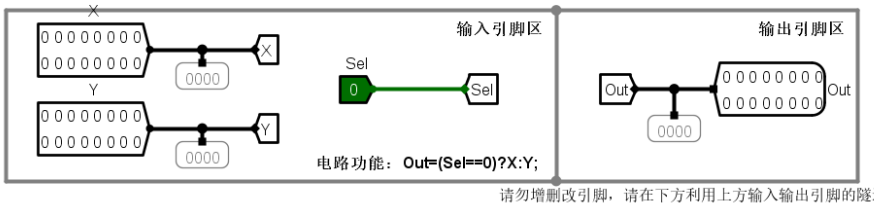
输入：16 位的输入 X.Y；选择控制信号 Sel
输出：16 位输出 Out
功能：out= （Sel==0）？X:Y（当 Sel 为 0，选择 X 作为输出；否则，选择 Y 输出）
约束条件：只能运用线路库、逻辑门组件，输入输出库自行构建
设计要点： 先构建 1 位的二路选择器，再并发为 16 位



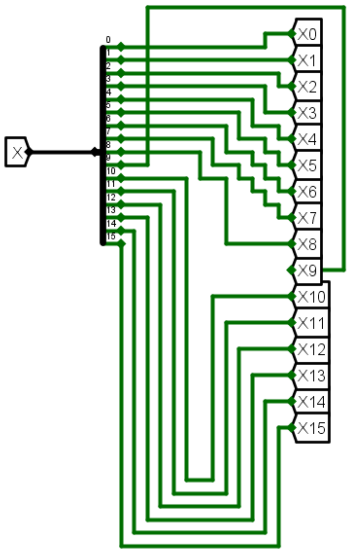
请勿增删改引脚，请在下方利用上方输入输出引脚的隧道标签信号构建电路，ctrl+d复制选择组件

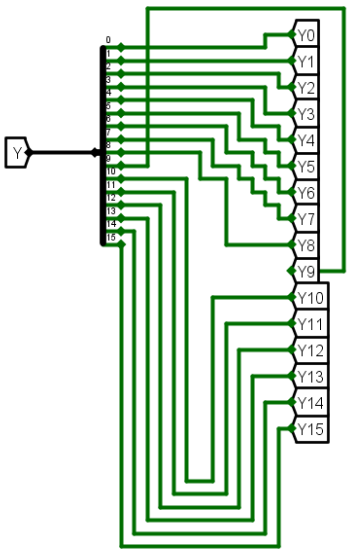


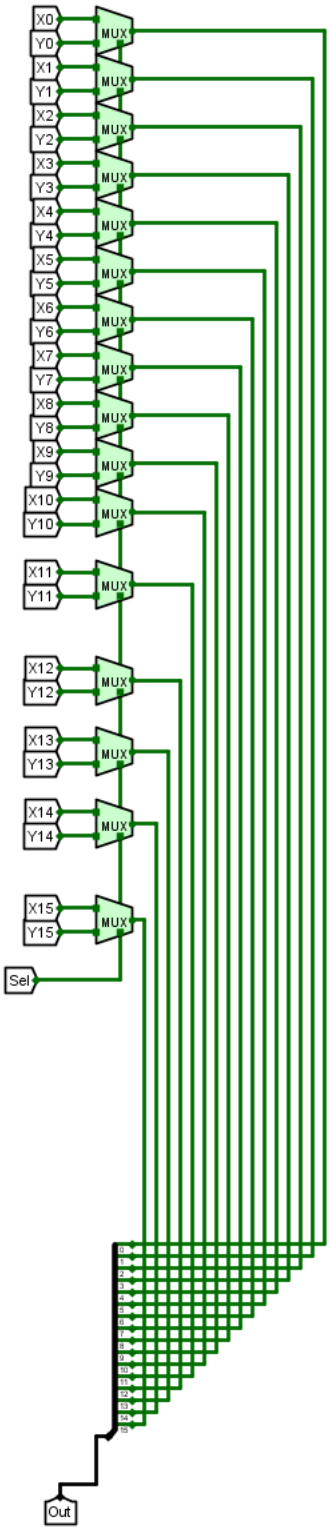
上方为 1 位，下方为 16 位



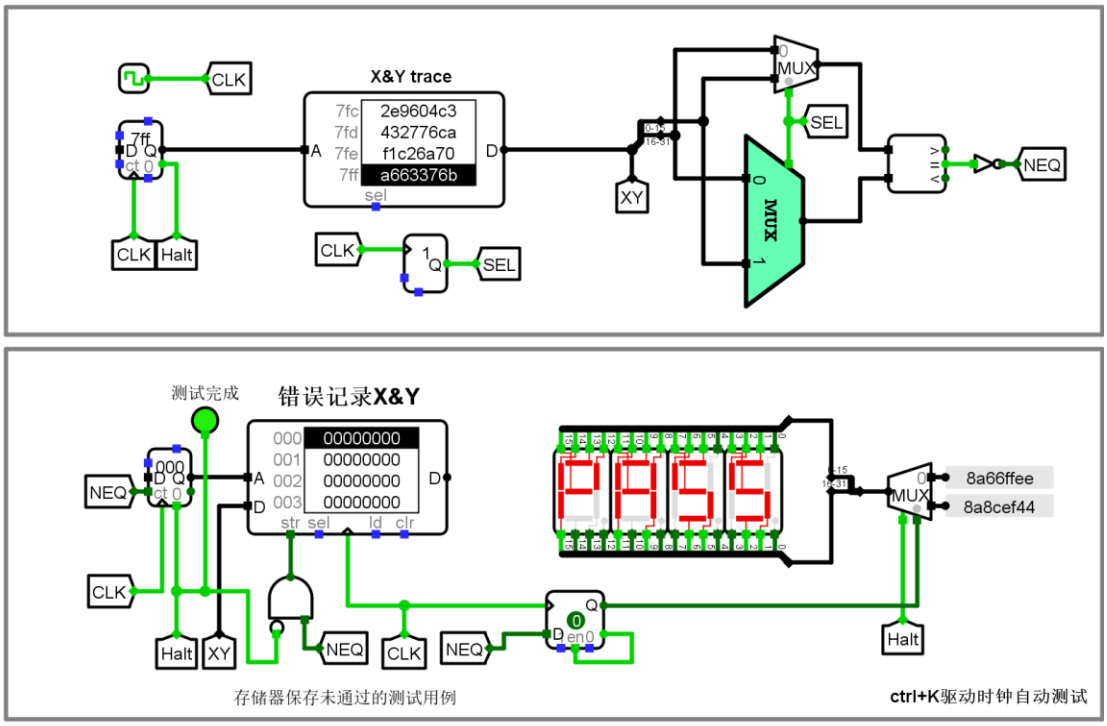
请勿增删改引脚，请在下方利用上方输入输出引脚的隧道标签信号构建电路，ctrl+d复制






X0
Y0
X1
Y1
X2
Y2
X3
Y3
X4
Y4
X5
Y5
X6
Y6
X7
Y7
X8
Y8
X9
Y9
X10
Y10
X11
Y11
X12
Y12
X13
Y13
X14
Y14
X15
Y15
Sel
MUX
Out

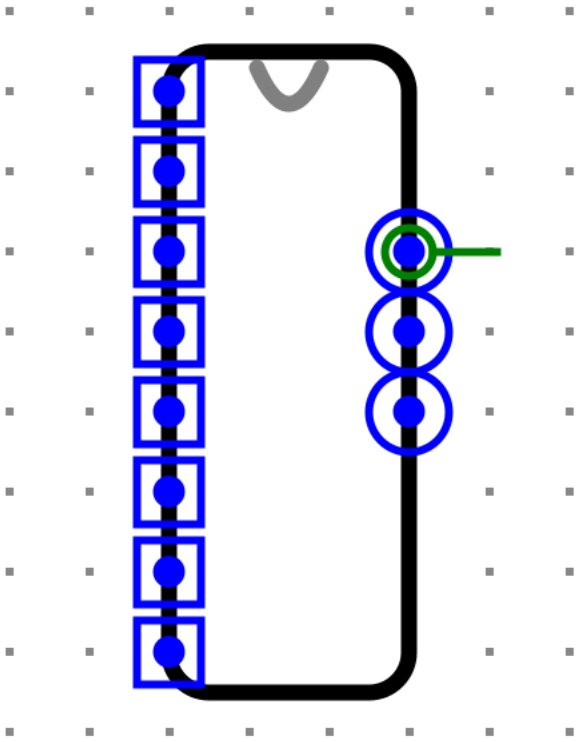
自动测试效果



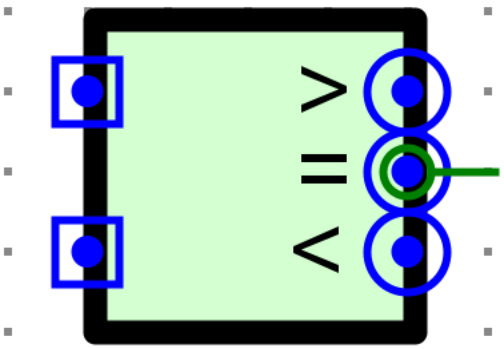
多路选择器自动测试

无符号比较器

4 位无符号比较器



16 位无符号比较器



输入：16 位的输入 X.Y
输出：大于（1 位），等于（1 位），小于（1 位）
功能：无符号的比较两个输入，输出结果
约束条件：只能运用线路库、逻辑门组件，输入输出库自行构建
先构建 4 位的无符号比较器，进而再扩展为 16 位 4 位无符号比较器：通过 excel 生成逻辑表达式，进而自动生成电路

4 位比较器（简易）真值表

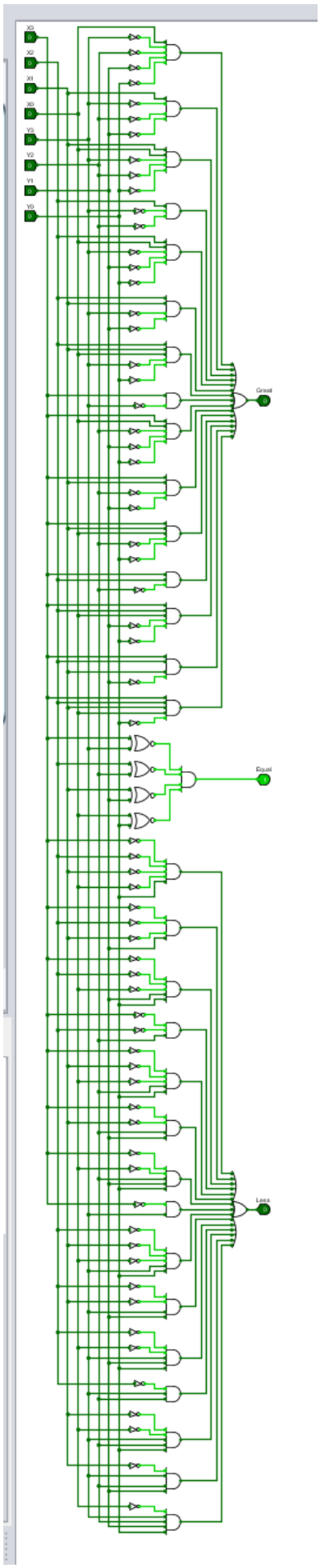
输入						输出		
X3	Y3	X2	Y2	X1	Y1	X0	Y0	Great	Equal	Less
>		X		X		X		1	0	0
=		>		X		X		1	0	0
=		=		>		X		1	0	0
=		=		=		>		1	0	0
=		=		=		=		0	1	0
<		X		X		X		0	0	1
=		<		X		X		0	0	1
=		=		<				0	0	1
=		=		=		<		0	0	1

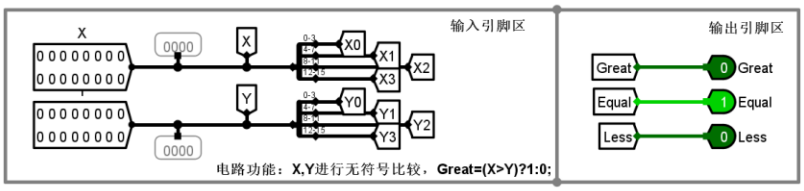
Great= X0 ~Y3 ~Y2 ~Y1 ~Y0 + X1 ~Y3 ~Y2 ~Y1 + X1 X0 ~Y3 ~Y2 ~Y0 + X2 ~Y3 ~Y2 + X2 X0 ~Y3 ~Y1 ~Y0 + X2 X1 ~Y3 ~Y1 + X2 X1 X0 ~Y3 ~Y0 + X3 ~Y3 + X3 X0 ~Y2 ~Y1 ~Y0 + X3 X1 ~Y2 ~Y1 + X3 X1 X0 ~Y2 ~Y0 + X3 X2 ~Y2 + X3 X2 X0 ~Y1 ~Y0 + X3 X2 X1 ~Y1 + X3 X2 X1 X0 ~Y0

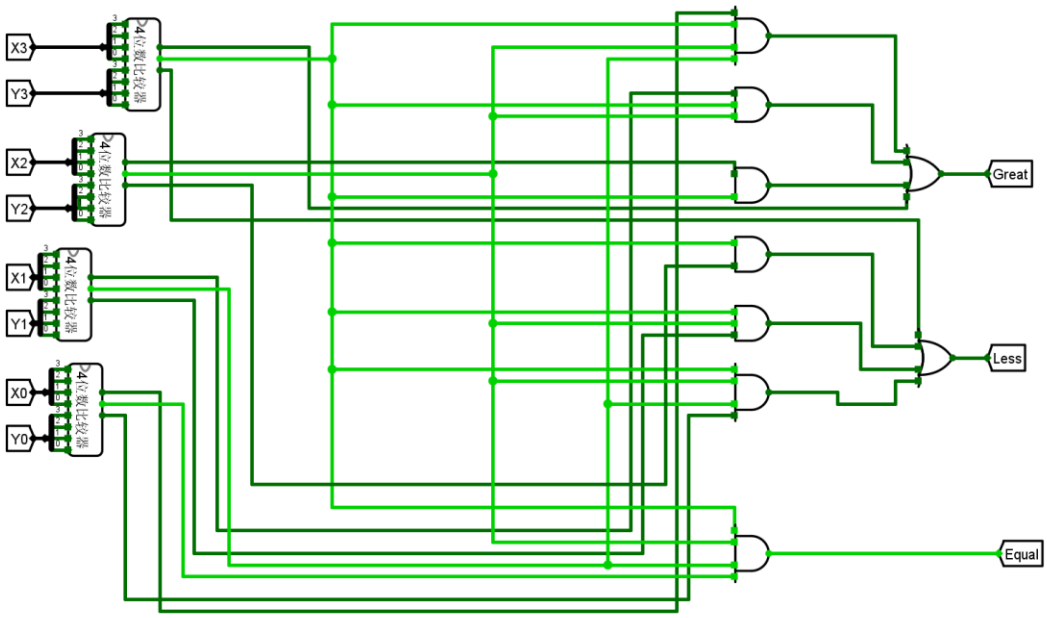
Equal= ~X3 ~X2 ~X1 ~X0 ~Y3 ~Y2 ~Y1 ~Y0 + ~X3 ~X2 ~X1 X0 ~Y3 ~Y2 ~Y1 Y0 + ~X3 ~X2 X1 ~X0 ~Y3 ~Y2 Y1 ~Y0 + ~X3 ~X2 X1 X0 ~Y3 ~Y2 Y1 Y0 + ~X3 X2 ~X1 ~X0 ~Y3 Y2 ~Y1 ~Y0 + ~X3 X2 ~X1 X0 ~Y3 Y2 ~Y1 Y0 + ~X3 X2 X1 ~X0 ~Y3 Y2 Y1 ~Y0 + ~X3 X2 X1 X0 ~Y3 Y2 Y1 Y0 + X3 ~X2 ~X1 ~X0 Y3 ~Y2 ~Y1 ~Y0 + X3 ~X2 ~X1 X0 Y3 ~Y2 ~Y1 Y0 + X3 ~X2 X1 ~X0 Y3 ~Y2 Y1 ~Y0 + X3 ~X2 X1 X0 Y3 ~Y2 Y1 Y0 + X3 X2 ~X1 ~X0 Y3 Y2 ~Y1 ~Y0 + X3 X2 ~X1 X0 Y3 Y2 ~Y1 Y0 + X3 X2 X1 ~X0 Y3 Y2 Y1 ~Y0 + X3 X2 X1 X0 Y3 Y2 Y1 Y0

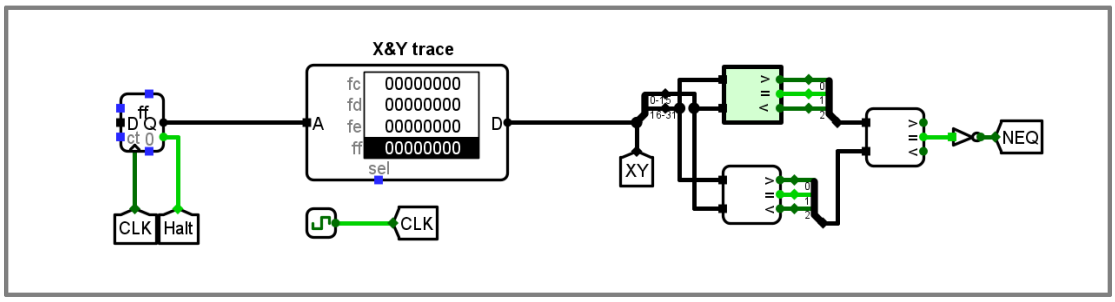
Less= ~X3 ~X2 ~X1 ~X0 Y0 + ~X3 ~X2 ~X1 Y1 + ~X3 ~X2 ~X0 Y1 Y0 + ~X3 ~X2 Y2 + ~X3

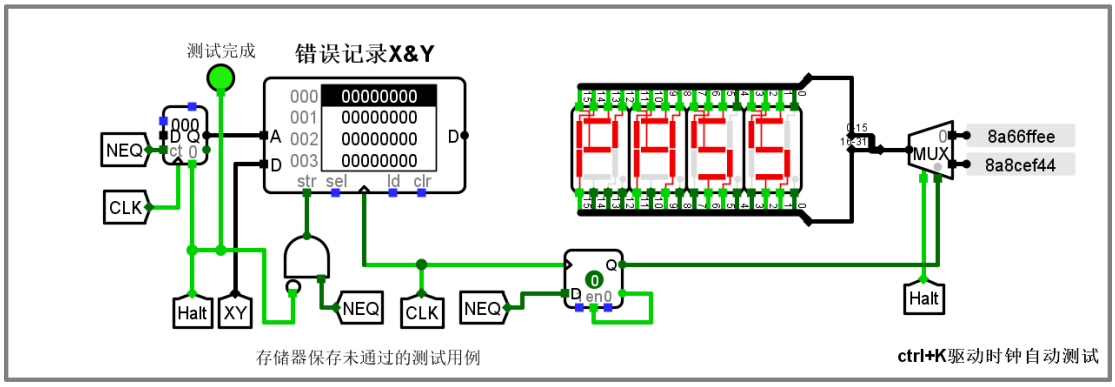
~X1 ~X0 Y2 Y0 + ~X3 ~X1 Y2 Y1 + ~X3 ~X0 Y2 Y1 Y0 + ~X3 Y3 + ~X2 ~X1 ~X0 Y3 Y0 + ~X2 ~X1 Y3 Y1 + ~X2 ~X0 Y3 Y1 Y0 + ~X2 Y3 Y2 + ~X1 ~X0 Y3 Y2 Y0 + ~X1 Y3 Y2 Y1 + ~X0 Y3 Y2 Y1 Y0


X3
X2
X1
X0
Y3
Y2
Y1
Y0
Great
Equal
Less



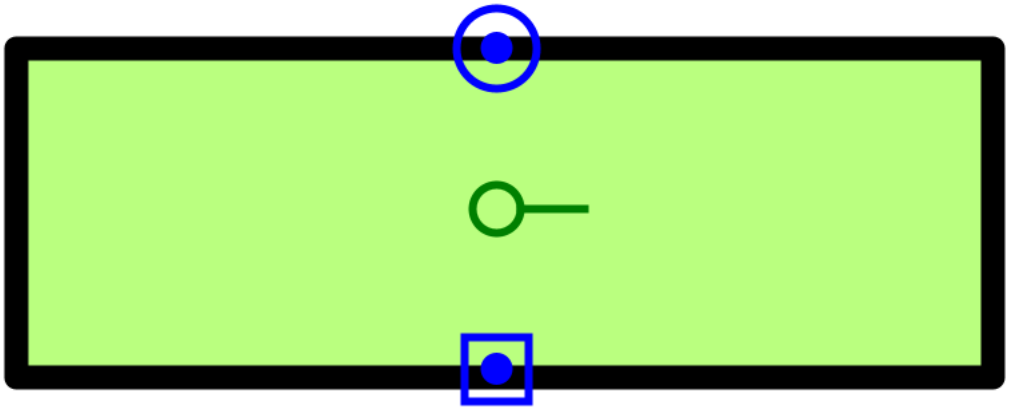




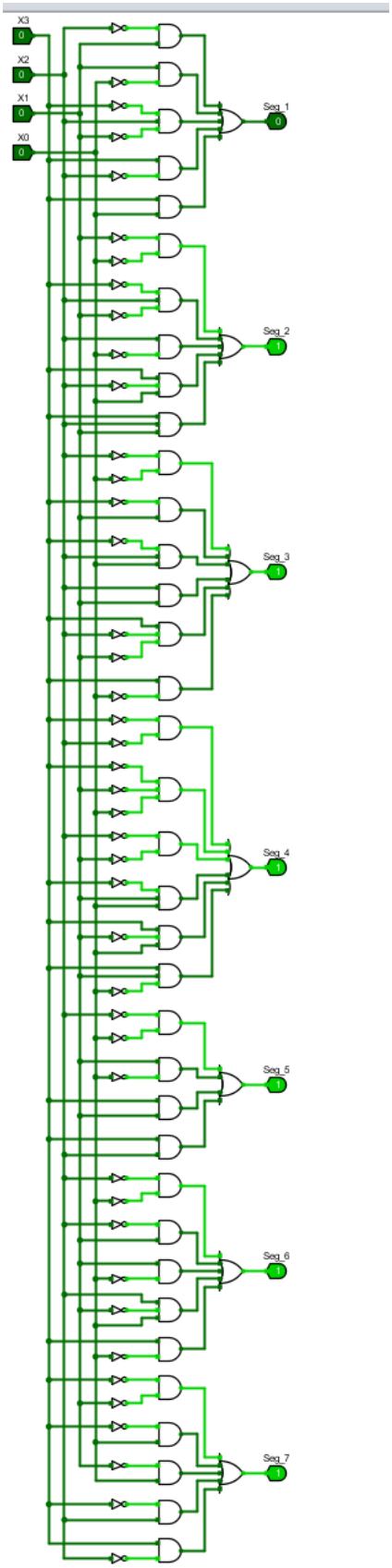


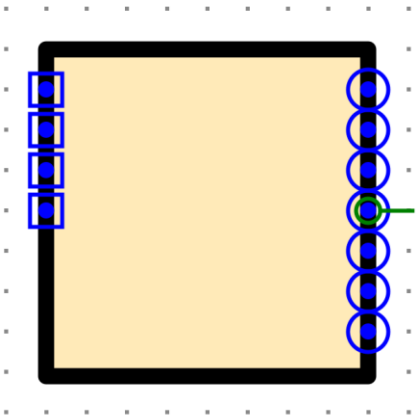
16位数无符号比较器自动测试

码表数码管驱动

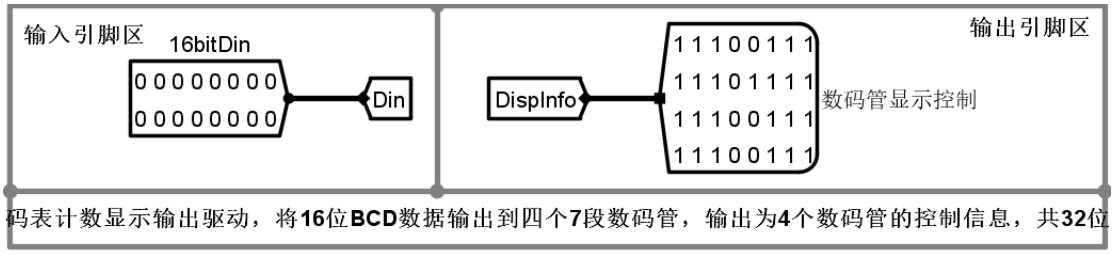


输入：16 位的 BDC 码（4 位的 10 进制数）
输出：4 个 7 段数码管的控制信号（共 32 位）
设计要点 利用已经过构建的一个 7 段数码管，并发形成 4 个 7 段数码管的驱动
数码管驱动如下

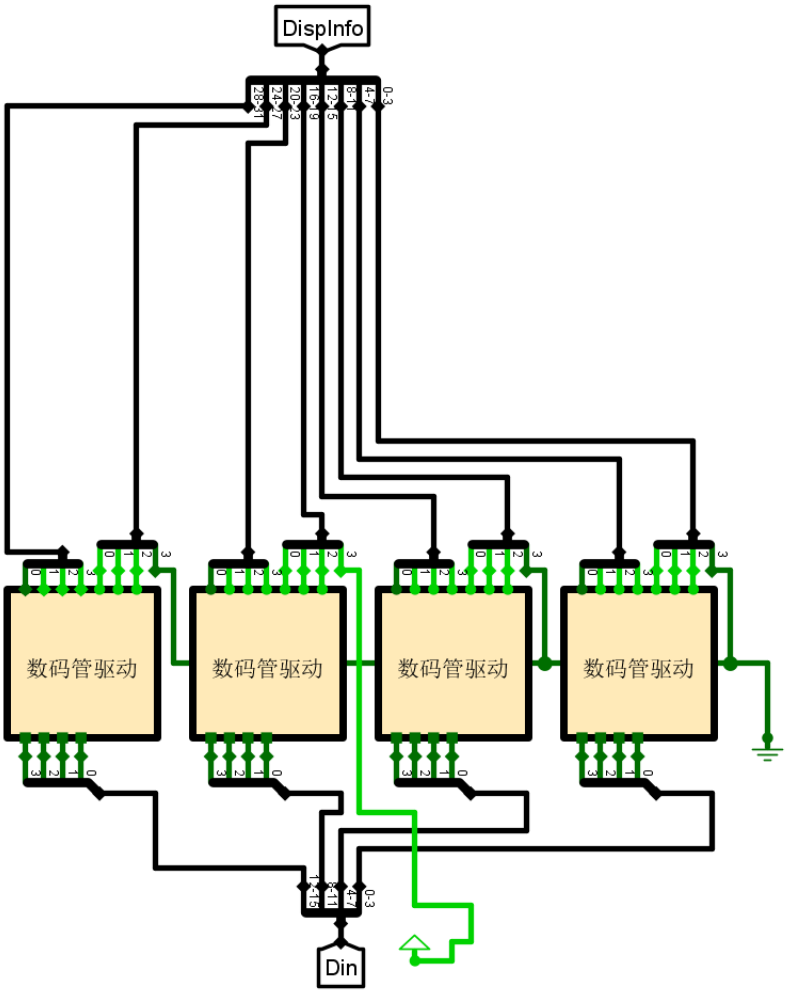


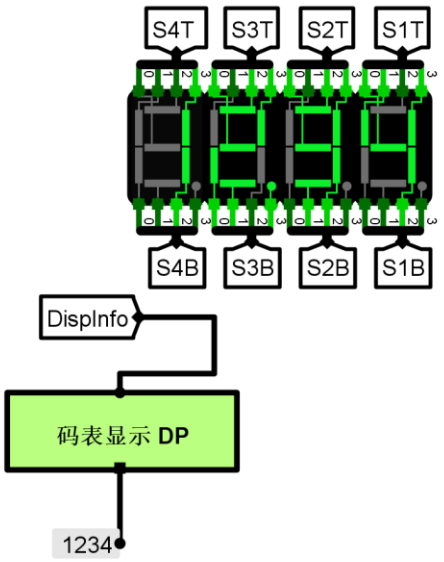


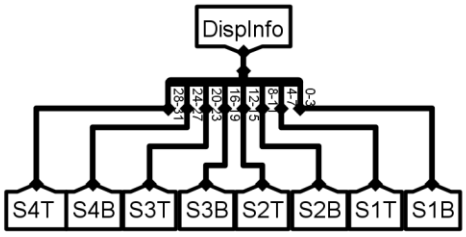
码表驱动如下



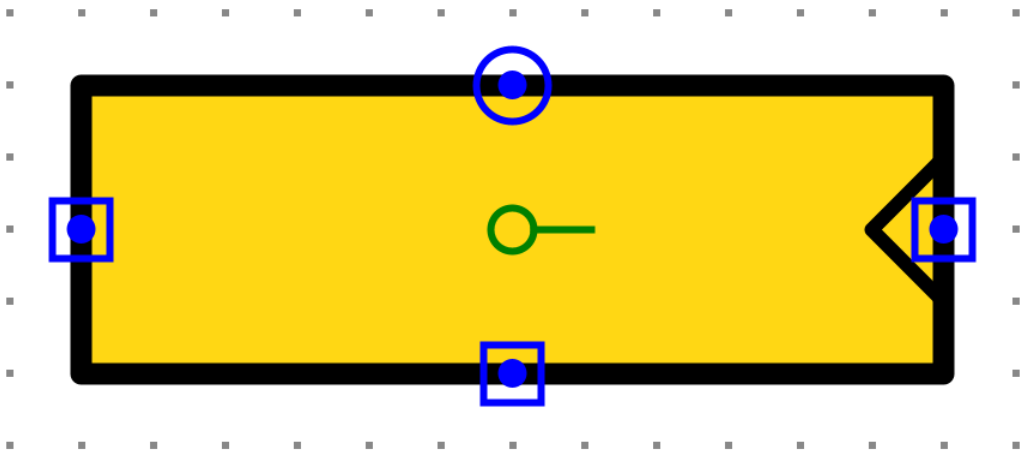
码表计数显示输出驱动，将**16**位**BCD**数据输出到四个**7**段数码管，输出为**4**个数码管的控制信息，共**32**位



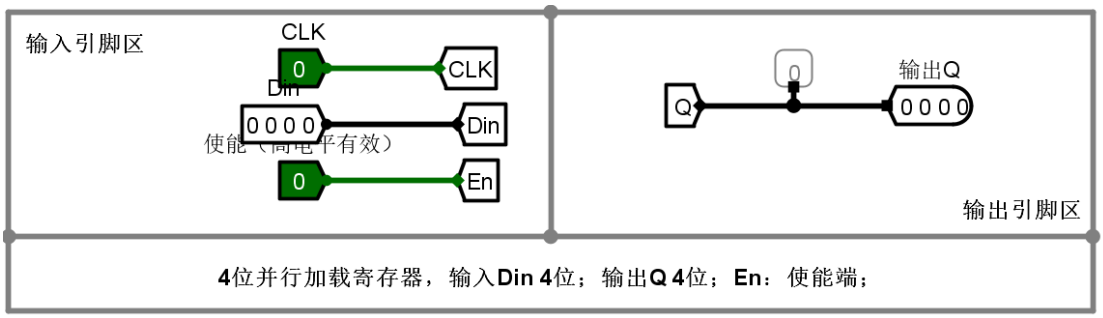




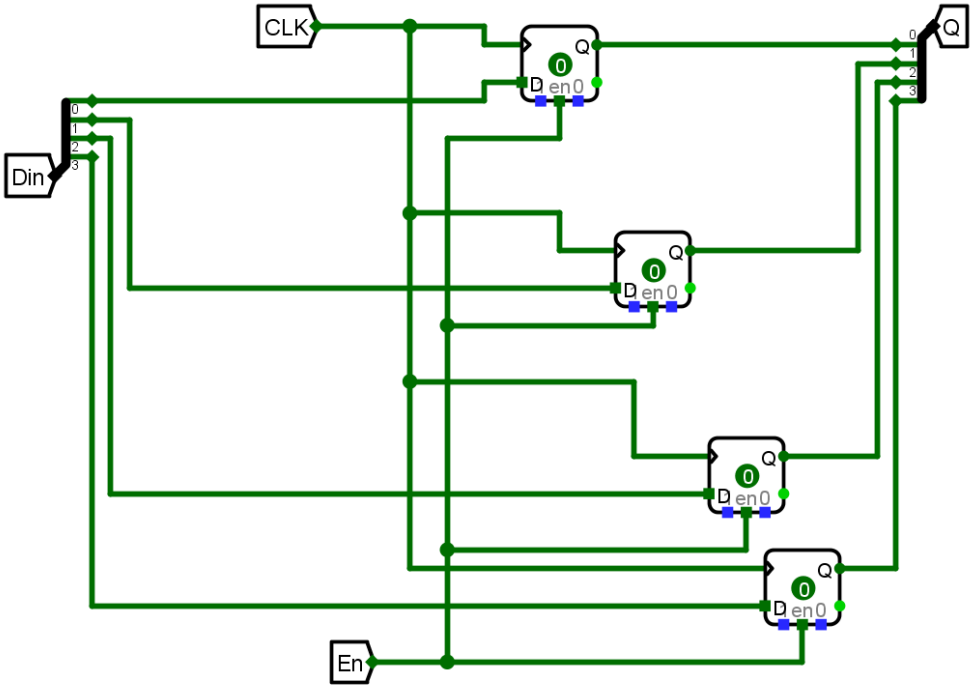
16 位寄存器

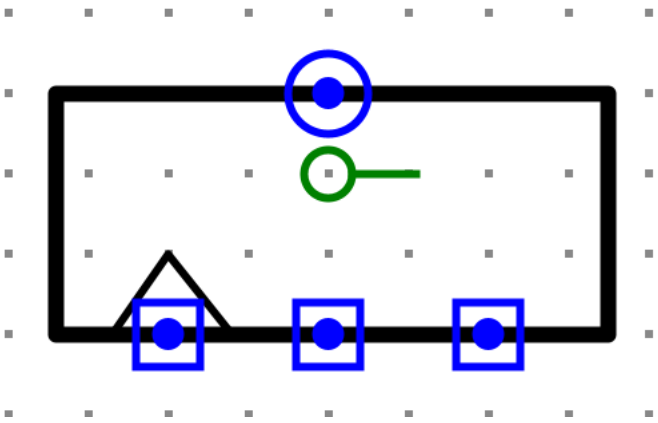


输入：16 位的输入 Din，使能信号 En
输出：16 位的输出 Q
功：Din -> Q
设计要点 先构建 4 位并行加载寄存器，再并联接形成一个 16 位并行加载寄存器 4 位并行加载寄存器
四位并行加载寄存器如下

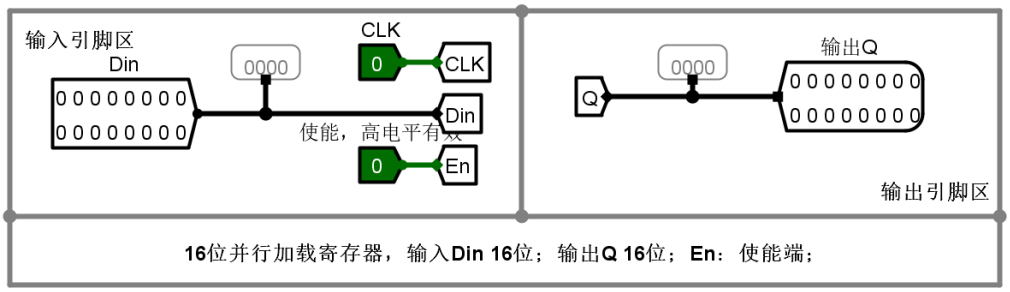


请勿增删改引脚，请在下方利用上方输入输出引脚的隧道标签信号构建电路，ctrl+d复制选择组件

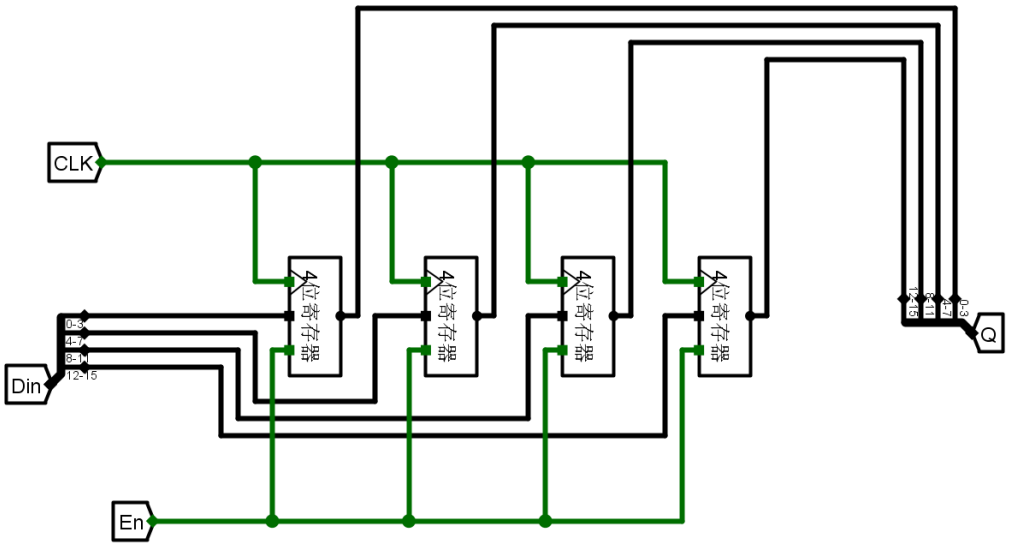




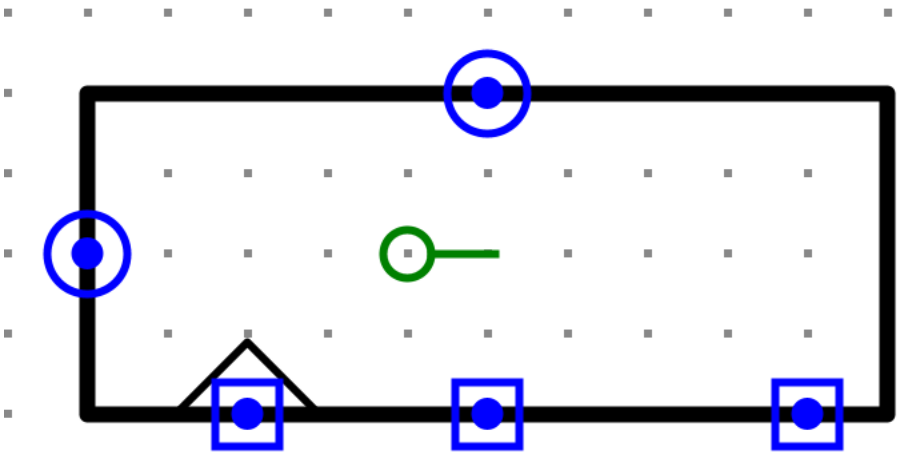
16 位并行加载寄存器如下



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



4 位 BCD 计数器

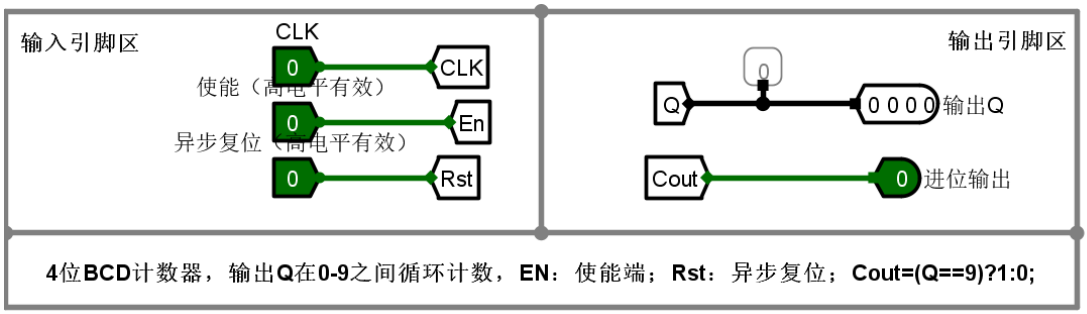


输入：时钟信号 CLK，使能信号 En，异步复位 Rst
输出：4 位输出 Q，进位输出信号 Cout

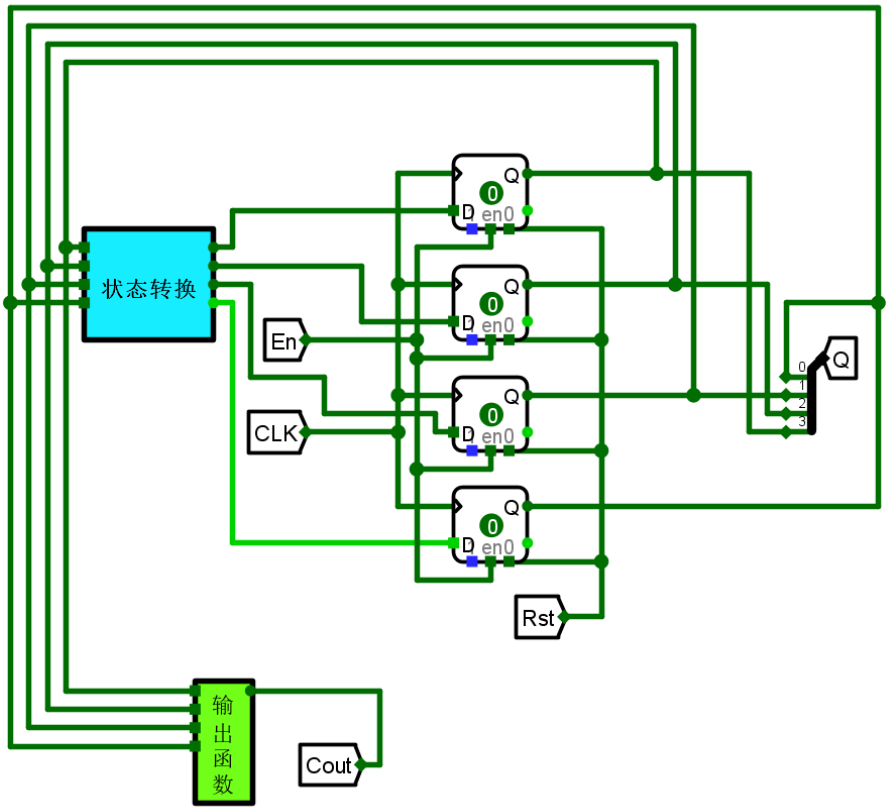
Q从9->0时，产生进位信号Cout

设计要点 需要设计两个子电路：“状态转换”子电路以及“输出函数”子电路

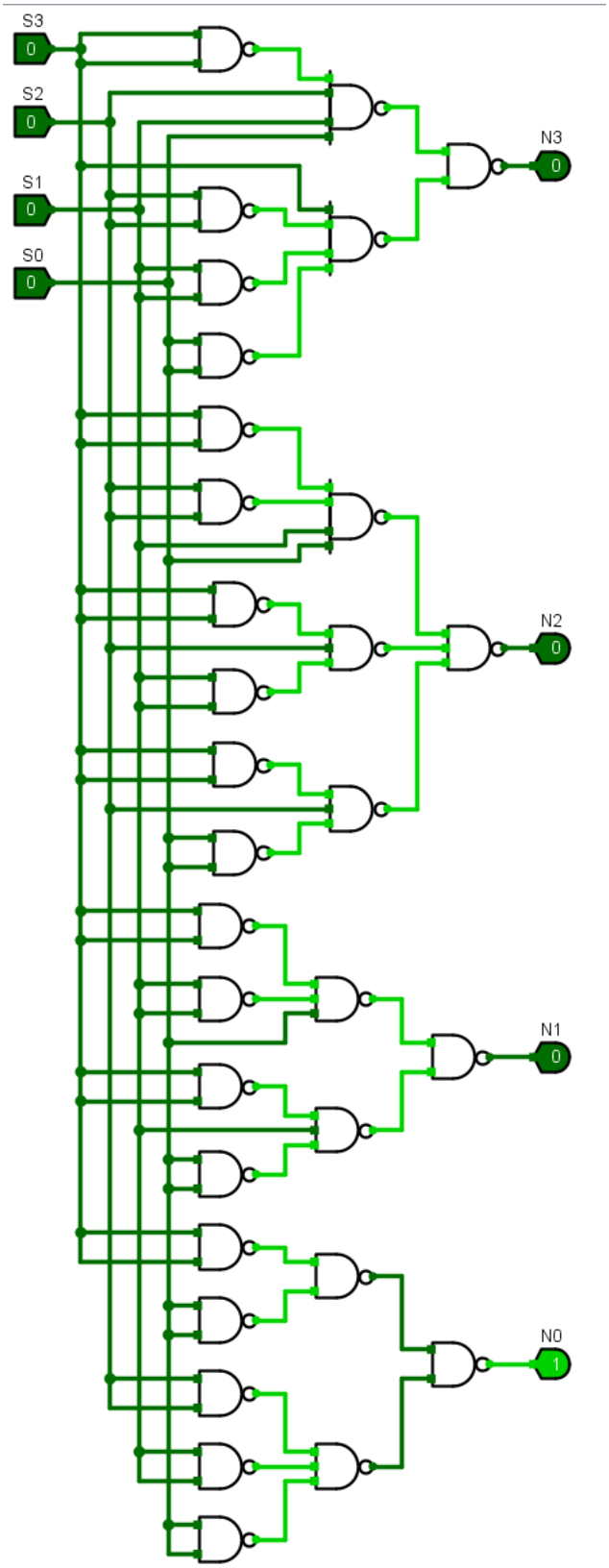
总电路



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



状态转换



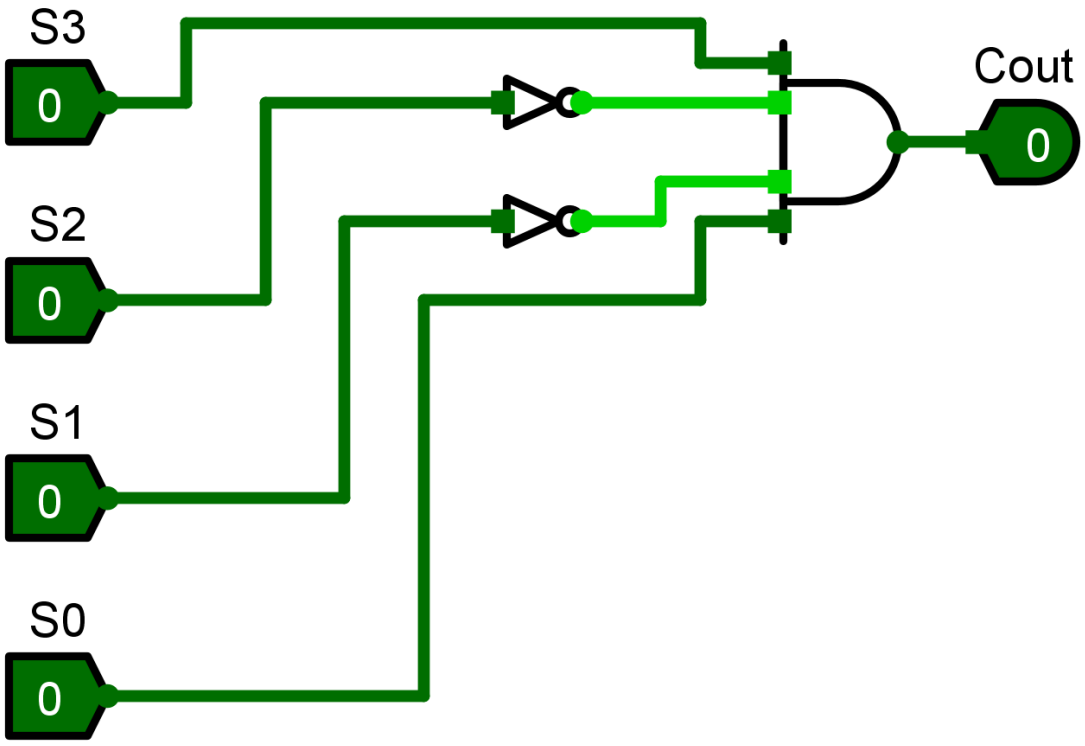
“状态转换”子电路： 1、填写“状态转换表”2、检查“触发器输入函数自动生成”的逻辑表达式 3、打开 logisim 中“BCD 计数器状态转换（自动生成）”文件，利用电路分 析自动生成 电路图

N3=~S3 S2 S1 S0 + S3 ~S2 ~S1 ~S0

N2=~S3 ~S2 S1 S0 + ~S3 S2 ~S1 + ~S3 S2 ~S0
N1=~S3 ~S1 S0 + ~S3 S1 ~S0
N0=~S3 ~S0 + ~S2 ~S1 ~S0

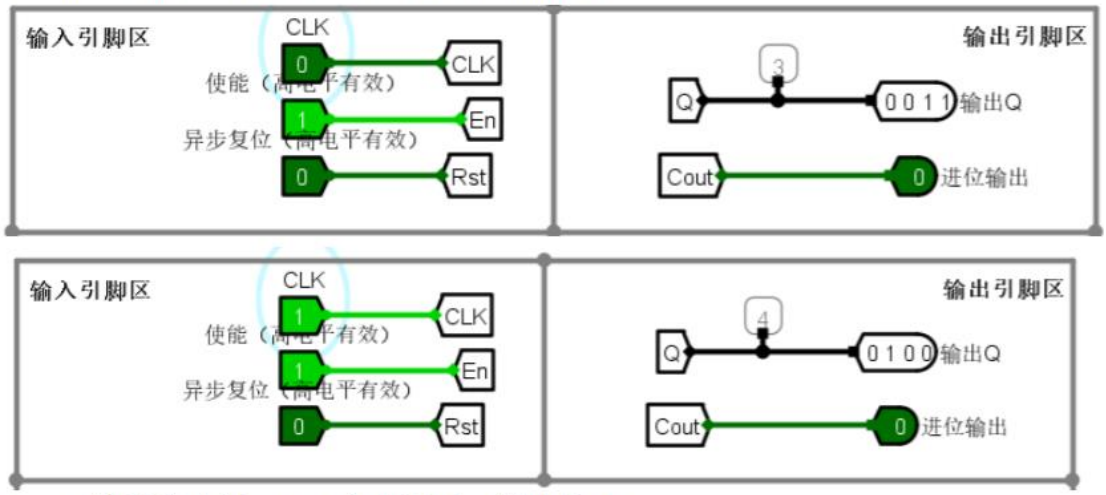
S3	S2	S1	S0	N3	N2	N1	N0
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	1	1	0	0	0	0

输出函数

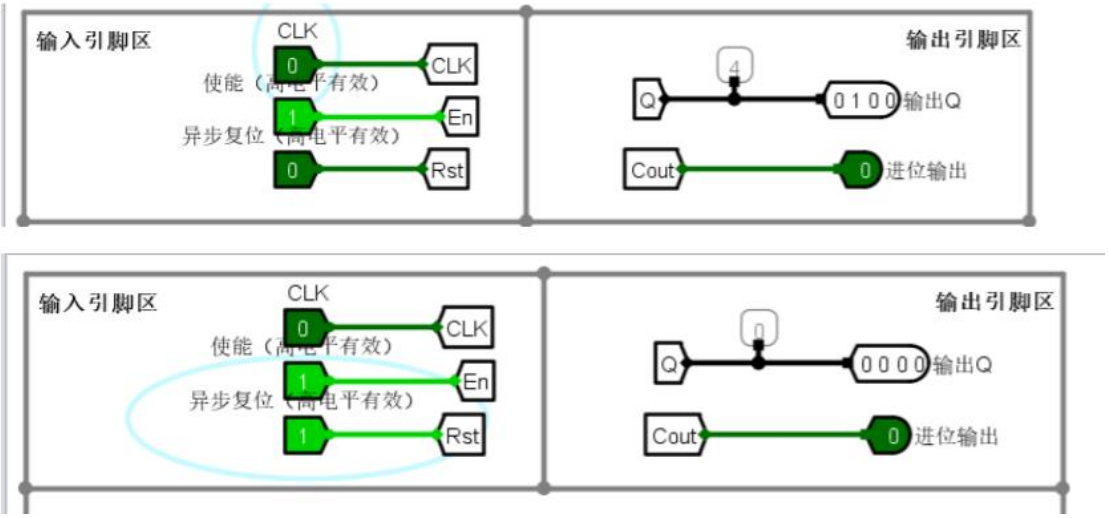


“输出函数”子电路： 1、写出真值表 or Cout 的逻辑表达式 2、打开 logisim 中“BCD 计数器输出函数（自动生成）”文件，利用电路分 析功能自动生成电路

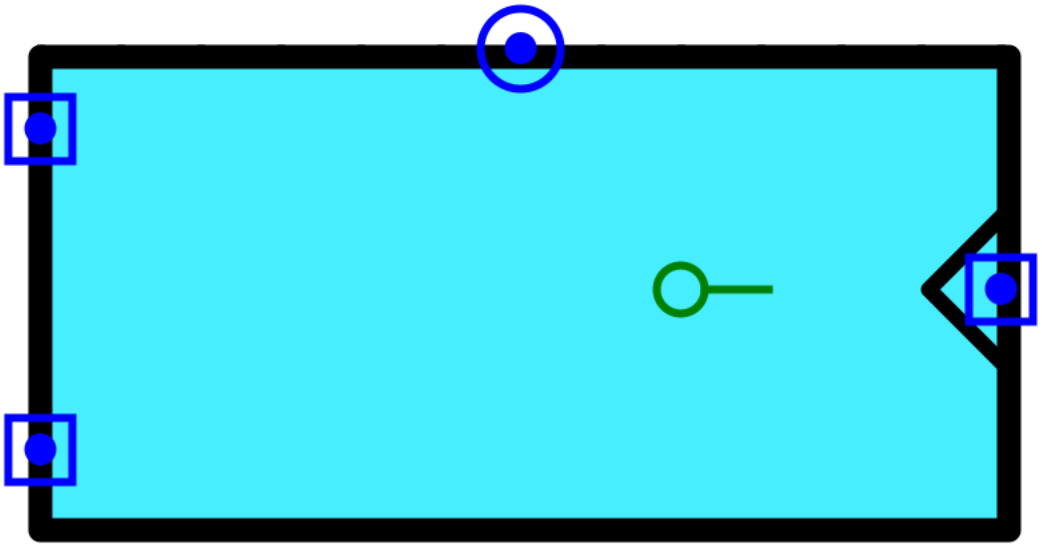
1、En 处于高电平，有 CLK 进入时，输出+1



2、En 处于高电平。Rst 高电平时，输出归 0

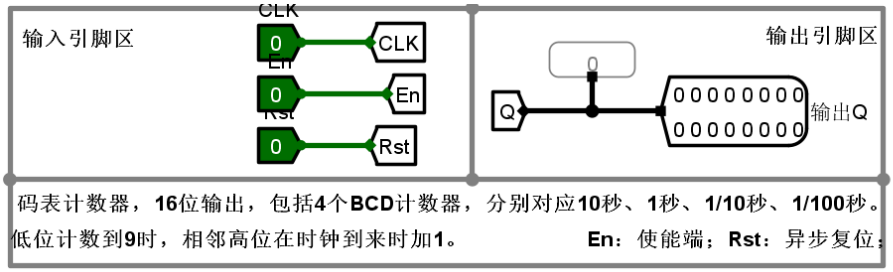


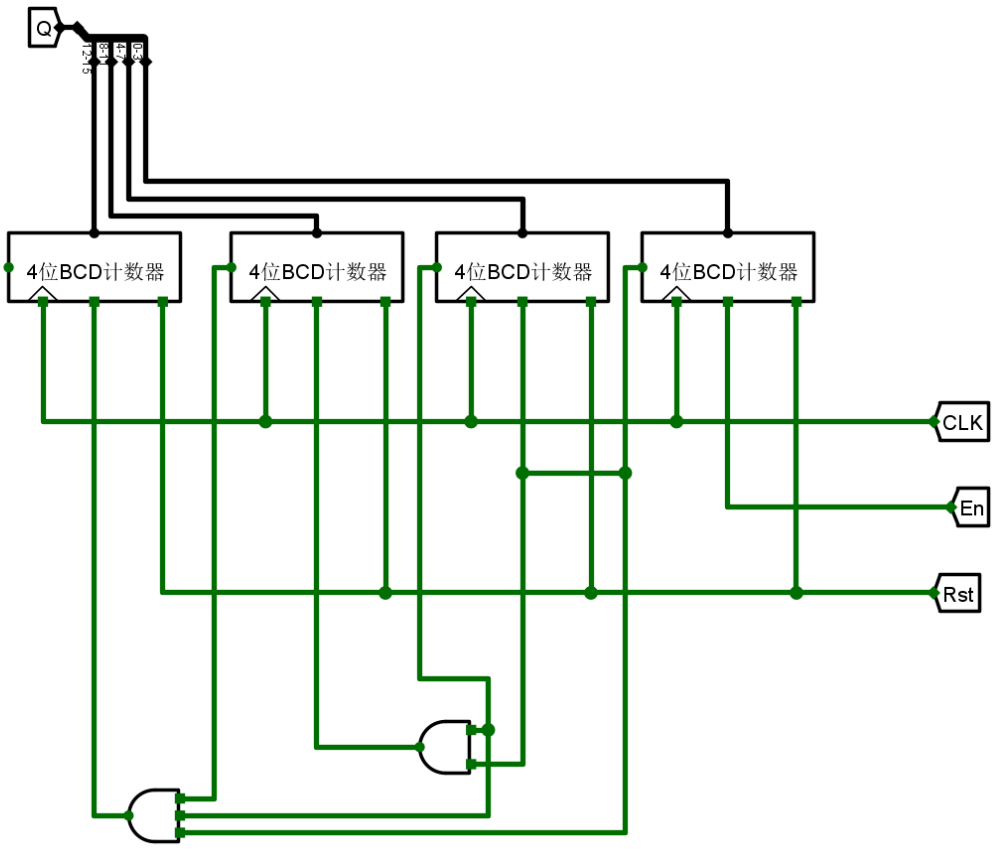
码表计数器

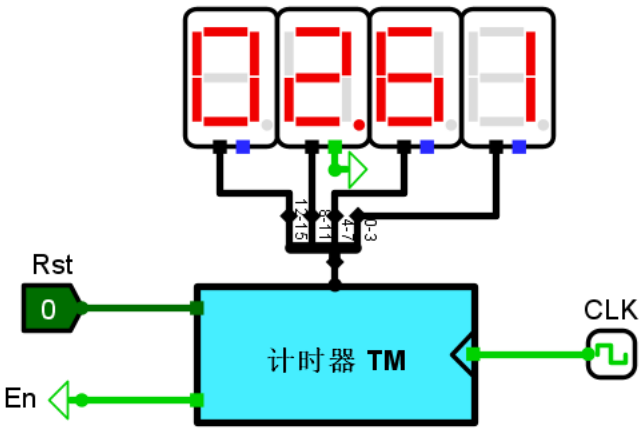


输入：时钟信号 CLK，使能信号 En，异步复位 Rst

输出：16 位输出 Q 结构：包含 4 个 BCD 码计数器
功能：低位计数器从 9 到 0 时，相邻高位计数器+1
设计要点 利用 4 个 4 位的 BCD 计数器级联而成







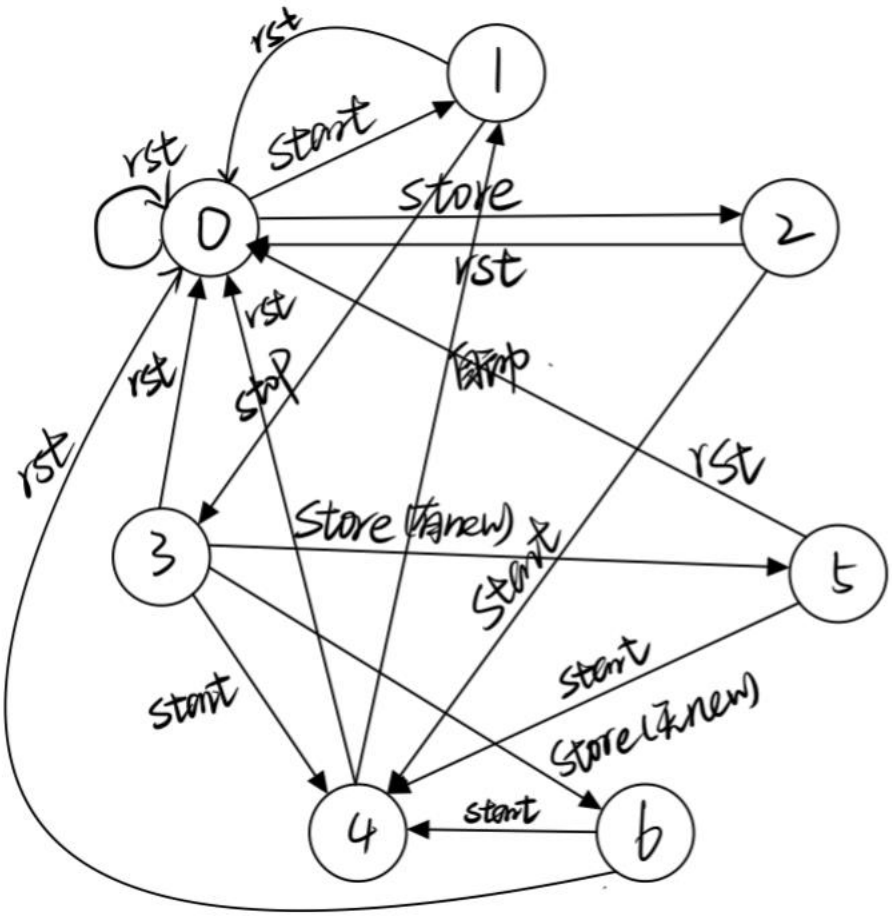
Ctrl+K驱动时钟自动测试

2. 数据通路分析与设计

#	功能部件	控制信号	输入	输出
1	时间计数器TM	TM-En,TM-Rst	CLK	时间计数输出16位
2	16位寄存器SD	SD-En	CLK，Din(16位)	Q(16位)
3	数码管显示DP		Din(16位)	DisplayInfo(32位)
4	比较器			
5	2路选择器	Sel		

3. 构建控制电路

3.1 数字码表的状态设计

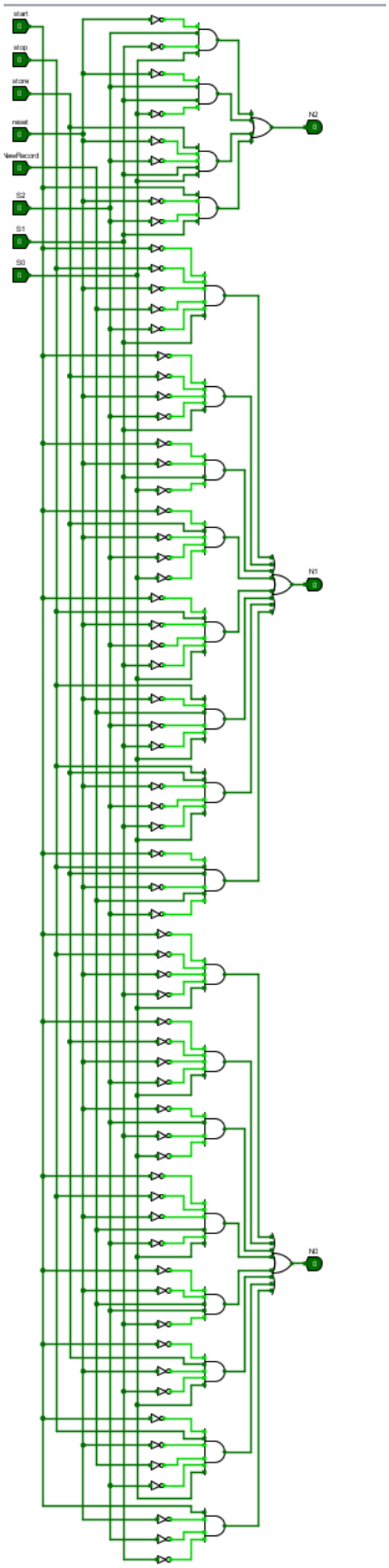


3.2 码表控制状态转换子电路的设计

N2= ~reset S2 ~S1 S0 + ~reset S2 S1 ~S0 + store ~reset ~S2 S1 S0 + start ~reset ~S2 S1

N1= ~start ~stop ~reset ~NewRecord ~S2 S1 + ~start ~store ~reset ~S2 S1 + ~start ~reset S1 ~S0 + ~start store ~reset ~S2 ~S0 + ~start stop ~reset ~S2 ~S1 S0 + stop ~reset NewRecord ~S2 ~S1 S0 + stop store ~reset ~S2 ~S1 S0 + ~start stop store ~reset NewRecord ~S2

N0= ~start ~stop ~reset ~S1 S0 + ~start ~store ~reset ~S2 S0 + ~reset S2 ~S1 ~S0 + ~start ~stop ~reset NewRecord ~S2 S0 + ~start ~reset NewRecord S2 ~S1 + ~start store ~reset ~S1 S0 + ~start stop ~reset ~NewRecord ~S2 S0 + start ~reset ~S2 ~S1



3.3 码表控制输出函数子电路的设计

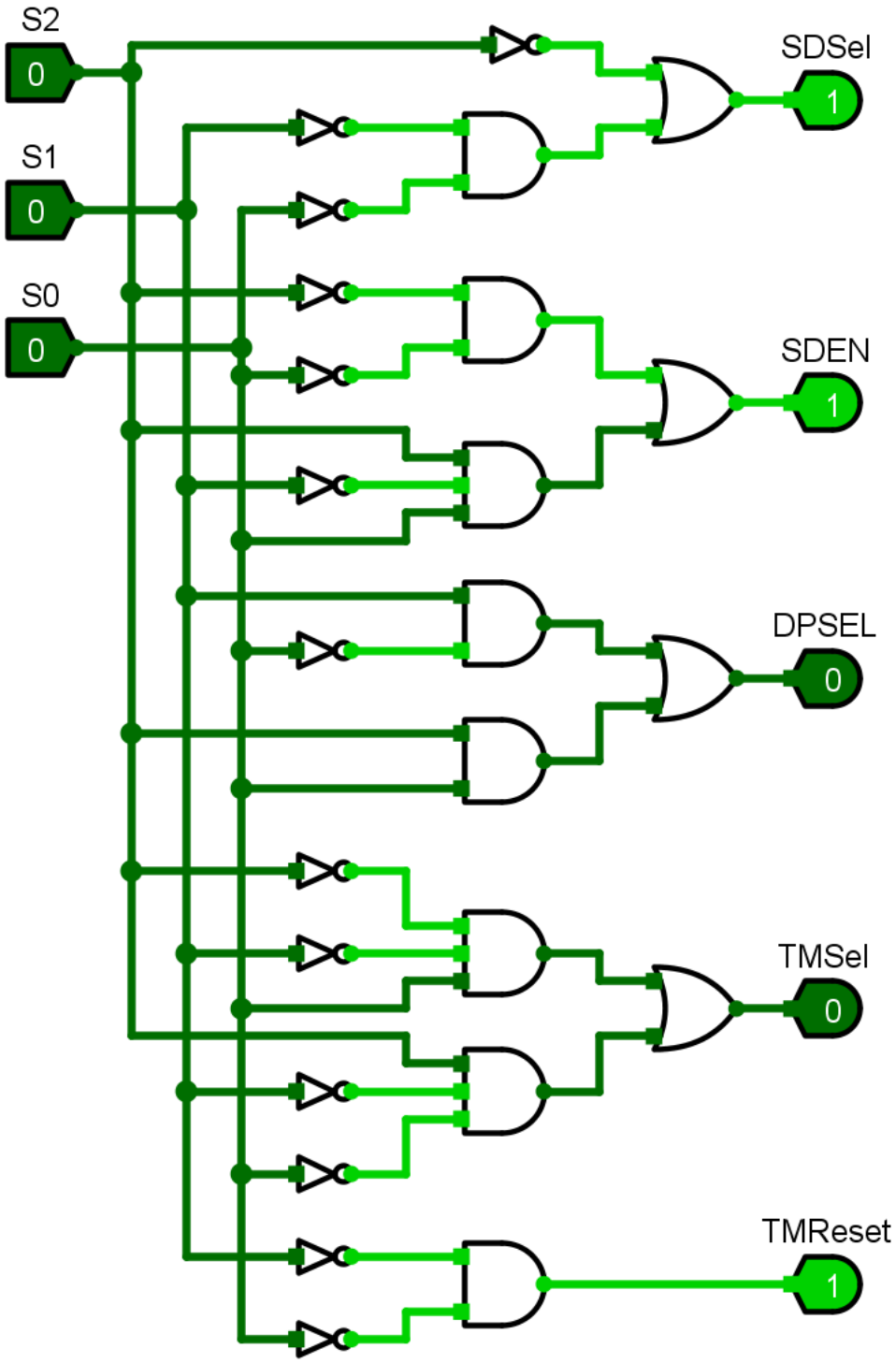
SDSel= ~S2 + ~S1 ~S0

SDEN= ~S2 ~S0 + S2 ~S1 S0

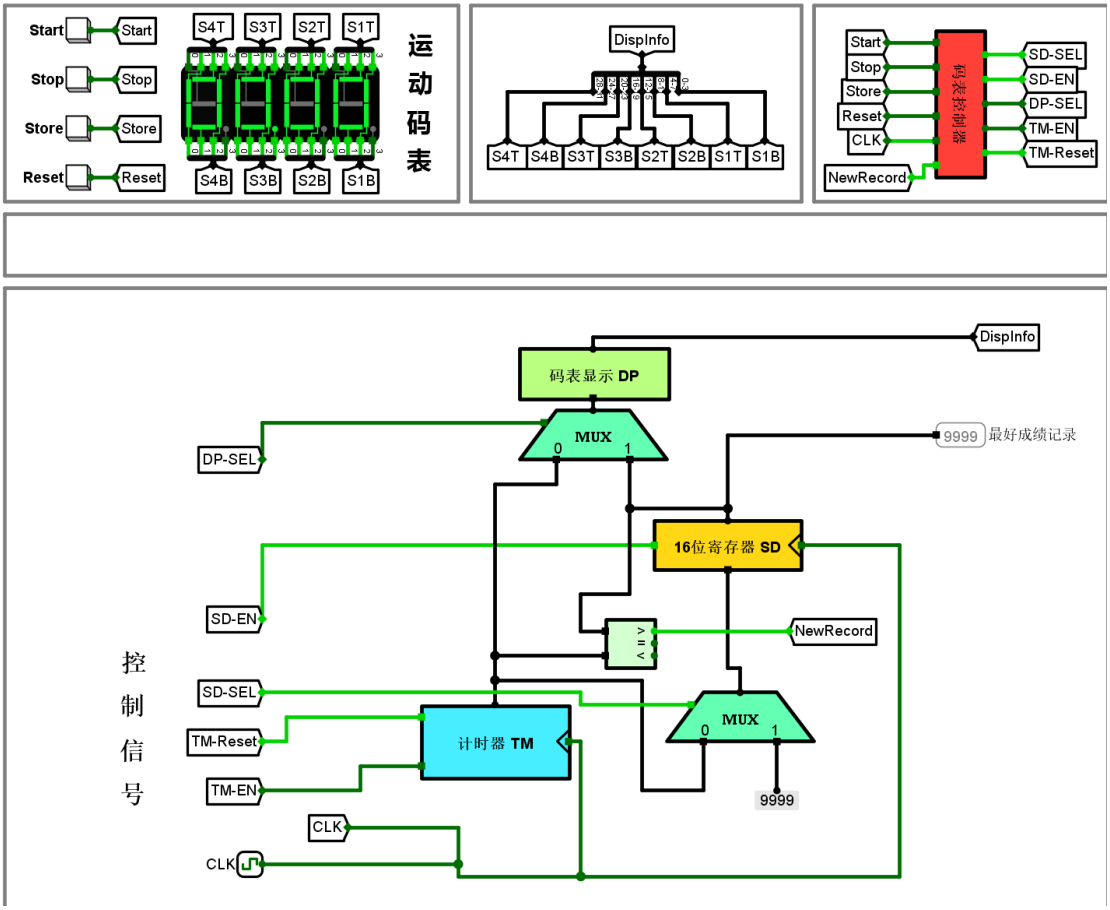
DPSEL= S1 ~S0 + S2 S0

TMEN= ~S2 ~S1 S0 + S2 ~S1 ~S0

TMReset= ~S1 ~S0

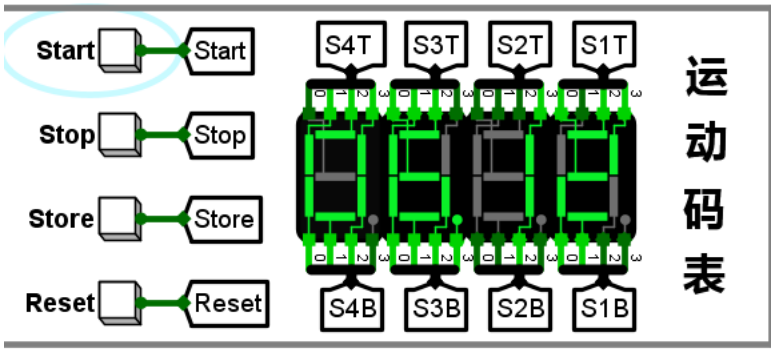


3.4 数字码表控制电路整体连接

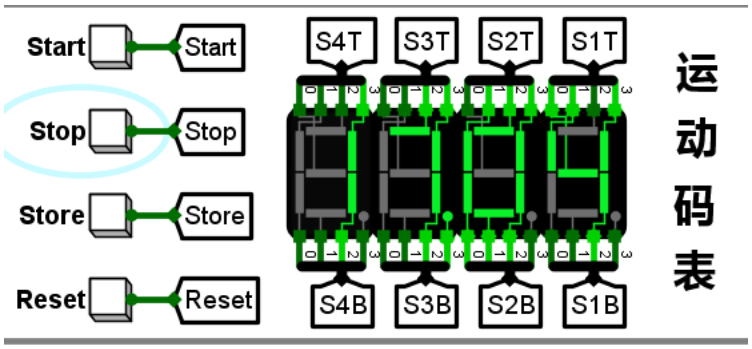


4. 系统集成调试

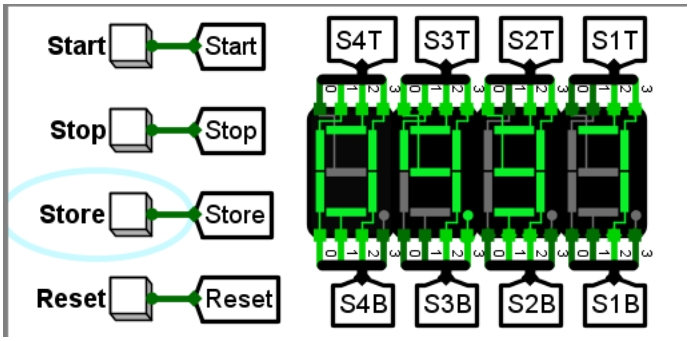
Start



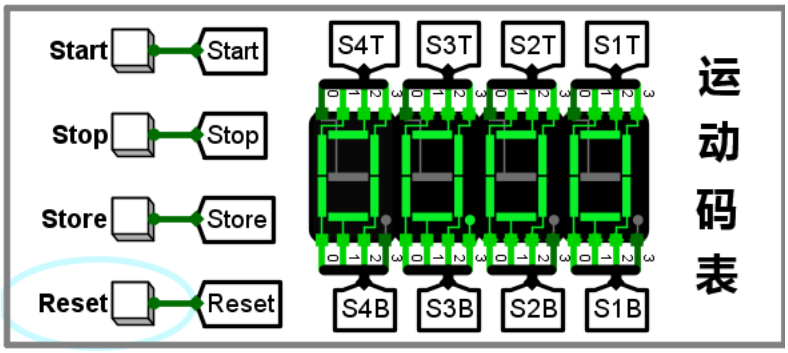
Stop



Store



Reset



四、实验心得体会

重启重装解决 99%问题

有时候并非设计错误，重新添加元件，复位电路，重启软件，就搞定了