
1 实验概述

1.1 实验名称

运动码表系统设计。

1.2 实验目的

本实验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四位比较器，多路选择器，利用同步时序逻辑构建 BCD 计数器，从简单的组合逻辑电路到复杂时序逻辑电路，最终集成实现为运动码表系统。

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：Logisim2.15.0.2 软件一套。

平台：https://www.educoder.net/classrooms/11930/shixun_homework

1.4 实验内容

设计一个运动码表系统，具体内容及要求如下：

输入：4 个按钮，分别为 Start、Stop、Store 和 Reset。

输出：4 个 7 段数码管显示数字，分别显示小时和分钟。

具体功能：

(1) 当按下 Start 时，计时器清零，重新开始计时；

(2) 当按下 Stop 时，计时器停止计时，显示计时数据；

(3) 当按下 Store 时，若当前计时数据小于系统记录，则更新系统记录，并显示当前计时数据；否则不更新系统记录，但显示系统记录。

(4) 当按下 Reset 时，复位，计时=0.00，系统记录=99.99。

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 实验过程

2.1 7 段数码管驱动电路设计

(1) 设计思路及设计过程

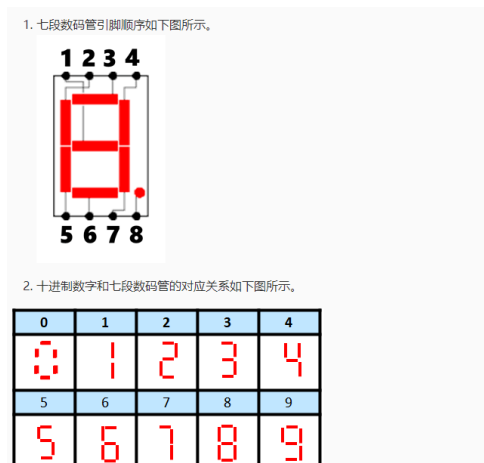


图 2.1-1 七段数码管

根据七段数码管的引脚顺序和十进制数字与七段数码管的对应关系，找到每个数字输入与显示的数码管的关系，利用真值表，来直接生成电路。

(2) 电路图

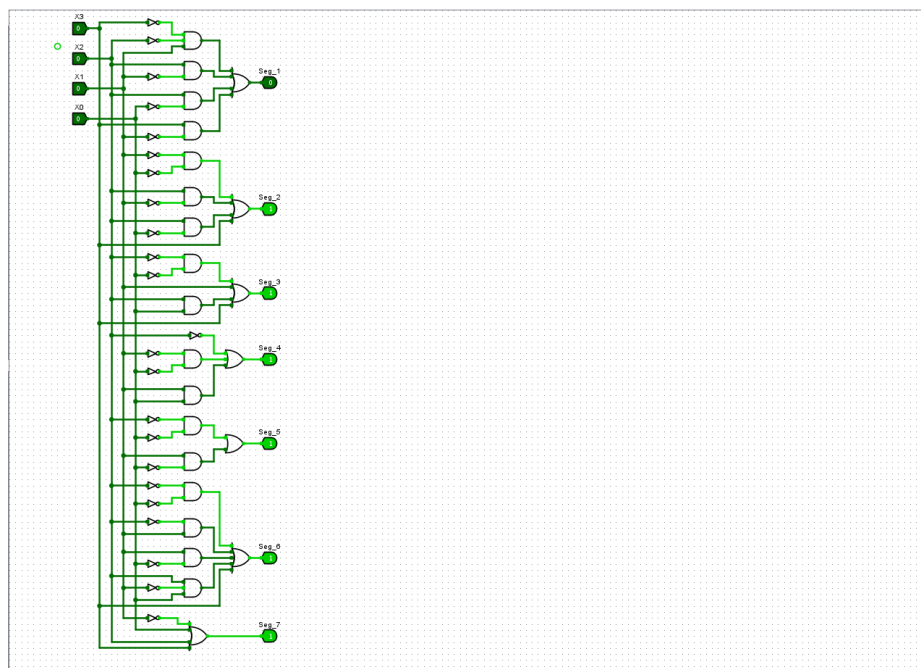


图 2.1-2 七段数码管电路图

(3) 测试图

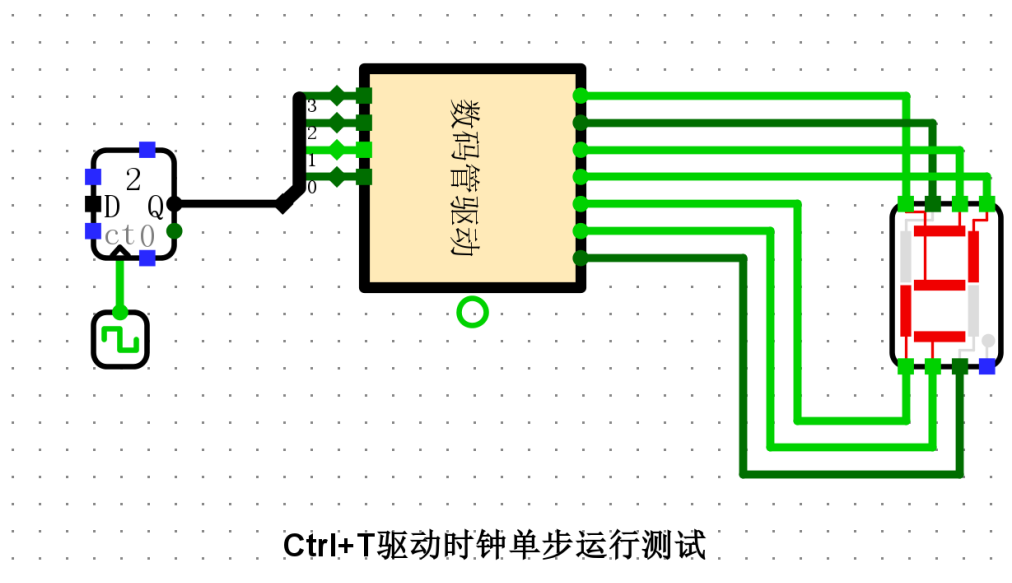


图 2.1-3 数码管驱动测试（一）

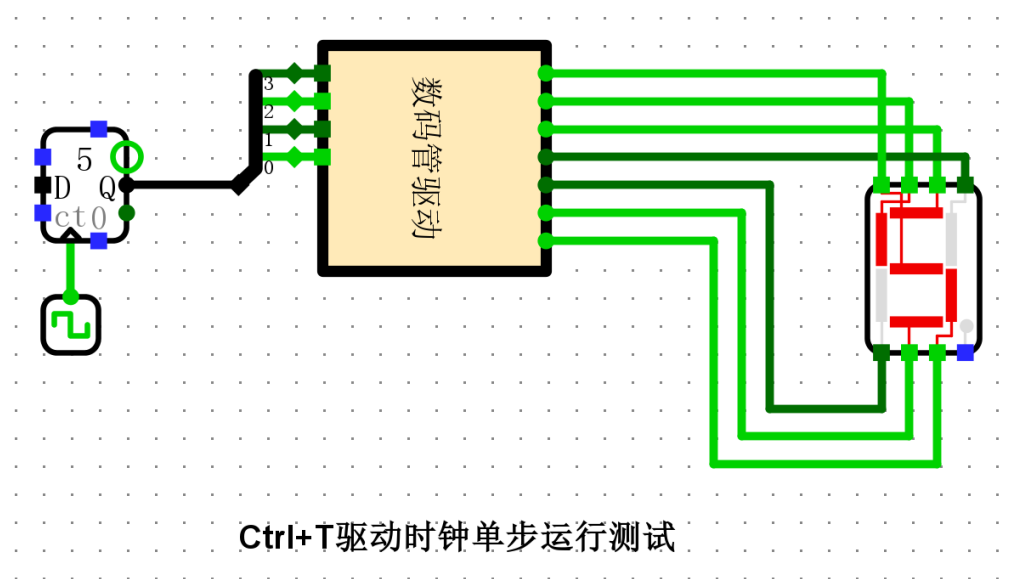


图 2.1-4 数码管驱动测试（二）

(4) 测试分析

本地测试效果如图所示，效果良好，并且顺利通过测试平台测试，顺利完成第一关。

2.2 2 选 1 选择器设计

(1) 设计思路及设计过程

根据电路功能 $Out=(Sel==0)?X0:X1$;直接赋值真值表, Sel 为 0 的时候让 $Out=X0$, Sel 为 1 的时候让 $Out=X1$, 生成电路即可。十六位的电路中, 将一位的电路先原封不动的照搬过来, 然后将与门都改成十六位, 再用分线器连接 Sel 输入即可。

(2) 电路图

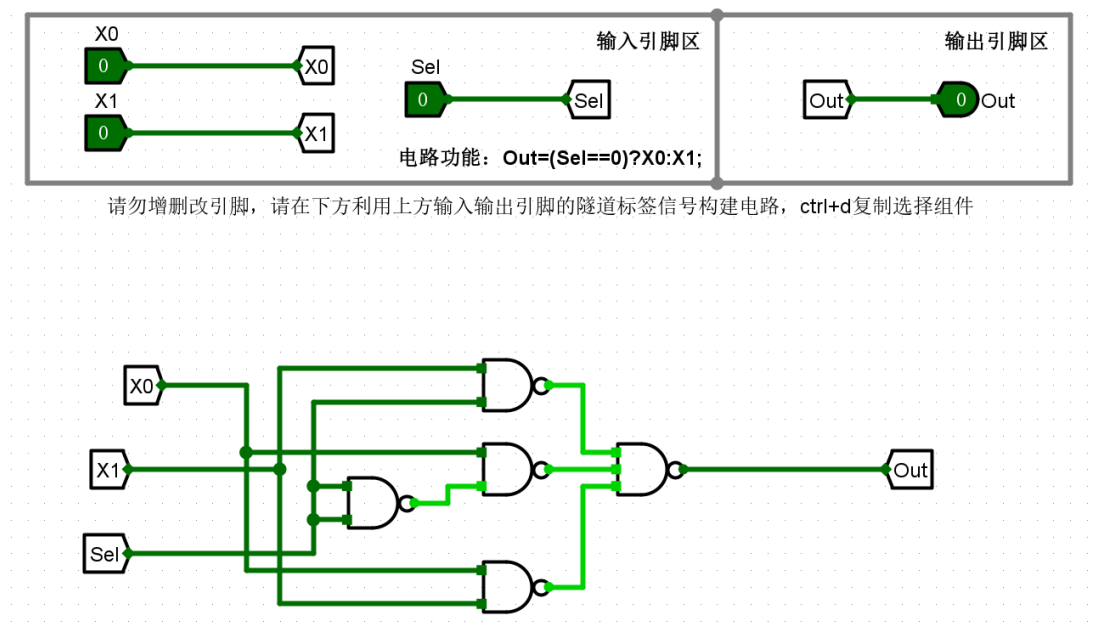


图 2.2-1 一位选择器电路图

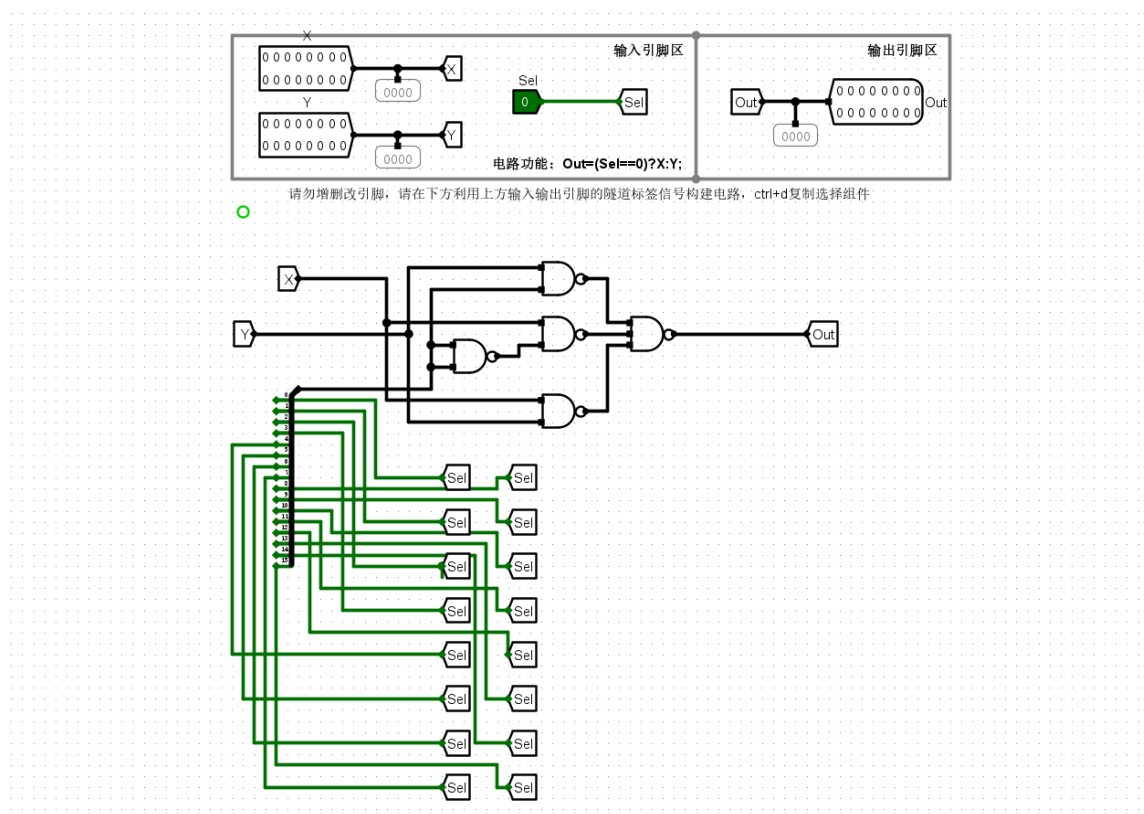


图 2.2-2 十六位选择器电路图

(3) 测试图

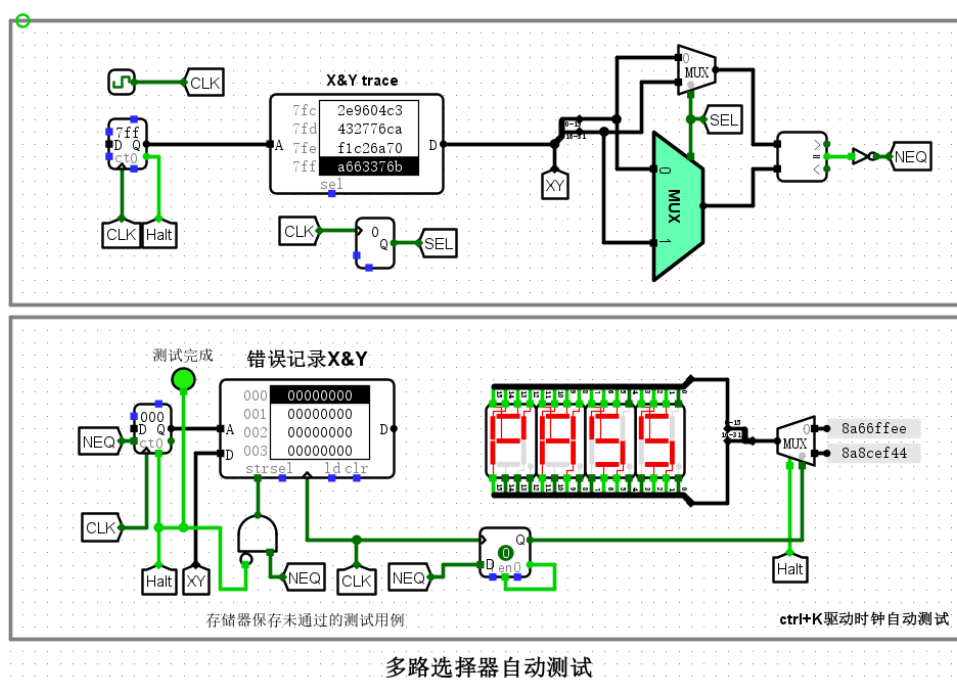


图 2.2-3 二路选择器测试图

(4) 测试分析

通过本地测试图可以发现，显示为 PASS，说明测试通过，效果良好，且通过平台测试，说明此二路选择器设计比较成功，成功通过第二关。

2.3 无符号比较器设计

(1) 设计思路及设计过程

四位无符号比较器：

1 位比较 $X:Y$ $\text{Great} = X \sim Y$ 说明：输入为 0， $\text{Less} = \sim X Y$ ，说明：输入为 01， $\text{Equal} = \sim (X \wedge Y) = (\sim X \sim Y + X Y)$ ，说明：输入为 00 或者 11，即输入相同，2 位比较 $X=X_1X_0:Y=Y_1Y_0$ ， $\text{Great} = X_1 \sim Y_1 + \sim (X_1 \wedge Y_1)X_0 \sim Y_0$
说明：X 的最高位 X_1 大于 Y 的最高位 Y_1 ，或者最高位相等时 X 的低位 X_0 大于 Y 的低位 Y_0 ， $\text{Less} = \sim X_1 Y_1 + \sim (X_1 \wedge Y_1) \sim X_0 Y_0$ ，说明：X 的最高位 X_1 小于 Y 的最高位 Y_1 ，或者最高位相等时 X 的低位 X_0 小于 Y 的低位 Y_0 ， $\text{Equal} = \sim (X_1 \wedge Y_1) \sim (X_0 \wedge Y_0)$ ，说明：X 的最高位 X_1 等于 Y 的最高位 Y_1 ，而且，X 的低位 X_0 等于 Y 的低位 Y_0 ，3 位、4 位比较依此类推。

十六位无符号比较器：

考虑四个四位比较器，将十六位比较拆分为四个四位比较，X 和 Y 相等只有一种情况，X 和 Y 的四个四位分别相等。 $X > Y$ 有四种情况，X 的高 4 位大于 Y 的高 4 位，X 的高 4 位等于 Y 的高 4 位且 X 的第二 4 位大于 Y 的第二 4 位，X 的高 4 位等于 Y 的高 4 位且 X 的第二 4 位等于 Y 的第二 4 位且 X 的第三 4 位大于 Y 的第三 4 位，X 的高 4 位等于 Y 的高 4 位且 X 的第二 4 位等于 Y 的第二 4 位且 X 的第三 4 位等于 Y 的第三 4 位且 X 的末 4 位大于 Y 的末 4 位，X 小于 Y 等同分析即可。

(2) 电路图

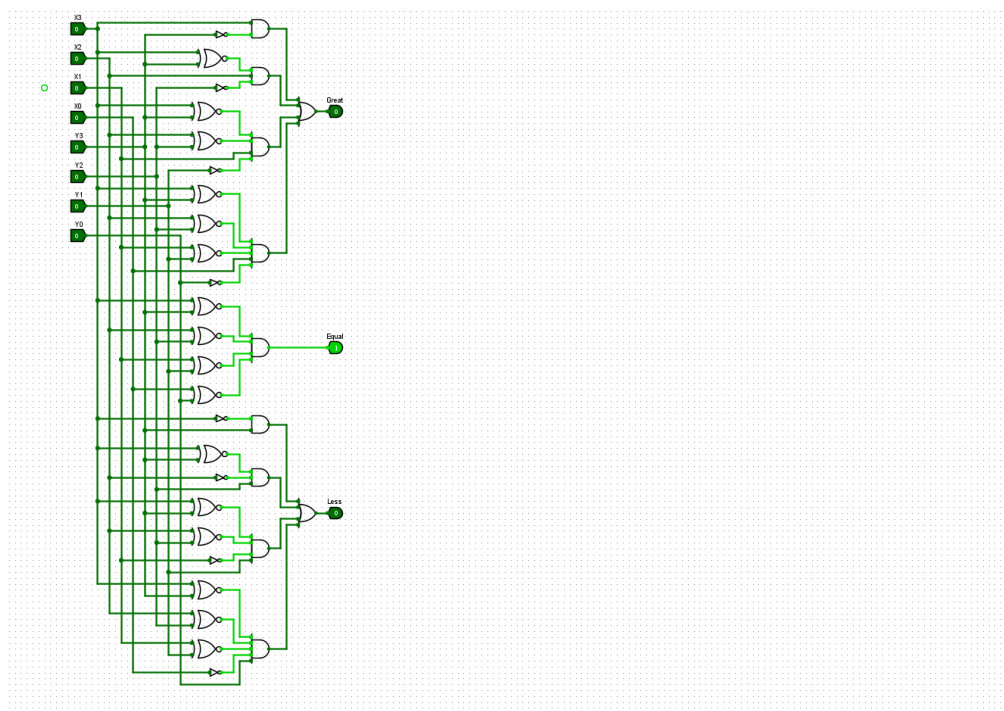


图 2.3-1 无符号比较器电路图（一）

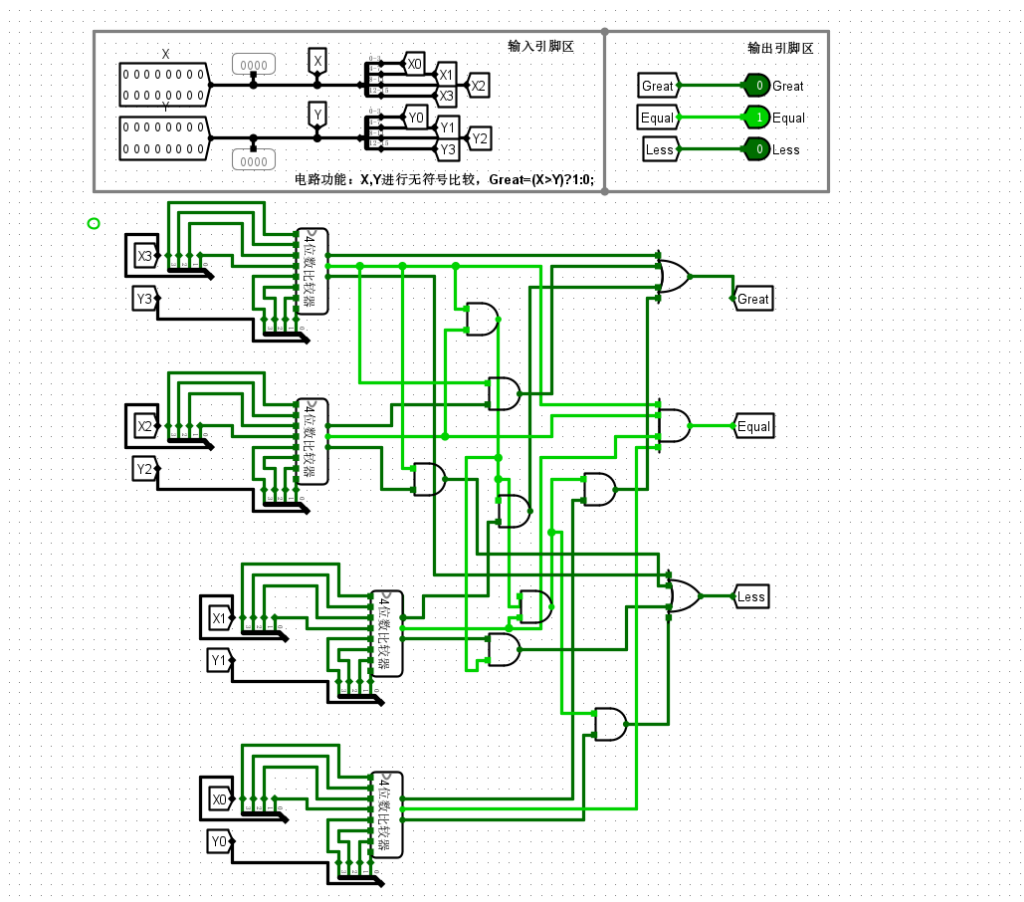
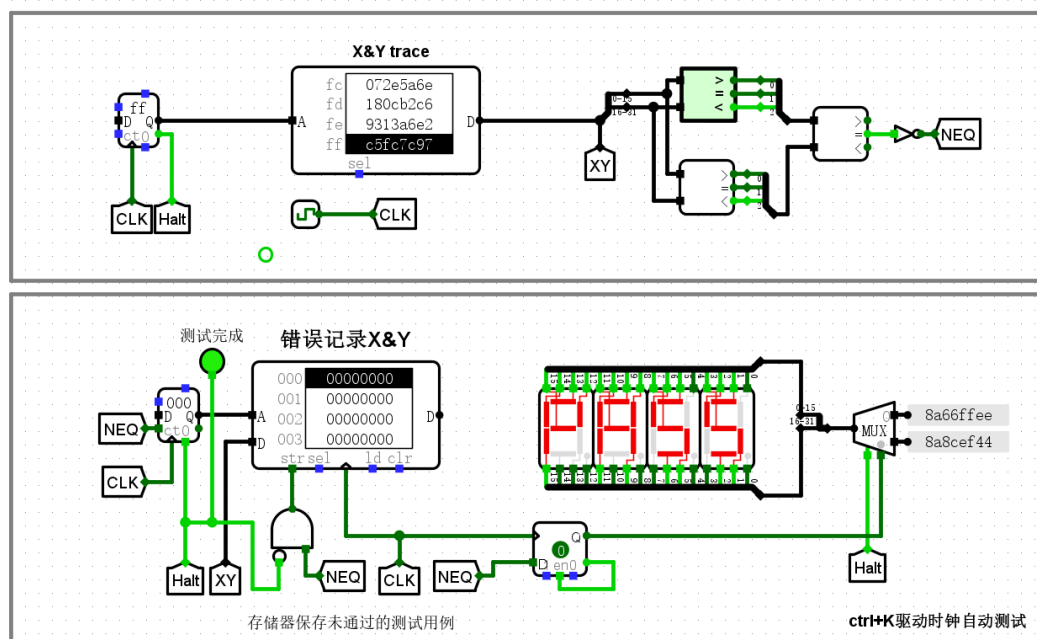


图 2.3-2 无符号比较器电路图（二）

(3) 测试图



16位数无符号比较器自动测试

图 2.3-3 无符号比较器测试图

(4) 测试分析

通过本地测试图可以发现，显示为“PASS”，说明测试通过，效果良好，且通过平台测试，说明此无符号比较器设计比较成功，能够通过各种测试样例，成功通过第三关。

2.4 并行加载寄存器

(1) 设计思路及设计过程

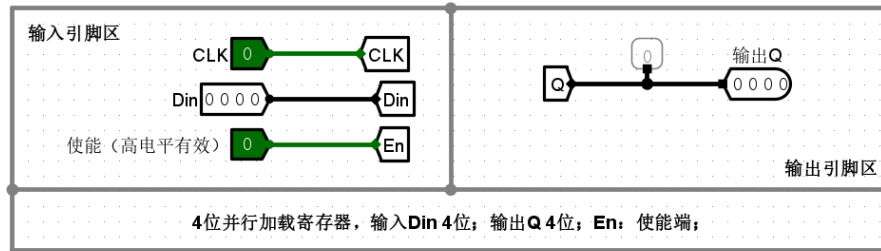
四位并行加载寄存器

Din 为寄存数据输入端，En 为使能输入端，高电平有效，CLK 为时钟脉冲，上升沿有效，根据电路功能，让使能端分别与相应的端口项链，输出用分线器分开，再连到 Q 即可。

十六位并行加载寄存器

用四个四位并行加载寄存器，连接 CLK，En，Din 到相对应的端口，输出用十六位分线器分开连到 Q 即可。

(2) 电路图



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

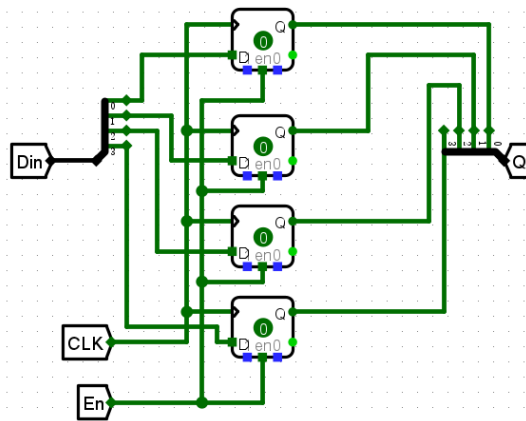
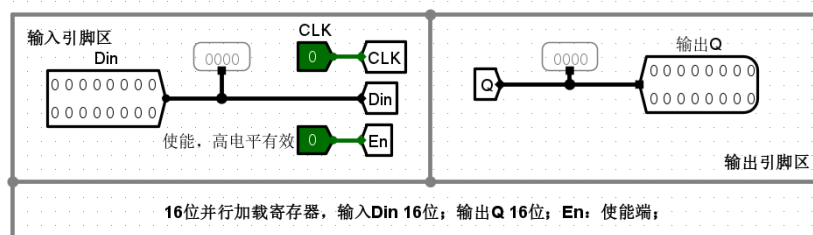


图 2.4-1 并行加载寄存器电路图（一）



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

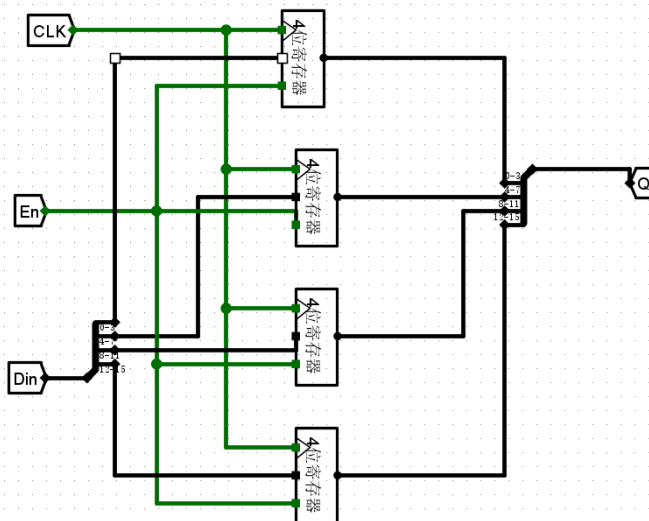
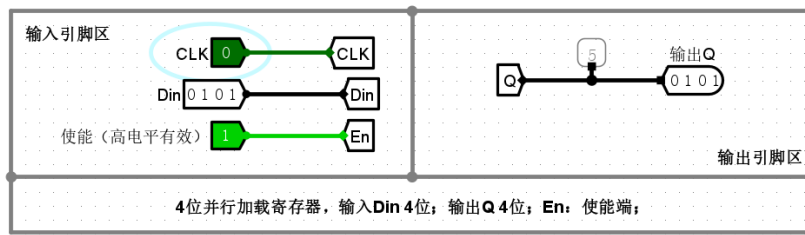


图 2.4-2 并行加载寄存器电路图（二）

(3) 测试图



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

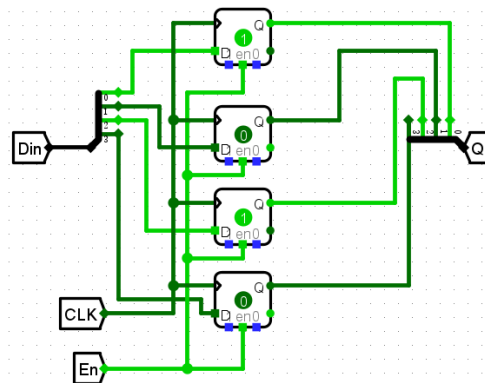
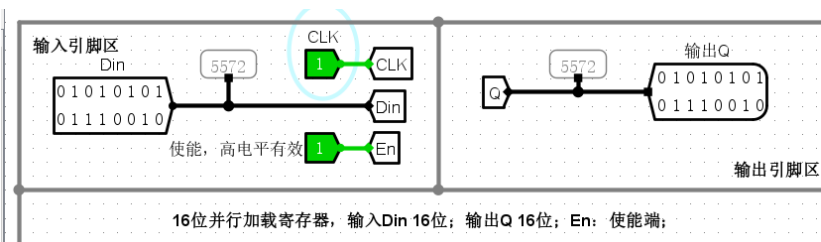


图 2.4-3 并行加载寄存器测试图（一）



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

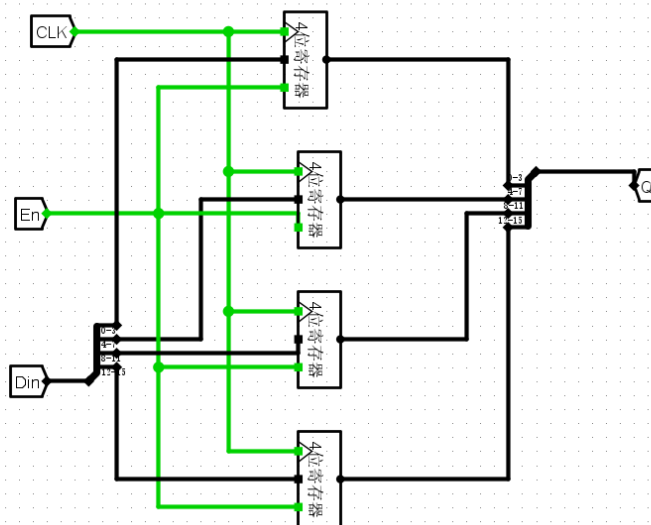


图 2.4-4 并行加载寄存器测试图（二）

(4) 测试分析

测试中给出了两个样例四位的 0101 和十六位的 5572，均得到了想要的结果，测试样例比较随机，经典，效果良好，且通过了平台测试，证明成功设计了并行加载寄存器。成功通过第四关。

2.5 码表计数器

(1) 设计思路及设计过程

码表计数分为四位，分别从高到低，计数自然是从最低位开始计，满了就往前进一位，所以倒数第二位计数的条件是最低位进位，所以倒数第二位的使能端要连接 En 和最低位进位标志的与，正数第二位计数的条件是后两位进位，所以他的使能端要连接 En 和后两位的进位标志的与，同理最高位。

下面介绍 4 位 BCD 计数器设计思路：

4 位 BCD 计数器设计需要 BCD 计数器状态转换和输出函数，先将 En，CLK，Rst 连接到对应位置，综合 $Cout=(Q==9)?1:0$ ；考虑，将寄存器的 Q 与分线器连接，先经过输出函数输出到 Cout，再来一个分支经过状态转换，再连接到 D，同时注意由于是同一个 Q，寄存器出来的 Q 要和器件 Q 相连即可。

下面介绍 BCD 计数器状态转换设计思路：

次态输出 N， $N=S+1$ 。同步时序电路状态转换 Excel 中，现态输入 0~9，次态输入 1~9，0 生成的表达式输入到 logisim 中，生成电路即可。

下面介绍 BCD 计数器输出函数设计思路：

进位输出，S=9 时，Cout=1.直接将下图的数据输入真值表生成电路即可。

1. Cnt	S	Cout
2. 00	0	0
3. 01	1	0
4. 02	2	0
5. 03	3	0
6. 04	4	0
7. 05	5	0
8. 06	6	0
9. 07	7	0
10. 08	8	0
11. 09	9	1
12. 0a	a	0

图 2.5-1 要输入的数据

(2) 电路图

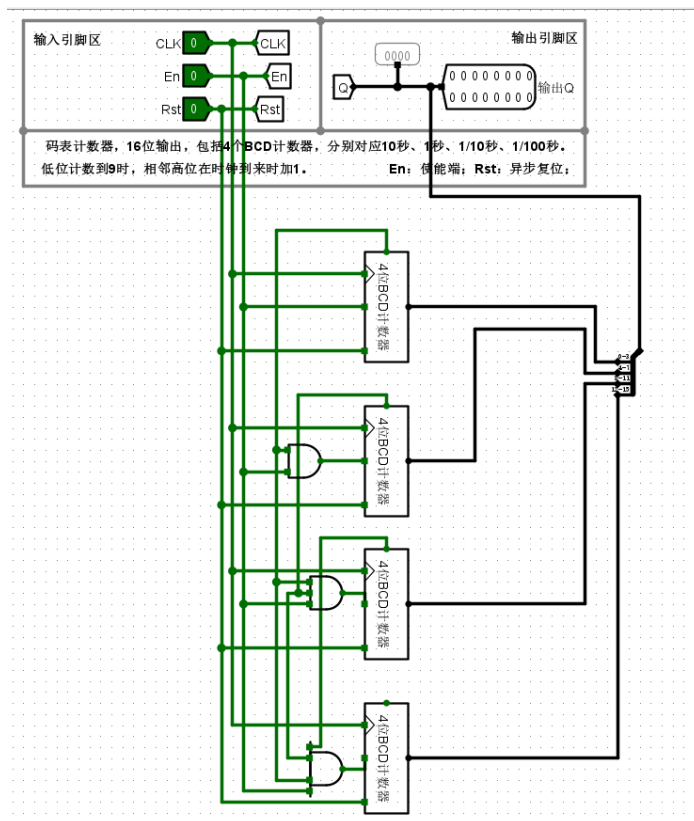


图 2.5-2 码表计数器电路图

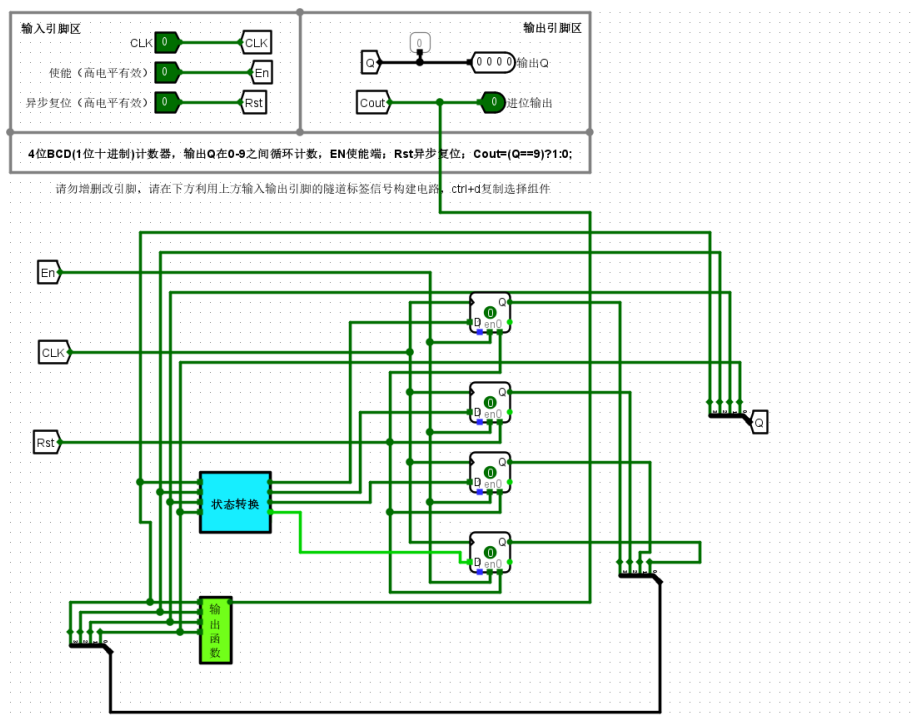


图 2.5-3 4 位 BCD 计数器电路图

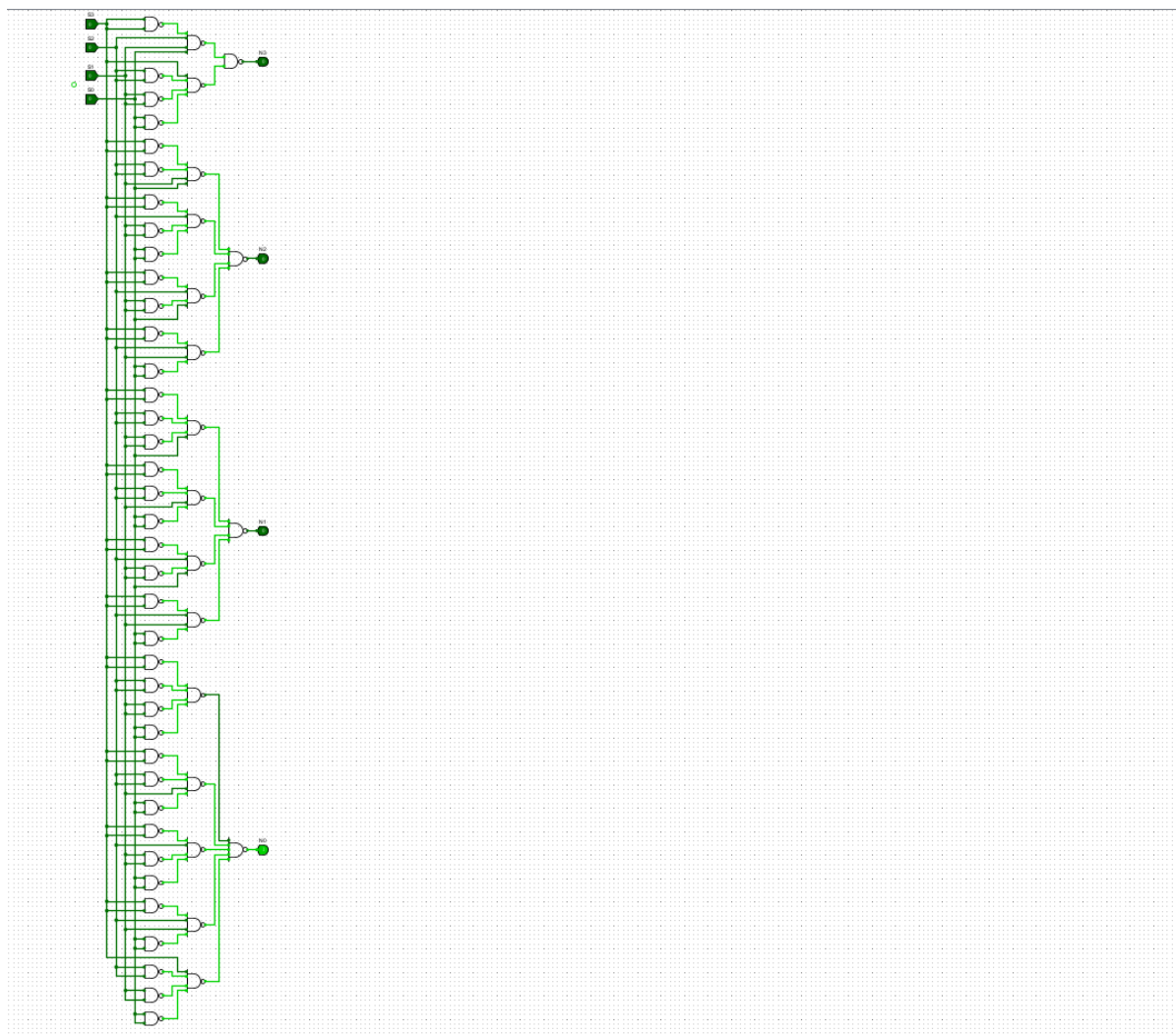


图 2.5-4 BCD 计数器状态转换电路图

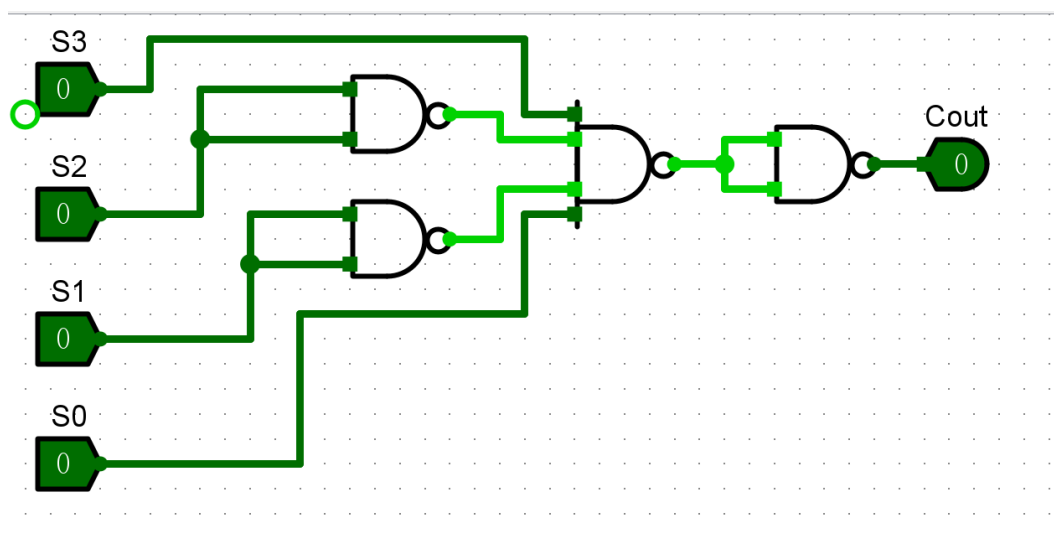


图 2.5-5 BCD 计数器输出函数电路图

(3) 测试图

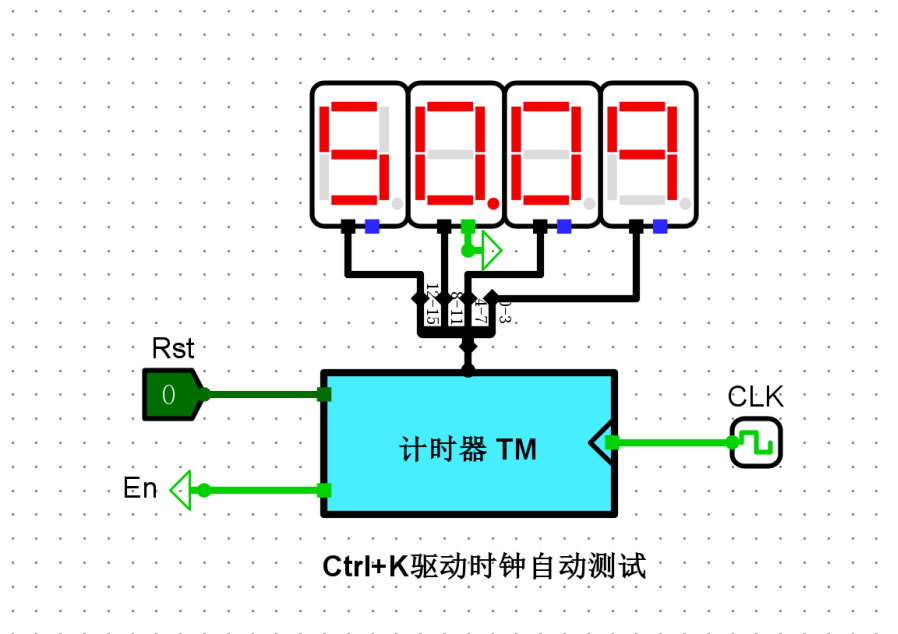


图 2.5-6 码表计数器测试图

(4) 测试分析

如图测试到 00.00 到 50.09，没有出现任何 bug，测试效果良好，且通过了平台的测试，说明已经成功设计出良好的码表计数器，成功通过第五关。

2.6 码表显示驱动设计

(1) 设计思路及设计过程

结合第一关设计的七段数码管驱动，将 Din 输入用 16 位分线器连接，再接入数码管驱动，再输出到分线器连接到 DispInfo 即可，唯一需要注意的地方是小数点，因为只有中间的小数点需要显示，所以用常量 1 连接中间的小数点，剩下的小数点位置连接常量 0 即可。

(2) 电路图

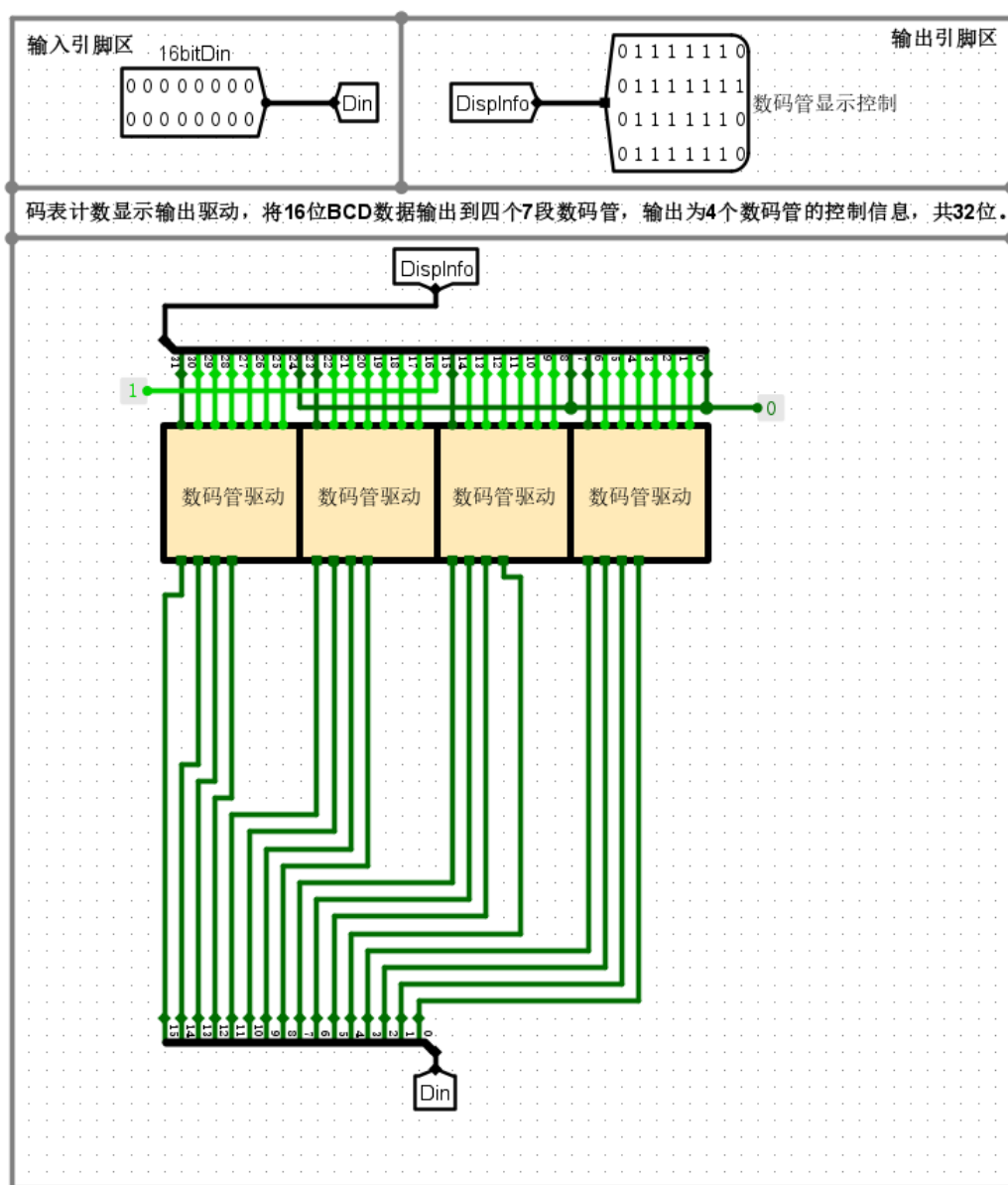


图 2.6-1 码表显示驱动电路图

(3) 测试图

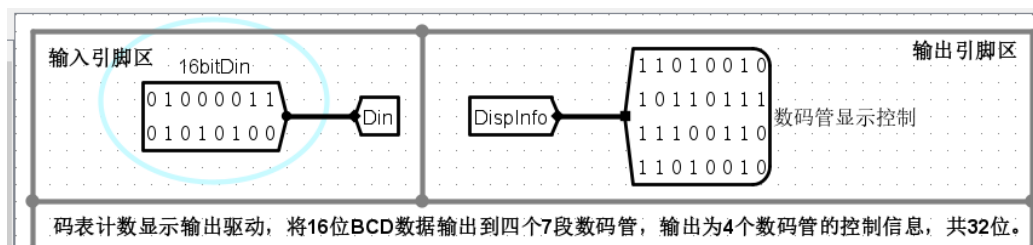


图 2.6-2 码表显示驱动测试图

(4) 测试分析

进行了如图测试的样例，输入 4354，应该显示 43.54，1101001 对应 4，末尾 0

对应没有小数点，1011011 对应 3，末尾 1 代表有小数点，1110011 对应 5，末尾 0 代表没有小数点，1101001 对应 4，末尾 0 代表没有小数点。手动测试成功，并成功通过平台测试，说明成功设计了码表显示驱动。

2.7 码表控制器

(1) 设计思路及设计过程

码表控制器需要实现码表控制器状态转换和输出函数，将 start, stop, store, reset, new record 与状态转换的对应端口连接起来，再连到寄存器的 D 端，Q 输出连接输出函数，再接一条分支接回状态转换，再把 CLK 连接到时钟端即可。

下面介绍状态转换设计思路：

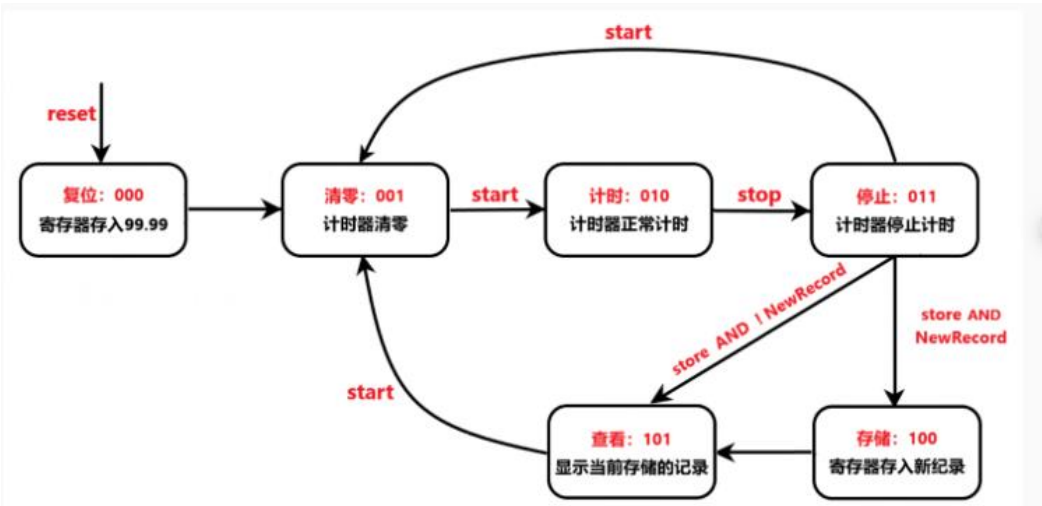


图 2.7-1 状态转换参考图

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	start	stop	store	reset	NewRecord						次态 10进制	N3	N2	N1	N0
0	0	1	0	2		1		0							3	0	0	1	1
0	0	1	1	3	1	0	0	0							1	0	0	0	1
0	0	1	1	3		0	1	0	0						5	0	1	0	1
0	0	1	1	3		0	1	0	1						4	0	1	0	0
0	1	0	0	4				0							5	0	1	0	1
0	1	0	1	5	1	0	0	0							1	0	0	0	1
0	1	0	1	5	0			0							5	0	1	0	1
0	0	1	0	2		0		0							2	0	0	1	0
0	0	1	1	3	0		0	0							3	0	0	1	1
0	0	0	1	1	0			0							1	0	0	0	1

图 2.7-2 状态转换条件设计

根据状态转换参考图，补出一些隐藏的状态转换，利用 Excel 算出相应表

达式，生成电路即可。

下面介绍输出函数设计思路：

1. Cnt	PS	SDsel	SDen	DPsel	TMen	TMreset
2. 00	0	0	1	1	0	1
3. 01	1	0	0	1	0	1
4. 02	2	0	0	1	1	0
5. 03	3	0	0	1	0	0
6. 04	4	1	1	1	0	0
7. 05	5	0	0	0	0	0

图 2.7-3 输出函数相应数值

当前状态(现态)					输入信号												
S3	S2	S1	S0	现态 10进制	start	stop	store	reset	NewRecord	0	0	0	Sdsel	Sden	Dpsel	Tmen	Treset
0	0	0	0	0									0	1	1	0	1
0	0	0	1	1									0	0	1	0	1
0	0	1	0	2									0	0	1	1	0
0	0	1	1	3									0	0	1	0	0
0	1	0	0	4									1	1	1	0	0
0	1	0	1	5									0	0	0	0	0

图 2.7-4 输出函数真值表

根据输出函数的相应数值，输入到 Excel 中，得到表达式，在 logisim 生成电路即可。

(2) 电路图

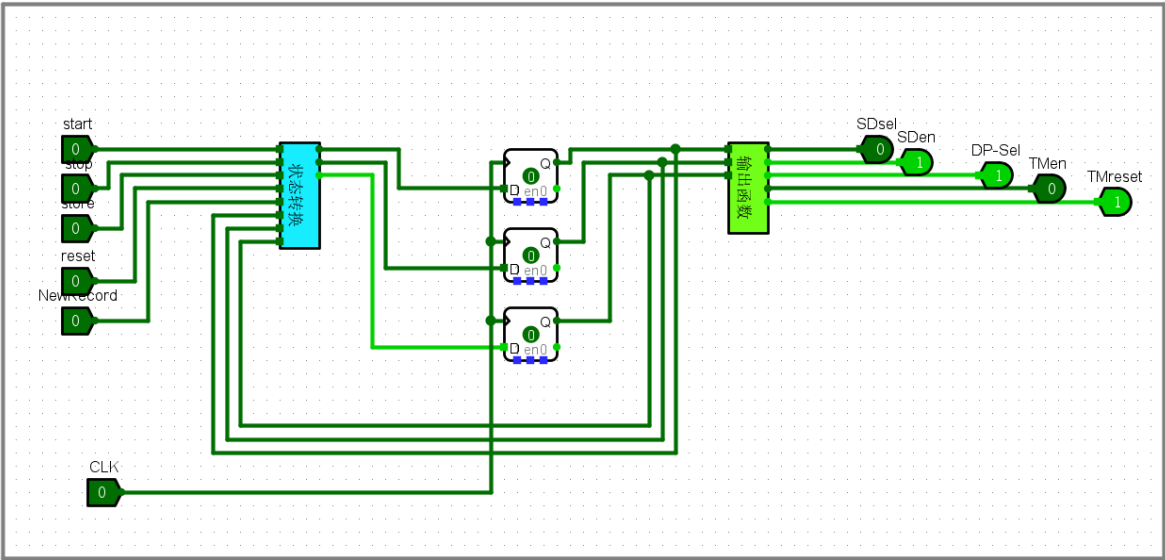


图 2.7-5 码表控制器电路图

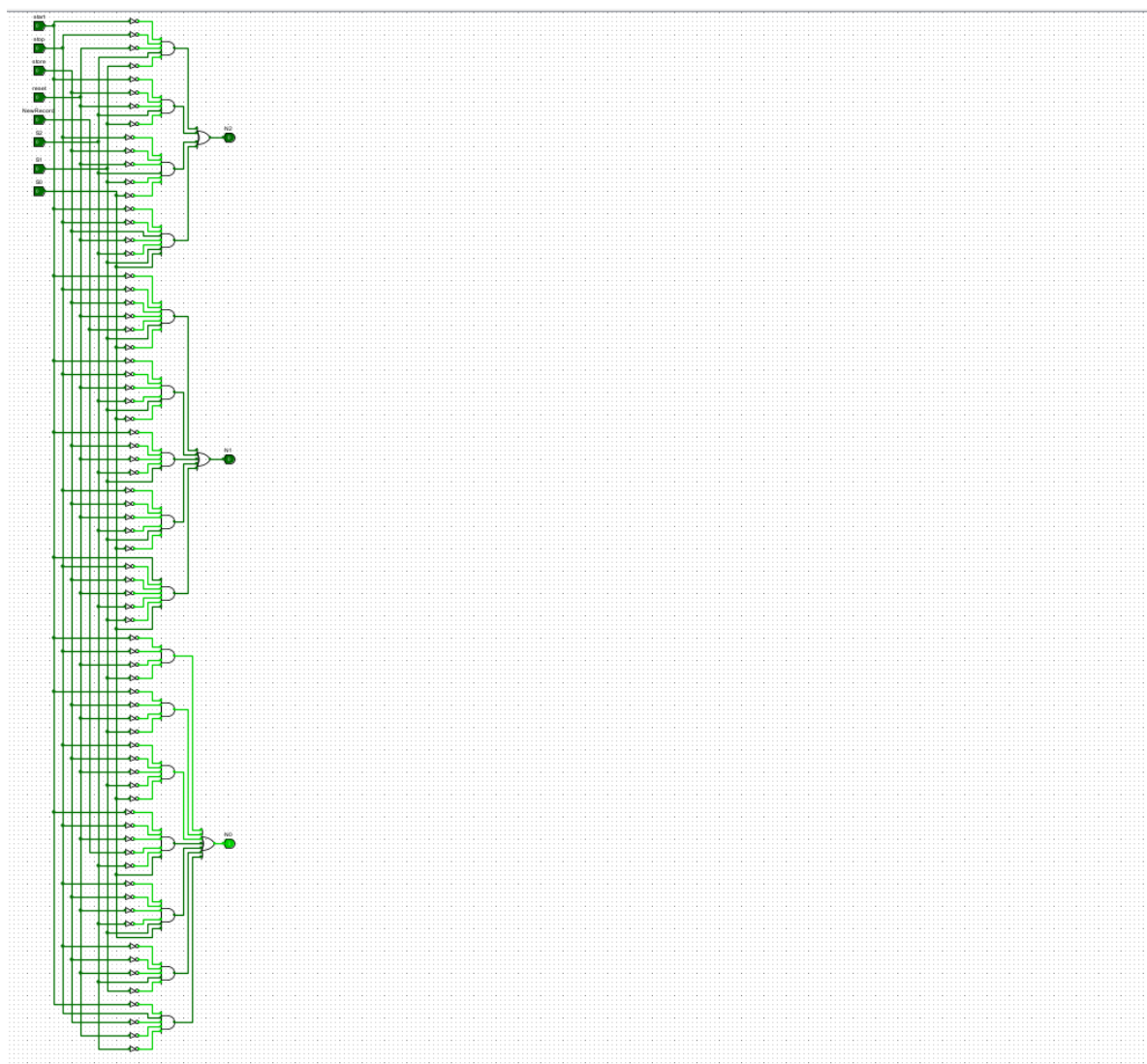


图 2.7-6 码表控制器状态转换电路图

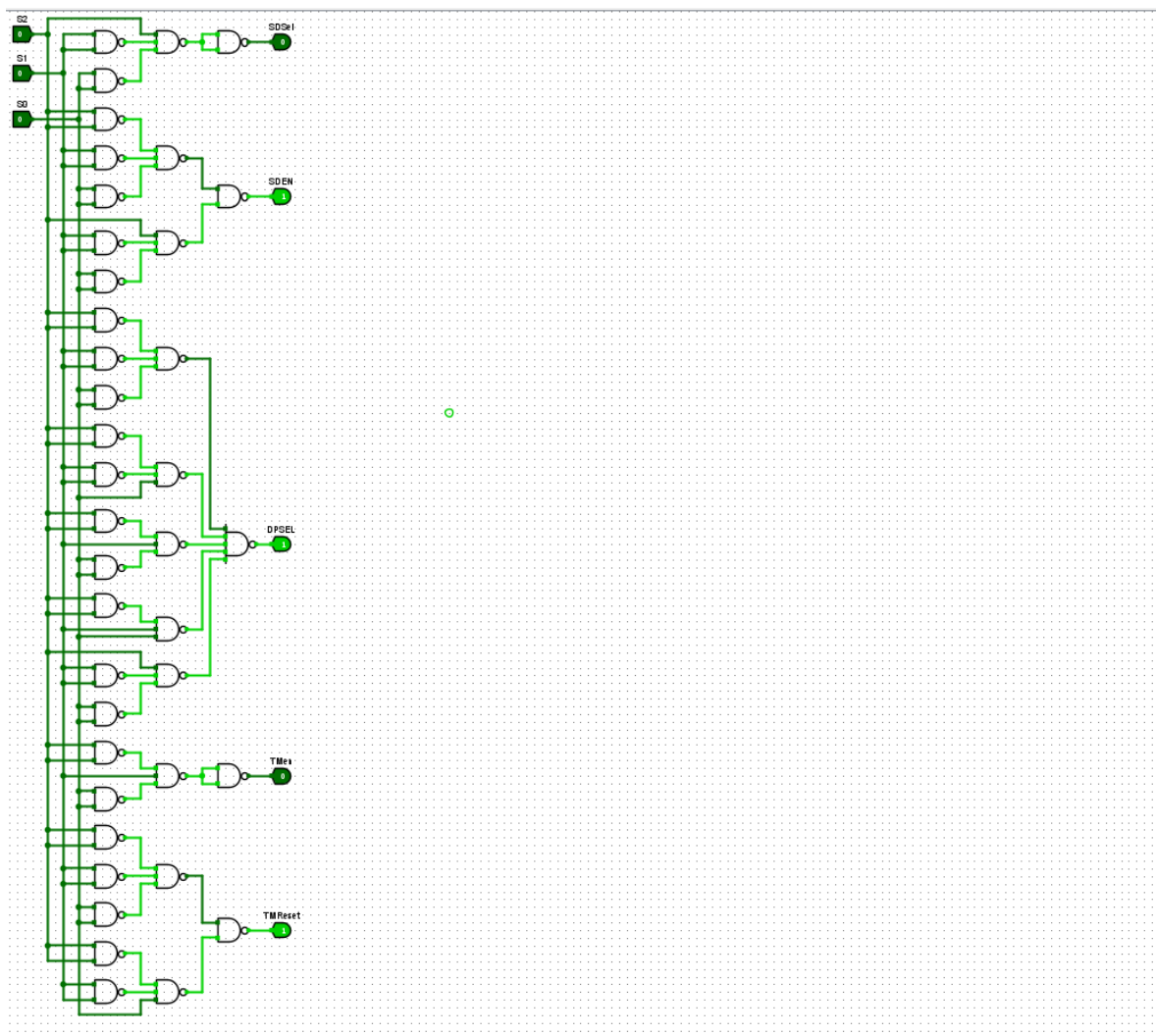


图 2.7-7 码表控制器计算函数电路图

(3) 测试图

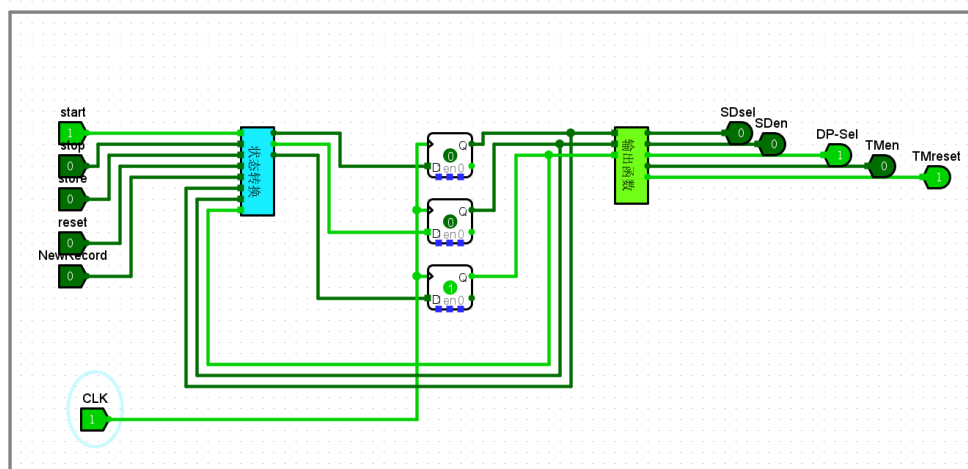


图 2.7-8 码表控制器计算函数测试图（一）

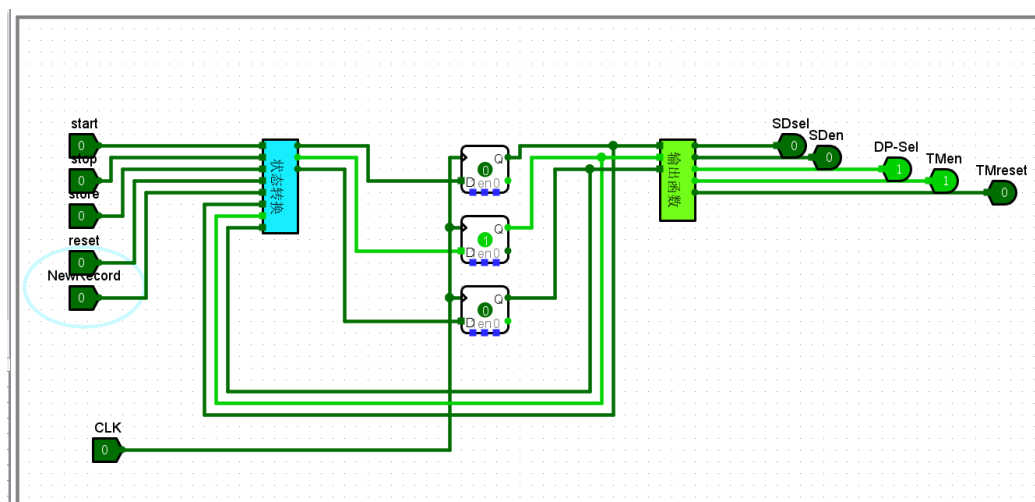


图 2.7-9 码表控制器计算函数测试图（二）

(4) 测试分析

如图测试，均满足平台测试的样例结果。并成功通过平台所有测试结果，成功设计出码表控制器。

2.8 运动码表系统设计（系统集成）

(1) 设计思路及设计过程

将 DP-SEL，SD-EN，SD-SEL，TM-Reset，TM-EN，CLK 先连接对应的端口，计时结果一个分支到无符号比较器，如果比最好历史记录好，则覆盖最好历史记录，再一条分支经过二路选择器，进入寄存器，再进入二路选择器，最后进行码表显示。

(2) 电路图

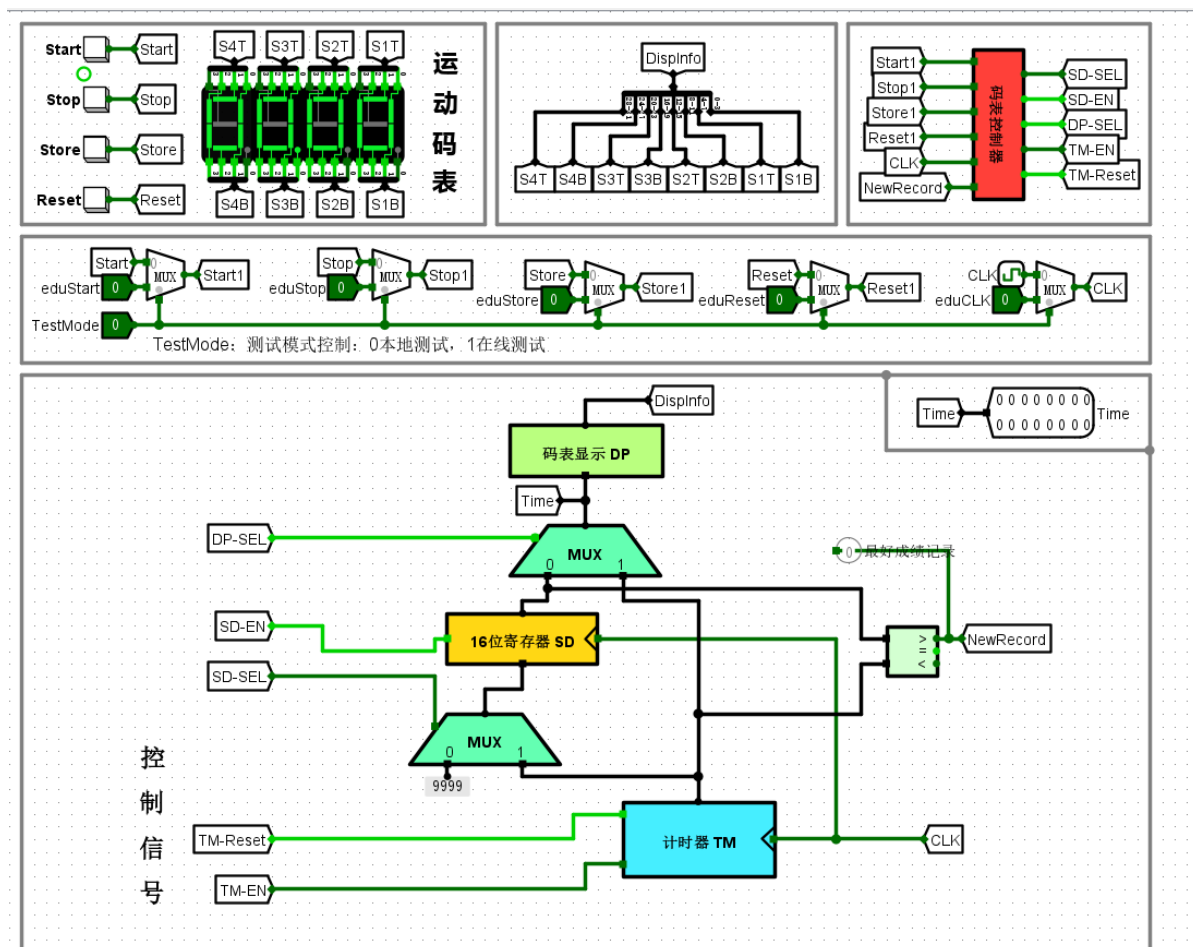


图 2.8-1 运动码表系统设计电路图

(3) 测试图

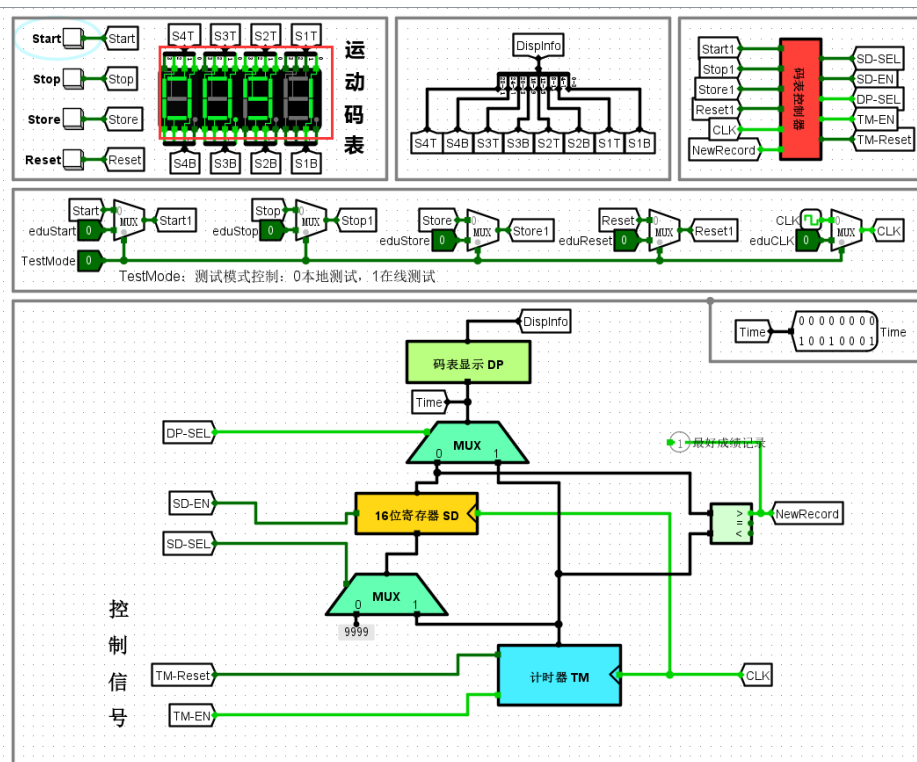


图 2.8-2 运动码表系统设计测试图（一）

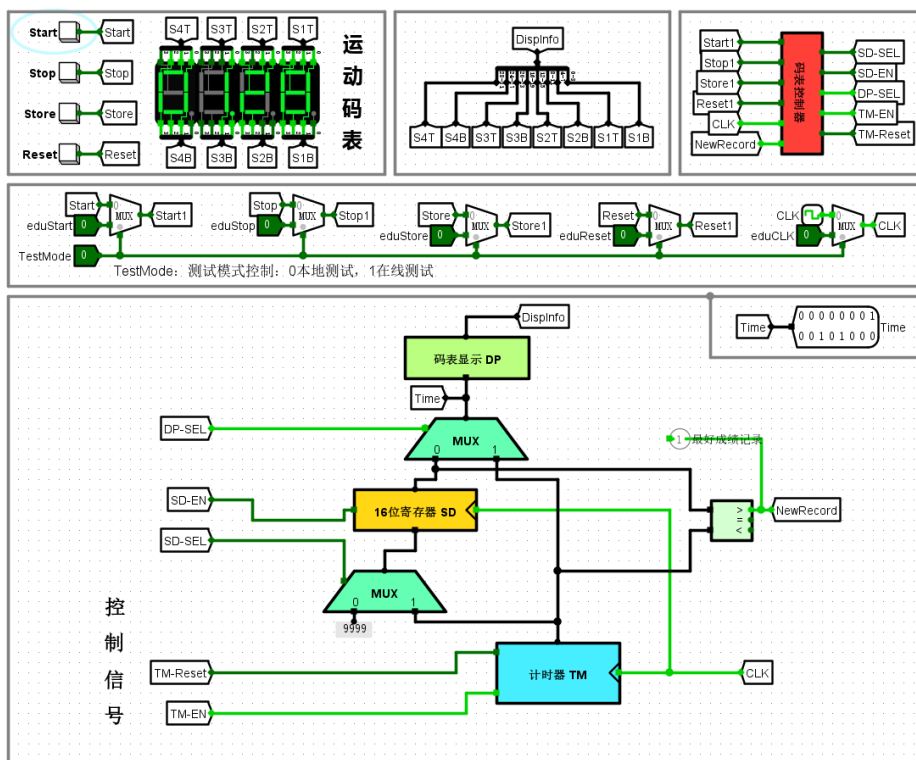


图 2.8-3 运动码表系统设计测试图（二）

(4) 测试分析

按住 Start 不放，Ctrl+K 开始自动测试，发现开始计时，并且没有任何错误，如图计时到 00.91 和 01.28，并且通过平台的全部测试样例，成功实现运动码表系统设计。