***2022***



**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： | 计卓2101 |
| 学 号： | U202112071 |
| 姓 名： | 王彬 |
| 电 话： | 15005276201 |
| 邮 件： | wangbin2002@hust.edu.cn |
| 完成日期： | 2022.12.19 |

**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 |
| 设计方案与实验过程 | 60 |  |  | |
| 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](#_Toc117868487)

[1.1 实验名称 1](#_Toc117868488)

[1.2 实验目的 1](#_Toc117868489)

[1.3 实验环境 1](#_Toc117868490)

[1.4 实验内容 1](#_Toc117868491)

[1.5 实验要求 2](#_Toc117868492)

[2 设计方案与实验过程 3](#_Toc117868493)

[2.1 方案设计 3](#_Toc117868494)

[2.2 实验过程 12](#_Toc117868495)

[3 设计总结与心得 41](#_Toc117868496)

[3.1 实验总结 41](#_Toc117868497)

[3.1.1遇到的问题及处理 41](#_Toc117868498)

[3.1.2设计方案存在的不足 41](#_Toc117868499)

[3.2 实验心得 41](#_Toc117868500)

[3.3 意见与建议 41](#_Toc117868501)

# 实验概述

## 实验名称

运动码表系统设计。

## 实验目的

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。

## 实验环境

软件：Logisim2.15.0.2软件一套。

平台：https://www.educoder.net

## 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，分别显示秒和百分秒。

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。

## 实验要求

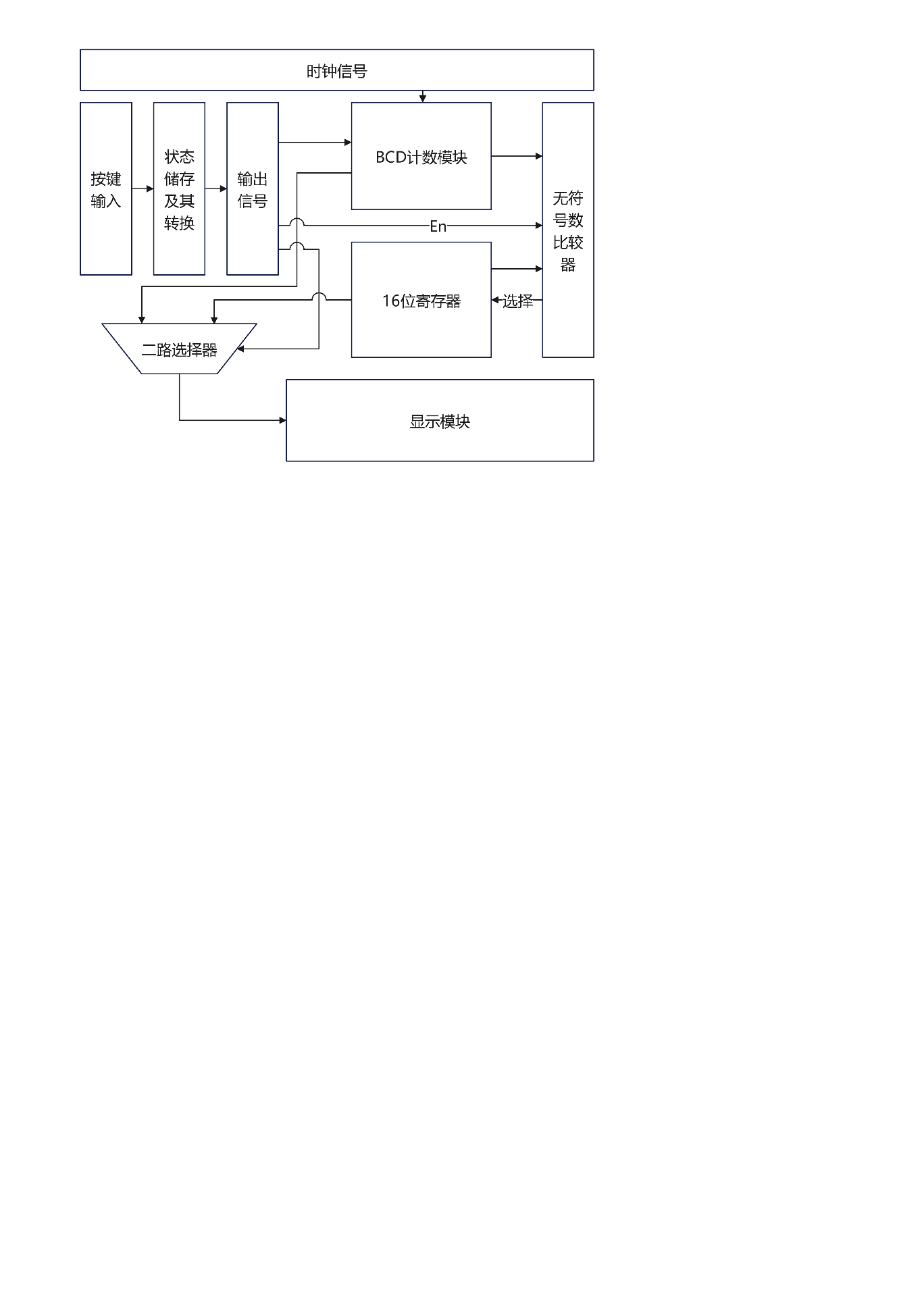
1. 根据给定的实验包，将运动码表系统切分为一个个实验单元；
2. 对每一个实验单元，按要求设计电路并使用Logisim软件进行虚拟仿真；
3. 设计好的电路在educoder平台上提交并进行评测，直到通过全部关卡。

# 设计方案与实验过程

## 方案设计

我们对电路的总体结构进行设计。该运动码表包括LED计数电路、数码管驱动电路、16位2路选择器、16位无符号整数比较器、16位并行加载寄存器、BCD计数器、码表计数器、码表显示驱动和码表控制器。我们的运动码表是由这一系列组件所构成的。

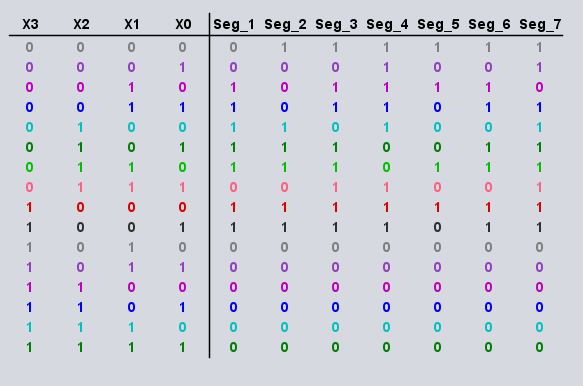
通过按键输入和码表所处的状态得到运动码表的次状态，于是对该状态输出为信号以统摄全局。输出信号操控计数模块是否开始计时，如果我们按下“停止”键即可停止计数，且将本次的成绩同寄存器比较，如果较小则填充入寄存器。显示为时间或寄存器的存储成绩。因此，我们对运动码表的大致信号传递图设计如下。



**2.1.1 7段数码管驱动电路设计**

我们对7段数码管驱动电路进行设计，通过填写真值表的方式使得其电路得以自动生成。通过数码管我们可以得到七个输出与四个输入对应的关系，填写好对应的真值，就可以得到表达式。

其真值表如下所示。



根据该真值表使用电路的自动生成功能实现对电路的自动生成。

**2.1.2 2路选择器（1位）电路设计**

1位二路选择器具有三个输入段X0, X1, Sel和一个输出段Out。当当Sel端为0时，输出X0；当Sel端为1时，输出X1。因此其电路的逻辑功能为Out = (Sel == 0)?X0:X1; 获得了其基本方法后，我们采用真值表的方式实现对其电路的生成。

其逻辑表达式为：



基于该逻辑函数实现对其电路的自动生成。

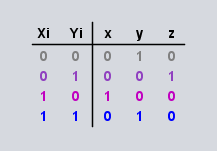
**2.1.3 2路选择器（16位）电路设计**

16位二路选择器可以视作为将16个1位二路选择器并行地连接在一起，将Sel端视为总线，并依次对每一位进行二路选择。因而，当Sel端为0时，输出X；当Sel端为1时，输出Y。

**2.1.4 1位无符号比较器电路设计**

1位无符号比较器具有两个一位输入X和Y，具有三个逻辑输出Greater，Equal和Less，分别以x，y，z作为别称。其功能为判断X和Y之间的大小关系。

根据其逻辑关系我们写出它的真值表：



通过该真值表建立电路即可。1位无符号比较器是构建4位和16位无符号比较器的基础。

**2.1.5 4位无符号比较器电路设计**

我们利用1位无符号比较器，构建4位无符号比较器。该比较器具有两个四位输入X和Y。我们的判断思路为，依次比较每一位X和Y的大小，若最高位X比Y大，则X大于Y；反之，则Y比X大。若第一位X和Y相等，则再比较下一位。如此反复比较四次，可以得到X和Y的大小关系。如果这四位X和Y均相等，则可以认为X等于Y。

我们根据这种思路，可以列出Greater的逻辑表达式为：

Greater = X3>Y3

+ X3=Y3 & X2>Y2

+ X3=Y3 & X2=Y2 & X1>Y1

+ X3=Y3 & X2=Y2 & X1=Y1 & X0>Y0。

判断相等和小于的输出和如上逻辑表达式类似。

我们根据这种逻辑表达式，对该器件的线路进行手动连接，即可得到其电路。

**2.1.6 16位无符号比较器电路设计**

16位无符号比较器和4位的无符号比较器同理。只需将4位无符号比较器中的1位无符号比较器更换为4位的即可。这样就可以实现16位数的逻辑比较。

**2.1.7 4位并行加载寄存器电路设计**

4位并行加载寄存器具有三个输入Din、使能端En和时钟沿CLK。它同时具有一个输出Q作为对Din的寄存。其功能为：当En为1时，输入数据；当En为0时，保持上次输入的数据。

为了实现该4位并行加载寄存器，我们需要四个D触发器，因为D触发器的触发是由时钟信号决定的，所以我们可以由逻辑关系可知：只需将D触发器的各类接口对Din的每一位依次并行地连接即可。

**2.1.8 16位并行加载寄存器电路设计**

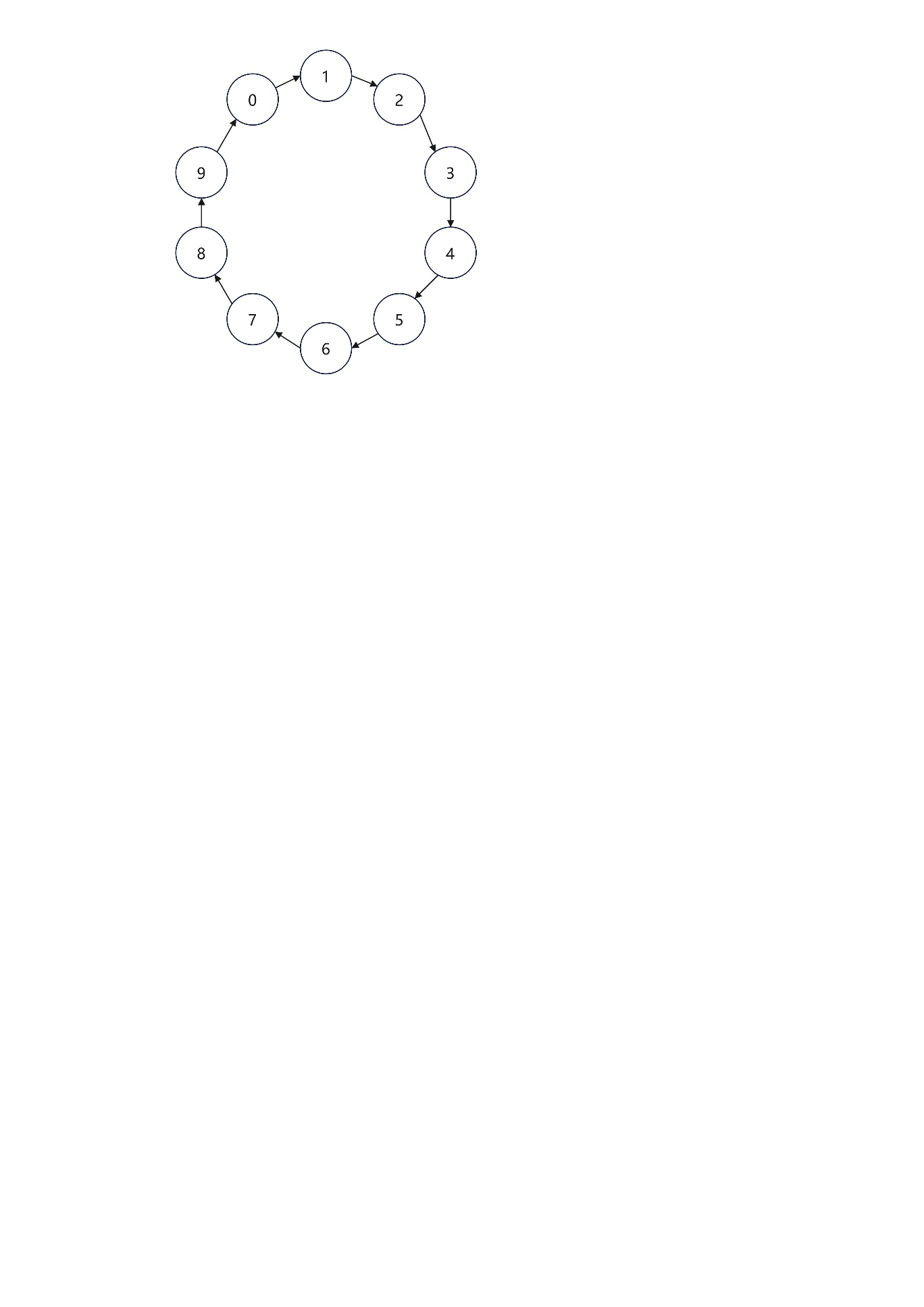
16位并行加载寄存器和4位的同理。只需将4位并行加载寄存器中的D触发器更换为4位并行加载寄存器的即可。这样就可以实现16位数的并行寄存。同时，需要使用分线器将每位数据进行分段处理。

**2.1.9 4位BCD计数器电路设计**

4位BCD计数器可以存储1个十位数，并进行依次的累加。该计数器主要组成部分有状态转换，触发器模块和输出函数。触发器模块用以存储该计数器的状态（共10个状态，0..9），输出函数为该计数器的进位。我们计数器使用了Moore电路。Moore电路的输出仅与现态有关，而Mealey的输出与现态，输入都有关。因此可以有效降低噪声，提高其稳定性。

触发器模块由四个 D触发器构成，以存储该计数器所处的状态，并作为输出Q以输出。每次受到时钟信号，都会通过状态转换电路更改到下一个状态。而四个D触发器的状态也会由输出函数电路的运作以得到输出的解。该输出函数为Cout = (Q==9)?1:0;

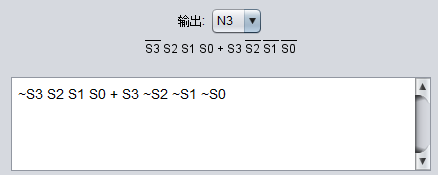
它的十个状态之间的转换关系为：

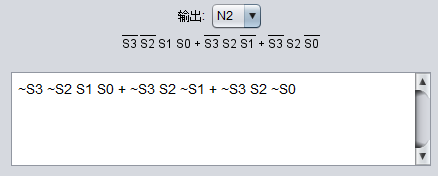


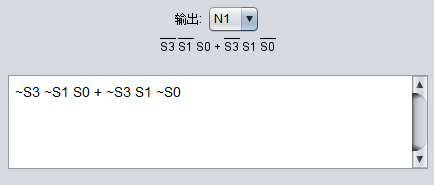
由同步时序电路状态转换表可以得到表达式。其转换表为：

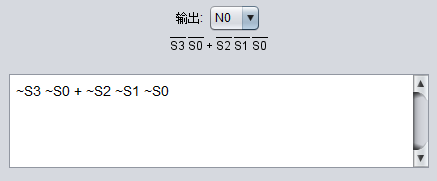


因此得到其逻辑表达式为：









根据这些逻辑函数，实现电路的自动生成即可。

对于输出函数，即BCD计数器的进位输出，直接判断Q是否为9即可，因此可以直接写出其逻辑表达式：

Cout = S3 ~S2 ~S1 S0

通过该表达式直接对输出函数的电路进行自动生成。

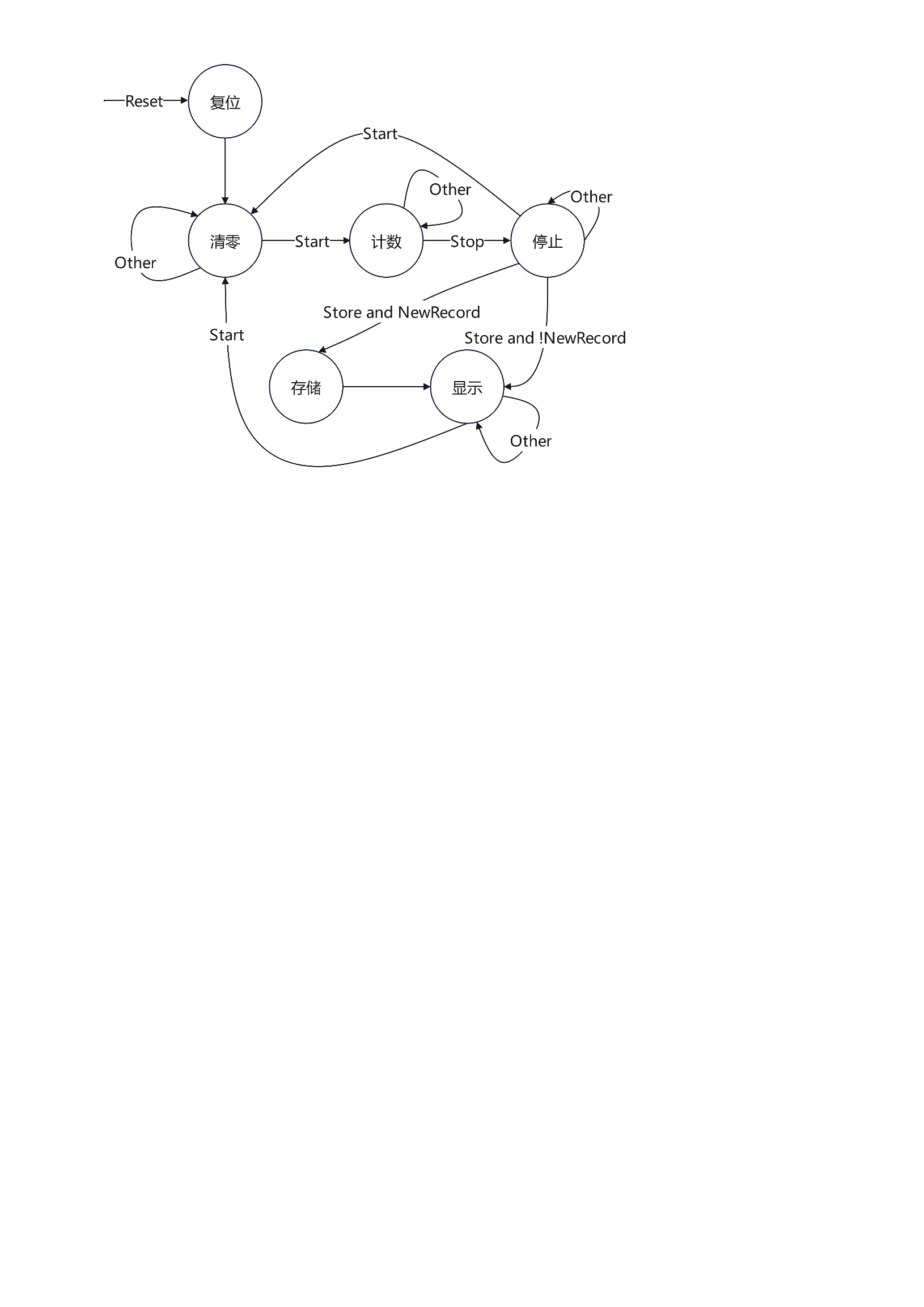
**2.1.10 码表计数器及其显示电路设计**

码表计数器只需将4个四位BCD计数器进行依次连接即可。而码表显示驱动只需将4个数码管驱动电路进行依次的连接即可。

**2.1.11 码表控制器电路设计**

为了实现对整个码表的控制，我们将码表控制器分为三个部分：状态转换、状态存储和输出函数。由于该运动码表有五个按键：start, stop, store, reset, NewRecord，和一个时钟端CLK，用户的输入和触发器当前所处的状态经过状态转换函数后，转换为次态，并通过输出函数电路确定该状态下的输出信号。输出信号有5个，分别是SDsel, SDen, DP-Sel, TMen, TMreset。

我们设计了六个状态，分别是复位、清零、计数、停止、显示和存储。这六个状态的功能如下：复位为清除寄存器中的记录，并将记录设为最大；清零即将计时器清零；计数即开始计时；停止为停止计时；存储为将目前的数字存储至寄存器，如果目前记录较大则不存储；显示为将寄存器中的记录显示至显像管。我们设计了该码表的状态和状态转移关系如下图：



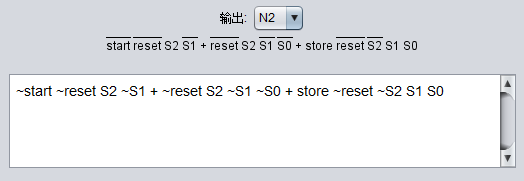
为了使得该设计落实，我们对设计的六个状态进行编码，实现对状态的储存。由于我们共设计了六个状态，只需要三个D触发器即可存储我们的所有状态 ，因此我们采用三位二进制数来编码状态。我们对状态的编码如下表：

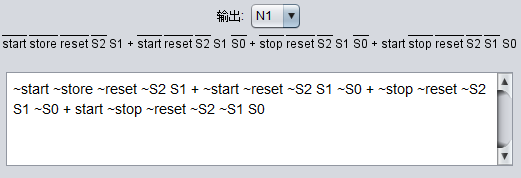
|  |  |
| --- | --- |
| 状态名 | 编码 |
| 复位 | 000 |
| 清零 | 001 |
| 计数 | 010 |
| 停止 | 011 |
| 存储 | 100 |
| 显示 | 101 |

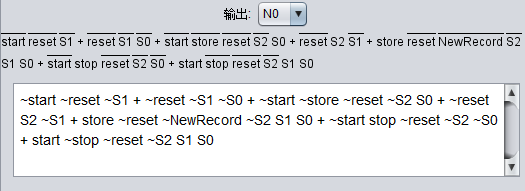
根据上述设计，我们填写了状态转移关系表。该表格详细地表现了各个状态之间的转换关系，其内容如下：



根据该表格所生成的码表控制器状态转换关系逻辑函数如下。





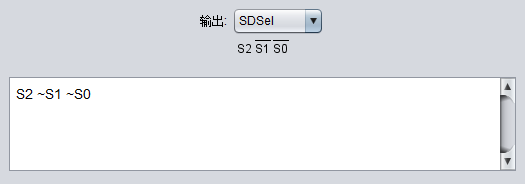


根据这几个逻辑表达式，实现状态转移函数的电路自动生成即可。

对于码表控制器的输出函数，我们写出每个状态对应的输出信号如下表。



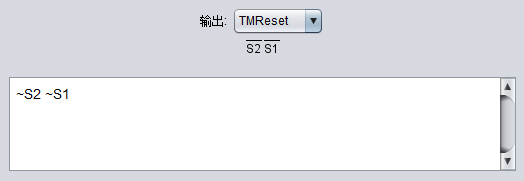
我们根据该输出函数的函数表，生成了现态的输出如下。











根据这些状态至信号函数的映射，我们对码表控制器输出函数电路进行自动生成即可。

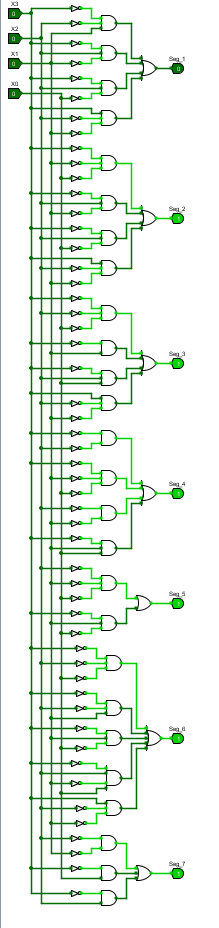
## 实验过程

按照设计方案，按照模块给出电路图以及测试图，对于测试图要求描述清楚模块的功能并对测试用例进行说明，最后进行测试分析。

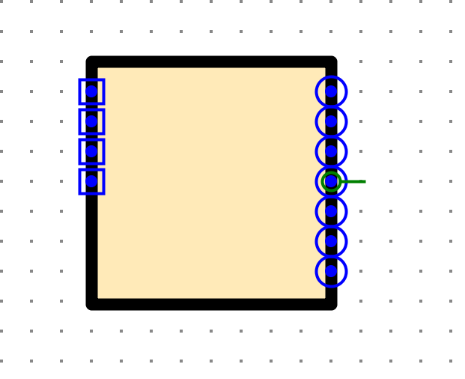
**2.2.1 7段数码管驱动电路**

（1）电路图

该模块的内部结构电路图如下。

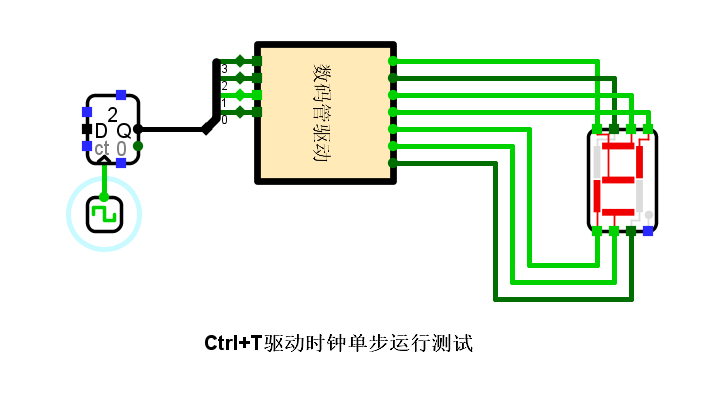


该模块的封装电路图如下。

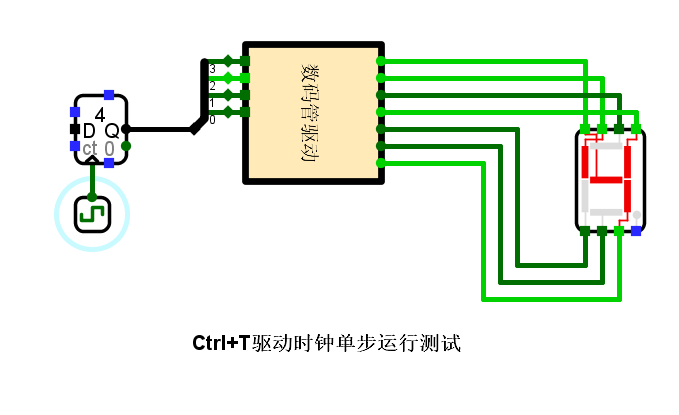


（2）测试图

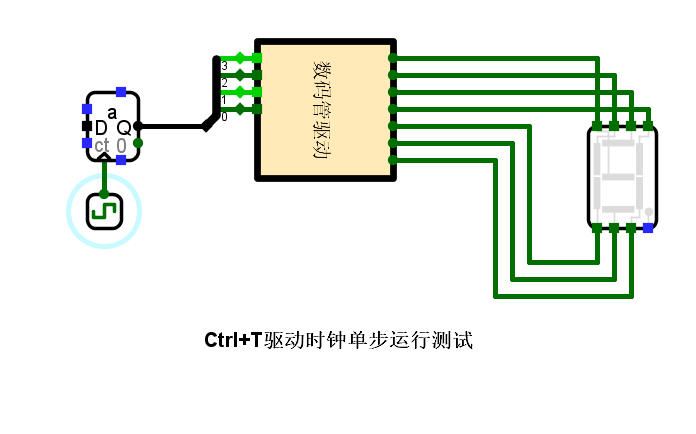
当状态为2且时钟端为上升沿时，其数码管驱动使得输出显像为2.



当状态为4且时钟端为下降沿时，其数码管驱动不改变，使得输出显像为4.



当状态为a且时钟端为下降沿时，其数码管驱动不改变。且由于状态a不合法，不是合规的输入，使得输出不显像。



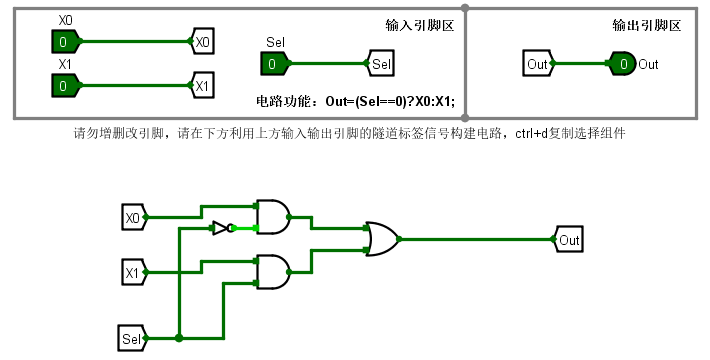
（3）测试分析

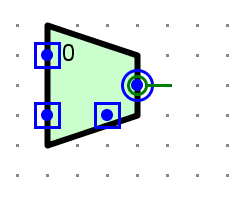
由于输出的恰为7段数码管驱动的信号，使得输出正确。

**2.2.2 2路选择器（16位）电路**

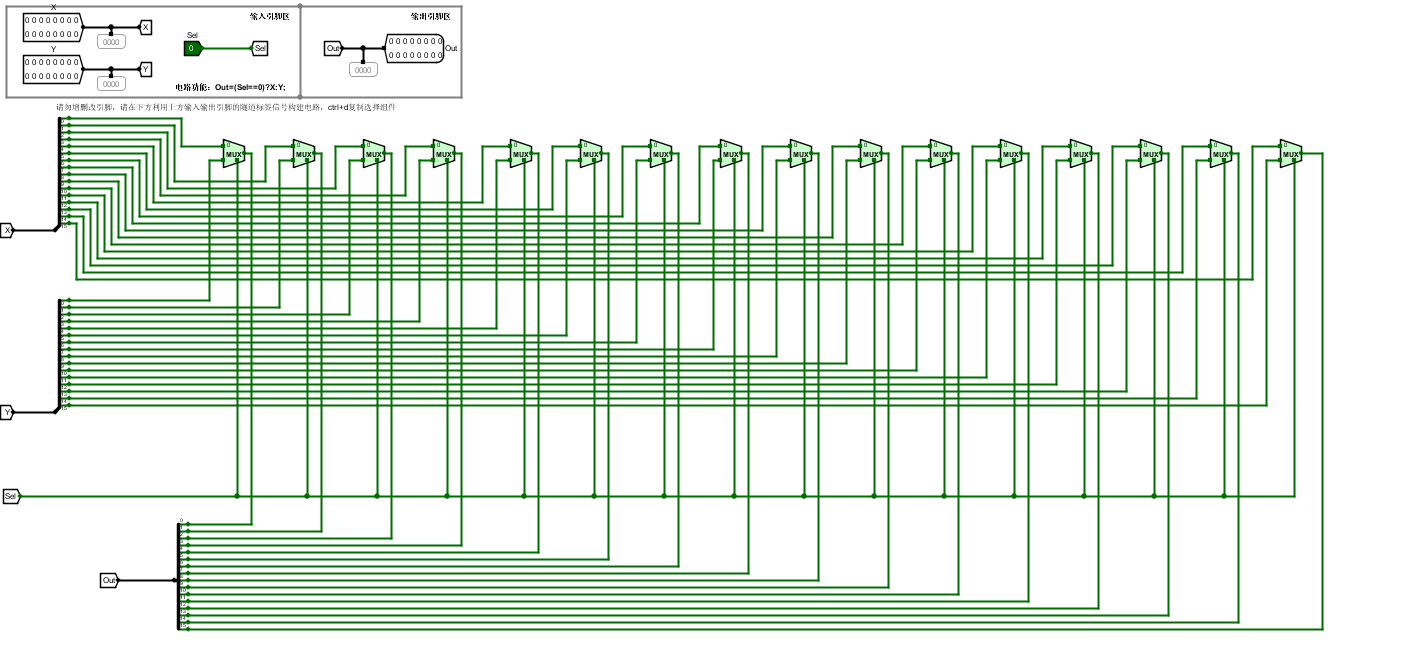
（1）电路图

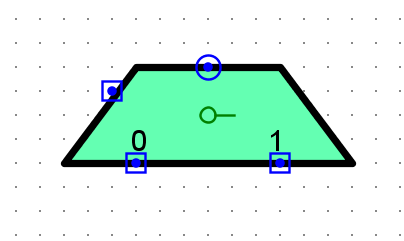
2路选择器（1位）的电路图和封装图如下。





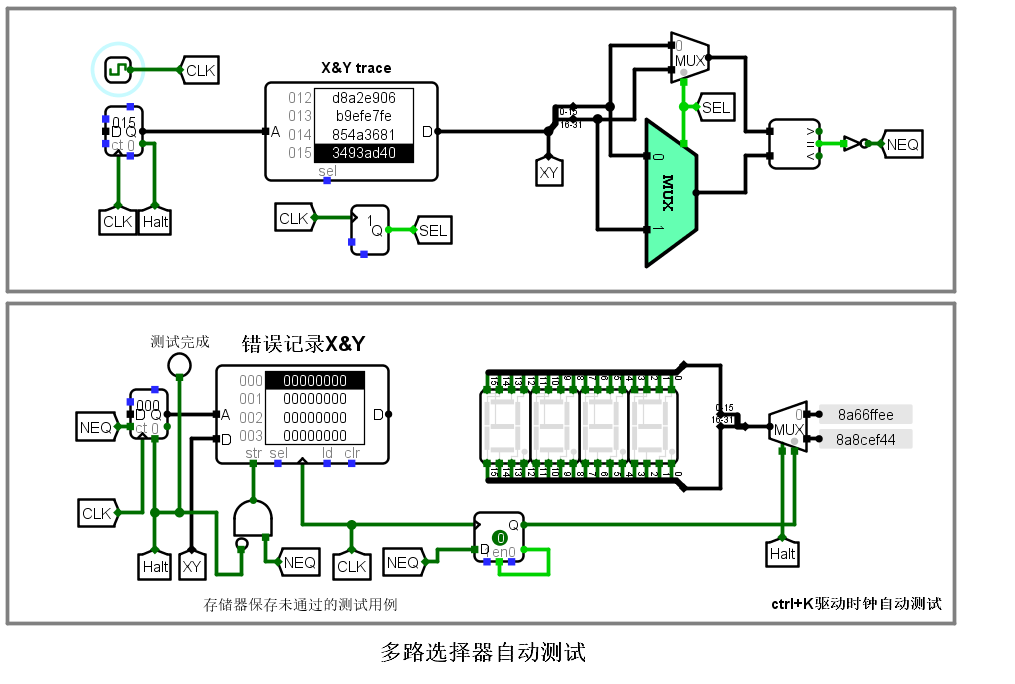
我们将16个1位2路选择器并行连接在一起，以构成16位二路选择器。2路选择器（16位）的电路图和封装图如下。

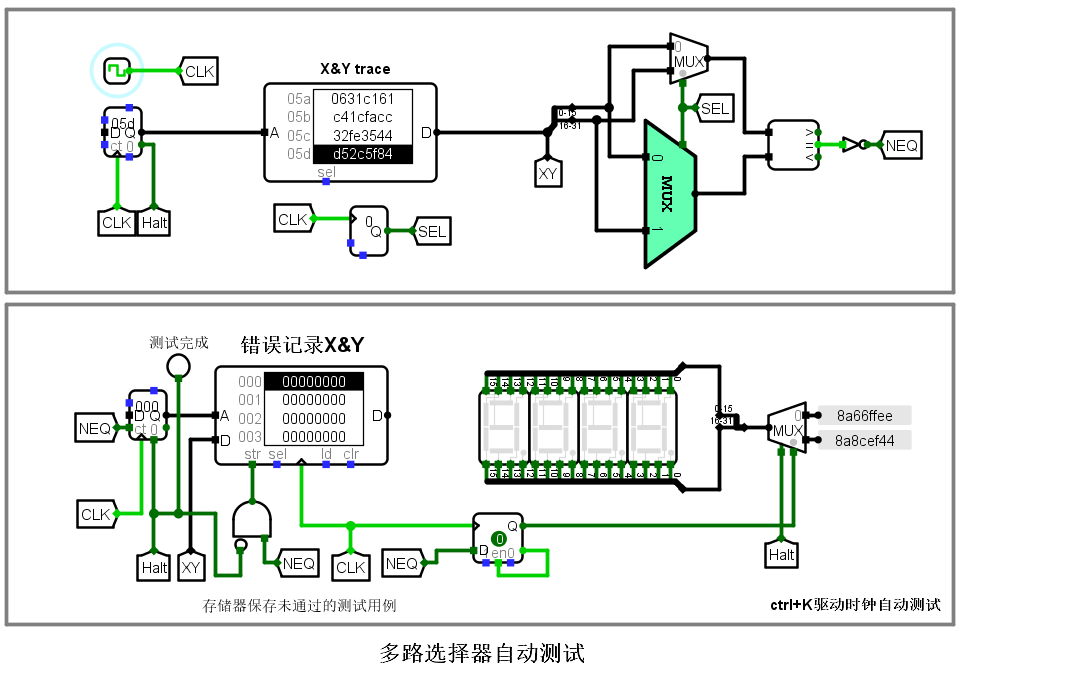




（2）测试图

由于该实验电路较为简易，我们使用自动测试器测试了一段时间，未见异常抛出。





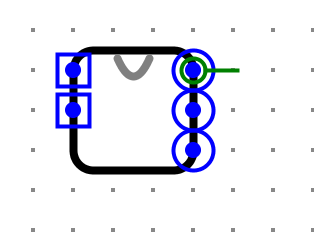
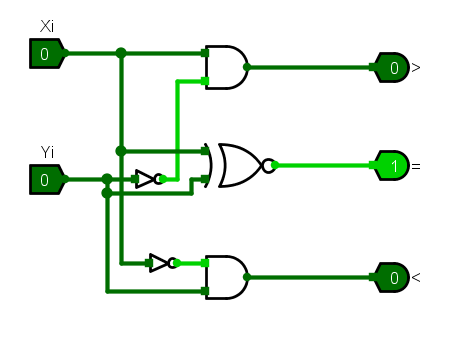
（3）测试分析

如上图，未见异常。

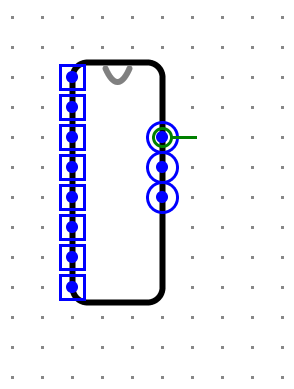
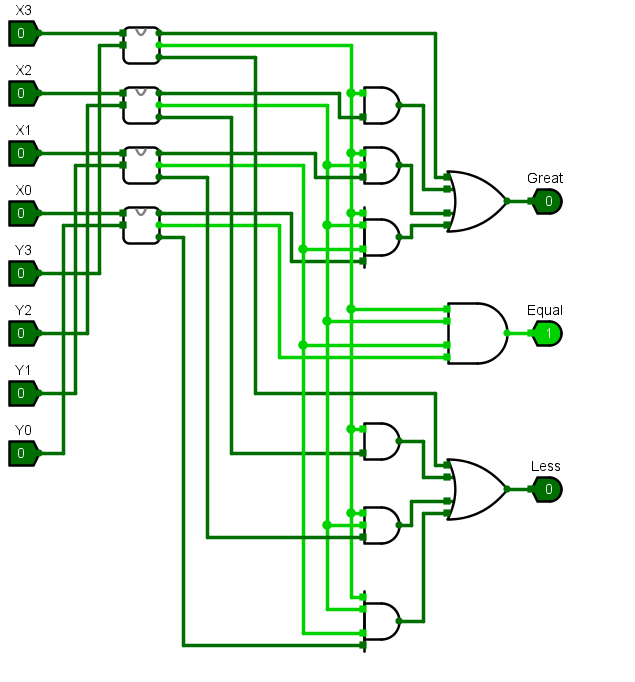
**2.2.3 16位无符号比较器电路**

（1）电路图

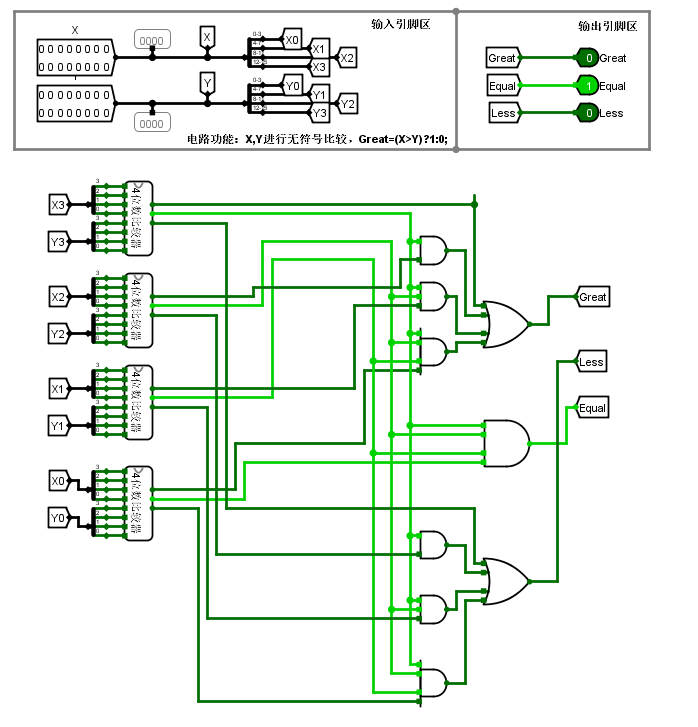
1位无符号比较器的电路图和封装图如下。

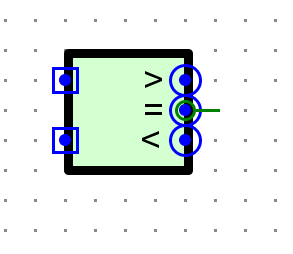


我们将4个1位无符号比较器并行连接在一起，加之以一些逻辑结构，以构成4位无符号比较器器。4位无符号比较器的电路图和封装图如下。



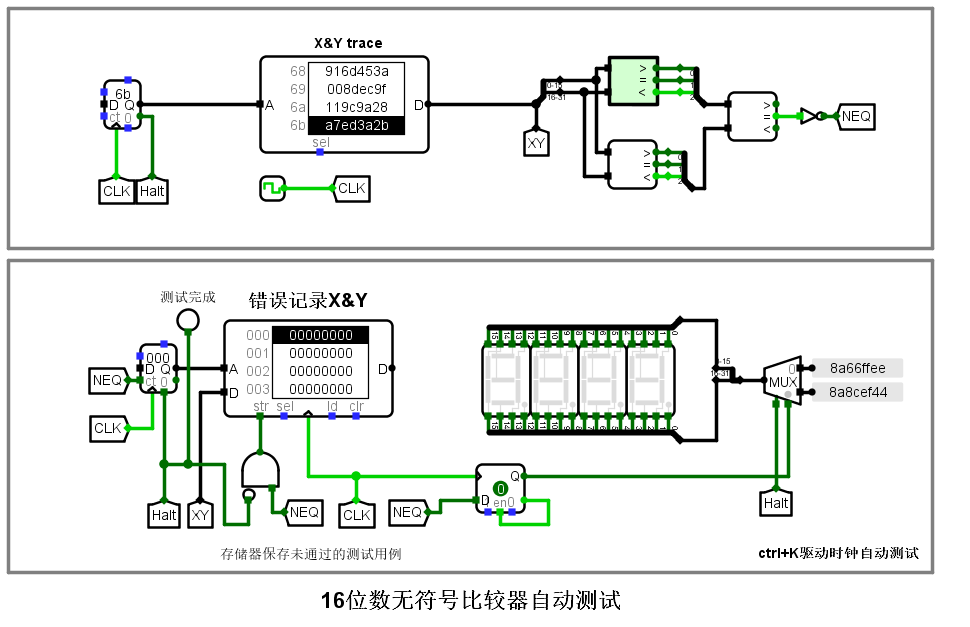
我们将4个4位无符号比较器并行连接在一起，加之以一些逻辑结构，以构成16位无符号比较器器。16位无符号比较器的电路图和封装图如下。

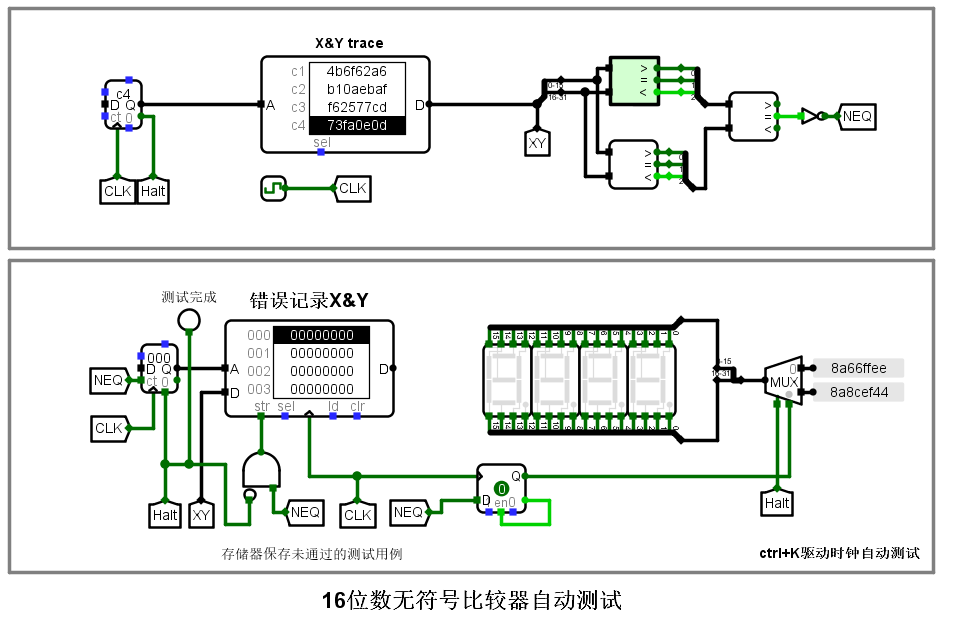




（2）测试图

由于该电路实现较简单，该测试中我们仍然使用16位无符号数比较器自动测试，并运行了一段时间，未见异常。





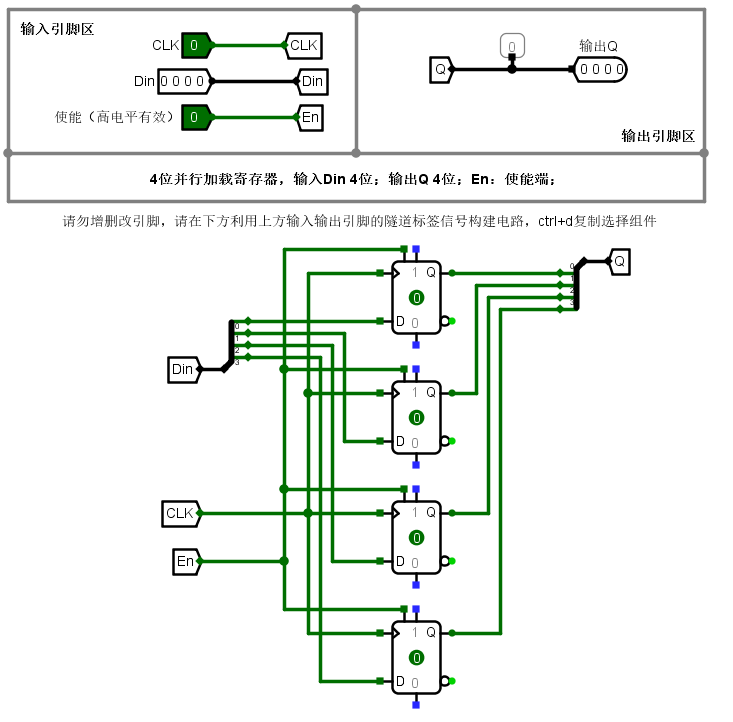
（3）测试分析

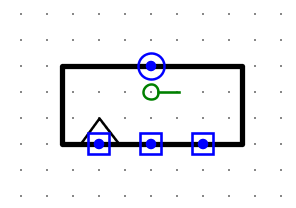
如上图，未见异常抛出，故验证了电路的正确性。

**2.2.4 16位并行加载寄存器电路**

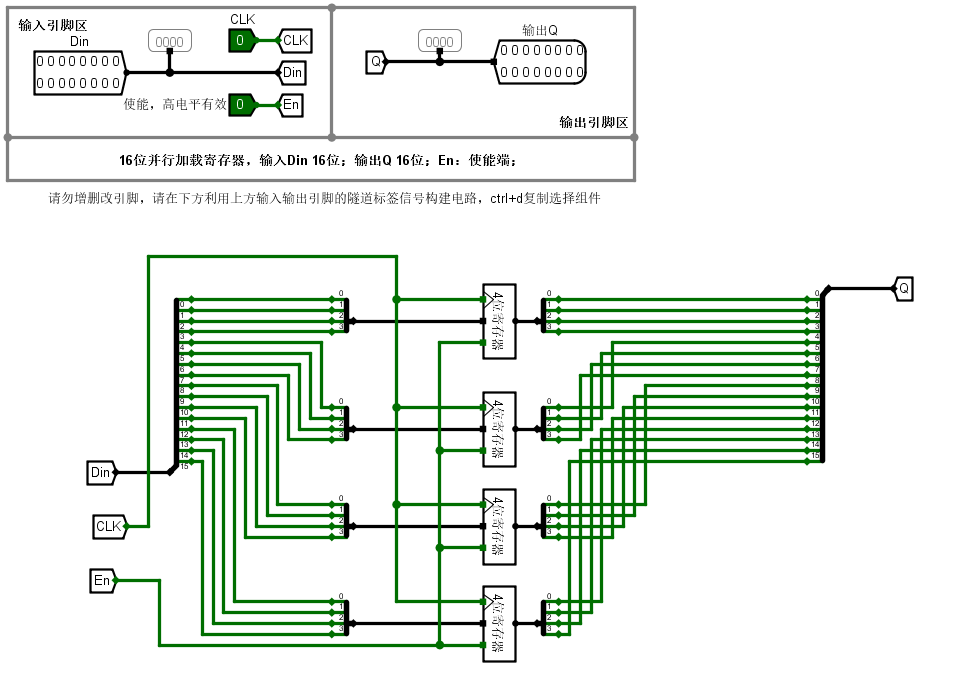
（1）电路图

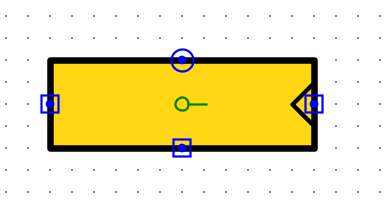
我们首先构建4位并行加载寄存器，它的主体部分是4个D触发器。4位并行加载寄存器的电路图和封装图如下。





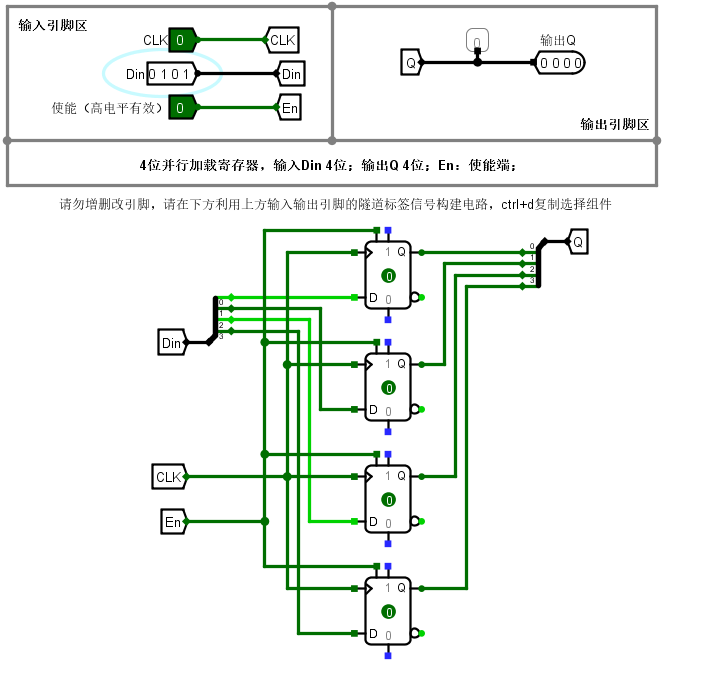
基于如上电路，我们将4个4位并行加载寄存器连接在一起，加之以一些逻辑结构，以构成16位并行加载寄存器。16位并行加载寄存器的电路图和封装图如下。



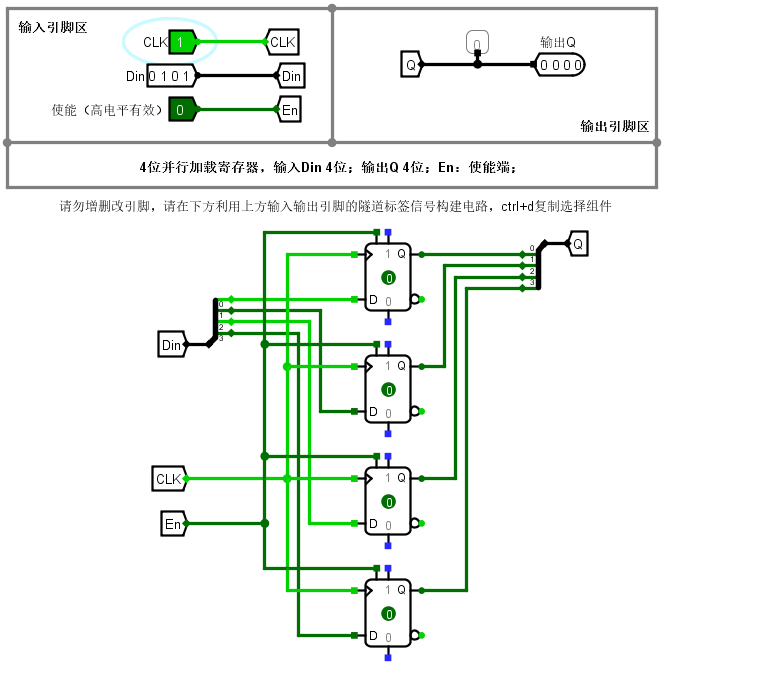


（2）测试图

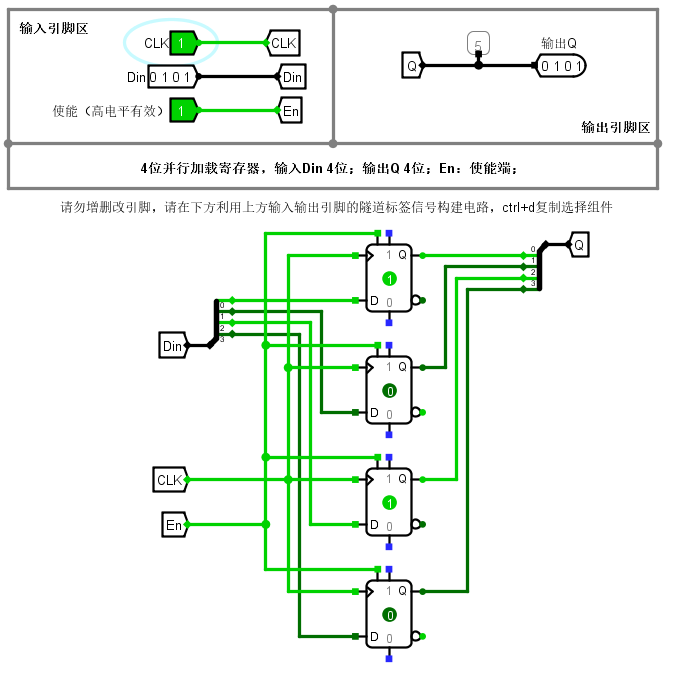
我们首先将Din设置为“0101”，未传来时钟信号CLK时，输出保持为“0000”。



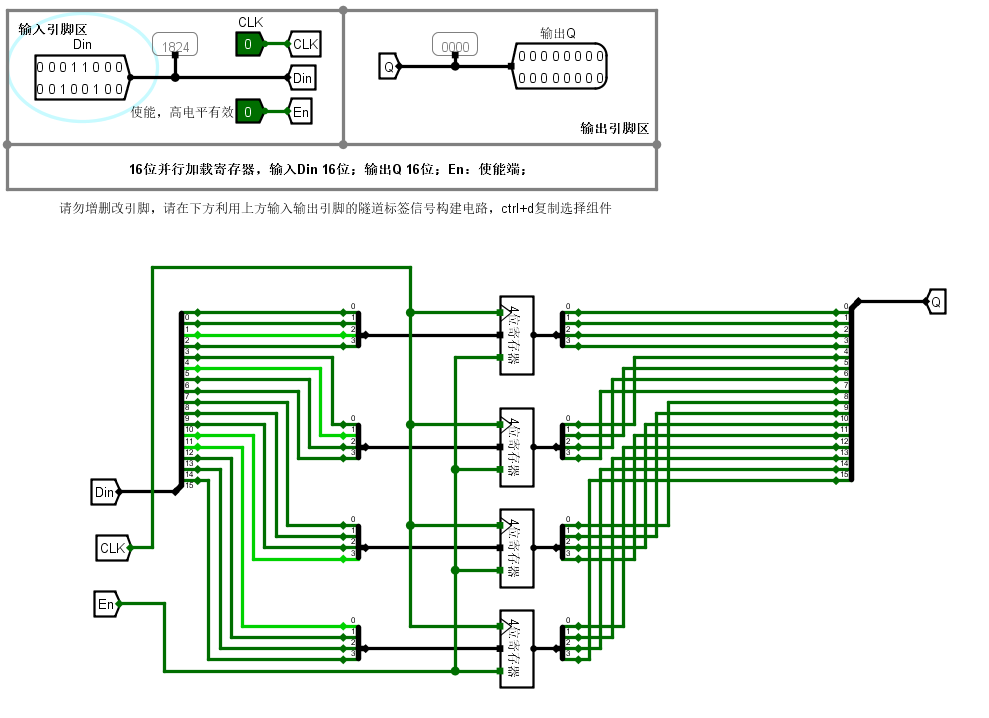
传来了时钟上升沿信号，但使能端En=0，输出仍然未发生变化。

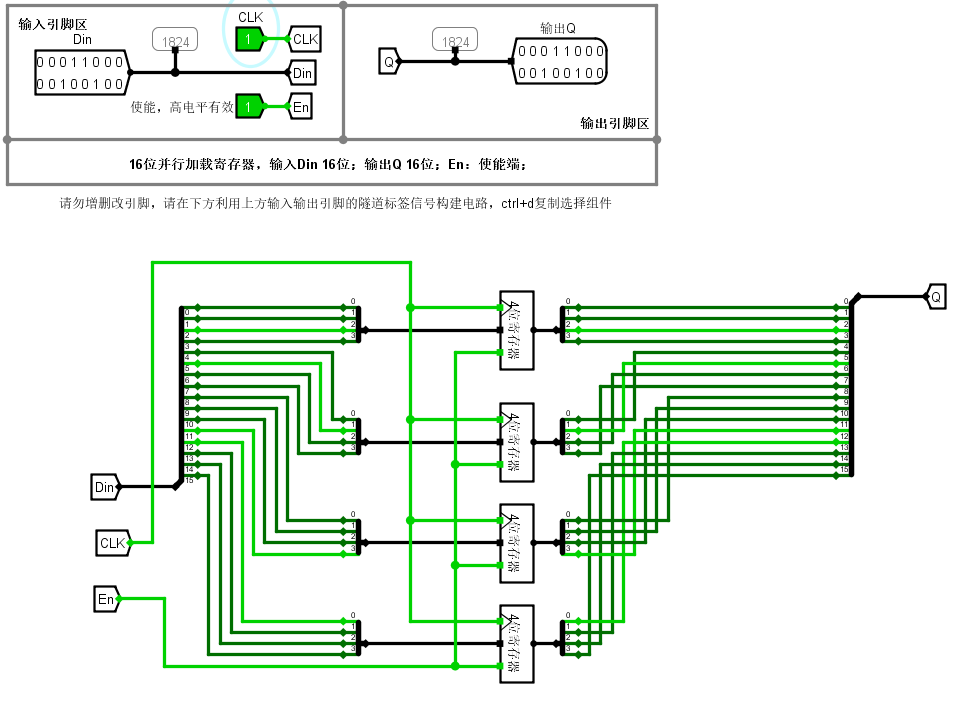


使能端为高电平，且具有时钟信号时，寄存器的值改为“0101”。



以上为对4位并行加载寄存器的测试。下面我们开始对16位并行加载寄存器测试。由于实验包中提供了对应的自动测试，我们运用实验包运行了一段时间，未见异常抛出。





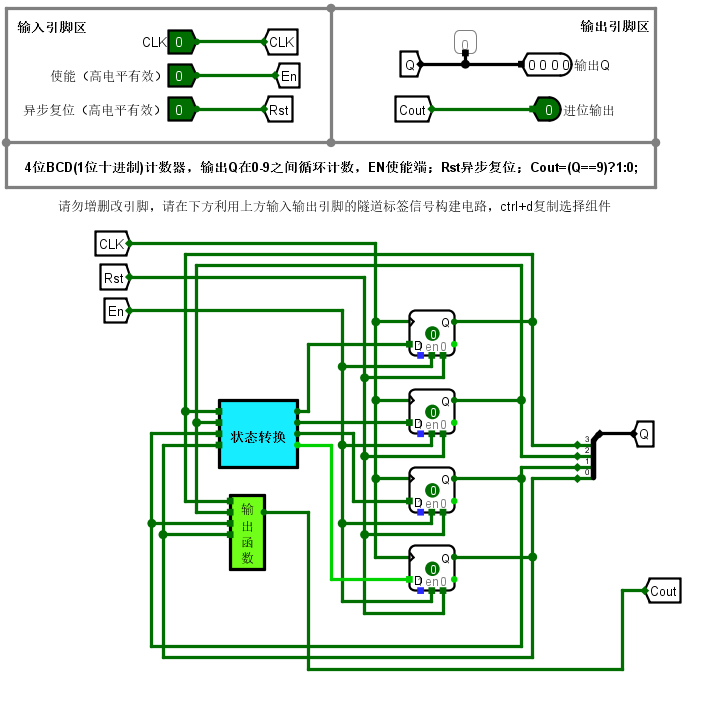
（3）测试分析

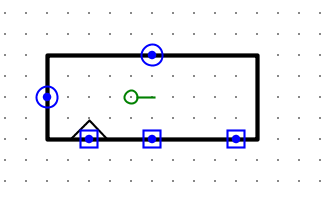
对于4位并行加载寄存器，当使能端为高电平，且时钟端为上升沿时，寄存器存储了相应的值，这表明了电路的正确性。而且自动测试未抛出异常，进一步确认了我们设计的电路是正确的。

**2.2.5 BCD计数器及其码表计数器电路**

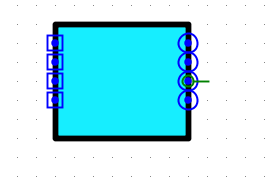
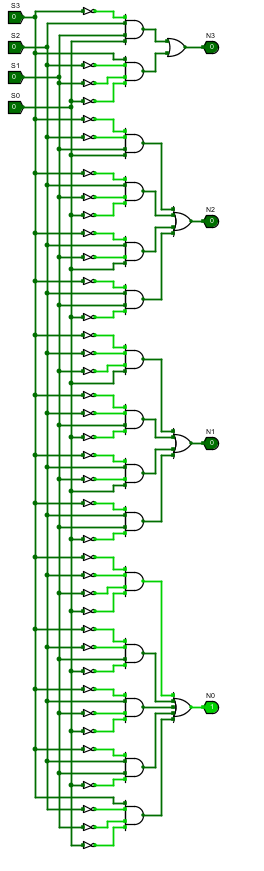
（1）电路图

我们首先构建4位BCD计数器。其电路图和封装图如下。

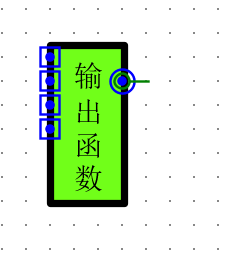
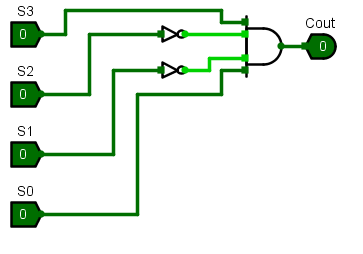




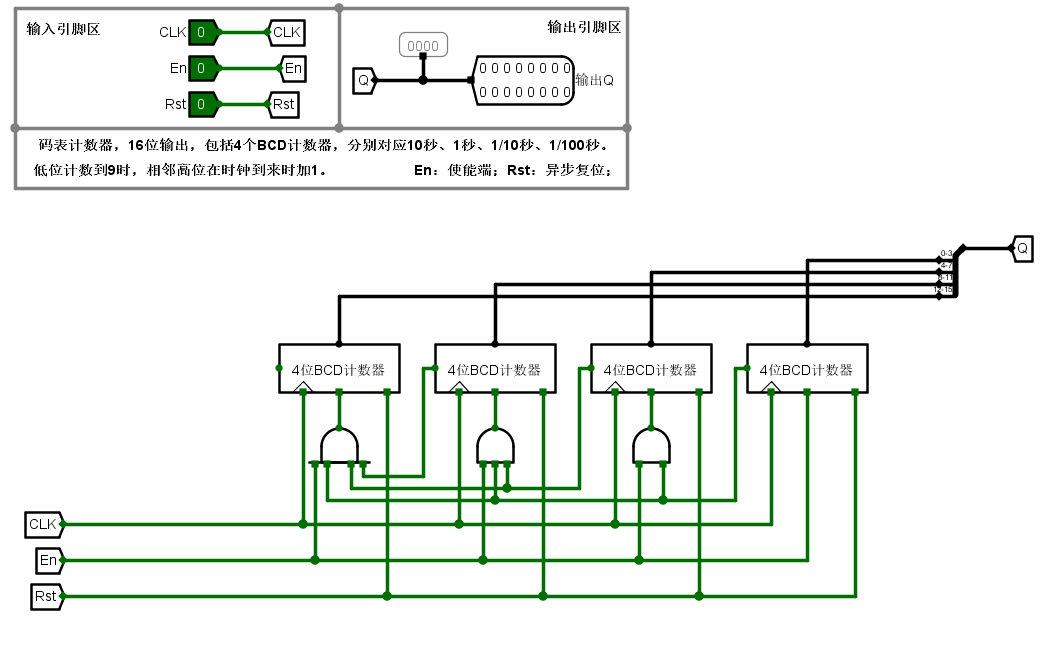
它的状态转换电路如下图。

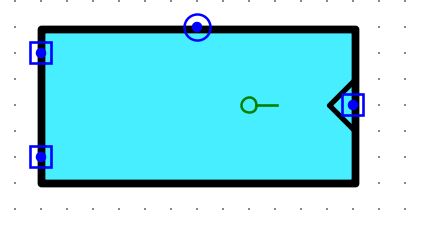


其输出函数电路如下图。



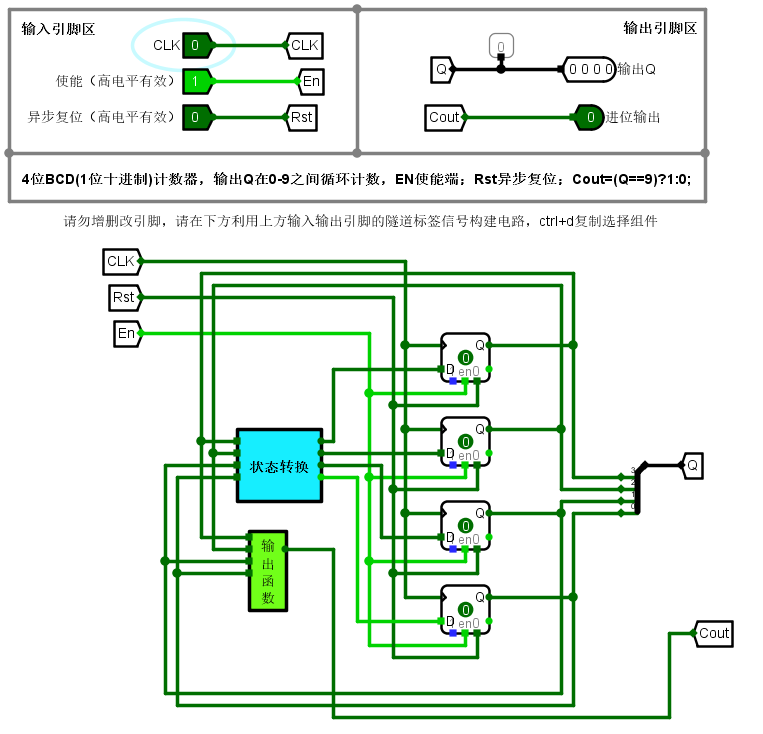
其次是码表计数器，即16位输出的4个BCD计数器。



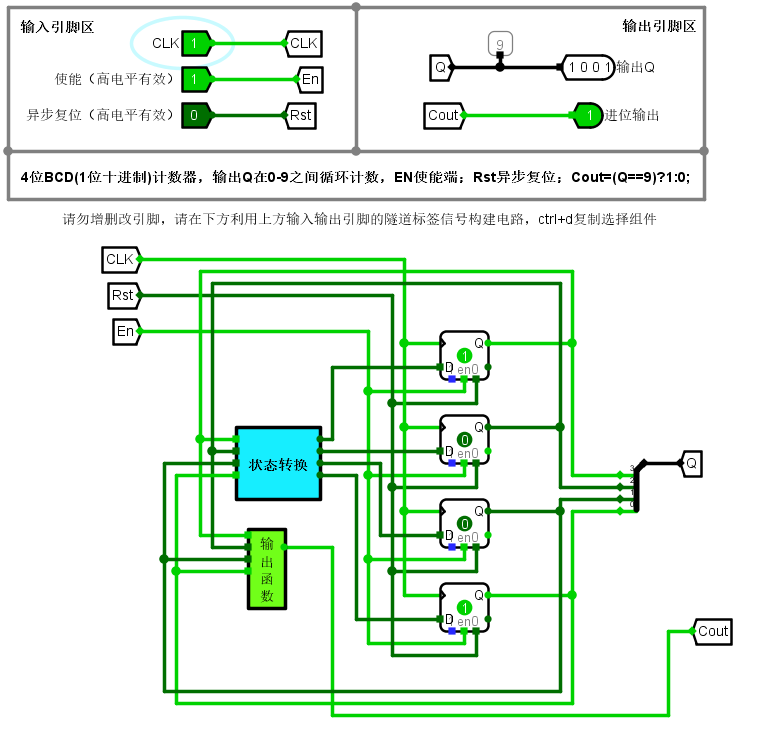


（2）测试图

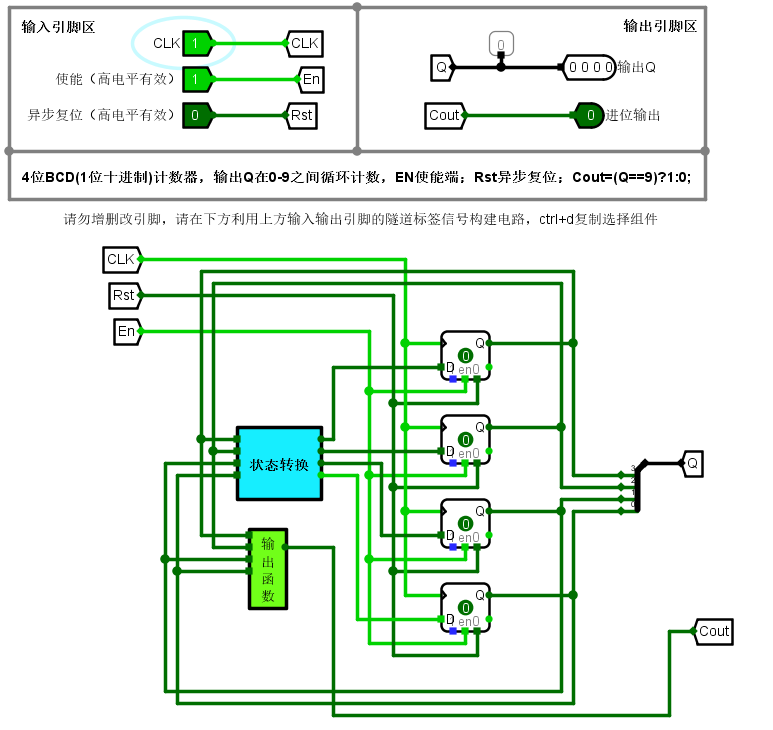
初始状态为“0”状态，我们持续给BCD计数器施加CLK信号。



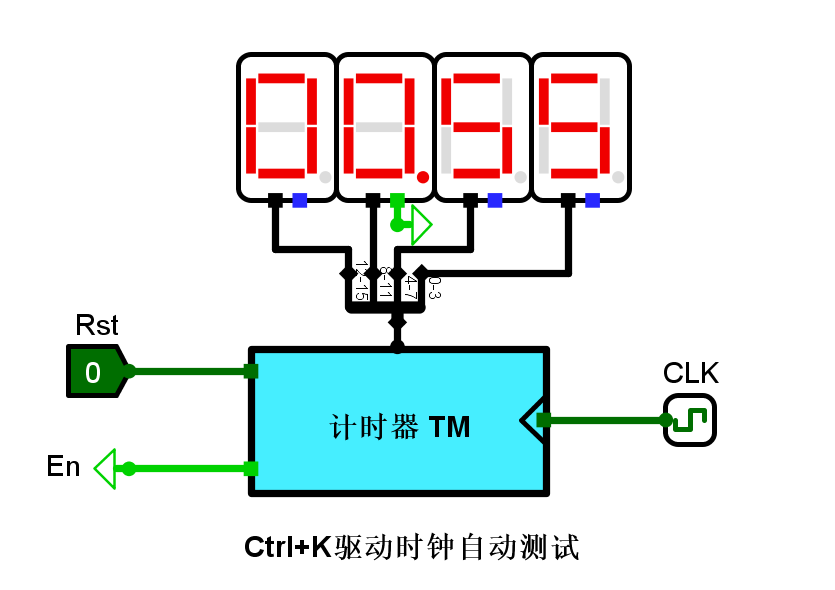
当次态为9时，我们输出Q为“1001”，且进位输出为1。

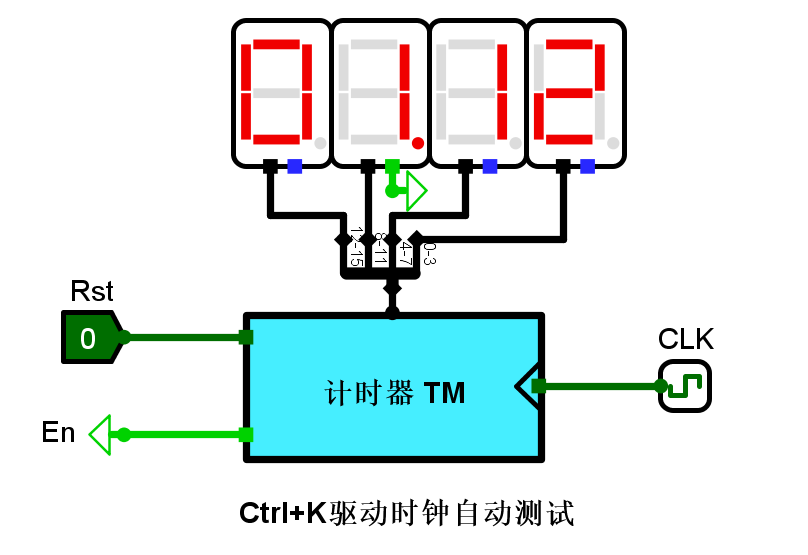


状态为9之后，再一次施加时钟信号，使状态更改为0状态，输出更改为0。



下面我们对集成的码表计数器进行自动测试。如下两图，验证了码表的正常运作。





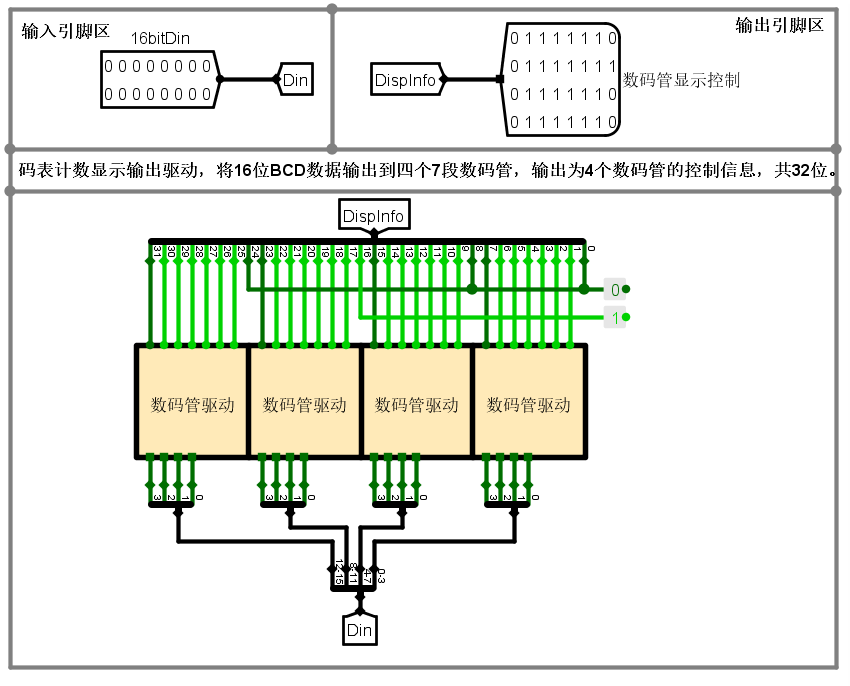
（3）测试分析

BCD计数器的状态转换正常，可以从状态9跳入状态0，且进位输出亦正常。而自动测试也佐证了我们电路设计的正确性。

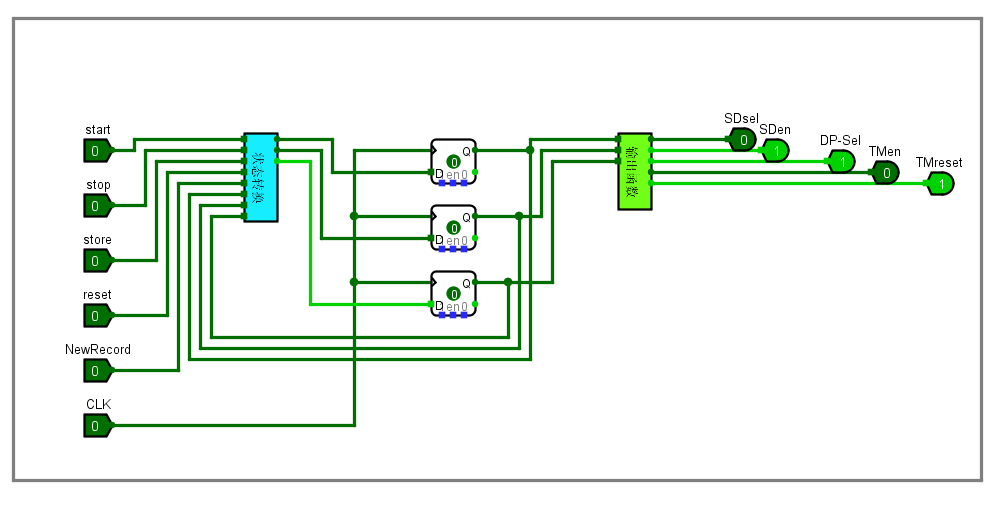
**2.2.6 码表控制器电路**

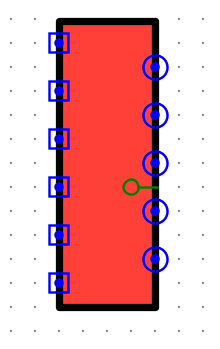
（1）电路图

首先我们给出码表计数显示驱动电路。

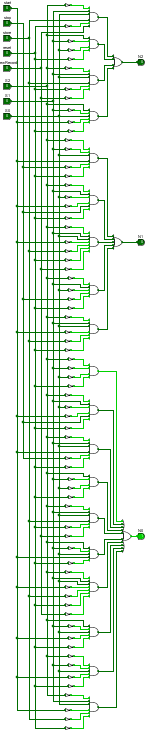


其次是码表控制器的总电路及其封装。

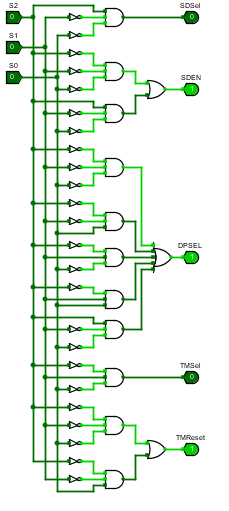




如下是码表控制器的状态转移电路。

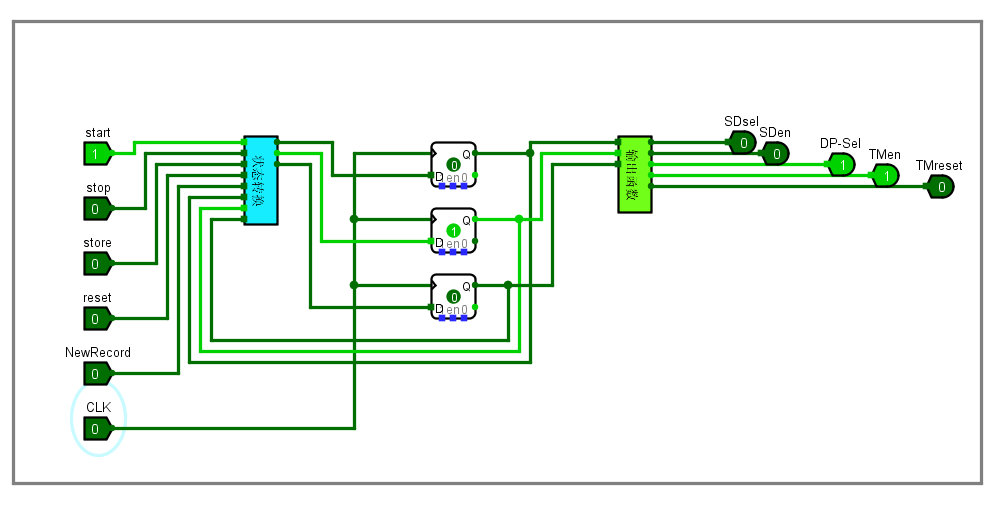


如下是码表控制器的输出函数电路。

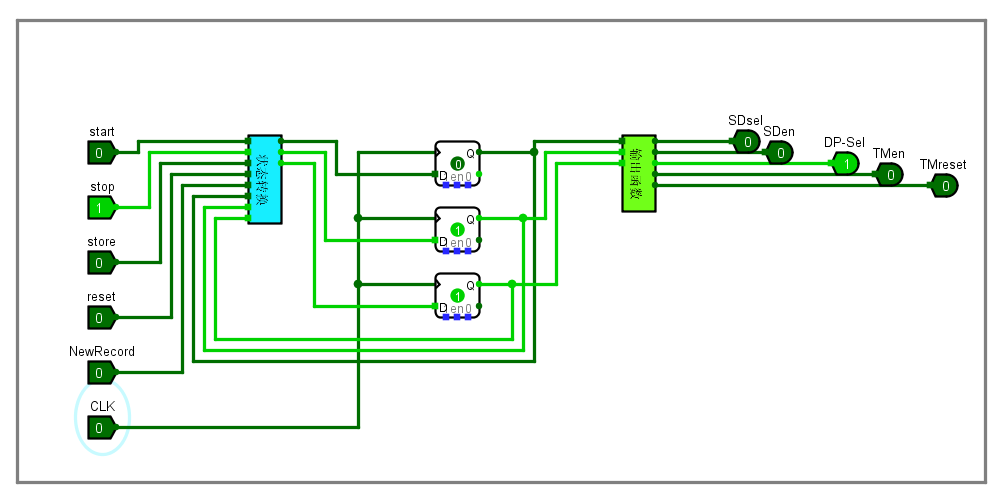


（2）测试图

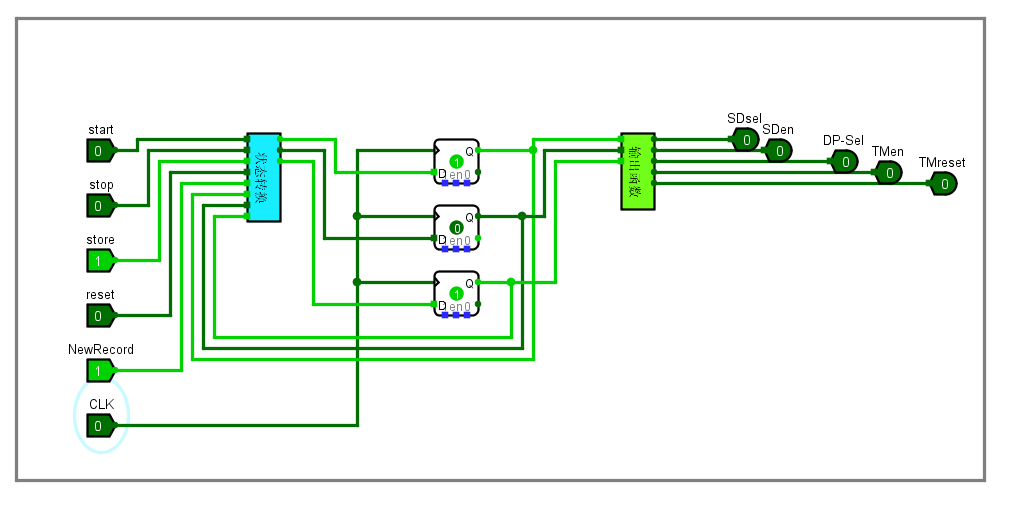
首先我们传入start信号。可见状态转换为“010”。



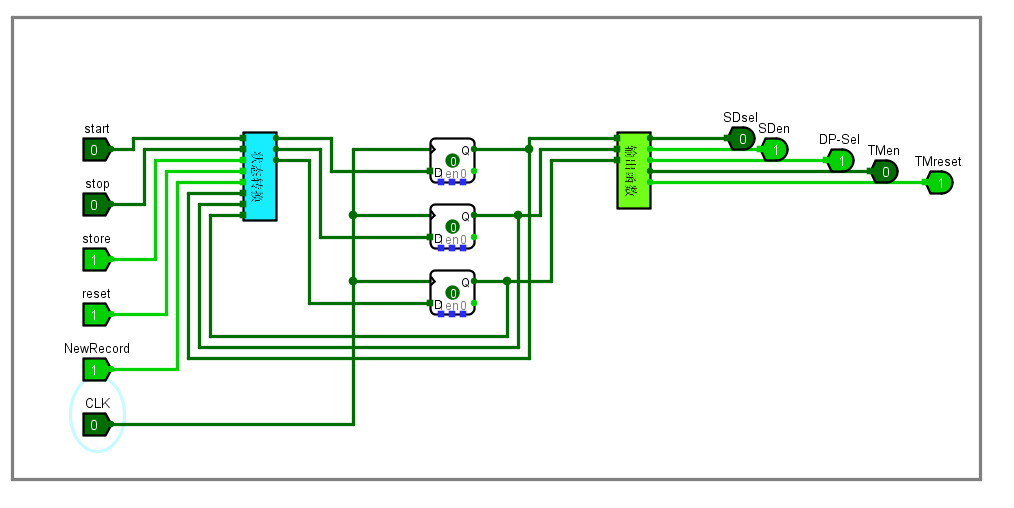
其次我们让码表停止。输入“stop”信号，码表状态更改为“011”，且输出函数为DP-Sel。



再次我们按下store键，即具有store和NewRecord输入。此时，对于的状态更改为“101”，由于我们这是连续传入了两次CLK信号，致使输出函数均为零。



由于reset的优先级最高，只要按下“reset”键，并且正处于时钟上升沿，码表即处于复位状态。



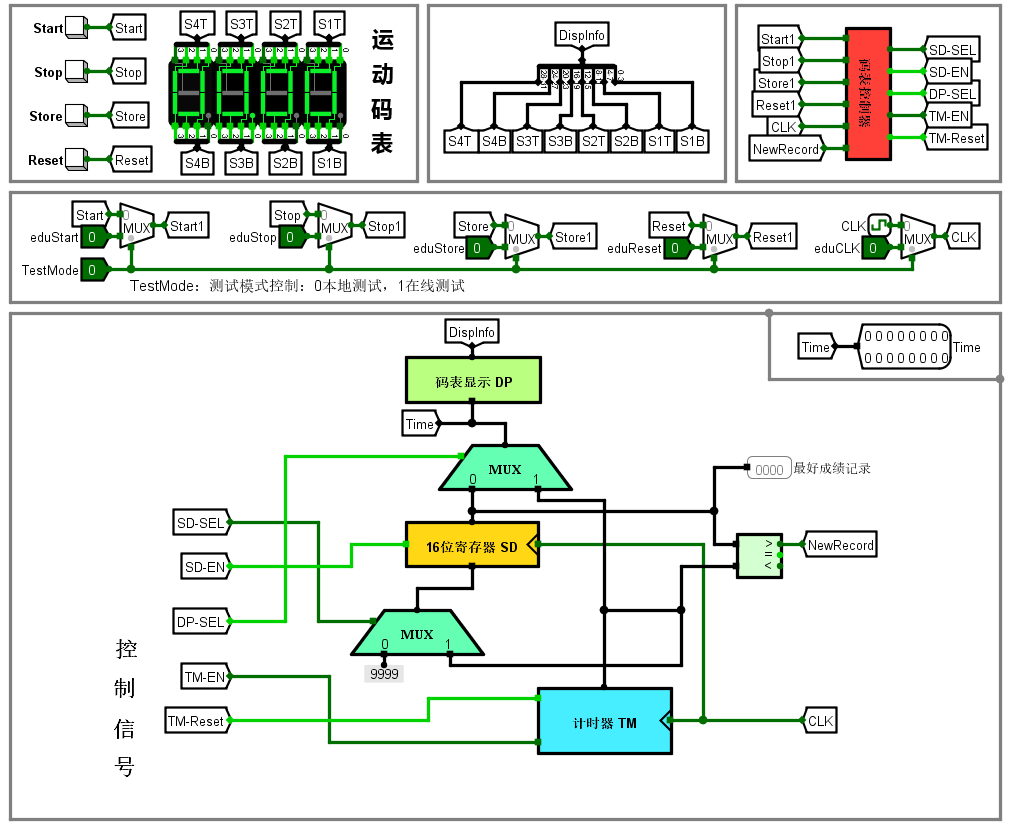
（3）测试分析

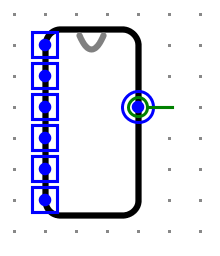
根据我们对码表状态的编码，和我们的测试，都证明了码表状态转移和输出的正确性。

**2.2.10 运动码表**

（1）电路图

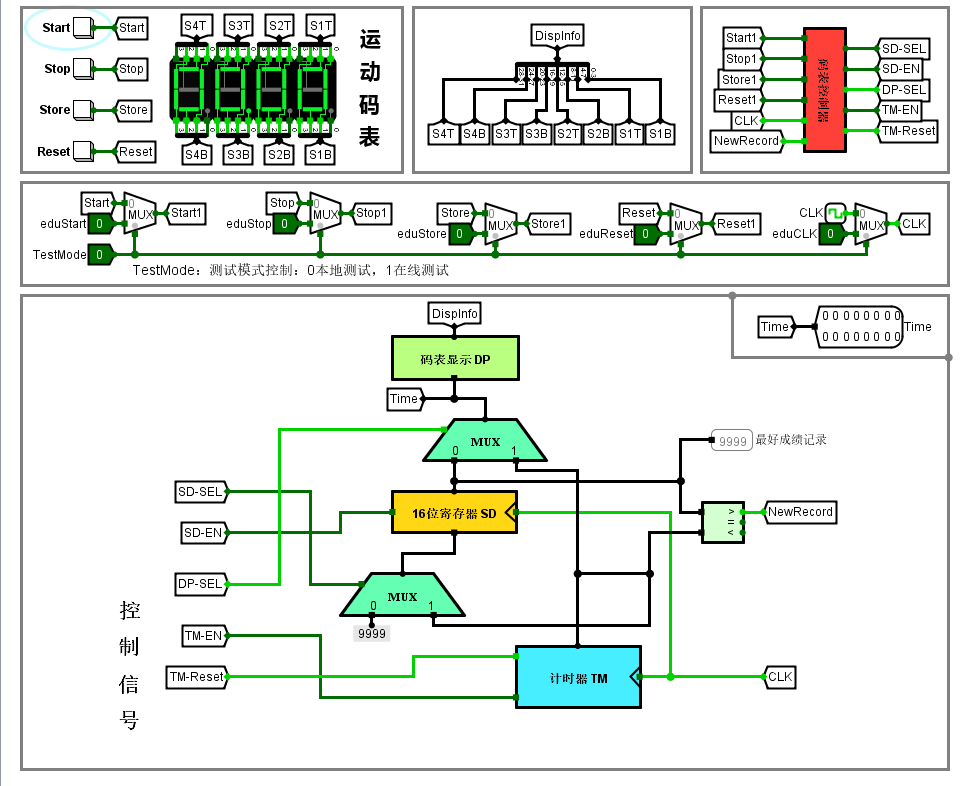
下为运动码表的总电路及其封装。



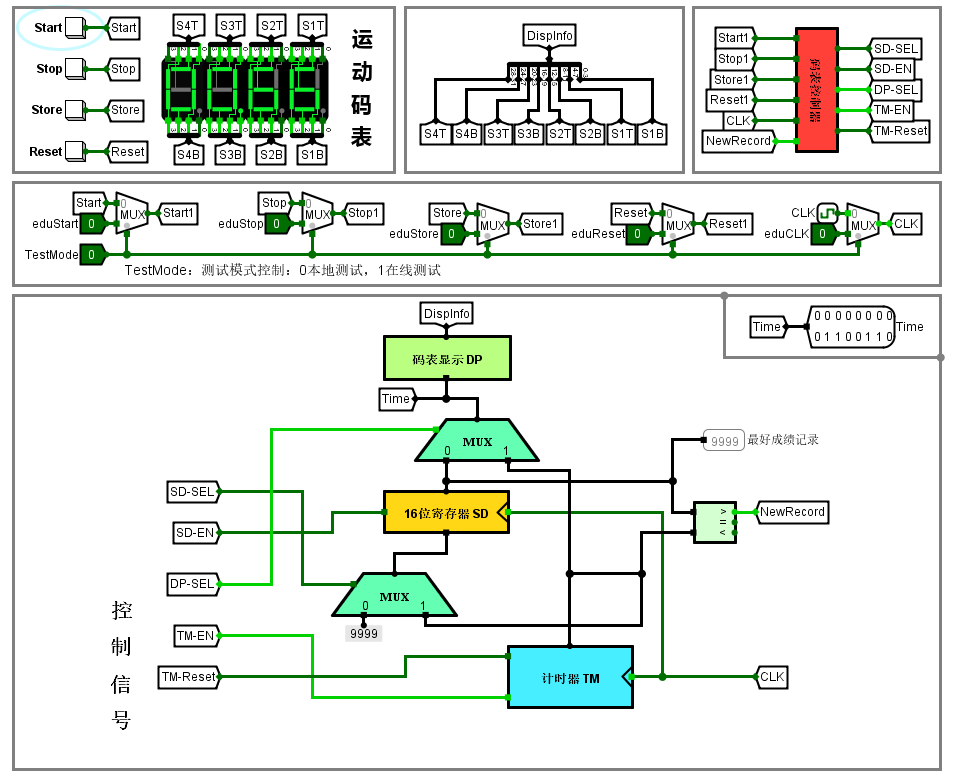


（2）测试图

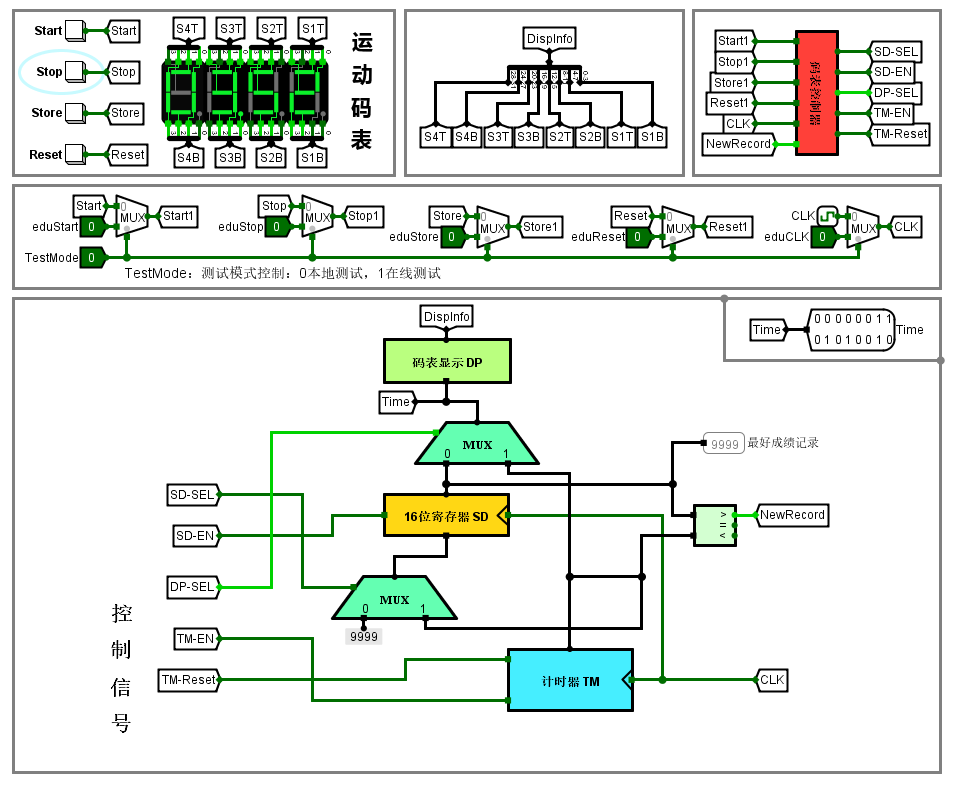
下图为运动码表的初始状态。



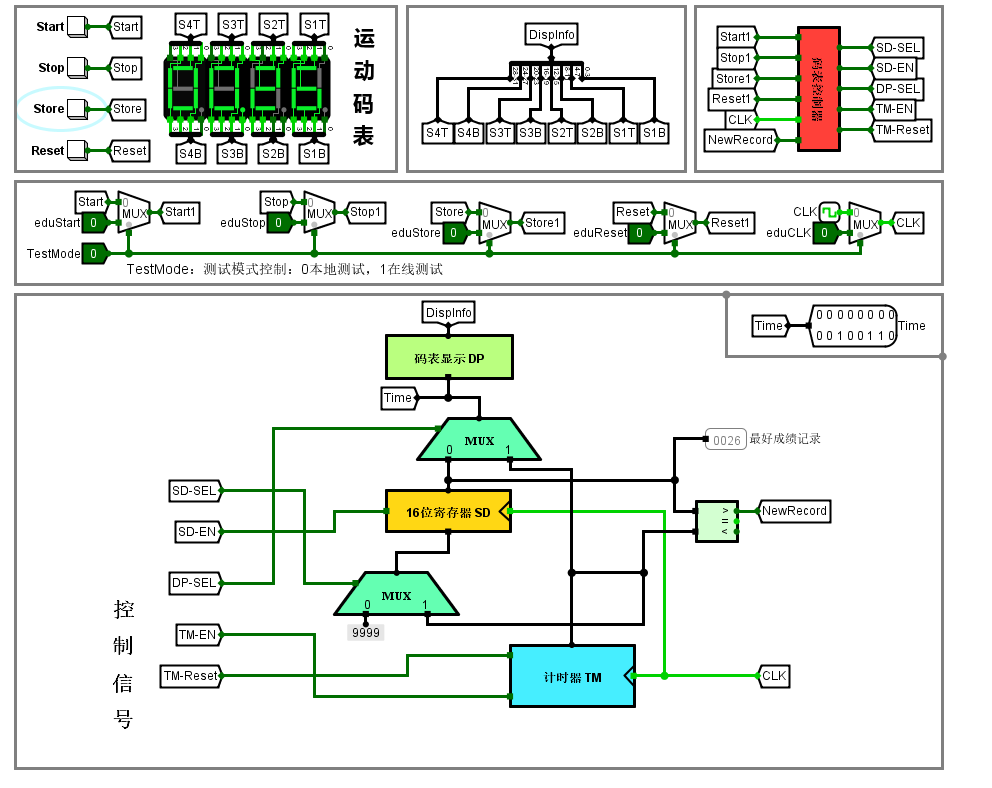
我们对码表进行计时操作。如下图，计时了0.66秒，寄存器存储为最大数99.99，因为还未有新的数据储存入寄存器。



如下图，我们在计数器为3.52秒时让计时器停止，可见码表正常停止。



我们再一次对让码表进行计时。这一次当码表为0.26秒时，我们让码表停止，且按下store键。可见屏幕上显示了0.26秒，且寄存器可存储了相应的值。



（3）测试分析

根据前文所述，码表工作正常。

# 设计总结与心得

## 实验总结

我们对实验过程进行总结。

## 3.1.1遇到的问题及处理

在实验具体设计过程中，我也遇到了一些困难。我印象较深刻的问题是码表控制器的状态转换电路设计中，我由于考虑不周全，数次使得电路输出错误的答案。后来我反复调试了reset相关的状态转移真值表，使得该模块终于输出了正确的答案。

另一个遇到的问题在于码表计数器中，我以为使能端可以是常数高电平，未将使能端与四个BCD计数器依次连接，导致了该电路的设计不能得到正确答案。后来我经过反复的观察发现了这一个漏洞，并解决了问题。

## 3.1.2设计方案存在的不足

我们所设计的码表功能不够齐全。在真实的码表应用场景中，需要记录多组数据以供教练使用，且我们未设计“暂停”键，只设计了“停止”键。实际码表经常需要暂停计时，而后恢复计时，而我们并未对该功能进行支持。

## 实验心得

数字电路是一门需要动手，需要“make hands dirty”的学科。在实验具体设计过程中，我对寄存器、BCD计数器等一系列基础元件有了更加深刻了理解，同时也对Logisim这一精简的软件深感其强大。

通过这门实验课，我基本学习了Logisim的软件使用方法，并且学习了数字电路的设计原则和基本方法。我们将根据数字电路的学习，为将来计算机组成原理的学习而继续加油！

## 意见与建议

这是一门有趣的实验。为了使实验更有意思，我建议可以加大实验的自由度，可以不止是按部就班地制作运动码表，而可以为该码表加入一些自定义元素，比如“暂停”键选项。

|  |
| --- |
| 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：王彬** |

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：计算机2101-U21010101-张三-运动码表实验报告

全班收齐后统一打包压缩交给指导教师。

|  |
| --- |
|  |