

华中科技大学

2023

数字电路与逻辑设计  
实验报告

专    业：	网络空间安全
班    级：	网安 2104 班
学    号：	U202112131
姓    名：	邬雪菲
电    话：	18172029686
邮    件：	744720424@qq.com
完成日期：	2023 年 4 月 17 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
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）\*0.6+实验报告\*0.4

---

---

## 目 录

<b>1</b>	<b>实验概述 .....</b>	<b>1</b>
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 .....	2
<b>2</b>	<b>实验总体设计 .....</b>	<b>3</b>
2.1	实验总体设计思路 .....	3
2.2	实验总体设计框架 .....	4
<b>3</b>	<b>实验过程 .....</b>	<b>5</b>
3.1	7 段数码管驱动电路设计 .....	5
3.2	2 选 1 选择器设计（1 位，16 位） .....	6
3.3	无符号比较器设计（16 位） .....	7
3.4	并行加载寄存器（4 位，16 位） .....	10
3.5	BCD 计数器设计（含状态机、输出函数和整体电路） .....	12
3.6	码表计数器设计（4 位十进制） .....	14
3.7	码表显示驱动设计 .....	16
3.8	码表控制器状态机设计 .....	17
3.9	码表控制器输出函数设计 .....	19
3.10	码表控制器 .....	21
3.11	运动码表 .....	22
<b>4</b>	<b>设计总结与心得 .....</b>	<b>24</b>
4.1	实验总结 .....	24
4.1.1	遇到的问题及处理 .....	24
4.1.2	设计方案存在的不足 .....	25
4.2	实验心得 .....	25

---

---

4.3 意见与建议.....	26
----------------	----

---

# 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](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 实验总体设计

### 2.1 实验总体设计思路

本实验要求使用 Logisim 设计仿真一个运动码表系统，需要 4 个按钮控制码表工作和 4 个 7 段数码管来显示计时。具体来说，本系统需要实现四个具体功能：启停计时、存储更新计时结果、比较计时数据、复位计时器，由此我设计了五个模块实现功能：计时器、存储器、比较器、显示器和控制器。

首先，是计时器模块的设计。按下 Start 按钮时，计数器清零并开始计时；按下 Stop 按钮时，计时器停止计时，并且将计时数据传送给显示器模块进行显示。按下 Store 按钮时，需要比较当前计时数据和系统记录的值，选择性更新系统记录，并将对应数据传送给显示器显示。按下 Reset 按钮时，计时器和系统记录复位为 00.00 和 99.99，并传送 00.00 至显示器显示。

其次是存储器，需要实现设计一个 16 位并行加载寄存器存储历史最优数据，并且该寄存器的数据可以由码表控制器控制修改和显示。一个 16 位并行加载寄存器由 4 个并行加载寄存器组成，而一个 4 位并行加载寄存器由 4 个 D 触发器组成，分别存储一位二进制位。

然后是比较器，作用是比较两个十六进制数据的大小，与控制器紧密联系协同发挥作用。若新数据更小，需要将其传输给存储器和显示器，否则不存储新数据，并将存储器数据返回显示器显示。

对于显示器模块，其包含 4 个数码管，功能是将计时数据或存储数据以七段数码管的形式显示。单个七段数码管驱动电路采用真值表方式进行构建。由码表控制器电路决定传送到显示器的数据。

最后，是码表控制器模块，用于连接各个模块和 4 个按钮输入，起到控制整个码表电路运行的作用。并且，控制器需引入二路选择器，以与存储器和比较器配合实现比较更新最优记录。

由设计思路，分别对各个模块进行设计，结合组合逻辑电路和复杂时序逻辑电路，逐步实现各个模块中的各个逻辑单元，最终实现运动码表系统。

## 2.2 实验总体设计框架

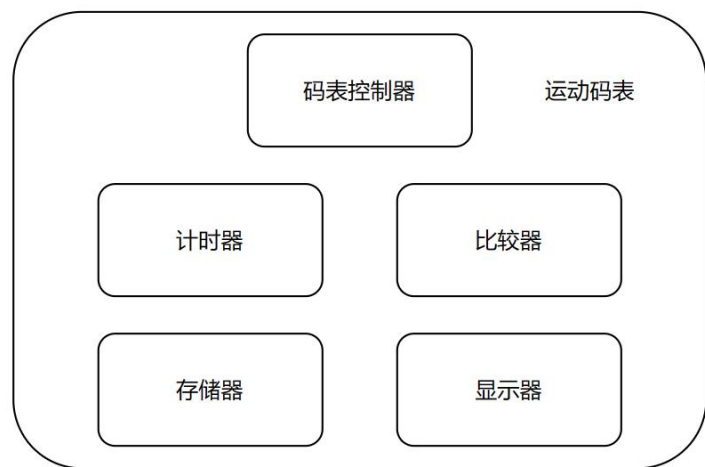


图 2-2-1 运动码表设计框架图

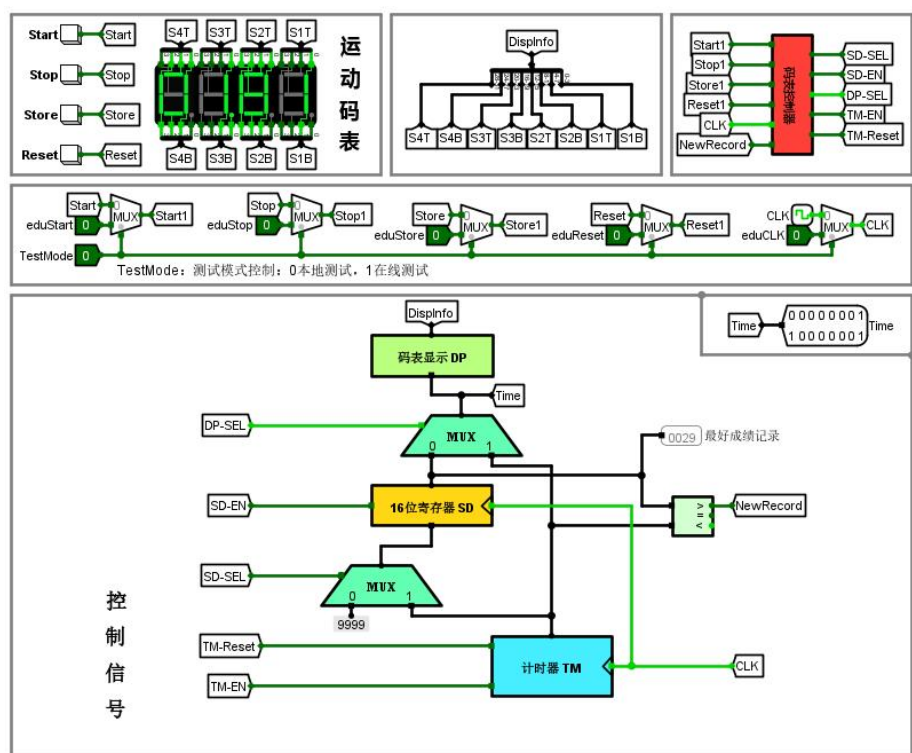


图 2-2-2 运动码表数据通路图



---

## 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

根据数码管的引脚顺序，依次将不同的四位二进制输入对应的数码管引脚列出来，由此得到真值表，完成输入与输出的映射，最后借助 Logisim 的“分析组合逻辑电路”功能填写真值表自动生成电路。

#### (2) 电路图

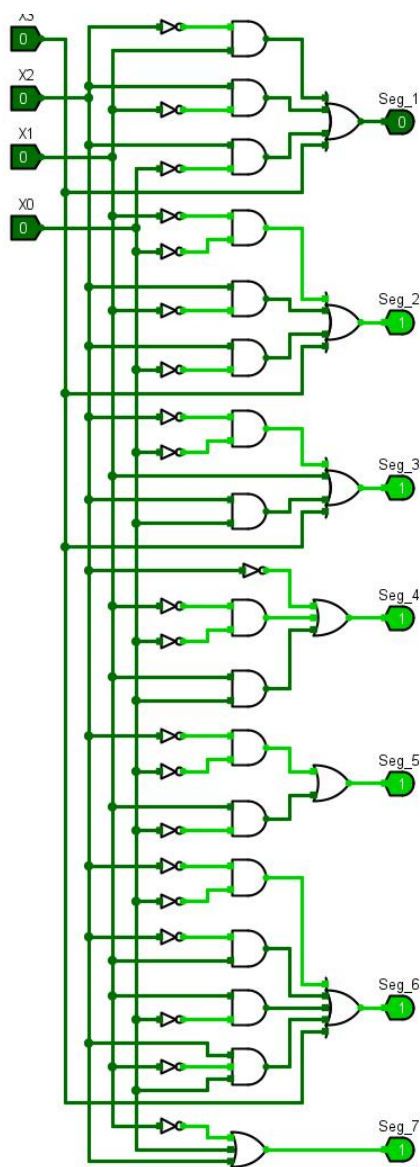


图 3-1-1 七段数码管驱动电路图

### (3) 测试图

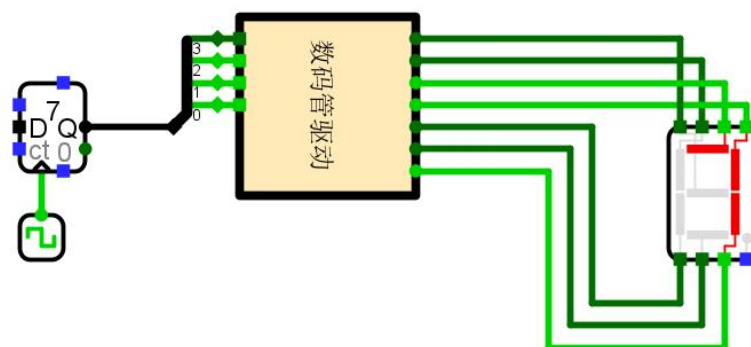


图 3-1-2 七段数码管驱动测试图

### (4) 测试分析

进入时钟单步模式，不断点击时钟端，看到 D 触发器之中的数字与七段数码管之中的数字一一对应，测试通过。在头歌的测试也均通过，说明设计成功。

## 3.2 2 选 1 选择器设计（1 位，16 位）

### (1) 设计思路及设计过程

#### A) 1 位 2 路选择器设计：

根据 2 路选择器的电路功能： $Out = (Sel == 0) ? x0 : x1$ ，可以得出电路逻辑表达式  $Out = \sim Sel * x0 + Sel * x1$ ，使用非门、与门、或门连接各个输入得到最终电路。

#### B) 16 位 2 路选择器设计：

与 1 位 2 路选择器类似，16 位选择器的功能是： $Out = (Sel == 0) ? x0 : x1$ ，只不过输入数据变成了 16 位，不妨使用复用器中的位选择器代替原来的与门，设置 Sel 为位选择器的选择端，同时修改相应器件的数据输出和输入位为 16 位，即可完成设计。

### (2) 电路图

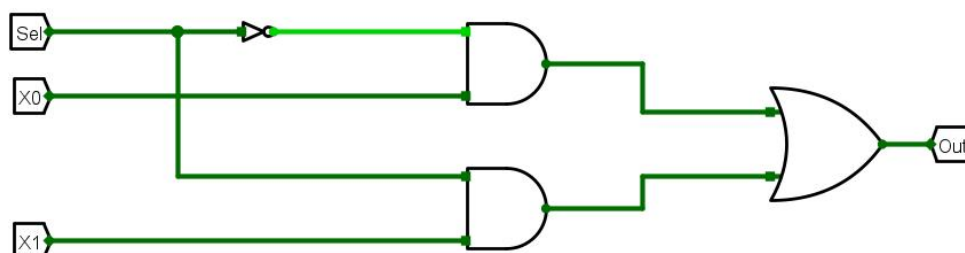


图 3-2-1 1 位 2 路选择器

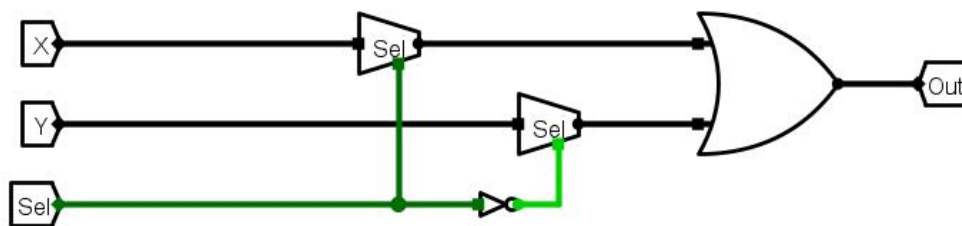
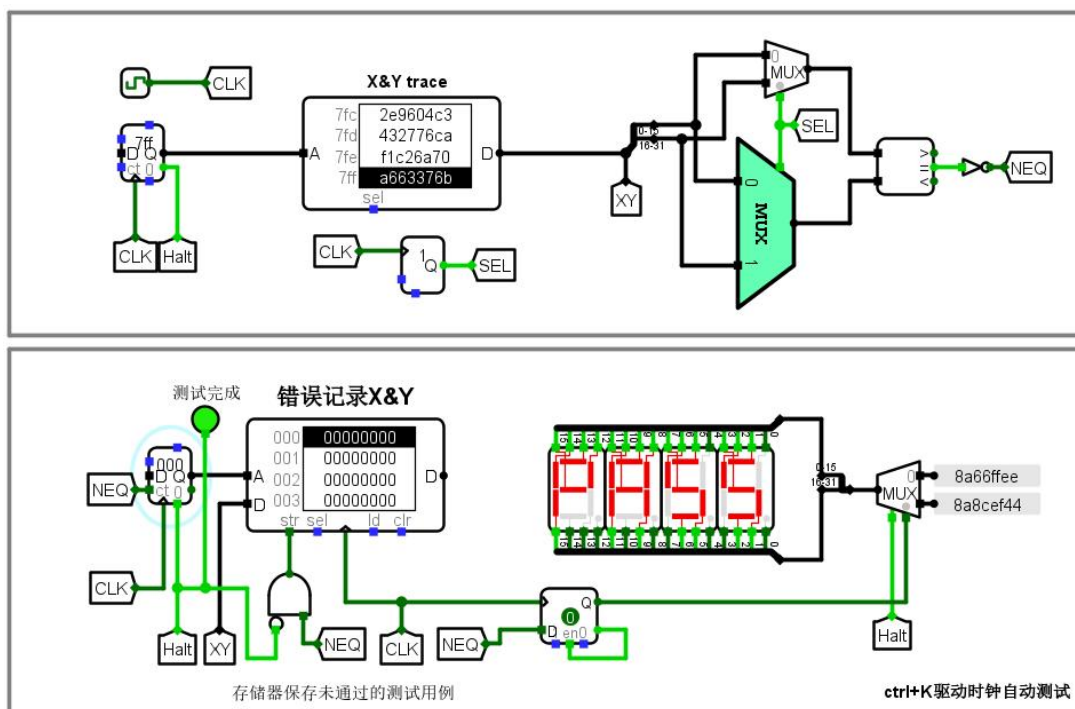


图 3-2-2 16 位 2 路选择器

### (3) 测试图



多路选择器自动测试

图 3-2-3 多路选择器测试图

### (4) 测试分析

根据提供的多路选择器自动测试，观察到 x 与 y 比较之后的输出均是正确的，且测试中也看到了数码管显示“PASS”的单词，在头歌平台的测试均通过，说明电路设计成功。

## 3.3 无符号比较器设计（16 位）

### (1) 设计思路及设计过程

首先设计 4 位的无符号比较器，再使用设计好的 4 位无符号比较器完成 16 位无符

号比较器的电路设计。对于 4 位无符号比较器，不妨从高位向低位来进行一一比较，若是高位  $x_i > y_i$  则直接得到  $x > y$  的结论，如果高位  $x_i < y_i$  则直接得到  $x < y$  的结论，如果相等则继续比较下一位，并重复之前的流程。如果四位相等则得到  $x = y$ 。由此逻辑可手动连接得到电路图。

对于 16 位无符号比较器，可以将它的每四位看作是一个，对 4 个 4 位的处理类比为 4 个 1 位的处理，则可使用和 4 位无符号比较器相同的逻辑设计电路。只是需要把比较两位的电路使用设计好的 4 位无符号比较器代替，则可判断两个数的对应 4 位的大小关系，由此完成设计。

## (2) 电路图

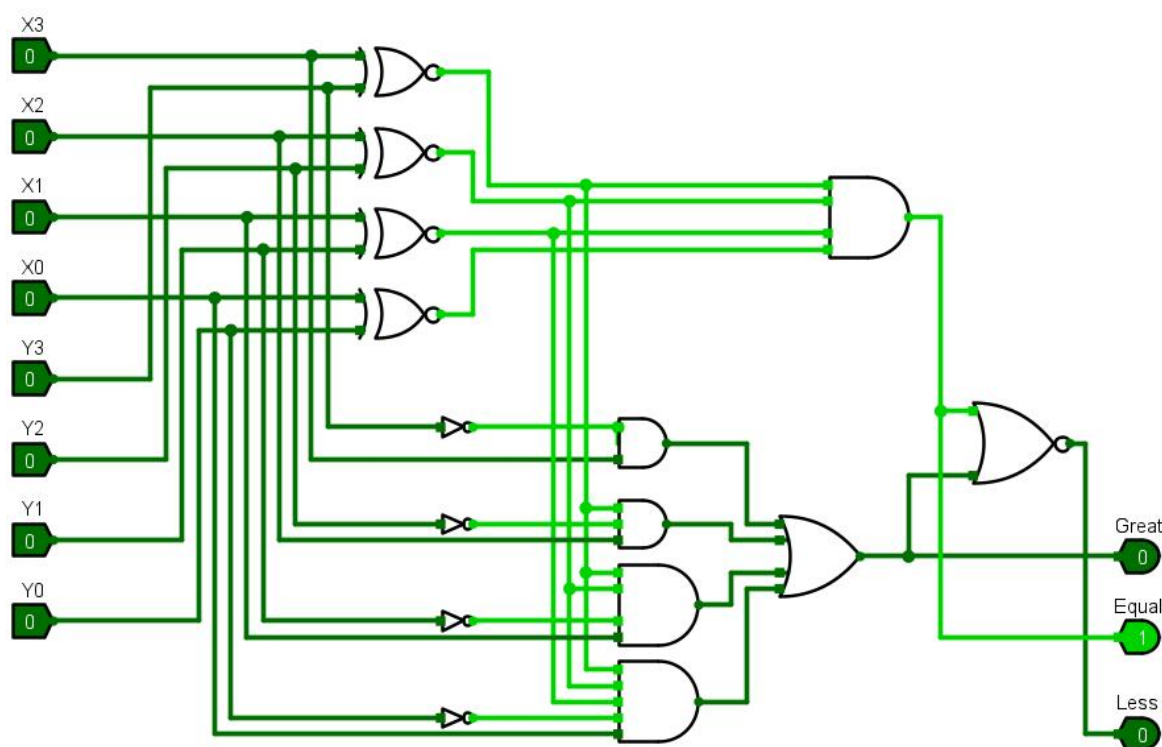


图 3-3-1 4 位无符号比较器电路图

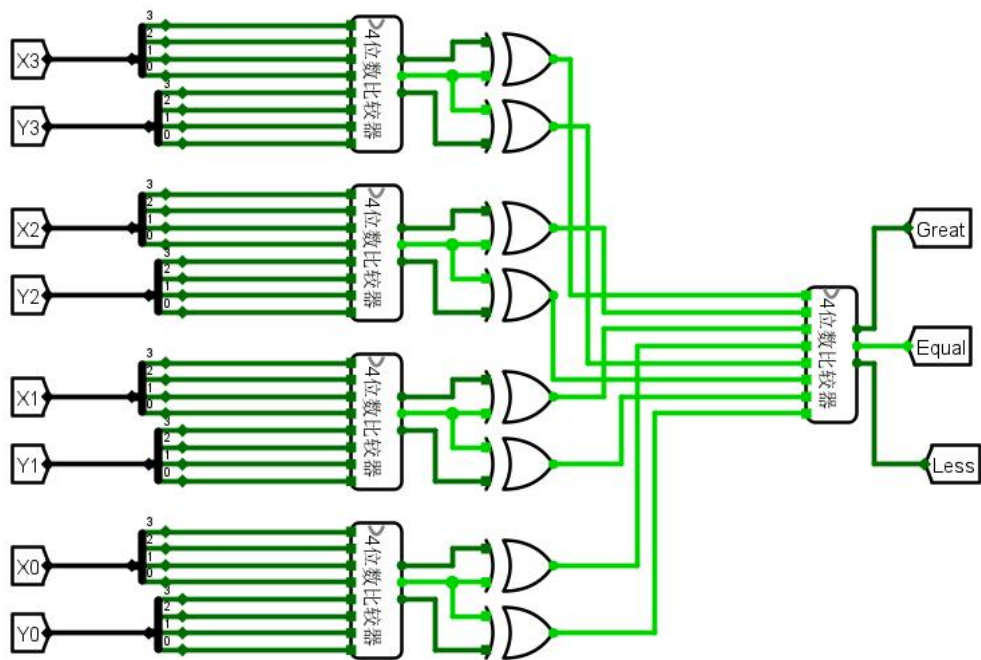
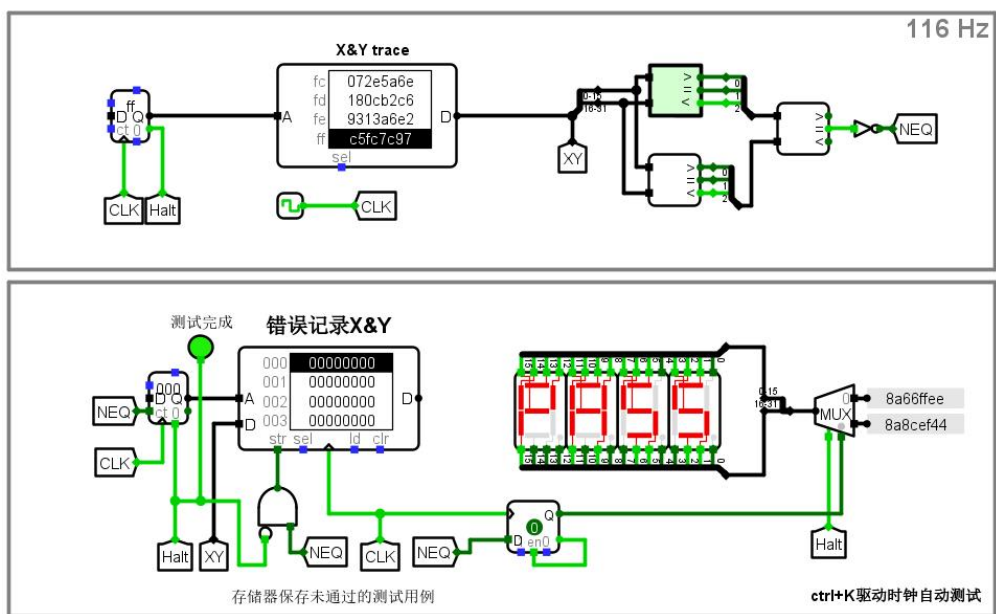


图 3-3-2 16 位无符号比较器电路图

### (3) 测试图



16位数无符号比较器自动测试

图 3-3-3 16 位无符号比较器测试图

### (4) 测试分析

根据自动测试的结果，看到了“PASS”单词，且头歌平台测试通过，可知电路设计无误。

### 3.4 并行加载寄存器 (4 位, 16 位)

#### (1) 设计思路及设计过程

首先设计 4 位的并行加载器，之后直接由 4 位的并行加载器组装得到 16 位并行加载器。

对于 4 位并行加载器，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，将触发器的使能端与时钟端和对应的输入相连之后，通过四位分线器将 D 触发器的输出汇总成一个 4 位的输出。

对于 16 位并行加载器，只需要 4 个 4 位并行加载器即可构成 16 位并行加载器。将 4 位寄存器的使能端与时钟端与对应的输入相连之后，使用 16 位分线将输入分为 4 个 4 位输入之后，依次与 4 个 4 位并行加载器相连，再将 4 个 4 位输出通过分线器汇总输出。

#### (2) 电路图

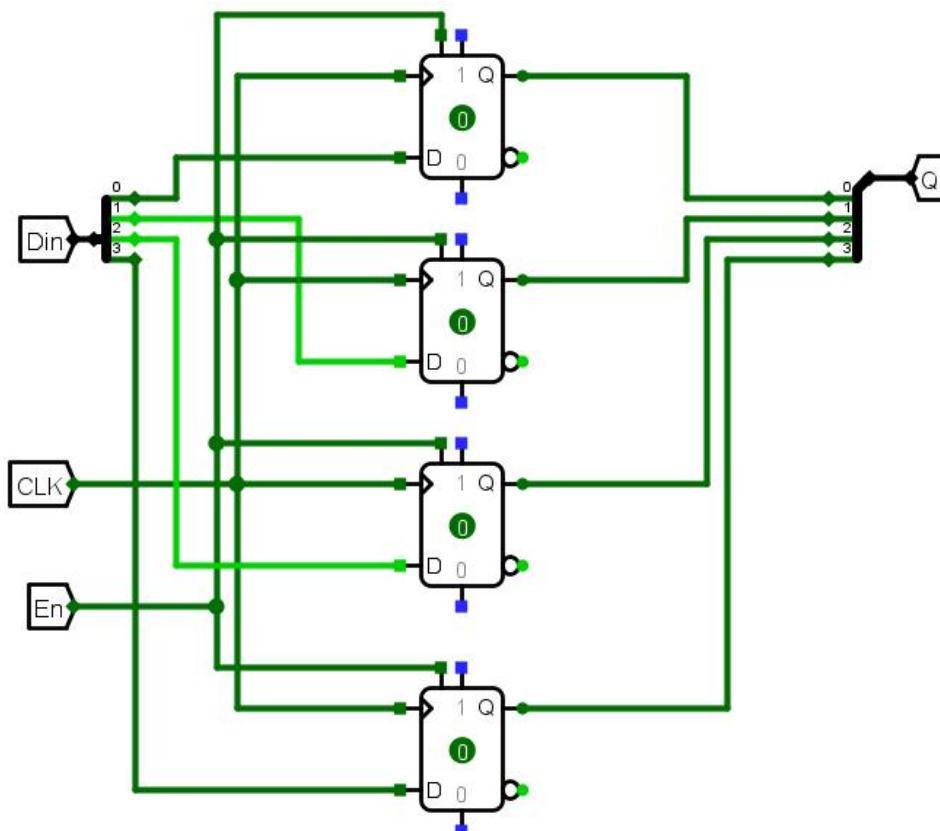


图 3-4-1 4 位并行加载寄存器电路图

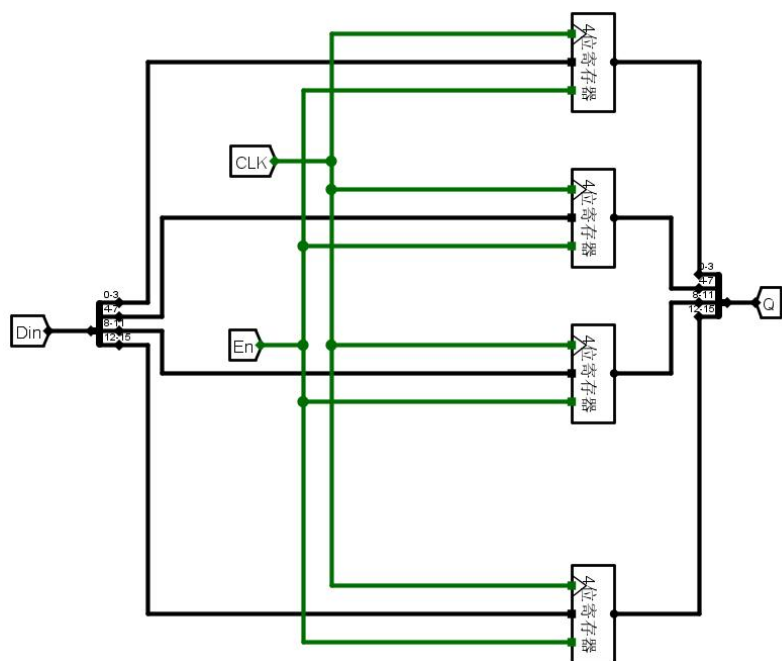


图 3-4-2 16 位并行加载寄存器电路图

### (3) 测试图

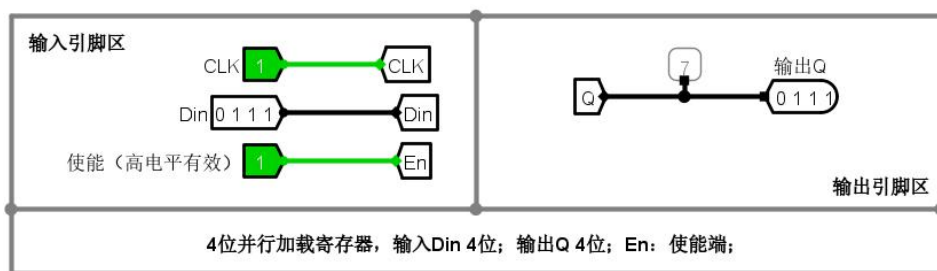


图 3-4-3 4 位并行加载器测试图

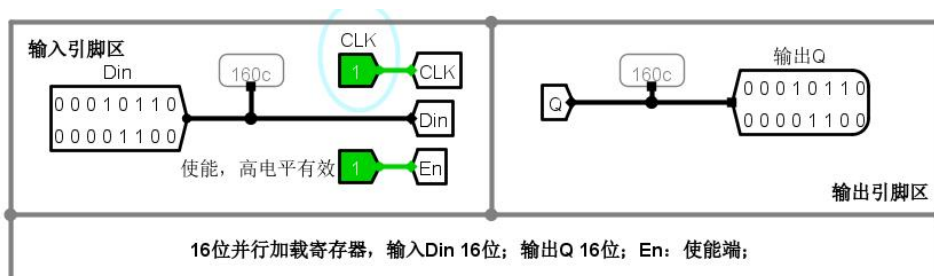


图 3-4-4 16 位并行加载器测试图

### (4) 测试分析

测试中给出了两个样例四位的 0111 和十六位的 160c，发现输出引脚区与输入引脚区的数据相同，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。



### 3.5 BCD 计数器设计（含状态机、输出函数和整体电路）

#### （1）设计思路及设计过程

##### ①整体电路：

先将四位的输入通过分线器分为四个一位的输入，再将 4 个一位输入分别连接到 4 个 D 触发器当中，然后将各个触发器的置位端、时钟端和使能端都和相应的输入相连，最后将 D 触发器的输出连接到状态机和输出函数中，状态转换后的输出接到 D 触发器的 D 输入。

##### ②状态机：

直接在相应的 excel 表之中填写相应的状态转换关系（如下图），通过 excel 逻辑表达式生成表得到相应的状态转换函数逻辑表达式，在 Logisim 之中以“工程—分析组合逻辑电路”的表达式生成的方式即可生成相关电路。

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	In1	In2	In3	In4	In5	In6	In7	In8	次态 10进制	N3	N2	N1	N0
0	0	0	0	0									1	0	0	0	1
0	0	0	1	1									2	0	0	1	0
0	0	1	0	2									3	0	0	1	1
0	0	1	1	3									4	0	1	0	0
0	1	0	0	4									5	0	1	0	1
0	1	0	1	5									6	0	1	1	0
0	1	1	0	6									7	0	1	1	1
0	1	1	1	7									8	1	0	0	0
1	0	0	0	8									9	1	0	0	1
1	0	0	1	9									0	0	0	0	0

图 3-5-1 BCD 计数器状态机 excel 表格图

##### ③输出函数：

当且仅当输入为 1001 时 cout 才为 1，故直接在相应的 excel 表之中填写相应的状态与输出的关系（如下图），通过 excel 表直接得到相应的输出函数表达式，在 Logisim 之中以“工程—分析组合逻辑电路”的表达式生成的方式即可生成相关电路。



当前状态(现态)					输入信号								
S3	S2	S1	S0	现态 10进制	In1	In2	In3	In4	In5	In6	In7	In8	Out1
0	0	0	0	0									
0	0	0	1	1									
1	0	1	0	2									
0	0	1	1	3									
0	1	0	0	4									
0	1	0	1	5									
0	1	1	0	6									
0	1	1	1	7									
1	0	0	0	8									
1	0	0	1	9									1

图 3-5-2 BCD 计数器输出函数器 excel 表格图

(2) 电路图

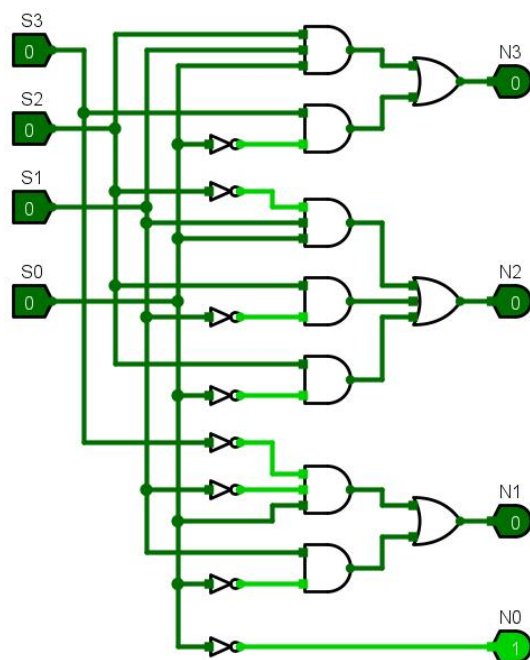


图 3-5-3 BCD 计数器状态机电路图

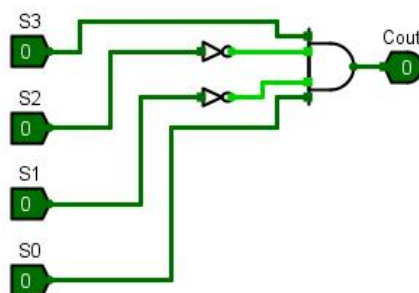


图 3-5-4 BCD 计数器输出函数电路图

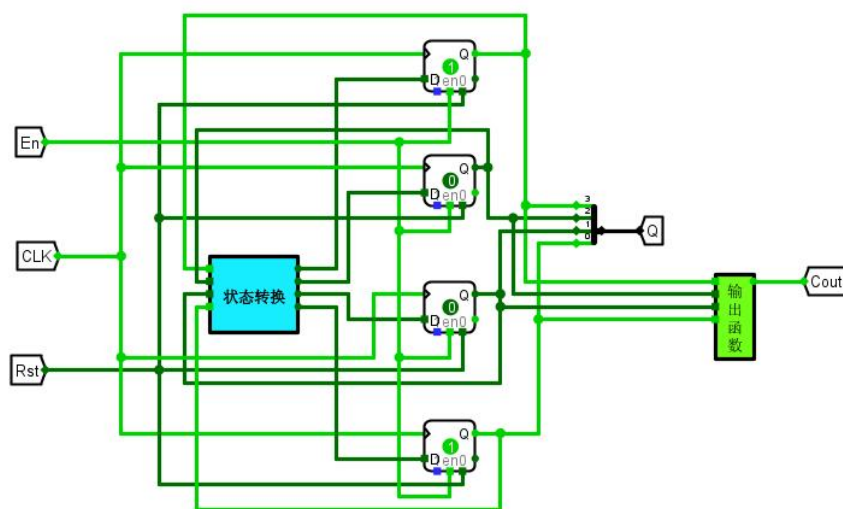


图 3-5-5 BCD 计数器电路图

### (3) 测试图

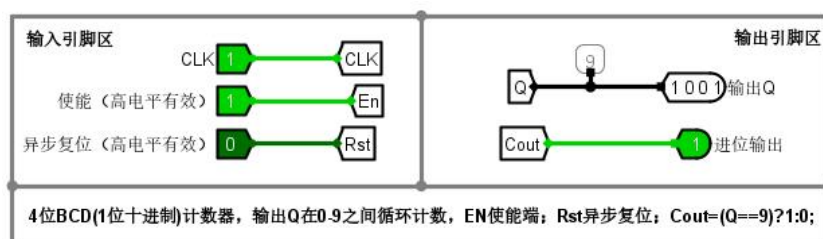


图 3-5-6 BCD 计数器测试图

### (4) 测试分析

随着时钟端的不断点击，输出引脚区的输出 Q 二进制在上升沿不断加一自增，且仅在输出为 9 时产生进位 1，且头歌平台测试均通过，由此可知设计成功。

## 3.6 码表计数器设计（4 位十进制）

### (1) 设计思路及设计过程

码表计数分为四位，分别从高到低表示个十百千，计数从最低位开始，满了就往前进一位，除了第一位的时钟端接真正的 CLK，其他三位接的是前一位的进位输出，因为只有前一位进位了后一位才应开始计数。

同时为了防止电路是 Moore 型导致输出前置的情况，这里添加了非门来消除相关情况。先将个位 BCD 计数器的置位、时钟端与相应的输入直接相连，而个位的进位输出与非门相连之后再接到十位的时钟端中。十位和百位的进位输出也和非门连接后

接入下一位的时钟端。这样在前一位输出为 1-8 时，进位输出为 0，而在其变为 9 后才输出 1，而在下一位的时钟端就是从 1 变到 0，是下降沿，故不触发计数，而当前一位变为 0，进位输出变为 1，下一位的时钟端重新由 0 变为 1，上升沿触发计数，符合设计要求。最后将所有四个 4 位计数器的输出通过分线器汇总之后再输出即可。

(2) 电路图

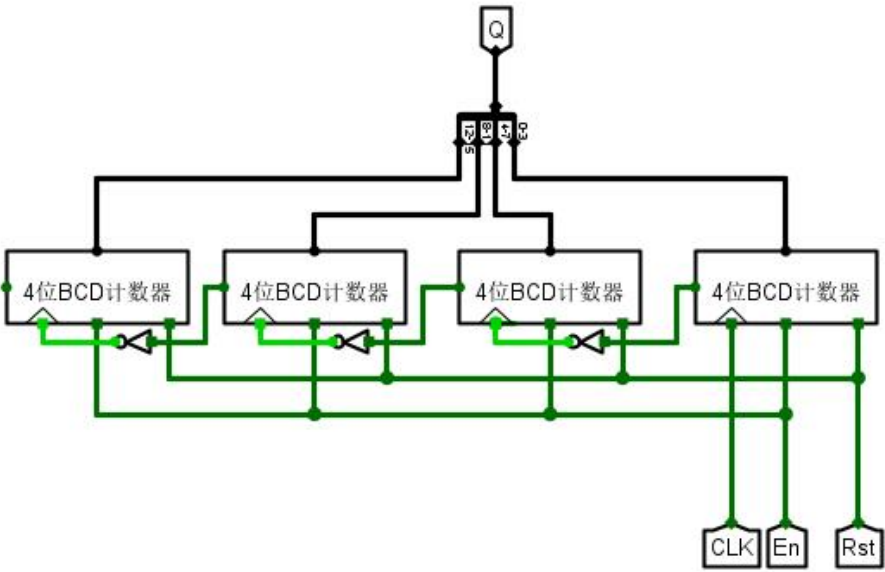


图 3-6-1 码表计数器电路图

(2) 测试图

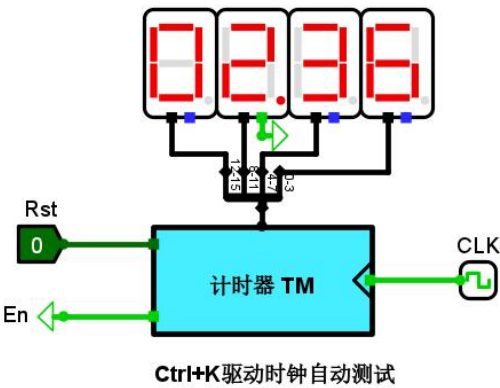


图 3-6-2 码表计数器测试图

(3) 测试分析

通过快捷键“ctrl+k”来驱动时钟端，观察判断得知电路输出无误，且通过头歌平台测试，说明设计成功。

### 3.7 码表显示驱动设计

#### (1) 设计思路及设计过程

将 16 位的输入通过分线器分为 16 个一位输入，根据之前的单个数码管驱动设计电路，以 4 个为一组分别连接到四个不同的七段数码管驱动元件上，将数码管驱动元件的输出与相应位置的 0 与 1 常量端输入通过 32 位分线器汇总输出。

#### (2) 电路图

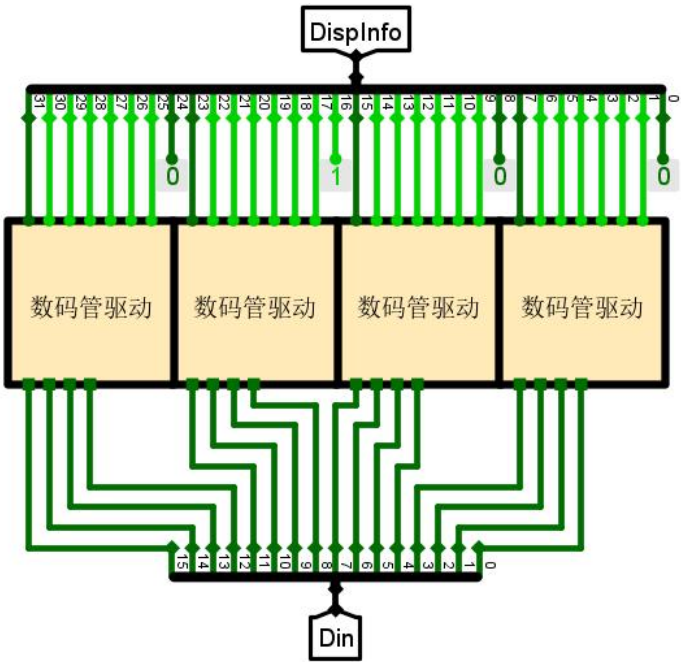


图 3-7-1 码表显示驱动电路图

#### (2) 测试图

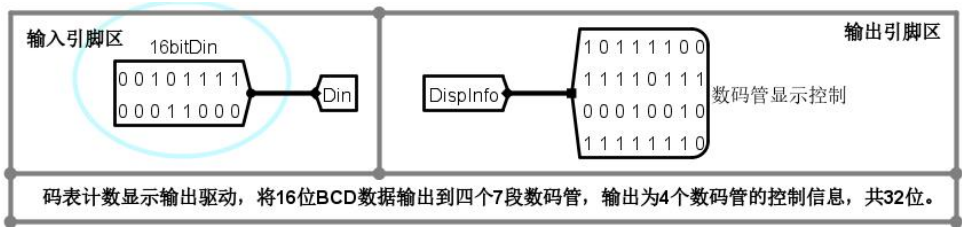


图 2-7-2 码表显示驱动测试图

#### (3) 测试分析

依次调整输入引脚区，看到输出引脚区也在相应地变化，且头歌平台测试通过，说明设计成功。

### 3.8 码表控制器状态机设计

#### (1) 设计思路及设计过程

根据题目给出的基本状态参考图（如下图），再根据总体设计中的功能说明补充隐含的状态转换条件、状态维持的条件等，借助填好的 excel 真值表，直接根据 excel 来生成状态表达式,使用 Logisim 的分析电路-表达式自动生成电路功能即可完成设计。

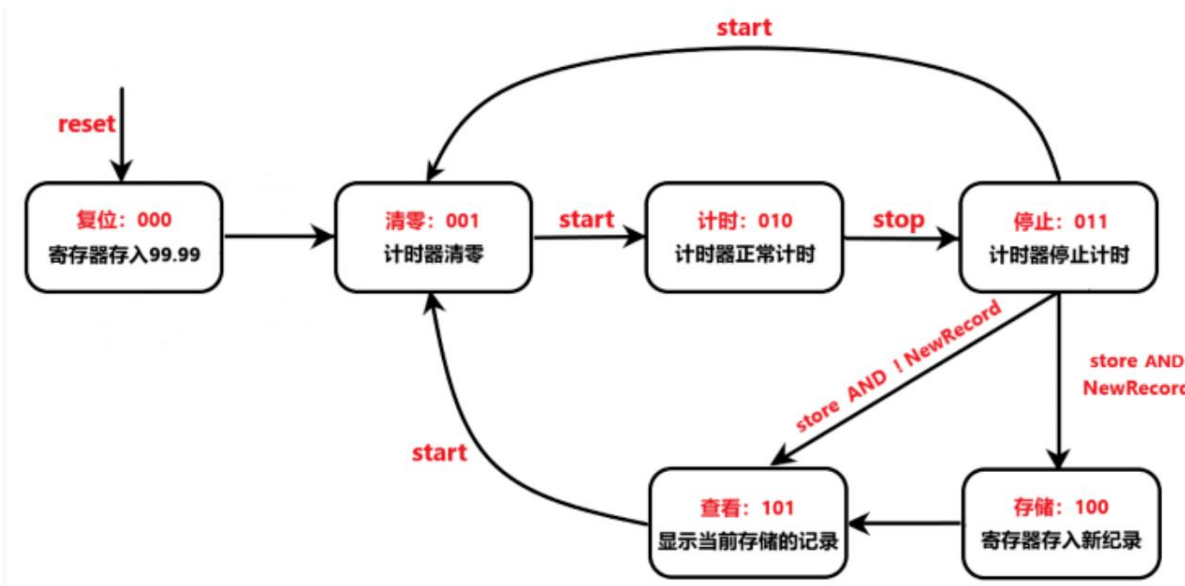


图 3-8-1 码表控制器状态机参考状态图

start	stop	store	reset	newrec	PS	NS
0	0	0	1	0	0	0
0	0	0	1	0	1	0
0	0	0	1	0	2	0
0	0	0	1	0	3	0
0	0	0	1	0	4	0
0	0	0	1	0	5	0
0	0	0	1	1	0	0
0	0	0	1	1	1	0
0	0	0	1	1	2	0
0	0	0	1	1	3	0
0	0	0	1	1	4	0
0	0	0	1	1	5	0
0	0	0	0	0	0	1
0	0	0	0	1	0	1
0	0	1	0	0	0	1
0	0	1	0	1	0	1

图 3-8-2 码表控制器状态机状态参考表（节选）

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 3-8-3 码表控制器状态机状态 excel 表（节选）

(2) 电路图

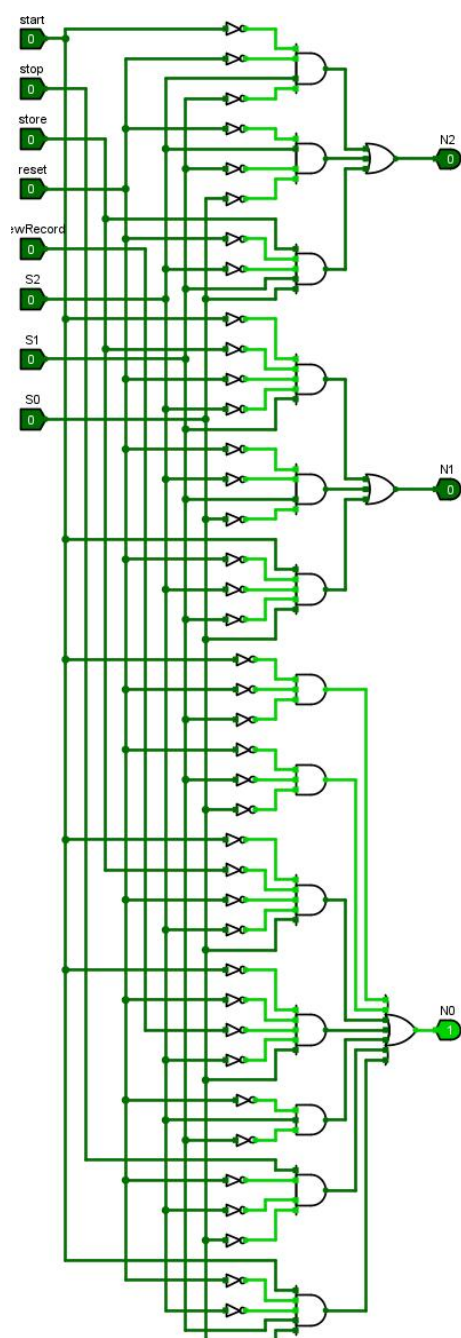


图 3-8-4 码表控制器状态机电路图

### (3) 测试图

Cnt	start	stop	store	reset	newrecPS	NS
00	0	0	0	1	0	0
01	0	0	0	1	0	1
02	0	0	0	1	0	2
03	0	0	0	1	0	3
04	0	0	0	1	0	4
05	0	0	0	1	0	5
06	0	0	0	1	1	0
07	0	0	0	1	1	1
08	0	0	0	1	1	2
09	0	0	0	1	1	3
0a	0	0	0	1	1	4

图 3-8-5 码表控制器状态机测试图

### (4) 测试分析

在 Logisim 之中没有相关测试的部分，在此直接通过头歌平台来进行测试。测试结果通过。根据测试结果可知，该电路成功实现了码表计时器需要的各种状态转换：按下 start 开始计时，按下 stop 停止计时，此时如果按下 store，打破记录则储存记录并显示，没有打破记录就显示最好记录。如果按下 start 则直接返回开始计时的状态。而在任何状态下，按下 reset 则直接返回最初状态。以上说明设计成功。

## 3.9 码表控制器输出函数设计

### (1) 设计思路及设计过程

本题可根据真值表填写 excel 表格自动生成逻辑表达式，然后再使用 Logisim 的分析组合逻辑电路功能使用表达式生成电路即可。

当前状态(现态)					输入信号										
S3	S2	S1	S0	现态 10进制	start	stop	store	reset	NewRecord	...	SDsel	SDen	DPsel	TMen	TMreset
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
0	1	1	0	6							1	0	1	1	0
0	1	1	1	7							0	0	0	0	0

图 3-9-1 码表控制器输出函数 excel 真值表



(2) 电路图

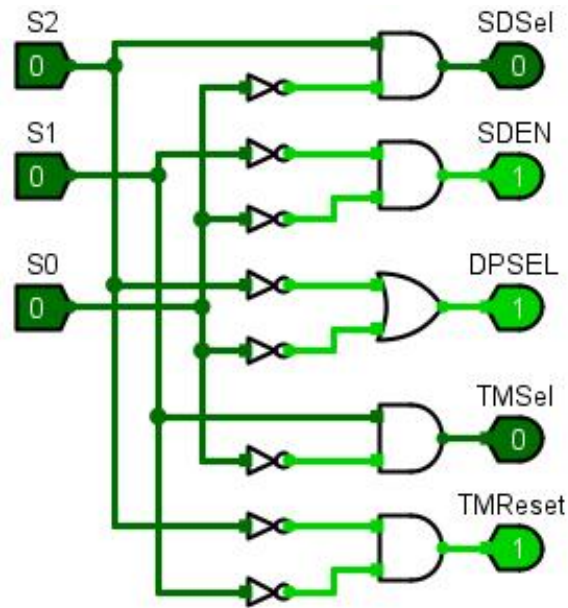


图 3-9-2 码表控制器输出函数电路图

(3) 测试图

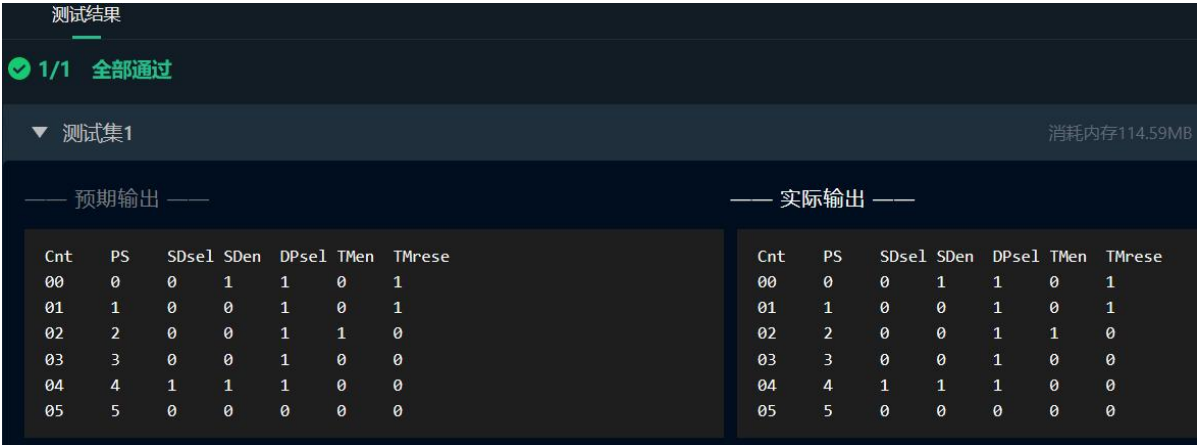


图 3-9-3 码表控制器输出函数测试图

(4) 测试分析

在 Logisim 之中没有相关测试的部分，在此直接通过头歌平台来进行测试。测试结果通过，说明设计成功。



### 3.10 码表控制器

#### (1) 设计思路及设计过程

参考状态转换表以及电路引脚说明(如下图), 由 start、stop、store、reset、NewRecord 五个状态控制的输入和三个状态变量 S0、S1、S2 输入来控制状态变量 S0、S1、S2 输出。由此将输出的三个次态 S0、S1、S2 和五个状态控制输入一起连接到状态转换器的八个输入中, 之后将状态转换器的 3 个输出与 3 个 D 触发器连接。并将 CLK 输入端与 D 触发器时钟端相连, 将 D 触发器输出连接到输出函数器, 输出函数器输出得到的结果再与对应输出端相连。这就完成了码表控制器的设计。

信号	I/O	位宽	说明
CLK	输入	1	时钟脉冲
start	输入	1	开始计时信号
stop	输入	1	停止计时信号
store	输入	1	存储计时记录信号
reset	输入	1	计时复位信号, 记录恢复为99.99
newrecord	输入	1	新的最好成绩记录信号
SDsel	输出	1	最好成绩记录的选择信号
SDen	输出	1	保存最好成绩记录的寄存器的使能信号
DPsel	输出	1	显示计时成绩记录的选择信号
TMen	输出	1	码表计数器使能信号
TMreset	输出	1	码表计数器复位信号

图 3-10-1 码表控制器电路引脚说明

#### (2) 电路图

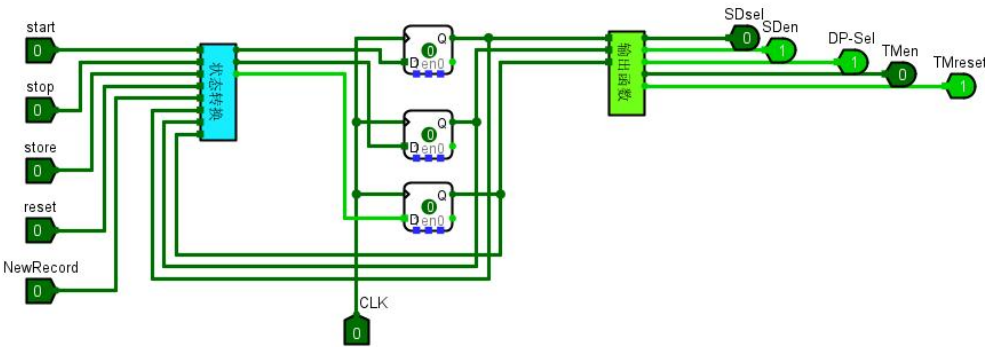


图 3-10-2 码表控制器电路图

### (3) 测试图

测试结果										
1/1 全部通过										
测试集1										
消耗内存113.02MB 代码执行时长: 2.64秒										
— 预期输出 —										
Cnt	start	stop	store	reset	NewRecSDsel	SDen	DPsel	TMen	TMrese	
00	0	0	0	1	0	0	1	0	1	
01	0	0	0	0	0	0	1	0	1	
02	0	0	0	1	0	0	1	0	1	
03	0	0	0	0	0	0	1	0	1	
04	1	0	0	0	0	0	1	0	1	
05	0	0	0	1	0	0	1	1	0	
06	0	0	0	0	0	1	1	0	1	
— 实际输出 —										
Cnt	start	stop	store	reset	NewRecSDsel	SDen	DPsel	TMen	TMrese	
00	0	0	0	1	0	0	1	0	1	
01	0	0	0	0	0	0	1	0	1	
02	0	0	0	1	0	0	1	0	1	
03	0	0	0	0	0	0	1	0	1	
04	1	0	0	0	0	0	1	0	1	
05	0	0	0	1	0	0	1	1	0	
06	0	0	0	0	0	1	1	0	1	

图 3-10-3 码表控制器测试图

### (4) 测试分析

在头歌平台的测试均通过，说明设计成功。

## 3.11 运动码表

### (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，计时器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，则更新最好历史记录，再一条分支经过二路选择器，进入寄存器保存，再进入二路选择器与最好成绩比较，最终将最好成绩送入码表显示器显示，完成设计。

### (2) 电路图

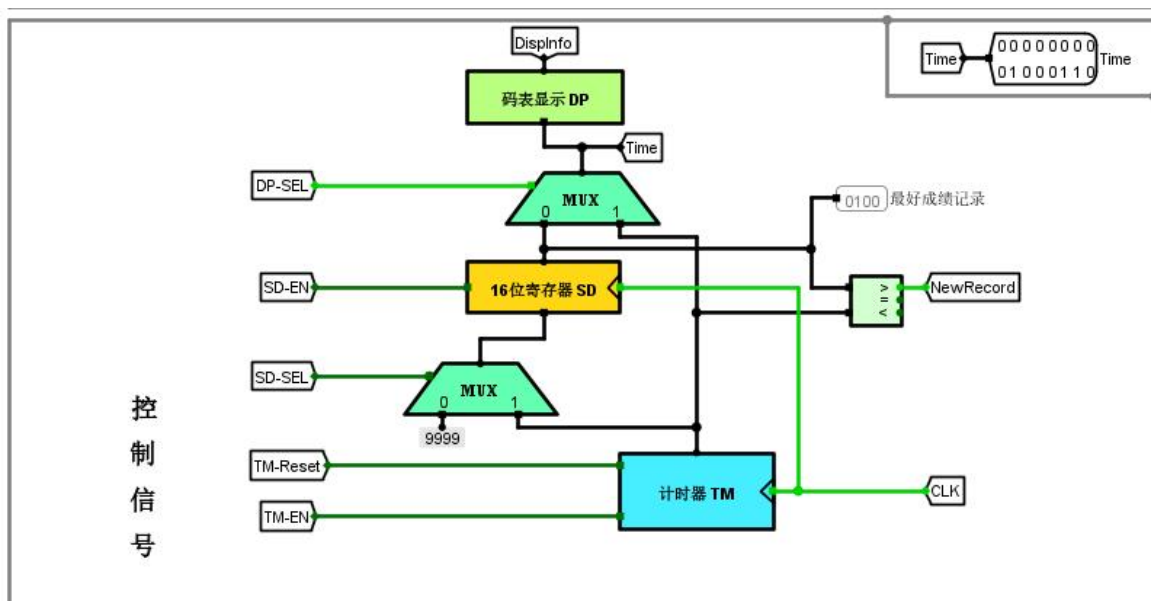


图 3-11-1 运动码表电路图

### (3) 测试图

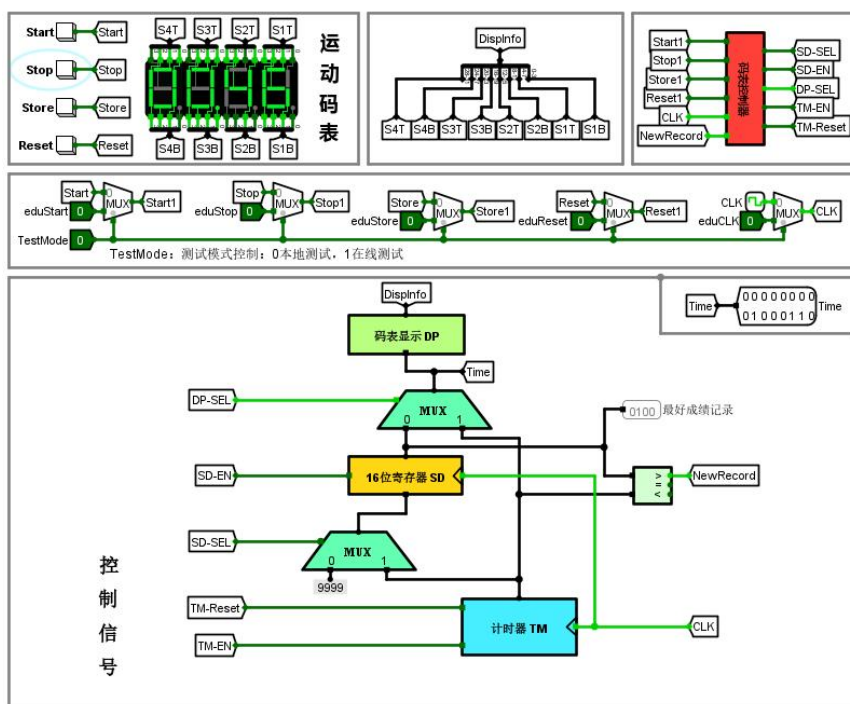


图 3-11-2 运动码表测试图

### (4) 测试分析

直接设置时钟连续，控制按钮进行测试分析，看到时间，最佳成绩，码表状态与码表显示等都没有错误，且通过头歌平台测试，说明设计成功。

---

---

## 4 设计总结与心得

### 4.1 实验总结

本实验综合性较强，整体任务是在 Logisim 上设计实现一个运动码表系统。具体来说，是根据本实验提供的完整的数字逻辑实验包，由简到难，层次递进，从器件到部件，从部件到系统，从简单的组合逻辑电路到复杂时序逻辑电路，对局部电路逐个完成设计和连接，最终集成实现为运动码表系统。

通过本实验的设计、仿真、验证循环反复的过程，我更好地理解了数字电路与逻辑设计这门课的理论知识并在实验中融会贯通，进而初步掌握小型数字电路系统的设计、仿真、调试方法。

#### 4.1.1 遇到的问题及处理

实验中，当做到码表控制器的关卡时我卡住了。之前的关卡独立看还没有出现什么大问题，之后的关卡却涉及整个码表的构建和连接。由于之前没有仔细研究码表总体构造就开始闯关，导致我做到最后几关的时候有点迷糊。不过在小组合作形式的帮助下，通过听取别的小组的分享和组内讨论，我对码表的设计有了新的认识，也重新梳理了自己的思路，最后成功完成了实验。

还有一个问题是 Logisim 的自动测试问题，前面几关的自动测试都没有问题，但是最后运动码表的集成电路那块自动测试不成功，码表显示器是乱码而不是正确的数字，然后点击按钮得到的反应不太正确，可是电路设计找不出问题，平台测试也通过了，周围人也是这种情况。反复检查码表电路和状态机设计电路都没有发现漏洞，最后发现是测试电路码表显示器的接线连接错了，研究了好久才发现，修改后果然能正确显示了。然后按钮不能完全正常地控制码表是我自己的测试操作有问题。因为时钟周期设置得太慢了，而按下按钮后太快放开，导致按钮输入改变时没有经历上升沿，所以无法触发 D 触发器改变，就无法正确执行功能。将时钟周期调到正常速度，并增加按下按钮的时间，就能保证码表正常工作了。

---

### 4.1.2 设计方案存在的不足

如前面所述的问题所示，码表功能是否正常实现与时钟周期和按钮时间有关，若按钮输入的改变不经历上升沿，则无法控制码表作出正确的反应。这是我认为设计方案存在的一个漏洞，如果坚持使用 D 触发器，目前我没有想到合适的设计方案能解决这个问题。

另外，有几个电路我使用了真值表和逻辑表达式生成电路，在输入输出情况比较多的情况下，其实是比较繁琐的，生成的电路也比较复杂，而且对于无关状态的分析也不方便。不过也比人工连线要方便许多。个别电路简单的其实可以自己设计电路，会比自动生成的电路更简洁。

## 4.2 实验心得

通过本次实验，我对与 Logisim 的使用有了更加深刻的理解。而我的动手能力与实操能力也在这里实验之中得到了锻炼。在积极完成电路图的同时，我对于书本上教授的课堂知识也有了更加深刻的理解。实验整体的过程有一个心情变化的过程，一开始没有什么感觉，做到后面几关卡住时很难受，但在做完并重新梳理设计思路写报告的过程对我来说很快乐，仿真测试的成功能让我体会到成就感，对知识的理解加深也让人舒心。这次码表实验体制完整，难度适宜，小组讨论展示的形式也很有帮助，在能锻炼我们能力的同时也让我们保持学习的兴趣。

在实验电路的设计中，更重要的是整体系统的设计思考，一开始由于没有从整体的设计思路出发构想，只是单纯为了通关而通关，没有总体的把握和构思，导致后几关卡住，一头雾水。

所幸课程提供了小组讨论和展示这一环节。听了其他组同学的精彩演说之后我才意识到自己缺少对整体电路的思考和研究，并且一些优秀同学对某个具体的逻辑电路的讲解也让人醍醐灌顶，之前一头雾水的点一下就清晰了。当然，之后和组员们的讨论和交流更给了我很大的帮助。一开始只是按部就班地完成独立的关卡，却没有对各个模块之间的联系以及整体布局作分析。第二次课才开始思考每个模块的功能和协作，原本不理解的题目和电路也一下子清晰起来了。可见，在电路设计中，一定要有目的性，要有很细的框架设计和全局思想，只有搞清楚码表是由哪些模块组装而成并且熟悉每个模块的设计才能进一步地具体实现最终的电路。

---

---

另外，整体的分析不仅是对整个电路而言，对局部电路也需要整体分析。如果直接先从自己明白的地方出发而不先分析电路的话，会发现会出现连接了电路之后还得删掉重来的情况。其次，在连接电路的时候，要给线路留够足够的空间，要调整好各个元件的位置。否则也得重来。

最重要的是，写报告的过程也让我重新整理了思路，让我得以重新审视自己的设计过程并修改和完善一些不足之处。而实验报告整体严格细致的要求也正和具体的电路设计一样，提醒我们做事一定要有全局的规划和全面的考虑，并且要注重细节，不得马虎。

### 4.3 意见与建议

建议在第一节的时候安排对任务的具体讲解，引导同学们先对整个系统进行整体分析再逐个设计电路，同时对仿真过程给予提示说明。或者在第一节就给出实验报告的任务要求文字版本，让同学们清楚实验要求。

---

---

## 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：**

---

---

## 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-运动码表实验报告

全班收齐后统一打包压缩交给老师。