華中科技大學

2021

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：计科 2011 班

学　　号：U202015084

姓　　名：张文浩

电　　话：13792295868

邮　　件：1457337635@qq.com

完成日期：2021.12.7

目　录

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

根据十进制数和七段数码管的对应关系和七段数码管的引脚顺序，设计出由十进制数的 4 位 BCD 码（输入）到数码管点亮状态（输出）的真值表，利用真值表直接生成符合要求的电路。

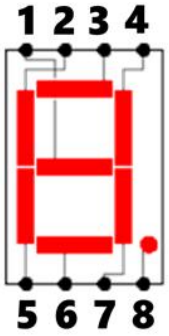


图 2.1 七段数码管引脚顺序

X3	X2	X1	X0	Seg_1	Seg_2	Seg_3	Seg_4	Seg_5	Seg_6	Seg_7
0	0	0	0	0	1	1	1	1	1	1
0	0	0	1	0	0	0	1	0	0	1
0	0	1	0	1	0	1	1	1	1	0
0	0	1	1	1	0	1	1	0	1	1
0	1	0	0	1	1	0	1	0	0	1
0	1	0	1	1	1	1	0	0	1	1
0	1	1	0	1	1	1	0	1	1	1
0	1	1	1	0	0	1	1	0	0	1
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	0	1	1	1	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	0	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	0	1	1
1	1	1	0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	1	1

图 2.2　七段数码管驱动电路真值表

七段数码管的引脚顺序和电路真值表如图 2.1，2.2 所示。

（2）电路图

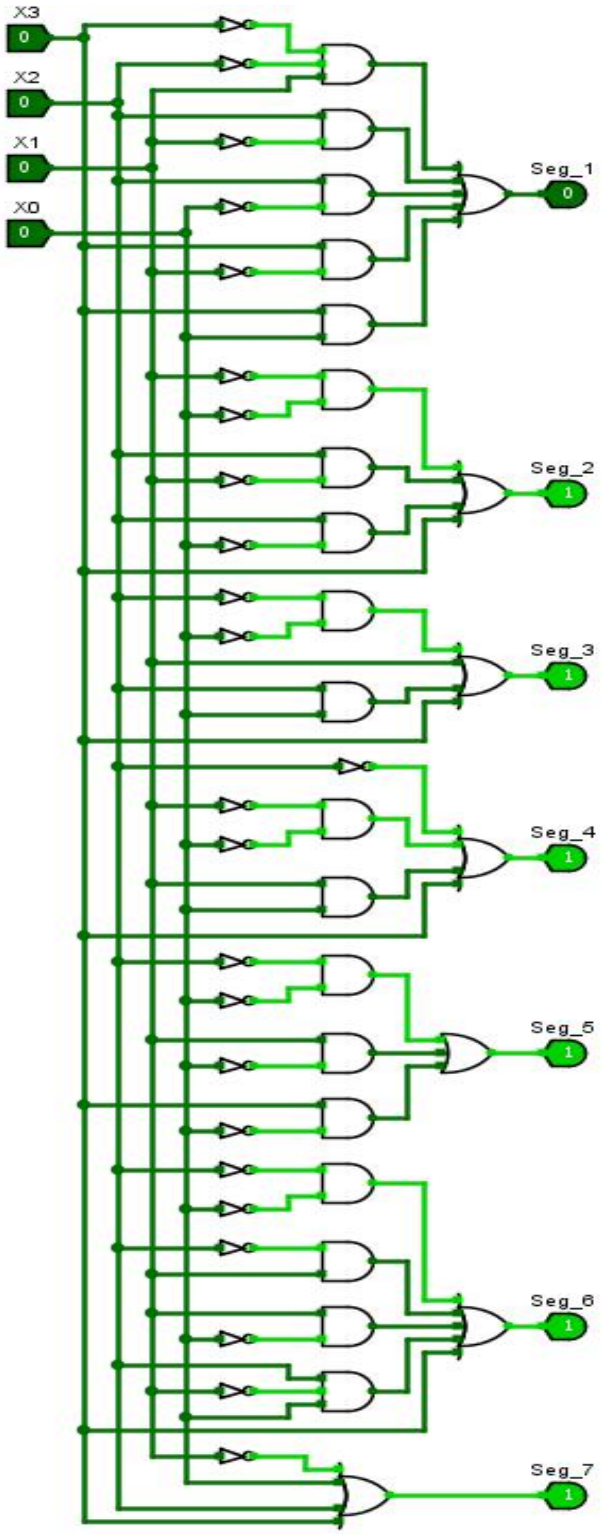


图 2.3　7 段数码管驱动电路

设计的电路如图 2.3 所示。

（3）测试图

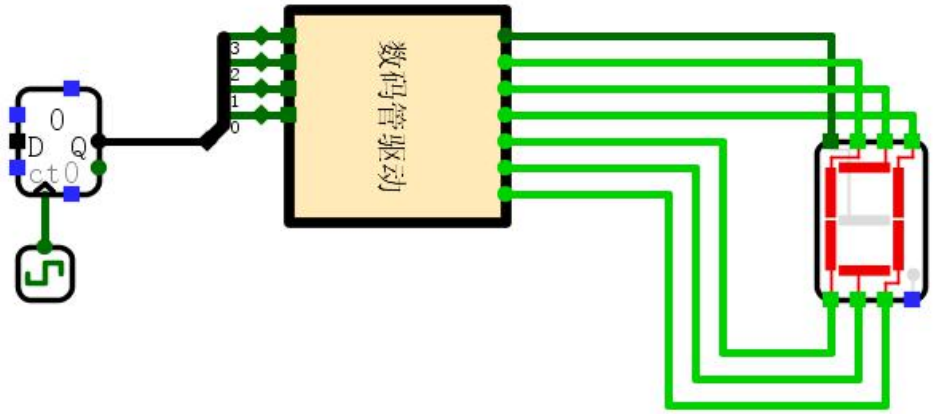


图 2.4　7 段数码管驱动电路测试电路

测试电路如图 2.4 所示。

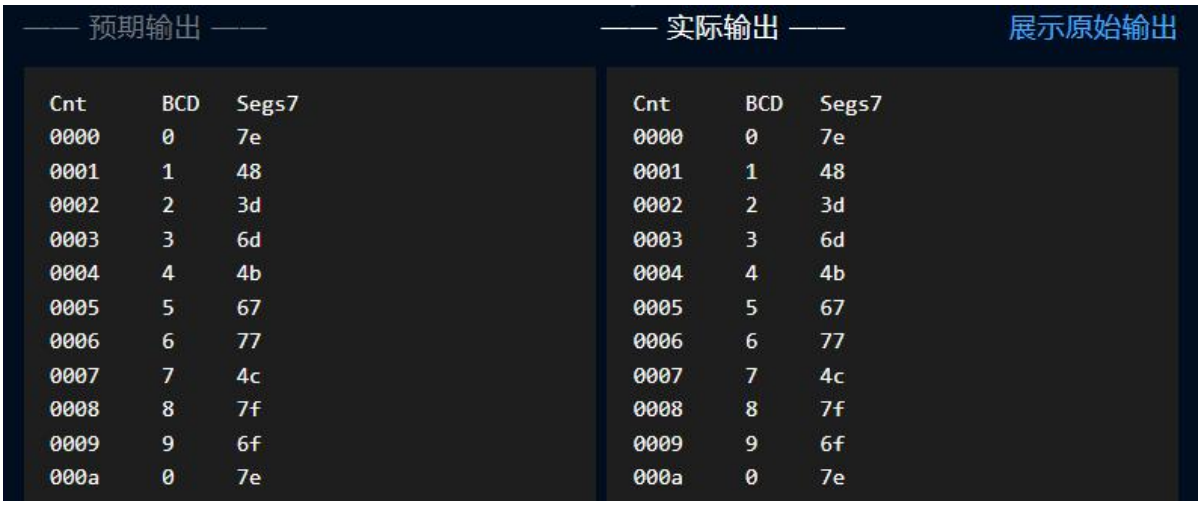


图 2.5　7 段数码管驱动电路测试样例

测试样例如图 2.5 所示。

（4）测试分析

由测试结果可知，该电路能在数码管上正确显示数字，电路输出正确，所设计的电路符合要求。

2.2 2 选 1 选择器设计（1 位）

（1）设计思路及设计过程

设输入为 X0，X1，Sel，当 Sel=0 时，输出 X0，当 Sel 为 1 时，输出 X1；根据逻辑功能，可以直接写出输出函数的表达式为 Sel X1 + X0 ~Sel。

（2）电路图

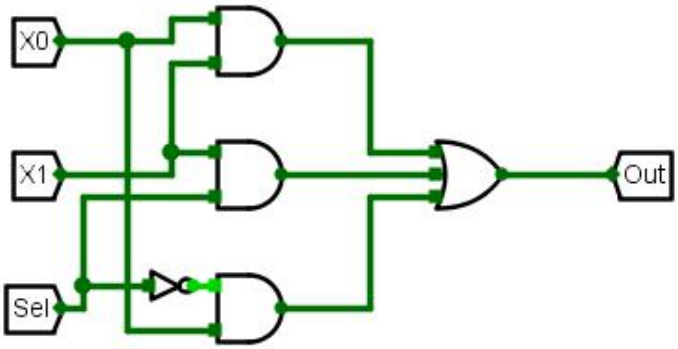


图 2.6　2 选 1 选择器（1 位）电路

设计的电路如图 2.6 所示。

（3）测试图

预期输出　　实际输出　　展示原始输出

Cnt	X0	X1	Sel	Out
0000	0	0	0	0
0001	1	0	0	1
0002	0	1	0	0
0003	1	1	0	1
0004	0	0	1	0
0005	1	0	1	0
0006	0	1	1	1
0007	1	1	1	1
0008	0	0	0	0

Cnt	X0	X1	Sel	Out
0000	0	0	0	0
0001	1	0	0	1
0002	0	1	0	0
0003	1	1	0	1
0004	0	0	1	0
0005	1	0	1	0
0006	0	1	1	1
0007	1	1	1	1
0008	0	0	0	0

图 2.7　2 选 1 选择器（1 位）测试样例

测试样例如图 2.7 所示。

（4）测试分析

由测试结果可知，该电路可以按要求正确实现当 Sel=0 时，输出 X0，当 Sel 为 1 时，输出 X1 的选择功能，所设计的电路符合要求。

2.3 2 选 1 选择器设计（16 位）

（1）设计思路及设计过程

两个 16 位二进制数的二选一问题，当 Sel=0 时输出 X，当 Sel=1 时输出 Y，可以采用 16 个一位二进制数的二选一电路进行电路并发。可以将两个 16 位数按位对齐形成 16 组，每一组都通过逻辑功能一致的二选一选择器，再将 16 个输出合并成一个 16 位数，即实现了两个 16 位数的选择功能。

（2）电路图

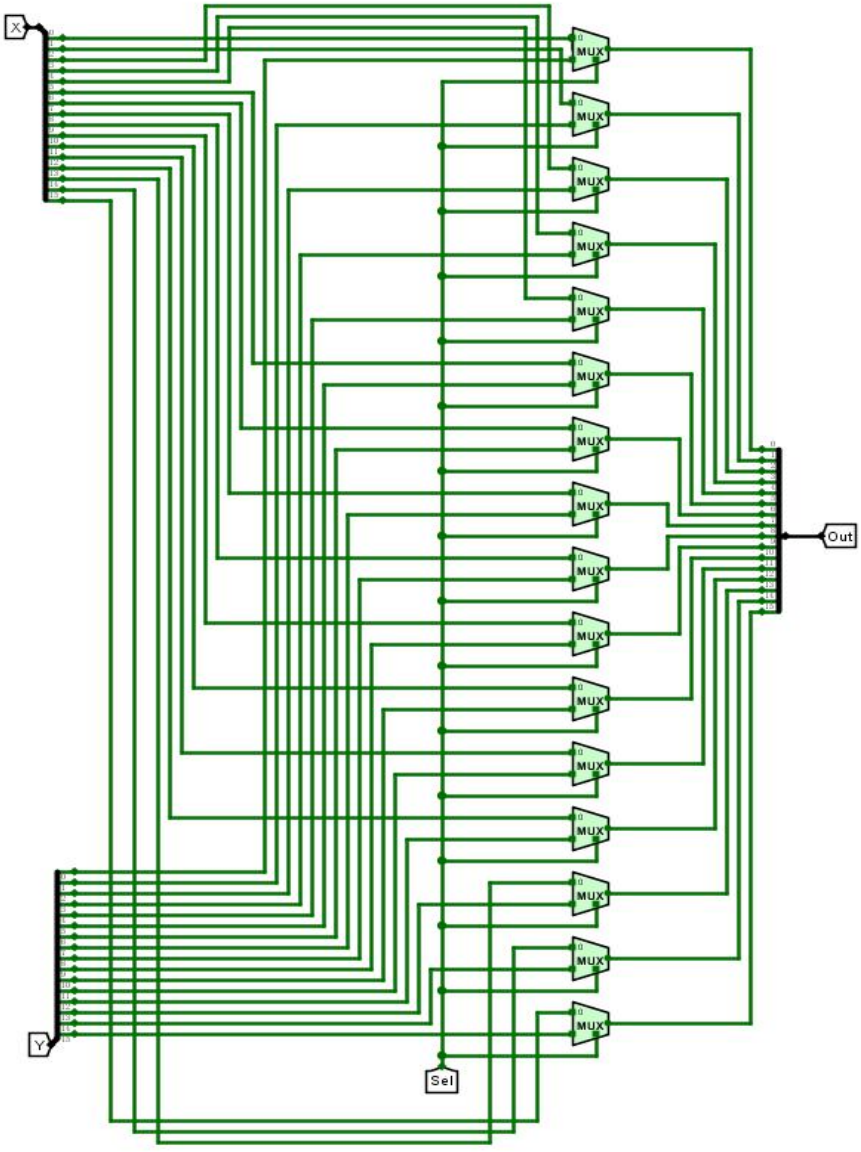


图 2.8　2 选 1 选择器（16 位）电路

设计的电路如图 2.8 所示。

（4）测试图

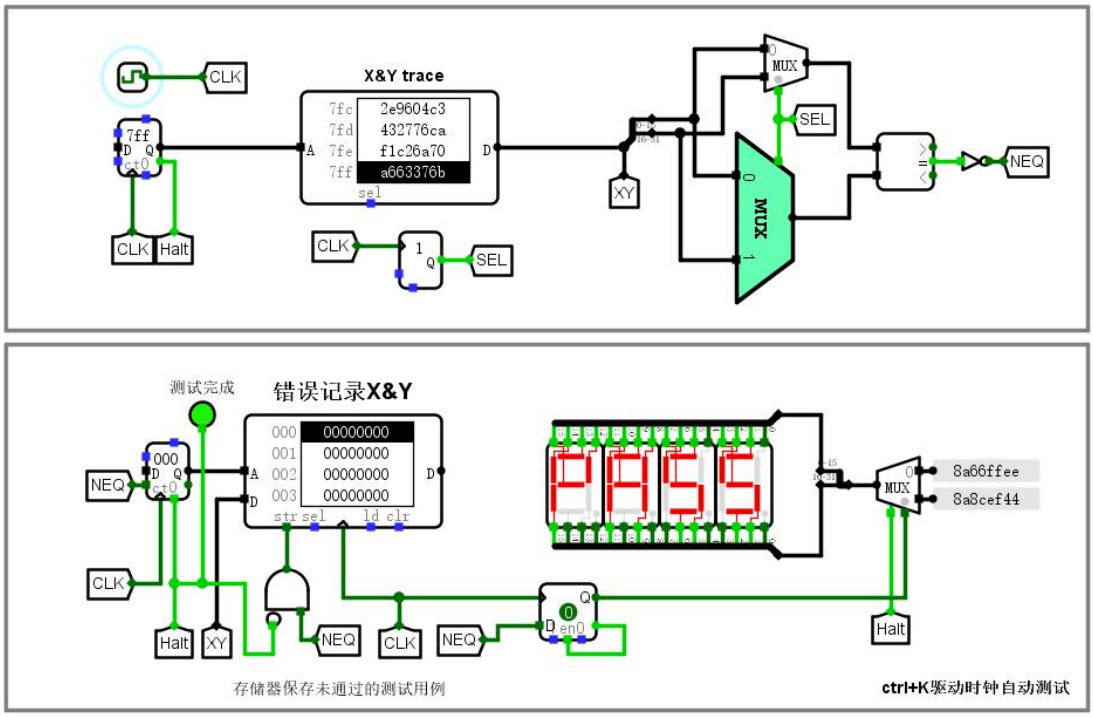


图 2.9　2 选 1 选择器（16 位）测试电路

测试电路如图 2.9 所示。

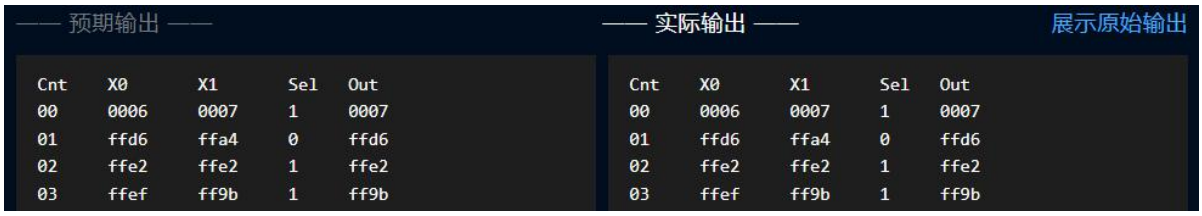


图 2.10　2 选 1 选择器（16 位）部分测试样例

部分测试样例如图 2.10 所示

（4）测试分析

由测试结果可知，该电路可以按要求正确实现两个 16 位二进制数 X0,X1 的选择功能，当 Sel=0 时输出 X0，当 Sel=1 时输出 X1，所设计的电路符合要求。

2.4 无符号比较器设计（16 位）

（1）设计思路及设计过程

首先构建 4 位无符号比较器。输入为四位无符号数 X 和四位无符号数 Y，输出为 Great、Less、Equal。输出和输入之间的关系如下。 X＞Y：Great=1 Less=0 Equal=0 X＜Y：Great=0 Less=1 Equal=0 X＝Y：Great=0 Less=0 Equal=1，下面构建输出函数：

1 位比较 X : Y

Great = X ~ Y(输入为 0)

Less = ~ X Y(输入为 01)

Equal = ~ (X ^ Y)=(~ X ~ Y + X Y) (输入为 00 或者 11，即输入相同)

2 位比较 X=X1X0 : Y=Y1Y0

Great = X1 ~ Y1 + ~ (X1 ^ Y1)X0~ Y0 (X 的最高位 X1 大于 Y 的最高位 Y1，或者最高位相等时 X 的低位 X0 大于 Y 的低位 Y0)

Less = ~ X1 Y1 + ~ (X1^ Y1)~ X0 Y0 (X 的最高位 X1 小于 Y 的最高位 Y1，或者最高位相等时 X 的低位 X0 小于 Y 的低位 Y0)

Equal = ~ (X1 ^ Y1) ~ (X0 ^ Y0) (X 的最高位 X1 等于 Y 的最高位 Y1，而且，X 的低位 X0 等于 Y 的低位 Y0)

3 位、4 位比较依此类推。

最后可以得到 4 位无符号比较器的输出函数为：

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

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

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

然后利用 4 位无符号比较器构建 16 位无符号比较器。

将两个 16 位数每 4 位分成一组，分别连接 4 个 4 位比较器，每一个比较器的 Great 和 Equal（x，y 相等或者 x>y 时输出 1），Less 和 Equal 都通过异或门（x，y 相等或者 x<y 时输出 1），共有 4 个 G-E 异或输出（1 个 4 位二进制数）和 4 个 L-E 异或输出（1 个 4 位二进制数），若 x>y,则一定有第一个 4 位二进制数大于第二个 4 位二进制数。将这 8 个输出再次通过一个 4 位比较器，得到最后的比较结果。

（2）电路图

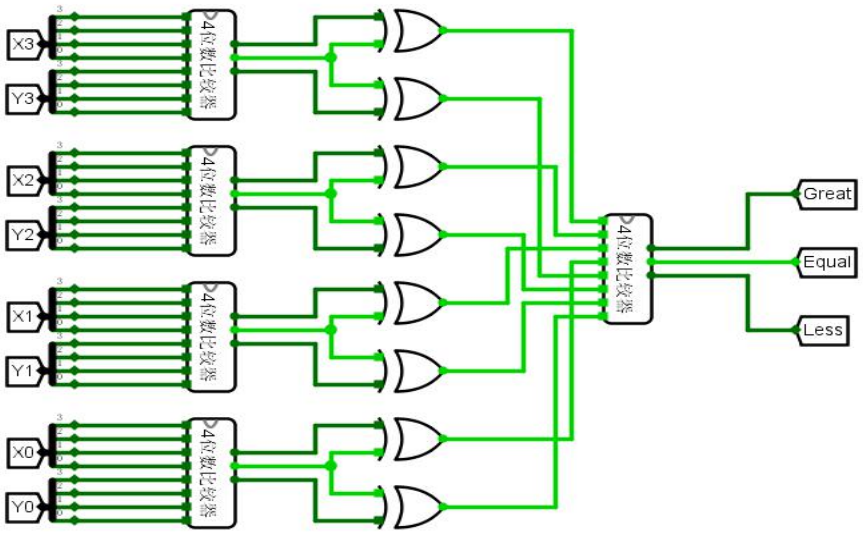


图 2.11　无符号比较器设计（16 位）电路

设计的电路如图 2.11 所示。

（3）测试图

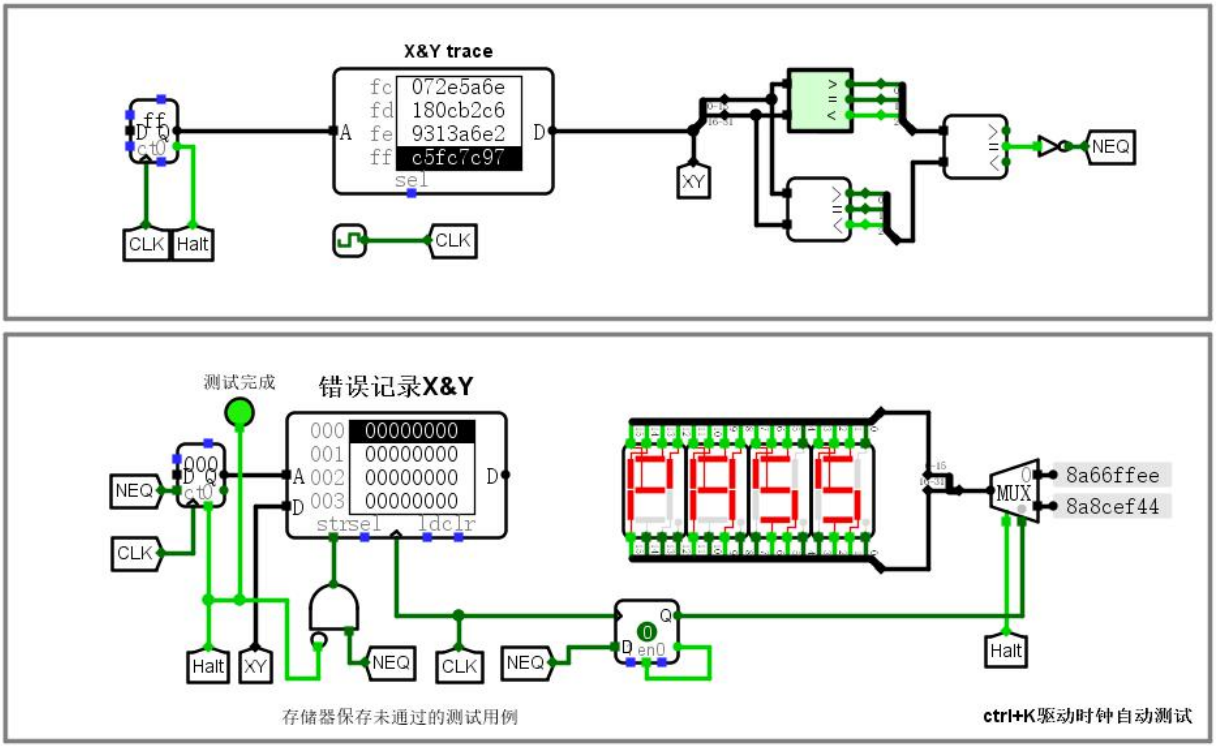


图 2.12　无符号比较器（16 位）测试电路

测试电路如图 2.12 所示。

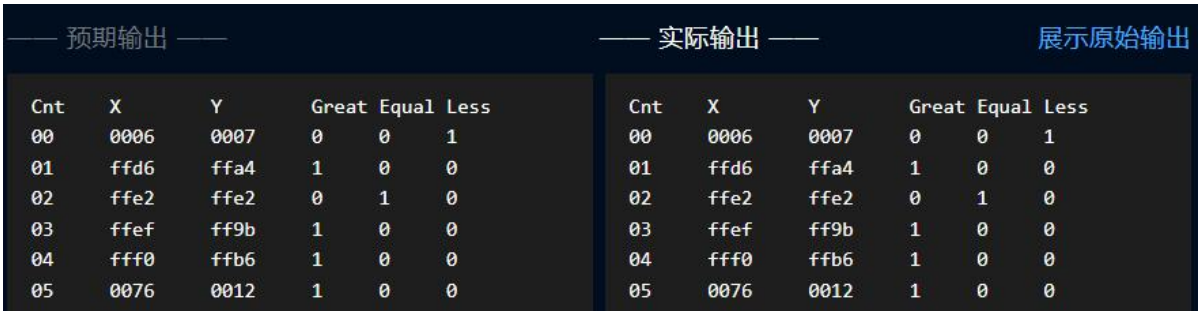


图 2.13　无符号比较器（16 为）部分测试样例

部分测试样例如图 2.13 所示。

（4）测试分析

由测试结果可知，该电路可以按要求正确实现两个无符号 16 位二进制数的比较功能，所设计的电路符合要求。

2.5 并行加载寄存器(4 位)

（1）设计思路及设计过程

寄存器（Register）的功能是存储二进制代码，它是由具有存储功能的触发器组合起来构成的。一个触发器可以存储 1 位二进制代码，故存放 n 位二进制代码的寄存器，需用 n 个触发器来构成。利用 4 个 D 触发器，将 4 位二进制数的每一位（现态）都输入触发器的 D，因为 D 触发器的次态等于 D，所以当时钟脉冲出现上升沿时，次态就会变成 D 的值，即把原来的现态变成了次态，实现“寄存”功能。

（2）电路图

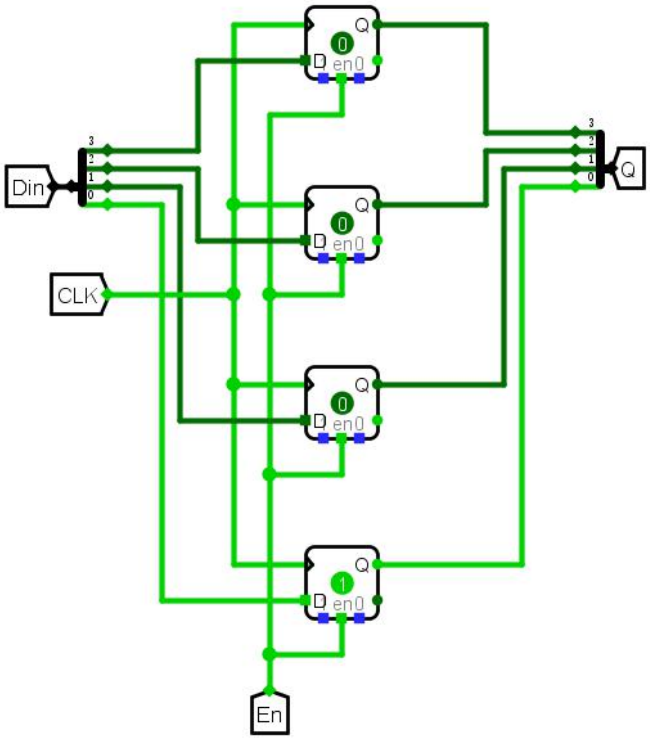


图 2.14　并行加载寄存器(4 位)电路

设计的电路如图 2.14 所示。

（3）测试图

Cnt	Din	En	Q
00	0	0	0
01	1	0	0
02	2	1	0
03	3	1	2
04	4	1	3
05	5	0	4
06	6	1	4
07	7	0	6
08	8	0	6

Cnt	Din	En	Q
00	0	0	0
01	1	0	0
02	2	1	0
03	3	1	2
04	4	1	3
05	5	0	4
06	6	1	4
07	7	0	6
08	8	0	6

图 2.15　并行加载寄存器（4 位）部分测试样例

部分测试样例如图 2.15 所示。

（4）测试分析

由测试结果可知，该电路可以按要求正确实现 4 位二进制数的寄存功能，当时钟脉冲出现上升沿时，寄存在 D 端的数据输出，所设计的电路符合要求。

2.6 并行加载寄存器(16 位)

（1）设计思路及设计过程

将 16 位二进制数分成 4 组，每组分别通过 4 位并行加载寄存器，将 4 组的输出组成一个 16 位二进制数，即实现了 16 位二进制数的寄存功能。

（2）电路图

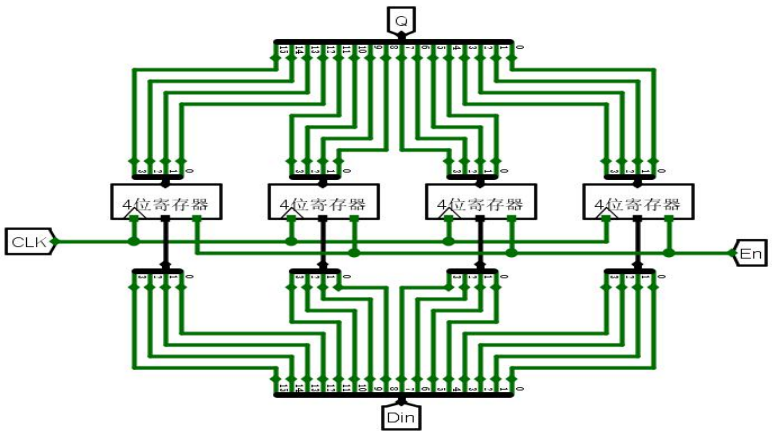


图 2.16　并行加载寄存器（16 位）电路

设计的电路如图 2.16 所示。

（3）测试图

Cnt	Din	En	Q
00	d7dc	1	0000
01	727a	0	d7dc
02	1518	0	d7dc
03	9cf9	1	d7dc
04	a07e	1	9cf9
05	9eaa	0	a07e
06	7176	0	a07e
07	e99b	1	a07e
08	0acc	0	e99b
09	dc7c	1	e99b
0a	df9b	1	dc7c
0b	0fcd	1	df9b
0c	d798	0	0fcd
0d	ab60	1	0fcd

Cnt	Din	En	Q
00	d7dc	1	0000
01	727a	0	d7dc
02	1518	0	d7dc
03	9cf9	1	d7dc
04	a07e	1	9cf9
05	9eaa	0	a07e
06	7176	0	a07e
07	e99b	1	a07e
08	0acc	0	e99b
09	dc7c	1	e99b
0a	df9b	1	dc7c
0b	0fcd	1	df9b
0c	d798	0	0fcd
0d	ab60	1	0fcd

图 2.17　并行加载寄存器（16 位）部分测试样例

部分测试样例如图 2.17 所示。

（4）测试分析

由测试结果可知，该电路可以按要求正确实现 16 位二进制数的寄存功能，所设计的电路符合要求。

2.7 BCD 计数器状态机设计

（1）设计思路及设计过程

计数器的目的是计数时钟脉冲，所以电路的输出(N3N2N1N0)应该等于输入(S3S2S1S0)加一，将真值表写出，按照真值表可以得到 N3N2N1N0 的表达式，根据表达式可以设计出电路。

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

图 2.18　BCD 计数器状态机真值表

（2）电路图

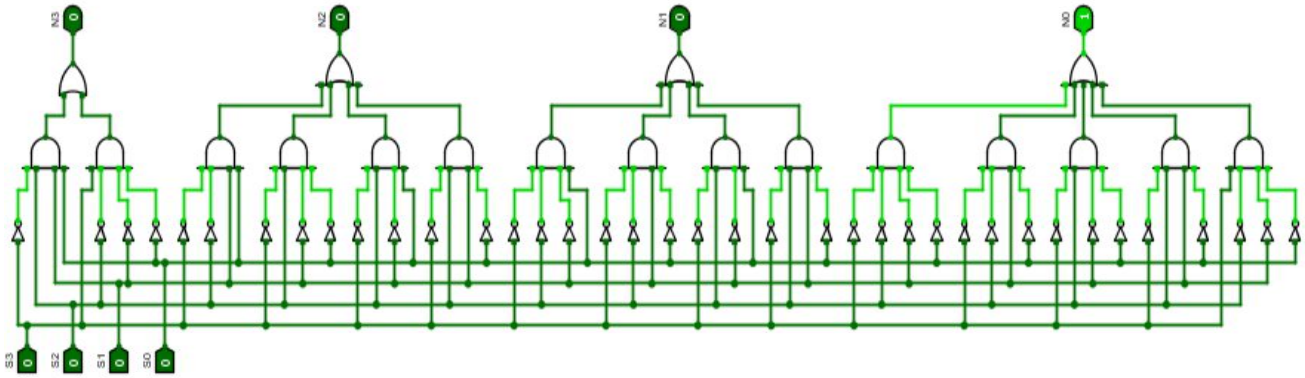


图 2.19　BCD 计数器状态机电路

设计的电路如图 2.19 所示。

（3）测试图

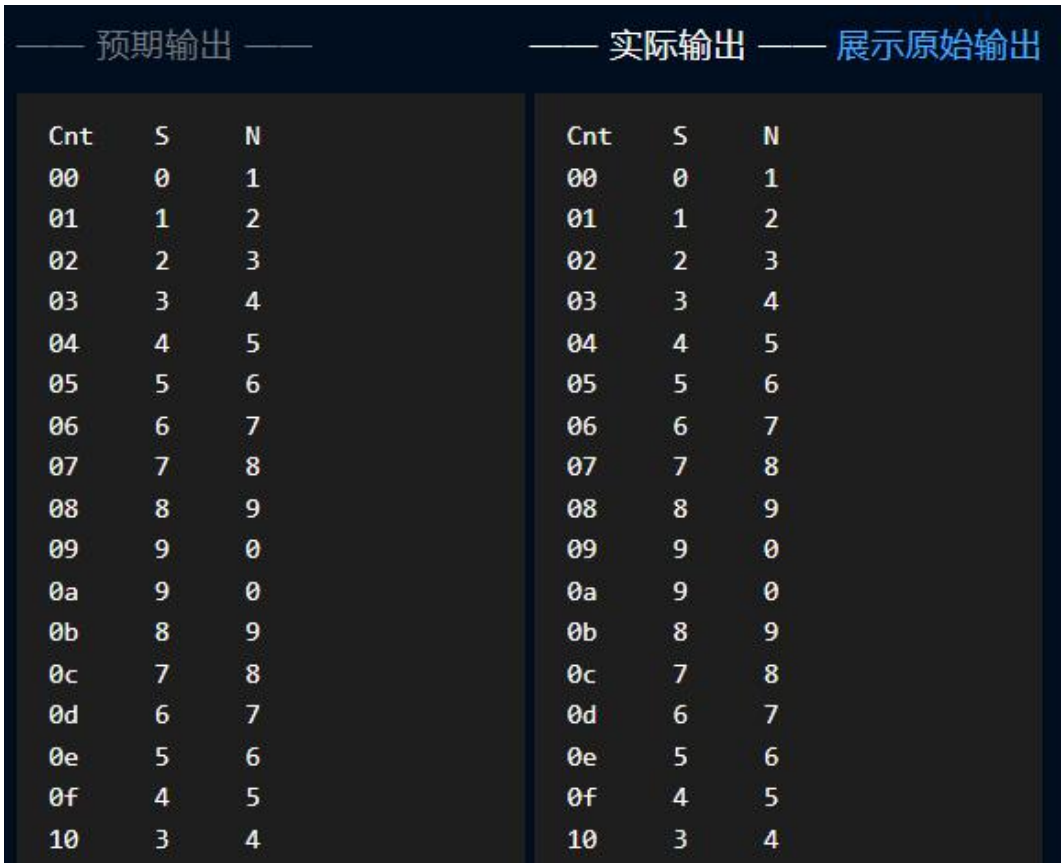


图 2.20　BCD 计数器状态机测试样例

测试样例如图 2.20 所示。

（4）测试分析

由测试结果可知，该电路可以按要求正确实现输出等于输入加一这一功能，所设计的电路符合要求。

2.8 BCD 计数器输出函数设计

(1) 设计思路及设计过程

当 S=9 时输出 1，否则输出 0，可以直接得到输出函数为：S3 ~S2 ~S1 S0，根据输出函数表达式可以设计电路。

(2) 电路图

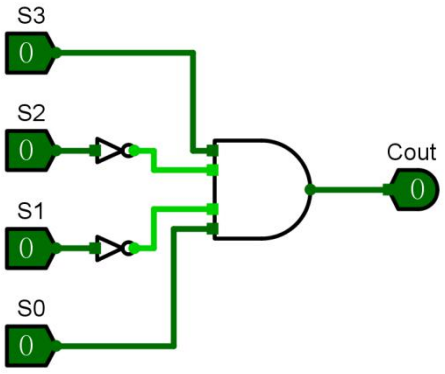


图 2.21　BCD 计数器输出函数电路

设计的电路如图 2.21 所示

(3) 测试图

图 2.22　BCD 计数器输出函数电路测试样例

测试样例如图 2.22 所示。

(4) 测试分析

由测试结果可知，该电路可以按要求正确实现当 S=9 时输出 1，否则输出 0 的功能，所设计的电路符合要求。

2.9　BCD 计数器设计（1 位十进制）

（1）设计思路及设计过程

利用 4 个 D 触发器，D 触发器的 4 个输出可以组成一个 4 位 BCD 码的输出，用来

表示计数次数，再将 4 个输出接入输出函数用来产生进位输出，同时 4 个输出接入状态转换模块，产生的输出再接入 D 触发器的 D 端，利用 D 触发器的次态等于 D，可以实现循环。

（2）电路图

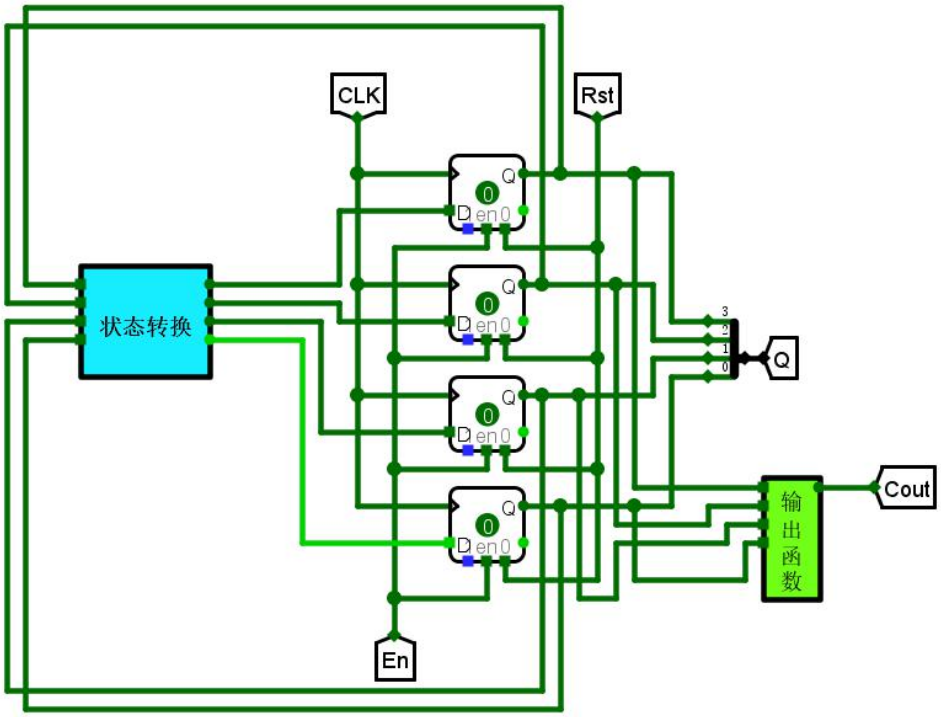
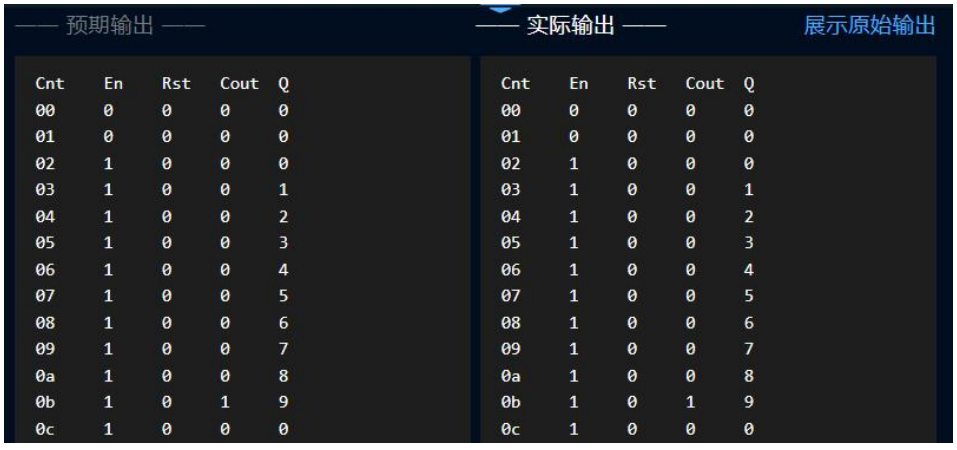


图 2.23　BCD 计数器设计（1 位十进制）电路

设计的电路如图 2.23 所示。

（2）测试图



Cnt	En	Rst	Cout	Q
00	0	0	0	0
01	0	0	0	0
02	1	0	0	0
03	1	0	0	1
04	1	0	0	2
05	1	0	0	3
06	1	0	0	4
07	1	0	0	5
08	1	0	0	6
09	1	0	0	7
0a	1	0	0	8
0b	1	0	1	9
0c	1	0	0	0

Cnt	En	Rst	Cout	Q
00	0	0	0	0
01	0	0	0	0
02	1	0	0	0
03	1	0	0	1
04	1	0	0	2
05	1	0	0	3
06	1	0	0	4
07	1	0	0	5
08	1	0	0	6
09	1	0	0	7
0a	1	0	0	8
0b	1	0	1	9
0c	1	0	0	0

图 2.24 BCD 计数器设计（1 位十进制）部分测试样例

部分测试样例如图 2.24 所示。

（4）测试分析

由测试结果可知，该电路可以按要求正确实现计数功能，所设计的电路符合要求。

2.10 码表计数器设计（4 位十进制）

（1）设计思路及设计过程

4 位十进制的码表计数器有 16 位输出，可以利用 4 个 BCD 计数器，分别对应 10 秒、1 秒、0.1 秒、0.01 秒。主要思想是，如果本计数器产生进位输出，那么下一级计数器应该加一，所以下一级计数器是否加一，取决于使能端和后面的所有计数器是否有进位输出，这样可以让 en 直接接在第一级计数器的使能端，然后前面的计数器的使能端可以由 en 和后面的计数器的进位输出的与来连接，时钟脉冲和异步复位正常接入，这样电路就可以构造完毕。

（3）电路图

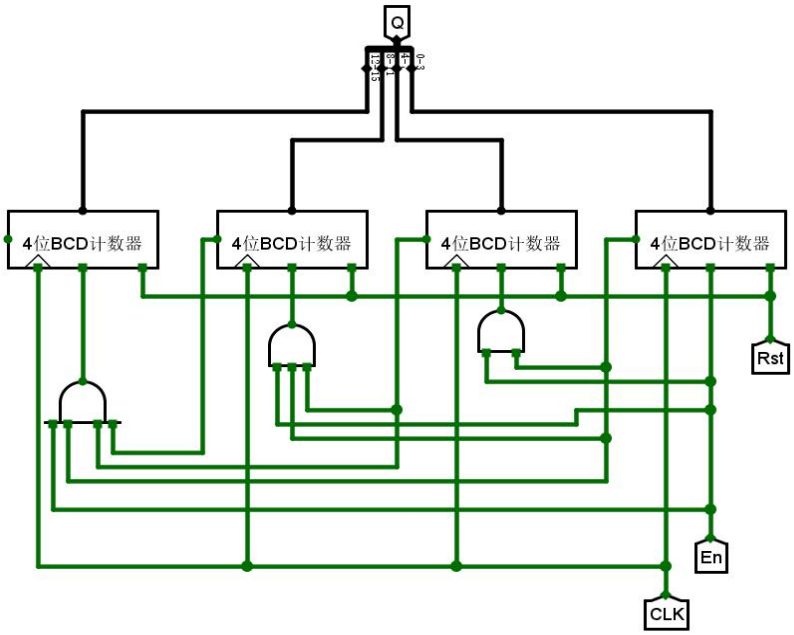


图 2.25 码表计数器设计（4 位十进制）电路

设计的电路如图 2.25 所示。

（3）测试图

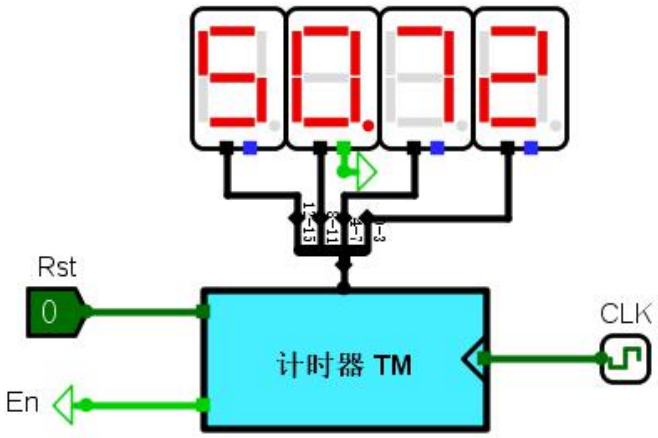
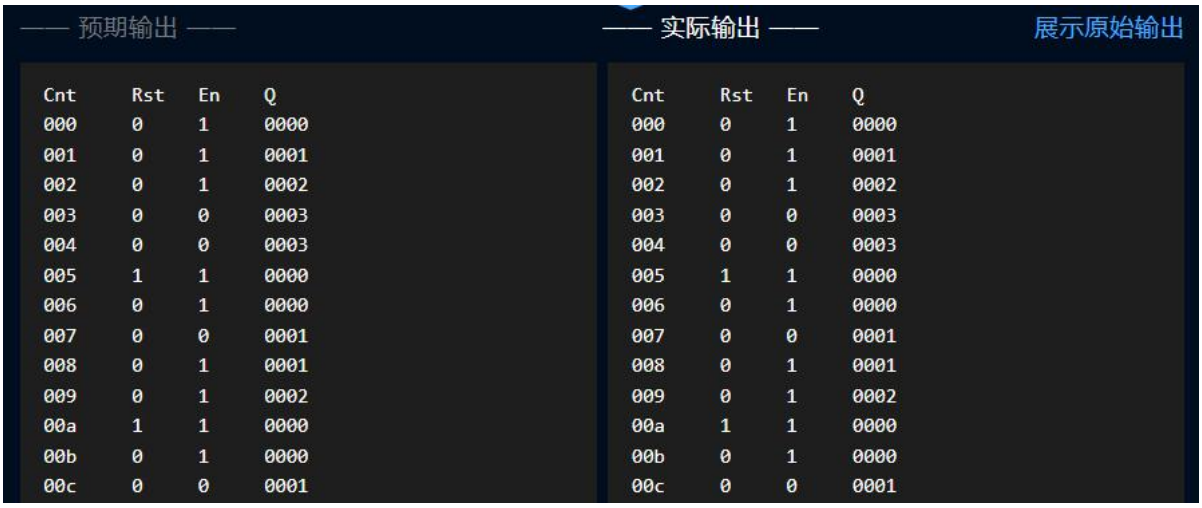


图 2.26　码表计数器设计（4 位十进制）测试电路

测试电路如图 2.26 所示。



Cnt	Rst	En	Q
000	0	1	0000
001	0	1	0001
002	0	1	0002
003	0	0	0003
004	0	0	0003
005	1	1	0000
006	0	1	0000
007	0	0	0001
008	0	1	0001
009	0	1	0002
00a	1	1	0000
00b	0	1	0000
00c	0	0	0001

Cnt	Rst	En	Q
000	0	1	0000
001	0	1	0001
002	0	1	0002
003	0	0	0003
004	0	0	0003
005	1	1	0000
006	0	1	0000
007	0	0	0001
008	0	1	0001
009	0	1	0002
00a	1	1	0000
00b	0	1	0000
00c	0	0	0001

图 2.27　码表计数器（4 位十进制）部分测试样例

部分测试样例如图 2.27 所示。

（4）测试分析

由测试结果以及测试电路的数码管输出可知，该电路可以按要求正确实现码表计数功能，所设计的电路符合要求。

2.11 码表显示驱动设计

（1）设计思路及设计过程

输入的 16 位 BCD 码每 4 个为一组，输入到数码管驱动中，对与每一个数码管驱动的 7 个输出，按正确次序连接到输出端即可，注意缺失的一位为小数点，要注意第二位数字的小数点也要显示。

（2）电路图

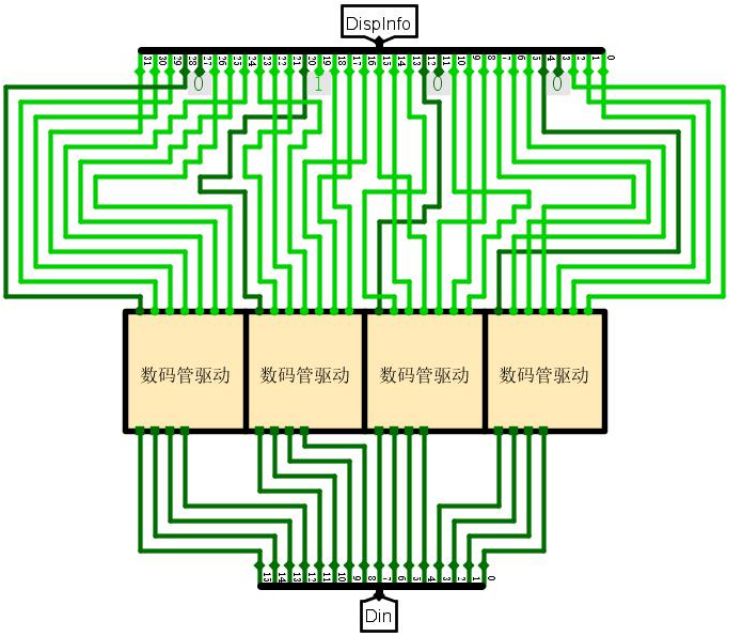


图 2.28　码表显示驱动设计电路图

设计的电路如图 2.28 所示。

（3）测试图

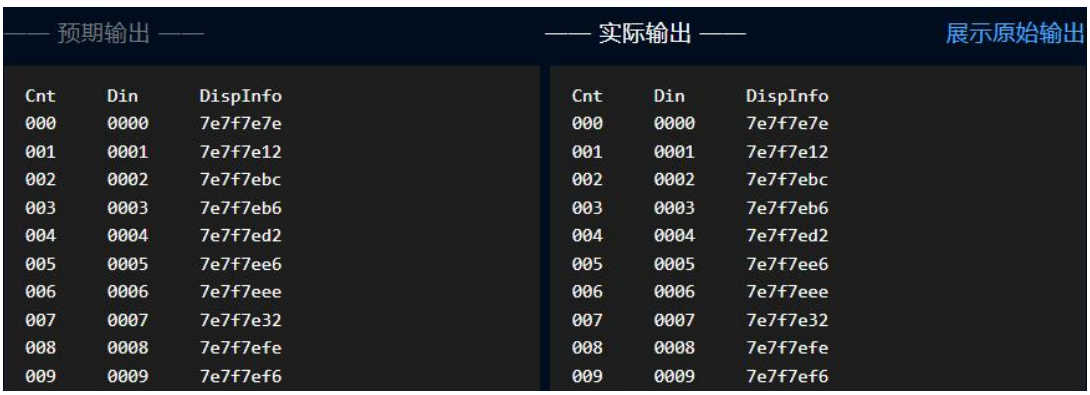


图 2.29 码表显示驱动部分测试样例

部分测试样例如图 2.29 所示。

（4）测试分析

由测试结果可知，该电路可以按要求正确实现码表显示功能，所设计的电路符合要求。

2.12 码表控制器状态机设计

（1）设计思路及设计过程

根据现态与次态之间的转换与输入之间的关系，可以画出如下的状态转移图

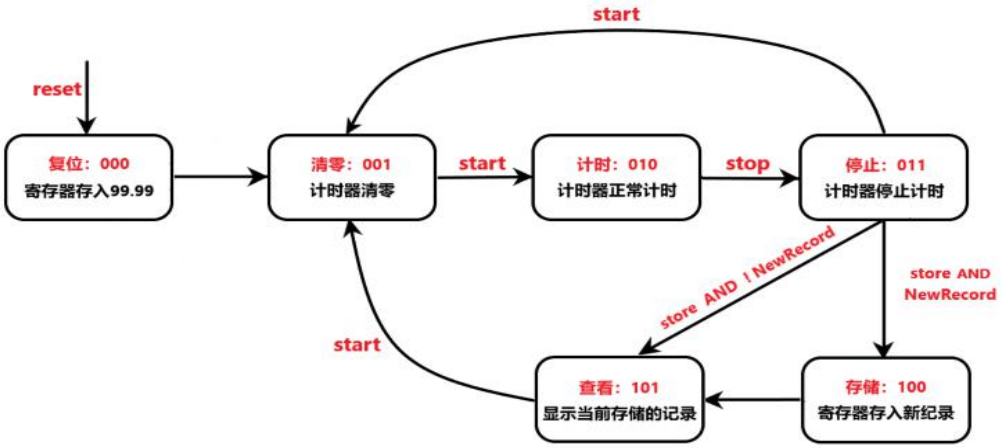


图 2.30　码表控制器状态转移图

经过补充状态转移路径之后，填好真值表，可以得到输出函数的表达式为：

N2：~start ~reset S2 ~S1 + ~reset S2 ~S1 ~S0 + ~start ~stop store ~reset ~S2 S1 S0

N1：~start ~store ~reset ~S2 S1 + ~reset ~S2 S1 ~S0 + start ~reset ~S2 ~S1 S0

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

根据输出函数表达式可以画出相应的电路图。

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	In6	In7	In8	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	1			0					2	0	0	1	0
0	0	1	0	2		1		0					3	0	0	1	1
0	0	1	1	3	0	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					1	0	0	0	1
0	0	1	1	3	1		0						1	0	0	0	1
0	0	1	1	3	0	0	1	0	0				5	0	1	0	1
0	0	0	1	1				1					0	0	0	0	0
0	0	1	0	2				1					0	0	0	0	0
0	0	1	1	3				1					0	0	0	0	0
0	1	0	0	4				1					0	0	0	0	0
0	1	0	1	5				1					0	0	0	0	0
0	0	0	0	0				1					0	0	0	0	0
0	0	0	1	1	0			0					1	0	0	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	1	0	1	5	0			0					5	0	1	0	1

图 2.31　码表控制器真值表

（2）电路图

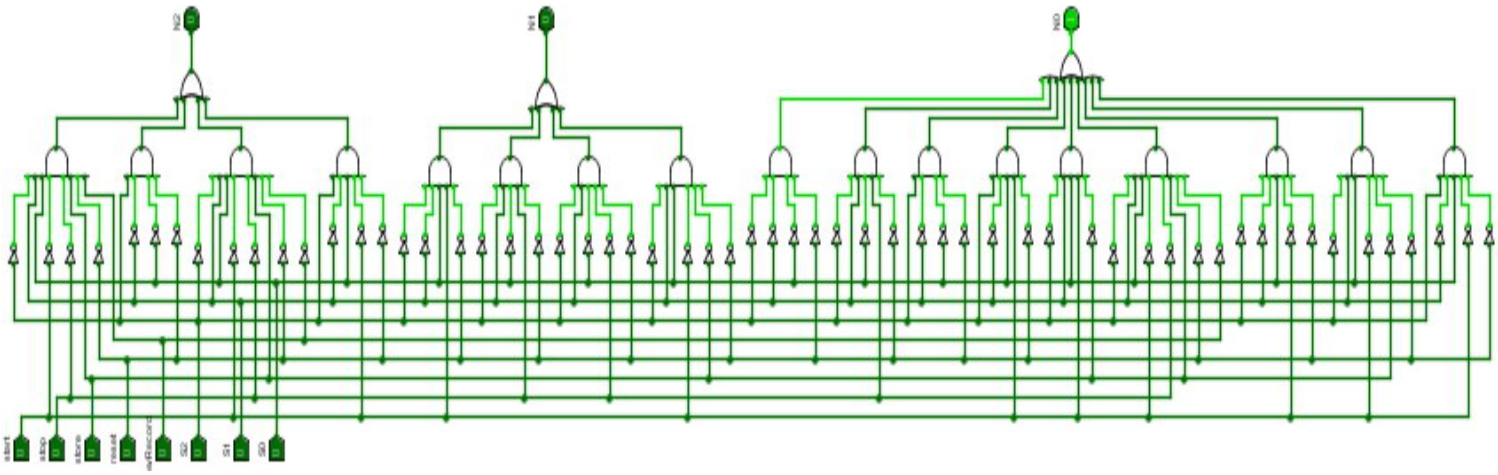
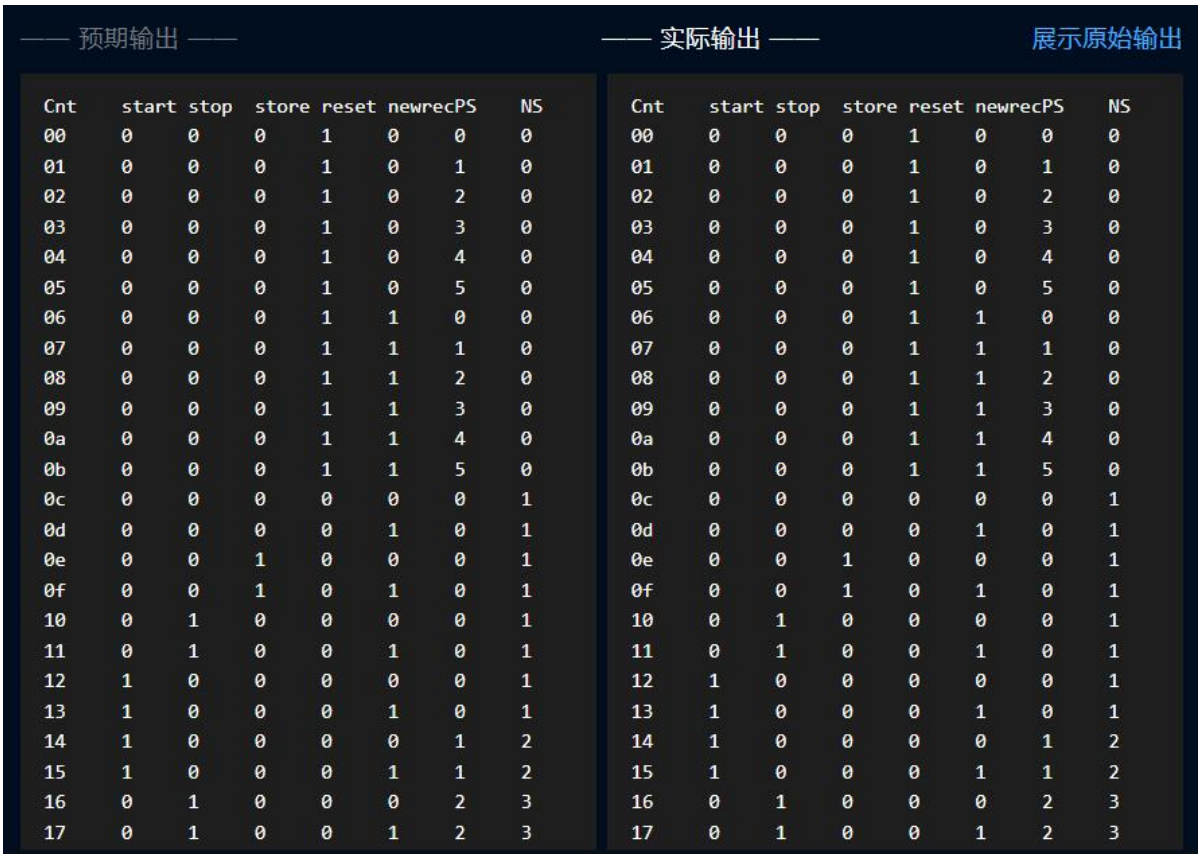


图 2.32 码表控制器状态机电路图

设计的电路如图 2.32 所示。

（3）测试图



—— 预期输出 ——

Cnt	start	stop	store	reset	newrecPS	NS
00	0	0	0	1	0	0	0
01	0	0	0	1	0	1	0
02	0	0	0	1	0	2	0
03	0	0	0	1	0	3	0
04	0	0	0	1	0	4	0
05	0	0	0	1	0	5	0
06	0	0	0	1	1	0	0
07	0	0	0	1	1	1	0
08	0	0	0	1	1	2	0
09	0	0	0	1	1	3	0
0a	0	0	0	1	1	4	0
0b	0	0	0	1	1	5	0
0c	0	0	0	0	0	0	1
0d	0	0	0	0	1	0	1
0e	0	0	1	0	0	0	1
0f	0	0	1	0	1	0	1
10	0	1	0	0	0	0	1
11	0	1	0	0	1	0	1
12	1	0	0	0	0	0	1
13	1	0	0	0	1	0	1
14	1	0	0	0	0	1	2
15	1	0	0	0	1	1	2
16	0	1	0	0	0	2	3
17	0	1	0	0	1	2	3

—— 实际输出 ——　　展示原始输出

Cnt	start	stop	store	reset	newrecPS	NS
00	0	0	0	1	0	0	0
01	0	0	0	1	0	1	0
02	0	0	0	1	0	2	0
03	0	0	0	1	0	3	0
04	0	0	0	1	0	4	0
05	0	0	0	1	0	5	0
06	0	0	0	1	1	0	0
07	0	0	0	1	1	1	0
08	0	0	0	1	1	2	0
09	0	0	0	1	1	3	0
0a	0	0	0	1	1	4	0
0b	0	0	0	1	1	5	0
0c	0	0	0	0	0	0	1
0d	0	0	0	0	1	0	1
0e	0	0	1	0	0	0	1
0f	0	0	1	0	1	0	1
10	0	1	0	0	0	0	1
11	0	1	0	0	1	0	1
12	1	0	0	0	0	0	1
13	1	0	0	0	1	0	1
14	1	0	0	0	0	1	2
15	1	0	0	0	1	1	2
16	0	1	0	0	0	2	3
17	0	1	0	0	1	2	3

图 2.33　码表控制器状态机部分测试样例

部分测试样例如图 2.33 所示。

（4）测试分析

由测试结果可知，该电路可以按要求正确实现码表控制器的状态转移功能，所设计的电路符合要求。

2.13　码表控制器输出函数设计

（1）设计思路及设计过程

设输出为

SDsel　　最好成绩记录的选择信号

SDen　　保存最好成绩记录的寄存器的使能信号

DPsel　　显示计时成绩记录的选择信号

TMen　码表计时器使能信号

TMreset 码表计时器复位信号

根据状态机的状态与输出的对应关系，可以得出真值表为：

当前状态(现态)				输入信号												
S2	S1	S0	现态10进制	start	stop	store	reset	NewRecord	In6	In7	In8	Out1	Out2	Out3	Out4	Out5
0	0	0	0									0	1	1	0	1
0	0	1	1									0	0	1	0	1
0	1	0	2									0	0	1	1	0
0	1	1	3									0	0	1	0	0
1	0	0	4									1	1	1	0	0
1	0	1	5									0	0	0	0	0

图 2.34　码表控制器输出函数真值表

根据真值表可以写出输出函数表达式为：

SDsel：S2 ~S1 ~S0

SDen：~S1 ~S0

DPsel：~S2 + ~S1 ~S0

TMen：~S2 S1 ~S0

TMreset：~S2 ~S1

根据输出函数表达式可以直接生成电路。

（2）电路图

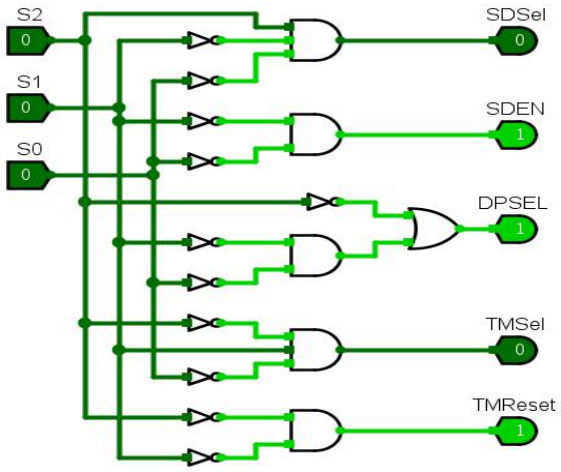


图 2.35　码表控制器输出函数电路

设计的电路如图 2.35 所示。

（3）测试图

Cnt	PS	SDsel	SDen	DPsel	TMen	TMrese
00	0	0	1	1	0	1
01	1	0	0	1	0	1
02	2	0	0	1	1	0
03	3	0	0	1	0	0
04	4	1	1	1	0	0
05	5	0	0	0	0	0

Cnt	PS	SDsel	SDen	DPsel	TMen	TMrese
00	0	0	1	1	0	1
01	1	0	0	1	0	1
02	2	0	0	1	1	0
03	3	0	0	1	0	0
04	4	1	1	1	0	0
05	5	0	0	0	0	0

图 2.36　码表控制器输出函数测试样例

测试样例如图 2.36 所示。

（4）测试分析

由测试结果可知，所有状态对应的输出都正常，所设计的电路符合要求。

2.14 码表控制器

（1）设计思路及设计过程

该模块实现的是所有按钮的功能，为了实现循环控制，可以利用 D 触发器，将 D 触发器的输出接入输出函数以及状态转换模块，同时状态转换模块的输出接入 D 触发器的 D 端，将其他输入端口接好，即可实现循环控制的功能。

（2）电路图

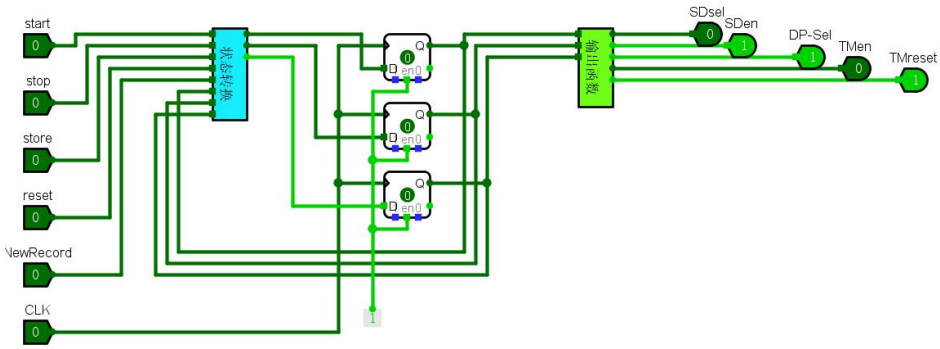
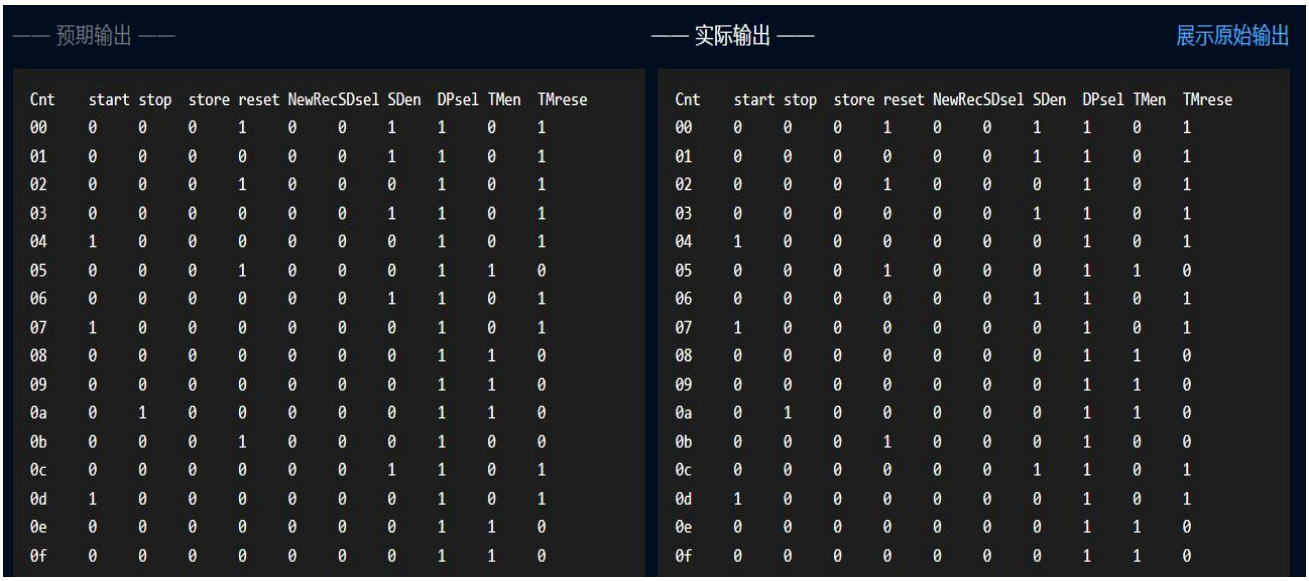


图 2.37　码表控制器电路图

设计的电路如图 2.37 所示。

（3）测试图



—— 预期输出 ——

Cnt	start	stop	store	reset	NewRec	SDsel	SDen	DPsel	TMen	TMrese
00	0	0	0	1	0	0	1	1	0	1
01	0	0	0	0	0	0	1	1	0	1
02	0	0	0	1	0	0	0	1	0	1
03	0	0	0	0	0	0	1	1	0	1
04	1	0	0	0	0	0	0	1	0	1
05	0	0	0	1	0	0	0	1	1	0
06	0	0	0	0	0	0	1	1	0	1
07	1	0	0	0	0	0	0	1	0	1
08	0	0	0	0	0	0	0	1	1	0
09	0	0	0	0	0	0	0	1	1	0
0a	0	1	0	0	0	0	0	1	1	0
0b	0	0	0	1	0	0	0	1	0	0
0c	0	0	0	0	0	0	1	1	0	1
0d	1	0	0	0	0	0	0	1	0	1
0e	0	0	0	0	0	0	0	1	1	0
0f	0	0	0	0	0	0	0	1	1	0

—— 实际输出 ——　　展示原始输出

Cnt	start	stop	store	reset	NewRec	SDsel	SDen	DPsel	TMen	TMrese
00	0	0	0	1	0	0	1	1	0	1
01	0	0	0	0	0	0	1	1	0	1
02	0	0	0	1	0	0	0	1	0	1
03	0	0	0	0	0	0	1	1	0	1
04	1	0	0	0	0	0	0	1	0	1
05	0	0	0	1	0	0	0	1	1	0
06	0	0	0	0	0	0	1	1	0	1
07	1	0	0	0	0	0	0	1	0	1
08	0	0	0	0	0	0	0	1	1	0
09	0	0	0	0	0	0	0	1	1	0
0a	0	1	0	0	0	0	0	1	1	0
0b	0	0	0	1	0	0	0	1	0	0
0c	0	0	0	0	0	0	1	1	0	1
0d	1	0	0	0	0	0	0	1	0	1
0e	0	0	0	0	0	0	0	1	1	0
0f	0	0	0	0	0	0	0	1	1	0

图 2.38　码表控制器部分测试样例

部分测试样例如图 2.38 所示。

（4）测试分析

由测试结果可知，该电路可以按要求正确实现状态的转移以及产生正确的输出输出，所设计的电路符合要求。

2.15　运动码表

（1）设计思路及设计过程

从下往上看，计时器 TM 通过计数器来实现计时，第一个选择器用来实现 9999 与实时计时的选择，16 位寄存器用来传递选择器的输出，第二个选择器用来实现第一个选择器的输出数据与计时器的输出数据的选择，第二个选择器的输出通过码表显示 DP 来输出出来。比较器用来记录更小的计时记录。

（2）电路图

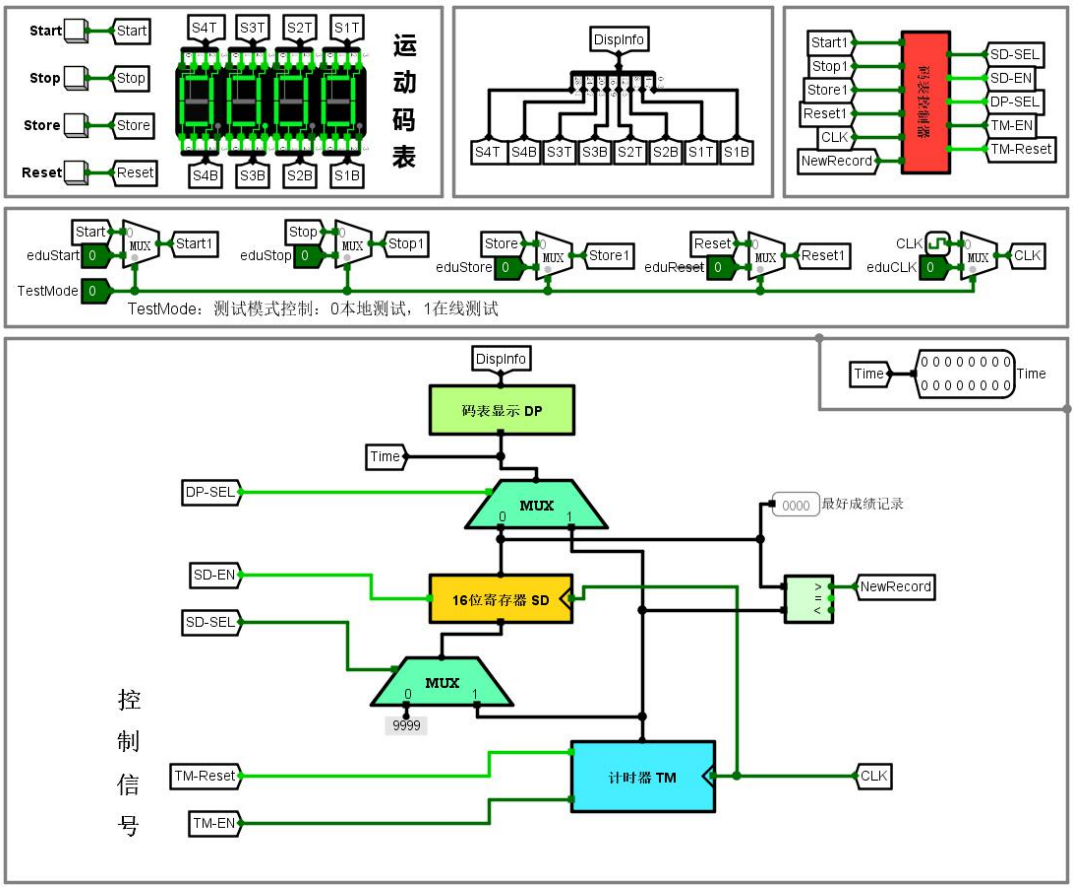
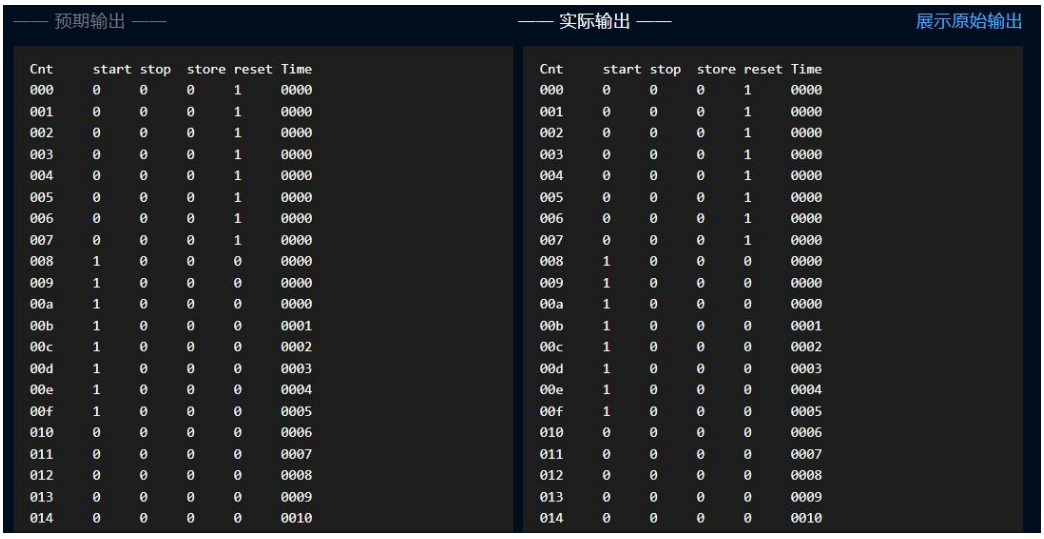


图 2.39 运动码表电路图

设计的电路如图 2.39 所示。

（2）测试图



预期输出

Cnt	start	stop	store	reset	Time
000	0	0	0	1	0000
001	0	0	0	1	0000
002	0	0	0	1	0000
003	0	0	0	1	0000
004	0	0	0	1	0000
005	0	0	0	1	0000
006	0	0	0	1	0000
007	0	0	0	1	0000
008	1	0	0	0	0000
009	1	0	0	0	0000
00a	1	0	0	0	0000
00b	1	0	0	0	0001
00c	1	0	0	0	0002
00d	1	0	0	0	0003
00e	1	0	0	0	0004
00f	1	0	0	0	0005
010	0	0	0	0	0006
011	0	0	0	0	0007
012	0	0	0	0	0008
013	0	0	0	0	0009
014	0	0	0	0	0010

实际输出 展示原始输出

Cnt	start	stop	store	reset	Time
000	0	0	0	1	0000
001	0	0	0	1	0000
002	0	0	0	1	0000
003	0	0	0	1	0000
004	0	0	0	1	0000
005	0	0	0	1	0000
006	0	0	0	1	0000
007	0	0	0	1	0000
008	1	0	0	0	0000
009	1	0	0	0	0000
00a	1	0	0	0	0000
00b	1	0	0	0	0001
00c	1	0	0	0	0002
00d	1	0	0	0	0003
00e	1	0	0	0	0004
00f	1	0	0	0	0005
010	0	0	0	0	0006
011	0	0	0	0	0007
012	0	0	0	0	0008
013	0	0	0	0	0009
014	0	0	0	0	0010

图 2.40　运动码表部分测试样例

部分测试样例如图 2.40 所示。

（4）测试分析

由测试结果可知，该电路可以按要求正确实现运动码表的开始、暂停、复位、记录成绩等功能，所设计的电路符合要求。

3　测试及故障调试

3.1 遇到的问题及处理

在最后整合电路的测评中，发现每一次时钟脉冲出现了两次计时，在询问了老师之后，猜测可能是由于触发器的上升沿下降沿选取错误。在仔细检查了每一个模块之后，发现在码表计数器模块，原本的设计是将本级计数器的输出端接到下一级计数器的时钟端,这样就需要将 BCD 计数器中的 D 触发器调节为下降沿才能保证有正确结果，而其他含有 D 触发器的模块中，所有的 D 触发器均为上升沿，这样上升沿和下降沿在同一个电路中使用，可能就会导致计时的错误。之后我将 BCD 计数器中的 D 触发器由下降沿改成了上升沿,将码表计数器的连接方式改成了将 en 和本级计数器之前的计数器的进位输出的与来作为本级计数器的使能端，这样调整之后就得到了正确的结果。

3.2 设计方案存在的不足

①只实现了正计时而没有实现倒计时。

②缺少定时功能。

③不能回看先前保存的记录

4　设计总结与心得

4.1 实验总结

本次实验将运动码表的功能进行分解，分解成计时器、选择器、寄存器、比较器、显示器、控制器等模块，分别实现这几个模块的功能，然后将这几个模块进行整合，实现运动码表的所有功能。

4.2 实验心得

本次实验真正体验了模块式设计的思想。从最基础的数码管，到最复杂的状态机，由易到难，层层深入。在设计每一块芯片时，除了要考虑完成当前模块的功能，还需要考虑与其他模块的衔接，就像上文中说的那样，注意触发器上升沿和下降沿的选用，否则就会因为不一致而导致整合电路的时候出现错误。而且在设计每一块芯片的时候，要先分析好每一个模块的功能，与课内知识结合，自己思考如何去实现所需要的功能，在有错误的时候自己去尝试改进，这才是本次实验的根本目的。

4.3 意见与建议

意见：在本次实验的码表显示驱动模块，头歌上所要求的连接方式在最后整合电路的时候出现了问题，并不能正确显示 00.00，若想要完成整合电路的测评，需要先改动码表显示驱动。

建议：

①最后整合码表电路的时候，可以将已经放好的原件去掉，自己思考该如何用之前已经实现的模块去组成一个符合要求的运动码表。

②可以在实验开始之前先引导学生思考需要哪些功能和模块，然后自己根据课内知识尝试实现。

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。

作者签名：张文浩