

华中科技大学

2020

逻辑与计算机设计基础
课程实验报告

专 业:	信息安全
班 级:	
学 号:	
姓 名:	
电 话:	
邮 件:	
完成日期:	2020 年 04 月 21 日

目 录

1	实验任务与要求	3
1.1	实验内容	3
1.2	具体要求	3
2	设计方案	4
2.1	LED 计数电路的设计	4
2.2	5 输入编码器的设计	4
2.3	数码管驱动模块设计	5
2.4	1 位 2 路选择器模块设计	6
2.5	16 位 2 路选择器模块设计	7
2.6	4 位无符号比较器模块设计	8
2.7	16 位无符号比较器模块设计	9
2.8	码表显示驱动模块设计	10
2.9	4 位并行加载寄存器模块设计	11
2.10	16 位并行加载寄存器模块设计	12
2.11	4 位 BCD 计数器模块设计	13
2.12	BCD 计数器状态转换模块设计	15
2.13	BCD 计数器输出函数模块设计	15
2.14	码表计数器模块设计	16
2.15	码表控制器模块设计	18
2.16	码表状态转换模块设计	18
2.17	码表控制器输出函数模块设计	20
2.18	运动码表模块设计	20
3	测试及故障调试	22
3.1	测试	22

3.2	遇到的问题及处理.....	25
3.3	设计方案存在的不足.....	26
4	设计总结与心得.....	27
4.1	实验总结.....	27
4.2	实验心得及建议.....	27
	参考文献.....	28

1 实验任务与要求

1.1 实验内容

通过围绕小型数字系统——数字码表进行相应的基本部件的电路设计，并最终完成能够实现开始计时、停止计时、更新记录和时间复位等功能的运动码表的整体设计和构建。

1.2 具体要求

1.完成 LED 计数电路、输入优先数据编码器和 7 段数码管显示译码器这 3 个组合逻辑电路的设计。熟悉输入输出引脚的元器件及标签用法，对引脚和逻辑门的属性进行熟悉，尝试一些快捷键的用法；掌握电路的封装方法；以及利用分析电路自动生成电路的方法。

2.完成 2 路选择器设计（16 位）、16 位无符号比较器和码表数码管显示驱动的设计和虚拟仿真。熟悉掌握组合逻辑电路特征及设计流程，理解模块的分层、迭代思想，熟悉利用 logisim 构建运动码表基本部件。

3.完成 16 位寄存器（16 位）、4 位 BCD 计数器和码表计数器的设计和虚拟仿真。熟悉掌握同步时序电路特征及设计流程，对构建状态图，状态转换逻辑、输出函数逻辑等有进一步的体会，熟悉利用 logisim 构建运动码表基本时序逻辑。

4. 熟悉数字系统基本特征及设计方法，熟练掌握数字系统设计基本流程，利用 Logisim 构建运动码表数据通路和控制逻辑，最终完成整个码表功能设计。

2 设计方案

2.1 LED 计数电路的设计

- 1.功能：用输出 1 的个数来表示输入引脚的编号。
- 2.输入：5 个输入变量 In1、In2、In3、In4、In5，五个输入互斥，即不能同时输入为 1。
- 3.输出：5 个输出变量 out 1、out 2、out 3、out 4、out 5。
- 4.要求：当 In_i ($1 \leq i \leq 5$) 按下（此时其他输入变量不被按下），所有的 out_j ($1 \leq j \leq i$) 输出为 1，而 out_j ($i < j \leq 5$) 输出为 0。
- 5.设计思路：

（1）根据要求构建真值表：该表中仅当 5 个输入至多 1 个“1”的时候，有确定的输出值，具体值按照上述要求（如 In₄=1 时，out₁=out₂=out₃=out₄=1，out₅=0）；真值表的其他行输出值均为 xxxxx 表示无关。

（2）将上述真值表填入 logisim 分析电路真值表中，自动生成电路，如图。

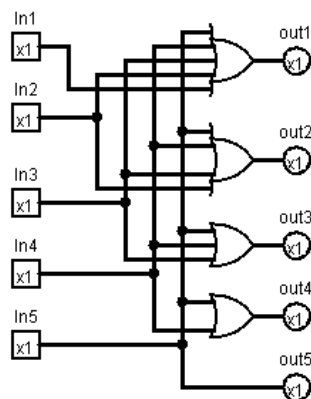


图 2-1 LED 计数电路

2.2 5 输入编码器的设计

- 1.功能：对五个输入中优先级最高的输入进行编码。
- 2.输入：5 个输入变量 In1、In2、In3、In4、In5，五个输入具有优先级，优先级排序位 $In5 > In4 > In3 > In2 > In1$ 。
- 3.输出：out 2、out 1、out 0。
- 4.要求：

当 In₅ 按下 (In₅ = 1)，out 2 out 1 out 0 = 1 0 1；

当 In4 按下 (In4 = 1), out 2 out 1 out 0 = 1 0 0 ;

当 In3 按下 (In3 = 1), out 2 out 1 out 0 = 0 1 1 ;

当 In2 按下 (In2 = 1), out 2 out 1 out 0 = 0 1 0 ;

当 In1 按下 (In1 = 1), out 2 out 1 out 0 = 0 0 1 。

5.设计思路:

(1) 根据要求构建真值表: 具体值按照上述要求 (如 In5=0、In4=1 时, 无论 In1、In2、In3 取值如何, out 2 out 1 out 0 = 1 0 0)

(2) 将上述真值表填入 logisim 分析电路真值表中, 自动生成电路, 如图。

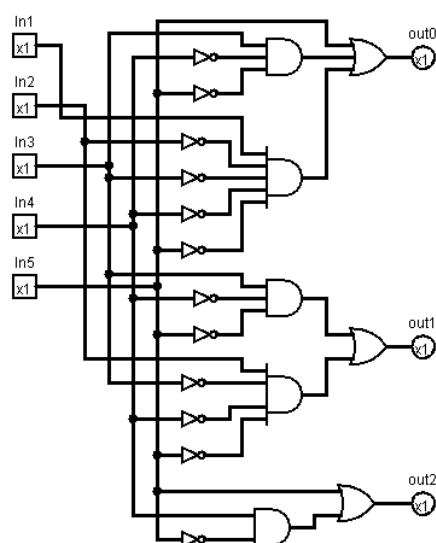


图 2-2 5 输入编码器电路

2.3 数码管驱动模块设计

1.功能: 利用 7 段数码管显示 4 位二进制数的 10 进制值。

2.输入: 4 位二进制数。

3.输出: 7 段显示器的 7 个输出控制信号 a、b、c、d、e、f、g。

4.设计思路：构建真值表，生成逻辑表达式，利用分析电路自动生成电路。

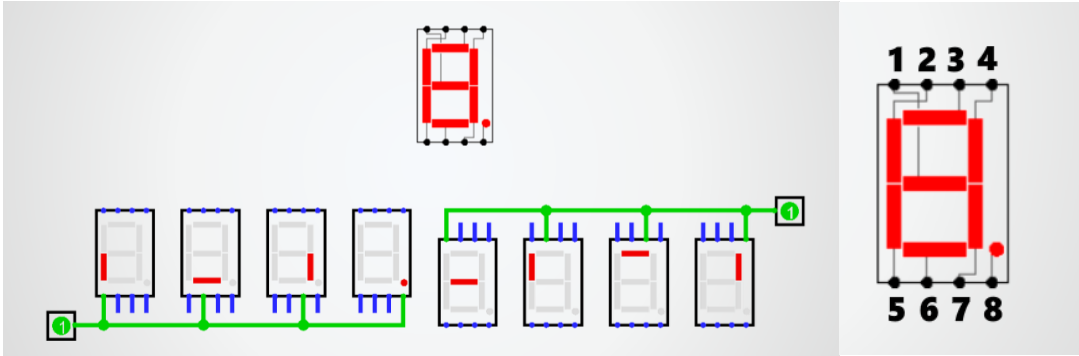


图 2-3 7 段数码管引脚与显示对应关系

(1) 根据上图 7 段数码管的引脚与相应数码管显示，构建从 4 位输入 X3~X0 到 7 位输出 Seg_1 到 Seg_7 的真值表如下图，其中小数点数码管可用电源单独控制。

X3	X2	X1	X0	Seg_1	Seg_2	Seg_3	Seg_4	Seg_5	Seg_6	Seg_7	对应数字
0	0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	0	0	1	0	0	1	1
0	0	1	0	1	0	1	1	1	1	0	2
0	0	1	1	1	0	1	1	0	1	1	3
0	1	0	0	1	1	0	1	0	0	1	4
0	1	0	1	1	1	1	0	0	1	1	5
0	1	1	0	1	1	1	0	1	1	1	6
0	1	1	1	0	0	1	1	0	0	1	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	1	0	1	1	9
1	0	1	0	0	0	0	0	0	0	0	
1	0	1	1	0	0	0	0	0	0	0	
1	1	0	0	0	0	0	0	0	0	0	
1	1	0	1	0	0	0	0	0	0	0	
1	1	1	0	0	0	0	0	0	0	0	
1	1	1	1	0	0	0	0	0	0	0	
1	1	1	1	0	0	0	0	0	0	0	

图 2-4 数码管驱动电路的真值表

(2) 由真值表利用 logisim 自动生成电路。

2.4 1 位 2 路选择器模块设计

1.电路功能：根据选择控制信号，将相应的输入进行输出。

2.输入：1 位数据输入 X 和 Y，1 个选择控制信号 Sel。

3.输出：16 位输出 Out。

4.要求：当 Sel=0 时，Out=X；当 Sel=1 时，Out=Y。

5.设计思路：

(1)由上述要求可得如下图真值表，利用卡诺图可得输出表达式 $Out = \sim(Sel) X0 + Sel$

X1。

Sel	X0	Y1	Out
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

图 2-5 1 位 2 路选择器真值表

(2) 由表达式构建电路，如图。

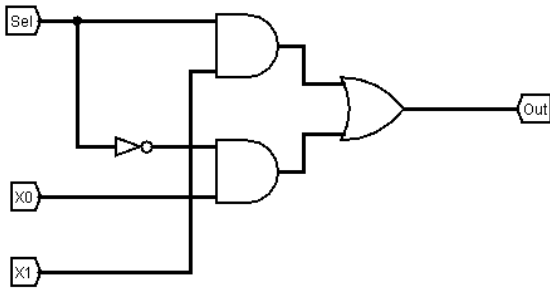
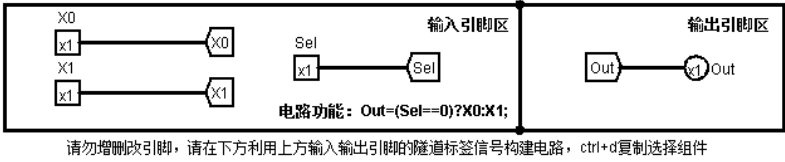


图 2-6 1 位 2 路选择器电路

2.5 16 位 2 路选择器模块设计

- 1.电路功能：根据选择控制信号，将相应的输入进行输出。
- 2.输入：16 位数据输入 X 和 Y，1 个选择控制信号 Sel。
- 3.输出：16 位输出 Out。
- 4.要求：当 Sel=0 时，Out=X；当 Sel=1 时，Out=Y。
- 5.设计思路：

(1) 利用上述 2.4 设计的 1 位 2 路选择器。将 16 位输入 X 和 Y 分别接一个 16 位分线器各获得 16 位线路。X 和 Y 对应的每一位接一个 1 位 2 路选择器，即 X 第 1 位和 Y 第 1 位接一个 1 位 2 路选择器、X 第 2 位和 Y 第 2 位接一个 2 路选择器...一共需要 16 个 1 位 2 路选择器，每一个对应 X 或 Y 的 1 位数据。

(2) 16 个 1 位 2 路选择器的 Sel 均由 16 位 2 路选择器的总 Sel 控制；16 个 1 位 2

路选择器的输出最后接一个 16 位分线器合并为 16 位输出 Out。

(3) 有上述分析构建的电路如下图。

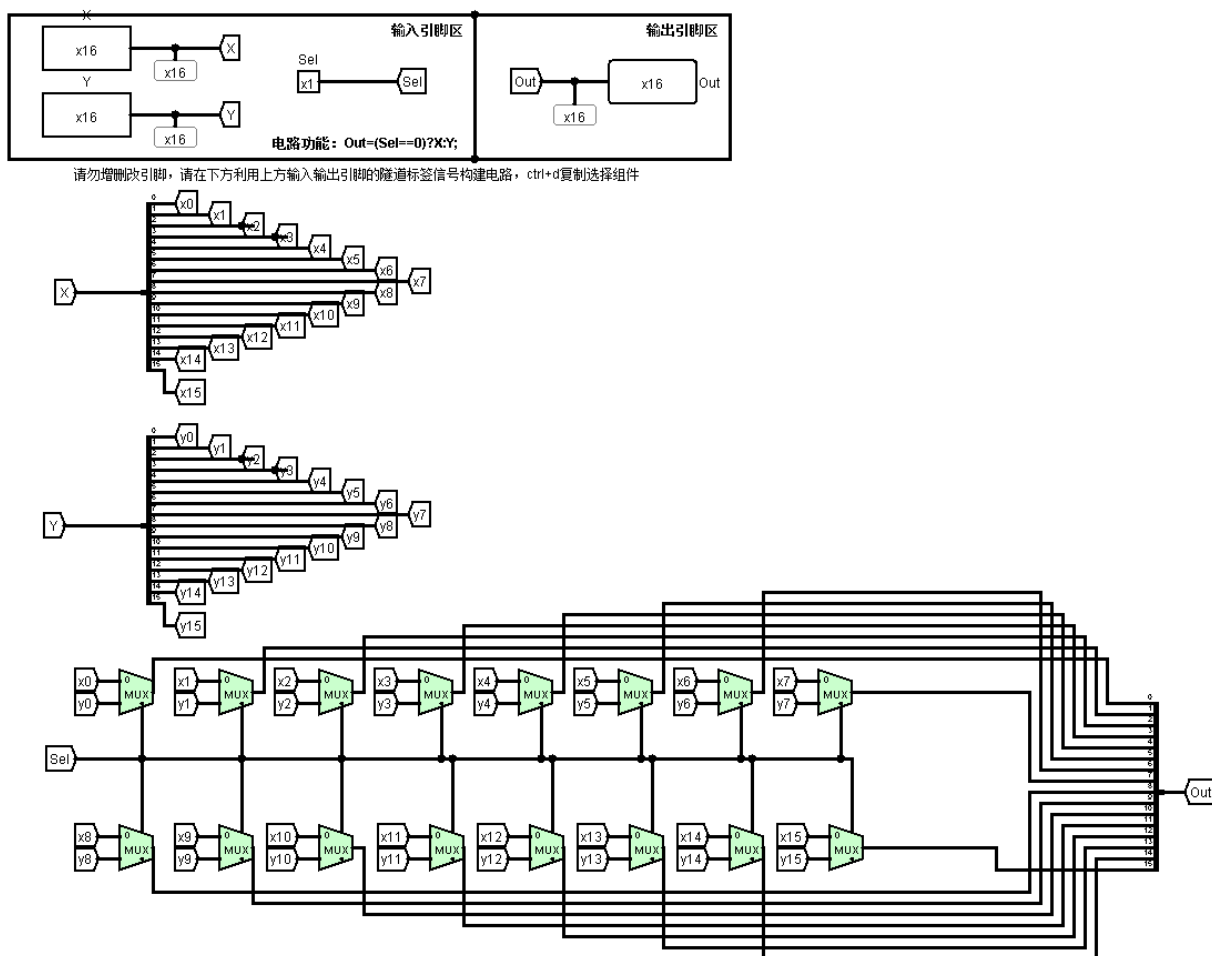


图 2-7 16 位 2 路选择器电路

2.6 4 位无符号比较器模块设计

- 1.功能：对两个 4 位的无符号数进行比较。
- 2.输入：2 个 4 位二进制数 X ($X_3X_2X_1X_0$) 和 Y ($Y_3Y_2Y_1Y_0$)。
- 3.输出：Great (1 位)、Equal (1 位)、Less (1 位)。
- 4.设计思路：

(1) 由分析易知：当 X 和 Y 的 4 位数字均相等时， $X=Y$ ，Equal 输出为 1。当 X 的高 i 位 ($0 \leq i \leq 3$) 与 Y 的高 i 位均相等，且 X 的第 i+1 位大于 Y 的第 i+1 位时， $X > Y$ ，Great 输出 1。当 X 的高 i 位 ($0 \leq i \leq 3$) 与 Y 的高 i 位均相等，且 X 的第 i+1 位小于 Y 的第 i+1 位时， $X < Y$ ，Less 输出 1。

(2) 由上述分析，可构建输出 Great、Equal、Less 与输入 X 和 Y 的表达式：

$$\begin{aligned} \text{Great} &= X_3 \sim Y_3 + \sim(X_3 \sim Y_3 + \sim X_3 Y_3) X_2 \sim Y_2 + \sim(X_3 \sim Y_3 + \sim X_3 Y_3 + X_2 \sim Y_2 + \sim X_2 Y_2) X_1 \sim Y_1 + \sim(X_3 \sim Y_3 + \sim X_3 Y_3 + X_2 \sim Y_2 + \sim X_2 Y_2 + X_1 \sim Y_1 + \sim X_1 Y_1) X_0 \sim Y_0 \\ \text{Equal} &= \sim(X_3 \sim Y_3 + \sim X_3 Y_3 + X_2 \sim Y_2 + \sim X_2 Y_2 + X_1 \sim Y_1 + \sim X_1 Y_1 + X_0 \sim Y_0 + \sim X_0 Y_0) \\ \text{Less} &= \sim X_3 Y_3 + \sim(X_3 \sim Y_3 + \sim X_3 Y_3) \sim X_2 Y_2 + \sim(X_3 \sim Y_3 + \sim X_3 Y_3 + X_2 \sim Y_2 + \sim X_2 Y_2) \sim X_1 Y_1 + \sim(X_3 \sim Y_3 + \sim X_3 Y_3 + X_2 \sim Y_2 + \sim X_2 Y_2 + X_1 \sim Y_1 + \sim X_1 Y_1) \sim X_0 Y_0 \end{aligned}$$

(3) 将上述表达式填入 logisim 分析电路的表达式框中，自动生成电路。

2.7 16 位无符号比较器模块设计

- 1.功能：对两个 16 位的无符号数进行比较。
- 2.输入：2 个 16 位数据输入 X 和 Y，由低位到高位每 4 位一组记为 X₀X₁X₂X₃ 和 Y₃Y₂Y₁Y₀。
- 3.输出：Great（1 位）、Equal（1 位）、Less（1 位）。
- 4.设计思路：

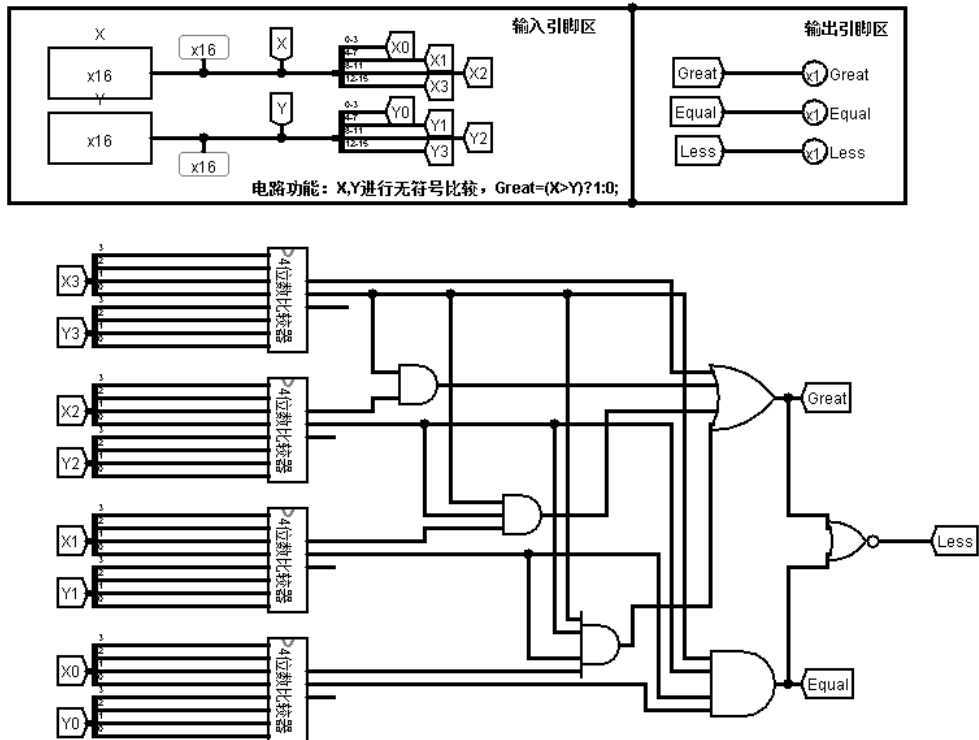
(1) 利用 4 个 2.6 设计的 4 位无符号比较器，可以实现输入 X 和 Y 每 4 位 X_i 和 Y_i(0≤i≤3)的大小比较,并且每个 4 无符号比较器都有 3 个输出 Great、Equal 和 Less。

(2) 16 位无符号比较器 Equal=1：此时，4 个 4 位无符号比较器的输出 Equal 应均等于 1，可将 4 个 Equal 输出接与门实现。

(3) 16 位无符号比较器 Great=1：此时有 4 种情况：X₃ 和 Y₃（即最高位）的 4 位数比较器 Great=1；X₃ 和 Y₃ 的 4 位数比较器 Equal=1，且 X₂ 和 Y₂ 的 4 位数比较器 Great=1；X₃ 和 Y₃、X₂ 和 Y₂ 两个 4 位数比较器 Equal=1，且 X₁ 和 Y₁ 的 4 位数比较器 Great=1；X₃ 和 Y₃、X₂ 和 Y₂、X₁ 和 Y₁ 三个 4 位数比较器 Equal=1，且 X₀ 和 Y₀ 的 4 位数比较器 Great=1。这四种情况中出现一种，16 位数比较器的 Great 即为 1。

(4) 16 位无符号比较器 Less=1：当 16 位无符号比较器 Great 和 Equal 均不为 1 时，Less=1，即 Less=∼（Great+Equal）。可将输出 Great 和 Equal 接或非门实现。

(5) 由上述分析可构建电路如下图。



2.8 码表显示驱动模块设计

- 1.功能：利用 4 个并行的 7 段数码管显示 16 位的 BCD 码。
- 2.输入：16 位 BCD 码 Din。
- 3.输出：4 个 7 段数码管的控制信号（32 位数据）DispInfo：

S4T,S4B,S3T,S3B,S2T,S2B,S1T,S1B 。

这 32 位数据中，每 8 位数据用来控制显示 1 个 7 段数码管的控制信号。

4.要求：

- (1) 将 16 位 BCD 码分为 4 组，每组 BCD 码（4 位）作为 1 个 7 段数码管的控制信号的输入；
- (2) 数码管中小数点单独控制，每个控制信号是 1 位；
- (3) 对于每个 7 段数码管，可直接调用第一次实验中已经设计封装的 7 段数码管的电路。

5.设计思路：

- (1) 将 16 位输入 Din 通过 16 位分线器分为 16 个 1 位线路，每 4 位成 1 组，作为

输入接入到 2.3 设计的数码管驱动模块中，共需要 4 个数码管驱动模块，分别对应码表显示驱动的每一个 7 段数码管。

(2) 每一个数码管的 7 位输出对应 7 段数码管数字部分的数据，而数码管的小数点由一常量单独控制，其中第三个 7 段数码管由码表实现需要小数点对应位为常量“1”，其余 3 个小数点对应位为常量“0”。

(4) 将数码管驱动的 7 位输出和 1 位小数点控制位共 8 位分成 2 个 4 位组，分别对应控制 7 段数码管的上引脚 SiT 和下引脚 SiB ($1 \leq i \leq 4$)，最后由分线器将 8 组共 32 位数据合并到一位输出 DispInfo。

(5) 由上述分析可构建电路如下图。

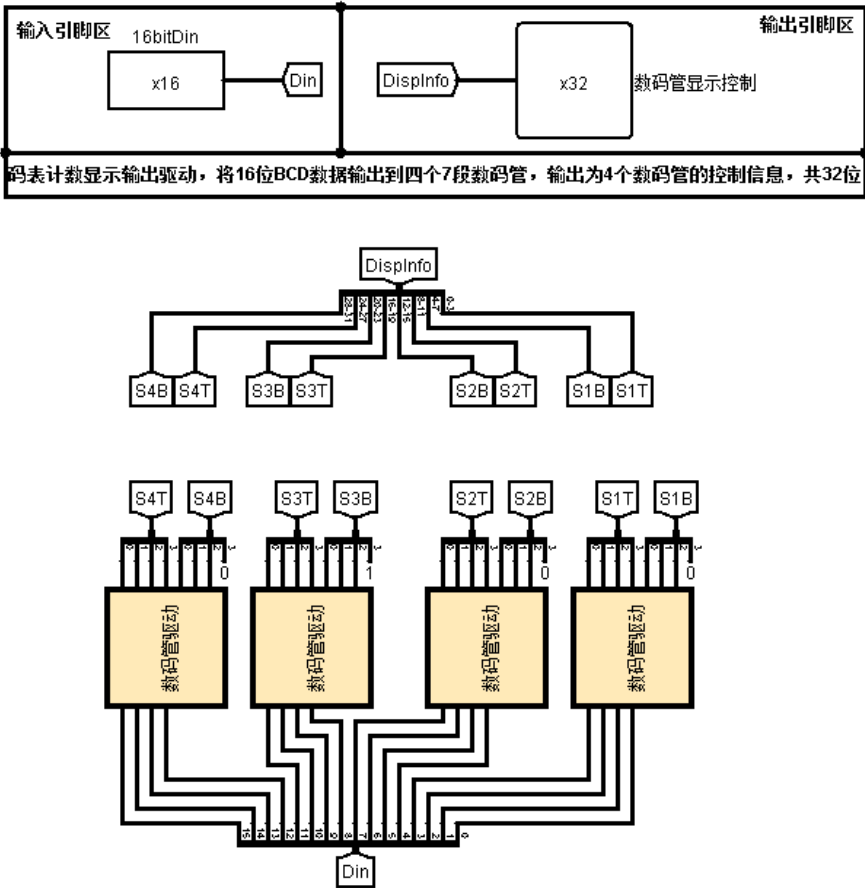


图 2-9 码表显示驱动电路

2.9 4 位并行加载寄存器模块设计

1. 电路功能：根据控制信号，将相应的输入进行输出。
2. 输入：4 位数据输入 Din，1 个使能信号 En, 1 个时钟信号 Clk。
3. 输出：4 位输出 Q。

4.要求：当 $En=1$ 且有时钟信号输入变化时（Clk 从 0 变化到 1）时， $Q=Din$ ；否则 Q 的值维持不变。使用 4 个 D 触发器设计。

5.设计思路：

（1）由 D 触发器特性知，在触发器工作状态（ $En=1$ ）时，有时钟信号时，触发器次态等于激励 D,无时钟信号时,触发器状态保持不变。因此，可由 4 个 D 触发器实现：触发器连有共同的使能端 En 和时钟端 CLK ； Din 通过分线器分为 4 位，每一位分别作为每个触发器的激励，触发器的次态接分线器构成 4 位输出 Q 。

（2）由上述分析可构建如下图电路。

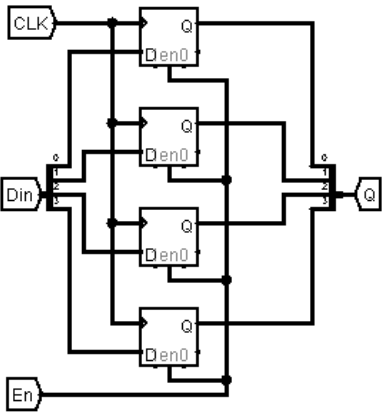
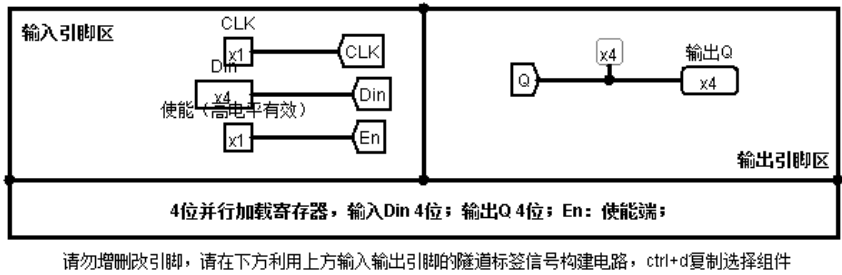


图 2-10 4 位并行加载寄存器电路

2.10 16 位并行加载寄存器模块设计

1.电路功能：根据控制信号，将相应的输入进行输出。

2.输入：16 位数据输入 Din ，1 个使能信号 En ，1 个时钟信号 Clk 。

3.输出：16 位输出 Q 。

4.要求：

(1) 当 $En=1$ 且有时钟信号输入变化时（Clk 从 0 变化到 1）时， $Q=Din$ ；否则 Q 的

值维持不变；

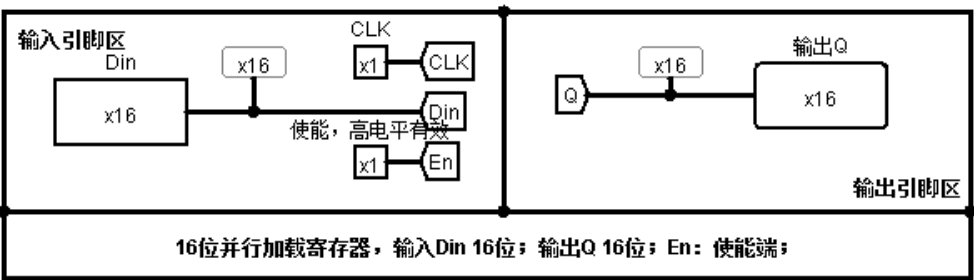
(2) 先使用 4 个 D 触发器设计 4 位的并行加载寄存器并进行封装；

(3) 用 4 个 4 位的并行加载寄存器进行并发得到 16 位的并行加载寄存器。

5.设计思路：

(1) 利用 4 个 2.9 设计的 4 位并行加载寄存器，连有共同的使能端 En 和时钟端 CLK；Din 通过分线器分为 4 个 4 位数据线路，分别作为每个 4 位并行加载寄存器的输入，4 位寄存器的输出接分线器构成 16 位输出 Q。

(2) 由上述分析可构建如下图电路。



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

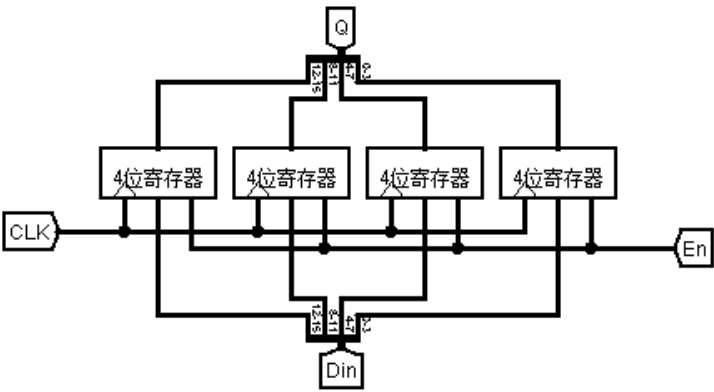


图 2-11 16 位并行加载寄存器电路

2.114 位 BCD 计数器模块设计

1.功能：对输入的时钟信号进行计数。

2.输入：时钟信号 Clk，使能信号 En，异步复位信号 Rst。

3.输出：4 位输出 Q，进位输出 Cout。

4.要求：

(1) 当 En=1 且有时钟信号输入变化时（Clk 从 0 变化到 1）时，Q 的值增加 1，且

Q 只能在 0-9 之间变化， $Cout=(Q==9)?1:0$ ；当 Rst 为 1 时，Q 和 Cout 均清零。

(2) 先设计 BCD 计数器的状态转换和输出函数电路并进行封装；

(3) 利用状态转换电路和输出函数逻辑电路，结合 D 触发器构建 4 位 BCD 计数器。

5. 设计思路：

(1) BCD 计数器状态转换和输出函数模块见 2.12 和 2.13，此处直接使用二者构建好的封装模块。

(2) 由分析可知，可用 4 个 D 触发器进行状态控制，当触发器处于工作状态，来一时钟信号，使电路的状态发生改变。

(3) 具体实现：触发器的使能端、复位端和时钟端分别接使能信号 En、异步复位信号 Rst 和时钟信号 Clk。触发器共 4 位输入（现态）由状态转换模块的 4 位输出提供，触发器共四位输出作为状态转换模块的 4 位输入、输出函数的 4 位输入和 4 位输出 Q。输出函数的输出为进位输出 Cout。

(4) 由上述分析可得如下图电路。

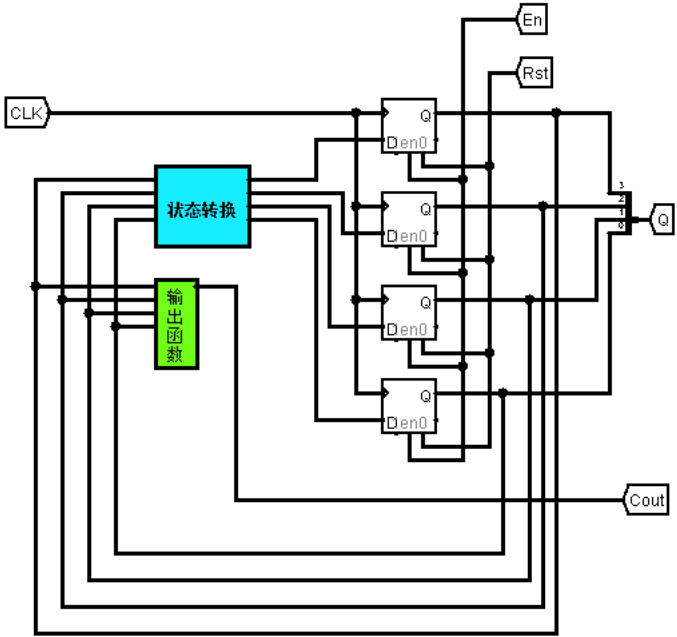
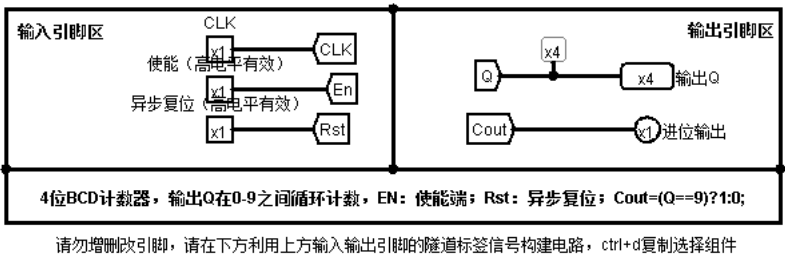


图 2-12 4 位 BCD 计数器电路

2.12 BCD 计数器状态转换模块设计

- 1.功能：实现电路现态到次态的状态转换。
- 2.输入：4 位现态 S3S2S1S0。
- 3.输出：4 位次态 N3N2N1N0。
- 4.设计思路：

(1) 由于 BCD 计数器输出 Q 仅能在 0~9 之间变化，因此电路应该有 10 种状态分别对应 0~9，状态的二进制表示对应 4 位输入和 4 位输出。由于是计数器，因此状态是连续的，即电路由 i 状态转换为 i+1 状态 ($0 \leq i \leq 8$)，有 9 状态转换为 0 状态。

(2) 根据上述分析，在同步时序电路状态转换表.xlsx 文件状态转换表中填入对应数据，如下图。

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	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 2-13 BCD 计数器状态转换表

(3) 由上述表格可获得 N3、N2、N1、N0 的输出表达式，填入 logisim 分析电路表达式表格中，自动生成电路。

2.13 BCD 计数器输出函数模块设计

- 1.功能：实现输出函数即进位 $Cout=(Q=9)?1:0$ 的电路。
- 2.输入：4 位现态 S3S2S1S0。
- 3.输出：进位输出 Cout。
- 4.设计思路：

(1) 由分析易知,现态为 0~8 时输出为 0,现态为 9 时输出为 1。

(2) 由上述分析，在同步时序电路状态转换表.xlsx 文件输出函数真值表中填入对

应数据，如下图。

当前状态(现态)					输入信号								输出
S3	S2	S1	S0	现态 10进制	In1	In2	In3	In4	In5	In6	In7	In8	Cout
0	0	0	0	0									
0	0	0	1	1									
0	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 2-13 BCD 计数器输出函数真值表

(3)由上述表格可获得 Cout 的输出表达式，填入 logisim 分析电路表达式表格中，自动生成电路如下图。

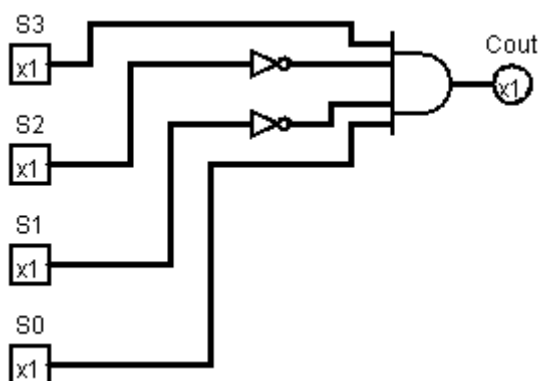


图 2-14 BCD 输出函数电路

2.14 码表计数器模块设计

1.功能：利用 4 个 BCD 码计数器显示码表计数，分别对应 10 秒，1 秒，1/10 秒，1/100 秒。

2.输入：时钟信号 Clk,使能信号 En,异步复位 Rst。

3.输出：16 位输出 Q，用于控制显示码表计数值

4.要求：

(1)利用 4 个 4 位的 BCD 码计数器级联而成，每个计数器在 0-9 之间变化，当最低位计数器从 9 变化为 0 时，高位计数器加 1；

- (2)当低两位计数器变化到 99 时，下一个时钟到来时，高一位的计数器加 1；
- (3)当低三位计数器变化到 9.99 时，下一个时钟到来时，最高一位的计数器加 1；
- (4)当 Rst=1 时，计数器清零。

5.设计思路：

(1) 有上述要求易知，16 位输出需要使用 4 个 4 位 BCD 计数器，4 个计数器接共同的时钟端 CLK、复位端 Rst 和使能端 En，输出由分线器合并为 16 位输出 Q。

(2) 分析易知，需要当低位计数器满足最大计数值（即 0.09、0.99、9.99）时，再来一个时钟脉冲，低位计数器变为 0，高位计数器计数加一。而由于计数器在 9 的时候输出进位信号 Cout=1，因此不能将进位信号直接当作高位计数器的时钟信号而是作为启动高位计数器的使能信号，这样有进位输出时，高位 BCD 计数器置为工作状态，下一个时钟信号到来时高位计数器可计数，同时低位计数器状态转为 0，无进位信号关闭高位计数器的工作状态。

(3) 由上述分析可知，码表的十位、个位、十分位对应的计数器的使能端需要使用与门，与门的输入除了模块本身的使能端 En，还包括低位的进位信号。构建如下图电路。

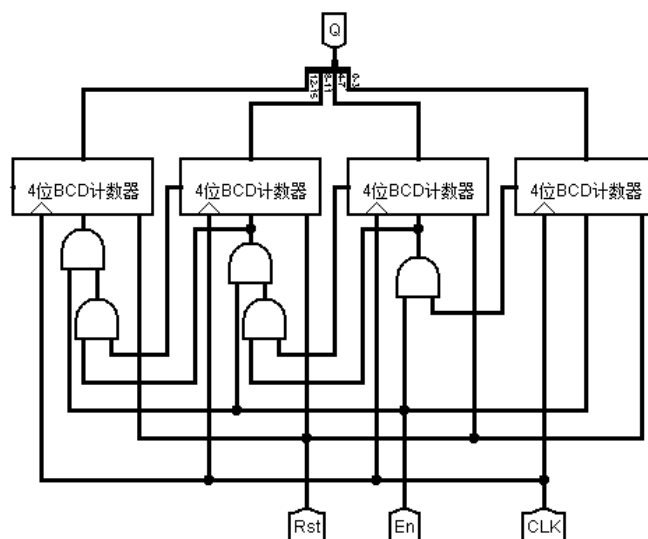
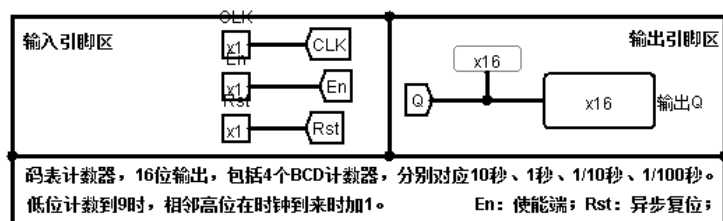


图 2-15 码表计数器电路

2.15 码表控制器模块设计

- 1.功能：实现输入外部控制信号，产生相应的内部控制信号以控制码表各模块。
- 2.输入：外部控制信号：开始计数信号 Start、停止计数信号 Stop、复位信号 Reset、更新计数信号 Store，新纪录信号 NewRecord 和时钟信号 CLK。
- 3.输出：内部控制信号：计数器使能信号 TM-EN，计数器复位信号 TM-Reset、99.99 和当前计时选择信号 SD-SEL，16 位寄存器使能信号 SD-EN，数码管显示选择信号 DP-SEL。
- 4.设计思路：

（1）码表控制器状态转换和输出函数模块见 2.16 和 2.17，此处直接使用二者构建好的封装模块。

（2）由分析可知，可用 4 个 D 触发器进行状态控制，当触发器处于工作状态，来一时钟信号，使电路的状态发生改变。同时由要求易知，电路的次态由现态和输入同时决定，而输出函数的信号由当前状态决定。

（3）给电路状态进行编码：0 为复位状态，1 为开始计数状态，2 为停止计数状态，3 为有新纪录更新记录状态，4 为无新纪录不更新记录状态，5 位持续计数状态。

（4）由上述分析可构建电路如下图。

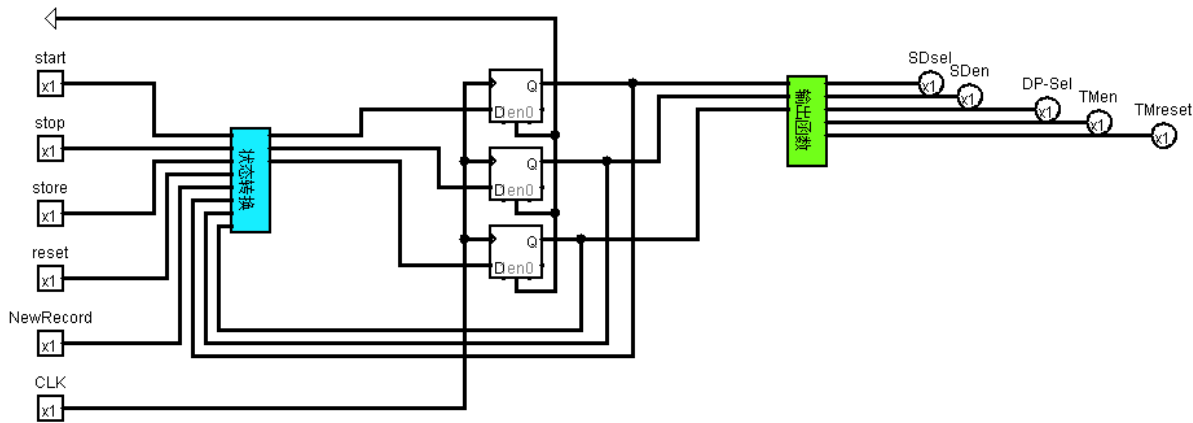


图 2-16 码表控制器电路

2.16 码表状态转换模块设计

- 1.功能：实现电路现态到次态的状态转换。
- 2.输入：外部控制信号：开始计数信号 Start、停止计数信号 Stop、复位信号 Reset、更新计数信号 Store，新纪录信号 NewRecord。当前状态 S3S2S1S0。

3.输出：下一状态：N3N2N1N0。

4.设计思路：

(1) 由 2.15 中的状态编码方式，依据码表需求构建状态转移关系如下：

复位状态 0: start=1 ——》开始计时状态 1; store=1 ——》无新纪录不更新记录状态 4; 其他情况保持不变。

开始计数状态 1 ——》持续计数状态 5

停止计数状态 2: start=1 ——》开始计时状态 1; store=1&&NewRecord=1 ——》有新纪录更新记录状态 3; store=1&&NewRecord=0 ——》无新纪录不更新记录状态 4; reset=1 ——》复位状态 0; 其他情况保持不变。

有新纪录更新记录状态 3: start=1 ——》开始计时状态 1; reset=1 ——》复位状态 0; 其他情况保持不变。

无新纪录不更新记录状态 4: start=1 ——》开始计时状态 1; reset=1 ——》复位状态 0; 其他情况保持不变。

持续计数状态 5: start=1 ——》开始计时状态 1; stop=1 ——》停止计数状态 2; store=1&&NewRecord=1 ——》有新纪录更新记录状态 3; store=1&&NewRecord=0 ——》无新纪录不更新记录状态 4; reset=1 ——》复位状态 0; 其他情况保持不变。

(2) 有上述状态转移关系，在同步时序电路状态转换表.xlsx 文件状态转换表中填入对应数据，如下图。

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	1	1	3	1	0	0	0					1	0	0	0	1
0	0	1	1	3	0	0	0	1					0	0	0	0	0
0	0	1	1	3	0			0					3	0	0	1	1
0	1	0	0	4	1	0	0	0					1	0	0	0	1
0	1	0	0	4	0			1					0	0	0	0	0
0	1	0	0	4	0			0					4	0	1	0	0
0	1	0	1	5	0	0	0	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	1	0	0					2	0	0	1	0
0	1	0	1	5	0	0	1	0	1				3	0	0	1	1
0	1	0	1	5	0	0	1	0	0				4	0	1	0	0
0	1	0	1	5	0	0	0	1					0	0	0	0	0

图 2-17 码表状态转化表

(3) 由上述表格可获得 N3、N2、N1、N0 的输出表达式，填入 logisim 分析电路

表达式表格中，自动生成电路。

2.17 码表控制器输出函数模块设计

1.功能：实现码表内部信号的输出。

2.输入：当前状态 S3S2S1S0。

3.输出：内部控制信号：计数器使能信号 TMEN，计数器复位信号 TMReset、99.99 和当前计时选择信号 SDSel, 16 位寄存器使能信号 SDEN, 数码管显示选择信号 DPSEL。

4.设计思路：

(1) 由 2.15 中的状态编码方式，依据码表需求可得各状态输出的内部控制信号：

复位状态 0：SDSel=SDEN=1；开始计数状态 1：TMReset=1；停止计数状态 2：无输出信号；有新纪录更新记录状态 3：SDEN=DPSEL=1；无新纪录不更新记录状态 4：DPSEL；持续计数状态：TMEN=1。

(2) 由上述分析，在同步时序电路状态转换表.xlsx 文件输出函数真值表中填入对应数据，如下图。

当前状态(现态)						输入信号												
S3	S2	S1	S0	现态 10进制		start	stop	store	reset	NewRecord	In6	In7	In8	SDSel	SDEN	DPSEL	TMEN	TMReset
0	0	0	1	1														1
0	0	1	0	2														
0	0	1	1	3											1	1		
0	1	0	0	4												1		
0	1	0	1	5													1	
0	0	0	0	0										1	1			1

图 2-18 码表输出函数真值表

(3) 由上述表格可获得内部信号的输出表达式，填入 logisim 分析电路表达式表格中，自动生成电路。

2.18 运动码表模块设计

1.功能：实现运动码表整体功能。

2.输入：外部控制信号：开始计数信号 Start、停止计数信号 Stop、复位信号 Reset、更新计数信号 Store 和时钟信号 CLK。

3.要求：

(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。

4.设计思路：

(1) 有要求易知：TM-Reset 接计时器接复位端,TM-EN 接计时器使能端,SD-EN 接 16 位寄存器使能端,CLK 接计时器 16 位寄存器时钟端,码表显示模块输出接 DispInfo。

(2) 由分析易知，16 位寄存器的输入可能是复位的 99.99，可能是最新记录，因此需要一个 2 路选择器对数据进行选择，2 路选择器控制端为 SD-Sel，输出即为寄存器的输入。

(3) 内部信号新纪录 NewRecord 有 16 位无符号比较器获得，比较器 X 接寄存器存储的记录，Y 接当前计时，若输出 Great=1，则证明当前计时小于之前记录，达到了新纪录，NewRecord=1。

(4) 由分析知，码表显示可能为当前计数值，也可能在无新纪录时显示寄存器内的记录，因此需要一个 2 路选择器，控制端为 DP-Sel，输出即为码表显示模块的输入。

(5) 由上述分析即可构建电路如下图。

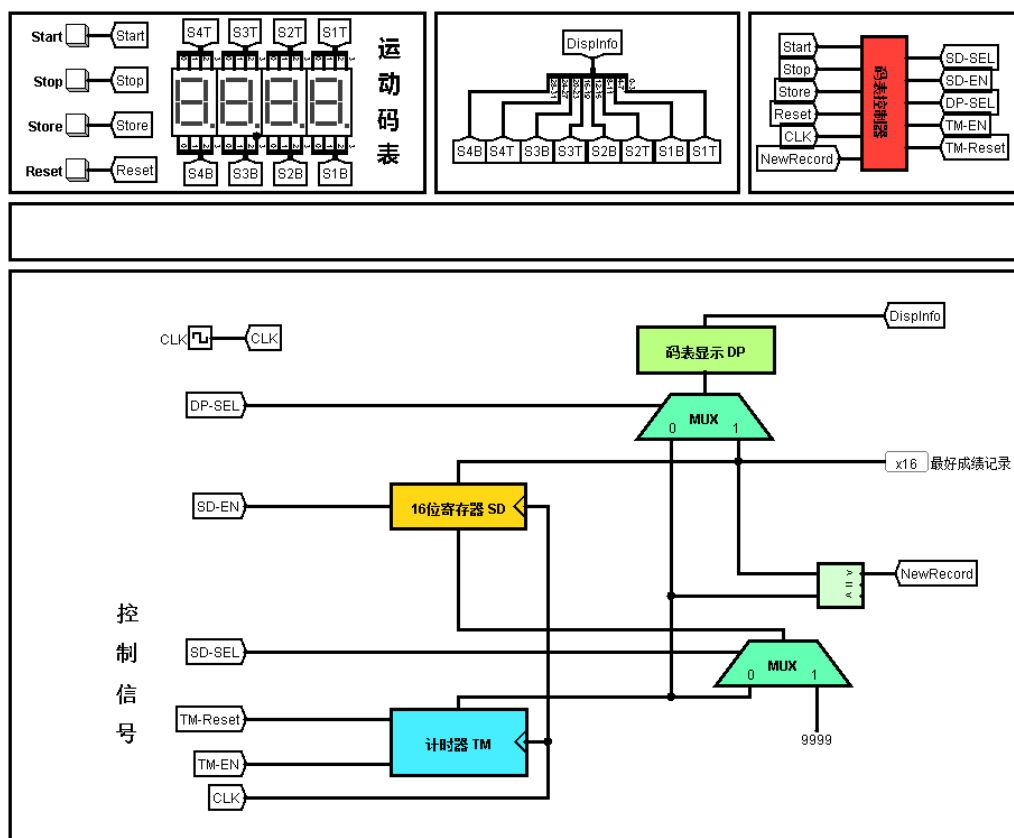


图 2-19 运动码表电路

3 测试及故障调试

3.1 测试

1.LED 计数电路测试

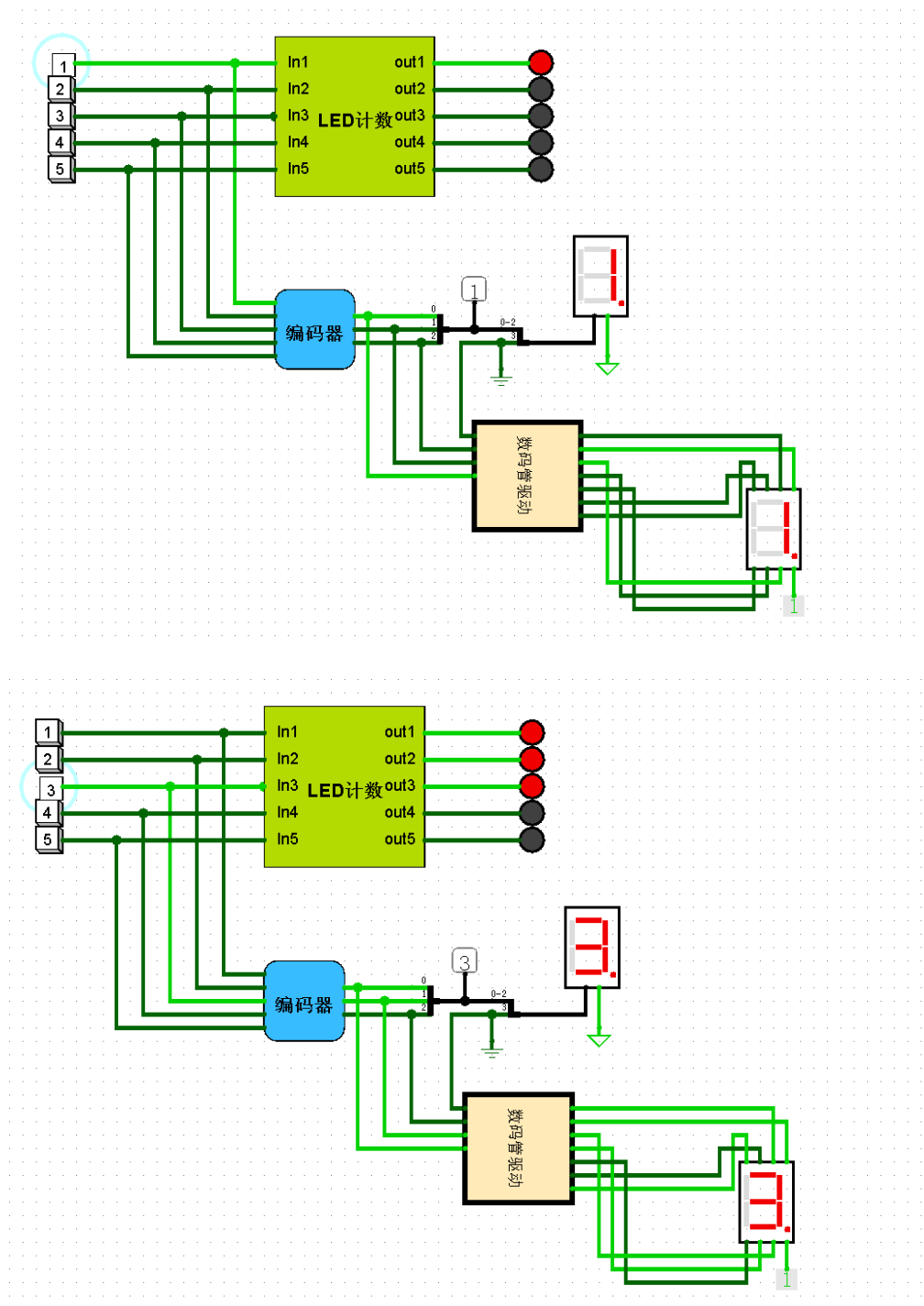


图 3-1 LED 计数电路测试图

2.2 路选择器测试

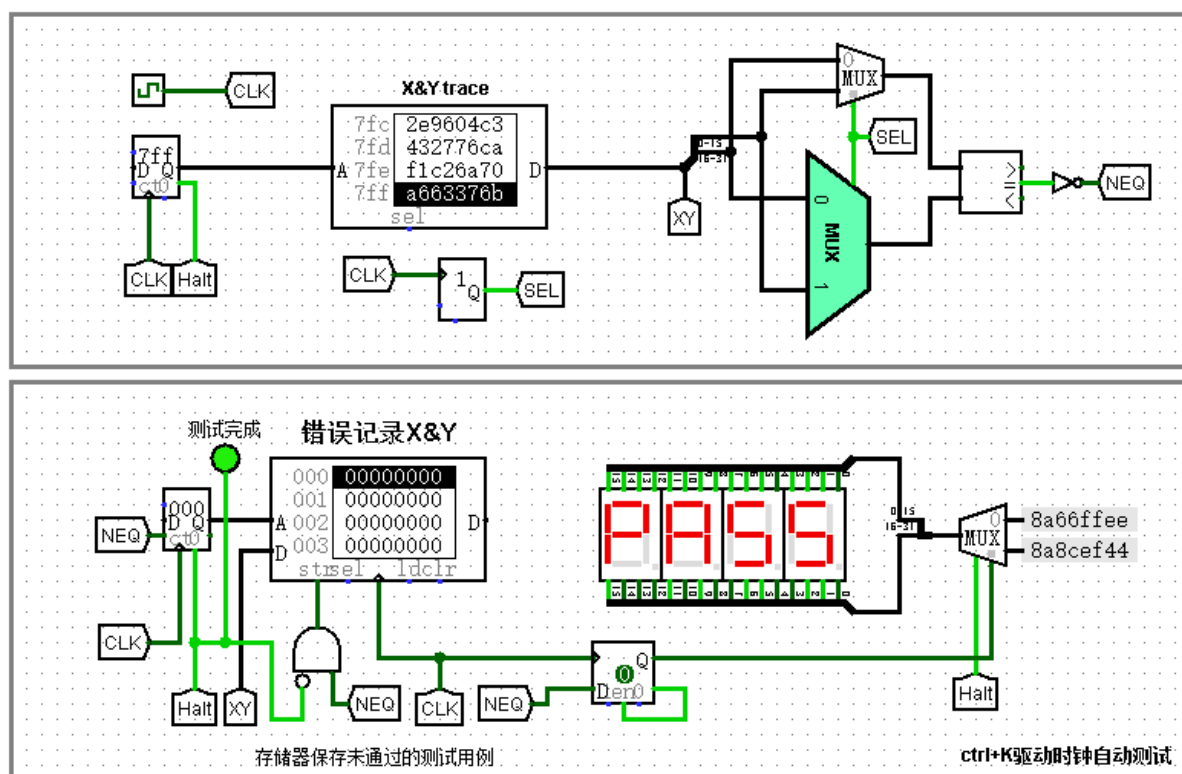


图 3-2 2 路选择器测试图

3.16 位无符号比较器测试

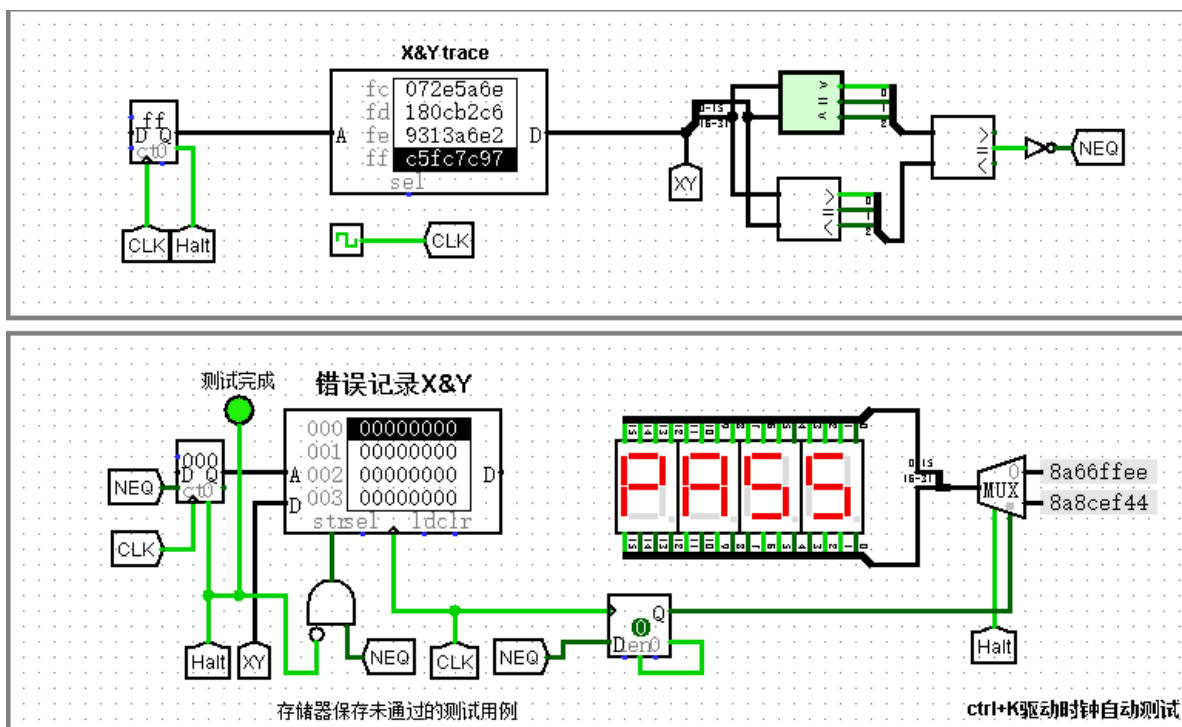


图 3-3 16 位无符号比较器测试图

4.码表数码管显示测试

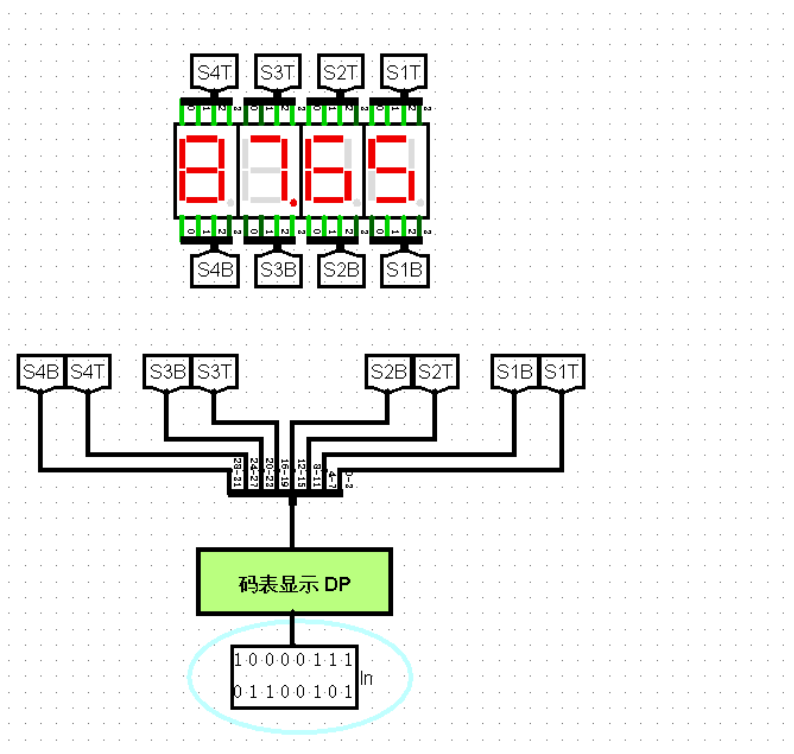
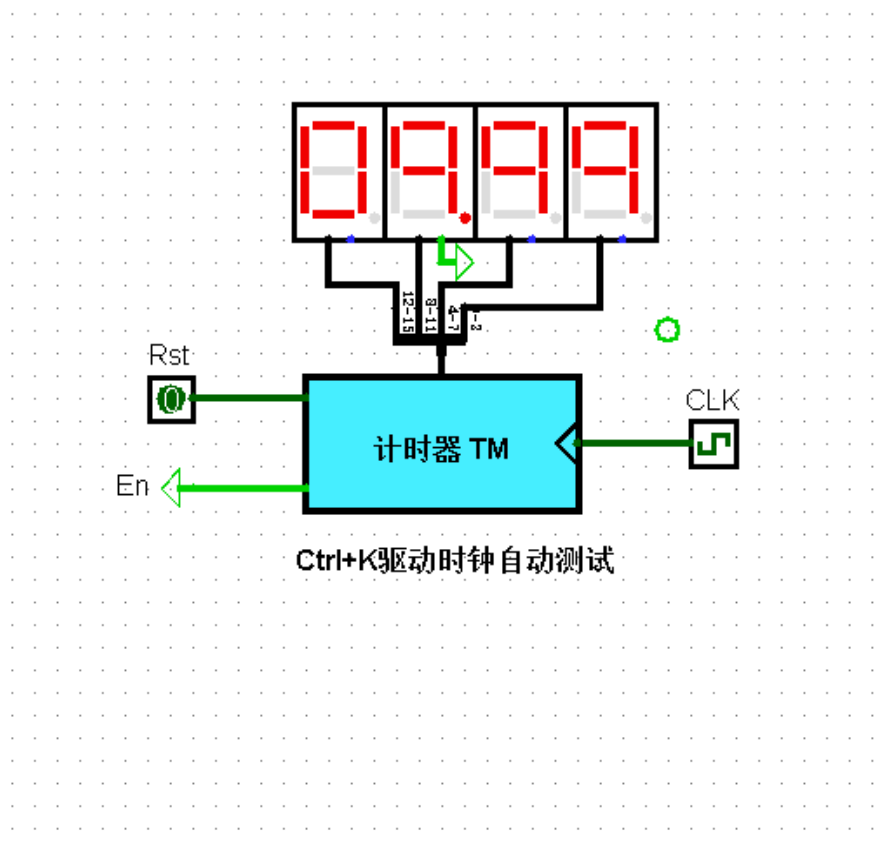


图 3-4 码表数码管显示图

5.码表计数器测试



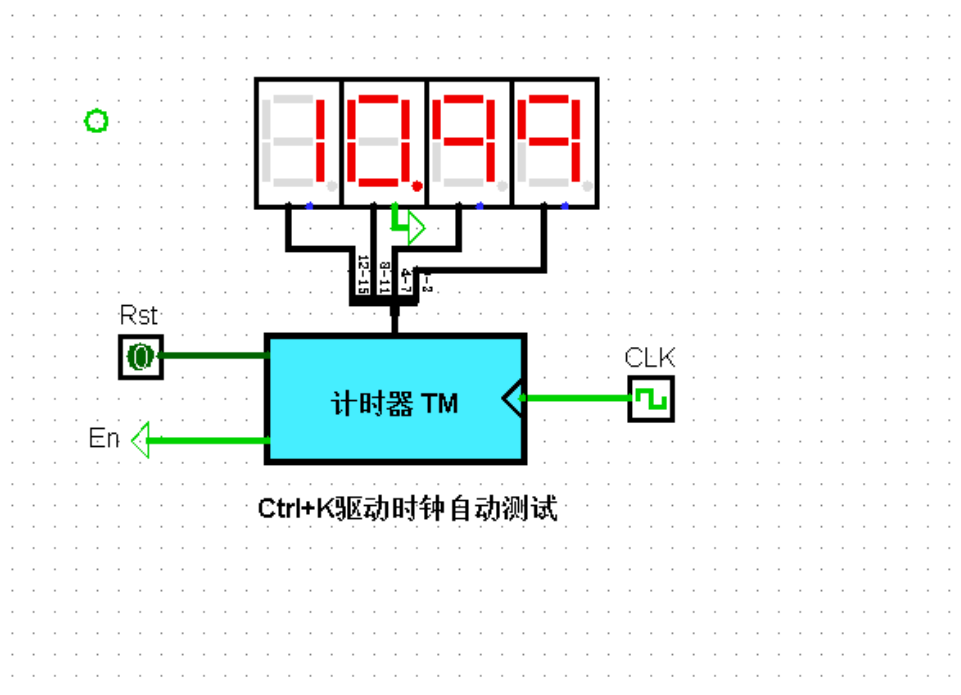


图 3-5 码表计数器测试图

6. 运动码表功能测试:

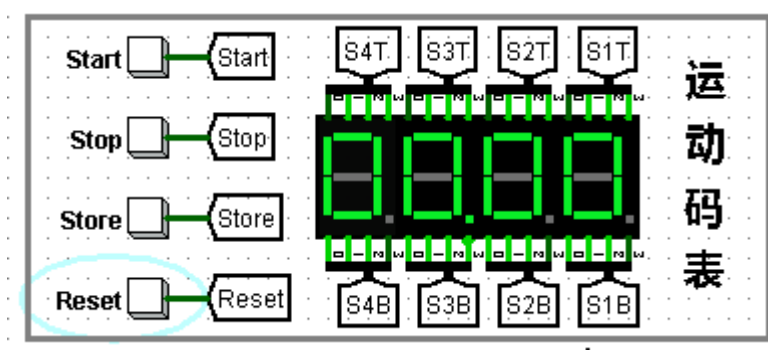


图 3-6 运动码表功能测试图 (gif 图)

3.2 遇到的问题及处理

1. 问题: 7 段数码管驱动测试并未成功, 实际显示与理论不对应。

处理: 个人设计的数码管显示的引脚对应方式和文件中自带的引脚连接方式不同, 最后根据文件引脚连接方式重新填写了 7 段数码管的真值表, 并构建了相应数码管驱动电路。

2. 问题: 探针值为负数。

处理: 在属性中把进制改为无符号十进制。

3. 问题: 16 位无符号比较器输出有错, 不能通过测试。

处理:通过仔细分析电路,最后发现输出的 Less 一个是用或非门链接 Great 和 Equal, 错用成了与非门, 应加强电路设计的细心。

4.问题: 码表计数器低位向高位进位时高位在低位是“9”时就已经加 1。

处理: 问题导致的原因在于低位计数器在状态为“9”时产生了向高位的输出, 若直接将进位输出接到高位计数器的时钟端, 则在“9”时高位计数器就开始计数。因此电路设计上存在问题, 后来经过分析, 选择将进位输出接到高位的使能端, 从而解决问题 (具体见 2.14)。

3.3 设计方案存在的不足

1.最后运动码表控制器的状态转移关系较为复杂, 可能存在冗余的转移关系, 可能有更简单的表达式以及相应的状态转换电路。

2.运动码表的功能还可以进一步优化完善。比如, 在停止计时或查看记录状态时选择按 start, 不是重新从 0 计时而是继续计时。

4 设计总结与心得

4.1 实验总结

本次实验一共分了 4 次进行，围绕数字码表进行相应的基本部件的电路设计，并最终完成能够实现开始计时、停止计时、更新记录和时间复位等功能的运动码表的整体设计和构建。基本掌握了 logisim 软件的使用方法，尤其是其电路仿真、分析电路自动生成电路的方法。熟悉了数字系统基本特征及设计方法，熟练掌握了数字系统设计基本流程。

4.2 实验心得及建议

1.实验心得：

这 4 次实验通过自己的努力最终完成了运动码表的设计和构建，很有成就感。在实验的过程中也遇到了很多问题，大部分都是由于分析电路不充分、设计电路不细心导致的。因此整个过程下来，也让我加深了对组合逻辑电路、时序逻辑电路以及数字系统的设计方法和步骤，感受到了真是的设计与做题的差异。对电路状态编码状态转换有了更深的认识，提高了我的电路设计能力和实验操作能力。整个过程下来很有收获。

2.建议：

希望老师对实验的难点有所提升，有时自己做可能有些无从下手。

参考文献

- [1] DAVID A. PATTERSON(美). 计算机组成与设计硬件/软件接口(原书第 4 版). 北京: 机械工业出版社.

一、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：黄浩山