

Digital Circuits and Systems*

Lab Report IV^{L^AT_EX}

* Teacher: Song Zhou. TA: Yun Xu (助教哥抱歉前几次报告这里好像搞错了)

实验四时序器件实验

张逸凯 171840708 (转专业到计科, 非重修)

Department of Computer Science and Technology

Nanjing University

zykhelloha@gmail.com

张逸凯 171840708

一、实验目的

1. 掌握常见时序器件的逻辑功能和使用方法。
2. 掌握时序器件的级联扩展的方法。
3. 掌握使用时序器件实现数字系统设计的步骤。

二、实验设备与器材

1. 数字逻辑电路实验箱。
2. 芯片
74HC00 四路两输入与非门 1 片
74HC02 四路两输入或非门 1 片
74HC74 双 D 触发器 2 片
74HC161 四位二进制异步清零计数器 1 片
74HC163 四位二进制同步清零计数器 1 片
74LS194 双向移位寄存器 2 片

三、实验内容及实验步骤

(一)

1. 分别利用 1 片 74HC161 清零端加一个逻辑门电路设计并实现 0, 1, ..., 11 模 12 的计数器; 以及 1 片利用 74HC163 的置数端加一个逻辑门电路, 设计并实现 3, 4, 5, ..., 14 模 12 的计数器, 分别将输出连接到一个 7 段数码管显示。

(1) 设计步骤:

方法 1(利用清零端实现): 74HC161 是四位二进制可预置的异步加法计数器, 设计模 12 的计数器, $12=8+4$, 可以将 Q2Q3 接入二输入与非门, 结果接入清零端即可实现。

方法 2(利用置数端实现): 74HC163 是四位二进制同步加法计数器, 从 3 开始, 容易想到利用输入端预置初始值, 再将 Q3Q2Q1 接入三输入与非门, 结果接入清零端即可实现

(2) 状态转移表:

现态	次态
0000	0001
0001	0010
0010	0011
0011	0100
0100	0101
0101	0110
0110	0111
0111	1000
1000	1001
1001	1010
1010	1011
1011	0000
Q3Q2Q1Q0	$Q3 \cdot Q2 \cdot Q1 \cdot Q0$

现态	次态
0011	0100
0100	0101
0101	0110
0110	0111
0111	1000
1000	1001
1001	1010
1010	1011
1011	1100
1100	1101
1101	1110
1110	0011
Q3Q2Q1Q0	$Q3 \cdot Q2 \cdot Q1 \cdot Q0$

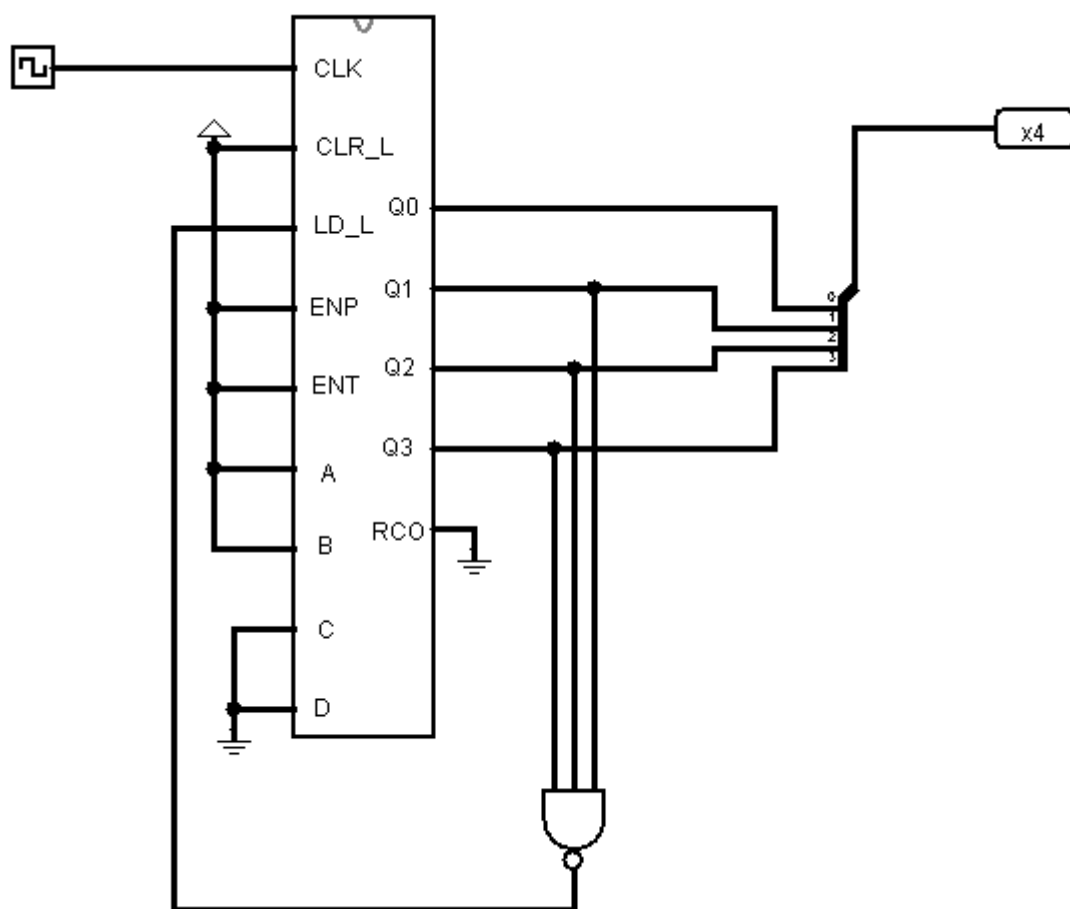
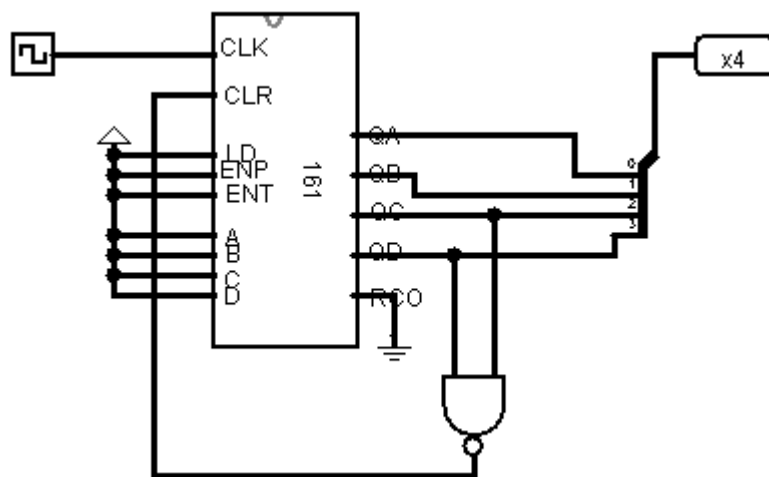
(3) 逻辑表达式:

可以合理利用清零端的终止位的特点. 注意这里的区别就是 161 是同步清零的, 163 是异步清零的.

$$161 \text{ 清零端} = (Q3 \cdot Q1)'$$

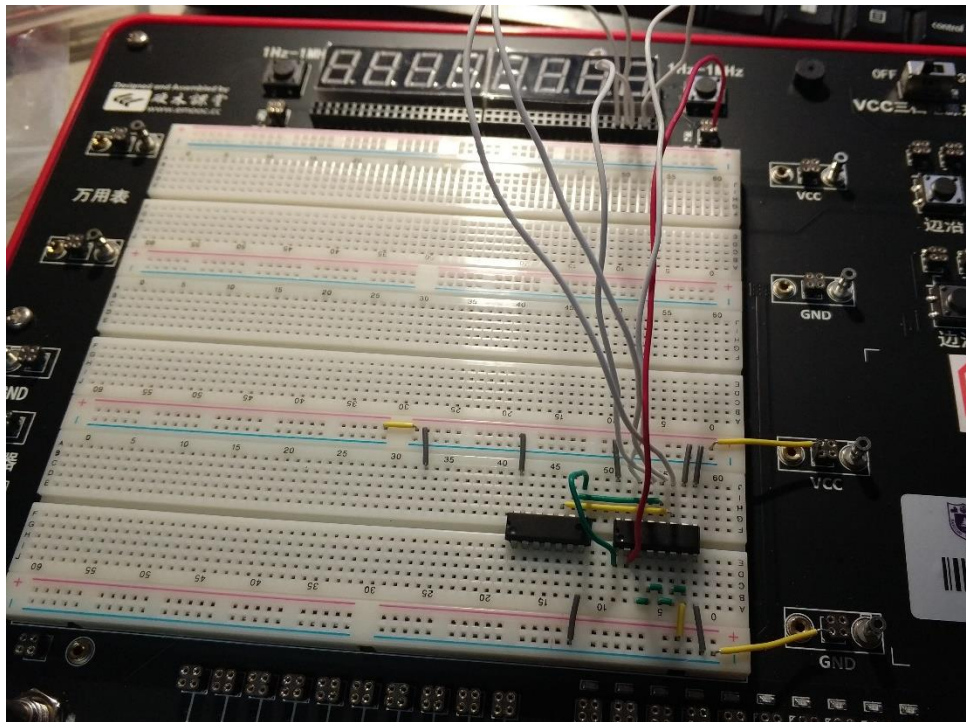
$$163 \text{ 清零端} = (Q3 \cdot Q2 \cdot Q1)'$$

(4) Logisim 电路图:

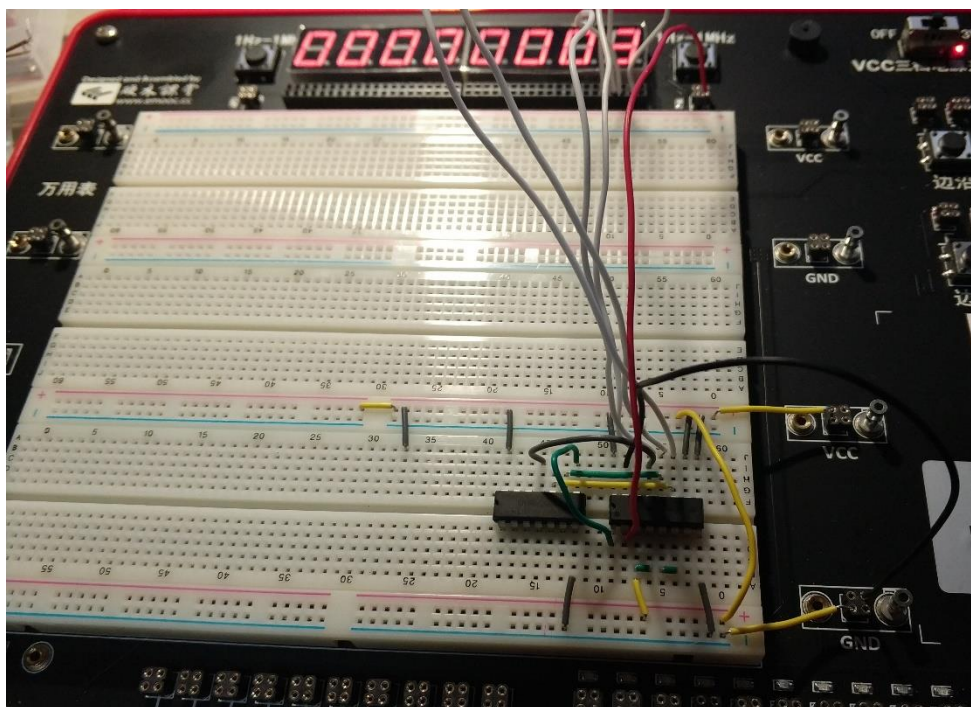


(5)实验照片：

74HC161:



74HC163:



实验验证结果正确.

(二) (已验收, 但是因为 708 太多了调不到没看到清零, 附件有 logisim 电路模拟文件, 可以看到是可以清零的.)

1. 利用 3 片 74HC163 (74HC161) 及少量逻辑门电路, 设计自己学号后 3 位 (如果后 3 位学号小于 100 的, 则加上 100 后, 进行计数) 的 BCD 加法计数器, 输入 1Hz 的连续脉冲累加计数, 并将输出连接到三个 7 段数码管显示。

2.设计步骤:

由于有 3 片 74HC163, 所以可以每片 163 代表三位数中的一位, 进位就利用清零端进行进位, 只有当低位要进位时, 高位才开始计数。或者可以设计成异步的, 低位的进位端接到高位的时钟端, 但是这样会产生一些问题在异步处理时达到学号之后只能使用清零端实现, 但是注意清零端也是每一个 163 发生进位时的输入, 所以在进位时进位产生的那个脉冲也会输入到 708 要接出来的那个与门, 这里就产生冲突了, 再加一个或门可以解决, 但是这样做的话还不如使用置数端, 放一个全 0 的数进行置数清零. 逻辑门可以更少.

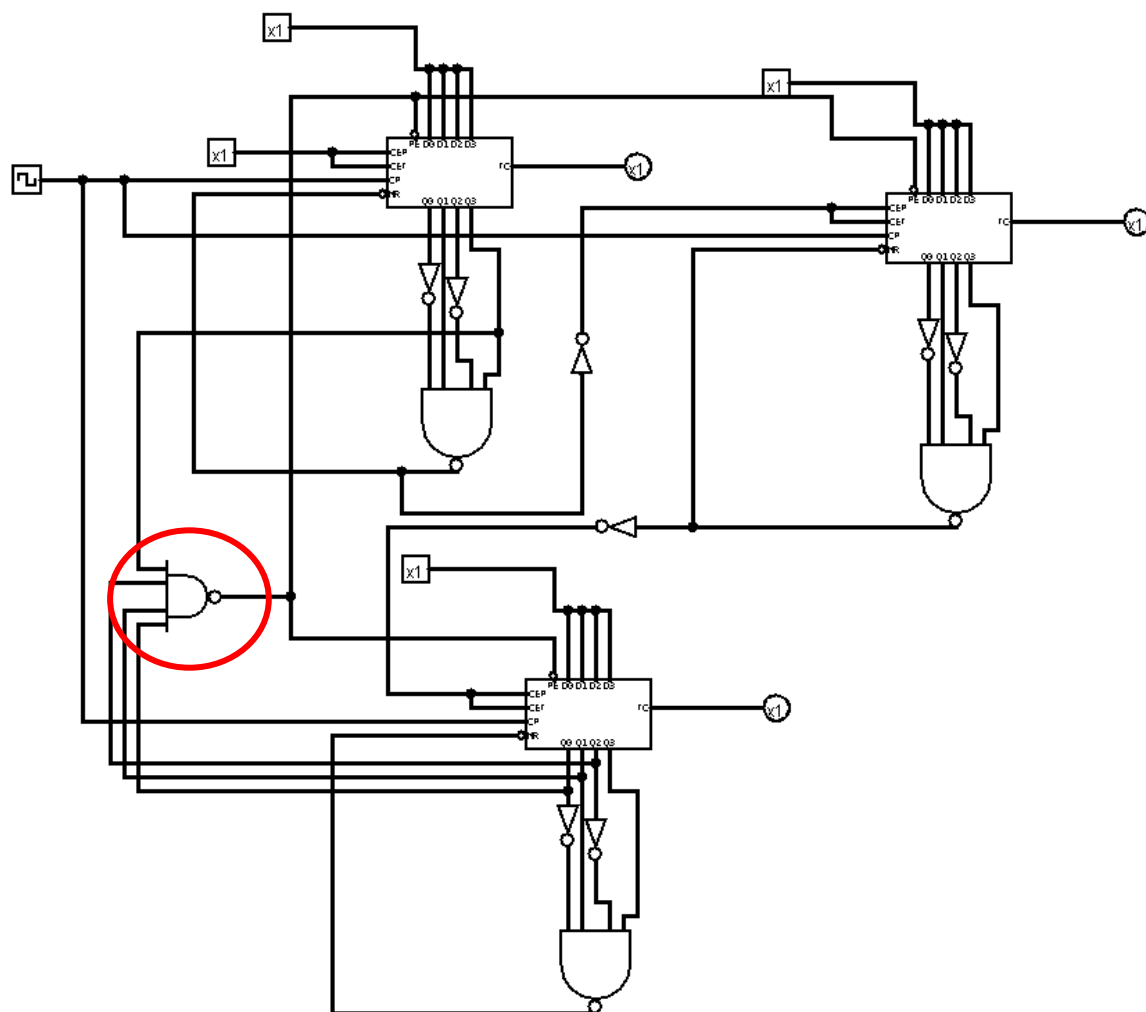
注意这里可以合理利用学号位 708 => 0111 0000 1000, 这样接入一个四输入的与非门就可以实现了.

3.状态转移表:

状态太多 (708 个), 列表列不下, 故在此做简单说明
第三位从 0~9, 第二位从 0~9, 第一位从 0~7, 当 3 位数达到 708 时, 次态为 000

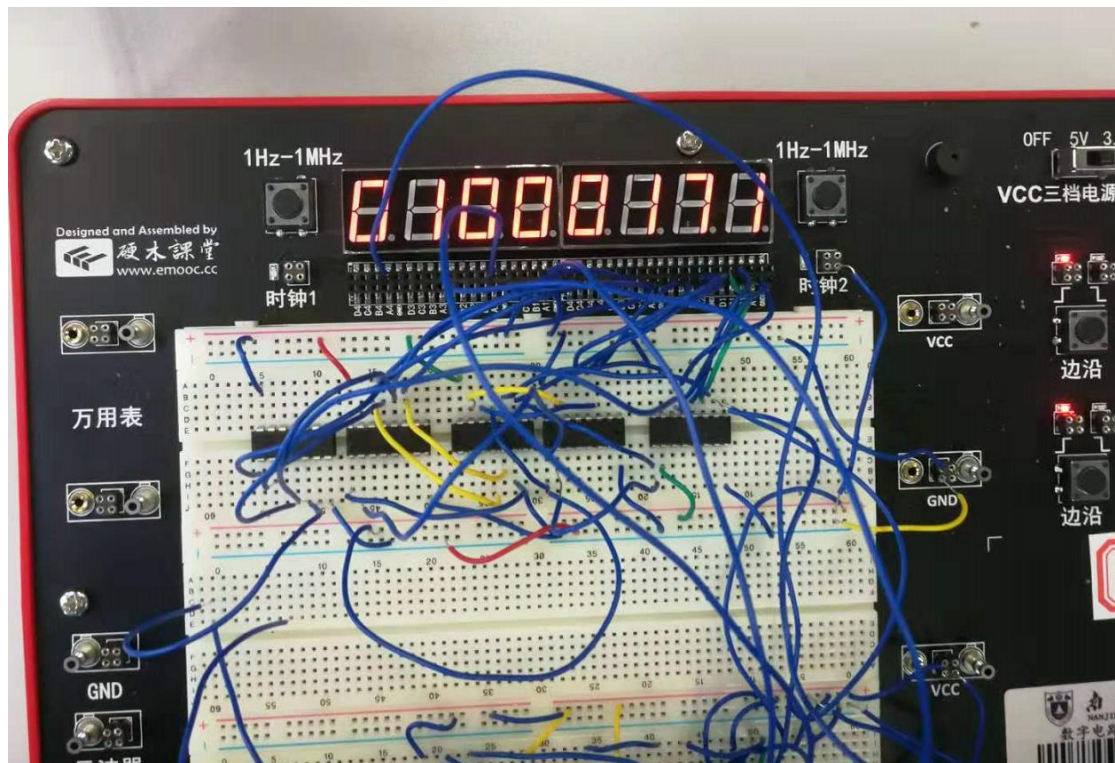
状态	次态
00X(X <=8)	00X+1
009	010
0AB(A, B <=9; A > 0)	0AB+1
099	100
DEF(DEF<=708)	DEF+1
708	000(开始新的周期)

4.电路原理图（由 Logisim 导出所以是黑色的，附件 lab4_2.circ 里有）：

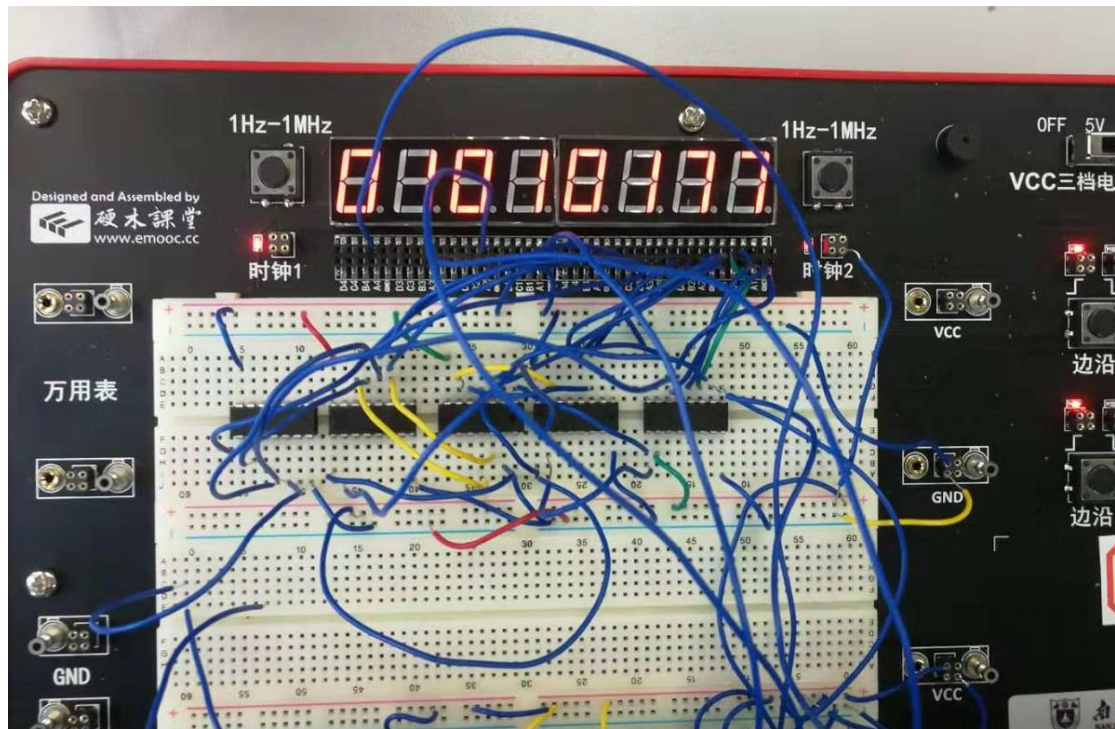


这里每一个封装好的 163 的输出(从左到右是从低位到高位)就是需要接到七段数码管上显示的内容, 由模拟文件(见附件 lab4_2.circ)可以发现 708 时会发生置数 0000 来清零. 就是一个置数的与非门逻辑, 为了防止上面说的可能出现的冲突问题(见图中画圈部分).

5.实验照片：



这里前四位是之后的线性反馈移位计数器，后四位是模学号计数器。



(三)

1.利用一片 74LS194、74HC86 和 74HC02，利用 **74LS194 左移功能**，实现一种 **4 位的包含全 0 状态的线性反馈移位计数器 LSFR**。观察输出端的状态变化，将结果记录下来，并连接到 7 段数码管显示。

设计步骤：

74LS194 可以进行左移，利用课件中讲的设计步骤，列出状态转移表，通过卡诺图化简，结合所给的 2 输入异或门和 2 输入或非门，可以容易实现四位的 **LSFR**，具体实现方法见下面的电路原理图。反馈方程是 $LIN=((Q0+Q1)+(Q1+Q2))' \oplus (Q2 \oplus Q3)$ 这是下面卡诺图化简之后的结果。

状态转移表：

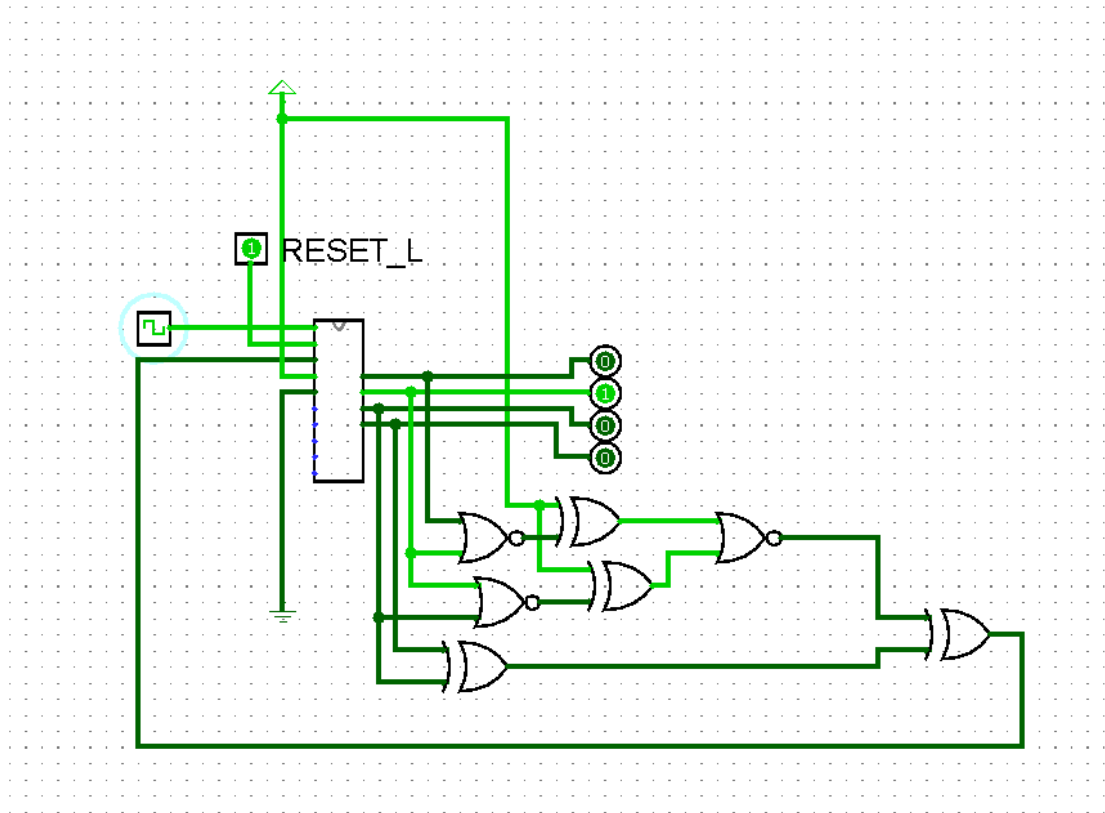
Q3Q2Q1Q0	Q3*Q2*Q1*Q0*	LIN
0000	0001	1
0001	0010	0
0010	0100	0
0100	1001	1
1001	0011	1
0011	0110	0
0110	1101	1
1101	1010	0
1010	0101	1
0101	1011	1
1011	0111	1
0111	1111	1
1111	1110	0
1110	1100	0
1100	1000	0
1000	0000	0

4.逻辑表达式：

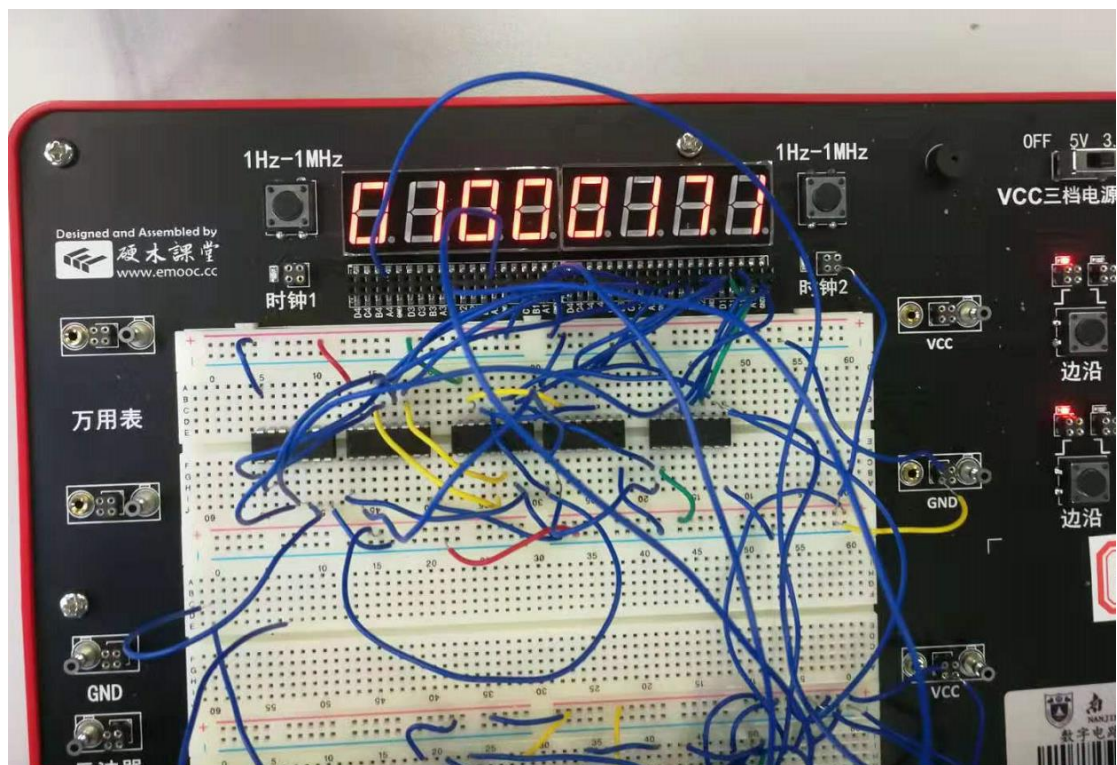
卡诺图化简: $LIN=((Q0+Q1)+(Q1+Q2))' \oplus (Q2 \oplus Q3)$

这个表达式已经满足只使用异或和或非的要求了，因为其中看似是或门的可以用异或实现，异或门的一个输入接到 1 就构成一个非门了！

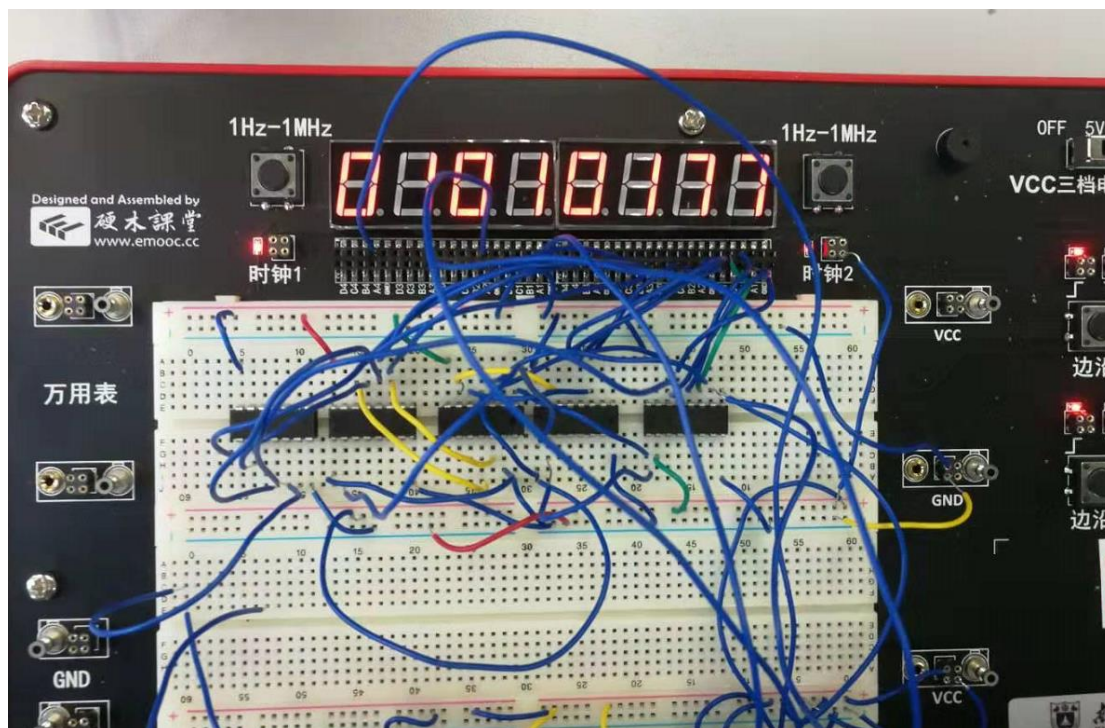
5.电路原理图：lab4_3.circ



6.实验照片：



这里前四位是之后的线性反馈移位计数器，后四位是模16计数器。



(四)

1.利用 74LS194 左移功能和少量门电路，完成二进制序列“1000111101”的循环生成。

2.设计步骤：

按照课件上给出的设计方法，画出状态图，列出状态转移表，通过卡诺图法化简得到 LIN 的逻辑表达式，

3.状态转移表：

Q3Q2Q1Q0	Q3*Q2*Q1*Q0*	LIN
1000	0001	1
0001	0011	1
0011	0111	1
0111	1111	1
1111	1110	0
1110	1101	1
1101	1011	1
1011	0110	0
0110	1100	0
1100	1000	0

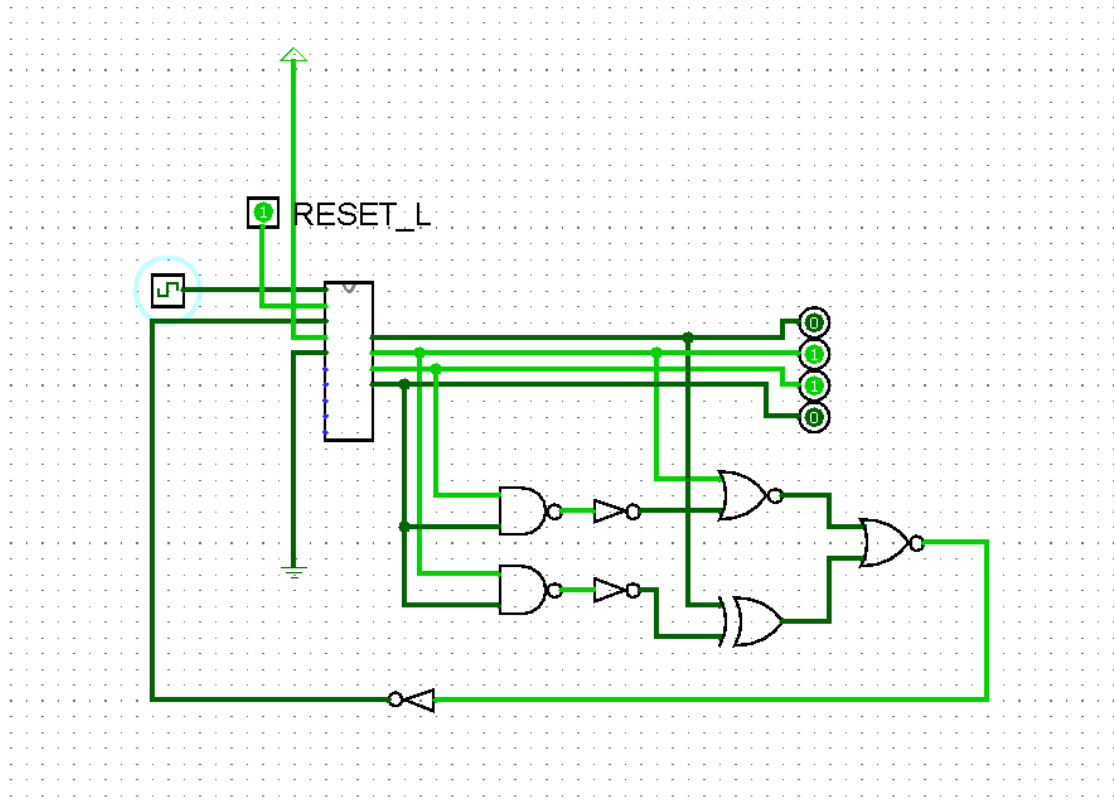
4.逻辑表达式：

经过卡诺图法化简后得到(化简过程不难，这里略)：

$$LIN = Q3' * Q0 + Q3' * Q1' + Q1' * Q0 + Q3 * Q2' * Q1' + Q3 * Q1 * Q0'$$

$$= (Q3 * Q1) \oplus Q0 + (Q1 + (Q2 * Q3))'$$

5.电路原理图：



四、思考题：

1.比较反馈清零法和反馈置数法的异同：

同：1.都是低电平有效，2.都是在时钟上升沿有效

清零法就是遇到进位(或者说是状态的循环最终状态)，清零输入端为有效电平，无论其他输入端是什么状态，片内所有触发器状态都置0；

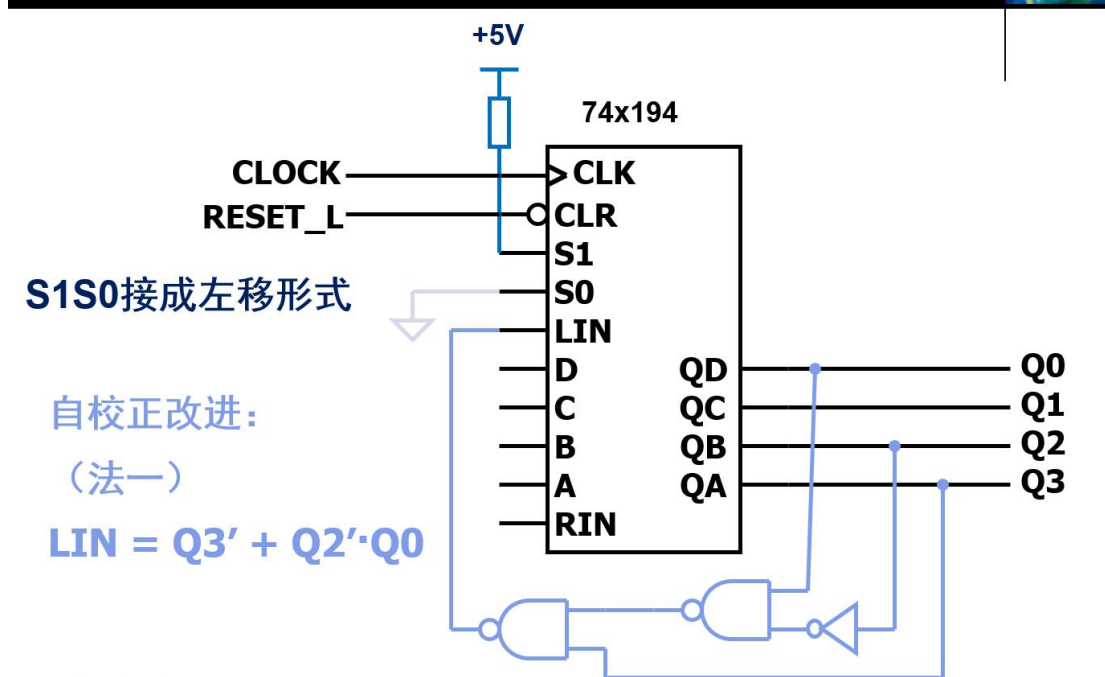
置数法不同的是，它有一个预置数据，当它为有效电平时，并不是清零（当然也可以清零，当预置数据全为0时），而是被置入预置数据。有时预置数据也与输出端连，置数的时候该位的数值与前一个状态的输出端（与置数端对应的）一样，其他各个位的数值与预置数据相同。置数端实现时状态的终止端和清零端略有不同，比如常常是1111时开始准备置数。

2.总结利用计数器实现任意进制计数器的方法：

先观察是否所需计数上限超过计数器本身设计，若未超过，则只需要取到合理的模即可。若超过则采用级连的办法，而级连又有几种方法可根据需要选择(同步和异步)：时钟端一致则可以通过控制每位进位来使不同位什么时候有效来实现，如果时钟端不一致时不仅要考虑异步问题，还要注意在低位进位给高位时钟端，而无论是否计数超过计数器设计上限，均需考虑同步和异步清零与置数问题。

注意这里模多少指的只是状态数有多少，在这些状态里面循环就可以。

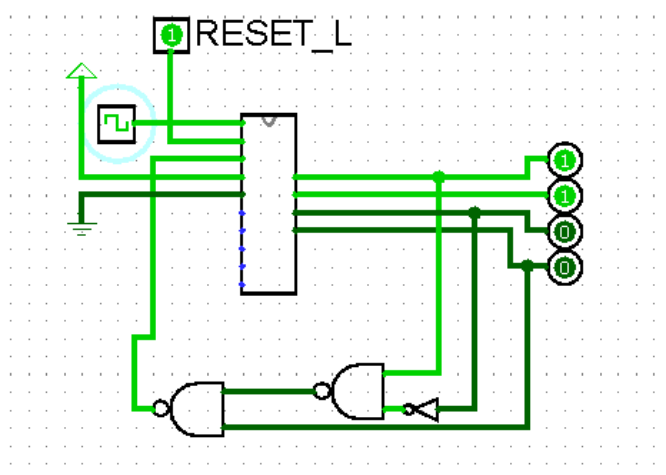
正如 PPT 以及老师讲的所示:



21

自启动其实就是需要处理所有无关状态，把那些无关状态进行“破圈”，表现在卡诺图上就是对无关项进行赋值，使那些无关的可以被破圈之后进入正常的圈（状态循环）。

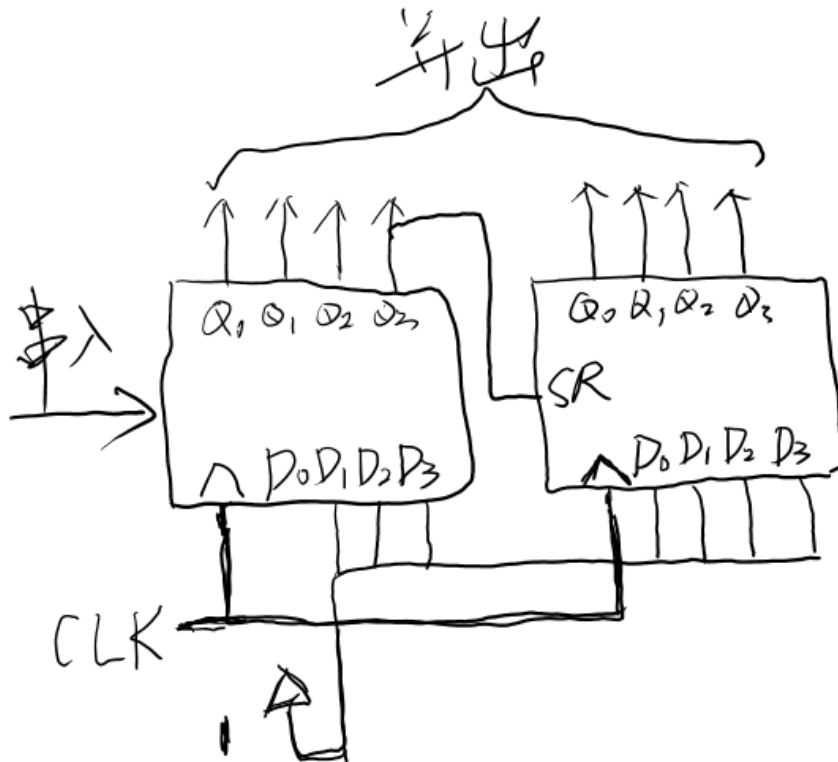
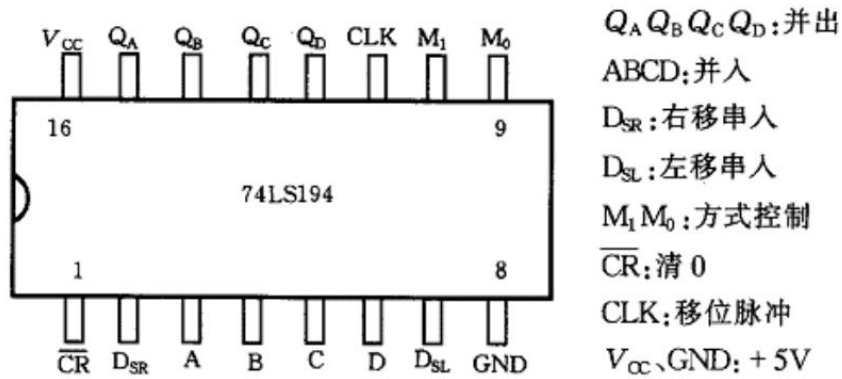
Logisim实现(经验证是正确的, 附件里有circ文件):



4. 利用 74LS194 设计实现八位二进制数数据的并行/串行转换原理图。

串行/并行转换是指串行输入的数码，经转换电路之后变换成并行输出。

首先了解 194 具体接口和结构：



电路中 S_0 端接高电平 1, S_1 受 Q_3 (第二片)控制, 二片寄存器连接成串行输入右移工作模式。 Q_3 (第二片)是转换结束标志。当 Q_3 (第二片)=1 时, S_1 为 0, 使之成为 $S_1 S_0 = 01$ 的串入右移工作方式, 当 Q_3 (第二片)=0 时, $S_1 = 1$, 有 $S_1 S_0 = 10$, 则串行送数结束, 标志着串行输入的数据已转换成并行输出了, 由下表可见, 右移操作八次之后, Q_3 (第二片)变为 0, $S_1 S_0$ 又变为 11, 说明串行输入结束。这时, 串行输入的数码已经转换成了并行输出了。当再来一个 CP 脉冲时, 电路又重新执行一次并行输入, 为第二组串行数码转换作好了准备。

串行/并行转换器状态表

CP	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	说明
0	0	0	0	0	0	0	0	0	清零
1	0	1	1	1	1	1	1	1	送数
2	d0	0	1	1	1	1	1	1	右移操作
3	d1	d0	0	1	1	1	1	1	
4	d2	d1	d0	0	1	1	1	1	
5	d3	d2	d1	d0	0	1	1	1	
6	d4	d3	d2	d1	d0	0	1	1	
7	d5	d4	d3	d2	d1	d0	0	1	
8	d6	d5	d4	d3	d2	d1	d0	0	
9	0	1	1	1	1	1	1	1	送数

并行/串行转换器状态表

CP	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	串 行 输 出						
0	0	0	0	0	0	0	0	0							
1	0	D1	D2	D3	D4	D5	D6	D7							
2	1	0	D1	D2	D3	D4	D5	D6	D7						
3	1	1	0	D1	D2	D3	D4	D5	D6	D7					
4	1	1	1	0	D1	D2	D3	D4	D5	D6	D7				
5	1	1	1	1	0	D1	D2	D3	D4	D5	D6	D7			
6	1	1	1	1	1	0	D1	D2	D3	D4	D5	D6	D7		
7	1	1	1	1	1	1	0	D1	D2	D3	D4	D5	D6	D7	
8	1	1	1	1	1	1	1	0	D1	D2	D3	D4	D5	D6	D7
9	0	D1	D2	D3	D4	D5	D6	D7							

