

实验 3：同步时序电路设计

一、实验目的

1. 学习时序逻辑电路设计的基本方法和步骤。
2. 掌握计数器和移位寄存器的设计方法。
3. 掌握数字时钟和乘法器的设计方法。
4. 掌握寄存器堆的设计方法。

二、实验环境

Logisim 2.16

三、实验内容

1、计数器实验

整体模块设计：

使用 4 个下降沿触发的 D 触发器，和相关的逻辑门实现 4 位二进制的同步计数器 CNTR4U。

输入输出引脚作用：

Clk 为共同时钟；LD 为载入引脚，载入计数起始的数字；CLR 是清零信号；ENT，ENP 为两个使能端；D0，D1，D2，D3 为一个二进制数，代表着起始计数的数字；Q0，Q1，Q2，Q3 为当前时钟的计数数字；RCO 为计数达到上限的标志。

原理图：

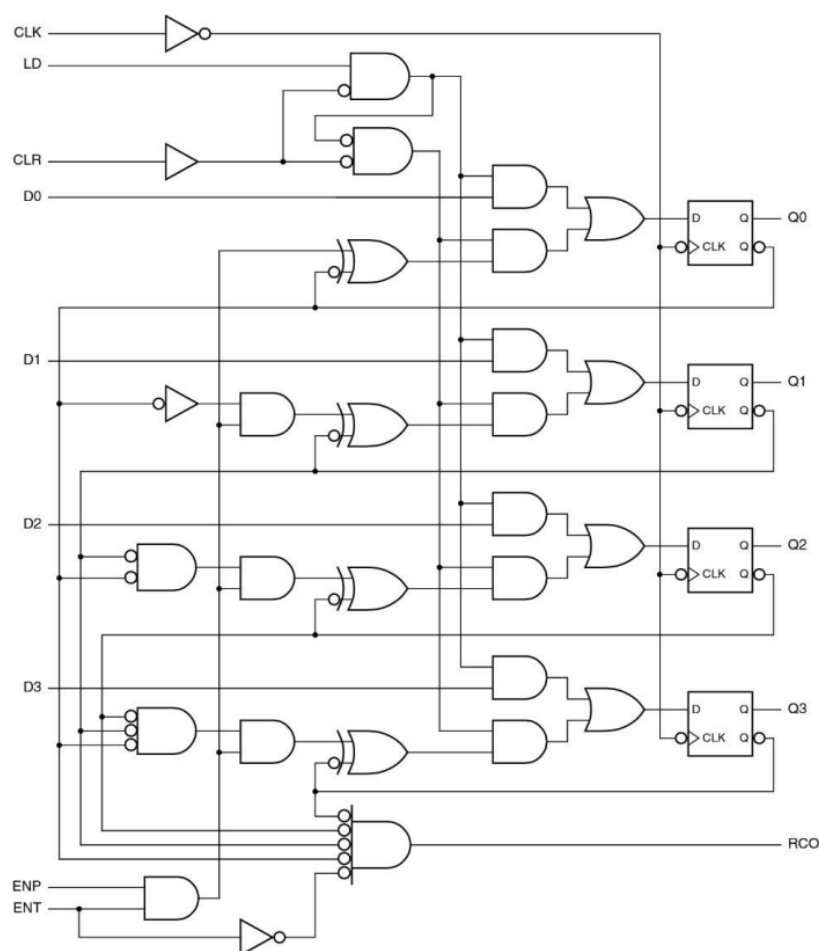
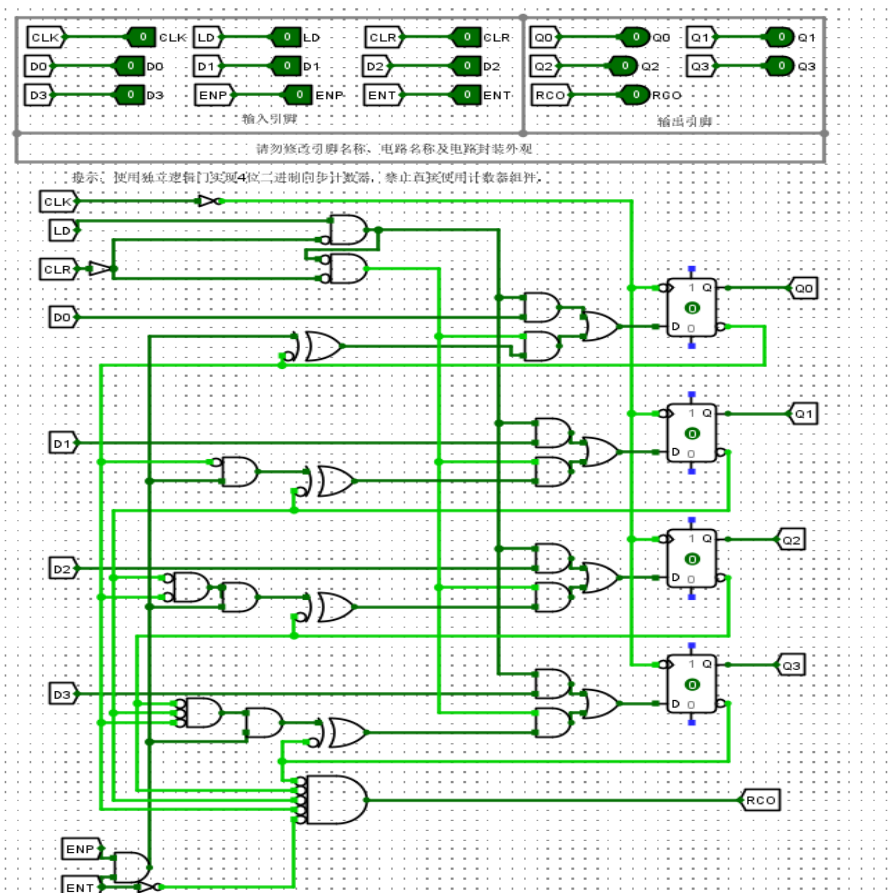
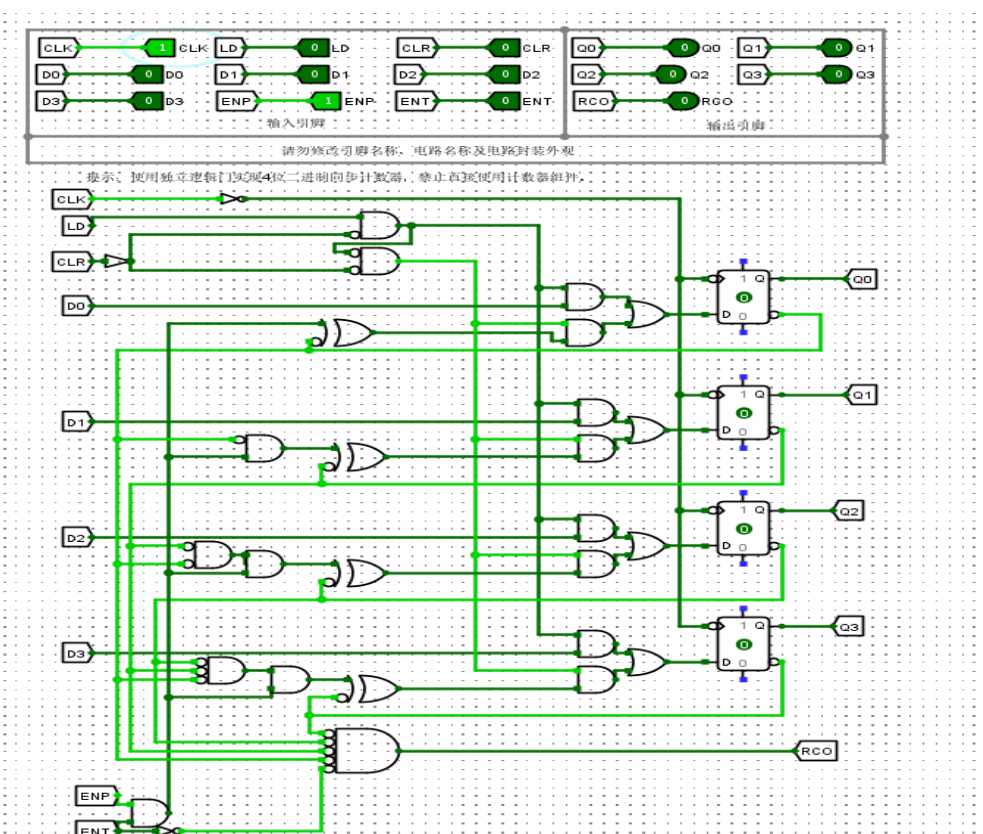


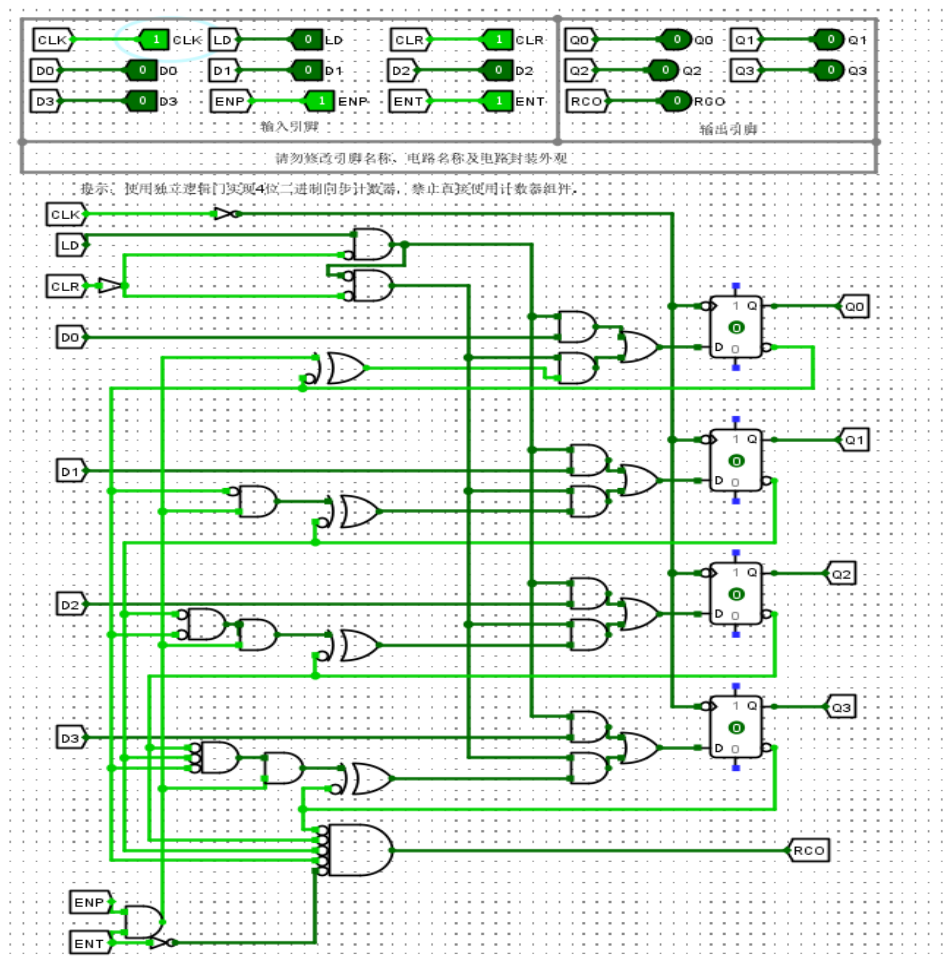
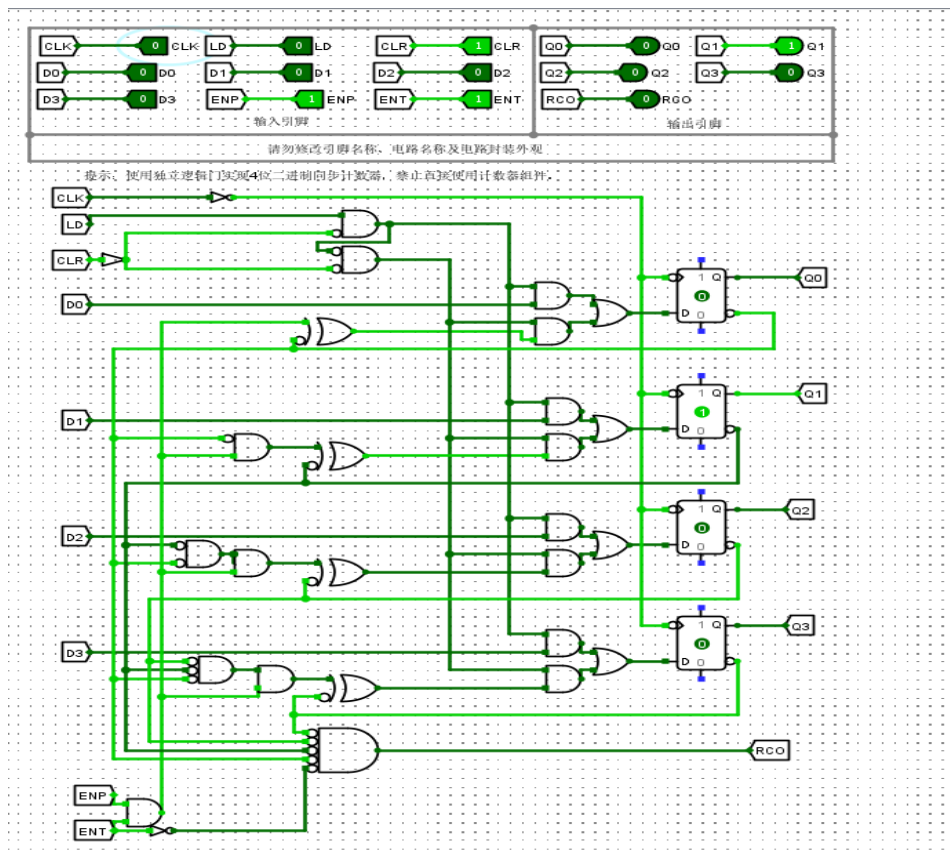
图 3.1 4 位二进制同步计数器原理图

电路图：



仿真测试图：





功能表：

表 3.1 4 位同步二进制计数器功能表

Inputs				Current State				Next State			
CLR	LD	ENT	ENP	Q3	Q2	Q1	Q0	Q3*	Q2*	Q1*	Q0*
1	x	x	x	x	x	x	x	0	0	0	0
0	1	x	x	x	x	x	x	D3	D2	D1	D0
0	0	0	x	x	x	x	x	Q3	Q2	Q1	Q0
0	0	x	0	x	x	x	x	Q3	Q2	Q1	Q0
0	0	1	1	0	0	0	0	0	0	0	1
0	0	1	1	0	0	0	1	0	0	1	0
...											
0	0	1	1	1	1	0	1	1	1	1	0
0	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	1	1	1	1	0	0	0	0

2、移位寄存器实验

整体模块设计：

使用 16 个与门， 4 个或门， 4 个下降沿触发的 D 触发器实现移位寄存器。

输入输出引脚：

CLR 是异步复位清零信号，为低电平有效信号，当它为低电平时，所有 D 触发器的状态输出为 0；S1S0 为控制端，11 为装载模式，10 为左移模式，01 为右移模式，00 为保持模式；Q3，Q2，Q1，Q0 为输出信号；D3，D2，D1，D0 为输入信号；Lin 为左移输入端，Rin 为右移输入端。

原理图：

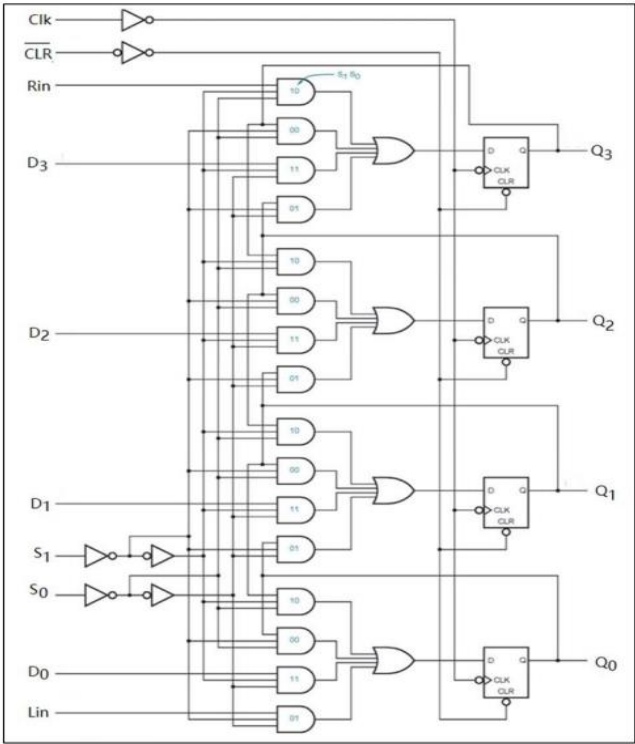
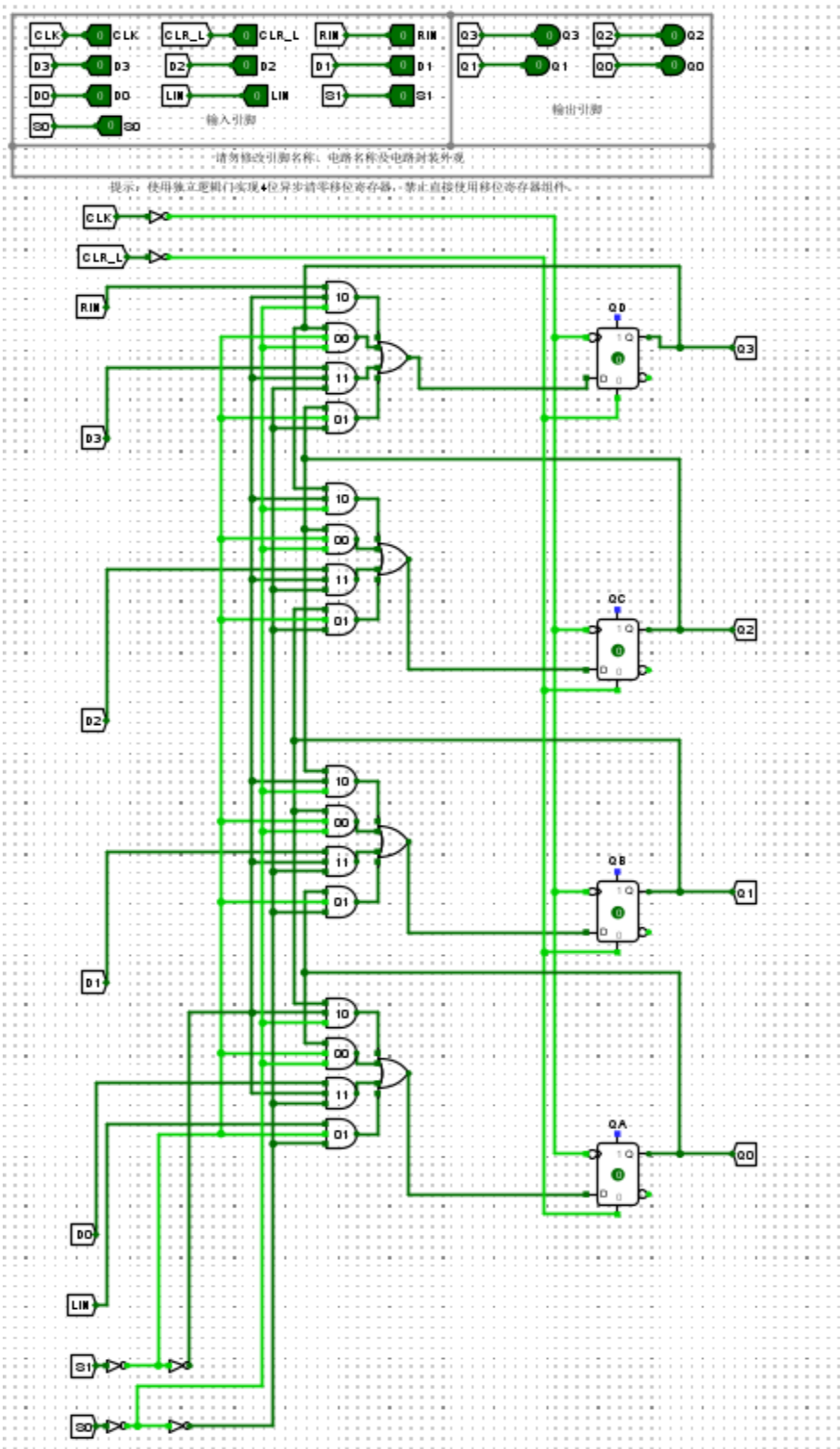
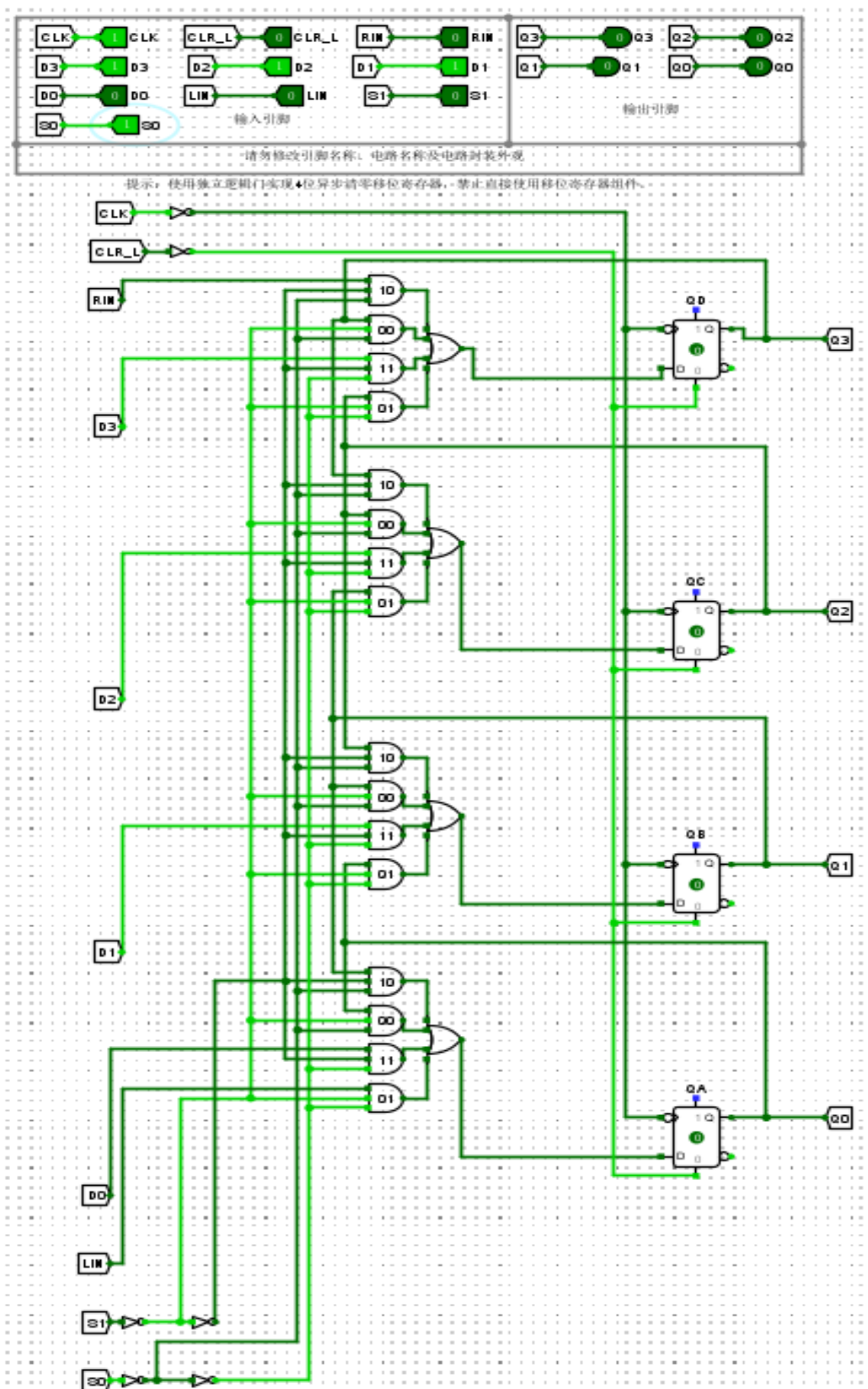


图 3.5 4 位移位寄存器原理图

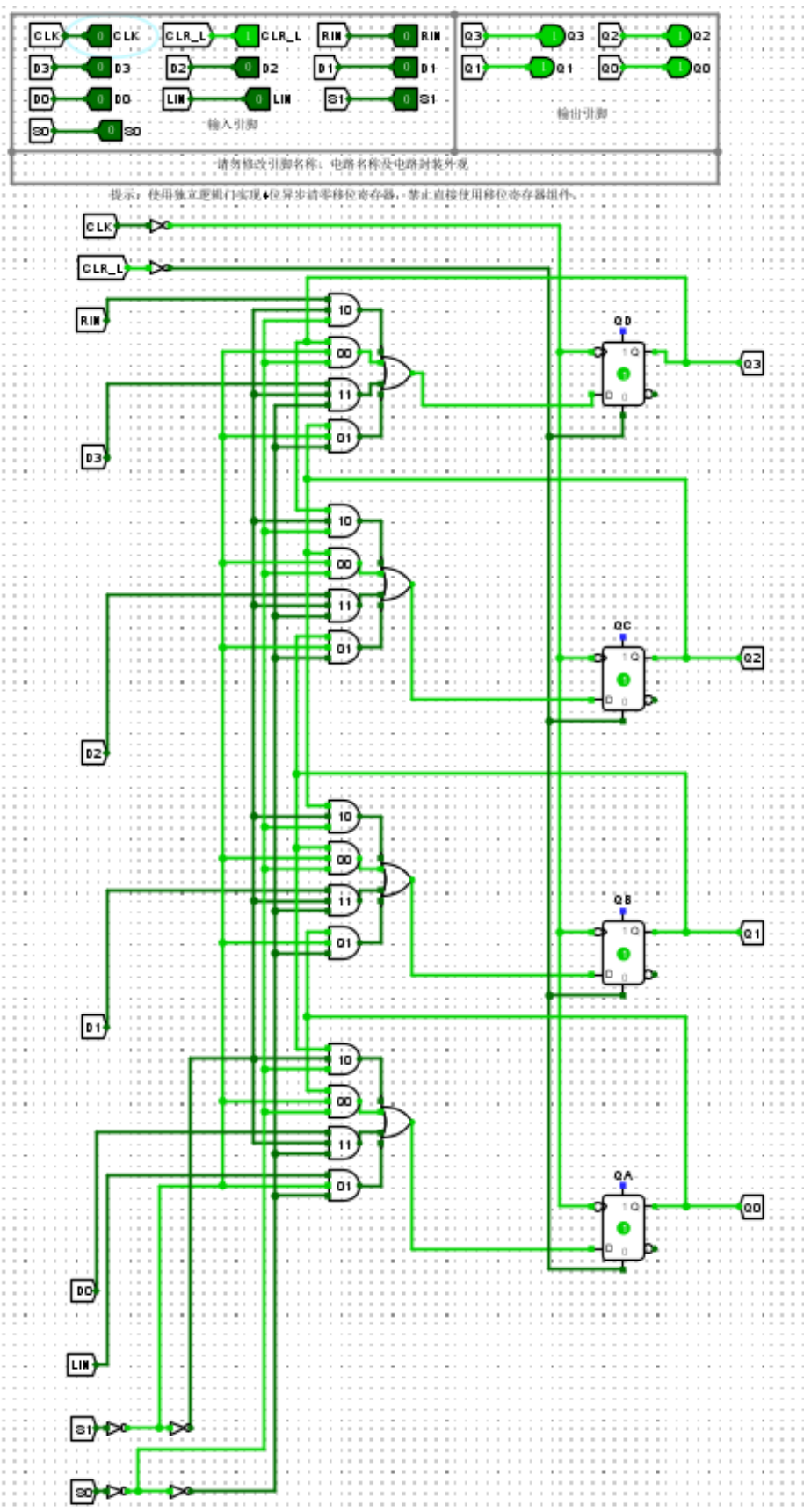
电路图:



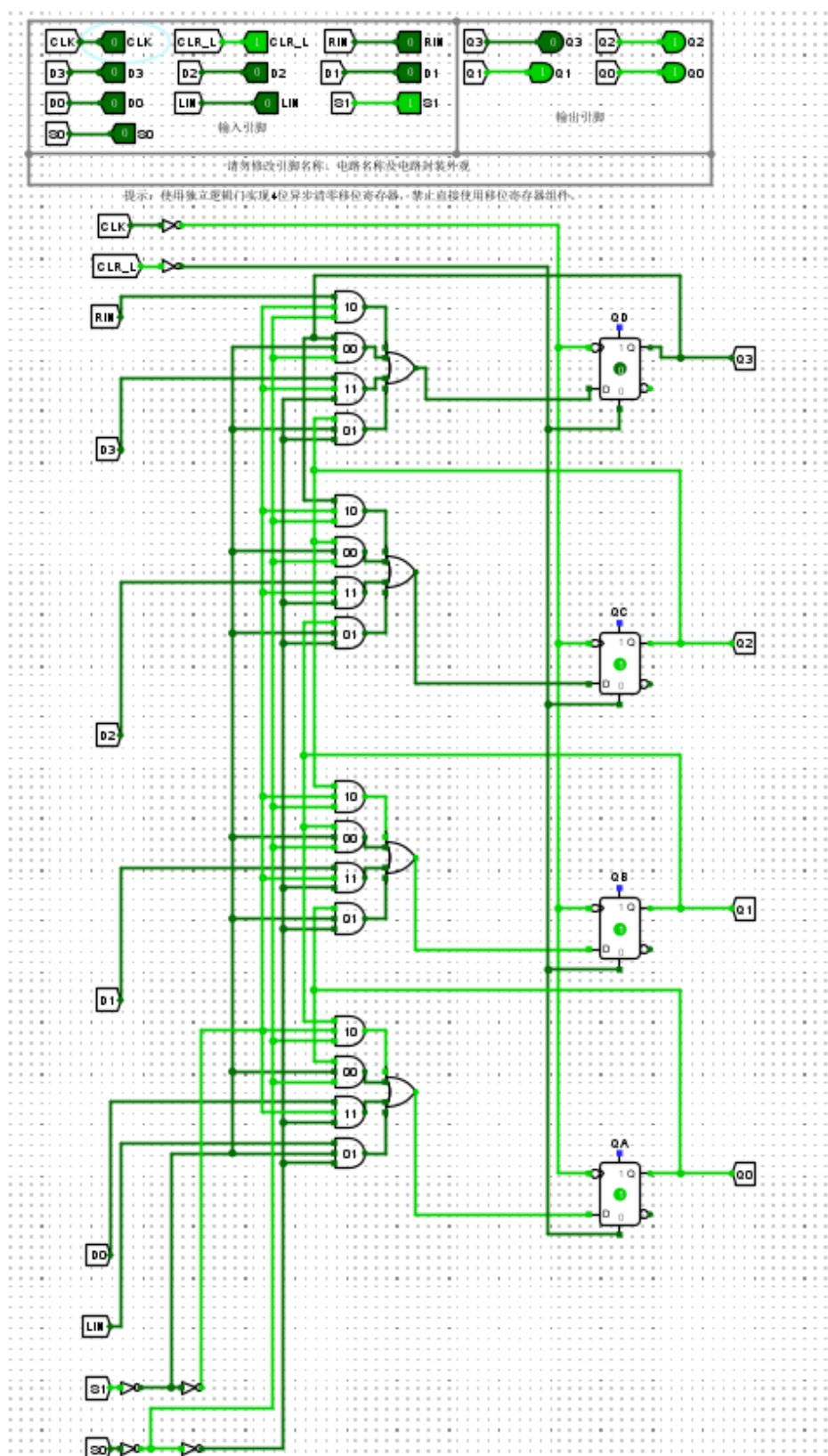
仿真测试图：
(清零状态)

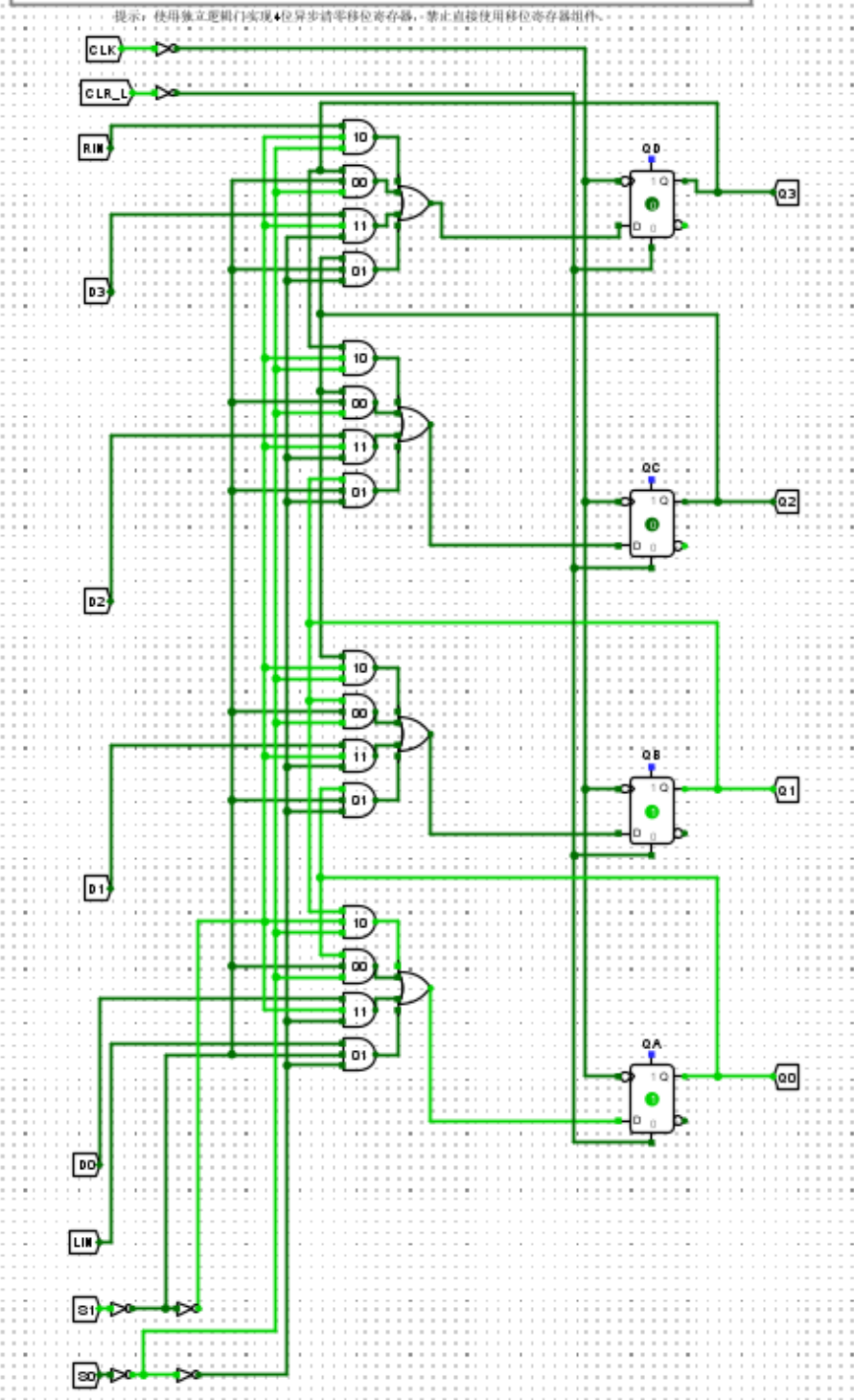
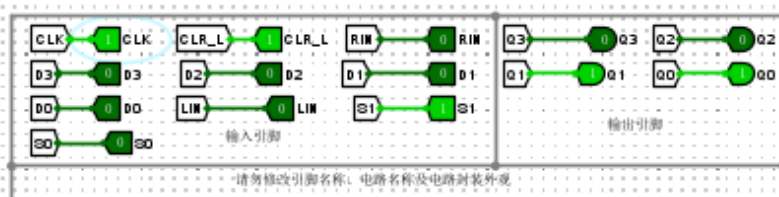


(保持状态)

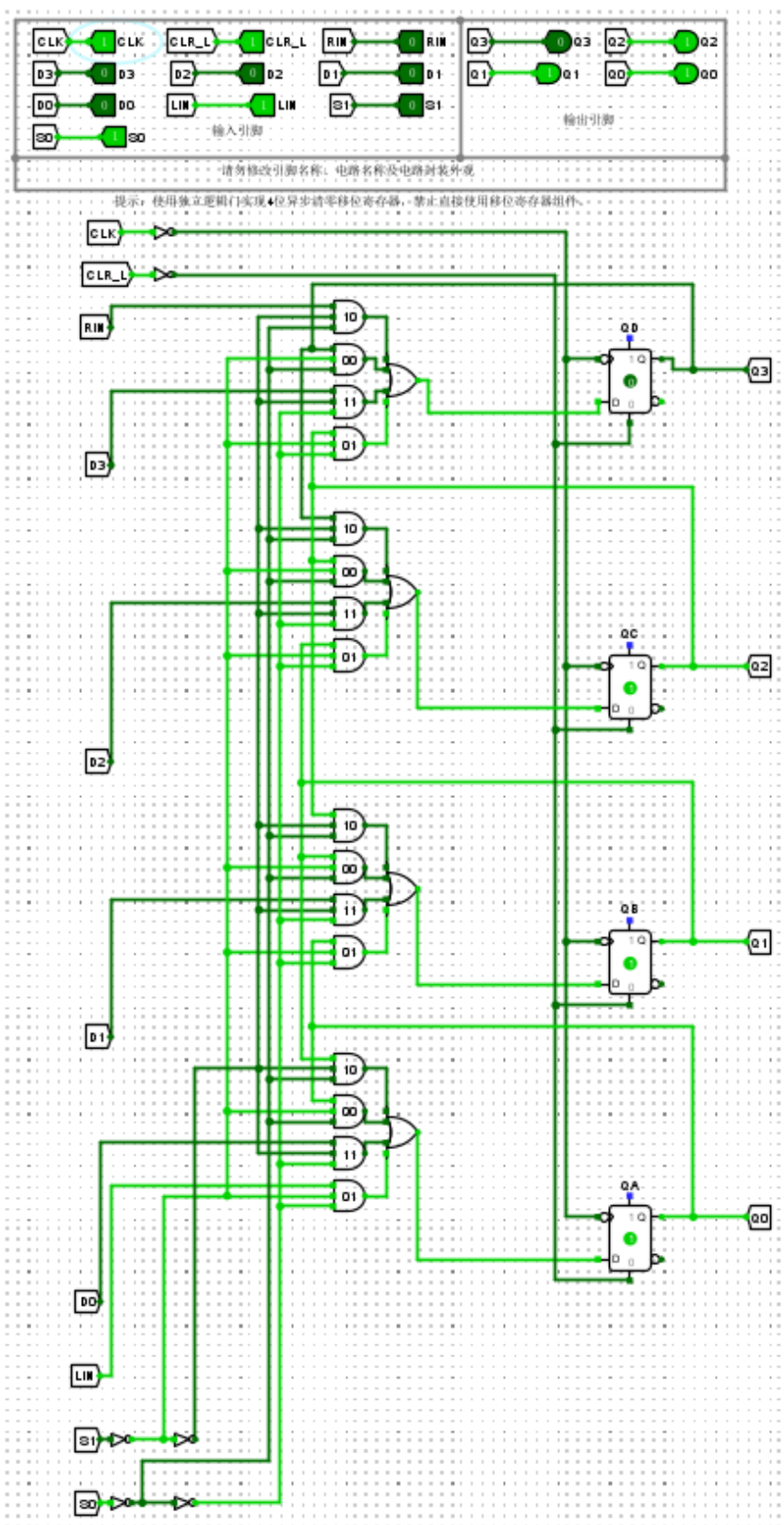


(右移状态)

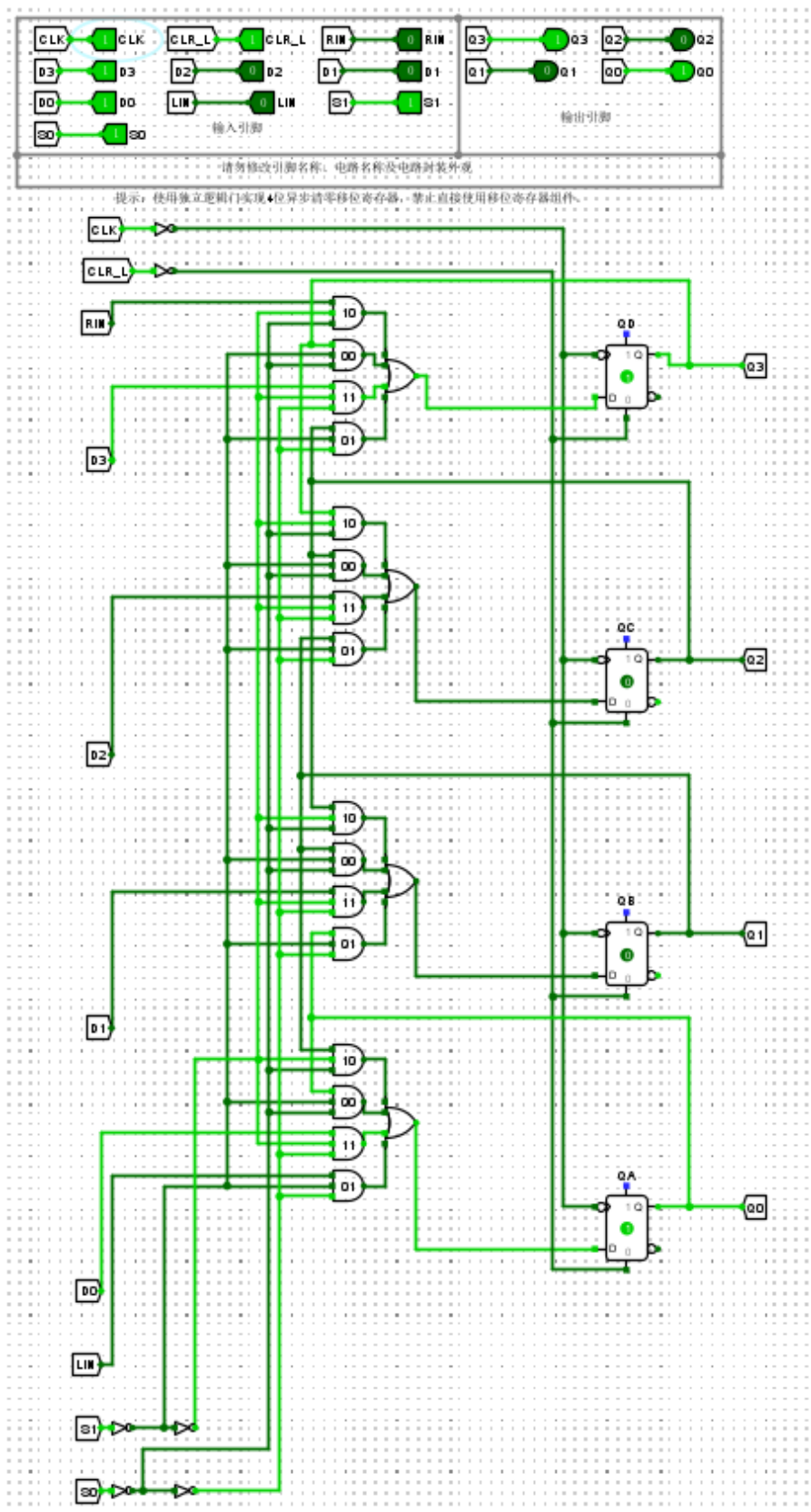




(左移状态) (原状态接上图)



(装载模式) (原状态接上图)



功能表：

表 11.9 4 位移位寄存器功能表

功能	输入			下一个状态			
	CLR	S1	S0	QA*	QB*	QC*	QD*
清零	1	x	x	0	0	0	0
保持	0	0	0	QA	QB	QC	QD
右移	0	0	1	RIN	QA	QB	QC
左移	0	1	0	QB	QC	QD	LIN
装载	0	1	1	A	B	C	D

3、4 位无符号数乘法器

整体模块设计：

利用移位器和加法器实现无符号原码一位乘法，利用计数器实现乘法的开始与结束。

原理：

计算机中两个无符号数相乘，类似手算乘法。主要步骤如下。

- ① 每次将乘数 Y 的一位乘以被乘数得 $X \times Y_i$ 后，就将该结果与前面所得的结果累加，得到 P_i ，称之为部分积。
- ② 在每次求得 $X \times Y_i$ 后，不是将它左移与前次部分积 P_i 相加，而是将部分积 P_i 右移一位与 $X \times Y_i$ 相加。
- ③ 对乘数中为 1 的位执行加法和右移运算，对为 0 的位只执行右移运算，而不需执行加法运算。

输入输出引脚作用：

RST 为复位信号，CLK 为共同时钟，STOP 为结束信号，LOAD 为装载乘数信号，X 为被乘数，Y 为乘数，Z 为乘法结果。

原理图：

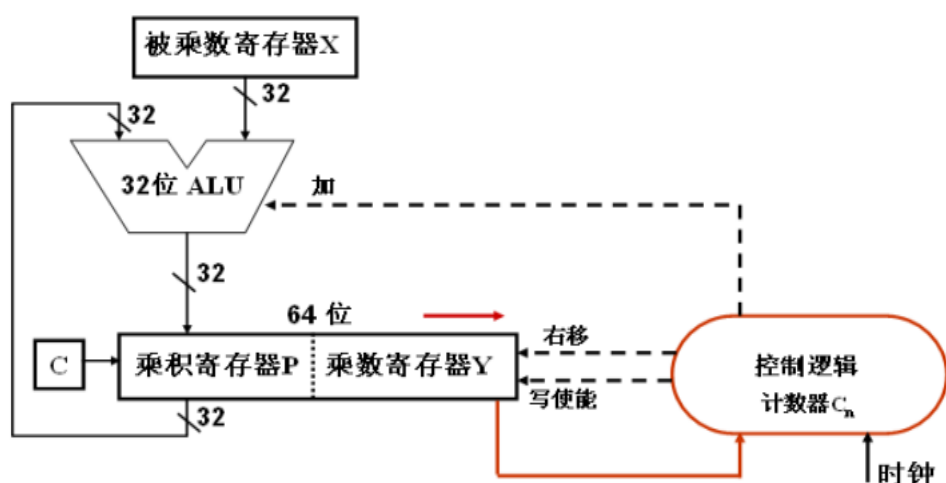
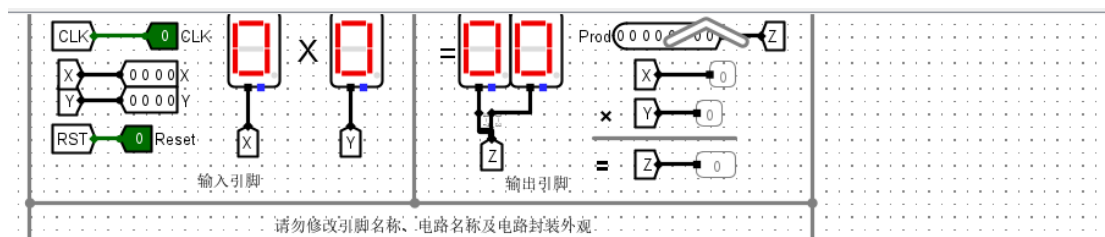
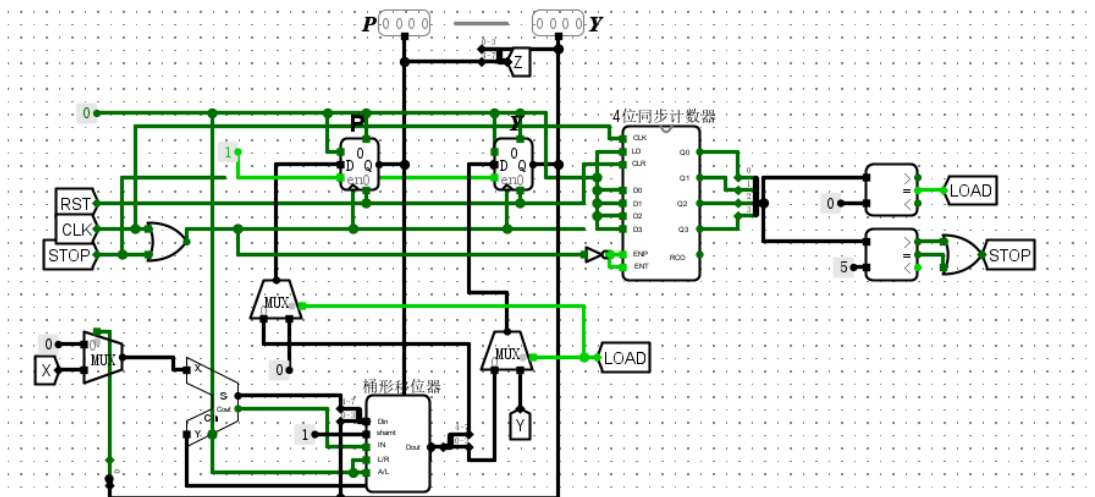


图 3.7 实现 32 位无符号数乘法运算的逻辑结构图

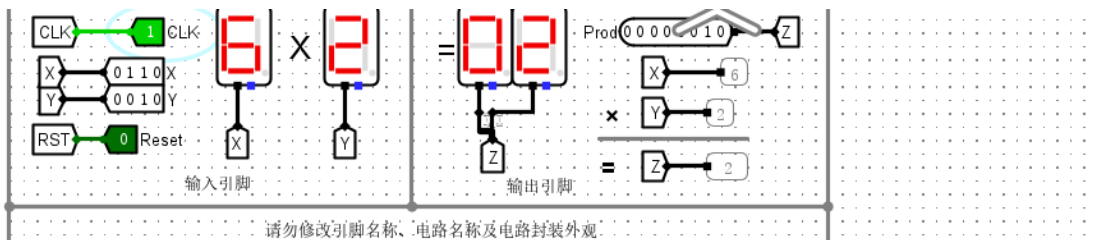
电路图



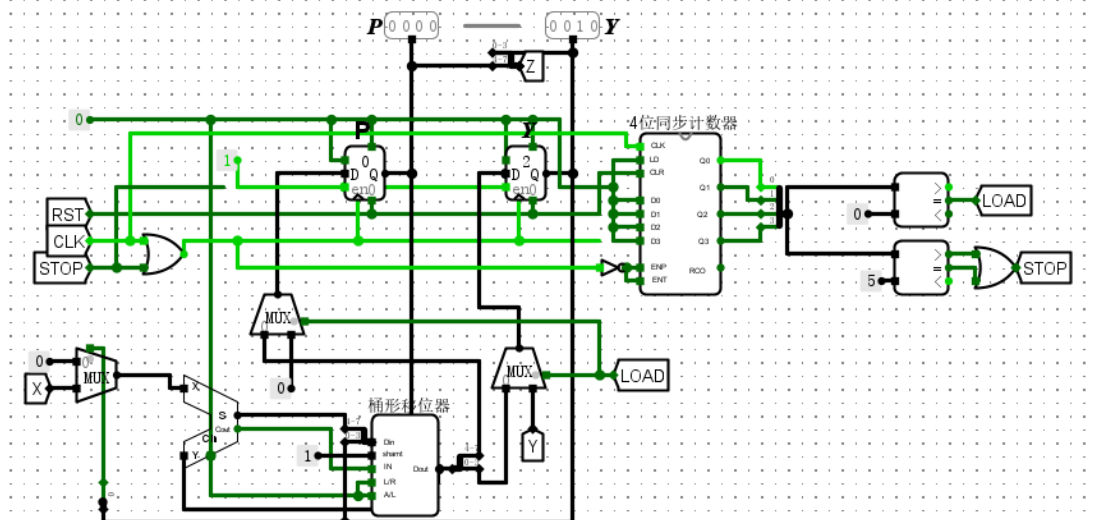
提示: 采用无符号数一位乘法实现4位二进制乘法器, 禁止直接使用乘法器组件

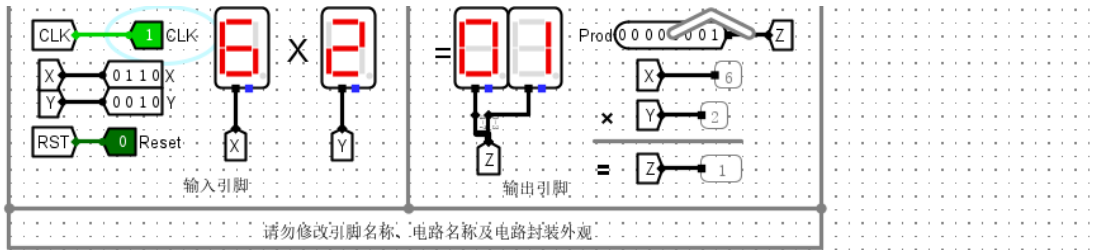


仿真测试图:

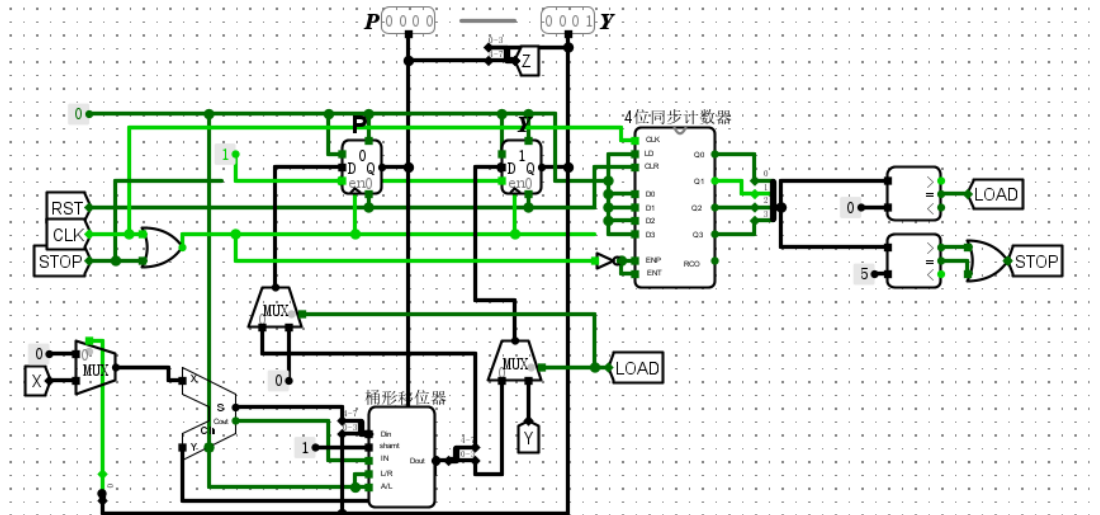


提示: 采用无符号数一位乘法实现4位二进制乘法器, 禁止直接使用乘法器组件

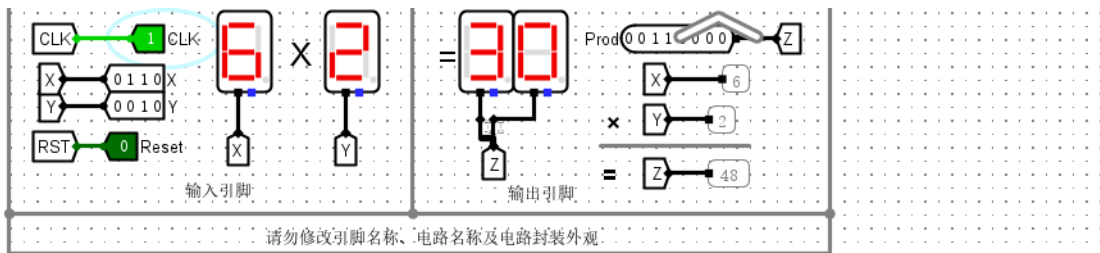




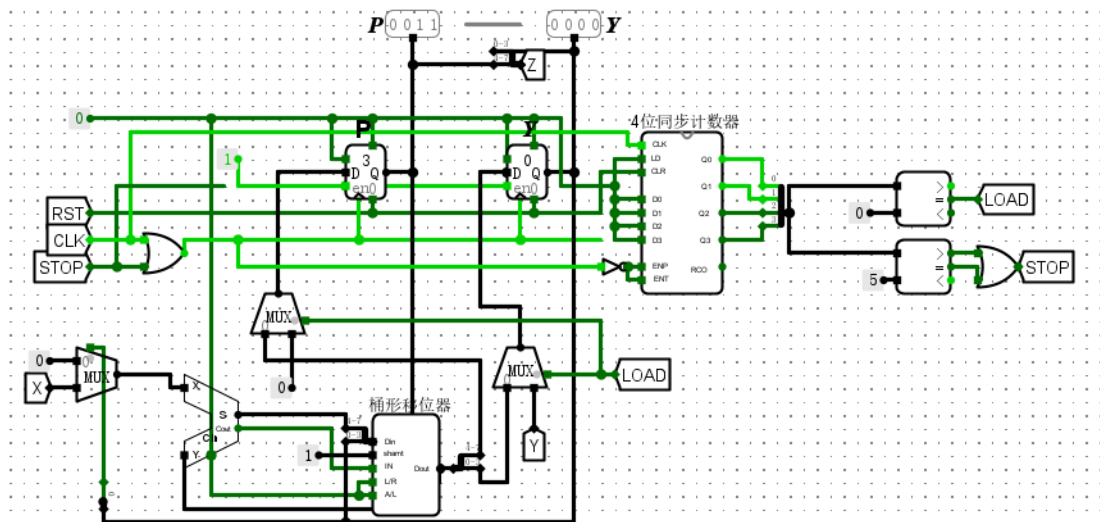
提示: 采用无符号数一位乘法实现4位二进制乘法器, 禁止直接使用乘法器组件



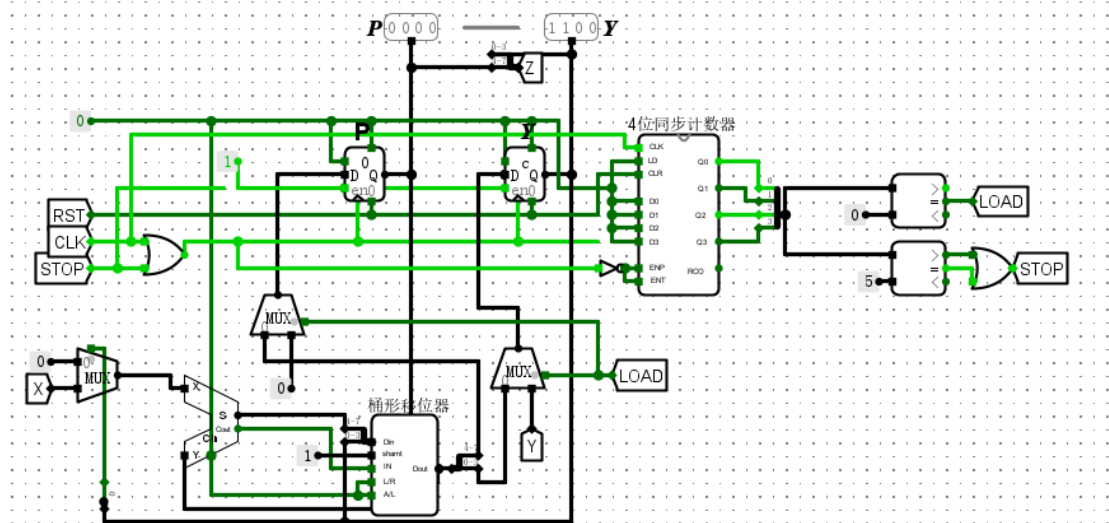
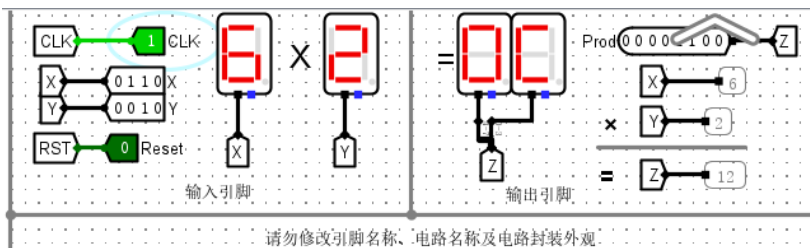
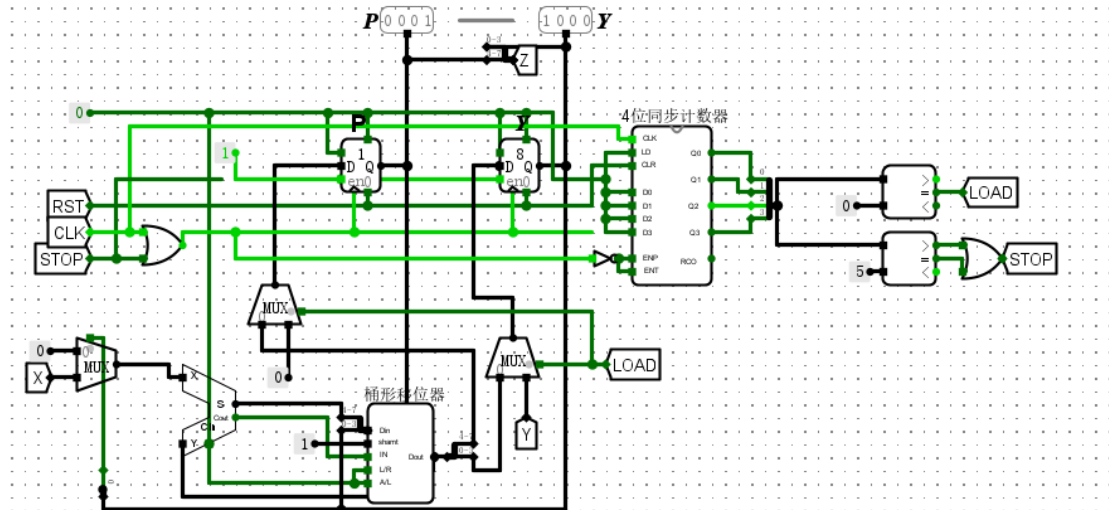
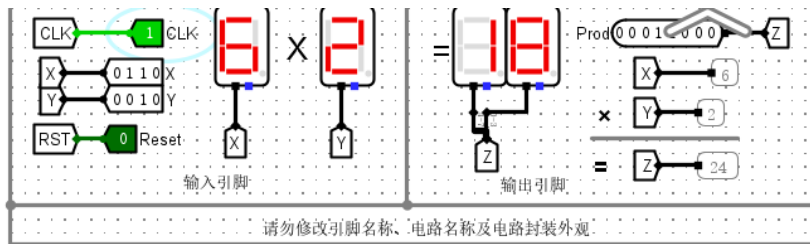
提示: 采用无符号数一位乘法实现4位二进制乘法器, 禁止直接使用乘法器组件

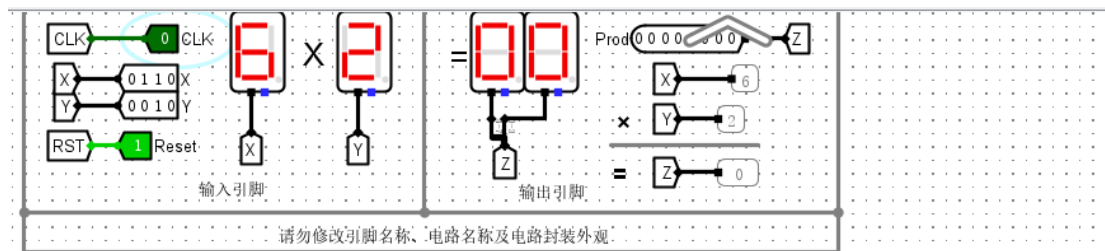


提示: 采用无符号数一位乘法实现4位二进制乘法器, 禁止直接使用乘法器组件

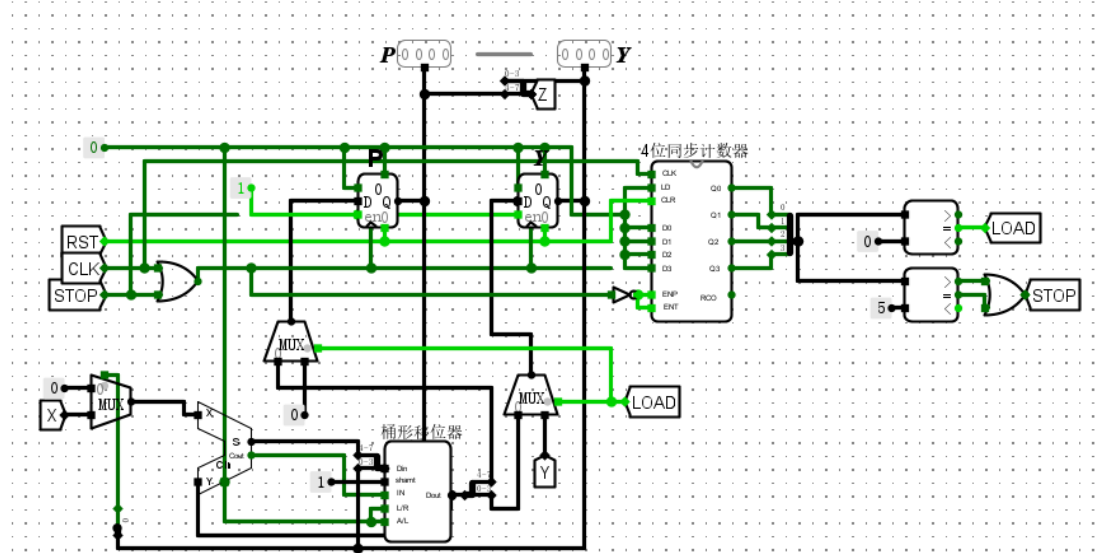


提示: 采用无符号数一位乘法实现4位二进制乘法器, 禁止直接使用乘法器组件





提示: 采用无符号数一位乘法实现4位二进制乘法器, 禁止直接使用乘法器组件



功能表:

Cnt	Reset	X	Y	Prod
00	1	7	7	00
01	0	7	7	00
02	0	7	7	07
03	0	7	7	3b
04	0	7	7	55
05	0	7	7	62
06	0	7	7	31
07	0	7	7	31
08	1	9	a	00
09	0	9	a	00
0a	0	9	a	0a
0b	0	9	a	05
0c	0	9	a	4a
0d	0	9	a	25
0e	0	9	a	5a
0f	0	9	a	5a
10	1	a	9	00
11	0	a	9	00
12	0	a	9	09
13	0	a	9	54
14	0	a	9	2a
15	0	a	9	15
16	0	a	9	5a
17	0	a	9	5a
18	1	f	f	00
19	0	f	f	00
1a	0	f	f	0f
1b	0	f	f	7f
1c	0	f	f	b7
1d	0	f	f	d3
1e	0	f	f	e1
1f	0	f	f	e1

4、寄存器堆实验

整体模块设计：

使用 2 个多路选择器，32 个 32 位寄存器实现寄存器堆 Regfile 的读写电路。

输入输出引脚作用：

WE 为写使能信号，RA 和 RB 为寄存器地址信号，busA 和 busB 为输出端口，RW 为写入寄存器地址，busW 为写入数据，CLK 为共同时钟。

原理图：

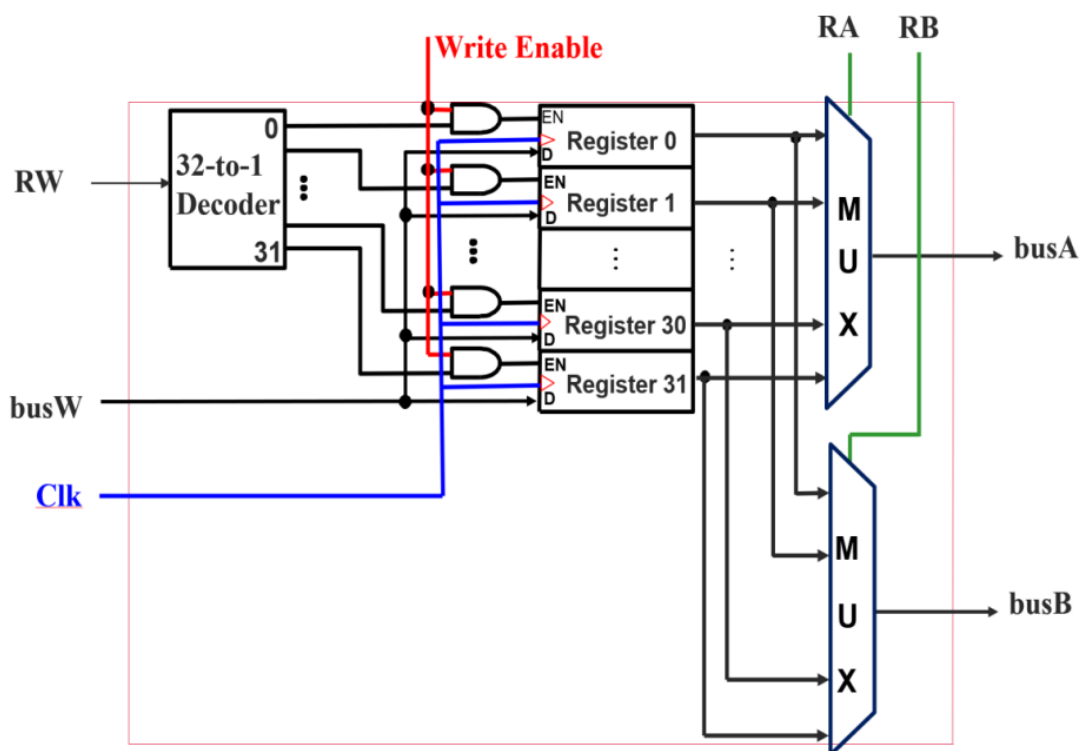
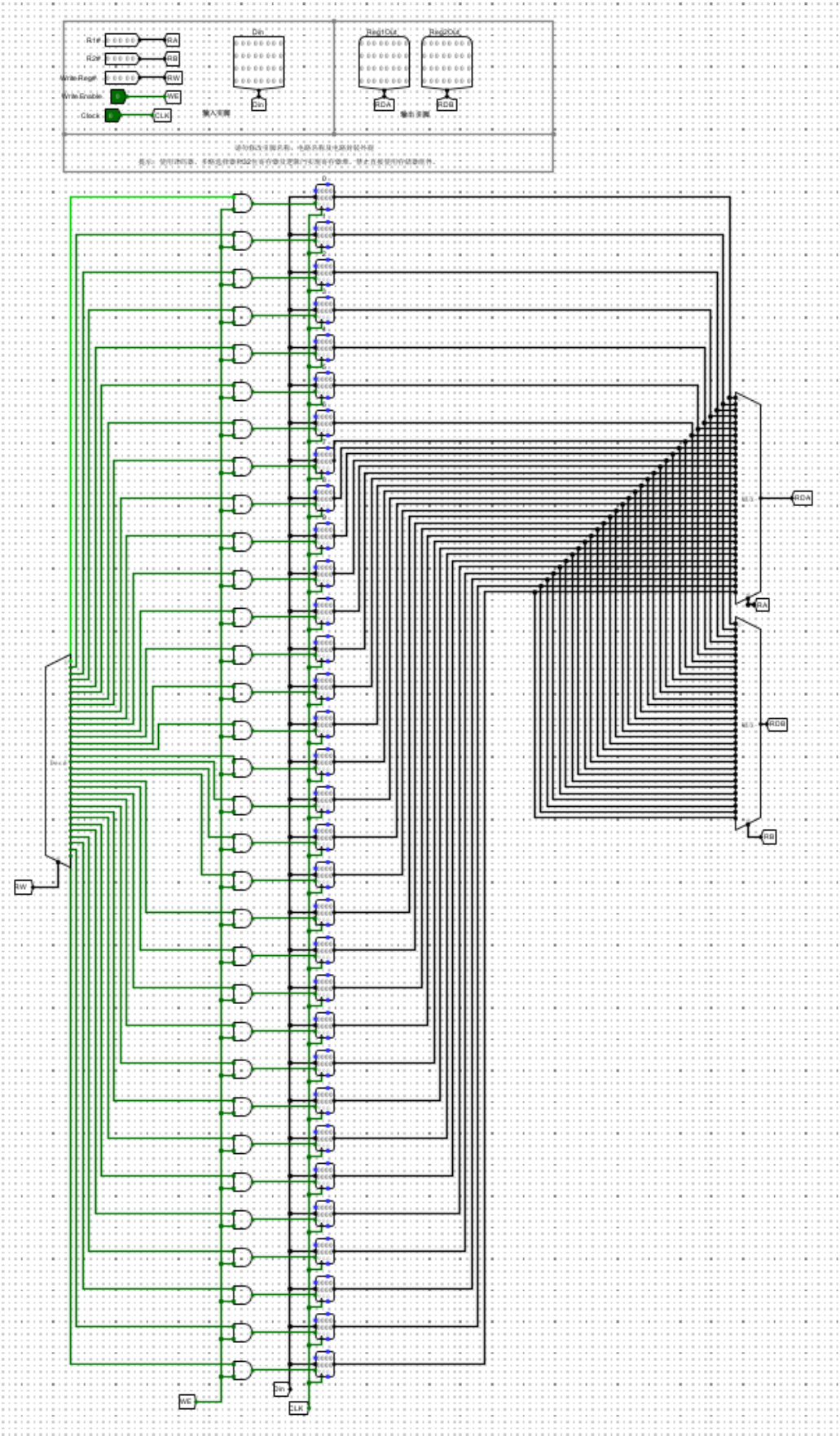
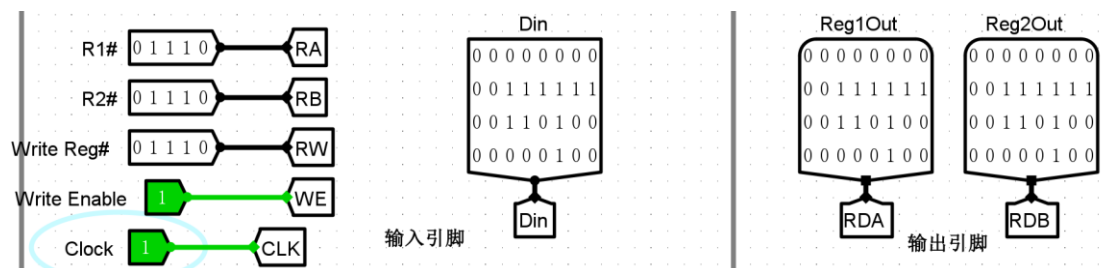
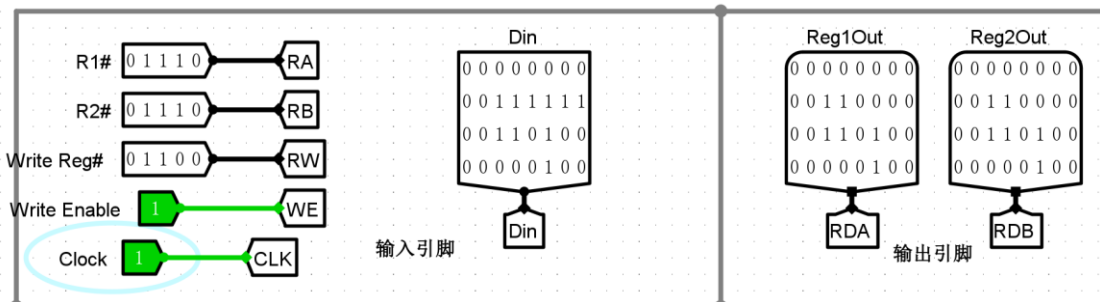
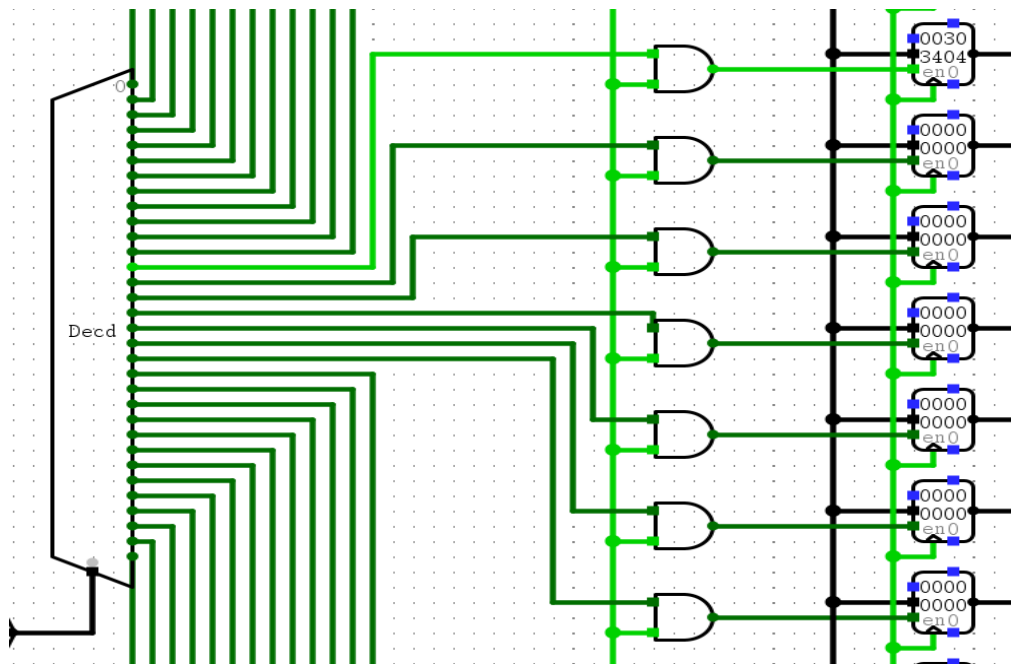
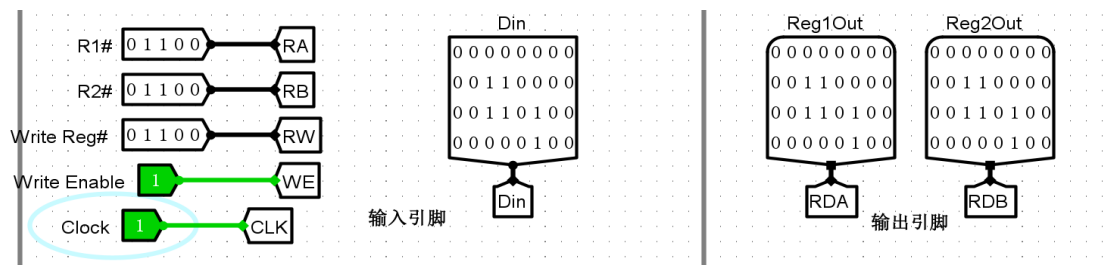


图 3.9 寄存器堆设计原理图

电路图：



仿真测试图：



功能表：

Cnt	WE	RW	RA	RB	Din	RDA	RDB
00	1	00	00	00	00000001	00000000	00000000
01	1	01	00	00	005deece	00000001	00000001
02	1	02	00	01	b61488df	00000001	005deece
03	1	03	01	02	f4111591	005deece	b61488df
04	1	04	02	03	023eaf12	b61488df	f4111591
05	1	05	03	04	b578fa6a	f4111591	023eaf12
06	1	06	04	05	d38a8b1c	023eaf12	b578fa6a
07	1	07	05	06	f5d50649	b578fa6a	d38a8b1c
08	1	08	06	07	202e3c08	d38a8b1c	f5d50649
09	1	09	07	08	812fba12	f5d50649	202e3c08
0a	1	0a	08	09	6755ebab	202e3c08	812fba12
0b	1	0b	09	0a	f6e7cab1	812fba12	6755ebab
0c	1	0c	0a	0b	e6bc21b9	6755ebab	f6e7cab1
0d	1	0d	0b	0c	a6c5abc6	f6e7cab1	e6bc21b9
0e	1	0e	0c	0d	58228a26	e6bc21b9	a6c5abc6
0f	1	0f	0d	0e	9dfc1362	a6c5abc6	58228a26
10	1	10	0e	0f	ac5b2754	58228a26	9dfc1362
11	1	11	0f	10	2f43454c	9dfc1362	ac5b2754
12	1	12	10	11	18c3dc9c	ac5b2754	2f43454c
13	1	13	11	12	1abbd85c	2f43454c	18c3dc9c
14	1	14	12	13	cb1b519a	18c3dc9c	1abbd85c
15	1	15	13	14	0f672c6a	1abbd85c	cb1b519a
16	1	16	14	15	e37362d1	cb1b519a	0f672c6a
17	1	17	15	16	39eae4b1	0f672c6a	e37362d1
18	1	18	16	17	95c7a81f	e37362d1	39eae4b1
19	1	19	17	18	9b8ab32e	39eae4b1	95c7a81f
1a	1	1a	18	19	64bb0d7c	95c7a81f	9b8ab32e
1b	1	1b	19	1a	72f86d97	9b8ab32e	64bb0d7c
1c	1	1c	1a	1b	52130e8e	64bb0d7c	72f86d97
1d	1	1d	1b	1c	1ee9d827	72f86d97	52130e8e
1e	1	1e	1c	1d	9056d989	52130e8e	1ee9d827
1f	1	1f	1d	1e	dc896b07	1ee9d827	9056d989
20	0	00	00	00	913171e3	00000001	00000001
21	0	01	00	00	6a830aaa	00000001	00000001
22	0	02	00	01	ebccdf55	00000001	005deece
23	0	03	01	02	1bead7a1	005deece	b61488df
24	0	04	02	03	70ce89af	b61488df	f4111591
25	0	05	03	04	d6c6f30f	f4111591	023eaf12

26	0	06	04	05	4f2f15f6	023eaf12	b578fa6a
27	0	07	05	06	0bfd3217	b578fa6a	d38a8b1c
28	0	08	06	07	74276d58	d38a8b1c	f5d50649
29	0	09	07	08	c6947ef2	f5d50649	202e3c08
2a	0	0a	08	09	e86e0a71	202e3c08	812fba12
2b	0	0b	09	0a	e71009ff	812fba12	6755ebab
2c	0	0c	0a	0b	d4203d58	6755ebab	f6e7cab1
2d	0	0d	0b	0c	bd817173	f6e7cab1	e6bc21b9
2e	0	0e	0c	0d	64d7cb03	e6bc21b9	a6c5abc6
2f	0	0f	0d	0e	5847aa13	a6c5abc6	58228a26
30	0	10	0e	0f	b183207a	58228a26	9dfc1362
31	0	11	0f	10	5333b778	9dfc1362	ac5b2754
32	0	12	10	11	fad2465a	ac5b2754	2f43454c
33	0	13	11	12	608b2671	2f43454c	18c3dc9c
34	0	14	12	13	1229d521	18c3dc9c	1abbd85c
35	0	15	13	14	bd93bfe7	1abbd85c	cb1b519a
36	0	16	14	15	57945ed9	cb1b519a	0f672c6a
37	0	17	15	16	149d0edc	0f672c6a	e37362d1
38	0	18	16	17	d4a71682	e37362d1	39eae4b1
39	0	19	17	18	97e417ee	39eae4b1	95c7a81f
3a	0	1a	18	19	2594f1d9	95c7a81f	9b8ab32e
3b	0	1b	19	1a	c9e504f9	9b8ab32e	64bb0d7c
3c	0	1c	1a	1b	e21915e7	64bb0d7c	72f86d97
3d	0	1d	1b	1c	794e8b3c	72f86d97	52130e8e
3e	0	1e	1c	1d	2c9b12e4	52130e8e	1ee9d827
3f	0	1f	1f	1f	71947694	dc896b07	dc896b07

5、数字时钟实验

整体模块设计：

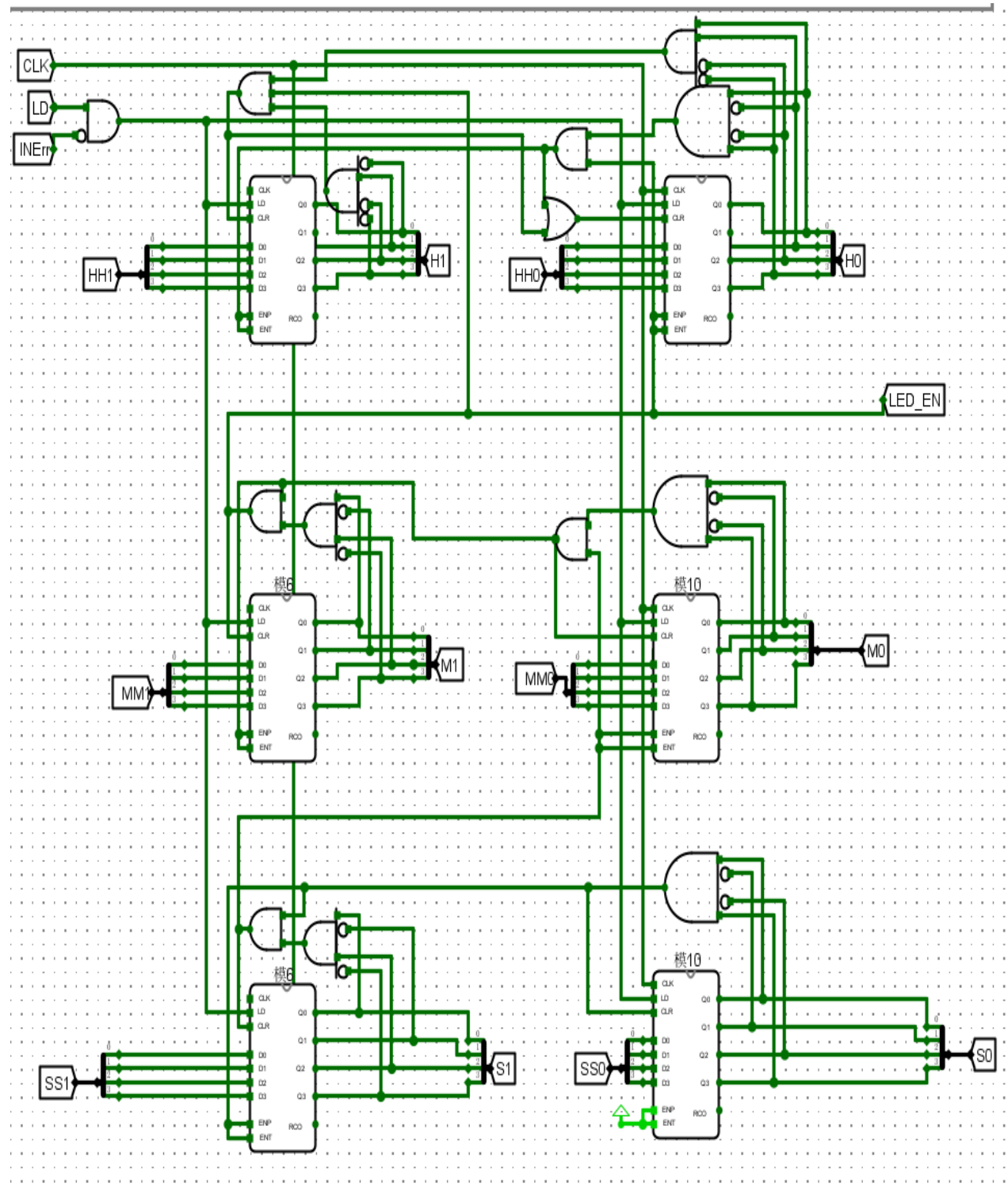
使用 6 个同步二进制计数器，实现分，秒的 bcd 码 60 进制计数器，时的 24 进制计数器。三色 RGB LED 灯按照格雷码顺序轮流点亮三色灯，使用异或逻辑门实现格雷码的转

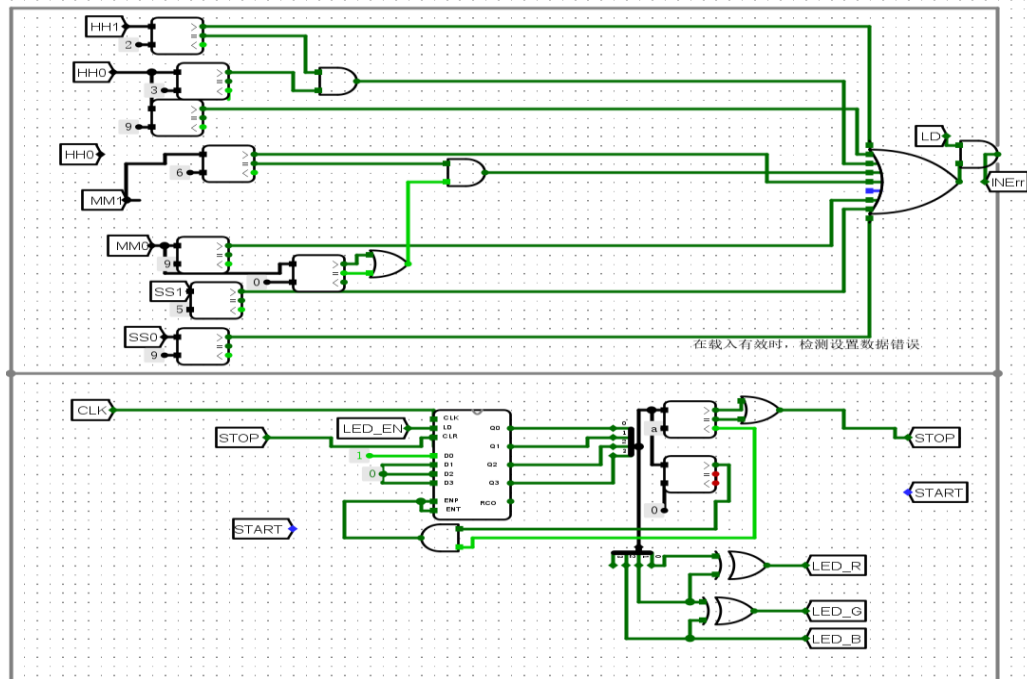
换。使用比较器检查时间数值的初始数值的合法性。

输入输出引脚：

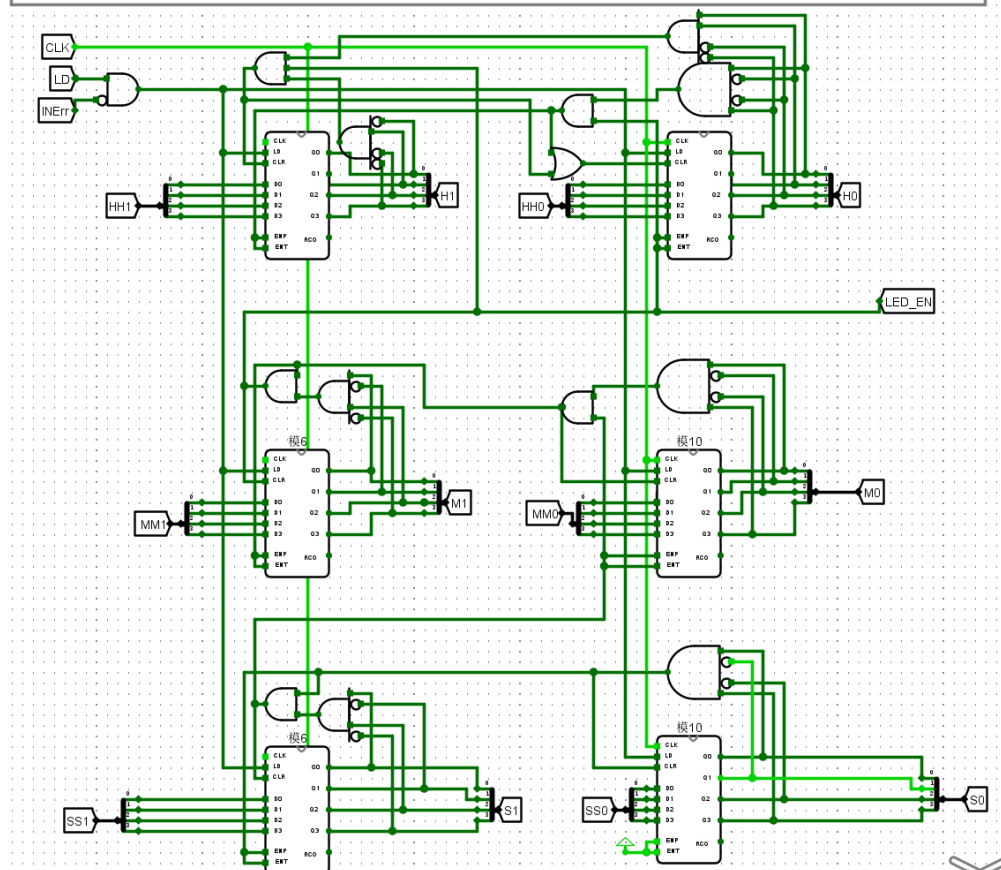
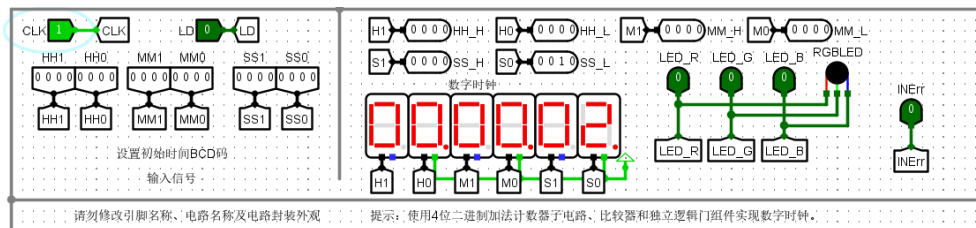
CLK 为共同时钟，LD 为载入信号，INErr 为错误信号，如果在有效数据范围之内，则载入时钟初值，否则 INErr 赋值 1，小时高低位 HH1 和 HH0、分钟高低位 MM1 和 MM0、秒数高低位 SS1 和 SS0，LED-B 为高位、LED-R 为低位。

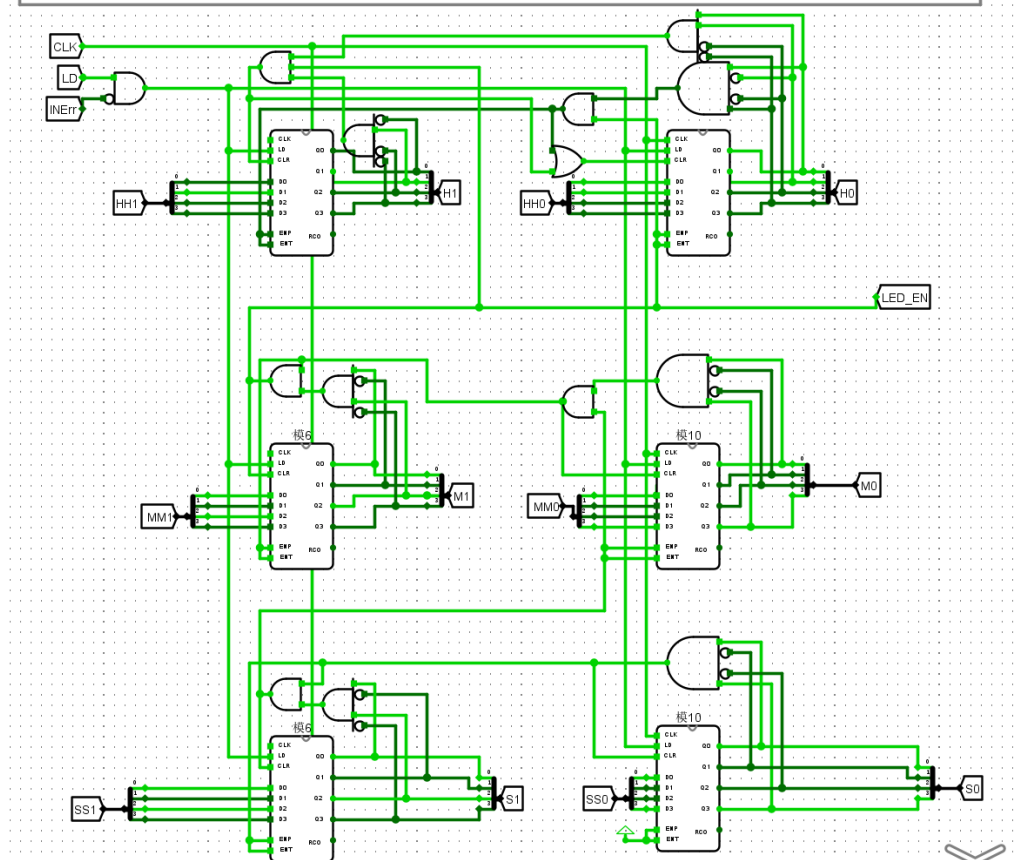
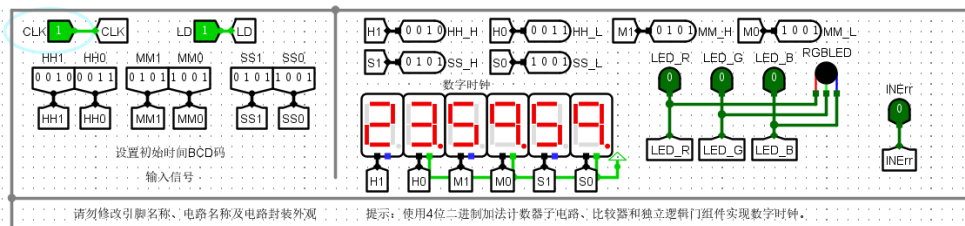
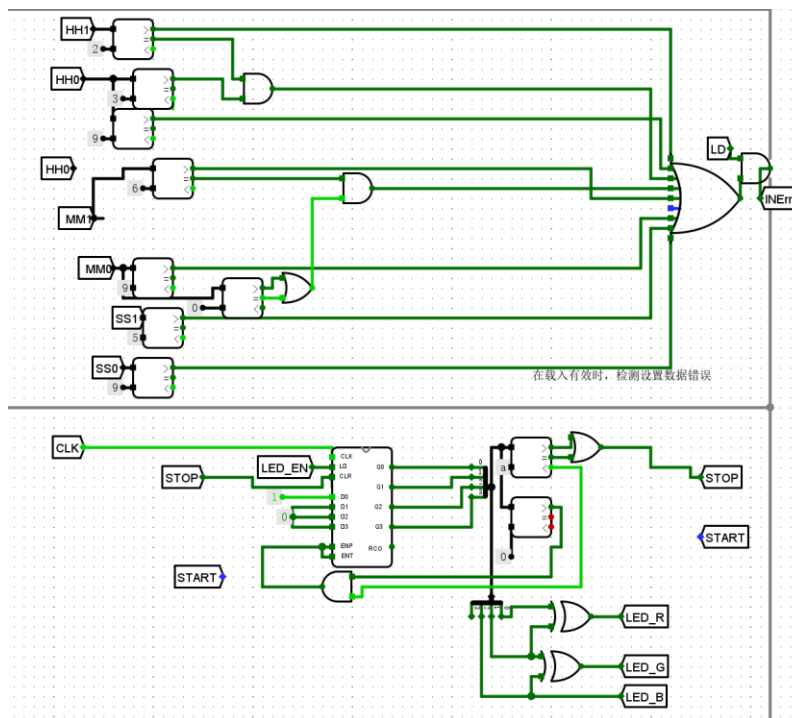
电路图：

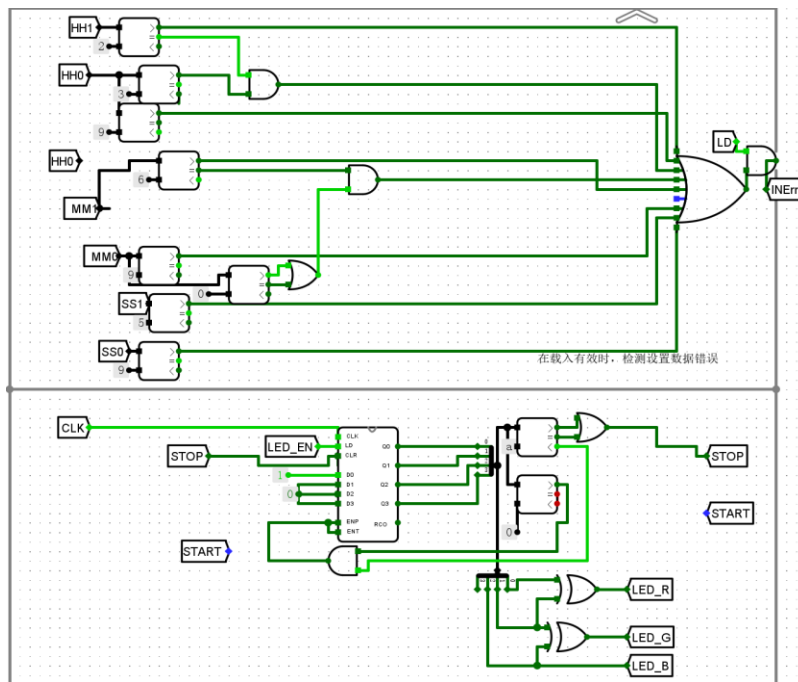




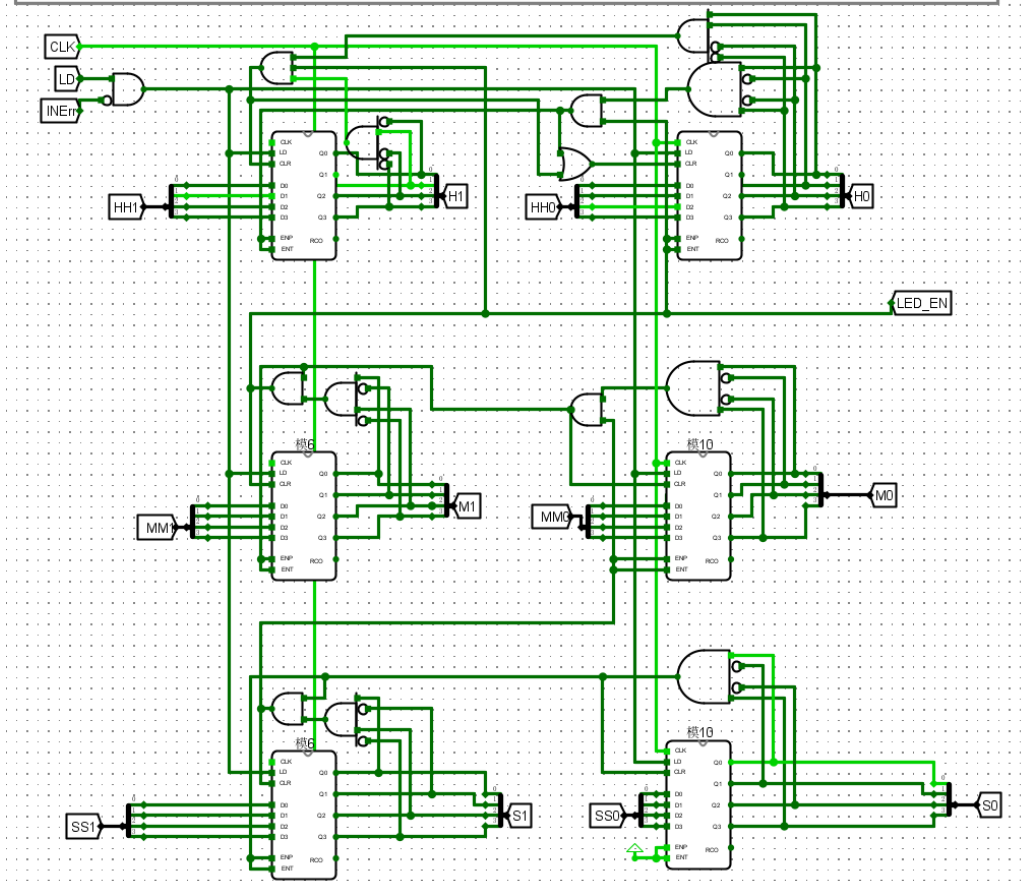
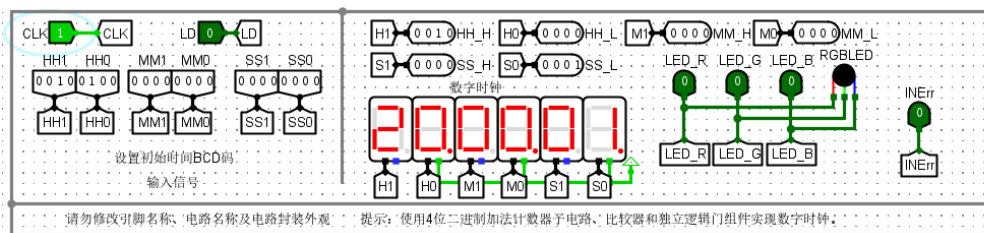
仿真测试图：



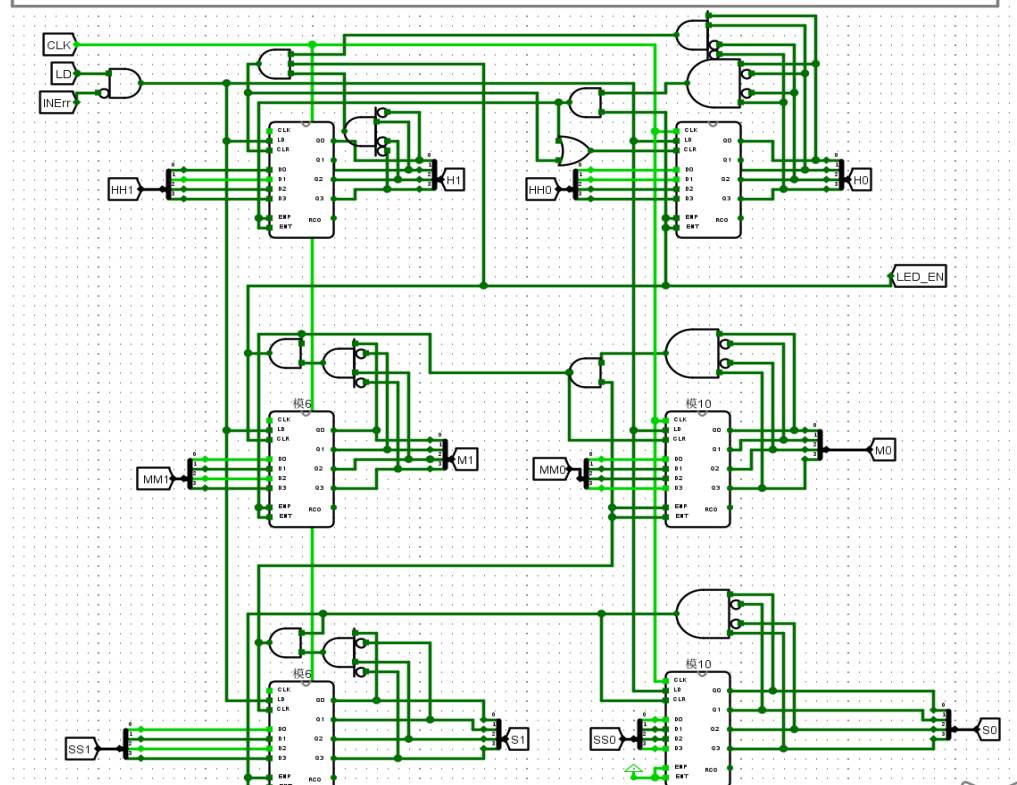
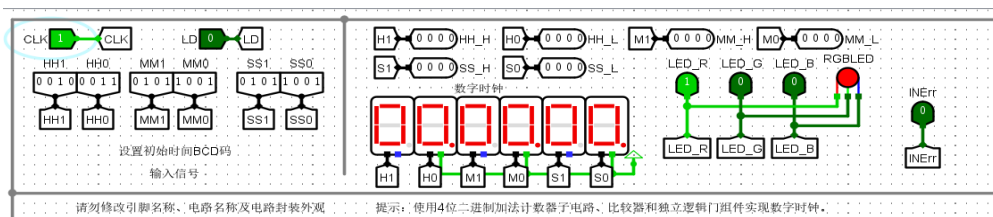
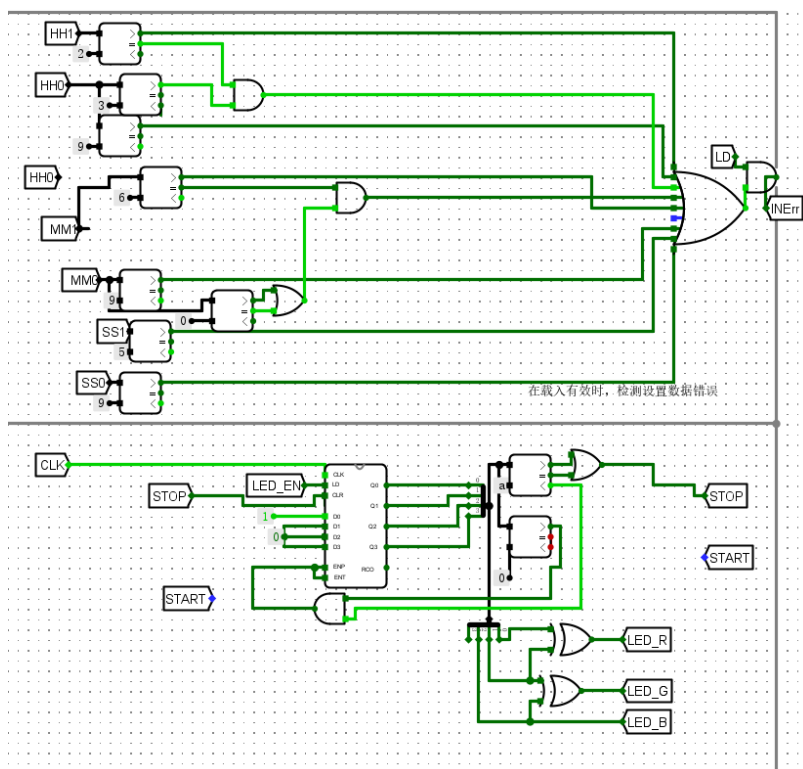


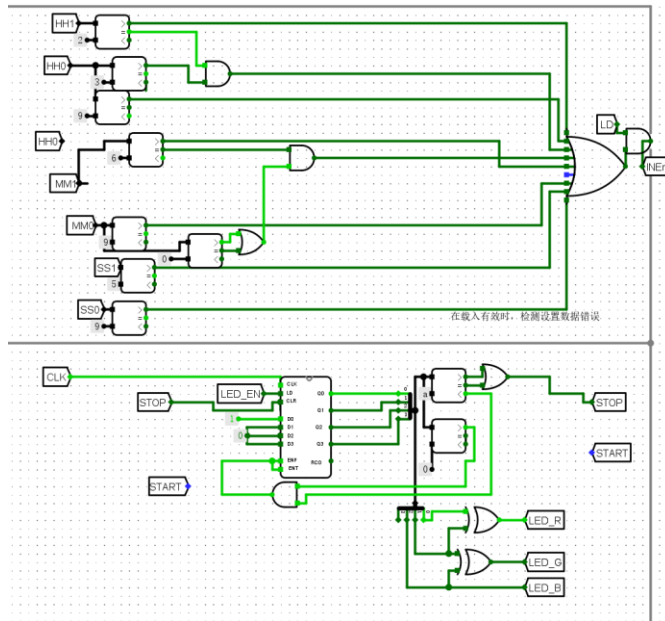


(输入状态同上图)

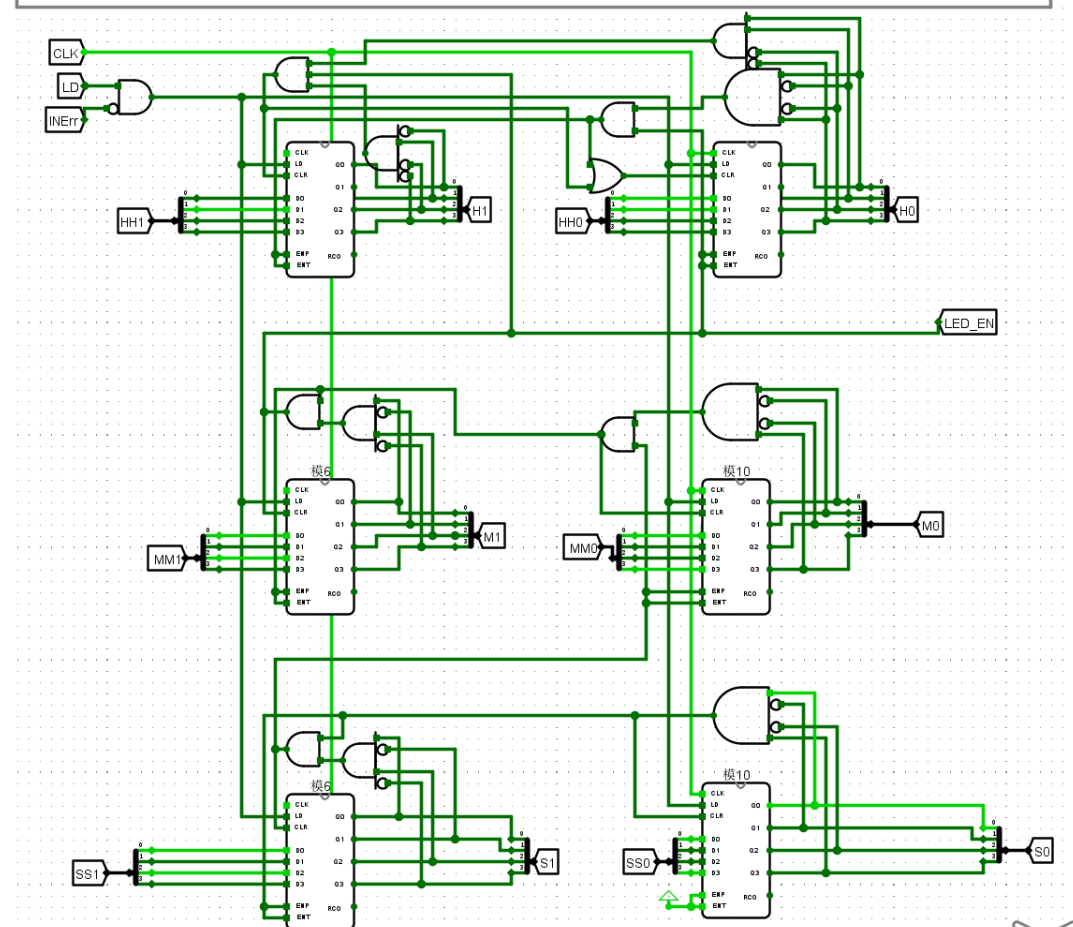
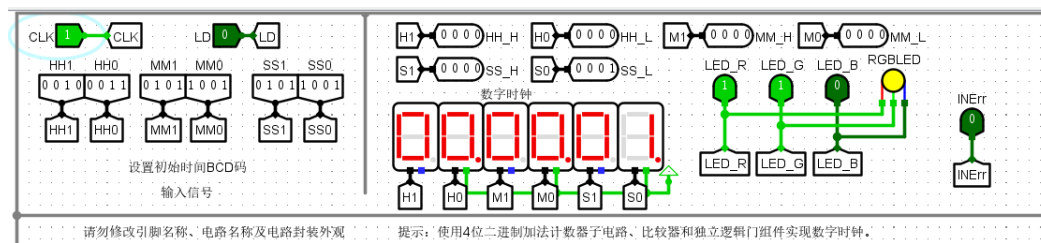


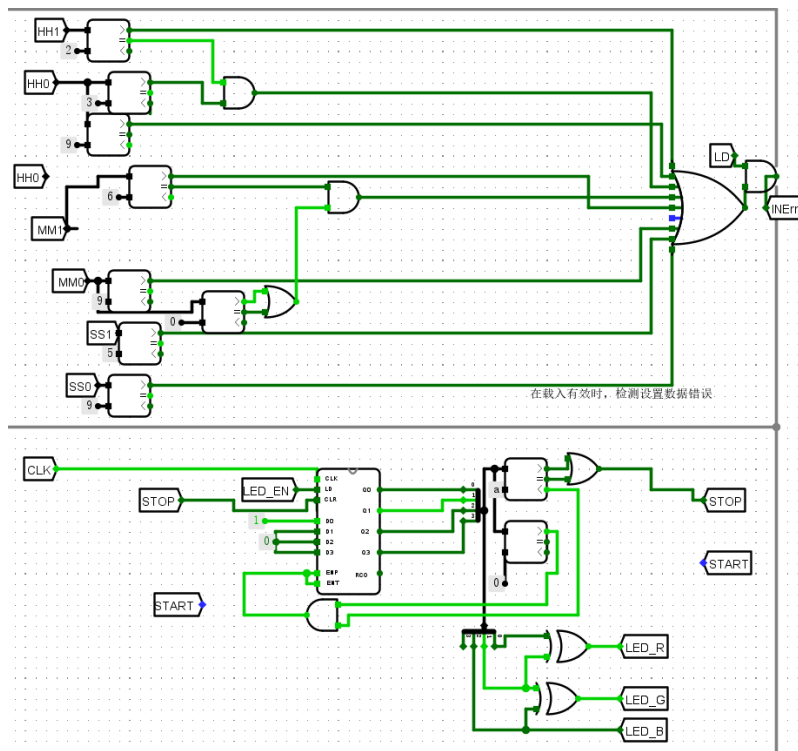
(输入状态同上图 没载入不报错)





(输入状态同上图)





(输入状态同上图)

功能表：

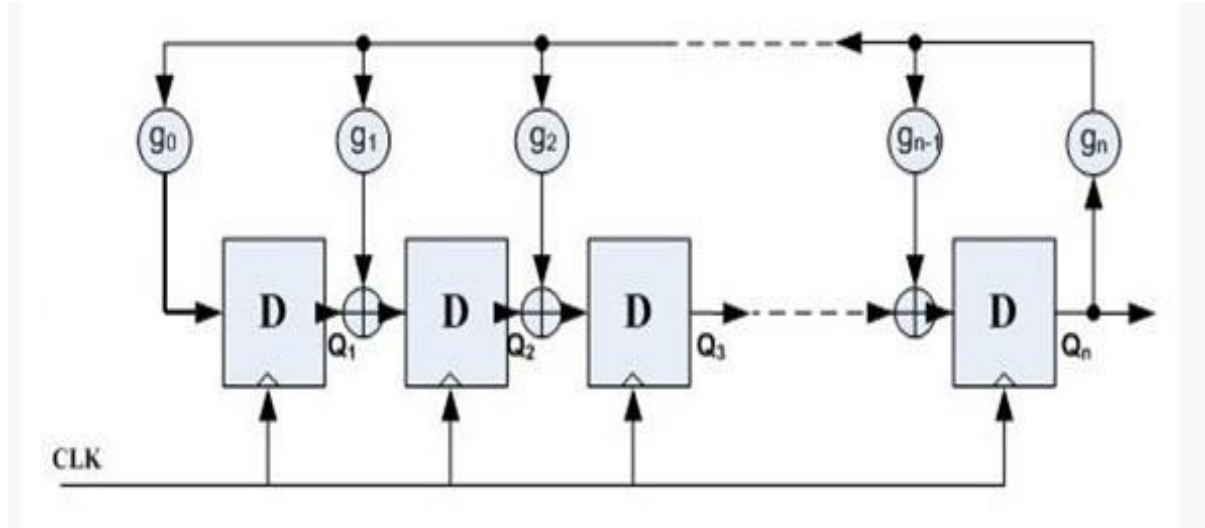
Cnt	LD	HH1	HH0	MM1	MM0	SS1	SS0	HH_H	HH_L	MM_H	MM_L	SS_H	SS_L	LED_R	LED_G	LED_B	INErr
00	1	2	3	5	9	4	9	0	0	0	0	0	0	0	0	0	0
01	0	2	3	5	9	4	9	2	3	5	9	4	9	0	0	0	0
02	0	0	0	0	0	0	0	2	3	5	9	5	0	0	0	0	0
03	0	0	0	0	0	0	0	2	3	5	9	5	1	0	0	0	0
04	0	0	0	0	0	0	0	2	3	5	9	5	2	0	0	0	0
05	0	0	0	0	0	0	0	2	3	5	9	5	3	0	0	0	0
06	0	0	0	0	0	0	0	2	3	5	9	5	4	0	0	0	0
07	0	0	0	0	0	0	0	2	3	5	9	5	5	0	0	0	0
08	0	0	0	0	0	0	0	2	3	5	9	5	6	0	0	0	0
09	0	0	0	0	0	0	0	2	3	5	9	5	7	0	0	0	0
0a	0	0	0	0	0	0	0	2	3	5	9	5	8	0	0	0	0
0b	0	0	0	0	0	0	0	2	3	5	9	5	9	0	0	0	0
0c	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0d	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
0e	0	0	0	0	0	0	0	0	0	0	0	0	2	0	1	0	0
0f	0	0	0	0	0	0	0	0	0	0	0	0	3	0	1	1	0
10	0	0	0	0	0	0	0	0	0	0	0	0	4	1	1	1	0
11	1	1	1	5	9	5	8	0	0	0	0	0	5	1	0	1	0
12	0	0	0	0	0	0	0	1	1	5	9	5	8	0	0	1	0
13	0	0	0	0	0	0	0	1	1	5	9	5	9	0	0	0	0
14	0	0	0	0	0	0	0	1	2	0	0	0	0	1	0	0	0
15	0	0	0	0	0	0	0	1	2	0	0	0	1	1	1	0	0
16	0	0	0	0	0	0	0	1	2	0	0	0	2	0	1	0	0
17	1	2	4	0	0	0	0	1	2	0	0	0	3	0	1	1	1
18	1	2	3	6	0	0	0	1	2	0	0	0	4	1	1	1	1
19	0	2	3	6	0	0	0	1	2	0	0	0	5	1	0	1	0
1a	1	1	2	3	4	5	6	1	2	0	0	0	6	0	0	1	0
1b	1	2	3	4	5	6	7	1	2	3	4	5	6	0	0	0	1
1c	0	0	0	0	0	0	0	1	2	3	4	5	7	1	0	0	0
1d	0	0	0	0	0	0	0	1	2	3	4	5	8	1	1	0	0
1e	0	0	0	0	0	0	0	1	2	3	4	5	9	0	0	0	0
1f	0	0	0	0	0	0	0	1	2	3	5	0	0	0	0	0	0

四、思考题

1、级联 2 个四位移位寄存器子电路实现生成 8 位二进制伪随机数生成电路

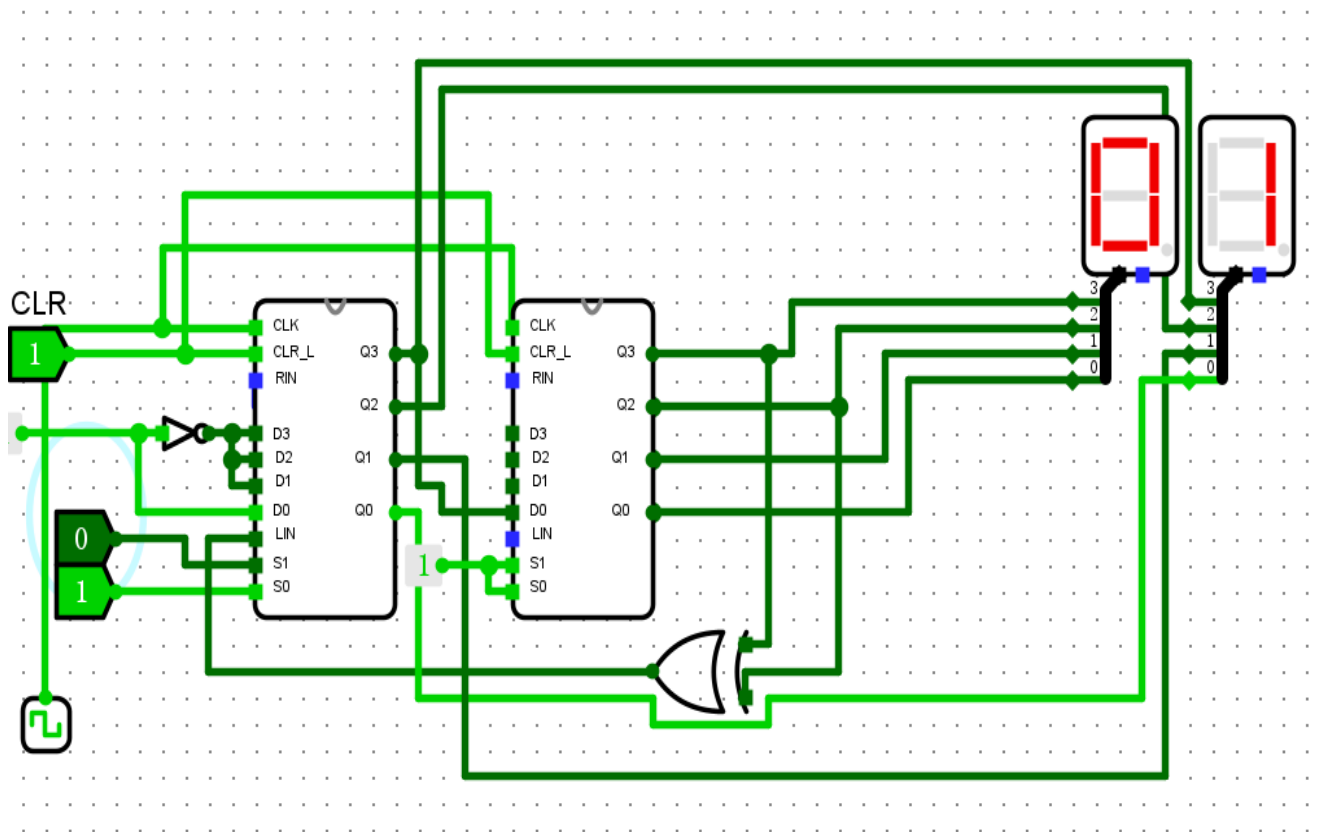
整体模块设计：

(查阅资料所得) 产生伪随机数的方法最常见的是利用 LFSR,它是由 n 个 D 触发器和若干个异或门组成的。其中, g_n 为反馈系数, 取值只能为 0 或 1, 取为 0 时表明不存在该反馈之路, 取为 1 时表明存在该反馈之路; n 个 D 触发器最多可以提供 $2^n - 1$ 个状态(不包括全 0 的状态), 为了保证这些状态没有重复, g_n 的选择必须满足一定的条件。

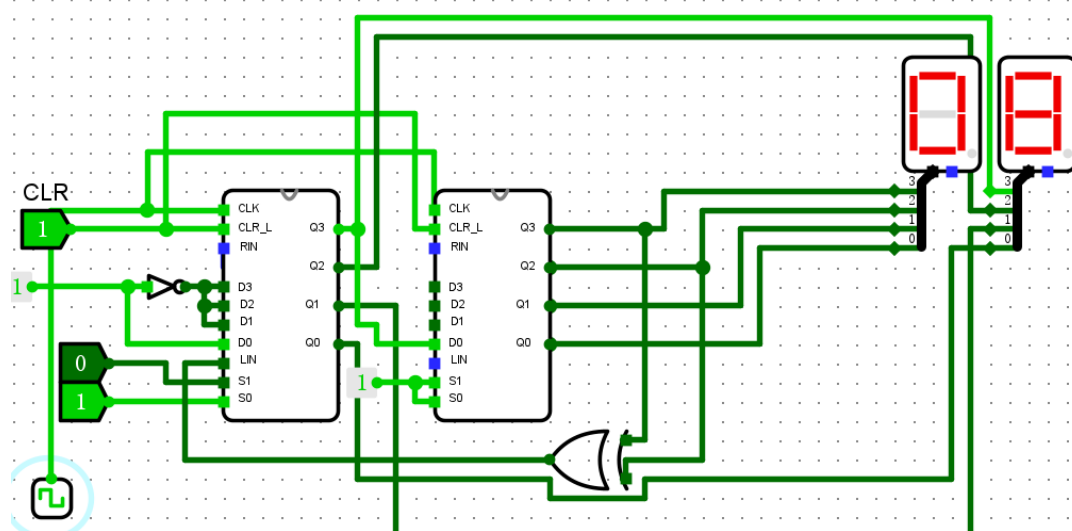
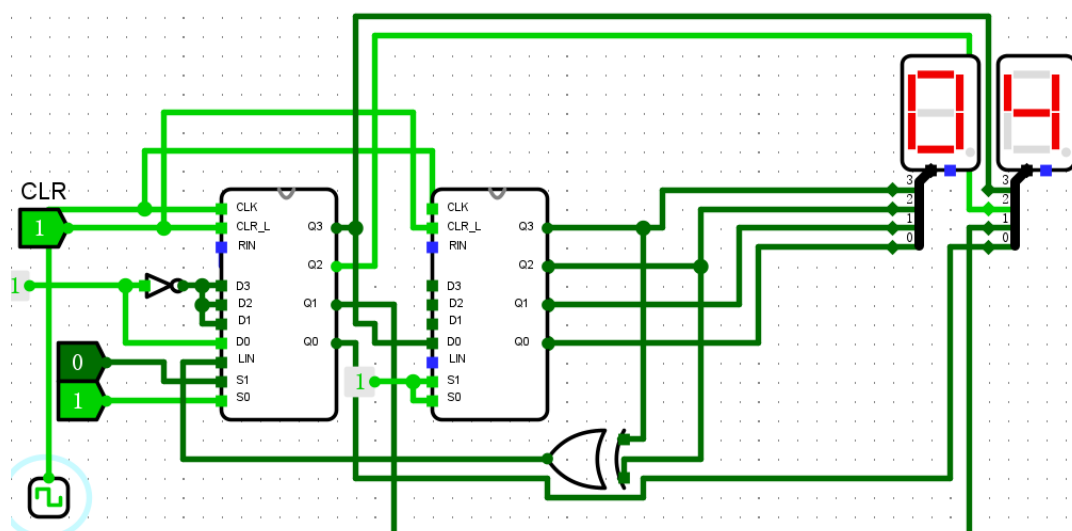
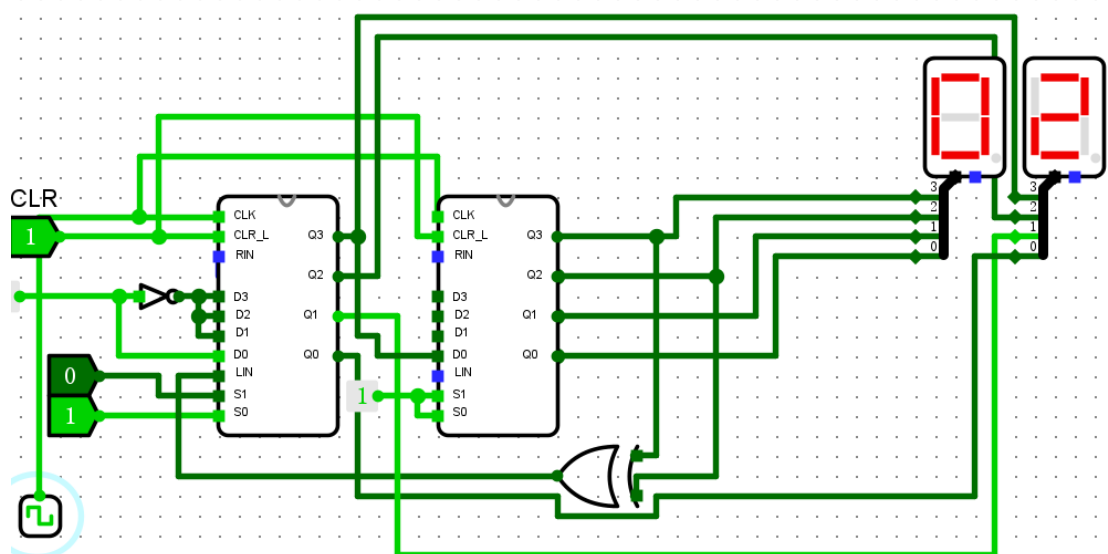


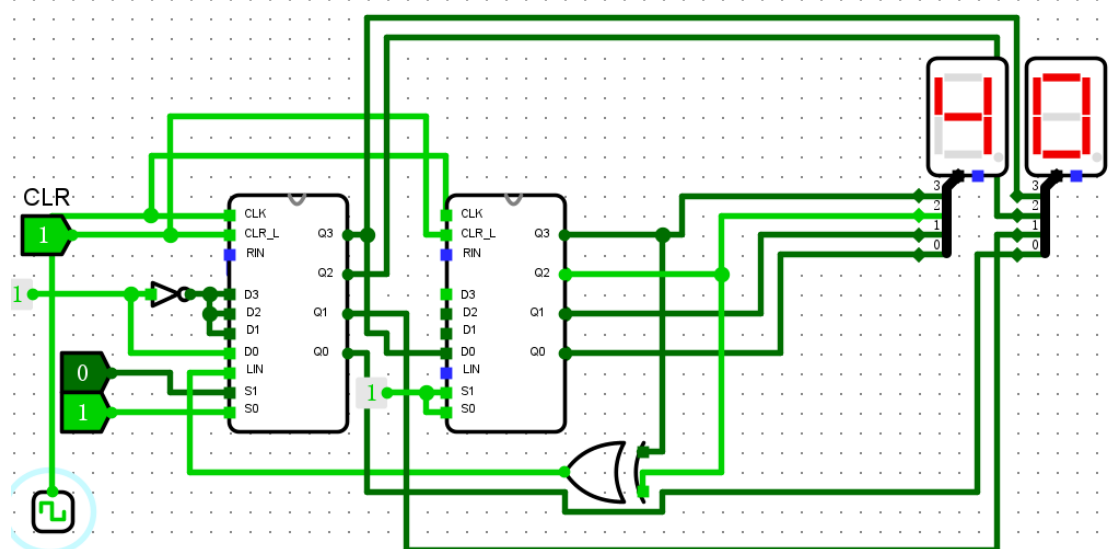
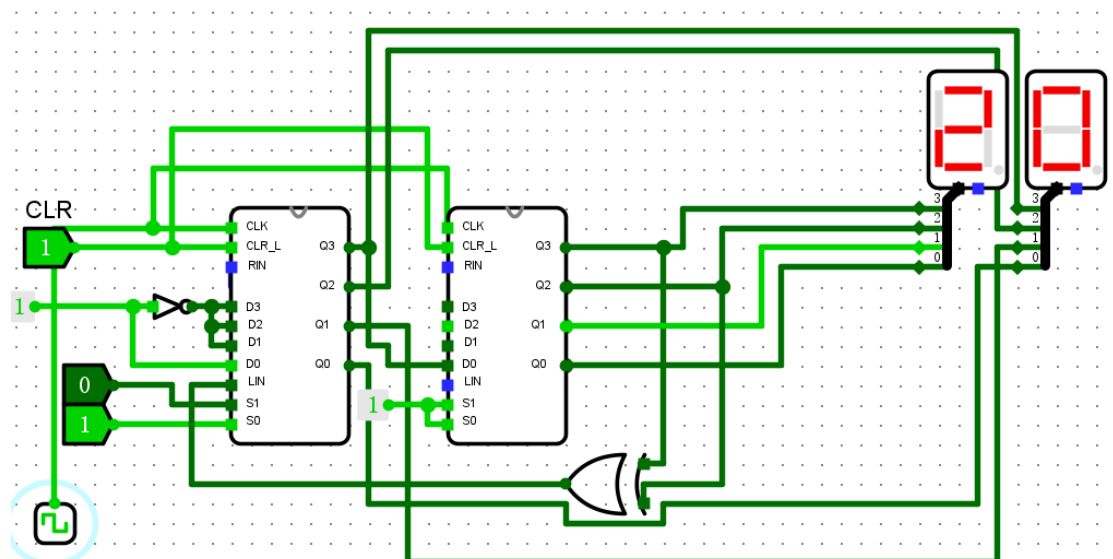
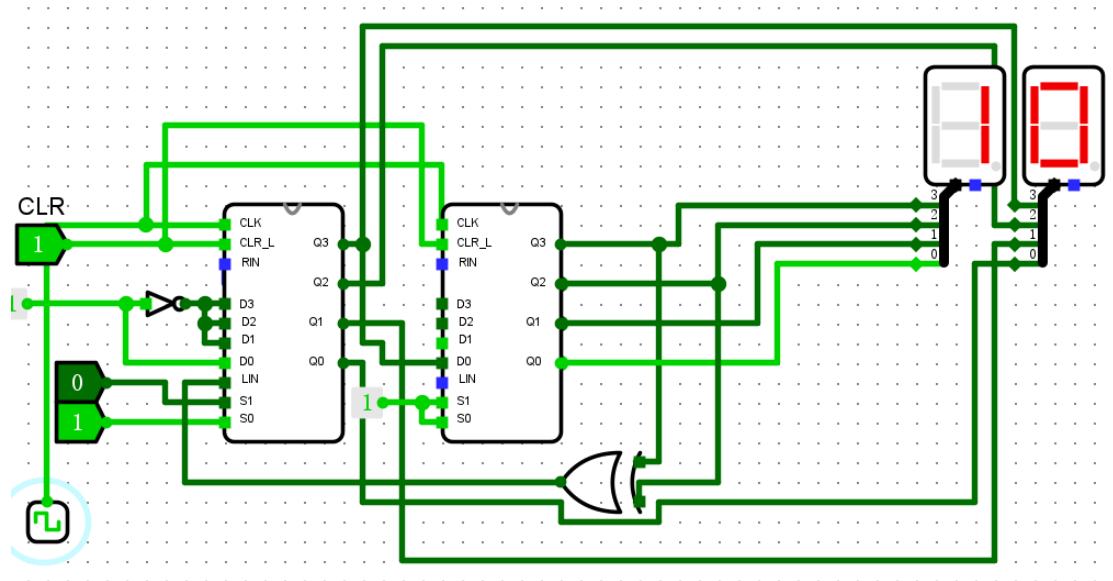
采用左移模式, 反馈方程 $LIN = Q_7 \oplus Q_6$, 生成一串伪随机数序列 1, 2, 4, 8, 10, 20, 40, 81, ...

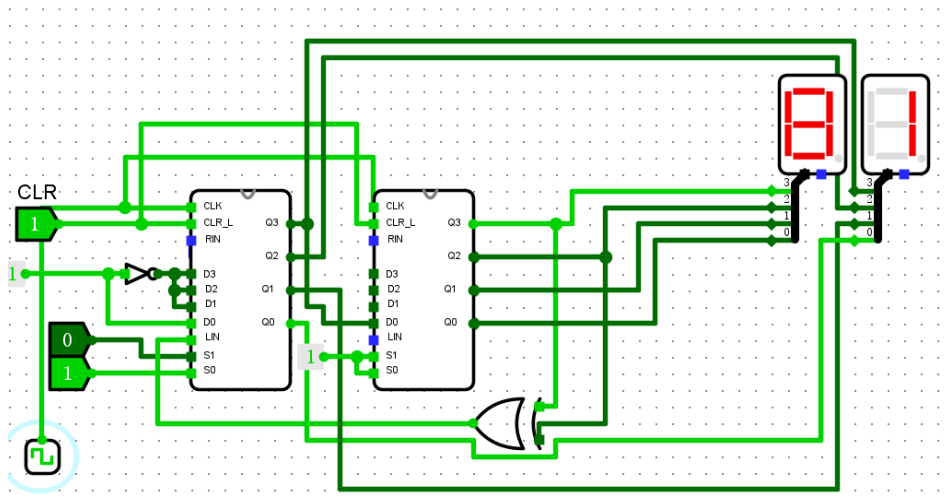
电路图：



仿真测试图：





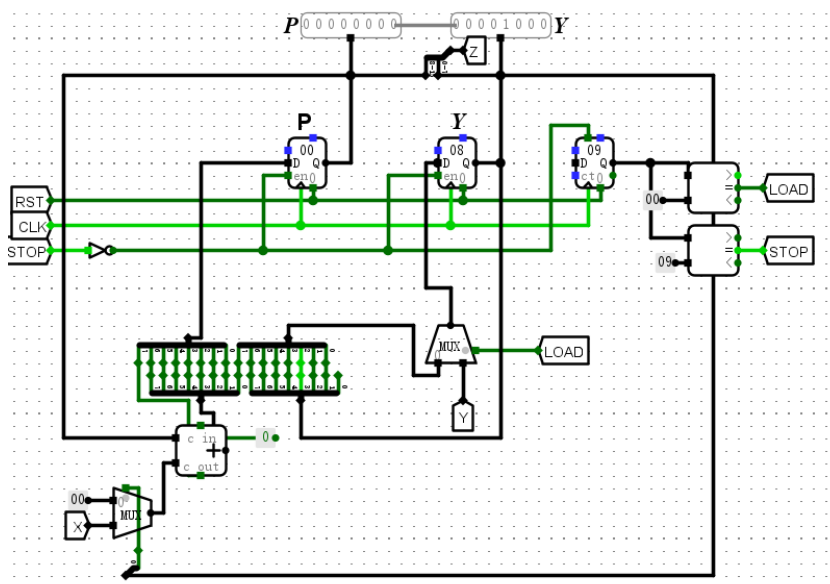


2、使用加法器实现 8 位无符号数的快速乘法器

整体模块设计：

使用同样的思路，无符号一位乘法，通过右移做加法实现 8 位无符号数的乘法。

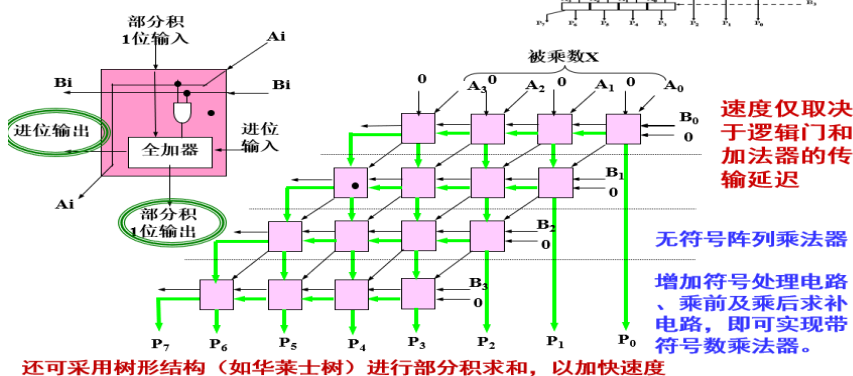
电路图：（在实验三上有所改进）



网上学习所得：

CRA阵列乘法器（不要求）

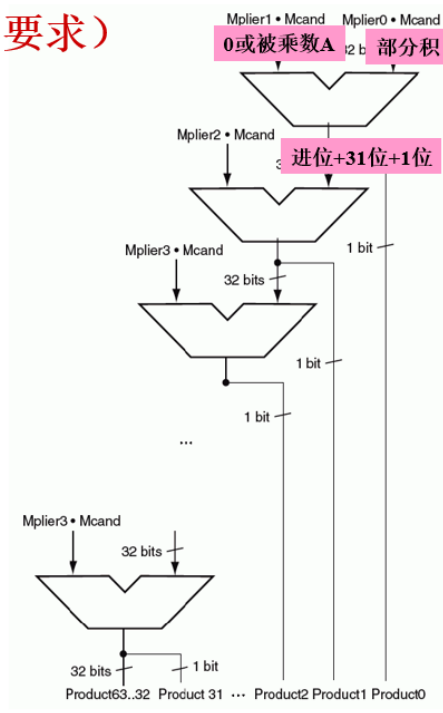
◆ 阵列乘法器：“细胞”模块的阵列



流水线快速乘法器（不要求）

- ◆ 为乘数的每位提供一个n位加法器
- ◆ 每个加法器的两个输入端分别是：
 - 本次乘数对应的位与被乘数相与的结果（即：0或被乘数）
 - 上次部分积
- ◆ 每个加法器的输出分为两部分：
 - 和的最低有效位 (LSB) 作为本位乘积
 - 进位和高31位的和组成一个32位数作为本次部分积

像流水一样，完全是串行，浪费加法器资源——但是，组合逻辑电路！无需控制器控制

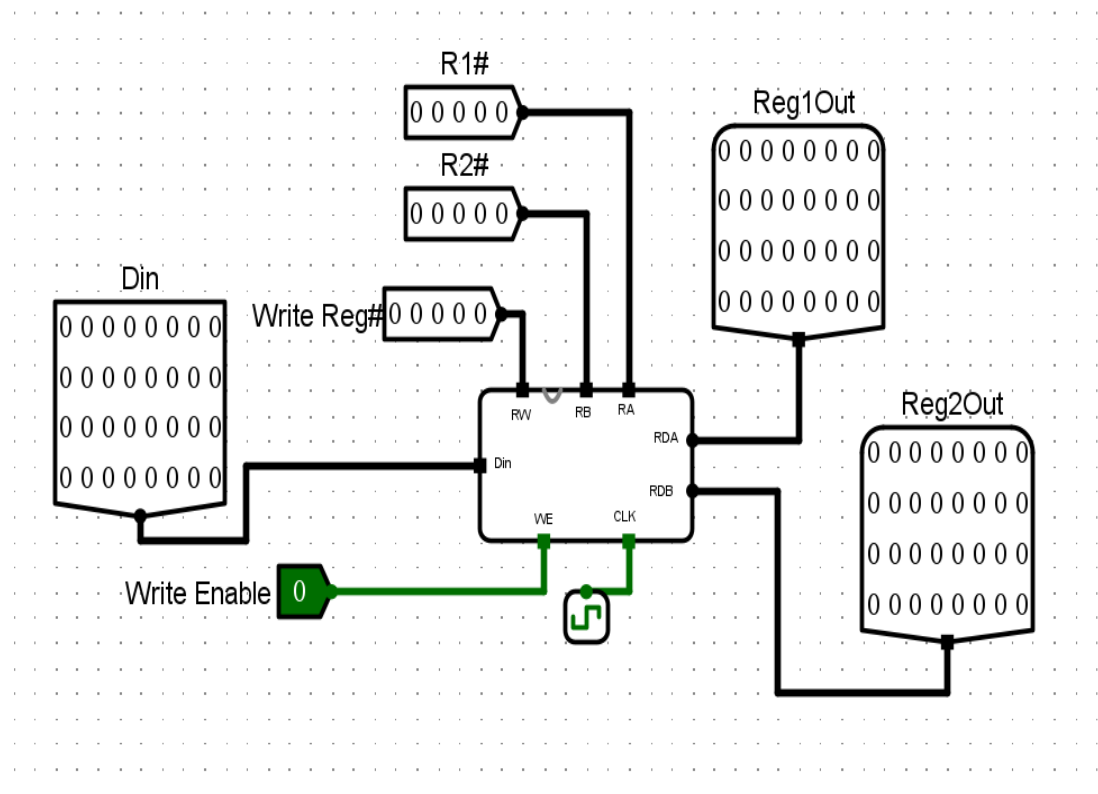


3、修改寄存器堆的设计电路，将输入信号分别连接寄存器的使能端和时钟端，验证寄存器堆的读写功能，分析使能信号和写入地址信号的先后时序关系变化是否影响到写入结果。

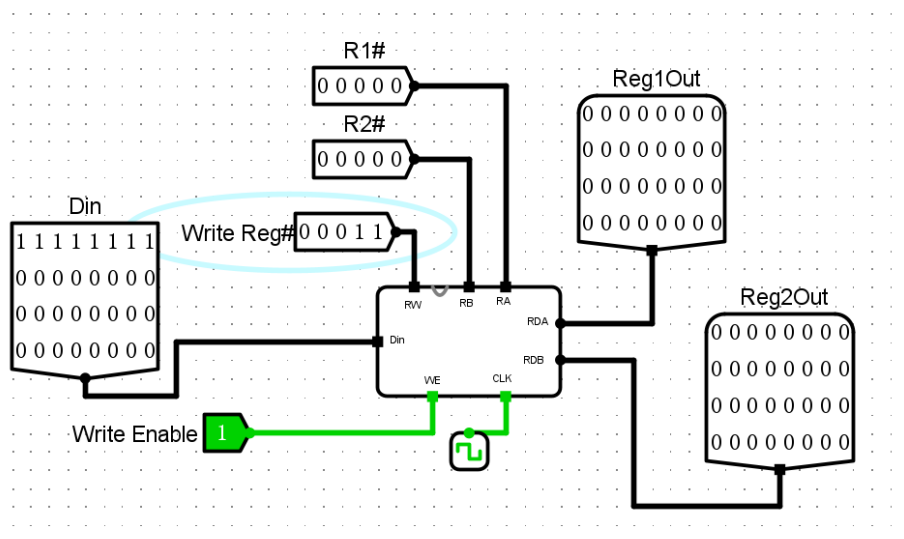
整体模块设计：

调用寄存器堆的子电路，放置地址端口，数据端口，时钟和使能部件实现读写功能。

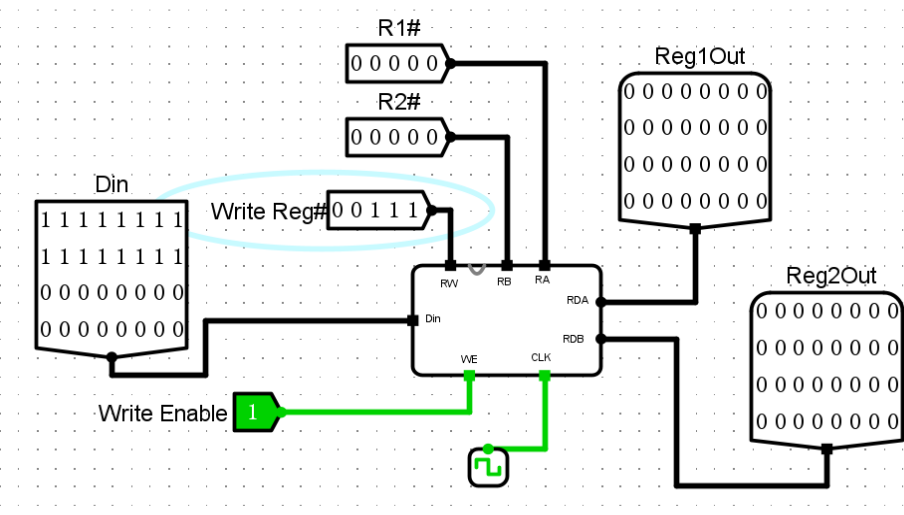
原理图（同时电路图）：



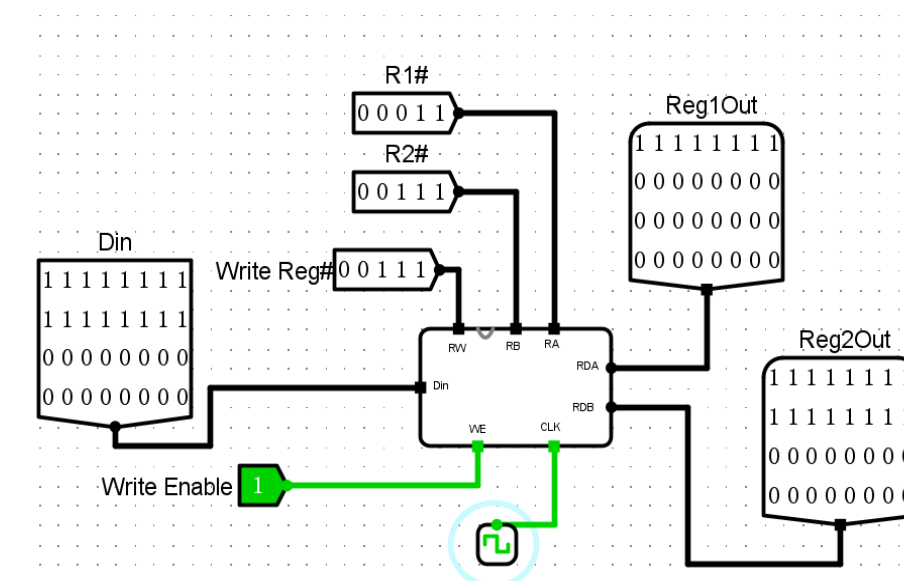
仿真测试图：（选择时钟单步方式进行仿真）先写入三号寄存器



(再写入七号寄存器)



读取三号和七号寄存器的数据



可能的变化和影响：

使能信号先于时钟信号：如果使能信号在时钟信号到来之前被激活，那么写入操作将在时钟上升沿到来时立即执行。这可能导致写入的数据与期望的时钟触发时刻不一致，可能会产生意外结果。

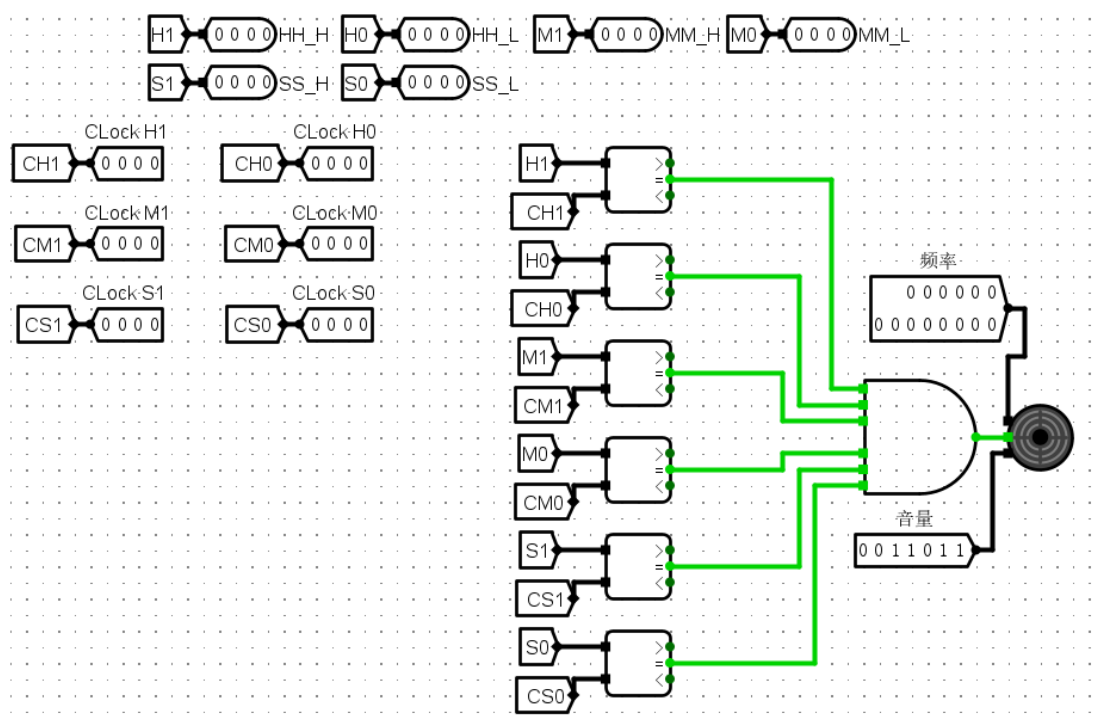
时钟信号先于使能信号：如果时钟信号在使能信号到来之前到来，那么写入操作将在使能信号激活后立即执行。这样的设计可能更符合预期的写入时序，但仍需要谨慎考虑时序关系带来的影响。

4、在数字时钟设计中如何添加闹钟的功能。

整体模块设计：

可以用比较器实现闹钟功能，也即是用比较器将时分秒逐一与闹钟设定的时间进行比较，时间到达闹钟所设定的时间，蜂鸣器响

电路图：（可根据喜好调整音量与频率）



五、错误现象及分析

实验连线较多，一开始经常会出现线路交叉的情况，后来注意观察线路有没有圆圈，且在线路交叉的地方预留多一点的空间，线路就会变得整洁了很多，出错的情况也少了。