《数字逻辑与计算机组成》实验报告二. 组合逻辑电路设计

1 实验

1.1 计数器实验

1.1.1 实验内容

构建 4 位二进制同步计数器电路。

1.1.2 实验整体方案设计

根据功能表和原理图可以做出。本实验不需要顶层模块设计图。

LD	LD 置 1,CLR 置 0 时,载入输入的值
CLR	置 1 清零计数器的值
ENT,ENP	有一个为 0 时,保持存储值;都为 1 时,开始计数
Q3,Q2,Q1,Q0	四个输出端次态计数

表 1: 计数器引脚作用

1.1.3 实验原理图和电路图

原理图实现:

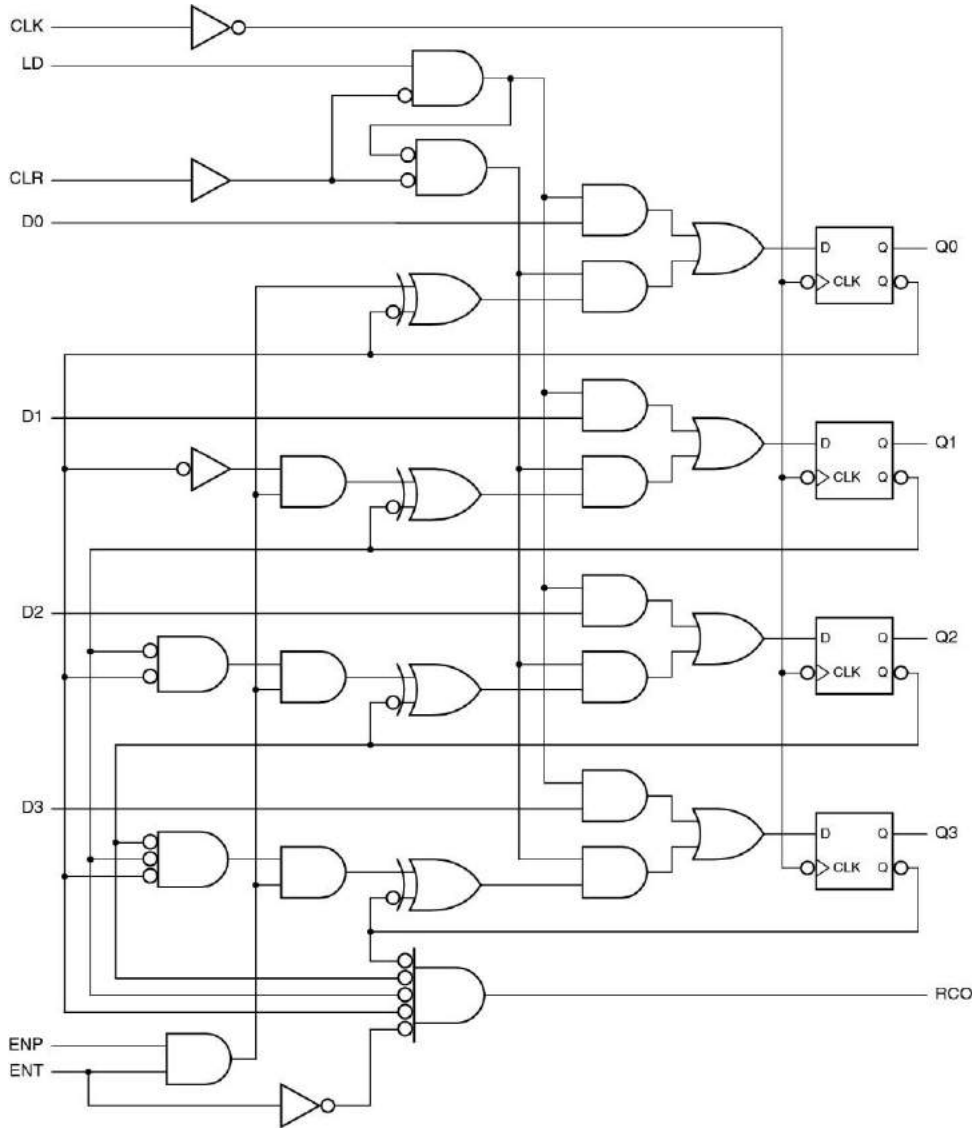

图 1: 计数器原理图

电路图实现:

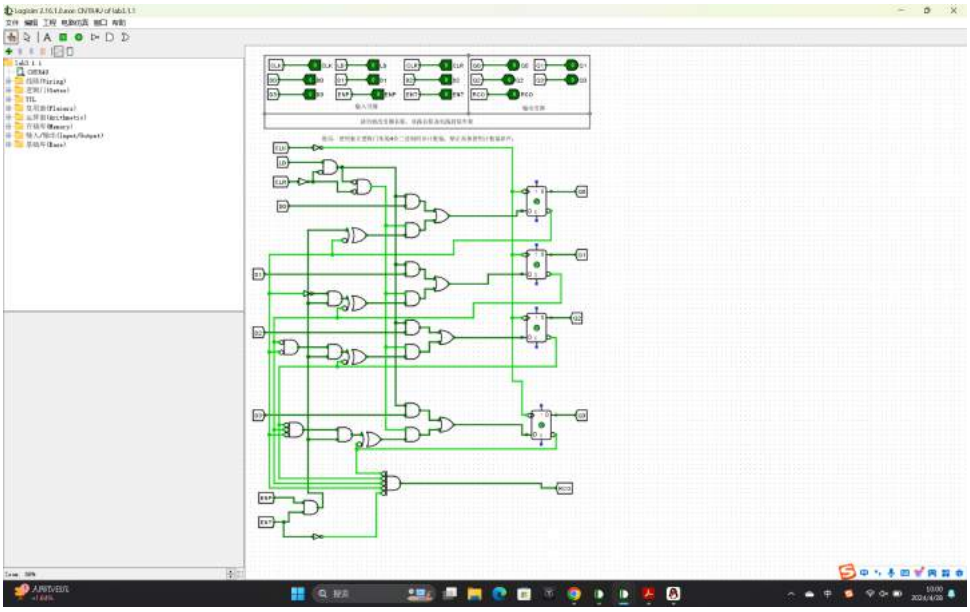

图 2: 计数器电路图

1.1.4 实验数据仿真测试图

分别展示了时钟信号驱动下对应的各种变化，从 LOAD 到 CLEAR 再到计数。(以上均为 clock 时钟上沿触发)

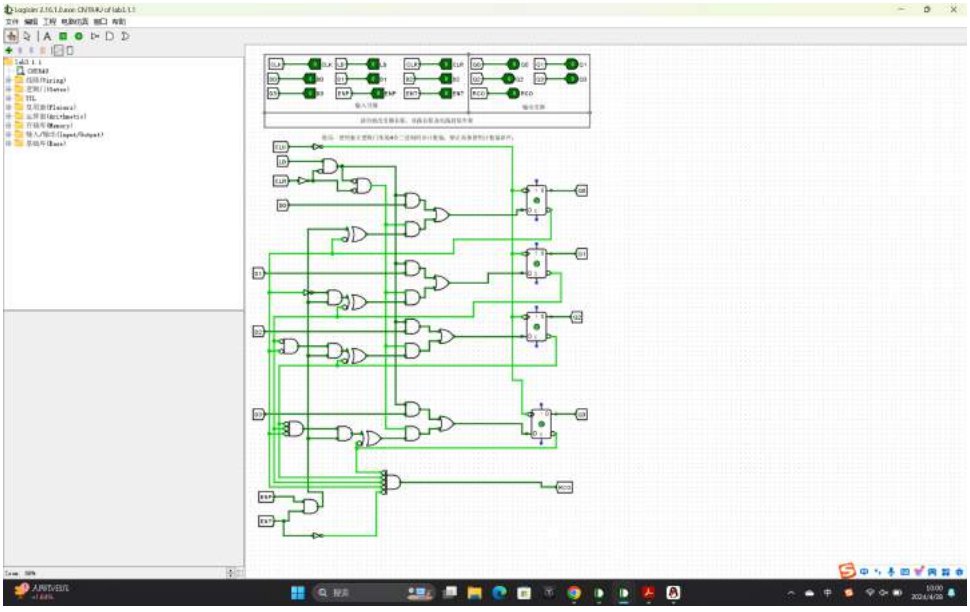

图 3: 初始状态

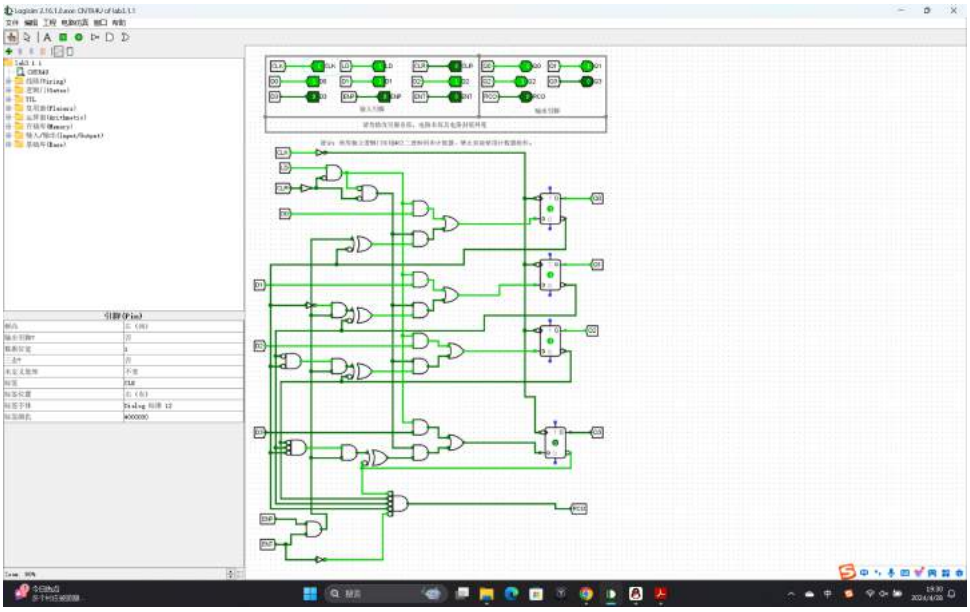

图 4: LD 功能

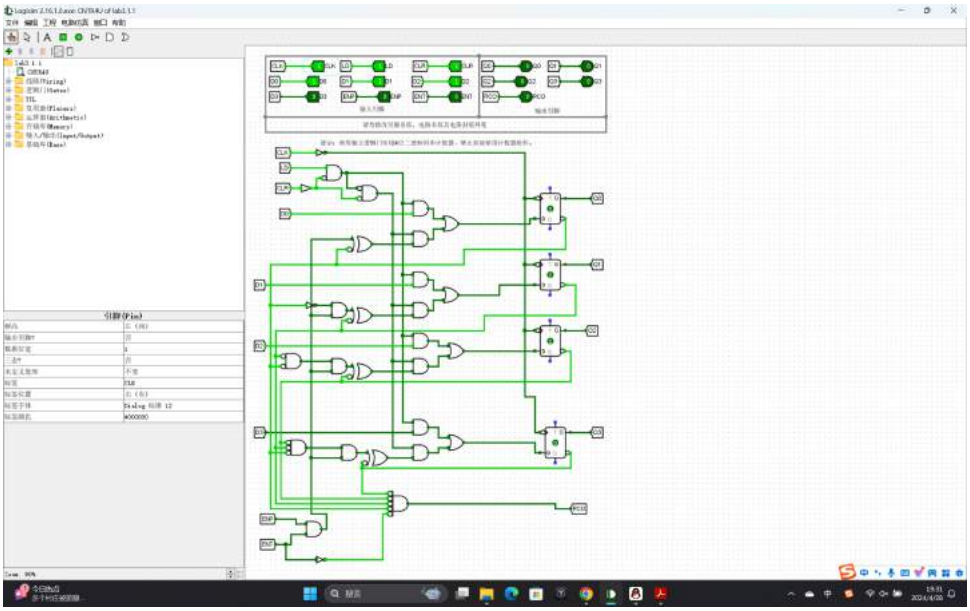

图 5: CLR 功能

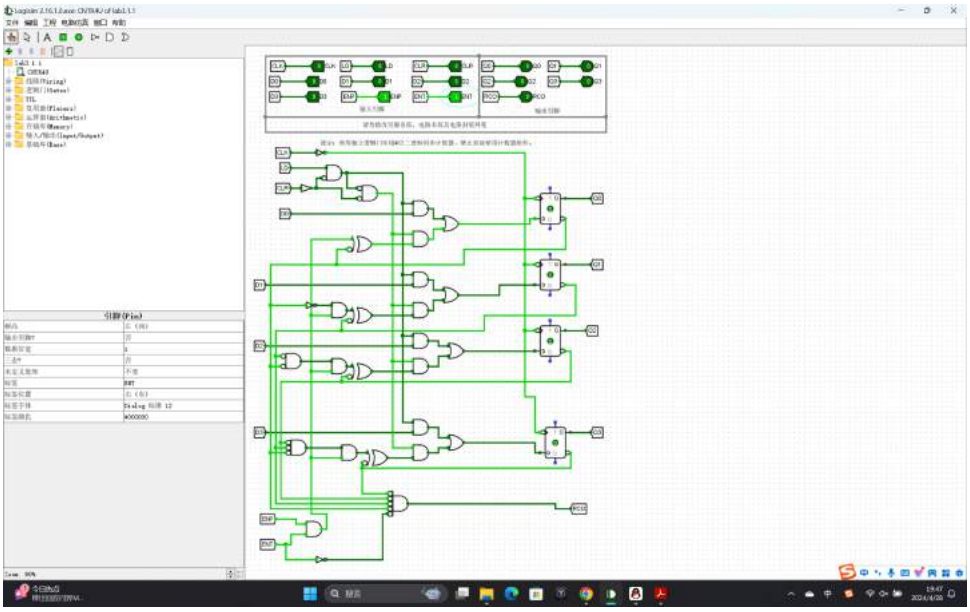

图 6: 计数器初始状态

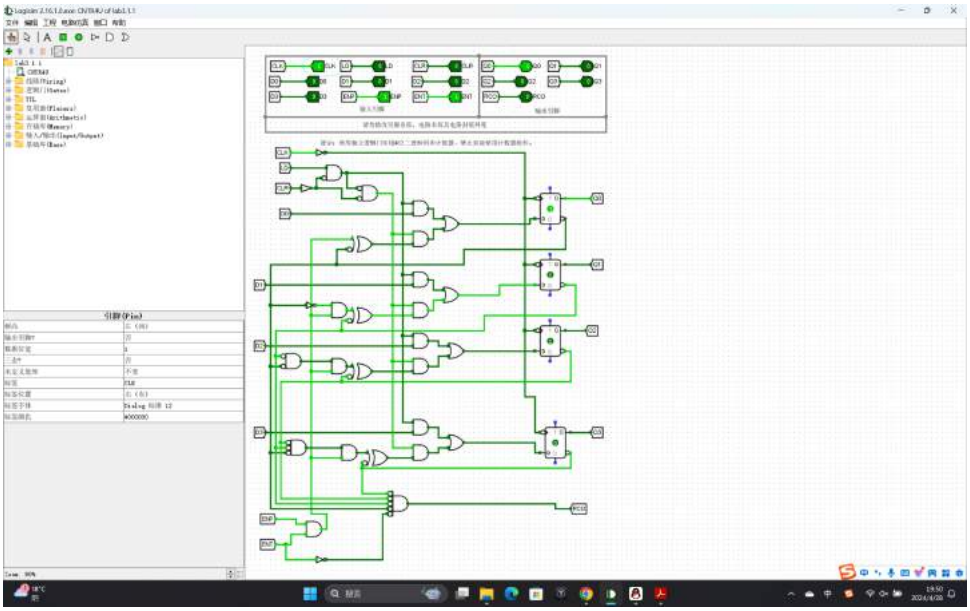

图 7: 第一次计数

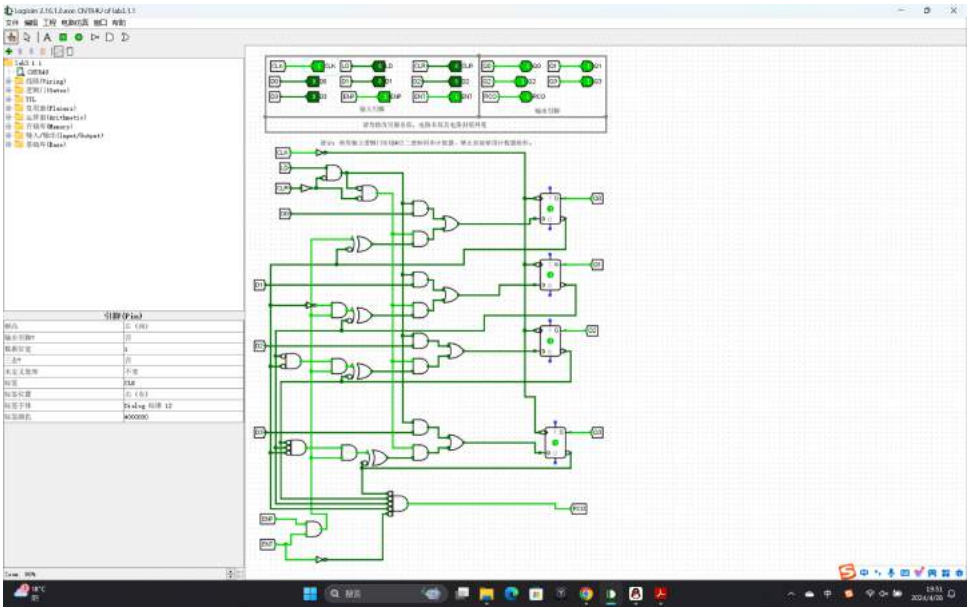

图 8: 最后一次计数

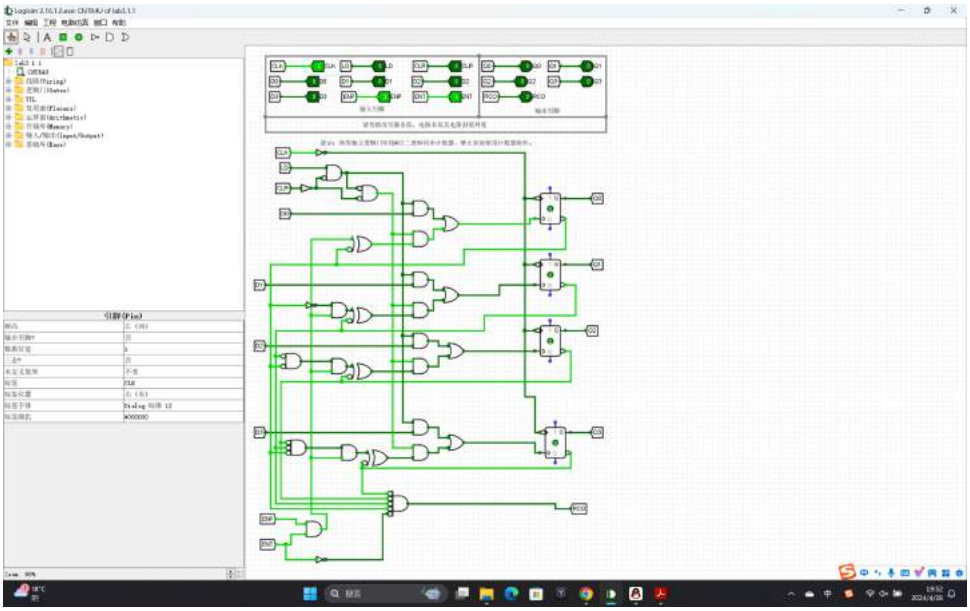

图 9: 回归到初始

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

图 10: 功能表

1.1.5 错误现象及分析

在第一次看到原理图时，忘记了加缓冲器，导致结果出现了错误。后来分析为什么结果有错误，做了时钟单步测试，发现如果没有缓冲器，门电路传递是不同步的，因此可以得出时序的不同会影响时序逻辑电路的结果。（不贴图了）

1.2 移位寄存器实验

1.2.1 实验内容

构建 4 位通用移位寄存器电路 SHRG4U。该移位寄存器带有异步复位（清 0）信号 CLR，它是低电平有效信号，当它为低电平时，所有 D 触发器的状态输出为 0。

1.2.2 实验整体方案设计

根据原理图可以画出。本实验不需要顶层模块设计图。

CLR	清零
S1，S0	两个输入（使能端）控制功能
Q3 Q0	四个次态

表 2: 移位寄存器引脚作用

1.2.3 实验原理图和电路图

原理图：

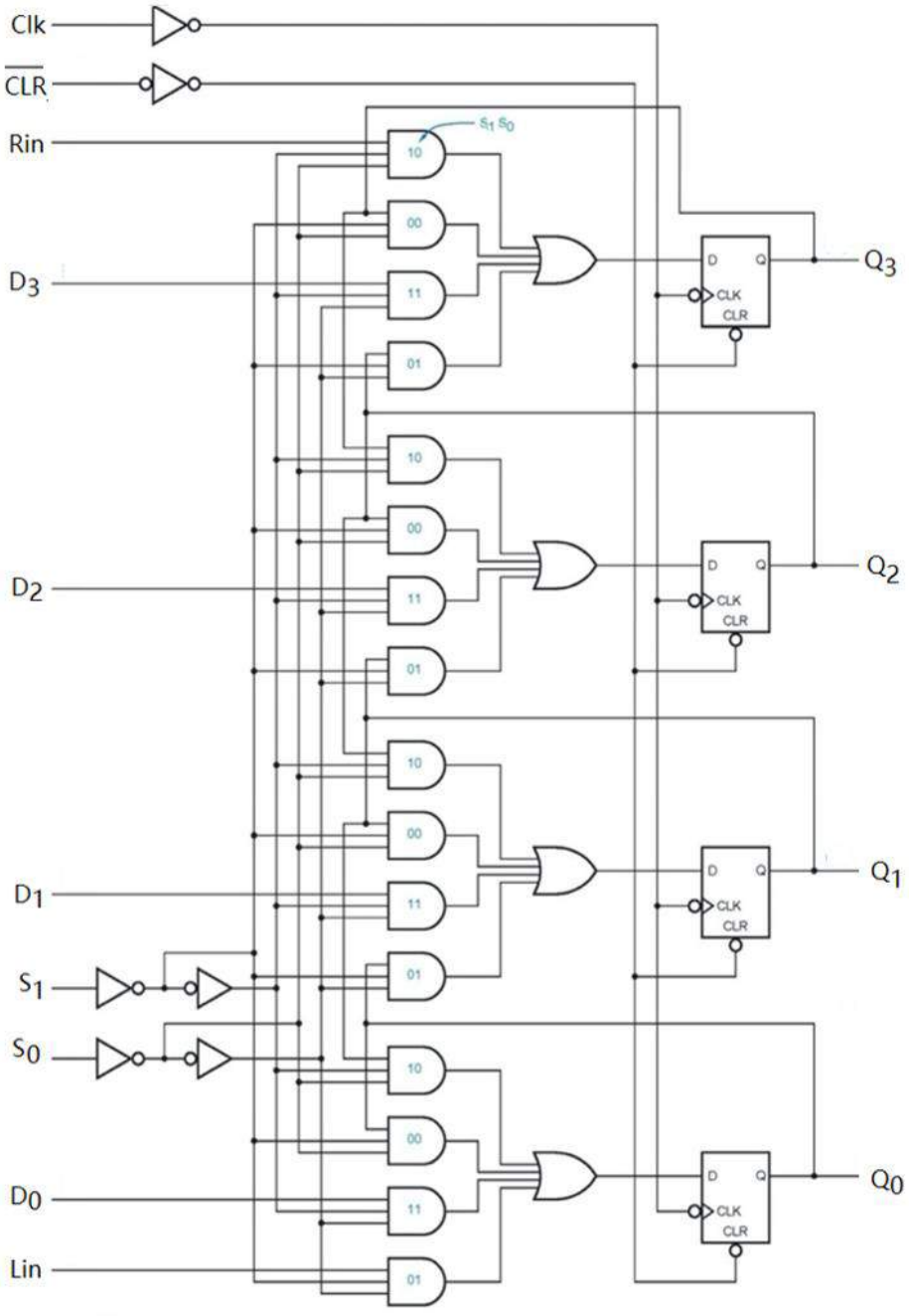

图 11: 移位寄存器原理图

电路图实现：

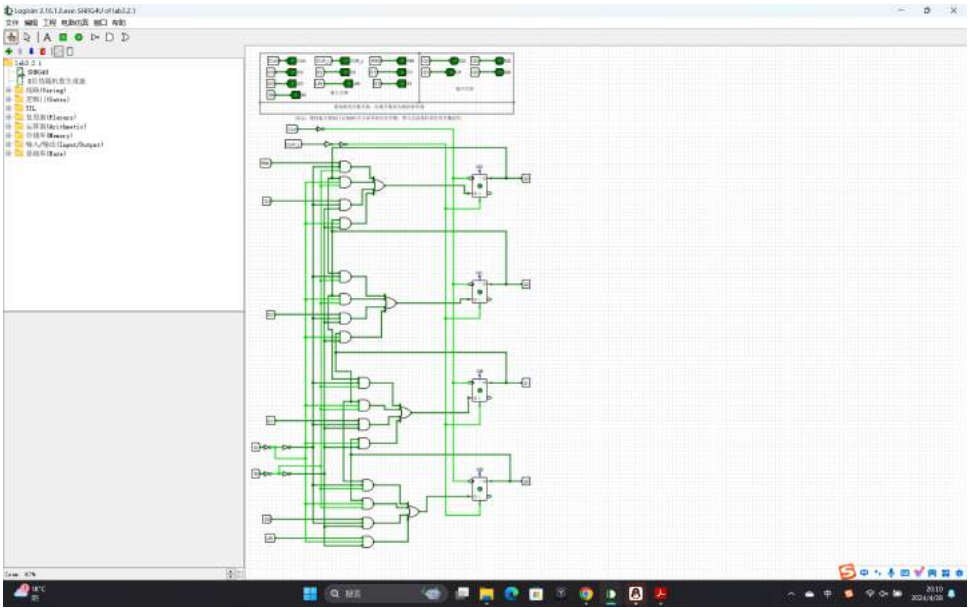

图 12: 移位寄存器电路图

1.2.4 实验数据仿真测试图

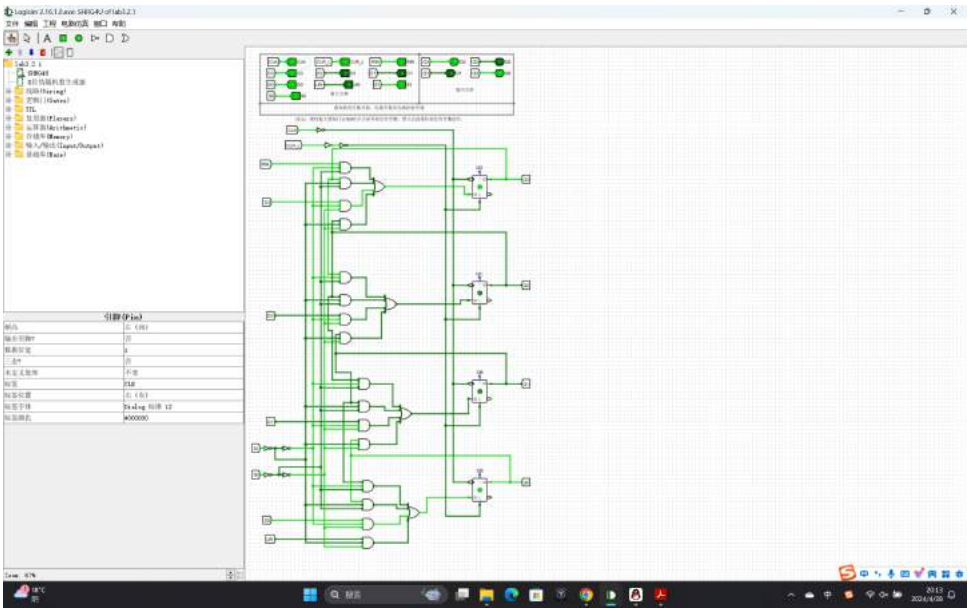

图 13: 装载

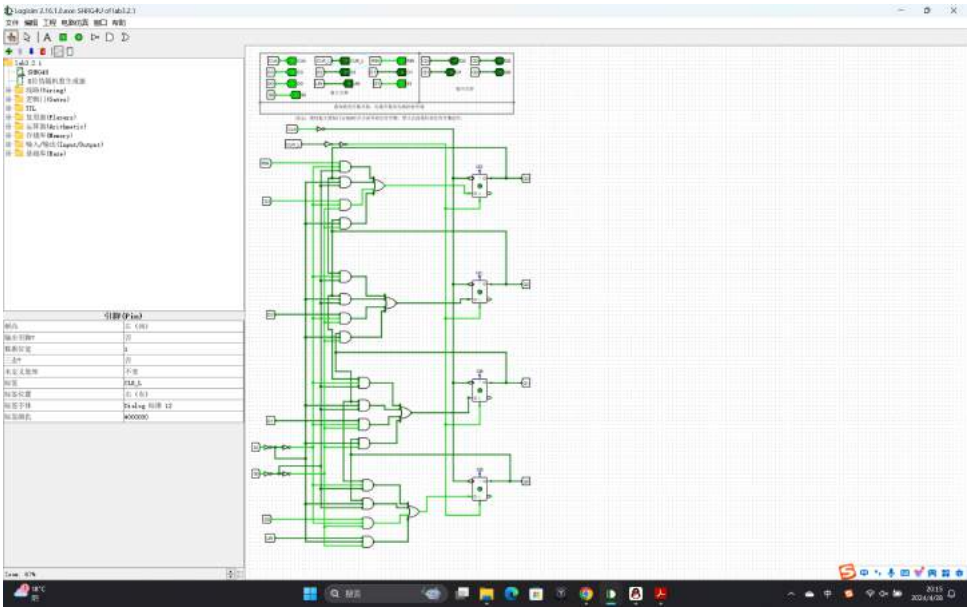

图 14: 清零

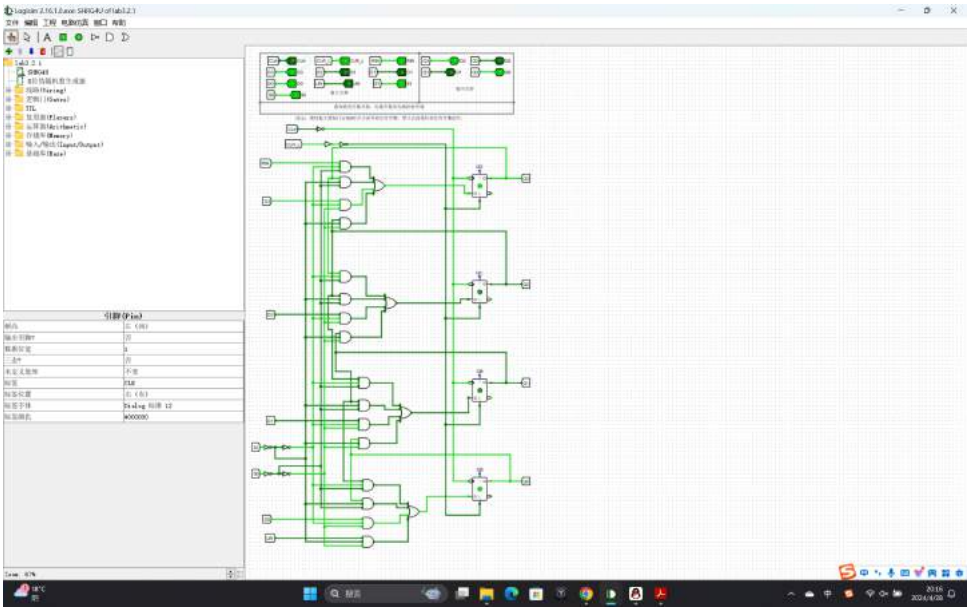

图 15: 保持

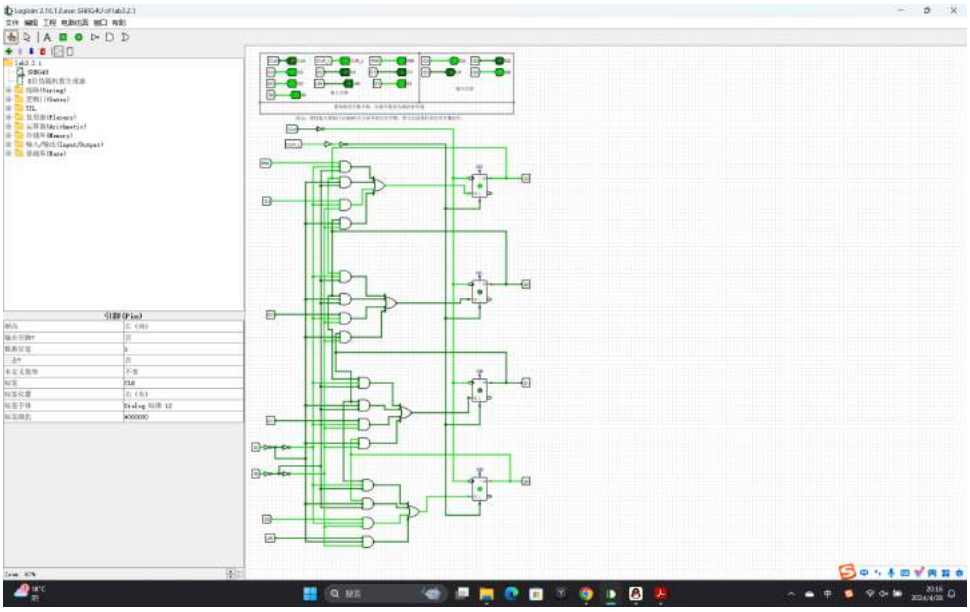

图 16: 保持

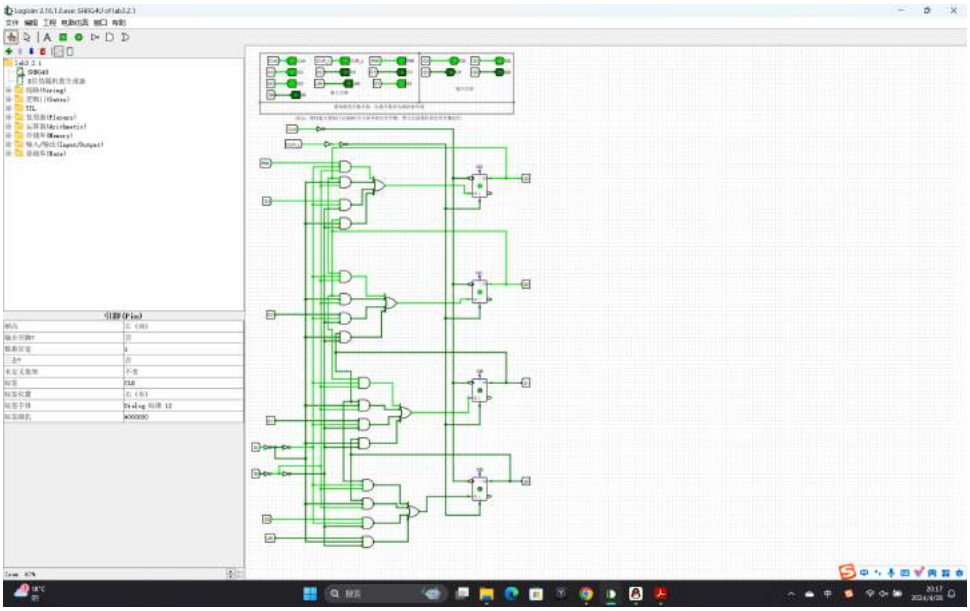

图 17: 右移

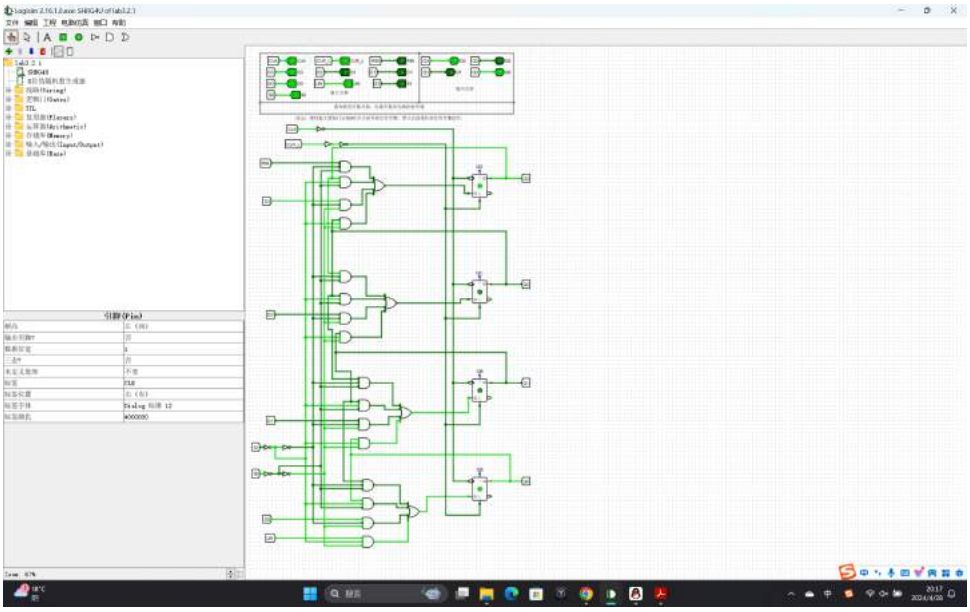

图 18: 左移

功能	CLR	S1	S0	Q3*	Q2*	Q1*	Q0*
清零	0	x	x	0	0	0	0
保持	1	0	0	Q3	Q2	Q1	Q0
右移	1	1	0	RIN	Q3	Q2	Q1
左移	1	0	1	Q2	Q1	Q0	LIN
装载	1	1	1	D3	D2	D1	D0

表 3: 功能表

1.2.5 错误现象及分析

在刚连完后，发现结果不对，发现是缓冲器的图标当成非门了，但是改成缓冲器后发现仍然不对，于是发现测试集是 CLR 为 0 时清零，但是需要 CLR 为 1 触发，因此还是需要一个非门＋缓冲器，但是发现还是过不去，于是无奈重连了一遍，通关了，应该是哪个输入端连到一起了没发现。

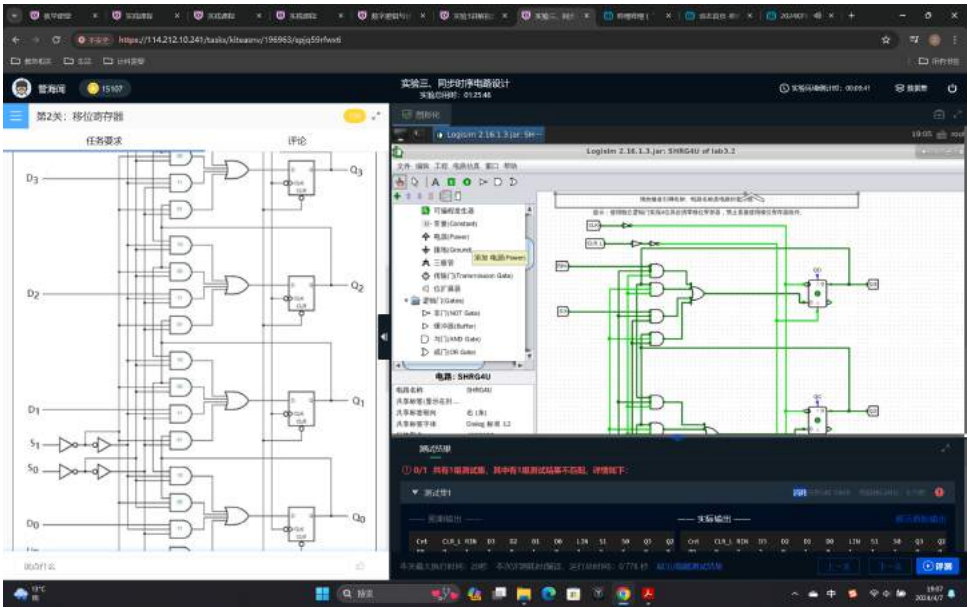

图 19: 错误电路图

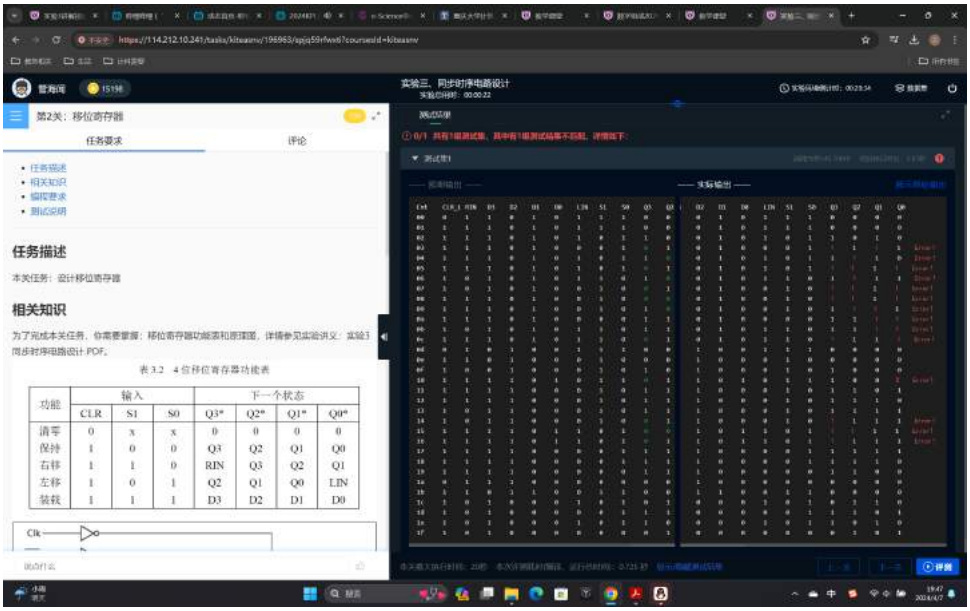

图 20: 错误图片

1.3 四位无符号数乘法器

1.3.1 实验内容

实现两个四位二进制无符号数相乘的功能并通过数码管将其转换成十六进制显示出来。

1.3.2 实验整体方案设计

计数器同上，加减法器，全加器，桶型移位器和上次的实验一样。实现手算一位乘法，比较器和计数器共同实现不同位数的乘积，到第四位停止，计数器为 0 时 load，为 1 4 时进行位数乘法；加减法器，移位器，寄存器和二路选择器共同实现移位对应位数的“乘法”。

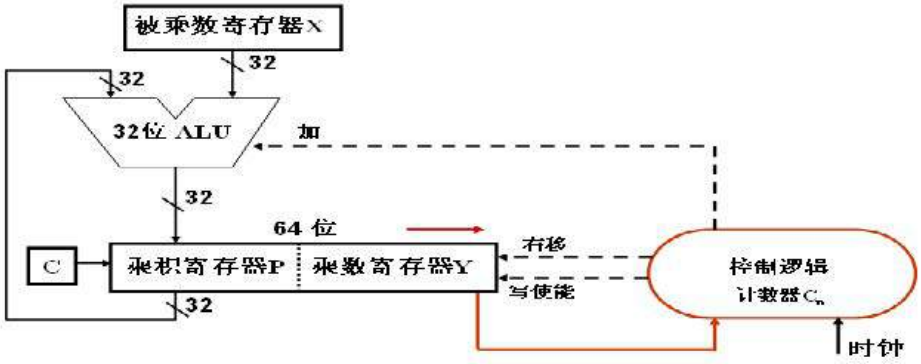

图 21: 思路

按照思路即可。

桶型移位器	乘完后逻辑左移一位(右移也可以实现)
寄存器	存得到的结果(2 个，因为结果 8 位)
STOP，LOAD	控制乘积次数
全加器	实现一位乘法
MUX	控制加(乘)的是什么位(y 还是 0)
Z	输出

表 4: 功能表

1.3.3 实验原理图和电路图

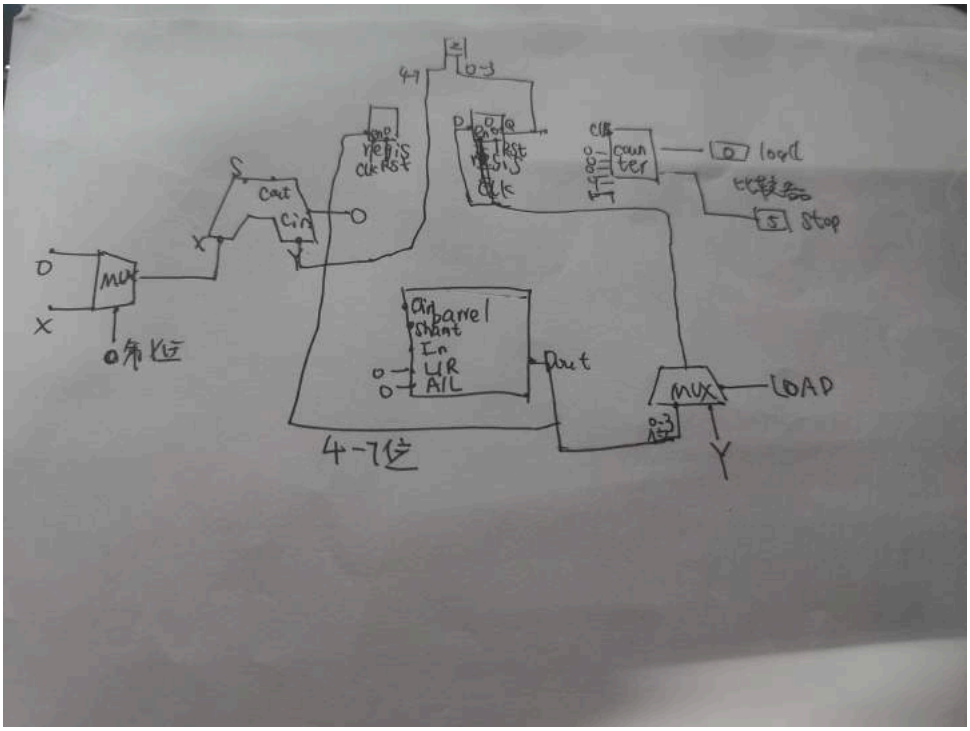

图 22: 乘法器原理图

电路图实现：

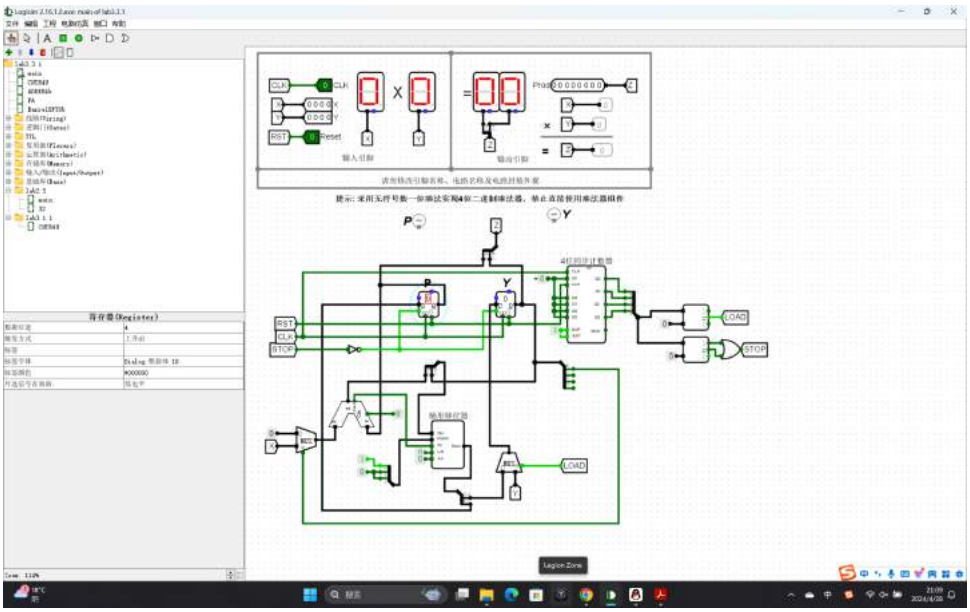

图 23: 乘法器电路图

1.3.4 实验数据仿真测试图

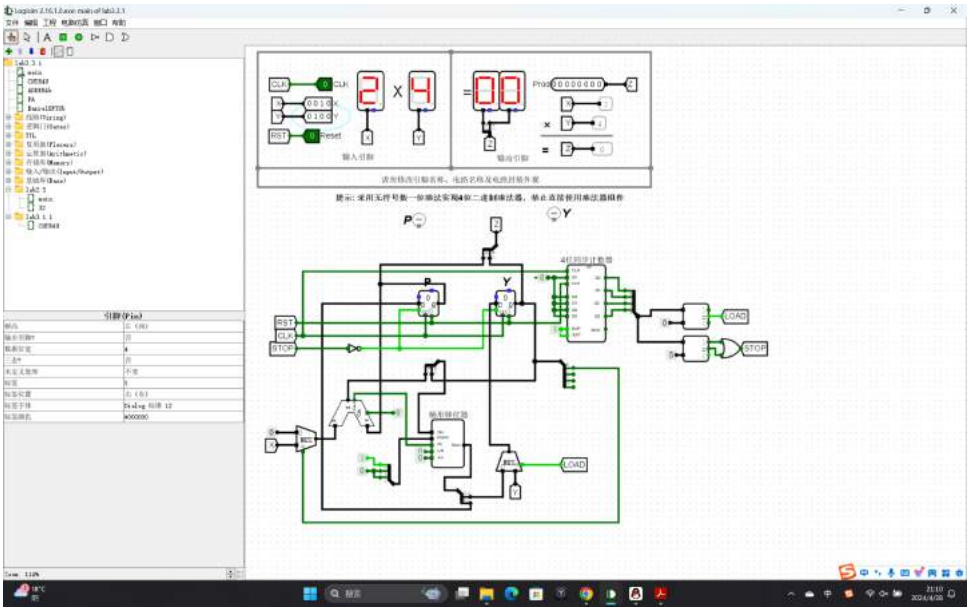

图 24: 初始

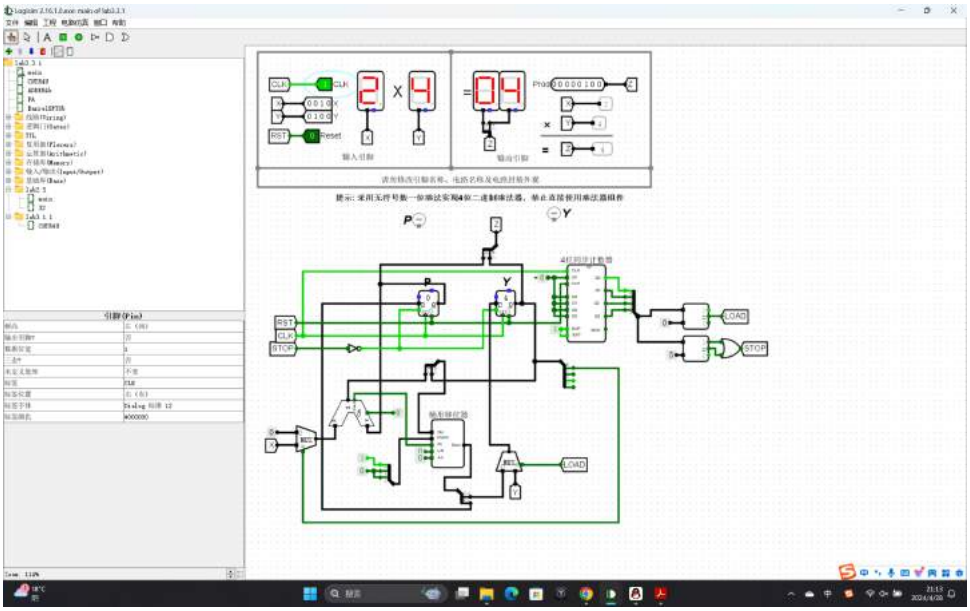

图 25: 第零位(寄存器初始)

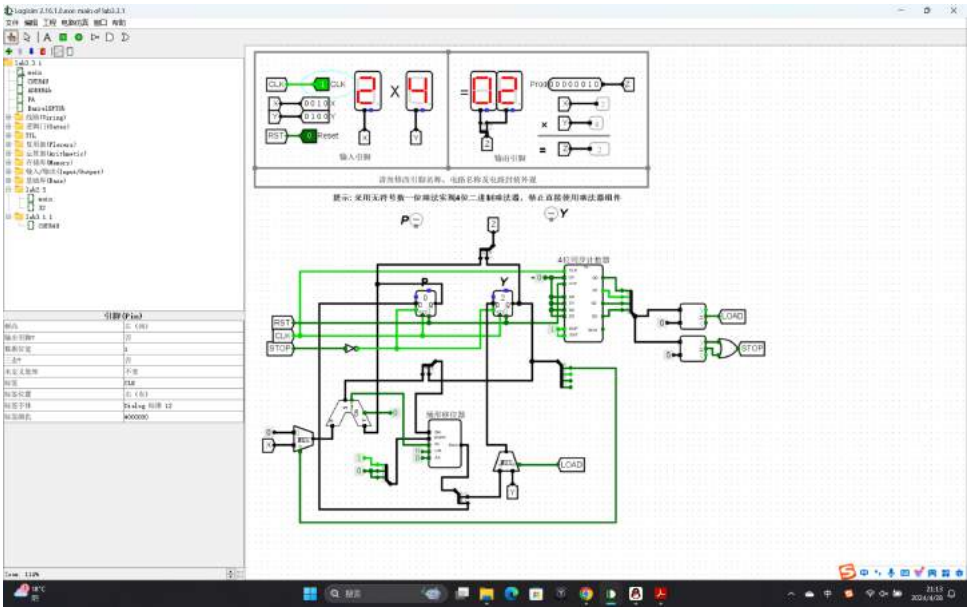

图 26: 第一位

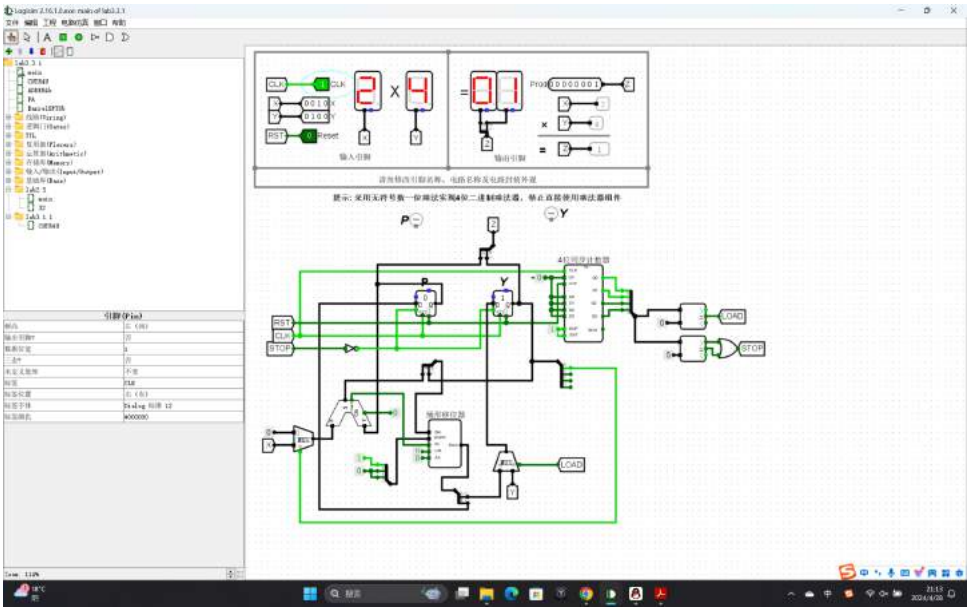

图 27: 第二位

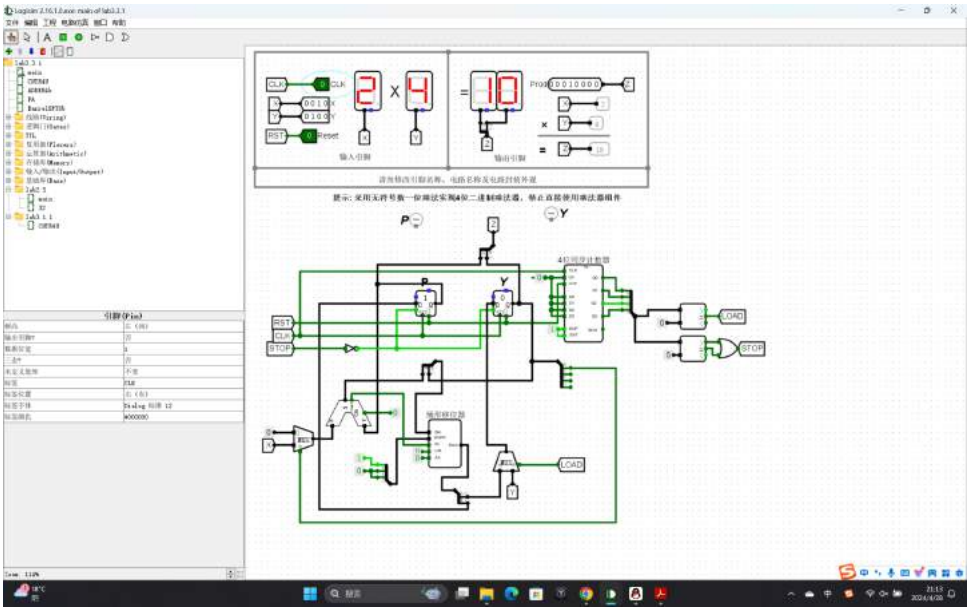

图 28: 第三位

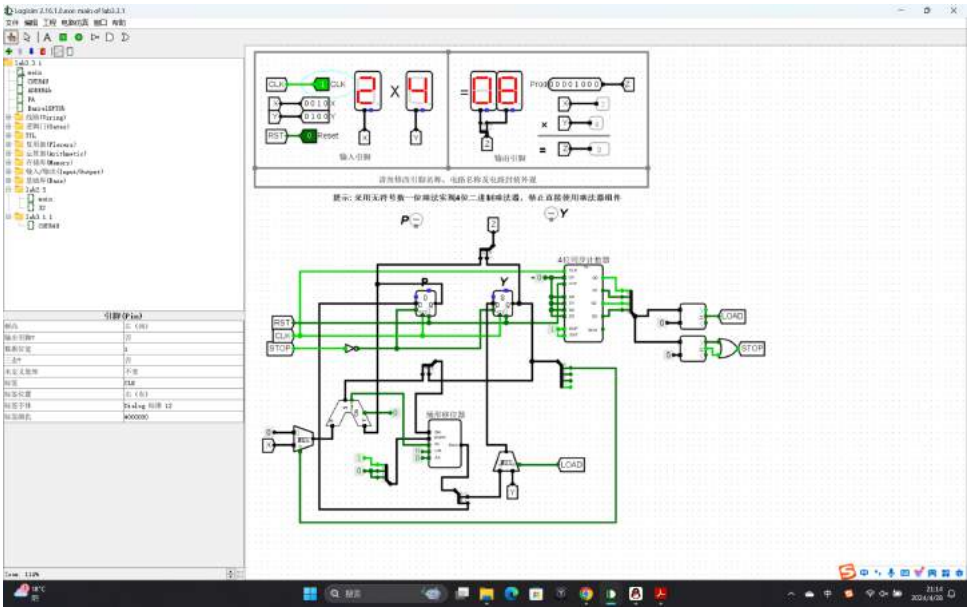

图 29: 第四位

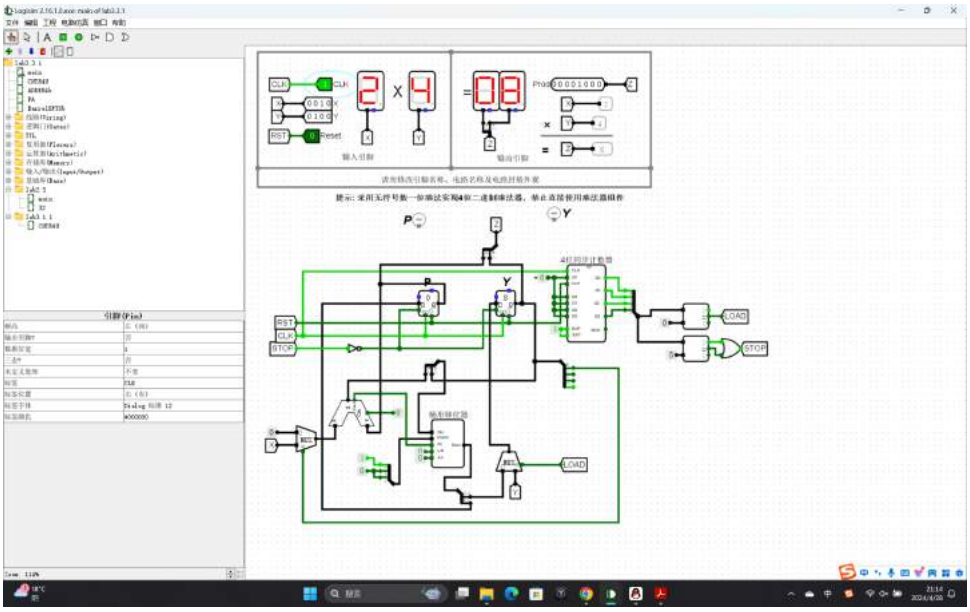

图 30: 以后不再变化

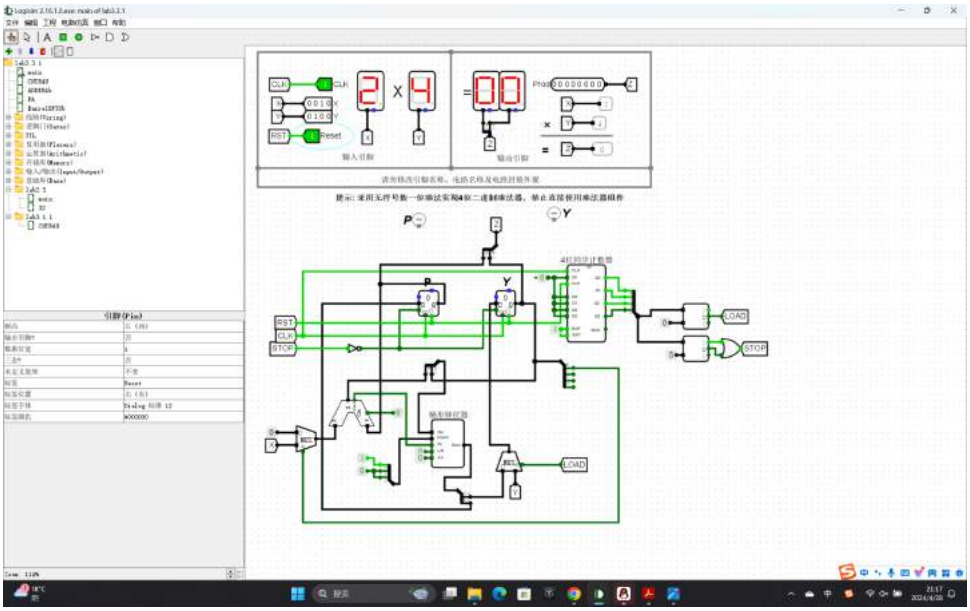

图 31: 输出置位

本实验不需要真值表。功能表在上面。

1.3.5 错误现象及分析

在实验过程中,没有遇到任何错误。

1.4 寄存器堆实验

1.4.1 实验内容

构建含有 32 个 32 位寄存器的寄存器堆 Regfile 的读写电路，包含两个读数据端口和一个写数据端口，并封装成子电路。

1.4.2 实验整体方案设计

寄存器堆的读操作属于组合逻辑操作，无须时钟控制，即当寄存器地址信号 RA 或 RB 到达后，经过一个“读取时间”的延迟，读出的数据输出到端口 busA 或 busB 上。寄存器堆的写操作则属于时序逻辑操作，需要时钟信号的控制，即在写使能信号（WE）有效的情况下，有效时钟触发边沿到来后将写入数据端口 busW 上的信息写入 RW 所指定的寄存器中。时序上要求在时钟边沿信号达到前，写使能信号 WE，写入寄存器地址 RW 和写入数据 busw。本实验不需要底层模块设计图。

RW	控制存入哪个寄存器
WE	读入使能端
CLOCK	时钟信号触发
DIN	输入值
RA，RB	控制读取哪个寄存器存的值

表 5: 功能表

1.4.3 实验原理图和电路图

原理图：

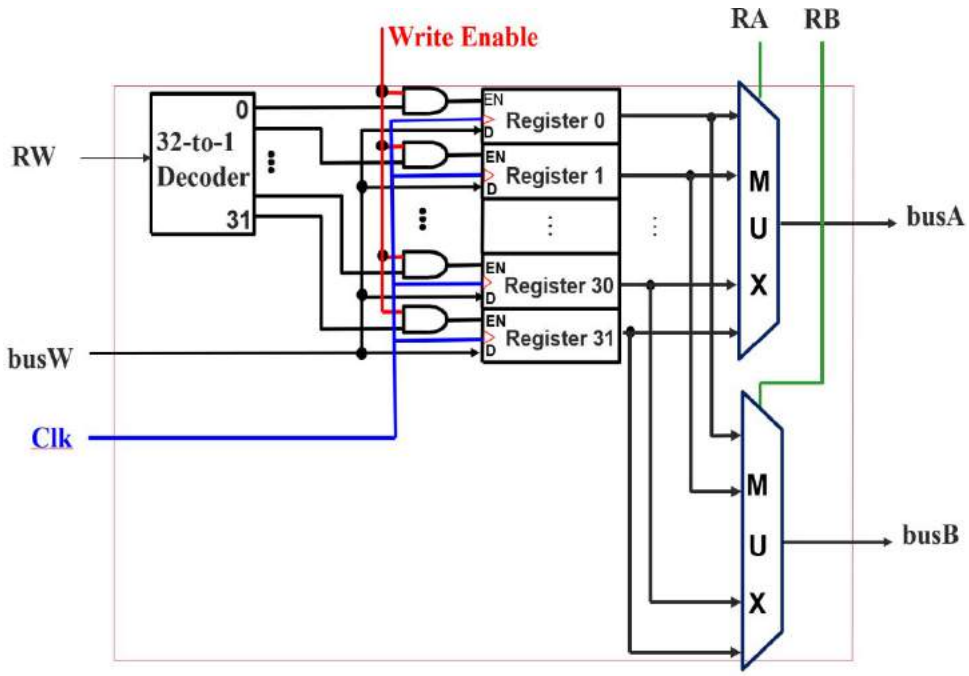

图 32: 原理图

电路图实现：

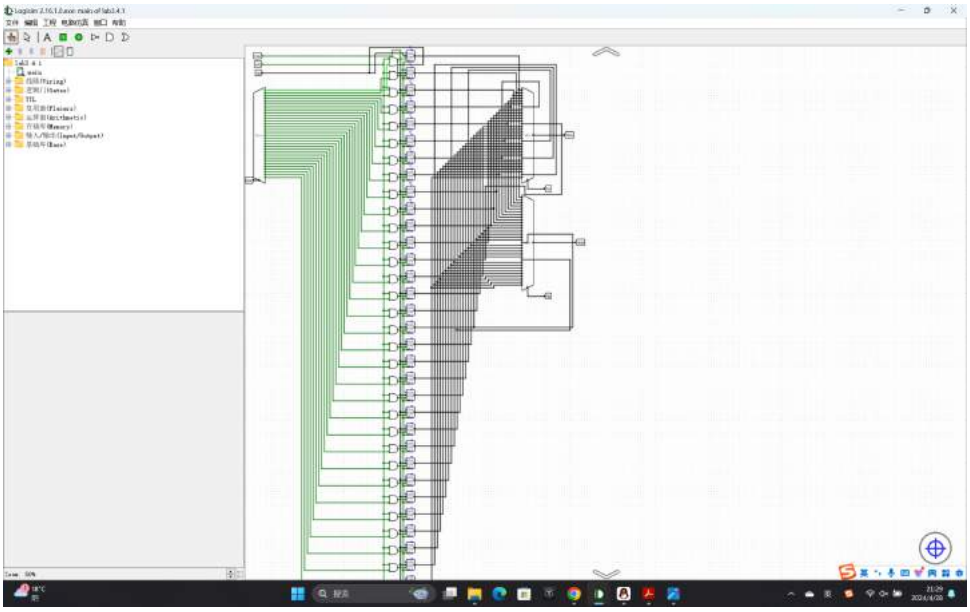

图 33: 电路图

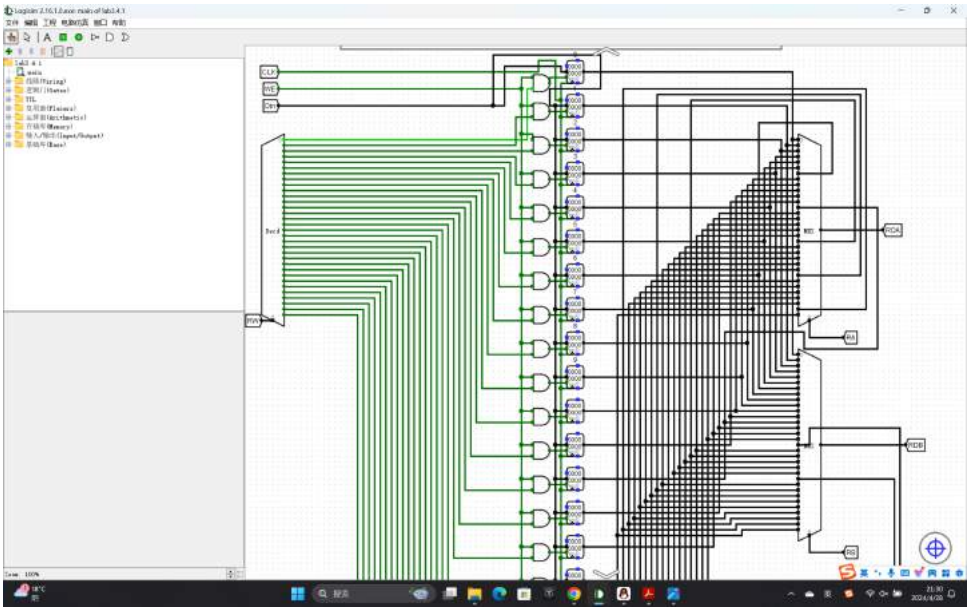

图 34: 电路图

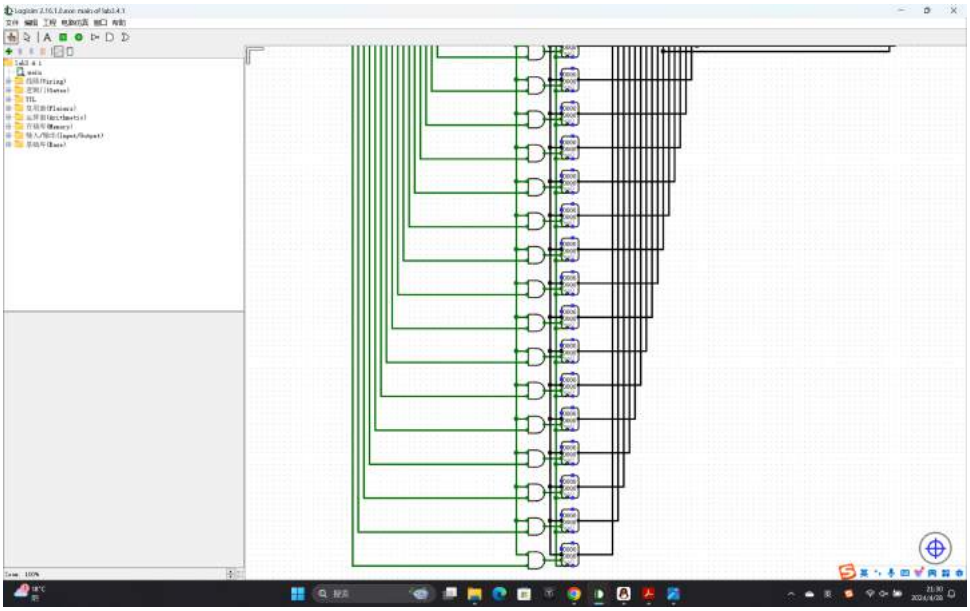

图 35: 电路图

1.4.4 实验数据仿真测试图

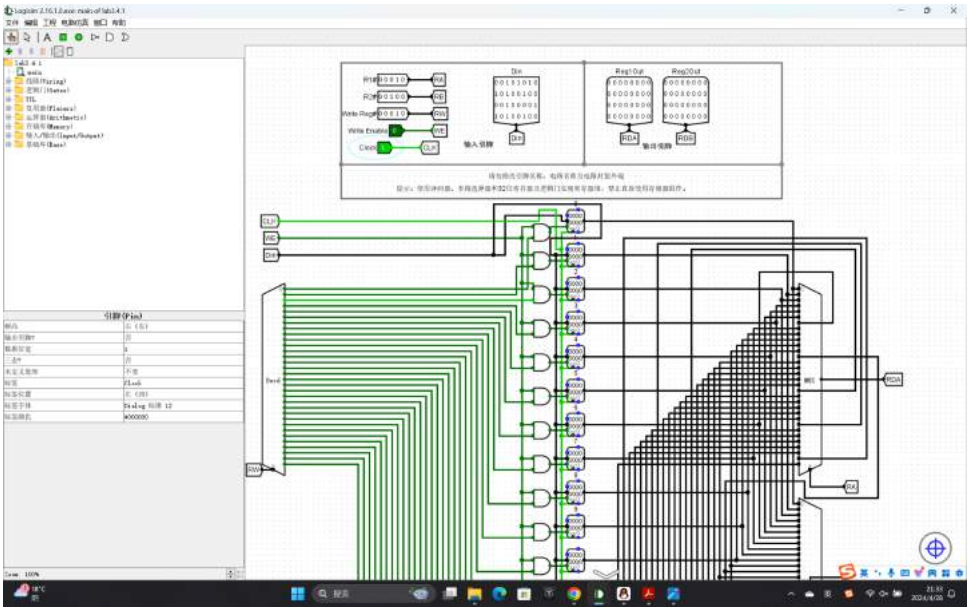

图 36: 无使能端开启

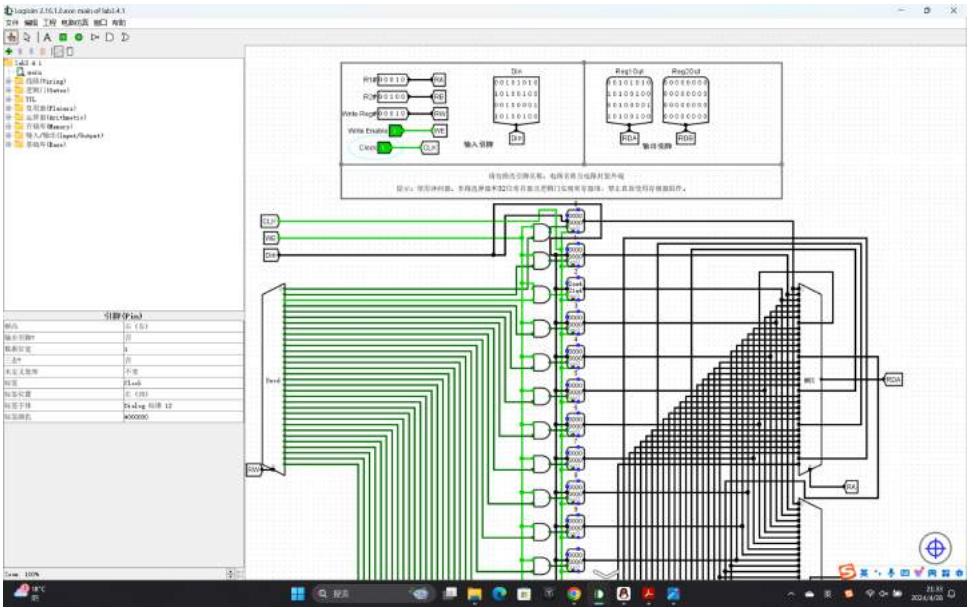

图 37: 开启使能端,Rb

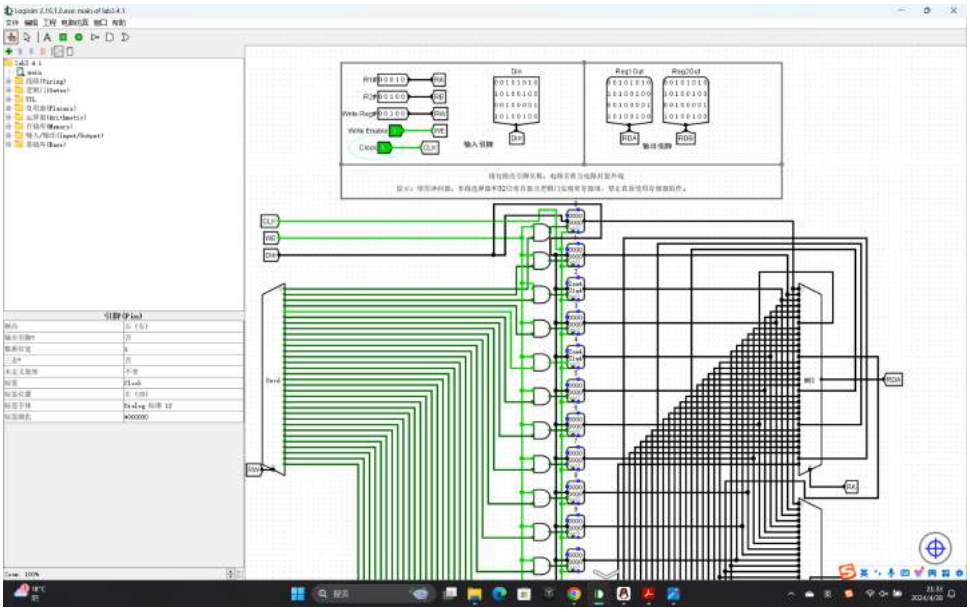

图 38: Ra

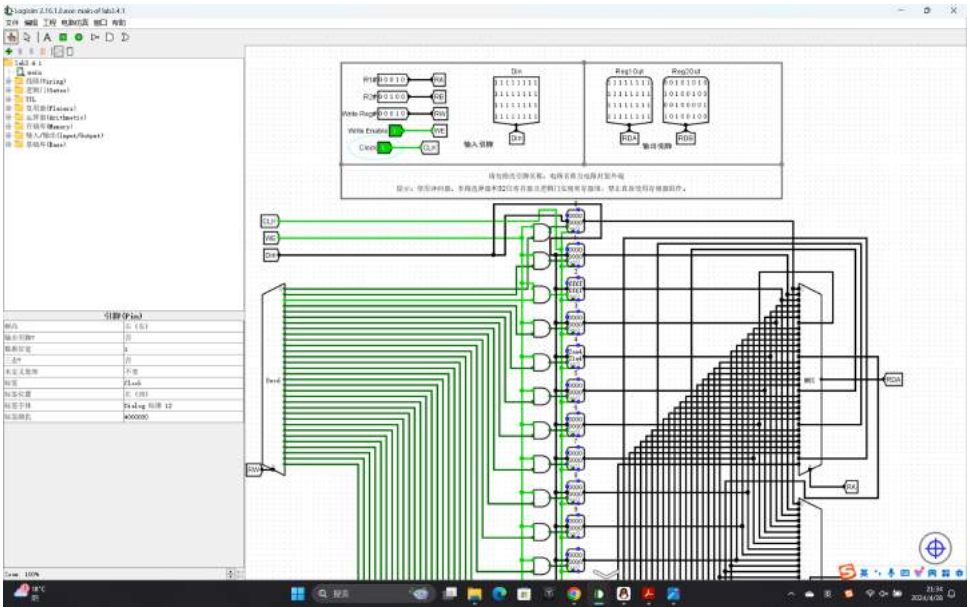

图 39: 更改 din 和一个寄存器的值

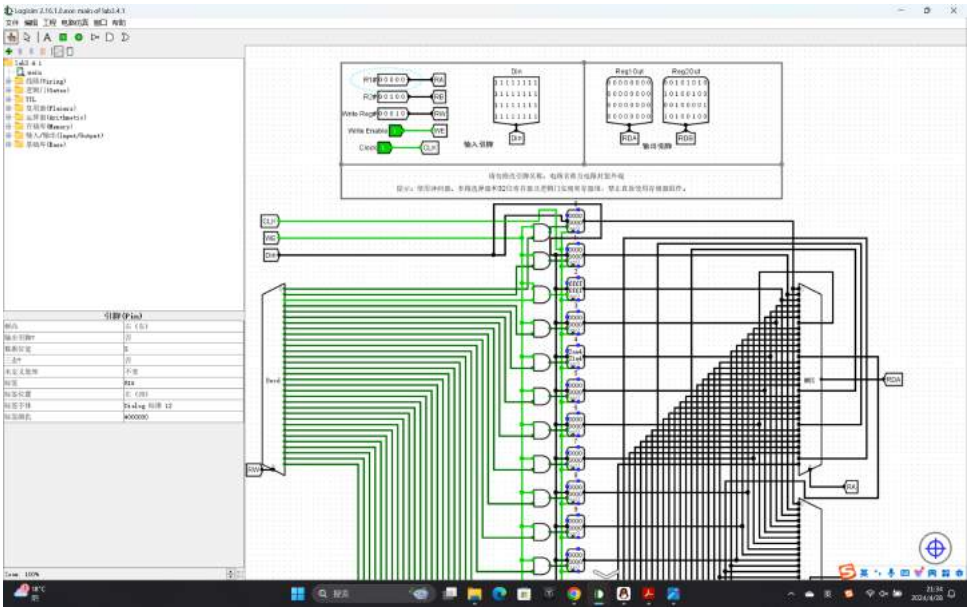

图 40: 释放 Ra 寄存器

1.4.5 错误现象及分析

在实验过程中，没有遇到任何错误。

1.5 数字时钟实验

1.5.1 实验内容

在 6 个 7 段数码管上显示数字时钟时分秒，当计时到 23:59:59 后进入 00:00:00，时分秒之间用小数点分隔；到整点时轮流点亮三色 LED 灯组件；使用 8421BCD 码设置初始时间，在载入时如果初始时间数值超出实际范围，则报错，且不能被载入。

利用同步计数器电路实现 0 9 十进制计数器、0 59 六十进制计数器和 0 23 二十四进制计数器。当时分秒的低位计数到 9 时，高位计数加 1，低位清零。当秒数计数到 59 后秒数清零，分钟计数加 1；当分钟计数达到 59 且秒数也到达 59 秒时，分钟清零，小时计数加 1 当小时计数到 23，且分钟和秒数都是 59 时，小时数清零。当载入信号 LD=1 有效时，检查设置时间数值：小时高低位 HH1 和 HH0、分钟高低位 MM1 和 MM0、秒数高低位 SS1 和 SS0，如果在有效数据范围之内，则载入时钟初值，否则 INErr 赋值 1。当计时到 59 分 59 秒进入整点时，三色 RGBLED 灯按照格雷码（蓝色为高位、红色为低位）的顺序轮流点亮三色灯（红色、黄色、绿色、青色、白色、品红、蓝色、黑色、红色、黄色），并持续 10 个时钟周期。需要独立设计电路进入计数状态，设置开始计数标志位和结束计数标志位，并实现格雷码的转换。

1.5.2 实验整体方案设计

电路主要分为三个部分。先从最简单的判断越界说起。用 8 个比较器，分别判断各个位数的合法性再用或门和 ld 与门即可得到 LNERR 信号。

第二个部分是 LED 灯，同样用计数器＋比较器的组合实现，异或门和 MUX 实现格雷码（只用三位，8 位循环），这样很怪，因为数据是 4 位，但是测试集如此。最后部分是主电路，要做到判断是否到 60/24/10 进位（连到上一个位的 CLR 清零），可以用门电路，也可以用比较器，这里用的是门电路。LD 和 LNERR 共同控制计数器的 LD，Q 不仅作为输出位还是下一位的 CLR，时高，时低，分高，分低，秒高，秒低的计数器依次连接。ENP，ENT 的连接也是通过逻辑门实现的，通过逻辑门和 ENP，ENT 可以实现时/分/秒之间的联系。

HH0 等 6 个	时/分/秒高/低位
计时器	控制单位时钟计数
CLK	时钟信号
LD，LNERR	控制溢出和检测数据错误
LED	三个 LED 灯

表 6: 功能表

1.5.3 实验原理图和电路图

原理图：

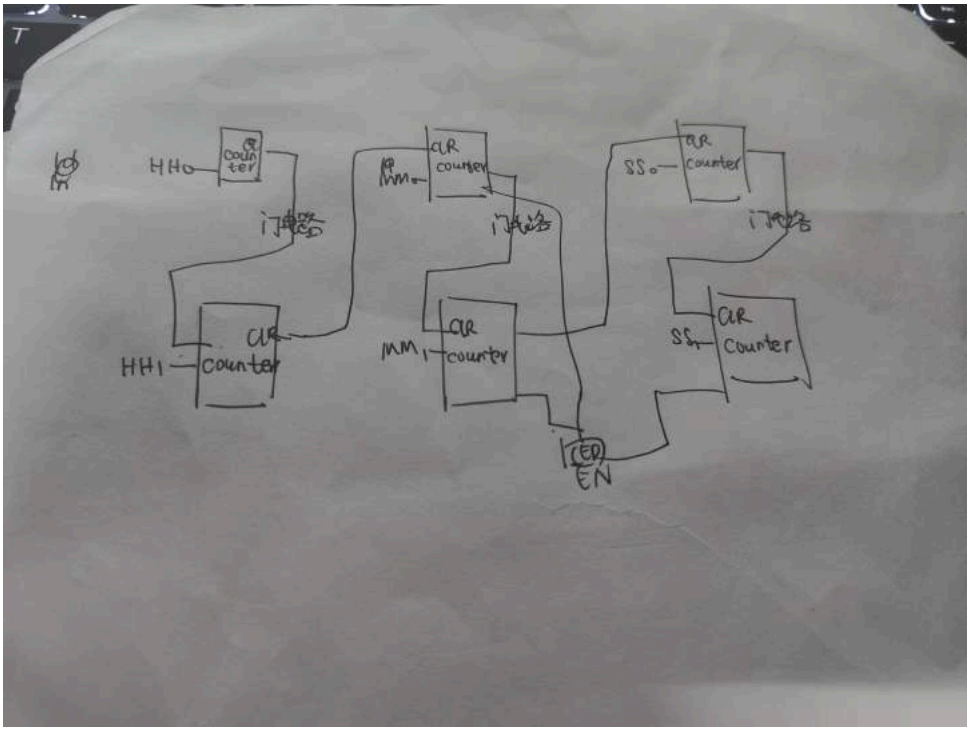

图 41: 原理图

电路图实现：

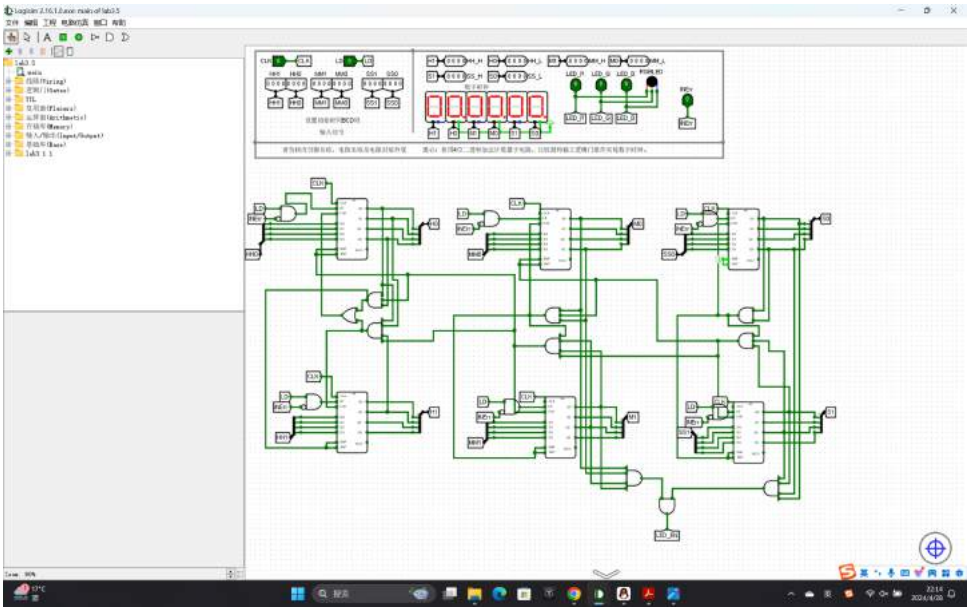

图 42: 主电路电路图

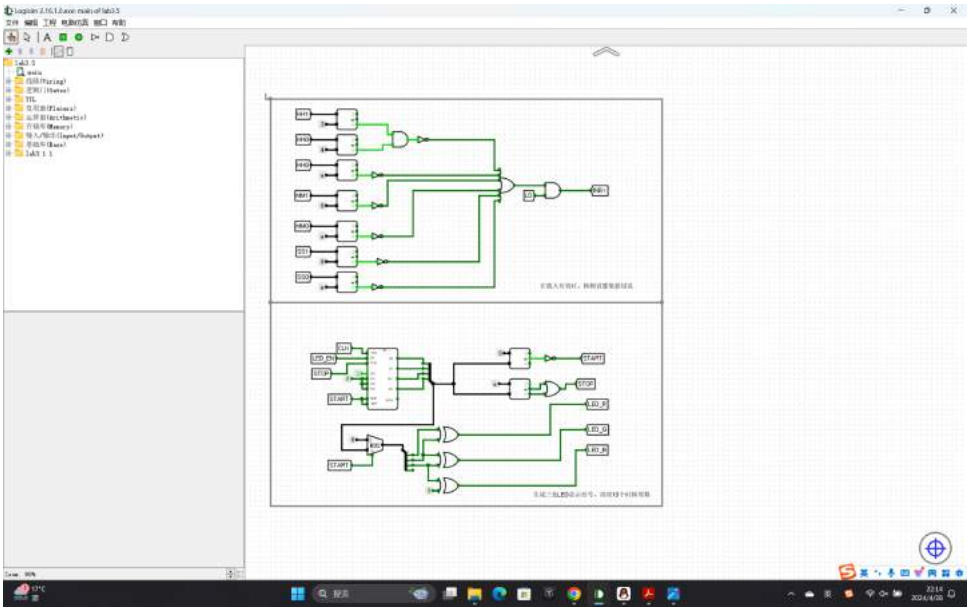

图 43: 电路图 2

1.5.4 实验数据仿真测试图

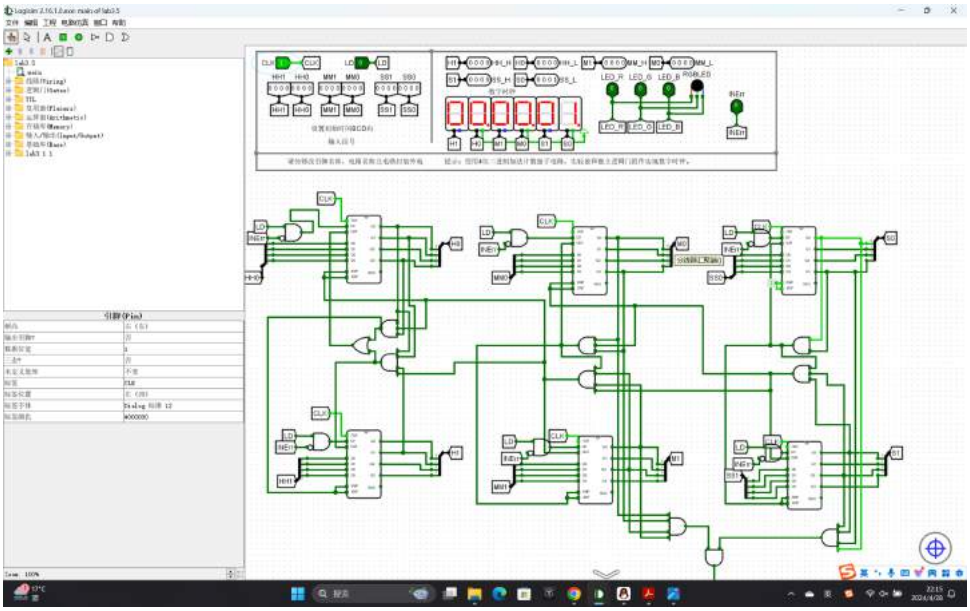

图 44: 初始

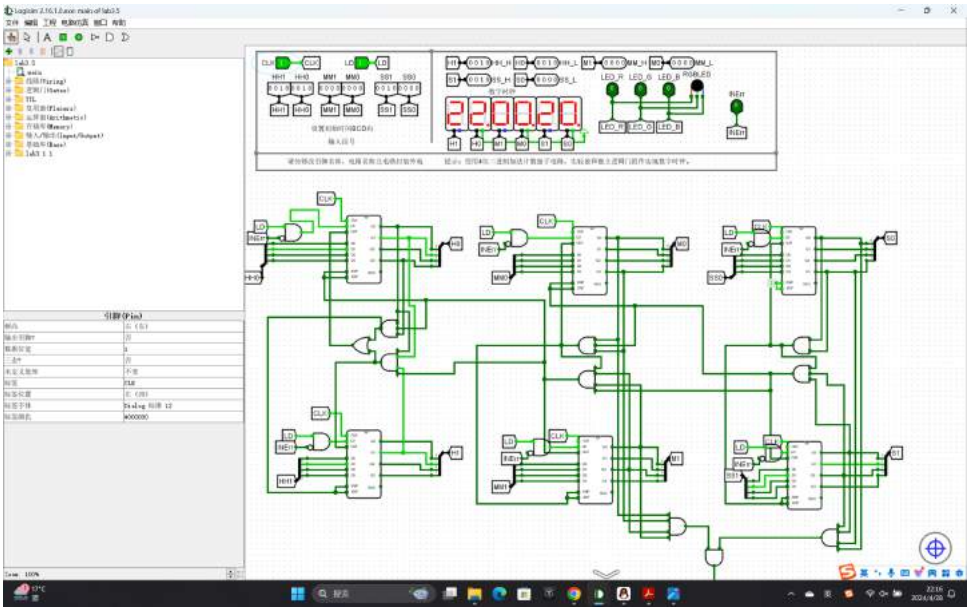

图 45: 加载

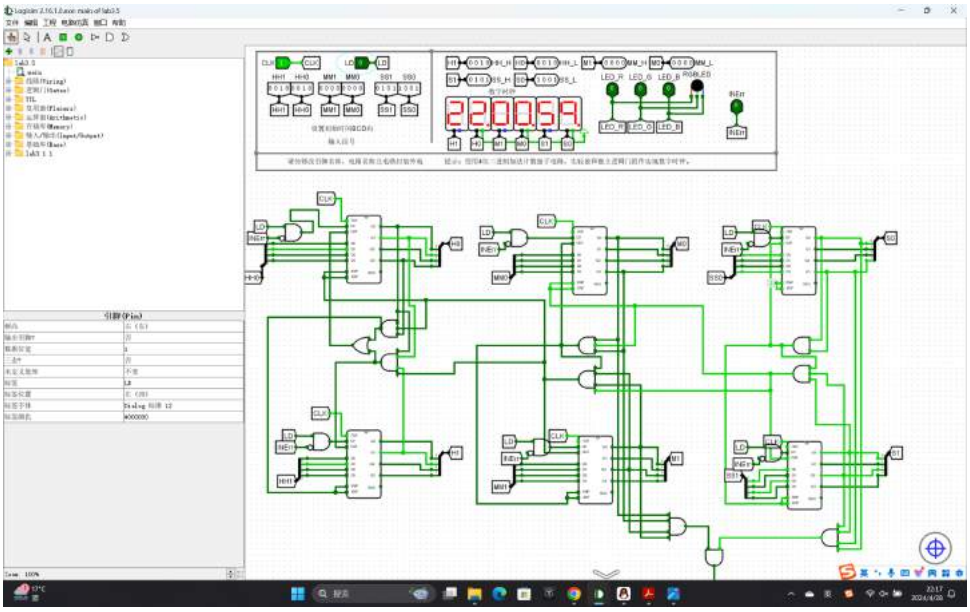

图 46: 进位 1

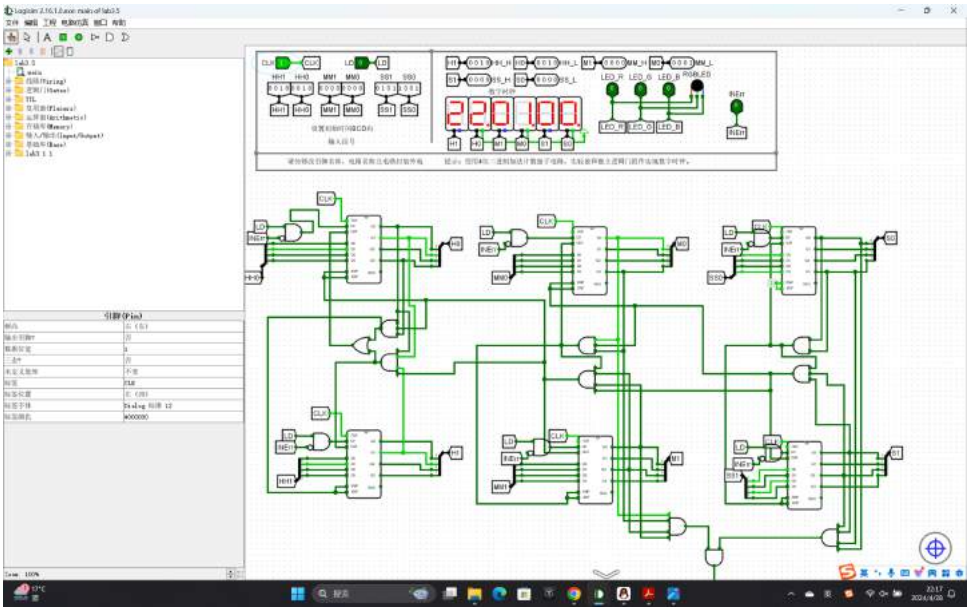

图 47: 进位 2

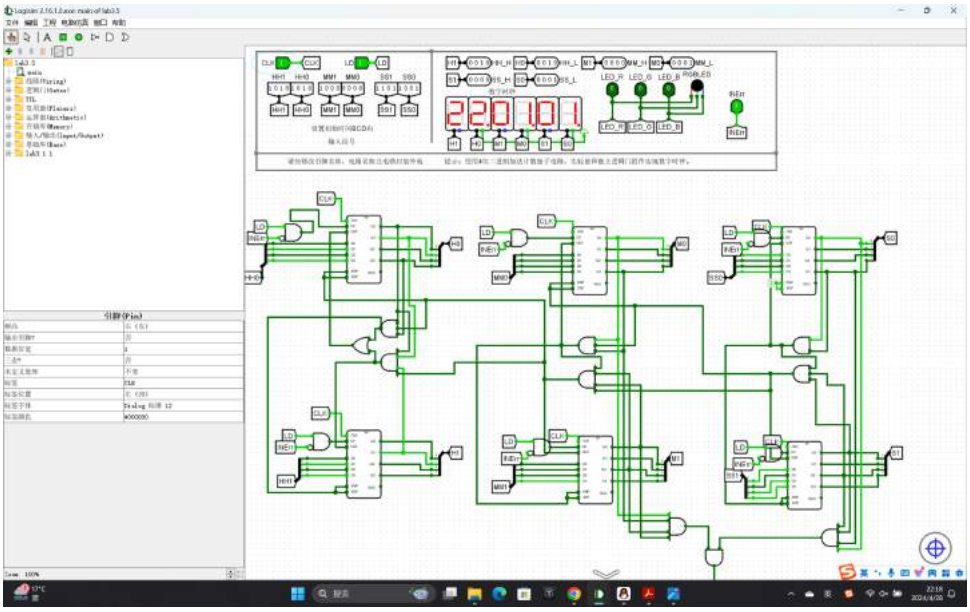

图 48: 溢出

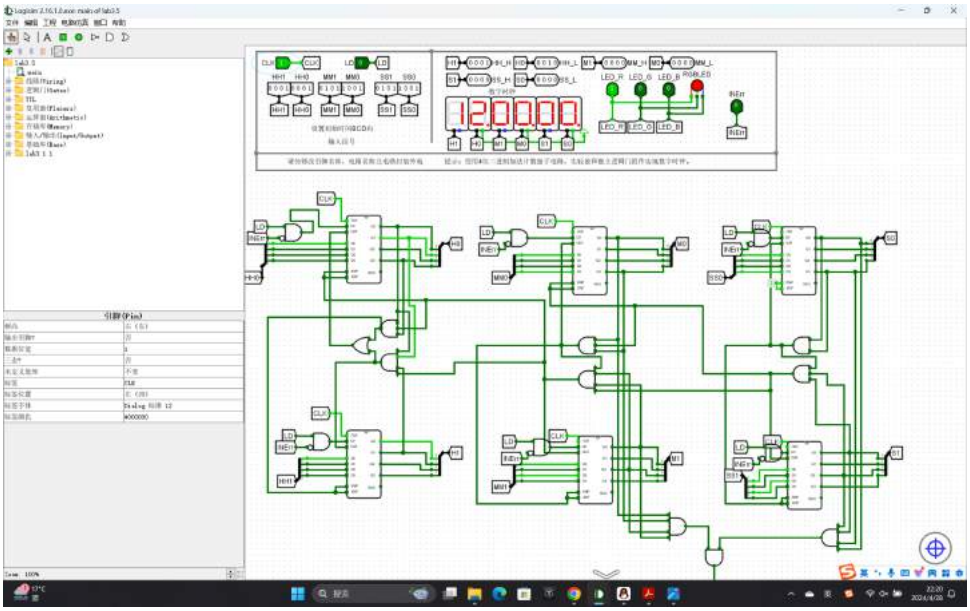

图 49: LED 灯 1

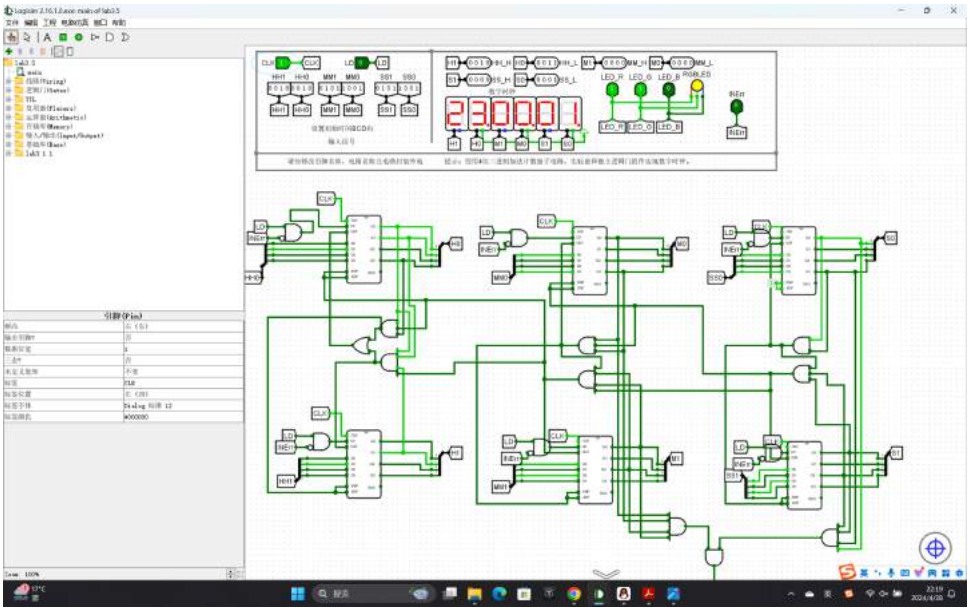

图 50: LED 灯 2

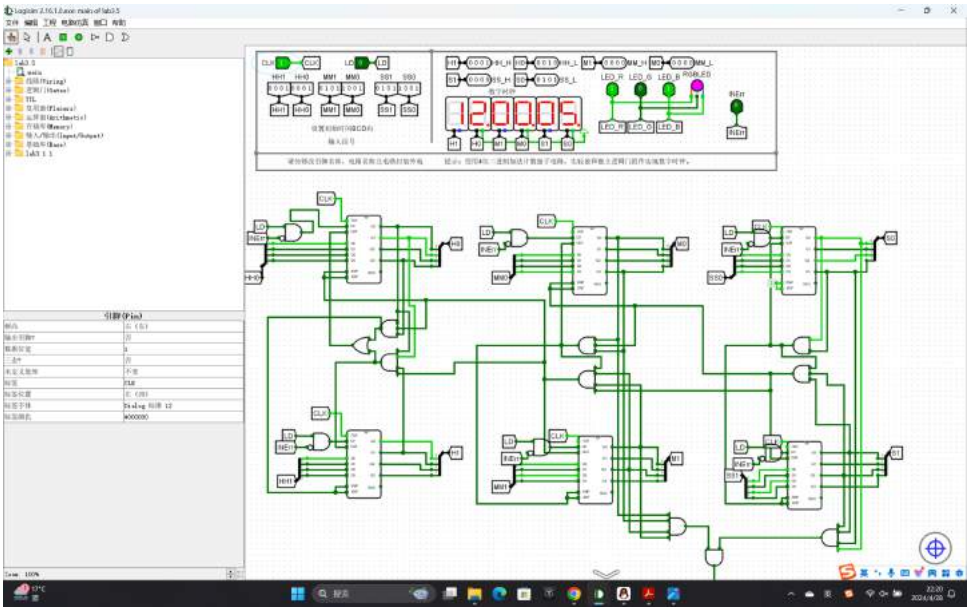

图 51: LED 灯 3

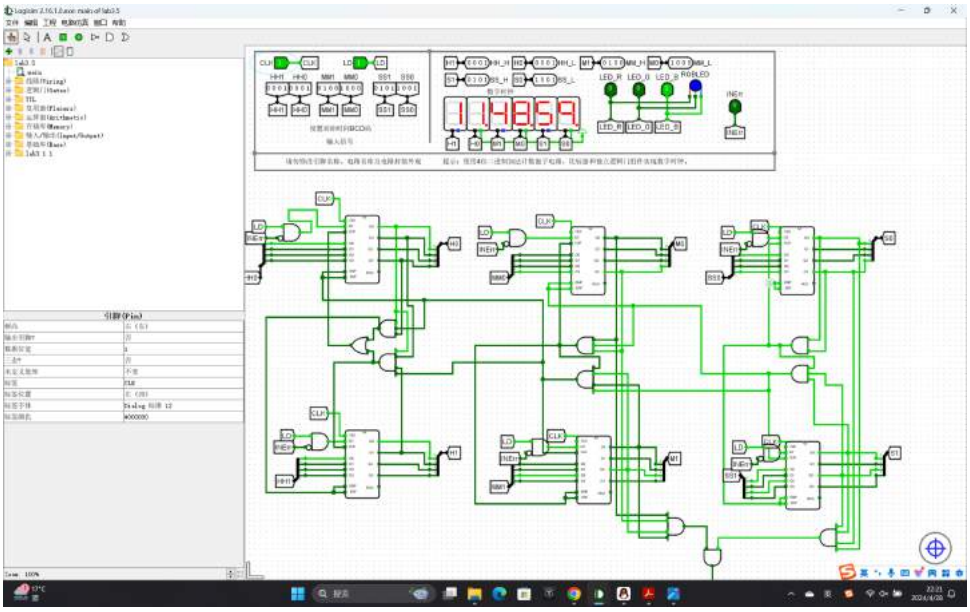

图 52: LED 灯异步不回归

不需要真值表。

1.5.5 错误现象及分析

在完成实验的过程中，遇到了如下几个关于面向测试集的错误：1.LED 灯的格雷码要求不够清晰，没有说明白留几位，看了测试集才知道 8 位一循环 2。主电路的测试集的 LED 灯没亮完就重新 LOAD 不会重置，之前的电路是重置的，感觉这里 PDF 也没说明白剩下的没有遇到问题，除了一开始尝试用 D 触发器和寄存器来实现发现时序逻辑会慢一个 CLOCK，后来改成逻辑门了。其实用比较器会更直观一点。

2 思考题

2.1 思考题 1

1. 级联 2 个 4 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。
如图，通过后四位异或成为新的最后一位（左移一位），通过设置合适的 DIN（这里设置的是 10111000），能生成一个很长的伪随机数，其实理论上，应该存更大的位置的，更直观，比如 32 位，64 位，很容易实现，就不改了。

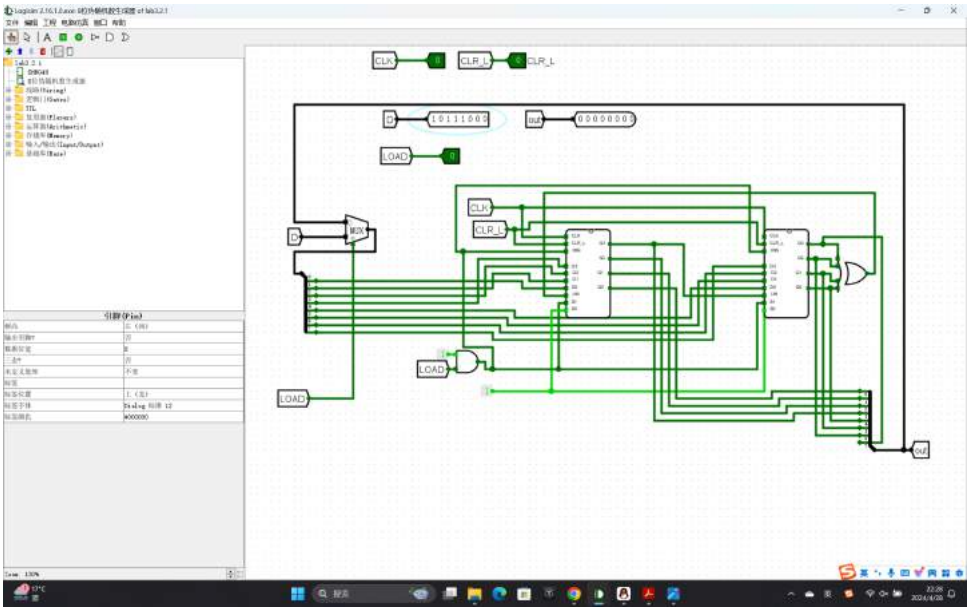

图 53: 思考题 1.1

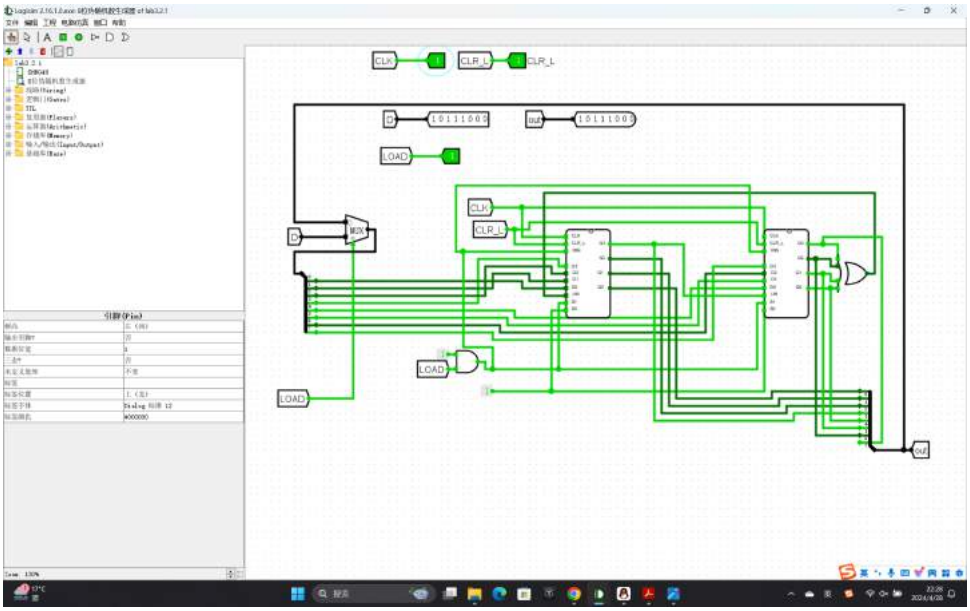

图 54: 思考题 1.2

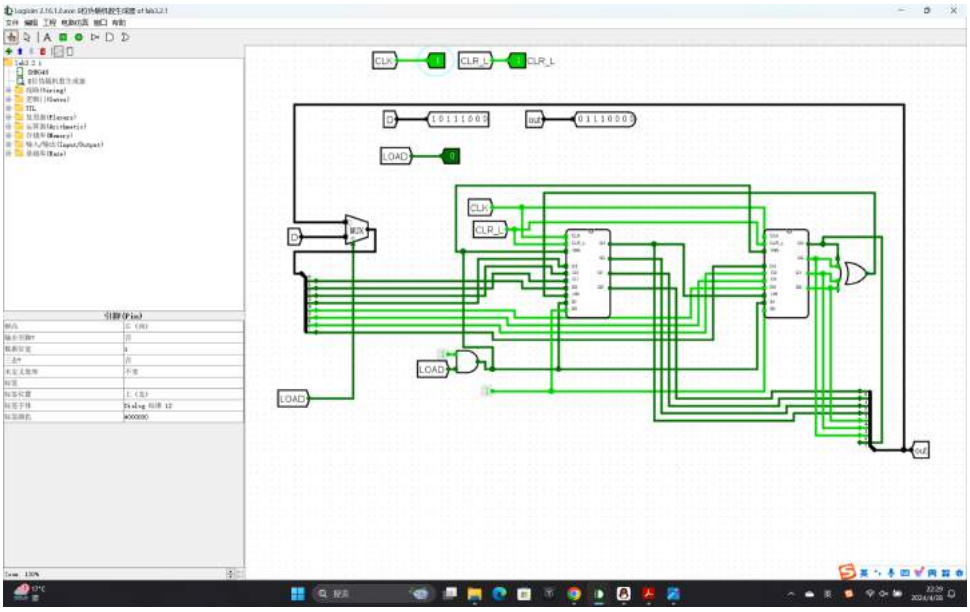

图 55: 思考题 1.3

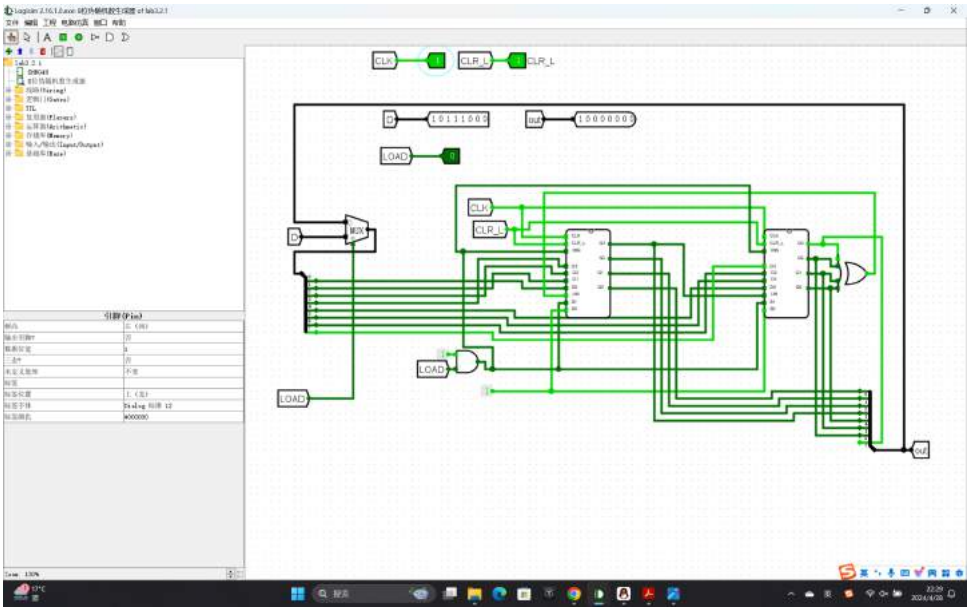

图 56: 思考题 1.4

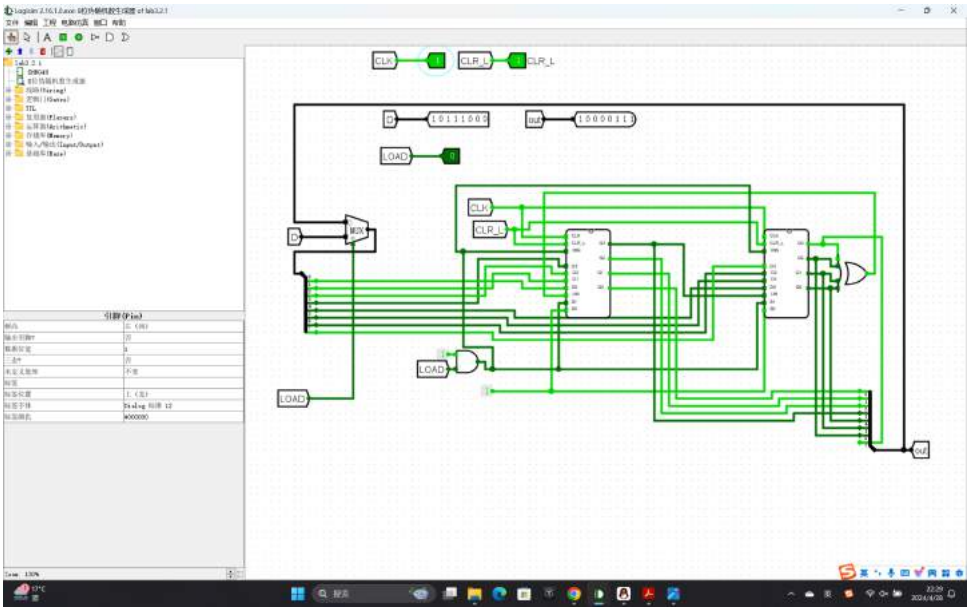

图 57: 思考题 1.5

2.2 思考题 2

2. 查找资料学习如何利用加法器实现 8 位无符号数的快速乘法器。

找到了一个实现五位无符号数的快速乘法器的图片，其实 8 位快速乘法器的也是一模一样的，只是位数的相加次数和器件数量需要增加。

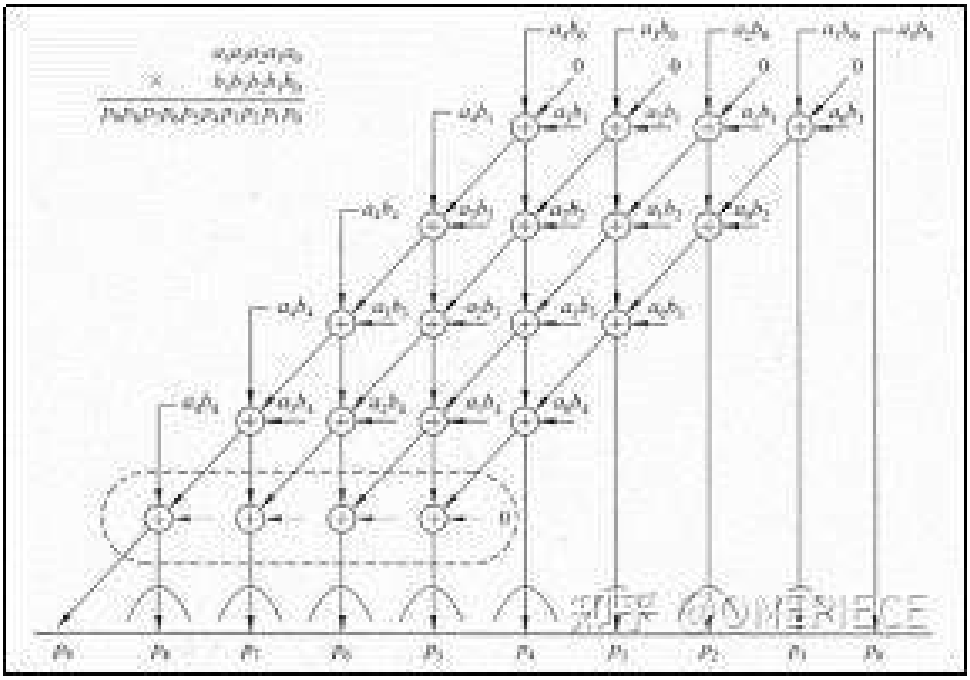

图 58: 思考题 2.1

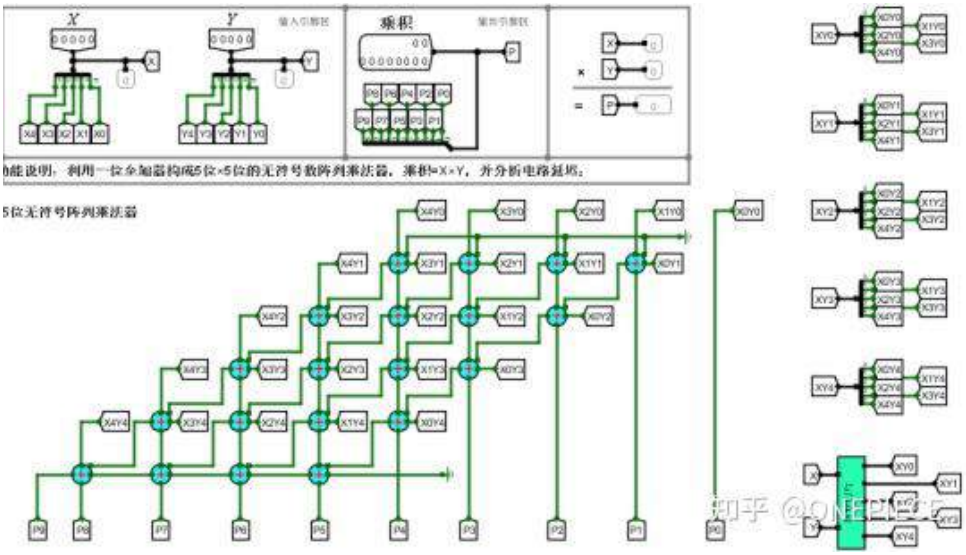

图 59: 思考题 2.2

2.3 思考题 3

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。

说实话,我没看懂这个题目 qwq 什么叫将输入信号连接使能端和时钟端啊,不是 32 位的吗,没看懂要求不过使能信号和写入地址信号的先后时序关系变化当然会影响写入结果,首先可以从新老原理图的结果不同可以看出。并且先写入才能加载,这两步是有时序的时间差距的,因此会影响写入结果。

2.4 思考题 4

4. 在数字时钟设计中如何添加闹钟的功能。如图,单独增加了一个闹钟模块。可以设定闹钟,让蜂鸣器响。(就不贴实际效果图了,请助教自己试试吧 qwq 真的很好玩的,声音没法贴上来)操作:先 LD 打开,数字时钟 CLK 一次设定时间,再将设定闹钟和闹钟开关都开成 1,CLK 一次;然后将设定闹钟换成 1,就可以啦,想关闭闹钟点一下就行。

原理是两个值对比,两个寄存器,第一个寄存器存设定值,和当前时间比较,相等且开启闹钟就响,按关闭瞬间切 0 使其置 0 停止闹钟。

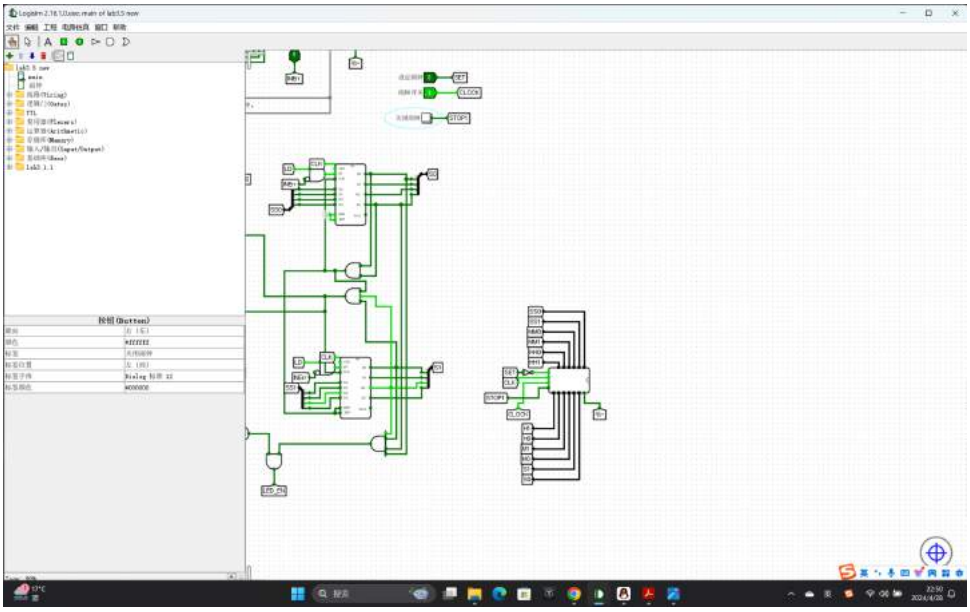

图 60: 思考题 4.1

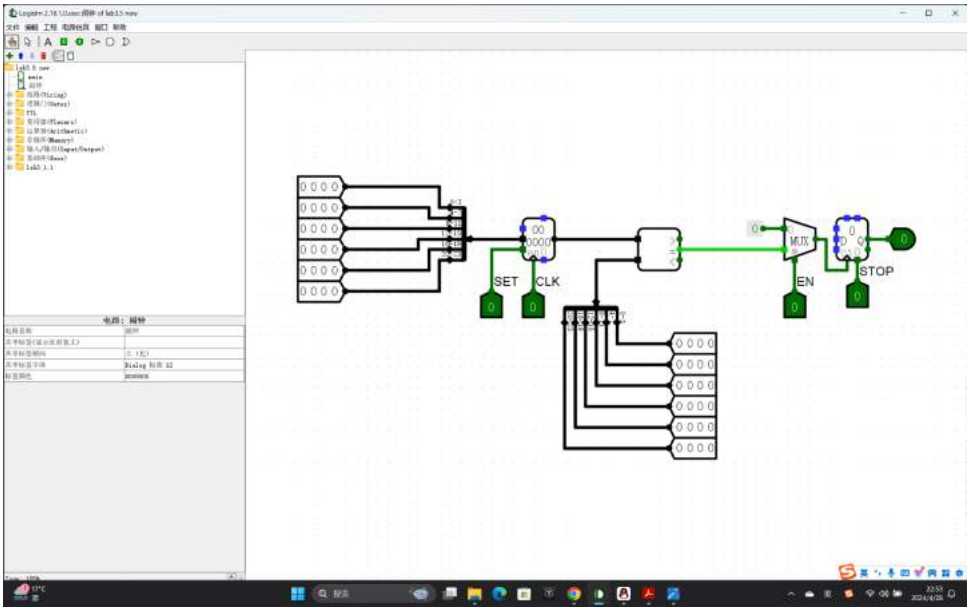

图 61: 思考题 4.2