《数字逻辑与计算机组成》实验报告四.ALU 设计

1 实验

1.1 四位先行进位加法器 CLA

1.1.1 实验内容

实现四位先行进位加法器 CLA。

1.1.2 实验整体方案设计

根据三个步骤，看懂就能画出来。原理和教材讲的一样，不赘述了。本实验不需要顶层模块设计图。CLA 只是这两个部件的组合。

Pi,Gi	生成进位信号 Pg,Gg
C0	最初的 Cin
Ci	4 位进位值，C4 用于 Cout，其他用于全加器 Cin
Gg，Pg	组间进位生成函数，组间进位传递函数

表 1: 4 位先行进位部件 CLU

P,G	生成 Pi,Gi
Cin	前进位（即先行进位部件的 Ci）
Ai,Bi	原先的初始数据
F	最终全加的结果

表 2: 支持 Pg,Gg 的全加器

1.1.3 实验原理图和电路图

原理图实现：

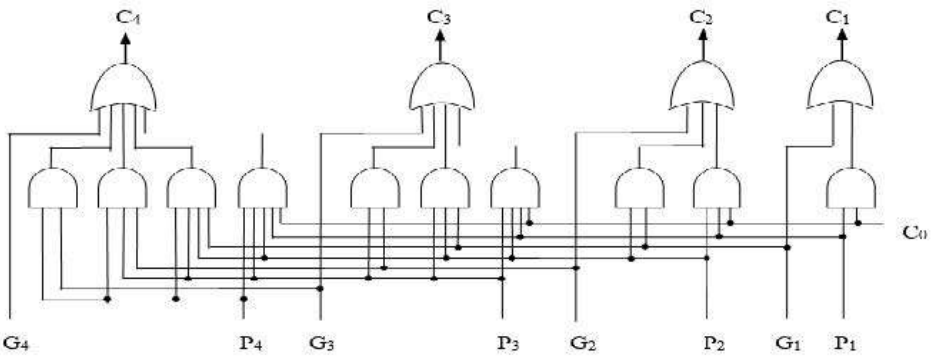

图 1: CLU 原理图

电路图实现：

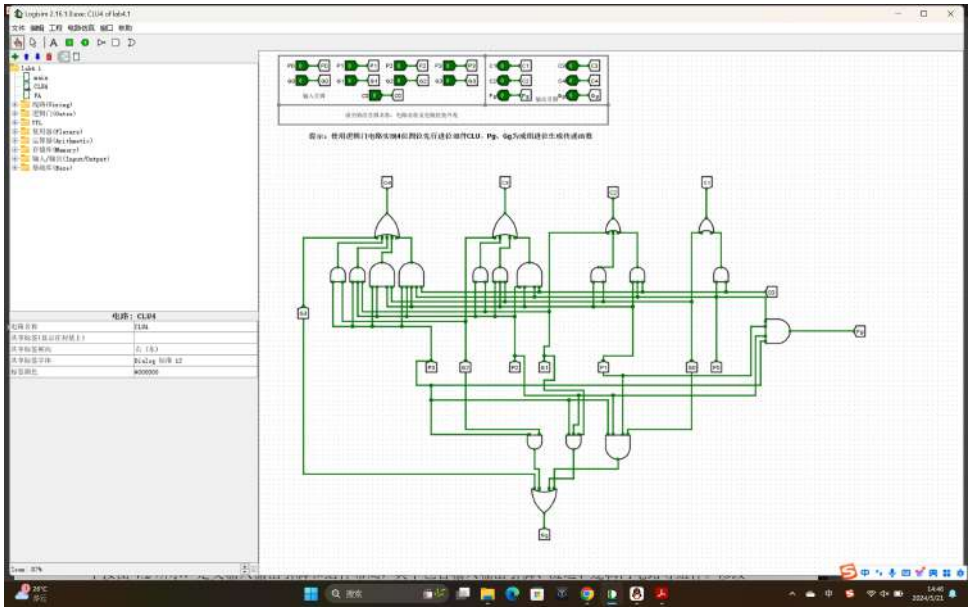

图 2: CLU 电路图

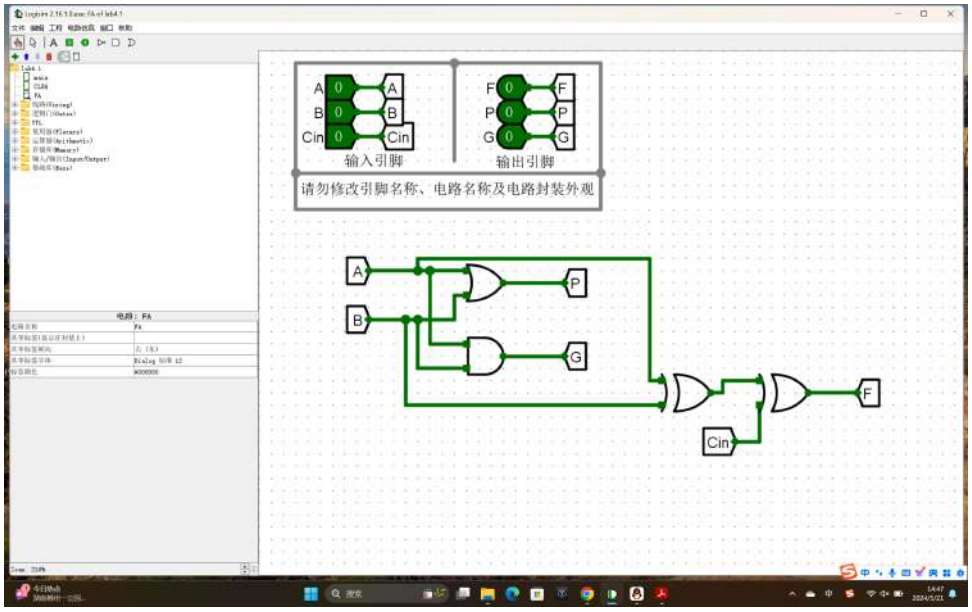

图 3: 全加器电路图

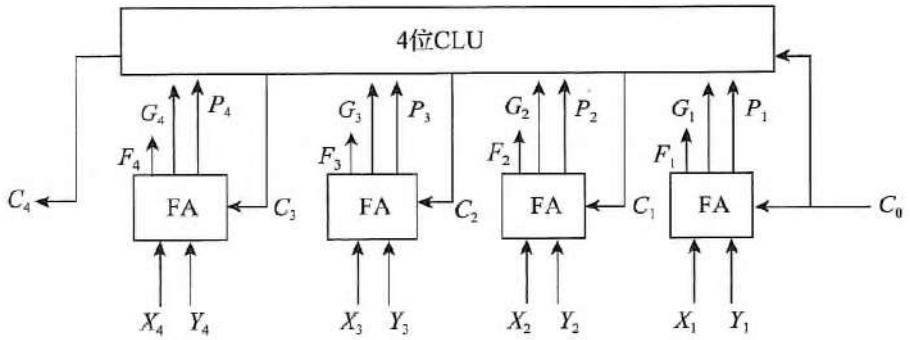

图 4: 先行进位加法器原理图

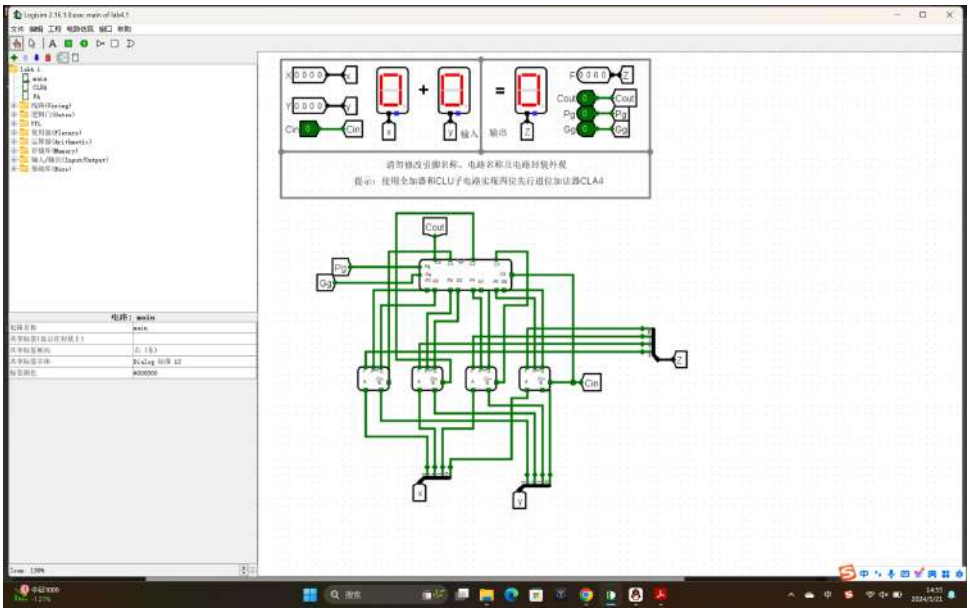

图 5: 先行电路图

1.1.4 实验数据仿真测试图

展示了正常加法,溢出进位,Pg,Gg。Pg,Gg 是给下面的实验用的。

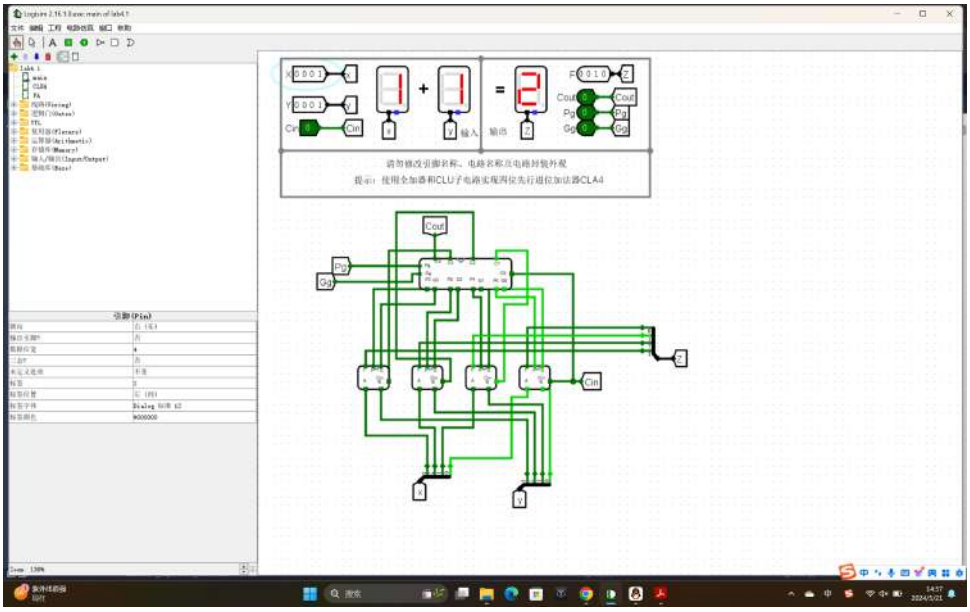

图 6: 模拟 1

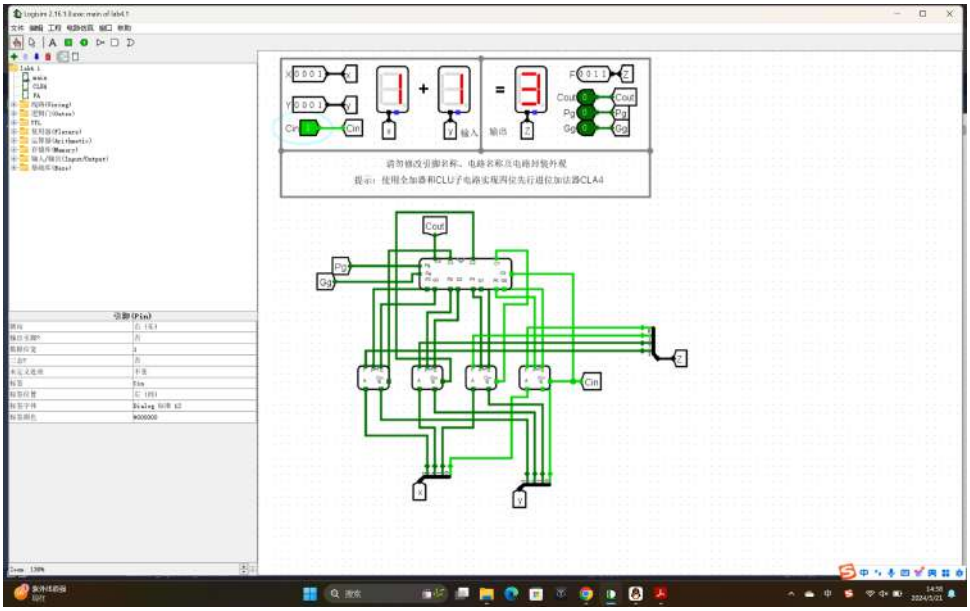

图 7: 模拟 2

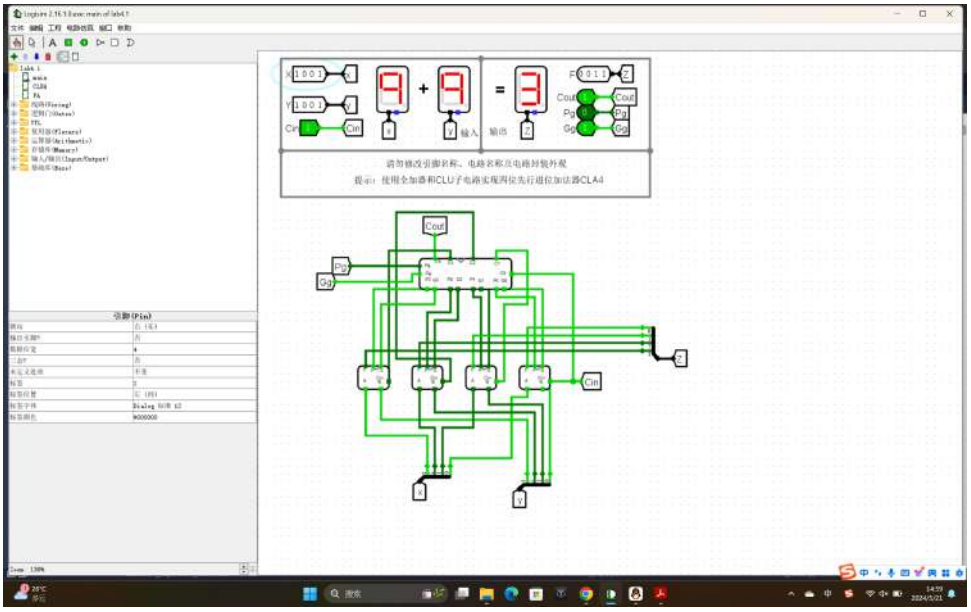

图 8: 模拟 3

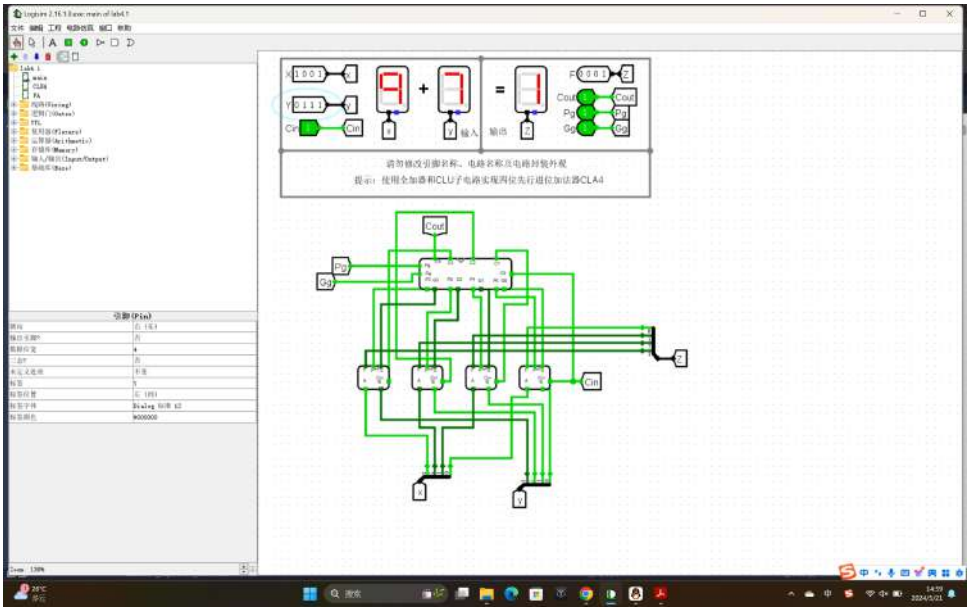

图 9: 模拟 4

不需要功能表。

1.1.5 错误现象及分析

在实验过程中,没有遇到任何错误。

1.2 16 位两级先行进位加法器实验

1.2.1 实验内容

对于一个 16 位加法器,可以分成 4 组,每组用一个 4 位先行进位加法器 CLA 实现。

1.2.2 实验整体方案设计

根据原理图可以画出。和第一个实验类似,只是全加器变成了 4 位先行进位。本实验不需要顶层模块设计图。

X,Y	两个 32 位输入
C0	先行进位
S	结果

1.2.3 实

原理

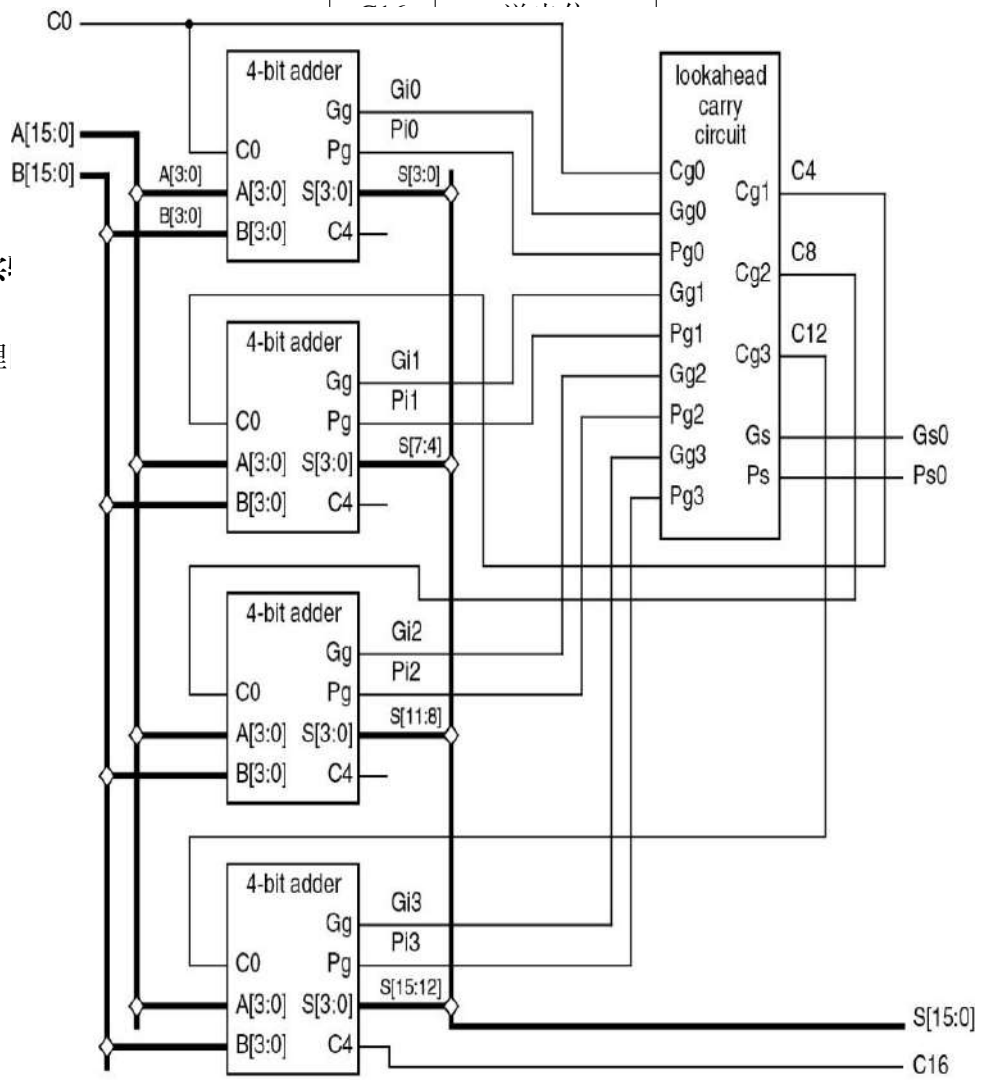

图 10: 原理图

电路图实现:

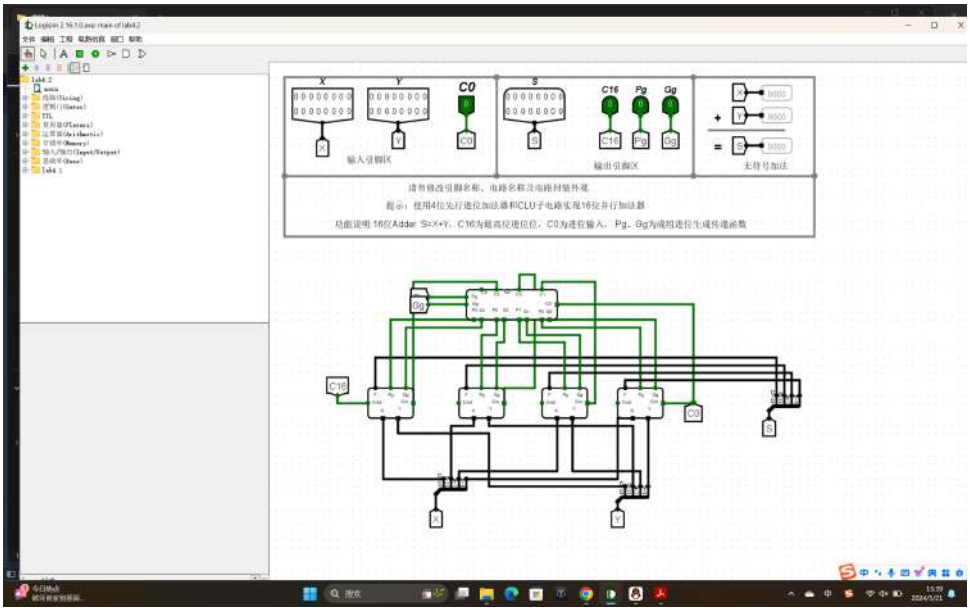

图 11: 电路图

1.2.4 实验数据仿真测试图

分别展示了普通加法,普通进位,最后一位进位,Pg 和 Gg。

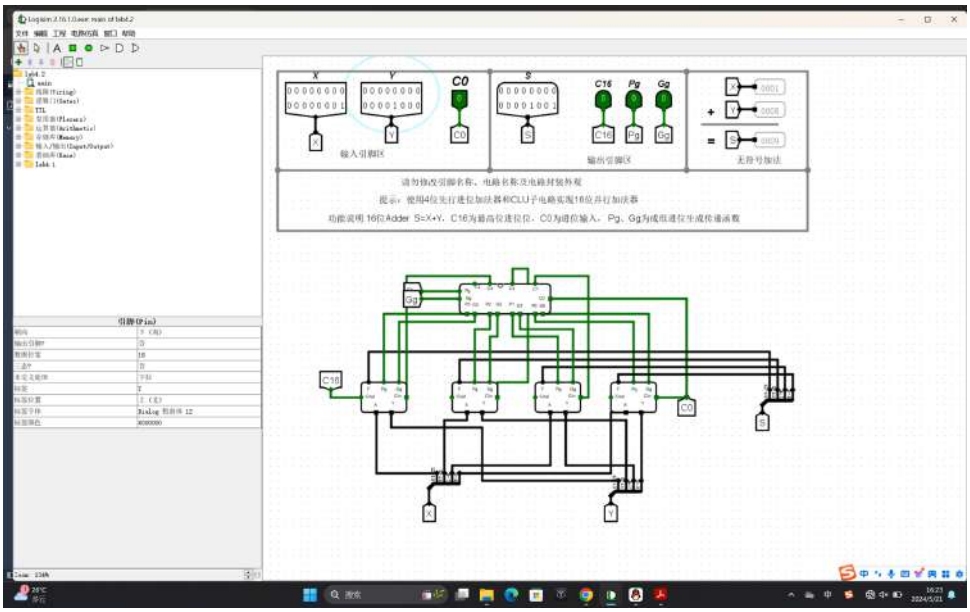

图 12: 普通加法

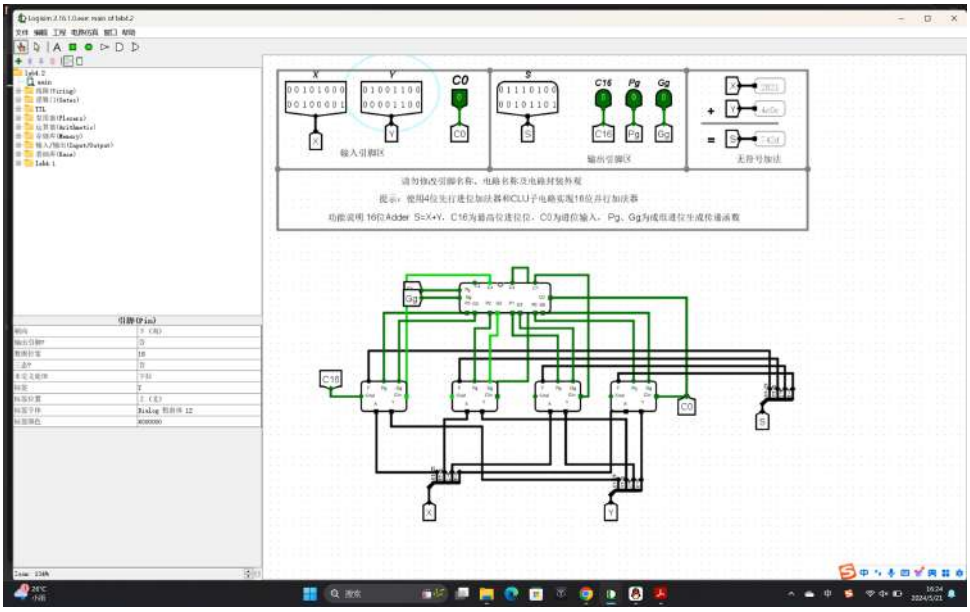

图 13: 普通进位

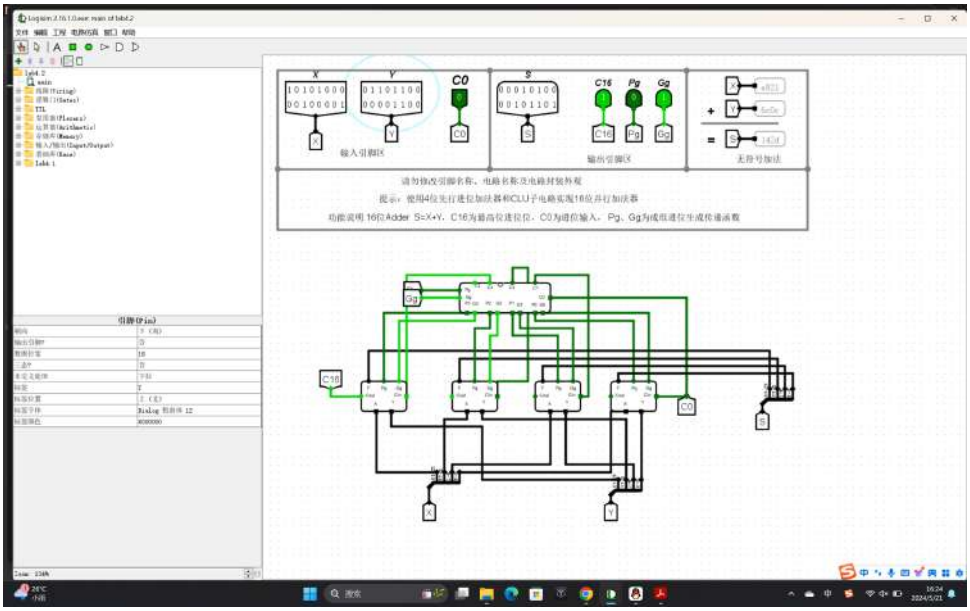

图 14: C16

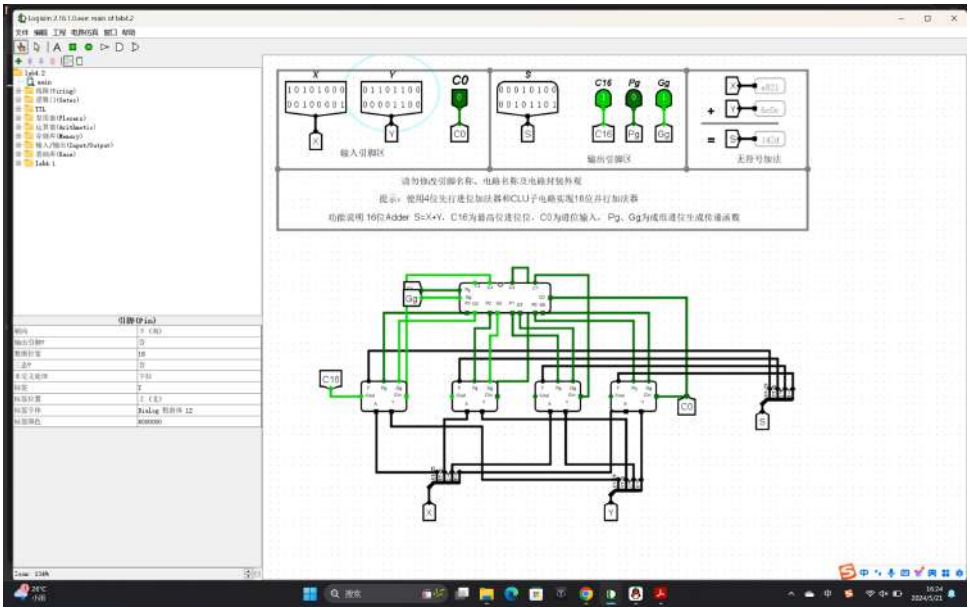

图 15: Pg,Gg

不需要功能表。

1.2.5 错误现象及分析

在实验过程中,没有遇到任何错误。

1.3 32 位快速加法器

1.3.1 实验内容

通过将两个 16 位两级先行进位加法器串行级联构建一个 32 位加法器,并根据给出的标志位生成电路原理图,在 32 位加法器中生成 CF、SF、OF、ZF 标志位。

1.3.2 实验整体方案设计

OF=Cn 异或 Cn-1,CF=Cout 异或 Cin,SF=Fn-1,ZF=1 当且仅当 F=0,OF 的式子是根据定义进行卡诺图化简得来的,溢出为 1,不溢出为 0。快速加法器很简单,串联即可。

1.3.3 实验原理图和电路图

不需要原理图,之前思考题已经有过思路,连上即可。电路图实现:

16 位先行进位加法器	用于串联形成 32 位
OF	溢出标志
CF	进位标志
SF	符号标志
ZF	零标志
Cin	输入
Cout	溢出位
result	结果

表 4: 功能表

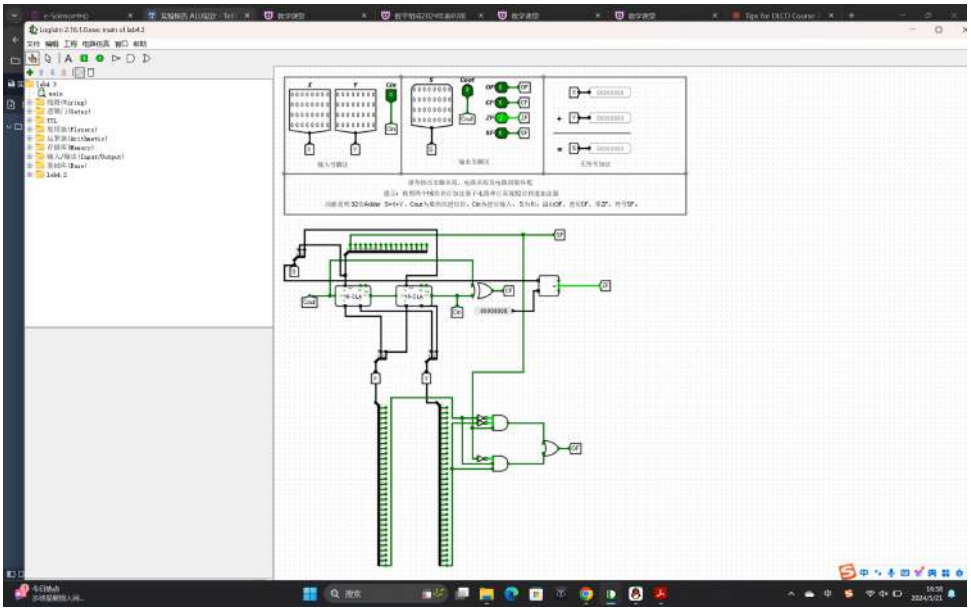

图 16: 电路图

1.3.4 实验数据仿真测试图

测试了正常加法,四个标志位和 cout。

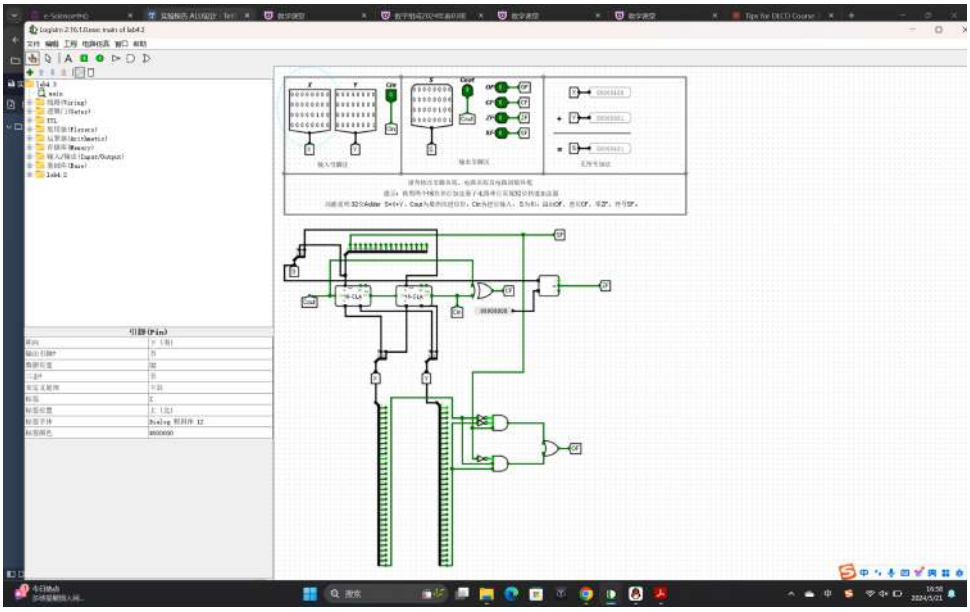

图 17: 正常加法

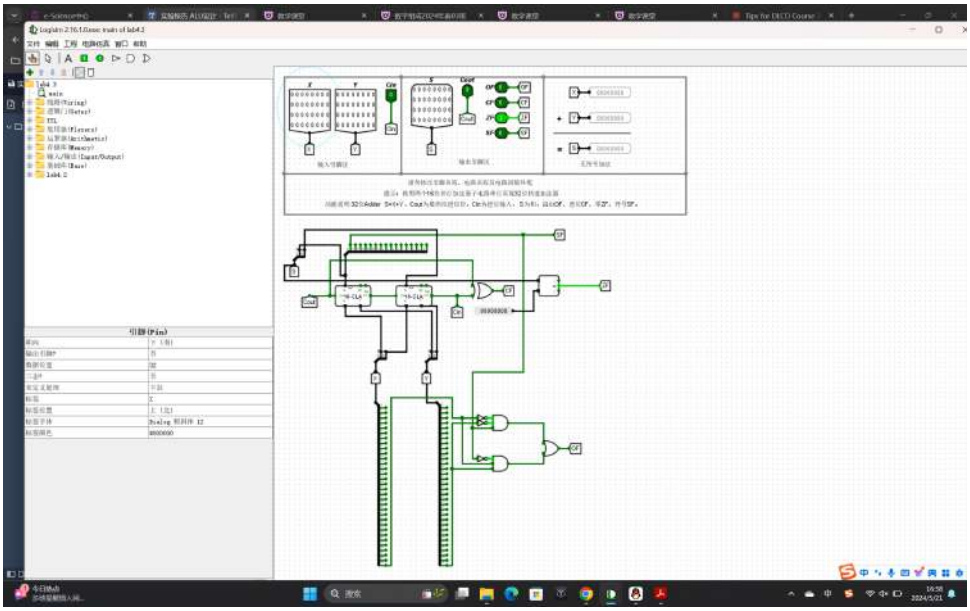

图 18: ZF

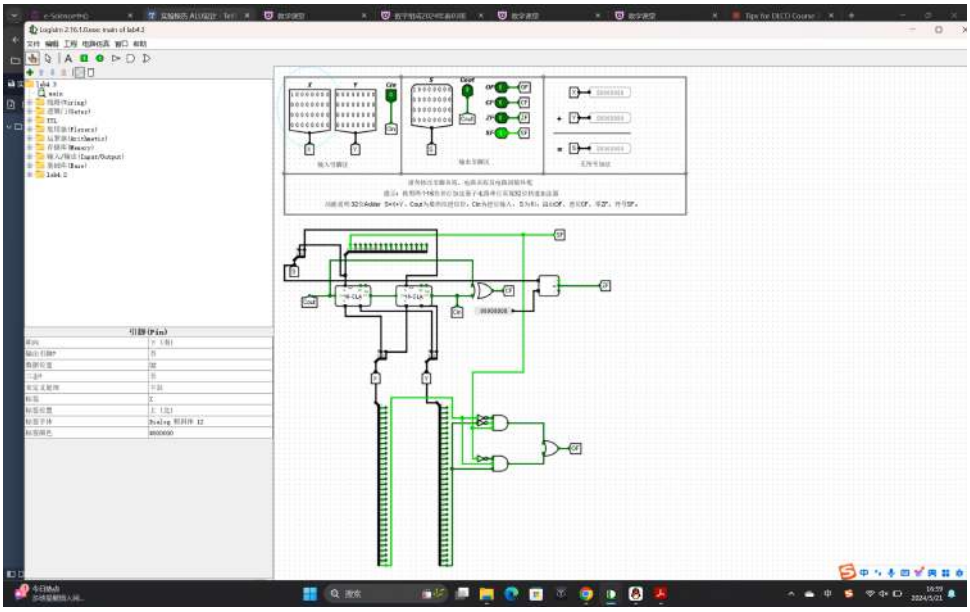

图 19: SF

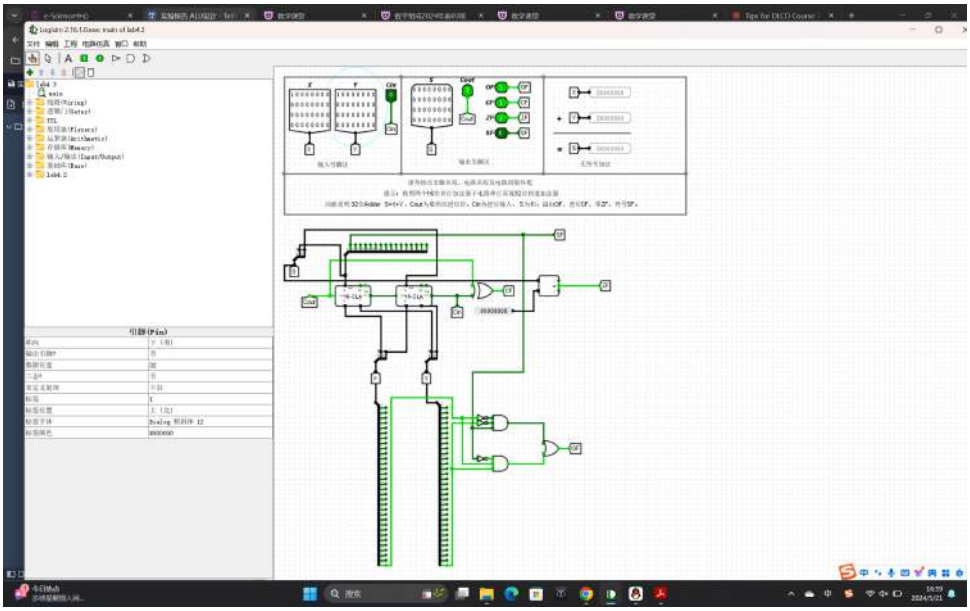

图 20: OF 和 Cout

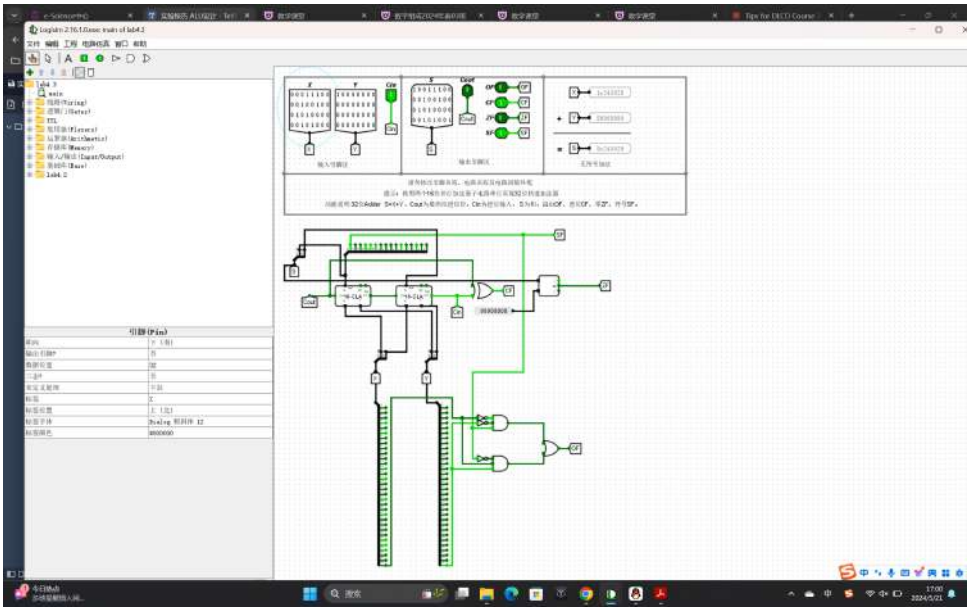

图 21: CF

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 32 位桶型移位器

1.4.1 实验内容

将实验二扩展成 32 位。

1.4.2 实验整体方案设计

最简单的方法就是仿照 8 位进行五个 32 位 MUX 的扩展。这个工作我已经在实验二思考题实现过了,就再贴一遍。本实验不需要底层模块设计图。

Datain	输入的数据
shamt	控制移动几位,自由修改
L/R	左移/右移
A/L	算术/逻辑

表 5: 功能表

1.4.3 实验原理图和电路图

原理图：有之前 8 位的原理图做基础，32 位只是简单扩展成移 1,2,4,8,16 而已，不需要原理图。电路图实现：

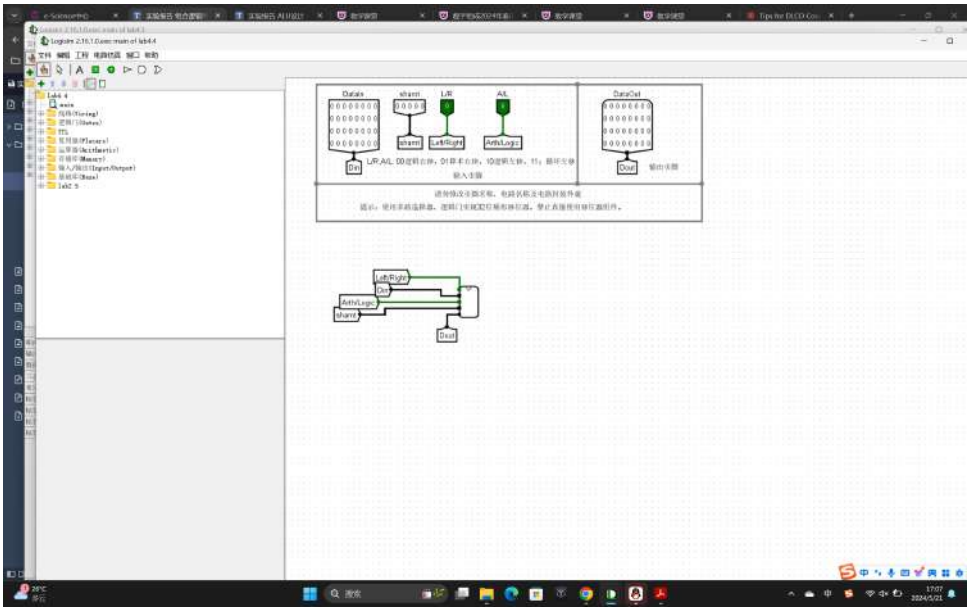

图 22: 电路图 1

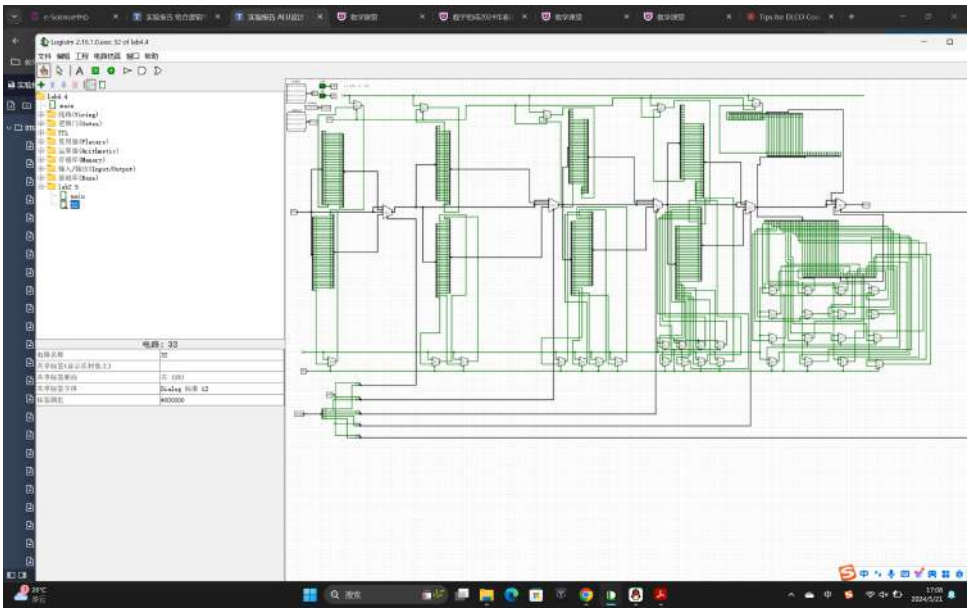

图 23: 电路图 2

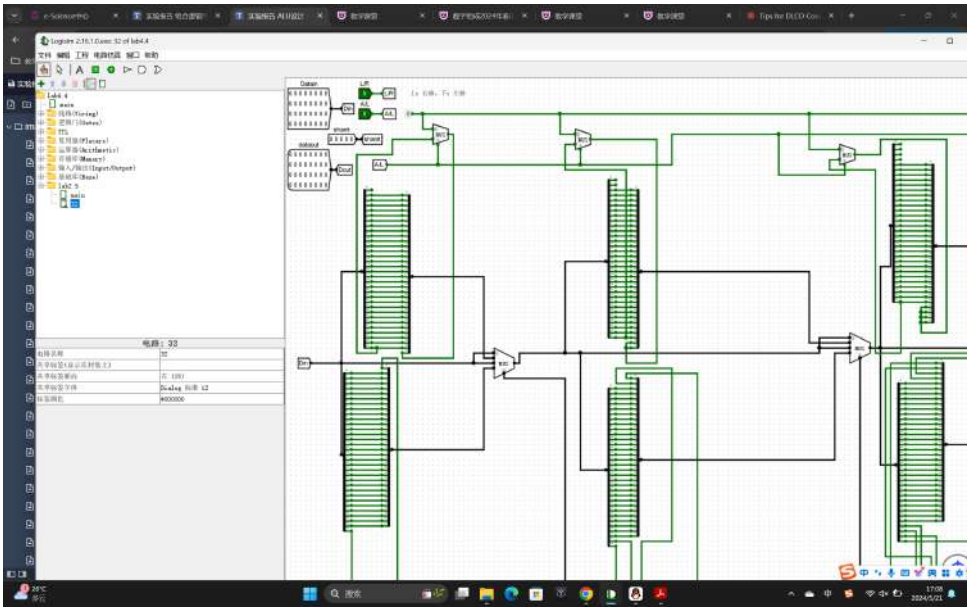

图 24: 电路图 3

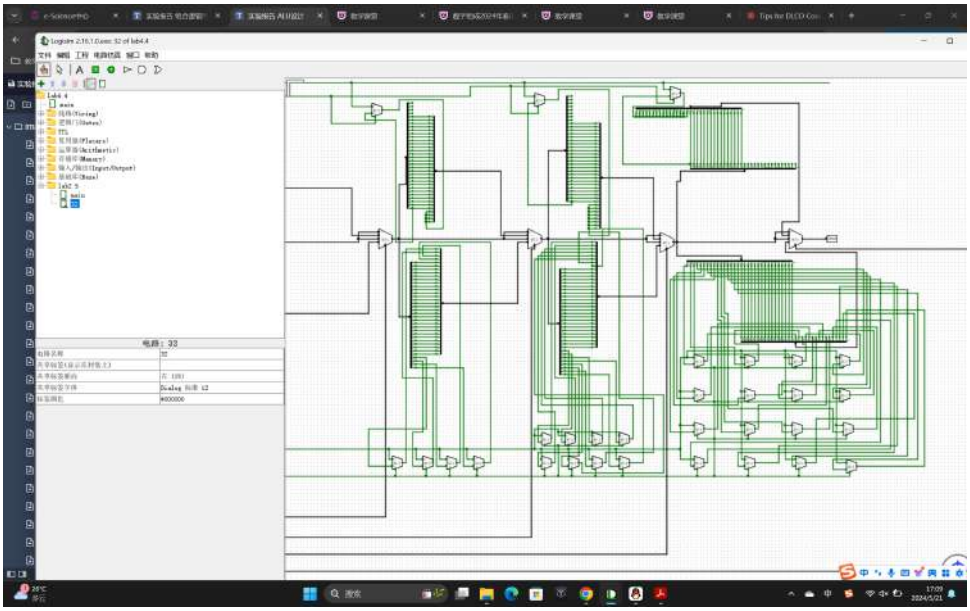

图 25: 电路图 4

1.4.4 实验数据仿真测试图

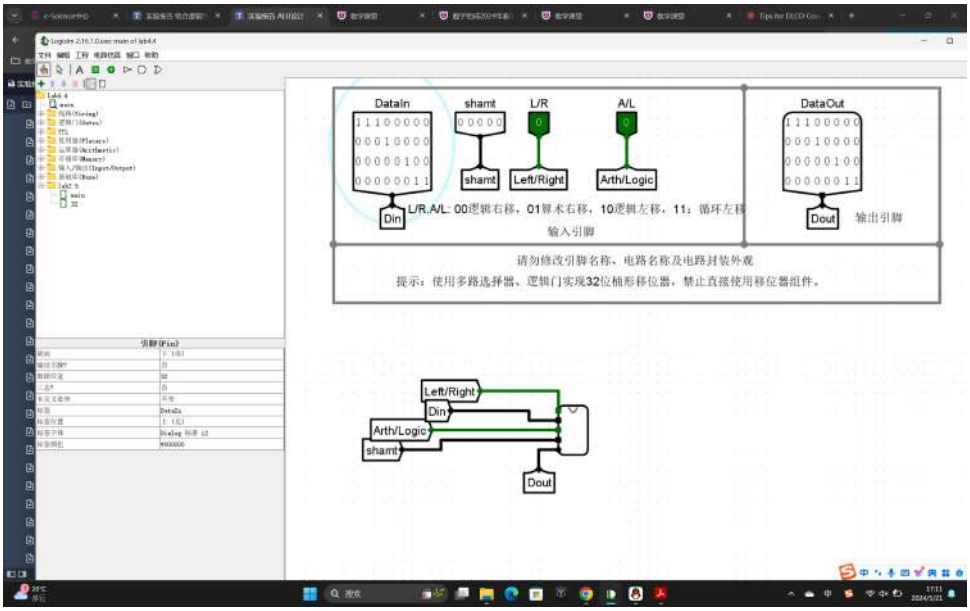

图 26: 初始状态

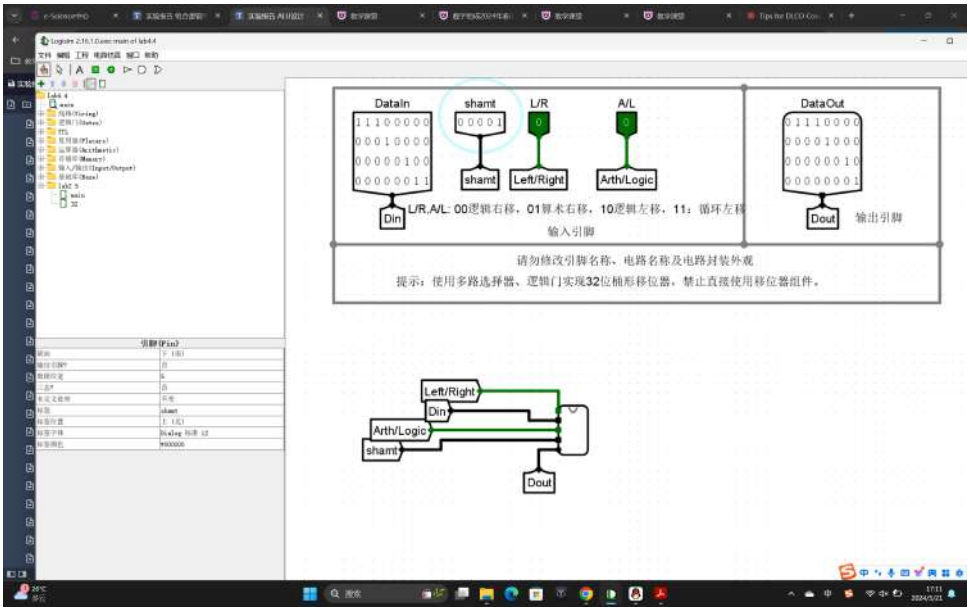

图 27: 逻辑右移一位

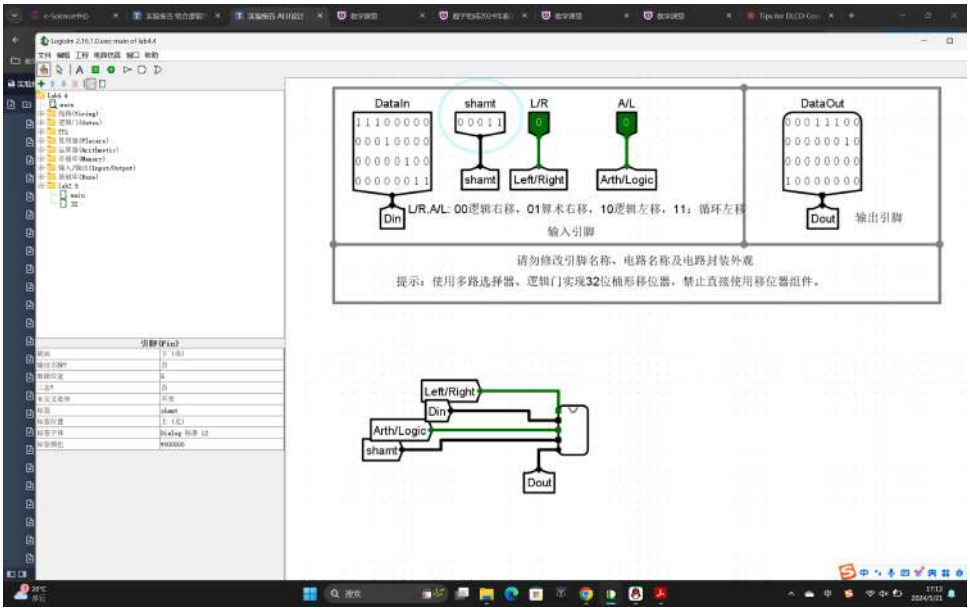

图 28: 逻辑右移三位

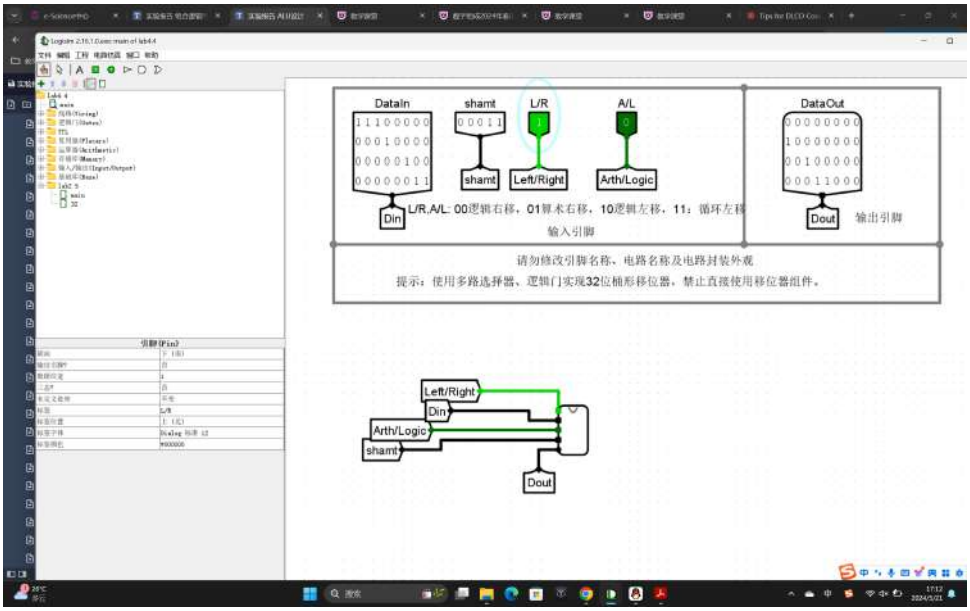

图 29: 逻辑左移三位

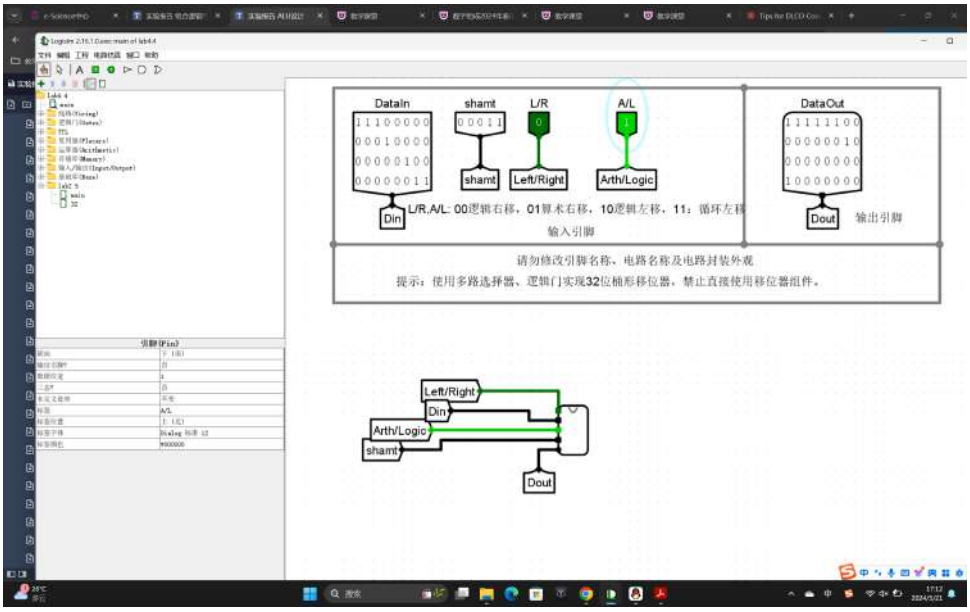

图 30: 算术右移三位

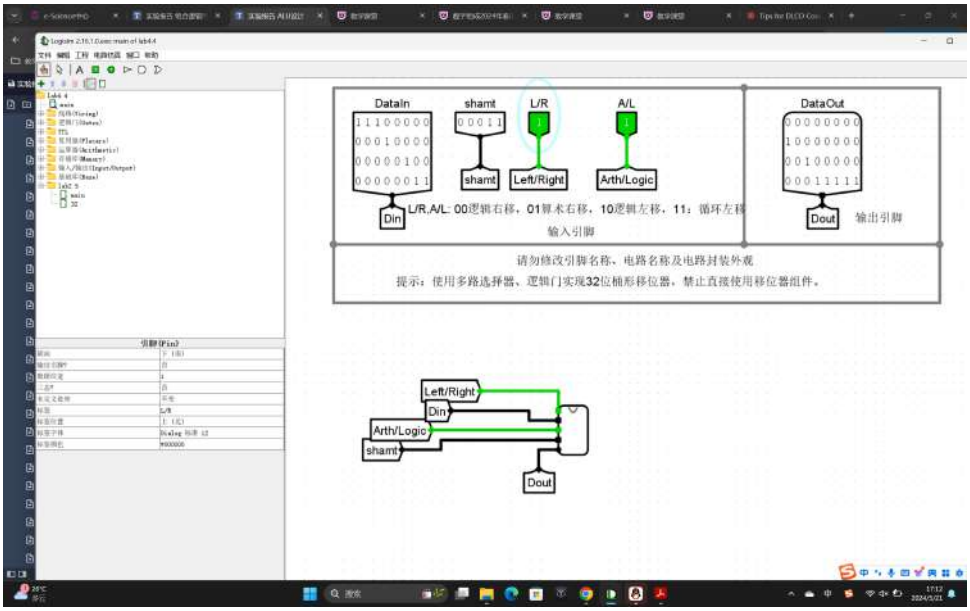

图 31: 循环左移三位

1.4.5 错误现象及分析

这个实验我没有改思考题二的实现电路的任何东西，直接过了，所以我想询问批实验二报告的助教（不是质问）为什么当时批的时候说我的 32 位有问题，我也把电路发过去了有图为证

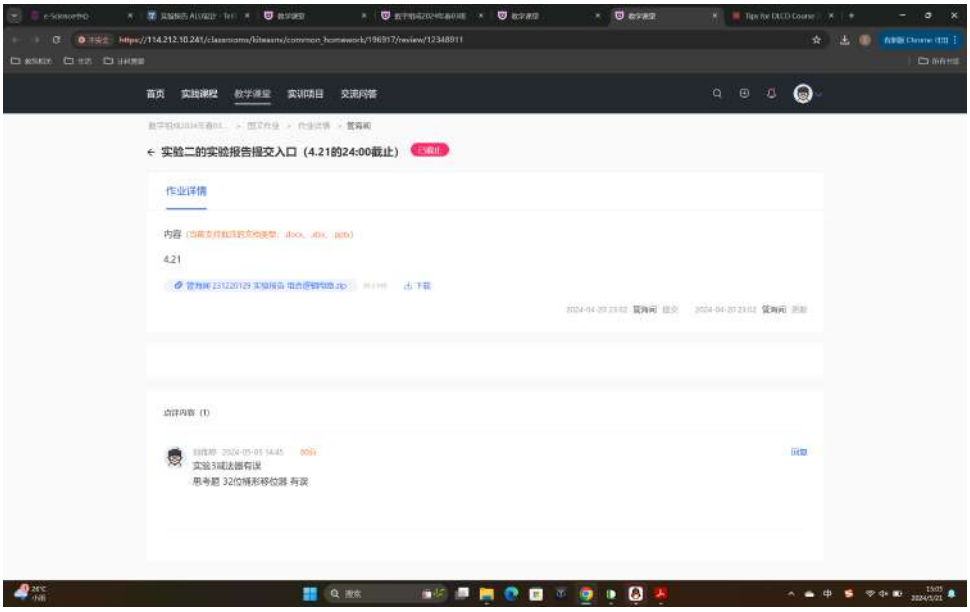

图 32: ???

1.5 ALU 设计实验

1.5.1 实验内容

实现 ALU 操作控制信号的生成和 ALU 的实现。(11 条指令)

1.5.2 实验整体方案设计

ALUCTR 只需要按照给出的卡诺图化简后的最简式极大项连线即可。对于主电路,按照原理图理解后连就行。SUBctr =m(2,3,8)

SIGctr = m2

ALctr = m13

SFTctr = m1

Opctr[2]= m(1,2,3,5,13,15)

Opctr[1]= m(2,3,4,6)

Opctr[0]= m(4,7,15)

ALUctr	控制信号
opctr 等·控制信号	分别控制不同指令
桶型移位器	逻辑右移一位
加法器,MUX	数据通路

表 6: 功能表

1.5.3 实验原理图和电路图

原理图：

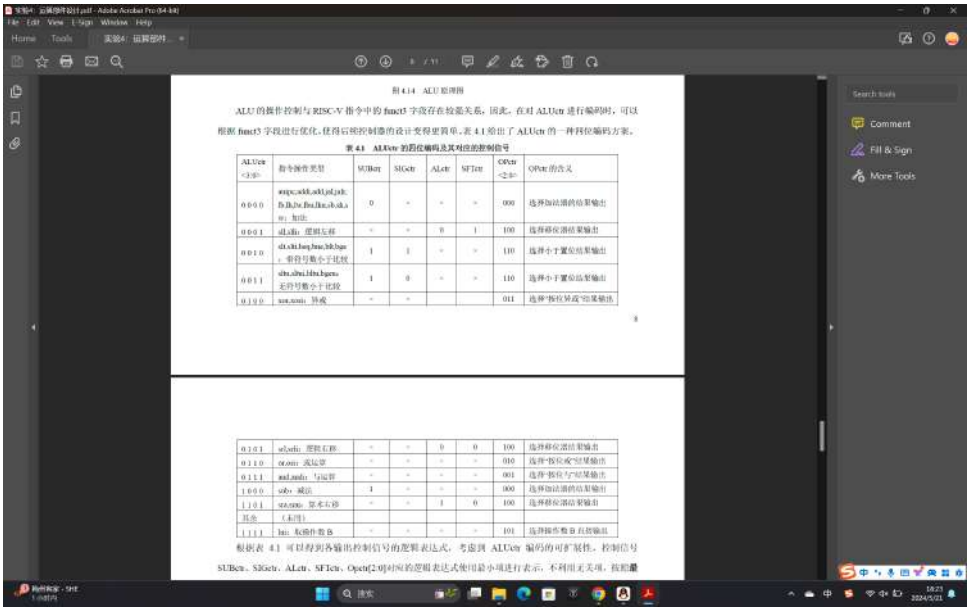

图 33: 原理图 1

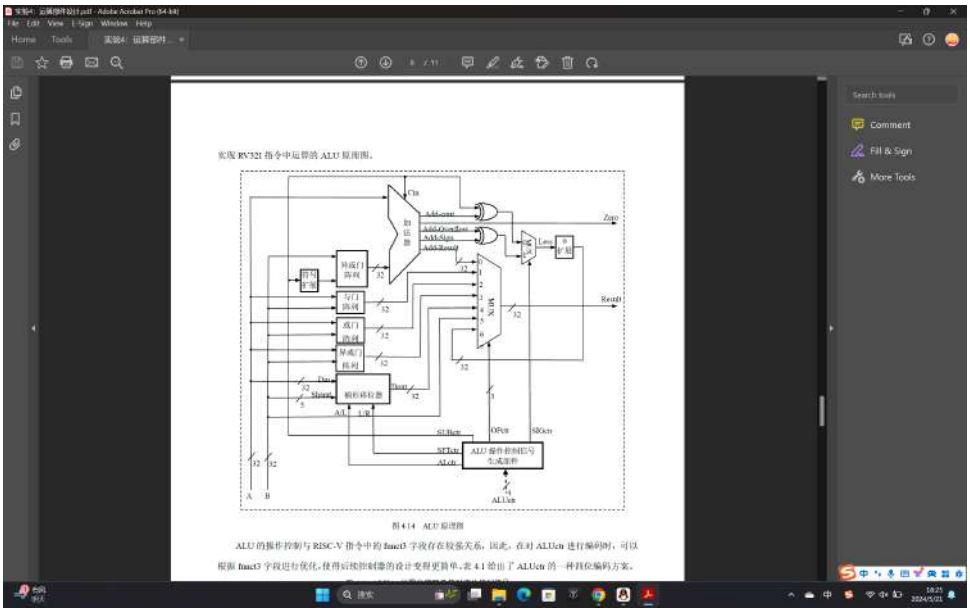

图 34: 原理图 2

电路图实现：

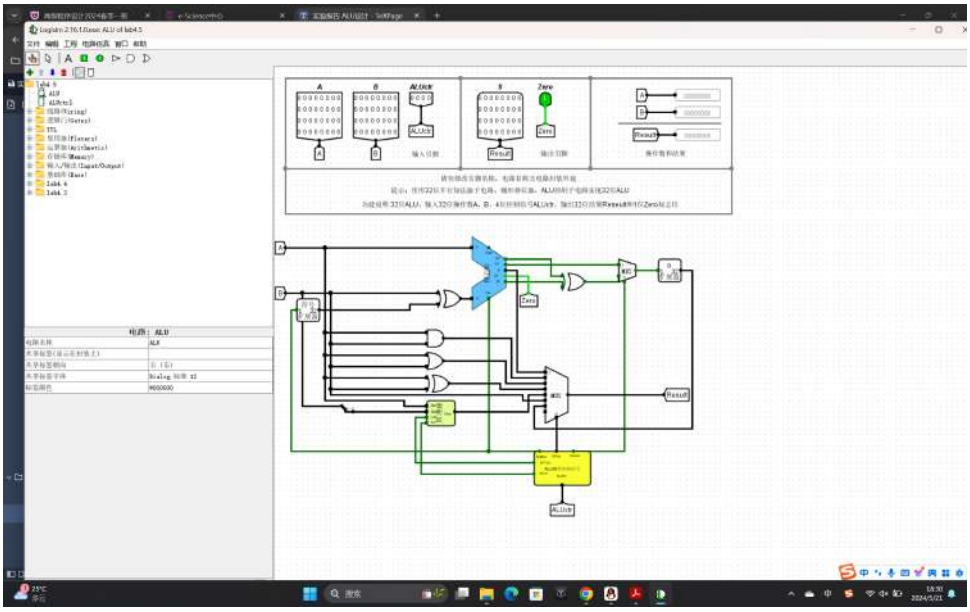

图 35: 主电路电路图

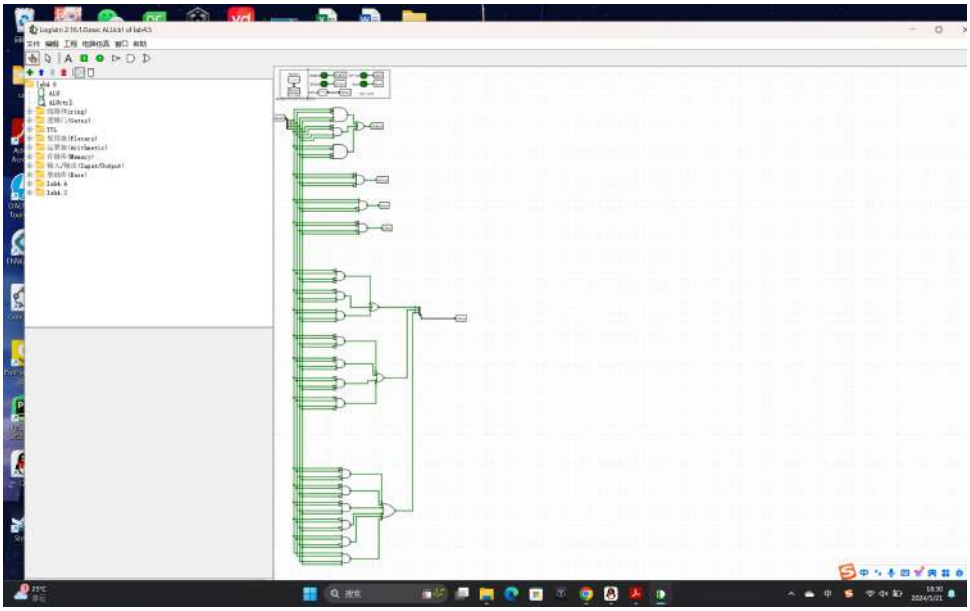

图 36: ALUCTR

1.5.4 实验数据仿真测试图

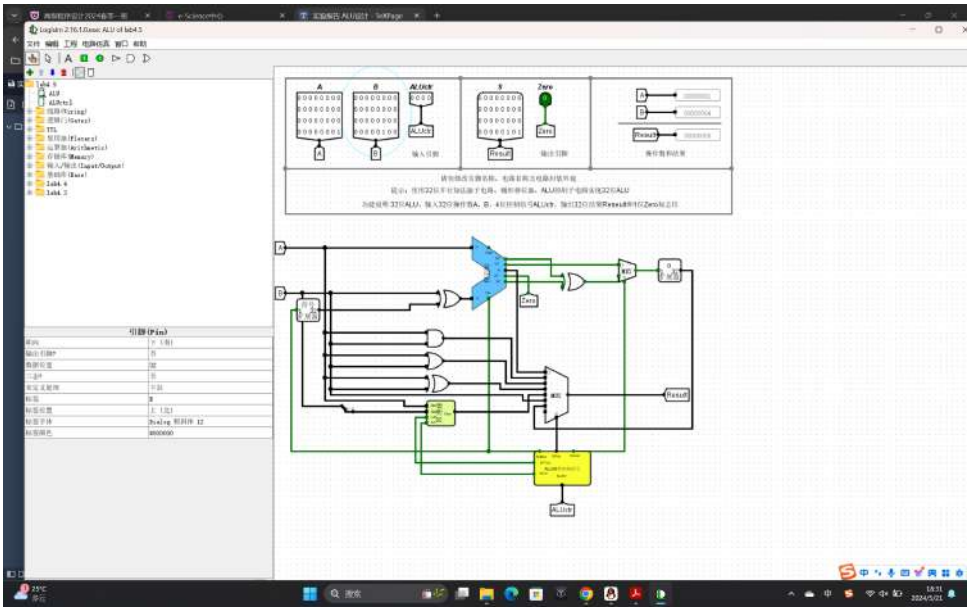

图 37: 加法

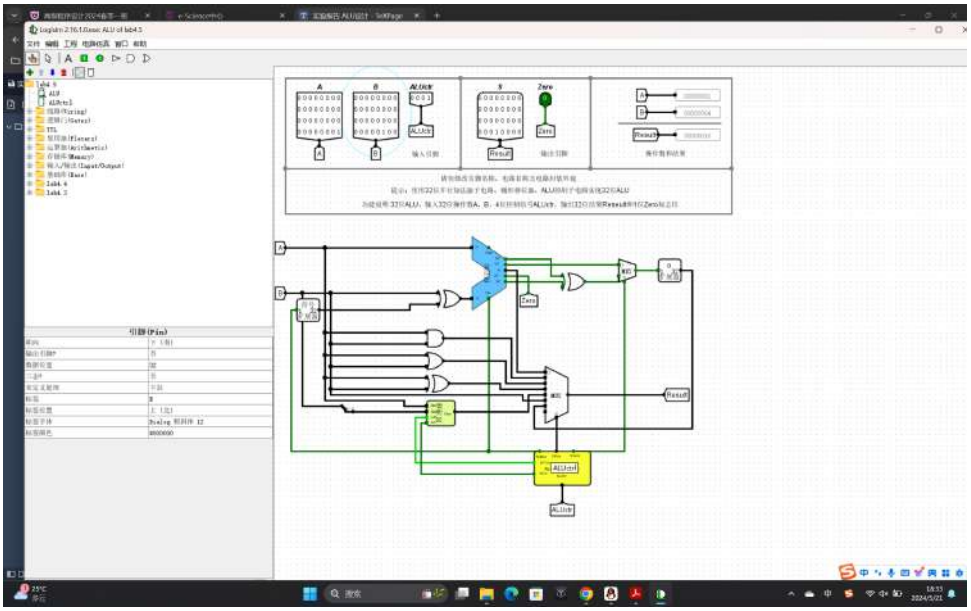

图 38: 逻辑左移 4 位

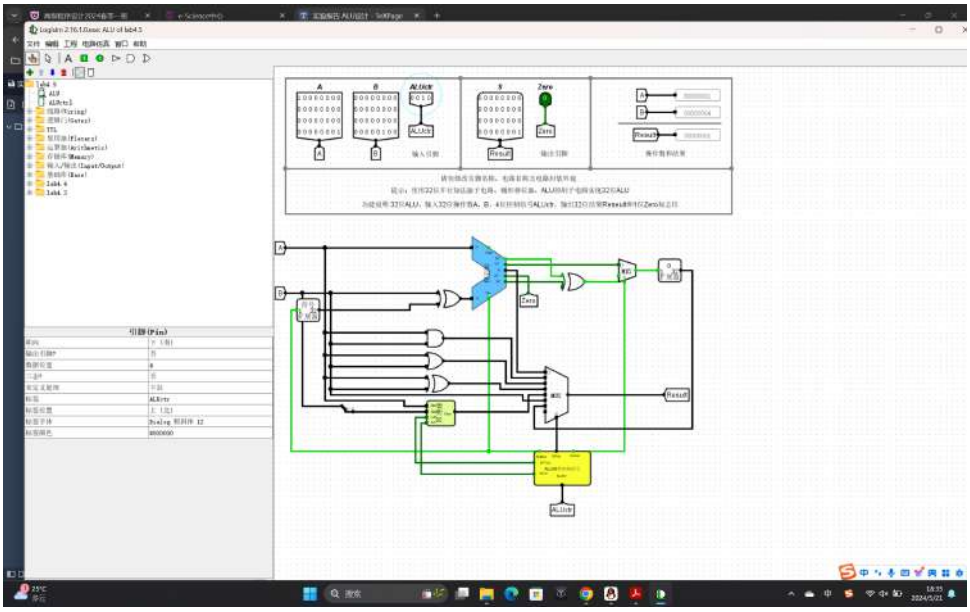

图 39: 带符号数小于比较

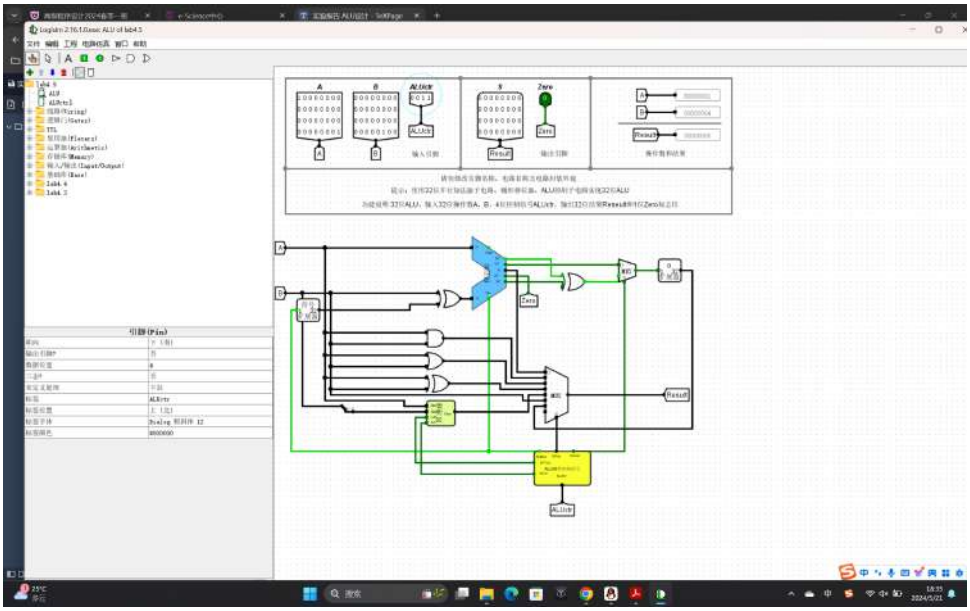

图 40: 按位异或

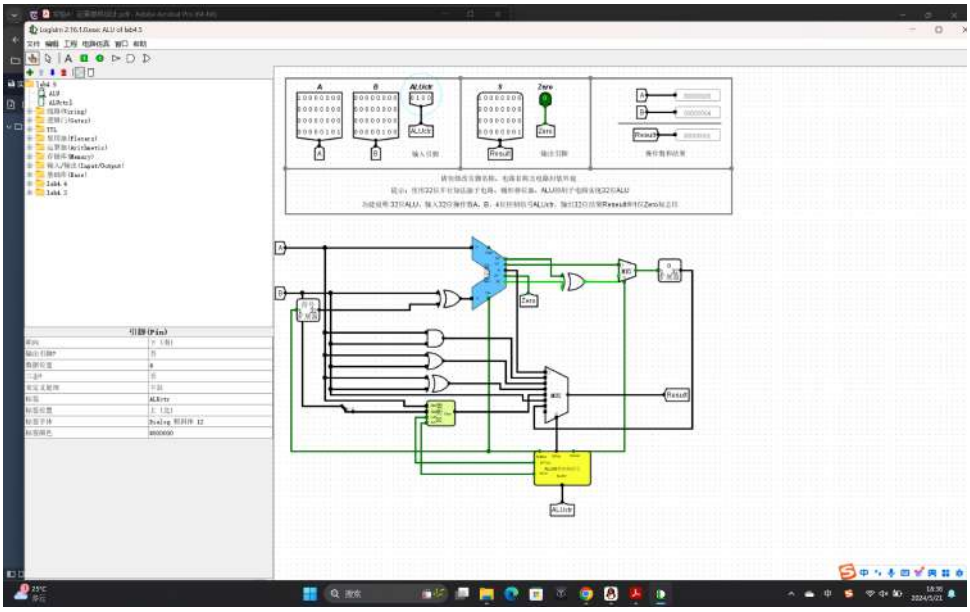

图 41: 逻辑右移

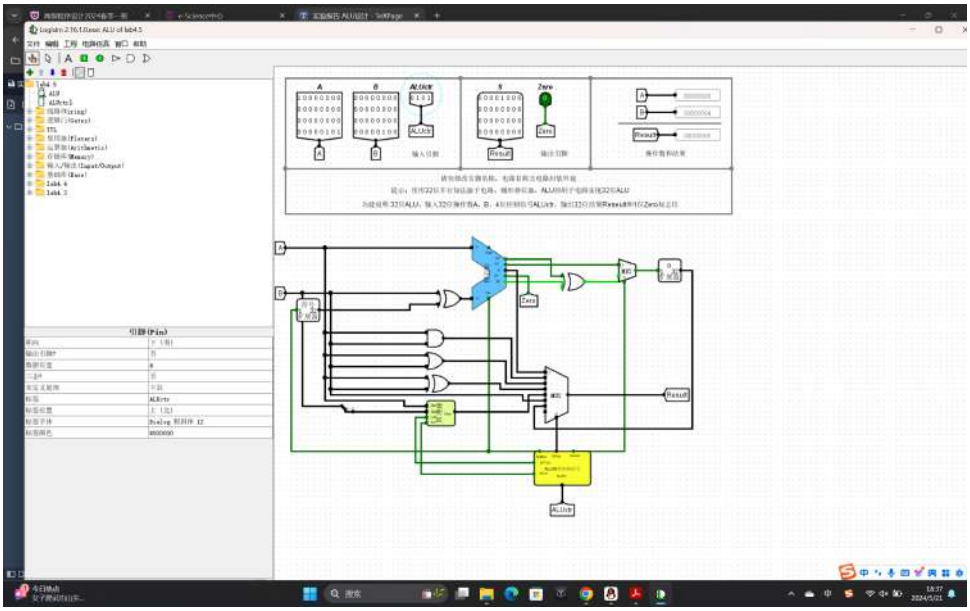

图 42: 或

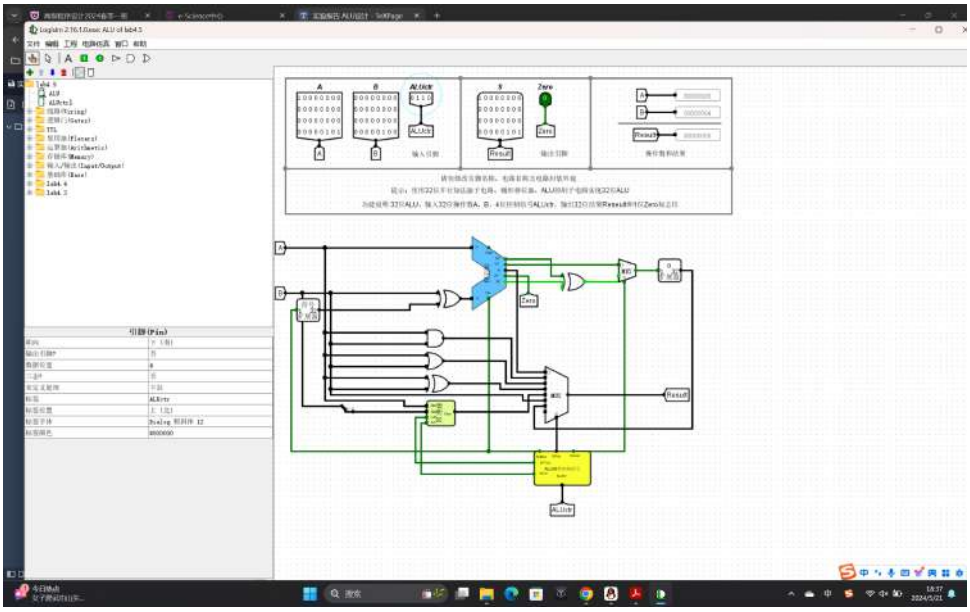

图 43: 与

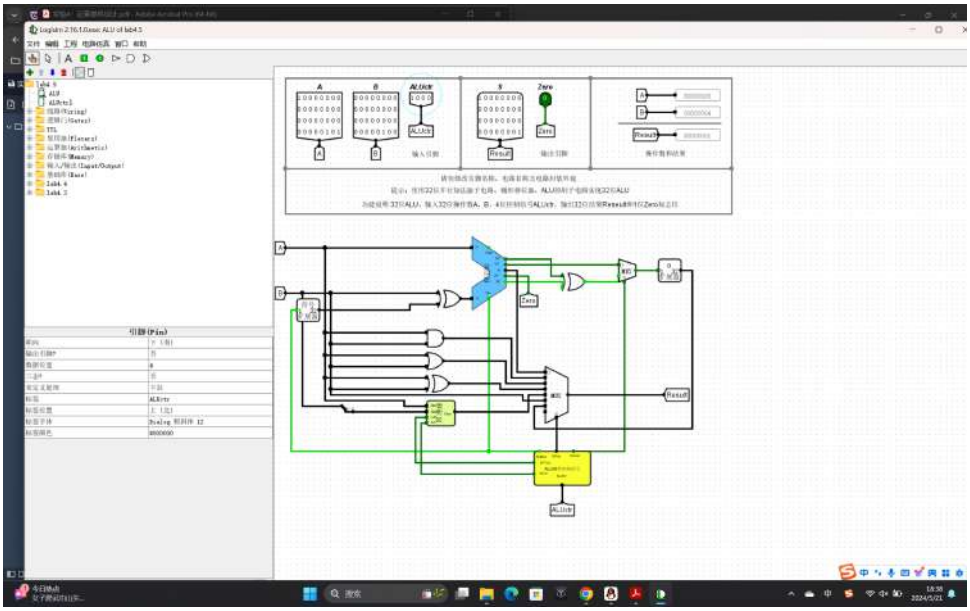

图 44: 减法

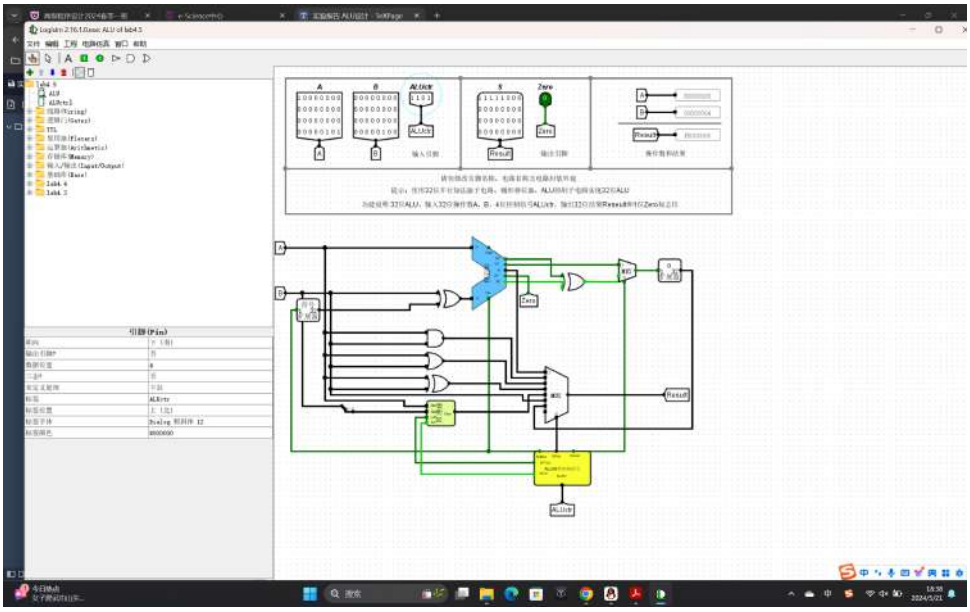

图 45: 算术右移

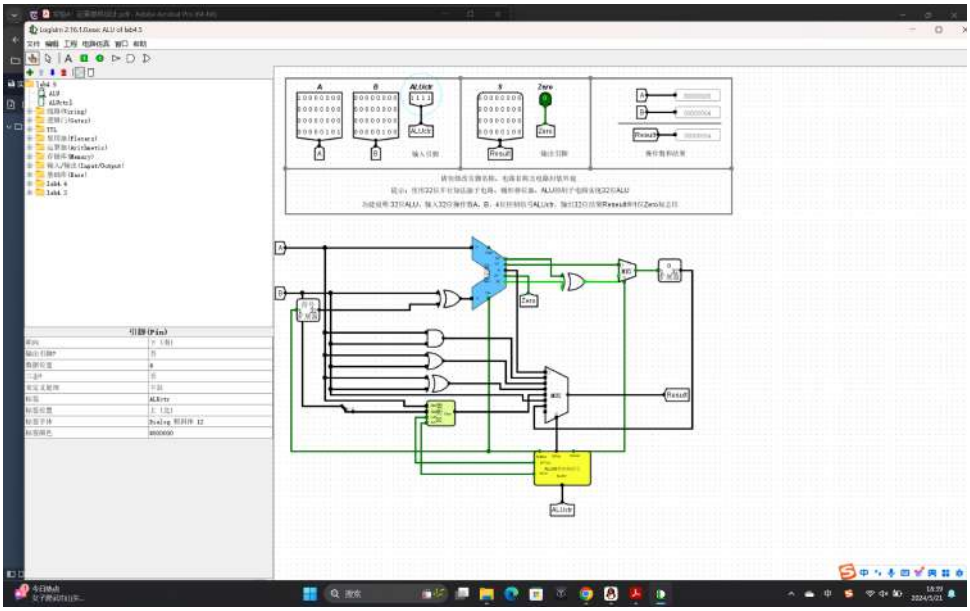

图 46: 取操作数 B

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,没有遇到任何错误。

2 思考题

2.1 思考题 1

1. 将实验 3 中的快速乘法器设计电路扩展到 32 位无符号数相乘，并探讨如何将该乘法器融合到实验中的 ALU 电路来实现乘法运算。

快速乘法器我之前实现的是 8 位的阵列乘法器，但是 32 位的是 32Œ32 的空间消耗，连线太折磨了，很难画出来，所以做了个大框架并且做了前三位，连线的规律参照教材快速乘法器一节，已明白原理，但是连线过于困难。

如果要融合到实验的 ALU 电路的话，还要进行数据截断，将 64 位的结果截断成低 32 位数据，用未用编码（如 1001）和 MUX 的一个空缺来融合如 ALU 电路。最后，我还是实现了一个和实验三 8 位乘法器相同的乘法器，将其扩展成了 32 位。如果有充分的时间和空间，32 位阵列乘法器是能完成的，（ps: 这个工作量真的对吗，还是题意是要连 Wallistree 型的）具体测试情况助教用文件自己测试一下即可

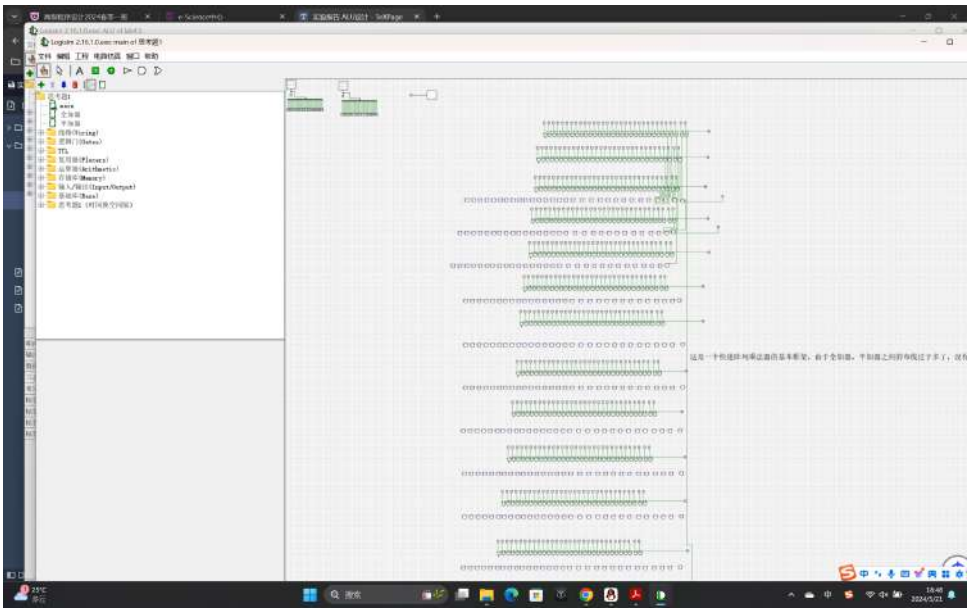

图 47: 阵列乘法器概览 1

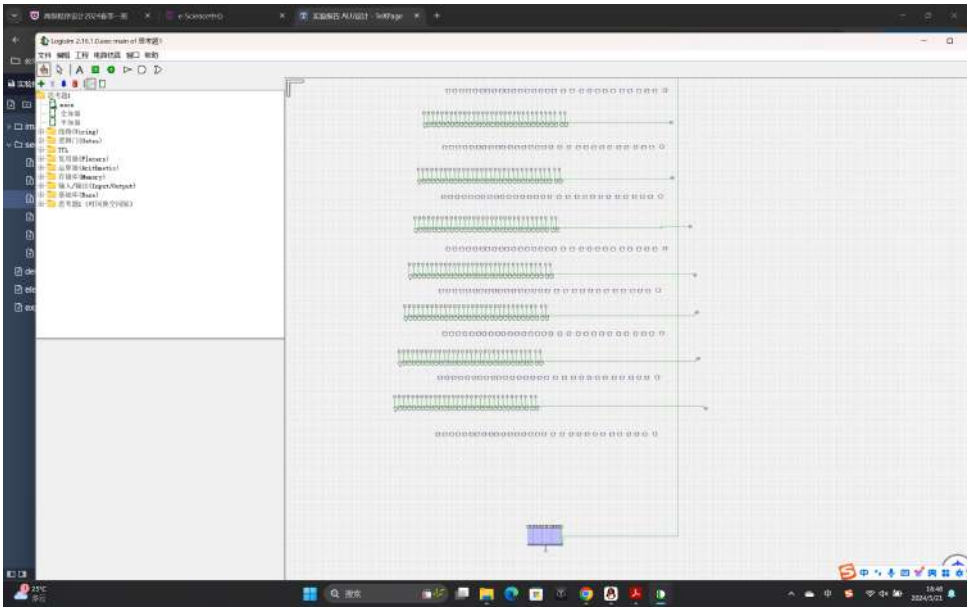

图 48: 概览 2

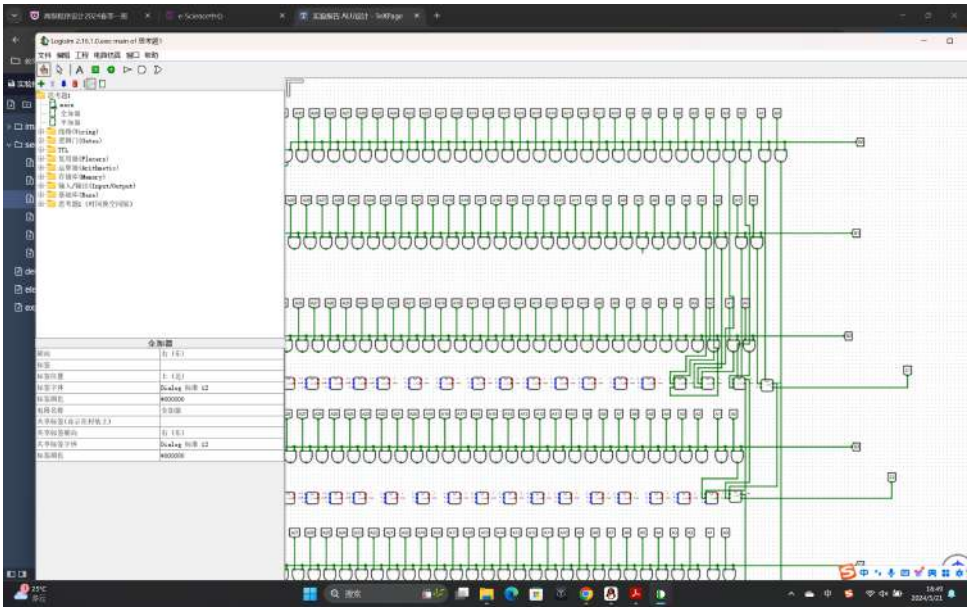

图 49: 概览 3

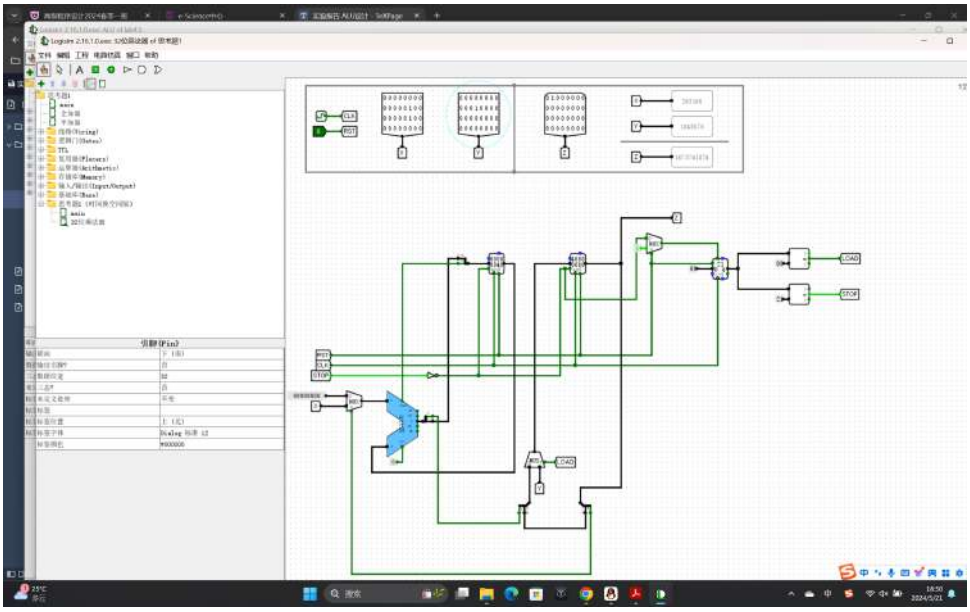

图 50: 非快速 32 位乘法器

2.2 思考题 2

2. 在 RV32I 中新增一条指令，然后在 ALU 中新增一个新运算，并通过测试数据进行验证。

这里加了一个乘法指令，对应的 ALUCTR 是 1001（之前的未用），主线连接利用了 MUX 的未用口，加了一个乘法器。（也可以当成思考题第一题的复用）

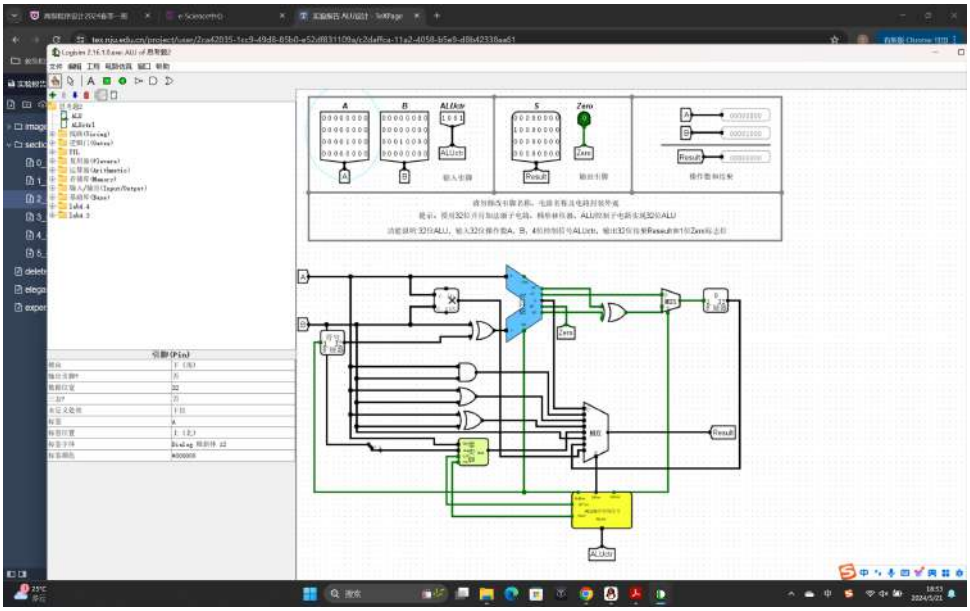

图 51: 主线的不同

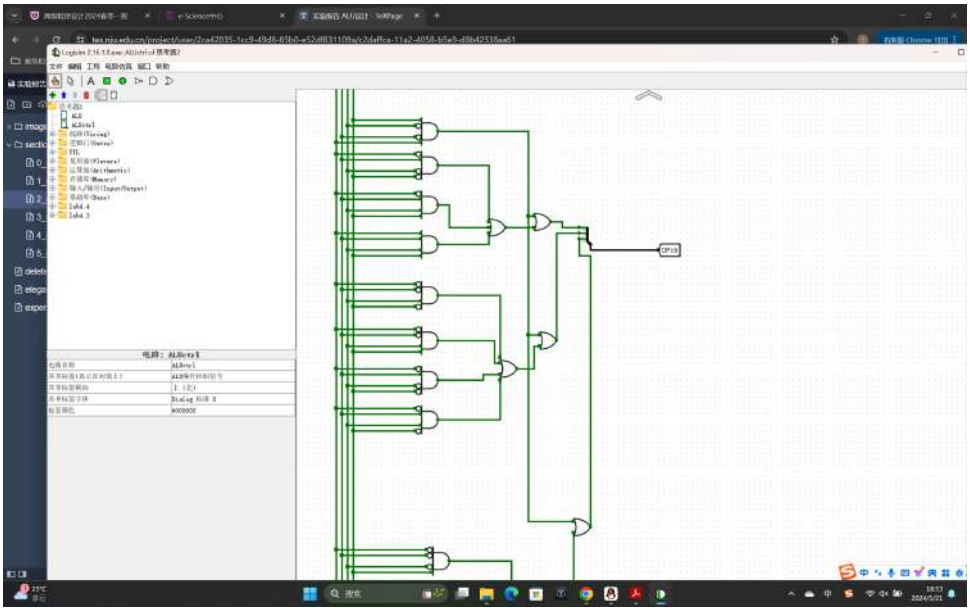

图 52: ALUCTR 的不同

2.3 思考题 3

3. 如何实现 32 位无符号数除法器?

和原先实验三的乘法器类似,不同的只是每次 clk 都让大的数减小的数,知道不能减为止,输出余数。

下面是电路实现。

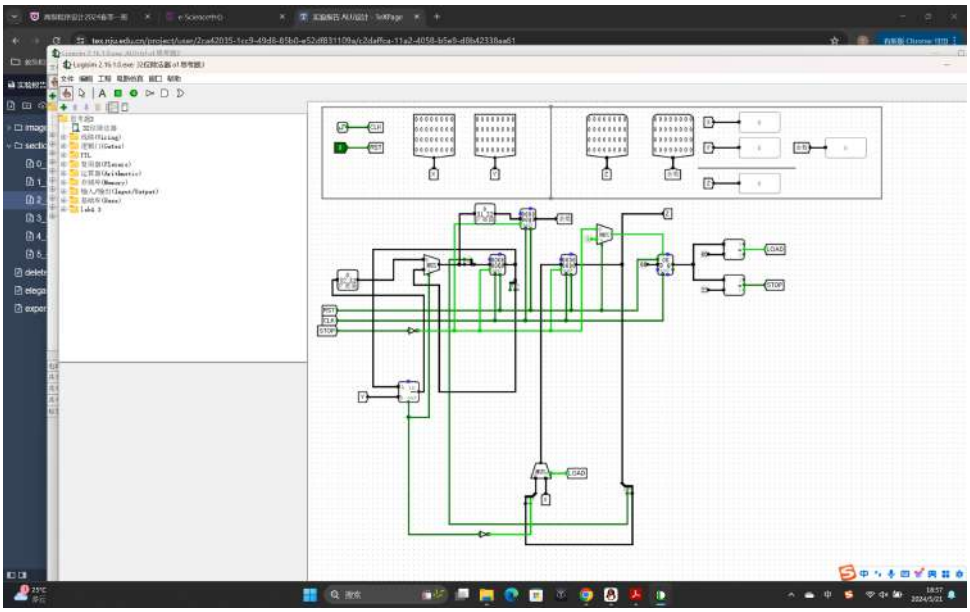

图 53: 电路图

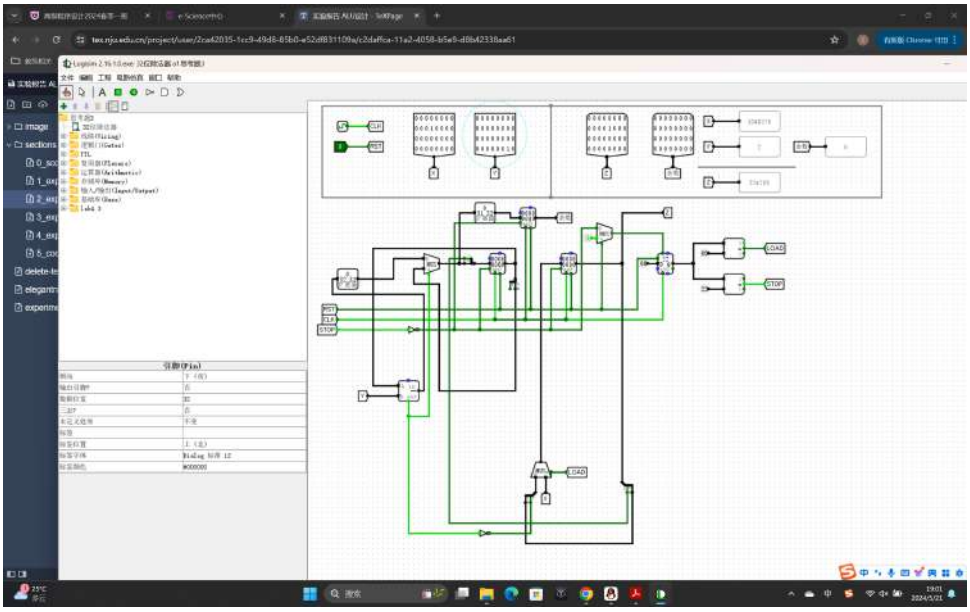

图 54: 测试