《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 3-8 译码器

1.1.1 实验内容

设计一个由反相逻辑门电路构成的 3-8 译码器,并对电路进行仿真测试,以验证电路的功能。

1.1.2 实验整体方案设计

由真值表化简最小项,以及芯片的原理图,可以做出。本实验不需要顶层模块设计图。

G1,G2A_L,G2B_L	三个使能端,用于芯片的扩展
Y0 Y7	输出端,输出译码的值,对应十进制数
ABC	输入端,代表三位二进制数

表 1: 3-8 译码器引脚作用

1.1.3 实验原理图和电路图

原理图实现:电路图实现:

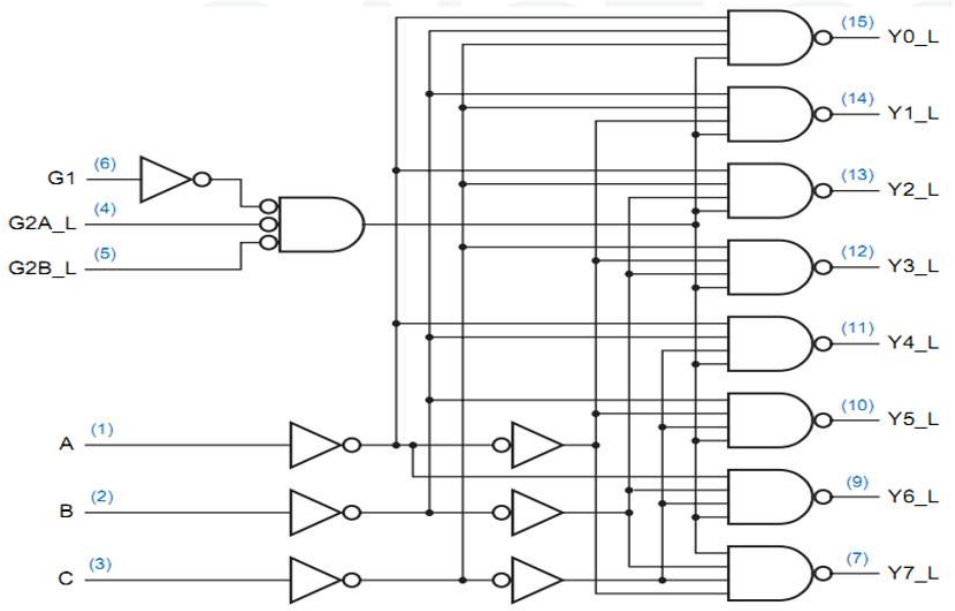

图 1: 3-8 译码器原理图

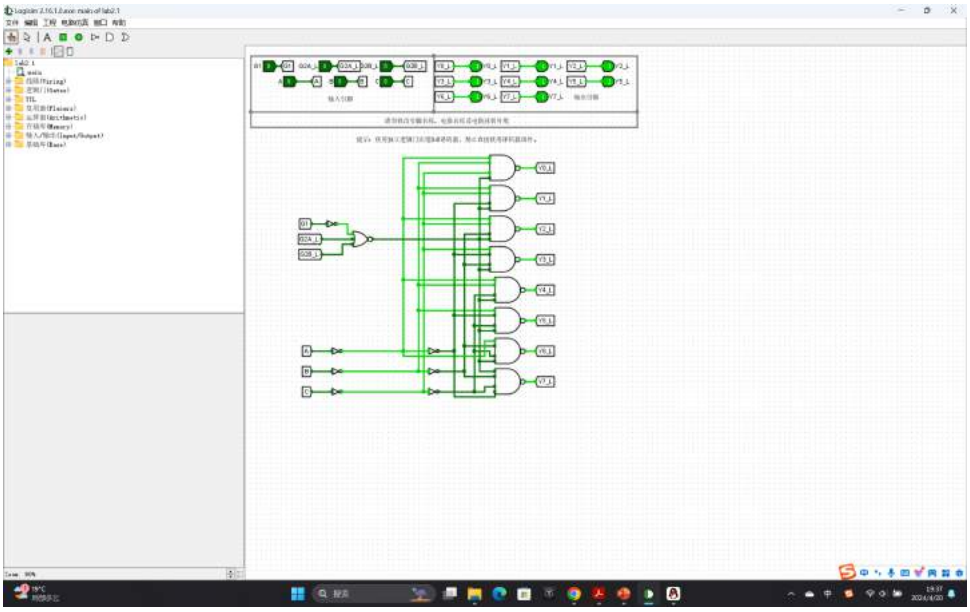

图 2: 3 输入多数表决器电路图

1.1.4 实验数据仿真测试图

分别展示了真值表对应的 4 种不同的主要情况。

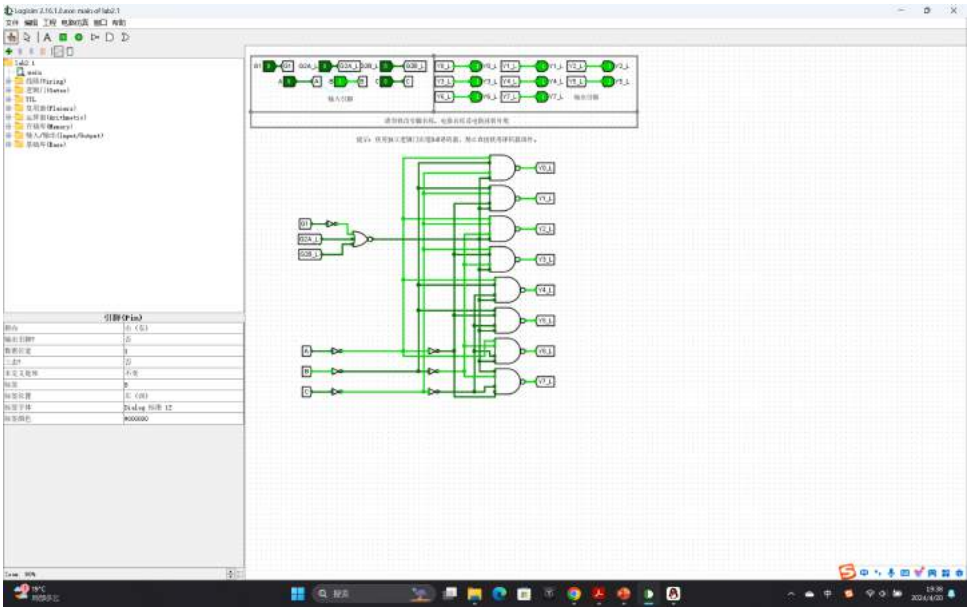

图 3: 3-8 译码器仿真测试图 1

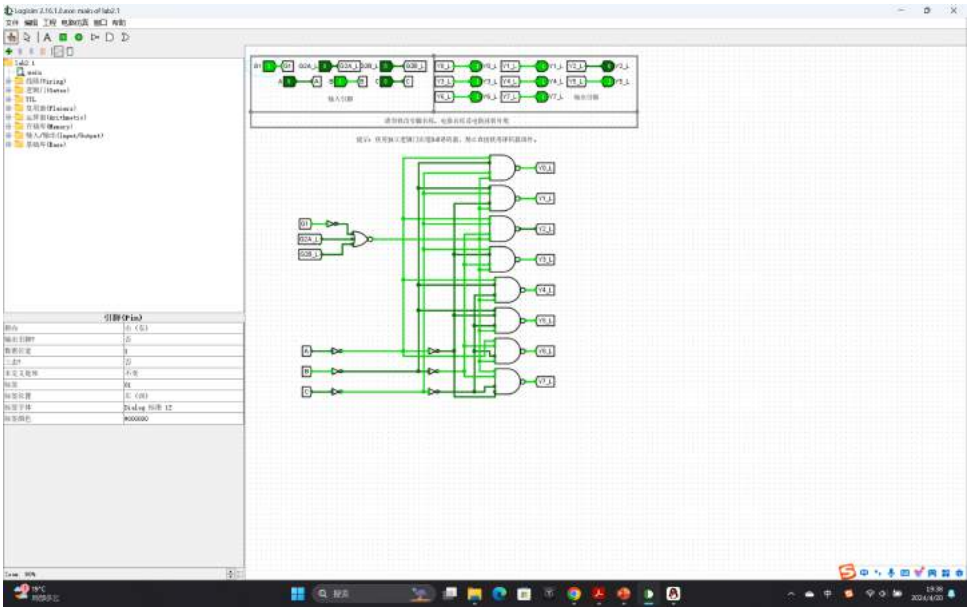

图 4: 3-8 译码器仿真测试图 2

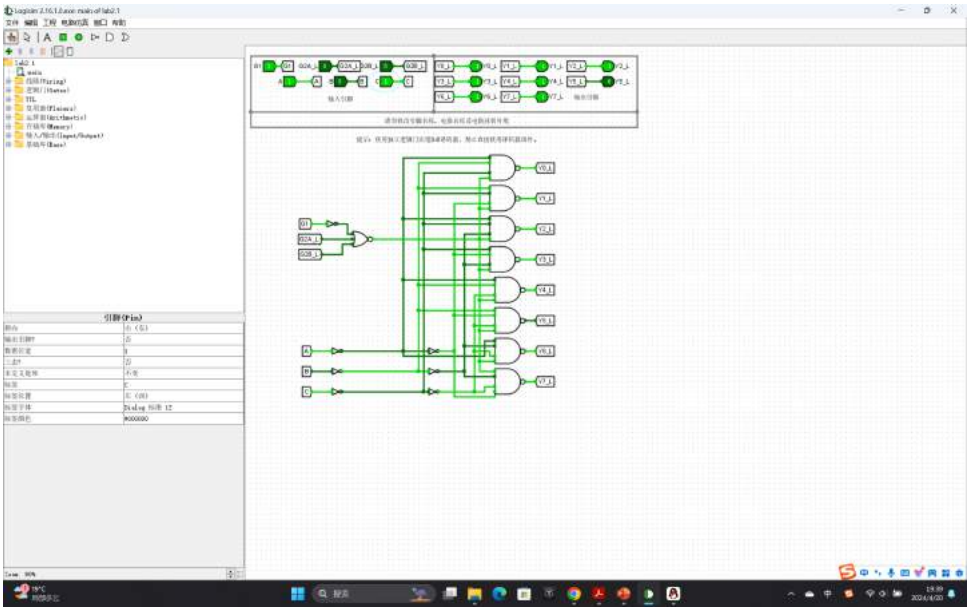

图 5: 3-8 译码器仿真测试图 3

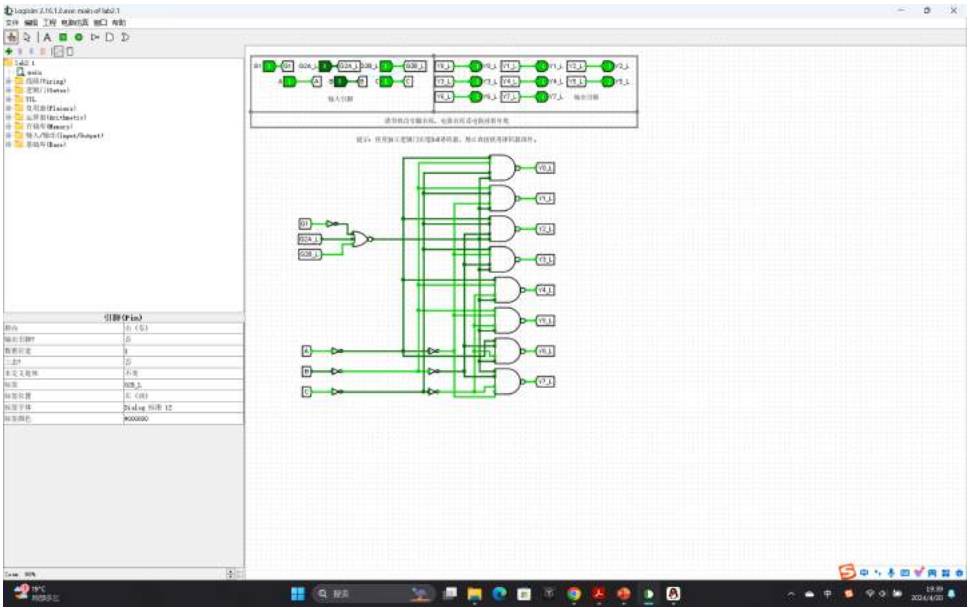

图 6: 3-8 译码器仿真测试图 4

G1	G2A_L	G3A_L	C	B	A	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0	x	x	x	x	x	1	1	1	1	1	1	1	1
x	1	x	x	x	x	1	1	1	1	1	1	1	1
x	x	1	x	x	x	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	1	0	0	0	0	0	0	1	0
1	0	0	0	1	0	0	0	0	0	0	1	0	0
1	0	0	0	1	1	0	0	0	0	1	0	0	0
1	0	0	1	0	0	0	0	0	1	0	0	0	0
1	0	0	1	0	1	0	0	1	0	0	0	0	0
1	0	0	1	1	0	0	1	0	0	0	0	0	0
1	0	0	1	1	1	1	0	0	0	0	0	0	0

表 2: 3-8 译码器真值表

1.1.5 错误现象及分析

在完成实验的过程中,没有遇到任何错误。

1.2 8-3 优先级编码器

1.2.1 实验内容

优先级编码器原理图,设计一个由逻辑门电路构成的 8-3 优先级编码器,并将编码器输出连接到一个十六进制数码管,通过数码管的输出显示来验证和测试电路。

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。

I0 I7	7 个输入端,有优先级
O0,O1,O2	输出端,二进制解码

表 3: 8-3 引脚作用

1.2.3 实验原理图和电路图

原理图:

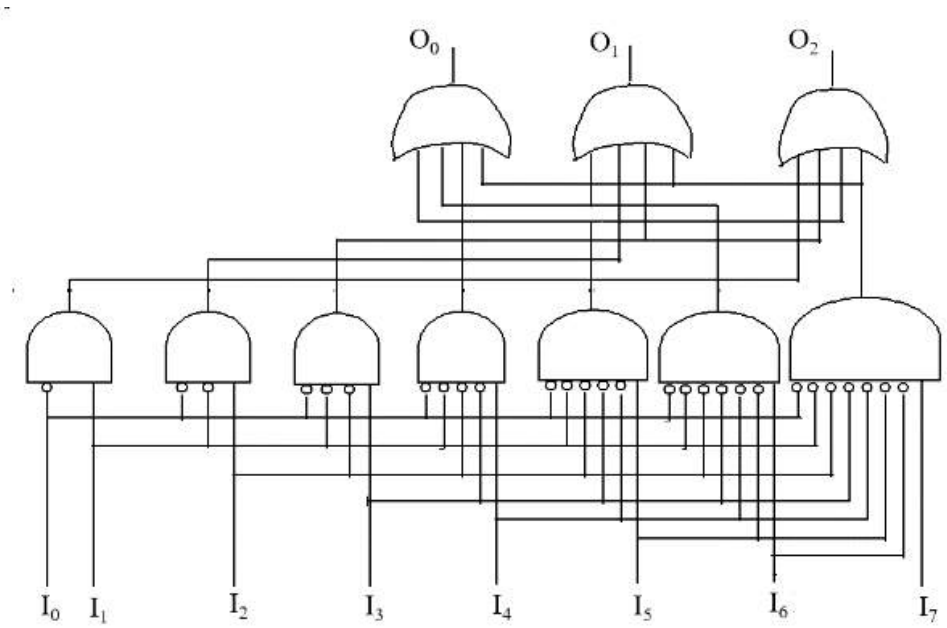

图 7: 8-3 原理图

电路图实现:

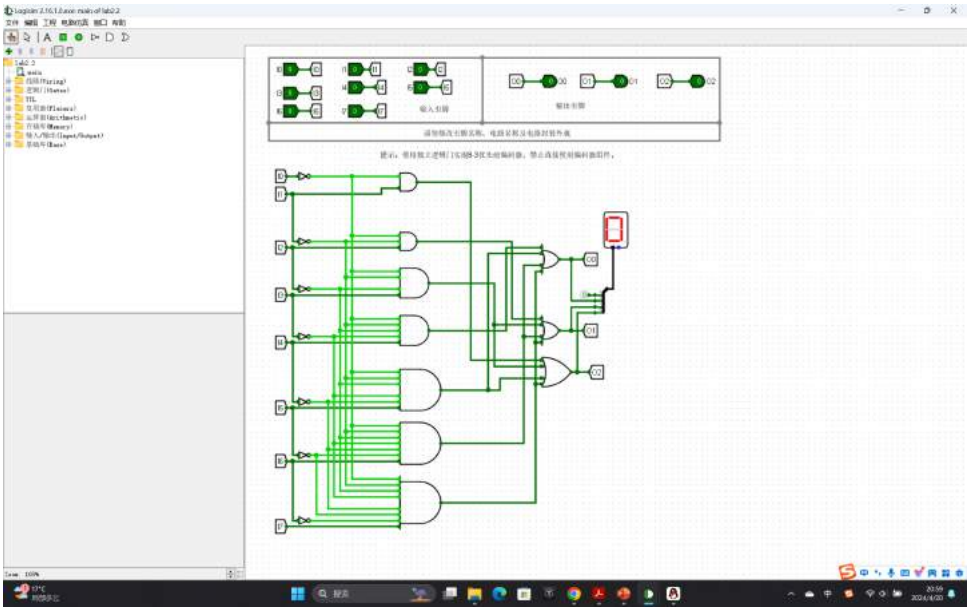

图 8: 8-3 电路图

1.2.4 实验数据仿真测试图

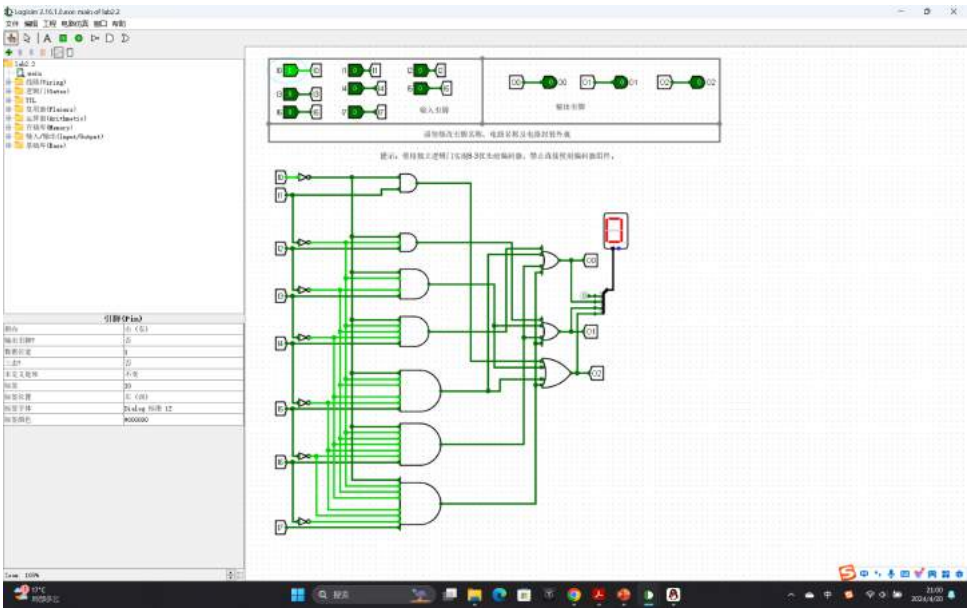

图 9: 8-3 仿真测试图 1

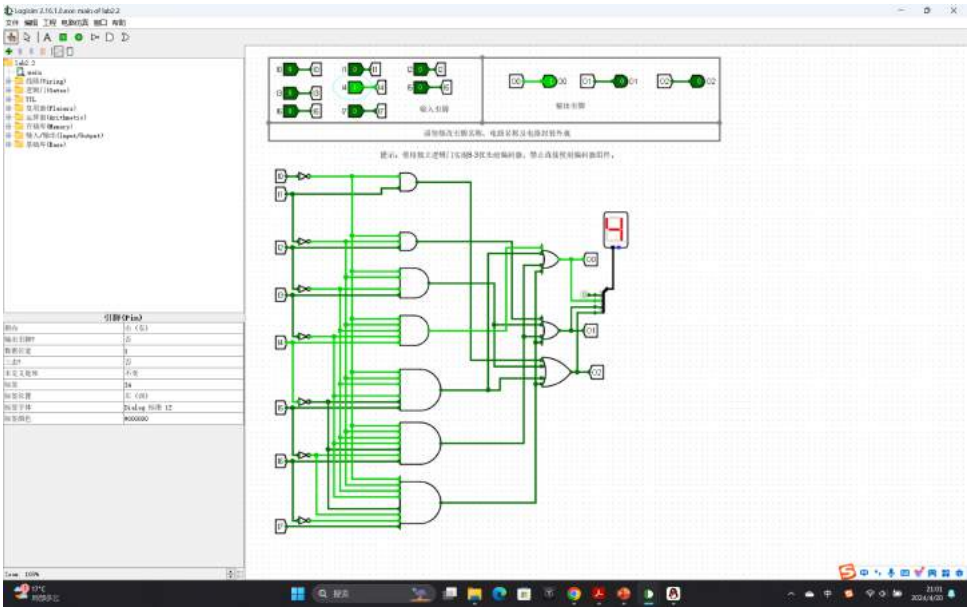

图 10: 8-3 仿真测试图 2

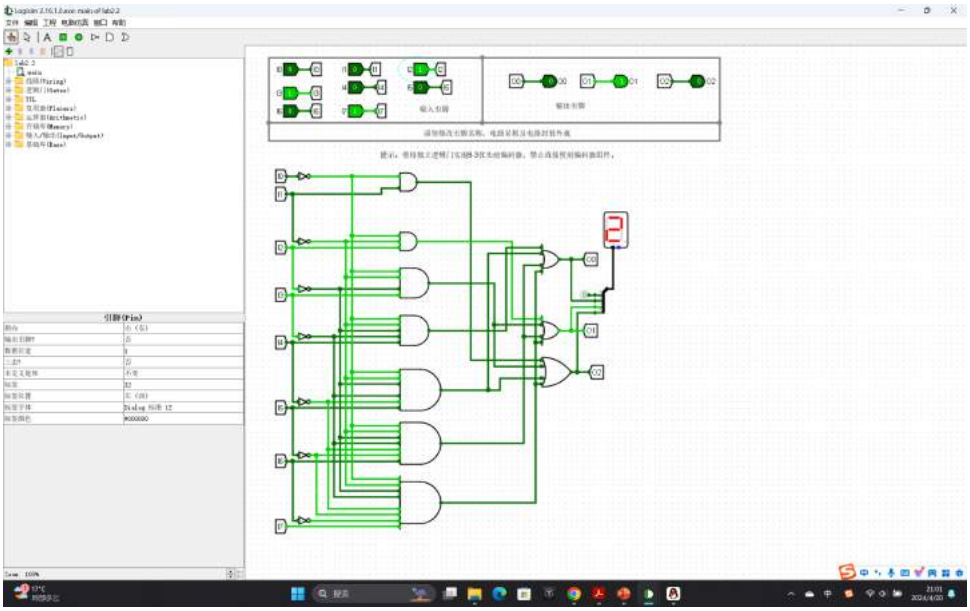

图 11: 8-3 仿真测试图 3

I0	I1	I2	I3	I4	I5	I6	I7	O0	O1	O2
1	x	x	x	x	x	x	x	0	0	0
0	1	x	x	x	x	x	x	0	0	1
0	0	1	x	x	x	x	x	0	1	0
0	0	0	1	x	x	x	x	0	1	1
0	0	0	0	1	x	x	x	1	0	0
0	0	0	0	0	1	x	x	1	0	1
0	1	1	0	0	0	1	x	1	1	0
0	0	0	0	0	0	0	1	1	1	1

表 4: 8-3 真值表

1.2.5 错误现象及分析

在完成实验的过程中,没有遇到任何错误。

1.3 加减法器

1.3.1 实验内容

串联 4 个全加器子电路实现 4 位串行进位加法器。将加数、被加数和和分别连接到 16 进制数码显示管进行验证。

1.3.2 实验整体方案设计

全加器和第一次实验的一样。

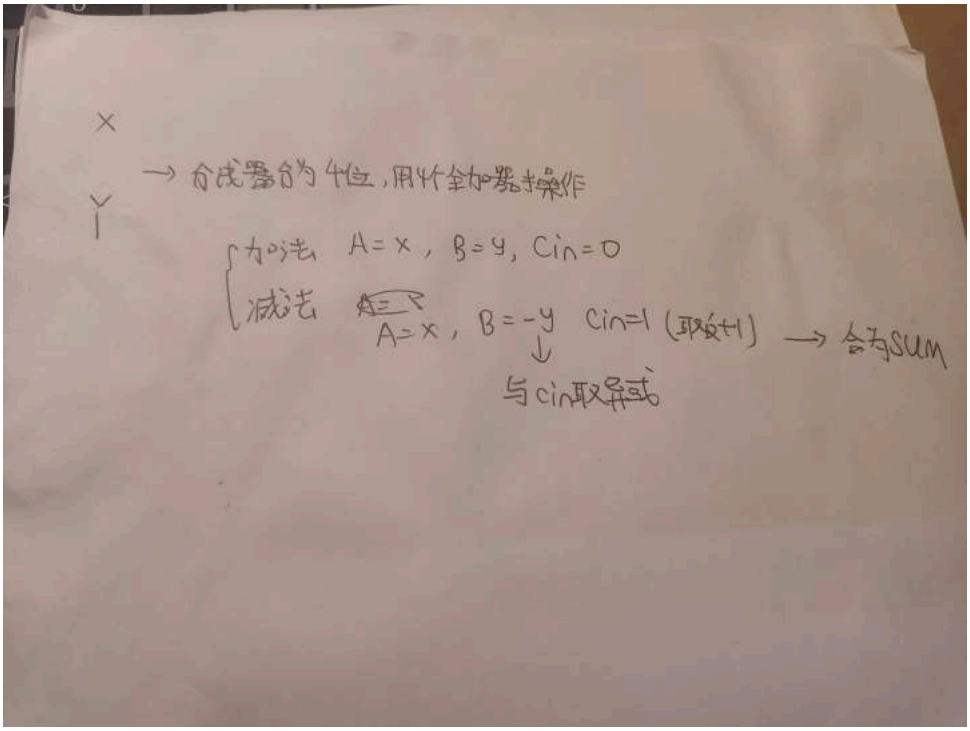

图 12: 思路

按照思路即可。

Cin	控制加减法，并作为进位
XY	两个四位用于加减法的数
SUM	结果
Cout	溢出位

表 5: 加减法器引脚作用

1.3.3 实验原理图和电路图

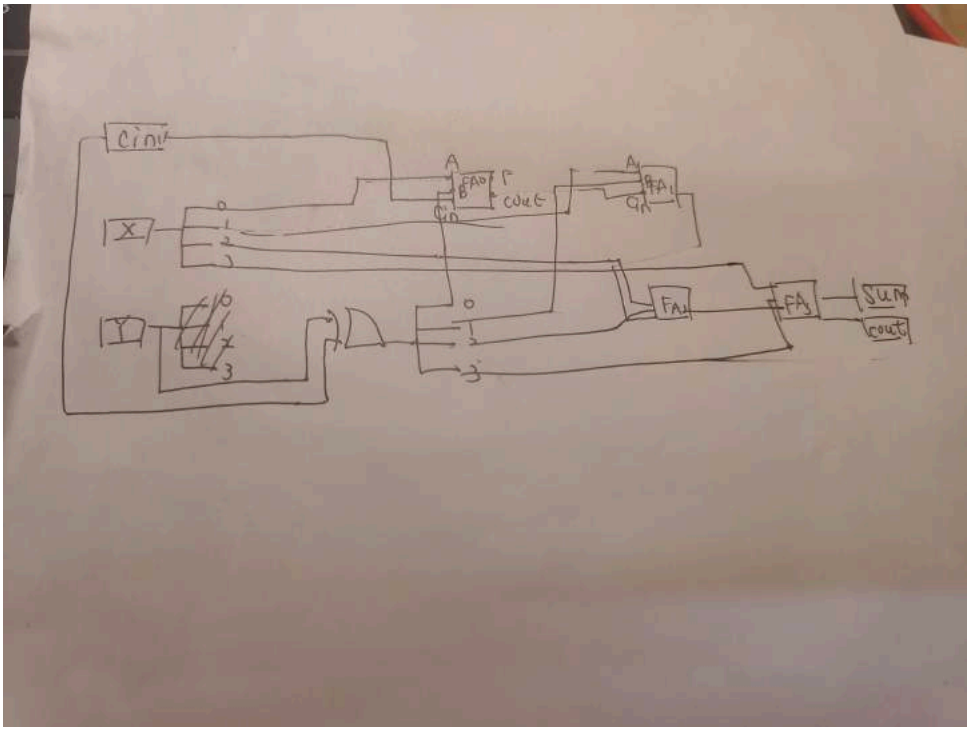

图 13: 加减法器原理图

电路图实现：

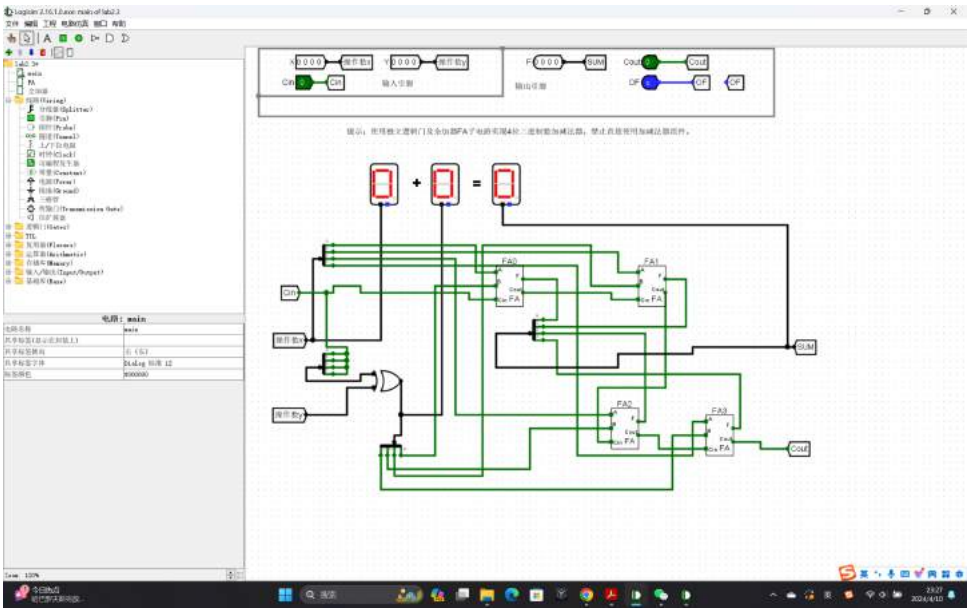

图 14: 加减法器电路图

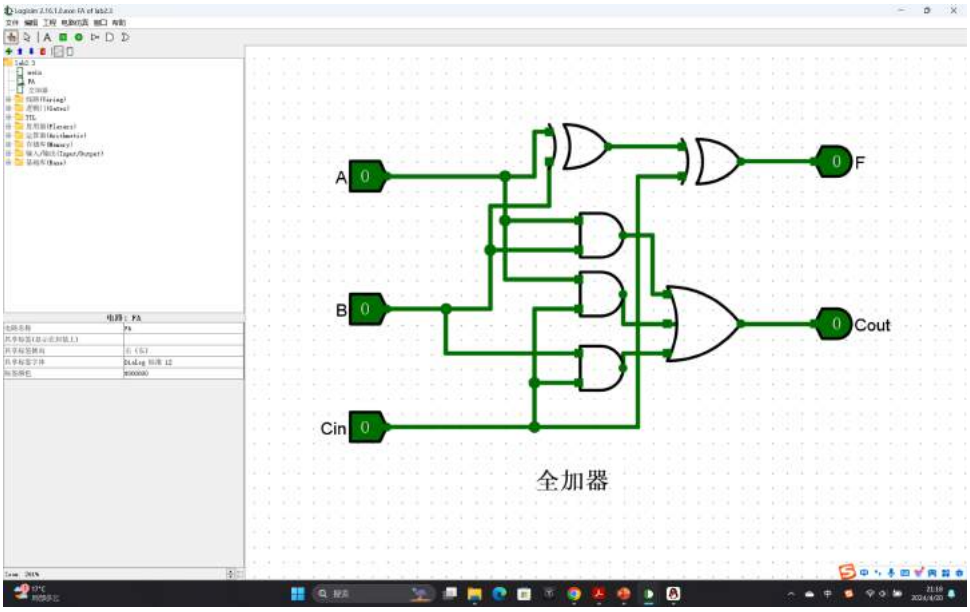

图 15: 全加器电路图

1.3.4 实验数据仿真测试图

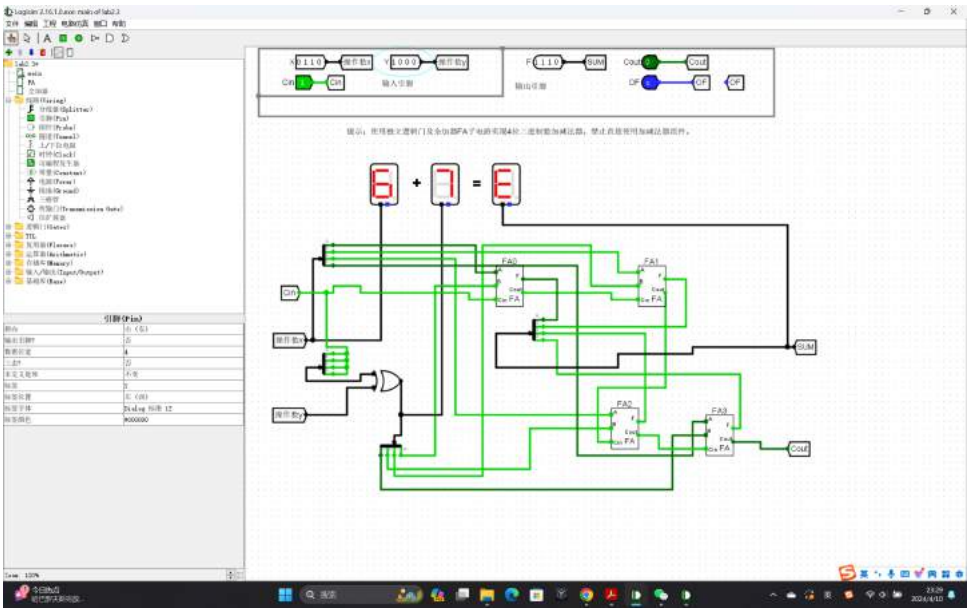

图 16: 加减法器仿真测试图 1

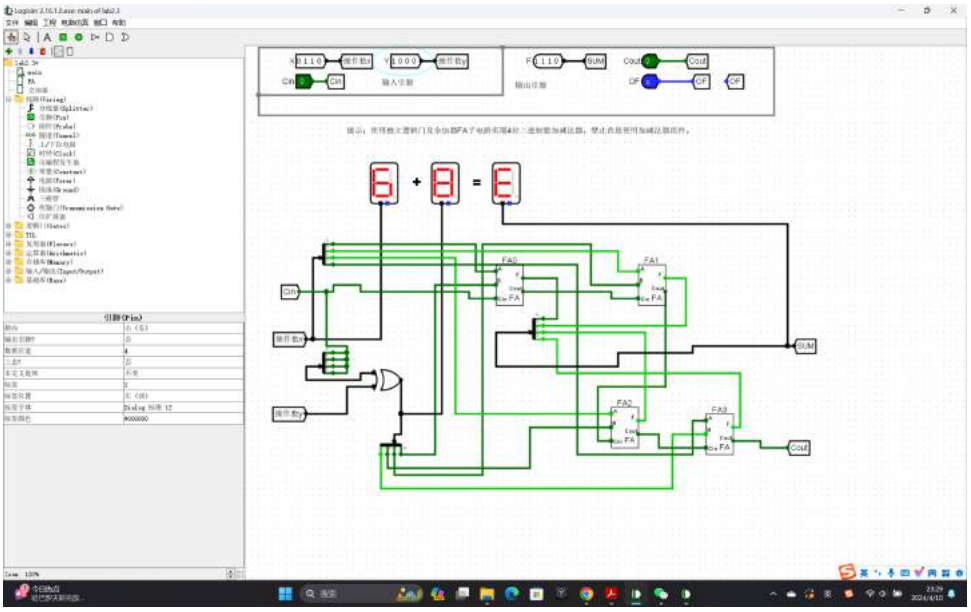

图 17: 加减法器仿真测试图 2

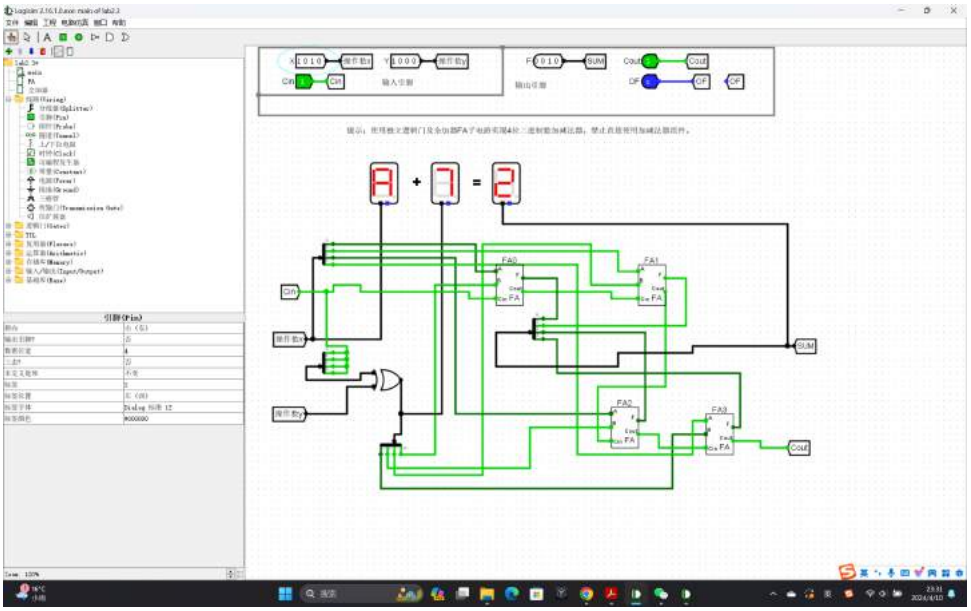

图 18: 加减法器仿真测试图 3

可以看出,减法时输出是正确的,但是数字时钟的显示不对,这主要是因为溢出的问题。结果上,加减法都正确,由 Cin 控制。本实验不需要真值表。符合 x+y,x-y 的结果,cin 控制加减即可。

1.3.5 错误现象及分析

在思考原理图期间,只用 B 和 cin 异或取反,但是没有加一,没有想明白 cin 的作用,用常量 0 连接了第一个全加器的 cin 输入口,后来修改后用 cin 连接得到正确结果。

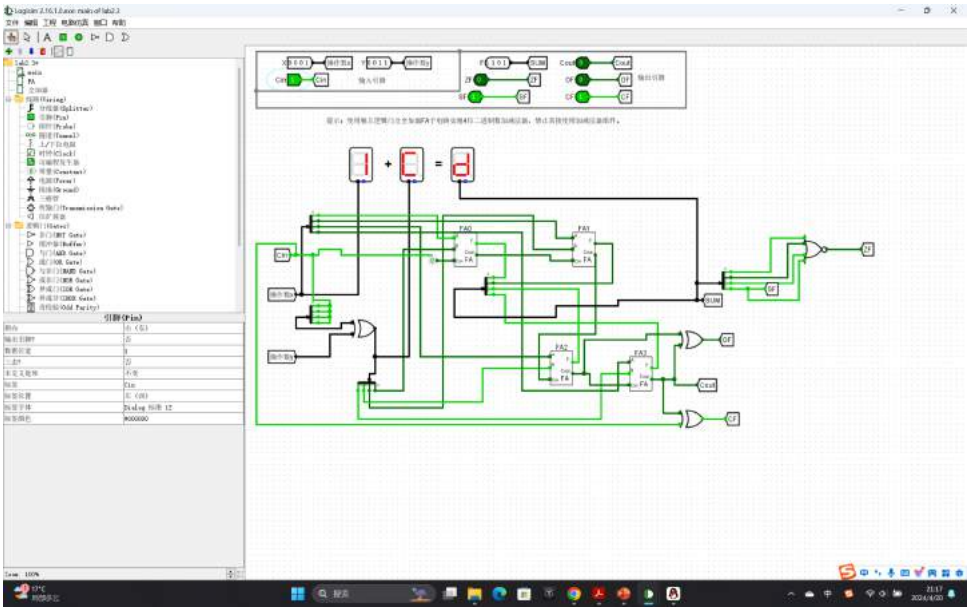

图 19: 错误

1.4 汉明码校验电路

1.4.1 实验内容

在 Logisim 的工作区中利用译码器、奇偶校验子电路、隧道和分线器等组件实现 7 位汉明码纠错功能。

1.4.2 实验整体方案设计

利用学习和思考和原理图解决问题。四位奇偶校验器用异或和同或即可。3-8 译码器同第一个实验。本实验不需要底层模块设计图。

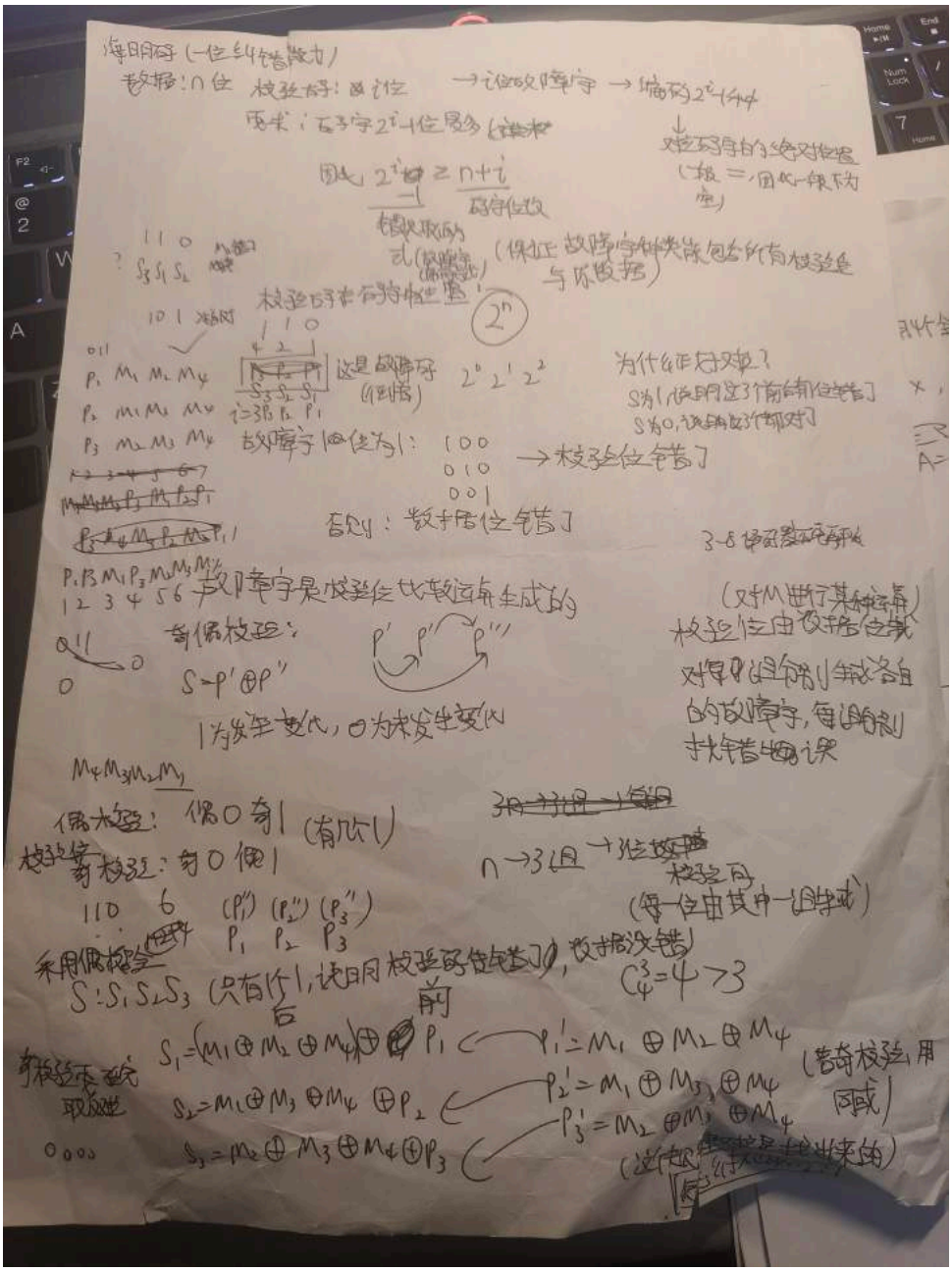

图 20: 思考 1

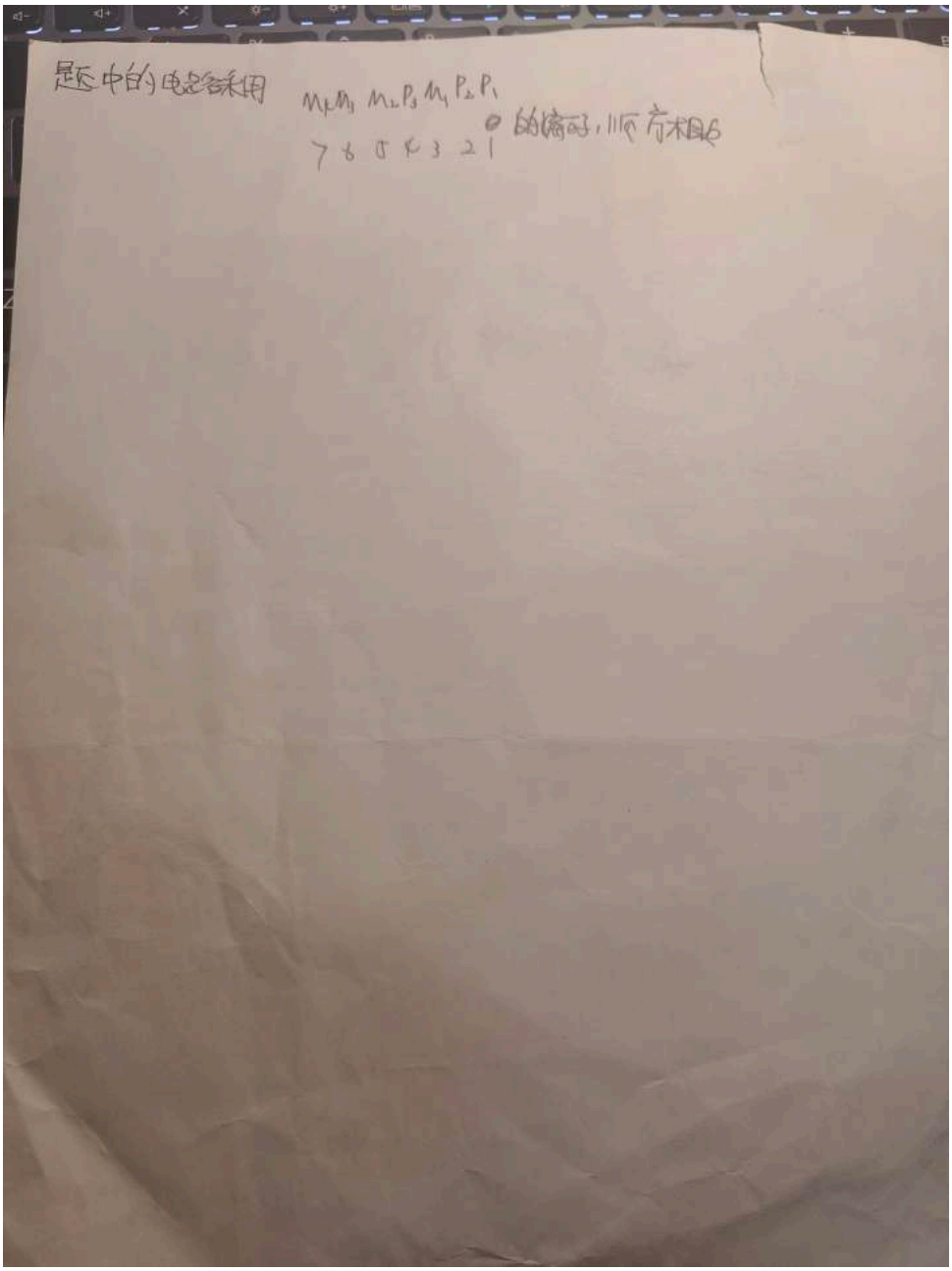

图 21: 思考 2

Input	输入数据
Noerror	没错误这个值为 1
output	输出错误的那一位

表 6: 汉明码引脚作用

1.4.3 实验原理图和电路图

原理图:电路图实现:

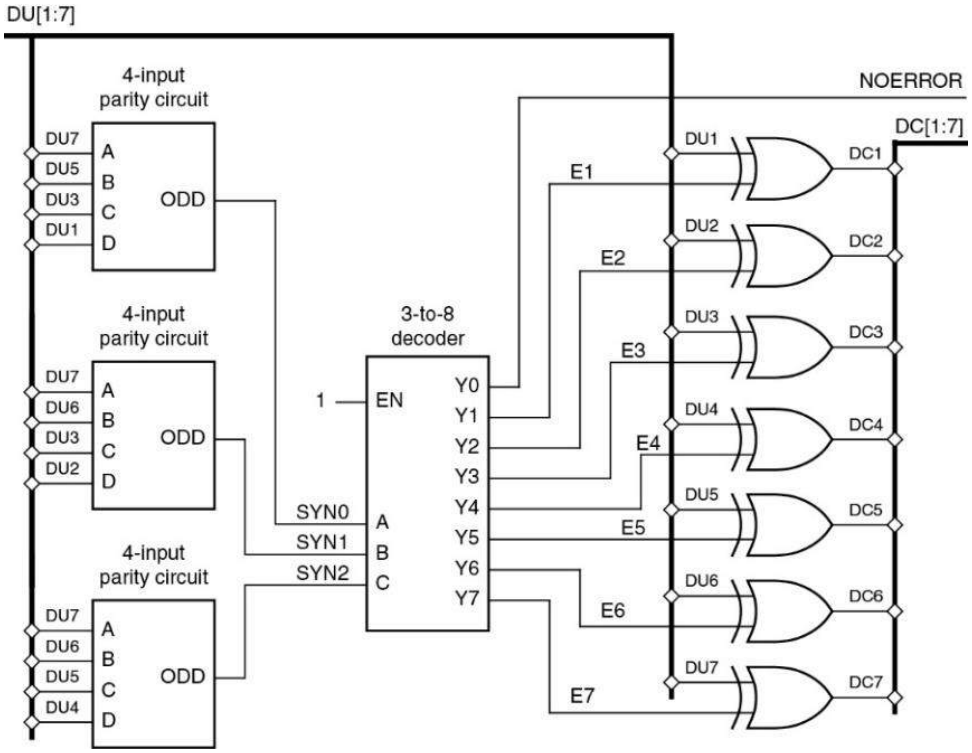

图 22: 汉明码原理图

1.4.4 实验数据仿真测试图

真值表是根据错的位数得出的。

1.4.5 错误现象及分析

在完成实验的过程中，遇到了和原理图完全一致，但是结果却不对的问题。经过分析，得出了问题原因：在 PDF 上给出的汉明码分析的编码和 oj 平台给出的编码是正好相反的，如果用 PDF 的讲解连线 Y0-Y7, 结果会正好相反！PDF 的编码是 p1p2m1p3m2m3m4, 而 oj 的编码是 m4m3m2p3m1p2m1! 建议以后学期的 PDF 或 oj 修改顺序。

1.5 8 位桶形移位器

1.5.1 实验内容

在 Logisim 的工作区中放置多路选择器分线器常量 0 等组件实现 8 位桶形移位器。

1.5.2 实验整体方案设计

基于一位的实现原理，思考如何连接成 8 位 MUX->MUX 只是基础的用于连接的工具，8 位移位器通过分别移 1 位，2 位，4 位，用 shamt 和 L/R 做使能端控制是否移动即可。L/R 和 A/L 是

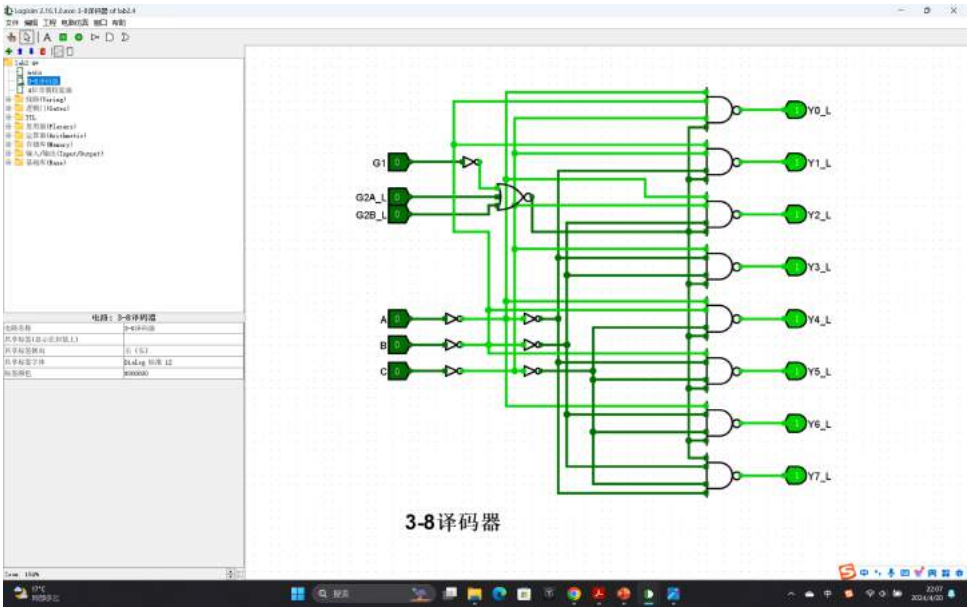

图 23: 汉明码电路图 1

MUX 的分路控制，移位用分线器解决，分为左移和右移，由于逻辑位的保留不同，还有循环左移，需要分别连移位器。

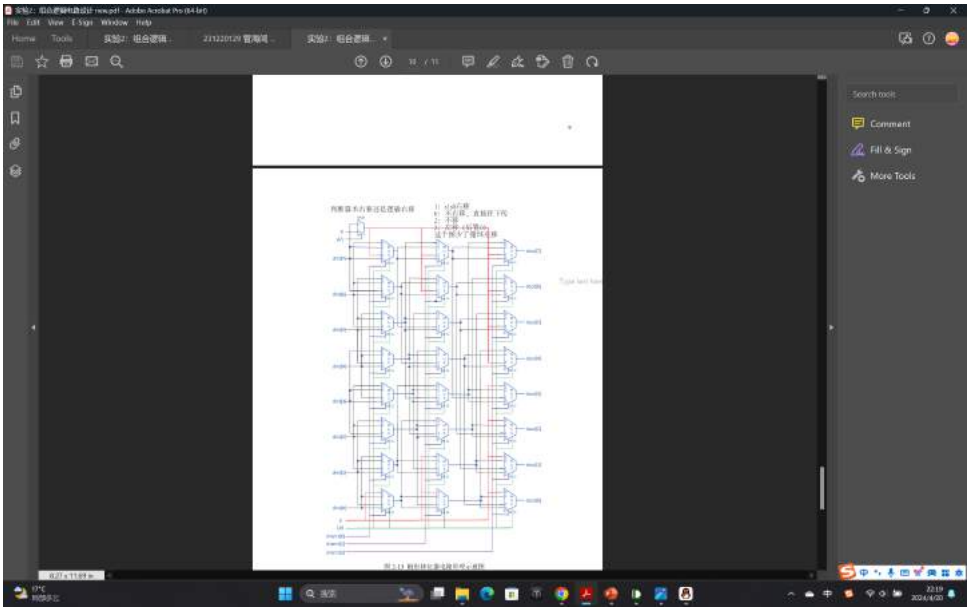

图 28: 参考

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 7: 8 位桶形移位器引脚作用

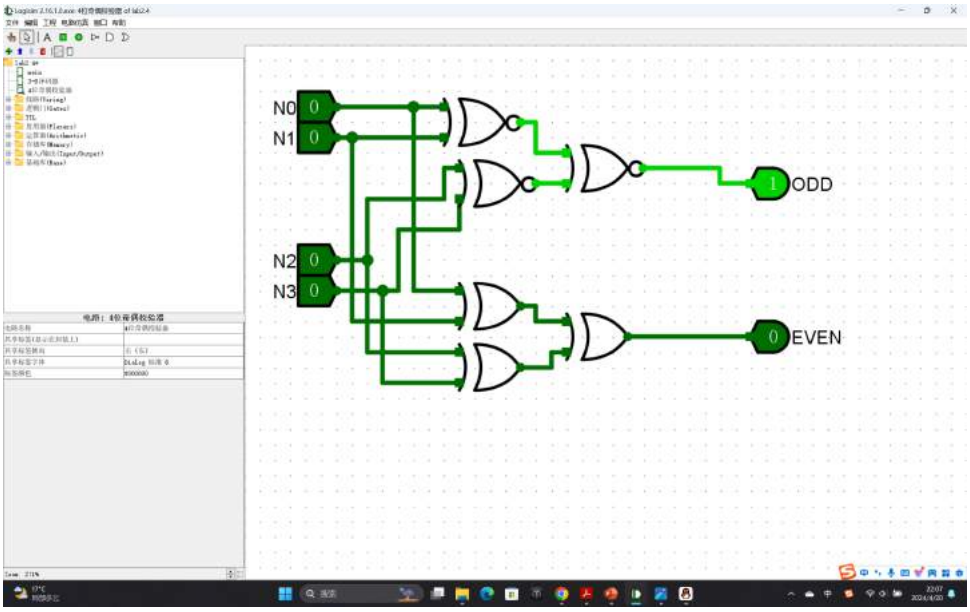

图 24: 汉明码电路图 2

1.5.3 实验原理图和电路图

原理图:

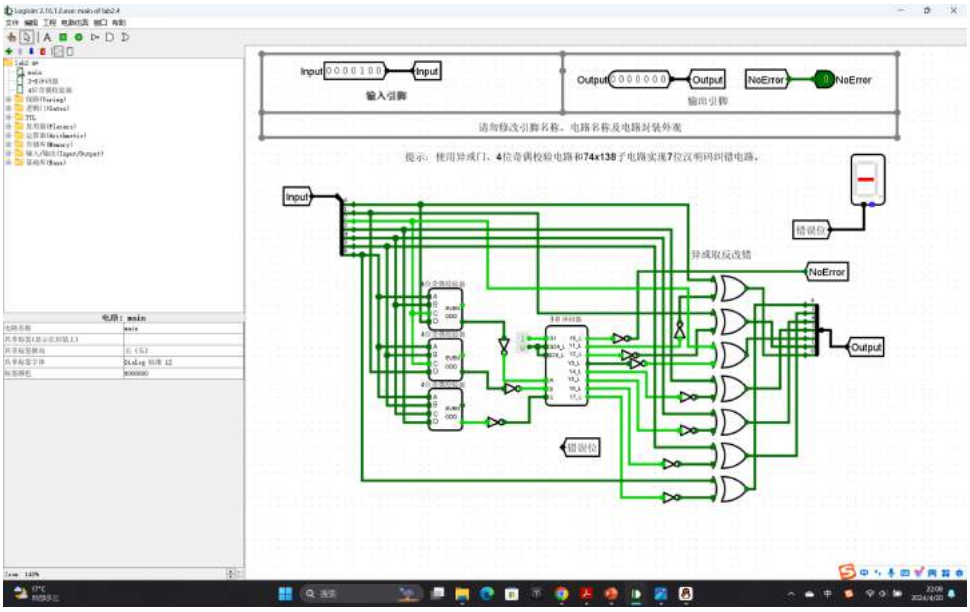

图 25: 汉明码电路图 3

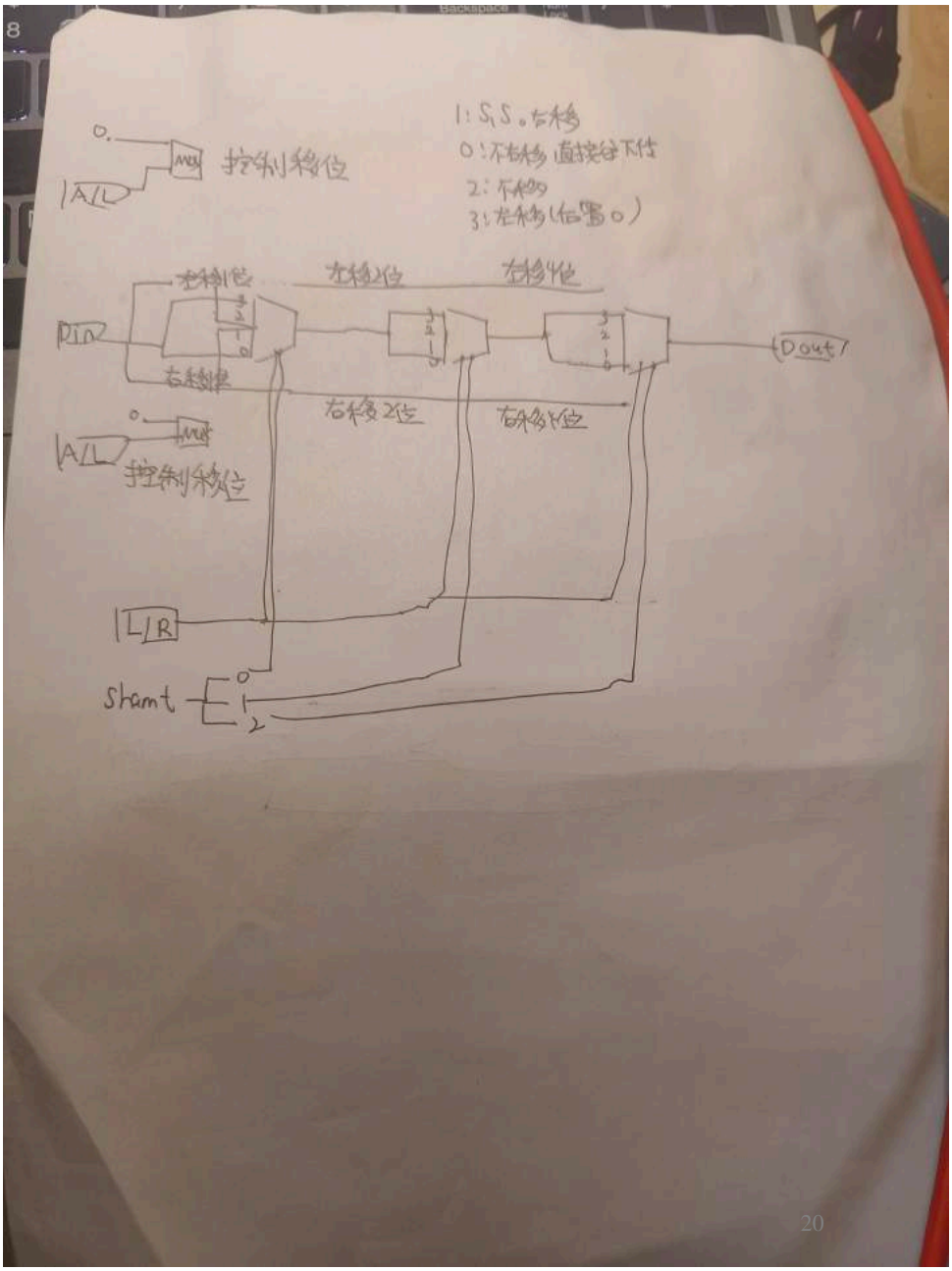

图 29: 原理图

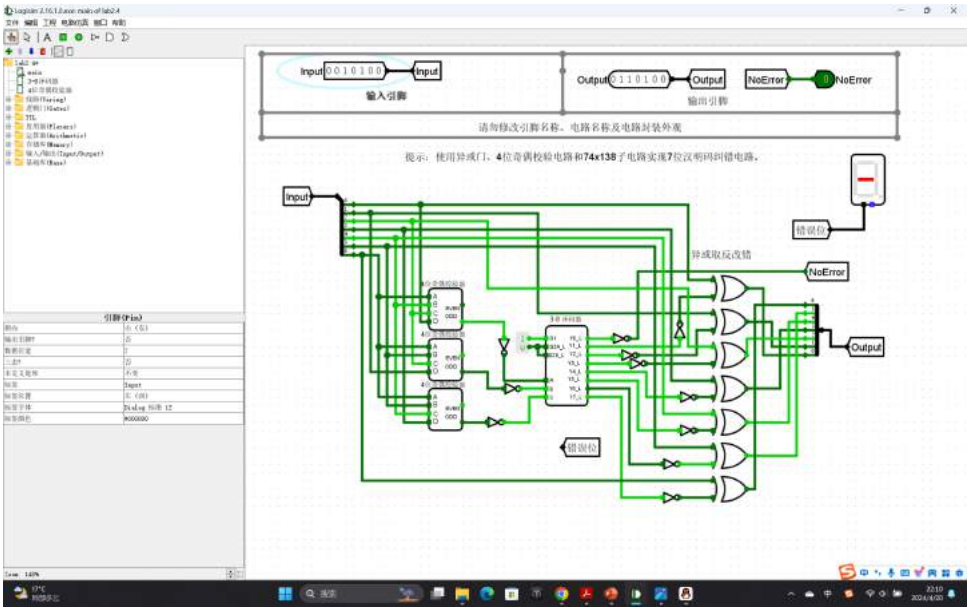

图 26: 汉明码仿真测试图 1

电路图实现：

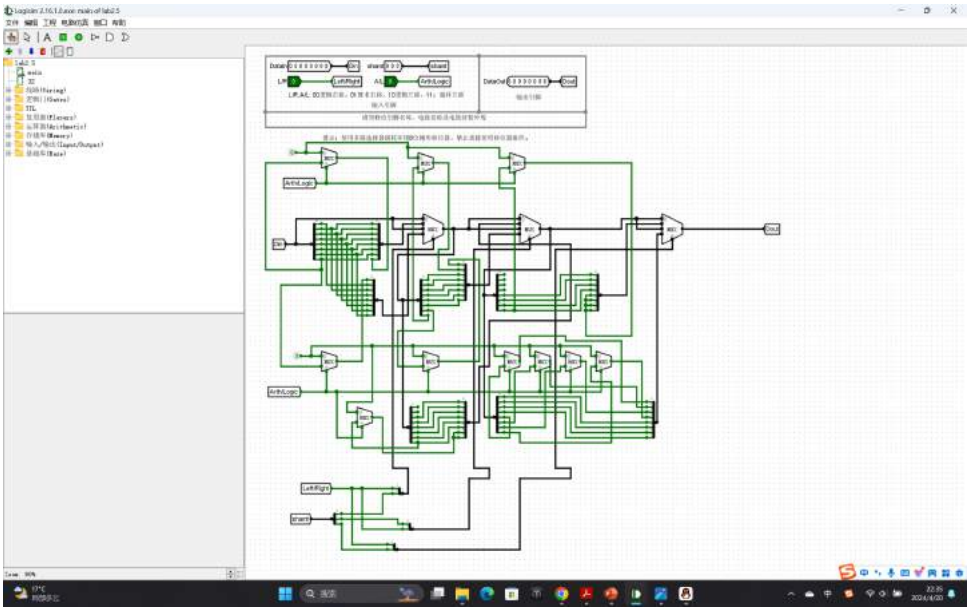

图 30: 电路图

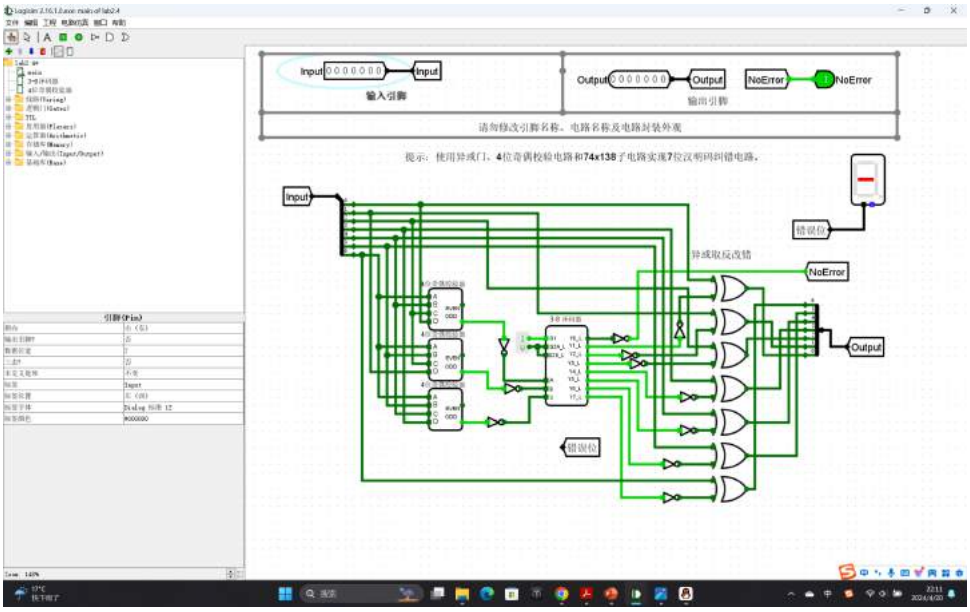

图 27: 汉明码仿真测试图 1

1.5.4 实验数据仿真测试图

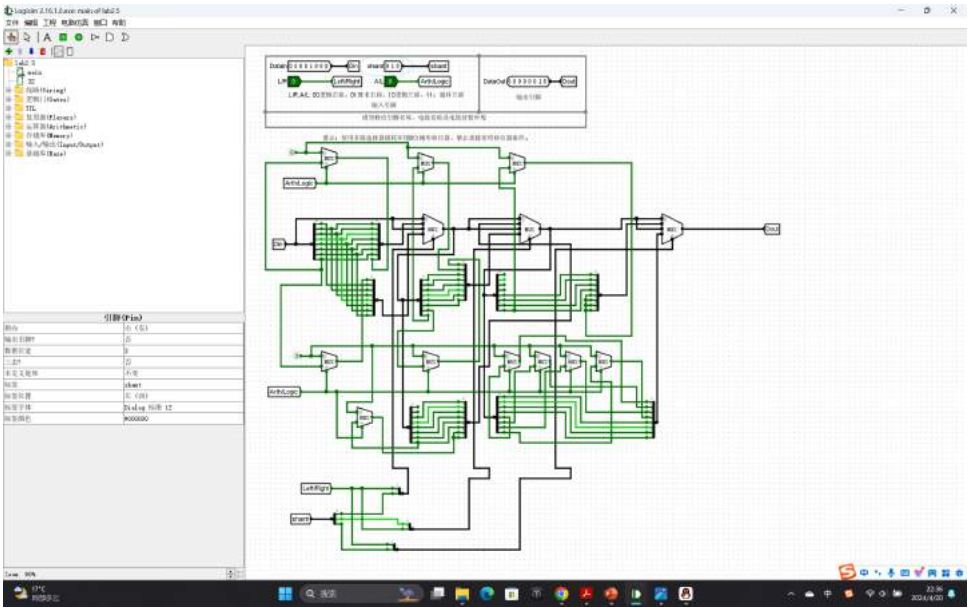

图 31: 仿真测试图 1

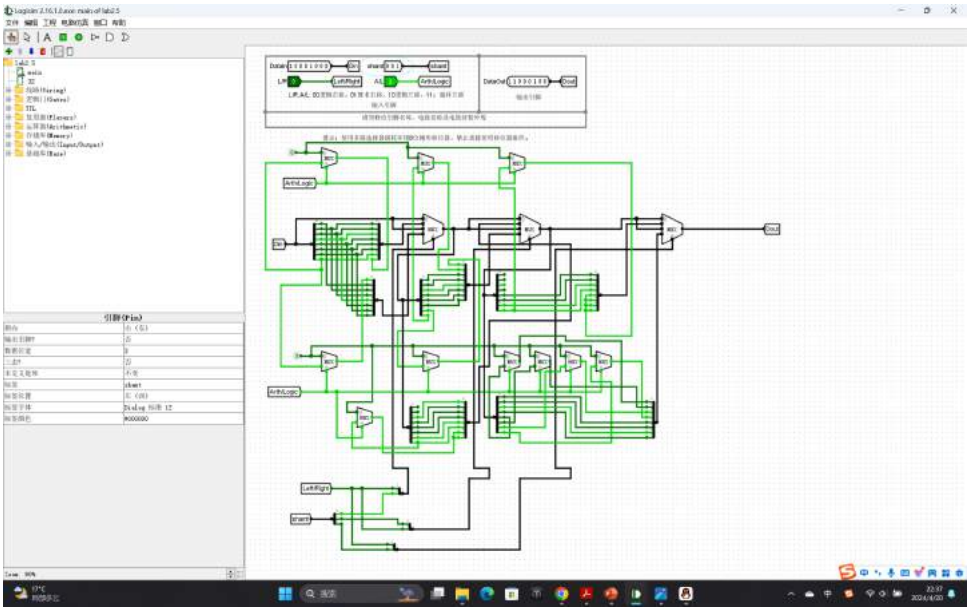

图 32: 仿真测试图 2

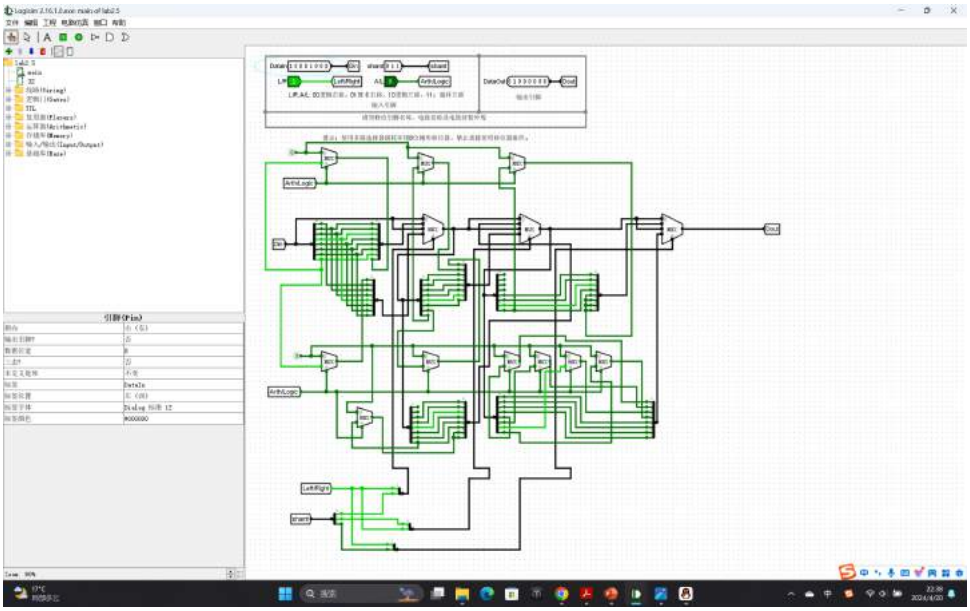

图 33: 仿真测试图 3

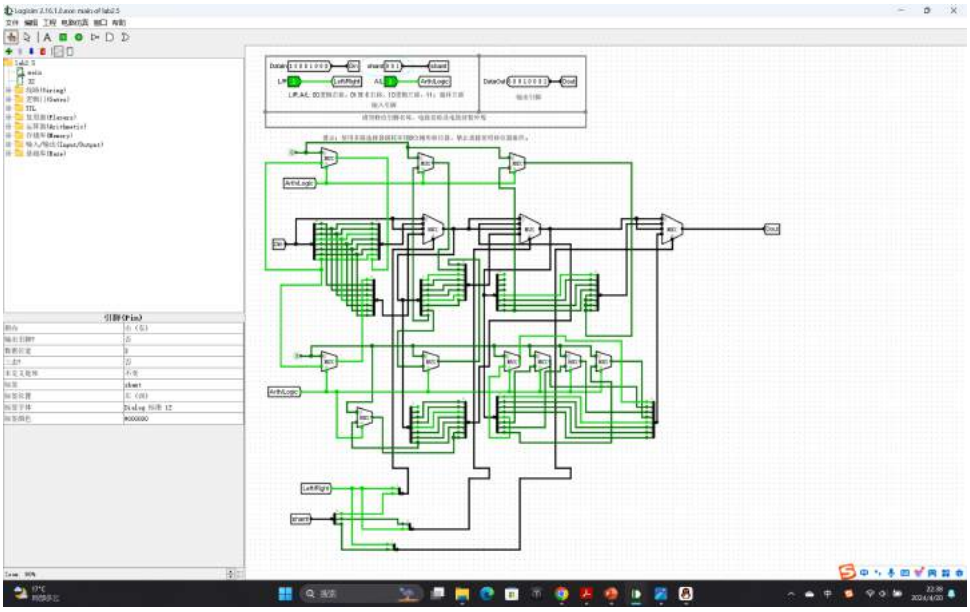

图 34: 仿真测试图 4

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. 修改实验中的加法器电路,生成进位标志 CF、溢出标志 OF、符号标志 SF 和结果为零标志位 ZF。

按照教材的四个标志原理 OF=cn 异或 cn+1,ZF=1 当且仅当 F=0,CF=Cout 异或 Cin,SF=Fn-1 直接连接即可。

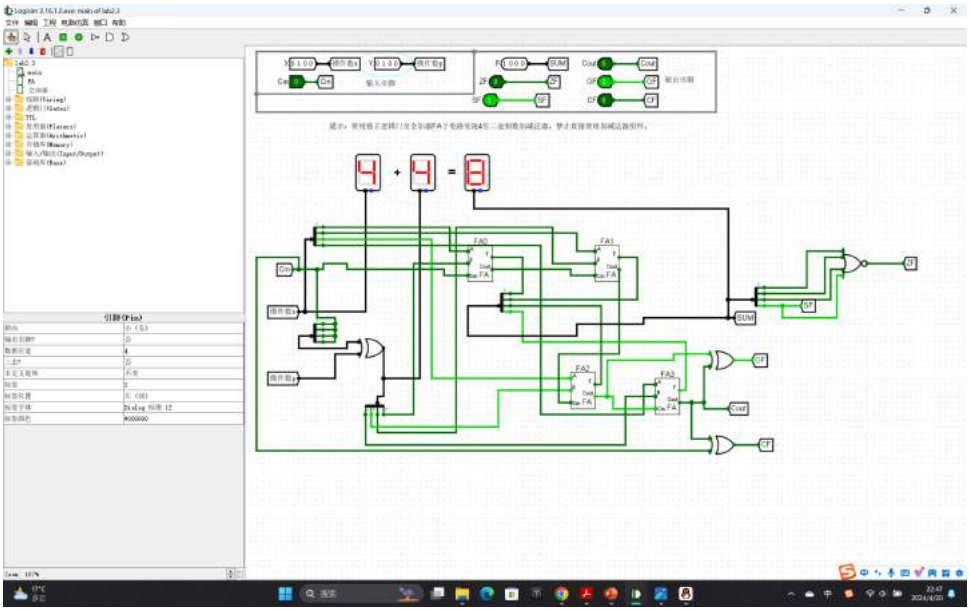

图 35: 思考题 1.1

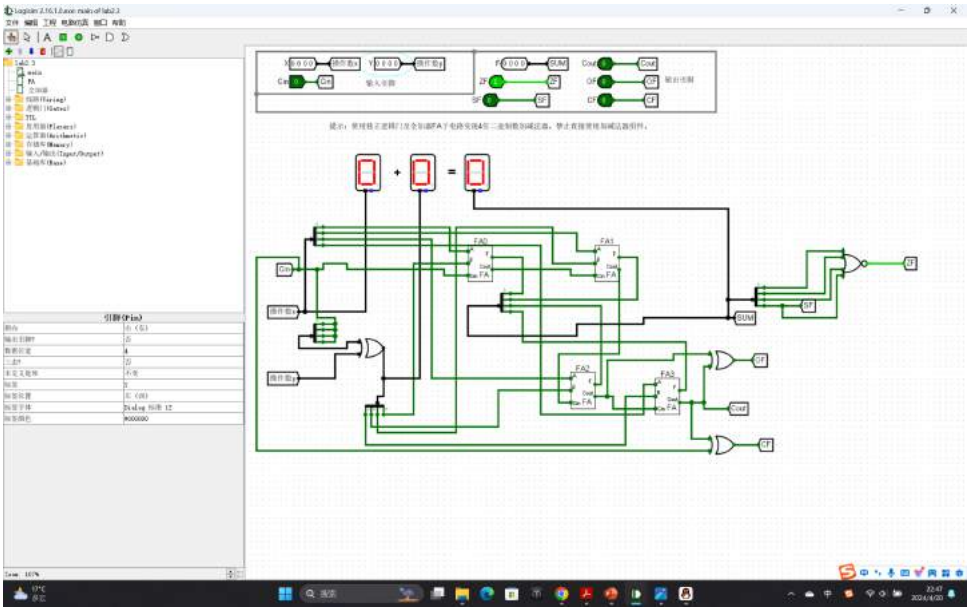

图 36: 思考题 1.2

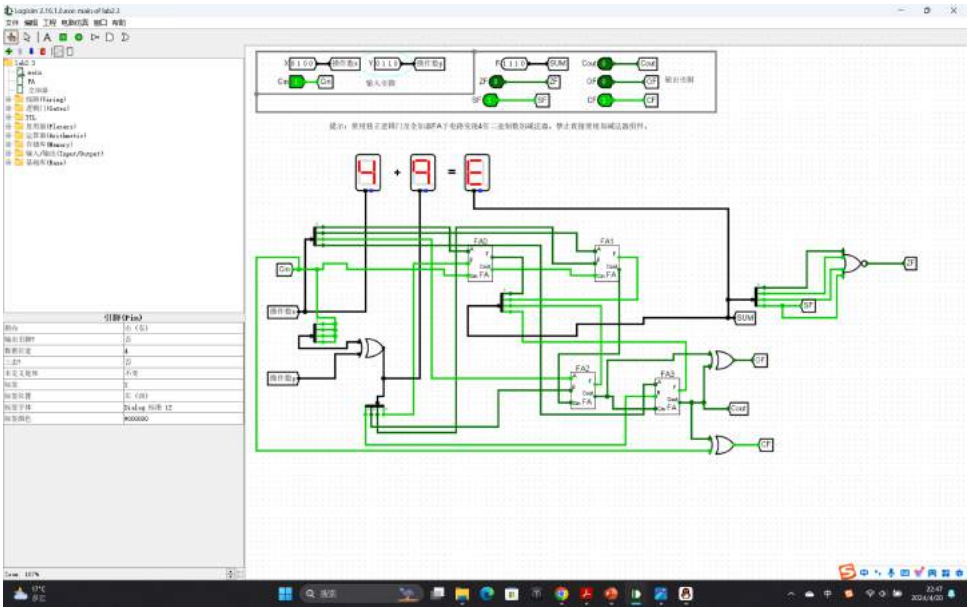

图 37: 思考题 1.3

2.2 思考题 2

2. 在执行比较指令时，通常使用减法运算后判断标志位的方式来实现，试通过上述加法器实验举例说明判别的方法。

如图，OF=SF 时说明 x 大于 y，反之小于。

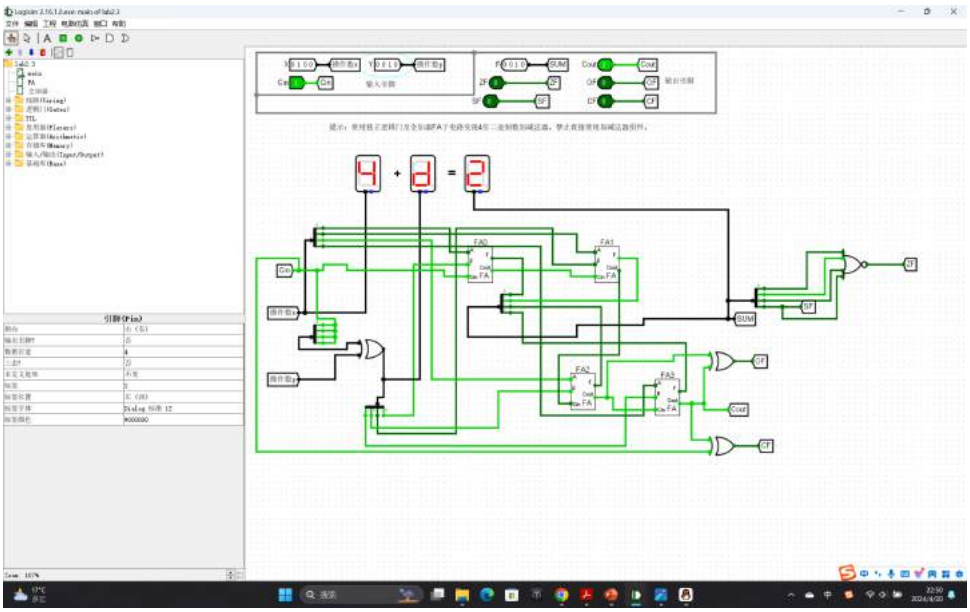

图 38: 思考题 2.1

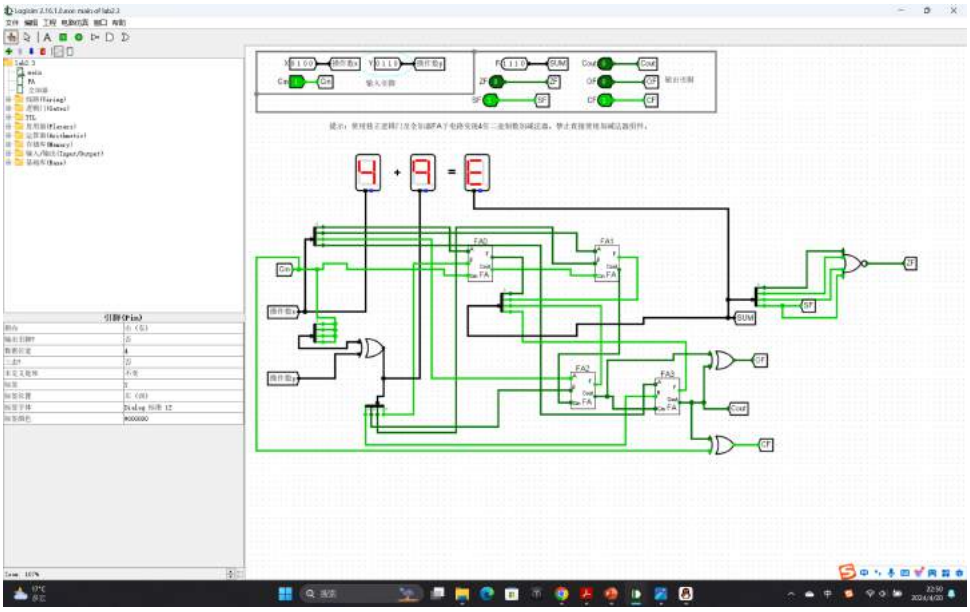

图 39: 思考题 2.2

2.3 思考题 3

3. 如何使用 8 位桶形移位器扩展到 32 位桶形移位器。

一个最简单的方法就是仿照 8 位进行五个 32 位 MUX 的扩展，但工作量极大(已给出电路)

还有一种理论可行的方案就是修改 8 位的电路，使其存储被移出的位数，用于传给下一个 8 位移位器作为输入，然后 4 个 8 位可以组成 32 位 8 位移位器，这个两个又能组成 32 位 16 位移位器，然后 2 个又能组成 32 位 32 位移位器。但是我没做出来

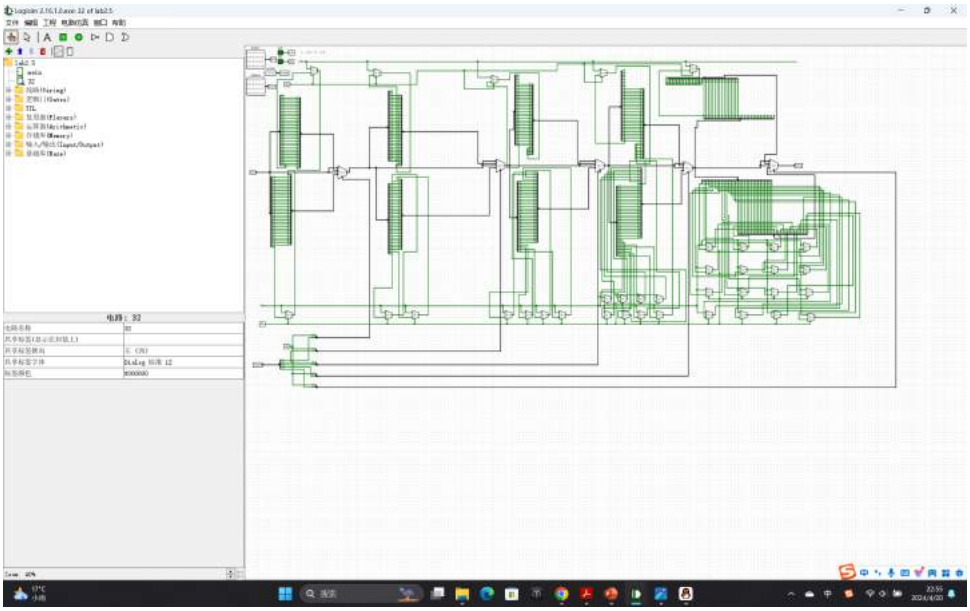

图 40: 思考题 3.1

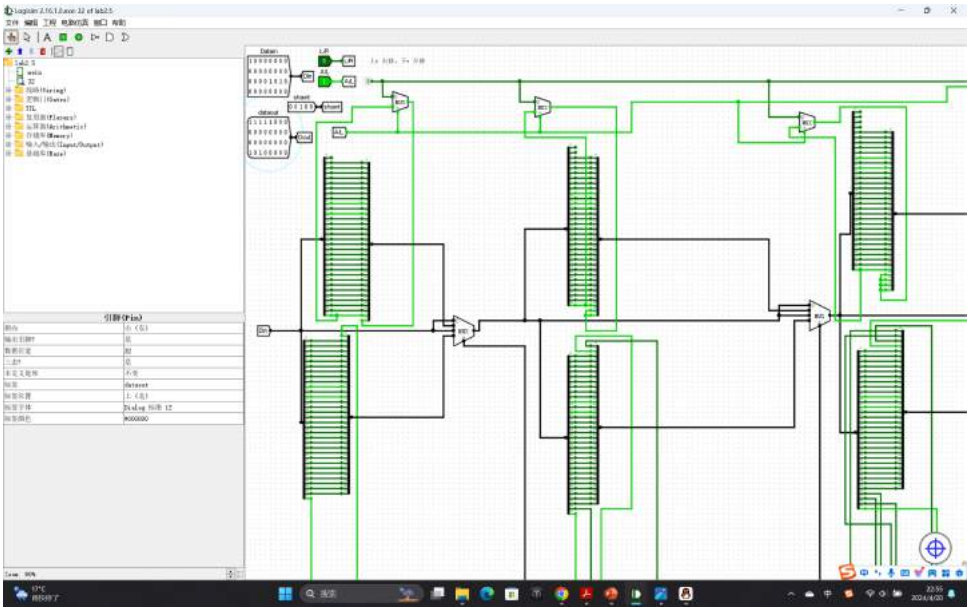

图 41: 思考题 3.2