



实验二

191098328 计算机科学与技术系 张世茂

191098328@smail.nju.edu.cn



2021-10-30

目录

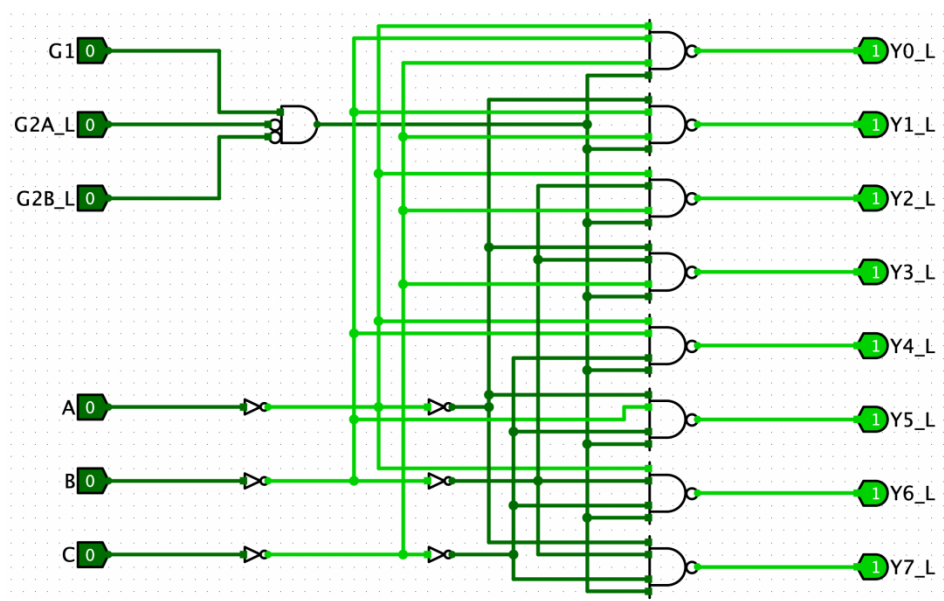
1. 3-8 译码器	2
1.1 实验操作流程	2
1.2 实验结果	2
1.3 实验错误与原因分析	5
2. 8-3 优先编码器	5
2.1 实验操作流程	5
2.2 实验结果	6
2.3 实验错误与原因分析	11
3. 4 位串行进位加法器	11
3.1 实验操作流程	11
3.2 实验结果	12
3.3 实验错误与原因分析	13
4. 1 位 ALU 电路	14
4.1 实验操作流程	14
4.2 实验结果	16
4.3 实验错误与原因分析	20
5. 思考题	21
6. 实验总结	22

1. 3-8 译码器

1.1 实验操作流程

设计 3-8 译码器，首先阅读实验手册上对于译码器功能的要求以及相关原理的介绍，可以得到相关的原理图。该电路需要设置 6 个输入端，其中包括 3 个控制端和 3 个数据输入端，以及 8 个输出端。

实验的电路原理图如下所示：



可以看到，8 个输出端都为低位有效，即编码结果对应的输出为低电位。当 G1 为低电位、G2A_L 或 G2B_L 为高电位时，译码电路不工作，所有的输出端都将表现出高电位；否则，电路正常工作，将三位输入端的编码转化为对应输出端的低位输出。该电路的译码功能是通过不同输出端所对应的逻辑与表达式实现的。

在实现电路后，对电路进行进一步的仿真检验。

1.2 实验结果

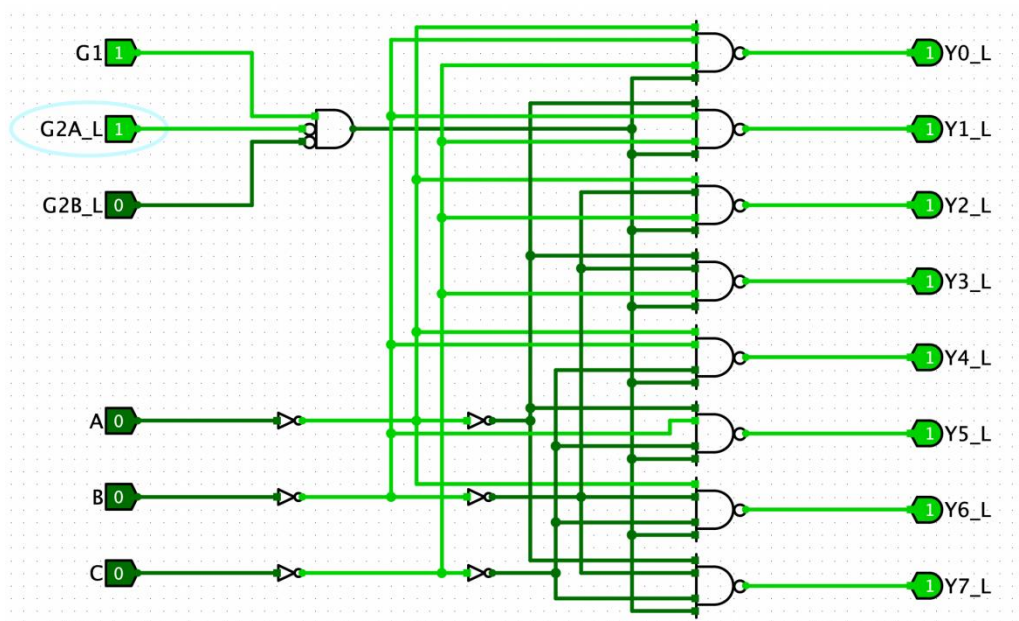
首先通过 Logisim 提供的功能生成所实现电路的真值表，观察是否有误。

得到如下的真值表：

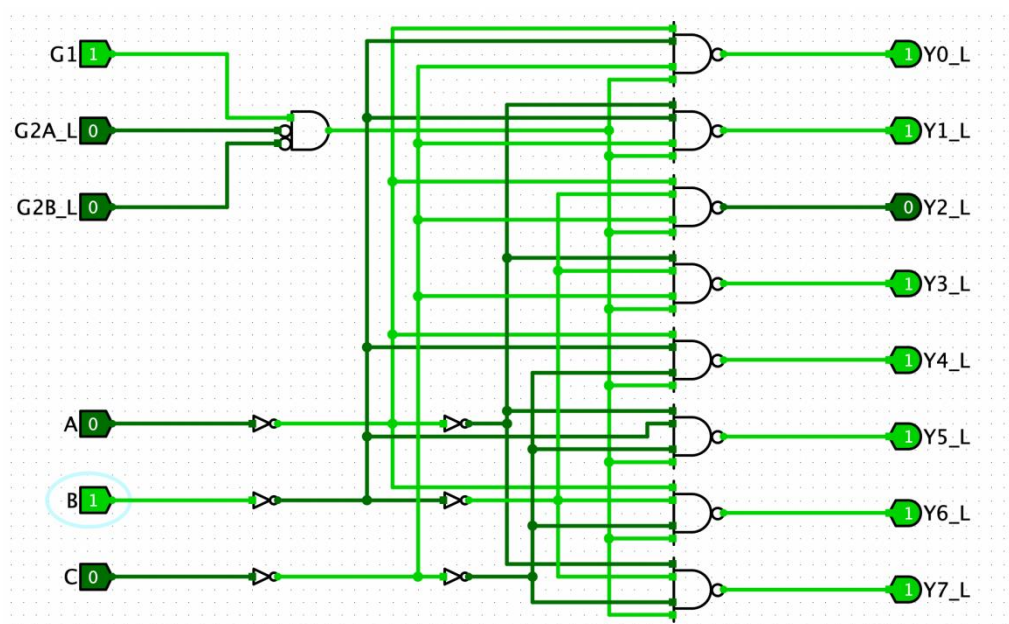
[illegible]

端的值都为全 1；在控制端有效时，数据输入端译码后结果对应的输出端为低电位，其它输出端皆为高电位。

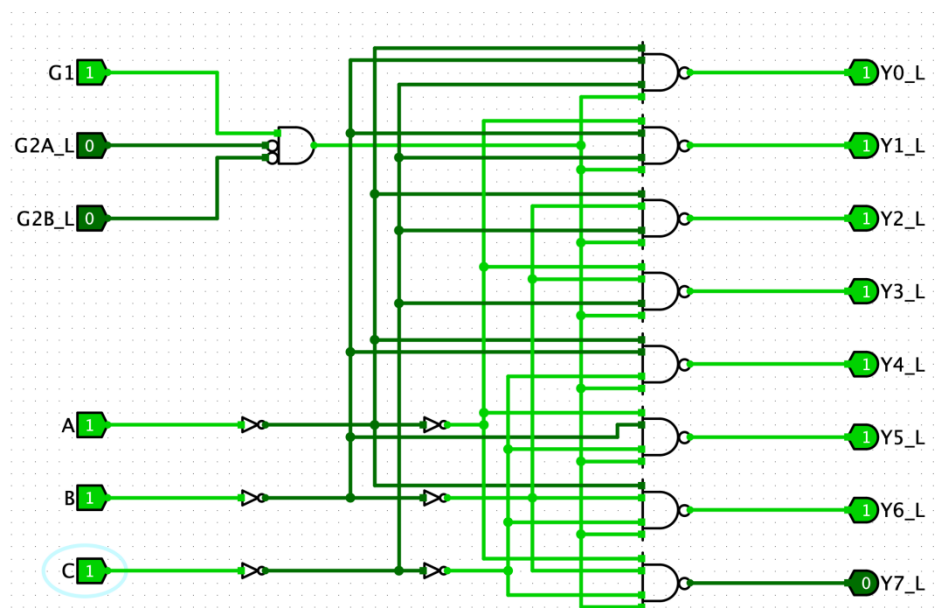
然后选取具有特征性的输入对电路进行仿真检验。



当 G1 和 G2B_L 有效，G2A_L 无效时，所有输出端显示为全 1。



当输入端全为有效时，将输入端编码置为 010，输出端对应的 Y2_L 输出 0，其余输出端为 1。



当控制端全为有效，数据输入端设为 111 时，输出端对应的 Y7_L 输出 0，其余输出端输出为 1。

可以发现，所设计的电路可以按照所预想的功能运行，一定程度上验证了所实现电路的正确性。

1.3 实验错误与原因分析

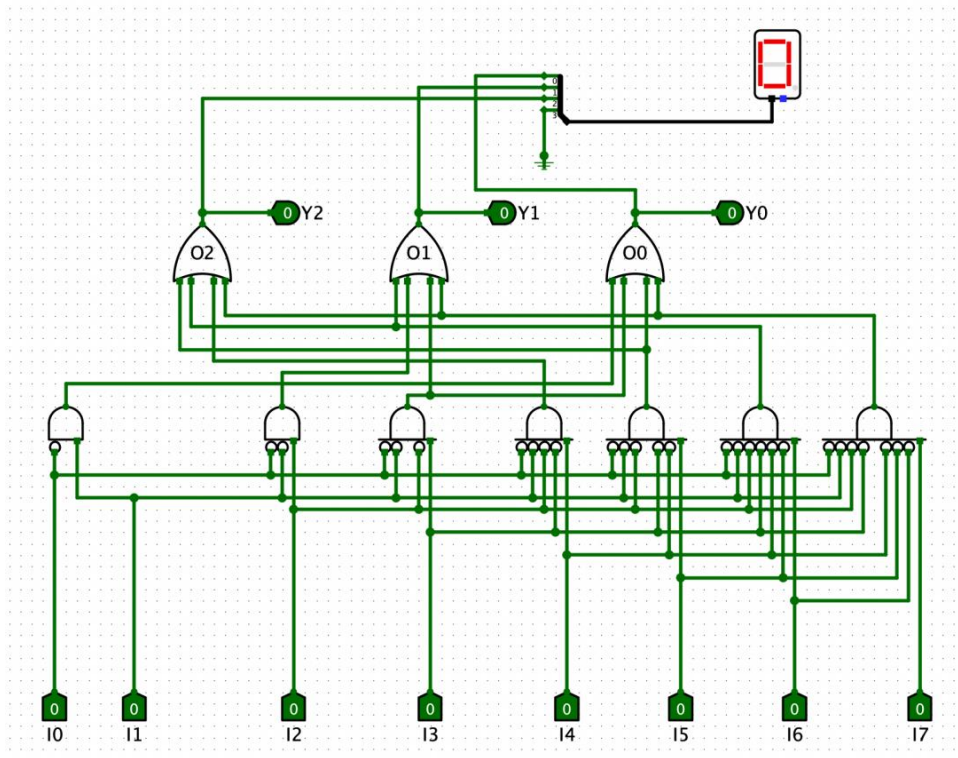
最初实现电路时由于忽略了实验手册上所表示的部分低位有效输入端，因此在验证实验电路时以为出现了实现上的错误，后来发现部分输入/输出端口为低位有效，应当留意电路上元件的命名特征，标注有“_L”的即为低位有效，在自己命名时也应当注意命名的规范。

2. 8-3 优先编码器

2.1 实验操作流程

要通过逻辑电路实现 8-3 优先编码器，需要设置 8 个输入端，根据优先级进行选择编码，并根据逻辑门实现组合逻辑。按照实验要求，最终通过 Logisim 中提供的十六进制数字显示元件进行输出显示，需要借助分线器将原先的多道输出结果送至数字显示器。

实验的电路原理如下图所示：



该电路的输入端中优先级排序为从左至右依次降低。由电路原理图可知，当高优先级输入端被置为有效后，后面低优先级的输入端对应的与门输出都为 0。O0、O1、O2 三个或门分别对应的是输出编码的从低到高三位，其输入即为该位为 1 时对应的输入对应与门的输出。

输出的编码共有三位，要通过十六进制数字显示元件显示出来，就需要通过分线器来连接。将 O0、O1、O2 的输出分别连接至分线器第 0、1、2 位上，由于分线器多出一位，因此将多出的一位最高位接地，将右端连接至十六进制数字显示元件。

为便于生成真值表进行检验，又在电路中加入三个输出端，分别对应输出编码的每一位的值。

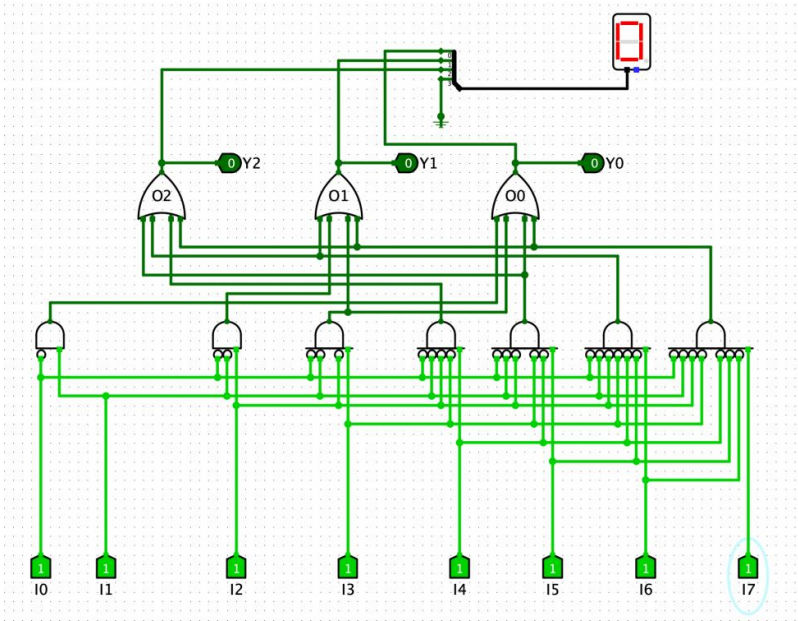
实现电路后，对电路进行进一步的观察和检验。

2.2 实验结果

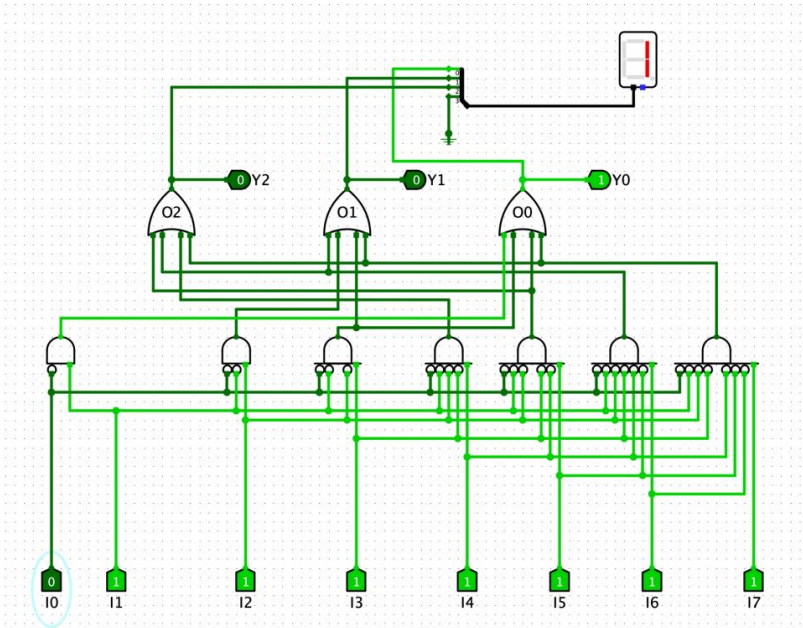
通过 Logisim 生成该电路的真值表如下：

1	1	1	0	1	1	1	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	1	1	0	0	0	1	0	0
1	1	1	1	0	0	1	0	0	0
1	1	1	1	0	0	1	1	0	0
1	1	1	1	0	1	0	0	0	0
1	1	1	1	0	1	0	1	0	0
1	1	1	1	0	1	1	0	0	0
1	1	1	1	0	1	1	1	0	0
1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	1	0	0
1	1	1	1	1	0	1	0	0	0
1	1	1	1	1	0	1	1	0	0
1	1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	0	1	0	0
1	1	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1	0	0

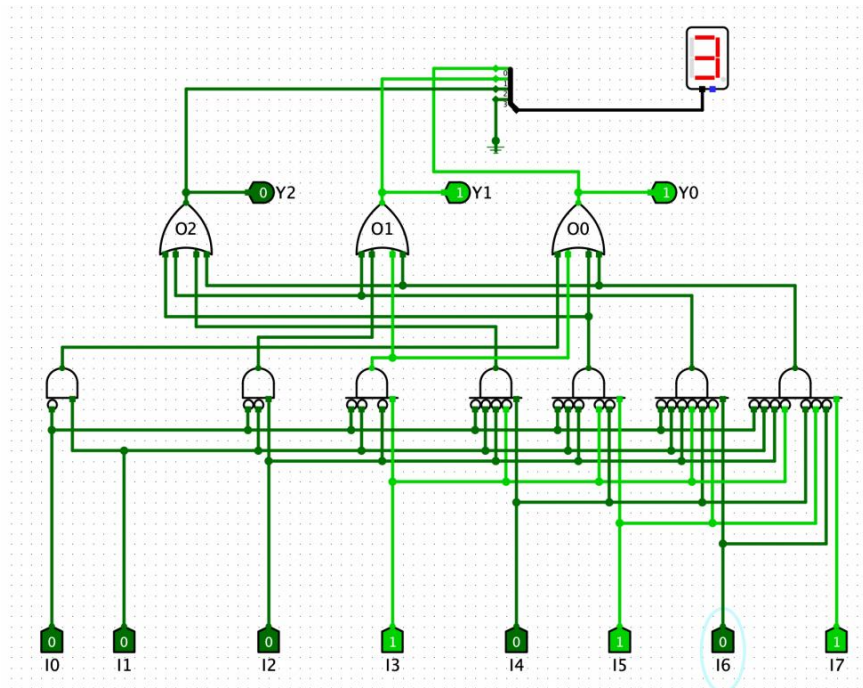
接着选取合适的输入对电路进行仿真检验。



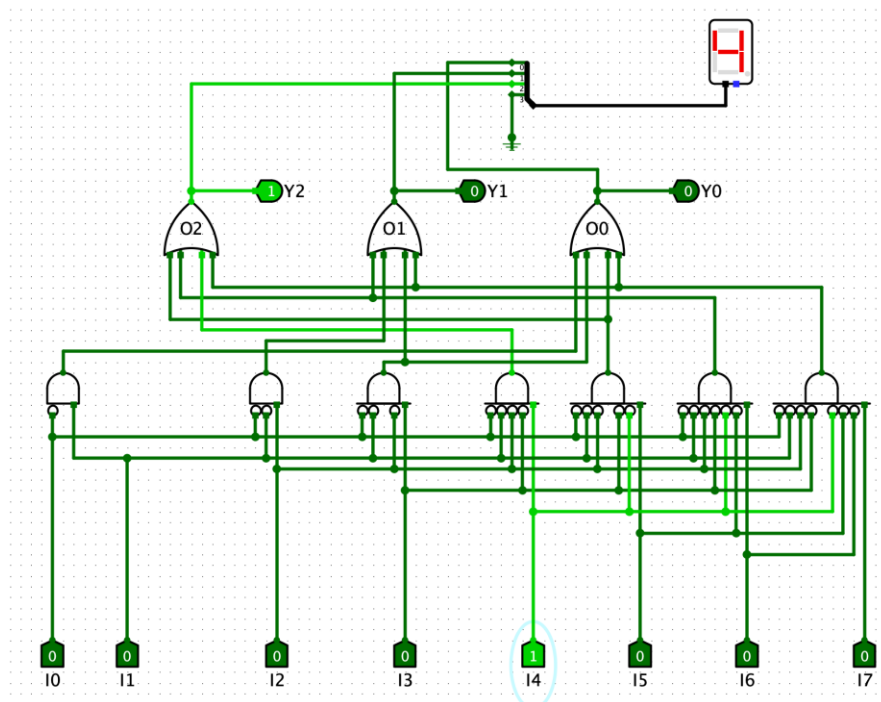
当所有的输入端全部置为有效时，输出端输出的是 0 对应的二进制表示，数码管上显示为 0。



当除 0 外所有输入端都置为有效时，输出端输出是 1 对应的二进制表示，数码管上显示为 1。



当 I3、I5、I7 输入端置为有效，其他输入端置为无效时，输出端输出的是 3 对应的二进制表示，数码管上显示为 3。



当只有 I4 输入端被置为有效时，输出端输出的是 4 对应的二进制表示编码，数码管上显示为 4。

通过上述的仿真检验，可以在一定程度上验证所实现电路的正确性。

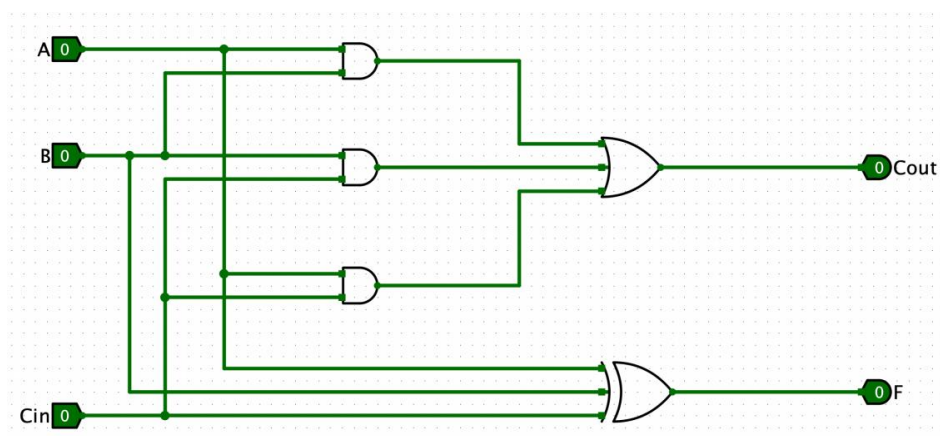
2.3 实验错误与原因分析

该实验主要的难点在于搜索学习并熟悉分线器在电路中的使用，在实验进行过程中并未遇到什么特殊的错误。

3. 4 位串行进位加法器

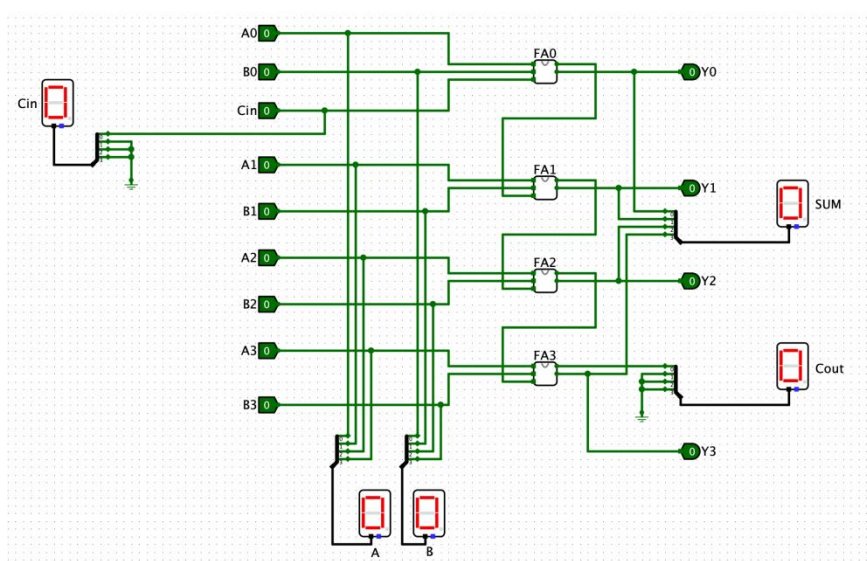
3.1 实验操作流程

要构建一个 4 位串行进位加法器，按照实验要求，首先要构造出一个全加器（FA）作为子电路，具体的电路原理图如下：



其中 A、B 为加数，Cin 为从下面传来的进位位，F 为输出的相加结果，Cout 为向上的进位位。当 A、B、Cin 中至少有两个为 1 时，向上进位位 Cout 为 1；当 A、B、Cin 中有奇数个 1 时，最终的和的结果为 1。

然后在已有的全加器基础上构建 4 位串行进位加法器，具体的电路原理图如下：



可以看到，为两个 4 位的加数输入各设置了 4 个输入端，另外还有一位 Cin 输入端作为低位的进位。该组合电路使用了 4 个全加器子电路进行实现，将加数输入、低位进位输入、和输出、向上进位输出都通过十六进制数码管显示出来。为了便于生成真值表进行观察，为 4 位输入的每一位都分配了一个输出引脚。

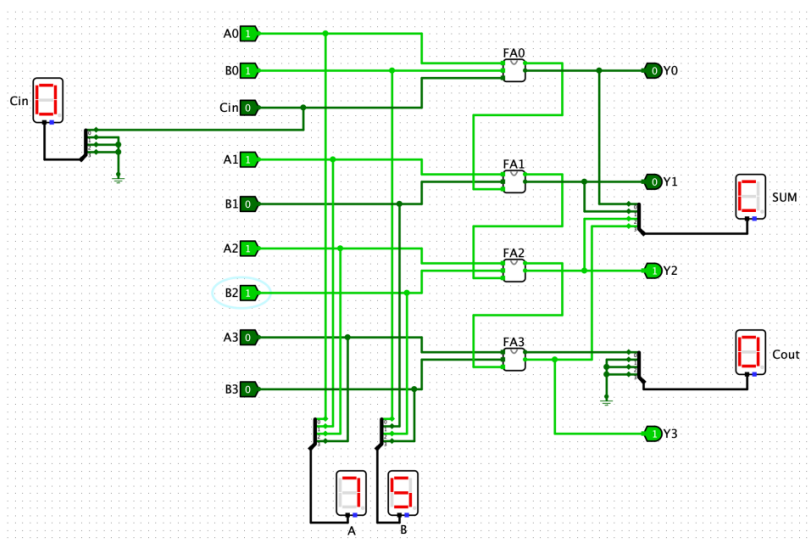
在电路实现完成后查看实验结果，对电路进行进一步的观察与检验。

3.2 实验结果

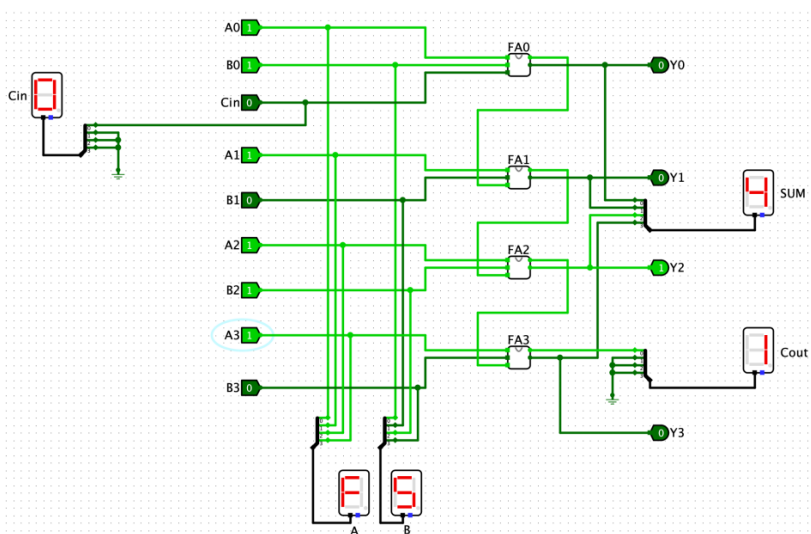
通过 Logisim 可以自动生成 4 位串行进位加法器的真值表，由于该电路共有 9 个输入端，因此真值表过长，不再在本实验报告中展示。

经过挑选样本进行验证，真值表结果与预期相符。

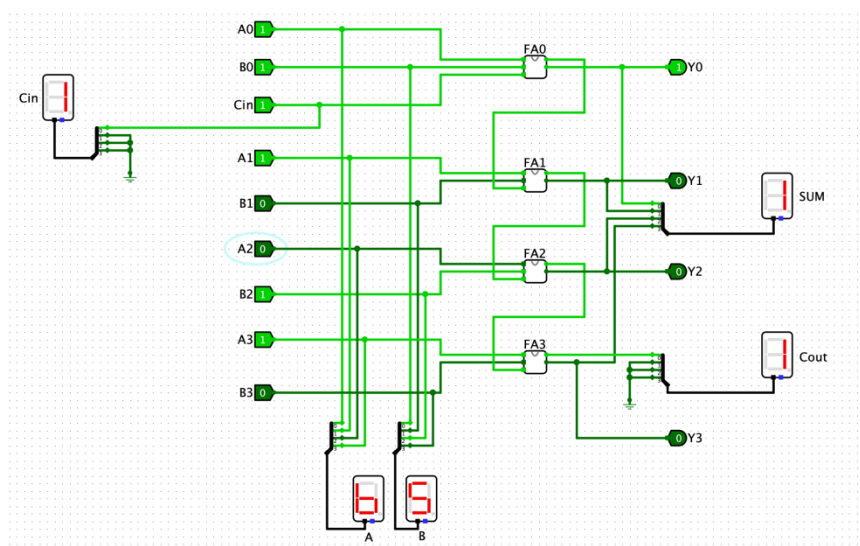
然后挑取合适的输入样例对所实现的电路进行检验。



当 $A=0x7$, $B=0x5$, $Cin=0x0$ 时，算出的和为 $0xC$ (12)，向上进位 Cout 为 0。



当 $A=0xF$, $B=0x5$, $Cin=0x0$ 时，算出的和为 $0x4$ ，向上进位 Cout 为 1。

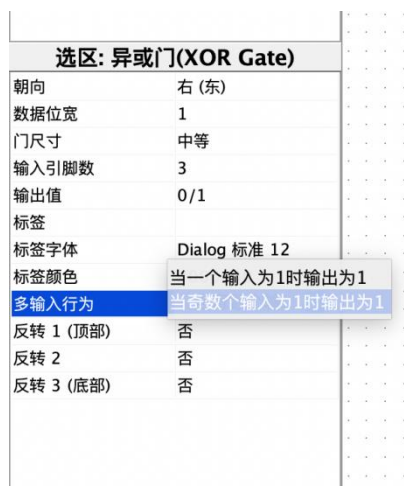


当 $A=0xb$ (11)， $B=0x5$ ， $Cin=1$ 时，算出的和为 $0x1$ ，向上进位 $Cout$ 为 1。

可以看到，任意挑选样例进行仿真检验所得出的结果都符合预期，可以在一定程度上验证所实现电路的正确性。

3.3 实验错误与原因分析

在进行该实验过程中，最初设计的 4 位串行进位加法器电路中 $Cout$ 输出一总是不对，但线路连接又没有问题，因此考虑是元器件或线路的功能出现了错误。后来发现在 Logisim 中，多输入异或门的功能需要进行特殊设置，如下图所示：



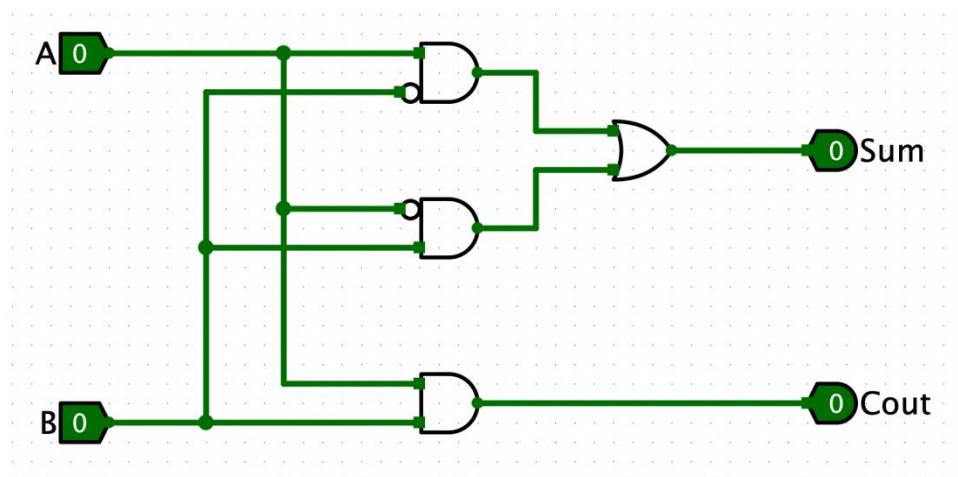
Logisim 中默认选择的是第一项“当一个输入为 1 时输出为 1”，因此会出现错误，应该手动改成“当奇数个输入为 1 时输出为 1”。然后再次验证，发现电路可以正常工作返回正确的结果。

4. 1 位 ALU 电路

4.1 实验操作流程

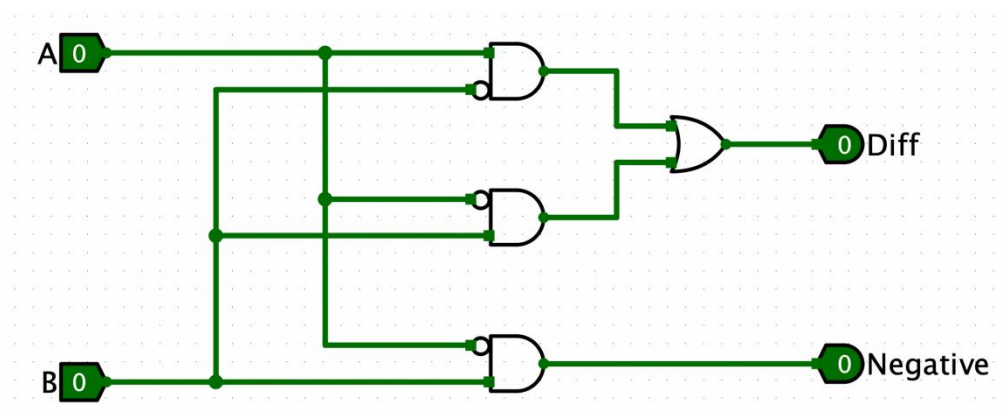
要实现一个 1 位的 ALU 组合电路（不考虑来自低位的进位或借位），根据实验手册，首先应当对子电路进行实现。实验中需要实现的子电路有：全加器电路、全减器电路、8-1 多路选择器电路。

实现的全加器电路原理图如下图所示：



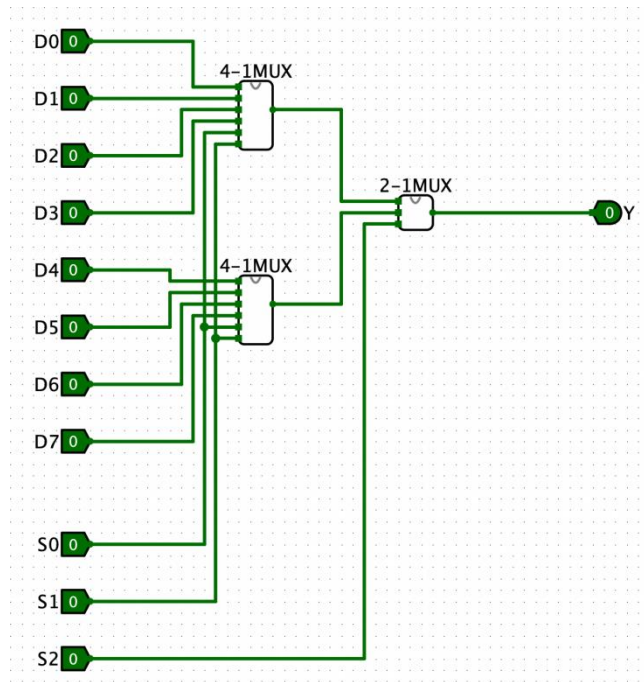
由于不考虑来自低位的进位，因此通过简单的与门和或门逻辑就可以实现。

实现的全减器电路原理图如下图所示：



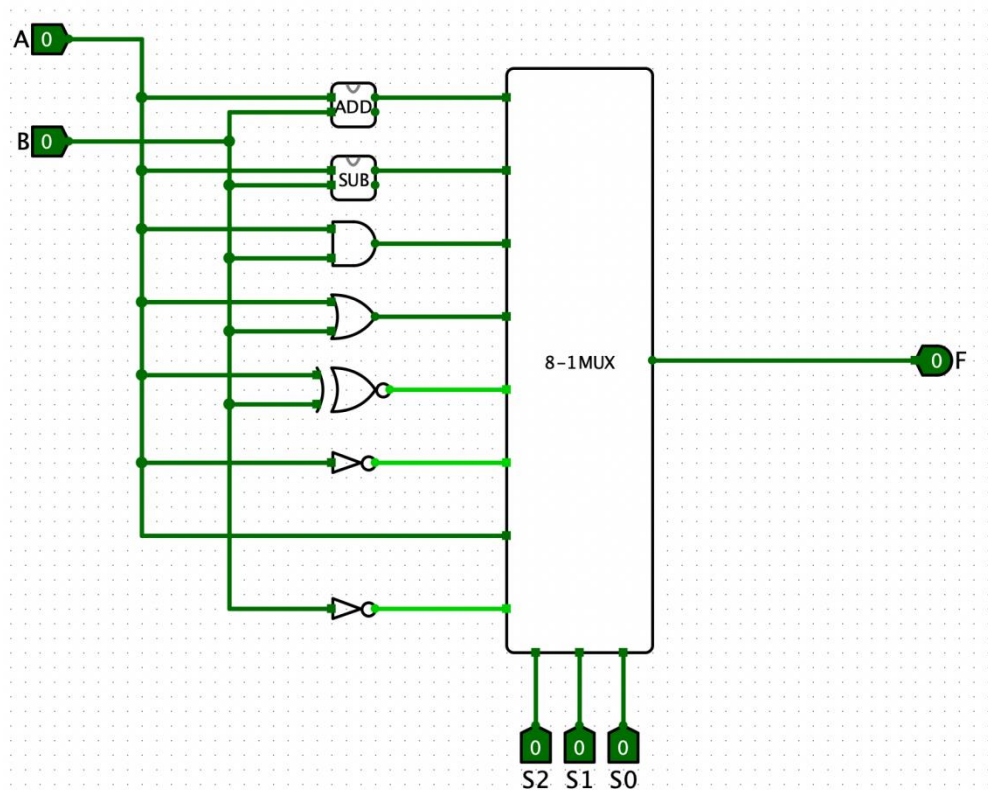
由于不考虑来自低位的借位，当且仅当 A、B 一个为 1，一个为 0 时差 Diff 输出为 1，当且仅当 A=0，B=1 时负值判断位/向上借位位 Negative 为 1。

构建 8-1 多路选择器电路时，通过首先构建 2-1 多路选择器子电路和 4-1 多路选择器子电路，然后再组合的方式进行实现。其电路原理图如下图所示：



可以看到，首先通过低两位选择端和两个 4-1 多路选择器分两组进行选择，将序号低两位符合选择端的两个输出选出来。然后再通过 2-1 多路选择器和最高位选择端进行选择，将序号最高位符合的那一项选出来。

然后根据以上实现的子电路和实验手册的要求实现如下图所示的 1 位 ALU：



根据电路原理图将器件合理摆放，适当调整元器件的形状、引脚位置和大小，并

正确连线后，进一步对电路进行观察和检验。

4.2 实验结果

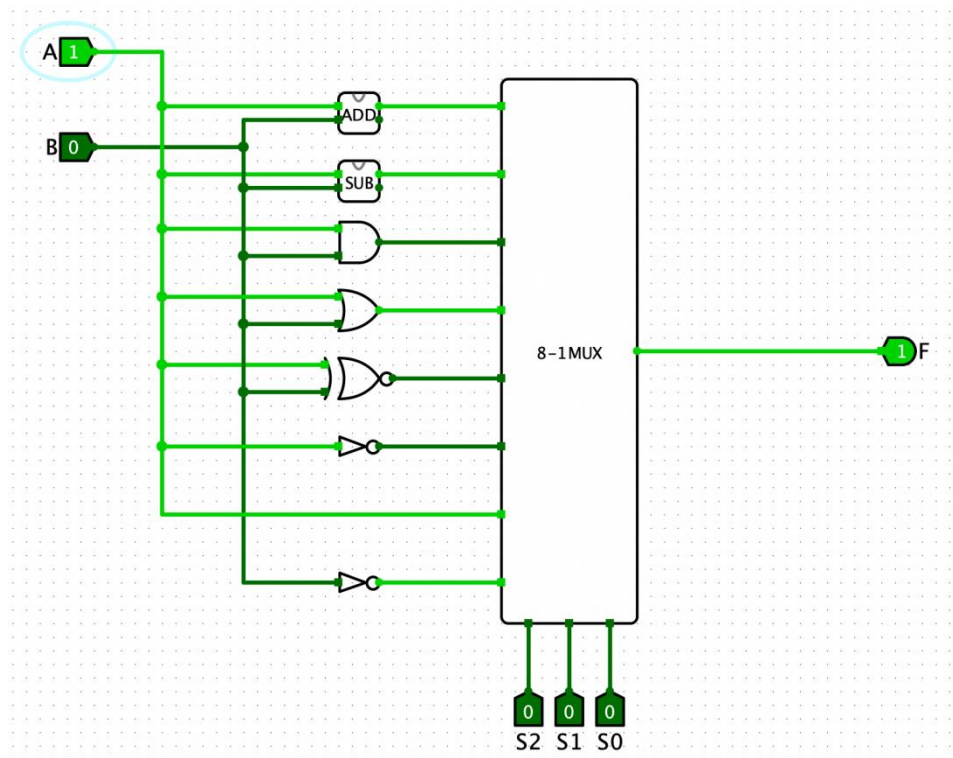
该电路通过 Logisim 自动生成的真值表如下：

A	B	S2	S1	S0	F
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	0	1	1
0	0	1	1	0	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	0	1	1
0	1	0	1	0	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	0	1	1
1	0	0	1	0	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	0	1	0
1	0	1	1	0	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	0	1	0
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	0

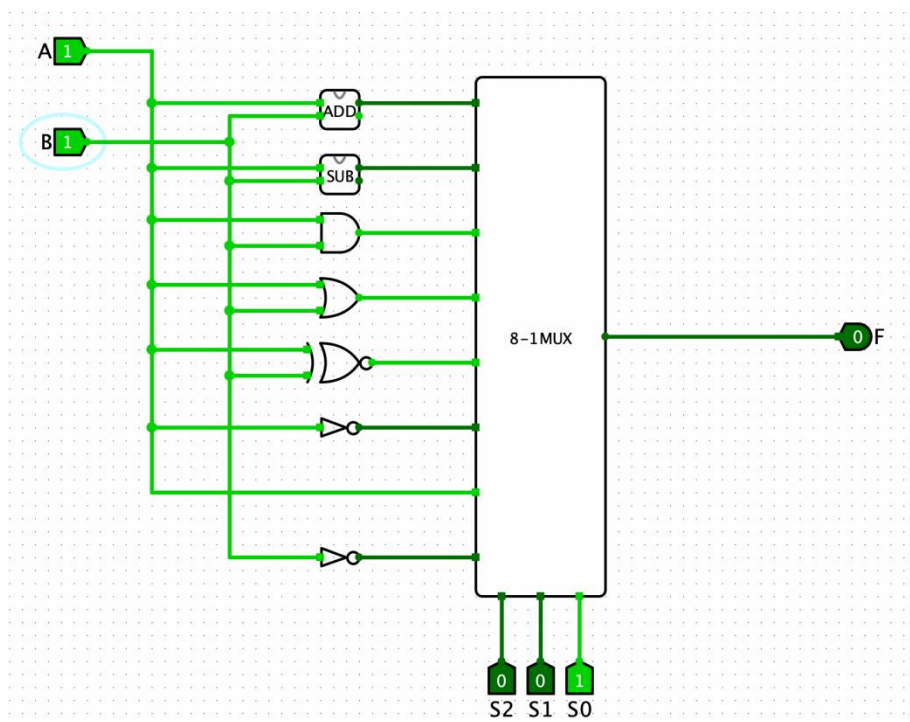
可以看到，在真值表中，当 S2S1S0 有不同的取值时，ALU 会执行不同的功能而产生相对应的输出。其输出与下表的功能相符合：

S3	S2	S1	功能
0	0	0	A 加 B
0	0	1	A 减 B
0	1	0	$A \cdot B$
0	1	1	$A + B$
1	0	0	A 异或非 B
1	0	1	A 非
1	1	0	A
1	1	1	B 非

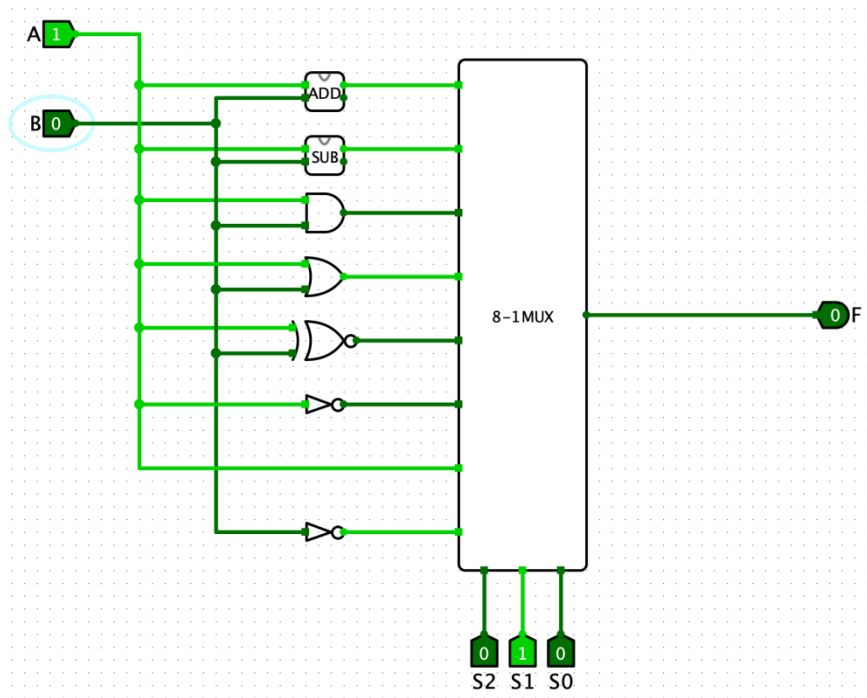
根据上面得到的结果，我们进一步选取输入对电路进行仿真检验。



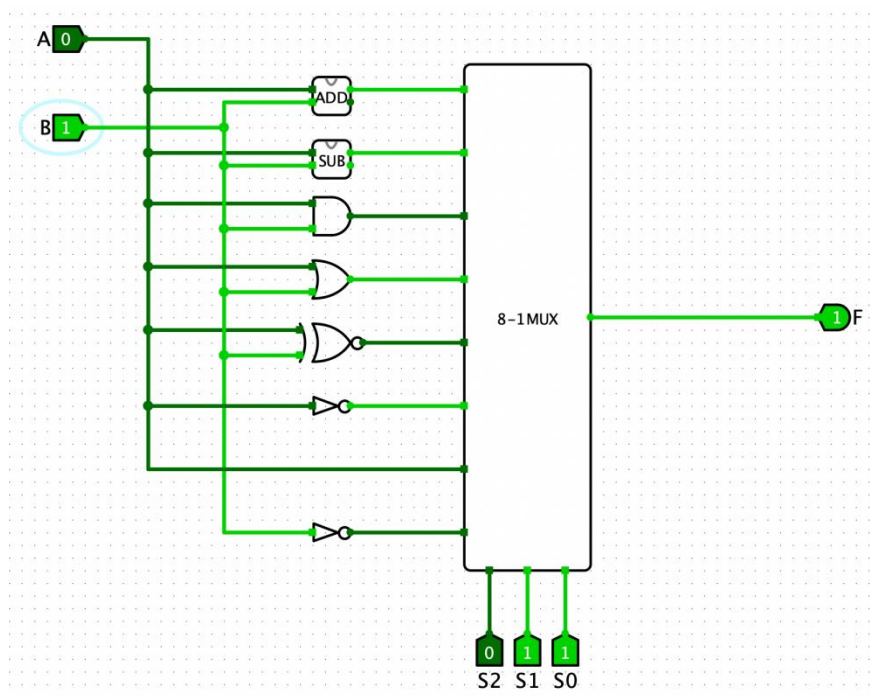
当 $S_2S_1S_0=000$, $A=1$, $B=0$ 时, ALU 执行 A 加 B, 输出端输出为 1。



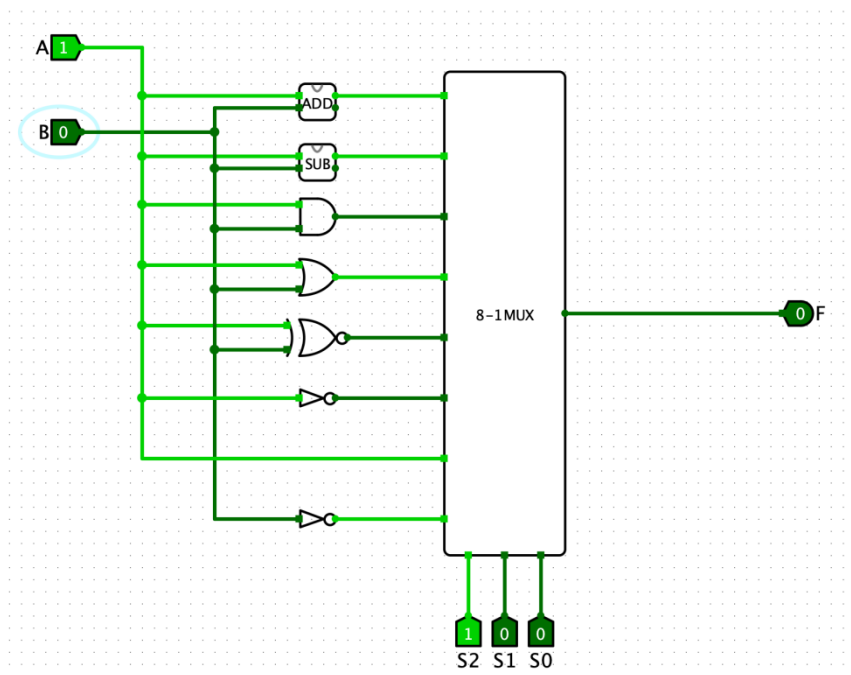
当 $S_2S_1S_0=001$, $A=1$, $B=1$ 时, ALU 执行 A 减 B, 输出端输出为 0。



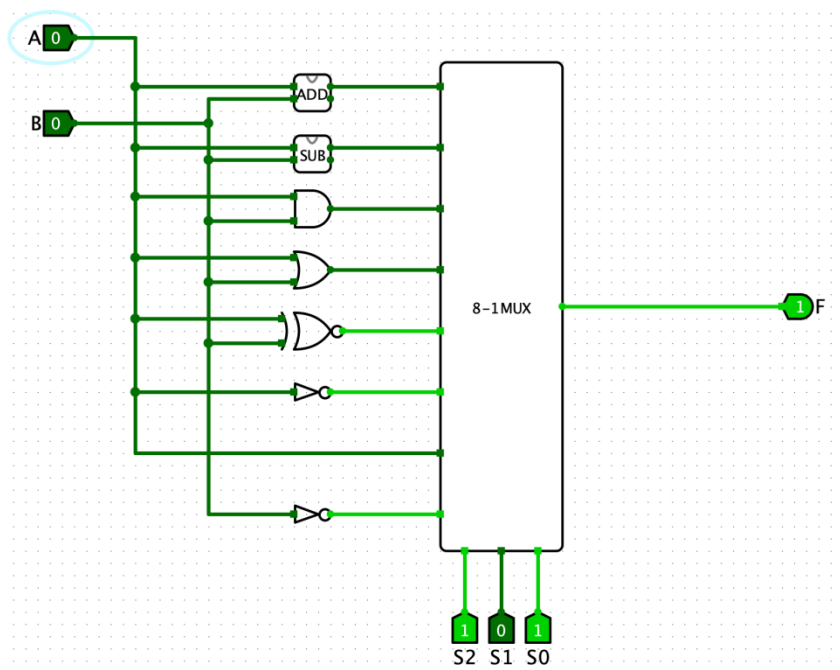
当 $S_2S_1S_0=010$, $A=1$, $B=0$ 时, ALU 执行 $A \cdot B$, 输出端输出为 0。



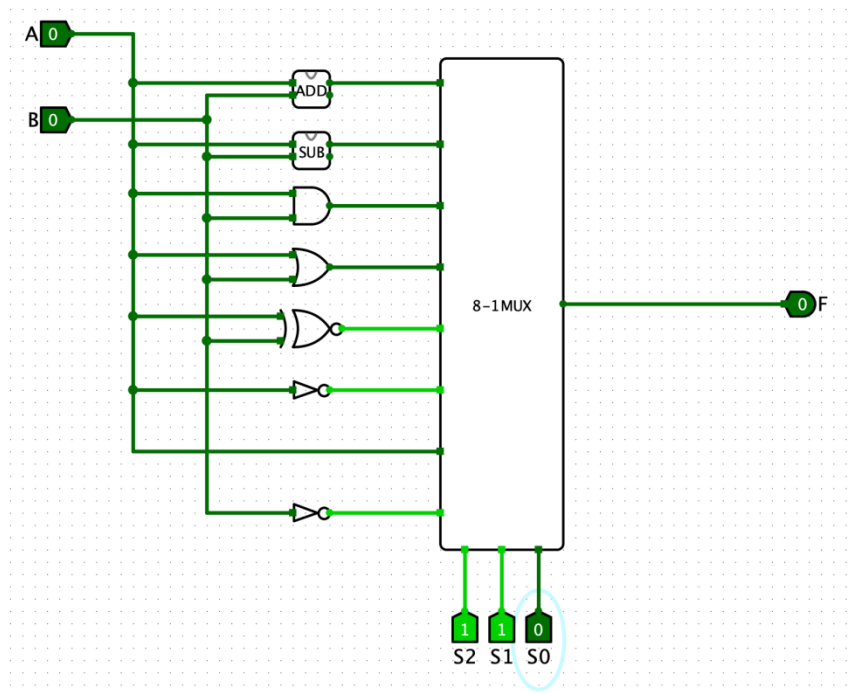
当 $S_2S_1S_0=011$, $A=0$, $B=1$ 时, ALU 执行 $A+B$ (A 或 B), 输出端输出为 1。



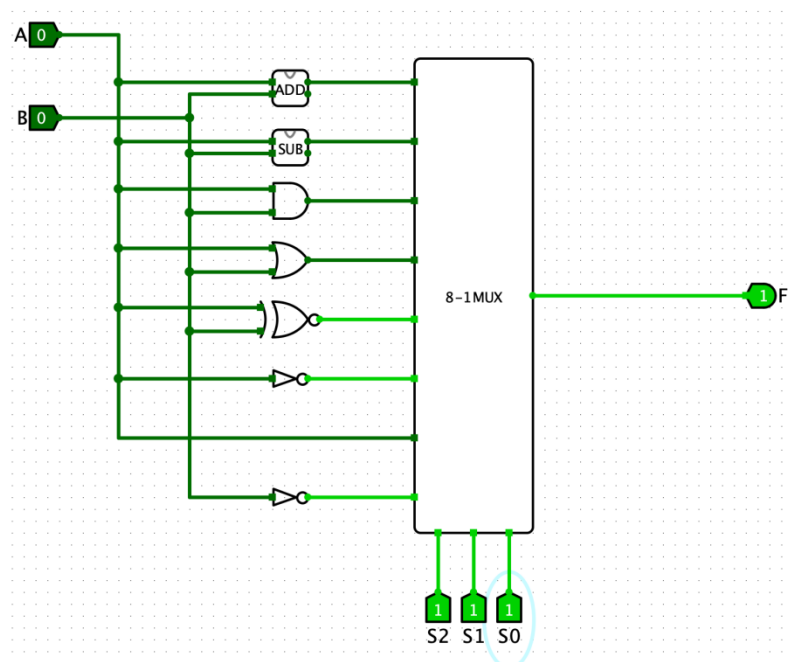
当 $S_2S_1S_0=100$, $A=1$, $B=0$ 时, ALU 执行 A 异或非 B, 输出端输出为 0。



当 $S_2S_1S_0=101$, $A=0$, $B=0$ 时, ALU 执行 A 非, 输出端输出为 1。



当 $S_2S_1S_0=110$, $A=0$, $B=0$ 时, ALU 输出 A, 输出端输出为 0。



当 $S_2S_1S_0=111$, $A=0$, $B=0$ 时, ALU 输出 B 非, 输出端输出为 1。

4.3 实验错误与原因分析

该部分实验主要是借助前面已有的知识基础实现一个综合性的组合电路, 主要部件在前面都已经实现过或实现过类似的部件, 因此这部分实验进行得较为顺利, 没有出现什么特殊的错误。

5. 思考题

1. 组合逻辑电路的一般设计步骤是什么？

a. 根据应用场景的描述进行组合逻辑电路的功能需求分析，确定输入变量和输出变量，并推演他们之间的逻辑关系，画出相应的真值表

b. 根据真值表中的输入输出关系，推导出输出函数的逻辑表达式，可采用代数法或卡诺图法等进行输出函数的逻辑化简，确定输出函数的最简逻辑表达式

c. 对得到的逻辑表达式进行变换以满足应用场景的具体实现要求，画出逻辑电路图和时序波形图，并对电路进行时序分析，必要时还需要对电路作出进一步的改进，以消除设计中存在的缺陷，如竞争冒险等问题

2. 测试电路功能有哪几种方式？

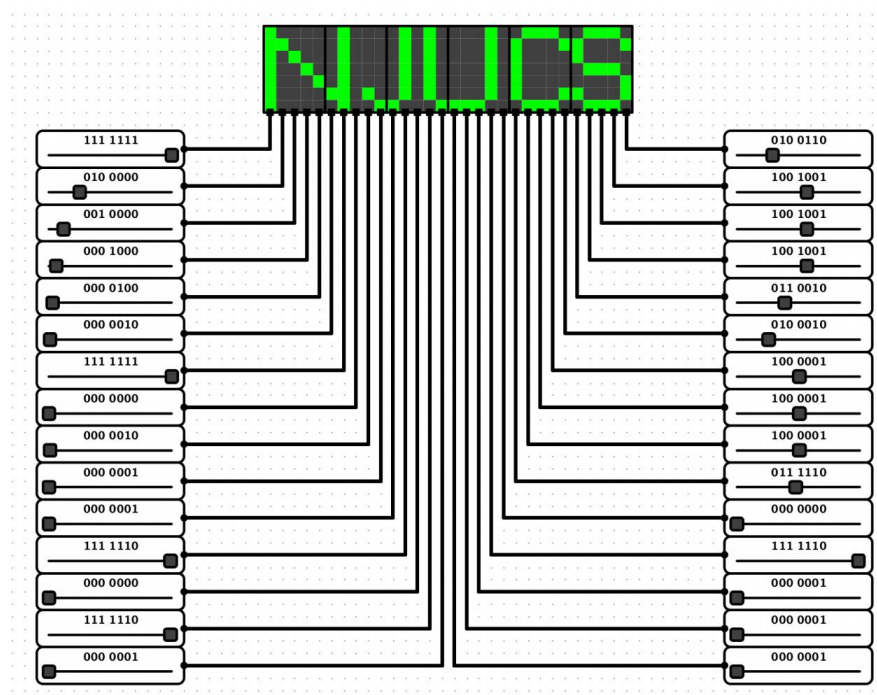
(1) 检查电路对应的真值表

(2) 连接输出组件并选取样例进行仿真测试，观察所得到的输出

(3) 绘制电路对应的时序波形图，观察得到的波形有无异常

3. 如何利用 Logisim 提供的 LED 矩阵显示 “NJUCS” 五个字符？

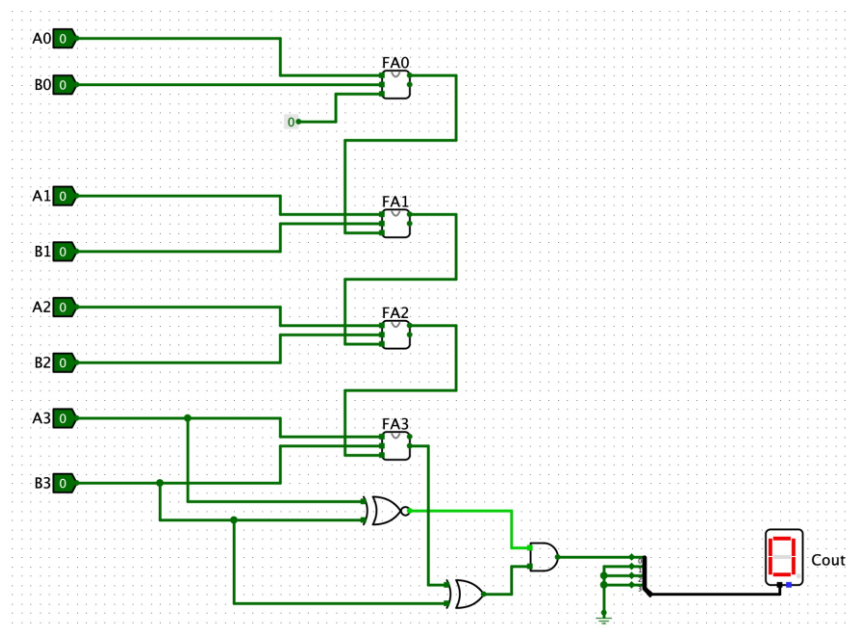
观察 Logisim 提供的 LED 矩阵，发现对于每一个其上有 5 个输入端，分别控制 LED 矩阵上的每一列的显示，每个输入端可以接受一个 7 位的输入。通过不断试验和调整，最终使用了六个 LED 矩阵，并通过 Logisim 提供的 Slider 元件产生 7 位输入。对每一列输入端所输入的数值进行不断调整，最终生成了如图所示的图像：



4. 简要说明 4 位二进制补码加法器溢出检测电路的设计思路。

设计一个 4 位二进制补码加法器溢出检测电路，其计算原理与普通加法器类似，但在最终的溢出检测逻辑上存在一定差别。在进行补码加法时，若两个加数的符号位不同，则认为不可能发生溢出，若两个加数的符号位相同，但最终得到的和的符号位与之不同，则认为发生了溢出，反之仍认为没有发生溢出。

其电路原理图如下图所示，Cout 即为判断是否溢出的标志位：



6. 实验总结

通过进行该实验对一些常见的组合逻辑电路的原理有了更清晰的认识，如译码器、优先编码器、全加器等。同时由于此次实验的实验手册所提供的信息较第一次实验更少了，因此有些电路需要自己进行设计，在设计过程中对如何根据需要设计符合要求的组合逻辑电路的整体流程有了更清晰的认识。