**实验2组合逻辑电路设计实验报告**

姓名： 学号：

**一、实验目的**

1.掌握组合逻辑电路的设计方法和步骤，实现译码器、编码器等基本组合逻辑电路。

2.掌握全加器的设计方法和原理，在 1 位全加器基础上实现一个 4 位串行进位加法器。

3.掌握多路选择器的应用。

4.掌握汉明码校验电路的设计方法。

**二、实验环境**

Logisim：https://github.com/Logisim-Ita/Logisim

**三、实验内容**

**1. 译码器实验**

输入引脚G2A\_L,G2B\_L，G1控制译码器工作，当且仅当G1=1，G2A\_L=G2B\_L =0的时候，该译码器正常进行工作。

输入引脚A,B,C用于表示输入二进制数ABC的各个位数的数值。

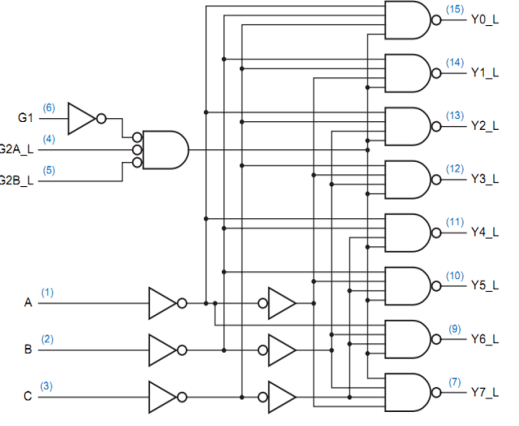
输出引脚Y0\_L,Y1\_l,Y2\_L,Y3\_L,Y4\_L,Y5\_L,Y6\_L,Y7\_L用于表示该二进制数在十进制下的

大小，Yi\_L=1时表示该二进制数下在十进制数下的大小为i，同一时间只有一个Yi\_L为1

实验步骤

1. 基本原理

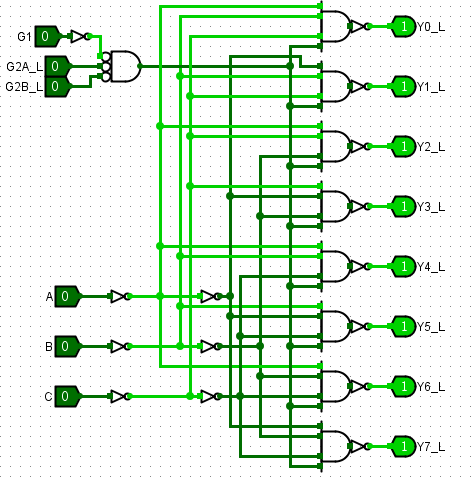
原理图:



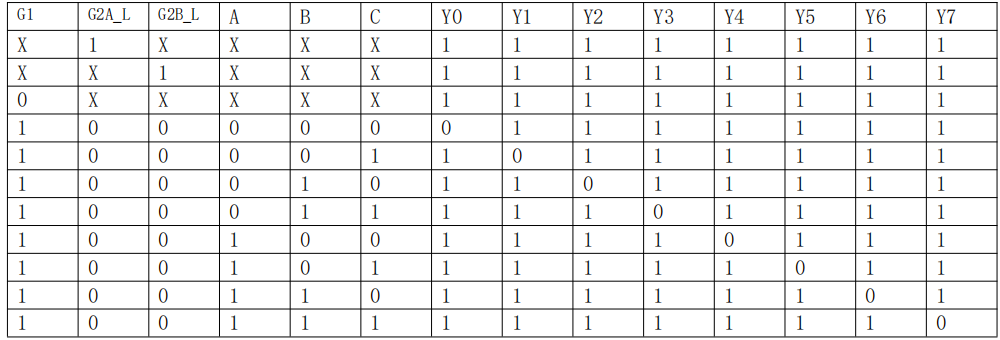
2)需要8 个 4 输入与非门、7 个非门、1 个与门、7 个输入引脚、8 个输出引脚

3)添加逻辑门，输入输出引脚并连线，最后进行仿真测试，验证是否正确

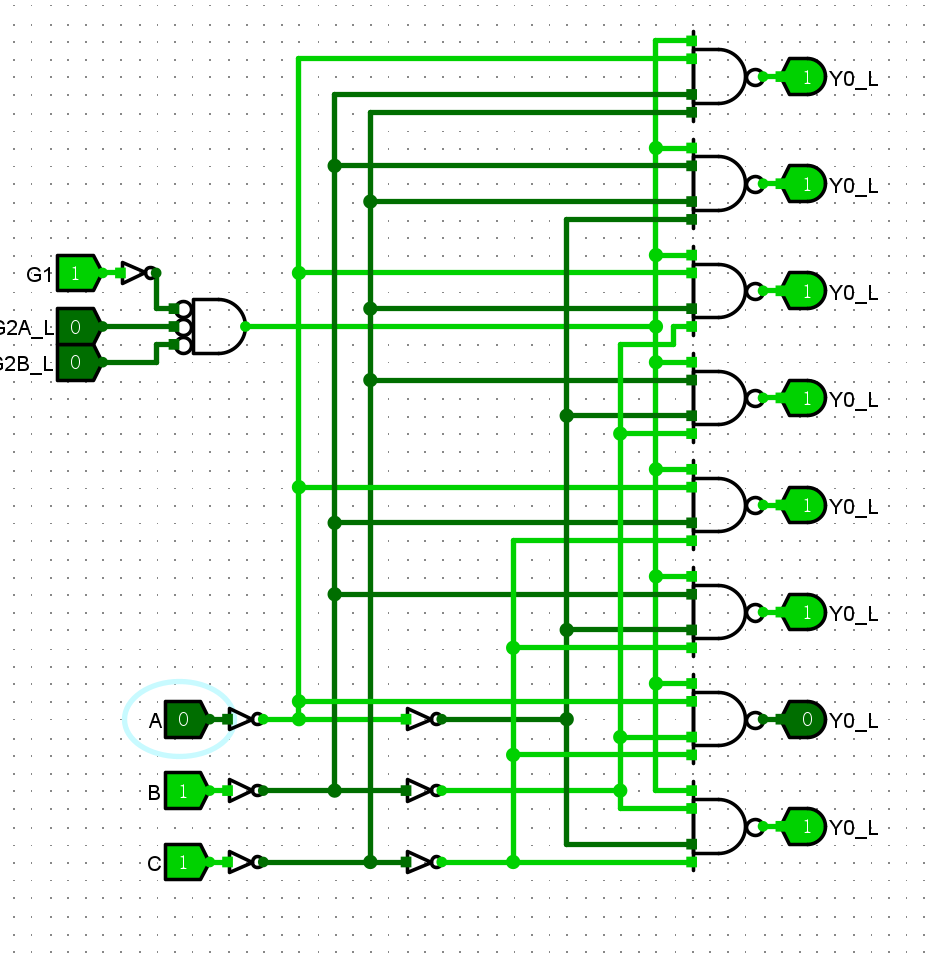
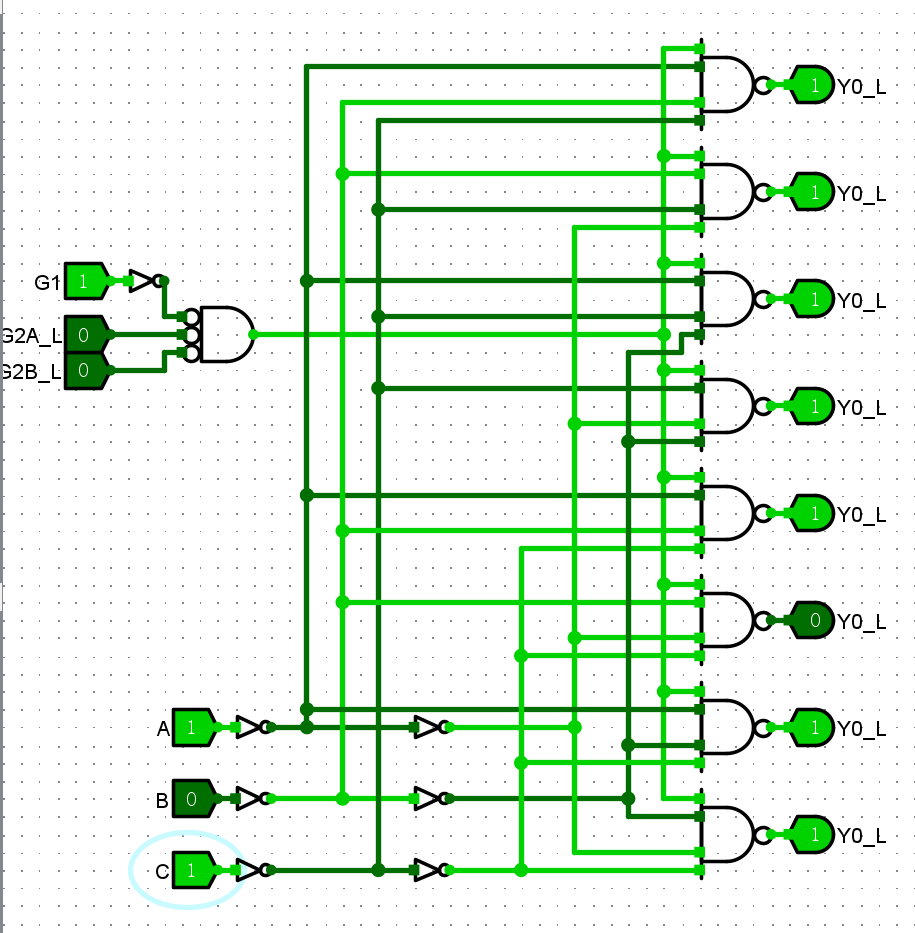
电路：



真值表:



仿真实验（部分）：



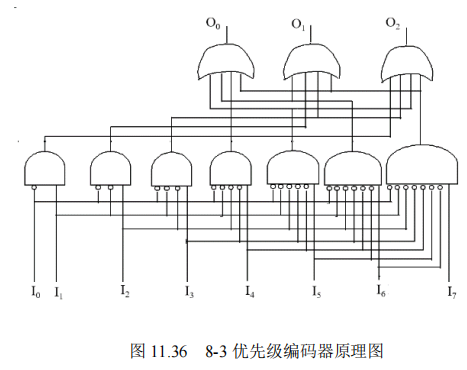
1. **编码器实验**

将十进制数转换为二进制表示O0O1O2，并用数字表示出来

实验步骤

1）基本原理

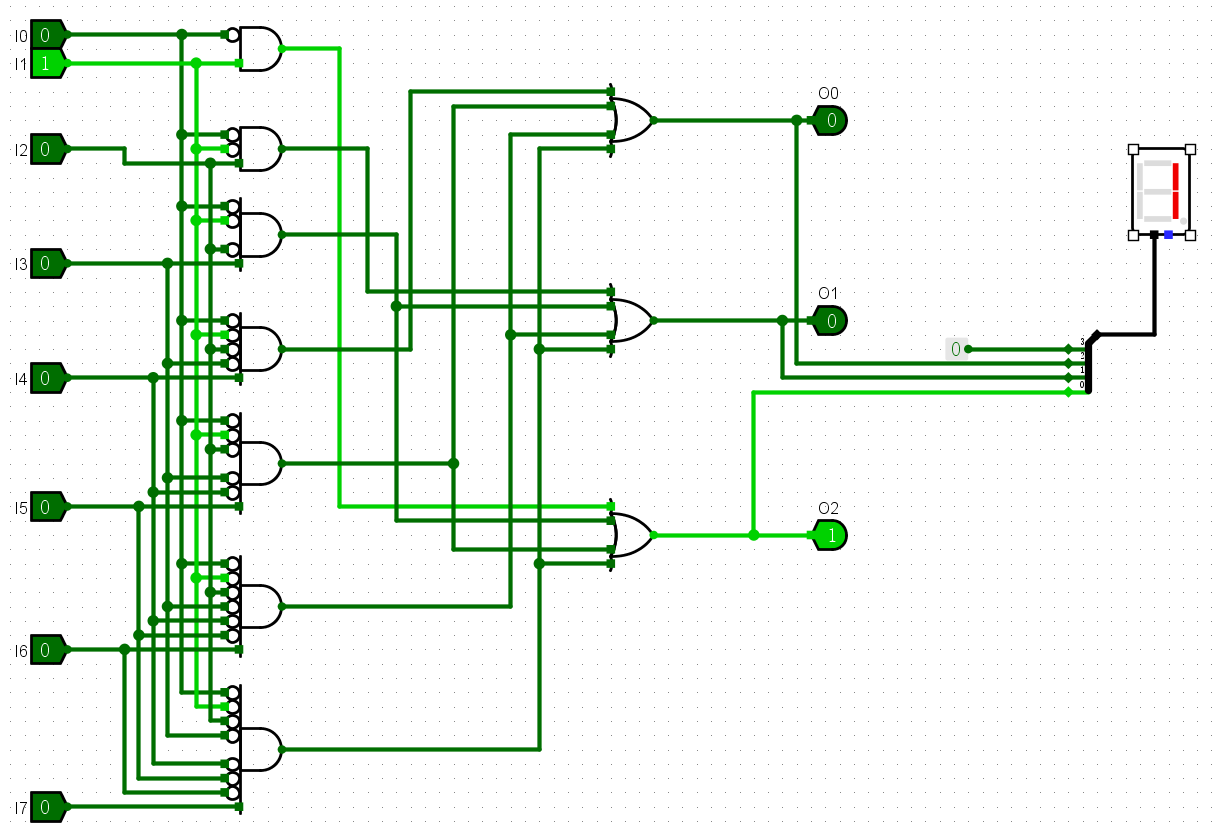
原理图:

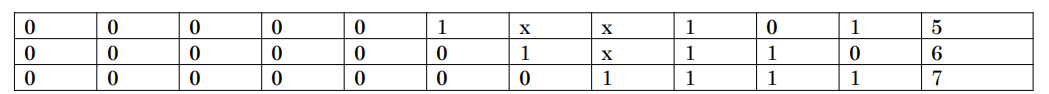


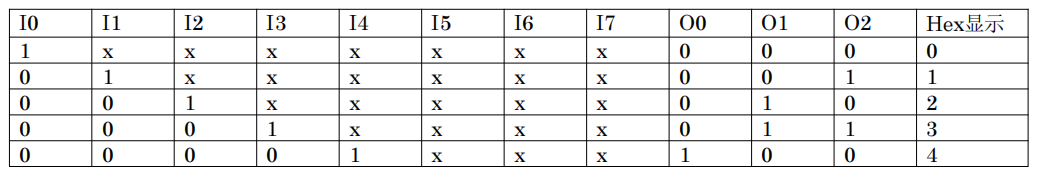
1. 按图索需

3)添加逻辑门，输入输出引脚并连线，最后进行仿真测试，验证是否正确

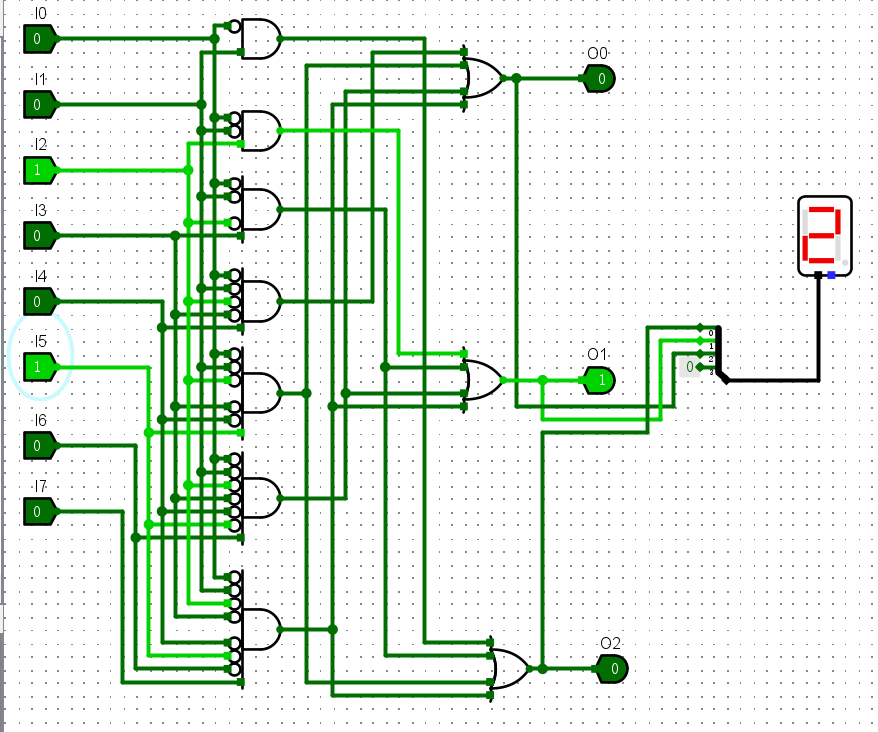
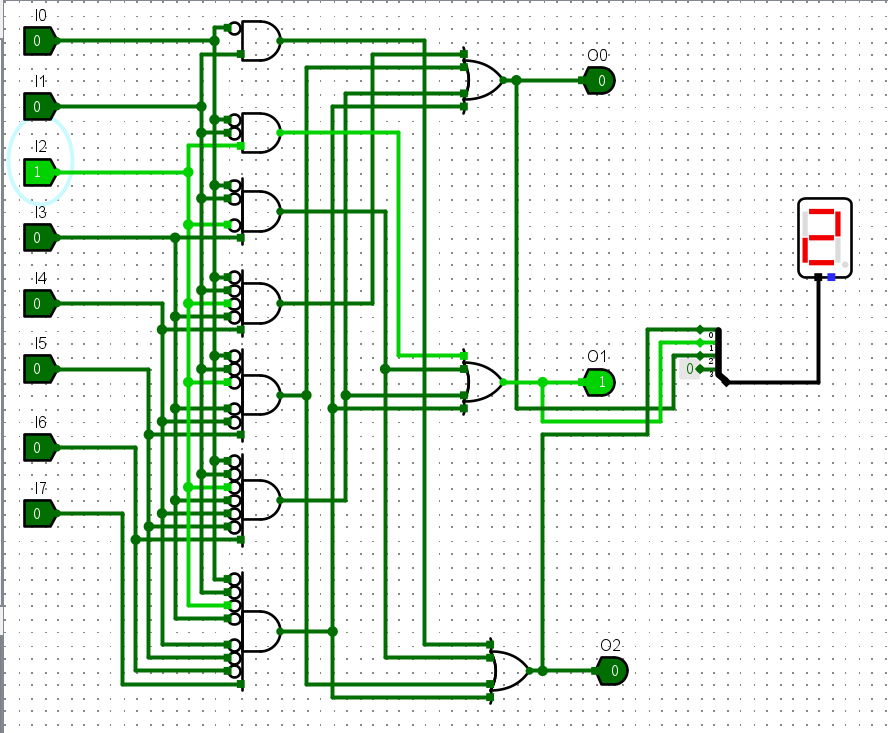
电路:



真值表



仿真实验（部分）：



不难发现这个例子中体现了“优先权”

**3.加法器实验**

设计一个全加器（FA），在此基础上将 4 个全加器串联成一个 4 位串行进位加法器。将输入、输出分别连接到 16 进制数码显示管

实验步骤

1. 基本原理

F=A^B^Cin （F为本位）

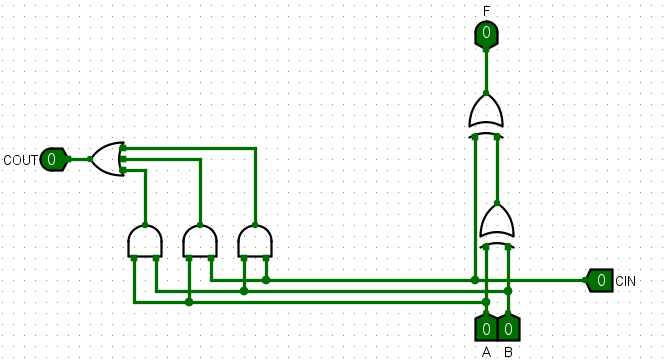
Cout=A B + B Cin +A Cin （cin为低位进位，cout为向高位进位）

1. 按照公式取所需

3）添加逻辑门，输入输出引脚并连线，最后进行仿真测试，验证是否正确

完成FA设计

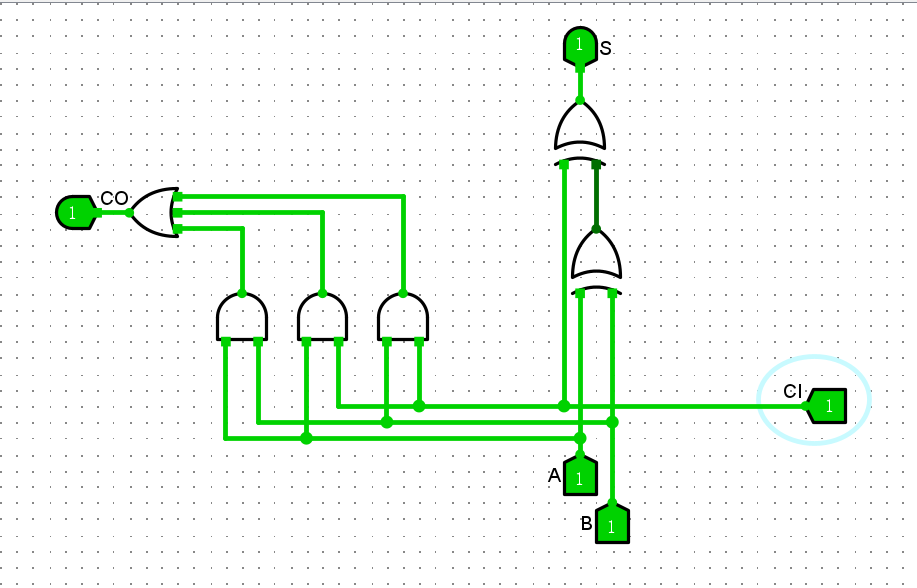
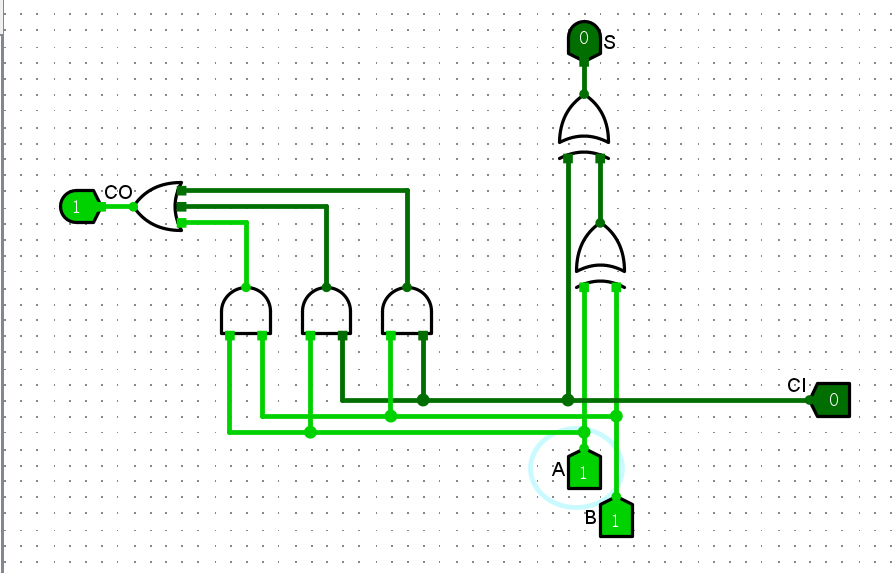
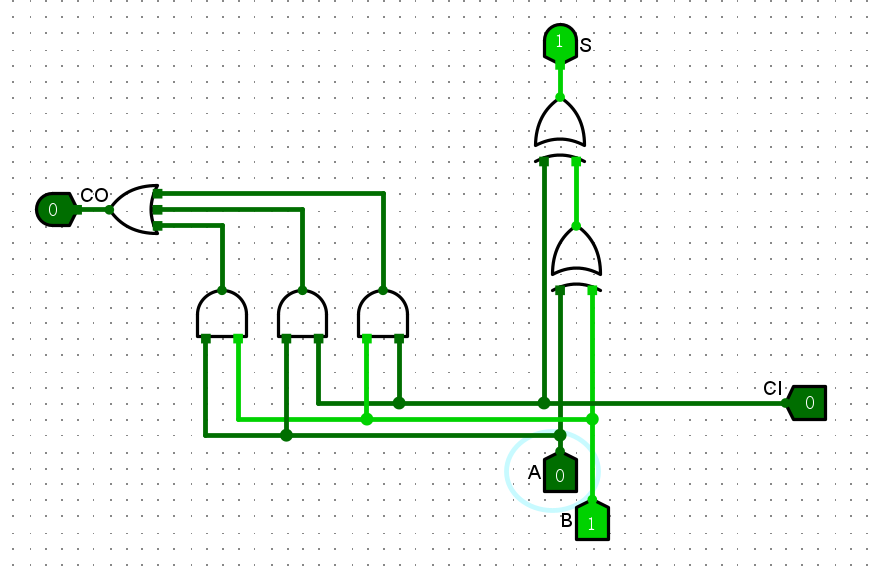
电路图:



真值表:

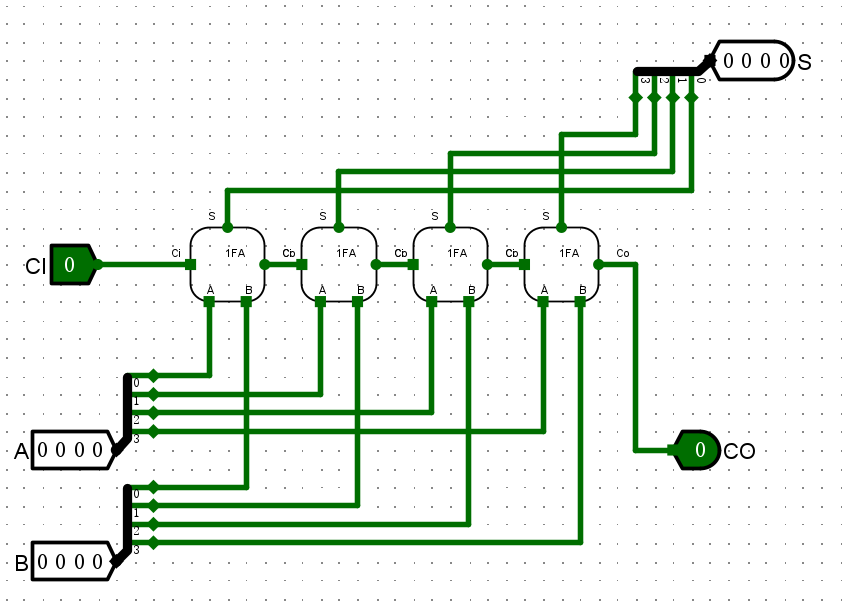
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | Cin | F | Cout |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

仿真实验：

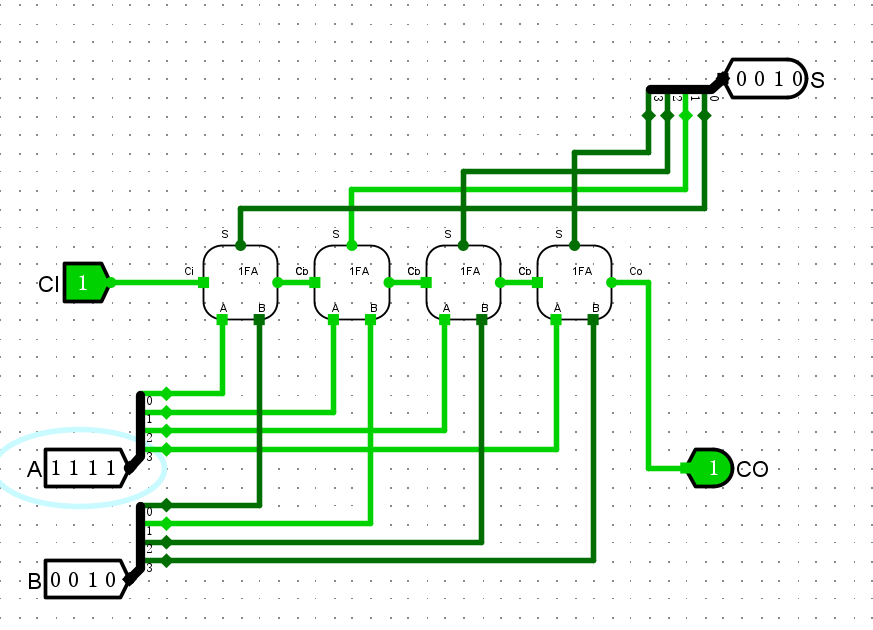
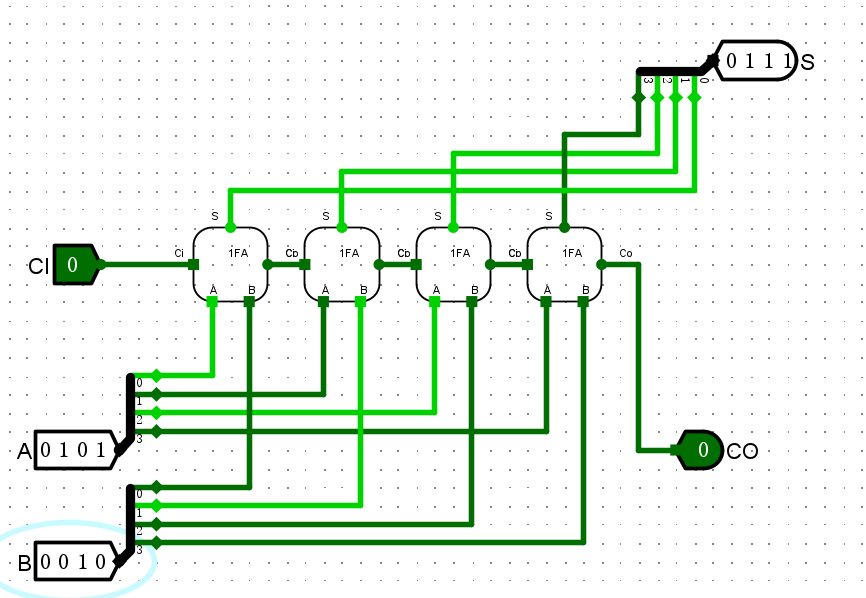


4位加法器

FA为子电路，设计封装电路后组装，最初进位为0，最终电路为

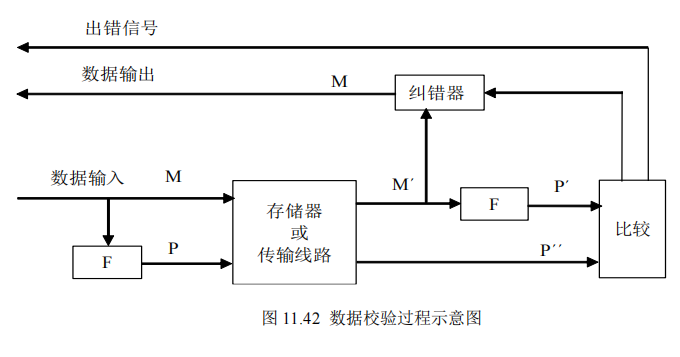


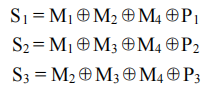
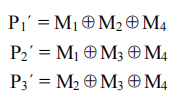
仿真实验：

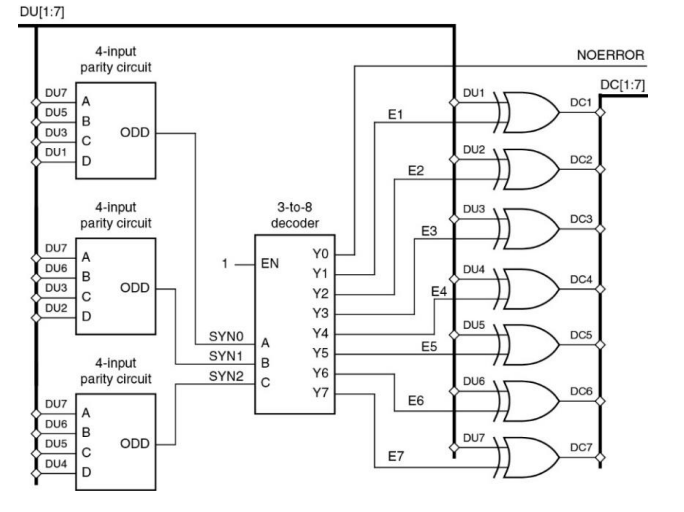


**4、汉明码校验电路**

1)基本原理



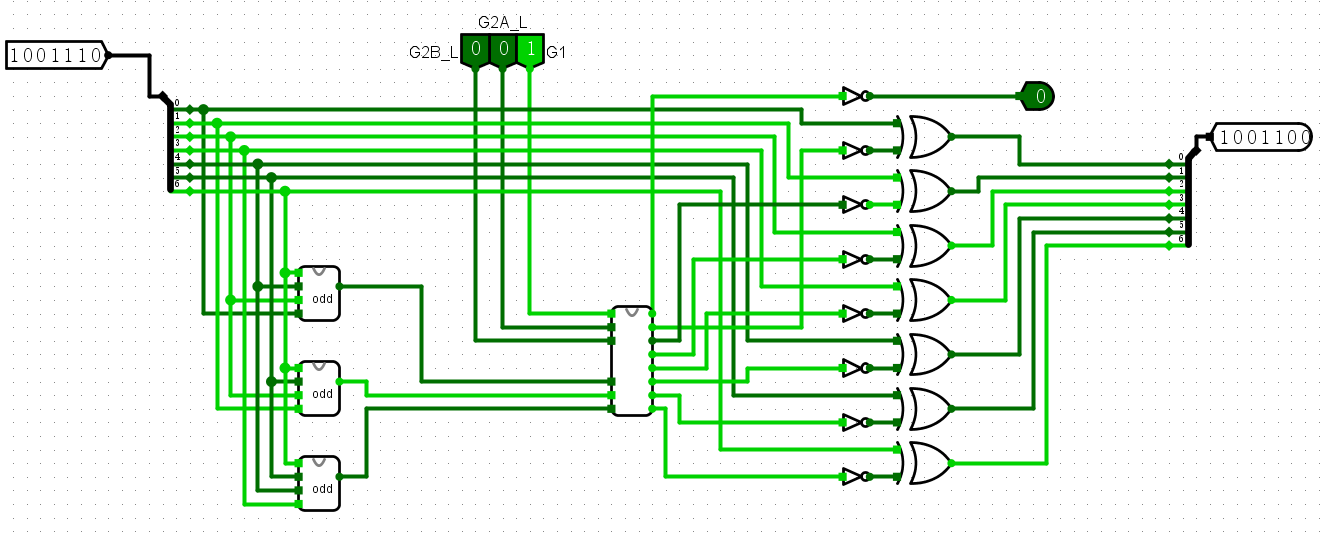




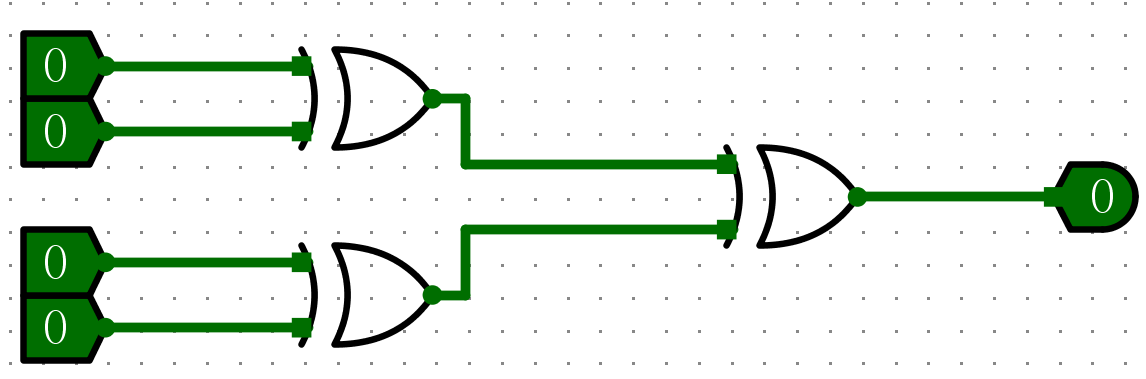
2）按需放置元件

3）添加逻辑门，输入输出引脚并连线，最后进行仿真测试，验证是否正确

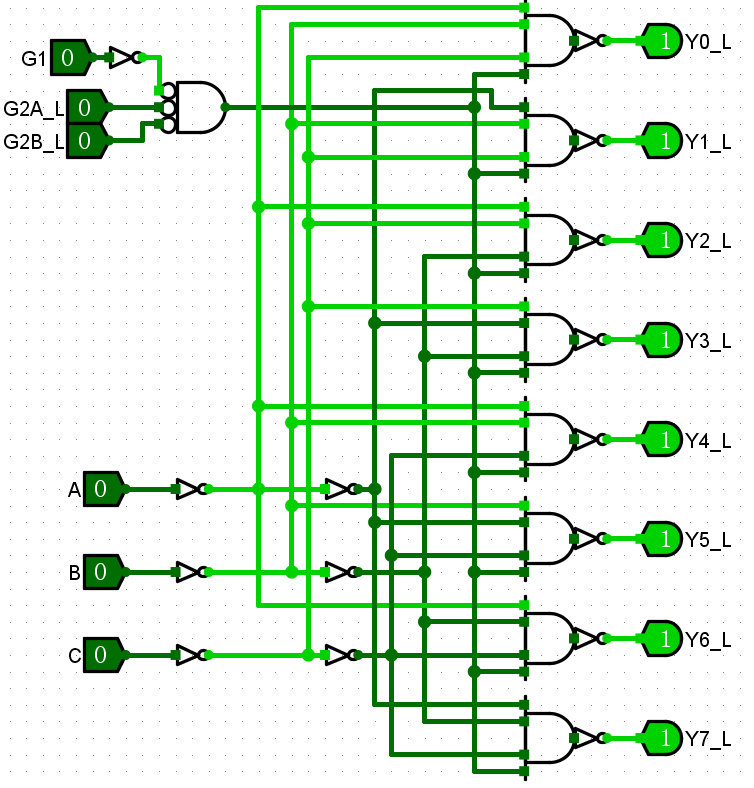
电路图:



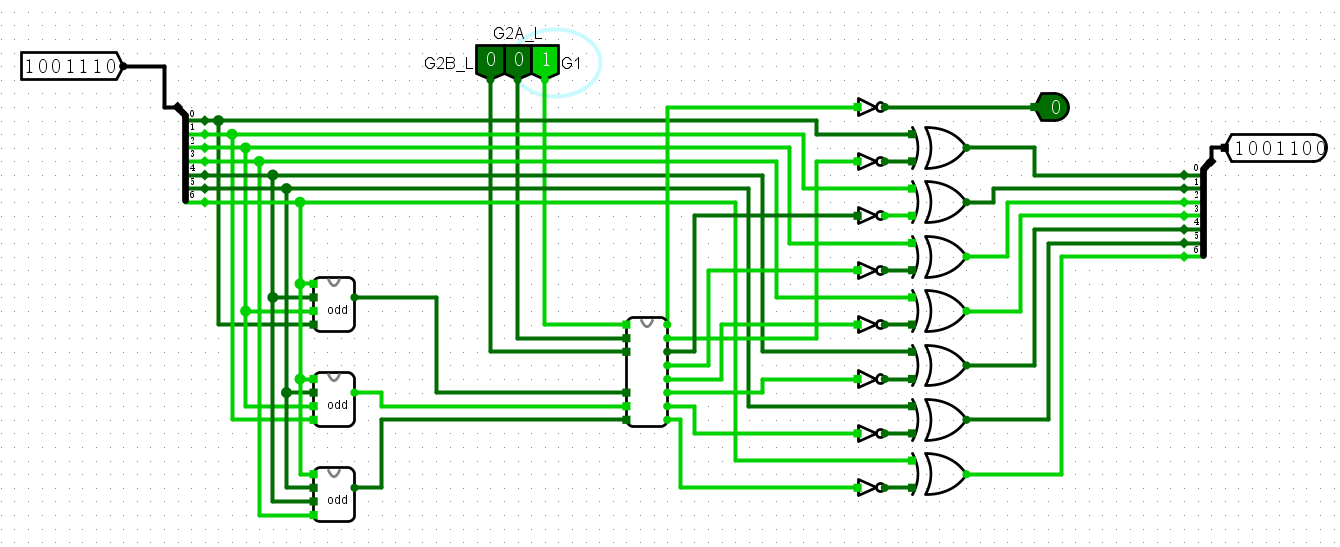
其中封装的4位偶校验器

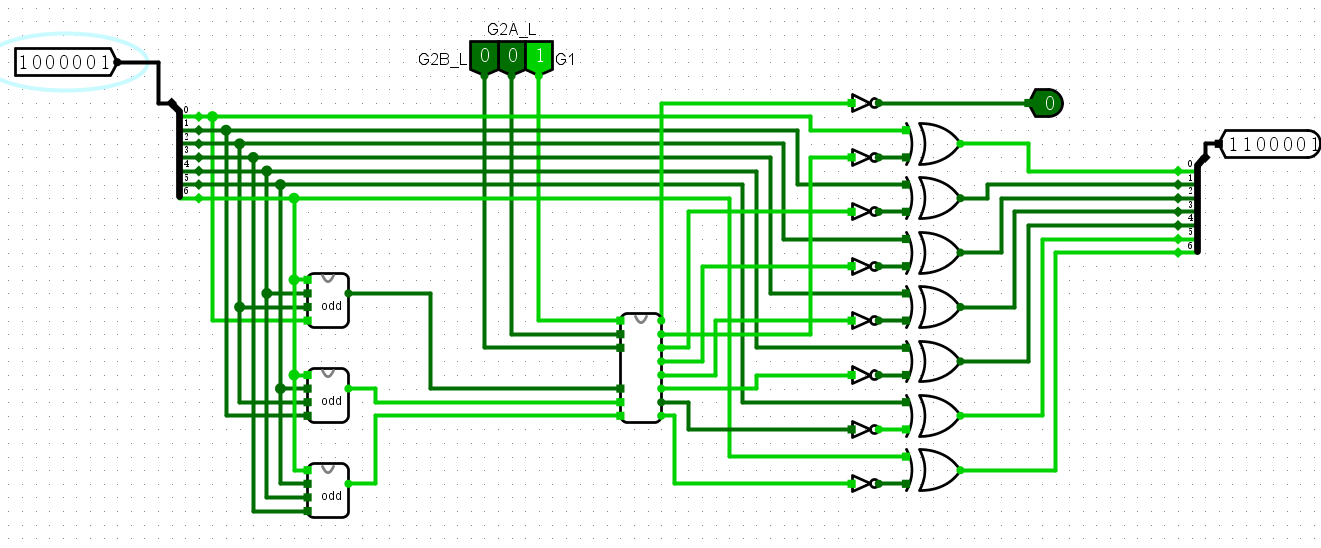


封装的3-8译码器为



在 DU[1:7]处分别输入 1000001 和 1001110 等 7 位二进制位串进行验证，如图验证结果正确





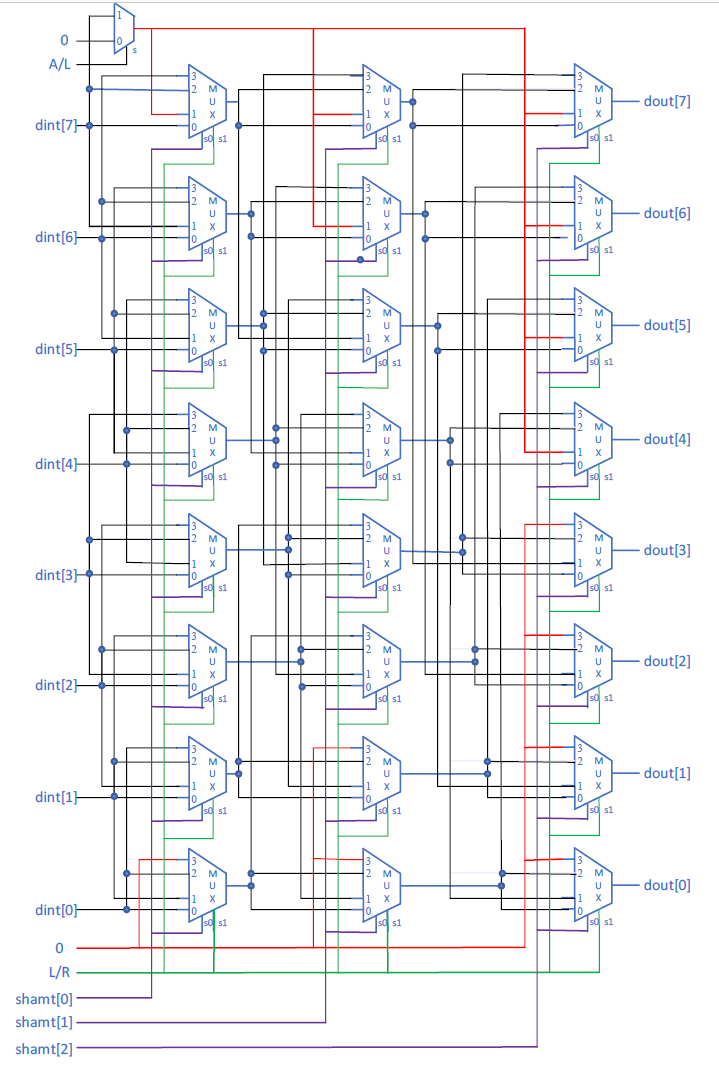
5、桶形移位器

1）基本原理



实验电路分三级实现0至7位的左移或右移。第一级利用shamt[0]来控制是否需要移动一位，第二级在第一级的移动结果上用shamt[1]来控制是否要移动两位，第三级在第二级的基础上再对应判断是否要移动四位。每个四路选择器有两位控制端，控制端低位S0为当前级是否需要移动，对应shamt[i]，当S0=0时，选中4路选择器的0号或2号输入端口，均不做任何移动。当S0=1时，控制端高位S1对应L/R输入，当S1=0时，表示右移，选中4路选择器的1号输入端口；当S1=1时，表示左移，选中4路选择器的3号输入端口。这两个输入端分别连接了数据低位或高位的上一级输出。对于算术和逻辑右移的操作，是通过A/L来选择移入的是0还是din[7]

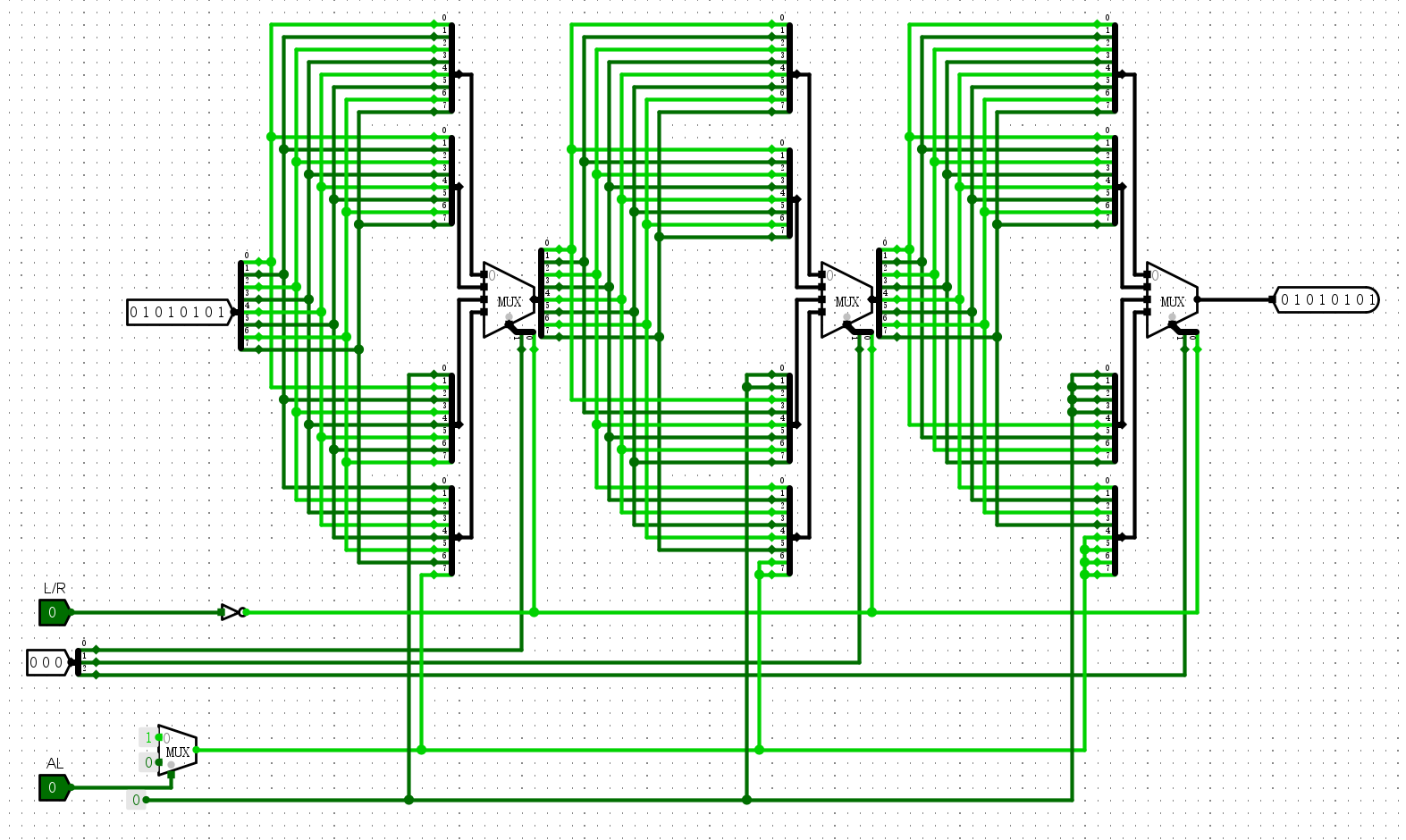
电路图如下：



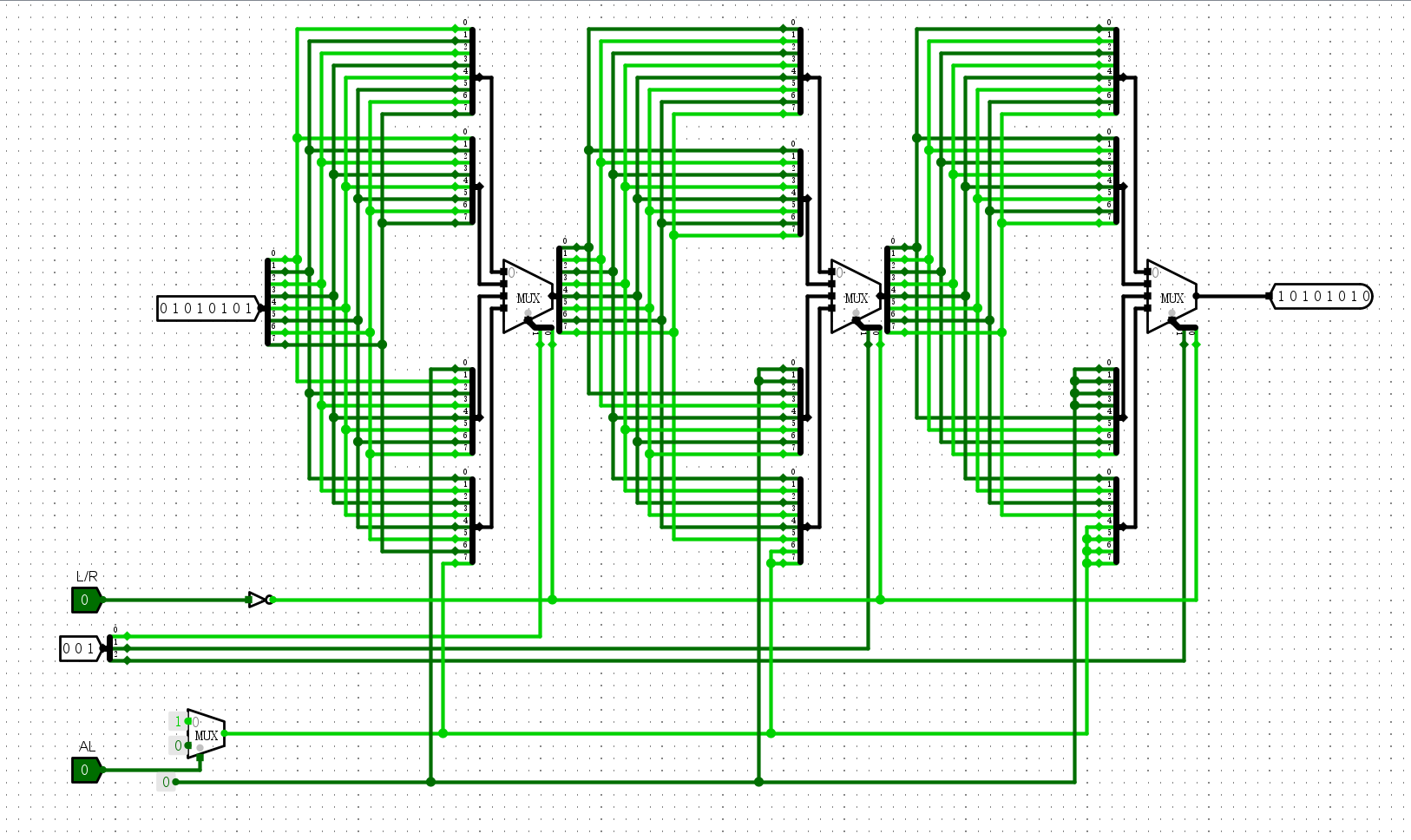
2）按需摆放电路元件

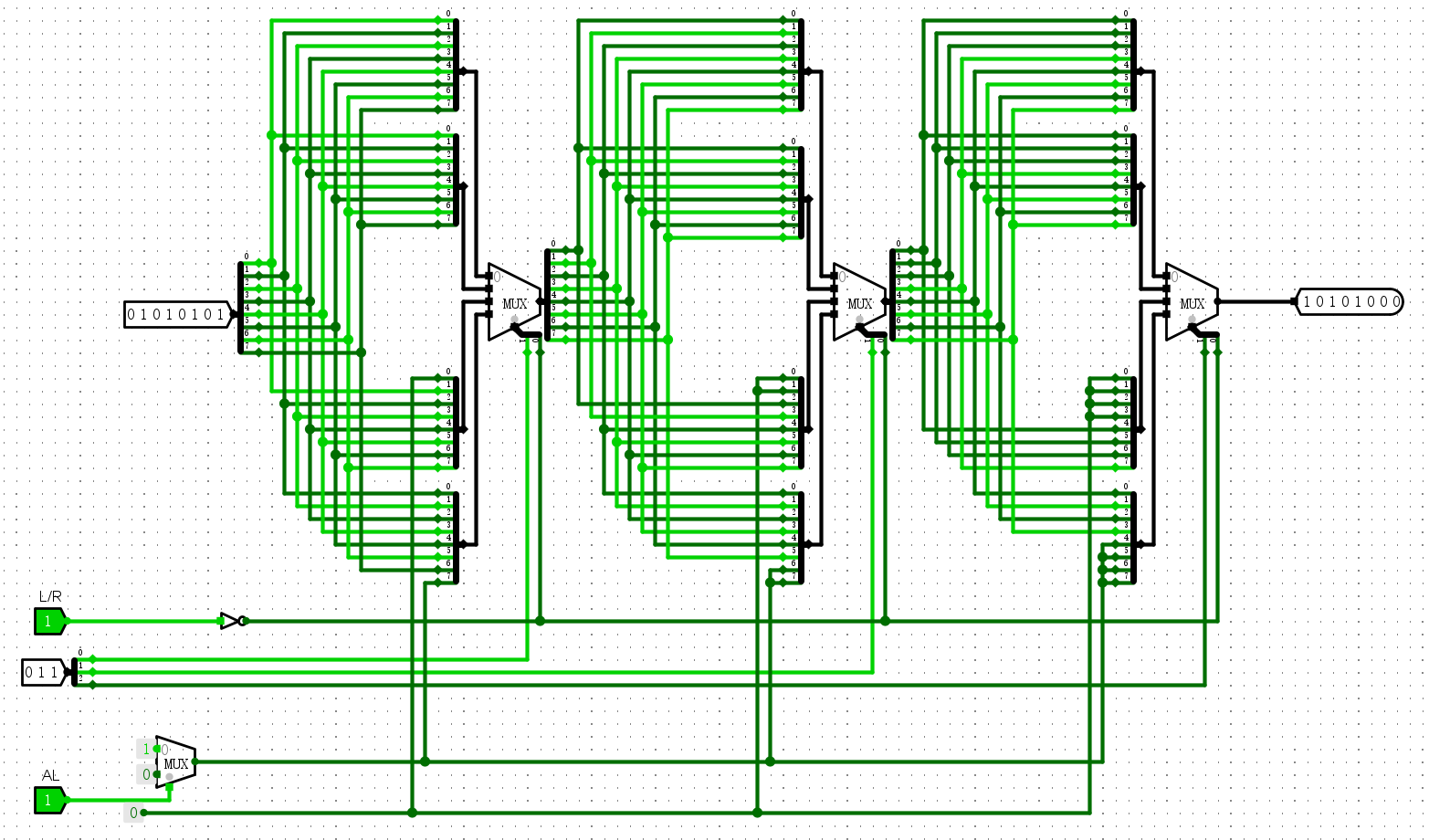
3）添加逻辑门，输入输出引脚并连线，最后进行仿真测试，验证是否正确

电路图：



仿真实验（部分）：

对“01010101”算数右移1位

对“01010101”逻辑左移3位

1. **实验中的错误**

实验中没有遇到明显的错误

**五、思考题**

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。

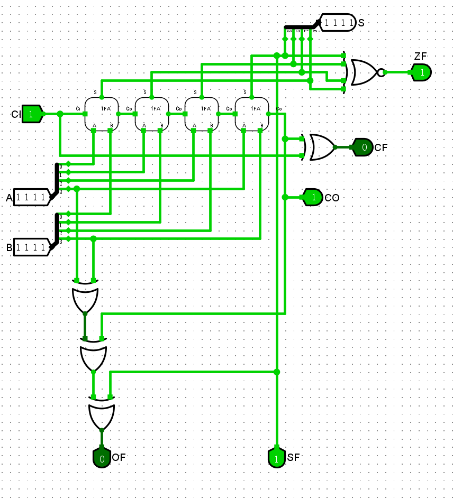
符号标志SF=F(n-1)（无符号没有意义）

溢出标志OF=Cn-1⊕Cn （无符号没有意义）

进位标志CF=Cout⊕Cin（带符号没有意义）

零标志ZF=！(F0\*F1\*...\*Fn-1)

其中Cn-1=An⊕Bn⊕Fn

所以电路图为：

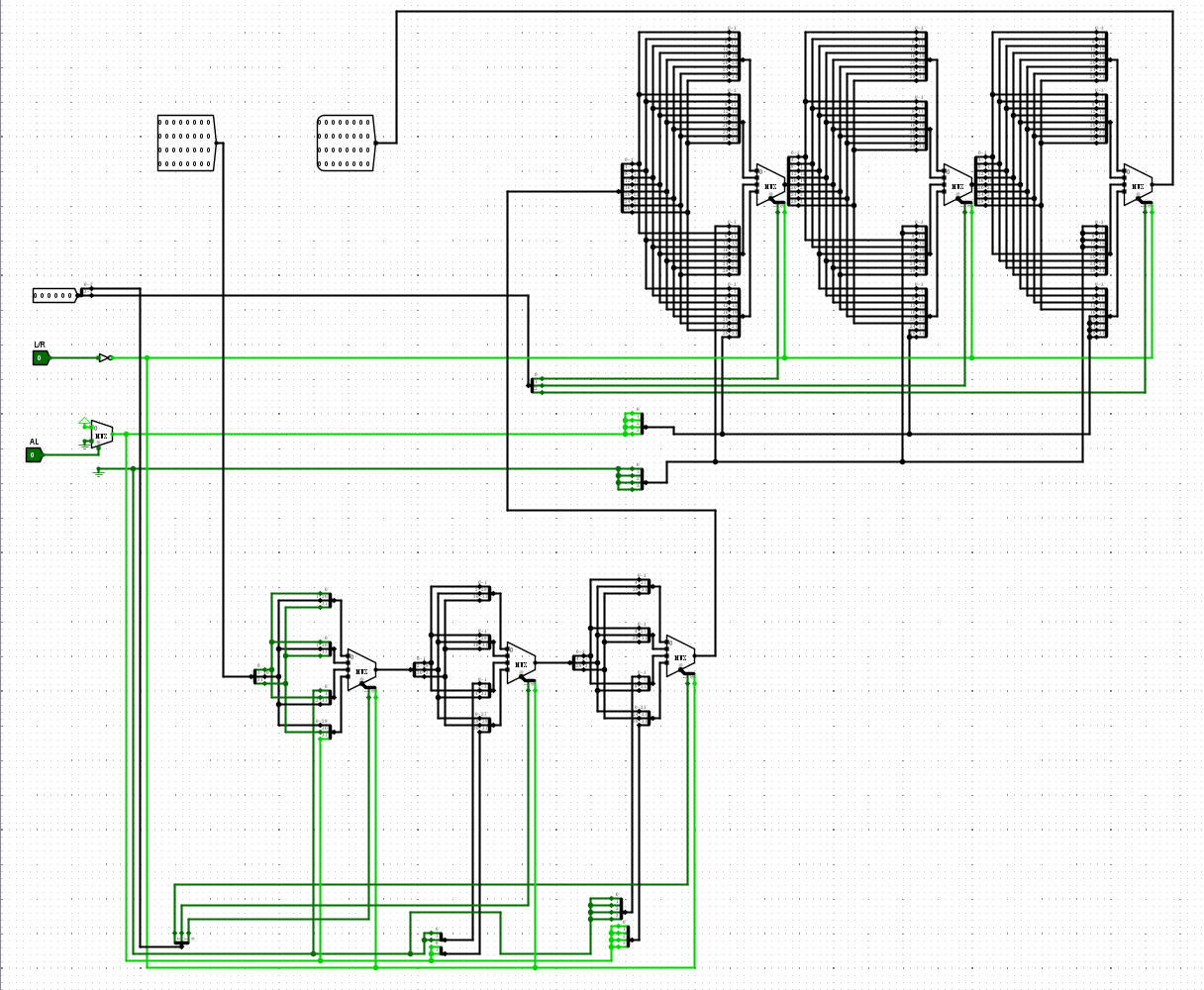
2、在执行比较指令时，通常使用减法运算后，判断标志位的方式来实现，试通过上述加法器实验举例说明判别的方法。

·有符号数比较大小：用（SF^OF）与ZF来判断大小。ZF=0时等大，ZF不等于0时，（SF^OF）为1则被减数小于减数，反之则反

·无符号数比较大小：用CF与ZF来判断大小。ZF=0时等大，ZF不等于0时，CF为1则被减数小于减数，反之则反

3、如何使用8位桶形移位器扩展到32位桶形移位器？

电路如下：



4、Logisim提供输出组件LED矩阵，通过点亮led灯的方式显示字符，修改LED矩阵行列属性为16\*16，显示“南大”两个汉字，字体可自选。

