

# 实验二报告

231220088 陈翔宇

## 实验内容

### 一、译码器

#### 基本原理

根据题目描述可以画出真值表

G1	G2A_L	G2B_L	A	B	C	Y0_L	Y1_L	Y2_L	Y3_L	Y4_L	Y5_L	Y6_L	Y7_L
0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	1	0	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	0	1	1	1	1	1	1	1	1
0	0	0	1	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1
0	0	1	0	0	0	1	1	1	1	1	1	1	1
0	0	1	0	0	1	1	1	1	1	1	1	1	1
0	0	1	0	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	1	1	1	1	1	1	1	1
0	0	1	1	0	0	1	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1	1	1	1
0	0	1	1	1	0	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	1	1	1	1	1	1	1	1	1
0	1	0	0	1	0	1	1	1	1	1	1	1	1
0	1	0	0	1	1	1	1	1	1	1	1	1	1
0	1	0	1	0	0	1	1	1	1	1	1	1	1
0	1	0	1	0	1	1	1	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1	1	1	1
0	1	0	1	1	1	1	1	1	1	1	1	1	1
0	1	1	0	0	0	1	1	1	1	1	1	1	1
0	1	1	0	0	1	1	1	1	1	1	1	1	1
0	1	1	0	1	0	1	1	1	1	1	1	1	1
0	1	1	0	1	1	1	1	1	1	1	1	1	1
0	1	1	1	0	0	1	1	1	1	1	1	1	1
0	1	1	1	0	1	1	1	1	1	1	1	1	1
0	1	1	1	1	0	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	1	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1	1	1	1	1	1	1
1	0	0	0	1	0	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1	1	1	1
1	0	0	1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	0	1	0	1	1	1	1	1	1	1
1	0	0	1	1	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	1	1	1	1	1	1
1	0	1	0	0	0	0	1	1	1	1	1	1	1
1	0	1	0	0	1	0	1	1	1	1	1	1	1
1	0	1	0	1	0	0	1	1	1	1	1	1	1
1	0	1	0	1	1	0	1	1	1	1	1	1	1
1	0	1	1	0	0	0	1	1	1	1	1	1	1
1	0	1	1	0	1	0	1	1	1	1	1	1	1
1	0	1	1	1	0	0	1	1	1	1	1	1	1
1	0	1	1	1	1	0	1	1	1	1	1	1	1
1	1	0	0	0	0	0	1	1	1	1	1	1	1
1	1	0	0	0	1	0	1	1	1	1	1	1	1
1	1	0	0	1	0	0	1	1	1	1	1	1	1
1	1	0	1	0	0	0	1	1	1	1	1	1	1
1	1	0	1	0	1	0	1	1	1	1	1	1	1
1	1	0	1	1	0	0	1	1	1	1	1	1	1
1	1	0	1	1	1	0	1	1	1	1	1	1	1
1	1	1	0	0	0	0	1	1	1	1	1	1	1
1	1	1	0	0	1	0	1	1	1	1	1	1	1
1	1	1	0	1	0	0	1	1	1	1	1	1	1
1	1	1	0	1	1	0	1	1	1	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1	1	1	1
1	1	1	1	0	1	0	1	1	1	1	1	1	1
1	1	1	1	1	0	0	1	1	1	1	1	1	1
1	1	1	1	1	1	0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1

化简后的表达式:

$$y_{0l} = \overline{G1} + C + B + A + G2B_L + G2A_L$$

$$Y_{1L} = \overline{G1} + \overline{A} + C + B + G2B_L + G2A_L$$

$$Y_{2L} = \overline{G1} + \overline{B} + C + A + G2B_L + G2A_L$$

$$Y_{3L} = \overline{G1} + \overline{A} + \overline{B} + C + G2B_L + G2A_L$$

$$Y_{4L} = \overline{G1} + \overline{C} + B + A + G2B_L + G2A_L$$

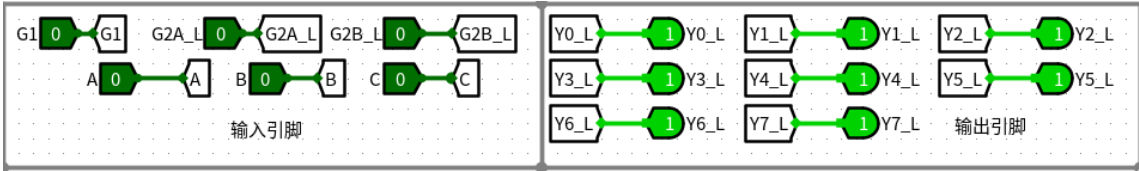
$$Y_{5L} = \overline{G1} + \overline{A} + \overline{C} + B + G2B_L + G2A_L$$

$$Y6_L = \overline{G1} + \overline{B} + \overline{C} + A + G2B_L + G2A_L$$

$$Y7_L = \overline{G1} + \overline{A} + \overline{B} + \overline{C} + G2B_L + G2A_L$$

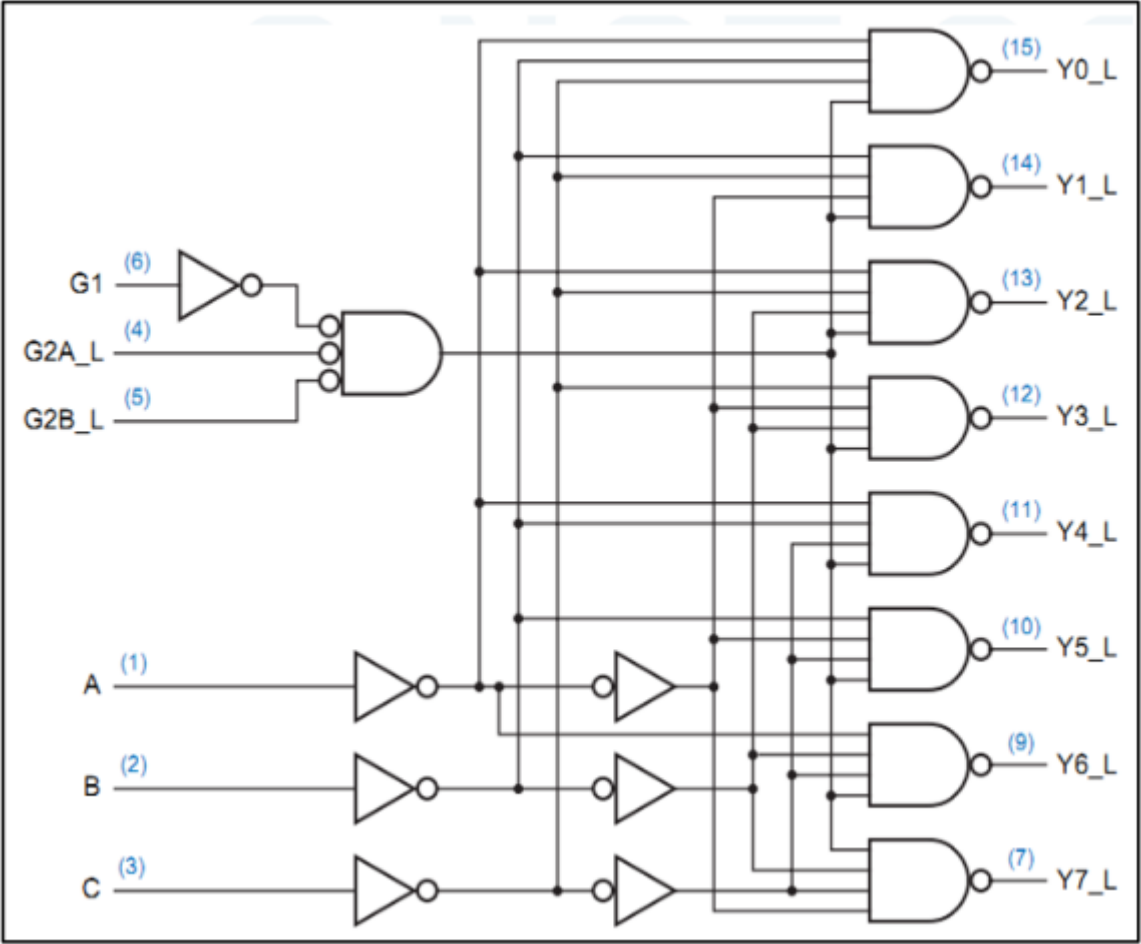
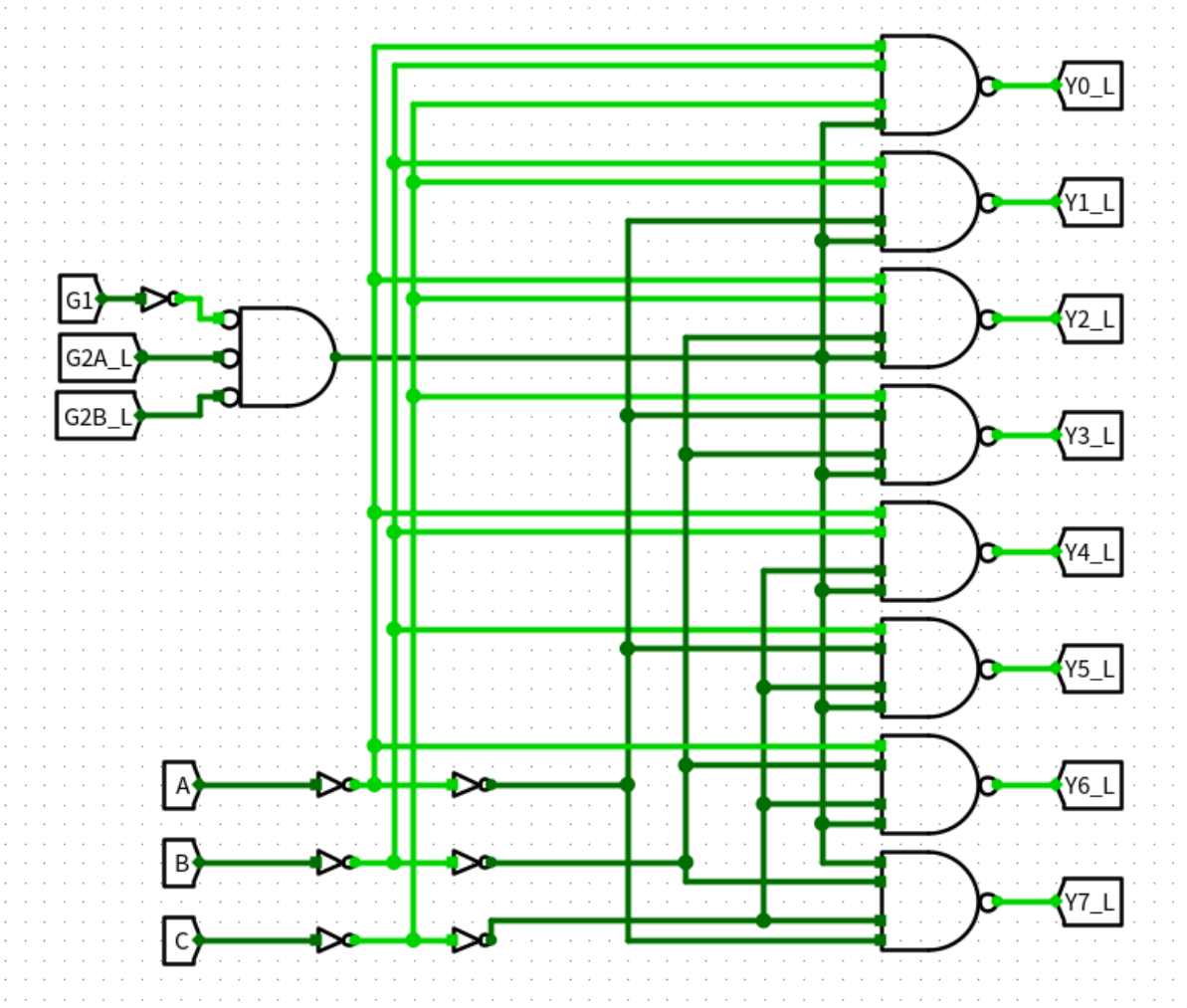
### 整体方案设计

#### 1. 输入输出引脚

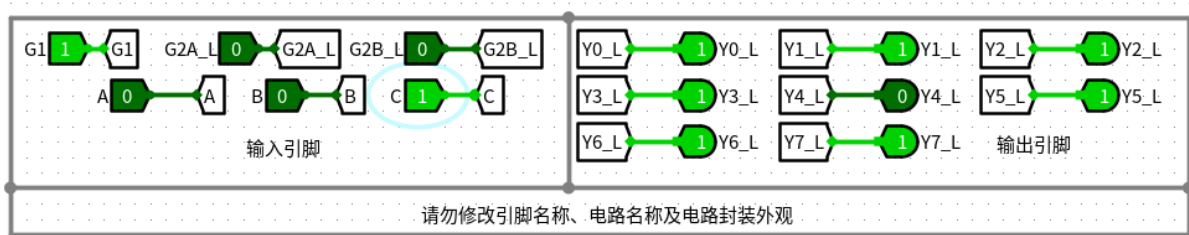


- A、B、C：表示三个输入
- Yn\_L ：输出译码结果

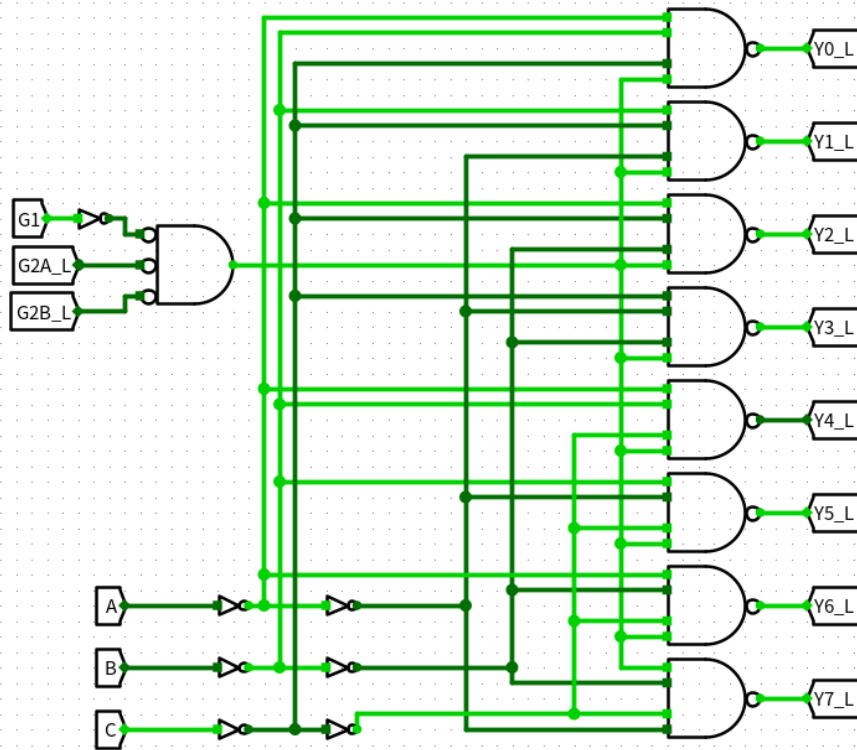
电路图 和 原理图

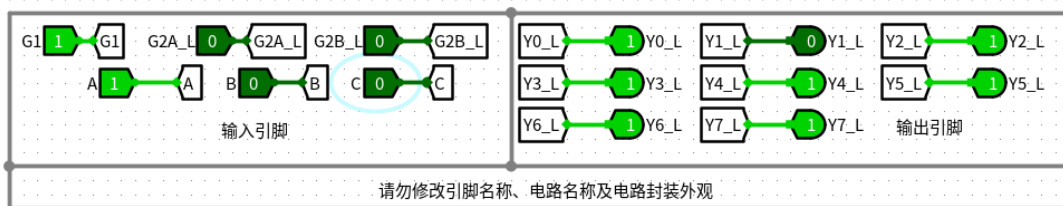


## 仿真测试

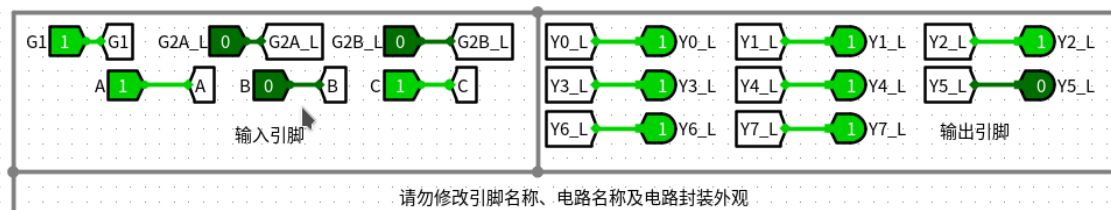
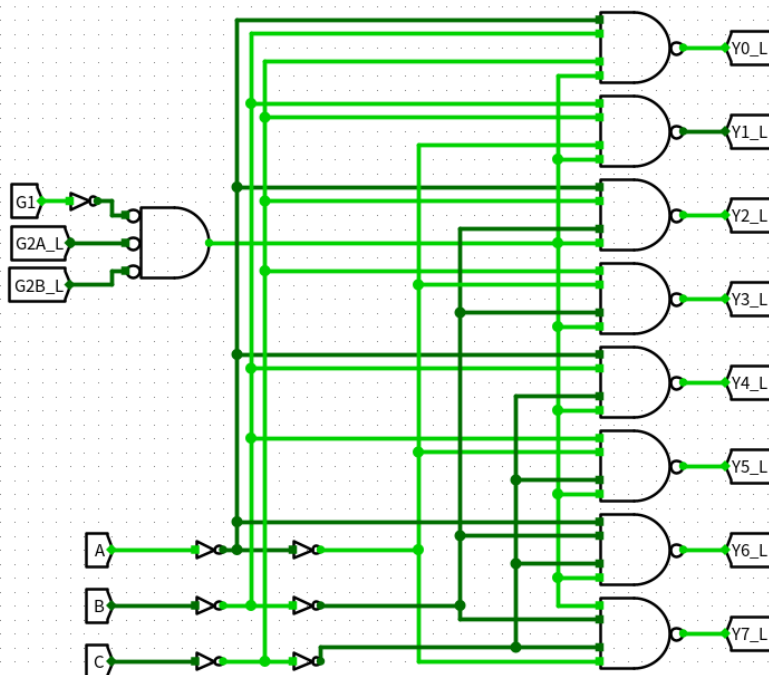


提示：使用独立逻辑门实现3-8译码器，禁止直接使用译码器组件。

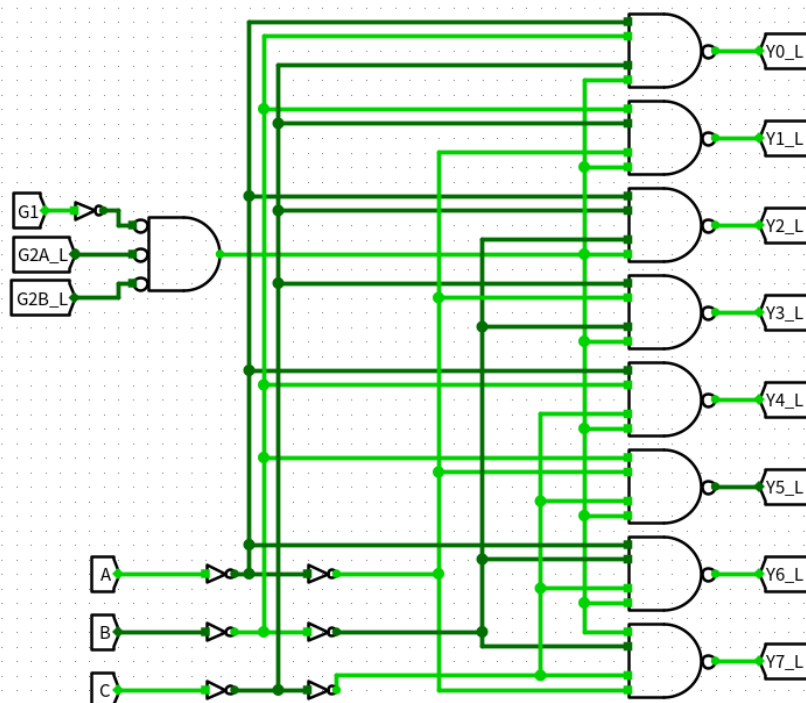




提示：使用独立逻辑门实现3-8译码器，禁止直接使用译码器组件。



提示：使用独立逻辑门实现3-8译码器，禁止直接使用译码器组件。



真值表：[真值表](#)

错误现象及分析

在完成实验过程中没有遇到任何错误。

二、编码器实验

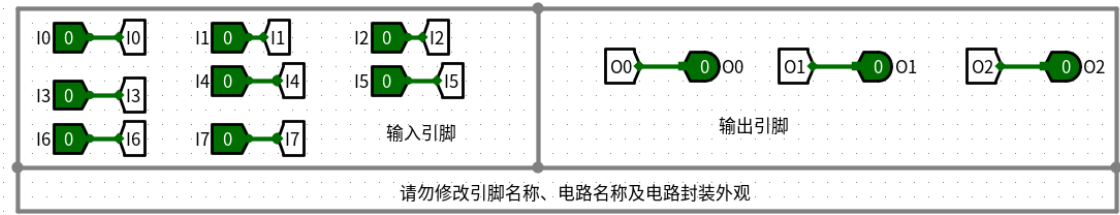
基本原理

是一种优先权编码器，使低位的输入作为高位的“使能”位  
真值表：

I <sub>0</sub>	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>	I <sub>4</sub>	I <sub>5</sub>	I <sub>6</sub>	I <sub>7</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

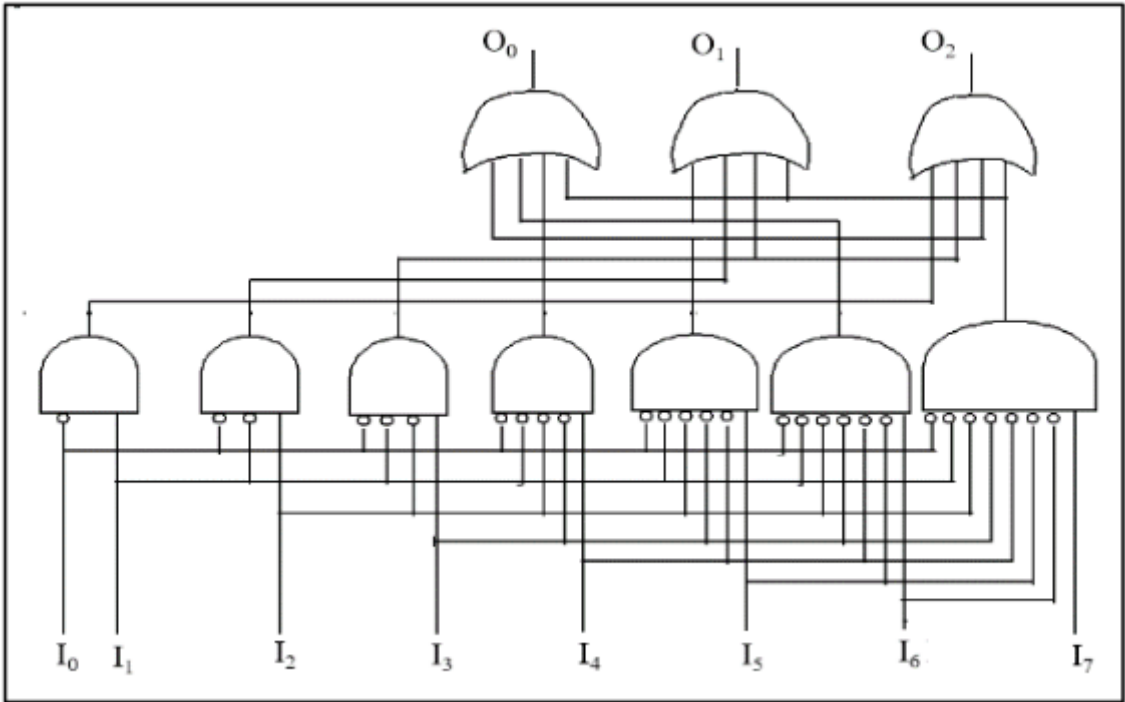
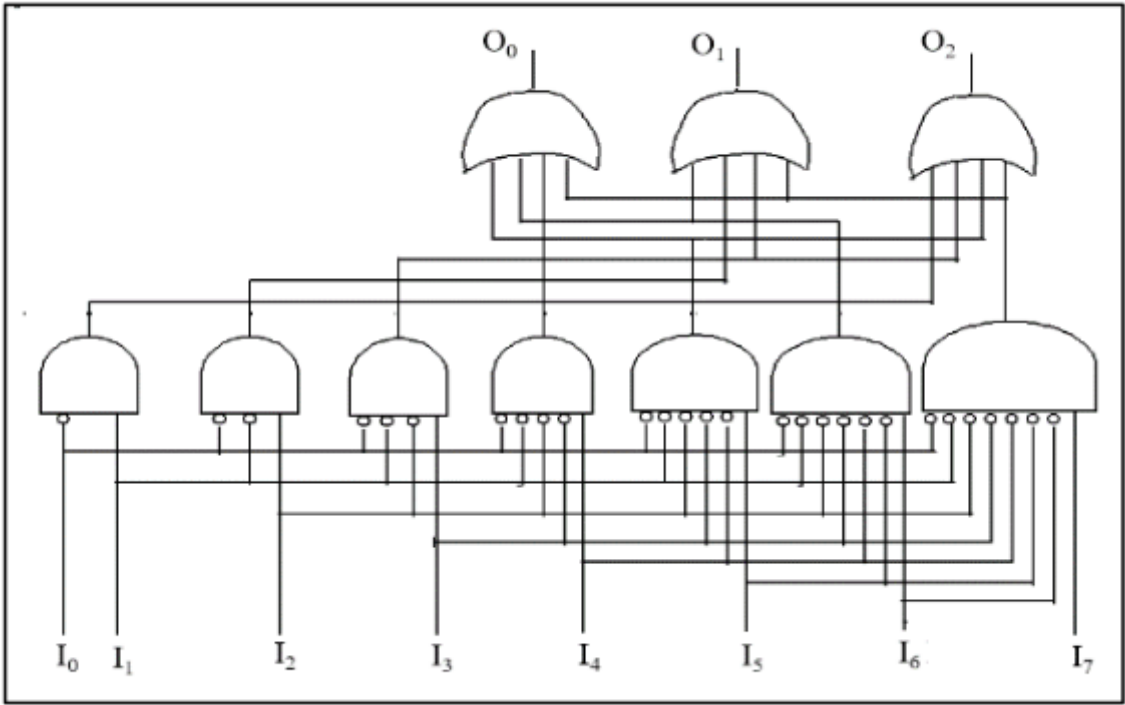
整体方案设计

1. 输入输出引脚

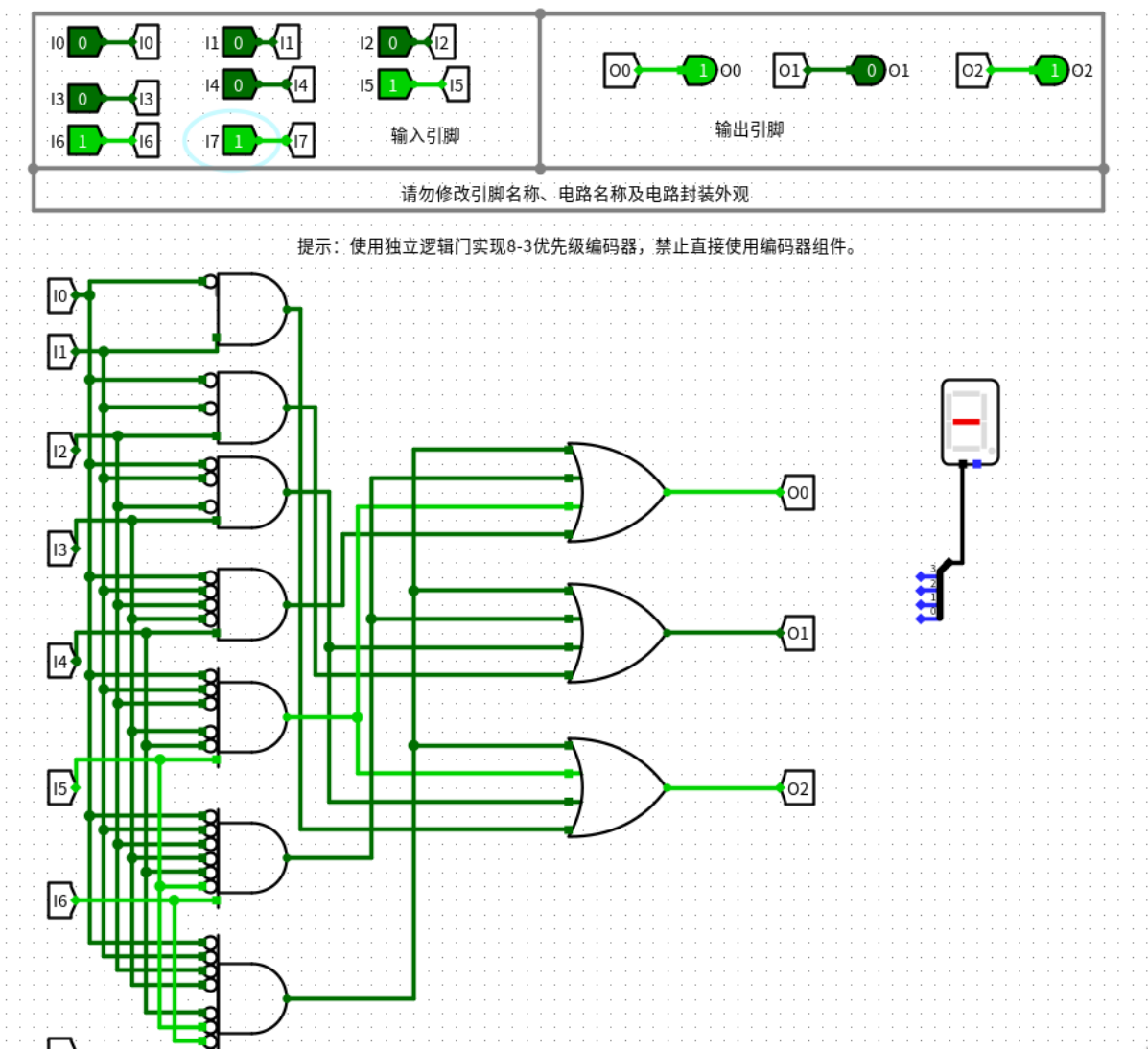


- In：表示n个编码输入
- O1 O2 O3：表示编码结果

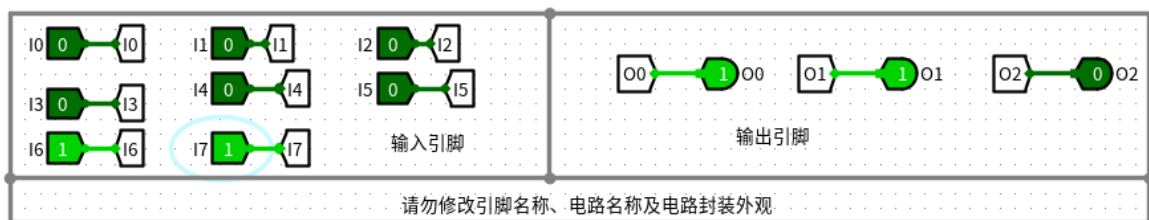
电路图



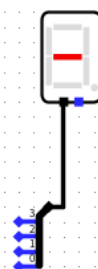
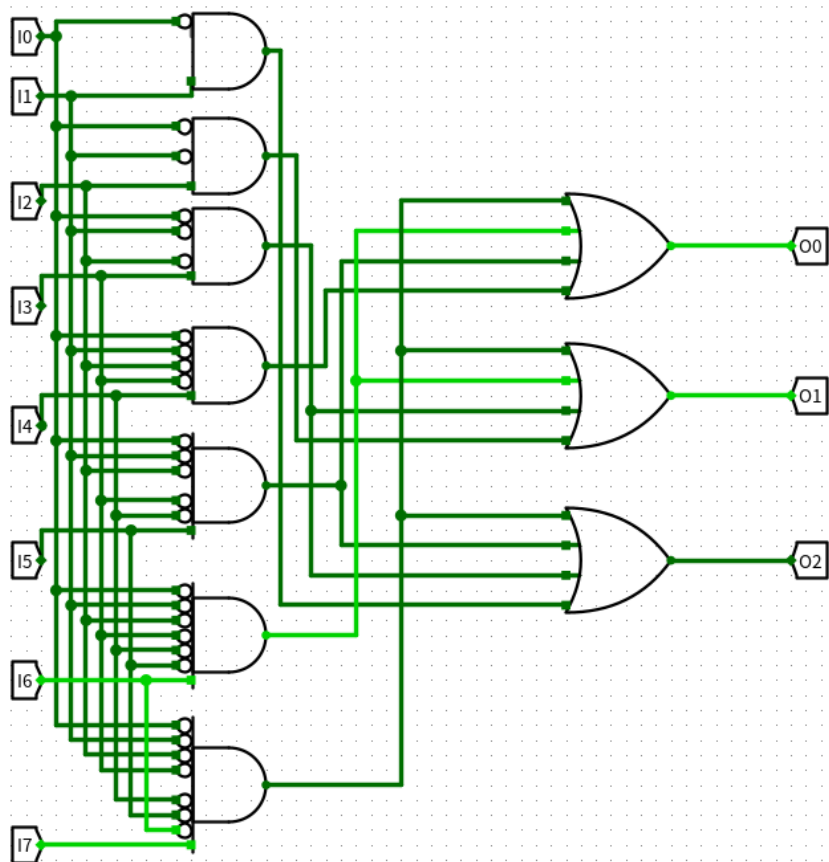
仿真测试

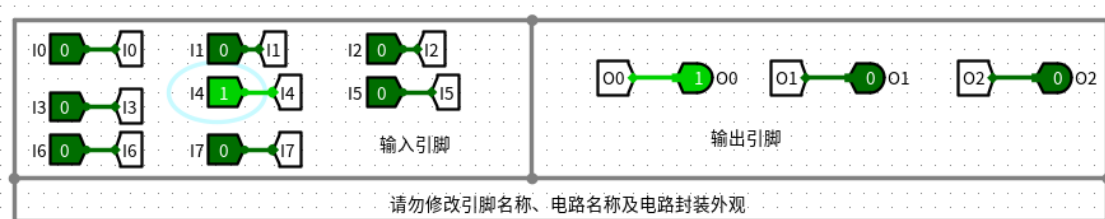




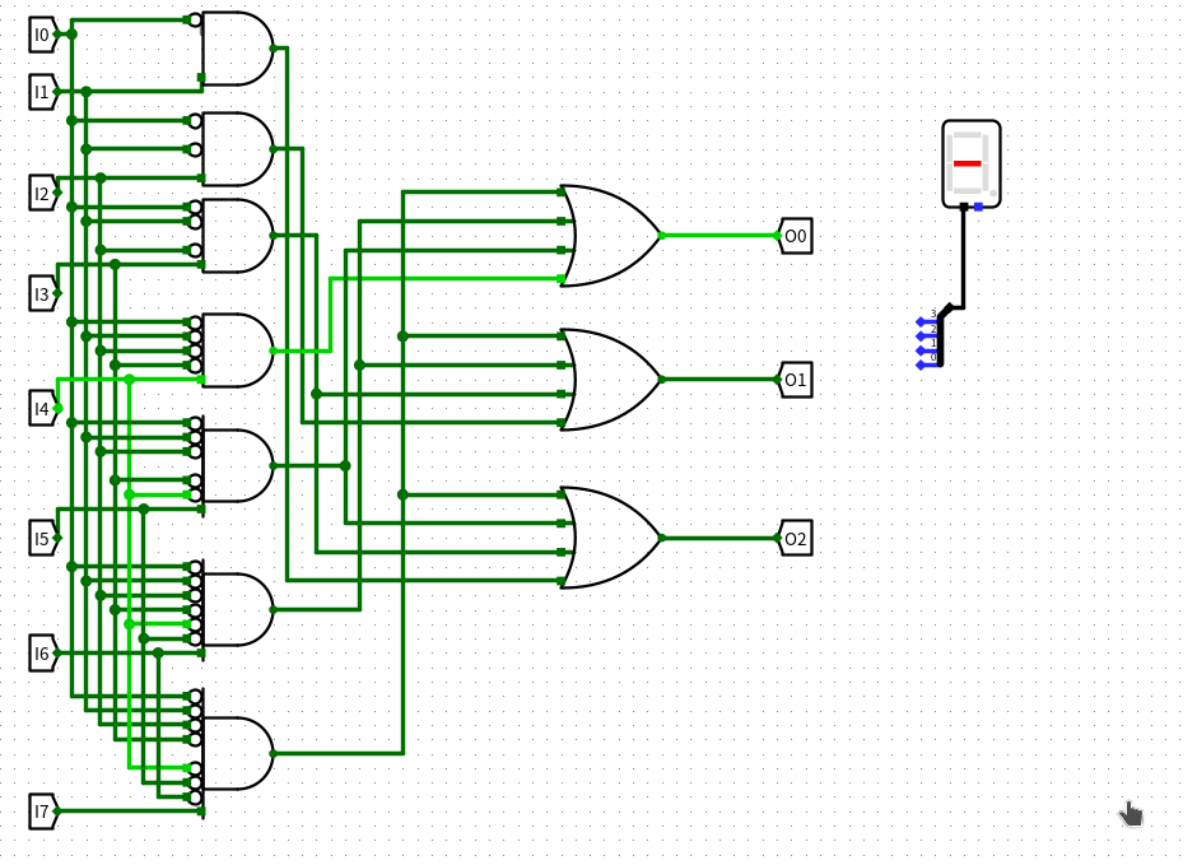


提示：使用独立逻辑门实现8-3优先级编码器，禁止直接使用编码器组件。





提示：使用独立逻辑门实现8-3优先级编码器，禁止直接使用编码器组件。



真值表：[真值表](#)

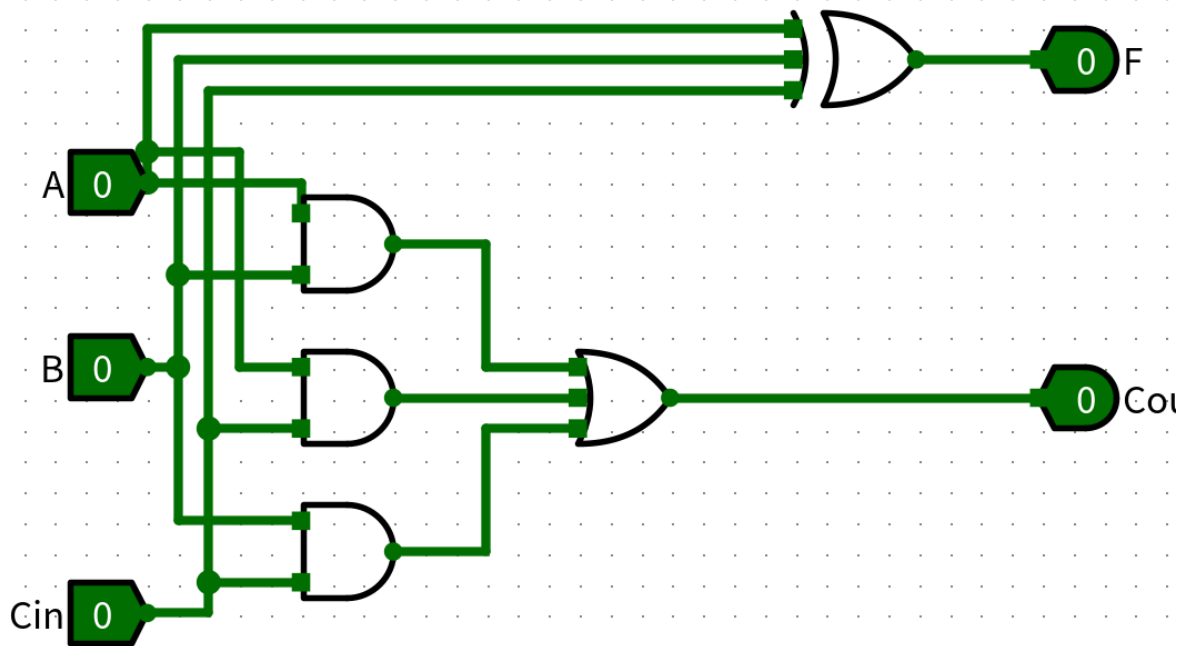
## 错误现象及分析

在完成实验过程中没有遇到任何错误。

## 三、加减法器实验

### 基本原理

全加器示意及其引脚



全加器

A	B	Cin	Sum	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

加法器原理

- 1

将全加器的进位输出（Cout）连接到下一个全加器的进位输入（Cin）。第一个全加器的进位输入（Cin）通常连接到0，表示低位没有进位。
- 2
- 3

将第一个全加器的两个输入（A和B）连接到加法操作的两个数的最低位。
- 4
- 5

最后一个全加器的和输出（Sum）以及所有中间全加器的和输出将共同构成最终的和。
- 6
- 7

最后一个全加器的进位输出（Cout）表示整个加法操作的最终进位状态。

1 将减数表示为补码表示，在与另一个数做加法运算。

整体方案设计

1. 输入输出引脚



- Y、X : 两个参与运算的数
- cin : 决定是否做减法

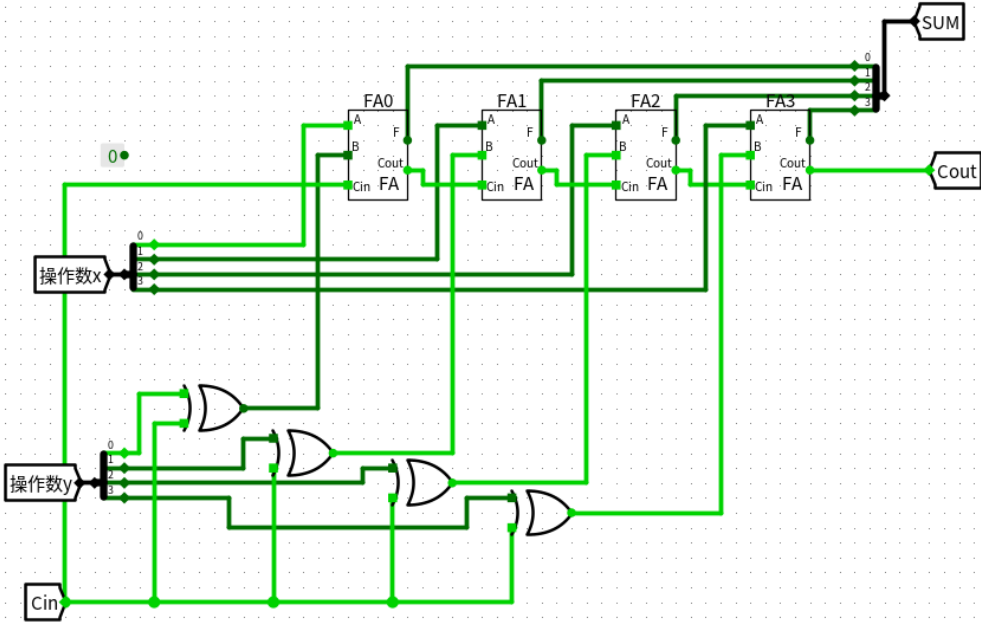
电路图



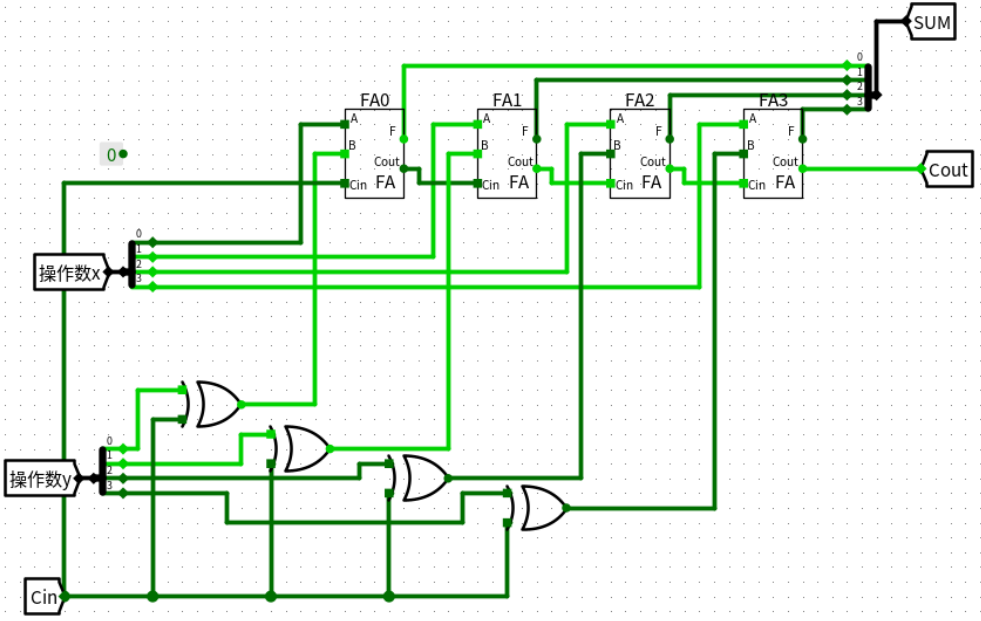
仿真测试

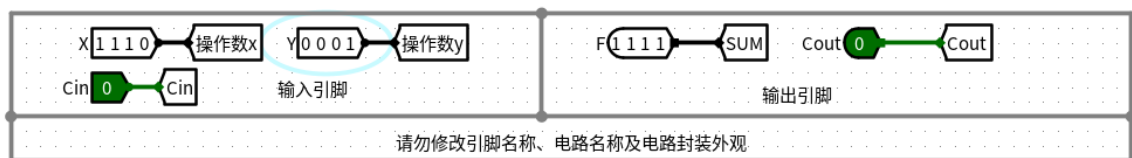


提示：使用独立逻辑门及全加器FA子电路实现4位二进制数加减法器，禁止直接使用加减法器组件。

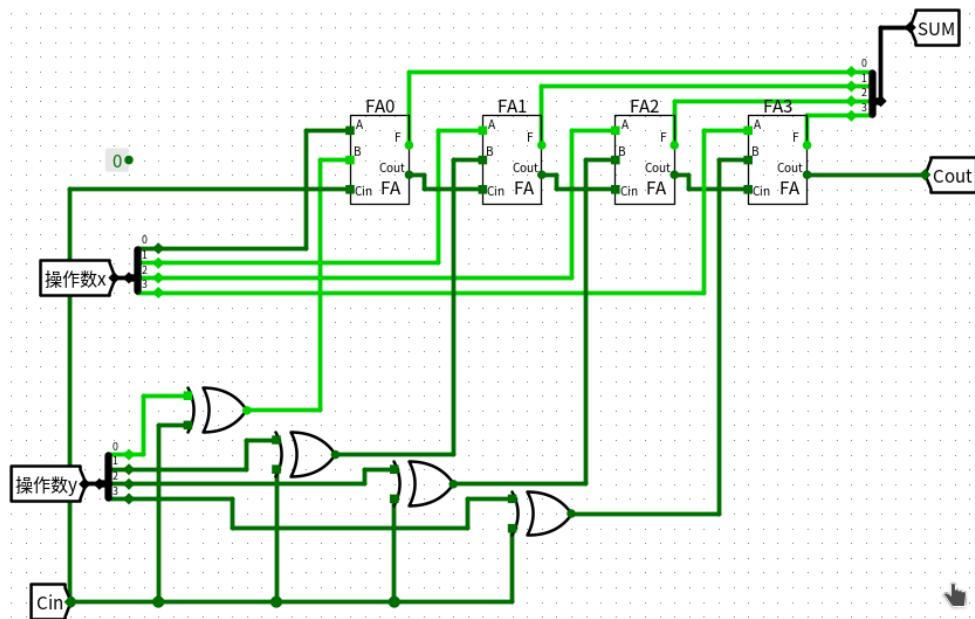
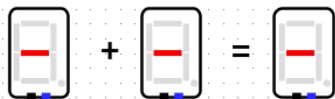


提示：使用独立逻辑门及全加器FA子电路实现4位二进制数加减法器，禁止直接使用加减法器组件。





提示：使用独立逻辑门及全加器FA子电路实现4位二进制数加减法器，禁止直接使用加减法器组件。



真值表：[真值表](#)

## 错误现象及分析

在完成实验过程中没有遇到任何错误。

## 四、汉明码校验电路

### 基本原理

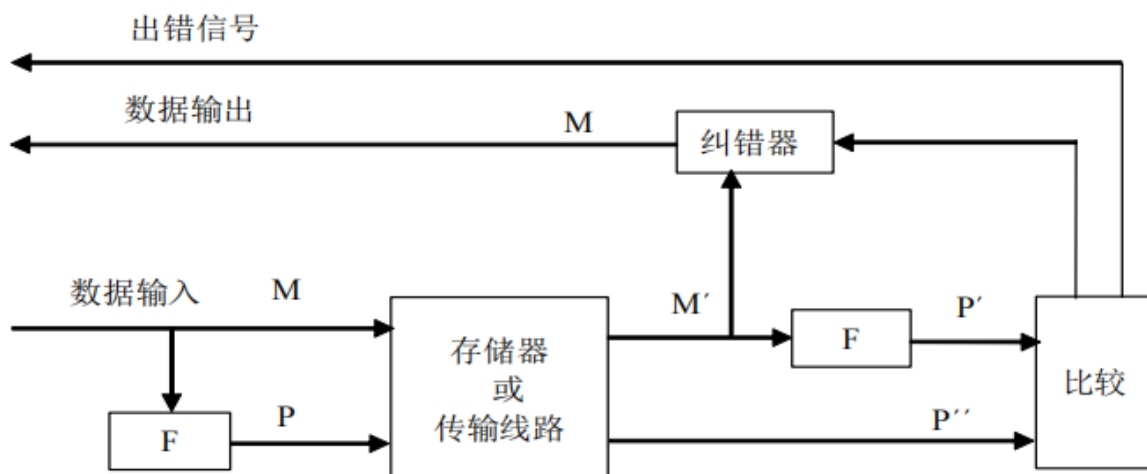
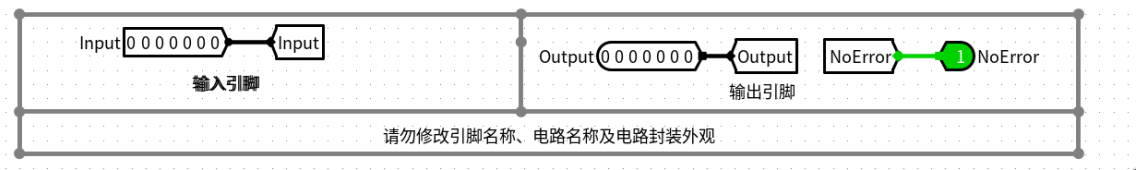


图 11.42 数据校验过程示意图

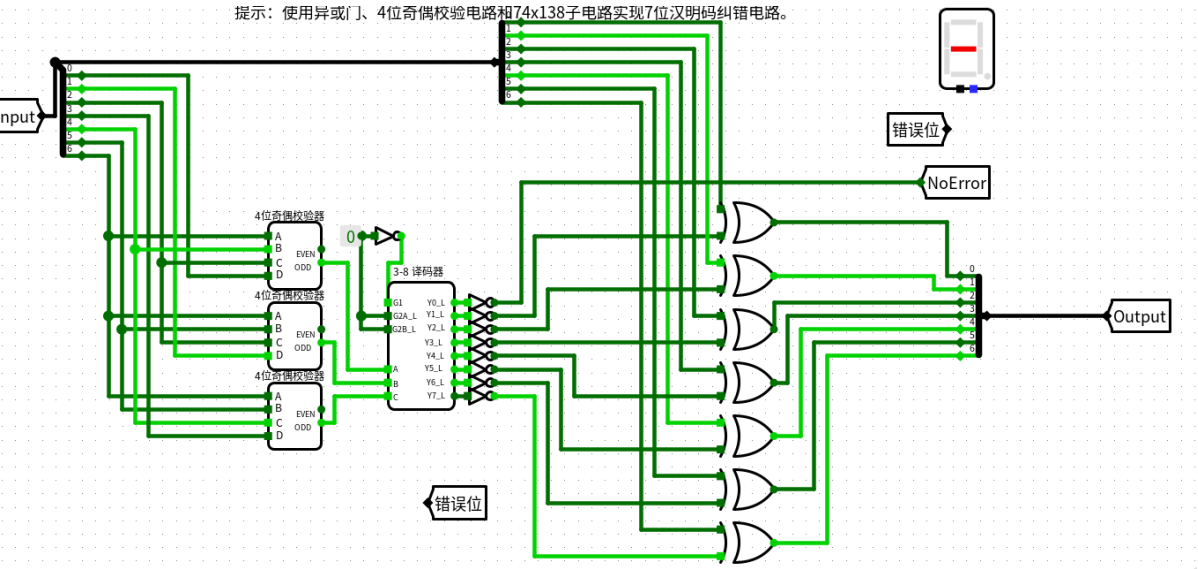
# 整体方案设计

## 1. 输入输出引脚

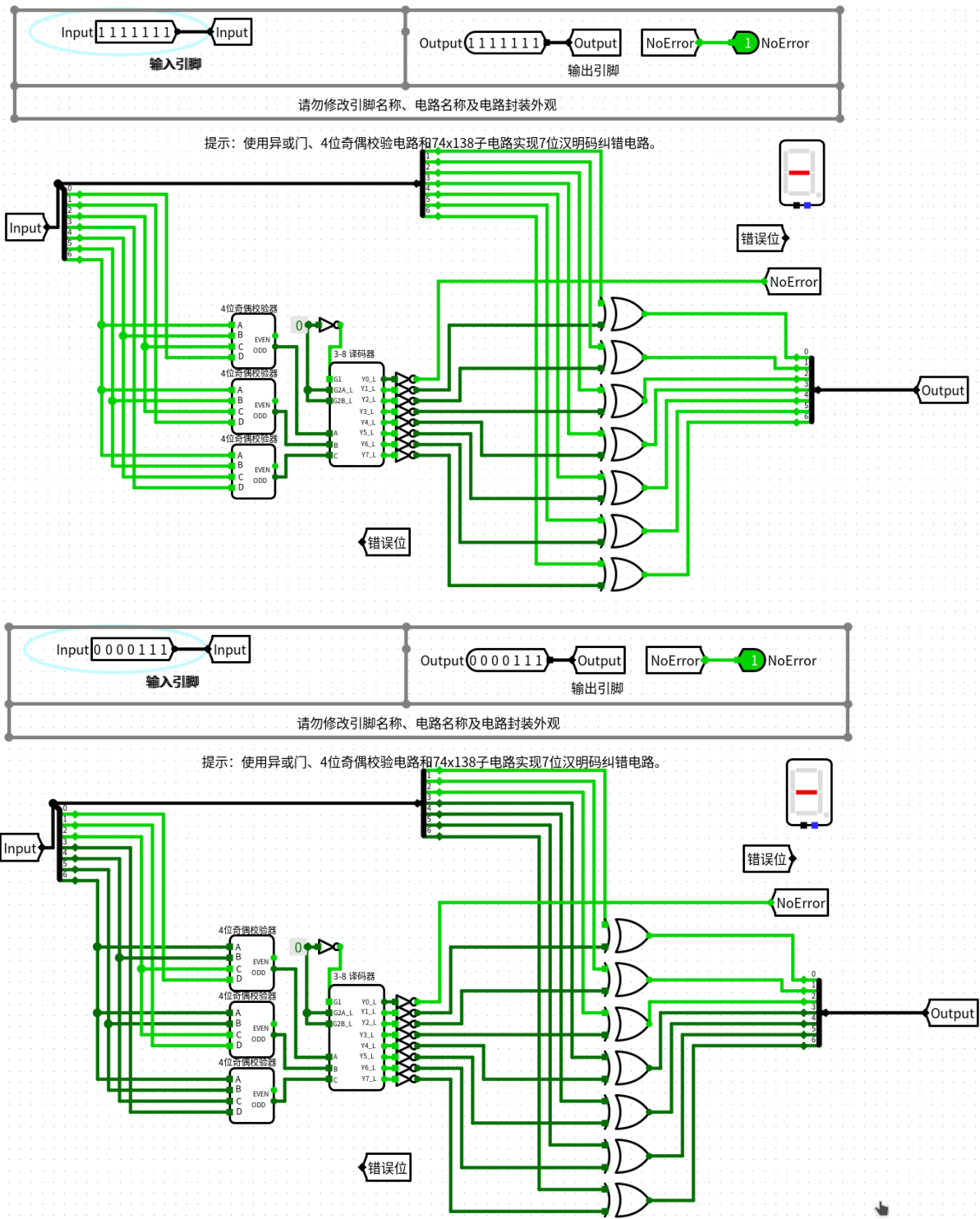


- Input : 7 位输入码
- Output : 7 位输出码

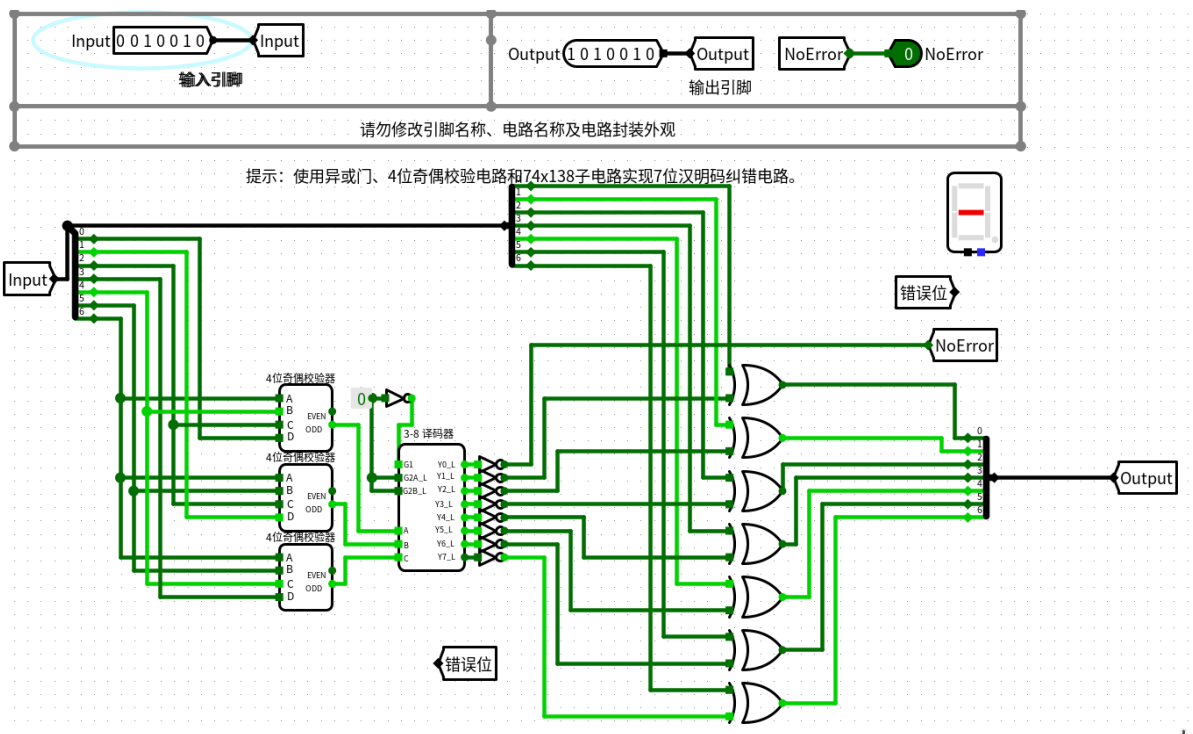
## 电路图



仿真测试







## 错误现象及分析

在完成实验过程中没有遇到任何错误。

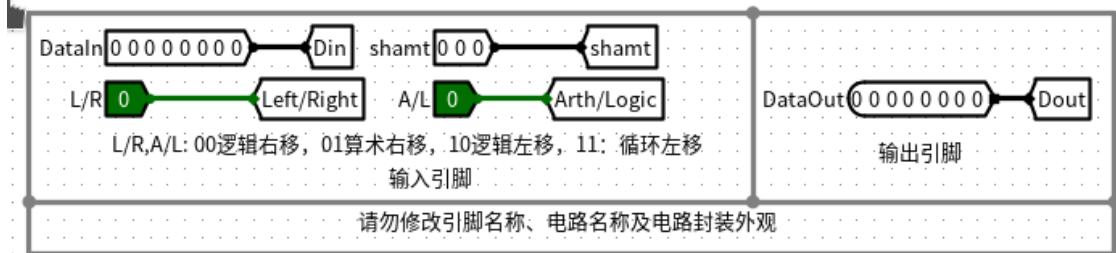
## 五、桶型移位器

### 基本原理

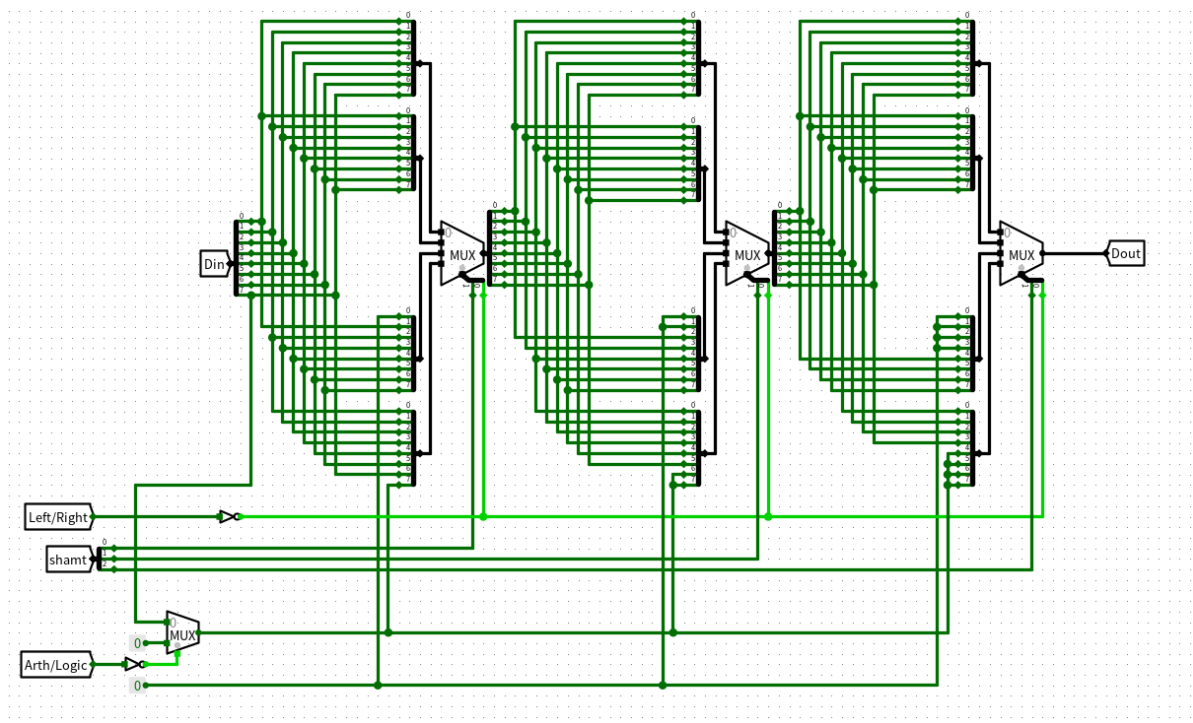
- 1 在移位操作之前，先将被移位数的最高位（对于左移是左边第一个位，对于右移是右边第一个位）暂存起来。
- 2 将被移位数的其余位按照指定的方向（左移或右移）进行移动指定的位数。
- 3 将暂存的最高位移到移位后的最高位位置，而新产生的最低位则用0或特定的值（如算术移位时用符号位填充）填充。
- 4 将暂存的最高位和移位后的数合并，形成最终的移位结果。

### 整体方案设计

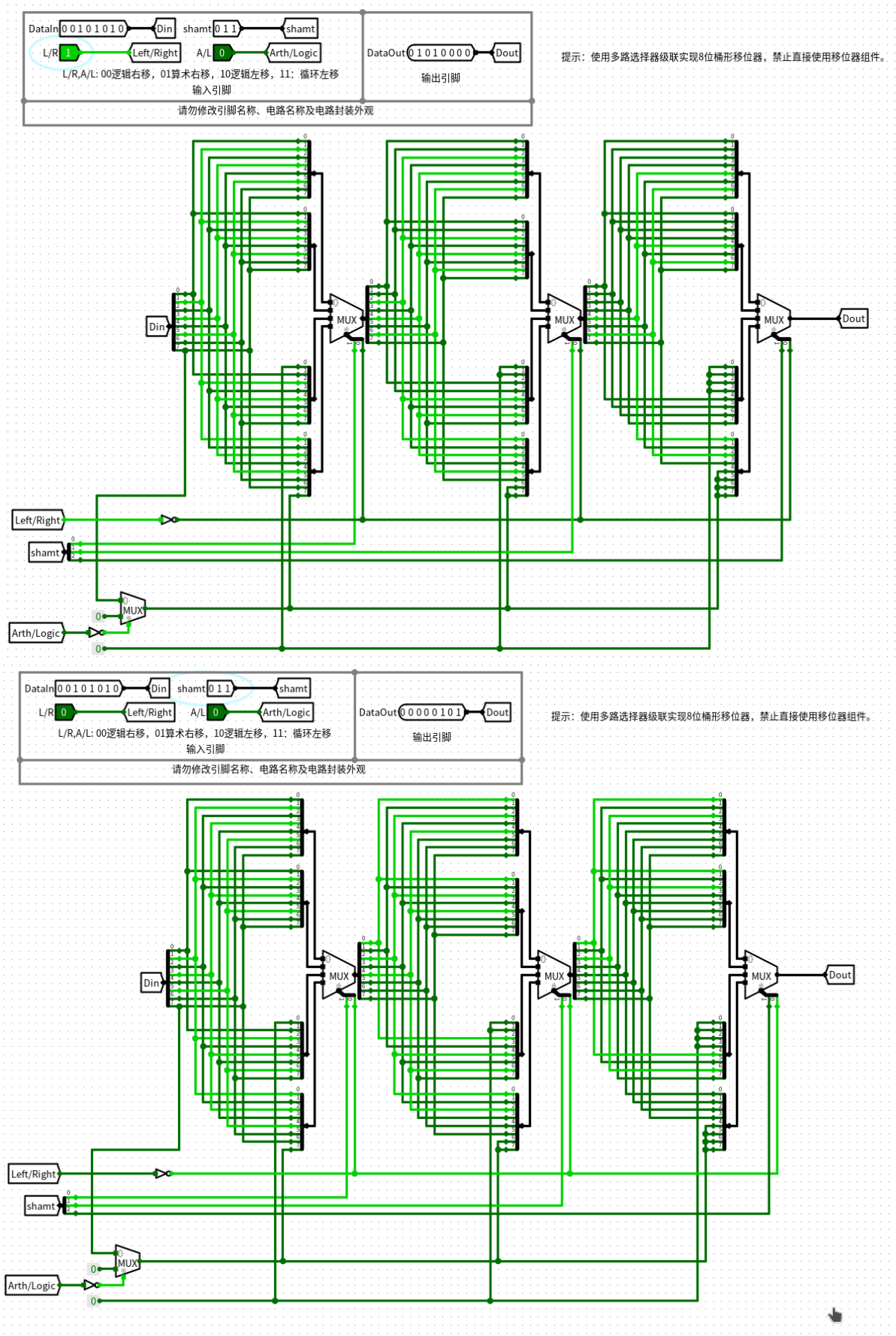
#### 1. 输入输出引脚

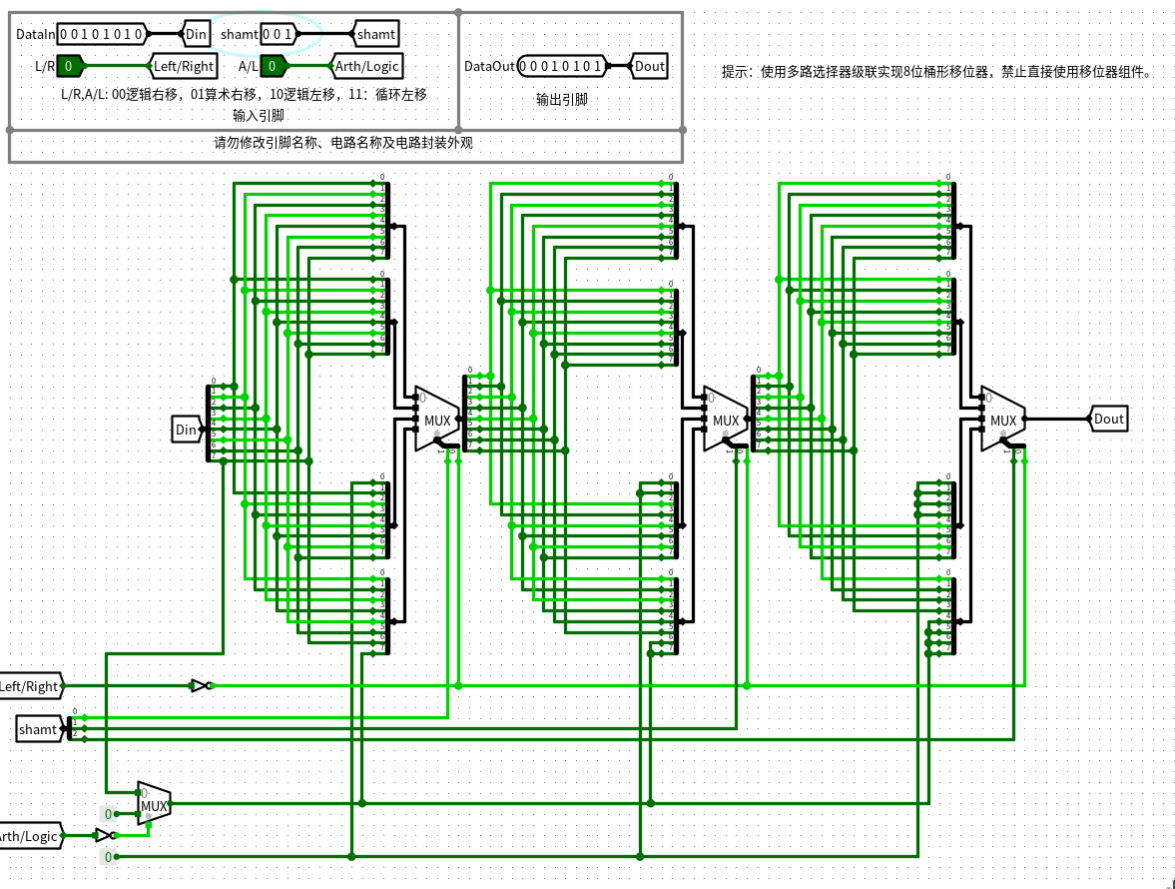


电路图



## 仿真测试



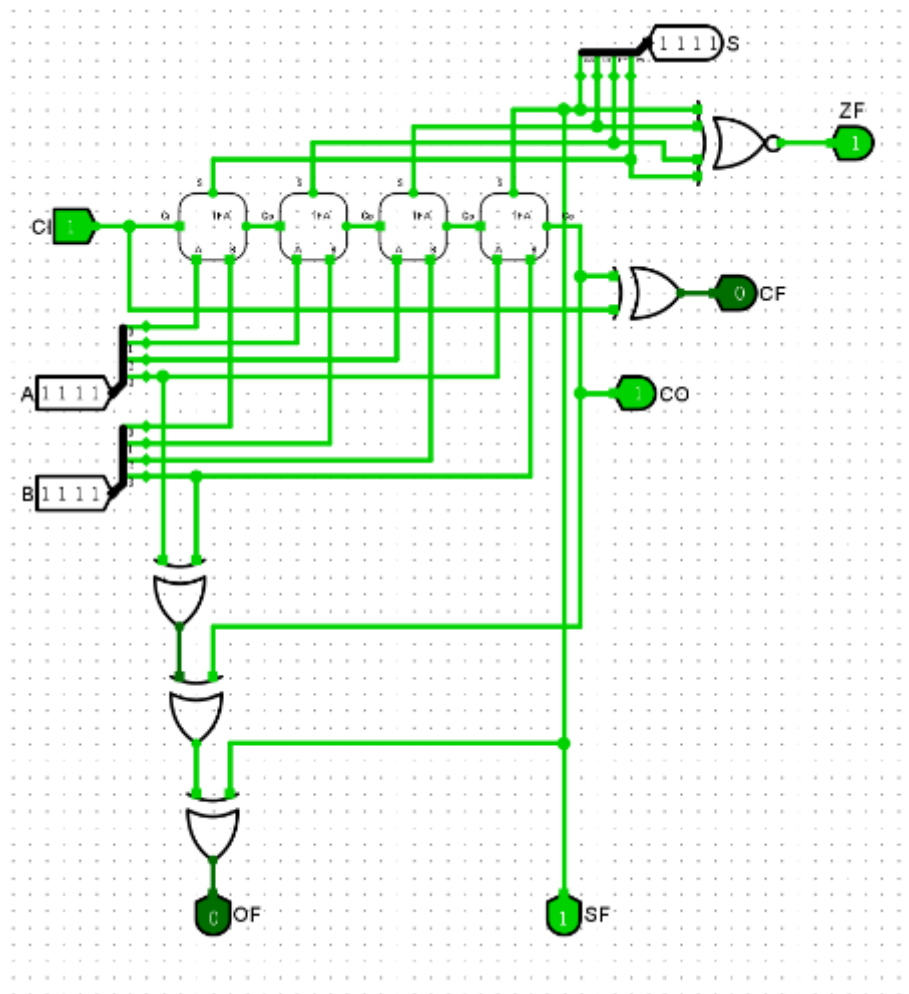


## 错误现象及分析

在完成实验过程中没有遇到任何错误。

## 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。



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。

1. 有符号数比较大小:

用  $(SF \wedge OF)$  与 ZF 来判断大小。ZF=0 时等大，ZF 不等于 0 时， $(SF \wedge OF)$  为 1 则被减数小于减数，反之则反

2. 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。

