

## 实验 2：组合逻辑电路设计

### 一、实验目的：

1. 学习组合逻辑电路的设计方法和步骤，掌握译码器，编码器等组合部件的设计方法。
2. 掌握串行加法器设计方法，理解减法和比较运算的实现方法。
3. 掌握汉明码校验电路的设计方法。
4. 掌握桶形移位器的设计方法。

### 二、实验环境

Logisim 2.16

### 三、实验内容

#### 1、译码器实验

整体模块设计：

使用 8 个 4 输入与非门，7 个非门，1 个与门，7 个输入引脚，8 个输出引脚来实现拥有 3 个使能控制端的输出端低电平有效的 3-8 译码器。

输入输出引脚作用：

G1, G2A\_L, G2B\_L, 为三个使能控制端，A, B, C 为三个输入端，从高到低位是 CBA，代表了一个二进制数，Y0\_L, Y1\_L, ……; Y7\_L 为输出端，代表了十进制数。

原理图：

$$EN = \overline{G1} + G2A\_L + G2B\_L$$

**例：Y6\_L = EN +  $\bar{C}$  +  $\bar{B}$  + A**

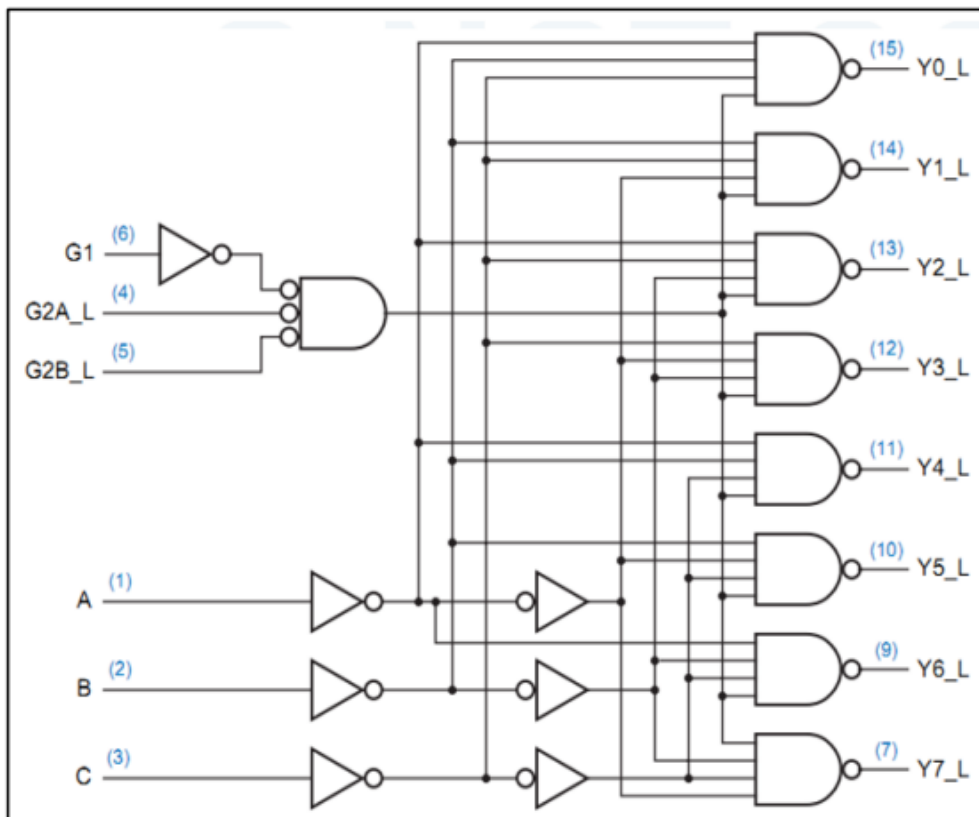
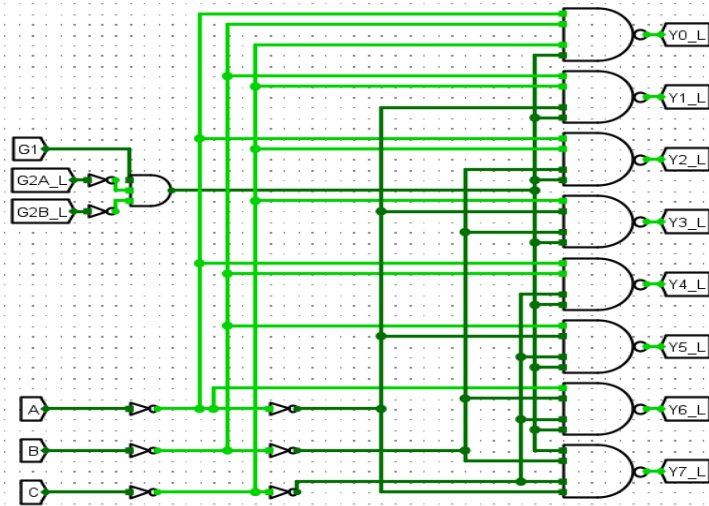
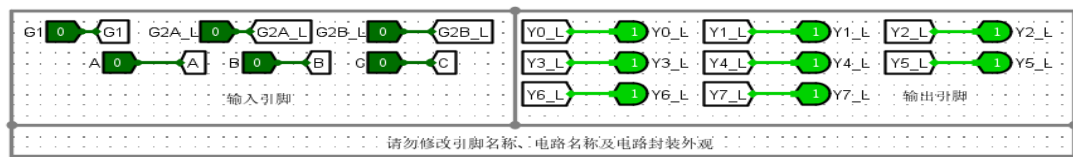
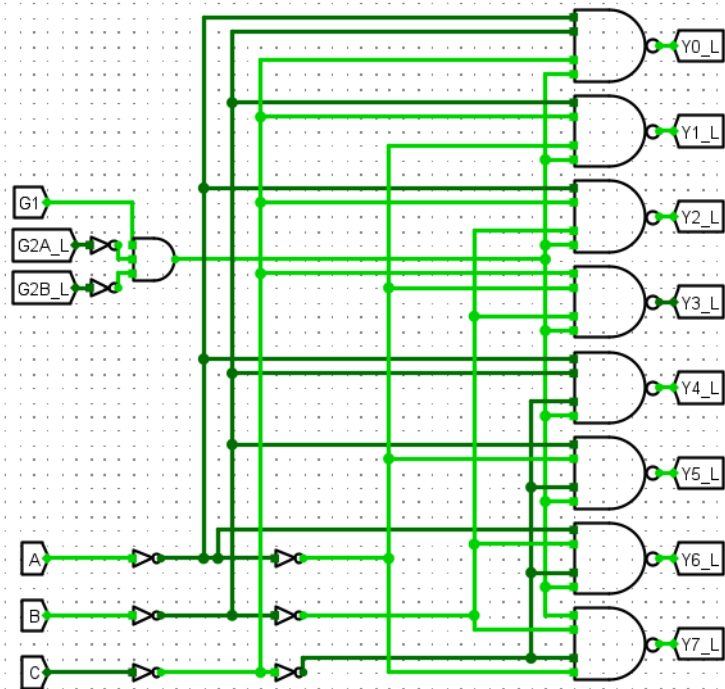
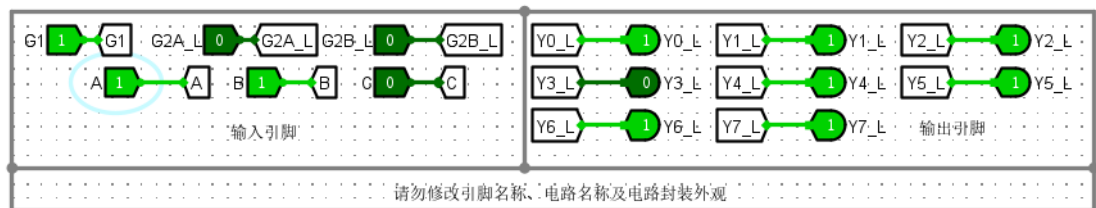


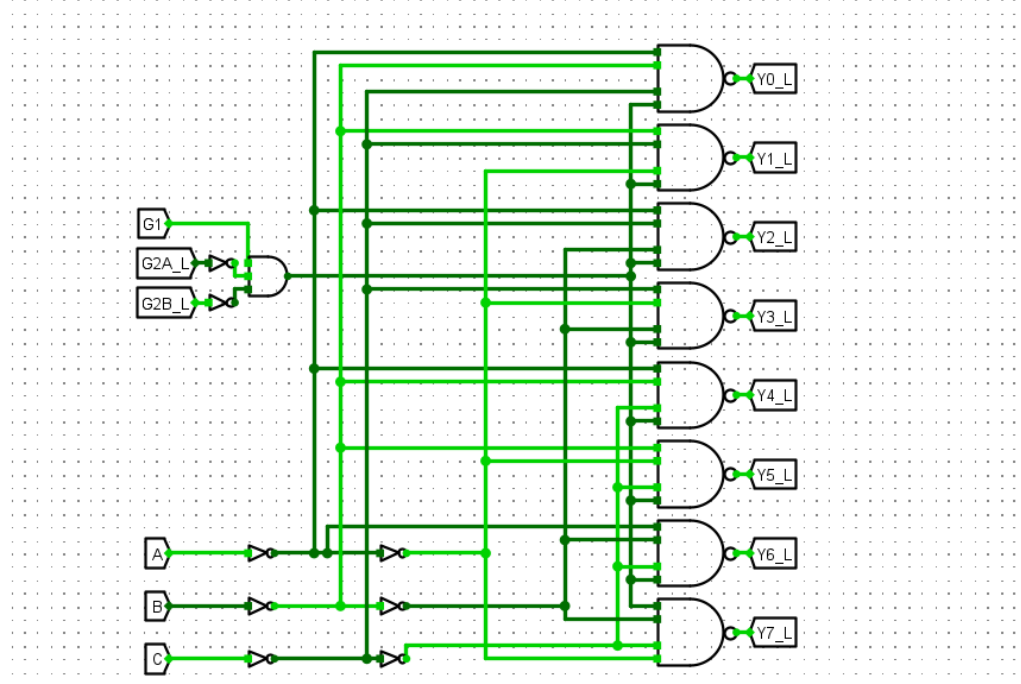
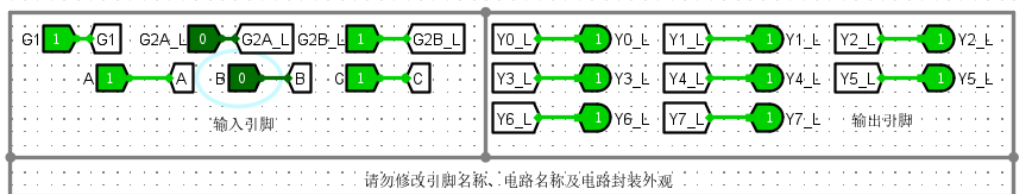
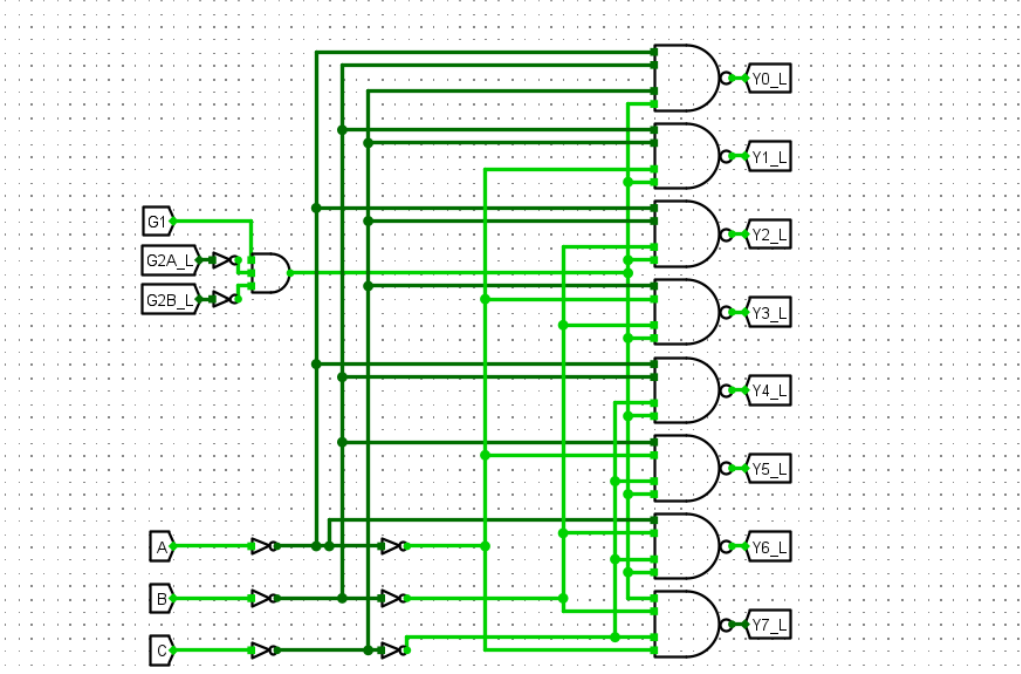
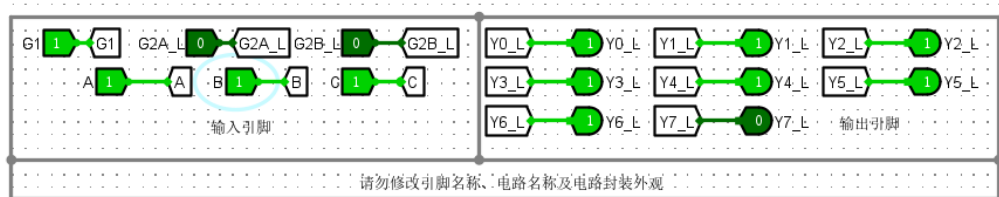
图 2.1 3-8 译码器 74X138 原理图

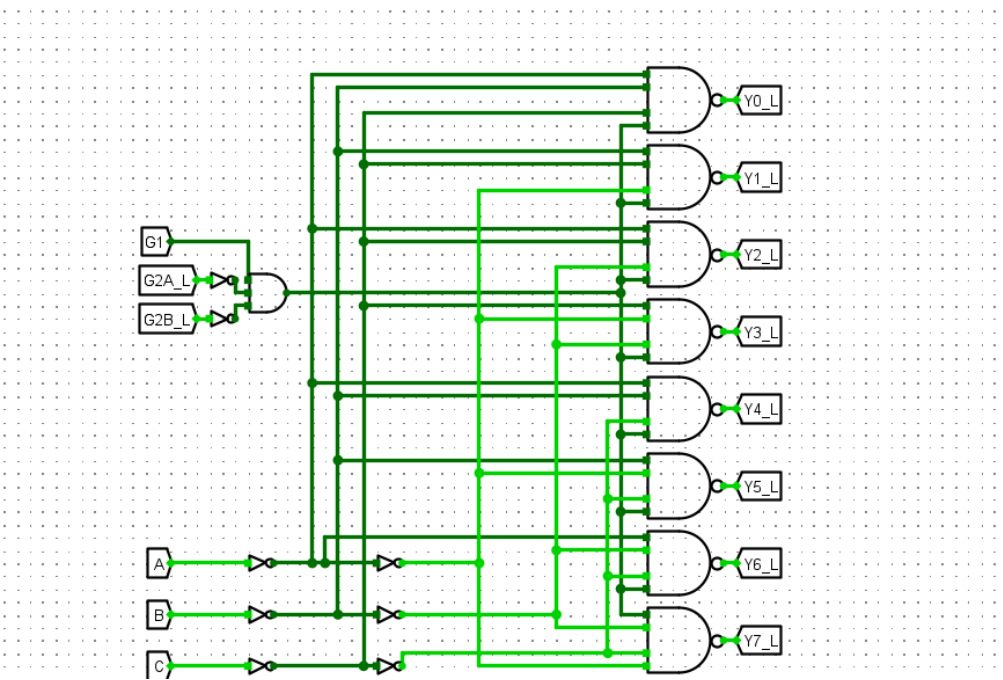
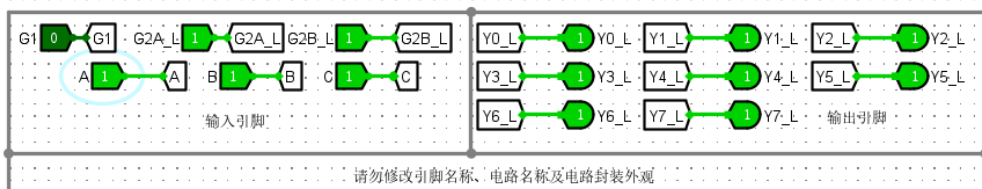
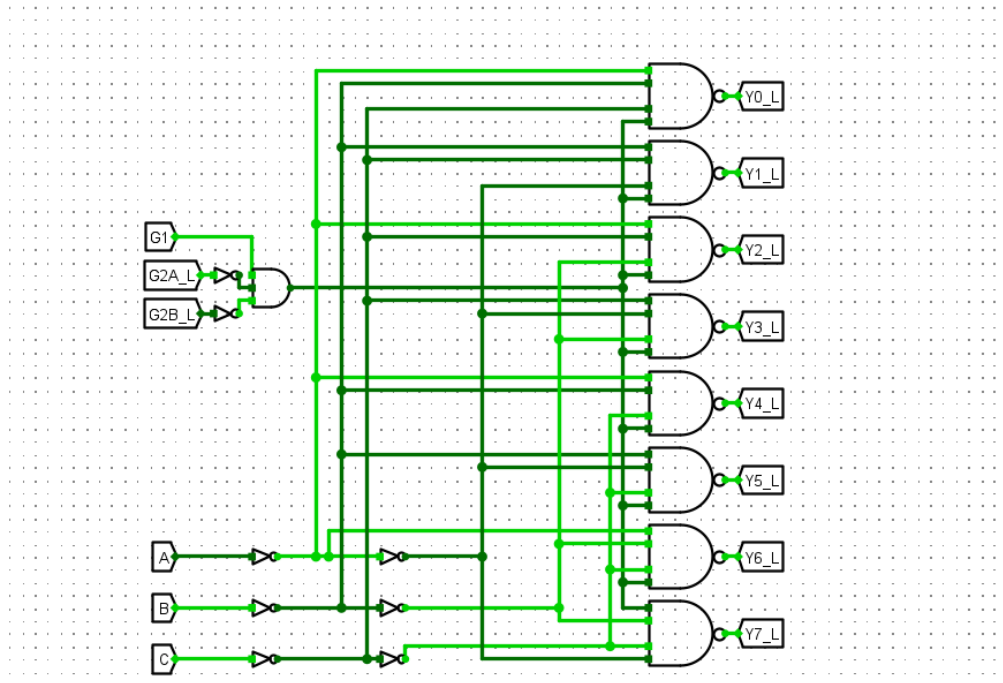
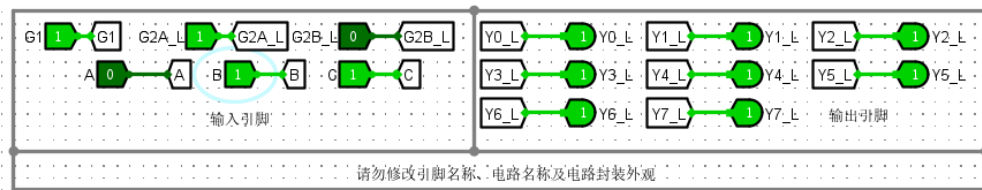
电路图：



仿真测试图：







真值表：

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

## 2、编码器实验

整体模块设计：

运用 3 个异或门，8 个多输入与门实现 8-3 优先级编码器，通过 1 个十六进制数码管显示并测试电路。

输入输出引脚作用：

I<sub>0</sub>, I<sub>1</sub>, ..., I<sub>7</sub> 为输入端，代表一个十进制数，O<sub>0</sub>, O<sub>1</sub>, O<sub>2</sub> 为输出端，代表一个二进制数。

原理图：

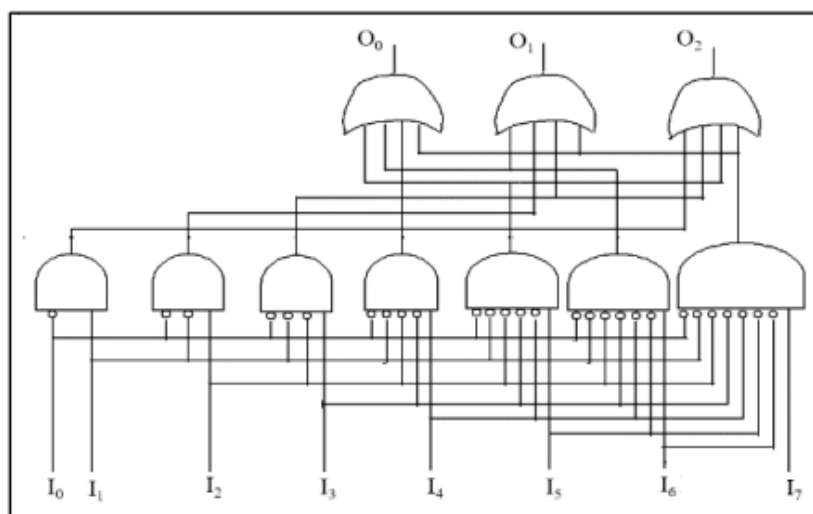
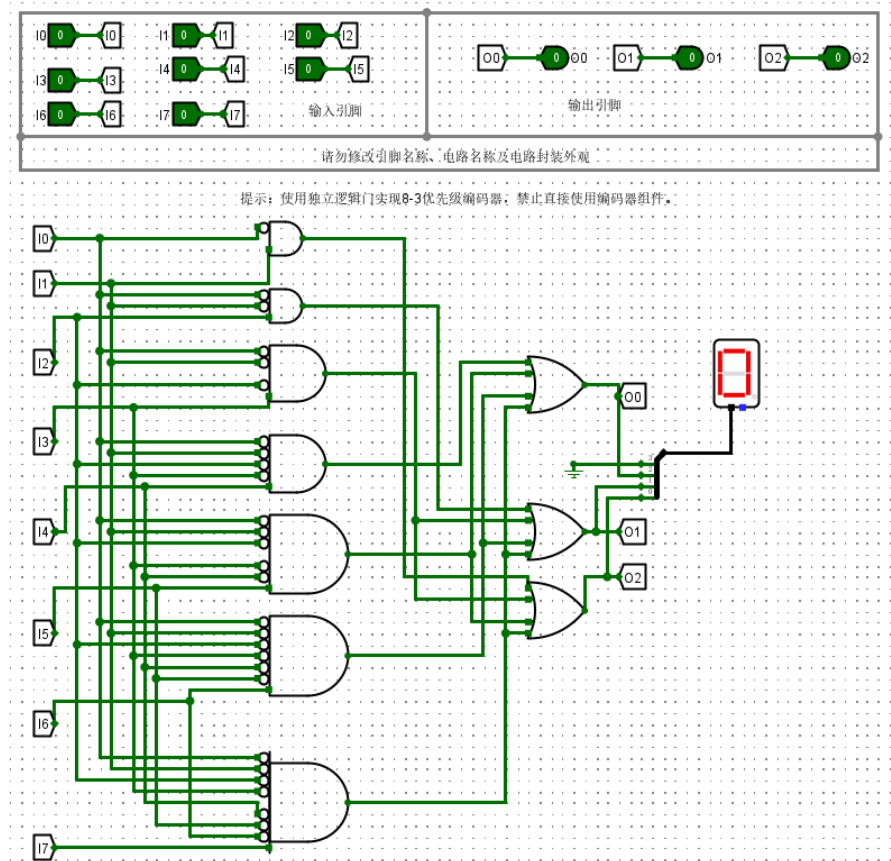
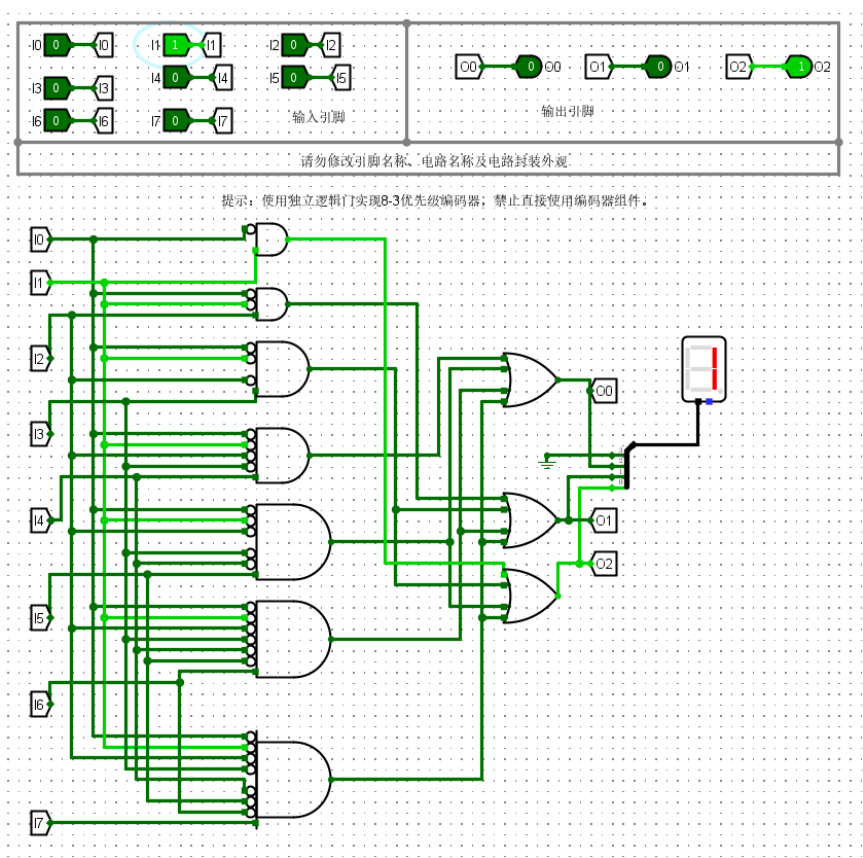


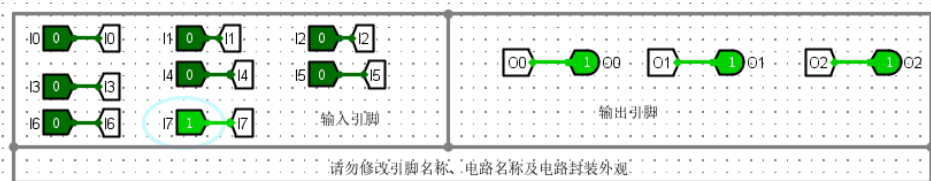
图 2.3 8-3 优先级编码器原理图

电路图：

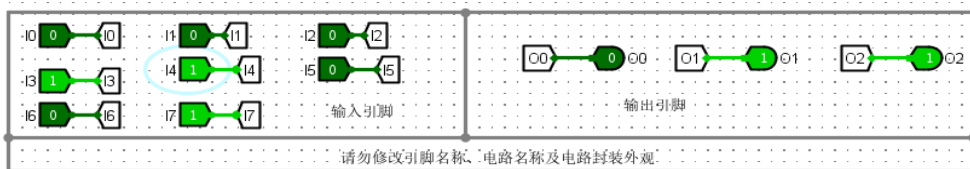
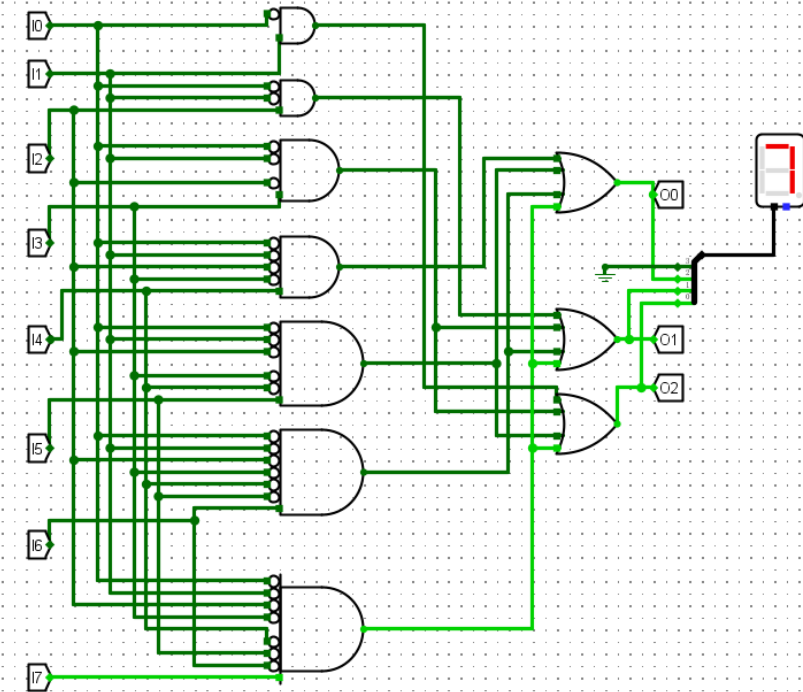


仿真测试图：

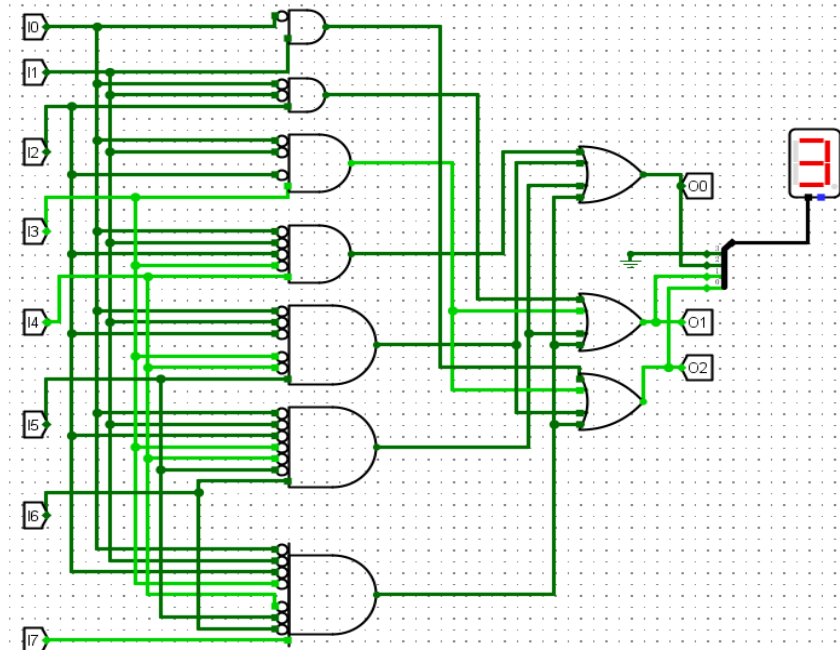


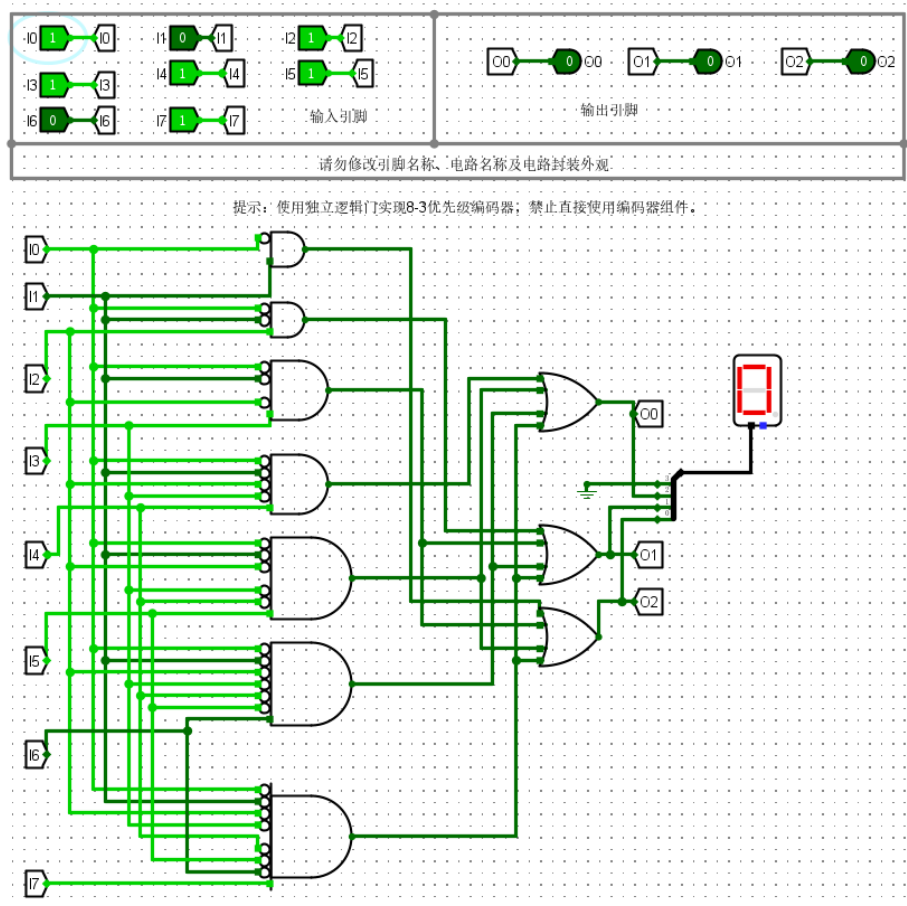


提示：使用独立逻辑门实现8-3优先级编码器；禁止直接使用编码器组件。



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：

| I0 | I1 | I2 | I3 | I4 | I5 | I6 | I7 | O0 | O1 | O2 | Hex显示 |
|----|----|----|----|----|----|----|----|----|----|----|-------|
| 1  | x  | x  | x  | x  | x  | x  | x  | 0  | 0  | 0  | 0     |
| 0  | 1  | x  | x  | x  | x  | x  | x  | 0  | 0  | 1  | 1     |
| 0  | 0  | 1  | x  | x  | x  | x  | x  | 0  | 1  | 0  | 2     |
| 0  | 0  | 0  | 1  | x  | x  | x  | x  | 0  | 1  | 1  | 3     |
| 0  | 0  | 0  | 0  | 1  | x  | x  | x  | 1  | 0  | 0  | 4     |
| 0  | 0  | 0  | 0  | 0  | 1  | x  | x  | 1  | 0  | 1  | 5     |
| 0  | 0  | 0  | 0  | 0  | 0  | 1  | x  | 1  | 1  | 0  | 6     |
| 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 7     |

### 3、加减法器实验

整体模块设计：

首先实现子电路全加器 FA，通过逻辑电路分别实现结果 F 和产生的进位 Cout。然后通过串联 4 个全加器子电路来实现 4 位串行进位加法器，将加数，被加数和最终的和分别连接到 16 进制数码显示管进行验证。在实现加法器的基础上，减法器则通过 Cin 的输入值来区分加减，Cin=0 时，执行补码加法 运算  $F=X+Y$ ；当 Cin=1 时，执行补码减法运算  $F=X-Y$ 。

输入输出引脚作用：

隧道操作数 x 代表被加数，隧道操作数 y 表示加数，Cin 为先前产生的进位（若为减法器，则是控制加减法的控制器），Cout 为最终加法结果产生的进位，Sum 为全加器最终的和的结果。



原理图：

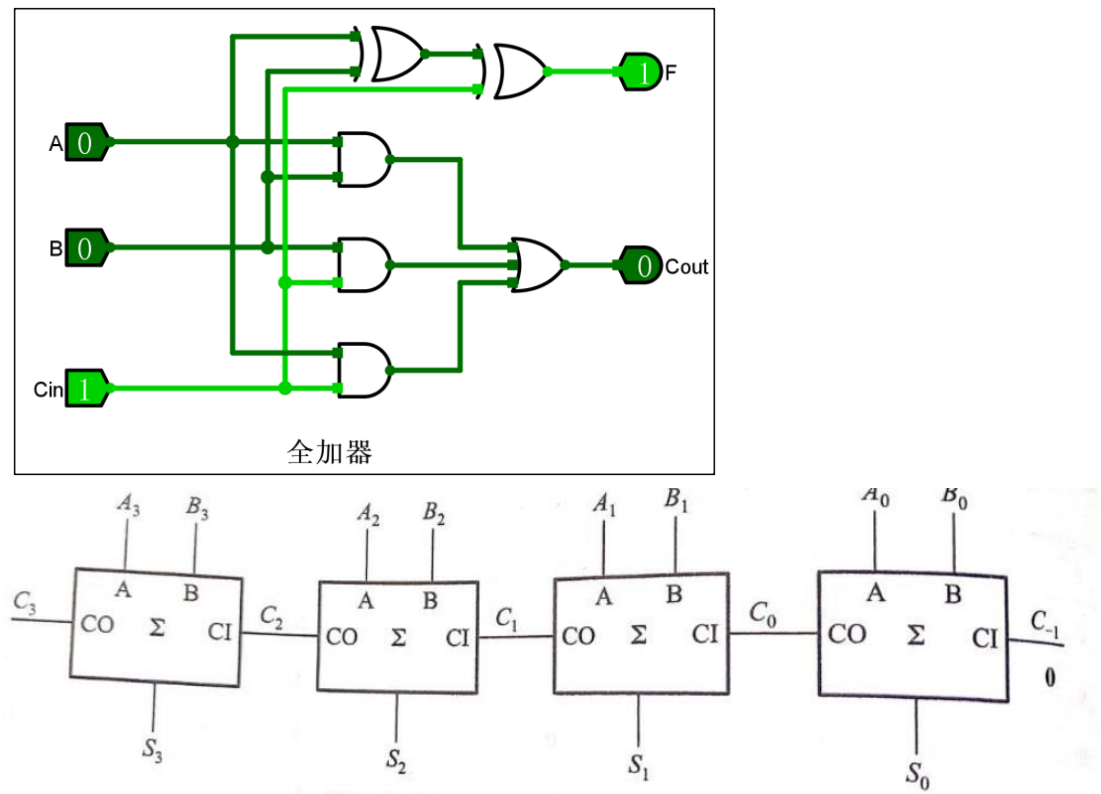
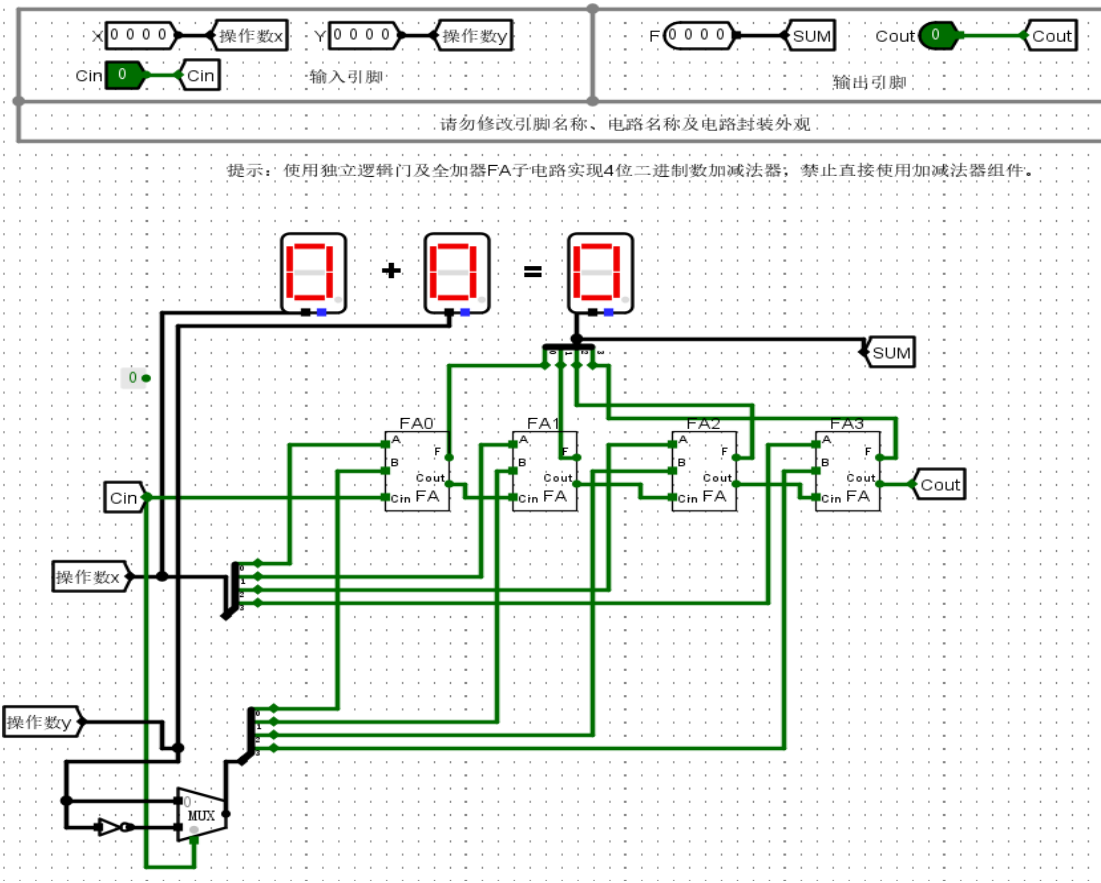


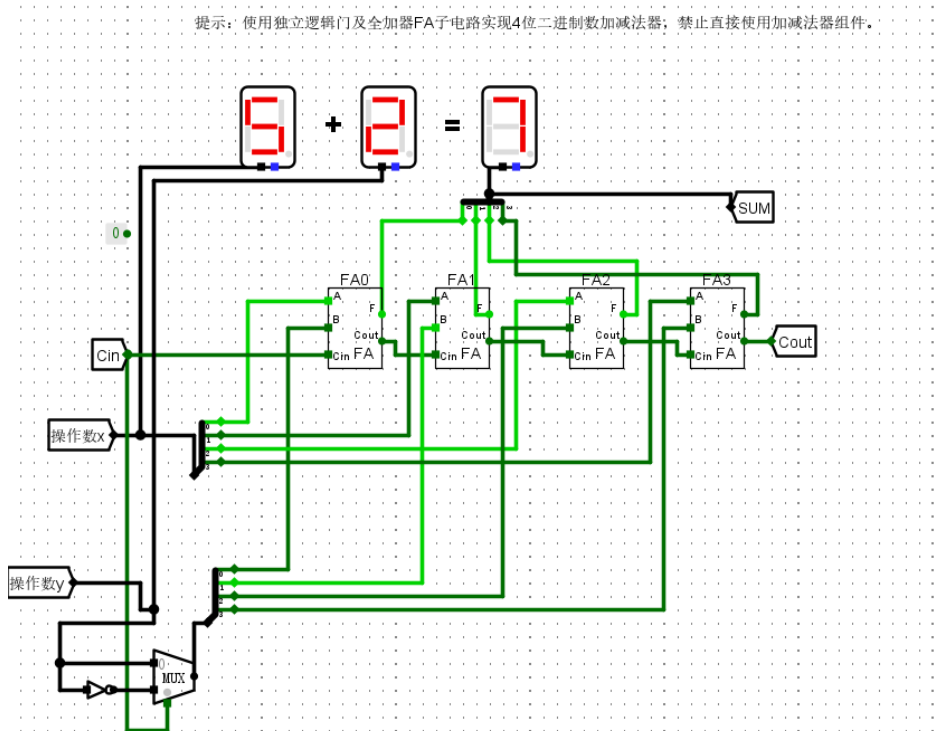
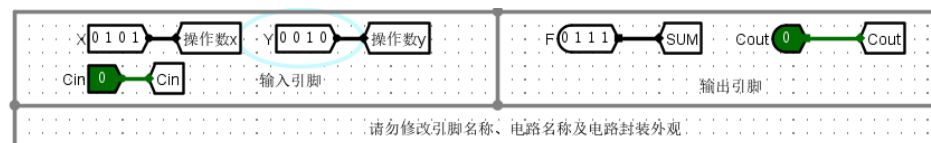
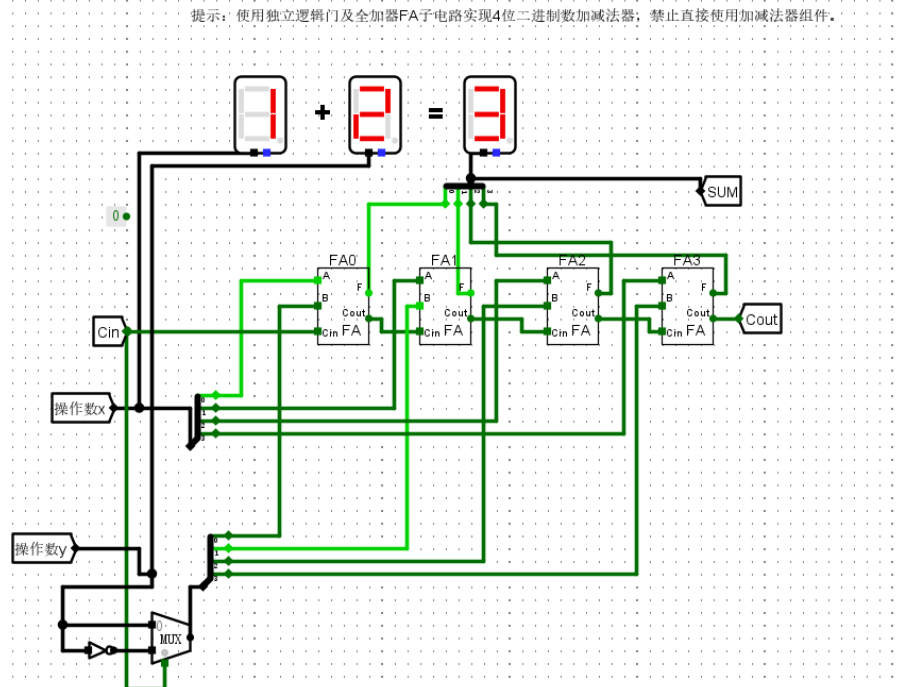
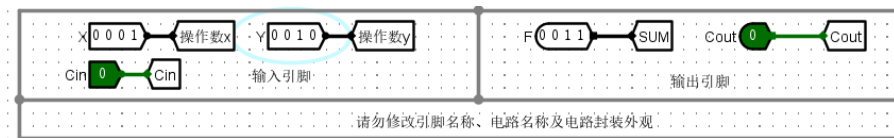
图 4.4.38 四位串行进位全加器

[https://blog.csdn.net/qz\\_45275571](https://blog.csdn.net/qz_45275571)

电路图：

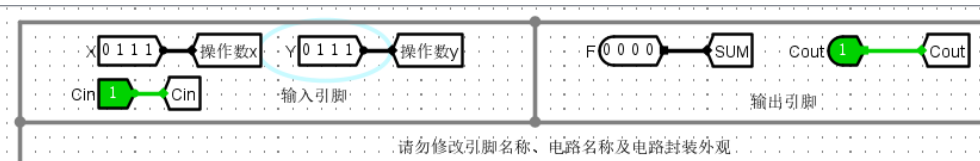
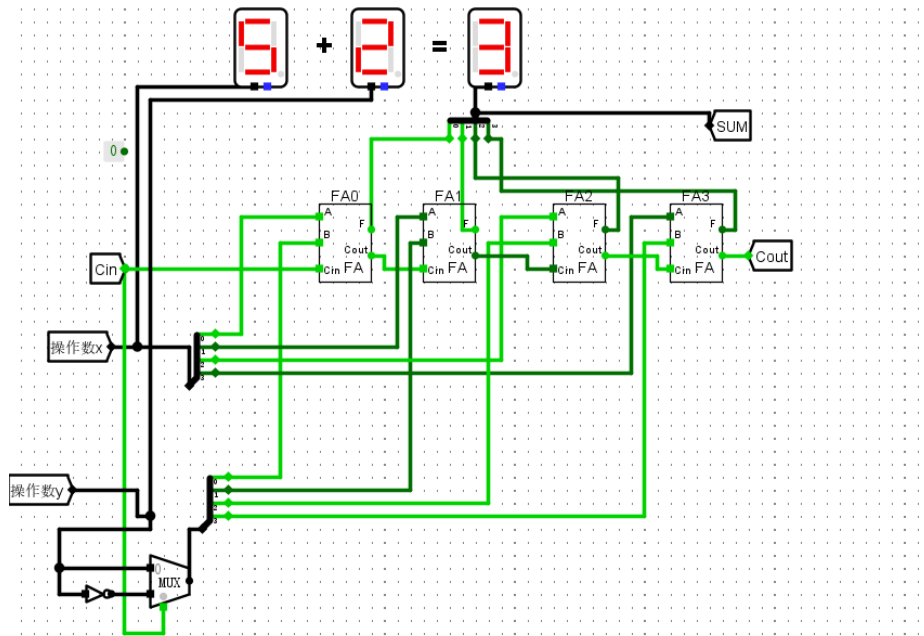


仿真测试图:

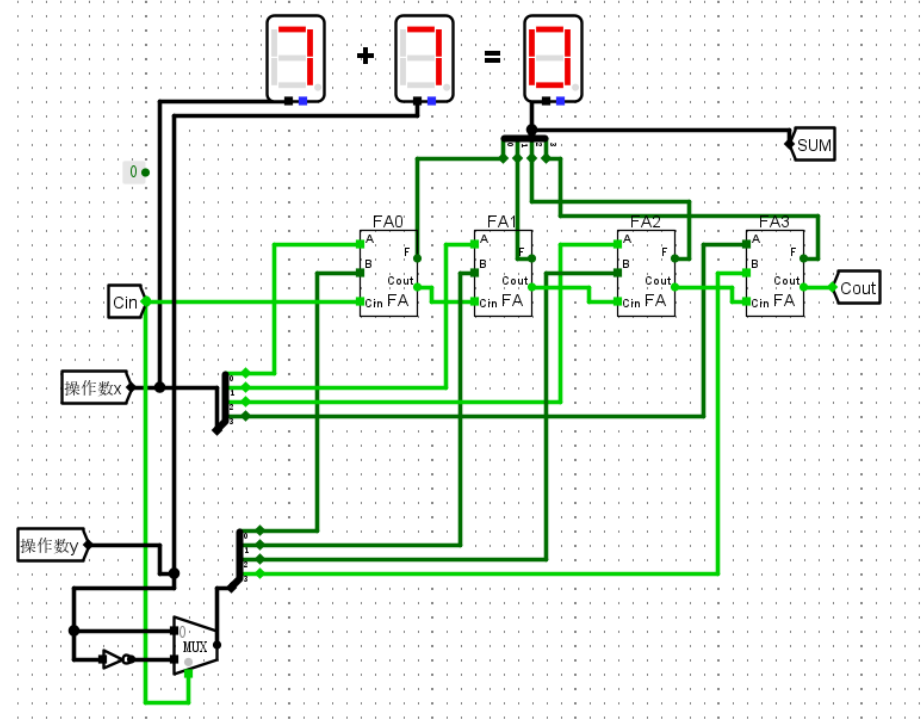




提示：使用独立逻辑门及全加器FA子电路实现4位二进制数加减法器；禁止直接使用加减法器组件。



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。



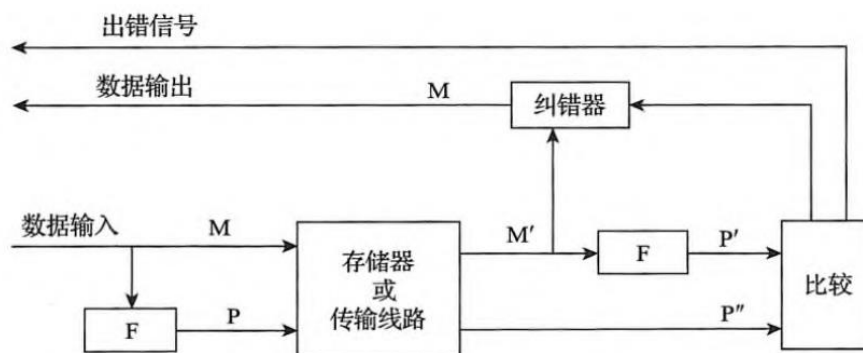
真值表：

| A | B | C <sub>in</sub> | F | C <sub>out</sub> |
|---|---|-----------------|---|------------------|
| 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                |

#### 4、汉明码校验电路

整体模块设计：

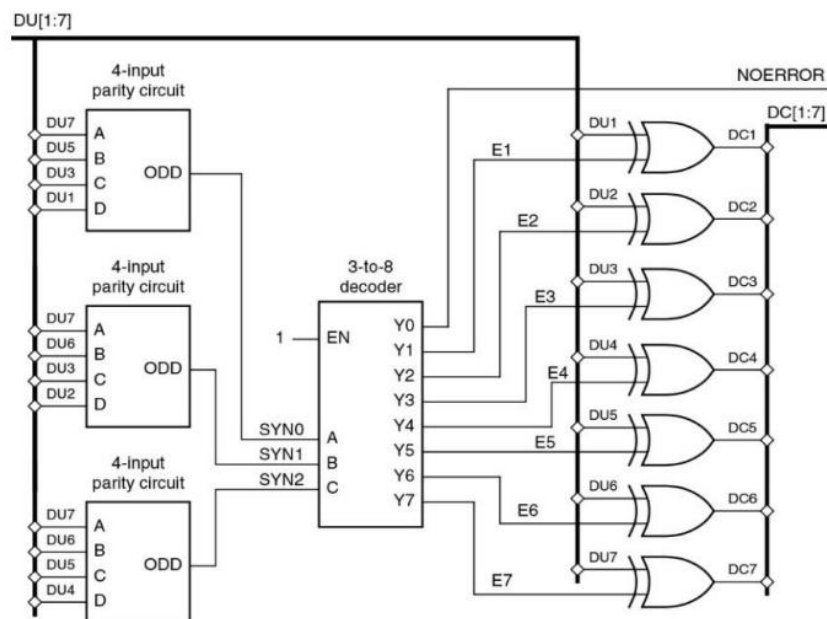
通过添加 3-8 译码器的子电路，4 位偶校验器的子电路，来控制生成 7 位的汉明码检/纠错电路。



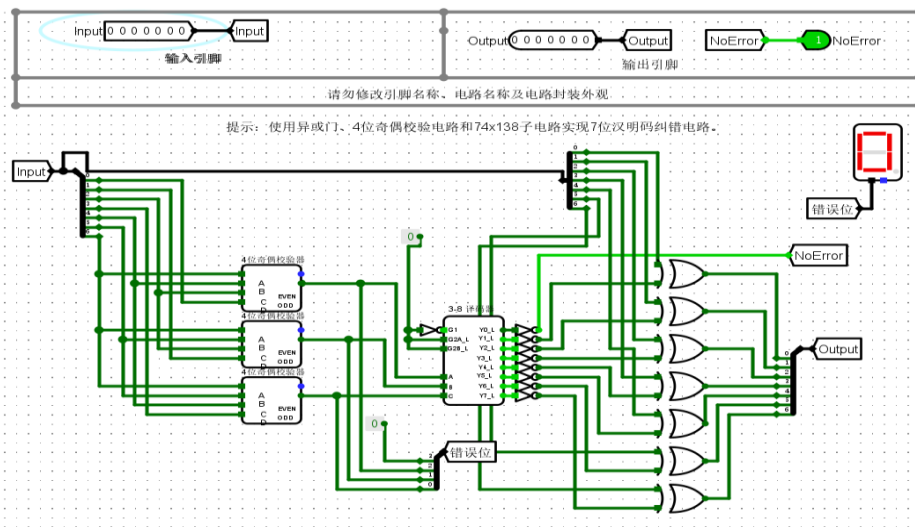
输入输出引脚：

Input 为输入，output 为输出，NoError 检验是否存在错误。

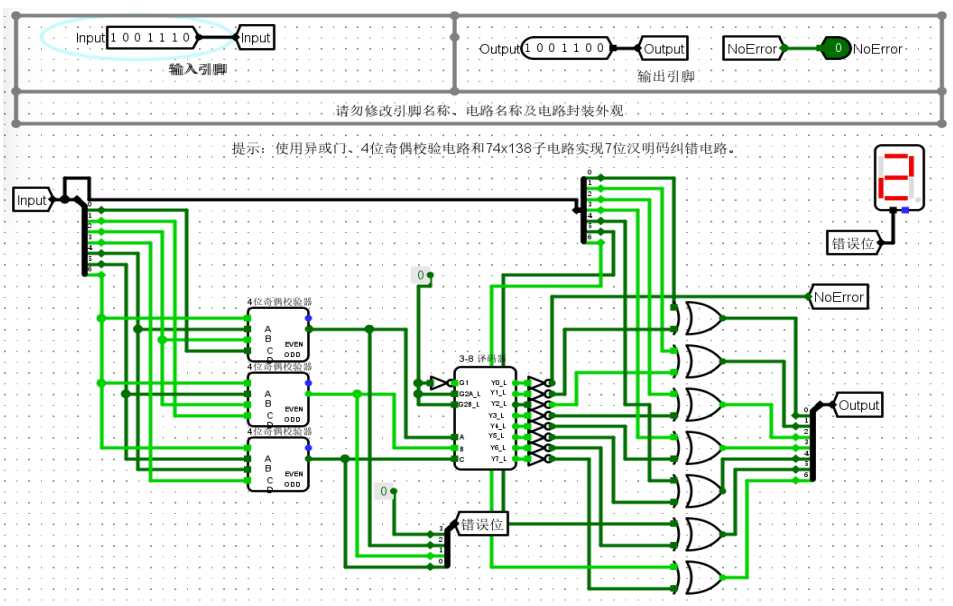
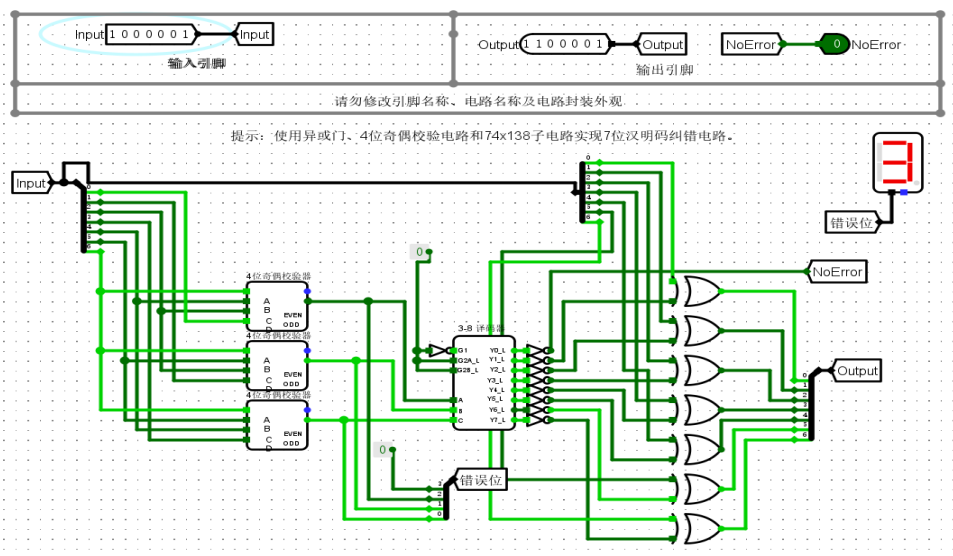
原理图：



电路图：



仿真测试图：



真值表：

| Input | NoError | Output | Input | NoError | Output | 39 | 0 | 19 | 39 | 0 | 19 |
|-------|---------|--------|-------|---------|--------|----|---|----|----|---|----|
| 00    | 1       | 00     | 00    | 1       | 00     | 3a | 0 | 2a | 3a | 0 | 2a |
| 01    | 0       | 00     | 01    | 0       | 00     | 3b | 0 | 33 | 3b | 0 | 33 |
| 02    | 0       | 00     | 02    | 0       | 00     | 3c | 0 | 34 | 3c | 0 | 34 |
| 03    | 0       | 07     | 03    | 0       | 07     | 3d | 0 | 2d | 3d | 0 | 2d |
| 04    | 0       | 00     | 04    | 0       | 00     | 3e | 0 | 1e | 3e | 0 | 1e |
| 05    | 0       | 07     | 05    | 0       | 07     | 3f | 0 | 7f | 3f | 0 | 7f |
| 06    | 0       | 07     | 06    | 0       | 07     | 40 | 0 | 00 | 40 | 0 | 00 |
| 07    | 1       | 07     | 07    | 1       | 07     | 41 | 0 | 61 | 41 | 0 | 61 |
| 08    | 0       | 00     | 08    | 0       | 00     | 42 | 0 | 52 | 42 | 0 | 52 |
| 09    | 0       | 19     | 09    | 0       | 19     | 43 | 0 | 4b | 43 | 0 | 4b |
| 0a    | 0       | 2a     | 0a    | 0       | 2a     | 44 | 0 | 4c | 44 | 0 | 4c |
| 0b    | 0       | 4b     | 0b    | 0       | 4b     | 45 | 0 | 55 | 45 | 0 | 55 |
| 0c    | 0       | 4c     | 0c    | 0       | 4c     | 46 | 0 | 66 | 46 | 0 | 66 |
| 0d    | 0       | 2d     | 0d    | 0       | 2d     | 47 | 0 | 07 | 47 | 0 | 07 |
| 0e    | 0       | 1e     | 0e    | 0       | 1e     | 48 | 0 | 4c | 48 | 0 | 4c |
| 0f    | 0       | 07     | 0f    | 0       | 07     | 49 | 0 | 4b | 49 | 0 | 4b |
| 10    | 0       | 00     | 10    | 0       | 00     | 4a | 0 | 4b | 4a | 0 | 4b |
| 11    | 0       | 19     | 11    | 0       | 19     | 4b | 1 | 4b | 4b | 1 | 4b |
| 12    | 0       | 52     | 12    | 0       | 52     | 4c | 1 | 4c | 4c | 1 | 4c |
| 13    | 0       | 33     | 13    | 0       | 33     | 4d | 0 | 4c | 4d | 0 | 4c |
| 14    | 0       | 34     | 14    | 0       | 34     | 4e | 0 | 4c | 4e | 0 | 4c |
| 15    | 0       | 55     | 15    | 0       | 55     | 4f | 0 | 4b | 4f | 0 | 4b |
| 16    | 0       | 1e     | 16    | 0       | 1e     | 50 | 0 | 52 | 50 | 0 | 52 |
| 17    | 0       | 07     | 17    | 0       | 07     | 51 | 0 | 55 | 51 | 0 | 55 |
| 18    | 0       | 19     | 18    | 0       | 19     | 52 | 1 | 52 | 52 | 1 | 52 |
| 19    | 1       | 19     | 19    | 1       | 19     | 53 | 0 | 52 | 53 | 0 | 52 |
| 1a    | 0       | 1e     | 1a    | 0       | 1e     | 54 | 0 | 55 | 54 | 0 | 55 |
| 1b    | 0       | 19     | 1b    | 0       | 19     | 55 | 1 | 55 | 55 | 1 | 55 |
| 1c    | 0       | 1e     | 1c    | 0       | 1e     | 56 | 0 | 52 | 56 | 0 | 52 |
| 1d    | 0       | 19     | 1d    | 0       | 19     | 57 | 0 | 55 | 57 | 0 | 55 |
| 1e    | 1       | 1e     | 1e    | 1       | 1e     | 58 | 0 | 78 | 58 | 0 | 78 |
| 1f    | 0       | 1e     | 1f    | 0       | 1e     | 59 | 0 | 19 | 59 | 0 | 19 |
| 20    | 0       | 00     | 20    | 0       | 00     | 5a | 0 | 52 | 5a | 0 | 52 |
| 21    | 0       | 61     | 21    | 0       | 61     | 5b | 0 | 4b | 5b | 0 | 4b |
| 22    | 0       | 2a     | 22    | 0       | 2a     | 5c | 0 | 4c | 5c | 0 | 4c |
| 23    | 0       | 33     | 23    | 0       | 33     | 5d | 0 | 55 | 5d | 0 | 55 |
| 24    | 0       | 34     | 24    | 0       | 34     | 5e | 0 | 1e | 5e | 0 | 1e |
| 25    | 0       | 2d     | 25    | 0       | 2d     | 5f | 0 | 7f | 5f | 0 | 7f |
| 26    | 0       | 66     | 26    | 0       | 66     | 60 | 0 | 61 | 60 | 0 | 61 |
| 27    | 0       | 07     | 27    | 0       | 07     | 61 | 1 | 61 | 61 | 1 | 61 |
| 28    | 0       | 2a     | 28    | 0       | 2a     | 62 | 0 | 66 | 62 | 0 | 66 |
| 29    | 0       | 2d     | 29    | 0       | 2d     | 63 | 0 | 61 | 63 | 0 | 61 |
| 2a    | 1       | 2a     | 2a    | 1       | 2a     | 64 | 0 | 66 | 64 | 0 | 66 |
| 2b    | 0       | 2a     | 2b    | 0       | 2a     | 65 | 0 | 61 | 65 | 0 | 61 |
| 2c    | 0       | 2d     | 2c    | 0       | 2d     | 66 | 1 | 66 | 66 | 1 | 66 |
| 2d    | 1       | 2d     | 2d    | 1       | 2d     | 67 | 0 | 66 | 67 | 0 | 66 |
| 2e    | 0       | 2a     | 2e    | 0       | 2a     | 68 | 0 | 78 | 68 | 0 | 78 |
| 2f    | 0       | 2d     | 2f    | 0       | 2d     | 69 | 0 | 61 | 69 | 0 | 61 |
| 30    | 0       | 34     | 30    | 0       | 34     | 6a | 0 | 2a | 6a | 0 | 2a |
| 31    | 0       | 33     | 31    | 0       | 33     | 6b | 0 | 4b | 6b | 0 | 4b |
| 32    | 0       | 33     | 32    | 0       | 33     | 6c | 0 | 4c | 6c | 0 | 4c |
| 33    | 1       | 33     | 33    | 1       | 33     | 6d | 0 | 2d | 6d | 0 | 2d |
| 34    | 1       | 34     | 34    | 1       | 34     | 6e | 0 | 66 | 6e | 0 | 66 |
| 35    | 0       | 34     | 35    | 0       | 34     | 6f | 0 | 7f | 6f | 0 | 7f |
| 36    | 0       | 34     | 36    | 0       | 34     | 70 | 0 | 78 | 70 | 0 | 78 |
| 37    | 0       | 33     | 37    | 0       | 33     | 71 | 0 | 61 | 71 | 0 | 61 |
| 38    | 0       | 78     | 38    | 0       | 78     | 72 | 0 | 52 | 72 | 0 | 52 |
|       |         |        |       |         |        | 73 | 0 | 33 | 73 | 0 | 33 |
|       |         |        |       |         |        | 74 | 0 | 34 | 74 | 0 | 34 |
|       |         |        |       |         |        | 75 | 0 | 55 | 75 | 0 | 55 |
|       |         |        |       |         |        | 76 | 0 | 66 | 76 | 0 | 66 |
|       |         |        |       |         |        | 77 | 0 | 7f | 77 | 0 | 7f |
|       |         |        |       |         |        | 78 | 1 | 78 | 78 | 1 | 78 |
|       |         |        |       |         |        | 79 | 0 | 78 | 79 | 0 | 78 |
|       |         |        |       |         |        | 7a | 0 | 78 | 7a | 0 | 78 |
|       |         |        |       |         |        | 7b | 0 | 7f | 7b | 0 | 7f |
|       |         |        |       |         |        | 7c | 0 | 78 | 7c | 0 | 78 |
|       |         |        |       |         |        | 7d | 0 | 7f | 7d | 0 | 7f |
|       |         |        |       |         |        | 7e | 0 | 7f | 7e | 0 | 7f |
|       |         |        |       |         |        | 7f | 1 | 7f | 7f | 1 | 7f |

## 5、桶形移位器

整体模块设计：

使用 3 个 8 位的四路选择器，实现移动类型给定（包含算数/逻辑/循环移位），移动位数确定的移位功能。

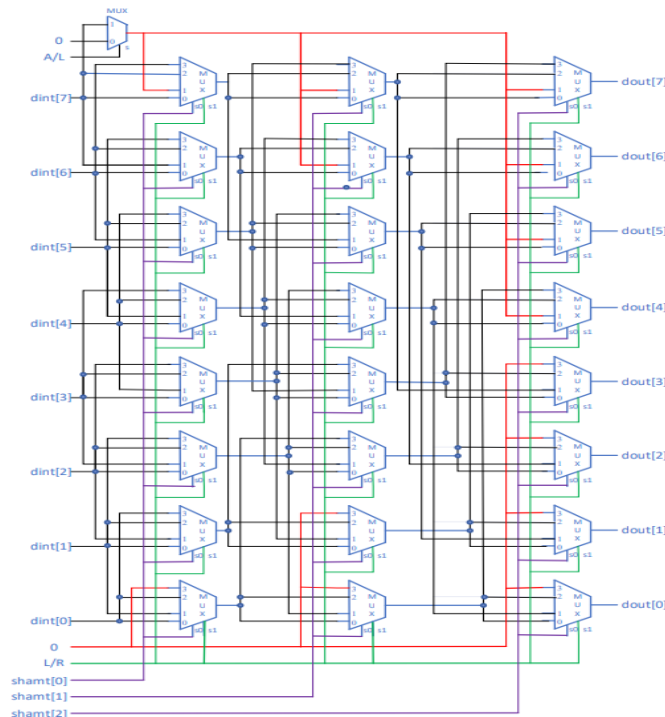
输入输出引脚功能：

Shamt 表示移位位数，选择段 L/R 表示左移和右移，选择段 A/L 为移位类型选择，din 为输入数据端，dout 为输出数据端。

原理图：

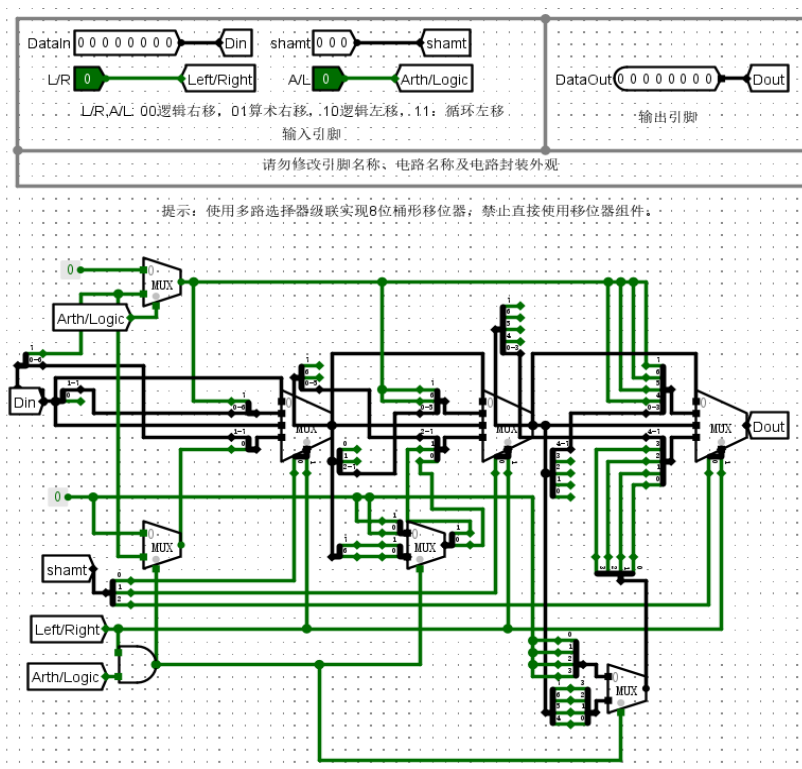
第一级利用 shamt[0]来控制是否需要移动一位，第二级在第一级的移动结果上用 shamt[1]

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。



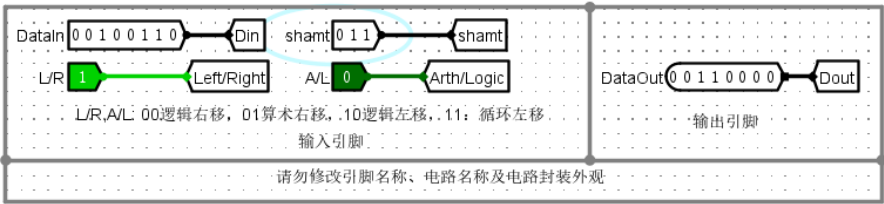
(此为不含循环移位的原理图)

电路图：

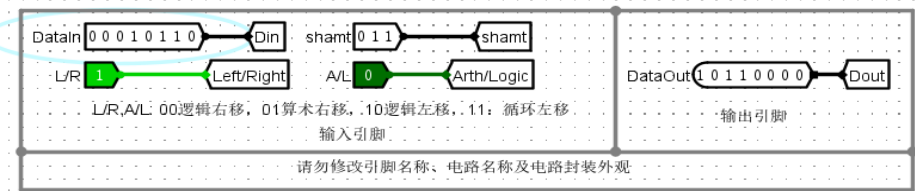
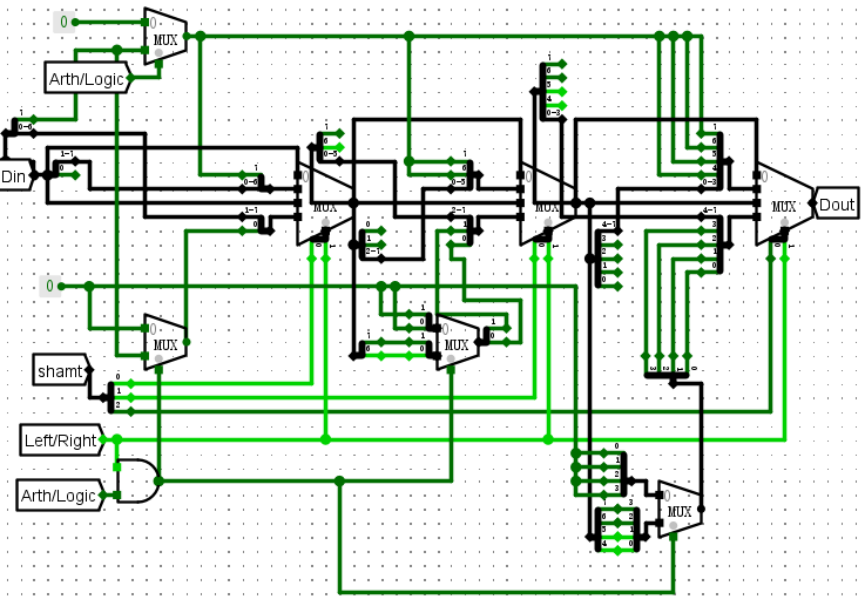


提示：使用多路选择器级联实现8位桶形移位器，禁止直接使用移位器组件。

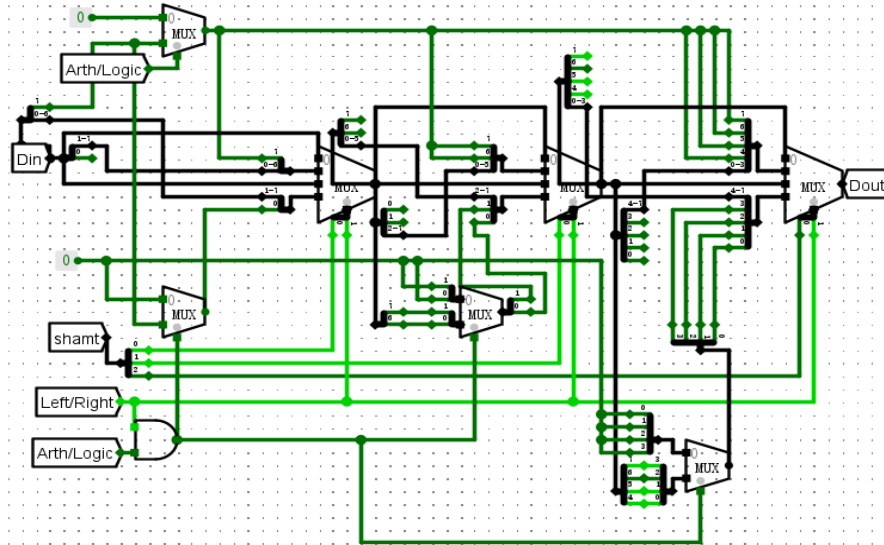
仿真测试图:



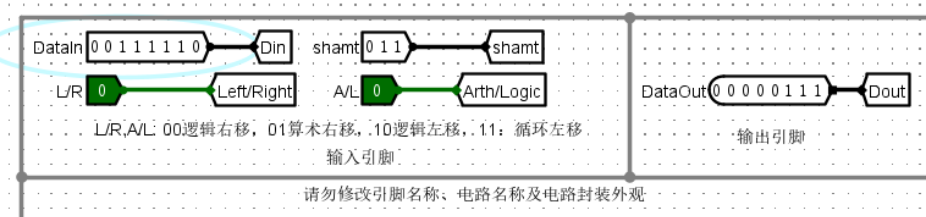
提示：使用多路选择器级联实现8位桶形移位器；禁止直接使用移位器组件。



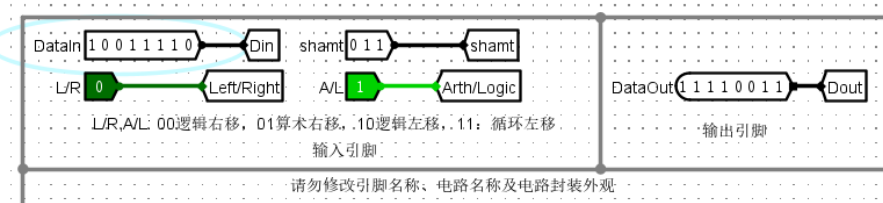
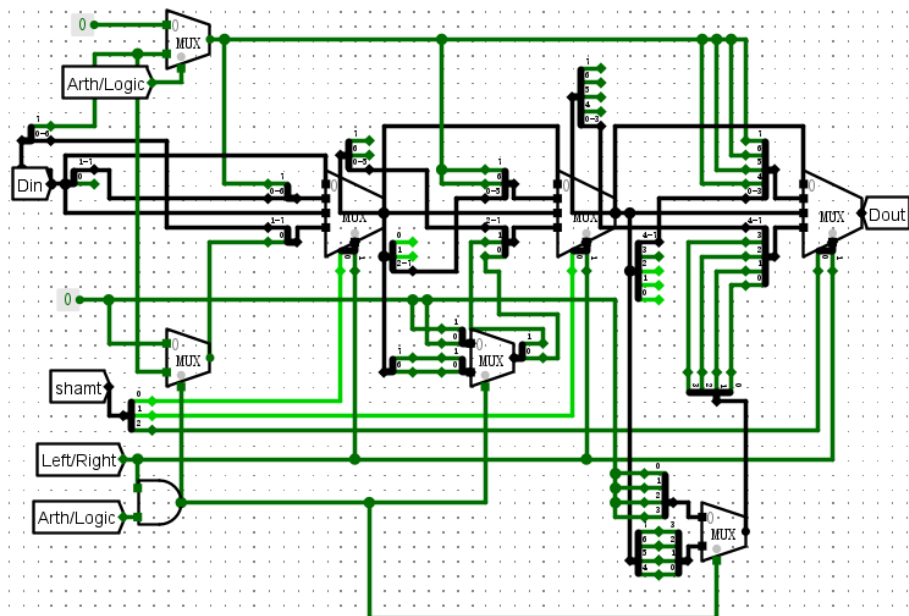
提示：使用多路选择器级联实现8位桶形移位器；禁止直接使用移位器组件。



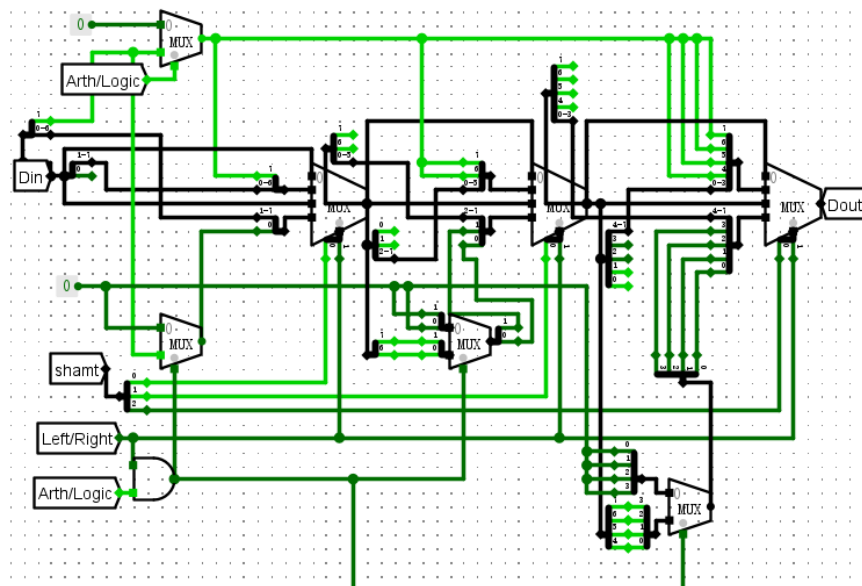


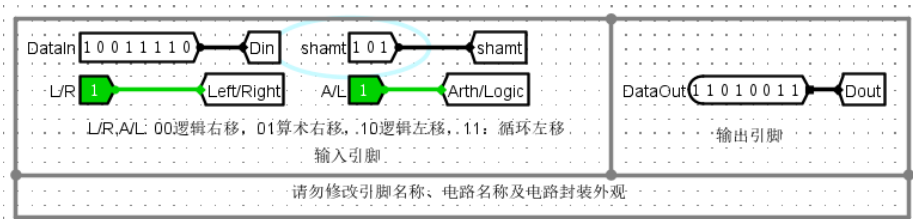


提示：使用多路选择器级联实现8位桶形移位器；禁止直接使用移位器组件。

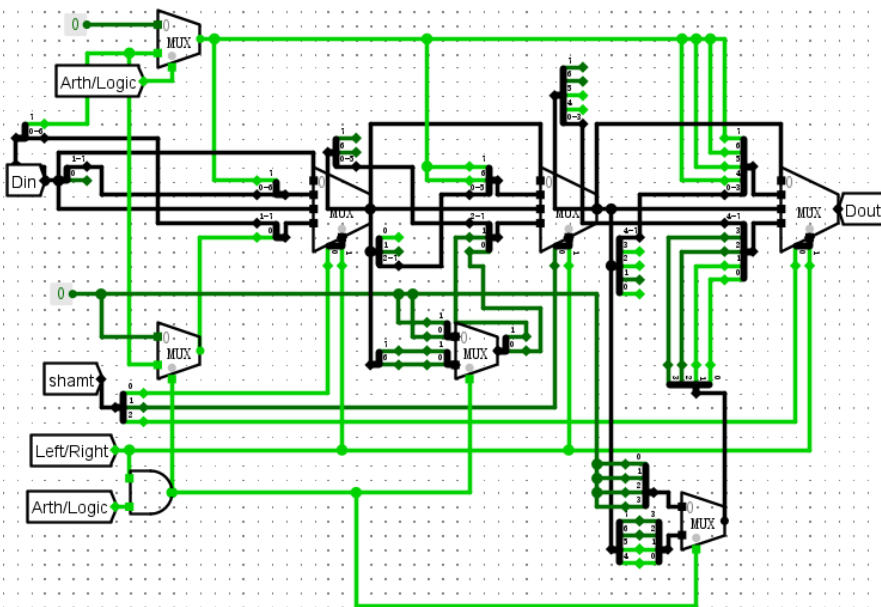


提示：使用多路选择器级联实现8位桶形移位器；禁止直接使用移位器组件。





提示：使用多路选择器级联实现8位桶形移位器；禁止直接使用移位器组件。



真值表：

| Din | shamt | LR | AL | Dout | Din | shamt | LR | AL | Dout |
|-----|-------|----|----|------|-----|-------|----|----|------|
| b7  | 0     | 0  | 0  | b7   | b7  | 0     | 0  | 0  | b7   |
| b7  | 1     | 0  | 0  | 5b   | b7  | 1     | 0  | 0  | 5b   |
| b7  | 2     | 0  | 0  | 2d   | b7  | 2     | 0  | 0  | 2d   |
| b7  | 3     | 0  | 0  | 16   | b7  | 3     | 0  | 0  | 16   |
| b7  | 4     | 0  | 0  | 0b   | b7  | 4     | 0  | 0  | 0b   |
| b7  | 5     | 0  | 0  | 05   | b7  | 5     | 0  | 0  | 05   |
| b7  | 6     | 0  | 0  | 02   | b7  | 6     | 0  | 0  | 02   |
| b7  | 7     | 0  | 0  | 01   | b7  | 7     | 0  | 0  | 01   |
| b7  | 0     | 1  | 0  | b7   | b7  | 0     | 1  | 0  | b7   |
| b7  | 1     | 1  | 0  | 6e   | b7  | 1     | 1  | 0  | 6e   |
| b7  | 2     | 1  | 0  | dc   | b7  | 2     | 1  | 0  | dc   |
| b7  | 3     | 1  | 0  | b8   | b7  | 3     | 1  | 0  | b8   |
| b7  | 4     | 1  | 0  | 70   | b7  | 4     | 1  | 0  | 70   |
| b7  | 5     | 1  | 0  | e0   | b7  | 5     | 1  | 0  | e0   |
| b7  | 6     | 1  | 0  | c0   | b7  | 6     | 1  | 0  | c0   |
| b7  | 7     | 1  | 0  | 80   | b7  | 7     | 1  | 0  | 80   |
| b7  | 0     | 0  | 1  | b7   | b7  | 0     | 0  | 1  | b7   |
| b7  | 1     | 0  | 1  | db   | b7  | 1     | 0  | 1  | db   |
| b7  | 2     | 0  | 1  | ed   | b7  | 2     | 0  | 1  | ed   |
| b7  | 3     | 0  | 1  | f6   | b7  | 3     | 0  | 1  | f6   |
| b7  | 4     | 0  | 1  | fb   | b7  | 4     | 0  | 1  | fb   |
| b7  | 5     | 0  | 1  | fd   | b7  | 5     | 0  | 1  | fd   |
| b7  | 6     | 0  | 1  | fe   | b7  | 6     | 0  | 1  | fe   |
| b7  | 7     | 0  | 1  | ff   | b7  | 7     | 0  | 1  | ff   |
| b7  | 0     | 1  | 1  | b7   | b7  | 0     | 1  | 1  | b7   |
| b7  | 1     | 1  | 1  | 6f   | b7  | 1     | 1  | 1  | 6f   |
| b7  | 2     | 1  | 1  | de   | b7  | 2     | 1  | 1  | de   |
| b7  | 3     | 1  | 1  | bd   | b7  | 3     | 1  | 1  | bd   |
| b7  | 4     | 1  | 1  | 7b   | b7  | 4     | 1  | 1  | 7b   |
| b7  | 5     | 1  | 1  | f6   | b7  | 5     | 1  | 1  | f6   |
| b7  | 6     | 1  | 1  | ed   | b7  | 6     | 1  | 1  | ed   |
| b7  | 7     | 1  | 1  | db   | b7  | 7     | 1  | 1  | db   |

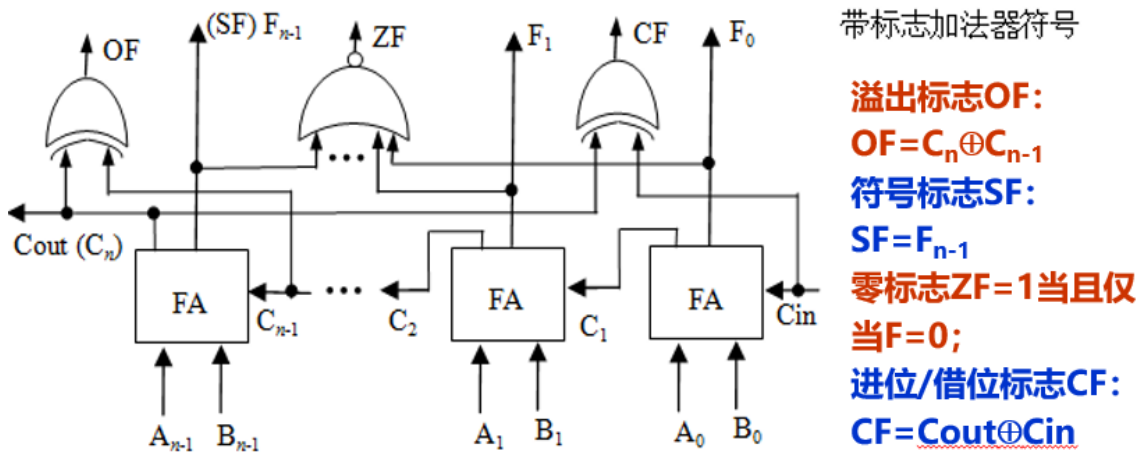
#### 四、思考题

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。

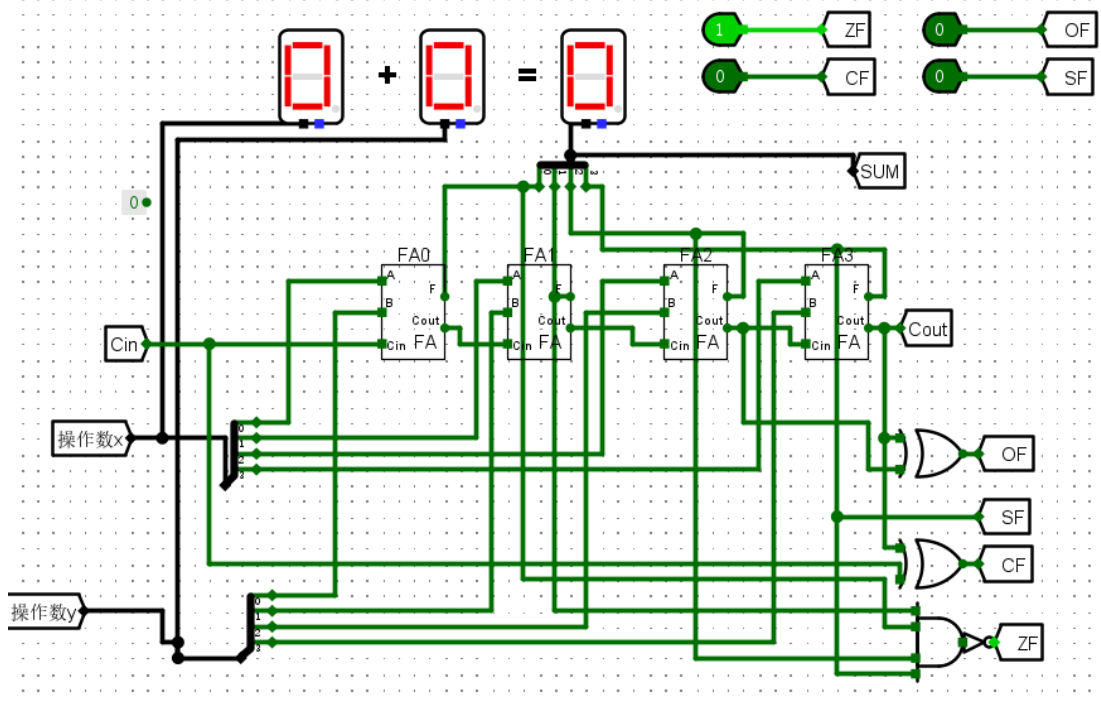
整体模块设计：

在全加器的基础上，通过增添异或门和与门实现标志位的生成。

原理图：



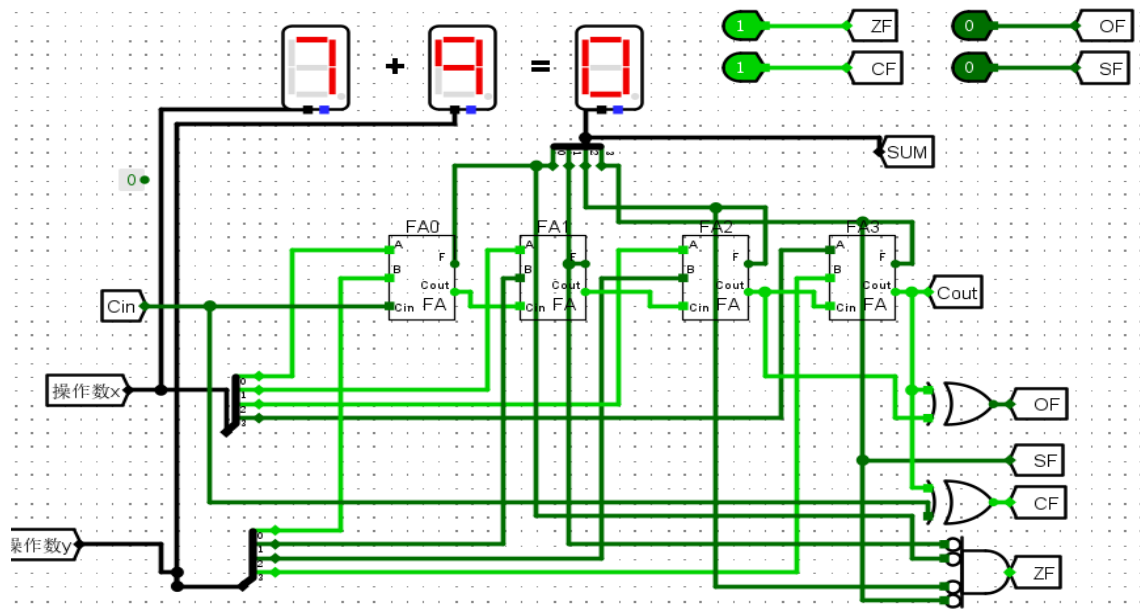
电路图：



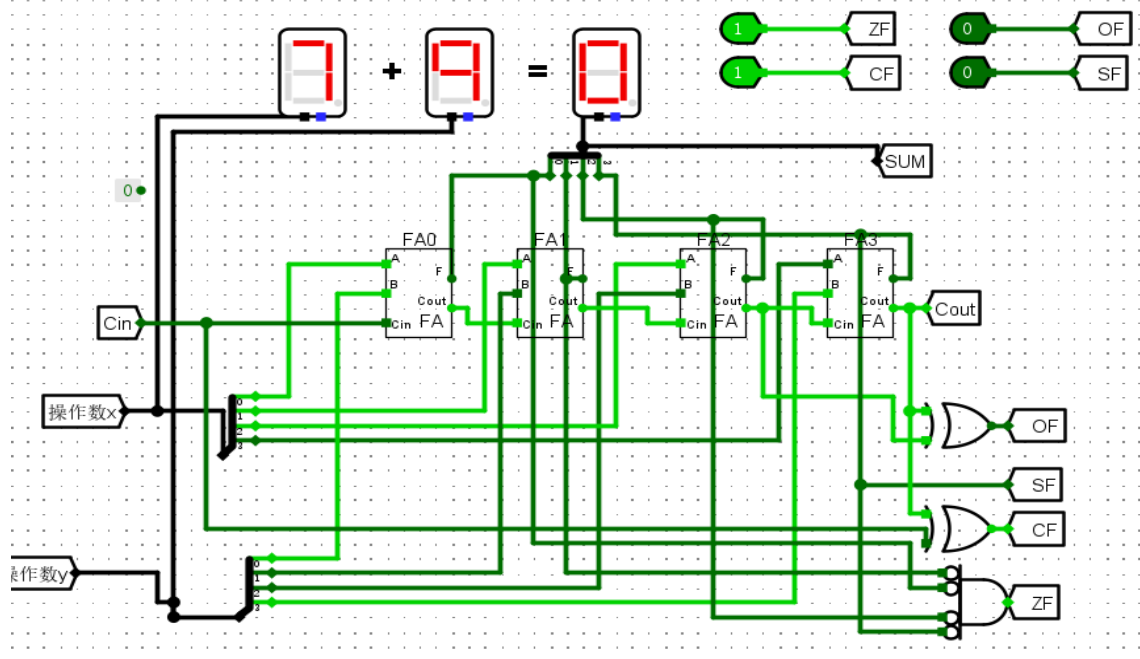
仿真测试图：

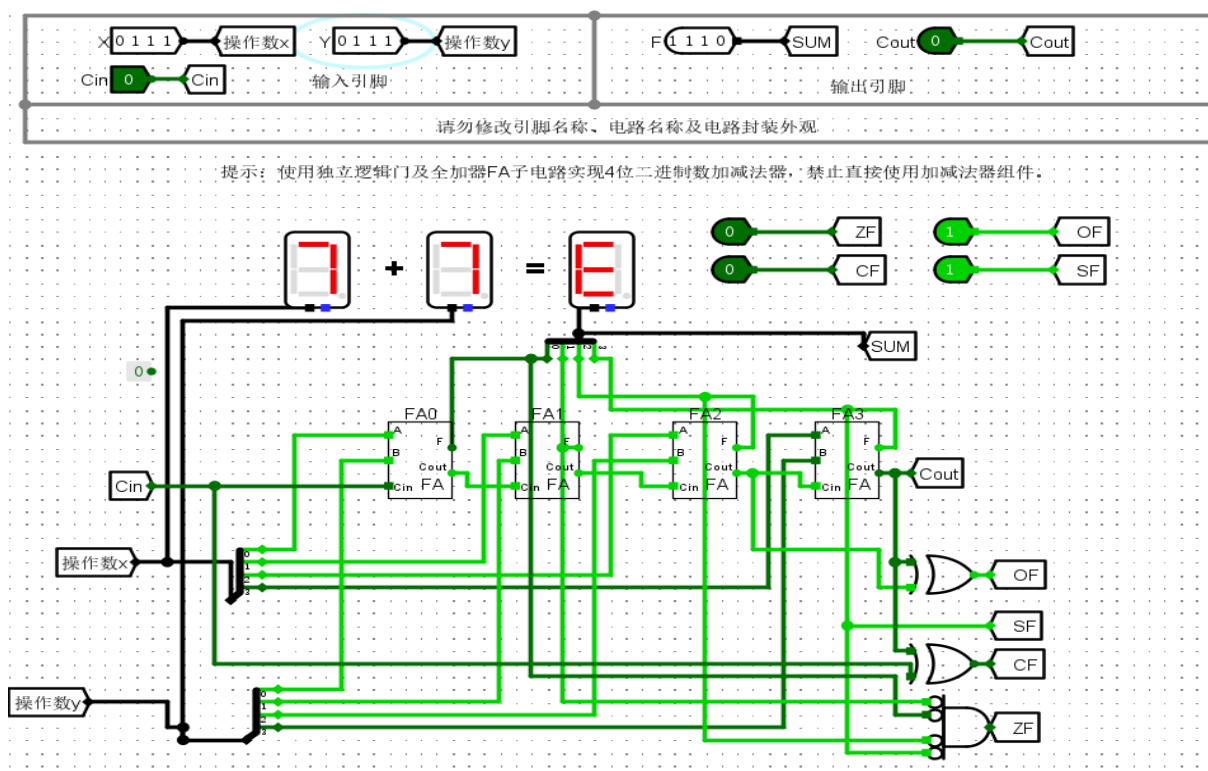


提示：使用独立逻辑门及全加器FA子电路实现4位二进制数加减法器，禁止直接使用加减法器组件。



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。





## 2、通过减法器执行比较指令

执行比较指令，也即是判定  $a$  与  $b$  的大小，此时可通过  $a-b$  来判定。

A:  $a, b$  均为无符号数。

若  $a < b$ : 那么结果不会溢出,  $OF=0$ ; 需要进行借位减法,  $CF=1$ ;  $ZF=0$ ;

若  $a > b$ : 那么结果仍然不会溢出,  $OF=0$ ; 不需要借位减法,  $CF=0$ ;  $ZF=0$ ;

若  $a = b$ :  $ZF=1$ ;

B:  $a, b$  均为带符号数。

若  $a > b$ : 符号仍然为正,  $SF=0$ ; 且没有发生溢出,  $OF=0$ ;  $ZF=0$ ;

若  $a < b$ : 符号为负,  $SF=1$ ; 仍然不可能发生溢出,  $OF=0$ ;  $ZF=0$ ;

若  $a = b$ :  $ZF=0$ ;

综上:

无符号数比较:

两有符号数相减判断结果正负:

$SF=OF$  时大于,  $SF \neq OF$  时小于;  
即  $SF \oplus OF=0$  时大于,  $SF \oplus OF=1$  时小于。

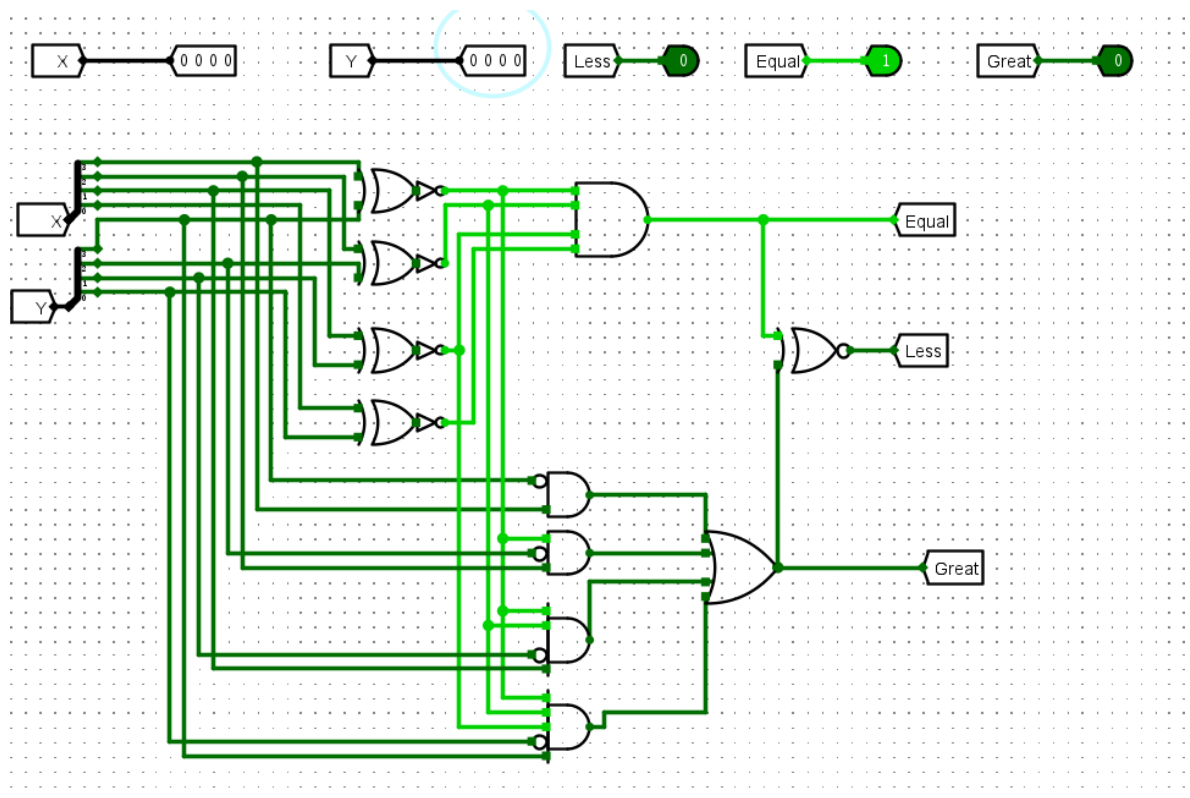
做减法以比较大小, 规则:  
Unsigned:  $CF=0$  时, 大于

3、使用逻辑门电路实现 4 位无符号二进制数比较器, 并扩展到带符号数的比较, 输出相等、大于 和 小于三个结果。

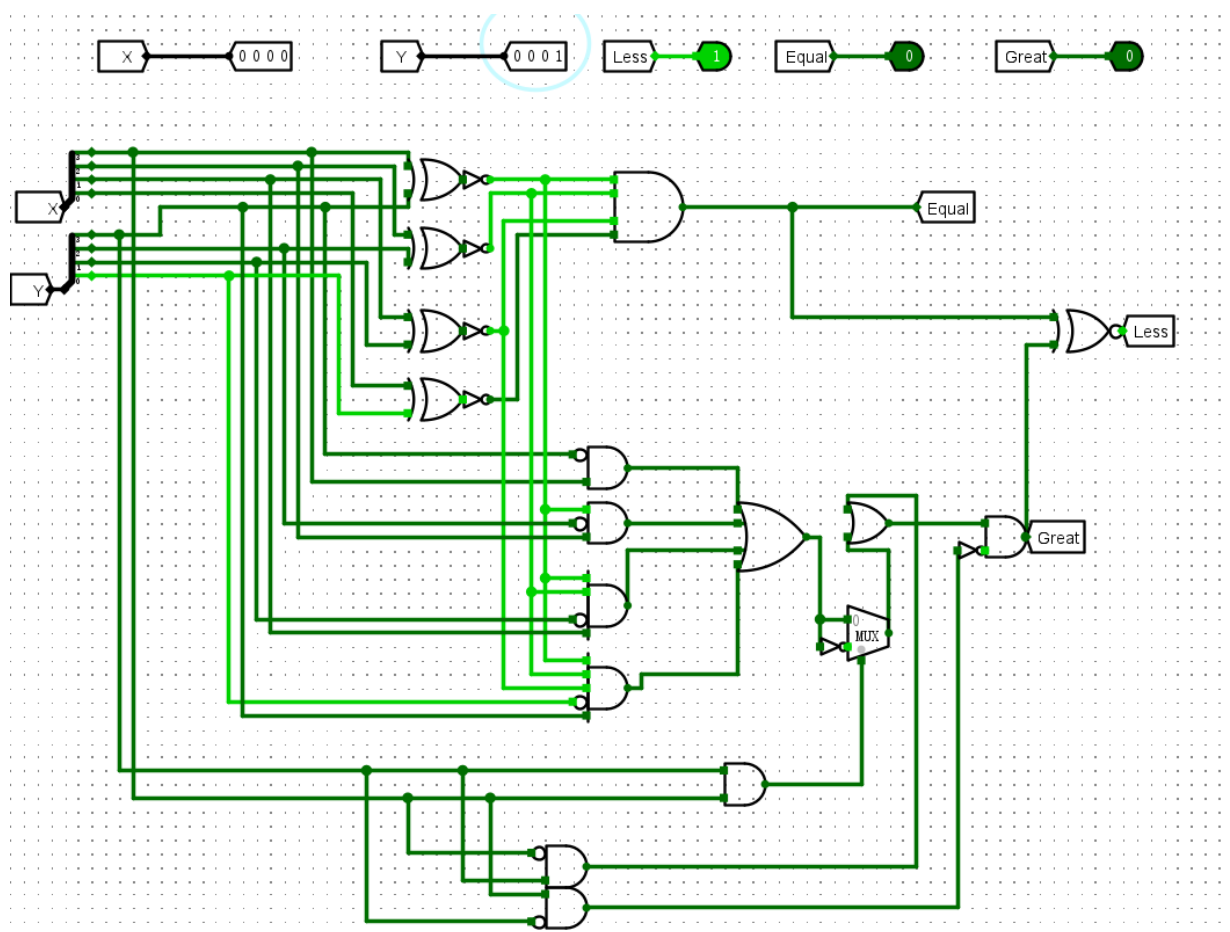
整体模块设计:

对无符号数, 高位优先, 从最高位到最低位依次比较; 对带符号数, 符号位优先比较, 在符号位相等的情况下, 依次比较。

原理图与电路图：(无符号)



(有符号):



#### 4、使用 8 位桶形移位器扩展到 32 位桶形移位器。

整体模块设计：

消防 8 位桶形移位器，将选择题增添到五个，代表移 1 位，2 位，4 位，8 位，16 位，32 位。Shamt 数组增添至五位。其余原理保持一致。

输入输出引脚：

与 8 位桶形移位器保持一致，移除了循环左移这一功能。

电路图：

