

实验 1: 基本逻辑部件设计

一、实验目的

1. 掌握 Logisim 软件的使用方法。
2. 掌握门电路的实现方法。
3. 掌握数字系统设计的基本设计方法。
4. 掌握子电路的设计和应用。
5. 熟练运用分线器、隧道、探针等 Logisim 组件的。

二、实验内容

1、利用基本逻辑门设计一个 3 输入多数表决器。

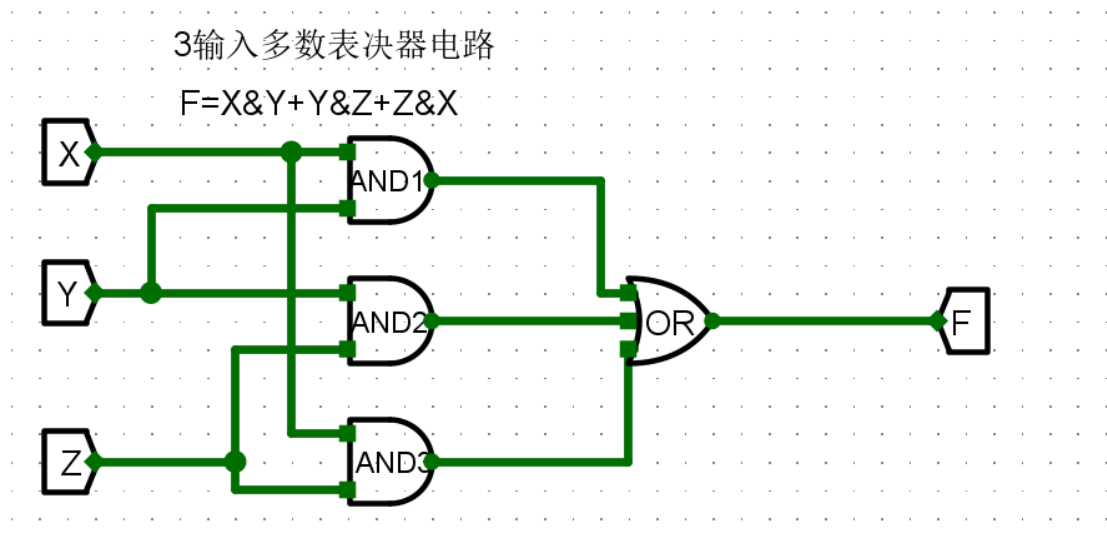
整体模块设计:

使用三个与门, 一个或门实现三输入的多数表决器。

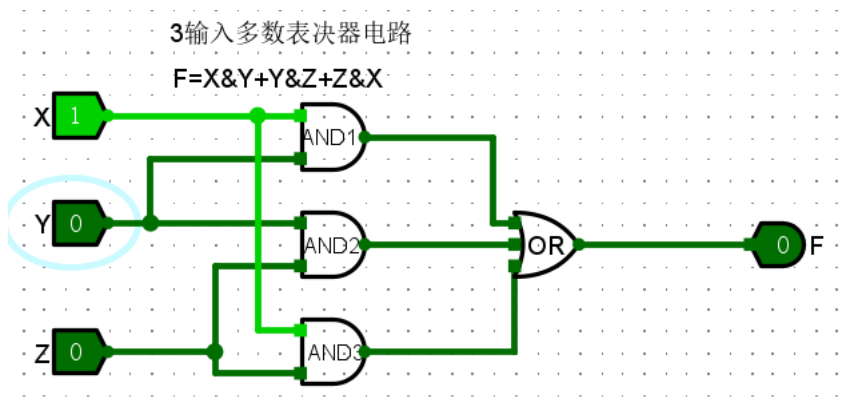
输入输出引脚作用:

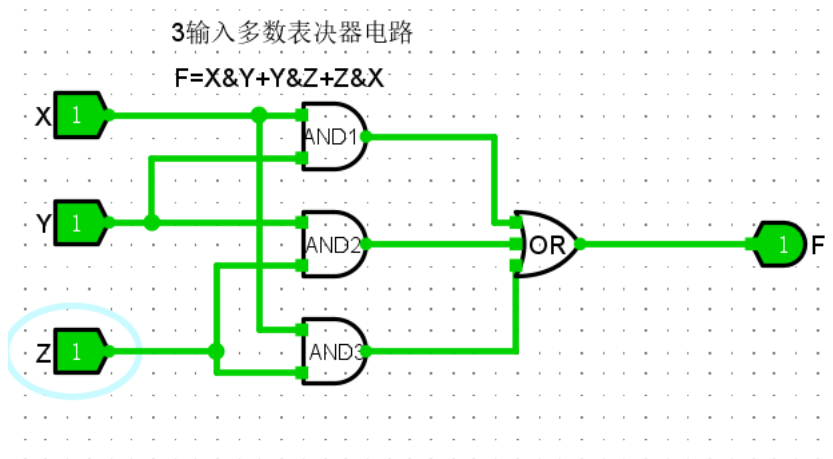
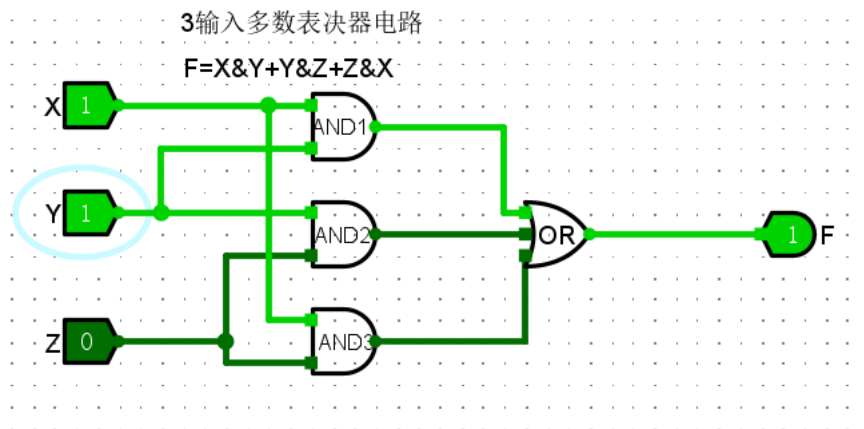
X,Y,Z 为输入端, F 为输出端

原理图与电路图: (此实验较为简单, 二者为一)



仿真测试图:





真值表：

X	Y	Z	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

2、利用 CMOS 晶体管构建两输入或门，并验证其功能。

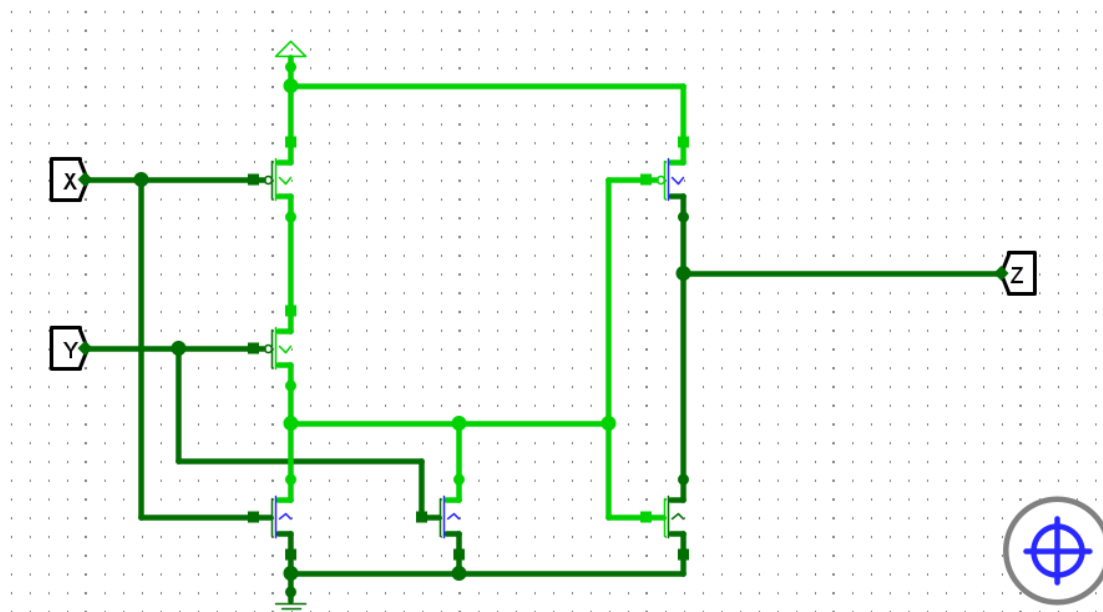
整体模块设计：

运用 3 对 CMOS 晶体管，1 个电源，1 个地线构建 2 输入的或门。

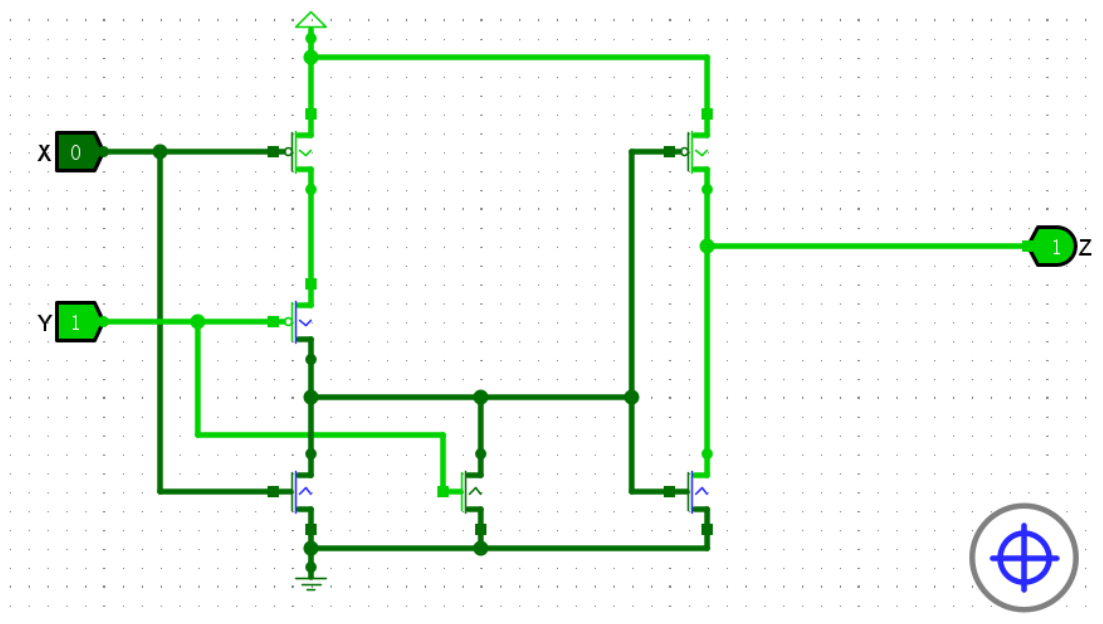
输入输出引脚作用：

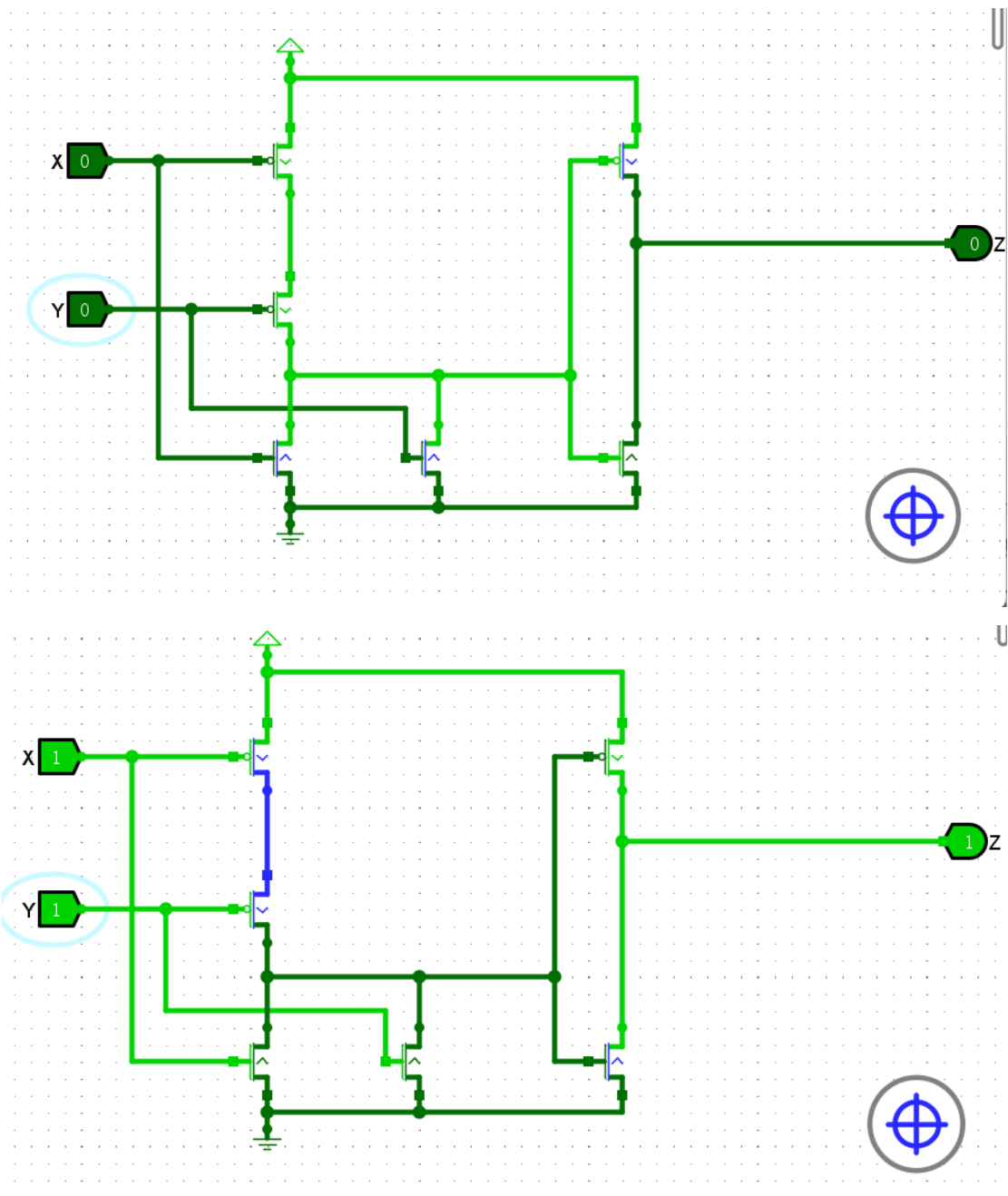
X, Y 为输入端，Z 为输出端

原理图和电路图：（实验较为简单，二者合一）



仿真测试图：





真值表：

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

3、利用基本逻辑门和 CMOS 晶体管实现多路选择器，并进行冒险检测。

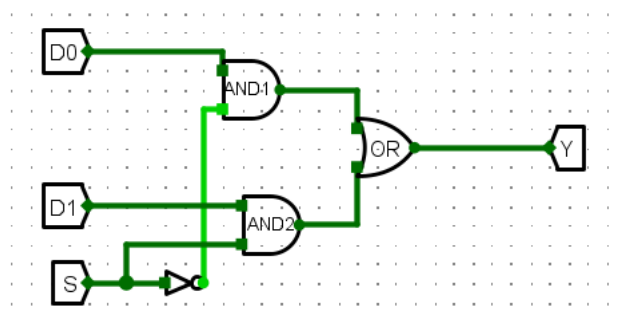
整体模块设计：

运用 2 个与门，一个或门，3 个输入引脚，1 个输出引脚实现两级与-或逻辑电路。

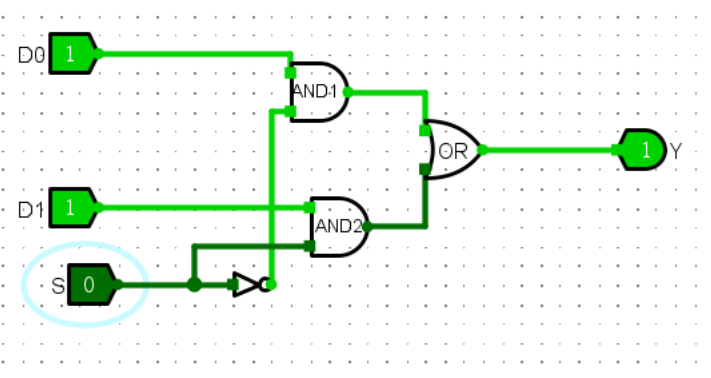
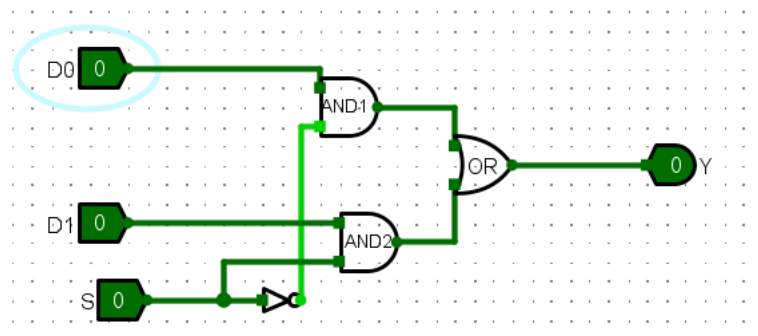
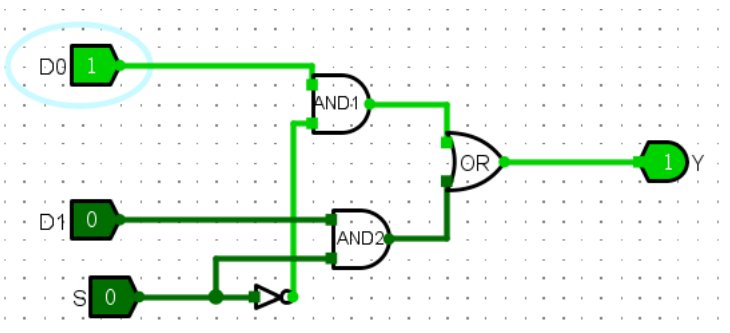
输入输出引脚作用：

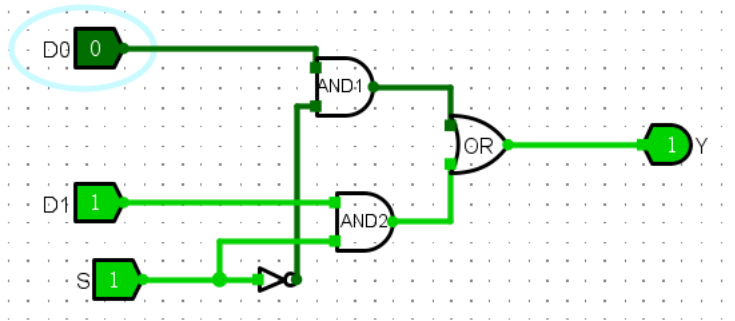
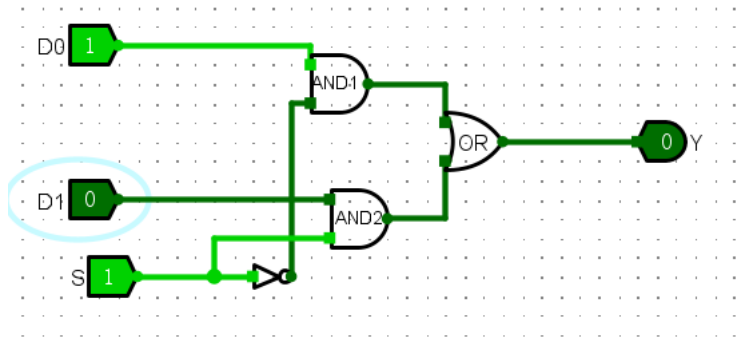
D0, D1 为输入端, S 为控制端, Y 为输出端

原理图和电路图：(实验简单，二者合一)



仿真测试图：

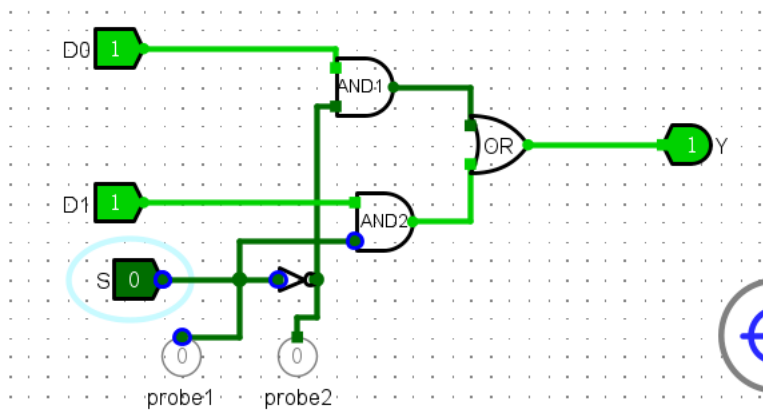


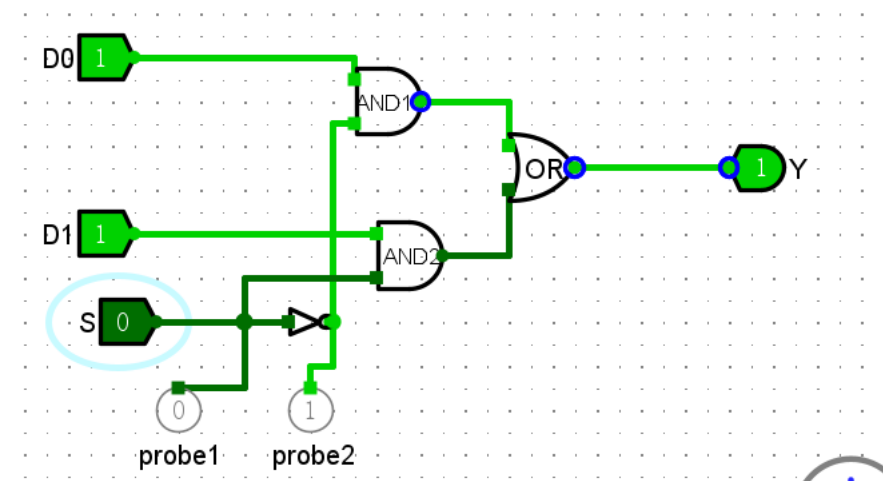
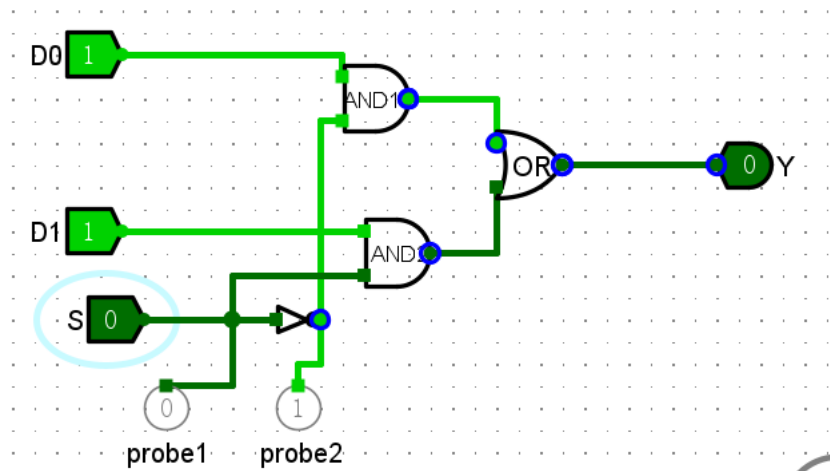
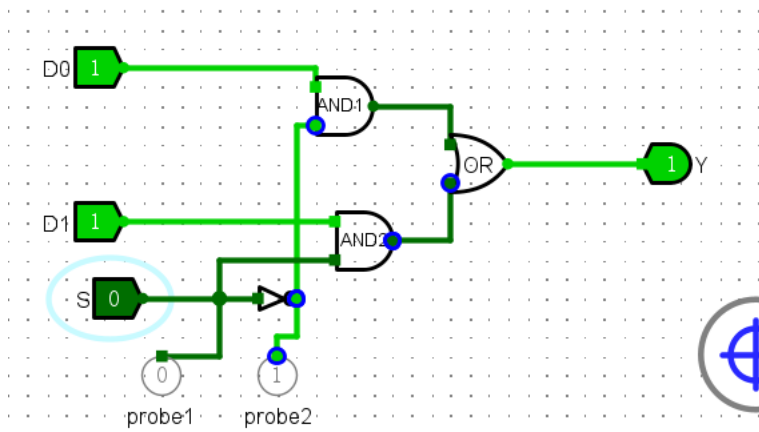


真值表:

D0	D1	S	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

冒险检测图:





4、利用传输门实现 2 路选择器

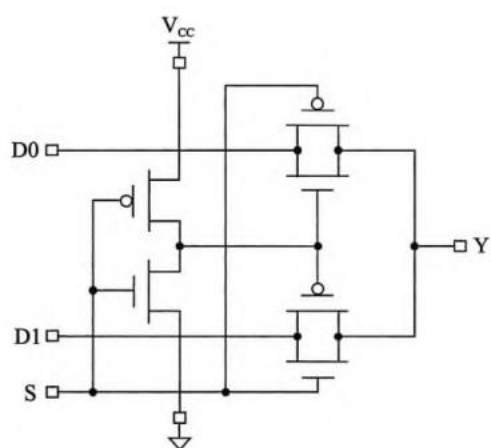
基本模块设计：

利用 1 对 CMOS 晶体管，2 个传输门，2 个输入引脚，1 个输出引脚，1 个电源，1 个地线来实现 2 路选择器。

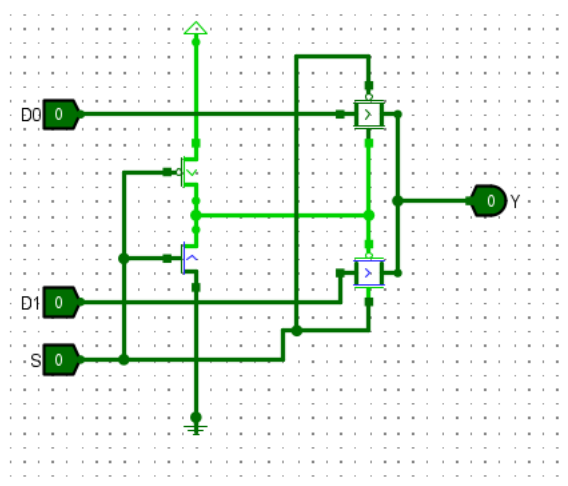
输入输出引脚作用：

D0, D1 为输入端，S 为控制端，Y 为输出端。

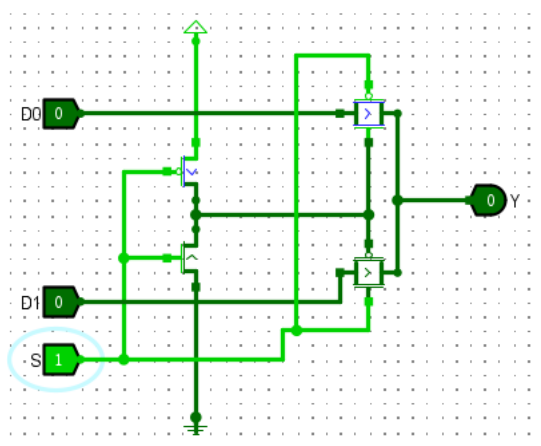
原理图：

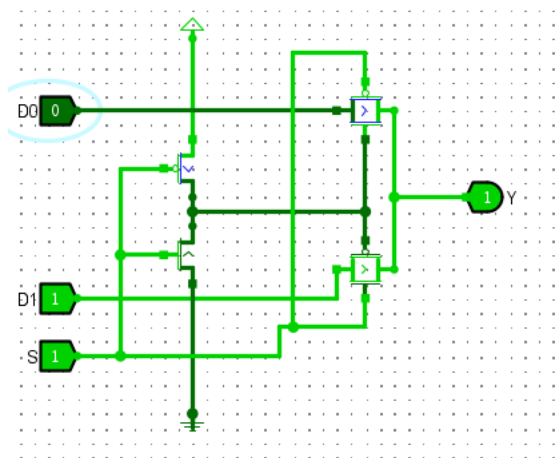
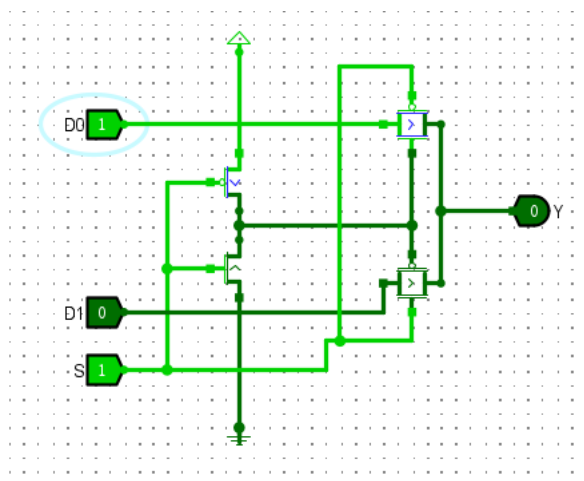


电路图：



仿真测试图：





真值表：

D0	D1	S	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

5、使用 2 路选择器子电路构建一个 4 路选择器。

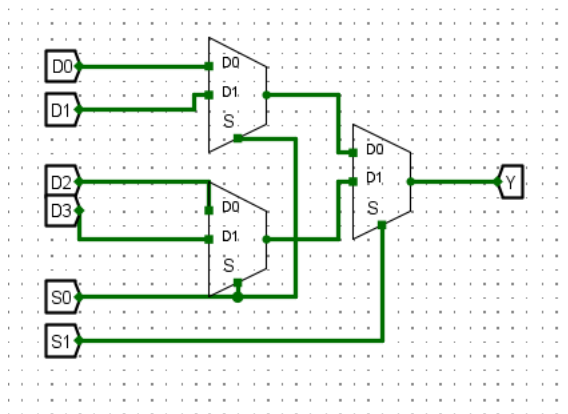
整体模块设计：

使用 3 个 2 路选择器，实现 4 路选择器。

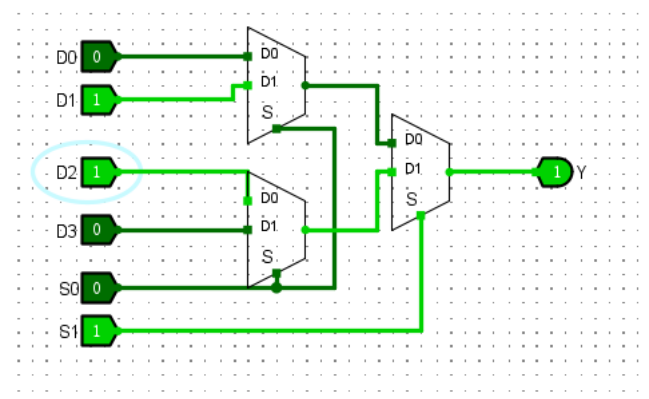
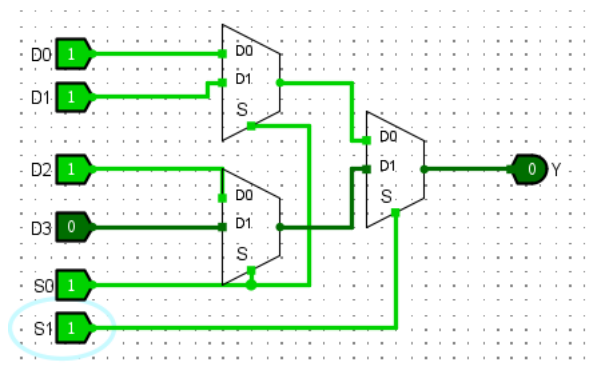
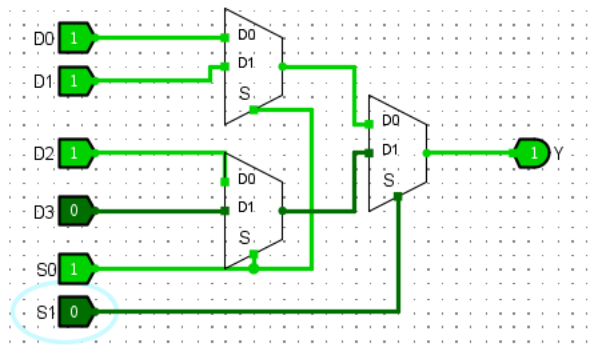
输入输出引脚：

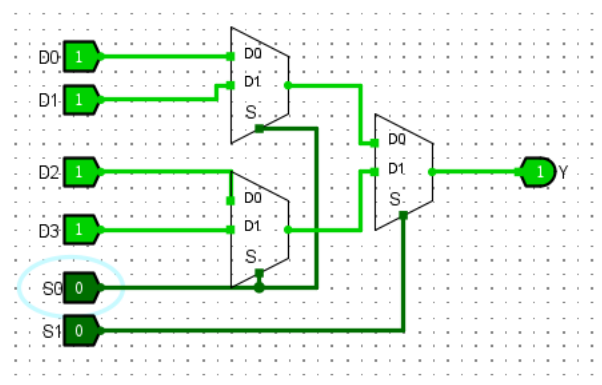
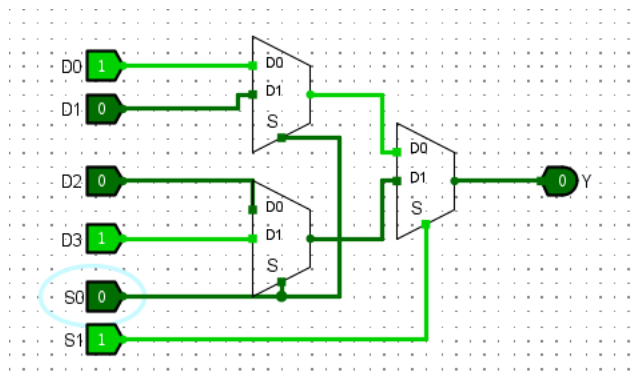
D0,D1,D2,D3 为输入端，S0, S1 为控制端，Y 为输出端

原理图和电路图：（二者合一）：



仿真测试图:

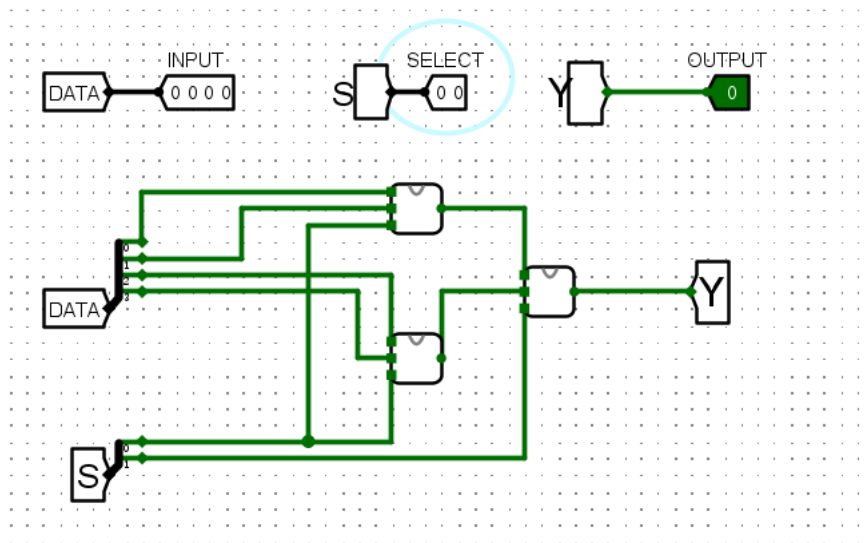




真值表：

6、隧道和集线器部件实验。

原理图和电路图：



三、错误现象及分析

在完成实验的过程中，没有遇到任何错误。

四、思考题

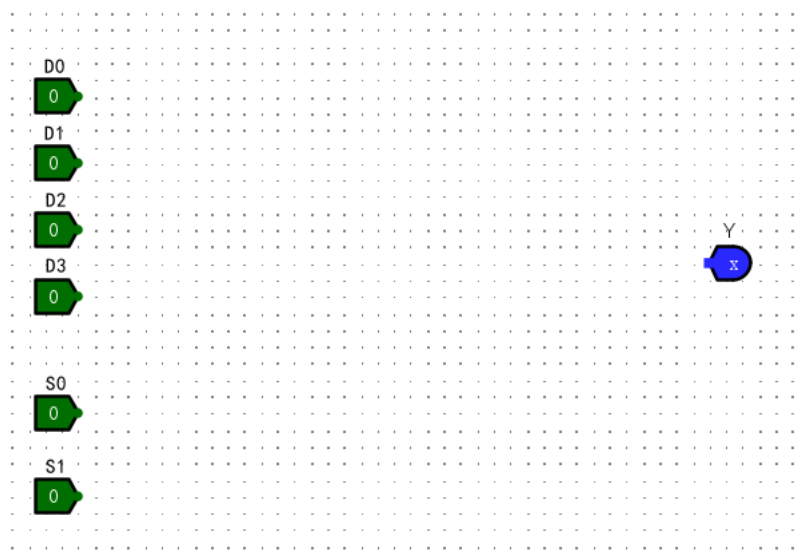
- 1、根据 Logisim 组合电路分析的功能，使用逻辑表达式设计方法选择与非门生成 4 选 1 多路选择器。

输入输出引脚作用：

D0,D1,D2,D3 为输入端，S0，S1 为控制端，Y 为输出端

实验过程：

设计好输入输出引脚：



输入逻辑表达式：

输出: Y

$D3 S0 S1 + D2 \overline{S0} S1 + D1 S0 \overline{S1} + D0 \overline{S0} \overline{S1}$

$D3 S0 S1 + D2 \sim S0 S1 + D1 S0 \sim S1 + D0 \sim S0 \sim S1$

清空 恢复 输入

生成电路

选择仅使用与非门生成：

生成电路

目标电路: 无标题

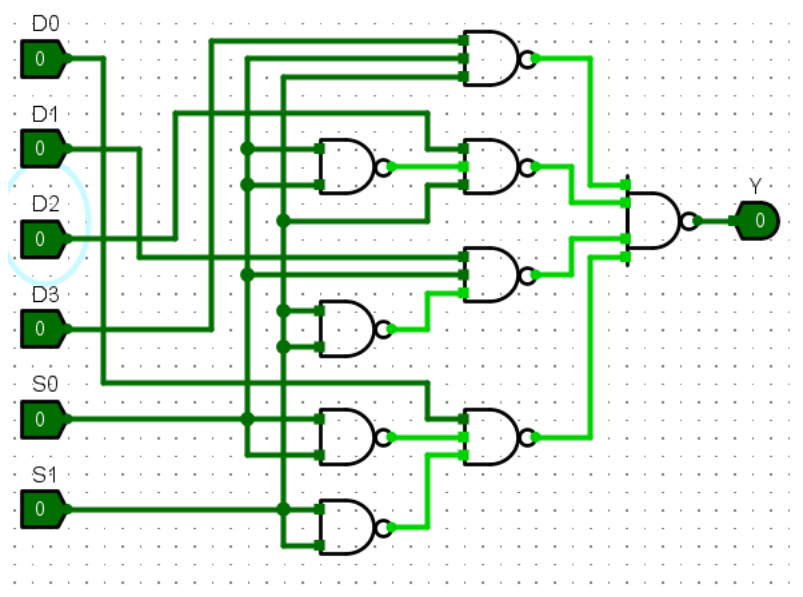
电路名: main

☐ 仅使用两输入逻辑门

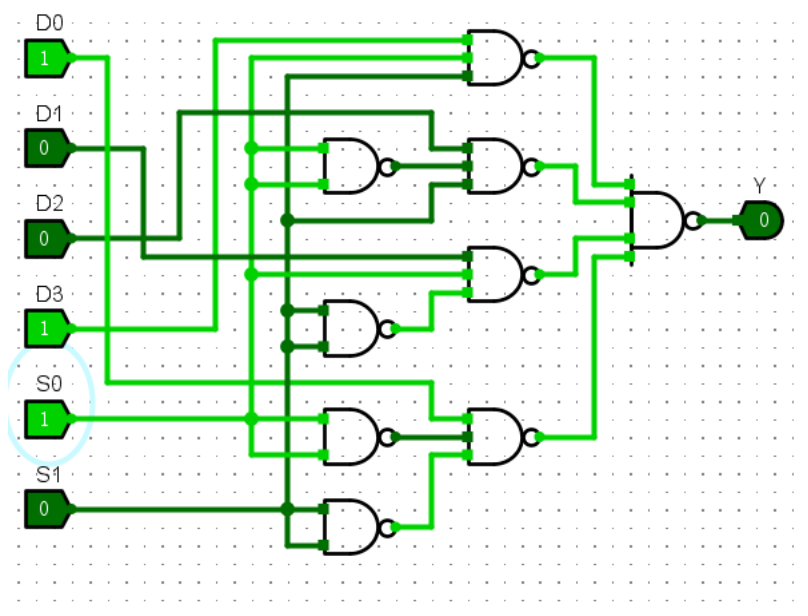
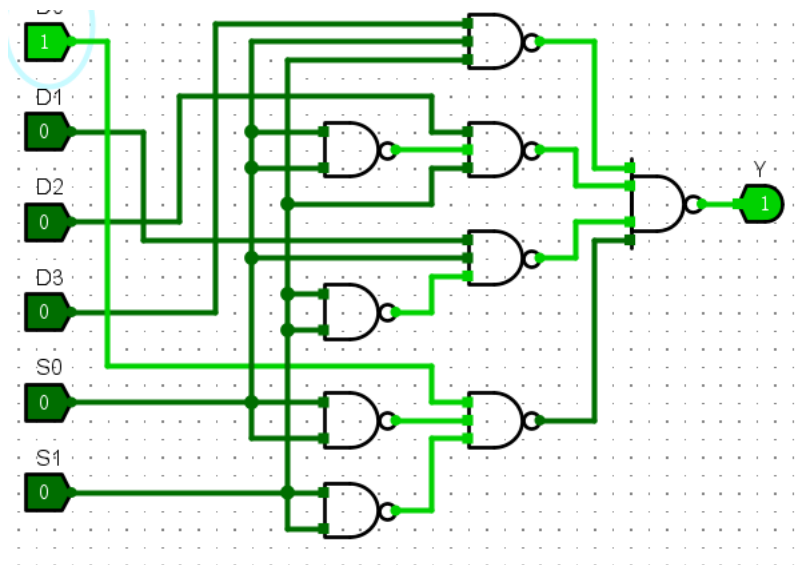
☒ 仅使用与非 (NAND) 门

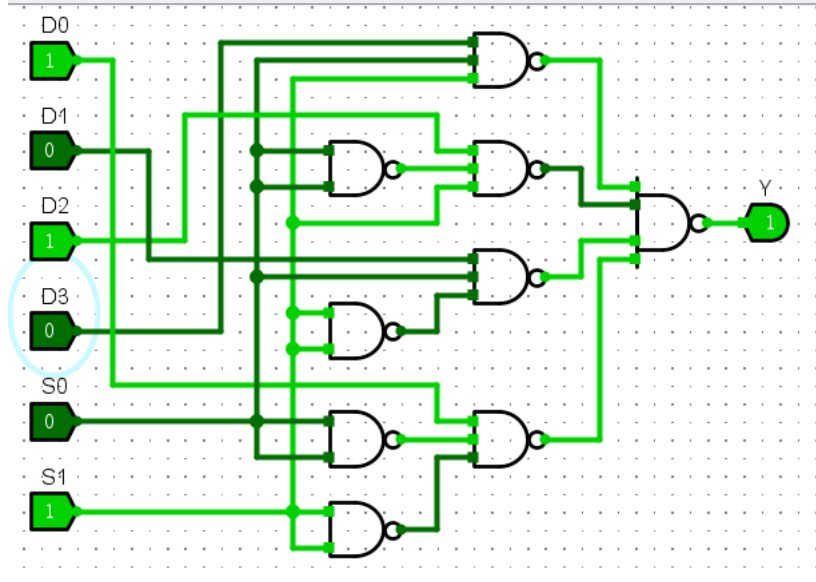
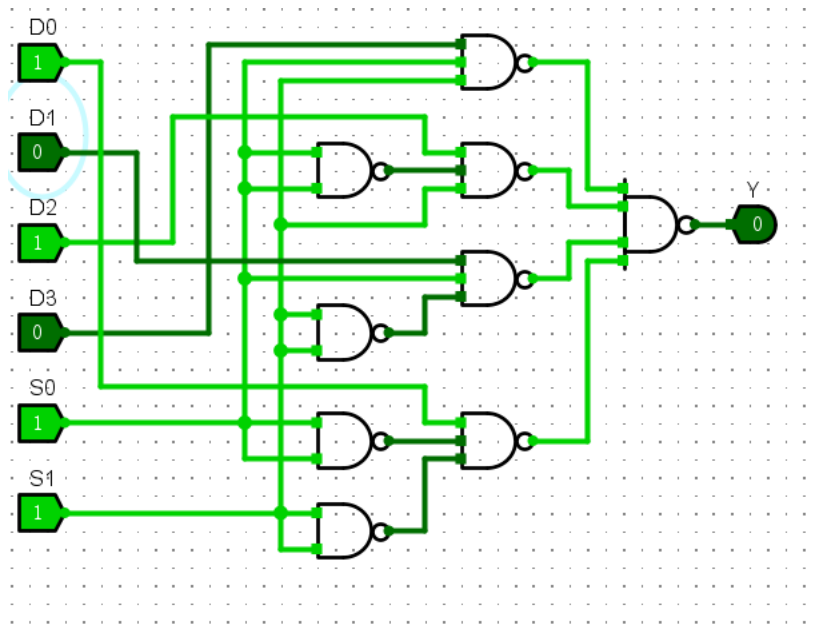
确定 取消

电路图：



仿真测试图：





真值表：

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

2、实现 4 位二进制数转换成格雷码的转换电路。

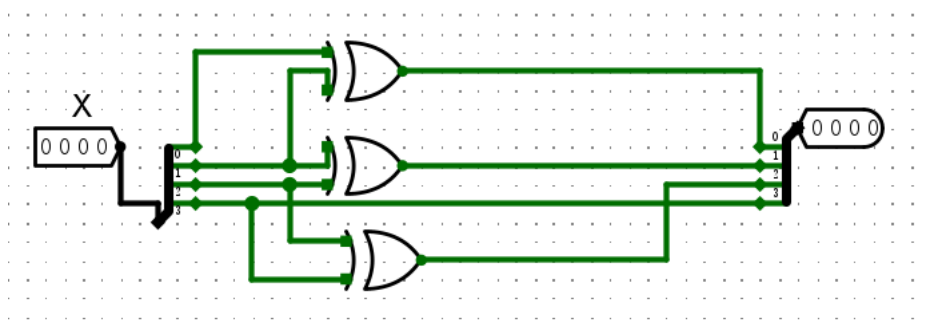
整体模块设计：

使用三个异或门通过逐位推进异或来实现二进制与格雷码的转换。

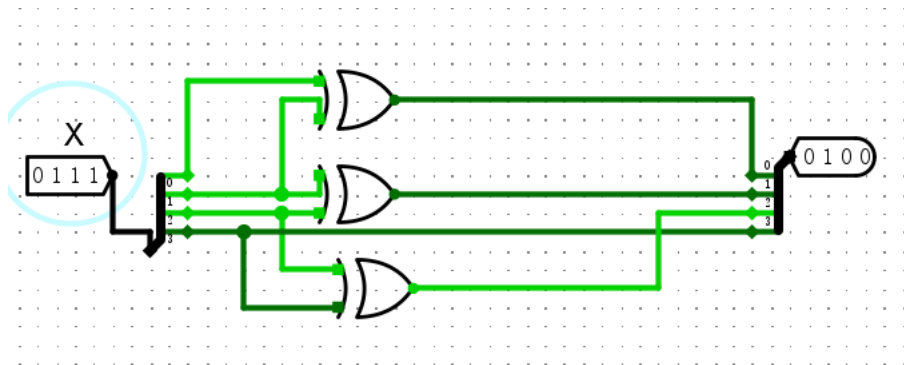
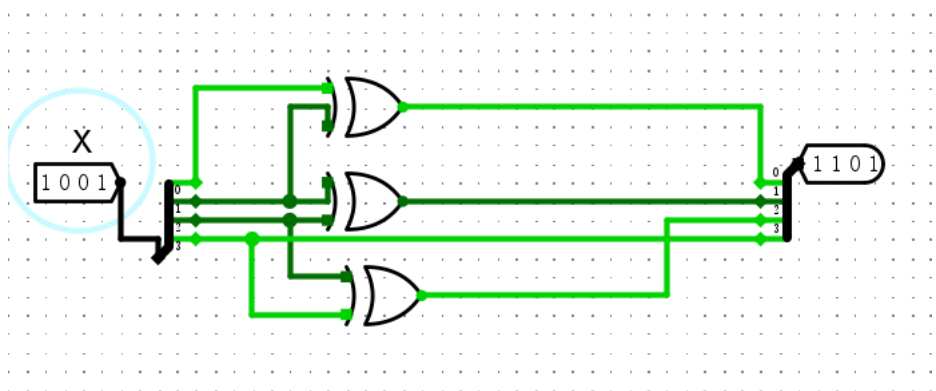
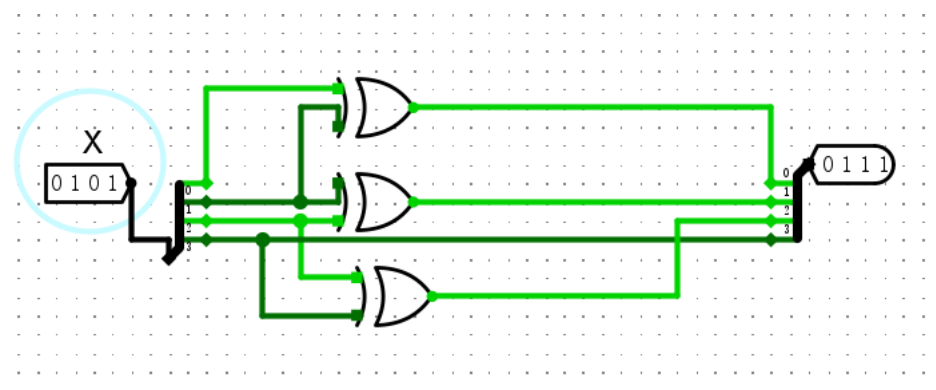
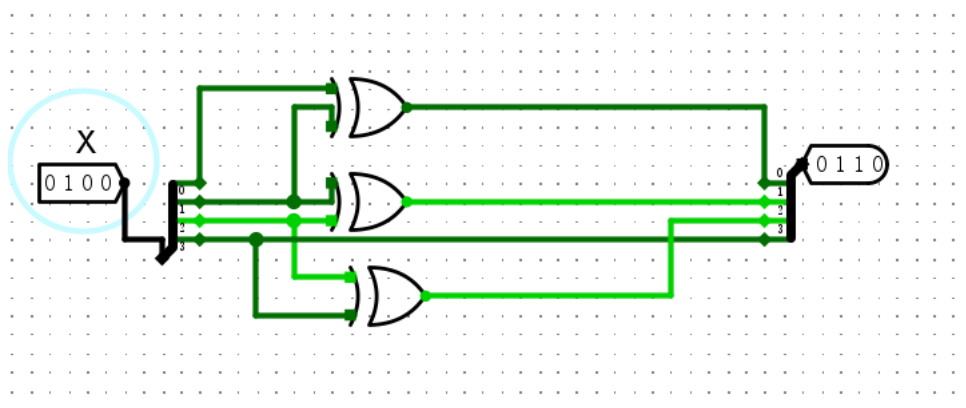
输入输出引脚作用：

X 为输入端，Y 为输出端。

原理图和电路图（二者合一）：



仿真测试图:



真值表：

十进制	二进制	格雷码
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

3、实现 4 位二进制数的奇偶校验位生成电路。

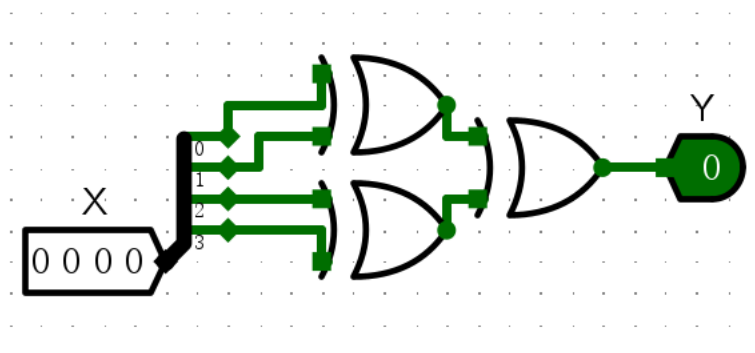
整体模块设计：

运用三个异或门实现 4 位二进制数的奇偶校验位生成电路，当有奇数个 1 时输出 1，偶数个 1 时输出 0。

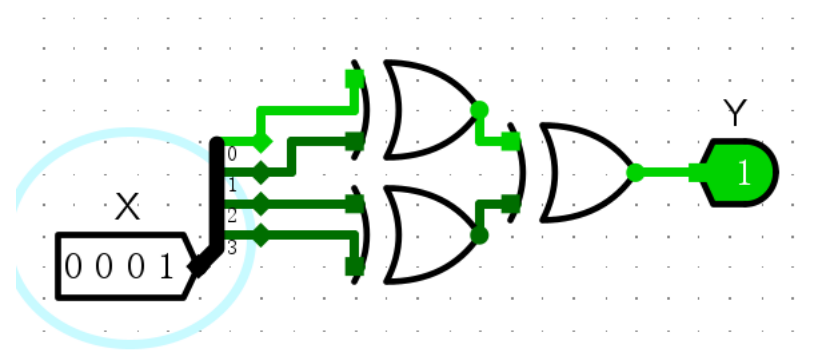
输入输出引脚作用：

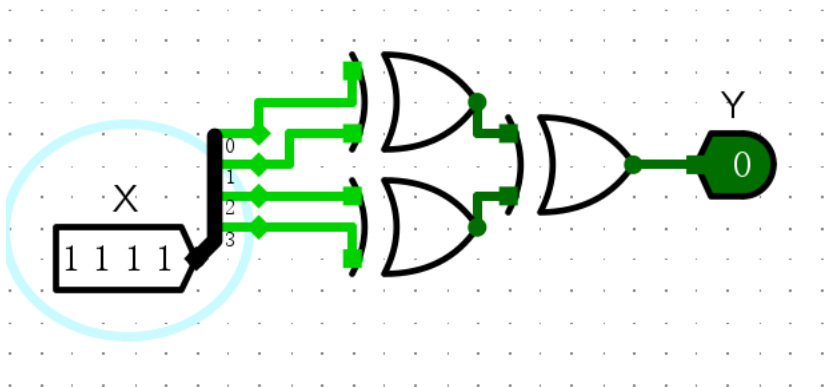
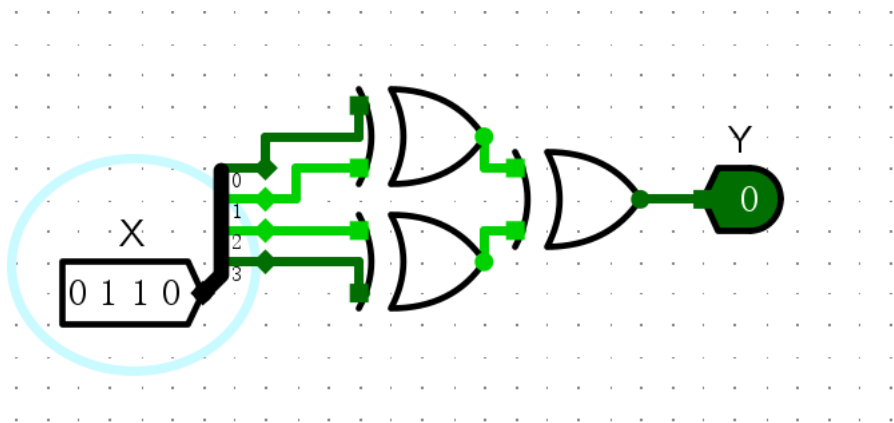
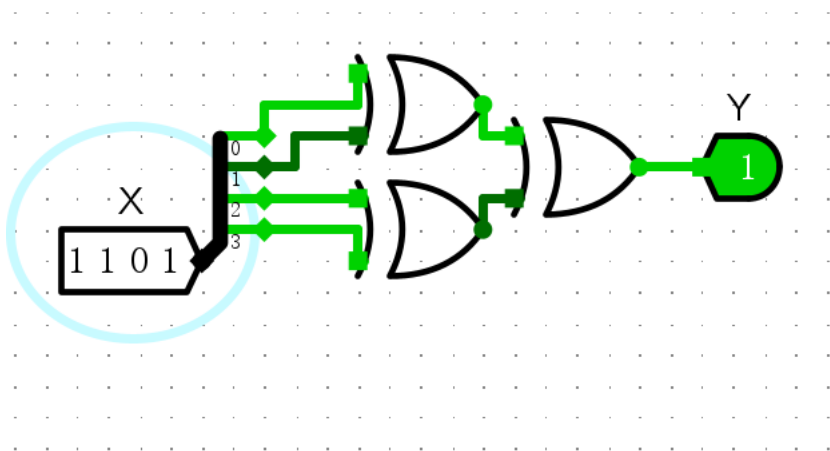
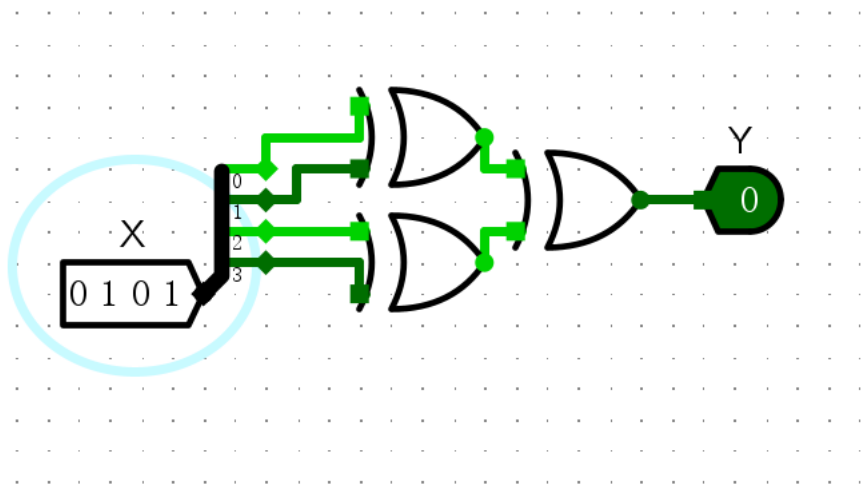
X 为输入端，Y 为输出端。

原理图和电路图（二者合一）：



仿真测试图：





真值表：

十进制	二进制	输出Y
0	0000	0
1	0001	1
2	0010	1
3	0011	0
4	0100	1
5	0101	0
6	0110	0
7	0111	1
8	1000	1
9	1001	0
10	1010	0
11	1011	1
12	1100	0
13	1101	1
14	1110	1
15	1111	0