

Report

介绍

该程序有 5 个可执行的操作，分别为**创建门**，**连接两个门**，**列出已添加的所有门的信息**，**设定输入**，和**运行**

1. **创建**：选择创建后，对要创建的门进行选择，一共有四种，与门、或门、非门、异或门，选择并命名后即创建成功

Ps.重复名称会导致创建失败

2. **连接**：选择连接后，在界面上弹出的“From“为提供**输入**的门，在后面输入该门的**名字**，“To”为**被输入**的门的**名字**，同样输入对应的门的**名字**。若有不存在的门的**名字**，或是输入与输入相连，则连接失败。

Ps.默认先连 input1，若 input1 已连接，则改连 input2。若都被连接，则连接失败

3. **列表**：列出所有门的信息，有名字，类型，输入和输出
4. **输入**：先输入想要设定输入的门的**名字**，再输入对应的两个输入，如果输入已经和其他门连接，则输入失败。
5. **运行**：得出结果，若有多输出，则需根据输出的门的**名字**自行判断

代码部分

三个文件（运用了继承、封装、多态，没有使用运算符重载和异常）

gate.h：该文件中有五个类，一个抽象类 Gate，以及四个派生类 AndGate, OrGate, NotGate, XorGate。数据成员中有 3 个 node 结构的数据，分别代表对象拥有的两个输入一个输出；2 个 string 类型的成员，name 为创建对象的名字，type 为它的类型。

operate.h：该文件主要包含了用户界面相关的函数，以及一个用于运行电路的递归函数。用了两个 map 一个 vector，其中 obj 是通过名字访问该对象，togate 是为了通过输出访问对象，glist 是记录所有门的名字。

main.cpp：最基础的用户交互界面

注意事项

1. 输入时请输入 0 和 1，因为本程序没有对输入是否合理进行判断
2. 在选择创建门的类型的界面时，切勿输入-1024，因为该数字比较特殊……
3. 该程序只能检查简单的环异常（即一个门的输入与输入连接），以

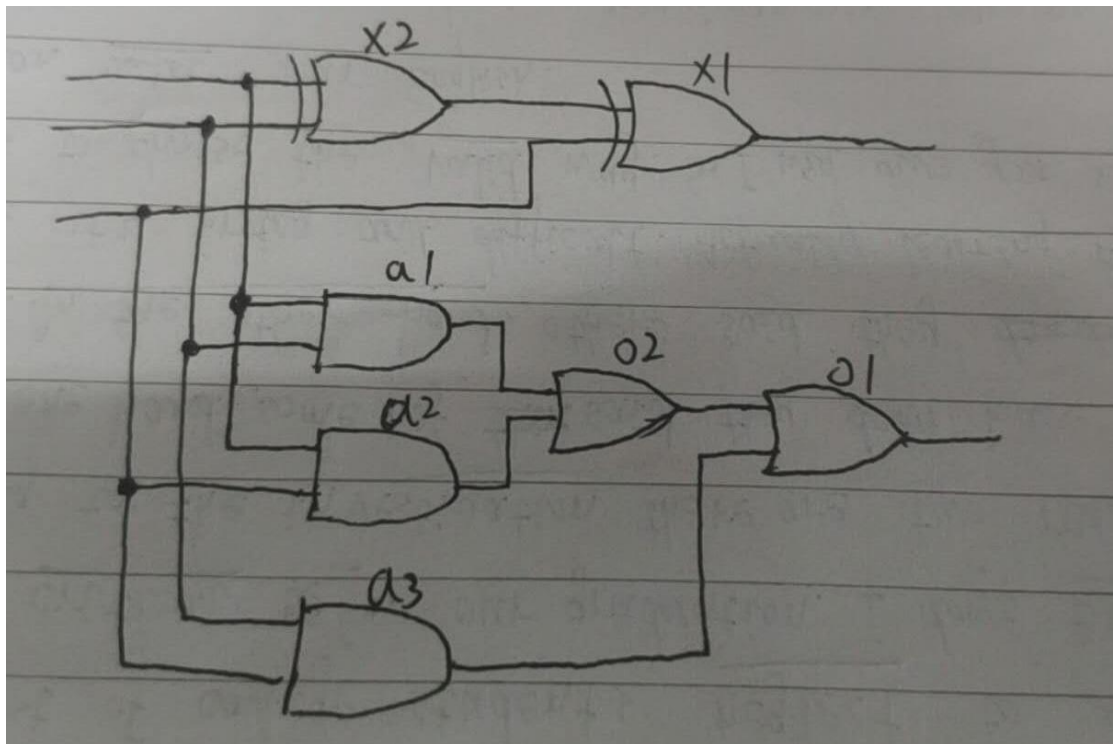
及可以检查没有输出的电路的环异常

4. 输入选项时切勿输入数字以外的字符，因为使用的变量是 int 的，没有修改（懒癌……）

例子

全加器为例

效果图（手绘版）



输入如下：

1 4 x1 1 4 x2 1 2 o1 1 2 o2 1 1 a1 1 1 a2 1 1 a3

2 x2 x1 2 a1 o2 2 a2 o2 2 o2 o1 2 a3 o1

4 x2 1 1 4 x1 0 4 a1 1 1 4 a2 1 0 4 a3 1 0

5

输出结果如下：

```
Type a number to perform the operation!
1-create a gate
2-connect a gate to other gate
3-List all the current gates
4-setinput
5-Running the current connection
Enter the number:5
The x1's output is : 0
The o1's output is : 1
```

x1 是本位，o1 是进位（输入为 1+1，进位为 0）

接着输入：

4 x2 1 1 4 x1 1 4 a1 1 1 4 a2 1 0 4 a3 1 1

5

输出为：

```
Type a number to perform the operation!
1-create a gate
2-connect a gate to other gate
3-List all the current gates
4-setinput
5-Running the current connection
Enter the number:5
The x1's output is : 1
The o1's output is : 1
```

上述操作即为改变相加两位和进位的输入，为 1 1 1

总结

大概能想到的就是上面那些内容。

本来是打算做 print 功能，将电路图打印出来，可是最后没想到合适

的方法，就放弃了。工作量上确实要比期中的小，但是一开始不太好下手，后面咨询了下大佬才有的思路（一开始没想到 STL）