程序设计II 期末Project - Simple Circuit实验报告

代码运行

依赖的库文件

除基本库 iostream、iomanip、string、sstream 外，此项目还引用了如下库：

Windows.h - 控制程序挂起，但只是用于最后程序退出时。若老师（师兄）在非windows环境下编译，可以将global.cpp中包含的此头文件以及函数void exiting()中将Sleep(5000)注释掉，不影响整个程序。带来不便抱歉！

map - STL容器使用，用于存储Gate类对象指针并建立索引。

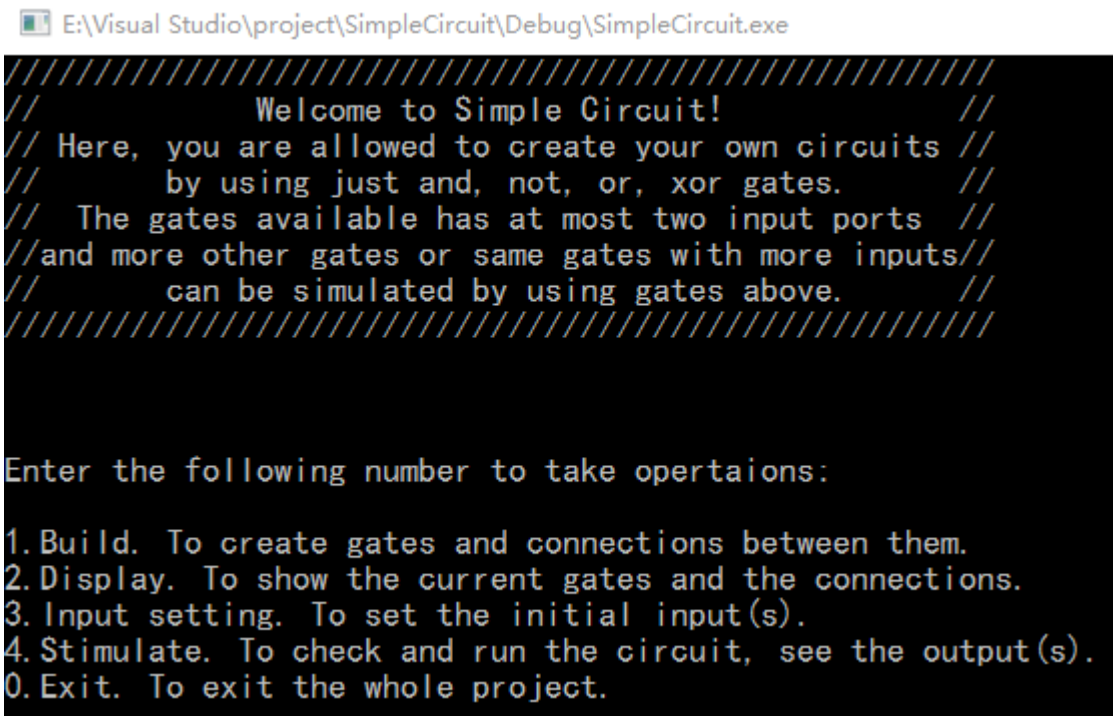
exception - 异常。用于检查电路逻辑错误并报错。

编译运行

项目在Windows Visual Studio 2017平台上~~大材小用地~~编写、编译。新建项目后添加源文件，由平台编译链接，再进行调试与运行、模拟。

使用方法

此命令行项目中每一步操作都附有完整清晰的操作说明和选择指引，用户通过键入操作相关编号进行选择、操作。在电路连接板块，用户通过先后键入两门并选择端口对两门进行连接。完成输入设定后即可进行检查，检查通过后即可运行，查看输出情况。所有过程均可在界面得到清晰的指引。



设计思路

对象封装

项目中最主要的封装对象是基本门对象。

父类**Gate**不代表任何元器件，但可作为基类定义各个共有的成员。

子类共有六类，分别为**与**、**非**、**或**、**异或门**以及**输入输出端口**。

各类的基本成员有：input1(&2)为该门的输入、output为该门的输出（以map存储，可有多个）、门名称、类型、门值的设定和储存。

继承的应用

实现中以Gate为基类，其他各类门（共六种）均由Gate父类public继承而来，方便了各种门的定义以及实现用父类指针访问子类对象的操作，在器件存储表示上均带来很大便利。

多态的应用

运用了继承的思想和纯虚函数，实现通过一种指针，即父类指针，完成对子类及子类成员的访问操作。

项目中用map<string, Gate*>存储电路中用户添加的所有门，并注意遍历、访问、修改、设置。

运算符重载的应用

实现中，简单地将Gate类的赋值运算符=重载为代表其门类型的int类变量的赋值运算，在新建门时可实现直接通过赋值完成门的类型表示与存储。

异常的应用

电路项目中，电路检查是重要的。因有了异常的知识基础，检查中抛弃了原本繁琐的判定结构，换以执行check()中遇错即抛出异常并进行报告的方式，使代码结构更加清晰可控，同时使查错报错功能变得更加完善、优化。

拓展

门设置

项目中主要的门器件除了最基本的与、或、非、异或之外，还设置了两种端口，虽然同样为Gate的子类，但并不是严谨意义上的门，而更像是Vivado中的I/O PORTS，方便用户设置输入输出端口，使电路更加清晰。同时，也方便了电路检查，便于程序检查出端口悬空和屏蔽未用器件。

异常检查

项目中除了基本的环检查外，还有以下错误检查：

对用户忽略的、未设置的端口进行检查，若有端口处于悬空状态将报错并报告悬空器件名称；

若用户未设置输入，将报错并提醒设置输入情况；

其他细节

项目中对一些细节问题进行了优化：

每一步操作都详细到位的介绍指引以及操作是否成功实现的信息；

添加了连线切断功能，门之间的连接可以多次修改，提高了容错率；

若用户所添加的门未参与电路运行，则会自动屏蔽该门而不会对电路产生影响；

display功能实现所有门连接情况：输入端及输出端的展示；

……

不足之处

界面

电路连接情况涉及**图**的相关知识，由于知识储备不足且未有较好的展示方法实现，项目中的电路展示只依靠简单地罗列和说明，不够直观具体。命令行版本在操作和观感上也较为一般，还有很多可以优化、提升“用户体验”的地方。

实现方法

电路的连接、检查采用的是比较“直白”、“傻瓜式”的算法，对时间空间的优化不足，只能实现基本功能，检查出基本的错误，还不算合格的简易版Circuit。

更完整的功能 更高级的电路

这次项目中，~~小心地避开了~~尚未涉及时序这一块的电路知识，只是停留在初级阶段，还可以更进一步进阶、完整。

代码架构与风格

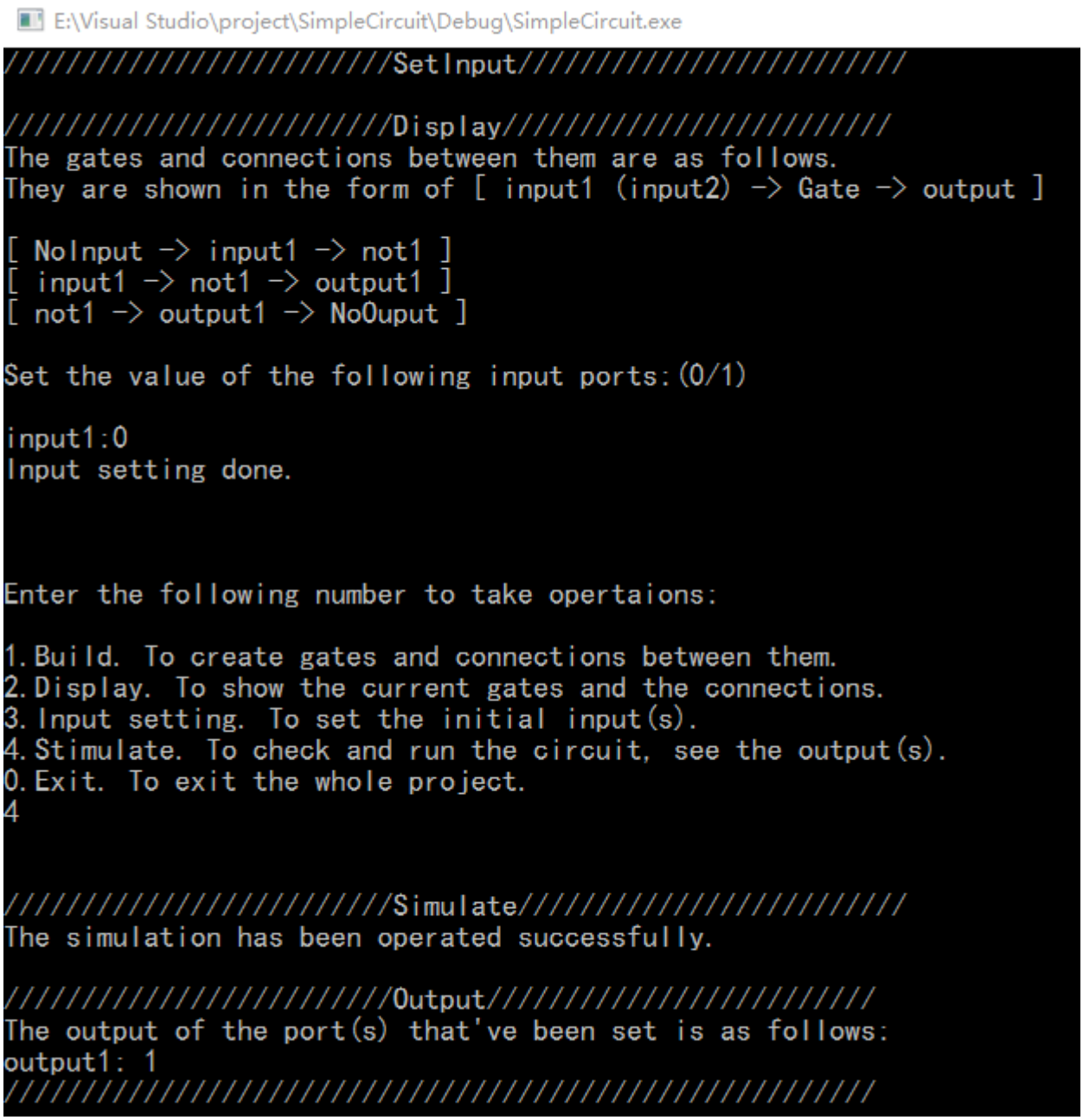
这是继Nethack后第二个c++项目，在对象封装、代码风格上可能还存在不少不规范的地方。源文件板块划分，全局、局部成员的运用还不熟悉，在多个源文件的联动上浪费了不少功夫，能力有待提升。

测试样例

最简单的非电路测试

> 1 1 2 5 6 0 1 2 input1 not1 not1 output1 0 0 3 X 4

（x=0/1，最终输出应为X取反）



环异常电路测试

> 1 1 1 2 5 6 0 1 2 input1 and1 1 and1 not1 not1 and1 2 not1 output1 0 0 3 0 4

（最终应报错电路中存在环异常）

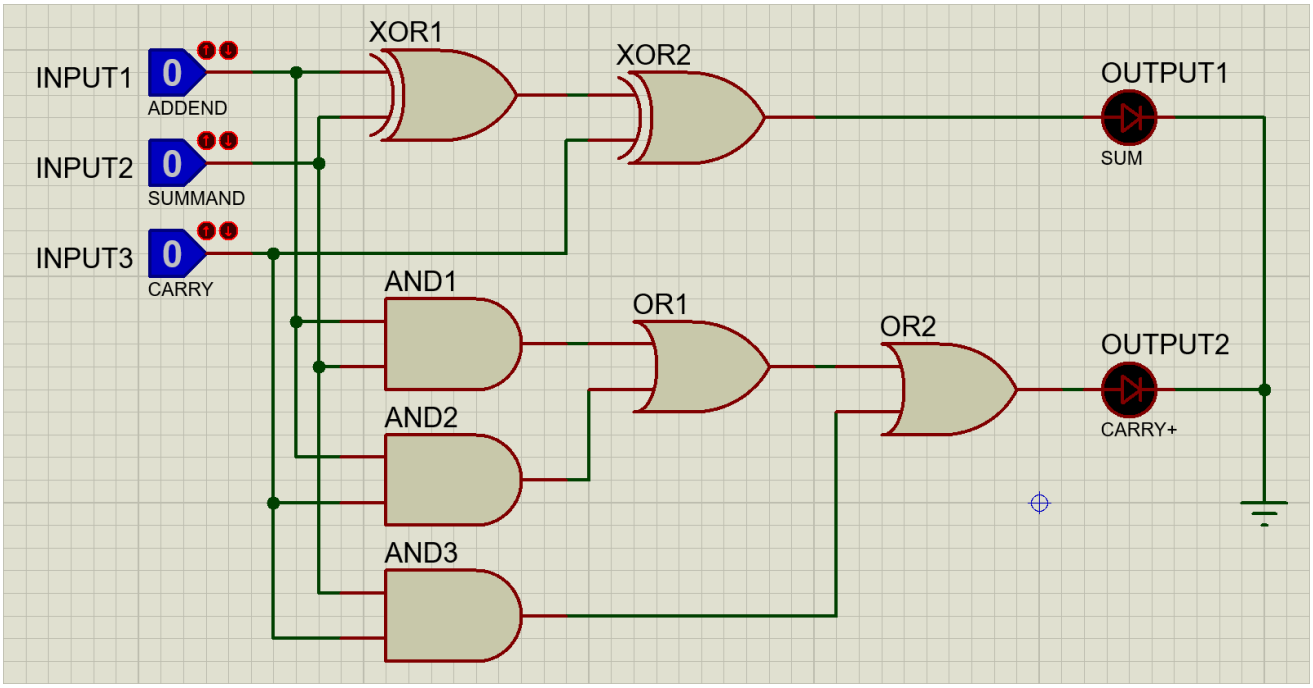
E:\Visual Studio\project\SimpleCircuit\Debug\SimpleCircuit.exe

```
Enter the following number to take opertaions:

1.Build. To create gates and connections between them.
2.Display. To show the current gates and the connections.
3.Input setting. To set the initial input(s).
4.Stimulate. To check and run the circuit, see the output(s).
0.Exit. To exit the whole project.
4


//////////////////////////Simulate//////////////////////////
There exits an error in your circuit:
There is a circle in your circuit, which is not allowed.
Correct it and try it again.
```

一位全加器测试



1 1 1 1 1 3 3 4 4 5 5 5 6 6 0 1 2 input1 xor1 1 input2 xor1 2 xor1 xor2 1 input3 xor2 2 xor2 output1 input1 and1 1 input2 and1 2 input1 and2 1 input3 and2 2 input2 and3 1 input3 and3 2 and1 or1 1 and2 or1 2or1 or2 1 and3 or2 2 or2 output2 0 0 3

//接下来三位X分别为两加数(0/1)及前一位全加器的进位(0/1)

X

X

X

4

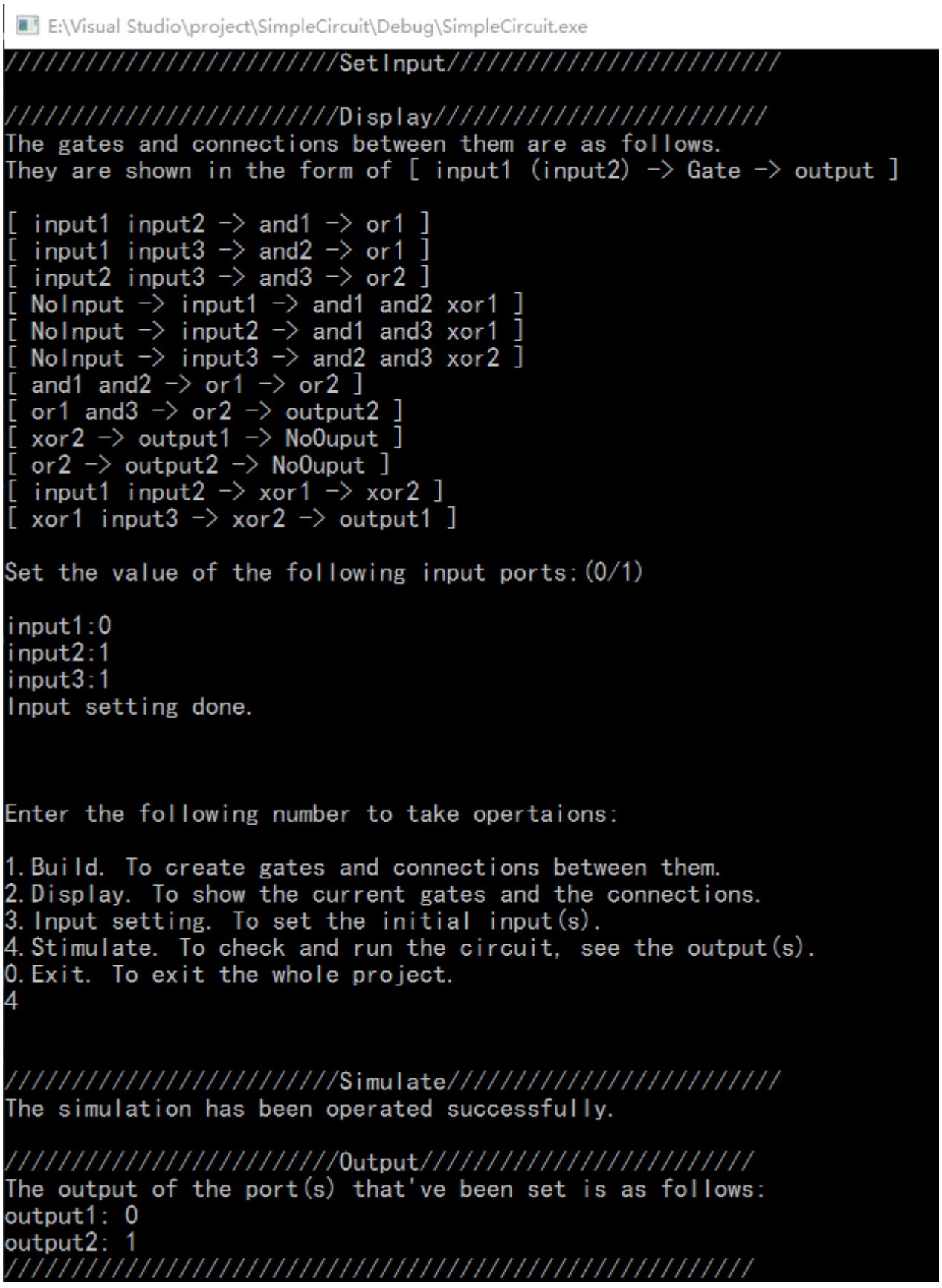
（输出有两位，分别为本位加法结果及进位输出。若XXX为000则输出为00，若XXX为010则输出10，若XXX为110则输出01，若XXX为111则输出为11……）

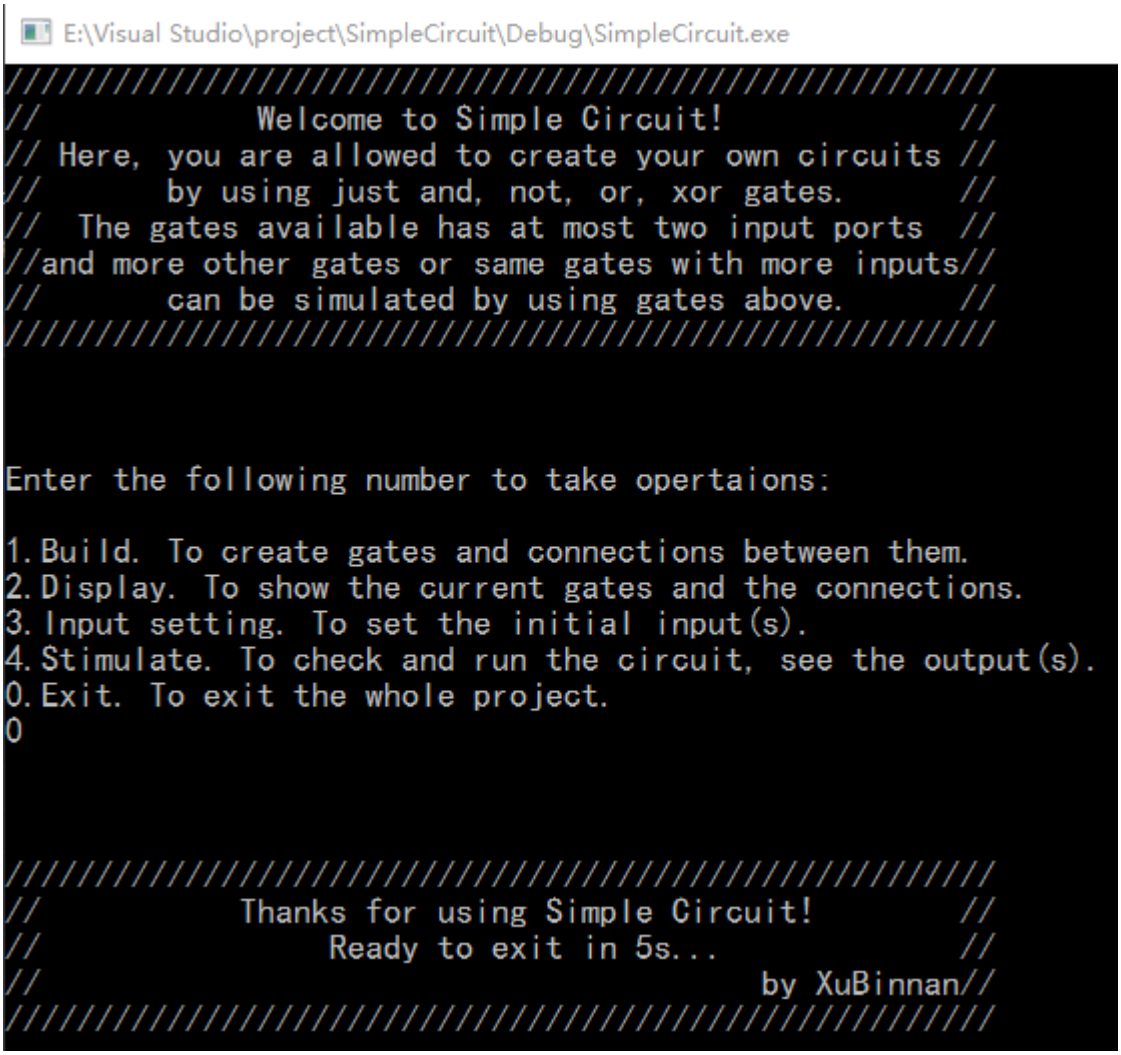
附无注释输入供复制（以011为输入）：

1 1 1 1 1 3 3 4 4 5 5 5 6 6 0 1 2 input1 xor1 1 input2 xor1 2 xor1 xor2 1 input3 xor2 2 xor2 output1 input1 and1 1 input2 and1 2 input1 and2 1 input3 and2 2 input2 and3 1 input3 and3 2 and1 or1 1 and2 or1 2 or1 or2 1 and3 or2 2 or2 output2 0 0 3

0 1 1

4


E:\Visual Studio\project\SimpleCircuit\Debug\SimpleCircuit.exe
//////////////////////SetInput//////////////////////
//////////////////////Display//////////////////////
The gates and connections between them are as follows.
They are shown in the form of [input1 (input2) -> Gate -> output]
[input1 input2 -> and1 -> or1]
[input1 input3 -> and2 -> or1]
[input2 input3 -> and3 -> or2]
[NoInput -> input1 -> and1 and2 xor1]
[NoInput -> input2 -> and1 and3 xor1]
[NoInput -> input3 -> and2 and3 xor2]
[and1 and2 -> or1 -> or2]
[or1 and3 -> or2 -> output2]
[xor2 -> output1 -> NoOuput]
[or2 -> output2 -> NoOuput]
[input1 input2 -> xor1 -> xor2]
[xor1 input3 -> xor2 -> output1]
Set the value of the following input ports:(0/1)
input1:0
input2:1
input3:1
Input setting done.
Enter the following number to take opertaions:
1.Build. To create gates and connections between them.
2.Display. To show the current gates and the connections.
3.Input setting. To set the initial input(s).
4.Stimulate. To check and run the circuit, see the output(s).
0.Exit. To exit the whole project.
4
//////////////////////Simulate//////////////////////
The simulation has been operated successfully.
//////////////////////Output//////////////////////
The output of the port(s) that've been set is as follows:
output1: 0
output2: 1
//


E:\Visual Studio\project\SimpleCircuit\Debug\SimpleCircuit.exe
//
//   Welcome to Simple Circuit!   //
// Here, you are allowed to create your own circuits //
//   by using just and, not, or, xor gates.   //
// The gates available has at most two input ports   //
//and more other gates or same gates with more inputs//
//   can be simulated by using gates above.   //
//
Enter the following number to take opertaions:
1.Build. To create gates and connections between them.
2.Display. To show the current gates and the connections.
3.Input setting. To set the initial input(s).
4.Stimulate. To check and run the circuit, see the output(s).
0.Exit. To exit the whole project.
0
//
//   Thanks for using Simple Circuit!   //
//   Ready to exit in 5s...   //
//   by XuBinnan//
//