中山大学数据科学与计算机学院本科生实验报告

(2018 年春季学期)

课程名称：程序设计　　任课教师： 黄方军

年级	17 级	专业（方向）	保密管理
学号	17340004	姓名	韩京

一、 实验题目

期末 project——Simple Circuit

要求：

在高中的时候（甚至初中），我们就已经简单的接触过电路知识，也知道了与、或、非、异或等一些基础门电路知识。这一次，我们就来简单的实现一个命令行版的门电路实验程序。

二、 实现内容

基础要求：

至少运用本学期学过的：运算符重载、继承、多态、异常、封装

至少实现与、非、或、异或门

要能完成基本的用户交互。比如指定哪两个门连接，等等

连接完成后，可以给定输入来获取输出

至少能够检查出简单的环异常

能够正确的连接，从而组成复杂组件，提交时请提供设计的用于测试的输入（比如连接成一个全加器）

```
class data {
    private:
        int value;
    public:
        void setValue(int i) {
            value = i;
        }
        int getValue() {
            return value;
        }
```

实现一个 data 类，成员变量是 value，两个成员函数 setValue 给属性赋值，由 getValue 返回值

```
data operator*(const data& d) { //AND
    data da;
    if(d.value == value == 1) da.value = 1;
    else da.value = 0;
    return da;
}
data operator+(const data& d) { //OR
    data da;
    if(d.value == value == 0) da.value = 0;
    else da.value = 1;
    return da;
}
data operator!() { //NOT
    data da;
    if(value) da.value = 0;
    else da.value = 1;
    return da;
}
data operator-(const data& d) { //XOR
    data da;
    if(d.value == value) da.value = 0;
    else da.value = 1;
    return da;
}
```

利用运算符重载，重载*运算符来代表 AND，重载+运算符来代表 OR，重载！来代表 NOT，重载-来代表 XOR

```
class gate:public data {
    private:
        int input1,input2;
        data output;
        string name;
```

data 的派生类 gate 类，input1，input2 用于输入门的编号，output 是通过输入计算后的结果，name 为门的名字

```
void calculate(data i1,data i2) {
    if(name == "AND") {
        output = AND(i1,i2);
    }
    else if(name == "OR") {
        output = OR(i1,i2);
    }
    else if(name == "NOT") {
        output = NOT(i1);
    }
    else if(name == "XOR") {
        output = XOR(i1,i2);
    }
    else if(name == "NAND") {
        output = NOT(AND(i1,i2));
    }
}
```

```
data AND(data in1,data in2) {
    data d;
    d = in1 * in2;
    return d;
}
data OR(data in1,data in2) {
    data d;
    d = in1 + in2;
    return d;
}
data NOT(data in1) {
    data d;
    d = !in1;
    return d;
}
data XOR(data in1,data in2) {
    data d;
    d = in1 - in2;
    return d;
}
```

函数通过判断门的名字进行对应门的计算

```
void setName(string name) {
    this->name = name;
}
string getName() {
    return name;
}
int getInt1() {
    return input1;
}
int getInt2() {
    return input2;
}
data getOut() {
    return output;
}
```

setName 用于给 name 赋值，getName 返回 name 的值，getInt1 用于返回 input1 的值，getInt2 用于返回 input2 的值，getOut 用于返回 output 的值

```
int main() {
//  data d1;
//  d1.setValue(1);
//  data d2;
//  d2.setValue(0);
//  data d3 = d2+!d1+d1;
//  cout<<d3.getValue();
    gate g[100];
    data d[100];
```

定义两个数组，用于存储 gate 和 data

```
for(int i =0;i<100;i++) {
    d[i].setValue(-1);
}
int n,input1,input2;
string name;
```

把 data 数组中的所有 value 的属性设置为-1

```
cout<<"Input total number of gates."<<endl;
cin>>n;

for(int i = 0;i < n;i++) {
    cout<<"Input names of gates,"<<"remain "<<n-i<<endl;
    cin>>name;
    if(name!="AND"&&name!="OR"&&name!="NOT"&&name!="XOR"&&name!="NAND") {
        cout<<"ilegal input."<<endl;
        i--;
    } else {
        g[i].setName(name);
    }
}
```

变量 n 控制输入门器件的个数
循环输入门的名字如 AND,OR,NOT 等

输入之后判断输入是否合法，如果不合法输出 **illegal**。

```
cout<<"Check out output interfaces of gates"<<endl;
for(int i = 0;i < n;i++) {
    cout<<i<<"."<<g[i].getName()<<" "<<endl;
}
```

打印门电路的编号和 **name**

```
bool check;

while(!check) {
    check = true;
    cout<<"Connect output interfaces to input of others,input 101 or 100 if you want 1 or 0 to be an input"<<endl;
    for(int i = 0;i < n;i++) {
        cout<<"To "<<i<<"."<<g[i].getName()<<" "<<endl;
        cin>>input1>>input2;
        if(99>input1>n||99>input2>n) {
            cout<<"out of range."<<endl;
            i--;
        } else {
            g[i].setIn(input1,input2);
        }
    }
    for(int i = 0;i < n;i++) {
        if(g[i].getInt1()==-1||g[i].getInt2()==-1) {
            cout<<"input of "<<i<<" is not full filled."<<endl;
            check = false;
        }
    }
    bool cnt = 0;
    for(int i = 0;i < n;i++) {
        if(g[i].getInt1()>99&&g[i].getInt2()>99) {
            cnt = 1;
        }
        if(g[i].getInt1()==i||g[i].getInt2()==i) {
            cnt = 1;
        }
    }
    if(!cnt) {
        cout<<"Loop circuit detected."<<endl;
        check = false;
    }
}
```

设置一个 **bool flag** 为 **check**

检查 **check** 是否为真，判断是否继续进行循环

接下来在循环中输入作为对应逻辑门输入的逻辑门编号，如果输入为默认的高低电平则输入 **101**，**100** 接下来判断是否有环路存在，这里采用了一个简单的方法，判断是否有两个输入均为 **100 101** 的逻辑门，若没有则可以判断出现了环路。

```
for(int i = 0;i < n;i++) {
    cout<<i<<"."<<g[i].getName()<<" "<<"input1:"<<g[i].getInt1()<<" input2:"<<g[i].getInt2()<<endl;
}
```

输出逻辑门编号和对应的输入的逻辑门的编号

```
bool tmp = 1;
while(!el) {
    el = 1;
    for(int i = 0;i < n;i++) {
        int i1 = g[i].getInt1();
        int i2 = g[i].getInt2();
        if(d[i].getValue()!=-1) continue;
        else {
            cout<<i<<"!";
            if(i1>99&&i2>99) {
                data d1;
                data d2;
                d1.setValue(i1-100);
                d2.setValue(i2-100);
                g[i].calculate(d1,d2);
                d[i] = g[i].getOut();
                cout<<i<<"."<<g[i].getName()<<": "<<d[i].getValue()<<endl;
            }
            else if(d[i1].getValue()!=-1&&i2>99) {
                data d2;
                d2.setValue(i2-100);
                g[i].calculate(d[i1],d2);
                d[i] = g[i].getOut();
                cout<<i<<"."<<g[i].getName()<<": "<<d[i].getValue()<<endl;
            }
```

设置一个 bool flag 为 el = 0，
检测 el 是否为真
循环 n 次，每次对 d[i]判断值是否存在，若存在则继续，如果不存在则进行下面的判断，对 g[i]两个输入进行判断，如果大于 99（即为 100 和 101）则将值转化为 data 类型，使用 gate 类自带的 calculate 函数进行运算，将计算的结果存到 d 数组中作为已知的值，后面的判断就是解决 input 是否为已知的问题

```
            else if(d[i2].getValue()!=-1&&i1>99) {
                data d1;
                d1.setValue(i1-100);
                g[i].calculate(d[i2],d1);
                d[i] = g[i].getOut();
                cout<<i<<"."<<g[i].getName()<<": "<<d[i].getValue()<<endl;
            }
            else if(d[i1].getValue()!=-1&&d[i2].getValue()!=-1) {
                data d1;
                data d2;
                g[i].calculate(d[i1],d[i2]);
                d[i] = g[i].getOut();
                cout<<i<<"."<<g[i].getName()<<": "<<d[i].getValue()<<endl;
            }
        }

}
```

同理判断两个值一个为 99+一个为已知值的情况
这里的关键在于输入 gate 的值为 in1 和 in2，而这两个 int 类型的变量存储的实际上是作为输入的逻辑门的编号。

```
        for(int i = 0;i < n;i++) {
            if(d[i].getValue() != -1) el =0;
        }
            for(int i = 0;i < n;i++) {
            if(d[i].getValue()==-1) tmp = 0;
        }
        if(tmp) {
            cout<<"successful running"<<endl;
            return 0;
        }
    }
}
```

最后将计算结果输出。这里存在的问题是如果出现了没有检测到的环路，程序会一直运行，但是也不会输出成功运行的信息，算是另一种检测环路的方法。

三、 实验结果

1.全为 AND

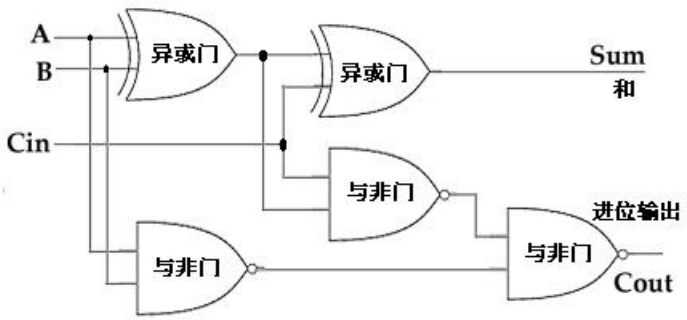
```
Input total number of gates.
3
Input names of gates,remain 3
AND
Input names of gates,remain 2
AND
Input names of gates,remain 1
AND
Check out output interfaces of gates
0.AND
1.AND
2.AND
Connect output interfaces to input of others,input 101 or 100 if you want 1 or 0 to be an input
To 0.AND
101 1
To 1.AND
101 100
To 2.AND
100 0
0.AND input1:101 input2:1
1.AND input1:101 input2:100
2.AND input1:100 input2:0
1.AND 0
0.AND 0
2.AND 1
```

2. 所有逻辑门各一个

```
选择C:\Users\hasee\Desktop\c\未命名1.exe
Input total number of gates.
5
Input names of gates,remain 5
AND
Input names of gates,remain 4
OR
Input names of gates,remain 3
NOT
Input names of gates,remain 2
XOR
Input names of gates,remain 1
NAND
Check out output interfaces of gates
0.AND
1.OR
2.NOT
3.XOR
4.NAND
Connect output interfaces to input of others,input 101 or 100 if you want 1 or 0 to be an input
To 0.AND
100 101
To 1.OR
100 101
To 2.NOT
100 101
To 3.XOR
100 101
To 4.NAND
100 101
0.AND input1:100 input2:101
1.OR input1:100 input2:101
2.NOT input1:100 input2:101
3.XOR input1:100 input2:101
4.NAND input1:100 input2:101
0.AND: 0
1.OR: 0
2.NOT: 1
3.XOR: 1
4.NAND: 1
successful running

--------------------------------
Process exited after 41.06 seconds with return value 0
请按任意键继续. . .
```

3. 全加器



$$S = A \oplus B \oplus C_{in}$$

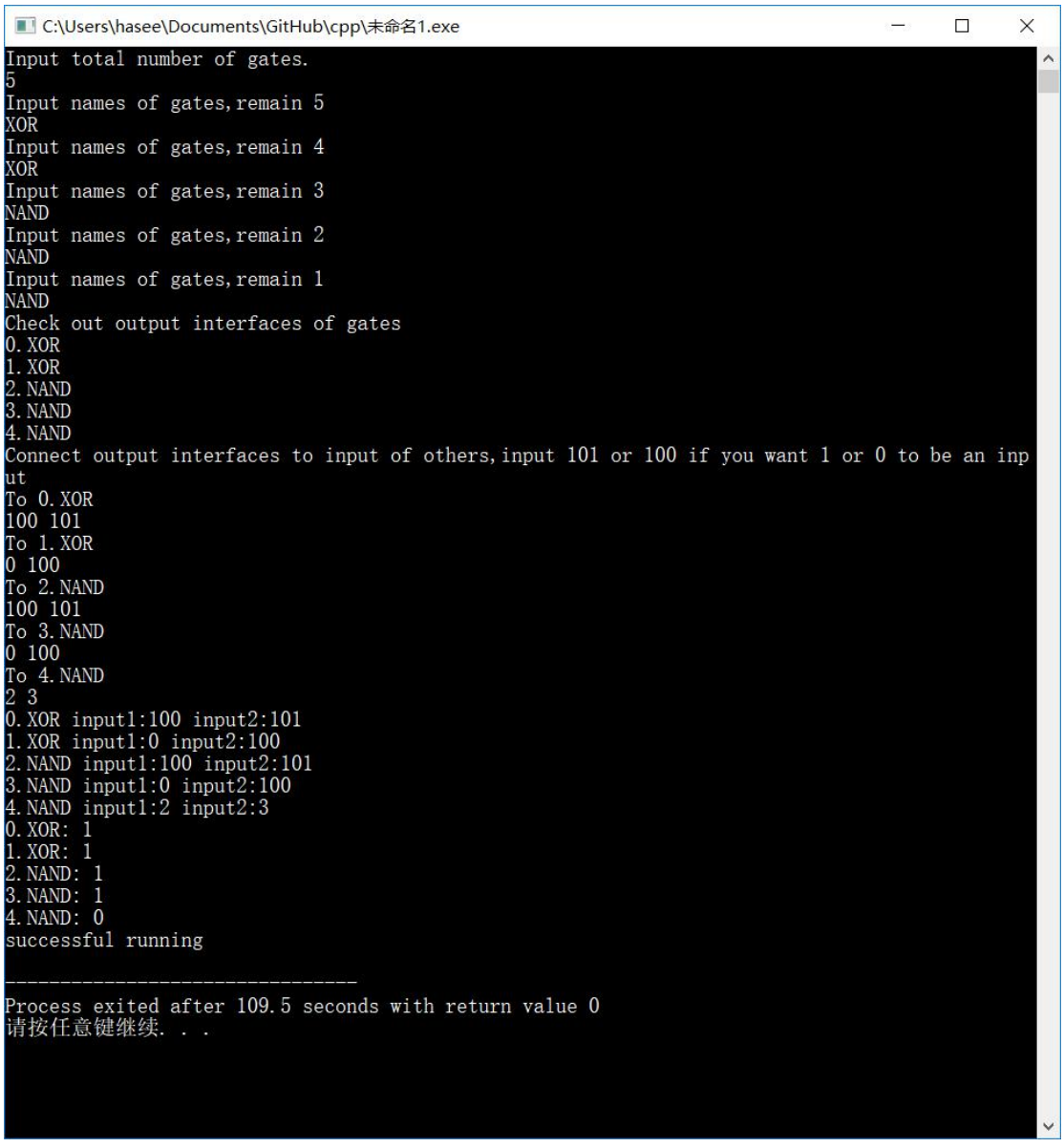
$$C_{out} = (A \cdot B) + (C_{in} \cdot (A \oplus B))$$

全加器由两个异或门和三个与非 门组成，连接图如上图

向相邻高位进位数为Ci

输入			输出	
Ai	Bi	Ci-1	Si	Ci
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

由真值表可得 A、B、Ci 输入为 010 时，Si、Cout 输出为 1 0，是在程序中为标号 1、4 的逻辑门的输出



四、 实验心得

通过完成这次项目，不仅得到了技能上的锻炼，而且也对数据科学有了新的认识。计算机科学作为一门实践科学，仅仅掌握了书本上的知识是远远不够的，还应该拓宽自己的眼界将所学应用于现实生活中的诸多领域。正如我们本次实验项目，在我们熟练和应用了计算机技术之后，工科的很多实验都可以得到模拟。这也是新时期以来计算机科学得以凌驾于其他学科的重要原因之一。

对我个人来说，一学期的学习使得我受益匪浅，但是由于没有学习之前数字电路的知识，完成实验稍显吃力，但是通过项目的锻炼，能力也得到了很大提升。通过完成这次项目，不仅得到了技能上的锻炼，而且也对数据科学有了新的认识。