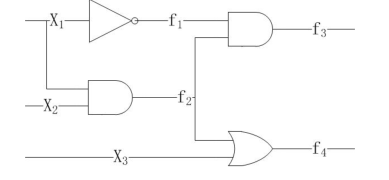
## 数据结构：

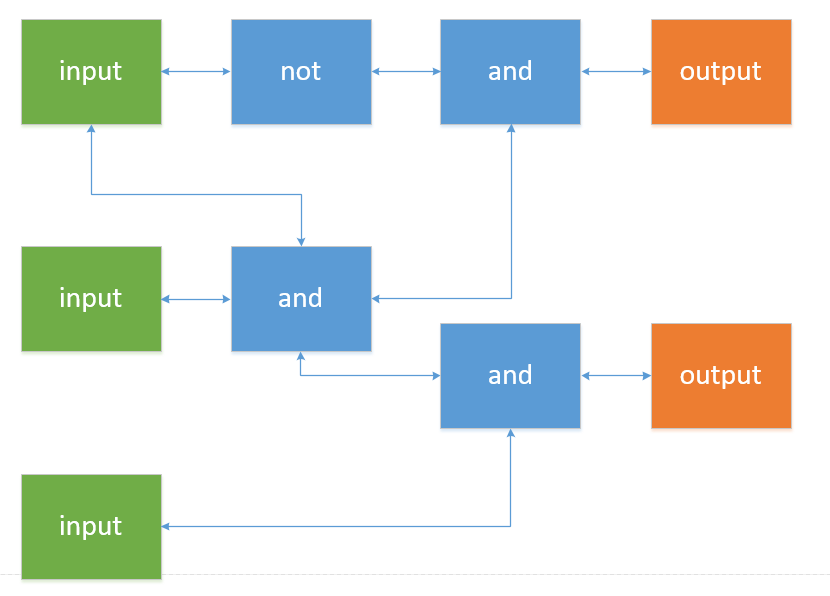
采用了双向的链式存储结构来表示电路（有向）图。考虑过使用邻接矩阵来表示，但是邻接矩阵过于稀疏，出于计算效率的考量，选择了链式结构。

以如下电路图为例：



构造完成后如下图，一共有9个门节点GateNode，每个门节点对象都有自己的类型定义type（输入、输出、非门、与门、或门……），最终计算结果由4个蓝色的门决定。除此之外，每个门都有一个状态量value（1：真，0：假，-1：未初始化）与权重weight（由其后继门的数量决定）。每个门都包含其前驱门pioneer和后继门successor的指针，可以实现门与门的直接访问。

同时，所有的门包含于一个电路Circuit对象，并有序排列在一个数组gate中，可实现对门节点的随机访问。



## 计算输出值：

程序需要计算两个输入向量v1，v2对门状态g1，g2的影响，将g1和g2做异或操作后求得电路的功耗。

求解方式采用从后到前的递归求解。即先计算输出门的value，计算前判断其前驱门是否未初始化，若未初始化，则递归计算前驱门的value，递归收敛于输入门。

### 爬山算法：

设计的过于粗糙就不多说了。只是简单的测试一下电路设计是否正确。

由于一次只反转一个变量，所以在设计爬山法时尝试了保留电路上次输入的状态，只计算与被反转变量有关的门节点（而不是将电路完全重置），但是计算中发现门节点对输出的耦合度非常高，每个变量都会与很多个门节点相关（在电路巨大的情况下，这个相关度更高），因此这个方法的效果并不是很理想。