## 逻辑运算与位运算

#### 一 知识点：

逻辑运算，补码，位运算。

1 逻辑运算

逻辑变量之间的运算成为逻辑运算，又称作布尔运算。

1）逻辑加法，|或运算，A或B只要有一个为1，其逻辑加的结果就为1。

0+0=0， 0∨0=0

0+1=1， 0∨1=1

1+0=1， 1∨0=1

1+1=1， 1∨1=1

2）逻辑乘法，&与运算，都同时取值为1时，其逻辑乘积才等于1。

0×0=0， 0∧0=0， 0·0=0

0×1=0， 0∧1=0， 0·1=0

1×0=0， 1∧0=0， 1·0=0

1×1=1， 1∧1=1， 1·1=1

3）逻辑否运算，非运算

0=1 “非”0等于1

1=0 “非”1等于0

4）逻辑异或运算，两个逻辑变量相异，输出才为1。

0⊕0=0 0同0异或，结果为0

0⊕1=1 0同1异或，结果为1

1⊕0=1 1同0异或，结果为1

1⊕1=0 1同1异或，结果为0

2 补码

计算机中的符号数有三种表示方法，即[原码](https://baike.baidu.com/item/%E5%8E%9F%E7%A0%81/1097586" \t "https://baike.baidu.com/item/%E8%A1%A5%E7%A0%81/_blank)、[反码](https://baike.baidu.com/item/%E5%8F%8D%E7%A0%81/769985" \t "https://baike.baidu.com/item/%E8%A1%A5%E7%A0%81/_blank)和补码。三种表示方法均有符号位和数值位两部分，符号位都是用0表示“正”，用1表示“负”，而数值位，三种表示方法各不相同。

在计算机系统中，数值一律用补码来表示和存储。原因在于，使用补码，可以将符号位和数值域统一处理；同时，加法和减法也可以统一处理。此外，补码与原码相互转换，其运算过程是相同的，不需要额外的硬件电路。

1）正整数的补码是其二进制表示，与[原码](https://baike.baidu.com/item/%E5%8E%9F%E7%A0%81" \t "https://baike.baidu.com/item/%E8%A1%A5%E7%A0%81/_blank)相同。

负整数的补码是在原码的基础上除符号位外其余位取反后+1。

1. 补码转为原码

正数的的补码与原码相同。

负数的补码转原码有两种方式,

第一种，除符号位，补码减1，取反码。

第二种，除符号位，取反码，加1。

#### 二 问题

1 用位运算求一个数的绝对值

对一个数进行位运算时，实际上是在这个数的补码上进行运算的。正数的补码是原码，负数的补码是除符号位之外的所有位取反后加1得到的数字。

方法一：

所以当数字a为负数时，绝对值为~a+1.

具体表达式为，先移位，int i=a>>31;如果a为正数，i等于0；如果a为负数，i等于1.如果i==0,返回a;如果i==1,返回~a+1.

int myAbs(int a){

int i=a>>31;

return i==0?a:(~a+1);

}

方法二：

已知，对于任何数，与0异或都保持不变，与-1（即oxffffffff）异或相当于取反。

先移位，int i=a>>31;再令a与i异或后再减i。减i即正数减0,负数加1，即可得到绝对值。

int myAbs(int a){

int i=a>>31;

return (a^i)-i;

}

# 网络