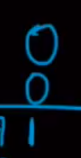
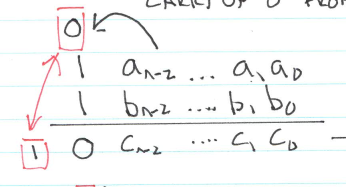
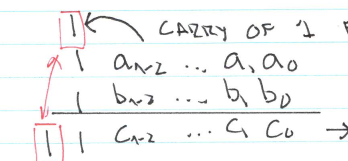
Overflow

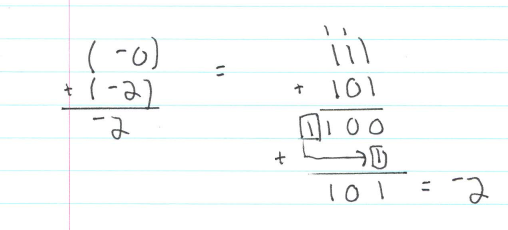
  第一种判断，负负加起来正，正证加起来负

第二种

 overflow

not overflow多出去的那位 和第一位 相同，不溢出，不相同，溢出

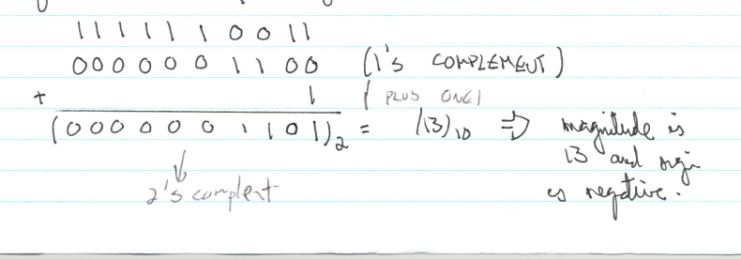
一进制中

要把多出来的那位加回来才能对

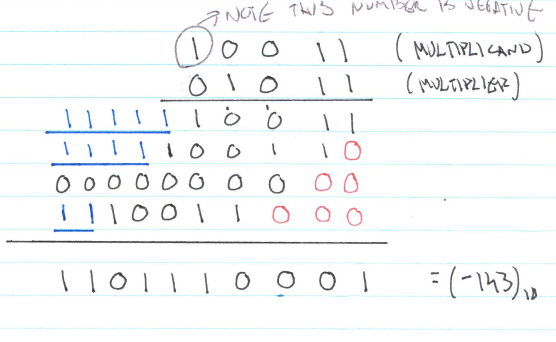
Sign extension 把一个五位的负数转成10位的负数

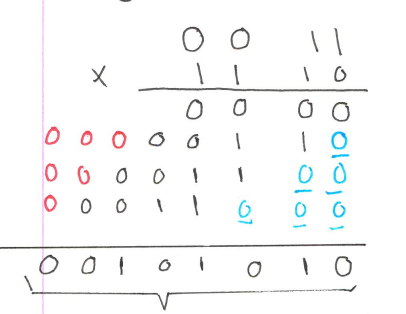
那么就在前面加5个1

比如求10011在10位中二进制补码



用sign extension的情况 ：负数×正数

-13乘11

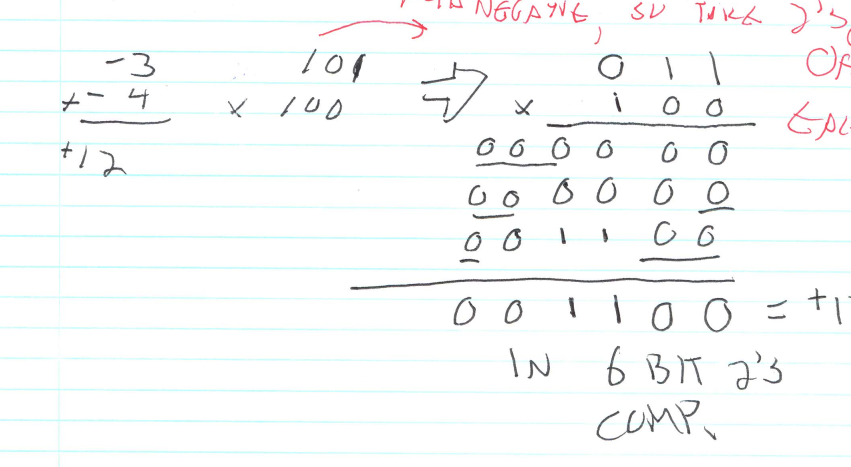
这样就错了，正数不需要sign extension

处理方法，正数相乘，不用extension

负数乘正数，负数求补码extension

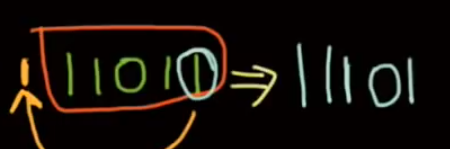
正数乘负数，转成负数乘正数

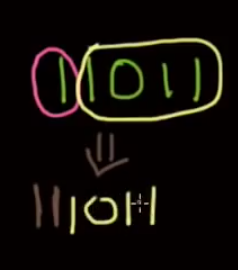
负数乘负数，抵消两个负数当正数乘正数 ，负数用补码表示，再求补码的补码



-3×-4=3×4

Booth算法

RSC前面不变，后面第一位移到前面

RSA 复制第一位，后面不变