

ICS-Lab1说明文档

19302010020 袁逸聪

bitAnd:用~和|完成&的功能

&为只对1和1输出1, 对其他输出0(抽象为对某组相同输入, 给出相同输出)

|为只对0和0输出0, 如使输入与输出都相反, 即可模拟以上抽象

故 $\sim(\sim x|\sim y)$ 即为 $x\&y$

getBytes:对4Bytes长度的数, 拿到从低往高, n下标处的Byte(下标从0起)

右位移将所需结果放到最低位, 因为n为Byte数而非bit, 位移位数为8倍n

再与0xff做&, 得到最低Byte

logicalShift:模拟逻辑右移

默认为算数右移, 对负数将在左边填充1, 现需要改成0

故先制作左边n个0, 右边全是1的过滤器。与直接位移结果&, 将左边部分清空为0

bitCount:返回数字中1的个数

原本为1的位, 表示实际上是 $2^{\text{其位数}}$, 而不是1, 我们需要让他们都表示1

如: 对于11, 前一个表示2而后一个表示1, 我们需要使他们都表示1

对前一位, $(x >> 1) \& 01$ 表示它是否是1

对后一位, $x \& 01$ 表示它是否是1

将二者相加, 就将原本n位处, 表示 2^n 的1转为表示1本身

对32位数, 采用二分查找, 减少运算步骤

先用01010101.....的模式, 右移1位进行两两比较。运算结束后, 每两位表示的是之前这两位的1的个数了

再永00110011.....的模式, 右移2位进行比较。反复翻倍规模, 直到32位的1都只表示1而非 2^n

bang:不用!模拟!x,即0输出1, 非0输出0

先取-x, 由于正数与负数首位必相反, 而0取相反数后首位仍相同

对0, -x|x的首位将是0, 否则会是1(此处用^代替|也一样)

最后通过位移取首位输出

tmin:返回最小的补码整数

即需要返回0x80000000，仅首位为1，将1左移31位即可

fitsBits:求x是否能在n位内被补码表示，能返回1不能返回0

测试样例可能有问题，在n=32的样例中要求答案为0，麻烦助教根据bits.c中代码批改

已知，补码表示的数，在左边任意做相同于最高位的拓展，其代表的数字大小不变

故先左移k再右移k，相当于将32-k位左边的位，改为以最高位拓展。对足够用32-k位表示的数，其值不应改变（否则意味着高位丢失部分信息，即只用低位无法表示）

先取32-n，作为要位移的量，然后比较x和x先左移再右移32-k的结果，结合^和!，使全部相同返回1、有不同返回0

divpwr2:返回 $x/(2^n)$ ，即正数向上取整、负数向下取整

正数情况下，直接右移即可，结果将向下取整。但负数情况下，期望做到的是向上取整

利用 (x/y) 向上取整 $= (x+y-1/y)$ 向下取整，当 $x < 0$ 时，先加上 2^n-1 再右移即可

negate:返回-x

根据补码表示的原理， $\sim x + 1$ 即为答案

isPositive:x>0返回1，否则返回0

$x >> 31$ 将最高位拓展，正数为0负数为1，+1后， ≥ 0 为1而 < 0 为0

但是，0的输入也要被排除，应而并上 $!(x)$ ，必须非零才能导出1

isLessOrEqual:x<=y返回1，否则返回0

首先：右移31位,再&1可以获得首位，取得变量的负号

分3类： $x < 0$ 且 $y \geq 0$ ；x和y同号，避免溢出地获取 $x-y$,看是否为负；x和y是否相等

1. 检测异号且x符号为负
2. 检测x-y相减的首位，需要&二者同号，以排除负减正溢出导致的误判
3. 用!和^检测x、y是否相等

ilog2:返回 $\text{floor}(\log_2(x))$,即x最高1的位数

对一个非负数，右移n位后为0，说明最高1的位数小于等于n，否则大于n

使n为16、8、4、2、1逐步进行二分查找

为0则进入下一轮，不为0则加上本轮查找范围，再进行下轮查找