

## 1 bitXor

根据真值表，异或当  $x \neq y$  时为正， $x = y$  时为假，因此通过  $x \& y$  和  $(\sim x) \& (\sim y)$  取出  $xy$  相同的情况，然后使用或非得到异或。

## 2 tmin

通过 1 左移 31 位构造 tmin。

## 3 isTmax

加 1 取反与自身相同的只有 tmax 和 tmin，而 tmin 加 1 后为 0，因此只要判断加 1 取反与原值相同且加一非零，则为 tmax。

## 4 allOddBits

当且仅当一个数字所有奇数位都为 1 时，其与奇数位全 0，偶数位全 1 的 0xAAAAAAAA 取或之后必定全 1，则位反后为 0，逻辑非后为 1。

## 5 negate

$x$  和  $\sim x$  之和为 0xffffffff 即 -1，所以  $-x$  等于  $\sim x + 1$ 。

## 6 isAsciiDigit

ascii 阿拉伯数字介于 0x30 和 0x39 之中，即高于第六位全 0，第五位和第四位为 1，第三位为 0 或第三位为 1 且第零位为 0。

## 7 conditional

使用两个逻辑非将非零全变成 1，取反后加 1 得到 0xffffffff，同时 0 会变成 0x00000000，则此时可以通过或和与得到  $x$  为 0 的输出，异或非和与得到  $x$  非零的输出，最后用或将两个输出合并。

## 8 isLessOrEqual

考虑  $x$  和  $y$  的符号，当  $x$  负  $y$  正时为真， $x$  正  $y$  负时为假，当同好时考虑  $x - y$ ，此时不会出现溢出，则  $x - y$  非负时为真，否则为假。

## 9 logicalNeg

$x$  与  $-x$  可以保证当且仅当  $x$  非 0 时最高位为 1，右移 31 位后为 -1，加一后为 0，而当  $x$  为 0 时最后为 1。

## 10 howManyBits

求  $x$  至少需要多少位表示，即求  $x$  的二进制表示前有多少连续相同的数字。通过与最高位进行异或非可将负数转为位非的结果，而正数保持不变。此时考虑最高位 1 的位置。通过二分法节省操作数，右移 16 位后两次位非将有 1 的情况全部设为 1，若为 1，则将  $b16$  设为 16， $x$  右移 16 位，即忽略  $x$  后面 16 为，考虑前 16 位中 1 的位置，以此类推，最后将所有求和后加 1，即为最少需要的位数。

## 11 floatScale2

当  $uf$  为 0 或 NaN 时，通过 if 分支直接返回  $uf$ ，否则则将  $exp$  部分加 1 即可。

## 12 floatFloat2Int

将 float 的  $sign$ ,  $exp$ ,  $frac$  部分分别取出存到  $s$ ,  $exp$ ,  $y$  中， $E$  为  $frac$  需要的偏移量。当  $exp$  为全 1，即  $uf$  为  $inf$  或 NaN，返回  $0x80000000u$ ，其他情况当  $E$  小于零 0， $frac$  右移， $E$  小于过多则为 0， $E$  大于等于 0， $frac$  左移，超过上限则为  $0x80000000u$ ，当  $s$  为 1 时，返回负值，否则返回正值。

## 13 floatPower2

求 2 的  $x$  次方，则将  $x$  加上偏移量后放到返回值的  $exp$  位置，需要判断是否溢出。

## 14 运行结果

Correctness Results			Perf Results		
Points	Rating	Errors	Points	Ops	Puzzle
1	1	0	2	7	bitXor
1	1	0	2	1	tmin
1	1	0	2	8	isTmax
2	2	0	2	9	allOddBits
2	2	0	2	2	negate
3	3	0	2	14	isAsciiDigit
3	3	0	2	11	conditional
3	3	0	2	21	isLessOrEqual
4	4	0	2	5	logicalNeg
4	4	0	2	36	howManyBits
4	4	0	2	24	floatScale2
4	4	0	2	30	floatFloat2Int
4	4	0	2	6	floatPower2
Score = 62/62 [36/36 Corr + 26/26 Perf] (174 total operators)					

图 1: 运行结果