

Homework 01_实验报告_10214602404

一. 解题思路

1. bitXor

我先找出来都是 0 和都是 1 的位，找都是 0 我用了与 $\&$ ($x\&y$)，找都是 1 我用了两者取反 \sim 的与 $\&$ ($\sim x\&\sim y$)。然后对前面取反，得到 0 和 1 相反的位以及都是 1 的位，对后面取反，得到 0 和 1 相反的位以及都是 0 的位。最后把两者与就可以得出异或结果。

2. tmin

补码最小值是 $2^{(n-1)}$ ，把 1 左移 31 位之后，最高符号位是 1，剩余补 0，得到 100.....00。

3. isTmax

当补码是最大值的时候，这个数字左移一位之后应该是 0xffffffff，那么它取反之后就应该是 0，也就是 $2x+1$ 取反不为 0，同时要避免这个数本身就是 0xffffffff 这个情况，即 $x+1$ 不为 0。

4. allOddBits

利用 1 先左移一位这时就构造了 10，此时再继续以 2 的倍数进行左移再并起来，就分别得到 1010、101010.....，最后看 x 和得到的最后数是否一致，先与再异或，为 0 则返回 1。

5. negate

根据阿贝尔群，只需要对一个数取反加 1 就可以得到这个数的相反数。

6. isAsciiDigit

先对 0x30 取反加 1，得到 -0x30，判断 $x-0x30$ 的符号位是否为 0，右移 31 位再取反，符号位是 0 返回 1，同理，判断 $0x39-x$ 的符号位是否为 0，最后同时满足用与运算。

7. Conditional

首先对 x 进行两次取非再变相反数，这样就可以把非 0x 转化称 0xffffffff，然后就可以利用与运算判断 x 与 y ，以及 $\sim x$ 和 z 是否相等，返回两者的并，这时就可以实现非零返回 y ，为零返回 z 。

8. isLessOrEqual

判断 x 是否小于等于 y ，要分两种情况：1.符号相反，谁是 0 谁就大；2.符号相同判断 $y-x$ 是否大于零。两种情况取并。

9. logicalNeg

取非时，为 0 返回 0，其余返回 1。可以先对所求数取相反数，若是 0 的话，0 的相反数还是 0。相反数与它本身取并集再右移 31 位可以得到它的符号位，为 0 则返回 1。

10. howManyBits

计算某个数字有多少个字节，一共有六种情况。先根据符号进行化简，为正不变，为负取反。然后不断利用 $!!(x >> 2^n) << n$ 来判断 2^n 为是否是 1，最后返回所有情况相加的结果。

11. floatScale2

因为 $uf = (-1)^s * M * 2^E$ ，先利用 0x7f800000 进行与计算再右移 23 位，得到 e ；利用与 10000000 判断 $sign$ 。此时会有四种情况：当 e 为 255 时，为 NaN，返回 uf ；如果 e 为 0，未规格化返回左移一位与符号位取并集；接着对 e 进行自加 1，判断是否到达 255，到达则返回 0x7f800000 与符号的并；否则返回这个数和 0x7f800000 交与 e 左移 23 位的结果。

12. floatFloat2Int

先提取出浮点数的 e 和 f ， $f+1$ ，因为是 int_32 ，所以 $f+1$ 。如果 $e-127 > 31$ ，溢出，返回 $infinity$ ；如果 $e-127 < 0$ 或者 $ef == 0$ ，返回 0；根据 e 对 f 进行移位，再判断 f 的符号与原来相不相同，相同返回 f ；正溢出，返回 $infinity$ ；负溢出，返回相反数。

二. Git log 截图

```
jovyan@jupyter-10214602404:~/10214602404/datalab-handout$ git log
commit 69a4943caa195a8593007bdcc293baa703ed41e8 (HEAD -> homework01, origin/homework01)
Author: LiFang <10214602404@stu.ecnu.edu.cn>
Date:   Wed Oct 11 10:20:09 2023 +0000
```

revise NO.3 & NO.4

```
commit 283629aa068a656eebafc446106f9c3c8985d400
Author: LiFang <10214602404@stu.ecnu.edu.cn>
Date:   Wed Oct 11 09:46:34 2023 +0000
```

resolve NO.11 & NO.12

```
commit f518abe6f9189708103a4c0371797589bc1b0ac9
Author: LiFang <10214602404@stu.ecnu.edu.cn>
Date:   Wed Oct 11 06:23:03 2023 +0000
```

resolve NO.10 & NO.11

```
commit cf6a641abdb9e54628fc9b6157aa7fcfe87b2ce5
Author: LiFang <10214602404@stu.ecnu.edu.cn>
Date:   Tue Oct 10 11:49:10 2023 +0000
```

resolve NO.7 & NO.8 & NO.9

```
commit 5391fea7bc9110dc59db18889c1224ac46d1lcea
Author: LiFang <10214602404@stu.ecnu.edu.cn>
Date:   Mon Oct 9 14:30:00 2023 +0000
```

resolve NO.5 & NO.6

```
commit 338fa7ecc251b62eb31ce7f15f78c4e4173b35d0
Author: LiFang <10214602404@stu.ecnu.edu.cn>
Date:   Mon Oct 9 13:27:14 2023 +0000
```

revise NO.2 & resolve NO.4

```
commit dlab985bf8537ad810dcaa630e2a74d13d46cd9
Author: LiFang <10214602404@stu.ecnu.edu.cn>
Date:   Sun Oct 8 13:49:30 2023 +0000
```

resolve NO.2 & NO.3

```
commit fac9e5afcc360ba572fbab7cc78f48c3a61e6764
Author: LiFang <10214602404@stu.ecnu.edu.cn>
Date:   Sun Oct 8 12:55:44 2023 +0000
```

resolve NO.1

三. 测试运行结果

jovyan@jupyter-10214602404:~/10214602404/datalab-handout\$./driver.pl

1. Running './dlc -z' to identify coding rules violations.

2. Compiling and running './btest -g' to determine correctness score.

gcc -O -Wall -m32 -lm -o btest bits.c btest.c decl.c tests.c

btest.c: In function 'test_function':

btest.c:332:23: warning: 'arg_test_range[1]' may be used uninitialized in this function [-Wmaybe-uninitialized]
if (arg_test_range[1] < 1)

3. Running './dlc -Z' to identify operator count violations.

4. Compiling and running './btest -g -r 2' to determine performance score.

gcc -O -Wall -m32 -lm -o btest bits.c btest.c decl.c tests.c

btest.c: In function 'test_function':

btest.c:332:23: warning: 'arg_test_range[1]' may be used uninitialized in this function [-Wmaybe-uninitialized]
if (arg_test_range[1] < 1)

5. Running './dlc -e' to get operator count of each function.

Correctness Results			Perf Results		
Points	Rating	Errors	Points	Ops	Puzzle
1	1	0	2	7	bitXor
1	1	0	2	1	twIn
1	1	0	2	8	isTrax
2	2	0	2	12	allOddBits
2	2	0	2	2	negate
3	3	0	2	11	isAsciiDigit
3	3	0	2	8	conditional
3	3	0	2	18	isLessOrEqual
4	4	0	2	6	logicalNeg
4	4	0	2	36	howManyBits
4	4	0	2	14	floatScale2
4	4	0	2	21	floatFloat2Int

Score = 56/56 [32/32 Corr + 24/24 Perf] (144 total operators)

jovyan@jupyter-10214602404:~/10214602404/datalab-handout\$