计算机体系架构　Lab02

范云潜 18373486

微电子学院 184111 班

日期：2020 年 10 月 1 日

Problem ex1

对于一个 32-bit 的整数 x，加法发生溢出的边界是 ~x，此时恰好得到 ~0x0，若是加数大于此边界（无符号类型）即可判断发生了溢出。

Listing 1: ex1 solution

```
# if the op1 and op2 hava different sign
# there will be no carry out.
# {op1_sign, op2_sign, ans_sign} = {1, 1, 0} or {0, 0, 1}
# in 2 instructions
# the largest op2 for op1 is ~op1 (unsigned)
# so we need to compare op2 and ~op1
# the 'not x' could be 'x xor 0xffffffff'

# test 1 passed

# li $t3, 0x80000000
# li $t4, 0x7fffffff

# test 2 passed

# li $t3, 0x00000000
# li $t4, 0xffffffff

# test 3 passed

# li $t3, 0x00000001
# li $t4, 0xffffffff

xori $t3, $t3, 0xffffffff
sltu $t2, $t3, $t4
```

Problem ex2

对于一个浮点数，当其越来越大，每两个数字之间的间距也会越来越

大，那么我们首先挑选一个很大的浮点数[1] {0b 0 | 111 1111 0 | 000 0000 0000 0000 0000 0000 = 170141183460469231731687303715884105728.000000 = 0x7f000000} 。通过 C 和 MARS 进行实验，发现的确如此。

Problem ex3

此处需要找到一个**最小的**浮点数，因此需要仔细考虑浮点数的间隔下界大于 1 的临界点，此时作为尾数域的最小变化会引起整个浮点数增加 2 ，那么指数是 24 = 0x18 ，那么指数域为 24+127 = 0x97 ，那么位表示为 {0b 0 | 100 1011 1 | 000 0000 0000 0000 0000 = 16777216.000000 = 0x4b800000} 。若是尝试 {0x4b800000 - 1= 0x4b7fffff = 16777215.000000} 则会失败，验证完毕。

Problem ex4

事实上浮点数的不可交换源自于浮点数的特殊值也就是 ∞ 和 NaN 。如果前两个值相加会溢出，而第三个值的提前出现会防止这种情况的出现。这样的集合可以表示为

$$\{a, b, c | a + b \text{ overflow } \&\& |a + c| < |a| \text{ thus } (a + c) + b \text{ don't overflow}\}$$

进行举例，最大的规格化值 {0b 0 1111 1110 1111 1111 1111 1111 1111 111 } ，它的一半 {0b 0 1111 1101 1111 1111 1111 1111 1111 111} ，它的相反数 {0b 1 1111 1111 1110 1111 1111 1111 1111 111 } 分别作为 a, b, c 。

Listing 2: ex4 solution

```
#include <stdio.h>
#include <inttypes.h>

int main()
{

    unsigned a = 0x7f7fffffu;
    float* ap = (float*)(&a);
    unsigned b = 0x7effffffu;
    float* bp = (float*)(&b);
    unsigned c = 0xfeffffffu;
    float* cp = (float*)(&c);
```

[1]通过 C 进行转换成浮点数

```
    printf("%f\n%f\n%f\n%f\n%f\n", (*ap), *bp, *cp, (*ap + *bp + *cp), (*ap + *
        cp + *bp));
}
```

输出为

```
340282346638528859811704183484516925440.000000
170141173319264429905852091742258462720.000000
-170141173319264429905852091742258462720.000000
inf
340282346638528859811704183484516925440.000000
```