

计算机体系结构 Lab02

make

1. 本程序的编译使用哪个编译器?
gcc
2. 采用哪个命令, 可以将所有程序全部编译?
make all
3. 采用哪个命令, 可以将所有上次编译的结果全部删除?
make clean
4. 文件中第几行生成 btest 的目标文件?
第11行。
5. 文件中第几行生成 fshow 的目标文件?
第14行。
6. 如果在 Makefile 文件中用要引用变量“FOO”, 怎么表示?
\$(FOO)

位级运算、数的编码

```
mcx@mcx-virtual-machine:~/code/lab02/code$ ./btest
Score  Rating  Errors  Function
  2      2      0    allOddBits
  4      4      0    isLessOrEqual
  4      4      0    logicalNeg
  5      5      0    floatScale2
  5      5      0    floatFloat2Int
Total points: 20/20
```

- allOddBits
先构造出一个仅奇数位为1的数0xAAAAAAAA,然后与x取位与, 再判断是否和0xAAAAAAAA相等。

```
1  int allOddBits(int x) {
2      int a = 0xAA;
3      a |= a<<8;
4      a |= a<<16;
5      return !((a&x)^a);
6  }
```

- isLessOrEqual

先比较符号位，符号相同则相减。当符号位相同时，两者相减一定不会溢出。

```
1  int isLessOrEqual(int x, int y) {
2      int ysign = (y >> 31) & 1;
3      int xsign = (x >> 31) & 1;
4      return (!(ysign & (!xsign))) //y<0,x>=0,return 0
5          &(((!ysign) & xsign) //y>=0,x<0,return 1
6              | ((x+~y)>>31&1));
7  }
```

- logicalNeg

通过4次位或操作，将所有位以或运算形式合并到最低位上。

```
1  int logicalNeg(int x) {
2      x |= x>>16;
3      x |= x>>8;
4      x |= x>>4;
5      x |= x>>2;
6      x |= x>>1;
7      return (x^1)&1;
8  }
```

- floatScale2

按照IEEE754标准分类处理即可。

```
1  unsigned floatScale2(unsigned uf) {
2      unsigned int sign = (uf>>31)<<31;
3      unsigned int exp = uf>>23 & 0xFF;
4      if(!exp){
5          //非规约或正负0
6          return uf<<1 | sign;
7      }
8      if(exp==0xFE){
9          //乘二后溢出为无穷大
10         return 0xFF<<23 | sign;
11     }
12     if(exp!=0xFF){
13         //其他数，指数位加一
14         return uf+(0x1<<23);
15     }
16     //Nan直接返回
17     return uf;
18 }
19
```

- floatFloat2Int

按照IEEE754标准分类处理。需要注意位移运算右操作数的范围。

```
1  int floatFloat2Int(unsigned uf) {
2      unsigned int exp = uf>>23&0xFF;
3      int res = ((uf<<9)>>9) + (0x1<<24);
4      if(exp==0x0 || exp<103) return 0x0; //绝对值小于1
5      if(exp>157) return 0x80<<24;      //溢出
6      if(exp>151) res = res<<(exp-151);  //实际指数大于24，左移
7      else res = res>>(151-exp);        //实际指数小于24，右移
8      if(uf>>31) res = ~res + 1;        //对于负数，取其补码
9      return res;
10 }
```