一、make

1.本程序的编译使用gcc编译器；

2.直接使用make或 make all即可完成所有程序的编译；

3.使用make clean可以将上次编译的所有结果删除

4.第10、11行生成btest的目标文件（即以btest为目标文件，作为开头的那一行）；

5.第13、14行生成fshow的目标文件（即以fshow为目标文件，作为开头的那一行）；

6.使用$(FOO)即为引用变量FOO。

二、位级运算、数的编码

1.

int allOddBits(int x) {

//your codes Here

if((0x55555555 | x) + 1)

return 0;

else return 1;

}

2.

int isLessOrEqual(int x, int y) {

//your codes here

if((x>>31)+1 && y>>31)

return 0;

if(x>>31 && (y>>31)+1)

return 1;

if((y+~x+1)>>31)

return 0;

else return 1;

}

3.

int logicalNeg(int x) {

//your codes here

if(x) return 0;

else return 1;

}

4.

unsigned floatScale2(unsigned uf) {

//your codes here

if(!(uf<<1))

return uf;

if((uf>0x80000000 && uf <=0x807FFFFF)||(uf>0x00000000 && uf <=0x007FFFFF))

return uf+((uf<<1)>>1);

if((uf>=0x7F800000 && uf <=0x7FFFFFFF)||(uf>=0xFF800000 && uf <=0xFFFFFFFF))

return uf;

return uf+0x00800000;

}

5.

int floatFloat2Int(unsigned uf) {

//your codes here

if((uf>=0x80000000 && uf<=0xBF7FFFFF)||uf<=0x3F7FFFFF)

return 0;

if((uf>0x4F7FFFFF && uf<=0x7FFFFFFF)||uf>0xCF7FFFFF)

return 0x80000000u;

if(((uf<<1)>>24)>=150)

return (-2\*(uf>>31)+1)\*(((uf & 0x007FFFFF)+ 0x00800000)<<(((uf<<1)>>24)-150));

else

return (-2\*(uf>>31)+1)\*(((uf & 0x007FFFFF)+ 0x00800000)>>(150-((uf<<1)>>24)));

}

运行结果：

