

Homework 7-Sol

Floating Point

1. Fill the table:

Fractional value	Binary representation	Decimal representation
5/8	0.101	0.625
107/32	11.01011	3.34375
29/32	0.11101	0.90625
3/16	0.0011	0.1875
85/64	1.010101	1.328125

2. Consider a 16-bit floating-point representation based on the IEEE floating-point format, with 1 sign bit, 6 exp bits, 9 frac bits, called **Float16**.

(1) Fill in the following table, please represent M in the form **x** or **x/y** where x is an integer and y is an integral power of 2, and represent Value in the form **a** or **a * 2^b** where a and b are integers.

Description	Hex	M	E	Value
-0	0x8000	0	-30	--
Largest negative Normalized value	0x8200	1	-30	$-1 * 2^{(-30)}$
$+\infty$	0x7E00	--	--	--
Largest Denormalized value	0x01FF	511/512	-30	$511 * 2^{(-39)}$
Number with hex representation 0x4BF7	0x4BF7	1015/512	6	$1015 * 2^{(-3)}$
Sample	0xC4A0	21/16	3	$-21 * 2^{(-1)}$

(2) Assume we use **IEEE round-to-even** mode to do the approximation, let a = 0x33CD, b = 0x2F89, compute a + b and a * b. The answer should also be represented in **Float16**.

a: 0|011 001|1 1100 1101 S1=0, E1=-6, M1=1.111001101

b: 0|010 111|1 1000 1001 S2=0, E2=-8, M2=1.110001001

a+b:

1.111001101

+0.01110001001

10.0101011101

$M=(10.0101011101)_2$; $M'=(1.001010111|101)_2 = (1.001011000)_2$

$E=E1+1=-5$

$Exp = E + bias = 26$

a+b = (0011010001011000)₂ = 0x3458

a*b:

$S=S1 \wedge S2=0$; $M=M1*M2=(11.010110111110110101)_2$; $E=E1+E2=-14$

$E'=E+1=-13$; $M'=(1.101011011|1110110101)_2=(1.101011100)_2$

$Exp=E'+bias=18$

a*b = (0010010101011100)₂ = 0x255C

3. Translate the following assembly into C codes. You can name the arguments and variables freely as you like.

```
pushq  %rbp
movq   %rsp, %rbp
movl   %edi, -4(%rbp)
vmovss %xmm0, -8(%rbp)
movq   %rsi, -16(%rbp)
vmovsd %xmm1, -24(%rbp)
vxorpd %xmm0, %xmm0, %xmm0
vcvtss2sd -4(%rbp), %xmm0, %xmm0
movq   -16(%rbp), %rax
vmovsd (%rax), %xmm1
vaddsd %xmm1, %xmm0, %xmm1
vcvtss2sd -8(%rbp), %xmm0, %xmm0
vmovsd -24(%rbp), %xmm2
vsubsd %xmm0, %xmm2, %xmm0
vmulsd %xmm0, %xmm1, %xmm0
popq   %rbp
ret
```

//Not the only solution

```
double funtc (int i, float f, double *dp, double d){
    return (i+*dp)*(d-f);
}
```