Recitation 5 Solution - Computer Organization, Spring 2013

Problem 1. An IEEE floating point representation uses 4 *exp* bits and 5 *frac* bits.

1) How many bits are needed to store these numbers?

   Sol: 1 + 4 + 5 = 10

2) What is the bias?

   Sol: 2^{4-1} -1 = 7

3) How many denormalized values are there?

   Sol: there are five fractional bits, so $2^5 = 32$, total: 64 (+ and -).

4) What is the binary representation of the smallest denormalized value that is greater than 0?

   Sol: 0 0000 00001

5) What is the smallest denormalized value that is greater than 0?

   Sol: $(.00001)_2 * 2^{1-7} = 2^{-5} * 2^{-6} = 2^{-11} = 1/2048 = 0.00048828125$

6) What is the binary representation of the smallest normalized value that is greater than 0?

   Sol: 0 0001 00000

7) What is the smallest normalized value that is greater than 0? **Sol:** $1.0\times2^{1-7} = 2^{-6} = 0.015625$

8) What is the binary representation of the largest normalized value? **Sol :** 0 1110 11111

9) What is the largest normalized value?

   Sol: *exp* =14, real exponent is 14- 7 = 7, so the value is $(1.11111)_2 \times 2^7$=11111100=252

10) How would the number 69 be represented? (Give the answer in binary and hex.)

    Sol: $69 = (1000101)_2 = (1.000101)_2 \times 2^6$ which cannot be represented with 5 *frac* bits.

11) How would the number 68 be represented? (Give the answer in binary and hex.)

    Sol: $68 = (1000100)_2 = (1.000100)_2 \times 2^6$ so E = 6 and *exp* = 6 + 7 = 13 and *frac* is 00010, so the answer is 0 1101 00010 = 0110100010 = 01 1010 0010 = 0x1a2

12) How would the number -6.25 be represented? (Give the answer in binary and hex.)

    Sol: $6.25 = (110.01)_2 = (1.1001)_2 \times 2^2$, so E = 2 and *exp* = 2 + 7 = 9 and *frac* is 10010, so the answers is 1 1001 10010 = 1100110010 = 11 0011 0010 = 0x332

13) The bits corresponding to 0x10 are stored in a variable that represents one of the numbers. What is its value?

    Sol: 0x10 = (10000)2 = 0 0000 10000 which is denormalized. *frac* = 10000 so the value is $(.10000)_2 \times 2^{-6} = (10000)_2 \times 2^{-11} = 16/2048 = 0.0078125$

14) The bits corresponding to 0x34a are stored in a variable that represents one of the numbers. What is its value?

    Sol: $0x34a = (001101001010)_2$ = 1 1010 01010, so *exp* = $(1010)_2$ = 10 and *frac* = 01010, so

$E = 10 - 7 = 3$ and the number is $-(1.01010)_2 \times 2^3 = -(1010.10)_2 = -10.5$

Problem 2. Assume variables x, f, and d are of type int, float, and double, respectively. Their values are arbitrary, except that neither f nor d equals $+\infty$, $-\infty$, or NaN . For each of the following C expressions, either argue that it will always be true (i.e., evaluate to 1) or give a value for the variables such that it is not true (i.e., evaluates to 0).

1) x == (int)(double) x
   Sol: Yes, since double has greater precision and range than int.

2) x == (int)(float) x
   Sol: No. For example, when x is Max

3) d == (double)(float) d
   Sol: No. For example, when d is 1e40, we will get $+\infty$ on the right

4) f == (float)(double) f
   Sol: Yes, since double has greater precision and range than float

5) f == -(-f)
   Sol: Yes, since a floating-point number is negated by simply inverting its sign bit

6) 1.0/2 == 1/2.0
   Sol: Yes, the numerators and denominators will both be converted to floating-point representations before the division is performed.

7) $d \times$ d >= 0.0
   Sol: Yes, although it may overflow to $+\infty$

8) (f+d) - f == d
   Sol: No, for example when f is 1.0e20 and d is 1.0, the expression f+d will be rounded to 1.0e20, and so the expression on the left-hand side will evaluate to 0.0, while the right-hand side will be 1.0.