Name: KEY **Id#**

ICS 233, Term 142

Computer Architecture & Assembly Language

Quiz# 4

Date: Thursday, April 2, 2015

Q1. Consider a simplified 8-bit floating point representation following the general guidelines of the IEEE format in representing normalized, denormalized, Nan, infinity and 0. Suppose that the number of bits used for the exponent is 3 and for the fraction is 4 bits.

- **(i)** Determine the smallest and largest positive values of normalized numbers.
- **(ii)** Determine the smallest and largest positive values of denormalized numbers.
- **(iii)** Determine the representation used for +0 and +∞.
- **(iv)** What is the largest and smallest error in this representation?

(i)

Number	S	Exp	Fraction	E	Value
Smallest normalize number	0	001	0000	1-3=-2	1*1/4=1/4
Largest normalize number	0	110	1111	6-3=3	31/2=15.5

(ii)

Number	S	Exp	Fraction	E	Value
Smallest denormalize number	0	000	0001	-2	1/16*1/4=1/64
Largest denormalize number	0	000	1111	-2	15/64≈1/4

(iii)

Number	S	Exp	Fraction
+0	0	000	0000
+∞	0	111	0000

(iii)

If we consider the largest values with E=110=3, we can see that these values range from 8 to 15.5. The values in this range are: 8, 8.5, 9, 9.5 …., 15.5. Thus, the largest error is 0.5/2=0.25. If we consider the smallest normalized values with E=001=-2, we can see that these values range from 1/4 to 31/16. The values in this range are: 1/4=16/64, 17/64, 18/64, 19/64 …., 31/16. Thus, the smallest error is 1/64*2=1/128.
Note that the same magnitude of error occurs in the representation of denormalized numbers as they are in the range of 1/64, 2/64,..,15/64.

Q2.

(i) What is the decimal value of the following single-precision floating-point number:

`0100 0011 0110 1001 1000 0100 0000 0000.`

$= + (1.1101001100001000...0)_2 * 2^{(134-127)} = + (1.1101001100001000\ldots0)_2 * 2^7$
$= + (11101001.100001000\ldots.0)_2$
$= + 233.515625$

(ii) Show the single-precision floating-point binary representation for: **555.9375**.

$555.9375=(1000101011.1111)_2 = (1.0001010111111)_2 * 2^9$
Exp. = 9 +127=136
Single precision binary representation:

`0100 0100 0000 1010 1111 1100 0000 0000`

(iii) Perform the following floating-point operation rounding the result to the **nearest even**. Perform the operation using **guard**, **round** and **sticky** bits.

```
       1.000 0000 1000 0000 0000 0000 000 x 2^37
-      1.000 0000 0000 0000 0100 0000 000 x 2^29
-----------------------------------------------------------
       1.000 0000 1000 0000 0000 0000 000 x 2^37
-      0.000 0000 1000 0000 0000 0000 010 x 2^37  (align)
-----------------------------------------------------------
      01.000 0000 1000 0000 0000 0000 000 x 2^37
+     11.111 1111 0111 1111 1111 1111 110 x 2^37 (2's complement)
-----------------------------------------------------------
      00.111 1111 1111 1111 1111 1111 110 x 2^37
= +    0.111 1111 1111 1111 1111 1111 110 x 2^37
= +    1.111 1111 1111 1111 1111 1111 100 x 2^36 (normalize)
= +  10.000 0000 0000 0000 0000 0000      x 2^36 (round)
= +   1.000 0000 0000 0000 0000 0000      x 2^37 (renormalize)
```