COE 308 – Computer Architecture

Assignment 3 Solution

Floating-Point Representation and Arithmetic

1. **(4 pts)** What is the decimal value of the following single-precision floating-point numbers?

a) `1010 1101 0001 0100 0000 0000 0000 0000` (binary)
b) `0100 0110 1100 1000 0000 0000 0000 0000` (binary)

Solution:

a) Sign is negative

Exponent value = $01011010_2 - 127 = -37$

Significand = $1.001\ 0100\ 0000\ 0000\ 0000\ 0000_2$

Decimal value = $-1.00101_2 \times 2^{-37} = -1.15625 \times 2^{-37} = -8.412826 \times 10^{-12}$

b) Sign is positive

Exponent value = $10001101_2 - 127 = 14$

Significand = $1.100\ 1000\ 0000\ 0000\ 0000\ 0000_2$

Decimal value = $1.1001_2 \times 2^{14} = 1.5625 \times 2^{14} = 25600$

2. **(3 pts)** Show the IEEE 754 binary representation for: `-75.4` in …

a) Single Precision
b) Double precision

Solution:

$75 = 1001011_2$

$0.4 = 0.\overline{0110}_2 = 0.01100110_2 \ ...$

$75.4 = 1001011.\overline{0110}_2 = 1.0010110\overline{110}_2 \times 2^6$

a) Single-Precison: Biased exponent = 6 + 127 = 133

$1\ 10000101\ 00101101100110011001101_2$ **(rounded to nearest)**

b) Double-Precision:Biased exponent = 6 + 1023 = 1029

$1\ 10000000101$

$0010110110011001100110011001100110011001100110011010_2$ **(rounded)**

3. **(6 pts)** x = `1100 0110 1101 1000 0000 0000 0000 0000` (binary)
and y = `0011 1110 1110 0000 0000 0000 0000 0000` (binary)
are single-precision floating-point numbers. Perform the following operations showing all work:

a) $x + y$
b) $x * y$

Solution:

Value of Exponent(x) = $10001101_2 - 127 = 14$

x = $-1.101\ 1000\ 0000\ 0000\ 0000\ 0000_2 \times 2^{14}$

Value of Exponent(y) = $01111101_2 - 127 = -2$

y = $1.110\ 0000\ 0000\ 0000\ 0000\ 0000_2 \times 2^{-2}$

a) x + y

$$
\begin{array}{l}
-\ 1.101\ 1000\ 0000\ 0000\ 0000\ 0000_2 \times 2^{14} \\
+\ 1.110\ 0000\ 0000\ 0000\ 0000\ 0000_2 \times 2^{-2} \\
\hline
-\ 1.101\ 1000\ 0000\ 0000\ 0000\ 0000_2 \times 2^{14} \\
+\ 0.000\ 0000\ 0000\ 0000\ 1110\ 0000_2 \times 2^{14}\ \text{(shift right 16)} \\
\hline
1\ 0.010\ 1000\ 0000\ 0000\ 0000\ 0000_2 \times 2^{14}\ \text{(2's complement)} \\
0\ 0.000\ 0000\ 0000\ 0000\ 1110\ 0000_2 \times 2^{14} \\
\hline
1\ 0.010\ 1000\ 0000\ 0000\ 1110\ 0000_2 \times 2^{14}\ \text{(add)} \\
\hline
-\ 1.101\ 0111\ 1111\ 1111\ 0010\ 0000_2 \times 2^{14}\ \text{(2's complement)}
\end{array}
$$

Result is negative and is normalized

All shifted out bits were zeros, so result is also exact

x + y = $1\ 10001101\ 101\ 0111\ 1111\ 1111\ 0010\ 0000_2$

b) x * y

Biased exponent(x*y) = $10001101_2 + 01111101_2 - 127$

Biased exponent(x*y) = $139 = 10001011_2$

Sign(x*y) = 1 (negative)

$$
\begin{array}{r}
1.101\ 1000\ 0000\ 0000\ 0000\ 0000_2 \\
\times\ 1.110\ 0000\ 0000\ 0000\ 0000\ 0000_2 \\
\text{1 1111} \qquad\qquad \\
\hline
1101\ 1000\ 0000\ 0000\ 0000\ 0000_2 \\
11011\ 0000\ 0000\ 0000\ 0000\ 000_2 \\
1.10110\ 0000\ 0000\ 0000\ 0000\ 00_2 \\
\hline
10.11110\ 1000\ 0000\ 0000\ 0000\ 0000_2
\end{array}
$$

Normalize by shifting right 1 bit and increment exponent

Significand = $1.011\ 1101\ 0000\ 0000\ 0000\ 0000_2$

Biased exponent = $139+1 = 140 = 10001100_2$

Significand is already rounded

x*y = $1\ 10001100\ 011\ 1101\ 0000\ 0000\ 0000\ 0000_2$

4. **(4 pts)** x = `0101 1111 1011 1110 0100 0000 0000 0000` (in binary)
and y = `0011 1111 1111 1000 0000 0000 0000 0000` (in binary)
and z = `1101 1111 1011 1110 0100 0000 0000 0000` (in binary)
represent single precision IEEE 754 floating-point numbers. Perform the following operations showing all work:

a) $x + y$
b) Result of (**a**) + z
c) Why is the result of (**b**) counterintuitive?

Solution:

a) $x = 1.011\ 1110\ 0100\ 0000\ 0000\ 0000_2 \times 2^{64}$

$y = 1.111\ 1000\ 0000\ 0000\ 0000\ 0000_2 \times 2^{0}$

Difference in exponent = 64

Shift significand of y right by 64 bits and add to x

The significand bits of y are truncated after rounding

x + y = x because y is too small with respect to x

Therefore, $x + y = 1.011\ 1110\ 0100\ 0000\ 0000\ 0000_2 \times 2^{64}$

b) **Result of (a) is** $x = 0\ 10111111\ 01111100100000000000000_2$

$z = 1\ 10111111\ 01111100100000000000000_2 = -x$

Therefore, Result of (a) + z = x − x = 0

$0\ 00000000\ 00000000000000000000000_2$

c) **We are computing (x+y) + z where z = -x**

Intuitively (x+y)+ -x = y which is not 0

However, in this example (x+y)+ -x = 0

This is because we have limited number of fraction bits

5. **(3 pts)** IA-32 offers an 80-bit extended precision option with a 1 bit sign, 16-bit exponent, and 63-bit fraction (64-bit significand including the implied 1 before the binary point). Assume that extended precision is similar to single and double precision.

a) What is the bias in the exponent?

b) What is the range (in absolute value) of normalized numbers that can be represented by the extended precision option?

Solution:

a) **With a 16-bit exponent, bias =** $2^{15} - 1 = 32767$

b) **largest normalized** $\approx 2 \times 2^{32767} = 2^{32768} = 1.415.. \times 10^{9864}$

smallest normalized: $1.0 \times 2^{-32766} = 2.8259.. \times 10^{-9864}$