Pathri Vidya Praveen
CS24BTECH11047
Homework 3

1. Convert the following decimal numbers into IEEE-754 floating-point format (write the final answer in Hex). Show all steps [6 marks]

a. -13.25 (single precision)

Sol:

Sign is negative , so S = 1
Decimal to binary. 13.25 = 13 + 0.25 = 1101.01 = 1.10101 x (2^3)
Actual exponent = 3 and bias = 127. Exponent = 3 + 127 = 130 = 10000010.
Fraction = 10101000000000000000000 (23 bits and padded with zeros)
32-bit binary = 1100 0001 0101 0100 0000 0000 0000 0000
Hexadecimal = 0xC1540000

b. 0.1 (single precision)

Sol:

Sign is positive , so S = 0
Decimal to binary. 0.1 = 0 + 0.1 = 0.0001100110011… = 1.1001100110011… x (2^-4)
Actual exponent = -4 and bias = 127. Exponent = -4 + 127 = 123 = 01111011.
Fraction = 10011001100110011001101 (23 bits and as 24th bit is 1 increment 23rd bit)
32-bit binary = 0011 1101 1100 1100 1100 1100 1100 1101
Hexadecimal = 0x3DCCCCCD

c. 156.75 (double precision)

Sol:

Sign is positive , so S = 0
Decimal to binary. 156.75 = 10011100.11 = 1.001110011 x (2^7)
Actual exponent = 7 and bias = 1023. Exponent = 7 + 1023 = 1030 = 10000000110.
Fraction = 0011100110000000000000000000000000000000000000000000

64-bit binary = 0100 0000 0110 0011 1001 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Hexadecimal = 0x4063980000000000

d. -0.0078125 (double precision)

Sol:

Sign is negative so S = 1
Decimal to binary = 1.0 x (2^-7)
Actual exponent = -7 and bias = 1023. Exponent = 1016 = 01111111000.
Fraction = 00000000….0(52 zeros)
64-bit binary = 1011 1111 1000 000…0(fraction)
Hexadecimal = 0xBF80000000000000

---

2. Convert the following hexadecimal values into their decimal equivalents. Show steps. [6 marks]

a. 0xC1200000 (single precision)

Sol :

32-bit binary = 1 10000010 01000000000000000000000
Fraction = 010 0000 0000 0000 0000 0000 = 01 = 0.25
Exponent = 10000010 = 130 and bias = 127. Actual exponent = 130 - 127 = 3.
S = 1 so negative .
Decimal equivalent = -1.01 x (2^3) = - 1010 = -10.0

b. 0x3F800000 (single precision)

Sol:

32-bit binary = 0 01111111 00000000000000000000000
Fraction = 00000000000000000000000 = 0.
Exponent = 01111111 = 127 and bias = 127. Actual exponent = 127 - 127 = 0.
S = 0 so positive.
Decimal equivalent = +1 x (2^0) = +1.0

c. 0xBFF0000000000000 (double precision)

Sol:

64-bit binary = 1 01111111111 000…0(52 zeros)
Fraction = 000…0(52 zeros) = 0.
Exponent = 01111111111 = 1023 and bias = 1023. Actual exponent = 1023 - 1023 = 0.
S = 1 so negative.
Decimal equivalent = -1.0 x (2^0) = -1.0

d. 0x4024000000000000 (double precision)

Sol:

64-bit binary = 0 10000000010   0100 00..0(48 zeros)
Fraction = 01000….0 = 01 = 0.25
Exponent = 10000000010 = 1026 and bias = 1023. Actual exponent = 1026 - 1023 = 3.
S = 0 so positive.
Decimal equivalent = +1.25 x (2^3) = +10.0

---

3. You are given two IEEE-754 single-precision numbers as 32-bit hex values: [4 marks]
A = 0x41480000 (single-precision)
B = 0xC0700000 (single-precision)
Perform the addition A + B and write the final answer in IEEE-754 double precision format.

Sol:

A in binary:
0 10000010 10010000000000000000000
Sign is positive because S = 0.
Actual exponent = Exponent - bias = 10000010 - bias = 130 - 127 = 3
Fraction = 1001 in binary = 0.5625 in decimal
A = +1.1001 x 2^3 in binary= +12.5 in decimal

B in binary:
1 10000000 11100000000000000000000
Sign is negative because S = 1.
Actual exponent = Exponent - bias = 128 - 127 = 1

Fraction = 111 in binary = 0.875 in decimal
B = -1.111 x 2^1 = -3.75 in decimal

A + B = (+1.1001 x 2^3) + (-1.111 x 2^1)
= (+1.1001 x 2^3) + (-0.01111 x 2^3) (align binary points)
= +1.00011 x 2^3 in binary (add significands and normalize)
= +1000.11
= 8.25 in decimal

S = 0 as A+B is positive.
E = actual exponent + bias = 3 + 1023 = 1026 = 10000000010
F = 00011000 00..0(44 zeros)
A+B = 0100 0000 0010 0001 1000 + 44 zeros = 0x4021800000000000
in IEEE-754 double precision format

---

4. You are given two IEEE-754 double-precision numbers as 64-bit hex values: [4 marks]
A = 0x4039000000000000 (double precision)
B = 0xC008000000000000 (double precision)
Perform the multiplication A x B and write the final answer in IEEE-754 single precision format.

Sol:

A in binary and decimal:
0 10000000011 1001 + 48 zeros
S = 0 so positive
Actual exponent = exponent - bias = 1027 - 1023 = 4
F = 1001 = 0.5625 in decimal
A = +1.1001 x 2^4 in binary = 25 in decimal

B in binary and decimal:
1 10000000000 1000 + 48 zeros
S = 1 so negative
Actual exponent = exponent - bias = 1024 - 1023 = 1
F = 1000 = 1 = 0.5 in decimal
B = -1.1 x 2^1 in binary = -3 in decimal

A x B = (2.5 x 10^1) x (-3 x 10^0) = -7.5 x 10^1 = -75 in decimal

AxB in IEEE-754 single precision format:

Sign is negative so S = 1
-75 = -1001011 = -1.001011 x 2^6
E = actual exponent + bias = 6 + 127 = 133 = 10000101
F = 00101100000000000000000
A x B = 1100 0010 1001 0110 0000 0000 0000 0000
A x B = 0xC2960000

---

5. Identify and explain one number which can be represented in a 32-bit signed integer format, but not in a 32-bit single precision floating point representation. [2 marks]

Sol:

32-bit signed integer can represent integers from -(2^31) to +(2^31 - 1) i.e from -2147483648 to +2147483647 . But IEEE-754 32-bit single precision floating point number has only 24 bits of precision due to 23 bits in mantissa and a leading 1 bit implicitly. So it can only represent integers upto 2^24. Beyond this , we cannot represent integers exactly because they require more precision than what we have. So we cannot represent 2^24 + 1 = 16777217 in IEEE-754 32-bit single precision floating point number but it can be represented in 32-bit signed integer.

---

6. Show one example to prove that addition is not associative for floating point numbers i.e., (a + b) + c ≠ a + (b + c) [3 marks]

Sol:

Choose a = 1.0 , b = 1.0 x 10^-7 , c = -1.0

(a+b) + c = (1.0 + (1.0 x 10^-7)) + (-1.0) = (1.0) + (-1.0) = 0.0
Here 1.0 x 10^-7 is rounded off to 0 while adding to 1 as we cannot represent it in single precision floating point.

a + (b + c) = (1.0) + (1.0 x 10^-7 + (-1.0)) = (1.0) + (-0.9999999) = 0.0000001 = 1.0 x 10^-7

So , addition is not associative for floating point numbers.

$(a + b) + c \neq a + (b + c)$