Justin Choi
February 13th 2017
CS 2150

Floating Point Conversion
CS 2150 Lab 4: Pre-Lab

1. Your magic (32 bit) floating point number is 2.42578125 This is the number that needs to be converted to (little endian) binary, and expressed in hexadecimal.

Sign: (Positive)

Encoded: 0

Exponent:

2.42578125/2^1 =1.212890625

1 + 127 = 128

128/2=64 r 0

64/2=32 r 0

32/2=16 r 0

16/2=8 r 0

8/2=4 r 0

4/2=2 r 0

2/2=1 r 0

1/2=0 r 1

Encoded: 1000 0000

Mantissa:

1.212890625-1=0.212890625

0.212890625=1/8 + 1/16 + 1/64 + 1/128 + 1/512 = (1/2)^3 + (1/2)^4 + (1/2)^6 + (1/2)^7 + (1/2)^9

Encoded: 0011 0110 1000 0000 0000 000

Justin Choi
February 13th 2017
CS 2150

In Binary: 0100 0000 0001 1011 0100 0000 0000 0000

In Hex (Big-endian): 0x401b4000

In Hex (Little-endian): 0x00401b40

∴2.42578125 = 0x401b4000 (big endian) = 0x00401b40 (little-endian)

2. Your other magic floating point number is, in hex, 0x00401fc1 This is the number that needs to be converted to a (32 bit) floating point number. Note that the hexadecimal printed above is in little-endian format!

In Hex (Little-endian): 0x00401fc1

In Hex (Big-endian): 0xc11f4000

In Binary: 0xc11f4000 = 1100 0001 0001 1111 0100 0000 0000 0000

Sign: (negative, since the first bit is 1)

Exponent:

1000 0010 = 0x82 = 82_f = 8 × 16^1 + 8 × 16^0 = 130

130 – 127 = 3

2^3=8

Mantissa:

001 1111 0100 0000 0000 0000

(1/2)^3 + (1/2)^4 + (1/2)^5 + (1/2)^6 + (1/2)^7 + (1/2)^9 = 0.244140625

1 + 0.244140625 = 1.244140625

In Float:

1.244140625 × 2^3 = 1.244140625 × 8 = 9.953125 (sign not taken care of yet)

∴since,the sign bit is negative, 0x00401fc1 (little endian) = 0xc11f4000 (big endian) =

Justin Choi
February 13th 2017
CS 2150

-9.953125