

Matt Calayo mjc5gh floatingpoint.pdf 2/14/18

Your magic (32 bit) floating point number is -8.828125

This is the number that needs to be converted to (little endian) binary, and expressed in hexadecimal.

Your other magic floating point number is, in hex, 0x00c01f40

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!

-8.828125

Sign bit is 1 because the number is negative

Exponent: $8.828125/2^3 = 8.828125/8 = 1.103515625$

Exponent is $3 + 127 = 130$ (1000 0010)

Mantissa:

$1.103515625 - 1 = .103515625$

$.103515625 = 1/16 + 1/32 + 1/128 + 1/512$

$= 0001\ 1010\ 1000\ 0000\ 0000\ 000$

In binary:

0000 0000 0000 0100 1101 0000 0001 1100

In hex:

0x00400dc1

0x00c01f40

Swapped to big-endian is 0x401fc000

Converted to binary is 0100 0000 0001 1111 1100 0000 0000 0000

Sign digit is 0 which means positive number

Exponent: $1000\ 0000 = 128$

Exponent offset $= 128 - 127 = 1$

Multiply the mantissa by $2^1 = 2$

Mantissa: $0011\ 1111\ 1000\ 0000\ 0000\ 000 = 1/8 + 1/16 + 1/32 + 1/64 + 1/128 + 1/256 + 1/512 =$

$0.248046875 + 1 = 1.248046875$

Final number: $2^1 * 1.248046875 = 2.49609375$