Adarsh Solanki
As5nr
2/13/12
floatingpoint.pdf

Magic 32-bit floating point number: -38.875

- Sign bit = 1 because number is negative
- Whole number part = 38
  - 38/2 = 19 remainder 0
  - 19/2 = 9 remainder 1
  - 9/2 = 4 remainder 1
  - 4/2 = 2 remainder 0
  - 2/2 = 1 remainder 0
  - 1/2 = 0 remainder 1
  - Result = 100110
- Decimal part = .875
  - .875 x 2 = 1.75 1
  - .75 x 2 = 1.5 1
  - .5 x 2 = 1 1
  - = 0.111

Number = 100110.111
= 1.00110111×2^5

1-bit sign bit = 1
8-bit exponent = 5 + bias of 127 = 132 = 1000 0100
23-bit Mantissa = 00110111000000000000000

1 1000 0100 00110111000000000000000
1100 0010 0001 1011 1000 0000 0000 0000
0xc21b8000 (big endian)
0x00801BC2 (little endian)

Other number is 0x00401f41 little-endian

Convert to big endian: 411f4000

Expand to decimal: 0100 0001 0001 1111 0100 0000 0000 0000
Sign-bit = 0

8-bit exponent = 1000 0010 – bias of 127 = 0000 $0011_2 = 3_{10}$

23-bit mantissa = 0011 1110 1000 0000 0000 000
$1.001111101 \times 2^3 = 1001.111101$
$1*2^3 + 0*2^2 + 0*2^1 + 1*2^0 + 1*2^{-1} + 1*2^{-2} + 1*2^{-3} + 1*2^{-4} + 0*2^{-5} + 1*2^{-6}$
8 + 1 + ½ + ¼ + 1/8 + 1/16 + 1/64
9.953125