IEEE 754 Conversion (32-bit Single Precision)

Bit Fields
Sign: 1 bit (31), 0=positive, 1=negative
Exponent: 8 bits (30-21), excess 127
Mantissa: 21 bits (20-0), normalized base 2 fraction

Note on Bit Pattern Representation
When a picture showing an IEEE 754 bit pattern is displayed, bits are numbered 0 to 31 from right to left. This is consistent with the convention that 0 is the least significant bit (LSB) and 31 is the most significant bit (MSB).

Conversion #1: Bits-to-Float (the "easy" direction)

- Divide the 32 bits value into three fields
- Convert the exponent from unsigned binary to unsigned decimal and subtract 127, call value e
- Convert mantissa to a floating point number between 1 and 1.999, call value m
- Float value = +/- m X 2^e

Worked Examples

- Bits = 43FC0000
- Binary = 0100 0011 1111 1100 0000 0000 0000 0000
- Sign = 0
- Exponent = 100 0011 1 = 1000 0111 = 128 + 7 = 135; 135 – 127 = 8
- Mantissa = 111 1100 0000 0000 0000 0000 = 1 + .5 + .25 + .125 + .0625 + .03125 = 1.96875
- Float = + 1.96875 X 2^8 = + 1.96875 X 256 = 504.0

Conversion #2: Float-to-Bits (the “harder” direction)

- Let f be the float value.
- Determine the largest power of two that is not greater than f, call it p
- $f = f/2^p \times 2^p$
- $m = f/2^p$, subtract 1 and convert remaining value to binary with each bit position a negative power of two
- e = p, add 127 and convert to binary
- If f is negative, sign=1, else sign=0

Worked Examples

f = 1208.0, normalize to f = (1208/1024) X 1024 = 1.1796875 X 2^{10}
m = 1.1796875, ignore the 1, m – 1 = 0.1796875, convert remainder to binary
Subtracting negative powers of two, lookup table could be helpful

Exponent	Decimal
$2^{-1} = 1/2$	0.5
$2^{-2} = 1/4$	0.25
$2^{-3} = 1/8$	0.125
$2^{-4} = 1/16$	0.0625
$2^{-5} = 1/32$	0.03125
$2^{-6} = 1/64$	0.015625
$2^{-7} = 1/128$	0.0078125
$2^{-8} = 1/256$	0.00390625
$2^{-9} = 1/512$	0.001953125
$2^{-10} = 1/1024$	0.0009765625

0.1796875 – 0.125 (2^{-3}) = 0.0546875
0.0546875 – 0.03125 (2^{-5}) = 0.0234375
0.0234375 – 0.015625 (2^{-6}) = 0.0078125
0.0078125 – 0.0078125 (2^{-7}) = 0

So mantissa = 00101110000000000000000

e = 10, add 127, so e + 127 = 137
Convert to unsigned binary: 137 = 128 + 8 + 1 = 10001001

Sign = 0

Complete 32 bit formatted number
= 0 + 10001001 + 00101110000000000000000
= 0100 0100 1001 0111 0000 0000 0000 0000
= 44970000

Alternative: Mantissa Table Lookup
In reality, the mantissa is 21 bits, which allows accuracy down to the level of 2^{-21}. But this table will only show mantissa values for 5 bits. You can directly calculate more for specific cases. For any entry, you should be able to explain the relationship. For example

01011 = 0.34375
01011 = $2^{-2} + 2^{-4} + 2^{-5}$ = 0.25 + 0.0625 + 0.03125 = 0.34375

For worked example #2, rather than directly converting the decimal to binary, we could look up the closest value in the table not greater than the decimal value.
Actual decimal value = 0.1796875
Closest value not greater than from table = 0.15625
Equivalent binary = 00101

Bits	Decimal
00000	0.00000
00001	0.03125
00010	0.06250
00011	0.09375
00100	0.12500
00101	0.15625
00110	0.18750
00111	0.21875
01000	0.25000
01001	0.28125
01010	0.31250
01011	0.34375
01100	0.37500
01101	0.40625
01110	0.43750
01111	0.46875
10000	0.50000
10001	0.53125
10010	0.56250
10011	0.59375
10100	0.62500
10101	0.65625
10110	0.68750
10111	0.71875
11000	0.75000
11001	0.78125
11010	0.81250
11011	0.84375
11100	0.87500
11101	0.90625
11110	0.93750
11111	0.96875

Exercises

Convert 232.0 to IEEE 754.

f = 232.0

Largest power of two not larger than f: ________________

f in normalized form: ________________

____

sign: ________________

exponent (add 127, convert to unsigned bin): ________________

____

mantissa (sub 1, use table to convert to bin): ________________

____

final format in hex: ________________

____

Convert 3D580000 to float.

binary: ________________

sign bit: ________________

exponent bits: ________________

mantissa bits: ________________

exponent value (convert to dec, sub 127): ________________

____

mantissa value (use table to get value, add 1): ________________

____

normalized form: ________________

____

value: ________________

____