Floating Point Numbers (IEEE Standard 754)

Why? We need to represent real numbers!

Single precision FP (32 bit):

$$\textbf{FP value} = (-1)^S \times (1 + F) \times 2^{(E - bias)}$$

Sign	Exponent (E)	Fraction (F) / Mantissa
31	23	0

For single precision FP, *S* = 1 bit, *E* = 8 bits, *F* = 23 bits, *bias* = 127.

For double precision FP, *S* = 1 bit, *E* = 11 bits, *F* = 52 bits, *bias* = 1023.

Question: Why do we use a bias?

"Special" single precision FP values:

±Zero:	E = 0, M = 0	NaN:	E = 255, M ≠ 0
±Infinity:	E=255, M = 0	Denormalized:	E = 0, M ≠ 0

(More on denormal numbers: http://en.wikipedia.org/wiki/Denormal_number)

Question: Convert the single precision FP representation, 0xC0B40000, to decimal.

Now we know how to convert from FP representations to decimals, how about the other way around? Google is always your best friend. For example, try this website: http://www.cs.cornell.edu/~tomf/notes/cps104/floating.html#dec2hex

MIPS Revisited

Since your project 2 is all about MIPS (and so is project 4, the MIPS datapath), we decide to give you a quick taste of how to decode MIPS instructions. Remember, each instruction in MIPS is a number!

Question: Convert "addi $t1, $t0, 5" to its HEX representation.

Question: Decode the following program and describe its function.

Memory Address	Instruction
0x00	0x0085402A
0x04	0x11000002
0x08	0x00A01020
0x0c	0x03E00008
0x10	0x00801020
0x14	0x03E00008

Memory Address	Instruction
0x00	
0x04	
0x08	
0x0c	
0x10	
0x14	