Homework 1

25% of homework grade, 2.5% of overall grade
out: 10/2 in class; in: 10/9 at the beginning of class

Integer and Floating Point Number Representations

Please type your homework. If you write make sure you write legibly. Show your work.

Integer

Some modern processors, such as the DEC Alpha (now owned by Intel), have a 64 bit (instead of 32 bit) two's complement integer format. For this section of the homework, assume such a format.

Problem 1

Suppose you have a 1 GHz processor and you can execute a 64 bit integer addition every cycle. How long will the following loop run?

```
unsigned long int i;  /* 64 bit integer */
for (i=0;i<(unsigned long)(0xdeadbeefL);i++) {}
```

Problem 2

A Full Adder (FA) logic block takes three input bits and provides two output bits. The input bits are summed to produce a two bit output. (a) Write a table showing what each combination of input bits produces at the output. (b) Draw a picture of how you might connect FAs together to create a (slow) 64 bit adder. You don't have to show every FA, just convey the idea of your solution. (c) Show how you could turn your 64 bit adder into a 64 bit subtractor using 64 NAND gates. If you're interested in how a *fast* adder works, do a google search for the term "carry lookahead adder". If you're interested in how addition is done in a multiplier, look for "carry save adder".

Problem 3

Write a small piece of C code that can determine how many bits there are in an unsigned long int on any machine. You are not allowed to use sizeof. You may use a loop.

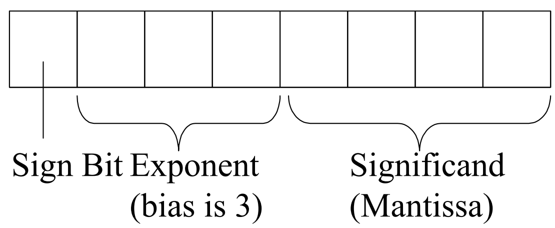
Problem 4

Most instruction sets provide an "add with carry" and a "subtract with borrow" instruction. (a) Explain how you might use these to implement arbitrary bit length integer representations. (b) Are the instructions necessary to do this? Explain why or why not.

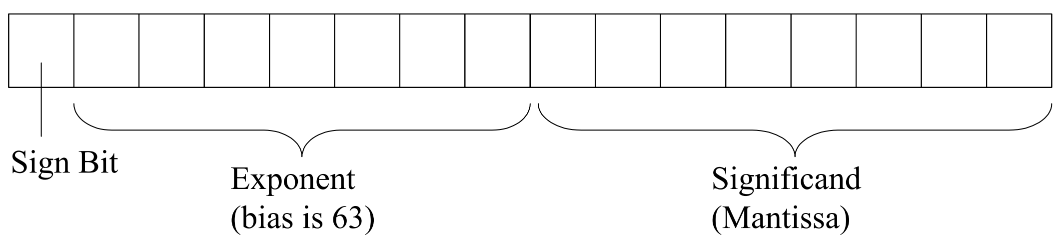
Floating Point

Consider the following two small floating point formats based on the IEEE standard:

- Little Format



- Big Format



Except for the sizes of these formats, the rules are those of the IEEE standard.

Problem 1

For both formats, determine the following values (in binary, hex, and decimal)

1. Largest positive finite number
2. Positive normalized number closest to zero
3. Largest positive denormalized number
4. Positive denormalized number closest to zero

Problem 2

Encode the following values in the 8 bit Little Format: ¾, -13/16, 44, and –104, show each in binary and hexadecimal.

Problem 3

Determine the values corresponding to the following Little Format bit patterns. The leftmost bit is the most significant

1. 1 011 0011
2. 0 111 1010

3. 1 001 0001
4. 0 100 1111
5. 1 100 0001

Problem 4

Convert the following 16 bit Big Format numbers into 8 bit Small Format numbers. Overflow should yield +/- infinity, underflow should yield +/- 0.0, and rounding should follow the "round-to-nearest-even" tie-breaking rule.

1. 0 0000010 00000000
2. 1 0011101 00000000
3. 0 0000110 00001111
4. 1 0001001 00000110
5. 1 1111111 10101010