16.482 / 16.561: Computer Architecture and Design

Spring 2015

Homework #2

Due **Thursday, 2/5/15**

Notes:

- While typed submissions are preferred, handwritten submissions are acceptable.
- Any electronic submission must be in a single file. Archive files will not be accepted.
- Electronic submissions should be e-mailed to Dr. Geiger at Michael_Geiger@uml.edu.
- This assignment is worth a total of 100 points.

1. (15 points) Show how the "refined multiply hardware" (slide #19 from Lec. 2) multiplies the 8-bit values 88 and 45, using an approach similar to the one demonstrated in lecture.

2. (10 points) In class, we briefly discussed that the hardware methods presented cannot correctly handle signed multiplication. Describe a method that could correctly handle all multiplication of signed integers without increasing the size of the registers or adders used. (In other words, any other algorithm that adds extra bits while performing the actual multiplication—including Booth's algorithm, for example—is not a valid solution.)

3. (25 points) Convert each of the following decimal values into single-precision IEEE floating-point format. Show all steps, including how you calculate the fraction and biased exponent stored in the number. *(Note: I encourage you to convert each result into hexadecimal, which will help ensure that your assignments are graded and returned relatively quickly!)*

a. -9.625
b. 23
c. 0.921875
d. -100.125
e. 2.05 (determine the closest approximation you can)

4. (25 points) Convert each of the following IEEE single-precision floating-point values into decimal values. Show all steps of your work.

a. 0x40540000
b. 0xbfb00000
c. 0x3f538000
d. 0xc2060000
e. 0xaabbccdd (determine the closest approximation you can)

5. (25 points) Compute the result of each floating-point arithmetic arithmetic operation below, in which each of the values is encoded in single-precision IEEE floating-point format. Recall that:

   - For floating-point addition, align the binary points, add the significands, then normalize the result.
   - For floating-point multiplication, add the exponents (taking care to only account for the bias once), multiply the significands, normalize the result, and then determine the sign.

All arithmetic should be done in binary, and results should be re-encoded in single-precision IEEE floating-point format.

a. `0x41e00000 + 0x42280000`
b. `0x40b80000 * 0x40500000`
c. `0xc1280000 + 0xc2308000`
d. `0x41240000 * 0xc1110000`