<u>Assignment #5</u>

(max = 95)

Read pages 178-196 in the *Computer Organization and Design* text. I have provided a set of notes ("Notes for Assignment #5) on this reading that can be found under Course Notes. Please refer to these notes as you <u>carefully</u> work through the assigned reading.

Afterwards, submit answers for the following problems:

1. Multiply 10_{10} by 11_{10} (the multiplier) using the hardware of Figure 3.3. Produce a table similar to Figure 3.6. As the text has done, use 4-bit (unsigned) numbers, rather than 32-bit numbers! (10 points)

2. This time, multiply 11_{10} by 12_{10} (the multiplier). Use the refined version of the hardware given in Figure 3.5, producing a table similar to the one that appears in the course notes. Use 4-bit (unsigned) numbers. (10 points)

3. Divide 14_{10} by 3_{10} using the hardware of Figure 3.8. Produce a table similar to Figure 3.10 (use my slightly modified algorithm that starts with Step 3 for the first iteration). Use 4-bit (unsigned) numbers. (10 points)

4. Divide 14_{10} by 3_{10} again. This time use the improved <u>non-restoring</u> version of the division algorithm. Produce a table like the one that appears in the course notes. Use 4-bit (unsigned) numbers. (10 points)

5. Consider the following sequence, which I'll refer to as the alternating Fibonacci sequence:

   1 -1 2 -3 5 -8 13 …

   Here $altfib_1 = 1$, $altfib_2 = -1$ and $altfib_n = altfib_{n-2} - altfib_{n-1}$ for $n > 2$. Write a MIPS program (call it **altfib.s**) that will produce and print numbers (5 per line) in the alternating Fibonacci sequence in such a way that the code detects when overflow takes place. The "offending" number should not be in your list of numbers, but you should display the bogus value that is produced [see my output; the next value in the list would have been 1134903170 – (-1836311903) = 1134903170 + 1836311903 = 2971215073, which is too large for a 32-bit 2's complement number; instead, it is interpreted as -1323752223]. You should use the elaboration on page 182 as a guide, but notice that you will need to alter things slightly since you are taking the difference of two numbers, not the sum. You might want to (carefully) use the "negu" instruction on page A-54. Here is output from my program:

```
Here are the alternating Fibonacci numbers that I produced:

1 -1 2 -3 5
-8 13 -21 34 -55
89 -144 233 -377 610
-987 1597 -2584 4181 -6765
10946 -17711 28657 -46368 75025
-121393 196418 -317811 514229 -832040
1346269 -2178309 3524578 -5702887 9227465
-14930352 24157817 -39088169 63245986 -102334155
165580141 -267914296 433494437 -701408733 1134903170
-1836311903

Value causing overflow = -1323752223
```

Don't forget to document your code! Submit a separate file called **altfib.s** as well as placing your code in this assignment submission; the Mentor will clarify what I mean by this. (50 points)

6. Recall that in question 9 in Assignment #2, we displayed the values of fact(n) for various values of n. We saw that you could only go up to a certain value of n and still expect to get a valid result. We also saw that eventually (as the value of n increased) the values being returned were simply zero. Now that you know how multiplication works, explain both of these phenomena (incorrect result and zero result). There is a way that you could have predicted the first value of n that would produce a result of zero. Explain that process. (5 points)

Your assignment is due by 11:59 PM (Eastern Time) on the assignment due date (consult Course Calendar on course website).