Assignment #2

(max = 90)

Read pages 62 through 102 of the *Computer Organization and Design* text (we will come back to the remainder of Chapter 1 in a later assignment). I have provided an extensive set of notes ("Notes for Assignment #2) on this reading that can be found under Course Notes. Please refer to these notes as you carefully work through the assigned reading.

Afterwards, submit answers for the following problems:

1. This is an adaptation of Exercise 2.3 from page 165 in the textbook. Assume that variables f, g, h, i and j are assigned to registers $s0, $s1, $s2, $s3 and $s4, respectively. Also, assume that the base address of the arrays A and B are in registers $s6 and $s7. For each of the following C statements, what is the corresponding MIPS code? (10 points total)

    a. f = g – A[4]; (2 points)
    b. f = B[i]; (3 points)
    c. B[8] = A[i-j]; (5 points)

2. What is the decimal value of the following? (4 points total … 2 points each)

    a. 1010111_2 (show calculation)
    b. $ab9_{16}$ (show calculation)

3. Given the 32-bit binary number: (detailed calculations not required)

    $1000\ 1001\ 1000\ 1001\ 0000\ 1011\ 0000\ 0011_2$ (5 points total)

    a. Write this number in hexadecimal. (1 point)
    b. As an unsigned integer, what is this number's decimal value? (1 point)
    c. As a two's complement integer, what is the decimal value? Explain process, but detailed calculations not required. (3 points)

4. Given the following hexadecimal number: (detailed calculations not required)

    $abcd1234_{16}$ (5 points total)

    a. Express this number in binary. (1 point)
    b. As an unsigned integer, what is this number's decimal value? (1 point)
    c. As a two's complement integer, what is the decimal value? Explain process, but detailed calculations not required. (3 points)

5. Changing "44" to "4", do Exercise 2.19.1 from page 168 in the text. (3 points)

6. Do Exercise 2.19.3 from page 168 in the text. (3 points)

7. For each of the following instructions, indicate the content of each instruction field (in decimal), the binary representation of the instruction in memory (show as 8 groups of 4 bits) and the equivalent hexadecimal representation of the instruction: (10 points)

```
sw      $t9, 60($sp)
add     $a2, $s6, $fp
```

8. Recall that I have placed the **leaf_example.s** driver program from the "Notes on Assignment #2" document under Course Materials. Download that file onto your machine, get into QtSpim and load the driver program. (12 points total)

   a. The first line in the actual leaf_example procedure is

      ```
      addi    $sp, $sp, -4
      ```

      At what address is that instruction stored? What is the hexadecimal representation of the instruction? Is this an I-format or R-format instruction? What is the value (in binary) of the constant field? (4 points)

   b. Run the program using the values 19, 64, -23 and 72 (in that order). What is the value returned by leaf_example? (1 point)

   c. Even though the program has completed, the value of i (-23) should still be in register $a2. How is that value represented in hexadecimal? What is the binary equivalent of that value? (2 points)

   d. Reinitialize and load the **leaf_example.s** code. Sometimes it is convenient to inspect the registers at a particular place in the execution of the code. Notice that the instruction immediately after the "jal leaf_example" instruction is "move $s0, $v0" (move is actually a pseudo-instruction; it is replaced by an "addu" instruction). This instruction is stored at address 0x004000c0. Also notice that the value returned by the leaf_example procedure should be in register $v0 when this instruction is about to execute. We can check this out by setting a breakpoint in the code. You can set a breakpoint by right-clicking on the line of code and selecting "set breakpoint". Now execute the program, using the values 10, 20, 30 and 40 (in that order). The program will pause when it is about to execute the instruction at your breakpoint. Select the "Abort" option. What value is in $v0 at this point? What values are in $a0 through $a3? (5 points)

9. Recall that I have placed the code for the fact function from the "Notes on Assignment #2" document under Course Materials (as **fact.s**). Download that file onto your machine. Starting with that code, add a driver program that will allow you to test the factorial function. Here is a sample execution of my program:

```
Welcome to the factorial tester!
Enter a value for n (or a negative value to exit): 0
0! is 1
Enter a value for n (or a negative value to exit): 6
6! is 720
Enter a value for n (or a negative value to exit): 9
9! is 362880
Enter a value for n (or a negative value to exit): 11
11! is 39916800
Enter a value for n (or a negative value to exit): 20
20! is -2102132736
Enter a value for n (or a negative value to exit): 100
100! is 0
Enter a value for n (or a negative value to exit): 3
3! is 6
Enter a value for n (or a negative value to exit): -1
Come back soon!
```

Your program should produce output that mirrors mine. Be sure to document your code (as I did in **leaf_example.s**). Submit a separate file called **fact.s** as well as placing your code in this assignment submission; the Mentor will clarify what I mean by this.

Look carefully at the execution results. Clearly there is a problem! There is no way that 20 factorial is a negative number! What is the largest value of n that we can use and get a correct result? What is that correct result (you might want to verify with a calculator)? Notice that I get a value of 0 for 100 factorial. Likewise, for 101 factorial, and so on. What is the largest value of n that produces a non-zero result (of course, it is still an incorrect result)? What do you think the problem is here? (38 points)

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