CSE 378 - Spring 2001

Machine Organization and Assembly Language Programming

Problem Set #1

Due Monday, October 8th, 2001

By the end of the first week, read Chapter 1, section 2 (the rest of Chapter 1 is optional), but suggested); Chapter 4, Sections 4.1 to 4.4; Chapter 3, Sections 3.1 to 3.4.; Appendix A.1 - A.4 Also read "The Big Picture" on p. 299.

Be sure you subscribe to the mailing list - send a message reading, "subscribe cse378" to majordomo@cs. ("subscribe cse378" must be the content, not the subject; leave the subject line blank)

Do the following problems, and hand in the answers by the start of class on Monday, October 8th:

1. Exercises 1.1 through 1.26, inclusive. (3 pts)
2. Exercise 4.1, with results in both binary *and* hexadecimal. (4 pts)
3. Exercise 4.3. (3 pts)
4. Exercise 4.4. Use the shortcut. Also translate the binary number into (unsigned) hexadecimal. (4 pts)
5. Exercise 4.7. (3 pts)
6. Exercise 4.11. (8 pts)
7. The Big Picture on page 299 mentions that bits have no inherent meaning. Given the bit pattern:

0100 0010 0111 1011 0001 1001 0010 0010

What does it represent, assuming it is:
a. A 32-bit two's complement integer? (Specify the result in decimal.) (4 pts)
b. A 32-bit unsigned integer? (Specify the result in decimal.) (4 pts)
c. Four consecutive, 8-bit ASCII values? (4 pts) *Use the ASCII table on p. 142 for the standard character values. Also note the following special codes which do not appear in the table:*

```
ASCII 0 = NUL (C string null terminator)ASCII 8 = BS (Backspace)     ASCII 9
= TAB ASCII 10 = LF (Line Feed) ASCII 13 = CR (Carriage Return)
```

8. Assume a 4-bit register and a 2's complement representation of integers. Give examples of adding two positive numbers with and without overflow, and of subtracting a negative number from a positive number with and without overflow. Show the 2's complement representation of the operands and the result in a manner similar to what is done in the book, pp. 220-221. (8 pts)