ECE199JL: Introduction to Computer Engineering **Fall 2012**
Homework 2 **Due: at start of lecture on Friday 7 September**

Operations on Bits

We are shifting the homework topics slowly so that you should see everything on your homework by Monday of the week it is due. The program this week will make use of the Wednesday material, since you only need to execute it.

Please do problems 2.20 and 2.24 from the textbook.

Here are five additional problems for this week:

1. Truth Tables
Let's say that we want to add two 2-bit numbers A and B using signed-magnitude representation.

A. How many combinations of the two operands are possible?

B. Write a truth table showing as output the 3-bit sum $S = A + B$ for each combination of A and B. Your truth table should have one row for each possible combination of the inputs (your answer to Part A); please order these in binary order on A and B. That is, first $A = 00$ and $B = 00$, then $A = 00$ and $B = 01$, then $A = 00$ and $B = 10$, and so forth, ending with $A = 11$ and $B = 11$.

C. Indicate the rows of the truth table for which more than one possible output pattern can be considered correct. (Draw arrows to all such rows, for example.)

2. Designing Representations
Look in the back of the book at the table illustrating the ASCII representation for English text. The table gives decimal and hexadecimal values corresponding to the 7-bit patterns used for ASCII. Note that the digits 0 through 9 are assigned consecutive binary patterns (consecutive integers in the table). Explain why this choice is useful. In other words, explain why a representation in which we assigned random distinct patterns for these ten digits would not be as good as the representation shown in the table.

3. Fun with integers and ASCII
Express the 32-bit binary integer 0100 0101 0100 0011 0100 0101 0010 0000 in hexadecimal. Interpret the four successive 8-bit bytes making up this integer as a four-character ASCII string; what does it say? Interpret the following hexadecimal string 20202001 as a 32-bit binary integer (write down that binary integer) and add it to the above integer to get a new 32-bit string. Interpret the four successive 8-bit bytes making up this new integer as a four-character ASCII string; what does it say?

4. Human-Friendly Notation
A. Explain why most engineers prefer to use hexadecimal (base 16) or octal (base 8) notation when discussing specific patterns of bits rather than using binary (base 2) notation directly.

B. Explain why most engineers prefer to use hexadecimal (base 16) or octal (base 8) notation when discussing specific patterns of bits rather than using decimal (base 10) notation.

5. Examining Logic Operations
This week's homework again requires only that you obtain and use a program. For interest, you may want to look at how the program `hw2.c` executes logic operations on bits.

Starting in your home directory, obtain a copy of the `hw2` directory from the class directory:

```
cp -r /class/ece199/hw2 .
```

Then change into the `hw2` directory (using: `cd hw2`) and build the program by typing: `make`.

You can run the program from the `hw2` directory by typing: `./hw2`

Use the program to solve problem 2.50 from the textbook.