ECE 252 Homework 5 Due Monday, Nov 1st

Instructions: This homework is to be done in your assigned groups. You should hand in ONE copy of the homework that lists your section number, names and UW ID number of all students. You must staple all pages of your homework together to receive full credit.
First contact for questions is TA Philip Garcia pcgarcia@wisc.edu

- **Problem 1 (2 points)**
  Every instruction in the system can be expressed both as machine code (its bit pattern), as well as the instruction name. For example the instruction "`ADD R3, R4, R5`" is `0001011100000101` in machine code. What instructions do the following machine code instructions represent?
  `1001010111111111`
  `0101111010000100`
  `0001100100100011`
  `1100000111000000`

- **Problem 2 (4 points)**
  Using the memory contents shown below, what would be the contents of Registers R1, R2, R3, R4, and R5 after executing the sequence of 4 instruction below (where execution starts at 0x3000). Assume when the instruction sequence starts, R1=0x3004, R2=0x3009, R3=0x0015, R4=0x3003, and R5=0x300B.

Address	**Instruction**
0x3000	LD R1, #9
0x3001	LDI R3, #9
0x3002	LEA R4, #8
0x3003	LDR R5, R2, #3
...	...
Address	**Data**
0x3009	0x0015
0x300A	0x300B
0x300B	0x300A
0x300C	0x3009

- **Problem 3 (3 points)**
  Solve Problem 5.4 on page 145 of ItCS using a memory that contains 4096 distinct locations each containing 16-bits of data.

- **Problem 4 (5 points)**

  a. How might one use a single LC-3 instruction to move the value in R3 into R4?

  b. The LC-3 has no subtract instruction. How could one perform the following operation using only three LC-3 instructions:
  R1 ← R6 − R2

  c. Using only a single LC-3 instruction, and without changing the contents of any other registers, how might one set the condition codes based on the value that resides in R3?

  d. Is there a sequence of LC-3 instructions that will cause the condition codes at the end of the sequence to be N=0, Z=1, P=1? Explain.

  e. Write an LC-3 instruction that clears the contents of R4.

- **Problem 5 (6 points)**
  The exclusive-or (XOR) operation produces a 1 if one, and only one of it inputs are 1, and has the truth table shown below. The LC-3 does not have an instruction to compute the XOR operation. Provide the set of LC-3 instructions that calculates the XOR of R1 and R2, and stores the results in R3. You can express your answers using assembly notation, and do not have to use machine code.

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

  A XOR B = ((NOT A) AND B) OR ((NOT B) AND A)
  Hint: You might want to use De Morgans law

- **Problem 6 (1 points)**
  We wish to execute a single LC-3 instruction that will subtract the decimal number 25 from register 1 and put the result in register 2. Can we do it, if yes, show how it would be done, if not, explain why not.

- **Problem 7 (1 point)**
  We want to increase the number of registers that we can specify in the LC-3 Add instruction to 32. Do you see any problem with that? Explain.

- **Problem 8 (2 points)**
  An LDR instruction, located at 0x3220, uses R3 as its base register. The value currently in in R3 is 0x4011. What is the largest address that this instruction can load from? Suppose we redefine the LDR offset to be zero-extended, rather than sign-extended. Then what would be the largest address that this instruction could load from? With the new definition, what would be the smallest address that this instruction could load from?

- **Problem 9 (1 points)**
  The purpose of this problem is to get you setup with the LC-3 simulator, which will be important for subsequent homework.
  You can download the LC-3 simulator from:
  `http://www.cis.upenn.edu/~milom/cse240-Fall06/pennsim/PennSim.jar`.
  In order to run the simulator, double click on the downloaded jar file. You can also run the simulator by opening command prompt window and going to the directory where you downloaded the simulator and typing:

  ```
  java -jar  PennSim.jar
  ```

  After opening the LC-3 simulator, assemble and load the lc3os.asm file (included from the website). Set a breakpoint at address 0x0205. Continue the execution of instructions until your break point is reached. What are the values contained in all of the registers (R0-R7, PC, MPR, PSR, and CC) at this point?