Mid-Term Exam Study Guide

What you may bring:

- A pen or pencil
- A calculator that converts between decimal, hex and binary
- A sheet of paper with hints and tips – printed or hand-written, double-sided, 8.5x11"
- A bottle of water

What you should not bring:

- Food
- Wrappers that make noise – gum wrapper, for example

What I will provide:

- A PDF reproduction of the MIPS Data Instruction Set (green card) – be advised that it does not contain everything, such as bgez
- Table A.10.2. MIPS Opcode map (the blue table)
- Table 1.1 for powers of ten
- Table 2.4 hexadecimal-binary conversion table

Themes to study:

1) Review problems 1.5 and 1.6 and be able to answer questions about the relative performance of processors given changes in instruction mixes and clock rates. Know how to calculate a CPI and clock cycles.
2) Know Amdahl's law, and be able to calculate the speedup caused by changing the performance of an instruction type.
3) Know how to manipulate MIPS instructions
    a. Given HEX, produce binary
    b. Complete an instruction format with the opcode, instruction type, and other fields from any input – the instruction (like add $t3, $s2, $0), HEX, decimal, etc.
    c. Be able to convert register names to numbers
    d. Know the various instruction types we've covered
4) Be able to convert simple C to MIPS and vice-versa
5) Be able to look at a block of MIPS code and determine what it does at a high level (sorting, matrix multiplication, converting lowercase-to-uppercase, that level)
6) Know how to define and calculate the CPI or IPC of a computer based on the data regarding clock cycle time, instruction count, types of instructions, cycles per instruction type or other similar combinations. Be able to solve for a variable given constraints like changing the speed of the program or the

number of cycles or splitting one type of instruction into two types of instructions. See **2.47.1**-**3** from HW #3 for examples, but we will go further.

7) Know how to do math with 2's complement numbers. Know the rules for converting 2's complement to standard and back. Know how to recognize overflow and underflow. Be able to convert a number from normal to 2's complement and back. This is a good tutorial from Cornell: http://www.cs.cornell.edu/~tomf/notes/cps104/twoscomp.html
8) Know how to carry numbers forward when adding and subtracting two numbers. This is a decent tutorial: http://www.swarthmore.edu/NatSci/echeeve1/Ref/BinaryMath/BinaryMath.html
9) Review the IEEE 754 Floating Point Standard. Be able to manipulate numbers:
    a. Adding, subtracting and multiplying numbers
    b. Know single- versus double-precision
    c. Know what overflow and underflow are in the floating point world (see page 198)
    d. Know how a 32-bit word is laid out according to IEEE 754
    e. Know how to convert a decimal number to a IEEE 754 floating point representation
10) Know how to do do multiplication similar to what we did in HW #4, problem 3.13. Also familiarize yourself with 3.17 and be able to take two decimal numbers and show how to shift-multiply them the way 3.17 does