# Computer Architecture

## Assignment 2

### Problem 1

Write a MIPS program to convert a decimal number to unsigned binary. Use console I/O to read in an integer from the user and to output the equivalent bit string. You may assume at most 32 bits in the representation and you don't have to do any bounds or error checking on the size of the integer. (Use shift and logic operations rather than div. You may want to write it in C++ first to get a feel for the algorithm.)

### Problem 2

Give the mask and logic operation required to perform each of the following actions on a value in an 8-bit register. (Remember we start numbering bits starting with 0 as the LSB (that is, far right bit.))

**(a)** Set bits 3 and 7.

**(b)** Clear bits 1, 2, and 3.

**(c)** Toggle bit 6.

### Problem 3

Perform the following 4-bit two's complement arithmetic and explain if overflow occurs. Show the steps of the math.

**(a)** 1101 + 1001

**(b)** 0101 + 0011

**(c)** 1110 + 0111

### Problem 4

Compute the 8-bit two's complement representation of the decimal number -55. First, convert it to unsigned binary. Then, complement that value by subtracting from $2^8$ or using the shortcut method we discussed in class. You may use a calculator to check your work, but show the steps in your calculation because the point is knowing the algorithm (yes, so we may implement it later at the machine level.)

## Problem 5

**(a)** Explain the difference between the `srl` and `sra` instructions in MIPS.

**(b)** Why is there not an `sla` instruction?

## Problem 6

**(a)** What is the largest magnitude number that can be stored in an `unsigned char` in C++?

**(b)** What is the largest magnitude **negative** number that can be stored in a `char` in C++?

**(c)** What is the range of signed numbers using a 16-bit data type?