<u>CISC 810: Fundamentals of Computational Science, Assignment 2</u>

Set September Oct 4th due Oct 18th

Notes: Questions that begin with "Research" may require you to look up auxiliary information outside of the lectures notes in class. To make this process easier, I have ensured that all the information you need can be found rapidly by Internet searches.

Q1. (a) What do the numbers 00000100_2 and 11111100_2 represent in two's complement notation? Calculate their product using binary multiplication and show your working.

(b) Prove that if $b = b_0 b_1 ... b_{n-1}$ is the n-bit representation of the integer x then shifting b *left* by i bits is equivalent to multiplying by 2^i, similarly show that a right shift by i bits is equivalent to dividing by 2^i (ignore the remainder).

Two's complement representations are derived from the notion of arithmetic on a finite cyclic group. As was discussed in the lecture, the value of 2^n is mapped back to zero (although the system has effectively overflowed in this case), *i.e.* $1111....1111_2$+1=$10000....0000_2$. For n-bits of precision we call the resulting arithmetic system 2^n *cyclic arithmetic.*

(c) If $b = b_0 b_1 ... b_{n-1}$ is an n-bit representation of an integer x prove that the two's complement of b, found by inverting every bit b_i and adding 1 to the final result, is the representation of $-x$ under 2^n cyclic arithmetic. (HINT: $2^n = 1 + \sum_{i=0}^{n-1} 2^i$)

Q2. (a) As we discussed in the lecture, floating point arithmetic is not associative, so that $(A + B) + C \neq A + (B + C)$. Demonstrate this using A=1234.565, B=45.68044, C=0.0003 (you may assume a floating-point system that carries 7 significant figures in decimal, and ensure you use appropriate rounding at each step of the calculations). Also test whether these particular values are distributive or not, *i.e.* if $(A + B) \times C = A \times C + B \times C$ and again show all the steps.

(b) What is the largest *integer* that can be represented exactly in the IEEE 754 floating-point standard at single precision? What about at double precision?

To evaluate the binary representation of a floating-point number $x = (-1)^s \times 2^e \times m$, you need to evaluate the sign bit, s, the value of exponent, e, and the mantissa m. Remember the exponent is biased by a number that depends upon the precision of the representation. You can find the exponent by successively testing values of e until $x/2^e = 1.yyy$ (so that the mantissa is described by a number $1.yyy$). The *stored* mantissa will then be the binary value of $0.yyy$, as we do not store the leading digit in a normalized system (it must be 1). To evaluate the binary representation we then just look for a sum of powers of 2^{-1} that are as close to the decimal representation as possible.

(c) Work out the IEEE 754 single precision representation of -0.140625_{10}. Give both the big endian and little endian representations.

Q3. Suppose we have a list of numbers that are passed into the the in-tray of the Little Man Computer. We want to add up and print the sum of all the numbers that pass through *until a zero is placed in the in-tray.* To make things slightly simpler, the instruction set has been extended to include a *conditional branch* instruction, BRZ, which will jump to a line of the code (which you specify) when a 0 value is in the calculator. The machine code representation for BRZ 22 (branch to line 22 on a zero in the calculator) would be 822. Write a program in LMC assembly language that performs this sum and outputs the value at the end. Also give the machine code representation. You can assume that all memory references are initially set to zero. (HINT: you will also need to use the JUMP command and to store the running total in memory somewhere. If you are having problems draw out a flow chart of what exactly needs to be done, you really shouldn't need more than ten lines of code).

Q4. (a) What type of line would match

`grep '^cheat$'`

(b) What type of line would match

`grep '\^s'`

(c) What type of line would match

`grep '^.$'`

(d) Which of the following expressions matches the regular expression search pattern

`grep -E '\^A(f|a)t\$'`

(i) faAFa (ii) Aata (iii) tafA (iv) Aft (v) Aat