CS 270 PA 5: More Assembly Programming

due Tuesday Nov 12, 5:00 pm

The assignment is to be done individually. The main goal of this programming assignment is to advance the skills of designing complex assembly language programs, and to master subroutines, get used to saving and restoring registers, and start understanding parameter passing.

This assignment builds on PA4. You need to write a program to add two 16-bit floating point numbers. You will use the algorithm that you saw in Quiz 1, stepping through the different conditions that you encountered in the three sample inputs (note that the quiz did not exhaustively cover all possibilities). Because you are forced to use the LC-3 instruction set, you will find the some very simple operations available in C require "creative thinking" to program in the LC-3. there is ample scope for extra credit, and we urge you to think creatively.

The problems below don't have to be done in the order given below, but we suggest that you at least make the plan and some progress on Problem 1 before moving to the others. In fact, early submission of one or more of the other programs will earn you extra credit.

Problem PA5.1: LSHbyK Write an LC-3 assembly function that left-shifts an input value by an integer, k, that is also provided as an input. The value of k is guaranteed to be a "small enough" integer. At each shift, the least significant bit is set to zero.

Problem PA5.2: RSHbyK Write an LC-3 assembly function that right-shifts an input value by an integer, k, that is also provided as an input. The value of k is guaranteed to be a "small enough" integer. At each shift, the most significant bit is set to zero.

Problem PA5.3: Lmost1 Write an LC-3 assembly function that determines the position of the leftmost 1 bit in a given 16-bit input. The program, like the two previous has two parameters, INPUT and K, but this time, K is an *output* parameter. When the subroutine returns successfully, K will be a "small enough" integer.

Problem PA5.4: FLadd Develop the main program to add two 16-bit floating point numbers. It has two inputs X and Y, and one output variable SUM.

Strategy

For each program, start with the template that is provided in the file:
`http://www.cs.colostate.edu/~cs270/.Fall13/Assignments/PA5/PA5.asm`

In PA4, you already submitted one of the three (sub) problems PA5.1...PA5.3, but we are not telling you which (it is a question on PA4).

Recall Quiz1. To solve Problem PA5.4, you will need to do a number of steps (extract the signs, the exponents and the fractional parts of the two arguments). Then you need to do some tests. For example, you will need to compare the signs, compare the exponents, and compare the fractional parts, leading to eight possible outcomes (or more if you need to worry about equalities). Managing these different cases is relatively easy in a high level programming language, but in assembly, you need to carefully *plan the layout* of your program.

A good strategy is as follows. First, develop a flowchart of the top-level control, until you make a final decision on the exact set of operations that you need to execute to produce the answer.

Write an LC-3 assembly program that simply executes the flow-chart correctly for a range of inputs, and instead of producing the correct answer, print out a sting (e,g,. "add frac-A to shifted-frac-B") so that you know that at least that the "control flow logic" is right. Test this out on a number of cases (the ones in Quiz1 form a good starting point, but they did not cover all possibilites.

Now think of what pieces you need for each part—it will be Problems PA5.1...PA5.3, but you may want to define additonal subroutines to make your program modular.

Grading & Extra Credit

This assignment has ample opportunity for extra credit, but the basic point distribution is as follows (only two of the first three will count–no double counting for work already done):
PA5.1.asm: 20
PA5.2.asm: 20
PA5.3.asm: 20
PA5.4.asm: 50
Coding Style, Clarity, Following Instructions: 10 pts

Extra Credit There is ample opportunity for extra credit.

- If you think of a clever, alternate way to do any of the parts or the main program, please describe it in a README file and explain in detail. Please also explain why you think it is better or elegant.

- For example, it is possible to do the main assignment using only two of the three Problems PA5.1...PA5.3. So, if you are able to deduce this, then you will not only be able to get extra credit, but also, be able to reduce the amount of work that you have to do–double benefit!

- For Problem PA5.1, there are many alternatives, such as using a sequence of "rotate" operations as we saw in the midterm, or by implementing a division through repeated subtraction, etc.
- If you submit one of three Problems PA5.1...PA5.3 (of course, the one one that is *not* PA4) by the PA4 deadline, you will receive extra credit.
- And finally, I have seen that students come up with many elegant solutions that we would never have imagined, and so this is your opportunity to educate us.

Submission Instructions Create a directory called PA5 to store the assembly files. Submit a file, PA5.tar.gz, generated from the following command (assuming you are running this command from inside PA5 directory):

```
cd ..; tar -czvf PA5.tar.gz PA5
```

Do not submit the assignment with lc3tools inside PA5. There must be exactly two files in your PA5, namely, PA5.asm and README. The name of the README file should be exactly uppper case letters "README" without any file extension. PLEASE DO NOT SUBMIT YOUR .obj or .sym files.

Coding style includes the proper commenting of code.