Laboratory Exercise 8

Subroutines in ARM assembly

Revision of November 17, 2021

The purpose of this lab is to:

1. learn how to write basic programs in assembly language
2. learn how to use loops and subroutines in assembly language

1 Preparation

Please refer to the Lab 7 handout for resources related to the ARM processor and the cpulator simulator. A portion of the mark for this lab will be allocated to questions posed by the TAs in your lab session.

2 Part I

In Lab 7, you were given sample code in Part 2 to review and understand. Now you will be modifying that code to use subroutines.

Specifically, perform the following:

1. Make a copy of the file part2.s from Lab 7 and call it part1.s.
2. Take the code which calculates the number of consecutive 1s and make it into a subroutine called ONES. Have the subroutine use register R1 to receive the input data and register R0 for returning the result. At the top of your file, make the subroutine global .global ONES. This will allow the automarker to call your subroutine from another file.
3. Add more words in memory starting from the label TESTNUM. You can add as many words as you like—but include at least 10 words. Terminate the list with a -1.
4. In your main program (in a separate file), call the newly-created subroutine in a loop for every word of data that you placed in memory. Keep track of the longest string of 1s in any of the words, and have this result in register R5 when your program completes execution. **Note: The cpulator simulator requires all your code to**

be in one file. For testing and debugging with the simulator, please put all your code in one file and then copy paste the subroutine to a separate file for submission. You can have multiple files when running on the DE1-SoC boards in lab.

5. Make sure to use the single-step capability of the simulator to step through your code and observe what happens when your subroutine call and return instructions are executed.

6. For the code you submit from this part, your file should only contain the ONES subroutine. The submitted file should not contain your main program or your TESTNUM list.

3 Part II

Write an assembly language program to sort a list of 32-bit unsigned numbers into ascending order. The first 32-bit number in the list gives the number of items in the list and the remainder of the entries are the numbers to be sorted. The list of data must be sorted "in place", meaning that you are not allowed to create a copy in memory of the list to do the sorting. The list you should sort, including the number count, can be defined using the .word directive at the end of your program as follows:

```
LIST: .word 10, 1400, 45, 23, 5, 3, 8, 17, 4, 20, 33
```

Note that the first number in the list is 10, indicating that there are 10 numbers following it. In your code you should have a main program that loops through the list of numbers as needed to perform the sorting operation. Use **bubble sort** to perform the sorting. Provide pseudo code, similar to C code, that illustrates how your sorting algorithm works. Then, write an assembly language program.

As your main program loops through the list of numbers it will be necessary to compare the values of pairs of list elements to see if they need to be swapped. You are to use a subroutine, called SWAP, to compare such list elements. The SWAP subroutine is passed the address of a list element, compares it to the following element in the list, and swaps the two elements in memory if necessary. The SWAP subroutine should return 1 if a swap is performed, and 0 if not. Pass the address of the first list element to SWAP in register R0, and also pass the return value back to the main program in register R0.

1. Your main program (start:) and your subroutine SWAP should be submitted as one file called part2.s. Please make both start and SWAP global.

2. You should test your code using the CPUlator simulator.
3. Prior to submitting your code, please remove the line starting with “LIST: ”.

4 Submission

Please submit part1.s and part2.s. Be sure to account for any submission guidelines mentioned above.