CS270 Recitation 6
LC-3 Assembler and Simulator

Goals

To understand how to use the LC-3 assembler and simulator, which are parts of the LC-3 Tools package. You will assemble, run, debug, and modify LC-3 machine and assembly code programs for this recitation.

The Assignment

Make a subdirectory called R6 for this recitation assignment, all files should reside in this subdirectory. The tools for this assignment have already been compiled and are located in the ~cs270/lc3tools/ directory[1].

Copy the following LC-3 assembly code into a file called multiply.asm in your R6 directory.

```
               .ORIG     x3000
                BR        MAIN            ; skip over the data
   ;
   ; start of data
   ;
   BEGIN_DATA  .FILL     xBBBB          ;; start of provided variables
   SIX         .FILL     x0006           ; Constant
   NUMBER      .BLKW     1               ; Variable
   RESULT      .BLKW     1               ; Variable
   END_DATA    .FILL     xEEEE          ;; end of provided variables
   ;
   ; start of program
   ;
   MAIN        LD        R1,SIX          ; R1 = 6
               LD        R2,NUMBER       ; R2 = NUMBER
               AND       R3,R3,#0        ; R3 = 0
   ;
   ; inner loop
   ;
   AGAIN       ADD       R3,R3,R2        ; Add NUMBER to R3
               ADD       R1,R1,#-1       ; Decrement loop counter
               BRp       AGAIN           ; Branch is R1 > 0
   ;
               HALT                      ; Program complete
               .END
```

Run the LC-3 assembler to build your assembly file into object (.obj) and symbol (.sym) files, using the following command:

```
   ~cs270/lc3tools/lc3as multiply.asm
```

[1] You can download the tools for your PC at home from here. For linux or Mac, you will need to compile the software, see.

Your directory should now contain multiply.asm, multiply.obj, and multiply.sym. The .obj file is the object file and contains the executable in a binary format. The .sym file contains the symbol table, which describes the memory address of each label.

Inspect the symbol file using a text editor to see the symbol table for the program, shown below. You can see the contents are the labels from the program and their corresponding memory addresses.

```
// Symbol table
// Scope level 0:
//      Symbol Name       Page Address
//      ----------------  ------------
//      BEGIN_DATA        3001
//      SIX               3002
//      NUMBER            3003
//      RESULT            3004
//      END_DATA          3005
//      MAIN              3006
//      AGAIN             3009
```

Part 1: Running and debugging assembled Code

Start the LC-3 simulator, used to run and debug the assembly program, using the command:

```
~cs270/lc3tools/lc3sim-tk &
```

Load the assembly program by selecting the browse button (bottom right) and opening multiply.obj. For the tasks below, you will be modifying multiply.asm, which will be shown to your TA. Copy and answer the short questions below in a separate text file, *answers.txt*, in your R6 directory.

DEBUGGING

Put the hex value 0x1111 into NUMBER (memory location 0x3003). To do this, type the memory address and value in the boxes above the LC-3 assembly code and hit Enter. STEP the program until the value in R3 is exactly 0x3333, watch as the register values change with each instruction. What is the value of the loop counter R1 if you stop at the BRP instruction? What is the condition code (CC) at this time? When will the condition code be different?

FIRST MODIFICATION

What would you change in the program to multiply by 5 instead of 6? Should you change the label as well? Make the change, reassemble the program, RESET the simulator, and make sure it works. Don’t forget to load the initial value into NUMBER again.

SECOND MODIFICATION

What would you change in the program to compute ((NUMBER-2) * 5)? Make the change, reassemble the program, RESET the simulator, and make sure it works. Don't forget to load the initial value into NUMBER again.

THIRD MODIFICATION

Is the value of NUMBER in memory changed by this program? If not, how can you modify the program to store the number computed back into the memory location labeled RESULT? Make the change, reassemble the program, RESET the simulator, and make sure it works. Don't forget to load the initial value into NUMBER again.

Part 2: Debugging Machine Code

Copy the following hexadecimal machine code into a file named example.hex:

```
3000 ; start program at location x3000
2206 ; load location 3007 into R1
2406 ; load location 3008 into R2
56E0 ; AND R3,R3,0
16C1 ; ADD R3,R3,R1
16C2 ; ADD R3,R3,R2
3603 ; store R3 into location 3009
F025 ; halt
1111 ; location 3007 = 0x1111 (constant)
0006 ; location 3008 = 0x0006 (constant)
0000 ; location 3009 = result (variable)
```

Convert the program to an object file as follows:

```
~cs270/lc3tools/lc3convert -b16 example.hex
```

The following should be answered in the answers.txt file used in Part 1. Examine the machine code above, and note that comments have been included to explain the program. Without using the simulator, try to figure out what the result will be by writing down the contents of registers R1, R2, and R3 after each instruction. Next, step through each instruction using the simulator and see what happens. Did the program do what you expected?

Conclusion

Show the TA your modified multiply.asm and answers.txt files. Finally, describe any challenges you faced during this assignment.