

CS270 Recitation 8

“LC-3 More Programming”

Goals

1. To extend your knowledge of LC-3 programming with a more complex algorithm.
2. To solidify your knowledge of how to use the LC-3 assembler and simulator to debug assembly code.

The Assignment

Make a subdirectory called R8 for the recitation, all files should reside in this subdirectory. Copy the file from the link to the R8 directory, a listing of the code with some comments removed is shown below.

<http://www.cs.colostate.edu/~cs270/.Fall14/recitations/R8/R8.asm>

```
; Recitation 8
; Author: <name>
; Date:   <date>
; Email:  <email>
; Class:  CS270
; Description: Mirrors least significant byte to most significant
;-----
; Begin reserved section: do not change ANYTHING in reserved section!

                .ORIG x3000

                JSR mirror          ; call function
                HALT

; Parameter and return value
Param           .BLKW 1            ; space to specify parameter
Result          .BLKW 1            ; space to store result

; Constants
One             .FILL #1           ; the number 1
Eight           .FILL #8           ; the number 8
Mask            .FILL x00ff        ; mask for most significant byte

; End reserved section: do not change ANYTHING in reserved section!
;-----
mirror          ; Mirrors bits 7:0 to 15:8
                ; ~20 lines of assembly code

                LD R0,Param          ; load pattern
                ; your code here
                ST R1,Result         ; store result
                RET

;-----
                .END
```

1) Use the LC-3 assembler to transform your assembly code into object code that can run on the LC-3 simulator:

```
$ ~cs270/lc3tools/lc3as R8.asm
```

2) Load the LC-3 simulator and the TA will help you step through an invocation of one of the LC-3 subroutines:

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

3) Implement the **mirror** subroutine, using the following algorithm:

- Loads the parameter into register 0
- Makes a copy of register 0 to register 1, which stores the result
- Load the mask into register 2
- Use the mask to clear bits 15:8 in the result
- Initialize register 2 to 1, this is the source mask
- Initialize register 3 to 1, this is the destination mask
- Initialize register 4 to 8, this is the counter
- Write a loop to shift register 3 left by eight bits, by adding it to itself
- Initialize register 4 to 8, again
- Write a loop that
 1. checks if the bit in the source mask is set in the parameter
 2. if so, adds the bit in the destination mask to the result
 3. then shifts the source mask left by adding it to itself
 4. then shifts the destination mask left by adding it to itself
 5. and finally decrements the counter until it reaches zero
- Store the result into the Result memory location and return

4) Test the **mirror** subroutine in the simulator using Param = 0x1234. The answer in Result should be 0x3434.

5) Submit to the drop box in RamCT for Recitation 8 and show your code to the TA.