

## **Student Name:**

### **CMSC 311 Computer Organization**

#### **LAB 5: Assembly Language programming Using LC-3 Simulator**

**Instructor:** Dr. Cang Ye

**Email:** [cye@vcu.edu](mailto:cye@vcu.edu)

**Office & Hours:** E2240, TR 12-1 PM

**TA:** Lingqiu Jin

**Email:** [jinl@mymail.vcu.edu](mailto:jinl@mymail.vcu.edu)

**Office & Hours:** E2264, MW 2-3 PM

## **Assignment:**

Write an LC-3 assembly language program that searches for even numbers in memory locations ranging from address A to address B (inclusive). Set R0 to the number of even numbers that are found. Your solution should use an iterative construct like the examples in the lecture. To test the program, let A to be the first available memory location after your code and let  $B=A+9$ .

## **Instructions:**

1. Your lab report is due no later than the date of the next class. Your lab report (in pdf format) should include your assembly code and a screenshot. Submit both your lab report and source codes in separate files via blackboard. Please comment your code to make it easy to understand.
2. The screenshot should show both your assembly code and simulation result. The code should start at address x3000. All register information, the assembly code, and the input data should be visible on the screenshot.
3. Grading policy: lab report 20%, source code 80%.

## **Source code**

```
.ORIG X3000           ;start the program at x3000
AND R0,R0,#0          ;clear R0 to store the result
LD R1, #-10           ;load r1 with the first address location
LD R4, #-11           ; load r4 with the first address
LD R5, #-11           ;load r5 with the first address
NOT R4,R4             ; change sign of r4
ADD R4, R4, #2         ;add r2 to r4 to calculate the loop
ADD R2, R4, R5         ;add r4 and r5 to calculate the loop
BRZ #7                ;if the R2 become 0 we more to HALT
LDR R3,R1,#0          ;load R3 with number in R1 address
```

AND R3,R3,#1 ; R3 = 0 if even, 1 if odd  
 BRnp #1 ;if odd, skip counter increment  
 ADD R0,R0,#1 ;even number increase counter  
 ADD R1,R1,#1 ;increase data pointer  
 ADD R2,R2,#-1 ;decrease loop count  
 BRNZP #-8 ;unconditional branch to start of loop  
 HALT  
 .END

In this program. I the counter will be stored in R2. Before running program. We will have to specify the start and end location addresses of the locations that hold the numbers. At x2FF8 for the beginning and x2FF9 for the last number. Also in this program, zeros will be treated as even number.

The screenshot displays the LC-3 Simulator interface in a web browser. The interface is divided into several sections:

- Memory:** A table showing memory addresses from x3000 to x300F. The instructions are:
 

0x	Label	Hex	Instruction
x3000		x5020	AND R0, R0, #0
x3001		x23F6	LD R1, x2FF8
x3002		x29F5	LD R4, x2FF8
x3003		x2BF5	LD R5, x2FF9
x3004		x993F	NOT R4, R4
x3005		x1922	ADD R4, R4, #2
x3006		x1505	ADD R2, R4, R5
x3007		x0407	BRz x300F
x3008		x6640	LDR R3, R1, #0
x3009		x56B1	AND R3, R3, #1
x300A		x0A01	BRnp x300C
x300B		x1021	ADD R0, R0, #1
x300C		x1261	ADD R1, R1, #1
x300D		x14BF	ADD R2, R2, #-1
x300E		x0FF8	BRnzp x3007
x300F		xF025	HALT
- Status:** Shows register values: R0: x0005, R1: x301A, R2: x0000, R3: x0000, R4: xCFF1, R5: x3019, R6: x0000, R7: x0000, PC: x300F, IR: x0407, PSR: x8002, CC: Z. It also includes buttons for 'Clear R0-R7' and 'Reset all registers'.
- Console:** A large text area for output, currently empty. It has buttons for 'Clear Input Buffer (0 characters)' and 'Clear Output'.
- Navigation:** Buttons for 'Step', 'Next', 'Finish', 'Run', 'Pause', 'Continue', and 'Unhalt'. A checkbox for 'Follow PC' is checked.

