Student Name:

CMSC 311 Computer Organization
LAB 6: Assembly Language programming Using LC-3 Simulator

Instructor: Dr. Cang Ye **Email:** cye@vcu.edu Office & Hours: E2240, TR 12-1 PM
TA: Lingqiu Jin **Email:** jinl@mymail.vcu.edu Office & Hours: E2264, MW 2-3 PM

Assignment:

1) Write an LC-3 assembly language program that divides a positive number X by another positive number Y and stores the integer part of the result in R3, i.e., R3 = int (X/Y). Your code should use an iterative construct.

2) Write an assembly program that produce the modulo of 2 positive numbers (i.e., the remainder of the division) and store the result in R4, i.e., R4 = X % Y.

The value of X should be stored in 0x3100 and the value of Y should be stored in 0x3101. Also, let X = 0x005A and Y = 0x0007. Your program should start at 0x3000.

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 screenshot(s). Submit both your lab report and source codes in separate files via blackboard. Please comment your code to make it easy to understand.

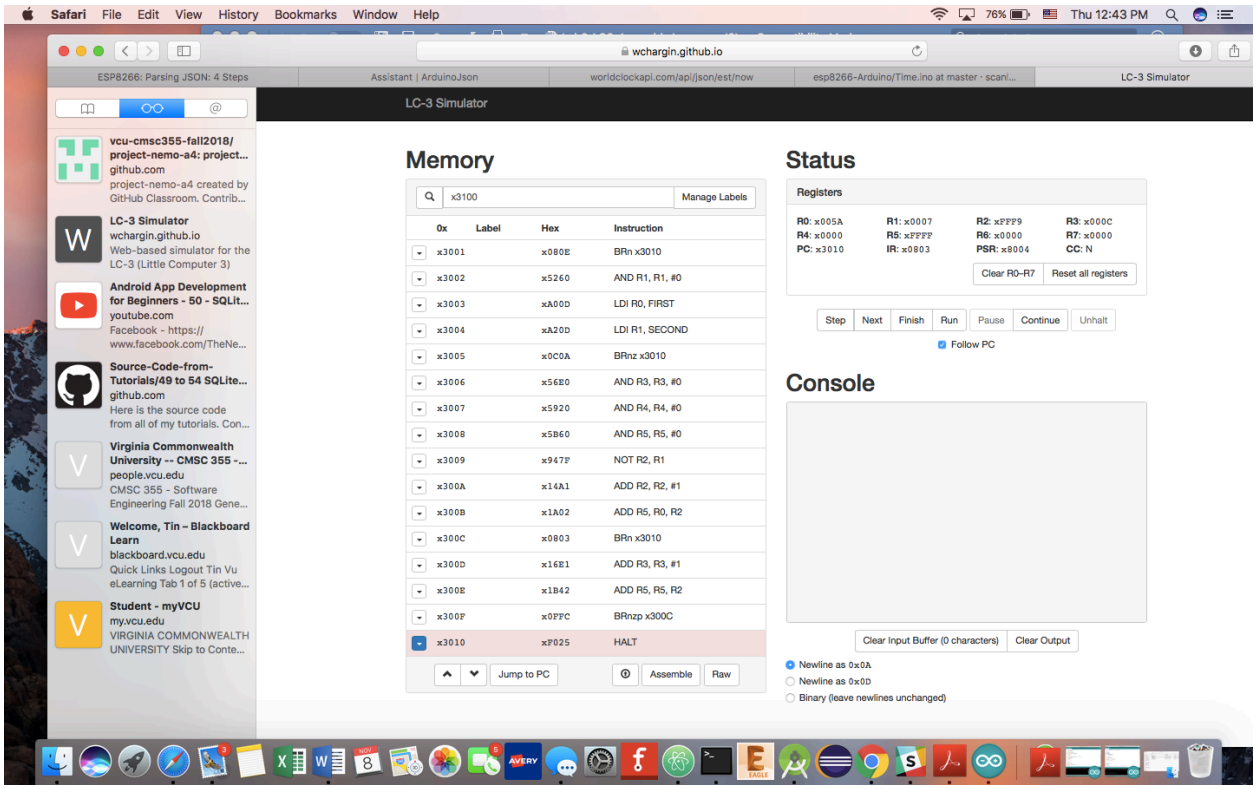
2. The screenshot(s) should show both your assembly code and simulation result. The code should start at address x3000. All register information, the assembly code, and the input data should be visible on the screenshot(s).

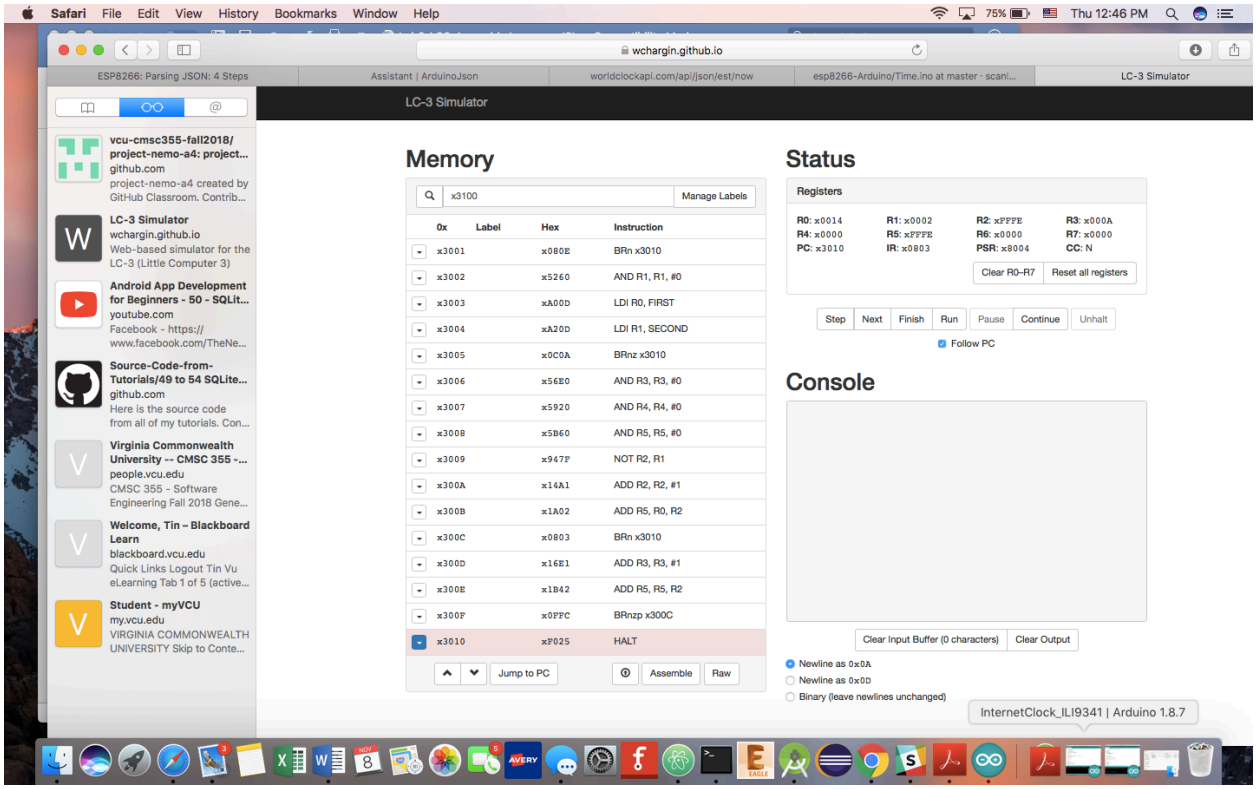
3. Grading policy: lab report 20%, source code 80%.

1) Write an LC-3 assembly language program that divides a positive number X by another positive number Y and stores the integer part of the result in R3, i.e., R3 = int (X/Y). Your code should use an iterative construct.
    a. In this program I used the subtraction method to find the division result of X and Y. First the program check if the bottom number is zero. Conditions are set to have no negative number as input in this program. The result will be stored in R3. We should see the top number and bottom number at R0 and R1 respectively at the end of each run also. Before running, we have to input the top and. Due to the available space of the screen! The first ADD instruction isn't shown.

CODE:

```
.ORIG x3000
AND R0,R0, #0 ;; R0, WILL HOLD THE FIRST NUMBER
BRn, #14      ;; NO NEGATIVE INTEGER IN THIS PROGRAM
AND R1,R1, #0 ;; R1, WILL HOLD THE SECOND NUMBER
LDI R0, FIRST
LDI R1, SECOND
BRnz, #10       ;; CAN'T DIVISE BY ZERO AND ALSO NO NEGATIVE INTEGER IN THIS PROGRAM
AND R3,R3, #0 ;; R3 WILL HOLD THE RESULT
AND R4,R4, #0 ;; R4, WILL HOLD THE REMAINING
AND R5, R5, #0 ;; HOLD THE TEMP RESULT
NOT R2, R1    ;; CHANGE SIGN OF THE SECOND NUMBER
ADD R2, R2, #1 ;;
ADD R5, R0, R2 ;; FIRST NUMBER - SECOND NUMBER
BRn, #3       ;;IF THE RESULT IS NEGATIVE--> HALT
;;ADD R0,R5,#0   ;; ELSE PUT THE RESULT OF R3 BACK INTO R0
ADD R3, R3, #1 ;; INCREASE THE COUNTER
ADD R5, R5,R2 ;; FIRST - SECOND AGAIN
BRnzp, #-4     ;; UNCONDITIONAL BRANCH
HALT
FIRST .FILL x3100
SECOND .FILL x3101
.END
```

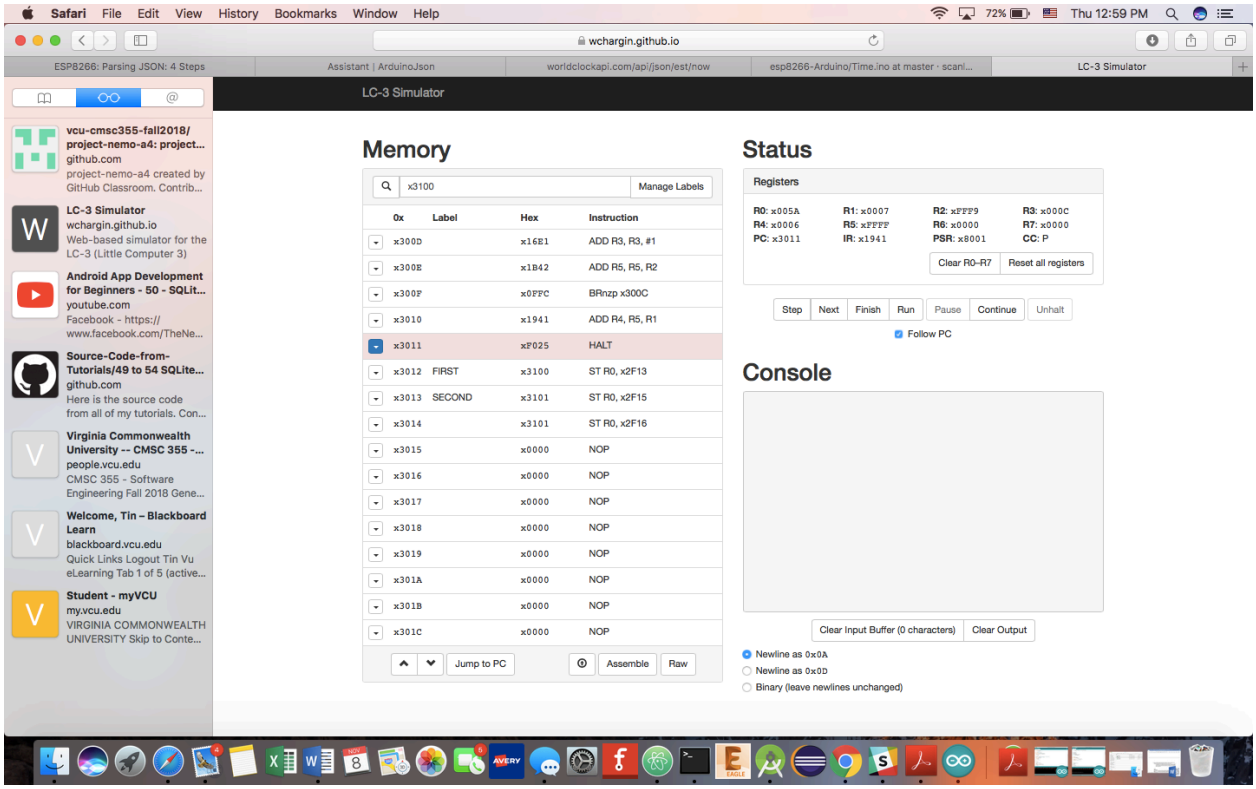

Safari File Edit View History Bookmarks Window Help
Thu 12:43 PM
wchargin.github.io
ESP8266: Parsing JSON: 4 Steps
Assistant | ArduinoJson
worldclockapi.com/api/json/est/now
esp8266-Arduino/Time.ino at master · scanl...
LC-3 Simulator
LC-3 Simulator
vcu-cmsc355-fall2018/project-nemo-a4: project...
github.com
project-nemo-a4 created by GitHub Classroom. Contrib...
LC-3 Simulator
wchargin.github.io
Web-based simulator for the LC-3 (Little Computer 3)
Android App Development for Beginners - 50 - SQLit...
youtube.com
Facebook - https://www.facebook.com/TheNe...
Source-Code-from-Tutorials/49 to 54 SQLite...
github.com
Here is the source code from all of my tutorials. Con...
Virginia Commonwealth University -- CMSC 355 -...
people.vcu.edu
CMSC 355 - Software Engineering Fall 2018 Gene...
Welcome, Tin - Blackboard Learn
blackboard.vcu.edu
Quick Links Logout Tin Vu eLearning Tab 1 of 5 (active...
Student - myVCU
my.vcu.edu
VIRGINIA COMMONWEALTH UNIVERSITY Skip to Conte...
Memory
x3100
Manage Labels
0x Label Hex Instruction
x3001 x080E BRn x3010
x3002 x5260 AND R1, R1, #0
x3003 xA00D LDI R0, FIRST
x3004 xA20D LDI R1, SECOND
x3005 x0C0A BRnz x3010
x3006 x56E0 AND R3, R3, #0
x3007 x5920 AND R4, R4, #0
x3008 x5B60 AND R5, R5, #0
x3009 x947F NOT R2, R1
x300A x14A1 ADD R2, R2, #1
x300B x1A02 ADD R5, R0, R2
x300C x0803 BRn x3010
x300D x16E1 ADD R3, R3, #1
x300E x1B42 ADD R5, R5, R2
x300F x0FFC BRnzp x300C
x3010 xF025 HALT
Jump to PC
Assemble
Raw
Status
Registers
R0: x005A R1: x0007 R2: xFFF9 R3: x000C
R4: x0000 R5: xFFFF R6: x0000 R7: x0000
PC: x3010 IR: x0803 PSR: x8004 CC: N
Clear R0-R7
Reset all registers
Step
Next
Finish
Run
Pause
Continue
Unhalt
Follow PC
Console
Clear Input Buffer (0 characters)
Clear Output
Newline as 0x0A
Newline as 0x0D
Binary (leave newlines unchanged)


Safari File Edit View History Bookmarks Window Help
Thu 12:46 PM
wchargin.github.io
ESP8266: Parsing JSON: 4 Steps
Assistant | ArduinoJson
worldclockapi.com/api/json/est/now
esp8266-Arduino/Time.ino at master · scanl...
LC-3 Simulator
LC-3 Simulator
vcu-cmsc355-fall2018/project-nemo-a4: project...
github.com
project-nemo-a4 created by GitHub Classroom. Contrib...
LC-3 Simulator
wchargin.github.io
Web-based simulator for the LC-3 (Little Computer 3)
Android App Development for Beginners - 50 - SQLit...
youtube.com
Facebook - https://www.facebook.com/TheNe...
Source-Code-from-Tutorials/49 to 54 SQLite...
github.com
Here is the source code from all of my tutorials. Con...
Virginia Commonwealth University -- CMSC 355 -...
people.vcu.edu
CMSC 355 - Software Engineering Fall 2018 Gene...
Welcome, Tin - Blackboard Learn
blackboard.vcu.edu
Quick Links Logout Tin Vu eLearning Tab 1 of 5 (active...
Student - myVCU
my.vcu.edu
VIRGINIA COMMONWEALTH UNIVERSITY Skip to Conte...
Memory
x3100
Manage Labels
0x Label Hex Instruction
x3001 x080E BRn x3010
x3002 x5260 AND R1, R1, #0
x3003 xA00D LDI R0, FIRST
x3004 xA20D LDI R1, SECOND
x3005 x0C0A BRnz x3010
x3006 x56E0 AND R3, R3, #0
x3007 x5920 AND R4, R4, #0
x3008 x5B60 AND R5, R5, #0
x3009 x947F NOT R2, R1
x300A x14A1 ADD R2, R2, #1
x300B x1A02 ADD R5, R0, R2
x300C x0803 BRn x3010
x300D x16E1 ADD R3, R3, #1
x300E x1B42 ADD R5, R5, R2
x300F x0FFC BRnzp x300C
x3010 xF025 HALT
Jump to PC
Assemble
Raw
Status
Registers
R0: x0014 R1: x0002 R2: xFFFE R3: x000A
R4: x0000 R5: xFFFE R6: x0000 R7: x0000
PC: x3010 IR: x0803 PSR: x8004 CC: N
Clear R0-R7
Reset all registers
Step
Next
Finish
Run
Pause
Continue
Unhalt
Follow PC
Console
Clear Input Buffer (0 characters)
Clear Output
Newline as 0x0A
Newline as 0x0D
Binary (leave newlines unchanged)
InternetClock_ILI9341 | Arduino 1.8.7

1) Write an assembly program that produce the modulo of 2 positive numbers (i.e., the remainder of the division) and store the result in R4, i.e., R4 = X % Y.
    a. This program is just a modified version of the first program in this lab. I added 2 lines after the loop to calculate the remainder of X and Y. all the preconditions and postconditions stay the same. Input at the same addresses. The remainder is stored at register R4.

CODE:

```
.ORIG x3000
AND R0,R0, #0 ;; R0, WILL HOLD THE FIRST NUMBER
BRn, #14      ;; NO NEGATIVE INTEGER IN THIS PROGRAM
AND R1,R1, #0 ;; R1, WILL HOLD THE SECOND NUMBER
LDI R0, FIRST
LDI R1, SECOND
BRnz, #10       ;; CAN'T DIVISE BY ZERO AND ALSO NO NEGATIVE INTEGER IN
THIS PROGRAM
AND R3,R3, #0 ;; R3 WILL HOLD THE RESULT
AND R4,R4, #0 ;; R4, WILL HOLD THE REMAINING
AND R5, R5, #0 ;; HOLD THE TEMP RESULT
NOT R2, R1    ;; CHANGE SIGN OF THE SECOND NUMBER
ADD R2, R2, #1 ;;
ADD R5, R0, R2 ;; FIRST NUMBER - SECOND NUMBER
BRn, #3       ;;IF THE RESULT IS NEGATIVE--> HALT
;;ADD R0,R5,#0   ;; ELSE PUT THE RESULT OF R3 BACK INTO R0
ADD R3, R3, #1 ;; INCREASE THE COUNTER
ADD R5, R5,R2 ;; FIRST - SECOND AGAIN
BRnzp, #-4     ;; UNCONDITIONAL BRANCH
ADD R4, R5, R1 ;; STORE THE REMAINDER
HALT
FIRST .FILL x3100
SECOND .FILL x3101
.END
```



SAMPLE RUN WITH X = 90 AND Y = 7

RESULT IS IN R3: 12 (x00C)

REMAINDER IS IN R4: 6 (x006)