CS 2110 Quiz 4

TOTAL POINTS

95 / 100

QUESTION 1

True or False 10 pts

1.1 BR vs. JMP **5 / 5**

✓ + **0 pts** Graded

✓ + **5 pts** Correct: False

1.2 TRAP x27 **5 / 5**

✓ + **0 pts** Graded

✓ + **5 pts** Correct: False

QUESTION 2

Tracing a Program 50 pts

2.1 Instruction 1 **6 / 7**

✓ + **0 pts** Graded

✓ + **1 pts** $$\text{PC}$$: $$3001_{16}$$

+ **1.5 pts** $$\text{R0}$$: $$24576_{10}$$

+ **0.5 pts** $$\text{R0}$$: Expression that evaluates to $$24576_{10}$$

✓ + **0.5 pts** $$\text{R0}$$: $$6000_{16}$$

✓ + **1.5 pts** $$\text{R1}$$: $$3$$

✓ + **1.5 pts** $$\text{R2}$$: $$0$$

✓ + **1.5 pts** $$\text{CC}$$: $$\text{p}$$

2.2 Instruction 2 **6 / 7**

✓ + **0 pts** Graded

✓ + **1 pts** $$\text{PC}$$: $$3002_{16}$$

+ **1.5 pts** $$\text{R0}$$: $$24576_{10}$$

+ **0.5 pts** $$\text{R0}$$: Expression that evaluates to $$24576_{10}$$

✓ + **0.5 pts** $$\text{R0}$$: $$6000_{16}$$

✓ + **1.5 pts** $$\text{R1}$$: $$3$$

✓ + **1.5 pts** $$\text{R2}$$: $$0$$

✓ + **1.5 pts** $$\text{CC}$$: $$\text{p}$$

2.3 Instruction 3 **6 / 7**

✓ + **0 pts** Graded

✓ + **1 pts** $$\text{PC}$$: $$3003_{16}$$

+ **1.5 pts** $$\text{R0}$$: $$24576_{10}$$

+ **0.5 pts** $$\text{R0}$$: Expression that evaluates to $$24576_{10}$$

✓ + **0.5 pts** $$\text{R0}$$: $$6000_{16}$$

✓ + **1.5 pts** $$\text{R1}$$: $$3$$

✓ + **1.5 pts** $$\text{R2}$$: $$0$$

✓ + **1.5 pts** $$\text{CC}$$: $$\text{p}$$

2.4 Instruction 4 **6 / 7**

+ **0 pts** Graded

✓ + **1 pts** $$\text{PC}$$: $$3004_{16}$$

+ **1.5 pts** $$\text{R0}$$: $$24577_{10}$$

+ **0.5 pts** $$\text{R0}$$: Expression that evaluates to $$24577_{10}$$

✓ + **0.5 pts** $$\text{R0}$$: $$6001_{16}$$

✓ + **1.5 pts** $$\text{R1}$$: $$3$$

✓ + **1.5 pts** $$\text{R2}$$: $$0$$

✓ + **1.5 pts** $$\text{CC}$$: $$\text{p}$$

2.5 Instruction 5 **6 / 7**

+ **0 pts** Graded

✓ + **1 pts** $$\text{PC}$$: $$3005_{16}$$

+ **1.5 pts** $$\text{R0}$$: $$24577_{10}$$

+ **0.5 pts** $$\text{R0}$$: Expression that evaluates to $$24577_{10}$$

✓ + **0.5 pts** $$\text{R0}$$: $$6001_{16}$$

✓ + **1.5 pts** $$\text{R1}$$: $$6$$

✓ + **1.5 pts** $$\text{R2}$$: $$0$$

✓ + **1.5 pts** $$\text{CC}$$: $$\text{p}$$

2.6 mem[x6000] **3.5 / 3.5**

✓ + **0 pts** Graded

✓ + **3.5 pts** Correct: $$3$$

+ **1.75 pts** Used .fill

2.7 mem[x6001] **3.5 / 3.5**

+ **0 pts** Graded

✓ + **3.5 pts** Correct: $$10_{10}$$ or $$A_{16}$$

+ **1.75 pts** Used .fill

2.8 mem[x6002] 3.5 / 3.5

+ **0 pts** Graded

✓ + **3.5 pts** **Correct: $$0$$**

+ **1.75 pts** Used .fill

2.9 mem[x7000] 4.5 / 4.5

+ **0 pts** Graded

✓ + **4.5 pts** **Correct: $$52_{10}$$ or $$34_{16}$$**

+ **2.25 pts** Used .fill

QUESTION 3

3 Tracing a Program: Input / Output 15 / 15

✓ + **0 pts** **Graded**

✓ + **15 pts** **Correct: $$\text{``135917"}$$**

+ **7.5 pts** Partial: Outputs only odd numbers

QUESTION 4

4 Pseudocode to Assembly 25 / 25

+ **0 pts** Graded

✓ + **7 pts** **Everything works correctly**

Note: Only check this if other criteria (except syntax errors) are all satisfied

✓ + **5 pts** **Correctly checks $$\text{R0 > 0}$$**

✓ + **5 pts** **Correctly checks $$\text{R1 < 0}$$**

✓ + **3 pts** **Accumulates the sum of $$\text{R0}$$ and $$\text{R1}$$ and $$\text{R2}$$ in $$\text{R2}$$**

✓ + **2.5 pts** **Decrements $$\text{R0}$$**

✓ + **2.5 pts** **Increments $$\text{R1}$$**

- **5 pts** Small syntax errors

- **10 pts** Significant syntax errors

CS 2110 — Quiz 4 (30 minutes)
October 10th, 2018

Full name: _

GT username: _

This quiz is worth a total of **100 points**.

You are allowed to use one sheet of scrap paper. Feel free to request scrap paper from your Teaching Assistants. **Please make sure that all of your answers are contained within the answer boxes or the fill-in lines.** Do not write your work in the answer boxes, **keep all of your work on your scrap paper.** You will **NOT** be given credit for just showing work. Having anything except the answer inside the boxes or above the fill-in lines reduces autograder performance and might cause incorrect results. **Make sure to write your name, username, and answers legibly. You will not receive credit for illegible answers.**

True or False

1. Please fill in the appropriate circle given the statement. No explanation is required.

   (a) The `BR` and `JMP` instructions use the same addressing mode. ○ True ● False 5

   (b) The instruction `TRAP x27` loads the PC with the address `x0027`. ○ True ● False 5

Tracing a Program

2. Fill in the entirety of **both tables** on the right side. For the first table, fill in the first 5 instructions executed. If HALT is reached, fill in "HALT" for the PC in the corresponding line. Do not fill in any rows after HALT. For the second table, fill it in as if the program has **completely executed**. 50

   For example, for the first table, after the first instruction is executed, record the contents of the registers in row 1.

   Note: Write the `PC` in hexadecimal and `R0`, `R1`, and `R2` in decimal. On rows you use, fill in all boxes regardless of whether or not their values are changed.

Label	Address	Instruction
	x3000	LD R0, START
LOOP	x3001	LDR R1, R0, #0
	x3002	BRZ DONE
	x3003	ADD R0, R0, #1
	x3004	ADD R1, R1, R1
	x3005	ADD R1, R1, R1
	x3006	LDI R2, ANSWER
	x3007	ADD R2, R1, R2
	x3008	STI R2, ANSWER
	x3009	BR LOOP
DONE	x300A	HALT
START	x300B	.fill x6000
ANSWER	x300C	.fill x7000
	...	...
	x6000	.fill #3
	x6001	.fill #10
	x6002	.fill #0
	...	...
	x7000	.fill #0

Instr. #	PC	R0	R1	R2	CC
Initial	x3000	0	3	0	z
1	x3001	x6000	3	Ø	p
2	x3002	x6000	3	Ø	p
3	x3003	x6000	3	Ø	p
4	x3004	x6001	3	Ø	p
5	x3005	x6001	6	Ø	p

Once the program has finished executing, what values are at the following memory addresses?

Address	Contents
x6000	#3
x6001	#10
x6002	#Ø
...	...
x7000	#52

Tracing a Program: Input / Output

3. Consider the LC-3 assembly program in the left column. The contents of memory starting at address x5000 are also provided in the middle column. Assuming the program begins executing at x3000 and continues until HALT, please indicate the final console output. Note that the ASCII value for character '0' (i.e. zero) is 48. 15

Note: OUT sends the character represented by the ASCII code contained in R0 to the console.

```
.orig x3000
      LD R1, OFFSET
      LD R2, ARRAY
      LD R3, LENGTH
LOOP  BRz DONE
      ADD R4, R2, R3
      LDR R4, R4, #-1
      AND R5, R4, #1
      BRz DECR
      ADD R0, R1, R4
      OUT
DECR  ADD R3, R3, #-1
      BR LOOP
DONE  HALT

ARRAY  .fill x5000
LENGTH .fill 10
OFFSET .fill 48 ; this is the character code for ASCII '0'
.end
```

```
.orig x5000
  .fill 2  0
  .fill 7  1
  .fill 6  2
  .fill 4  3
  .fill 1  4
  .fill 9  5
  .fill 5  6
  .fill 4  7
  .fill 3  8
  .fill 1  9
.end
```

Console output:

135917

Pseudocode to Assembly

4. Consider the following incomplete LC-3 program. There is a positive number in register R0 and a negative number in R1. Register R2 has been initialized to zero. Fill in the blank space with valid assembly code such that the completed program, when run, will accumulate the sum of R0 and R1 in R2, decrement R0, and increment R1. Feel free to use both columns for your code. 25

```
.orig x3000
    ; Assume the following: R0 > 0, R1 < 0,
    ;    and R2 = 0.
    ; Convert the pseudocode below to
    ;    assembly:
    ; while (R0 > 0 || R1 < 0) {
    ;     R2 = R2 + R1 + R0
    ;     R0 = R0 - 1
    ;     R1 = R1 + 1
    ; }

    AND R4,R4,Ø
```

```
LOOP
ADD R4, R1, RØ       ; R1 + RØ
ADD R2, R2, R4       ; R2 = R2 + R1 + RØ
ADD RØ, RØ, #-1      ; RØ = RØ - 1
ADD R1, R1, #1       ; R1 = R1 + 1
BRN LOOP
ADD RØ, RØ, Ø
BRP LOOP

      HALT
   .end
```