CS-2110 A/B/C Quiz 4 (B)

[redacted]

TOTAL POINTS

88 / 100

QUESTION 1

1 1a **5 / 5**

✓ **+ 5 pts** *Correct (x4006)*

+ 2 pts off by one (x4005)

+ 0 pts incorrect

QUESTION 2

2 1b **8 / 8**

✓ **+ 8 pts** *Correct (xFFFB)*

+ 5 pts not in hex but correct (-5)

+ 0 pts incorrect

QUESTION 3

3 1c **4 / 8**

+ 8 pts Correct (x000C)

✓ **+ 4 pts** *x9DAC (did not execute the .fill)*

+ 6 pts correct but in binary (00...1100) or decimal (12)

+ 0 pts incorrect

QUESTION 4

4 1d **8 / 8**

✓ **+ 8 pts** *Correct (x0000)*

+ 4 pts did not recognize RA was modified (x0001 or 1)

+ 0 pts incorrect

QUESTION 5

5 1ei **2 / 4**

✓ **+ 2 pts** *Correct (FALSE)*

+ 2 pts Valid Explanation:

- This instruction changes the PC/MAR/MDR

- This instruction could change CC

+ 0 pts incorrect

QUESTION 6

6 1eii **4 / 4**

✓ **+ 2 pts** *Correct (FALSE)*

✓ **+ 2 pts** *Valid Explanation:*

- .blkw or .stringz can take up multiple lines in memory

- .orig & .end take no memory

+ 0 pts incorrect

QUESTION 7

7 1eiii **4 / 4**

✓ **+ 2 pts** *Correct (FALSE)*

✓ **+ 2 pts** *Valid Explanation:*

- JSRR is used because we are unsure how far away the subroutine is in memory

+ 0 pts incorrect

QUESTION 8

8 1eiv **4 / 4**

✓ **+ 2 pts** *Correct (FALSE)*

✓ **+ 2 pts** *Valid Explanation:*

- When popping from the stack, the stack pointer must be incremented

- change second line to ADD R6, R6, 1

+ 0 pts incorrect

QUESTION 9

9 2a **15 / 15**

✓ **+ 15 pts** *Fully Correct*

Example:

![Screenshot_2023-11-08_at_4.28.38_PM.png](/files/bd2a7e7e-3cd3-42e3-bcf2-bba56a535c3b)

+ 1 pts while loop structure

+ 3 pts correct logic to set CC based on a comparison to N

eg.

- decrements N each iteration
- uses register to track iterations and increments each iteration & compares to N

+ 2 pts correct BR condition to end loop after N iterations

+ 2 pts correctly loads value from memory

+ 2 pts correctly doubles value

+ 2 pts correctly stores value back in memory

+ 2 pts increment address

+ 1 pts branch to top of while loop

+ 0 pts incorrect

QUESTION 10

10 2b **15 / 15**

✓ **+ 15 pts** *Fully Correct*

Example:

![Screenshot_2023-11-08_at_4.31.57_PM.png](/files/1a559e19-34e9-4650-b1a3-a85fb471ae1c)

+ 2 pts While loop structure

+ 2 pts correctly loads character from memory into a register

+ 2 pts Branch on correct CC to end (BRz)

- also allow if they end loop after printing null character

+ 1 pts Loop structure for polling the DSR

+ 2 pts correctly loads DSR value

+ 2 pts BRzp to POLL loop

- cannot only by BRz or BRp because the remaining bits are unknown

+ 2 pts correctly stores the character in DDR

+ 1 pts increments address

+ 1 pts branches to top of while loop

+ 0 pts incorrect

- 1 pts Minor errors

QUESTION 11

11 3 **19 / 25**

Part A (addresss)

✓ **+ 8 pts** *addresses correct*

(x3000- x3006, x300A)

+ 6 pts all addresses correct except last one

+ 4 pts all off by one

+ 2 pts all off by one & last address incorrect

+ 0 pts incorrect

Part A (Hexadecimal)

+ 12 pts Fully correct

✓ **+ 2 pts** *1. xB404 (Propagate address error- offset would be different)*

✓ **+ 2 pts** *2. x94BF*

+ 2 pts 3. x07FD (Propagate address error- offset would be different)

✓ **+ 2 pts** *4. x3A02*

+ 2 pts 5. xF025

+ 2 pts 6. x300A (propagate error from address of L_E)

+ 0 pts incorrect

Part B

✓ **+ 5 pts** *fully correct or correct based on part A addresses*

L_A = x3000

L_B = x3001

L_C = x3005

L_D = x3006

L_E = x300A

+ 4 pts 4 correct

+ 3 pts 3 correct

+ 2 pts 2 correct

+ 0 pts incorrect

+ 0 pts incorrect

Your Initials: _____

Name [PRINT CLEARLY]: [redacted]

GT username (e.g. gburdell3): [redacted]

CS 2110: Computer Organization and Programming
Gupta/Conte/Adams Fall 2023

QUIZ 4
VERSION B

This exam is given under the Georgia Tech Honor Code System. Anyone found to have submitted copied work instead of original work will be dealt with in full accordance with Institute policies.

Georgia Tech Honor Pledge: "*I commit to uphold the ideals of honor and integrity by refusing to betray the trust bestowed upon me as a member of the Georgia Tech community.*"

[MUST sign:] Ivan

- THIS IS A CLOSED BOOK, CLOSED NOTES EXAM
- NO CALCULATORS
- This examination handout has **6** pages.
- Do all your work in this examination handout.
- Only the front of exams sheets will be scanned. Do **not** write your answer on the back of the exam sheets.
- Please write your initials at the top of each page
- WHERE NEEDED, SHOW ALL YOUR INTERMEDIATE RESULTS TO RECEIVE FULL CREDIT

In case you forgot, here are some good facts to know:

Hex	Dec
0x1	1
0x2	2
0x3	3
0x4	4
0x5	5
0x6	6
0x7	7
0x8	8
0x9	9
0xA	10
0xB	11
0xC	12
0xD	13
0xE	14
0xF	15

x	2^x
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16,384
15	32,768
16	65,536

Problem	Points	Score
1	45	
2	30	
3	25	
TOTAL	100	

GOOD LUCK!

More good facts to know:

$1K = 2^{10}$
$1M = 2^{20}$
$1G = 2^{30}$
$1T = 2^{40}$
$1P = 2^{50}$
$1E = 2^{60}$

Your Initial [redacted]

1. [45 pts] Answer the following short questions. Show your work (where needed) to receive full credit.

(a) Consider the following fragment of an assembly program:

```
.ORIG x4000
THE_STRING      .STRINGZ   "Conte"
THE_LABEL       .BLKW      1
.END
```

What is the address corresponding to THE_LABEL?

x4006

(b) Consider the following short program and give the value in R3 after the program terminates:

```
.ORIG x3000
AND R3,R3,0
ADD R3,R3,-1
AND R3,R3,-8
ADD R3,R3,3
HALT
.END
```

R3 contains (**in hex**): 0x FFFB

(c) Consider the following short program and give the value in R5 after the program terminates:

```
    .ORIG x3000
    LD R5,X
    LD R1,Y
    ADD R5,R1,R5
X   .FILL x5B6C
    HALT
Y   .FILL x4240
    .END
```

R5 contains (**in hex**): 0x9DAC

(d) Consider the following short program and give the value in R2 after the program terminates:

```
    .ORIG x3000
    AND R2,R2, 0
    NOT R2,R2
    JSR SUB
    NOT R2,R2
    ADD R2,R2,2
    HALT
SUB ADD R7,R7, 2
    ADD R2,R2, 1
    RET
    .END
```

R2 contains (**in hex**): 0x 0000

Your Initials: [redacted]

(e) Answer the following true/false questions by circling "true" or "false," and then give a reason for each answer:

TRUE or (FALSE)	The following instruction has no effect on the state of the LC-3: `AND R3, R3, -1` Why or why not? The following operation changes the value of the Register R3, thus changing the state.
TRUE or (FALSE)	Every line of LC-3 assembly, including assembler directives (pseudo-ops), takes up exactly 1 memory location. Why or why not? pseudo-ops like .orig don't take a place in memory, it's a way to know where the program begins. Additionally .stringz "CS2110" takes 7 memory locations although it's on one line.
TRUE or (FALSE)	To call a subroutine from another assembly file, the programmer must use .EXTERNAL and JSR (not JSRR). external is small-case operation. and we need to use JSRR instead of JSR to not be restricted by the PCOffset11 => [-1024, 1023]

TRUE or (FALSE)	The following code will pop R3 off the stack:
	`LDR R3, R6, 0`
	`ADD R6, R6, -1`
	Why or why not?
	R6 has to be incremented by 1;
	ADD R6, R6, 1

Your Initial [redacted]

2. [30 pts] Write the LC-3 program instruction(s) required for the following short problems. Please use the lines to format your code and only write one instruction per line. Some lines may be left blank. You can create labels as needed.

(a) Initial Conditions:

Assume R3 contains the address of the start of a block of memory.
Assume R4 contains a positive number N.

Problem:

The address in R3 is the start of an array of N elements. For each element in this array, replace it with double the original value (i.e., X should be updated to 2 * X).

```
.ORIG x3000
        AND R2, R2, 0     ; i = 0
        AND R2, R2, 0
        NOT R4, R4
        ADD R4, R4, 1     ; -R4 = -N
WHILE   ADD R2, R2, R4    ; i < length   i - length < 0
        BRz DONE
        AND R2, R2, 0
        LDR R2, R3, 0
        ADD R2, R2, R2    ; arr[i] * 2 = R2
        STR R2, R3, 0     ; mem[R3] = R2
        ADD R3, R3, 1
        ADD R2, R2, 1     ; i++
        BR WHILE
DONE    HALT
        .END
```

(b) Initial conditions:

Assume R3 contains the address of a string in memory that ends with a null (x0000).

Problem:

Using memory mapped I/O, print this string to the display. Assume the program will run with supervisor privileges. You may not use any TRAP instruction besides TRAP x25 (HALT).

```
.ORIG x3000
        AND R2, R2, 0
        AND R4, R3, 0

WHILE   LDR R4, R3, 0   ; R4 = mem[R3]
        ADD R2, R4, 0
        BRz DONE        ; R4 == '\0'

POLL    LDI R2, DSR_ADDR
        BRzp POLL

        STI R4, DDR_ADDR

        ADD R3, R3, 1
        BR WHILE

DONE    HALT
DSR_ADDR    .FILL   xFE04
DDR_ADDR    .FILL   xFE06
.END
```

Your Initials: [redacted]

3. [25 pts] Assemble the following program and populate the symbol table.

(a) Show the machine code in hexadecimal for the following program. Also show the address for where each line of machine code is located in memory. Do not fill in shaded cells. Show your work (where needed) to receive full credit.

Program	Address (GRADED!)	Binary (NOT GRADED!)	Hexadecimal (GRADED!)
.ORIG x3000			
L_A STI R2,L_C	x3000	1011 010 000000100	xB404
L_B NOT R2,R2	x3001	1001 010 010 11 1111	x94BF
BRzp L_A	x3002	0000011 1 1111 1110	x07FE
ST R5,L_D	x3003	0011 101 0 0000 0010	x3A02
TRAP x25	x3004	1111 0000 0010 1001	xF029
L_C .FILL L_E	x3005	0011 010 000000000	x3400
L_D .BLKW 4	x3006		
L_E .FILL xFFFF	x300A		
.END			

(b) Fill in the symbol table below for the program from part a. Unused rows should be left blank

Label	Address
L-A	x3000
L-B	x3001
L-C	x3005
L-D	x3006
L-E	x300A