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



# 88 / 100

$\cap$ II	FCT	TO	N	1
UU	ロンコ	10	IVI	- 1

# 11a5/5

- √ + 5 pts Correct (x4006)
  - + 2 pts off by one (x4005)
  - + **0 pts** incorrect

## **QUESTION 2**

# 21b8/8

- √ + 8 pts Correct (xFFFB)
  - + 5 pts not in hex but correct (-5)
  - + 0 pts incorrect

#### **QUESTION 3**

## 31c4/8

- + 8 pts Correct (x000C)
- $\checkmark$  + 4 pts x9DAC (did not execute the .fill)
- **+ 6 pts** correct but in binary (00...1100) or decimal (12)
- + 0 pts incorrect

#### **QUESTION 4**

# 41d8/8

- √ + 8 pts Correct (x0000)
- + 4 pts did not recognize RA was modified (x0001 or 1)
  - + 0 pts incorrect

## **QUESTION 5**

# 51ei 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-3cd3-42e3bcf2-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

Name [PRINT CLEARLY]:	

Your Initials:

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:] lwan

- THIS IS A CLOSED BOOK, CLOSED NOTES EXAM
- NO CALCULATORS

GT username (e.g. gburdell3):

- 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:

		X	2*	
Hex	Dec		1	2
0x1	1		2	4
0x2	2		3	8
0x3	3		4	16
0x4	4		5	32
0x5	5		6	64
0x6	6		7	128
0x7	7		8	256
0x8	8		9	512
0x9	9		10	1024
0xA	10		11	2048
0xB	11		12	4096
0xC	12		13	8192
0xD	13		14	16,384
0xE	14		15	32,768
0xF	15		16	65,536

Problem	Points	Score
1	45	
2	30	
3	25	
TOTAL	100	

GOOD LUCK!

More good facts to know:

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

- 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 | R3-0 ,-1 | 01000 8 | 10111 | 1000 | 1000 | 8 | 10111 | 1000 | 1000 | 8 | 10111 | 1000 | 8 | 10111 | 1000 | 8 | 10111 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 100

R3 contains (in hex):

DX FFF B

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

ORIG x3000 RS = Mem [X] = x SB 6 C

LD R5, X
LD R1, Y

ADD R5, R1, R5

X .FILL x5B6C
HALT
Y .FILL x4240
.END

R5 contains (in hex):

CARDAC

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

ORIG x3000

AND R2,R2, 0 R2=0

NOT R2,R2

JSR SUB

NOT R2,R2

ADD R2,R2,2

HALT

SUB ADD R7,R7, 2 (2 = 0)
ADD R2,R2, 1

RET .END

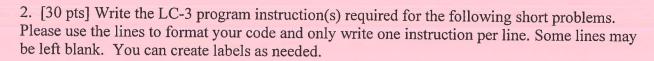
R2 contains (in hex):

01 0000 KO



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

TRUE or	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 live Orig don't take a flace in memory, it's a way to know where the program begins. Additionally . String?  "(SQ110" takes 7 memory locations although it's on one like.
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 heed  to use JSRR instead of JSR to not be  restricted by the Profisety => [-1024, 1023]
TRUE or	The following code will pop R3 off the stack:  LDR R3, R6, 0  ADD R6, R6, -1  Why or why not?  R6 vns to be in (remented by 1;  ADD R6, R6, 1



# (a) Initial Conditions:

DONE

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	×3000 AND R2. R2.0	1 = 0	/-
	ANO R2 R2. 0	)   -0	
	NOT Ry, Ry		
WHILE	A00 Ry, Ry, 1	,' - Ry => -N	th 1 Weight Lo
WITTE	Brz DONE	1 12 705	1-1003111
	AND RO. RO.D		
-	LOR R2, R3.0		
-	ADD R2, R2, R	a ; arrti]	
-	ADD 83 83 1	) i men [	R3] = R2
	ADD 82, 82, 2	11+	
	BR WHILE		
	The second second		
	1-10-11-11-11-11-11-11-11-11-11-11-11-11		
HALT			
.END			

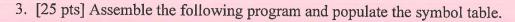
# (b) Initial conditions:

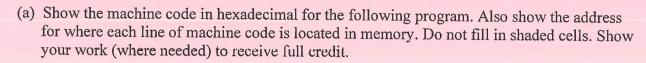
DONE

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 *3000	AND RY, RY, D
WHILE	LDR Ry, R3,0; Ry: mem [R8] ADD R2, Ry, 0 BR2 DONE ; Ry = = 1\0'
POLL	LOI R2, DSR-ADDR BRZP POLL STI B4. DOR-ADDR
	ADD R3, R3, 2 BR WHILE
_	FILL xFE04 FILL xFE06





Program	Address (GRADED!)	Binary (NOT GRADED!)	Hexadecimal (GRADED!)
.ORIG x3000	FEBRUARY CO. NO.		
L_A STI R2,L_C	X3000	2012 010 0000100	xB 404
L_B NOT R2,R2	x3001	200201001011 1111	X94BF
BRzp L_A	x 300 2	0000011 1 1111 1110	×07FE
ST R5, L_D	x3003	0011 101 10 0000 0010	x3 402
TRAP x25	43004	1111 00000001 1001	x f 0 2 9
L_C .FILL L_E	x3005	00110100000000	x3400
L_D .BLKW 4	x 3006		
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

renfination - Ro

(Affine LC) - Ro

(Affine LC) - Ro

(Affine LC) - Ro

Label	Address
L-A	x 3000
L-B	x 3001
L- C	x 3005
1-0	x3006
L-E	x 300 A

mentes so so = LE-2

$$3005 = 30011ff$$
  $000000000$   $0010. 00011001 = 25$ 

$$46: 4 = 2,111 1101 = 2$$

$$11111 1110 = 2$$

$$1 = C. fill L-E = 5 S+[L-C] = L-E = 5T[3005] = -1$$