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

Aaryan Vinay Potdar

TOTAL POINTS

94 / 100

QUESTION 1

1 1a **5 / 5**

✓ **+ 5 pts** *Correct (x5006)*

+ 2 pts off by one (x5005)

+ 0 pts incorrect

QUESTION 2

2 1b **8 / 8**

✓ **+ 8 pts** *Correct (xFFFF)*

+ 5 pts not in hex (-1)

+ 0 pts incorrect

QUESTION 3

3 1c **8 / 8**

✓ **+ 8 pts** *Correct (x0001)*

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

+ 0 pts incorrect

QUESTION 4

4 1d **4 / 8**

+ 8 pts Correct (x0009)

✓ **+ 4 pts** *x9B69 (did not execute the .fill)*

+ 0 pts incorrect

+ 6 pts correct but in binary (00...1001)

QUESTION 5

5 1ei **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 6

6 1eii **4 / 4**

✓ **+ 2 pts** *Correct (FALSE)*

✓ **+ 2 pts** *Valid Explanation:*

- When pushing to the stack, the stack pointer must be decremented

- change first line to ADD R6, R6, -1

+ 0 pts incorrect

QUESTION 7

7 1eiii **4 / 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 8

8 1eiv **4 / 4**

✓ **+ 2 pts** *Correct (FALSE)*

✓ **+ 2 pts** *Valid Explanation:*

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

- .orig / .end take up no memory

+ 0 pts incorrect

QUESTION 9

9 2a **15 / 15**

✓ **+ 15 pts** *Fully Correct*

Example:

![IMG_523246006A29-1.jpeg](/files/5357dc0a-5335-4960-b464-4da6e3b01fad)

+ 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 multiples value by 4

+ 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:

![IMG_AB11D6BCBB23-1.jpeg](/files/9354d821-2978-494f-831e-d9dc86ffe984)

+ 1 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 **23 / 25**

Part A (addresss)

+ 8 pts addresses correct

(x3000- x3006, x3009)

✓ **+ 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 (Hexidecimal)

+ 12 pts Fully correct

✓ **+ 2 pts** *1. xBC04*

✓ **+ 2 pts** *2. x5DA3*

✓ **+ 2 pts** *3. x07FD*

✓ **+ 2 pts** *4. x3602*

✓ **+ 2 pts** *5. xF025*

✓ **+ 2 pts** *6. x3009 (propagate error from address of L_Z)*

+ 0 pts incorrect

Part B

✓ **+ 5 pts** *fully correct or correct based on part A addresses (propagate error)*

L_V = x3000

L_W = x3001

L_X = x3005

L_Y = x3006

L_Z = x3009

+ 4 pts Missing 1 label or 1 incorrect address

+ 3 pts Missing 2 labels or 2 incorrect addresses

- 1 pts minor mistake (will check submissions for better partial options)

+ 0 pts incorrect

+ 0 pts incorrect

Your Initials: AP

Name [PRINT CLEARLY]: AARYAN POTDAR

GT username (e.g. gburdell3): apotdar31

CS 2110: Computer Organization and Programming

Gupta/Conte/Adams Fall 2023

QUIZ 4

VERSION C

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:]

- 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 Initials: AP

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 x5000                  0 1 2 3 4 5
A_STRING   .STRINGZ    "Adams" 0
A_LABEL          .BLKW      1
.END
```

What is the address corresponding to A_LABEL?

$\boxed{x5006}$

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

```
.ORIG x3000
AND R5,R5,0     → 0
ADD R5,R5,-1      -1
AND R5,R5,-4
ADD R5,R5,3
HALT
.END
```

R5 = 0, -1 → FFFF

4: 0100

flip 1011

+ 1

imm5 → 11100

```
AND  1111 1111 1111 1111
     1111 1111 1111 1100
     -------------------
     1111 1111 1111 1100
   + 0000 0000 0000 0011
     -------------------
     1111 1111 1111 1111  → xFFFF
```

-4 +3 → -1

R5 contains (in hex): 0x FFFF

(c) Consider the following short program and give the value in R4 after the program terminates:

```
     .ORIG x3000
     AND R4,R4, 0   → 0
     ADD R4,R4, 2   → 2
     JSR FUNC         -1
     ADD R4,R4,-2      1
     HALT
FUNC ADD R7,R7, 1
     ADD R4,R4, -1
     RET
     .END
```

HALT

R7 → 3003 + 1 = 3004

0000 0000 0000 0001 → x0001

R4 contains (in hex): 0x 0001

(d) Consider the following short program and give the value in R4 after the program terminates:

```
   .ORIG x3000
   LD R4,A      5929
   LD R0,B      4240
   ADD R4,R0,R4
A  .FILL x5929
   HALT
B  .FILL x4240
  .END
```

```
 0101 1001 0010 1001
 0100 0010 0100 0000
 -------------------
 1001 1011 0110 1001   → 9B69
```

R4 contains (in hex): 0x9B69

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

TRUE or [✓ FALSE]	To call a subroutine from another assembly file, the programmer must use .EXTERNAL and JSR (not JSRR). the programmer must use JSRR. JSR will not work as it cannot jump as far as JSRR. Also, JSRR can link a base R holding a address in another subproutine, with help of .enternal.

TRUE or [✓ FALSE]	The following code will push R3 onto the stack:
	`ADD R6, R6, 1`
	`STR R3, R6, 0`
	Why or why not?
	the contents of R3 are written to the address that R6/Stack pointer is pointing to. R3 register can be used for other purposes as its previous contents are not lost. furthermore, for push, we will use: ADD R6,R6, -1 instruction.

TRUE or [✓ FALSE]	The following instruction has no effect on the state of the LC-3: `AND R4, R4, -1` Why or why not? ~~[illegible]~~ AND with -1 means AND-ing with FFFF = 1111 1111 1111 1111$_b$ this can't change ~~[illegible]~~ the current value stored in R4, But ~~[illegible]~~ AND instruction sets C.C, changing the state of LC3.
TRUE or [✓ FALSE]	Every line of LC-3 assembly, including assembler directives (pseudo-ops), takes up exactly 1 memory location. Why or why not? • String z takes up length of string + 1 for terminating character nbre of locations. • blkw takes the nbr of memory locations specified.

Your Initials: AP

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 R5 contains the address of the start of a block of memory.

Assume R1 contains a positive number N. ↳ Happens atleast once.

Problem:

The address in R5 is the start of an array of N elements. For each element in this array, replace it with quadruple the original value (i.e., X should be updated to 4 * X).

```
        .ORIG x3000

        ADD  R2, R1, 0        ; R2 = N
        NOT  R2, R2           ;
        ADD  R2, R2, 1        ; R2 = -N
                              ; negative # on bus
FOR     BRzp DONE             ; check if 0 / +ve

        LDR  R3, R5, 0        ; R3 = mem[R5]
        ADD  R3, R3, R3       ; R3 * 2
        ADD  R3, R3, R3         (R3 ×2) ×2 = 4R3

        STR  R3, R5, 0        ; write to address

        ADD  R5, R5, 1        ; increment R5
        ADD  R2, R2, #1       ; increment R2

        BR   FOR              ; Branch



DONE    HALT
        .END
```

Your Initials: AP

(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

        POLL    LDI   R1, DSR_ADDR
                BRzp  POLL
                LDR   R2, R3, 0
                BRz   DONE
                STI   R2, DDR_ADDR
                ADD   R3, R3, #1
                BR    POLL

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

Your Initials: AP

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.

3005-(3000+1) = 4

off= 3000-(3003) = -3

of = 3006-3004=2

Program	Address (GRADED!)	Binary (NOT GRADED!)	Hexadecimal (GRADED!)
.ORIG x3000			
L_V STI R6,L_X → #4	x3000	1011 1100 0000 0100	0xBC04
L_W AND R6,R6, 3	x3001	0101 110 1 1010 0011	0x5DA3
BRzp L_V	x3002	0000 011 1 1111 1101	0x07FD
ST R3,L_Y	x3003	0011 0110 0000 0010	0x3602
TRAP x25	x3004	1111 0000 0010 0101	0xF025
L_X .FILL L_Z	x3005		0x3007
L_Y .BLKW 3	x3006		
L_Z .FILL xB3E7	x3007		
.END			

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

Label	Address
L_V	x3000
L_W	x3001
L_X	x3005
L_Y	x3006
L_Z	x3007