Homework 8

● Graded

Student

Aaryan Vinay Potdar

Total Points

100 / 100 pts

Question 1

Overview

0 / 0 pts

✔ **+ 0 pts** Correct

+ 0 pts Incorrect

Question 2

Input and Output

12 / 12 pts

2.1 **Read Instructions**

6 / 6 pts

✔ **+ 6 pts** Both correct (LDI and LDR)

+ 3 pts Partially Correct:
(1) one correct (LDI OR LDR)
(2) both correct, one incorrect

+ 0 pts Incorrect

2.2 **Write Instructions**

6 / 6 pts

✔ **+ 6 pts** Both correct (STI and STR)

+ 3 pts One correct (STI or STR) OR Two correct and one incorrect

+ 0 pts Incorrect

Question 3

Interrupts **16** / 16 pts

3.1 **(no title)** **6** / 6 pts

✔ **+ 6 pts** Correct (0x0199, x0199)

+ 0 pts Incorrect

3.2 **(no title)** **6** / 6 pts

✔ **+ 6 pts** Correct (0x1066, x1066)

+ 0 pts Incorrect

3.3 **(no title)** **4** / 4 pts

✔ **+ 4 pts** Correct (0xFE80, 0x4096, 0x0140, 0x2FFF)

+ 3 pts Partially correct -
(1) 3 out of 4 correct
(2) selects 4 correct + 1 incorrect

+ 2 pts Partially correct -
(1) 2 out of 4 correct
(2) selects 4 correct + 2 incorrect

+ 1 pt Partially correct -
(1) 1 out of 4 correct
(2) selects all

+ 2 pts Partially correct -
Gives the inverse of the correct answer (0x2110, 0x1332, 0x0500)

+ 0 pts Incorrect

Question 4

Trap Microcode — **23** / 23 pts

4.1 **(no title)** — **5** / 5 pts

✔ **+ 5 pts** Correct (C)

+ 0 pts Incorrect

4.2 **(no title)** — **6** / 6 pts

✔ **+ 2 pts** Correct MC - Selects that processor is in supervisor mode

✔ **+ 4 pts** Correct explanation - makes one of the following points:
(1) explains that trap vector table is in privileged memory
(2) any explanation mentioning that we need to be in supervisor mode to execute trap subroutines

+ 2 pts Partially correct explanation - does one of the following:
(1) minor logical error in explanation
(2) attempts explanation but logic is unclear

+ 0 pts Incorrect

4.3 **(no title)** — **6** / 6 pts

✔ **+ 6 pts** Correct -
(1) Save R6 in saved_ssp
(2) Set R6 to saved_usp

+ 3 pts Partially correct -
(1) selects one correct only
(2) selects 2 correct + 1 incorrect

+ 0 pts Incorrect

4.4 **(no title)** — **6** / 6 pts

✔ **+ 6 pts** Correct -
(1) old PSR is pushed
(2) old PC is pushed

+ 3 pts Partially correct -
(1) one correct
(2) two correct + one incorrect

+ 0 pts Incorrect

Question 5

Trap Execution — **8** / 8 pts

+ 0 pts Incorrect

✔ **+ 8 pts** Correct (0x2DE4)

Question 6

The Processor Status Register — **13** / 13 pts

6.1 (no title) — **4** / 4 pts

+ 0 pts Incorrect

✔ + 4 pts Correct (user mode)

6.2 (no title) — **5** / 5 pts

✔ + 5 pts Correct (4)

+ 0 pts Incorrect

6.3 (no title) — **4** / 4 pts

✔ + 4 pts Correct (P)

+ 0 pts Incorrect

Question 7

Interrupt Execution — **16** / 16 pts

7.1 (no title) — **4** / 4 pts

+ 0 pts Incorrect

✔ + 4 pts Correct (0x3051, x3051)

7.2 (no title) — **4** / 4 pts

+ 0 pts Incorrect

✔ + 4 pts Correct (0x8401, x8401)

7.3 (no title) — **4** / 4 pts

✔ + 4 pts Correct (0x2031, x2031)

+ 0 pts Incorrect

7.4 (no title) — **4** / 4 pts

✔ + 4 pts Correct (0x0601, x0601)

+ 0 pts Incorrect

Question 8

Returning from Interrupts **12** / 12 pts

8.1 (no title) **6** / 6 pts

✔ **+ 2 pts** Correct MC - selects False

✔ **+ 4 pts** Correct explanation - does ONE of the following
(1) mentions RTI, we use that instead
(2) explains that RET is just JMP R7, which in trap handlers we don't use R7

+ 2 pts Partially correct explanation -
(1) attempts to explain, but incorrect logic

+ 0 pts Incorrect

8.2 (no title) **6** / 6 pts

✔ **+ 2 pts** Correct MC - selects False

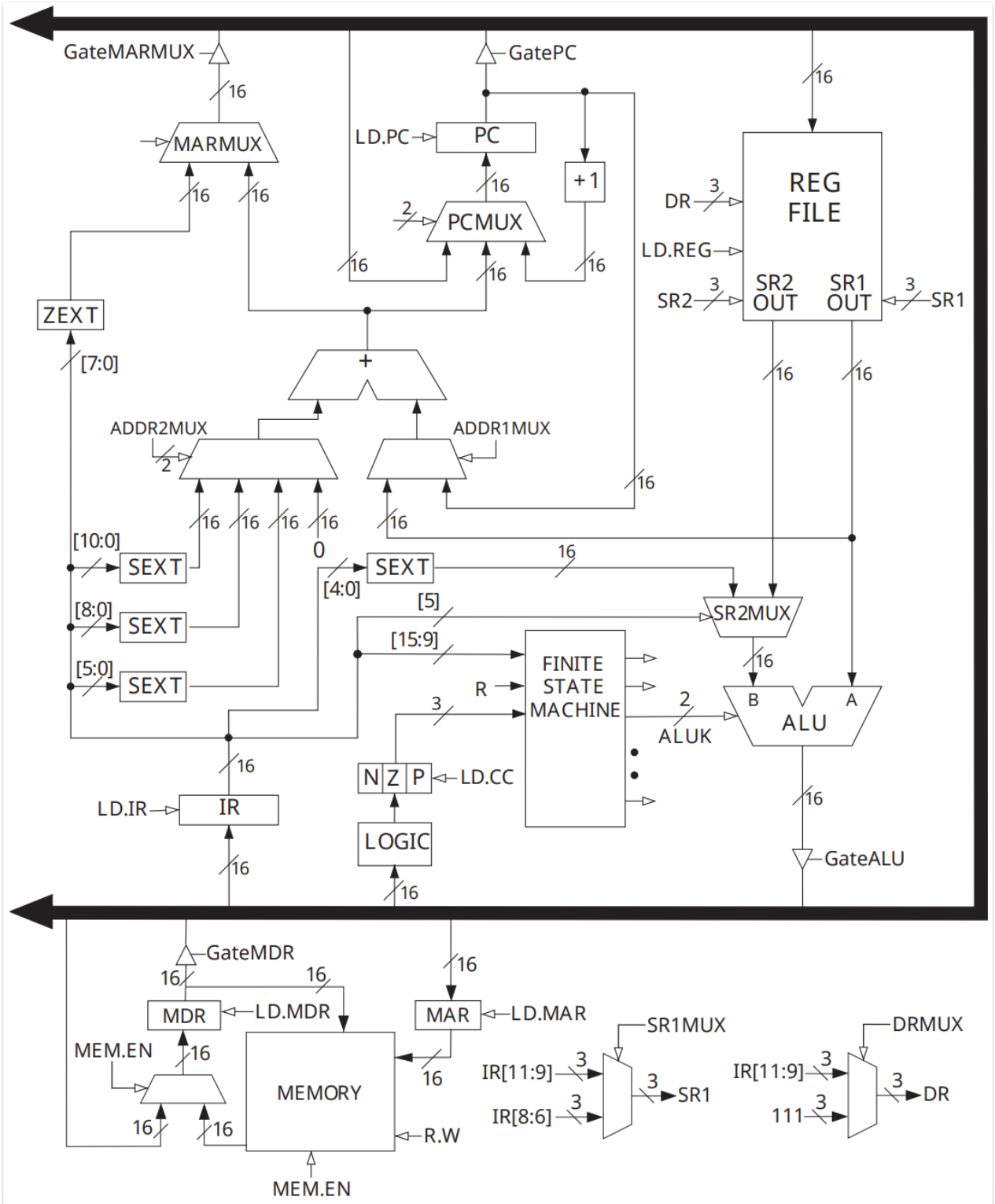
✔ **+ 4 pts** Correct explanation -
(1) gives example of interrupt during interrupt
(2) explains other valid situation where we would not change to user mode

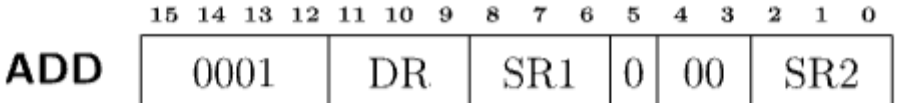
+ 2 pts Partially correct explanation -
(1) attempts correct explanation, but incorrect logic

+ 0 pts Incorrect

Q1 Overview

0 Points





ADD

0001	DR	SR1	1	imm5

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AND	0101			DR			SR1				0	00		SR2		
AND	0101			DR			SR1				1	imm5				

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT	1001			DR			SR				111111					

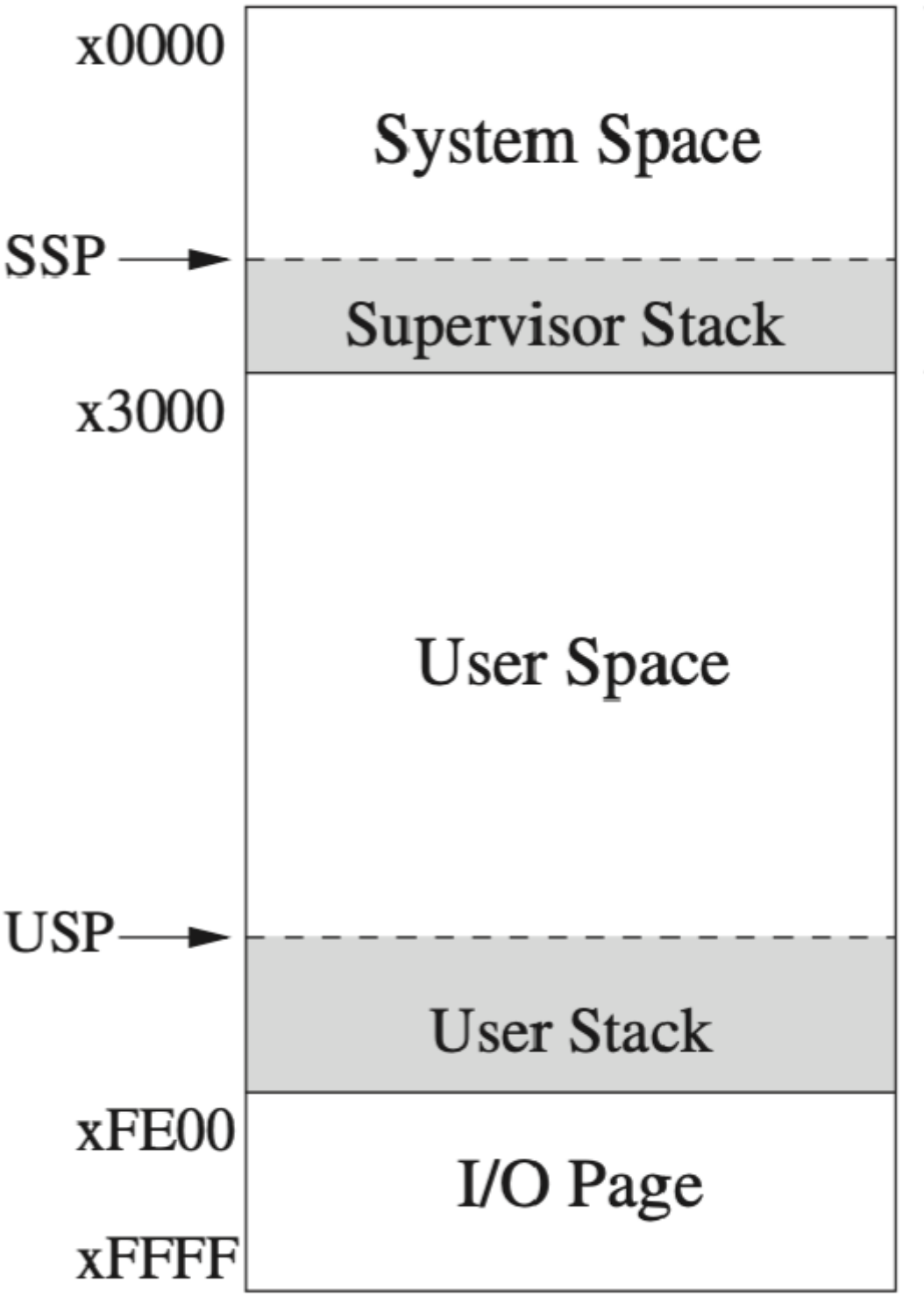
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR	0000				N	Z	P	PCoffset9								

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JMP	1100				000			BaseR			000000					
JSR	0100				1	PCoffset11										
JSRR	0100				0	00		BaseR			000000					

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LD	0010			DR				PCoffset9								
LDI	1010			DR				PCoffset9								
LDR	0110			DR				BaseR			offset6					
LEA	1110			DR				PCoffset9								

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST	0011			SR				PCoffset9								
STI	1011			SR				PCoffset9								
STR	0111			SR				BaseR			offset6					

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRAP	1111				0000				trapvect8							



This homework is worth a total of 100 points.

We have provided LC-3 datapath and instruction set here, but LC-3 reference materials can **also be found in Canvas > Files**.

This question (Q1) cannot be answered. It's used for formatting instructions. Do not worry about Gradescope saying you haven't answered one question. It's this

one!

Please complete the following problems. The collaboration policy for the course still applies. Refer to the syllabus for details regarding this policy.

Q2 Input and Output

12 Points

Q2.1 Read Instructions

6 Points

Which of the following instructions can be used to read a value from a device with a memory-mapped I/O register?

- [x] LDI
- [x] LDR
- [ ] LEA
- [ ] RTI
- [ ] STI
- [ ] STR

Q2.2 Write Instructions
6 Points

Which of the following instructions can be used to store a value to a device with a memory-mapped I/O register?

- [ ] LDI
- [ ] LDR
- [ ] LEA
- [ ] TRAP
- [x] STI
- [x] STR

Q3 Interrupts
16 Points

Use the following scenario for Q3.1 and Q3.2:

> I have a switch attached to the LC-3 that causes an interrupt when it is pressed, with the interrupt vector `0x99`. I also have an interrupt service routine that begins at address `0x1066`.
>
> To make my service routine handle the interrupt caused by pressing the switch, I need to ensure a certain entry in the interrupt vector table is set to a certain value.

Q3.1
6 Points

What is the **address** of the entry in the **interrupt vector table**?

Please answer in hexadecimal using 4 digits, and add the prefix `0x` in front of your answer. (ex. 0x0211)

0x0199

Q3.2
6 Points

What **value** should be stored in that entry?

Note: Please answer in hexadecimal using 4 digits, and add the prefix `0x` in front of your answer. (ex. `0x0211`)

0x1066

Q3.3
4 Points

Which of these addresses are unsafe locations for an interrupt service routine to begin at?

Note: The LC-3 memory layout is included at the top of this assignment.

- [ ] 0x2110
- [x] 0xFE80
- [ ] 0x1332
- [x] 0x4096
- [x] 0x0140
- [ ] 0x0500
- [x] 0x2FFF

Q4 Trap Microcode
23 Points

Q4.1
5 Points

What is the PC set to on execution of a TRAP instruction?

- ○ PC ← TrapVect8
- ○ PC ← MEM[SEXT(TrapVect8)]
- ◉ PC ← MEM[ZEXT(TrapVect8)]
- ○ PC ← MEM[TrapVect8]
- ○ PC ← MEM[MEM[SEXT(TrapVect8)]]

Q4.2
6 Points

What is the mode of the processor after the execute phase of a TRAP instruction?

- ○ The processor is in user mode
- ◉ The processor is in supervisor mode
- ○ The processor's mode doesn't change

Explain your answer in three sentences or less.

The Trap instruction updates the privilege mode. If in user mode, the current R6 is stored in saved_usp. Trap instruction sets R6 to saved_ssp and PC to MEM[ZEXT(TrapVect8)]. As the Trap vector table is in the system space, the processor is moved to supervisor mode.

Q4.3
6 Points

Which of the following operations regarding R6 happen during the execution of a RTI instruction?

Assume the RTI is moving from supervisor mode to user mode. Note that we are not implying any ordering to these events.

- [ ] Save R6 in the saved_usp
- [x] Save R6 in the saved_ssp
- [x] Set R6 to saved_usp
- [ ] Set R6 to saved_ssp

Q4.4
6 Points

Which of the following registers are pushed onto the stack during a TRAP instruction's execution?

Note that we are not implying any ordering to these events.

- [x] The old PSR is pushed
- [x] The old PC is pushed
- [ ] The old R5 is pushed
- [ ] The old R7 is pushed
- [ ] All registers are pushed

Q5 Trap Execution

8 Points

Given the following excerpt from the LC-3's memory, after the instruction `0xF003` is executed, what is the address of the next instruction to execute?

Memory address	Value in Memory
0x0000	0x18D8
0x0001	0x08FE
0x0002	0x0794
0x0003	0x2DE4
0x0004	0x1EC9
0x0005	0x1D42
0x0006	0x2613
0x0007	0x214B
0x0008	0x23B3
0x0009	0x203A
0x000A	0x086E
0x000B	0x05CB
0x000C	0x0CEB

Memory address	Value in Memory
`0x000D`	`0x13F5`
`0x000E`	`0x2832`
`0x000F`	`0x1ACA`

Please answer in hexadecimal using 4 digits, and add the prefix `0x` *in front of your answer. (ex.* `0x0211`*)*

0x2DE4

Q6 The Processor Status Register

13 Points

For the following questions, assume the value of the PSR is `0x8401`.

Q6.1

4 Points

What is the current mode?

- [ ] Supervisor mode
- [x] User mode

Q6.2

5 Points

What is the current priority? Write your answer in decimal.

4

Q6.3

4 Points

What are the current condition codes?

- [ ] N
- [ ] Z
- [x] P

Q7 Interrupt Execution

16 Points

Assume the **PSR holds** `0x8401` (just as in Question 6). This is the state of memory after the execution of an **STI instruction at address** `x3050`:

Memory Address	Value in Memory
0x0042	0x1331
....	...
0x0142	0x2031
....	...
0x3050	0xB41D

Now, suppose the processor **receives an interrupt with vector** `0x42`**.**

Q7.1

4 Points

Assume the **incoming interrupt has a priority of 4**, what is the address of the **next instruction to be executed**?

Please answer in hexadecimal and add a prefix of `0x` *in front of your answer. (ex.* `0x2110`*)*

0x3051

Q7.2
4 Points

What is the PSR when the next instruction executes? Assume the condition codes are unchanged.

Please answer in hexadecimal and add a prefix of `0x` *in front of your answer. (ex. 0x2110)*

0x8401

Q7.3
4 Points

If the **incoming interrupt** had priority 6, what is the **address of the next instruction** to be executed?

Please answer in hexadecimal and add a prefix of `0x` *in front of your answer. (ex.* `0x2110`*)*

0x2031

Q7.4
4 Points

What would be the PSR when the next instruction **executes** if the interrupt had a priority of 6? Assume the condition codes are unchanged.

Please answer in hexadecimal and add a prefix of `0x` *in front of your answer. (ex.* `0x2110`*)*

0x0601

Q8 Returning from Interrupts
12 Points

Q8.1
6 Points

Using RET to return from a trap handler subroutine is sufficient to safely return to user mode.

- ○ True
- ◉ False

Explain your answer in three sentences or less.

The RET instruction is used to return from a subroutine call. Its functionality is similar to JMP R7. RET does not restore PSR, hence, it does not return to previous privilege mode.

Q8.2
6 Points

RTI always changes the processor mode to user mode.

- ○ True
- ◉ False

Explain your answer in three sentences or less

The RTI instruction restores previous PSR, putting the processor back into the previous privilege mode. If we were in user mode when we called the TRAP, and hopped onto Supervisor mode, RTI will set LC-3 back to User mode. If we were in Supervisor mode when we called the TRAP, RTI will keep LC-3 in Supervisor mode upon return.