Homework 5

● Graded

Student

Aaryan Vinay Potdar

Total Points

100 / 100 pts

Question 1

Overview and References — 0 / 0 pts

✔ **+ 0 pts** Correct

+ 0 pts Incorrect

Question 2

Instructions — **9** / 9 pts

2.1 Memory Locations — 6 / 6 pts

✔ **+ 6 pts** Correct - 0x4003

+ 0 pts Incorrect

2.2 HALT — 3 / 3 pts

✔ **+ 3 pts** Correct (there is no instruction)

+ 0 pts Incorrect

Question 3

Mystery Instruction — **12** / 12 pts

3.1 (no title) — 8 / 8 pts

✔ **+ 8 pts** Correct (LD)

+ 0 pts Incorrect

3.2 (no title) — 4 / 4 pts

✔ **+ 4 pts** Correct:
MEM.EN, LD.MDR

+ 2 pts Partial Credit:
(1) Selects MEM.EN, LD.MDR and ONE other signal
(2) Selects ONLY MEM.EN or ONLY LD.MDR

+ 0 pts Incorrect:
(1) More than 3 signals selected
(2) Zero signals selected
(3) Only INCORRECT signals selected

Question 4

STN **14** / 14 pts

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

+ **0 pts** Incorrect

✔ + **4 pts** Correct (3)

4.2 (no title) **10** / 10 pts

✔ + **10 pts** Correct: Gate.ALU, ALUK = NOT, LD.MDR

+ **6.67 pts** Partial Credit - does one of the following:
(1) 3 correct signals and one incorrect signal
(2) 2 correct signals

+ **3.33 pts** Partial Credit - does one of the following:
(1) 3 correct signals, two incorrect signals
(2) 1 correct signal

+ **0 pts** Incorrect

Question 5

LC-3 Instructions **24** / 24 pts

5.1 (no title) **8** / 8 pts

✔ + **8 pts** Correct:
AND, R3, R2, R1

+ **6 pts** 3 out of 4 correct

+ **4 pts** 2 out of 4 correct

+ **2 pts** 1 out of 4 correct

+ **0 pts** Incorrect

5.2 (no title) **8** / 8 pts

✔ + **8 pts** Correct

+ **5.33 pts** 2 out of 3 correct

+ **2.66 pts** 1 out of 3 correct

+ **0 pts** Incorrect

5.3 (no title) **8** / 8 pts

✔ + **8 pts** Correct (1001 1100 1011 1111)

+ **0 pts** Incorrect

Question 6

Condition Codes **11** / 11 pts

6.1 **(no title)** **3** / 3 pts

✔ **+ 3 pts** Correct (LD, AND, NOT, ADD)

+ 2.25 pts Partially correct - does one of the following:
(1) 3 correct signals selected
(2) 4 correct signals selected, one incorrect signal selected

+ 1.5 pts Partially correct - does one of the following:
(1) 2 out of 4 correct signals selected
(2) 3 out of 4 correct signals selected AND one incorrect signal selected

+ 0.75 pts Partially correct - does one of the following:
(1) 1 out of 4 correct signals selected
(2) 2 out of 4 correct signals selected AND one incorrect signal selected

+ 0 pts Incorrect - no signals selected OR all signals selected

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

✔ **+ 4 pts** Correct - P

+ 2 pts Partial credit - selects P and ONE other signal

+ 0 pts Incorrect

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

✔ **+ 4 pts** Correct

+ 3.2 pts Partial credit - does ONE of the following:
(1) 4 out of 5 correct
(2) selects all

+ 2.4 pts Partial credit - 3 out of 5 correct

+ 1.6 pts Partial credit - 2 out of 5 correct

+ 0.8 pts Partial credit - 1 out of 5 correct

+ 0 pts Incorrect

Question 7

(no title) **18** / 18 pts

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

✔ **+ 6 pts** Correct (0x2110)

+ 0 pts Incorrect

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

✔ **+ 6 pts** Correct (0x1332)

+ 0 pts Incorrect

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

✔ **+ 6 pts** Correct (0xFACE)

+ 0 pts Incorrect

Question 8

(no title) **12** / 12 pts

✔ **+ 12 pts** Correct (instructions or binary)

```
NOT R0, R0
NOT R1, R1
AND R0, R1, R0
NOT R0, R0
```

```
1001000000111111
1001001001111111
0101000000000001
1001000000111111
```

```
1001 0000 0011 1111
1001 0010 0111 1111
0101 0000 0000 0001
1001 0000 0011 1111
```

+ 10 pts places result in R1 or another register (otherwise correct)

+ 9 pts 3/4 instructions correct

+ 6 pts 2/4 instructions correct

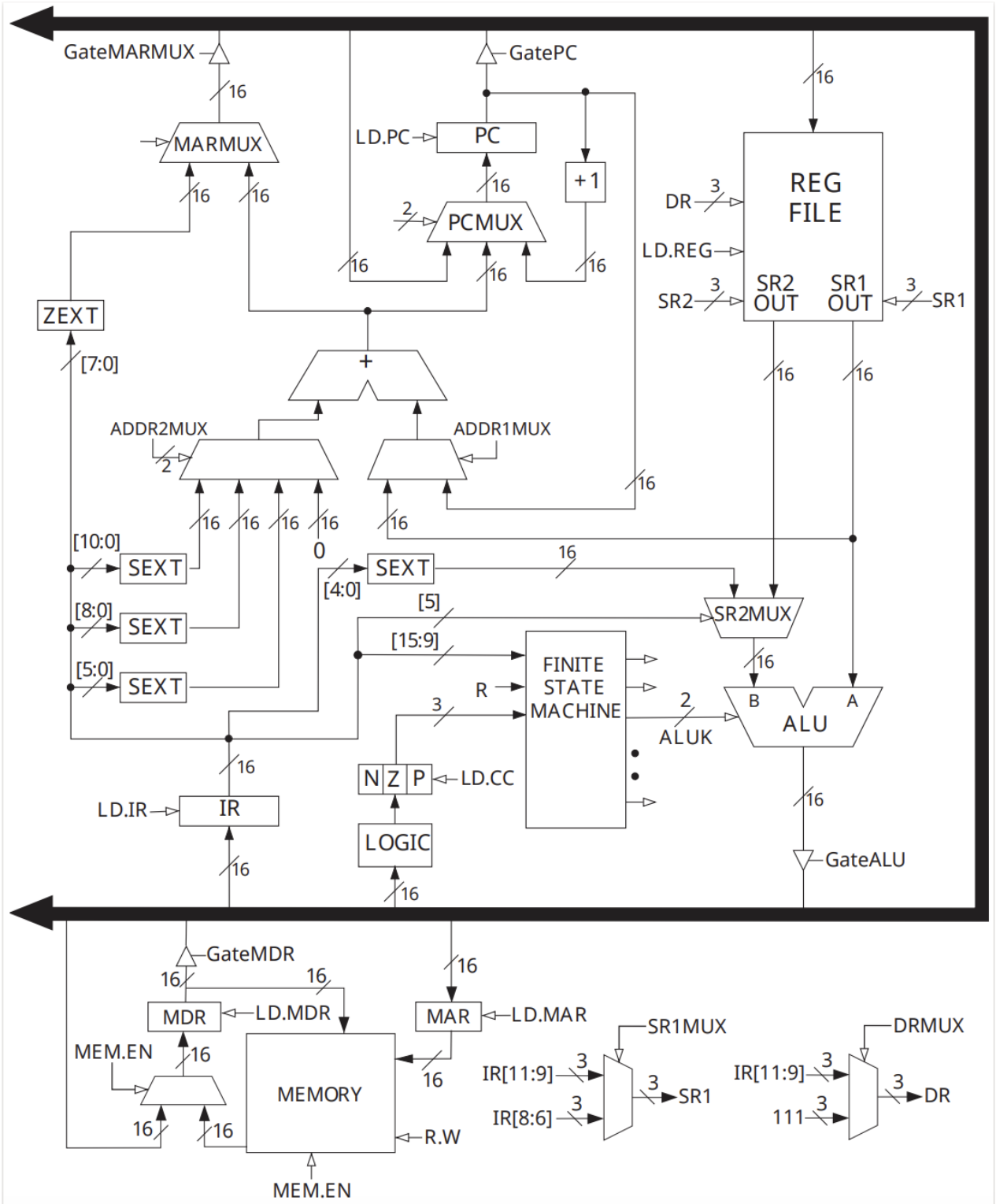
+ 3 pts improper instructions, missing operands (NOT R0, AND R0, R1)

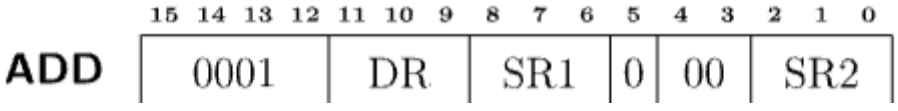
+ 3 pts attempts demorgan's (uses NOT, AND) but does not execute correctly

+ 0 pts Incorrect or does not use exactly 4 instructions

Q1 Overview and References

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 Instructions
9 Points

Q2.1 Memory Locations
6 Points

You are given a program that begins at memory location x4000. What is the value in the `PC` during the **execution phase** of the `NOT` instruction?

Please use **hexadecimal** notation **and** include the prefix `0x` in your answer (ex. `0x1234`)

Layout of memory addresses and instructions contained within them:

Memory Address	Data
x4000	ADD
x4001	ADD
x4002	NOT
x4003	AND
x4004	AND

0x4003

Q2.2 HALT
3 Points

The `HALT` instruction can **stop** the instruction cycle of the LC-3, which stops any more instructions from executing.

Which instruction is needed to **restart** the instruction cycle to continue executing more instructions?

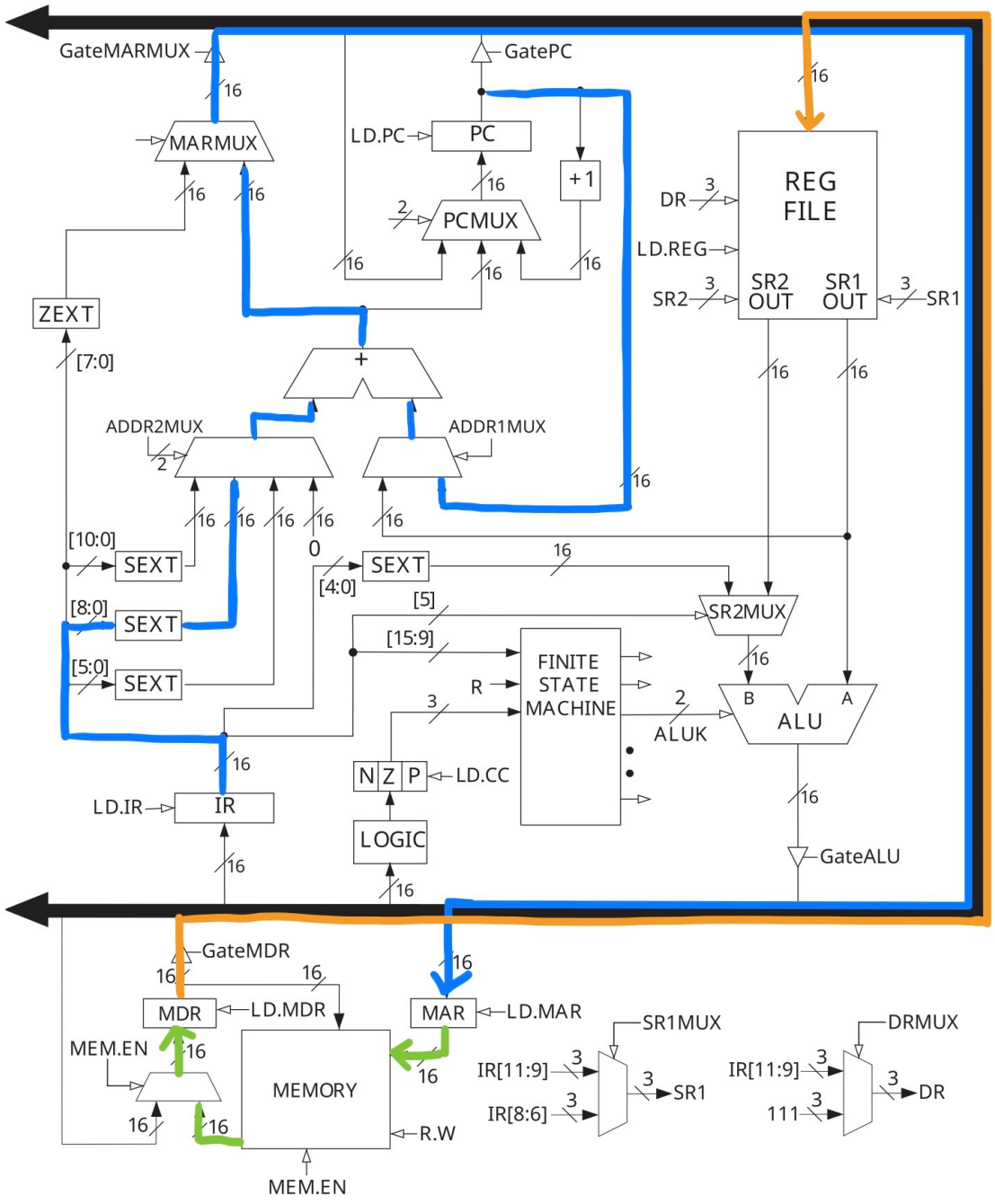
- ○ `RESTART`
- ○ `FETCH`
- ○ `CONTINUE`
- ◉ There is no instruction

Q3 Mystery Instruction

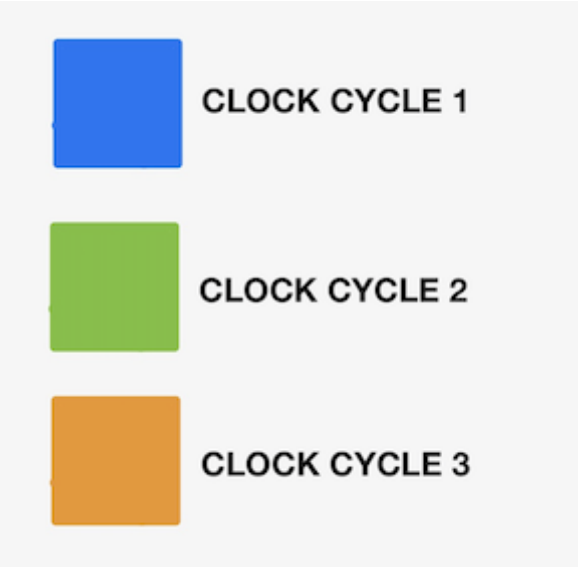
12 Points

We have provided for you below an instruction traced onto the LC-3 datapath. The instruction has **already been fetched and decoded** (not pictured), and is now in the **execution phase**.

This instruction executed in **3 clock cycles**, and the cycles are labeled by **color** (see key below). Answer the following questions about this mystery instruction.



CLOCK CYCLE LABELS:



Q3.1
8 Points

What is this instruction? Write your answer in **ALL CAPS** (ex. STR, AND)

LD

Q3.2
4 Points

Which of the following signals *must* be asserted on the **SECOND** clock cycle?

Select all that apply

- [x] MEM.EN
- [ ] MEM.WE
- [x] LD.MDR
- [ ] LD.MAR
- [ ] GATE.MDR

Q4 STN

14 Points

The 2110 TAs want to create a new LC-3 instruction called `Store Not`, or `STN`. `STN` works like `ST` but we want to invert the **value** of the source register before we store it into the memory address.

Assume this instruction **does not change value at the regfile** after execution.

Assume that the **value** we want to invert is **already** loaded into SR1.

Q4.1

4 Points

What is the **smallest** number of microstates (clock cycles) execution of this instruction would require?
Assume that this instruction has already been fetched and decoded.

3

Q4.2
10 Points

Assume `PC* + PCoffset9` has been loaded into the `MAR` during the **first** clock cycle of `STN`.

Which of the following signals should be asserted in the **SECOND** clock cycle?

Select all that apply

- [x] Gate.ALU
- [x] ALUK = NOT
- [ ] ALUK = PASS
- [ ] LD.REG
- [ ] LD.MAR
- [x] LD.MDR

Q5 LC-3 Instructions

24 Points

Q5.1

8 Points

Given the following LC-3 Instruction in binary, fill out the following information.

0101011010000001

What is this instruction? Write your answer in **ALL CAPS** (ex. NOT, LD, etc.)

AND

What is the DR? (ex. R0)

R3

What is SR1? (ex. R0)

R2

What is SR2? (ex. R3)

R1

Q5.2
8 Points

Given the following LC-3 instruction in hexadecimal, fill out the following information.

0xE74E

What is this instruction? Write your answer in ALL CAPS (ex. NOT, LD, etc.)

LEA

What is the DR? (ex. R0)

R3

What is `PCOffset9` **in decimal**?

-178

Q5.3
8 Points

Given the following instruction in LC-3 Assembly, what is its binary equivalent? Please put a space in between every **4 bits**. (ex. 1010 0001 0101 0001)

Instruction: NOT R6, R2

1001 1100 1011 1111

Q6 Condition Codes
11 Points

Q6.1
3 Points

Which of the following **LC-3 instructions** update the condition codes?

Select all that apply

- [x] LD
- [x] AND
- [x] NOT
- [x] ADD
- [ ] HALT
- [ ] JSR

Q6.2
4 Points

What is stored in the CC register after the following two LC-3 instructions, encoded in binary, execute?

0101010010100000
0001010010100001

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

Q6.3
4 Points

Which of the following `Branch` instructions will update PC during the execute macrostate?

NZP is the value currently in the CC register.

- [x] BRn, NZP= 100
- [x] BRzp, NZP= 010
- [x] BRnzp, NZP= 010
- [x] BRp, NZP= 001
- [ ] BRnp, NZP= 010
- [x] BRnz, NZP= 010

Q7
18 Points

Memory

Address	Value
0x2110	0x0831
0xBEEF	0x2110
0xDEAD	0x1332
0x5178	0x8120
0xCAFE	0xDEAD
0x9918	0x9000
0xFACE	0xABCD

Q7.1
6 Points

After the execution of the given instruction, what will be the value in **R2**? You can assume that at the time of **execution**, PC = 0xBEBB.

Please answer **in hexadecimal** and put a `0x` in front of your answer. (ex. `0x1234`)

Instruction: 0010010000110100

0x2110

Q7.2
6 Points

After execution of the given instruction, what will be the value in **R3**? You can assume that at the time of **execution**, PC = 0xCB1B.

Please answer **in hexadecimal** and put a `0x` in front of your answer. (ex. `0x1234`)

Instruction: 1010011111100011

0x1332

Q7.3
6 Points

Given the following instruction, what address in memory will be **updated**?

You can assume that at the time of **execution**, PC = 0xCADE and R3 holds 0xFABE.

Please answer **in hexadecimal** and put a `0x` in front of your answer. (ex. `0x1234`)

Instruction: 0111001011010000

0xFACE

Q8
12 Points

Write **four LC-3 instructions** for a program that calculates `R0` **OR** `R1`. Place the **result** of the operation in **R0**.

You must use **exactly four** instructions to receive full credit.

A or B is equal to NOT(NOT A AND NOT B)

Instructions:
1. NOT R0, R0
2. NOT R1, R1
3. AND R0, R0, R1
4. NOT R0, R0