Assessment 4

Due Aug 5 at 12:07pm **Points** 100 **Questions** 30
Available Aug 5 at 11am - Aug 5 at 12:08pm about 1 hour **Time Limit** 60 Minutes

Instructions

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	60 minutes	76.78 out of 100 *

* Some questions not yet graded

ⓘ Correct answers are hidden.

Score for this quiz: **76.78** out of 100 *
Submitted Aug 5 at 12pm
This attempt took 60 minutes.

This assessment is to be completed on an **individual** basis, no collaboration. This assessment is timed. You have 60min to complete it. It is recommended you take the questions in order. If you jump around you are likely to miss a question. It is worth 13.5% of your grade. Just relax and answer the questions deliberately and thoughtfully.

Some multiple choice questions will take you 30sec or less. Some might take you as much as 90 to 120sec, but on average you should average just over a minute per multiple choice. This should leave you enough time to complete the 3 "programming" type questions.

If you need to contact me during the exam I will be on BBCollaborate "Erics Office Hours" and also on MS Teams. Do **not** use email

Question 1

3 / 3 pts

The value furthest left on a number line (smallest) that can be represented by a 5-bit **unsigned** binary number is (enter **5-bit binary** number) 00000

The value furthest right (largest) that can be represented by a 5-bit **unsigned** binary number is (enter **5-bit binary** number) 11111

Answer 1:

00000

Answer 2:

11111

Question 2

3 / 3 pts

The value furthest left on a number line (smallest) that can be represented by a 4-bit **2's complement (signed)** binary number is (enter **4-bit binary** number) 1000 . The value of that number in decimal is -8

The value furthest right (largest) that can be represented by a 4-bit **2's complement (signed)** binary number is (enter **4-bit binary** number) 0111 . The value of that number in decimal is 7

Answer 1:

1000

Answer 2:

-8

Answer 3:

0111

Answer 4:

7

Question 3

3 / 3 pts

Represent the value -7 as a 4-bit **2's complement** number 1001

Represent the value 8 as a 4-bit **2's complement** number cannot be represented

Answer 1:

1001

Answer 2:

cannot be represented

Question 4

3 / 3 pts

The largest value (in decimal) that can be represented by a **5-bit unsigned binary fixed point** number that has 2 fractional bits is 7.75

The value (in decimal) of the number 01101 in this format would be 3.25

Answer 1:

7.75

Answer 2:

3.25

Question 5

3 / 3 pts

For the following 5-bit operations indicate **when they would overflow** a 5-bit result

11001 + 11100 if numbers are unsigned

01100 + 00100 if numbers are 2's comp

01010 + 10111 if numbers are unsigned

10010 + 10110 overflow for both unsigned and 2's comp

Answer 1:

if numbers are unsigned

Answer 2:

if numbers are 2's comp

Answer 3:

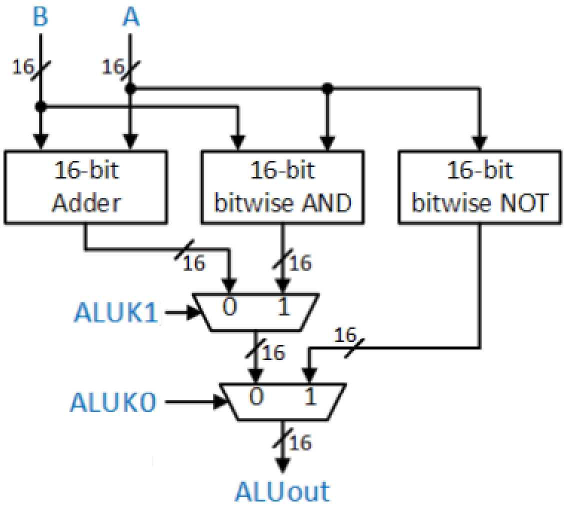
if numbers are unsigned

Answer 4:

overflow for both unsigned and 2's comp

Question 6

3 / 3 pts



What values to the mux select signals **ALUK1**and **ALUK0** need to be to execute the following instructions:

AND R2, R3, #4

ALUK1= 1

ALUK0 = 0

NOT R3, R3

ALUK1 = X (don't care)

ALUK0= 1

Answer 1:

1

Answer 2:

0

Answer 3:

X (don't care)

Answer 4:

1

Question 7

3 / 3 pts

Add the following two hex numbers

9E

+ 13

B1

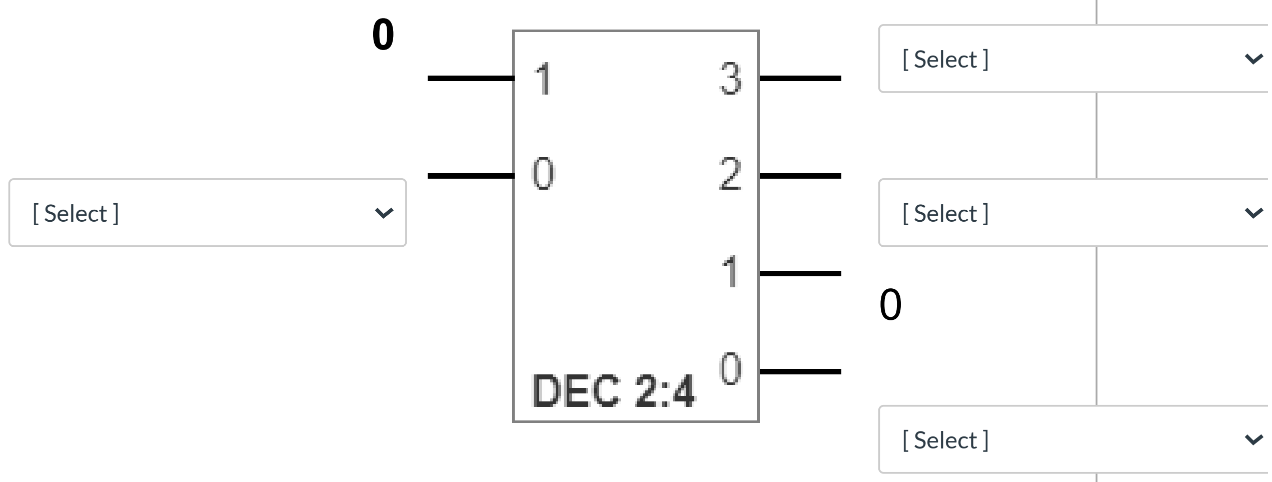
Answer 1:

B1

Question 8

3 / 3 pts

Fill in what the remaining signal value must be for the 2:4 decoder



Answer 1:

0

Answer 2:

0

Answer 3:

0

Answer 4:

1

Question 9

3 / 3 pts

How many flip-flops are required to implement a FSM that has 11 states?

4

Partial

Question 10

2.25 / 3 pts

Which of these is more abstract LC-3 Assembly

Which of these is more abstract logic gates

Which of these is more abstract problem statement

Which of these is more abstract a sketch of a person

Answer 1:

LC-3 Assembly

Answer 2:

logic gates

Answer 3:

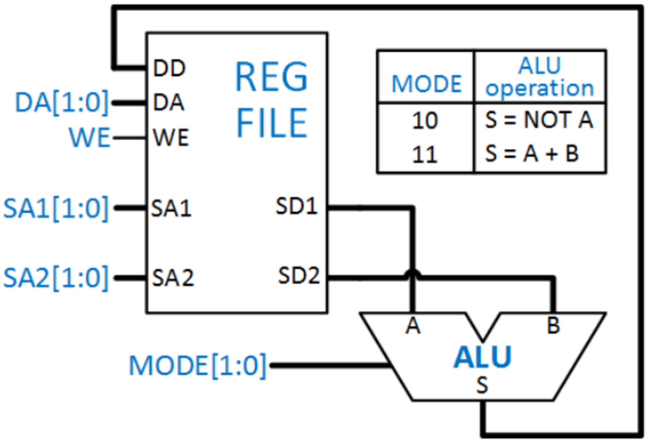
problem statement

Answer 4:

a sketch of a person

Question 11

3 / 3 pts



MODE	ALU operation
10	S = NOT A
11	S = A + B

Given the data path and signals

what operation is being performed?

DA = 11, **WE** = 1, **SA1** = 10, **SA2** = 01, **MODE** = 10

R3 <-- NOT(R2)

Answer 1:

R3

Answer 2:

NOT(

Answer 3:

R2

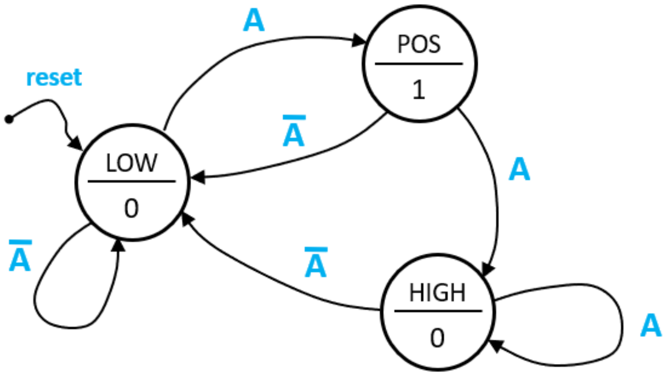
Answer 4:

)

Partial

Question 12

3 / 4 pts



The input(s) signals for this state diagram is/are: A, clk, reset

The output(s) signal for this state diagram is/are: not named, but just a single bit wide

This state machine that implements this diagram would require 2 flops

The best description of this state diagram is toggles (changes) its output every time the input toggles (changes)   **HINT**: when is the only time the output is high and can the machine stay in that state?

Answer 1:

A, clk, reset

Answer 2:

not named, but just a single bit wide

Answer 3:

2

Answer 4:

toggles (changes) its output every time the input toggles (changes)

Question 13

3 / 3 pts

I/O peripherals are typically controlled through memory mapped registers. These are registers that are located outside of main memory and the register file . We read and write to these registers using load/store instructions. They are referenced by these instructions through assigned memory addresses .

Answer 1:

located outside of main memory and the register file

Answer 2:

load/store

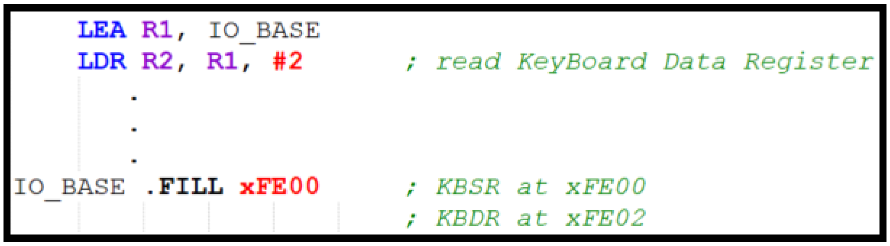
Answer 3:

assigned memory addresses

Question 14

3 / 3 pts

The method we would typically use for reading the keyboard data register is shown below:



```
     LEA R1, IO_BASE
     LDR R2, R1, #2         ; read KeyBoard Data Register
         .
         .
         .
IO_BASE .FILL xFE00         ; KBSR at xFE00
                            ; KBDR at xFE02
```

Use the drop downs below to accomplish the same thing (*load* ***KBDR*** *into* ***R2***).

```
          LDI       R2,       KBDR
               .
               .
               .
 KBDR     .FILL     xFE02
```

Answer 1:

LDI

Answer 2:

R2,

Answer 3:

KBDR

Answer 4:

KBDR

Answer 5:

.FILL

Question 15

3 / 3 pts

An analog to digital converter would be considered to be an input device

A WiFi module would be considered to be a bi-directional device

A motor speed controller would be considered to be an output device

A push button would be considered to be an input device

Answer 1:

an input device

Answer 2:

a bi-directional device

Answer 3:

an output device

Answer 4:

an input device

Question 16

3 / 3 pts

Match these statements about copying data *(in a single instruction)* on an LC-3 processor:

To copy data from memory to the register file we use?

It has to be a LOAD instructio

To copy data from one register to another register we use?

It has to be an OPERATE instr

To copy data from a register in the register file to memory we use?

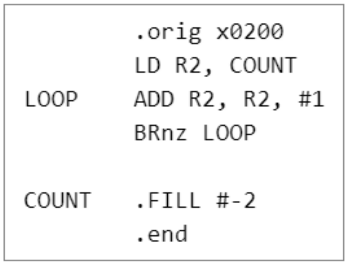
It has to be be a STORE instru ⌄

To copy data from one memory location to another we use?

Multiple instructions, cannot l ⌄

Question 17

3 / 3 pts



```
        .orig x0200
        LD R2, COUNT
LOOP    ADD R2, R2, #1
        BRnz LOOP

COUNT   .FILL #-2
        .end
```

The instruction at label LOOP (**ADD R2**, **R2**, #1) is executed 3 times.

if the **BRnz** was changed to **BRn** it would be executed 2 times

Answer 1:

3

Answer 2:

2

Question 18

Not yet graded / 7 pts

The following problem will be manually graded and will show as zero when you first submit. Your grade will be updated later.

Download this code and flesh it out to perform a read of a character from the LC-3 keyboard and place it in **R0**. You are **not allowed** to use a call to an OS trap instruction to do it.

When you are satisfied with your code copy and paste it into the "essay question box" below.

Your Answer:

```
; Filename: Read Keyboard
; Author: Ariel Fu
; Description: Read a character from the keyboard and place in R0
; Does not have to be a sub-routine (no context save/restore)
; Need to ensure keyboard has character ready before reading it

.orig x0200

START
LD R1, IO_BASE
; add code here
GET_CHAR
LDR R2, R1, #0 ; get the status of the keyboard
BRzp GET_CHAR

LDR R0, R1, #2


BR START ; loop forever (do it again)

IO_BASE .FILL xFE00 ; KBSR lives here
; KBDR is at xFE02

.end
```

Question 19

3 / 3 pts

In general, if PennSim is paused at some arbitrary address **ADDR**, clicking ***Step*** will execute a single instruction and then pause. If you click ***Next*** when the instruction at **ADDR** is **JSR**, **JSRR**, or a **TRAP** instruction, PennSim will execute multiple instructions and the next instruction to execute is the one at address ADDR + 1

Answer 1:

execute a single instruction

Answer 2:

execute multiple instructions

Answer 3:

ADDR + 1

Partial

Question 20

2 / 3 pts

When you write LC-3 code intended to be run with the LC3-OS your code must be located at address x3000 . This is done with a(n) .ORIG directive. The OS code is not run until your code calls a TRAP

Answer 1:

located at address x3000

Answer 2:

.ORIG

Answer 3:

is not run until your code calls a TRAP

Question 21

Not yet graded / 7 pts

The following problem will be manually graded and will show as zero when you first submit. Your grade will be updated later.

Download **this template** and flesh it out to perform the classic "Hello World!" program that prints "Hello World!" to the console. You are allowed to use LC-3 TRAP instructions to do this.

When you are satisfied with your code copy and paste it into the "essay question box" below.

Your Answer:

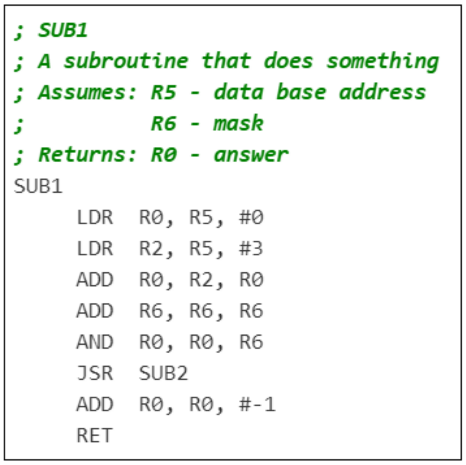
```
; Filename: HelloWorld.asm
; Author: Ariel Fu
; Description: A classic loved by all (prints Hello World! to console)

.orig x3000

START

LEA R0, HELLO_WORLD
PUTS

BR START ; loop forever (do it again)
```

HELLO_WORLD .STRINGZ "Hello World!"

.end

Question 22

3 / 3 pts



```
; SUB1
; A subroutine that does something
; Assumes: R5 - data base address
;          R6 - mask
; Returns: R0 - answer
SUB1
     LDR  R0, R5, #0
     LDR  R2, R5, #3
     ADD  R0, R2, R0
     ADD  R6, R6, R6
     AND  R0, R0, R6
     JSR  SUB2
     ADD  R0, R0, #-1
     RET
```

Which registers should be part of the context save/restore for this subroutine?

If performing a save/restore would be wasteful, but not functionally incorrect then answer "could be"

R0 cannot be saved/restored

R2 must be saved/restored

R5 could be (but would not have to be) saved/restored

R6 must be saved/restored

R7 must be saved/restored

Answer 1:

cannot be

Answer 2:

must be

Answer 3:

could be (but would not have to be)

Answer 4:

must be

Answer 5:

must be

Question 23

3 / 3 pts

You have been asked to use a single LC-3 instruction to copy a value in memory to a register.

Which, if any, of the below LC-3 instructions can you use?

ADD can't use

NOT can't use

LD can use

LDI can use

LEA can't use

STI can't use

Answer 1:

can't use

Answer 2:

can't use

Answer 3:

can use

Answer 4:

can use

Answer 5:

can't use

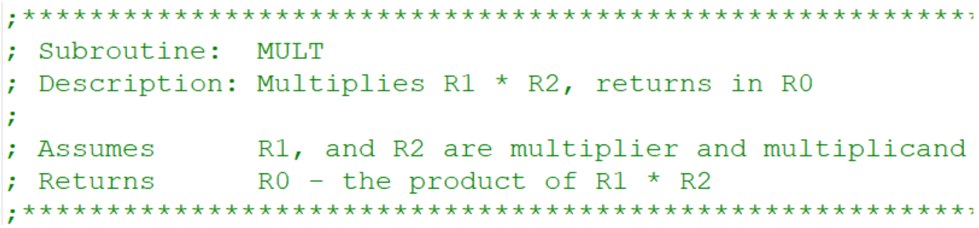
Answer 6:

can't use

Question 24

3 / 3 pts

The header for the MULT routine we used in HW12 is given below:



```
;********************************************************;
; Subroutine:  MULT
; Description: Multiplies R1 * R2, returns in R0
;
; Assumes      R1, and R2 are multiplier and multiplicand
; Returns      R0 – the product of R1 * R2
;********************************************************;
```

As we were performing successive iteration (guess and check) to compute the **SQRT** we were building up our guess in R2. Fill in the code that would be needed between forming our newest guess and computing its square. If no code is needed enter "no code" (without quotes). This problem will be looked at manually to check for auto grading mistakes.

SQRT_LOOP

ADD R2, R2, R5 ; form newest "guess" to check

AND R1, R2, R2

JSR MULT

Answer 1:

AND R1, R2, R2

Question 25

2 / 2 pts

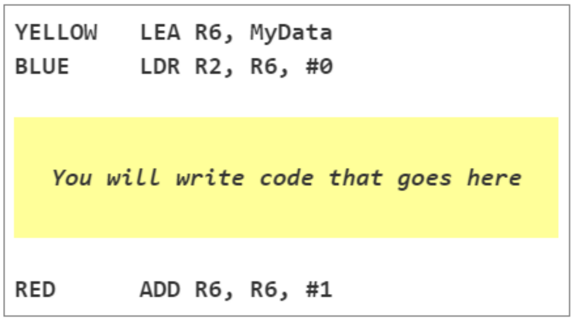
You will need to refer to your LC-3 Programmer's Reference sheet.

How would the LC-3 assembler encode the **GETC** TRAP mneumonic. (enter a 4-digit hex encoding)

F020

Question 26

Not yet graded / 4 pts



```
YELLOW   LEA R6, MyData
BLUE     LDR R2, R6, #0

    You will write code that goes here

RED      ADD R6, R6, #1
```

This problem will be manually graded, so your score will initially be zero.

Write code that will go in the yellow box that will call a subroutine **IS_BLUE** if and only if **R2** is **positive** after the **LDR** instruction.

You **cannot** add any new labels - you may only use (if needed) the labels already defined.

Your Answer:

BRp IS_BLUE

Question 27

3 / 3 pts

Due to limitations in rendering inside drop boxes $\overline{B}$ will be shown as **B_bar**

1 AND **B** = B

B AND **B** = B

0 OR **B** = B

1 OR **B** = 1

1 XOR **B** = B_bar

Answer 1:

B

Answer 2:

B

Answer 3:

B

Answer 4:

1

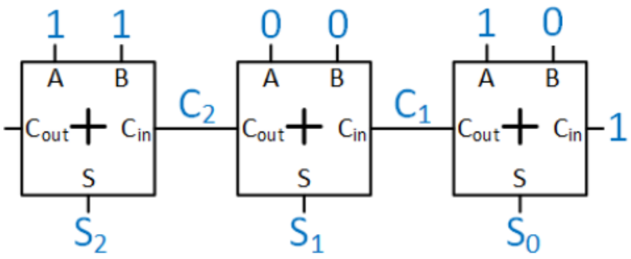
Answer 5:

B_bar

Partial

Question 28

3.33 / 4 pts



If the operands were considered **unsigned** the description of the operation (in decimal) would be:

Operand A		Operand B		Cin
5	+	4	+	1

If the operands were considered 2's complement the description of the operation would be

Operand A		Operand B
-3	+	-4

With regard to overflow both scenarios represent an operation that overflows

Answer 1:

5

Answer 2:

4

Answer 3:

1

Answer 4:

-3

Answer 5:

-4

Answer 6:

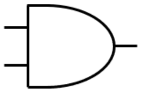
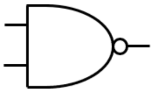
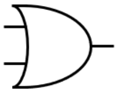
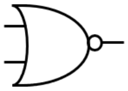
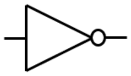
both scenarios represent an operation that overflows

Partial

Question 29

1.2 / 3 pts

Implement the given equation the most efficient way using the gates shown. How many of each gate does your implementation use?

$F = \overline{A \cdot B + C}$	
	1
	0
	1
	0
	1

Answer 1:

1

Answer 2:

0

Answer 3:

1

Answer 4:

0

Answer 5:

1

Question 30

3 / 3 pts

In the ethics video for self-driving cars which of the following scenarios were part of the dilemma

☑ a box fell off a truck in front of your car

□ A motorist was pointing a gun at you

☑ one motor cycle rider had a helmet and one did not

□ a semi-tractor trailer was beside you

□ The driver over rode the decision of the self-driving car

☑ A SUV was a possible victim of your self-driving car's "decision"

Quiz Score: **76.78** out of 100