Question 2: Boolean Expression to Truth Table (4 Variables)

Fill in the truth table for a Boolean function given by

$$F(A,B,C,D) = \bar{A}\bar{B}\bar{C}\bar{D} + ABC + AD + BD$$

A	B	C	D	F
0	0	0	0	1 100%
0	0	0	1	0 100%
0	0	1	0	0 100%
0	0	1	1	0 100%
0	1	0	0	0 100%
0	1	0	1	1 100%
0	1	1	0	0 100%
0	1	1	1	1 100%
1	0	0	0	0 100%
1	0	0	1	1 100%
1	0	1	0	0 100%
1	0	1	1	1 100%
1	1	0	0	0 100%
1	1	0	1	1 100%
1	1	1	0	1 100%
1	1	1	1	1 100%

This question is complete and cannot be answered again.

Correct answer

A	B	C	D	F
0	0	0	0	1
0	0	0	1	0

0	0	1	0	`0`
0	0	1	1	`0`
0	1	0	0	`0`
0	1	0	1	`1`
0	1	1	0	`0`
0	1	1	1	`1`
1	0	0	0	`0`
1	0	0	1	`1`
1	0	1	0	`0`
1	0	1	1	`1`
1	1	0	0	`0`
1	1	0	1	`1`
1	1	1	0	`1`
1	1	1	1	`1`

Submitted answer 5
Submitted at 2023-05-06 15:30:42 (CDT) 100%

hide

A	B	C	D	F
0	0	0	0	`1` 100%
0	0	0	1	`0` 100%
0	0	1	0	`0` 100%
0	0	1	1	`0` 100%
0	1	0	0	`0` 100%
0	1	0	1	`1` 100%
0	1	1	0	`0` 100%
0	1	1	1	`1` 100%
1	0	0	0	`0` 100%
1	0	0	1	`1` 100%
1	0	1	0	`0` 100%
1	0	1	1	`1` 100%

1	1	0	0	0 100%
1	1	0	1	1 100%
1	1	1	0	1 100%
1	1	1	1	1 100%

Submitted answer 4
Submitted at 2023-05-06 15:29:44 (CDT) invalid, not gradable

show

Submitted answer 3
Submitted at 2023-05-06 15:29:35 (CDT) invalid, not gradable

show

Show/hide older submissions

Exam 4

Assessment overview

Question

Total points: 5 /5

Auto-graded question

You are viewing the question instance of a different user and so are not authorized to report an error.

Previous question Next question

Personal Notes

No attached notes

Notes can't be added or deleted because the assessment is closed.

Staff information

Staff user:

Ujjal Kumar Bhowmik ubhowmik@illinois.edu

▸ Student details

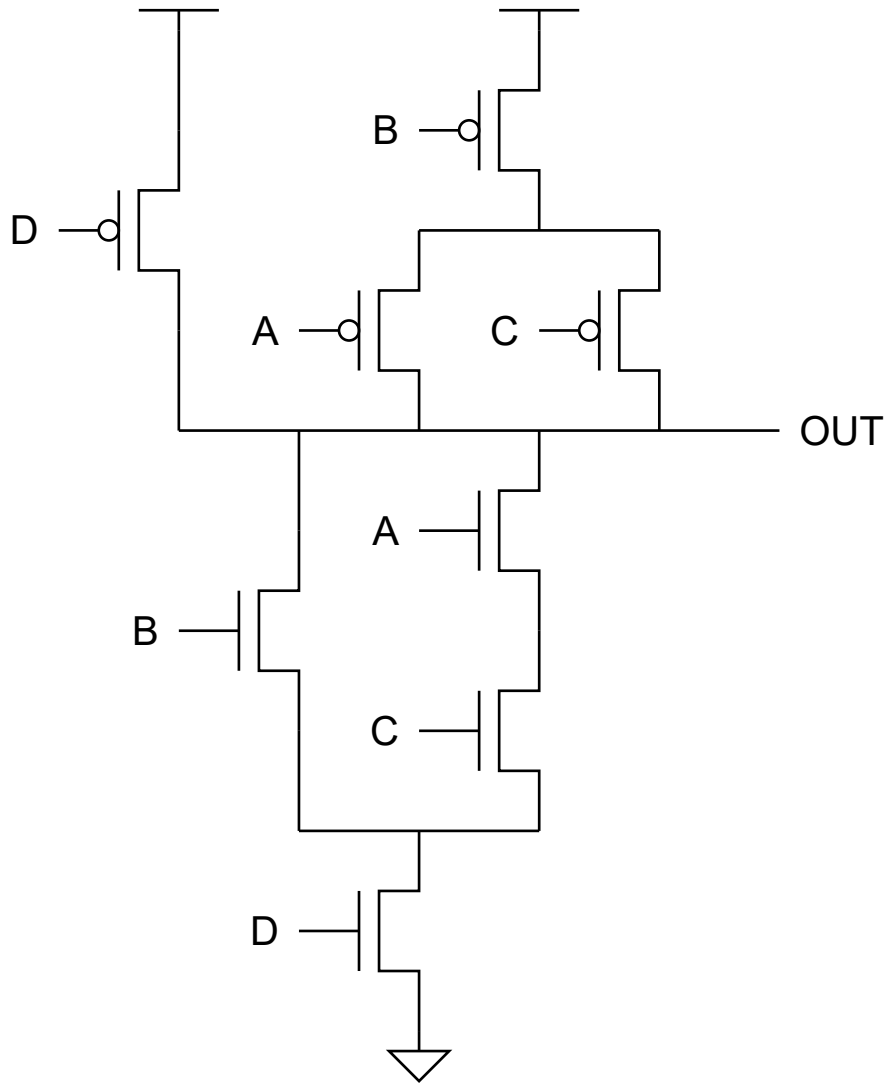
Question:

QID: M01-08
Title: Boolean Expression to Truth Table (4 Variables)
Started at: 2023-05-06 15:28:13 (CDT)
Duration: 00:02:32.648169

Show/Hide answer

Question 3: CMOS Gate Truth Table (4 variables)

Enter the truth table for the CMOS shown below.



A	B	C	D	OUT
0	0	0	0	1 100%
0	0	0	1	1 100%
0	0	1	0	1 100%
0	0	1	1	1 100%
0	1	0	0	1 100%
0	1	0	1	0 100%
0	1	1	0	1 100%
0	1	1	1	0 100%
1	0	0	0	1 100%
1	0	0	1	1 100%
1	0	1	0	1 100%
1	0	1	1	0 100%

1	1	0	0	1	100%
1	1	0	1	0	100%
1	1	1	0	1	100%
1	1	1	1	0	100%

This question is complete and cannot be answered again.

Correct answer

A	B	C	D	OUT
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Submitted answer 5

Submitted at 2023-05-06 15:35:36 (CDT) 100%

hide

A	B	C	D	OUT
0	0	0	0	1 100%
0	0	0	1	1 100%
0	0	1	0	1 100%
0	0	1	1	1 100%
0	1	0	0	1 100%
0	1	0	1	0 100%
0	1	1	0	1 100%
0	1	1	1	0 100%
1	0	0	0	1 100%
1	0	0	1	1 100%
1	0	1	0	1 100%
1	0	1	1	0 100%
1	1	0	0	1 100%
1	1	0	1	0 100%
1	1	1	0	1 100%
1	1	1	1	0 100%

Submitted answer 4

Submitted at 2023-05-06 15:35:02 (CDT) invalid, not gradable

show

Submitted answer 3

Submitted at 2023-05-06 15:34:29 (CDT) invalid, not gradable

show

Show/hide older submissions

Exam 4

Assessment overview

Question

Total points: 5 /5

Auto-graded question

You are viewing the question instance of a different user and so are not authorized to report an error.

Question 4: Missing Instruction

An LC-3 computer starts with the following register and memory contents. In addition, the word at address x3000, which is not shown, contains an instruction, one of ADD, AND, BR, LD, LDI, LDR, LEA, NOT, ST, STI, or STR. *Note:* The LEA instruction does not change the CC register (3rd edition behavior).

Registers		Memory
R0: x2B50	**R4:** x60A0	
R1: xBEDF	**R5:** x401C	**x3001:** x3002
R2: x301D	**R6:** x301C	**x3002:** x4001
R3: x401B	**R7:** x16E1	**x4000:** x3001
PC: x3000	**CC:** b010	**x4001:** xD710

All other memory locations start with x0000. After the instruction at address x3000 is executed, the contents of registers and memory are:

Registers		Memory
R0: x2B50	**R4:** x60A0	
R1: xBEDF	**R5:** x401C	**x3001:** x3002
R2: x301D	**R6:** x301C	**x3002:** x4001
R3: x401B	**R7:** x16E1	**x4000:** x3001
PC: x300B	**CC:** b010	**x4001:** xD710

What value could be stored at address x3000? Enter your answers in **hexadecimal**. There may be more than one correct answer, however, your answer must be one of the instructions ADD, AND, BR, LD, LDI, LDR, LEA, NOT, ST, STI, or STR.

x3000: x0E0A 100%

This question is complete and cannot be answered again.

Correct answer

x3000: `x060A`

```
BRzp  #10
```

Submitted answer
Submitted at 2023-05-06 15:40:02 (CDT) 100%

hide

x3000: `x0E0A` 100%

CORRECT!

Exam 4

Assessment overview

Question

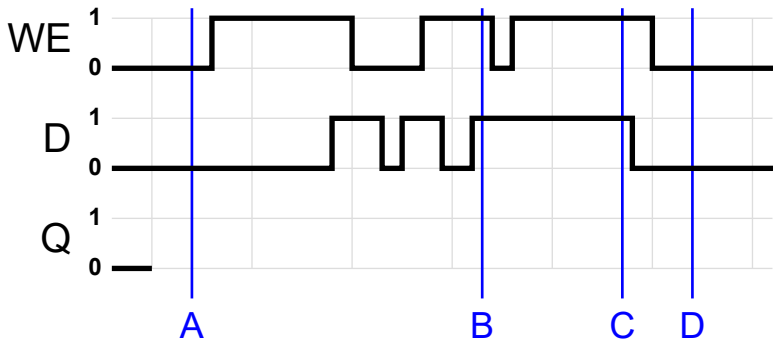
Total points: 5 /5

Auto-graded question

You are viewing the question instance of a different user and so are not authorized to report an error.

Question 5: D Latch Timing Diagram

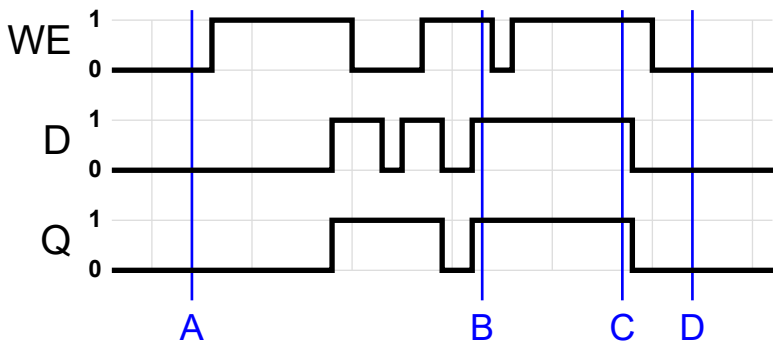
Shown below is a timing diagram for a **D latch**. Enter the values of signal Q at each of times A, B, C, D indicated in the diagram.



Q at time A:	0	100%
Q at time B:	1	100%
Q at time C:	1	100%
Q at time D:	0	100%

This question is complete and cannot be answered again.

Correct answer



Q at time A: `0`

Q at time B: `1`

Q at time C: `1`

Q at time D: `0`

Submitted answer 5
Submitted at 2023-05-06 15:42:27 (CDT) 100%

hide

Q at time A: `0` 100%

Q at time B: `1` 100%

Q at time C: `1` 100%

Q at time D: `0` 100%

Submitted answer 4
Submitted at 2023-05-06 15:42:14 (CDT) saved, not graded

show

Submitted answer 3
Submitted at 2023-05-06 15:41:44 (CDT) invalid, not gradable

show

Question 6: The Sum of Its Parts

Recall that a full adder takes three inputs, denoted A, B, and C_{IN}, and produces two outputs, denoted S and C_{OUT}. Inputs A and B are the 1-bit addends and input C_{IN} is the carry-in bit. Output S is the 1-bit sum and C_{OUT} is the carry out.

The following questions ask you if it is possible to construct a full adder using certain combinations of components.

Assume the following:

- Logic gates have no limits on the number of inputs,
- Connections (wires) do not count as components,
- Any component input can be connected to ground (logic-0) or supply (logic-1), and
- Adder inputs A, B, and C_{IN} are also available complemented.

Important: Do not guess. *You only have one attempt for this problem!*

Is it possible to build a full adder from one XOR gate, three AND gates, and one OR gate?

◉ Yes ☐ ○ No 100%

Is it possible to build a full adder from nine NAND gates?

◉ Yes ☐ ○ No 100%

Is it possible to build a full adder from nine NOR gates?

◉ Yes ☐ ○ No 100%

Is it possible to build a full adder from ten XOR gates?

○ Yes ◉ No ☐ 100%

Is it possible to build a full adder from two 4-to-1 multiplexers?

◉ Yes ☐ ○ No 100%

Is it possible to build a full adder from one 8-to-1 multiplexer?

○ Yes ◉ No ☐ 100%

Is it possible to build a full adder from one 3-to-8 decoder and two OR gates?

◉ Yes ☐ ○ No 100%

Is it possible to build a full adder from one 3-to-8 decoder and two XOR gates?

◉ Yes ☐ ○ No 100%

Is it possible to build a full adder from one 2-to-1 multiplexer and one XOR gate?

○ Yes ◉ No ☐ 100%

This question is complete and cannot be answered again.

Correct answer

Is it possible to build a full adder from one XOR gate, three AND gates, and one OR gate?

Yes

Is it possible to build a full adder from nine NAND gates?

Yes

Is it possible to build a full adder from nine NOR gates?

Yes

Is it possible to build a full adder from ten XOR gates?

No

Is it possible to build a full adder from two 4-to-1 multiplexers?

Yes

Is it possible to build a full adder from one 8-to-1 multiplexer?

No

Is it possible to build a full adder from one 3-to-8 decoder and two OR gates?

Yes

Is it possible to build a full adder from one 3-to-8 decoder and two XOR gates?

Yes

Is it possible to build a full adder from one 2-to-1 multiplexer and one XOR gate?

No

Submitted answer 5

Submitted at 2023-05-06 15:48:23 (CDT) 100%

hide

Is it possible to build a full adder from one XOR gate, three AND gates, and one OR gate?

Yes 100%

Is it possible to build a full adder from nine NAND gates?

Yes 100%

Is it possible to build a full adder from nine NOR gates?

Yes 100%

Is it possible to build a full adder from ten XOR gates?

No 100%

Is it possible to build a full adder from two 4-to-1 multiplexers?

Yes 100%

Is it possible to build a full adder from one 8-to-1 multiplexer?

No 100%

Is it possible to build a full adder from one 3-to-8 decoder and two OR gates?

Yes 100%

Is it possible to build a full adder from one 3-to-8 decoder and two XOR gates?

Yes 100%

Is it possible to build a full adder from one 2-to-1 multiplexer and one XOR gate?

No 100%

Submitted answer 4
Submitted at 2023-05-06 15:47:15 (CDT) invalid, not gradable

show

Submitted answer 3
Submitted at 2023-05-06 15:44:58 (CDT) invalid, not gradable

show

Show/hide older submissions

Exam 4

Assessment overview

Question

Total points: 10 /10

Auto-graded question

You are viewing the question instance of a different user and so are not authorized to report an error.

Previous question Next question

Personal Notes

No attached notes

Notes can't be added or deleted because the assessment is closed.

Question 7: Divisible by 3: Part 0

This is the first part of a three-part problem. You should complete this part first.

For this problem, you will design a bit-sliced unit to test whether an integer P given in *n*-bit unsigned representation is divisible by 3. Your divisible-by-3 unit should output a two-bit value, denoted UV, with the meaning given by the table below.

UV	**Meaning**
00	P is divisible by 3
01	P mod 3 = 1
10	P mod 3 = 2
11	Not used

What should be the output of the divisible-by-3 unit for the following values of P?

UV when P = 00:	00	100%
UV when P = 01:	01	100%
UV when P = 10:	10	100%
UV when P = 11:	00	100%
UV when P = 100:	01	100%
UV when P = 101:	10	100%
UV when P = 110:	00	100%

This question is complete and cannot be answered again.

Correct answer

Submitted answer
Submitted at 2023-05-06 15:51:03 (CDT) 100%

hide

Exam 4

Assessment overview

Question

Total points: 10 /10

Auto-graded question

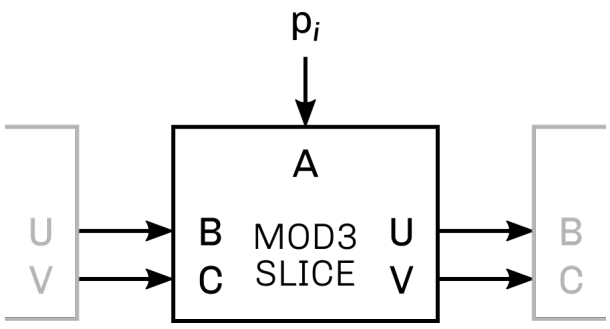
You are viewing the question instance of a different user and so are not authorized to report an error.

Question 8: Divisible by 3: Part 1

This is the second part of a three-part problem. You should complete first part first.

For this problem, you will design a bit-sliced unit to test whether an integer P given in *n*-bit unsigned representation is divisible by 3. Your divisible-by-3 unit should output a two-bit value, denoted UV, with the meaning given by the table below on the right.

Each bit slice takes a bit p_i of P = $p_{n-1}...p_1p_0$, provided as input A to the bit slice, and the two-bit output of the bit slice to the left as inputs B and C. The most significant bit p_{n-1} is the input to the leftmost bit slice and least significant bit p_0 is the input to the rightmost bit slice.



UV	Meaning
00	P is divisible by 3
01	P mod 3 = 1
10	P mod 3 = 2
11	Not used

Give a **minimal SOP** expression for the bit slice output variable U, a **minimal POS** expression for the bit slice output variable V, and the initial values for B and C to be provided as the first bit slice's inputs B and C. Note that U and V must be functions of A, B, and C only.

Hint: Check your bit-sliced circuit with several test values to make sure that it produces the expected values of U and V at the least-significant bit position.

Syntax: When entering Boolean expressions, use ' (single quote) after a variable to represent negation, + (plus) to represent OR, and * (asterisk) or juxtaposition to represent AND. For example, enter the expression $(\mathrm{A}+\bar{\mathrm{B}})\mathrm{C}$ as `(A+B')C`.

U = A'C+AB 100%

V = C'(A'+B')(A+B) 100%

B_0 = 0 100%

C_0 = 0 100%

This question is complete and cannot be answered again.

Correct answer

Solution not shown for this problem.

Submitted answer 3
Submitted at 2023-05-06 16:01:05 (CDT) 100%

hide

U = `A'C+AB` 100%

V = `C'(A'+B')(A+B)` 100%

B_0 = `0` 100%

C_0 = `0` 100%

Submitted answer 2
Submitted at 2023-05-06 15:59:11 (CDT) invalid, not gradable

show

Question 9: Divisible by 3: Part 2

This is the final part of a three-part problem. You should complete part0 and part1 before continuing.

Design a Moore FSM that takes one bit as input and outputs 1 if and only if the sequence of bits seen so far, taken as an integer in unsigned representation with the most significant bit given first, is divisible by 3. An example of an input and output sequence is shown below.

IN: 1001101110100010000
OUT: 0001000000111100000

Note that the output during a given cycle reflects whether the input sequence seen up to and including the *preceeding* cycle is an integer in unsigned representation that is divisible by 3. For example, the first six output bits shown above (000100) are calculated from the input bits (100110) as follows:

IN	Bits so far	Dec	OUT (next cycle)
1	1	1	0
0	10	2	0
0	100	4	0
1	1001	9	1
1	10011	19	0
0	100110	38	0

Your solution should have no more than 4 states.

Use variable A to denote the input bit and variables B and C to denote the state. The output of the FSM should be 1 if the input sequence is divible by 3, and 0 if the input sequence is not divisible by 3.

Give Boolean expression for the next state variables B^+ and C^+, the initial state B_0C_0 (2 bits), and a Boolean expression for the output. The output expression must be a function of B and C only. Boolean expressions do not need to be optimal.

Syntax: Use ' (single quote) after a variable to represent negation, + (plus) to represent OR, and * (asterisk) or juxtaposition to represent AND. For example, enter the expression $A\bar{B} + C$ as `AB' + C`.

B^+ =	A'C+AB	100%
C^+ =	C'(A'+B')(A+B)	100%
B_0C_0 =	00	100%
OUT =	B'C'	100%

This question is complete and cannot be answered again.

Correct answer

Solution not shown for this problem.

Submitted answer 4
Submitted at 2023-05-06 16:14:32 (CDT) 100%

hide

B^+ = `A'C+AB` 100%

C^+ = `C'(A'+B')(A+B)` 100%

B_0C_0 = `00` 100%

OUT = `B'C'` 100%

Note: There are multiple correct solutions, and your score for the problem is based *only* on the output of your FSM as defined by the equations above. Partial credit is awarded based on how well your FSM agrees with the correct output, *not* on whether a given equation is part of some correct solution.

Submitted answer 3
Submitted at 2023-05-06 16:13:16 (CDT) invalid, not gradable

show

Submitted answer 2
Submitted at 2023-05-06 16:08:09 (CDT) invalid, not gradable

show

Show/hide older submissions

Exam 4

Assessment overview

Question

Total points: 10 /10

Auto-graded question

You are viewing the question instance of a different user and so are not authorized to report an error.

Previous question Next question

Personal Notes

No attached notes

Notes can't be added or deleted because the assessment is closed.

Staff information

Staff user:

Ujjal Kumar Bhowmik ubhowmik@illinois.edu

▸ Student details

Question:

QID: Exams/Final/mod3-pt2
Title: Divisible by 3: Part 2
Started at: 2023-05-06 15:43:02 (CDT)
Duration: 00:13:27.403814

Show/Hide answer

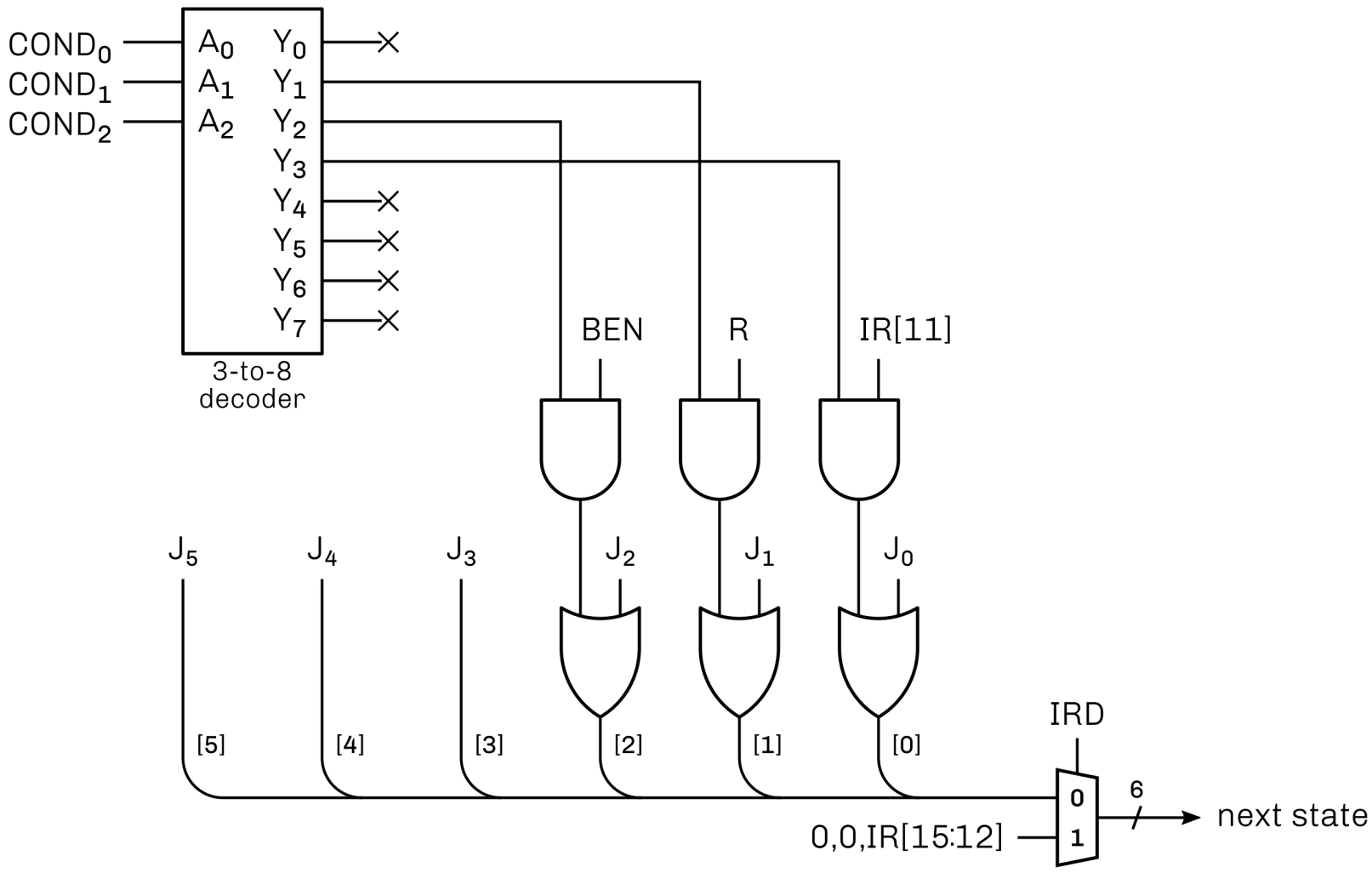
Assessment Instance:

Question 10: Finding LC-3 Microsequencer Sequencing bits (J, COND, IRD)

To answer the following questions, use the LC-3 microsequencer (given below) and LC-3 state diagram (LC-3 Handout)

LC-3 Microsequencer

The following diagram shows the LC-3 microsequencer used in this class. It is functionally equivalent to Figure C.5 in the Patt & Patel textbook.



For LC-3 State Diagram, you may also consult the LC-3 Handout.

State 26

For the following questions, enter the sequencing bits for state 26.

Select the value of the IRD field (1 bit) for state 26:

○ X ◉ 0 ○ 1 100%

Select the value of the COND field (3 bits) for state 26:

○ XXX ◉ 000 ○ 001 ○ 010 ○ 011 100%

Enter the value of the J field (6 bits) in **binary** for state 26:

J: 011001 100%

State 24

For the following questions, enter the sequencing bits for state 24.

Select the value of the IRD field (1 bit) for state 24:

○ X ◉ 0 ○ 1 100%

Select the value of the COND field (3 bits) for state 24:

○ XXX ○ 000 ◉ 001 ○ 010 ○ 011 100%

Enter the value of the J field (6 bits) in **binary** for state 24:

J: 011000 100%

State 4

For the following questions, enter the sequencing bits for state 4.

Select the value of the IRD field (1 bit) for state 4:

○ X ◉ 0 ☐ ○ 1 100%

Select the value of the COND field (3 bits) for state 4:

○ XXX ○ 000 ○ 001 ○ 010 ◉ 011 ☐ 100%

Enter the value of the J field (6 bits) in **binary** for state 4:

J: 010100 100%

This question is complete and cannot be answered again.

Correct answer

State 26

Select the value of the IRD field (1 bit) for state 26:

0

Select the value of the COND field (3 bits) for state 26:

000

J: `011001`

State 24

For the following questions, enter the sequencing bits for state 24.

Select the value of the IRD field (1 bit) for state 24:

0

Select the value of the COND field (3 bits) for state 24:

001

J: `011000`

State 4

For the following questions, enter the sequencing bits for state 4.

Select the value of the IRD field (1 bit) for state 4:

0

Select the value of the COND field (3 bits) for state 4:

011

J: `010100`

Submitted answer 9
Submitted at 2023-05-06 16:22:19 (CDT) 100%

hide

State 26

Select the value of the IRD field (1 bit) for state 26:

0 100%

Select the value of the COND field (3 bits) for state 26:

000 100%

J: `011001` 100%

State 24

For the following questions, enter the sequencing bits for state 24.

Select the value of the IRD field (1 bit) for state 24:

0 100%

Select the value of the COND field (3 bits) for state 24:

001 100%

J: `011000` 100%

State 4

For the following questions, enter the sequencing bits for state 4.

Select the value of the IRD field (1 bit) for state 4:

0 100%

Select the value of the COND field (3 bits) for state 4:

011 100%

J: `010100` 100%

Submitted answer 8
Submitted at 2023-05-06 16:21:19 (CDT) invalid, not gradable

show

Submitted answer 7
Submitted at 2023-05-06 16:21:05 (CDT) invalid, not gradable

show

Show/hide older submissions

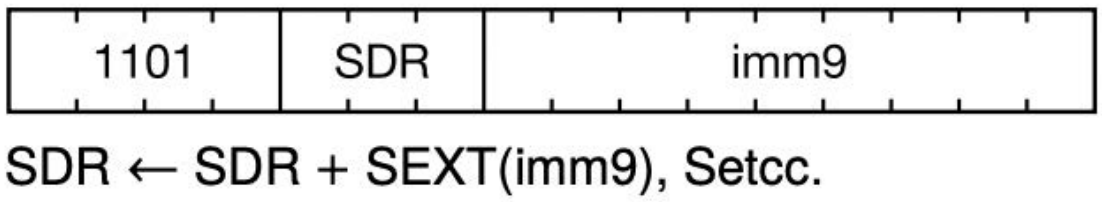
Exam 4

Assessment overview

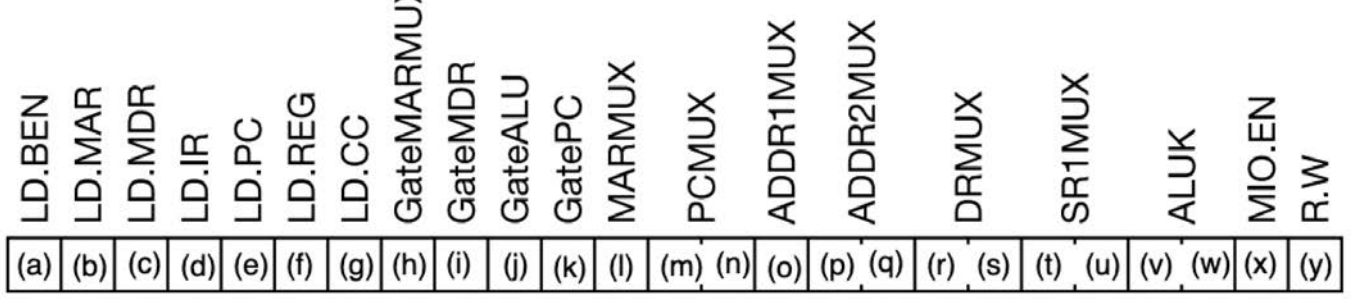
Question 11: Extending the LC-3

The LC-3 ADD instruction has two forms: one where both operands are in registers, and another where the second operand is a 5-bit two's complement immediate value. For this problem, you will add a new instruction, Big Add, or BAD for short, using the reserved opcode 1101. The first operand of the new instruction is a register, and the second operating a 9-bit two's complement value. To fit a 9-bit immediate value into a 16-bit instruction word, the source and destination registers are the same.

The instruction has the following format and execute stage behavior:



Write down the control word for the execute state of the BAD instruction, and then enter the signal values into the online exam form. Your control word should implement the behavior given in RTL above. Signals that may be either 0 or 1 must be entered as X (don't care).



(a) =	0	100%
(b) =	0	100%
(c) =	0	100%
(d) =	0	100%
(e) =	0	100%
(f) =	1	100%
(g) =	1	100%
(h) =	1	100%
(i) =	0	100%
(j) =	0	100%
(k) =	0	100%
(l) =	1	100%
(m) =	X	100%
(n) =	X	100%
(o) =	1	100%
(p) =	1	100%

(q) =	0	100%
(r) =	0	100%
(s) =	0	100%
(t) =	0	100%
(u) =	0	100%
(v) =	X	100%
(w) =	X	100%
(x) =	0	100%
(y) =	X	100%

What is the state number of the execute state of the BAD instruction? (Enter decimal number.)

13 100%

What is the next state after the execute state of the BAD instruction? (Enter decimal number.)

18 100%

This question is complete and cannot be answered again.

Correct answer

(a) = 0

(b) = 0

(c) = 0

(d) = 0

(e) = 0

(f) = 1

(g) = 1

(h) = 1

(i) = 0

(j) = 0

(k) = 0

(l) = 1

(m) = X

(n) = X

(o) = 1

(p) = 1

(q) = 0

(r) = 0

(s) = 0

(t) = 0

(u) = 0

(v) = X

(w) = X

(x) = 0

(y) = X

What is the state number of the execute state of the BAD instruction? Ans:- 13

What is the next state after the execute state of the BAD instruction?? Ans:- 18

Submitted answer 12
Submitted at 2023-05-06 16:37:42 (CDT) 100%

hide

Submitted answer 11
Submitted at 2023-05-06 16:33:45 (CDT) invalid, not gradable

show

Submitted answer 10
Submitted at 2023-05-06 16:33:11 (CDT) invalid, not gradable

show

Show/hide older submissions

Exam 4

Assessment overview

Question

Total points: 10 /10

Auto-graded question

You are viewing the question instance of a different user and so are not authorized to report an error.

Previous question Next question

Personal Notes

No attached notes

Notes can't be added or deleted because the assessment is closed.

Staff information

Staff user:

Ujjal Kumar Bhowmik ubhowmik@illinois.edu

▸ Student details

Question 12: Take Another Load Off Your Shoulders

The LC-3 LDI instruction is not strictly necessary and can be replaced by a sequence of two or more other instructions. In this problem, you will write an instruction sequence that has the same effect as an LDI instruction without using that instruction. **Note:** This is *not* the same question that you had on Homework 12, which asked you to find an instruction sequence that is equivalent to the LD instruction.

Enter your answers in **hexadecimal**. You may use up to 5 instructions. Fill unused instruction slots with NOP instructions (x0000). Your answer will be graded by executing the instruction sequence starting at the first address (x3800) until the instruction at the last address (x3804) finishes executing.

Give an instruction sequence that is equivalent to an instruction **`LDI R5, #4`** placed at address x3800 in memory. Your instruction sequence must have the same effect on the contents of registers R0–R7, CC, and memory as the original LDI instruction. Your solution **must not** execute an LDI instruction.

x3800:	x2A04	100%
x3801:	x6B40	100%
x3802:	x0000	100%
x3803:	x0000	100%
x3804:	x0000	100%

You may consult the LC-3 Handout.

This question is complete and cannot be answered again.

Correct answer

Solution not shown for this problem.

Submitted answer 2
Submitted at 2023-05-06 16:44:24 (CDT) 100%

hide

x3800: **x2A04** 100%

x3801: **x6B40** 100%

x3802: **x0000** 100%

x3803: **x0000** 100%

x3804: **x0000** 100%

You may consult the LC-3 Handout.

Submitted answer 1
Submitted at 2023-05-06 16:42:34 (CDT) invalid, not gradable

show

Exam 4

Assessment overview

Question 13: Analyzing an LC-3 Assembly Language Program: Parity

The following assembly language program is intended to determine the parity of the 16 bits in R7, however several instructions are missing. Your task is to fill in the the missing instructions based on the comments. The comments on each line describe what the instruction on that line should do, except for the comment on line 7, which is a bit vague.

```
1:                                      ; set R6 to zero
2:                                      ; test if R7 is neg, zero, or pos
3:    LOOP   BRzp     SKIP              ; skip next instr if bit[15] is zero
4:           ADD      R6, R6, #1        ; increment R6
5:    SKIP                             ; shift R7 left
6:           BRnp     LOOP              ; repeat while R7 is not zero
7:                                      ; the important bit?
8:           BRz      EVEN              ; parity is even
9:           BRp      ODD               ; parity is odd
```

Additional information about the missing instructions:

1. The program finishes by branching to EVEN or ODD, depending on whether the parity of the value in R7 on entry into the program was even or odd. Recall that a set of bits has *even parity* if it has an even number of 1 bits, and *odd parity* if the number of 1 bits is odd.
2. The missing instructions do not read from memory or write to memory.
3. The programs uses R6 to hold an intermediate value.
4. Register R7 does not hold the original value when the program completes.

Line 1:	AND R6, R6, #0	100%
Line 2:	ADD R7, R7, #0	100%
Line 5:	ADD R7, R7, R7	100%
Line 7:	AND R6, R6, #1	100%

This question is complete and cannot be answered again.

Correct answer

```
1:           AND      R6, R6, #0       ; set R6 to zero
2:           AND      R7, R7, #-1      ; test if R7 is neg, zero, or pos
3:    LOOP   BRzp     SKIP             ; skip next instr if bit[15] is zero
4:           ADD      R6, R6, #1       ; increment R6
5:    SKIP   ADD      R7, R7, R7       ; shift R7 left
6:           BRnp     LOOP             ; repeat while R7 is not zero
7:           AND      R6, R6, #1       ; the important bit?
8:           BRz      EVEN             ; parity is even
9:           BRp      ODD              ; parity is odd
```

Submitted answer 4

Submitted at 2023-05-06 16:51:59 (CDT) 100%

hide

Line 1: `AND R6, R6, #0` 100%

Line 2: `ADD R7, R7, #0` 100%

Line 5: `ADD R7, R7, R7` 100%

Line 7: `AND R6, R6, #1` 100%

Submitted answer 3
Submitted at 2023-05-06 16:46:11 (CDT) invalid, not gradable

show

Submitted answer 2
Submitted at 2023-05-06 16:45:46 (CDT) invalid, not gradable

show

Show/hide older submissions

Exam 4

Assessment overview

Question

Total points: 10 /10

Auto-graded question

You are viewing the question instance of a different user and so are not authorized to report an error.

Previous question Next question

Personal Notes

No attached notes

Notes can't be added or deleted because the assessment is closed.

Staff information

Staff user:

Ujjal Kumar Bhowmik ubhowmik@illinois.edu

▸ Student details

Question:

QID: Exams/Final/parity-asm
Title: Analyzing an LC-3 Assembly Language Program: Parity
Started at: 2023-05-06 16:06:13 (CDT)
Duration: 00:07:35.259025

Show/Hide answer

Assessment Instance:

AID: MockTest_FinalExam
Started at: 2023-05-06 15:27:38 (CDT)
Duration: 01:24:21.029963