

# EECS 270 *Final Exam*

## Spring 2023

Name: \_\_\_\_\_ unique name: \_\_\_\_\_

Sign the honor code:

I have neither given nor received aid on this exam nor observed anyone else doing so.

\_\_\_\_\_

---

### NOTES:

1. *The exam is open book and open class notes plus anything you wrote yourself. You may not use any device capable of communication (cell phones, calculators with wireless, etc.)*
2. You have about 120 minutes for the exam total.
3. Some questions may be harder than others. Manage your time wisely.
4. *If you need additional room for an answer, the last 2 pages are blank and you may use them as part of your answer. Just be sure to indicate you've done so on the page of the question.*

1. Fill in each blank or circle the best answer. [13 points, -2 per wrong or blank answer, min 0]

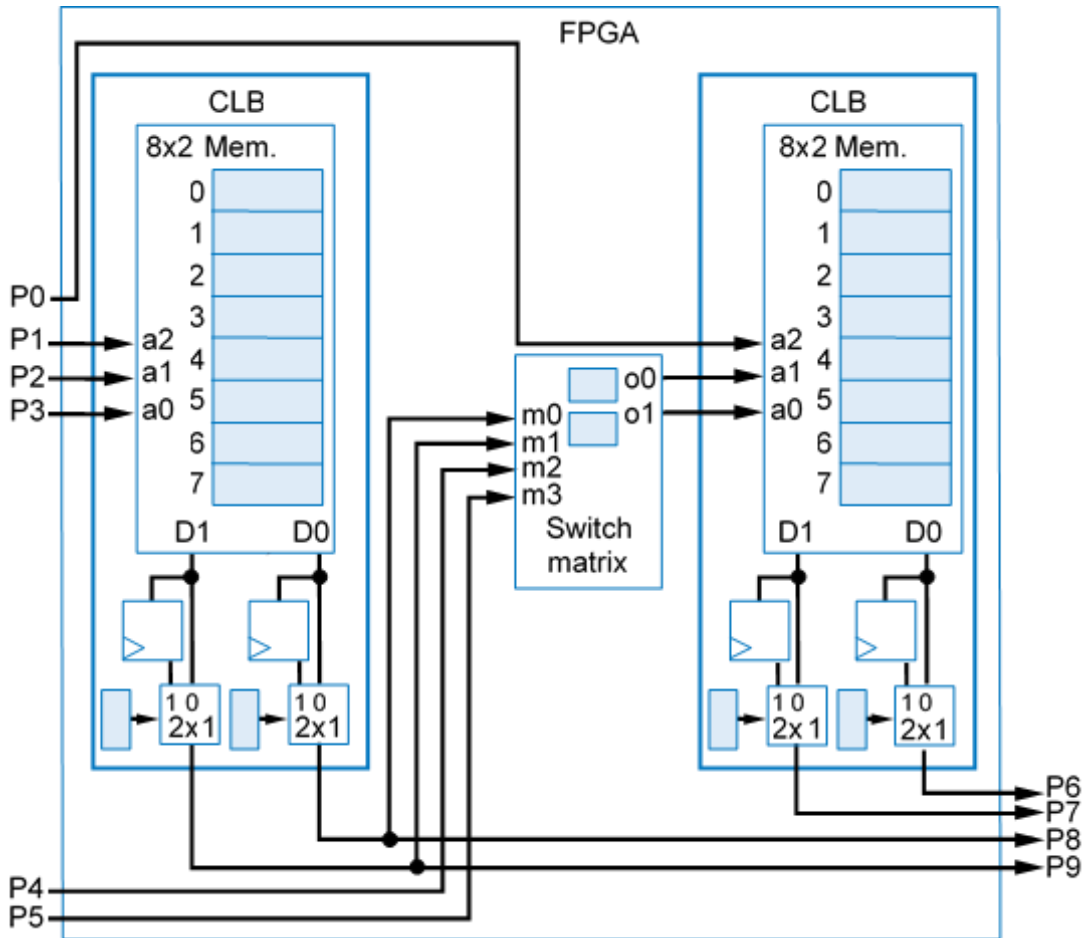
- a) The 5-bit 2's complement representation of -12 is \_\_\_\_\_.
- b) If a given function has A, B, and C as inputs, there are \_\_\_\_\_ maxterms in the equation  $A + !B$ .
- c) You are building a memory where each address references 8 bits of data. You are using a 1024 by 1024 memory array to do so. You need \_\_\_\_\_ address bits, and the column MUX select input requires \_\_\_\_\_ of those address bits.
- d) SRAM cells tend to be larger / smaller than DRAM cells.
- e) "4A.C" base 16 is equal to \_\_\_\_\_ base 10.
- f) If a 4-bit ripple-carry adder (RCA) has a worst-case delay of 1ns, you would expect a 256-bit RCA to have a delay of about 4 / 20 / 60 / 120 / 1024 ns.
- g) If a 4-bit carry look-ahead adder (CLA) has a worst-case delay of 1ns, you would expect a 256-bit cascaded CLA (with a base size of 4-bits, as done in class) to have a delay of about 4 / 20 / 30 / 60 / 1024 ns.
- h) \_\_\_\_\_ is a prime implicant of  $AB + !A!C$  but is not an *essential* prime implicant.

2. Say you have an external signal attached to the D input of a flip-flop. That signal is not synchronized to the local clock. And say that the flip-flop has a set-up time of 2ns, a hold time of 3ns, and a clock-to-Q time of 4ns. [4 points]

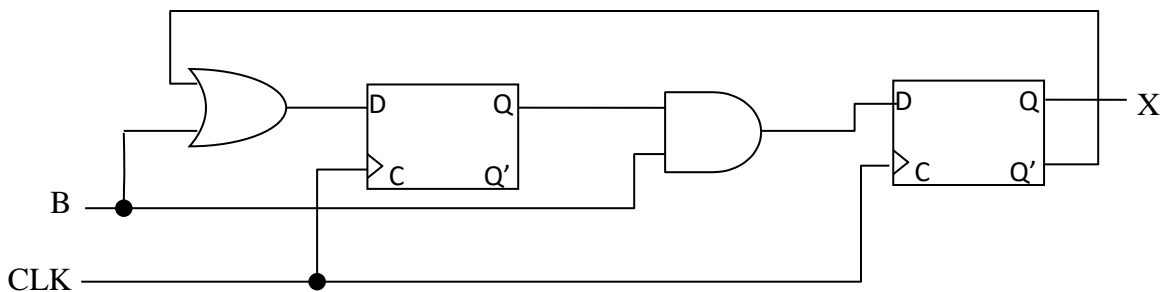
- a) If the D flip-flop's clock is 20 MHz, what is the probability of a timing violation? \_\_\_\_\_
- b) If the D flip-flop's clock is 100MHz, what is the probability of a timing violation? \_\_\_\_\_

3. Using a Kmap, find the minimal product-of-sums for  $A = \Sigma_{(w,x,y,z)}(1, 3, 4, 6, 7, 8, 12, 13) + d(9, 14)$ .  
Clearly show your work. **[6 points]**

P0	P1	P2	P3	P4	P5	P6	P7	P8	P9
			B	B					



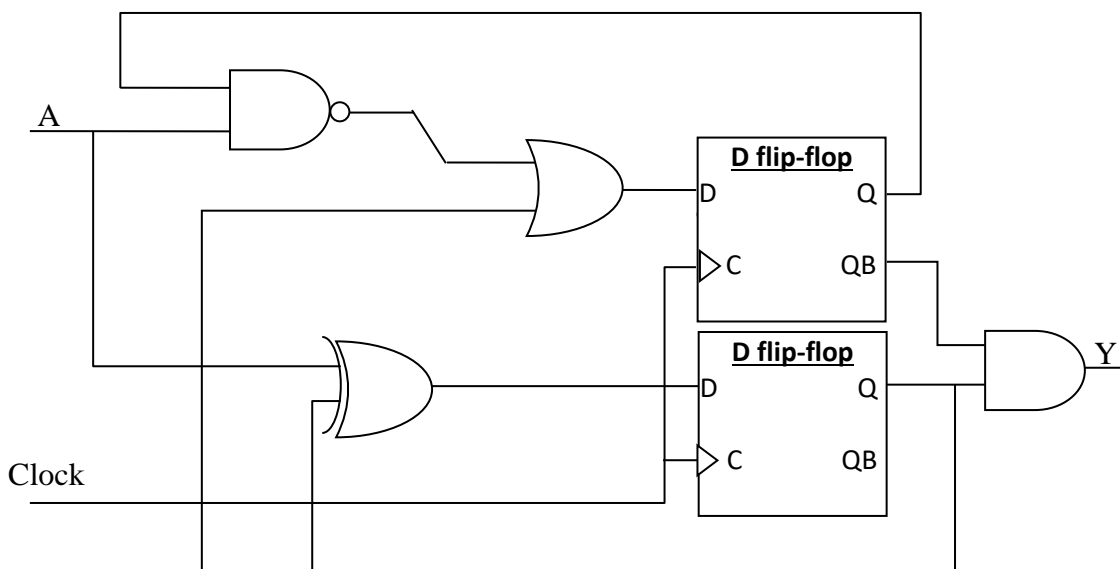
4. Fill in blank boxes as needed to implement the following circuit including the ones on the top. Leave boxes blank if their values don't matter. You may use temporary values to connect outputs back into inputs. Be sure to use the names given (B, X) for inputs and outputs. [11 points]



	Min	Max
<b>OR/AND</b>	2ns	6ns
<b>NOR/NAND</b>	1ns	3ns
<b>NOT</b>	1ns	2ns
<b>XOR/XNOR</b>	4ns	9ns

<b>DFF:</b>		Min	Max
	<i>Clock to Q</i>	1ns	5ns
	<i>Set-up time</i>	4 ns	
	<i>Hold time</i>	6 ns	

5. Assuming the input A *always* arrives 2ns after the rising edge of the clock, answer the following questions. **[10 points]**
- Where do inverter pairs need to be added needed to ensure that there are no hold time violations? Add them in a way that maximized the frequency this circuit can be clocked at. *If there is more than one way to add inverter pairs which maximizes the frequency of the circuit, add as few as possible.* **[5]**
  - What is the lowest clock period that could be safely used to clock this circuit (after taking into account the changes you made above)? Clearly show your work. **[5]**



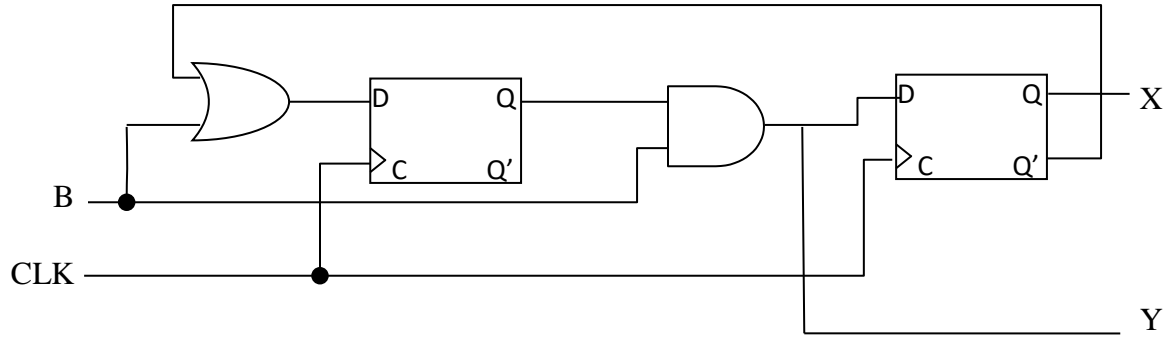
6. Design a Moore-type state-transition diagram which has one input A and one output X. X should go high iff A has been high for at least two cycles (total, not in a row) since either reset (when the FSM starts) or the last time A has been low for two cycles in a row. For example:

A=01010101001001001010101

X=00011111100000000011111

For full credit you must use as few states as possible. **[10 points]**

7. Write a Verilog module, Bob, which implements the following circuit. The inputs are B and Clock. The outputs are X and Y. The code must be correct, clear, and reasonably concise to get full points. **[10 points]**



8. Consider the following state table.

Present state	Next state		Output Z
	x=0	x=1	
A	B	F	0
B	C	D	1
C	A	D	1
D	E	B	1
E	A	B	1
F	C	F	0

Minimize the number of states in this machine and provide a state table the state for your minimized machine. Show your work. **[12 points]**



9. Implement the following circuits. Use the given names for inputs and outputs. **[14 points]**
- a) Provide a circuit which implements the following logic using only 2-input AND/OR/XOR gates and inverters. For full credit you must use as few 2-input gates as possible. All inputs and outputs are one bit. **[4]**

assign B=y?x:!x;

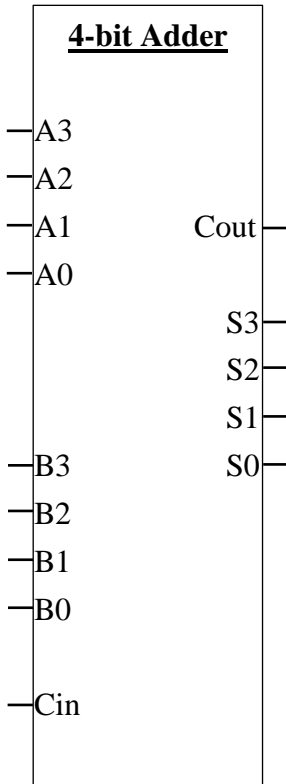
- b) Provide a circuit which implements a half-adder (inputs A, B; outputs S, Cout) using only 2-to-1 MUXes. For full credit you must use as few MUXes as possible. **[5]**
- c) Provide a circuit which implements a 2-to-1 MUX (inputs A, B, and select; outputs X) using only a decoder and one other standard gate (any number of inputs). **[5]**

10. Create a circuit which takes a 4-bit signed-magnitude number and converts it into a 4-bit 2's complement number. You may use a 4-bit adder, a 4-bit 2 to 1 MUX, and up to 3 standard 2-input gates and any number of inverters. The 4-bit adder has been drawn for you. You may freely make use of ground and power. **[10 points]**



*This converter takes its input,  $S[3:0]$  and treats it as a signed-magnitude number.*

*The output,  $T[3:0]$  is a 4-bit 2's complement number.*



*This page is intentionally left blank. See the cover page for an explanation.*

*This page is intentionally left blank. See the cover page for an explanation.*