University of Washington – Computer Science & Engineering

Spring 2022 Instructor: Mark Wyse 2022-05-17

CSE 369 QUIZ 2

Name: _____

Student ID Number: _____

Please do not turn the page until 11:30.

Instructions

- This quiz contains 4 pages, including this cover page.
- Please write your name on and turn in all pages of scratch paper.
- Please clearly indicate (box, circle) your final answer.
- The quiz is closed book and closed notes.
- Please silence and put away all cell phones and other mobile or noise-making devices.
- Remove all hats, headphones, and watches.
- You have 30 minutes to complete this quiz.

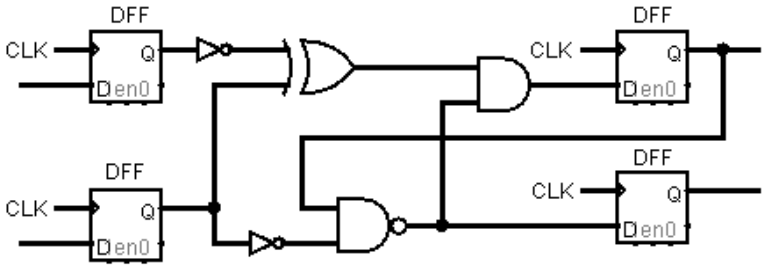
Advice

- Read questions carefully before starting. Read *all* questions first and start where you feel the most confident to maximize the use of your time.
- There may be partial credit for incomplete answers; please show your work.
- Relax. You are here to learn.

Question	Points	Score
(1) SL & Timing	6	
(2) FSM Implementation	10	
(3) FSM Design	12	
Total:	**28**	

Question 1: Sequential Logic & Timing [6 pts]

Consider the following circuit diagram with $t_{\text{setup}} = \mathbf{8\ ns}$ (10^{-9} s), $t_{\text{C2Q}} = \mathbf{12\ ns}$, $t_{\text{NOT}} = \mathbf{4\ ns}$, $t_{\text{XOR}} = \mathbf{16\ ns}$, $t_{\text{NAND}} = \mathbf{8\ ns}$, and $t_{\text{AND}} = \mathbf{12\ ns}$.

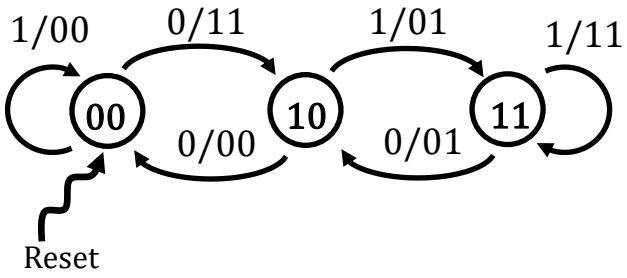


(A) Calculate the **minimum clock period** that will allow the circuit to function correctly. Make sure to *include appropriate units in your answer*. [3 pts]

(B) Calculate the **maximum hold time** that will allow the circuit to function correctly. Make sure to *include appropriate units in your answer*. [3 pts]

Question 2: Finite State Machine Implementation [10 pts]

(A) Fill in the provided truth table based on the FSM shown. [2 pts]



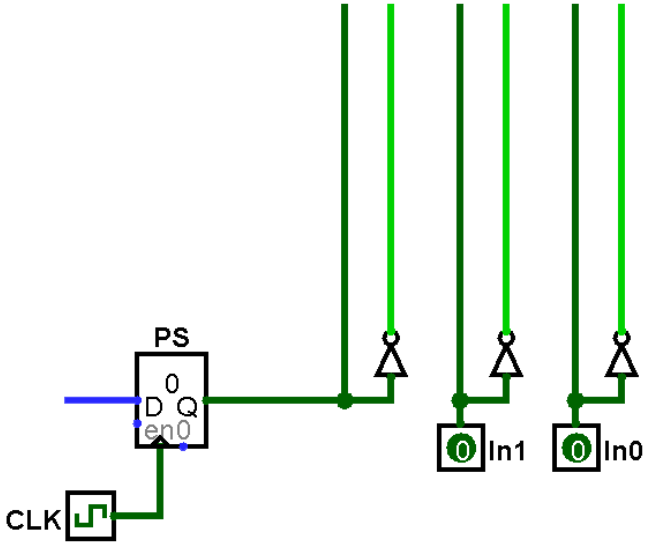
PS_1	PS_0	In	NS_1	NS_0	Out_1	Out_0
0	0	0	1	0	1	1
0	0	1	0	0		
0	1	0	X	X	X	X
0	1	1			X	X
1	0	0	0	0		
1	0	1	1	1	0	1
1	1	0	1	0	0	1
1	1	1			1	1

(B) Complete the circuit diagram below using *minimal logic* based on the truth table shown below. **Show a K-map for each of NS and Out and use only 2-input logic gates in the circuit implementation.** [8 pts]

PS	In_1	In_0	NS	Out
0	0	0	1	X
0	0	1	0	0
0	1	0	1	1
0	1	1	0	0
1	0	0	0	X
1	0	1	1	0
1	1	0	0	X
1	1	1	1	0

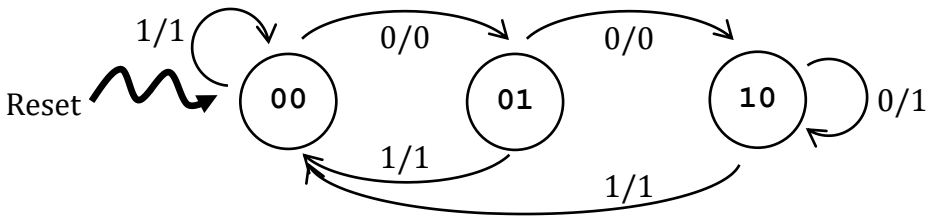
Wire connection:

Wire crossing:



Question 3: Finite State Machine Design [12 pts]

The following FSM is a string manipulator; it outputs a modified version of its stream of inputs:



(A) Assume we pass the following stream of inputs (left-to-right) immediately after resetting the FSM. What is the corresponding stream of outputs? [4 pts]

Input: `0 0 0 1 0 0 0 0`
Output:

(B) As *briefly* as you can, describe how this FSM manipulates its input stream. [3 pts]

(C) Complete the testbench `initial` block to *thoroughly* test the state diagram. You need to fill in all bolded blanks. You are welcome to fill out the Verilog comments to help you keep track of state, but these will not be graded. [5 pts]

```
initial begin
                        Reset <= 1; In <= 1;
     @(posedge clk);
     @(posedge clk);  Reset <= 0;  // state: 00
     @(posedge clk);  In <= ____;  // state: 00
     @(posedge clk);  In <= ____;  // state: ____
     @(posedge clk);  In <= ____;  // state: ____
     @(posedge clk);  In <= ____;  // state: ____
     @(posedge clk);  In <= ____;  // state: ____
     @(posedge clk);  In <= ____;  // state: ____
     @(posedge clk);  $stop();
end
```