University of Washington – Computer Science & Engineering
Winter 2026 Instructor: Naomi Alterman 2026-02-24

CSE 369 QUIZ 2

(VERSION A)

Name: ______________________________

Student ID Number: __________________

Please do not turn the page until 12:20.

Instructions

- This quiz contains **6** pages, including this cover page. You may use the backs of the pages for scratch work.
- 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, smart glasses, watches, and other digital wearables.
- You have 30 (+5) minutes to complete this quiz.

Advice

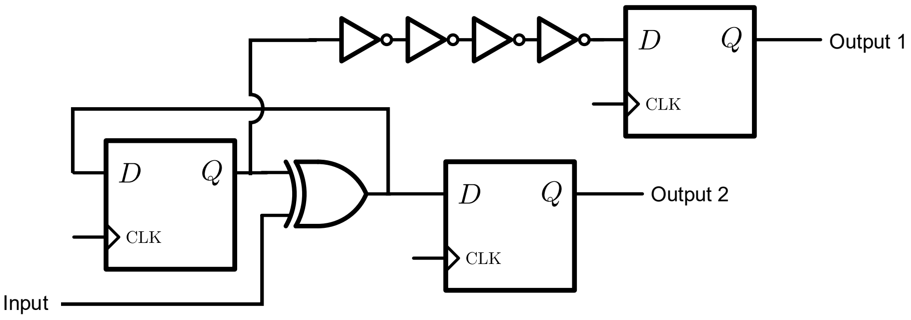
- Read questions carefully before starting. Read *all* questions first and start where you feel the most confident to maximize the use of your time.
- There may be partial credit for incomplete answers; please show your work.
- Relax. If you've been practicing, you got this. If you haven't, you'll learn something now.

Question	Points	Score
(1) SL & Timing	9	
(2) FSM Implementation	14	
(3) FSM Design	7	
Total:	**30**	

(This page left intentionally blank)

Question 1: Sequential Logic & Timing [9 pts]

Consider the following circuit diagram with $t_{setup} = \mathbf{11}$ **ns**, $t_{C2Q} = \mathbf{9}$ **ns**, $t_{NOT} = \mathbf{3}$ **ns**, and $t_{XOR} = \mathbf{10}$ **ns**. Assume that `In` changes **7 ns** after every clock trigger and that there are no timing requirements for `Output` ports. Assume all CLK inputs are connected to the same clock source.



(A) Calculate the **minimum clock period** that will allow the circuit to function correctly. [4 pts]

ns

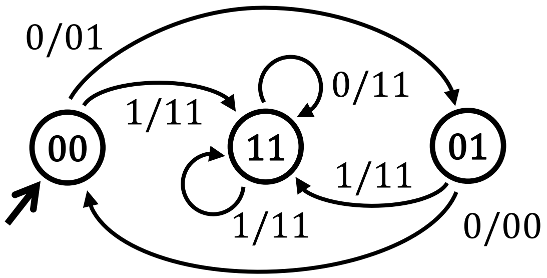
(B) Calculate the **maximum hold time** (t_{hold}) that will allow the circuit to function correctly. [4 pts]

ns

(C) What is the purpose of the inverter chain in the top path of the circuit? [1 pt]

Question 2: Finite State Machine Implementation [14 pts]

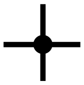
(A) Fill in the provided truth table based on the FSM shown. [8 pts]

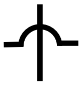


PS_1	PS_0	In	NS_1	NS_0	Out_1	Out_0
0	0	0	0	1	0	1
0	0	1			1	1
0	1	0				
0	1	1	1	1		
1	0	0	x	x	x	x
1	0	1	x	x	x	x
1	1	0				
1	1	1				

(B) Complete the circuit diagram below using *minimal logic* based on the truth table shown below. **Use only 2-input logic gates.** [6 pts]

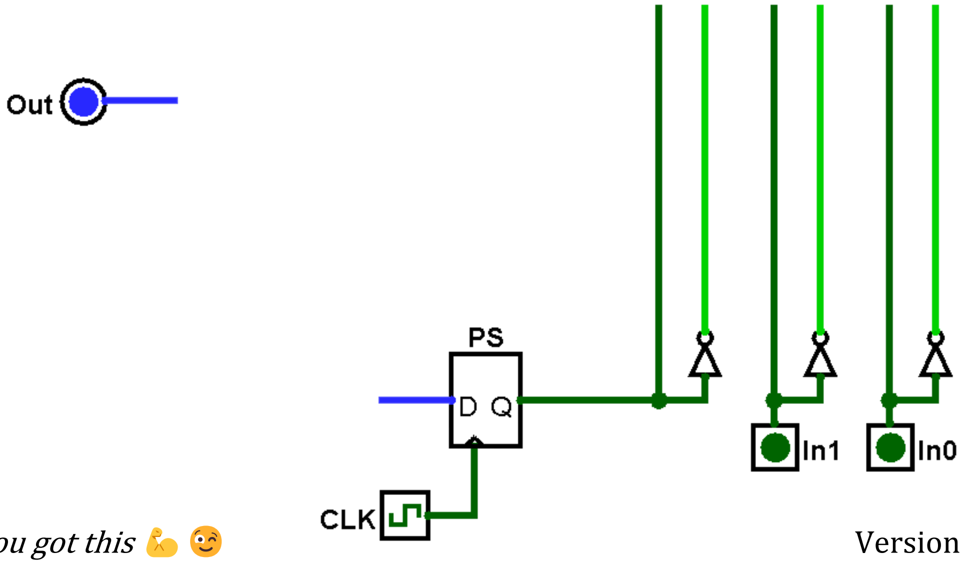
PS	In_1	In_0	NS	Out
0	0	0	x	x
0	0	1	0	1
0	1	0	0	1
0	1	1	1	1
1	0	0	1	0
1	0	1	1	0
1	1	0	1	1
1	1	1	1	1

Wire connection: 

Wire crossing: 

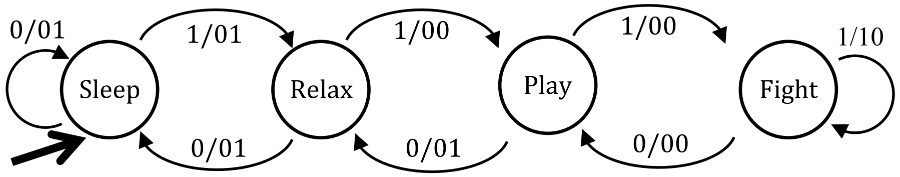
	00	01	11	10
0				
1				

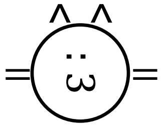
	00	01	11	10
0				
1				



Question 3: Finite State Machine Design [7 pts]

We've built an FSM representing what will happen if we pet Danni the Cat at various times. The input bit represents the presence of your hand petting her. The output bits represent her state of {claws_out, purring}





(A) How many total rows are in the truth table for this FSM? How many of the rows are filled with Don't Cares? [2 pts]

Rows:	Don't Care Rows:

(B) Complete the test bench `initial` block to *thoroughly* test the FSM by filling in all bolded blanks. You may fill out the comments to track the state, but these won't be graded. [4 pts]

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

(C) Based on the two outputs of this FSM alone, what advice would you give to someone who wanted to pet Danni but not get scratched by her claws? [1pt]

(This page left intentionally blank)