

# CSE 369 QUIZ 2

Name: \_\_\_\_\_  
Student ID  
Number: \_\_\_\_\_

**Please do not turn the page until 2:20.**

## Instructions

- This quiz contains **8** 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

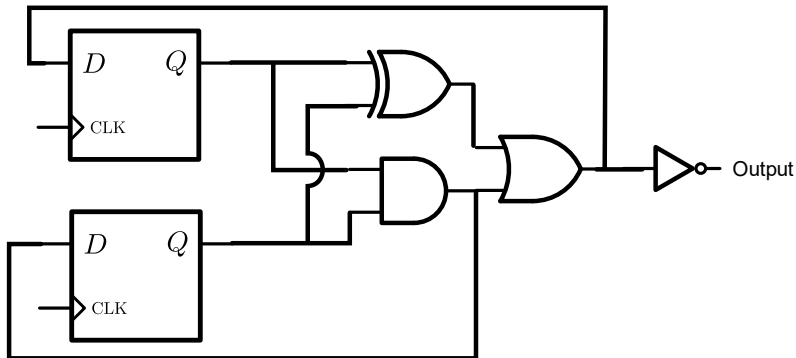
- Read questions carefully before starting. Read *all* questions first and start where you feel the most confident to maximize the use of your time.
- There may be partial credit for incomplete answers; please show your work.
- Relax. If you've been practicing, you got this. If you haven't, you'll learn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

Question	Points	Score
(1) SL & Timing	14	
(2) FSM Implementation	12	
(3) FSM Design	18	
<b>Total:</b>	<b>44</b>	

(This page left intentionally blank)

**Question 1: Sequential Logic & Timing [14 pts]**

Consider the following circuit diagram with  $t_{setup} = 9 \text{ ns}$ ,  $t_{C2Q} = 3 \text{ ns}$ ,  $t_{NOT} = 3 \text{ ns}$ ,  $t_{AND} = t_{OR} = 6 \text{ ns}$  and  $t_{XOR} = 9 \text{ ns}$ . Assume there are no timing requirements for the Output port. Assume all CLK inputs are connected to the same clock source.



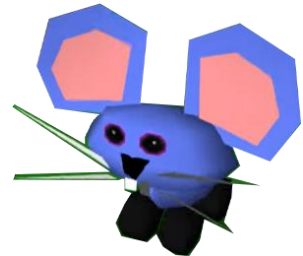
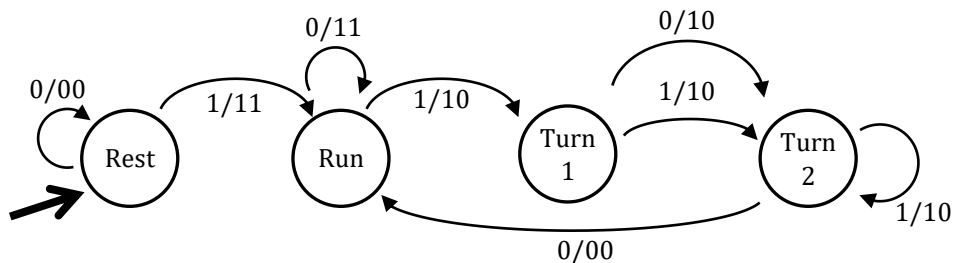
(A) Calculate the **minimum clock period** that will allow the circuit to function correctly. [8 pts]

(B) Calculate the **maximum hold time** ( $t_{hold}$ ) that will allow the circuit to function correctly. [6 pts]



### Question 3: Finite State Machine Design and Testing [18 pts]

We're building a little robot mouse toy for Danni the Cat to play with. Its behavior is controlled by an FSM. The input bit "HIT" represents when it's being hit by her paw. The two output bits {"WHEEL\_L", "WHEEL\_R"} control the two wheels the mouse rides around on.



- (A) Complete the test bench initial block to *thoroughly* test the FSM by filling in the blanks. You may not need all blanks (just leave them empty). You may fill out the comments to track the state, but these won't be graded. [8 pts]

```

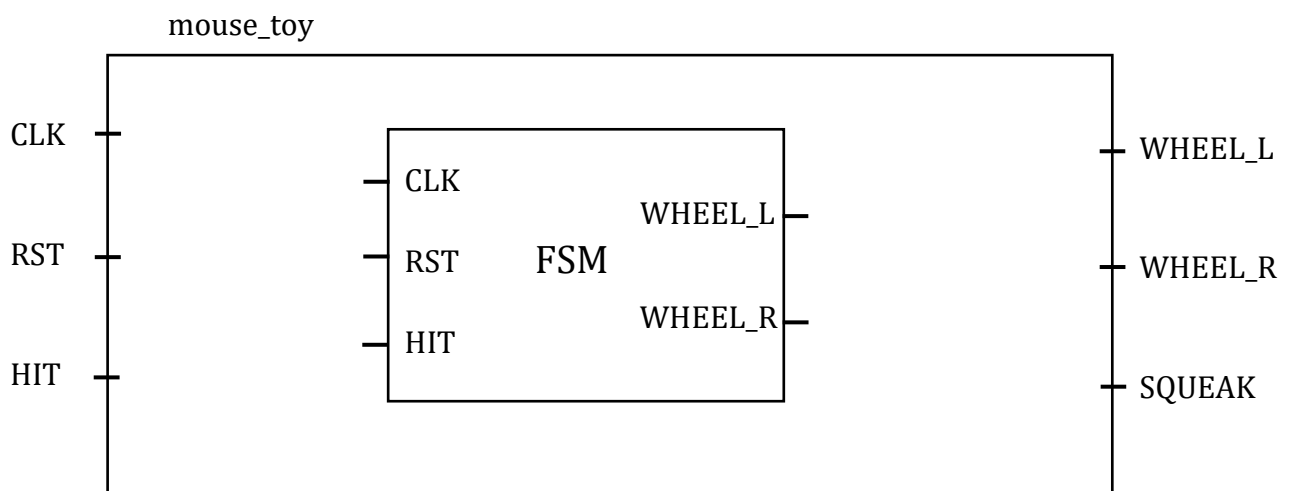
initial begin
    reset = 1; In = 0;

    @(posedge clk); reset = 0; // present-state: rest
    @(posedge clk); In = 1; // present-state: rest
    @(posedge clk); In = 0; // present-state: run
    @(posedge clk); In = 1; // present-state: run
    @(posedge clk); In = 0; // present-state: turn 1
    @(posedge clk); In = 0; // present-state: turn 2
    @(posedge clk); In = ___; // present-state: run
    @(posedge clk); In = ___; // present-state: ___
    @(posedge clk); In = ___; // present-state: ___
    @(posedge clk); In = ___; // present-state: ___
    @(posedge clk); In = ___; // present-state: ___
    @(posedge clk); In = ___; // present-state: ___
    @(posedge clk); In = ___; // present-state: ___
    @(posedge clk); In = ___; // present-state: ___
    @(posedge clk); In = ___; // present-state: ___
    $stop();
end
  
```

(B) What is the minimum number of flip flops we need to store the state of this FSM? [1 pt]

(C) Which state cannot be returned to once we've transitioned out of it? What are the implications this has for testing it? [3 pt]

(D) Complete the block diagram below to show how our FSM connects to the other toy components. The external "hit" sensor is **active low** and the wheel motors are **active high**. There is also a speaker that squeaks when powered (also **active high**). The mouse should squeak whenever it is in motion. [6 pt]



(Use this page for scratch work)

*Remember, you got this* 🦵 😊

(Use this page for scratch work)

*Remember, you got this* 🦵 😊