University of Washington – Computer Science & Engineering

Winter 2024 Instructor: Justin Hsia 2024-02-20

CSE 369 QUIZ 2

Name: _Molly_Model_____

Student ID Number: _1234567_____

Please do not turn the page until 12:20.

Instructions

- This quiz contains 4 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, and watches.
- You have 30 (+5) minutes to complete this quiz.

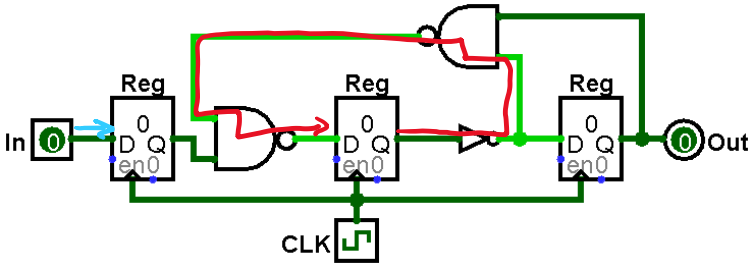
Advice

- 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. You are here to learn.

Question	Points	Score
(1) SL & Timing	6	6
(2) FSM Implementation	10	10
(3) FSM Design	11	11
Total:	**27**	27

Question 1: Sequential Logic & Timing [6 pts]

Consider the following circuit diagram with $t_{\text{setup}} = \mathbf{8\ ns}$ (10^{-9} s), $t_{\text{C2Q}} = \mathbf{14\ ns}$, $t_{\text{NAND}} = \mathbf{23\ ns}$, and $t_{\text{NOT}} = 7\ \mathbf{ns}$. Assume that `In` changes **6 ns** after every clock trigger.



(A) Calculate the **minimum clock period** that will allow the circuit to function correctly. [3 pts]

75 ns

The critical path is shown above in red.
We need $t_{\text{C2Q}} + t_{\text{NOT}} + t_{\text{NAND}} + t_{\text{NAND}} \leq t_{\text{period}} - t_{\text{setup}}$.
Then $t_{\text{period}} \geq 14 + 7 + 23 + 23 + 8 = 75$ ns.

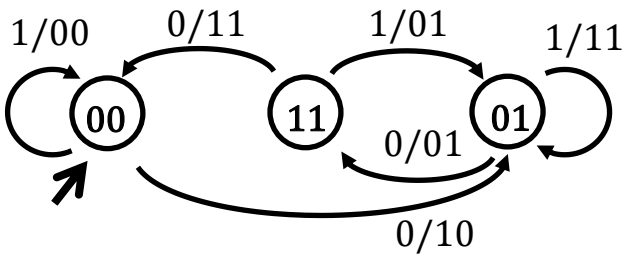
(B) Calculate the **maximum hold time** (t_{hold}) that will allow the circuit to function correctly. [3 pts]

6 ns

The shortest path to a register input is shown above in blue.
We need $t_{\text{In}} \geq t_{\text{hold}}$.
Then $t_{\text{hold}} \leq 6$ ns.

Question 2: Finite State Machine Implementation [10 pts]

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



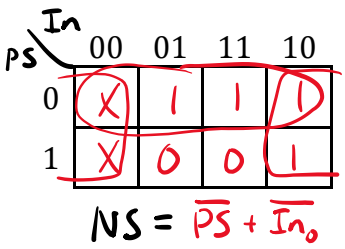
PS_1	PS_0	In	NS_1	NS_0	Out_1	Out_0
0	0	0	**0**	**1**	1	0
0	0	1	0	0	0	0
0	1	0	1	1	**0**	**1**
0	1	1	0	1	1	1
1	0	0	**X**	**X**	X	X
1	0	1	X	X	X	X
1	1	0	0	0	**1**	**1**
1	1	1	0	1	0	1

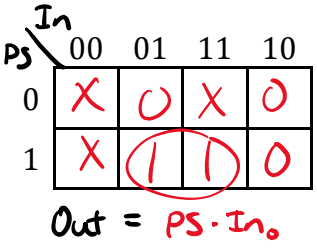
(B) Complete the circuit diagram below using *minimal logic* based on the truth table shown below. **Use only 2-input logic gates.** [8 pts]

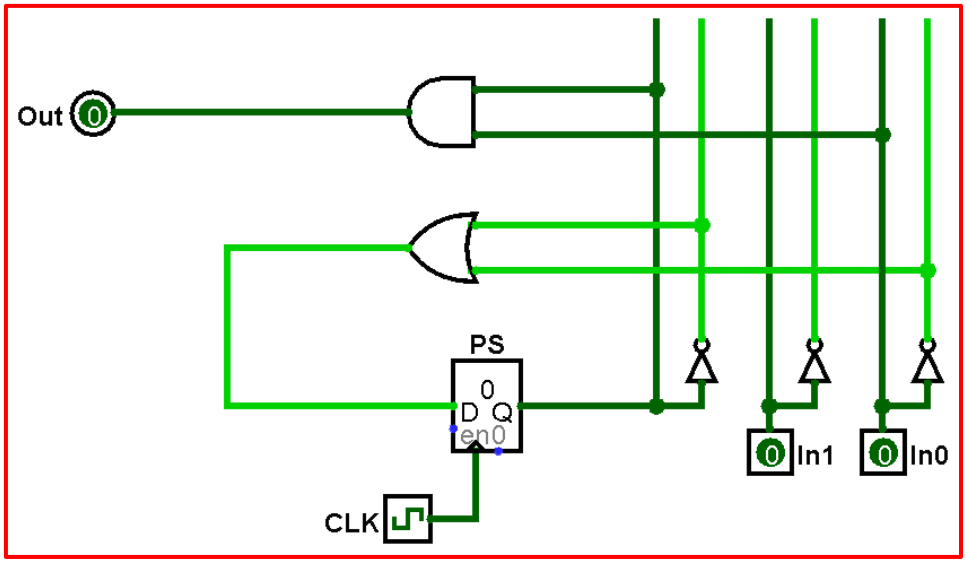
PS	In_1	In_0	NS	Out
0	0	0	X	X
0	0	1	1	0
0	1	0	1	0
0	1	1	1	X
1	0	0	X	X
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

Wire connection:

Wire crossing:

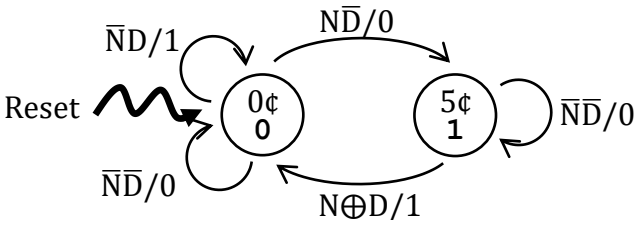






Question 3: Finite State Machine Design [11 pts]

Recall the 10¢ gumball-dispensing, no-change-giving vending machine FSM from lecture that can only take one coin at a time:



(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]

1 state + 2 input bits → $2^3 = 8$ rows in TT.
One missing transition (ND) for each state.

Rows: **8**	Don't Care Rows: **2**

(B) Complete the test bench `initial` block to *thoroughly* test the FSM. 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. Don't worry about situations we don't expect to see during normal operation. [5 pts]

```
initial begin
                          N <= 1;      D <= 0;      // state: 0
      @(posedge clk);     N <= 0___;   D <= 1___;   // state: 1__
      @(posedge clk);     N <= 0;      D <= 0;      // state: 0__
      @(posedge clk);     N <= 1;      D <= 0;      // state: 0__
      @(posedge clk);     N <= 0___;   D <= 0___;   // state: 1__
      @(posedge clk);     N <= 1;      D <= 0;      // state: 1__
      @(posedge clk);     N <= 0;      D <= 1;      // state: 0__
      @(posedge clk);
      $stop();
end
```

(C) If we increase the cost of gumballs at 15¢ and accept quarters (Q: 25¢), draw the new state diagram. The vending machine still does not give change and can only take one coin at a time. No need to assign binary encodings to the states (*i.e.*, only give state names). [4 pts]