University of Washington – Computer Science & Engineering

Winter 2018 Instructor: Justin Hsia 2018-02-20

CSE 369 QUIZ 2

Name: _Perry_Perfect_____

UWNetID: _perfect_____

Please do not turn the page until 10:30.

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 25 minutes to complete this quiz.

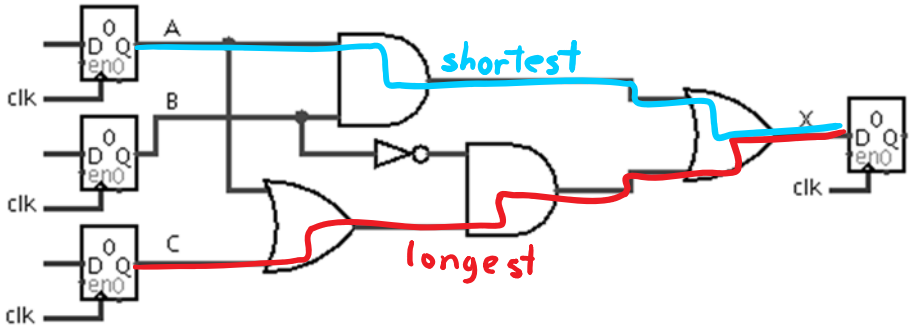
Advice

- 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	8	8
(3) FSM Design	12	12
Total:	**26**	26

Question 1: Sequential Logic & Timing [6 pts]

Consider the following circuit diagram with t_{setup} = **5 ns** (10^{-9} s), t_{C2Q} = **15 ns**, t_{NOT} = **2 ns**, t_{OR} = **10 ns**, and t_{AND} = **10 ns**.



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

$t_{period,min}$ = 50 ns

The critical path is shown above in red and goes through two ORs and one AND gate.
So we need $t_{C2Q} + t_{OR} + t_{AND} + t_{OR} \leq t_{period} - t_{setup}$.
Then $t_{period} \geq 15 + 10 + 10 + 10 + 5 = 50$ ns.

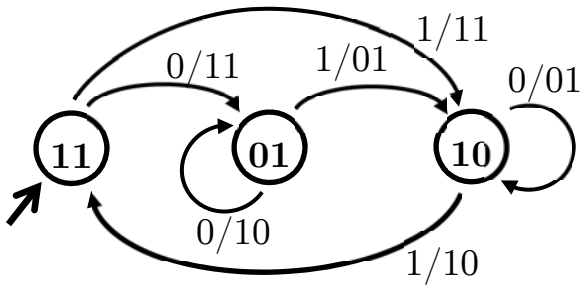
(B) What is the theoretical **maximum hold time** for this circuit to function correctly? Make sure to *include units*. [3 pts]

$t_{hold,max}$ = 35 ns

Shortest path is shown above in light blue and goes through one AND and one OR gate.
So we need $t_{C2Q} + t_{AND} + t_{OR} \geq t_{hold}$ and $t_{hold} \leq 15 + 10 + 10 = 35$ ns.

Question 2: Finite State Machine Implementation [8 pts]

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



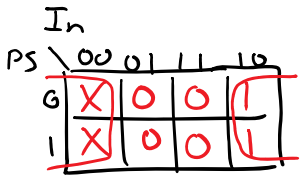
PS_1	PS_0	In	NS_1	NS_0	Out_1	Out_0
0	0	0	**X**	X	X	X
0	0	1	X	**X**	X	X
0	1	0	0	1	**1**	0
0	1	1	1	0	0	**1**
1	0	0	1	0	0	**1**
1	0	1	1	1	**1**	0
1	1	0	0	**1**	1	1
1	1	1	**1**	0	1	1

(B) Complete the circuit diagram below using *minimal logic* based on the truth table shown below. You are welcome to use 2- and 3-input logic gates. [6 pts]

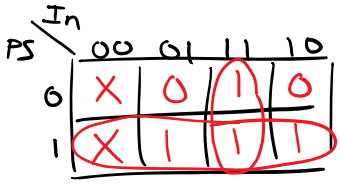
PS	In_1	In_0	NS	Out
0	0	0	X	X
0	0	1	0	0
0	1	0	1	0
0	1	1	0	1
1	0	0	X	X
1	0	1	0	1
1	1	0	1	1
1	1	1	0	1

Wire connection:

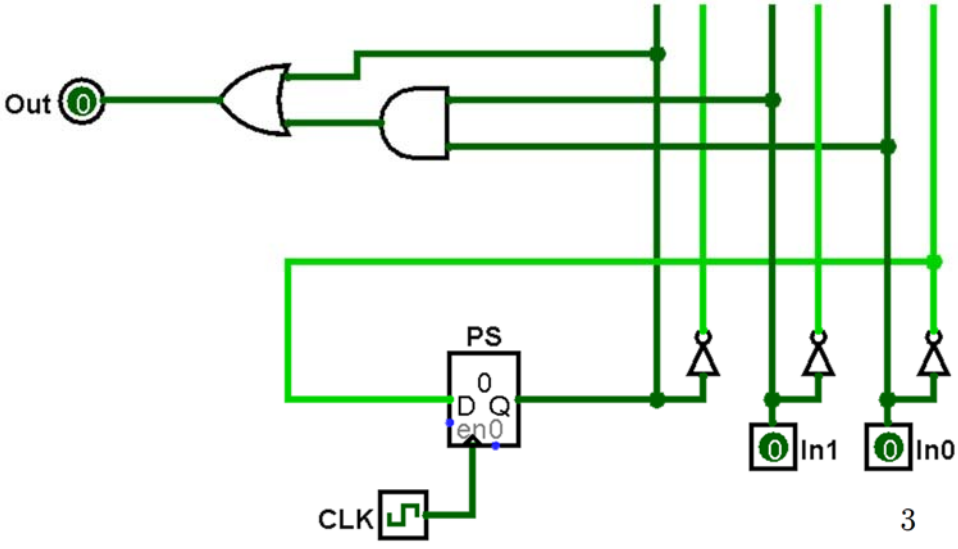
Wire crossing:



$NS = \overline{In_0}$



$Out = PS + In_1 In_0$



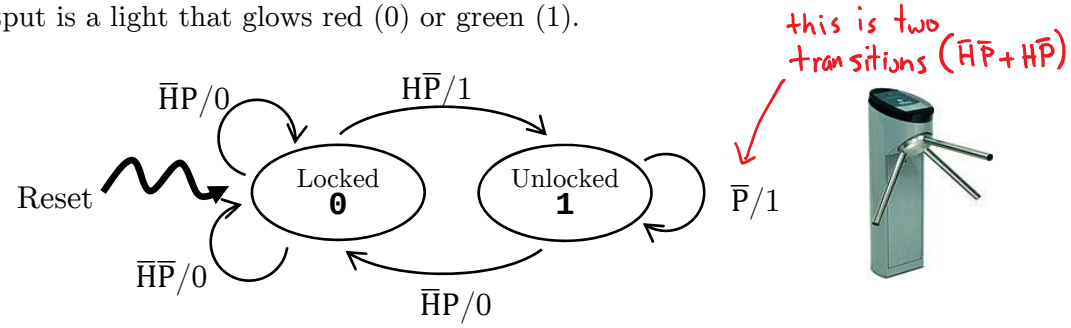
Question 3: Finite State Machine Design [12 pts]

(A) We are designing an FSM for a vending machine that gave no change, dispenses at 20¢, and accepts pennies (1¢), nickels (5¢), and dimes (10¢). [4 pt]

19 states and 3 input bits.

State *Bits*: **5 bits**	Max Transitions Per State: 2^3=**8**

(B) The following FSM represents a *turnstile*, which is locked until someone swipes their Husky ID (input H) and then locks once you push through (input P) the unlocked gate. The output is a light that glows red (0) or green (1).



Complete the testbench `initial` block to *thoroughly* test the state diagram. Even though they may be unnecessary, please fill in all blanks. You are welcome to fill out the Verilog comments to help you keep track of state, but these will not be graded. [6 pts]

```
initial begin
                        H <= _0_;   P <= _0_;   // state:  0
     @(posedge clk);    H <= 1;     P <= 0;     // state: _0_
     @(posedge clk);    H <= _1_;   P <= _0_;   // state: _1_
     @(posedge clk);    H <= 0;     P <= 0;     // state: _1_
     @(posedge clk);    H <= _0_;   P <= _1_;   // state: _1_
     @(posedge clk);    H <= 0;     P <= 1;     // state: _0_
     @(posedge clk);
     $stop();
end
```

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

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

1 state bit + 2 input bits → $2^3 = 8$ rows in truth table.

6 transitions shown/covered in state diagram, so remaining are don't cares.