University of Washington – Computer Science & Engineering

Spring 2018   Instructor: Justin Hsia   2018-05-15

CSE 369 QUIZ 2

Name: _____

UWNetID: _____

Please do not turn the page until 10:00.

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	
(2) FSM Implementation	9	
(3) FSM Design	11	
Total:	**26**	

Question 1: Sequential Logic & Timing [6 pts]

Consider the following circuit diagram with $t_{period} = \mathbf{150}$ **ns** $(10^{-9}$ s$)$, $t_{C2Q} = \mathbf{45}$ **ns**, $t_{setup} = \mathbf{15}$ **ns**, and $t_{hold} = \mathbf{10}$ **ns**.
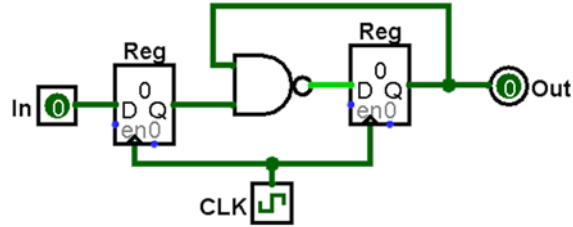
(A)   Assume that `In` does not violate any timing constraints. Calculate the **maximum NAND gate delay** that will allow the circuit to function correctly. Make sure to *include units.* [3 pts]
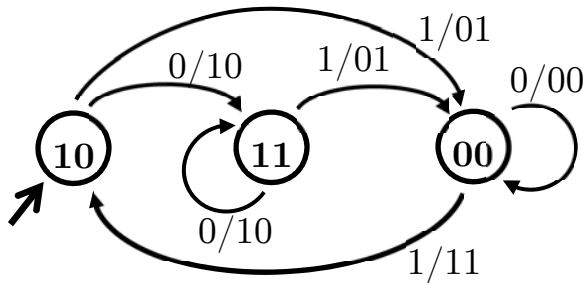
$$t_{NAND,max} = \underline{\hspace{3cm}}$$

(B)   Within what range of times (measured from each clock trigger) will changing the input `In` **not** cause a *timing violation*? Answer using inclusive interval notation: $[t_{start}, t_{end}]$. [3 pts]

$$[\underline{\hspace{1.5cm}} , \underline{\hspace{1.5cm}}] \text{ ns}$$

Question 2: Finite State Machine Implementation [9 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	0	0	
0	0	1	1	0		1
0	1	0	X		X	X
0	1	1		X	X	X
1	0	0		1	1	0
1	0	1	0		0	1
1	1	0	1	1		0
1	1	1	0	0	0	

(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. [7 pts]

PS	In$_1$	In$_0$	NS	Out
0	0	0	0	0
0	0	1	X	X
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	X	X
1	1	0	0	0
1	1	1	1	0

Wire connection:

Wire crossing:

3
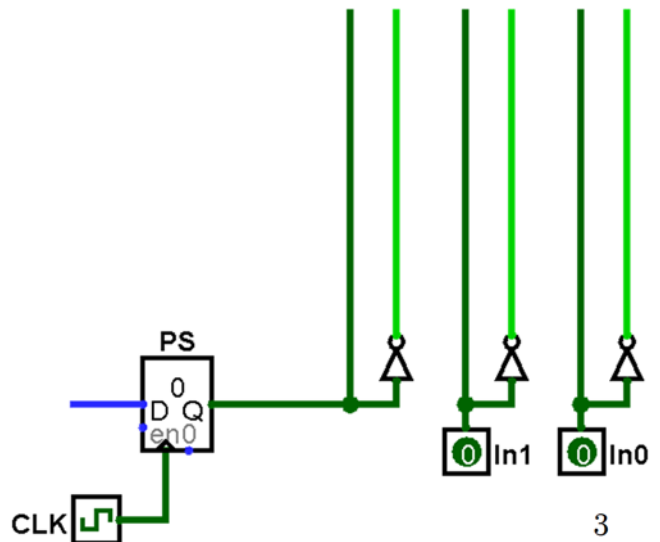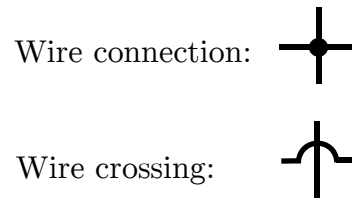
Question 3: Finite State Machine Design [11 pts]

For this problem, consider the FSM below:

0/0   1/0   0/1   0/0   1/0

Reset   (00)   (01)   (10)   (11)

0/0   1/1   1/0
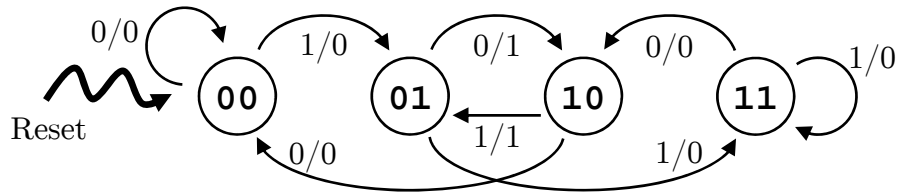
(A)   Answer the following about the corresponding *truth table*. [2 pt]

Rows:	Rows of Don't Cares:

(B)   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. [5 pts]

```
initial begin

                  In <= _____;    // state: 00

  @(posedge clk);  In <= 1;       // state: _____

  @(posedge clk);  In <= 1;       // state: _____

  @(posedge clk);  In <= 1;       // state: _____

  @(posedge clk);  In <= _____;   // state: _____

  @(posedge clk);  In <= _____;   // state: _____

  @(posedge clk);  In <= _____;   // state: _____

  @(posedge clk);  In <= _____;   // state: _____

  @(posedge clk);                 // state: _____

  $stop();
end
```

(C)   What two input sequences does this FSM "recognize" (*i.e.* when it outputs a 1)? [4 pt]