University of Washington – Computer Science & Engineering

Winter 2018 Instructor: Justin Hsia 2018-03-06

CSE 369 QUIZ 3

Name: ______________________________

UWNetID: ______________________________

Please do not turn the page until 10:40.

Instructions

- This quiz contains 4 pages, including this cover page.
- Show scratch work for partial credit, but put your final answers in the boxes and blanks provided.
- 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 35 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) Building Blocks	12	
(2) Shift Registers	9	
(3) Filter	11	
Total:	**32**	

Question 1: Building Blocks [12 pts]

In Lab 5, one landing light sequence (left-to-right) was **100** → **010** → **001** → 100 → ⋯.
Implement this sequence as a **counter** with **Reset** and **Enable** signals using a *minimal number of logic gates* plus *routing elements* discussed in class. Reset to the **100** state.

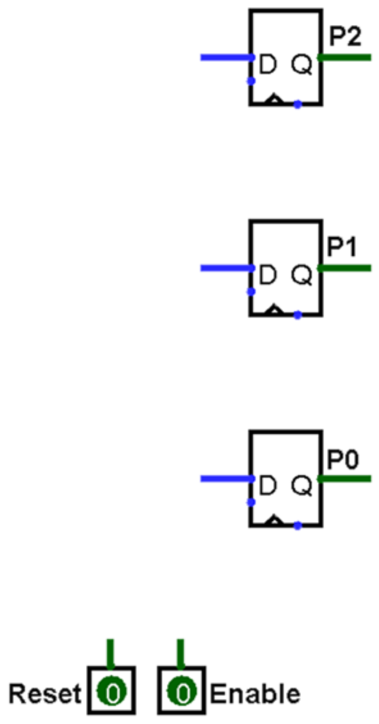
- Make sure you label the corresponding selector bits for ports of routing elements.
- Assume the clock inputs are connected properly for you.

PS_2	**PS_1**	**PS_0**	**NS_2**	**NS_1**	**NS_0**
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

NS_2	00	01	11	10
0				
1				

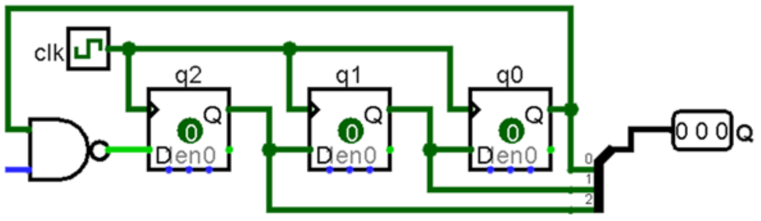
NS_1	00	01	11	10
0				
1				

NS_0	00	01	11	10
0				
1				



Question 2: Shift Registers [9 pts]

(A) We are designing a 3-bit LFSR to use as a pseudo-random number generator, but we only have a 2-input NAND gate available. Assuming we start in state **000**, which state bit (`q2` or `q1`) should we connect along with `q0` to the NAND gate in order to get the longest possible cycle? What is the length of the cycle we end up in? [5 pt]



Bits:	______ and `q0`
Cycle length:	

(B) The Verilog code below is supposed to implement an *enabled* version of the LFSR above with bits `q2` and `q1` connected to the NAND gate. Find errors in the code and rewrite the offending lines in the boxes. [4 pt]

```
module LFSR (Q, enable, reset, clk);
  input        enable, reset, clk;
  output [2:0] Q;

  always_ff @(posedge clk)
    if ( reset )
      Q <= 3'b000;
    else if ( enable )
      Q  = { ~(Q[2] ^ Q[1]), Q[2], Q[0] };

endmodule
```

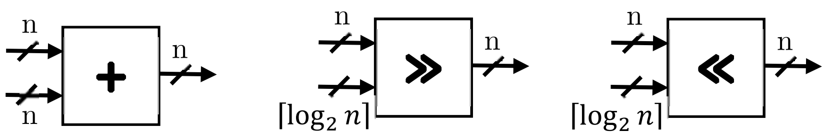
1)

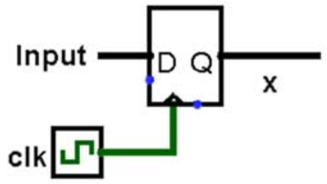
2)

Question 3: Filter [11 pts]

A **boxcar** (or moving average) **filter** computes the average of the last k samples and can be used to smooth out sudden input spikes (*i.e.* a low-pass filter).

(A) Draw a circuit below that computes the moving average of the last 4 inputs, *i.e.* $k = 4$ and $y_i=(x_i+x_{i-1}+x_{i-2}+x_{i-3})/4$, for a signal x, where x_{i-1} is the value of x_i from the previous clock cycle and so on. The given register is there for synchronization; you should *not* connect anything directly to `Input`. You can freely use registers, constants, and the following logic blocks: [8 pts]





(B) Let $t = 0$ be a rising edge of the clock with the signal x_0 on the `Input` bus. If the clock period is 100 ps, $t_{C2Q} = 10$ ps, $t_{add} = 20$ ps, and $t_{shift} = 5$ ps, at what time will y_3 be computed (*i.e.* at what time will y_3's correct value first be known)? *Include units!* [3 pts]

$t_{y3} =$