

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.

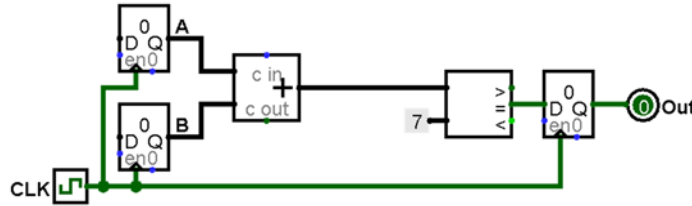
Advice

- 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) Timing Revisited	10	
(2) Building Blocks	12	
(3) Shift Registers	10	
Total:	32	

Question 1: Timing Revisited [10 pts]

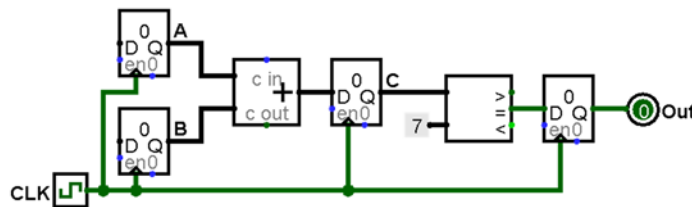
Consider the following circuit diagram with: $t_{setup} = 40$ ps, $t_{hold} = 10$ ps, $t_{c2Q} = 60$ ps, $t_{ADD} = 80$ ps, and $t_{COMP} = 100$ ps. Fill in your answers in the boxes below, *including units*.



- (A) What is the minimum clock period that will ensure proper behavior? [2 pt]

- (B) For a particular set of inputs A_i and B_i , how long does it take to compute the associated output Out_i ? Measure from the moment A and B update to the moment Out updates. You may use the variable t_{period} (answer to part A) in your answer. [2 pt]

Now we add a register between the adder and comparator:



- (C) What is the new minimum clock period that will ensure proper behavior? [2 pt]

- (D) For a particular set of inputs A_i and B_i , how long does it now take to compute the associated output Out_i ? Measure from the moment A and B update to the moment Out updates. You may use the variable t_{period} (answer to part C) in your answer. [2 pt]

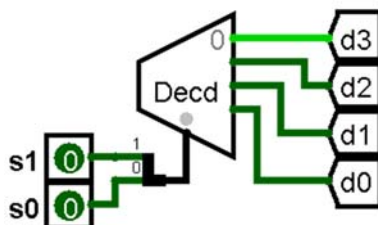
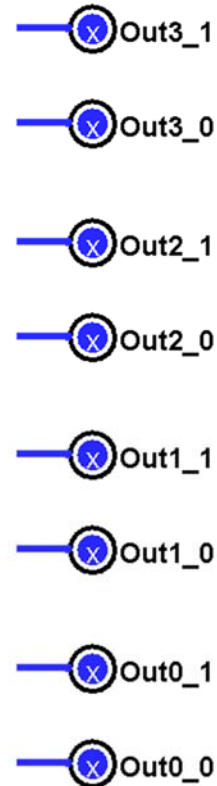
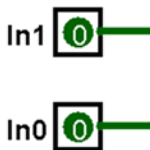
- (E) Does adding this extra register help or hurt? *Explain briefly.* [2 pt]

Question 2: Building Blocks [12 pts]

- (A) Implement a 2:4 binary decoder below using only NOT, AND, and OR gates. The 2 input bits (s_1 and s_0) set the corresponding output bit (one of d_0 through d_3) high. [4 pt]



- (B) Implement a 2-bit, 1-to-4 DEMUX below using only NOT, AND, and OR gates. This passes the 2 input bits onto 1 of 4 sets of outputs ($OutN_x$). Assume you have a working 2:4 binary decoder and *write in the signals d_0 , d_1 , d_2 , and d_3 where needed.* [8 pt]



Question 3: Shift Registers [10 pts]

We have a 4-bit linear feedback shift register (LFSR) that goes through the following state sequence: 0000 → 1000 → 1100 → 0110 → 0011 → 0001 → 1000 → ...

- (A) Circle one: This LFSR is shifting bits to the **LEFT** / **RIGHT**. [1 pt]
- (B) The bit that is shifted in is a function of two of the LFSR bits. *We number the bits starting from 0 increasing from right to left* (like standard binary). What is the name of the gate being used and which two bits are its inputs? **Hint**: all named gates are associative (*i.e.* $F(0,1) = F(1,0)$). [6 pt]

Gate:	
Bits:	_____ and _____

-
- (C) The Verilog code below is supposed to implement a 4-bit parallel-in, serial-out (PISO) shift register. The output is the highest state bit. It will shift in the lowest bit of the input bus. Find errors in the code and rewrite the offending lines in the boxes. [3 pt]

```
module PISO (out, in, load, shift, clk);
    output    out;
    input  [3:0] in;
    input          load, shift, clk;
    wire  [3:0] state;

    always_ff @(posedge clk) begin
        if ( load )
            state = in;
        else if ( shift )
            state <= {in[3], state[2:0]};
        end

    assign out = state[3];

endmodule
```

1)

2)

3)