

Name & Std. No.: Riley Lawson 116555487 **Lab Section:** 6

Date: 11/12/2020

Submission Instructions:

Prelab:

- 1. Complete the prelab**
- 2. Submit this report with the prelab completed to Canvas before your lab starts**

Lab:

- 1. Complete the lab according to the instructions**
- 2. Take screenshots of your ModelSim waveform (note: to receive points your NetID has to be present in the screenshot) and insert them into this document.**
- 3. Include screenshots of any related block design files or Verilog files in the report**
- 4. Complete this report and reupload it to Canvas**

PRELAB:

Complete the prelab and make sure you have your designs and circuit diagrams ready before the lab session. You may refer to your text book, Chapter 6.

Q1. Design a simple counting device (Section 2.0).

Number of States: 6

Number of State Variables: 4

State Table:

Present State	Next State		Output
	w=0	w=1	
A	A	B	0
B	B	C	1
C	C	D	2
D	D	E	3
E	E	F	4
F	F	A	5

State-Assigned Table:

Present State	Next State		Output
	w=0	w=1	
000	000	001	000
001	001	010	001
010	001	100	010
011	001	100	011
100	100	101	100
101	101	000	101

Canonical SOP Expressions for Next State Logic:

$$Y_0 = !w!y_1!y_0 + !w!y_2!y_0 + w!y_1!y_0 + w!y_2!y_0$$

$$Y_1 = !w!y_2!y_1y_0 + w!y_2y_1 + !y_2y_1!y_0$$

$$Y_2 = !w!y_2!y_1y_0 + y_2!y_1!y_0 + wy_2!y_1$$

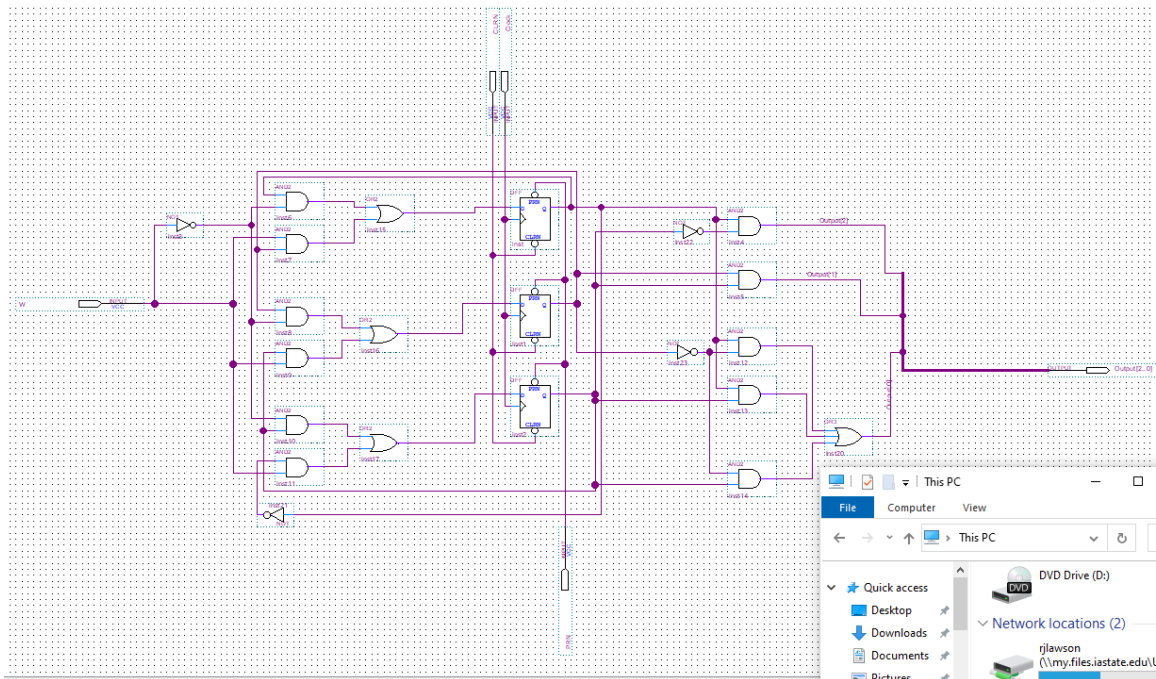
Simplified Next State Logic Expressions:

$$Y_0 = w \text{ XOR } y_0$$

$$Y_1 = wy_0 \text{ XOR } y_1$$

$$Y_2 = wY_0Y_1 \text{ XOR } y_2$$

Circuit Diagram:



Q2. Design a simple counter (Section 3.0).

Number of States: 4

Number of State Variables: 5

State Table:

Present State	Next State		Output
	w=0	w=1	
A	A	B	0
B	B	C	2
C	C	D	4
D	D	A	5

State-Assigned Table:

Present State	Next State		Output
	w=0	w=1	
000	000	010	000
010	010	100	010
100	100	101	100
101	101	000	101

Canonical SOP Expressions for Next State Logic:

$$Y1 = !Q1wQ0 + Q1!w!Q0$$

$$Y2 = !wQ0 + w!Q0$$

Simplified Logic Expressions:

$$Y1 = Q1 \text{ XOR } wQ0$$

$$Y2 = !wQ0 \text{ XOR } Q0$$

Next State Logic Verilog Code:

```
module circuit_nsl(w, Q1, Q0, Y1, Y0);

    input w, Q1, Q0;
    output Y1, Y0;

    assign Y0 = w ^ Q0;
    assign Y1 = Q1 ^ (w & Q0);

endmodule
```

Output Logic Verilog Code:

```
module circuit_ol(Q1, Q0, Z2, Z1, Z0);

    input Q1, Q0;
    output reg Z2, Z1, Z0;

    always @(Q1 or Q0)

    begin
        case({Q1,Q0})
            2'b00: {Z2, Z1, Z0} = 3; b000;
            2'b01: {Z2, Z1, Z0} = 3; b010;
            2'b10: {Z2, Z1, Z0} = 3; b100;
            2'b11: {Z2, Z1, Z0} = 3; b101;

        endcase

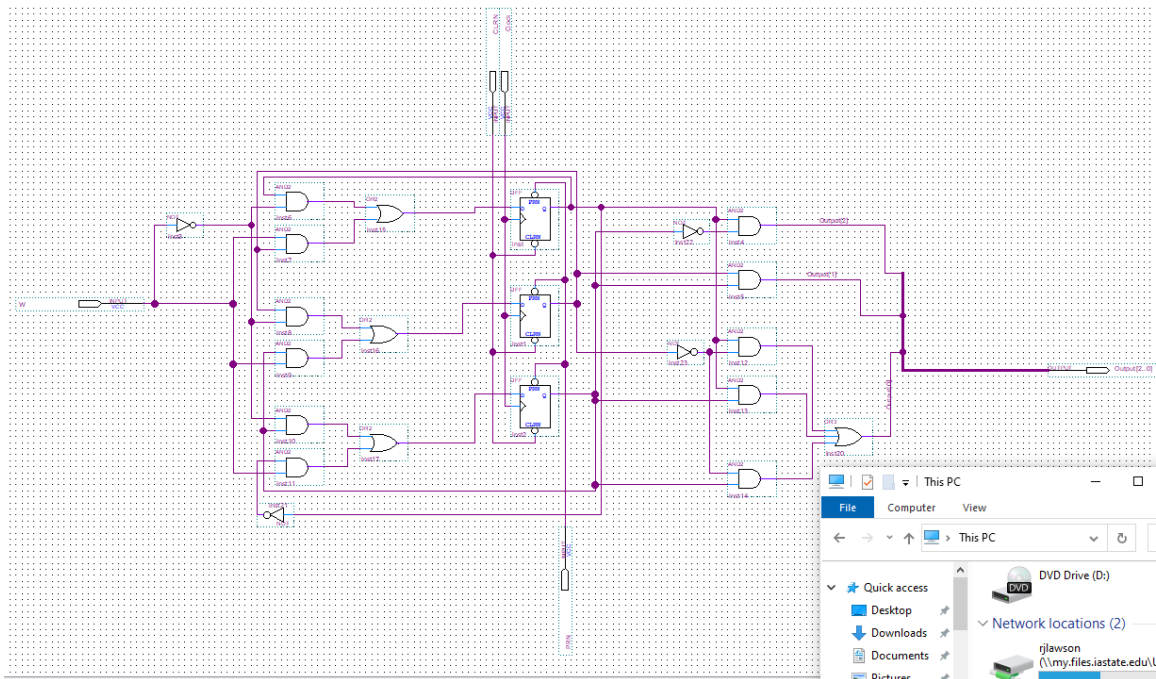
    endmodule
```

LAB:

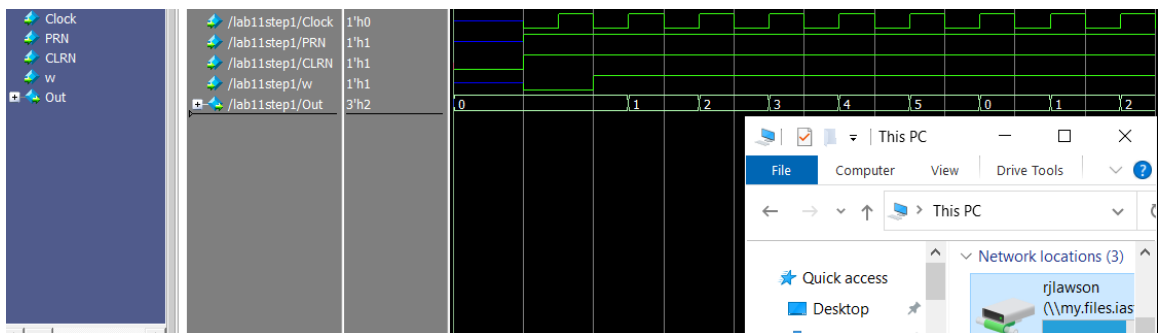
2.0 A Simple Counting Device

Screenshots:

<<<Insert a screenshot of your module-6 counter BDF here>>>



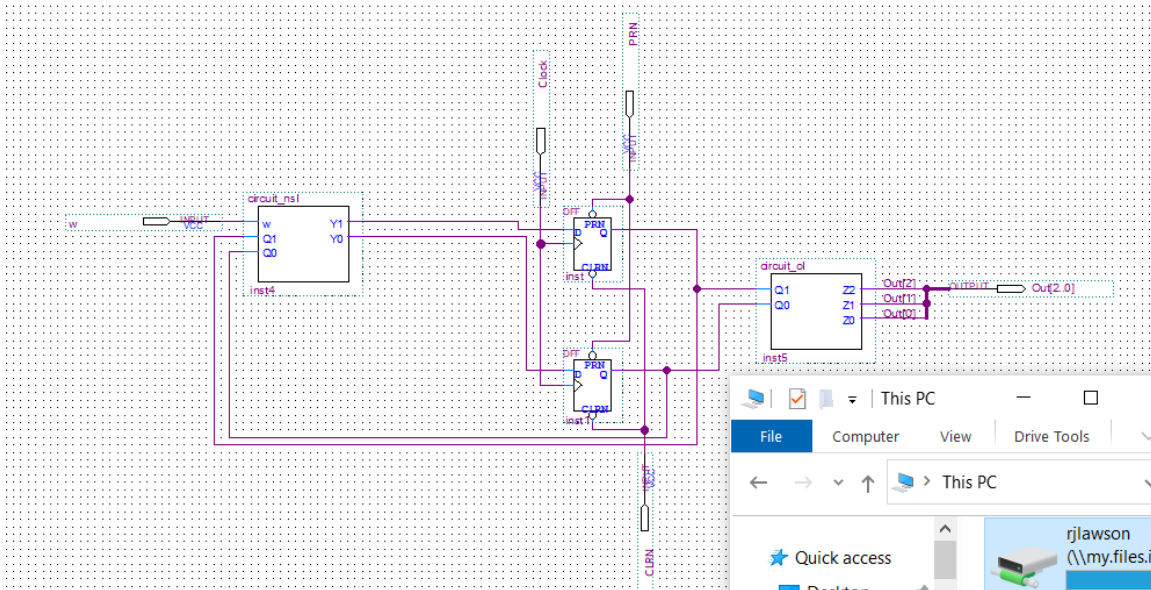
<<< Insert a screenshot of your waveform for your modulo-6 counter here>>>



3.0 A Simple Counter

Screenshots:

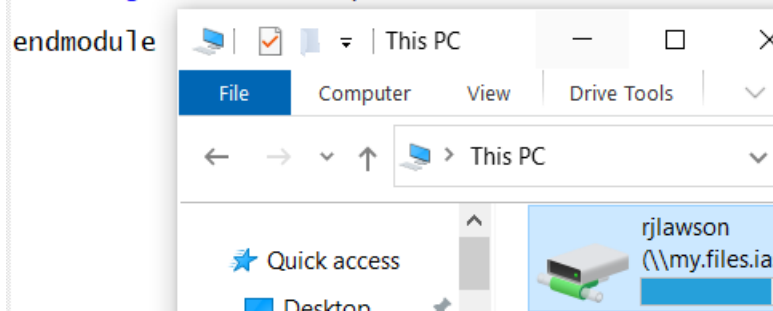
<<<Insert a screenshot of your simple counter here>>>



<<<Insert a screenshot of your next state logic here>>>

```
module circuit_nsl (w, Q1, Q0, Y1, Y0);
    input w, Q1, Q0;
    output Y1, Y0;

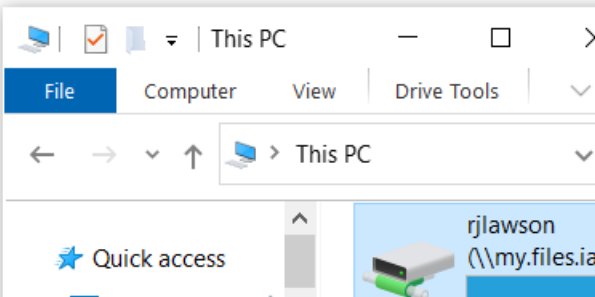
    assign Y1 = Q1 ^ (w & Q0);
    assign Y0 = w ^ Q0;
endmodule
```



<<<Insert a screenshot of your output logic here>>>

```
module circuit_01 (Q1, Q0, Z2, Z1, Z0);
    input Q1, Q0;
    output reg Z2, Z1, Z0;

    always @(Q1 or Q0)
    begin
        case({Q1 or Q0})
            2'b00: {Z2, Z1, Z0} = 3'b000;
            2'b01: {Z2, Z1, Z0} = 3'b010;
            2'b10: {Z2, Z1, Z0} = 3'b100;
            2'b11: {Z2, Z1, Z0} = 3'b101;
        endcase
    end
endmodule
```



<<< Insert a screenshot of your waveform for your simple counter here>>>

