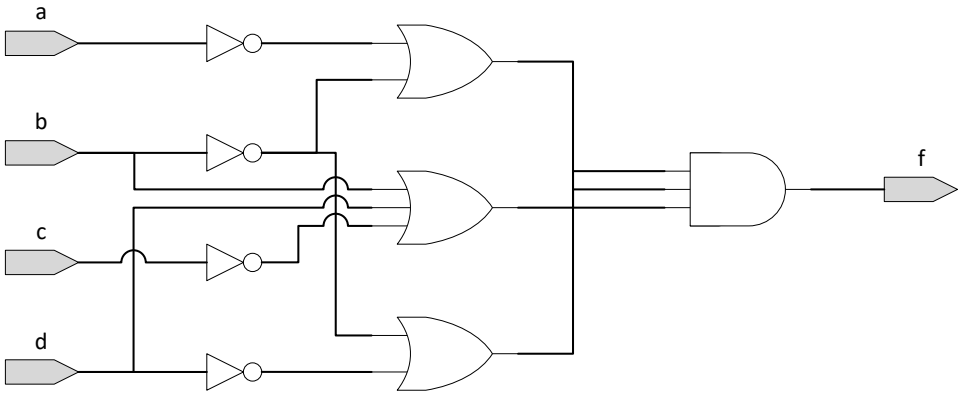
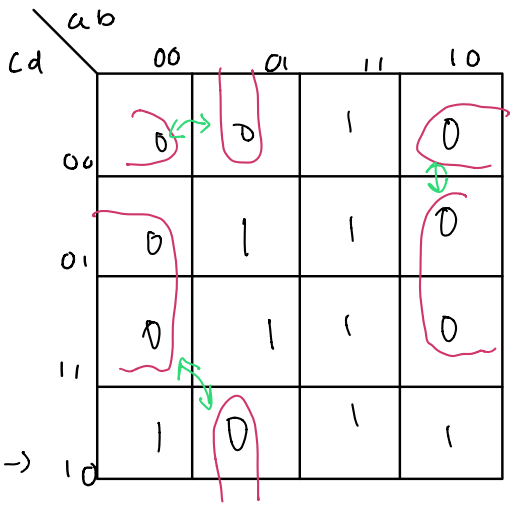
CPE 322 **EXAM I** **Name** Nolan Anderson

Spring Semester 2021

Work should be performed systematically and neatly with the final answer being underlined. This exam is open book, open notes, Closed neighbor/device/browser. Allowable items on desk include: exam, pencils, and pens. All other items must be removed from student's desk. Students have approximately 90 minutes (1 1/2 hours) to complete this exam. Best wishes!

1. [10 points] For the logic network shown below, find all static 0 hazards. For each 0-hazard found, specify the conditions which will cause the hazard to appear at the output (i.e. specify the logic values of the variables which are constant and clearly specify the variable that is assumed to be changing). If there are no 0-hazards found, use a K-map to show why this is the case.





$(a'+b')(b+c'+a)(d'+b')$

$(AB)+(B'CD')+(DB)$

0 1 0

using ABCD : 0000

a): a = 0, b = 0 ↔ 1, c = 1, d = 0 ↔ 1 — 0101 ↕ 0010

b): a = 1, b = 0, c = 0, d = 0 ↔ 1 — 1000 ↕ 1001

c: a = 0, b = 0 ↔ 1, c = 0, d = 0 — 0000 ↕ 0100

2. [12 points] Short Answer:
   a. What is the basic definition of combinational logic? What is the basic definition of sequential logic?

- Combinational -> Output depends solely on input, present input only and no memory.
- Sequential -> Has memory, depends on sequence of inputs, not always syncronous

   b. Does sequential logic always require a clock signal? Explain why or why not?

Yes, you need to know when to perform the operation. This clock signal lets you do that.

   c. What is the major difference between a blocking and non-blocking procedural assignment statement in Verilog?

Blocking -> (=) -> evaluation happens before next statement in block

non blocking <= -> allows several statements to evaluate at once

   d. Give an example where switching a blocking assignment to non-blocking (and vice-versa) leads to different behavior?

When you are debouncing and blocking causes to long of a delay. Switch to non blocking to get the delay after the input.

   e. What are the major differences between inertial and transport delays? Why are both modeled in Verilog?

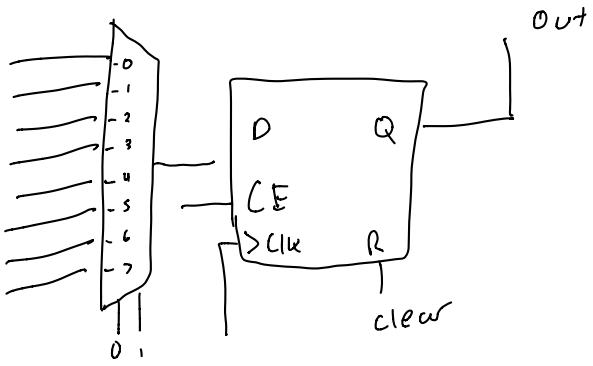
Inertial happens before the assignment happens
Transport happens after the assignment.

Gives you more options for reducing bad signals and weird delays.

   f. What happens to user specified timing parameters when a digital logic circuit is synthesized?

They are ignored

3. [10 points] Use a single 8-to-1 MUX and a rising-edge triggered D Flip-flop to create a rising-edge triggered JK Flip-flop. Assume positive logic where a connection to VCC will be interpreted as a logic 1 and a connection to GND will be interpreted as a logic 0.



4. [8 points] Write a short Verilog description of a negative-edge triggered set/reset flip-flop that in addition to the normal active-high synchronous Set and Reset inputs also has an asynchronous Clear input that is active high (i.e. a Logic 1 clears the flip-flop output independent of the clock). Do this using Verilog Procedural Statements.

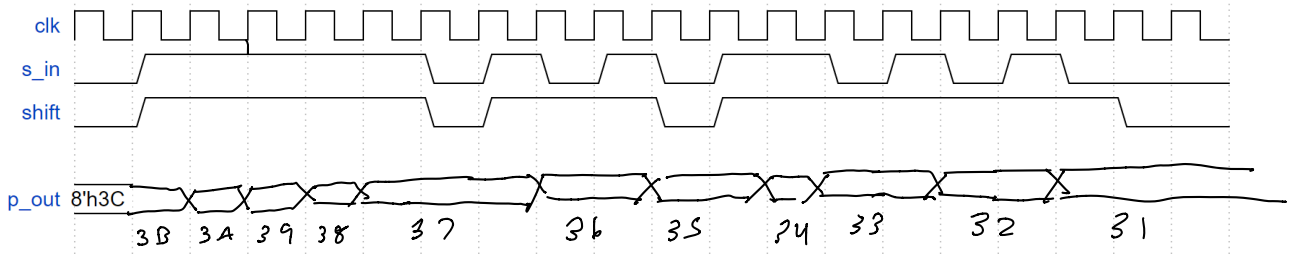
```
module SRFF (input Clock, Set, Reset, Clear, output reg Q);
    Input Clock, set, Reset, clear;
    output reg Q
    Input A; //using A as input
    always @( negedge Clock);
     begin
      if (set)
          assign Q = A;
      else if (reset)
          assign Q = Q;
      else if( clear)
          assign Q = 0;
    end
end module
```

5. [10 points] Write a Verilog Model for a Serial-in, Parallel-out 8-bit Shift Register. See the port definitions to determine required behavior.

```
// The following I/O ports are defined as follows:
// clk    - global 1-bit lock signal, all operations must occur on the rising edge
// s_in   - 1-bit serial data in
// shift  - high to enable s_in to be clocked into the shift register, low keeps the shift
//          register contents unchanged
// p_out  - 8-bit parallel data out, where bit 0 is the least recent (first) serial bit to be
//          shifted in and bit 7 is the most recent (last) serial bit to be shifted in
module serial_to_parallel (input clk, s_in, shift, output reg p_out[7:0])
```

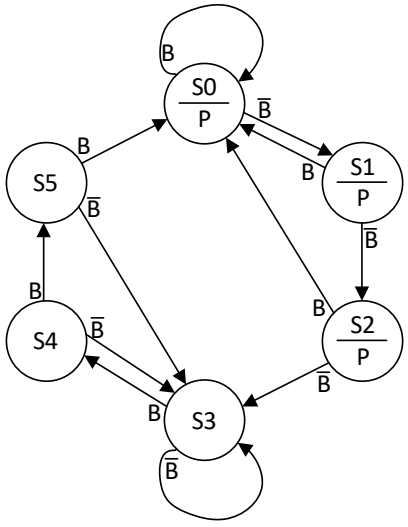
```
input clk, s_in, shift;
output reg p_out[7:0];

always @(posedge clk)
begin
  if (shift==2'b00)
    p_out[7:0] = p_out[7:0]
  else if (shift==1)
    begin
      p_out[7] = p_out[7-s_in; //shift in.
      p_out[6] = p_out[7-s_in;
      p_out[5] = p_out[7-s_in;
      p_out[4] = p_out[7-s_in;
      p_out[3] = p_out[7-s_in;
      p_out[2] = p_out[7-s_in;
      p_out[1] = p_out[7-s_in;
    end

end module
```

6. [10 points] Based on the Serial-in, Parallel-out 8-bit Shift Register defined in problem 5, with p_out[7:0] initially containing the value 8'h3C, complete the following timing diagram by listing the p_out signal value for every clock cycle in sequence:



assuming 8'h is before each one.

7. [10 points] For the following Extended State Transition Graph shown below:



a. Identify the type of synchronous sequential network it represents, and explain (Mealy FSM? Moore FSM? Combined Mealy/Moore?)

More network, only depends on present state

b. Create an equivalent Algorithmic State Chart representation

	Next State b=0	Next State b=1	Output
S0	S_1	S_0	1
S1	S_2	S_0	1
S2	S_3	S_0	1
S3	S_3	S_4	0
S4	S_3	S_5	0
S5	S_3	S_0	0

What is P? I am assuming P=1.

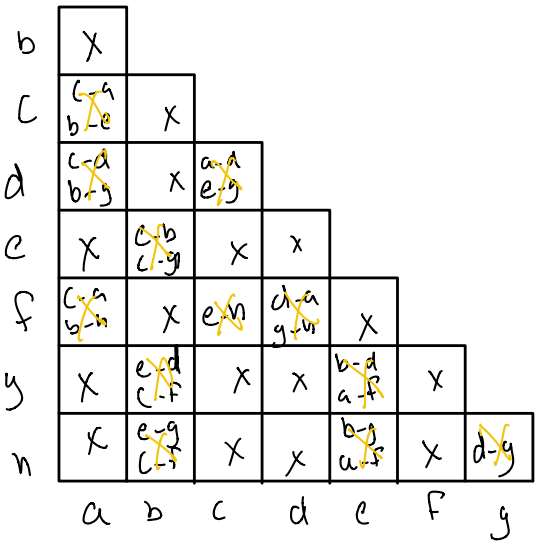
c. Write a behavioral Verilog HDL model that implements the state transition graph. In your model include a synchronous active high reset input signal that will always place the design in state S0 on the active edge of the next clock pulse regardless of the current state of the network.

Out of time.

8. [10 points] Reduce the following state table to a minimum number of states clearly identifying the states that are equivalent with one another. Show the final reduced state table.

Present State	Next State X=0	Next State X=1	Present Output
a	c	b	1
b	e	c	0
c	a	e	1
d	d	g	1
e	b	a	0
f	a	h	1
g	d	f	0
h	g	f	0

For your convenience, an iteration chart is provided below – please label grid along axes, and create additional copies if needed.



None are equivalent

Out of time, would redo though.

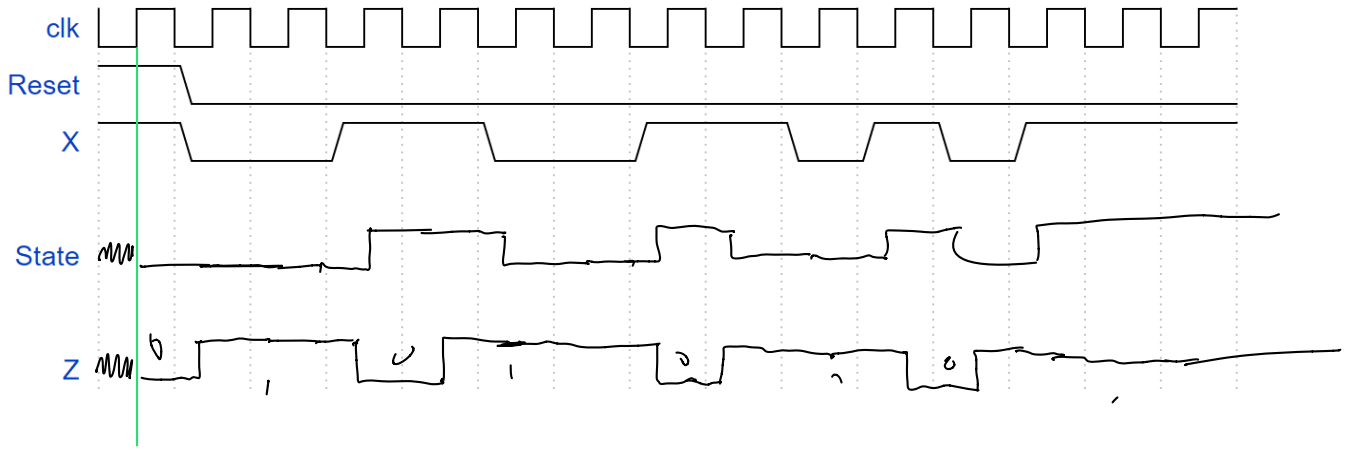
9. [5 points] True/False – Circle the correct answer
   a. The order of continuous assignment statements determines the order in which they are analyzed.
      TRUE or FALSE
   b. Verilog's synthesis tools treat white spaces (space or tab) and carriage returns differently.
      TRUE or FALSE
   c. There can only be one always block in a module.
      TRUE or FALSE
   d. In Verilog, Behavioral Implementations generally give the designer the most control on how the module will actually be implemented during the synthesis process.
      TRUE or FALSE
   e. A signal type must be declared as a net type (e.g. wire or tri) in Verilog in order to be a valid target for a procedural assignment.
      TRUE or FALSE

10. [5 points] Develop a Verilog model for a 4-to-1 multiplexer where the general inputs represent a 4-bit bus and the output is a single signal. Label the inputs I[3:0], S, and the output O.

```
module four_to_one ( input I[3:0], S, output O)

    input I[3:0];
    input A, B, ;
    input S;
    output O

    assign O = (~A && ~B && I[0]) || (~A && B && I[1]
                || (A && ~B && I[2]) || ( A && B && I[3]);

endmodule
```

11. [10 points] Complete the following timing diagram for the output signal **Z**, and the internal input signal **state**. Assume that a basic functional RTL simulation is to be performed.

```
module fsm_network (input clk, X, Reset, output reg Z);
reg state, next_state;
always @(state, X)
   case (state)
        0 : if (X) begin Z=0; next_state=1; end
            else begin Z=1; next_state=0; end
        1 : if (X) begin Z=1; next_state=1; end
            else begin Z=1; next_state=0; end
   endcase
always @(posedge clk)
   if (Reset) state=0;
   else state = next_state;
```



What type of sequential network does this Verilog Model represent?

Mealy - depends on state and inputs

Out of time ...