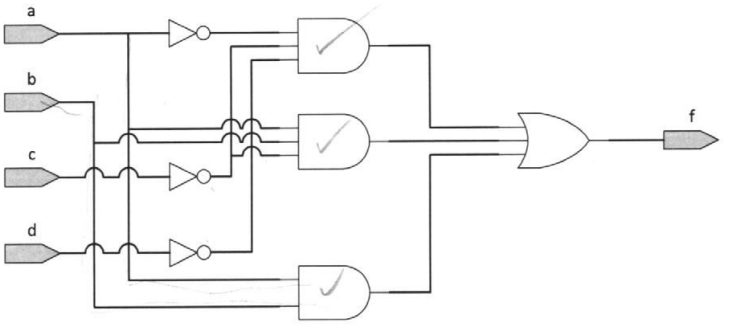
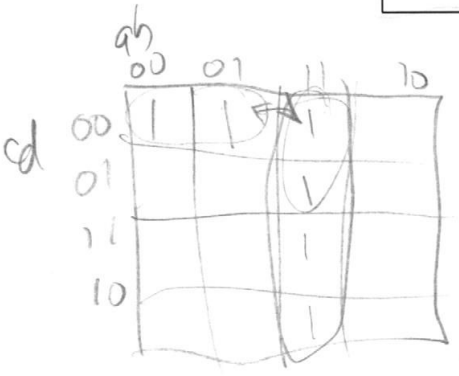
322 EXAM I Name _____

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 1 hazards. For each 1-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 1-hazards found, use a K-map to show why this is the case.





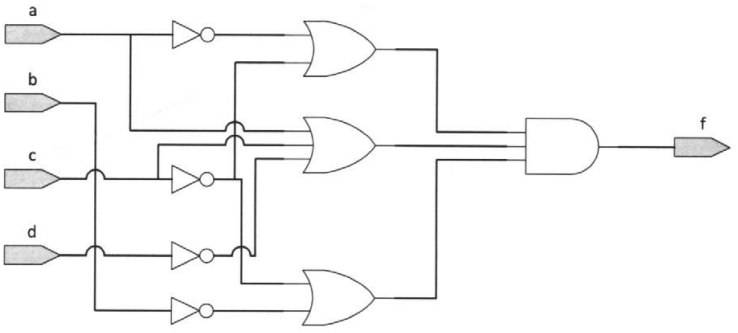
a 0 → 1
c = 0
d = 0
b = 1

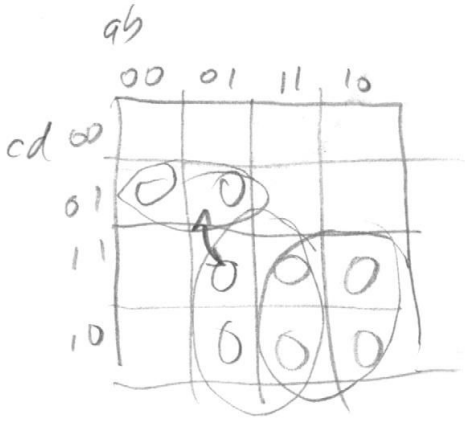
CPE 322 **EXAM I** **Name** _____

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.





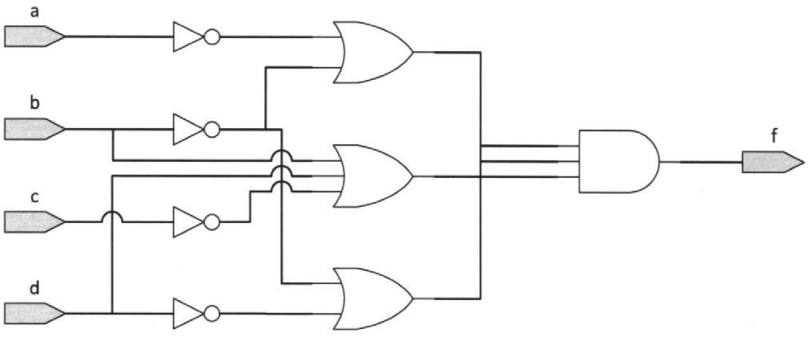
a = 0
b = 1
d = 1
c = 0 → 1

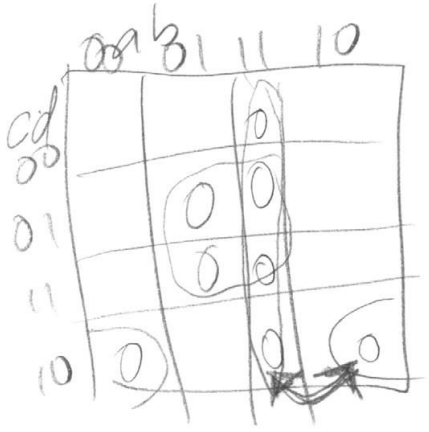
CPE 322 **EXAM I** **Name** _____

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.





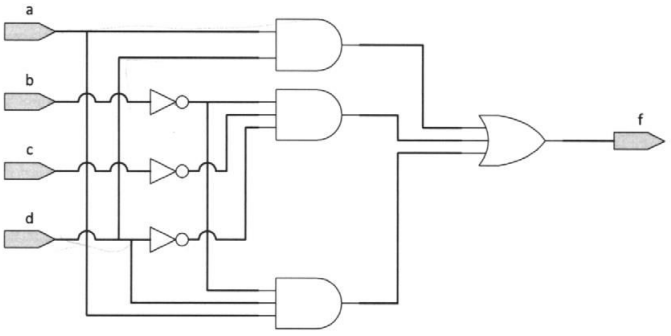
a = 1
c = 1
d = 0
b = 0 ⟷ 1

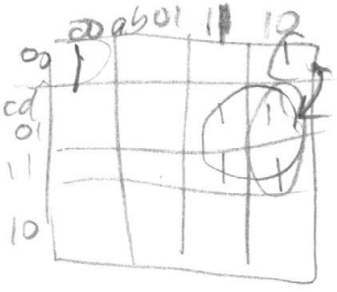
CPE 322 **EXAM I** **Name** _____

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 1 hazards. For each 1-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 1-hazards found, use a K-map to show why this is the case.





a = 1
b = 0
c = 0

d 0 ⟷ 1

2. [12 points] Short Answer:

a. What is the basic definition of combinational logic? What is the basic definition of sequential logic?

combinational logic – outputs depend only upon inputs, no memory/states involved

sequential – contains memory/state, output depends on state

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

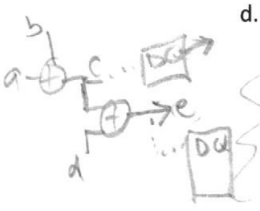
no – in the case of latches it depends on a gate instead of a clock

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

blocking (=) assigns the new value immediately before proceeding with the remainder of the procedure

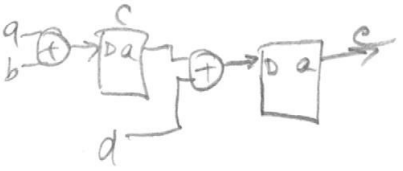
non-blocking (<=) assigns only at the end of the procedure

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




```
...
c = a+b;
e = d+c;
...
```

```
...
c <= a+b;
e <= d+c;
...
```



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

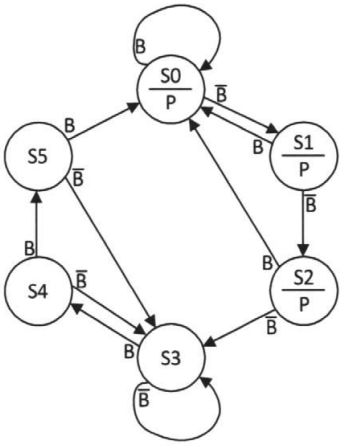
inertial: models gate switching, thus it suppresses pulses that do not equal or exceed the delay

transport: models propagation delay, does not swallow glitches/pulses

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

They are ignored

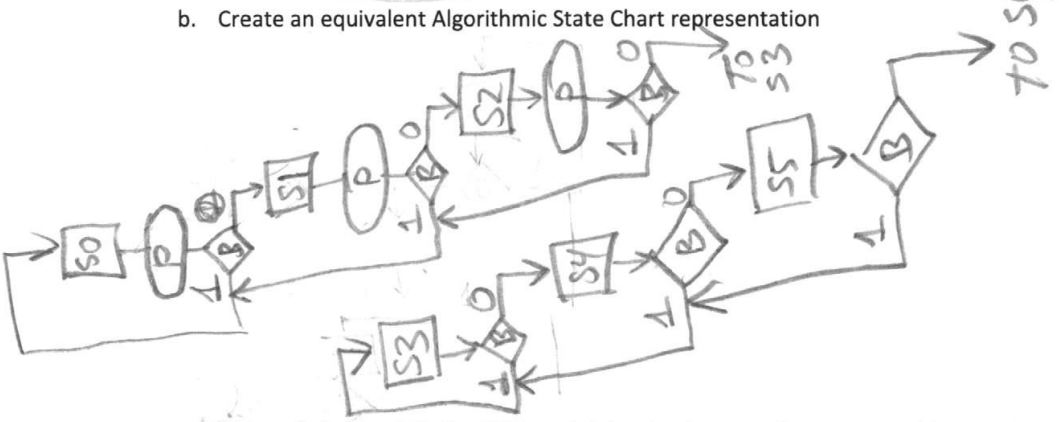
7. [10 points] For the following Extended State Transition Graph shown below:



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

Moore

b. Create an equivalent Algorithmic State Chart representation



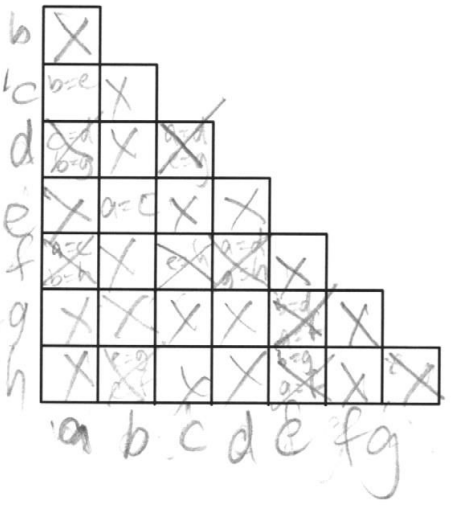
c. Write a behavioral Verilog HDL model that implements the state transition graph. In your model include a synchronous active high reset input signal that will always place the design in state S0 on the active edge of the next clock pulse regardless of the current state of the network.

```
let S0=000 S1=001 S2=010 S3=011 S4=100 S5=101
always @ (posedge clk) begin
    if (reset) st <= 3'b0;
    else if ((st==0 || st==1 ||st==2) && B==1) st <= 3'b0;
    else if (st==3 || st==4 || st==5) && B==0) st <= 3'b011;
    else if (st==5) st <= 3'b000;
    else st <= st + 3'b001;
end
assign P <= (st <= 3'b010)? 1'b1 : 1'b0;
```

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~~ a	b	1
b	~~e~~ b	~~c~~ a	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.

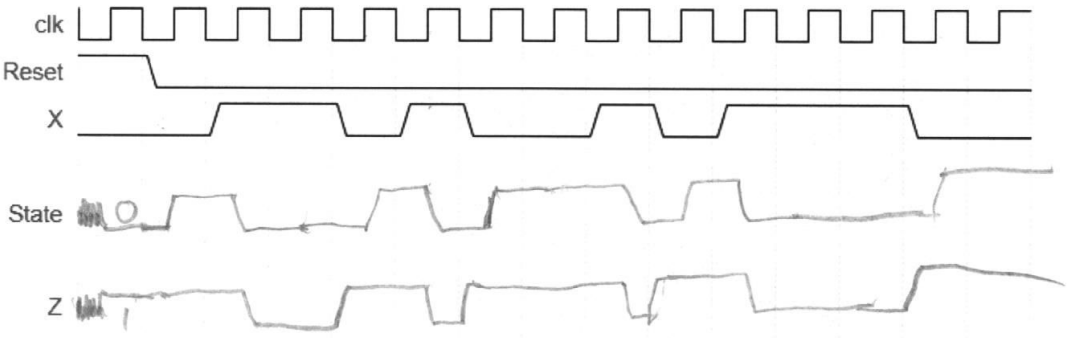


b = e
a = c

present state	next state X=0	next state X=1	present output
a	a	b	1
b	b	a	0
d	d	g	1
f	a	h	1
g	d	f	0
h	g	f	0

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=0; end
             else begin Z=1; next_state=1; end
         1 : if (X) begin Z=1; next_state=0; end
             else begin Z=1; next_state=1; end
   endcase
always @(posedge clk)
   if (Reset) state=0;
   else state = next_state;
```

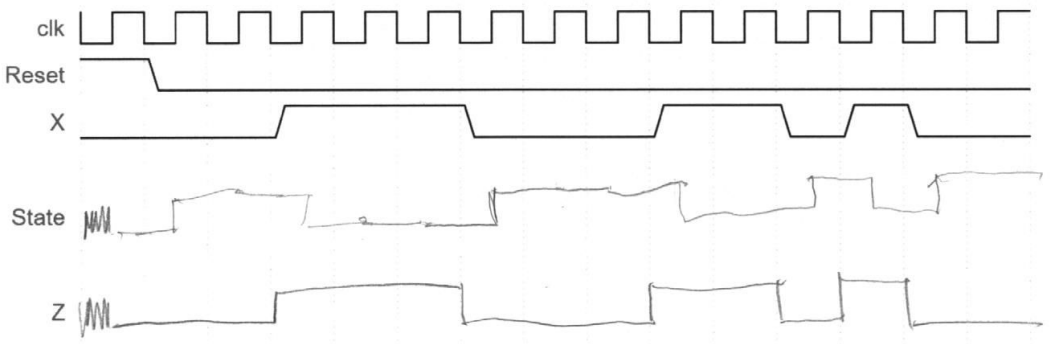


What type of sequential network does this Verilog Model represent?

mealy

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.

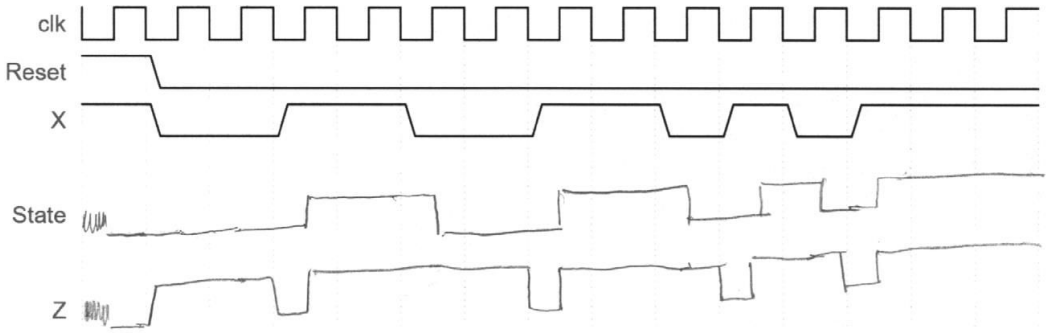
```
module fsm_network (input clk, X, Reset, output reg Z);
reg state, next_state;
always @(state, X)
   case (state)
         0 : if (X) begin Z=1; next_state=0; end
             else begin Z=0; next_state=1; end
         1 : if (X) begin Z=1; next_state=0; end
             else begin Z=0; next_state=1;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



What type of sequential network does this Verilog Model represent?

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.

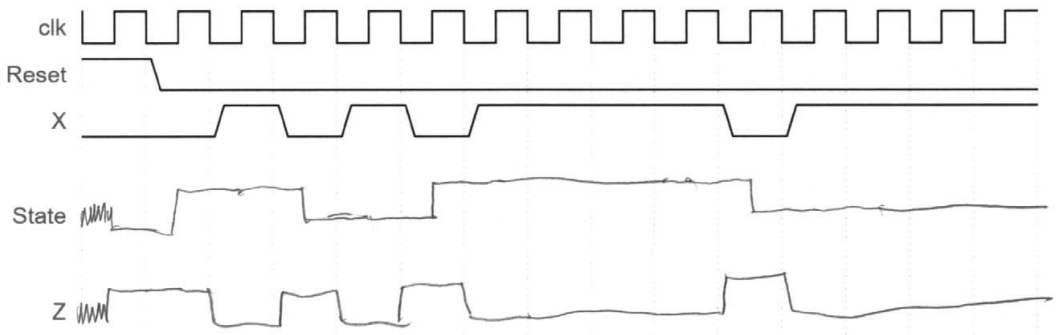
```
module fsm_network (input clk, X, Reset, output reg Z);
reg state, next_state;
always @(state, X)
   case (state)
         0 : if (X) begin Z=0; next_state=1; end
             else begin Z=1; next_state=0; end
         1 : if (X) begin Z=1; next_state=1; end
             else begin Z=1; next_state=0;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



What type of sequential network does this Verilog Model represent?

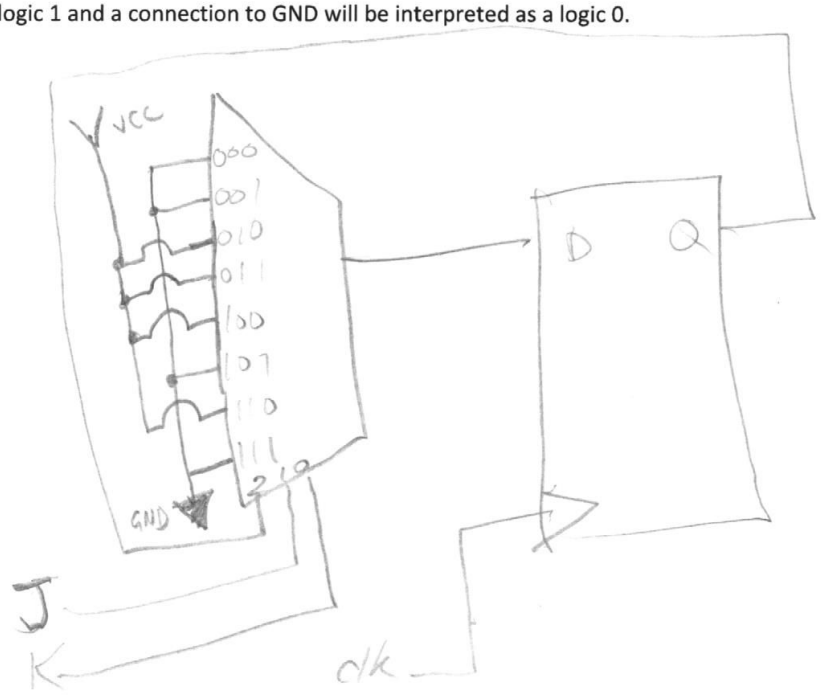
11. [10 points] Complete the following timing diagram for the output signal **Z**, and the internal input signal **state**. Assume that a basic functional RTL simulation is to be performed.

```
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=0; end
             else begin Z=1; next_state=1; end
         1 : if (X) begin Z=0; 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?

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.

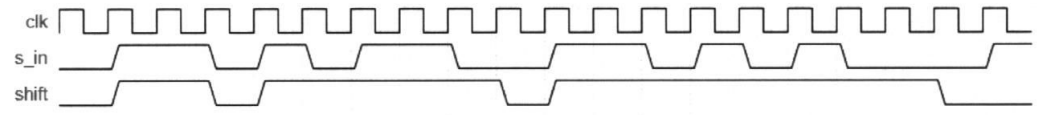
```
module SRFF (input Clock, Set, Reset, Clear, output reg Q);

always @ (negedge clock or posedge Clear)
begin
    if (Clear)
        Q <= 1'b0;
    else if (Set)
        Q <= 1'b 1;
    else if (Reset)
        Q <= 1'b 0;
end
endmodule
```

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])
```

```
always @(posedge clk)
begin
    if (shift)
        p_out [7:0] <= {s_in, p_out[7:1]};
end
```

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'h55, complete the following timing diagram by listing the p_out signal value for every clock cycle in sequence:

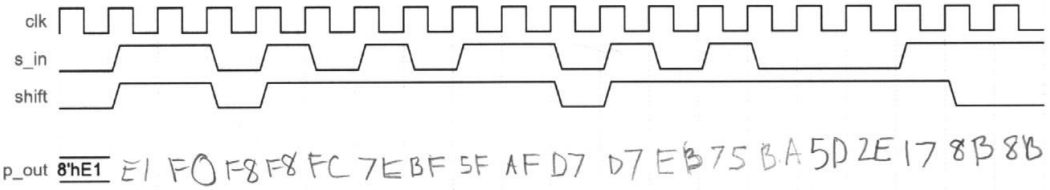


p_out 8'h55 55 AA D5 D5 EA 75 BA DD GE GE B7 DB 6D B6 5B AD 5G 2B 2B

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])
```

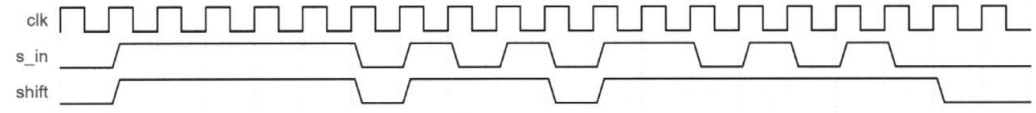
6. [10 points] Based on the Serial-in, Parallel-out 8-bit Shift Register defined in problem 5, with p_out[7:0] initially containing the value 8'hE1, complete the following timing diagram by listing the p_out signal value for every clock cycle in sequence:



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])
```

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:

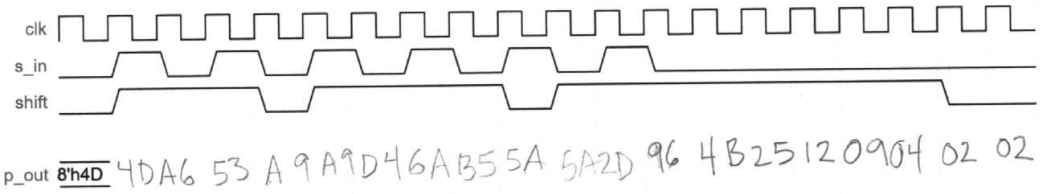


p_out 8'h3C 3C 9E CF E7 F3 F9 F9 FC 7E BF BF DF EF 77 BB 5D AE 57 57

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])
```

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'h4D, complete the following timing diagram by listing the p_out signal value for every clock cycle in sequence:



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 (input [3:0]I, [1:0]S, output O)
begin
    assign O = I[S];

    or

    assign O = S[1] ? S[0] ? I[3] : I[2] :
               S[0] ? I[1] : I[0];

end module
```