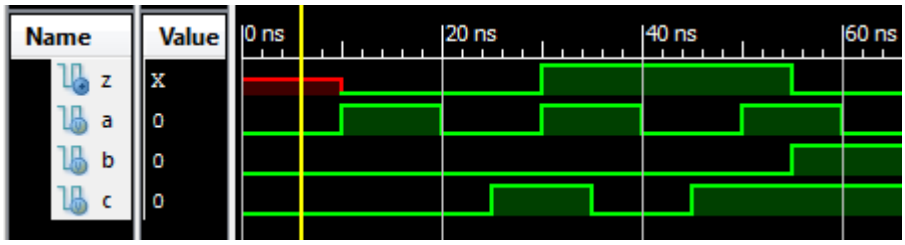
Name:_____

Honor Pledge: I am adhering to the Honor Code while taking this test.

Signature:_____ Date:_____

1. The figure shows the simulation of a D flip-flop. **z** is the output.



A. Which input is normally called "clear" or "reset"? b

B. Why is the value of the output unknown before 10 ns? We are in a "store" state, and we don't know what value is being stored.

C. If **b** falls to 0 at 68 ns, while **a** remains at 0 and **c** remains at 1, will the output change? no

D. (Five points.) Write a Verilog module (not the test fixture) to implement this circuit.

```
module partF (input a, input b, input c, output reg z);

always @ (posedge a or posedge b)

if (b == 1) z <= 0;

else z <= c;

endmodule
```

E. (Two points.) Write a Verilog module to implement a circuit whose output becomes 0 immediately when the clear input goes low (not high); otherwise, the circuit is identical to the previous case.

```
module partF (input a, input b, input c, output reg z);

always @ (posedge a or negedge b)

if (b == 0) z <= 0;

else z <= c;

endmodule
```

2. Using D flip-flops and logic gates, design a Mealy machine that detects the sequence 1001.

A. First, list all the states (s0, s1, etc.) that we need to define, and briefly define them.

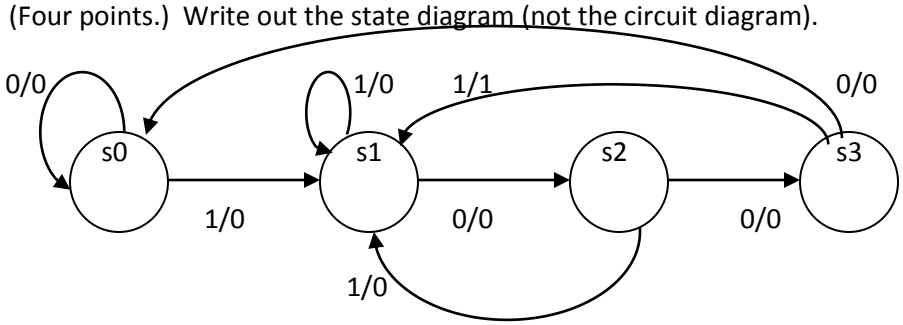
s0: nothing detected

s1: first 1 detected

s2: 10 detected

s3: 100 detected

B. (Four points.) Write out the state diagram (not the circuit diagram).



C. How many flip-flops do you need? 2

D. Write out the necessary truth table, showing how Next State and Output depend on Present State and Input.

	q1	q0	In	D1	D0	Out
s0	0	0	0	0	0	0
s0	0	0	1	0	1	0
s1	0	1	0	1	0	0
s1	0	1	1	0	1	0
s2	1	0	0	1	1	0
s2	1	0	1	0	1	0
s3	1	1	0	0	0	0
s3	1	1	1	0	1	1

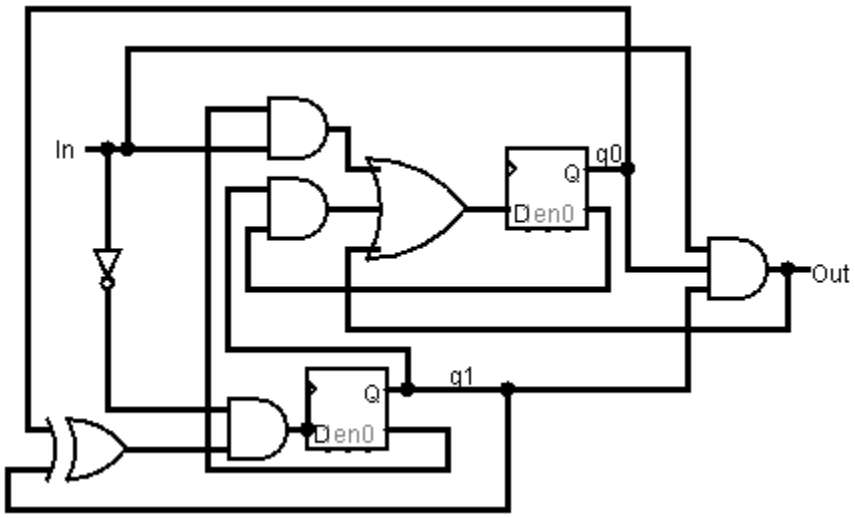
E. (Three points.) Write a logic equation for each bit's Next State and for the Output.

D0 = ~q1 & In | q1 & ~q0 | q1 & q0 & In

D1 = ~In & (q1 ^ q0)

Out = q1 & q0 & In

F. (Three points.) Write out the circuit diagram.



G. Write an expression for D0 (as a function of q0, q1, and In) that requires only three logic gates (including NOTs). Your response for part G may be the same as part of your response for part E.

D0 = In | q1 & ~q0

H. Write an expression for D1 (as a function of q0, q1, and In) that requires only three logic gates (including NOTs). Your response for part H may be the same as part of your response for part E.

D1 = ~In & (q1 ^ q0)