

ECE2030b - HW-5 v.2 Due Monday 10/21 during class. - ANSWERS

Problem 1. Using Finite State Machine techniques, design a circuit to:

Detect when sequential input X delivers 3 logic 1's in a row.  
Do not detect overlapping sequences.

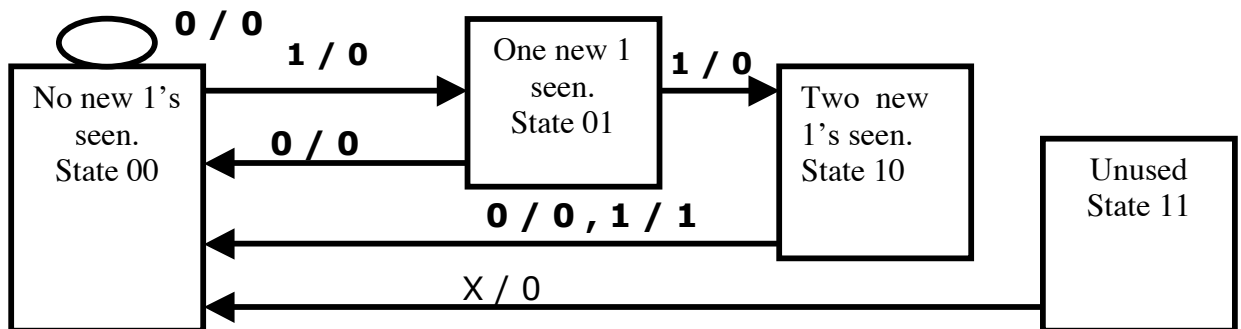
Example:

**Input:** 01101111011111001111110

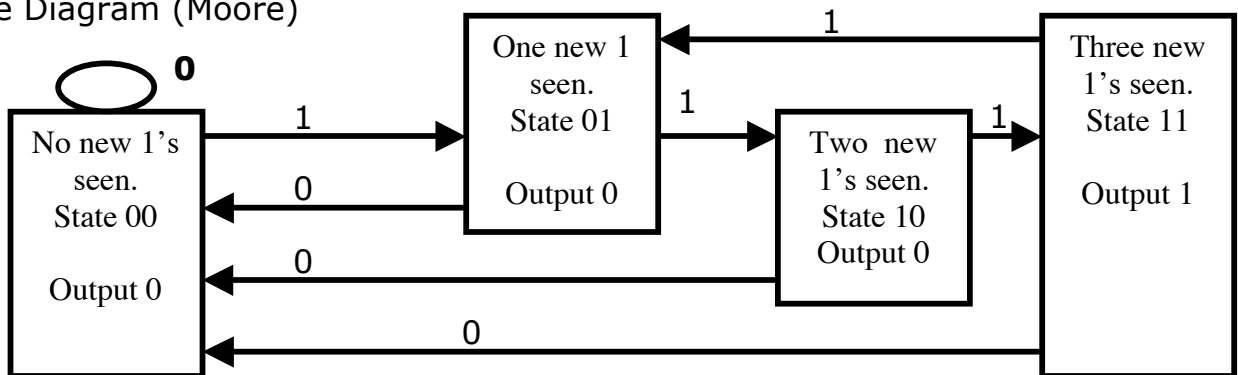
**Output:** 00000010000100000010010

- Draw a State Diagram showing all possible states and transitions.
- Draw a logic table for the Next State bits ( $N_i$ ) and the Output bit ( $Q$ ), as a function of Present State bits ( $P_i$ ) and Input bit ( $X$ ).
- Draw Karnaugh maps for the separate outputs,  $N_i$  and  $Q$ .
- Draw a logic diagram showing the necessary registers and combinatorial logic blocks.

A. State Diagram (Mealy)



State Diagram (Moore)



Check List: Does every state have exits defined for all inputs (0,1)?

B. Logic or Truth Tables:

Meely					
Present	State	Input	Next	State	Output*
P1	P0	X	N1	N0	Q
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	0	0	1
1	1	0	0	0	0
1	1	1	0	0	0

- For Meely Machine, output occurs while machine is in state 10 and X=1.

Moore Machine:

Moore					
Present	State	Input	Next	State	Output*
P1	P0	X	N1	N0	Q
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	0	1	1

For Moore Machine, output occurs while machine is in state 11. Logic for Q can be designed as a function of N1,N0

B. Karnaugh Maps for Moore Machine:

N1: X \ P1,P0	00	01	11	10
0	0	0	0	0
1	0	1	0	1

$$N1 = X ( P1' P0 + P1 P0' ) = X ( P1 \text{ XOR } P0 )$$

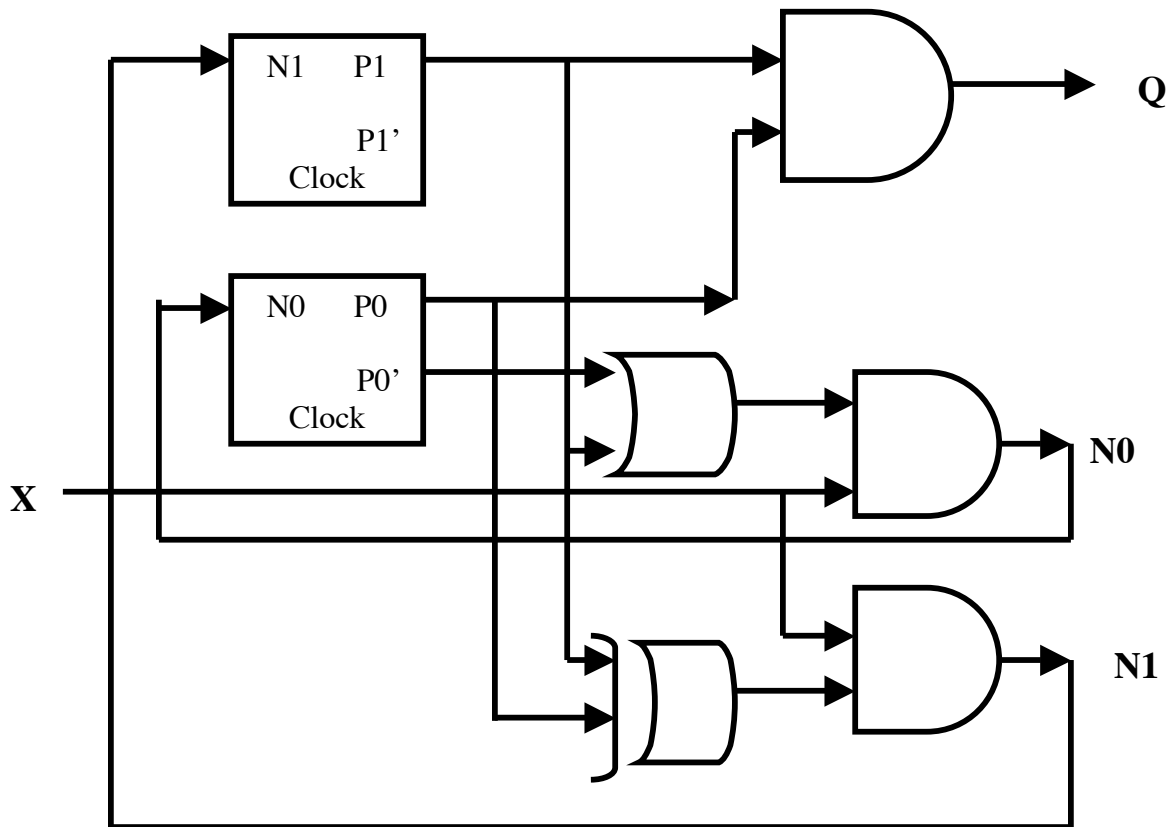
N0: X \ P1,P0	00	01	11	10
0	0	0	0	0
1	1	0	1	1

$$N0 = X ( P0' + P1)$$

Q: P1 \ P0	0	1
0	0	0
1	0	1

$Q = P0 P1$  (note: for Moore Machine, Q is function of present state (P1,P0)).

D. Logic Diagram (Moore)



Solution as a Meely Machine is also acceptable.