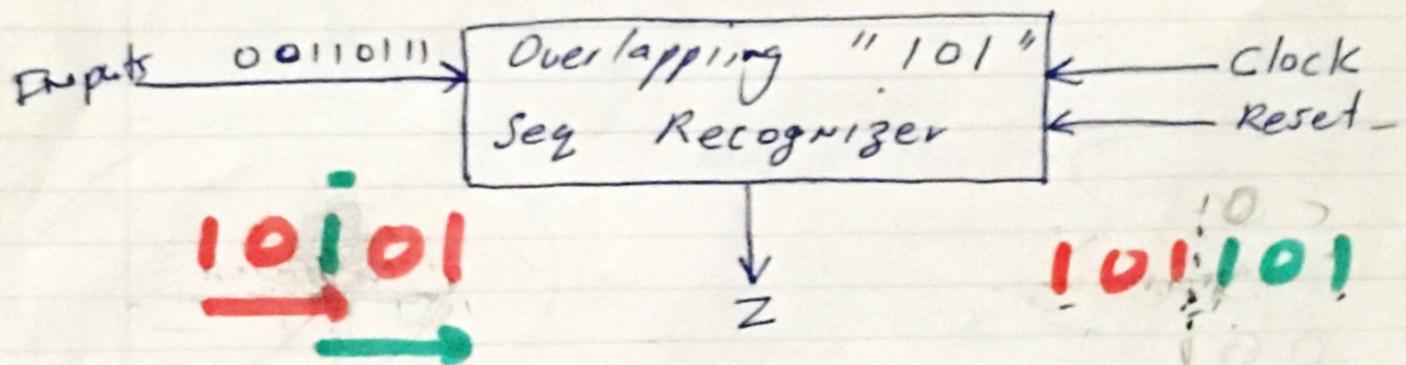
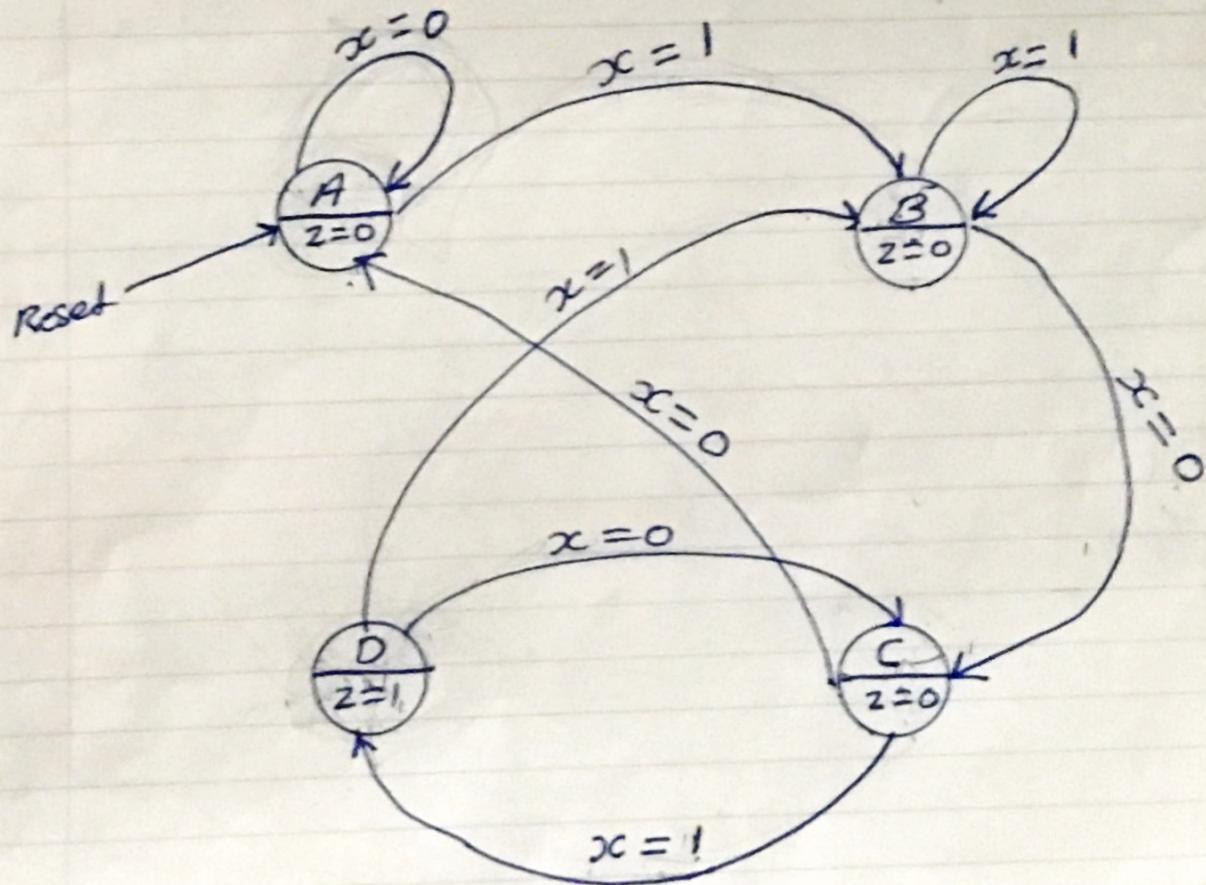


Example : Design of a Moore FSM that detects overlapping Seg "101"



Step 1: Create a Moore FSD - Finite State Diagram



Design of Moore FSM that detects Overlapping Sequence "101" - Cont.

Step 2 : Determine the minimum number of bits required to store the states

$$\text{Number of bits} = \log_2 [k] = \log_2 [4] = 2$$

$k = \# \text{ of states}$

Step 3 : From the FSD, create the Truth table for NSG + DG

$$A = 00, \quad B = 01,$$

$$C = 10, \quad D = 11$$

NSG

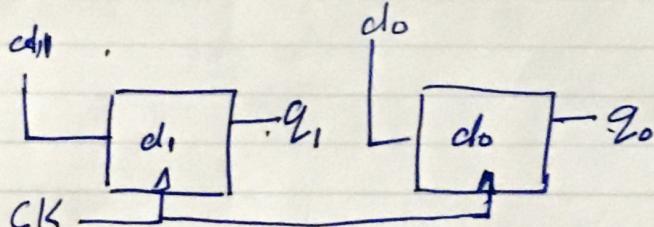
	Current state			Input X	Next state	
	q ₁	q ₀	X		d ₁	d ₀
1	A	0	0	0	0	0
2		0	0	1	0	1
3	B	0	1	0	1	0
4		0	1	1	0	1
5	C	1	0	0	0	0
6		1	0	1	1	1
7	D	1	1	0	1	0
8		1	1	1	0	1

Step 3 - Cont

Design of a Moore FSM that detects Overlapping Sequence "101"

Step 3: Create Output Generator (OG)

	OG		
	Current state		output
	q_1	q_0	Z
1	0	0	A
2	0	1	B
3	1	0	C
4	1	1	D



$clk = \text{clock signal}$

Design of a Moore FSM that detects Overlapping Sequence "101"

Step 4: From the truth table, determine
Min SOP for each of the
states - var ch. 0, output Z

$$\begin{aligned}
 d_1 &= \overline{x} \cdot \bar{q}_1 q_0 + x \cdot q_1 \bar{q}_0 + \overline{x} \cdot q_1 q_0 \\
 &= \overline{x} \cdot \bar{q}_1 q_0 + \overline{x} \cdot q_1 \bar{q}_0 + x \cdot \bar{q}_1 \bar{q}_0 \\
 &= \overline{x} \left(\bar{q}_1 q_0 + q_1 \bar{q}_0 \right) + x \cdot \bar{q}_1 \bar{q}_0 \\
 &\quad \underbrace{\qquad}_{\begin{array}{l} q_0 = 0 \\ q_0 = 1 \end{array}} \qquad \downarrow \qquad \qquad \qquad \rightarrow \qquad \text{Complement Law} \\
 &= \overline{x} q_0 + x \cdot \bar{q}_1 \bar{q}_0 \\
 d_1 &= \boxed{\overline{x} q_0 + x \cdot \bar{q}_1 \bar{q}_0}
 \end{aligned}$$

$$d_0 = \{x\}; \quad z = q_1 q_0.$$

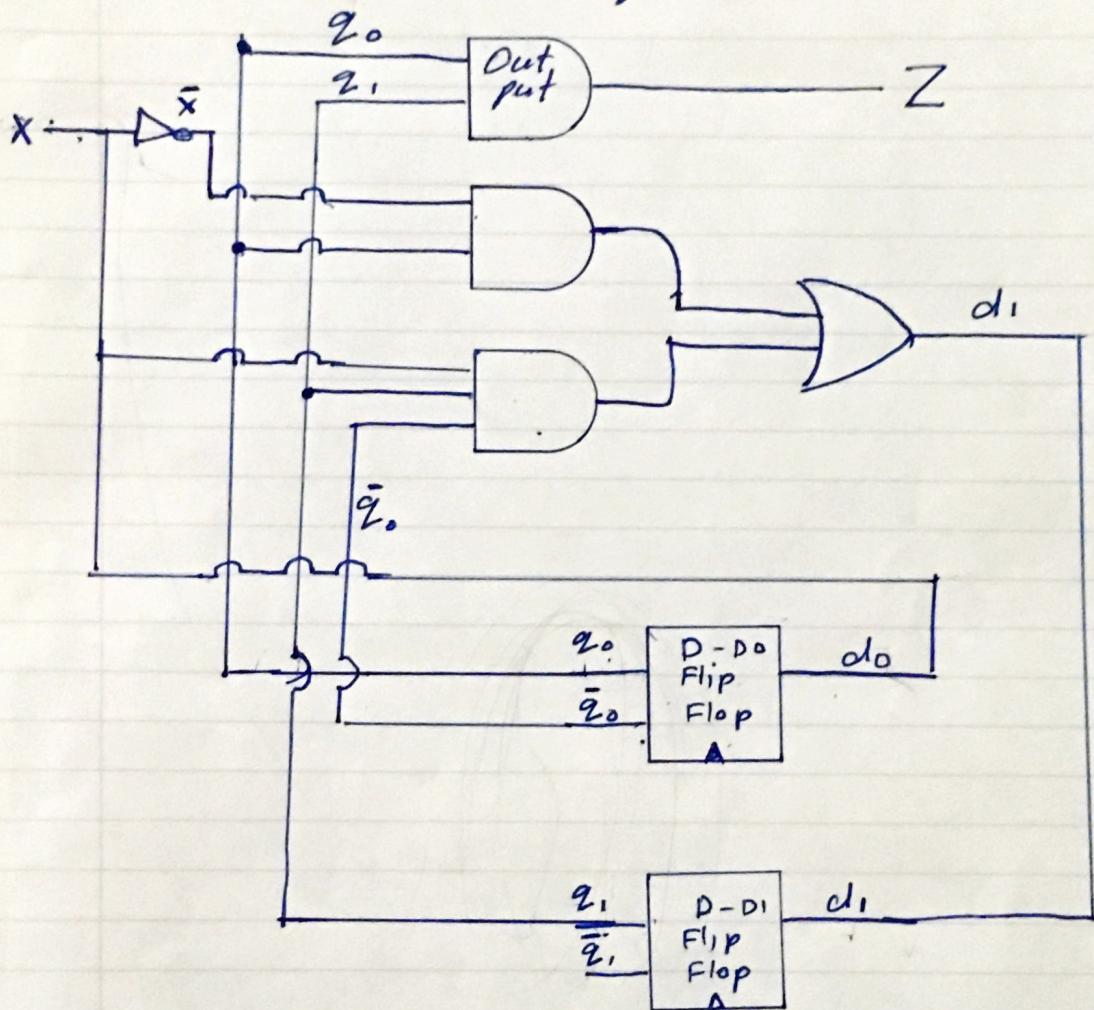
$$\underbrace{q_1 q_0 + q_1 q_0}_{e} \Leftrightarrow q_0$$

$g_0 = 0 \Rightarrow$ whole expression = 0

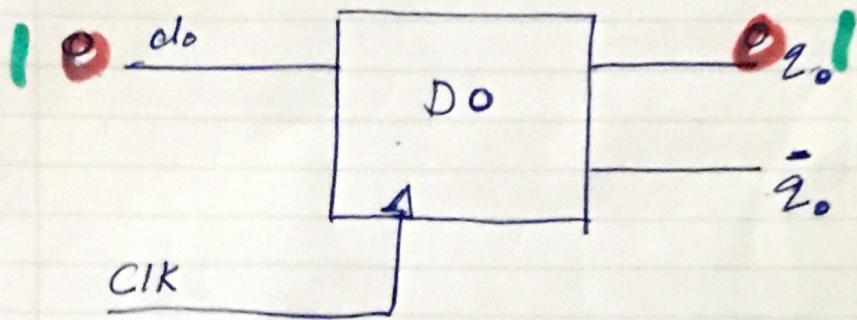
$$\bar{q}_1 \circ + q_1 \circ =$$

Design of a Moore FSM that detects "101" Sequence

Step 5: Built or design the circuit that detects sequence "101"



D flip Flops



$CLK = Clock$

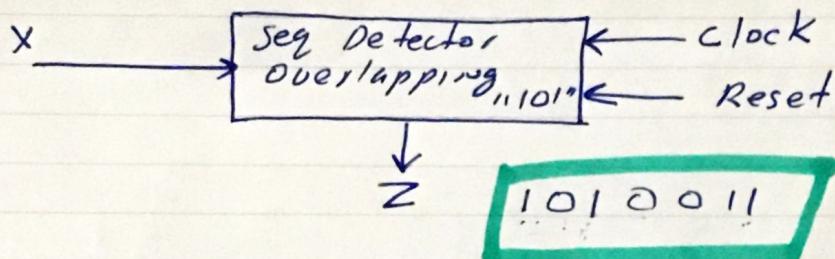
$d_0 = Input \text{ (1 bit)}$

$q_0 = Output$

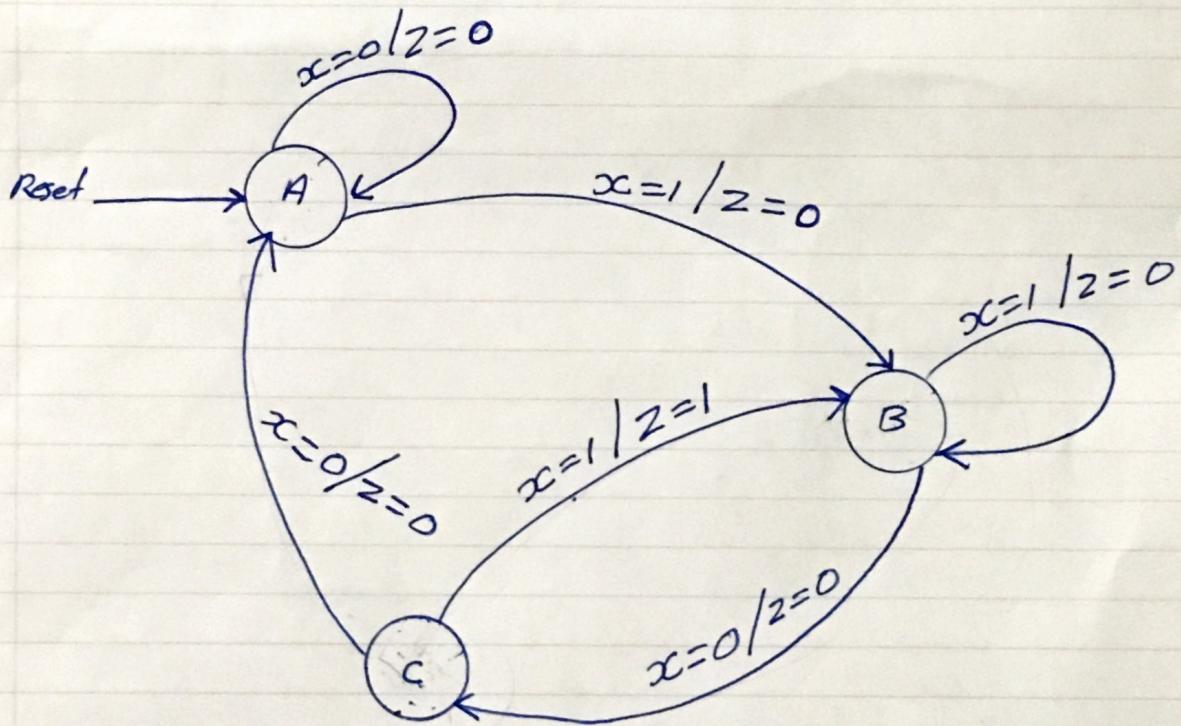
$\bar{q}_0 = Inverted \text{ Output}$

Design of Mealy FSM - Overlapping Sequence "101"

* Mealy output is assigned to the arcs and not to the states



Step 1: Create a Mealy Finite state Diagram



Design of Mealy FSM - Overlapping Sequence "101"

Step 2: Determine the Min number of states/bits required to store the states

$$\text{Number of bits} = \log_2 [k] = \log_2 [3] \approx 2$$

k = Total # of states

Step 3: From FSD, Create the truth table.
Let 00 = A, 01 = B, 10 = C, 11 = D.

NSG / OG						
	Current States		Input	Next state d ₁ d ₀		
	z ₁	z ₀	x			z
1	A	0 0	0	0 0	A	0
2		0 0	1	0 1	B	0
3	B	0 1	0	1 0	C	0
4		0 1	1	0 1	B	0
5	C	1 0	0	0 0	A	0
6		1 0	1	0 1	B	1
7	D	1 1	0	d d	d	d
8		1 1	1	d d	d	d

Design of Mealy FSM - Overlapping Sequence "101"

Step 4: Determine the logical Expression

$$d_1 = q_0 \bar{x} ; d_1 = q_0 \bar{q}_1 \bar{x}$$
$$d_0 = x \quad z = q_1 x$$

Step 5: Draw the Circuit Diagram

