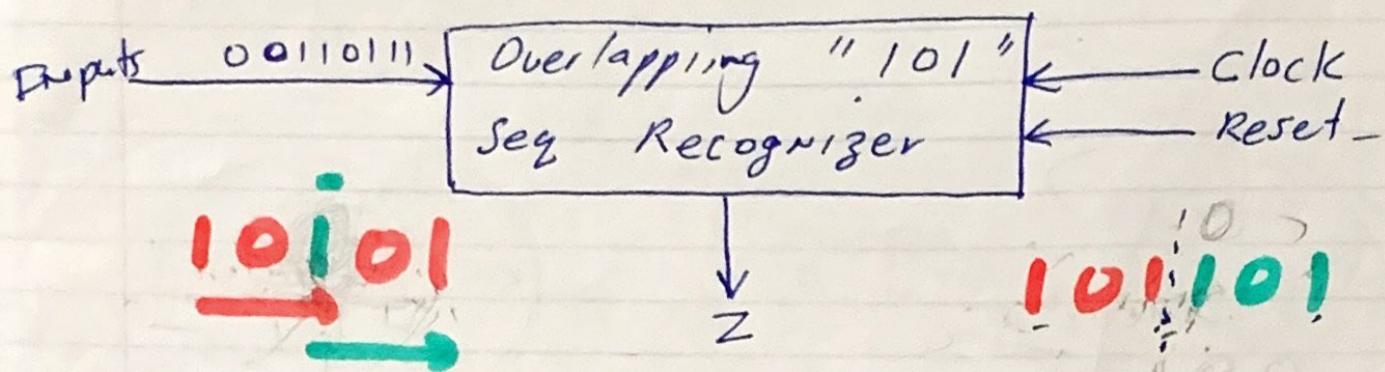
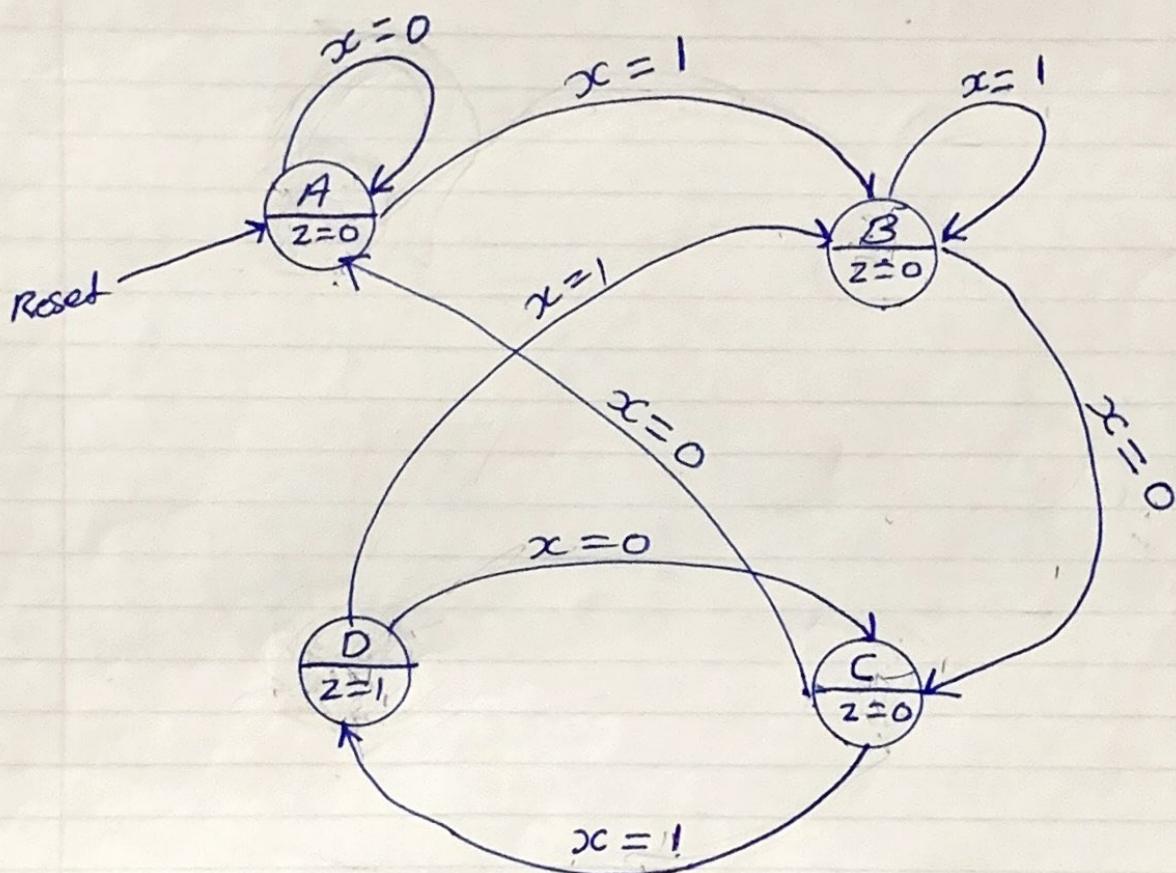


Example : Design of a Moore FSM that detects overlapping seq "101"



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

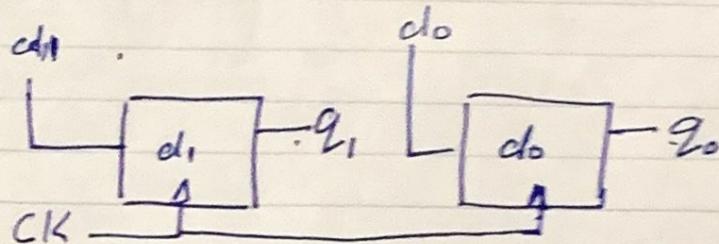
	Current state		Input X	Next state		
	$q_1$	$q_0$	X	$d_1$	$d_0$	
1	A	0	0	0	0	A
2		0	0	1	0	B
3	B	0	1	0	1	C
4		0	1	1	0	B
5	C	1	0	0	0	A
6		1	0	1	1	D
7	D	1	1	0	1	C
8		1	1	1	0	B

### Step 3 - Cont

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

Step 3: Create Output Generator (OG)

Current state		O.G.	output
	$q_1$	$q_0$	$Z$
1	0	0	A
2	0	1	B
3	1	0	C
4	1	1	D



clk = 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 d<sub>1</sub>, d<sub>0</sub>, output Z

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

Complement Law	
$\bar{q}_0 (\bar{q}_1 + q_1)$	$q_0 \cdot 1$
$q_0 \cdot 1$	$q_0$

$$d_0 = \boxed{\overline{x}}; \quad Z = q_1 q_0$$

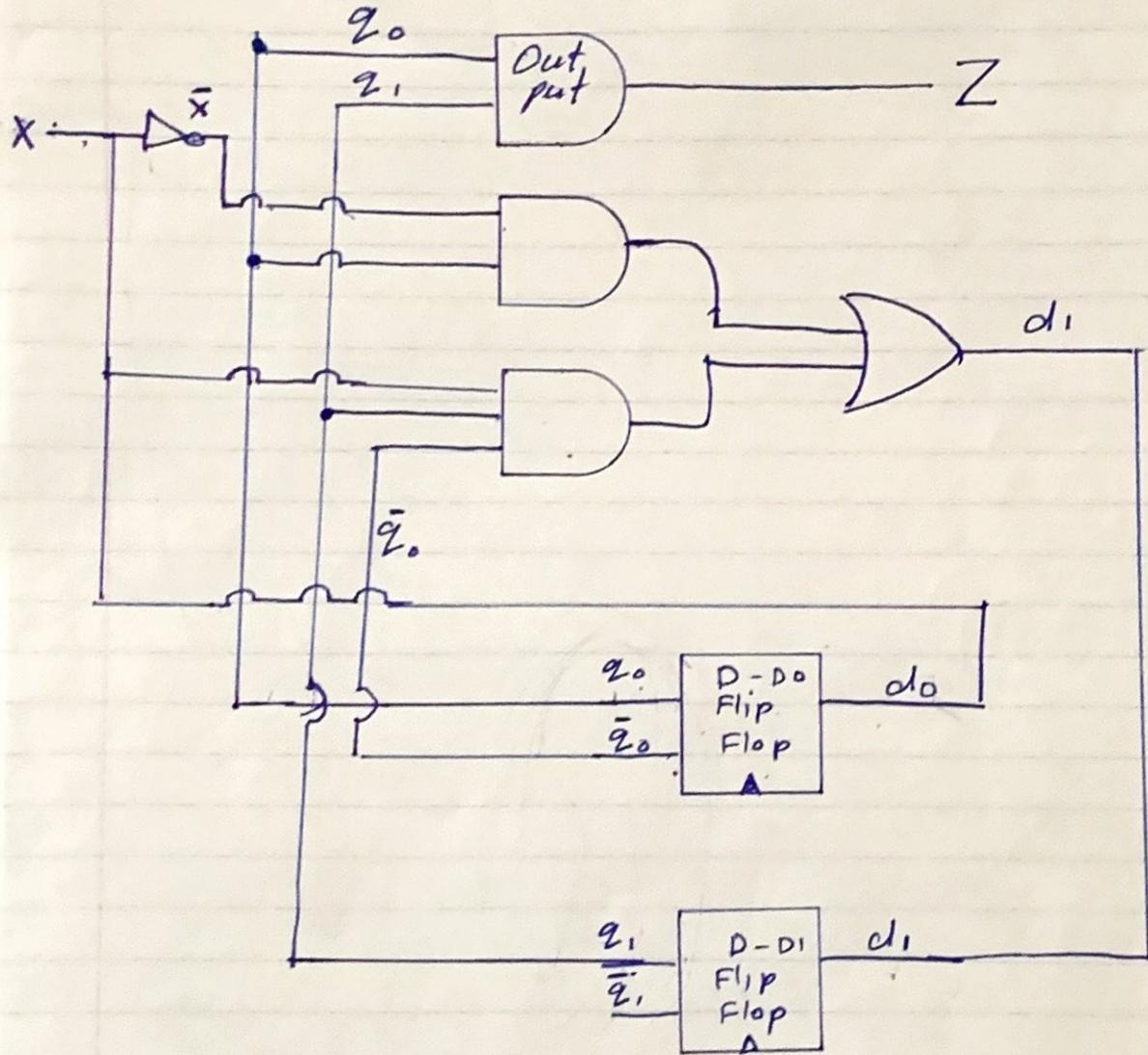
$$\overline{q}_1 q_0 + q_1 \bar{q}_0 \Leftrightarrow q_0$$

$$q_0 = 0 \Rightarrow \text{whole expression} = 0$$

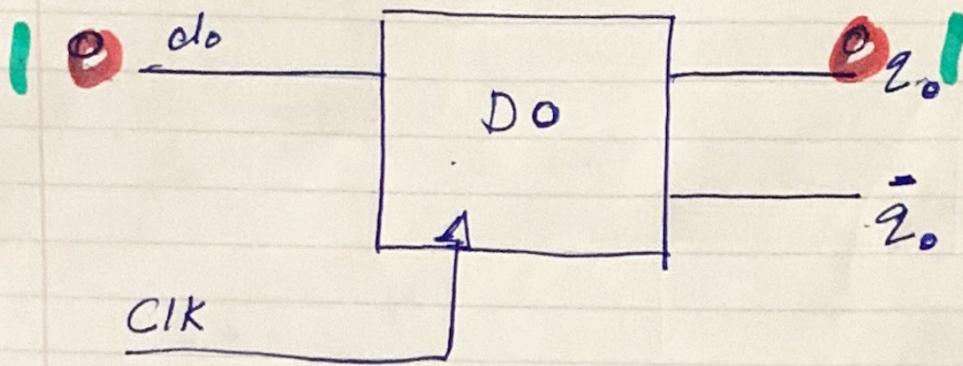
$$\overline{q}_1 \cdot 0 + q_1 \cdot 0 =$$

Design of a Moore FSM that detects "101" Sequence

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



# D flip Flops



$Clk = \text{Clock}$

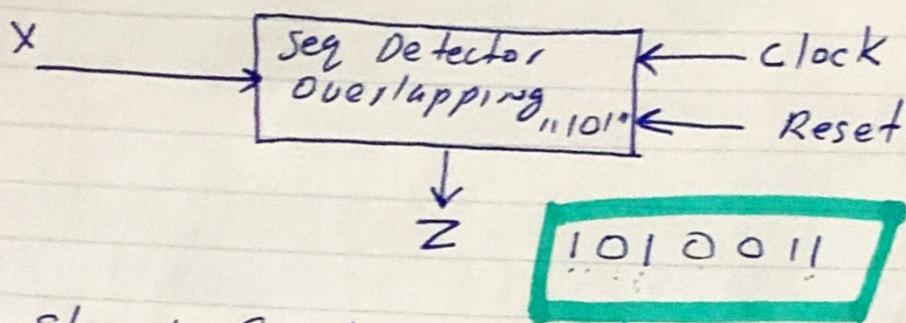
$d_o = \text{Input (1 bit)}$

$q_o = \text{Output}$

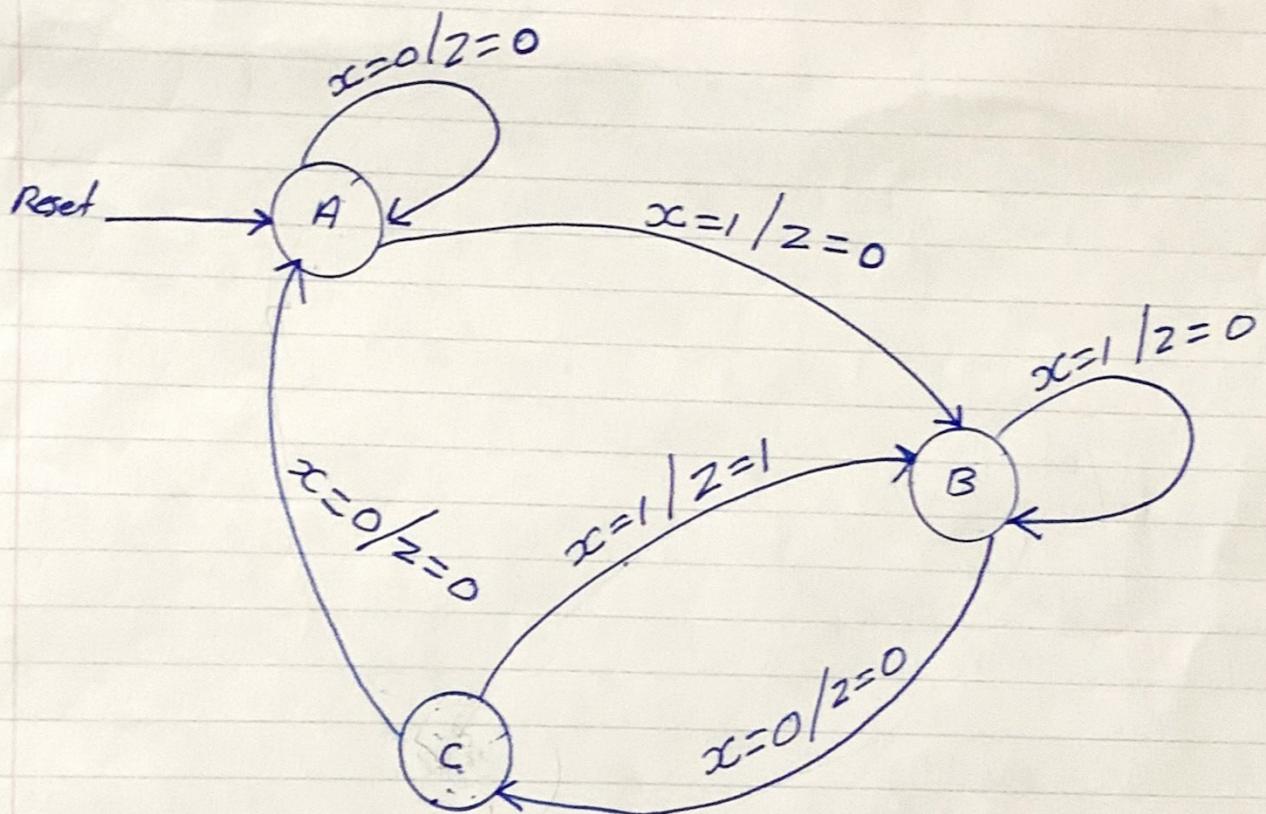
$\bar{q}_o = \text{Inverted Output}$

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 $z_1 z_0$	Input X	Next state $d_1 d_0$		Z
			d <sub>1</sub>	d <sub>0</sub>	
1	A	0 0	0	0 0	A
2		0 0	1	0 1	B
3	B	0 1	0	1 0	C
4		0 1	1	0 1	B
5	C	1 0	0	0 0	A
6		1 0	1	0 1	B
7	D	1 1	0	d d	d
8		1 1	1	d d	d

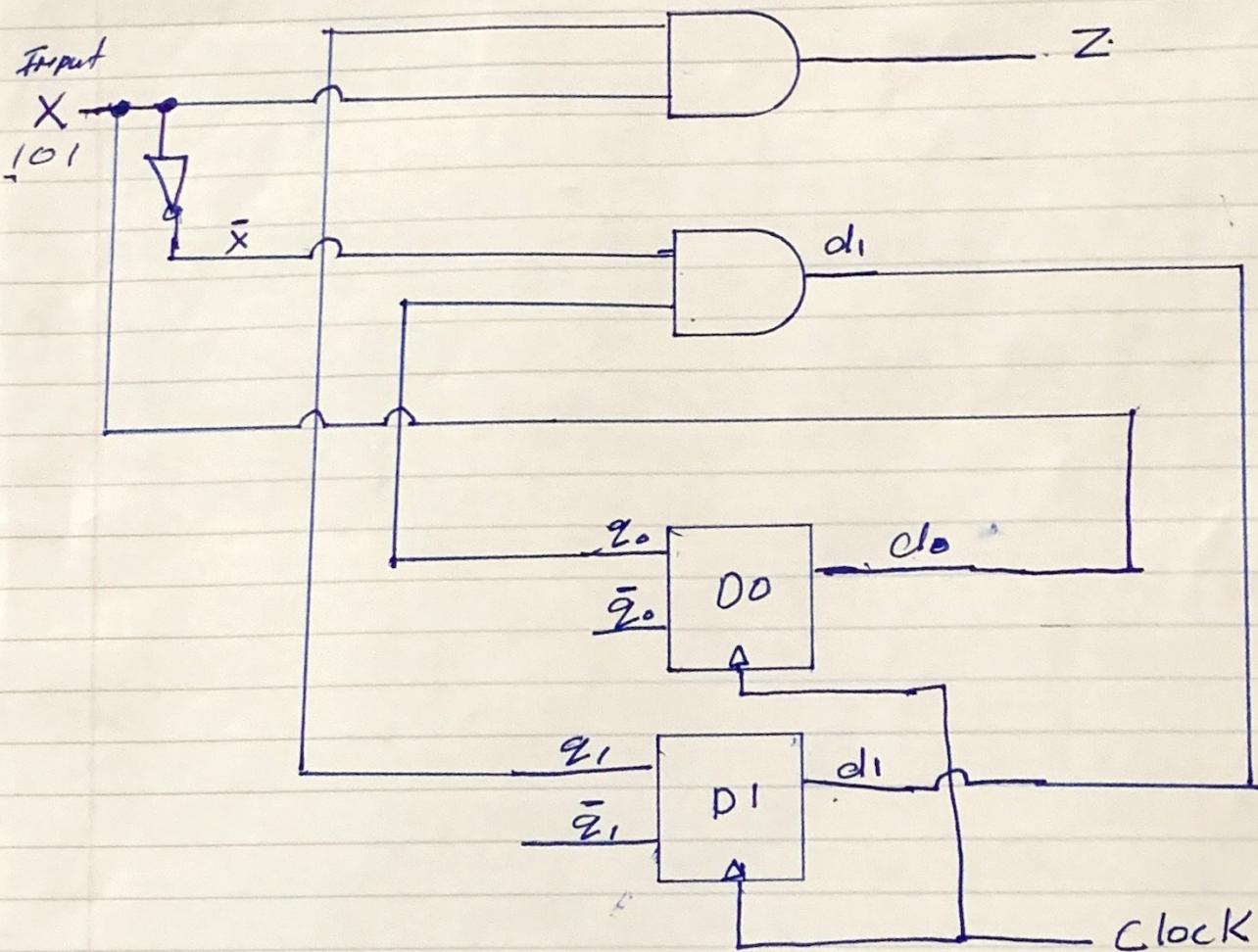
Design of Mealy FSM - Overlapping Sequence "101"

Step 4: Determine the logical Expression

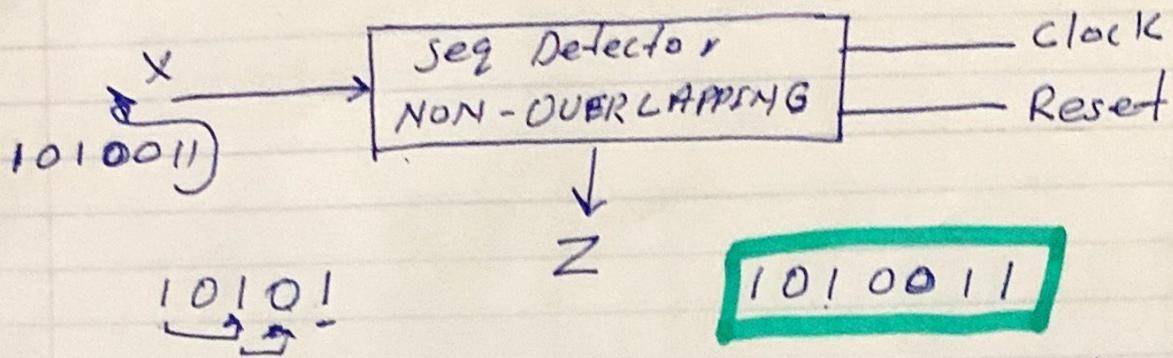
$$d_1 = q_0 \bar{x} ; d_1 = q_0 \bar{q}_1 \bar{x}$$

$$d_0 = x \quad z = q_1 x$$

Step 5: Draw the Circuit Diagram



Example: Design of Mealy Finite State machine that detects NON-OVERLAPPING Sequence "101"



Step 1: Create a mealy FSD - Finite state Diagram.

Let  $A = 00$ ,  $B = 01$ ,  $C = 10$ ,  $D = 11$

