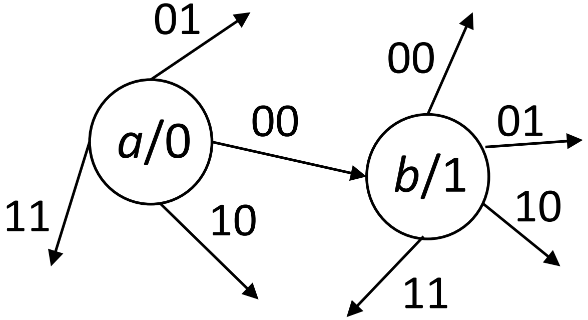
Digital Electronic Circuits
Section 1 (EE, IE)

Lecture 27

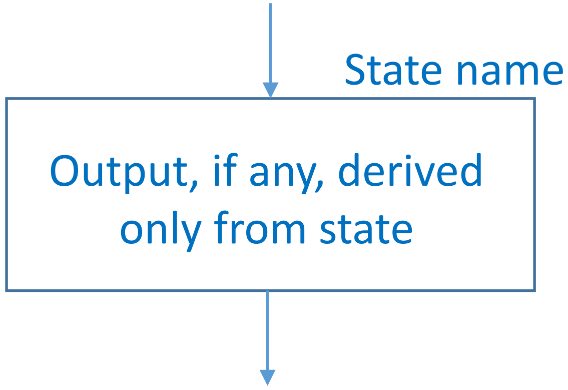
Algorithmic State Machine (ASM) Chart

- State Transition Diagram though compact is unsuitable for describing large state machines.
- Algorithmic State Machine (ASM) chart is a flow diagram like representation to design sequential digital electronic circuits.



n input: 2^n branch at each state

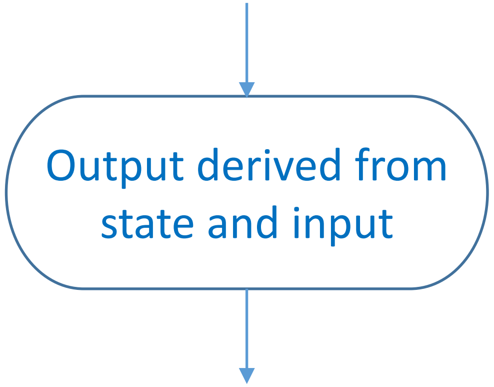
Basic Components:



State name

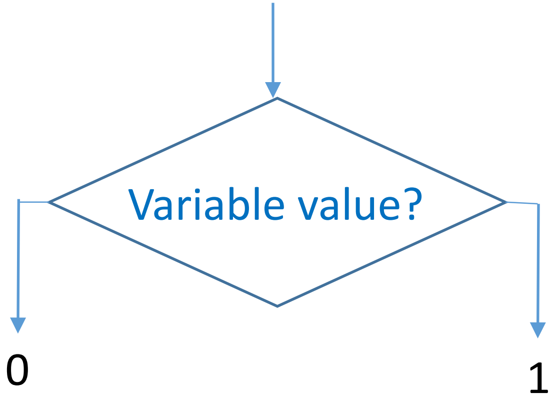
Output, if any, derived only from state

State Box



Output derived from state and input

Conditional Output Box



Variable value?

Decision Box

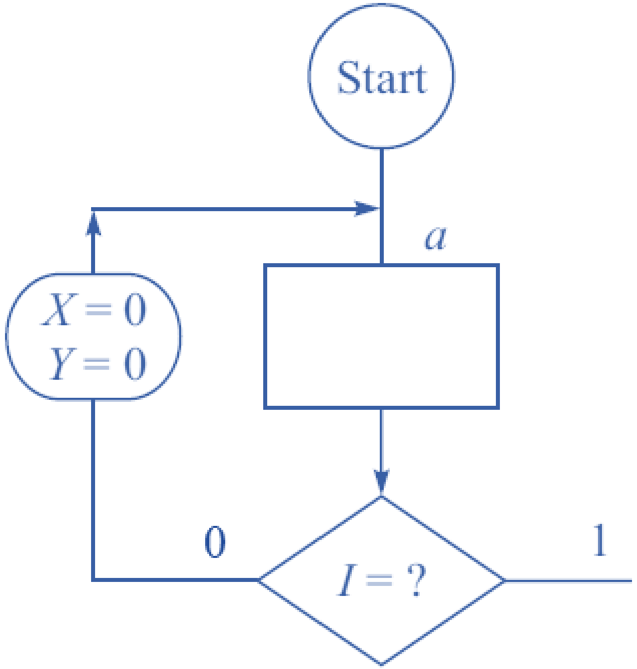
ASM Chart: Mealy Model - Part 1

Problem Statement:

Design circuit for a vending machine that takes only Rs. 5 and Rs. 10 coin as inputs to deliver a product that is priced Rs. 15. Coin sensing is as follows.

I	J	Activity
0	X	No coin deposited
1	0	Rs. 5 deposited
1	1	Rs. 10 deposited

Other than the output, *X* for product , there is another output, *Y* to return Rs. 5 if Rs. 20 is received by the machine anyhow.



At every clock trigger, I is sensed. If I = 0, state is maintained. Both output are 0.

State Definition:

a: Initial state i.e. money accumulated is zero.

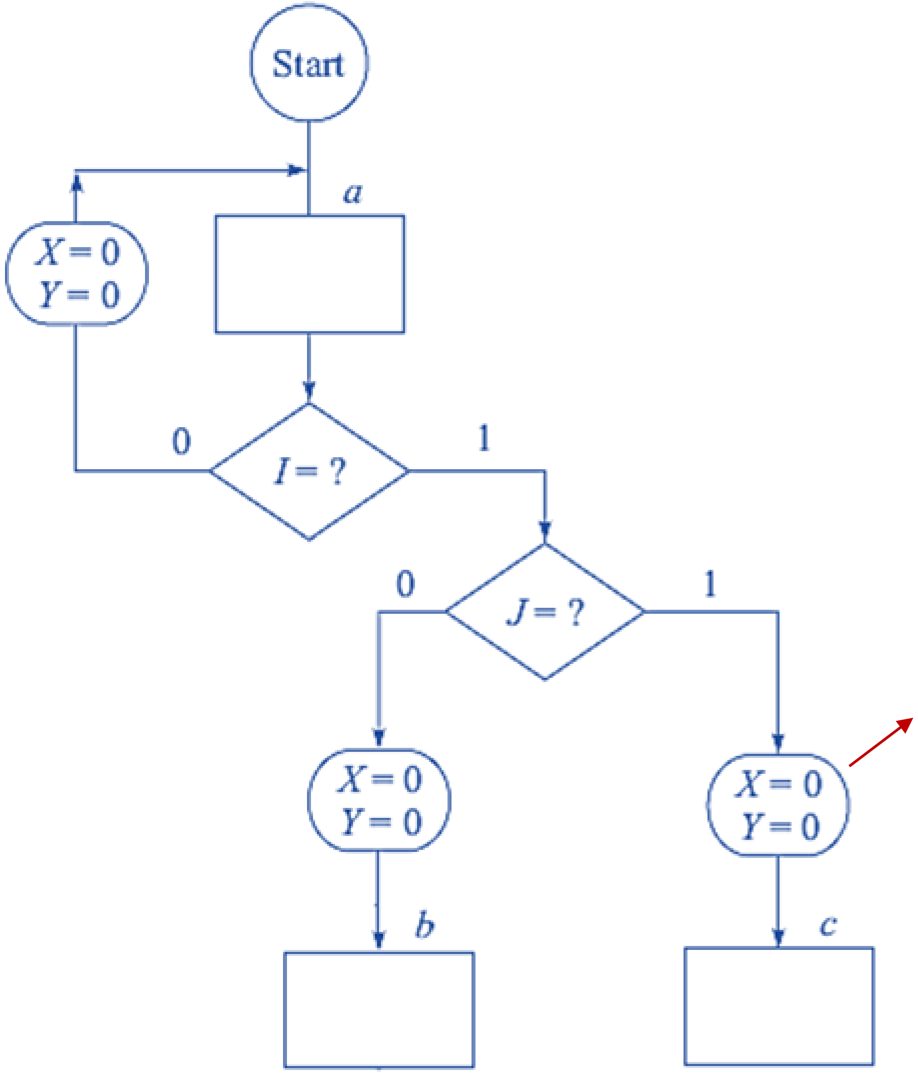
b: Rs. 5 accumulated.

c: Rs. 10 accumulated.

Part 2

At state a, If $I = 1$, a coin has been deposited.

Then decision is taken on $J = 0$ / $J = 1$ whether the circuit goes to state b / state c, at next clock trigger.



At state a, if $IJ = 11$, then, output $XY = 00$.

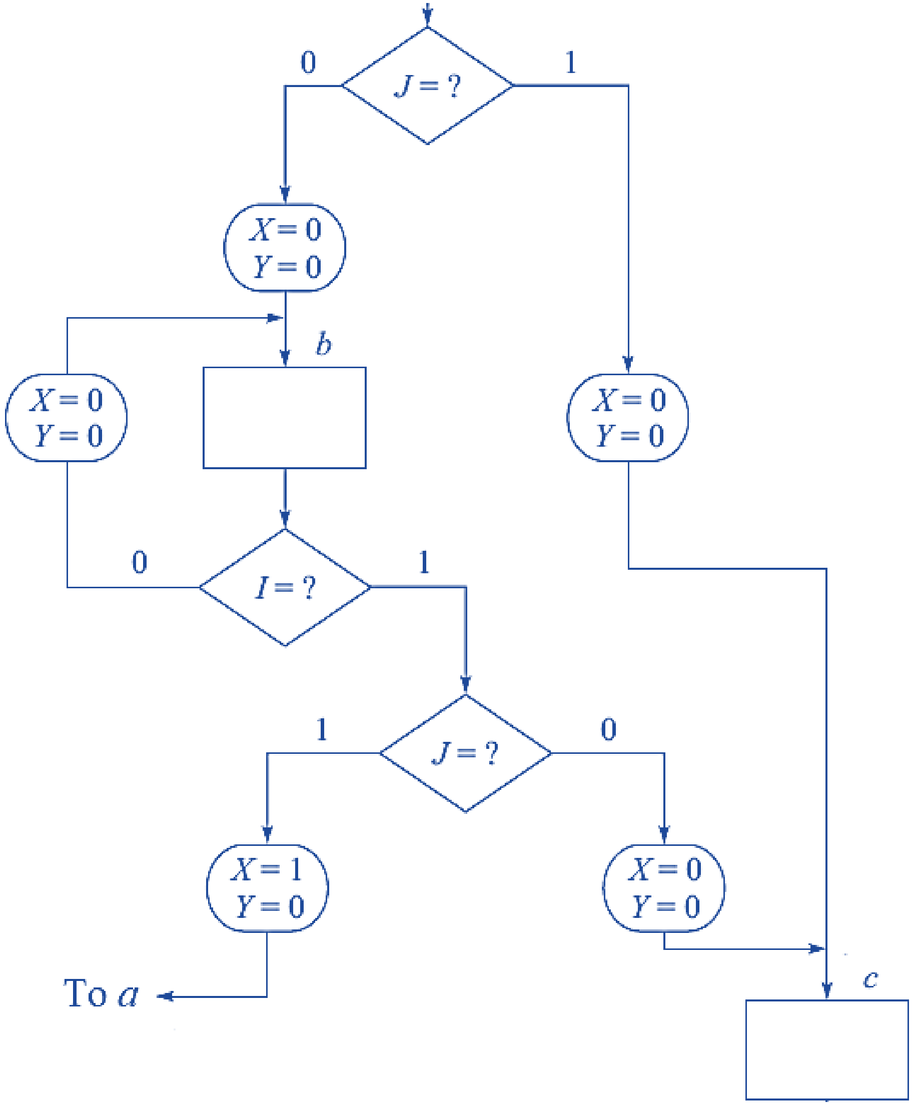
State Definition:

a: Initial state i.e. money accumulated is zero.
b: Rs. 5 accumulated.
c: Rs. 10 accumulated.

Part 3

At state *b*, If *I* = 1, a coin has been deposited.

Then decision is taken on *J* = 0 / *J* = 1 whether the circuit goes to state *c* with *XY* = 00 / delivers product with *XY* = 10 and goes to state *a*, at next clock trigger.



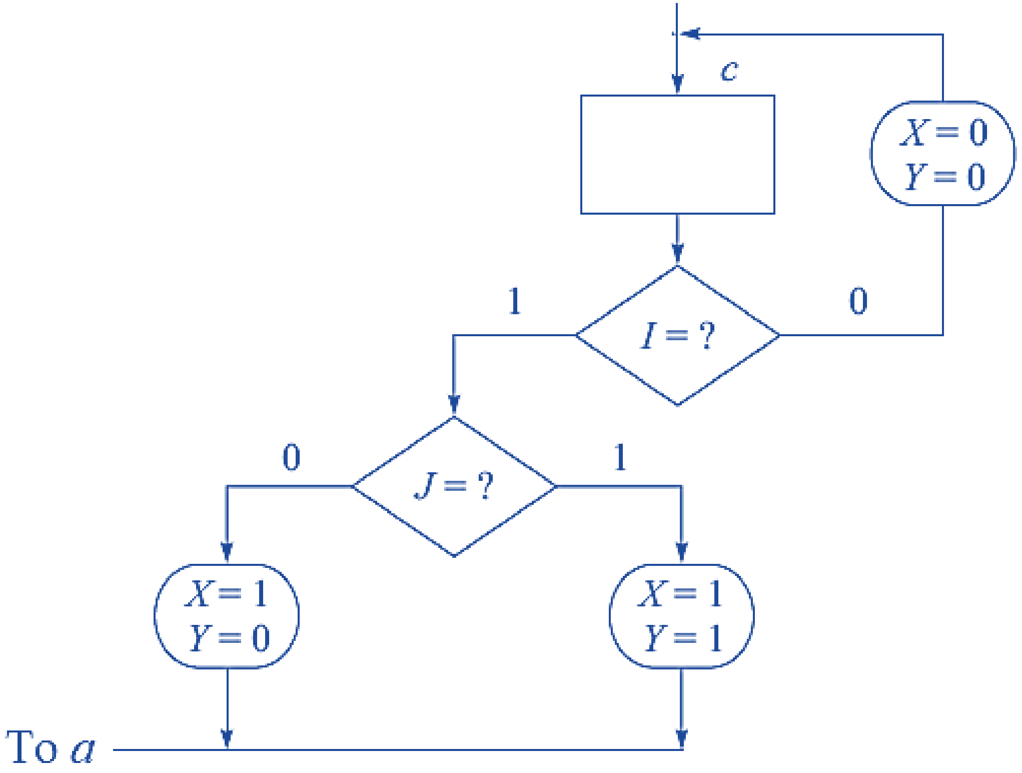
State Definition:

a: Initial state i.e. money accumulated is zero.
b: Rs. 5 accumulated.
c: Rs. 10 accumulated.

Part 4

At state *c*, If I = 1, a coin has been deposited.

Then decision is taken on J = 0 / J = 1 whether only product is delivered with *XY* = 10 to go to state *a* / product delivery and Rs. 5 return happen with *XY* = 11 and circuit goes to state *a*, at next clock trigger.

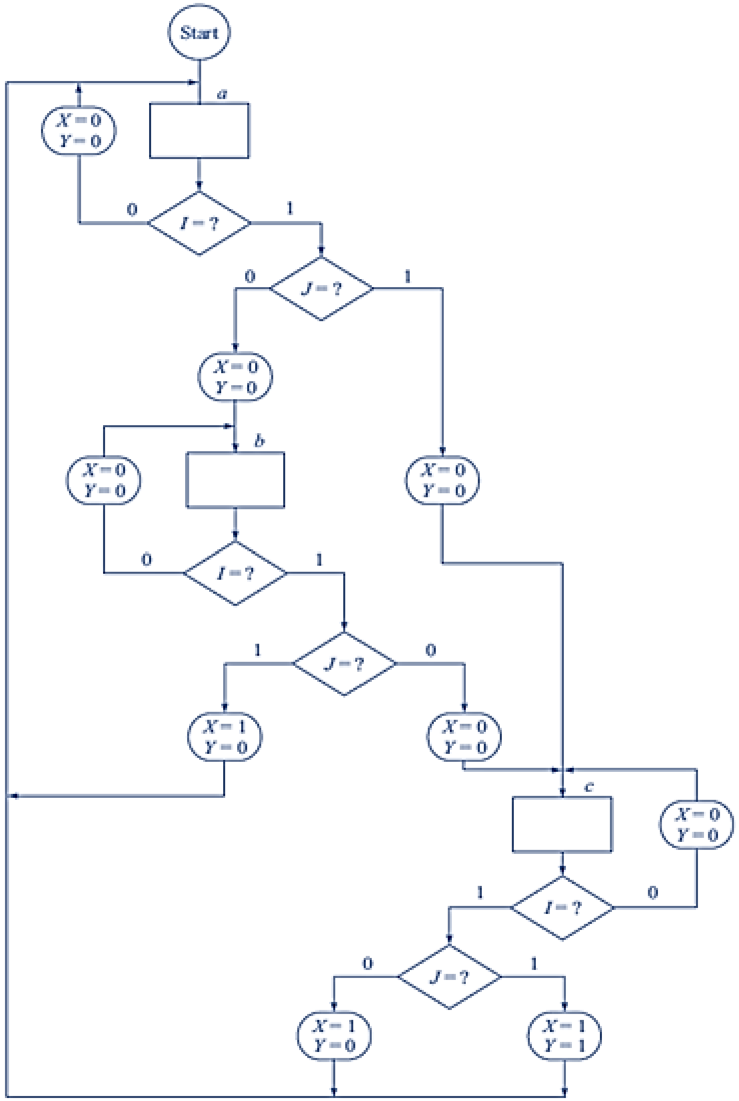


State Definition:

a: Initial state i.e. money accumulated is zero.
b: Rs. 5 accumulated.
c: Rs. 10 accumulated.

Full ASM Chart

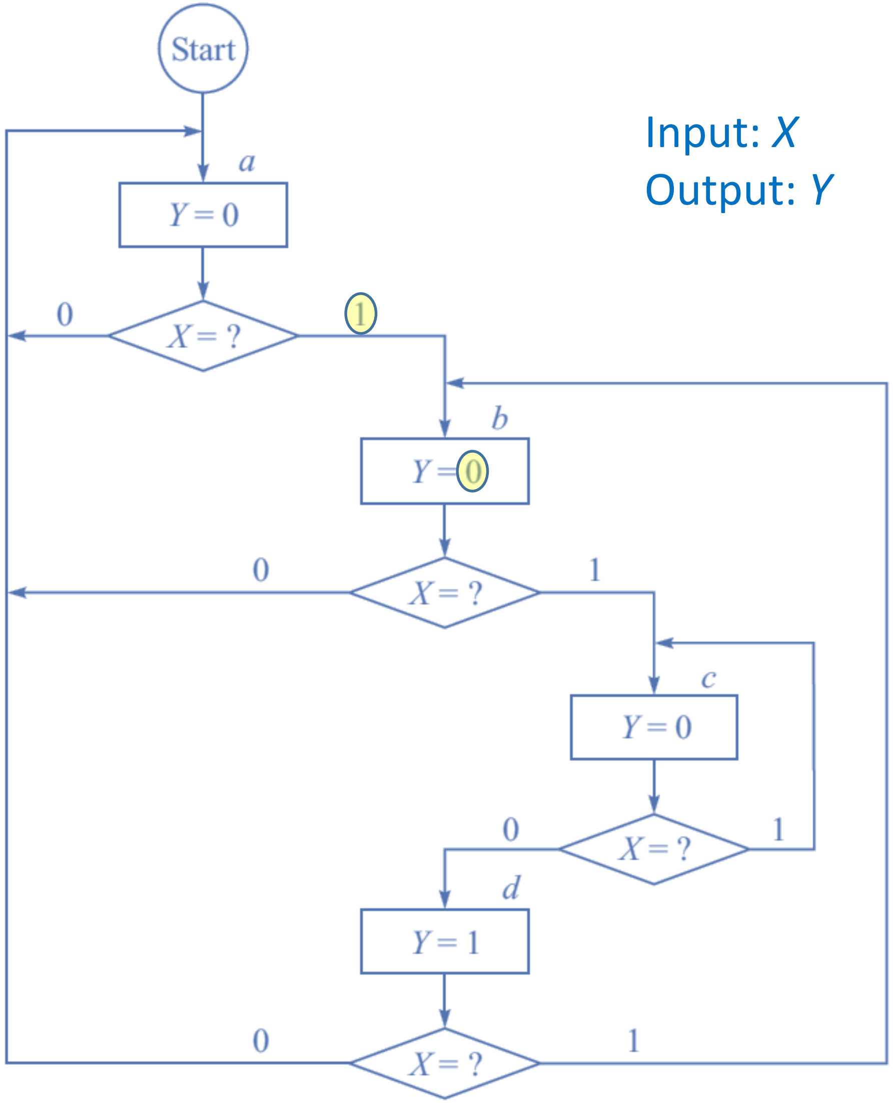
Full ASM Chart for the vending machine example.



ASM Chart: Moore Model

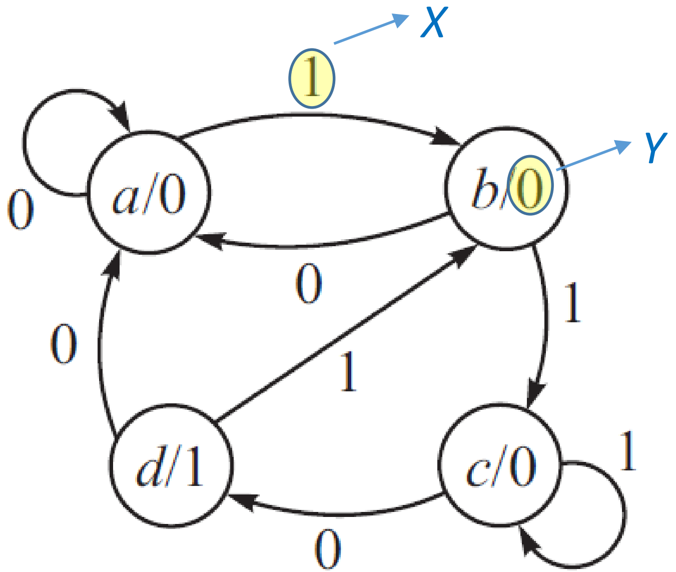
Problem Statement:

A sequence detector for ‘110’ from a binary data stream is to be designed.



ASM Chart

Input: X
Output: Y



State Transition Diagram

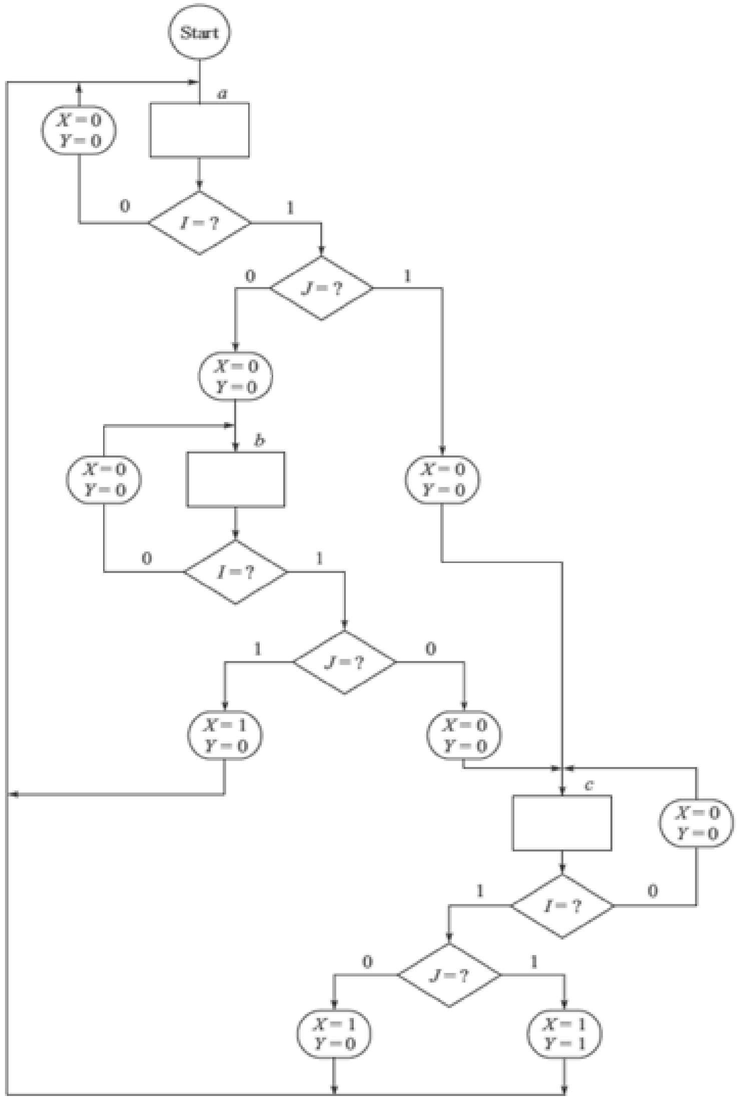
State Assignment

Problem Statement:

Design circuit for a vending machine that takes only Rs. 5 and Rs. 10 coin as inputs to deliver a product that is priced Rs. 15. Coin sensing is as follows.

I	J	Activity
0	X	No coin deposited
1	0	Rs. 5 deposited
1	1	Rs. 10 deposited

Other than the output, X for product , there is another output, Y to return Rs. 5 if Rs. 20 is reached anyhow.

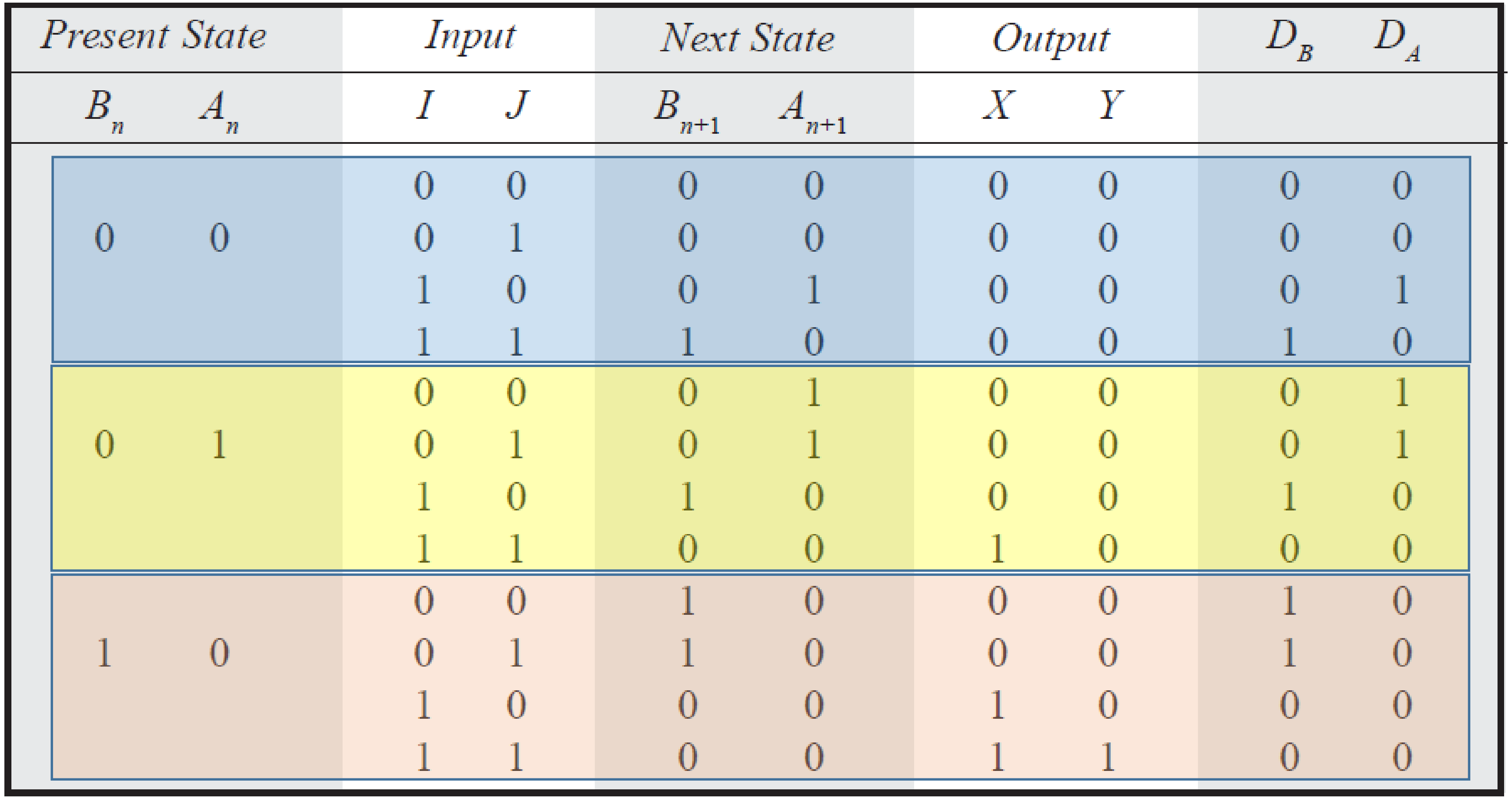


a: Initial state i.e. money accumulated is zero.
b: Rs. 5 accumulated.
c: Rs. 10 accumulated.

Two flip-flops: *BA*

State	*B*	*A*
a	0	0
b	0	1
c	1	0

State Table



Present State		*Input*		*Next State*		*Output*		D_B	D_A
B_n	A_n	I	J	B_{n+1}	A_{n+1}	X	Y		
0	0	0	0	0	0	0	0	0	0
		0	1	0	0	0	0	0	0
		1	0	0	1	0	0	0	1
		1	1	1	0	0	0	1	0
0	1	0	0	0	1	0	0	0	1
		0	1	0	1	0	0	0	1
		1	0	1	0	0	0	1	0
		1	1	0	0	1	0	0	0
1	0	0	0	1	0	0	0	1	0
		0	1	1	0	0	0	1	0
		1	0	0	0	1	0	0	0
		1	1	0	0	1	1	0	0

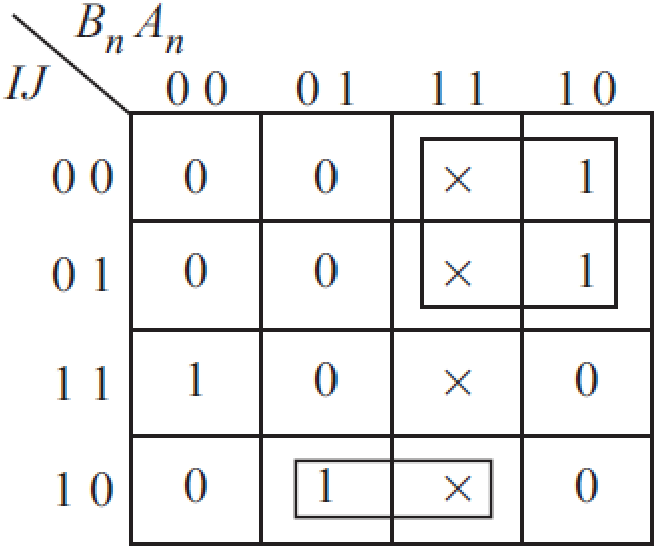
a: Initial state i.e. money accumulated is zero.

b: Rs. 5 accumulated.

c: Rs. 10 accumulated.

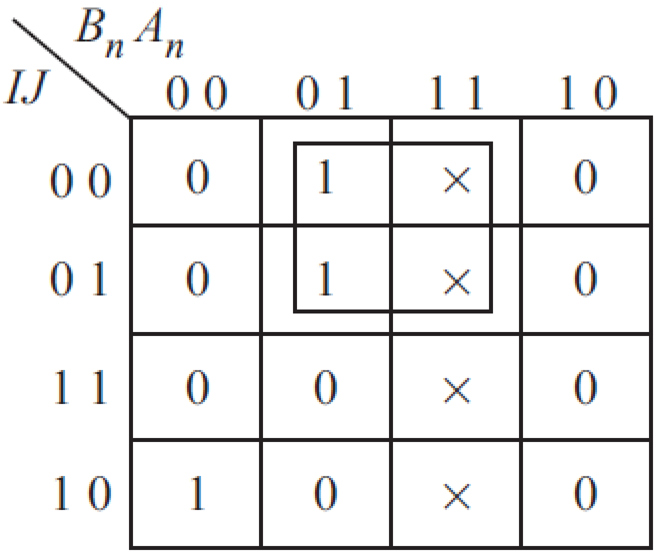
State	*B*	*A*
a	0	0
b	0	1
c	1	0

Design Equations



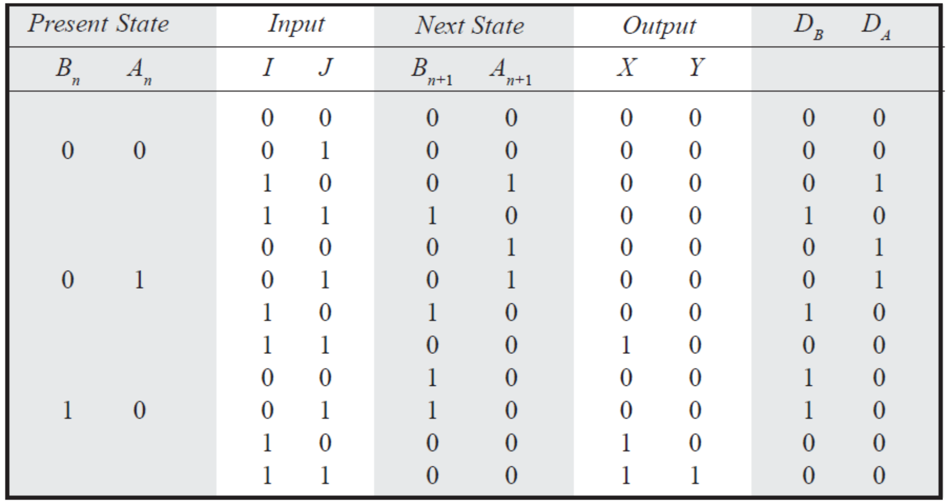
IJ \ B_nA_n	0 0	0 1	1 1	1 0
0 0	0	0	×	1
0 1	0	0	×	1
1 1	1	0	×	0
1 0	0	1	×	0

$$D_B = \bar{I}B_n + I\bar{J}A_n + IJ\bar{B}_n\bar{A}_n$$



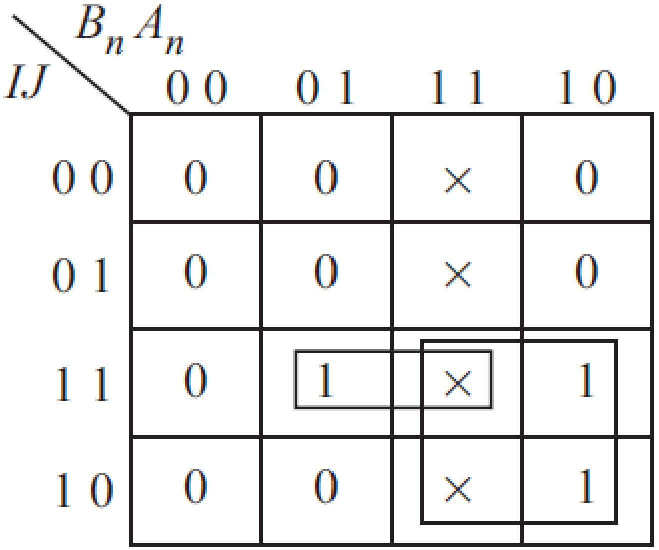
IJ \ B_nA_n	0 0	0 1	1 1	1 0
0 0	0	1	×	0
0 1	0	1	×	0
1 1	0	0	×	0
1 0	1	0	×	0

$$D_A = \bar{I}A_n + I\bar{J}\bar{B}_n\bar{A}_n$$



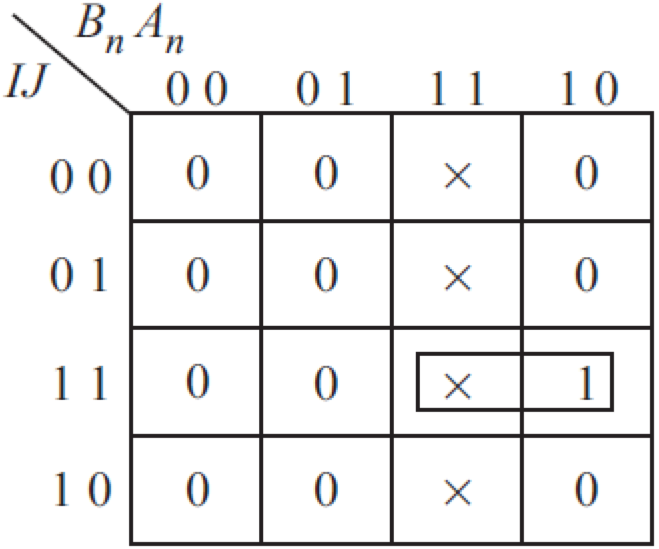
Present State		*Input*		*Next State*		*Output*		D_B	D_A
B_n	A_n	I	J	B_{n+1}	A_{n+1}	X	Y		
0	0	0	0	0	0	0	0	0	0
		0	1	0	0	0	0	0	0
		1	0	0	1	0	0	0	1
		1	1	1	0	0	0	1	0
0	1	0	0	0	1	0	0	0	1
		0	1	0	1	0	0	0	1
		1	0	1	0	0	0	1	0
		1	1	0	0	1	0	0	0
1	0	0	0	1	0	0	0	1	0
		0	1	1	0	0	0	1	0
		1	0	0	0	1	0	0	0
		1	1	0	0	1	1	0	0

Design Equations



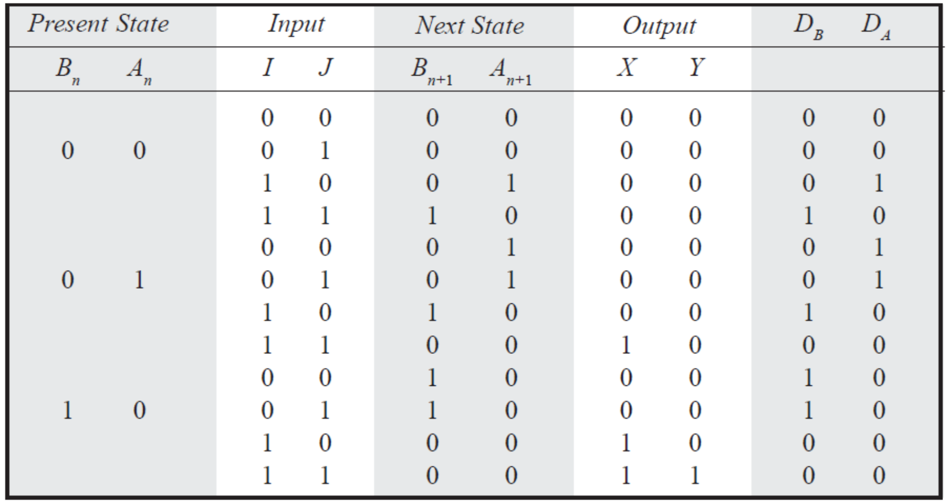
IJ \ $B_n A_n$	00	01	11	10
00	0	0	×	0
01	0	0	×	0
11	0	1	×	1
10	0	0	×	1

$$X = I B_n + I J A_n$$



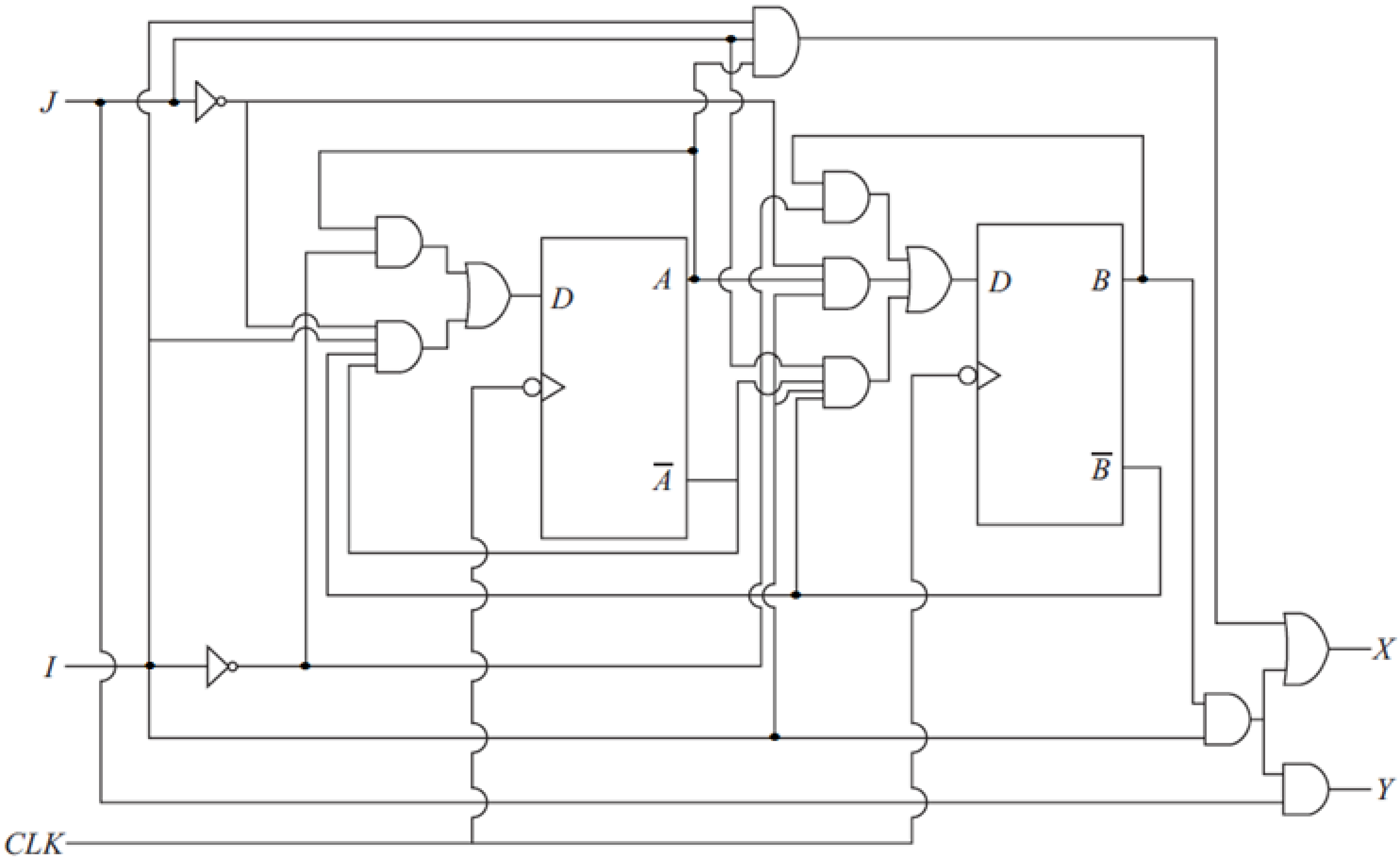
IJ \ $B_n A_n$	00	01	11	10
00	0	0	×	0
01	0	0	×	0
11	0	0	×	1
10	0	0	×	0

$$Y = I J B_n$$



Present State		*Input*		*Next State*		*Output*		D_B	D_A
B_n	A_n	I	J	B_{n+1}	A_{n+1}	X	Y		
0	0	0	0	0	0	0	0	0	0
		0	1	0	0	0	0	0	0
		1	0	0	1	0	0	0	1
		1	1	1	0	0	0	1	0
0	1	0	0	0	1	0	0	0	1
		0	1	0	1	0	0	0	1
		1	0	1	0	0	0	1	0
		1	1	0	0	1	0	0	0
1	0	0	0	1	0	0	0	1	0
		0	1	1	0	0	0	1	0
		1	0	0	0	1	0	0	0
		1	1	0	0	1	1	0	0

Circuit Diagram



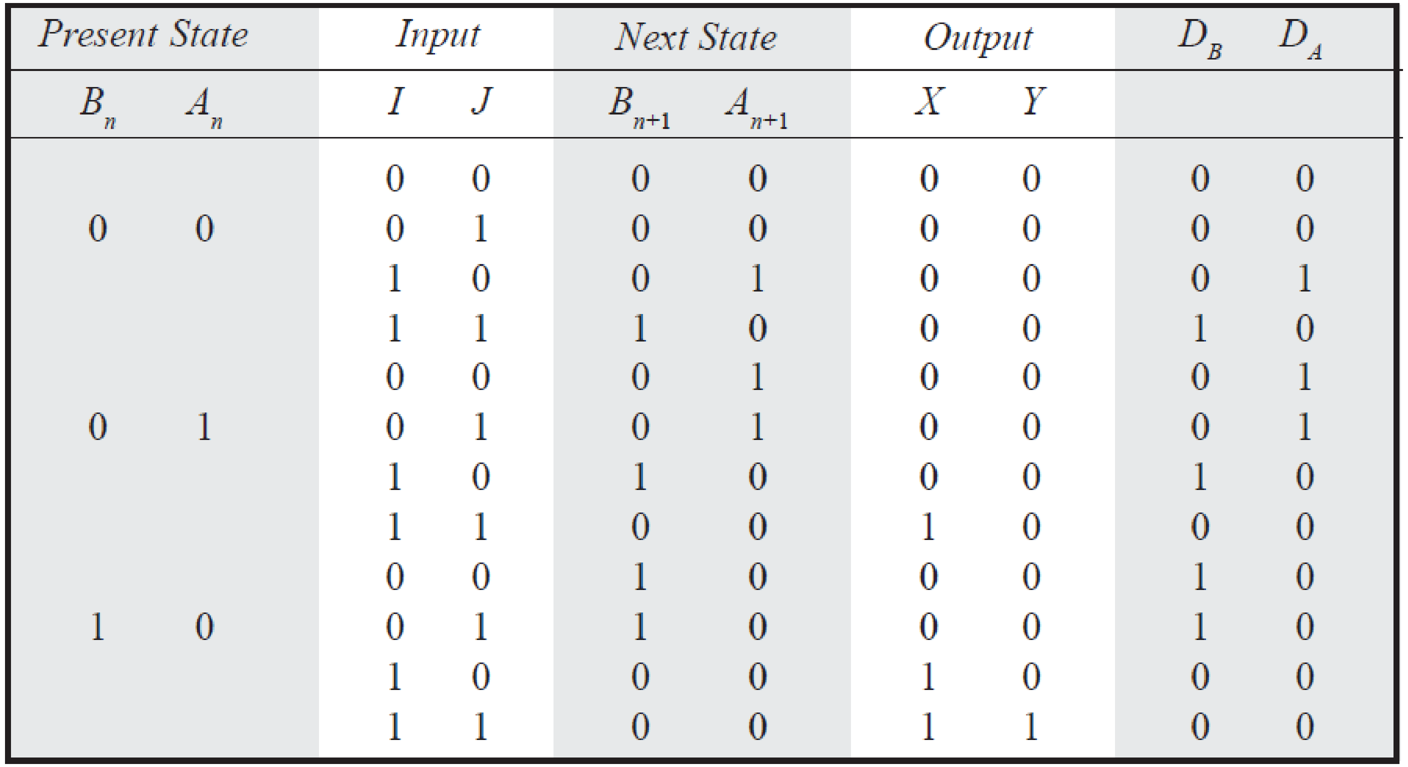
$$D_B = \overline{I}B_n + I\,\overline{J}\,A_n + I\,J\,\overline{B}_n\overline{A}_n$$

$$D_A = \overline{I}A_n + I\,\overline{J}\,\overline{B}_n\overline{A}_n$$

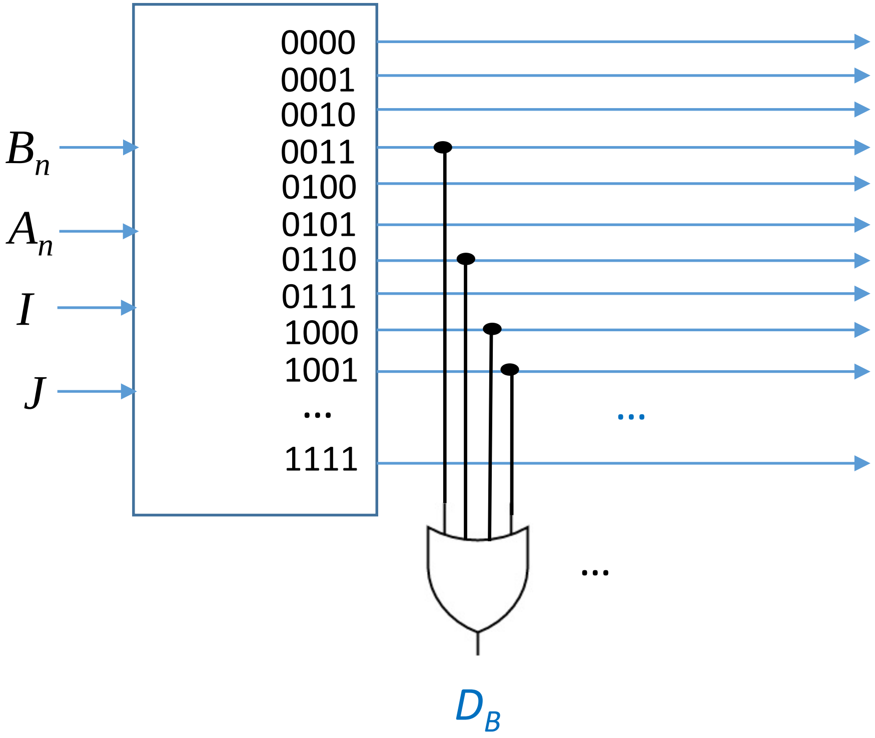
$$X = I\,B_n + I\,J\,A_n$$

$$Y = I\,J\,B_n$$

Use of Decoder-OR



Present State		*Input*		*Next State*		*Output*		D_B	D_A
B_n	A_n	I	J	B_{n+1}	A_{n+1}	X	Y		
0	0	0	0	0	0	0	0	0	0
		0	1	0	0	0	0	0	0
		1	0	0	1	0	0	0	1
		1	1	1	0	0	0	1	0
0	1	0	0	0	1	0	0	0	1
		0	1	0	1	0	0	0	1
		1	0	1	0	0	0	1	0
		1	1	0	0	1	0	0	0
1	0	0	0	1	0	0	0	1	0
		0	1	1	0	0	0	1	0
		1	0	0	0	1	0	0	0
		1	1	0	0	1	1	0	0



$D_B = F(B_n, A_n, I, J) = \sum m(3,6,8,9)$ $D_A = \sum m(2,4,5)$ $X = \sum m(7,10,11)$ $Y = \sum m(11)$

References:

❑ Donald P. Leach, Albert P. Malvino, and Goutam Saha, Digital Principles & Applications 8e, McGraw Hill