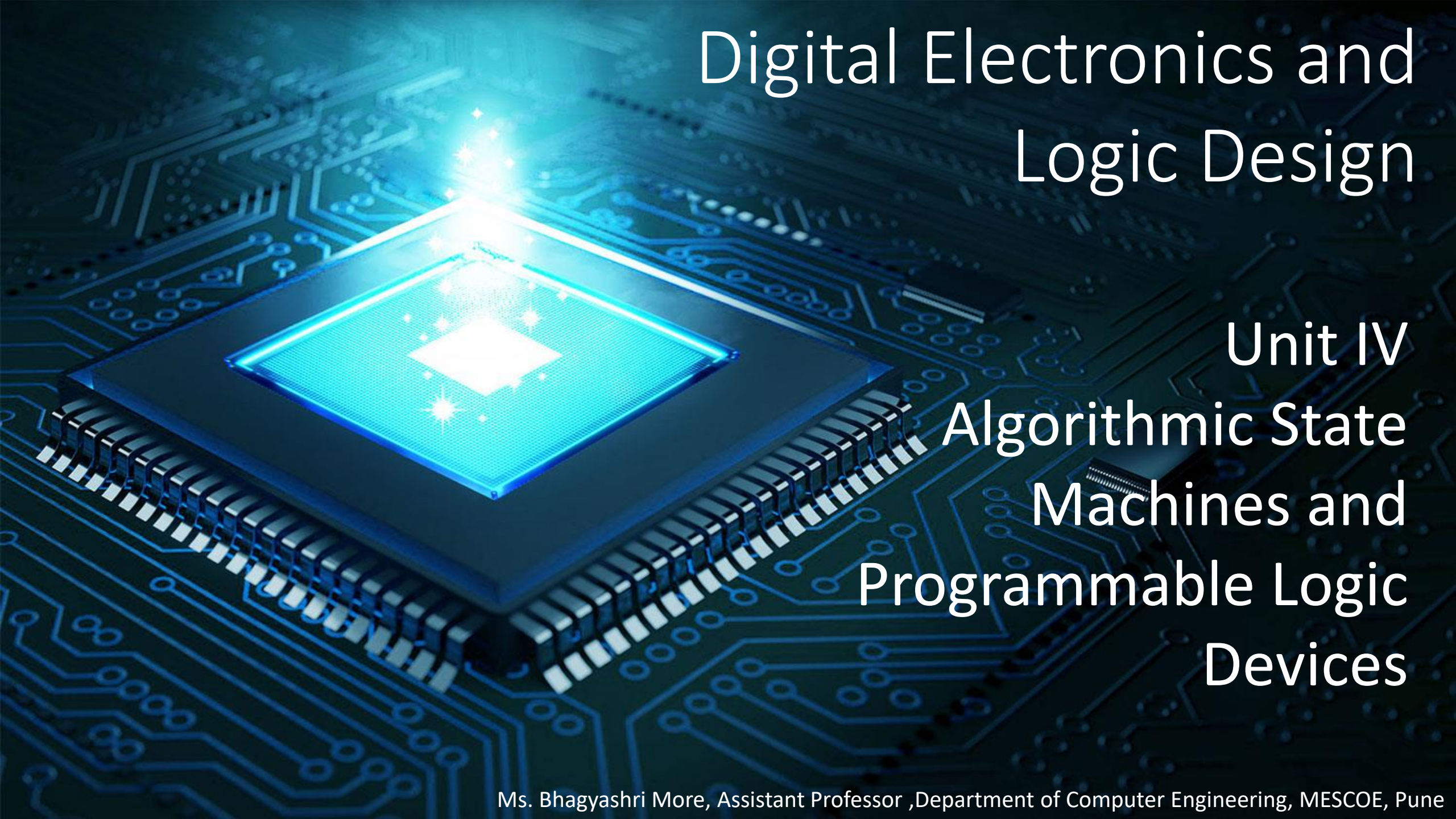


Digital Electronics and Logic Design

Unit IV
Algorithmic State Machines and Programmable Logic Devices

Ms. Bhagyashri More, Assistant Professor ,Department of Computer Engineering, MESCOE, Pune

Agenda

01

Algorithmic State Machine: Finite state machine (FSM) and ASM, ASM charts, notations, Construction of ASM chart and realization of sequential circuits.

02

PLDs: PLD, ROM as PLD, Programmable Logic Array (PLA), Programmable Array Logic (PAL), Designing Combinational Circuits using PLDs.

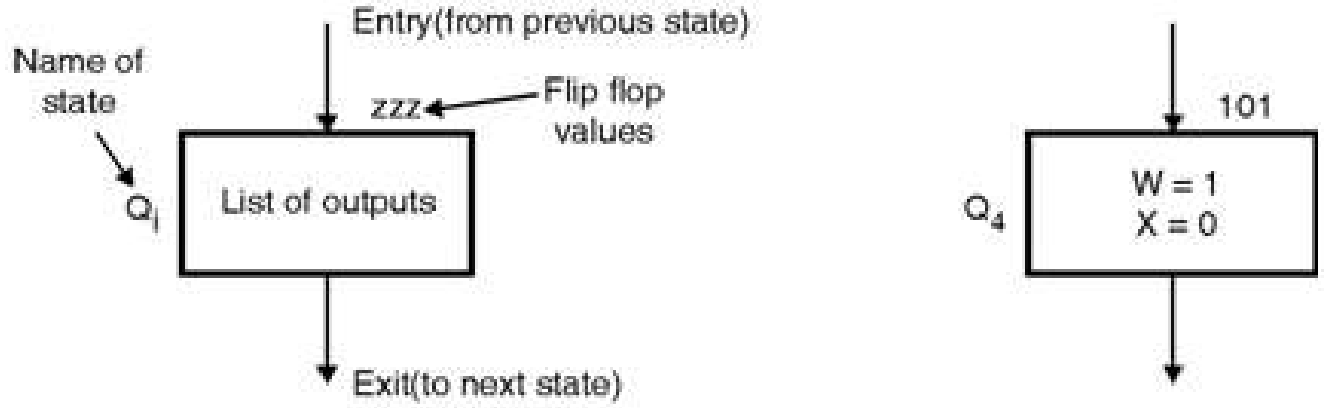
Algorithmic State Machine

1. ASM is an algorithm consists of a few steps, which is used to simplify a sequential digital system.
2. An ASM chart is resembles a conventional flow chart but the difference is, a conventional flow chart does not have timing relationships but the ASM takes timing relationship into account.
3. An ASM chart describes the sequence of events as well as the timing relationship between the states of a sequential controller and the events that occur while going from one state to the other.
4. It is employed to design a sequential circuit having a large number of external inputs because with a large number of external inputs it becomes very difficult to use state tables for designing the circuit.
5. ASM Chart Notations : The different blocks used in the ASM chart are,

- The state box
- The decision box
- The conditional box

Algorithmic State Machine

State Box: The state box is used to indicate the control sequence in the state. The general description and typical example of state box is as shown in Figure below.

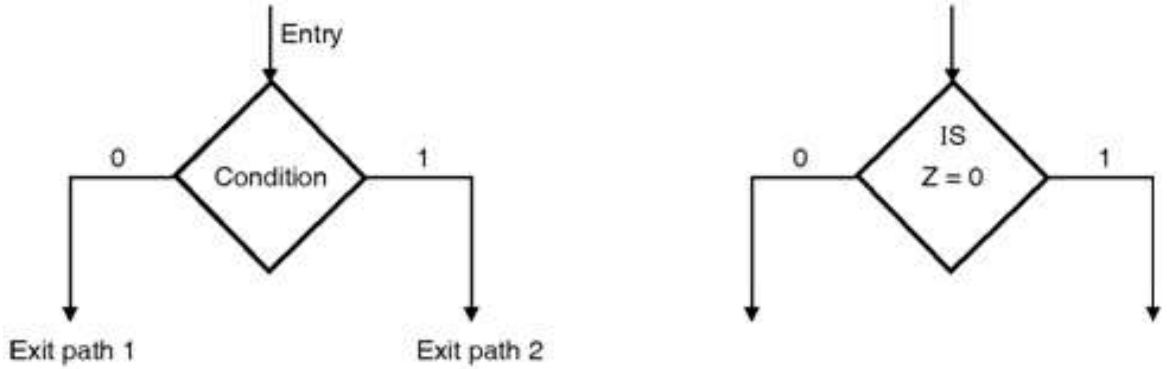


State box description :

- A state box is a rectangular in shape.
- The circuit outputs that can occur whenever the circuit is in the corresponding state regardless of input values are listed inside the box.
- On the right hand top corner of the box the list of flip flop values which represents state are written.
- The name of the state for example, Qi, Si, etc. can be written at the left corner of the rectangular box.
- The input to the state box is indicated by Entry•and output to next state is denoted by Exit.

Algorithmic State Machine

Decision Box: The decision box defines the decision based on certain condition which shows the effect of an input on the control system. The general description and typical example of decision box is as shown in Figure below.

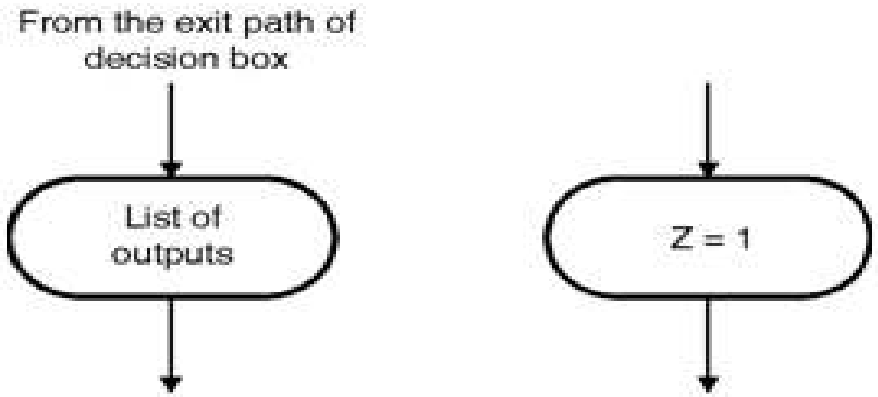


Decision box description :

- A decision box is a diamond shape box.
- The decision box has one entry and two exit paths depends upon the whether the input condition is true or false.
- The input condition to be tested is written inside the diamond box.
- Each value either true or false depend upon an expression inside the diamond box leads with exit paths from that box.
- These paths are connected to the blocks corresponding to the next states of the circuit.

Algorithmic State Machine

Conditional Box : The conditional box element of the ASM chart is unique which is not used in conventional flow chart. The general description and typical example of conditional box is shown in Figure below.

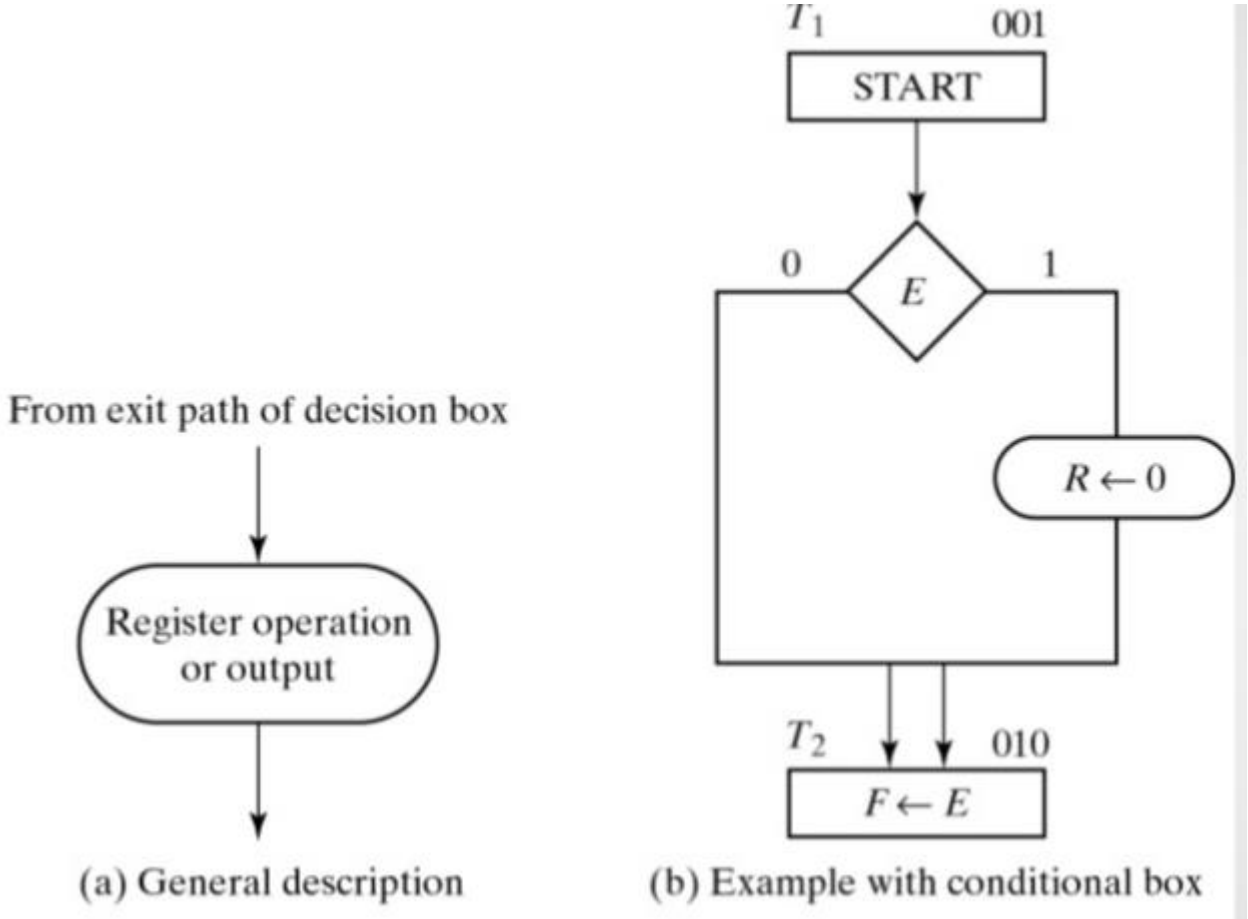


Conditional box description :

- A conditional box is a oval shape box.
- The difference between state box and conditional box is only the rounded corner.
- The input path for the conditional box always comes from one of the exit paths of a decision box.
- The outputs produced are listed inside the conditional box are generated during a given state.

Algorithmic State Machine

Example of Conditional Box:

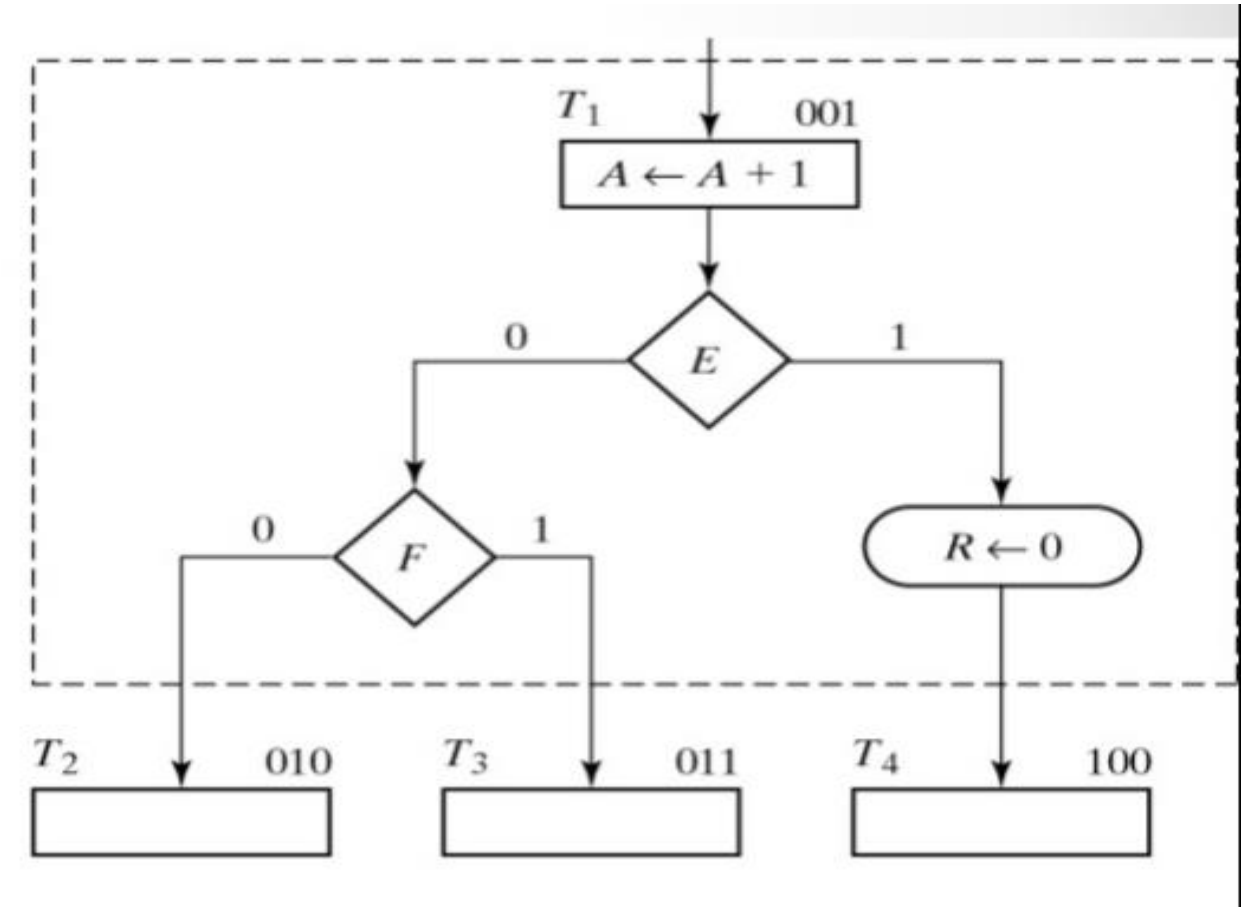


Algorithmic State Machine

Example:

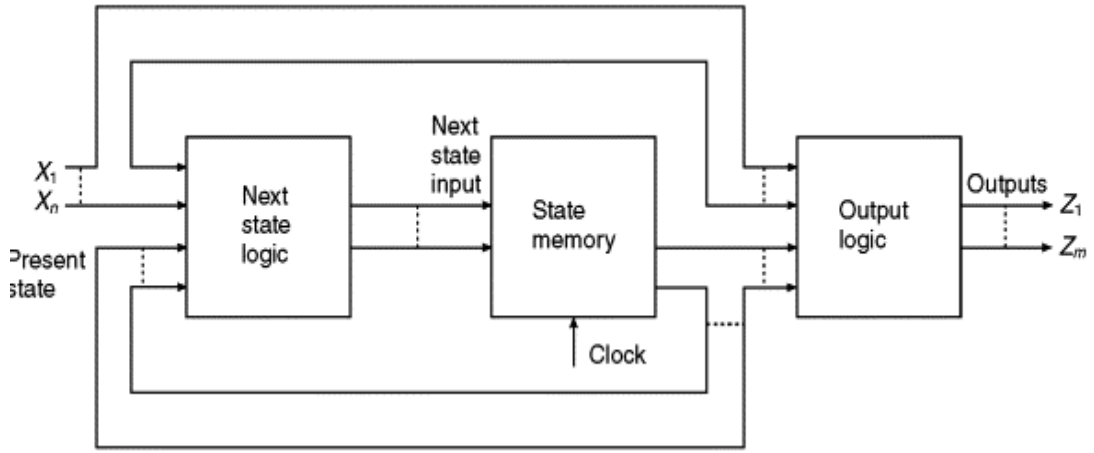
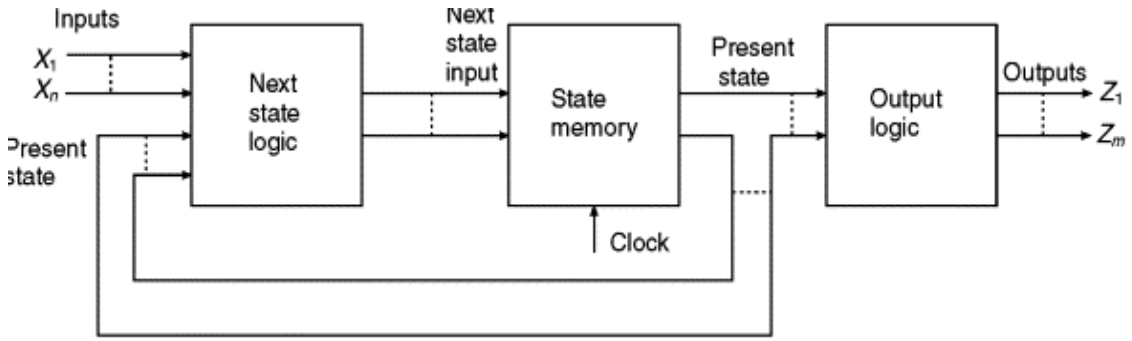
1. **Register A incremented.**
2. **Condition E evaluated.**
3. **Based on evaluation results**
4. **state,**
   T2 or T3 or T4 entered.

“All operations within block occurring during single edge of transition.”



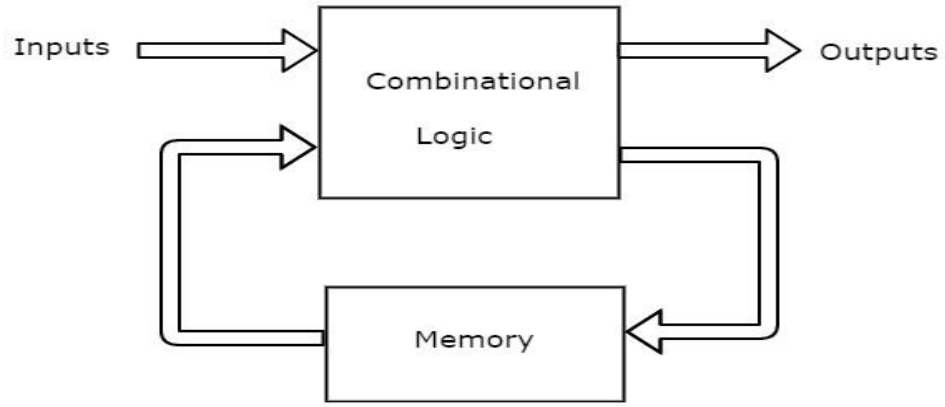
Finite State Machine

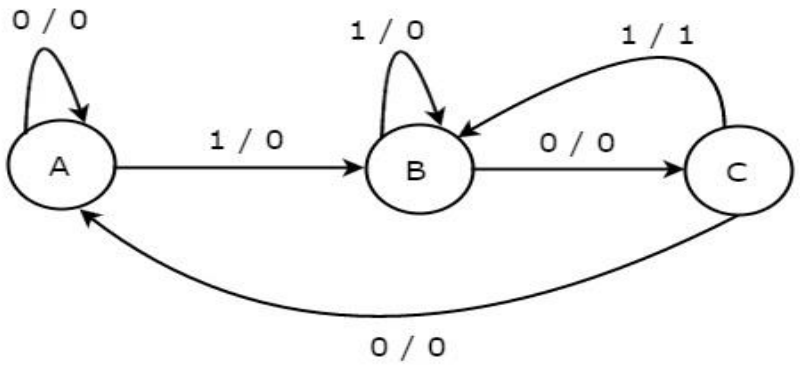
A synchronous sequential circuit is also called as **Finite State Machine** FSMFSM, if it has finite number of states. There are two types of FSMs.
Mealy State Machine

Mealy State Machine	Moore State Machine
	

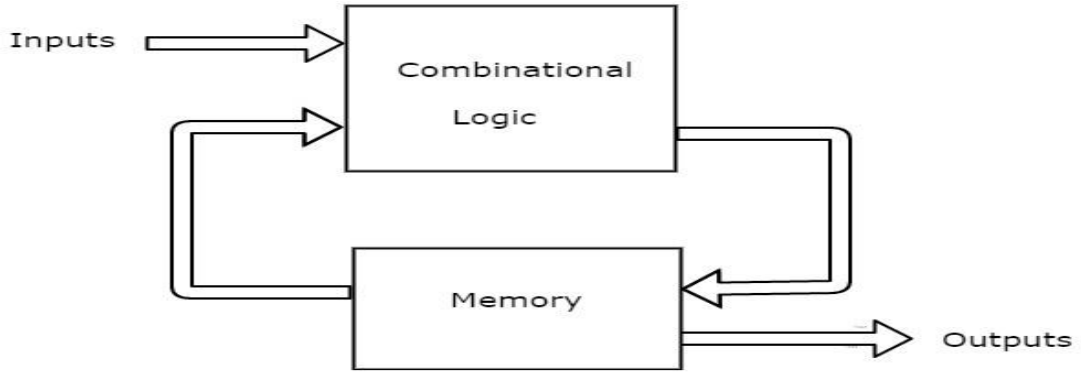
Finite State Machine

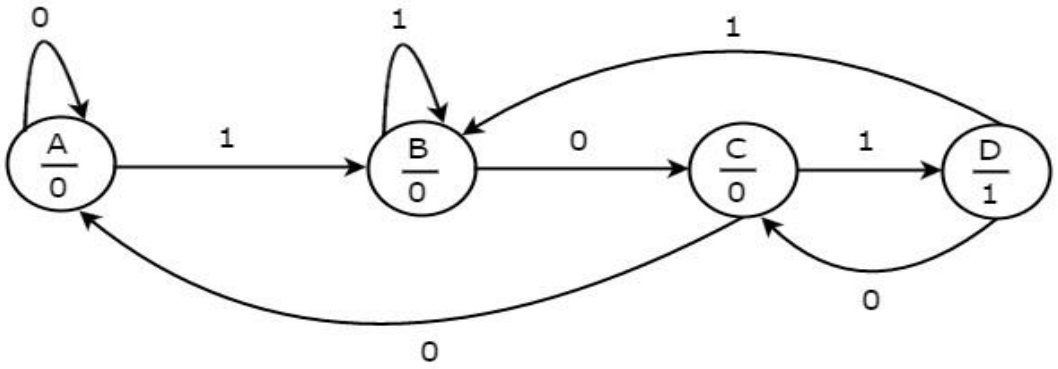
Mealy State Machine





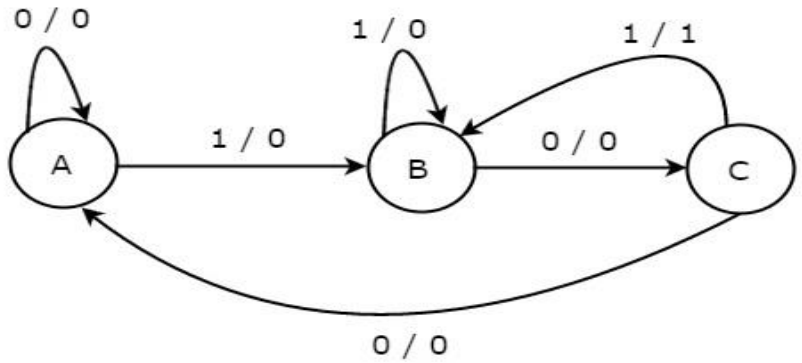
Moore State Machine





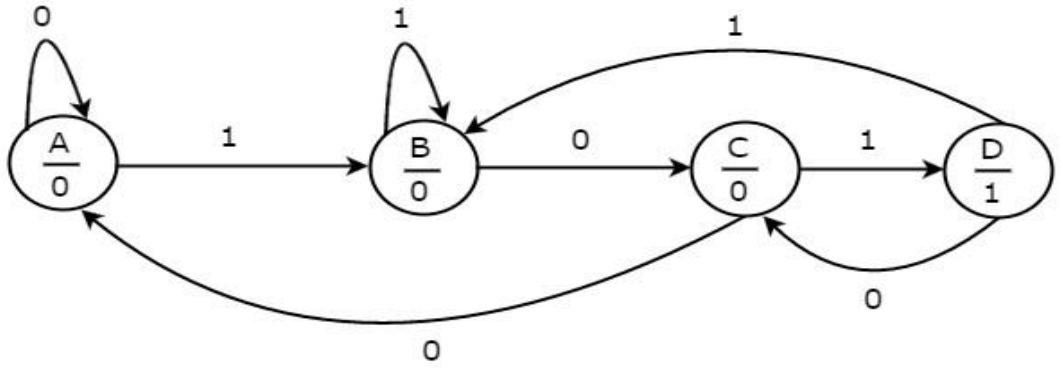
Finite State Machine

Mealy State Machine



P.S.	N.S		O/P Y	
	X=0	X=1	X=0	X=1
A	A	B	0	0
B	C	B	0	0
C	A	B	0	1

Moore State Machine



P.S.	N.S		O/P Y
	X=0	X=1	
A	A	B	0
B	C	B	0
C	A	D	0
D	C	B	1

Algorithmic State Machine

Example: Draw an ASM chart and state table for 2 bit up-down counter having mode control input M.

Solution:

As it is 2 bit up-down counter it has 2^2 = 4 state

Consider,

M=1 UP Counting

M=0 DOWN Counting

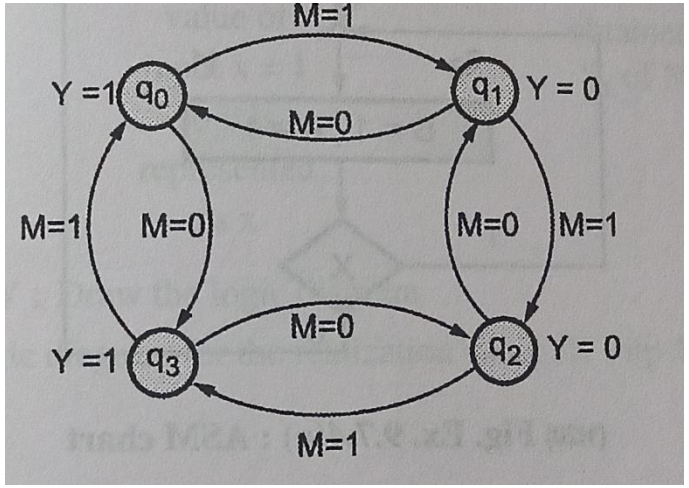
When count becomes minimum (00) or maximum (11) the circuit should be produce output=1.

As it is 2 bit counter, then there will be four states, q0 = 00,q1 = 01, q2 = 10, q3 = 11

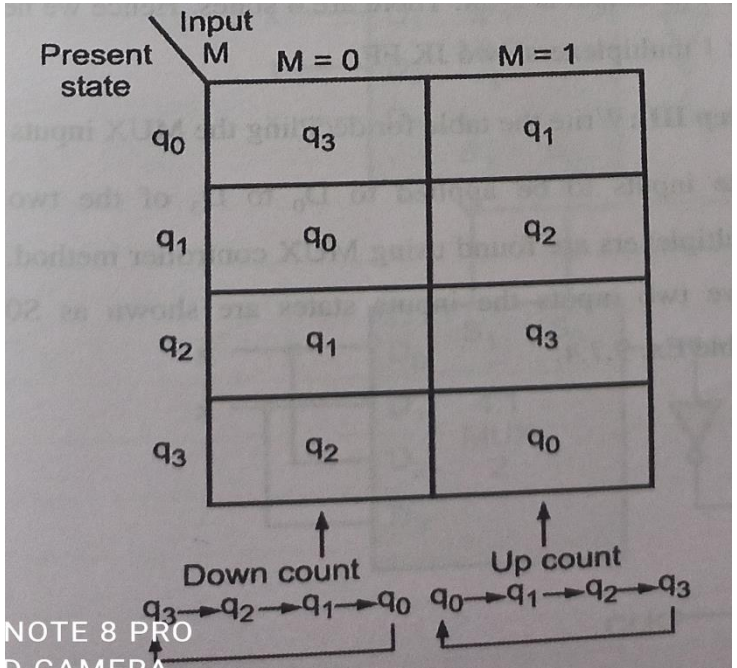
When M=1 the q0 -> q1 -> q2 -> q3 -> q0

When M=0 the q3 -> q2 -> q1 -> q0 -> q3

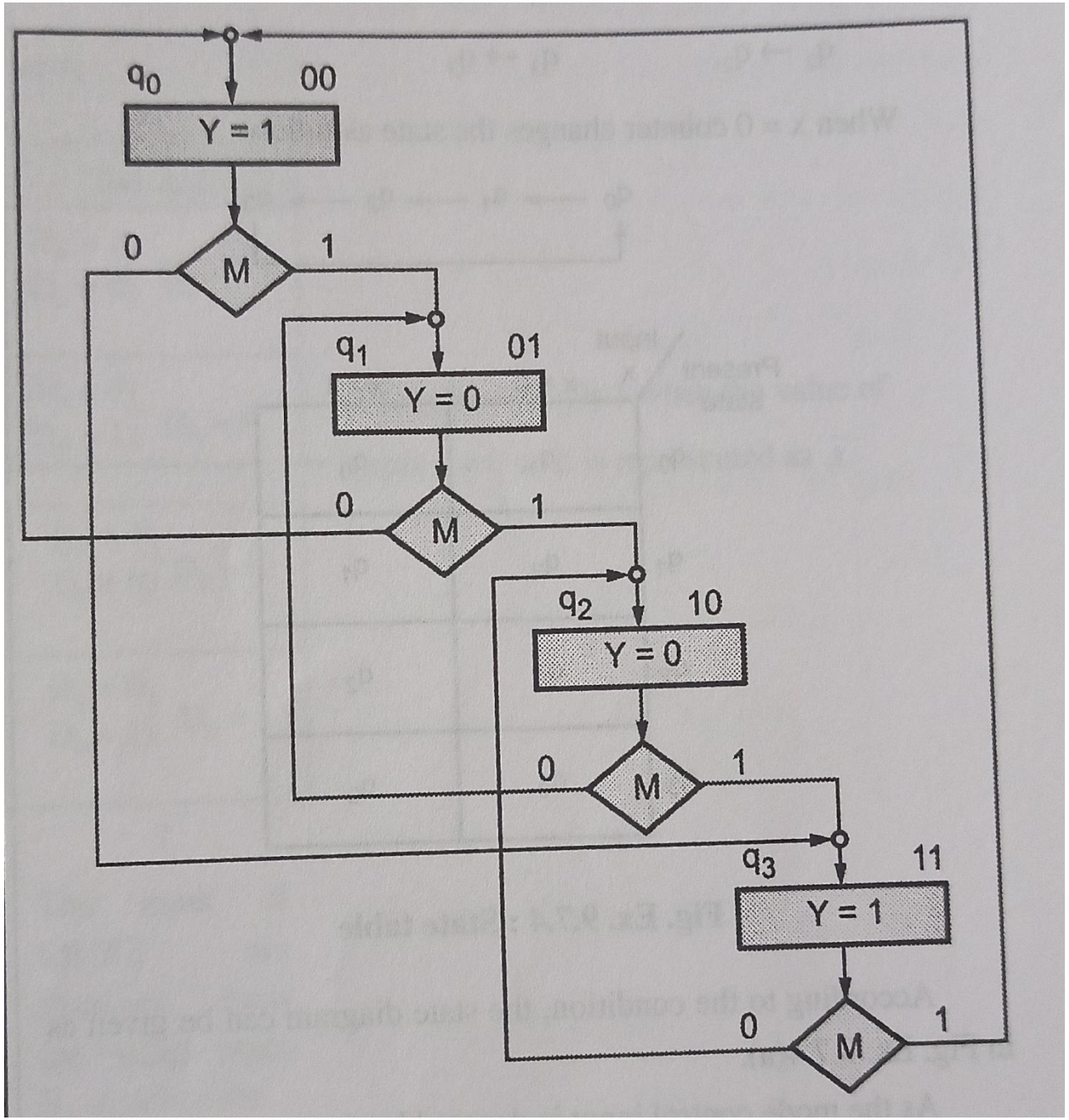
State Diagram:



State Table:



ASM Chart:



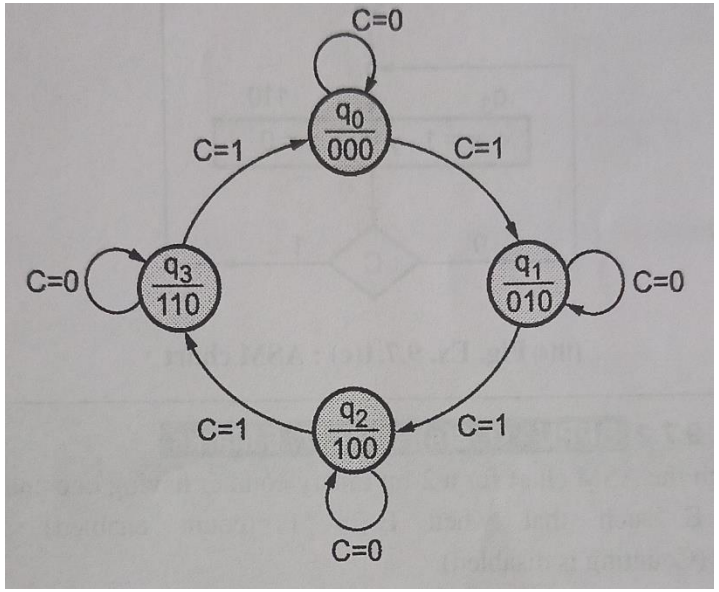
Algorithmic State Machine

Example: Draw an ASM chart and state diagram for the synchronous circuit having the following description,
The circuit has control input C, clock and output x,y,z.

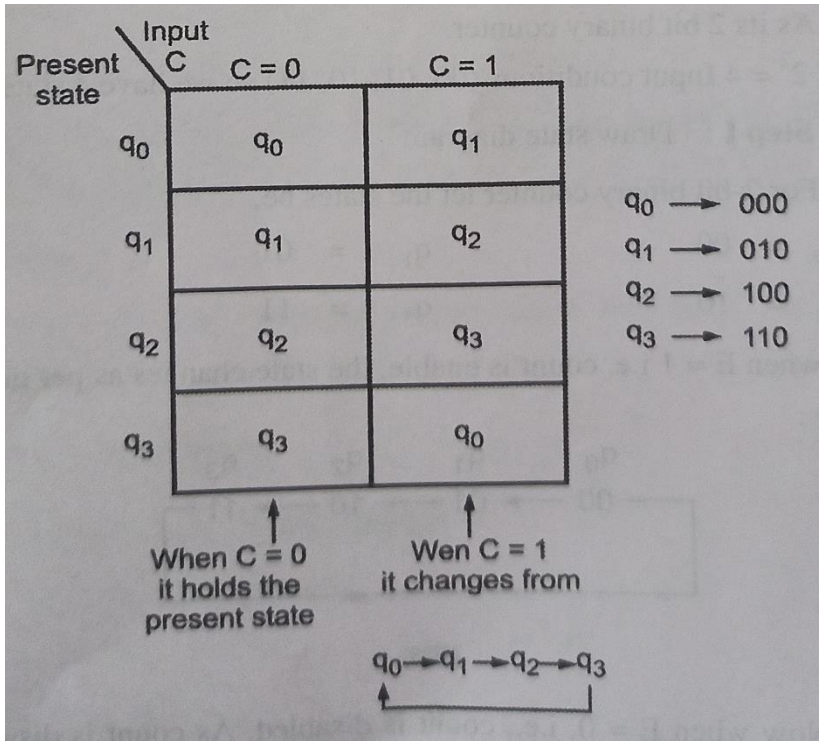
1. **If C=1, on every rising edge of the clock and output x,y and z changes from 000-> 010 -> 100 -> 110 -> 000 and repeats**
2. **If C=0, then the circuit holds the present state.**

Solution: Let consider,
q0 = 000
q1 = 010
q2 = 100
q3 = 110

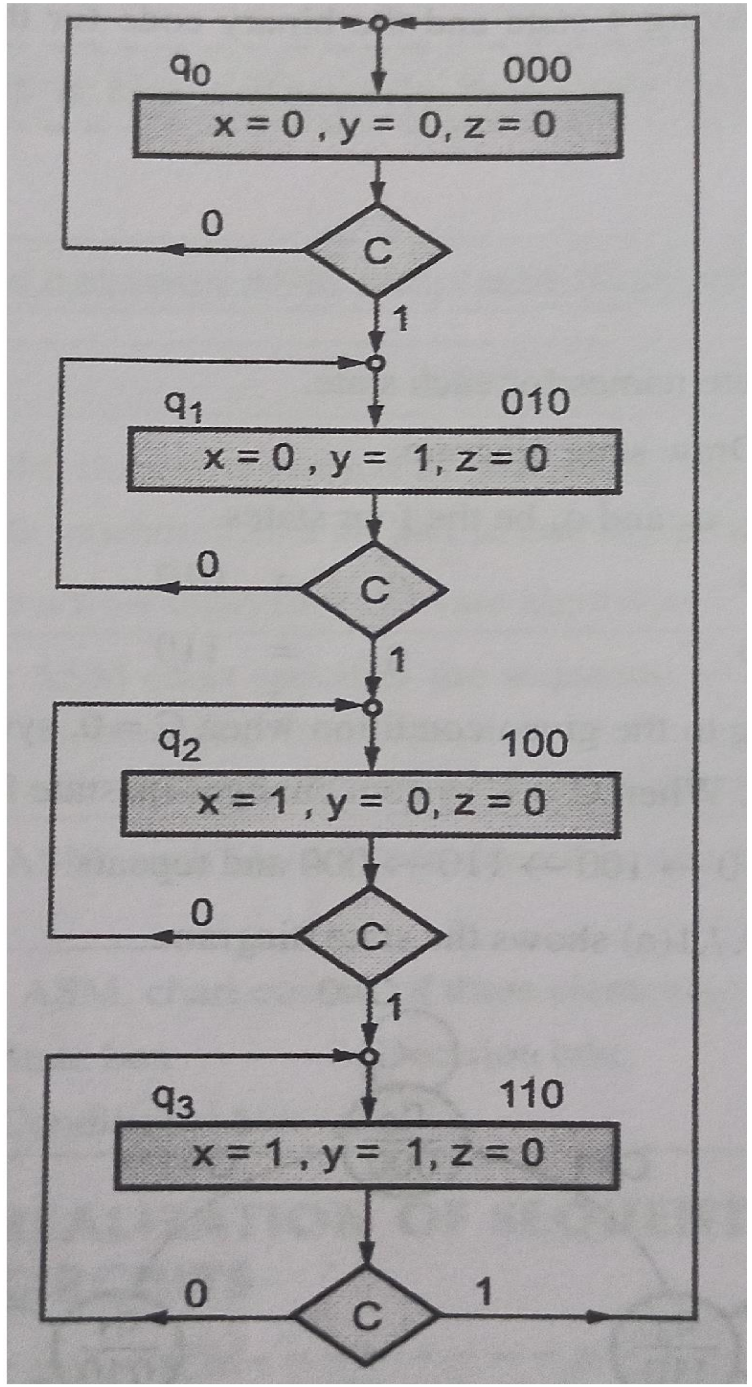
State Diagram



State Table



ASM Chart:


q_0
000
x = 0 , y = 0, z = 0
0
C
1
q_1
010
x = 0 , y = 1, z = 0
0
C
1
q_2
100
x = 1 , y = 0, z = 0
0
C
1
q_3
110
x = 1 , y = 1, z = 0
0
C
1

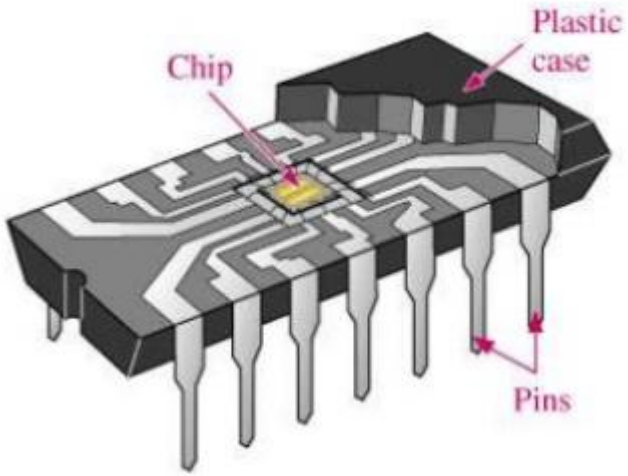
Agenda

Algorithmic State Machine: Finite state machine (FSM) and ASM, ASM charts, notations, Construction of ASM chart and realization of sequential circuits.

PLDs: PLD, ROM as PLD, Programmable Logic Array (PLA), Programmable Array Logic (PAL), Designing Combinational Circuits using PLDs.

Programmable Logic Devices

1. A logic device is an electronic component which performs a definite function which is decided at the time of manufacture and will never change. For example, a not gate always inverts the logic level of the input signal and does/can-do-nothing else.



2. On the other hand, **Programmable Logic Devices** (PLDs) are the components which do not have a specific function associated with them.
3. These can be configured to perform a certain function by the user, on a need basis and can further be changed to perform some other function at the later point of time, i.e. these are re-configurable. However, the amount of flexibility offered depends on their type.

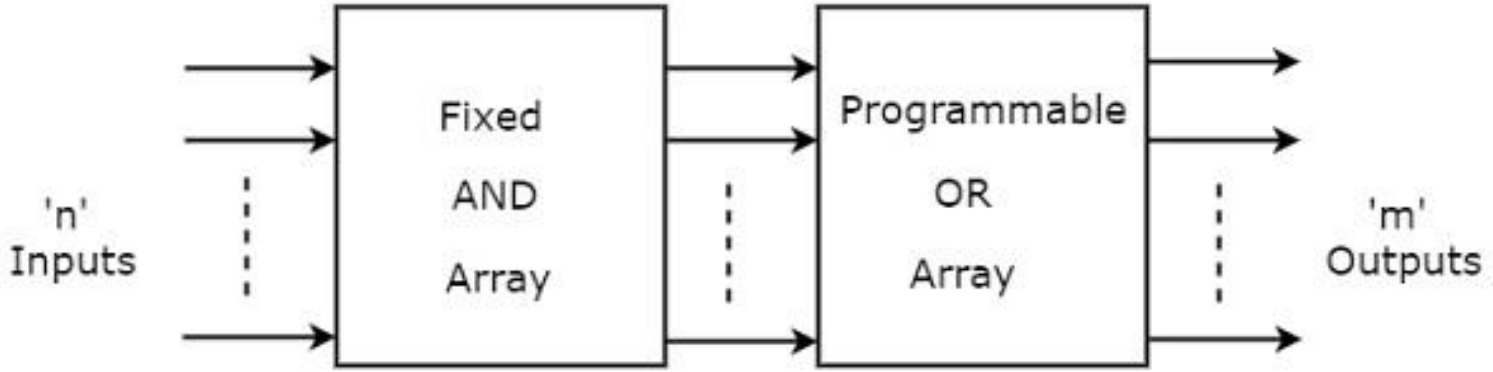
Programmable Logic Devices

Types of PLDs

1. **Simple Programmable Logic Device (SPGA)**
   - Programmable Read Only Memory (PROM)
   - Programmable Logic Array (PLA)
   - Programmable Array Logic (PAL)
2. **Complex Programmable Logic Device (CPGA)**
3. **Field Programmable Gate Array (FPGA)**

Programmable Read Only Memory PROM

Read Only Memory ROMROM is a memory device, which stores the binary information permanently. That means, we can't change that stored information by any means later. If the ROM has programmable feature, then it is called as **Programmable ROM** PROMPROM. The user has the flexibility to program the binary information electrically once by using PROM programmer.
PROM is a programmable logic device that has fixed AND array & Programmable OR array. The **block diagram** of PROM is shown in the following figure.



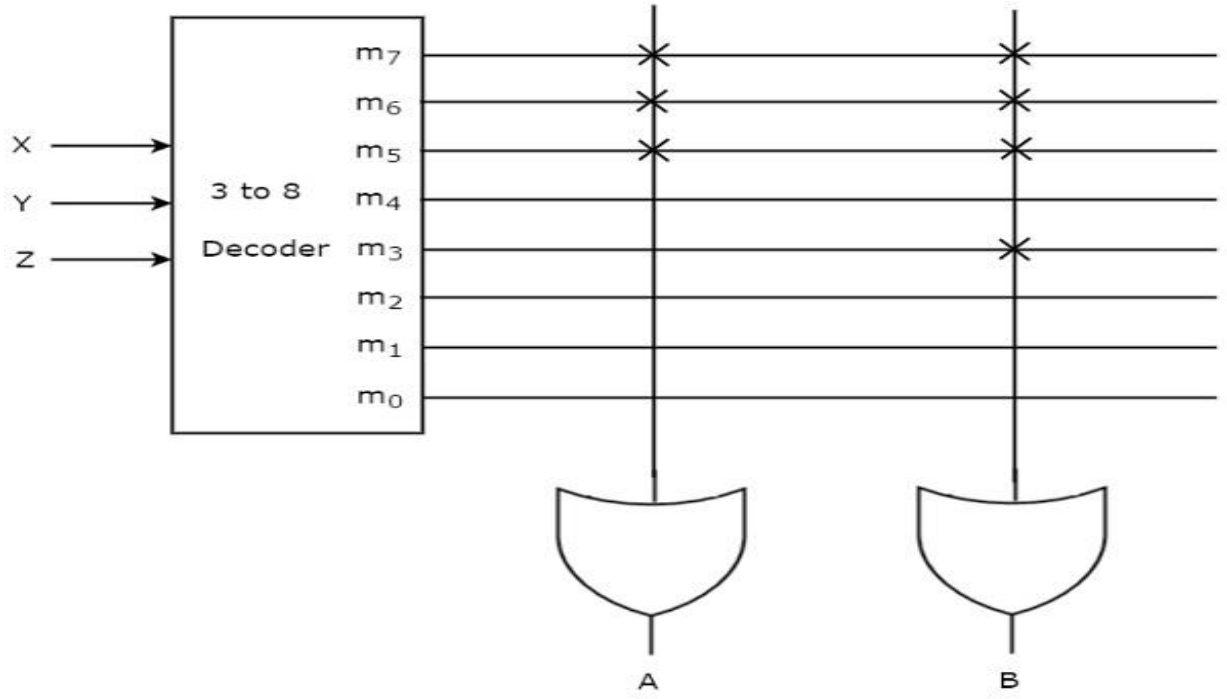
Programmable Read Only Memory PROM

Example:

Let us implement the following **Boolean functions** using PROM.

A(X,Y,Z)=∑m(5,6,7)
B(X,Y,Z)=∑m(3,5,6,7)

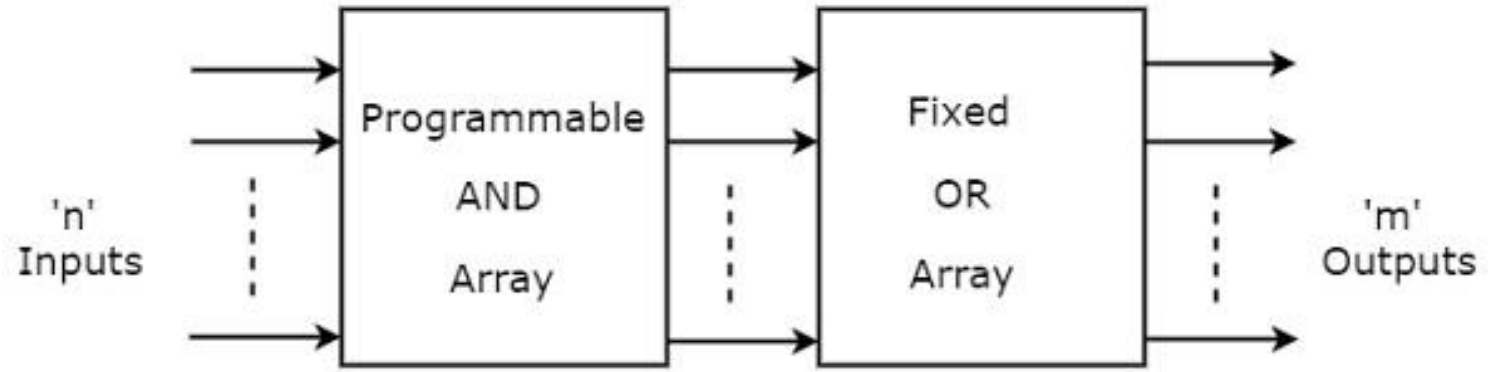
The given two functions are in sum of min terms form and each function is having three variables X, Y & Z. So, we require a 3 to 8 decoder and two programmable OR gates for producing these two functions. The corresponding **PROM** is shown in the following figure.



Here, 3 to 8 decoder generates eight min terms. The two programmable OR gates have the access of all these min terms. But, only the required min terms are programmed in order to produce the respective Boolean functions by each OR gate. The symbol ‘X’ is used for programmable connections.

Programmable Array Logic (PAL)

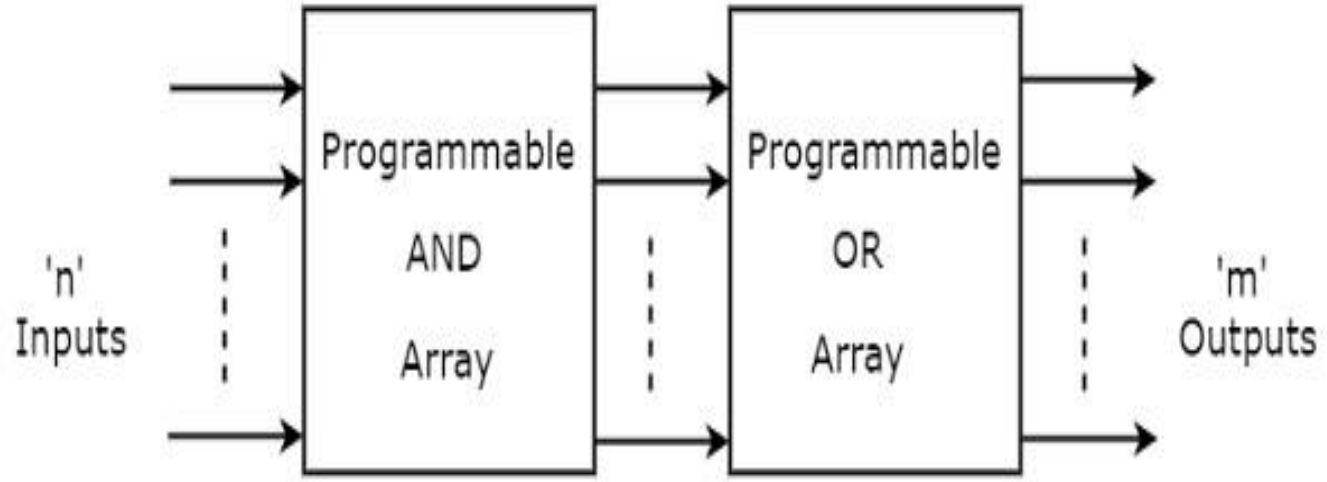
PAL is a programmable logic device that has Programmable AND array & fixed OR array. The advantage of PAL is that we can generate only the required product terms of Boolean function instead of generating all the min terms by using programmable AND gates. The **block diagram** of PAL is shown in the following figure.



Here, the inputs of AND gates are programmable. That means each AND gate has both normal and complemented inputs of variables. So, based on the requirement, we can program any of those inputs. So, we can generate only the required **product terms** by using these AND gates.
Here, the inputs of OR gates are not of programmable type. So, the number of inputs to each OR gate will be of fixed type. Hence, apply those required product terms to each OR gate as inputs. Therefore, the outputs of PAL will be in the form of **sum of products form**.

Programmable Logic Array (PLA)

PLA is a programmable logic device that has both Programmable AND array & Programmable OR array. Hence, it is the most flexible PLD. The **block diagram** of PLA is shown in the following figure.

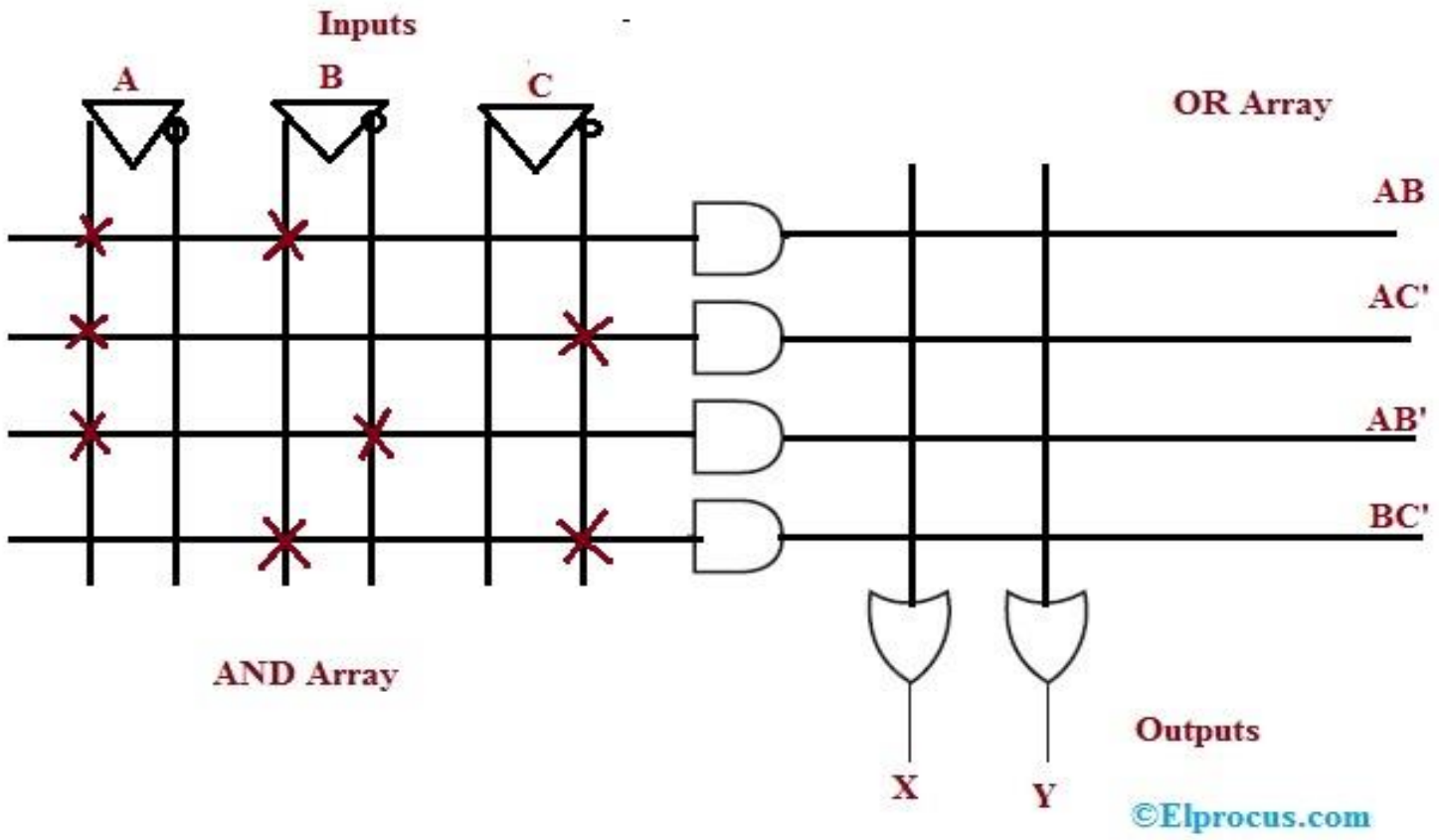


Here, the inputs of AND gates are programmable. That means each AND gate has both normal and complemented inputs of variables. So, based on the requirement, we can program any of those inputs. So, we can generate only the required **product terms** by using these AND gates.

Here, the inputs of OR gates are also programmable. So, we can program any number of required product terms, since all the outputs of AND gates are applied as inputs to each OR gate. Therefore, the outputs of PAL will be in the form of **sum of products form**.

Example of PAL

1. Implement the following **Boolean expression** with the help of **programmable array logic (PAL).**
   X =AB + AC’
   Y= AB’ + BC’

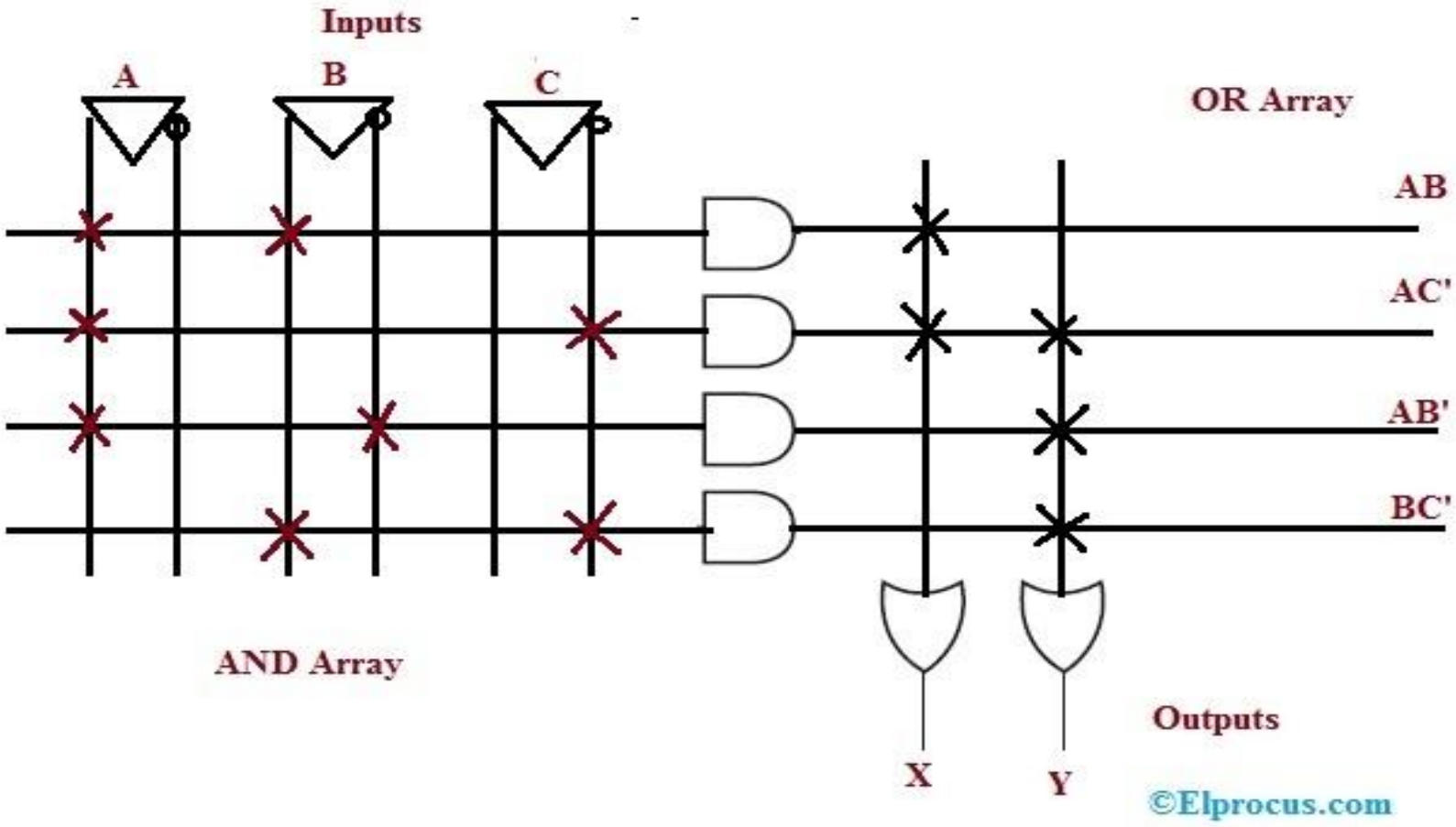


Example of PLA

1. Implement the following Boolean expression with the help of programmable logic array (PLA)

X = AB + AC'

Y = AB' +BC + AC'



Difference Between PLA and PAL

Programmable Array Logic (PAL)	Programmable Logic Array (PLA)
The full form of PAL is programmable array logic	The full form of the PLA is a programmable logic array
The construction of PAL can be done using the programmable collection of AND & OR gates	The construction of PLA can be done using the programmable collection of AND & fixed collection of OR gates.
The availability of PAL is less prolific	The availability of PLA is more
The flexibility of PAL programming is more	The flexibility of PLA is less
The cost of a PAL is expensive	The cost of PLA is middle range
The number of functions implemented in PAL is large	The number of functions implemented in PLA is limited
The speed of PAL is slow	The speed of PLA is high