Unit 3:
Algorithmic State Machines

by

Prof. Sujata Wakchaure

Introduction

- It is a sequential network which controls a digital system that carries out a step-by-step procedure or algorithm

- Drawbacks of state diagrams for real systems:
  - Many inputs & many outputs -> awkward
  - Not a clear structure for illustrating/designing control flow

Introduction (Contd..)

- Some problems analogous to before
  - Combinational:
    - Small problems – truth tables ok/easy
    - Adders, Mux – TT get out of hand
  - Sequential:
    - Small – state diagrams easy
    - Real, Data – state diagrams not helpful

Finite State Machine (FSM)

- A generic model for sequential circuits used in sequential circuit design
- It is a mathematical model of computation
- It is an abstract machine that can be in exactly one of a finite number of states at any given time
- FSMs are an incredibly powerful tool when designing digital circuits
- FSM Consist of
  - States
  - State Transitions

FSM (Contd..)

- State :
  - It defines behavior & may produce action (Moore M/C)

- State Transition:
  - It is movement from one state to another & may produce actions (Mealy M/C)

- A FSM Must have:
  - An initial state which provides a starting point
  - A current state, which remembers the product of the last state transition

Algorithmic State Machine (ASM)

- ASM method is a method for designing FSM
- It is used to represent diagrams of digital integrated ckt
- ASM is nothing but step by step procedure or algorithm
- It is an advanced version of state diagram
- An IMP advantage of ASM is, it an describe both Combinational and Sequential ckt

ASM Chart

- ASM Chart is a high-level flowchart-like notation to specify the hardware algorithms in digital systems.

- It is a method of describing the sequential operations of a digital system

- ASM Chart consist of an interconnections of 4 types of basic element
  1. State name
  2. State Box
  3. Decision Box
  4. Conditional Output Box

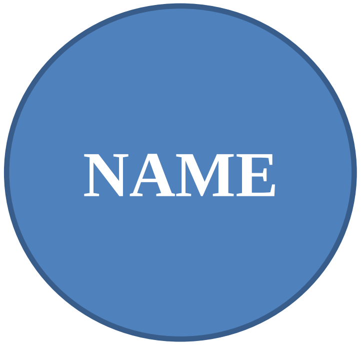
ASM Chart (Contd..)

- ASM charts are like flowcharts, with a few crucial differences. Be careful, especially with timing.
  - State Box
  - Decision Box
  - Combinational Box

- Major differences from flowcharts are:
  - uses 3 types of boxes: state box (similar to operation box), decision box and conditional box
  - contains exact (or precise) timing information; flowcharts impose a relative timing order for the operations

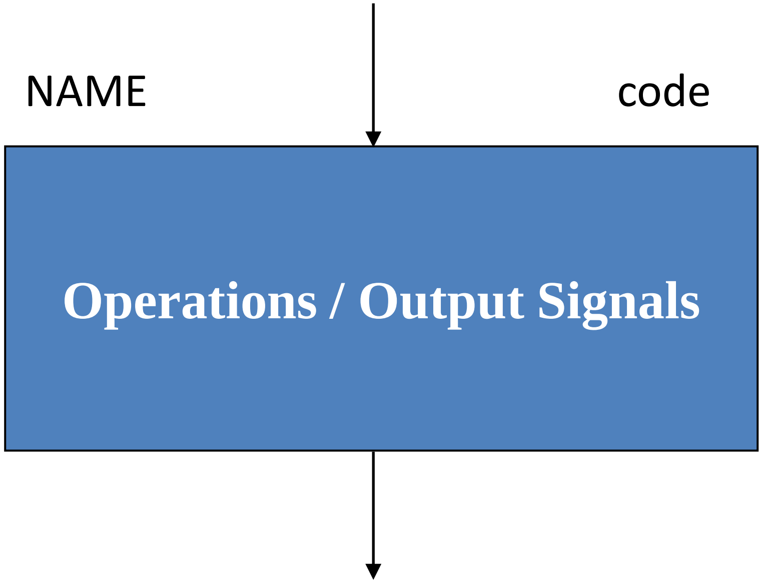
ASM Chart (Contd..)

- **State Name:**
  - The name of the state is indicated inside the circle & circle is placed in the top left corner or the name is placed without the circle



ASM Chart (Contd..)

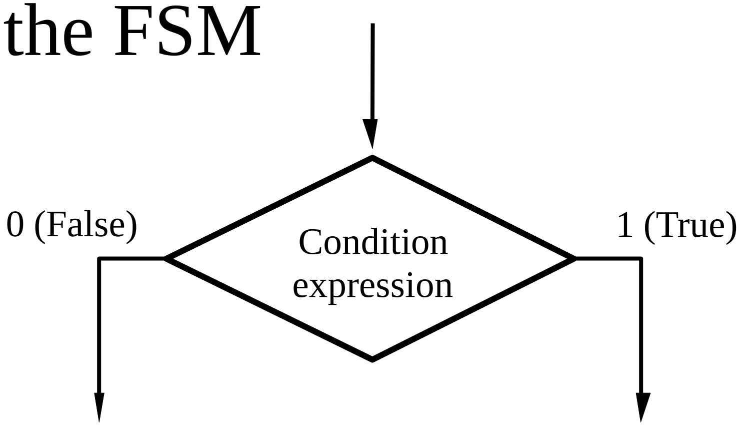
- **State Box:**
  - one box per system state
  - represents a state equivalent to a node in a state diagram or a row in a state table.
  - **Moore-type outputs are listed inside of the box.**



i.e. optional binary state code

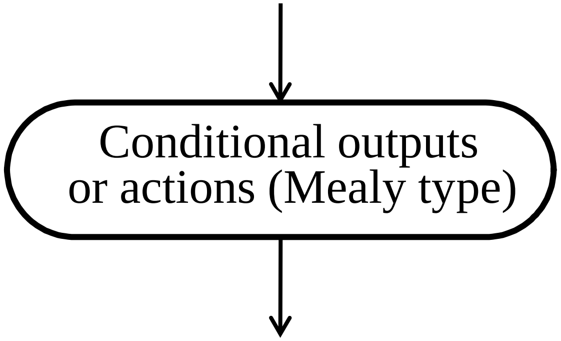
ASM Chart (Contd..)

- **Decision box:**
  - Indicates that a given condition is to be tested and the exit path is to be chosen accordingly the condition expression may include one or more inputs to the FSM



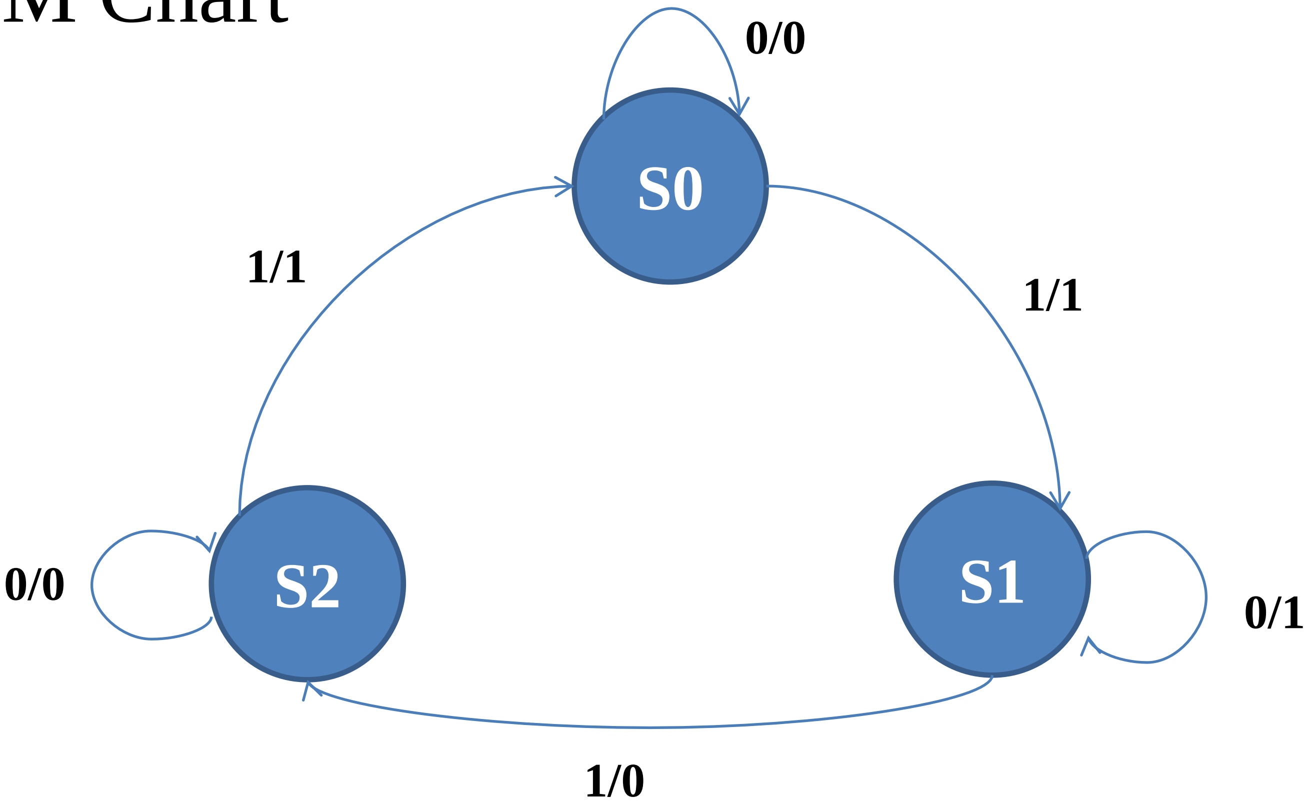
ASM Chart (Contd..)

- **Conditional O/P Box: (Mealy Box)**
  - Denotes output signals that are of the Mealy type
  - The condition that determines whether such outputs are generated is specified in the decision box.

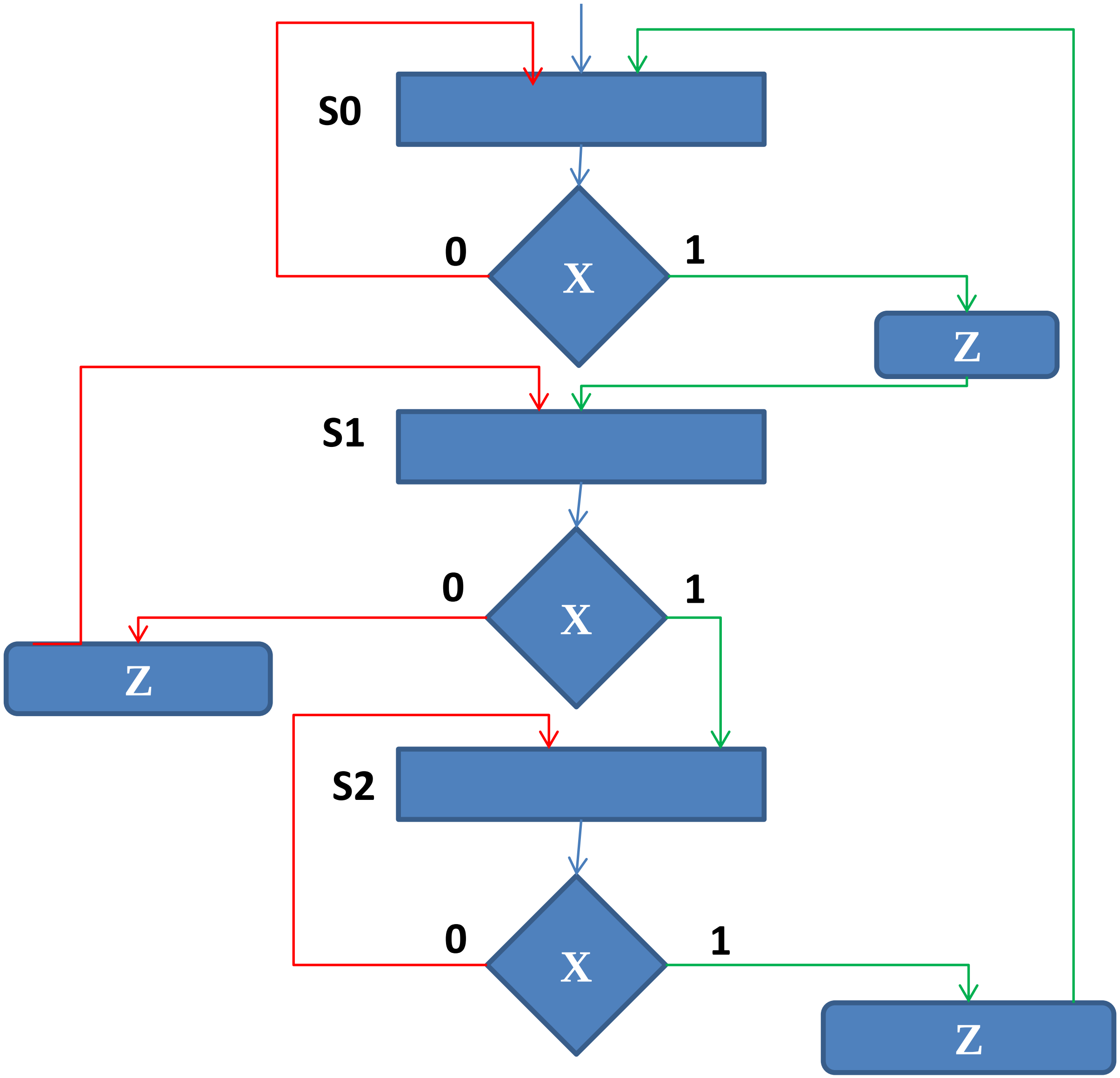


Example (Mealy State Machine)

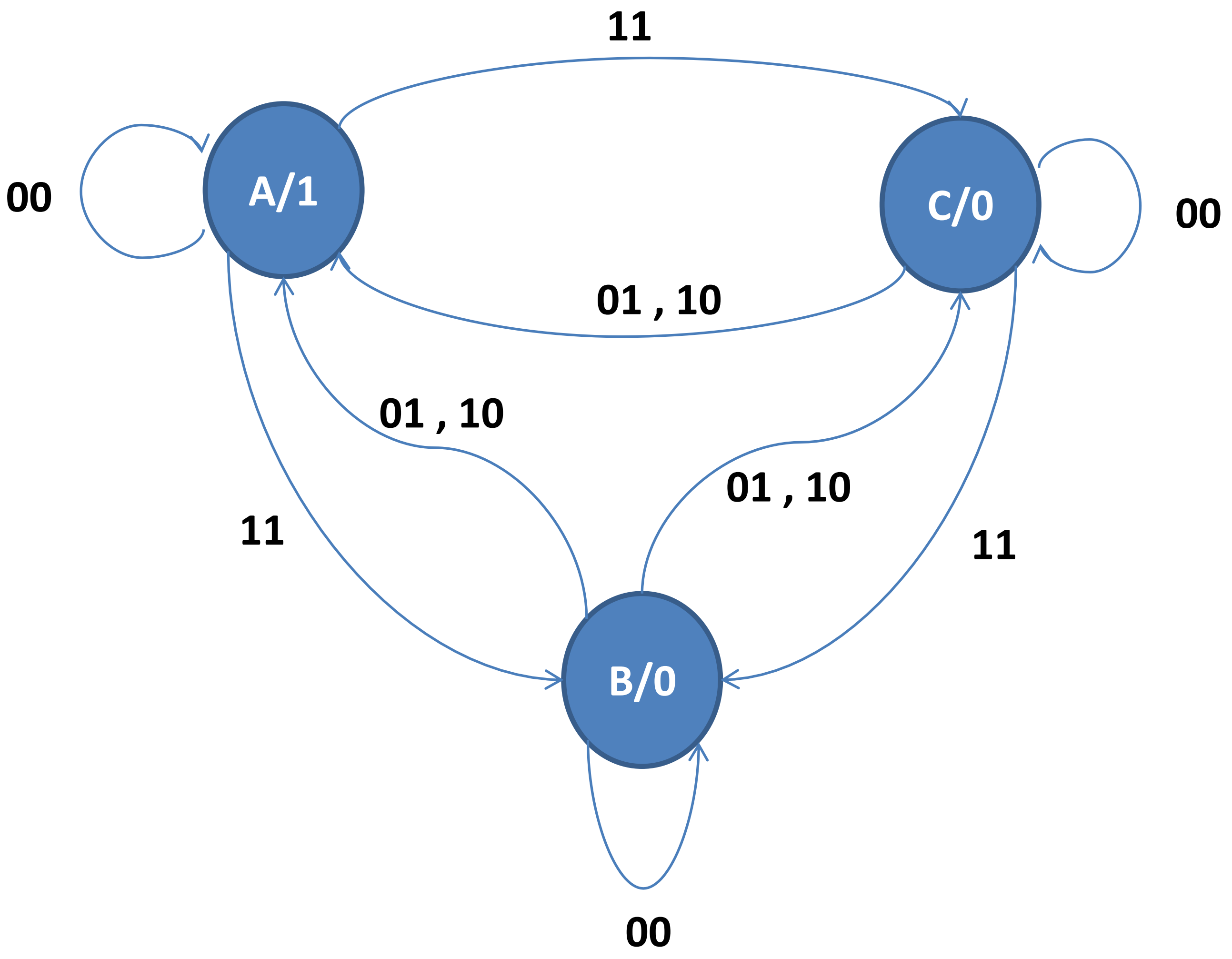
- Consider following state diagram & design ASM Chart



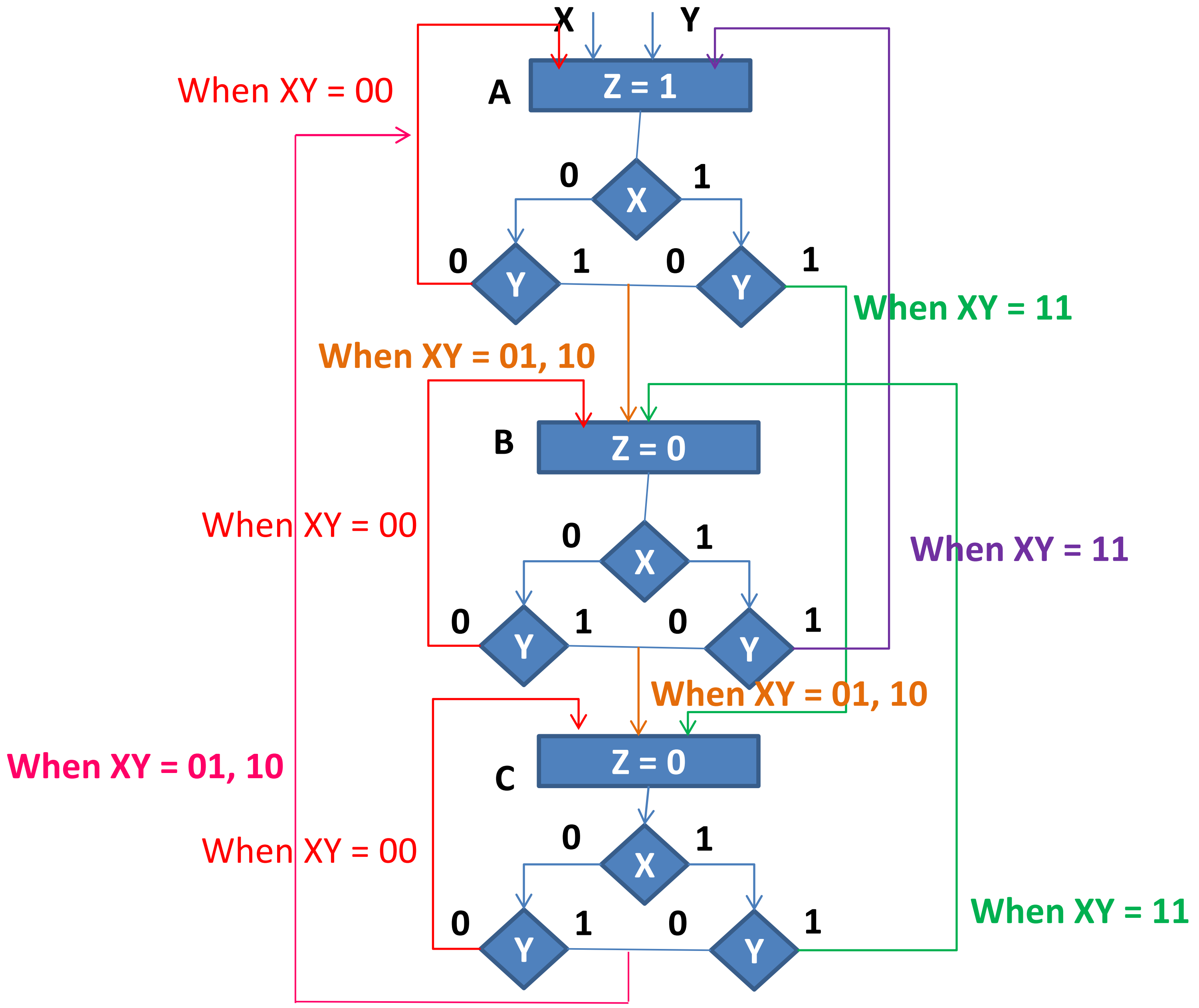
Example (Contd..)



Example (Moore State Machine)



Example (Contd..)



Multiplexer Controller Method

- It is a simpler & straight forward method for realization of combinational ckt for any controller
- In this method, the gates and flip-flops are replaced by mux & registers respectively
- In this method, there are 3 levels components
- The first level consist of mux that determine next state of the register
- The second level contain a register that holds the present binary state
- The third level has the decoder that provides a separate O/P for each control state
- Sometimes combinational ckt is used in place of decoder

Multiplexer Controller Method (Contd..)

- Multiplexer decides the next state of the register as O/P of mux has been connected to FF I/P

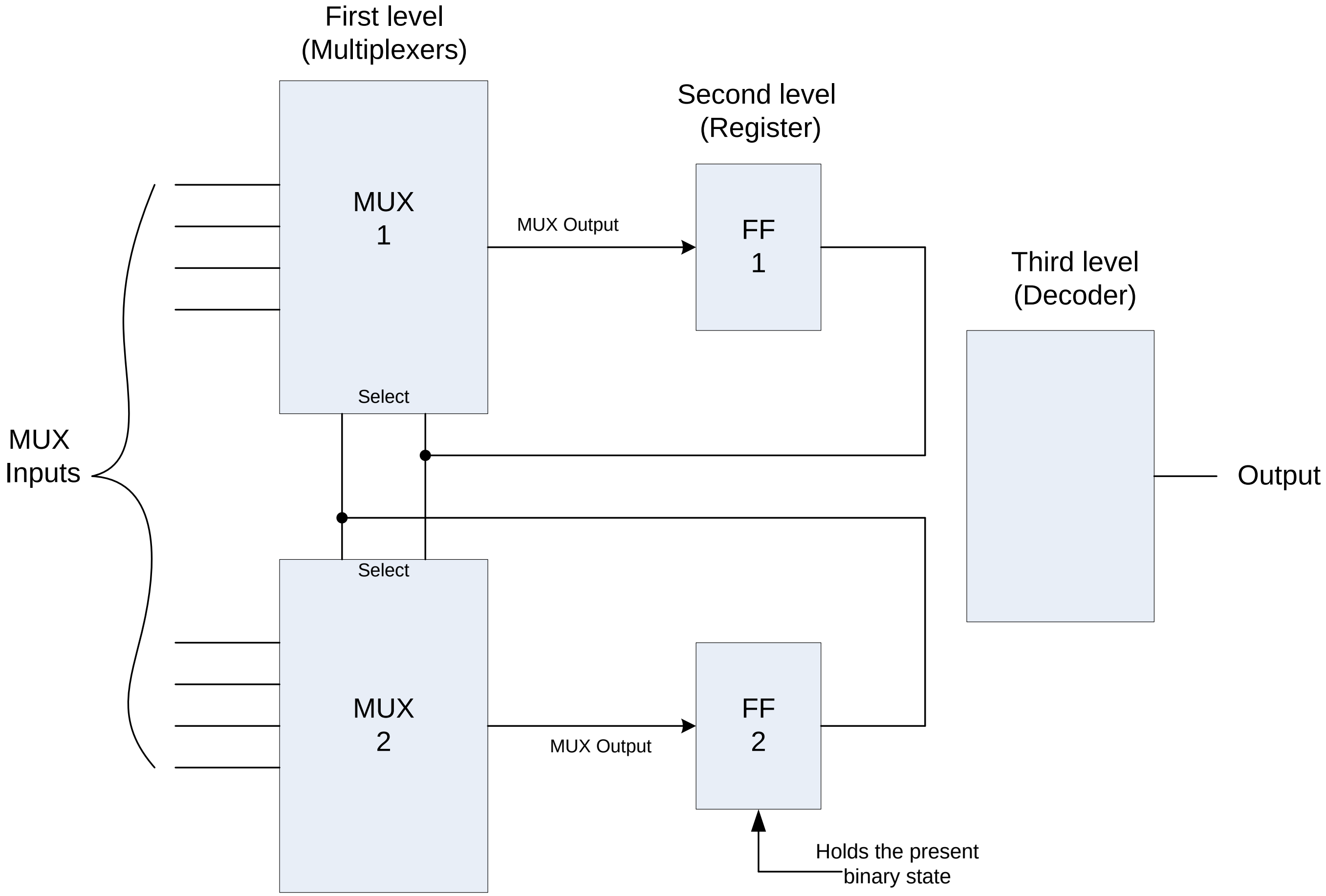


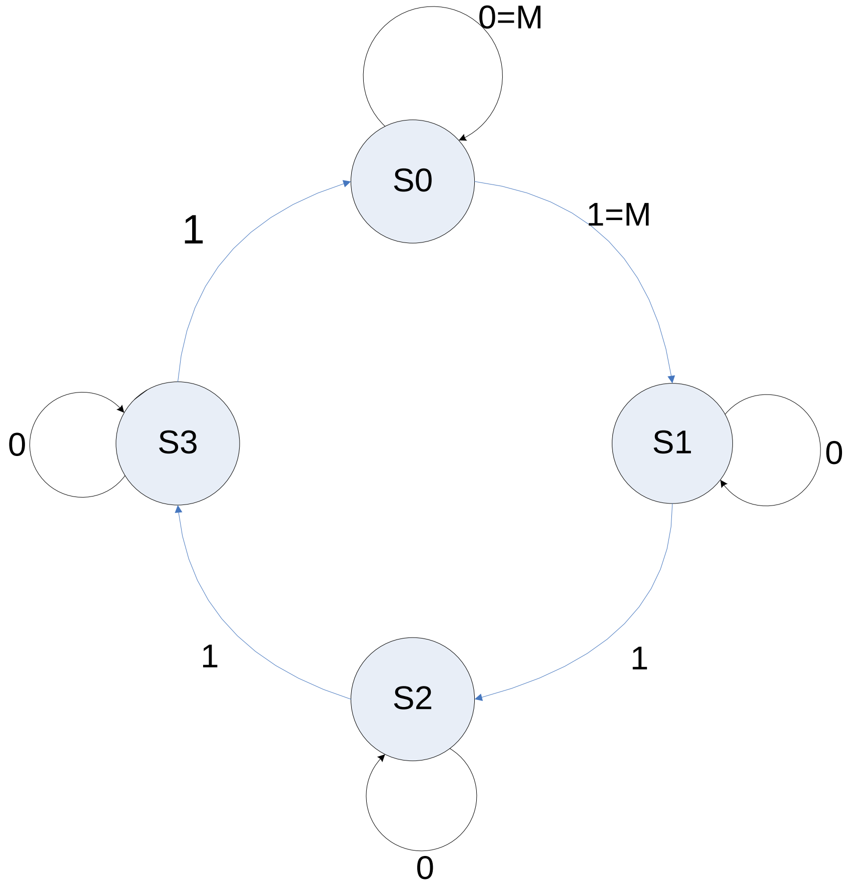
Fig. Block schematic for a 3-level scheme for multiplexer design

Example

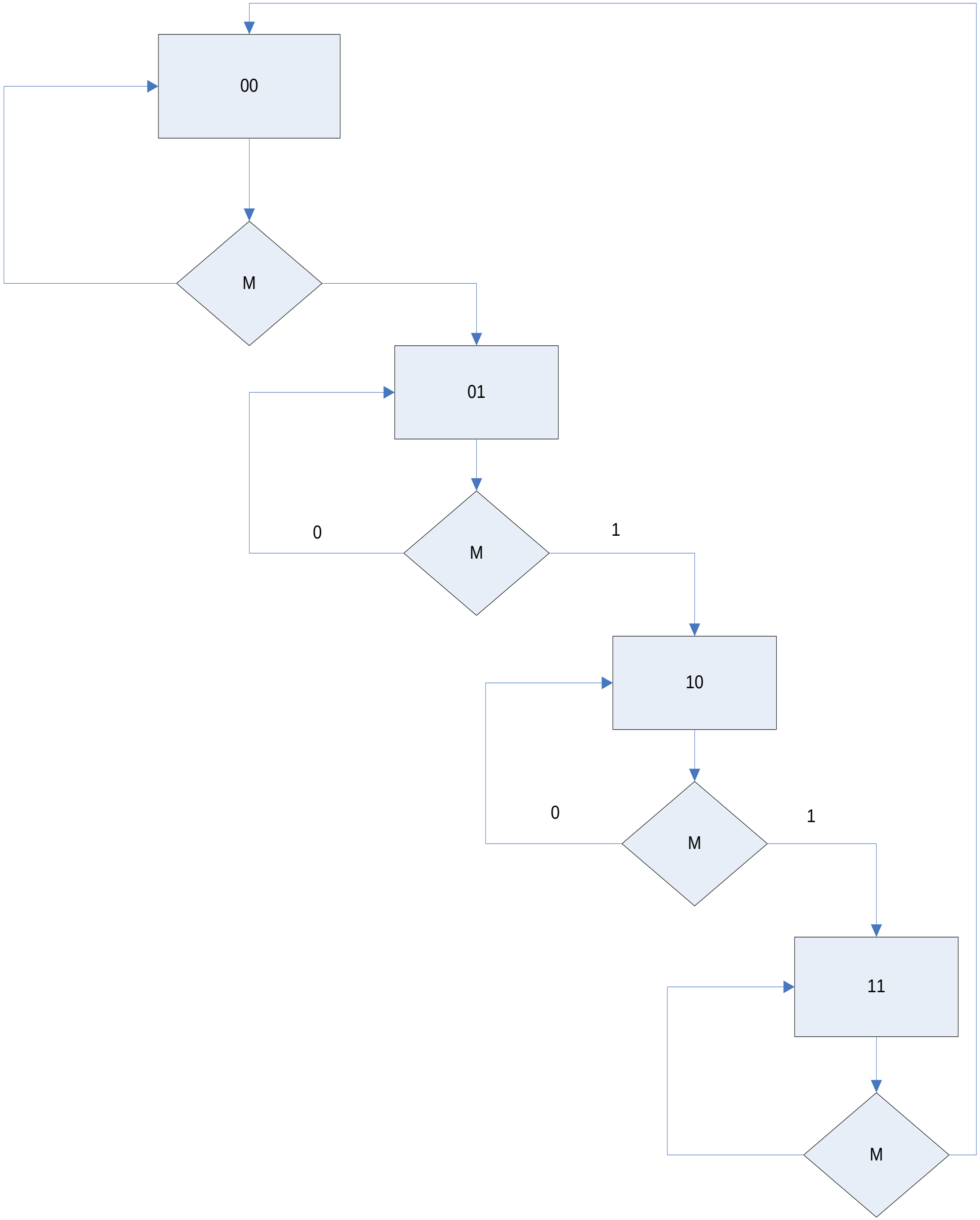
- Design of 2 bit Up counter using multiplexer controller method
  - 2 bit counter has 4 states i.e. 00,01,10,11.
  - In the state diagram if mode control M =0, counter will be latched in the same state and will start incrementing to the next state if M=1.

Example (Contd..)

- Step 1: State Diagram



- **Step 2: ASM Chart**



Example (Contd..)

- Step 3: State Transition Table

Mode control i/p	Present state (Qn)		Next state (Q n+1)	
M	**Qb**	**Qa**	**Qb+1**	**Q a+1**
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0

Example (Contd..)

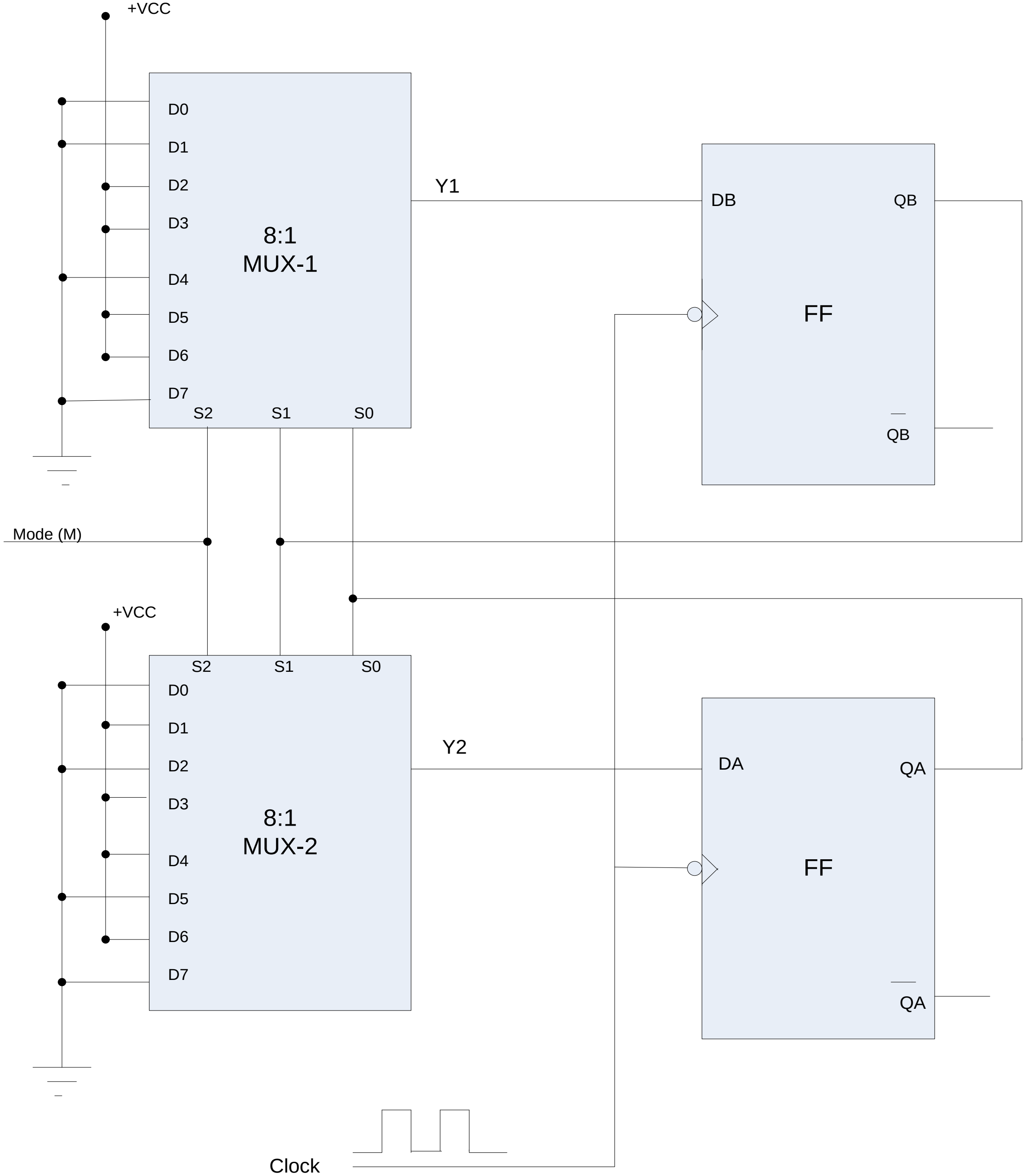
- Step 4: Excitation Table of D FF

Present State	**Next state**	**Input**
Qn	Qn+1	Dn
0	0	0
0	1	1
1	0	0
1	1	1

Example (Contd..)

- Step 5: State Table

Mode control i/p	Present state (Qn)		Next state (Qn+1)		Input	
M	Qb	Qa	Qb+1	Qa+1	Db	Da
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	1	0	1	0
0	1	1	1	1	1	1
1	0	0	0	1	0	1
1	0	1	1	0	1	0
1	1	0	1	1	1	1
1	1	1	0	0	0	0

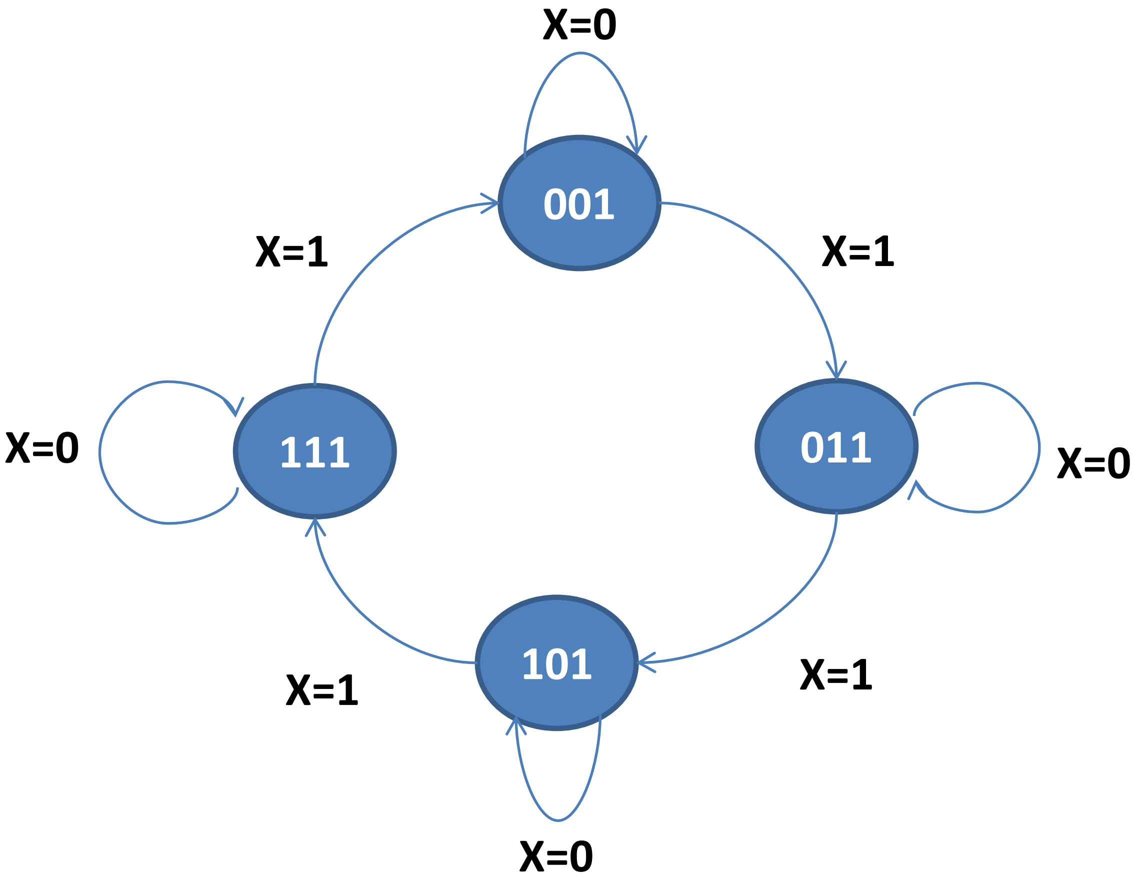

+VCC
D0
D1
D2
D3
8:1
MUX-1
D4
D5
D6
D7
S2
S1
S0
Y1
DB
QB
FF
QB
Mode (M)
+VCC
S2
S1
S0
D0
D1
D2
D3
8:1
MUX-2
D4
D5
D6
D7
Y2
DA
QA
FF
QA
Clock

Example

- Design a Sequence Generator Circuit to generate the sequence 1-3-5-7 using MUX Controller based ASM approach
- Considerations
  - If control I/P X=0, the sequence generator ckt in the same state
  - If control I/P X=1, the sequence generator ckt goes into next state

Example (Contd..)

- Step 1: State Diagram

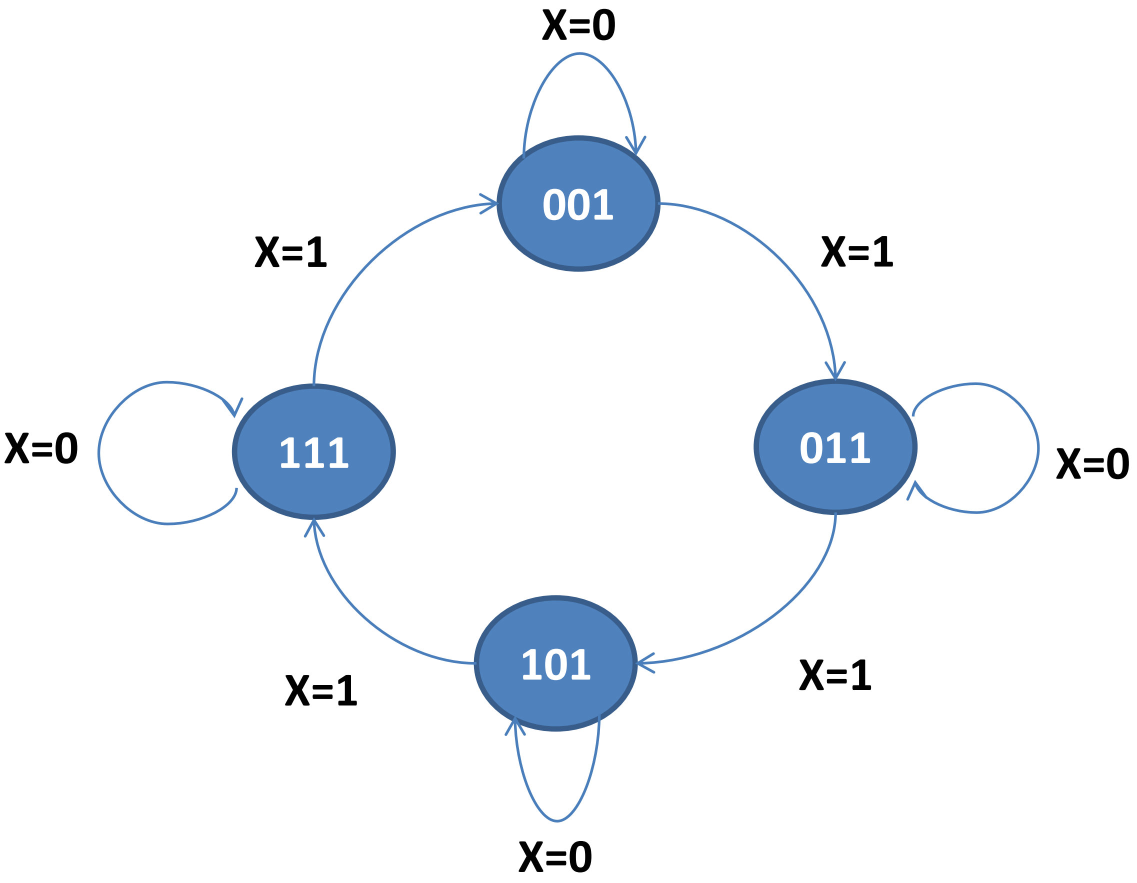


q0 = 001
q1 = 011
q2 = 101
q3 = 111

Example (Contd..)

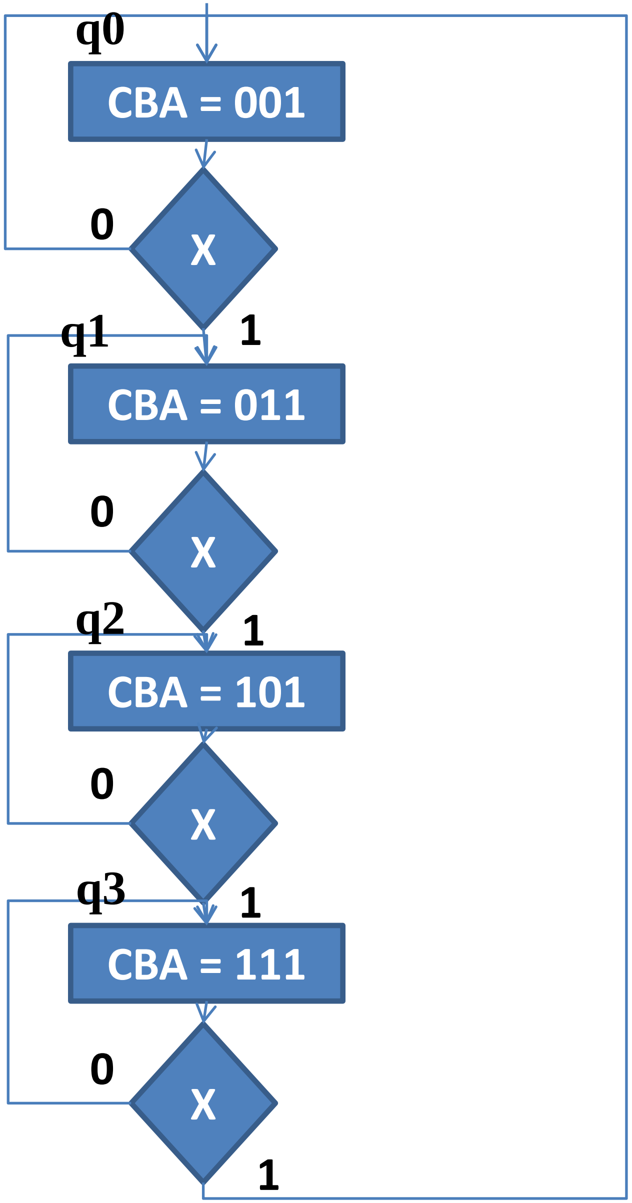
- Step 2: State Table

Present State	Next State	
	X=0	X=1
q0	001	011
q1	011	101
q2	101	111
q3	111	001

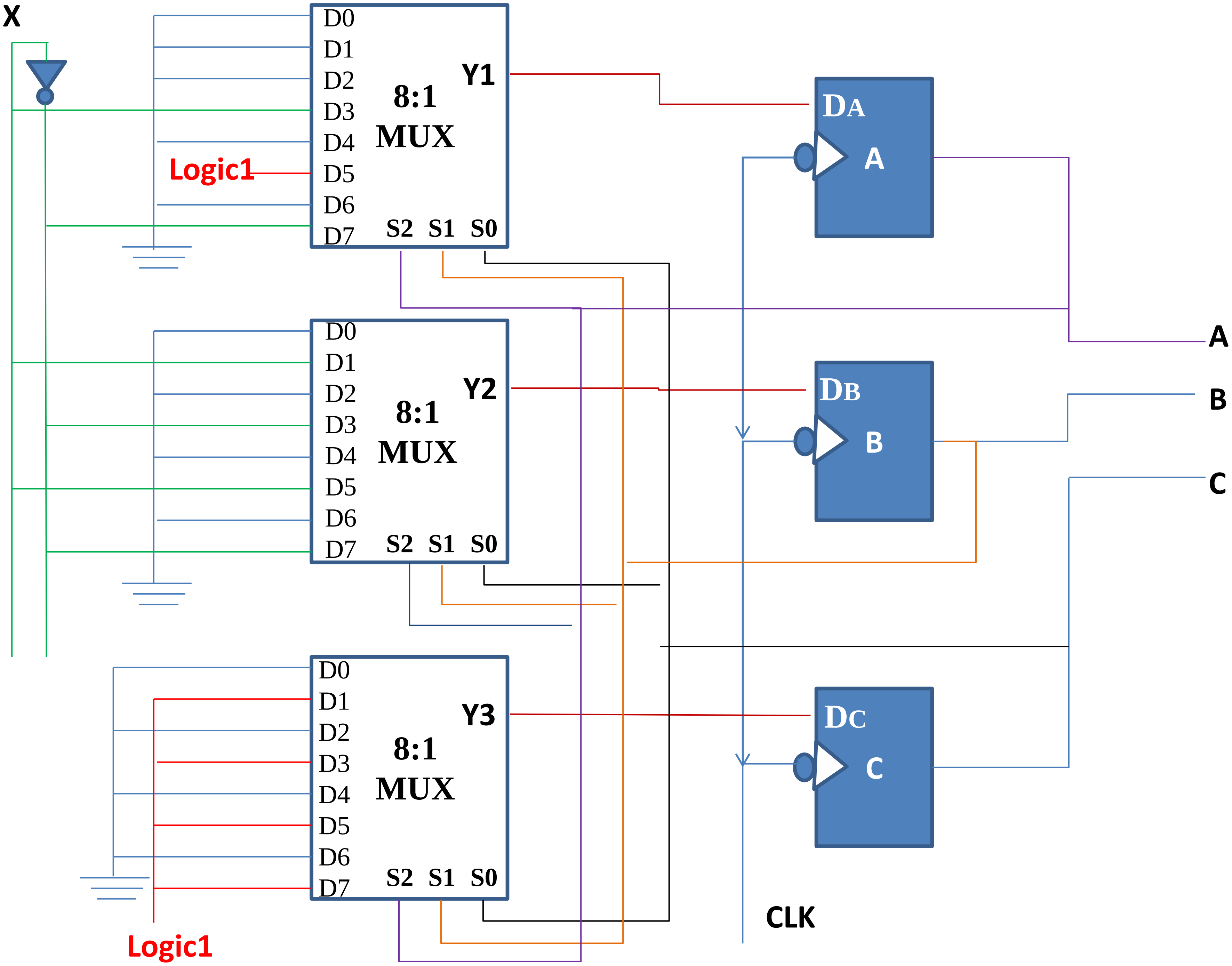


Example (Contd..)

- Step 3: ASM Chart



Present State			Next State			I/P	MUX Input					
C	B	A	C+	B+	A+	X	MUX 1		MUX 2		MUX 3	
0	0	0	0	0	0	0	D0=0	0	0	0	0	0
0	0	0	0	0	0	1	D0=0		0		0	
0	0	1	0	0	1	0	D1=0	0	0	X	1	1
0	0	1	0	1	1	1	D1=0		1		1	
0	1	0	0	0	0	0	D2=0	0	1	0	0	0
0	1	0	0	0	0	1	D2=0		0		0	
0	1	1	0	1	1	0	D3=0	X	1	$\overline{X}$	1	1
0	1	1	1	0	1	1	D3=1		0		1	
1	0	0	0	0	0	0	D4=1	0	0	0	0	0
1	0	0	0	0	0	1	D4=0		0		0	
1	0	1	1	0	1	0	D5=1	1	0	X	1	1
1	0	1	1	1	1	1	D5=1		1		1	
1	1	0	0	0	0	0	D6=1	0	1	0	0	0
1	1	0	0	0	0	1	D6=0		0		0	
1	1	1	1	1	1	0	D7=1	$\overline{X}$	1	$\overline{X}$	1	1
1	1	1	0	0	1	1	D7=0		0		1	


X
D0
D1
D2
D3
D4
D5
D6
D7
8:1
MUX
Y1
S2 S1 S0
Logic1
DA
A
A
D0
D1
D2
D3
D4
D5
D6
D7
8:1
MUX
Y2
S2 S1 S0
DB
B
B
C
D0
D1
D2
D3
D4
D5
D6
D7
8:1
MUX
Y3
S2 S1 S0
DC
C
Logic1
CLK