Synchronous Sequential Circuit Design

by

Prof. Sujata Wakchaure

Introduction

- Mealy and Moore models are the basic models of state machine

- **Moore Model:**
  - A state m/c which uses only entry actions, so that its O/P depends on the state is called as **Moore Model** (i.e. O/P depends only on Present state)

- **Mealy Model:**
  - A state m/c which uses only I/P actions, so that its O/P depends on the State and also on I/P, is called a**s Mealy Model.** (i.e. O/P depends on Present state and Present I/P)

Moore Circuit / Moore State Machine

- Sequential ckt consist of combinational ckt & memory block
- Memory block is nothing but FF
- In this case O/P depends on I/P
- But sometimes it doesn’t depend upon I/P, in that case we call it as **Moore ckt.**

Moore Circuit (Contd..)

- **How O/P is generated in Moore ckt?**
  - O/P is the function of Present state only

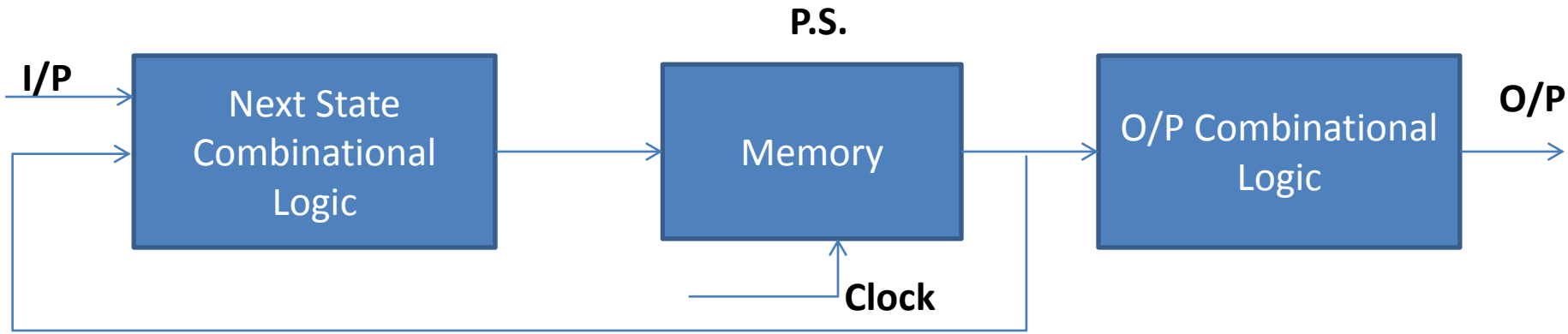
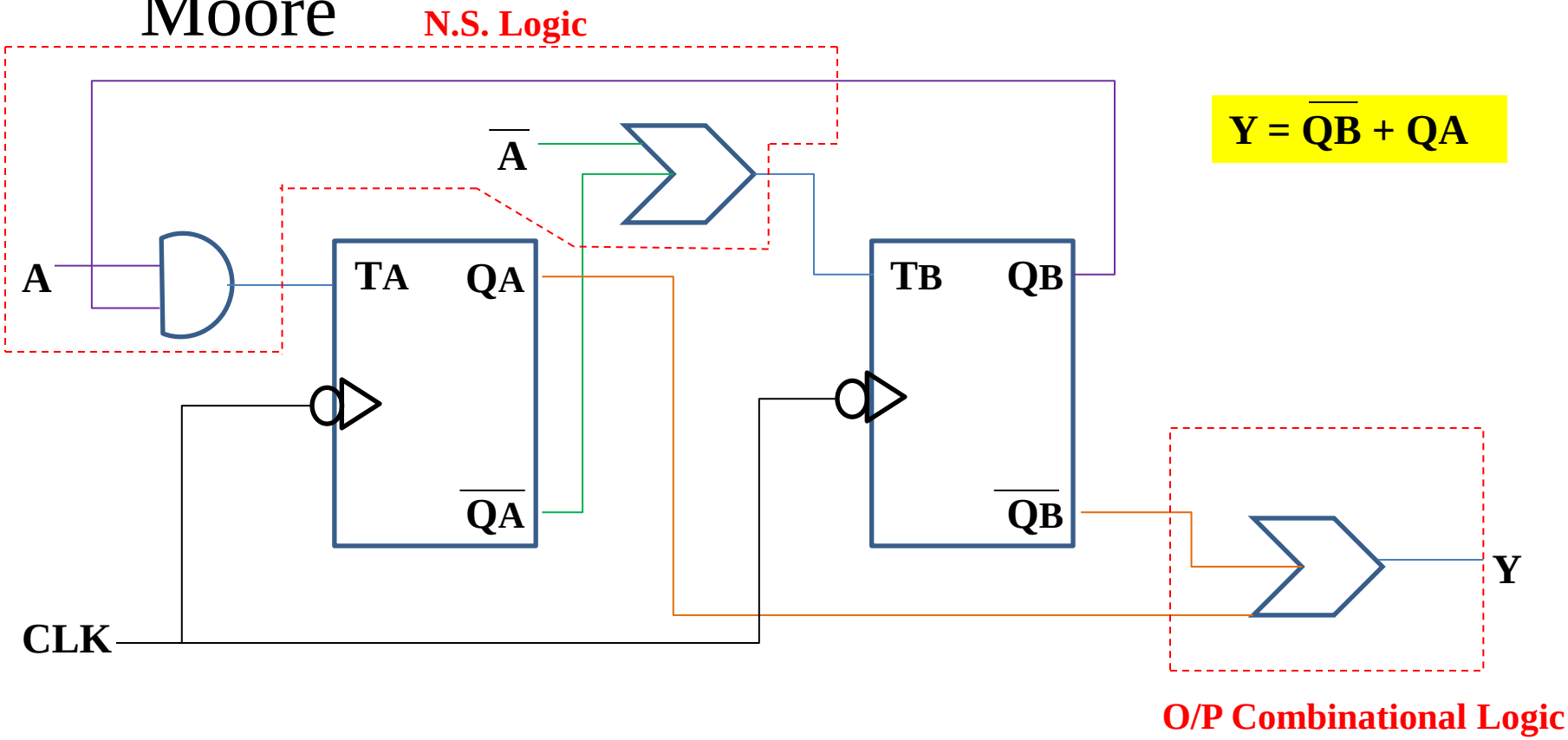


Fig: Next State Combinational Logic / Next State Decoder

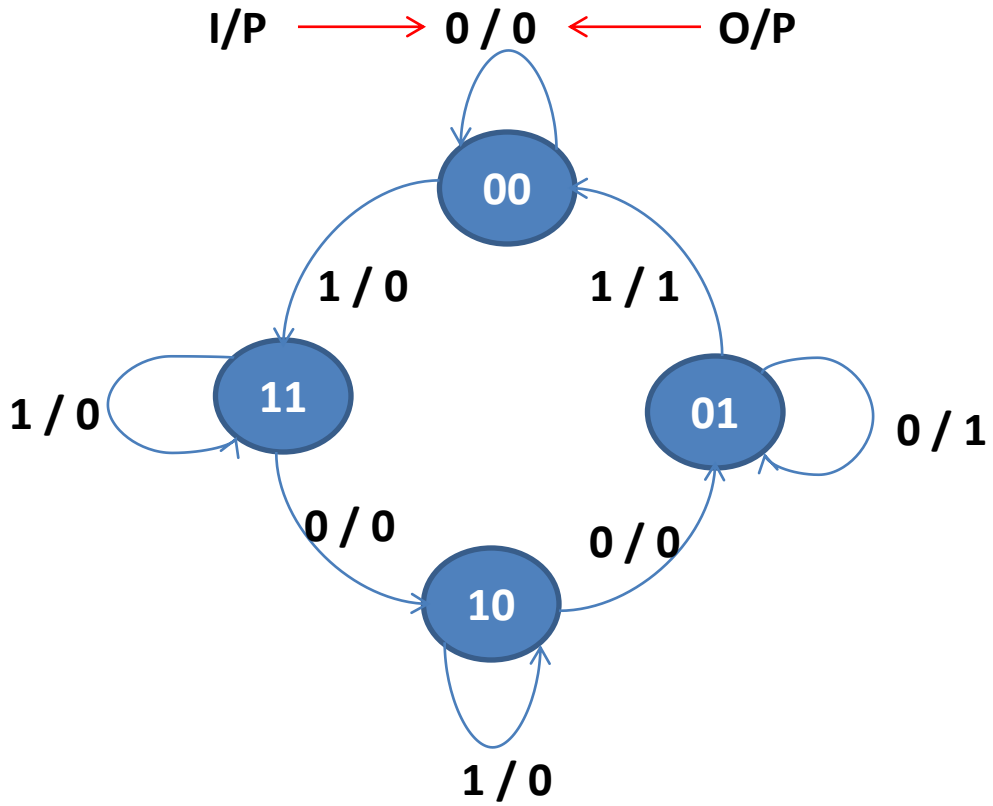
Moore Circuit (Contd..)

- Find whether the following ckt is Mealy or Moore



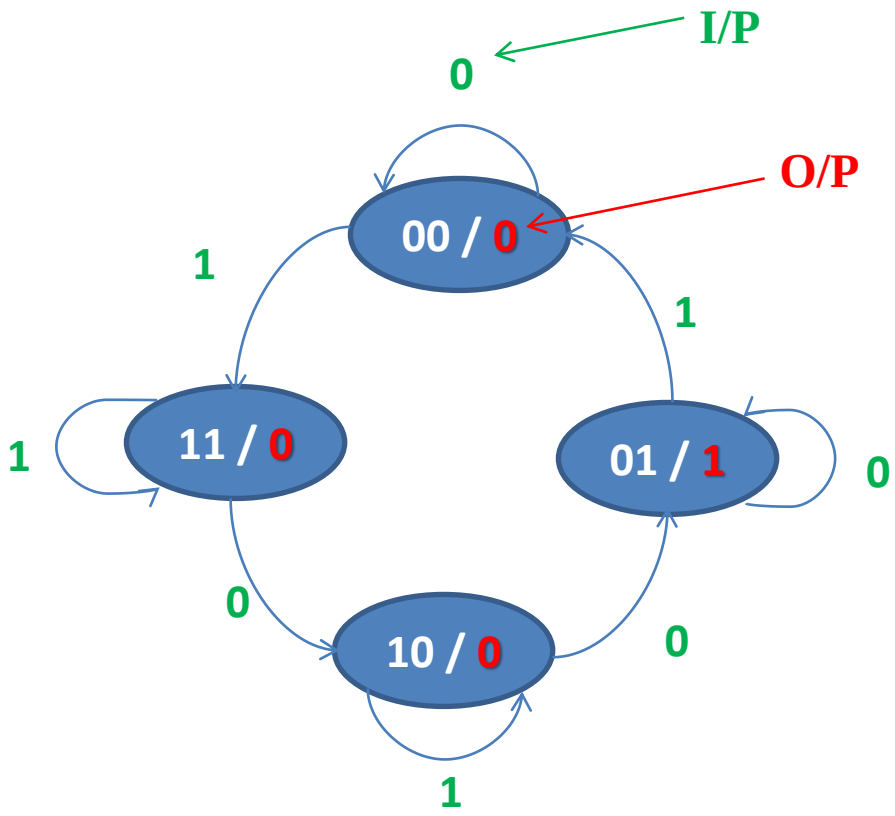
Moore Circuit (Contd..)

- **Normal State Diagram:**



Moore Circuit (Contd..)

- **Moore State Diagram:**



Mealy State Machine

- The O/P is function of the Present State as well as the I/P

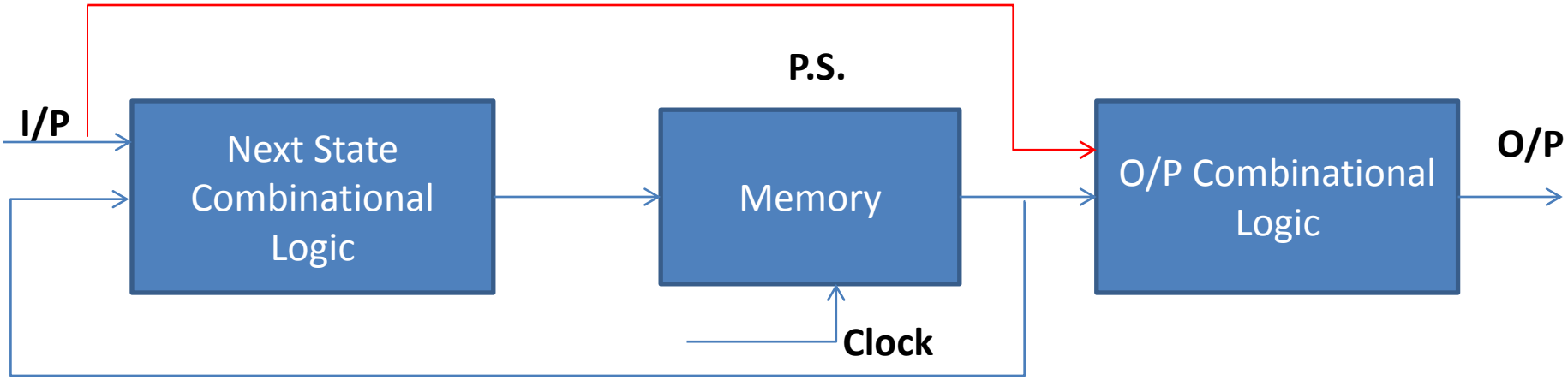
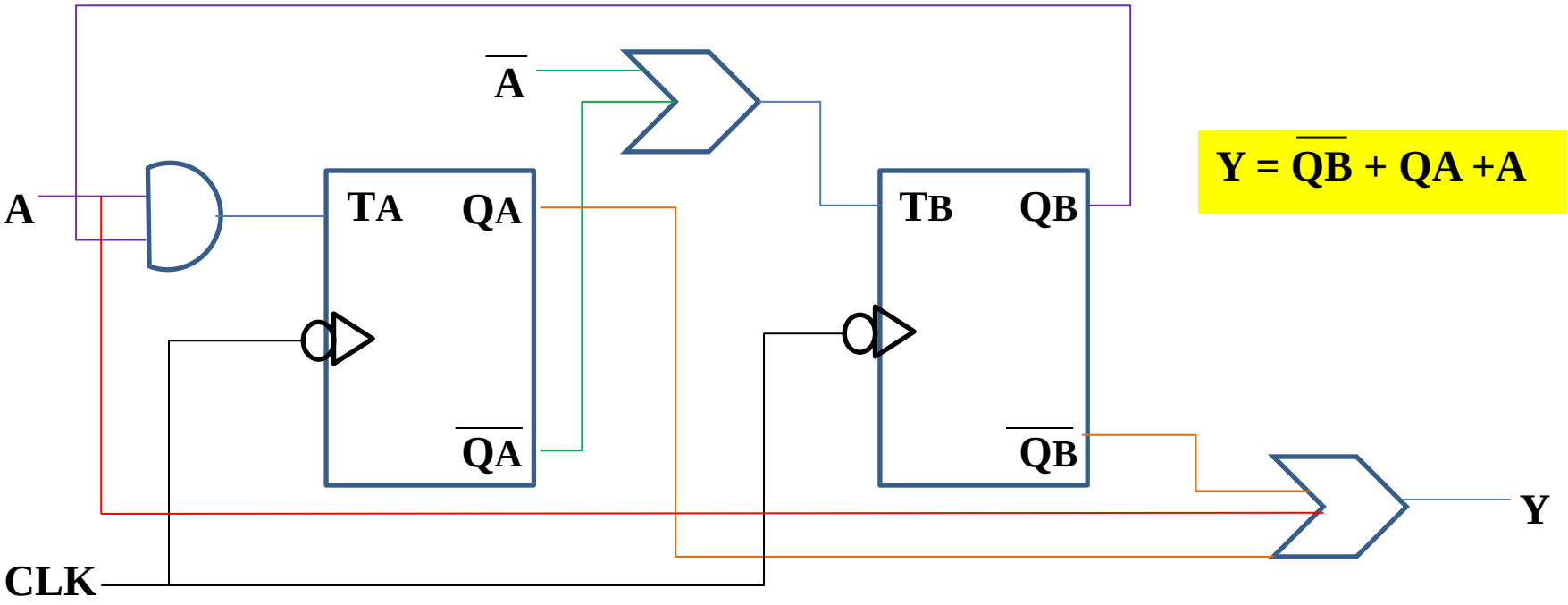


Fig: Mealy State Machine

Mealy State Machine (Contd..)



$Y = \overline{QB} + QA + A$

Design Procedure

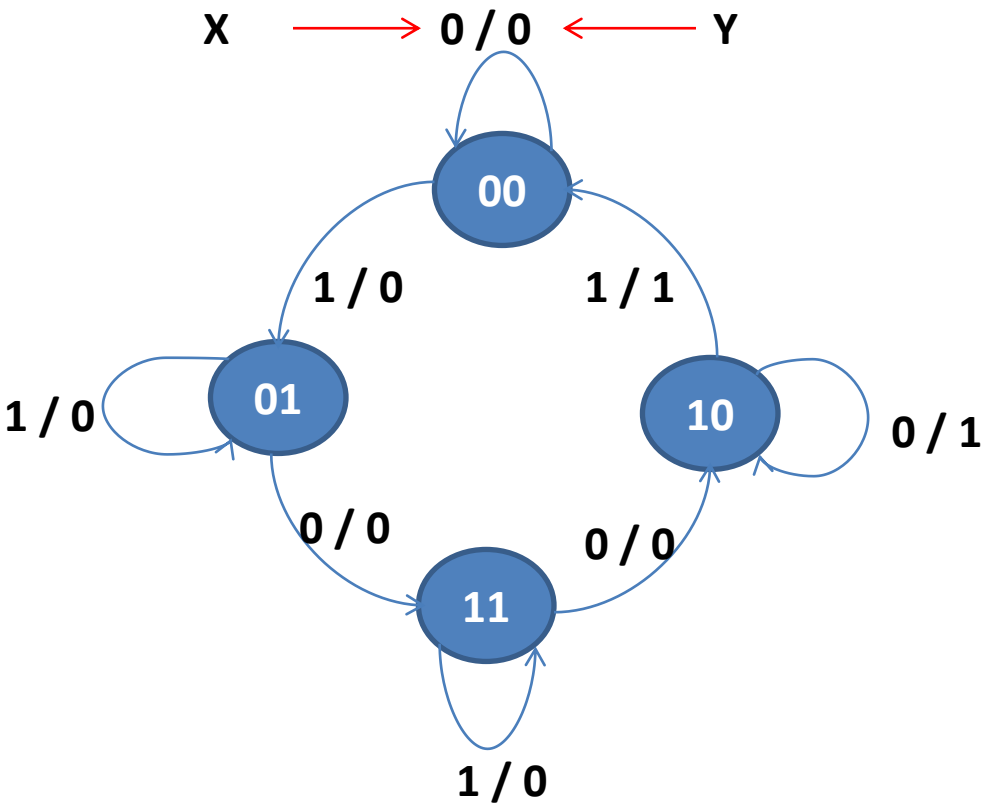
- Step 1: A state dia or timing dia is given , which describes the behavior of the ckt that is to be designed
- Step 2: Obtain the State table
- Step 3: The no. of states can be reduced by state reduction method
- Step 4: Do State assignment (if required)
- Step 5: Determine the no. of FF required & Assign letter symbol

Design Procedure (Contd..)

- Step 6: Decide the type of FF to be used
- Step 7: Derive the ckt excitation table from state table
- Step 8: Obtain the expression for ckt O/P & FF I/P
- Step 9: Implement the ckt

Example

- Step 1: State Diagram



Example (Contd..)

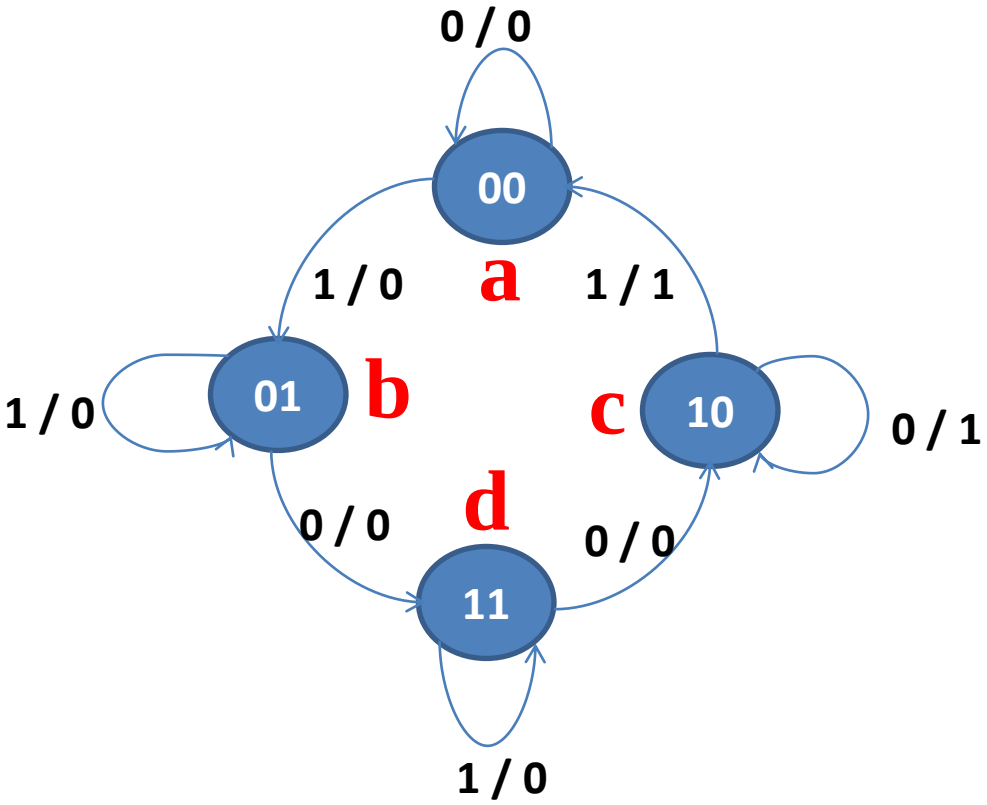
- Step 2: State Table

Present State		Next State				Output (Y)	
QA	QB	X=0		X=1		X=0	X=1
0	0	0	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	1	0	0	0	1	1
1	1	1	0	1	1	0	0

- Step 3: State Reduction
  - Check N.S.
  - If the N.S. for the 2 states are same then only reduction is possible
  - Here there is no state reduction

Example (Contd..)

- Step 4: State assignment (If required)



a = 00
b = 01
c = 10
d = 11

Example (Contd..)

- Step 5: No. of FF required? Assign letter symbol
  - 4 states can be implemented by using **2 FFs**
  - Letter Symbol:
    - FF-1 = A
    - FF-2 = B

- Step 6: Decide FF
  - **T FF**

Example (Contd..)

- Step 7: Ckt Excitation table

Present State		X	Next State		FF Inputs		Y
QA	QB		QA+	QB+	TA	TB	
0	0	**0**	0	0	0	0	0
0	0	**1**	0	1	0	1	0
0	1	**0**	1	1	1	0	0
0	1	**1**	0	1	0	0	0
1	0	**0**	1	0	0	0	1
1	0	**1**	0	0	1	0	1
1	1	**0**	1	0	0	1	0
1	1	**1**	1	1	0	0	0

Truth Table of T FF

CLK	T	Qn+1
0	X	Qn
1	0	Qn
1	1	$\overline{Qn}$

Characteristic Table

Qn	Qn+1	T
0	0	0
0	1	1
1	0	1
1	1	0

Example (Contd..)

- Step 8: Expressions
- Use K-Map

$$TA = \overline{QA}.QB.\overline{X} + QA.\overline{QB}.X$$

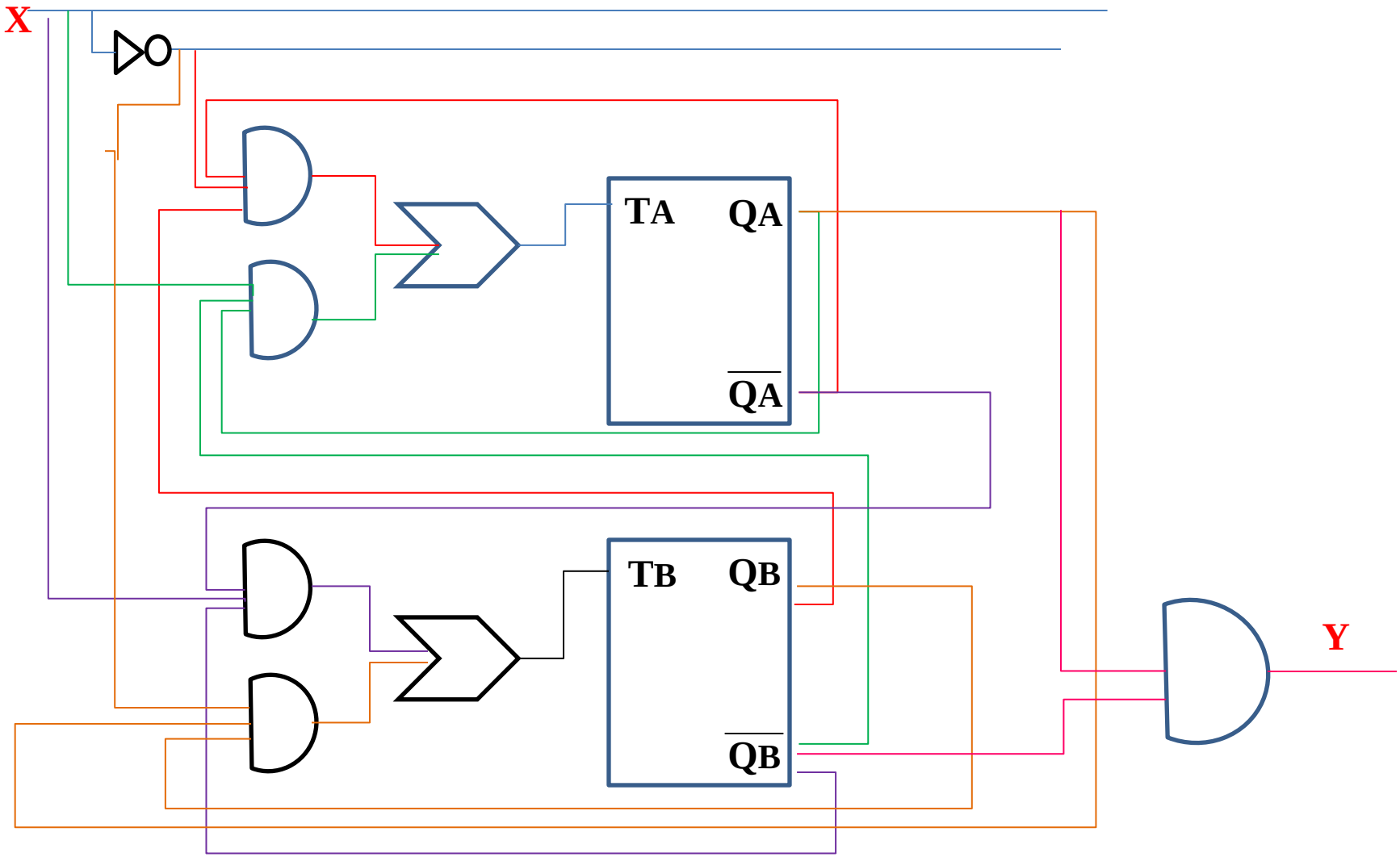
$$TB = \overline{QA}.\overline{QB}.X + QA.QB.\overline{X}$$

$$Y = QA.\overline{QB}.\overline{X} + QA.\overline{QB}.X$$
$$= QA\overline{QB}(\overline{X}+X)$$
$$= QA\overline{QB}.1$$

$$Y = QA\overline{QB}$$

Example (Contd..)

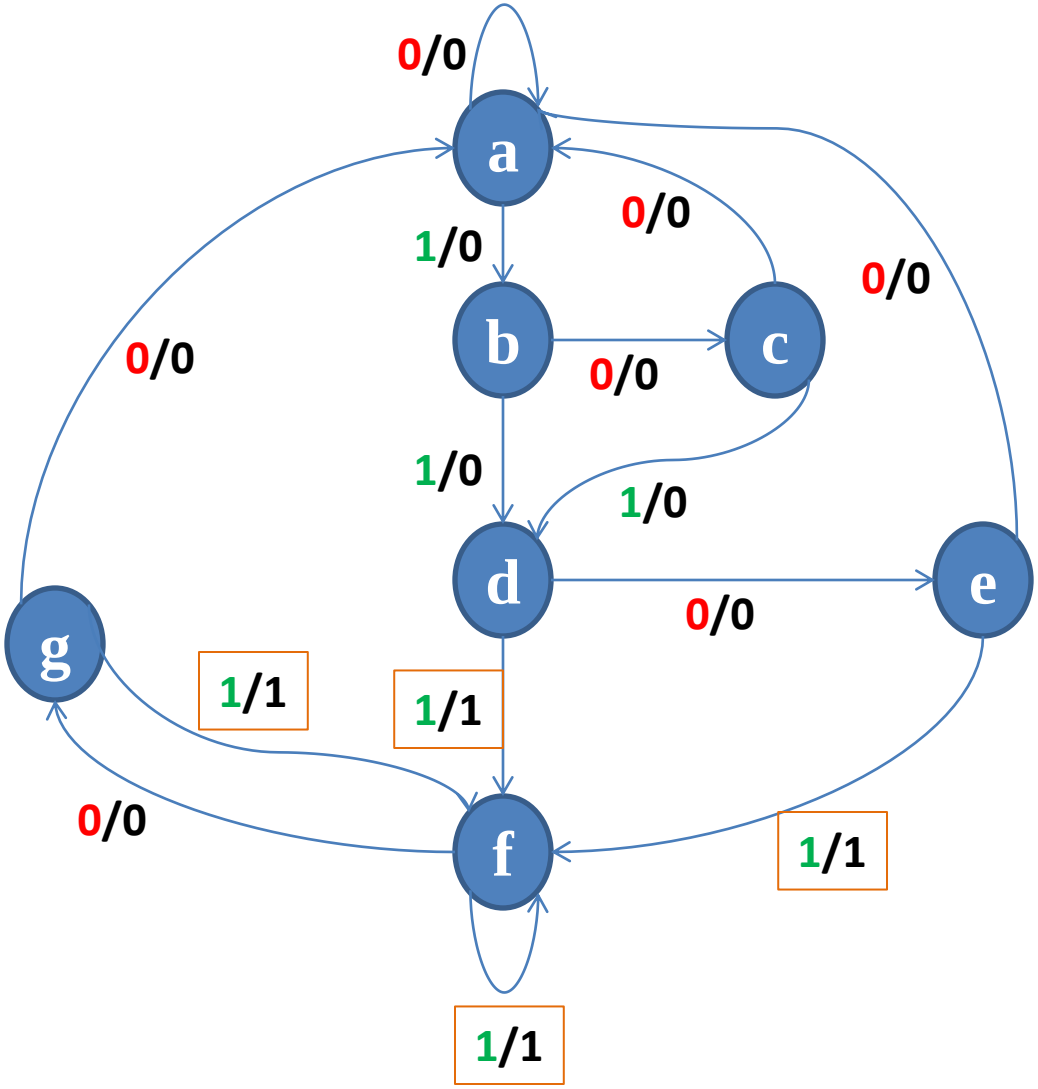
- Step 9: Implement Circuit



State Reduction

Present State	Next State		Output	
	X=0	X=1	X=0	X=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

State Reduction (Contd..)



State Reduction (Contd..)

- State Assignment:

For 4 States	For 5 States
a=00	a=000
b=01	b=001
c=10	c=010
d=11	d=011
	e=100

- If Next State & O/P of 2 Present States are same, then we can eliminate one state

State Reduction (Contd..)

Present State	Next State		Output	
	X=0	X=1	X=0	X=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

Therefore, **e = g**

State Reduction (Contd..)

Present State	Next State		Output	
	X=0	X=1	X=0	X=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	**e = g**	f	0	1
e	a	f	0	1
f	g	f	0	1

Therefore, **d = f**

State Reduction (Contd..)

Present State	Next State		Output	
	X=0	X=1	X=0	X=1
a	a	b	0	0
b	c	d	0	0
c	a	D	0	0
d	**e = g**	f = d	0	1
e	a	f = d	0	1

State Reduction (Contd..)

- **Modified State Diagram:**