

Unit 3

Synchronous Sequential logic

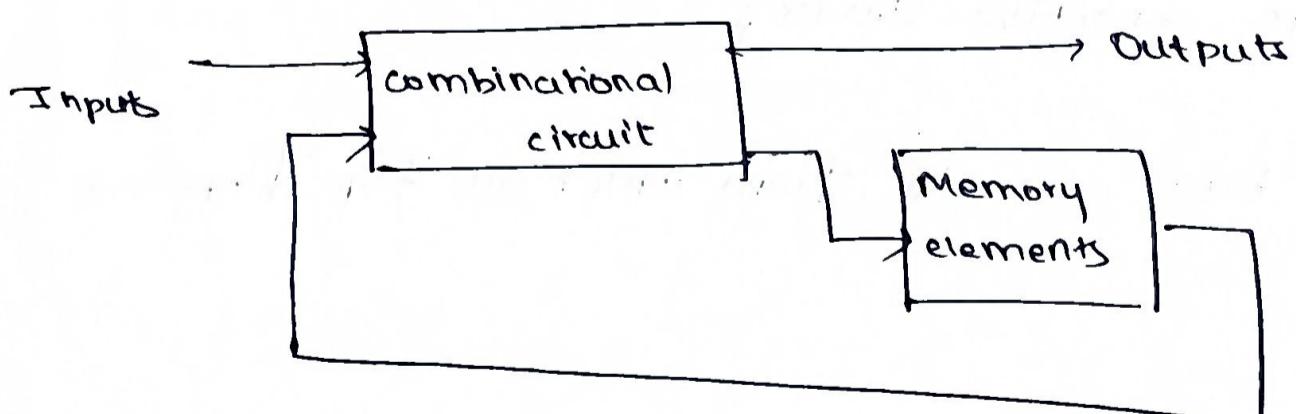
Syllabus:

sequential circuits - storage elements : latches, flipflops

analysis of clocked sequential circuits, state reduction and assignment - design procedure - registers and counters

Sequential Circuits

- consist of a combinational circuit along with a storage element
- storage device stores binary information that defines the state of the sequential circuit at a given time
- The next state of the storage elements is a function of external inputs and the present state.



* Synchronous and Asynchronous Sequential Circuits

(3)

Synchronous

→ behaviour is defined from the knowledge of its signals at discrete instants of time

Asynchronous

→ behaviour is defined by input signals and the order in which the inputs change

→ timing devices like a clock are used to generate clock pulses

→ Time delay devices are used.

* Storage Elements : Latches and Flipflops

→ Storage elements that operate with signal levels = latches

→ Storage elements controlled by a clock transition = flipflops

→ Latches = level sensitive devices

→ Flipflops = edge sensitive devices

→ Latches are the basic circuits from which all flipflops are constructed.

* Latches

① S-R Latch

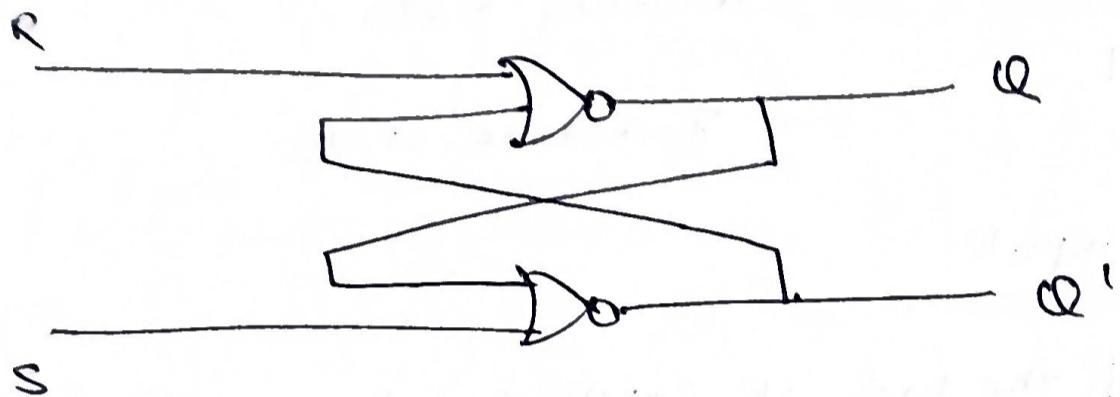
→ has 2 cross coupled NOR gates

3

If $Q = 1, Q' = 0 \rightarrow$ set state

$Q = 0, Q' = 1 \rightarrow$ reset state

→ However, If both inputs are 1 at the same time, then the device will enter an unpredictable state called the forbidden state.



Function Table

S	R	Q	Q'
1	0	1	0
0	0	1	0
<hr/>			
0	1	0	1
<hr/>			
0	0	0	1
<hr/>			
1	1	0	0
<hr/>			

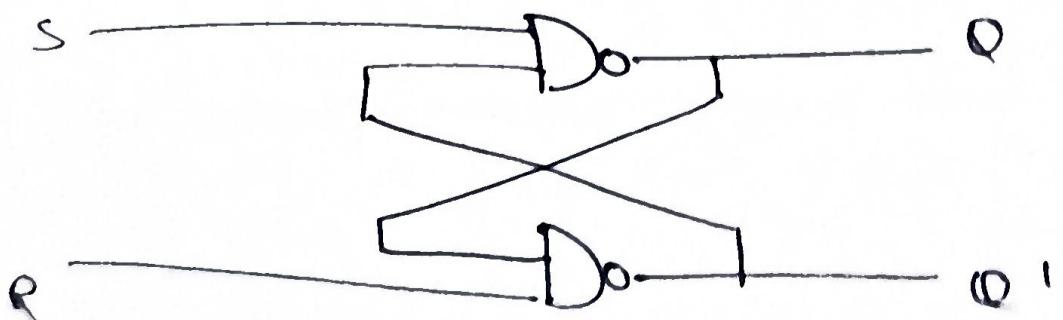
→ memory state

→ memory state

→ forbidden state

② S'R' latch

→ NAND form of SR latch

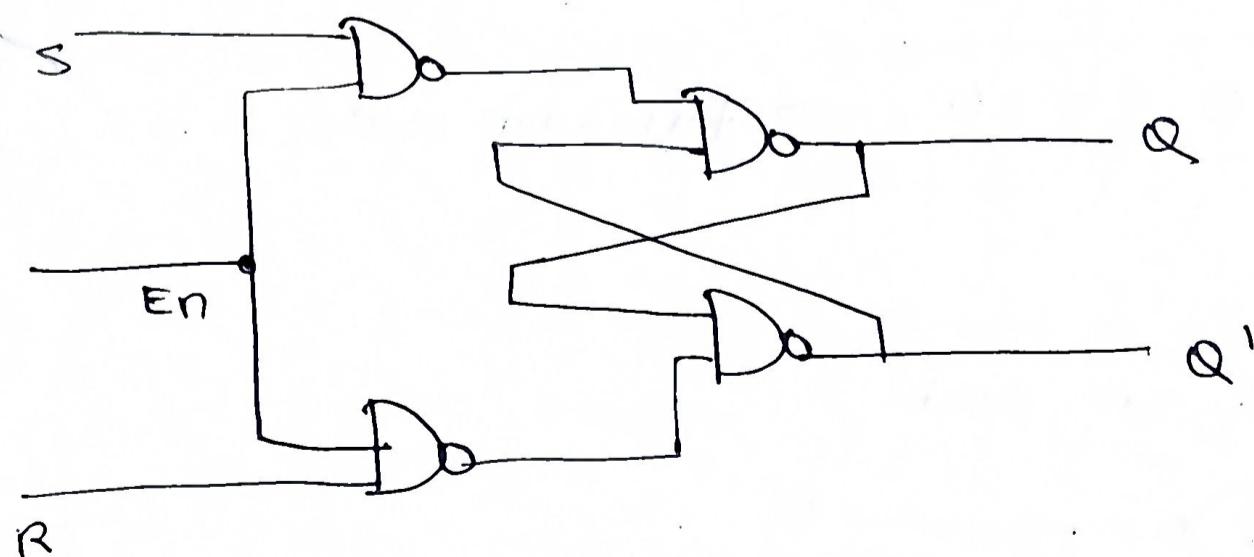


Function Table

S	R	Q	Q'	
1	0	0	1	
1	1	0	1	→ memory state
0	1	1	0	
1	1	1	0	→ memory state
0	0	1	1	→ forbidden state

③ SR with control input

- The operation of the basic SR latch can be modified by providing an additional input signal that determines when the state of the latch can be changed.
- The circuit consists of the NAND form of the basic SR latches (S'R' form) along with 2 additional NAND gates
- The control input En acts as an enable signal for the other 2 inputs.



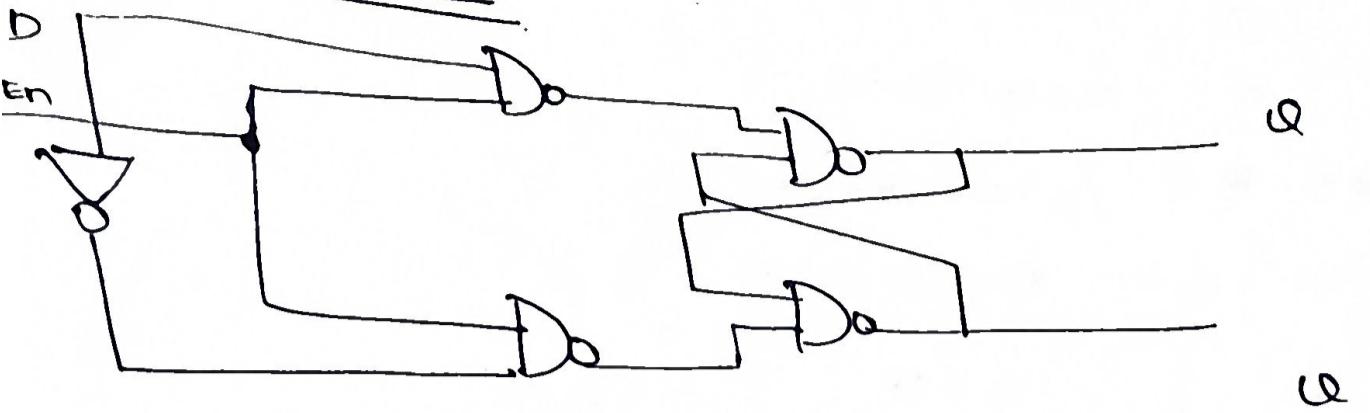
Function Table

En	S	R	Next State
0	0x	0x	No change.
1	0	0	No change
1	0	1	$Q = 0$
1	1	0	$Q = 1$
1	1	1	Indeterminate

④ D latch - Transparent Latch

- The undesirable state of the indeterminate state in the SR state can be eliminated by ensuring that the inputs S & R are never equal to 1 at the same time.
- This is done using a D latch.
- The D latch has only 2 inputs D (data) and En (enable).
- The D input goes to the S input; D's complement is applied to the R input.
- When $En = 1, D = 1 \Rightarrow Q = 1$
- $En = 1, D = 0 \Rightarrow Q = 0$
- The D latch is so called because of its ability to hold data in its internal storage.
- The output directly follows from the input.

Logic Diagram



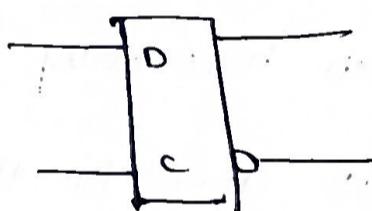
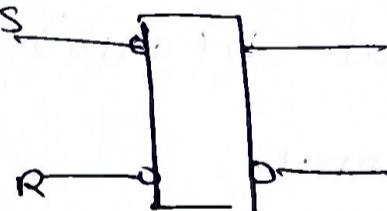
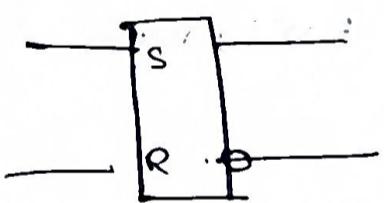
		Next State of Q
En	D	
0	x	No change
1	0	$Q = 0$ → Reset state
1	1	$Q = 1$ → Set state.

Graphic Symbols for Latches

SR Latch

S'R' Latch

D Latch



Flip Flops

- A flip flop is a circuit with 2 stable states that can be used to store binary data.
- The stored input can be changed by applying varying inputs.
- Flip flop circuits are constructed in such a fashion that they operate properly when they are part of a sequential

circuit that employs a common clock.

Changing the state of the flip flop

- The state of the flip flop is switched by a change in the control input.
- This momentary change is called a trigger and the transition it causes is said to trigger the flip flop.
- Flip flops are edge triggered.

Positive Edge Triggered Flip Flop : When the clock makes a transition from 0 to 1.



Negative Edge Triggered Flip Flop : When the clock makes a transition from 1 to 0.



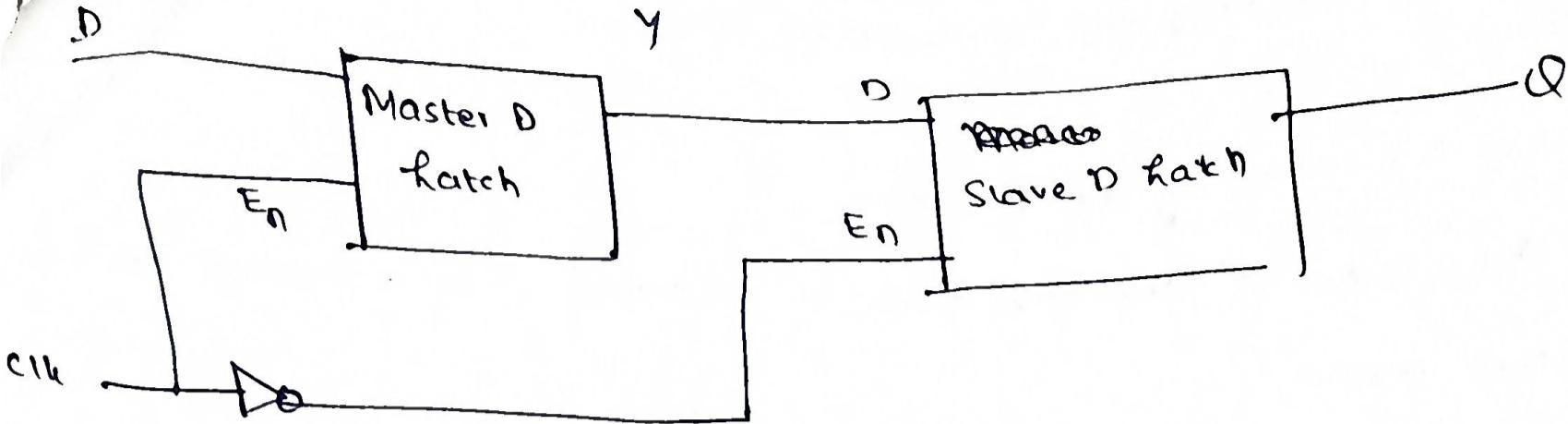
* Edge Triggered D Flip Flop

→ An edge triggered D flip flop consists of 2 D latches, and an inverter

→ The first latch = master latch

The second latch = slave latch

→ This kind of flip flop is called the master-slave flip flop.



→ The circuit is such that its output changes only at the negative edge of the synchronizing clock

→ If $C=0$: → the output of the inverter is 1

→ the master latch is disabled

→ the slave latch is enabled.

→ the output Q is the master output.

→ If $C=1$: → the inverter is 0

→ the slave latch is deactivated

→ the master latch is enabled

→ changes may occur

→ output not affected, since slave latch is deactivated.

→ If $C=0$: → the master is disabled

→ the output at Y is transferred to the slave latch, which is enabled

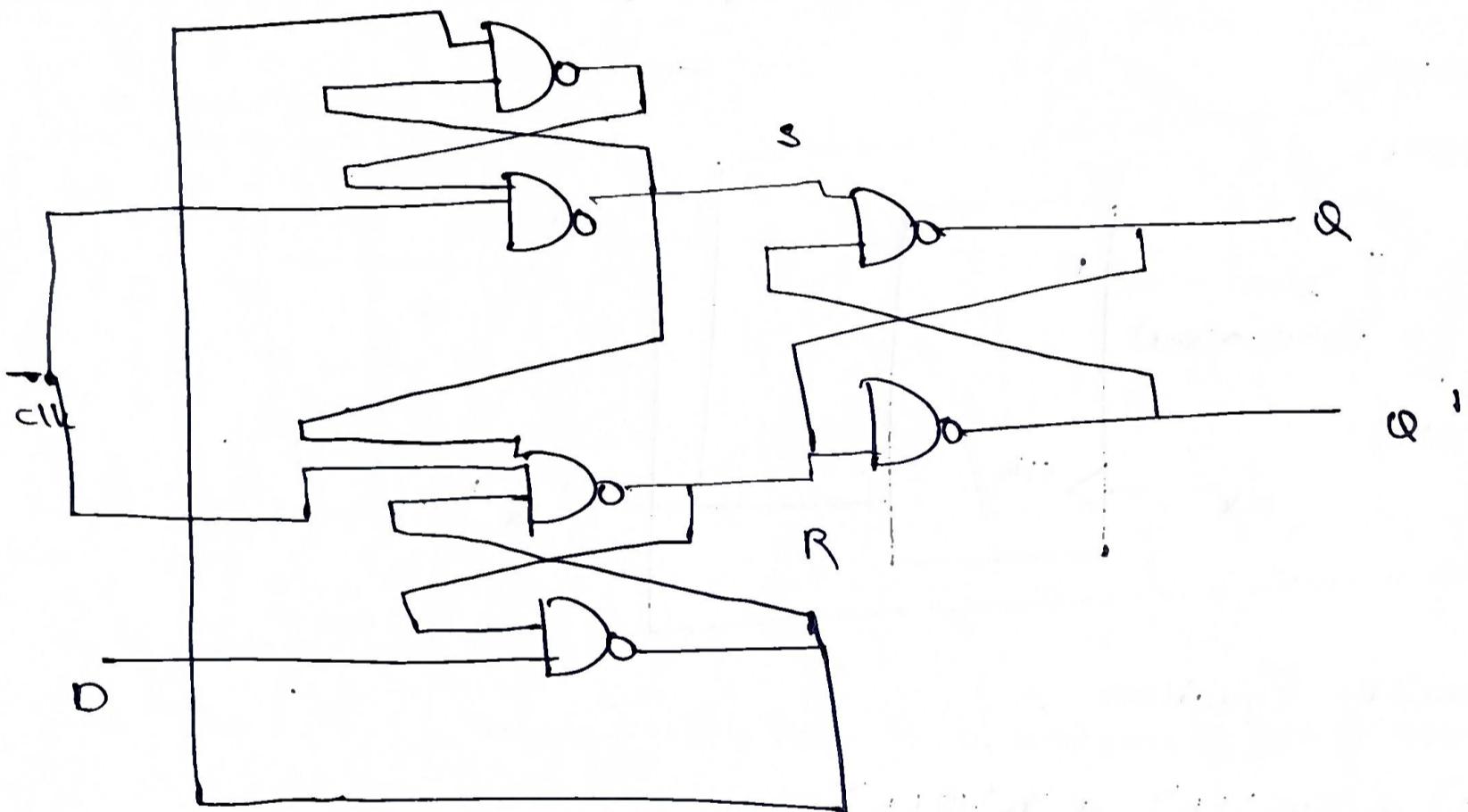
→ there is a change in the o/p.

∴ There is a change in the o/p of the flip-flop only during the transition of the clock from 1 to 0.

(9)

* D Type Positive Edge Triggered Flip Flop

- this flip-flop is made up of 3 SR Latches
- Q Latches respond to the external D (data) and clock inputs
- the third latch provides the output for the flip flop.



- There is a change in the output only when the clock transitions from 0 to 1.

Characteristic Equation

$$Q(t+1) = D$$

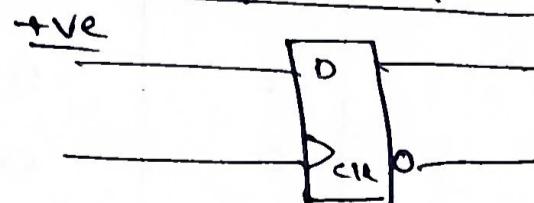
Characteristic Table

D	Q(t+1)
0	0
1	1

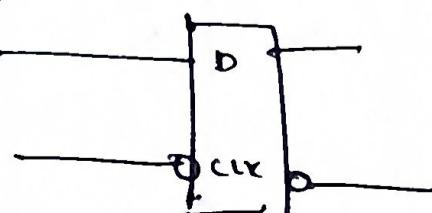
Reset

Set

Graphic Symbols

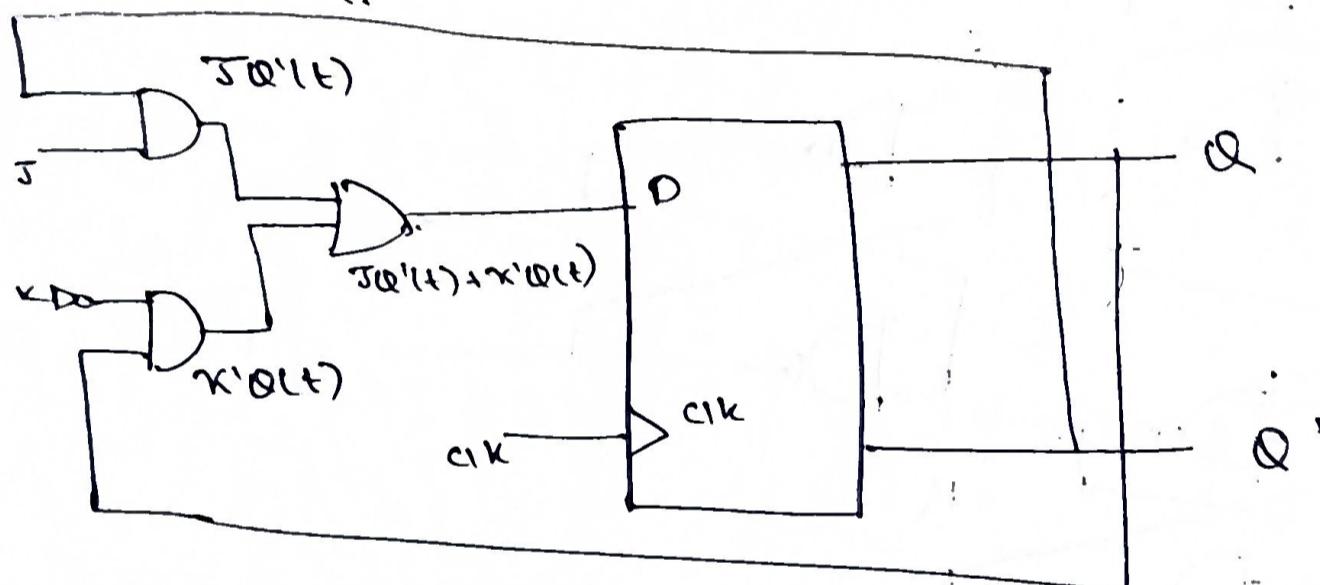


-ve



Q) JK Flip Flop

- There are 3 operations that can be performed using flip-flops: set it to 1, reset to 0, or complement the output.
- The JK flip flop performs all 3 operations.
- It has 2 inputs and performs 3 operations.
- Consider the following JK flip flop made with a D flip flop and gates.



Characteristic Equation

$$Q(t+1) = JQ'(t) + K'Q(t)$$

Truth Table

$Q(t)$	J	K	$Q'(t)$	$JQ'(t)$	K'	$K'Q(t)$	$JQ'(t) + K'Q(t)$	$Q(t+1)$
0	0	0	1	0	1	0	0	0
1	0	0	0	0	1	1	1	1
0	0	1	1	0	0	0	0	0
0	1	0	0	0	0	0	0	0
0	1	0	1	1	1	0	1	1
1	0	0	0	0	1	1	1	1
1	0	1	0	0	1	0	1	1
0	1	1	1	1	0	0	1	1
1	1	1	0	0	0	0	0	0

$Q(t)$	$Q(t+1)$	Function
$JQ' + K'Q$		
0 1	0 1	$Q(t)$
0 1	0 0	0
0 1	1 1	1
0 1	0 0	$Q'(t)$

Characteristic Table

J	K	Q
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$Q'(t)$

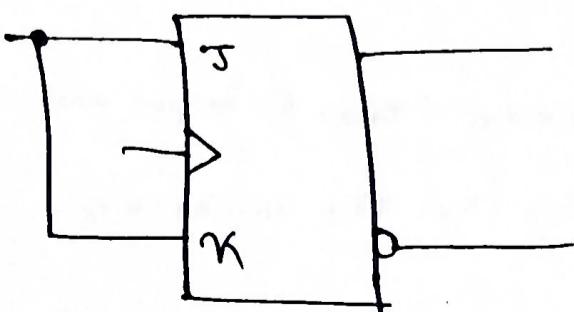
③ T- Flip Flop

→ The T (Toggle) flip flop is a complementing flip flop.

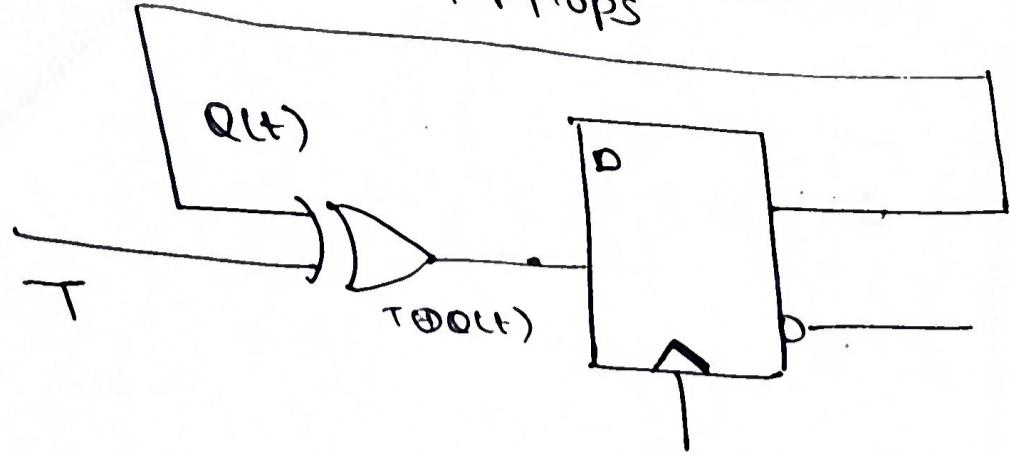
→ It can be created in the following methods :

(a) using JK Flip

→ by tying the J and K inputs together



(b) using D flipflops



Characteristic Equation

$$Q(t+1) = T \oplus Q(t)$$

$Q(t)$	T	$Q(t+1)$	$T \oplus Q(t)$	$Q(t)$
0	0	0	0	$Q(t)$
1	0	1	1	
0	1	1	1	$Q'(t)$
1	1	0	0	

Characteristic Table

$Q(t)$	$Q(t+1)$
0	$Q(t)$
1	$Q'(t)$

Definitions

- Set-Up Time: minimum time during which the D input must be maintained at a constant value, prior to the occurrence of the clock transition.

- Hold Time: minimum time during which the input must not change after the application of the positive transition of the clock

- Propagation Delay Time : Interval between the trigger edge and the ~~ear~~ stabilization of the output to a new state.

Analysis of Clocked Sequential Circuits

The analysis of clocked sequential circuits consists of 3 parts

1. The State Equation

Specifies the next state as a function of the present state and inputs

2. The State Table

consists of the present state, input next state and output.

3. The State Diagram

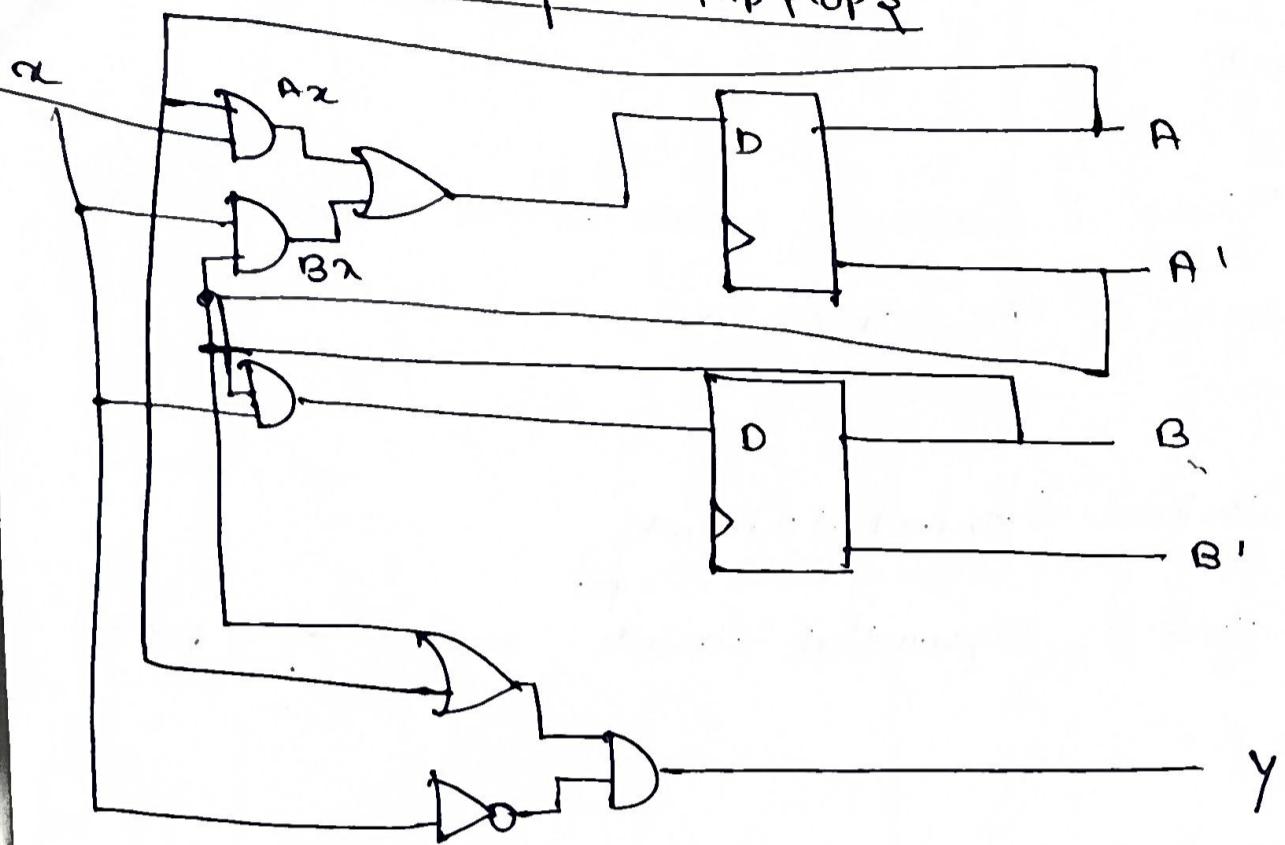
→ information from a state table is represented graphically in a state diagram. The state is represented by a circle and transitions between states are indicated by directed lines connecting the circles.

Analysis Procedures

- Find the input equation
- Substitute the input equations into the flip flop characteristic eqn, to obtain the state eqns

→ Use the state equation to find the next state value

Ex: Analysis using D flip flops



Finding the state equations

$$A(t+1) = Ax + Bx$$

$$B(t+1) = A'x$$

$$Y = Ax' + Bx'$$

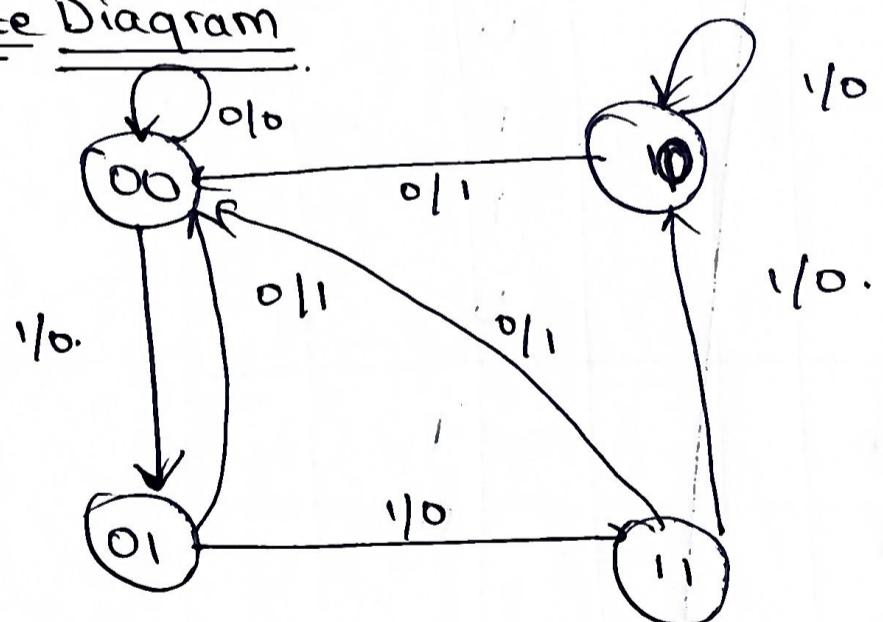
State Table

Present State		x	Ax Bx	A'	A'x	x'	Ax'	Bx'	Ax + Bx	Ax' + Bx'	A(t+1)	B(t+1)	A'(x)
A	B												
0	0	0	0	0	1	0	1	0	0	0	0	0	0
0	0	1	0	0	1	1	0	0	0	0	0	0	1
0	1	0	0	0	1	0	1	0	1	0	1	0	0
0	1	1	0	1	1	1	0	0	0	0	1	0	1
1	0	0	0	0	0	0	1	1	0	0	1	0	0
1	0	1	1	0	0	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	1	1	1	0	0	1	0
1	1	1	1	1	0	0	0	0	0	0	1	0	0

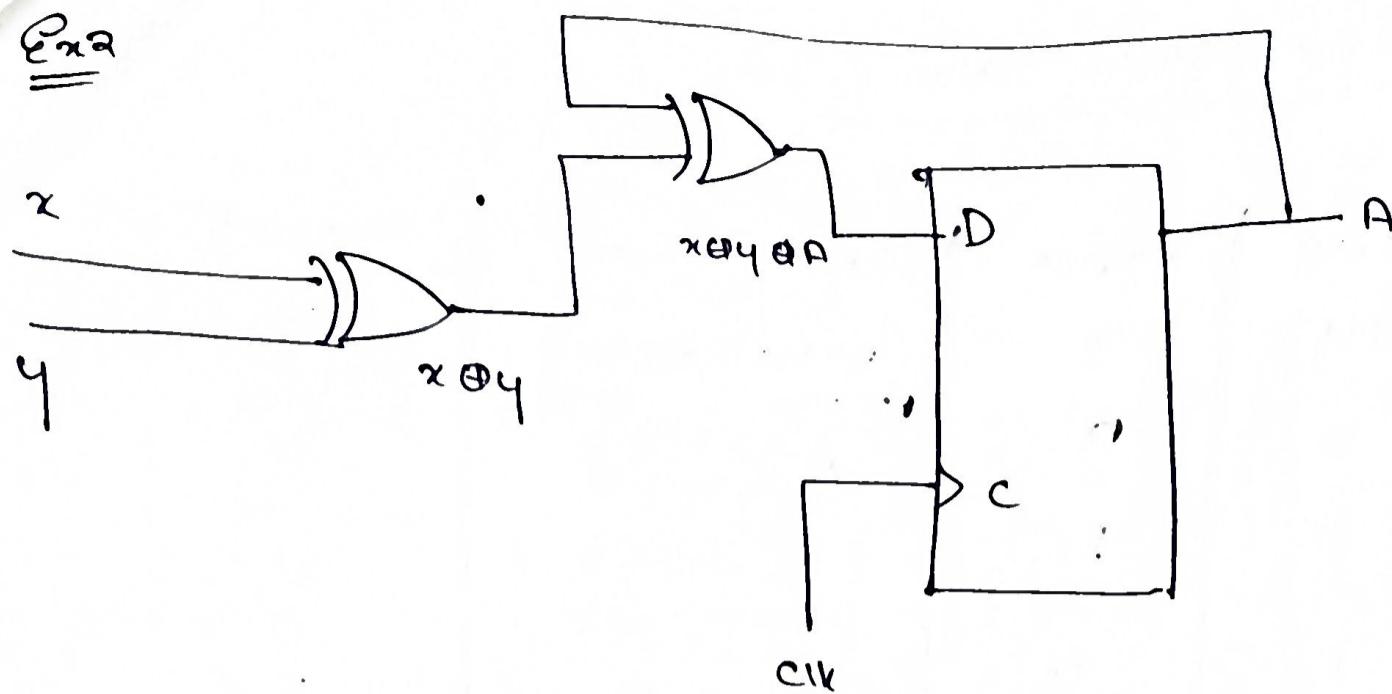
Consolidated State Table

Present State $A(t)$ $B(t)$	Input x	Next State			Output y
		$A(t+1)$ $Ax + Bx$	$B(t+1)$ $A'x$	$A'(x)$	
0 0	0	0	0	0	0
0 0	1	0	0	1	0
0 1	0	0	0	0	1
0 1	1	1	1	1	0
1 0	0	0	0	0	1
1 0	1	1	0	0	0
1 1	0	0	0	0	1
1 1	1	1	0	0	0

State Diagram



EQA



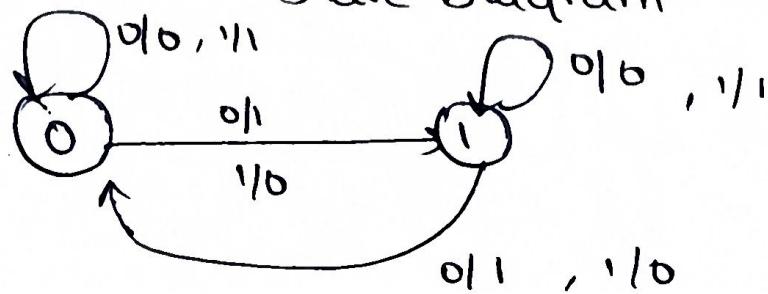
State Equations

$$A(t+1) = x \oplus y \oplus A$$

State Table

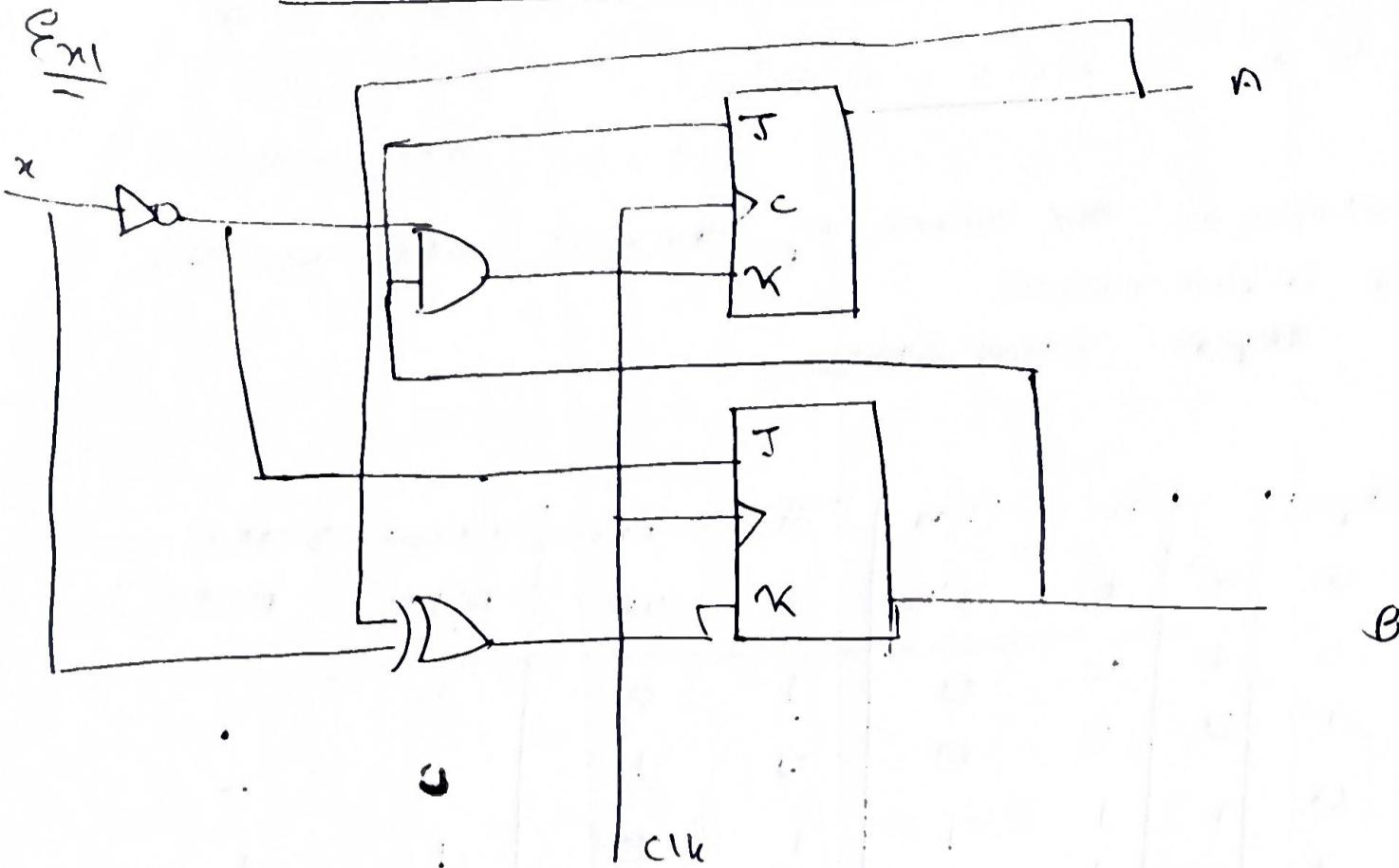
Present State A	Inputs		Next State $A(t+1)$ $x \oplus y \oplus A$
	x	y	
0	0	0	0
	0	1	1
	1	0	1
	1	1	0
1	0	0	1
	0	1	0
	1	0	0
	1	1	1

State Diagram



* Analysis of JK Flip Flops

(17)



Step 1: Find the input equations

$$J_A = B \quad K_A = Bx'$$

$$J_B = x' \quad K_B = A \oplus x$$

Step 2: Substitute i/p equations in the characteristic eqns of
o JK flip flop.

① Characteristic Equation

$$Q(t+1) = JQ'(t) + K'Q(t)$$

$(A \oplus x)'$
is $A \odot x$

$$A(t+1) = J_A' + K'_A A \Rightarrow BA' + (Bx')' A$$

$$B(t+1) = J_B' + K'_B B \Rightarrow x'B' + (Ax' + x'B')' \cdot B$$

$$A(t+1) = BA' + AB' + Ax$$

$$B(t+1) = B'x' + ABx + A'Bx'$$

$$A(t+1) = A'B + AB' + Ax$$

$$JA = B$$

$$KA = x'B$$

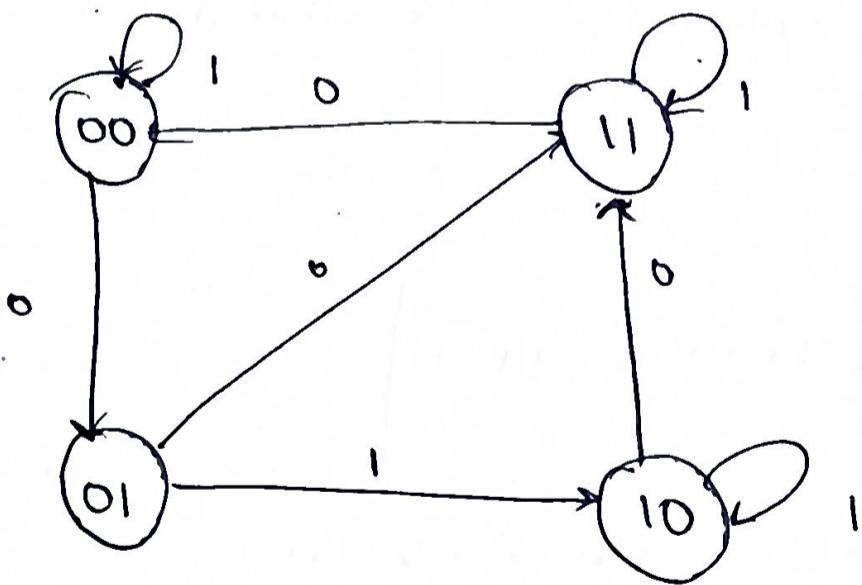
$$JB = x'$$

$$KB = x \oplus A$$

$$B(t+1) = B'x' + ABx + A'Bx'$$

Step 3 : To determine the values of the next state, and the flip flop input values
 Present State Input Next State

Present State		Input		JA		KA		JB		KB		Next State	
A	B	x	x'	B	x'B	x'	x ⊕ A	x'	x'B	x ⊕ A	A(t+1)	B(t+1)	
0	0	0	1	0	0	0	0	1	0	0	0	1	
0	0	1	0	0	0	0	0	0	1	0	0	0	
0	1	0	1	1	1	1	1	1	0	0	1	1	
0	1	1	0	1	0	0	1	0	1	1	1	0	
1	0	0	1	0	0	0	0	1	1	1	1	1	
1	0	1	0	0	0	0	0	0	0	0	1	0	
1	1	0	1	1	1	1	1	1	1	1	0	0	
1	1	1	0	1	1	0	0	0	0	0	1	1	



Ex2 A sequential circuit has 2 JK flipflops A2 B

and one input x . The circuit is described as follows

$$J_A = x \quad K_A = B'$$

$$J_B = x \quad K_B = A$$

(i) Derive the state equations $A(t+1)$ and $B(t+1)$, and draw the state table and state diagrams

$$Q(t+1) = JQ'(t) + K'Q(t)$$

$$A(t+1) = J_A A' + K_A' A \Rightarrow xA' + BA$$

$$B(t+1) = J_B B' + K_B' B \Rightarrow xB' + A'B$$

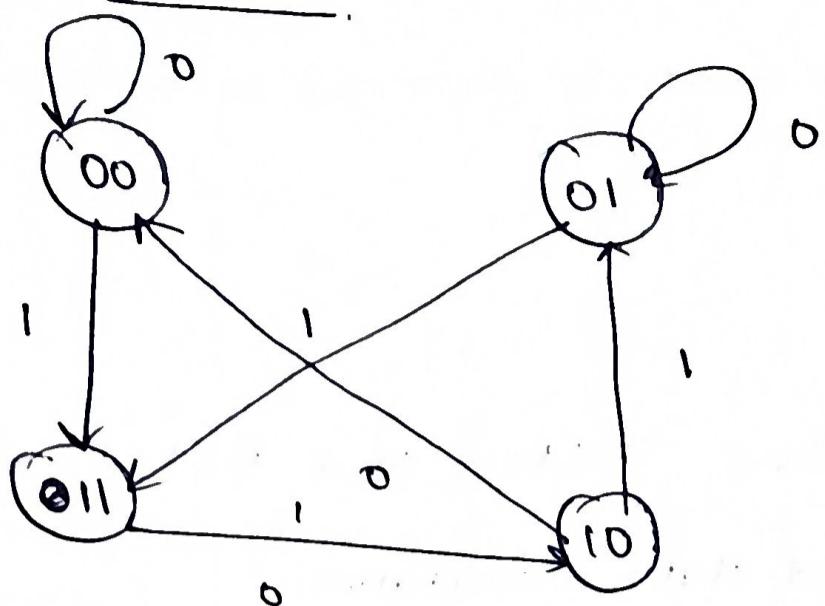
char Table

J	K	ns	$Q(t)$
0	0	ns	$Q(t)$
0	1	0	$Q(t)$
1	0	1	$Q(t)$
1	1	0	$Q'(t)$

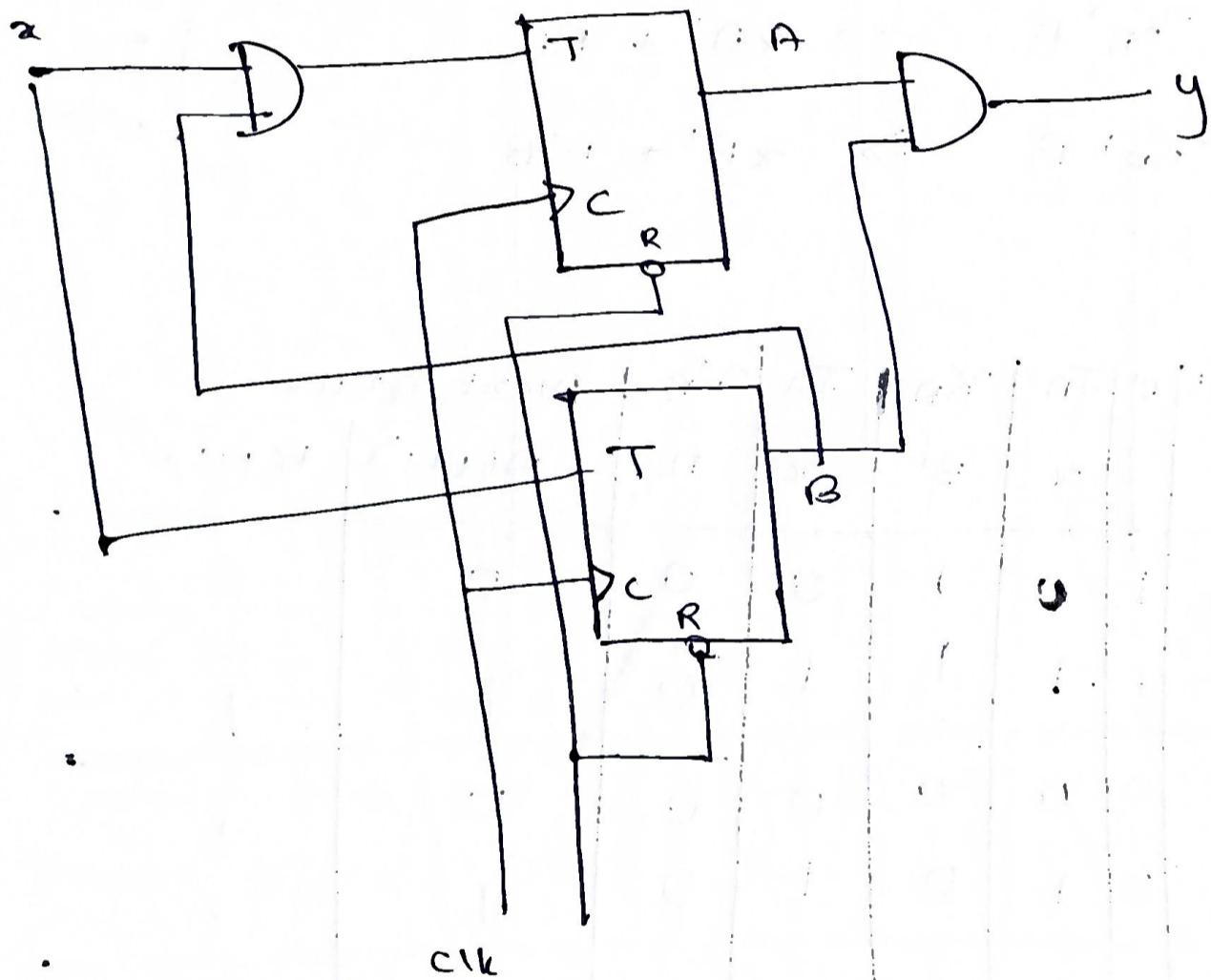
State Table

Present State		Input x	J_A	K_A	J_B	K_B	Next State		
A	B		x	x	B'	x	A	$A(t+1)$	$B(t+1)$
0	0	0	1	0	1	0	0	0	0
0	0	1	1	1	1	1	0	1	1
0	1	0	0	0	0	0	0	0	1
0	1	1	0	1	0	1	0	1	1
<hr/>									
1	0	0	1	0	1	0	1	0	0
1	0	1	1	1	1	1	1	0	1
<hr/>									
1	1	0	0	0	0	0	1	1	0
1	1	1	0	1	0	1	1	1	0

State Diagram



* Analysis with T Flip Flops



Step1: Input Equations

-Ta . Ba

$$T_B : x$$

y : AB

Characteristic Eqn \Rightarrow Table

$$Q(t+1) = T \oplus Q = TQ' + QT'$$

T	Next State
0	$Q(t)$
1	$Q'(t)$

Step 2: State Equations

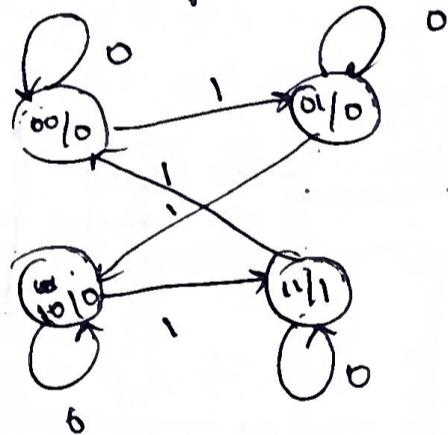
$$\begin{aligned} A(t+1) &= TAQ' + CQT_A' \\ &= -BxU \end{aligned}$$

$$A(t+1) = A'Bx + AB' + x'A$$

$$A(t+1) = A'Bx + Ax' + AB'$$

$$\begin{aligned} B(t+1) &= TBQ' + BTB' \\ &= x'B' + Bx' \end{aligned}$$

$$B(t+1) = B \oplus x$$



State Table

Present State		Input x	TA TB		Next State		Output Y
A	B		Bx	x	A(t+1)	B(t+1)	
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	0	0	1	0
0	1	1	1	1	1	0	0
1	0	0	0	0	1	0	0
1	0	1	0	1	1	1	0
1	1	0	0	0	1	1	1
1	1	1	1	1	0	0	1

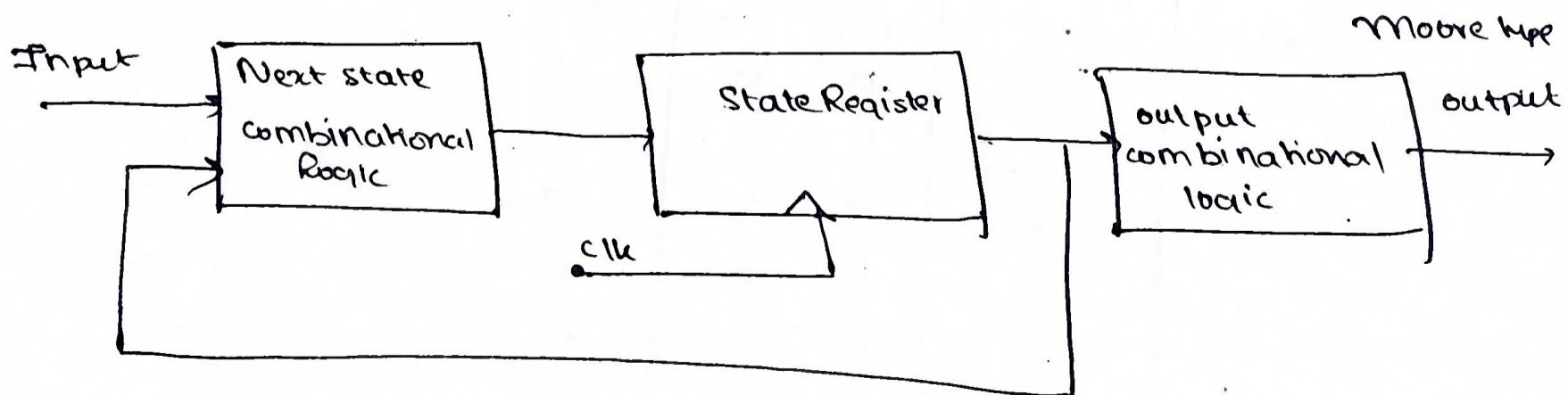
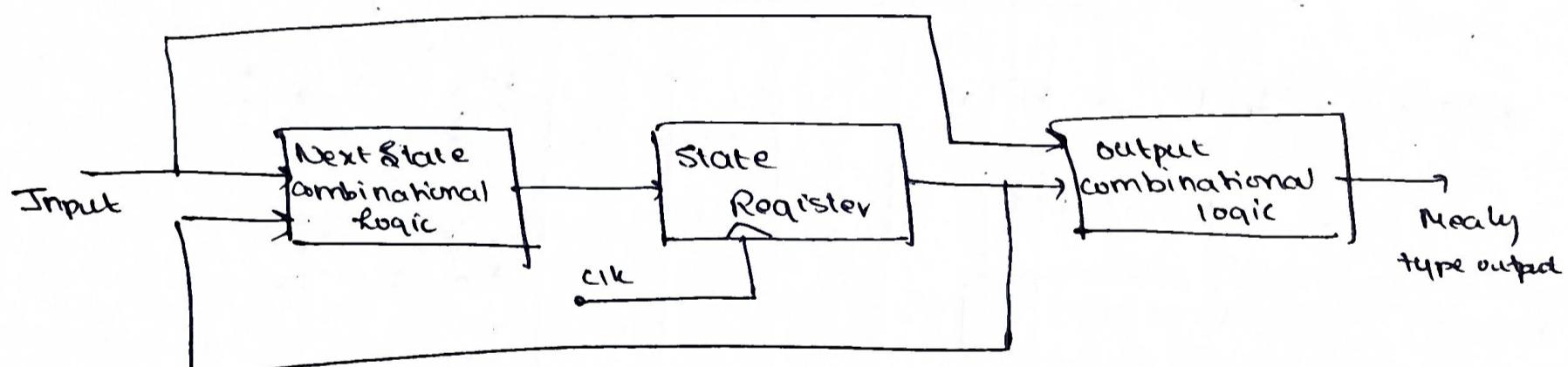
Mealy and Moore Models

→ The most general model of a sequential circuit has inputs outputs and internal states. It is customary to distinguish between 2 models of sequential circuits: the Mealy model and the Moore model.

→ These 2 models differ in the way the output is generated.

Mealy model: o/p is a function of both the present state and inputs

Moore model: o/p is a function of the present state only

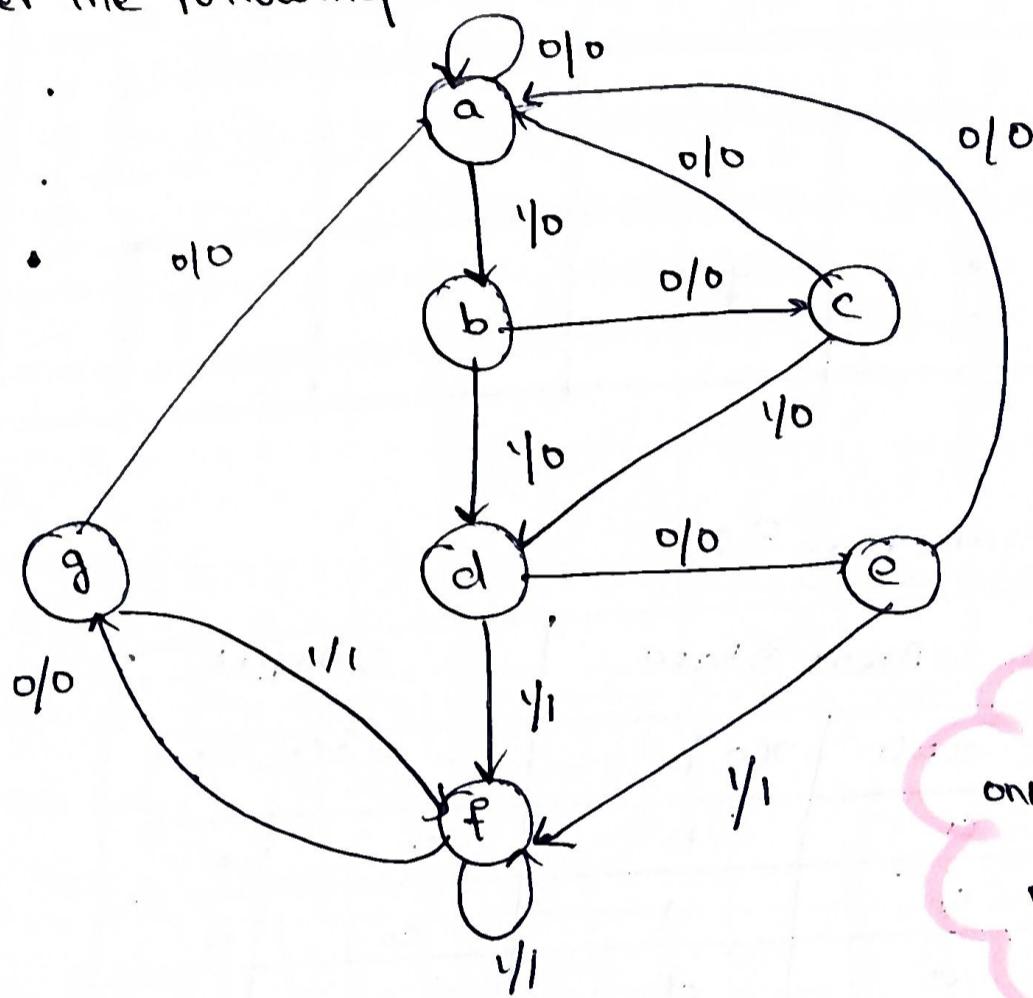


State Reduction and Assignment

→ The reduction in the number of flip flops in a sequential circuit is called state reduction.

Ex:

Consider the following state diagram



Note: Reduction can only be done if the next state and output are the same.

The corresponding state table is

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

Reduction 1: $\ominus e = g$

State Table after reduction 1:

eg?

Present State

Next State

Output

$x=0$

$x=1$

$x=0$

$x=1$

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 e

f

0

1

Reduction 2: d=f.

State Table after reduction 2:

Present State

Next State

Output

$x=0$

$x=1$

$x=0$

$x=1$

a

b

0

0

b

d

0

0

c

d

0

0

d

f

0

1

e

a

0

1

e

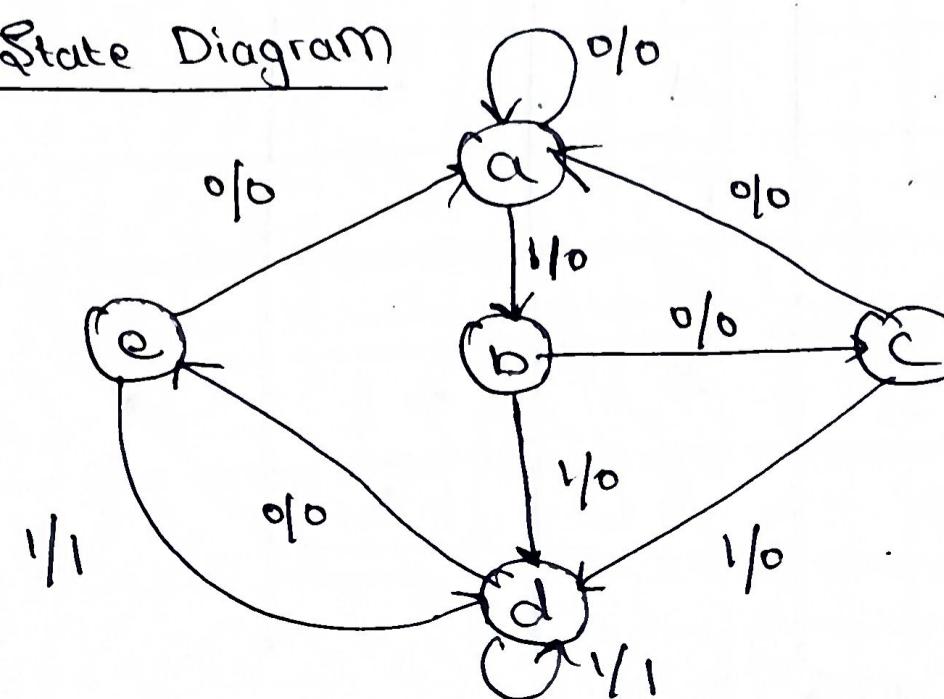
f

0

1

Though the next state values a \equiv d match, the state cannot be reduced. since the outputs do not match

Reduced State Diagram



Ex-2 Reduce the following state table and draw

Q5

the corresponding reduced state diagram.

Present State	Next State		Output	
	$x=0$	$x=1$	$x=0$	$x=1$
a	e	a	0	0
b	c	d	0	0
c	e	f	0	0
d	h	b	1	0
e	c	d	0	0
f	e	a	1	1
g	h	g	0	1
h	h	b	1	0

Ans: Reduction 1: $h=d$

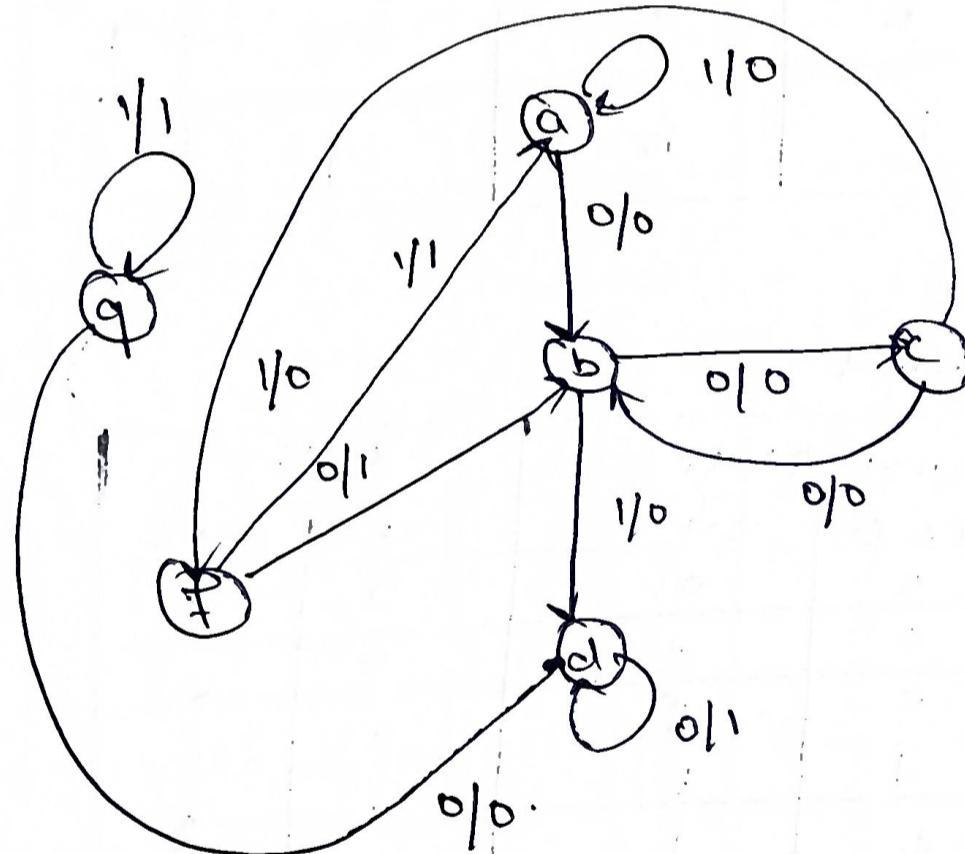
State Table after reduction 1

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

Reduction 2: $b=e$

State table after reduction 2:

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



State Assignment

- In order to design a sequential circuit with physical components, it is necessary to assign unique binary coded values to the states.
- For a circuit with m states, the codes must contain n bits, where $2^n \geq m$.

→ Common assignment methods include binary assignment, gray code, and the one-hot assignment.

One-hot configuration

- This configuration uses as many bits as there are states in the circuit.
- At any given time, only one bit is equal to 1, while all other bits are kept to 0.
- One-hot encoding usually leads to simpler decoding logic for the next state and the output.

Possible Binary State Assignments

State	Assignment 1	Assignment 2	Assignment 3
	Binary	Gray code	One-hot
a	0 0 0	0 0 0	0 0 0 0 1
b	0 0 1	0 0 1	0 0 0 1 0
c	0 1 0	0 1 1	0 0 1 0 0
d	0 1 1	0 1 0	0 1 0 0 0
e	1 0 0	1 1 0	1 0 0 0 0

Ex1 Assign binary states to the given reduced state table

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	d	0	1
e	a	d	0	1

Ans.

Present State	Next State		Output	
	$x=0$	$x=1$	$x=0$	$x=1$
000	000	001	0	0
001	010	011	0	0
010	000	011	0	0
011	100	011	0	1
100	000	011	0	1

Design Procedure

Steps

1. From the necessary specifications, derive a state diagram and state table.
2. Reduce the no. of states if necessary
3. Assign binary values to the states

4. Obtain the binary coded state table.

(27)

b. Choose the type of flip flop to be used.

6. Derive the simplified expressions for the flip flop input

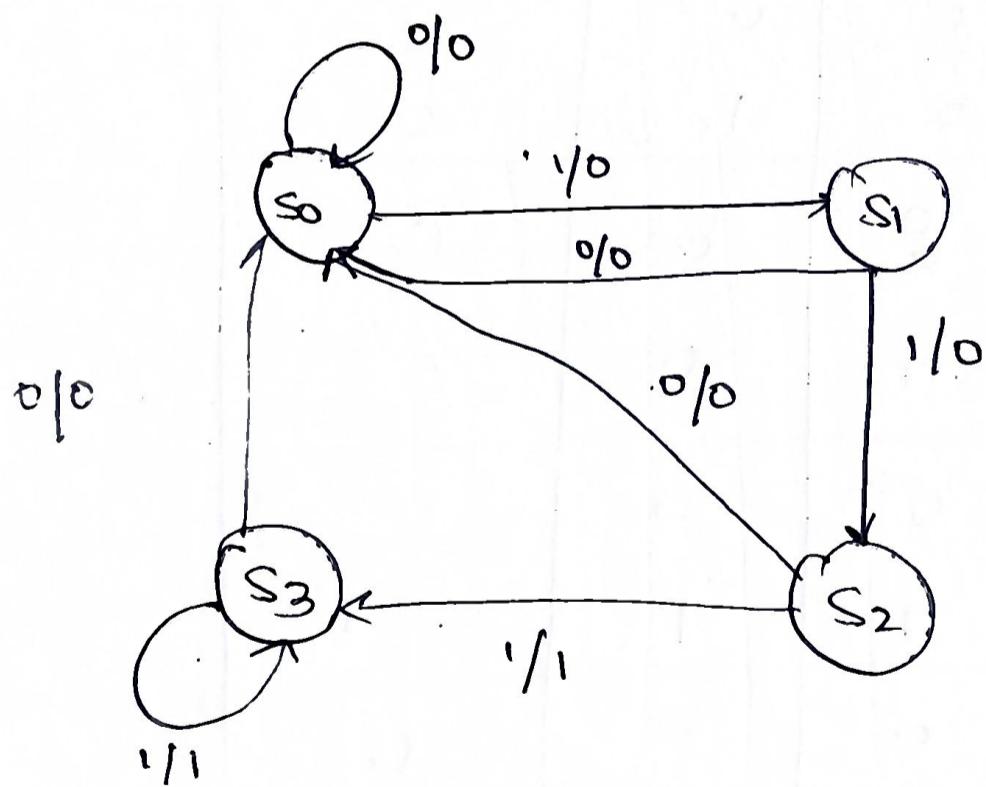
and output equations.

7. Draw the logic diagram.

① Design using D Flip Flops

Q: Design a circuit that detects a sequence of 3 or more consecutive 1's in a string of bits coming through an input line. (Sequence detector)

Step 1: State Diagram



Step 2: State Assignment

S₀ : 00

S₁ : 01

S₂ : 10

S₃ : 11

Step3: State Table

Present State	Next State Input		Output Next State
s_0	s_0	$x = 1$	00
s_1	s_0	$x = 0$	01
s_2	s_0	$x = 1$	00
s_3	s_0	$x = 0$	11

Step3: Transition Table

Present State	Input		Next State		Output
	$x = 0$	$x = 1$	$A(t+1)$	$B(t+1)$	
s_0	0	0	0	0	0
	0	1	0	1	0
s_1	0	1	0	0	0
	1	1	1	0	0
s_2	1	0	0	0	0
	1	1	1	1	1
s_3	1	1	0	0	0
	1	1	1	1	1

Step 4 : Minimized expressions for D_A and D_B
and the o/p

$$D_A = A(1+1)$$

$$D_B = B(1+1)$$

(i) For $D_A = \Sigma(3, 5, 7)$

		00	01	11	10
		0	0	1	0
A	B	0	1	3	2
0	0	0	1	1	0
1	0	4	5	7	6

$$\begin{array}{r} A \quad Bx \\ \hline 0 & 1 & 1 \\ 1 & 1 & 1 \\ \hline Bx \end{array} \qquad \begin{array}{r} ABx \\ \hline 1 & 0 & 1 \\ 1 & 1 & 1 \\ \hline Ax \end{array}$$

$$D_A = Ax + Bx$$

(ii) For D_B

		00	01	11	10
		0	1	0	0
A	B	0	1	3	2
0	0	0	1	1	0
1	0	4	5	7	6

$$\begin{array}{r} A \quad Bx \\ \hline 0 & 0 & 1 \\ 1 & 0 & 1 \\ \hline B'x \end{array} \qquad \begin{array}{r} A \quad Bx \\ \hline 1 & 0 & 1 \\ 1 & 1 & 1 \\ \hline Ax \end{array}$$

$$D_B = Ax + B'x$$

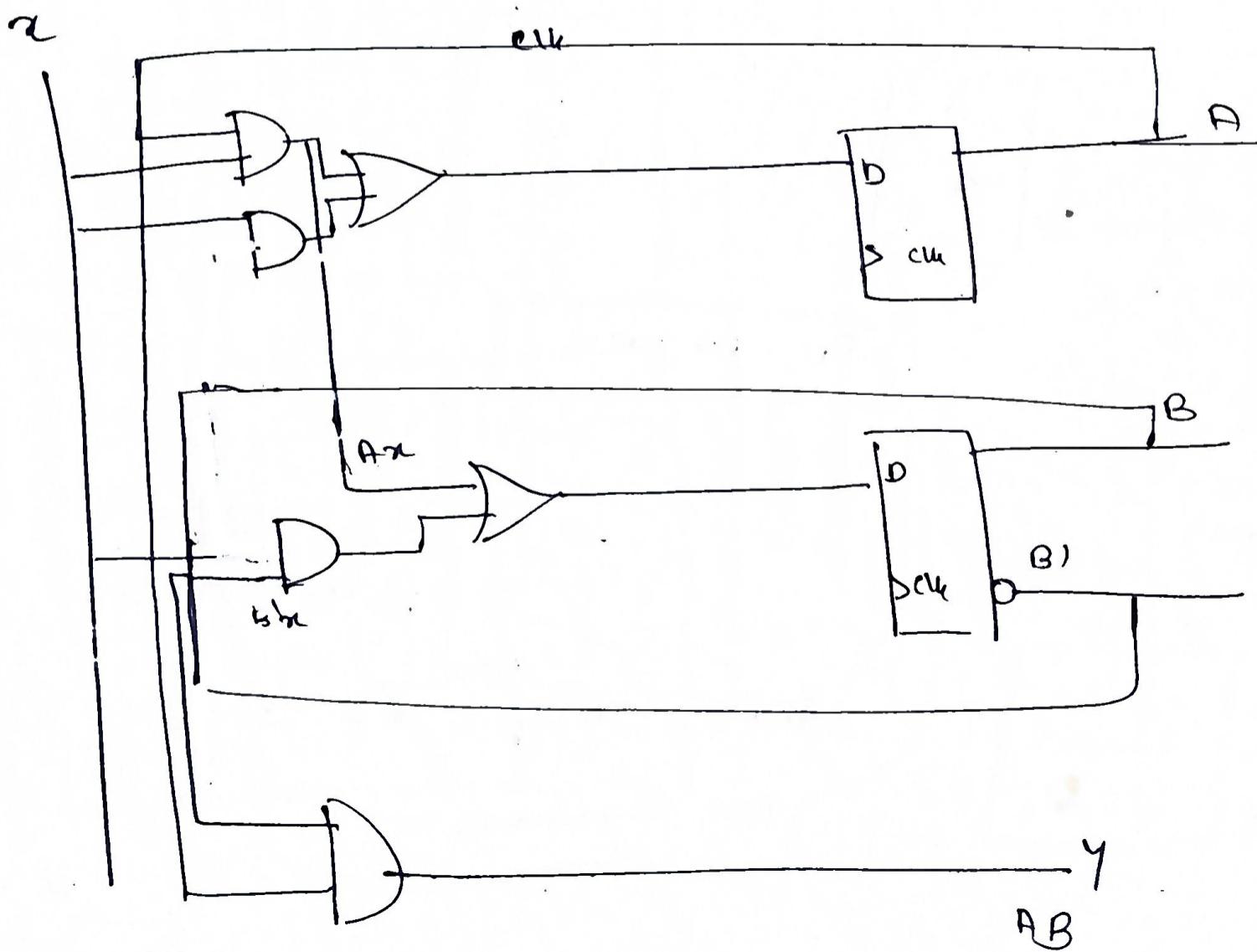
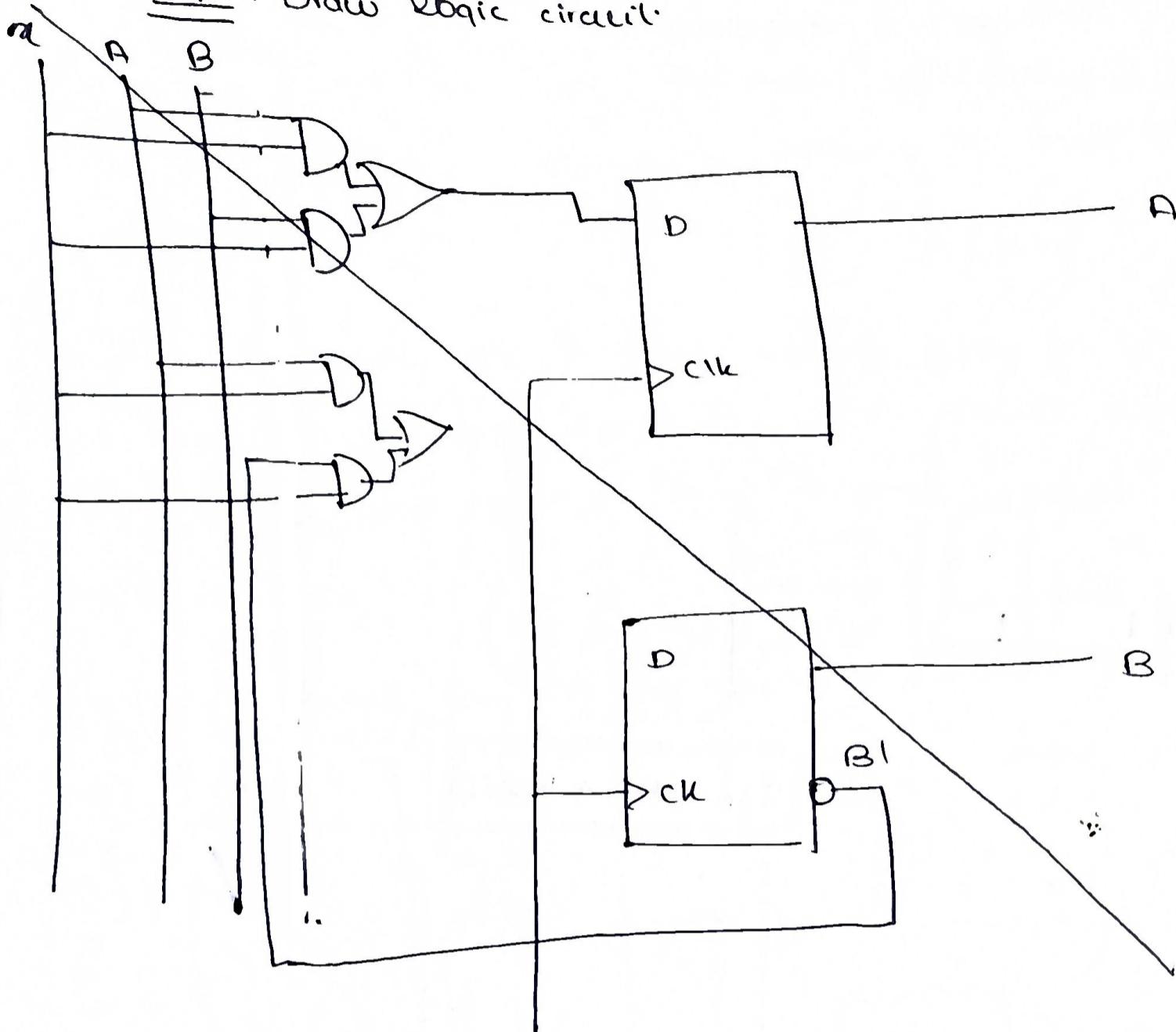
(iii) For the output

		00	01	11	10
		0	1	3	2
A	B	0	1	3	2
0	0	0	1	3	2
1	0	4	5	7	6

→ TB is correct

Note: $TB_{O/P} = AB$
last 2 columns are 1.

Step 5: Draw logic circuit.



Excitation Tables

- During design using JK and T flip-flops, it is necessary to derive a functional relationship between the state table and the input equations.
- When the transition from the present state to the next state is known, it is necessary to find out which inputs would give the required change. This kind of table is called an excitation table.

Excitation table for JK Flip-Flop

$Q(t)$	JK	$Q(t+1)$
0	0 0	0
0	0 1	0
0	1 0	1
0	1 1	1
1	0 0	1
1	0 1	0
1	1 0	0
1	1 1	0

Excitation Table for JK

$Q(t)$	$Q(t+1)$	JK
0	0	0 X
0	1	1 X
1	0	X 1
1	1	X 0

Grouping

$Q(t)$	$Q(t+1)$	JK
0	0	0 0
0	1	0 1
1	0	1 0
1	1	1 1

} 0X
 } 1X
 } X1
 } X0

Excitation Table for T Flip Flop

$Q(t)$	T	$Q(t+1)$
0	0	0
0	1	1
1	0	1
1	1	0

Reversing, Excitation Table for T FLIP

$Q(t)$	$Q(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

* Design using JK Flip Flops

Ex Design a circuit for the given state table

Present State		Input	Next State	
A	B	x	A	B
0	0	0	0	0
		1		1
0	1	0	1	0
		1		1
1	0	0	1	0
		1		1
1	1	0	1	1
		1		0

Step1: Finding the Flip Flop Inputs

Present State		Next State		FlipFlop Inputs			
A	B	A	B	JA	X _A	J _B	X _B
0	0	0	0	0	X	0	X
0	0	1	0	0	X	1	X
0	1	0	1	1	X	X	1
0	1	1	0	0	X	X	0
1	0	0	1	X	0	0	X
1	0	1	1	X	0	1	X
1	1	0	1	X	0	X	0
1	1	1	0	X	1	X	1

Excitation Table
 $Q(t) \rightarrow Q(t+1)$ J K

0	0	0	X
0	1	X	1
1	0	X	1
1	1	X	0

Step2: Deriving simplified equations for JA, X_A, J_B, and X_B

JA

		00	01	11	10
		A	Bx		
A		0	0	0	1
0					
1		X	X	X	X

$$\begin{array}{r}
 A \ B \ x \\
 0 \ 1 \ 0 \\
 x \ 1 \ 0 \\
 \hline
 Bx' \\
 \end{array}$$

$JA = Bx'$

JB

		00	01	11	10
		A	Bx		
A		0	1	X	X
0					
1		0	1	X	X

$$\begin{array}{r}
 A \ B \ x \\
 0 \ 0 \ 1 \\
 0 \ 1 \ 1 \\
 1 \ 0 \ 1 \\
 1 \ 1 \ 1 \\
 \hline
 x \\
 \end{array}$$

$JB = x$

X_A

		00	01	11	10
		A	Bx		
A		0	X	X	X
0					
1		0	0	1	0

$$\begin{array}{r}
 A \ B \ x \\
 0 \ 1 \ 1 \\
 1 \ 1 \ 1 \\
 \hline
 Bx \\
 \end{array}$$

$X_A = Bx$

X_B

	Bx	00	01	11	10
0	x	x	0	1	
1	x	x	1	0	

$$\begin{array}{r}
 ABx \\
 000 \\
 010 \\
 \hline
 A'x' \\
 \end{array}
 \quad
 \begin{array}{r}
 ABx \\
 101 \\
 111 \\
 \hline
 Ax \\
 \end{array}$$

$$X_B = A'x' + Ax$$

$$X_B = A \odot x$$

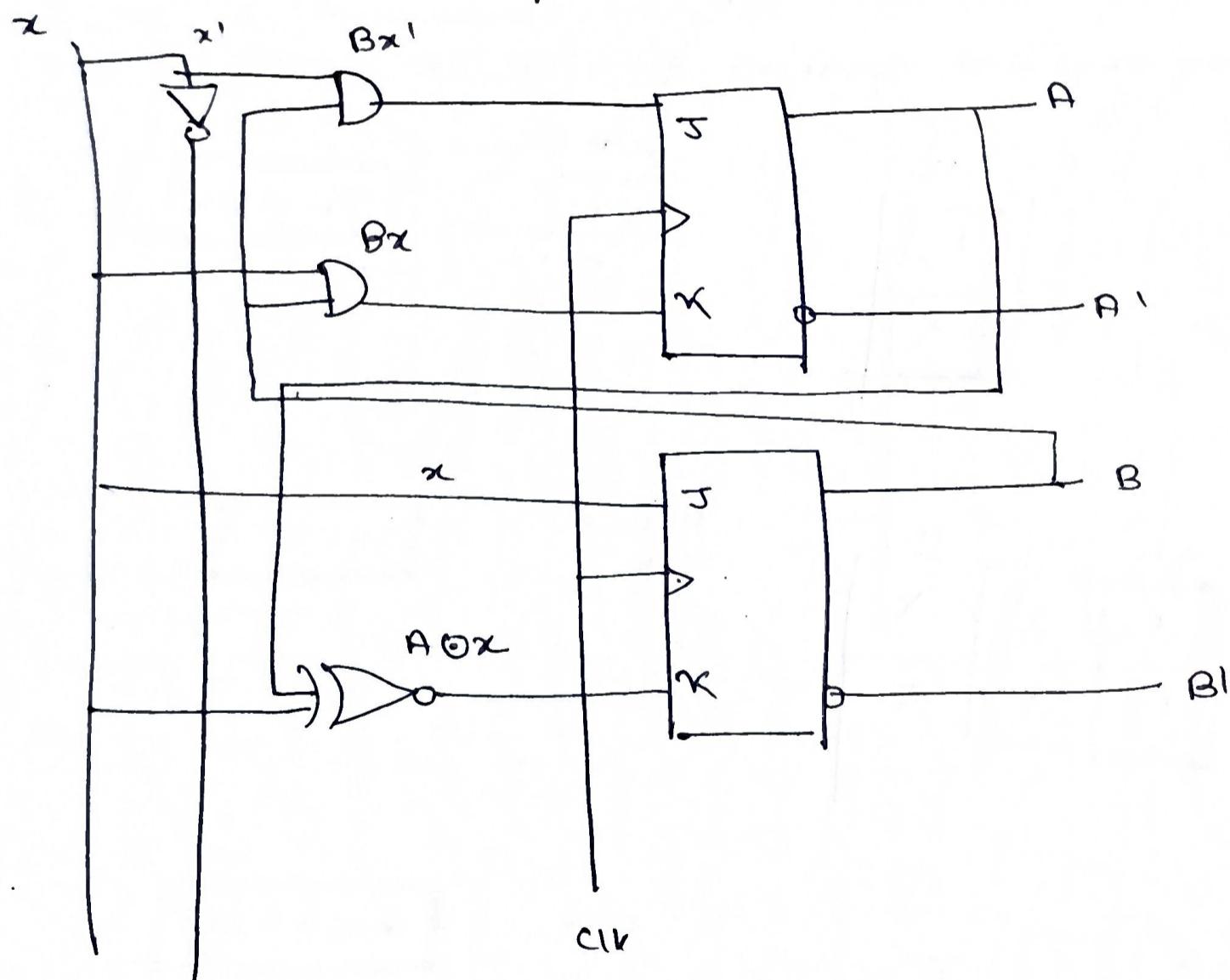
Step 3: Drawing the corresponding logic circuit

$$J_A = Bx'$$

$$J_B = x$$

$$K_A = Bx$$

$$K_B = A \odot x$$



*Advantage of using JK flip flops for design

→ Excitation table consists of many don't care entries.

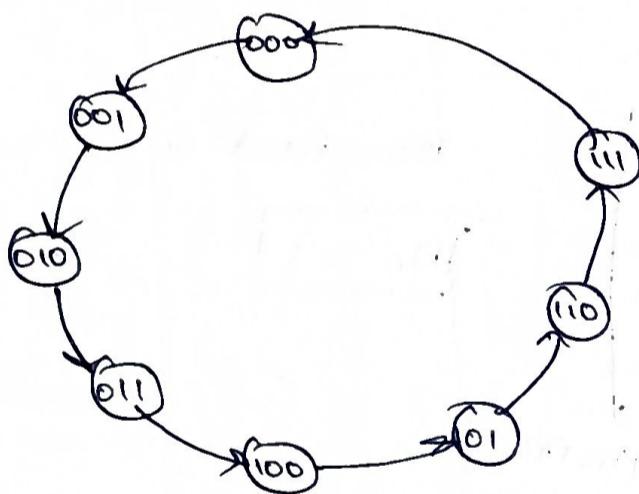
→ The combinational circuit is likely to be simpler, because of those don't cares.

*Design using T FlipFlops

Ex: Design a 3-bit binary counter using T flip flops (Mod 8 counter)

(An n bit binary counter consists of n flipflops that can count in binary from 0 to $2^n - 1$.)

Ans Step1: State Diagram



Excitation Table.

$Q(t)$	$Q(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

Step2: State Table

Present State $A_2\ A_1\ A_0$	Input			Next State $A_2\ A_1\ A_0$	Flip Flop Inputs		
	I_{A_2}	I_{A_1}	I_{A_0}		TA_2	TA_1	TA_0
0 0 0	0	0	1	0 0 1	0	0	1
0 0 1	0	1	0	0 1 0	0	1	1
0 1 0	0	1	1	0 1 1	0	0	1
0 1 1	1	0	0	1 0 0	1	1	1
1 0 0	1	0	1	1 0 1	0	0	1
1 0 1	1	1	0	1 1 0	0	1	1
1 1 0	1	1	1	0 1 1	0	0	1
1 1 1	0	0	0	0 0 0	1	1	1

Step 3: Obtaining simplified input equations

(i) $T A_2$

A_2	A_1, A_0	00	01	11	10
0	0	0	1	1	0
1	0	0	1	1	0

$$\begin{array}{r} A_2 \ A_1 \ A_0 \\ 0 \quad 1 \quad 1 \\ 1 \quad 1 \quad 1 \\ \hline A_1, A_0 \end{array}$$

$$T A_2 = A_1, A_0$$

(ii) $T A_1$

A_2	A_1, A_0	00	01	11	10
0	0	1	1	1	0
1	0	1	1	1	0

$$\begin{array}{r} A_2 \ A_1 \ A_0 \\ 0 \quad 0 \quad 1 \\ 0 \quad 1 \quad 1 \\ 1 \quad 0 \quad 1 \\ 1 \quad 1 \quad 1 \\ \hline A_0 \end{array}$$

$$T A_1 = A_0$$

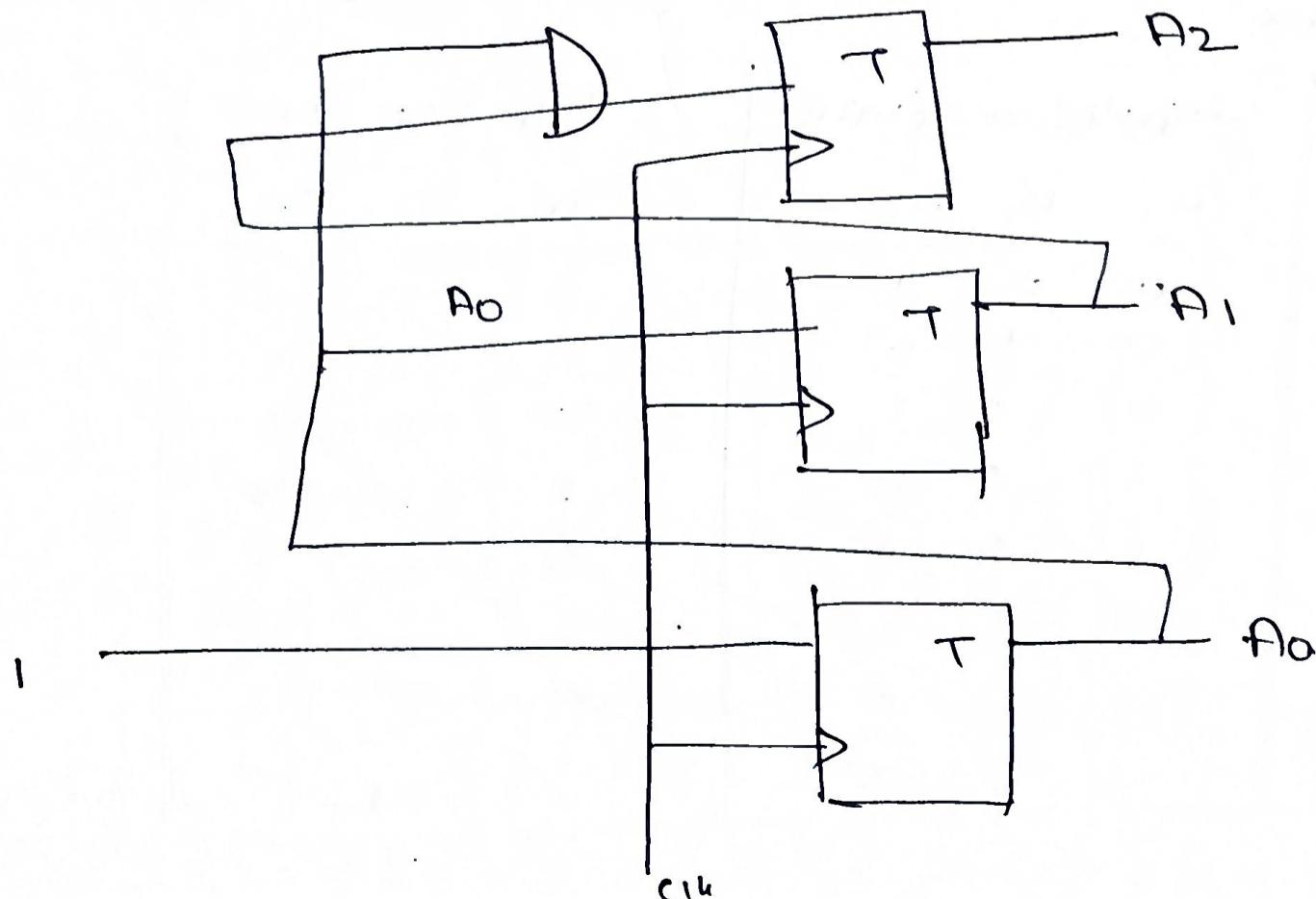
(iii) $T A_0$

A_2	A_1, A_0	00	01	11	10
0	1	1	1	1	1
1	1	1	1	1	1

$$\begin{array}{r} A_2 \ A_1 \ A_0 \\ 1 \quad 1 \quad 1 \\ \hline A_0 \end{array}$$

$$T A_0 = 1$$

Step 4: Drawing the circuit diagram



Registers and Counters

Counters

- A counter is a register that goes through a predetermined sequence of binary states.
- The gates in a counter are connected in such a way so as to produce the prescribed sequence of states.
- The input pulses may be clock pulses, or may originate from some external source and may occur at fixed intervals of time, or at random.
- Counters are of 2 categories: (i) ripple counters
 (ii) synchronous counters

Ripple Counters : a flip-flop output transition serves as a source for triggering other flip-flops.

Synchronous Counters : In synchronous counters, the clock inputs of all the flip-flops receive the common clock.

① Binary Ripple Counter

- consists of a series connection of complementing flip flops.
- The flip flop with the LSB receives the incoming clock pulses.

→ The complementing flip flops can be obtained from a JK flip-flop tied together, or with a T flip-flop.

Alternatively, a D flip flop can be used with the complement output connected to the D input.

Working

- The LSB A_0 receives the clock pulse.
- A_0 is complemented with each count pulse input.
- Every time A_0 goes from 1 to 0, it complements A_1 .
- Every time A_1 goes from 1 to 0, it complements A_2 .
- This process follows for higher order bits as well.

Construction

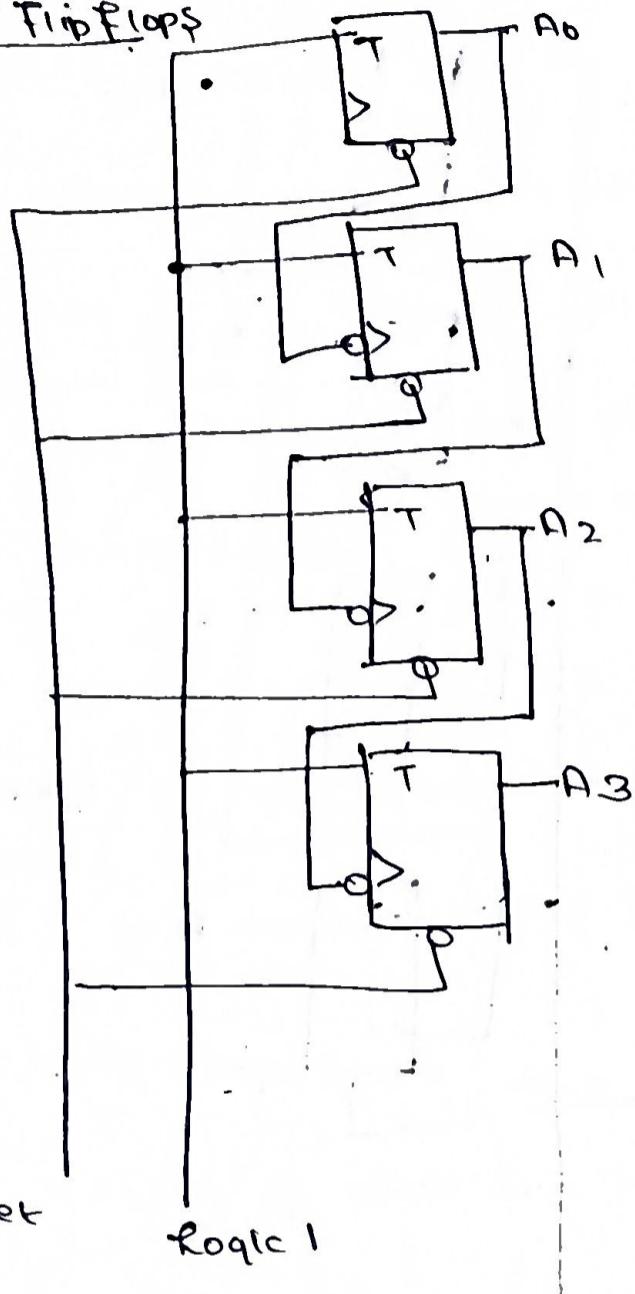
A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Binary Ripple-Up Counter

with a T Flip Flop

- T inputs are connected to a permanent logic 1.
- The bubble in front of the dynamic indicator symbols shows that the flip flop responds to the negative edge transition of the input.

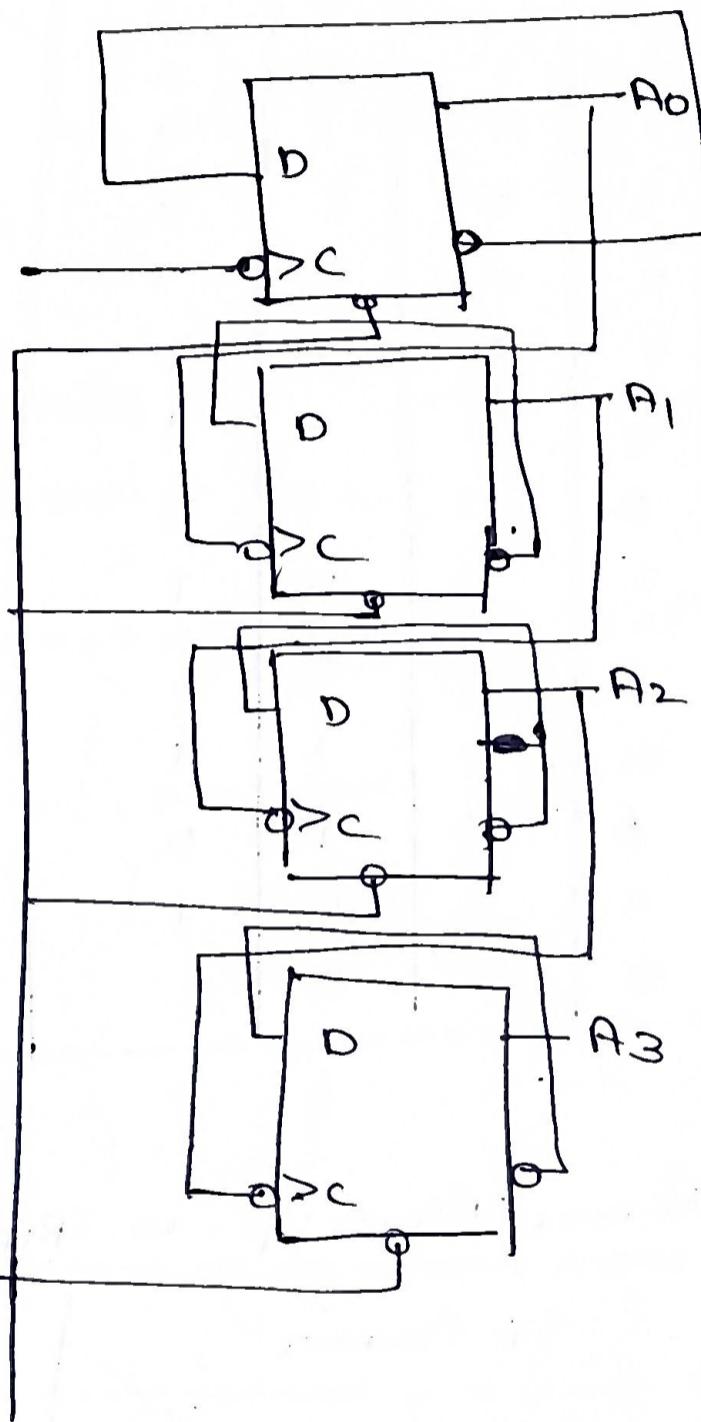
(i) with T flip-flops



(ii) with D flip-flops

→ The complement output is connected to the D input.

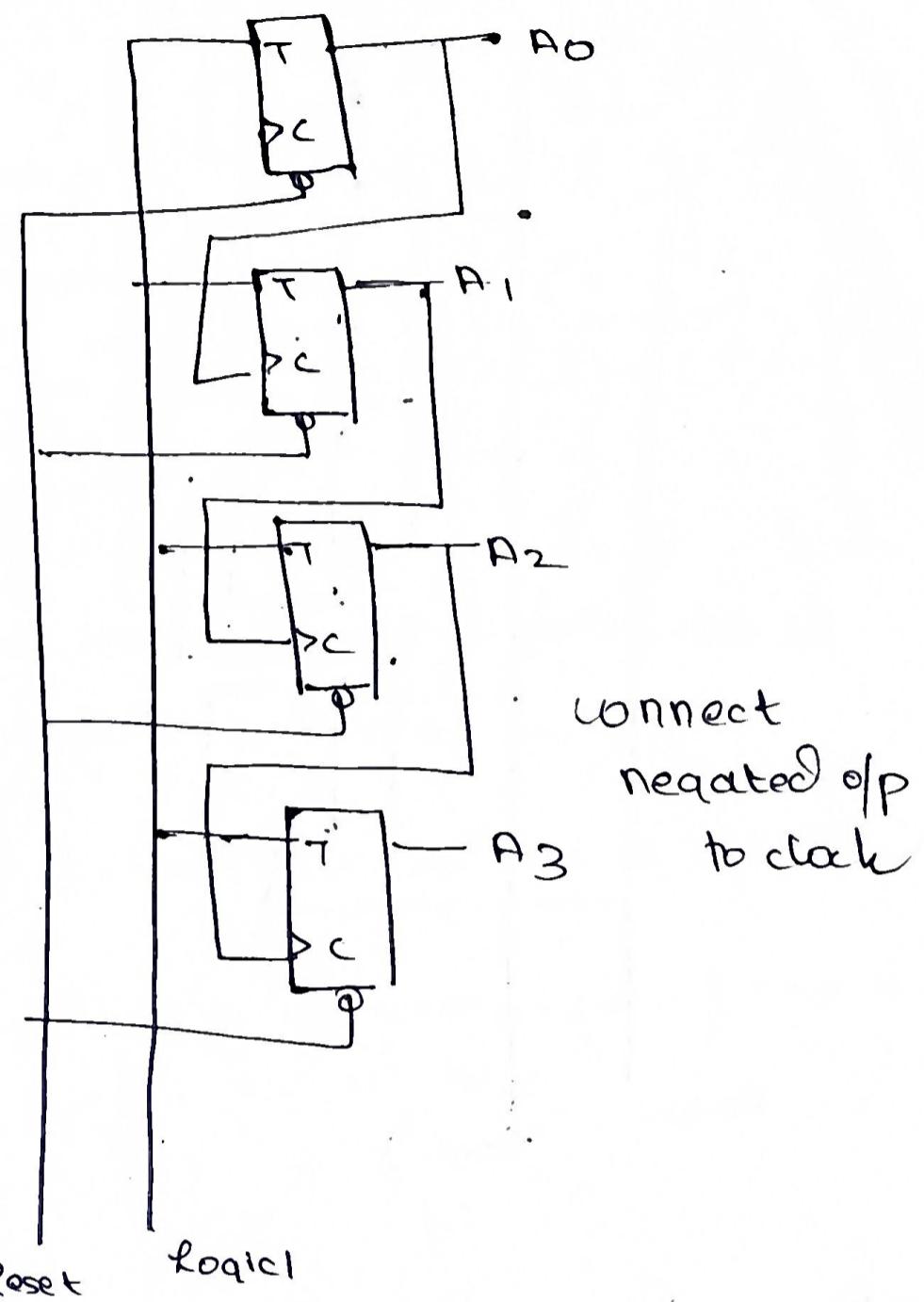
41



* Binary Ripple-Down Counter

- A binary counter with a reverse count is called a binary countdown counter.
- The counter starts from binary 15, and continues to binary 0, and back to 15.
- The LSB is complemented with each clock pulse.
- There is a change in other bits if the LSB goes from 0-1.
- This means that the binary ripple down counter is positive-edge triggered.

A_3	A_2	A_1	A_0
1	1	1	1
1	1	1	0
1	1	0	1
1	0	0	0
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	0
0	0	0	1
0	0	0	0



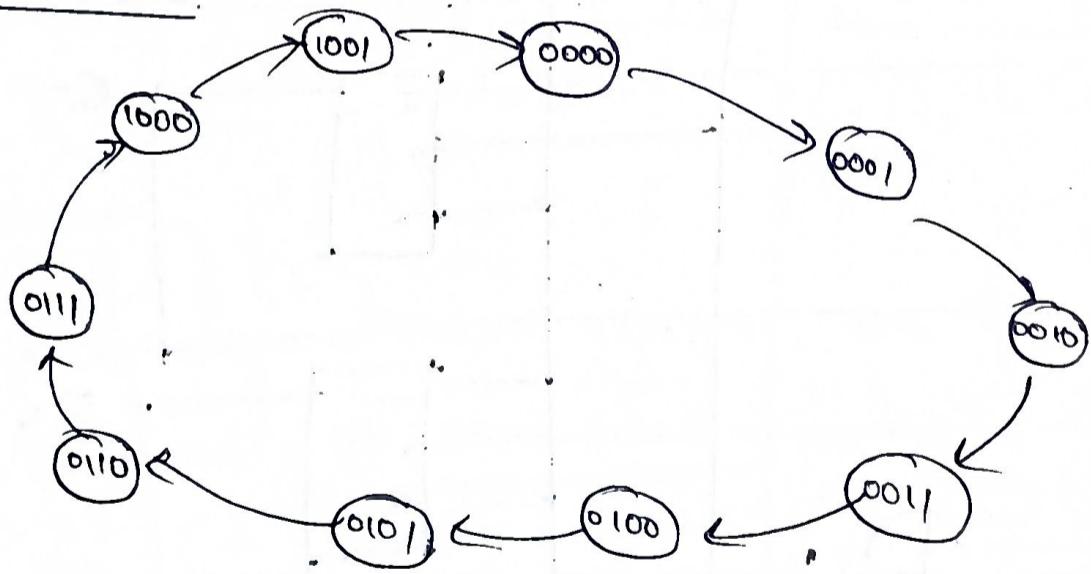
Binary Ripple Up vs. Ripple Down Counter

Up Counter	Down Counter
(i) 4 bit counter counts from 0-15	(i) 4 bit counter counts from 15-0
(ii) Flip flop is triggered when moving from 0 to 1.	(ii) Flip flop is triggered when moving from 1 to 0.
(iii) Use a -ve triggered FF	(iii) Use a +ve triggered FF
(iv) $clk = 0$	(iv) $clk = 1$

* BCD Ripple Counter

- A decimal counter follows a sequence of 10 states and returns to 0 after the count of 9.
- The counter must have at least four flip-flops to represent each decimal digit.
- A BCD ripple counter is an asynchronous circuit, and its state changes are not synchronized to a common clock.
- A BCD ripple counter can be implemented using JK flipflops
- The J & K inputs are connected to a permanent 1 signal or to the output of other flip flops.

State Diagram



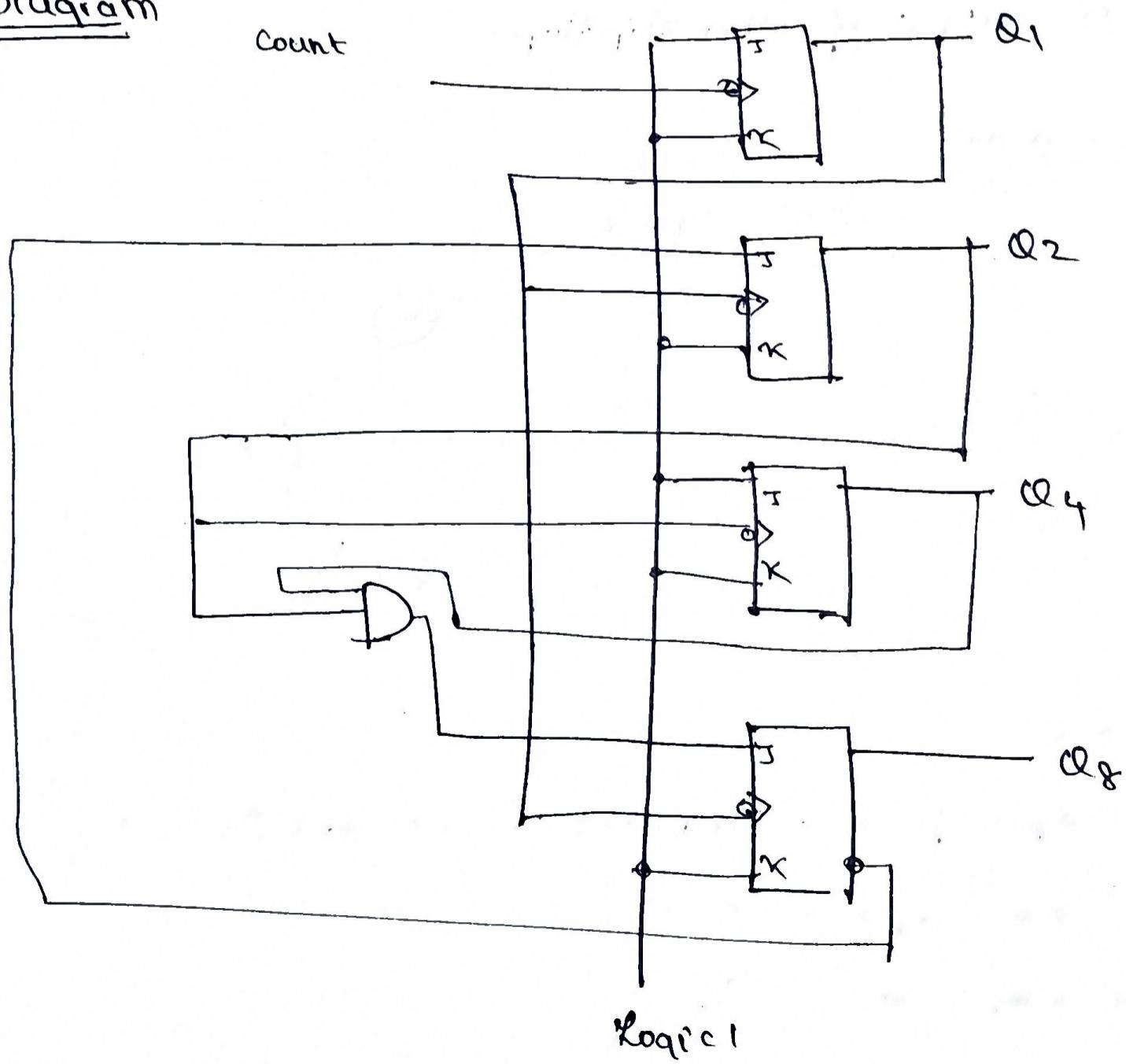
Construction

- The output of Q_1 is connected to Q_2 and Q_8 . (c)
- Q_8 's ~~JK~~ T input is $Q_2 Q_4$
- The K inputs are all 1.

State Table

Present State	Next State	
Q ₈ Q ₄ Q ₂ Q ₁ 0 0 0 0	Q ₈ Q ₄ Q ₂ Q ₁ 0 0 0 1	Q ₁ toggles at every clock
0 0 0 1	0 0 1 0	Q ₂ toggles when Q ₁ moves from 1 to 0, except when $Q_8 = 1$
0 0 1 0	0 0 1 1	$C_{LK2} = Q_1$
0 0 1 1	0 1 0 0	Q ₄ toggles when Q ₂ moves from 1 to 0
0 1 0 0	0 1 0 1	$C_{LK4} = Q_2$
0 1 0 1	0 1 1 0	
0 1 1 0	0 1 1 1	
0 1 1 1	1 0 0 0	Q ₈ toggles when Q ₁ moves from 1 to 0
1 0 0 0	1 0 0 1	$C_{LK8} = Q_1$
1 0 0 1	0 0 0 0	

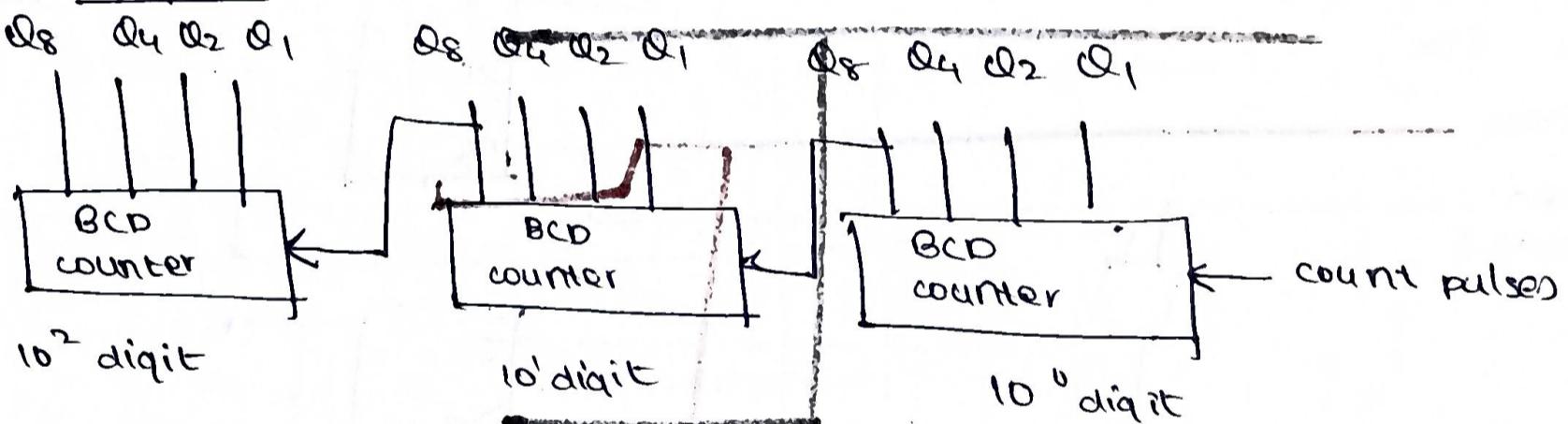
Logic Diagram



Decade Counters

- The BCD counter is a decade counter, since it counts from 0 to 9.
- To count in decimal from 0 to 99, we need a 2 decade counter, and from 0 - 999, a 3-decade counter.
- Multiple decade counters can be constructed by connecting BCD counters in cascade, one for each decade.
- The inputs to the second and third decades come from Qs of the previous decade.

Block Diagram

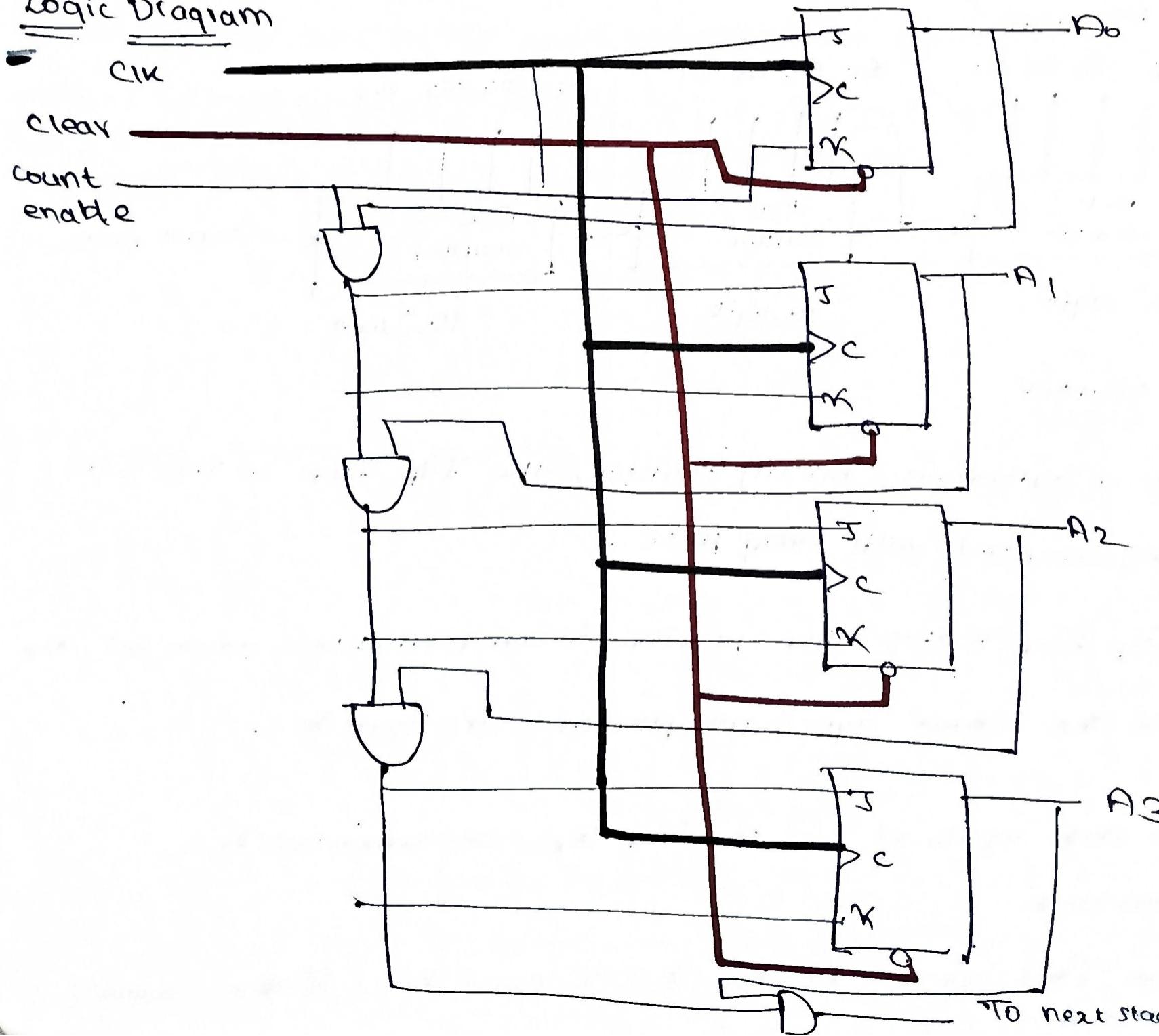


Binary Counter

- In a synchronous binary counter, the flip flop in the MSB is complemented with every pulse.
- A flip flop in any other position is complemented, when all the bits in the lower significant positions are equal to 1.
- The clock inputs of ~~all the flip flops~~ are connected to a common clock.
- The first stage A_0 , has $J = K = 1$, if the counter is enabled.

- The other J and K inputs are equal to 1 only if all of the previous least significant stages are equal to 1 and the count is enabled.
- The chain of AND gates generates the required logic for the J & K inputs in each stage.
- The counter can be extended to any no. of stages ~~only if~~, with each stage having an additional flip-flop and an AND gate that gives an output of 1 only if all the previous flip flop outputs are 1.

Logic Diagram



Binary Up-down counter

- In a binary up-down counter, the states move from 0000 to 1111 or vice versa.
- The bit in the least significant position is complemented with each pulse, during downward counting.
- Any other bit is complemented if all lower significant bits are equal to zero.
- A single circuit can implement both up and down counting operations, using T flip-flops.
- It has an up control input and a down control input.
- The following set up ensures for up & down counting

Up	Down	Operator
1	0	Toggle
0	1	Down
0	0	No change
1	1	Up

- Priority is given to the Up input over the down input.

- Simplify the flip flop input equations

$$TQ_1 = 1$$

$$TQ_2 = Q_8'Q_1$$

$$TQ_4 = Q_2Q_1$$

$$TQ_8 = Q_8Q_1 + Q_4Q_2Q_1$$

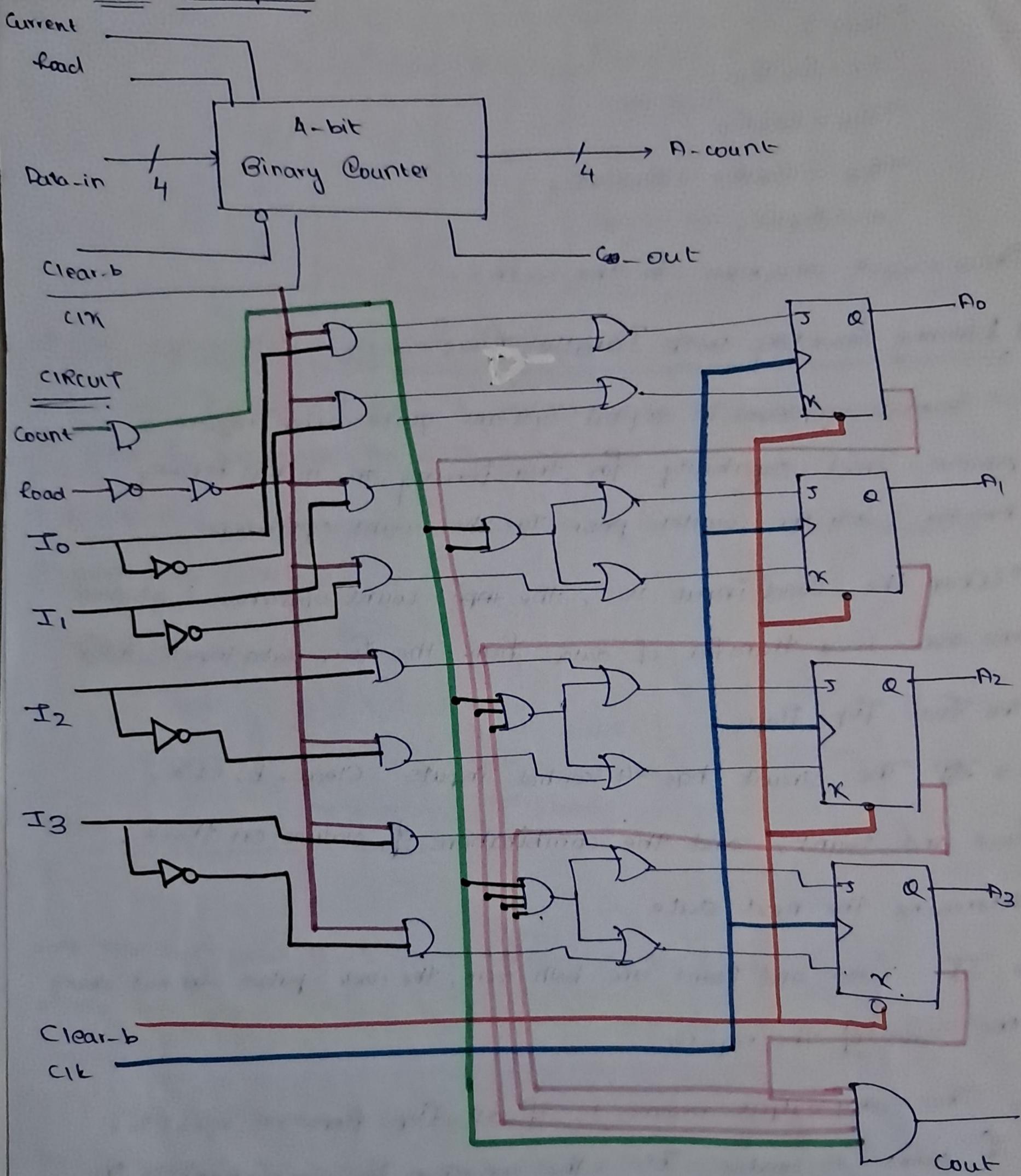
$$q = Q_5Q_1$$

Draw circuit diagram for the same.

* Binary Counters with Parallel Load

- Counters employed in digital systems quite often require a parallel-load capability for transferring an initial binary number into the counter prior to the count operation.
- When the load input is 1, the input count operation is disabled and there is a transfer of data from the four data inputs into the four flip-flops.
- ~~If~~ The circuit has 4 control inputs Clear - b, CLK, Load and Count, and the combination of values on these determine the next state.
- If Load and Count are both zero, the clock pulses do not change the state of the register.
- The carry output becomes 1, if all flip-flops are equal to 1 if count is enable. This is the condition for complementing the flipflop.
- The carry output is useful for expanding the counter to more than 4 bits..

Block Diagram



Function Table

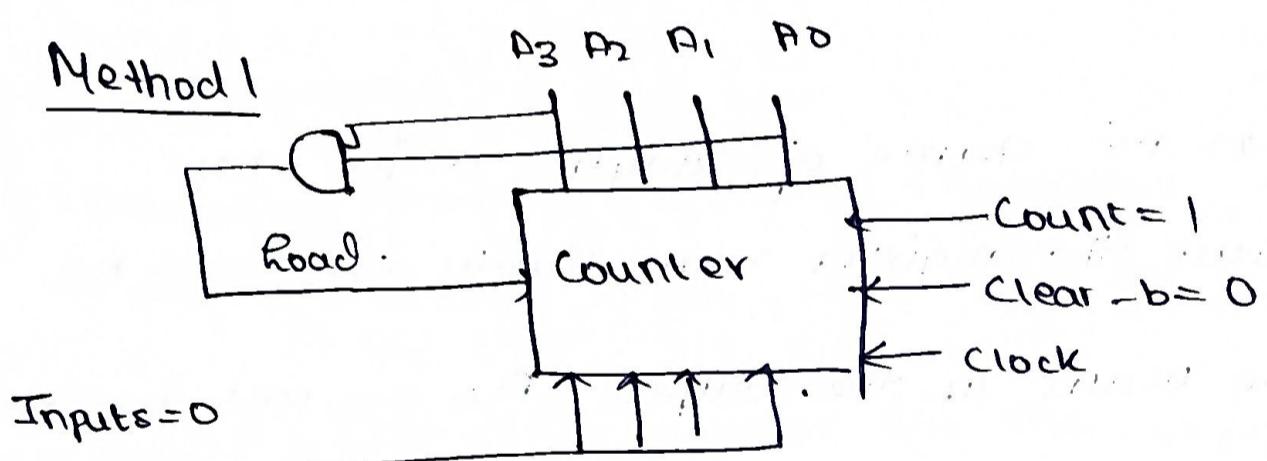
Clear-b	CK	Load	Count	Function
0	X	X	X	Clear to 0
1	↑	1	X	Load Inputs,
1	↑	0	1	Count next binary state
1	↑	0	0	No change.

* BCD counter with a parallel Load

→ There are 2 ways in which a counter with a parallel load can be used to generate the BCD count.

→ In each case, the count control is set to 1 to enable the count through the CK input.

Method 1



→ The AND gate detects the occurrence of the state 1001.

→ At this stage, 'Load' is asserted, and 0's are loaded into the register.

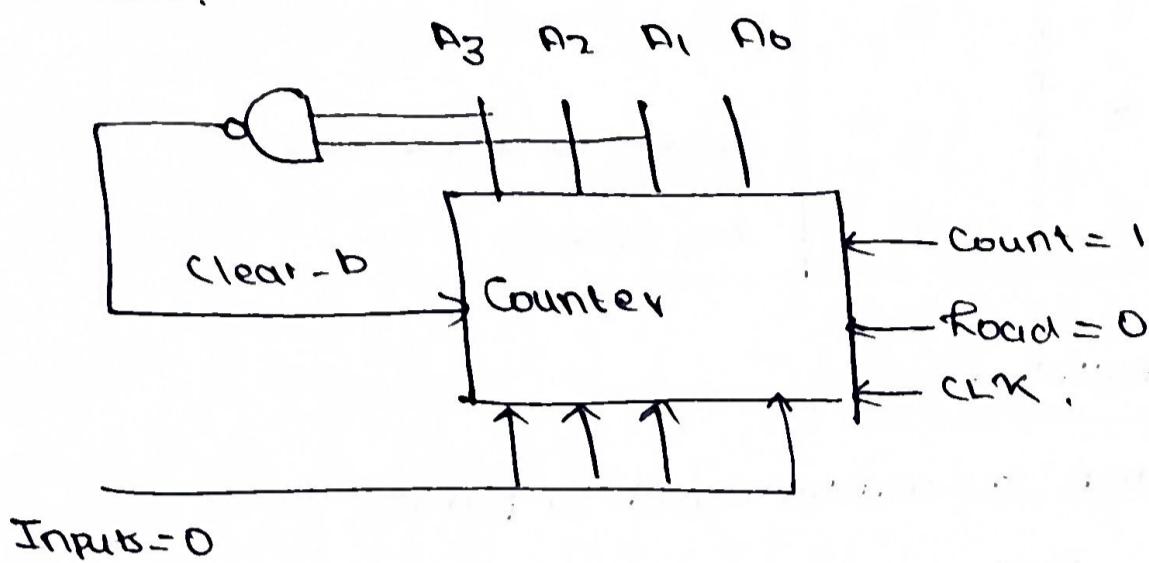
→ As long as the output of the AND gate is 0, each positive edge clock increments the counter by 1.

→ When the o/p becomes 1001, A₃ and A₀, the o/p of the AND gate becomes 1.

→ For the next edge, rather than counting further, 0's are loaded into

→ Thus, the circuit goes through 0000 to 1001 and back to 0000.

Method 2:



→ In this scenario, the NAND gate detects the count of 1010, but as soon as this count occurs, the register is cleared, asynchronously.

→ The count 1010 has no chance of staying on for any appreciable time, because the register immediately goes back to 0000.

→ A momentary spike occurs in the output A₀ as 1010 transitions to 1011, and then immediately to 0000.

→ This spike may be undesirable, and for that reason, this configuration is not recommended.

→ If the counter has a synchronous clear input, it is possible to clear the counter with the clock, after the occurrence of the 1001 count.

* Counters with Unused States

53

- A circuit with n flip flops has 2^n binary states, but there are occasions when a sequential circuit uses fewer than this maximum possible no. of states.
- The unused states may be considered as don't-care conditions.
- It is important to note that once a circuit is designed and constructed, outside interference during its operation may cause the circuit to enter one of the unused states.
- It is thus necessary to ensure that the circuit eventually goes back into one of its valid states, so that it can resume normal operation.
- Otherwise, there is a chance that the circuit circulates among the unused states, with no way of bringing it back.

Example: Design a counter for the sequence 0, 1, 2, 4, 5, 6 using a JK flip-flop. Analyze the next states of the unused states.

Present State			Next State			Flip Flop Inputs					
A	B	C	A	B	C	J _A	K _A	J _B	K _B	J _C	K _C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	0	0	1	X	1	X	X	0
1	0	0	1	0	0	1	X	X	1	0	X
1	0	1	1	0	1	X	0	0	X	1	X
1	1	0	0	0	0	X	1	X	X	1	0

Simplify the inputs of the JK flipflops

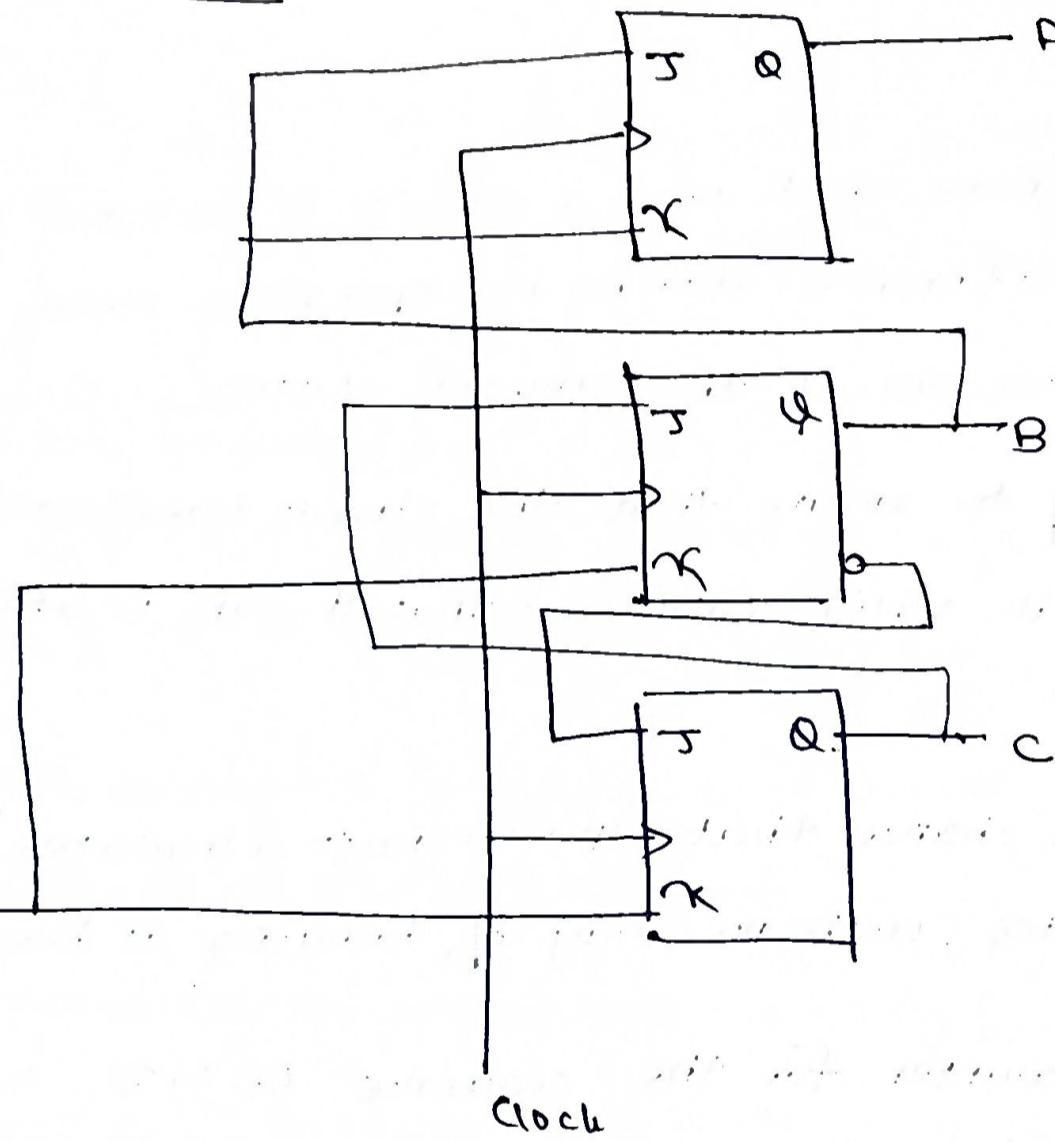
Simplifying,

$$J_A = B \sim \quad R_D = B \sim$$

$$J_B = C \sim \quad K_B = 1 \sim$$

$$J_C = B^1 \quad K_C = 1 \checkmark$$

Circuit Diagram

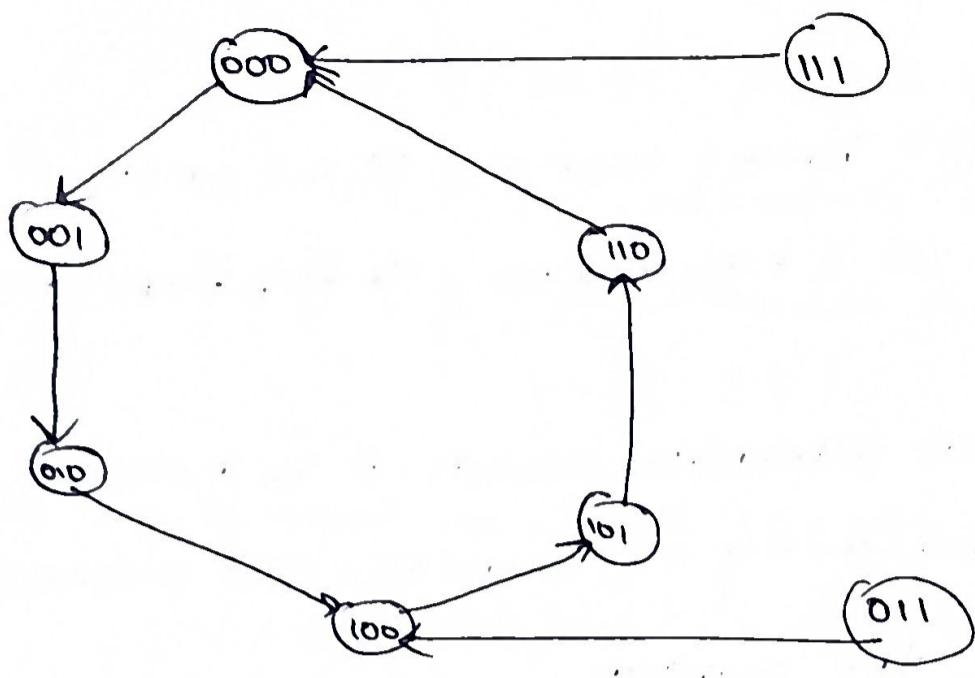


Analysis of unused states

If the counter happens to be at 011 due to an error signal, after a clock pulse, it would go to 100.

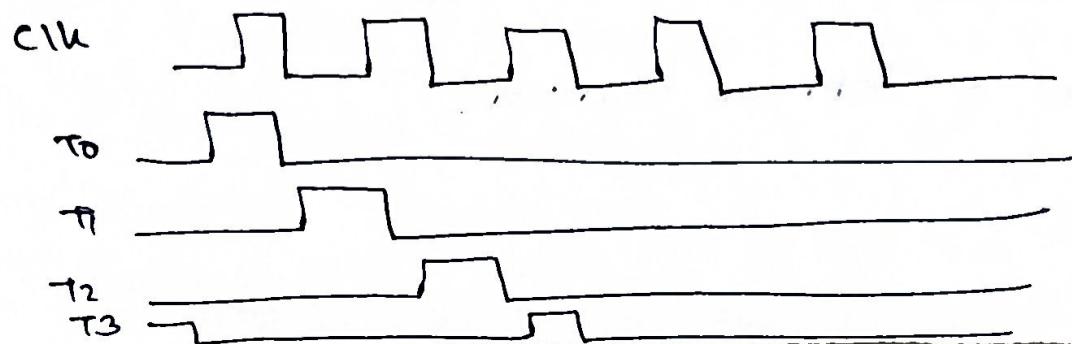
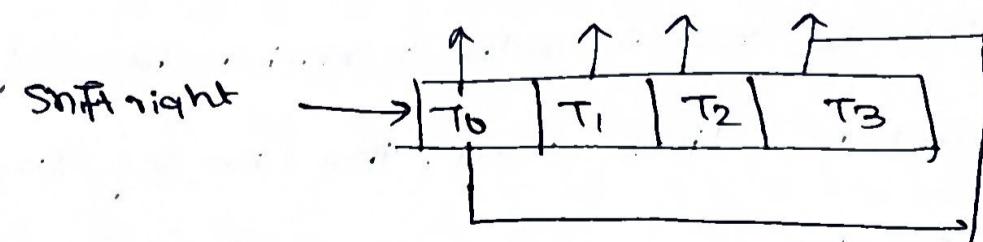
Similarly, the next state of 111 would be 000

→ Thus, if the circuit ever goes to any of the unused states because of outside interference, the next clock pulse transfers it to one of the valid states, and the counter counts correctly
→ Thus, the counter is self-correcting.



* Ring Counters

- Timing signals can be generated using a shift register, or using a counter with a decoder.
 - A ring counter is a circular shift register with only one flip flop being set at an any particular time.
 - A single bit is shifted from one flip flop to the next.
- Example : Consider an 8-4-2-1 ring counter.
- The initial value of the register is 1000.
 - The single bit is shifted right with every clock pulse, and circulates from T_3 to T_0 .



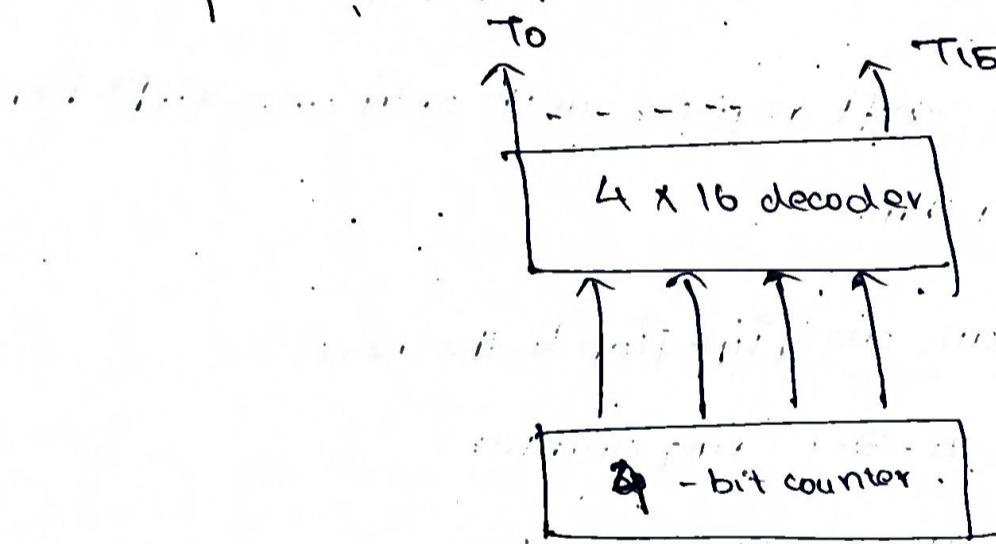
→ However, to generate 2^n timing signals, 2^n flip flops would be required.

→ For ex. To make 2^4 timing signals, 16 bit shift registers can be connected as a ring counter., 16 flip flops are needed.

To counter this problem, an alternate design is to have timing signals generated by using a counter, and a decoder to decode each state of the counter.

∴ To make 2^4 timing signals, a 4 bit counter can be used along with a 4×16 decoder.

Only 4 flip flops would be needed in this case



Johnson Counter

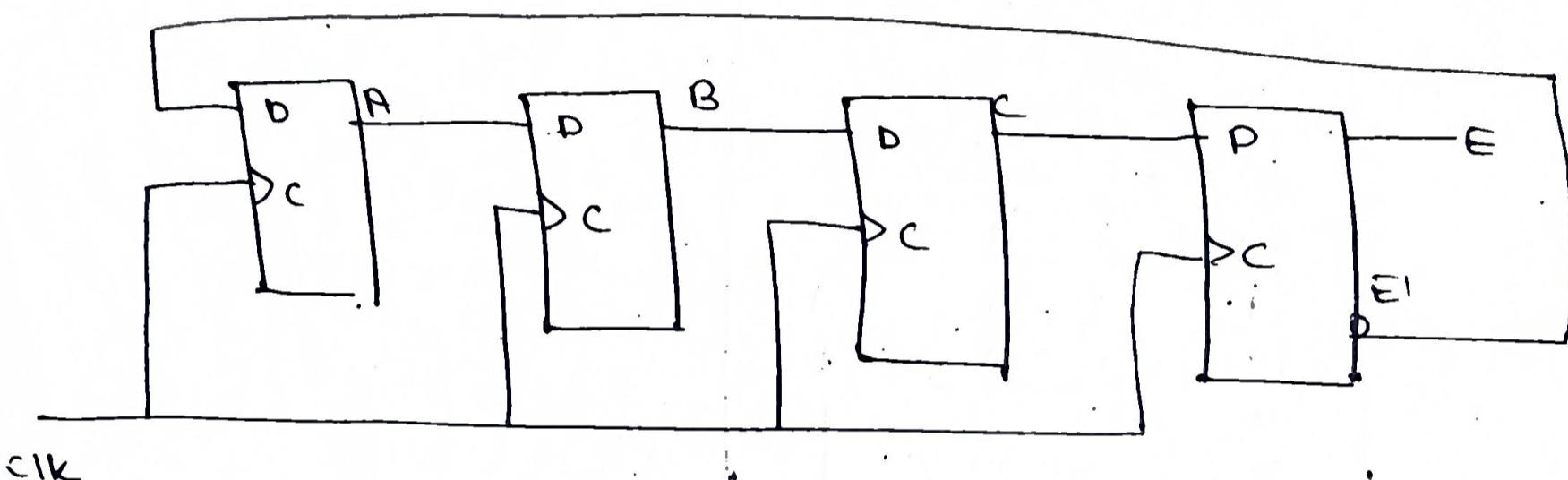
→ Timing signals can also be made with a combination of a shift register and a decoder. This way, the no. of flip flops is lesser than in a ring counter.

→ Such a counter is called a Johnson counter.

Switch Tail Ring Counter

57.

- The no. of states in a k -bit ring counter can be doubled, if the complemented output of the last flip flop's complemented output is connected to the input of the first flip flop.
- This is called a switch-tail ring counter.



- The shift register shifts its contents to the right with every clock pulse, and at the same time the complemented value of the E flip flop is transferred to the A flip flop.

Johnson Counters with a switch tail ring counter and decoder

- It has a k -bit switch tail ring counter.
- There are 2^k decoding gates to provide outputs for 2^k timing signals.

Method of decoding

- An all 0 state → complement the extreme flip-flop outputs.
- An all 1 state → take the normal QD of the 2 extreme flip-flops.

→ all other states decoded using an adjacent 1,0 or 0,1 pattern.

Count Sequence and Decoding

Sequence No

Flip flop outputs

A B C · E

AND Gate required for output

1	0	0	0	0
2	1	0	0	0
3	1	1	0	0
4	0	1	1	0
5	1	1	1	0
6	0	1	1	1
7	0	0	1	1
8	0	0	1	0

$A'E'$

AB'

BC'

CE'

AE

$A'B$

$B'C$

$C'E$

Note : To fill flip-flop o/p's complement last bit, make it the first bit, push the first 3 bits to the right

Disadvantages

→ If the circuit goes into an unused state, it will continually move from one invalid state to another, and never find its way to a valid state.

Correcting Procedure : → disconnect o/p from flip flop B going to C enable input of flip flop C as $D_C \cdot (A + C)B$

∴ Johnson counters can be constructed for any no. of timing sequences

No. of flip flops needed = $1/2$ no. of timing signals.

No. of decoding gates = no. of timing signals,
only 2 if p gates are needed.