UNIT-IV

SEQUENTIAL LOGIC

Sequential circuits, latches, flip-flops Analysis of clocked sequential circuits, State Reduction and assignment, Design procedure, Registers, shift registers, Ripple counters, Synchronous counters, other counters.

* Introduction: *

→ Digital Electronics is classified in to combinational logic & Sequential logic.

→ combinational logic output depends on the present input levels, whereas sequential logic output depends on stored levels and also the input levels. (i.e present input and past outputs)

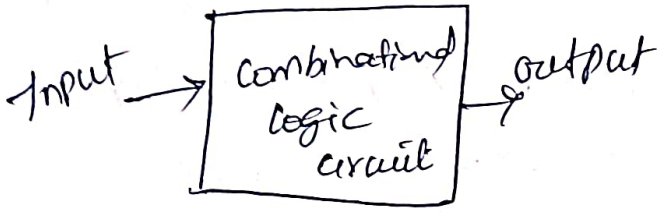
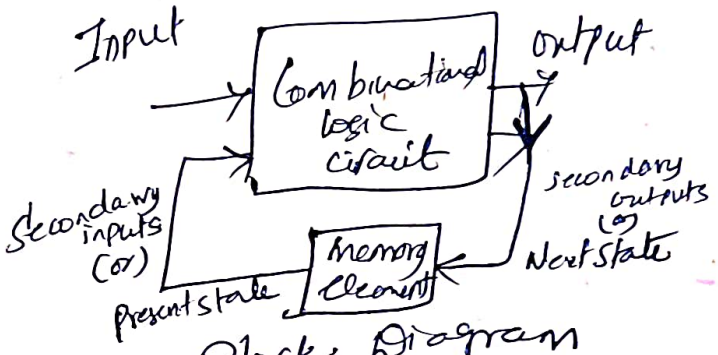
* Basic Architectural difference btw combinational and logic circuits *

→ The memory elements are devices capable of storing binary information. The Binary Information stored in the memory elements at any given time defines the state of the sequential circuits.

→ The input and the present state of the memory element determines the output. memory elements next state is also a function of external inputs and present state.

→ A sequential circuit is a specified by a time sequence of inputs, outputs, and internal states.

Differences Between Combinational and Sequential logic circuits :-

① Combinational logic circuit	Sequential logic circuit
 Block diagram	 Block Diagram
② In combinational circuits, the output variables are at all times dependent on the combination of input variables	② In sequential circuits, the output variables depend ~~only~~ on the present input variables but they also depend upon the past history of these input variables
③ memory unit is not required in combinational circuits	③ memory unit is required to store the past
④ Combinational circuits are faster ~~than~~ in speed because the delay between input and output is due to propagation delay of gates.	④ Sequential circuits are ~~slowly~~ slower than the combinational circuits.
⑤ Easy to design	⑤ Harder to design.
⑥ There is no feedback	⑥ There is at least one memory element in the feedback path.
⑦ Clock signal is not required	⑦ Clock signal is required

(8) Time is not an Important parameter

(9) They are easier to Implement with the help of basic logic gates but costly : Their Implementation requires more hardware.

(10) Used for Arithmetic as well as Boolean operations

(11) Elementary building BLOCKS : logic gates.

(12) Circuits don't have clock, they don't require triggering

(13) Examples of combinational circuits are. half adder, full adder, magnitude comparator, multiplexer, demultiplexers etc.

(14) faster in speed

(15) Parallel adder one more example

(8) Time is an Important parameter. for timing and synchronizing of different circuit elements, a clock signal is necessary

(9) They are difficult to Implement but less costly than sequential circuits.

(10) Mainly used for storing data.

(11) Elementary building blocks : Flip Flops.

(12) Circuits are clock dependent : They need triggering.

(13) Examples are flipflops registers, counters. etc.

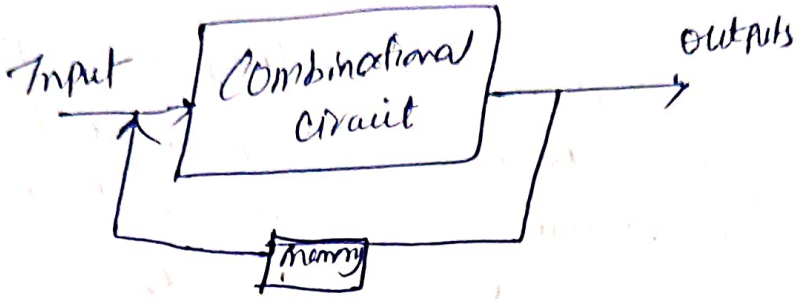
(14) slower compared to combinational circuit

(15) serial adder one more example

* Sequential Circuits *

→ Sequential ckt is a combinational ckt with memory

→ The output of sequential ckt depends upon present inputs and past outputs (present state)



→ The information stored in sequential ckt represents present state.

→ The present state & present inputs are define output next state

→ These are classified two types

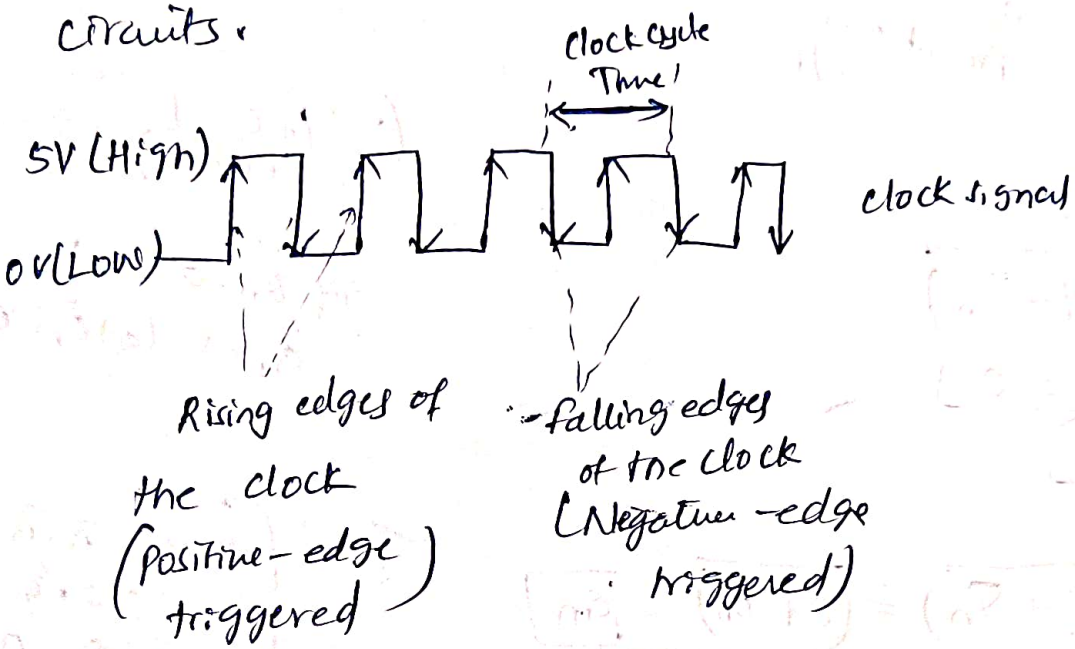
- Synchronous
  - clock pulse present
  - Eg: flip flops — eg SR-flip flop, JK-flip flop, T-flip flop, D-flip flop
- Asynchronous
  - clock pulse not required
  - Eg: Latches — eg SR-latch, JK-latch, T-Latch, D-Latch.

*** Latch *** Latch is a digital electronics device that will stores single bit information either '0' and '1'.

*** flip flop *** A flip flop is an electronic circuit with two stable states that can be used to store binary data. The stored data can be changed by applying varying inputs.

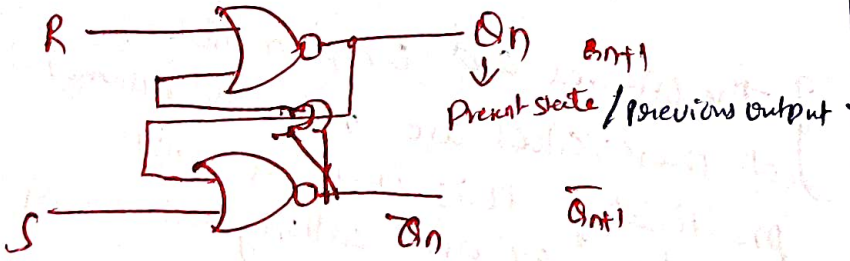
→ flip flops and latches are fundamental building blocks of digital electronic systems used in computers, communications, and many other types of systems.

*** Clock Pulse *** A Clock pulse is a signal that oscillates between high and low states, & is used to synchronize the operations of flip flops & other components in digital circuits.

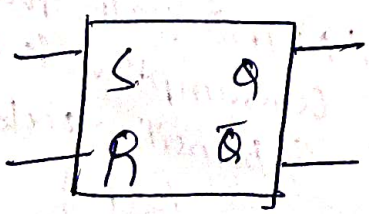


* Latches *

* SR Latch (NOR) *



Logic symbol



S	R	Q_n (Present state)	Q_{n+1}	
0	0	0	0	No change
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0	x	undefining
1	1	1	x	

(i) $S=0$; $R=0$

$Q_{n+1} = \overline{0+\bar{Q}_n} = \bar{\bar{Q}}_n = Q_n \Rightarrow \boxed{Q_{n+1} = Q_n}$

$\bar{Q}_{n+1} = \bar{Q}_n$

↓ Next state O/P ; ↓ Present state O/P.

(ii) $S=0$; $R=1$

$Q_{n+1} = \overline{(1+\bar{Q}_n)}$

$\boxed{Q_{n+1} = \bar{1} = 0}$

$\boxed{\bar{Q}_{n+1} = 1}$

(∵ $1+0=1$, $1+1=1$)

If $Q_n = 0$ (or) 1

$\bar{Q}_n = 1$ (or) $\bar{Q}_n = 0$

(iii) $S=1$; $R=0$

$Q_{n+1} = \overline{(R+\bar{Q}_n)} = \overline{(0+\bar{Q}_n)} = \boxed{Q_n}$

$\boxed{\bar{Q}_{n+1} = \bar{Q}_n}$

$\boxed{Q_{n+1} = 1}$

$\bar{Q}_{n+1} = \overline{(S+Q_n)}$

$= \overline{(1+Q_n)}$

$= \bar{1} = 0$

$\overline{A+B} = \bar{A}\cdot\bar{B}$

$= 0 + \bar{Q}_n$

$= Q_n$

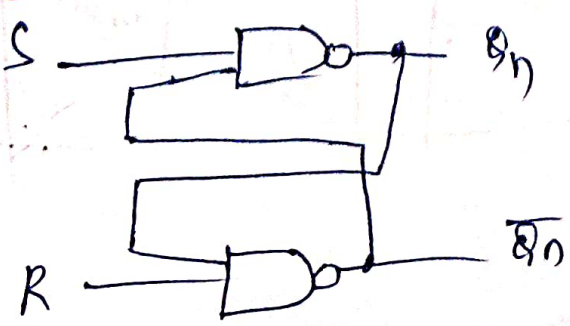
(iv) $S=1, R=1$

$Q_{n+1} = \overline{(1+\bar{Q}_n)} = \bar{1} = 0$

$\bar{Q}_{n+1} = \overline{(1+Q_n)} = \bar{1} = 0$

} The original and complements of the values are not valid (same) practically this state is not existed so we are calling this one as "undetermine state"

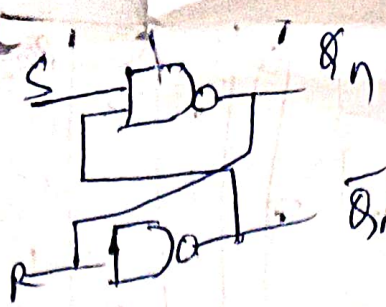
* SR Latch using NAND *



(∵ NAND gate power consumption is less & transition rate is fast)

(1) S = 0, R = 1

$Q_n = 1$; $\bar{Q}_n = 0$



S = 1 ; R = 1 (memory state)

$Q_n = 1$; $\bar{Q}_n = 0$

NAND gate

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

(4)

(2) S = 1 ; R = 0 ; $Q_n = 0$; $\bar{Q}_n = 1$

S = 1 ; R = 1 ; $Q_n = 0$; $\bar{Q}_n = 1$

memory state

(3) S = 0 ; R = 0 ; $Q_n = 1$, $\bar{Q}_n = 1$ → doesn't exist

S	R	Q	$\bar{Q}$
0	0	Invalid.	
0	1	(1)	0 → set state
1	0	(0)	1 → Reset conditions
1	1	memory state / No change	

Gated SR Latch

→ They are synchronizing with enable input

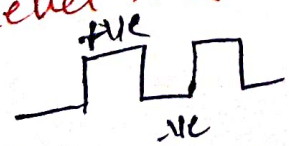
→ They are level triggered.

(Triggering is the process of applying an external signal to induce transition from one state to another state)

Triggering: The signal used for triggering is either a pulse of short duration (or) a step voltage

Triggering
- level triggering
- edge

level triggering

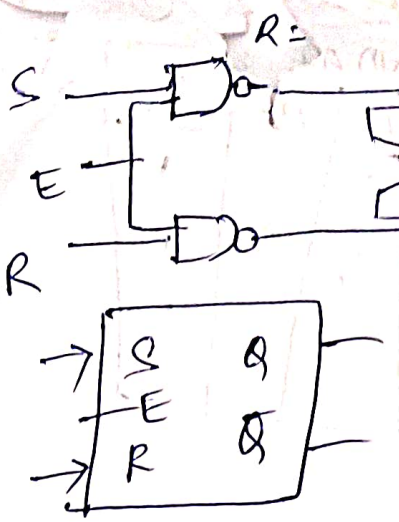


[occurs in latches]

edge

low to high → +ve edge

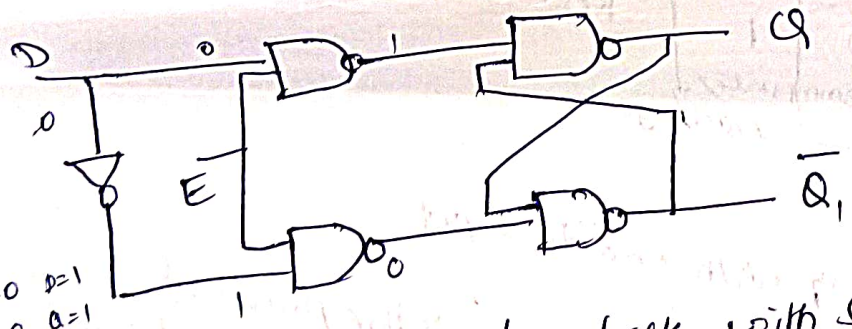
high to low → -ve edge.



E	S	R	Q	Q(n+1)	State
1	0	0	0	0	No change
1	0	0	1	1	
1	0	1	0	0	Reset
1	0	1	1	0	
1	1	0	0	1	Set
1	1	0	1	1	
1	1	1	0	X	Invalid
1	1	1	1	X	
0	X	X	0	0	No change
0	X	X	1	1	

SR → Set & Reset
D → Data / Delay / Transparent
JK → Jack Kilby
(Invented by Jack Kilby in 1950)
T → Toggle

D-Latch



D=0 D=1
Q=0 Q=1
$\bar{Q}$=1 $\bar{Q}$=0

→ There is a one drawback with SR latch that when the inputs S & R = 1, the next state values cannot be predicted

→ This will be overcome by using D-Latch

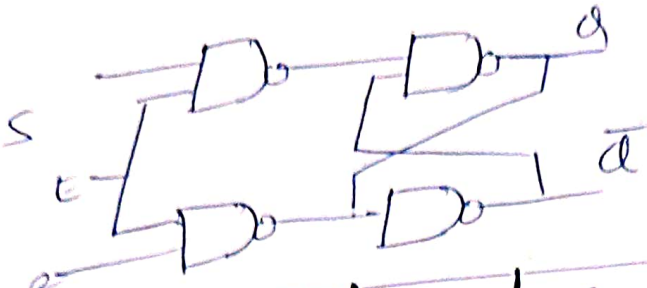
→ Data latch is also known as D-latch

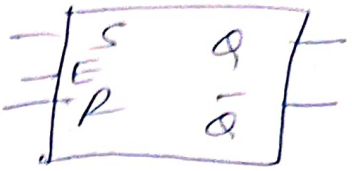
Truth table

E	D	Q	$\bar{Q}$
1	0	0	1
1	1	1	0

* Gate SR latch *

→ It is also known as Synchronous . They are Synchronising with Enable I/P . They are level triggred.

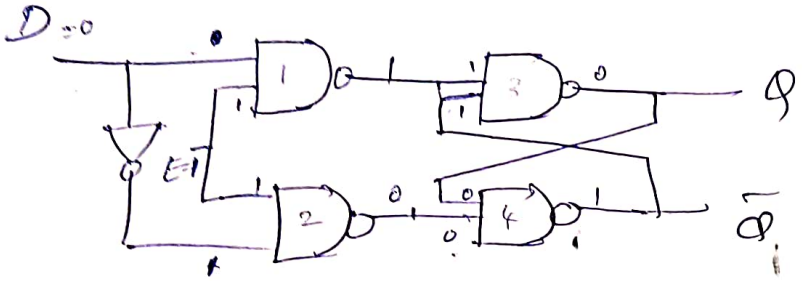




E	S	R	Q_n	Q_{n+1}	State
1	0	0	0	0	NO change
1	0	0	1	1	
1	0	1	0	0	Reset
1	0	1	1	0	
1	1	0	0	1	Set
1	1	0	1	1	
1	1	1	0	X	Invalid
1	1	1	1	X	
0	X	X	0	0	No change
0	X	X	1	1	
			P.S	N.S.	

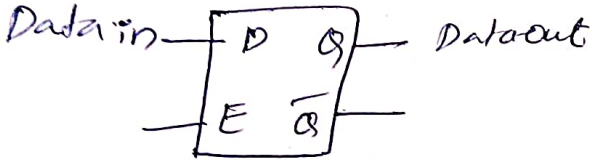
D-Latch

→ The D-latch is a logic ckt most frequently used for storing data in digital systems. It is based on the SR-latch, but doesn't have an "undefined (or) invalid state problem.



NAND gate

A	B	$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0



D-Latch symbol

→ If D=0 ; & E=1 next 1st NAND gate o/p is 1;
& not gate 1, E=1 " 2nd " " " is 0;

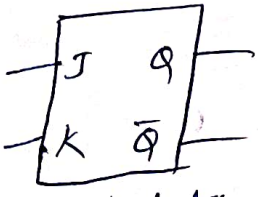
→ Next 4th NAND gate o/p we don't know, then we are assuming that Q=0 (or) 1, but in this case iam considering Q=0;
next 4th gate i/p 0 & Q=0, then o/p of 4th NAND gate is 1.

→ Next 3rd gate i/ps are 1 & $\bar{Q}$=1, then o/p of 3rd NAND gate is 0

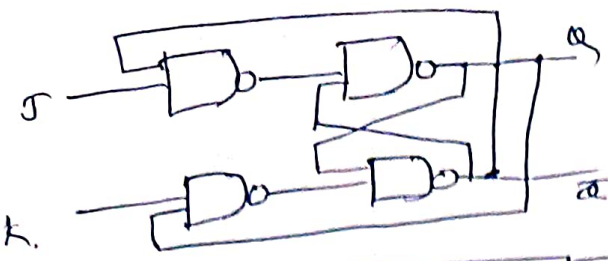
→ So If we apply D=0 ; Q=0 ; $\bar{Q}$=1
If D=1 ; Q=1 ; $\bar{Q}$=0

→ In D-latch the same input is received at the output side. So it is known as "transparent latch".
& It is taking some time to give response at the o/p side, so it is known "Delay latch"

JK latch



JK Latch symbol



NAND

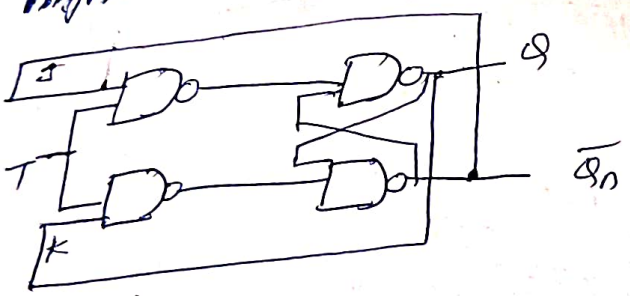
A	B	$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

(6)

J	K	Q_n	Q_{n+1}	state
0	0	0	0	No change
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0	1	Toggle.
1	1	1	0	

* T-Latch *

→ When the JK i/p's of JK latch are shorted we get the 'T-latch', In T-latch o/p's are toggled when the inputs are high.



T	Q_n	Q_{n+1}	
0	0	0	
0	1	1	
1	0	1	Toggled
1	1	0	

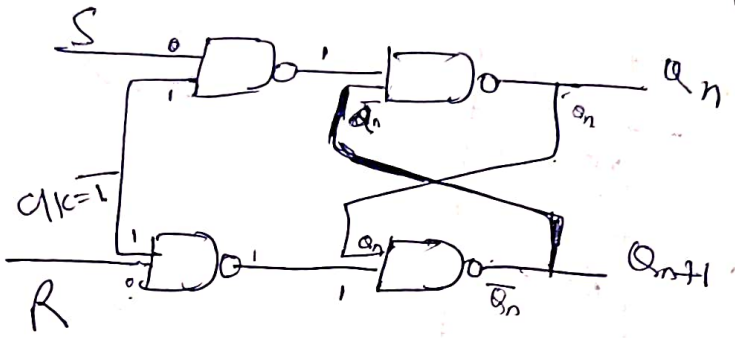
* Flip flops *

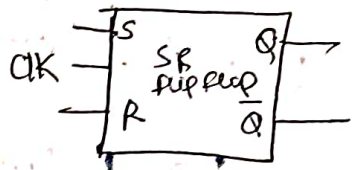
→ It is a sequential ckt which has two states and can be to store state info/m. each flip flop can store 1 bit information. We will use the clock pulses in flip flops.

→ flip flops are fundamental building blocks of digital electronics systems used in computers, communications, and many other types of systems.

Types: ① SR flipflop ② JK flipflop
③ D-flipflop ④ T-flipflop

(1) SR flip-flop : It is SR latch using NAND gates & an additional enable i/p (i.e clock).





Block diagram

NAND

A	B	$\overline{AB}$	AB
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

⇒ When S=0, R=0 ; $1 \times \overline{Q_n} = Q_n$

$1 \times Q_n = \overline{Q_n}$ } No change

⇒ When S=0; R=1 ; $1 \times \overline{Q_n} = Q_n = 0$ (Q_{n+1})

$0 \times Q_n = \overline{0} = 1$ ($\overline{Q_{n+1}}$)

S=0, R=1 $1 \times \overline{Q_n} = Q_n$

S=0; R=1 $Q_n = 0$

1 1 ⇒ $0 \times Q_n = 0$

⇒ When S=1; R=0; $1 \times \overline{Q_n} = \overline{0} = 1$ (Q_{n+1})

$1 \times Q_n = \overline{Q} = 0$

S=1 ; R=0 ⇒ $0 \times \overline{Q_n} = \overline{0}$

∴ } D

⇒ When S=1; R=1 ⇒ $Q_{n+1} = \overline{0 \times \overline{Q_n}} = \overline{0} = 1$

$\overline{Q_{n+1}} = \overline{0 \times Q_n} = \overline{0} = 1$

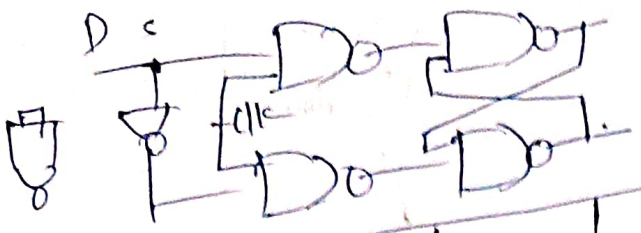
S=1, R=1

$0 \times \overline{Q_n} = \overline{0} = 1$

Truth Table

CLK	S	R	P.S Q_n		Next state Q_{n+1}	
1	0	0	X	0	0	No change
				1	1	
1	0	1	X	0	0	Reset
				1	0	
1	1	0	X	0	1	set
				1	1	
1	1	1	X	0	X	Not possible.
				1	X	

* D-Flip Flop *



Clk	D	Q_n	Q_{n+1}	State
↑	0	0	0	Reset
↑	0	1	0	
↑	1	0	1	Set
↑	1	1	1	
0	X	0	0	No change
0	X	1	1	

* SR flipflop *

Truth table

S	R	PS Q_n	NS Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	Indeterminate
1	1	1	

Excitation table

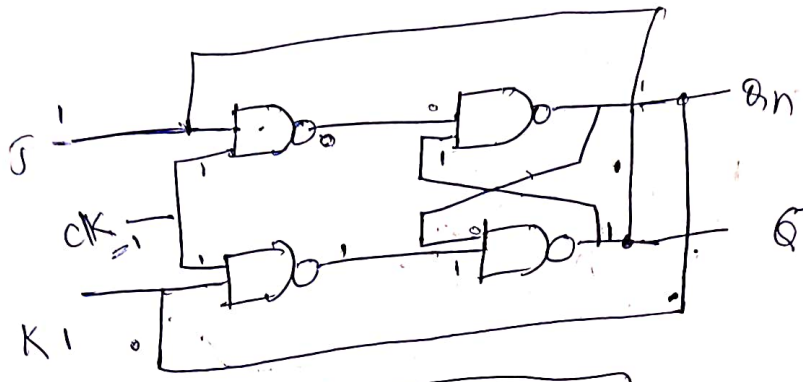
Ps Q_n	NS Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

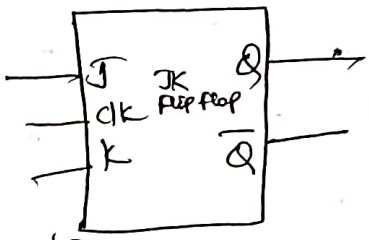
① S R
0 0
0 1
0 X

② only ① S R
0 0
0 1

X (0 0) (0 1)

② JK flip flop :-





Block diagram

NAND gate

A	B	$\overline{AB}$	$\overline{\overline{AB}}$
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

J	K	Q_n	Q_{n+1}
0	0	X	Q_n (M.S)
0	1	X	0 (Reset)
1	0	X	1 (set)
1	1	X	$\overline{Q_n}$ Toggle

(i) $J=0$; $K=0$ (G) $\Rightarrow Q_{n+1} = 1 \cdot \overline{Q_n} = Q_n$

$\Rightarrow \overline{Q_{n+1}} = \overline{1 \cdot Q_n} = \overline{Q_n}$

(ii) $J=0$; $K=1$ (R)

$Q_{n+1} = 1 \cdot \overline{Q_n} = Q_n = 0$

$\overline{Q_{n+1}} = \overline{Q_n \cdot \overline{Q_n}} = \overline{0} = 1$

(iii) $J=1$; $K=0$ (S)

$Q_{n+1} = \overline{Q_n \cdot \overline{Q_n}} = \overline{0} = 1$

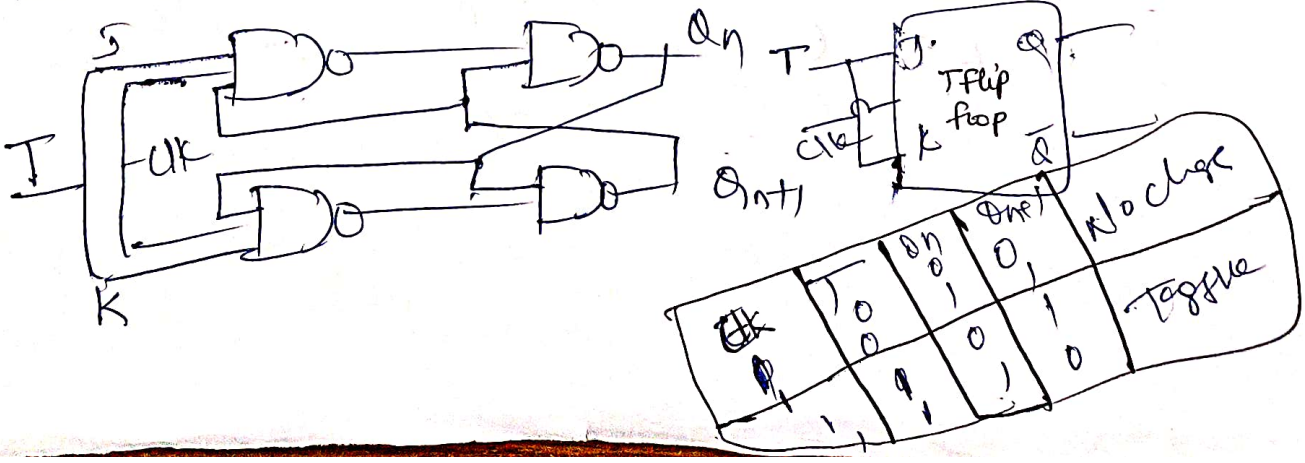
(iv) $J=1$; $K=1$ (R)

Assume $Q_n = 0$ & $\overline{Q_n} = 1$

$Q_{n+1} = 0 \cdot \overline{Q_n} = \overline{0} = 1$

$\overline{Q_{n+1}} = \overline{1 \cdot Q_n} = \overline{Q_n} = 0$

T flip flop



Characteristic eqn

S	R	PS Q_n	NS Q_{n+1}
0	0	x	Q_n
0	1	x	0
1	0	x	1
1	1	x	ID

S R	Q_n	Q_{n+1}	
0 0	0	0	
0 0	1	1	
0 1	0	0	Reset
0 1	1	0	
1 0	0	1	set
1 0	1	1	
1 1	0	X	
1 1	1	X	

K-map

SR / Q_n	00	01	11	10
0			x	1
1	1		x	1

$Q_{n+1} = Q_n\bar{R} + S$

Truth table

J	K	PS Q_n	NS Q_{n+1}
0	0	x	Q_n
0	1	x	0
1	0	x	1
1	1	x	$\bar{Q}_n$

Q_n	Q_{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

J K

0 0 x

0 1 x

1 0 x

1 1 x

0 0

1 1

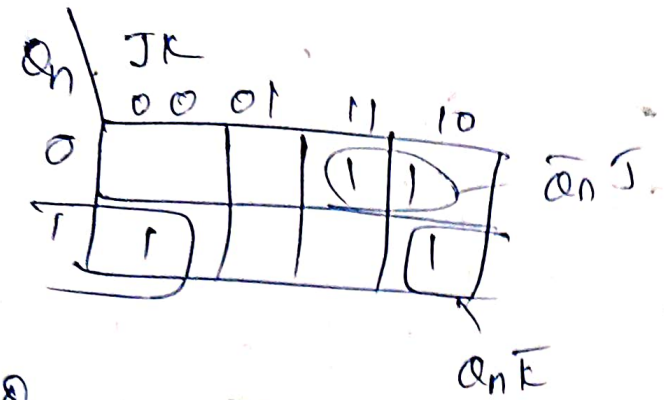
0 0

1 0

Characteristic eqn JK.

J	K	Q_n	Q_{n+1}
0	0	x	Q_n
0	1	x	0
1	0	x	1
1	1	x	$\bar{Q}_n$

J	K	Q_n	Q_{n+1}	
0	0	0	0	
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	set
1	0	1	1	
1	1	0	1	Toggle
1	1	1	0	



$$Q_{n+1} = \bar{Q}_n J + Q_n \bar{K}$$

Excitation table

* D flip flop *

D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

Excitation table

PS Q_n	NS Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

* T flip flop *

T truth table

T	PS Q_n	NS Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Excitation table

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

* Flip flop conversions *

(1) Identify available and required flip flop
(2) make characteristic table for required flip flop
(3) make excitation table for available flip flops
(4) write boolean expression for available flip flop
(5) Draw the circuit

* Convert JK to D-flip flop *

Available FF = JK

Required FF = D

for required characteristic table for D

Q_n	Q_{n+1}	D	J	K
0	0	0	0	X
0	1	1	1	X
1	0	0	X	1
1	1	1	X	0

Excitation table for JK

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

for J

Q_n \ D	0	1
0	X	X
1		

J = D

for K

Q_n \ D	0	1
0	X	X
1	1	0

K = $\bar{D}$

J = D K = $\bar{D}$

* D Flipflop to T Flipflop conversions *

Available ⇒ D-ff

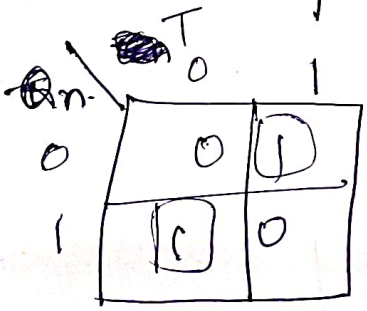
Required ⇒ T-ff

Required characteristic table for T-FF

Q_n	Q_{n+1}	T	D
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	1

Excitation table for DFF

Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1



$$D = \bar{T}Q_n + \bar{Q}_n T$$

$$D = T \oplus Q_n$$

* Conversions of flip flops *

JK to D-FF ⇒ $J=D$ and $K=\bar{D}$

JK to T-FF ⇒ $J=K=T$

JK to SR FF ⇒ $J=S$ and $K=R$

SR-FF to D-FF ⇒ $S=D$ and $R=\bar{D}$

SR-FF to T-FF ⇒ $S=T\bar{Q}$ and $R=TQ$

SR-FF to JK-FF ⇒ $S=J\bar{Q}$ and $R=KQ$

D-FF to SR-FF ⇒ $D=S+\bar{R}Q$

D-FF to T-FF ⇒ $D=T\oplus Q$

D-FF to JK FF ⇒ $D=J\bar{Q}+\bar{K}Q$

T-FF to D-FF ⇒ $T=D\oplus Q$

T-FF to JK-FF ⇒ $T=J\bar{Q}+KQ$

T-FF to SR-FF ⇒ $T=S\bar{Q}+RQ$

* Race around condition (or) Racing in JK flip flop.

⇒ Race around condition is rise in JK flip flop.

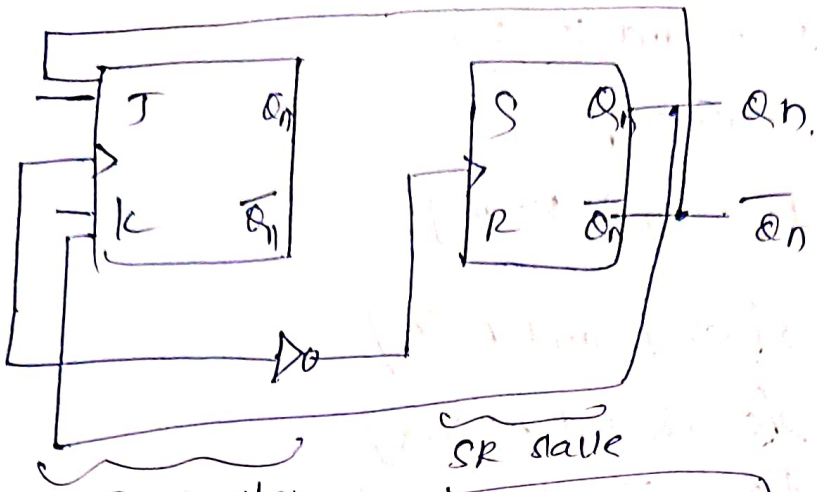
⇒ When $J=K=1$ Race around condition is work

⇒ Race around condition is avoided by using Master slave flip flop

⇒ Master work on +ve clock cycle

⇒ slave work on -ve clock cycle.

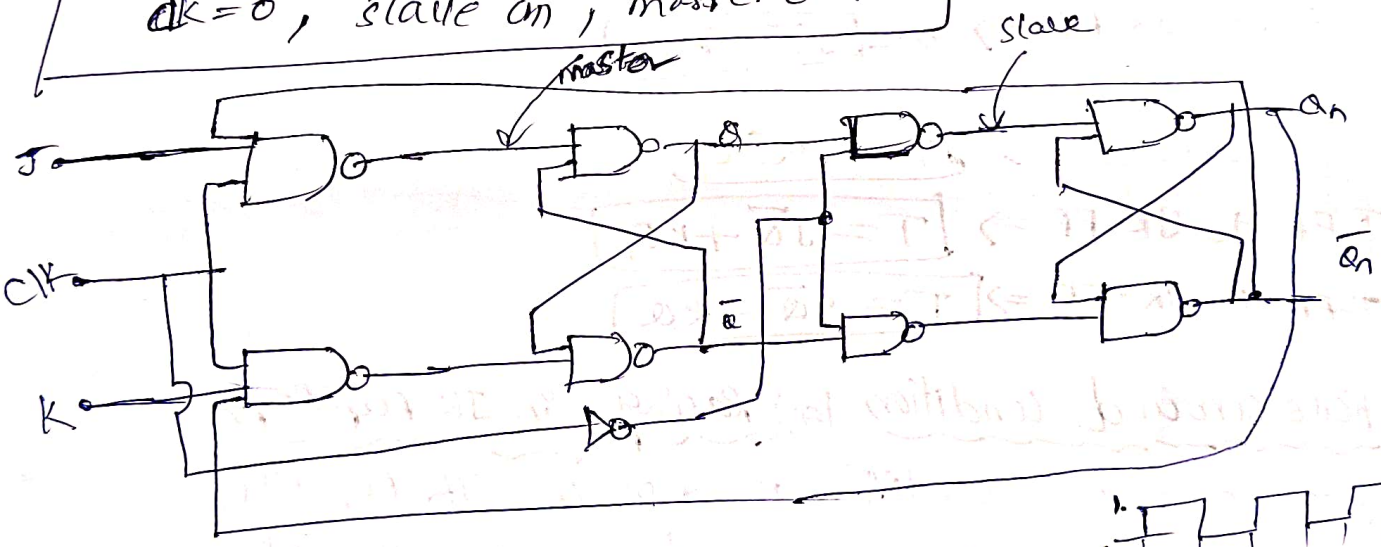
⇒ for JK flipflop if $J=K=1$ & if clock is too long. then state of flipflop keep on toggle which leads to uncertainity in detaining o/p state of flipflop This problem is called "Race around condition"

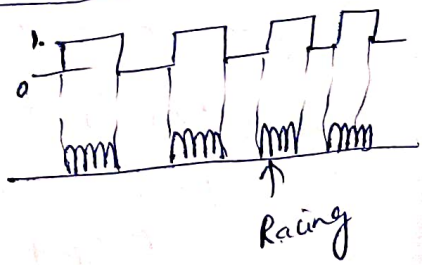


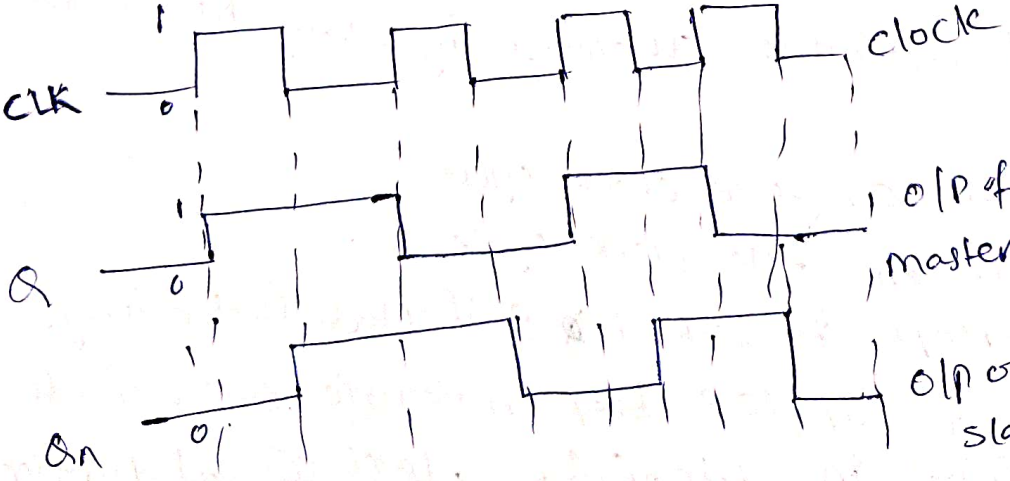
Clock = 1; J = 1; K = 1

CLK = 1, master on, slave off

CLK = 0, slave on, master off







Timing Diagram

So, if we write the value of our o/p when J&K & our clock is high, then the o/p Q_{n+1} will be $\overline{Q_n}$ ie

Clock	J	K	Q_{n+1}
1	1	1	$\overline{Q_n}$

✱ State Reduction ✱

→ State reduction in digital electronics is the process of reducing the number of states in a state table while maintaining the same external input-output requirements. This is done by identifying & merging states that have the same next state entries & outputs.

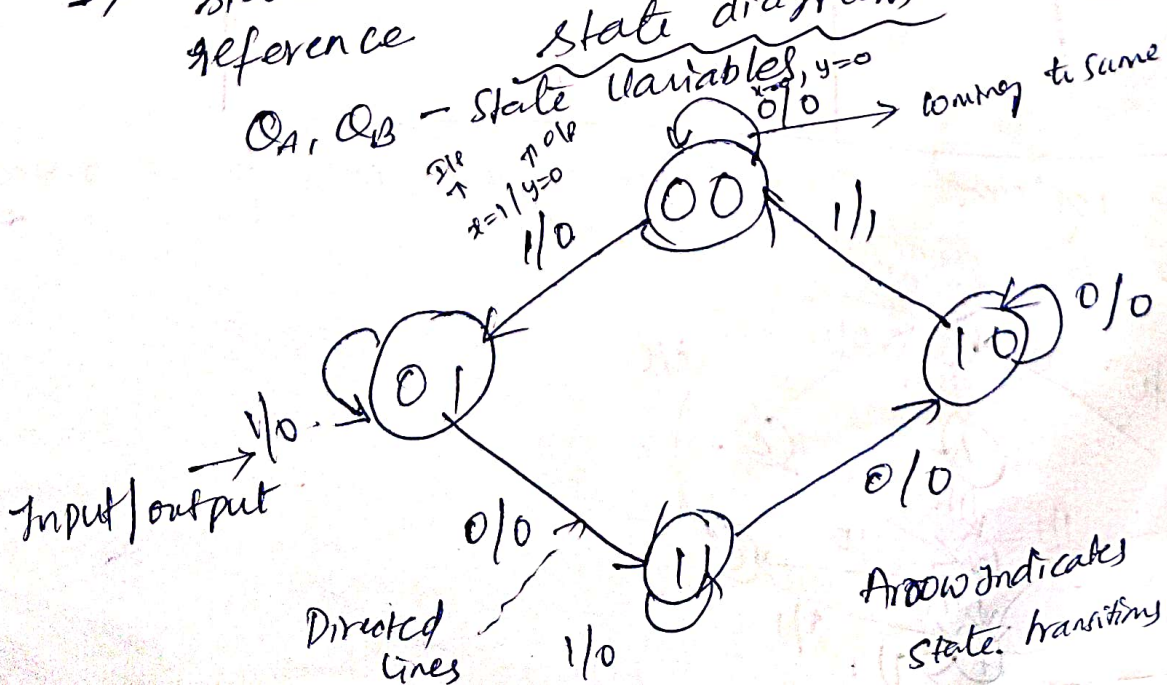
→ state reduction is important because it can lead to simpler designs, which can improve efficiency & reduce resource consumption. [✱ i.e Reducing the no. of flip flops in a sequential ckt is refered to as the state Reduction]✱

→ Sequential circuits that may simplify a design by reducing the number of gates & flip flops it used, reducing the no of flipflops reduces the cost of a circuit.

✱ State Assignment ✱
✱ state diagram ✱ : The pictorial representation of the relationships between the present state, the input & the next state and the output of a sequential ckt.

→ Information available in state table is represent graphically using state diagram.

→ State diagram is drawn by using state table as reference



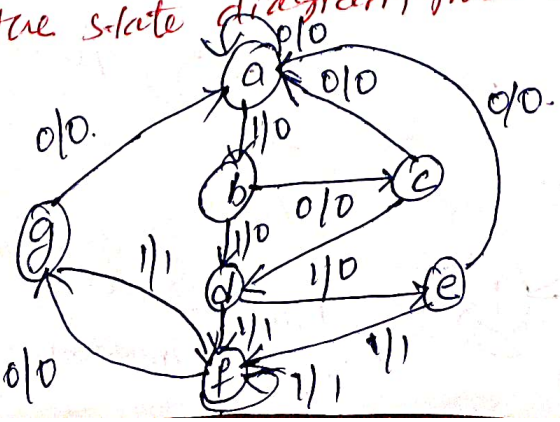
* State table *

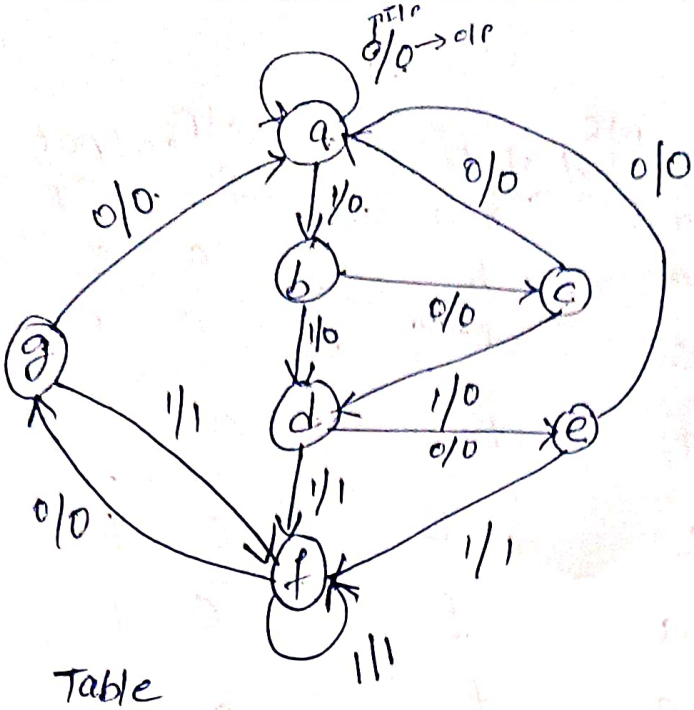
→ Relates the inputs & outputs
→ Outputs Q_A, Q_B are state inputs
→ These are called state variables
→ x represents external input
→ y represents the output
→ y depends on the state variables and external input 'x'.

* General form of State table * → Relates present state Next state & output

Present state	Next state $x=0$ Q_{A+1} Q_{B+1}	Next state $x=1$ Q_{A+1} Q_{B+1}	output y $x=0$	output y $x=1$
Q_A Q_B				

(8) Draw the reduced state table & scelled state diagram for the state diagram given below.





x → input
y → output

State Table

Present state	Next state x=0	Next state x=1	Output x=0	Output 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

e = g

Reducing table

Reducing table:

P.S Present state	N.S Next state x=0	N.S Next state x=1	O/P output x=0	O/P output 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	e	f	0	1

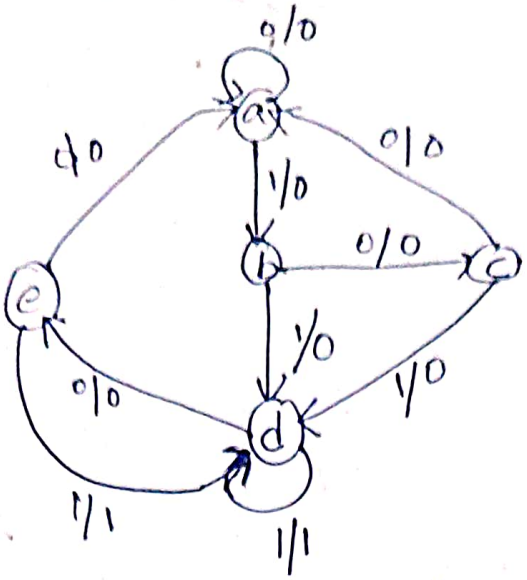
$\therefore$ d = f

Reducing table

Present state	Next state x=0	Next state x=1	output x=0	output 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

[$\because$ By comparing c/e the next states are equal but outputs are not equal. So, this is final reduced state table]

Now we have to draw state diagram for reduced state table



Reduced state diagram.

State assignment:

No. of states = m

No. of Flip flops = n

where $2^n \geq m$

$m = 5$ $\quad 2^3 \geq 5$

$8 \geq 5$ condition is satisfied

Present state	Assignment 1 Binary	Assignment 2 Gray code	Assignment 3 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

Reduced state table with binary assignment 1

Present State	Next State x=0	Next State x=1	Output x=0	Output 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

This Reduced state table with binary Assignment 1,

→ This table is called as transition table of sequential circuit. This table is used for implementing the ckts.

~~Design Procedure~~ :-

* Definations *

* State assignment :- To represent states in the state diagram using binary values instead of alphabets.

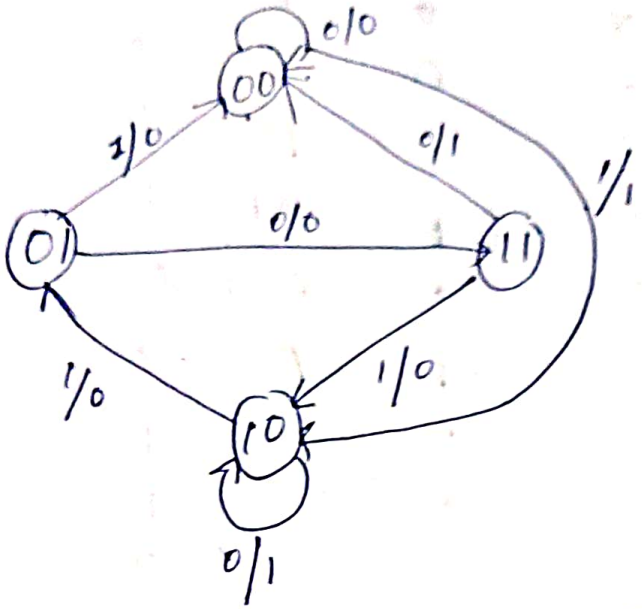
* State diagram :- Information available in state table is represented graphically using state diagram.

→ State diagram is drawn by using state table as Reference arrows indicate state transitions.

* Design procedure for clocked sequential circuits *

1. state diagram
2. state table
3. State Reduction
4. State assignment
5. Determine number of flip flops required
6. Derive circuit excitation table from state table.
7. Implement circuit

(6) Design a sequential circuit that has 1 input and 1 output state diagram is shown. (14)



State table

Present state		Next state				Output	
		$x=0$		$x=1$		$x=0$	$x=1$
A	B	A	B	A	B	y	y
0	0	0	0	1	0	0	1
0	1	1	1	0	0	0	0
1	0	1	0	0	1	1	0
1	1	0	0	1	0	1	0

→ state table shows that circuit goes through four states therefore we require 2 flip flops.

[n^m = no of states
n = no of flip flops]

using T-flipflop:
Excitation table

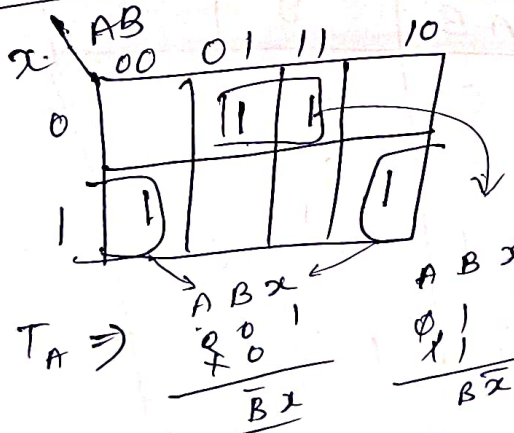
Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

$T = Q_n \oplus Q_{n+1}$

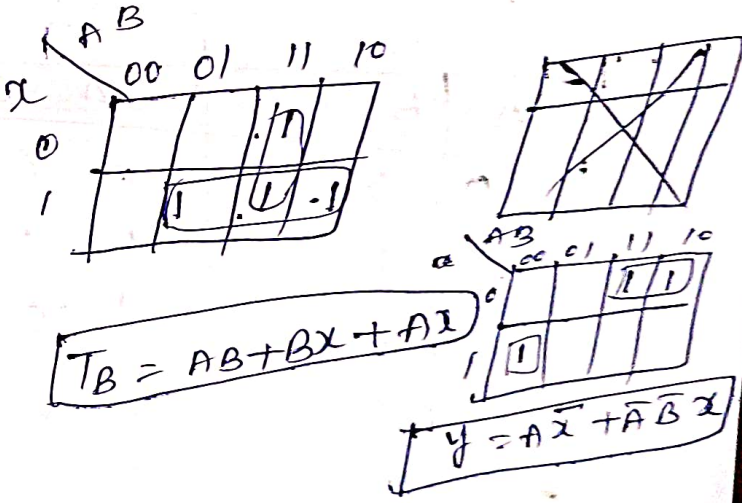
* circuit excitation table *

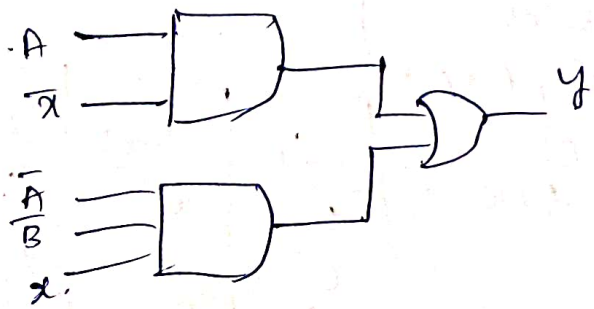
Present state A	Present state B	Input x	Next state A	Next state B	flip flop TA	flip flop TB	output y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	0	1	1	0
0	1	1	1	1	1	0	0
1	0	0	0	0	1	0	0
1	0	1	0	1	1	1	0
1	1	0	0	0	1	1	0
1	1	1	1	0	0	1	0

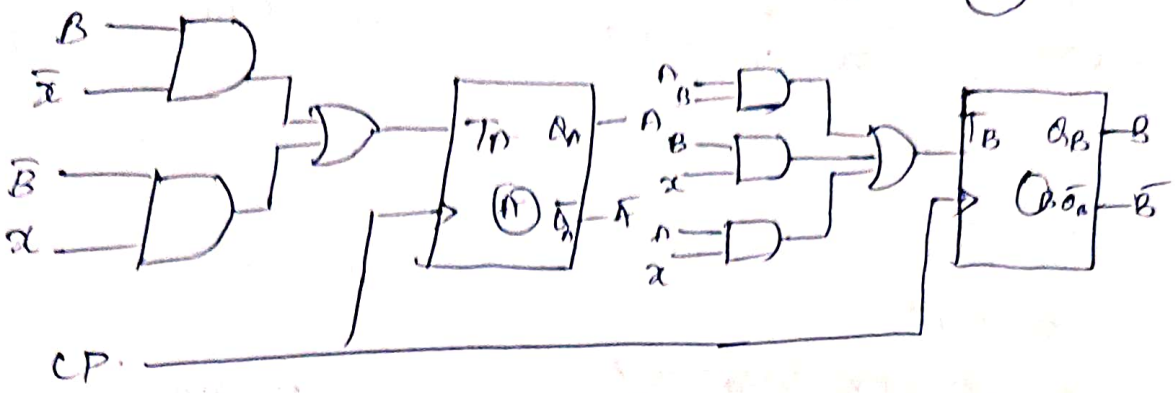
K-map:



$$T_A = B\bar{x} + \bar{B}x$$







Circuit diagram

* Registers *

→ flip flop is a 1-bit memory cell.

→ To increase the storage capacity we have to use group of flipflops. This group of flip flops is known as "Register"

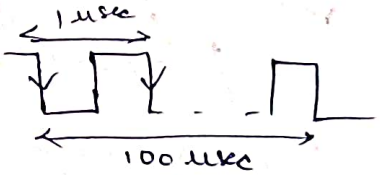
→ The n-bit register consist of 'n' number of flipflops and is capable of storing "n-bit" word.

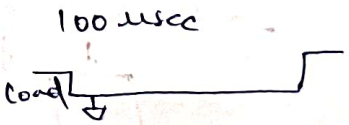
nibble = 4 bits
byte = 8 bits
word = 16 bits
double word = 32 bits
quad word = 64 bits

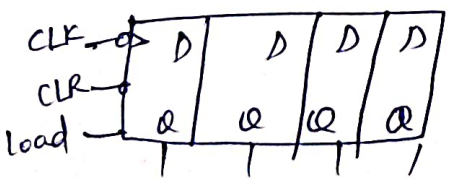
If ~~freq~~ = 1 MHz = 10^6 Hz.

Time period = $\frac{1}{f}$ sec.

$T = 1\ \mu sec$







$\frac{1}{10^6} = 10^{-6}$ = 1 μsec

In synchronous ckts = clock ↑ and load ↑ (i.e clock increases & load is increased).

Asynchronous circuits = Only load ↑ses.

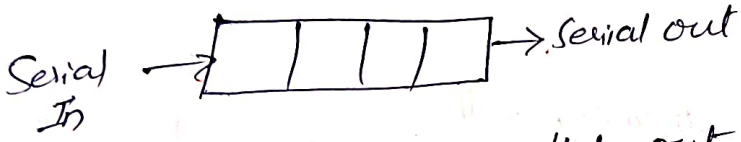
To store the 4 bits, we need 4 Flipflops.

→ **Shift registers**

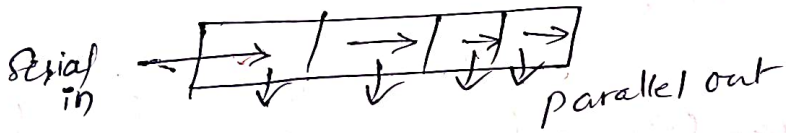
→ Shift register are used to transfer the data.

1. SISO
2. CIPO
3. PISO
4. PIPO

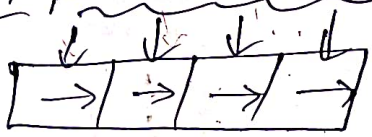
① SISO → Serial in Serial out
→ Data in right (or) left Direction



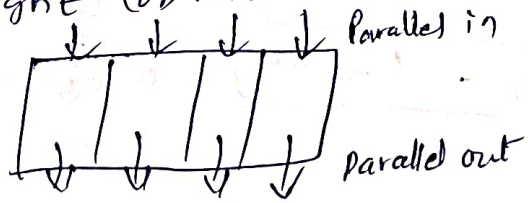
② SIPO → Serial in parallel out
→ Data in right (or) left direction



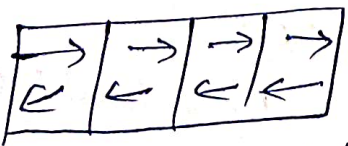
③ PISO → Parallel in Serial out



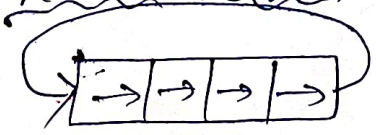
④ PIPO → Parallel in parallel out
→ Right (or) left Direction.



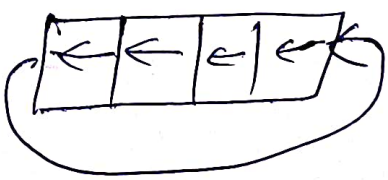
⑤ Bidirectional



⑥ Rotate Right → Anti clock to Right side



⑦ Rotate left → Anti clock wise leftside

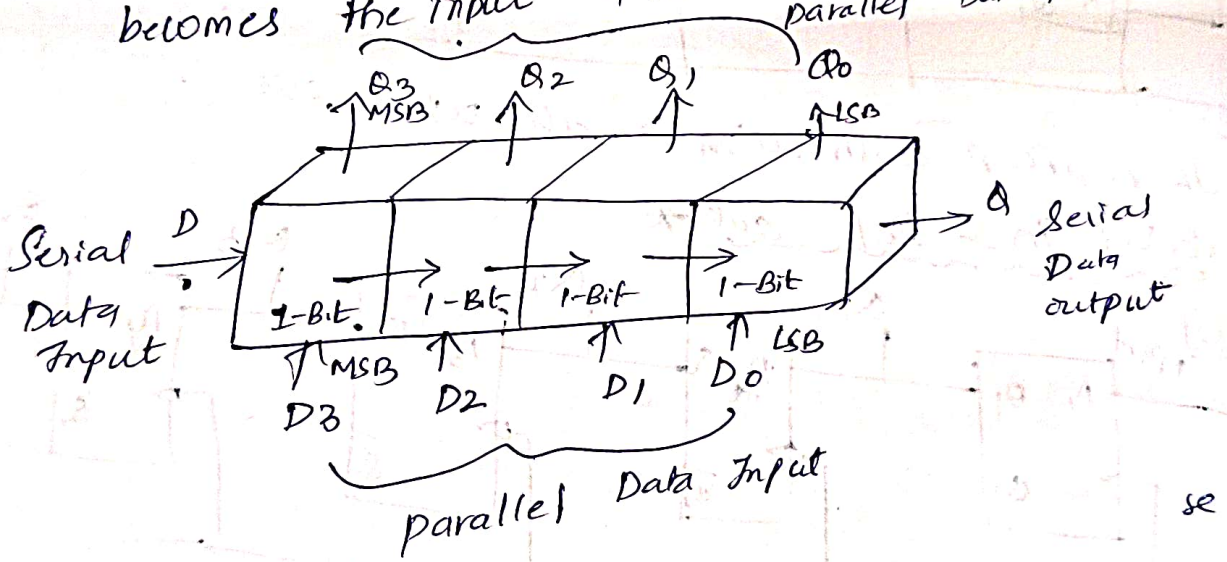


* Applications *

→ Data storage device → In calculators
→ Data transmission device → In computers
→ Serial to parallel } convert to one form to another form
→ parallel to serial }

* The shift register is another type of sequential logic circuit that can be used for the storage (or) the transfer of binary data.

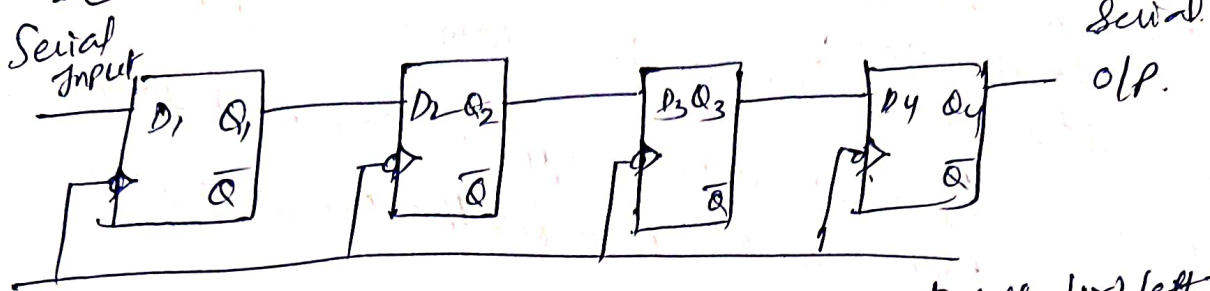
→ The shift register basically consists of several single bit "D-Type Data latches", one for each data bit, either a logic '0' (or) a '1', connected together in a serial arrangement so that the output from one data latch becomes the input of the next latch & so on.



se

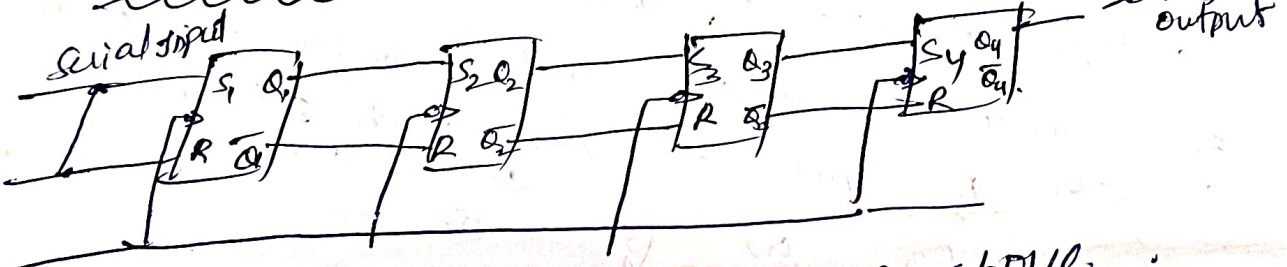
Q Design 4 Bit Serial in & serial out Shift Register

* D-flip flop *



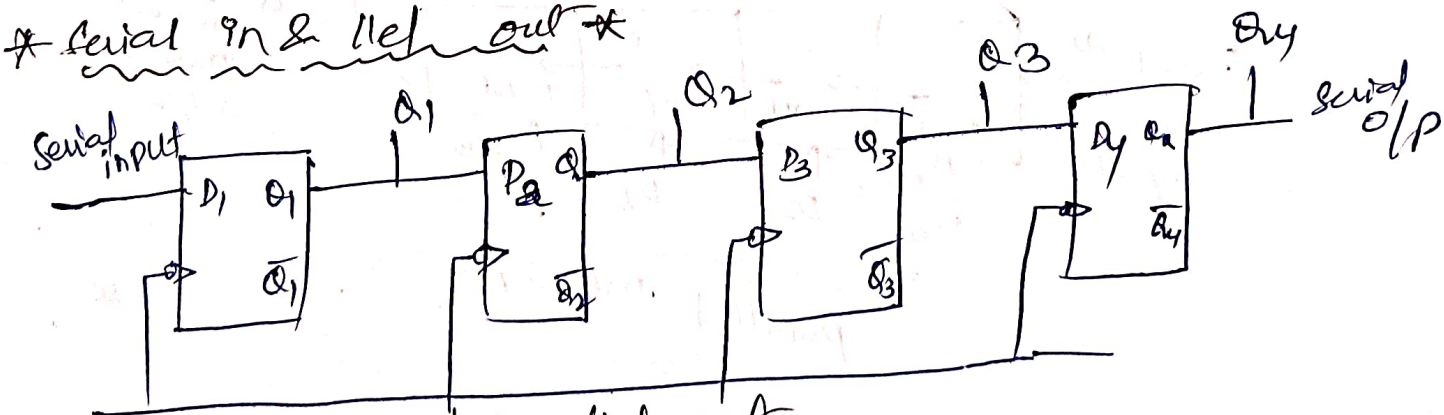
* The shift can be changed from right - to left (or) left to right by interchanging the D & Q states

* SR flip flop *

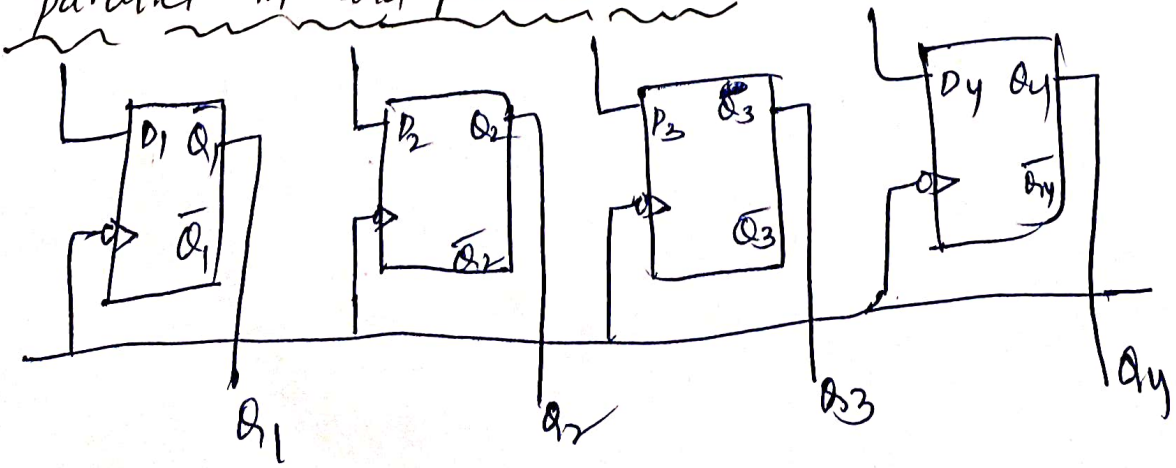


* JK flipflop construction is also as above.

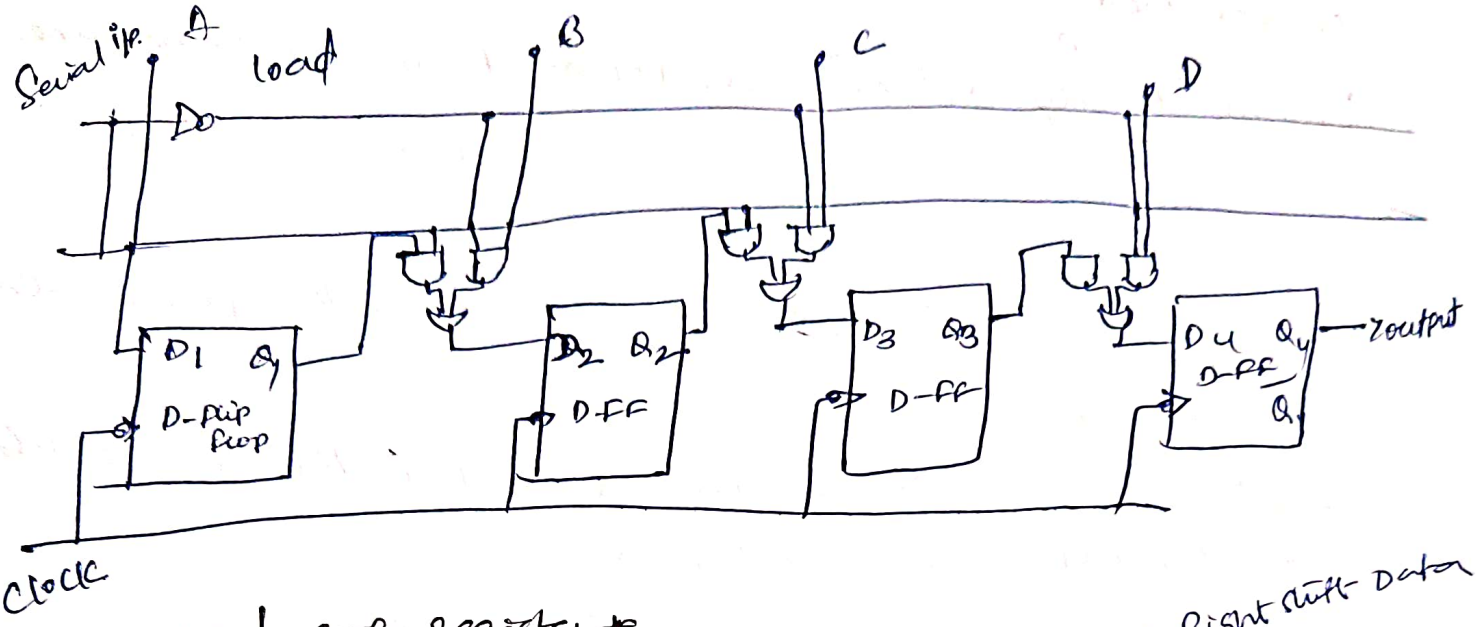
* Serial in & Serial out *



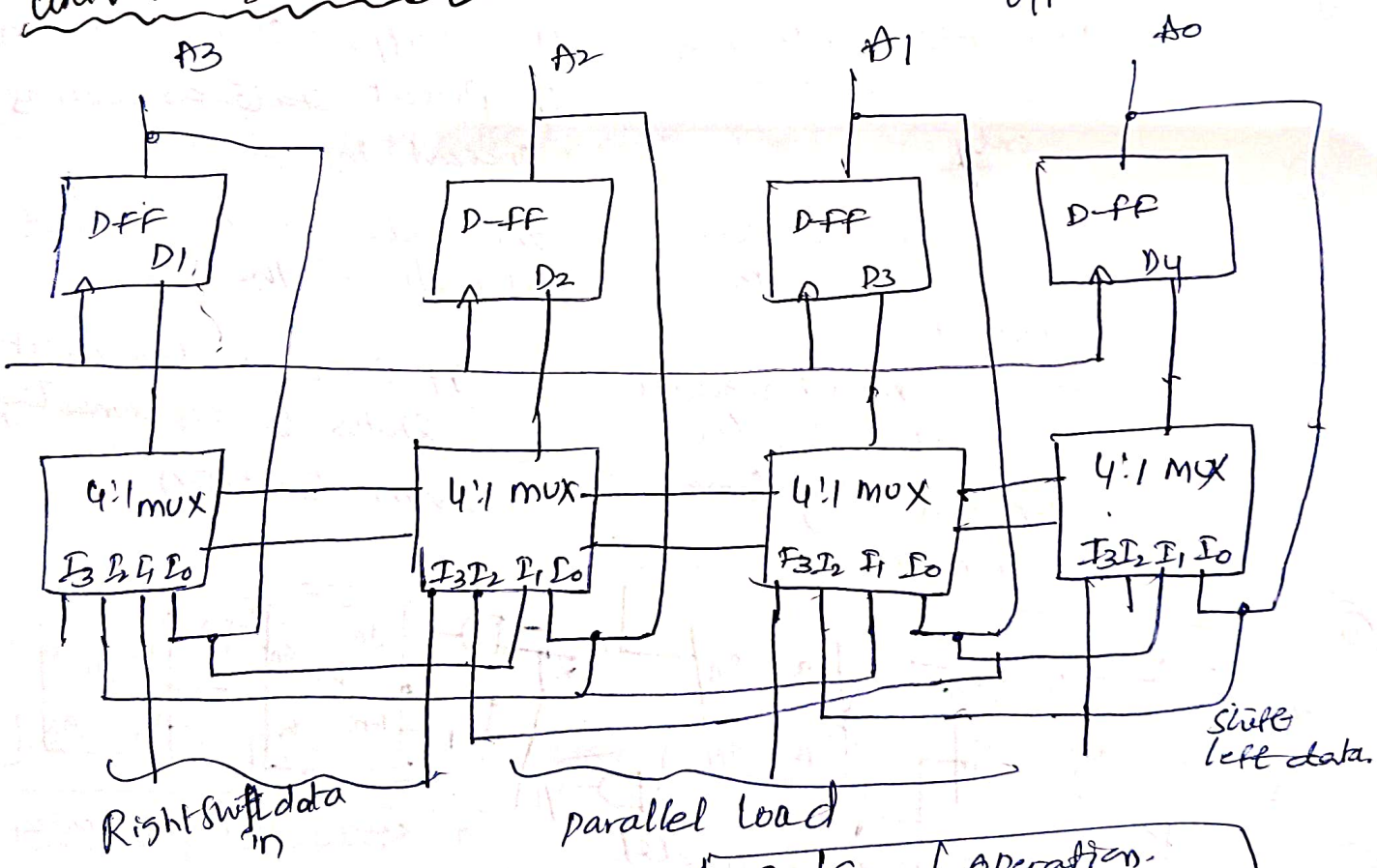
* parallel in and parallel out



* Parallel in serial out *



* universal shift register *



S_1	S_0	Operation
0	0	NC
0	1	SR (Shift Right)
1	0	SL (Shift left)
1	1	parallel load

(a) List out the two types of clocked sequential circuits & compare them.
(or)
Compare mealy and moore model of finite state machine

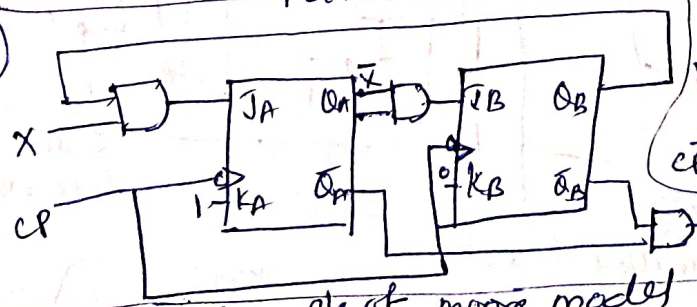
→ The synchronous (or) clocked sequential circuits are represented by two models

Moore model :- The o/p depends only on the present state of flip flops.

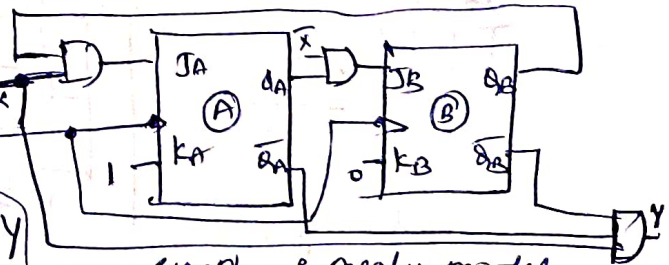
Mealy model :- The o/p depends on both the present state of flip flop (s) & on the inputs

	Moore model	mealy model
①	It's o/p is a function of present state only	It's output is a function of present state as well as present input
②	Input changes does not effect on the output	Input changes may effect the o/p of the ckt
③	Moore model requires more no. of states for implementing same function	It requires less no. of states for implementing same function

④

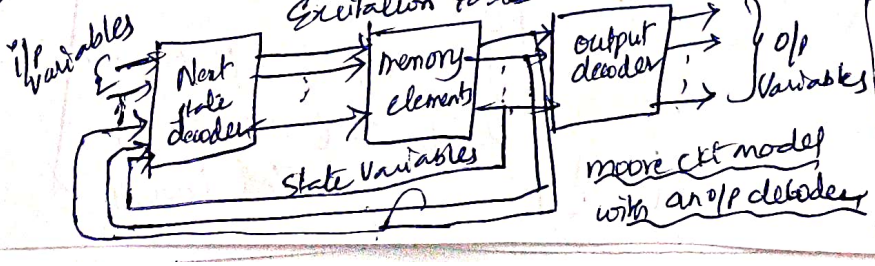


Example of moore model

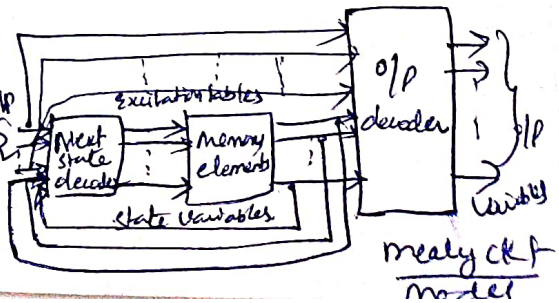


Example of mealy model

⑤



moore ckt model with an o/p decoder



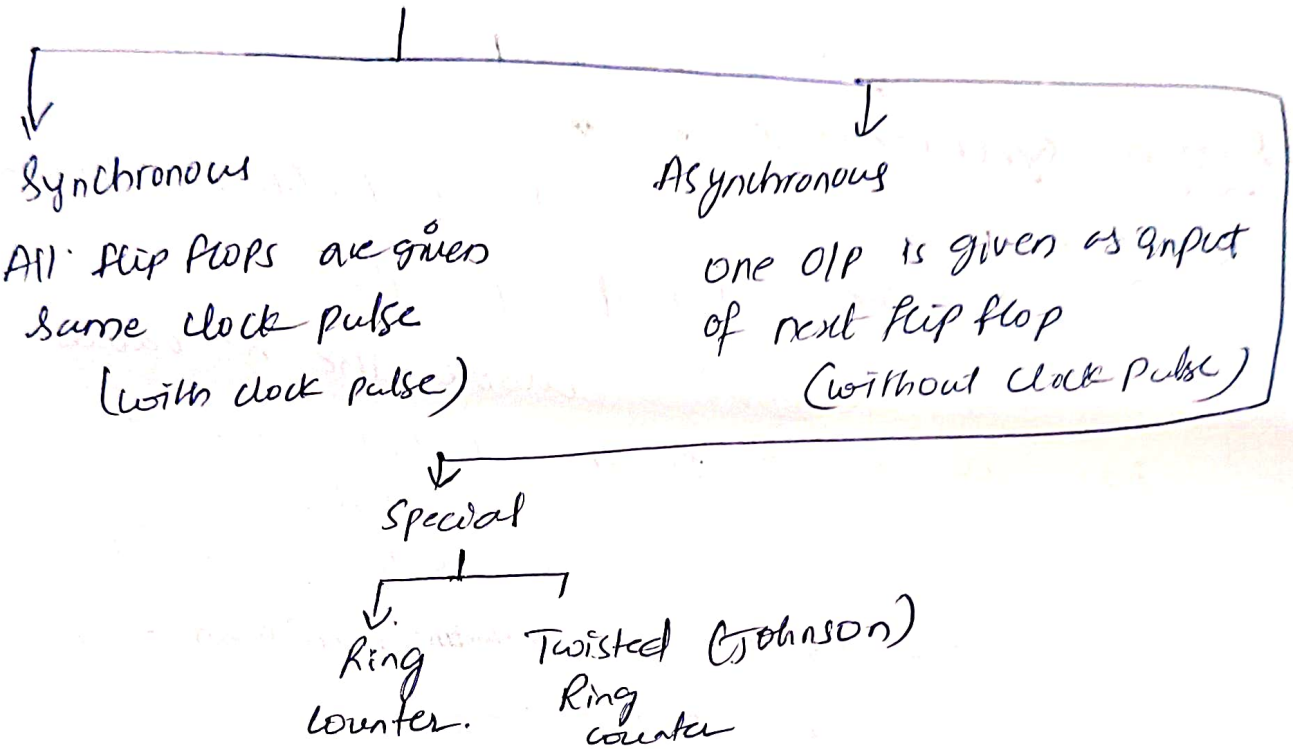
mealy ckt model

Counters

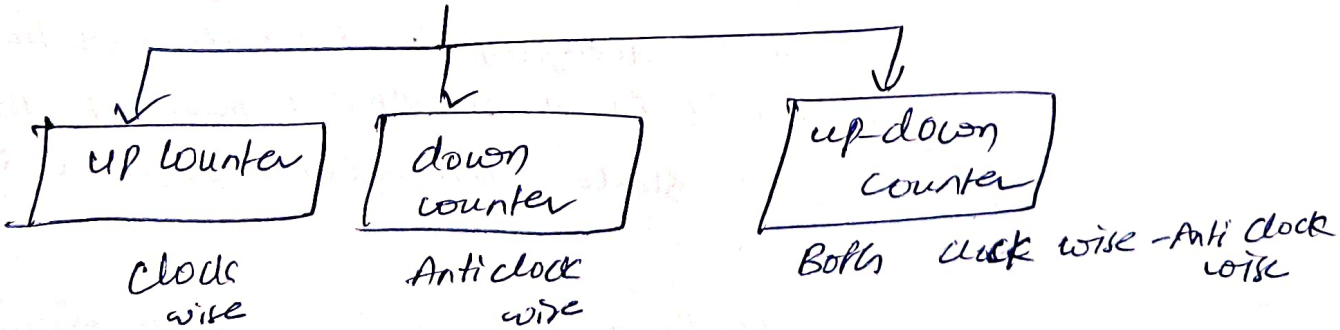
→ Counter is a Sequential Circuit. A digital circuit which is used for a counting pulses is known as counter

→ Counter is a device which stores (and sometimes displays) the no of times a particular event (or) process has occured often in relationship to a clock signal.

→ Types of counters: (Depending on the clock pulses applied Counters are two types),



① Synchronous counter



for example, in UP counter a counter increases count for every rising edge of clock. Not only counting, a counter can follow the certain sequence based on our design like any random sequence 0,1,3,2... They can also be designed with the help of flip flops. They are used as frequency dividers where the freq. of given pulse waveform is divided.

→ Counters are sequential ckt that count the no. of pulses can be either in binary code (or) BCD form. The main properties of counter are timing, sequencing, & counting. Counters works in two modes ① up counter ② Down counter.

* Design synchronous counter *

Step 1:- Based on the discreption of problem determined the required no of flip flops

$N \leq 2^n$ choose the 'n' value

has as small as possible. N is the mode of the counter
↓
No. of states

n → is the required no of flipflops

choose the 'n' as ~~minimum~~ minimum as possible

Ex: $5 \leq 2^3 = 8$

$5 \leq 8$

Step 2: State diagram & Draw the state diagram with all possible state. The state diagram is also called as transition diagram. It is a graphical means of the sequence of state Through the counter is progress.

Step 3: choose of flipflop & Excitation table given in the problem. Otherwise choose default.

Step 4 Minimal expression for excitation obtain minimal Expression by K-map.

Step 5 : Logic diagram

→ Draw the logic diagram according to the minimized Expression

(a) Design mod – 6, gray code synchronous counter by using T-flip flop.

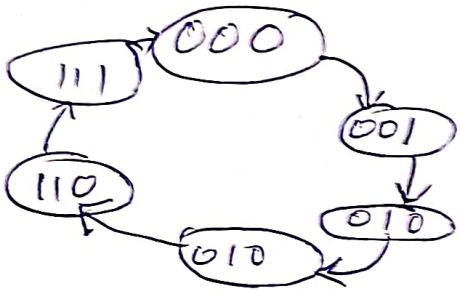
(i) $N = 6$

$$6 \leq 2^3 \Rightarrow 6 \leq 8$$

(ii) The no of flip flops is need = 3

(iii) Here for designing we default up counter

(iv) 3-Bit Gray code is needed.



3 BT Gray code

0 – 000 – 1
1 – 001 – 2
2 – 011 – 3
3 – 010 – 4
4 – 110 – 5
5 – 111 – 6
6 –
7 –

Excitation table

PS			NS					
Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	T_3	T_2	T_1
0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	0	1	0
3	0	1	1	0	0	0	0	1
2	0	1	0	1	0	1	0	0
6	1	1	0	1	1	0	0	1
7	1	1	1	0	0	1	1	1
5	1	0	1	0	1	0	1	0
4	1	0	0	0	0	1	0	0

Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	1
1	1	0

T_3 (K-map)

Q_3 \ Q_2Q_1	00	01	11	10
0	0	0	0	1
1	X	X	1	0

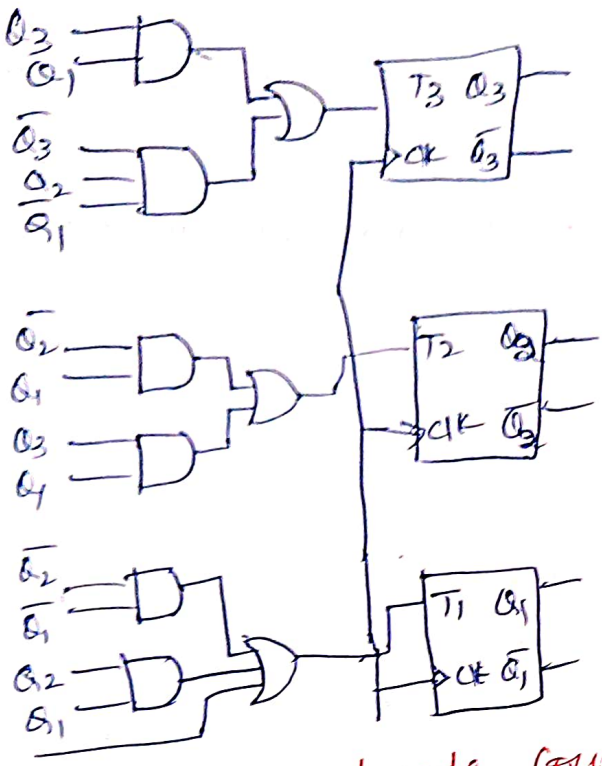
T_2 (K-map)

Q_3 \ Q_2Q_1	00	01	11	10
0	0	1	0	0
1	X	X	1	

T_1 – K-map

Q_3 \ Q_2Q_1	00	01	11	10
0	1	0	1	0
1	0	X	1	1

$T_3 = \bar{Q}_3 Q_2 \bar{Q}_1 + Q_3 Q_1$ $\quad T_2 = \bar{Q}_2 Q_1 + Q_1 Q_3$ $\quad T_1 = \bar{Q}_2 \bar{Q}_1 + Q_2 Q_1 + Q_3$



Q) Design decade counter using T-flip flop

Excitation table

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

$N \leq 2^n$

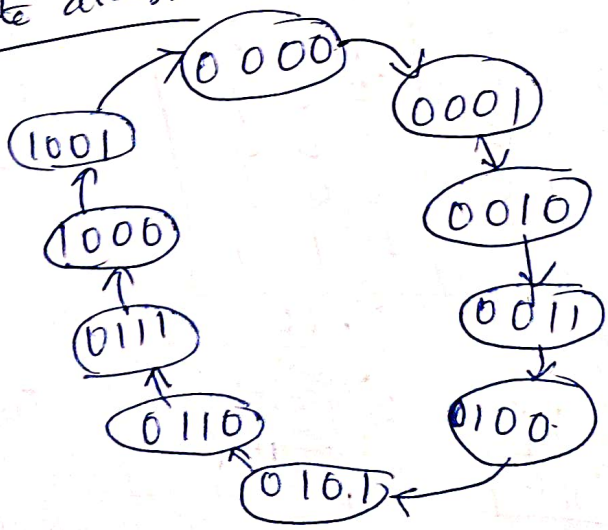
$10 \leq 2^4$

$10 \leq 16$

No. of states = 10

No. of flip flops = 4

State diagram

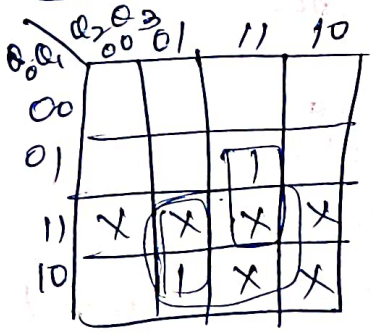


Truth table

Present state $Q_0\ Q_1\ Q_2\ Q_3$	Next state $Q_0\ Q_1\ Q_2\ Q_3$	$T_0\ T_1\ T_2\ T_3$
0 0 0 0	0 0 0 1	0 0 0 1
0 0 0 1	0 0 1 0	0 0 1 1
0 0 1 0	0 0 1 1	0 0 0 1
0 0 1 1	0 1 0 0	0 1 1 1
0 1 0 0	0 1 0 1	0 0 0 1
0 1 0 1	0 1 1 0	0 0 1 1
0 1 1 0	0 1 1 1	0 0 0 1
0 1 1 1	1 0 0 0	1 1 1 1
1 0 0 0	1 0 0 1	0 0 0 1
1 0 0 1	0 0 0 0	1 0 0 1
1 0 1 0	x x x x	x x x x
1 0 1 1	x x x x	x x x x
1 1 0 0	x x x x	x x x x
1 1 0 1	x x x x	x x x x
1 1 1 0	x x x x	x x x x
1 1 1 1	x x x x	x x x x

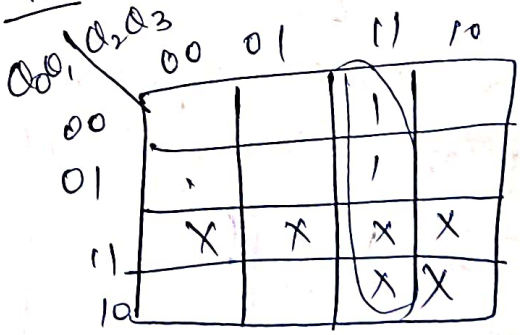
K-map

T_0



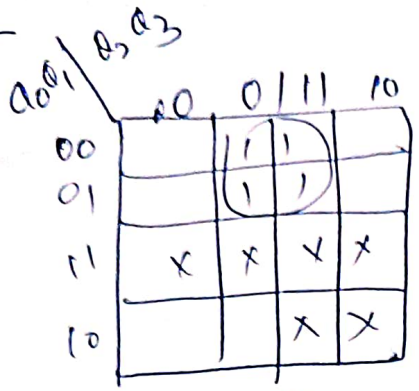
$T_0 = Q_1Q_2Q_3 + Q_0\bar{Q}_2Q_3$

T_1



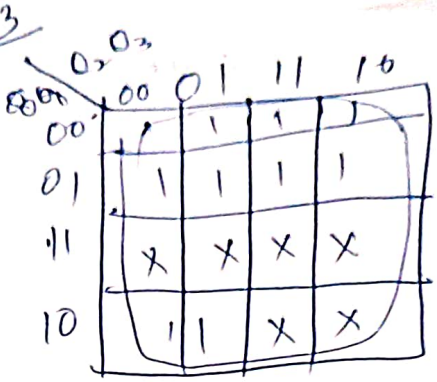
$T_1 = Q_2Q_3$

T_2



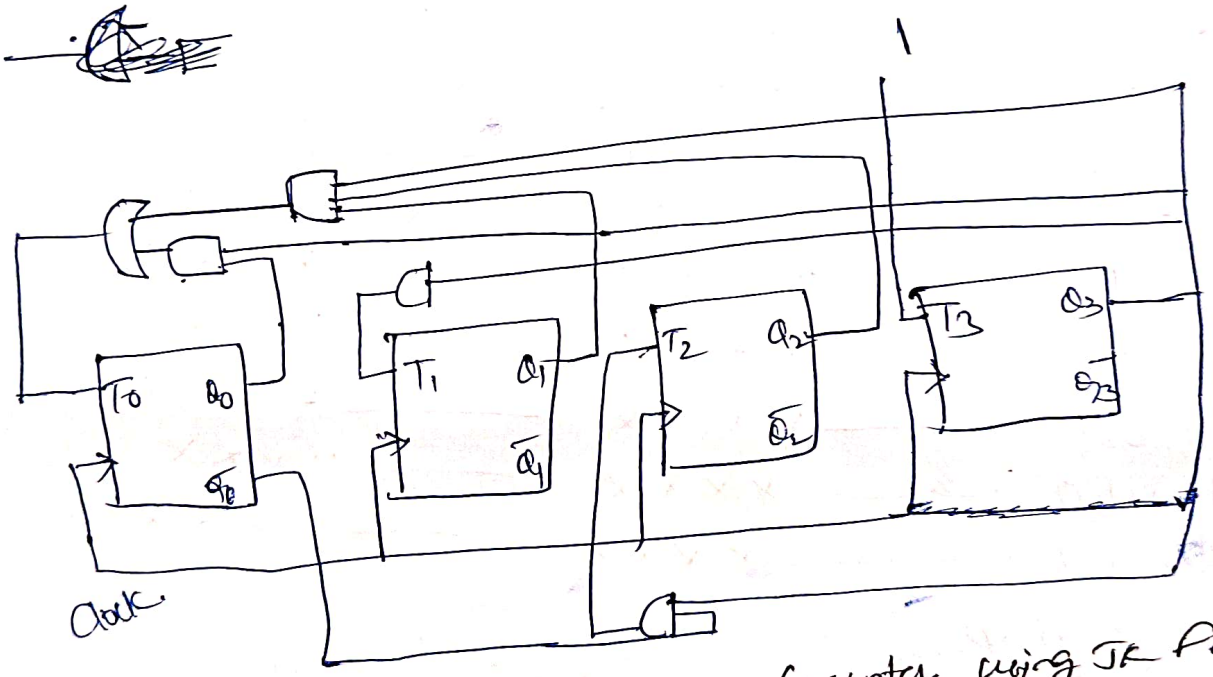
$T_2 = \bar{Q}_0 Q_3$

T_3



$T_3 = 1$

logic diagram



(a) Design a 2-Bit Synchronous Counter using JK FF

Excitation table for JK flip flop.

Q_n	Q_{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

equation

$N \leq 2^n$

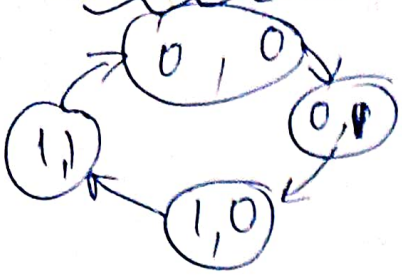
$N \leq 2^2$ $\quad n = 4$

$n \leq 4$

No. of states 4

No. of flip flops = 2

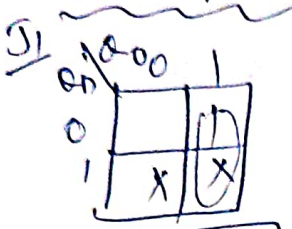
State diagram



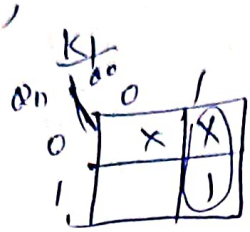
Truth table

P.S	N.S	$J_1 K_1$	$J_2 K_2$
0 0	0 1	0 X	1 X
0 1	1 0	1 X	X 1
1 0	1 1	X 0	1 X
1 1	0 0	X 1	X 1

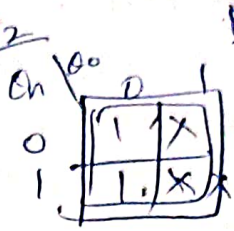
K-map for J_1



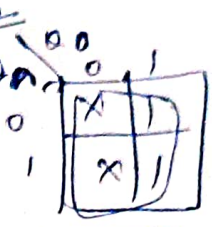
$J_1 = Q_0$



$K_1 = Q_0$

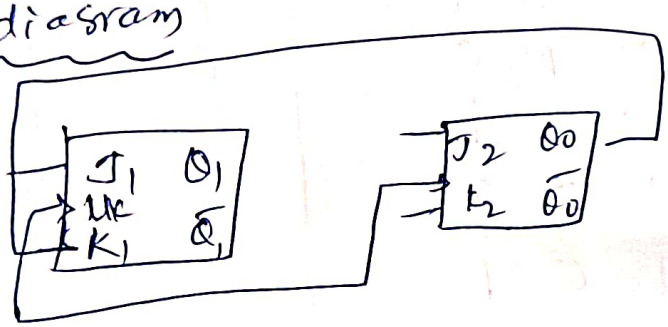


$J_2 = 1$



$K_2 = 1$

Logic diagram



* Asynchronous counters *

→ In this type of counters there is a connection btw the O/p of 1st flip flop & clock input of next flip flop.

→ All flip flops are not clocked simultaneously.

→ clock is quite simple even for more no. of states

→ The performance of the device is low speed.

→ Example Ripple counter.

→ Ripple counter is a special type of Asynchronous counter in which the clock pulse ripples through the ckt. The no. of flipflops in the cascaded arrangement depends upon the no. of different logic states that it goes through before it repeats the sequence a parameter known as modulus of the counter. A n-bit ripple counter can count up to 2^n states. It is also known as MOD n counter.

* V.Imp → The other name for asynchronous counter is Ripple counter.

(a) Design Mod-6 Asynchronous counter by using T-flip flop

$N \leq 2^n \quad N \leq 2^3$

→ 3 flip flops are needed

→ 0 to 7 numbers it means 8 states

→ we need 0 to 5 states it means 6 states.

After clock pulse	state Q_3 Q_2 Q_1	R
0	0 0 0	0
1	0 0 1	0
2	0 1 0	0
3	0 1 1	0
4	1 0 0	0
5	1 0 1	0
6	1 1 0	1
7	1 1 1	X

Q_3 \ Q_2Q_1	00	01	11	10
0	0	0	0	0
1	0	0	X	1

$Q_3\ Q_2\ Q_1$

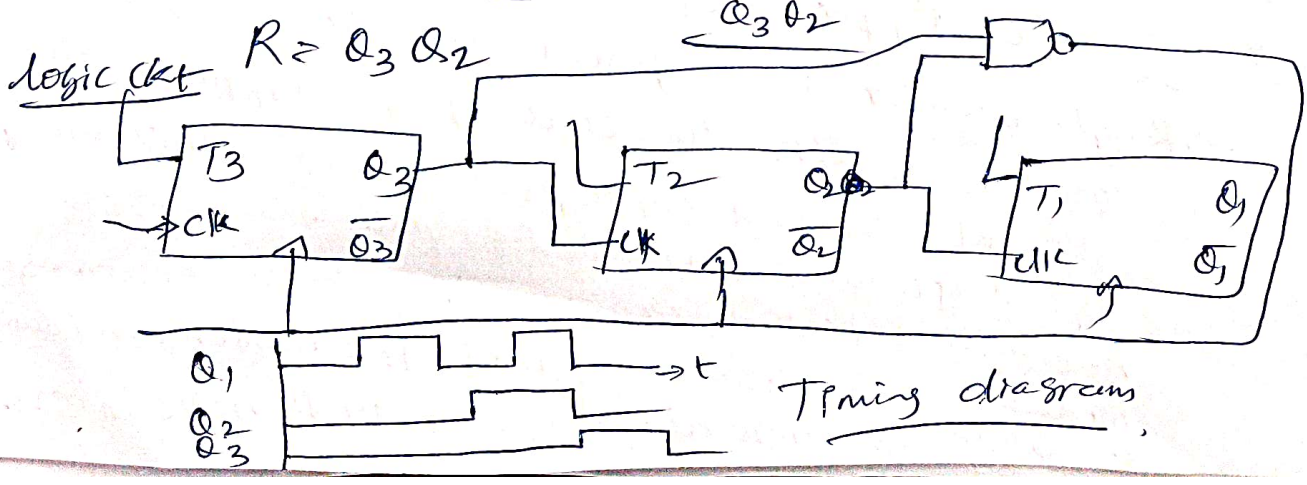
1 1 X

1 0

$Q_3 Q_2$

$R = Q_3 Q_2$

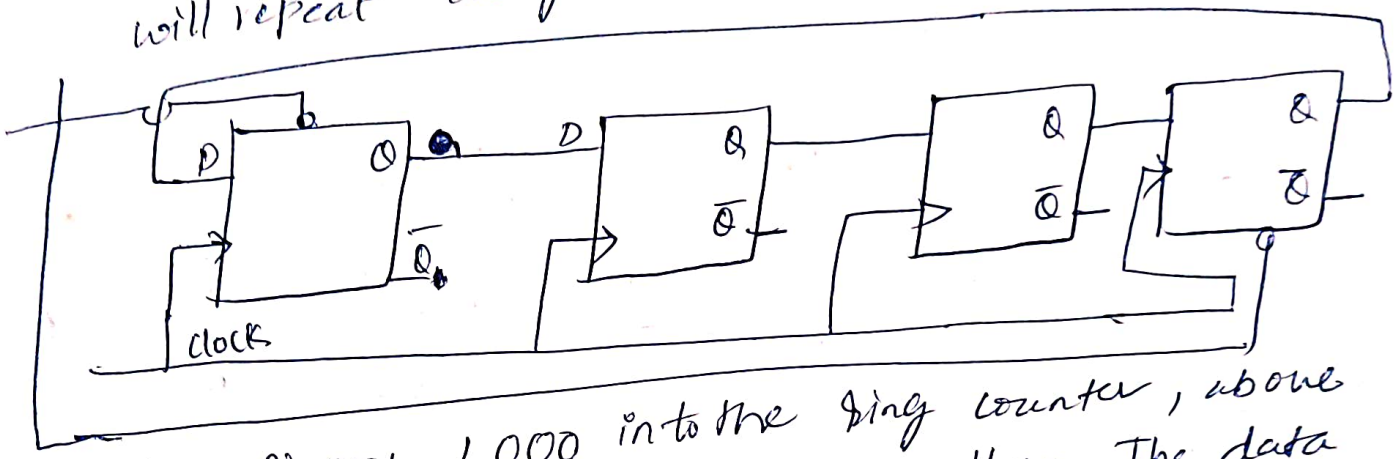
logic ckt



Timing diagrams.

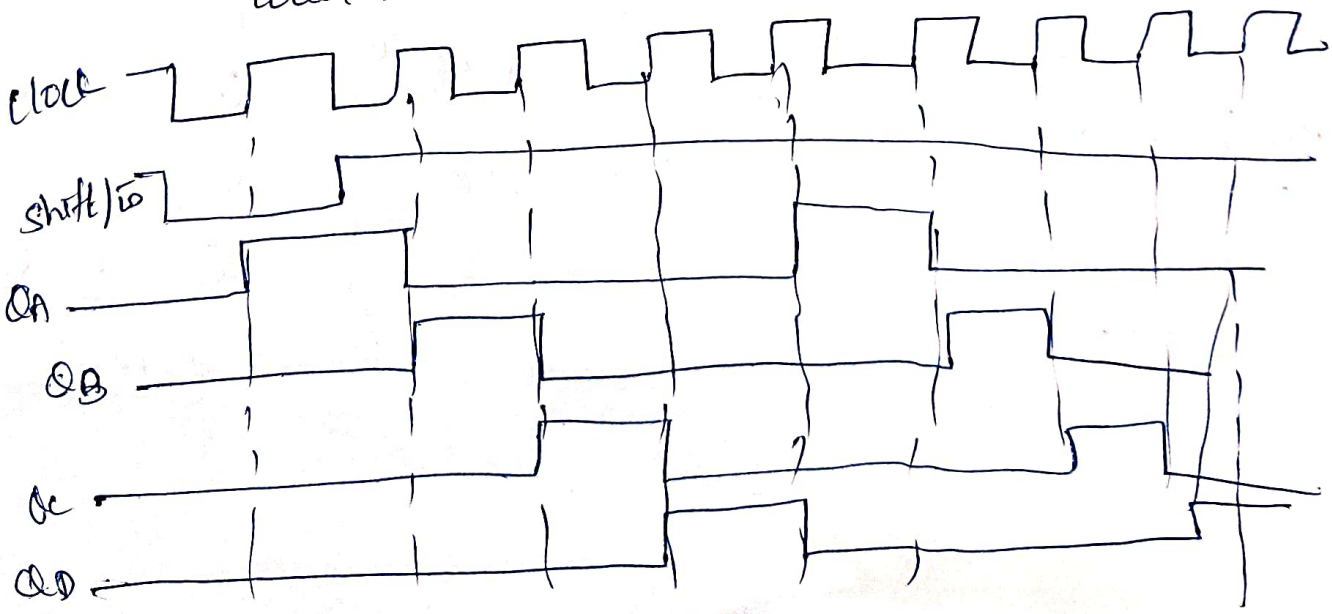
Ring counter

→ If the o/p of a shift register is fed back to the input, a ring counter results the data pattern contained with the shift register will recirculate as long as clock pulses are applied. For example, the data pattern will repeat every four clock pulses in the figure below



→ loading Binary 1000 into the ring counter, above prior to shifting yeilds a variable pattern. The data pattern for a single stage repeats every four clock pulses in our 4-stage examples. The waveform for all four stages look the same, except for the one clock time delay from one stage to the next See figure below.

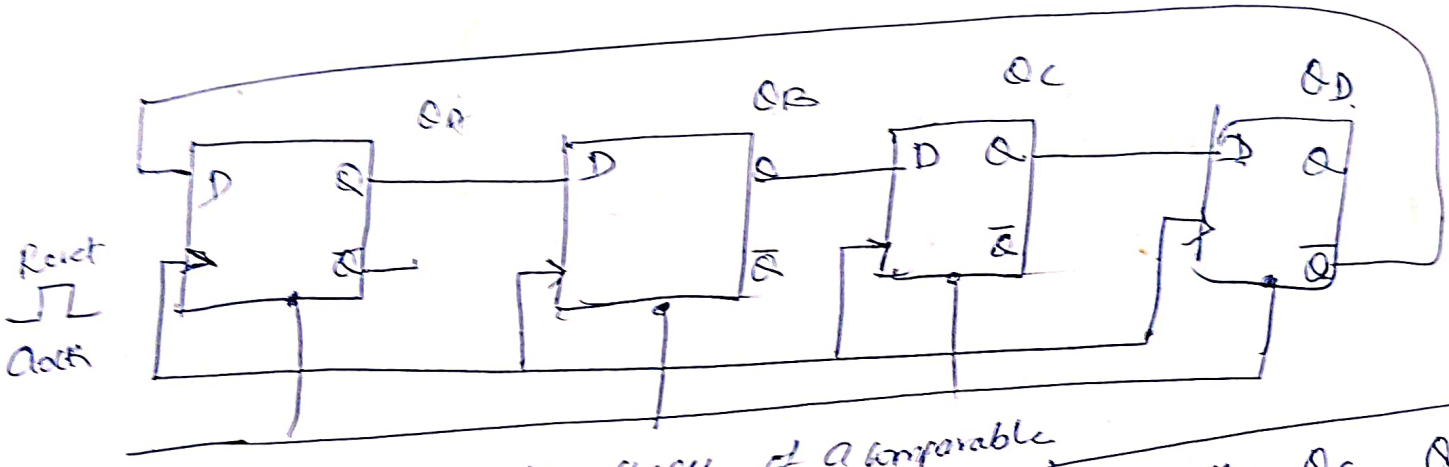
load 1000 into 4-stage ring counter to shift



Timing diagram

Johnson Counter

The Switch-tail ring counter, also known as the Johnson counter, overcomes some of the limitations of the ring counter. Like a ring counter a Johnson counter is a shift register fed back on itself.



* It requires half the stages of a comparable ring counter for a given division ratio. If the complement o/p of a ring counter is fed back to the input instead of the true output, a Johnson counter results. The difference between a ring counter & a Johnson counter is, which o/p of the last stage is feedback (Q (or) Q'). Carefully compare the feedback connection below to the previous ring counter.

Q_A	Q_B	Q_C	Q_D
0	0	0	0
1	0	0	0
1	1	0	0
1	1	1	0
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	1
Repeat			