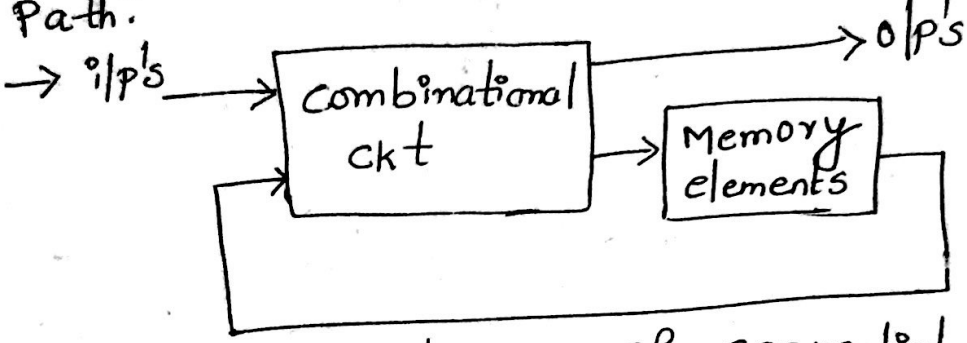
Analysis and Synthesis of Sequential circuits:-

Sequential circuits:- → In case of sequential circuits the output will depends on the present input and as well as past output.

→ It consists of a combinational circuit to which storage elements are connected to form a feedback path.



Block diagram of sequential circuit

→ The binary information stored in these elements at any given time defines the state of the sequential circuit at that time: present state.

→ **Next state:-** The present state that is fedback via memory element and external i/p's given to a combinational ckt will determines the output and this will become the next state of sequential circuit.

Present state			Next state		
A_n	B_n	C_n	A_{n+1}	B_{n+1}	C_{n+1}
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	0		

Differences between combinational and sequential circuits:-

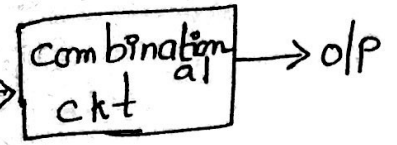
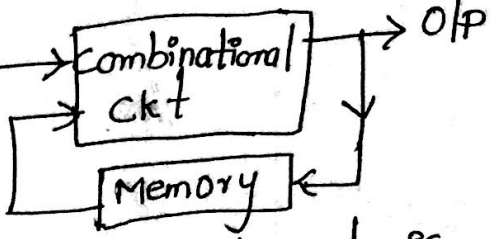
→ The behavior of an asynchronous Sequential circuit depends upon the input signals at any instant of time and the order in which the inputs change. [storage elements: time delay devices]

Differences between Synchronous and Asynchronous

Synchronous	Asynchronous
1. The i/p signal can affect the memory elements upon the activation of the clock pulse.	1. The change in the i/p signal can affect memory element at any instant of time.
2. Memory elements: clocked Flipflops.	2. Memory elements: unclocked Flipflops (or) time delay elements.
3. Synchronization is employed with the help of clock pulse.	3. There is no synchronization, hence it is a combinational ckt with feedback.
4. The maximum operating speed of clock pulses depends on the time delays (or) propagation delay of a circuit.	4. Because of absence of clock pulse, asynchronous ckts are operated faster than the synchronous ckt.
5. Easy to design	5. More difficult to design.
6. overcome instability ∴ time is easily broken down into independent discrete so separately.	6. Because of feedback among Logic gates it becomes unstable at times

Synchronous clocked sequential circuit:-

→ A Synch seq ckt uses signals that affect the storage elements at only discrete instants of time.

Combinational ckt	Sequential ckt
1. The outputs will depends upon the present inputs	1. The outputs will depends upon the present input and past output.
2. 	2. 
3. Memory element is not present.	3. Memory element is present.
4. Combinational ckts are easy to design but require more hardware.	4. Sequential circuit's are difficult to design but requires less hardware.
5. CCkt's are faster in speed because there is no memory element.	5. Due to the presence of memory element sequential circuit works with less speed.
6. More expensive ckt.	6. cheaper
7. Ex:- parallel adder	7. Ex:- Serial adder.
8. Less flexibility o/p depends on p/p only.	8. More flexibile o/p depends on p/p & past o/p

Classification of sequential circuits:-

→ Here classification of Sequential circuits is a function of the timing of their signals.

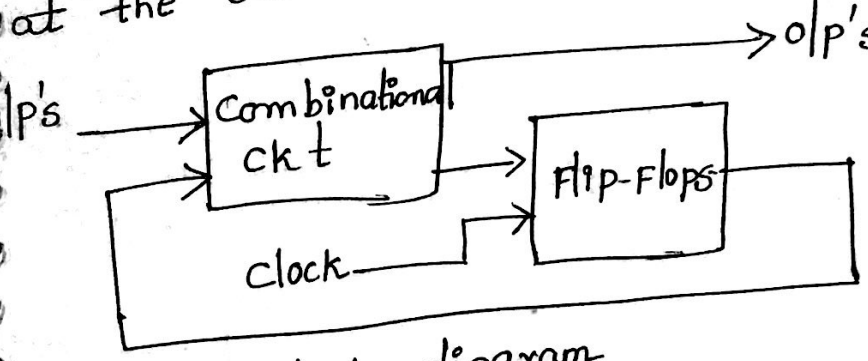
→ 2 types.

→ A Synchronous Sequential circuit, is a system whose behavior can be defined from the knowledge of its signals at discrete instants of time.

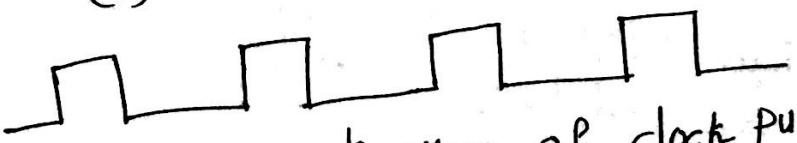
→ Synchronization is achieved by a timing device called a clock generator, which provides a clock signal having the form of a periodic train of clock pulses. (clk)

→ The clock pulses are distributed throughout the system in such a way that storage elements are affected only with the arrival of each pulse.

→ For eg, a ckt that is to add and store two binary numbers would compute their sum from the values of the numbers and store the sum at the occurence of a clock pulse.



(a) Block diagram



(b) Timing diagram of clock pulses.

Asynch ckt + Logic gates [feedback]

		NAND
0	0	1
0	1	1
1	0	1
1	1	0

Synchronous clocked sequential circuits-

→ synchronous sequential circuits that use clock pulses to control storage elements are called as clocked sequential circuits.

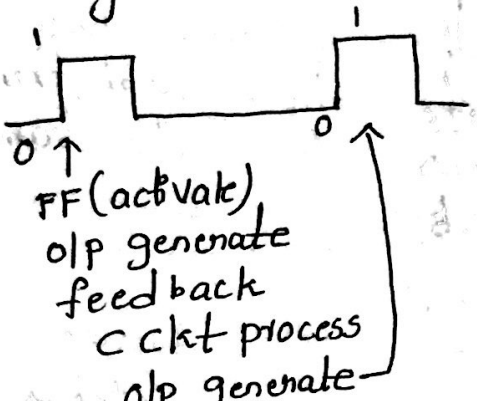
→ The storage elements (memory) used in clocked sequential circuits are called as flipflops.

→ A FF is a binary storage device capable

of storing one bit of information.

→ A sequential circuit may use many flip-flops to store as many bits as necessary.

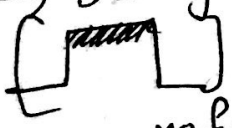
→ A FF will not respond to a o/p's of combinational ckt until a clock pulse is changed from 0 to 1.

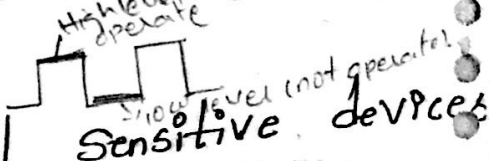


→ FF / Bistable multivibrator
2 states
Logic 0 & Logic 1.

Latches :- → A storage element in a digital circuit that can maintain a binary state as long as power is delivered to the circuit. indefinitely.

→ storage elements that operate with signal levels rather than signal transitions are referred to as latches.





→ Latches are said to be level sensitive devices FF's are edge-sensitive devices.

→ Latches are the basic circuits from which all flip-flops are constructed. Latches are used for the design of asynchronous sequential circuits.

1) SR Latch :-

→ The SR latch is a circuit with two cross-coupled NOR gates (or) two cross-coupled NAND

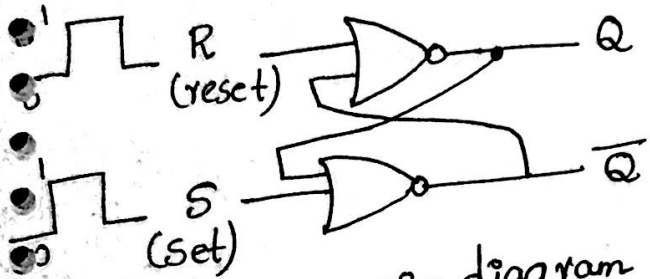
gates as S = Set, R = Reset

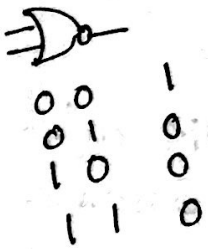
→ The Latch has two useful states. when $Q=1$, $\bar{Q}=0$ the latch is said to be in set state.

o/p $Q=0$, $\bar{Q}=1$ → reset state.

→ Q, $\bar{Q}$ are - complement of each other.

→ when both i/p's are 1 the o/p's are 0's known as unpredictable / undefined / metastable state.



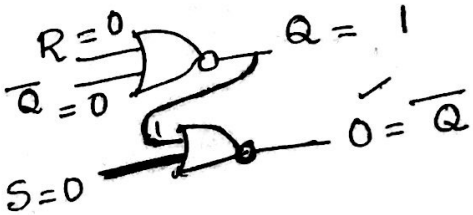


(a) Logic diagram

(a). Let $S=1$, $R=0$, (assume $\bar{Q}=0$)

$\Rightarrow Q = 1$.

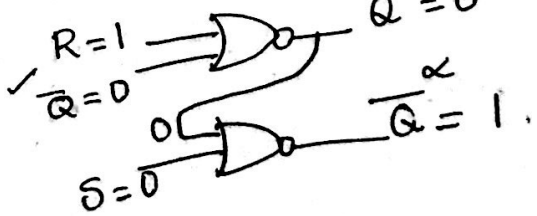
(b). $S=0$, $R=0$



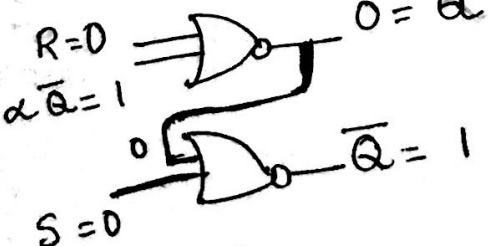
(No change)

— = 1st step
— = 2nd step.

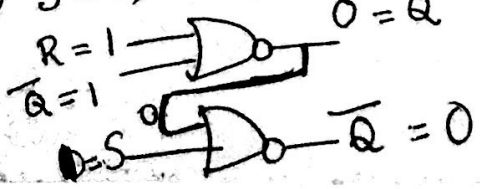
(c). $S=0$, $R=1$



(d). $S=0$, $R=0$



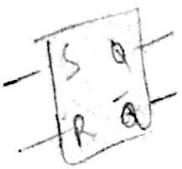
(e) $S=1$, $R=1$

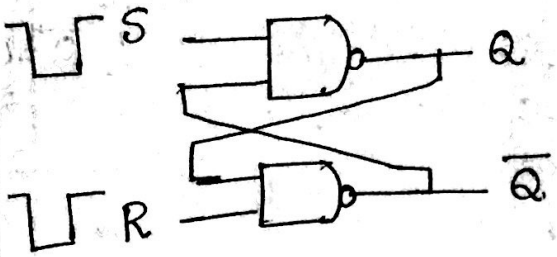


not possible

(b) Function table.

S	R	Q	$\bar{Q}$	
1	0	1	0	set
0	0	1	0	(No change)
0	1	0	1	Reset
0	0	0	1	(No change)
1	1	0	0	[Not Possible] (or) undefined. forbidden





SR Latch with NAND

S	R	Q	$\bar{Q}$	
1	0	0	1	
1	1	0	1	(after S=1, R=0)
0	1	1	0	
1	1	1	0	(after S=0, R=1)
0	0	1	1	X

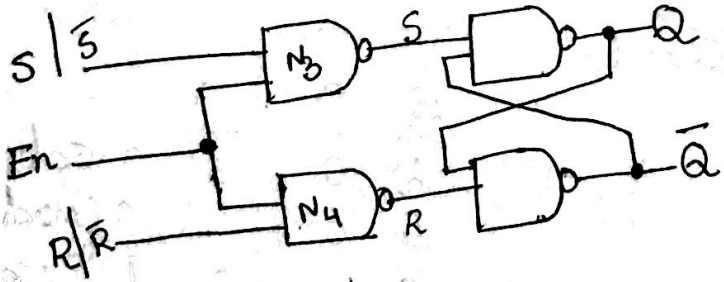
→ In comparing the NAND with the NOR Latch, note that the i/p signals for the NAND require the complement of those values used for the NOR latch.

→ Because the NAND latch requires a '0' signal to change its state, → 'SR' Latch

→ The primes (bars over the letters) designate the fact that the inputs must be in their complement form to activate the circuit.

→ An SR latch with a control input is shown below. It consists of the basic SR latch and two additional NAND gates.

→ The two gates are called as (N3, N4) steering / control gates because they are used to control the outputs



Logic diagram

En	$\bar{S}$	$\bar{R}$	Next state of Q
0	x	x	No change
1	0	0	No change
1	0	1	Q=0; reset
1	1	0	Q=1; set
1	1	1	Indeterminate

→ The control input En acts as an enable signal for the other two inputs.

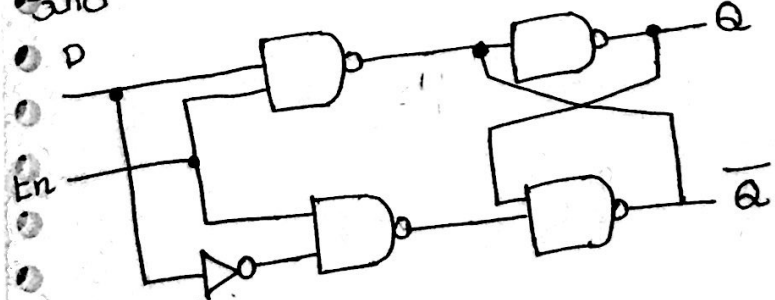
→ The outputs of the NAND gates stay at the

logic-1 level as long as the enable signal remains
at 0. → quiescent condition for the SR latch.
When the enable input goes to 1, information from
the S (or) R input is allowed to affect the
latch.
When En returns to 0, the circuit remains in its
current state. The control input disables the circuit
by applying 0 to En, so that the state of the
output doesnot changes regardless of the values
of S and R.


: Latch

D (Data) Latch (or) Transparent Latch :-

→ one way to eliminate the undesirable condition
of the indeterminate state in the SR latch is to
ensure that inputs S and R are never equal to
at the same time.
→ This is done in 'D' Latch, it has only two inputs D
and En. The D input goes directly to the S input,
and its complement is applied to the R input.

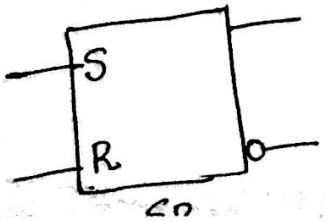


↓ Logic diagram.

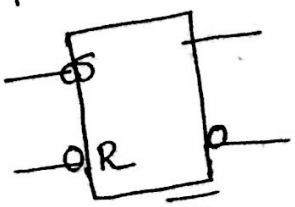
En	D	Next state of Q
0	X	No change
1	0	0 → reset
1	1	1 → set.

(1) En=1, D=0, $\overline{Q_n}$ = 1

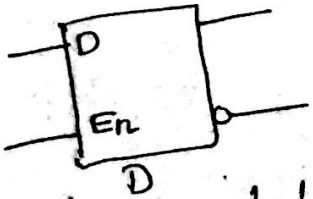
NAND also
→ Next state of Q | $Q_{n+1} = D$ is the characteristic eqn.



En



$\overline{S}$ $\overline{R}$

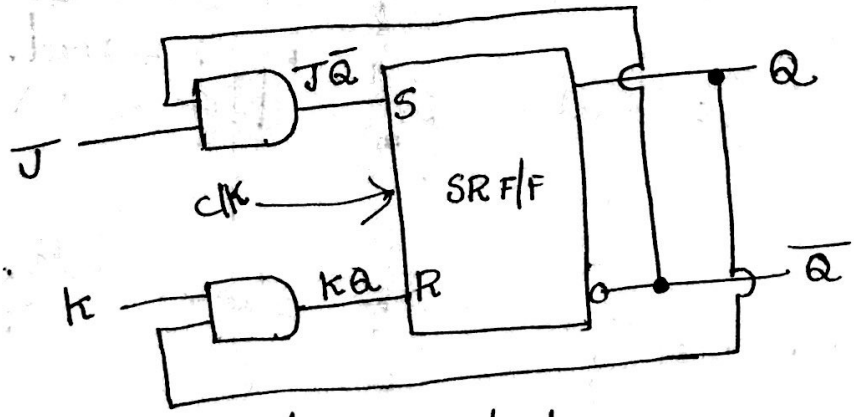


Graphic symbols

→ The D latch receives that designation from its ability to hold data in its internal storage.
→ It is used as a temporary storage for binary information between a unit and its environment, so it is called a transparent latch.

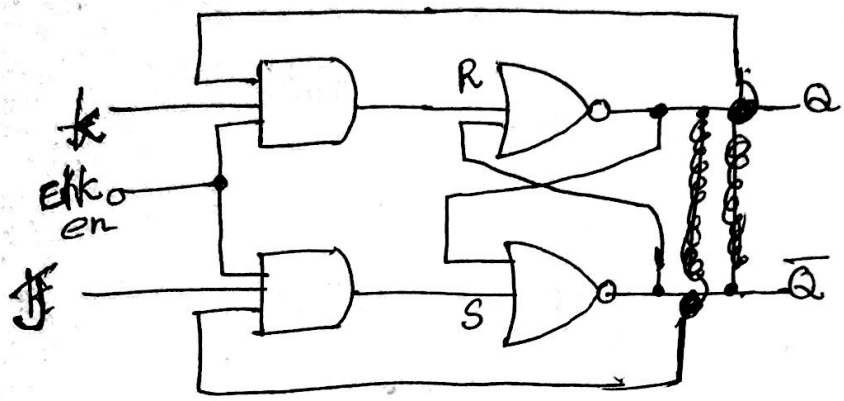
Flip-Flops :-

Note :- 1) Latch responds only at positive levels.
2) FF responds at edge triggerings.
→ In order to overcome the indeterminate state in SR F/F we change a little modification to a ckt which results in J-k FF.
→ Here the input signal doesnot go to the S input of the SR FF directly but it is gated via an AND gate.
→ The J input of the AND gate is one of the input lines of the new FF. The other i/p of the AND gate is the $\bar{Q}$ output of the Flipflop. Similarly the new i/p line k is created via AND gate.



J/k F/F ckt obtained by modifying the S/R F/F ckt.

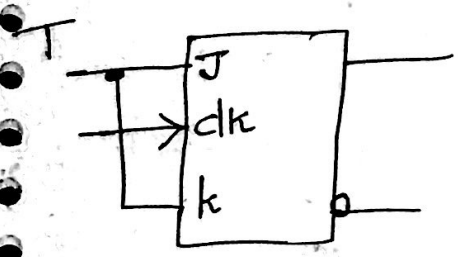
→ It can now be seen that the feedback to one of the AND gates is Q, whereas the feedback to the other AND gate is $\bar{Q}$. (∵ Q & $\bar{Q}$ are complement to each other) so, the two AND gates cannot produce the output 1 simultaneously



J	K	Q_{n+1}	
0	0	No change	
0	1	0	Reset
1	0	1	Set
1	1	$\bar{Q}$	comple

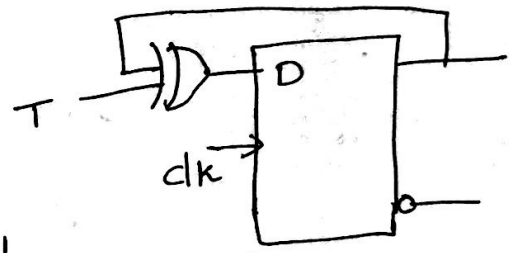
characteristic table.

T (Toggle F/F) :-



T=0, $\bar{Q}_n$=1

T F/F by D :-



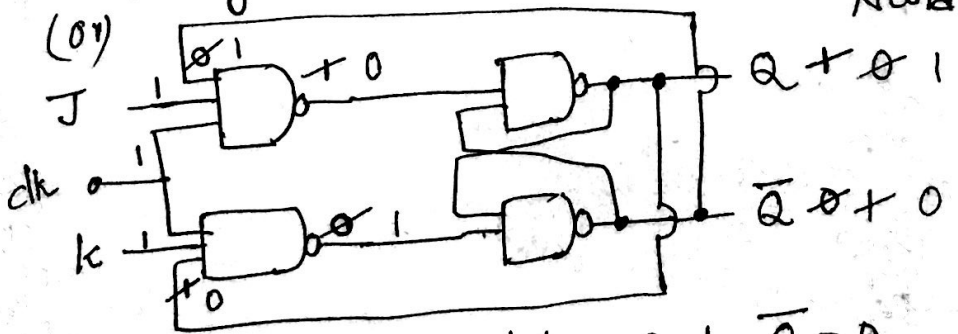
Func J K
0 0
0 1
1 0
1 1

↓ Later we discuss them in conversions.

→ Toggling ≡ complementing

Nand = complemented o/p

(or)



1) clk=1, J=1, k=0, no 3rd i/p, Q=1, $\bar{Q}$ = 0
2) clk=1, J=0, k=1, Q=0, $\bar{Q}$=1
3) clk =1, J=1, k=1, assume Q=1, $\bar{Q}$=0

Flip Flop characteristic Tables :-

J	K	Q_{n+1}
0	0	No change
0	1	0 reset
1	0	1 set
1	1	$\overline{Q_n}$ complement

R	S	Q_{n+1}
0	0	No change
0	1	1 set
1	0	0 reset
1	1	indeterminate

D	Q_{n+1}
0	0
1	1

T	Q_{n+1}
0	Q(t) No change
1	$\overline{Q}$(t) complement

characteristic equations :-

1) $Q_{n+1} = D$ 2) $Q_{n+1} = T \oplus Q_n$

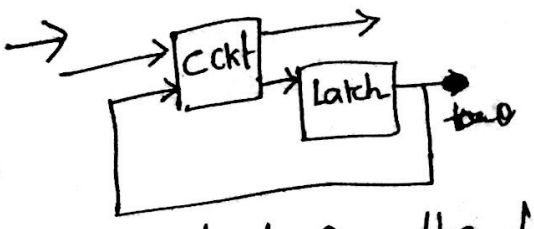
3) $Q_{n+1} = J\overline{Q_n} + \overline{K}Q_n$ 4) $Q_{n+1} = S + \overline{R}Q_n$

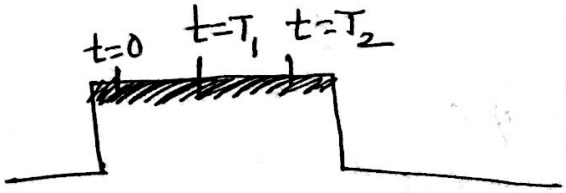
→ These eqn's are derived from characteristic tables and ckt dgm.

Race around condition :-

→ In an Jk Latch ckt, the o/p's are responds only to positive levels.

Response to positive level.





a) at t=0, the Latch takes i/p from cc & produces

o/p and again given to Latch at $t = T_1$

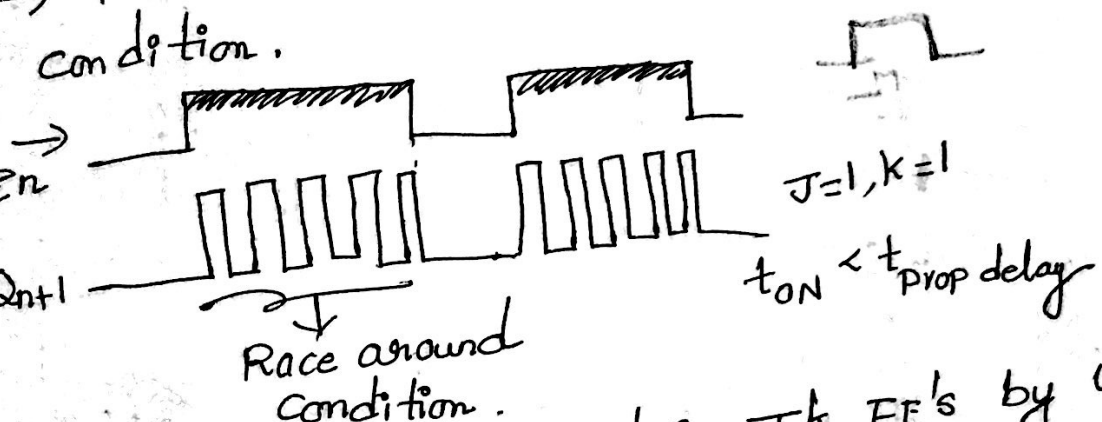
b) At $t = T_1$, again the Latch takes input from CCkt & produces o/p & again given to Latch at $t = T_2$

→ Like this manner o/p's of Latch changes as

$Q = 0 \to 1 \to 0 \to 1 - - -$ } unreliable values when $J = K = 1$.

$\bar{Q} = 1 \to 0 \to 1 \to 0 - - -$ }

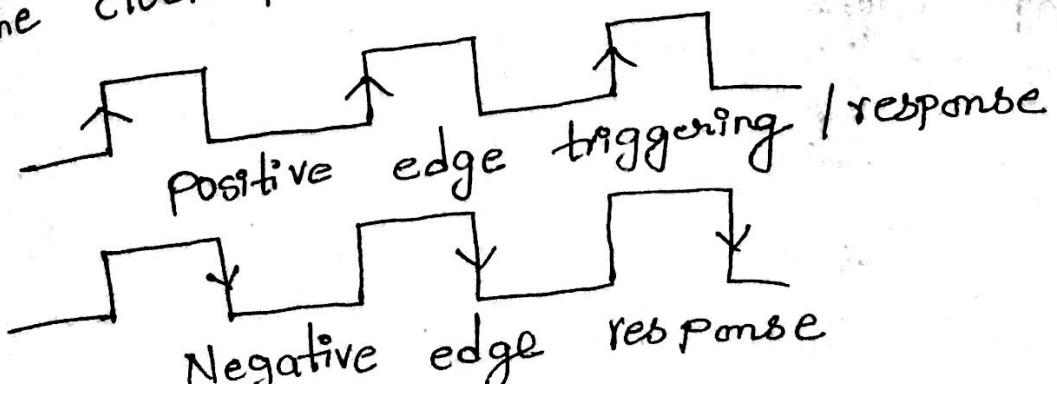
→ This is called as Toggling (or) race around condition.

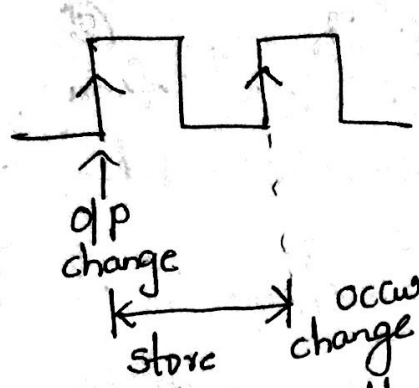


→ This can be overcomed in JK FF's by using 3 ways:

1) To employ 2 latches, in a Special configuration that separates the o/p of one FF and prevents it being affected while the i/p to the other FF is changing ie, master-slave JK FF.

2). Another way is to produce a FF that triggers only during a signal transition (from 0 to 1 (or) from 1 to 0) and disabled during the rest of the clock pulse.

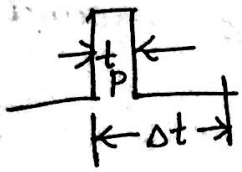


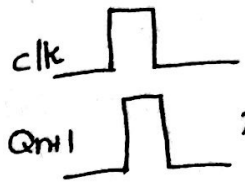


3) To maintain a condition
ie, pulse width ≤ propogation delay time.

→ propogation time is defined as the time taken by the circuit to produce the output for a given inputs.

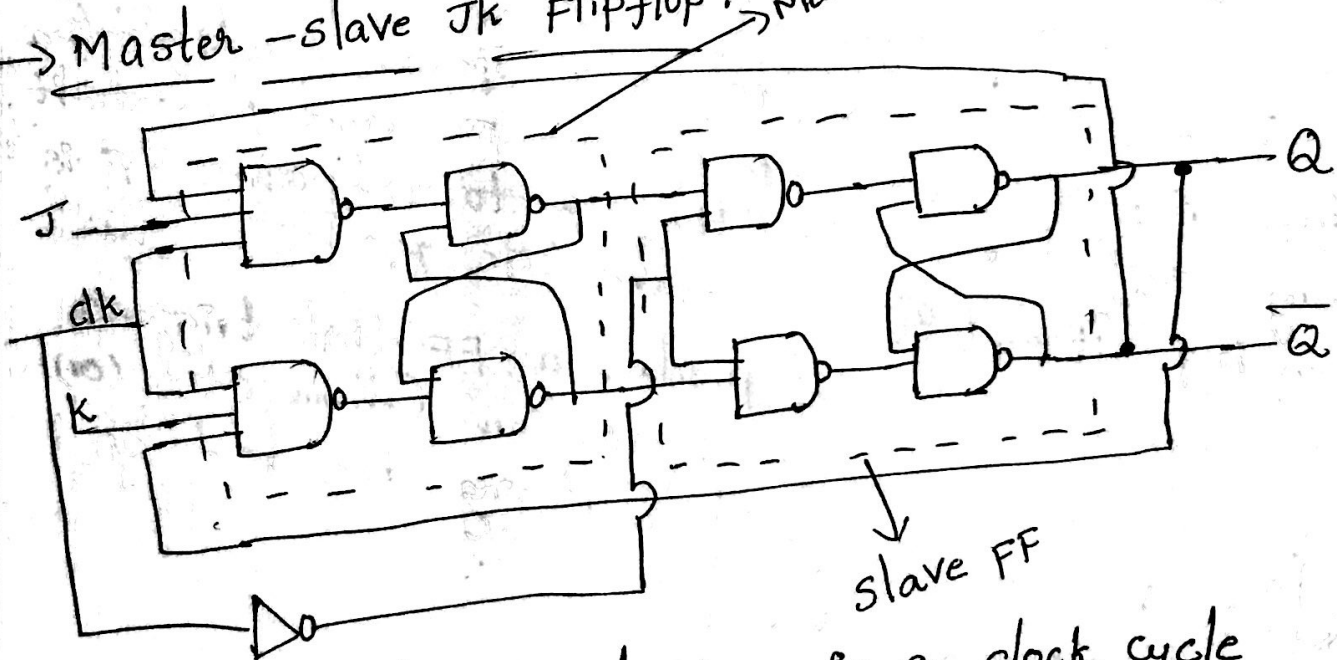
$t_p \leq \Delta t$



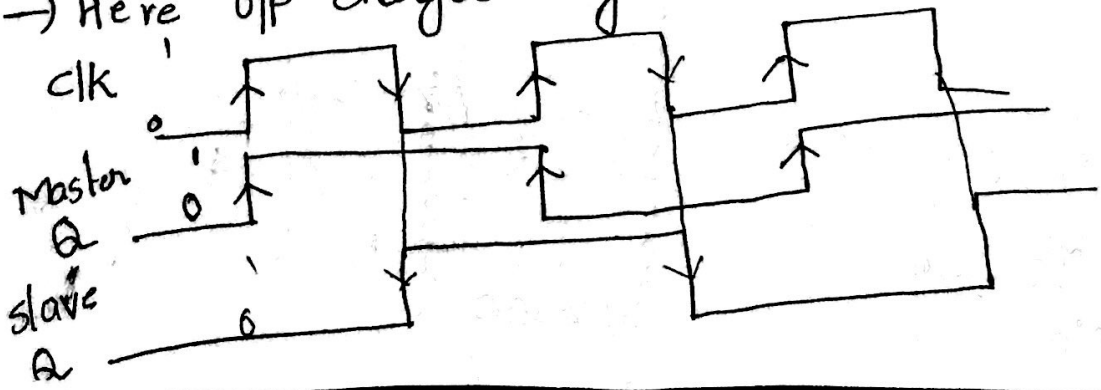


→ But it can't be used practically ∵ such a narrow pulse width cannot be generated (By ↑ freq) but it cannot be used practically due to internal delays of a circuit.

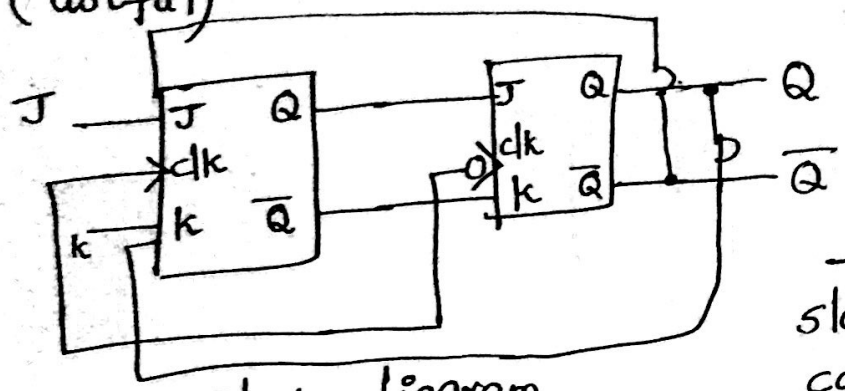
→ Master-slave JK Flipflop :-



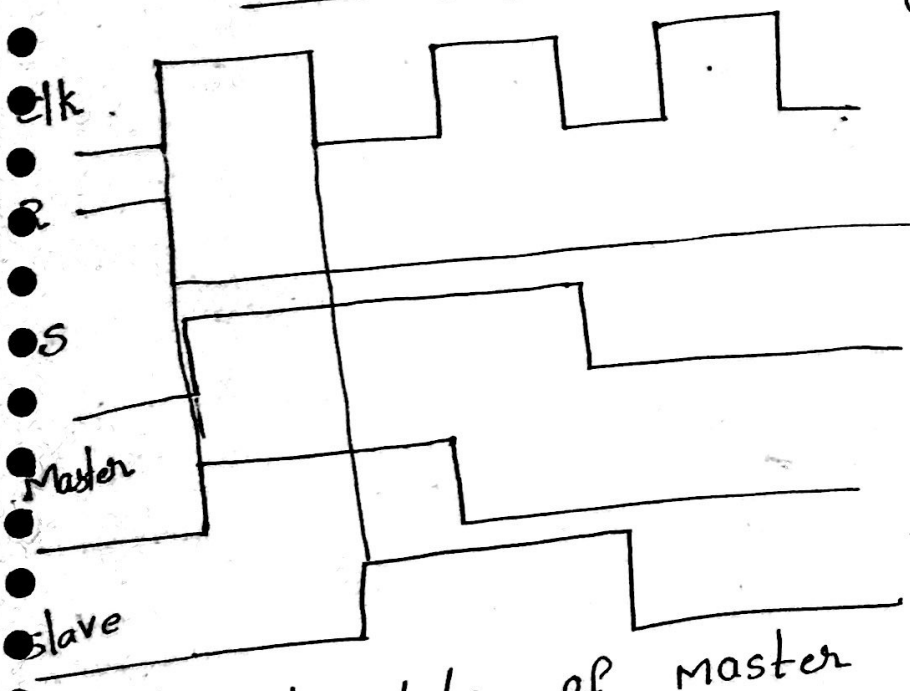
→ Here o/p changes only once in a clock cycle



→ constant toggling is not there, toggling is a controllable one where as racing uncontrolled change (undesirable).
(useful)



Block diagram



→ If clk=1, then slave clk=0, then we can't observe any o/p (∵ Master o/p is in locking condition).

→ So, clk=0, i.e, FF changes o/p only once so eliminating 'racing'.

→ J=K=1, master toggles on the ↓ & slave copies the o/p of master on ↓. Now, feedback i/p to the master FF are complemented but as it is ↓ master FF inactive. race around condition by FF is determined

→ The o/p state of master J and K inputs at the +ve edge clock pulse. The output state of master is then transferred as an input to the slave FF.

→ The slave FF uses this input at -ve clock pulse to determine its output state.

Conversions of Flip-flops:-

→ Inorder to do conversions we need excitation tables:

T FF :-

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

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

D FF :-

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

$Q_{n+1} = D$

SR FF :-

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

JK FF :-

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

→ Extables are obtained from characteristic eqn's:

$X = 0, 1$.

1) SR to T FF :-

available FF → T → driven element → S, R → Q = FF o/p.

Input T	Q_n	Q_{n+1}	FF i/p's S	R
0	0	0	0	X
1	0	1	1	0
1	1	0	0	1
0	1	1	X	0

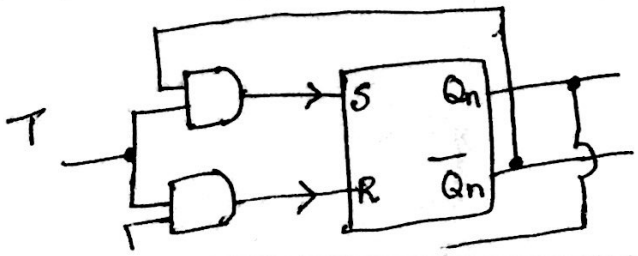
2nd 0 1st 3rd

T \ Q_n	0	1
0	0	X
1	(1)	0

$S = T\overline{Q_n}$

T \ Q_n	0	1
0	X	0
1	0	1

$R = T Q_n$

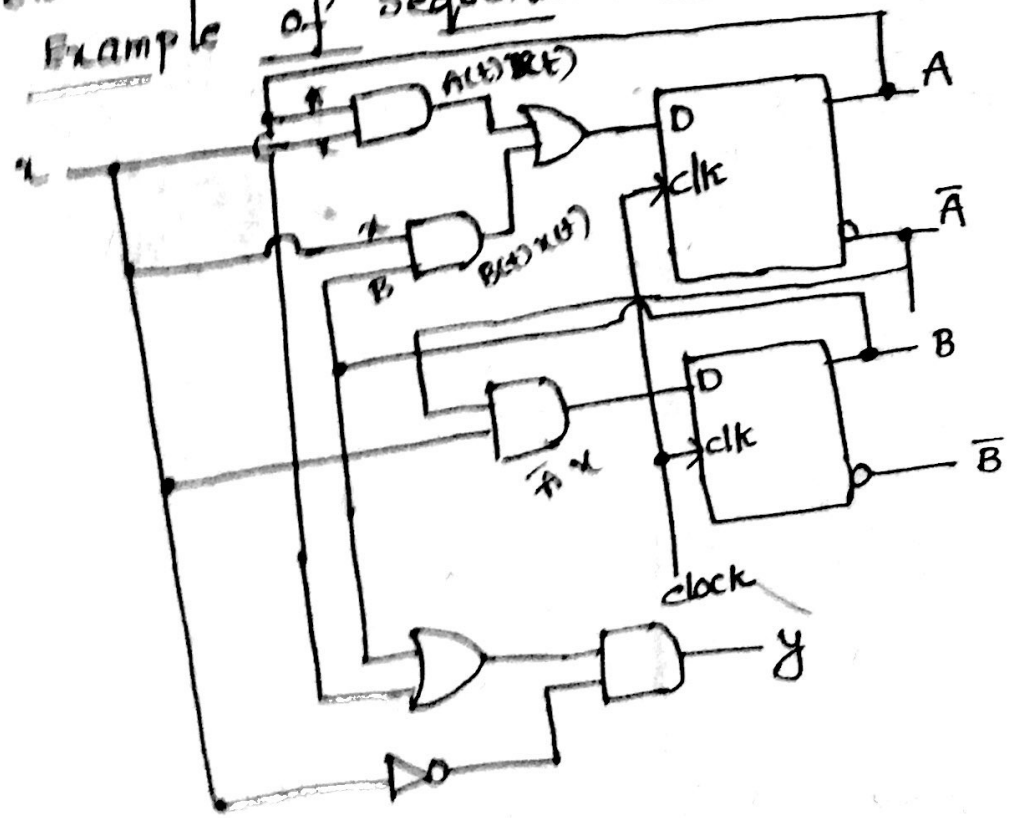


2) Jk to D FF 3) T to D FF
4) SR to Jk FF 4) Jk to T FF

Analysis of clocked Sequential circuits :-

→ Analysis describes what a given circuit will do under certain operating conditions.
→ The behavior of a clocked sequential circuit is determined from the inputs, outputs and the state of its flip-flops.
→ The outputs and the next state are both a function of the inputs and the present state.
→ The analysis of a sequential circuit consists of obtaining a table or a diagram.
→ A logic diagram is recognized as a clocked sequential circuit if it includes flip-flops with clock inputs.

Example of Sequential circuit :-



i/p's = A, B, x
o/p = y

State equations :- → describes behavior of a clocked sequential ckt

→ It is also called as transition equation specifies the next state as a function of present state & inputs. $A(t) = PS$, $A(t+1) = N.S$

i) $A(t+1) = A(t)\,x(t) + B(t)\,x(t)$

ii) $B(t+1) = \overline{A}(t).x(t)$

iii) $y(t) = [B(t) + A(t)]\,\overline{x}(t)$

State Table :-

→ The time sequence of inputs, outputs and flip-flop states can be enumerated in a state table. (transition table)

→ The table consists of PS, i/p's, NS & o/p.

State Table :-

Present state		input	Next state		output
A	B	x	A^+	B^+	Y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

BCD 1010 0001 0000

State diagram :- → The information available in a state table can be represented graphically in the

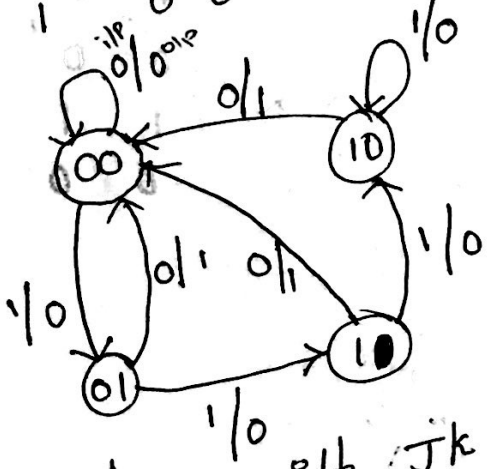
form of a state diagram.

→) In this type of diagram, a state is represented by a circle, and the (clock triggered) transitions between states are indicated by directed lines connecting the circles.

Reduced form of state Table:-

PS	NS		O/P	
	x=0	x=1	x=0	x=1
A B	A B	A B	Y	Y
0 0	0 0	0 1	0	0
0 1	0 0	1 1	1	0
1 0	0 0	1 0	1	0
1 1	0 0	1 0	1	0

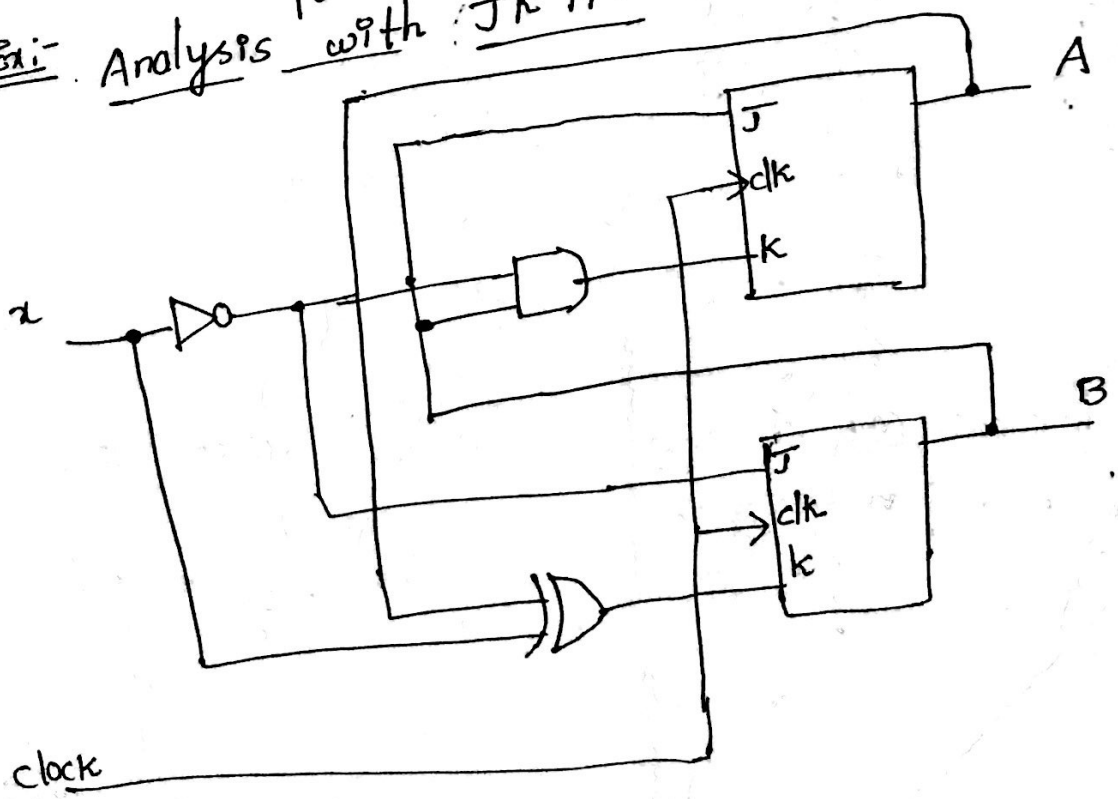
i/p | o/p



ckt dgm → State eqn's → State Table → State dgm.

modu-12

Ex:- Analysis with JK FF:-



1) State eqn's:-

i) $J_A = B$, $K_A = B\bar{x}$

ii) $J_B = \bar{x}$, $K_B = \bar{A}x + A\bar{x}$

$A(t+1) = J_A\bar{A} + \bar{K}_A A$

$B(t+1) = J_B\bar{B} + \bar{K}_B B$

$= \bar{A} \oplus x$

$A(t+1) = B\bar{A} + \bar{B}\bar{x}A$

2) State Table:-

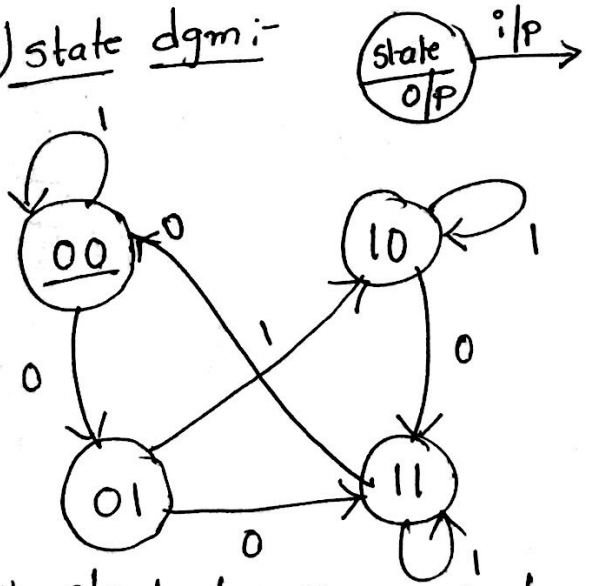
PS		i/p	NS		Flip Flop inputs			
			$A^+_{(t+1)}$	$B^+_{(t+1)}$				
A	B	x			J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

3rd 2nd

①st

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

3) state dgm:-

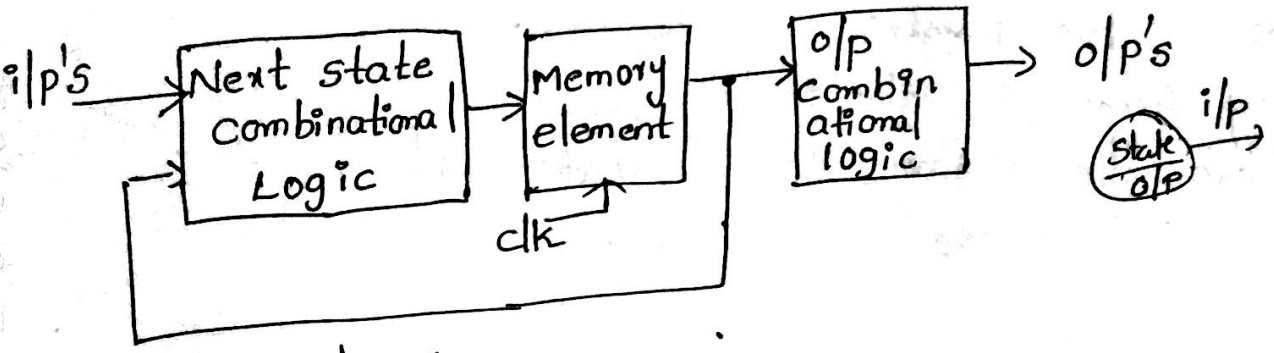


⇒ The synchronous (or) clocked sequential ckts are represented by two models;

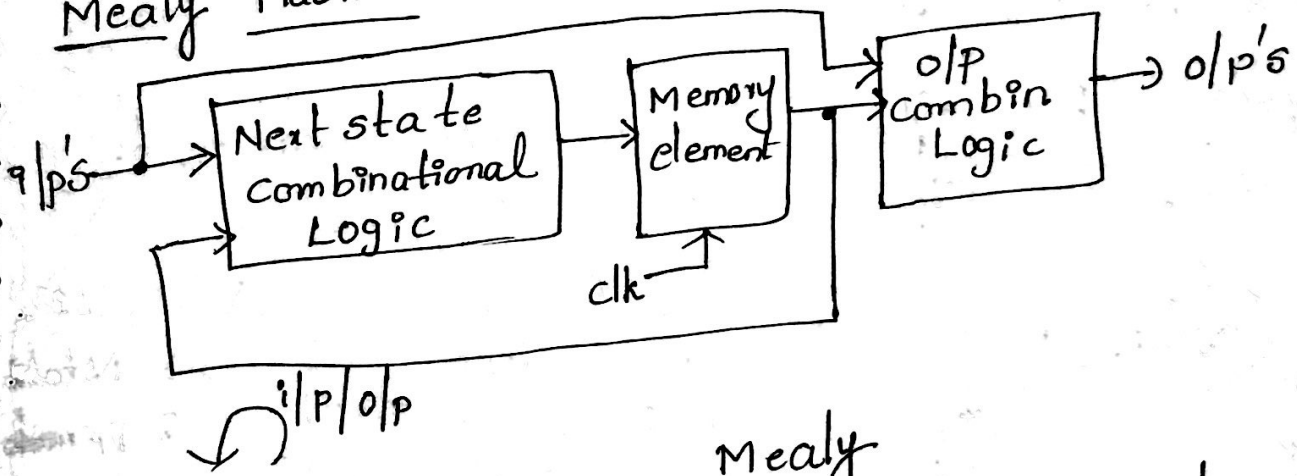
i) Moore Model:- The output depends only on the present states of the FF.

2) Mealy ckt :- The output depends on both the present state of the FF's and on the inputs.

Moore machine :-



Mealy Machine :-



Moore

→ It's output is a function of present state only

→ I/p changes doesn't affect the output.

→ This circuit requires more no. of states for implementing the functions

→ $z(t) = \lambda\{s(t)\}$ (o/p, ps)

Mealy

→ It's o/p is a function of present state as well as present i/p.

→ I/p changes may affect the output of the circuit

→ Less no. of states

→ $z(t) = \lambda\{s(t), x(t)\}$ (i/p)

synchronous sequential ckts

Design procedure :-

Steps :- 1) From the word description and specifications of the desired operation, derive a state diagram for the circuit.

2) Reduce the number of states if necessary & Assign binary values to the states

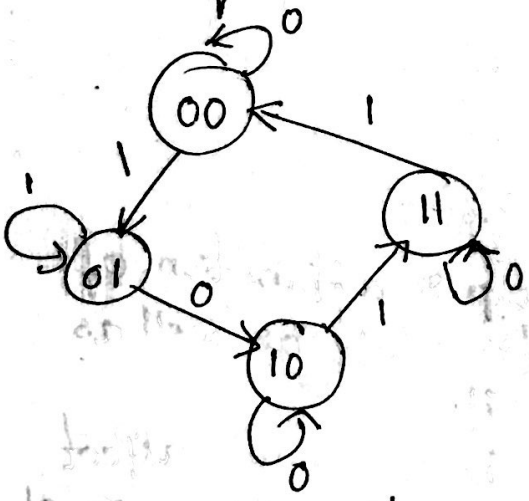
3) obtain the binary coded state table

4). choose the type of flip-flops to be used.

5) Derive the simplified FF i/p eqn's & o/p eqn's

6). Draw the logic diagram.

Example I :- Given the following state diagram, design the Sequential circuit using JK Flip-flops.



State Table

PS	NS	
A B	x=0	x=1
	$A^+ B^+$	$A^+ B^+$
00	00	01
01	10	01
10	10	11
11	11	00

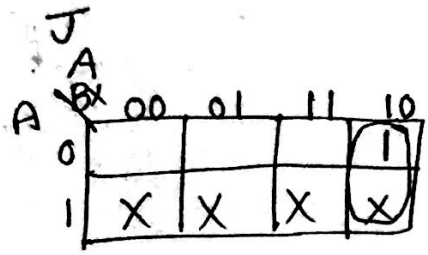
2^m = States.
2^2 = 4 states.
2 = FF needed

JK FF Excitation Table

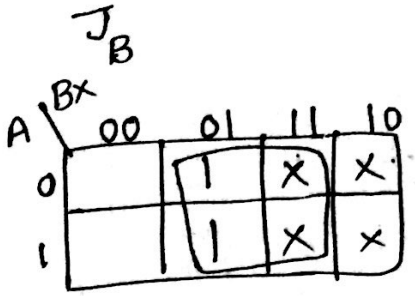
Q	Q^+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

PS		I/p	NS		FF i/p's			
A	B	x	A^+	B^+	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

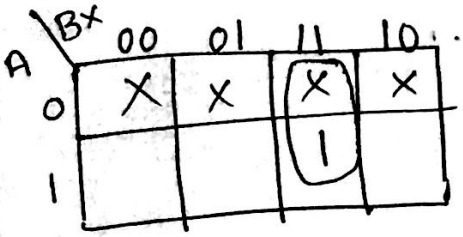
K-map :-



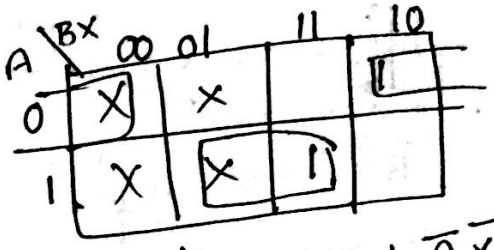
$J_A = B\bar{X}$



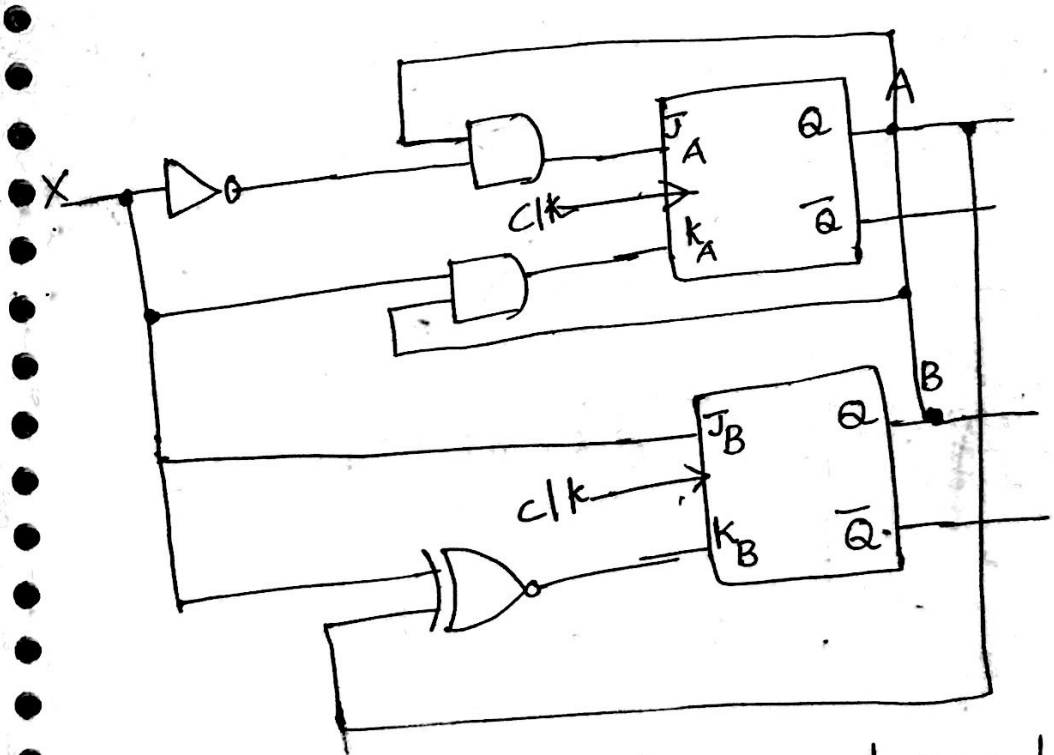
$J_B = X$



$K_A = B.X$



$K_B = AX + \bar{A}\bar{X} = A \odot X$

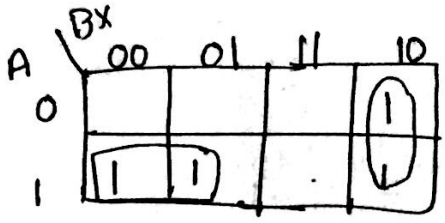


⇒ For the same state diagram design the sequential circuit using D FF's :-

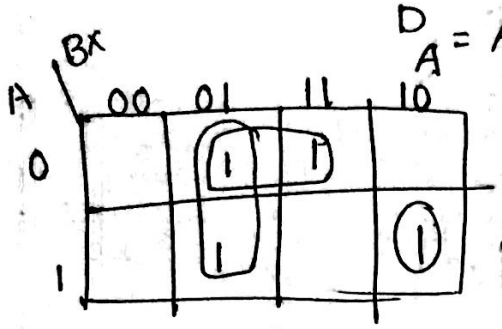
$\therefore Q_{n+1} = D$

PS		I/P	NS		O/P (extra)
A	B	X	A^+ D_A	B^+ D_B	Y
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	0	0
1	1	1	0	0	0

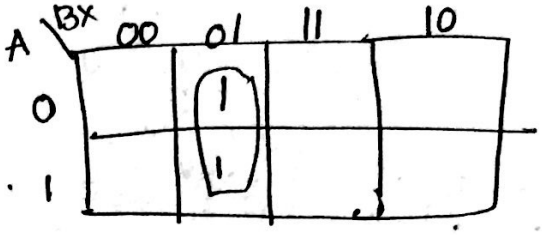
K-map :-



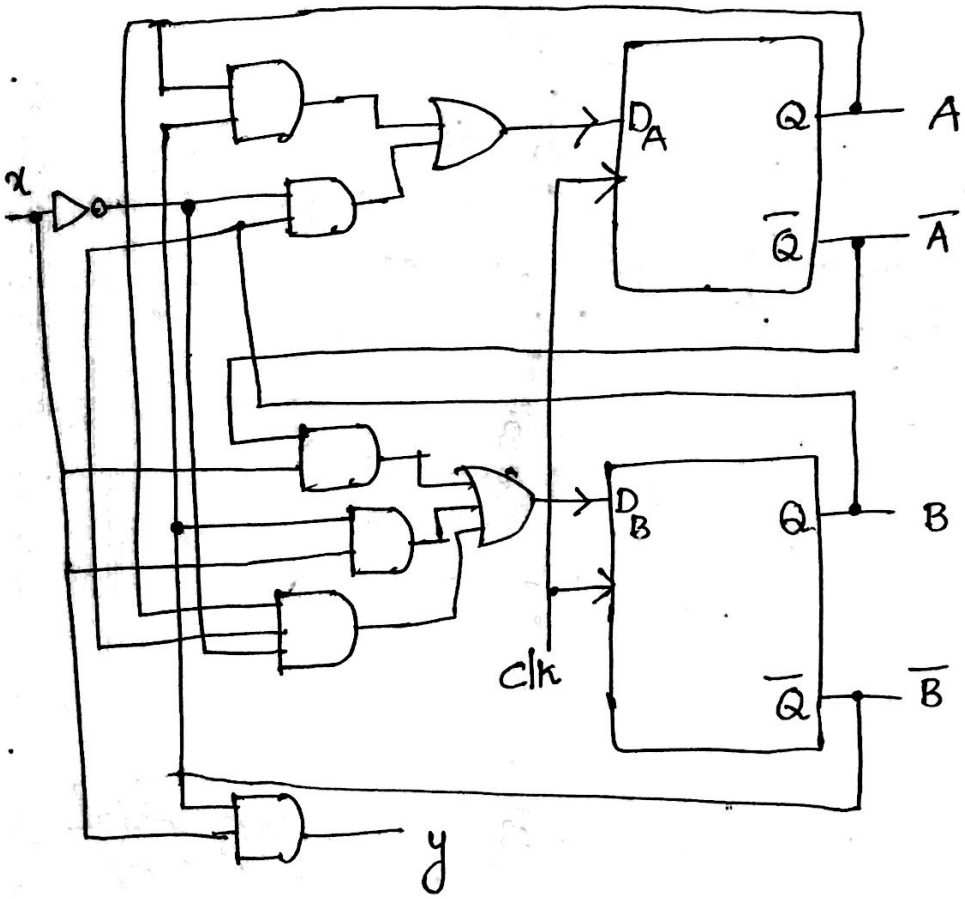
$D_A = A^+ = AB' + B.x'$



$D_B = B^+ = \bar{A}x + \bar{B}.x + A.B.\bar{x}$



$y = \bar{B}.x$



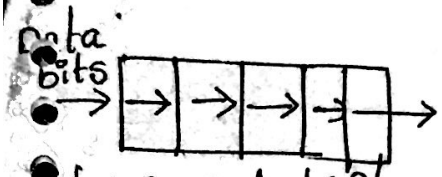
Registers :-

→ The group of flipflops used to store the information is called registers.

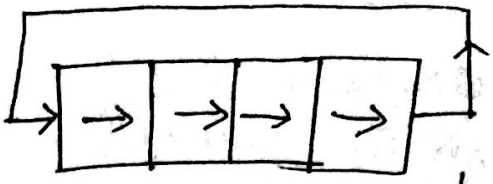
→ we can WRITE a data by using SI, PI

→ we can READ a data by using SO, PO

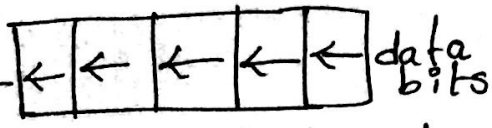
Shift registers:- Here there is an movable data



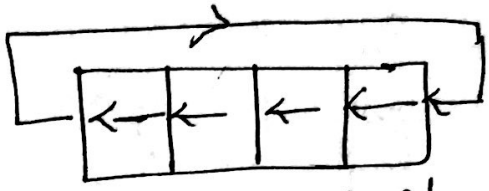
(a) Serial shift right than out



(b) Rotate right

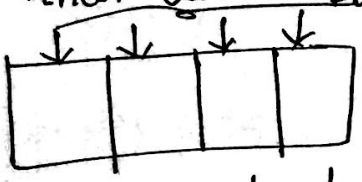


(c) Serial shift left

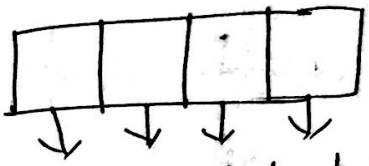


(d) Rotate Left

than out Databits



(e) Parallel shift in

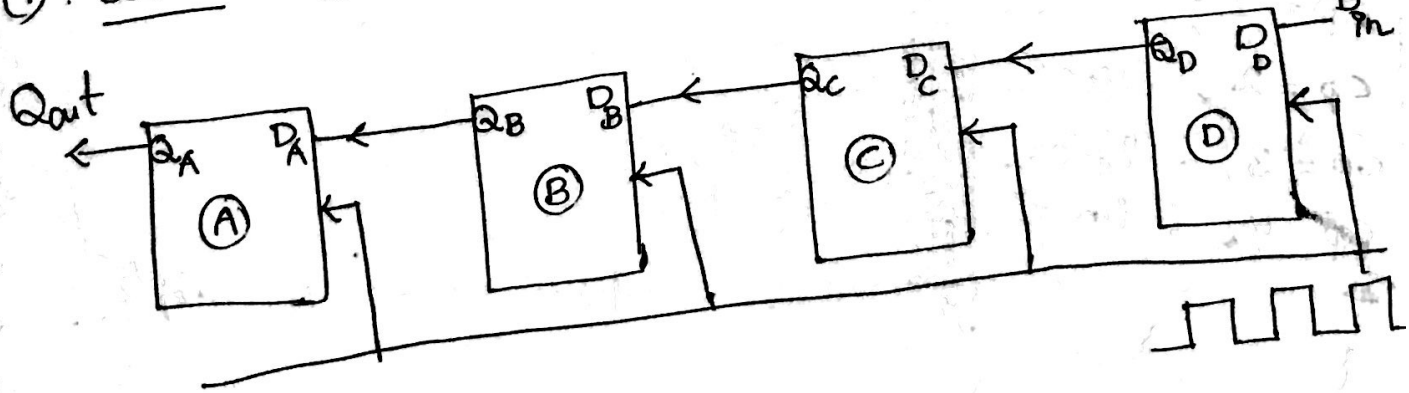


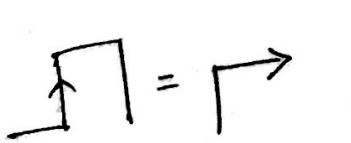
(f) parallel shift out

→ A register which is going to shift right, left, Sends and receives data serially and parallel is called universal shift registers.

→ To design shift registers SR, JK, D Flipflops are used but not T-FF due to its toggling nature

(1). Serial in Serial out (SISO) :-





$\therefore Q_{n+1} = D$

→ To send of 4 bit of information = 1111

1st clock pulse $D_{in} = 1$

$Cp = 1 \Rightarrow Q_A Q_B Q_C Q_D = 0001$

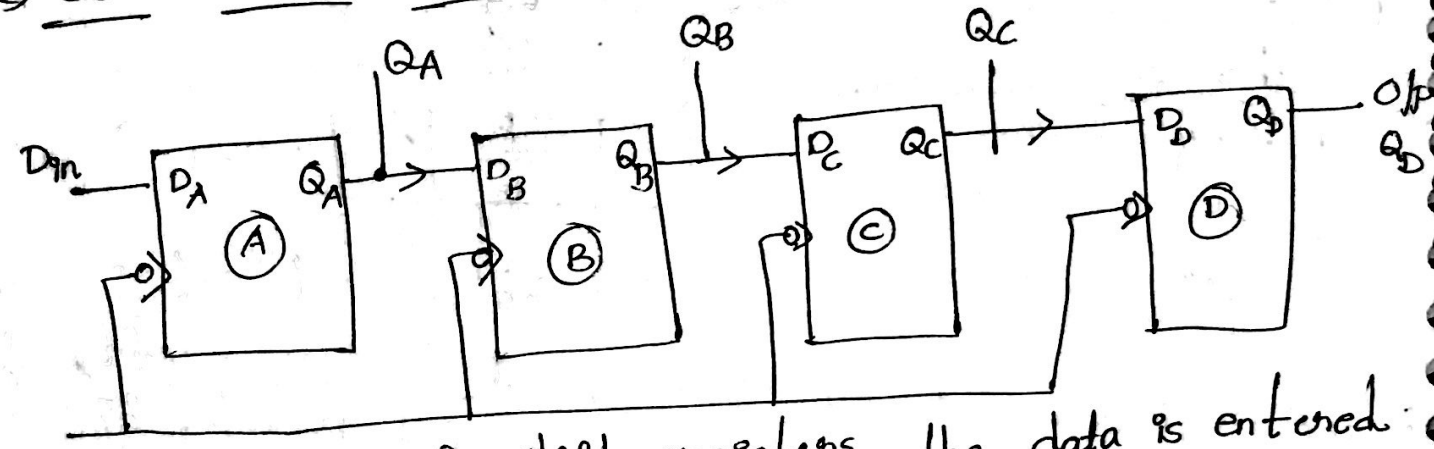
$Cp = 2 \Rightarrow Q_A Q_B Q_C Q_D = 0011$

$Cp = 3 \Rightarrow Q_A Q_B Q_C Q_D = 0111$

$Cp = 4 \Rightarrow Q_A Q_B Q_C Q_D = 1111$

→ Here Cp is serving (or) helpful to transfer the information.

(2) Serial in Parallel out :-



→ In this type of shift registers the data is entered same as the previous case and the collection of output is in parallel manner that means Q_A, Q_B, Q_C, Q_D are obtained simultaneously instead of bit-by-bit basis. (1000) 0001

$Cp = 1$, $Q_A = 0$

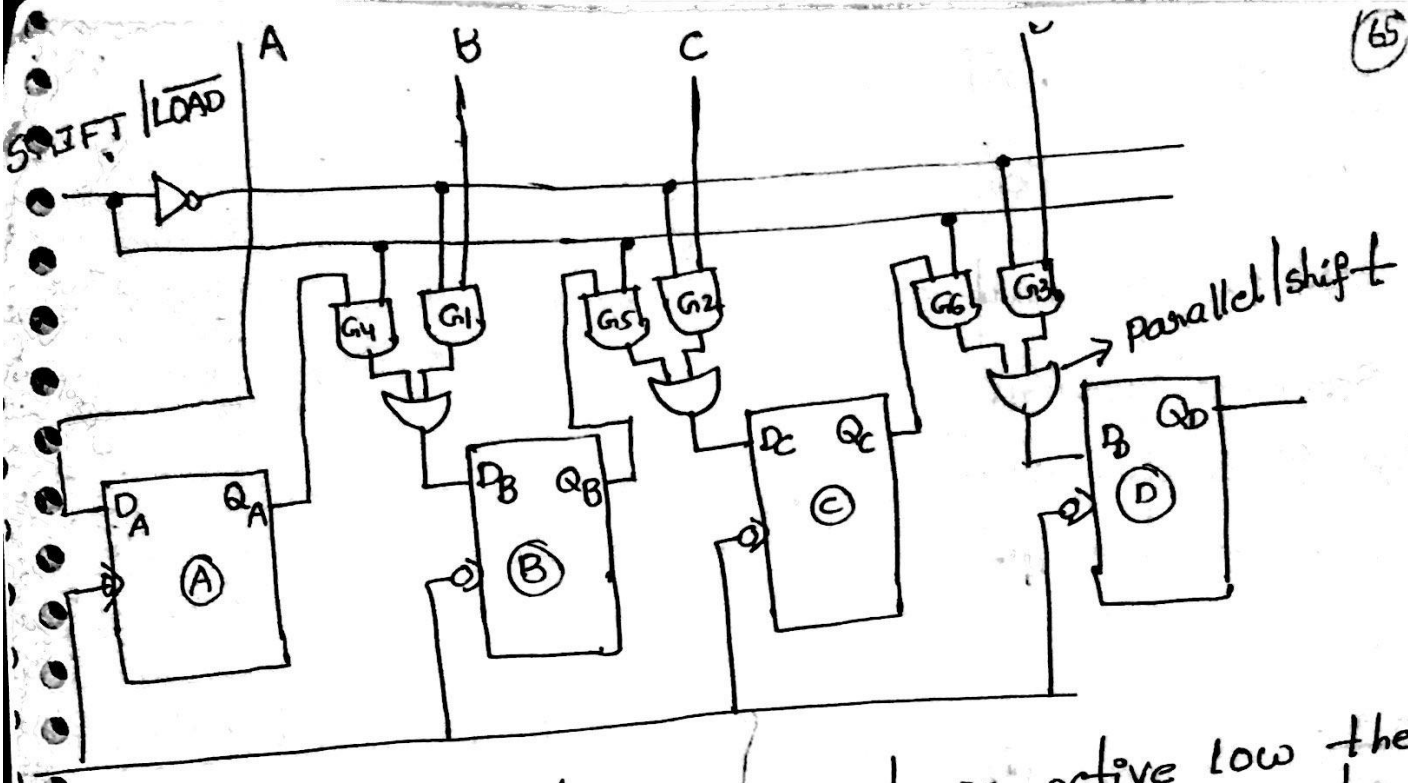
$cp = 2$, i/p at D_B = 0100, $Q_B = 0$

$Cp = 3$, i/p at D_C = 0010, $Q_C = 0$

$cp = 4$, i/p at D_D = 0001, $Q_D = 1$

→ we are getting output parallely as well as shifting is present.

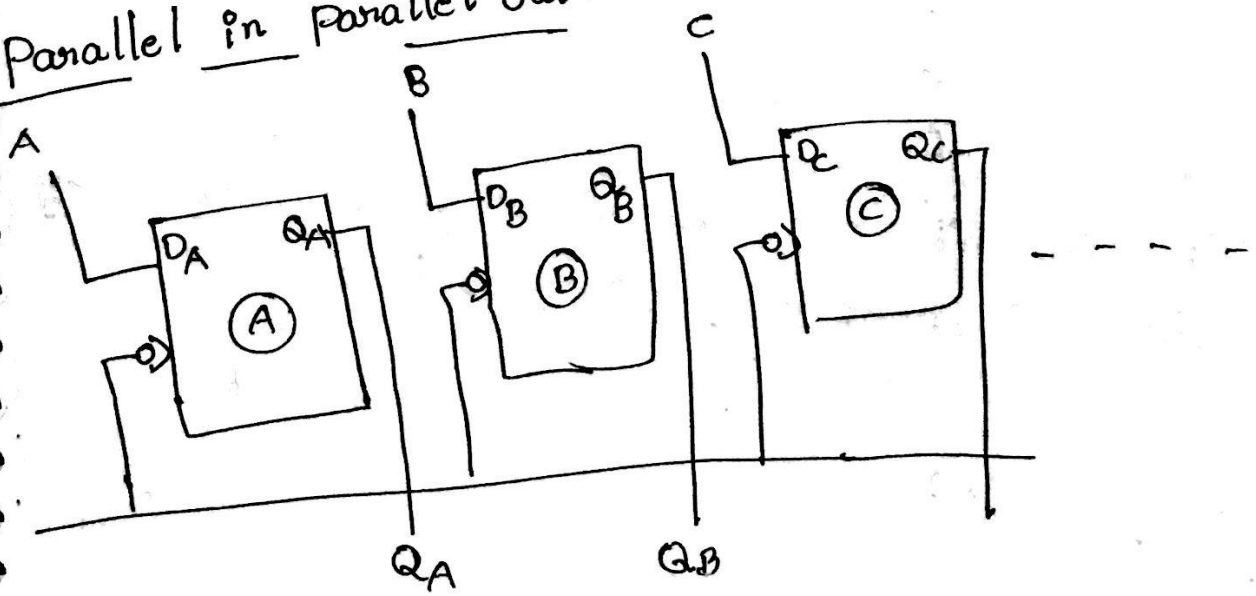
3) Parallel in Serial out :-



→ when the shift / $\overline{Load}$ signal is active low then the gates G1, G2, G3 are in enable mode and remaining AND gates G4, G5, G6 are in disabled mode & vice versa

→ The function of the OR gate used in these registers is parallel entry data operation (or) shift operation. This can be done by the inputs obtained from above AND gates.

Parallel in parallel out :-



Applications of shift registers:-

1. Serial to Parallel conversion (Spatial to temporal code conversion)
2. parallel to serial conversion (temporal to spatial code conversion)
3. Sequence generator.
4. Multiplication and division.
5. Ring counter and Twisted ring counter.
6. Digital delay line (SISO)

***:-

1) Left shift operation is nothing but multiplied by 2.

L ← R

Eg: Q3 Q2 Q1 Q0

0 1 0 1 = 5

1 0 1 0 = 10

→ shift left by n-positions is equivalent to multiplication by 2^n.

no of positions

$2 * 2^2 = 2 * 2^2 = 8$

$N \times 2^n$

n = given number

Eg:- $2^{(n)} = 0010$

0 0 1 0

0 1 0 0 1st

1 0 0 0 2nd = 8

2) a) If LSB bit = 0, then right shift operation by one position is same as Division by 2

L → R.

Eg: Q3 Q2 Q1 Q0(LSB)

1 0 1 0 = 10

0 1 0 1 = 5

(b) If LSB = 1, then right shift operation gives integer division by 2

Eg: Q_3 Q_2 Q_1 Q_0

0 1 0 1 = 5

0 0 1 0 = 2 (instead of 2.5)

→ Delay line:

n = 4

a) SISO = $(2n-1)T_c$ $7T_c$

b) SIPO = $n.T_c$ = $4T_c$

c) PISO = $(n-1)T_c$ = $3T_c$.

universal shift register:-

→ The most general shift register has the following capabilities:

i) A clear control to clear the register to 0.

ii) A clock input to synchronize the operations

iii) A shift right control to enable the shift right operation & serial i/p & o/p lines associated with SR

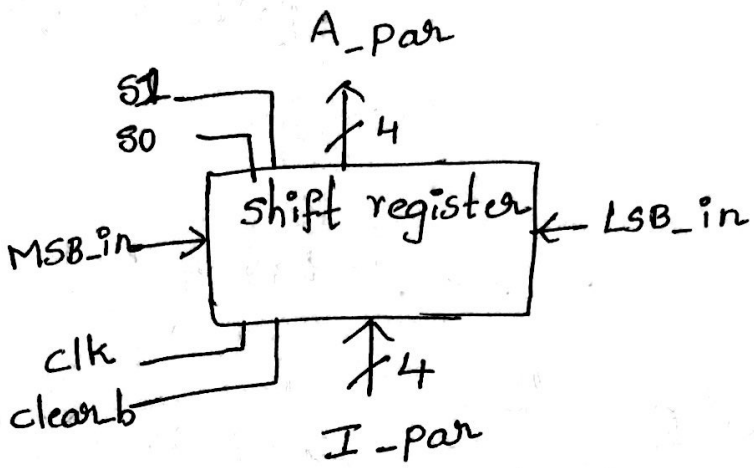
iv) A SL control to enable shift left operation & serial i/p & o/p lines associated with SL.

v) A Parallel Load control to enable a parallel transfer and n input lines associated with the parallel transfer

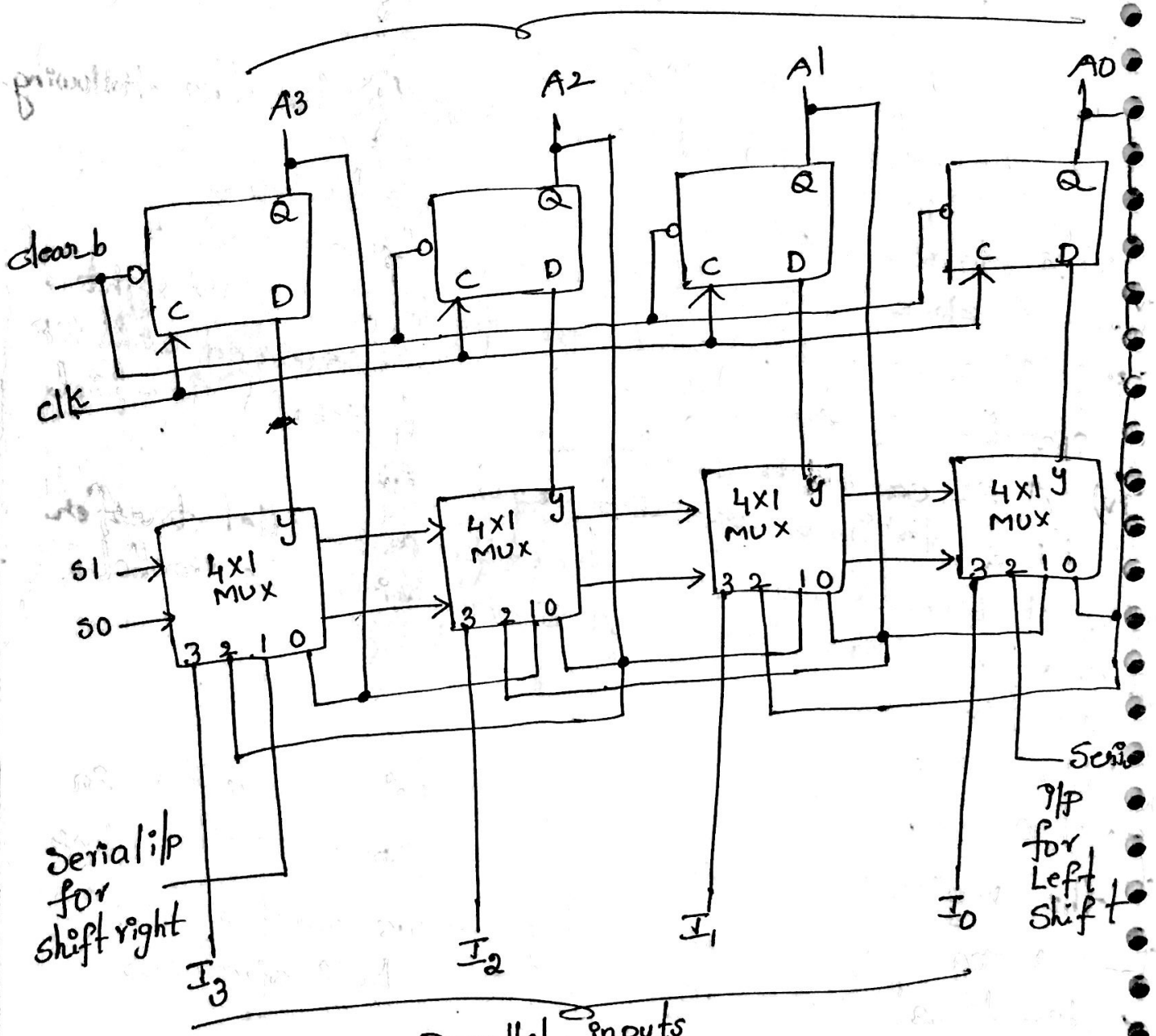
vi) n parallel o/p lines.

vii) A control state that leaves the information in the register unchanged in response to the clock.

→ A reg capable of shifting in one direction → uni directional shift register & in both directions = bidirectional shift register.



Block diagram



Ckt diagram.

→ The ckt consists of 4 DFF's & 4 Mux's. The 4 Mux's have 2 common selection lines S1 S0. I/P 0 in each mux is selected when S1S0 = 00, I/P 1 " " " " " S1S0 = 01 & so on.

→ The selection inputs control the mode of operation.

Mode control		Reg operation
S1	S0	
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel Load.

→ when S1S0 = 00, the present value of the reg is applied to the D i/p's of the FF's. This condition forms a path from the output of each FF into the i/p of the same FF. (o/p recirculates to the i/p). → The next clock edge transfers into each FF the binary value it held previously ie, No change.

→ S1S0 = 01, terminal 1 of Mux i/p's has a path to the D i/p's of the FF's. This causes a shift right operation with the serial input transferred into flip flop A_3.

Eg :-

→ [Txter] —channel→ [Rxer]

Txter: PISO

channel: Serial, SISO, Low cost (∵ n Txsion lines for far distance cost↑)

Rxer: SIPO.

Counters :-

→ The counter is driven by a clock signal and can be used to count the number of clock cycles. [frequency divider ckt].

Ring counter:- shift register can be used as ring counter when Q_0 o/p terminal is connected to serial i/p terminal.

→ An n-bit ring counter can have "n" different output states. It can count n-clock pulses.

Twisted ring counter:- It is also called Johnson ring counter. It is formed when $\overline{Q_0}$ o/p terminal is connected to the serial input terminal of the shift register. It can have max of 2n different o/p states.

→ The largest binary number that can be represent by an n-bit counter has a decimal equivalent of $2^n - 1$ Eg :- n=3, $2^3 - 1 = 7$

→ A counter having n FF's can have 2^n o/p states

→ It can count either in the up mode / down mode

→ The Modulus of a counter is the total no. of states Eg :- Mod-8 (000 to 111)

→ The o/p signal frequency of mod-n counter is 1/nth of the i/p clk frequency. Hence that counter is also called ÷n counter

→ The no. of FF's required to construct Mod-10 (or) ÷ 10 counter = 4. (decade counter)

→ formula : $2^{n-1} < N \leq 2^n$ N = Mod Number, n = FF's

→ Synchronous < Series carry / Parallel carry.

→ Based on the nature of the clock pulse applied The counters are classified into:

1) Asynchronous / Ripple (carry) / Series counter:-

→ In asynchronous counter the output of first FF is given as input (clock) to the next FF as a clock pulse. This processes continues such counters are known as Asynchronous counters.

2) Synchronous counter:- / Parallel

In Synchronous counter clock pulse is applied simultaneously to all the Flipflops.

Differences between Asynch and synchronous:-

Asynchronous	Synchronous
1. In this type of counters FF's are connected in such a way that the output of first FF derives the clock for the next FF.	1. In this type there is no connection between the output of first FF and clock i/p of next FF.
2. All the FF's are not clocked simultaneously.	2. All the FF's are clocked simultaneously.
3. Logic ckt is very simple even for more no. of states (2^n).	3. Design involves complex logic ckt as the no. of states increases.
4. Main drawback of this counter is there low speed as the clock is propogated through no. of FF's before it reaches Last FF	4. They are performed with high speed even when the no. of FF's are increased.

1) Design 4-bit asynchronous counter using JKFF's?

Sol:- 4-bit = 2^4 states = 16 states

$= 2^n$ = 4 FF's are required.

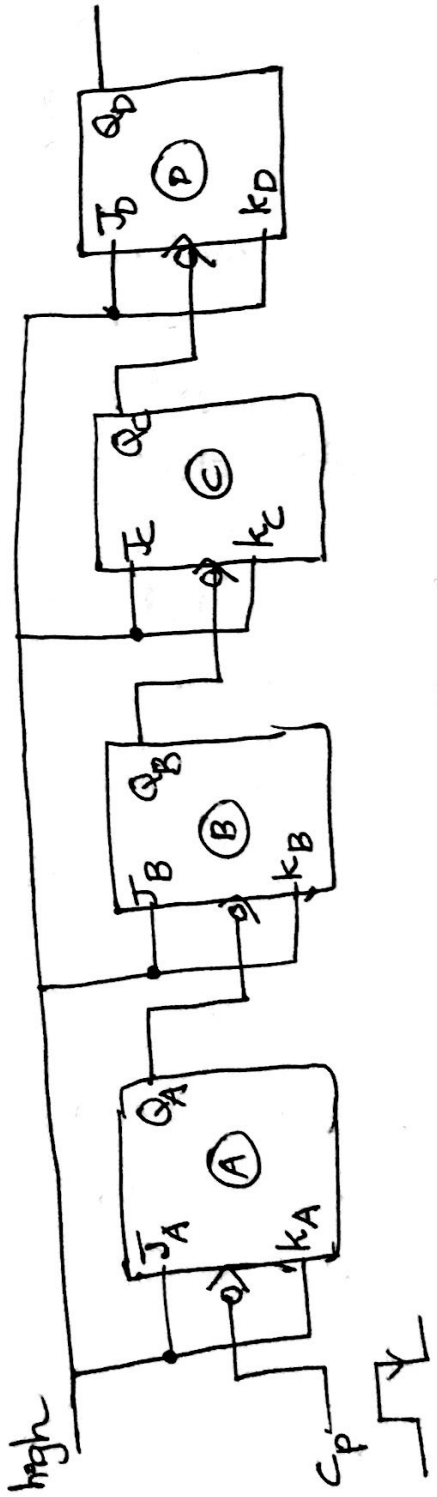
Excitation table for JK FF

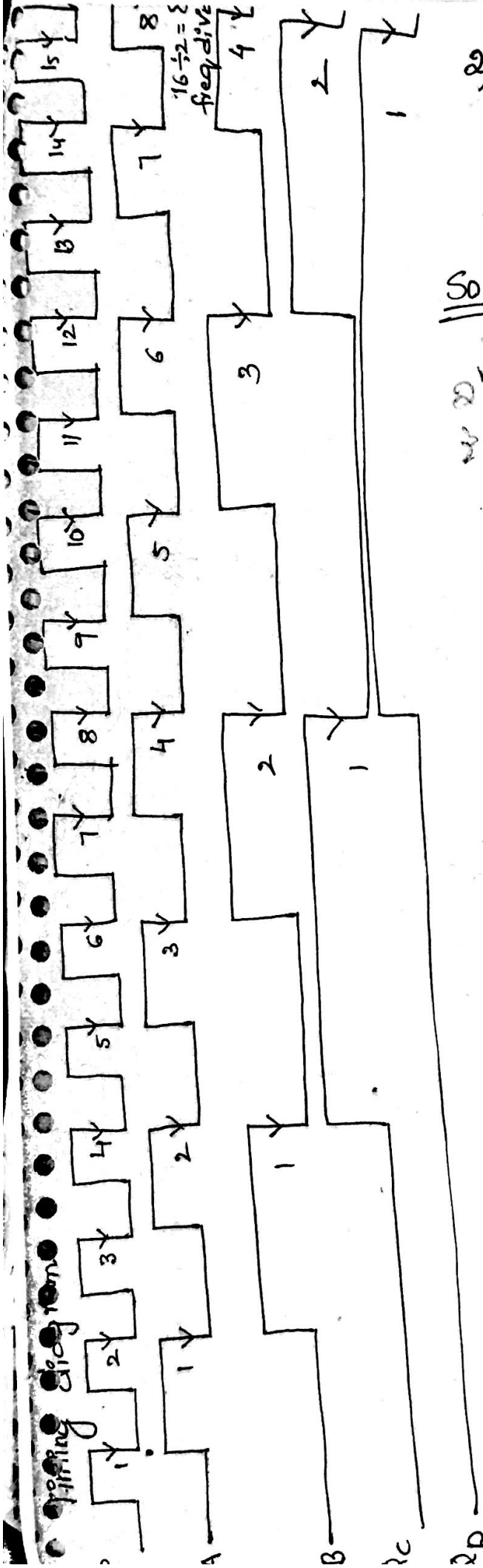
Q_n	Q_{n+1}	J	k
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

1) Here NS of 1st FF is given as Clk for 2nd FF so, there is no need of excitation table

State Table :-

CP	Q_D	Q_C	Q_B	Q_A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1





2). Design a MOD-5 synchronous counter using JK flip-flops and implement it. Also - construct a timing diagram.

Sol:- MOD - N

$2^n \geq N$

$2^n \geq N$ $2^2 \geq 5$ Not possible

$2^3 \geq 5$ possible ✓

So, we require 3 Flip Flops to design MOD-5 synchronous counter.

Excitation table for JK :-

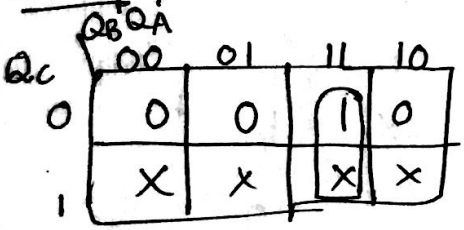
Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

PS	NS
0	1
1	2
2	3
3	4
4	0
5	X
6	X
7	X

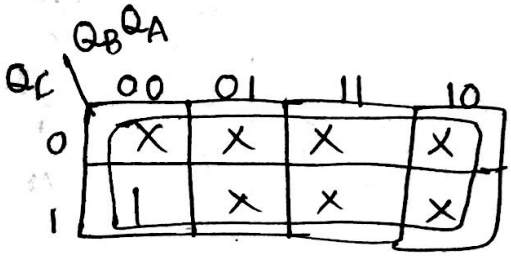
State Table:-

PS			NS			Flip-Flop Inputs					
Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	J_C	k_C	J_B	k_B	J_A	k_A
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	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	0	0	0	X	1	0	X	0	X
1	0	1	X	X	X	X	X	X	X	X	X
1	1	0	X	X	X	X	X	X	X	X	X
1	1	1	X	X	X	X	X	X	X	X	X
1st			2nd level			3rd level					

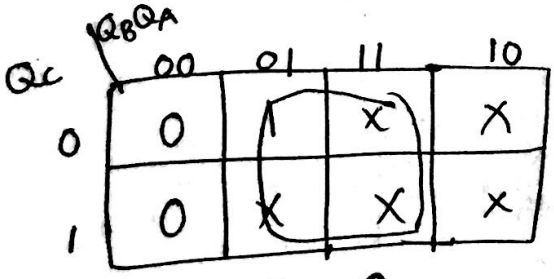
K-map :-



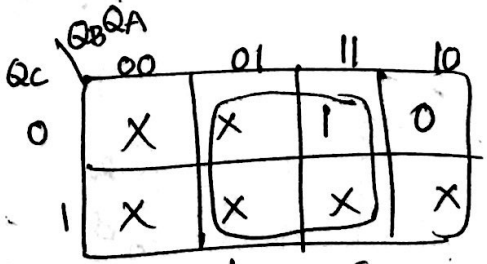
$J_C = Q_B Q_A$



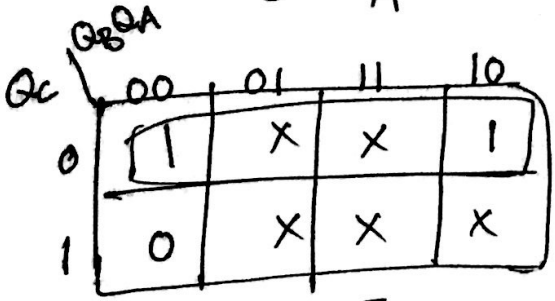
$k_C = 1$



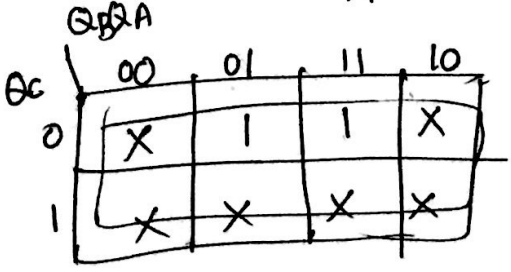
$J_B = Q_A$



$k_B = Q_A$

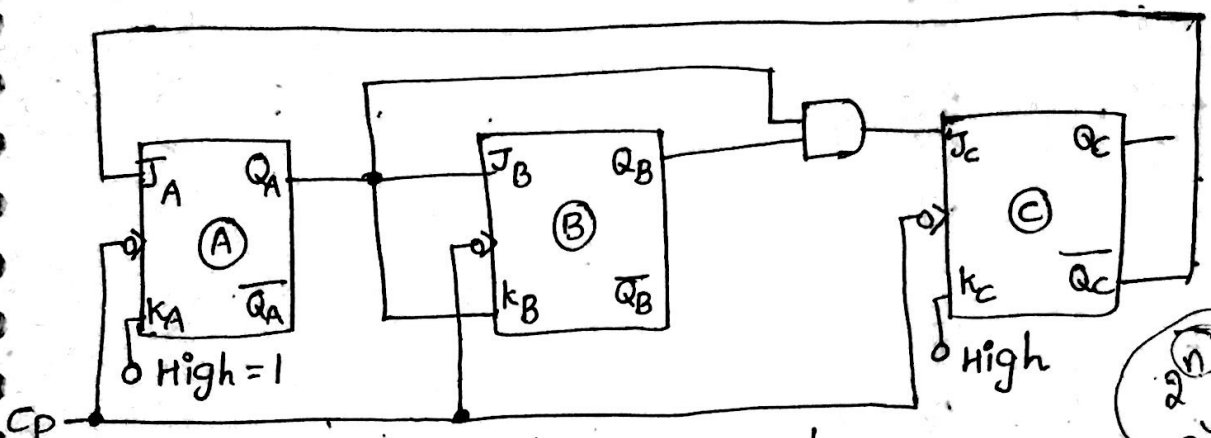


$J_A = \overline{Q_C}$



$k_A = 1$

Logic diagram:-



MOD-5 synchronous counter.

$2^n \geq N$
$2^4 \geq 12$
$2^4 \geq 10$

Synchronous decade / MOD-10 counter:-

$2^n \geq N$, $2^4 \geq 10$

FF's = 4.

State Table:-

PS				NS				FF inputs			
Q_D	Q_C	Q_B	Q_A	Q_{D+1}	Q_{C+1}	Q_{B+1}	Q_{A+1}	T_D	T_C	T_B	T_A
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

5 K-map:-

$T_D = Q_A Q_D + Q_A Q_B Q_C$

$Q_D Q_C$ \ $Q_B Q_A$	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	X	X	X	X
10	0	1	X	X

$T_C = Q_A Q_B$

$Q_D Q_C$ \ $Q_B Q_A$	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	X	X	X	X
10	0	0	X	X

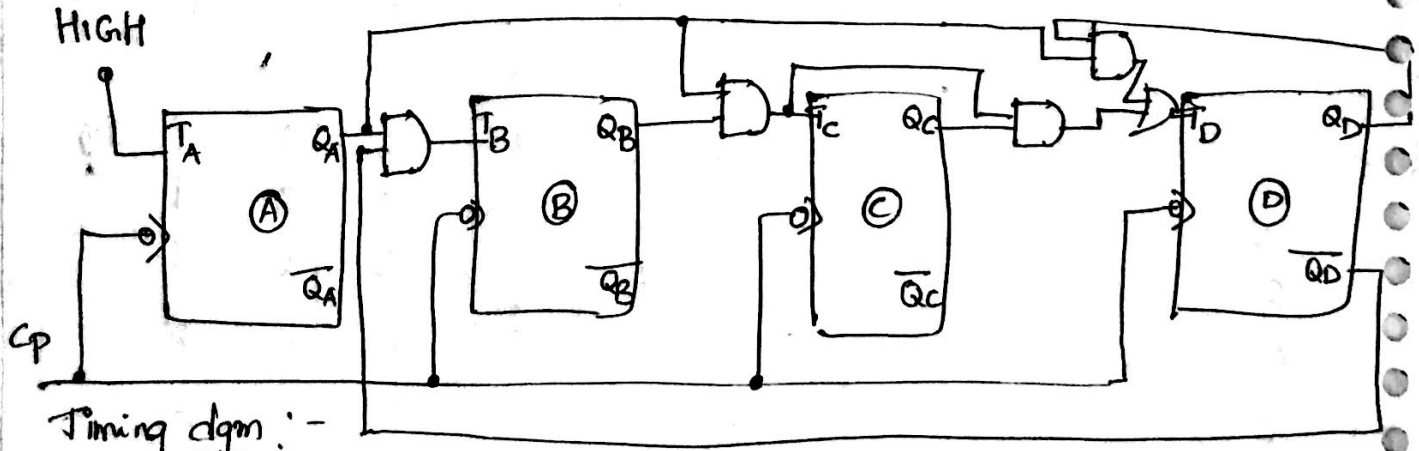
$Q_D Q_C$ \ $Q_B Q_A$	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	X	X	X	X
10	0	0	X	X

$T_B = Q_A \overline{Q_D}$

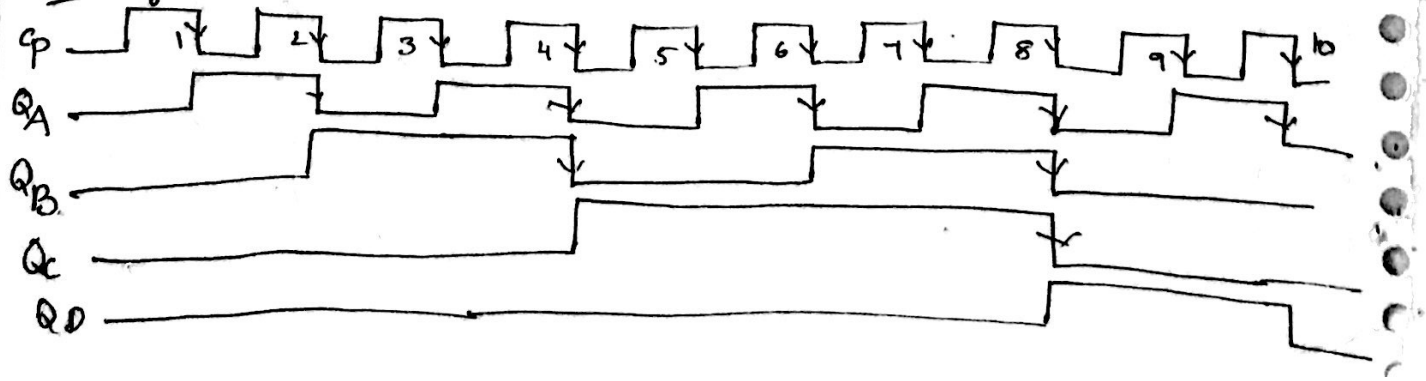
$Q_D Q_C$ \ $Q_B Q_A$	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	X	X	X	X
10	1	1	1	1

$T_A = 1$

Logic diagram:-



Timing dgm:-



4) Design of UP/DOWN Synchronous counters:-

→ An up/down counter is also called as bidirectional counter. Usually up/down operation of the counter is controlled by UP/$\overline{\text{DOWN}}$ signal.

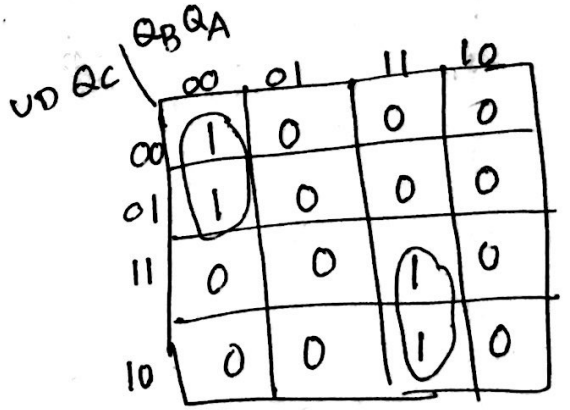
→ when this signal is high, counter goes through up sequence ie, 0, 1, 2, - - - n. when UP/$\overline{\text{DOWN}}$ = 0 counter follows reverse Sequence ie, n, n-1, n-2 - - - -1, 0.

→ For 3-bit counters these sequences are;
0, 1, 2, 3, 4, 5, 6, 7 for up operation & 7, 6, 5, 4, 3, 2, 1, 0 for DOWN operation.

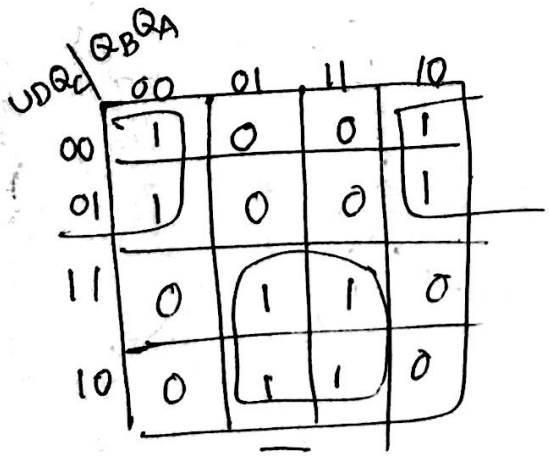
State Table :-

C_P	UP Q_C	Q_B	Q_A	DOWN
0	0	0	0	
1	0	0	1	
2	0	1	0	
3	0	1	1	
4	1	0	0	
5	1	0	1	
6	1	1	0	
7	1	1	1	

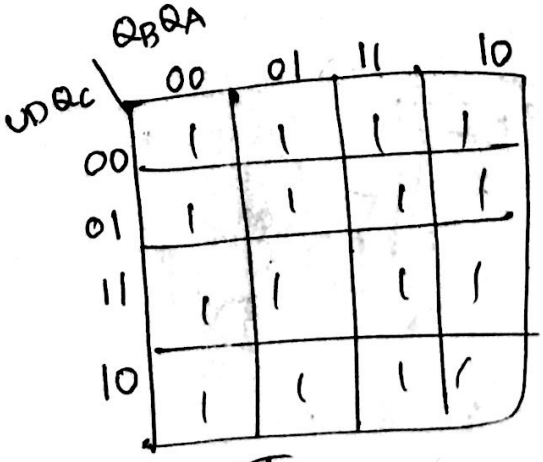
k-map :-



$T_C = \overline{U_D}\,\overline{Q_B}\,\overline{Q_A} + U_D Q_B Q_A$



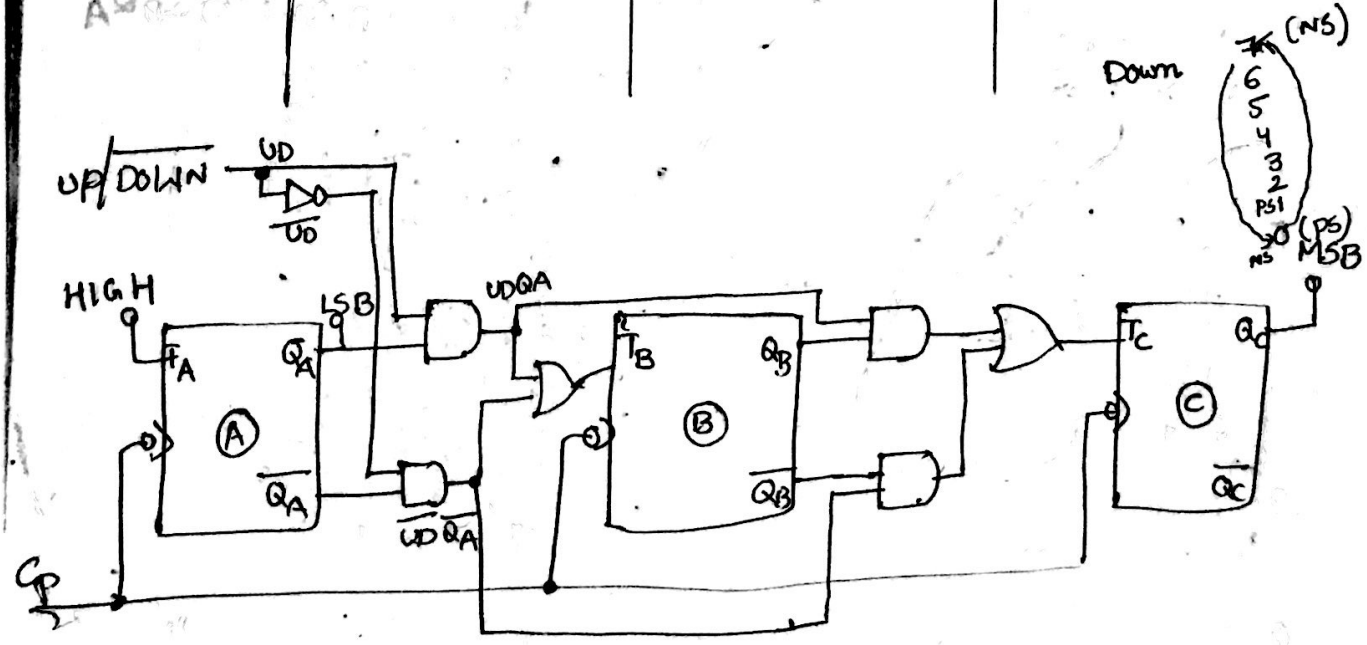
$T_B = \overline{U_D}\,\overline{Q_A} + U_D Q_A$



$T_A = 1$

TT :-

I/P	PS			NS			FF i/p's		
UP/DOWN (UD)	Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	T_C	T_B	T_A
0	0	0	0	1	1	1	1	1	1
0	0	0	1	0	0	0	0	0	1
0	0	1	0	0	0	1	0	1	1
0	0	1	1	0	1	0	0	0	1
0	1	0	0	0	1	1	1	1	1
0	1	0	1	0	0	0	0	0	1
0	1	1	0	1	0	1	0	1	1
0	1	1	1	1	1	0	0	0	1
1	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	0	0	1	1
1	0	1	0	0	1	1	0	0	1
1	0	1	1	1	0	0	1	1	1
1	1	0	0	1	0	1	0	0	1
1	1	0	1	1	1	0	0	1	1
1	1	1	0	1	1	1	0	0	1
1	1	1	1	0	0	0	1	1	1



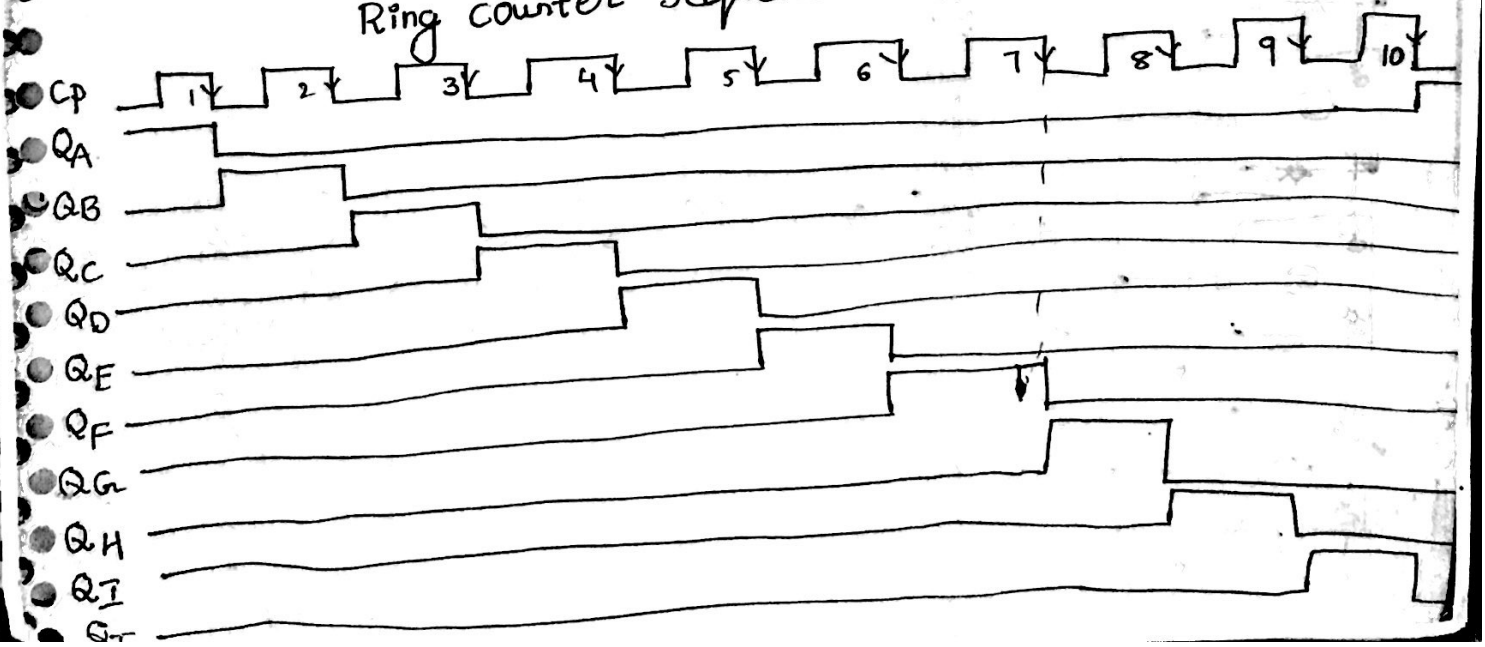
Ring counter:- → The Q o/p of each stage is connected to the D input of the next stage and the output of last stage is fed back to the input of first stage.

→ The $\overline{CLR}$ followed by $\overline{PRE}$ makes the o/p of first stage to '1'. & remaining outputs are zero. $Q_A = 1$, Q_B --- $Q_J = 0$.

→ The 1st clk pulse produces $Q_B = 1$ & remaining o/p's are zero. [10 bits] (0 --- 9)

Clk pulse	Q_A	Q_B	Q_C	Q_D	Q_E	Q_F	Q_G	Q_H	Q_I	Q_J
0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	1

Ring counter sequence (10-bits).



Shift/ Johnson/ Twisted Ring / Moebius Counter

→ In a Johnson counter, the Q o/p of each stage of flip-flop is connected to the D input of the next stage.

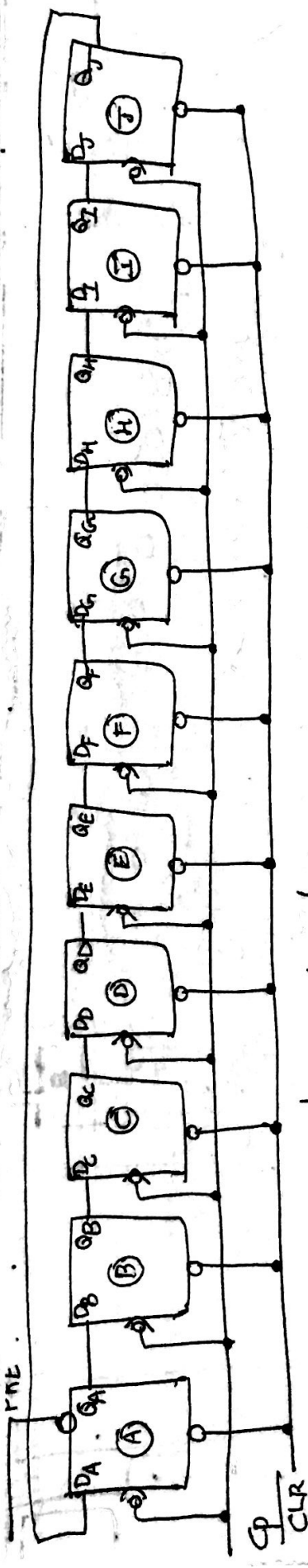
→ The single exception is that the complement output of the last flip-flop is connected back to the D-input of the first FF.

→ Initially the register (all FF's) is cleared. so all the outputs, $Q_A Q_B Q_C Q_D = 0000$. The o/p of last stage $Q_D = 0$. ∴, complement output of last stage, $\overline{Q_D}$ is one

→ This is connected back to the D input of first stage. so, $D_A = 1$.

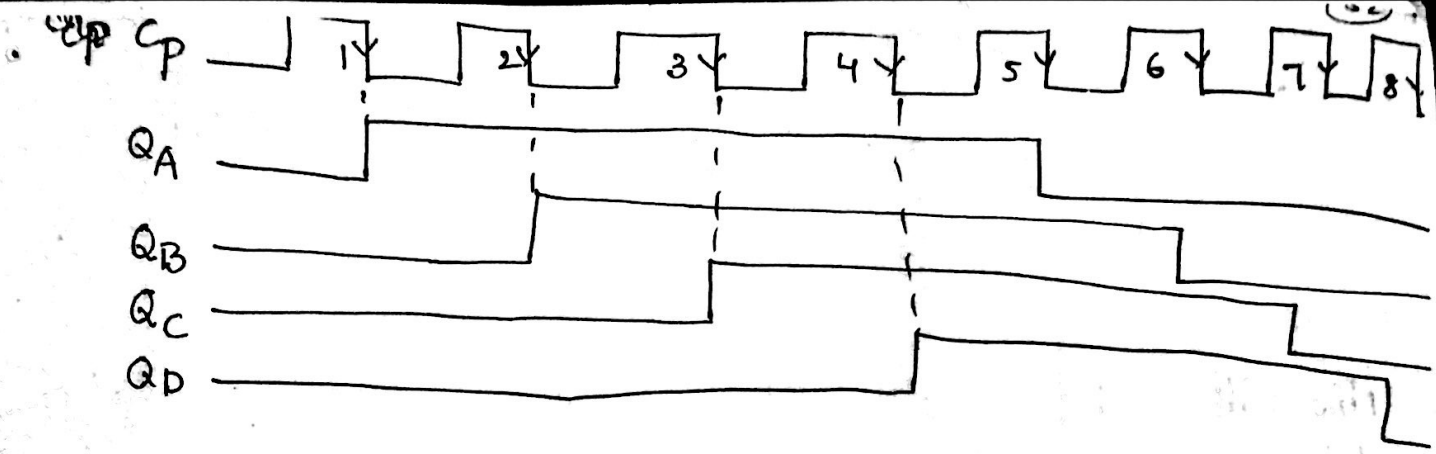
→ The 1st falling clock edge produces $Q_A = 1$ & $Q_B = 0$, $Q_C = 0$, $Q_D = 0$.

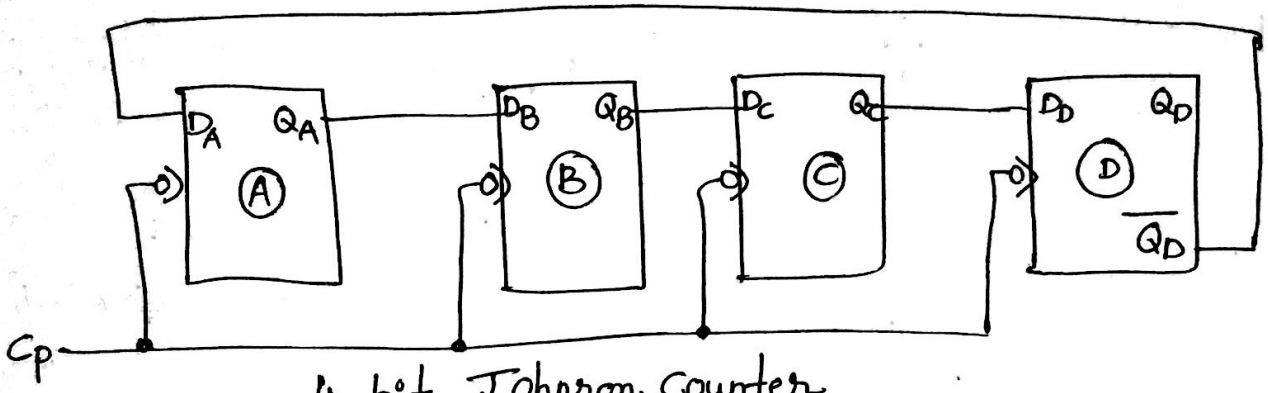
→ The next clock pulse produces $Q_A = 1$, $Q_B = 1$, $Q_C = 0$, $Q_D = 0$



C_P	Q_A	Q_B	Q_C	Q_D ($\overline{Q_D}$ = 1)
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1

4-bit Johnson sequence





4-bit Johnson Counter

Hint :- n stage Johnson counter will produce 2n

n = no. of stages (FF's)

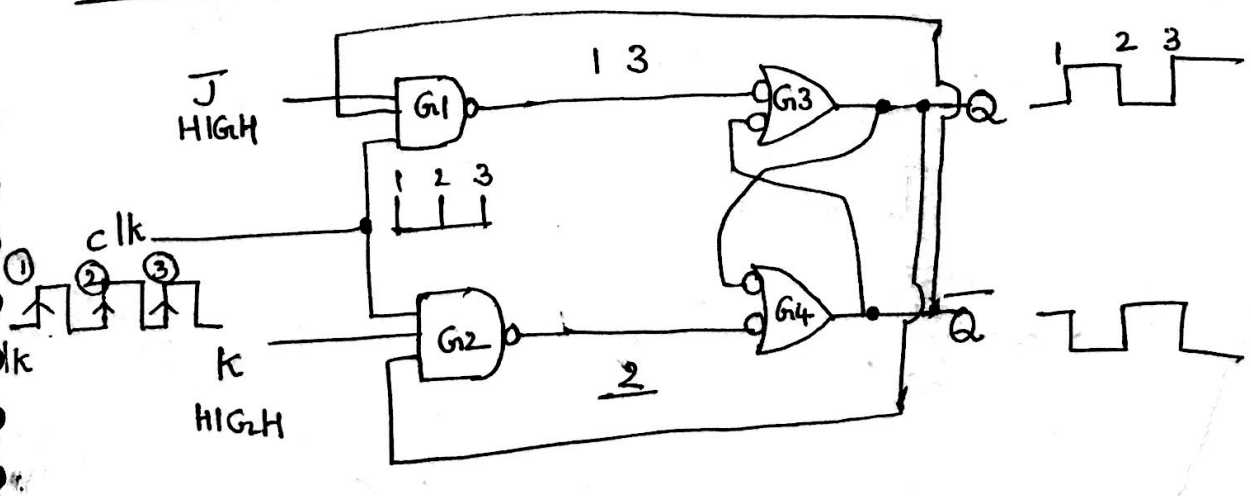
4 bit = 4 FF's = 2n = 2*4 = 8 stages 0 to 7

5-bit = 5 FF's = 2n = 2*5 = 10 stages 0 to 9

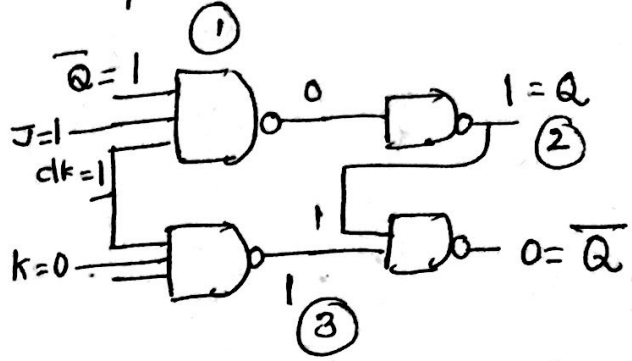
10-bit = 10 FF's = 2n = 2*10 = 20 stages 0 to 19.

JK F|F :-

Rayleigh Riccan



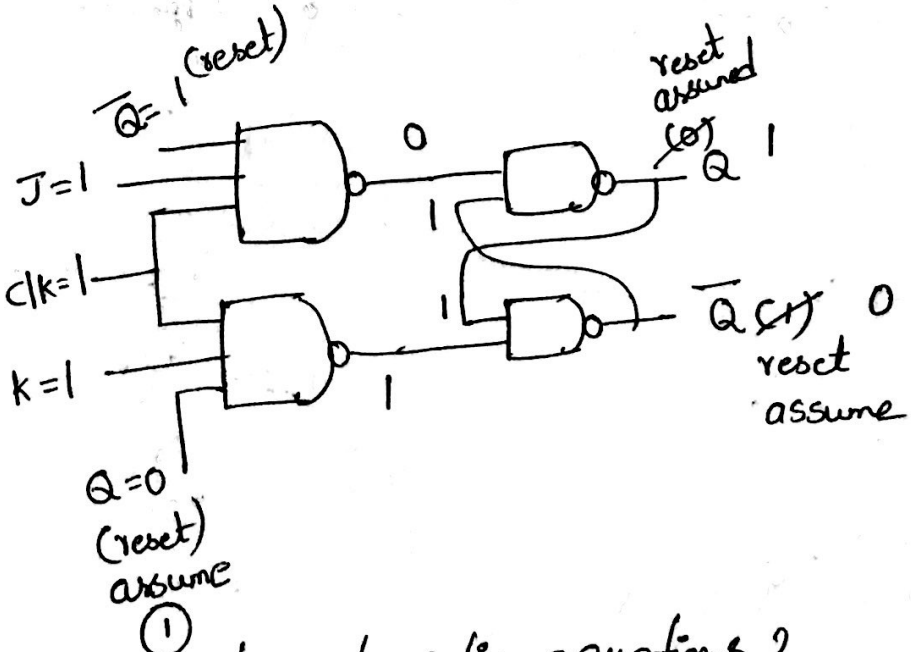
→ Let us assume that the F/F is RESET and $\overline{J}=1$, $k=0$. when a clk pulse occurs, a leading-edge spike indicated by ① is passed via gate G1 because $\overline{Q}=1$, $J=1$



This will cause the latch portion of the FF to change to the SET state.

→ The FF is now SET. $\overline{J}=0$, $k=1$, the next clock spike indicated by 2 will pass through gate G_2 because $Q=1$ & $k=1$. ⇒ RESET.

→ The difference in operation occurs when both the J and k inputs are HIGH. TO see this, assume that the F/F is RESET. [$Q=0$, $\overline{Q}=1$]. The high on the $\overline{Q}$ enables gate G1, so the clock spike indicated by ③ passes through to set the FF.



→ How to obtain characteristic equations?

Sol:- characteristic eqn's are obtained from characteristic

11 | 01