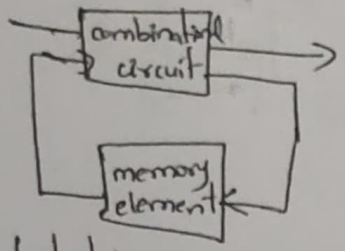
mood-book

UNIT-5

Sequential circuit:- In sequential circuit the output not only dependson Present inputs but also depends on Past inputs & outputs.



In sequential circuits flipflops & latches are used as memory element

latch:- → It is a one bit storage device → It has 2 states 1-state & 0-state. → It is also called as bistable multivibrator

→ A memory element without clock input is called as "latch".

Gated latch:- → The gated latches have enable input (or) gating signal

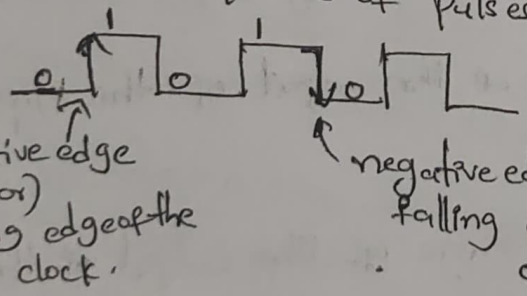
→ In gated latches the output will be change only when input changes & enable input is a logic 1 (high).

→ If the enable input is Low the output will not change even if the input changes. i.e.,

Flipflop:- → It is a one bit storage device. → It has 2 stable states which are called as 1-state & 0-state.

→ Flipflops have clock input and output changes only in response to the rising edge (or) falling edge of the clock.

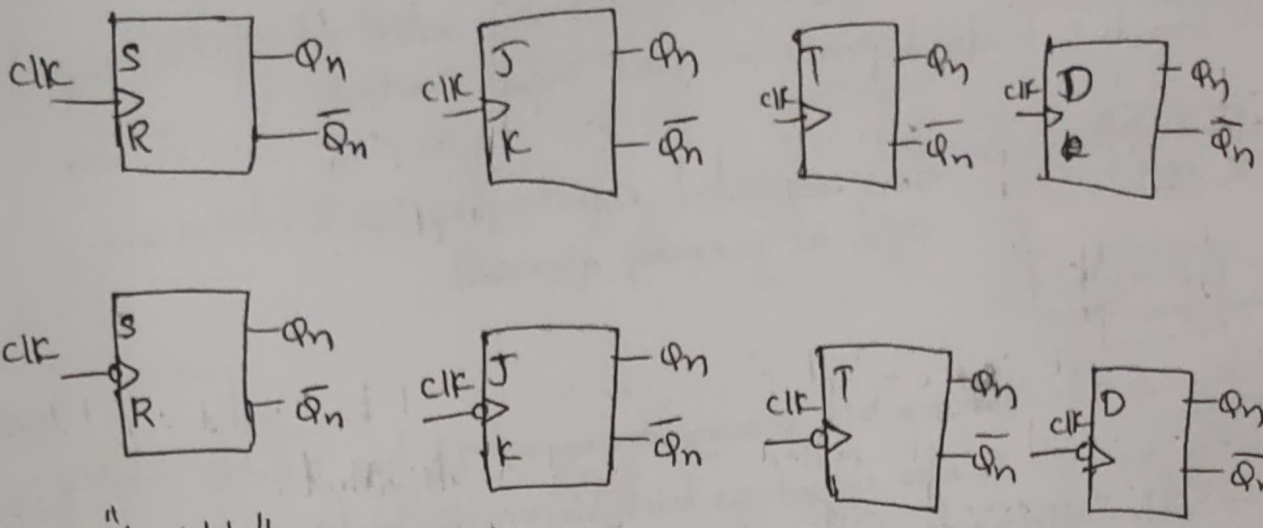
→ clock is a sequence of pulses.



Types of Flipflops:-

- SR flipflop → set-reset flipflop
- D flipflop → Data flipflop
- JK flipflop
- T flipflop → Toggle flipflop

Flipflops :-



→ A "bubble" on the clock input indicates that flip-flop is a "negative edge triggered" flipflop.

→ "Absence of bubble" indicates that flipflop is a "positive edge trigger" flipflop.

→ In positive edge trigger flipflop the output changes inresponse to a 0 to 1 transition (rising edge) in the clock input.

→ In negative edge trigger flipflop the output changes in response to a 1 to 0 transition (falling edge) in the clock input.

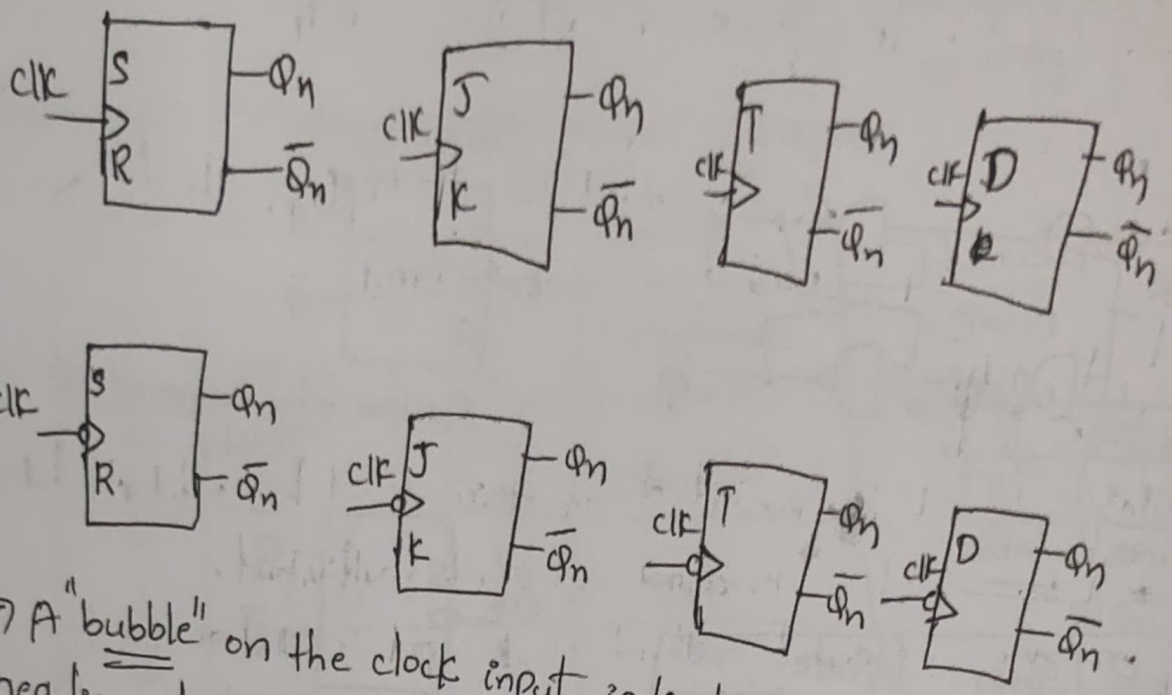
Present state (Q_n):- It is stage of the output of the flipflop at the time clock is applied

Next state (Q_{n+1}):- It is stage of the output of the flipflop after clock has been applied.

Characteristic table:- The characteristic table describe the operation of flipflop.

Excitation table:- This table give the information about input of the flipflops when output of the flipflop verified before & after clock pulse applied.

Flipflops:-



→ A "bubble" on the clock input indicates that flip-flop is a "negative edge triggered" flipflop.

→ Absence of bubble" indicates that flipflop is a "positive edge trigger" flipflop.

→ In positive edge trigger flipflop the output changes inresponse to a 0 to 1 transition (rising edge) in the clock input.

→ In negative edge trigger flipflop the output changes in response to a 1 to 0 transition (falling edge) in the clock input.

Present state (Q_n):- It is stage of the output of the flipflop at the time clock is applied

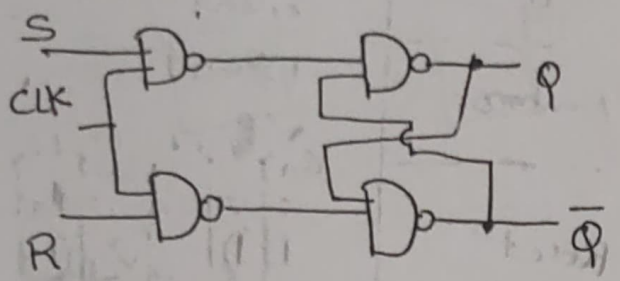
Next state (Q_{n+1}):- It is stage of the output of the flipflop after clock has been applied.

Characteristic table:- The characteristic table describe the operation of flipflop.

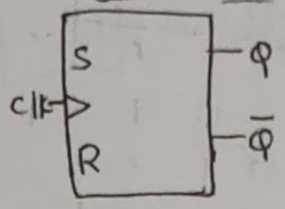
Excitation table:- This table give the information about the flipflops when output of the flipflop

(set-Reset)

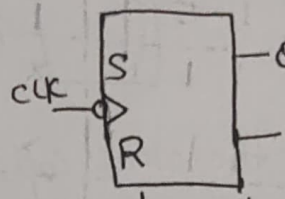
SR flipflop:-



logic symbol



Positive edge triggered SR flipflop



Negative edge triggered SR flipflop

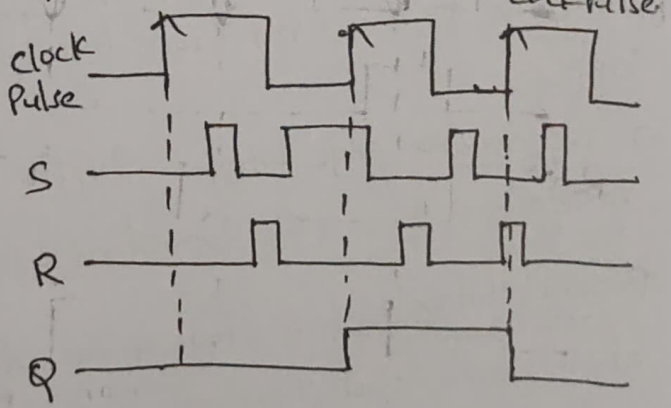
Characteristic table

S	R	Q_{n+1}	State
0	0	~~Not~~ Q_n	Nochange
0	1	0	Reset
1	0	1	set
1	1	Notallowed	Indeterminate

CLK	S	R	Q_n	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	X
↑	1	1	1	X
0	X	X	0	0
0	X	X	1	1

Truth table for Positive edge clocked SR flipflop

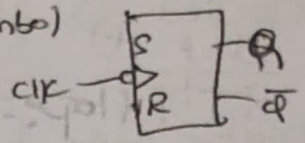
→ In this Positive edge trigged SR flipflop the output responds to the S & R inputs only at the Positive edges of the clock Pulse



Input and output waveforms for Positive edge trigsered clocked SR flipflop

Negative edge triggered SR flipflop :-

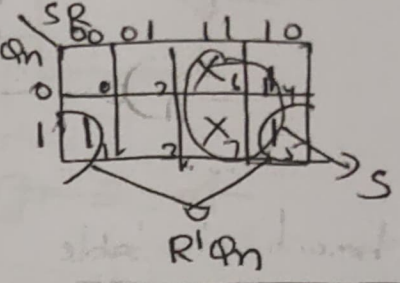
logic symbol



Truth table for negative edge clocked SR flipflop

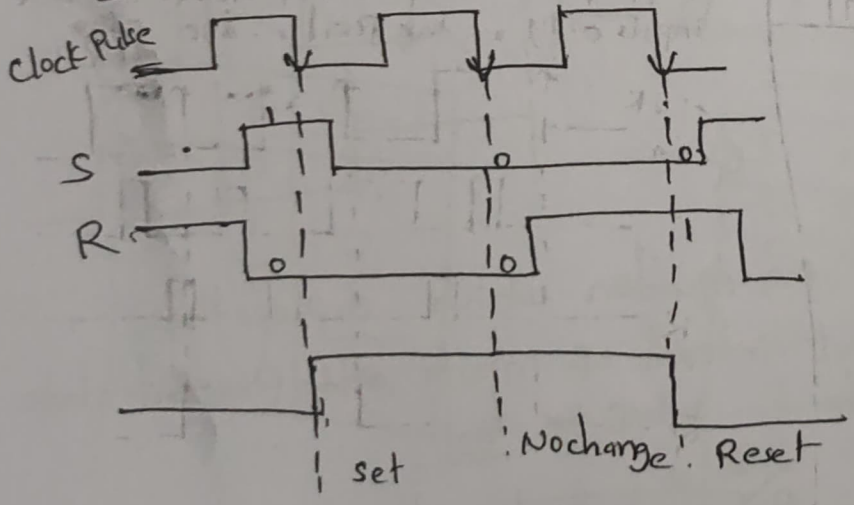
Clock Pulse	S	R	Q_n	Q_{n+1}	state
↓	0	0	0	0	Nochange
↓	0	0	1	1	
↓	0	1	0	0	Reset
↓	0	1	1	0	
↓	1	0	0	1	set
↓	1	0	1	1	
↓	1	1	0	X	Indeterminate
↓	1	1	1	X	
0	X	X	0	0	Nochange
0	X	X	1	1	

characteristic equation (Q_{n+1})



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

Input & output waveforms for negative edge triggered clocked SR flipflop.



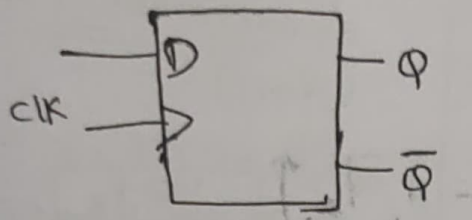
In the negative edge triggered flipflop, the negative edge detector circuit is used & the circuit output responds at the negative edges of the clock Pulse.

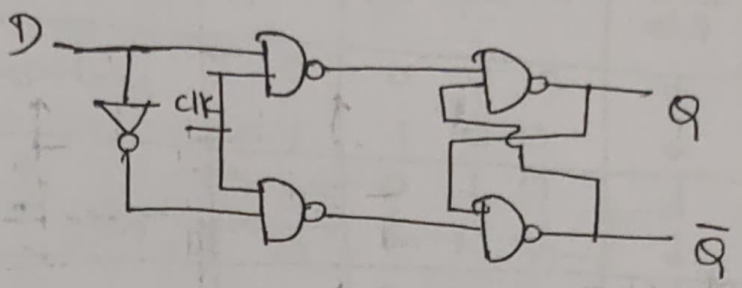
Excitation table :-

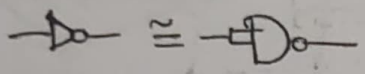
Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

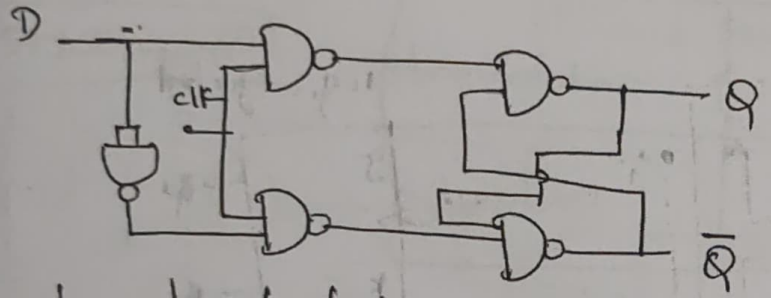
Clocked D flipflop:-

Positive edge triggered D flipflop.



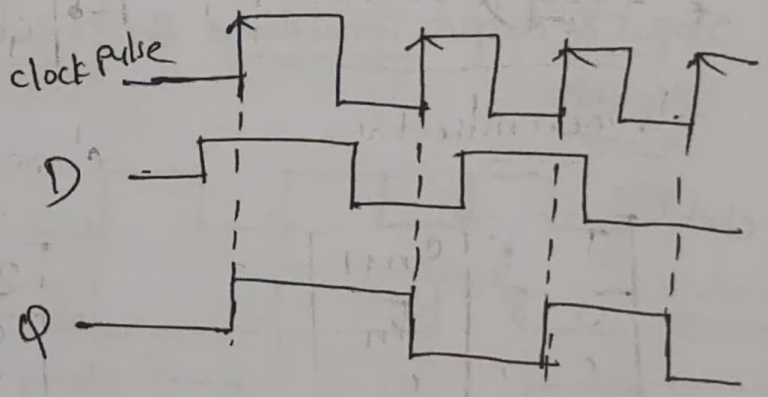






characteristic table

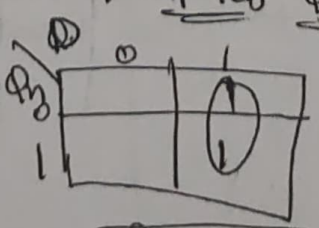
cp	D	Q_{n+1}
↑	0	0
↑	1	1
0	X	Q_n



characteristic equation:- An equation which expresses the next state of the output interms of inputs and present state is called characteristic equation.

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

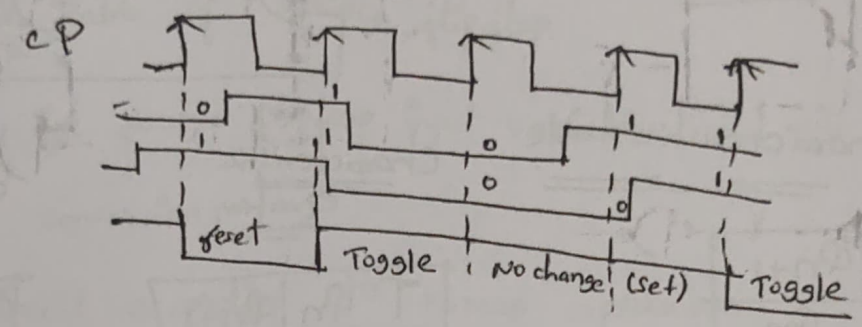
K-map for Q_{n+1}



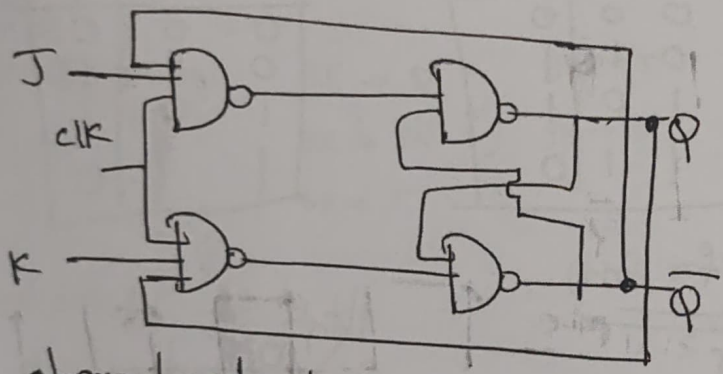
$Q_{n+1} = D$

Excitation table :-

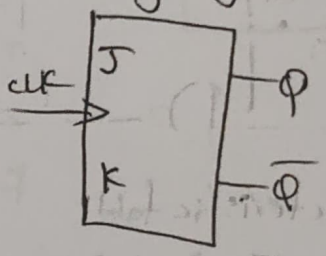
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1



JK flipflop :-



logic symbol



characteristic table

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

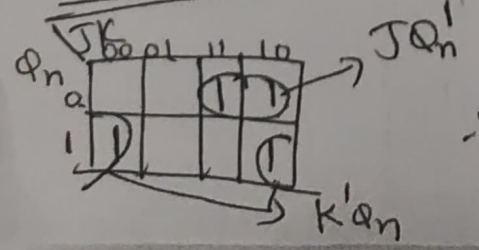
characteristic equation

J	K	Q_n	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	1
1	1	1	0

Excitation table

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

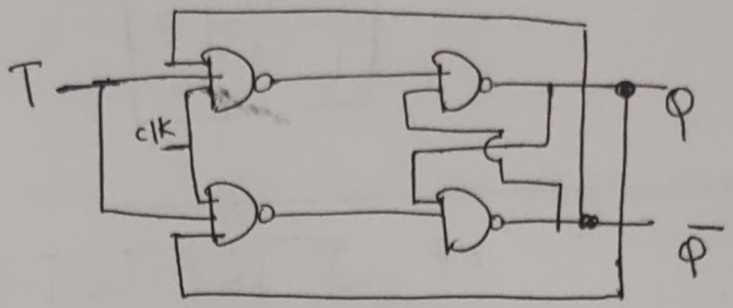
kmap for Q_{n+1}



$\therefore Q_{n+1} = JQ_n' + K'Q_n$

T flipflop :(Toggle flipflop):-

logic symbol



characteristic table

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

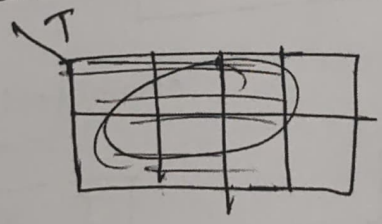
Characteristic equation

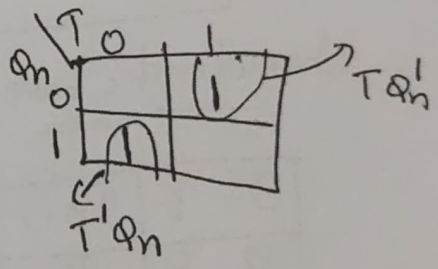
T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

excitation table

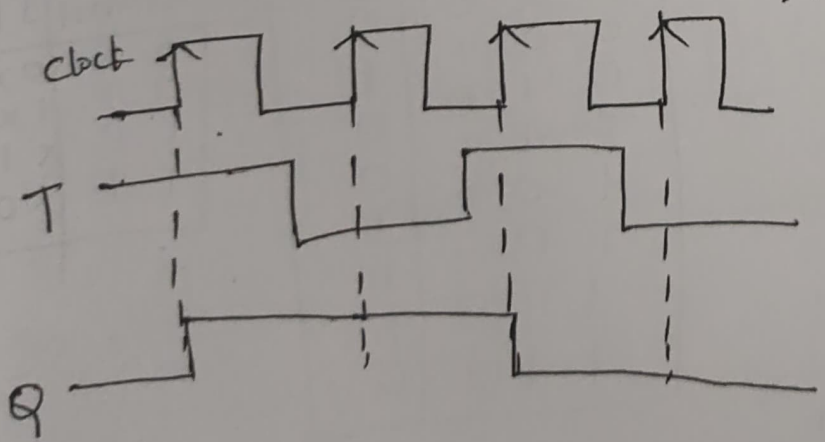
Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

K-map for Q_{n+1}





$\therefore Q_{n+1} = TQ_n' + T'Q_n$



Flip flop conversions

Steps :-

(i) Consider characteristic table for destination flip flop & extend it as excitation table of source flipflop.

(ii) Draw K-map for source flipflop input variables, it will provide expression for conversion

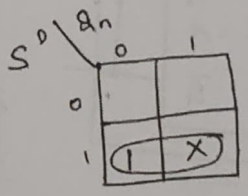
(iii) Draw the circuit according to K-map expressions.

Conversions : SR → D, JK → T, T → D, SR → JK, JK → D, D → T, SR → T

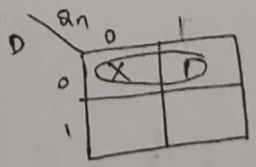
* Convert SR flf → D flf

(i)

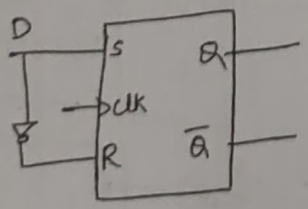
D	Q_n	Q_{n+1}	S	R
0	0	0	0	X
0	1	0	0	1
1	0	1	1	0
1	1	1	X	0



S = D



R = $\bar{D}$



* Convert SR to JK

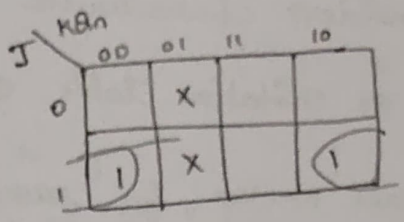
Excitation table of SR

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

JK clk table

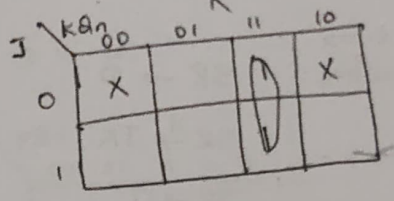
J	K	Q_n	Q_{n+1}	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

S

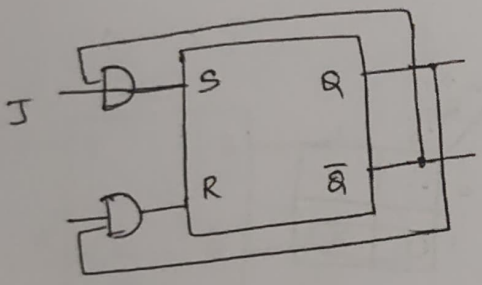


$S = J\bar{Q}_n$

R



$R = KQ_n$

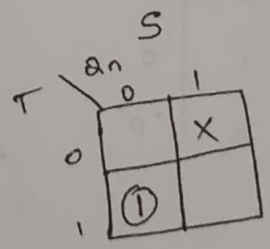


* Convert SR to T

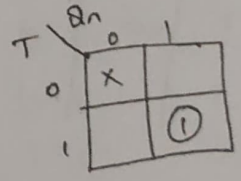
T clk table

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

S

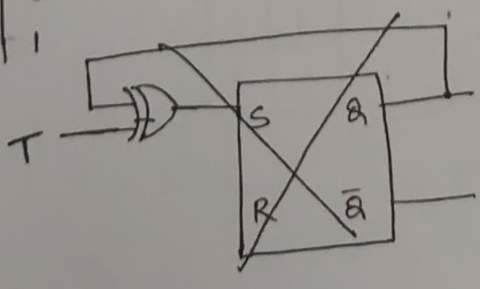


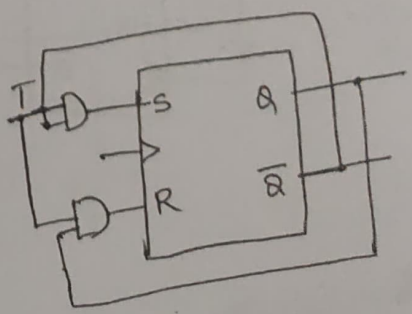
R



$S = T\bar{Q}_n$

$R = TQ_n$



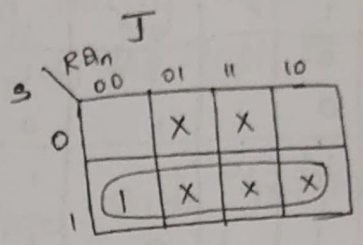


* Conval JK to SR

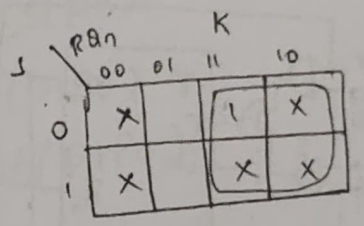
chg table of SR

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

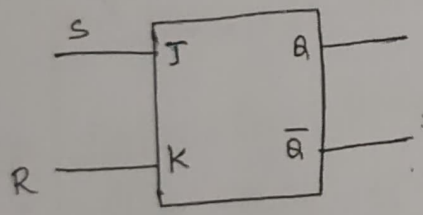
J	K
Q_n Q_{n+1}	J K
0 0	0 X
0 1	1 X
1 0	X 1
1 1	X 0



J = S



K = R

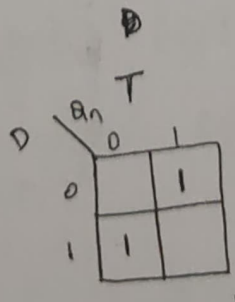


* Convert T to D

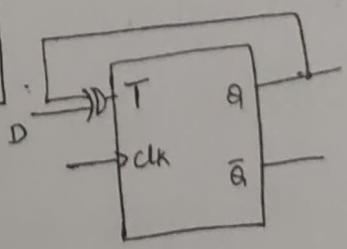
D chg table

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

0	0	0
0	1	1
1	0	1
1	1	0



$T = D \oplus Q_n$

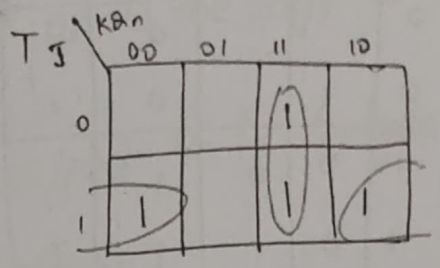


* Convert T to JK flip flop

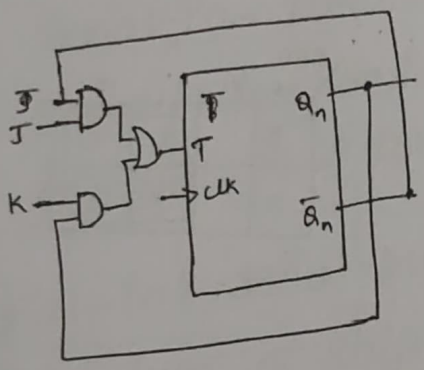
JK

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

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



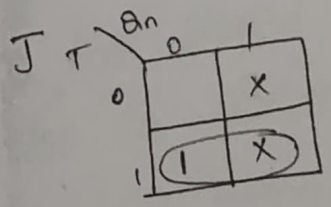
$T = J\bar{Q}_n + KQ_n$

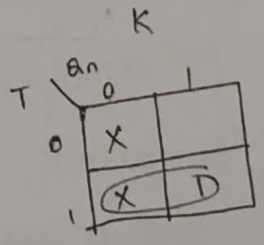


* Convert JK to T

T

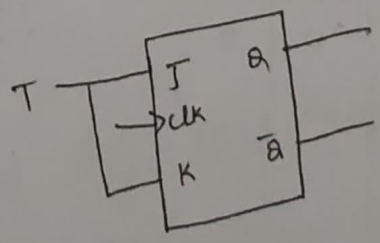
T	Q_n	Q_{n+1}	J	K
0	0	0	0	X
0	1	1	X	0
1	0	1	1	X
1	1	0	X	1





$J = T\bar{Q}_n$ $J = T$

$K = T$



Shift Register:-

A register that is used to store binary information is known as a memory register. A register capable of shifting binary informa-tion either to the right or to the left is called a shift register.

Shift registers are classified in to 4-types

1. Serial-in - Serial out (SISO)
2. Serial-in - Parallel out (SIPO)
3. Parallel-in - Serial out (PISO)
4. Parallel-in - Parallel out (PIPO)

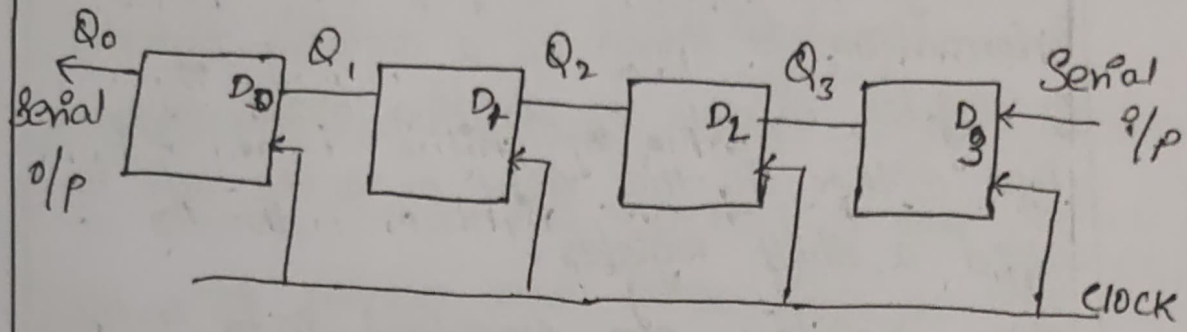
* Serial-in - Serial-out Shift register:-

This type of shift register accepts data Serially, i.e one bit at a time on a single input line. It produces the stored information on its single output also in serial form.

Data can be shifted left using shift-left register or shifted right using shift-right register.

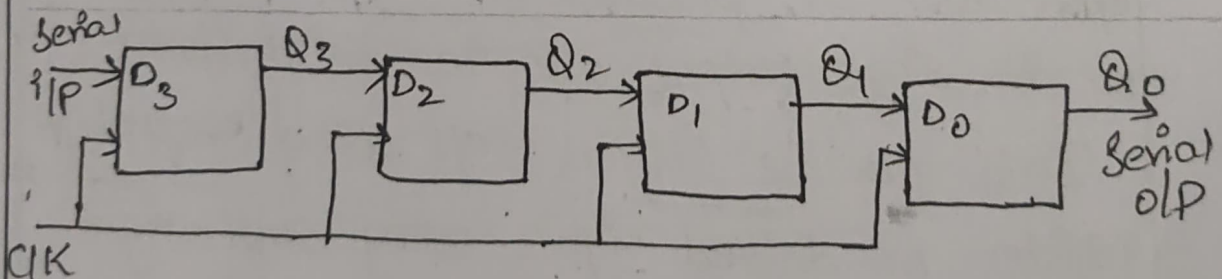
Shift left register:- A Shift left register can be built using D-flip-flops as

Shown in the fig.



CLK	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	1	1
4	ϕ	1	1	1

* **Shift Right Register:** A shift right register can be built using D-flip-flops as shown.



CLK	Q_3	Q_2	Q_1	Q_0
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

Serial-in-parallel-out shift Register :-

A 4-bit serial-in-parallel-out shift register is as shown below. It consists of one serial input, and outputs are taken from all the flip-flops parallely.

* In this register data is shifted in serially but shifted out in parallel.
* In order to shift the data out in parallel, it is necessary to have all the data available at the output at the same time.
* Once the data is stored, each bit appears on its respective output line simultaneously.

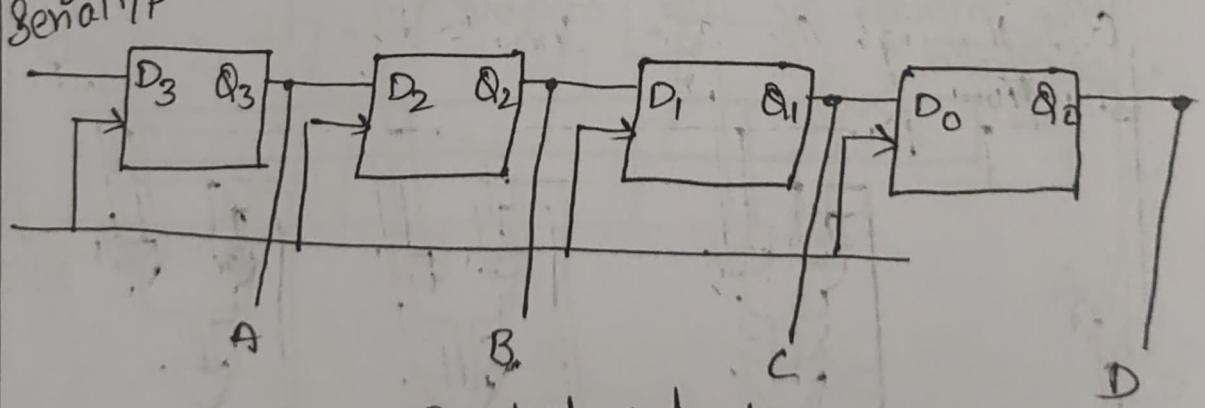


fig :- 4-bit Serial-in parallel-out shift register.

* Parallel-in-Serial-Out shift register:-

A four bit PISO register has 4 inputs A, B, C, D. this A, B, C, D be the four parallel data i/p lines and, Shift/$\overline{Load}$ is a control pin that allows the four bits of data on to register on parallel. or shift data in serial.

→ when Shift/$\overline{Load}$ is low, And gates G_1 through G_3 are enabled, allowing the data at parallel i/ps ie B, C, & D to the 'D' i/p of respective flip-flop.

→ when Shift/$\overline{Load}$ is high, And gates G_1 through G_3 are disabled & the remaining And gates G_4 through G_6 are enabled:

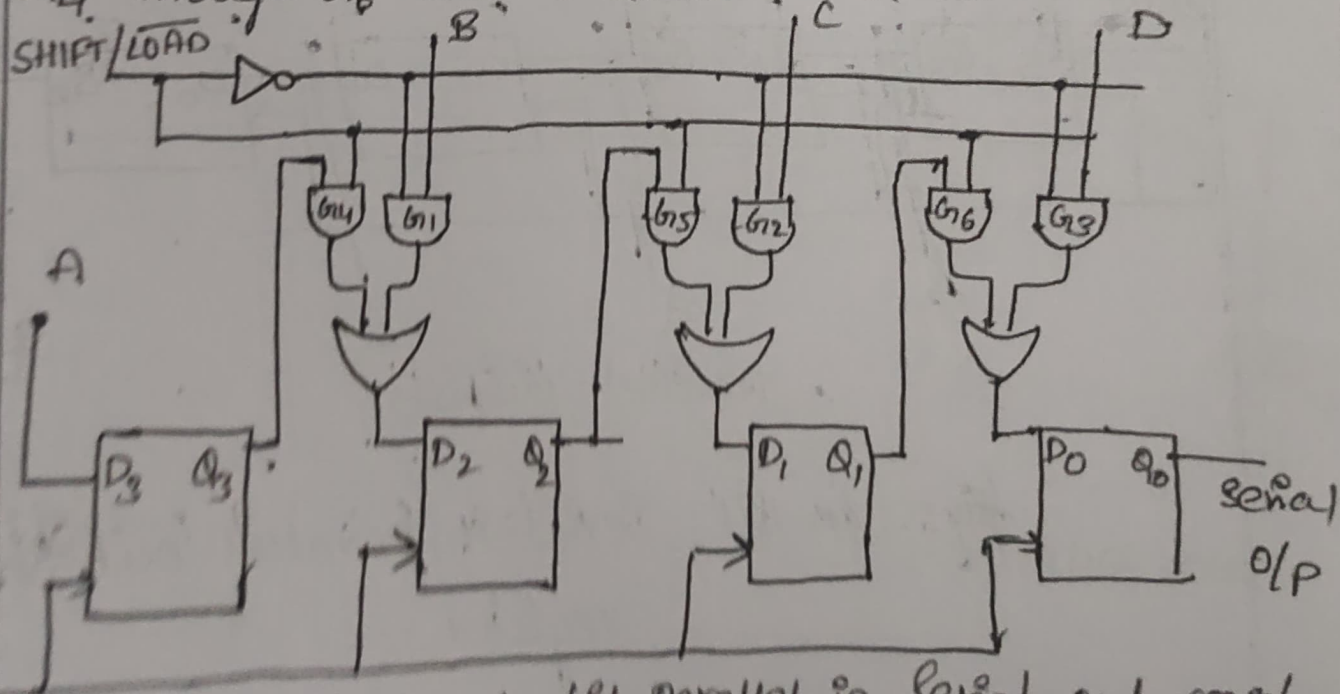


Fig:- 4-bit Parallel-in-Serial-out register.

Parallel-in-parallel-out Register:-

A simple 4-bit parallel-in-parallel-out shift register using D-flip-flop is shown here. the parallel inputs to be entered should be applied at A, B, C, D inputs which are directly connected to delay (D) inputs. Now by applying a clock pulse, these inputs are entered in to the register and are immediately available at the outputs. Q_1 Q_2 Q_3 & Q_0. Parallel o/ps

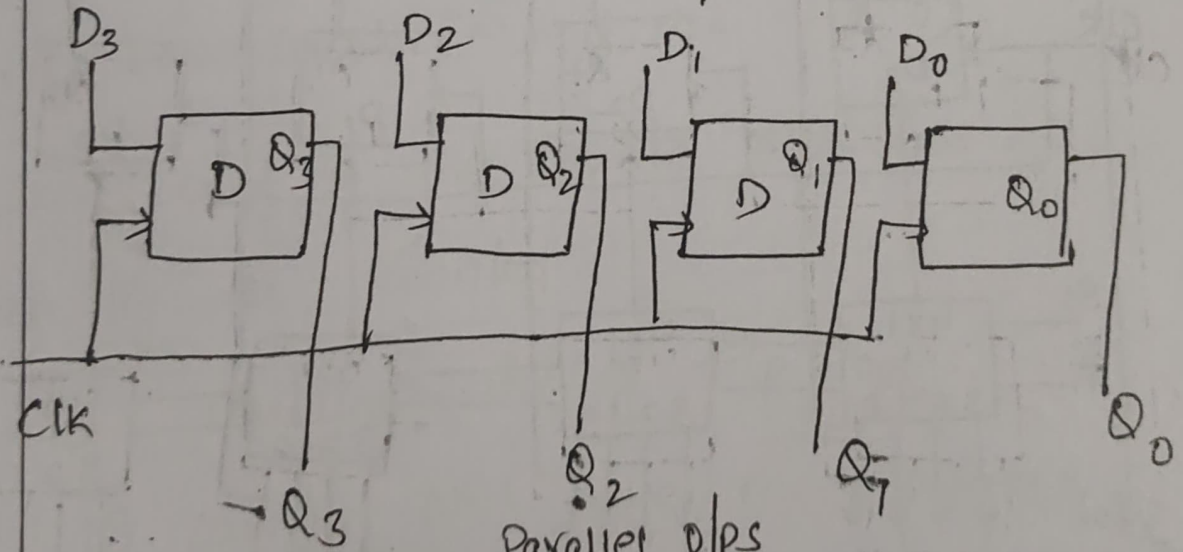


fig:- 4-bit parallel-in-parallel-out shift register.

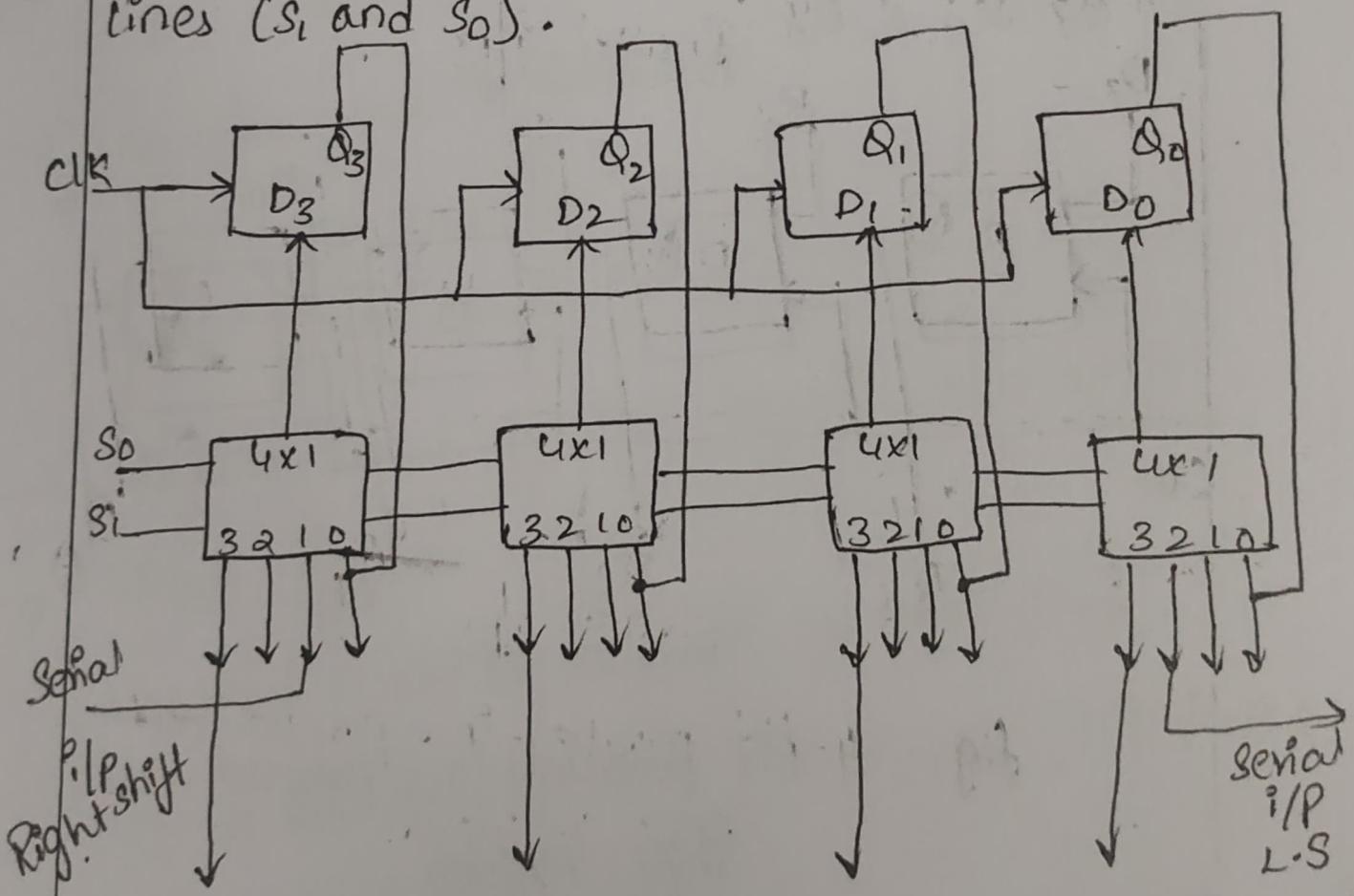
UNIVERSAL SHIFT REGISTER:-

A universal shift register is a register which has both the right shift & left shift with parallel load capabilities.

→ universal shift registers are used as memory elements in computers.

→ A n-bit universal shift register consists of n-flip flops and n 4x1 multiplexers.

→ All the multiplexers share the same select lines (S_1 and S_0).



Counters:

→ Counter is a digital device used to count number of pulses & it can also be used as frequency divider

→ Classified into two types:- according to clock cycle.

1) Asynchronous counter
2) Synchronous "

→ counter can count in two ways i.e

1. Up counter (0, 1, 2 - - N)
2. Down counter (N, N-1, N-2 - - 1, 0)

→ Present count of the counter represents state of the counter.

→ counter contains set of flip flops.

A, n-bit counter will have n- flip flops & 2^n states.

→ Each state frequency $= \frac{\text{total freq}}{2^n}$

Counters are of two types

→ Asynchronous Counters

→ Synchronous Counters.

Asynchronous Counters:

A binary ripple/ asynchronous Counter Consists of a series Connection of Complementing flipflops, with the output of each flipflop Connected to the clock input of the next higher order flip-flop. The flipflop holding the least significant bit receives the incoming clock pulses. A Complementing flipflops Can be obtained from a JK flipflop with the J and K inputs tied together, or from a T flipflop. A third alternative is to use a D flipflop with the Complementing output Connected to the D input. In this way, the D input is always Complement of the present state and the next clock pulse will Cause the flipflop to Complement.

2-bit Asynchronous Counter:

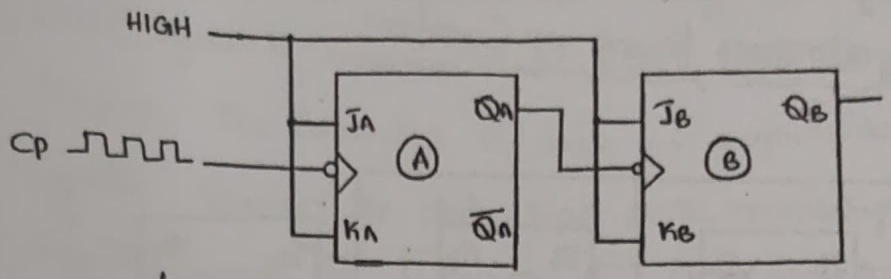
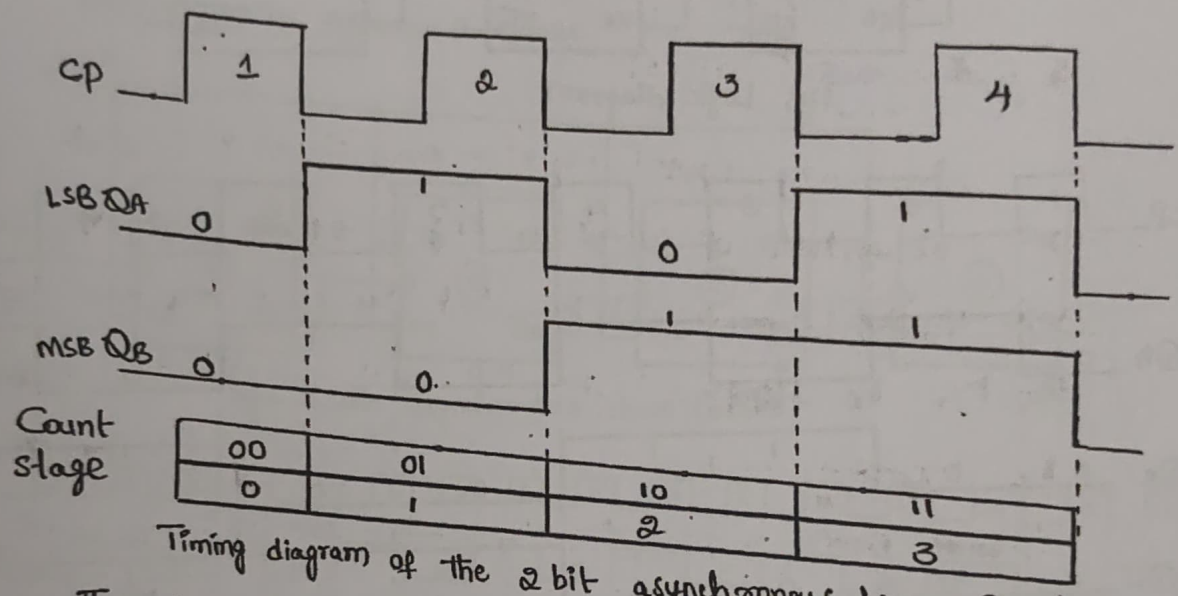


fig: A two-bit asynchronous binary Counter.



Timing diagram of the 2 bit asynchronous binary Counter.

The above 2 bit Counter is designed with JK flipflops. The clock signal is Connected to the clock input of only first stage flipflop. The clock input of the second stage flipflop is triggered by the QA output of first stage. Because of the inherant propagation delay time through a flipflop, a transition of the input clock pulse and a transition of the QA output of first stage Can never occur exactly at the same time. Therefore the two flipflops are never Simultaneously triggered. which results in asynchronous Counter operation.

In the timing diagram of two bit Counter the changes in the state of the flipflop outputs in response to the clock. J and K input

of JK flipflops are tied to logic HIGH. hence output will toggle for each negative edge of the clock input.

3-bit Asynchronous binary up Counter:

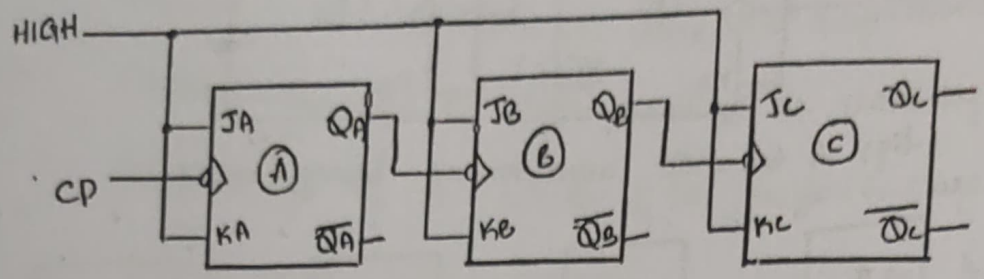
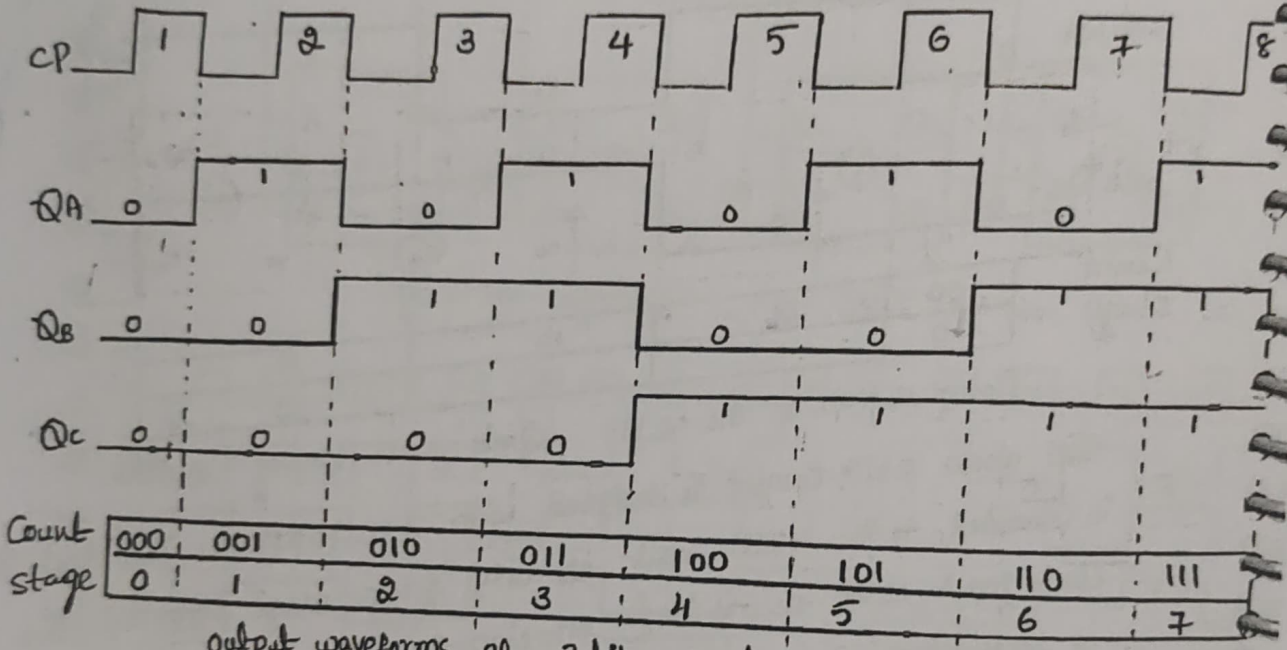


fig: Logic diagram.



output waveforms of 3 bit asynchronous Counter.

Asynchronous / Ripple Down Counter:

If the output of the Counter is incoemented by one for each clock transition, therefore, we call such Counters as up Counters. In Case of down Counters the Count will be down from a maximum Count to zero.

4 bit Asynchronous down Counter: The following figure is the 4 bit Asynchronous down Counter using J.K flip flops. Here the 1111 clock signal is connected to the clock input of only first flip flop. This connection is same as asynchronous ripple up Counter. However the clock input of the remaining flip flops is triggered by the $\overline{Q_A}$ output of the previous stage instead of Q_A output of the previous stage.

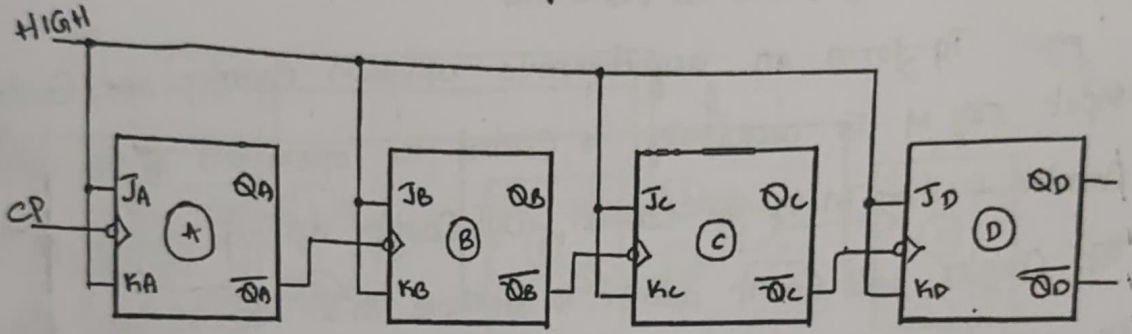
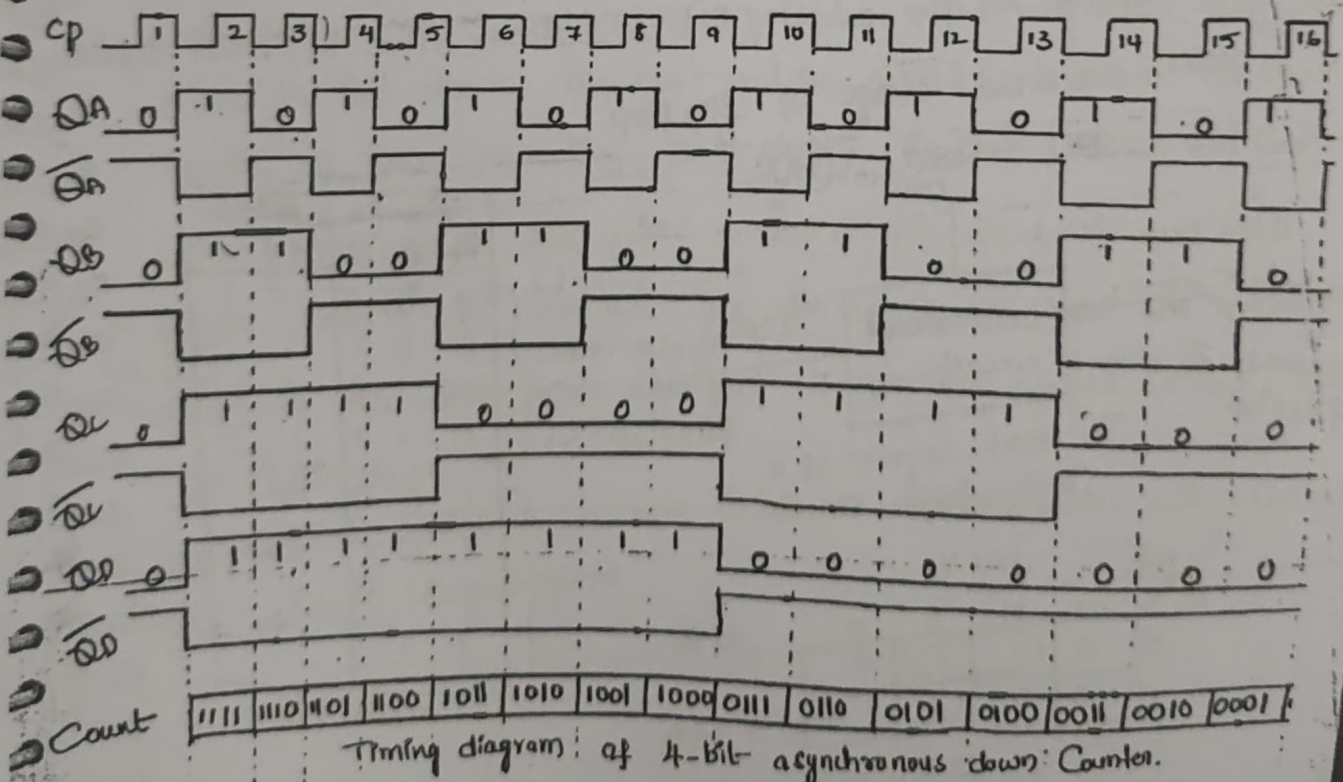


Fig: 4 bit asynchronous down Counter.

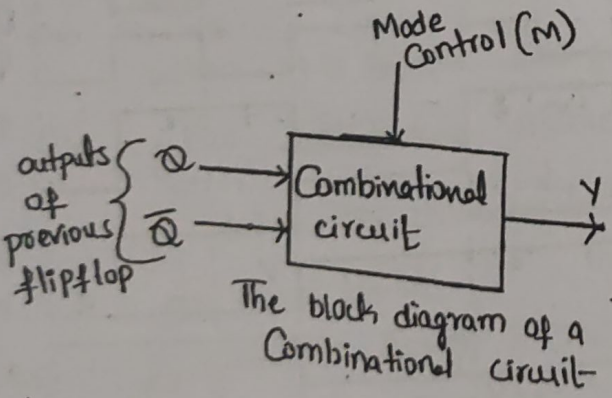


Timing diagram of 4-bit asynchronous down Counter.

Down Counters are not as widely used as up Counters. They are used in situations where it must be known when a desired number of input pulses has occured. In these situations the down Counter is preset to the desired number and then allowed to count down as the pulses are applied. When the Counter reaches zero state it is detected by a logic gate whose output then indicates that the preset number of pulses have occured.

Asynchronous up/Down Counter:

To form an asynchronous up/down Counter one Control input say M is necessary to Control the operation of the up/down Counter. When M=0, the Counter will Count up and when M=1, the Counter will Count down. To achieve this the M input should be used to Control whether the normal flip flop output (Q) or the inverted flip-flop output ($\bar{Q}$) is fed to drive the clock signal of the successive stage of flip flop.



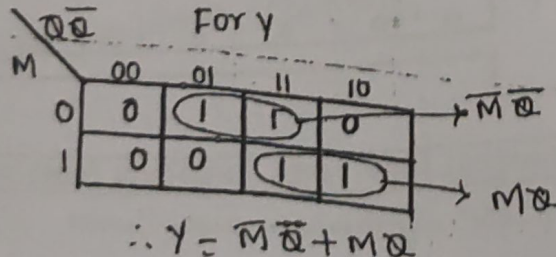
The block diagram of a Combinational circuit

Inputs			output
M	Q	$\bar{Q}$	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

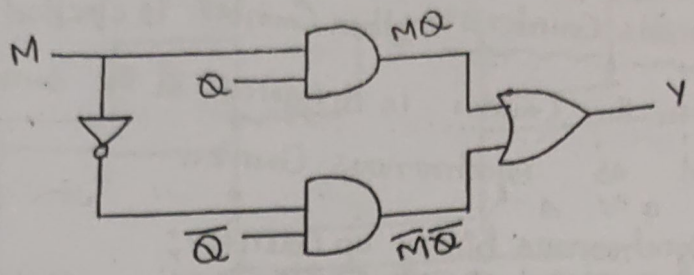
M=0 → $Y=\bar{Q}$ for down Counting

M=1 → $Y=Q$ for up Counting

Truth table



$\therefore Y = M\bar{Q} + MQ$



Logic diagram.

3-bit up/down Counter: The 3 bit up/down Counter will Coun from 000 upto 111 when the mode Control input M is 1 and from 111 down to 000 when mode Control input M is 0.

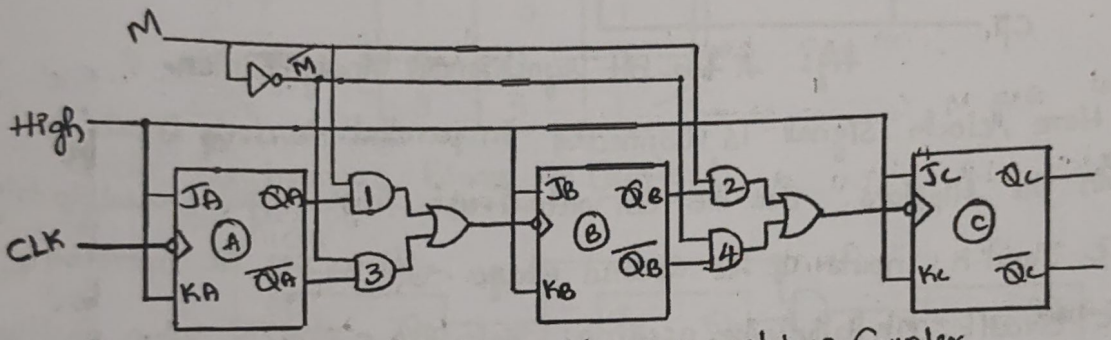


fig: 3 bit asynchronous up/down Counter.

A logic 1 on M enables AND gates 1 and 2 disables AND gates 3 and 4. This allows the Q_A and Q_B outputs to drive the clock inputs of their respective next stages. so that Counter will Count up when M is logic 0. AND gates 1 and 2 are disabled and AND gates 3 and 4 are enabled. This allows the $\overline{Q_A}$ and $\overline{Q_B}$ outputs to drive the clock inputs of their respective stages so that Coun will Count down.

Assignment: Design a 4-bit up/down ripple Counter with a Control for up/down Counting.

Synchronous Counters: When Counter is clocked such that each flipflop in the Counter is triggered at the same time, the Counter is called as Synchronous Counter.

2 bit Synchronous binary up Counter:

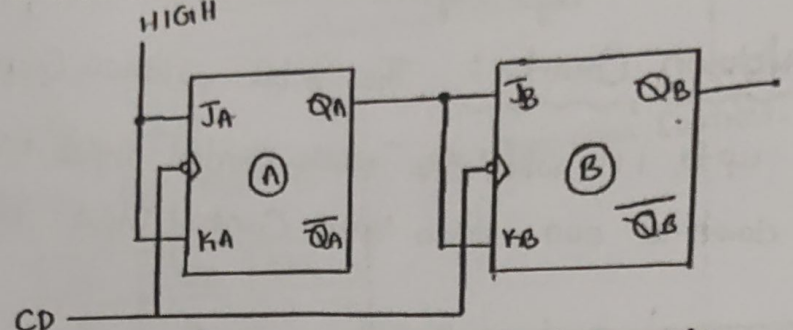
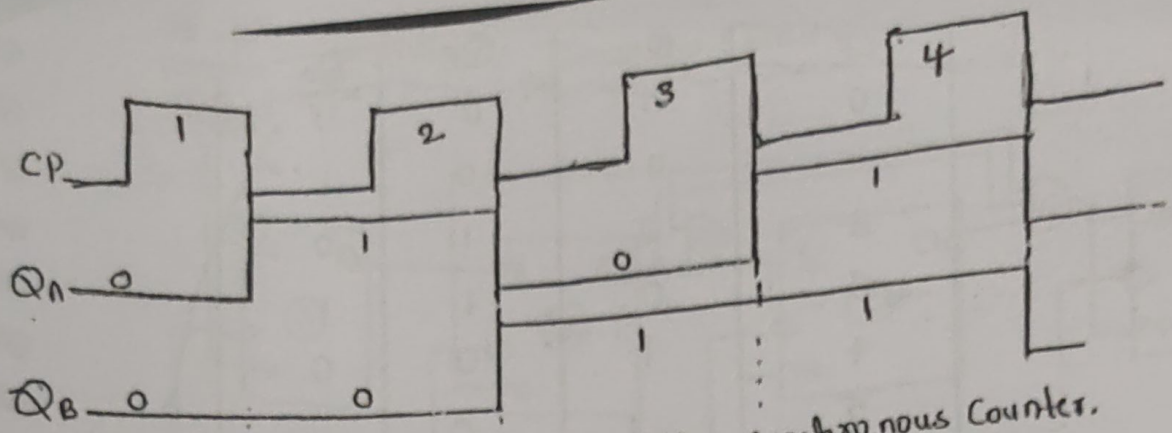


fig: A two-bit synchronous binary Counter.

Here, clock signal is Connected in parallel to clock inputs of both the flipflops. But the Q_A output of first stage is used to drive the J and K inputs of the second stage. let us see the operation of the circuit. Initially we assume that $Q_A = Q_B = 0$. When positive edge of the first clock pulse is applied, flipflop A will toggle because $J_A = K_A = 1$, whereas flipflop B output will remain zero because $J_B = K_B = 0$. After first clock pulse $Q_A = 1$ and $Q_B = 0$. At negative going edge of the second clockpulse both flip flops will toggle because they both have a toggle Condition on their J and K inputs ($J_A = K_A = J_B = K_B = 1$). Thus After second clock pulse, $Q_A = 0$ and $Q_B = 1$. At negative going edge of the third clock pulse flipflop A toggles making $Q_A = 1$, but flipflop B remains set i.e $Q_B = 1$. Finally at the leading edge of the fourth clockpulse both flipflops toggle as their JK inputs are at logic 1. This results $Q_A = Q_B = 0$ and Counter recycled back to its original state.



Timing diagram for the 2 bit synchronous Counter.

CP	Q_B	Q_A
0	0	0
1	0	1
2	1	0
3	1	1

3-bit Synchronous Binary up Counter:

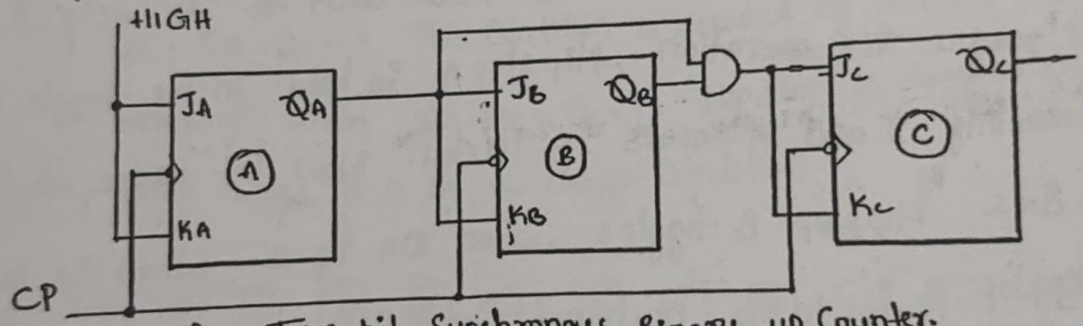
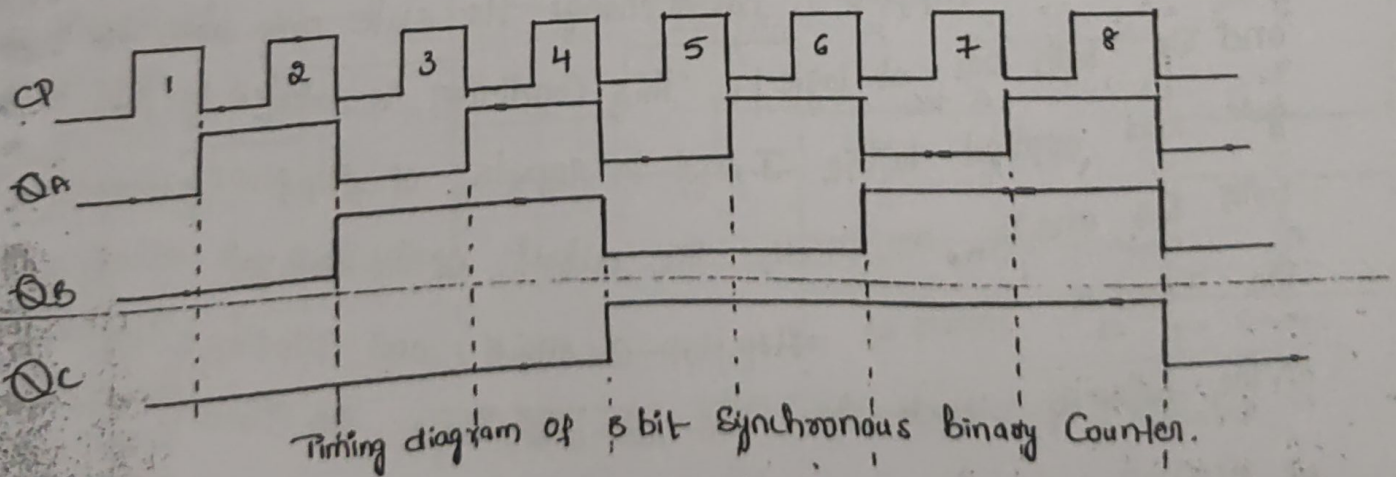


fig: Three bit Synchronous Binary up Counter.



Timing diagram of 3 bit Synchronous binary Counter.

CP	Q_C	Q_B	Q_A
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

Q_A changes on each clock pulse as we progress from its original state to its final state and then backs to its original state. To produce this operation, flipflop A is held in the toggle mode by connecting J and K inputs to HIGH. Now let us see what flipflop B does. Flipflop B toggles, when Q_A is 1. When Q_A is a 0, flipflop B is in the no change condition and remains in its present state. flipflop C has to change its state only when Q_B and Q_A both are at logic 1. This condition is detected by AND gate and applied to the J and K inputs of flipflop C. Whenever both Q_A and Q_B are HIGH, the output of the AND gate makes the J and K inputs of flipflop C HIGH, and flipflop C toggles on the following clock pulse. At all other times, the J and K inputs of flipflop C are held LOW by the AND gate output, and flipflop does not change state.

4-bit Synchronous Binary up Counter:

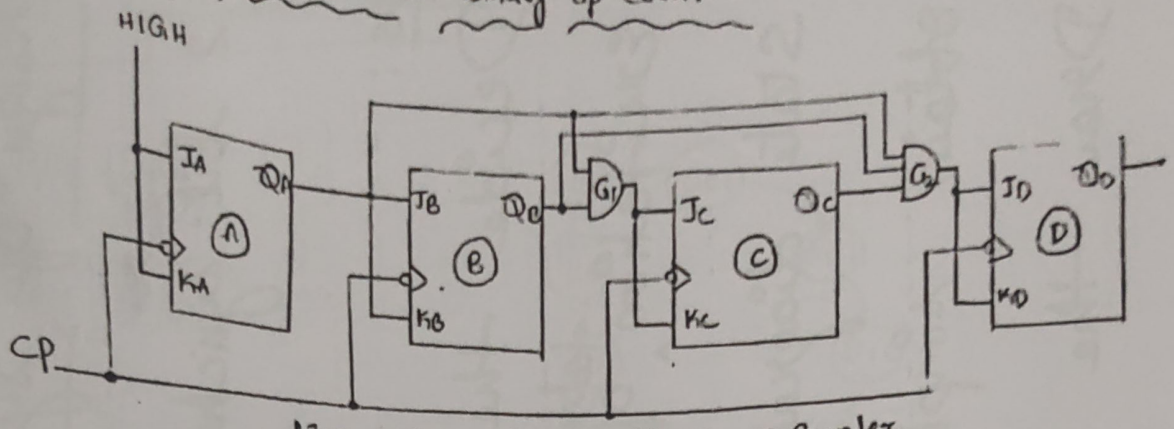
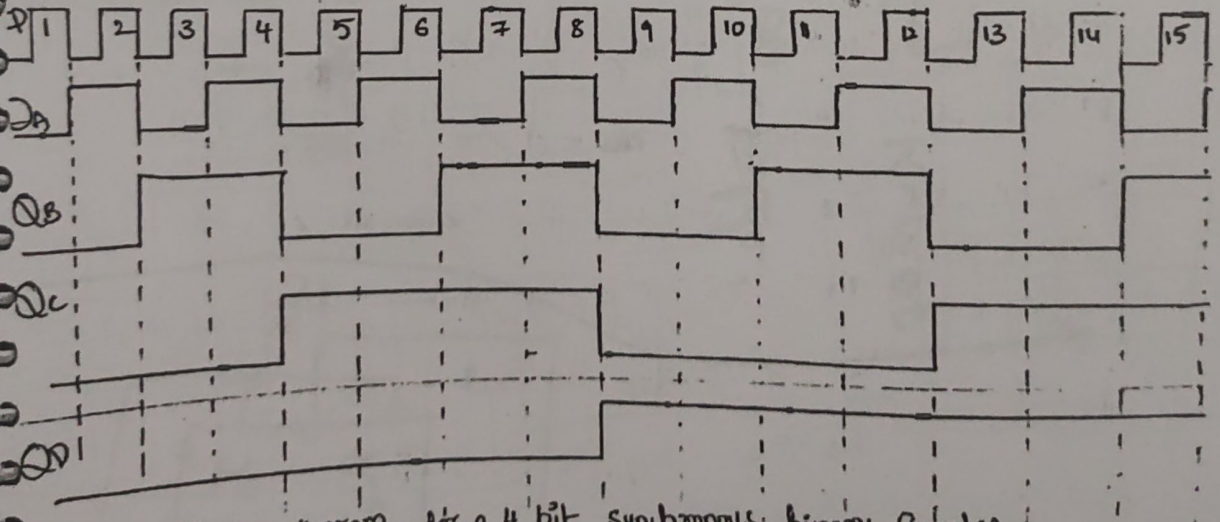


fig: 4 bit synchronous binary up Counter.

The 4 bit Synchronous binary up Counter is implemented with negative edge triggered flipflops, the transitions occur at the negative edge of the clock pulse. In this circuit, first three flipflops work same as 3-bit Counter.

For the fourth stage, flipflop has to change the state when $Q_A = Q_B = Q_C = 1$. This Condition is decoded by 3-input AND gate G_2. Therefore when $Q_A = Q_B = Q_C = 1$, flipflop D toggles and for all other times it is in no change Condition.



Timing diagram for a 4 bit synchronous binary Counter.

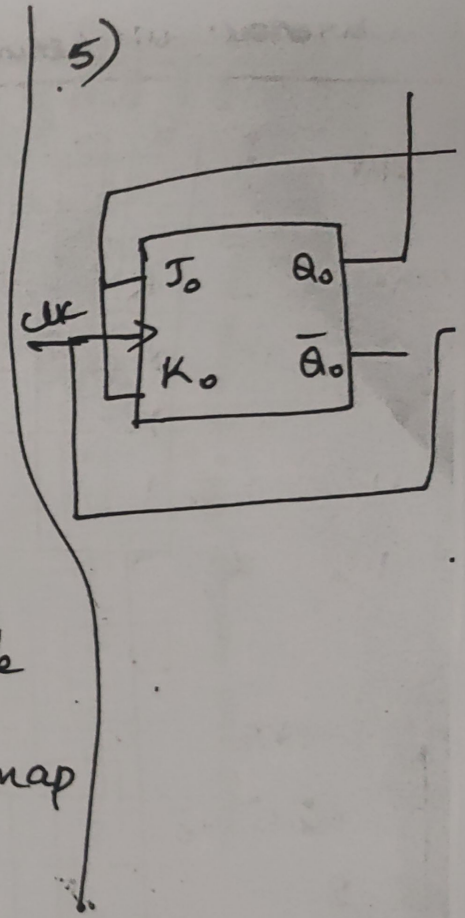
Design of synchronous counters

→ 2 bit synchronous counter

Steps:

1) Decide the no. of flip flops
2) Excitation table of flip flop
3) State diagram & circuit excitation table
4) Obtain simplified equations using K-map
5) Draw the logic diagram

5)



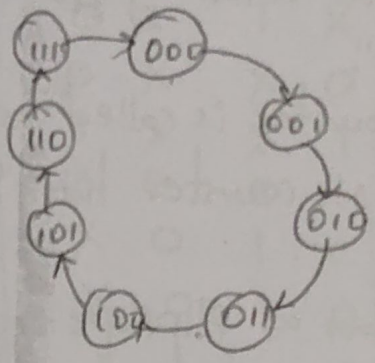
3 bit synchronous counter

→1) 3 flip flops (T)

→2)

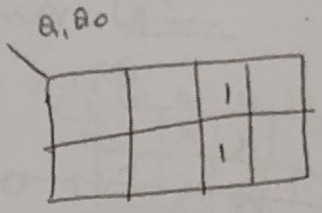
Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

3)



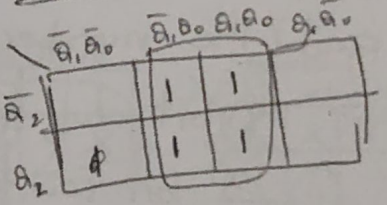
PS			NS			i/p		
Q_2	Q_1	Q_0	Q_2^*	Q_1^*	Q_0^*	T_2	T_1	T_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

4) T_2



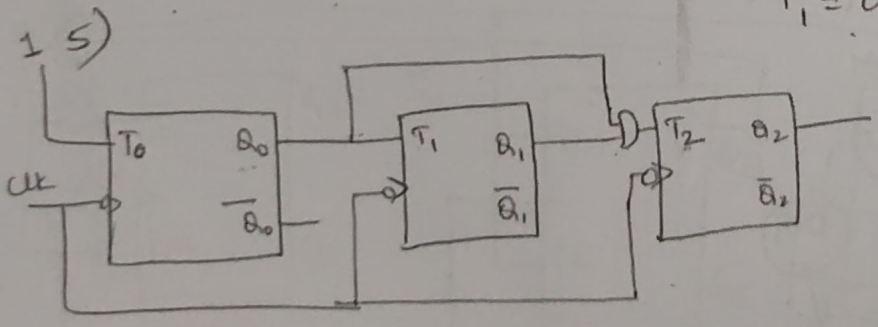
$T_2 = Q_1 Q_0$

T_1

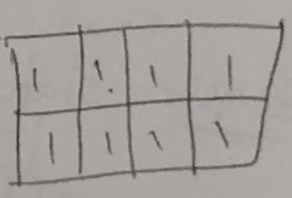


$T_1 = Q_0$

5)



T_0



$T_0 = 1$

Modulus of counter

→ The number of states through which the counter can passes before returning to initial state is called "modulus of counter".

→ modN counter is called as divided by N counter. since that modD-N counter has "N" number of distinctive states.

→ The number of flipflops required to construct modD-N counter are $N \leq 2^n$

n = number of flipflops required

N = number of distinctive states

---

1) Design a 3 bit synchronous counter using JK-flipflop

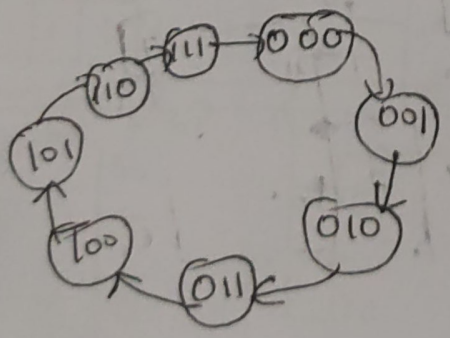
step1:- Find the number of flipflops required

n bit counter have n flipflops

∴ 3 bit counter = 3 flipflops.

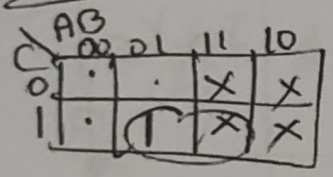
∴ number of distinctive states = $2^3 = 8$

step2:- state diagram



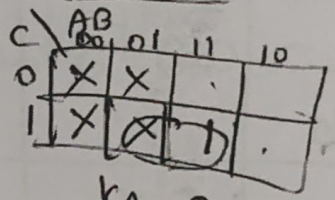
PS (Present state) $Q_A Q_B Q_C$	Next state (NS) $Q_A(t+1)\ Q_B(t+1)\ Q_C(t+1)$	J_A K_A	J_B K_B	J_C K_C
0 0 0	0 0 1	0 X	0 X	1 X
0 0 1	0 1 0	0 X	1 X	X 1
0 1 0	0 1 1	0 X	X 0	1 X
0 1 1	1 0 0	1 X	X 1	X 1
1 0 0	1 0 1	X 0	0 X	1 X
1 0 1	1 1 0	X 0	1 X	X 1
1 1 0	1 1 1	X 0	X 0	1 X
1 1 1	0 0 0	X 1	X 1	X 1

K map for J_A



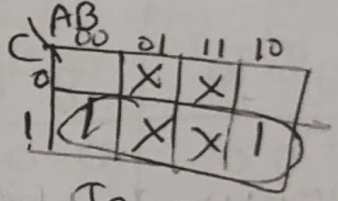
$J_A = BC$ i.e. $Q_B Q_C$

K map for K_A



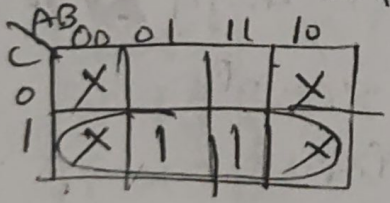
$K_A = BC$ $Q_B Q_C$

J_B



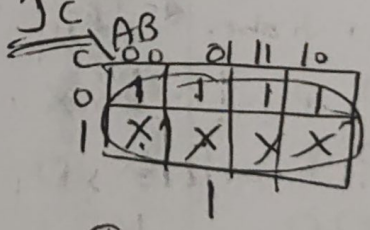
$J_B = C$ Q_C

K_B



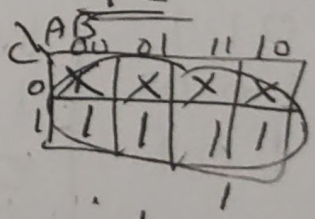
$K_B = C$ Q_C

J_C

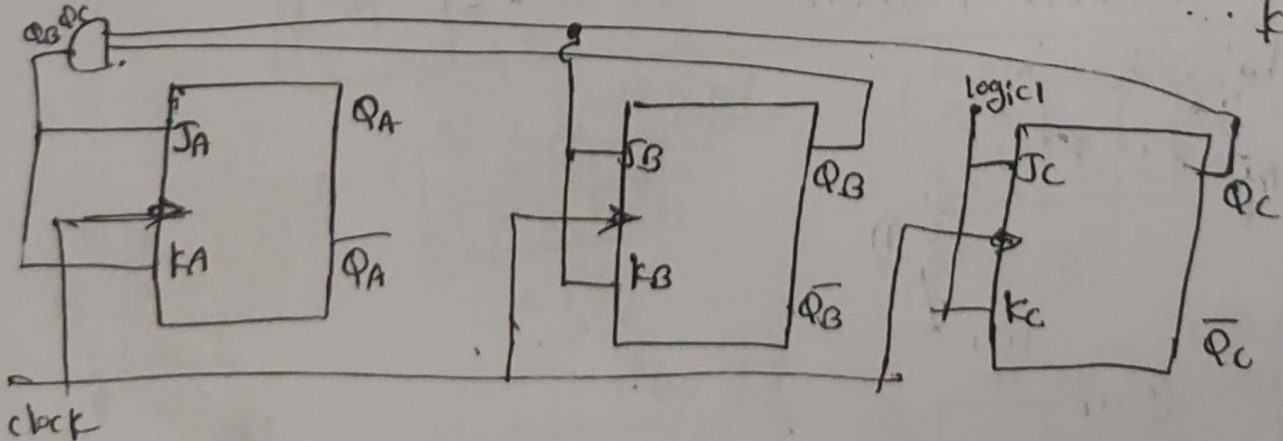


$J_C = 1$

K_C



$\therefore K_C = 1$



(2) Design a mod-6 counter using JK flipflop

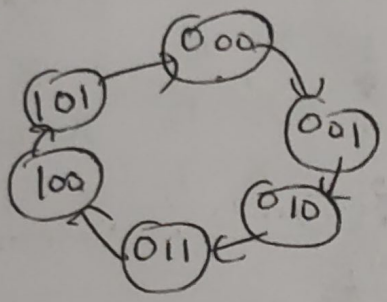
Soln:- mod 6 counter has 6 distinctive states

$N \leq 2^n$; $6 \leq 2^n$

$n = 3$

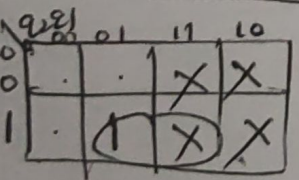
∴ 3 flipflops, 6 distinctive states.

state diagram:-

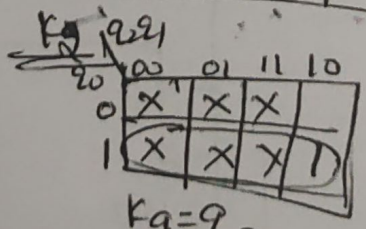


PS $q_2 q_1 q_0$	NS $Q_2 Q_1 Q_0$	$J_2 K_2$	$J_1 K_1$	$J_0 K_0$
0 0 0	0 0 1	0 X	0 X	1 X
0 0 1	0 1 0	0 X	1 X	X 1
0 1 0	0 1 1	0 X	X 0	1 X
0 1 1	1 0 0	1 X	X 1	X 1
1 0 0	1 0 1	X 0	0 X	1 X
1 0 1	0 0 0	X 1	0 X	X 1

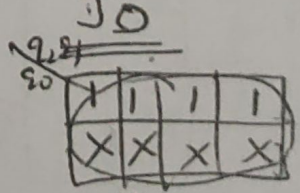
K-map for J_2



$J_2 = q_1 q_0$

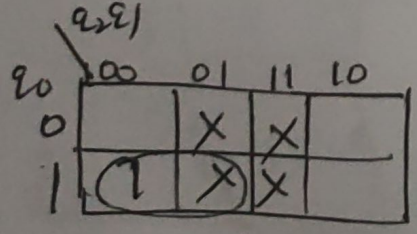


$K_2 = q_0$



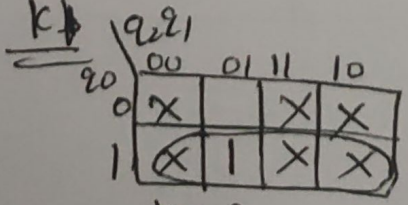
$J_0 = 1$

J_1



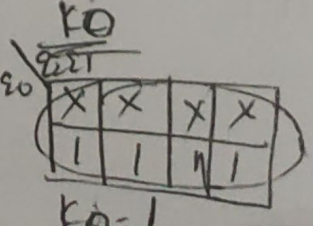
$J_1 = q_2' q_0$

K_1



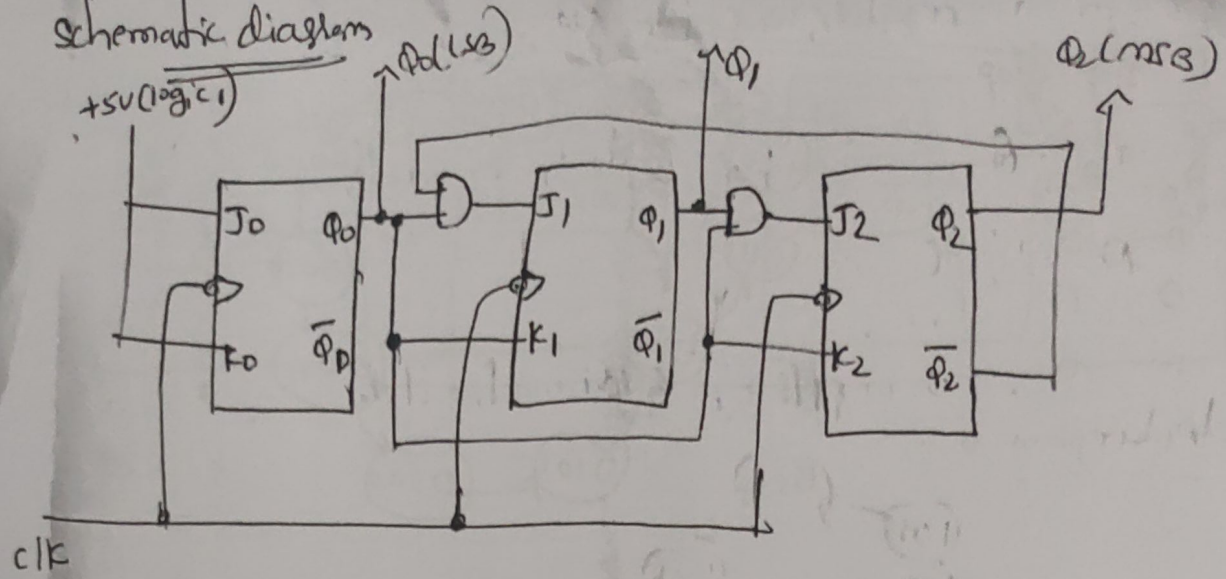
$K_1 = q_0$

K_0



$K_0 = 1$

Schematic diagram



circuit diagram of mod-6 synchronous counter

Problems

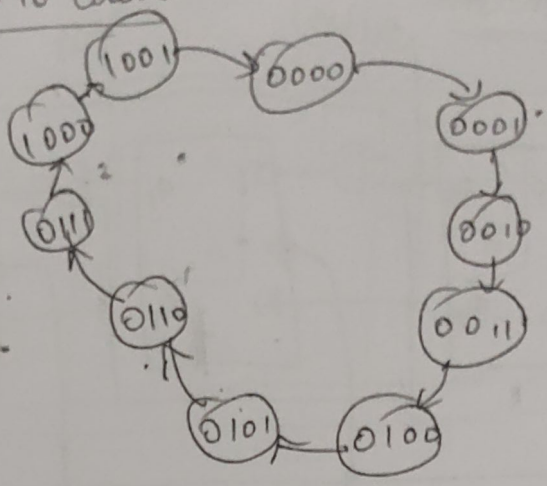
① Design a mod-3 counter, mod-5 counter using JK flip flops

1) T flip flop

2)

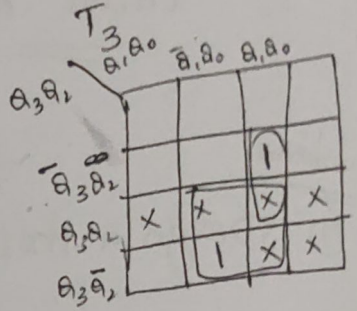
Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Mod-10 Counter



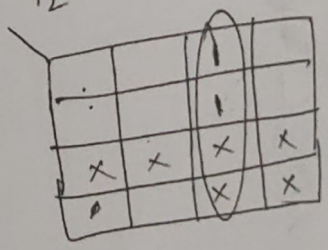
PS	NS	T_3	T_2	T_1	T_0
$Q_3\ Q_2\ Q_1\ Q_0$	$Q_3^*\ Q_2^*\ Q_1^*\ Q_0^*$				
0 0 0 0	0 0 0 1	0	0	0	1
0 0 0 1	0 0 1 0	0	0	1	1
0 0 1 0	0 0 1 1	0	0	0	1
~~0 0 1 1~~	~~0 1 0 0~~				
0 0 1 1	0 1 0 0	0	1	1	1
0 1 0 0	0 1 0 1	0	0	0	1
0 1 0 1	0 1 1 0	0	0	1	1
0 1 1 0	0 1 1 1	0	0	0	1
0 1 1 1	1 0 0 0	1	1	1	1
1 0 0 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

T_3



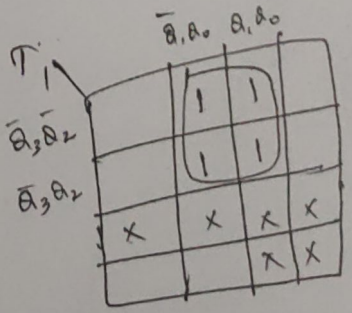
$T_3 = Q_3Q_0 + Q_3Q_1Q_0$

T_2



$T_2 = Q_1Q_0$

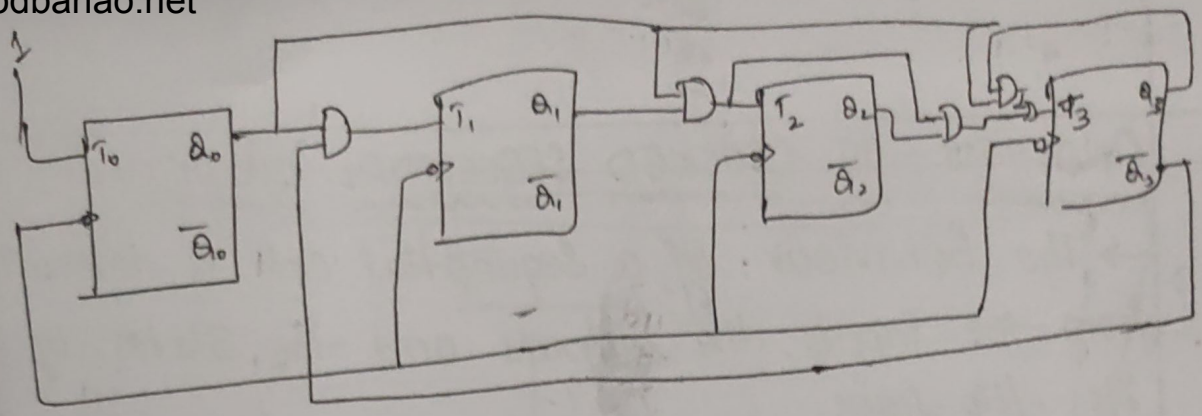
T_1

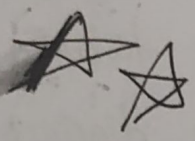


$T_1 = \bar{Q}_3Q_0$

$T_0 = 1$

24,500





ANALYSIS OF CLOCKED SEQUENTIAL CIRCUITS:-

→ The behaviour of a sequential ckt is determined from the inputs, the outputs and the states of its flip-flops.

→ Both the outputs and next state are a function of the inputs & present state.

The analysis of a sequential ckt consists of -

→ obtaining state table

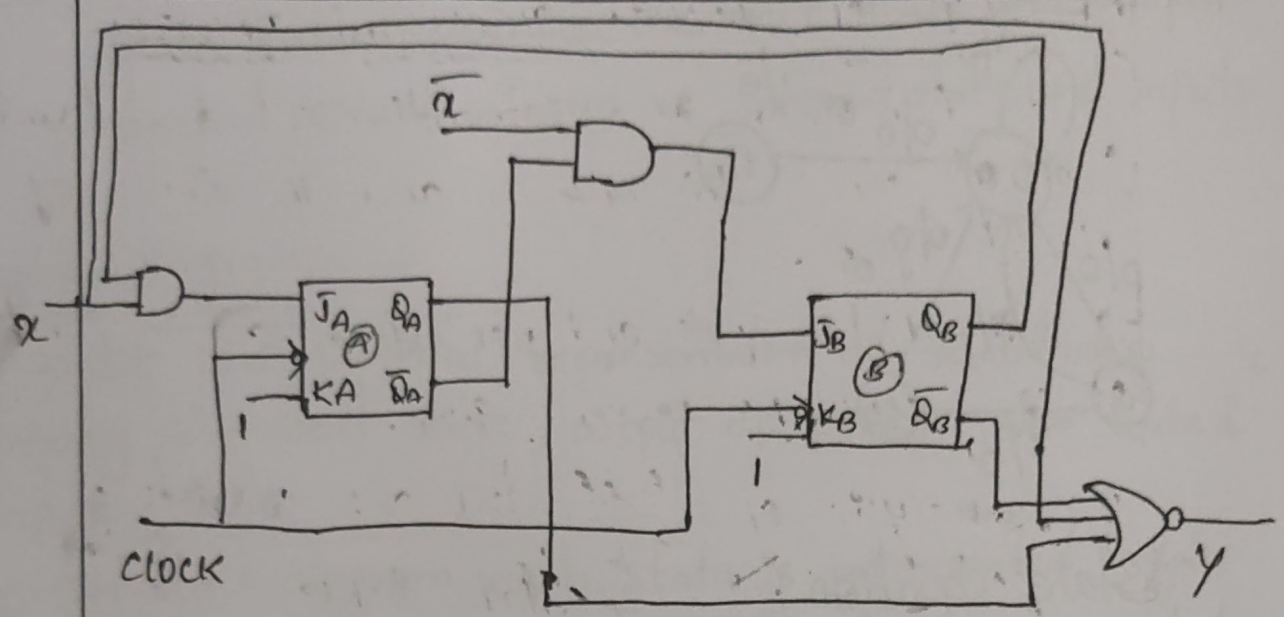
→ obtaining state diagram

→ write state equations.

State table:- The time sequence of i/ps, o/p's and flip-flops state may be enumerated in a table which is known as state-table. It consists of three sections like present state, next state and output.

State diagram:- The graphical representation of state table, which conveys same information as state table.

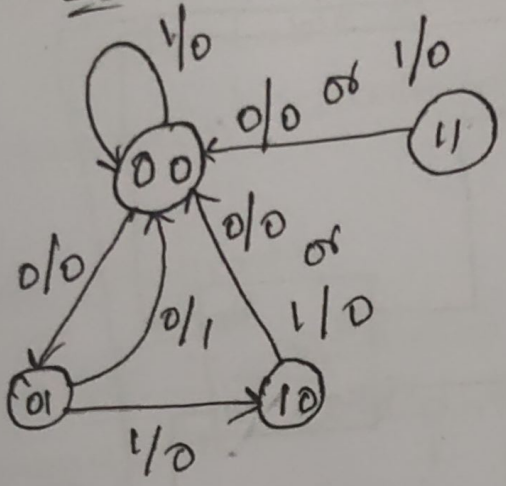
State Equation:- A state equation is an algebraic expression that specifies the condition for a flip-flop state transition.



State table:

Present state		Next state				O/P	
		$x=0$		$x=1$		$x=0$	$x=1$
A	B	A	B	A	B	y	y
0	0	0	1	0	0	0	0
0	1	0	0	1	0	1	0
1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0

State diagram:



State Equation:

$A(n+1) = \bar{A}BX$

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

State table:-

State table specifies the next state and output of sequential circuit interms of Present state and inputs.

State Diagram:-

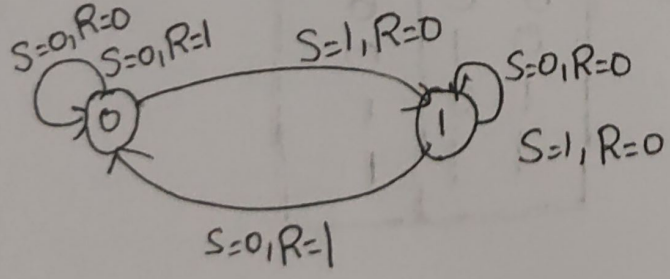
→ It is a pictorial representation of relationship among the next state, output of present ~~sate~~ state & inputs.

→ In state diagram each state is represented by a "circle" and the transition between states represented by "arc"s.

→ The label of each "arc" as a notation "x/y" where "x" is input and "y" is output.

State diagrams for various flipflops

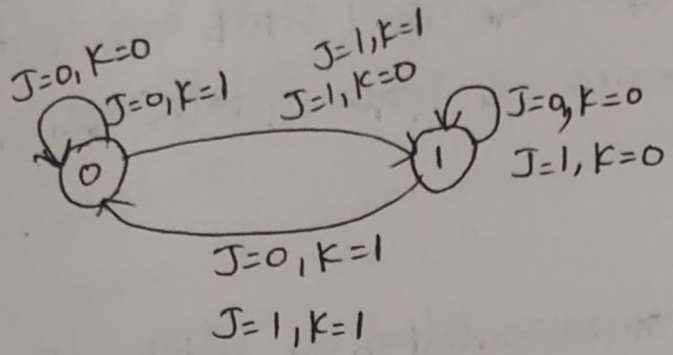
SR flipflop



S	R	Q_n	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	X
1	1	1	X

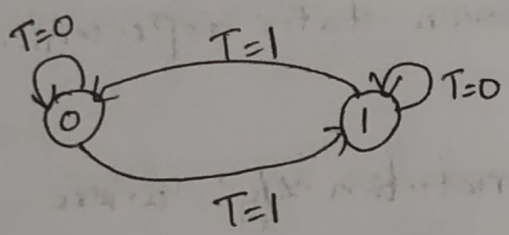
ment

JK flipflop



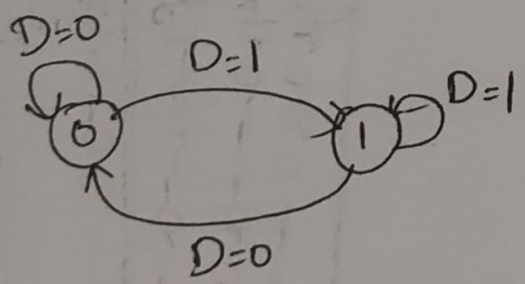
J	K	Qn	Qn+1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

T flipflop



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

D flipflop



D	Qn	Qn+1
0	0	0
0	1	0
1	0	1
1	1	1

VImp

State reduction and state assignment

→ The reduction of number of flipflops in a sequential circuit is referred to as "state reduction".

→ In general "n" number of flipflops will produce "2^n" number of states.

→ For the purpose of state reduction we have to identify the equivalent states.

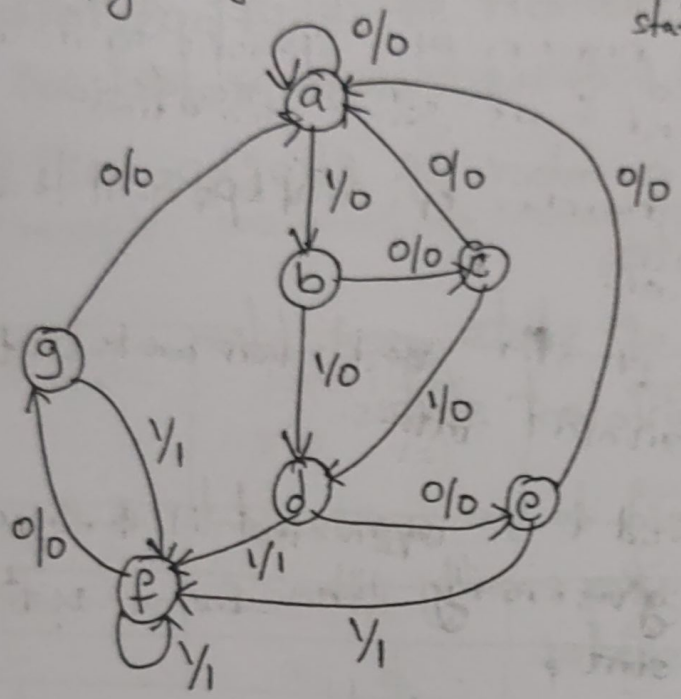
→ 2 states are said to be equivalent if for each input condition gives exactly the same output and the same next states.

→ when 2 states are equivalent we can remove one of the state without altering the input, output equations.

State assignment:- The ~~binary~~ process of assigning binary values to the states of the sequential machine is known as "state assignment.

WIMP

Design a sequential circuit for the state diagram as shown in figure. (Draw the reduced state table, state diagram and state assignment)



$\frac{x}{y} = \frac{\text{input}}{\text{output}}$

Soln:

Present state	Next state $x=0$	Next state $x=1$	Present state output (z) $x=0$	Present state output (z) $x=1$
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

→ The states e and g are equal, since, they produce same output & the same next state.

→ Now, we can remove the either one of the state e (or) g.

→ For example, we want to remove "g" replace "g" with "e".

∴

Present state	Next state x=0	Next state x=1	output (z) x=0	output (z) 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

→ In the above state table 2 states "d" and "f" are equal. Now we can remove any one of the state d (or) f, For Eg. if we remov "f" then replace "f" with "d".

Reduced state table

Present state	Next state x=0	Next state x=1	output (z) x=0	output (z) 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

→ The number of flipflops required $\log_2 5$ (∵ reduced state table is having 5 Present state)

$\therefore \log_2 5 > 2$

∴ 3 flipflops are required to design a sequential circuit

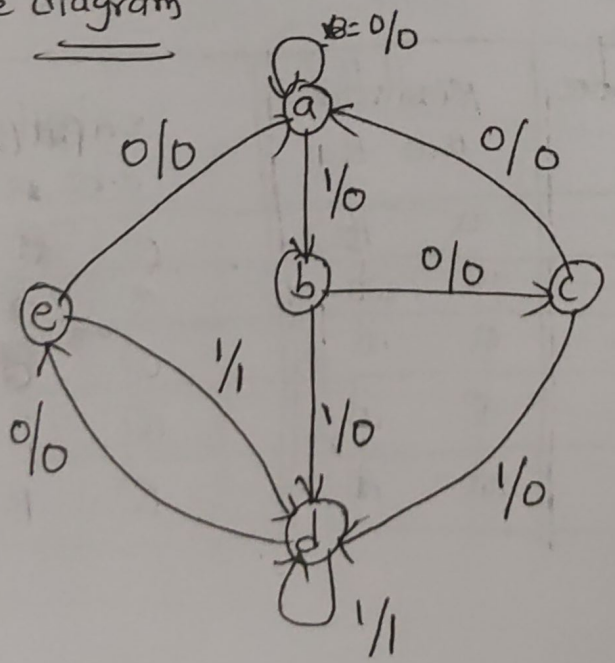
∴ we have to assign 3 bit binary numbers to each state.

Assign a → 000
b → 001
c → 010
d → 011
e → 100

Binary assignment

Present state A BC	Next state ($A^+B^+C^+$) x=0	x=1	output (z) x=0	x=1
000	000	001	0	0
001	010	011	0	0
010	000	011	0	0
011	100	011	0	1
100	000	011	0	1

Reduced state diagram



(2) Reduce the number of states in the following state and assign the binary numbers

Present state	Next state x=0	Next state x=1	output(z) x=0	output(z) x=1
a	f	b	0	0
b	d	c	0	0
c	f	e	0	0
d	g	a	1	0
e	d	c	0	0
f	f	b	1	1
g	g	h	0	1
h	g	a	1	0

Soln:-

From the given table states "d" and "h" are equal. Since, they produce same output and the same nextstate.

→ Now we can remove either one of the state e (or) g

Ex: if we remove "h" replace h with "d"

∴

Present state (PS)	(NS) Nextstate x=0	Nextstate x=1	output (z) x=0	output (z) x=1
a	f	b	0	0
b	d	c	0	0
c	f	e	0	0
d	g	a	1	0
e	d	c	0	0
f	f	b	1	1
g	g	d	0	1

→ b, e are equal ∴ remove e & replace "e" with "b"

PS	NS x=0	NS x=1	O/P(z) x=0	O/P(z) x=1
a	f	b	0	0
b	d	c	0	0
c	f	b	0	0
d	g	a	1	0
f	f	b	1	1
g	g	d	0	1

state "c" can be eliminated

→ states a & c are equal ∴ remove "c" & replace "c" with "a"

PS	NS x=0	NS x=1	O/P(z) x=0	O/P(z) x=1
a	f	b	0	0
b	d	a	0	0
d	g	a	1	0
f	f	b	1	1
g	g	d	0	1

∴ This is the final reduced state.

reduced state diagram