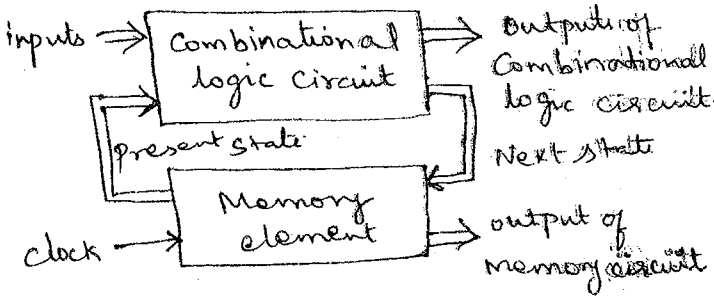
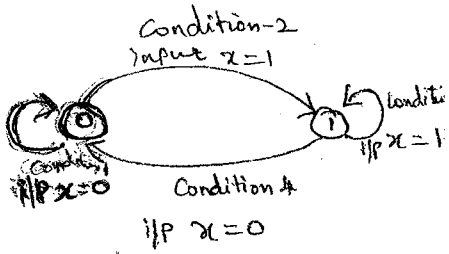
UNIT-III

SYNCHRONOUS SEQUENTIAL LOGIC

When the combinational circuit output not only depends on the present state but also the previous state, the circuit is known as sequential logic circuit.



Block diagram of sequential logic circuit



Two state machine

* The output of combinational circuit is stored in memory elements, and is used as one of the input variable for combinational circuit
* The output of combinational circuit depens on external inputs and input from memory element.
* A memory element is a device which can store information interms of 1 or 0 and its state can be modified by clock signal and data inputs
* A flip-flop is one bit memory element which can store 1 or 0.

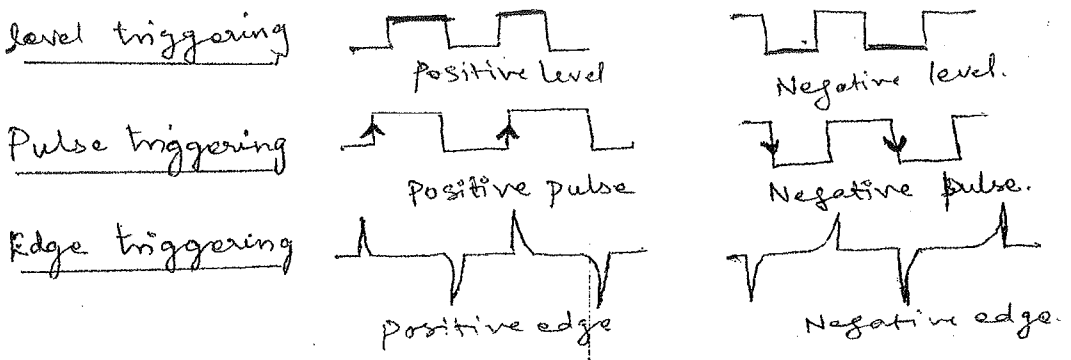
LATCHES

* Latch is a bistable device capable of staying in either of 2 states : set and reset for an indefinite time period.
* Latches are similar to flipflops as they have two states.
* But the difference between latches and flip-flops is in the method of changing their states

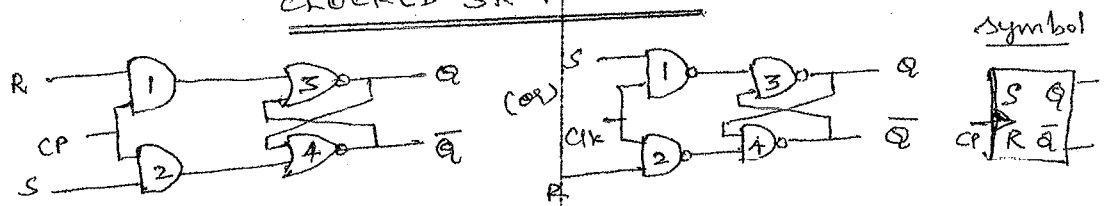
Latch	Flip-flop.
* A latch is an electronic sequential logic circuit used to store information in an asynchronous arrangement.	*A flip-flop is an electronic sequential logic circuit used to store information in an synchronous arrangement. It has two stable states and maintains its states for an indefinite period until a trigger pulse is applied.
* One latch can store one bit information, but output state changes only in response to data input.	* One flip-flop can store one bit data, but output state changes with trigger pulse only.
* Latch is an asynchronous device and it has no clock input.	* flip-flop has clock input and its output is synchronised with clock pulse.
* Latch holds a bit value and it remains constant until new inputs force it to change.	*flip-flop holds a bit value and it remains constant until a trigger pulse is received.
* Latches are level-sensitive and the output tracks the input when the level is high. Therefore as long as the level is logic level 1, the output can change if the input changes.	*. Flip-flops are edge-Sensitive They can store the input Only when there is either a rising or falling edge of the clock.

FLIP FLOPS

* A flipflop is a device which changes its state at the times when a change is taking place in clock signal. The flip-flop is triggered by either positive (leading) edge or negative (trailing) edge of clock signal.
* In edge triggering, the output of flip-flop can be changed only when the clock pulse is applied.
* If the clock signal changes from low to high state and the output changes due to inputs, it is called positive edge triggering.
* When the clock signal changes from high to low state and the output changes due to inputs, this condition is called negative edge triggering.



CLOCKED SR FLIP-FLOP.



Case 1:

for $S=0, R=0, Clk=0$, the flipflop remains in its present state. Q remains unchanged even for $S=0, R=0$ and $clk=1$, flip-flop remains in its present state. This condition will not affect the output of flip-flop $Q_n = Q_{n+1}$.

Case 2:

$S=0, R=1, clk=0$, flip flop remains in its

Present state, when clock = 1, $Q_{n+1} = 0$. Thus S=0, R=1, clk=1 flipflop resets to '0' state

case 3:

S=1, R=0, clk=0, flip flop remain in its Present state. But for S=1, R=0 and clk=1, set state of flip-flop is reaches flipflop to $Q_{n+1}=1$.

case 4:

S=1, R=1, clk=1, indeterminate state $\bar{Q}$ & Q_{n+1} both are same. ∴ It is indeterminate

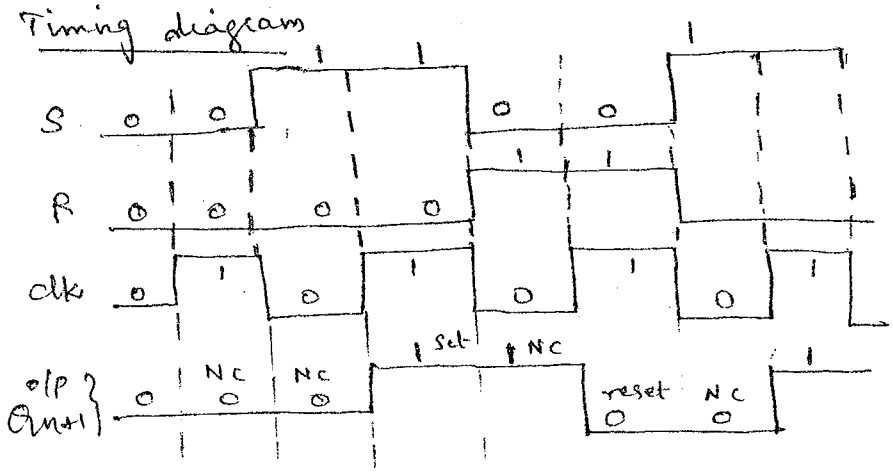
clk	S	R	Q_n	Q_{n+1}	state
1	0	0	0	0	No change
1	0	0	1	1	
1	0	1	0	0	reset
1	0	1	1	0	
1	1	0	0	1	set
1	1	0	1	1	
1	1	1	0	X	Indeterminate
1	1	1	1	X	
0	X	X	0	0	No change.
0	X	X	1	1	

characteristic equation

S \ RQ_n	00	01	11	10
0	0	1	0	0
1	1	1	X	X

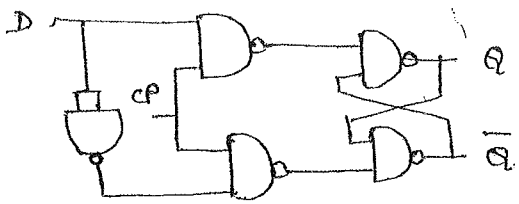
$Q_{n+1} = S + \bar{R} Q_n$.

Timing diagram

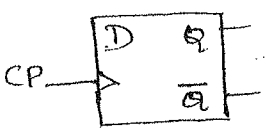


D-FLIPFLOP

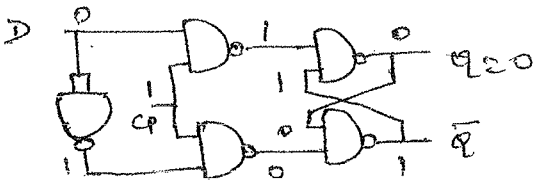
* D-latch with enable replaced as clock pulse is D flipflop.



symbol.

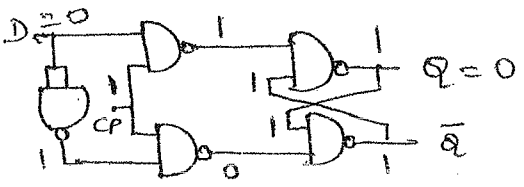


CP=1, D=0, Q=0



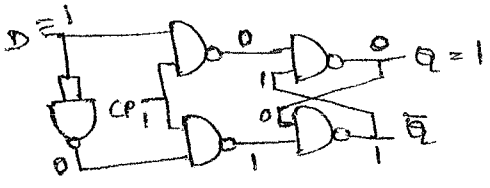
$Q_n = 0$, D=0, $Q_{n+1} = 0$.

CP=1, D=0, Q=1



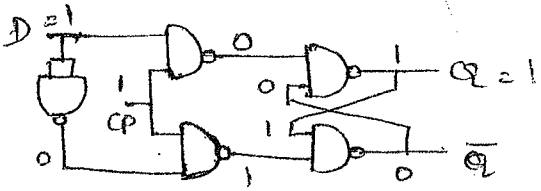
$Q_n = 1$, D=0, $Q_{n+1} = 0$.

CP=1, D=1, Q=0



$Q_n = 0$, D=1, $Q_{n+1} = 1$

CP=1, D=1, Q=1.

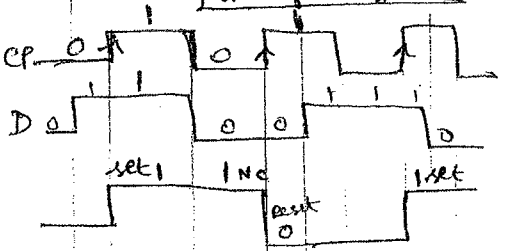


$Q_n = 1$, D=1, $Q_{n+1} = 1$.

IIIrdly When CP=0 whatever is the D input the next state is same as previous state.

CP	D	Q_n	Q_{n+1}	state
↑	0	0	0	reset
	0	1	0	
↑	1	0	1	set.
	1	1	1	
0	X	X	Q_n	No change.

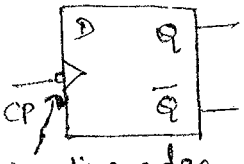
Timing diagram



Positive edge triggered D-flip flop.

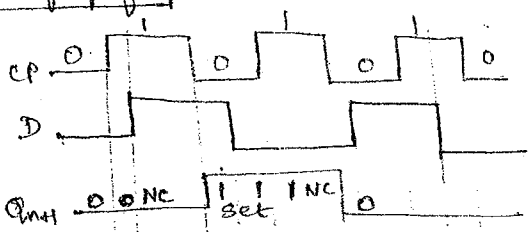
Negative edge triggered D flip flop.

symbol



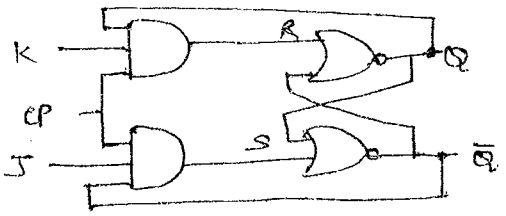
Negative edge triggered

CP	D	Q_{n+1}
↓	0	0
↓	1	1
0	X	Q_n.

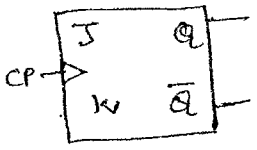


Timing diagram.

CLOCKED JK FLIPFLOP.



symbol.



* When J & K are both low, both AND gates are disabled. clock pulses has no effect and Q and $\bar{Q}$ retains their last values.
* When J is low and K is high, the lower AND gate is disabled, so theres no way to set the flipflop. The only possibility is reset. When Q is high, the upper gate passes a reset trigger as soon as the next positive clock pulse arrives.
* This forces Q to become low. Therefore, J=0 & K=1 means that the next positive clock pulse resets the flip flop unless Q is already reset.

Case 1: J → high, K → low } upper gate disabled, so there is no way to reset the flip-flop.

* The only possibility is to set the flip-flop if it is ~~to~~ set the flip-flop if it is not previously set.
* When Q=0, $\bar{Q}$ is high and hence the lower gate passes a SET trigger on the next positive clock pulse.

This drives Q into the high state.

Case 2:

When J = 1, k = 0 means next positive clock pulse set the flip-flop unless Q is already set.

Case 3:

When J & k both high, its possible to set or reset flipflop. If Q is high upper gate passes a reset trigger on the next positive clock edge.

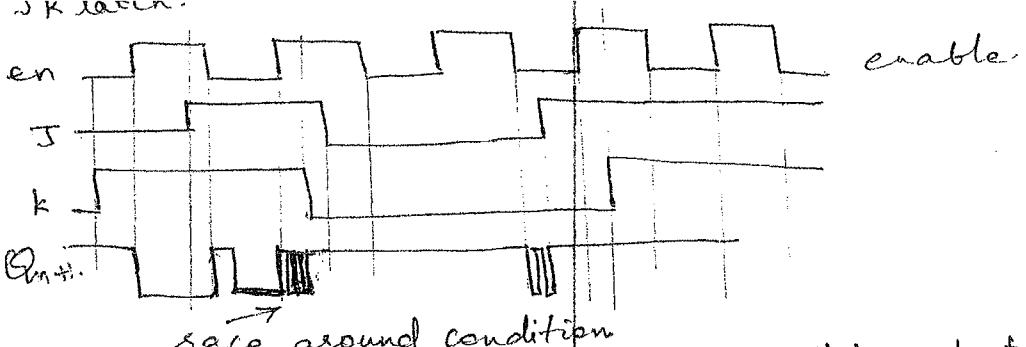
If Q is low, the lower gate passes a SET trigger on the next positive clock edge.

Q changes to complement of last state.

Case 4:

When J = k = 1, means output of the flipflop will toggle on the next positive clock edge.

JK latch.

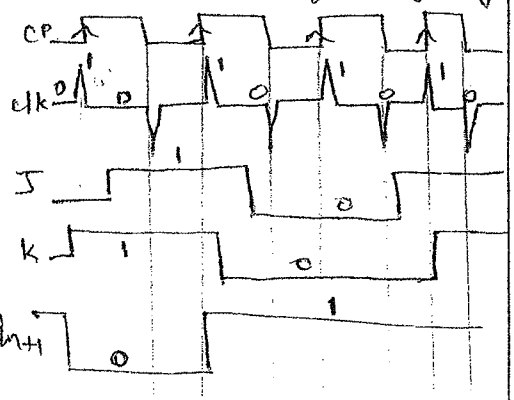


CP	J	K	Q_n	Q_{n+1}
↑	0	0	0	0
↑	0	0	1	1
↑	0	1	0	0
↑	0	1	1	0
↑	1	0	0	1
↑	1	0	1	1
↑	1	1	0	1
↑	1	1	1	0

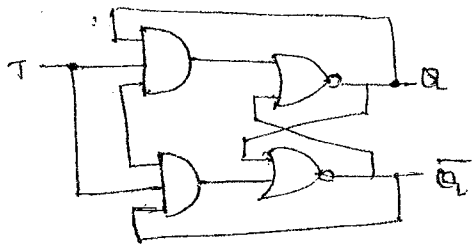
≡

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

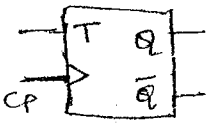
Positive edge triggering



T FLIPFLOP.



symbol



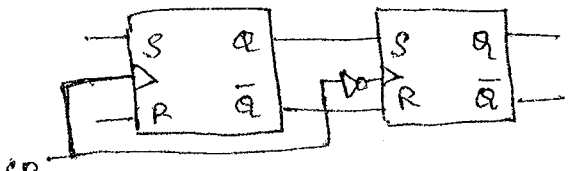
* When T=0, both AND gate are disabled no change in output.

* When T=1, J=1, K=1 output toggles.

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

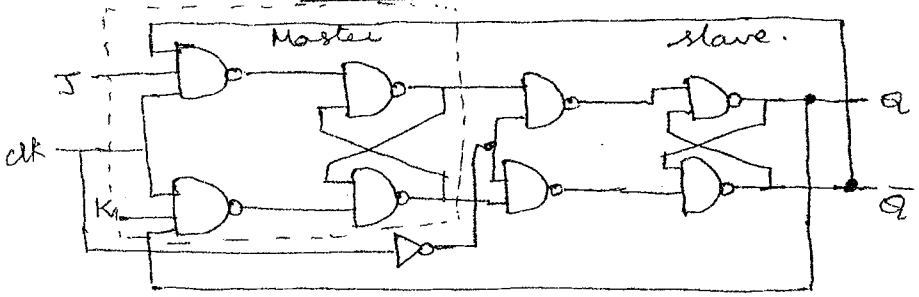
T	Q_{n+1}
0	Q_n
1	$\overline{Q_n}$

MASTER-SLAVE SR FLIP-FLOP.

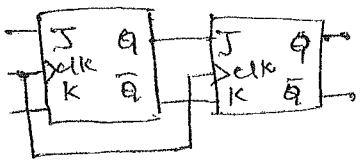


* Master slave flip flop is constructed from two flipflops.
* One circuit serves as master and the other as slave and the overall circuit is referred to as master-slave flip-flop.
* Master flip-flop, slave flip flop and a inverter
* flip flop's are positive level triggered, but inverter connected at clock input of slave flip-flop forces it to trigger at negative level.

MASTER SLAVE JK FLIPFLOP.

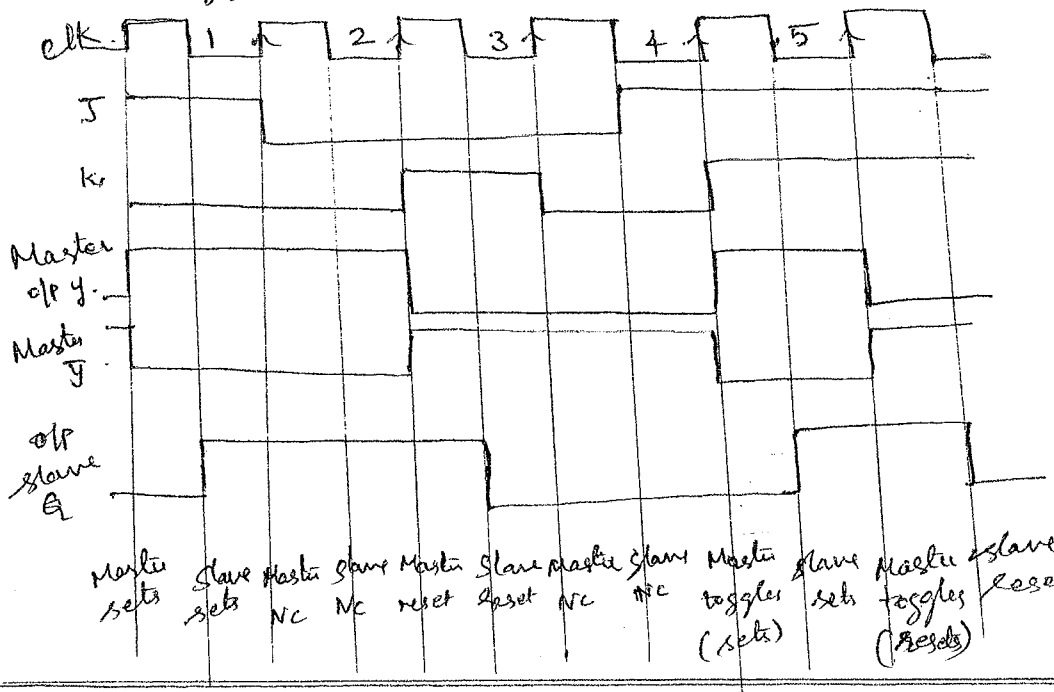


* clocked flipflop introduces some problems. If a flipflop is a level triggered one, it changes state according to its inputs as soon as the clock becomes 1.
* But due to feedbacks the inputs can change before the clock switches off to '0'.
* ∴ Another state transition occurs for the same clock pulse which is called racing ∴ Master slave is used.



* The first flipflop is called master driven by the positive edge of the clock pulse.
* Second flipflop called the slave is driven by the negative edge.

* When J=K=0, No change.
* " J=1, K=0, Master sets on the positive clock edge. high @ o/p of Master drives the J input of slave. When negative clock edge hits, slave also sets. The slave flipflop copies the action of master flipflop.
* When J=0, K=1, Master resets on the leading edge and slave resets on the trailing edge.
* When J=K=1 Master toggles on the positive edge and slave toggles on the negative edge.



FLIP FLOP EXCITATION TABLE

S-R flipflop.

* There are four possible transition from the present state to the next state.

S R truth table.

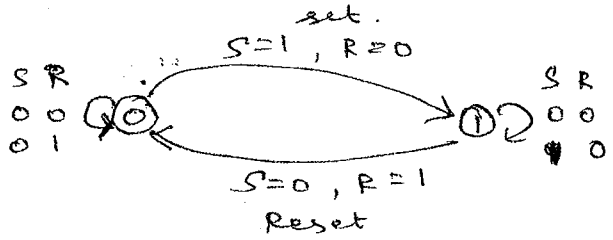
S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	x

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

SR 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

state transition diagram



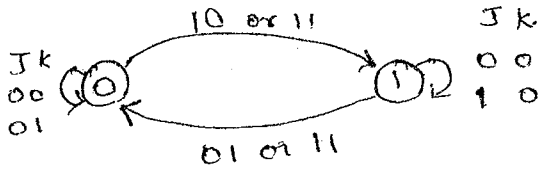
Jk flip flop.

J	k	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

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

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

State transition diagram



D-Flip flop

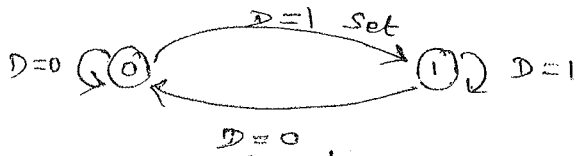
D	Q_{n+1}
0	0
1	1

D truth table.

D excitation table

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

State transition diagram



T-Flip flop / Complement flip flop.

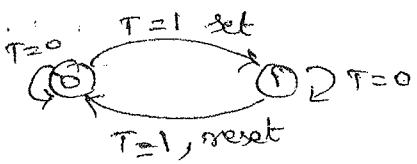
T truth table

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

T excitation table

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

State transition diagram

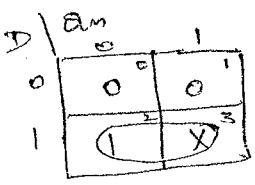


Conversion of flip-flop

SR flip flop to D flip flop

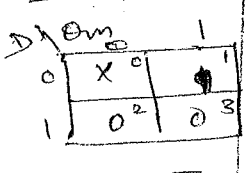
Input	Present	Next state	Flip flop input	
D	Q_n	Q_{n+1}	S	R
0	0	0	0	X
1	0	1	1	0
0	1	0	0	1
1	1	1	X	0

For S



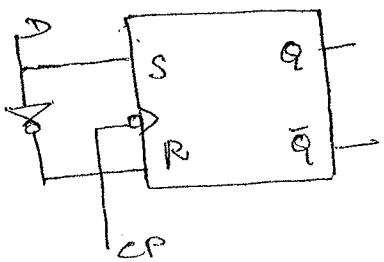
$S = D$

For R



$R = \bar{D}$

SR to D Flip flop Conversion



SR flip-flop to JK flipflop.

Inputs		Present	Next state	Flip flop input	
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

For S

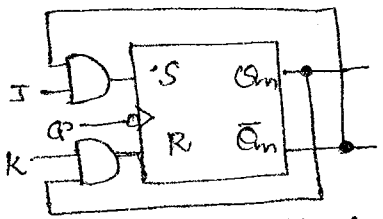
J \ KQ_n	00	01	11	10
0	0 (0)	X (1)	0 (3)	0 (2)
1	1 (4)	X (5)	0 (7)	1 (6)

$S = J\overline{Q_n}$

For R

J \ KQ_n	00	01	10	11
0	X (0)	0 (1)	1 (3)	X (2)
1	0 (4)	0 (5)	1 (7)	0 (6)

$R = K Q_n$



SR flipflop to JK flipflop.

SR flipflop to T flipflop.

Input	Q_n	Q_{n+1}	Flip-flop input	
T	Present	Next state	S	R
0	0	0	0	X
1	0	1	1	0
1	1	0	0	1
0	1	1	X	0

For S

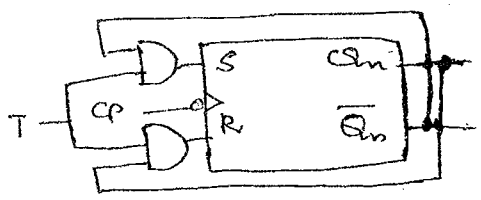
T \ Q_n	0	1
0	0 (0)	X (1)
1	1 (2)	0 (3)

$S = T\overline{Q_n}$

For R

T \ Q_n	0	1
0	X (0)	0 (1)
1	0 (2)	1 (3)

$R = T Q_n$



JK flip-flop to T flip-flop

Input T	Present state Q_n	Next state Q_{n+1}	Flip flop Inputs J	K
0	0	0	0	X
0	1	1	X	0
1	0	1	1	X
1	1	0	X	1

For J

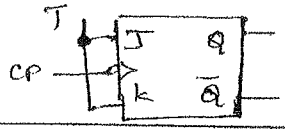
T \ Q_n	0	1
0	0	X
1	1	X

$J = T$

For K

T \ Q_n	0	1
0	X	0
1	X	1

$K = T$



JK flip-flop to D flip flop.

Input D	Present state Q_n	Next state Q_{n+1}	Flip flop Input J	K
0	0	0	0	X
0	1	0	X	1
1	0	1	1	X
1	1	1	X	0

for J

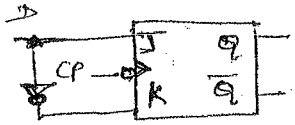
D \ Q_n	0	1
0	0	X
1	1	X

$J = D$

For K

D \ Q_n	0	1
0	X	1
1	X	0

$K = \bar{D}$



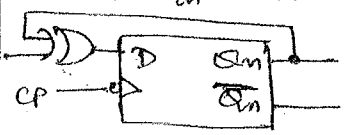
D flip flop to T flip flop.

Input T	Present State Q_n	Next state Q_{n+1}	Flip flop input D
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

For D

T \ Q_n	0	1
0	0	1
1	1	0

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



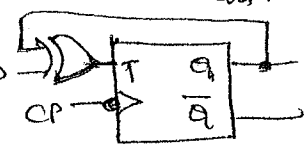
T flip flop to D flip flop

Input D	Present state Q_n	Next state Q_{n+1}	Flip flop input T
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

For T

D \ Q_n	0	1
0	0	1
1	1	0

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



Assignment

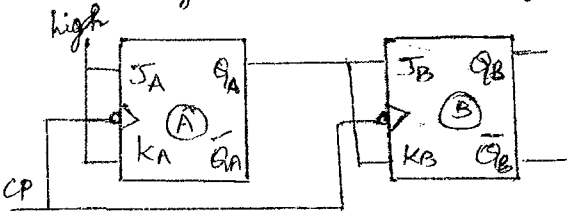
1. JK flip flop to SR flip flop.
2. D " to SR "

COUNTERS

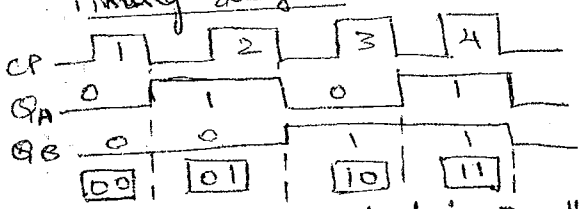
SYNCHRONOUS COUNTERS

When counter is clocked such that each flip-flop in the counter is triggered at the same time, the counter is called as synchronous counter.

2-bit synchronous binary up counter



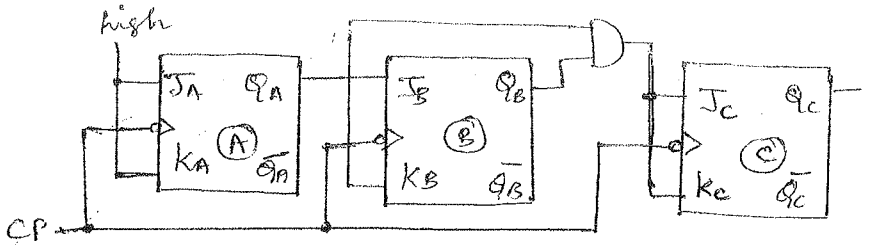
Timing diagram



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

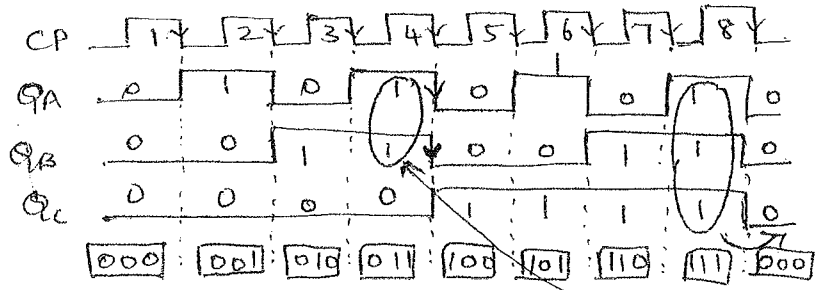
* clock signal is connected in parallel to clock inputs of both the flip-flops
* Q_A output of first stage is used to drive the J and K inputs of second stage.
* Q_A output of first stage is used to drive the J & K inputs of second stage.
* Let us assume $Q_A = Q_B = 0$.
* When positive edge of first clock pulse is applied, flip flop A will toggle because $J_A = K_A = 1$, whereas flip flop A will toggle because $J_A = K_A = 1$, whereas flip flop 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 clock pulse both flip flops will toggle because they both have a toggle condition on their J and K inputs ($J_A = K_A = J_B = K_B = 1$).
* After second clock pulse, $Q_A = 0$ and $Q_B = 1$. At negative going edge of the third clock pulse flip-flop A toggles making $Q_A = 1$, but flip-flop B remains set (i.e) $Q_B = 1$.
* At 4th clock both flip flop toggle as their JK inputs are at logic 1. This results $Q_A = Q_B = 0$ & counter regains back to its original state.

3-bit SYNCHRONOUS BINARY UP COUNTER.



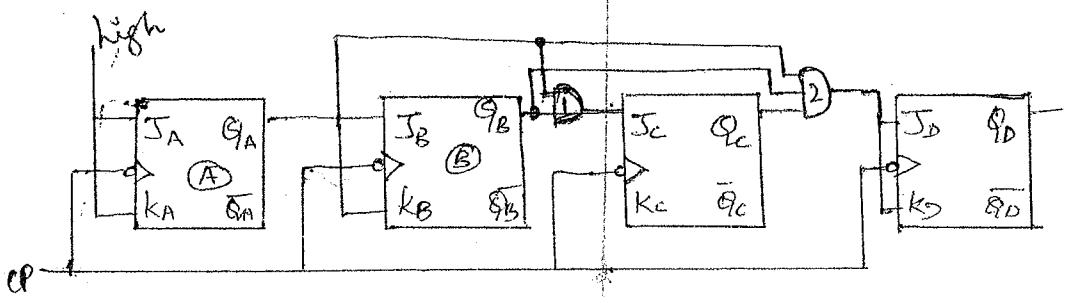
CP	Q_C	Q_B	Q_A
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

state sequence

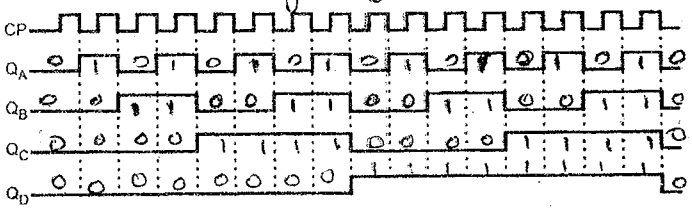


* Q_A changes on each clock pulse.
* flipflop A is held in toggle mode by connecting J and K inputs to HIGH.
* flipflop B toggles, when $Q_A = 1$ & when $Q_A = 0$ flipflop B remains in No change mode.
* flipflop C has to change its state only when Q_B and Q_A both are at logic 1.
* This condition is detected by AND gate and applied to J and K inputs of flipflop C, whenever Q_A & $Q_B = 1$, the AND gate makes J & K inputs of flipflop C high and it toggles clock pulse.
* J & K inputs of flipflop C are held low by AND gate output & flipflop doesnot change state.

4-bit SYNCHRONOUS BINARY UP COUNTER.



Timing diagram



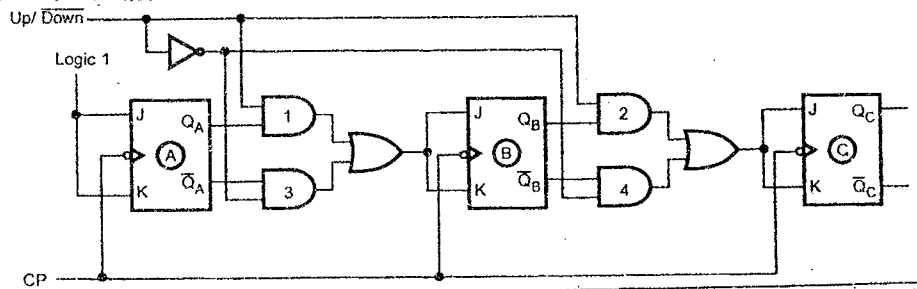
* the first three flip-flop work same as 3 bit counter
* for 4th stage, flip-flop has to change the state when $Q_A = Q_B = Q_C = 1$.
* This condition is decoded by 3-input AND gate G_{12}.
∴ D flip flop toggles and for all other times it is in no change condition.

SYNCHRONOUS UP/DOWN COUNTER.

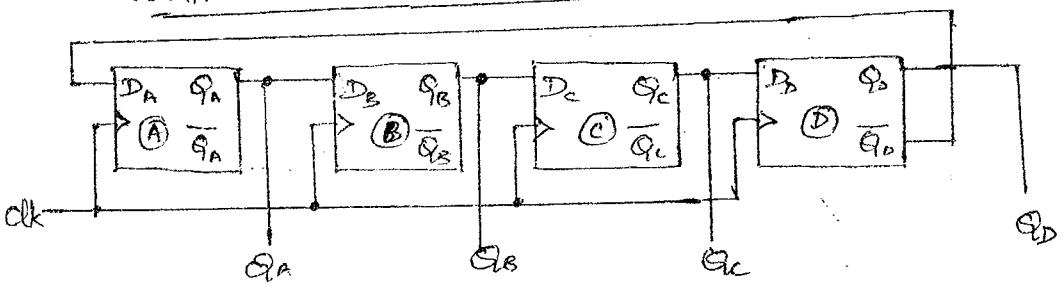
Up/Down	clk	Q_C	Q_B	Q_A
Up	0	0	0	0
	1	0	0	1
	2	0	1	0
	3	0	1	1
	4	1	0	0
	5	1	0	1
	6	1	1	0
	7	1	1	1

Up/Down	clk	Q_C	Q_B	Q_A
down	0	1	1	1
	1	1	1	0
	2	1	0	1
	3	1	0	0
	4	0	1	1
	5	0	1	0
	6	0	0	1
	7	0	0	0

* When logic 1 → gate 1 and 2 AND gates are enabled. and disables gate 3 and 4 → flip flop counts up.
* When logic 0 → gate 3 and 4 AND gates are enabled and disables gate 1 and 2 → flip-flop counts down.



JOHNSON COUNTER (or) TWISTED RING COUNTER/SHIFT COUNTER

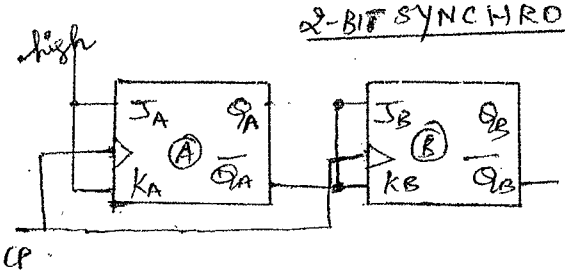


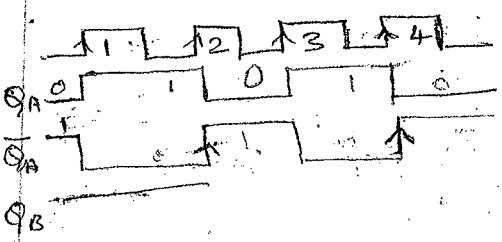
TRUTH TABLE

clk	Q_A	Q_B	Q_C	Q_D
0	0	0	0	0
↑	1	0	0	0
↑	1	1	0	0
↑	1	1	1	0
↑	1	1	1	1
↑	0	1	1	1
↑	0	0	1	1
↑	0	0	0	1
↑	0	0	0	0

* $\overline{Q_D}$ output is connected to D input of first flip flop A.
* Initially the counter is reset to '0' (ie) $Q_A Q_B Q_C Q_D = 0000$ & $D_B = D_C = D_D = 0$ Whereas $D_A = 1$ $\because \overline{Q_D} = D_A$.
* When 1st clock pulse is given, first flipflop is set to 1 others are reset to '0' (ie) $Q_A Q_B Q_C Q_D = 1000$
* Now $D_A = 1$. & $\overline{Q_D} = 1$ and $D_B = 1$ as $Q_A = 1$, but $D_C = D_D = 0$ as $Q_B = Q_C = 0$.
* When 2nd clock pulse is applied, 1st & 2nd flipflop are set to 1 while 3rd & 4th flip flop are '0'.
* After 3rd clock pulse $Q_A Q_B Q_C Q_D = 1110$, after 4th clock pulse $Q_A Q_B Q_C Q_D = 1111$.
* On the occurance of 5th clock pulse, the 1st flipflop is reset to '0' and others 3 flipflop are set to 1. (ie) $Q_A Q_B Q_C Q_D = 0111$.
* '0' is shifted from left to right until the Counter content $Q_A Q_B Q_C Q_D = 0000$ (ie) the initial state.

2-BIT SYNCHRONOUS DOWN COUNTER

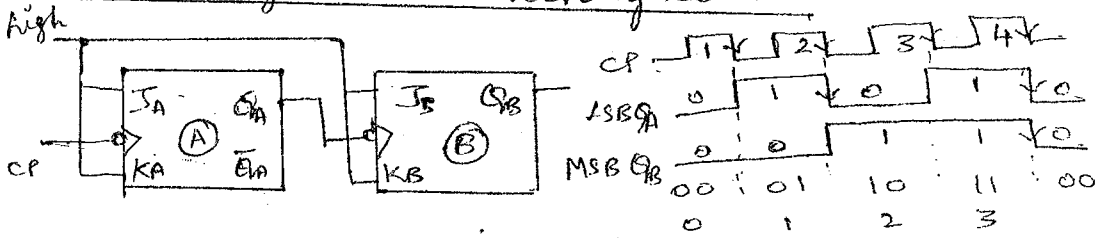




CP	Q_B	Q_A
0	1	1
1	1	0
2	0	1
3	0	0

ASYNCHRONOUS COUNTERS.

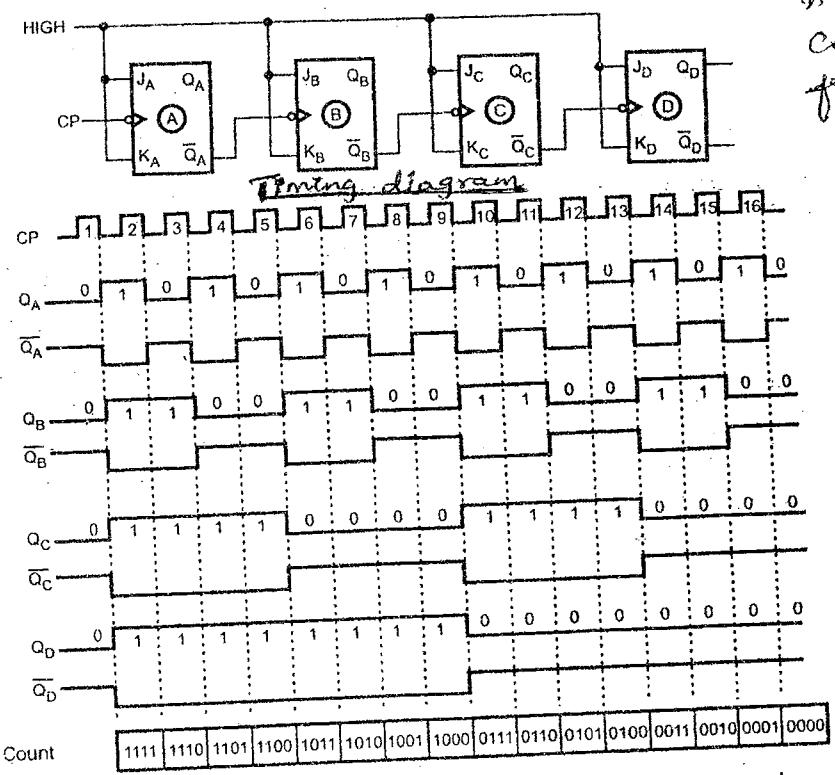
2-bit Asynchronous binary counter



LSB ⇒ receives incoming clock pulse.

* J & K tied together to get complement so T flip flop.
* The first Flipflop A is connected to negative edge triggered clock pulse.
* The flip flop B receives the clock pulse as the output of flip flop A (i.e) Q_A.

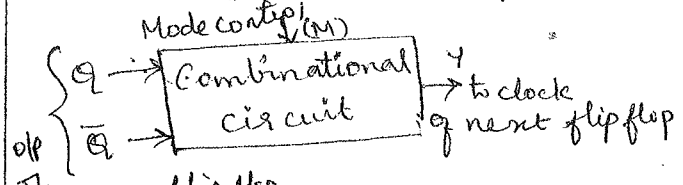
Asynchronous / ripple down counter



* down counter will count downwards from maximum count to zero.
* The clock pulse is connected only to the 1st flipflop A
* The rest of the flipflop is triggered by the Complemented output of the flipflop.
* JK together will toggles for every negative edge of clock input.
* Outputs are taken from Q_D Q_C Q_B Q_A. Once it reaches 0000, the Counter stops Counting.

ASYNCHRONOUS UP | DOWN COUNTER

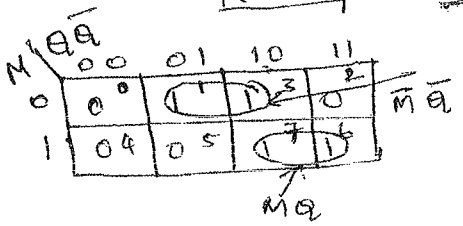
* Control input is necessary let us take it as M = 1 to count up and M = 0, to count down.



Truth table

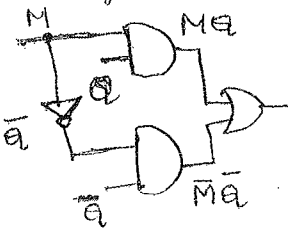
Inputs			output	
M	Q	$\bar{Q}$	Y	
0	0	0	0	$Y = \bar{Q}$ down counting
0	0	1	1	
0	1	0	0	
0	1	1	1	
1	0	0	0	Y = up counting
1	0	1	0	
1	1	0	1	
1	1	1	1	

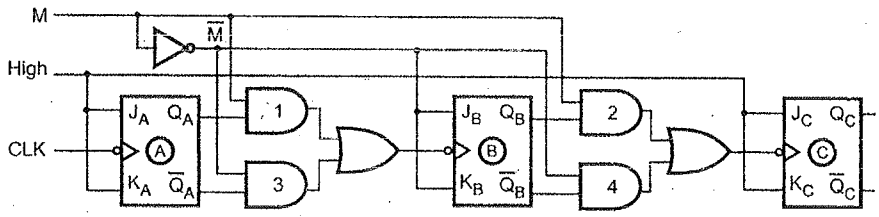
k-map

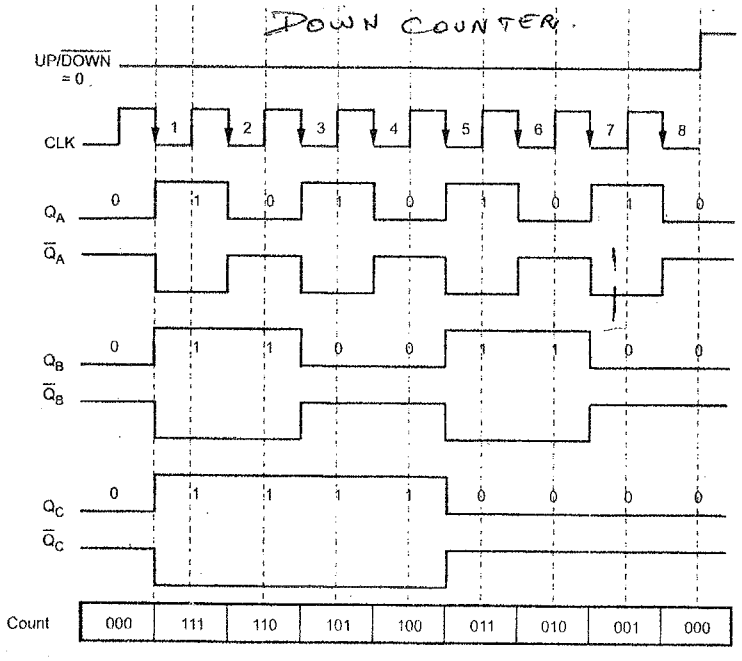


$Y = \bar{M}\bar{Q} + M\bar{Q}$.

logic diagram



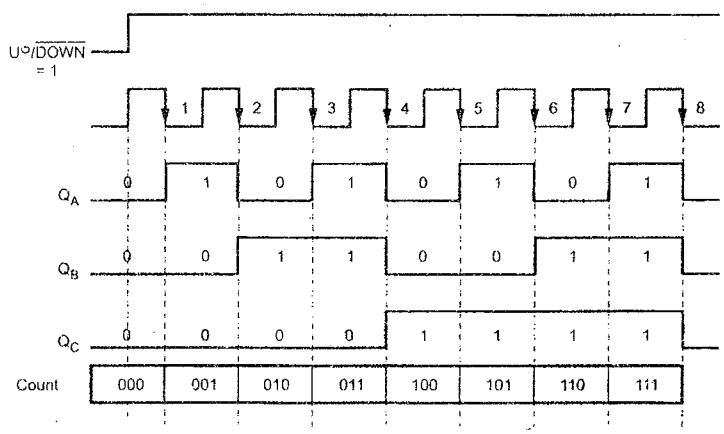




* If M = logic 0, then AND gate 3 and 4 are enabled and $\bar{Q}_A$, $\bar{Q}_B$ drive clock down and counts from 111 to 000. and AND gate 1 & 2 are disabled.

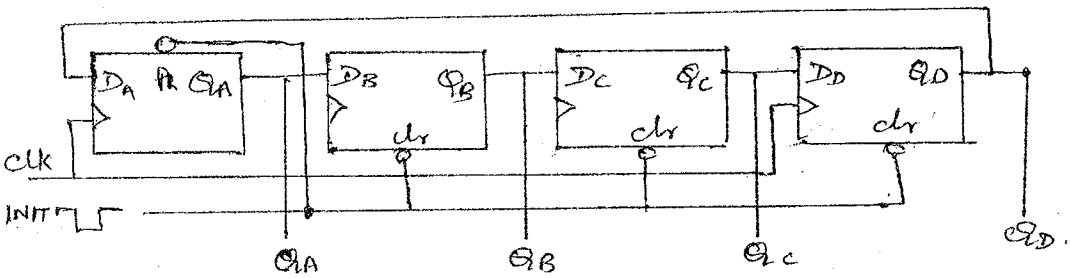
* If control input M = logic 1, then AND gate 1

UP COUNTER.



and 2 are enabled and AND gate 3 and 4 are disabled. Thus Q_A Q_B outputs drive the next stages and counts up from 000 to 111.

RING COUNTER.



INIT	clk	Q_D	Q_C	Q_B	Q_A
L	X	1	0	0	0
H	↑	0	1	0	0
H	↑	0	0	1	0
H	↑	0	0	0	1

* single 1 is made to circulate around register as long as clock pulses are applied.

* When INIT input is low, 1st flipflop is set to 1, all the others are cleared to 0. ∴ $Q_A Q_B Q_C Q_D = 1000$.

* When 1st clock pulse is applied, the 2nd flipflop is set to 1, while other three flipflops are reset to '0'. ∴ $Q_A Q_B Q_C Q_D = 0100$.

* After 2nd clock pulse 1 in the second flipflop is shifted to 3rd flipflop and output is $Q_A Q_B Q_C Q_D = 0010$.

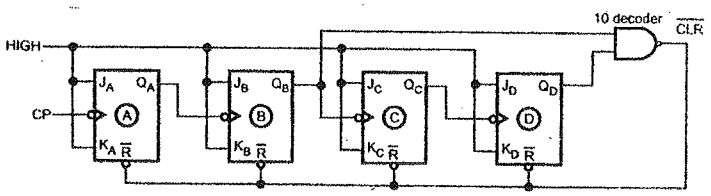
* At the occurence of 4th clock pulse, output is $Q_A Q_B Q_C Q_D = 0001$.

* On the 5th clock pulse $Q_A Q_B Q_C Q_D = 1000$ i.e. initial state.

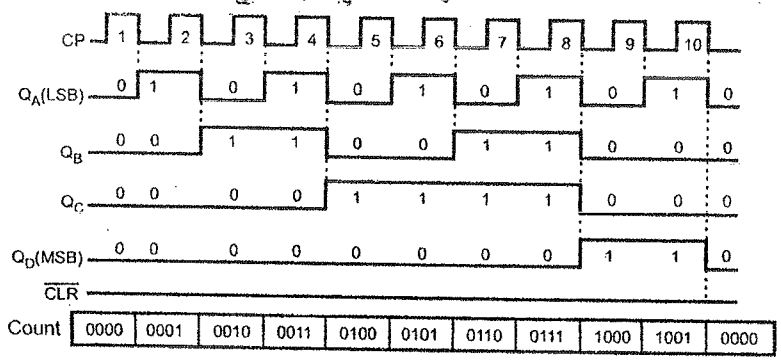
∴ '1' is shifted or circulated around the register as long as clock pulses are applied.

* The ring counter has only 4 valid states. (ie) 1000, 0100, 0010, 0001. It can hang or enter into any one of the invalid state, ∴ it should be avoided.

MOD-N | BCD RIPPLE COUNTER | DECADE COUNTER.



Timing diagram



* The binary counter has maximum number of states equal to 2^n, where n is the number of flipflops in counter
* Counters can also be designed to have a number of states in their sequence that is less than 2^n
* This decade counter is upto 0000 to 1001 (0 to 9). ∴ also used as BCD ripple counter.
* The resetting of counter after 1001 is done with the help of reset inputs of each flipflop($\overline{R}$)
* In case of BCD decade counter, reset input is activated using NAND gate when 1010 state is reached

COUNTER DESIGN

1. Obtain transition table
2. Determine number of flipflop needed is $2^n \geq N$
   Where n → number of flip flops, N = number of stages.
3. Choose the type of flip-flops to be used.
4. From the transition table, desire the circuit excitation table
5. Use K-map or anyother simplification method to desire the circuit flip-flop input function
6. Draw the logic diagram.

---

1. Design a synchronous mod-6 counter using clocked JK flipflops.

1. $2^n \geq N$ $\quad 2^0 \neq 6$, $2^1 \neq 6$, $2^2 \neq 6$, $2^3 \geq 6$ ✓

$\therefore n = 3$, $N = 6$., Number of flipflops = 3.

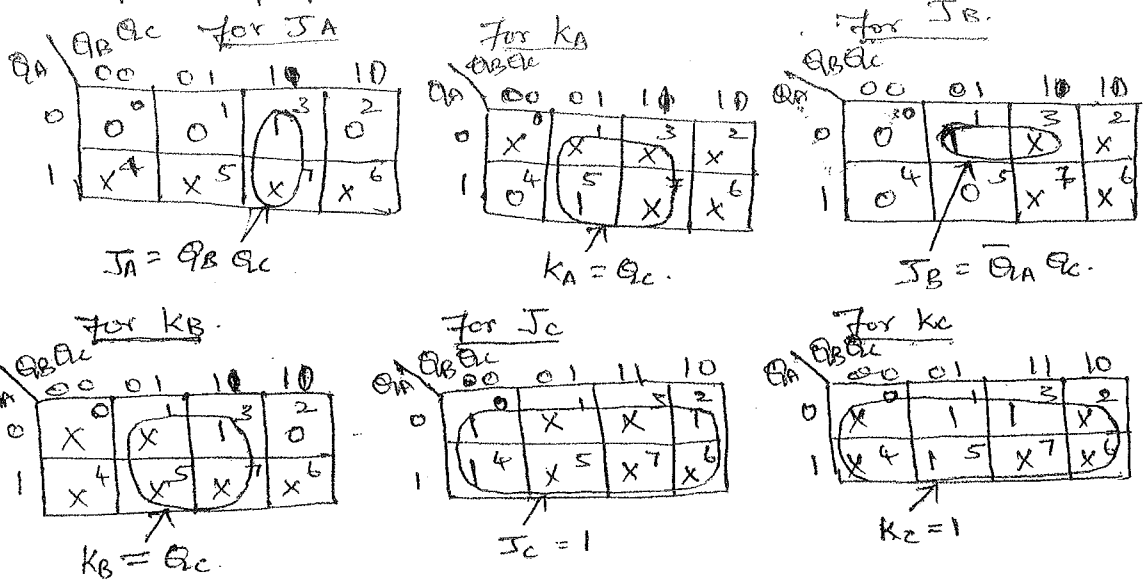
2. Excitation table for JK flip flop.

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

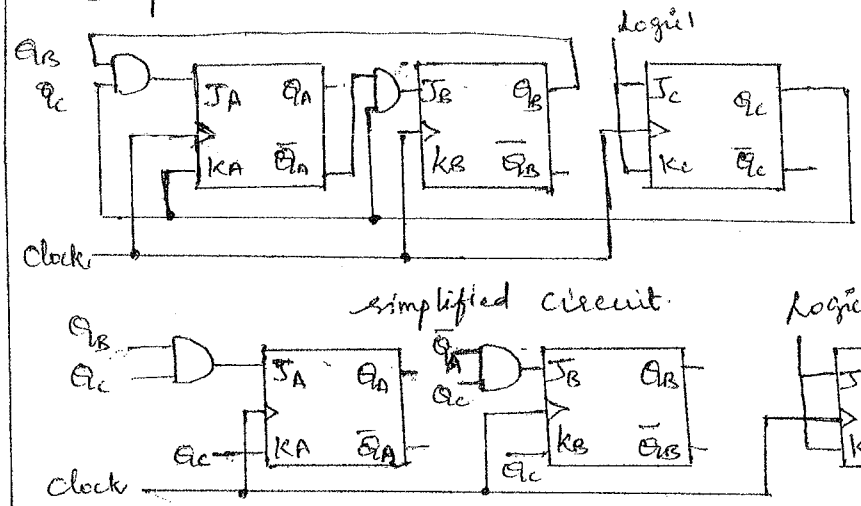
3. Transition table

Present state			Next state			Flip flop inputs.					
Q_A	Q_B	Q_C	Q_{A+1}	Q_{B+1}	Q_{C+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	1	0	X	X	1
1	1	0	X	X	X	X	X	X	X	X	X
1	1	1	X	X	X	X	X	X	X	X	X

4. K-map simplification for flip flop inputs



5. Implement the counter.



simplified circuit.

---

2. Design of UP/DOWN syncheonous counter/bidirectional counter.

Let us design a 3 bit counter.

CP	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	

2. Excitation table

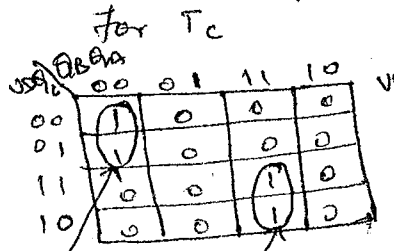
Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

3. Transition table

Input UP / DOWN (UD)	Present state Q_C	Q_B	Q_A	Next state Q_{C+1}	Q_{B+1}	Q_{A+1}	Flip flop inputs T_C	T_B	T_A
down 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	1	1
0	1	0	0	0	1	1	1	1	1
0	1	0	1	1	0	0	1	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	0	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

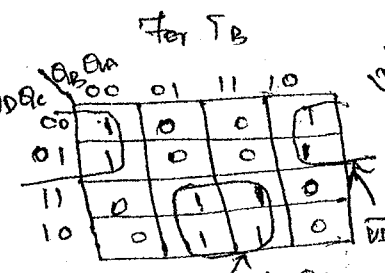
4. K-map simplification

for T_C



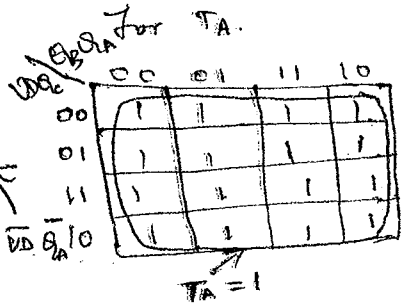
$T_C = \bar{U}_D \bar{Q}_B \bar{Q}_A + U_D Q_B Q_A$

for T_B

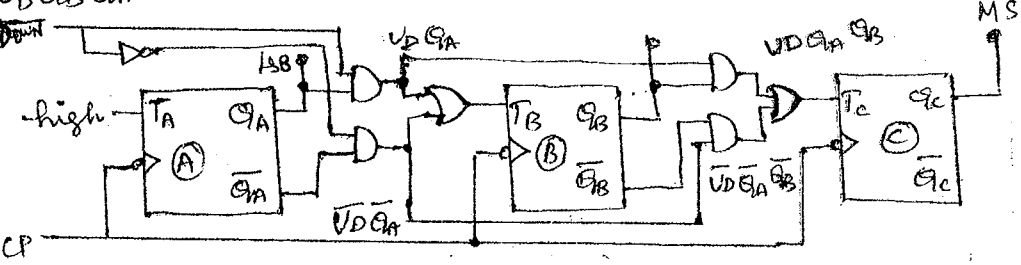


$T_B = U_D Q_A + \bar{U}_D \bar{Q}_A$

for T_A



5.



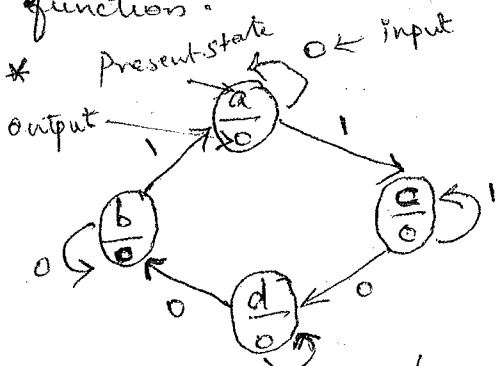
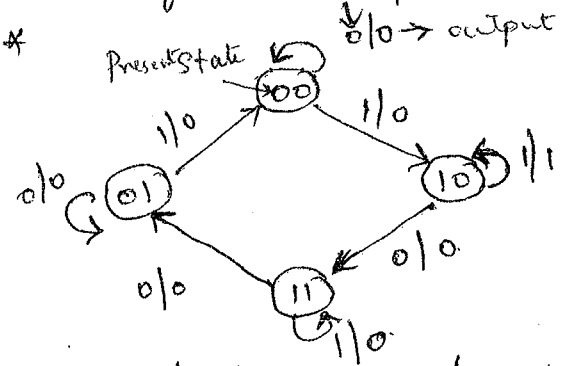
ANALYSIS OF CLOCKED SEQUENTIAL CIRCUIT

State equation: A state equation also called transition equation is one which specifies the next state as a function of present state and inputs.

state table: A state table is a table inwhich the time sequence of inputs, outputs and flipflop state can be enumerated, also called transition table. It consist of four sections - present state, input, next state and output.

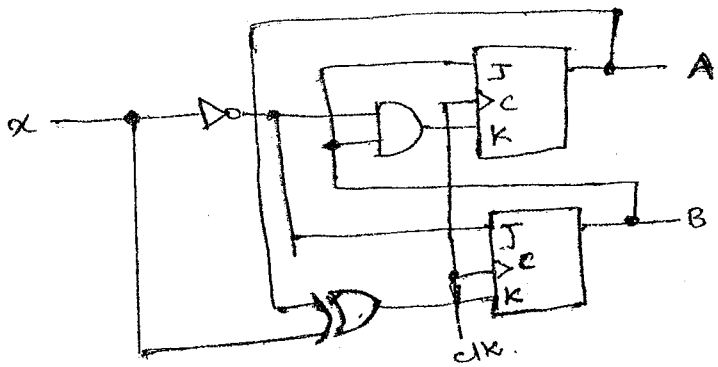
state diagram: It is the graphical representation of information available in a state table. The state is represented in circle and the transition between states are indicated by directed lines connecting the circles.

Moore Vs Mealy circuit model.

Moore circuit	Mealy circuit.
* Its output is a function of present state only. * Input changes doesnot affect the output. * Moore circuit requires more number of states for implementing same function.	* Its output is a function of present state as well as present input. * Input changes may offer the output of the circuit * It requires less number of states for implementing same function.
* Present state, 0 ← input, output  * referred as Mealy FSM/Machine	* input, 0/0 → output  * referred as Moore FSM / Machine.

Moore Model: Procedure for analysis using JK flip flop

* determine the flip flop input equations interms of present state and input variables.
* List the binary values of each input equation.
* Substitute the input equations into the flip flops characteristic equation to obtain the state equation.
* Use the corresponding state equation to determine the next state values in the state table.



* circuit has no outputs, ∴ no output column needed in the state table.

flip-flop input-output equations

$J_A = B$ $\quad K_A = B\bar{x}$

$J_B = \bar{x}$ $\quad K_B = A \oplus x$ ——— (1)

⇒ characteristic equation of JK is

$Q_{n+1} = J\bar{Q} + \bar{K}Q$ ——— (2)

Sub. A & B instead of Q. in (2)

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

$B(t+1) = J\bar{B} + \bar{K}B$ ——— (3).

Sub. eqn. (1) in (3).

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

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

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

$= \bar{B}\bar{x} + [\overline{A\bar{x} + \bar{A}x}]B.$

$= \bar{B}\bar{x} + [\overline{A\bar{x}} \cdot \overline{\bar{A}x}]B. = \bar{B}\bar{x} + [(\bar{A}+x)(A+\bar{x})]B$

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

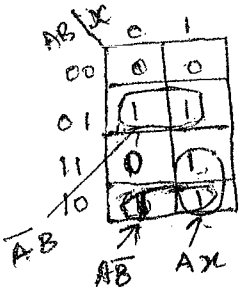
$B(t+1) = \bar{B}\bar{x} + \bar{A}\bar{x}B + AB\bar{x}$

state table

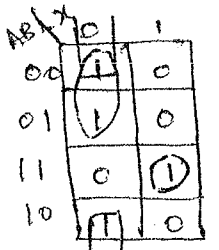
Present state		input	next state		flip flop inputs			
A	B	x	A(t+1)	B(t+1)	J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	1
1	1	0	0	0	1	1	1	0
1	1	1	1	1	1	0	0	0

K map for next state

for A(t+1)



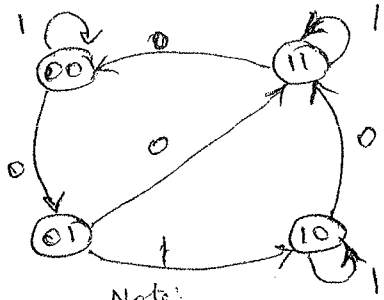
for B(t+1)



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

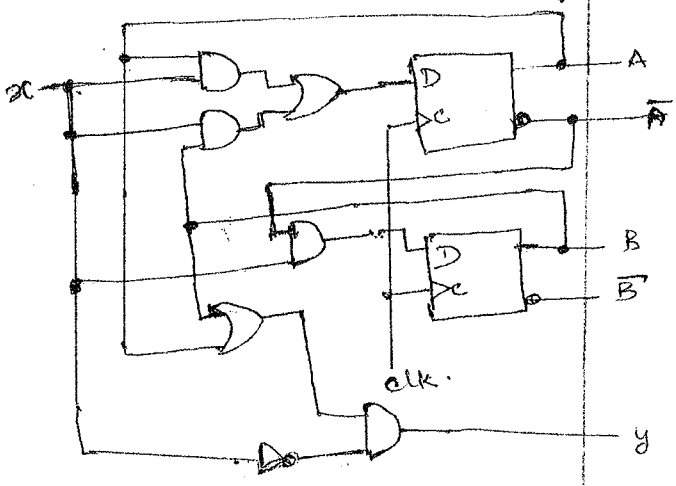
$A(t+1) = \bar{A}B + A\bar{B} + Ax$

state diagram



Note: since circuit has no output

MEALY MODEL : Analysis using D-Flip flop.



1. Flip flop i/p - o/p equation

$D_A = Ax + B\bar{x}$

$D_B = \overline{A+x}$

2. Characteristic equation of D flip flop.

$Q_{n+1} = D$

Sub. A & B instead of Q

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

$= D_A$

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

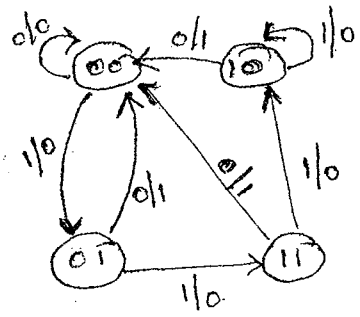
$= D_B$

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

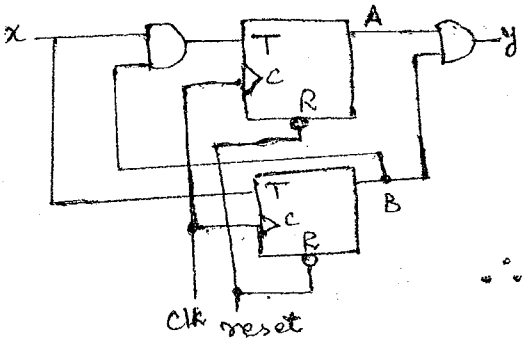
state table

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

state diagram



ANALYSIS WITH T-FLIPFLOP



characteristic equation

$Q(t+1) = T \oplus Q = \bar{T}Q + T\bar{Q}$

$T_A = Bx$

$T_B = x$

$y = AB$

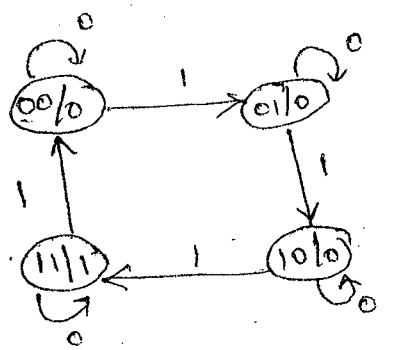
$\therefore A(t+1) = (\bar{B}\bar{x})A + Bx\bar{A}$

$= A\bar{B} + A\bar{x} + \bar{A}Bx$

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

state table

Present state	Input	Next state	output
AB	x	AB	y
0 0	0	0 0	0
0 0	1	0 1	0
0 1	0	0 1	0
0 1	1	1 0	0
1 0	0	1 0	0
1 0	1	1 1	0
1 1	0	1 1	1
1 1	1	0 0	1



state diagram

(Moore's model)

STATE REDUCTION:

* To reduce the number of gates and flip flops during the design.

* State reduction algorithm:

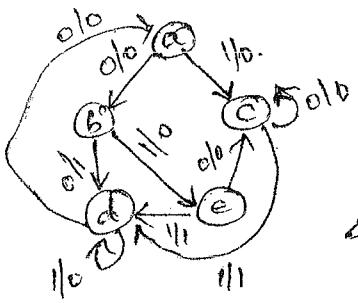
* They are concerned with the procedures for reducing the number of states in a input-output requirements unchanged.

* 'm' flipflop produce 2^m states, a reduction in the number of states result in a reduction in the number of flipflops is also reduced.

* This technique avoids redundant states.

* Two states are said to be redundant or equivalent if every possible set of outputs generate exactly same output and same next state

* When 2 states are equivalent, one of them can be removed without altering the input-output relationship.

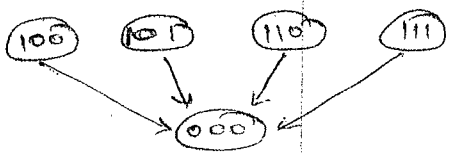


* Letter symbols are used instead of binary.

* because internal states are not important but only input-output sequences are important

STATE ASSIGNMENT

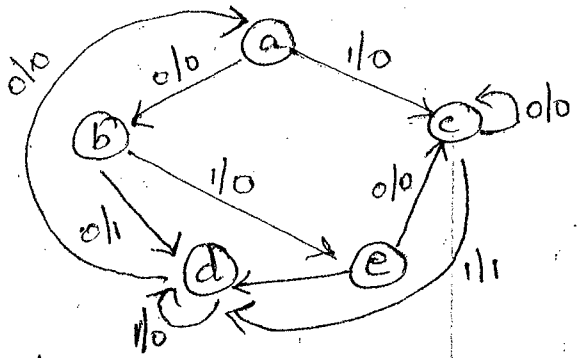
Rule 1: State having same next states for a given input condition should have assignment which can be grouped into logically adjacent cells in K-map.



Rule 2:

State that are the next state of a single state should have assignment which can be grouped into logically adjacent cells in K-map.

1. Determine the state table for the given state diagram.



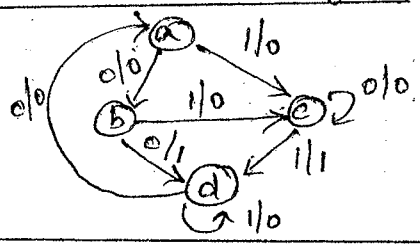
State table

Present state	Next state		Output	
	X=0	X=1	X=0	X=1
a	b	c	0	0
b	d	e	1	0
c	c	d	0	1
d	a	d	0	0
e	c	d	0	1

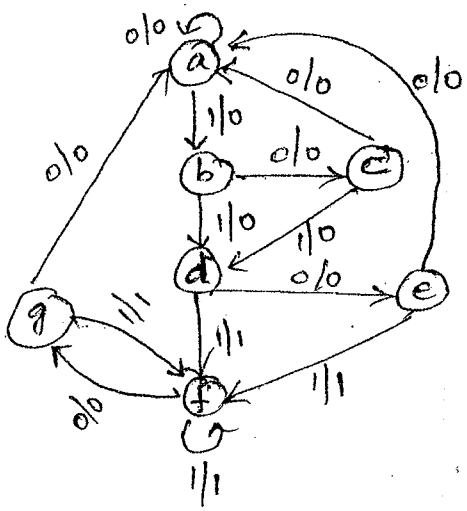
Reduced state table

Present state	Next state		output	
	X=0	X=1	X=0	X=1
a	b	c	0	0
b	d	c	1	0
c	c	d	0	1
d	a	d	0	0

Reduced state diagram.



2.



State table

Present state	Next state		output	
	x=0	x=1	x=0	x=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

* When two states are equivalent, one of them can be removed without altering the input-output relationship

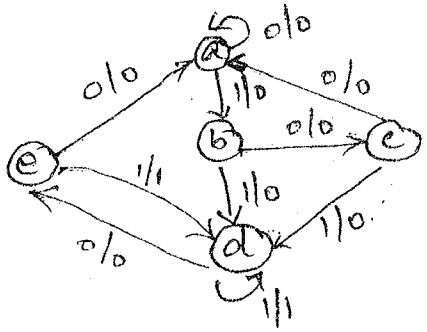
* 'g' and 'e' are two such states ∴ One can be removed.

Reducing State table

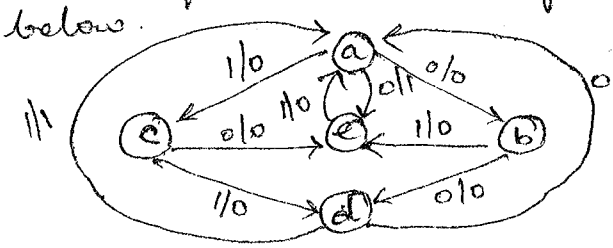
Present state	Next state		output	
	x=0	x=1	x=0	x=1
a	e	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f d	0	1
e	a	f d	0	1
~~f~~	~~f e~~	f	0	1
~~g~~	a	f		

Reduced state stable

Present State	Next state		output	
	x=0	x=1	x=0	x=
a	a	b	0	0
b	e	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1



3. Design a sequential circuit for the state diagram shown below.



Random assignment

a = 000
b = 001
c = 010
d = 011
e = 100

state table

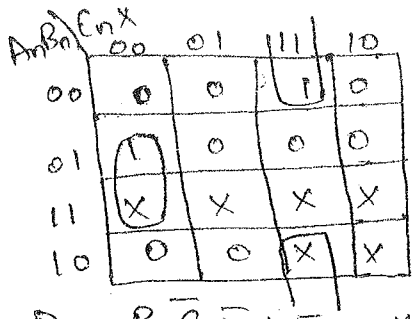
Present state	Next state	
	x = 0	x = 1
a	b	c
b	d	e
c	e	d
d	a	a
e	a	a

Excitation table with above assignment.

Present state			Input	Next state			output
A_n	B_n	C_n	x	A_{n+1}	B_{n+1}	C_{n+1}	z
0	0	0	0	0	0	1	0
0	0	0	1	0	1	0	0
0	0	1	0	0	1	1	0
0	0	1	1	1	0	0	0
0	1	0	0	1	0	1	0
0	1	0	1	0	1	0	0
0	1	1	0	0	0	0	1
0	1	1	1	0	0	0	1
1	0	0	0	0	0	0	0
1	0	0	1	x	x	x	x
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

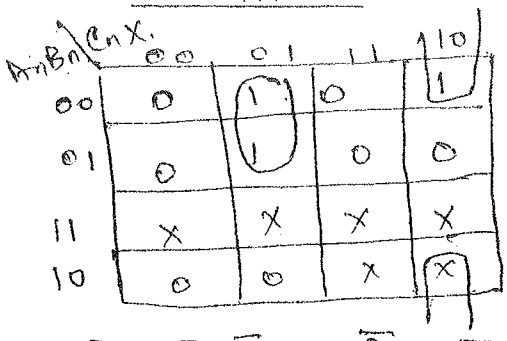
K-map

For $A_{n+1} = D_A$



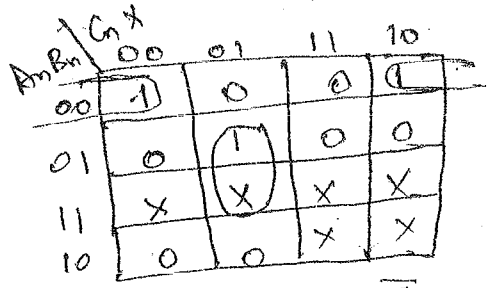
$D_A = B_n\bar{C}_n\bar{X} + \bar{B}_nC_nX$

For $B_{n+1} = D_B$



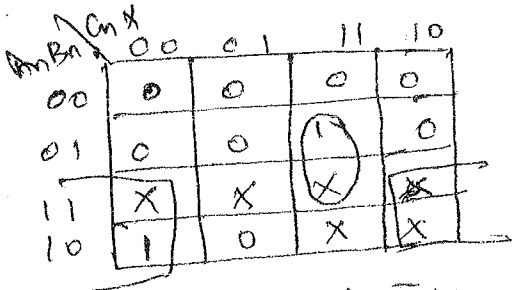
$D_B = \bar{A}_n\bar{C}_nX + \bar{B}_nC_n\bar{X}$

For $C_{n+1} = D_C$



$D_C = \bar{A}_n\bar{B}_n\bar{X} + B_n\bar{C}_nX$

For Z



$Z = B_nC_nX + A_n\bar{X}$

The random assignment require

- 1 3 input AND functions
- 1 2 " " "
- 4 " " OR "

12 gates with 31 inputs.

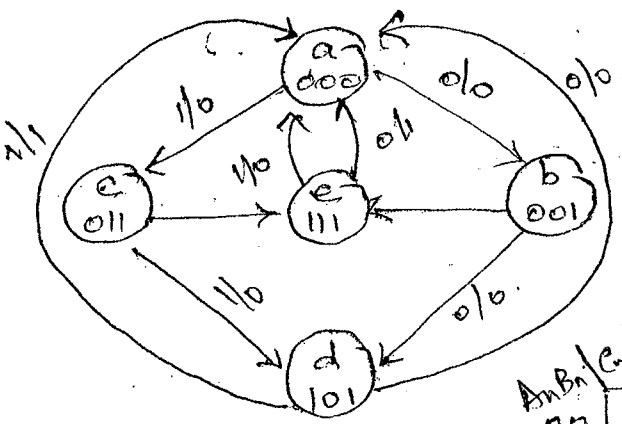
Applying state assignment rules

Rule 1 says that: e & d must be adjacent &
b & c " " "

Rule 2 says that: b & c " " "
e & d " " "

k-map showing b·c d·e adjacent to each other

Present state			Input	Next state			output
A_n	B_n	C_n	x	A_{n+1}	B_{n+1}	C_{n+1}	z
0	0	0	0	0	0	1	0
0	0	0	1	0	1	1	0
0	0	1	0	1	0	1	0
0	0	1	1	1	1	1	0
0	1	0	0	x	x	x	x
0	1	0	1	x	x	x	x
0	1	1	0	x	x	1	0
0	1	1	1	1	0	1	0
0	1	0	0	x	x	x	x
1	0	0	1	x	x	x	x
1	0	0	0	x	x	0	0
1	0	1	1	0	0	0	1
1	0	1	0	0	0	x	x
1	1	0	1	x	x	x	x
1	1	0	0	x	x	0	1
1	1	1	0	0	0	0	0
1	1	1	1	0	0		



A \ BC	00	01	11	10
0	a 0	b 1	c 3	2
1	4	d 5	e 7	6

$A_{n+1} = D_A = \bar{A}_n C_n$.

k-map For A_{n+1}

A_nB_n \ C_n	00	01	11	10
00	0	0	1	1
01	x	x	1	1
11	x	x	0	0
10	x	x	0	0

For D_B

A_nB_n \ C_nX	00	01	11	10
00	0	1	1	0
01	X	X	0	1
11	X	X	0	0
10	X	X	0	0

$B_{n+1} = D_B = \bar{A}_n \bar{B}_n X + \bar{A}_n B_n \bar{X}$

For D_C.

A_nB_n \ C_nX	00	01	11	10
00	1	1	1	1
01	X	X	1	1
11	X	X	0	0
10	X	X	0	0

$C_{n+1} = D_C = \bar{A}_n$

For Z:

A_nB_n \ C_nX	00	01	11	10
00	0	0	0	0
01	X	X	0	0
11	X	X	0	1
10	X	X	1	0

$Z = A_n B_n \bar{X} + A_n \bar{B}_n X$.

It requires

4 3 input AND function

1 2 " " "

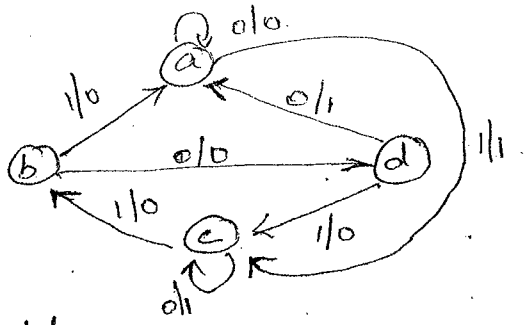
2 " " OR "

7 gate with 18 inputs.

DESIGN PROCEDURE FOR CLOCKED SYNCHRONOUS SEQUENTIAL LOGIC.

* Obtain the state table
* Reduce the state table.
* Assign binary values to state assignment.
* Determine number of flipflop needed.
* choose the type of flipflop.
* From the state table, derive the circuit excitation table & function tables.
* using k-map derive the circuit output functions and the flip flop input functions.
* draw the logic diagram.

1. A sequential circuit has one input and one output. The state diagram is shown below. Design the sequential circuit with (a) T flip flop. (b) RS flip flop (c) JK flip flop.



Mealy's Model

Let a=00, b=01
c=10, d=11.

1) state table

Present state	Next state X=0	Next state X=1	Output X=0	Output X=1
a	a	c	0	1
b	d	a	0	0
c	c	b	1	0
d	a	c	1	0

Present state AB	Next state X=0 AB	Next state X=1 AB	Output X=0	Output X=1
0 0	0 0	1 0	0	0
0 1	1 1	0 0	0	0
1 0	1 0	0 1	1	0
1 1	0 0	1 0	1	0

* No equivalent states ∴ No reduction.
* From the state table it is understood that there are 4 states. ∴ 2 flipflops are required.
(Number of states = 2^m, where m = number of flipflops)
* states are assigned as a=00, b=01, c=10, d=11.

using T flipflop.

* use the excitation table for T flipflop and determine the excitation table for the given circuit.

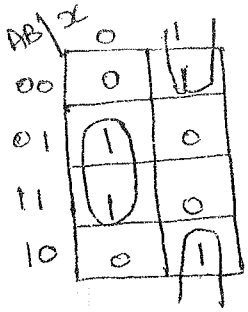
Present state A B	Input x	Next state A B	flipflop inputs T_A	flipflop inputs T_B	outputs y
0 0	0	0 0	0	0	0
0 0	1	1 0	1	0	1
0 1	0	1 1	1	0	0
0 1	1	0 0	0	1	0
1 0	0	1 0	0	0	1
1 0	1	0 1	1	1	0
1 1	0	0 0	1	1	1
1 1	1	1 0	0	1	0

T flip flop excitations:

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

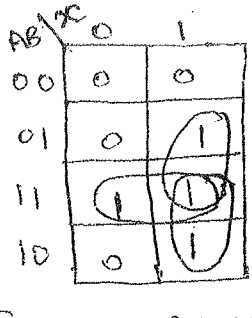
K-map simplification

For T_A



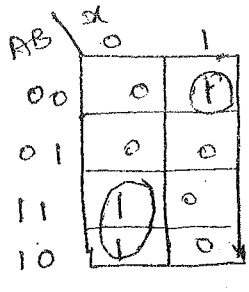
$T_A = B\bar{X} + \bar{B}X$

For T_B

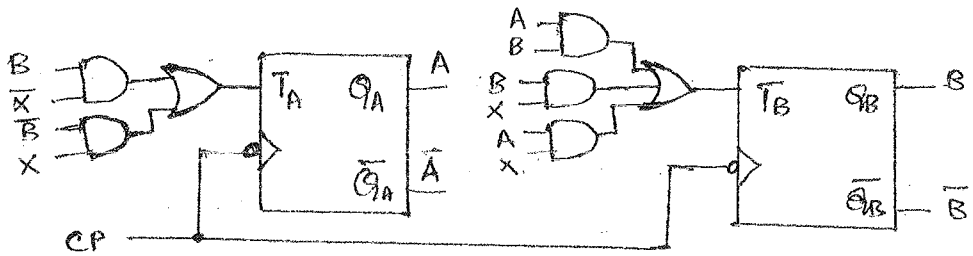


$T_B = AB + AX + BX$

For output Y



$Y = A\bar{X} + \bar{A}\bar{B}X$

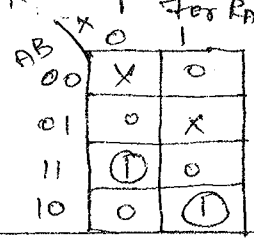


Output: $A\bar{X} + \bar{A}\bar{B}X$ → output Y.

Using RS flipflop

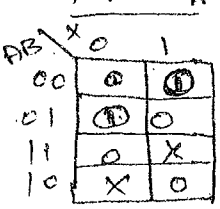
Present state	Input	Next state	flipflop inputs				output
AB	X	AB	R_A	S_A	R_B	S_B	Y
00	0	00	X	0	X	0	0
00	1	10	0	X	X	0	1
01	0	11	0	1	0	X	0
01	1	00	X	0	1	0	0
10	0	10	0	X	X	0	1
10	1	01	1	0	0	0	0
11	0	00	1	0	1	0	1
11	1	10	0	X	1	0	0

K-map for R_B



$R_B = AB\bar{X} + A\bar{B}X$

K-map for S_A



$S_A = \bar{A}BX + \bar{A}B\bar{X}$

For R_B.

AB \ X	0	1
00	X	X
01	0	1
11	1	1
10	X	0

$R_B = AB + B\bar{X}$

For S_B.

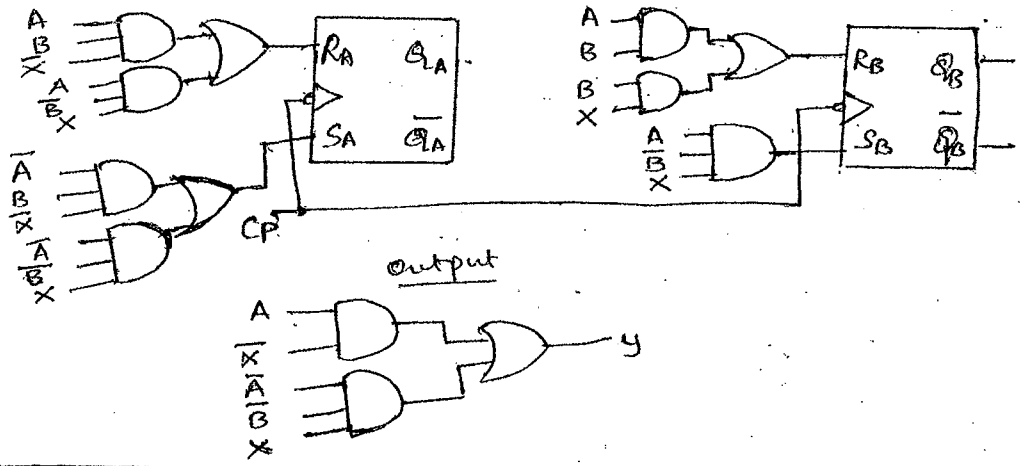
AB \ X	0	1
00	0	0
01	X	0
11	0	0
10	0	1

$S_B = A\bar{B}X$

Output Y.

AB \ X	0	1
00	0	1
01	0	0
11	1	0
10	1	0

$Y = A\bar{X} + \bar{A}\bar{B}X$



using JK flipflop.

Present state AB	Input X	Next state A B	flipflop inputs J_A	k_A	J_B	k_B	output Y
0 0	0	0 0	0	X	0	X	0
0 0	1	1 0	1	X	0	X	1
0 1	0	1 1	1	X	X	0	0
0 1	1	0 0	0	X	X	1	0
1 0	0	1 0	X	0	0	X	1
1 0	1	0 1	X	1	1	X	0
1 1	0	0 0	X	1	X	1	1
1 1	1	1 0	X	0	X	1	0

K-map.

For J_A.

AB \ X	0	1
00	0	1
01	1	0
11	X	X
10	X	X

$J_A = B\bar{X} + \bar{B}X$

For k_A.

AB \ X	0	1
00	X	X
01	X	X
11	1	0
10	0	1

$k_A = B\bar{X} + \bar{B}X$

For J_B.

AB \ X	0	1
00	0	0
01	X	X
11	X	X
10	0	1

$J_B = AX$.

For K_B.

AB \ X	0	1
00	X	X
01	0	1
11	1	1
10	X	X

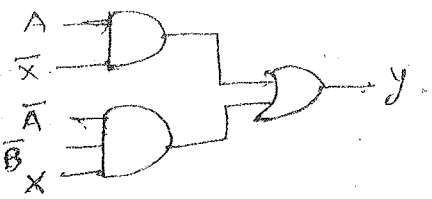
$K_B = A + X$

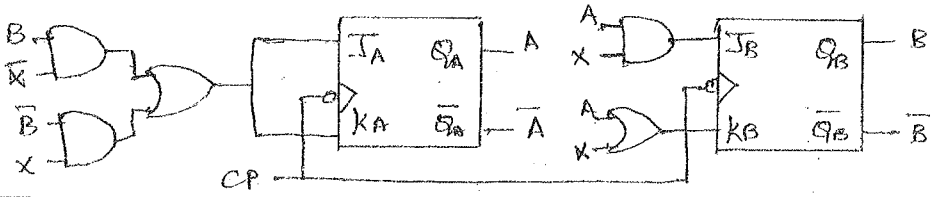
For Y

AB \ X	0	1
00	0	1
01	0	0
11	1	0
10	1	0

$Y = A\bar{X} + \bar{A}\bar{B}X$

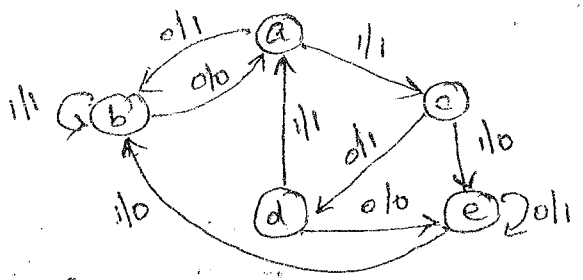
Output Y.



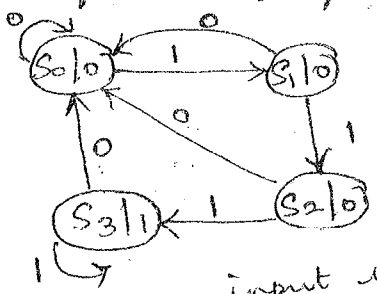


Assignment:

1. Design a sequential circuit for the state diagram shown below use JK flipflop



2. Design sequential sequence detector using D, T, JK flipflop

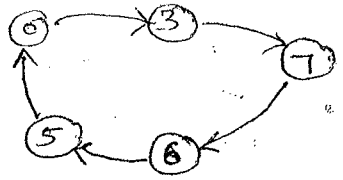


sequence detector (ie) it detects three or more consecutive 1's in a string. It starts with state S_0. If the input is '0', the circuit stays in the same state, if the input is '1', it goes to S_1 to indicate a '1' was detected as so on.

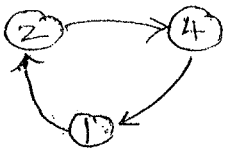
LOCK OUT CONDITION.

In a counter of reset state of some unused state is again an unused state and if it never arrives at a used state, then the counter is said to be in lockout condition.

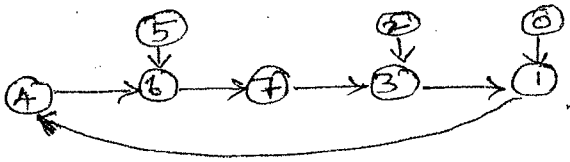
desired sequence



unused state forming lockout



1. Design a synchronous counter for $4 \to 6 \to 7 \to 3 \to 1 \to 4$ .... Avoid lockout condition. use JK type design.



∴ states 5, 2 & 0 are forced to go into 6, 3 & 1 state respectively to avoid lockout condition.

Excitation table

Present state			Next state			Flip-flop inputs					
Q_A	Q_B	Q_C	Q_{A+1}	Q_{B+1}	Q_{C+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	1	0	0	1	X	0	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	0	0	1	0	X	X	1	X	0
1	0	0	1	1	0	X	0	1	X	0	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	1	1	X	1	X	0	X	0

K-map.

For J_A.

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

$J_A = \overline{Q}_B Q_C$

For K_A

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

$K_A = Q_B Q_C$

For J_B

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

$J_B = Q_A$

For K_B

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

$K_B = \overline{Q}_A Q_C$

For J_C

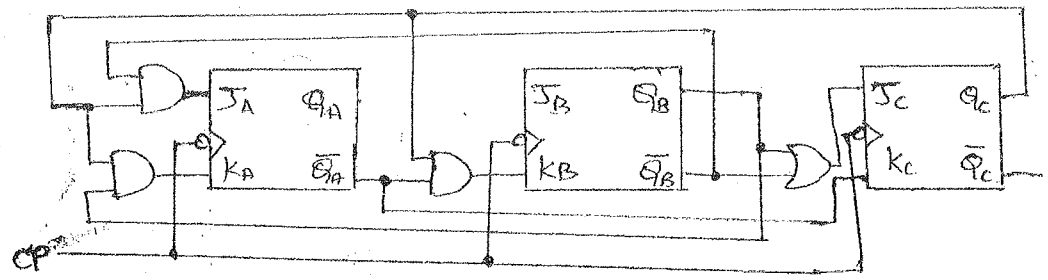
Q_A \ $Q_B Q_C$	00	01	11	10
0	1	X	X	1
1	0	X	X	1

$J_C = \overline{Q}_A + Q_B$

For K_C

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

$K_C = \overline{Q}_B$



SEQUENCE GENERATOR USING COUNTERS.

Number of flip flops required

1. Find number of 1's in sequence.
2. " " " 0's " "
3. Take maximum out of 2.
4. If N is the required number of flip-flop choose minimum value of 'n' to satisfy equation $\max(0s, 1s) \leq 2^n - 1$.

1. Find the number of flip flops required to generate sequence 1101011

$N = 7$.

$N = \max(2, 5) \leq 2^n - 1$.

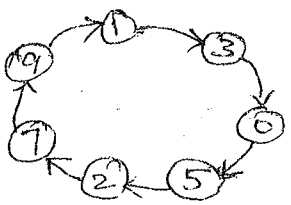
$5 \leq 2^{1-1} \not\leq 1$, $5 \leq 2^{2-1} \not\leq 2$

$5 \leq 2^{3-1} \not\leq 4$, $5 \leq 2^{4-1} \leq 8$ $\boxed{n = 4}$.

0 0 0 0 ~ m_0
0 0 0 1 ~ m_1
0 0 1 0 ~ m_2
0 0 1 1 ~ m_3
0 1 0 0 m_4
0 1 0 1 ~ m_5
0 1 1 0 m_6
0 1 1 1 ~ m_7
1 0 0 0 m_8
1 0 0 1 ~ m_9

A B C D	states
0 0 0 1	1
0 0 1 1	3
0 0 0 0	0
0 1 0 1	5
0 0 1 0	2
0 1 1 1	7
1 0 0 1	9

state diagram



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

Present state				Next state				Flip flop inputs			
Q_A	Q_B	Q_C	Q_D	Q_{A+1}	Q_{B+1}	Q_{C+1}	Q_{D+1}	J_A K_A	J_B K_B	J_C K_C	J_D K_D
0	0	0	0	0	1	0	1	0 x	1 x	0 x	1 x
0	0	0	1	0	0	1	1	0 x	0 x	1 x	x 0
0	0	1	0	0	1	1	1	0 x	1 x	x 0	1 x
0	0	1	1	0	0	0	0	0 x	0 x	x 1	x 1
0	1	0	0	x	x	x	x	x x	x x	x x	x x
0	1	0	1	0	0	1	0	0 0	x 1	1 x	x 1
0	1	1	0	x	x	x	x	x x	x x	x x	x x
0	1	1	1	1	0	0	1	1 1	x 1	x 1	x 0
1	0	0	0	x	x	x	x	x x	x x	x x	x x
1	0	0	1	0	0	0	1	x 1	0 x	0 x	x 0
1	0	1	0	x	x	x	x	x x	x x	x x	x x
1	0	1	1	x	x	x	x	x x	x x	x x	x x
1	1	0	0	x	x	x	x	x x	x x	x x	x x
1	1	0	1	x	x	x	x	x x	x x	x x	x x
1	1	1	0	x	x	x	x	x x	x x	x x	x x
1	1	1	1	x	x	x	x	x x	x x	x x	x x

For J_A.

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

$J_A = Q_B Q_C$

For K_A.

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

$K_A = 1$

For J_B.

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

$J_B = \overline{Q_D}$

For K_B.

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

$K_B = 1$

For J_C

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

$J_C = \overline{Q_A} Q_D$

For K_C

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

$K_C = Q_D$

For J_D.

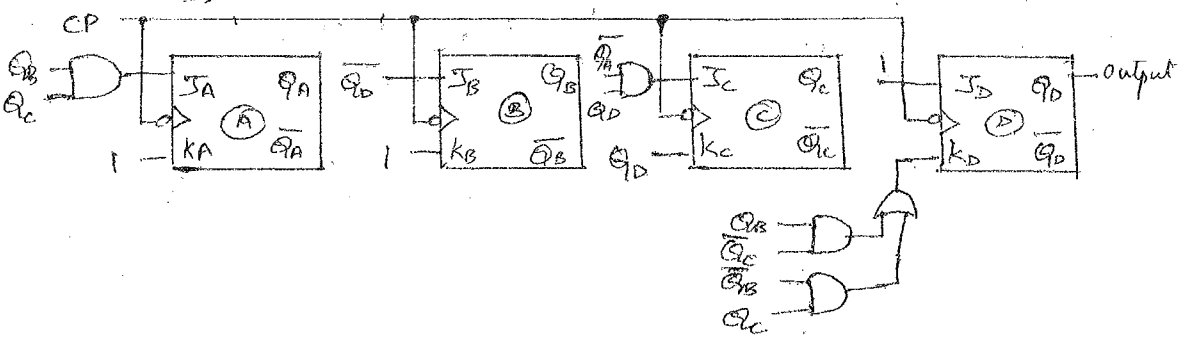
$Q_A Q_B$ \ $Q_C Q_D$	00	01	11	10
00	1	X	X	1
01	X	X	X	X
11	X	X	X	X
10	X	X	X	X

$J_D = 1$

For K_D

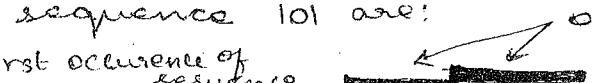
$Q_A Q_B$ \ $Q_C Q_D$	00	01	11	10
00	X	0	1	X
01	X	1	0	X
11	X	X	X	X
10	X	0	X	X

$K_D = Q_D \overline{Q_C} + \overline{Q_B} Q_C$



SEQUENCE DETECTOR.

*The sequence detector detects the specified input sequence. In this circuit ~~z~~ output goes high when a prescribed input sequence occurs. A typical input sequence and the corresponding output sequence for desired input sequence 101 are:

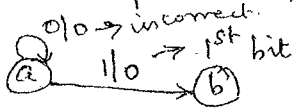


X = 0 0 1 1 0 1 1 0 0 1 0 1 0 1 1

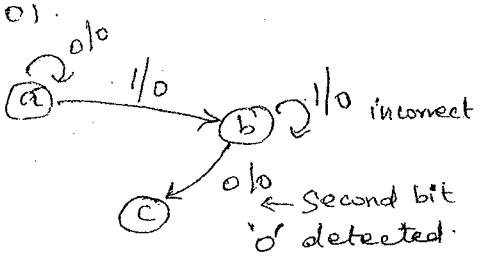
Z = 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0

1. Design a mealy type sequence detector to detect a serial input sequence of 101.

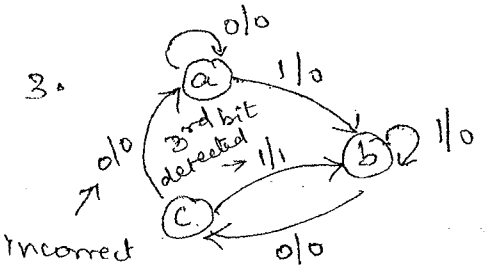
1.



2.



3.



4.

Present state	Next state AB, x=0	Next state AB, x=1	Output Z, x=0	Output Z, x=1
AB				
a	a	b	0	0
b	c	b	0	0
c	a	b	0	1

5. assigning a=00, b=01, c=10.

Present state A B	Next state AB, x=0	Next state AB, x=1	Output Z, x=0	Output Z, x=1
0 0	0 0	0 1	0	0
0 1	1 0	0 1	0	0
1 0	0 0	0 1	0	1

For A^+

AB \ x	0	1
00	0	0
01	1	0
10	X	X
11	0	0

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

For B^+

AB \ x	0	1
00	0	1
01	0	1
11	X	X
10	0	1

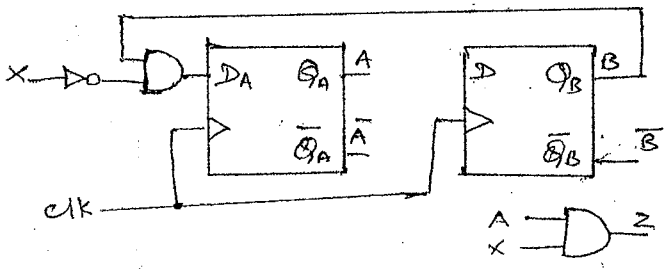
$B^+ = x$

For z

AB \ x	0	1
00	0	0
01	0	0
11	X	X
10	0	1

$z = Ax$

Logic diagram



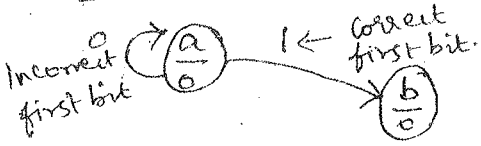
2. Design a Moore type sequence detector to detect a serial input sequence of 101.

* In a Moore type design output depends only on flipflop states. ∴ in the state diagram of Moore machine, the output is written with the state instead of with the transition between the states.

* The process of determining the state diagram is similar to that used for machine. Let us start with

state 'a' as an initial state

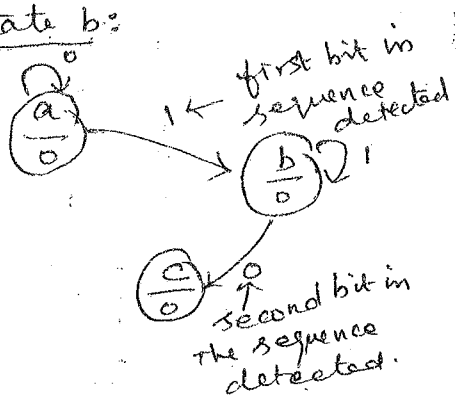
1. state a:



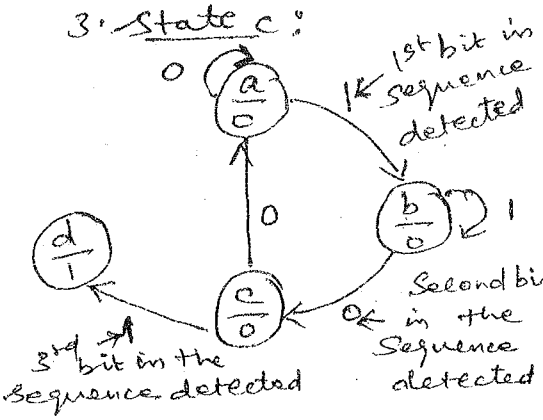
* when input is 1, we have detected the first bit in the sequence, hence we have to go the next state to detect the next bit in the sequence.

* when input is 0, we have to remain in state 'a' because bit 0 is not the first bit in the sequence. In both cases output is 0 since we have not yet detected all the bits in the sequence.

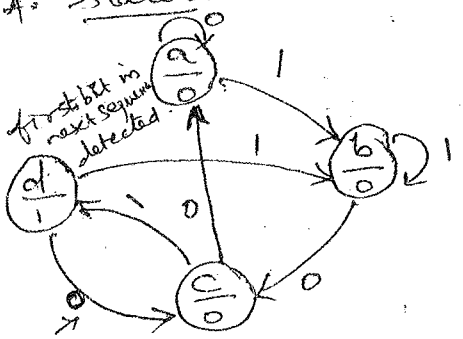
2. state b:



3. State c:



4. state d:



5. state table

Present State	Next state		output
	X=0	X=1	Z
a	a	b	0
b	c	b	0
c	a	d	0
d	c	b	1

6. Excitation table

PS AB	NS X=0	NS X=1	O/P Z
00	0	0	0
01	1	0	0
10	0	0	0
11	1	0	1

For A

AB \ X	0	1
00	0	0
01	(1)	0
11	0	0
10	(1)	0

$A^+ = \bar{A}B\bar{X} + A\bar{B}\bar{X}$

$= \bar{X}(A \oplus B)$

For B

AB \ X	0	1
00	0	(1)
01	0	(1)
10	0	(1)
11	0	(1)

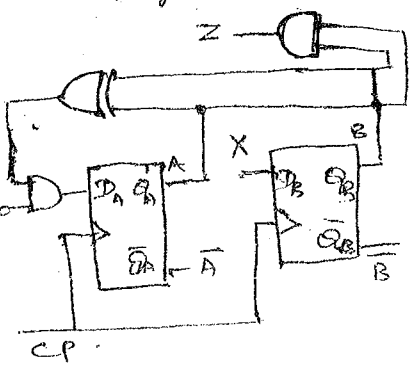
$B^+ = X$

For Z

A \ B	0	1
0	0	0
1	0	(1)

$Z = AB$

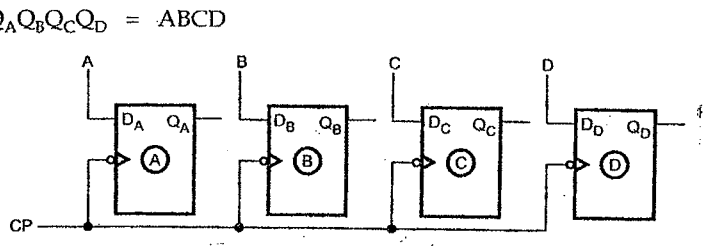
Logic diagram



REGISTERS

* A register is a group of flip flop. Each flip flop is capable of storing one bit of information.
* An n bit register consist of group of n flip-flops capable of storing n bits of binary information.
* In addition to flip-flops, a register may have combinational gates that perform certain data processing tasks.
* A register consist of a group of flip flops and gates that effect their transition. The flipflop hold the binary information and gates determine how the information is transferred into register.
* A register capable of shifting its binary information is one or both direction is called Shift register.

Buffer register:

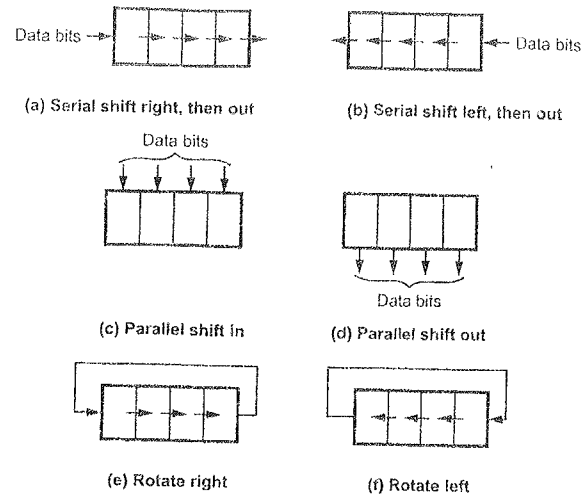


* Each D flipflop is triggered with a common negative edge clock pulse.
* The input bits set up the flipflops for loading.
* When the first negative clock edge arrives, the stored binary information becomes $ABCD = Q_A Q_B Q_C Q_D$.
* The Number of flipflop stages in a register determines its total storage capacity.

SHIFT REGISTERS:

* The data in a register can be moved from stage to stage within the register or into or out of the register upon application of clock pulses.

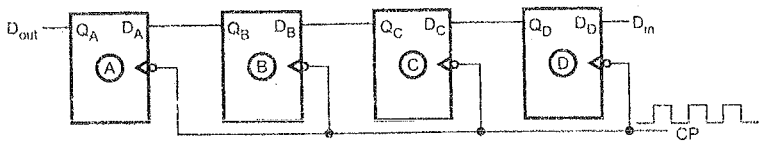
BASIC MOVEMENT IN REGISTERS



MODES OF OPERATION

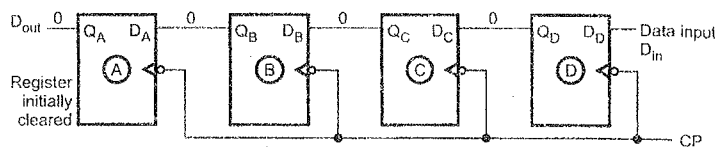
1. Serial in Serial out shift register

(i) Serial in serial out shift left register

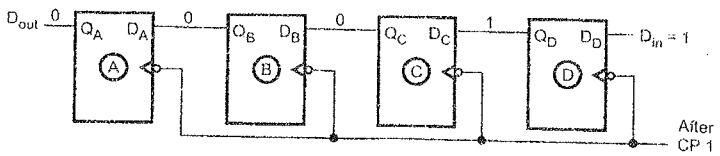


Let us include a 4 bit number 1111 into register beginning with most bit. Initially register is cleared.

$Q_A Q_B Q_C Q_D = 0000$



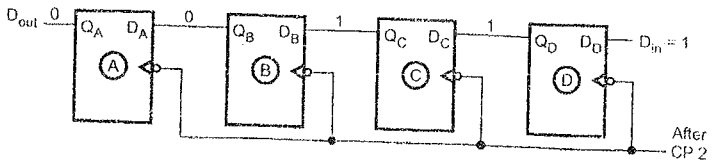
a) When data 1111 is applied serially (i.e) rightmost 1 is applied as Din.



$D_{in} = 1$, $Q_A Q_B Q_C Q_D = 1$ for the 1st clock pulse the 1st flipflop is set and accepts the data in. $D_{in} = 1$. As D flipflop $D_{in} = Q_A = 1$

$Q_A Q_B Q_C Q_D = 0001$.

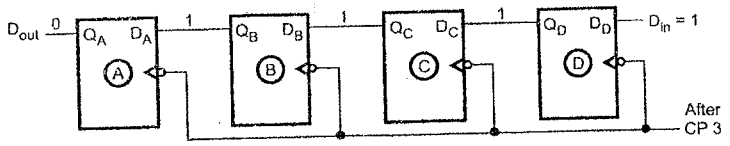
b)



* At the next clock pulse the next flipflop

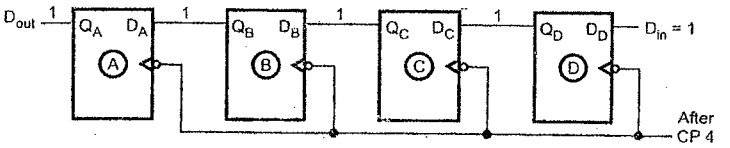
Q_B accepts the data and Q_C flipflop is set and register contents becomes $Q_A Q_B Q_C Q_D = 0011$.

c)



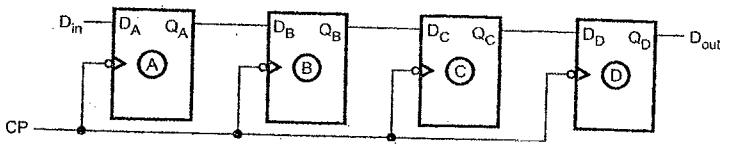
When third clock pulse is reached the flipflop Q_B is set and $Q_A Q_B Q_C Q_D = 0111$.

d)



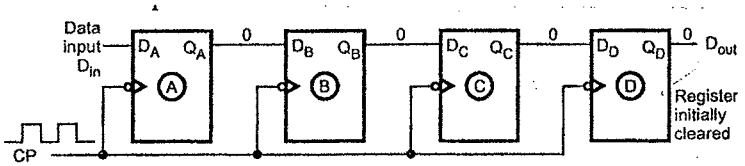
The fourth falling clock pulse given $Q_A Q_B Q_C Q_D = 1111$.

(ii) Serial in serial out shift right register



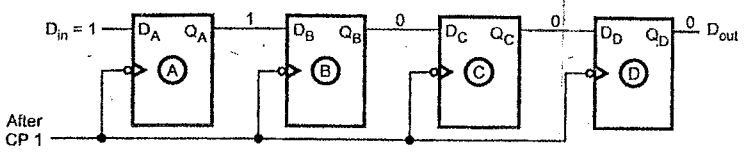
The entry of four bit binary number 1111 into the register, beginning with the left-most bit.

Initially the register is cleared. So $Q_A Q_B Q_C Q_D = 0000$.



When data is 1111 is applied serially

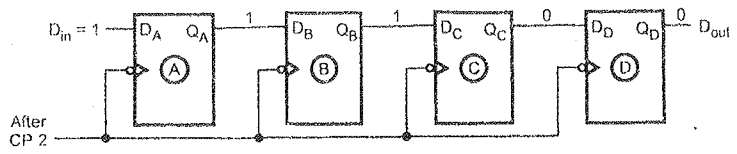
a) (i) leftmost 1 is applied as Din, $D_{in} = 1$, $Q_A Q_B Q_C Q_D = 0000$



At the arrival of the first falling clock edge sets the left-most flip flop and stored word becomes

$Q_A Q_B Q_C Q_D = 1000$.

b)

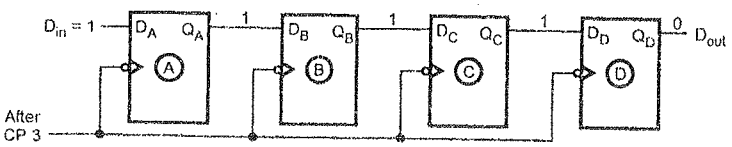


* When the (second) rising falling clock edge

bits, the Q_B flip flop sets and the register contents become,

$Q_A Q_B Q_C Q_D = 1100$

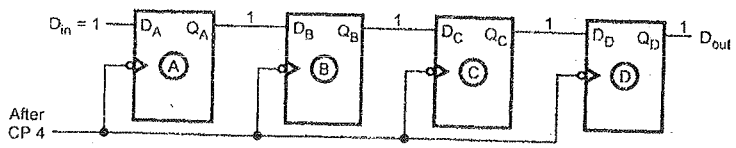
c)



* When the third falling clock edge

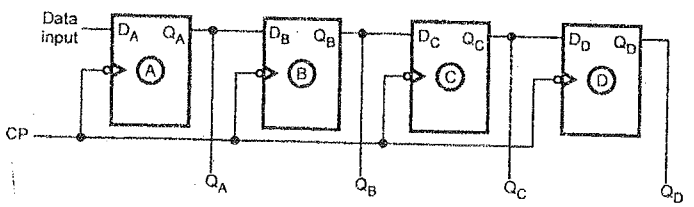
the result in $Q_A Q_B Q_C Q_D = 1100$.

d)



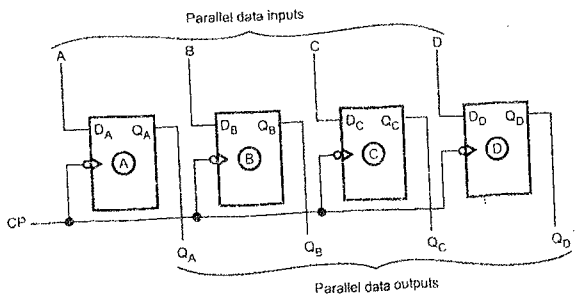
* For the 4th falling clock edge gives ..... $Q_A Q_B Q_C Q_D = 1111$.

2. SERIAL IN PARALLEL OUT SHIFT REGISTER.



* The 1st clock pulse entered the flipflop A is set Q_A output is obtained and the 2nd clock pulse Q_A is given as input to D_B and Q_B output is obtained and then D_A gets the 2nd data. The third clock pulse Q_C is set and Q_D is set for fourth clock pulse. That is the data bit is entered into the register serially and output is got parallelly.

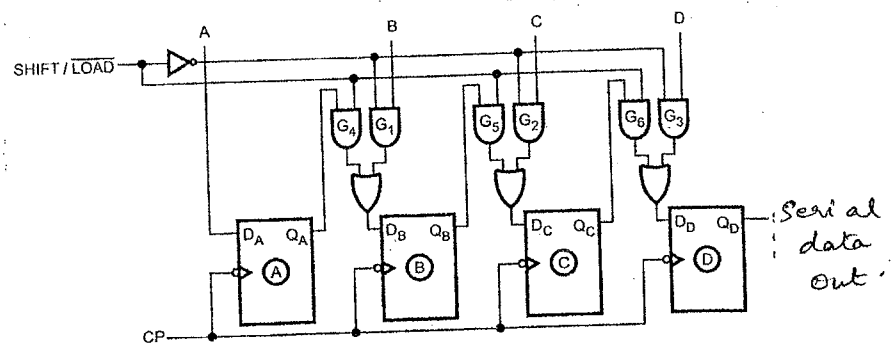
3. Parallel in Parallel out Shift register



* data inputs are entered in parallel (ie) simultaneously into their respective stages on parallel lines and output data

are taken parallelly. (ie) There is simultaneous entry of all data bits and the bits appear on parallel outputs simultaneously.

4. PARALLEL IN SERIAL OUT SHIFT REGISTER.



* 4 input lines A, B, C, D entering parallel into the register.
* shift / $\overline{load}$ is the control input which allows shift or loading data operation of the register.

shift / $\overline{load}$ is low:

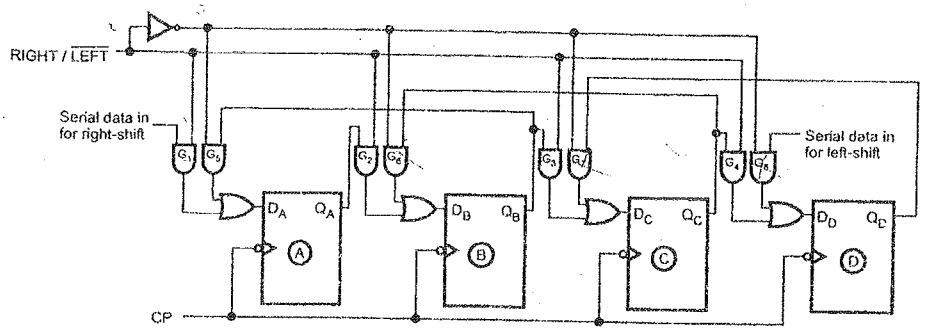
* When shift / $\overline{load}$ is low, gates G_1, G_2, G_3 are enabled, allowing each input data bit to be applied to D input of its respective flip flop.
* When a clock pulse is applied, the flipflops with D=1 will SET and those with D=0 will RESET.
* 4 bits are stored simultaneously.

shift / $\overline{load}$ is high:

* gates G_1, G_2, G_3 are disabled and gates G_4, G_5, G_6 are enabled.
* This allows the data with bits to shift right from one stage to next.
* OR gates at D input of flipflops allows either the parallel data entry operation or

Shift operation, depending on which AND gates are enabled by level on the shift / $\overline{\text{load}}$ input.

5. BIDIRECTIONAL SHIFT REGISTER



* This allows the shifting of data towards the right or left depending on the control line.
* The RIGHT/$\overline{\text{LEFT}}$ is the control input signal which allows data shifting either towards right or towards left.

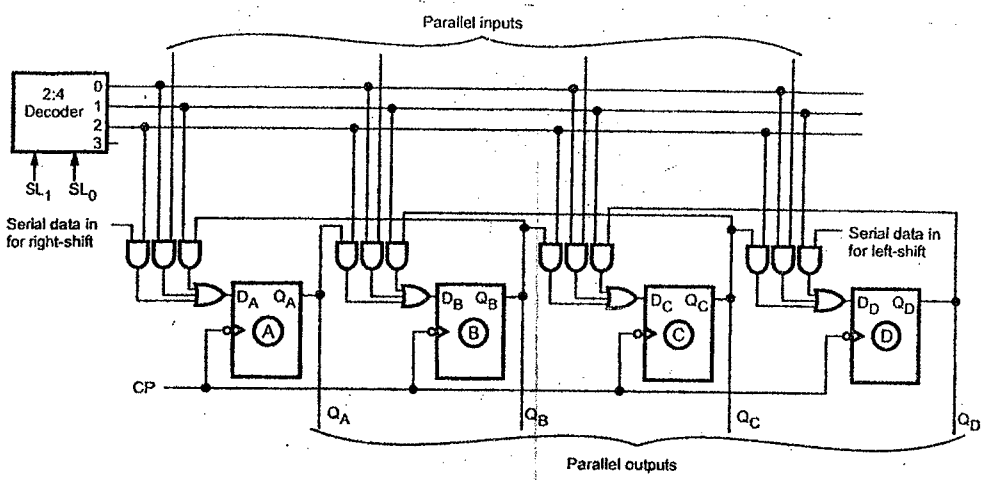
RIGHT/$\overline{\text{LEFT}}$ is high:

* When RIGHT/$\overline{\text{LEFT}}$ signal is high, gates G_1, G_2, G_3, G_4 are enabled. The state of Q output of each flip-flop is passed through the D input of the following flip-flop.
* When a clock pulse arrives, the data are shifted one place to the right.

RIGHT/$\overline{\text{LEFT}}$ is low:

When RIGHT/$\overline{\text{LEFT}}$ signal is low, the gates G_5, G_6, G_7, G_8 are enabled. The output Q of each flipflop is passed through the D input of the preceding flipflop. When clock pulse arrives, the data is shifted one place to left.

6. BIDIRECTIONAL SHIFT REGISTER WITH PARALLEL LOAD.



* Decoders are used for selecting the options and D flip flop are used.

* When parallel load capability is added to the shift register, the data entered in parallel can be taken out in serial fashion by shifting the data stored in the register is called bidirection shift register with parallel load.

*.

SL1	SL0	selected source.
0	0	parallel input
0	1	Output of right adjacent FF
1	0	" " left "
1	1	—

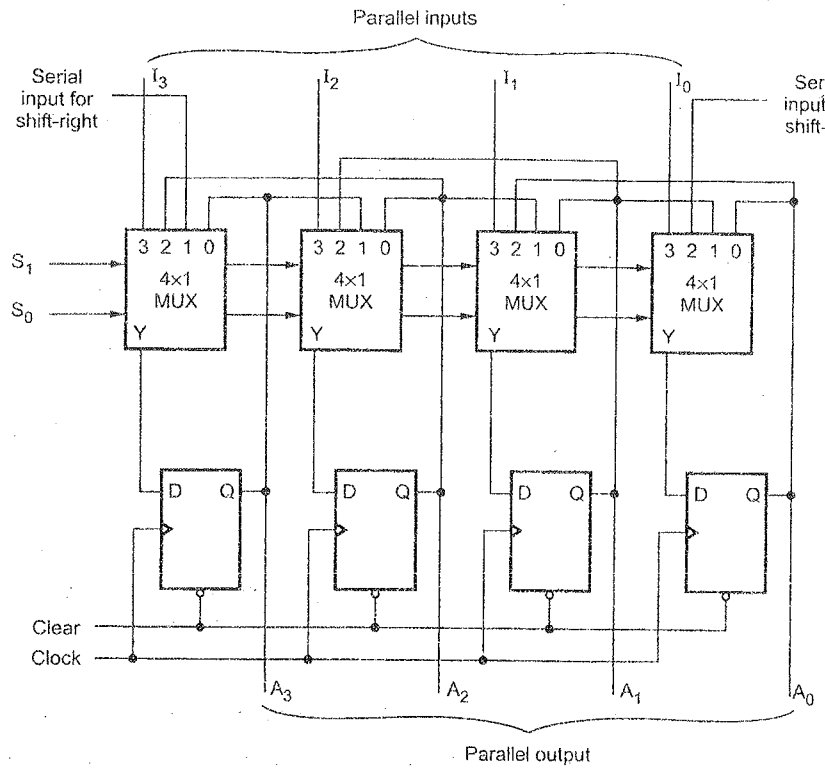
* When SL1 SL0 = 00, data from parallel inputs is loaded into 4 bit register.

* When SL1SL0 = 01, data within the register is shifted 1-bit left.

* When select line are 10 the data bits within the register is shifted 1-bit right

7. UNIVERSAL SHIFT REGISTER.

* A register capable of shifting both side(right and left) and parallel load capabilities is called Universal shift register.



* It consist of 4 flipflops and 4 multiplexers.
* Two selection line are present for common selection inputs S_1 and S_0.
*

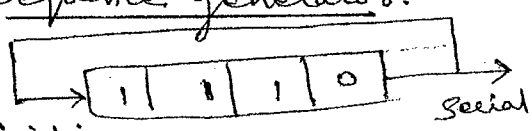
Mode Control		Register operation
S_1	S_0	
0	0	No change
0	1	Shift - right
1	0	shift - left
1	1	Parallel - load.

* When $S_1 S_0 = 00$ → input 0 is selected and the present value of register is applied to D inputs of flipflops. This results no change in register value.

* When $S_1S_0 = 01$, input 1 is selected and circuit connections are such that it operates as a right shift register.
* When $S_1S_0 = 10$, input 2 is selected and circuit connections are such that it operates as a left-shift register.
* When $S_1S_0 = 11$, the binary information on the parallel input lines is transferred into the register simultaneously and it is parallel load operation

APPLICATION OF SHIFT REGISTERS:

1. Serial-in-Serial out (SISO). used to introduce time delay of digital signal.
2. Serial in parallel out (SIPO) shift register can be used to convert data in serial form to parallel form.
3. Parallel in Serial out (PISO) shift register can be used to convert data in parallel form to serial form.
4. A shift register with Serial output connected back to serial input is called shift register counter eg: Johnson counter and ring counter
5. Sequence generator:



* Left most flipflop accepts serial input and right most flip-flop gives serial data output.

6. Sequence detector:
   * to detect desired sequence.