Counters designed without lock out problem are called as self starting counters

Counters - (illegible)

Lock Out Condition

In shortened modulus counters, there may occur this when it is switched on or at any time during counting. If at any time this counter falls into some unused or invalid states (for eg - in mod 12 counter states from 12 to 15 are invalid), the subsequent clock pulses may not cause it to come to a valid state. Then the counter become useless, as it stops working. Such counters with this feature is said to suffer from the problem of lock out.

To avoid this, while designing a logic circuit that resets the FF or that changes its state to a valid one should be included. ie all invalid states (if any exists) should be considered ~~and~~ in design and these should be allowed to change to a valid state after a clock pulse. So no don't cares are permitted in the design

Combining - diff Modulo Counters.

Different mod counters can be combined together to get another mod counter. eg:- a mod 2 & a mod 5 connected together gives a mod 10 counter. The connection between counters may be ripple but both should work in synchronism whether the connected ones are synchronous or asynchronous.

Synchronous Counters / parallel counters.

All FFs are triggered simultaneously or parallely by the clock pulses. Here propagation delay won't add together & is equal to the propagation delay of a single FF plus that of the gates involved. Additional points

- ✓ can operate at higher frequencies than that of ripple counter. (Since less time delay)
- ✓ high speed
- ✓ less severe decoding problems
- ✓ more circuitory elements
- ✓ presetting operation is called as loading of the counter.

Design of synchronous counters

- ✓ First identify the no. of FFs required. If mod N. Then no. of bits is selected such that the smallest value of n, for which no. of states.

$N \leq 2^n$,

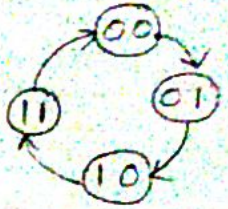
- draw the state diagram showing all possible states in the required sequence
- write down the present state as a table including all possibilities in that counter in the usual sequence.
- write the next state ~~as it~~ in the next column corresponding to each present state.
- Complete the excitation table of ~~each~~ FF to be used, for all the written ps & ns.
- Determine the expressions for the i/ps of each FF using K-map
- Draw the logic diagram accordingly.

q) Design a mod 4 synchronous ~~counter~~ using JK flip flop.

mod 4 ⇒ to count from 0 to 3 i.e 00 to 11 only 2 bits, only 2 FFs are required. ie using 2 FFs = $2^2 = 4$, ie only 4 states can be counted ie from 0 to 3. so full modulus so no lockout condition (it is only for shortened modulus) It is a self starting counter. unless specified design an up counter

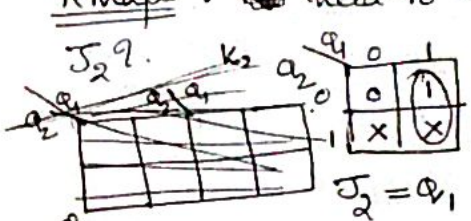
Present state Q_2 Q_1	Next state Q_2 Q_1	J_2	K_2	J_1	K_1
0 0	0 1	0	x	1	x
0 1	1 0	1	x	x	1
1 0	1 1	x	0	1	x
1 1	0 0	x	1	x	1

State diagram

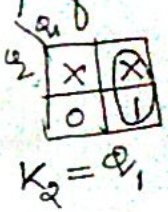


Kmaps . need to draw k-maps for all i/ps

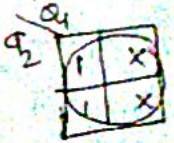
J_2 ?.



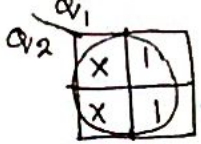
$J_2 = Q_1$



$K_2 = Q_1$

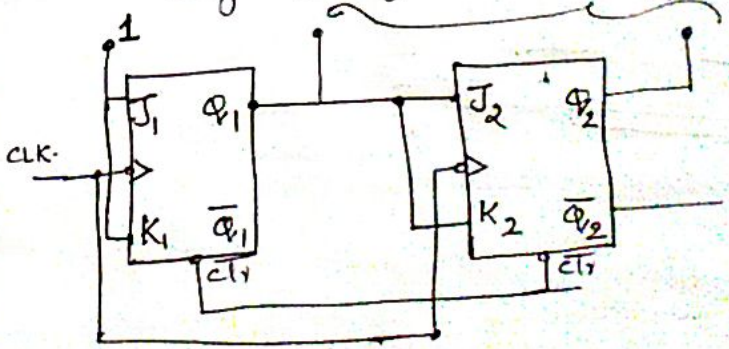


$J_1 = 1$



$K_1 = 1$..

Logic Diagram

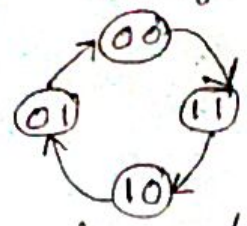


Mod-4 synchronous up counter.

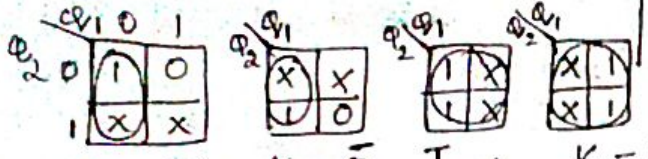
Mod 4 synchronous down counter

Present state Q_2 Q_1	Next state Q_2 Q_1	J_2	K_2	J_1	K_1
0 0	1 1	1	x	1	x
0 1	0 0	0	x	x	1
1 0	0 1	x	1	1	x
1 1	1 0	x	0	x	1

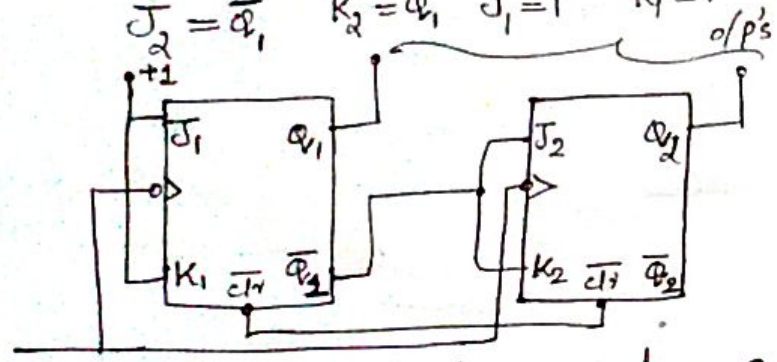
State diagram



Write present state in the next sequence & next state as per state diagram



$J_2 = \overline{Q_1}$ $K_2 = \overline{Q_1}$ $J_1 = 1$ $K_1 = 1$



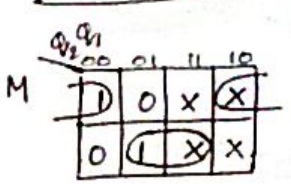
Mod 4 synchronous down counter

③ Mod 4 synchronous up down counter

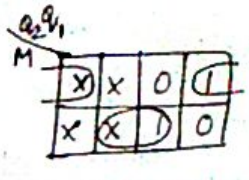
Include an external control bit named 'M' to control the counting. By Design it considering M also as parameter along with the output bits while drawing K maps.

Assume up counting for M = 1, & down counting for M = 0.

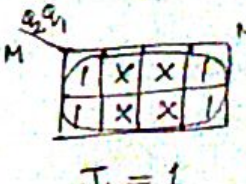
M	Present state Q_2 Q_1	Next state Q_2 Q_1	J_2	K_2	J_1	K_1
0 (down counting)	0 0	1 1	1	x	1	x
0	0 1	0 0	0	x	x	1
0	1 0	0 1	x	1	1	x
0	1 1	1 0	x	0	x	1
1 (up counting)	0 0	0 1	0	x	1	x
1	0 1	1 0	1	x	x	1
1	1 0	1 1	x	0	1	x
1	1 1	0 0	x	1	x	1



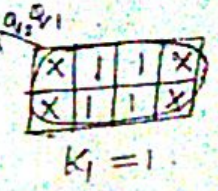
$J_2 = \overline{M} + MQ_1$



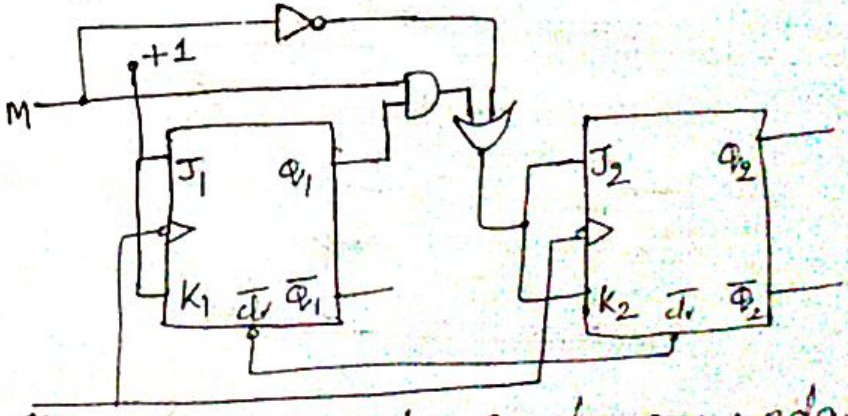
$K_2 = \overline{M} + MQ_1$



$J_1 = 1$



$K_1 = 1$



Mod 4 synchronous up down counter

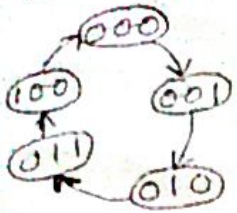
Mod 5 Counter using T flip flop

1) mod 5 → to count from 0 to 4, i.e. 000 to 100 ⇒ 3 bits → 3 FFs.

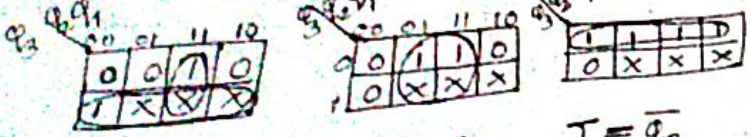
1) with lock-out

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

state diagram



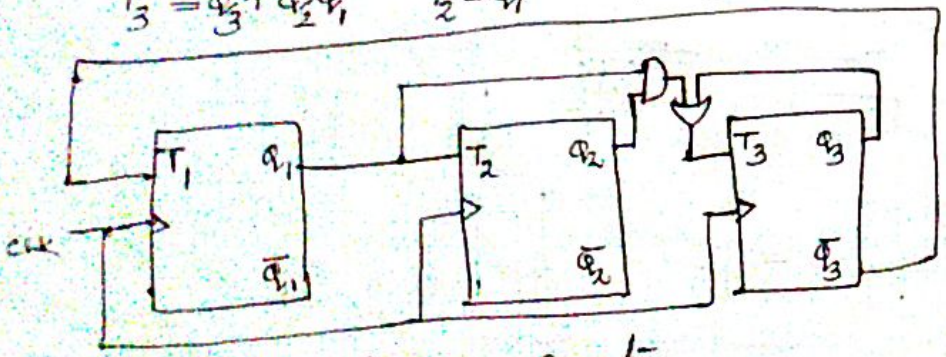
The cells in K-maps corresponding to the invalid states 5, 6 & 7 are represented as don't cares.



$T_3 = Q_3 + Q_2Q_1$

$T_2 = Q_1$

$T_1 = \overline{Q_3}$



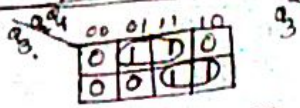
Mod 5 Synchronous Counter

② without lock-out / self starting counter

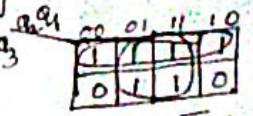
All invalid states i.e 5, 6, 7 should be transformed to a valid state (say zero state) after a CLK.

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

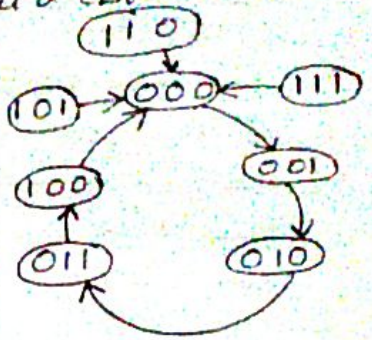
$T_3 = Q_3 + Q_2Q_1$

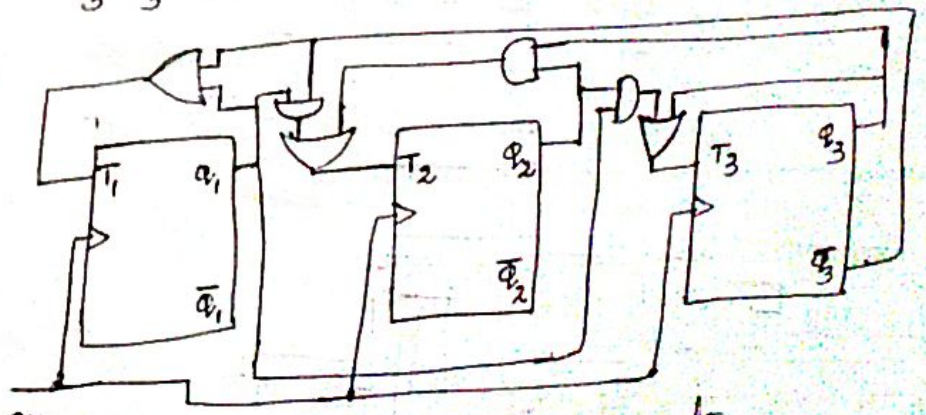


$T_2 = \overline{Q_3}Q_1 + Q_3Q_2$



$T_1 = \overline{Q_3} + Q_1$





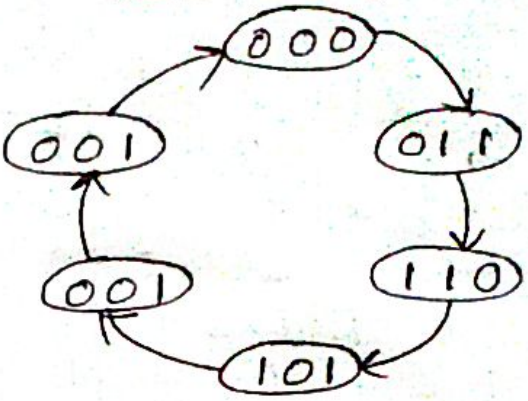
Mod-5 - self starting Counter

Sequence Generator

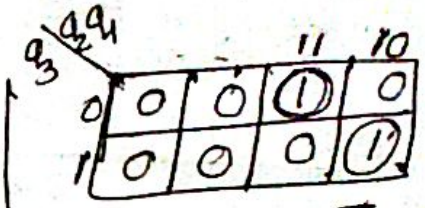
Generate the given sequence 0, 3, 6, 5, 1, 0, 3, ... using DFF.

The invalid states are 2, 4, 7. All these states should be transferred to a valid state. say zero here.

State Diagram



Present state			Next state					
Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	D_3	D_2	D_1
0	0	0	0	1	1	0	1	1
0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0
0	1	1	1	1	0	1	1	0
1	0	0	0	0	0	0	0	0
1	0	1	0	0	1	0	0	1
1	1	0	1	0	1	1	0	1
1	1	1	0	0	0	0	0	0

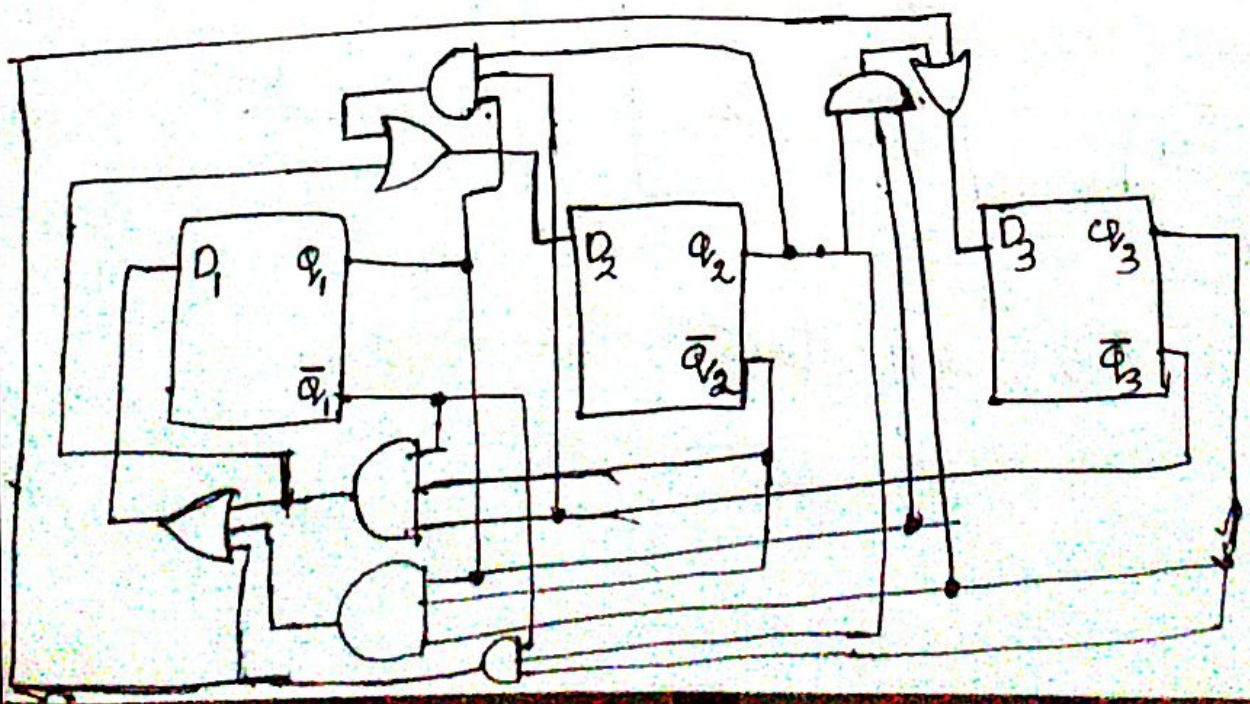


$D_3 = q_3q_2q_1 + q_3\bar{q}_2\bar{q}_1$

$D_2 = \bar{q}_3\bar{q}_2\bar{q}_1 + \bar{q}_3q_2q_1$

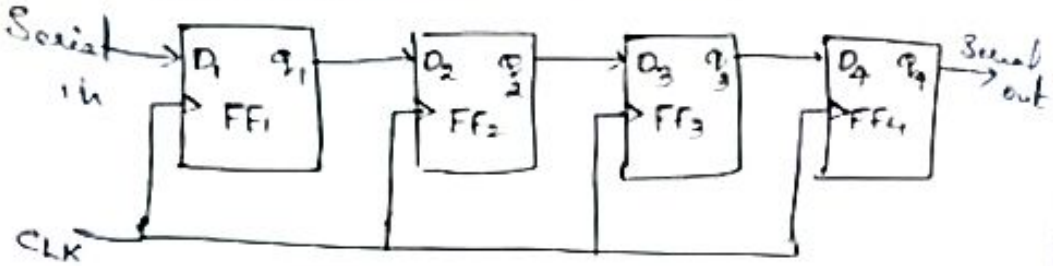
$= \bar{q}_3(\bar{q}_2\bar{q}_1 + q_2q_1)$

$D_1 = \bar{q}_3\bar{q}_2\bar{q}_1 + q_3\bar{q}_2q_1 + q_3q_2\bar{q}_1$



Shift Registers

SISO (serial in serial out)



serial in – 1 bit input
serial out – 1 bit output.
o/p of 1st is given as i/p to 2nd & so on & o/p of last FF (shift right)

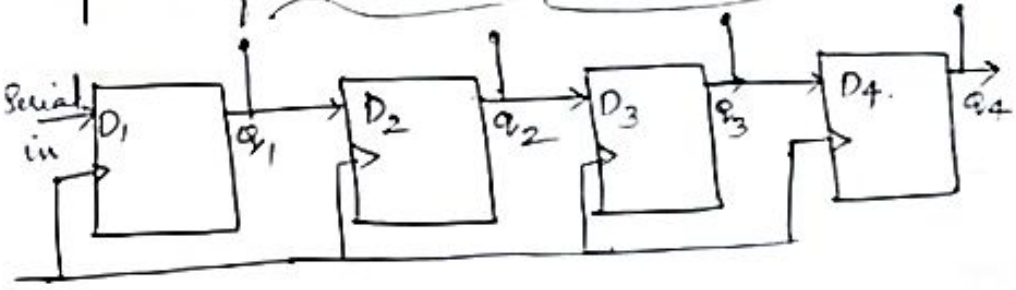
here +ve edge triggering. (the serial i/p)
so after 1st +ve edge of CLK, value of Q_A, comes at Q_1 & acts as i/p to 2nd. after 2nd CLK edge, this value reaches Q_2 i.e at o/p of 2nd FF, after 3rd CLK, it reaches the o/p point of 3rd FF i.e at Q_3 & after 4th CLK pulse, it reaches the final o/p point i.e at Q_4. Thus in a 4-bit SISO register, the data i/p will reach the o/p point only after 4 clock pulses. If there are only 3 FFs (3 bit) it will take 3 pulses.

the following table shows the data flow in a SISO register. Initial value as zeros at o/p points

CLK	D_1 (serial i/p)	Q_1	Q_2	Q_3	Q_4 (serial o/p)
0	0	0	0	0	0
1	1	1	0	0	0
2	0	0	1	0	0
3	1	1	0	1	0
4	0	0	1	0	1

SIPO (serial in parallel out)

serial in (only bit as i/p)
parallel o/p (no: of o/p bits = no of FF. i.e o/p of each FF)
Consider a 4 bit register, so 4 FFs are required
i/p of 1st FF as serial data i/p. connect o/p of 1st to i/p of 2nd, o/p of 2nd to i/p of 3rd & so on, since data i/p is serial type. take o/p's from the o/p points of all 4 FFs



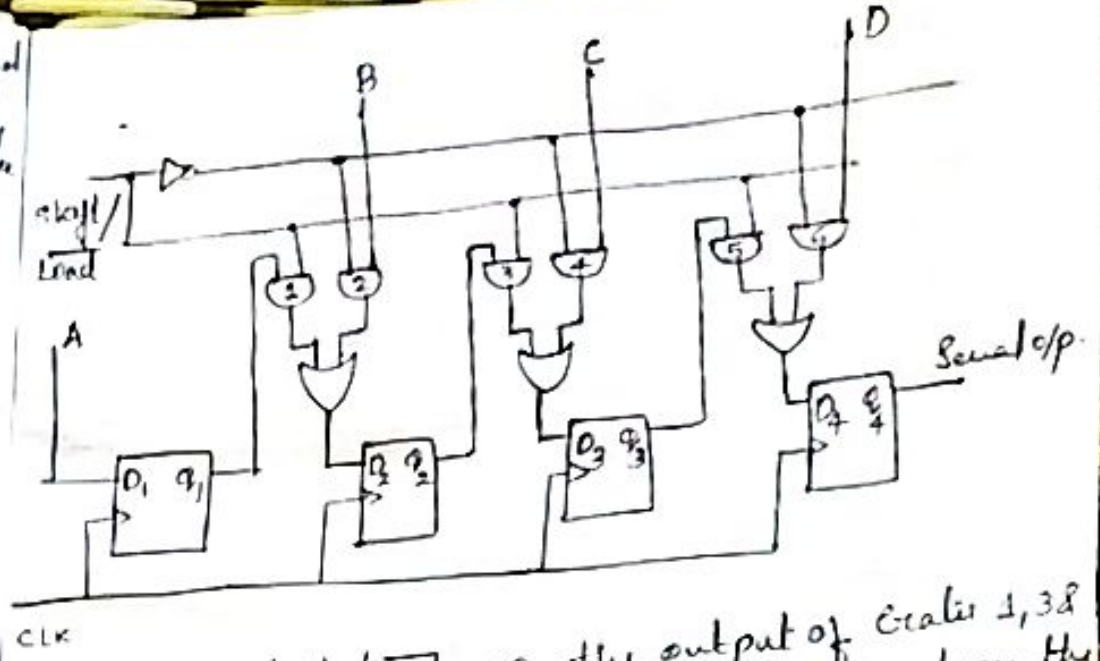
Here also during 1st clk pulse serial i/p is transferred to o/p of 1st, during 2nd it will be transferred to the o/p side of 2nd & finally after 4th pulse it will reach the o/p point of last FF

	Serial	Parallel o/ps			
CLK	D_1 (i/p)	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0	0
1	1	1	0	0	0
2	1	1	1	0	0
3	0	0	1	1	0
4	1	1	0	1	1

cleared initially

PISO

Parallel i/p ⇒ no no of i/p bits = no of FFs & all i/p bits of each FF all inputs are given simultaneously and the o/p is to be taken from the last FF (i only 1 bit). Here the data should be loaded simultaneously (no diff values at diff i/ps) but should be transferred for shifted to the next FF also so for that an external control i/p known as shift/load is used.

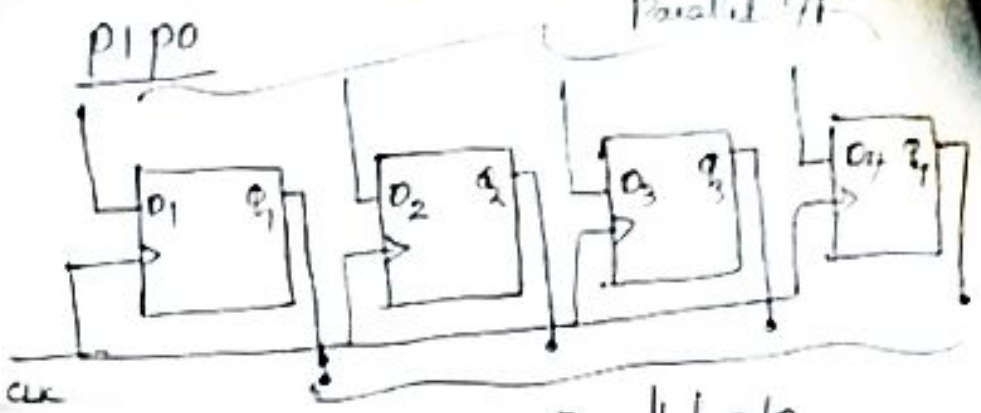


when shift/load = 0, the output of gates 1, 3 & 5 are zero, so o/p of each OR gate depends on the 2nd AND gate. Thus depends on of the i/ps to gates 2, 4 & 6 is now 1, so the o/p will be values of B, C, D since so under this condition the parallel data will be directly loaded to the input point of respective FFs. Now apply the CLK signal, after 1st CLK, the parallel data loaded will be transferred to the respective FFs output side i.e A to Q_1, B to Q_2, C to Q_3 & D to Q_4. Now change, shift/Load to 1, then o/ps of gates 2, 4 & 6 become 0, then o/p of OR gate is o/p of the 2nd AND gate (1, 3 & 5). One of the

i/ps of gate 1, 3 & 5 is now 1. Then its o/p will be the other i/ps i.e Q_1, Q_3, & Q_3. So what happens is that, when shift/load = 1, the o/ps of previous FF comes to the i/p of the next FF i.e Q_1 at D_2, Q_2 at D_3 & Q_3 at D_4. Now during the next pulse this data will be transferred to the o/ps of the respective FF. Thus the 4 parallel bits loaded, can be taken out after 4 clock pulses only. The next set of parallel data will be given only after 4 pulses.

When ABCD = 1010.

Shift/load	CLK	D_1 D_2 D_3 D_4 . Q_1 Q_2 Q_3 Q_4
0	0	1 0 1 0 0 0 0 0 ⇒ clear initially Parallel data loaded.
1.	1	1 0 1 0 → data o/p
1	2	0 1 0 1 → "
1	3	0 0 1 0 → "
1	4	– – – 1 → "



Parallel i/p = no of i/ps = no of o/ps = no of parallel o/p FFs

data is given & taken simultaneously. The given data can be taken out at every clr pulse.

	in				out			
CLK	D_1	D_2	D_3	D_4	Q_1	Q_2	Q_3	Q_4
1	1	0	1	0	1	0	1	0
2	1	0	1	1	1	0	1	1

Applications of shift registers

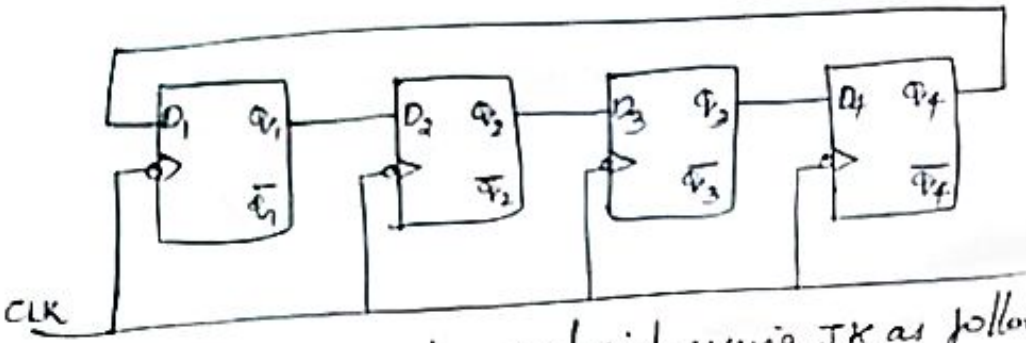
1. To convert serial data to parallel form
2. To convert parallel data to serial form
3. Universal shift register — can convert both parallel to serial & serial to parallel

(4) For generating time delays, by controlling the clock frequency and by fixing the no. of stages (FFs) in the register

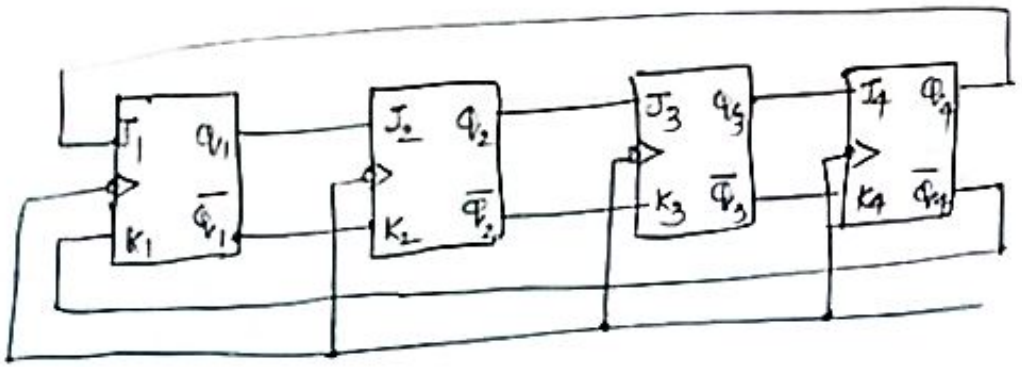
(5) Ring Counter

(6) Johnson Counter

Ring Counter - modification of SISO register, by connecting the Q output of last FF to the 'D' input of first FF.



The same can be realized using JK as follows.



here J_1 & K_1 are compliment values of Q_4 & $\bar{Q}_4$. so the J value will be directly transferred to the output side at every clock edge. It is fed back from

Working

Set all FFs except the first one are cleared initially. Then Preset the first flip (any one can be preset as per the requirement). Thus $Q_1 = 1$ & $Q_2 = Q_3 = Q_4 = 0$, initially. Now after clock pulse. At every -ve CLK edge (Since -ve edge triggered as shown in fig) the data at i/p will be transferred to o/p. The data is shifted in the sequence table below

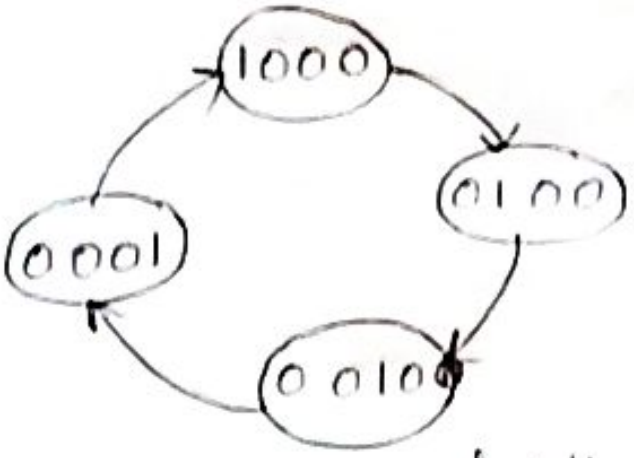
CLK	D_1	D_2	D_3	D_4	Q_1	Q_2	Q_3	Q_4
0	0	0	0	1	0	0	0	1
1	1	0	0	0	1	0	0	0
2	0	1	0	0	0	1	0	0
3	0	0	1	0	0	0	1	0
4	0	0	0	1	0	0	0	1

Initially, i/p to 1st FF is $D_1 = Q_4 = 0$, & $D_2 = Q_1 = 1$, $D_3 = Q_2 = 0$, & $D_4 = Q_3 = 0$. After 1st CLK pulse, Q_1 becomes $D_1 = 0$, & $Q_2 \to 1$, $Q_3 \to 0$ & $Q_4 \to 0$

Sequence Table

CLK	Q_1	Q_2	Q_3	Q_4
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0
5	0	1	0	0
6	0	0	1	0
7	0	0	0	1

This continues & the continues as a ring
Hence the name.
State diagram - the output states in the sequence obtained is shown in the following state diagram.

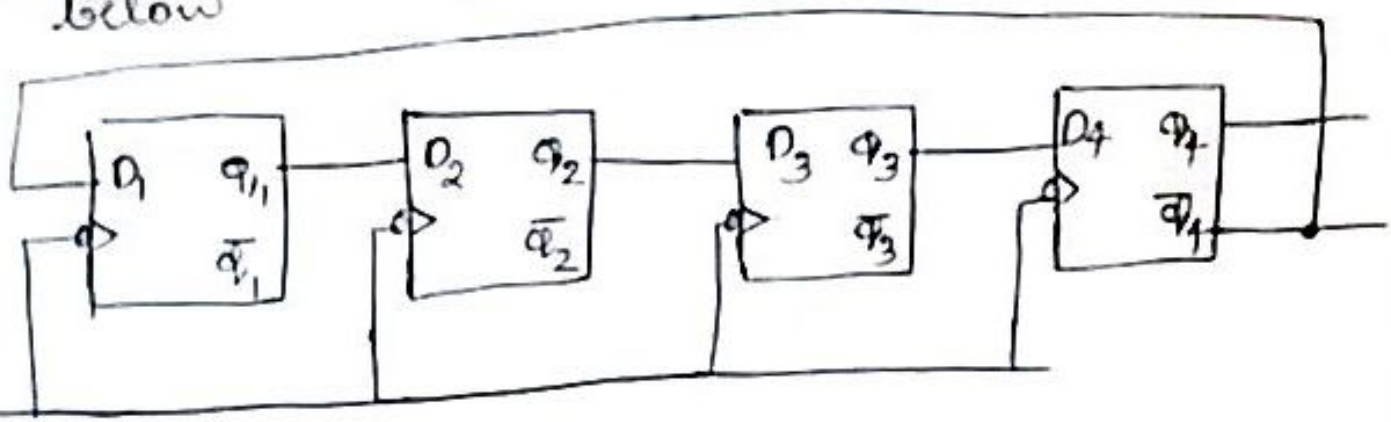


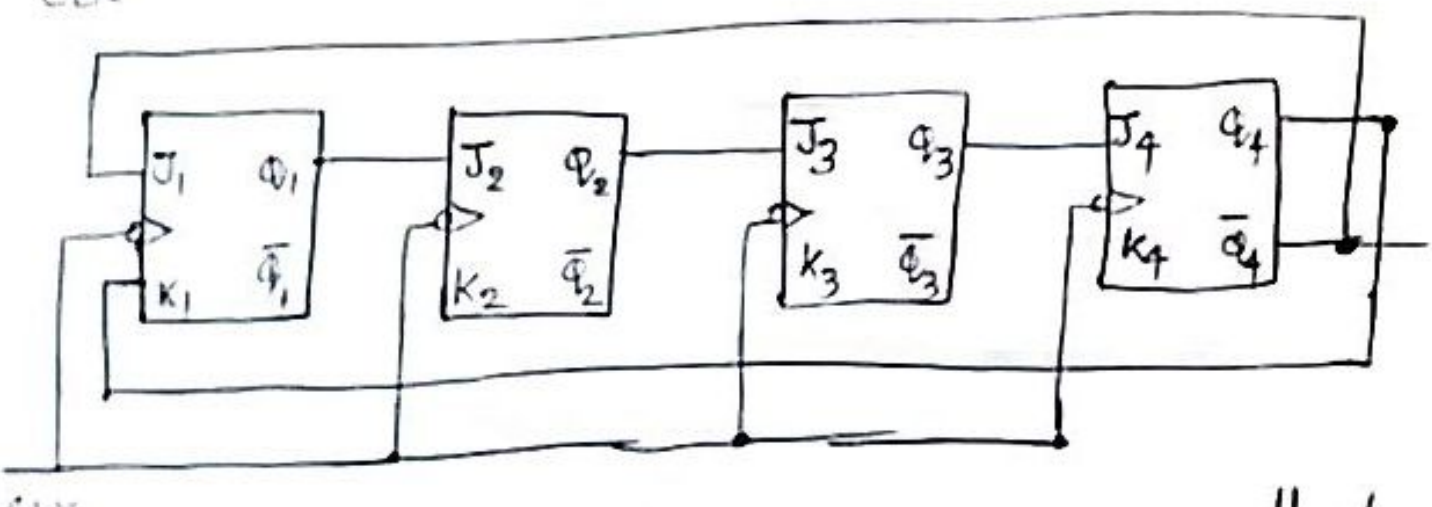
The no. of states = no. of FFs = mod of the counter
Here it is 4. It can count only 4 bits
where as an n bit ripple counter can count 2^n bits.
Advantage - no decoder is required,
- it's synchronous type
- no gates external to FFs
- so very fast, but suffer from lock out

Johnson Counter / Twisted ring counter

Modification of SISO register again, but the feedback provided is from the inverted output ($\bar{Q}$) of the last FF to the

D input of first FF is similar to ring, Q output of each stage is connected to the D input of next stage. Same $\bar{Q}$ of 4th FF is connected back to the first, it is called as twisted ring counter. Logic diagram using D & JK, state diagram & sequence table are shown below





CLK	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0

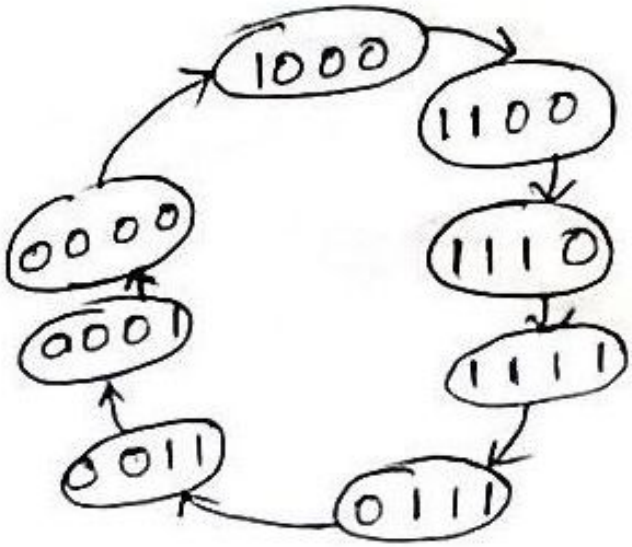
} at CLK = 0, all o/p are zero picked all FFs are reset initially after each CLK, the level or value of Q_1 will be shifted to Q_2, value of Q_2 to Q_3, Q_3 to Q_4 & last the value of $\bar{Q}_4$ to Q_1 and the sequence

is as shown in the sequence table.

CLK	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0
9	1	0	0	0

Sequence Table.

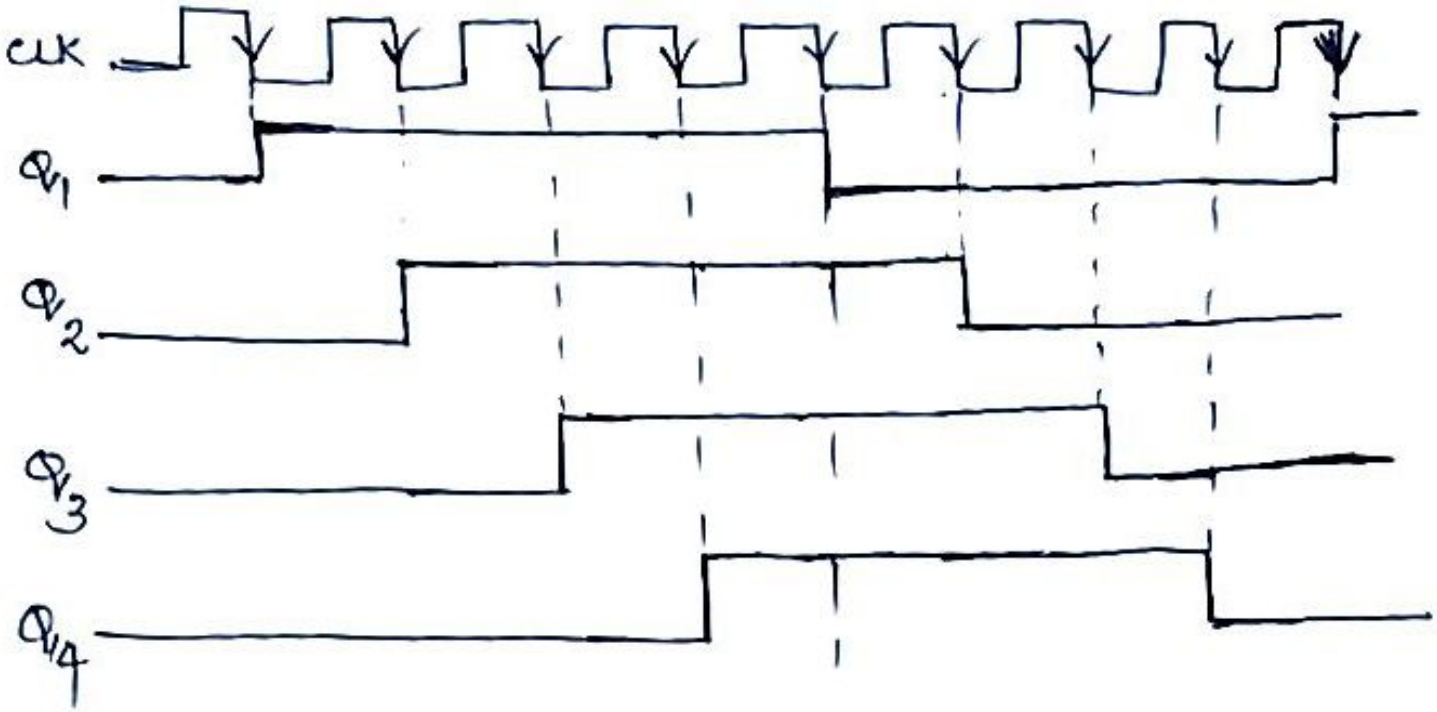
state diagram.



An n FF johnson counter can have 2n states and can count upto 2n pulses. So it is a mod 2n counter.

features :- more economical than ring
- less economical than ripple
- requires two i/p gates for decoding
- suffers from lock out

Timing diagram.



Timing diagram of ring counter