SWITCHING THEORY AND LOGIC DESIGN

UNIT-4

BY

S.SAGAR KRISHNA,

ASSISTANT PROFESSOR,

GAYATRI VIDYA PARISHAD COLLEG OF ENGINEERIN(A)

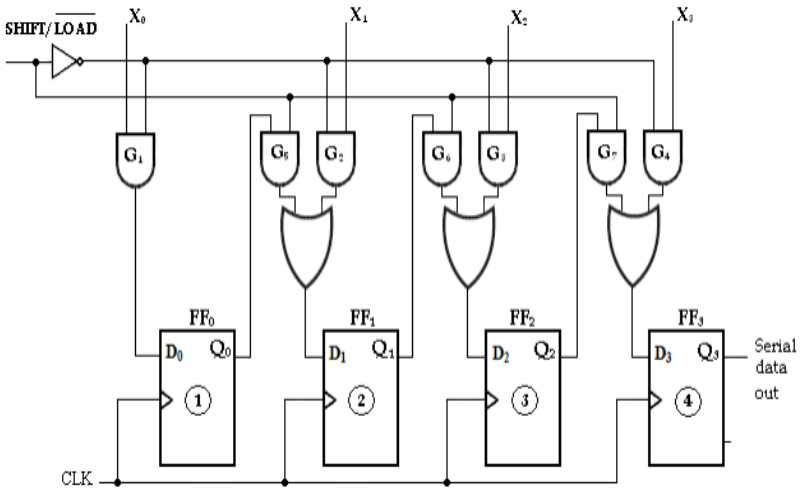
PARALLEL IN SERIAL OUT SHIFT REGISTER

In this type, the bits are entered in parallel i.e., simultaneously into their respective stages on parallel lines.

There are four data input lines, X0, X1, X2 and X3 for entering data in parallel into the register.

SHIFT/ LOAD input is the control input, which allows four bits of data to load in parallel into the register.

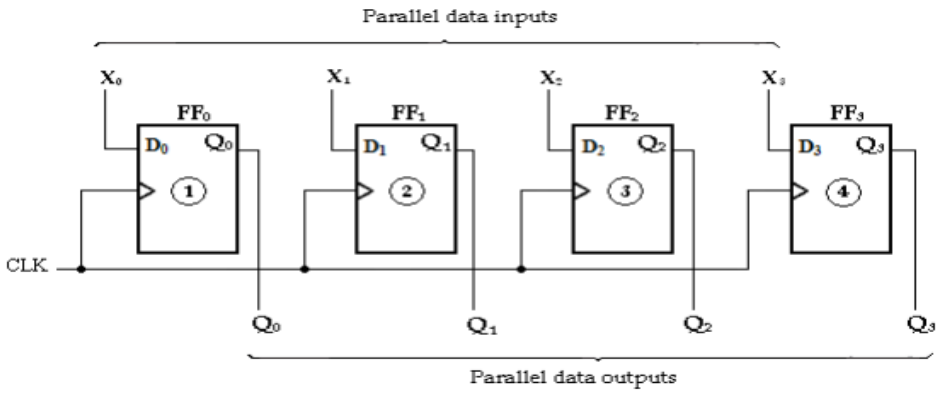
When SHIFT/LOAD is LOW, gates G1, G2, G3 and G4 are enabled, allowing each data bit to be applied to the D input of its respective Flip-Flop. When a clock pulse is applied, the Flip-Flops with D = 1 will set and those with D = 0 will reset, thereby storing all four bits simultaneously.



PARALLEL IN SERIAL OUT SHIFT REGISTER

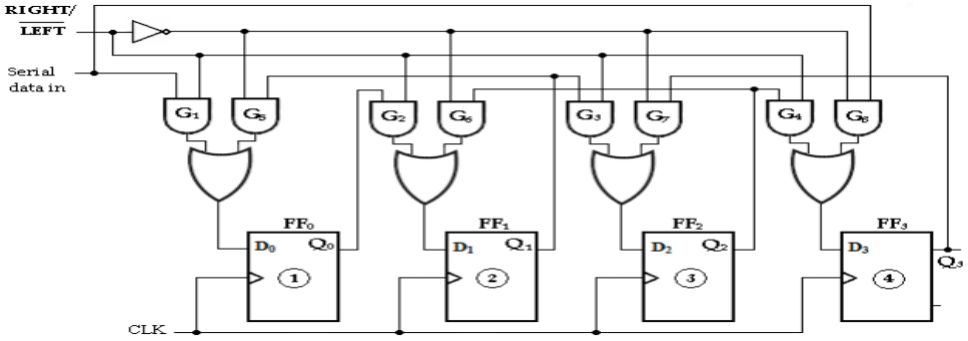
When SHIFT/LOAD is HIGH, gates G1, G2, G3 and G4 are disabled and gates G5, G6 and G7 are enabled, allowing the data bits to shift right from one stage to the next. The OR gates allow either the normal shifting operation or the parallel data entry operation, depending on which AND gates are enabled by the level on the SHIFT/LOAD input.

Parallel-In Parallel-Out Shift Register:



BI-DIRECTIONAL SHIFT REGISTER

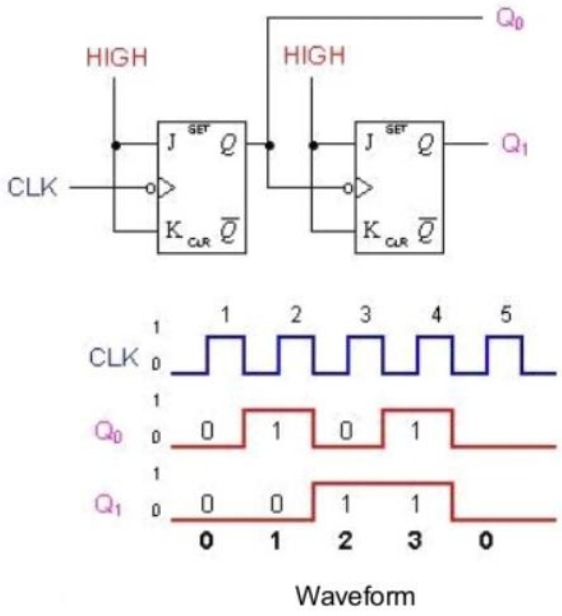
- Here data can be shifted either left or right.
- A HIGH on the RIGHT/LEFT control input allows data bits inside the register to be shifted to the right, and a LOW enables data bits inside the register to be shifted to the left. When the RIGHT/LEFT control input is HIGH, gates G1, G2, G3 and G4 are enabled, and the state of the Q output of each Flip-Flop is passed through to the D input of the following Flip-Flop. When a clock pulse occurs, the data bits are shifted one place to the right.
- When the RIGHT/LEFT control input is LOW, gates G5, G6, G7 and G8 are enabled, and the Q output of each Flip-Flop is passed through to the D input of the preceding Flip-Flop. When a clock pulse occurs, the data bits are then shifted one place to the left.



COUNTER

- A *counter* is a register that goes through a predetermined sequence of states upon the application of clock pulses.
  - Asynchronous counters
  - Synchronous counters
- **Asynchronous Counters** (or *Ripple counters*)
  - the clock signal (CLK) is only used to clock the first FF.
  - Each FF (except the first FF) is clocked by the preceding FF.
- **Synchronous Counters**,
  - the clock signal (CLK) is applied to all FF, which means that all FF shares the same clock signal,
  - thus the output will change at the same time.

2 BIT ASYNCHRONOUS COUNTER OR MOD-4 UP COUNTER

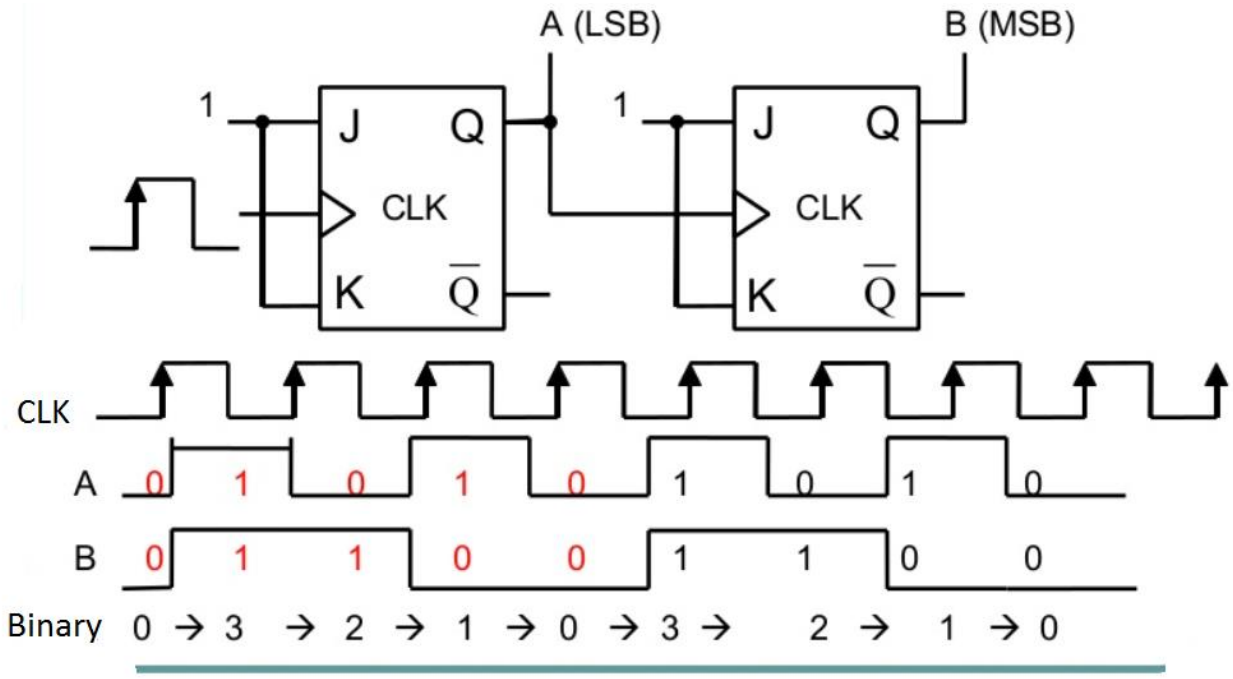


Waveform

A two-bit asynchronous counter is shown on the left. The external clock is connected to the clock input of the first flip-flop (FF0) only. So, FF0 changes state at the falling edge of each clock pulse, but FF1 changes only when triggered by the falling edge of the Q output of FF0.

If working not clear view this video
https://www.youtube.com/watch?v=iaIu5SYmWVM&list=PLuYnCh-Sh1Xd5cLa-CfK883tPmJwrjSwF

ASYNCHRONOUS MOD-4 DOWN COUNTER



DESIGN PROCEDURE ASYNCHRONOUS COUNTER

- Each Flip-Flop will give 2 states 0 and 1.So N flip-flop will give 2^n states
- First Determine no of Flip-Flop Required. If it is a 2 bit 2 flip-flop are required.

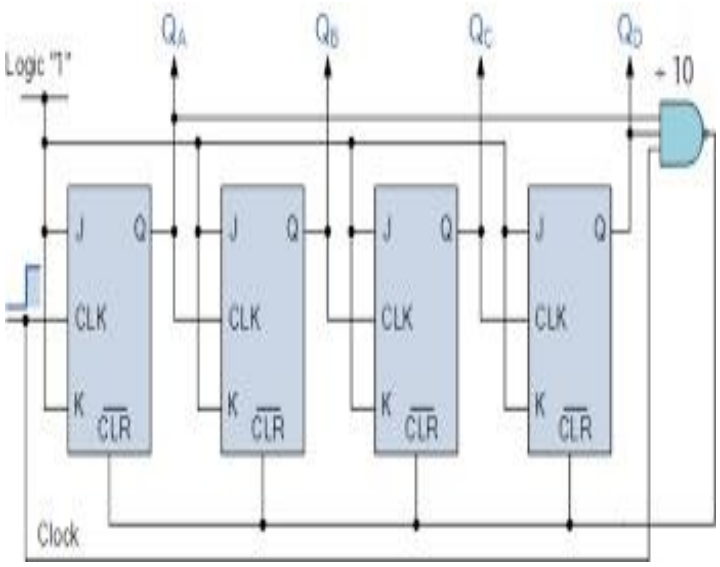
MOD-n Counter :(If n is a multiple of 2)then

- if Mod-16 counter is written then it has 16 states starting from 0000 to 1111 in binary. or to get 16 states, n has to be 4.

MOD-n Counter :

- If n is not a multiple of 2
- First identify no of states, then write the state in binary format. Then the maximum state is considered if it is a r bit then r flip flop are required. But r flip-flop will give 2^r States. In order to get maximum state use AND gate for active clear or NAND gate for low clear
- Ex:MOD-10 counter means it has 10 states starting from 0000 to 1001 in binary since the last state is a 4 bit number 4 flip flop are required. But 4 flip-flops will give output as 16 states. So we design the circuit in such a way at 1010 the flip-flop has to be reset to zero by connecting NAND gate with inputs as Q3,Q1 for low clear.

MOD-10 ASYNCHRONOUS COUNTER



Logic Diagram

Input pulse count	*Counter states*			
	A	*B*	*C*	*D*
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	0
4	0	0	1	0
5	1	0	1	0
6	0	1	1	0
7	1	1	1	0
8	0	0	0	1
9	1	0	0	1
10 (0)	0	0	0	0

Functional Table here Q_D=A,Q_C=B,O_B=C,Q_A=D

MOD-10 ASYNCHRONOUS COUNTER

Working:

- At first clock output will be 0001 then clr=1 as Q_D=0,Q_B=0 then the output will change to 0010 at next clock then clr=1 as Q_D=0,Q_B=1.This process will continue until output reaches 1001
- When output is 1001 clr=1 as Q_D=1,Q_B=0 so at next clock output reaches 1010 but immediately clr=0 as Q_D=1,Q_B=1 then all the flip-flop will reset to 0000 and then once again count continues

If any doubts view this video
https://www.youtube.com/watch?v=fKVZpupyP_o

Disadvantages: It will take more time for operation as clock is not common for all flip-flops but it is easy to design

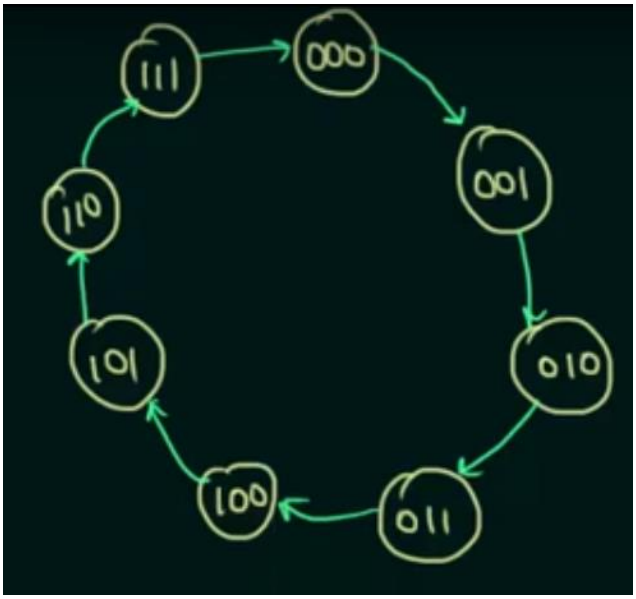
SYNCHRONOUS COUNTER DESIGN PROCEDURE

Advantage: Faster in operation and more components are required

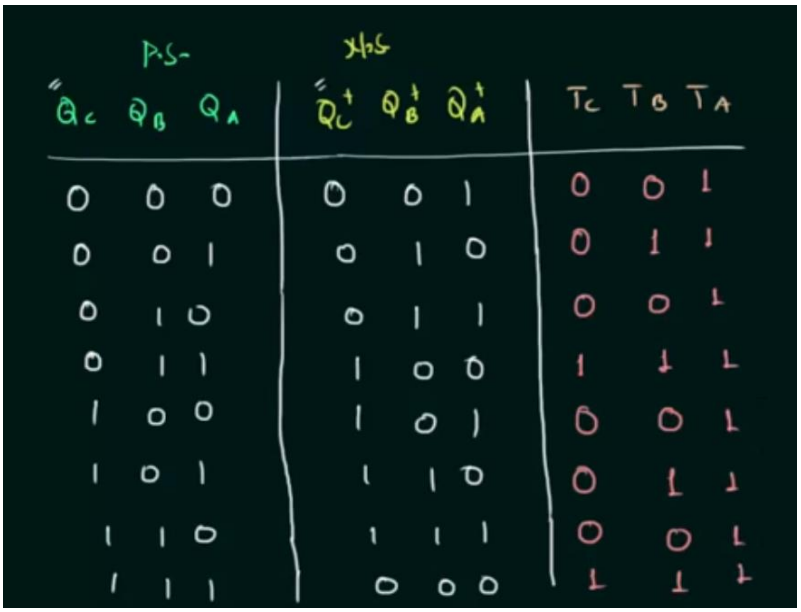
Design Steps:

1. The given problem is determined with a state diagram.
2. From the state diagram, obtain the state table.
3. Assign binary values to each state (Binary Assignment) if the state table contains letter symbols.
4. Determine the number of Flip-Flops.
5. Choose the type of Flip-Flop (SR, JK, D, T) to be used.
6. From the state table, circuit excitation and output tables.
7. Using K-map or any other simplification method, derive the circuit output functions and the Flip-Flop input functions.
8. Draw the logic diagram.

MOD-8 SYNCHRONOUS COUNTER USING T FF

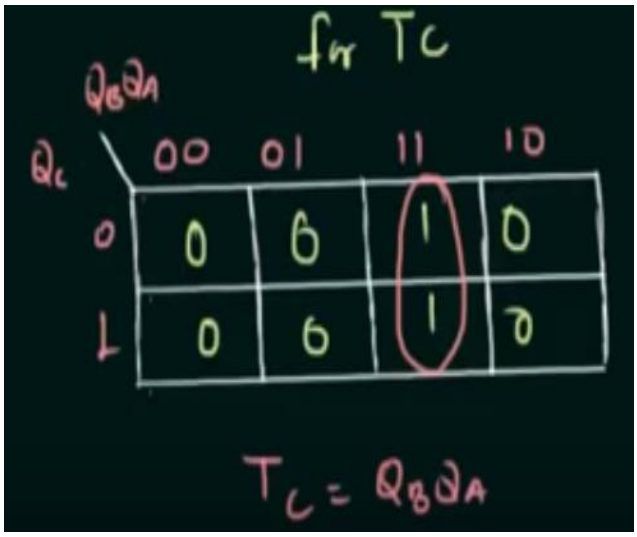


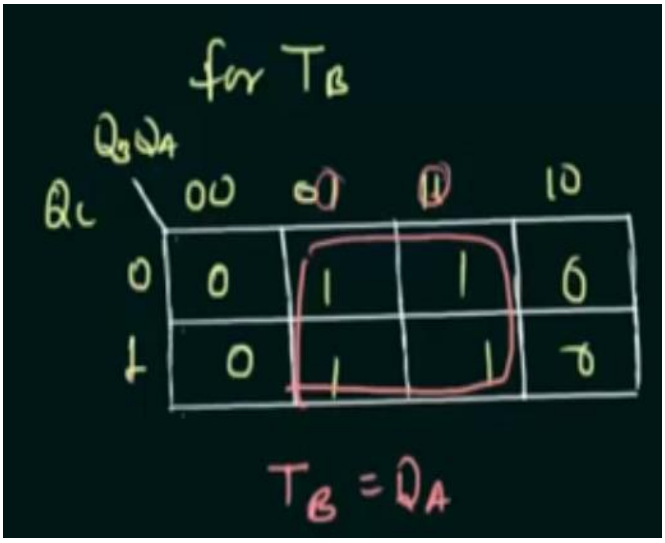
State Diagram



State table along with excitation table

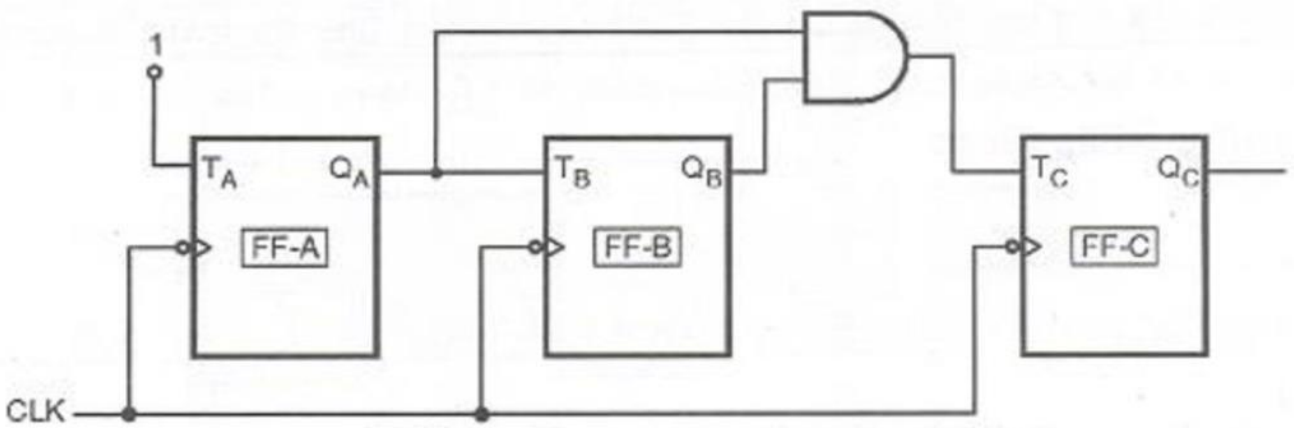
MOD-8 SYNCHRONOUS COUNTER USING T FF





TA=1 AS ALL ARE 1 OR CAN VERIFY THROUGH K-MAPS

MOD-8 SYNCHRONOUS COUNTER USING T FF



3 bit synchronous counter using T-flip flops

MOD-8 SYNCHRONOUS COUNTER USING T FF

- draw the State Diagram with 8 states starting from 000 to 111
- As last number is 111 3 flip-flops are required
- From the State Diagram write the State Table if PS=000 then NS=001
- If PS=001 then NS=010 like this write for every state and if PS=111 then NS=000
- After that write excitation table using PS,NS
- Then obtain Boolean Expressions for T_a, T_b, T_c
- Then draw the logic diagram

See this video if you have doubts
https://www.youtube.com/watch?v=6e8oV2blkGs

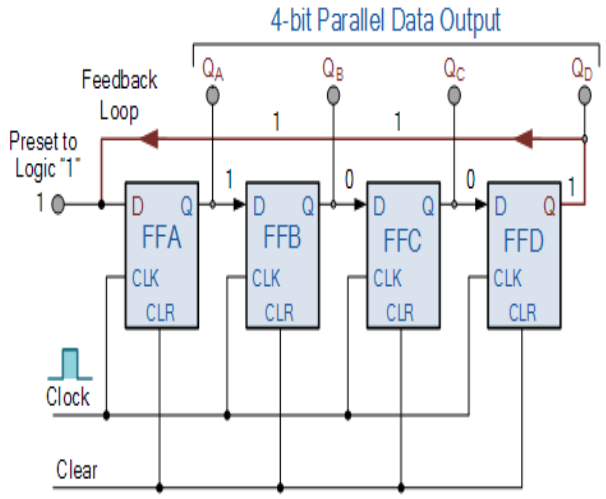
MOD-8 SYNCHRONOUS COUNTER USING T FF

- draw the State Diagram with 8 states starting from 000 to 111
- As last number is 111 3 flip-flops are required
- From the State Diagram write the State Table if PS=000 then NS=001
- If PS=001 then NS=010 like this write for every state and if PS=111 then NS=000
- After that write excitation table using PS,NS
- Then obtain Boolean Expressions for T_a, T_b, T_c
- Then draw the logic diagram

See this video if you have doubts
https://www.youtube.com/watch?v=6e8oV2bIkGs

Design 3 bit up/down Synchronous counter using T Flip-Flops
https://www.youtube.com/watch?v=svFUEJkoeVY

RING COUNTER



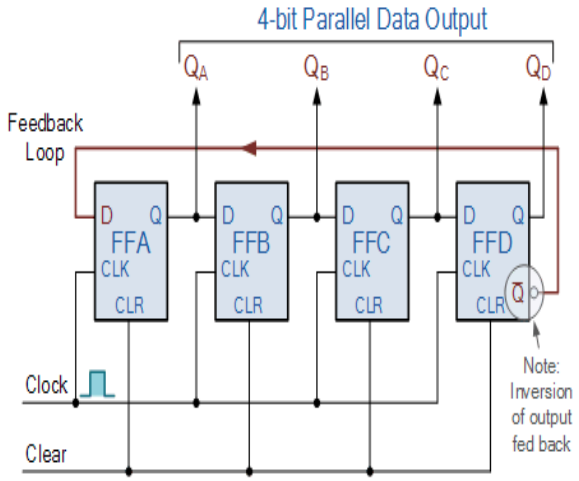
- "CLEAR" signal is firstly applied to all the flip-flops together in order to "RESET" their outputs to a logic "0" level and then a "PRESET" pulse is applied to the input of the first flip-flop (FFA) before the clock pulses are applied. This then places a single logic "1" value into the circuit of the ring counter.
- So on each successive clock pulse, the counter circulates the same data bit between the four flip-flops over and over again around the "ring" every fourth clock cycle.

FUNCTIONAL TABLE OF RING COUNTER

Q_A	Q_B	Q_C	Q_D
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0
AND THE PROCESS CONTINUES FOR EACH CLOCK			

JOHNSON COUNTER OR TWISTED RING COUNTER

In this the inverted output of the last stage flip flop is connected to the input of first flip flop



- "CLEAR" signal is firstly applied to all the flip-flops together in order to "RESET" their outputs to a logic "0" level.
- Then $Q_D'=1$ so at the next clock output will be 1000. As the inverted output Q is connected to the input D this 8-bit pattern continually repeats. For example, "1000", "1100", "1110", "1111", "0111", "0011", "0001", "0000"

FUNCTIONAL TABLE OF JOHNSON COUNTER

Q_A	Q_B	Q_C	Q_D
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
1	0	0	0
1	1	0	0
Sequence continues			

QUESTIONS:

- Design a 4 bit even parity generator
- Convert SR Flip-Flop to JK Flip-Flop
- Explain the working of Universal Shift Register.
- Design a Mod-8 Asynchronous Counter.
- Design a Mod-12 Synchronous Counter.

THANK YOU