Objective:

To design registers and counters.

Syllabus:

Registers and Counters ,Design of registers, bidirectional shift registers, universal shift register, design of ripple counters, synchronous counters and variable modulus counters, ring counter, Johnson counter.

Course Outcomes:

Students will be able to:

- Understand the design of register.
- Understand shift registers and their types.
- Design bidirectional and universal shift register.
- Understand synchronous and asynchronous counters.
- Design Mod Counters.
- Understand the design of ripple, ring and Jhonson counters.

Introduction:

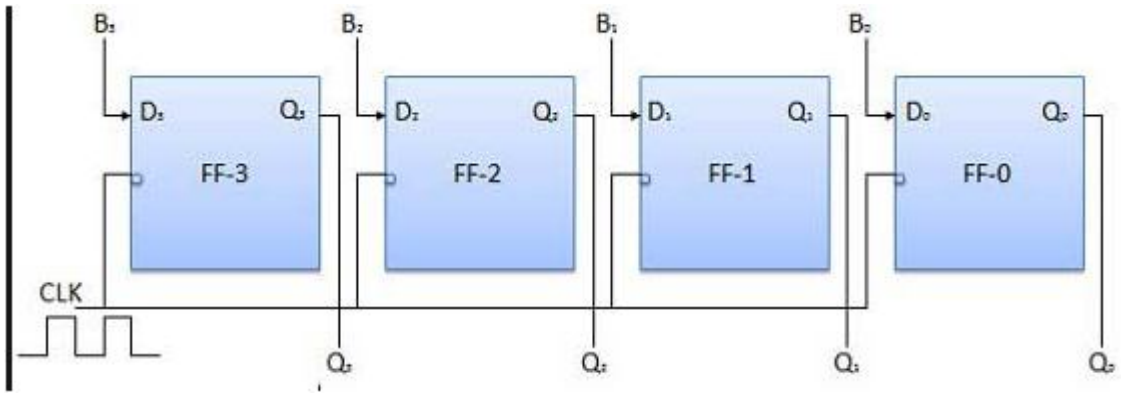
Circuits that include flip-flops are usually classified by the function they perform. Two such circuits are registers and counters:

1. **Register** is a group of flip-flops. Its basic function is to hold information within a digital system so as to make it available to the logic units during the computing process.
2. **Counter** is essentially a register that goes through a predetermined sequence of states.

Registers:

- Registers are groups of flip-flops (FF), where each flip-flop(FF) is capable of storing one bit of information.
- An *n*-bit register is a group of *n* flip-flops.
- The basic function of a register is to hold information in a digital system and make it available to the logic elements for the computing process.
- Registers consist of a finite number of flip-flops. Since each flip-flop is capable of storing either a "0" or a "1", there is a finite number of 0-1 combinations that can be stored into a register. Each of those combinations is known as *state* or *content* of the register.

- With flip-flops we can store data bitwise but usually data does not appear as single bits. Instead it is common to store data words of n bit with typical word lengths of 4, 8, 16, 32 or 64 bit. Thus, several flip-flops are combined to form a register to store whole data words.
- Registers are synchronous circuits thus all flip-flops are controlled by a common clock line.



Shift Registers

- Information often comes bitwise i.e. one bit at every clock pulse. Shift registers are used to store such data.
- A shift register has one serial input. Every clock pulse one bit is loaded from serial in into the first flip-flop of the register while all the actual flip-flop contents are shifted to the next flip-flop, dropping the last bit.
- Shift registers may feature a serial output so that the last bit that gets shifted out of the register can be processed further. It is thus possible to build up a chain of shift registers by connecting each serial out to another shift register's serial in, effectively creating a single big shift register.
- It is also possible to create a Cyclic register by connecting the serial out to the same register's serial in.

- Shift register circuits may also feature additional parallel-in functionality that allows manipulation of individual bits.

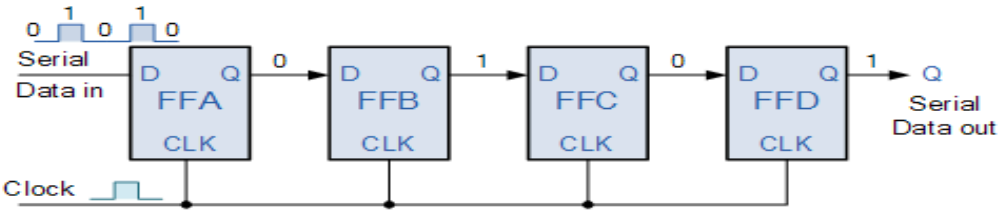
4 types of shift registers

- **Serial-in to Parallel-out (SIPO)** - the register is loaded with serial data, one bit at a time, with the stored data being available at the output in parallel form.
- **Serial-in to Serial-out (SISO)** - the data is shifted serially "IN" and "OUT" of the register, one bit at a time in either a left or right direction under clock control.
- **Parallel-in to Serial-out (PISO)** - the parallel data is loaded into the register simultaneously and is shifted out of the register serially one bit at a time under clock control.
- **Parallel-in to Parallel-out (PIPO)** - the parallel data is loaded simultaneously into the register, and transferred together to their respective outputs by the same clock pulse.

Serial-in to Serial-out (SISO) Shift Register

- The data is allowed to flow straight through the register and out of the other end. Since there is only one output, the DATA leaves the shift register one bit at a time in a serial pattern, hence the name **Serial-in to Serial-Out Shift Register** or **SISO**.
- The SISO shift register is one of the simplest of the four configurations as it has only three connections, the serial input (SI) which determines what enters the left hand flip-flop, the serial output (SO) which is taken from the output of the right hand flip-flop and the sequencing clock signal (Clk).
- The logic circuit diagram below shows a generalized serial-in serial-out shift register:

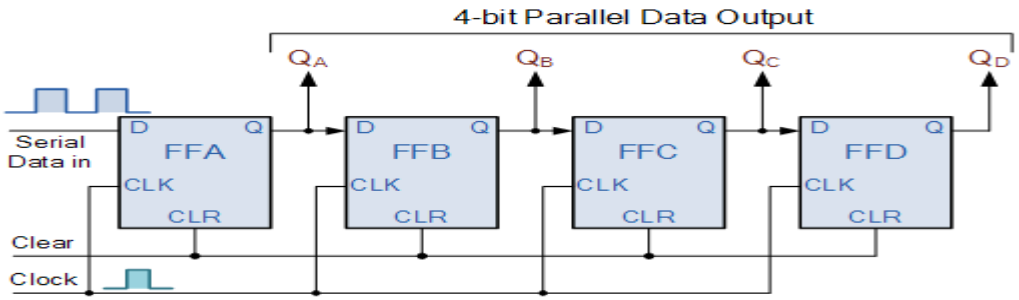
4-bit Serial-in to Serial-out Shift Register:



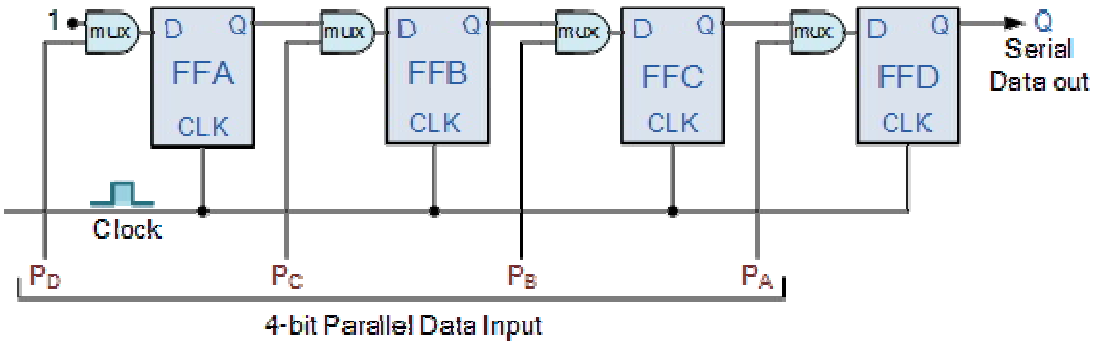
Serial-in to Parallel-out (SIPO) Shift Register

The operation is as follows. Lets assume that all the flip-flops (FFA to FFD) have just been RESET (CLEAR input) and that all the outputs Q_A to Q_D are at logic level "0" ie, no parallel data output.

- If a logic "1" is connected to the DATA input pin of FFA then on the first clock pulse the output of FFA and therefore the resulting Q_A will be set HIGH to logic "1" with all the other outputs still remaining LOW at logic "0". Assume now that the DATA input pin of FFA has returned LOW again to logic "0" giving us one data pulse or 0-1-0.
- The second clock pulse will change the output of FFA to logic "0" and the output of FFB and Q_B HIGH to logic "1" as its input D has the logic "1" level on it from Q_A. The logic "1" has now moved or been "shifted" one place along the register to the right as it is now at Q_A.
- When the third clock pulse arrives this logic "1" value moves to the output of FFC (Q_C) and so on until the arrival of the fifth clock pulse which sets all the outputs Q_A to Q_D back again to logic level "0" because the input to FFA has remained constant at logic level "0".
- The effect of each clock pulse is to shift the data contents of each stage one place to the right, and this is shown in the following table until the complete data value of 0-0-0-1is stored in the register. This data value can now be read directly from the outputs of $Q_A$to Q_D.



Parallel-in to Serial-out Shift Register

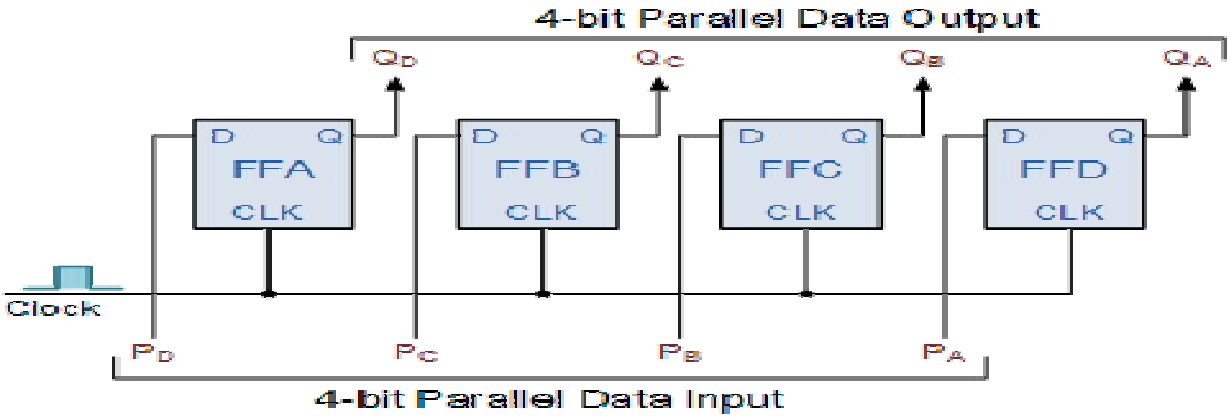


As this type of shift register converts parallel data, such as an 8-bit data word into serial format, it can be used to multiplex many different input lines into a single serial DATA stream which can be sent directly to a computer or transmitted over a communications line.

Parallel-in to Parallel-out (PIPO) Shift Register

- The final mode of operation is the Parallel-in to Parallel-out Shift Register.
- This type of shift register also acts as a temporary storage device or as a time delay device similar to the SISO configuration above.
- The data is presented in a parallel format to the parallel input pins P_A to P_D and then transferred together directly to their respective output pinsQ_A to Q_A by the same clock pulse.

- Then one clock pulse loads and unloads the register.
- This arrangement for parallel loading and unloading is shown below.
- The PIPO shift register is the simplest of the four configurations as it has only three connections, the parallel input (PI) which determines what enters the flip-flop, the parallel output (PO) and the sequencing clock signal (Clk).
- Similar to the Serial-in to Serial-out shift register, this type of register also acts as a temporary storage device or as a time delay device, with the amount of time delay being varied by the frequency of the clock pulses. Also, in this type of register there are no interconnections between the individual flip-flops since no serial shifting of the data is required.

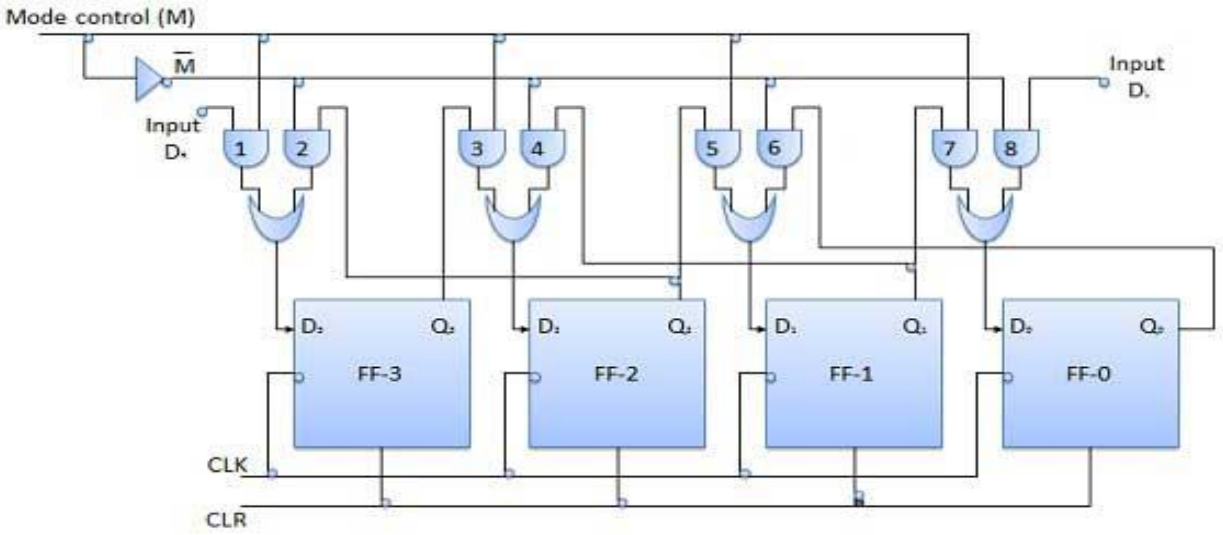


Bidirectional Shift Register

- If a binary number is shifted left by one position then it is equivalent to multiplying the original number by 2. Similarly if a binary number is shifted right by one position then it is equivalent to dividing the original number by 2.
- Hence if we want to use the shift register to multiply and divide the given binary number, then we should be able to move the data in either left or right direction.
- Such a register is called bi-directional register. A four bit bi-directional shift register is shown in fig.

- There are two serial inputs namely the serial right shift data input DR, and the serial left shift data input DL along with a mode select input (M).

Block Diagram



Operation

S.N.	Condition	Operation
1	**With M = 1 − Shift right operation**	If M = 1, then the AND gates 1, 3, 5 and 7 are enabled whereas the remaining AND gates 2, 4, 6 and 8 will be disabled. The data at D_R is shifted to right bit by bit from FF-3 to FF-0 on the application of clock pulses. Thus with M = 1 we get the serial right shift operation.
2	**With M = 0 − Shift left operation**	When the mode control M is connected to 0 then the AND gates 2, 4, 6 and 8 are

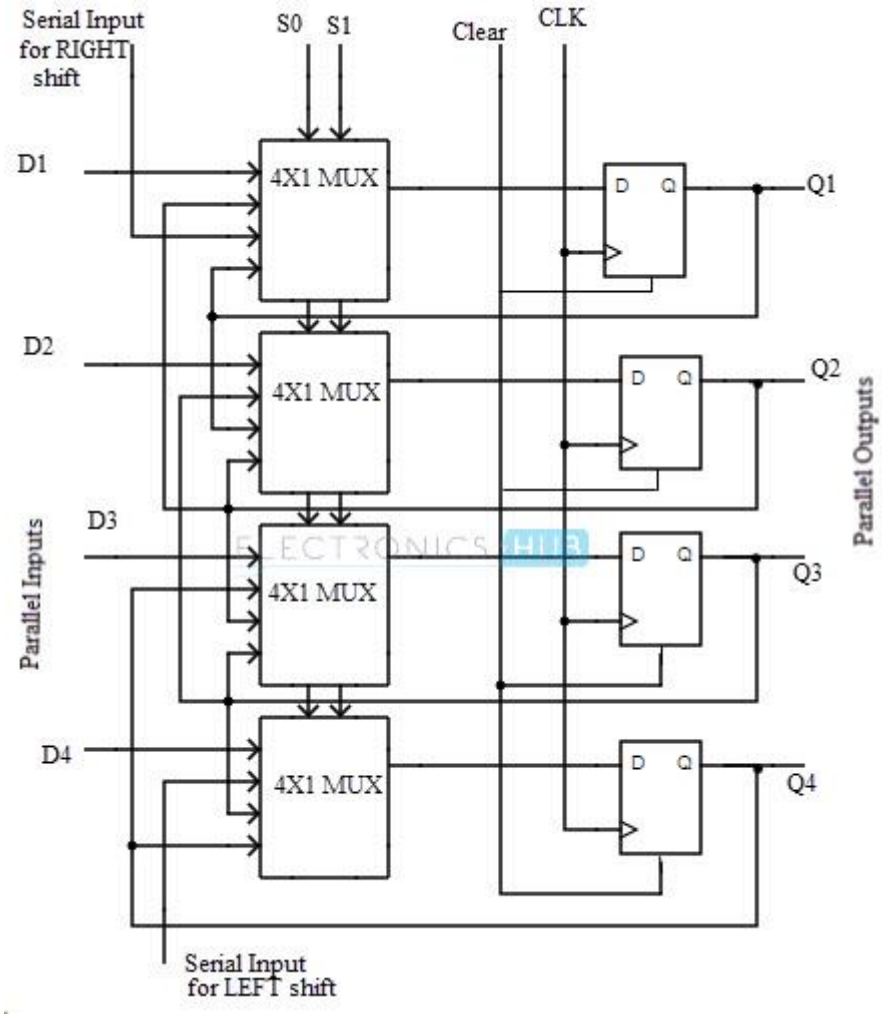
		enabled while 1, 3, 5 and 7 are disabled. The data at D_L is shifted left bit by bit from FF-0 to FF-3 on the application of clock pulses. Thus with M = 0 we get the serial right shift operation.

Universal Shift Register

- A shift register which can shift the data in only one direction is called a uni-directional shift register.
- A shift register which can shift the data in both directions is called a bi-directional shift register.
- Applying the same logic, a shift register which can shift the data in both directions as well as load it parallely, is known as a universal shift register.
- The shift register is capable of performing the following operation −
- Parallel loading
- Lift shifting
- Right shifting

The mode control input is connected to logic 1 for parallel loading operation whereas it is connected to 0 for serial shifting. With mode control pin connected to ground, the universal shift register acts as a bi-directional register. For serial left operation, the input is applied to the serial input which goes to AND gate-1 shown in figure. Whereas for the shift right operation, the serial input is applied to D input.

Block Diagram



Counters

Counter is a sequential circuit. A digital circuit which is used for a counting pulses is known counter. Counter is the widest application of flip-flops. It is a group of flip-flops with a clock signal applied. Counters are of two types.

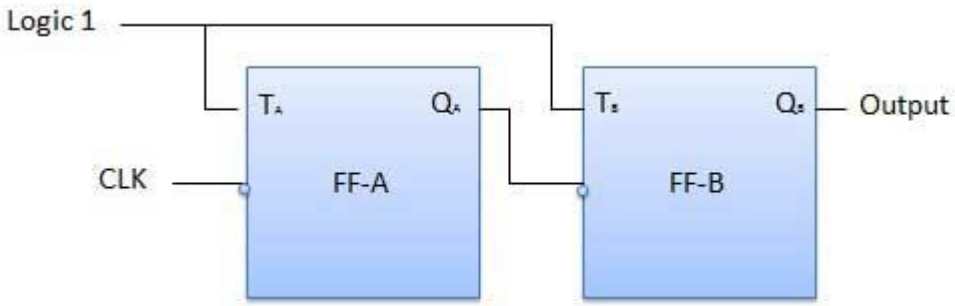
- Asynchronous or ripple counters.

- Synchronous counters.

Asynchronous or ripple counters

The logic diagram of a 2-bit ripple up counter is shown in figure. The toggle (T) flip-flop are being used. But we can use the JK flip-flop also with J and K connected permanently to logic 1. External clock is applied to the clock input of flip-flop A and Q_A output is applied to the clock input of the next flip-flop i.e. FF-B.

Logical Diagram



Operation

S.N.	Condition	Operation
1	**Initially let both the FFs be in the reset state**	Q_BQ_A = 00 initially
2	**After 1st negative clock edge**	As soon as the first negative clock edge is applied, FF-A will toggle and Q_A will be equal to 1. Q_A is connected to clock input of FF-B. Since Q_A has changed from 0 to 1, it is treated as the positive clock edge by FF-B. There is no

		change in $Q_B$because FF-B is a negative edge triggered FF. Q_BQ_A = 01 after the first clock pulse.
3	**After 2nd negative clock edge**	On the arrival of second negative clock edge, FF-A toggles again and Q_A = 0. The change in Q_A acts as a negative clock edge for FF-B. So it will also toggle, and $Q_B$will be 1. Q_BQ_A = 10 after the second clock pulse.
4	**After 3rd negative clock edge**	On the arrival of 3rd negative clock edge, FF-A toggles again and Q_A become 1 from 0. Since this is a positive going change, FF-B does not respond to it and remains inactive. So Q_B does not change and continues to be equal to 1. Q_BQ_A = 11 after the third clock pulse.

5	**After 4th negative clock edge**	On the arrival of 4th negative clock edge, FF-A toggles again and Q_A becomes 1 from 0. This negative change in $Q_A$acts as clock pulse for FF-B. Hence it toggles to change $Q_B$from 1 to 0. Q_BQ_A = 00 after the fourth clock pulse.

Truth Table

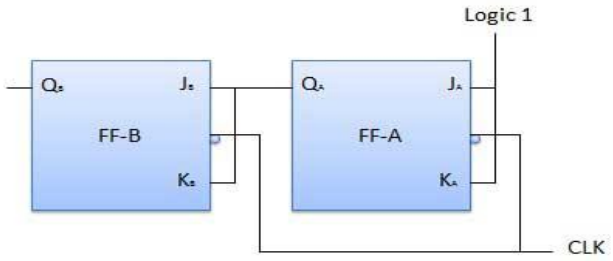
Clock	Counter output		State number	Deciimal Counter output
	Q_B	Q_A		
Initially	0	0	—	0
1st	0	1	1	1
2nd	1	0	2	2
3rd	1	1	3	3
4th	0	0	4	0

Synchronous counters

If the "clock" pulses are applied to all the flip-flops in a counter simultaneously, then such a counter is called as synchronous counter.

2-bit Synchronous up counter

The J_A and K_A inputs of FF-A are tied to logic 1. So FF-A will work as a toggle flip-flop. The J_B and K_B inputs are connected to Q_A.



Operation

S.N.	Condition	Operation
1	**Initially let both the FFs be in the reset state**	Q_BQ_A = 00 initially.
2	**After 1st negative clock edge**	As soon as the first negative clock edge is applied, FF-A will toggle and Q_A will change from 0 to 1. But at the instant of application of negative clock edge, Q_A , J_B = K_B = 0. Hence FF-B will not change its state. So Q_B will remain 0. Q_BQ_A = 01 after the first clock pulse.
3	**After 2nd negative clock edge**	On the arrival of second negative clock edge, FF-A toggles again and Q_A changes from 1 to 0. But at this instant Q_A was 1. So J_B = K_B= 1 and FF-B will toggle.

		Hence Q_B changes from 0 to 1. Q_BQ_A = 10 after the second clock pulse.
4	**After 3rd negative clock edge**	On application of the third falling clock edge, FF-A will toggle from 0 to 1 but there is no change of state for FF-B. Q_BQ_A = 11 after the third clock pulse.
5	**After 4th negative clock edge**	On application of the next clock pulse, Q_A will change from 1 to 0 as Q_B will also change from 1 to 0. Q_BQ_A = 00 after the fourth clock pulse.

Classification of counters

Depending on the way in which the counting progresses, the synchronous or asynchronous counters are classified as follows −

- Up counters
- Down counters
- Up/Down counters

UP/DOWN Counter

Up counter and down counter is combined together to obtain an UP/DOWN counter. A mode control (M) input is also provided to select either up or down mode. A combinational circuit is required to be designed and used between each pair of flip-flop in order to achieve the up/down operation.

- Type of up/down counters
- UP/DOWN ripple counters
- UP/DOWN synchronous counter

UP/DOWN Ripple Counters

In the UP/DOWN ripple counter all the FFs operate in the toggle mode. So either T flip-flops or JK flip-flops are to be used. The LSB flip-flop receives clock directly. But the clock to every other FF is obtained from (Q = Q bar) output of the previous FF.

- **UP counting mode (M=0)** − The Q output of the preceding FF is connected to the clock of the next stage if up counting is to be achieved. For this mode, the mode select input M is at logic 0 (M=0).
- **DOWN counting mode (M=1)** − If M = 1, then the Q bar output of the preceding FF is connected to the next FF. This will operate the counter in the counting mode.

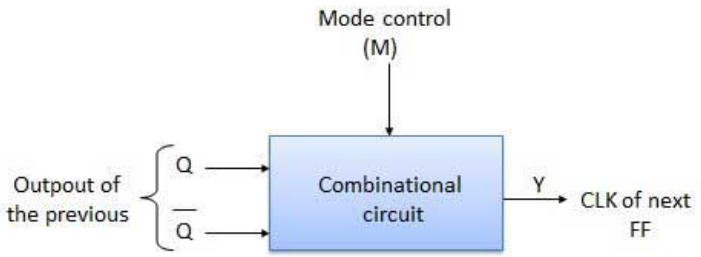
Example

3-bit binary up/down ripple counter.

- 3-bit − hence three FFs are required.
- UP/DOWN − So a mode control input is essential.
- For a ripple up counter, the Q output of preceding FF is connected to the clock input of the next one.
- For a ripple up counter, the Q output of preceding FF is connected to the clock input of the next one.

- For a ripple down counter, the Q bar output of preceding FF is connected to the clock input of the next one.
- Let the selection of Q and Q bar output of the preceding FF be controlled by the mode control input M such that, If M = 0, UP counting. So connect Q to CLK. If M = 1, DOWN counting. So connect Q bar to CLK.

Block Diagram



Truth Table

Inputs			Outputs	
M	Q	$\overline{Q}$	Y	
0	0	0	0	Y = Q for up counter
0	0	1	0	
0	1	0	1	
0	1	1	1	
1	0	0	0	$Y = \overline{Q}$ for up counter
1	0	1	1	
1	1	0	0	
1	1	1	1	

Operation

S.N.	Condition	Operation
1	**Case 1 − With M = 0 (Up counting mode)**	If M = 0 and M bar = 1, then the AND gates 1 and 3 in fig. will be enabled whereas the AND gates 2 and 4 will be disabled. Hence Q_A gets connected to the clock input of FF-B and $Q_B$gets connected to the clock input of FF-C. These connections are same as those for the normal up counter. Thus with M = 0 the circuit work as an up counter.
2	**Case 2: With M = 1 (Down counting mode)**	If M = 1, then AND gates 2 and 4 in fig. are enabled whereas the AND gates 1 and 3 are disabled. Hence Q_A bar gets connected to the clock input of FF-B and Q_B bar gets connected to the clock input of FF-C. These connections will produce a down counter. Thus with M = 1 the circuit works as a down counter.

Modulus Counter (MOD-N Counter)

The 2-bit ripple counter is called as MOD-4 counter and 3-bit ripple counter is called as MOD-8 counter. So in general, an n-bit ripple counter is called as modulo-N counter. Where, MOD number = 2^n.

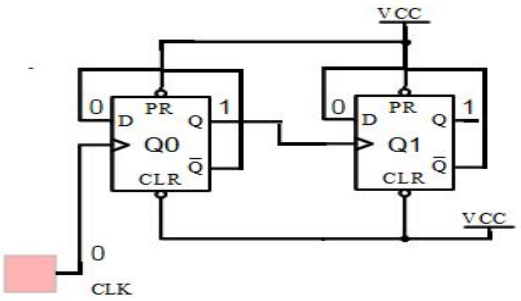
Type of modulus

- 2-bit up or down (MOD-4)
- 3-bit up or down (MOD-8)
- 4-bit up or down (MOD-16)

Ripple Counter

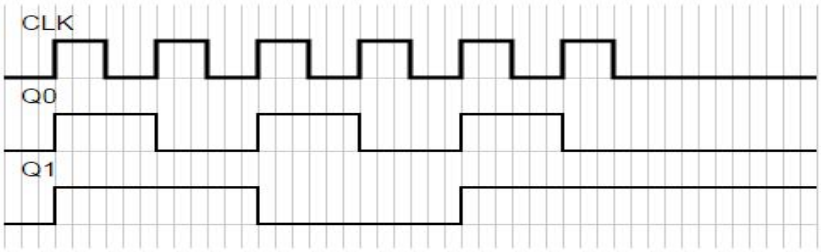
- A ripple counter is an asynchronous counter where only the first flip-flop is clocked by an external clock. All subsequent flip-flops are clocked by the output of the preceding flip-flop.
- Asynchronous counters are also called ripple-counters because of the way the clock pulse ripples it way through the flip-flops.
- The MOD of the ripple counter or asynchronous counter is 2^n if n flip-flops are used.
- For a 4-bit counter, the range of the count is 0000 to 1111 (2^4-1). A counter may count up or count down or count up and down depending on the input control. The count sequence usually repeats itself. When counting up, the count sequence goes from 0000, 0001, 0010, ... 1110 , 1111 , 0000, 0001, ... etc. When counting down the count sequence goes in the opposite manner: 1111, 1110, ... 0010, 0001, 0000, 1111, 1110, ... etc.
- The complement of the count sequence counts in reverse direction. If the uncomplemented output counts up, the complemented output counts down. If the uncomplemented output counts down, the complemented output counts up.

- Asynchronous counters are slower than synchronous counters because of the delay in the transmission of the pulses from flip-flop to flip-flop. With a synchronous circuit, all the bits in the count change synchronously with the assertion of the clock.
- Examples of synchronous counters are the **Ring and Johnson counter**.The circuit below uses 2 D flip-flops to implement a divide-by-4 ripple counter ($2^n = 2^2 = 4$). It counts down.



STATE TABLE

COUNT	Q1	Q0
3	1	1
2	1	0
1	0	1
0	0	0



Ring Counter

- A **ring counter** is a type of counter composed of a type of circular shift register. The output of the last shift register is fed to the input of the first register.
- The Hamming distanceof an Overbeck counter is 2, the Hamming distance of a Johnson counter is 1.
- There are two types of ring counters:
  - A **straight ring counter** or **Overbeck counter** (most probably named after Wilcox Pratt Overbeck connects the output of the last shift register to the first shift register

input and circulates a single one (or zero) bit around the ring. For example, in a 4-register one-hot counter, with initial register values of 1000, the repeating pattern is: 1000, 0100, 0010, 0001, 1000... . Note that one of the registers must be pre-loaded with a 1 (or 0) in order to operate properly.

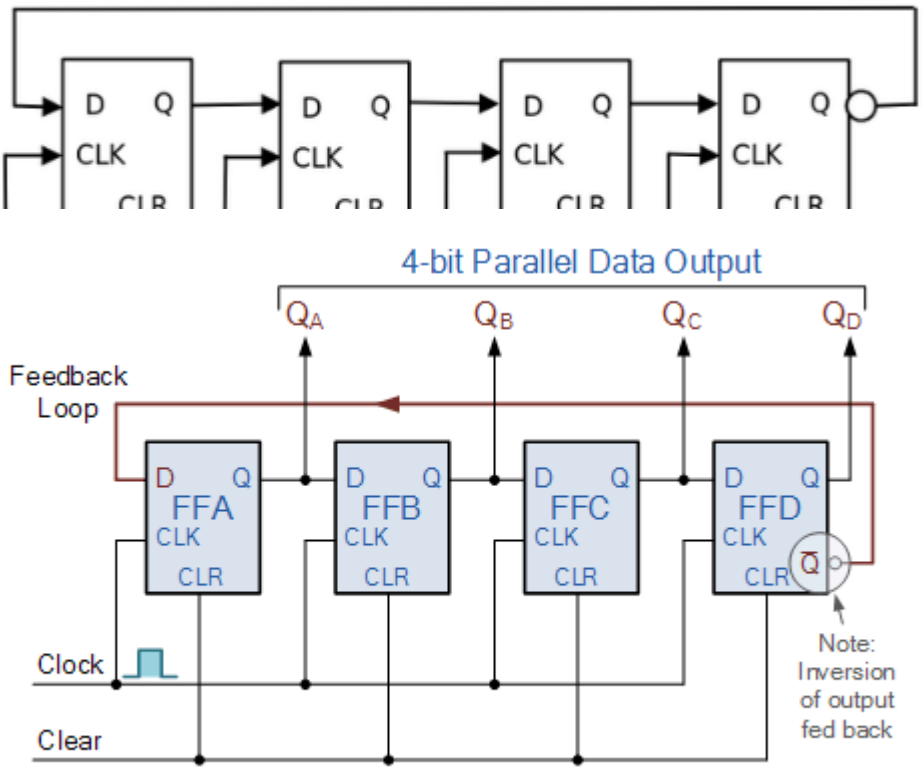
- A **twisted ring counter**, also called **switch-tail ring counter**, **walking ring counter**, **Johnson counter** (from using Libaw-Craig code (de) aka Johnson code (de), named after Robert Royce Johnson - the patent holder) or **Möbius counter** (from its abstract similarities with Möbius bands, named after August Ferdinand Möbius), connects the complement of the output of the last shift register to the input of the first register and circulates a stream of ones followed by zeros around the ring. For example, in a 4-register counter, with initial register values of 0000, the repeating pattern is: 0000, 1000, 1100, 1110, 1111, 0111, 0011, 0001, 0000...

Straight ring/*Overbeck counter*					Twisted ring/*Johnson counter*				
State	Q0	Q1	Q2	Q3	State	Q0	Q1	Q2	Q3
0	1	0	0	0	0	0	0	0	0
1	0	1	0	0	1	1	0	0	0
2	0	0	1	0	2	1	1	0	0
3	0	0	0	1	3	1	1	1	0
0	1	0	0	0	4	1	1	1	1
1	0	1	0	0	5	0	1	1	1
2	0	0	1	0	6	0	0	1	1
3	0	0	0	1	7	0	0	0	1
0	1	0	0	0	*0*	0	0	0	0

Jhonson Counter

- Johnson counters are often favoured, not just because they offer twice as many count states from the same number of shift registers, but because they are able to self-initialise from the all-zeros state, without requiring the first count bit to be injected externally at start-up.

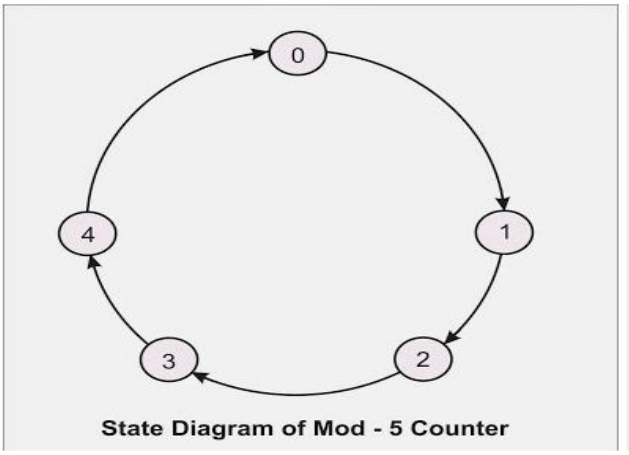
- The Johnson counter generates a Gray code, a code in which adjacent states differ by only one bit.
- A Johnson counter is a ring with an inversion.



Design of mod-5 synchronous counter

- The number of flip-flops required for mod-5 counter is three. This is found from the equation $2^m \geq n \geq 2^{m-1}$ where n = 5. Let us assume that the mod-5 counter has 5 states.
- Step 1 : State Diagram : The state diagram for the mod-5 counter can be drawn as shown in the below figure. Here, it is assumed that the state transition from one state to another takes place when the clock pulse is applied.

- Step 2 : State Table : From the above state diagram, one can draw PS = NS table as shown in Table



State Diagram of Mod - 5 Counter

PS NS table for Mod-5 Counter

Present State (PS)	Next State (NS)
0	1
1	2
2	3
3	4
4	0

The above state table does not have any redundant state because no two states are equivalent. So there is no modification in the above state table.

Step 3 : Sate Assignment : let us assign there state variables to state 0, 1, 2, 3, 4, as follows 0 : 000, 1 = 001 2 : 010 3 : 011 4 : 100 then PS – NS table gets modified as shown in table.

PS – NS table for Mod-5 Counter in binary

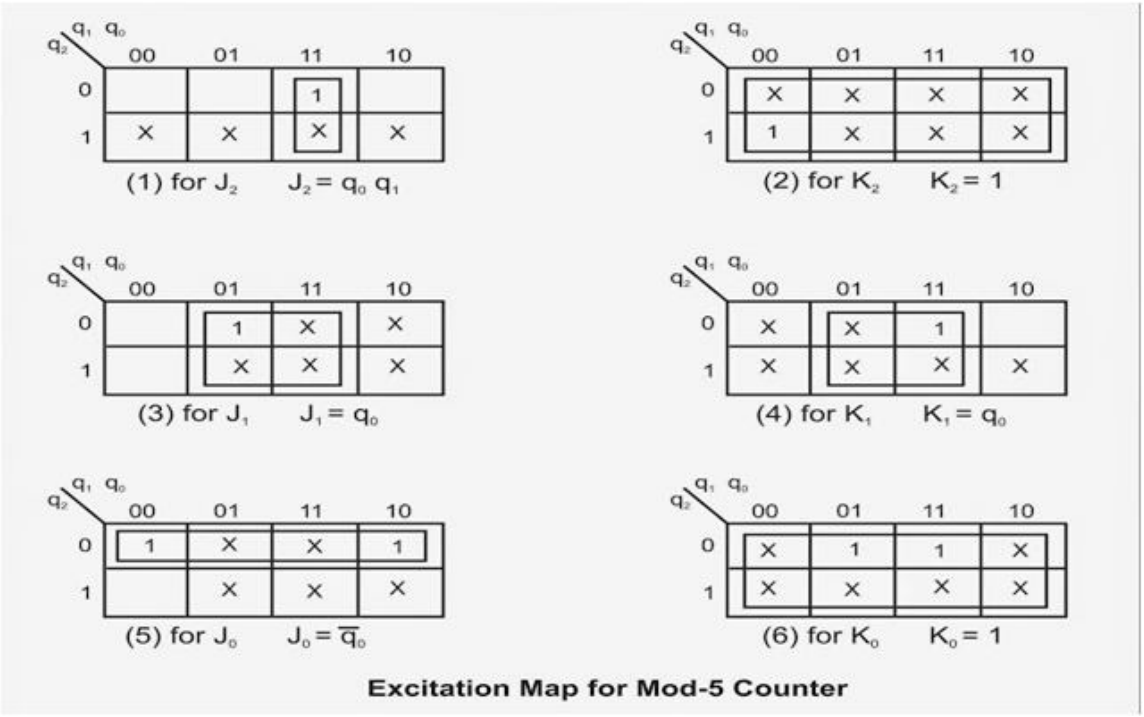
Present State (PS)			Next State (NS)		
q_2	q_1	q_0	Q_2	Q_1	Q_0
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0
-----	-----	-----	-----	-----	-----
1	0	1	X	X	X
1	1	0	X	X	X
1	1	1	X	X	X

X : Don't care whether it may be 0 or 1

Step 4 : Excitation Table : The JK flip-flop is selected for the counter design because it result in simplified circuit. The excitation table having entries for flip-flop inputs (J2, K2, J1 K1 J0 K0) can drawn from above PS-NS table and using the application table of JK flip-flop. The below table gives the excitation values of mod-5 – Counter.

Excitation Table for Mod-5 Counter

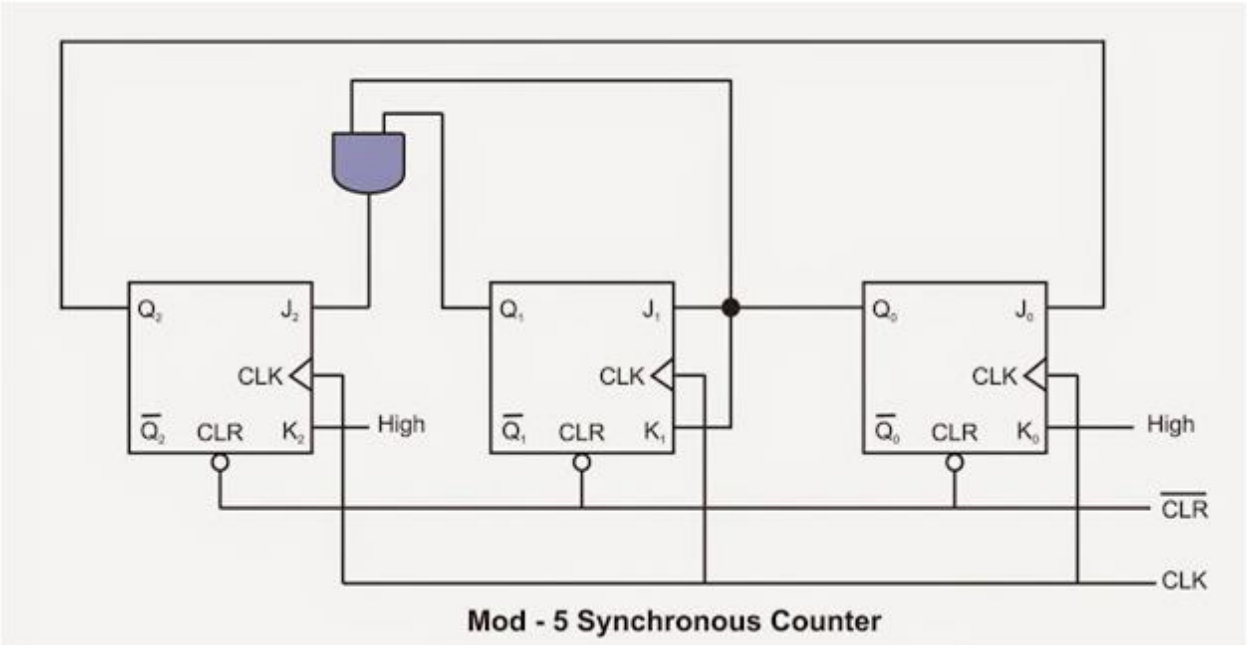
PS			NS			Excitation Inputs					
q_2	q_1	q_0	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	x	1
1	0	0	0	0	0	x	1	0	x	0	x
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
1	0	1	X	X	X	X	X	X	X	X	X
1	1	0	X	X	X	X	X	X	X	X	X
1	1	1	X	X	X	X	X	X	X	X	X



Excitation Map for Mod-5 Counter

Step 5 : Excitation Maps : The excitation maps for J2, K2 J1, K1, J0 and K0 inputs of the counter can be drawn in below figure the excitation table.

Step 6 : Schematic Diagram : Using the above equations, the circuit diagram for the mod-5



Mod - 5 Synchronous Counter

UNIT-VI
Assignment-Cum-Tutorial Questions
SECTION-A

Objective Questions

1. A group of binary cells is called [ ]
   A. counter B. register C. latch D. flip-flop
2. Simplest registers only consists of [ ]
   A. Counter B.EPROM C. latch D. flipflop
3. What type of register would shift a complete binary number in one bit at a time and shift all the stored bits out one bit at a time? [ ]
   A. SISO B. PIPO C.SIPO D.PISO
4. What is a shift register that will accept a parallel input and can shift data left or right called? [ ]
   A. end around B. conversion C. bidirectional universal D. tri-state
5. How many flip-flops are required to make a MOD-32 binary counter? [ ]
   A. 3 B. 45 C. 5 D. 6
6. A MOD-16 ripple counter is holding the count 1001_2. What will the count be after 31 clock pulses? [ ]
   A. 1000_2 B. 1010_2 C. 1011_2 D. 1101_2
7. By default counters are incremented by [ ]
   A. 1 B. 2 C. 3 D. 4
8. Three decade counter would have [ ]
   A. 2 BCD counters B. 3 BCD counters C. 4 BCD counters D. 5 BCD counters
9. A decimal counter has [ ]
   A. 5 states B. 10 states C.15 states D.20 states
10. Ripple counters are also called [ ]
    A. SSI counters B. asynchronous counters
    C. synchronous counters D. VLSI counters
11. Binary counter that count incrementally and decrementally is called [ ]

A. up-down counter B. LSI counters C. down counter D. up counter

12. A ripple counter's speed is limited by the propagation delay of: [ ]

A. all flip-flops and gates B. each flip-flop

C. the flip-flops only with gates D. only circuit gates

13. When two counters are cascaded, the overall MOD number is equal to the _____ of their individual MOD numbers.

A. product B.sum C. log D. reciprocal [ ]

14. A BCD counter is a _____. [ ]

A. binary counter

B. full-modulus counter

C. decade counter

D. divide-by-10 counter

15. How many flip-flops are required to construct a decade counter? [ ]

A.10 B.8 C.5 D.4

16. How many different states does a 3-bit asynchronous counter have? [ ]

A.2 B.4 C.8 D.16

17. One of the major drawbacks to the use of asynchronous counters is: [ ]

A. low-frequency applications are limited because of internal propagation delays

B. high-frequency applications are limited because of internal propagation delays

C. asynchronous counters do not have major drawbacks and are suitable for use in high- and low-frequency counting applications

D. asynchronous counters do not have propagation delays and this limits their use in high-frequency applications

18. Once an up-/down-counter begins its count sequence, it cannot be reversed.

A. True [ ]

B. False

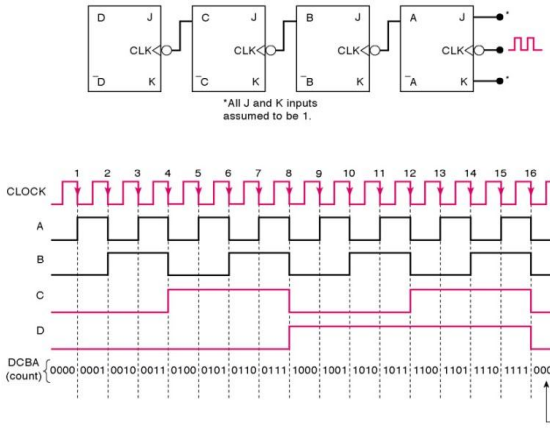
19. Three cascaded modulus-5 counters have an overall modulus of [ ]

A. 5 B. 25 C. 125 D. 500

SECTION-B

Descriptive Questions

1. Explain shift register.
2. Explain bi-directional shift register.
3. Explain universal shift register.
4. Explain synchronous counters.
5. Explain Asynchronous counters.
6. Explain synchronous UP/DOWN counter.
7. Design a synchronous BCD counter with T flip-flops
8. What would be the MOD number of the counter if three more FFs were added?
9.



The counter in the above Figure starts off in the 0000 state, and then clock pulses are applied. Some time later the clock pulses are removed, and the counter FFs read 0011. How many clock pulses have occurred?

10. Construct an appropriate MOD-10 counter

Section C:

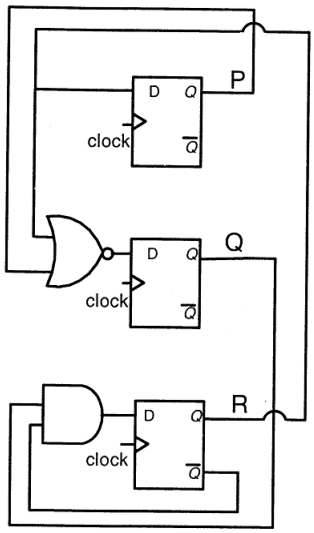
Questions asked in GATE

1. We want to design a synchronous counter that counts the sequence 0-1-0-2-0-3 and then repeats. The minimum number of J-K flip-flops required to implement this counter is

[GATE 2016]

(A) 1 **(B)** 2 **(C)** 4 **(D)** 5

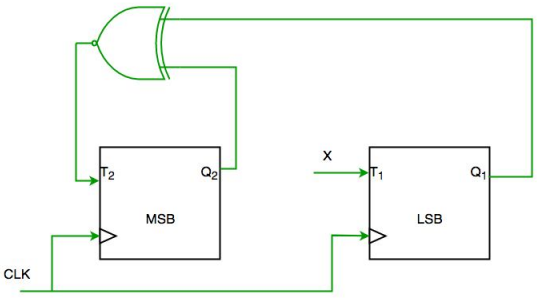
2. Consider the following circuit involving three D-type flip-flops used in a certain type of counter configuration. **[GATE 2011]**



3. If at some instance prior to the occurrence of the clock edge, P, Q and R have a value 0, 1 and 0 respectively, what shall be the value of PQR after the clock edge? [ ]

(A) 000 **(B)** 001 **(C)** 010 **(D)** 011

4. Consider the partial implementation of a 2-bitt counter using T flip-flops following the sequence 0-2-3-1-0, as shown below **[GATE 2004]**



To complete the circuit, the input X should be [ ]

(A) Q_2' (B) Q_2+Q_1 (C) $(Q_1 \oplus Q_2)'$ (D) $Q_1 \oplus Q_2$