Logic Gates - V

Comprehensive Course on Digital Logic Design 2023/2024

B V Reddy • Lesson 35 • Dec 8, 2022

SEQUENTIAL CIRCUITS

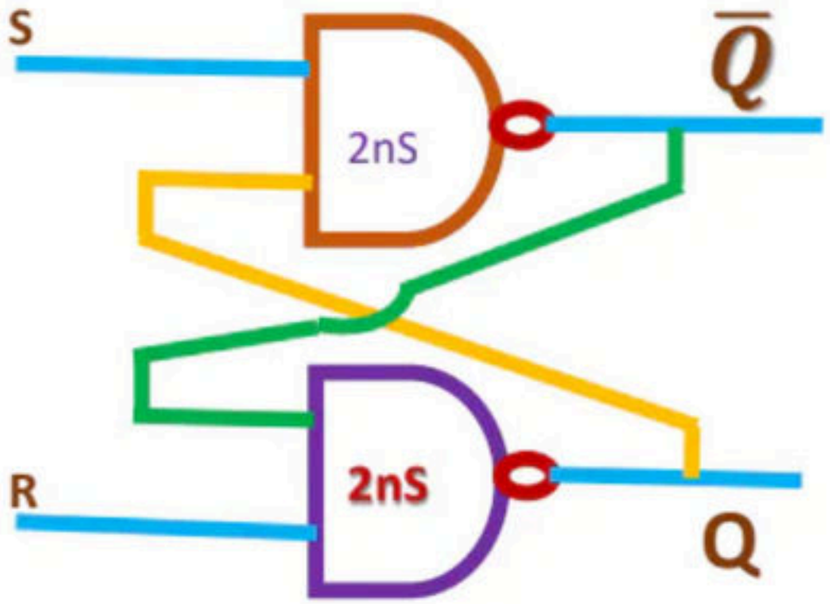
The logic circuit whose outputs at any instant of time depends on the present inputs as well as on the past outputs are called as sequential circuits, in sequential circuits , the output signals are feedback to the input side .

Sequential Circuits

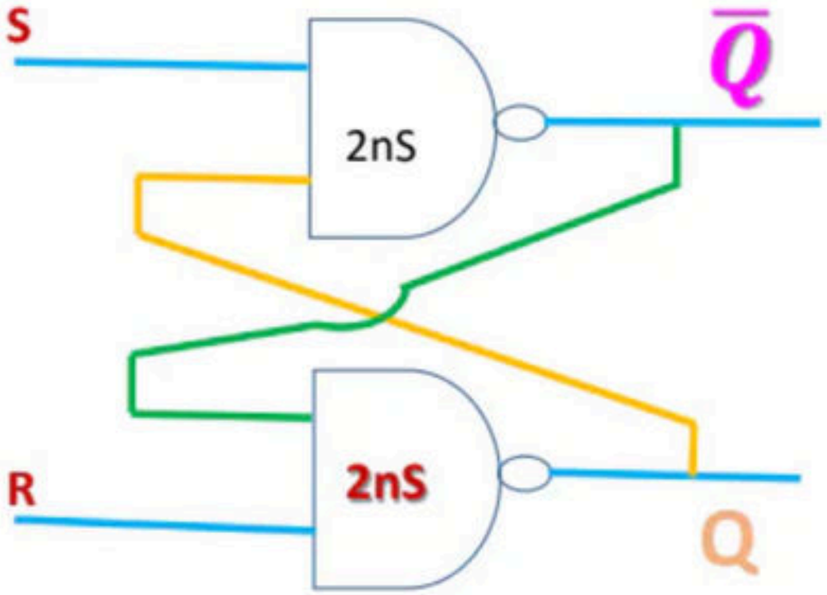
- Latch
- Flip flops
- Shift Register
- Counters
- Finite State Machines

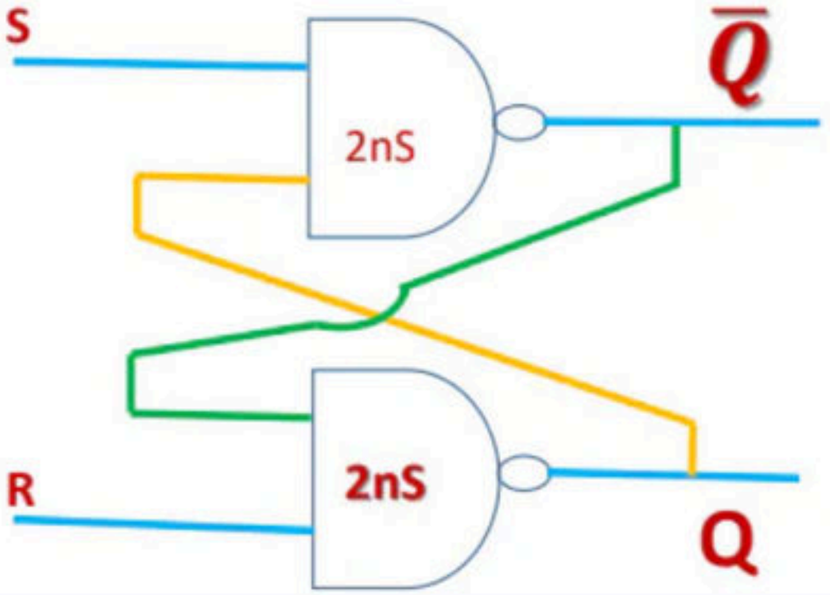
Latch

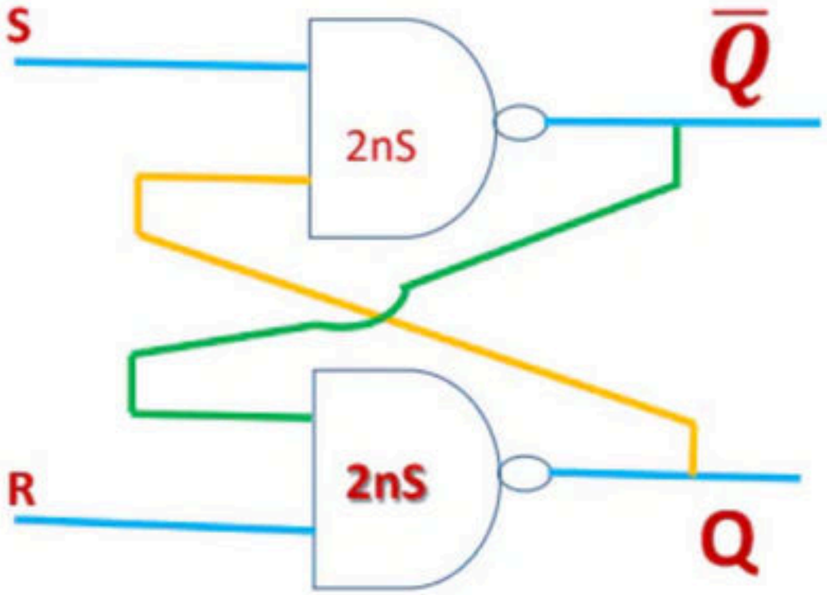
NAND Latch

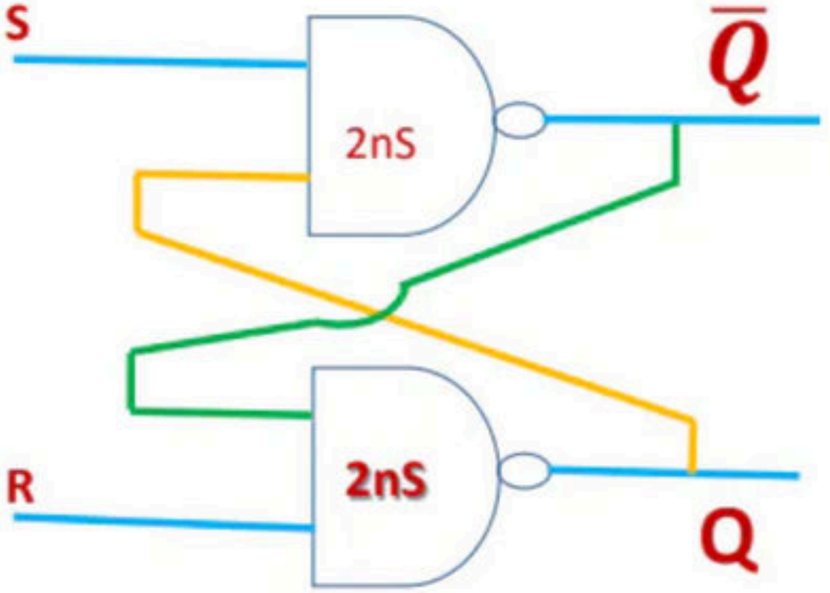


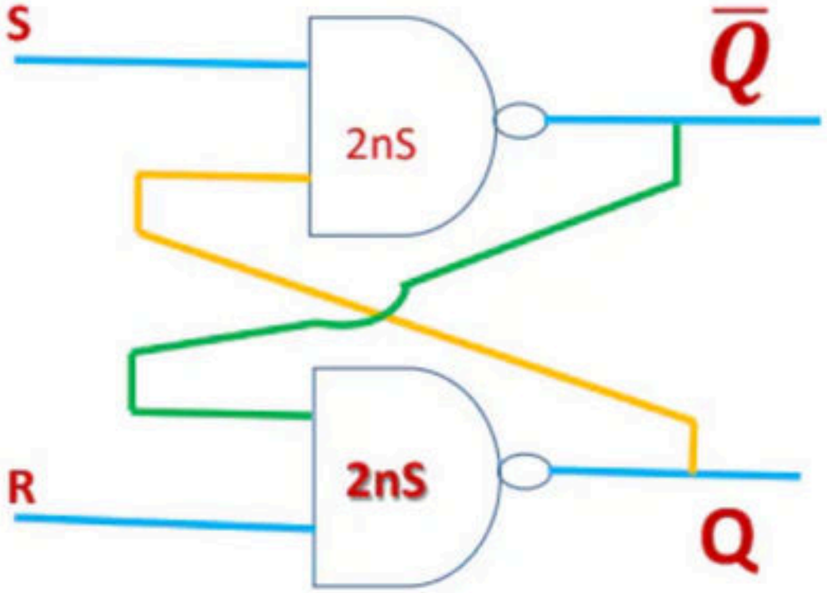
S	R	Q_n	Q_{n+1}	$\overline{Q_{n+1}}$

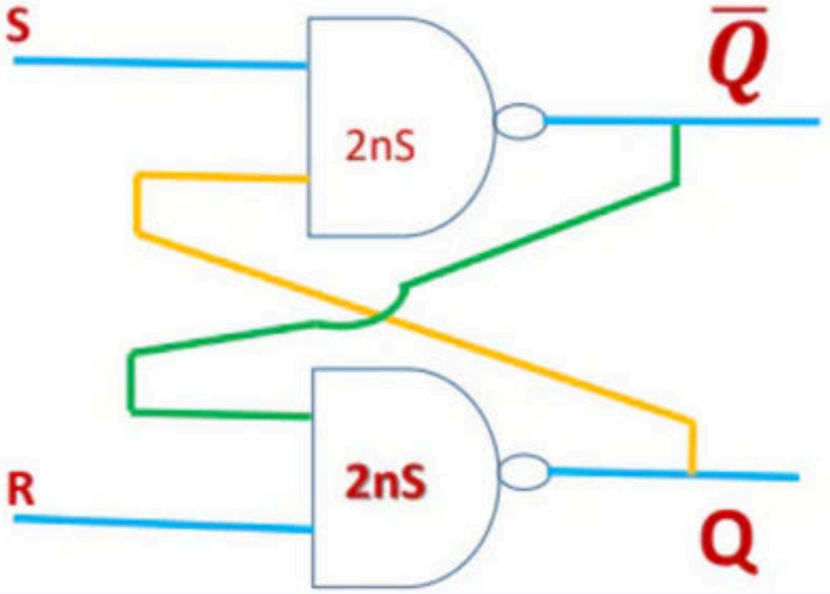

S
$\overline{Q}$
2nS
2nS
R
Q

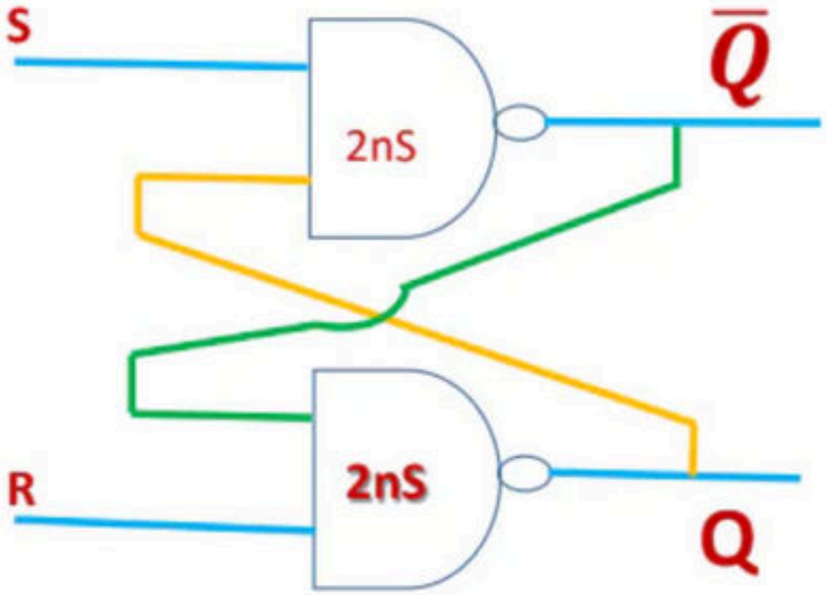

S
$\overline{Q}$
2nS
2nS
R
Q

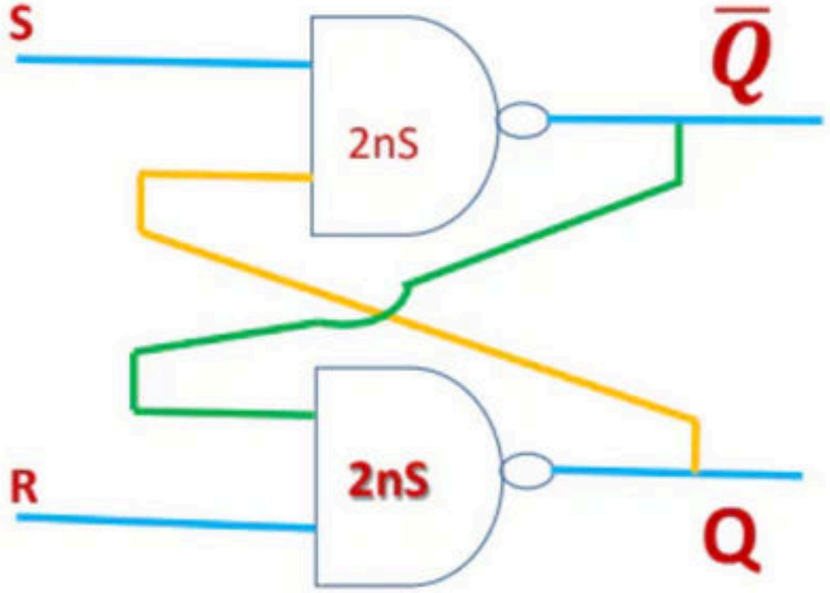

S
$\overline{Q}$
2nS
2nS
R
Q


S
$\overline{Q}$
2nS
2nS
R
Q


S
$\overline{Q}$
2nS
2nS
R
Q

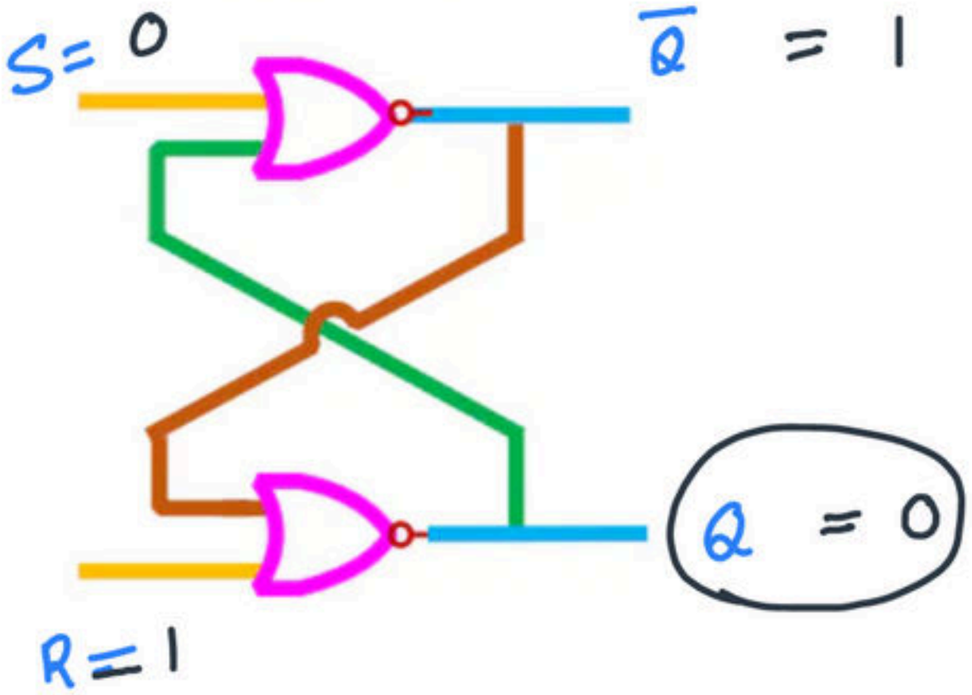

S
$\overline{Q}$
2nS
2nS
R
Q


S
$\overline{Q}$
2nS
2nS
R
Q

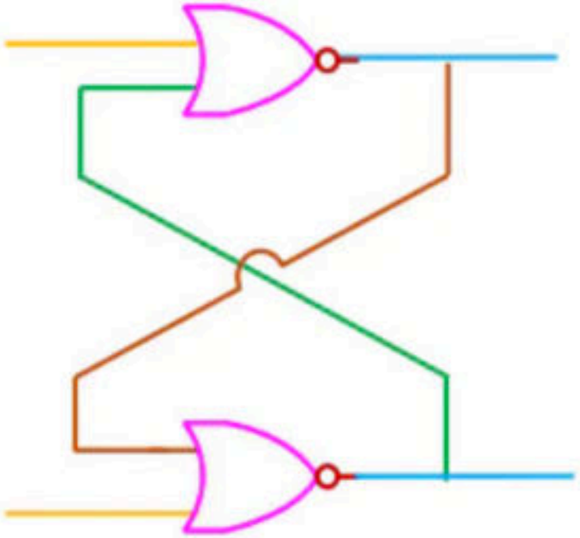

S
$\overline{Q}$
2nS
2nS
R
Q

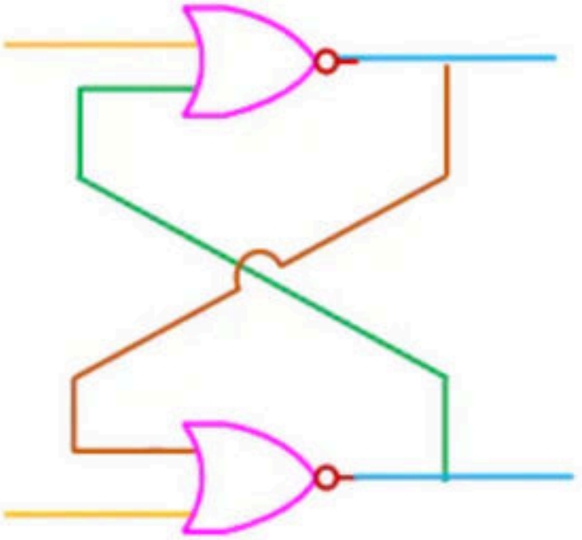
S	R	Q_{n+1}	State

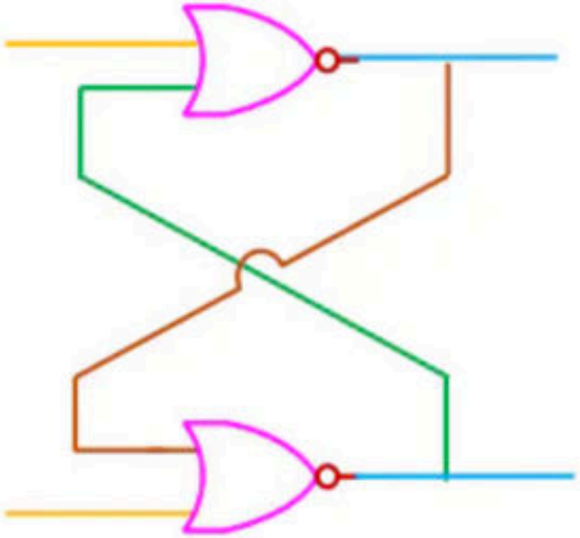
NOR Latch

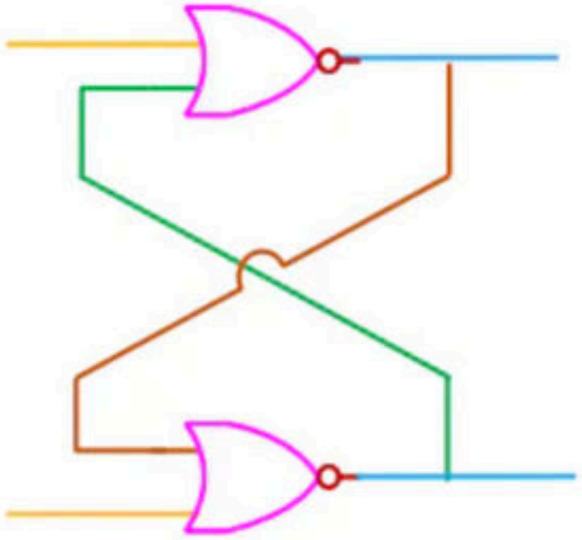


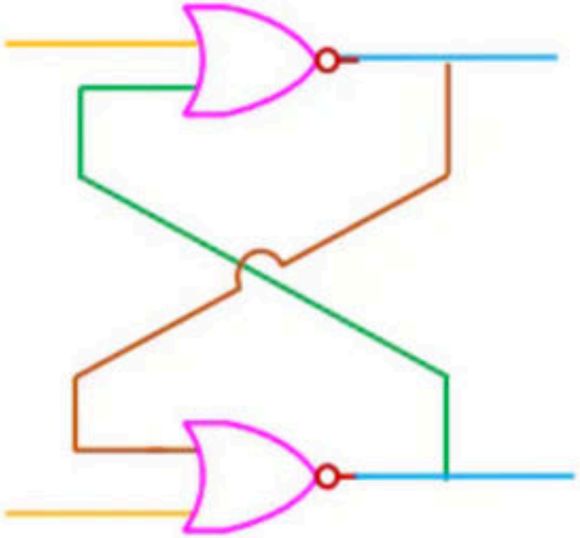
S	R	Q_n	Q_{n+1}	$\overline{Q_{n+1}}$

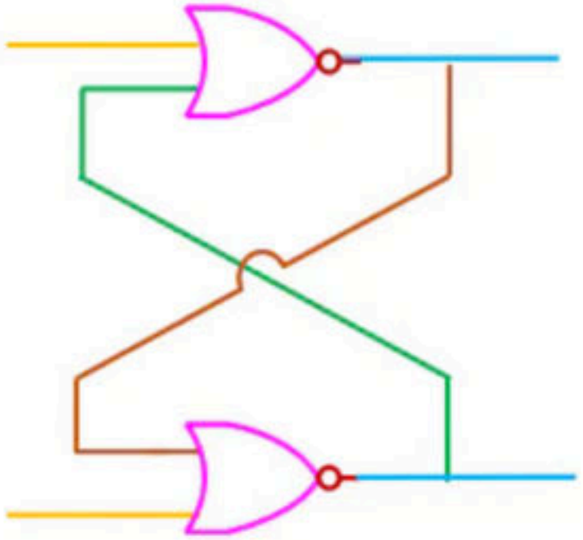


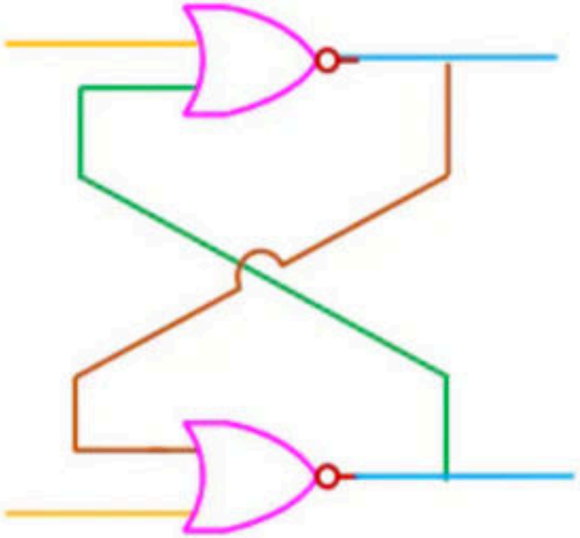


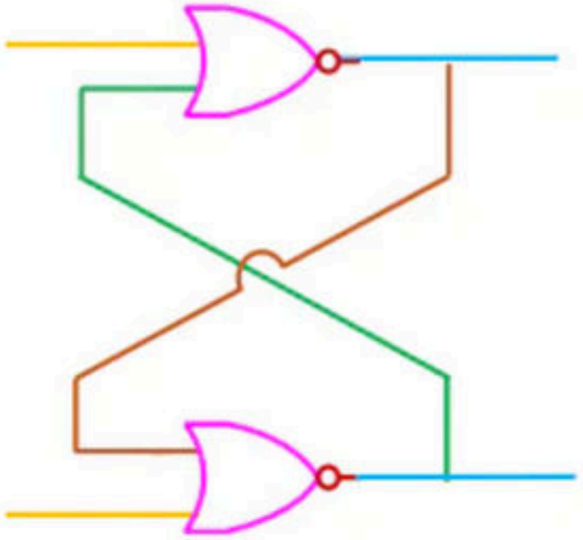




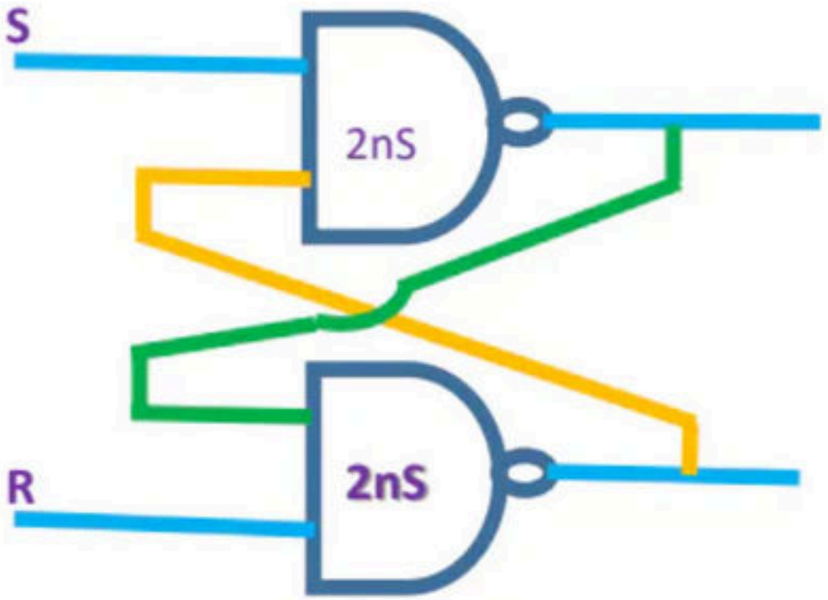


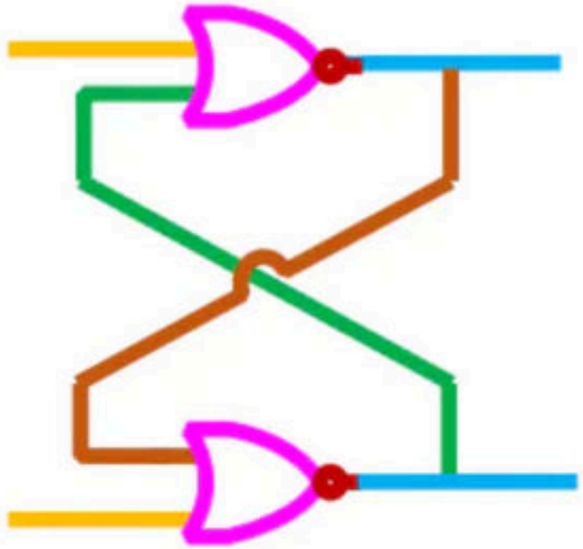






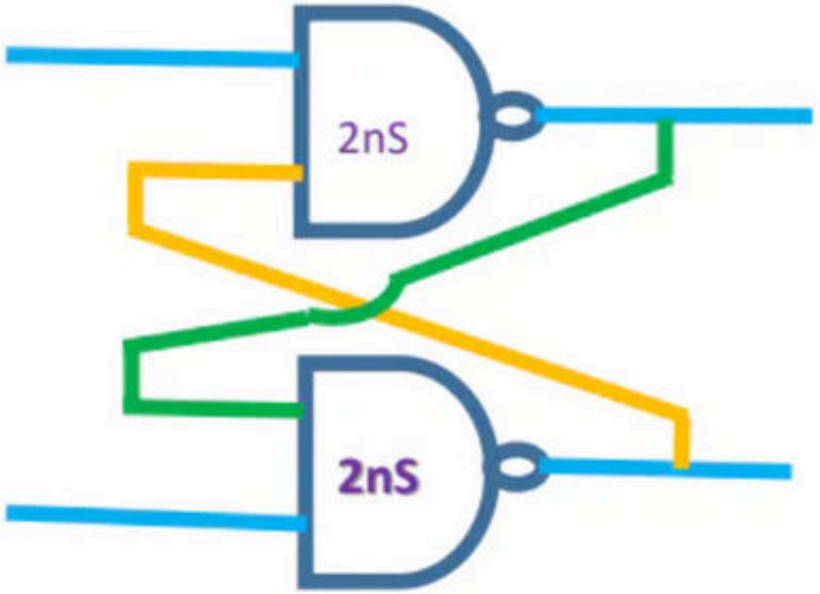
S	R	Q_{n+1}	State

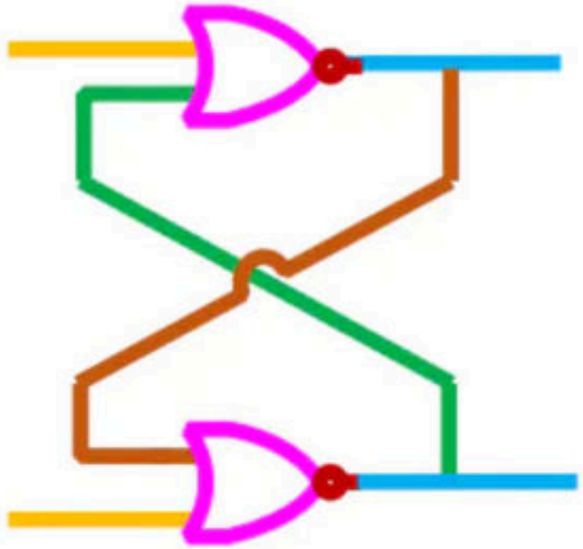




S	R	Q_{n+1}	State

S	R	Q_{n+1}	State

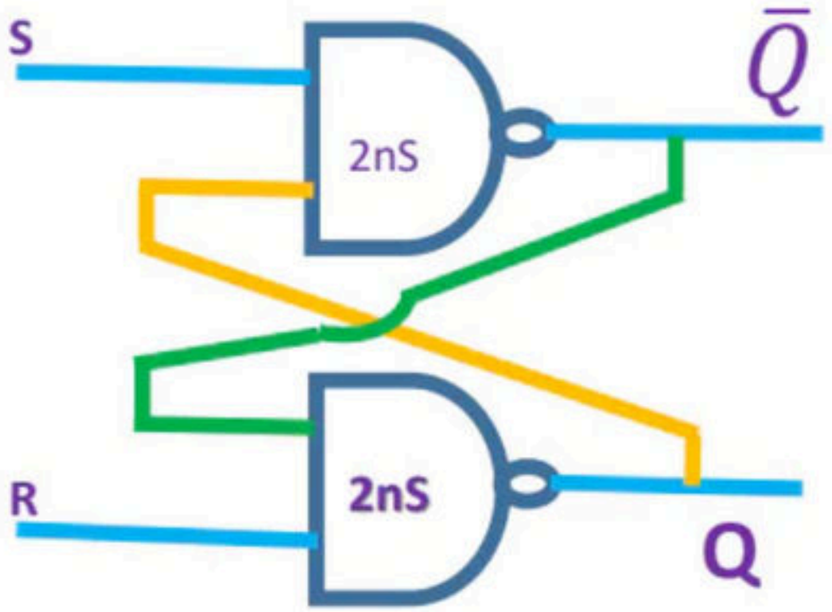




S	R	Q_{n+1}	State

S	R	Q_{n+1}	State

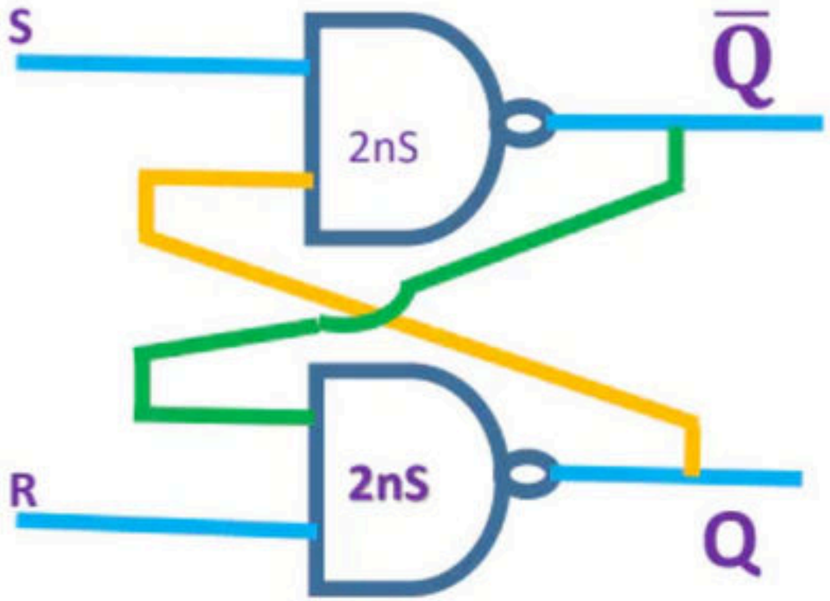
Q) Assume initially Q = 0 , for given circuit then find output for the following sequence SR = 01→10→11→00→01→11→10



- Out put of combinational circuit depends on input combinations
- Output of sequential circuits depends on input sequence
- For unequal delay of gates also the operation is valid

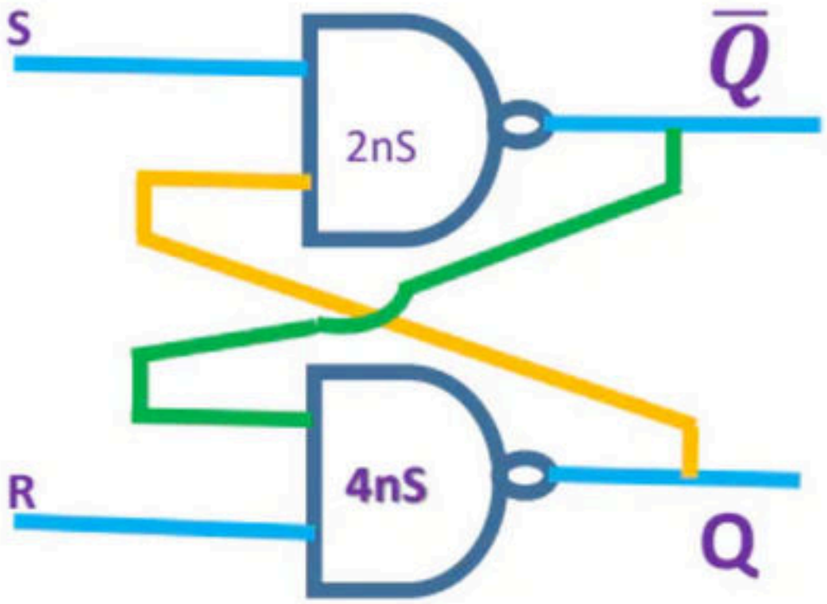
Q) Assume initially Q=0 , for given circuit then find output for the following sequence

SR = 00---->11 , assume equal delay



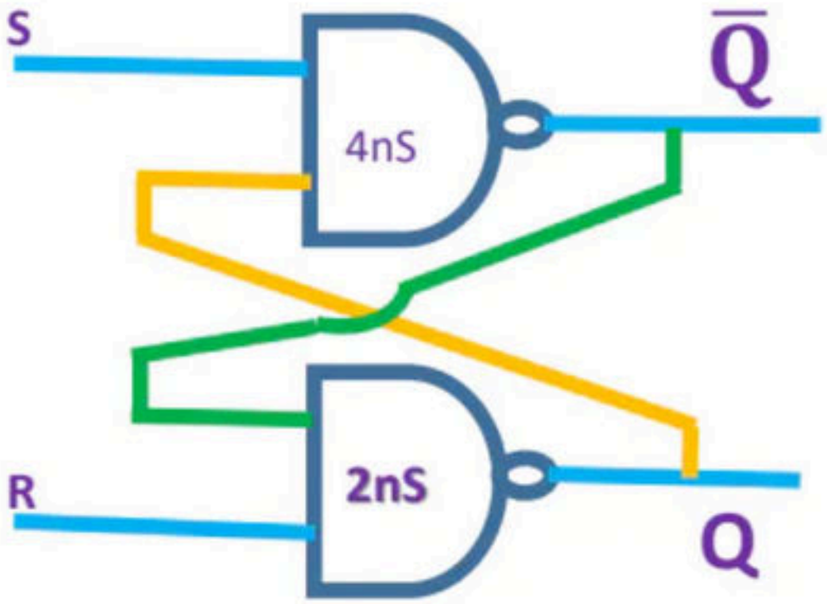
Q) Assume initially Q = 0 , for given circuit then find output for the following sequence

SR = 00---->11 , unequal delay



Q) Assume initially Q = 0 , for given circuit then find output for the following sequence

SR = 00---->11 , unequal delay



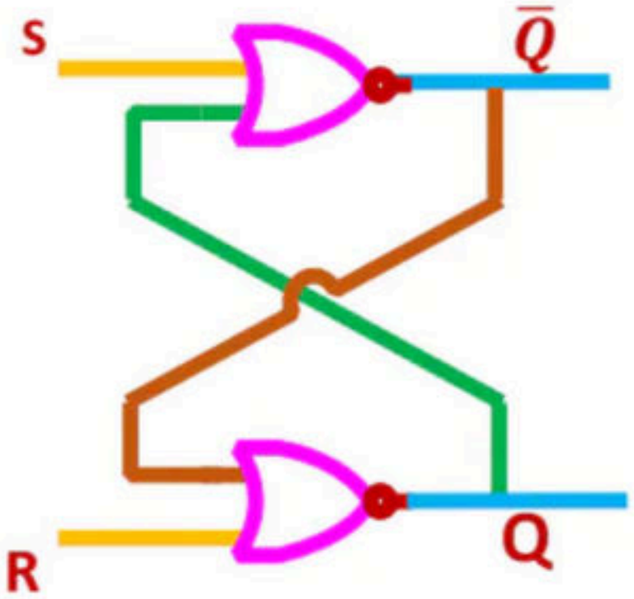
For **SR NAND** latch , if the input sequence is **00 followed by 11** , then the following cases arises

1. If the delay of both gates are same then we don't have any stable output , the output is oscillatory , this condition is known as ***critical race***

2. However if the delay of both gates are not equal then there exist a stable output , but it depends on the individual delay of the gates

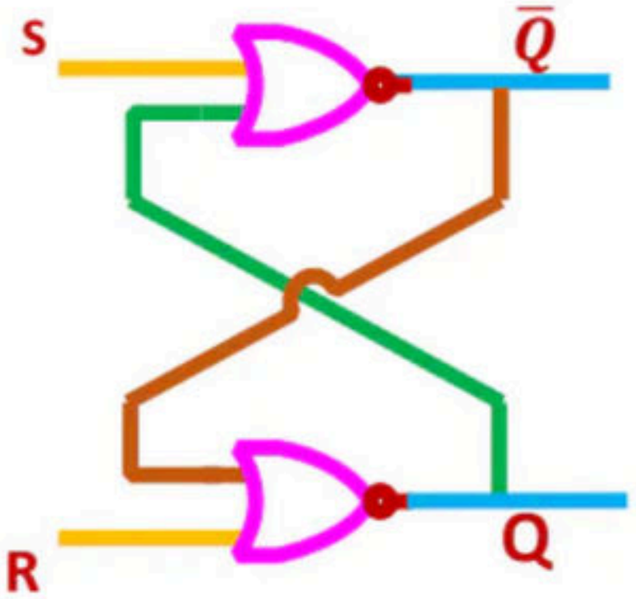
Q) Assume initially Q=0 , for given circuit then find output for the following sequence

SR = 01→10→00→11→01→11→10



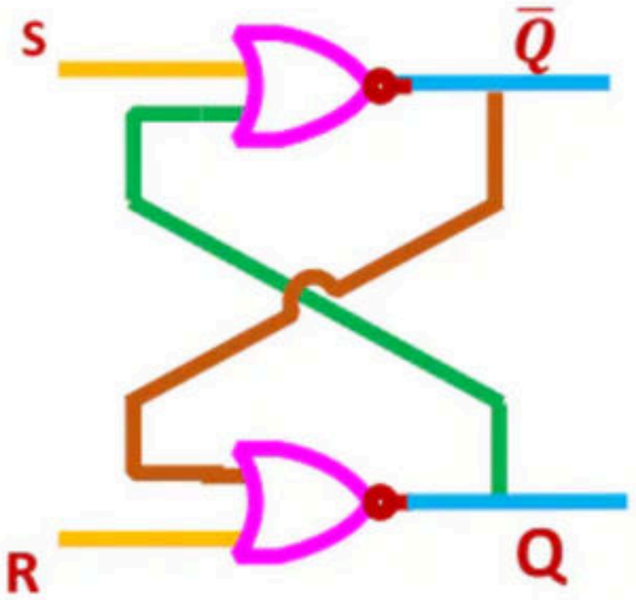
Q) Assume initially Q=0 , for given circuit then find output for the following sequence

SR =11→00 , Assume equal delay



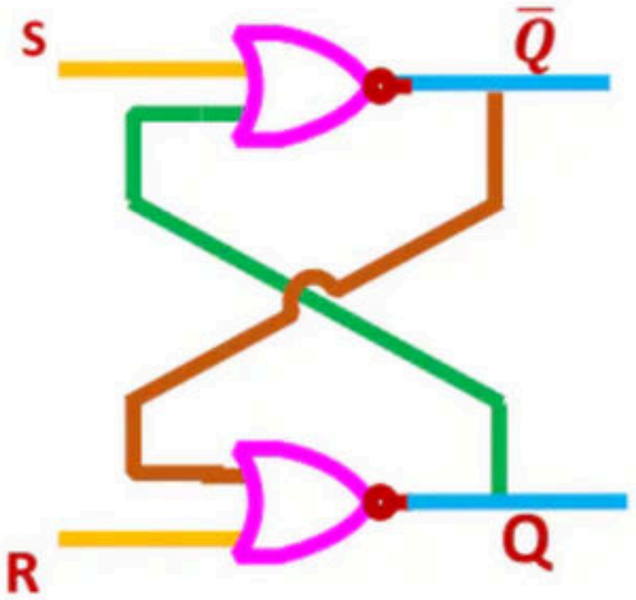
Q) Assume initially Q=0 , for given circuit then find output for the following sequence

SR =11→00 , Assume unequal delay



Q) Assume initially Q=0 , for given circuit then find output for the following sequence

SR =11→00 , Assume unequal delay

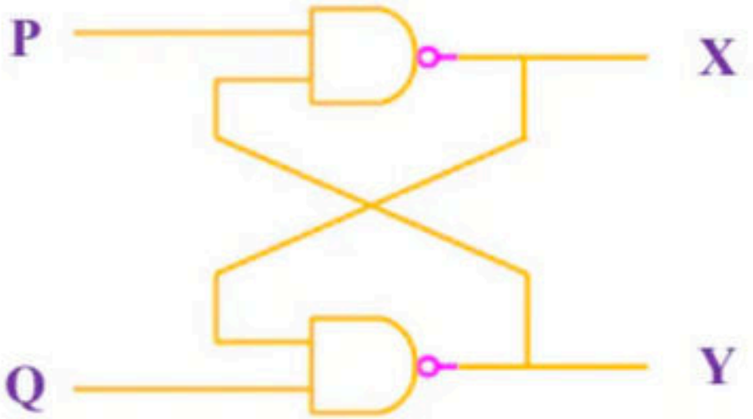


For **SR NOR** latch , if the input sequence is 11 followed by 00 , then the following cases arises

- If the delay of both gates are same then we don't have any stable output , the output is oscillatory , this condition is known as ***critical race***

- However if the delay of both gates are not equal then there exist a stable output , but it depends on the individual delay of the gates

Q. In the latch circuit shown, the NAND gates have non-zero, but unequal propagation delays. The present input condition is P = Q = '0'. If the input condition is changed simultaneously to P = Q = '1', the outputs X and Y are



(A) X = '1', Y = 'l'

(B) Either X = '1', Y='0' or X = '0', Y = 'l'

(C) Either X = '1', Y=l or X = '0', Y = '0'

(D) X = '0', Y = '0'

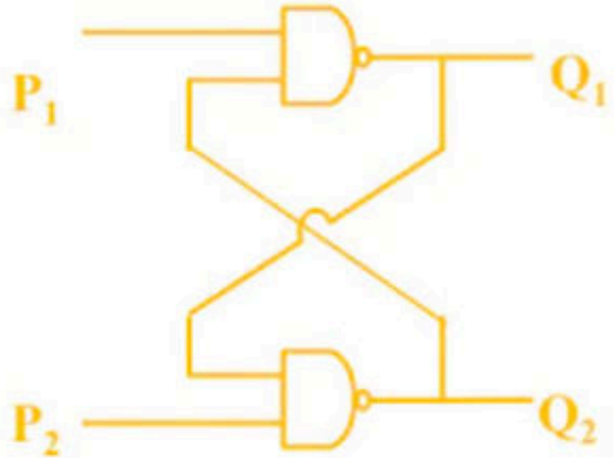
Q. Refer to NAND and NOR latches shown in the figure. The inputs (P_1, P_2) for both the latches are first made (0,1) and then, after a few seconds, made (1, 1). The corresponding stable outputs (Q_1, Q_2) are

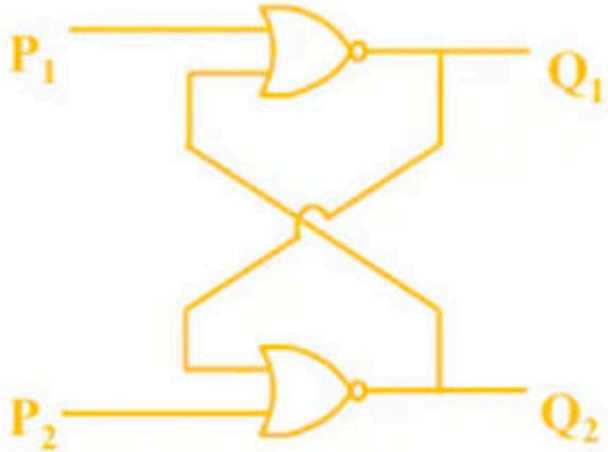
(A)NAND: first (0,1) then (0,1) NOR: first (1,0) then (0,0)

(B)NAND: first (1,0) then (1,0) NOR: first (1,0) then (1,0)

(C)NAND: first (1,0) then (1,0) NOR: first (1,0) then (0,0)

(D)NAND: first (1,0) then (1,1) NOR: first (0,1) then (0,1)





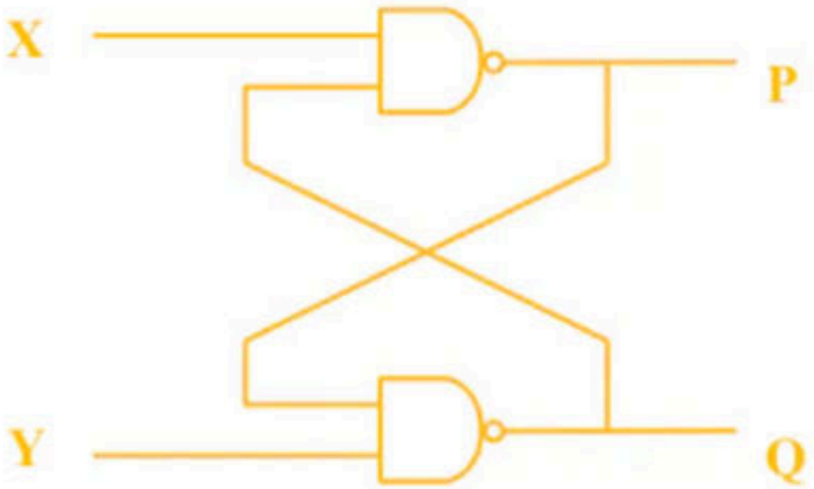
Q. The following binary values were applied to the X and Y inputs of the NAND latch shown in the figure in the sequence indicated below:
X = 0, Y = 1; X = 0, Y = 0; X = 1, Y = 1. The corresponding stable P, Q outputs will be:

(A) P = 1, Q = 0; P = 1, Q = 0; P = 1, Q = 0 or P = 0, Q = 1

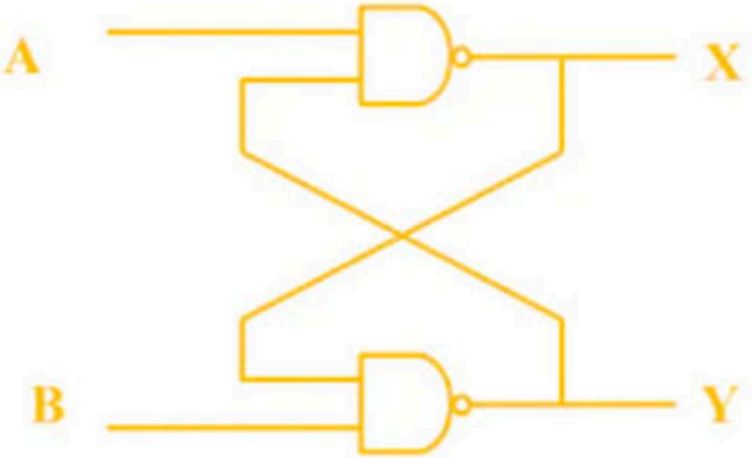
(B) P = 1, Q = 0; P = 0, Q = 1; or P = 0, Q = 1; P = 0, Q = 1

(C) P = 1, Q = 0; P = 1, Q = 1; P = 1, Q = 0 or P = 0, Q = 1

(D) P = 1, Q = 0; P = 1, Q = 1; P = 1, Q = 1 or P = 0, Q = 1



Q. The given figure, A = 1 and B = 1, the input B is now replaced by a sequence 101010….
The outputs x and y will be.



(A)Fixed at 0 and 1, respectively.

(B)x = 1010 …. While y = 0101 ….

(C)x = 1010 …. and y = 0101 ….

(D)Fixed at 1 and 0, respectively.

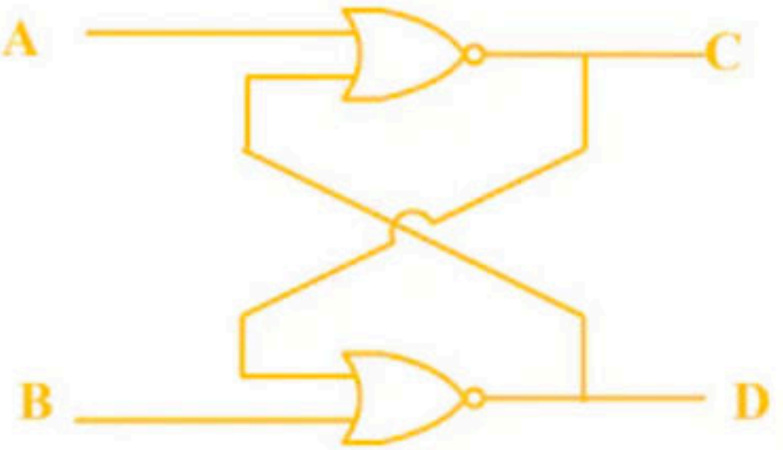
Q. In the circuit shown in figure, when inputs A = B = 0, the possible logic states of C and D are

(A)C = 0, D = 1 or C = 1, D = 0

(B)C = 1, D = 1 or C = 0, D = 0

(C)C = 1, D = 0

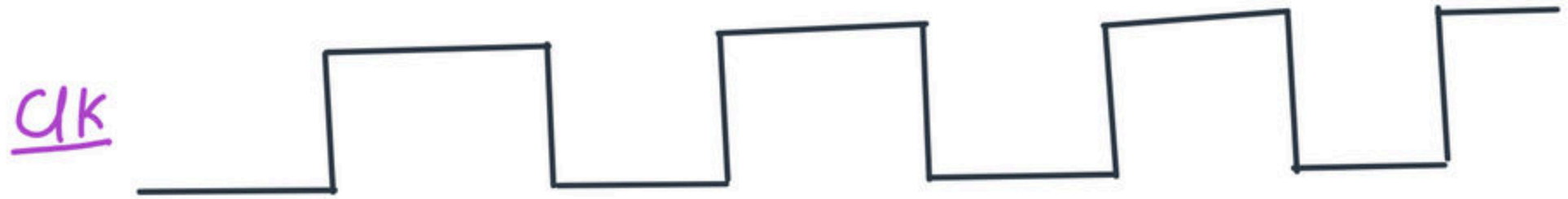
(D)C = 0, D = 1



FLIP FLOP

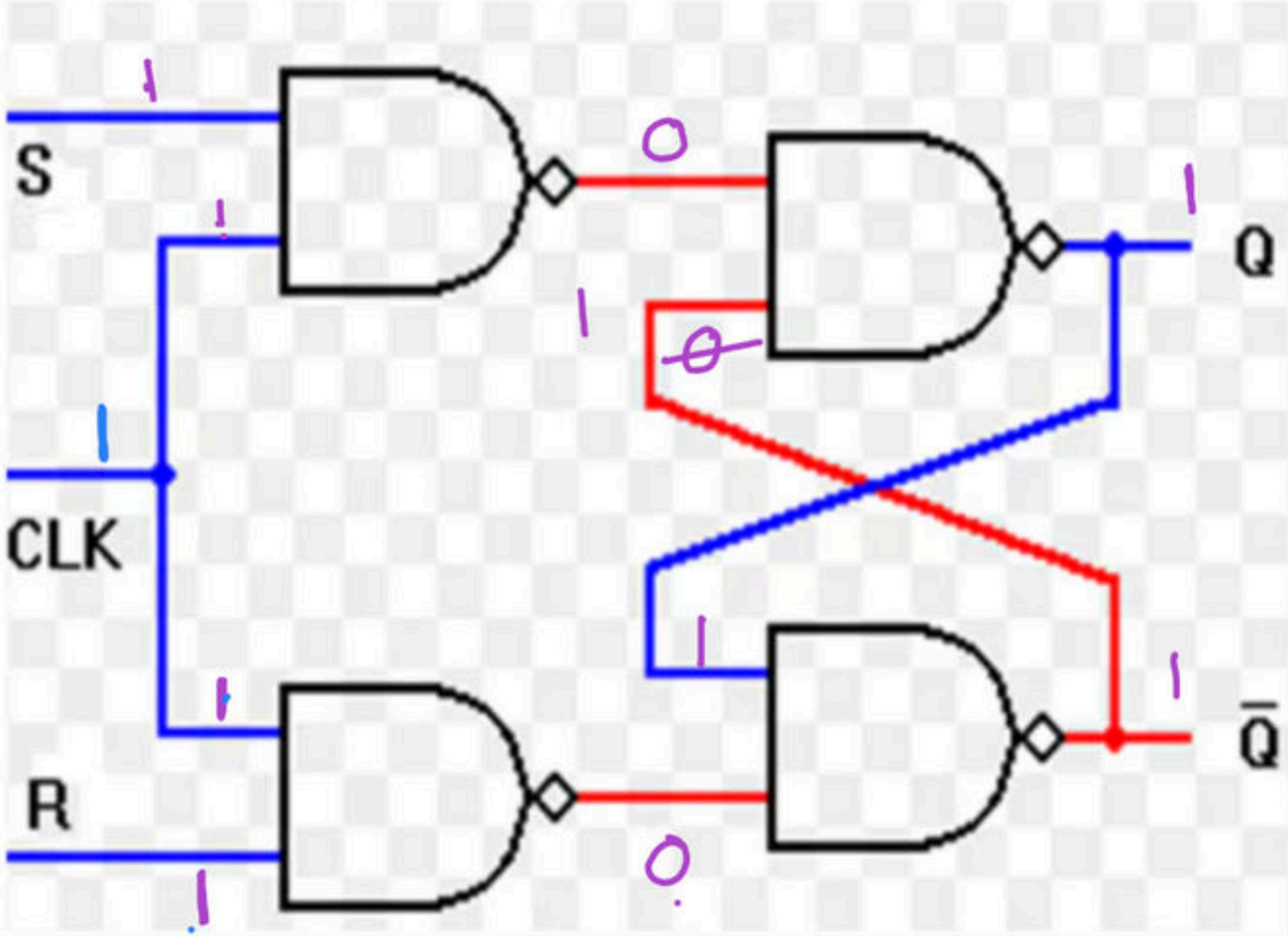
In a latch the output changes immediately in response to external input , so to have an additional control , we are introducing a signal called " ***CLOCK*** " , whose purpose is same as Enable pin of Decoder.

Latch + Clock = Flip Flop



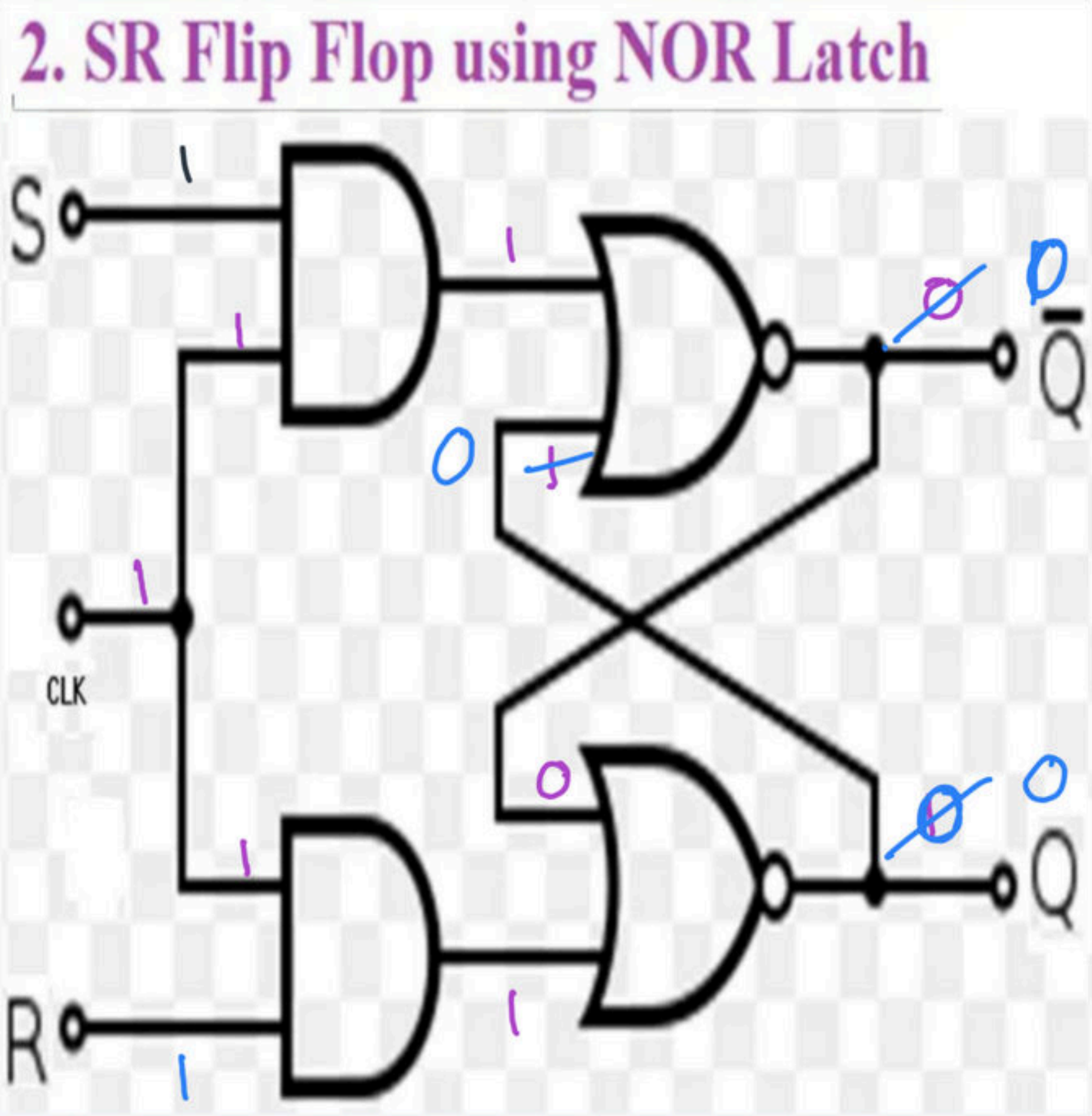
SR Flip Flop

1. SR Flip Flop using NAND Latch



CLK	S	R	Q	Q+
0	x	x	x	Q
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	x
1	1	1	1	x

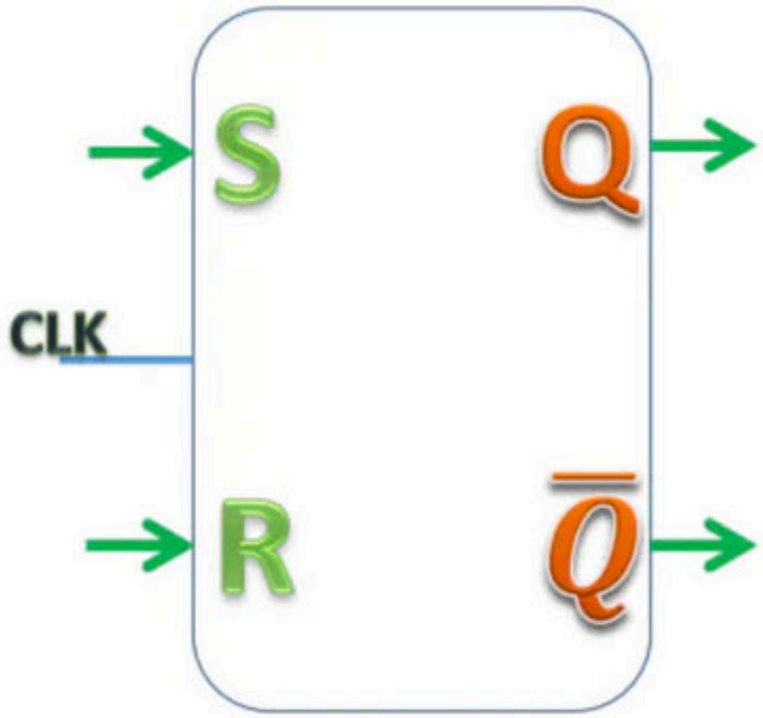
2. SR Flip Flop using NOR Latch



CLK	S	R	Q	Q+
0	x	x	a	a
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	x
1	1	1	1	x

- Latches are universally not unique and hence their truth tables are not unique .
- Flip Flops are universally unique, and their truth tables are unique.

S R Flip Flop



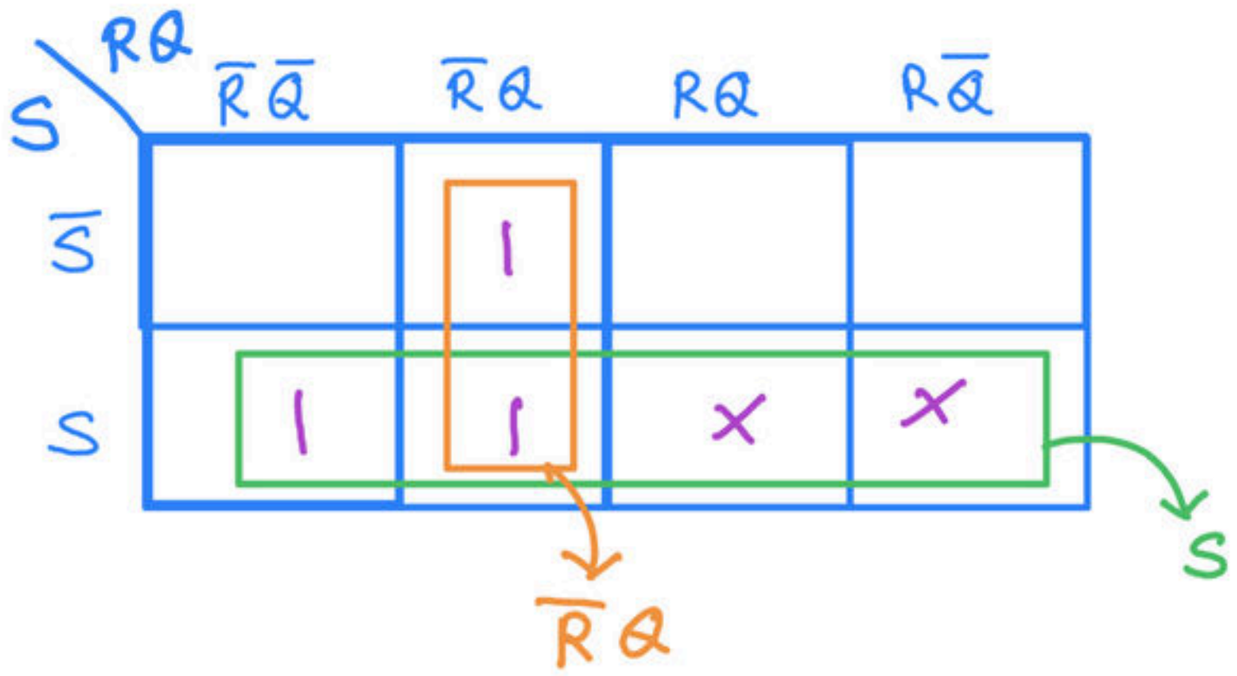
CLK	S	R	Q+	State
0	×	×	Q	Hold
1	0	0	Q	Hold
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	×	Invalid.

Characteristic table

CLK	S	R	Q	Q+
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	x
1	1	1	1	x

Characteristic Equation

$$Q^+(S, R, Q) = \Sigma m(1, 4, 5) + d(6, 7)$$



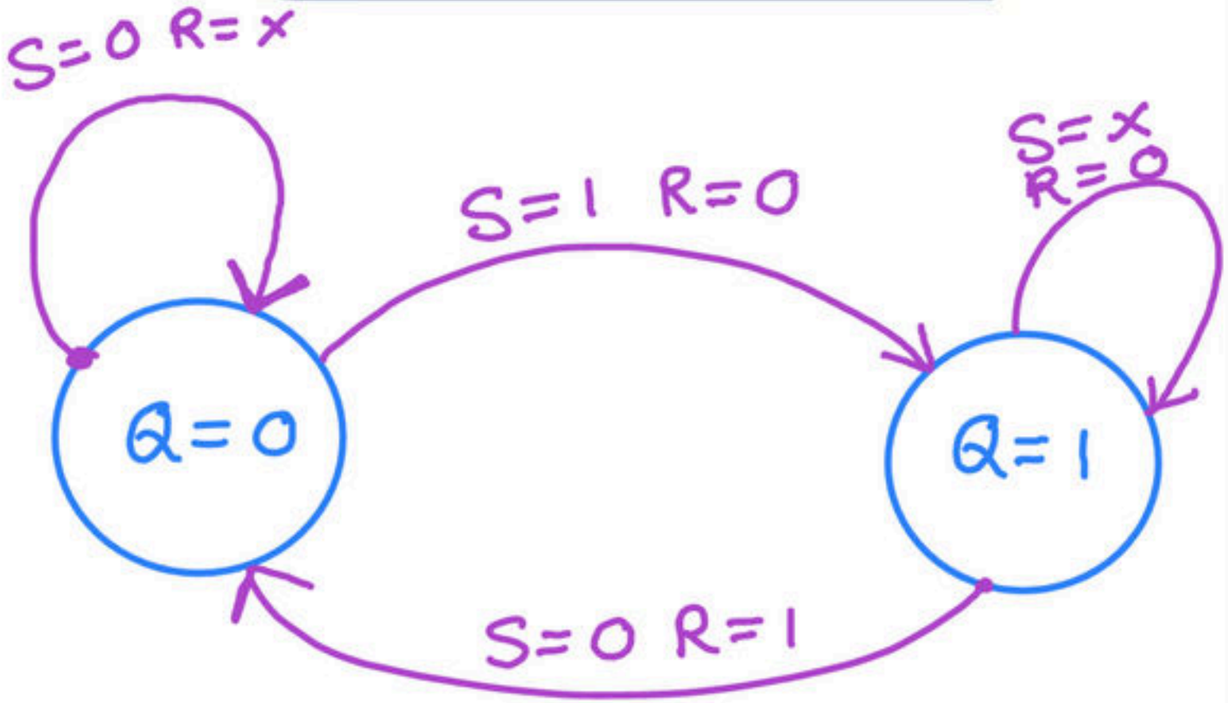
$$Q^+(S, R, Q) = S + \bar{R}Q$$

not valid $S = R = 1$.

Excitation table

Q	Q+	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

State Diagram



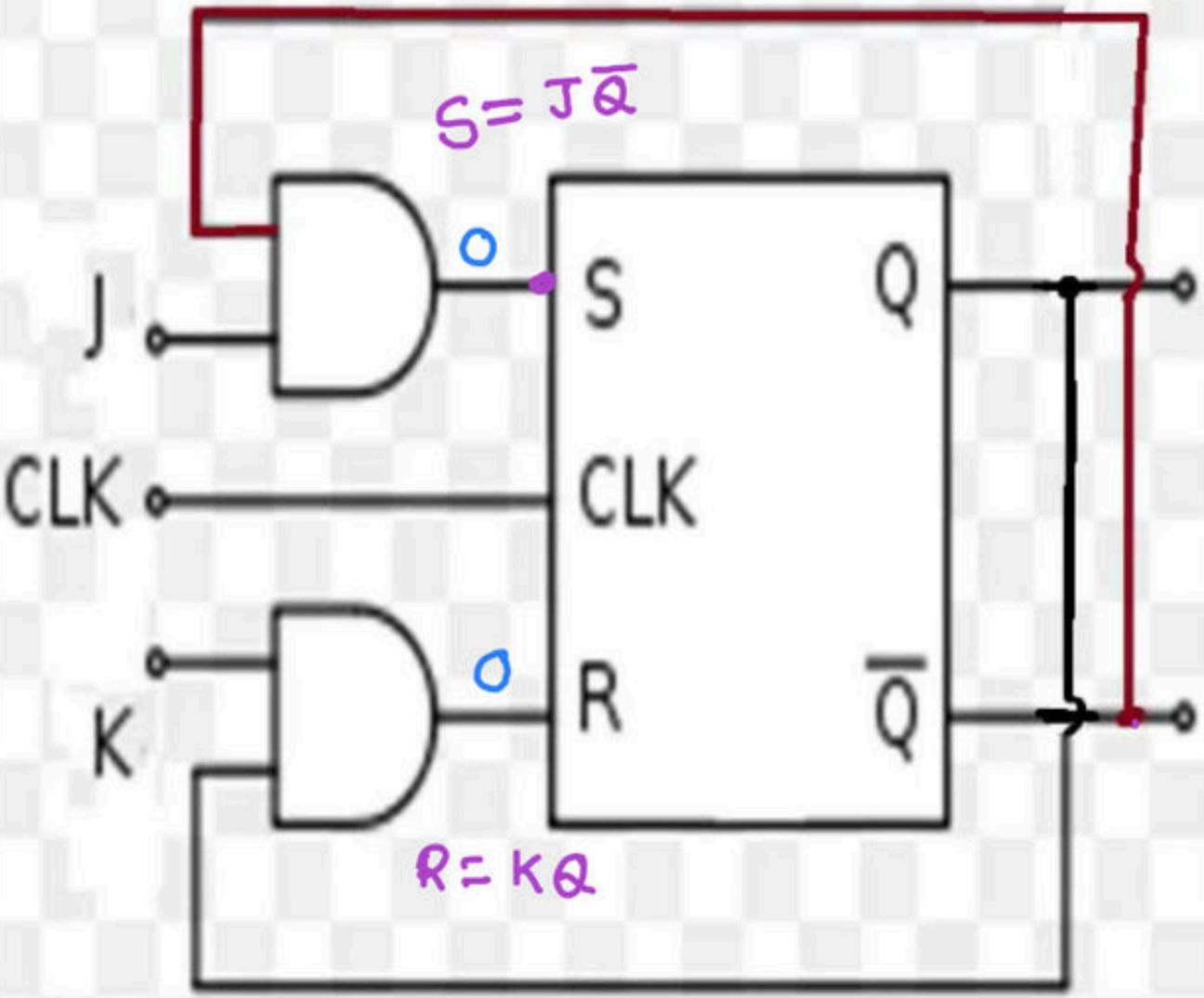
Drawbacks

Because of presence of invalid state there is a restriction on the sequence of the applied input ,since it leads to **CRITICAL RACE**, which is undesirable .

$Q^+ = J\bar{Q} + \bar{K}Q$.

J K Flip Flop

$Q^1 = 0 + 1Q$



CLK	J	K	Q+
0	×	×	Q
1	0	0	Q
1	0	1	0
1	1	0	1
1	1	1	$\bar{Q}$

$$Q^+ = S + \bar{R}Q$$

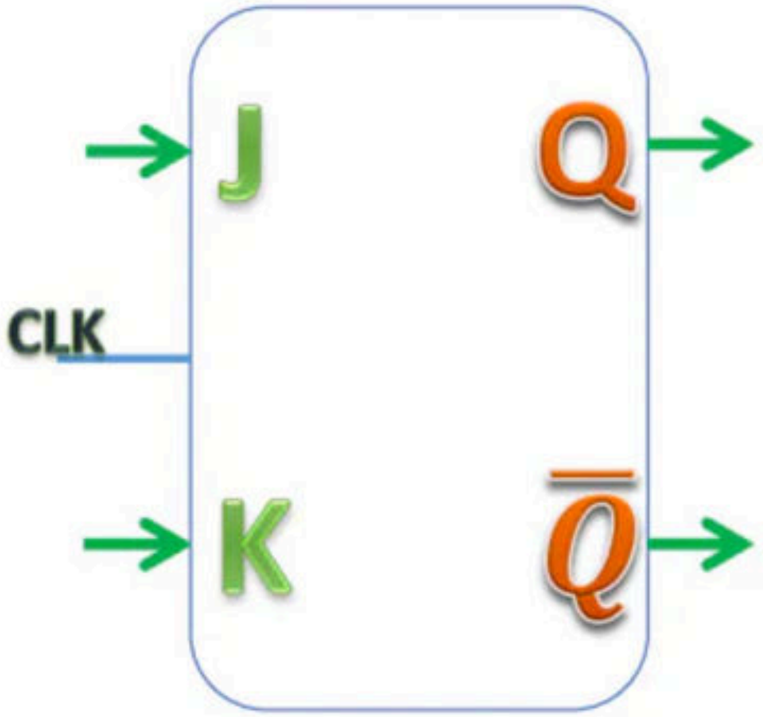
$$S = J\bar{Q} \qquad R = KQ.$$

$$Q^+ = J\bar{Q} + \overline{KQ}\,Q$$

$$Q^+ = J\bar{Q} + (\bar{K}+\bar{Q})Q$$

$$\boxed{Q^+ = J\bar{Q} + \bar{K}Q.}$$

J K Flip Flop



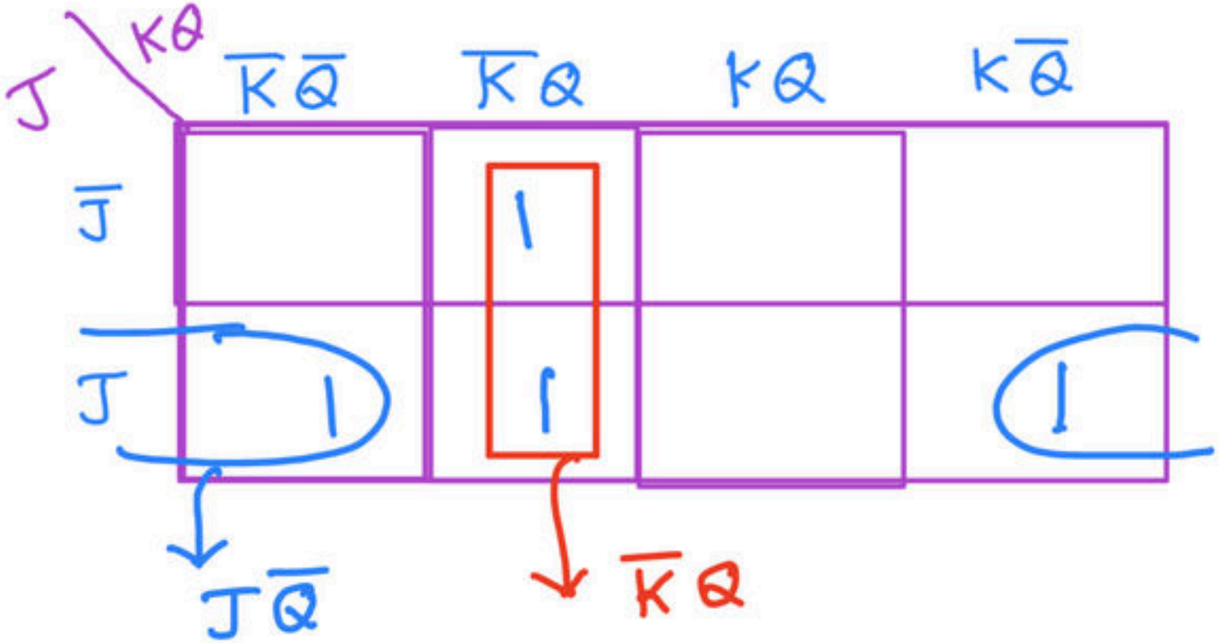
CLK	J	K	Q+	State
0	×	×	Q	Hold
1	0	0	Q	Hold
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	$\overline{Q}$	Toggle.

Characteristic table

CLK	J	K	Q	Q+
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Characteristic Equation

$Q^+(J, K, Q) = \Sigma m(1, 4, 5, 6)$

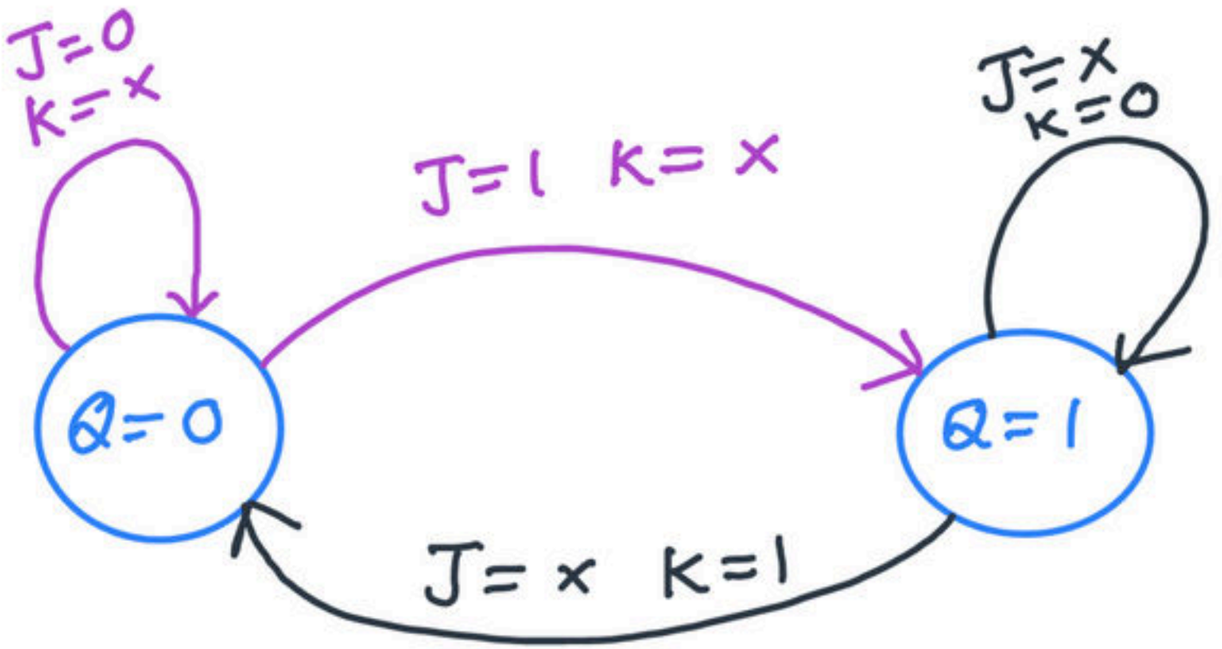


$$Q^+(J, K, Q) = J\bar{Q} + \bar{K}Q$$

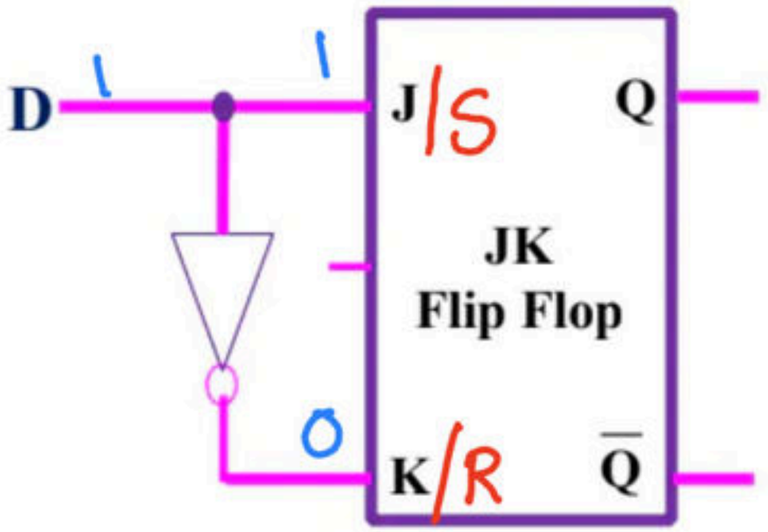
Excitation table

Q	Q +	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

State Diagram



D Flip Flop



CLK	D	Q+
0	x	Q
1	0	0
1	1	1

$$Q^+ = D$$

Characteristic table

CLK	D	Q	Q+
0	x	Q	Q
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Characteristic Equation

$$Q^+(D, Q) = \Sigma m(2, 3)$$

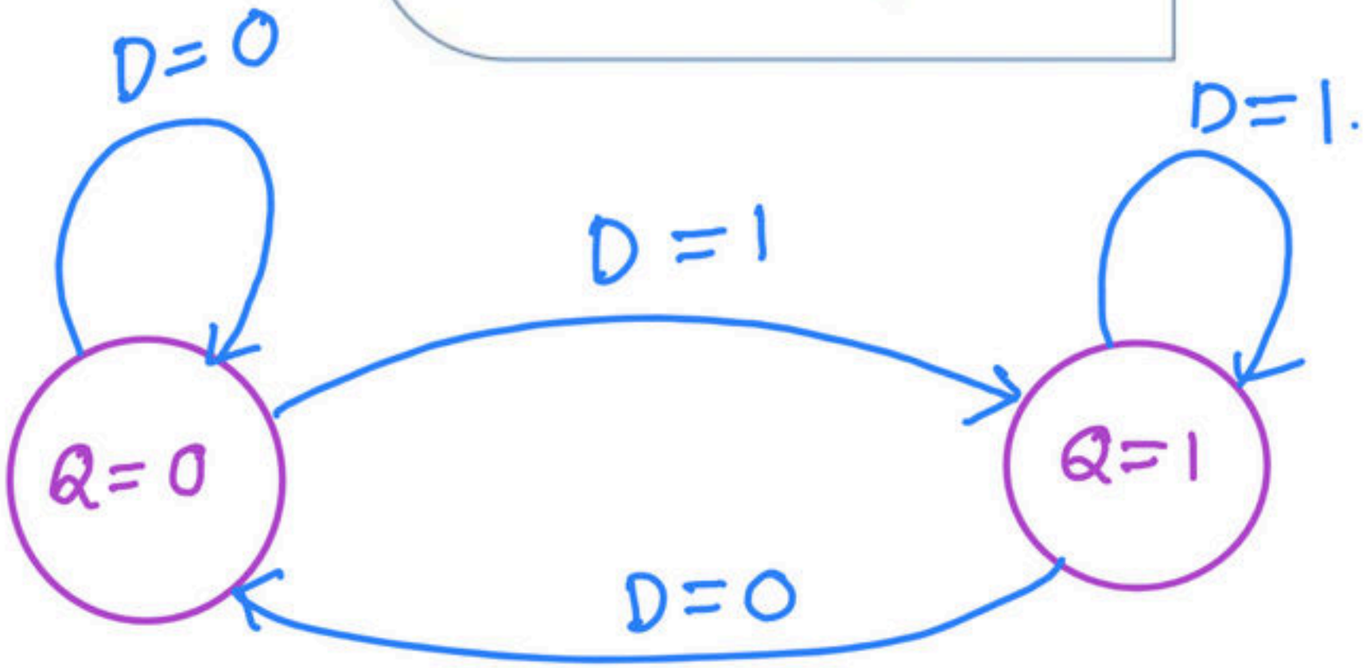
$$= D\overline{Q} + DQ$$

$$\boxed{Q^+(D, Q) = D}$$

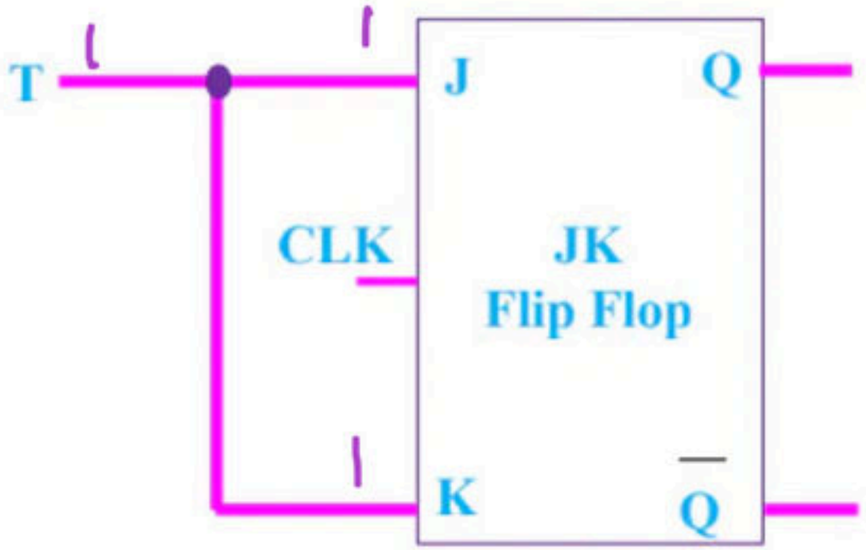
Excitation table

Q	Q+	D
0	0	0
0	1	1
1	0	0
1	1	1

State Diagram



T Flip Flop



CLK	T	Q+
0	X	Hold
1	0	Hold
1	1	Toggle.

Characteristic table

CLK	T	Q	Q+
0	X	Q	Q
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Characteristic Equation

$$Q^+(T, Q) = \Sigma m(1, 2)$$

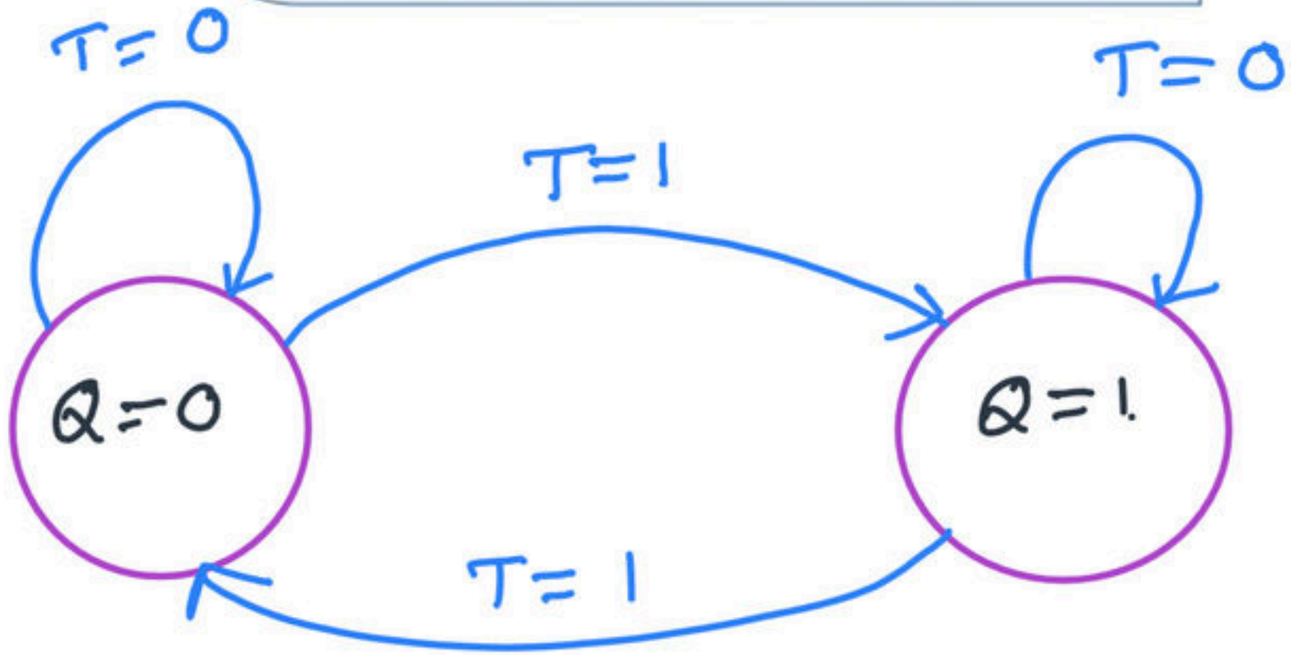
$$= \bar{T}Q + T\bar{Q}$$

$$\boxed{Q^+(T, Q) = T \oplus Q}$$

Excitation table

Q	Q+	T
0	0	0
0	1	1
1	0	1
1	1	0

State Diagram



Digital Short Notes

Follow me @uncademy/bvreddy
for the full length course
93980 21419

Use the Code: **BVREDDY**, to get maximum discount ,
complete notes ,DDPs and Short Notes

Positive logic system

High voltage corresponds to logic " 1"
Maximum positive value is taken as logic '1

+5V ----> logic "1"

0V ----> logic "0 "

Negative logic system

High voltage corresponds to logic " 0"
Maximum positive value is taken as logic '0
'

+5V ----> logic "0"

0V ----> logic "1 "

A positive logic system is converted into negative logic system by using the concept of duality

Finding the dual of a given Boolean expression

1. $* \leftrightarrow +$
2. $0 \leftrightarrow 1$
3. Keep the variables as it is

BVREDDY

OR -Operation

A+ 0 = A
1+A= 1
A+A=A
$A+\bar{A}= 1$

AND-Operation

$A* 1 = A$
$A* 0 = 0$
$A* A = A$
$A* \bar{A} = 0$

BVREDDY

Commutative Law

A+B= B+A

$A* B = B * A$

Associative Law

A+B+C= (A+B)+C= (B+C)+A= (C+A)+B

$A* B * C = (A * B) * C = (B * C) * A = (C * A) * B$

Trick to find Complement

1. Find the dual
2. Complement the each literals

Distribution Law (Mingle wala)

A(B+C)=AB+AC

A+BC= (A+B)(A+C)

D- Morgan's Law

$\overline{AB} = \bar{A} + \bar{B}$

$\overline{A + B} = \bar{A}\bar{B}$

Transposition theorem (T- 1)

(A+B)(A+C)= A+BC

Transposition theorem (T- 2)

$(A+B)(\bar{A} + C)$ =AC+$\bar{A}B$

Canonical form : Each minterm (maxterms) contains all the Boolean variables

F(A, B,C)= ABC+ $\bar{A}BC + AB\bar{C}$ ---->SOP

F(A, B, C) = (A+B+C) (A+$\bar{B}$ + C)($\bar{A}$ + B + $\bar{C}$) --> POS

Minimal Form : The minimized form of Boolean expression

F(A, B,C)= $BC + AB$

F(A, B, C) = (A+B) (A+$\bar{B}$)($\bar{A}$ + $\bar{C}$)

Literal : A Boolean variable either in normal form (or) complimented form is known as literal

Minterm :Each term in canonical SOP representation is known as minterm

Maxterm: Each term in canonical POS representation is known as maxterm

Consensus theorem (Rajinikanth wala)

$$AB + \bar{A}C + BC = AB + \bar{A}C$$

$$(A+B)(\bar{A} + C)(B + C) = (A + B)(\bar{A} + C)$$

BVREDDY

1. Maximum possible minterms = 2^n
2. Maximum possible maxterms = 2^n

BVREDDY

3. Number of minterm's + number of maxterm's = 2^n
4. The sum of all the minterms = **ONE**
5. The product of all maxterms = **ZERO**
6. Minterm's and maxterm's of same index are **compliment** to each other
7. By using 2- Boolean variables total number of possible Boolean functions = 16
8. By using n- Boolean variables total number of possible Boolean functions = 2^{2^n}
9. By using 2- Boolean variables total number of possible Boolean functions having at most 3- minterms = $4_{C_0} + 4_{C_1} + 4_{C_2} + 4_{C_3} = 15$
10. By using 2- Boolean variables total number of possible Boolean functions having at most 3- maxterms = 15
11. By using 2- Boolean variables total number of possible Boolean functions having 3- minterms = $4_{C_3} = 4$
12. By using n- Boolean variables total number of possible Boolean functions having 2- minterms = $2^n{}_{C_2}$
13. By using 5- Boolean variables total number of possible Boolean functions having at most 3- minterms = $32_{C_0} + 32_{C_1} + 32_{C_2} + 32_{C_3}$

BVREDDY

Neutral Function :

The number of minterms = number of maxterms

Mutually exclusive terms

The mutually exclusive term of m_i is m_{2^n-i-1}

BVREDDY

Self Dual Expression

If one time dual of the Boolean expression result the same expression , then it is called as self dual expression

Eg : f = AB+BC+AC

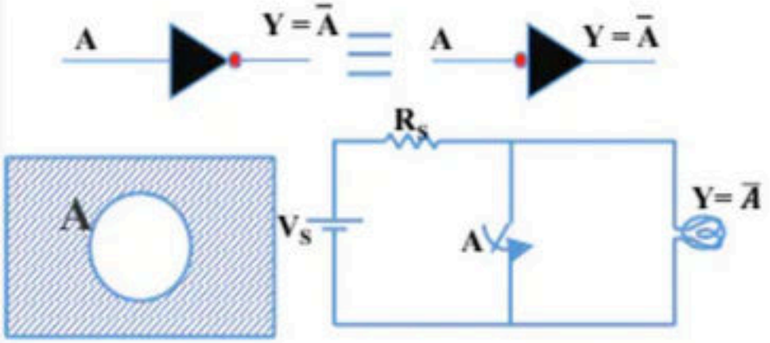
Conditions for the given expression is self dual

1. The number of minterms = number of maxterms (Neutral Function)
   number of minterms+ number of maxterms = 2^n
   number of minterms = number of maxterms = 2^{n-1}
2. If m_i belongs to f , then m_{2^n-i-1} should belongs to $\bar{f}$
3. The number of self dual functions = $2^{2^{n-1}}$

NOT GATE

$Y = \bar{A}$

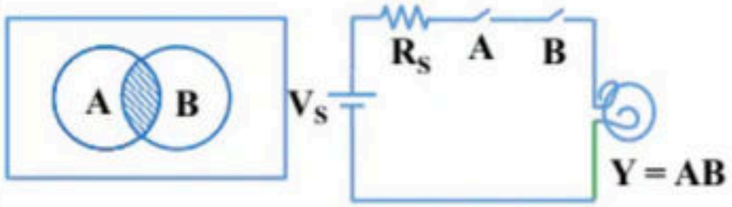
The output is the complement of the input

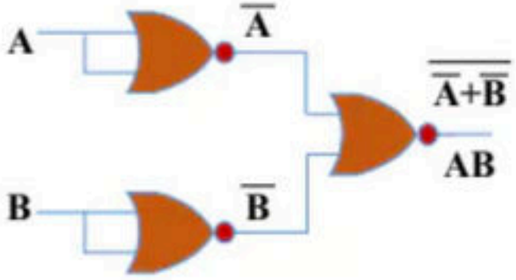


AND GATE

Y = AB

- Output is '0' if any one input '0'
- Y = AB = $\Sigma(3) = \Pi(0,1,2)$
- Enable input ⟹ 1
- Disable input ⟹ 0
- Commutative law ⟹ Obeys
- Associative law ⟹ Obeys

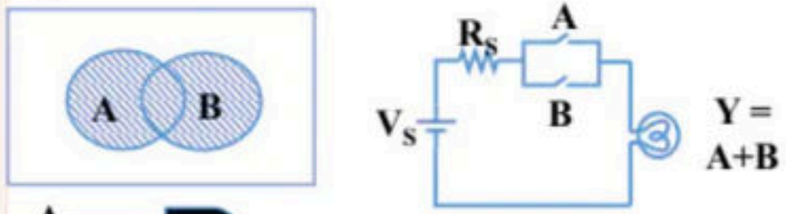


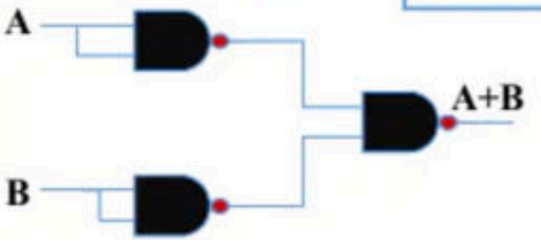


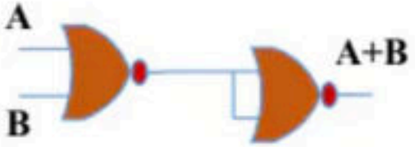
OR GATE

Y =A+B

- Output is '1' if anyone of the inputs are '1'
- Y = A+B = $\Sigma(1,2,3) = \Pi(0)$
- Enable input ⟹ 0
- Disable input ⟹ 1
- Commutative law ⟹ Obeys
- Associative law ⟹ Obeys





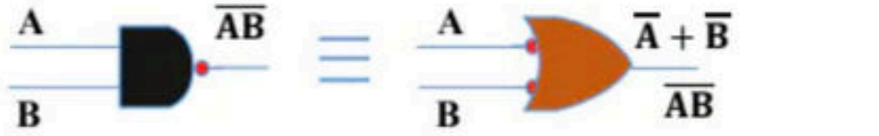


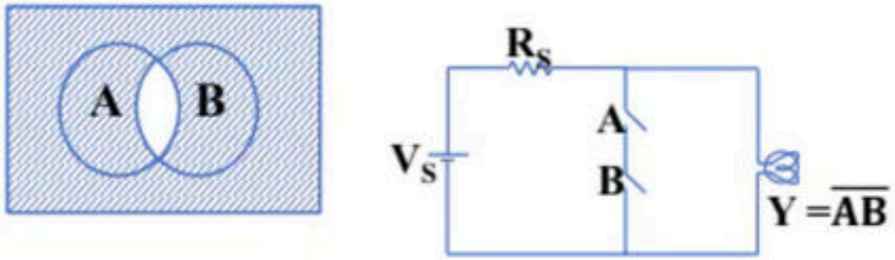
NAND GATE

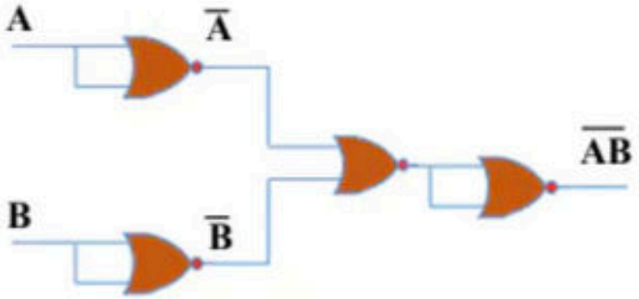
$$Y = \overline{AB}$$

- Output is '1' if any one input is '0'
- $Y = \overline{AB} = \sum(0,1,2) = \prod(3)$
- Enable input --1
- Disable input– 0
- Commutative law ---> Obeys
- Associative law ----> not Obeys

BVREDDY





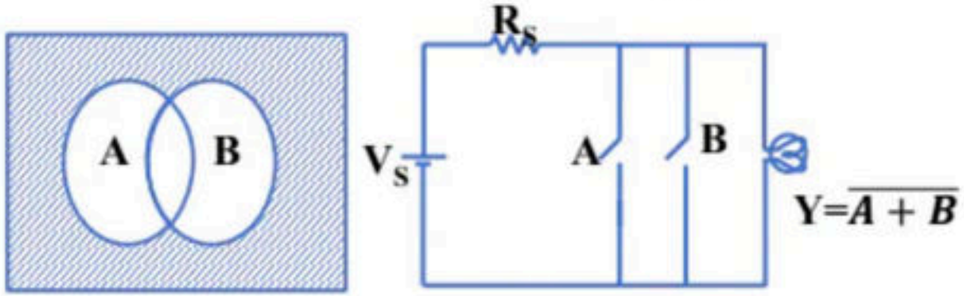


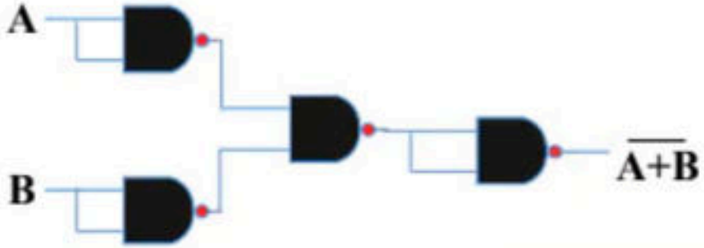
BVREDDY

NOR- GATE

$$Y = \overline{A + B}$$

- Output is '0' if any one of the input is '1'
- $Y = \overline{A + B} = \sum(0) = \prod(1,2,3)$
- Enable input --0
- Disable input– 1
- Commutative law ---> Obeys
- Associative law ----> not Obeys





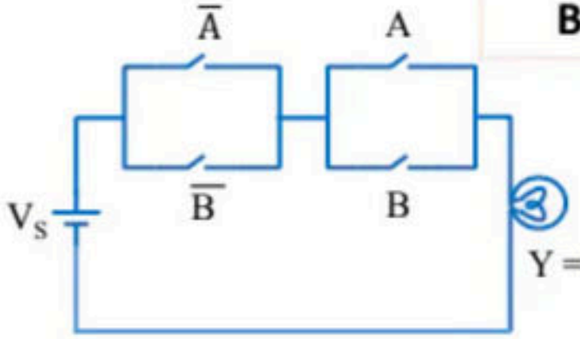
BVREDDY

BVREDDY

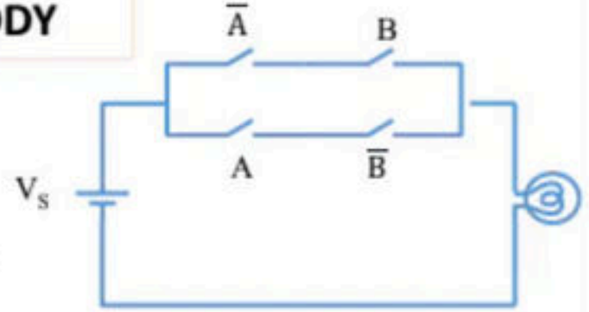
EX-OR GATE

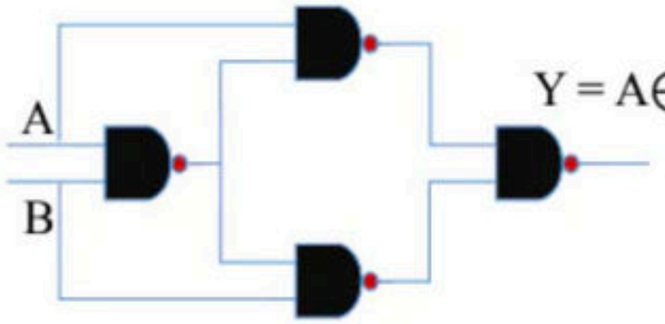
- Output is '1' for odd number of '1's in the input
- $\mathbf{Y = A \oplus B = \sum(1,2) = \Pi(0,3)}$
- $\mathbf{Y = A \oplus B \oplus C = \sum(1,2,4,7)}$
- $\mathbf{Y = A \oplus B \oplus C \oplus D = \sum(1,2,4,7,8,11,13,14)}$
- Commutative law $\Longrightarrow$ Obeys
- Associative law $\Longrightarrow$ Obeys
- $A \oplus 0 = A$
- $A \oplus 1 = \overline{A}$
- $A \oplus A = 0$
- $A \oplus \overline{A} = 1$
- $A \oplus A \oplus A \oplus \ldots\ldots\ldots\ldots n \text{ times} = \begin{cases} A, & n \text{ is odd} \\ 0, & n \text{ is even} \end{cases}$
- $A \oplus \bar{A}B = A + B$
- $AB \oplus BC = B(A \oplus C)$

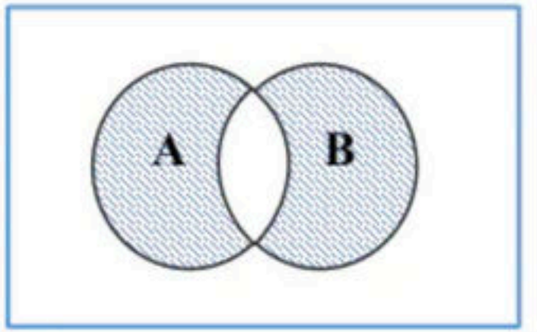
BVREDDY

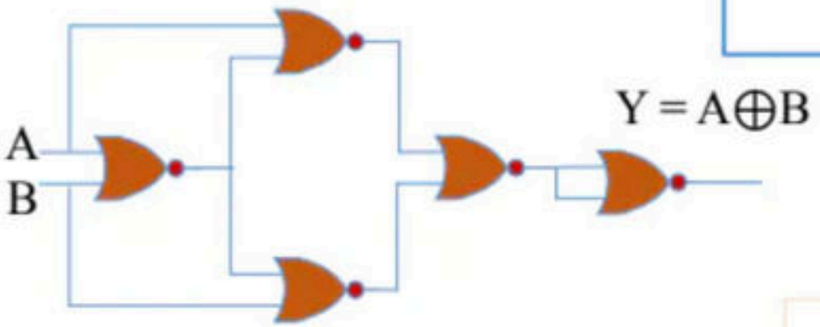


BVREDDY





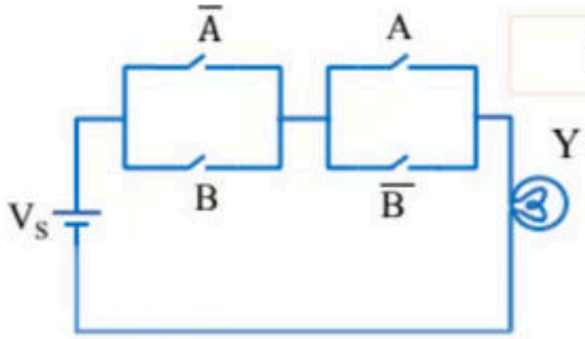


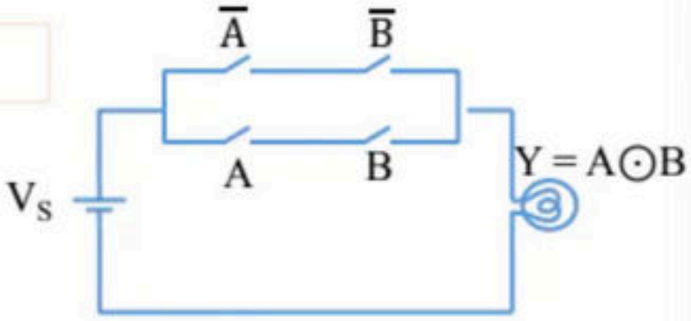


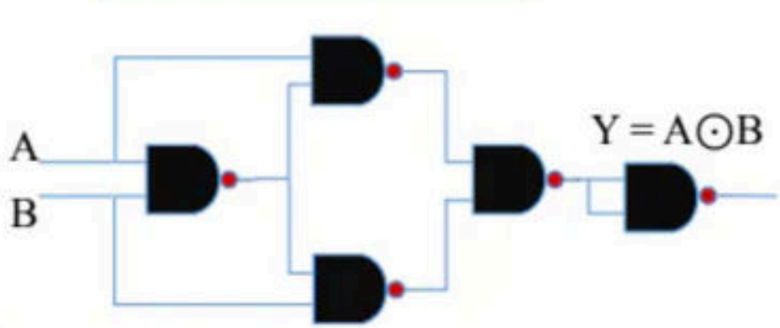
BVREDDY

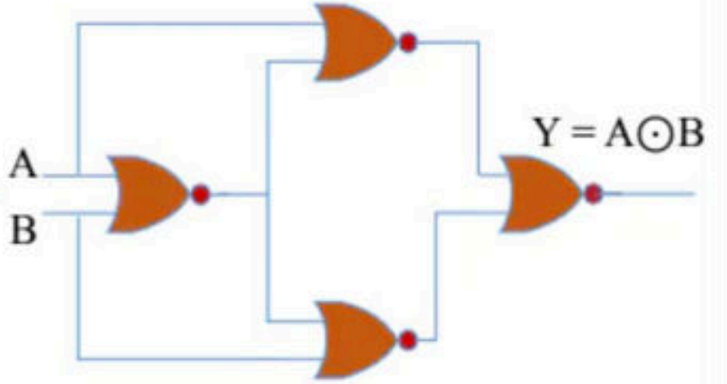
EX-NOR GATE

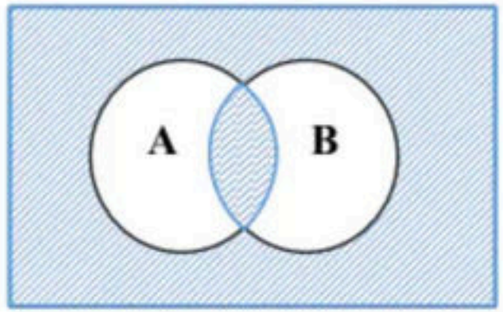
- Output is '1' for even number of '1's in the input
- $Y = A \odot B = \sum(0,3) = \Pi(1,2)$
- Commutative law $\Longrightarrow$ Obeys
- Associative law $\Longrightarrow$ not Obeys
- $A \odot 0 = \overline{A}$
- $A \odot 1 = A$
- $A \odot A = 1$
- $A \odot \overline{A} = 0$
- $A \odot A \odot A \odot$.....n times $= \begin{cases} \overline{A}, & n \text{ is odd} \\ 1, & n \text{ is even} \end{cases}$
- $\overline{A \odot B} = A \oplus B$
- $A \oplus \overline{B} = A \odot B$
- $\overline{A} \oplus B = A \odot B$
- $\overline{A} \oplus \overline{B} = A \oplus B$
- $A \odot B \odot C = \sum(0,3,5,6)$
- $A \oplus B \oplus C = \sum(1,2,4,7)$
- $(A \odot B) \odot C = \sum(1,2,4,7)$
- $(A \odot C) \odot B = \sum(1,2,4,7)$
- $A \oplus B \oplus C = (A \odot B) \odot C = (A \odot C) \odot B$
- $A \odot B = \overline{A} \oplus B = A \oplus \overline{B} = \overline{A} \odot \overline{B}$
- $A \oplus B = A \odot \overline{B} = \overline{A} \odot B = \overline{A} \oplus \overline{B}$
- $\overline{A \oplus B \oplus C} = A \odot B \odot C = [A \oplus B] \odot C = A \odot [B \oplus C]$











EX-OR GATE	EX-NOR GATE
Output is '1' for odd number of '1's in the input	Output is '1' for even number of '1's in the input
Odd number of 1's detector	Even number of 1's detector
Inequality detector	Equality detector
Anti-coincident gate	Coincident gate

	No. of NAND GATES	No . of NOR GATES
NOT	1	1
AND	2	3
OR	3	2
EX-OR	4	5
EX-NOR	5	4
NAND	1	4
NOR	4	1

- For a n- variable Boolean expression , the maximum number of literals = n
- For a n- variable K- Map if group is done by considering 2^m number of cells , then the resulting term from that group contains (n- m) number of literals .
- *8 cells* – 2^3 cells → Octet --> 3 variables eliminated
- *4 cells* – 2^2 cells → Quad ---> 2 variables eliminated
- *2 cells* – 2^1 cells → Pair ---> 1 variables eliminated

- Minimal expression may not be unique .
- The minimal expression = (All EPI's) + (Optional PI's)
- If all PI's are EPI's , then the minimal expression is unique
- The sufficient condition for a K-map to have unique solution is number of PI's = number of EPI's

Use the Code : BVREDDY

K- Map

Implicant : Each minterm in canonical SOP expression is known as Implicant .

Prime Implicant is a product term , obtained by combining maximum possible cells in the K- Map. While doing so make sure that a smaller group is not completely inside a bigger group .

Essential Prime Implicant : A prime Implicant is an EPI , if and only if it contains at least one minterm which is not covered by multiple groups

All EPI's are PI's , but vice versa not true

EPI ≤ PI

Minterm mode

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	$\bar{A}\bar{B}\bar{C}\bar{D}$ 0	$\bar{A}\bar{B}\bar{C}D$ 1	$\bar{A}\bar{B}CD$ 3	$\bar{A}\bar{B}C\bar{D}$ 2
$\bar{A}B$	$\bar{A}B\bar{C}\bar{D}$ 4	$\bar{A}B\bar{C}D$ 5	$\bar{A}BCD$ 7	$\bar{A}BC\bar{D}$ 6
AB	$AB\bar{C}\bar{D}$ 12	$AB\bar{C}D$ 13	$ABCD$ 15	$ABC\bar{D}$ 14
$A\bar{B}$	$A\bar{B}\bar{C}\bar{D}$ 8	$A\bar{B}\bar{C}D$ 9	$A\bar{B}CD$ 11	$A\bar{B}C\bar{D}$ 10

Maxterm mode

AB \ CD	C+D	C+$\bar{D}$	$\bar{C}$+$\bar{D}$	$\bar{C}$+D
A+B	$\bar{A}$+$\bar{B}$+$\bar{C}$+$\bar{D}$ 0	A+B+C+$\bar{D}$ 1	A+B+$\bar{C}$+$\bar{D}$ 3	A+B+$\bar{C}$+D 2
A+$\bar{B}$	A+$\bar{B}$+C+D 4	A+$\bar{B}$+C+$\bar{D}$ 5	A+$\bar{B}$+$\bar{C}$+$\bar{D}$ 7	A+$\bar{B}$+$\bar{C}$+D 6
$\bar{A}$+$\bar{B}$	$\bar{A}$+$\bar{B}$+C+D 12	$\bar{A}$+$\bar{B}$+C+$\bar{D}$ 13	$\bar{A}$+$\bar{B}$+$\bar{C}$+$\bar{D}$ 15	$\bar{A}$+$\bar{B}$+$\bar{C}$+D 14
A+$\bar{B}$	$\bar{A}$+B+C+D 8	$\bar{A}$+B+C+$\bar{D}$ 9	$\bar{A}$+B+$\bar{C}$+$\bar{D}$ 11	$\bar{A}$+B+$\bar{C}$+D 10

Use the Code : BVREDDY ,to get the maximum discount

Number systems

- Base (b) is always a positive integer .
- In general b ≥ 0

Base	Different digits
2 (Binary)	0 , 1
8(Octal)	0,1,2,3,4,5,6,7
10 (Decimal)	0,1,2,3,4,5,6,7 ,8,9
16 (Hexadecimal)	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

r' s Complement

BVREDDY

r' s Complement of the number (N) $= r^n - N$

r --------> Radix

n -------> number of integer digits

N --------> given number

Use the Code : BVREDDY

(r-1) ' s Complement

(r-1) ' s Complement of the number (N) $= r^n - r^{-m} - N$

r --------> Radix

n -------> number of integer digits

m -------> number of decimal digits

N --------> given number

BVREDDY

(r-1) ' s Complement of the number (N) $= r^n - r^{-m} - N$

r' s Complement of the number (N) = (r-1)'s complement + r^{-m}

if m= 0

r' s Complement of the number (N) = (r-1)'s complement + 1

Unsigned Number Representation

BVREDDY

- Strictly applicable for positive numbers
- There is no sign bit concept

  + 5 ------> 101

  - 5 -----> not allowed
- Range = 0 to $2^n - 1$

Signed Magnitude representation

- Valid for both positive and negative numbers .
- Sign bit concept is used .

Sign bit = 0 , for ⊕Ve number

= 1, for ⊖ *ve number*

Range = - $(2^{n-1} - 1)$ to + $(2^{n-1} - 1)$

Use the Code : BVREDDY

1's Compliment representation

In this ⊕Ve numbers are represented as normal binary number with MSB '0'

BVREDDY

Representation of ⊖ *ve number*

1. Write the binary equivalent of magnitude
2. Take its 1's compliment

➢ Range = - $(2^{n-1} - 1)$ to $+(2^{n-1} - 1)$

Overflow

Over flow occurs in signed arithmetic operations if two same sign numbers are added and result exceeds with given number of bits . Overflow can be avoided by taking extra bits

1.By using carry bits

C_{in} ⟶ *carry into MSB*

C_{out} ⟶ *carry out from MSB*

if $C_{in} \oplus C_{out} = 0$, no overflow occurs

$C_{in} \oplus C_{out} = 1$, over flow occurs

2. By using Sign Bits

X ----------> Sign bit of 1st number

Y ----------> Sign bit of 2nd number

Z----------> Sign bit of Resultant

BVREDDY

Over flow = $XY\bar{Z} + \bar{X}\bar{Y}Z$

2's Compliment representation

In this ⊕Ve numbers are represented as normal binary number with MSB '0'

Representation of ⊖ *ve number*

1. Write the binary equivalent of magnitude
2. Take its 2's compliment

➢ Range = - (2^{n-1}) to $+(2^{n-1} - 1)$

BCD (Binary Coded Decimal)Code

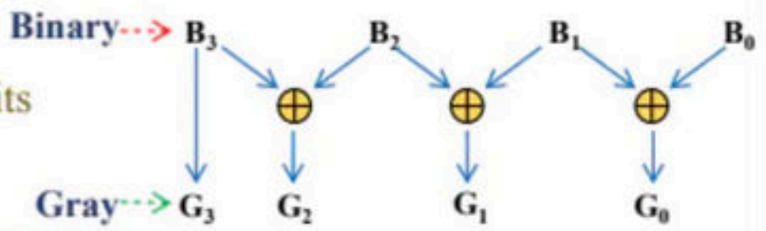
In this code each decimal number is represented by a separate group of 4- bits

➢ It uses only 0 to 9

➢ 0 to 9 are valid BCD Code

➢ 10, 11, 12 , 13 , 14 ,15 are invalid BCD Code

➢ Coding method is very simple but it requires more number of bits .

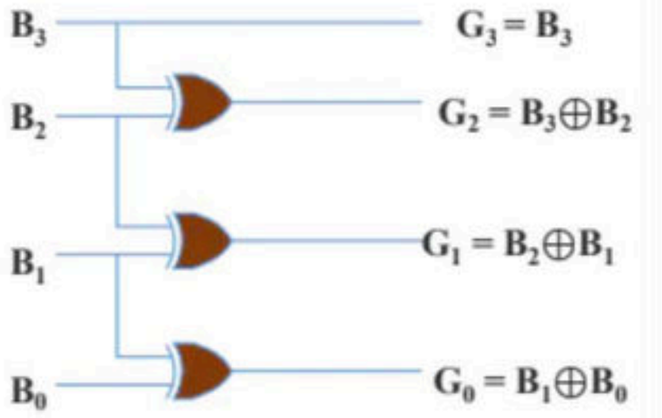


EX-3 Code

The EX-3 code can be derived from the natural BCD code by adding 3 to each coded number

Valid EX -3 : 3 ,4,5,6,7,8,9,10,11,12

Invalid EX-3 : 0,1,2,13,14,15



Gray Code

➢ Non weighted code

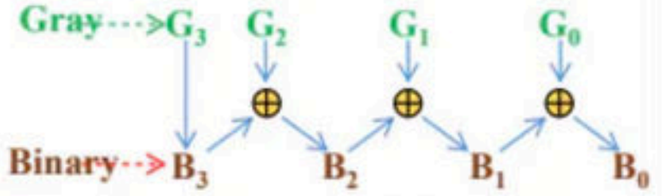
➢ Unit distance code

➢ Cyclic code

➢ Reflective code

➢ Minimum error code

BVREDDY

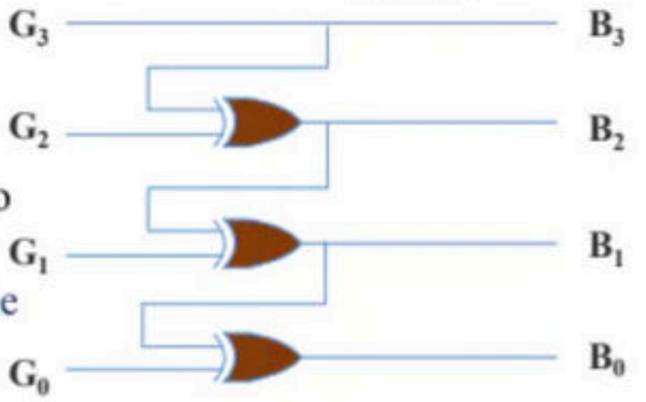


BVREDDY

SELF COMPLEMENTING CODE

A code is said to be self complementing, if the 1' complement of a number N is equal to the 9's complement of the number.

➢ For a code to be self complementing, the sum of all its weights must be 9 .



2421	5211	4311	3321	EX-3

BVREDDY

HA

1. Logical expression for Sum = $A \oplus B$
2. Logical expression for Carry = AB
3. Minimum number of NAND Gates = **5**
4. Minimum number of NOR Gates = **5**

FA

1. Logical expression for Sum = $A \oplus B \oplus C$
2. Logical expression for Carry = $AB+(A \oplus B)C$
3. Minimum number of NAND Gates = 9
4. Minimum number of NOR Gates = 9

HS

1. Logical expression for Difference = $A \oplus B$
2. Logical expression for Barrow = $\bar{A}B$
3. Minimum number of NAND Gates = **5**
4. Minimum number of NOR Gates = **5**

FS

1. Logical expression for Difference= $A \oplus B \oplus C$
2. Logical expression for Barrow = $\bar{A}B + (\overline{A \oplus B})C$
3. Minimum number of NAND Gates = 9
4. Minimum number of NOR Gates = 9

Half Adder

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Half Subtractor

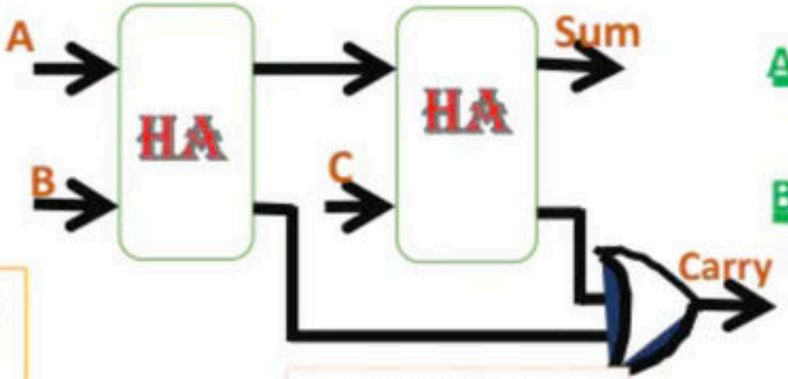
A	B	Difference	Barrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Full Adder

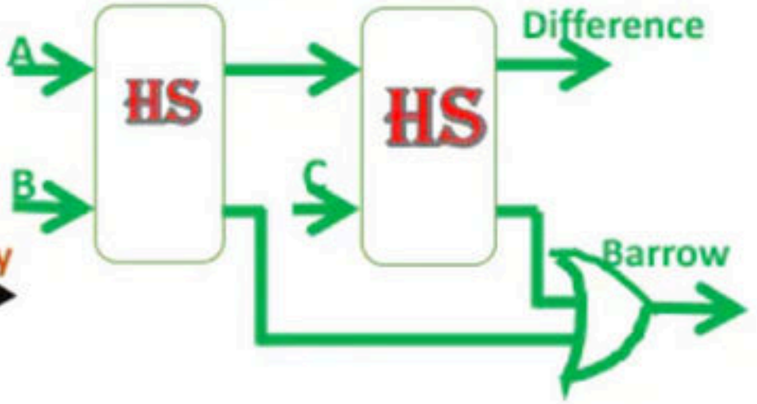
A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Full Subtractor

A	B	C	Difference	Barrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



BVREDDY



BVREDDY

Use the Code : BVREDDY

FS : A- B- C

Difference = $A \oplus B \oplus C$

Barrow = $\bar{A}B + (\overline{A \oplus B})C$

= $\bar{A}B + \bar{A}C + BC$

FS : B- C- A

Difference = $A \oplus B \oplus C$

Barrow = $\bar{B}C + (\overline{B \oplus C})A$

= $A\bar{B} + \bar{B}C + AC$

FS : C- A- B

Difference = $A \oplus B \oplus C$

Barrow = $\bar{C}A + (\overline{C \oplus A})B$

= $A\bar{C} + B\bar{C} + AB$

Binary Multiplier

Number of AND gates required = m× n

Number of Adders required = m+n-2

m----------> number of bits in A

n ----------->number of bits in B

In general for n- bit Parallel Adder

Worst case Delay = (n-1)$\left((t_{pd})\, carry \right.$ + $Max(sum, carry)$

Look Ahead Carry Adder

- In this adder ,the carry dependency of Ripple Carry Adder (RCA) is eliminated
- This is the fastest adder among all
- This adder have the maximum complexity

BVREDDY

Hardware Requirements

L1 : n- XOR + n- AND

L2 : $\frac{n(n+1)}{2}$ − AND

L3 : n- OR

carry

L4 : n- XOR

sum

Total number of gates for carry = $3n + \frac{n(n+1)}{2}$

Total number of gates for sum = $4n + \frac{n(n+1)}{2}$

Worst delay for Carry = Max(xor ,and)+ $(t_{pd})_{and}$ + $(t_{pd})_{or}$

Worst case delay for Sum = Max(xor ,and)+ $(t_{pd})_{and}$ + $(t_{pd})_{or}$ + $(t_{pd})_{xor}$

BVREDDY

For n- bit Magnitude Comparator

Total number of input combinations = 2^{2n}

Lesser than combinations = $\frac{2^{2n} - 2^n}{2}$

Greater than combinations = $\frac{2^{2n} - 2^n}{2}$

Equal combinations = 2^n

For 3- bit magnitude comparator

$Y_1(A<B) = \bar{a}_2 b_2 + (a_2 \odot b_2)\bar{a}_1 b_1 + (a_2 \odot b_2)(a_1 \odot b_1)\bar{a}_0 b_0$

$Y_2(A=B) = (a_2 \odot b_2)(a_1 \odot b_1)(a_0 \odot b_0)$

$Y_3(A>B) = a_2 \bar{b}_2 + (a_2 \odot b_2)a_1 \bar{b}_1 + (a_2 \odot b_2)(a_1 \odot b_1)a_0 \bar{b}_0$

BVREDDY

For 4-bit Magnitude Comparator

$Y_1(A<B) = \overline{a_3} b_3 + (a_3 \odot b_3)(\overline{a_2} b_2) + (a_3 \odot b_3)(a_2 \odot b_2)(\overline{a_1} b_1) + (a_3 \odot b_3)(a_2 \odot b_2)(a_1 \odot b_1)(\overline{a_0} b_0)$

$Y_2(A=B) = (a_3 \odot b_3)\,(a_2 \odot b_2)\,(a_1 \odot b_1)\,(a_0 \odot b_0)$

$Y_3(A>B) = a_3 \overline{b_3} + (a_3 \odot b_3)(a_2\, \overline{b_2}) + (a_3 \odot b_3)(a_2 \odot b_2)(a_1 \overline{b_1}) + (a_3 \odot b_3)(a_2 \odot b_2)\,(a_1 \odot b_1)(a_0 \overline{b_0})$

Multiplexer (MUX)

- Data selector
- Many to one
- Universal logic gate
- Parallel to serial converter

$2^n \times 1$

2^n ----------> number of data inputs

n ----------> number of select inputs

1 ----------> number of outputs

BVREDDY

Demultiplexer

- One input to many output
- Data distributor
- One to many circuit

1×2^n

n---------> number of select lines

2^n --------> number of output lines

1 ----------> number of inputs

BVREDDY

Decoder

Decoder is a multi input ,multi output logic circuit which coverts coded input into coded output , where the input and output codes are different

$n \times 2^n$

n ----------> number of inputs

2^n ----------> number of outputs

Decoder is a special case of Demultiplexer , in which the select lines or Demultiplexer are treated as input's to the decoder and input of Demultiplexer is treated as Enable input of the Decoder

Inputs ⟺ Enable

Select lines ⟺ Inputs

Logic Gate	Number of MUX required
BUFFER	1
NOT	1
AND	1
OR	1
NAND	2
NOR	2
EX-OR	2
EX-NOR	2
HA	3
HS	2

Encoder

Encoder is a combinational circuit , which is used to convert

1. Octal to binary (8 × 3 encoder)
2. Decimal to Binary (10× 4 encoder)
3. Hexadecimal to Binary (16 × 4 encoder)

$2^n X\ n$

n ----------> number of outputs

2^n ----------> number of inputs

- For an Encoder at a time only one among the all inputs is high , reaming all inputs should be zero
- If multiple inputs are simultaneously high, then the output is not valid, to avoid this restriction we will go for priority encoder.

1. By using one 4 × 1 Mux → All 2 variable functions; Some but not All 3 variable functions

BVREDDY

2. By using one 4 × 1 Mux + NOT Gate → All 2 variable functions; All 3 variable functions

3. By using one 8 × 1 Mux → All 3 variable functions; Some but not All 4 variable functions

4. By using one 8 × 1 Mux + NOT Gate → All 3 variable functions; All 4 variable functions

5. n- variable function → **One $2^n \times 1$ MUX**; **One $2^{n-1} \times 1$ MUX + one NOT Gate**

BVREDDY

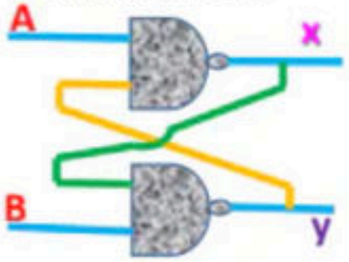
Sequential Circuits

The logic circuit whose outputs at any instant of time depends on the present inputs as well as on the past outputs are called sequential circuits, in sequential circuits ,the output signals are fed back to the input side .

BVREDDY

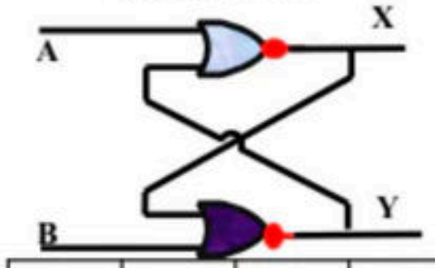
- Out put of combinational circuit depends on input combinations .
- Output of sequential circuits depends on input sequence.
- For unequal delay of gates also the operation is valid

NAND LATCH



A	B	X	Y
0	0	1	1
0	1	0	1
1	0	1	0
1	1	Memory	

NOR LATCH



A	B	X	Y
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Memory	

For **SR NAND** latch , if the input sequence is
00 --------> 11 , then the following cases arises

BVREDDY

- If the delay of both gates are same then we don't have any stable output , the output is oscillatory , this condition is known as critical race
- However if the delay of both gates are not equal then there exist a stable output , but it depends on the individual delay of the gates

For **SR NOR** latch , if the input sequence is
11 --------> 00 , then the following cases arises

- If the delay of both gates are same then we don't have any stable output , the output is oscillatory , this condition is known as critical race .
- However if the delay of both gates are not equal then there exist a stable output , but it depends on the individual delay of the gates .

FLIP FLOP

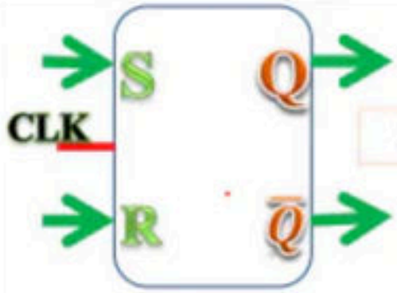
In a latch the output changes immediately in response to external input , so to have an additional control , we are introducing a signal called " ***CLOCK*** " , whose purpose is same as Enable pin of Decoder.

Latch +Clock = Flip Flop

Latches are universally not unique and hence their truth tables are not unique .

Flip Flops are universally unique , and their truth tables are unique .

BVREDDY



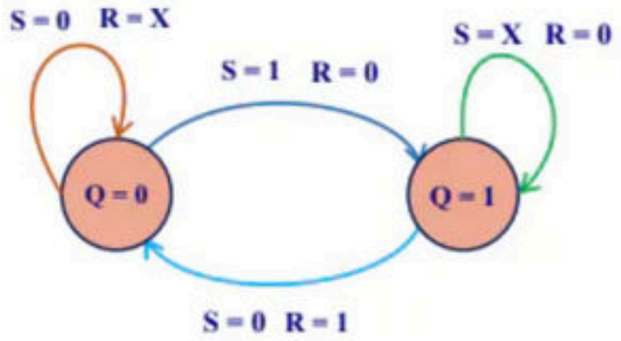
BVREDDY

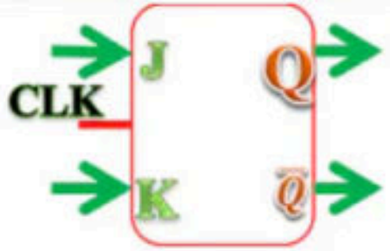
$$Q^+ = S + \bar{R}Q$$

CLK	S	R	Q+	State
0	×	×	Q	Memory
1	0	0	Q	Memory
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	×	Invalid

CLK	S	R	Q	Q+
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	×
1	1	1	1	×

Q	Q+	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0



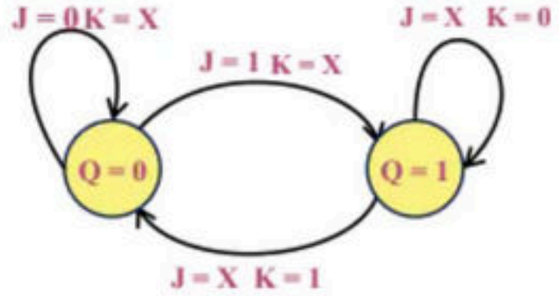


$$Q^+ = J\bar{Q} + \bar{k}Q$$

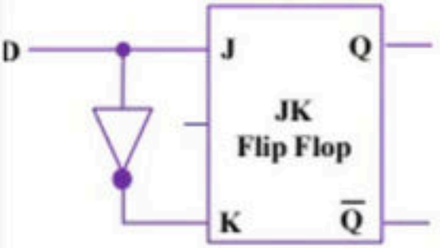
CLK	J	K	Q+	State
0	×	×	Q	Memory
1	0	0	Q	Memory
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	$\bar{Q}$	Toggle

CLK	J	K	Q	Q+
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Q	Q+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0



D Flip Flop



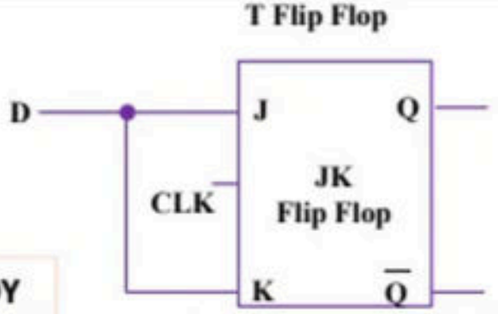
$Q^+ = D$

CLK	D	Q+
0	X	Hold
1	0	0
1	1	1

CLK	D	Q	Q+
0	X	Q	Q
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

BVREDDY

T Flip Flop



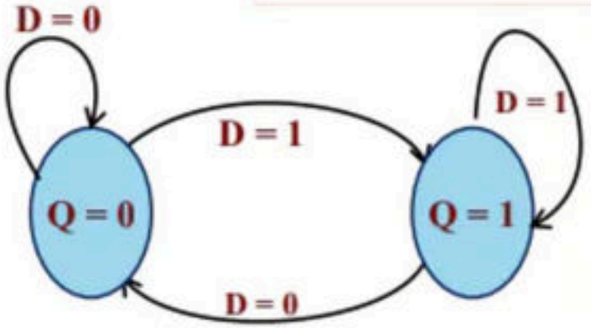
$Q^+ = T \oplus Q$

CLK	T	Q+
0	X	Hold
1	0	Q
1	1	Toggle

CLK	T	Q	Q+
0	X	Q	Q
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

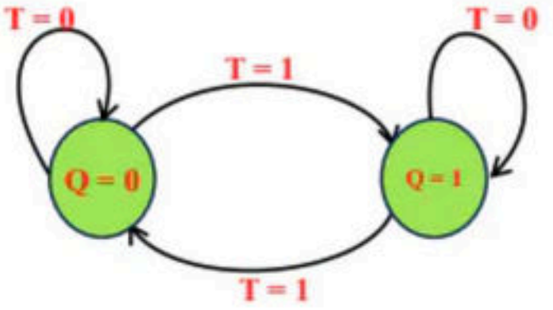
BVREDDY

Q	Q+	D
0	0	0
0	1	1
1	0	0
1	1	1



Use the Code : BVREDDY

Q	Q+	T
0	0	0
0	1	1
1	0	1
1	1	0

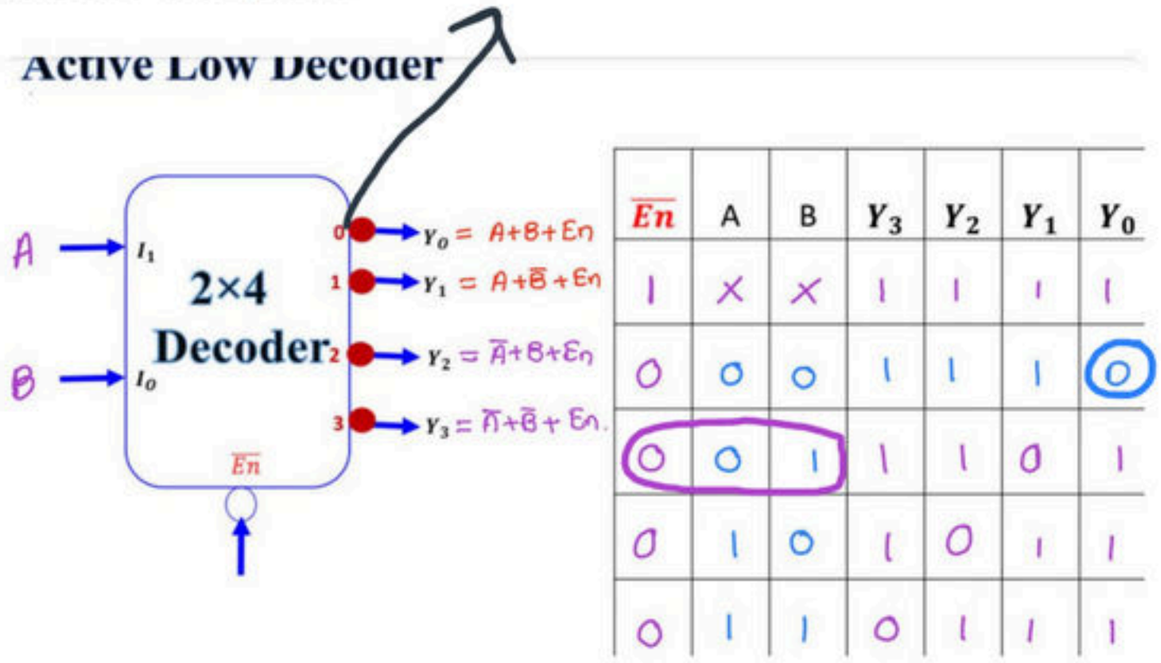


BVREDDY

unacademy

▲ 1 • Asked by Thomas

this one also pleae



$\overline{En}$	A	B	Y_3	Y_2	Y_1	Y_0
1	X	X	1	1	1	1
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1

$Y_0 = (A+B+\overline{En})$

$\overline{En} = 0 \quad A=0 \; B=1.$

$Y_0 = 0+1+1 = 1$

$Y_1 =$

BVREDDY

Race Around Condition

The output of the FF changes to 0 →1 →0 …. Continuously at the starting of the next clock the output is uncertain , which is called as Race Around Condition (RAC)

RAC occurs in any FF if the following conditions satisfies

1. If the FFs are operated in level triggering
2. if (tpd) < (Tclk)on ,
3. If the FFs are operated in Toggle mode

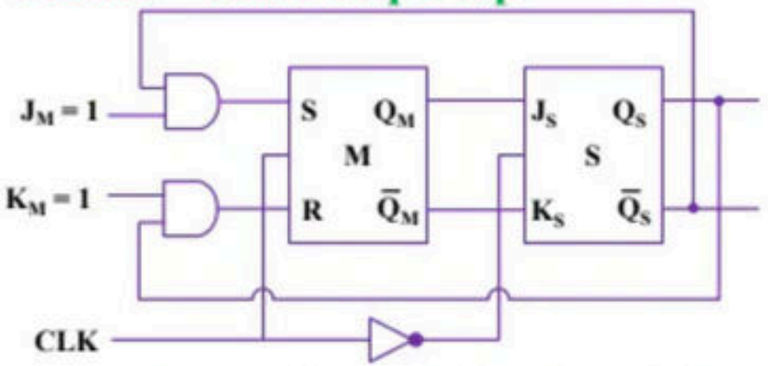
If the above 3 conditions satisfies simultaneously then there is a continuous race in the output of the FF between 0 and 1 to reach the next state , who will be the winner of the race in not certain , that depends on tpd and (Tclk) on .

Remedy

1. (Tclk)on < (tpd) < T
2. By using Edge triggered FF
3. By using Master Slave FF

BVREDDY

Master – Slave Flip Flop



1. In case of Master Slave configuration , Master is applied with input clock and Slave is applied with inverted clock , so out of two FFs at a time only one of the FF respond and other will not respond . As a result, Many times toggling in a single clock cycle has been converted to one time toggle , hence *RAC is avoided* .
2. In Master Slave configuration , command signal is generated by master FF and the response of the command signal is given by slave FF
3. Master slave FF can store 1 – bit of data

BVREDDY

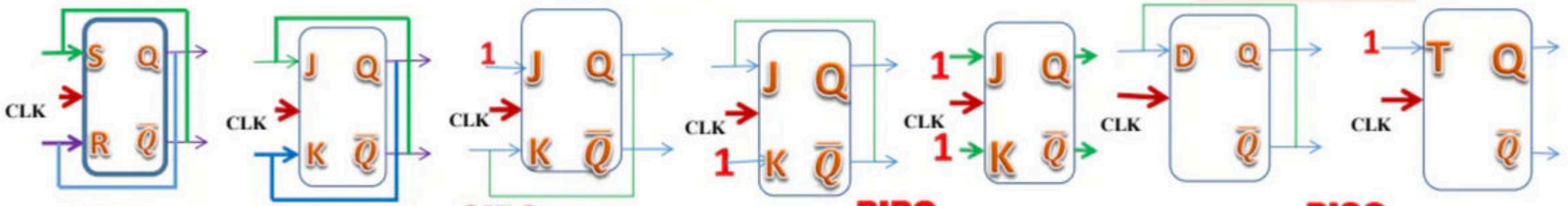
CONVERSATION of FLIP FLOP

BVREDDY

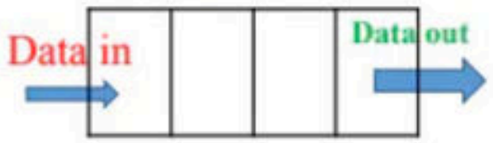
JK to SR J = S K = R	**SR to JK** $S = J\bar{Q}$ $R = KQ$	**D to SR** $D = S+\bar{R}Q$	**T to SR** $T = S\bar{Q}+RQ$
JK to D J = D $K = \bar{D}$	**SR to D** S = D $R = \bar{D}$	**D to JK** $D = J\bar{Q}+\bar{K}Q$	**T to JK** $T = J\bar{Q}+KQ$
JK to T J = T K = T	**SR to T** $S = T\bar{Q}$ $R = TQ$	**D to T** $D = T \oplus Q$	**T to D** $T = D \oplus Q$

BVREDDY

Toggle Modes



SISO

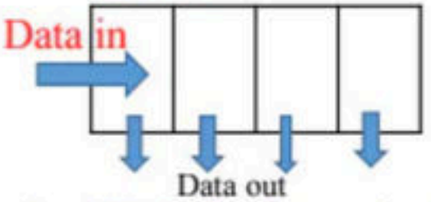


- SISO Configuration has only
  - 1- input
  - 1- output
- For SISO configuration

for storing = (n) CP
for retrieving = (n-1) CP
Total number clock pulses = 2n-1

BVREDDY

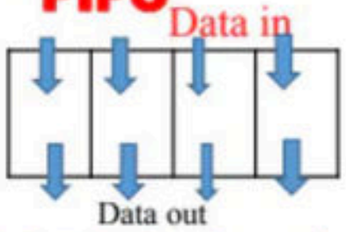
SIPO



- SIPO Configuration has only
  - 1- input
  - 4- output
- For SIPO configuration

for storing = (n) CP
for retrieving = 0 CP
Total number clock pulses = n

PIPO



- PIPO Configuration has only
  - 4- input
  - 4- output
- For PIPO configuration

for storing = 1 CP
for retrieving = 0 CP
Total number clock pulses = 1

PISO

- PISO Configuration has only
  - 4- input
  - 1- output
- For PISO configuration

for storing = 1 CP
for retrieving = (n-1)CP
Total number clock pulses = n

Counters

State of a Counter : Any possible output of a counter is known as its state , for a n – bit counter the maximum possible states are 2^n
The states which are counted by the counter are called as *valid states* , and the states which are not counted (skipped) by the counter are called as invalid states .

Modulus of a Counter : The minimum number of clocks needed to get the counting pattern repeats is called as Modulus of a counter

BVREDDY

Design equation of a counter

$$2^n \geq N$$

$$n \geq log_2 N$$

n----> number of Flip Flops
N-----> MOD no. of a counter

BVREDDY

ASYNCHRONOUS COUNTER

BVREDDY

- Different FFs are applied with different clocks
- For only one FF external clock is applied ,which is LSB and output of one FF will acts as clock to next FFs
- FFs are operated in toggle mode
- Fixed counting sequence
  1. up counter
  2. down counter

ये वक्त भी गुजर जाएगा

This time will also pass

- ⊖ve Edge trigger and Q as a clock --------> Up counter
- ⊖ve Edge trigger and $\bar{Q}$ as a clock --------> Down counter
- ⊕ve Edge trigger and Q as a clock --------> Down counter
- ⊕ve Edge trigger and $\bar{Q}$as a clock --------> Up counter
- The disadvantages of the ripple counter is that transition states are present due to delay of the FF (Decoding errors) .
- If only one FF changes its state ,then no transition states will be present , if more than one FF changes its states than transition states present.

BVREDDY

- To avoid decoding errors strobe signal is used .
- Strobe signal is kept low for 3tpd , for 3- bit counter , so that transition states are not reflected, and after 3tpd strobe signal is made high .
- If delay each FF is t_{pd} , *then*

$$T_{CLK} \geq n\, t_{pd}$$

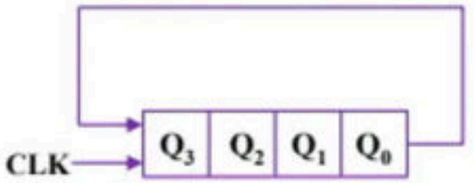
$$f_{CLK} \leq \frac{1}{t_{pd}}$$

Use the Code : BVREDDY

RING COUNTER

- Ring counter is a synchronous counter , it is a shift register in which last FF output is connected to the first FF input .
- In ring counter only one FF output is logic '1 ' and it will rotate with clock .
- Ring counter performs right shift operation .

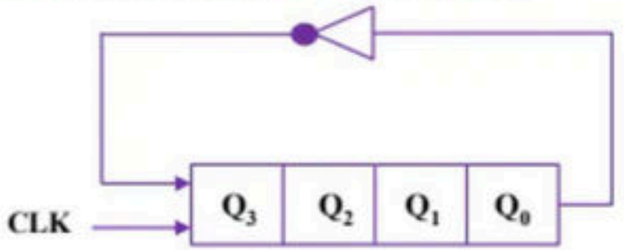
BVREDDY



- Decoding logic of ring counter is simple and does not require any external logic circuit
- If all the outputs of FFs initially zero , then the Ring counter does not start .
- If more than one FF outputs' are high initially, then the ring counter enters into unused state and never come out of unused state , this is called as **Lock out problem** .

BVREDDY

JOHNSON RING COUNTER



Johnson Ring counter

Twisted Ring counter

Switch tail counter

Walking Counter

Creeping counter

Mobies counter

Ring counter	Johnson ring counter
1. Mod No = n BVREDDY	1. Mod No = 2n
2. Number of used states= n Number of unused states = $2^n - n$	2. Number of used states= 2n Number of unused states =$2^{2n} - n$
3.Time period of each FF = n(T_{CLK})	3.Time period of each FF = 2n(T_{CLK})
4. Frequency of each FF = $\frac{f_{clk}}{n}$	4. Frequency of each FF = $\frac{f_{clk}}{2n}$
5. Suffer from lock out problem	5. Suffer from lock out problem
6.Decoding logic is simple	6.Decoding logic requires AND and NOR gates

FINITE STATE MACHINE

Synchronous Sequential circuits are also called as Finite State Machine (FSM)

There are two types of FSMs

1. Mealy State Machine

- The output of Mealy State Machine is a function of present state as well as present input
- to detect n – bit sequence by using Mealy modal n number of states are required

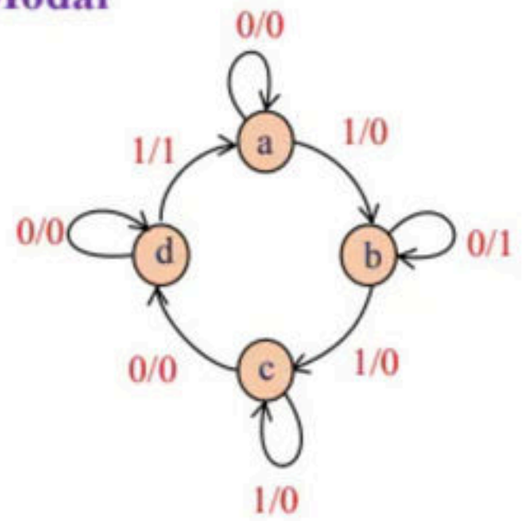
BVREDDY

2. Moore State Machine

- The output of Moore State Machine is a function of present state only
- To detect **n – bit sequence** by using Moore modal **(n+1)** number of states are required

Mealy Modal

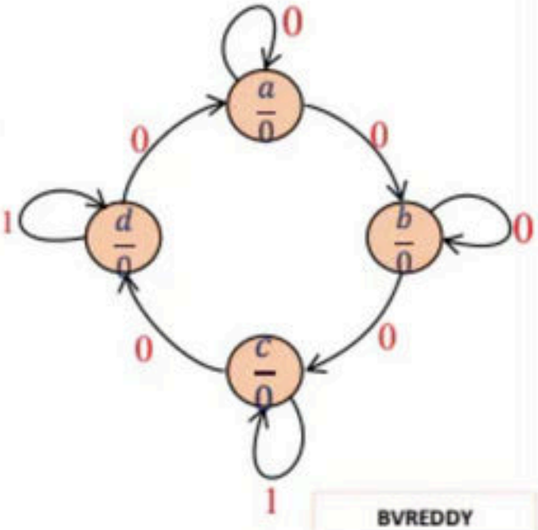
Present state	NS , O/P	
	X =0	X= 1
a	a , 0	b , 0
b	b, 1	c,0
c	d, 0	c , 0
d	d, 0	a , 1



BVREDDY

Moore Modal

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

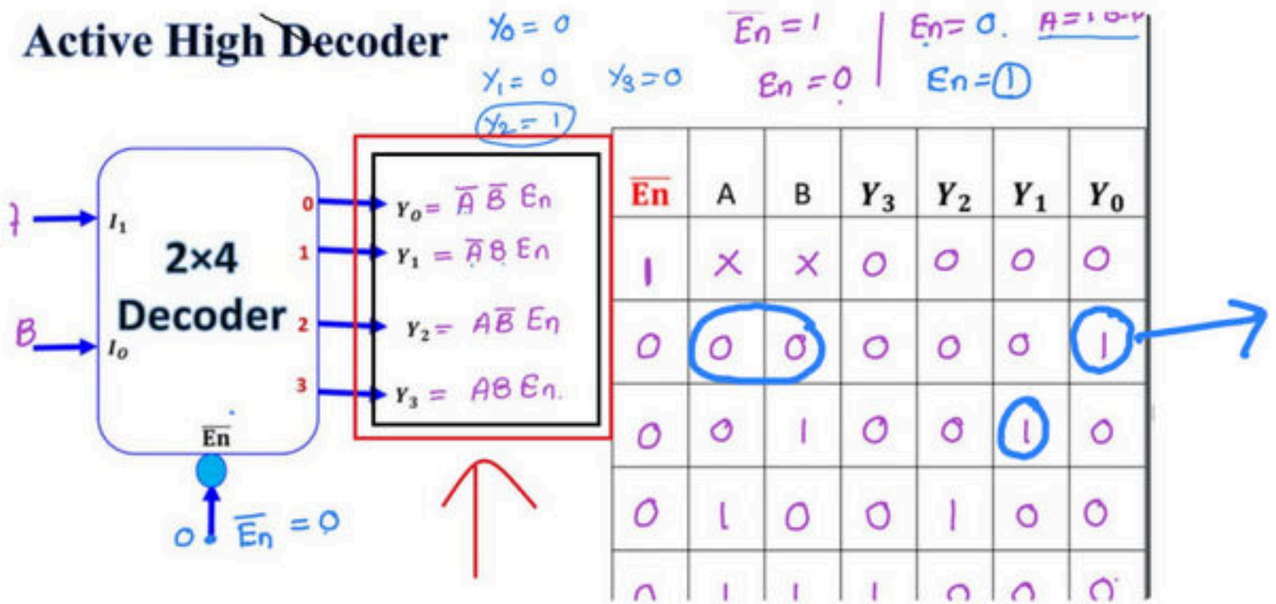


BVREDDY

unacademy

▲ 1 • Asked by Thomas

sir please explain equation of y0, y1, y2 and y3



$Y_0 = 0$ $Y_1 = 0$ $Y_3 = 0$ $Y_2 = 1$

$\overline{En} = 1$ $En = 0$ | $En = 0$. $En = 1$

$\overline{En}$	A	B	Y_3	Y_2	Y_1	Y_0
1	X	X	0	0	0	0
0	0	0	0	0	0	1
0	0	1	0	0	1	0
0	1	0	0	1	0	0
0	1	1	1	0	0	0

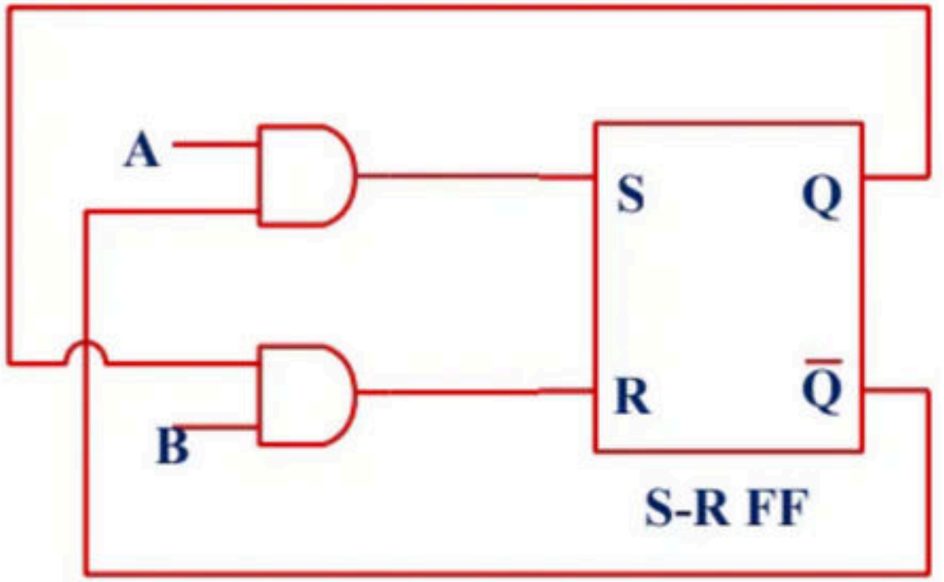
$Y_0 = \overline{A}\,\overline{B}\,En$

$Y_1 = \overline{A}\,B\,En.$

Q.The two inputs A and B are connected to an R-S FF via two AND gates as shown in the figure. If A= 1 and B = 0, the output $Q\bar{Q}$ is

(A) 00 (B) 10

(C) 01 (D) 11



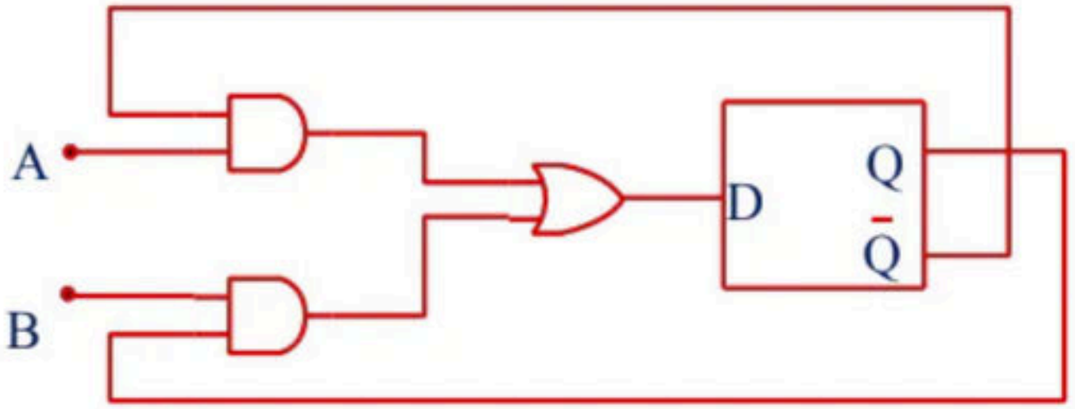
Q. What is represented by the digital circuit given above?

(a) An SR flip-flop with A=S and B=R

(b) A JK flip-flop with A=k and B=J

(c) A JK flip-flop with A=J and B=$\overline{K}$

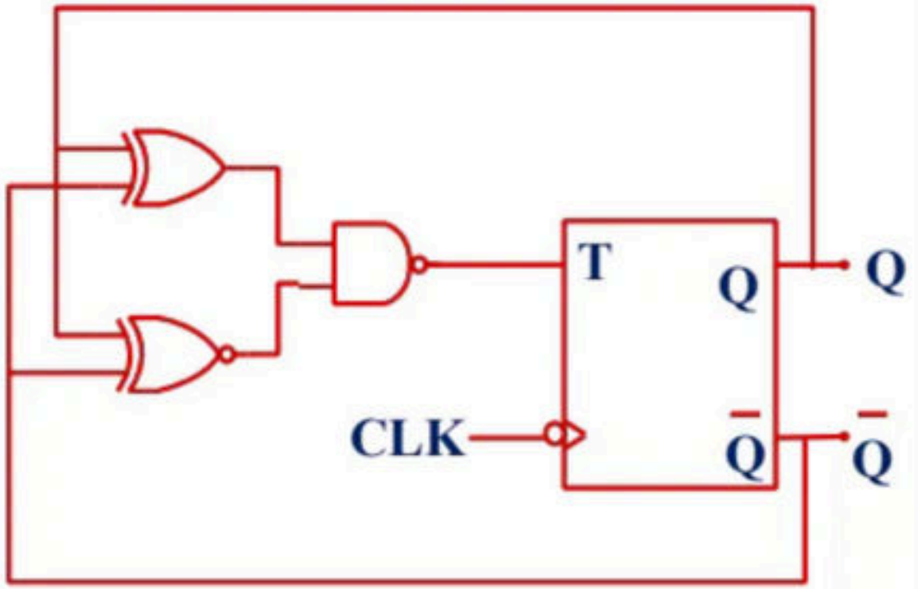
(d) An SR flip-flop with A=R and B=S



Q. The clock frequency applied to the digital circuit shown in the figure below is 1kHz. If the initial state of the output of the flip-flop is 0, then the frequency of the output waveform Q in kHz is

(A) 0.25 (B) 0.5

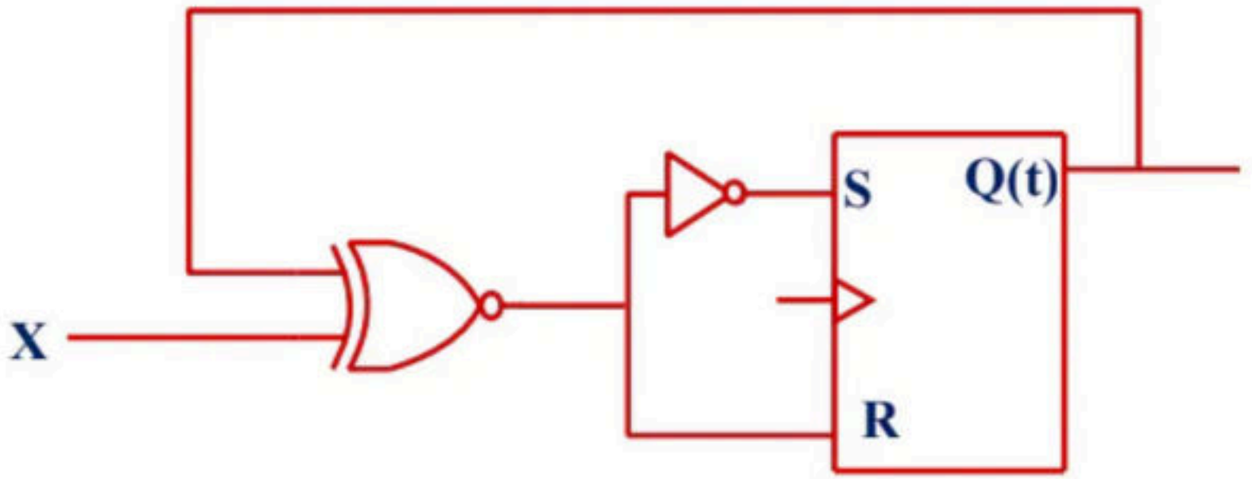
(C) 1 (D) 2



Q. Consider the circuit shown in the figure. The expression for the next state Q (t+1) is

(a) $xQ(t)$ (b) $x \oplus Q(t)$

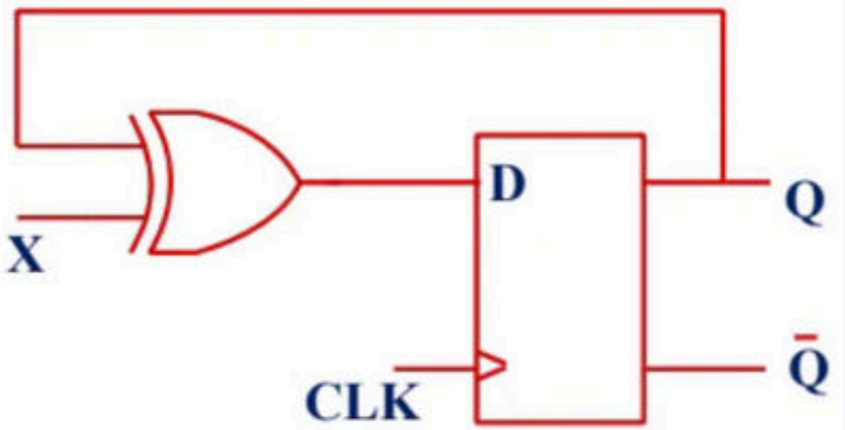
(c) $x\overline{Q}(t)$ (d) $x \odot Q(t)$



Q. The digital circuit shown in figure works as a

(A)JK flip-flop (B) Clocked RS flip-flop

(B)T flip-flop (D) Ring counter



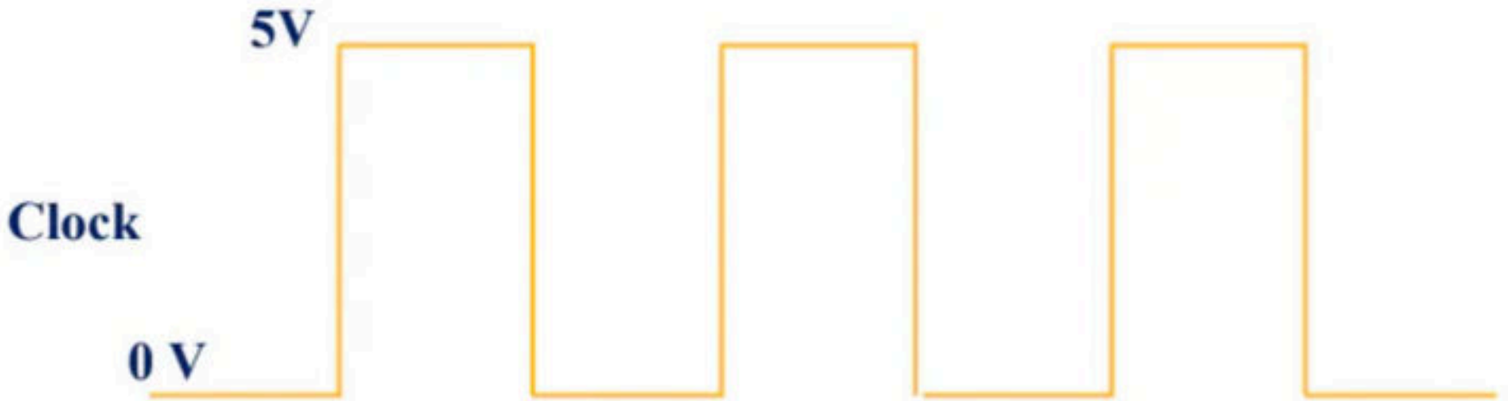
Triggering

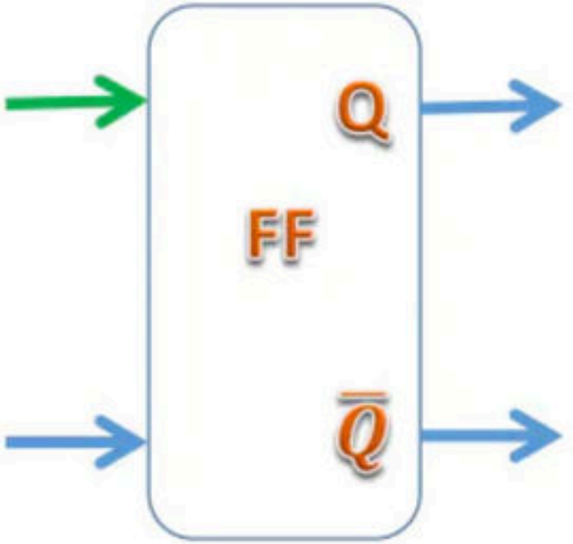
The momentary change in control input of a flip flop to switch it from one state to the other state is called Trigger and the transition it causes is said to trigger the flip flop . The process of applying the control signal to change the state of flip flop is called triggering.

There are 4- types of triggering the flip flops .

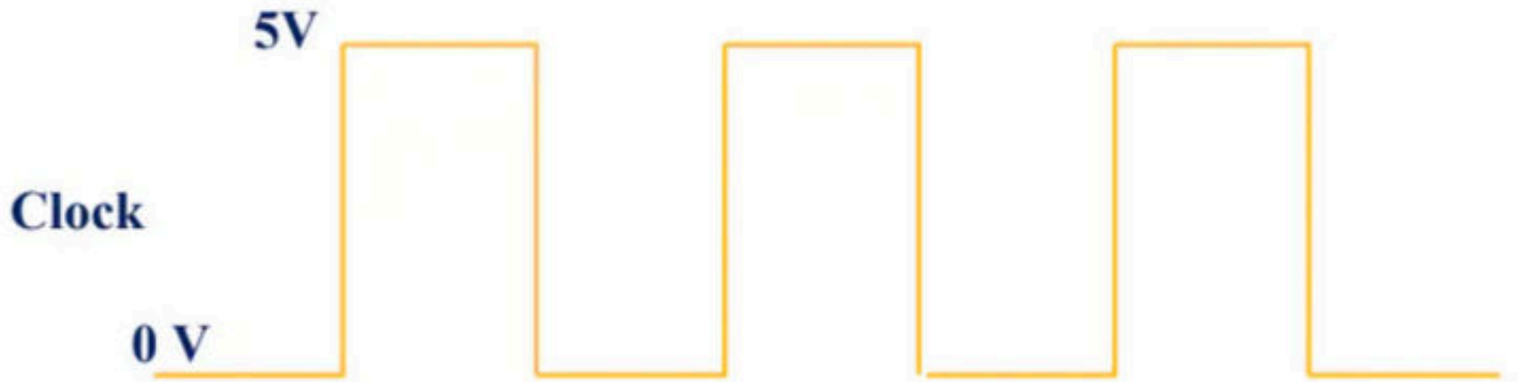
1. Positive level triggering
2. Negative level triggering
3. Positive Edge triggering
4. Negative Edge triggering

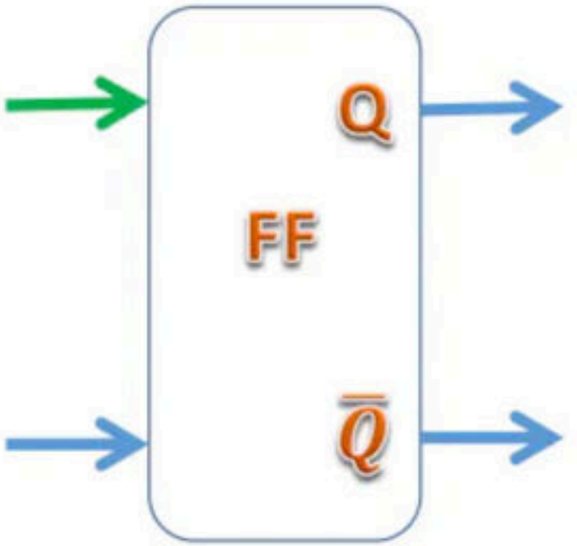
1. High (Positive) Level Triggering



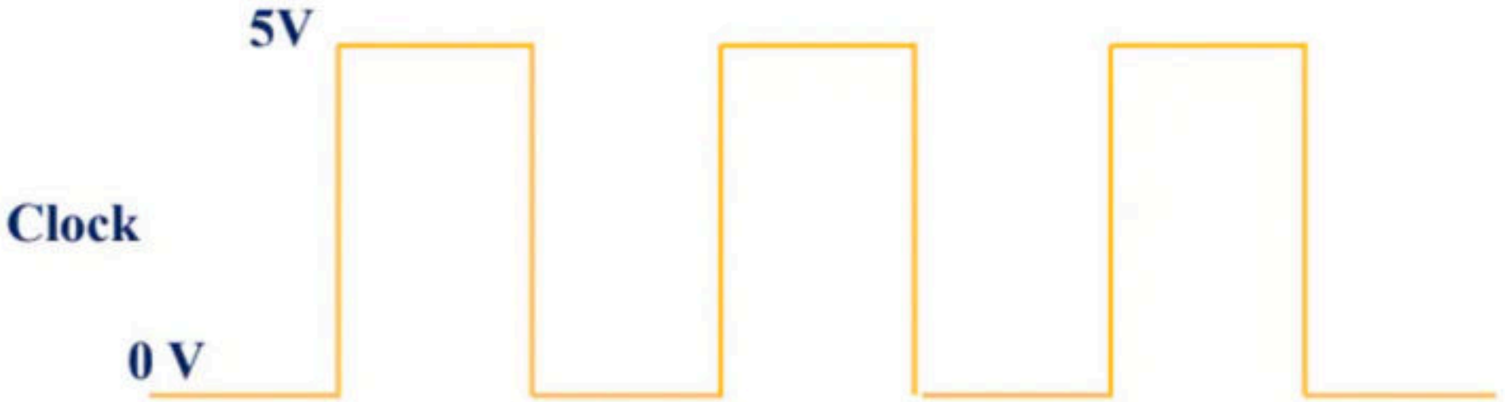


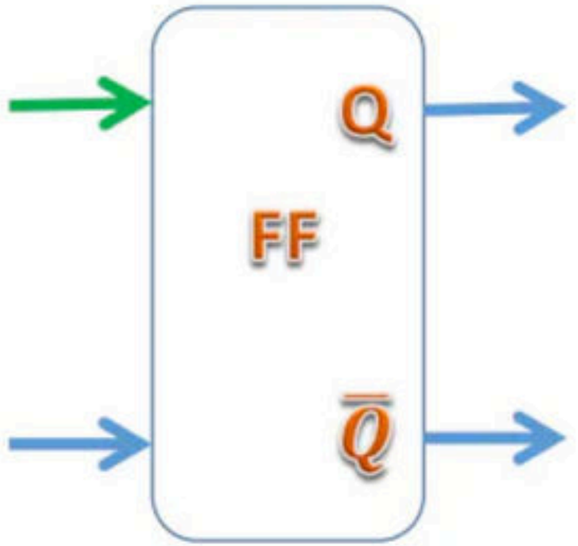
2. Low (Negative) Level Triggering



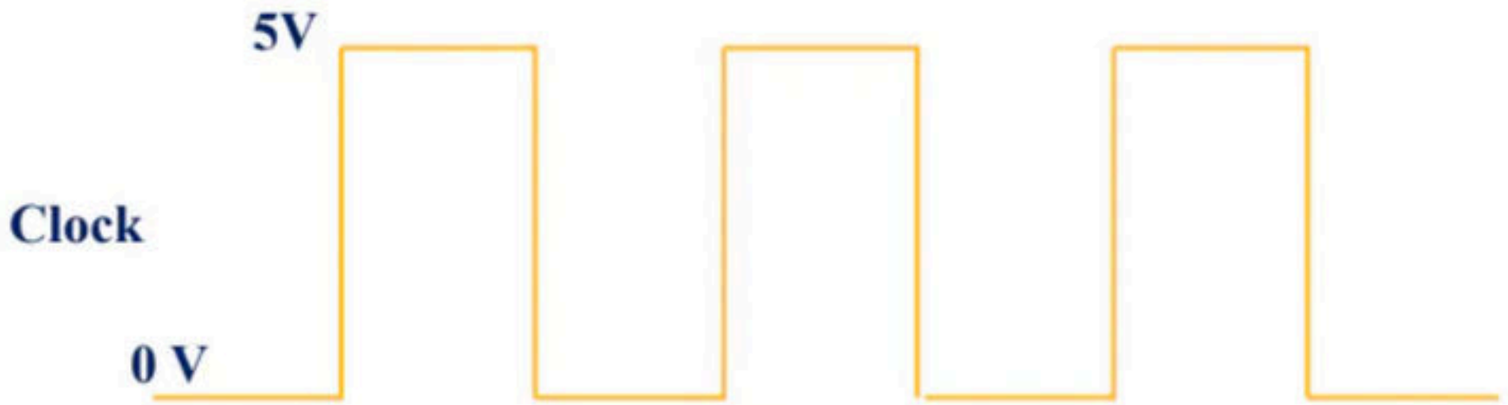


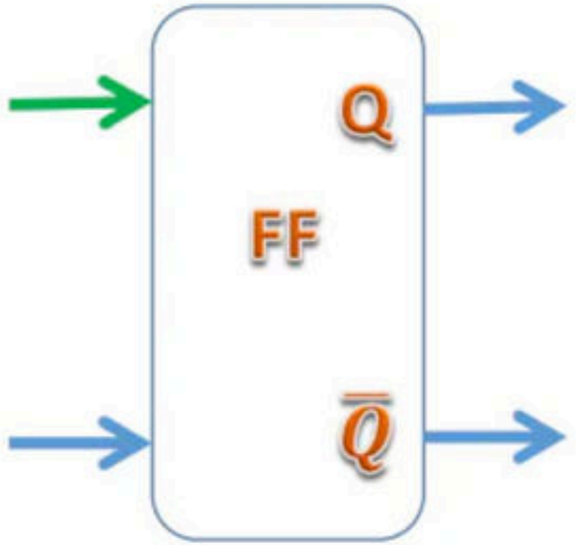
3. Positive Edge triggering



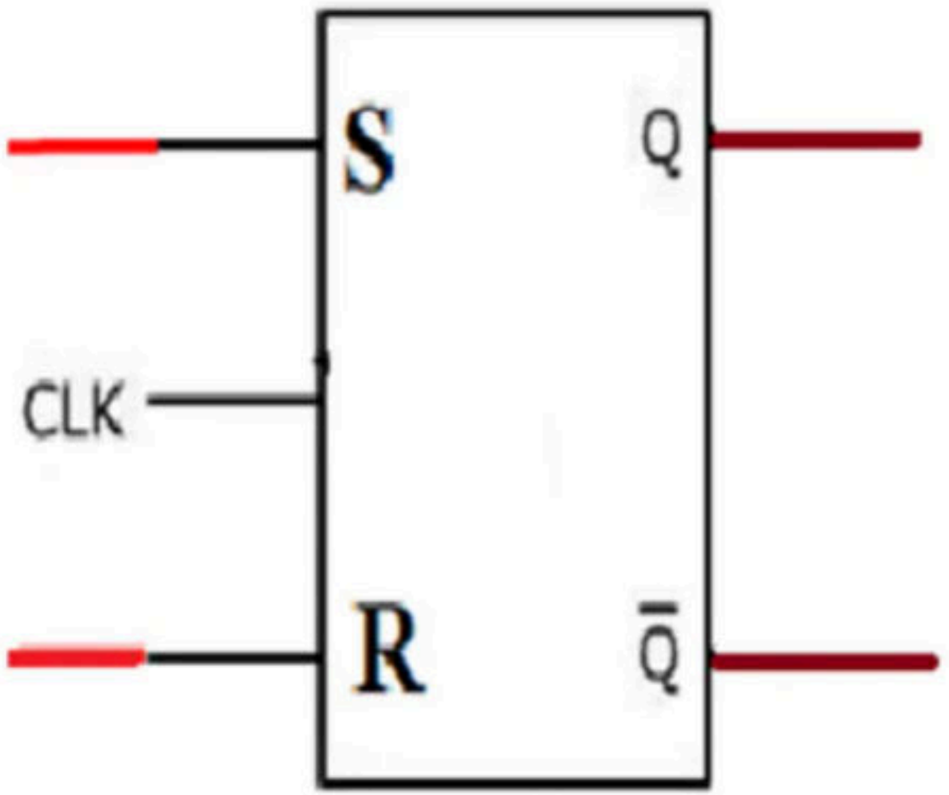


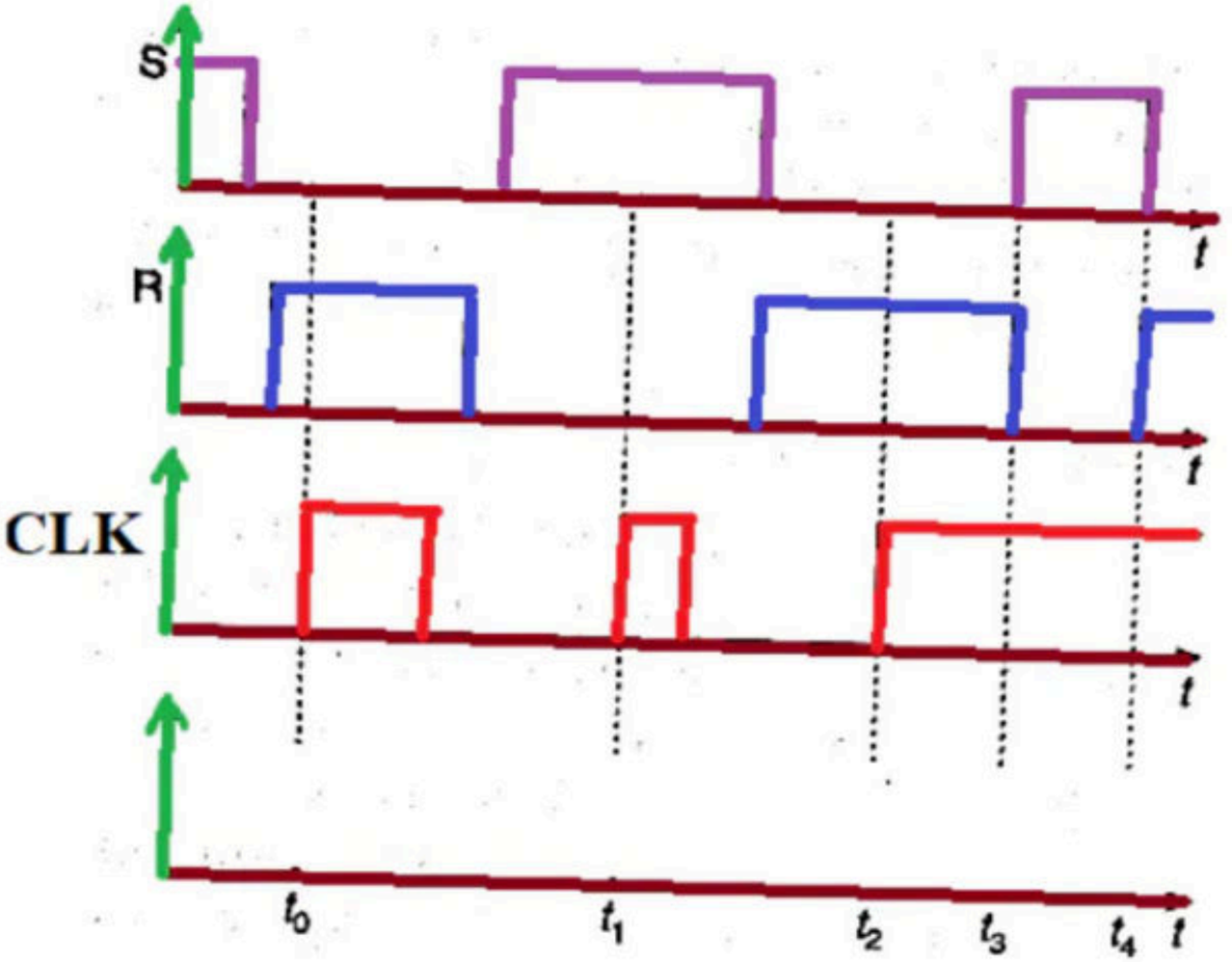
4. Negative Edge triggering



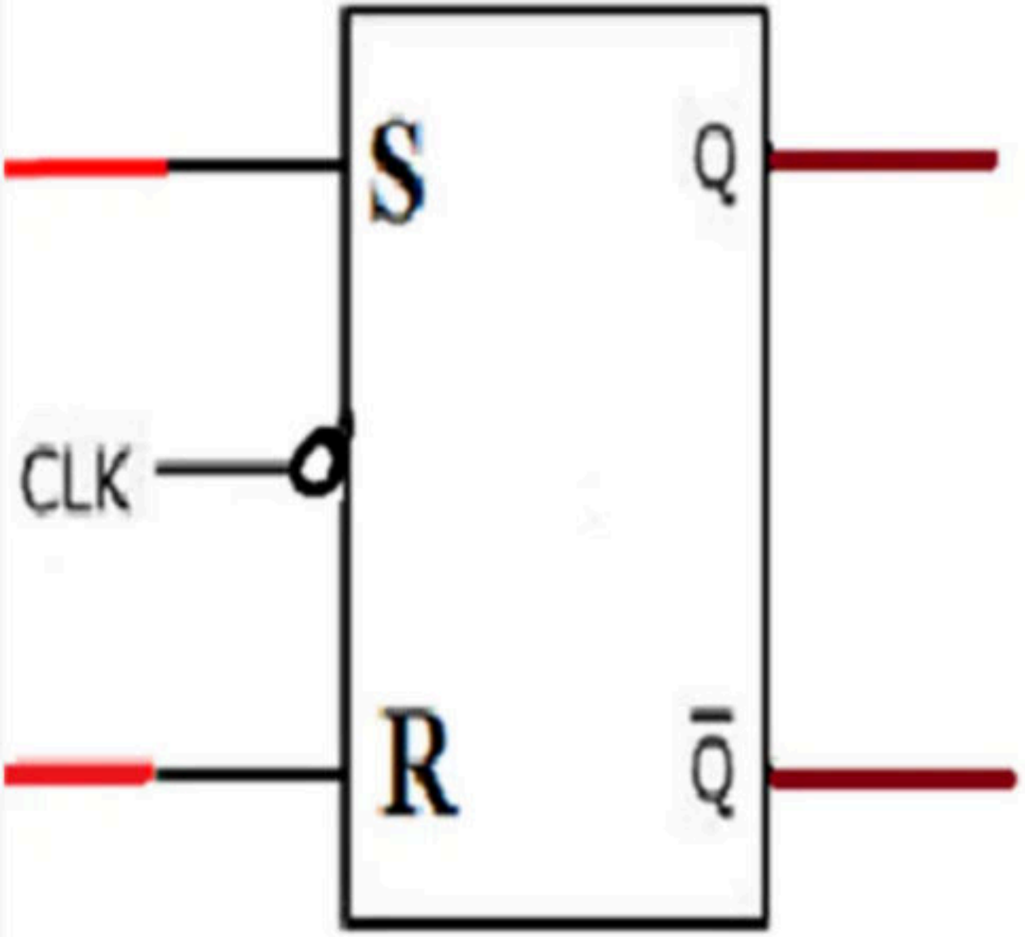


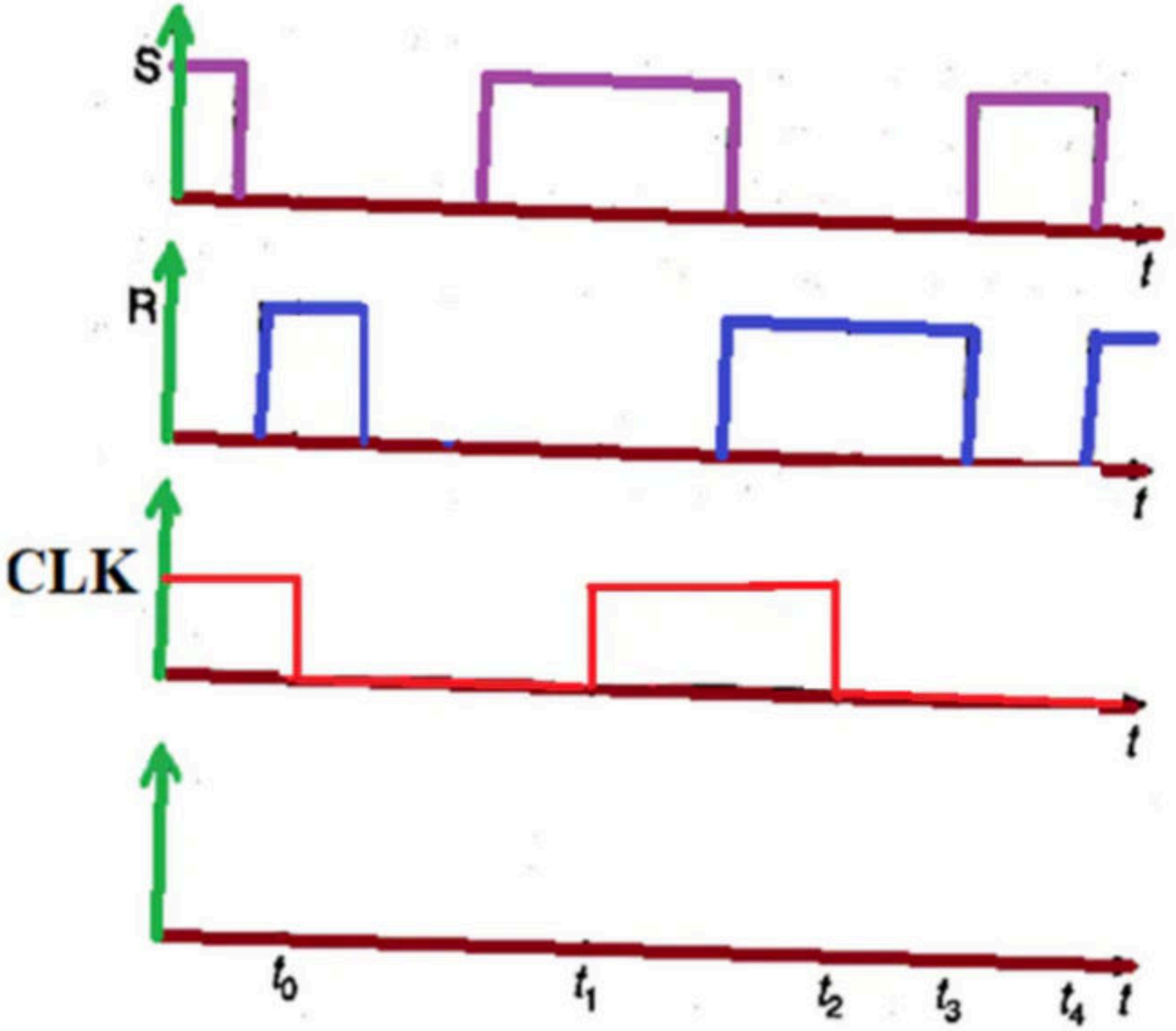
Draw the output waveform



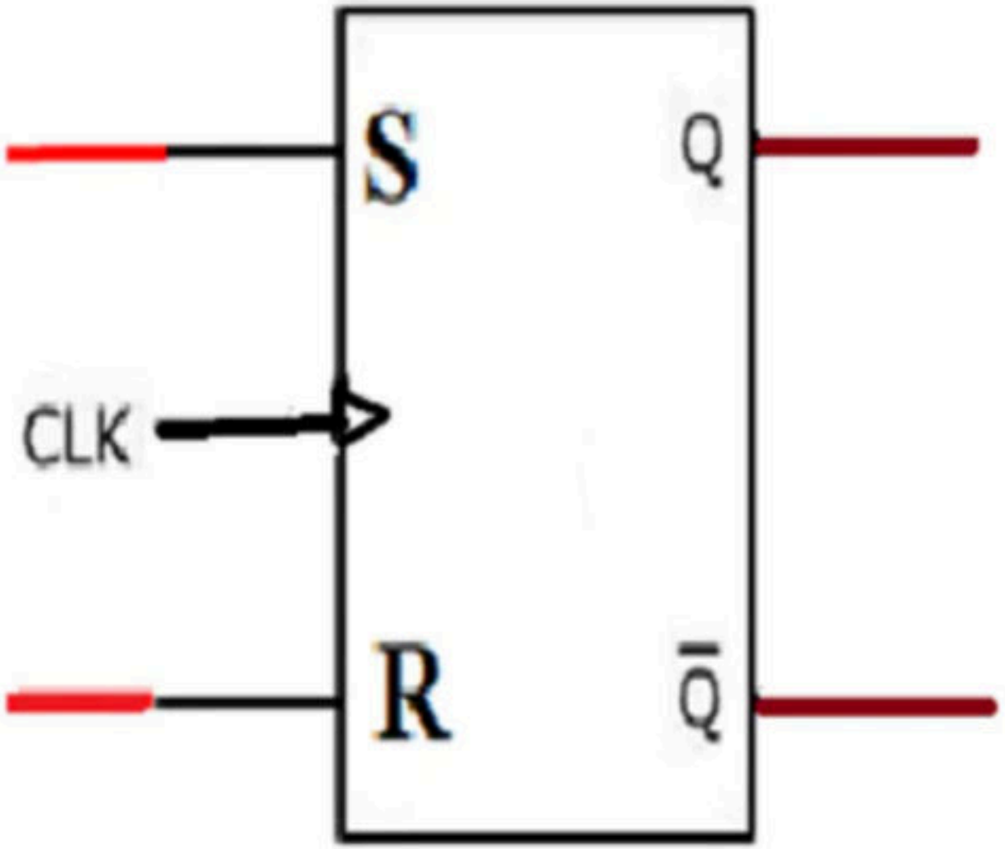


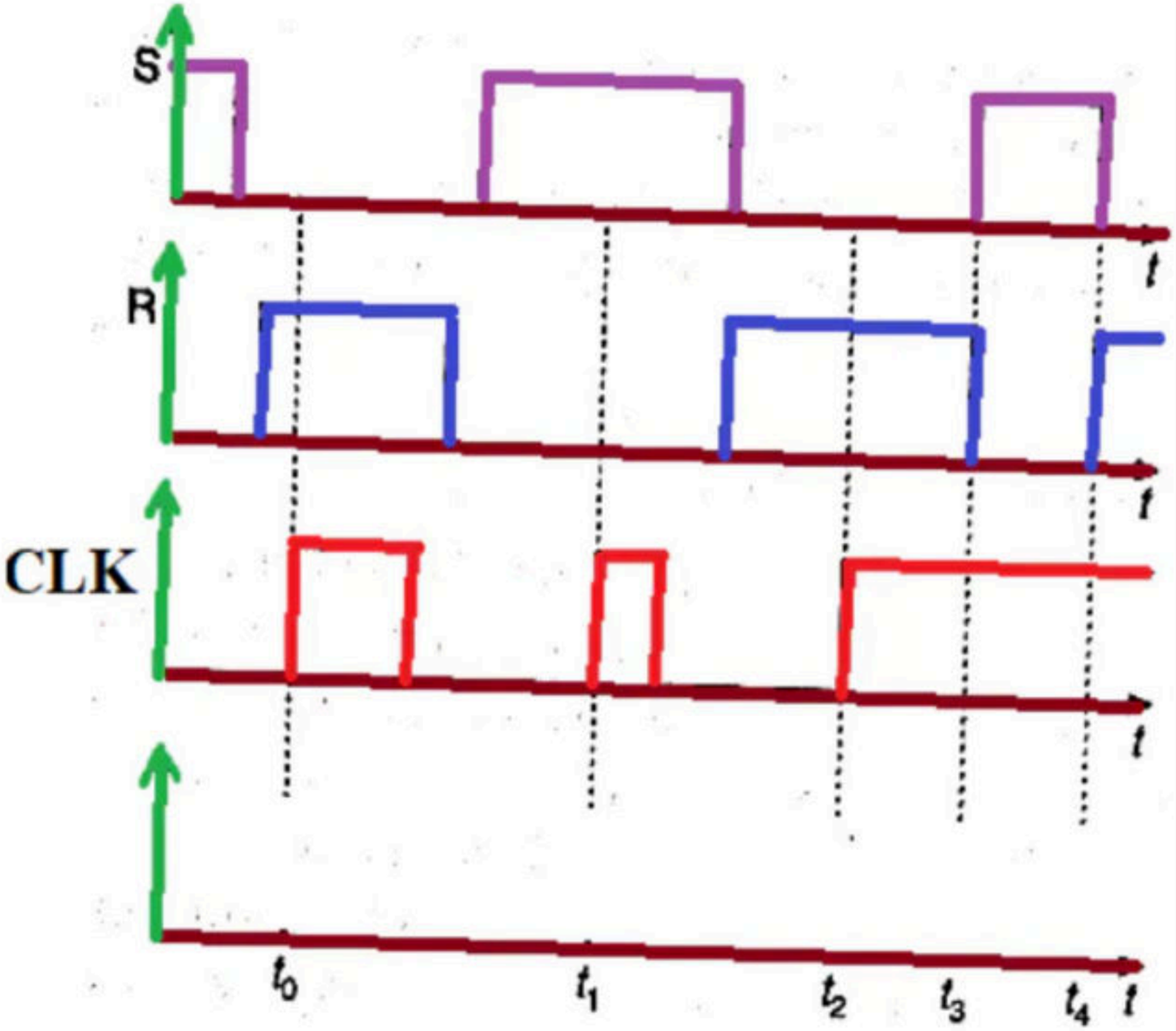
Draw the output wave form



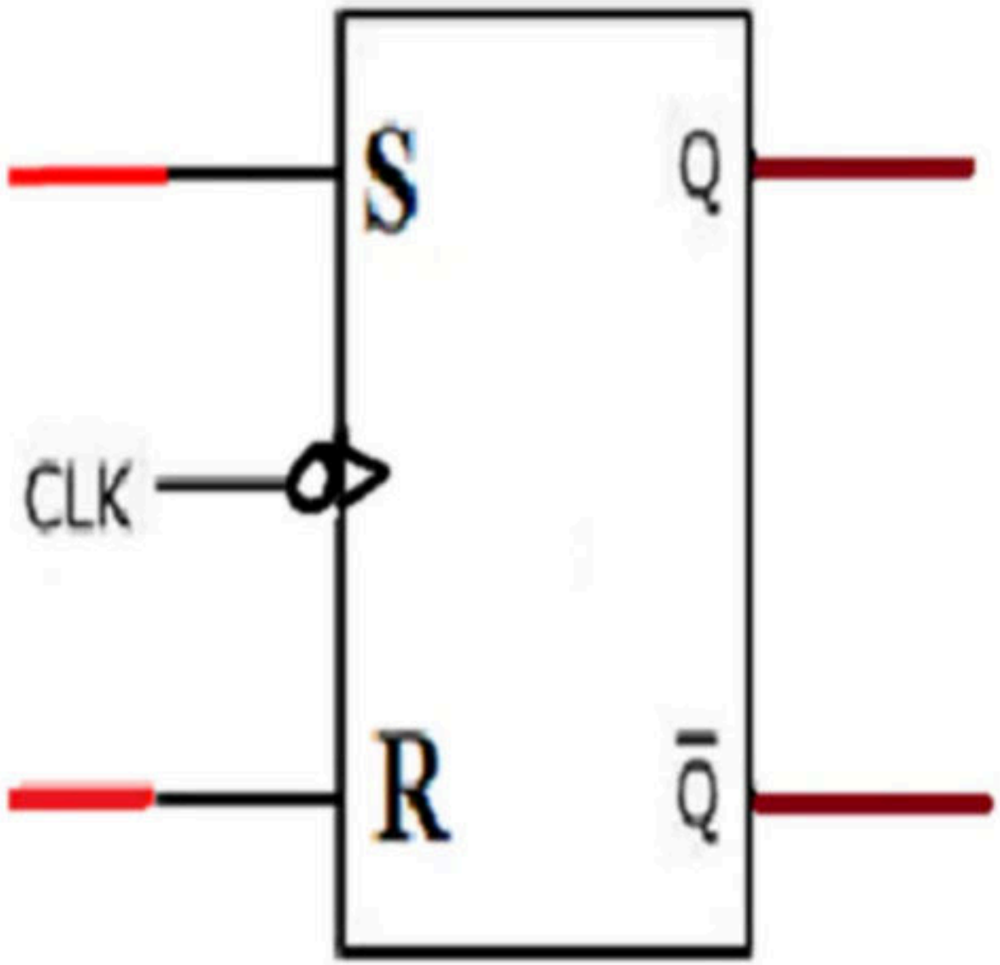


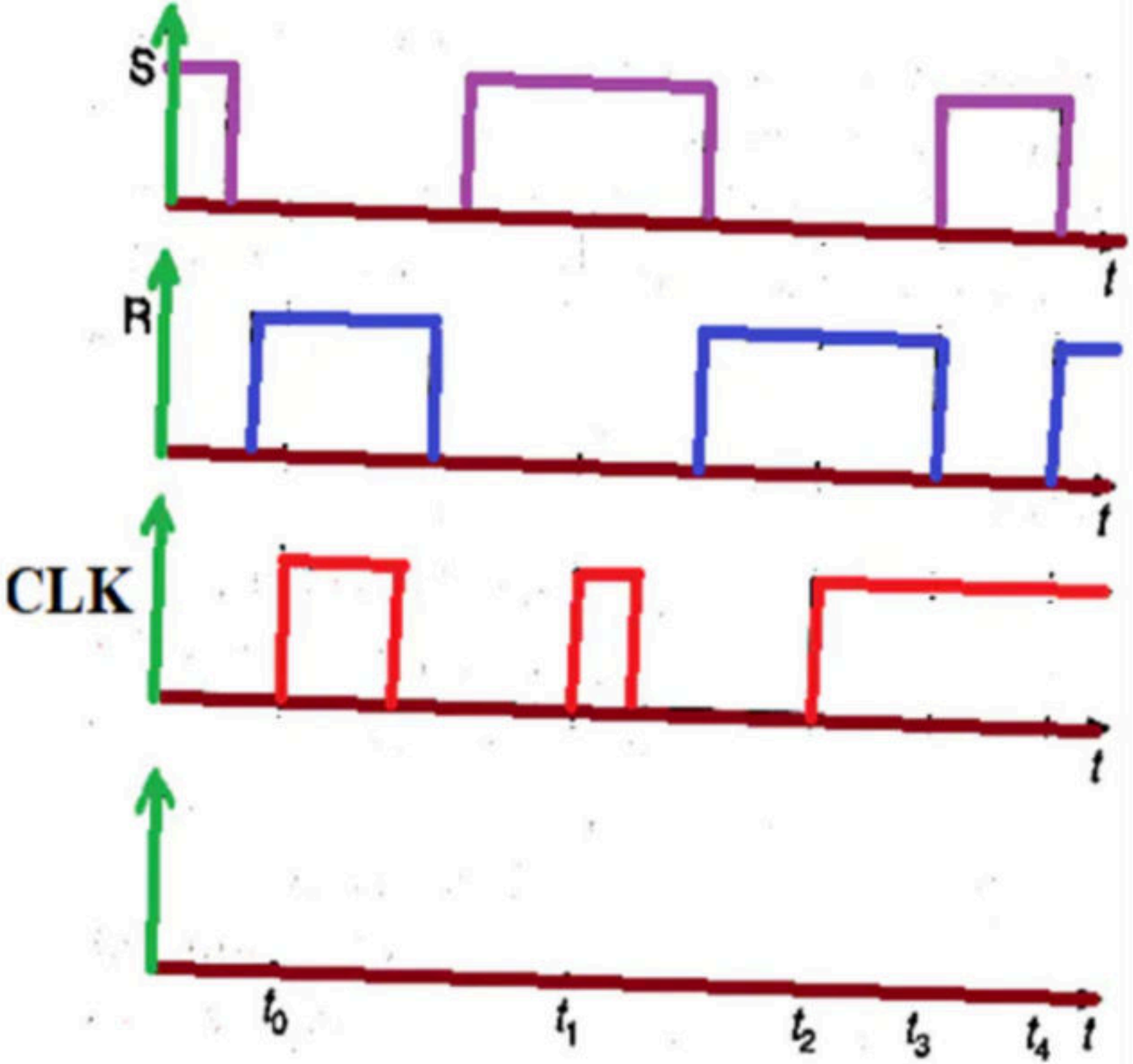
Draw the output wave form



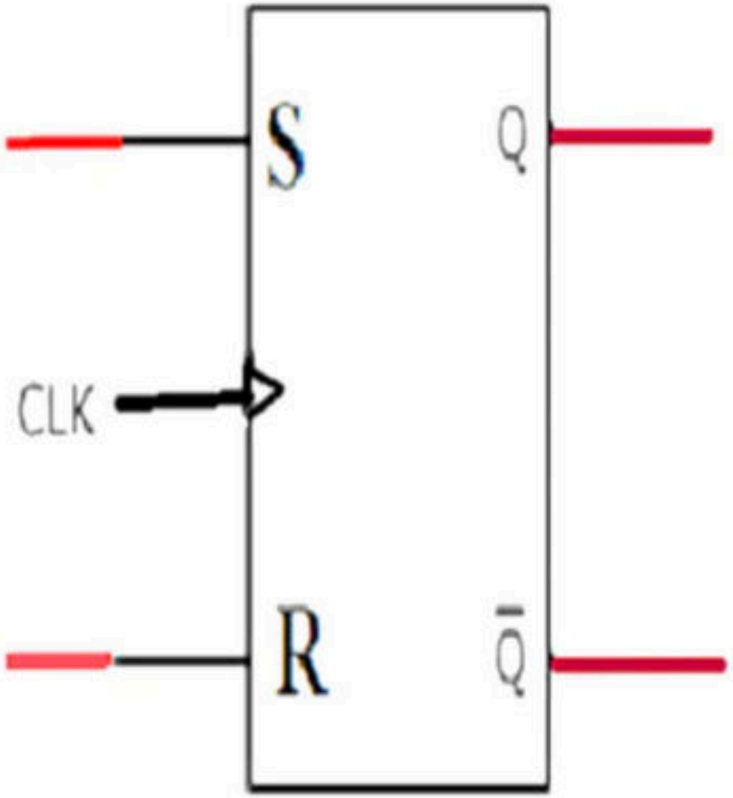


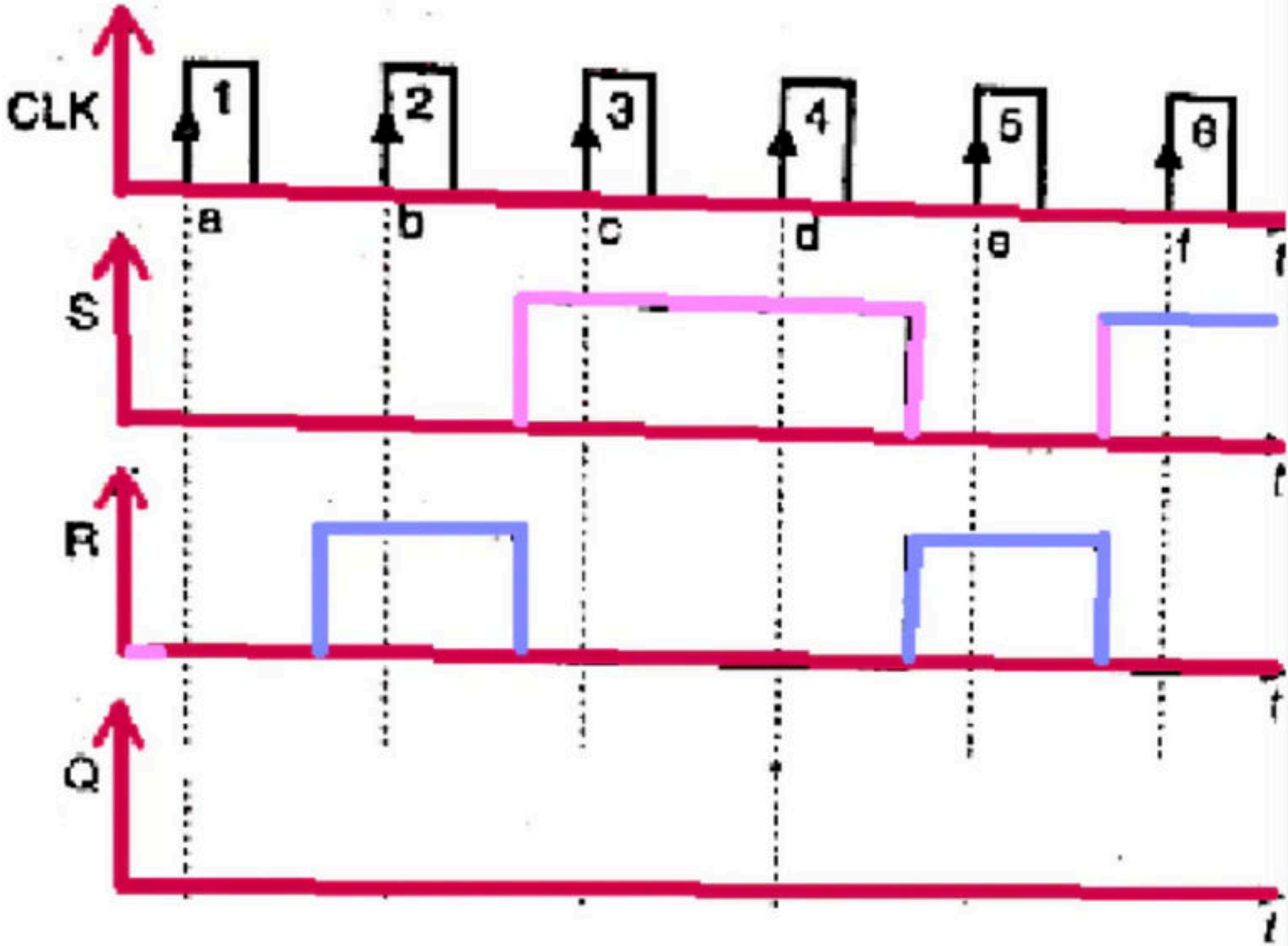
Draw the output wave form



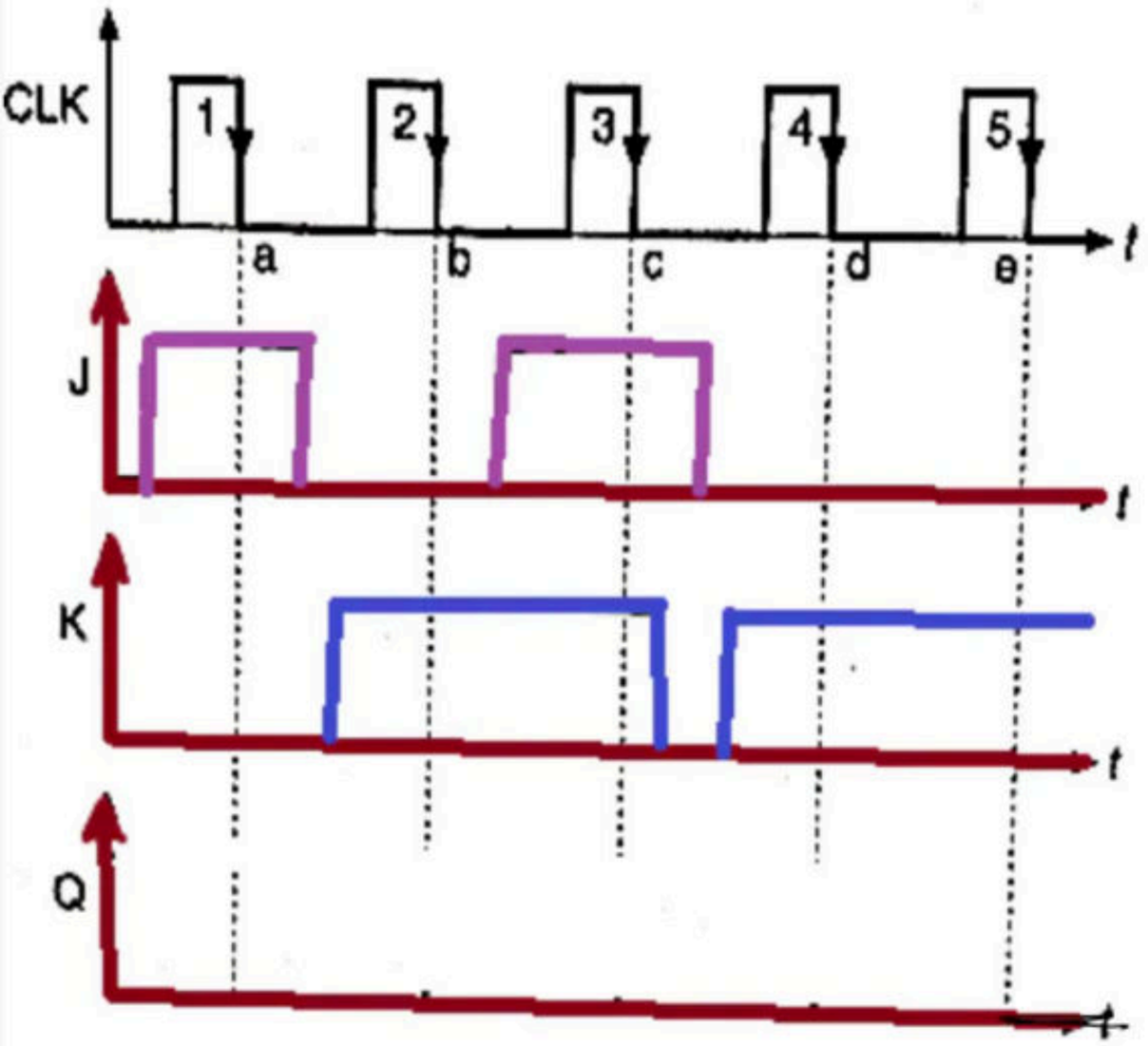


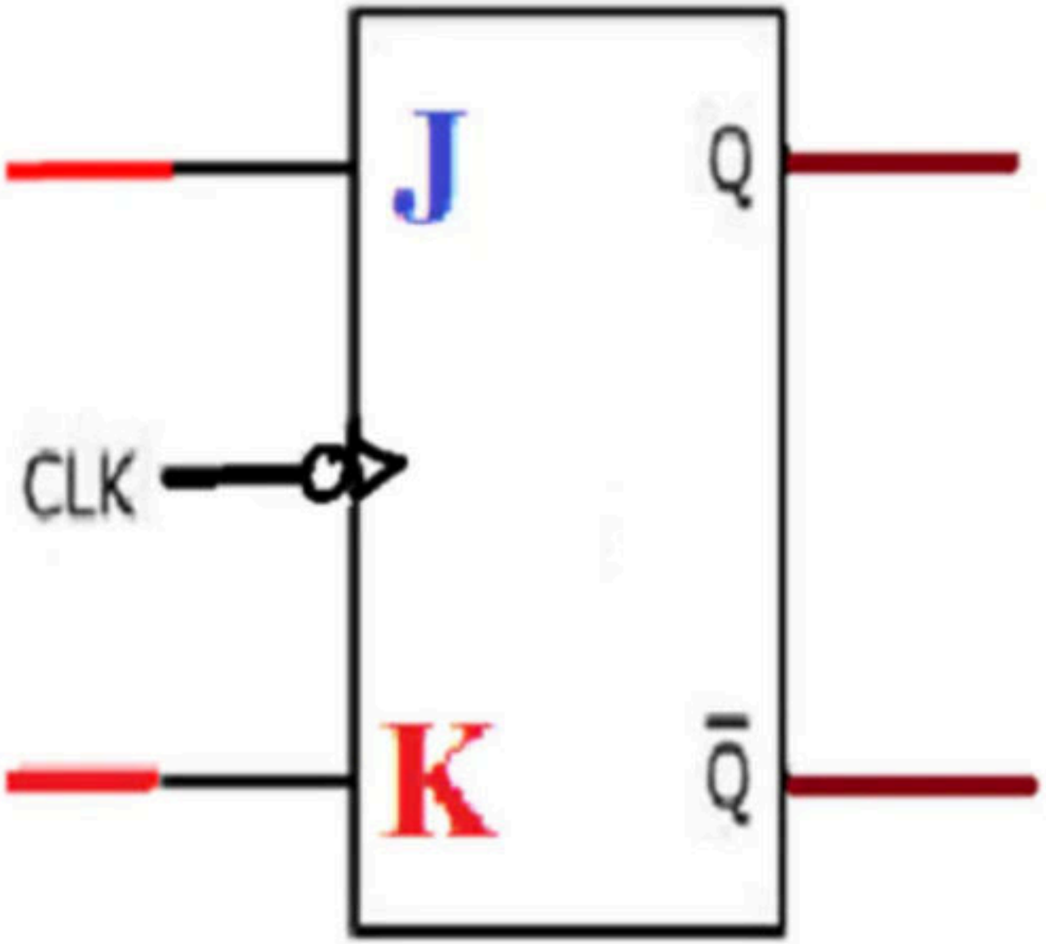
Draw the output wave form



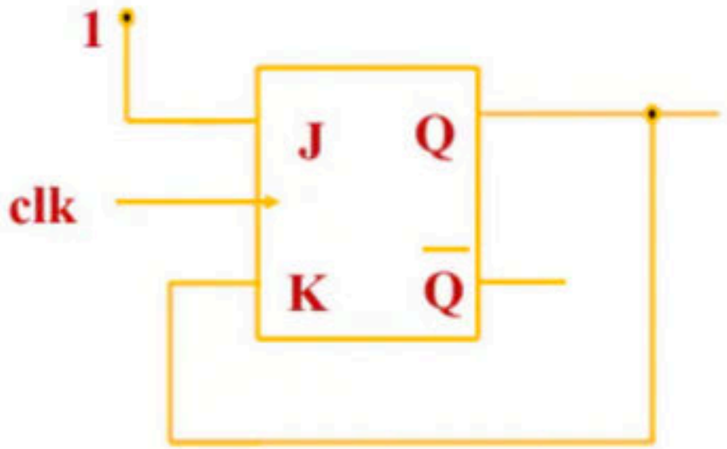


Draw the output waveform





Q. In the circuit shown in the figure, Q=0, initially. When clock pulses are applied, the subsequent states of 'Q' will be.



(a) 1, 0, 1, 0, 1,……

(b) 0, 0, 0, 0,…..

(c) 1, 1, 1, 1,…..

(d) 0, 1, 0, 1,…..

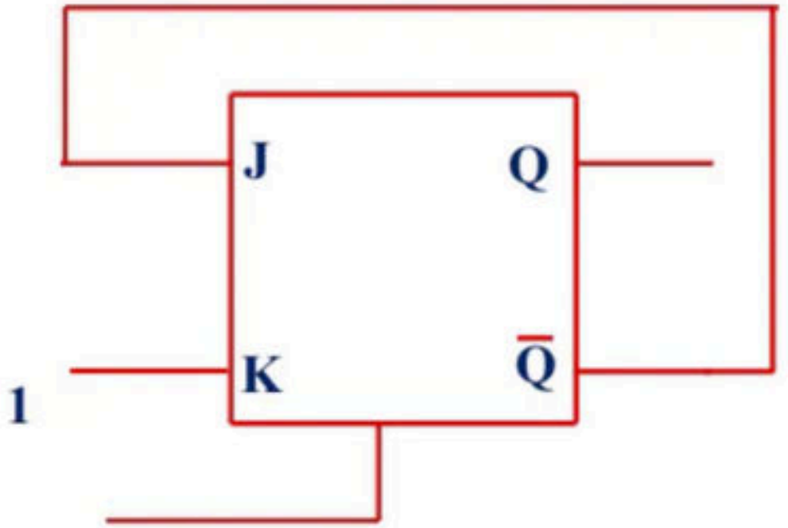
Q. Consider the following J-K flip-flop. In the above J-K flip-flop, $J = \overline{Q}$ and $K = 1$. Assume that the flip-flop was initially cleared and then clocked for 6 pulses. What is the sequence at the Q output?

(a) 01000

(b) 011001

(c) 010010

(d) 010101



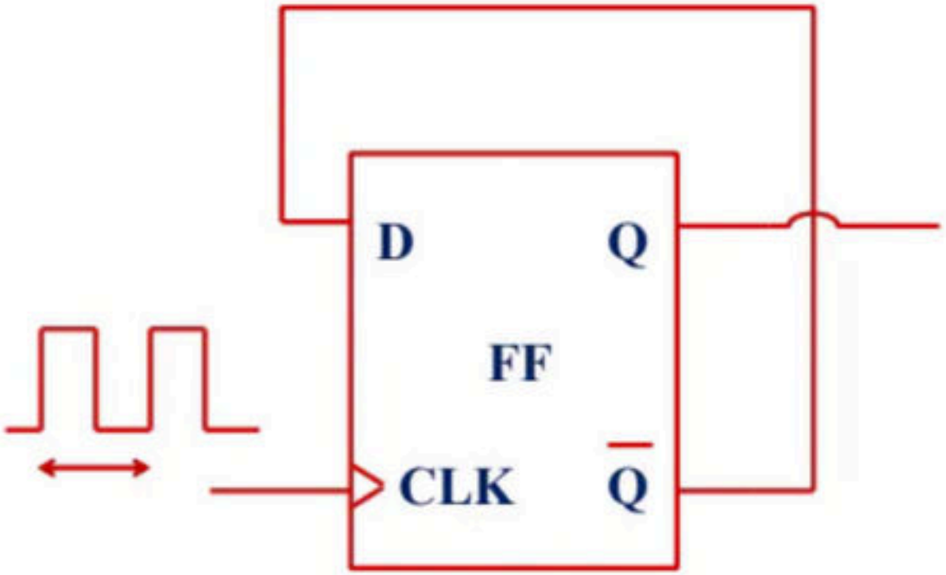
Q. The frequency of the clock signal applied to the rising edge triggered D flip-flop shown in figure is 10kHz. The frequency of the signal available at Q is.

(A) 10 kHz

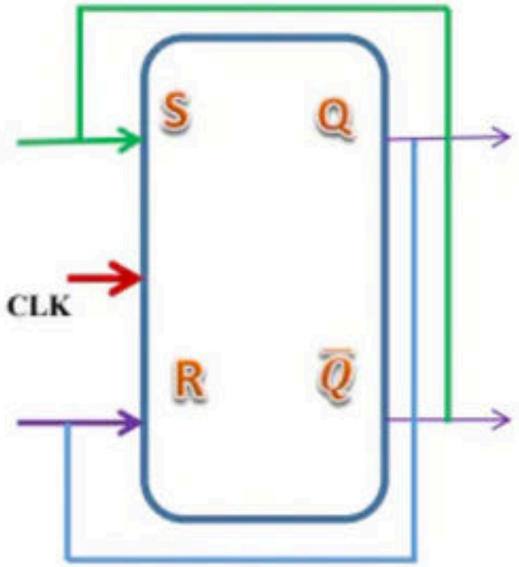
(B) 2.5 kHz

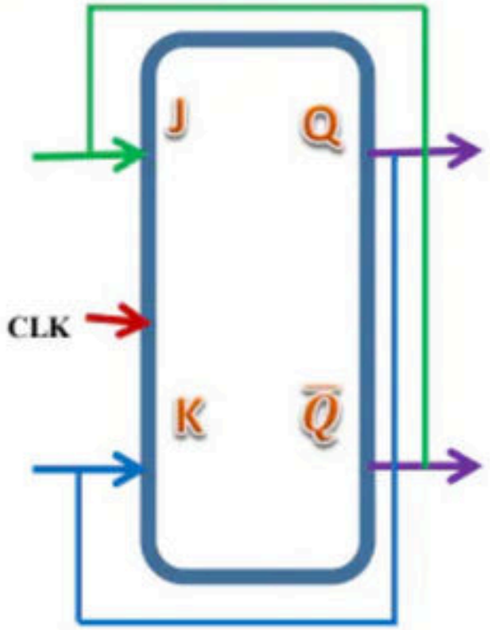
(C) 20 kHz

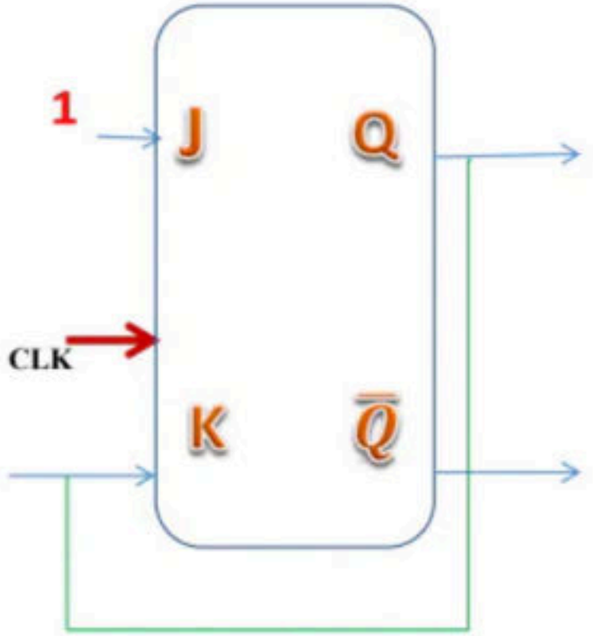
(D) 5 kHz

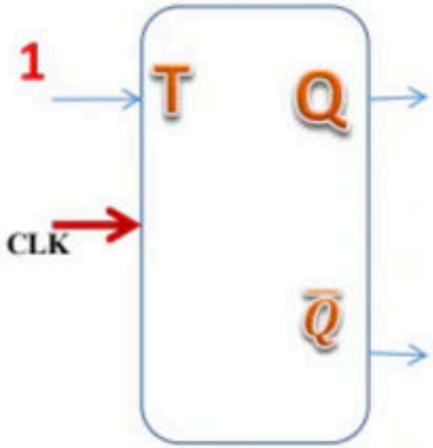


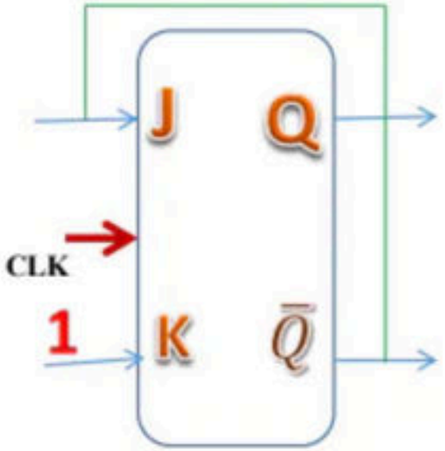
Toggle Modes

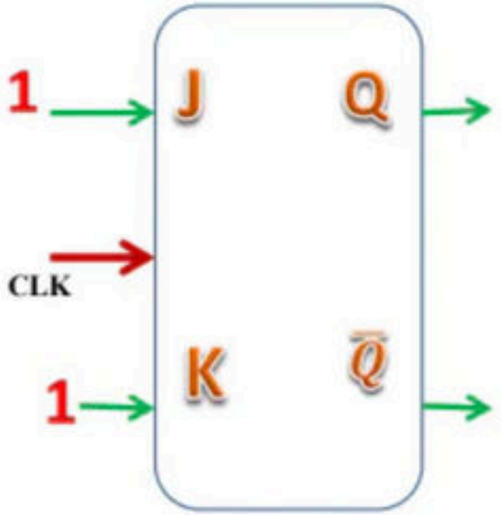


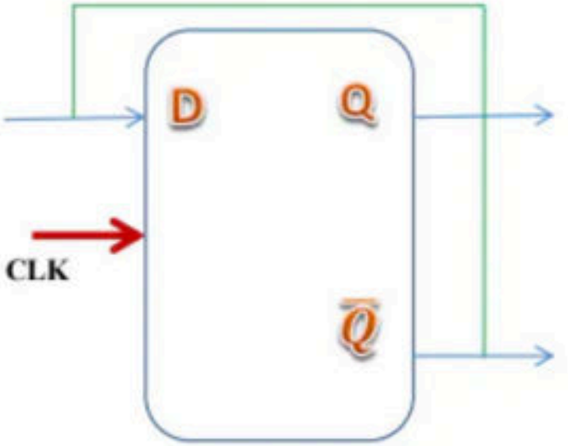












Q. The output Q_n of a J-K flip-flop is zero. It changes to 1 when a clock pulse is applied. The input J_n and K_n are respectively (× represents don't care condition):

(a) 1 and ×

(b) 0 and ×

(c) × and 0

(d) × and 1

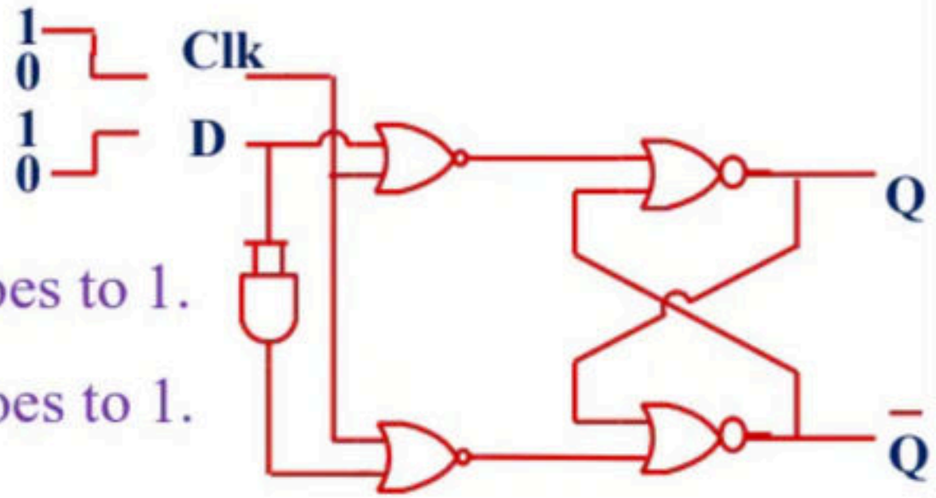
Q. For the circuit shown in the figure, D has a transition from 0 to 1 after CLK changes from 1 to 0. Assume gate delays to be negligible. Which of the following statements is true?

(A)A goes to 1 at the CLK transition and stays at 1.

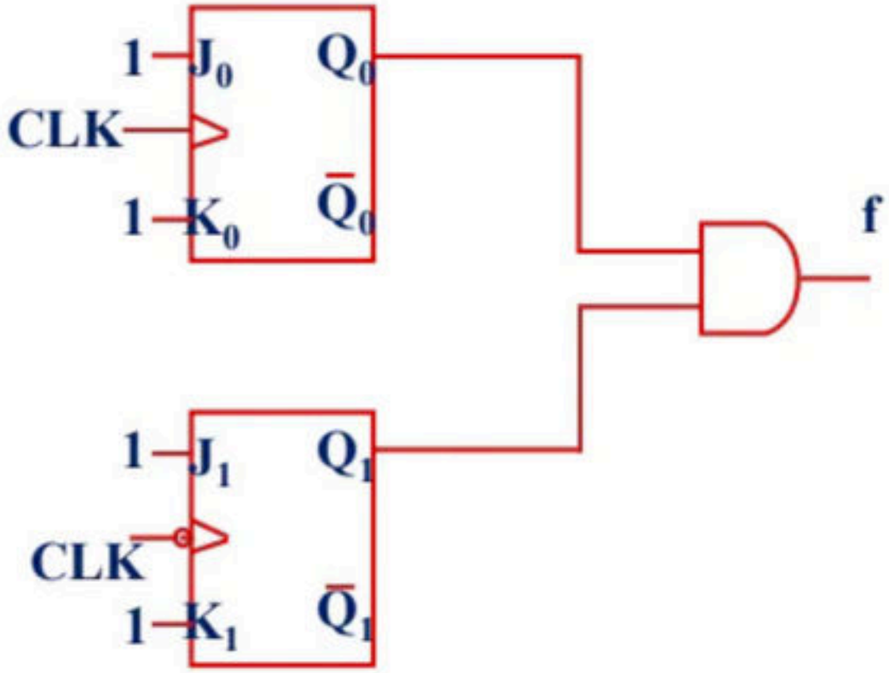
(B)Q goes to 0 at the CLK transition and stays at 0.

(C)Q goes to 1 at the CLK transition and goes to 0 when D goes to 1.

(D)Q goes to 0 at the CLK transition and goes to 1 when D goes to 1.

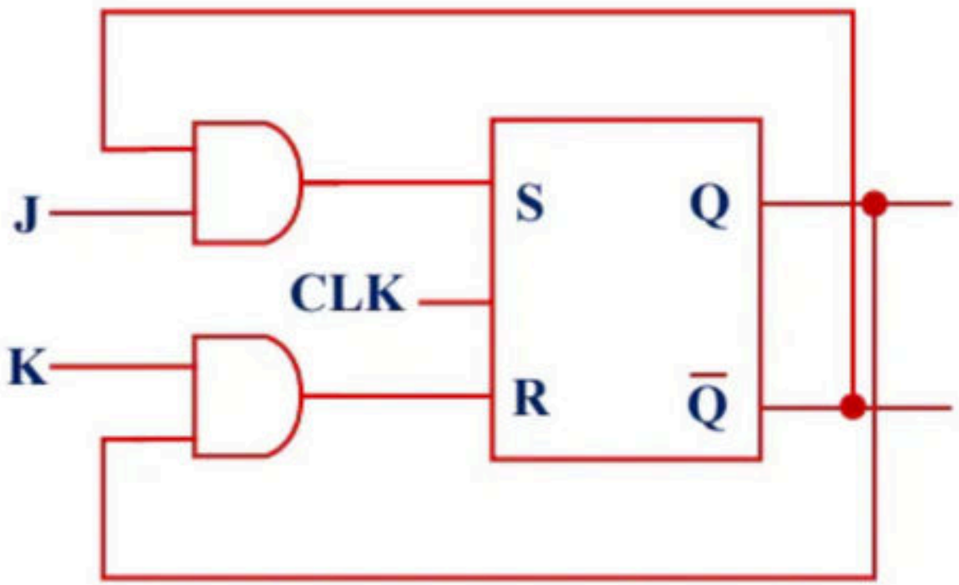


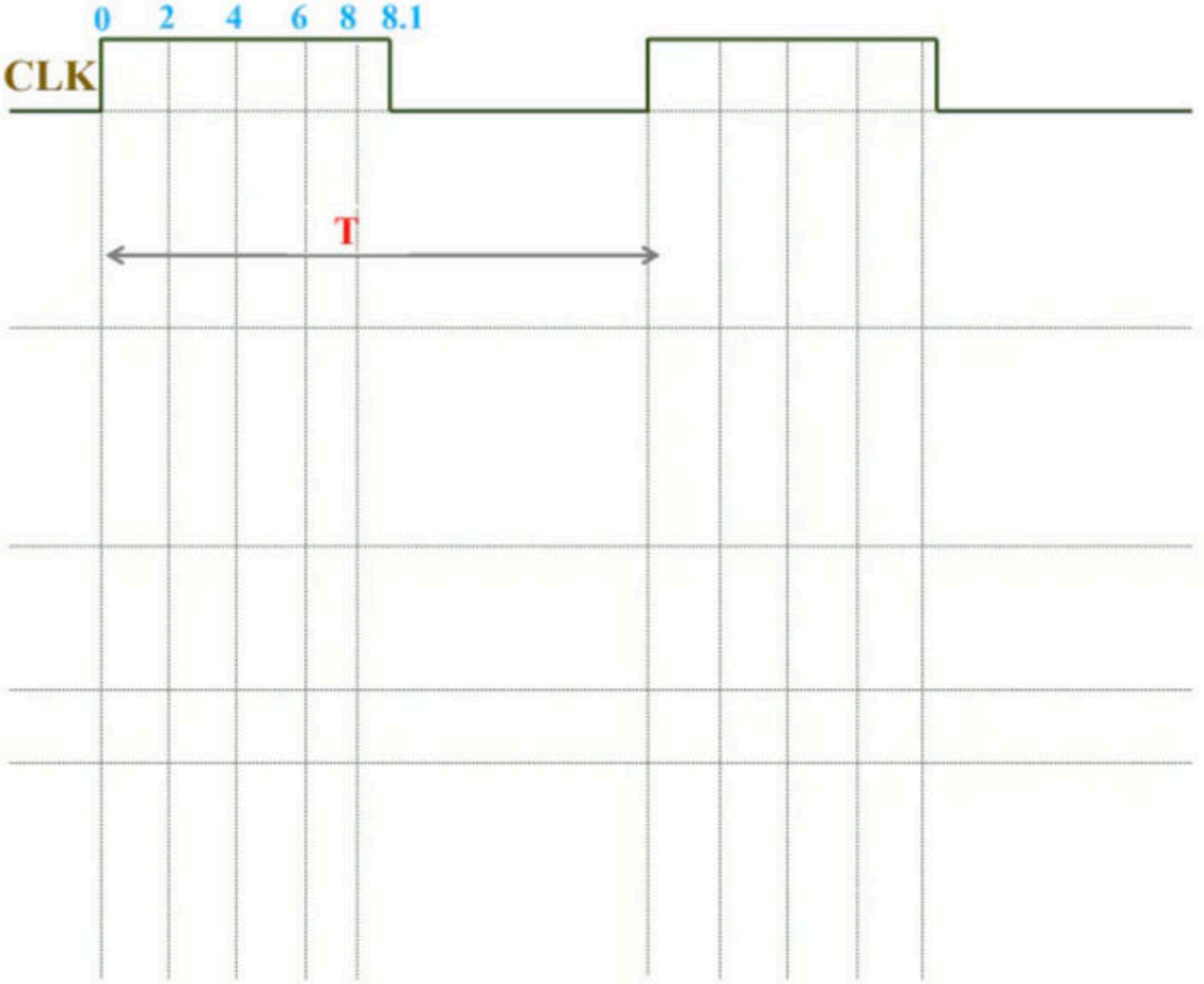
Q) Find the frequency and duty cycle of output , if the clock frequency is 10MHz



Race Around Condition

Race around condition



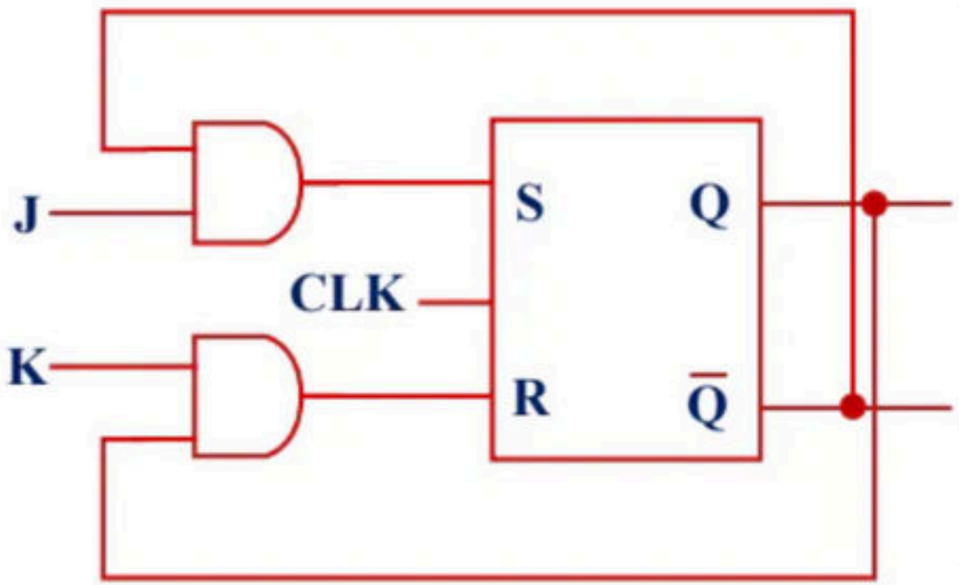


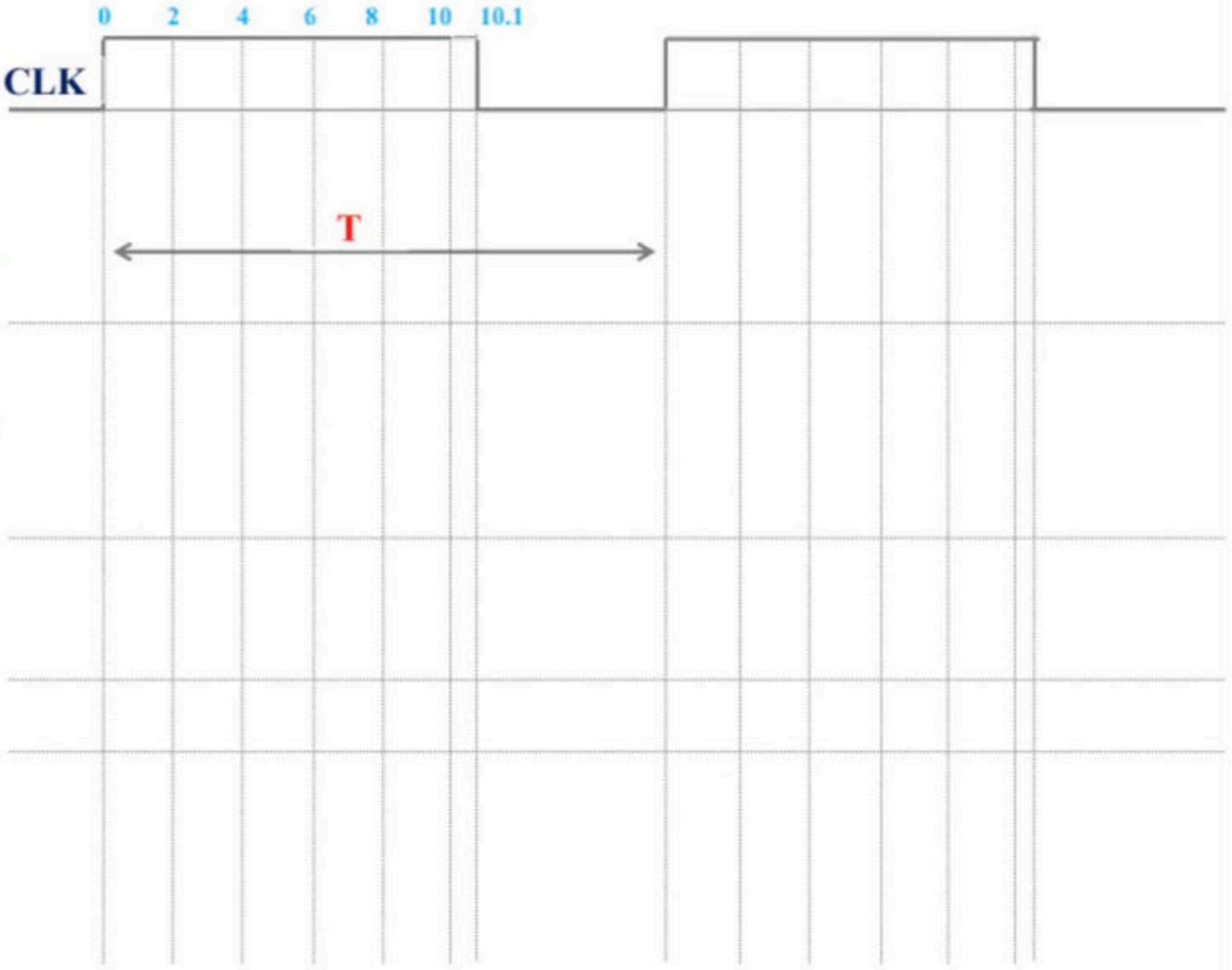
Case – 1

$T_{CLK} = 16.2ns$

$t_{pd} = 2ns$

Race around condition





Case – 2

$T_{CLK} = 20.2ns$

$t_{pd} = 2ns$

The output of the FF changes to 0 →1 →0 …. Continuously at the starting of the next clock the output is uncertain, which is called as Race Around Condition (RAC).

RAC occurs in any FF if the following three conditions satisfies

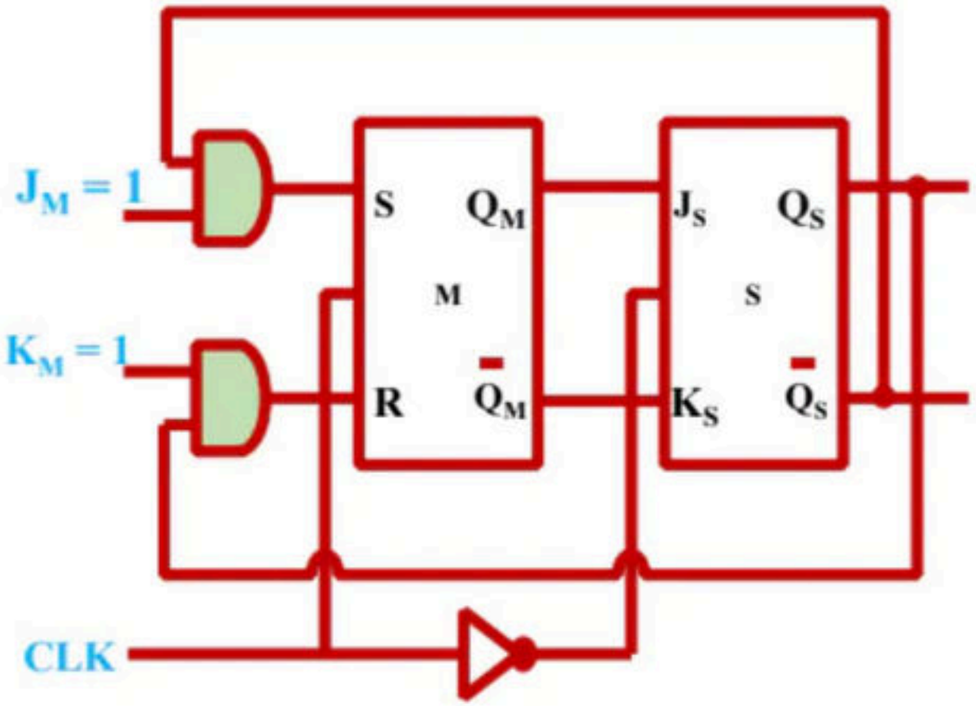
1. If the FFs are operated in level triggering
2. if (tpd) < (Tclk)on ,
3. If the FFs are operated in Toggle mode

If the above three conditions satisfies simultaneously then there is a continuous race in the output of the FF between 0 and 1 to reach the next state , who will be the winner of the race is not certain , that depends on tpd and (Tclk) on .

Remedy

1. (Tclk)on < (tpd) < T
2. By using Edge triggered
3. By Master – Slave Configuration

Master – Slave Configuration



1. In case of Master Slave configuration , Master is applied with input clock and Slave is applied with inverted clock , so out of two FFs at a time only one of the FF respond and other will not respond . As a result, Many times toggling in a single clock cycle has been converted to one time toggle , hence *RAC is avoided* .

2. In Master Slave configuration , command signal is generated by master FF and the response of the command signal is given by slave FF

3. Master slave FF can store 1 – bit of data

Conversation of FFs

Steps

1. Write the Characteristic table (state table) of the required FF
2. Match the excitation table of the given FF to required FF
3. Write excitation expression
4. Minimize logical expression
5. Implement logic circuit

Q) Convert the SR FF to JK FF

Q) Convert the SR FF to the XY FF whose truth table is given below

X	Y	Q+
0	0	1
0	1	$\bar{Q}$
1	0	Q
1	1	0

JK to SR

$J = S$

$K = R$

SR to JK

$S = J\overline{Q}$

$R = KQ$

D to SR

$D = S+\overline{R}Q$

T to SR

$T = S\overline{Q}+RQ$

JK to D

$J = D$

$K = \overline{D}$

SR to D

$S = D$

$R = \overline{D}$

D to JK

$D = J\overline{Q}+\overline{K}Q$

T to JK

$T = J\overline{Q}+KQ$

JK to T

$J = T$

$K = T$

SR to T

$S = T\overline{Q}$

$R = TQ$

D to T

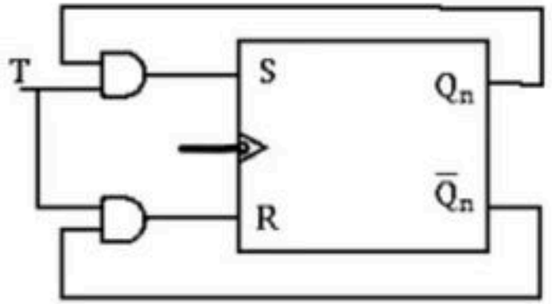
$D = T \oplus Q$

T to D

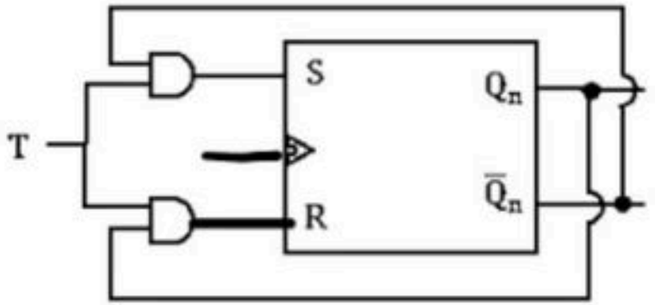
$T = D \oplus Q$

3. A T flip flop can be implemented by S-R flip flops. Identify the correct implementations

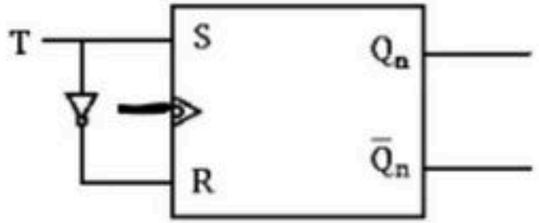
(A)



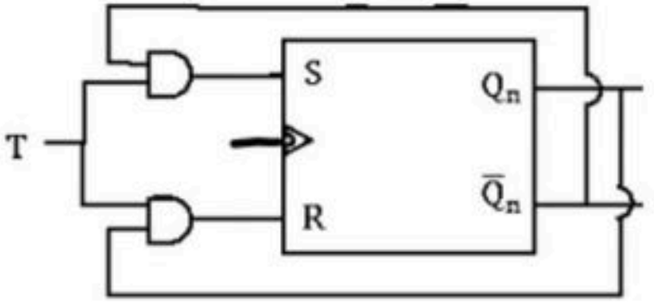
(B)



(C)



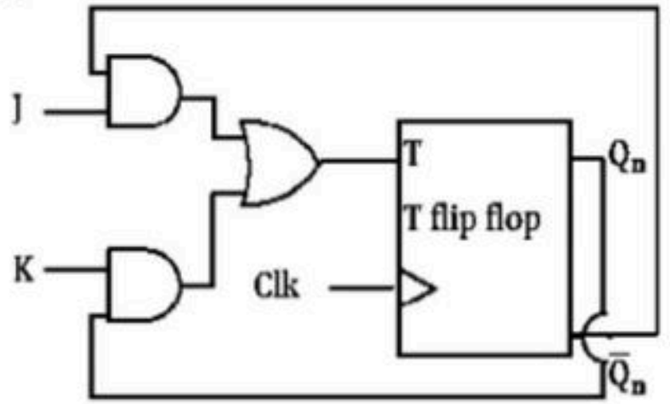
(D)



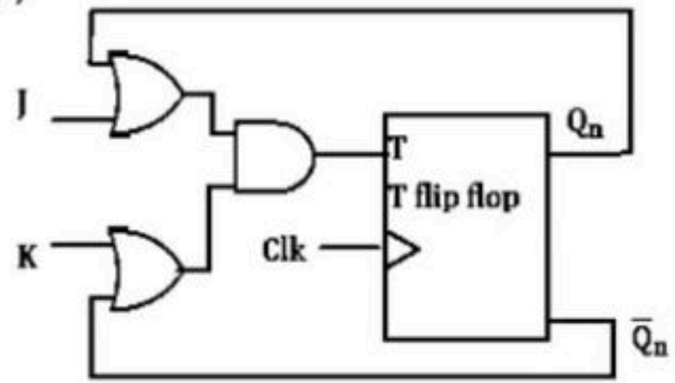
72. A JK flip flop can be implemented by T flip-flops. Identify the correct implementation

GATE (EE-2014)

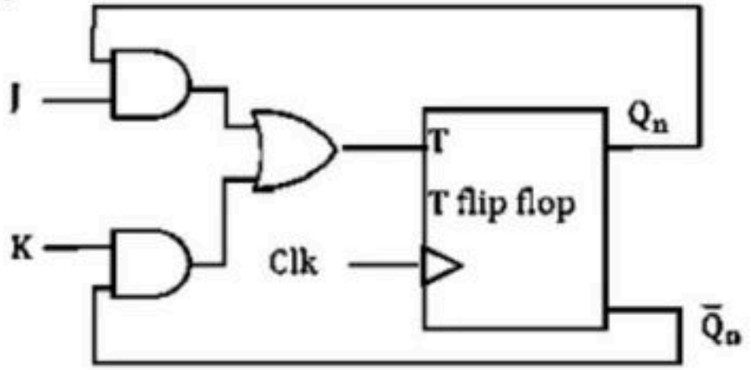
(A)



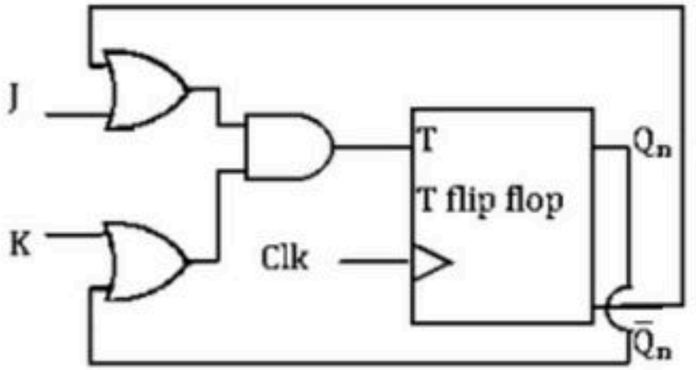
(B)



(C)



(D)



Shift Registers

Shift Registers

- A FF is a single bit memory element , which can not store multiple bits at a time , so we combine number of FFs the resultant circuit can serve this purpose , is known as shift registers .
- Since no data manipulation is required , so we prefer D- FFs for this purpose

As we know for D –FF

$$Q+ = D$$

To store n –bits , n – FFs are required

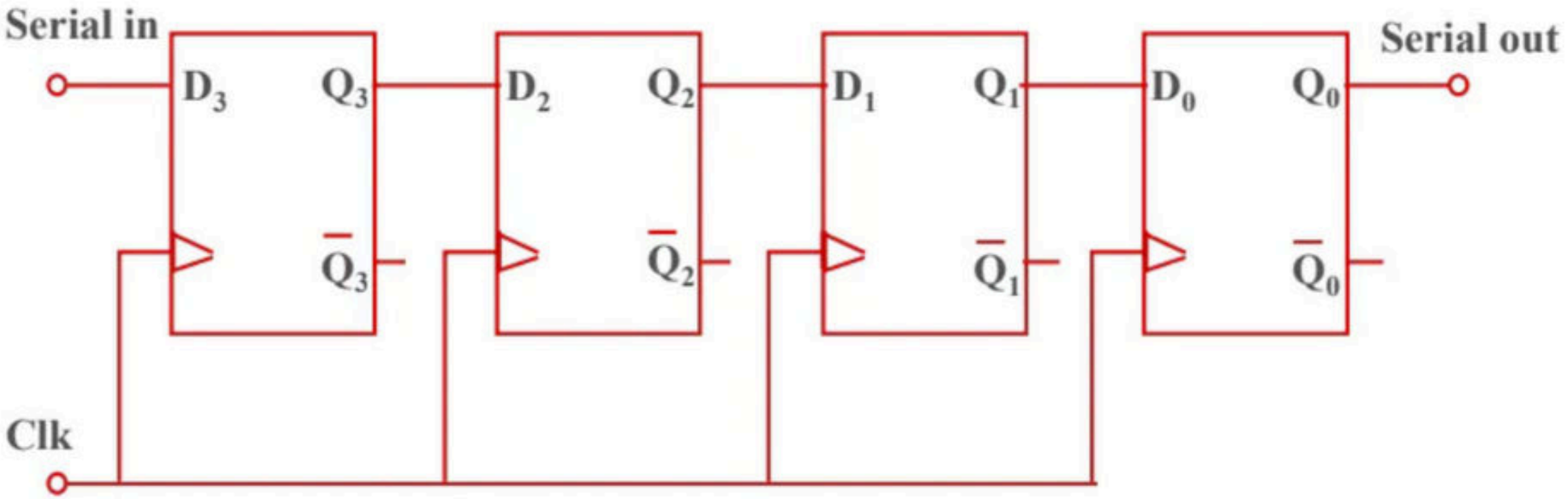
The data is available in two forms

1.Serial data (Temporal code)

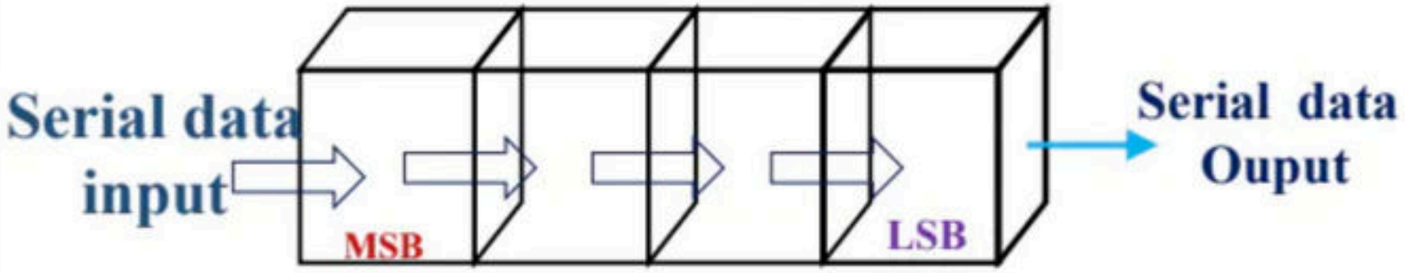
2. Parallel data (spatial code)

- Depending on i/p and o/p , registers are classified into 4 types
  1. Serial In Serial Out (SISO)
  2. Serial In Parallel Out (SIPO)
  3. Parallel In Parallel Out (PIPO)
  4. Parallel In Serial Out (PISO)

Serial In Serial Out



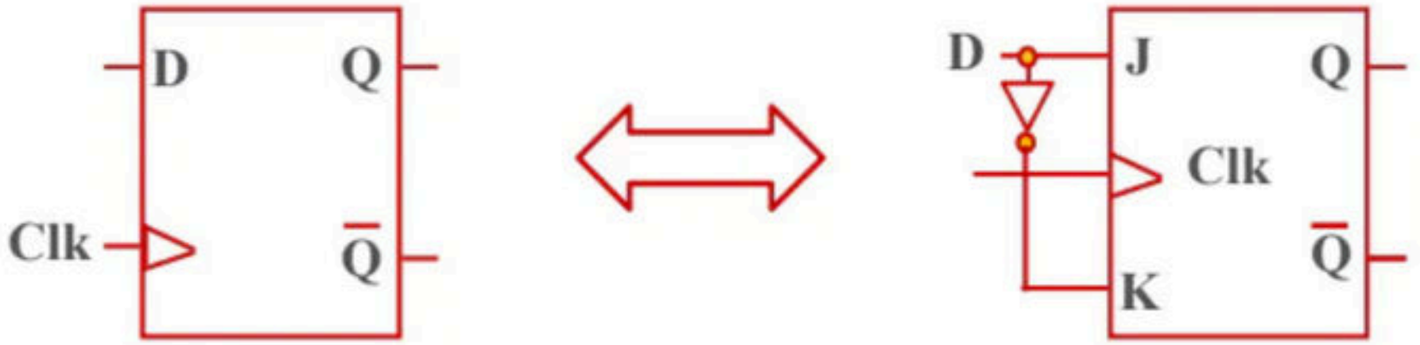
Block Diagram representation



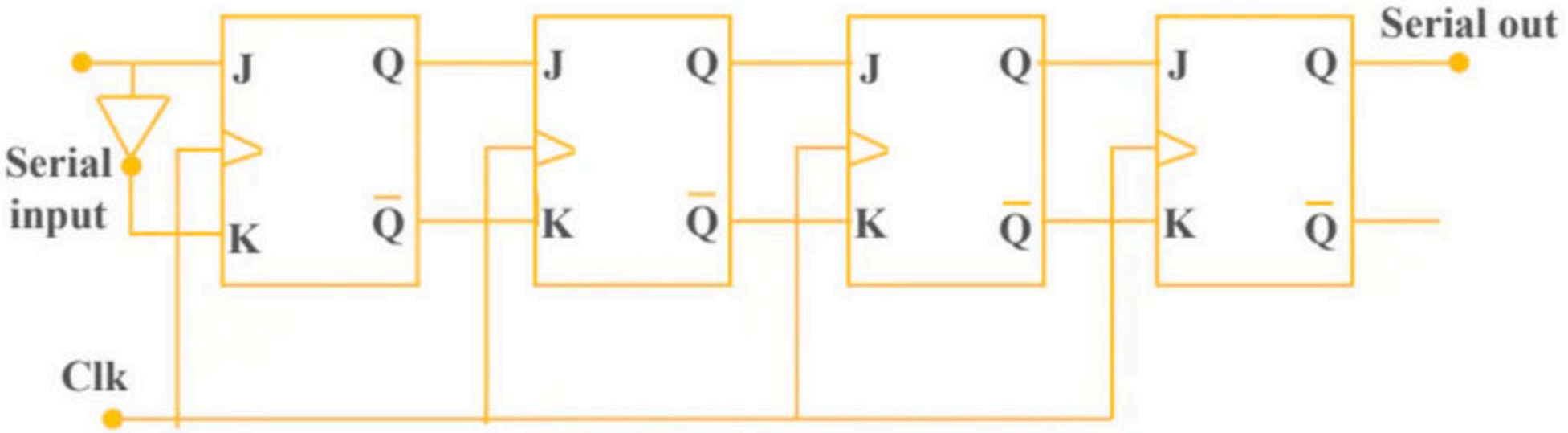
- SISO Configuration has only
  - 1- input
  - 1- output
- For SISO configuration
  - for storing = Clock pulses
  - for retrieving = clock pulses
  - Total number clock pulses =

CLK	INPUT	Q3	Q2	Q1	Q0

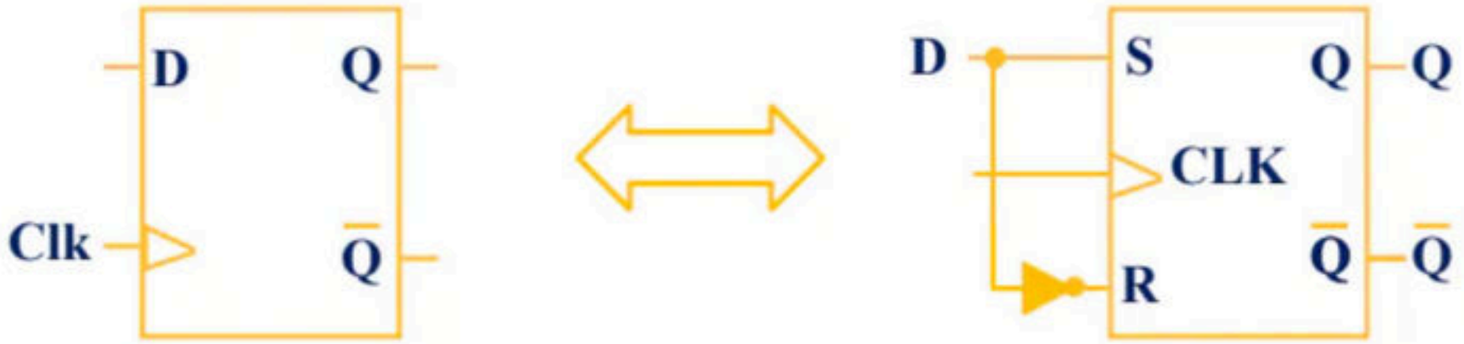
JKFF ⟺ D FF



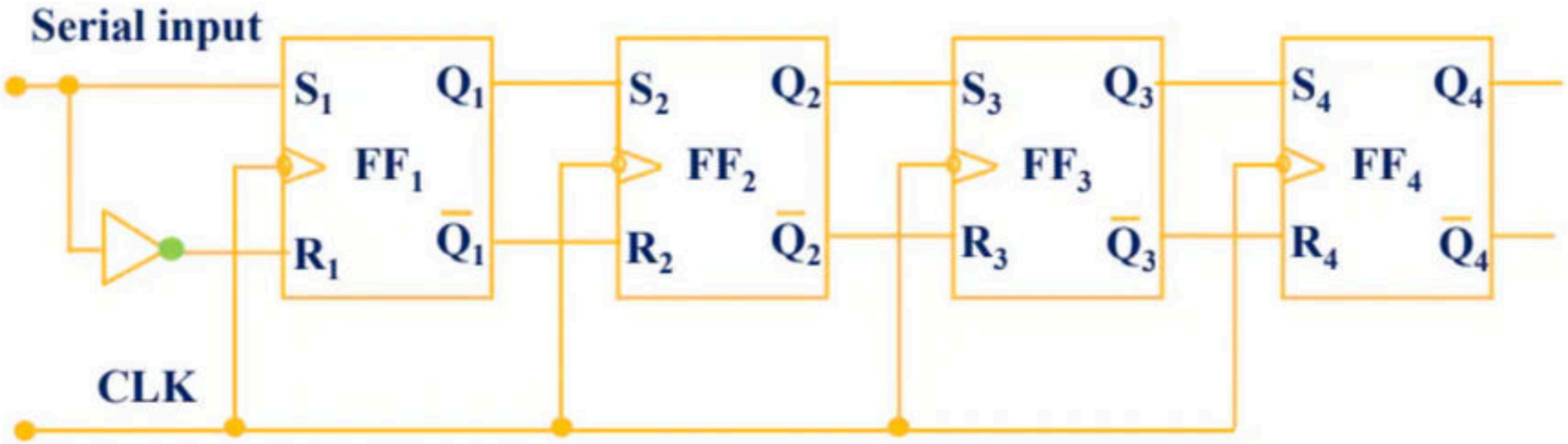
SISO Shift Register using JK FF



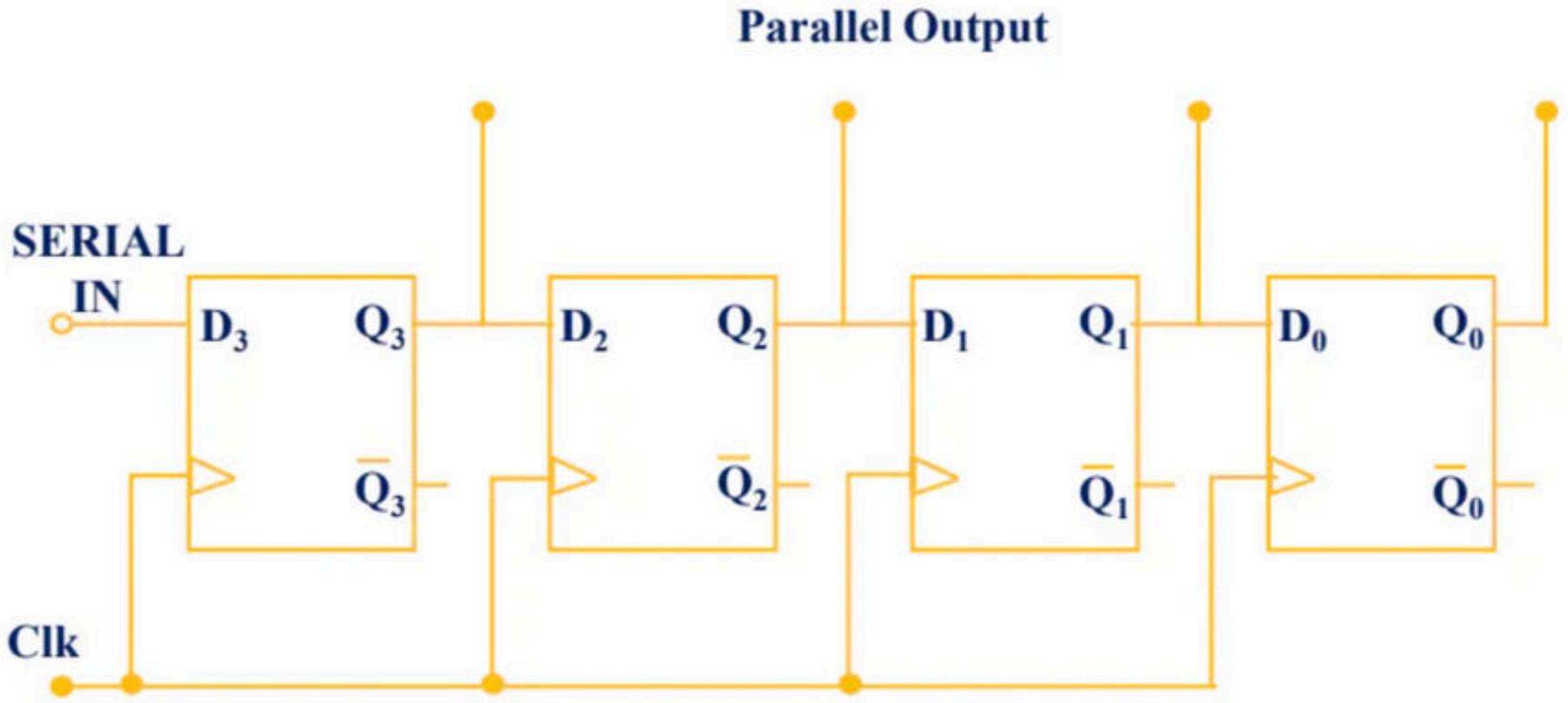
SR FF ⟺ D FF



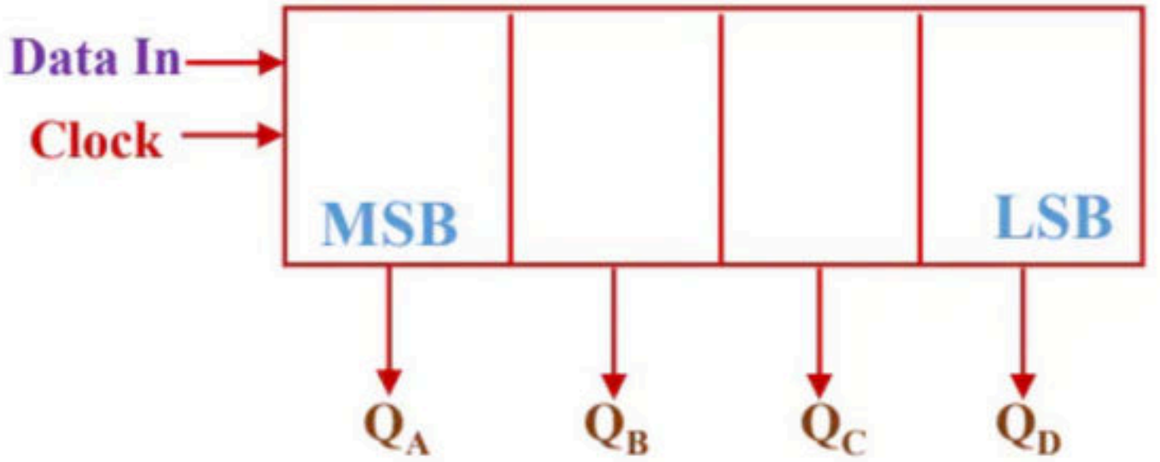
SISO using SR FF



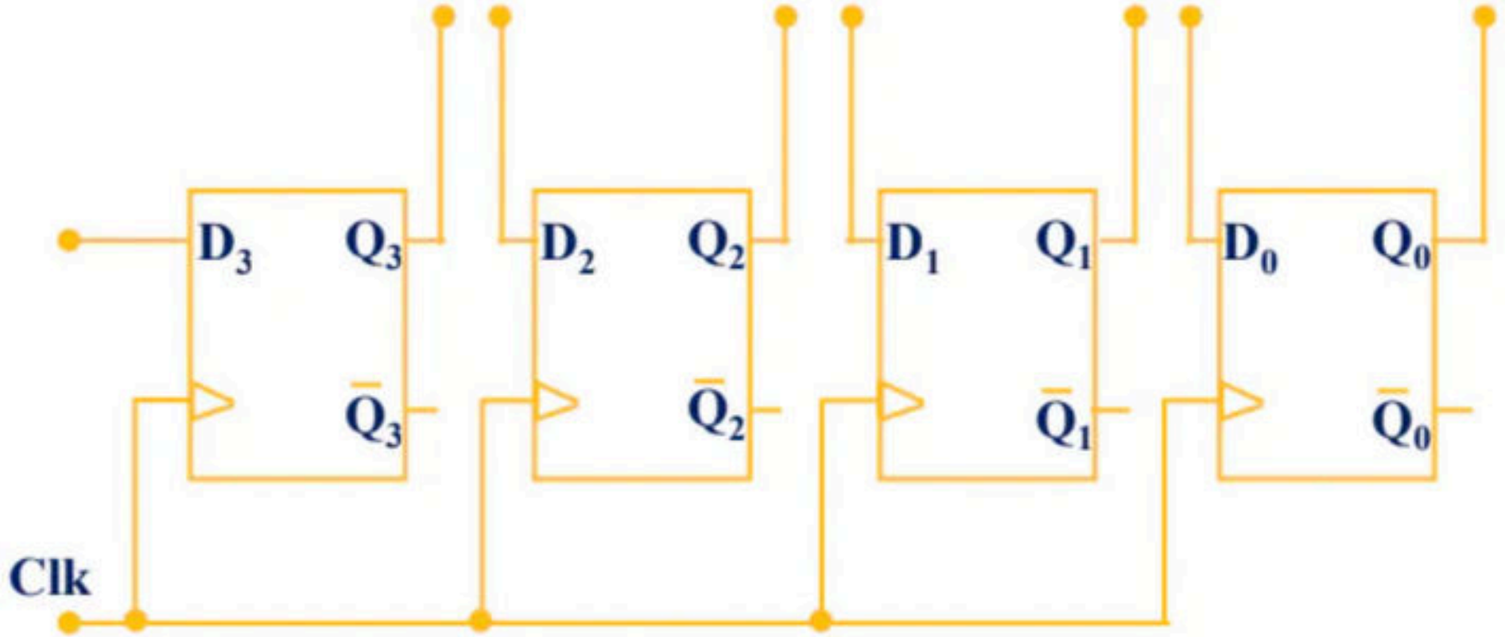
Serial In Parallel Out



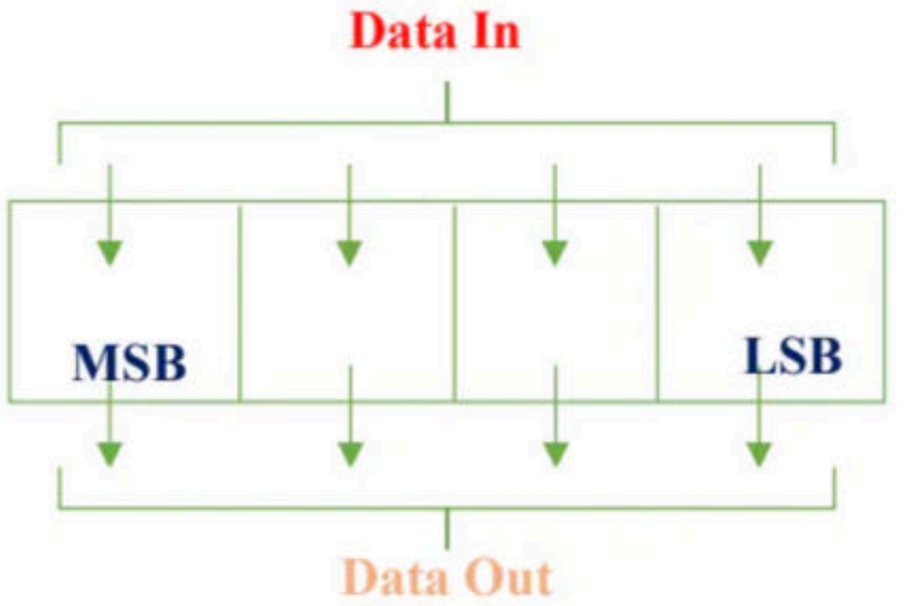
- SIPO Configuration has only
  - 1- input
  - 4- output
- For SIPO configuration
  - for storing = Clock pulses
  - for retrieving = Clock pulses
  - Total number clock pulses =



Parallel In Parallel Out



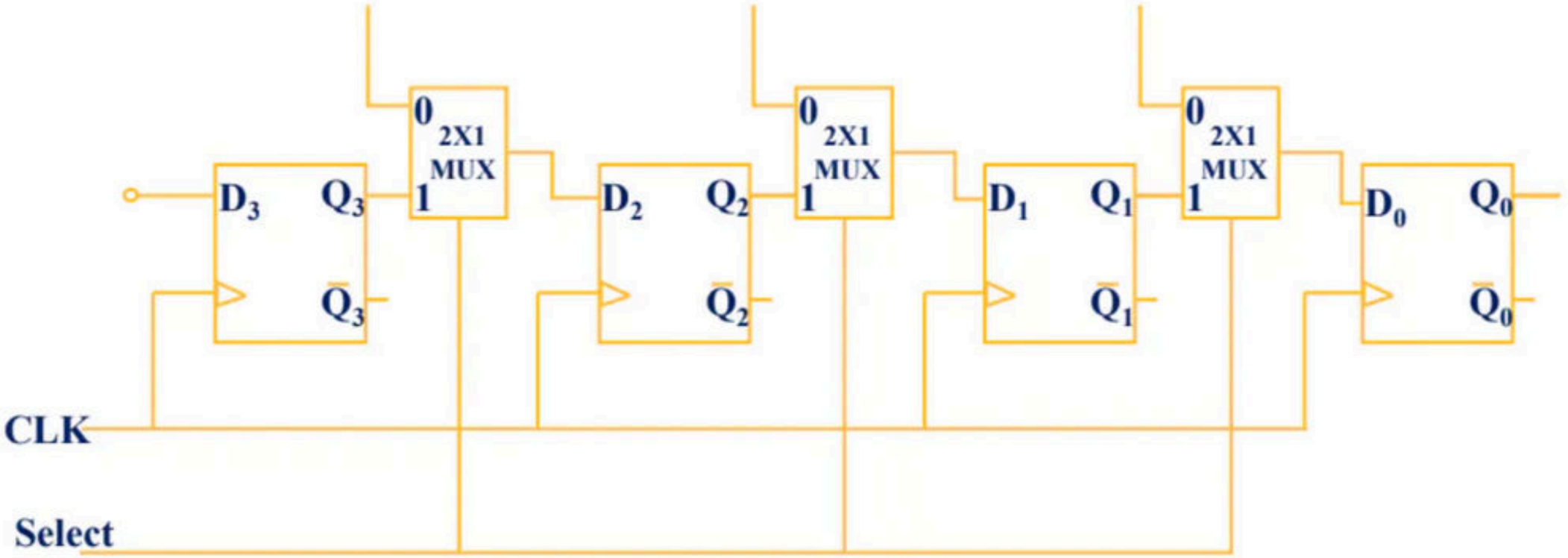
- PIPO Configuration has only
  - 4- input
  - 4- output
- For PIPO configuration
  - for storing = Clock pulses
  - for retrieving = Clock pulses
  - Total number clock pulses =



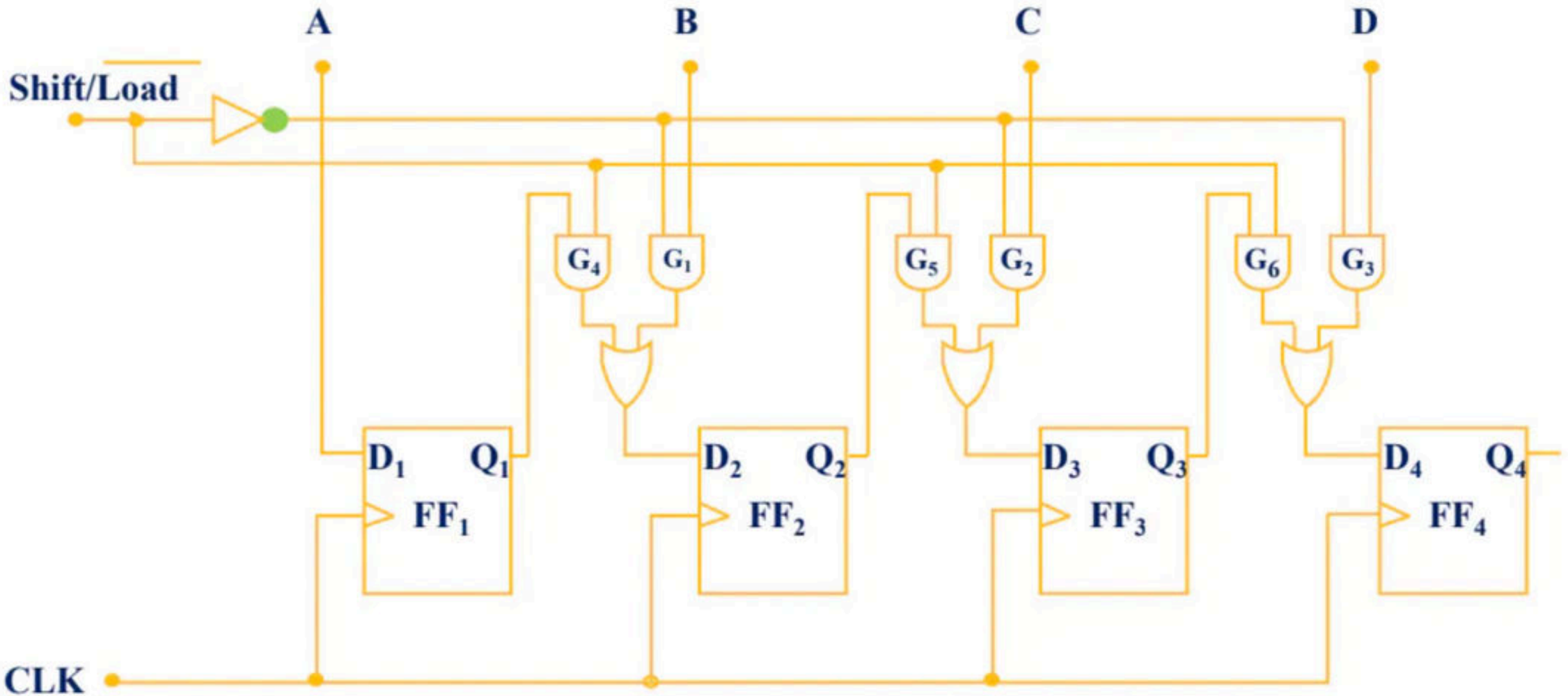
Parallel-IN Parallel-OUT Shift register

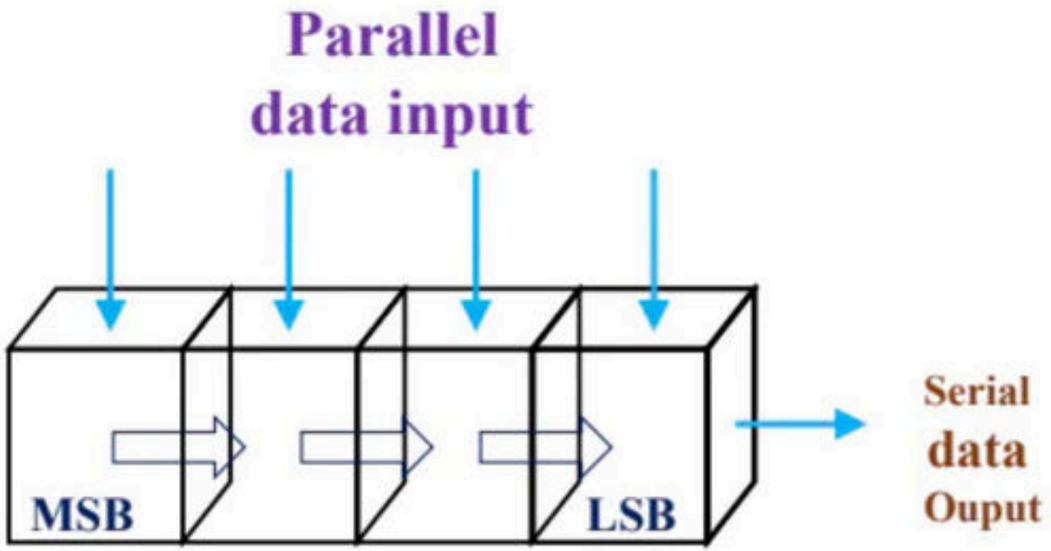
Parallel In Serial Out

Parallel In Serial Out



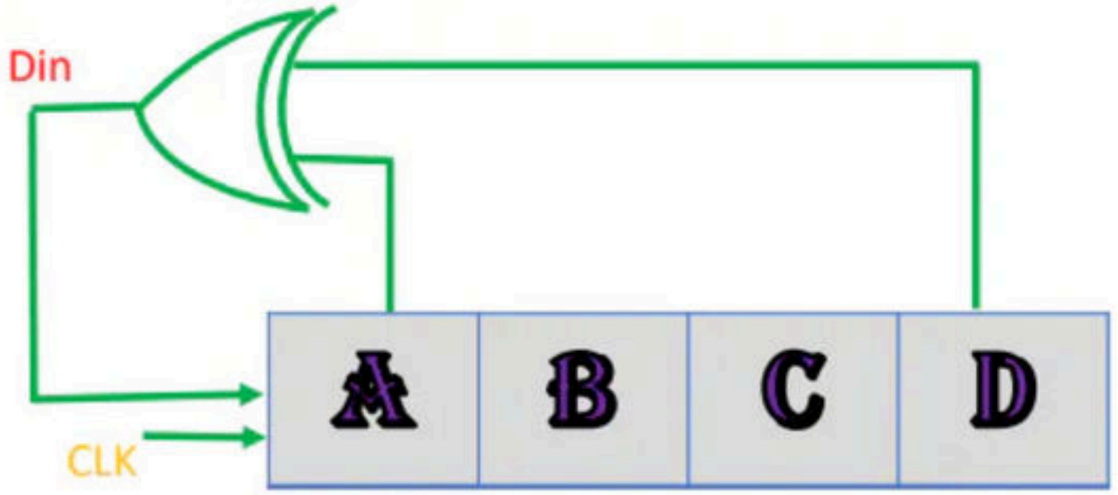
Parallel In Serial Out





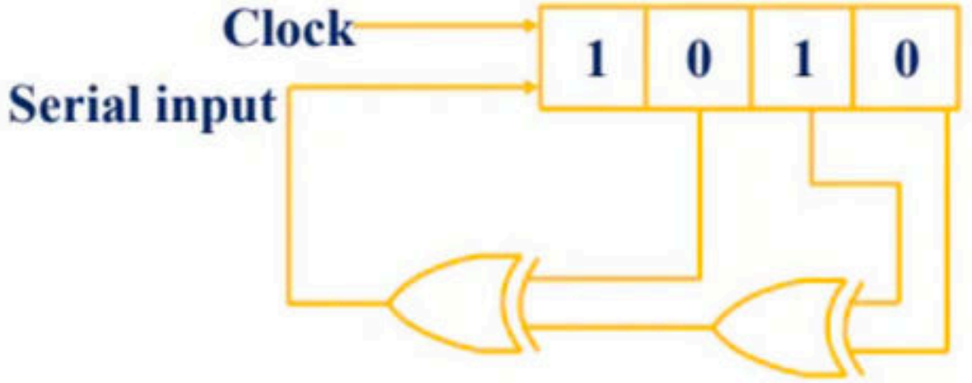
- PISO Configuration has only
  - 4- input
  - 1- output
- For PISO configuration
  - for storing = Clock pulses
  - for retrieving = Clock pulses
  - Total number clock pulses =

Q) A 4 – bit shift register circuit configured for right shift operation Din ----->A ,is shown , if the present state of the shift register is ABCD= 1101 , the number of clock cycles required to reach the state ABCD = 1111 is …..

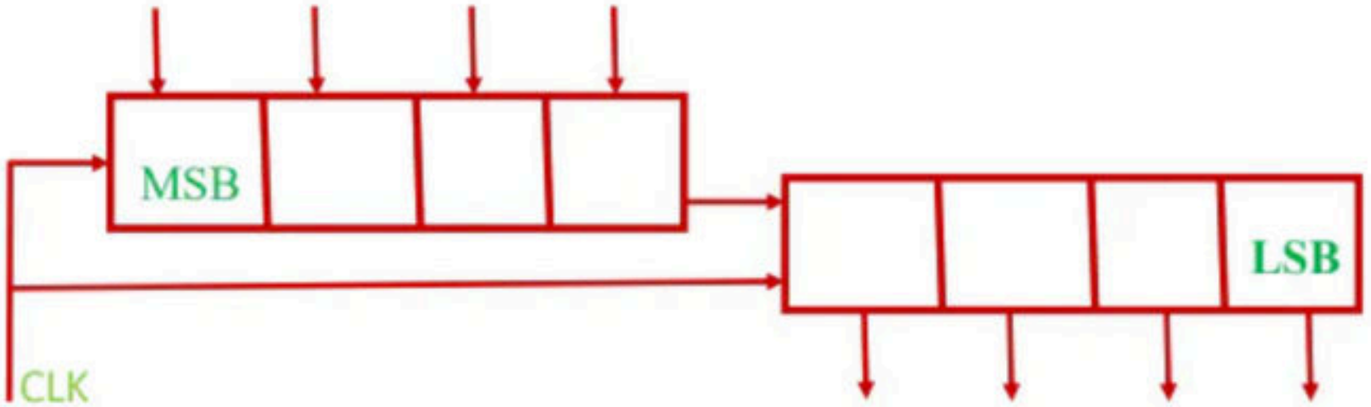


Q. The shift register shown in figure is initially loaded with the bit pattern 1010. Subsequently the shift register is clocked, and with each clock pulse the pattern gets shifter by one bit position to the right. With each shift, the bit at the serial input is pushed to the left most position (MSB). After how many clock pulses will the content of the shift register become 1010 again?

(A)3 (B) 7 (C) 11 (D) 15

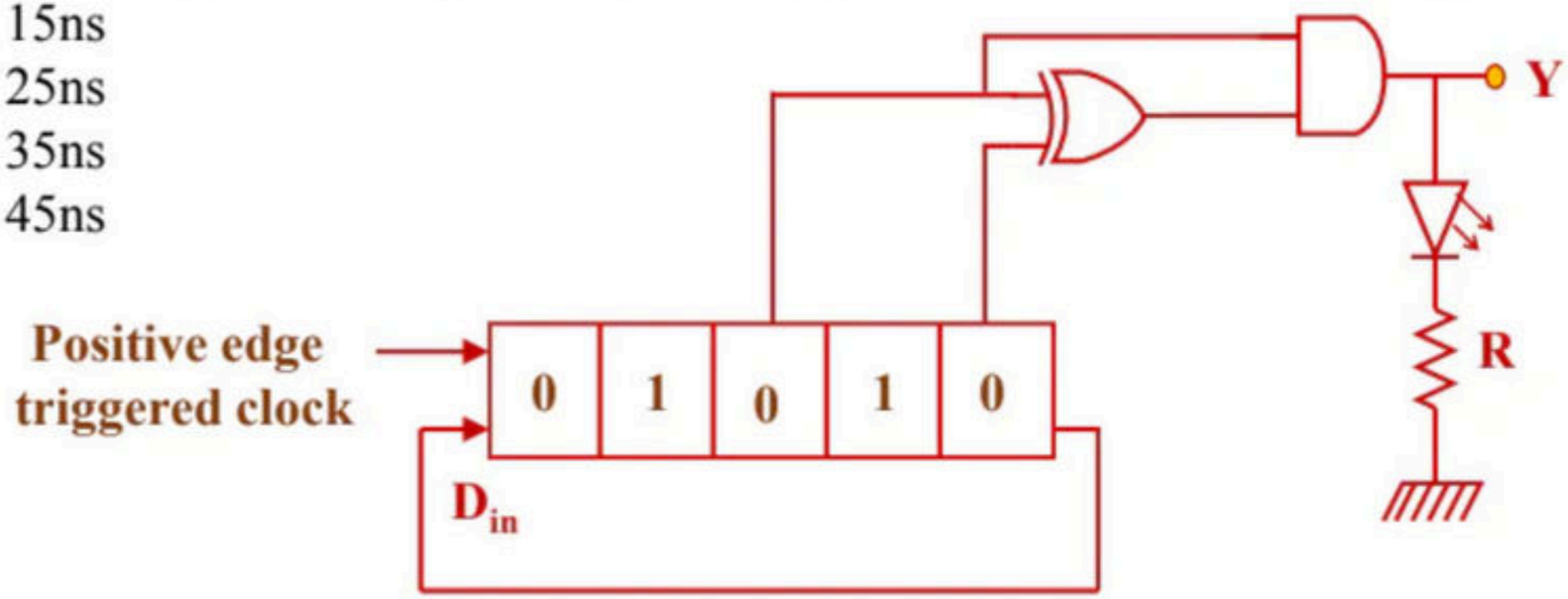


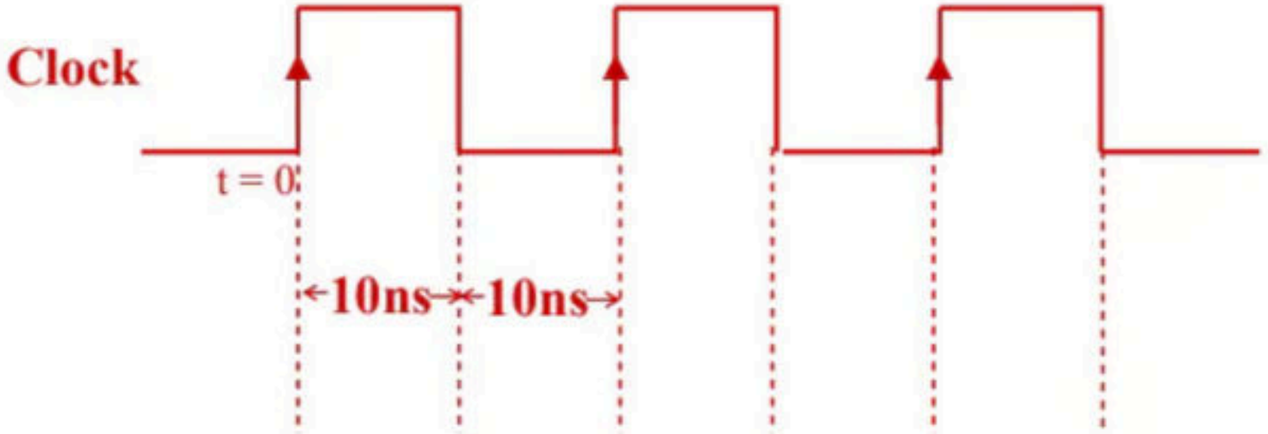
Q) An 8- bit register is made of one 4- bit PISO register (synchronous loading) cascaded with a 4- bit SIPO register as shown in the figure below , then total number of clock pulses required to perform write and read operations for one byte is -----



Q) Consider a serial in parallel out , right shift register circuit with initial contents as 01010 as shown in the figure . This circuit is operated with a positive edge triggered clock signal as given in figure . If +5V and 0V are used to represent logic-1 and logic-0 respectively , the at which of the following instances, the LED will be in ON state .

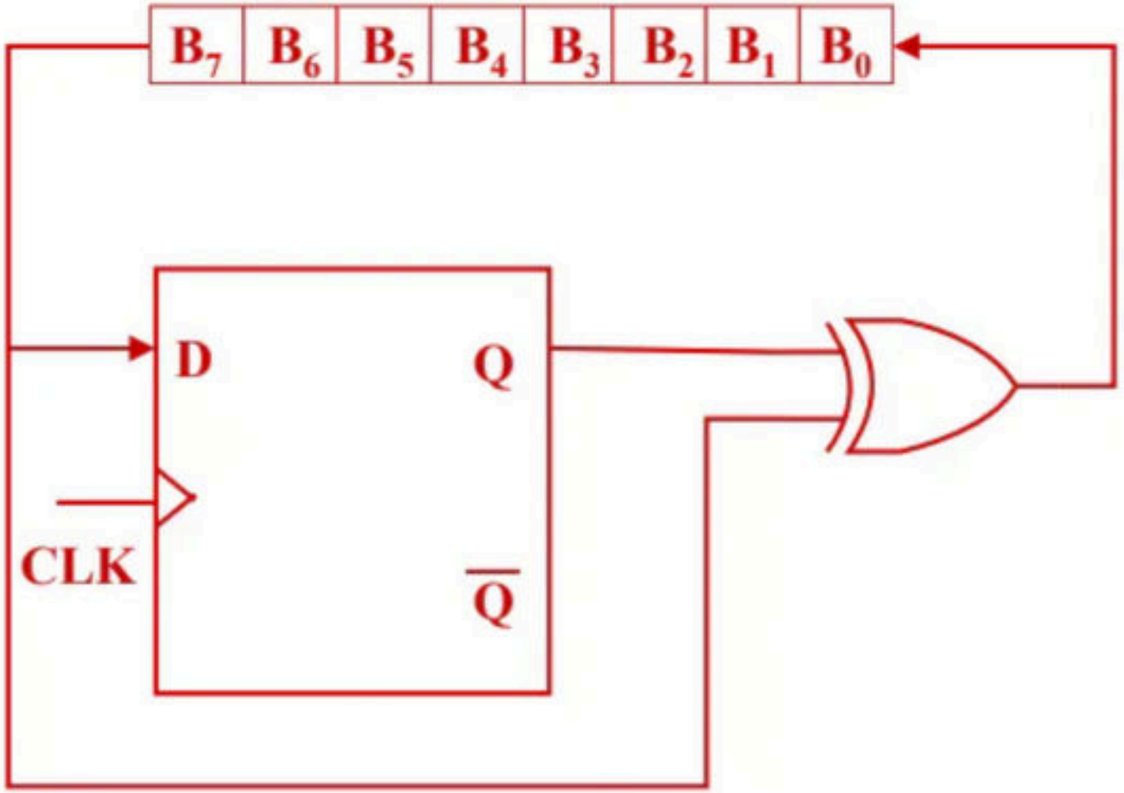
a) 15ns
b) 25ns
c) 35ns
d) 45ns





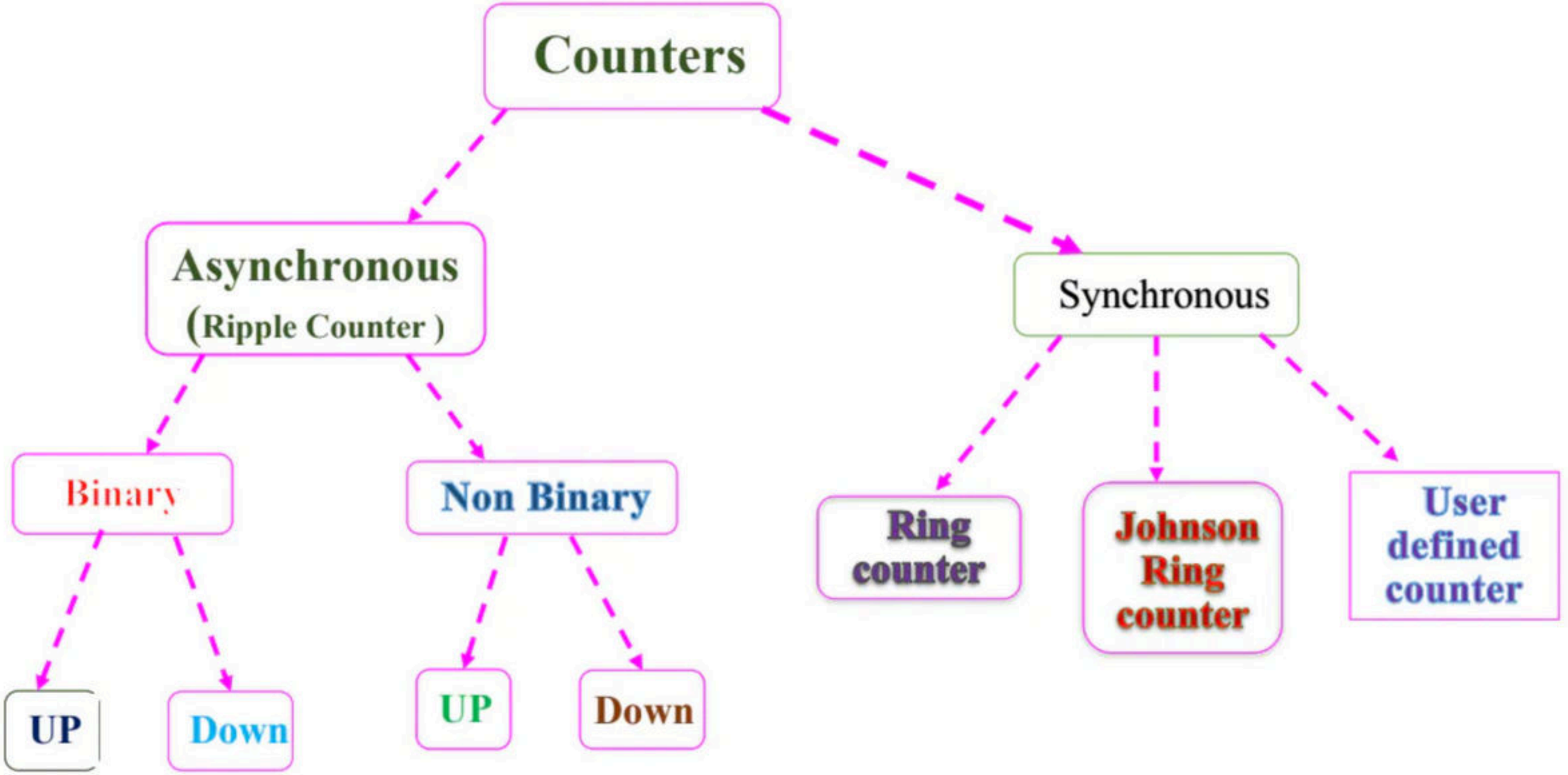
Q) An 8-bit register and D flip flop shown in figure below are synchronized with same clock , assuming the flip flop is initially cleared. The circuit act as a

a) Binary to 2's compliment converter
b) Binary to Gray code converter
c) Binary to 1's compliment converter
d) Binary to EX-3 code converter



Counters

Counters are the combination of various FFs , that generates a desired counting patterns when the clocks are applied


Counters
Asynchronous
(Ripple Counter)
Synchronous
Binary
Non Binary
Ring counter
Johnson Ring counter
User defined counter
UP
Down
UP
Down

Comparison of Asynchronous and Synchronous counters

Asynchronous counters	Synchronous counters
1.Different flip flops are applied with different clocks	1. All flip flops are applied with same clocks
2.Design and implementation is very simple even for more number of states	2.Design and implementation becomes tedious and complex as the number of states increases
3.Slower	3.Faster compared to Asynchronous counters
4.Transistion states are present	4.Transistion states are not present
5.Only fixed counting sequence is possible to implement ---->up counting ----->down counting	5. Any counting sequence is possible

State of a counter

Any possible output of a counter is known as its state , for a n– bit counter the maximum possible states are 2^n.

- The states which are counted by the counter are called as valid states .

- The states which are not counted (skipped) by the counter are called as invalid states .

Binary Counter and Non Binary Counter

If the counter counts all the possible states , with out skipping any states , then it is called as binary counter , otherwise non binary counter .

Modulus of a counter (Mod number)

The minimum number of clocks needed to get the complete counting pattern repeats is called as Modulus of a counter . (number of used states)

00 ------> 01 ----->10-----> 11 -------> 00-------->01 --------->10------>11

Q) What is the MOD number of the counting sequence .

00 --> 00 -->10--> 10 --> 01-->01 -->11-->11-->00 --> 00 -->10--> 10 --> 01-->01 -->

Q) What is the MOD number of the counting sequence .

00 --> 01 -->00--> 10 --> 00-->11 -->00-->01 -->00 --> 10 -->00--> 11 --

Design equation of counter

1FF ------->

2FF ------->

3FF ------->

By using n – FFs , the maximum possible states =


MOD- N
Counter


MOD- x
Counter
MOD- y
Counter
MOD- z
Counter

Asynchronous Counter

- Different FFs are applied with different clocks
- For only one FF external clock is applied ,which is LSB and output of one FF will acts as clock to next FFs
- FFs are operated in toggle mode
- Fixed counting sequence
    1. up counter
    2. down counter

3-Bit Ripple counter

- $Q_0 - Toggles\ for\ every\ \oplus ve\ edge\ of\ clock$
- $Q_1 - Toggles\ for\ every\ \oplus ve\ edge\ of\ Q_0$
- $Q_2 - Toggles\ for\ every\ \oplus ve\ edge\ of\ Q_1$

3-Bit Ripple Down counter

- $Q_0 - Toggles\ for\ every\ \ominus ve\ edge\ of\ clock$
- $Q_1 - Toggles\ for\ every\ \ominus ve\ edge\ of\ Q_0$
- $Q_2 - Toggles\ for\ every\ \ominus ve\ edge\ of\ Q_1$

Note :

1. ⊖ve Edge trigger and Q as a clock --------> Up counter

2. ⊖ve Edge trigger and $\bar{Q}$ as a clock --------> Down counter

3. ⊕ve Edge trigger and Q as a clock --------> Down counter

4. ⊕ve Edge trigger and $\bar{Q}$ as a clock --------> Up counter

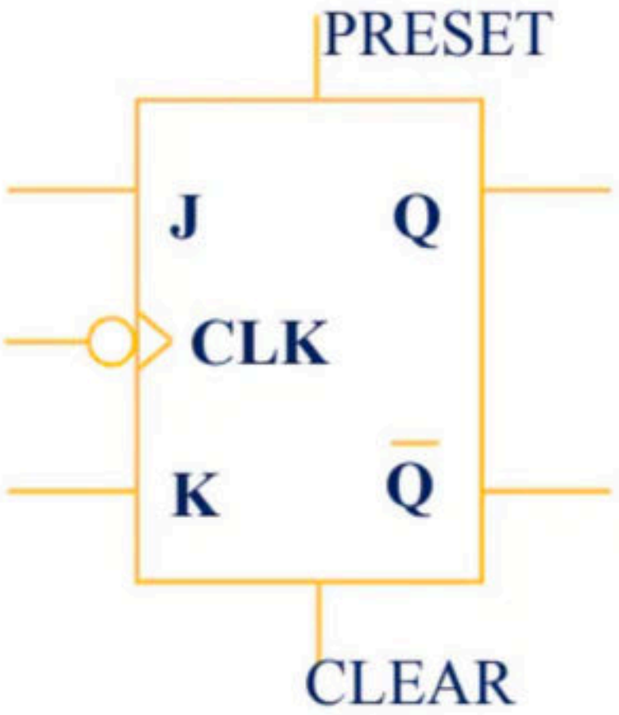
State Diagram

- The disadvantages of the ripple counter is that transition states are present due to delay of the FF (Decoding errors) .
- If only one FF changes its state ,then no transition states will be present , if more than one FF changes its states than transition states present.
- To avoid decoding errors *strobe signal* is used .
- Strobe signal is kept low for 3tpd , for 3- bit counter , so that transition states are not reflected, and after 3tpd strobe signal is made high .

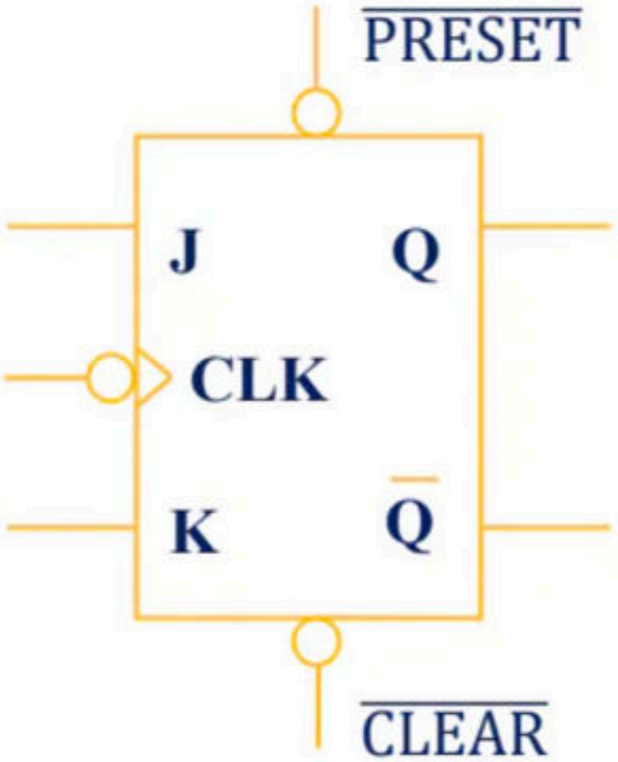
3-Bit Ripple Up/Down counter

Asynchronous Clear
and
Asynchronous Preset

- In order to reduce the number of states, feedback signal is applied to counter through CLEAR or PRESET signal
- CLEAR control is used to Reset the Flip Flop
- PRESET control is used to set the Flip Flop
- Asynchronous Clear and Preset , are independent of clock signal .
- Clear and Preset are called as over writing pins



Asynchronous Pre-set	Asynchronous Clear	FF response



Asynchronous $\overline{PRESET}$	Asynchronous $\overline{CLEAR}$	FF response

Q) Design a MOD-6 up counter

Q) Design a MOD-6 down counter

Q) Design a MOD-5 up counter

Q) Design a BCD up counter

Delay analysis(Asynchronous counter)

If the delay of each FF is t_{pd}, then the over all delay for

n- bit asynchronous counter is =

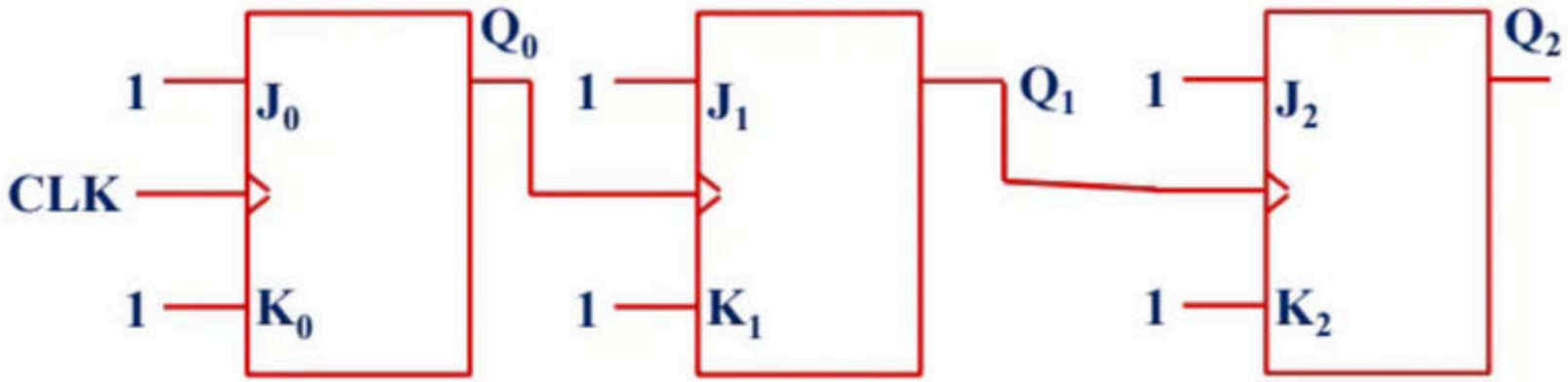
Q)Find

a) MOD number

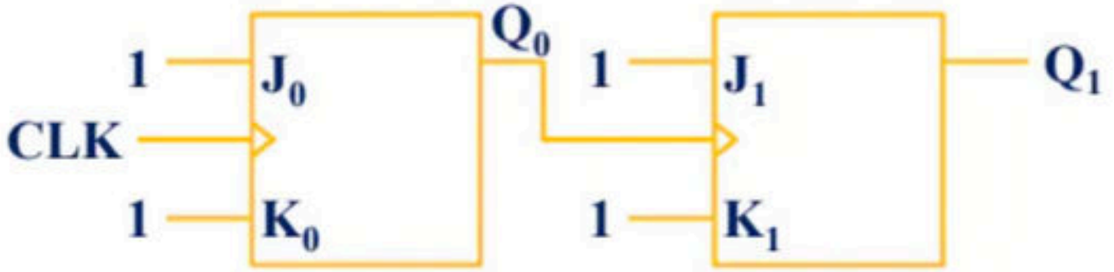
b) Output frequency of the counter , if the input frequency is 10MHz

c) if the delay of each FF is 50 ns , then the maximum frequency of the clock

d) state of the counter after 500 clocks if the initial state is 100



Q. For each of the positive edge-triggered J-K flip flop used in the following figure; the propagation delay is ΔT. Which of the following waveforms correctly represents the output at Q_1?

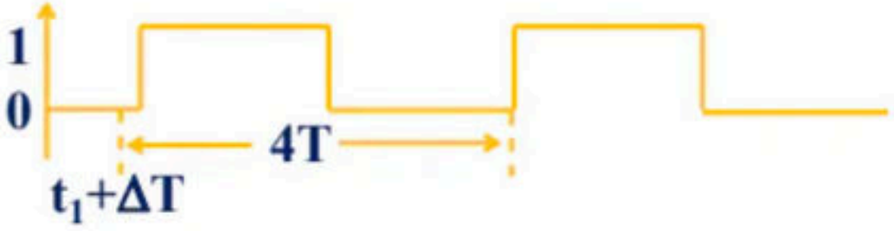


(a)

(b)

(c)

(d)

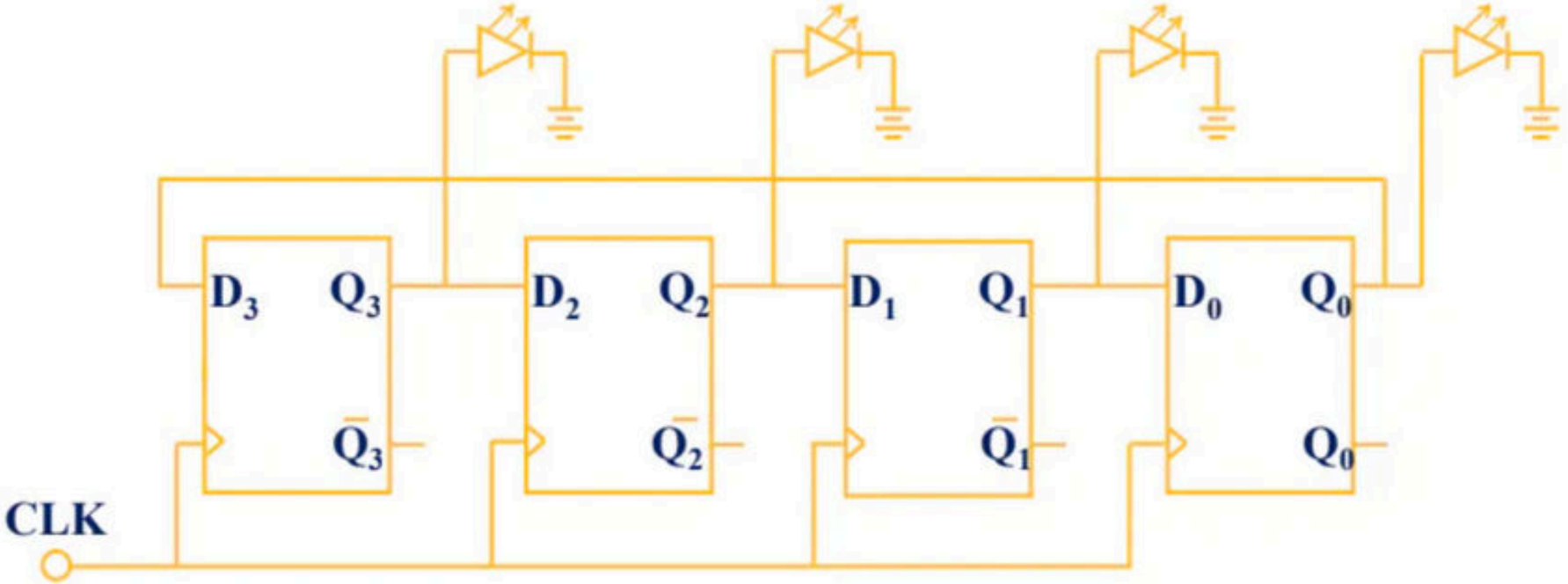


Synchronous Counter

Synchronous Counter

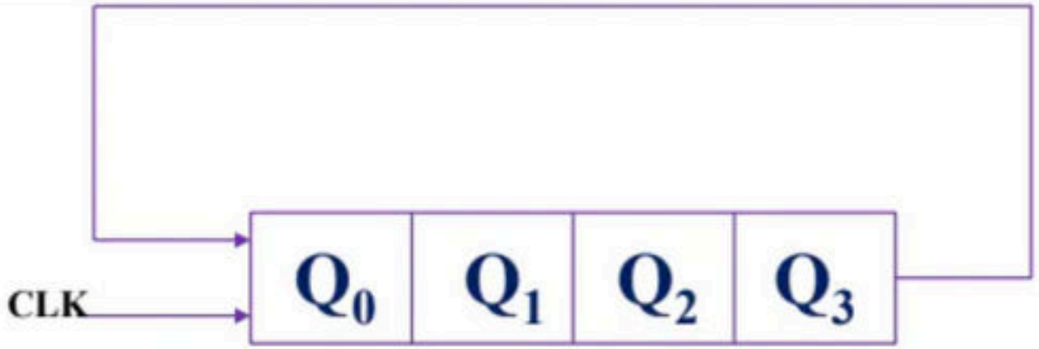
1. Ring Counter
2. Johnson Ring Counter
3. User defined Counter

4–Bit Ring Counter



- Ring counter is a synchronous counter , it is a shift register in which last FF output is connected to the first FF input .

- Ring counter performs right shift operation .



State Diagram

CLK	Q_0	Q_1	Q_2	Q_3	Decoding logic

Timing Diagram

Repeating pattern

Clk	Q_0

$T_{Q_0} =$

$f_{Q_0} =$

D =

Clk	Q_1

$T_{Q_1} =$

$f_{Q_1} =$

D =

Clk	Q_2

$T_{Q_2} =$

$f_2 =$

D =

Clk	Q_3

$T_{Q_3} =$

$f_{Q_3} =$

D =

For 4- bit Ring counter

- Used states =

- Unused states =

- The phase shift between successive wave form =

- If the delay of each Flip Flop is t_{pd} , then over all delay ,

- If the Flip Flops are having different delay then over all delay ,

Q) What happens if the Ring counter will enters into any of its unused states

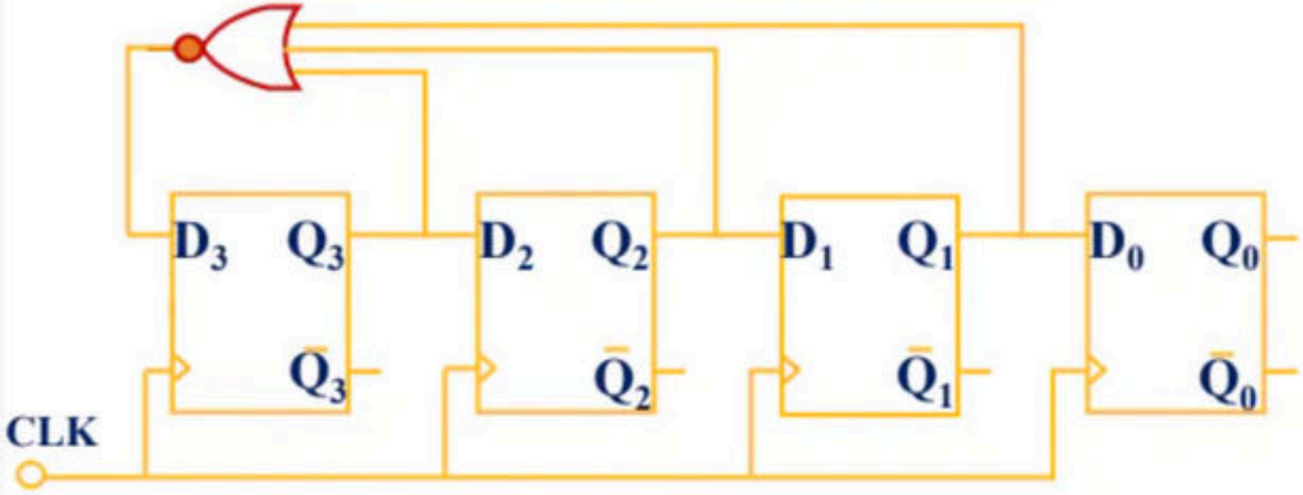
Advantage

- Decoding logic of ring counter is simple and does not require any external logic circuit .

Draw back of Ring counter

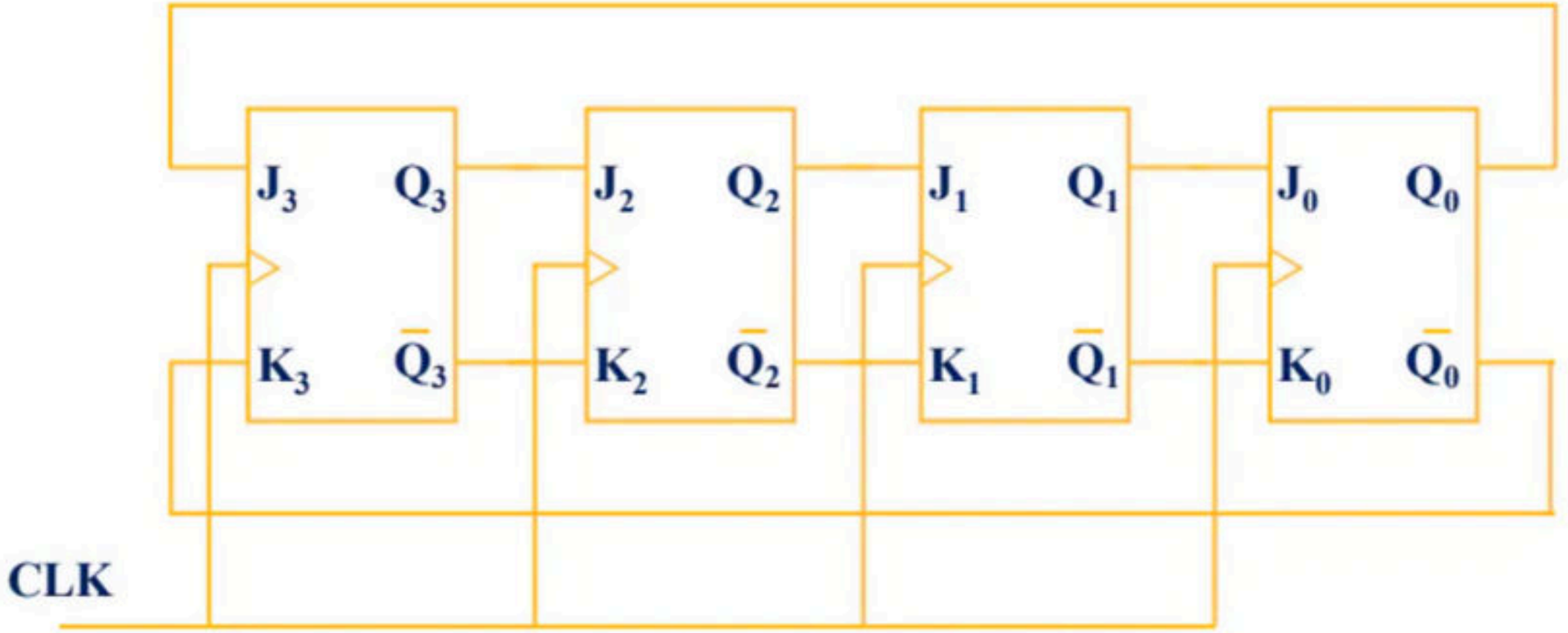
- If all the outputs of FFs initially zero , then the Ring counter does not start .
- If more than one FF outputs' are high initially , then the ring counter enters into unused state and never come out of unused state , this is called as **Lock out problem .**

Self Starting Ring counter

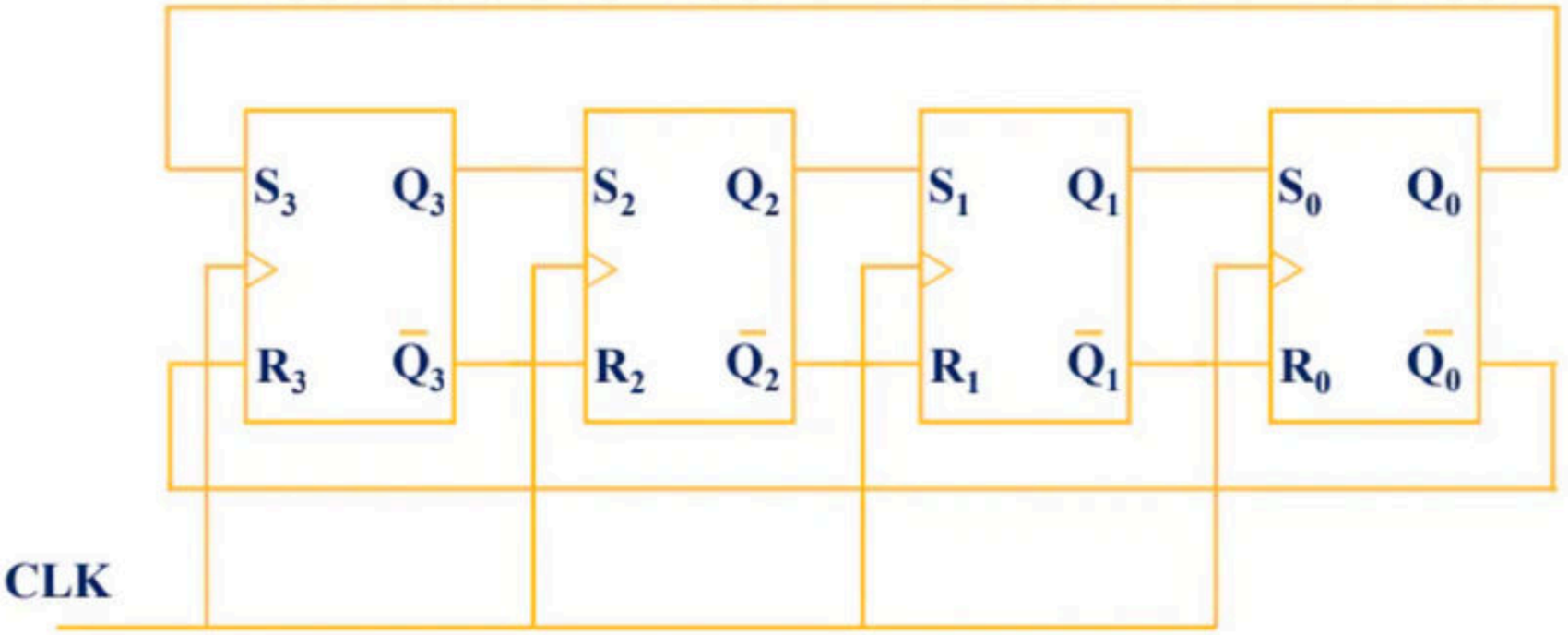


CLK	input	Q_3	Q_2	Q_1	Q_0

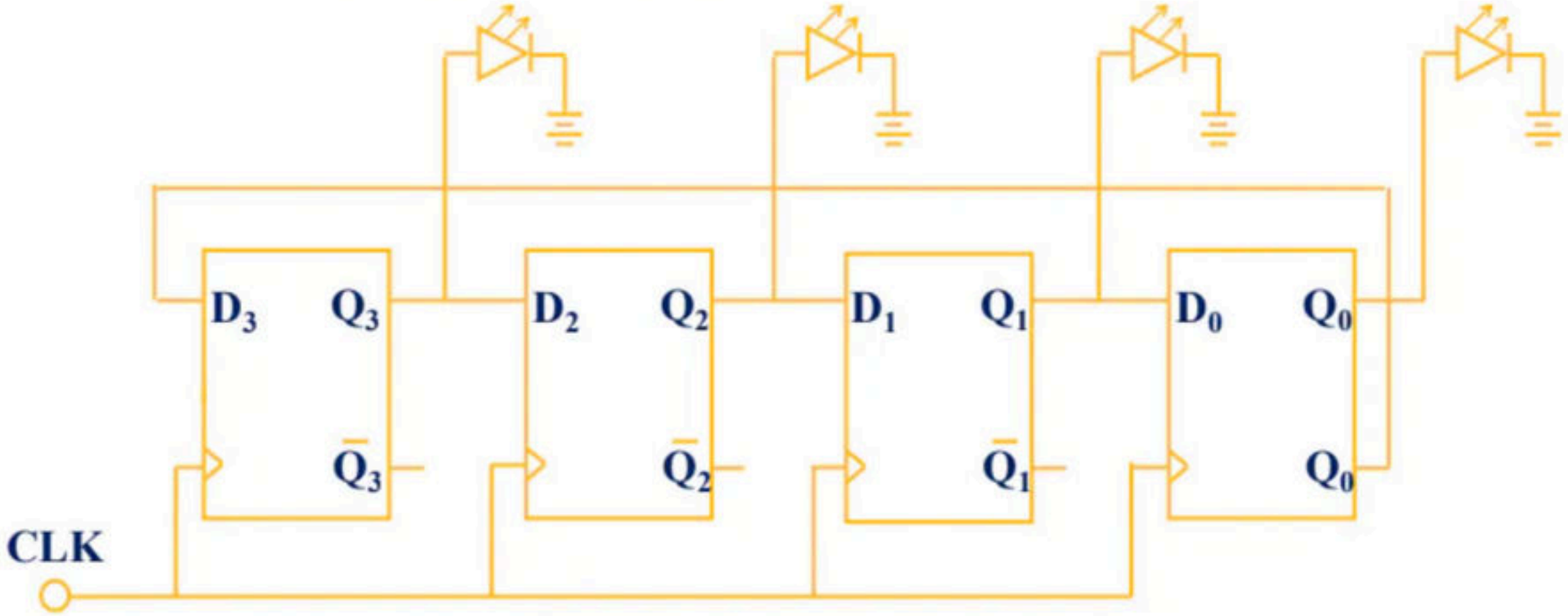
Ring counter using JK flip flop

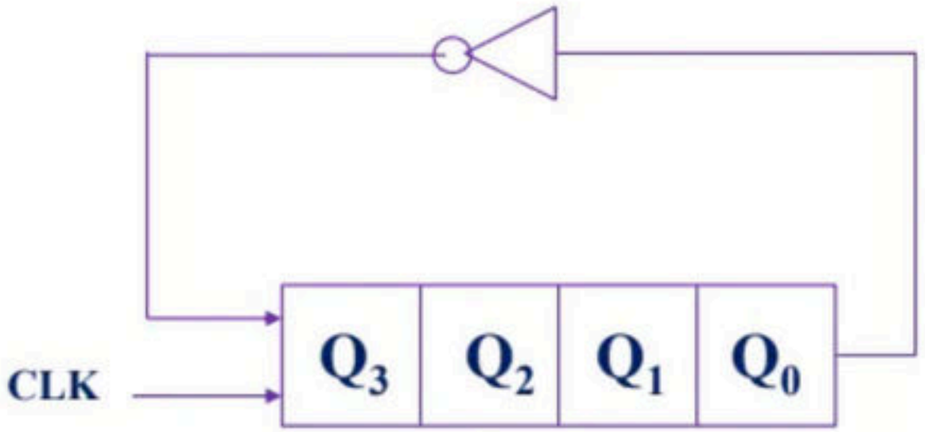


Ring counter using SR flip flop



Johnson Ring counter





CLK	Q_3	Q_2	Q_1	Q_0	Decoding logic

Repeating pattern

Clk	Q_0

$T_{Q_0} =$

$f_{Q_0} =$

D=

Clk	Q_1

$T_{Q_1} =$

$f_{Q_1} =$

D=

Clk	Q_2

$T_{Q_2} =$

$f_{Q_2} =$

D=

Clk	Q_3

$T_{Q_3} =$

$f_{Q_3} =$

D=

For 4- bit Johnson Counter

- Used states =
- Unused states =
- Mod No =
- Frequency of each FF =

For n- bit Johnson Counter

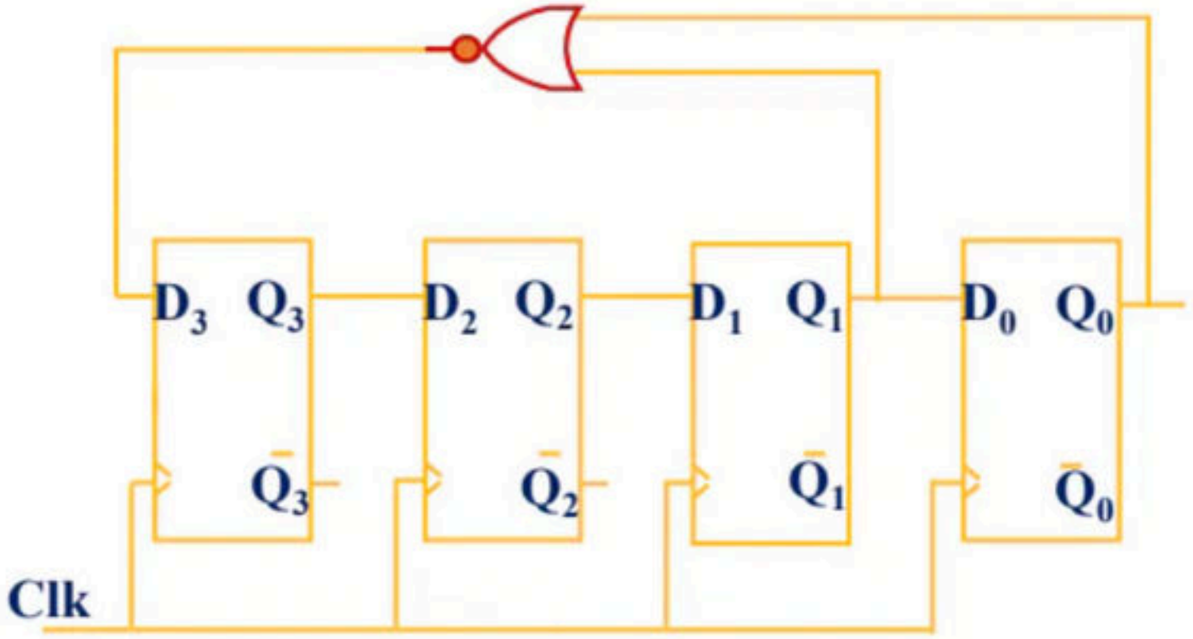
- Number of used states =
- Number of unused states =
- Mod No =
- Frequency of each FF =

Q) What happens when Johnson counter enters into any one of its unused states .

Draw back of Johnson Ring counter

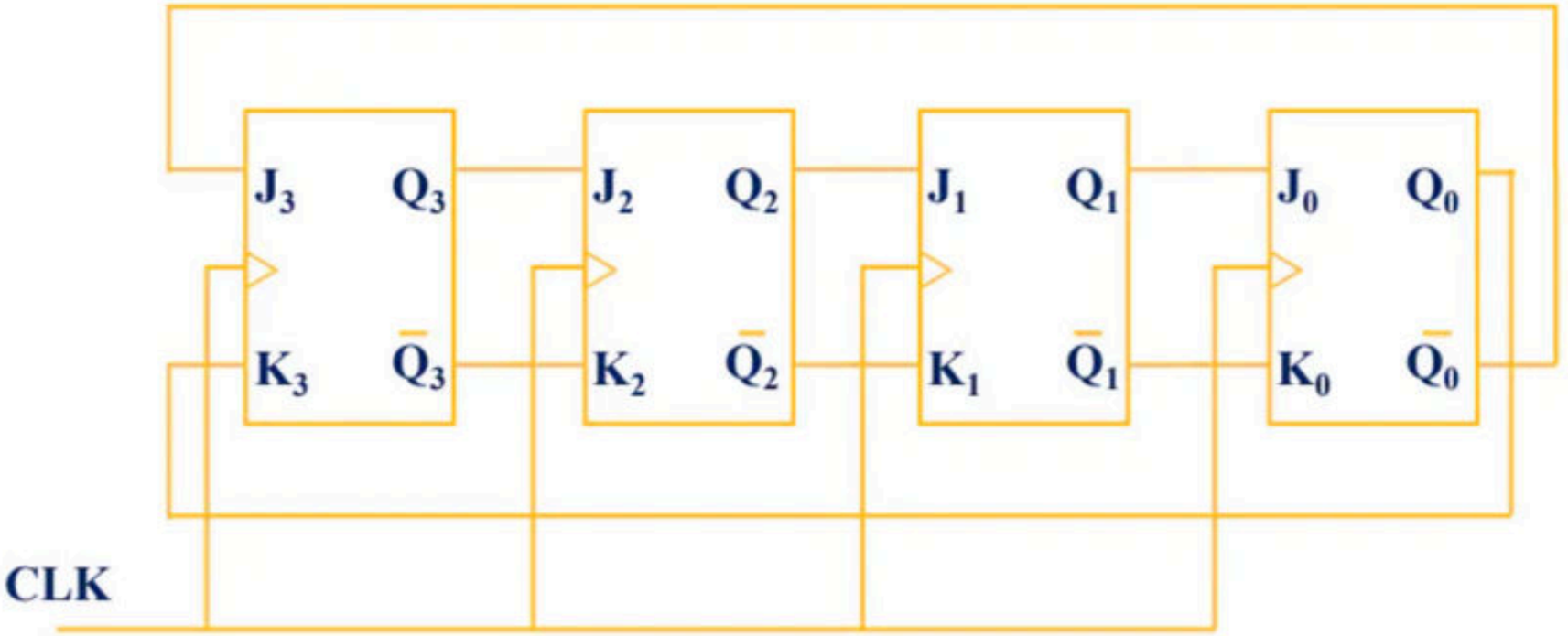
If the Johnson counter enters into any of its unused state , it completely stay in the unused states only .

Johnson Ring counter to prevent lock out problem



CLK	Q_3	Q_2	Q_1	Q_0	Decoding logic

Johnson Ring counter with JK FF



Johnson Ring counter

Twisted Ring counter

Switch tail counter

Walking Counter

Creeping counter

Mobies counter

Ring counter	Johnson ring counter
1. Mod No =	1. Mod No =
2. Number of used states= Number of unused states =	2. Number of used states= Number of unused states =
3.Time period of each FF =	3.Time period of each FF =
4. Frequency of each FF =	4. Frequency of each FF =
5. Suffer from lock out problem	5. Suffer from lock out problem
6.Decoding logic is simple	6.Decoding logic requires AND and NOR gates

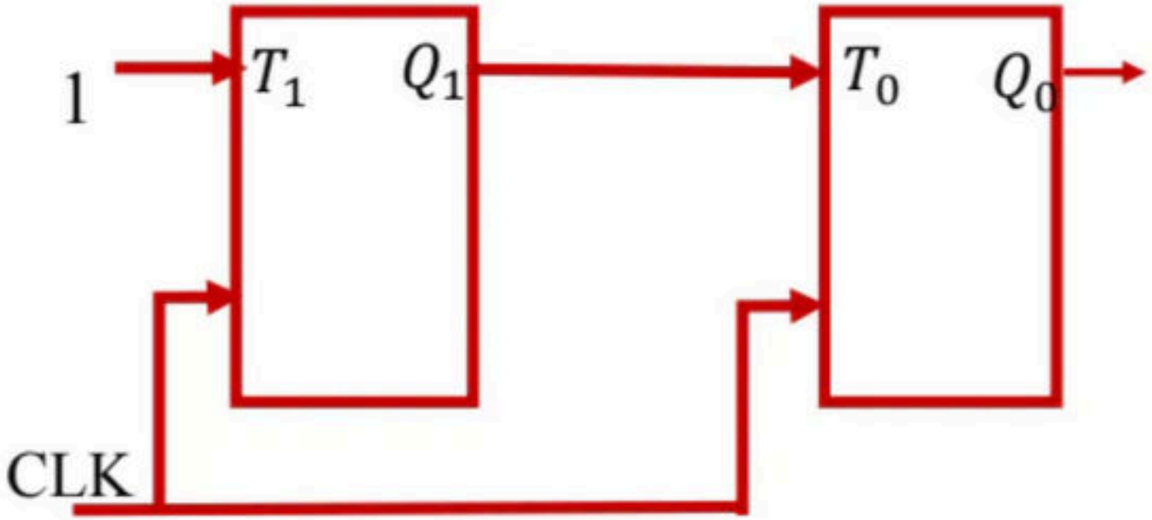
User defined counter design

Q) Design synchronous counter , whose counting sequence is

Q_1Q_0 = 00---> 10 ---> 01---> 11 --->00----> 10 ….. By using T- FF

Q) Design synchronous counter , whose counting sequence is

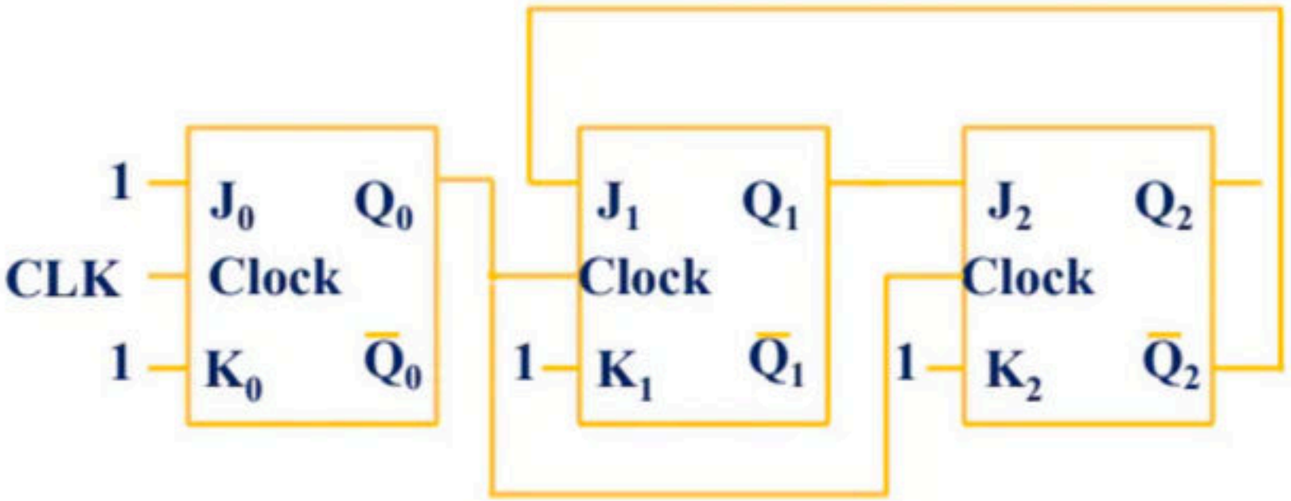
Q_1Q_0 = **00 ----> 10 ---> 01----> 11 ------>00-----> 10 …..** By using D- FF



Q) Find the counting sequence of the following
If the initial state of the counter $Q_1Q_0 = 00$ then the state of counter after

a) 236 clocks b) 251 clocks c) 333 clocks

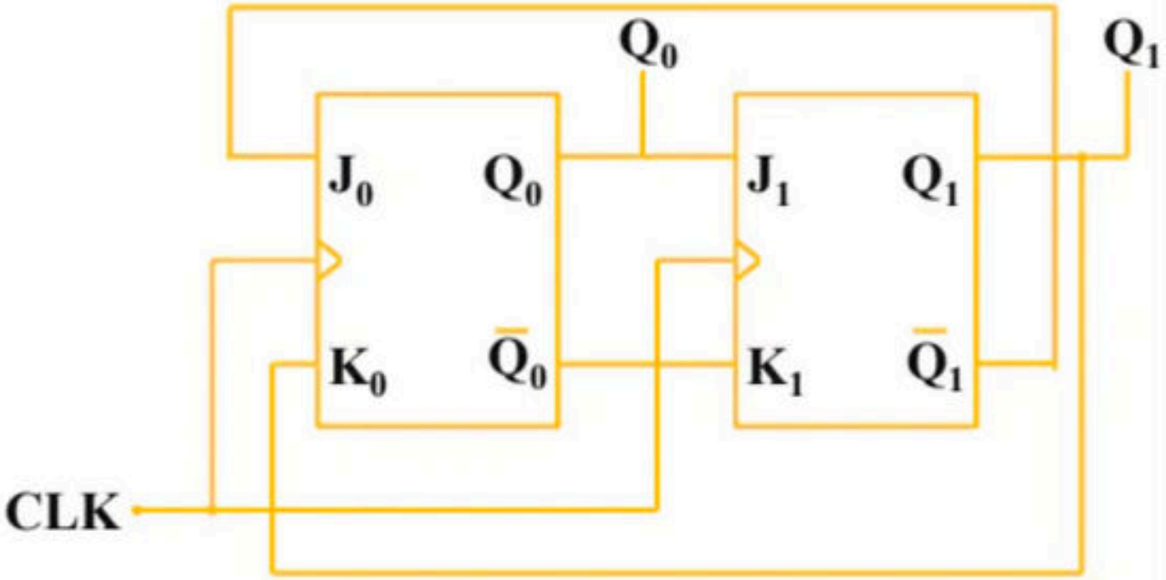
Q). The figure shown a digital circuit constructed using negative edge triggered J-K flip flops. Assume a starting state of $Q_2Q_1Q_0$ will repeat after _____ number of cycles of the clocks



Q) In the following sequential circuit, the initial state (before the first clock pulse) of the circuit is $Q_1Q_0 = 00$. The state (Q_1Q_0), immediately after the 333^{rd} clock pulse is.

(A)00 (B) 01

(C) 10 (D) 11



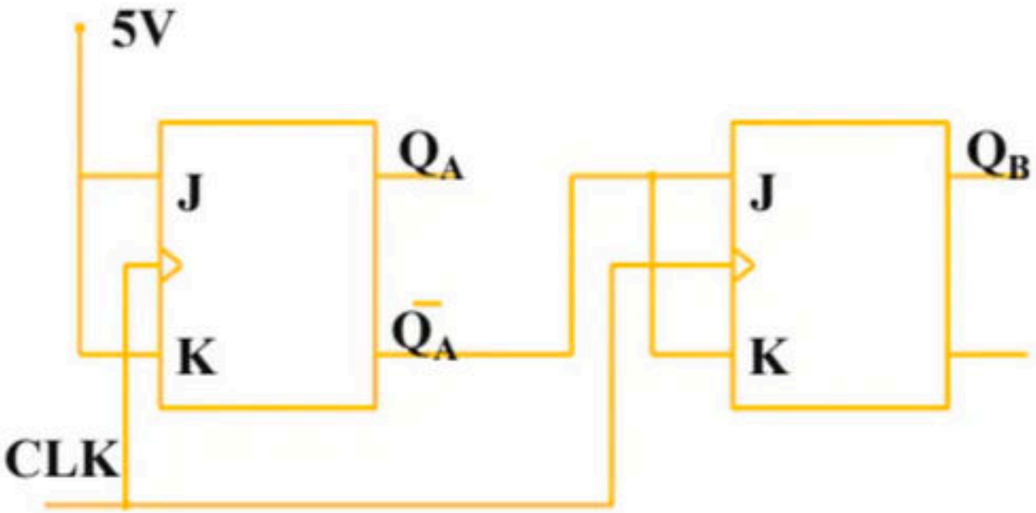
Q). The current state Q_A Q_B of a two JK flip-flop system is 00. Assume that the clock rise-time is much smaller than the delay of the JK flip-flop. The next state of system is.

(A) 00

(B) 01

(C) 11

(D) 10



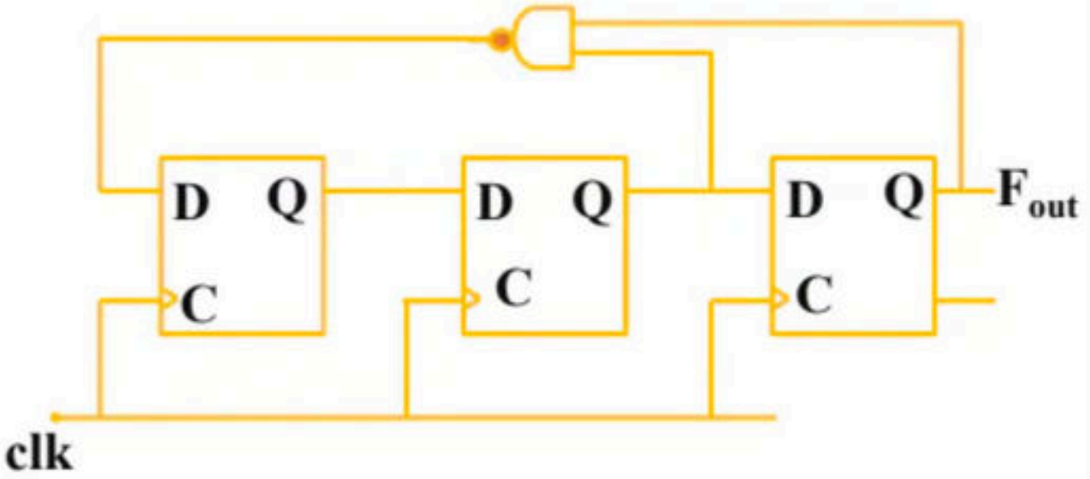
Q) . Which one of the following statements is true about digital circuit shown in the figure?

(a)It can be used for dividing the input frequency by 3.

(b)It can be used for dividing the input frequency by 5.

(c)It can be used for dividing the input frequency by 7.

(d)It cannot be reliably used as a frequency divider due to disjoint internal cycles.

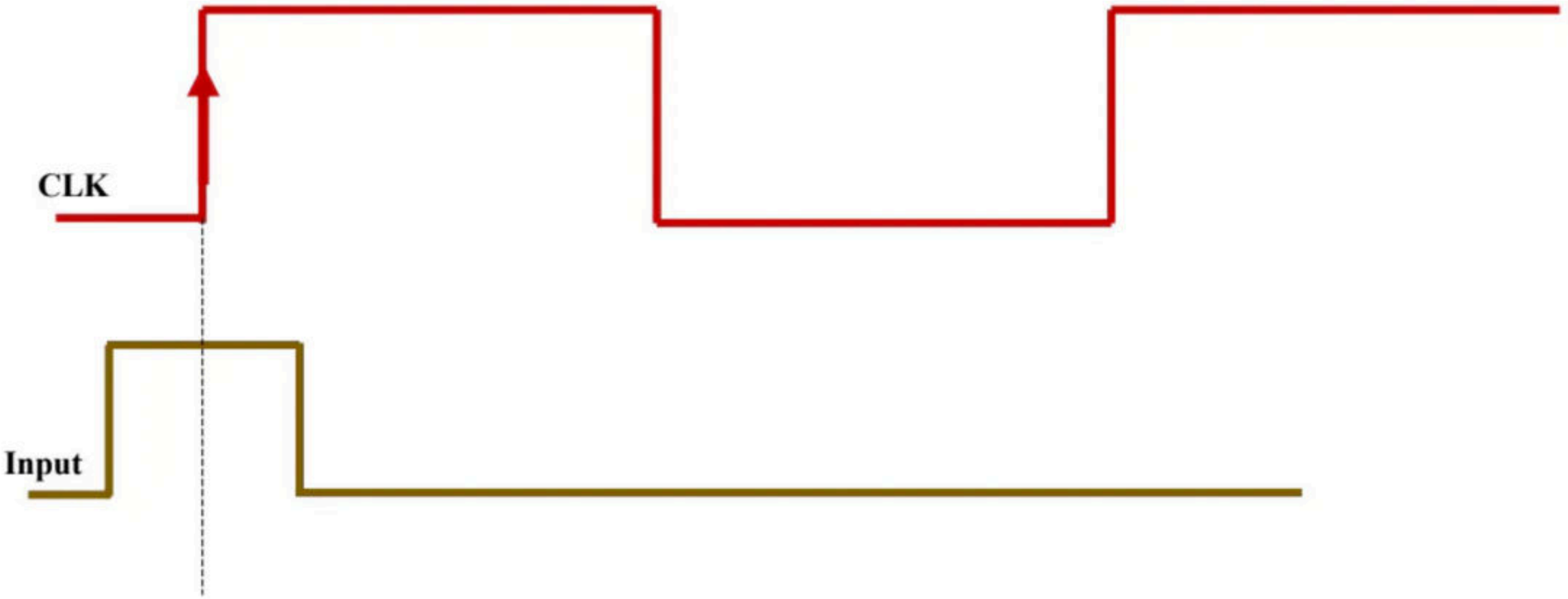


Set up time

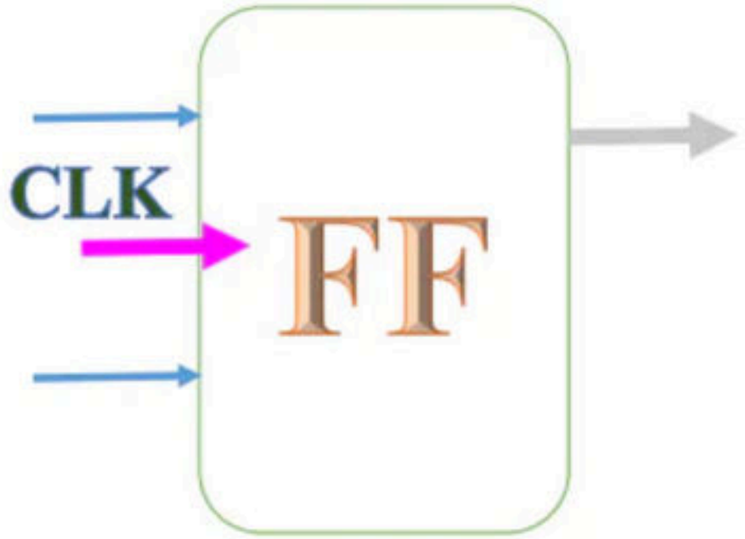
It is the minimum amount of time before the active edge of the clock, the input signal should remain constant

Hold time

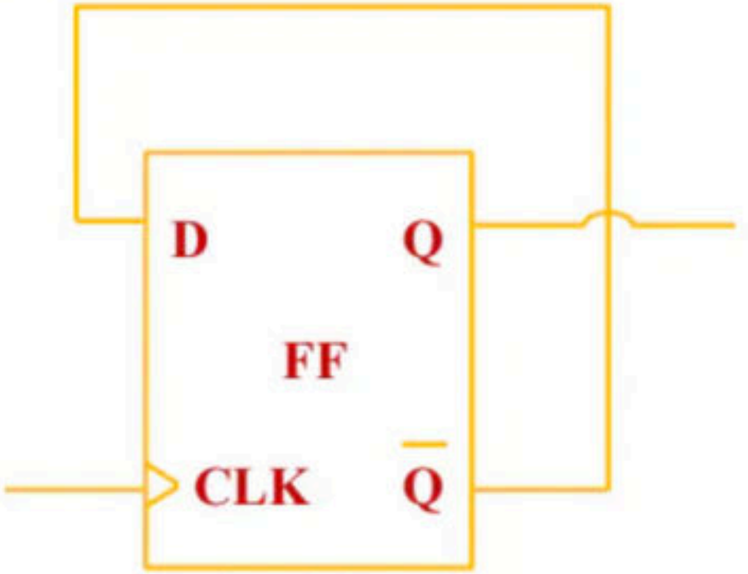
It is the minimum amount of time after the active edge of the clock, the input signal should remain constant


CLK
Input

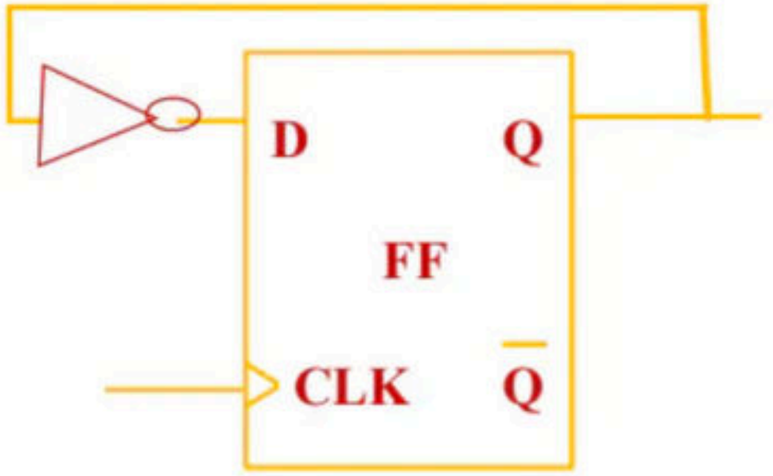
Delay in single FF



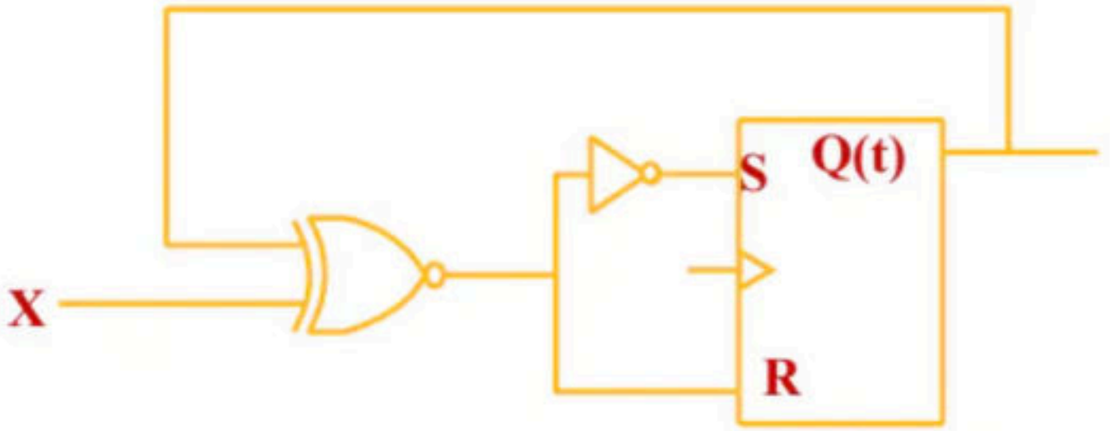
Q) Find the maximum frequency of operation for the following circuit, if the propagation delay of FF is 20ns and setup time and hold time are 10ns each .



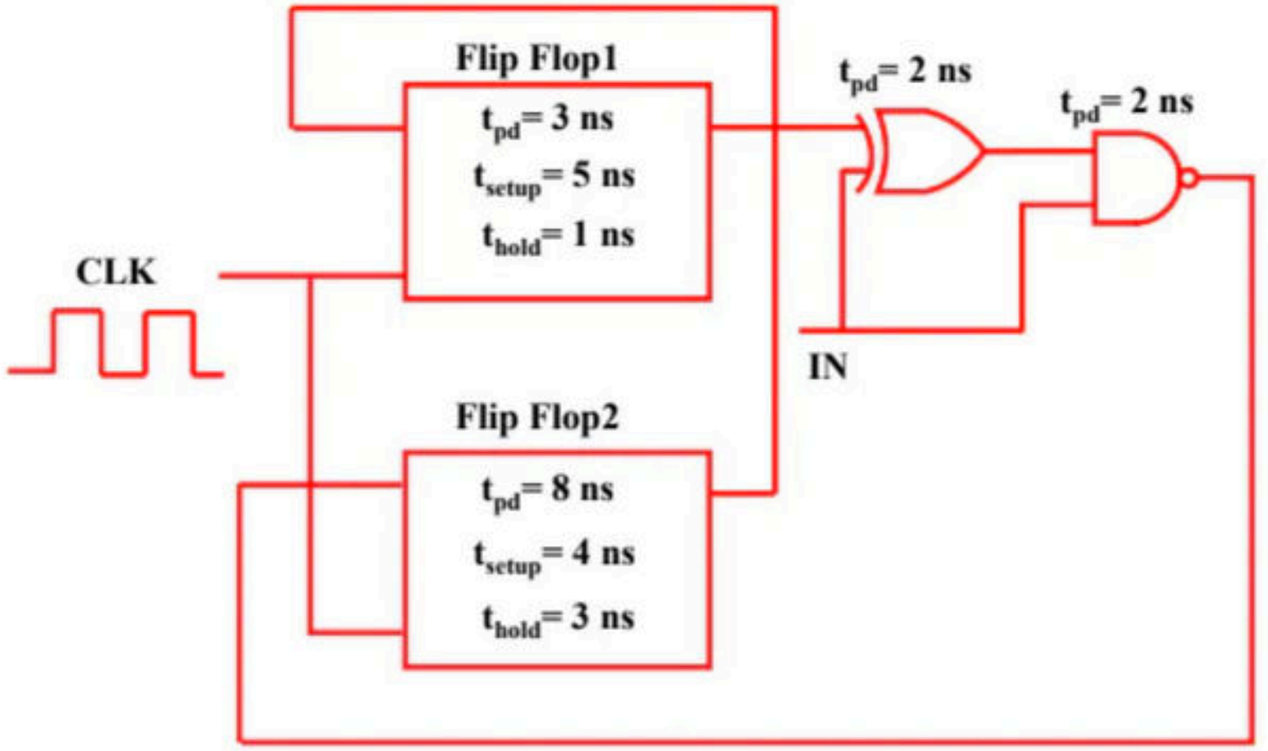
Q) Find the maximum frequency of operation for the following circuit, if the propagation delay of FF is 20ns and inverter is 5ns .



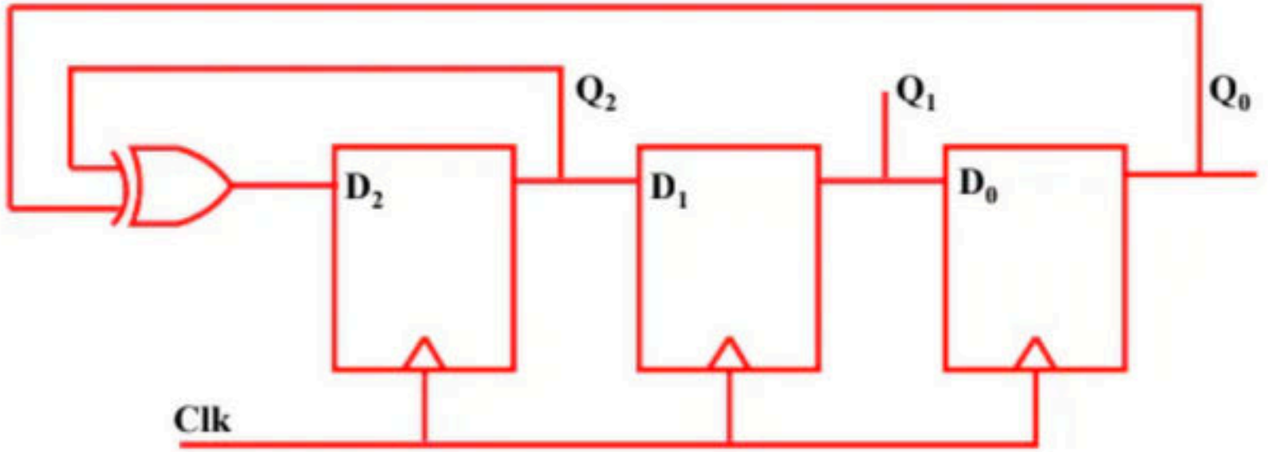
Q) Find the maximum frequency of operation for the following circuit, if the propagation delay of FF is 20ns , XNOR gate is 10ns and inverter is 5ns .



Q. For the components in the sequential circuit shown below, t_{pd} is the propagation delay, t_{setup} is the setup time, and t_{hold} is the hold time. The maximum clock frequency (rounded off to the nearest integer), at which the given circuit can operate reliably, is_____ MHz.



Q. The propagation delay of the exclusive-OR (XOR) gate in the circuit in the figure is 3 ns. The propagation delay of all the flip-flops is assumed to be zero. The clock (Clk) frequency provided to the circuit is 500 MHz. Starting from the initial value of the flip-flop outputs $Q_2Q_1Q_0 = 111$ with $D_2 = 1$, the minimum number of triggering clock edges after which the flip-flop outputs $Q_2Q_1Q_0$ becomes 100 (in integer) is _____



FINITE STATE MACHINE

FINITE STATE MACHINE

Synchronous Sequential circuits are also called as Finite State Machine (FSM)

There are two types of FSMs

1.Mealy State Machine

2. Moore State Machine

Mealy State Machine

The output of Mealy State Machine is a function of present state as well as present input

$$Z(t) = f\,[s(t)\,,\, x(t)]$$

State Diagram

The state diagram or state graph is a pictorial representation of the relationships between the present state , the input , the next state and the output of a sequential circuits, i.e the state diagram is a pictorial representation of the behavior of a sequential circuit

State Diagram

State table

Present state	NS , O/P	
	X =0	X= 1
a	a , 0	b , 0
b	b, 1	c,0
c	d, 0	c , 0
d	d,0	a , 1

Moore State Machine

The output of Moore State Machine is a function of present state only

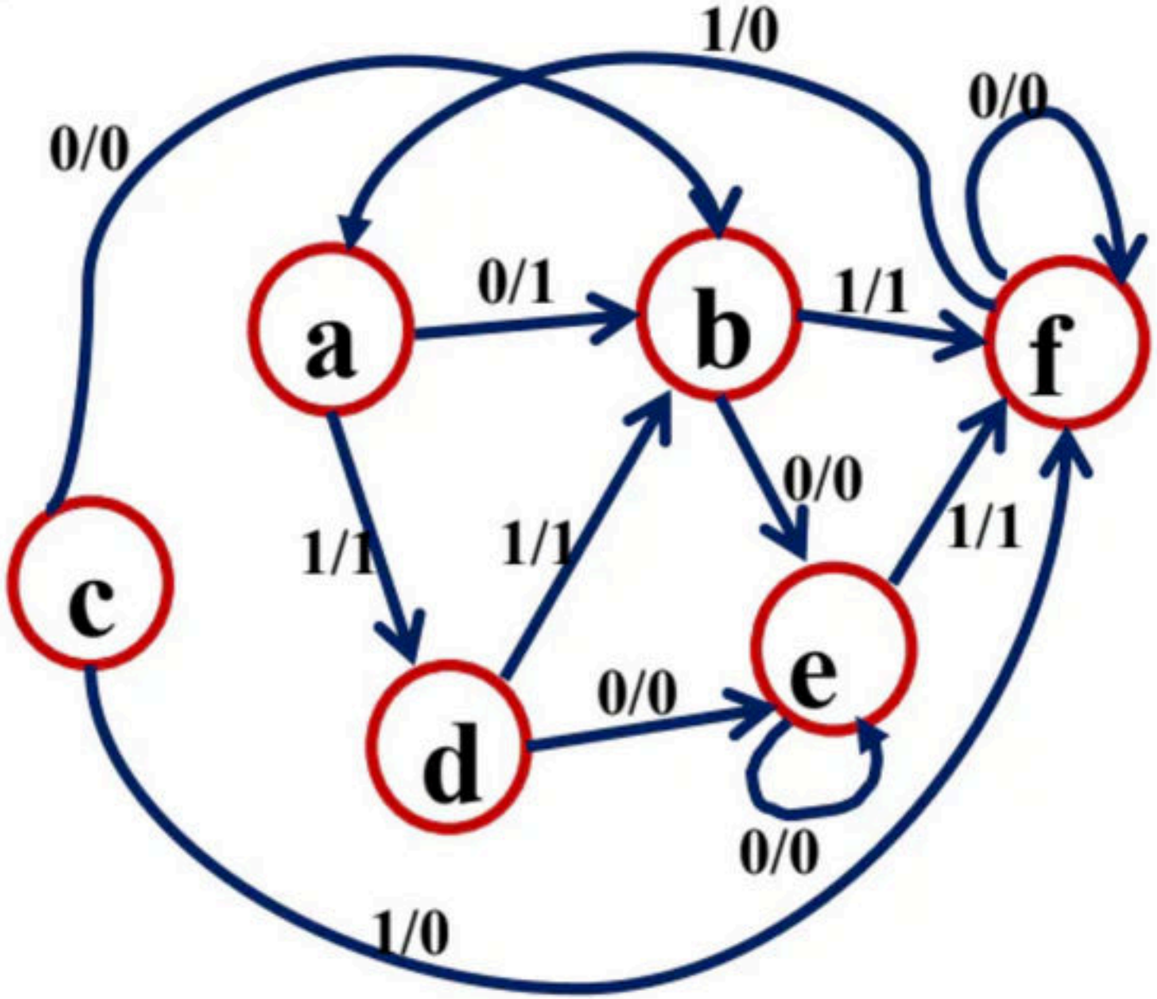
$$Z(t) = f[\, s(t) \,]$$

State Diagram

State table

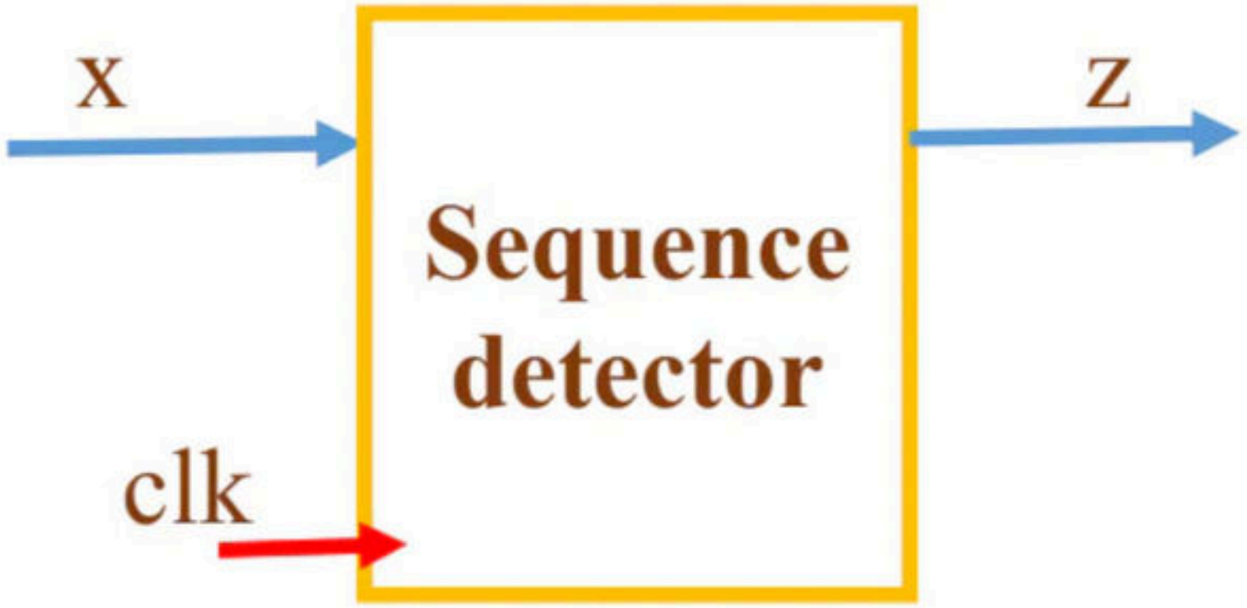
Present state	Next State		Output
	X =0	X= 1	
a	a	b	0
b	b	c	0
c	d	c	0
d	a	d	1

Q. Find the reduced state table and reduced state diagram



Sequence Detector

A Sequence detector is sequential machine which produces an output 1 every time the desired sequence is detected and an output 0 at all other times.



There are two types of sequence detector

1. Over lapping sequence detector
2. Non over lapping sequence detector

Q. Find the output of the non over lapping sequence detector to detect 1101 for the input sequence

X= 1101101101

Q. Find the output of the over lapping sequence detector to detect 1101 for the input sequence

X= 1101101101

Q. Find the output of the non over lapping sequence detector to detect 11011 for the input sequence

X= 11011101101

Q. Find the output of the over lapping sequence detector to detect 11011 for the input sequence

X= 1101101101

Q) A sequence detector is designed to detect precisely 3 digital inputs, with overlapping sequences detectable, for the sequence (1,0,1) and input data (1,1,0,1,0,0,1,1,0,1,0,1,1,0,) what is the output of this detector

State Diagram to detect the given sequence

We can develop the state diagram to detect the given sequence by using two modals

1.Mealy modal

2.Moore modal

Mealy modal

To detect n – bit sequence by using Mealy modal n- number of states are required

Moore modal

To detect **n – bit sequence** by using Moore modal **(n+1)** number of states are required

Q. Develop the non overlapping sequence detector and state table to detect 1101 by using mealy modal

Q. Develop the overlapping sequence detector and state table to detect 1101 by using mealy modal

Q. Develop the non overlapping sequence detector and state table to detect 1101 by using Moore modal

Q. Develop the overlapping sequence detector and state table to detect 1101 by using Moore modal

Q. Develop the non overlapping sequence detector and state table to detect 1010 by using mealy modal

Q. Develop the overlapping sequence detector and state table to detect 1010 by using mealy modal

Q. Develop the non overlapping sequence detector and state table to detect 1010 by using Moore modal

Q. Develop the overlapping sequence detector and state table to detect 1010 by using Moore modal

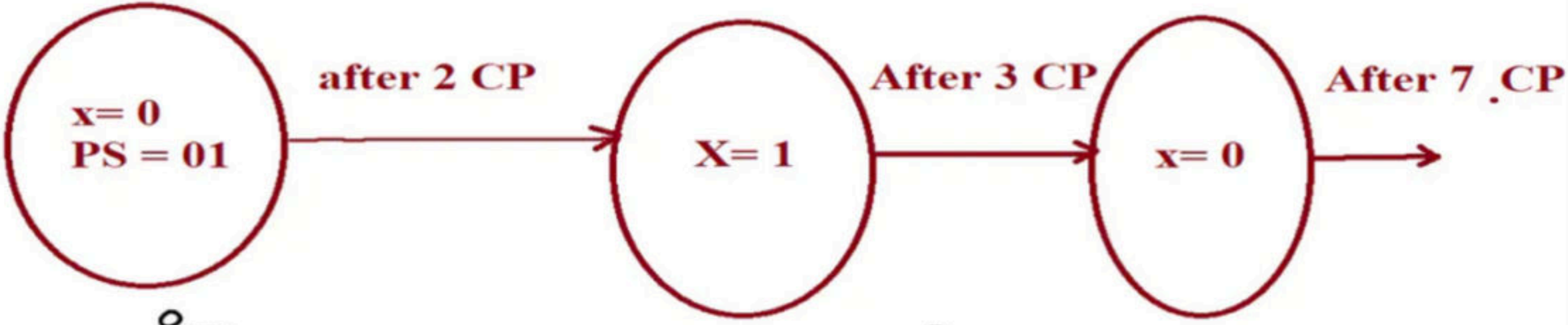
Q. Draw the state diagram (Mealy and Moore) for all the FF

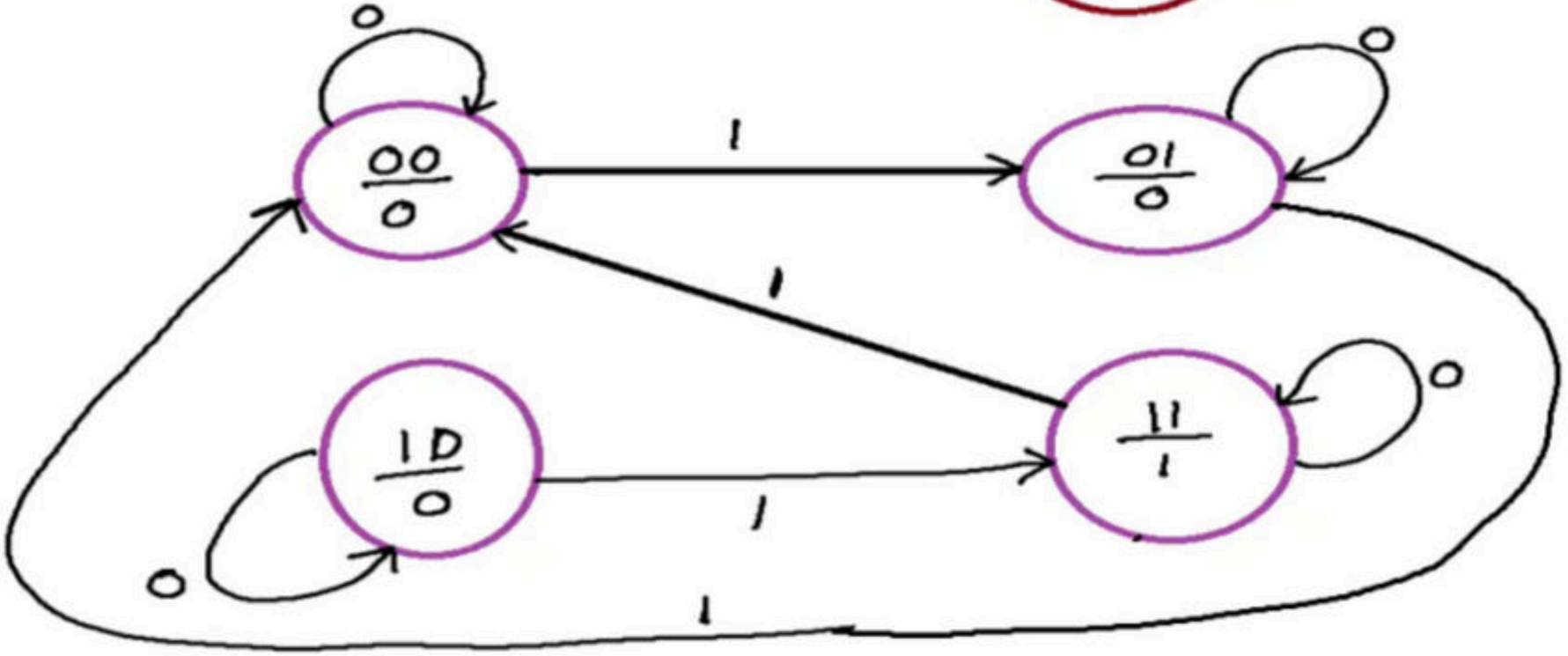
consider the state diagram find

a) if x= 0 PS =11 then NS after 10 Clock pulses

b) if x= 1 PS = 01 then NS after 4 Clock pulses

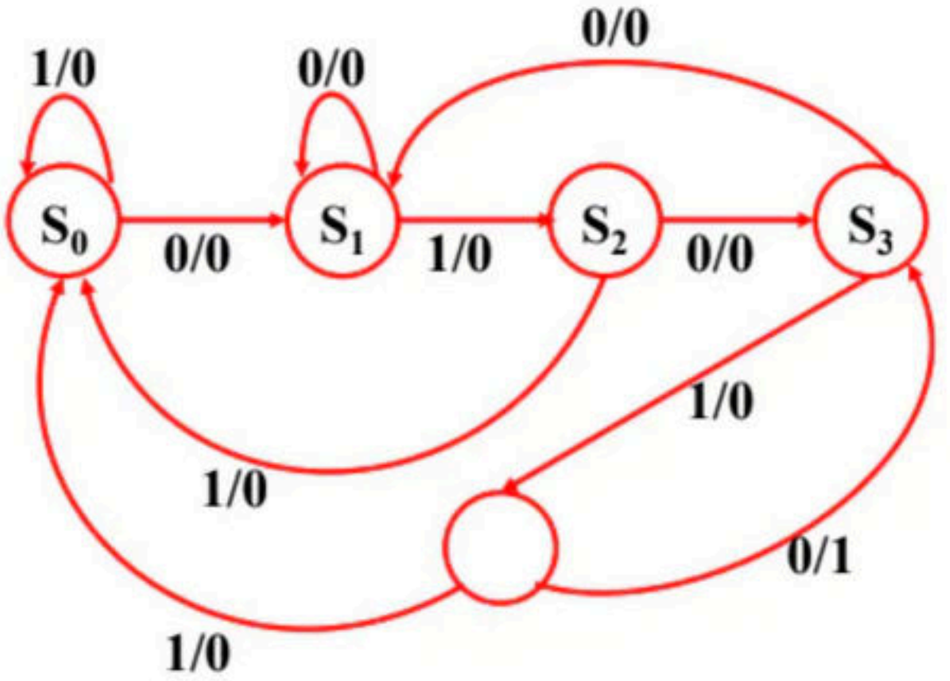
c)





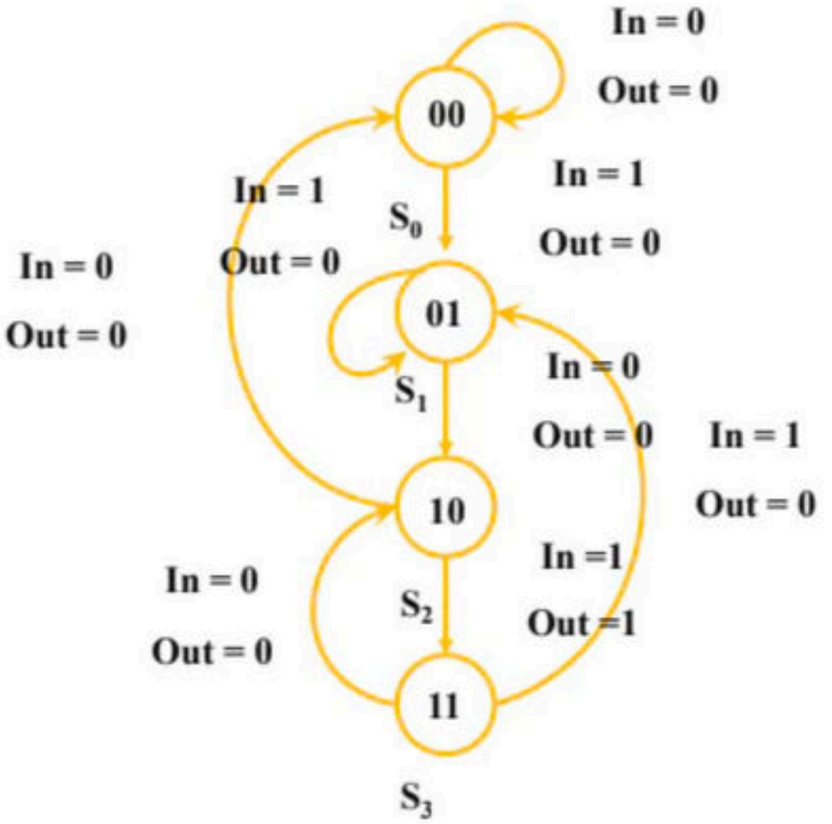
Q. The state diagram of a sequence detector is shown below. State S_0 is the initial state of the sequence detector. If the output is 1, then

(a) the sequence 01010 is detected
(b) the sequence 01011 is detected
(c) the sequence 01001 is detected
(d) the sequence 01110 is detected



Q. The state diagram of a finite state machine (FSM) designed to detect an overlapping sequence of three bits is shown in the figure. The FSM has an input 'In' and an output 'Out'. The initial state of the FSM is S_0.

If the input sequence is 101011101001101, starting with the left-most bit then the number times 'Out' will be 1 is _____.



Q. Develop the non overlapping sequence detector and state table to detect 1111 by using mealy modal

Q. Develop the overlapping sequence detector and state table to detect 1111 by using mealy modal

Q. Develop the non overlapping sequence detector and state table to detect 1111 by using Moore modal

Q. Develop the overlapping sequence detector and state table to detect 1111 by using Moore modal

Q. Develop the non overlapping sequence detector and state table to detect 1110 by using mealy modal

Q. Develop the overlapping sequence detector and state table to detect 1110 by using mealy modal

Q. Develop the non overlapping sequence detector and state table to detect 1110 by using Moore modal

Q. Develop the overlapping sequence detector and state table to detect 1110 by using Moore modal

Q. Develop the non overlapping sequence detector and state table to detect 0011 by using mealy modal

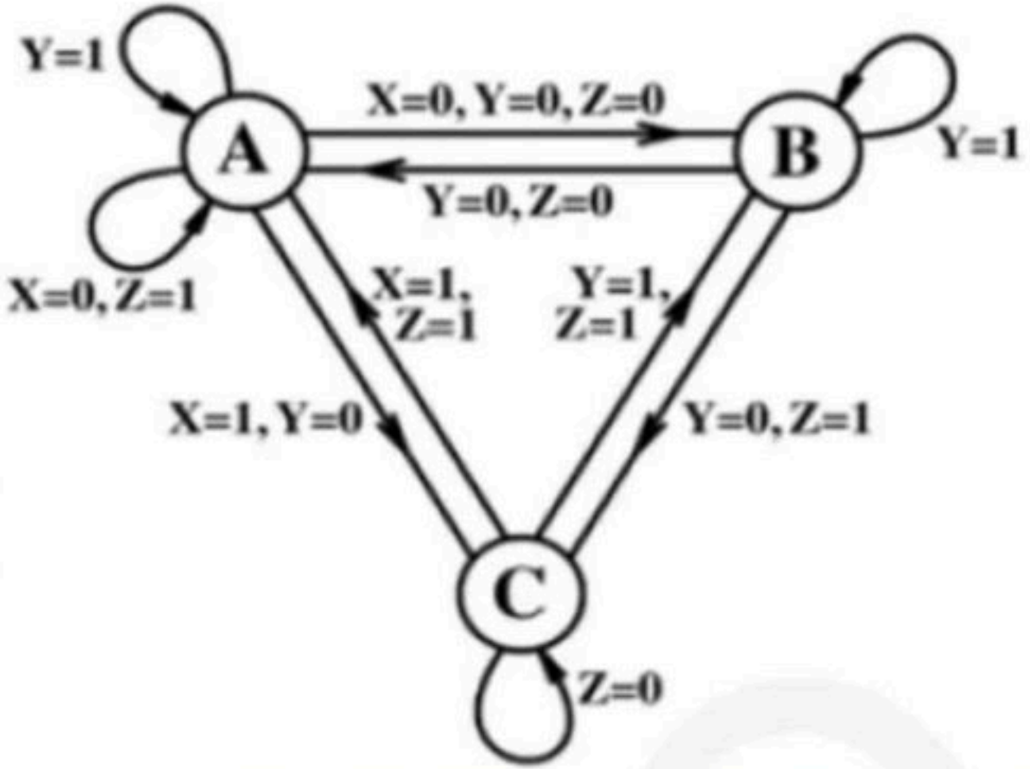
Q. Develop the overlapping sequence detector and state table to detect 0011 by using mealy modal

Q. Develop the non overlapping sequence detector and state table to detect 0011 by using Moore modal

Q. Develop the overlapping sequence detector and state table to detect 0011 by using Moore modal

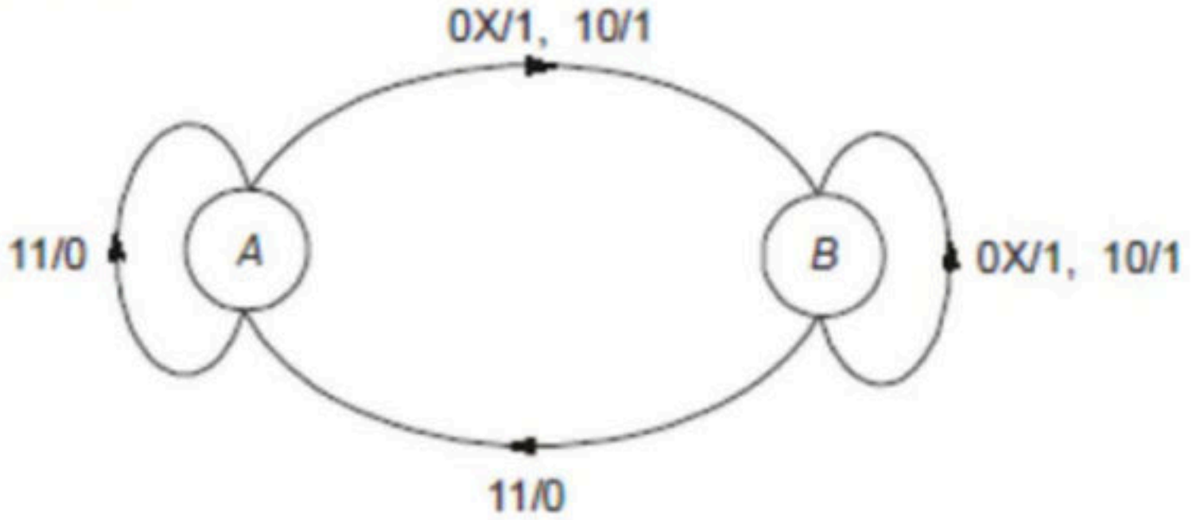
The state transition diagram for a finite state machine with states A, B and C, and binary inputs X, Y and Z, is shown in the figure.

Which one of the following statements is correct?



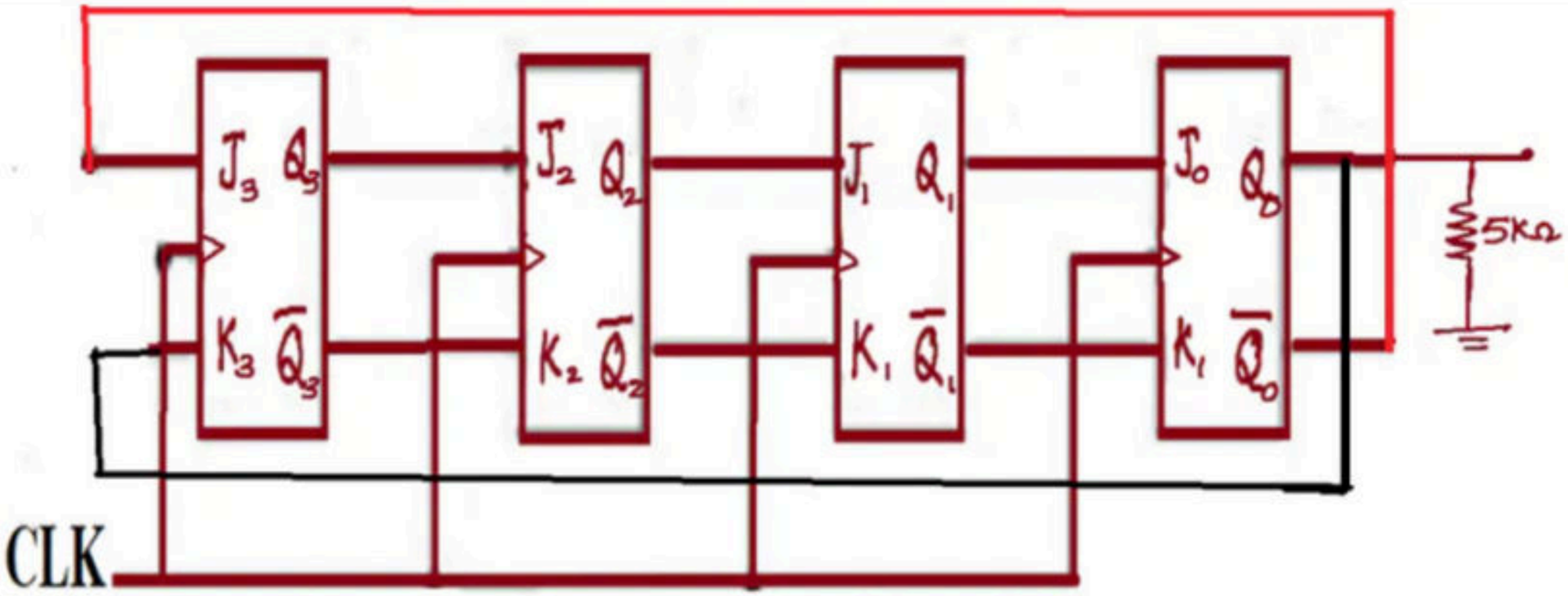
A. Transitions from State A are ambiguously
B. Transitions from State B are ambiguously
C. Transitions from State Care ambiguously
D. All of the state transitions are defined unambiguously.

The state diagram of a Mealy circuit is shown in figure. Where "*X*" represents the don't care condition..



the circuit corresponding to the given state diagram can be used as _____.

a. OR gate
b. AND gate
c. NOR gate
d. NAND gate



Q) Find

a) power dissipation 5 KΩ

b) frequency of each FF

c) If the transition probability of data bit at each is 0.7 , find the average value of Q1

Q) If the initial state of $Q_2Q_1Q_0 = 110$, then find

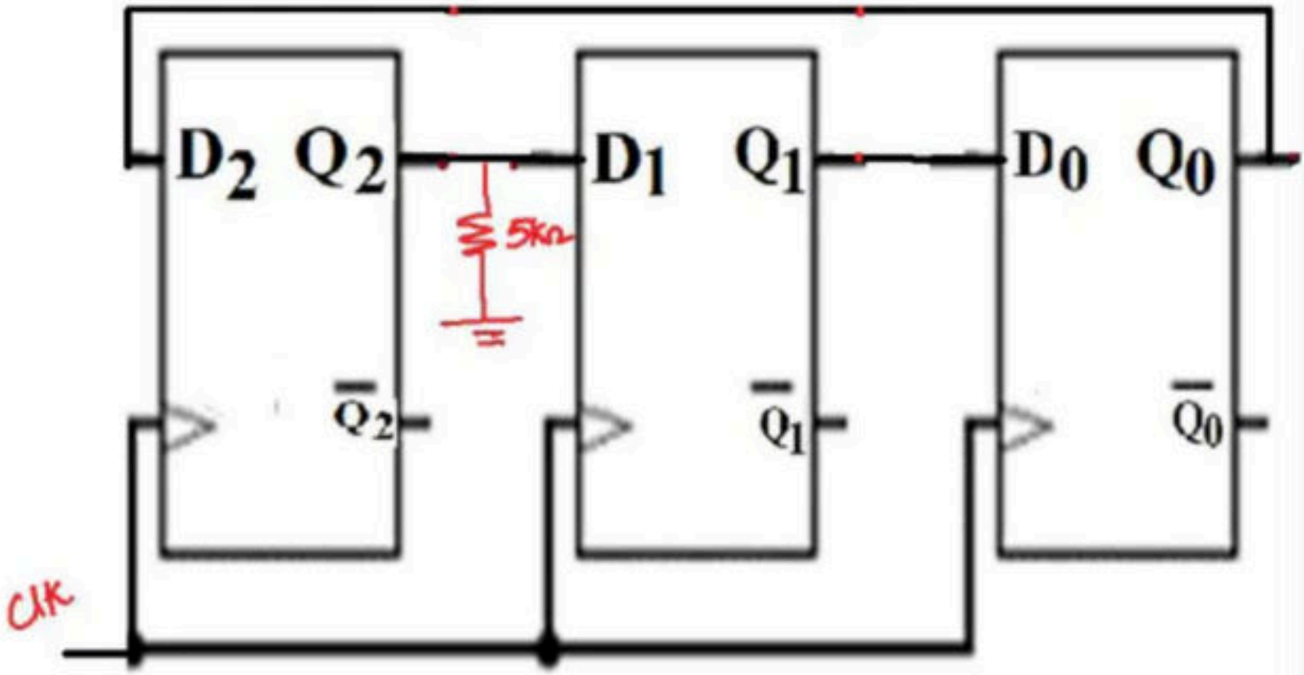
a) MoD No

b) MSV of wave form of Q_1

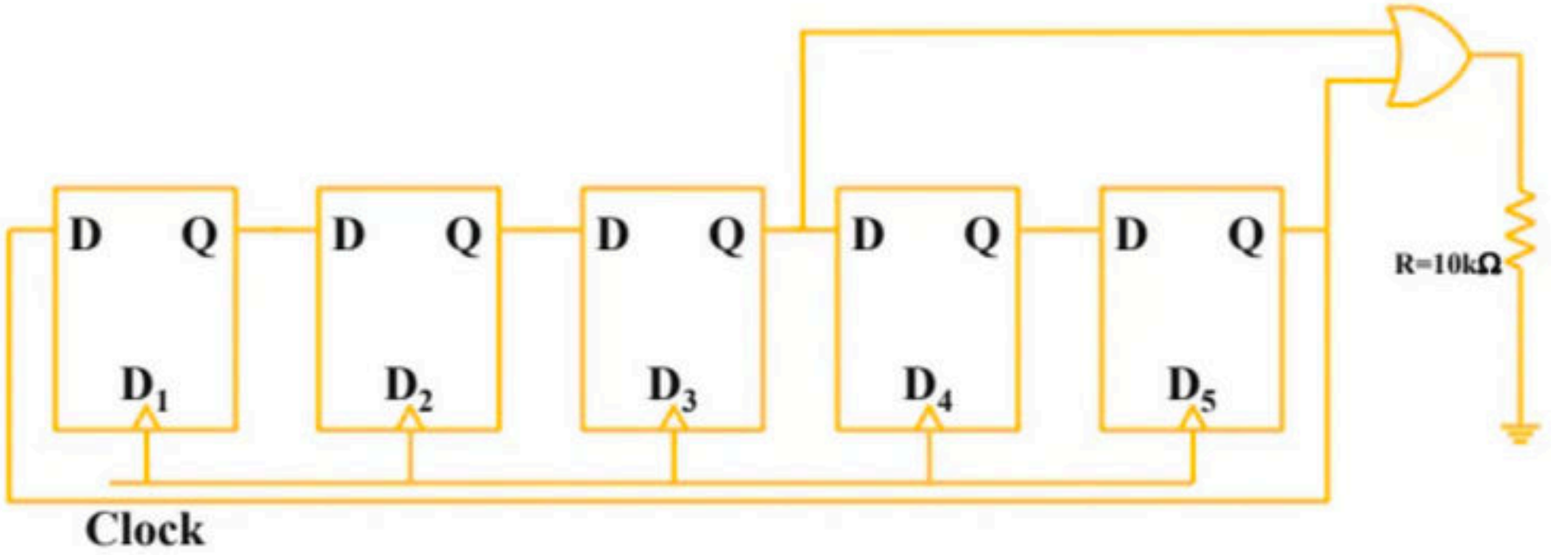
c) P_{diss} in 5kΩ

d) Average value of wave form of Q_0

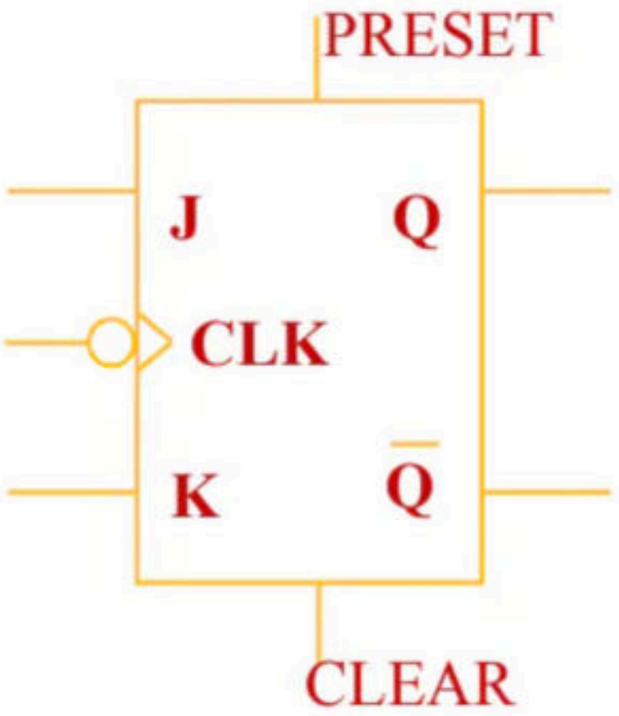
e) Frequency of $Q_2Q_1Q_0$



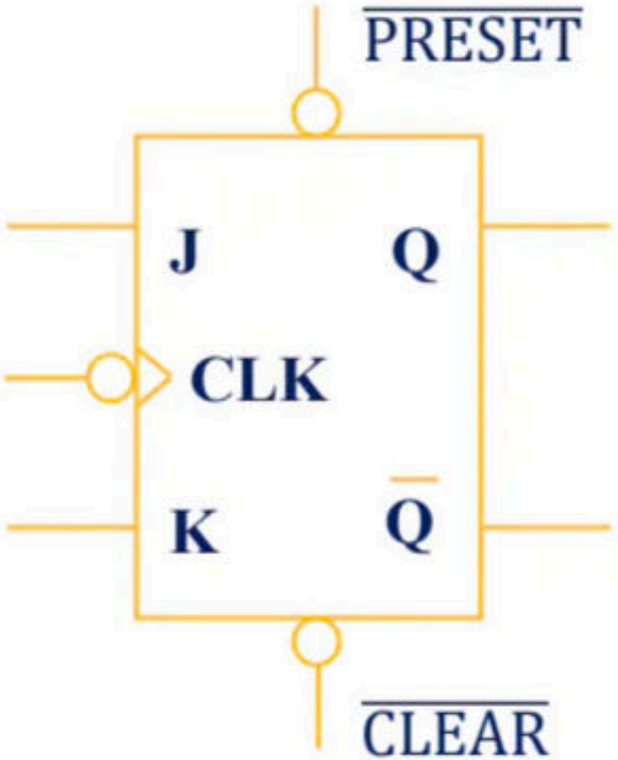
14. Assume that all the digital gates in the circuit shown in the figure are ideal, the resistor R = 10kΩ and the supply voltage is 5V.
The D flip-flips D_1, D_2, D_3, D_4 and D_5 are initialized with logic values 0, 1, 0, 1 and 0, respectively. The clock has a 30% duty cycle. The average power dissipated (in m W) in the resistor R is _____.



Asynchronous Clear
and
Asynchronous Preset

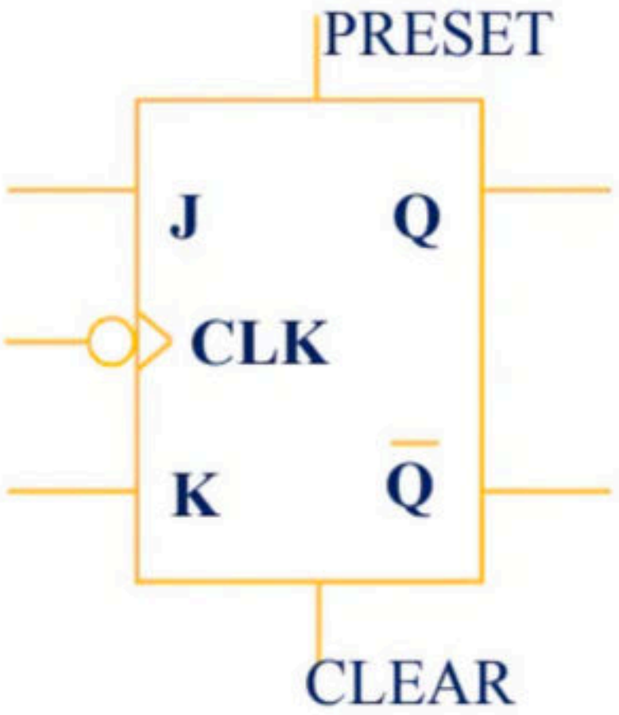


PRESET	CLEAR	FF response

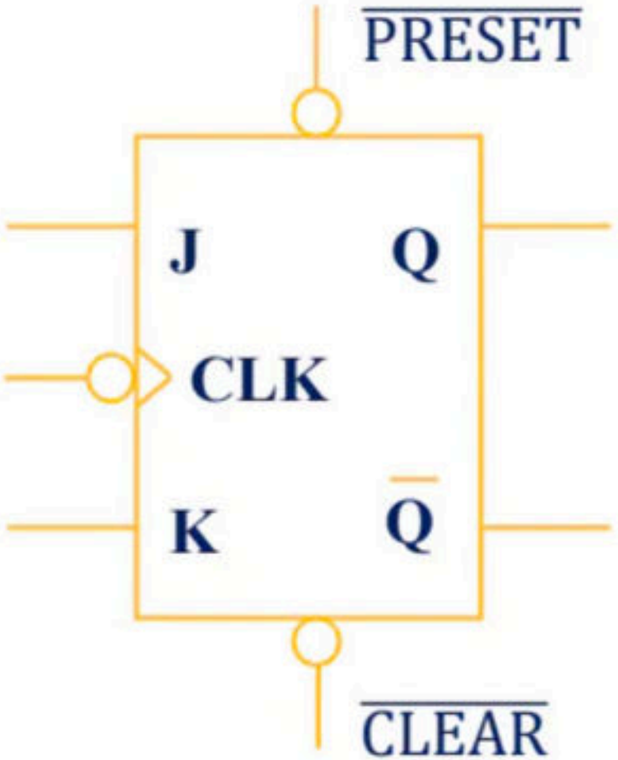


$\overline{\text{PRESET}}$	$\overline{\text{CLEAR}}$	FF response

Synchronous Clear
and
Synchronous Preset



PRESET	CLEAR	CLK	FF response

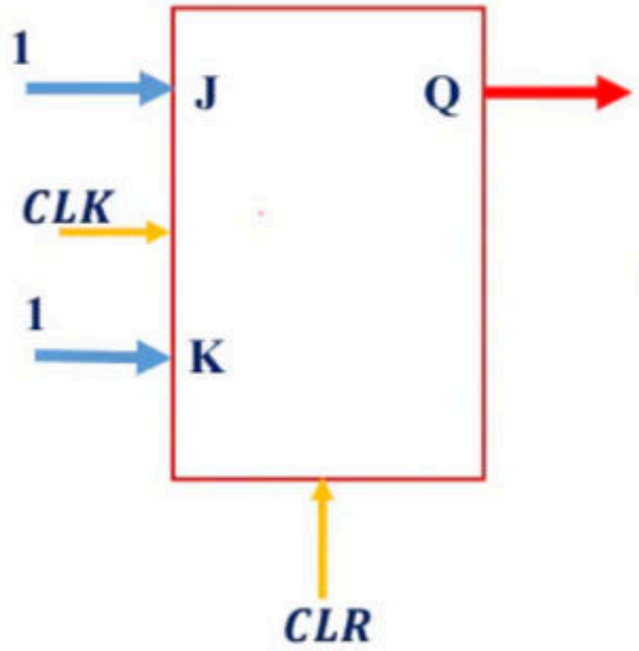


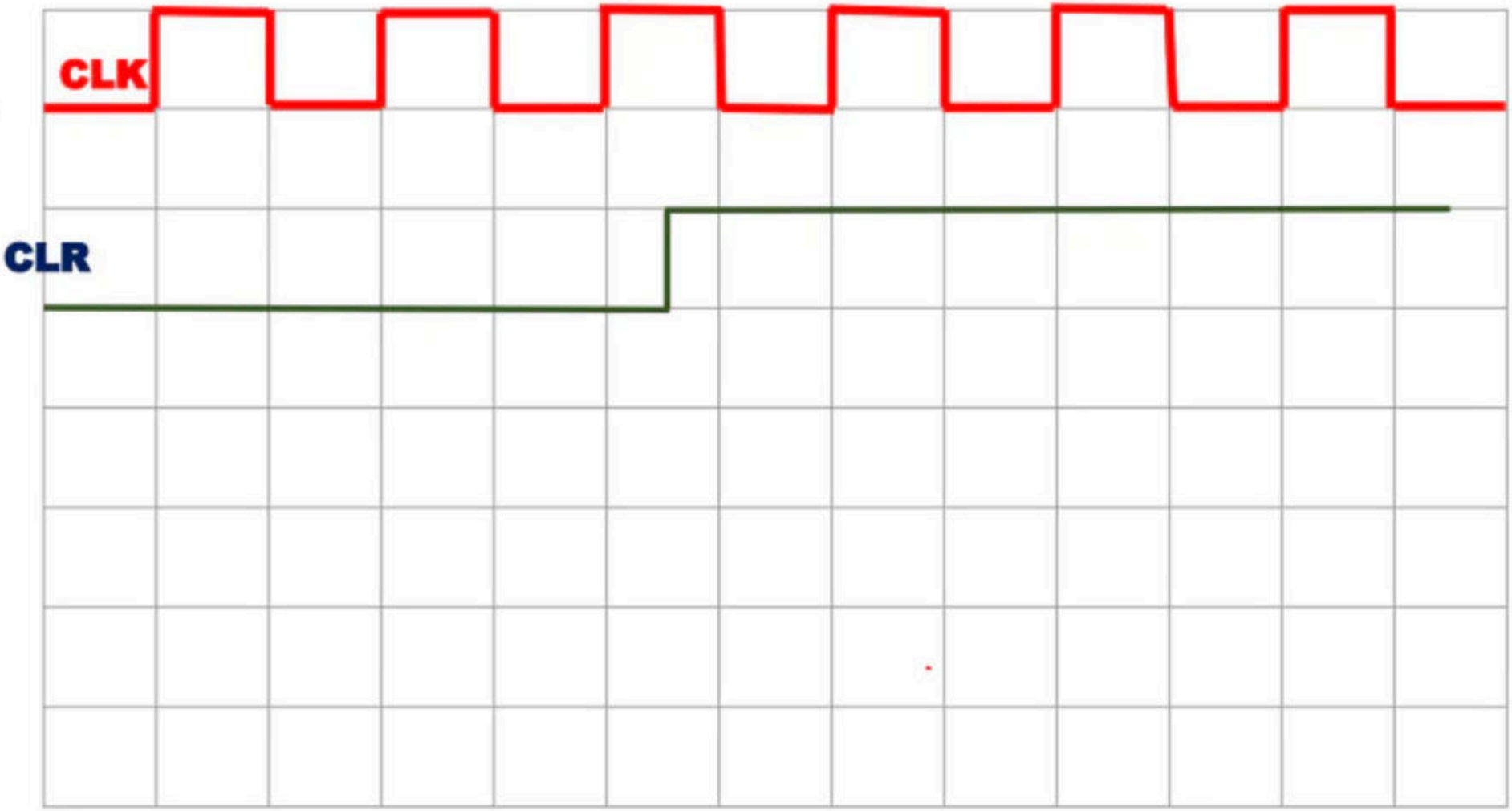
PRESET	CLEAR	CLK	FF response

Q) Draw the output wave form of the JK FF ,

a) Asynchronous CLR

b) Synchronous CLR





Q) Find the Mod number of the counter

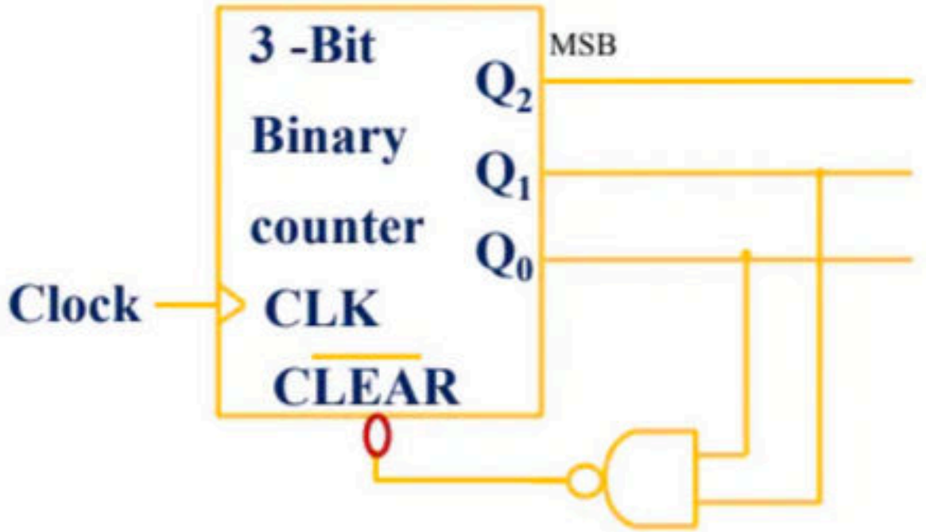
a) If (tpd) comb = 0 , Asynchronous Clear

b) If (tpd) comb = 0 , synchronous Clear

c) If (tpd) comb = Tclk , Asynchronous Clear

d) If (tpd) comb = Tclk , synchronous Clear

e) If (tpd) comb < Tclk , Asynchronous Clear



(tpd) comb = 0 , Asynchronous Clear

(tpd) comb = 0 , synchronous Clear

If (tpd) comb = Tclk , Asynchronous Clear

If (tpd) comb = Tclk , synchronous Clear

If (tpd) comb < Tclk , synchronous Clear

If (tpd) comb $<$ Tclk , Asynchronous Clear

Note :

1. If (tpd)comb = nTclk

MoD No of Synchronous counter = MoD No of Asynchronous counter

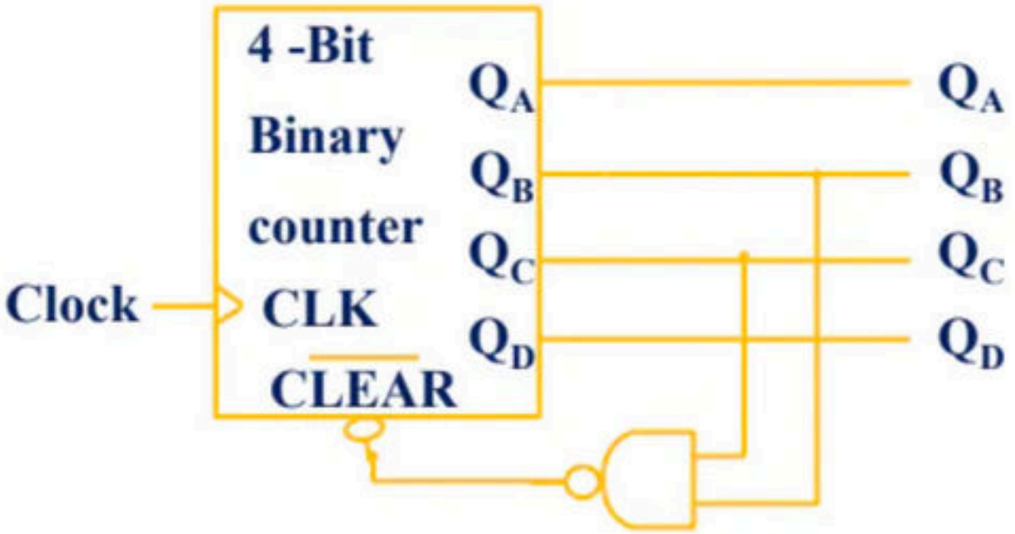
2. If (tpd) comb ≠ nTclk

MoD No of Synchronous counter = MoD No of Asynchronous counter + 1

Assumptions

1. Consider Up counter , if not mentioned
2. Q(subscript high) = MSB , if not mentioned
3. Assume Asynchronous CLR , if not mentioned
4. Assume (tpd)comb < Tclk , if not mentioned

Q. A mod-n counter using a synchronous binary up-counter with synchronous clear input is shown in the figure.
The value of n is _____.



Q. The figure shows a binary counter with synchronous clear input with the decoding logic shown, the counter works as a

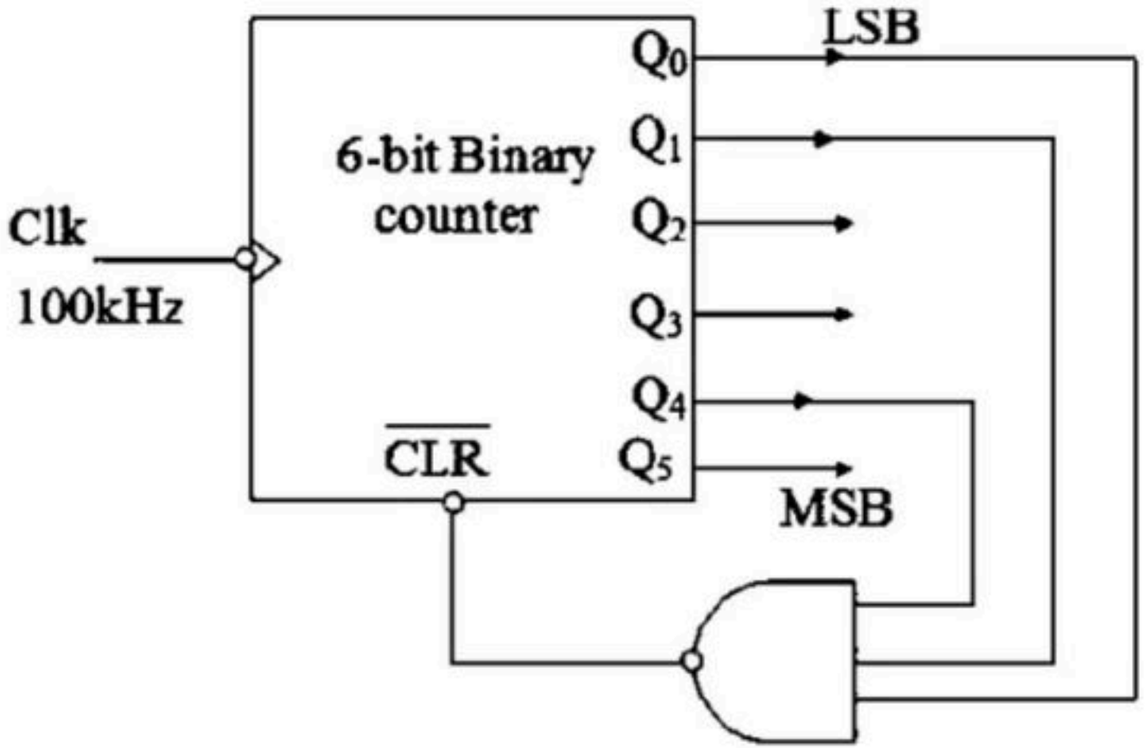
(A)mod-2 counter

(B) mod-4 counter

(C) mod-5 counter

(D) mod-6 counter

9. **A mod K counter using Asynchronous Binary up counter with synchronous clear input is shown below**



The output frequency in kHz is _____

87. The figure shows a binary counter with synchronous clear input with the decoding logic shown, the counter works as a

(A)mod-2 counter

(B) mod-4 counter

(C) mod-5 counter

(D) mod-6 counter

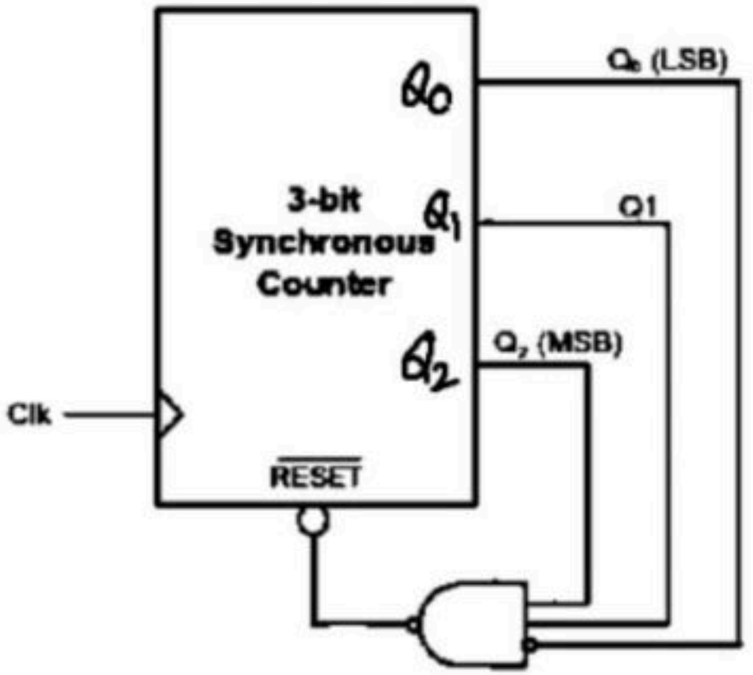
105. For the circuit shown in the figure, the delay of the bubbled NAND gate is 2 ns and that of the counter is assumed to be zero. If the clock (Clk) frequency is 1 GHz, then the counter behaves as a **GATE (ECE-2016)**

(a) mod-5 counter

(b) mod-6 counter

(c) mod-7 counter

(d) mod-8 counter



108. In the circuit shown below, a positive edge-triggered D Flip-Flop is used for sampling input data D_{in} using clock CK. The XOR gate output 3.3 volts for logic HIGH and 0 volts for logic LOW levels. The data bit and clock periods are equal and the value of $\Delta T/T_{CK} = 0.15$, where the parameters ΔT and T_{CK} are shown in the figure. Assume that the Flip-Flop and the XOR gate are ideal. If the probability of input data bit (D_{in}) transition in each clock period is 0.3, the average value (in volts, accurate to two decimal places) of the voltage at node X, is_____.