



Sequential Ckts - VII

Comprehensive Course on Digital Logic Design 2023/2024

Combinational Logic Circuit

Combinational Logic Circuit

- The present output depends on present input only
- In combinational circuits feedback and clock is not present

- HA
- HS
- FA
- FS
- Parallel Adder
- Carry look ahead Adder
- Binary Multiplier
- Magnitude Comparators
- Multiplexer
- Demultiplexers
- Decoder
- Encoder
- Priority Encoder
- Code converters

Half Adder

For the addition of two single bits



A	B	Sum	Carry

Logic Circuit

Half Adder using NAND Gates

Half Adder using NOR Gates

Half Subtractor (A-B)

For the subtraction of two single bits



A	B	Difference	Borrow

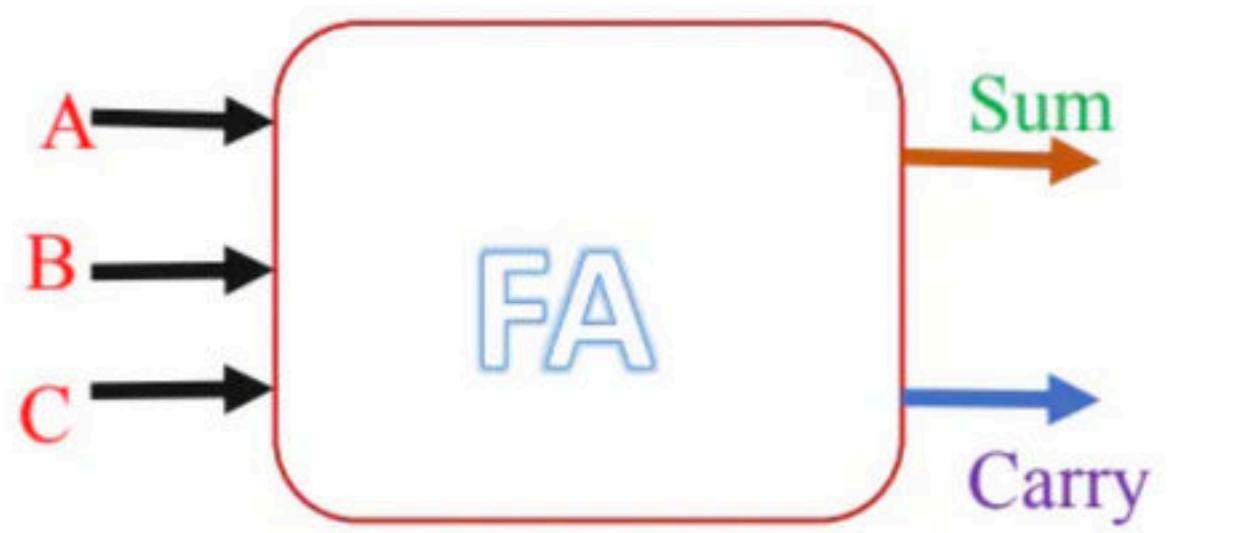
Logic Circuit

Half Adder / Half Subtractor

Half Subtractor using NAND Gates

Half Subtractor using NOR Gates

Full Adder



Full Adder with two Half Adders

Full Adder with two Half Adders

Full Adder using NAND Gates

Full Adder using NOR Gates

Full Subtractor(A - B-C)



Logic Circuit

Full Subtractor with two Half Subtractors

Full Subtractor with two Half Subtractors

Full Subtractor using NAND Gates

Full Subtractor using NOR Gates

FS : A- B- C

FS : B- C- A

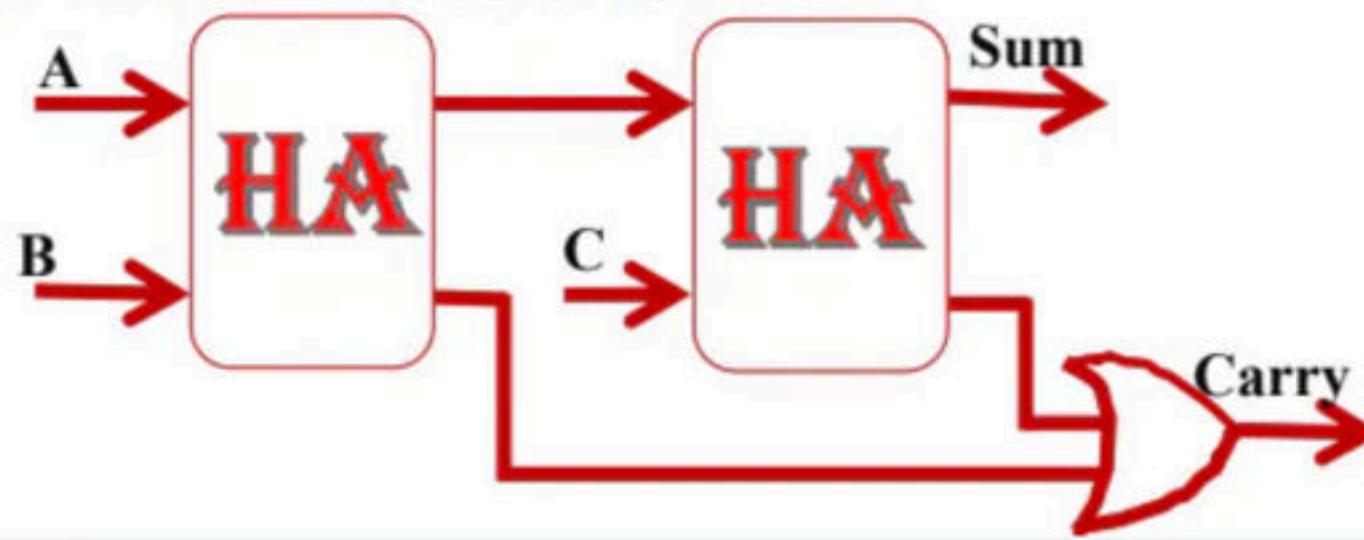
FS : C- A- B

HA

1. Logical expression for Sum =
2. Logical expression for Carry =
3. Minimum number of NAND Gates =
4. Minimum number of NOR Gates =

FA

1. Logical expression for Sum =
2. Logical expression for Carry =
3. Minimum number of NAND Gates =
4. Minimum number of NOR Gates =

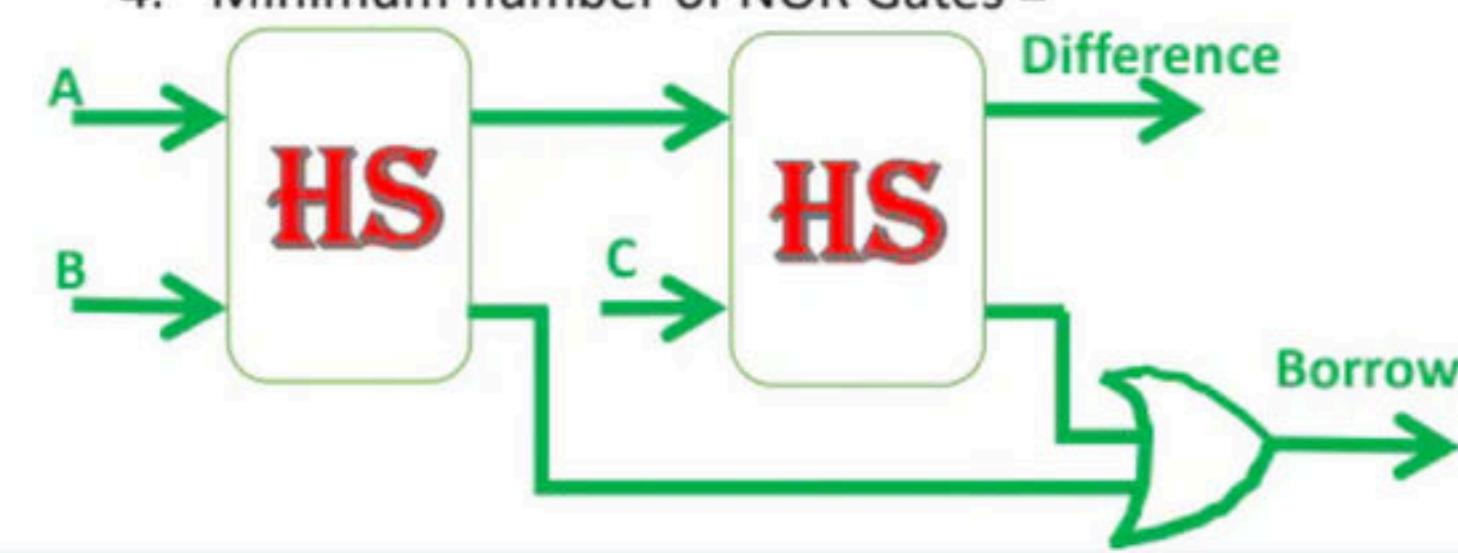


HS

1. Logical expression for Difference =
2. Logical expression for Borrow=
3. Minimum number of NAND Gates =
4. Minimum number of NOR Gates =

FS

1. Logical expression for Difference=
2. Logical expression for Borrow =
3. Minimum number of NAND Gates =
4. Minimum number of NOR Gates =



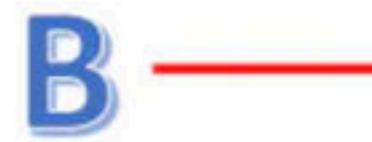
Ripple Carry Adder (Parallel Adder)

A  a_3

a_2

a_1

a_0

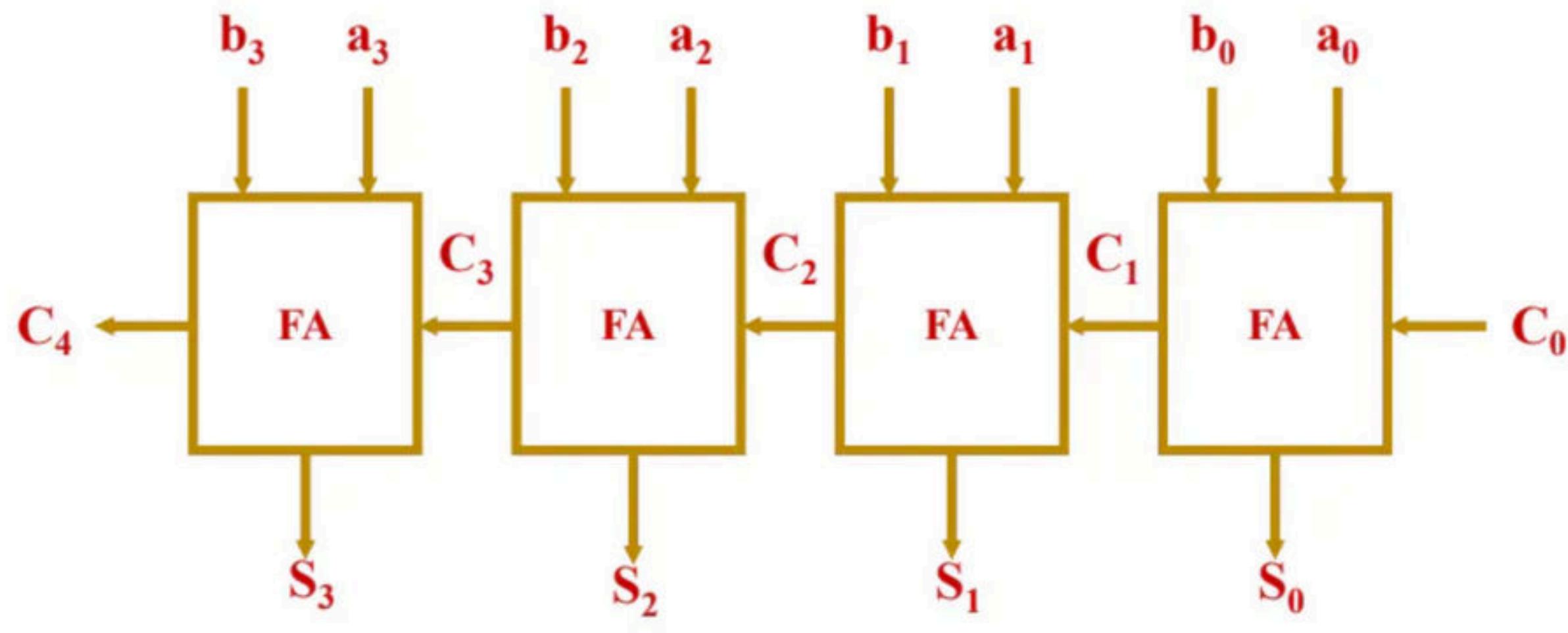
B  b_3

b_2

b_1

b_0





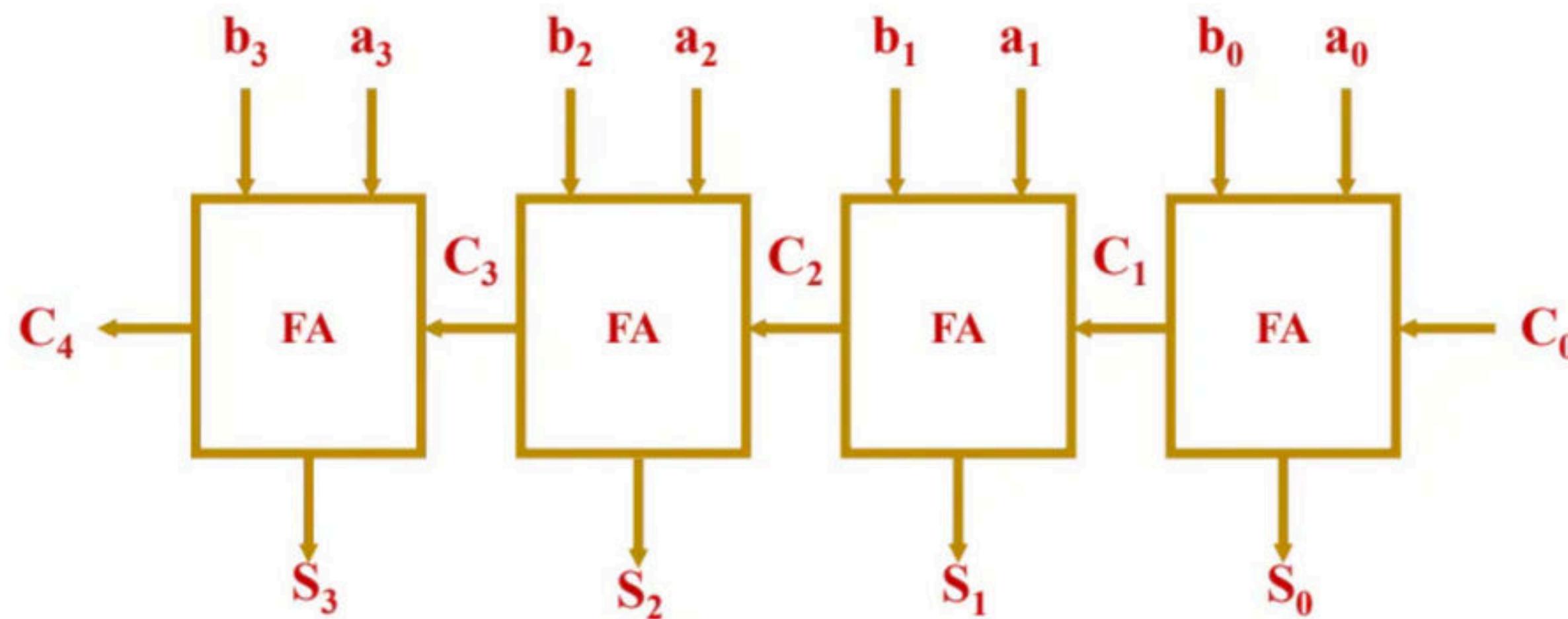
Note :

To implement 4-bit parallel adder

To implement n-bit parallel adder

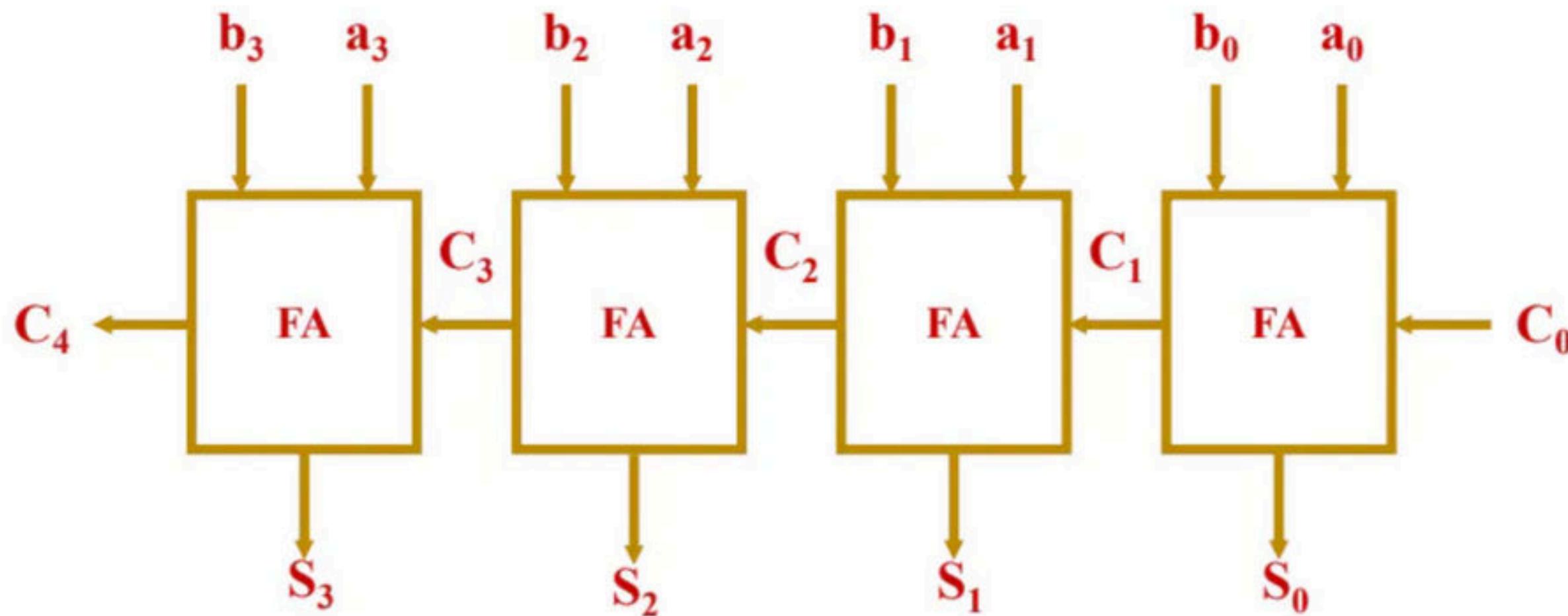
Delay Analysis

Case : 1 $(tpd)_{sum} > (tpd)_{carry}$



Delay Analysis

Case : 2 $(tpd)_{sum} < (tpd)_{carry}$



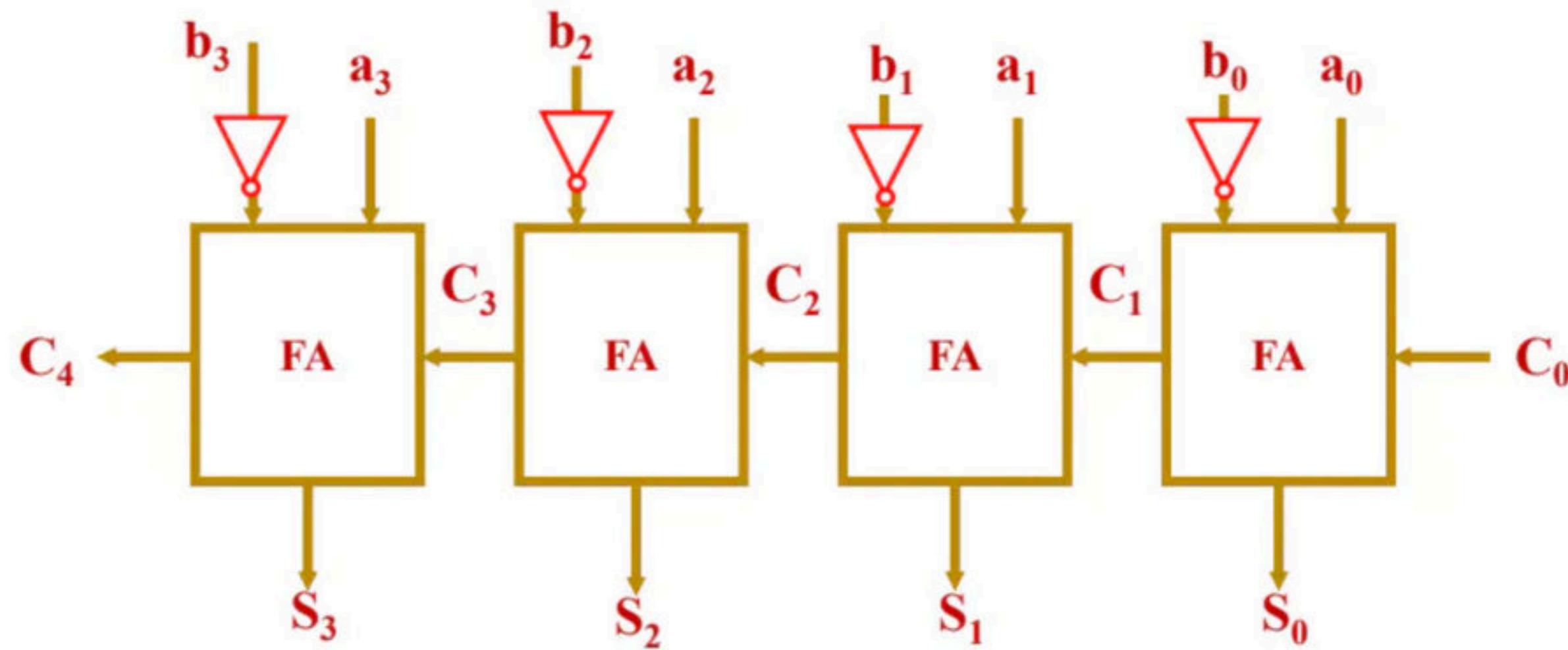
In general for n- bit Parallel Adder

Delay =

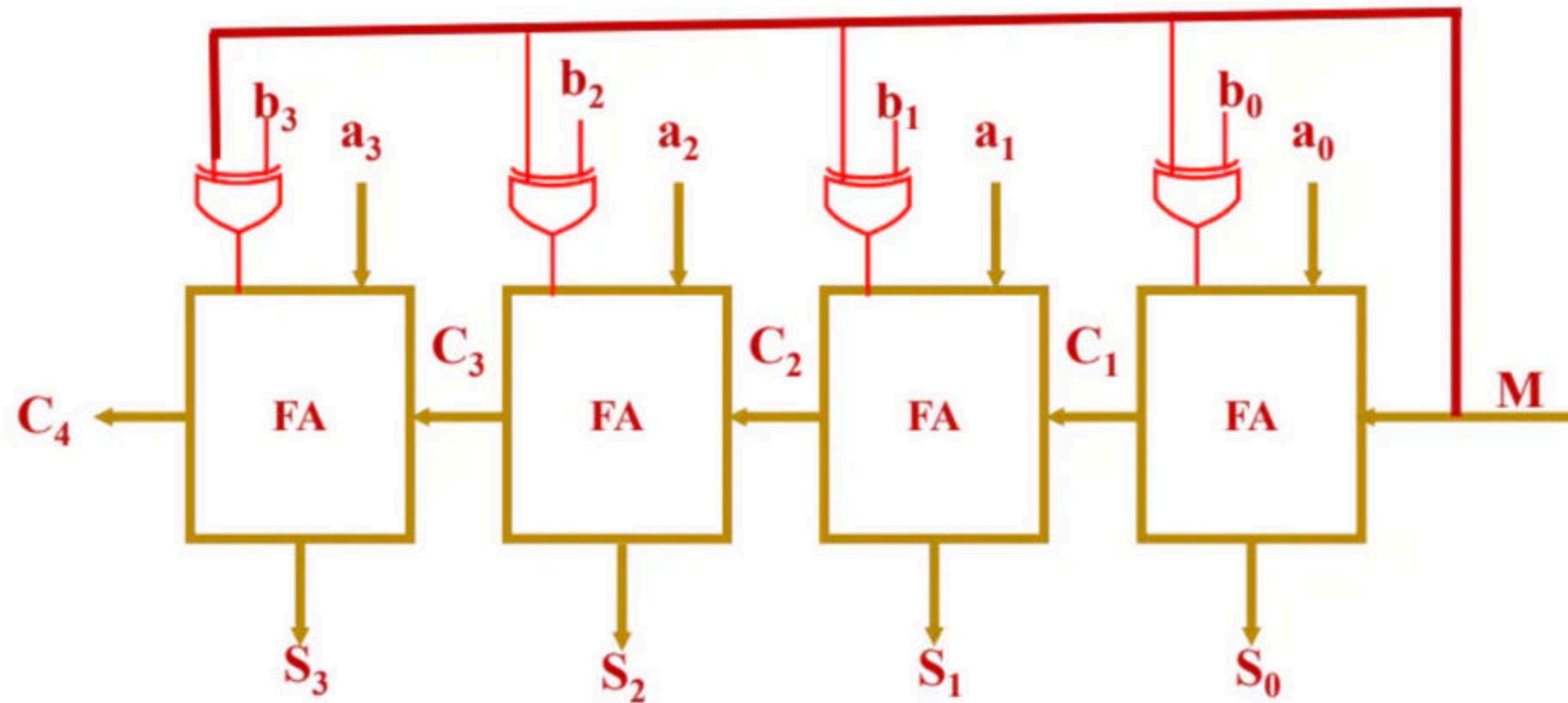
Q) A 16-bit RCA is realized using 16 identical FAs , if the (tpd)carry = 12ns ,
(tpd)sum= 15ns , then the overall delay is ----- ns

Q) A 4-bit RCA is implemented using 4-FAs , if the propagation delay of XOR – Gate is twice the delay of AND/OR Gate , then the overall delay of 4- bit RCA if the delay of AND/OR Gate is $1.2 \mu\text{sec}$

Parallel Subtractor

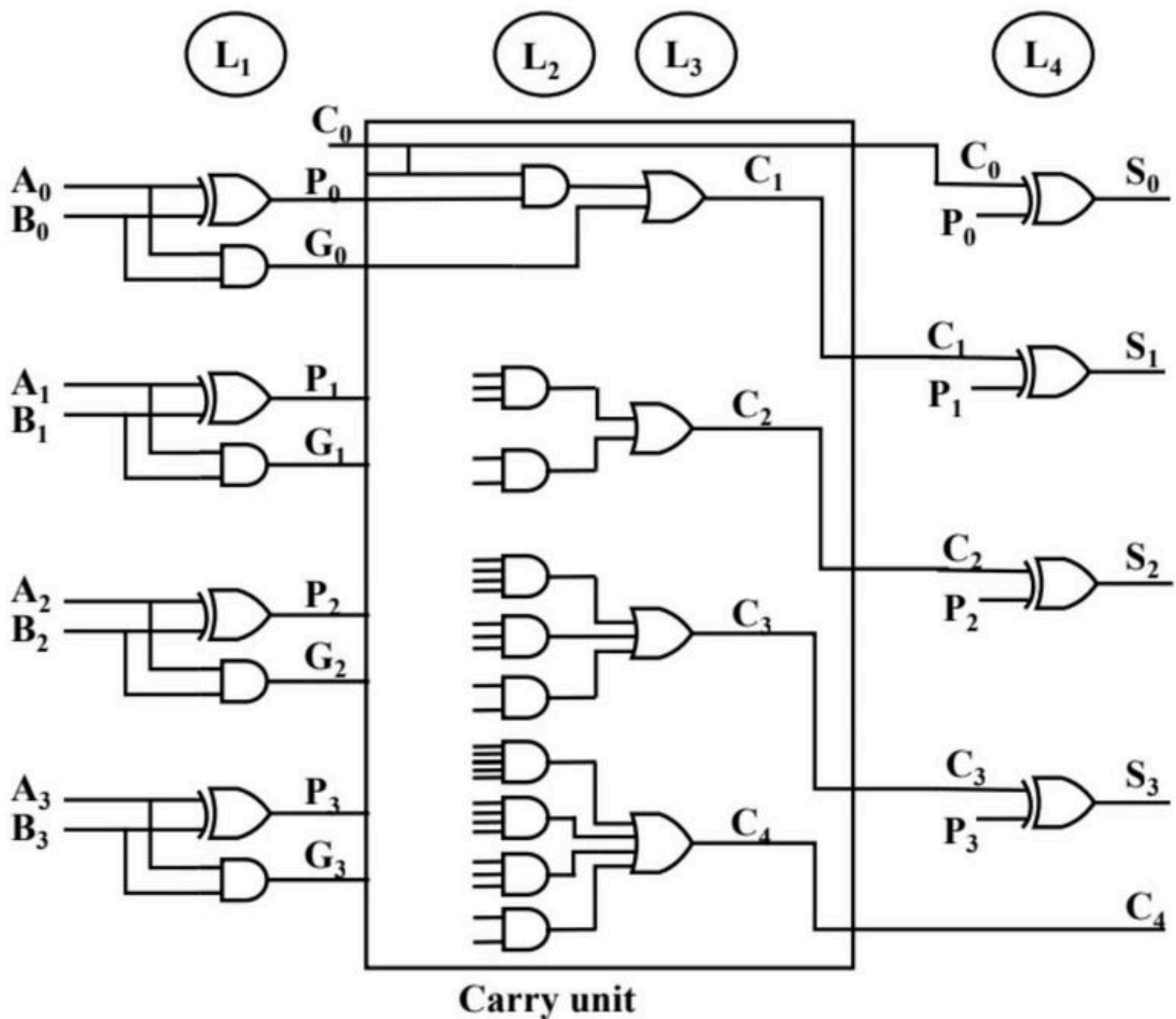


Parallel Adder/ Subtractor



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



Hardware Requirements

L 1 :

L 2 :

L 3 :

L 4 :

Total number of gates for carry =

Total number of gates for sum =

Delay Analysis

Carry =

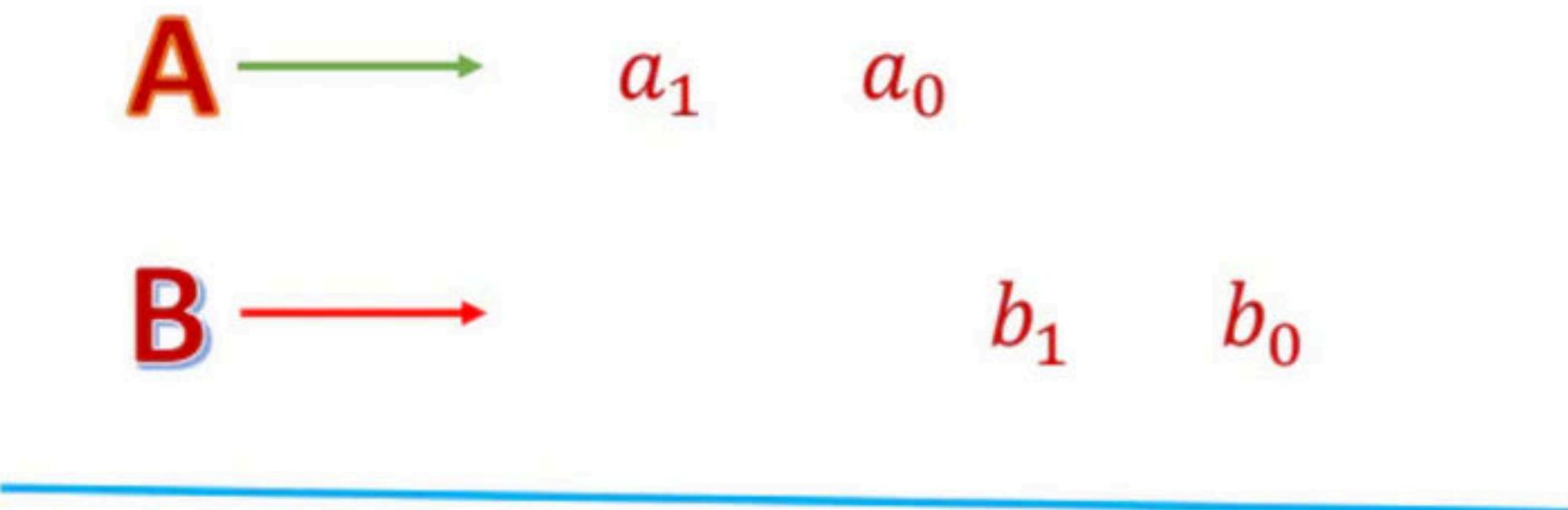
Sum =

Q) The minimum number of gates required for the implementation 4- bit look ahead carry adder are -----

Q) During the implementation of carry look ahead adder , if carry generator (G_i) and carry propagator (P_i) are available, then the minimum number of gates required are

Q) In 4- bit look ahead carry adder is implemented with the following gates NOT , AND, OR , NAND , NOR calculate the minimum time required to generate sum if each gate has 1 unit

Binary Multiplier



A 

a_2 a_1 a_0

B 

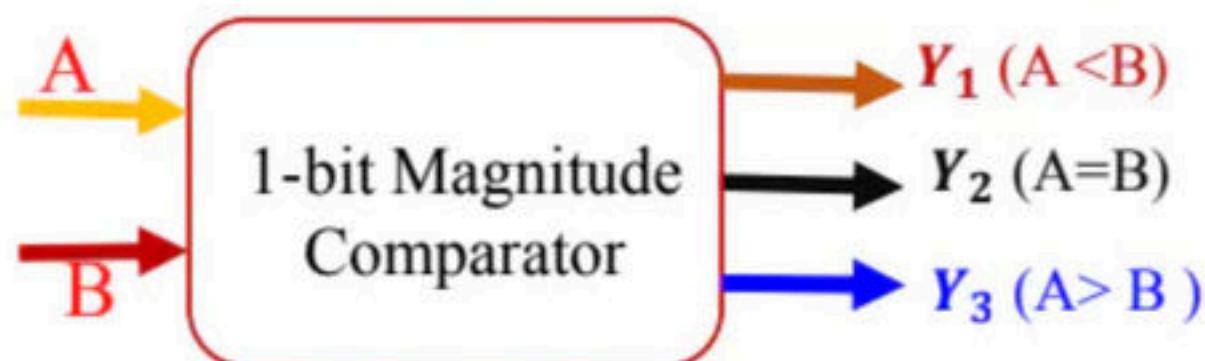
b_1 b_0



Magnitude Comparator

To compare the magnitude of two binary numbers .

1-bit magnitude comparator



A	B	Y_1 (A < B)	Y_2 (A = B)	Y_3 (A > B)

Y_1 (A < B) =

Y_2 (A = B) =

Y_3 (A > B) =

Logic circuit

1-bit Magnitude Comparator

Total number of input combinations =

Lesser than combinations =

Greater than combinations =

Equal combinations =

2-bit Magnitude Comparator



For 2-bit Magnitude Comparator

Total number of input combinations =

Lesser than combinations =

Greater than combinations =

Equal combinations =

For 2-bit Magnitude Comparator

$Y_1 (A < B) =$

$Y_2 (A = B) =$

$Y_3 (A > B) =$

For 3- bit Magnitude Comparator

$Y_1 (A < B) =$

$Y_2 (A = B) =$

$Y_3 (A > B) =$

For 4-bit Magnitude Comparator

$Y_1 (A < B) =$

$Y_2 (A = B) =$

$Y_3 (A > B) =$

For n-bit Magnitude Comparator

Total number of input combinations =

Lesser than combinations =

Greater than combinations =

Equal combinations =

Q) Find the number of 1- bit comparators , AND gates and OR gates required to implement 2- bit Comparator

Q) Find the number of 1- bit comparators , AND gates and OR gates required to implement 4- bit Comparator

Multiplexer (MUX)

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

The general structure of a Mux

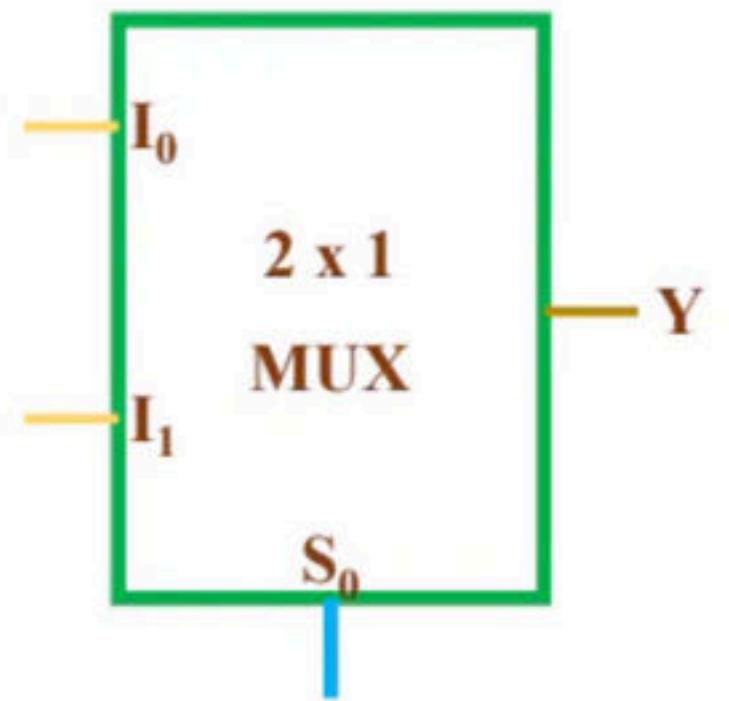
$2^n \times 1$

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

n -----> number of select inputs

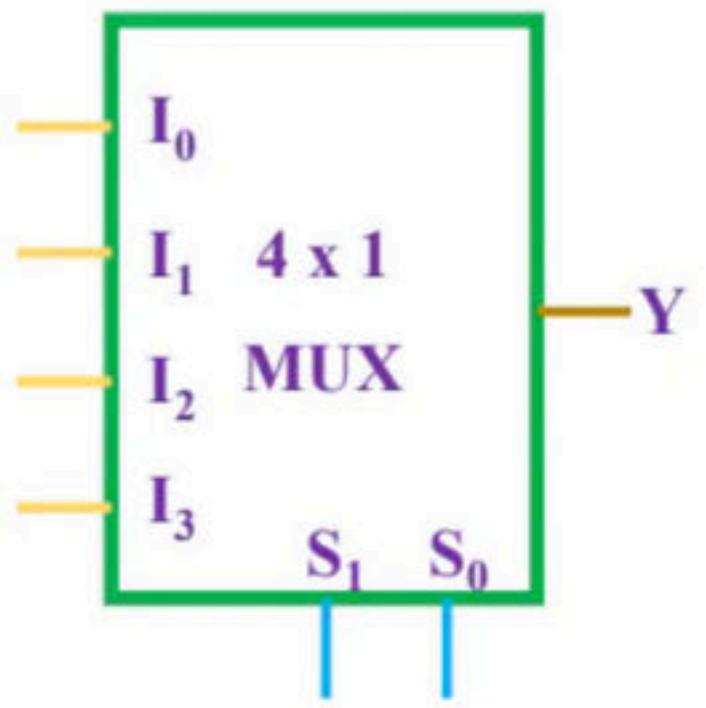
1 -----> number of outputs

2×1 MUX



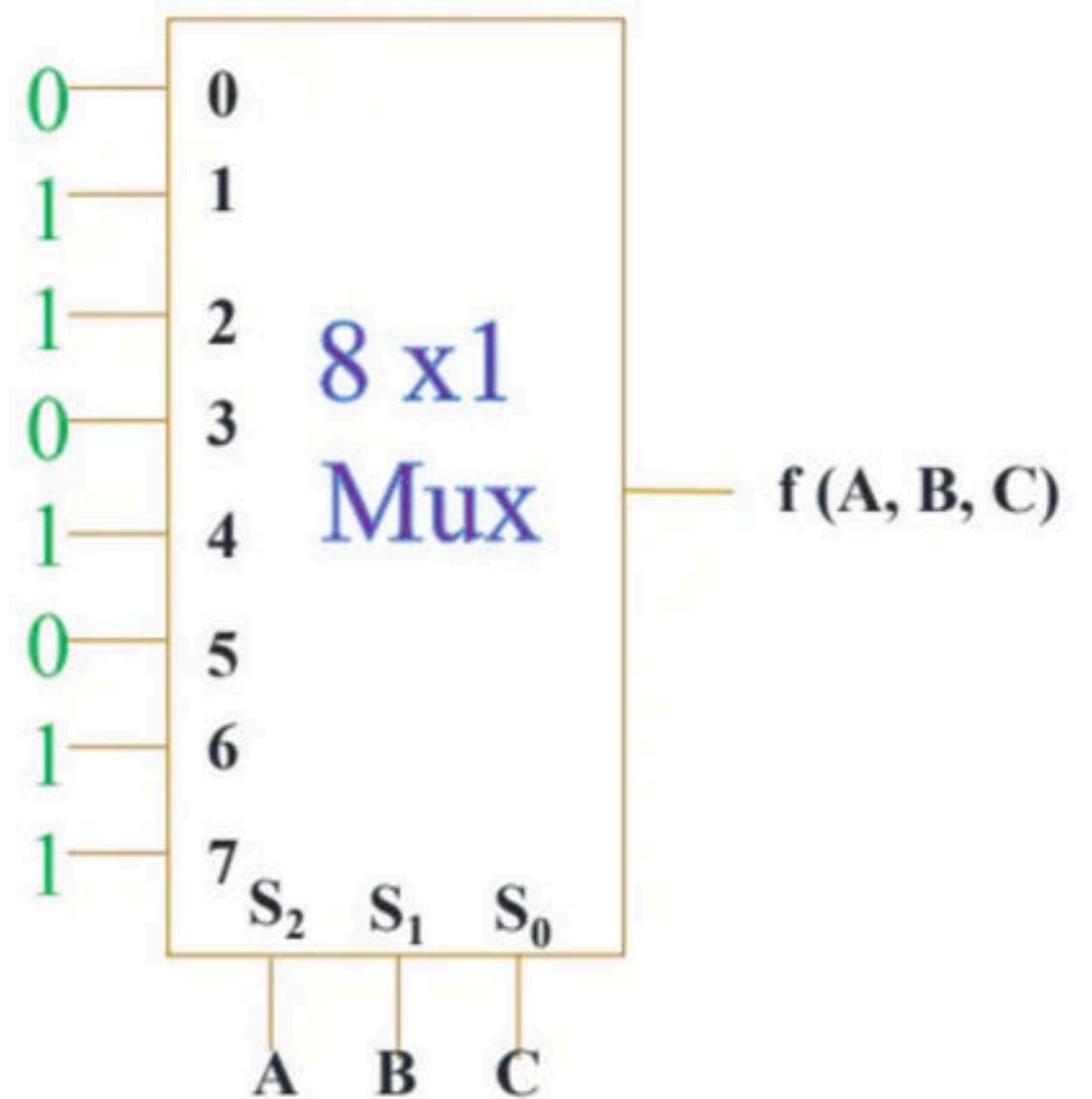
Logic circuit

4×1 MUX



S_1	S_0	Y

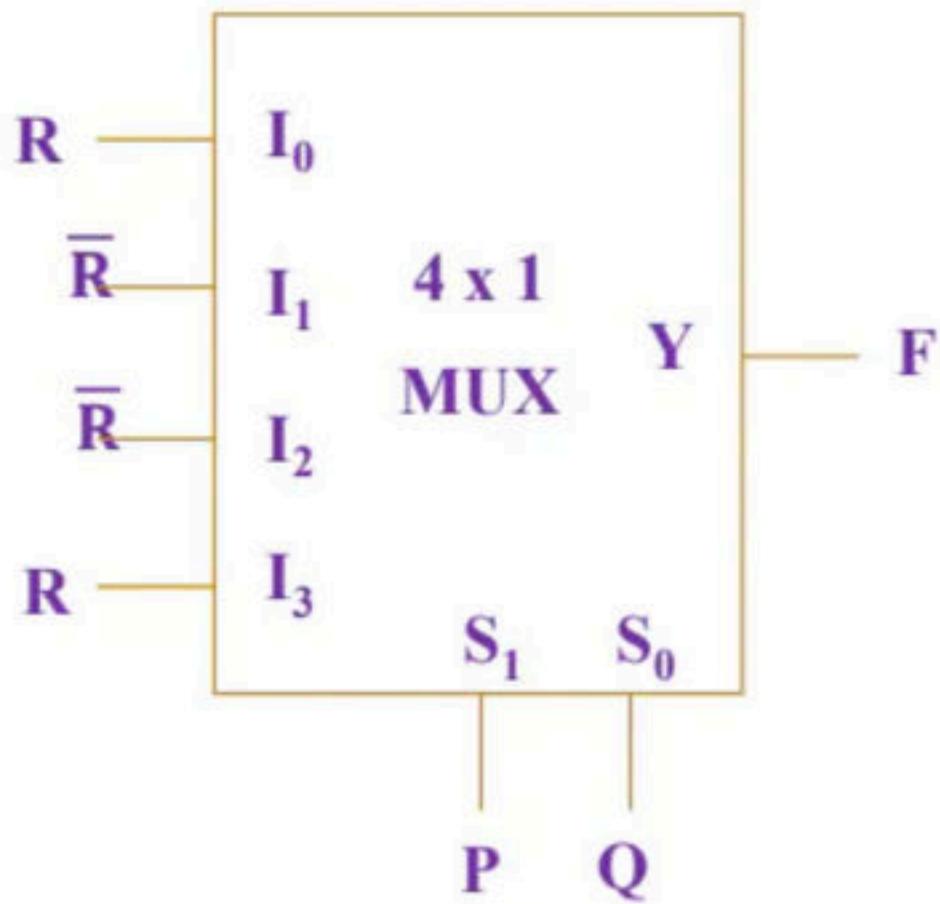
Q) Find the minterms



Q. The output F of the multiplexer circuit shown below expressed in terms of the inputs P, Q and R is

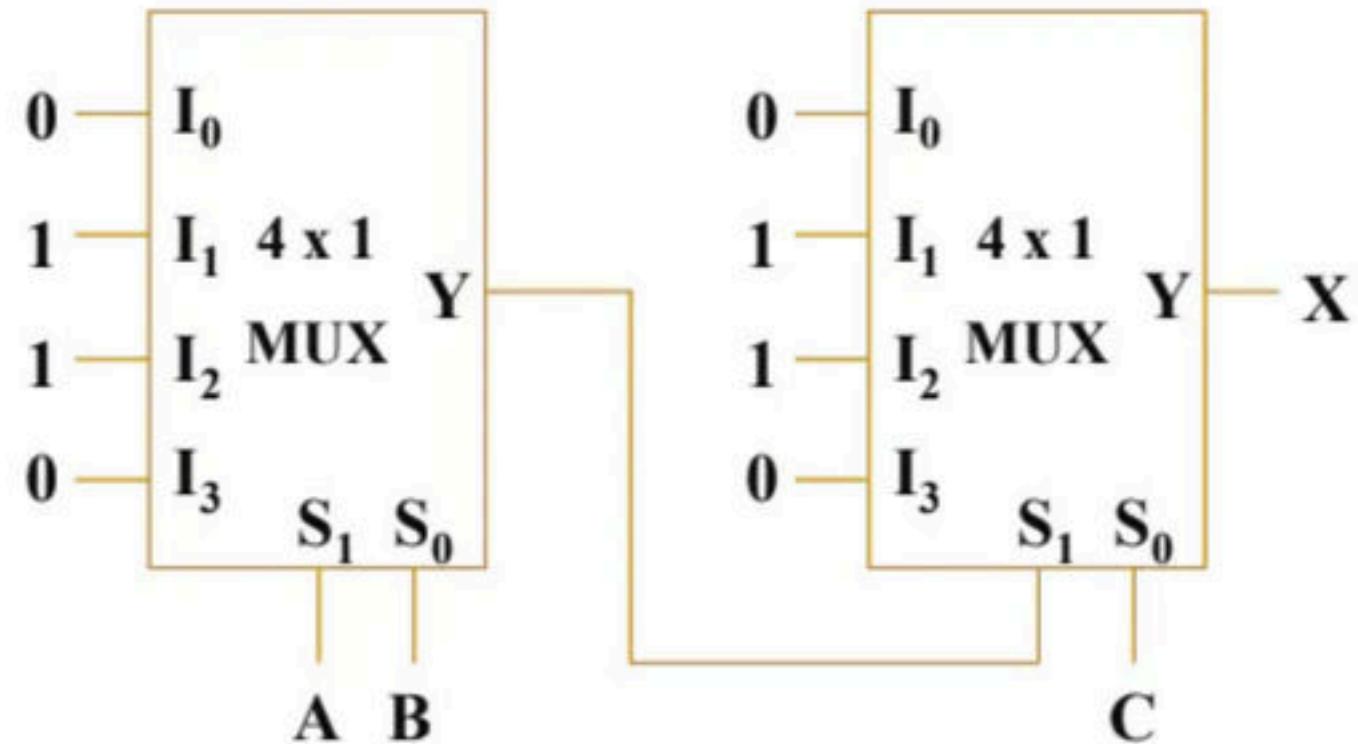
- (a) $F = P \oplus Q \oplus R$
- (c) $F = (P \oplus Q)R$

- (b) $F = PQ + QR + RP$
- (d) $F = (P \oplus Q)\bar{R}$



Q. in the following circuit, X is given by

- (a) $X = A\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C + ABC$
- (b) $X = \bar{A}BC + A\bar{B}C + AB\bar{C} + \bar{A}\bar{B}\bar{C}$
- (c) $X = AB + BC + AC$
- (d) $X = \bar{A}\bar{B} + \bar{B}\bar{C} + \bar{A}\bar{C}$



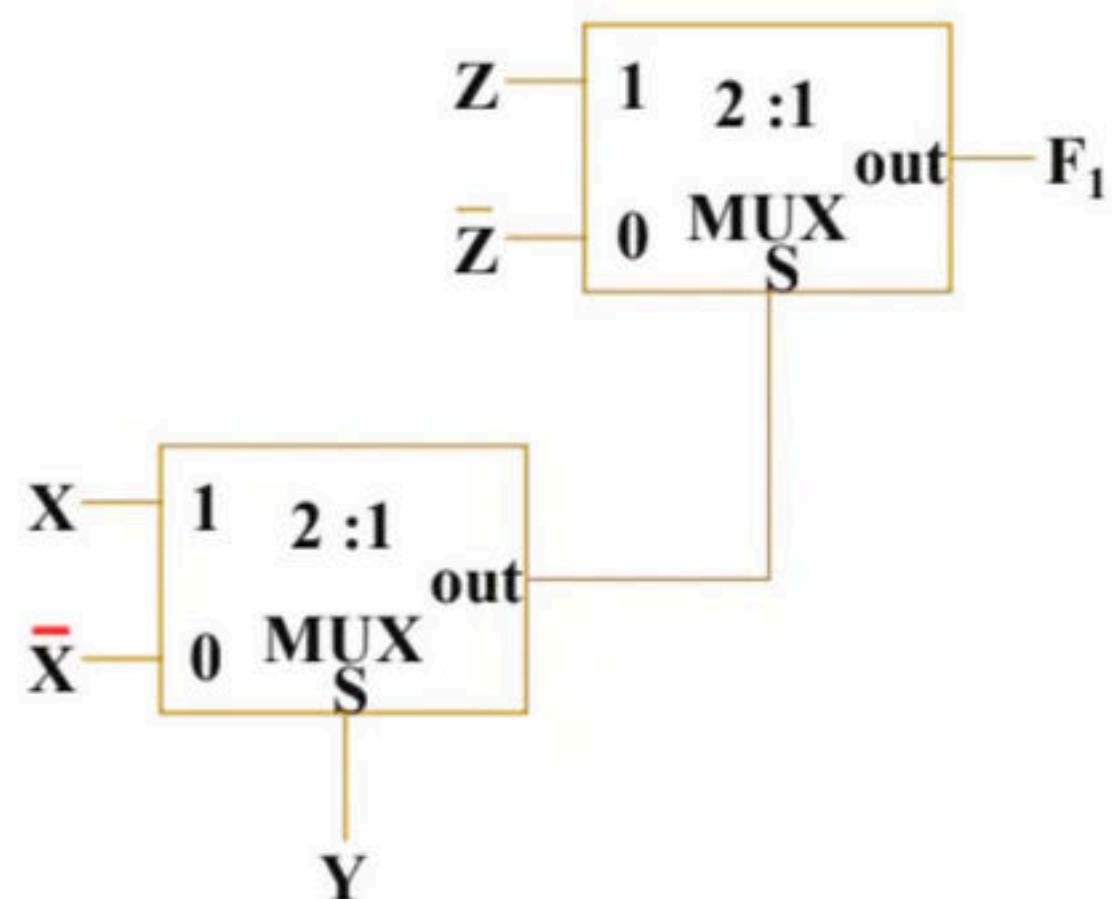
Q. A MUX circuit shown in the figure below implements a logic function F_1 . The correct expression for F_1 is.

(a) $(\bar{X} \oplus Y) \oplus Z$

(c) $(X \oplus Y) \oplus \bar{Z}$

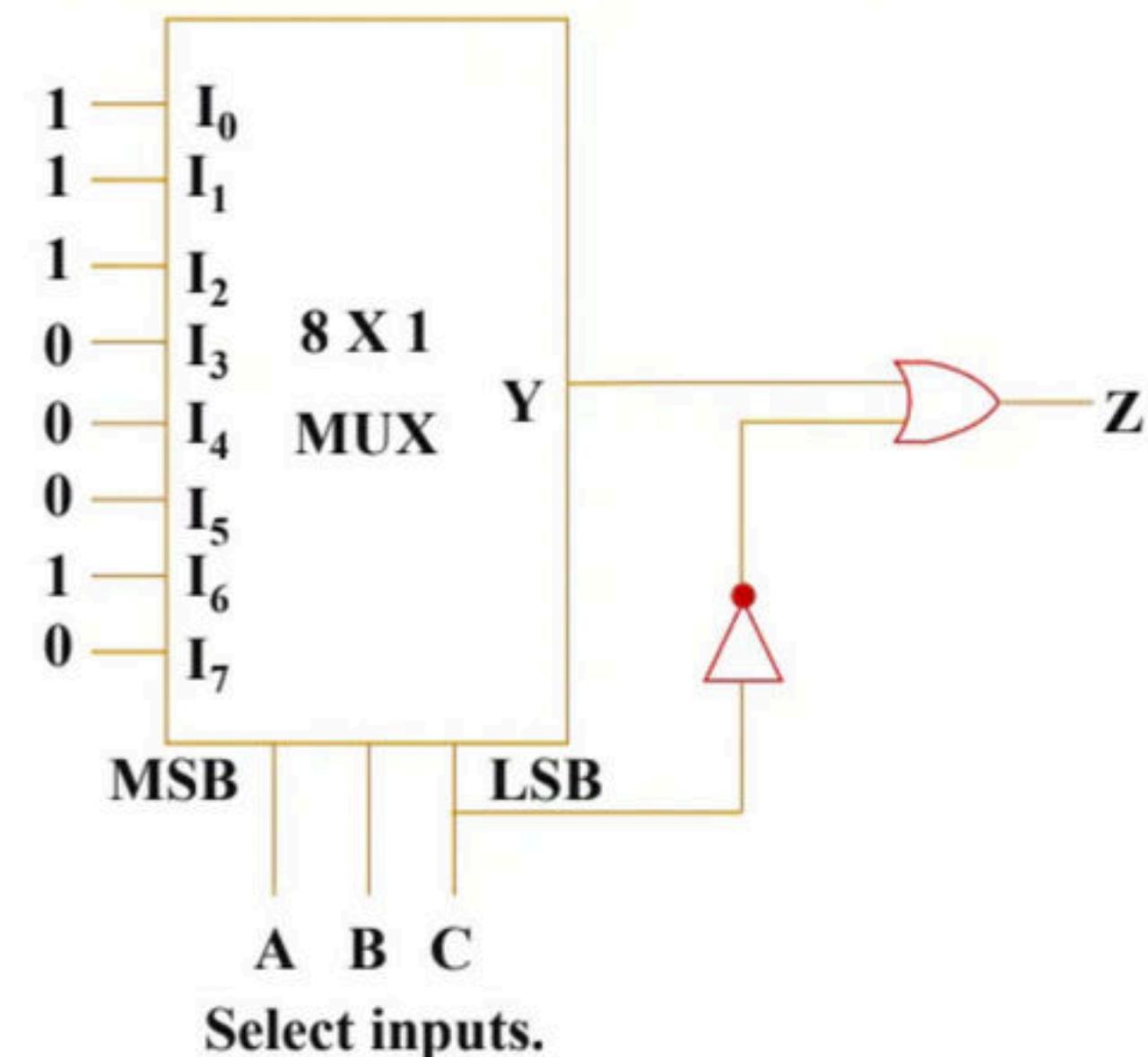
(b) $\overline{(\bar{X} \oplus Y) \oplus Z}$

(d) $(X \oplus Y) \oplus Z$



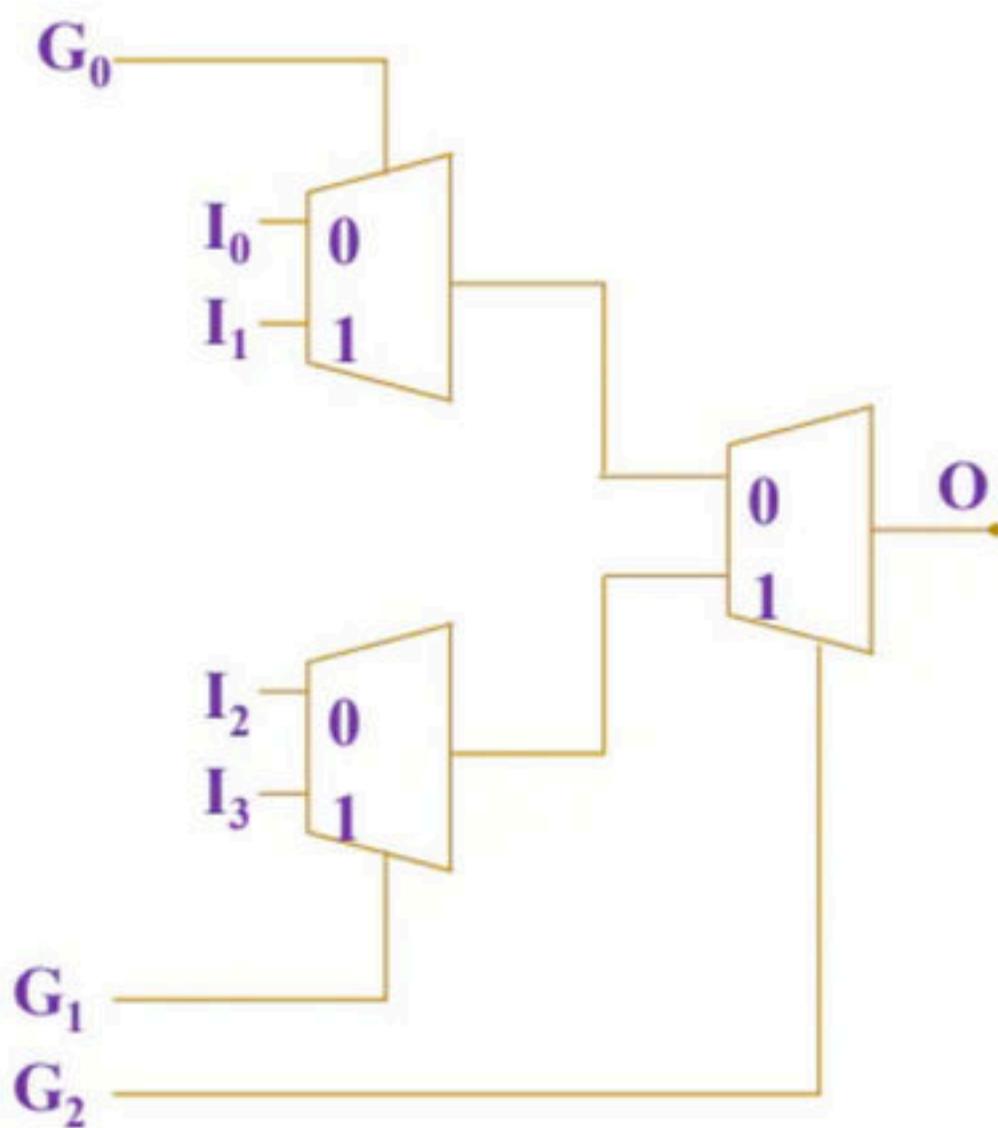
Q. A combinational circuit using an 8×1 multiplexer as shown in the figure. The minimized expression for the output (Z) is

- (a) $C(\bar{A} + \bar{B})$
- (b) $C(A + B)$
- (c) $\bar{C} + \bar{A}\bar{B}$
- (d) $\bar{C} + AB$



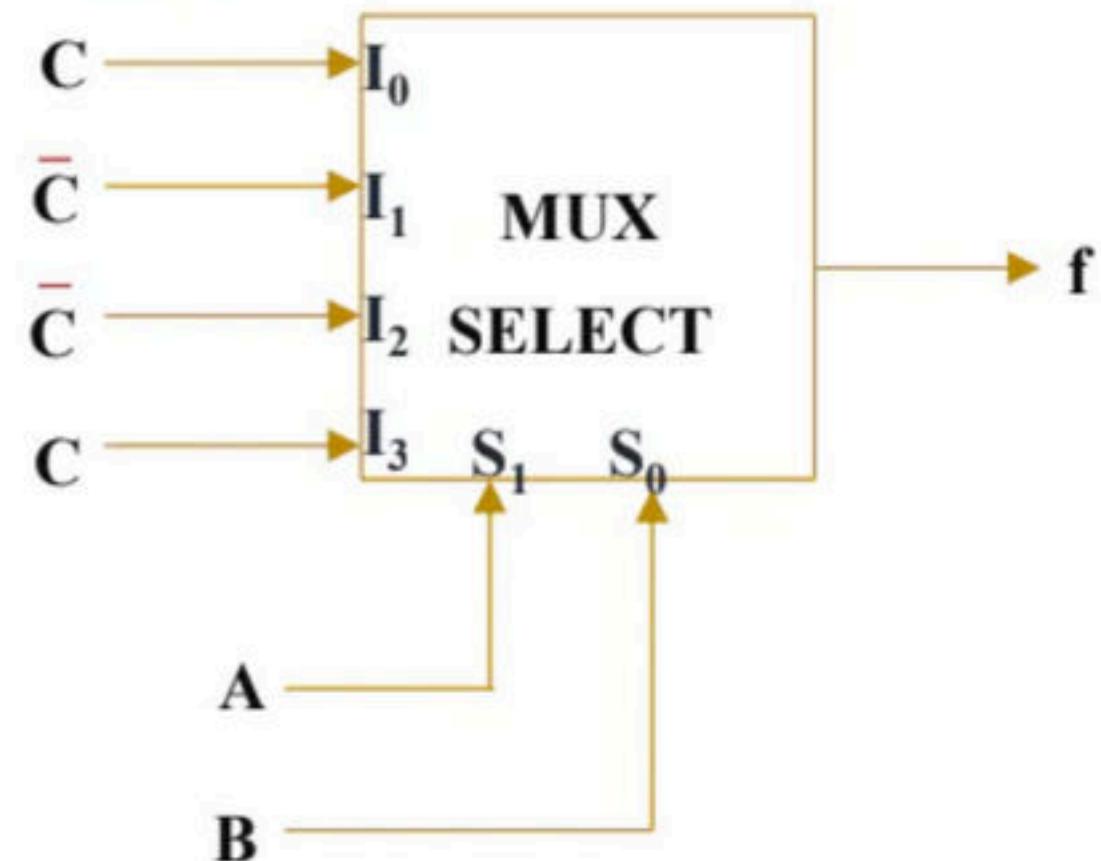
Q. The cell of a Field Programmable Gate Array is shown in the figure. It has three 2 to 1 multiplexers with their select lines G_0 , G_1 , G_2 and 4 digital signal input lines I_0 , I_1 , I_2 and I_3 . The logical function that relates the output O to the select and signal input lines is.

- (a) $\bar{G}_0\bar{G}_1I_2 + \bar{G}_0\bar{G}_1I_3 + \bar{G}_2\bar{G}_1I_0 + \bar{G}_2\bar{G}_1I_1$
- (b) $\bar{G}_0I_2 + \bar{G}_0G_1 + \bar{G}_2I_0 + \bar{G}_2\bar{G}_1I_1 + G_0$
- (c) $\bar{G}_0\bar{G}_2I_0 + G_0\bar{G}_2I_1 + G_2\bar{G}_1I_2 + G_2G_1I_3$
- (d) $G_2G_1\bar{I}_2 + \bar{G}_2\bar{G}_1\bar{I}_3 + G_2\bar{G}_0I_0 + G_0\bar{G}_2I_1$



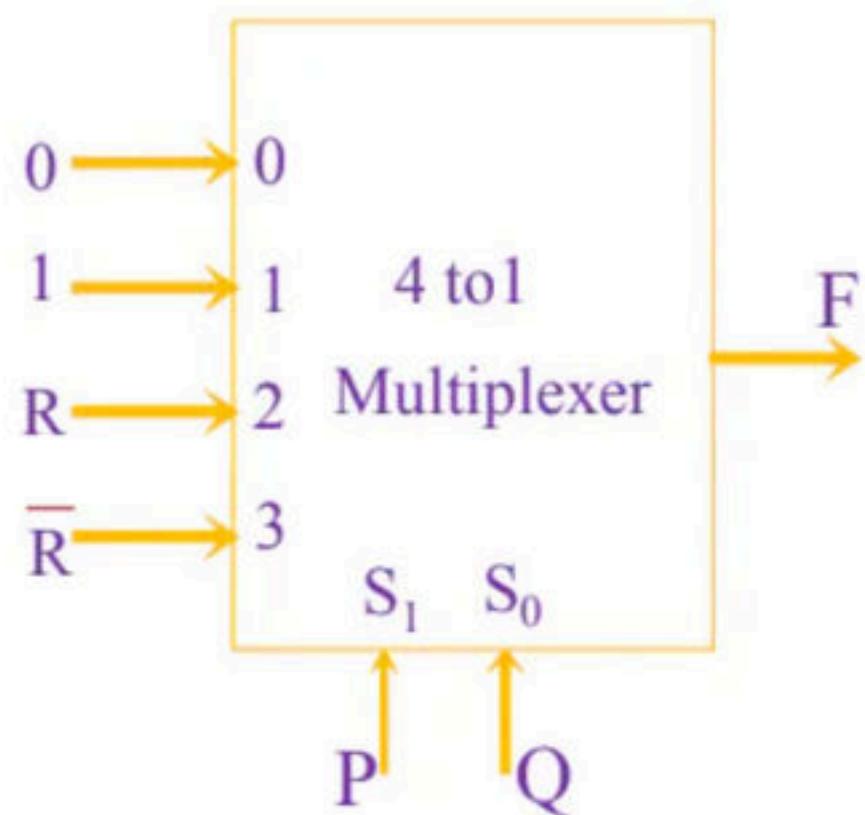
Q. The output 'F' of the multiplexer circuit shown in the figure will be

- (a) $AB + B\bar{C} + \bar{C}A + \bar{B}\bar{C}$
- (b) $A \oplus B \oplus C$
- (c) $A \oplus B$
- (d) $\overline{ABC} + \overline{ABC} + AB\bar{C} + ABC$



Q. Consider the 4-to-1 multiplexer with two lines S_1 and S_0 given below. The minimal sum of products form of the Boolean expression for the output F of the Multiplexer is

- (A) $\bar{P}Q + Q\bar{R} + P\bar{Q}R$
- (B) $\bar{P}Q + \bar{P}Q\bar{R} + PQ\bar{R} + P\bar{Q}R$
- (C) $\bar{P}QR + \bar{P}Q\bar{R} + Q\bar{R} + P\bar{Q}R$
- (D) $PQ\bar{R}$



Q. Consider the following combinational function block involving four Boolean variables x, y, a, b where x, a, b are inputs and y is the output.

$f(x, y, a, b)$

{

if (x is 1) $y = a$;

else $y = b$;

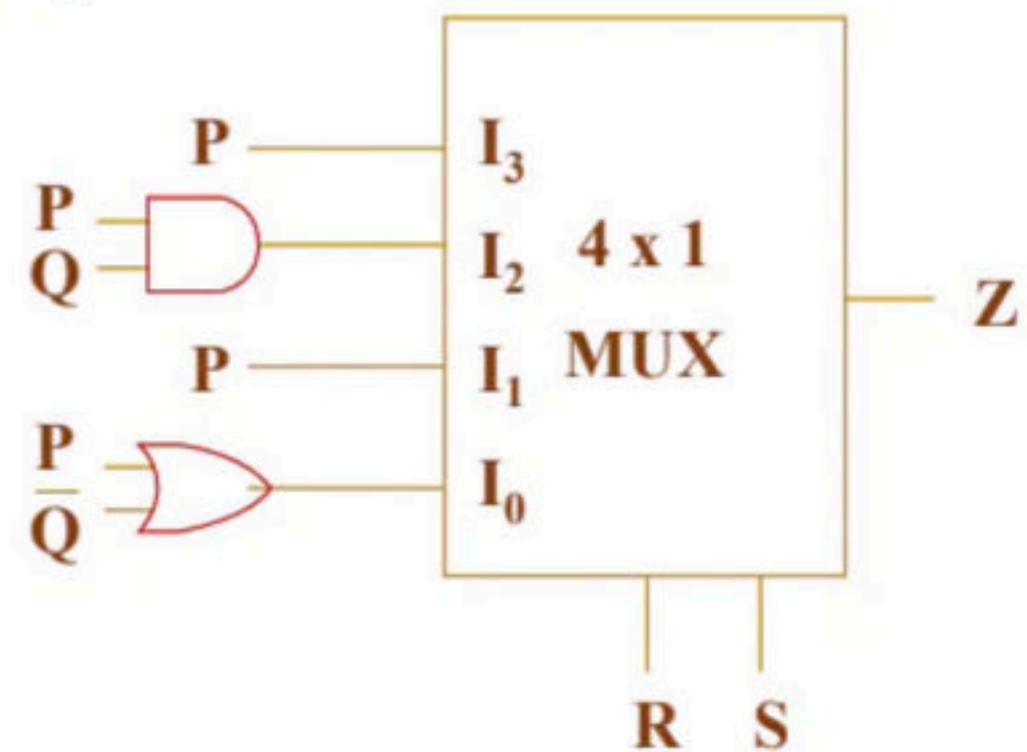
}

Which one of the following digital logic blocks is the most suitable for implementing this function?

- (A) Full adder
- (B) Priority encoder
- (C) Multiplexor
- (D) Flip-flop

Q. For the circuit shown in the following figure $I_0 - I_3$ are inputs to the 4:1 multiplexer R(MSB) and S are control bits. The output Z can be represented by

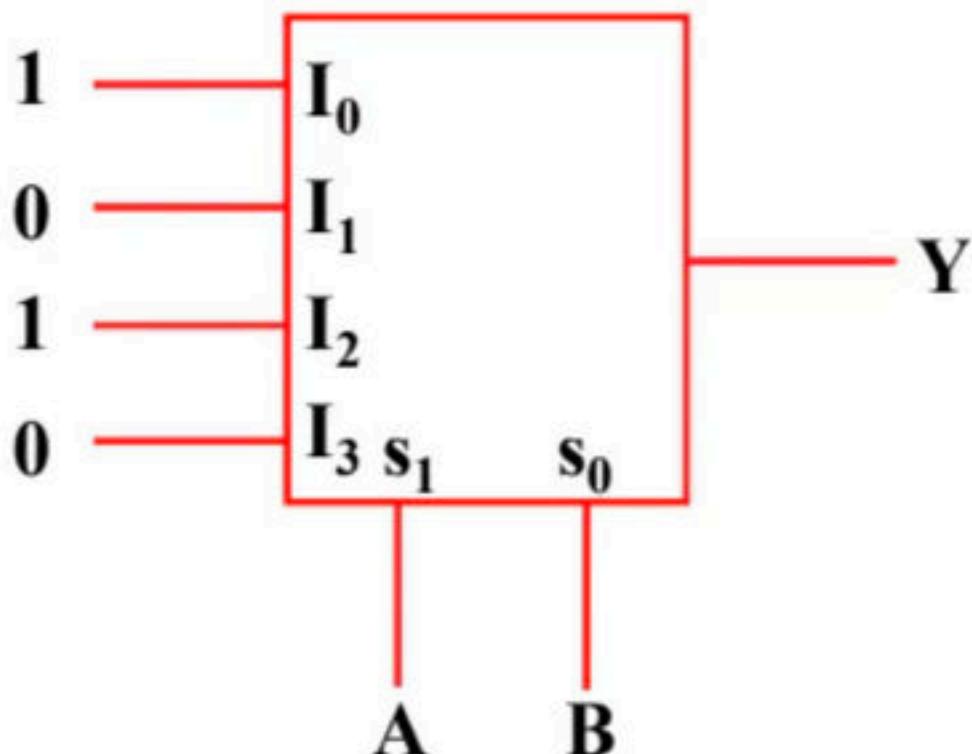
- (a) $PQ + P\bar{Q}S + \bar{Q}\bar{R}\bar{S}$
- (b) $P\bar{Q} + PQ\bar{R} + \bar{P}\bar{Q}\bar{S}$
- (c) $P\bar{Q}\bar{R} + \bar{P}QR + PQRS + \bar{Q}\bar{R}\bar{S}$
- (d) $PQR + PQRS + P\bar{Q}\bar{R}S + \bar{Q}\bar{R}\bar{S}$



Q. The logical expressions of the output of a 4×1 multiplexer shown below is

- (a) $A + \bar{B}$
- (c) \bar{A}

- (b) \bar{B}
- (d) B



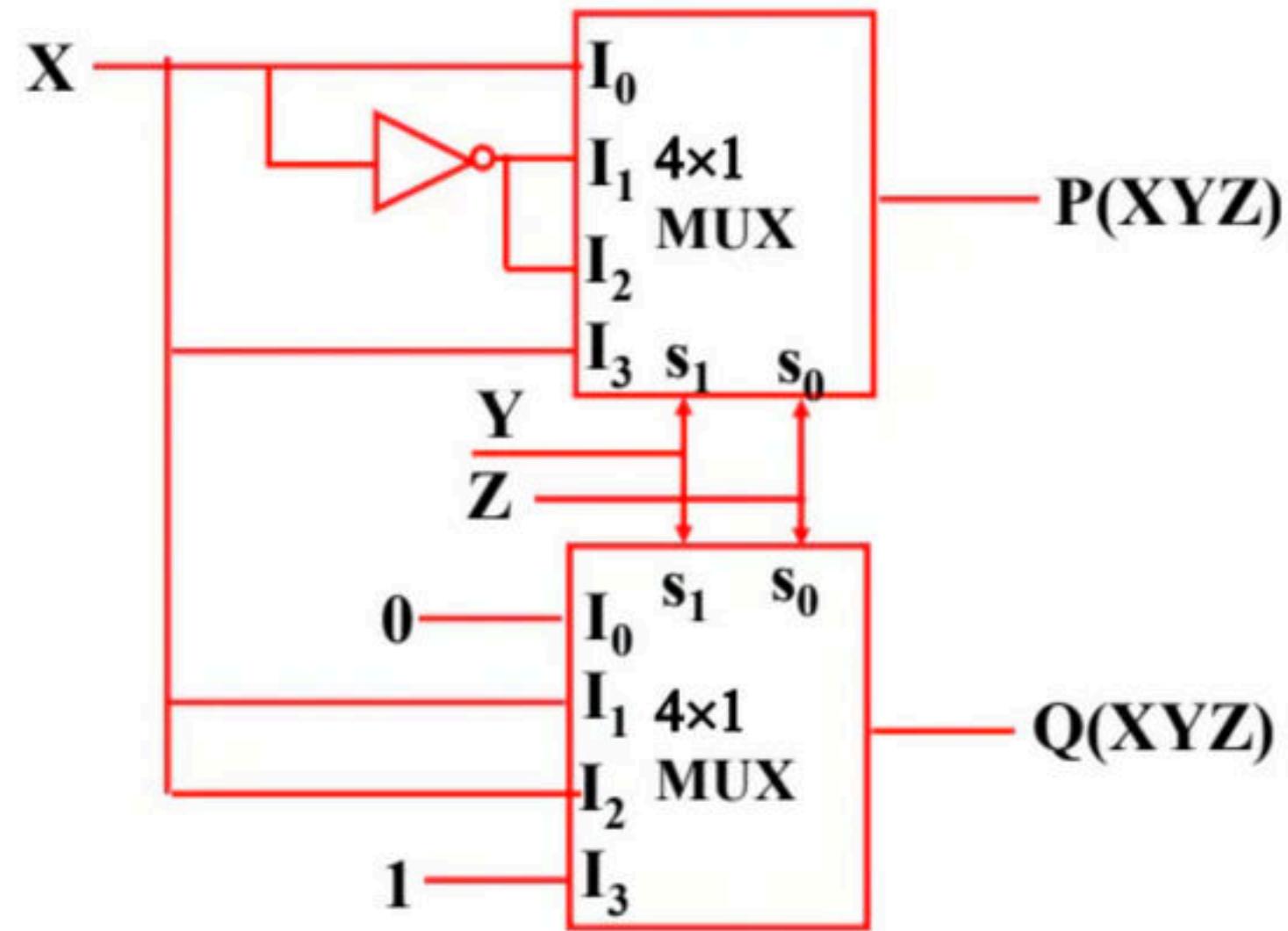
Q. The multiplexer circuit function as

(a) Full subtractor

(c) Two output comparator

(b) Full adder

(d) Half adder



Q. If the logic expression of the outputs in the circuit shown in figure A and B are same, then select the correct combination of signals to be connected to the inputs of multiplexer

- | | | | | |
|-----|-------|-----------|-----------|-------|
| | I_0 | I_1 | I_2 | I_3 |
| (a) | C | 0 | \bar{C} | 1 |
| (b) | C | C | \bar{C} | C |
| (c) | C | \bar{C} | \bar{C} | C |
| (d) | 1 | C | \bar{C} | 1 |

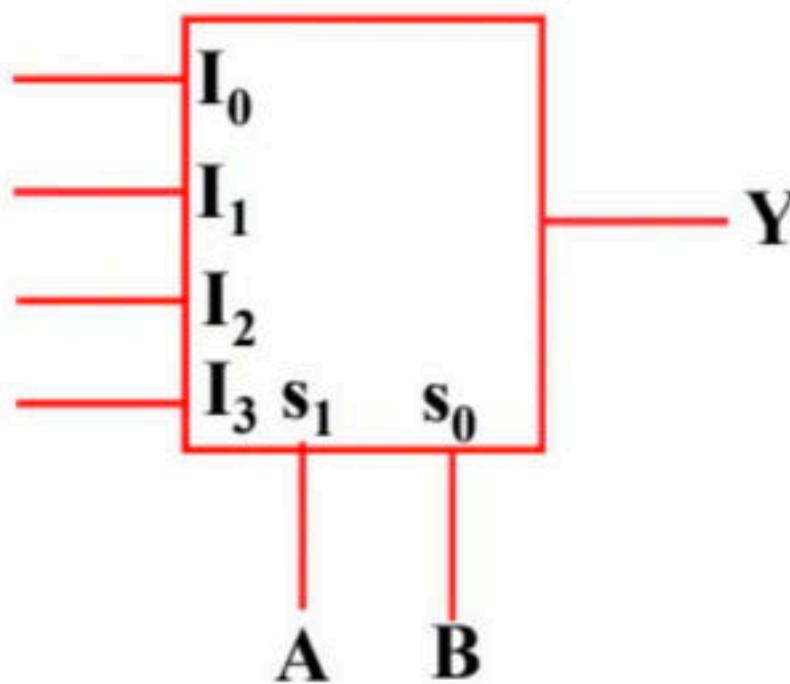


Figure A

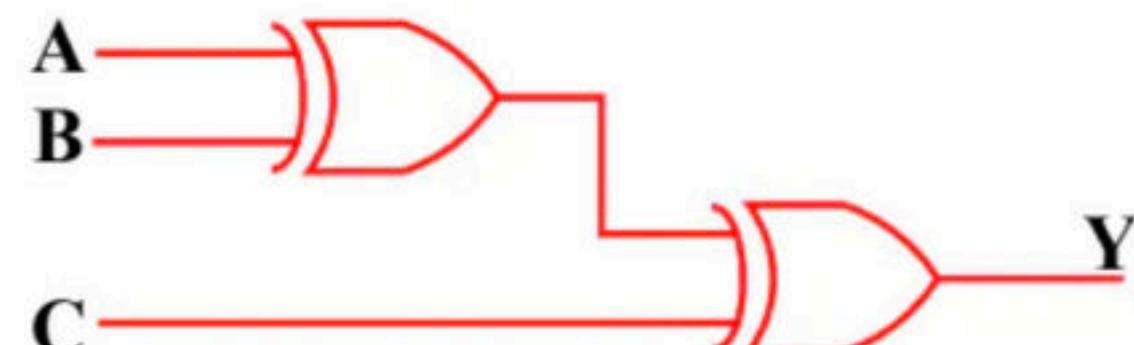


Figure B

Q. A combinational circuit using 4×1 mux is shown in figure

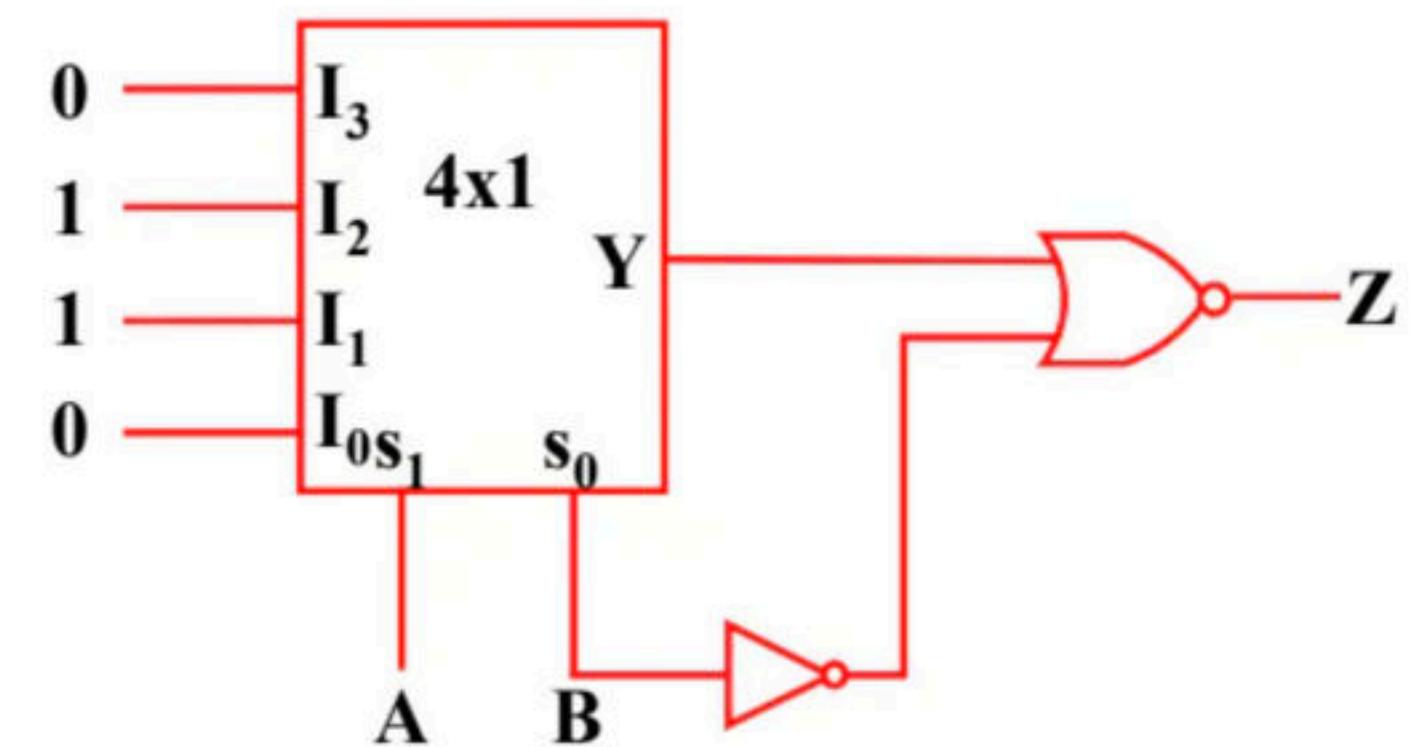
The output Z is

(a) $A + B$

(b) $\overline{A} \oplus B$

(c) AB

(d) $(\overline{A} + \overline{B})$



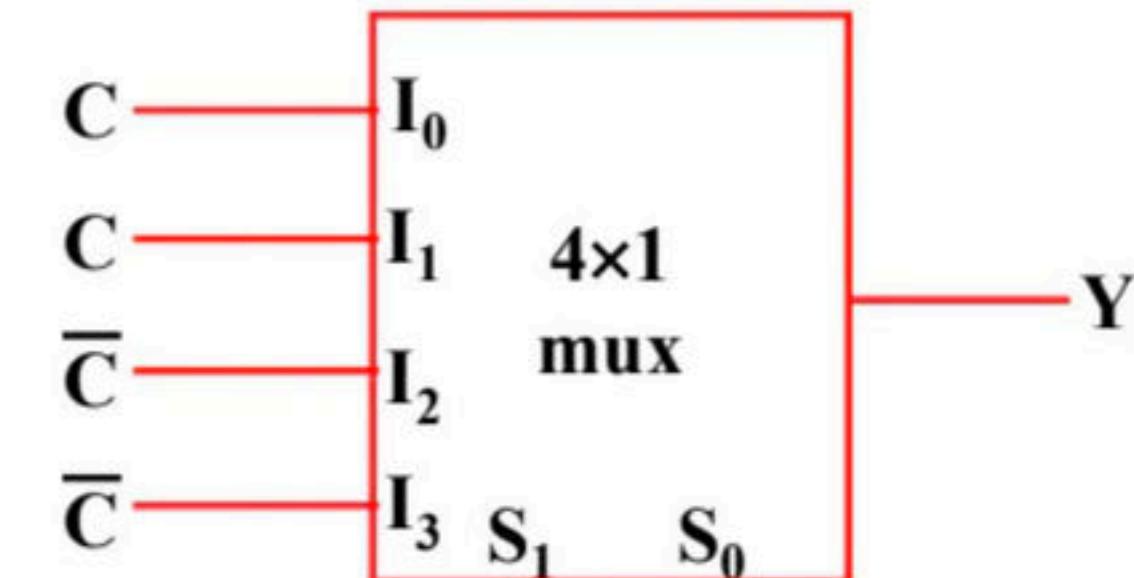
Q. The expression for Y is

(a) $A \oplus B \oplus C$

(b) $(A \oplus B)C + AB\bar{C}$

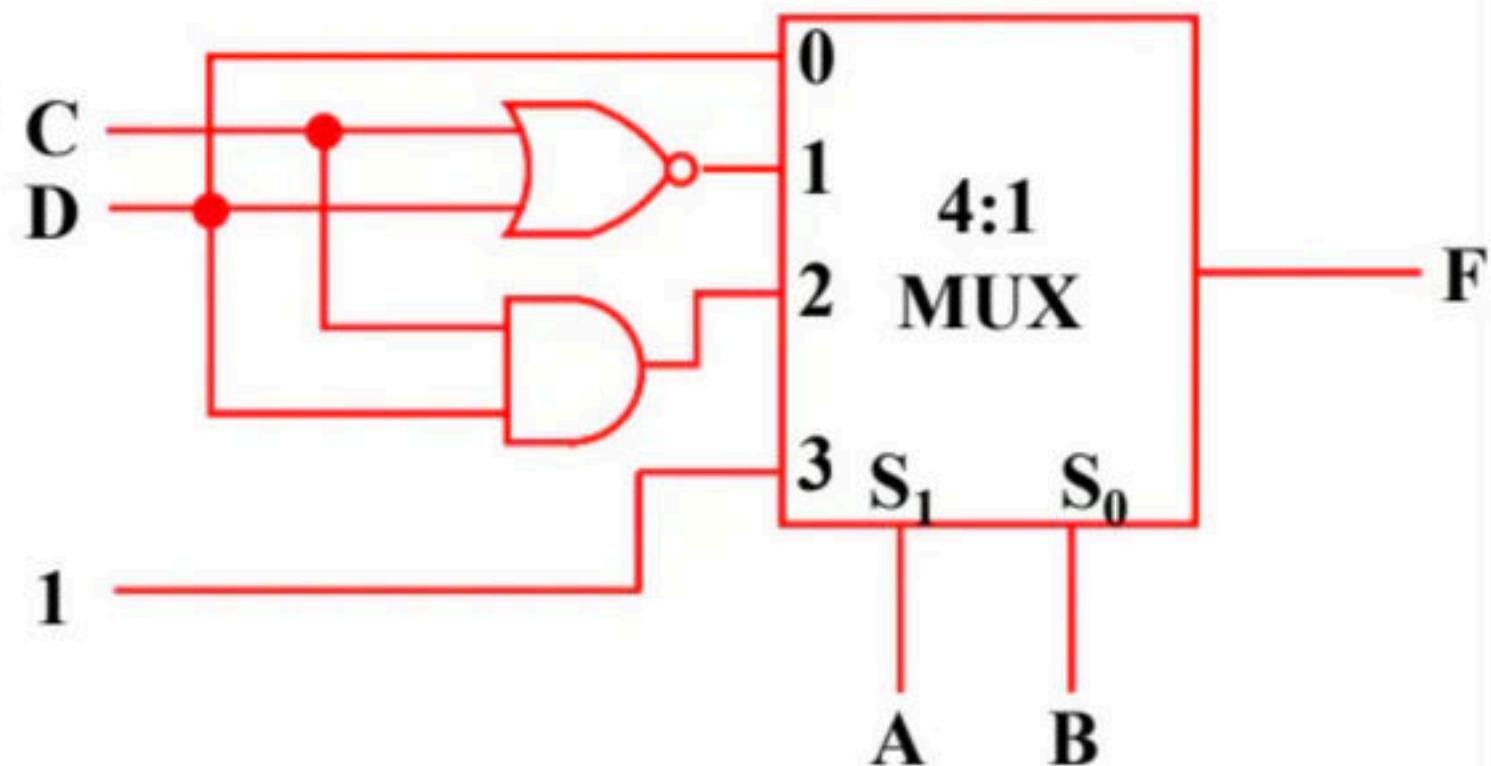
(c) $AB + A \oplus B$

(d) $ABC + (A \oplus B)\bar{C}$



Q. The Boolean function realized by the following circuit is

- (a) $F(A, B, C, D) = \Sigma(1, 3, 4, 11, 12, 13, 14, 15)$
- (b) $F(A, B, C, D) = \Sigma(0, 2, 4, 5, 9, 10, 11)$
- (c) $F(A, B, C, D) = \Sigma(1, 8, 14, 15)$
- (d) $F(A, B, C, D) = \Sigma(0, 2, 6, 8, 14, 15)$



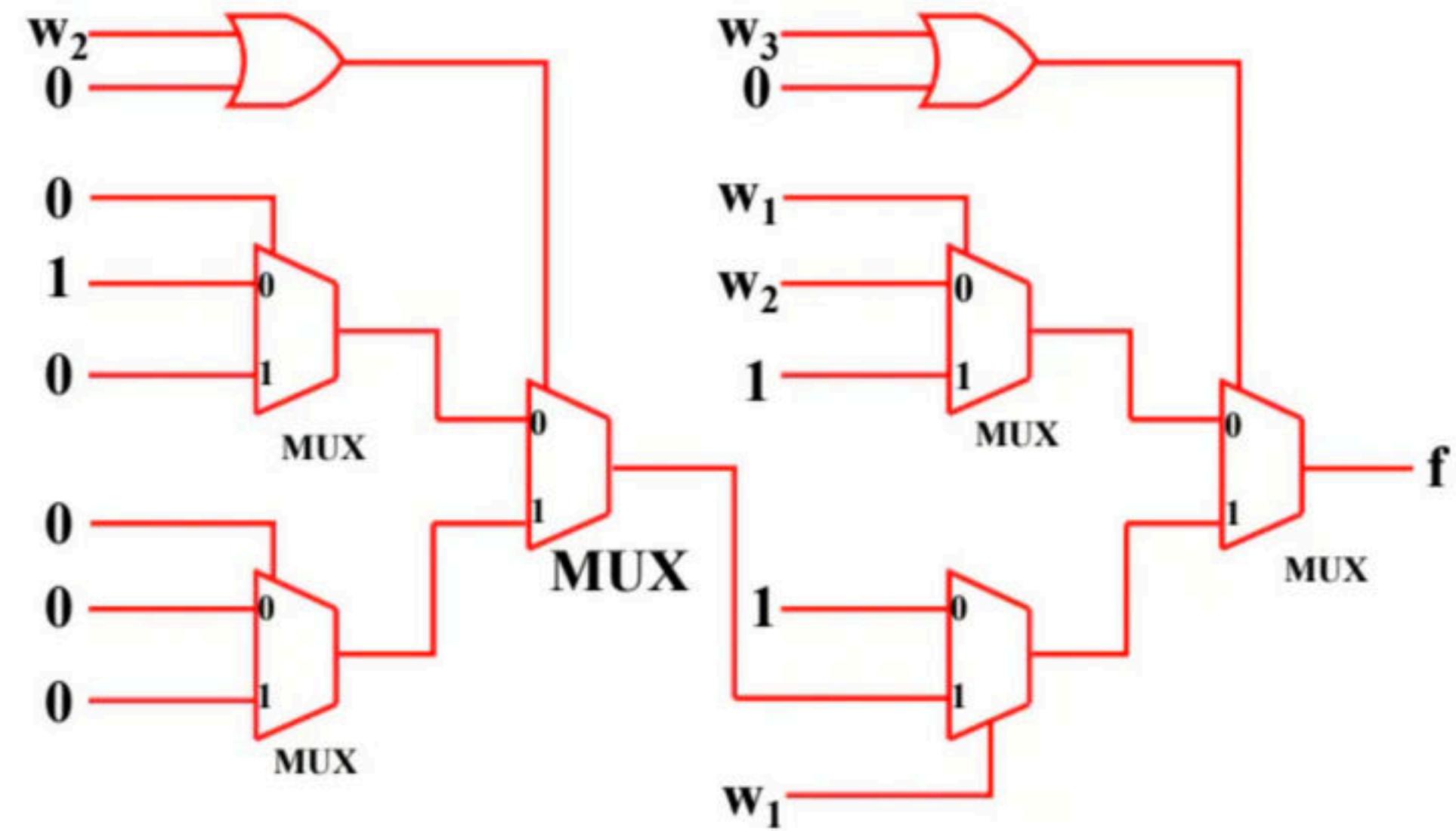
Q. The logic circuit shown below implements

(a) $(w_1 \oplus w_2) + (w_2 \oplus w_3)$

(b) $(w_2 \oplus w_1) + (w_1 \oplus w_3)$

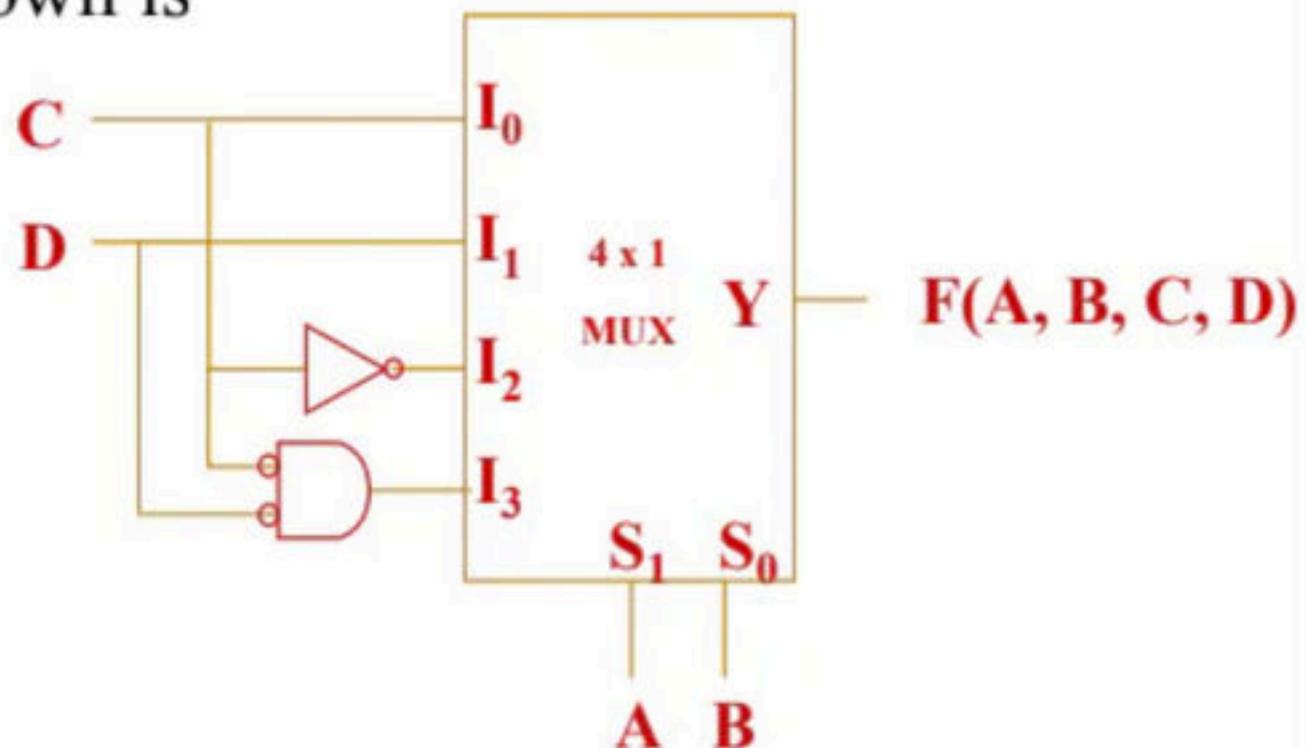
(c) $(w_2 \oplus w_3) + (w_1 \oplus w_3)$

(d) $(w_1 \oplus w_2 \oplus w_3)$



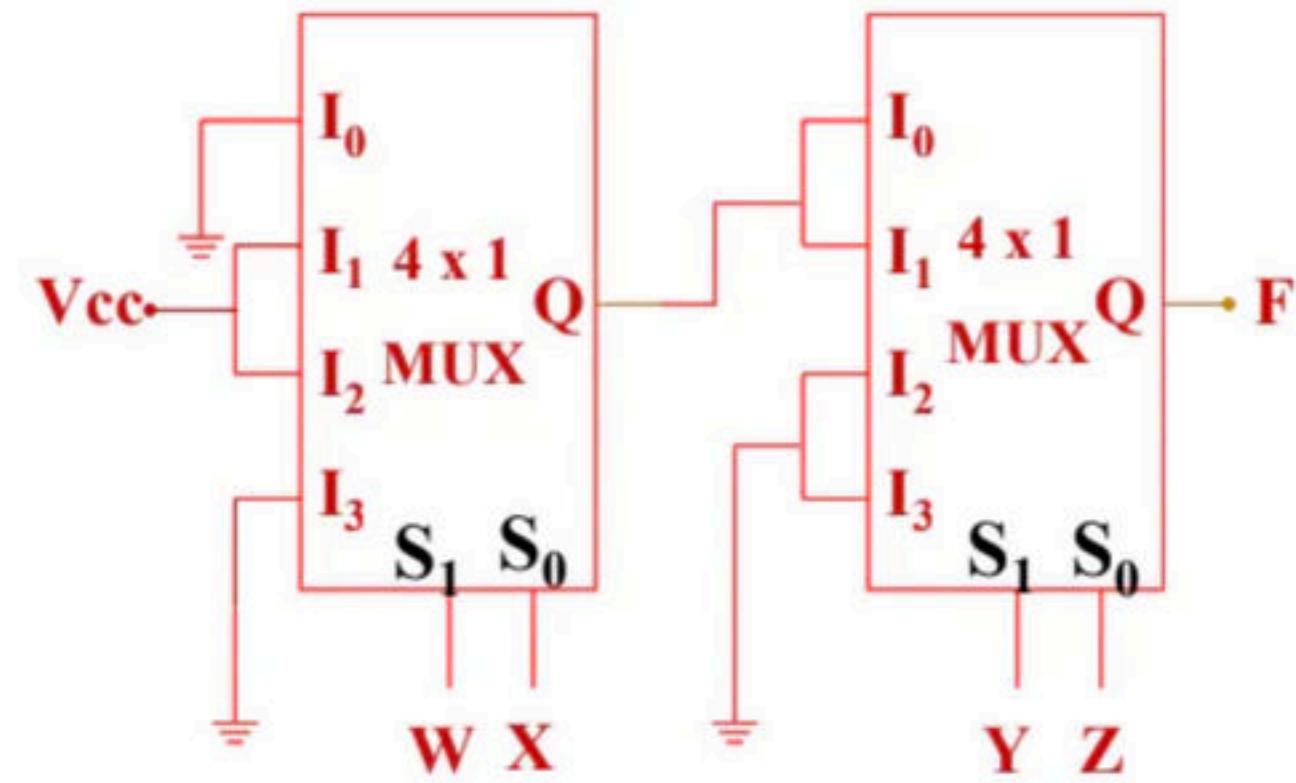
Q. The Boolean function realized by the logic circuit shown is

- (a) $F = \sum m(0,1,3,5,9,10,14)$
- (b) $F = \sum m(2,3,5,7,8,12,13)$
- (c) $F = \sum m(1,2,4,5,11,14,15)$
- (d) $F = \sum m(2,3,5,7,8,9,12)$

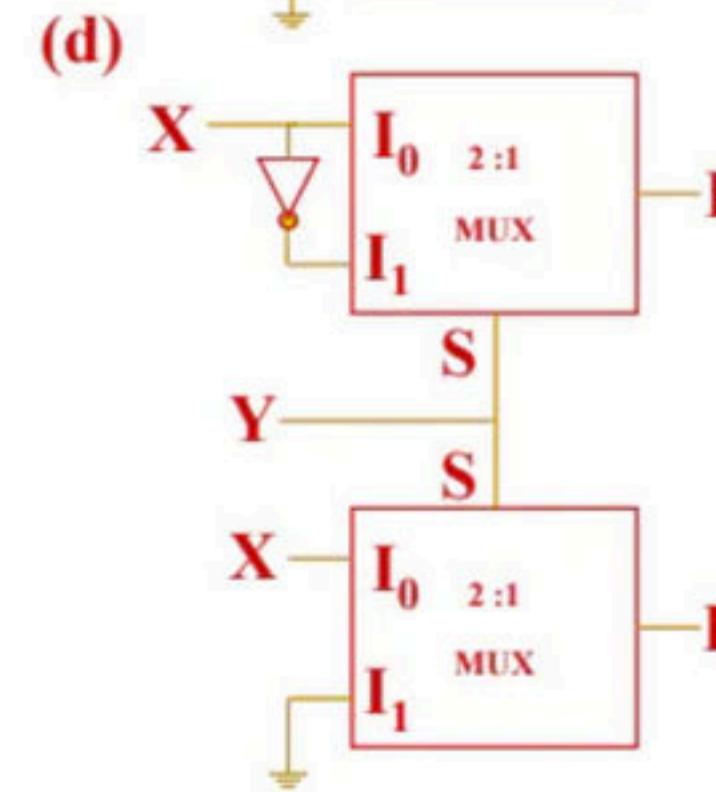
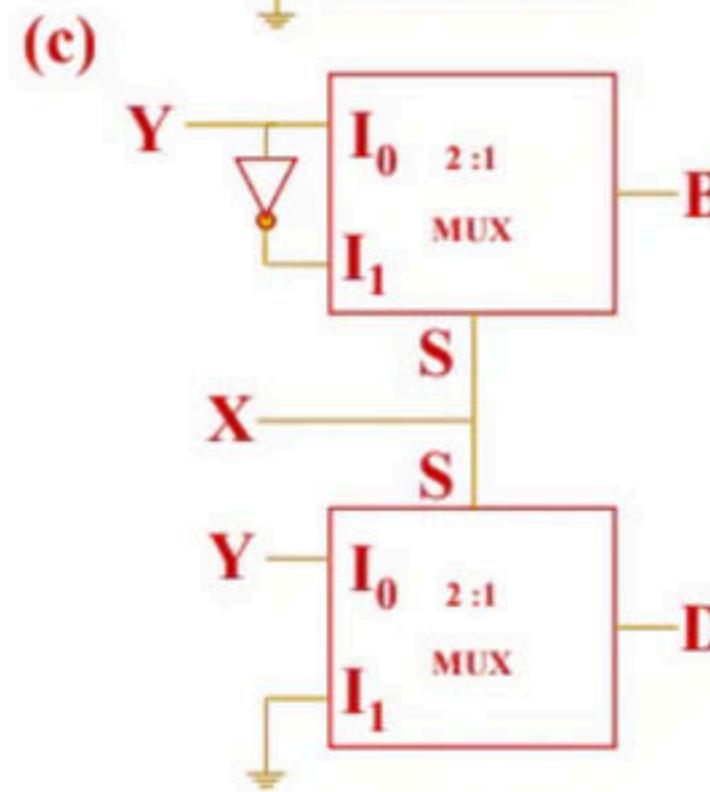
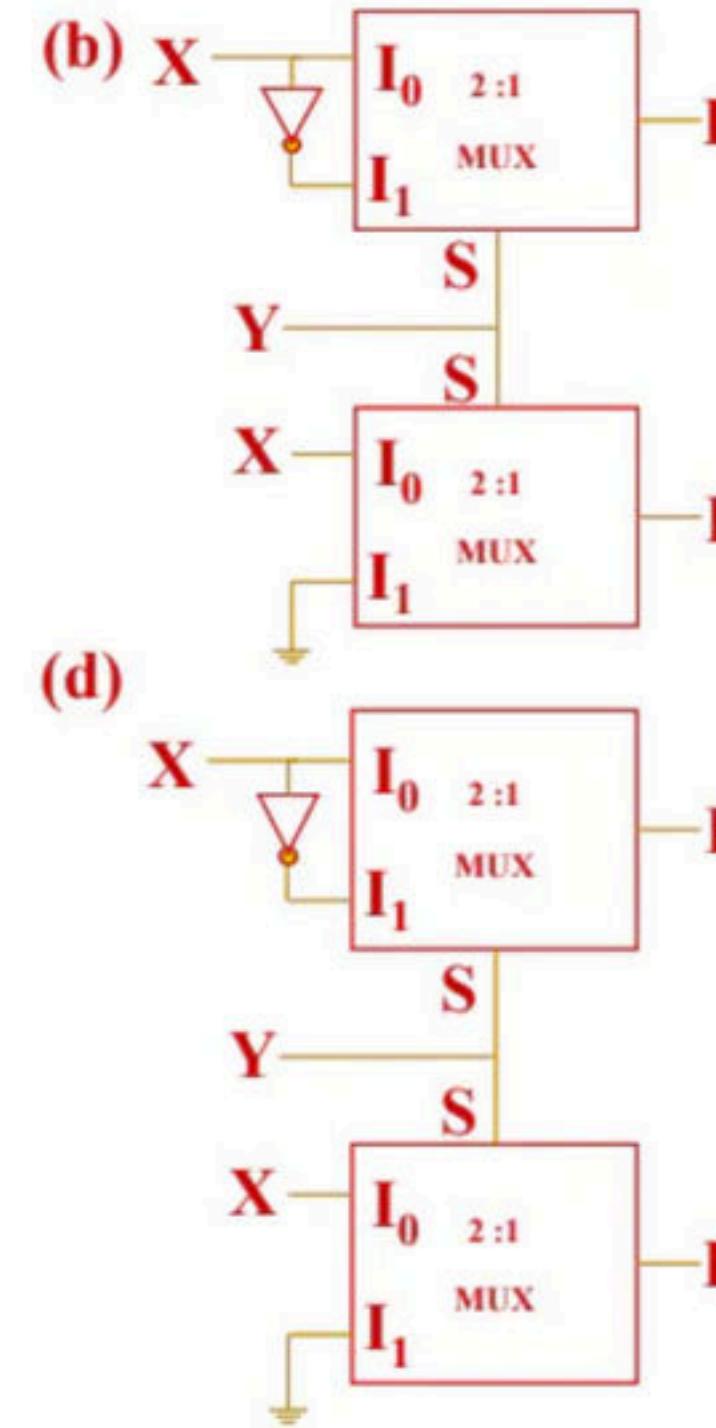
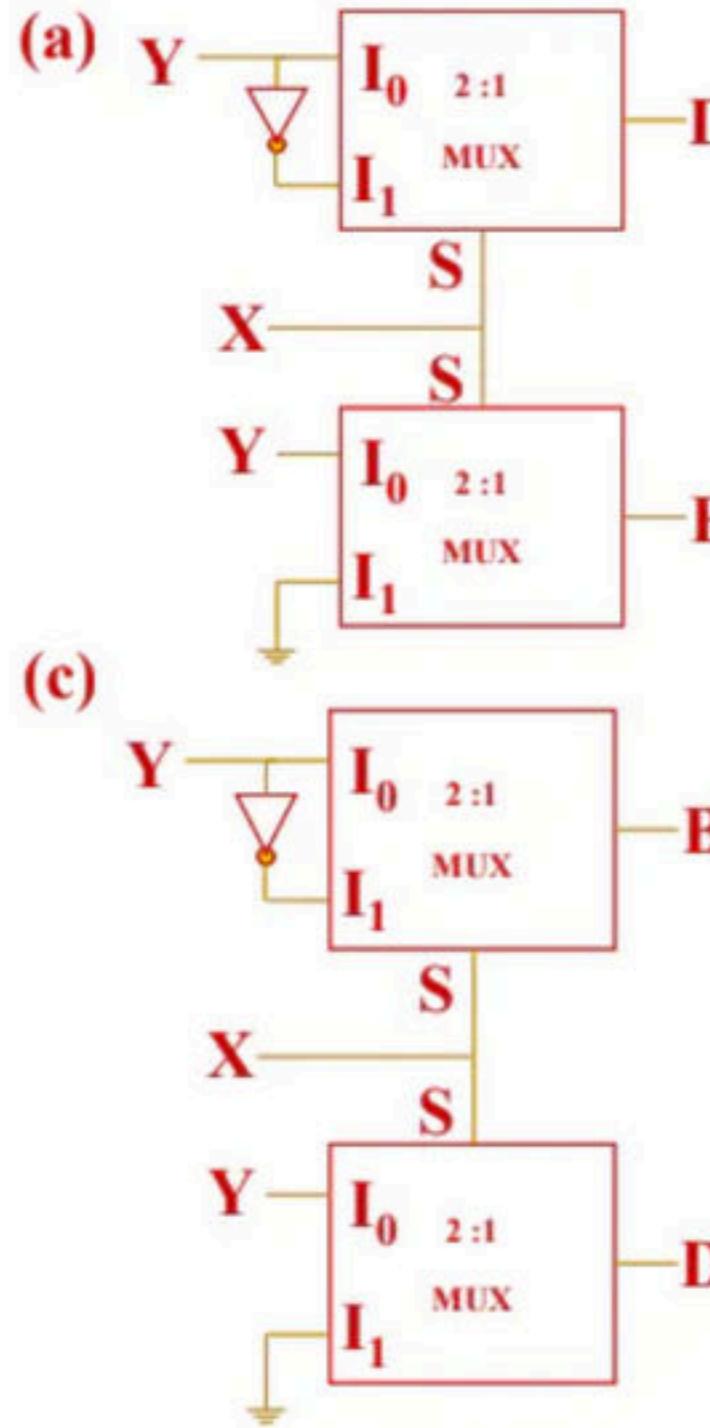


Q. In the circuit shown, W and Y are MSBs of the control inputs. The output F is given by

- (a) $F = W\bar{X} + \bar{W}X + \bar{Y}\bar{Z}$
- (b) $F = W\bar{X} + \bar{W}X + \bar{Y}Z$
- (c) $F = W\bar{X}\bar{Y} + \bar{W}XY$
- (d) $F = (\bar{W} + \bar{X}) + \bar{Y}Z$



Q. If X and Y are inputs and the Difference (D = X - Y) and the Borrow (B) are the outputs, which one of the following diagrams implements a half subtractor?



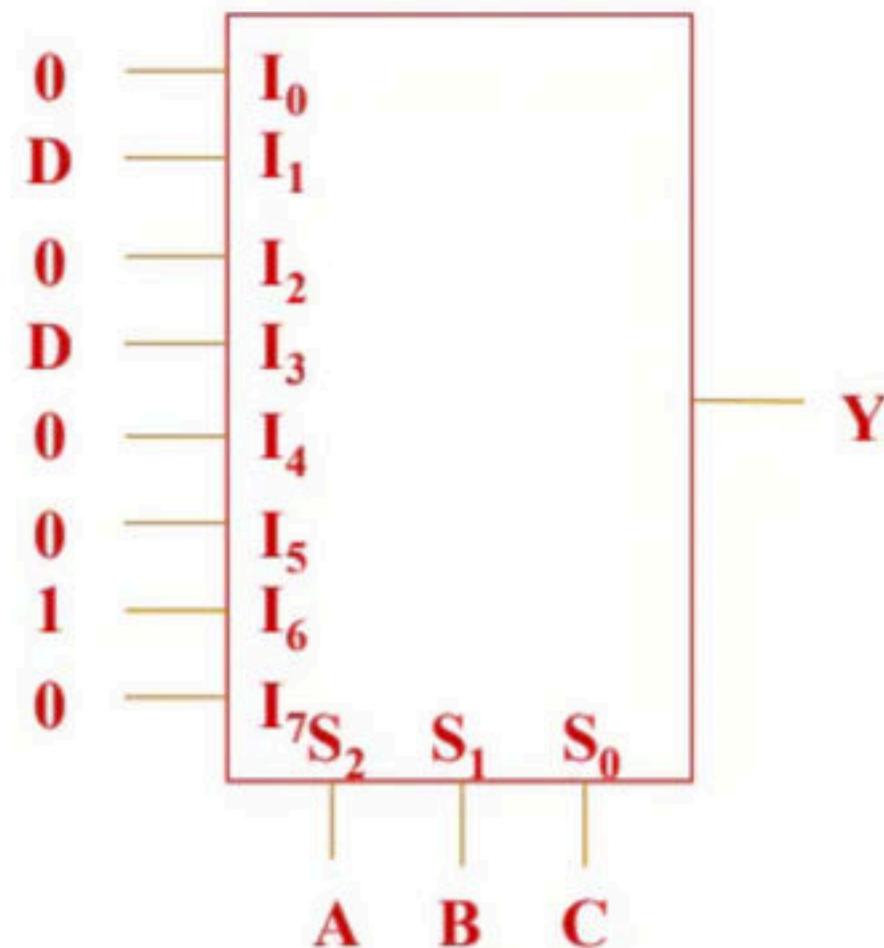
q. An 8-to-1 multiplexer is used to implement logical function Y as shown in the figure. The output Y is given by.

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

(c) $Y = AB\bar{C} + \bar{A}CD$

(b) $Y = \bar{A}BC + A\bar{B}D$

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



Q) Design a logic circuit $f(A, B, C) = \sum m(0, 1, 3, 6, 7)$ using suitable MUX

Q) Design a logic circuit $f(A, B, C) = \sum m(0, 1, 3, 6, 7)$ using 4×1 MUX

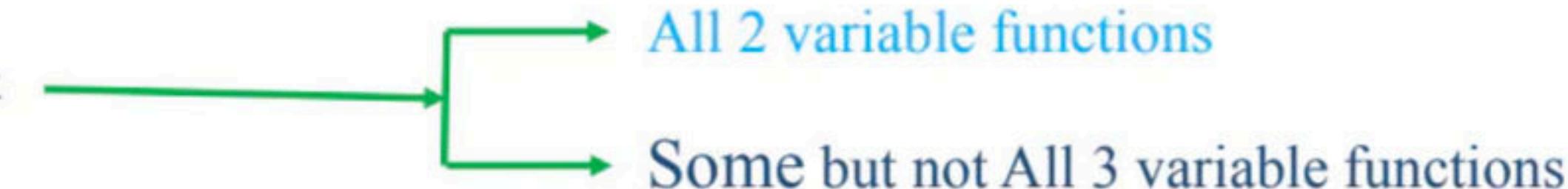
- a) AB as select lines
- b) BC as select lines
- c) AC as select lines

BC as select lines

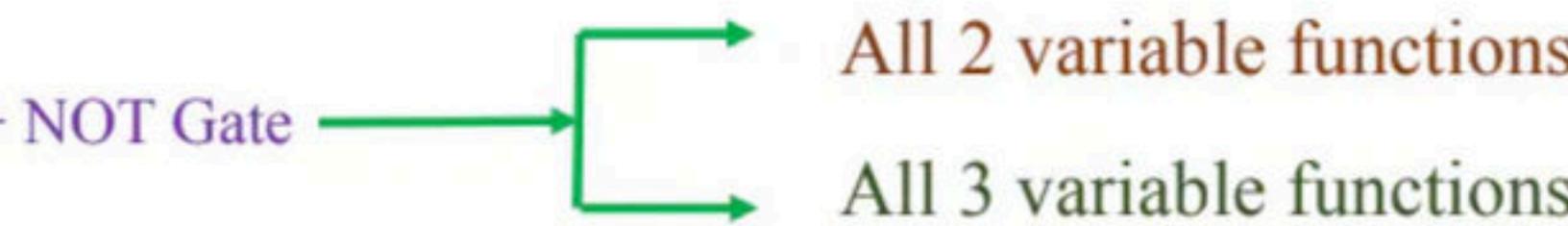
AC as select lines

Note :

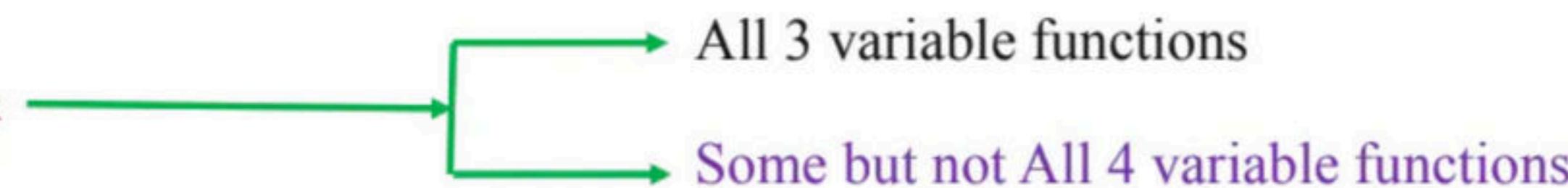
1. By using one 4×1 Mux



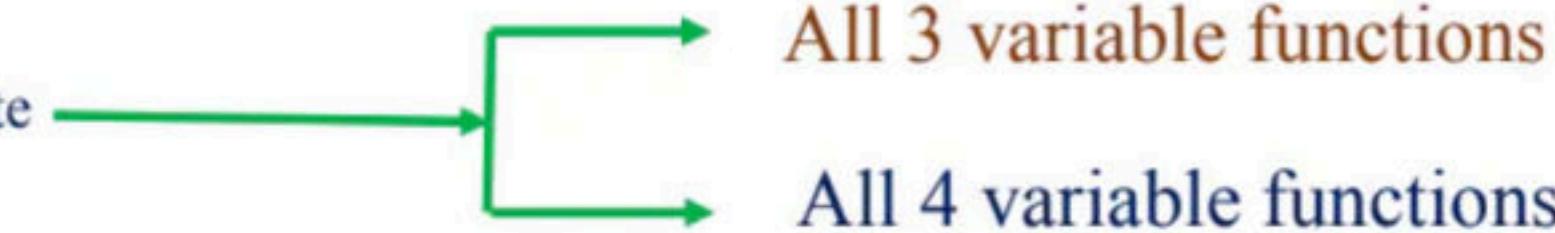
2. By using one 4×1 Mux + NOT Gate



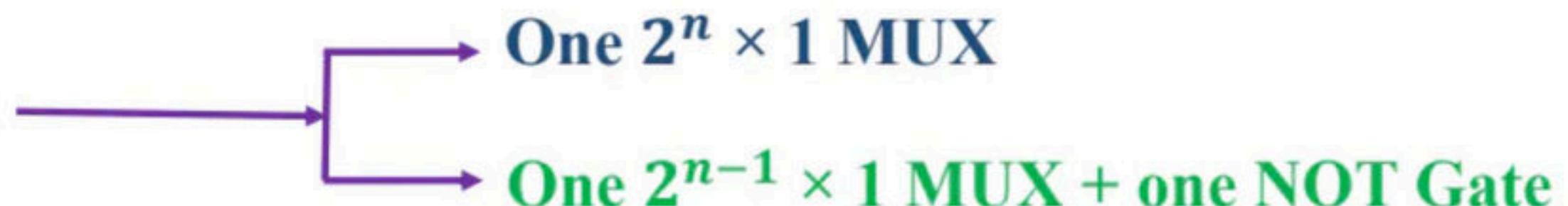
3. By using one 8×1 Mux



4. By using one 8×1 Mux + NOT Gate



5. n-variable function



Q) Suppose only one mux and one inverter are allowed to be used to implement Boolean function of n- variables , what is the minimum size of the mux needed

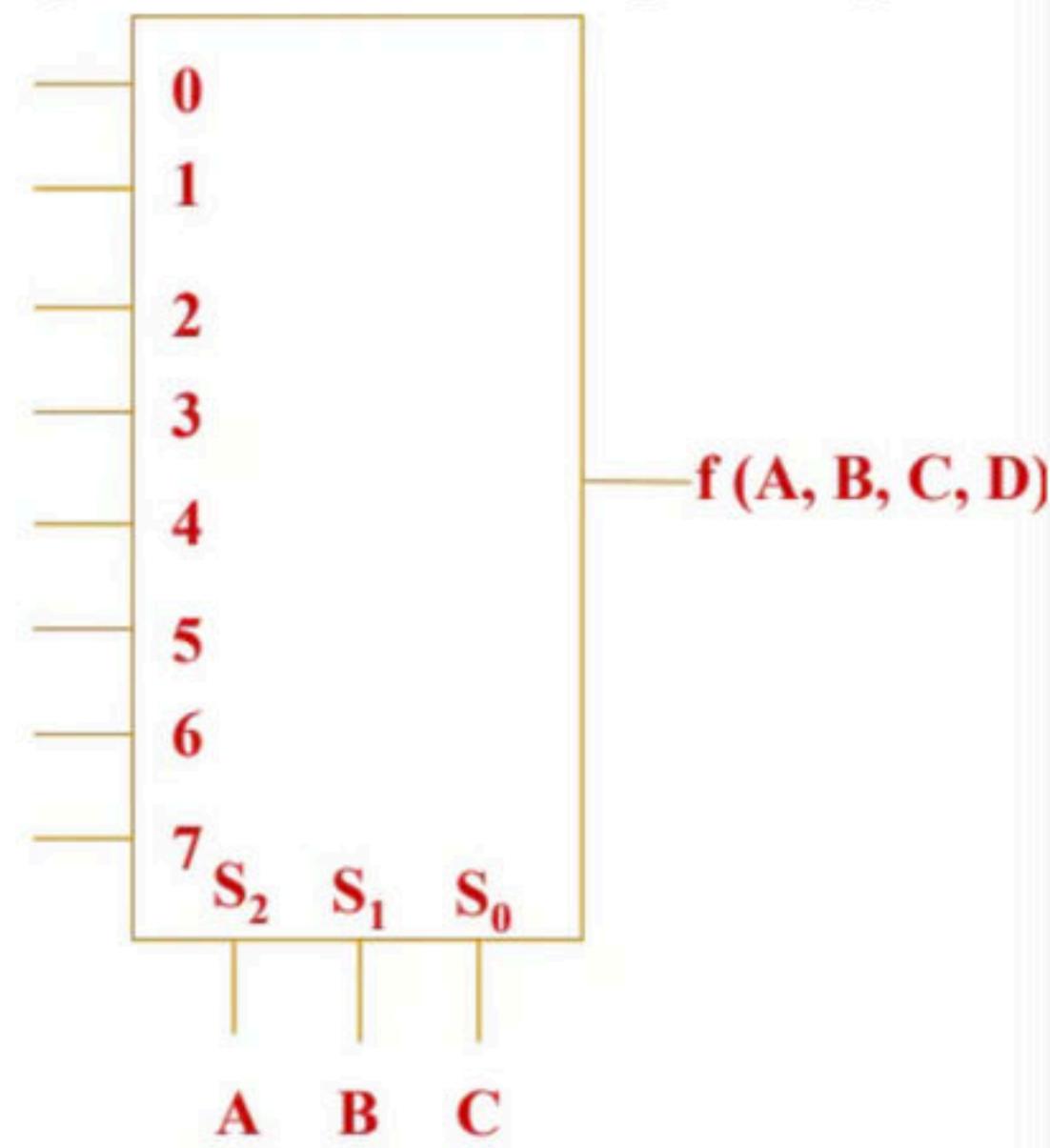
- a) $2^n \times 1$ MUX**
- b) $2^{n+1} \times 1$ MUX**
- c) $2^{n-1} \times 1$ MUX**
- d) $2^{n-2} \times 1$ MUX**

Q) Without using any additional circuitry an 8×1 mux can be used to obtain

- a) Some but not all Boolean functions of 3 variables
- b) All functions of 3 variable & none of 4- variables
- c) All function's of 4 variables
- d) All functions of 3 variables and some but not all functions of 4 variables

Q. A Boolean function $f(A, B, C, D) = \pi(1, 5, 12, 15)$ is to be implemented using an 8×1 multiplexer (A is MSB). The inputs ABC are connected to the select inputs $S_2 S_1 S_0$ of the multiplexer, respectively. Which one of the following options gives the correct inputs to pins 0,1,2,3,4,5,6,7 in order?

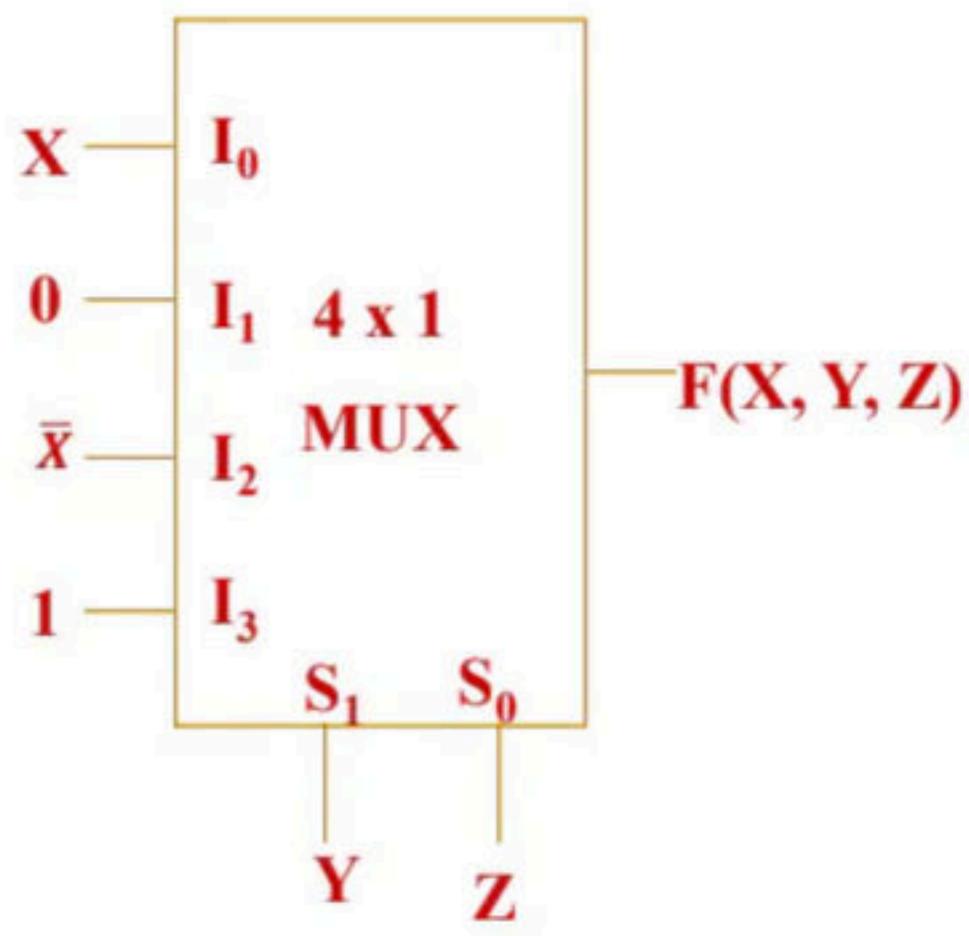
- (a) D, 0, D, 0, 0, 0, \bar{D} , D
- (b) \bar{D} , 1, \bar{D} , 1, 1, 1, D, \bar{D}
- (c) D, 1, D, 1, 1, 1, \bar{D} , D
- (d) \bar{D} , 0, \bar{D} , 0, 0, 0, D, \bar{D}



Q. A 4 to 1 multiplexer to realize a Boolean function $F(X, Y, Z)$ is shown in the figure below. The inputs Y and Z are connected to the selectors of the MUX (Y is more significant). The canonical sum-of-product expression for $F(X, Y, Z)$ is

- (a) $\sum m(2,3,4,7)$
- (c) $\sum m(0,2,4,6)$

- (b) $\sum m(1,3,5,7)$
- (d) $\sum m(2,3,5,6)$

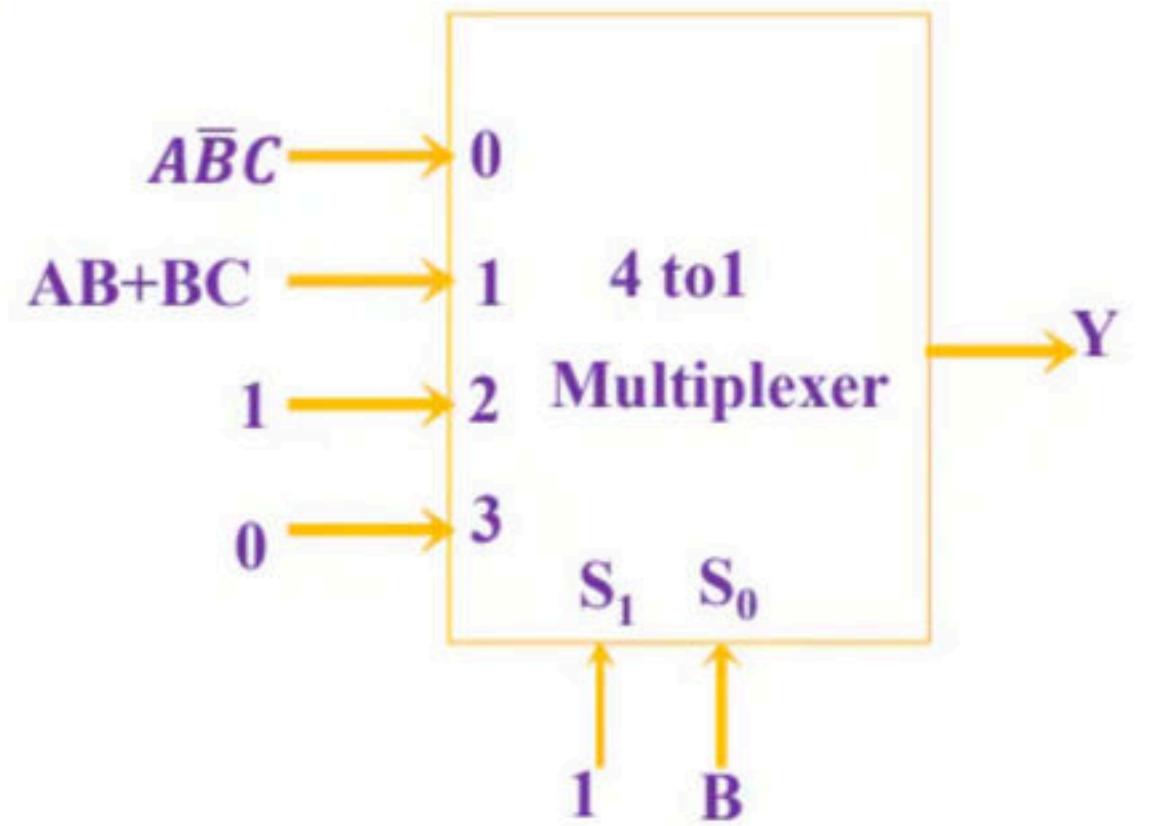


Q) Design a logic circuit $F(A, B, C) = \sum m(0, 3, 6, 7)$ using 2×1 MUX by using A as select line

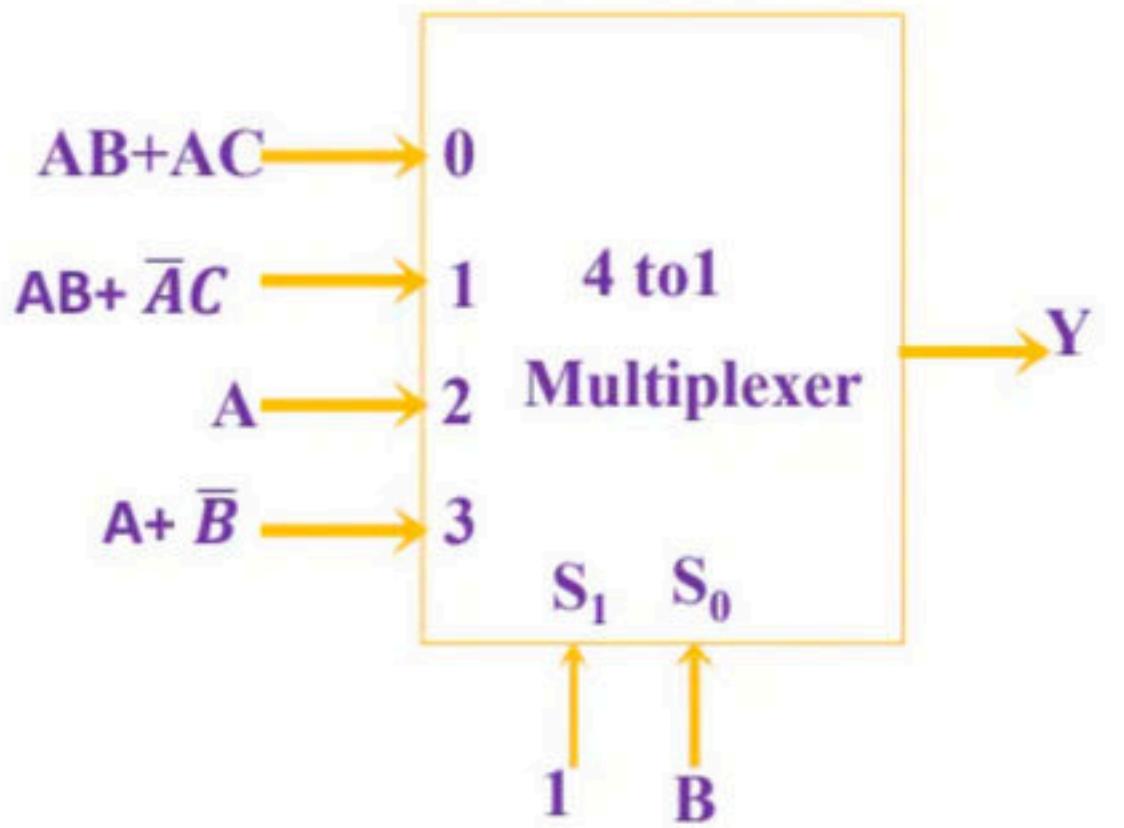
Q) Design a logic circuit $F(A, B, C) = \sum m(0, 3, 6, 7)$ using 2×1 MUX by using B as select line

MUX as Universal Gate

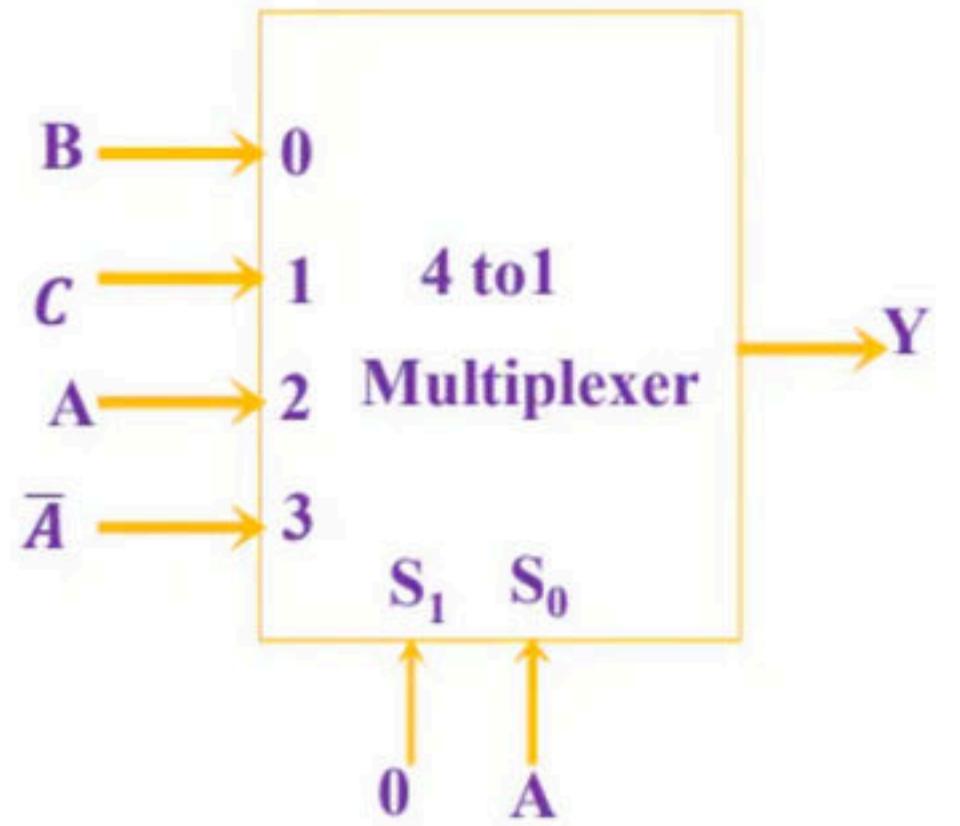
Q) Find the logic expression



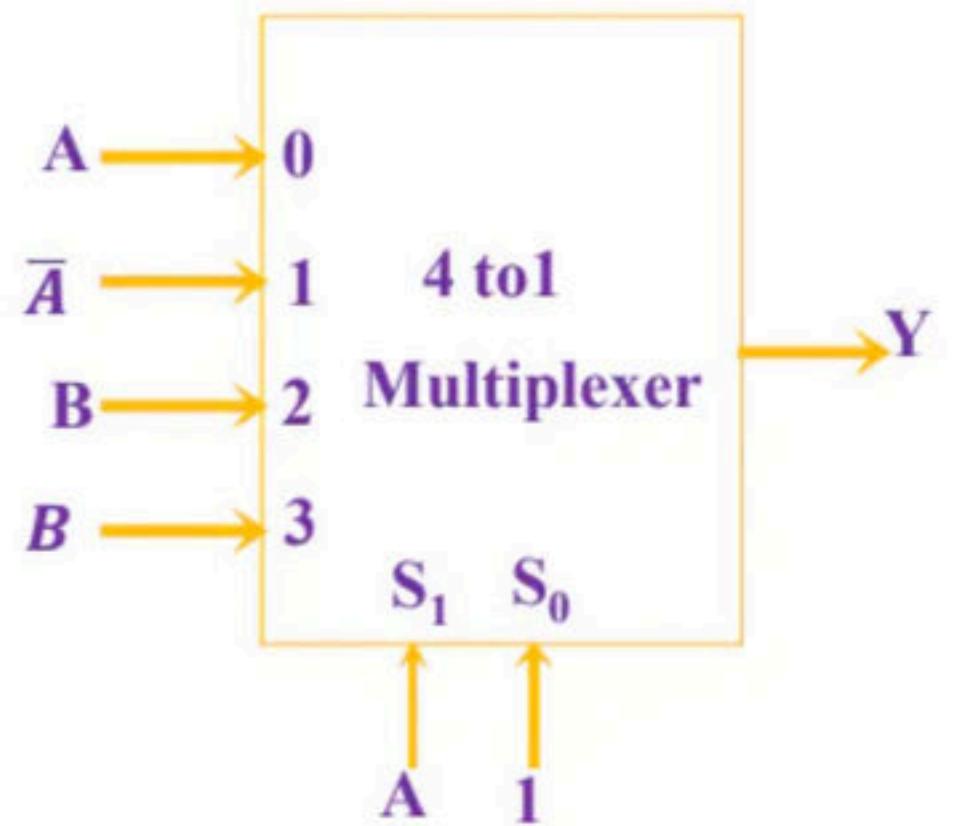
Q) Find the logic expression



Q) Find the logic expression



Q) Find the logic expression



Implementation of Higher order MUX using lower order MUX

Q) Design 4×1 MUX using 2×1 MUX

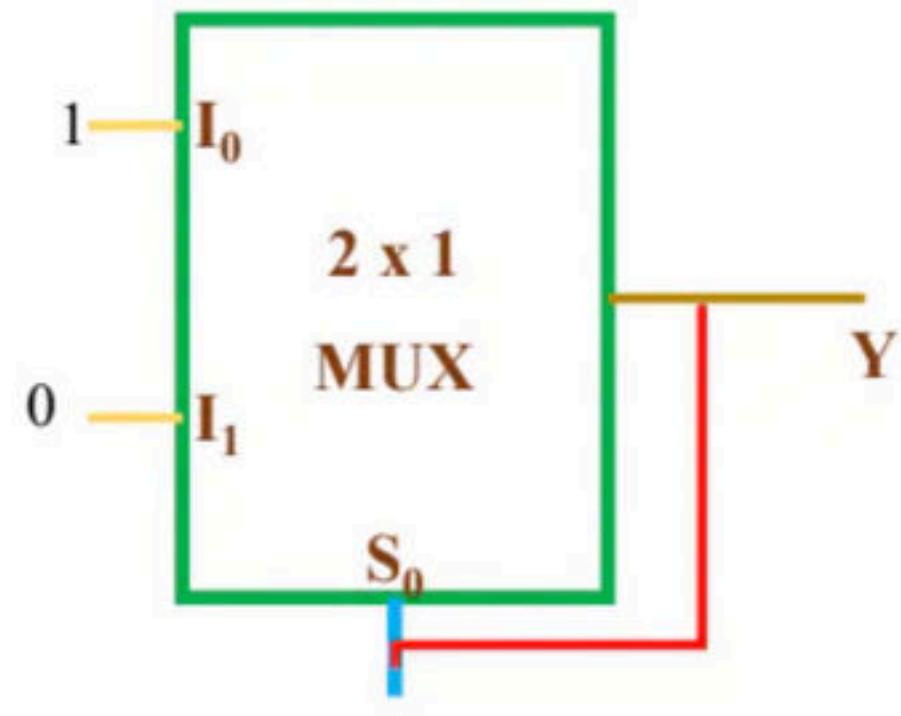
Q) Design 8×1 MUX using 2×1 MUX

Q) Design 32×1 MUX using 4×1 MUX

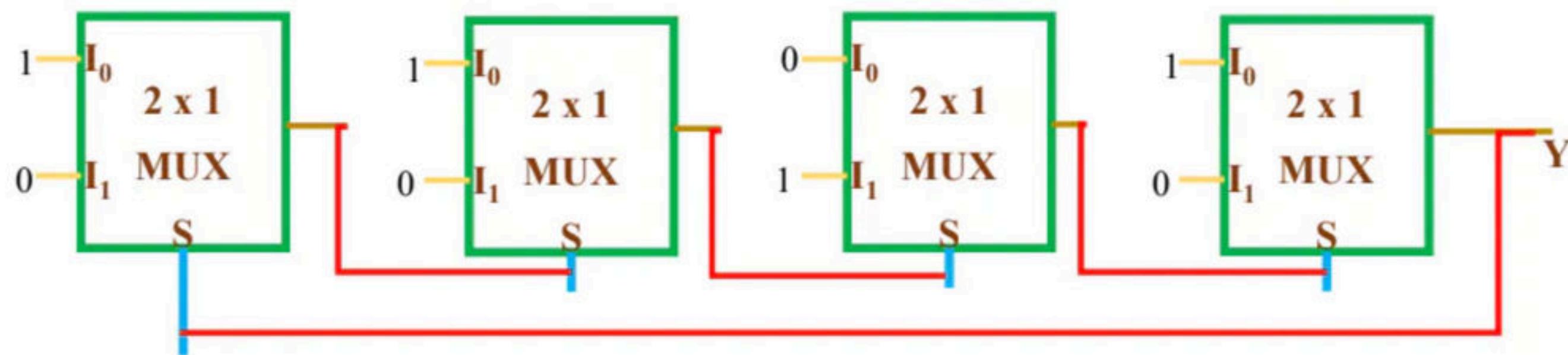
Q) Design 8×1 MUX using 4×1 MUX

Delay Analysis of MUX

Q) Draw the output waveform of the circuit , if the delay of the MUX is tpd

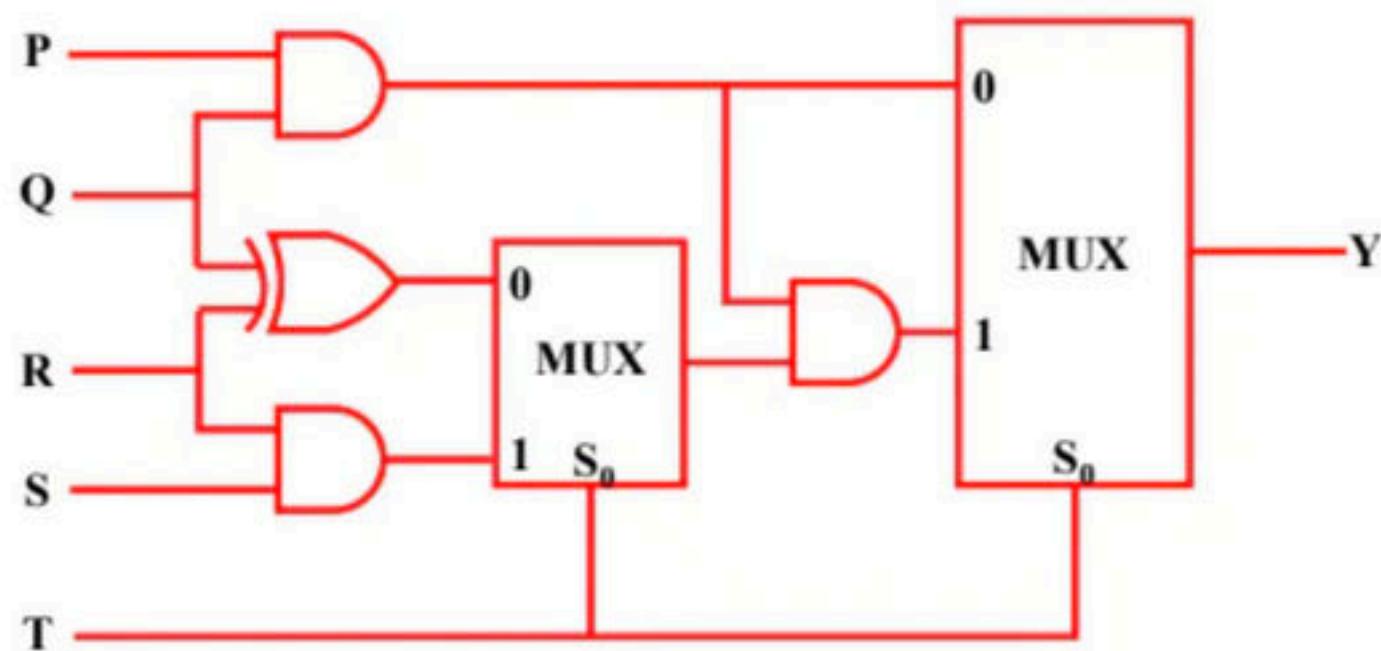


Q) Find the delay of the output Y , if the delay of each mux is 1ns



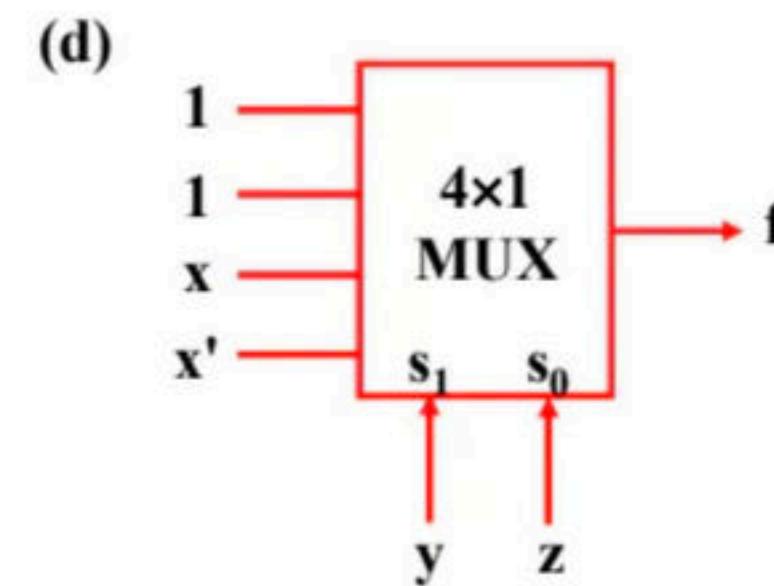
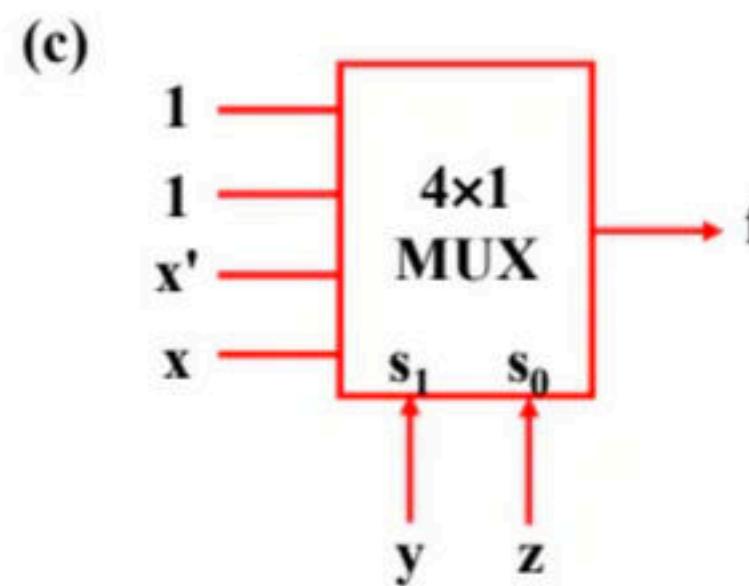
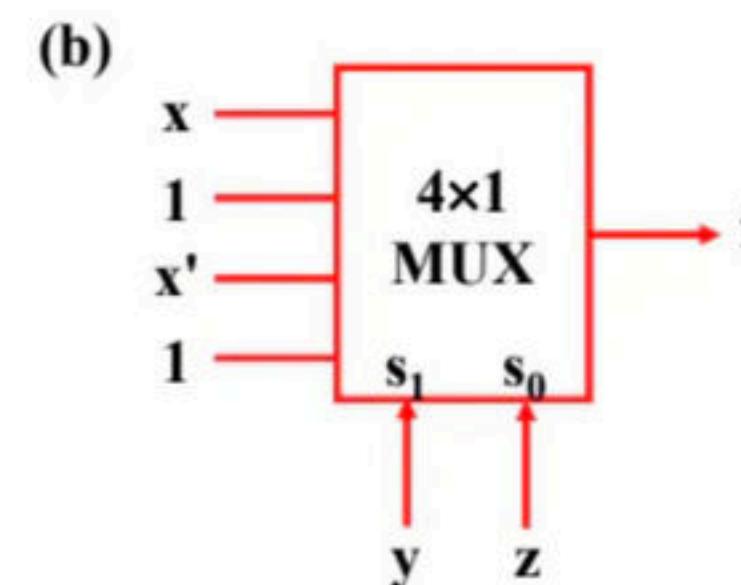
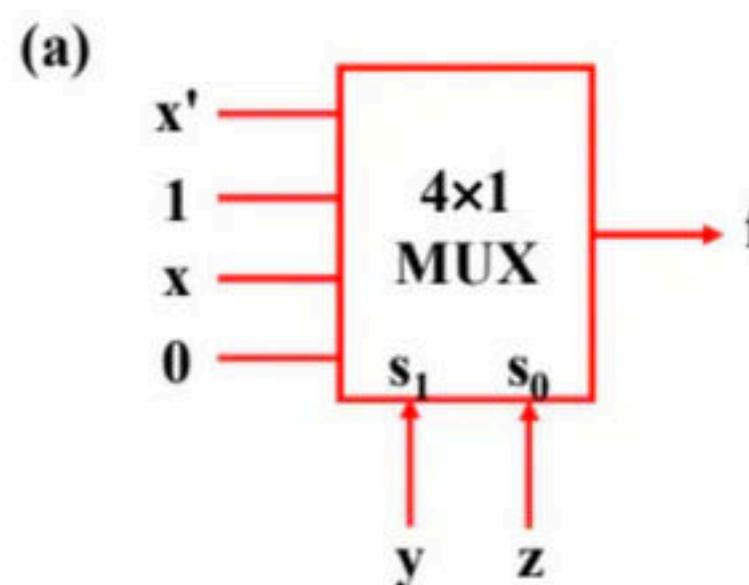
Q. The propagation delays of the XOR gate, AND gate and multiplexer (MUX) in the circuit shown in the figure are 4 ns, 2 ns and 1 ns, respectively. If all the inputs P, Q, R, S and T are applied simultaneously and held constant, the maximum propagation delay of the circuit is

- (a) 3 ns (b) 5 ns
 (c) 6 ns (d) 7 ns



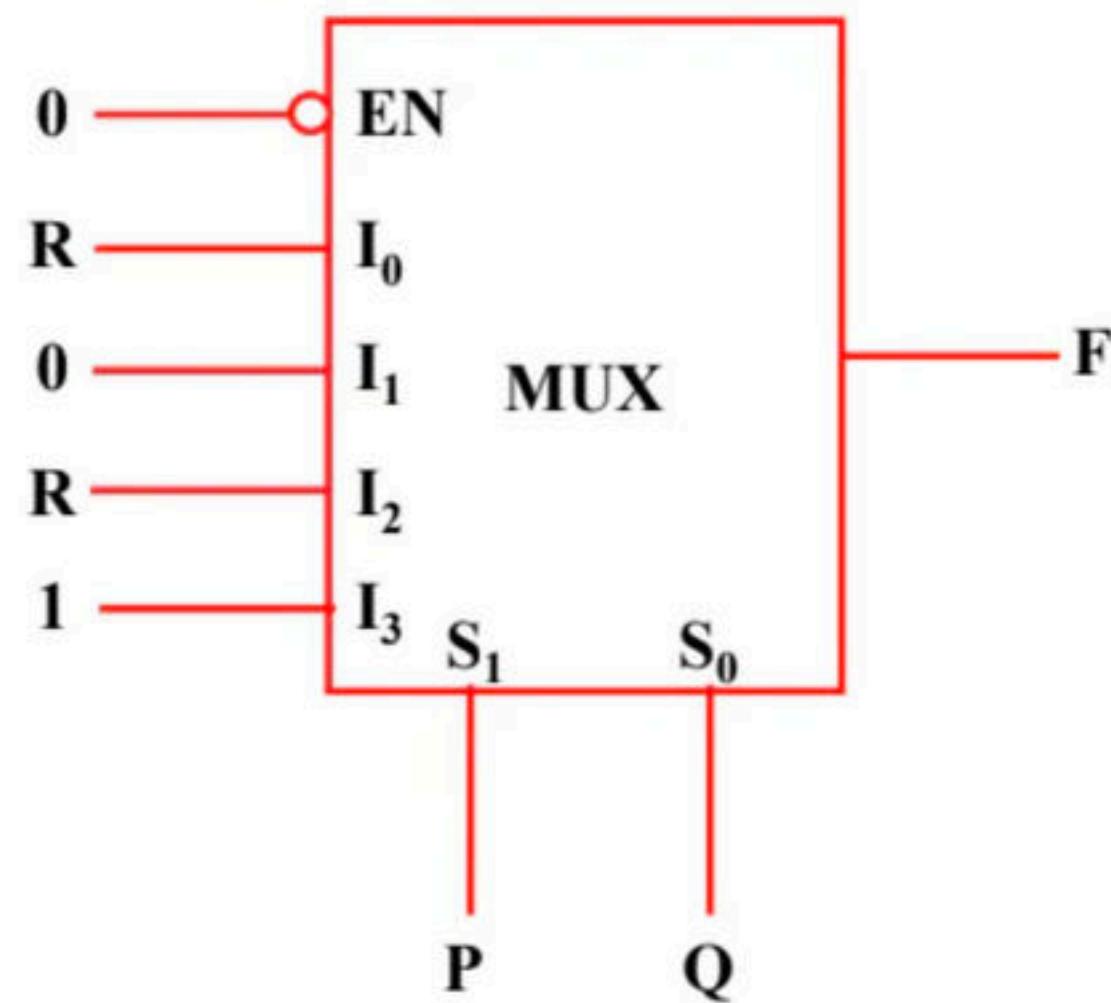
Q. Which one of the following circuits implements the Boolean function given below?

$$f(x, y, z) = m_0 + m_1 + m_3 + m_4 + m_5 + m_6, \text{ where } m_i \text{ is the } i^{\text{th}} \text{ minterm.}$$



Q. The figure below shows a multiplexer where S_1 and S_0 are the select lines. I to I_0 are the input data lines, EN is the enable line, and $F(P, Q, R)$ is the output. F is

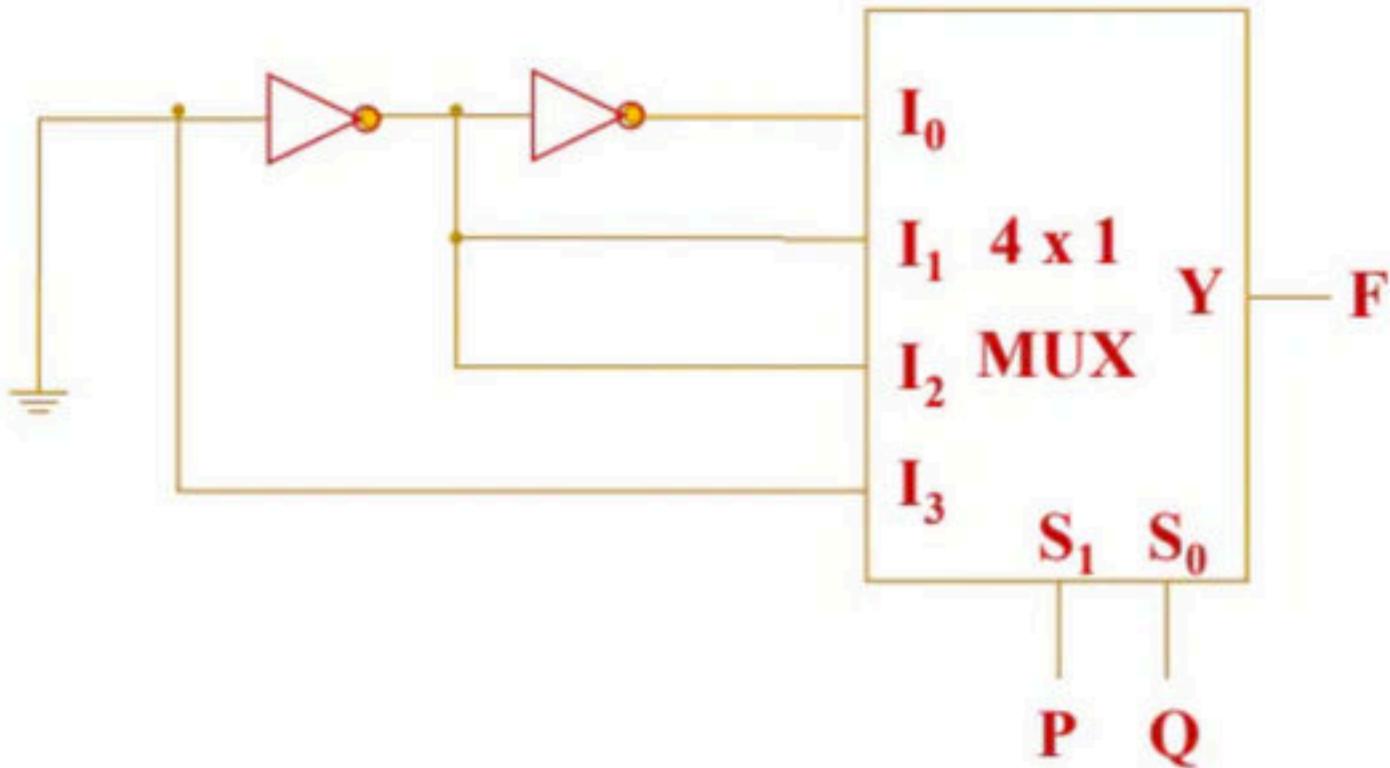
- (a) $\bar{Q} + PR$.
- (b) $P + Q\bar{R}$.
- (c) $PQ + \bar{Q}R$.
- (d) $P\bar{Q}R + \bar{P}Q$.



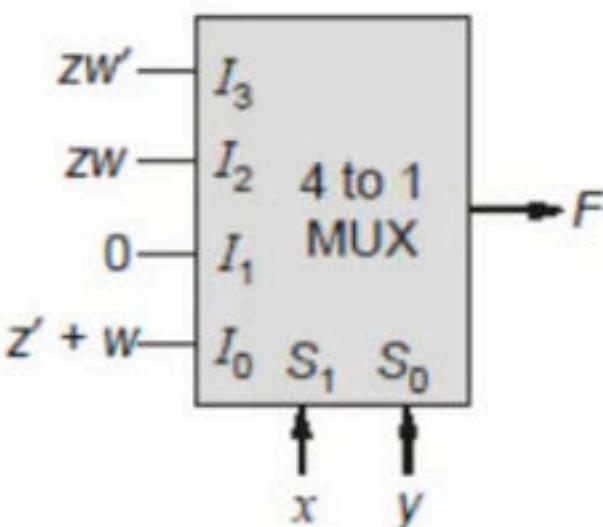
Q. The logic function implemented by the circuit below is (ground implies logic 0)

- (a) $F = \text{AND}(P, Q)$
- (c) $F = \text{XNOR}(P, Q)$

- (b) $F = \text{XOR}(P, Q)$
- (d) $F = \text{OR}(P, Q)$



A 4×1 multiplexer with two selector lines is used to realize a Boolean function, F having four Boolean variables X, Y, Z and W as shown below. S_0 and S_1 denote the least significant bit (LSB) and most significant bit (MSB) of the selector lines of the multiplexer respectively. I_0, I_1, I_2, I_3 are the input lines of the multiplexer.



The canonical sum of product representation of F is

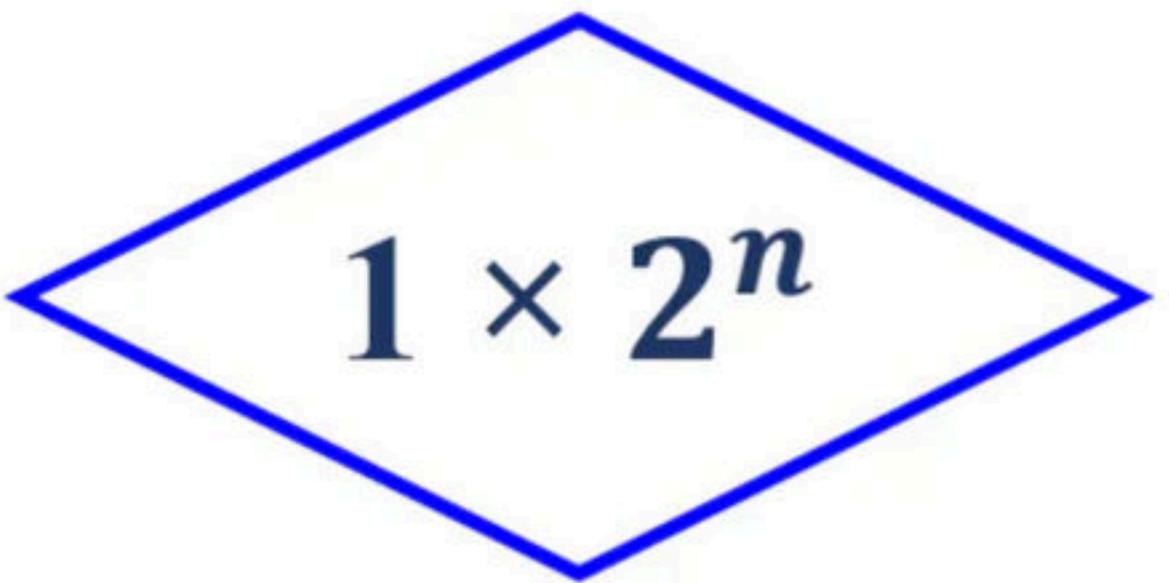
- (a) $F(X, Y, Z, W) = \Sigma m(0, 1, 3, 14, 15)$
- (b) $F(X, Y, Z, W) = \Sigma m(0, 1, 3, 11, 14)$
- (c) $F(X, Y, Z, W) = \Sigma m(2, 5, 9, 11, 14)$
- (d) $F(X, Y, Z, W) = \Sigma m(1, 3, 7, 9, 15)$

Demultiplexer

A demultiplexer is a circuit that receives information on a single line and transmits to one of the 2^n possible output lines , according to the selection lines .

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

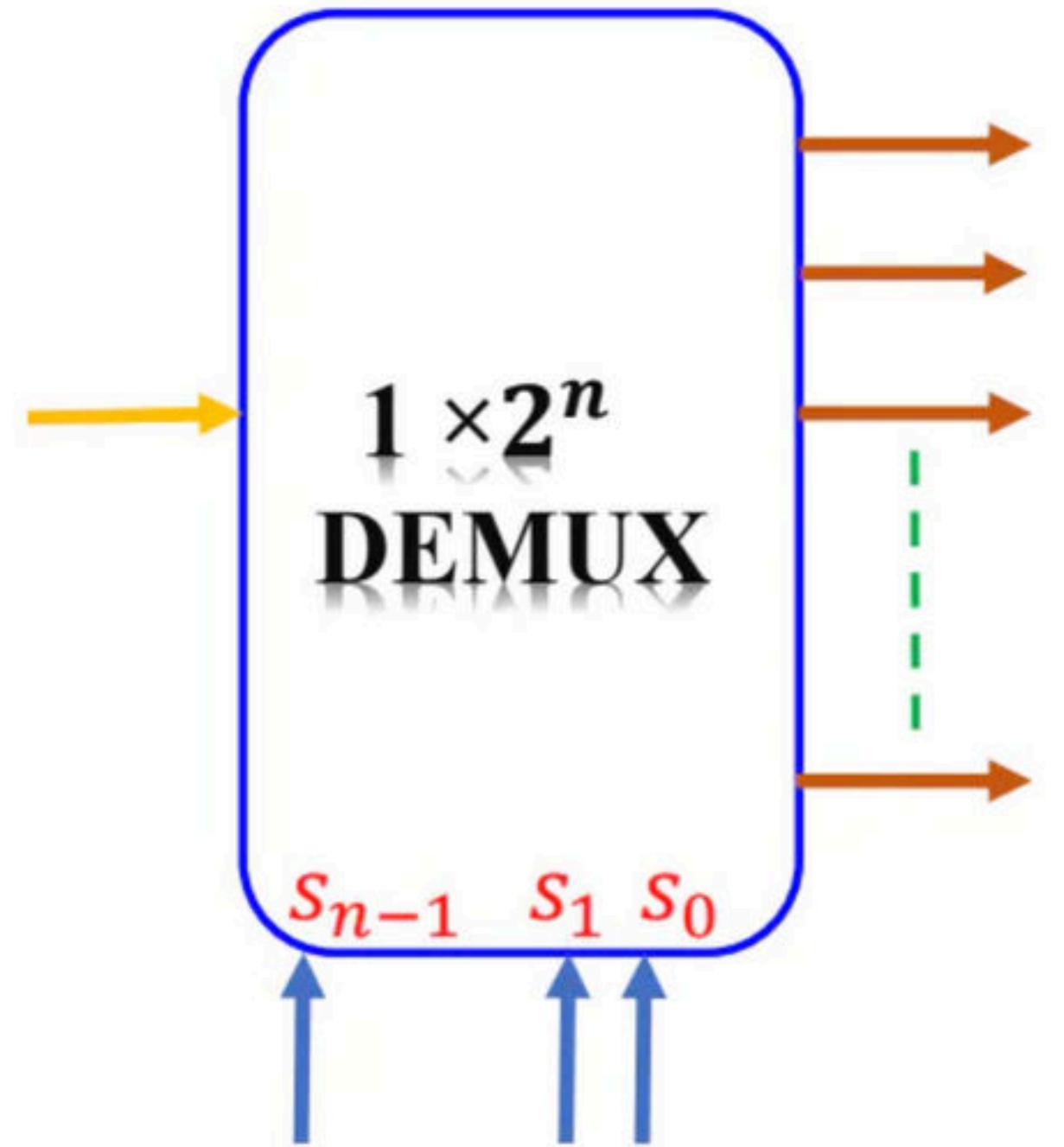
General structure



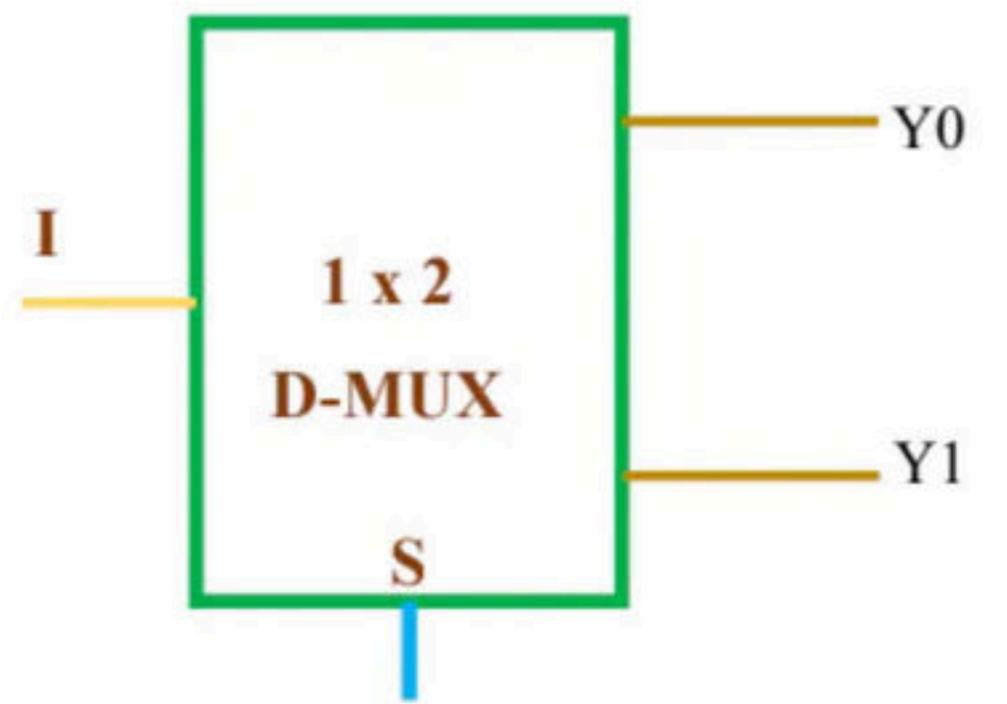
n -----> number of select lines

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

1 -----> number of inputs



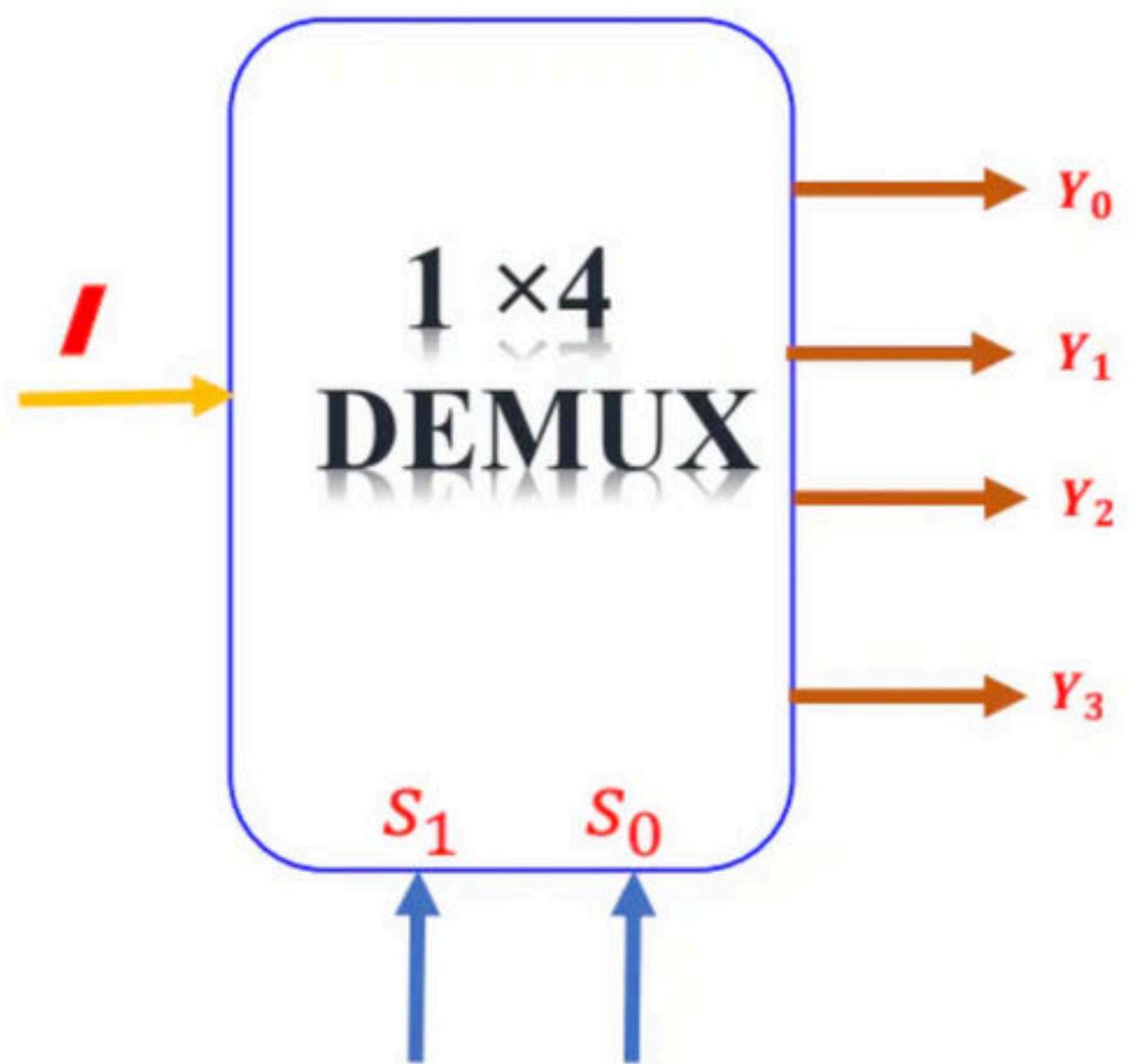
1×2 DEMUX



S	Y0	Y1

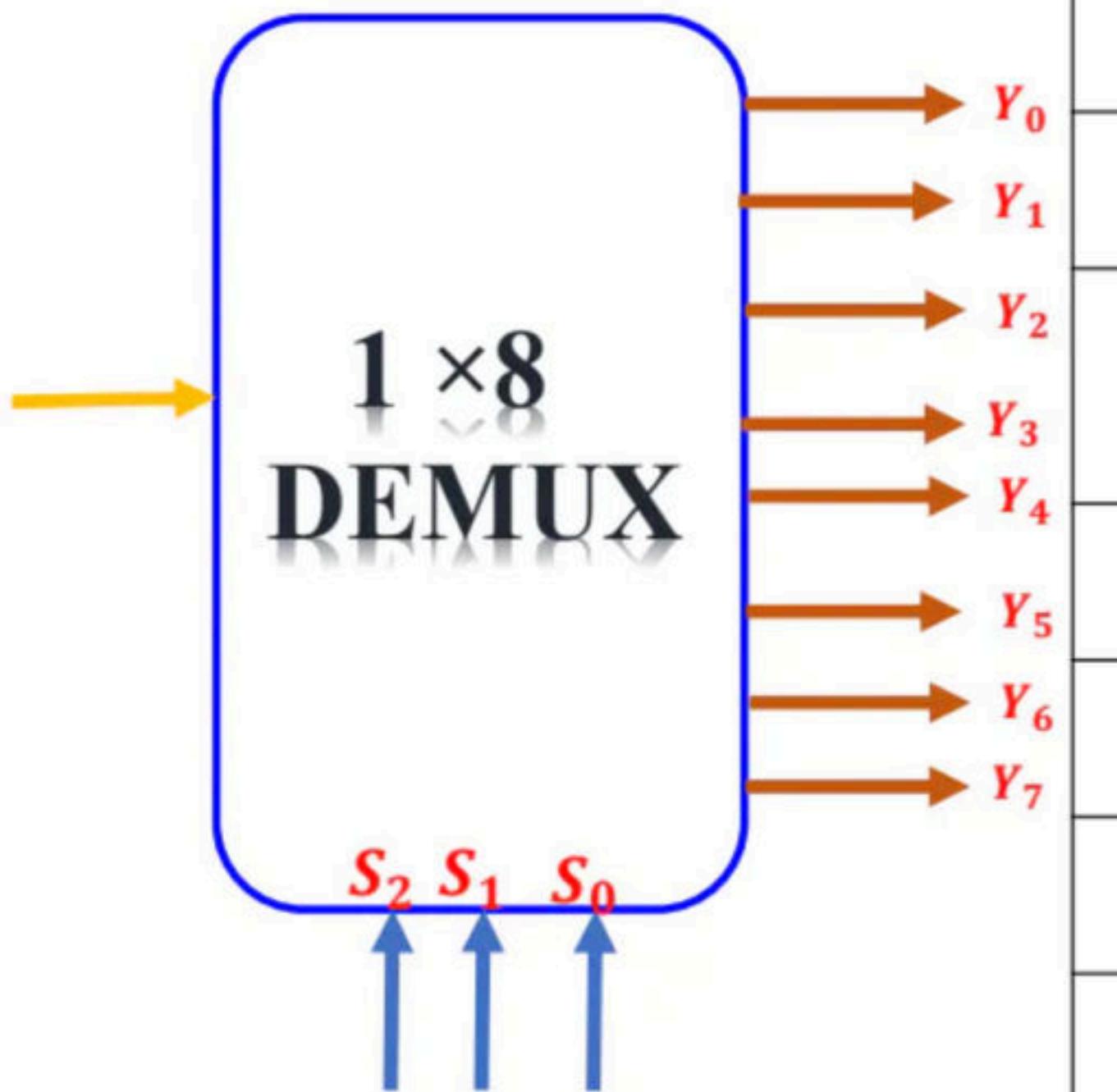
Logic circuit

1×4 Demultiplexer



S_1	S_0	Y_3	Y_2	Y_1	Y_0

1 ×8 DEMUX

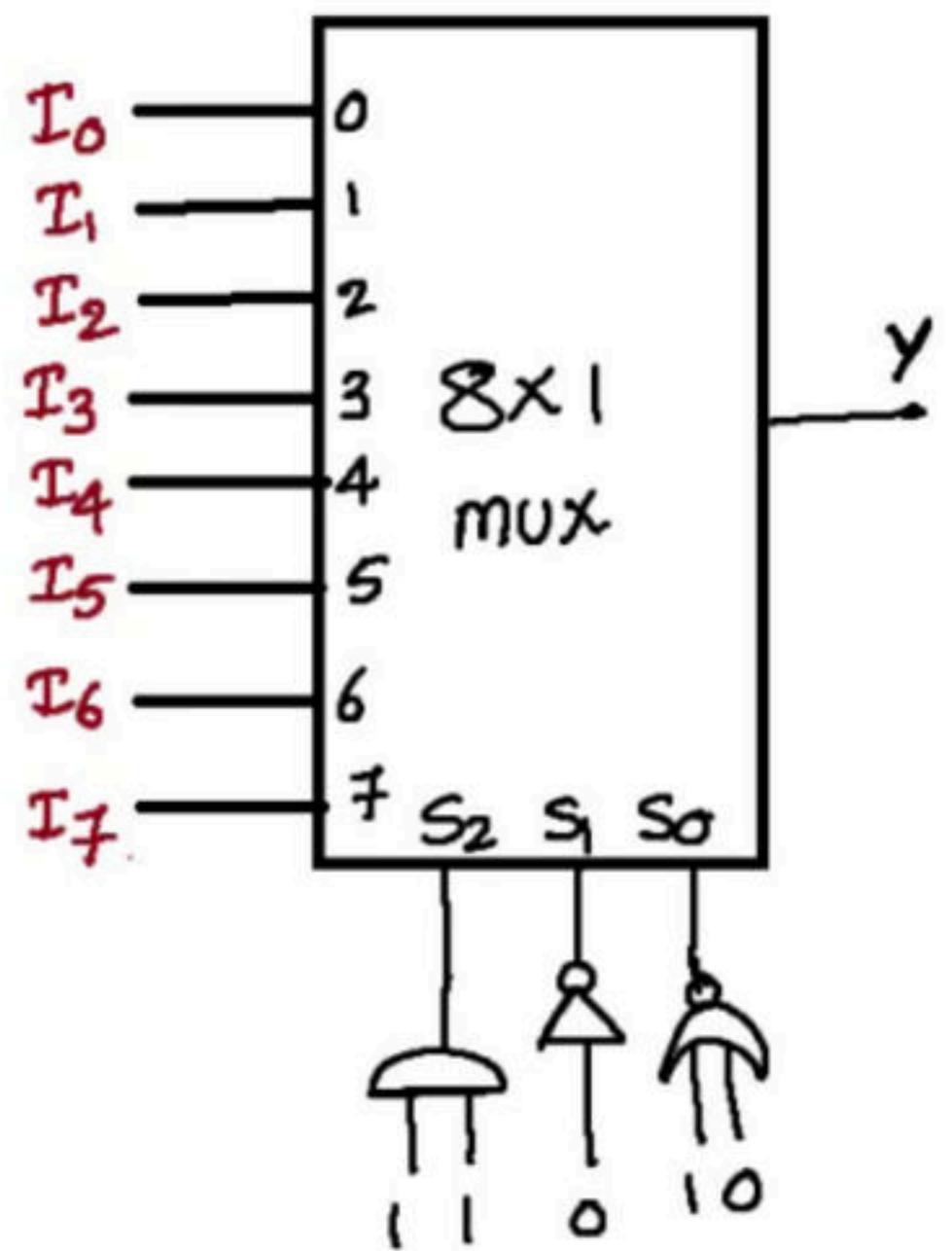


Q) Implement HA using 1×4 DEMUX

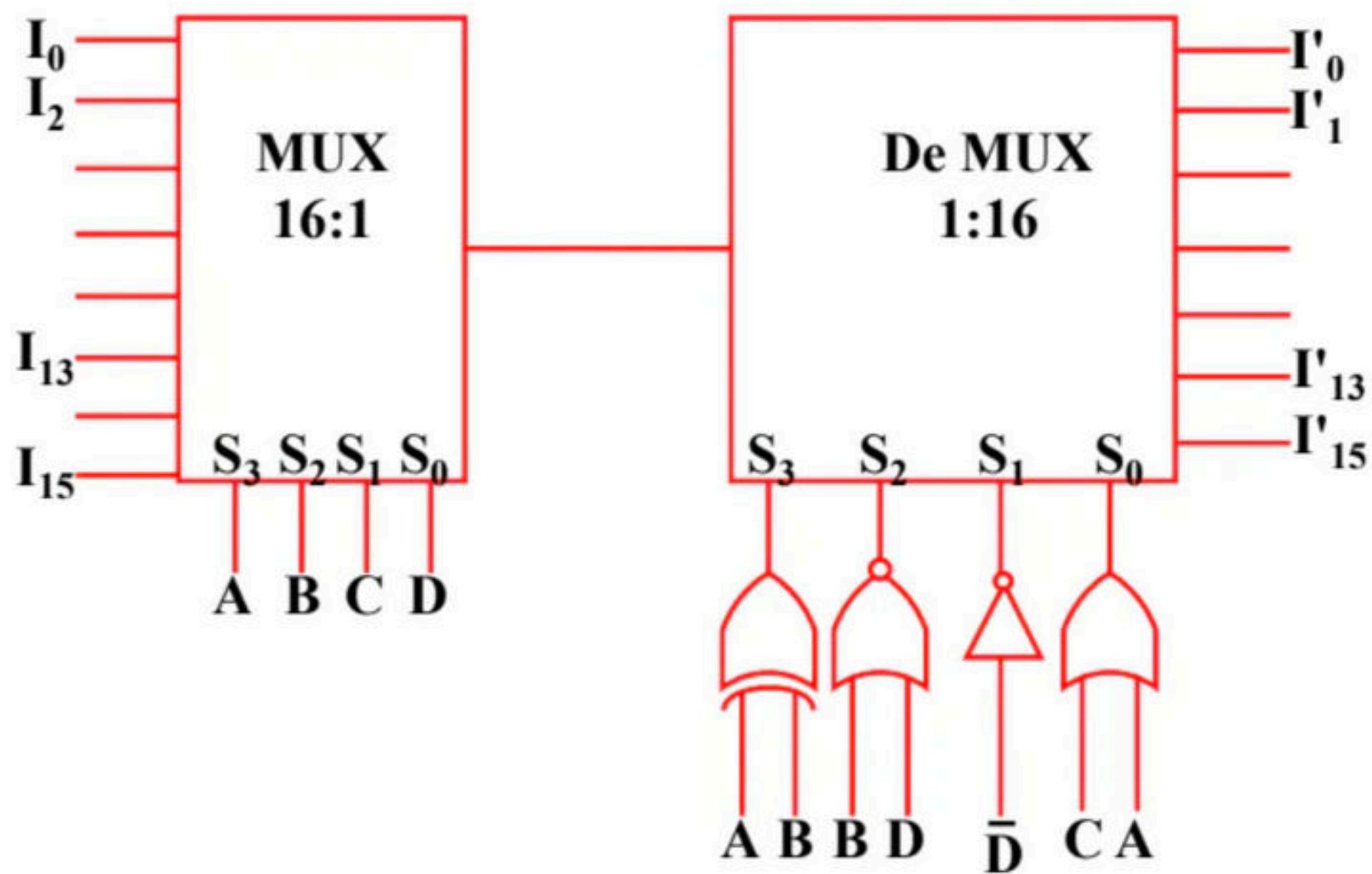
Q) Implement HS using 1×4 DEMUX

Q) Implement FA using 1×8 DEMUX

Q) The output of the mux (Y) is



Q. Consider the logical circuit given below , Input at line I_{13} in 16×1 MUX corresponds to output at line I'_n of 1×16 De-MUX. The value of ‘n’ is _____.



Implementation of higher order Demux using lower order
Demux

Q) Implement 1×4 Demux using 1×2 Demux

Q) Implement 1×16 Demux using 1×2 Demux

Q) Implement 1×8 Demux using 1×4 Demux

Decoder

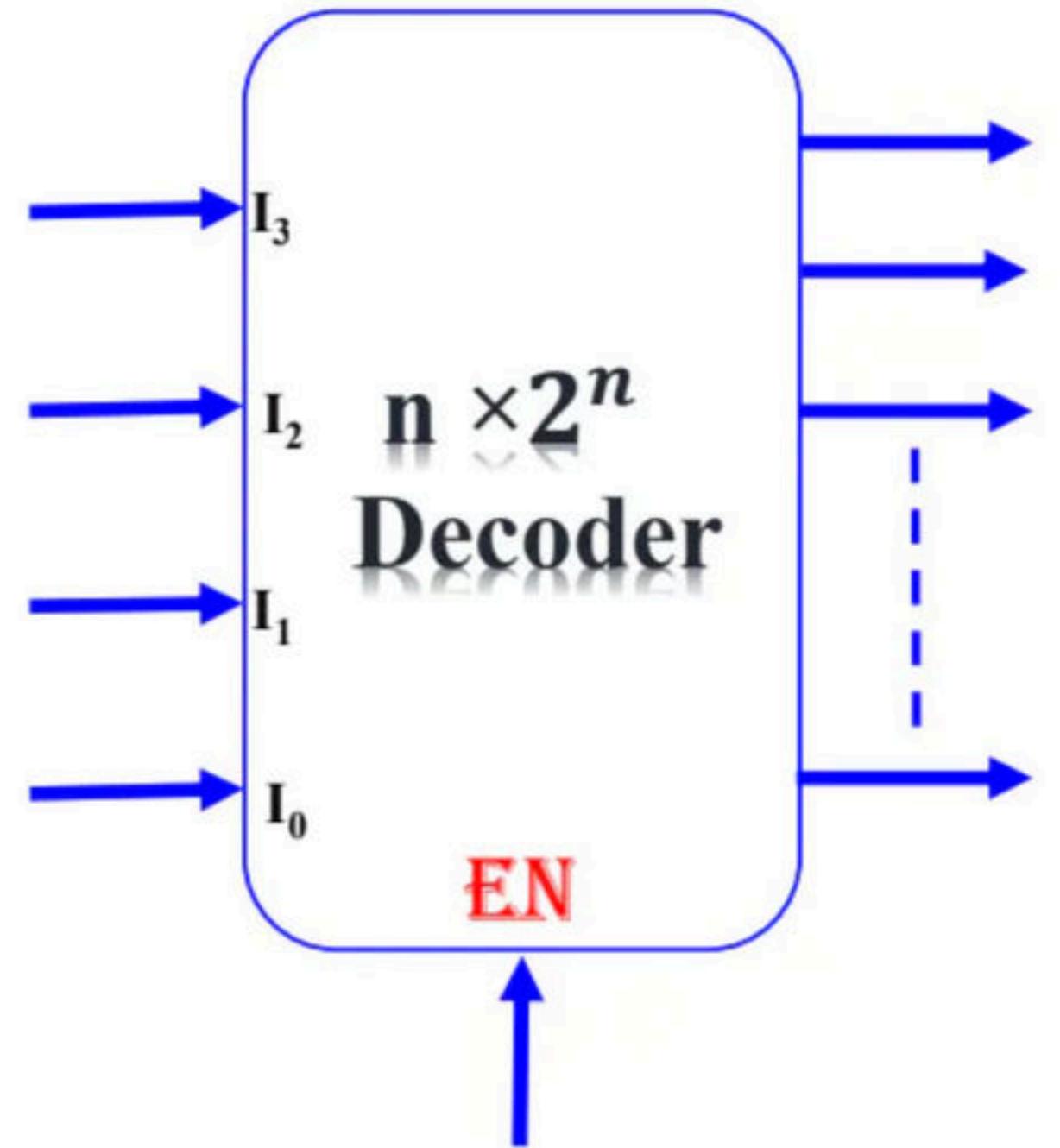
Decoder is a multi input ,multi output logic circuit which converts coded input into coded output , where the input and output codes are different .

General structure

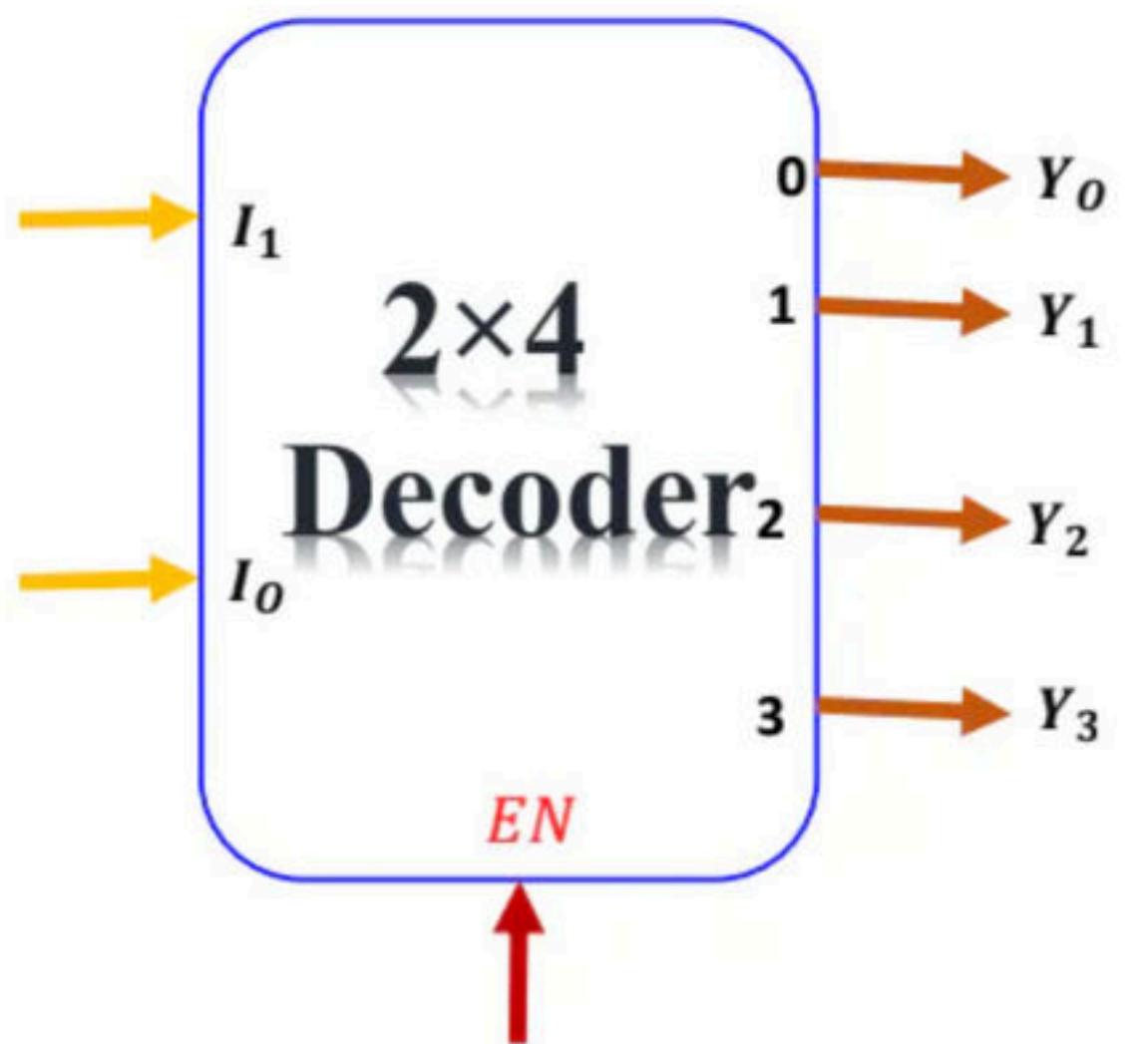
$$n \times 2^n$$

n -----> number of inputs

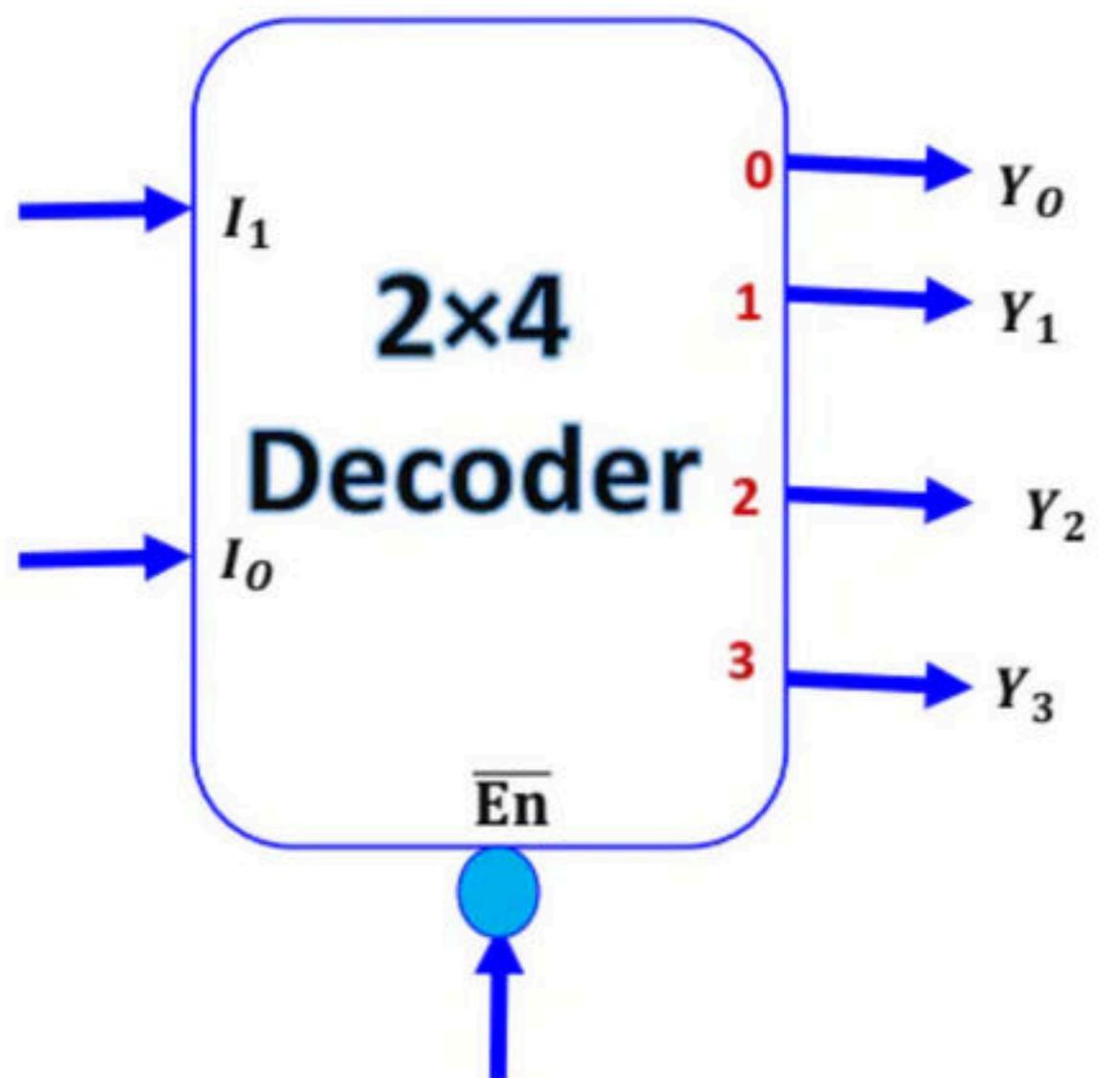
2^n -----> number of outputs



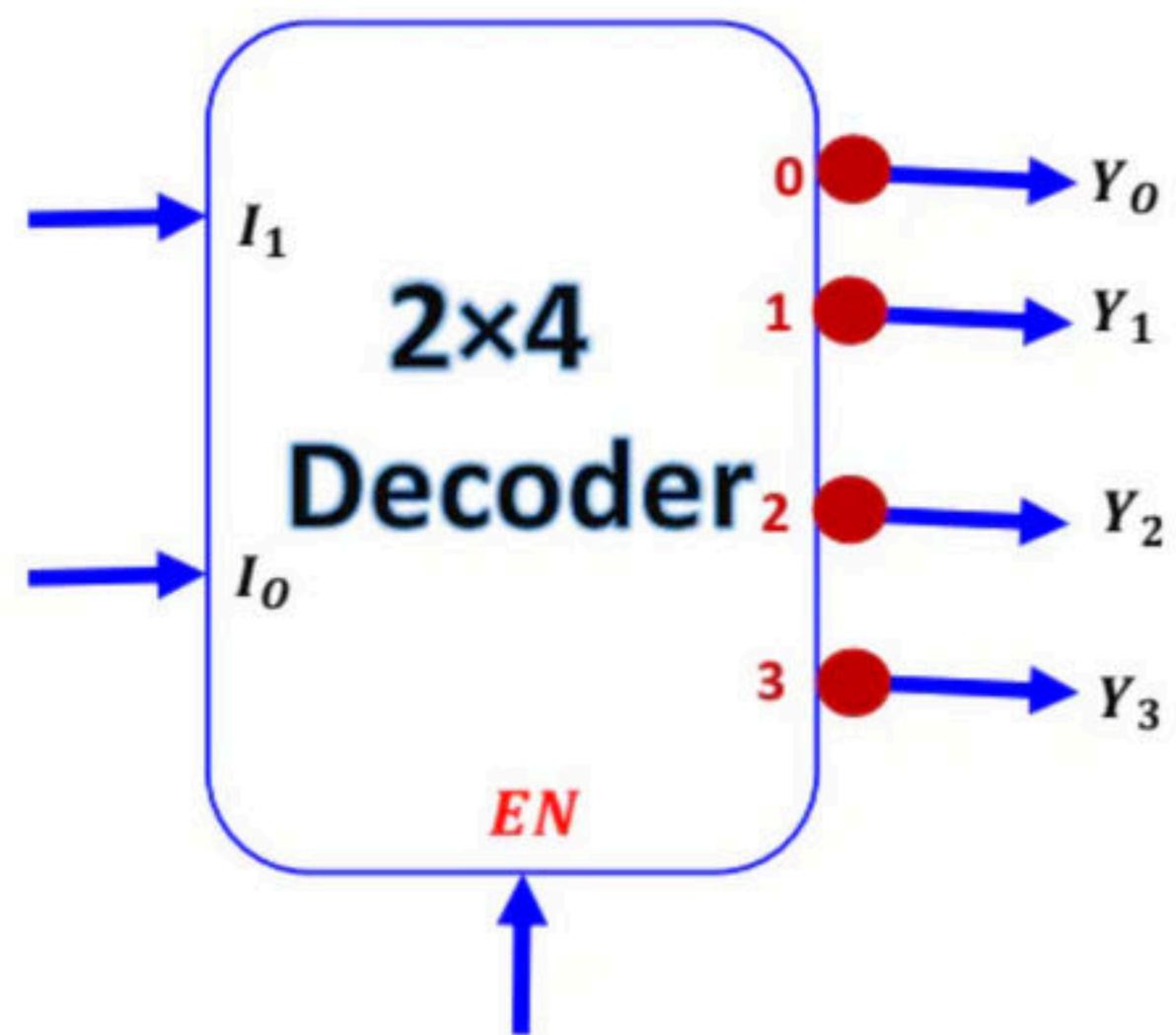
Active High Decoder



Active High Decoder

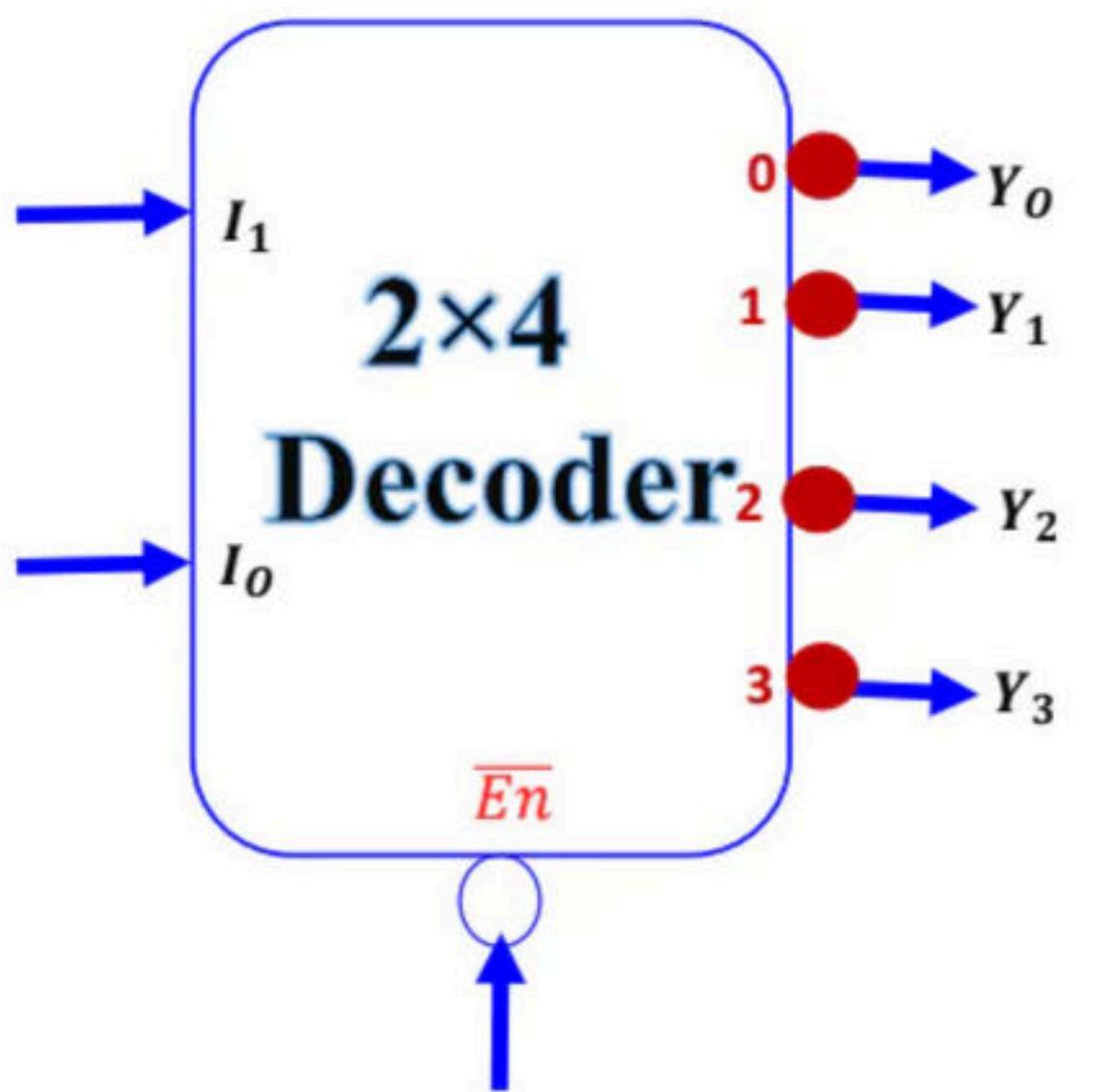


Active Low Decoder

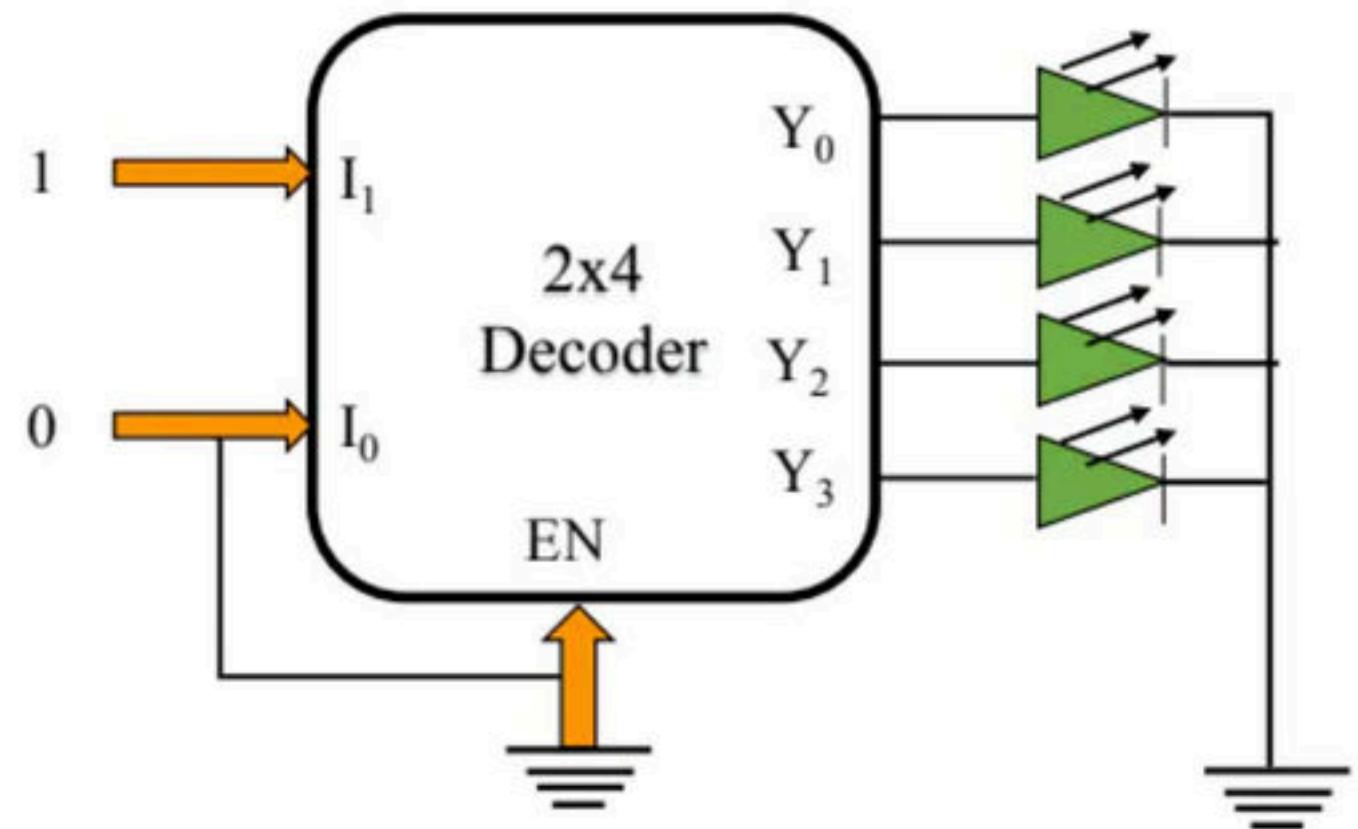


En	A	B	Y_3	Y_2	Y_1	Y_0

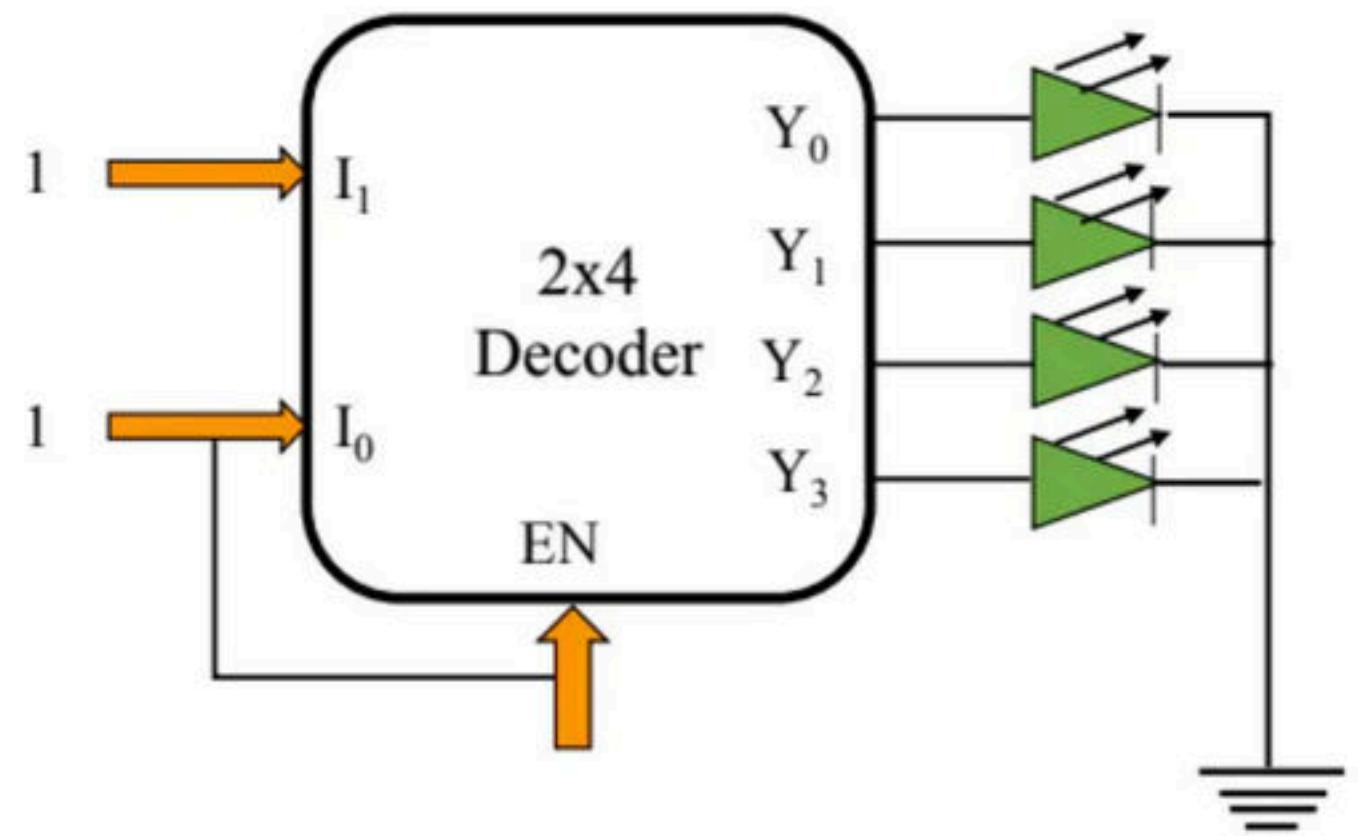
Active Low Decoder



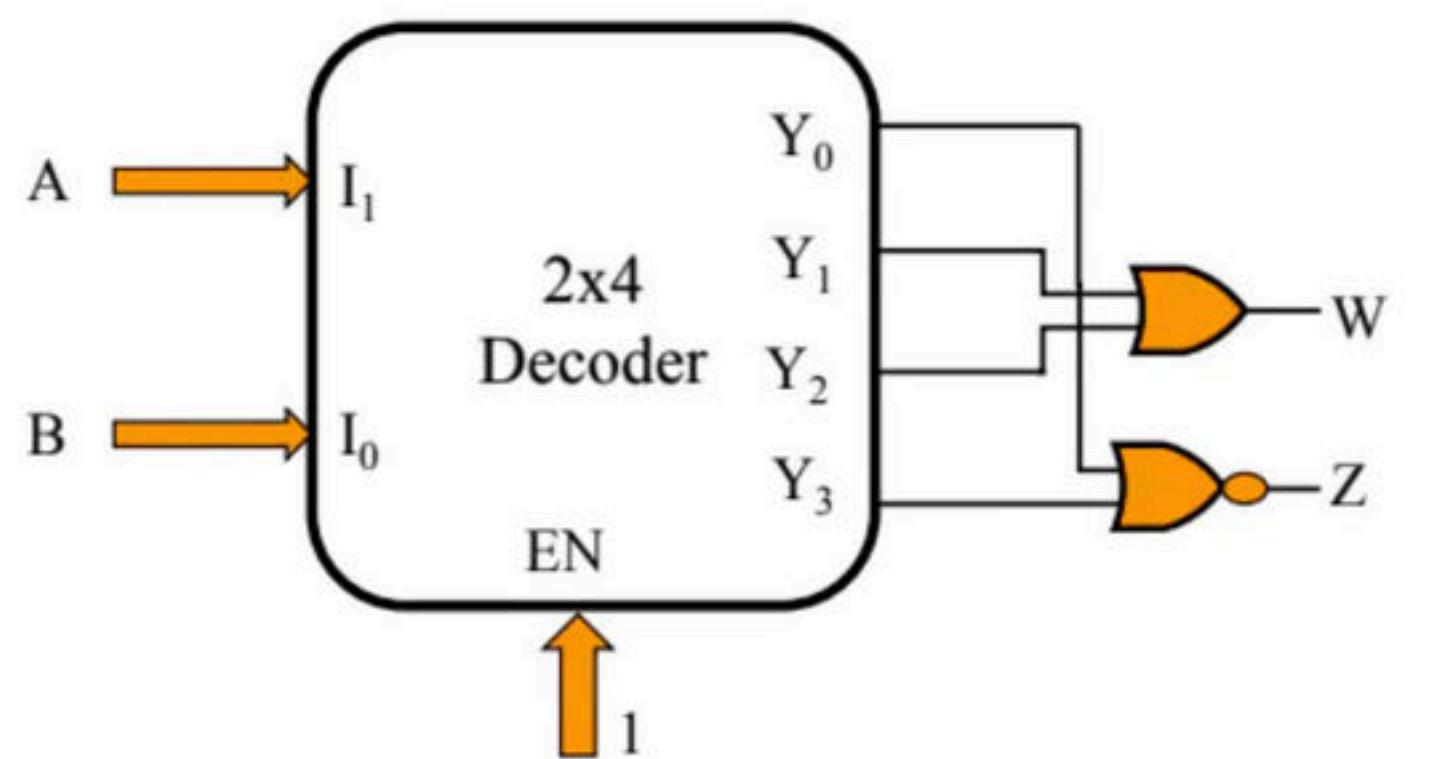
Q) Which of the following LED will glows



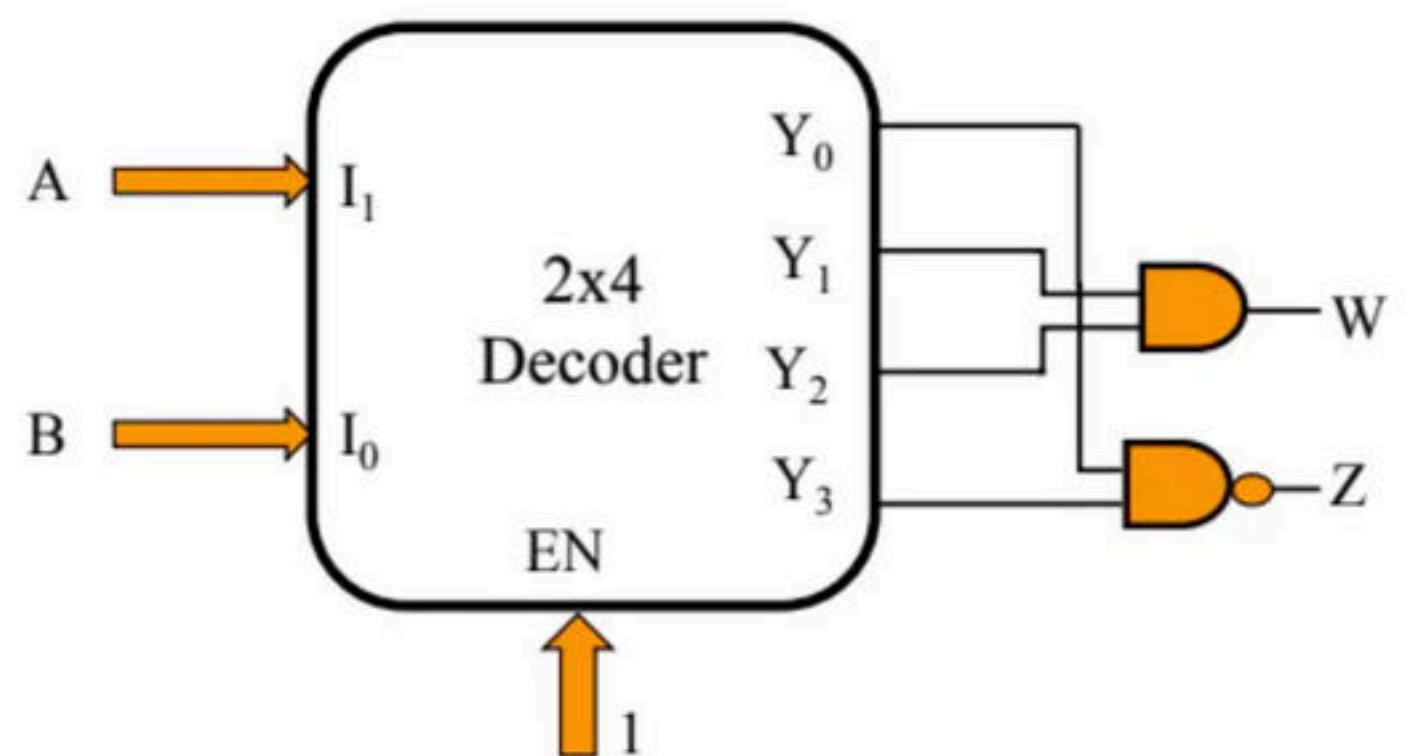
Q) Which of the following LED will glows



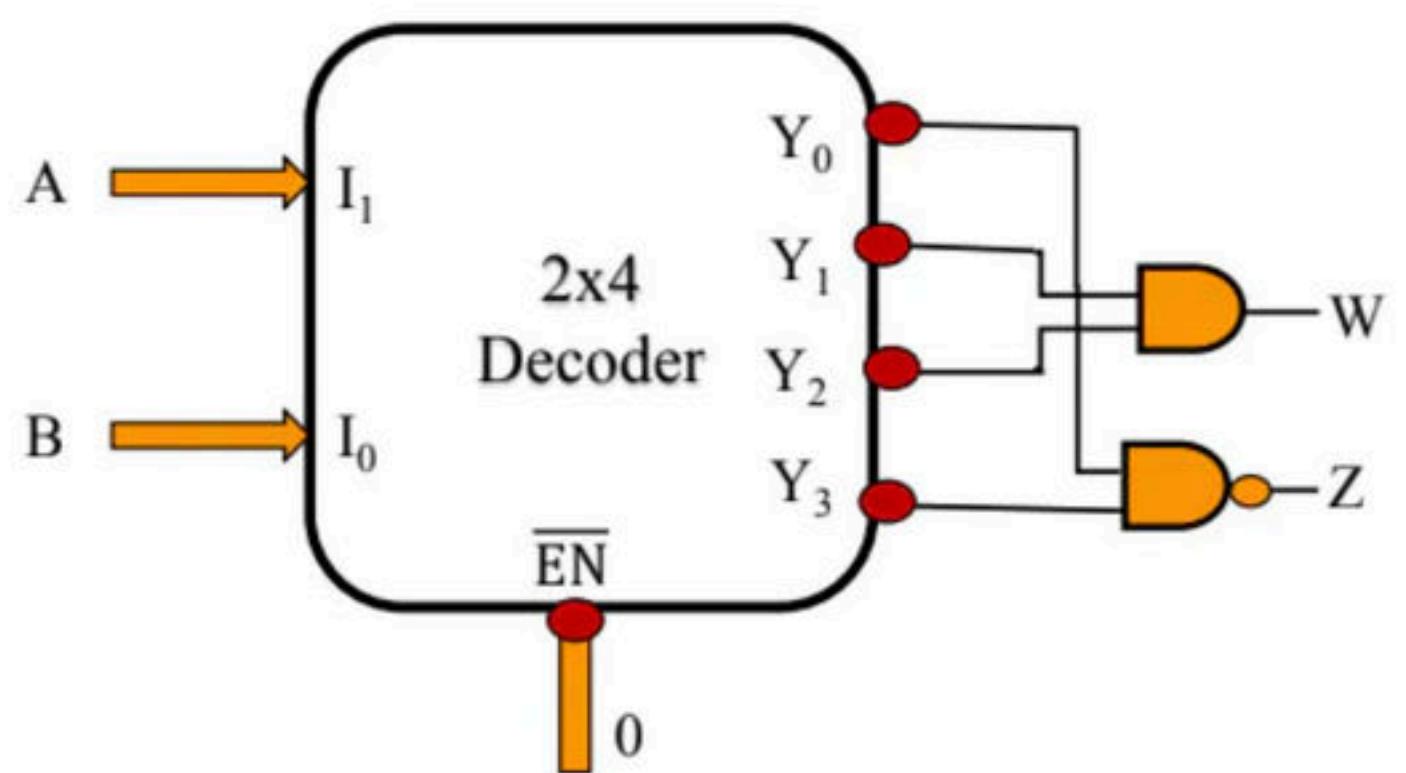
Q) Find the logic expression of W and Z



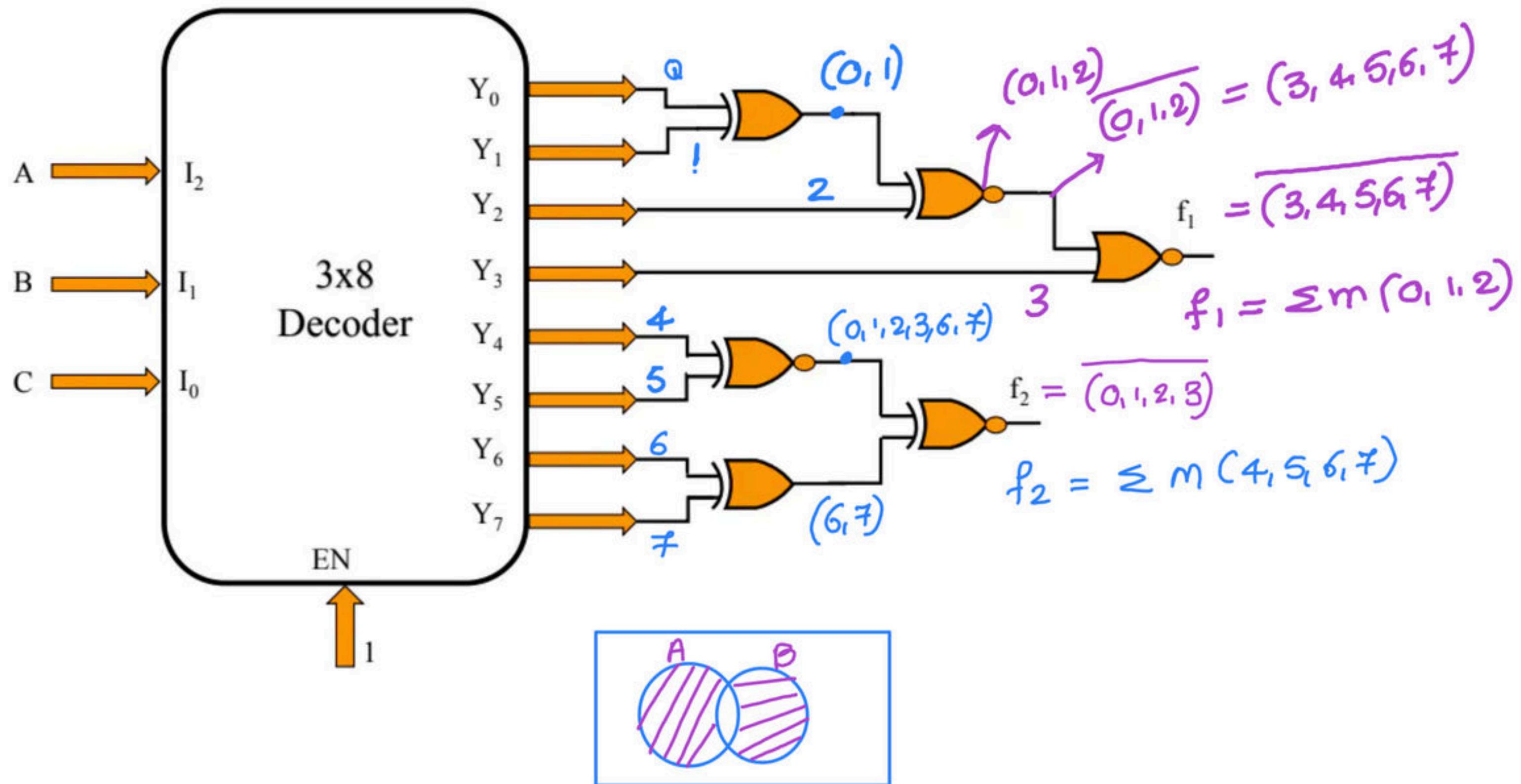
Q) Find the logic expression of W and Z



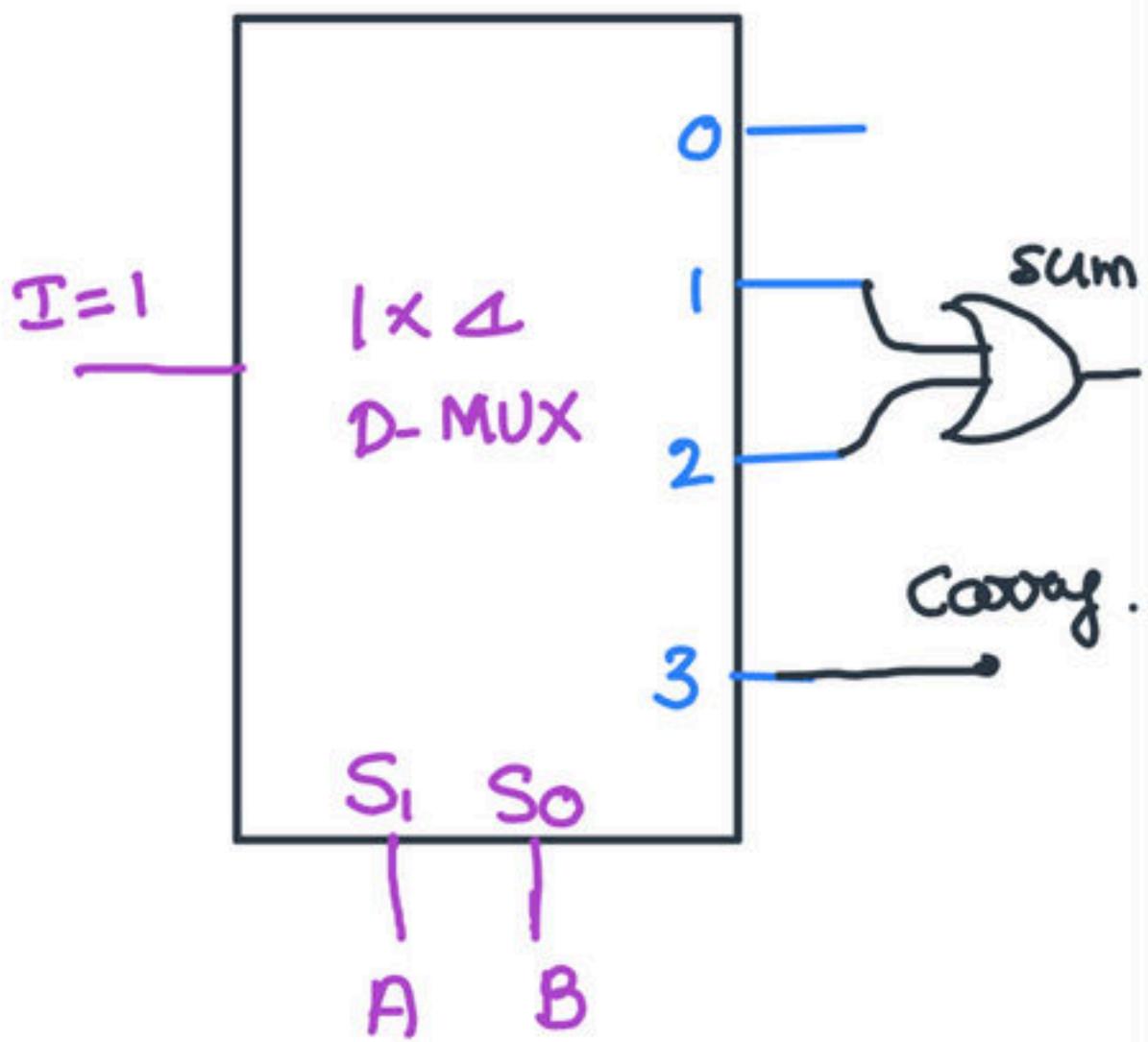
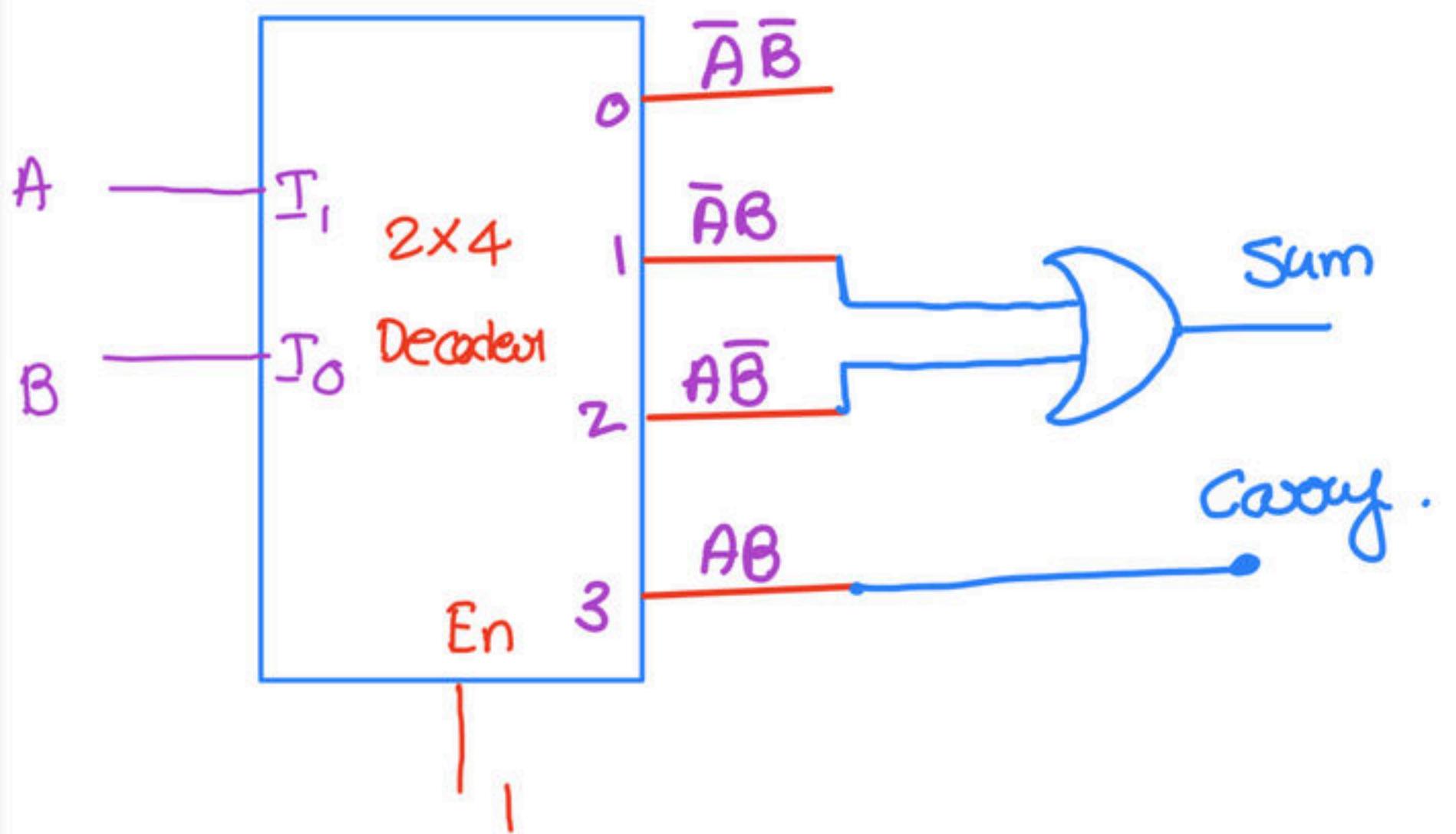
Q) Find the logic expression of W and Z



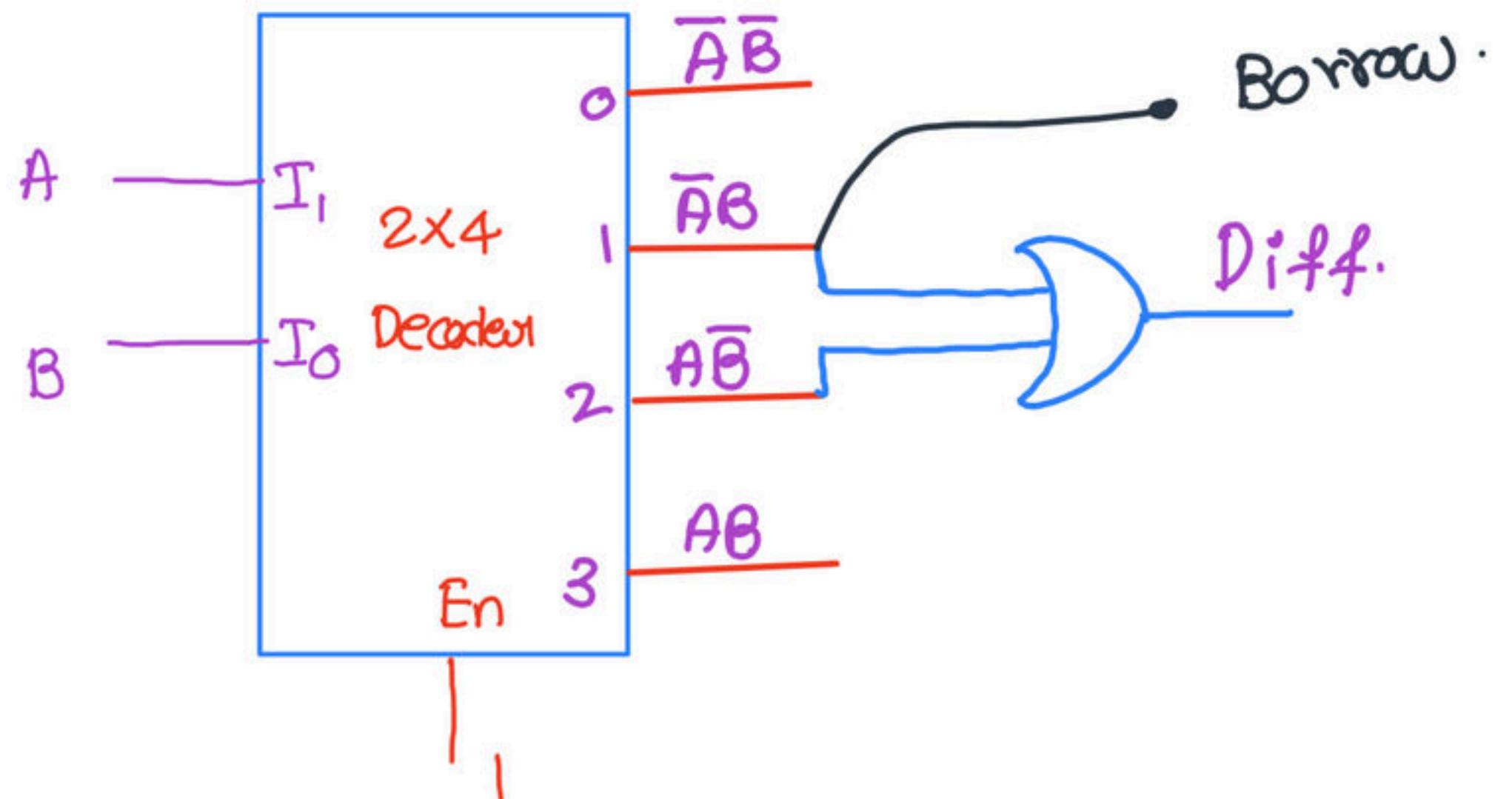
Q) The logic expression of F1 and F2



Q) Implement HA using 2×4 decoder



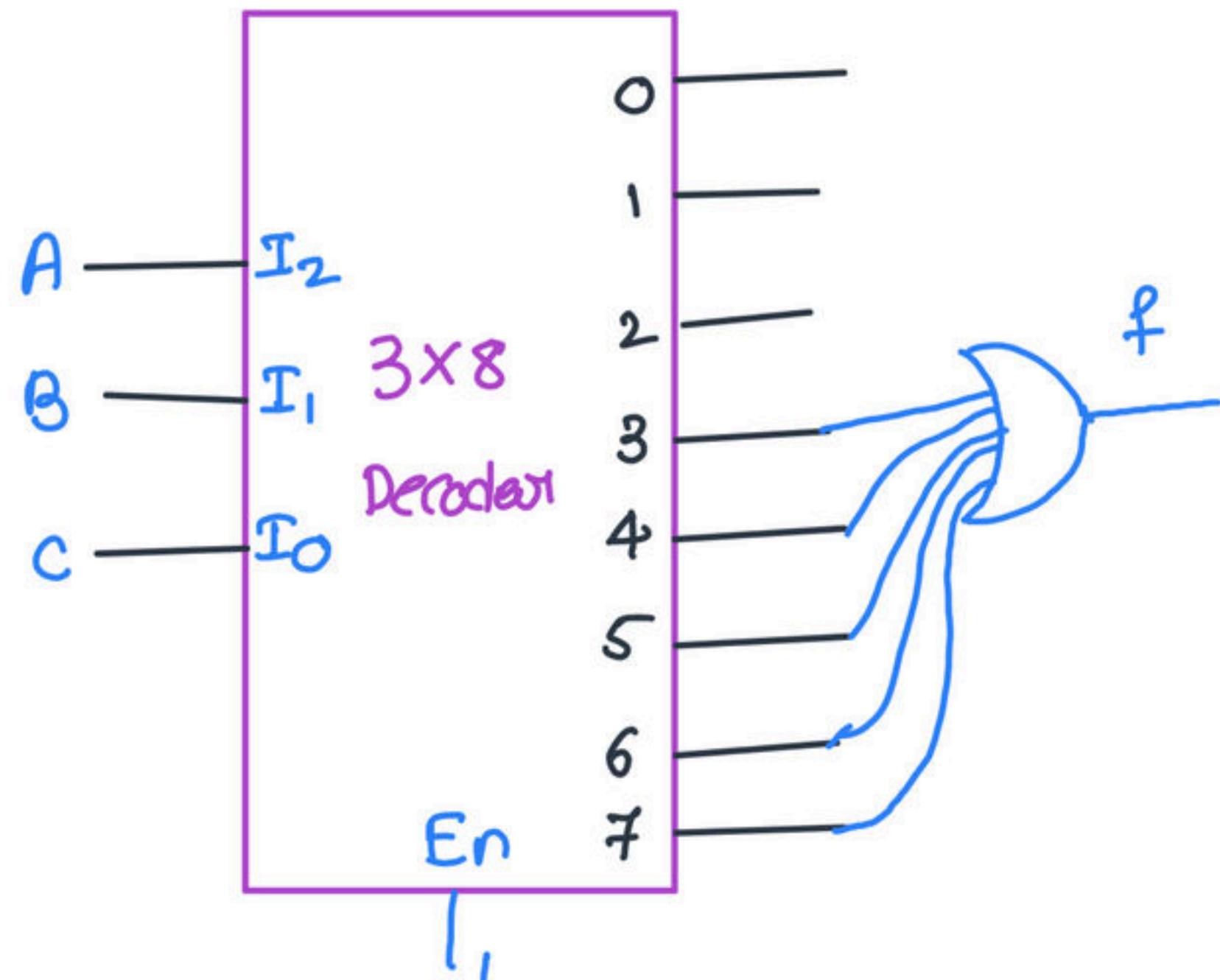
Q) Implement HS using 2×4 decoder



Q) implement $F(A, B, C) = A + BC$, using decoder

100 011
101 111.
110
111

$$f = \sum m(3, 4, 5, 6, 7)$$



Conversation of
Demultiplexer <-----> Decoder



Inputs \longleftrightarrow Enable

Select lines \longleftrightarrow Inputs

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

Implementation of higher order Decoders using lower order Decoders

Q) Implement $4 \times \underline{16}$ decoder using 2×4 decoder

$$\frac{16}{4} = 4 \rightarrow L_2(C, D) \quad f(A, B, C, D)$$

$$\begin{array}{r} \frac{4}{4} = 1 \\ \hline 5 \end{array} \rightarrow L_1(\text{MSB})(A, B)$$

Q) Implement 3×8 decoder using 2×4 decoder

$$\frac{8}{4} = 2 \quad (\text{2x4 decoder}) \rightarrow L_2 \quad (\text{B,C})$$

$$\frac{2}{4} = 1 \quad (\text{1x2 decoder}) \rightarrow L_1 \quad (\text{MSB})(A)$$

3

$f(A_1 B, C)$

Q) Implement 4×16 decoder using 3×8 decoder

$$\frac{16}{8} = 2 \quad (\text{3x8 decoder}) \rightarrow (\text{B, C, D})$$

$$\frac{2}{8} = 1. \quad (\text{1x2 decoder}) \rightarrow (\text{msb})(A)$$

$$\underline{\quad \quad \quad 3}$$

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)

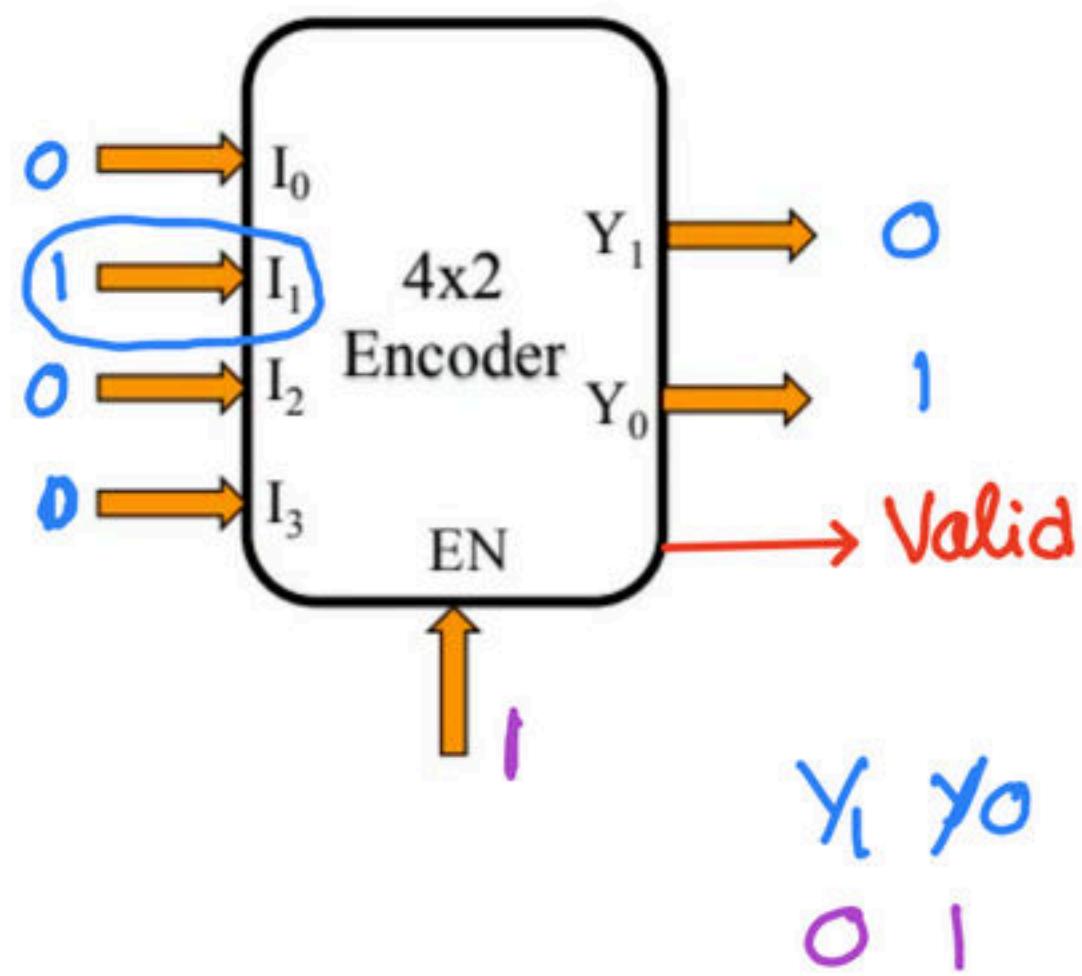
General structure

$2^n \times n$

n -----> number of outputs

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

4 X 2 Encoder

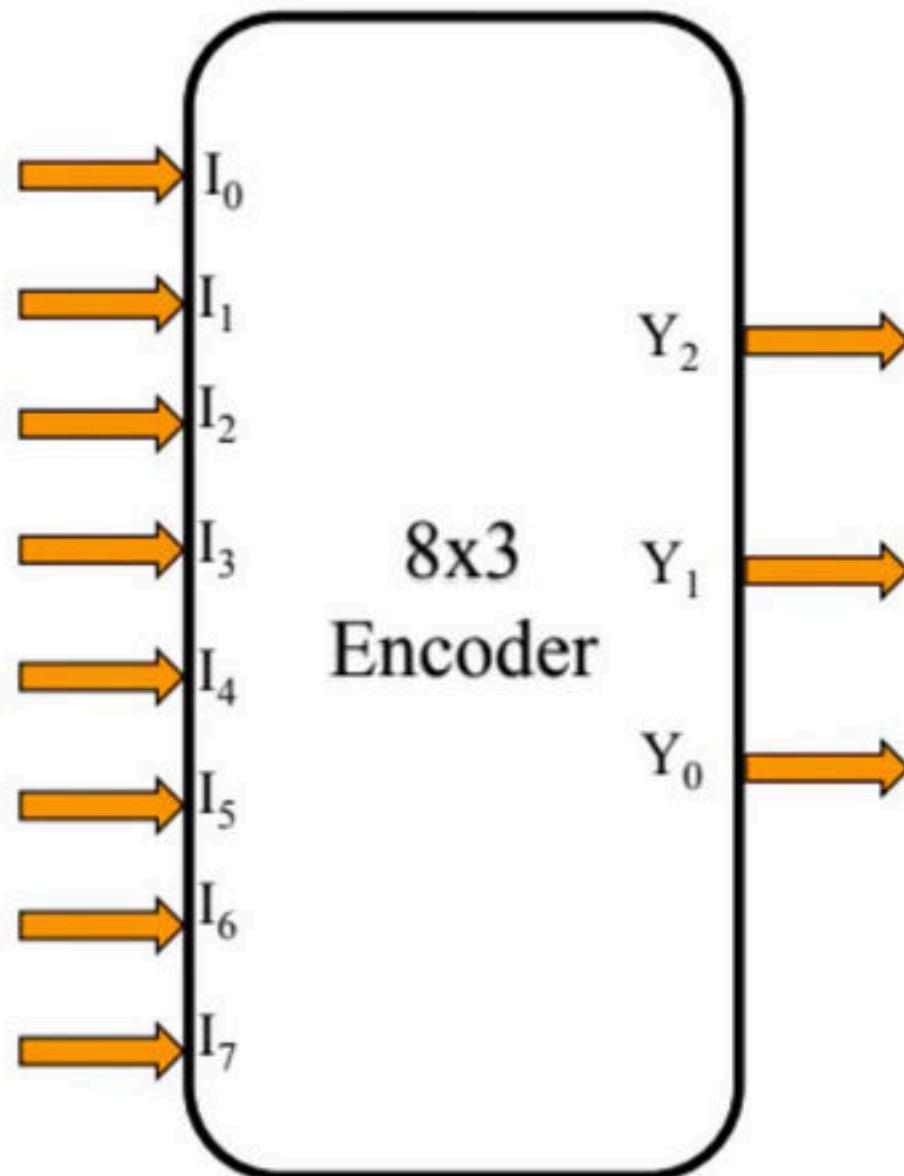


I_3	I_2	I_1	I_0	Y_1	Y_0	Valid
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	1	0	0	1	0	1
1	0	0	0	1	1	1
0	0	0	0	X	X	0

Drawbacks of Encoder

- For an Encoder at a time only one among the all inputs is high , remaining 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.

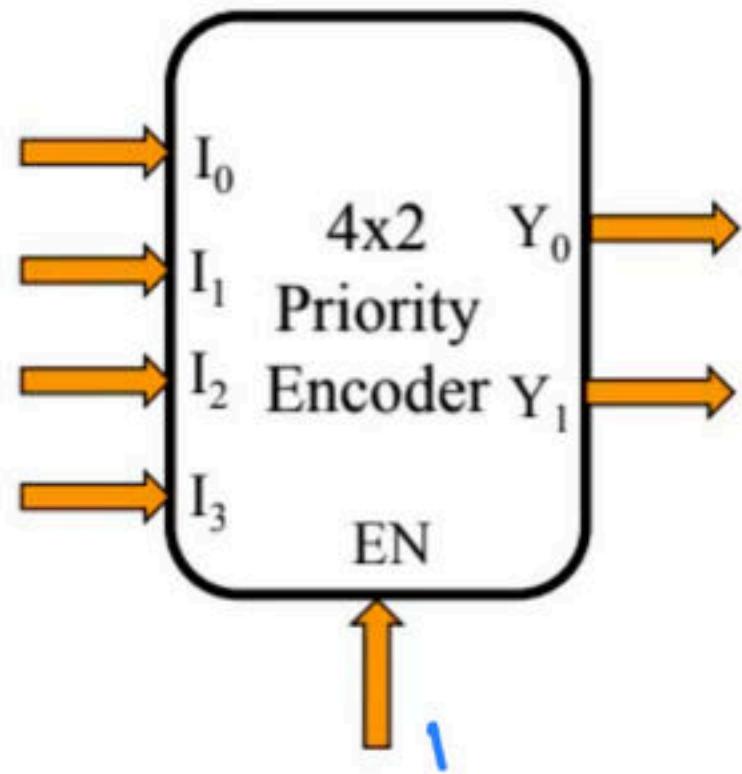
8 X 3 Encoder



Priority Encoder

Priority encoder assign priority to every input and whenever higher priority input is one , then other inputs are not consider

Priority Encoder



I3	I2	I1	I0	Y1	Y0	Valid
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

$I_3 > \underline{I_2} > I_1 > I_0$

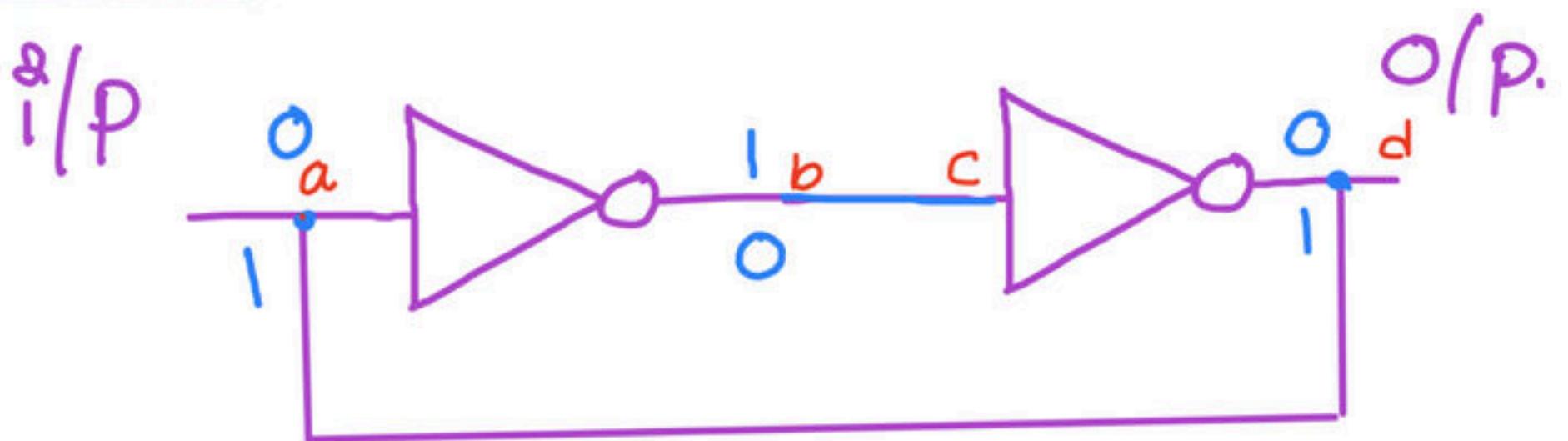
SEQUENTIAL CIRCUITS

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

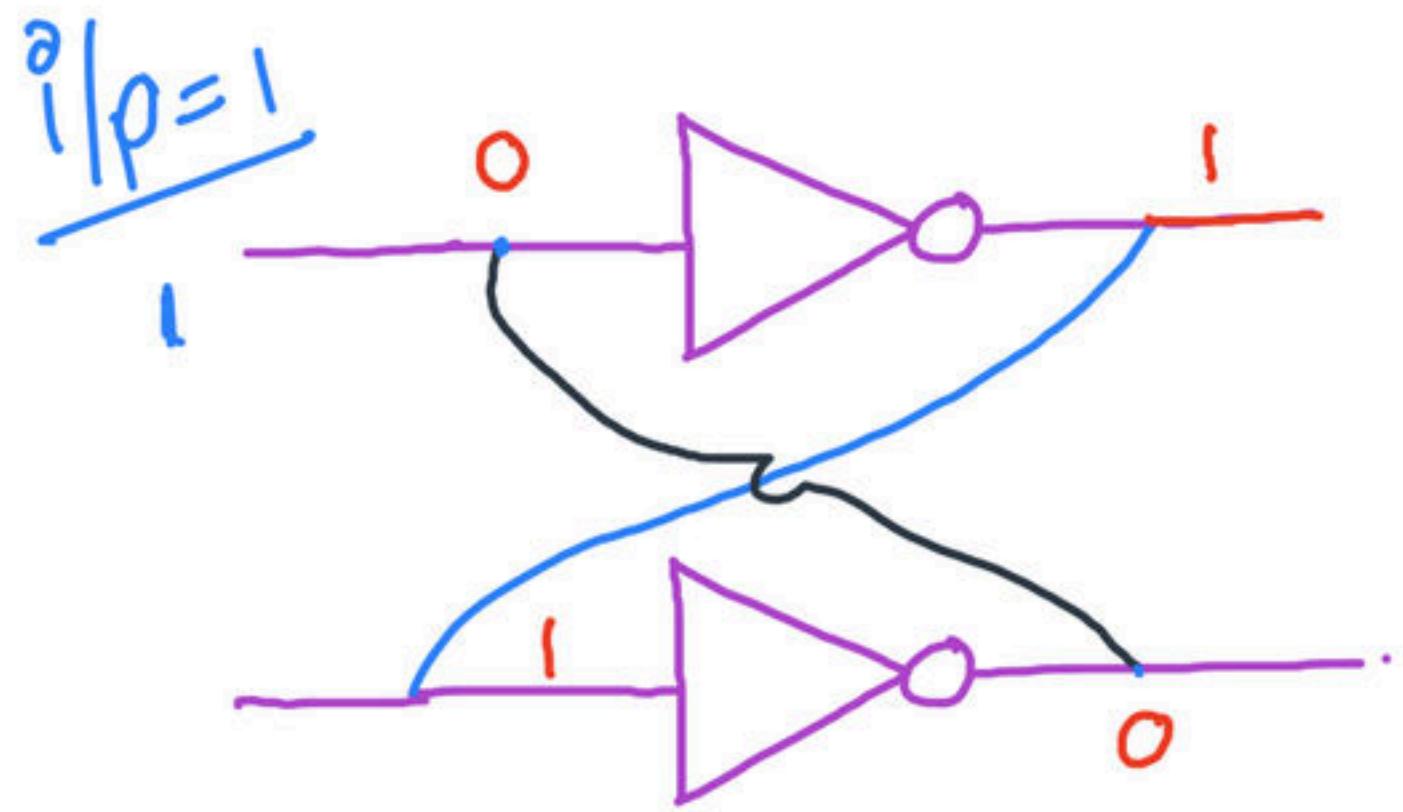
- Latch ✓
- Flip flops ✓
- Shift Register ✓
- Counters ✓
- Finite State Machines ✓

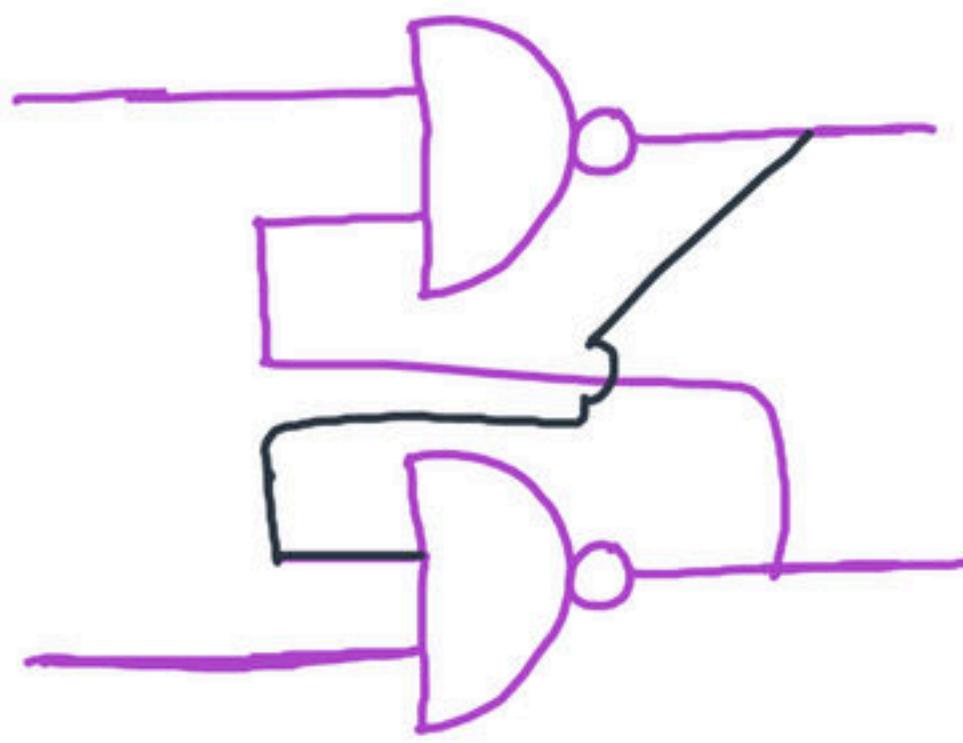
Latch



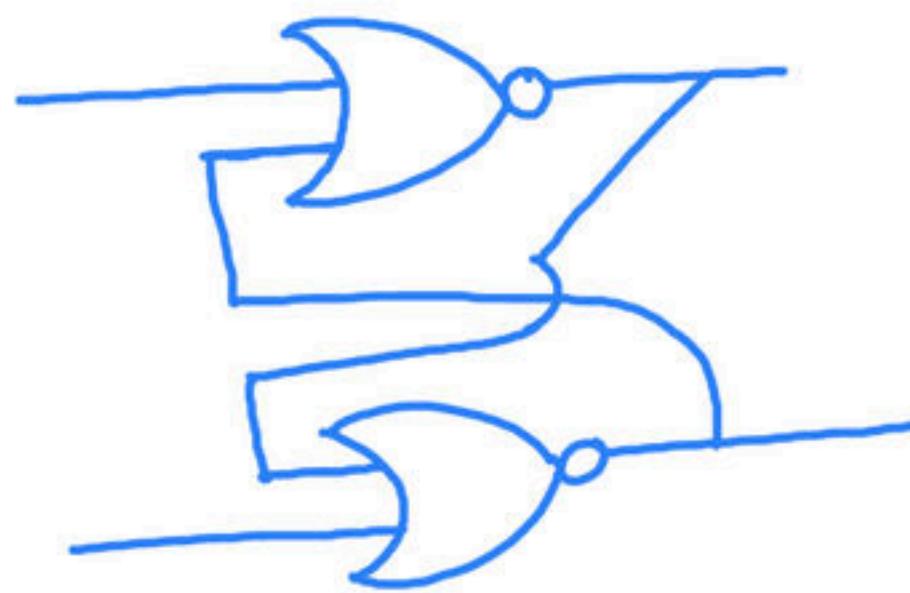
→ Bistable ckt

→ Basic Memory ckt.



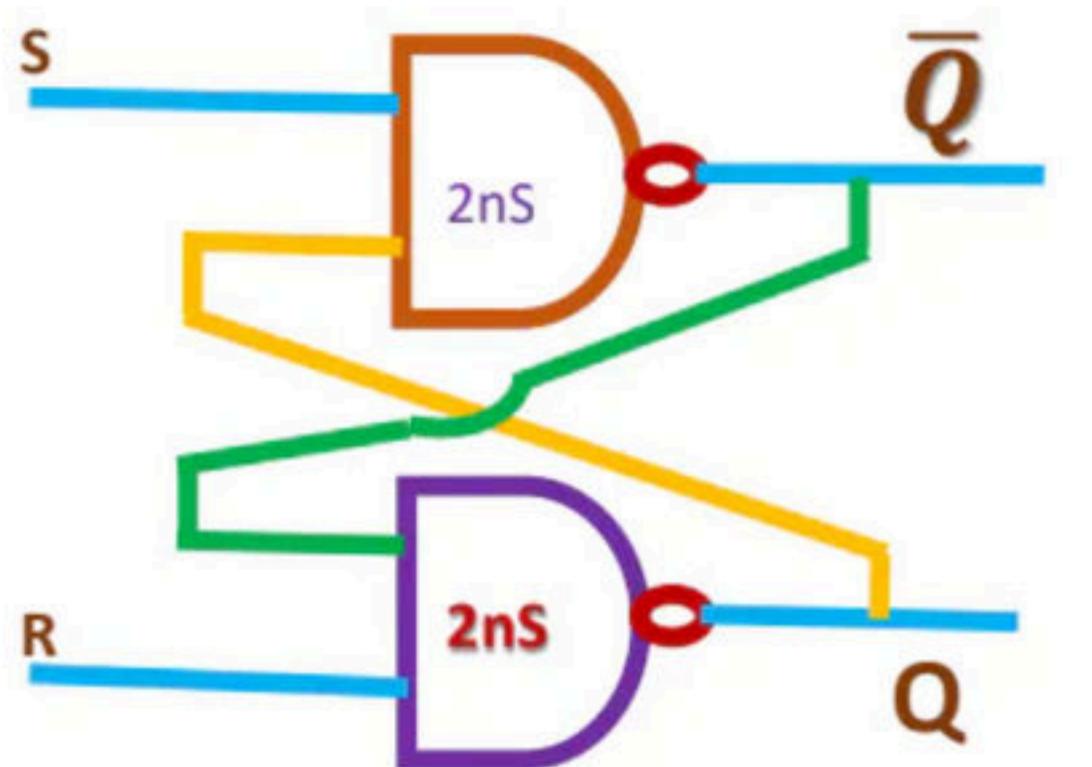


NAND latch.



NOR latch

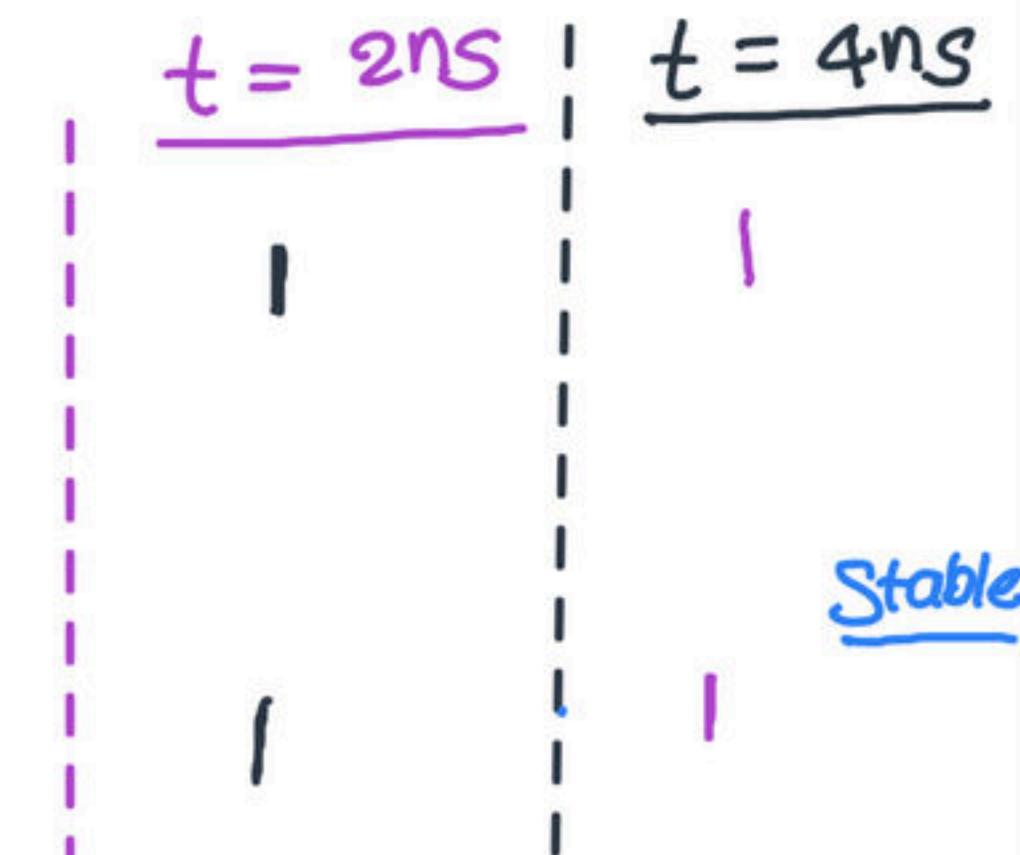
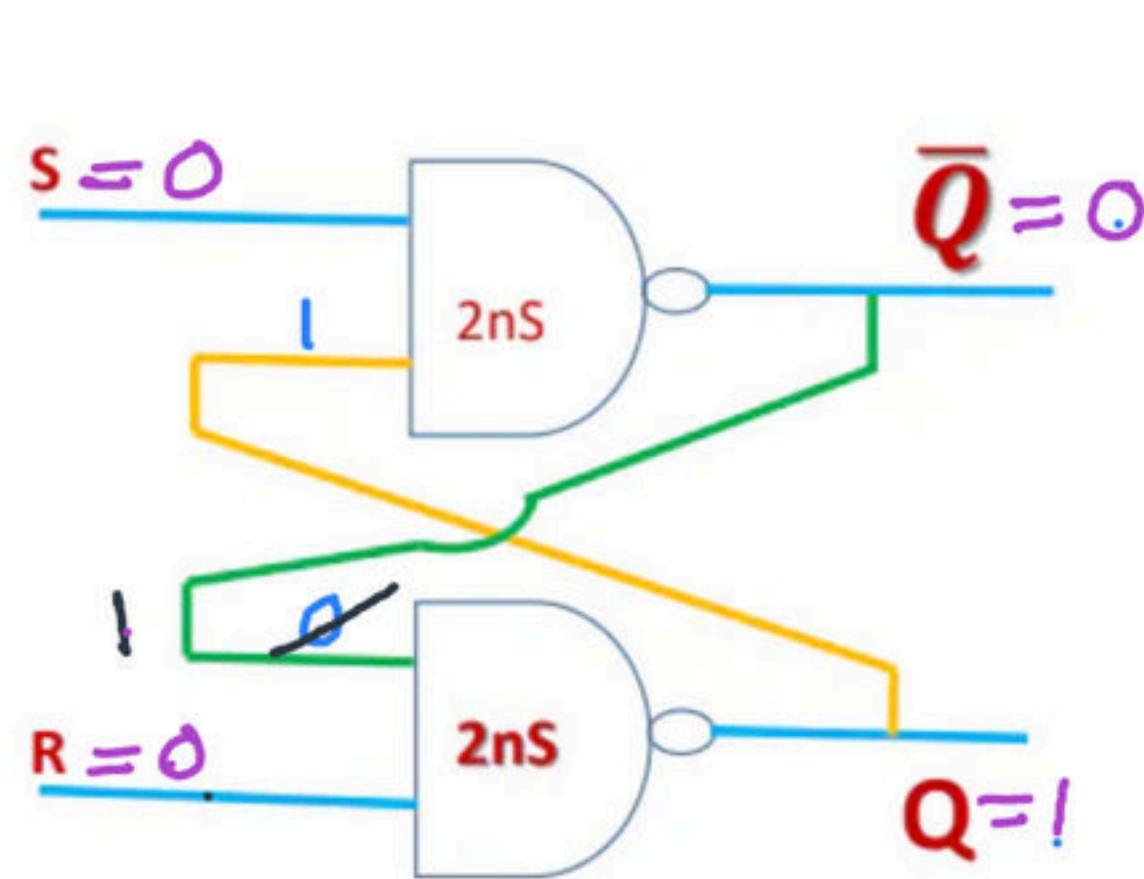
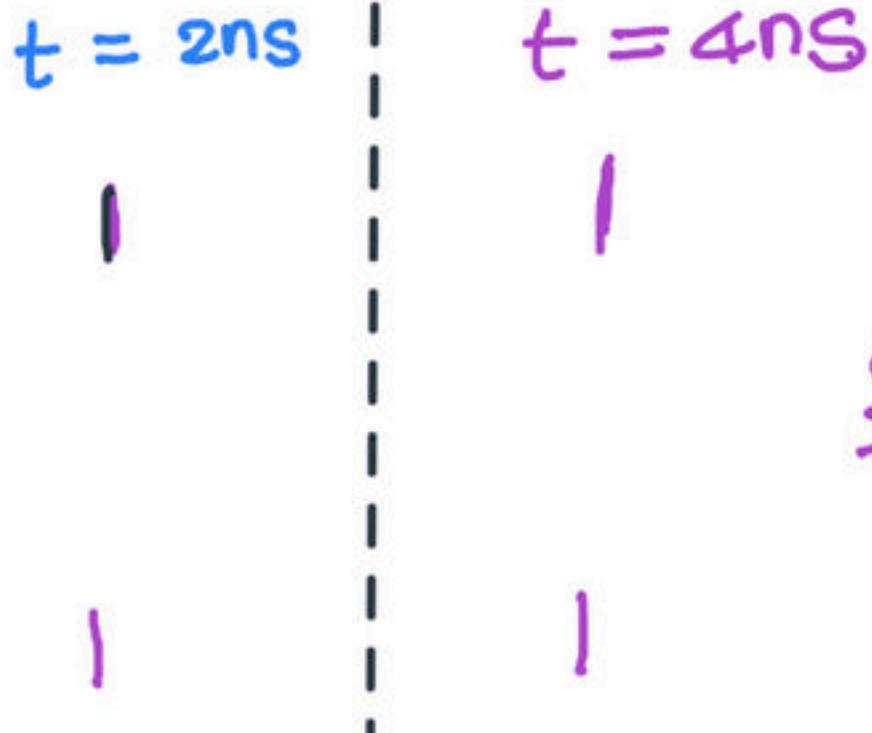
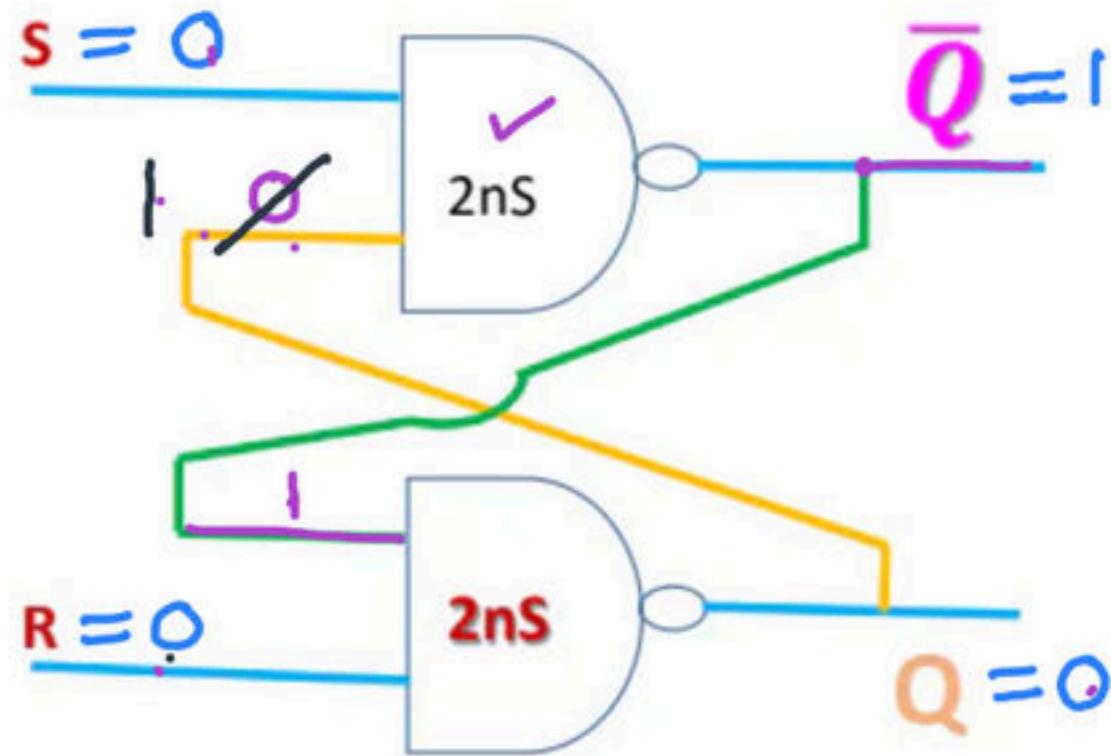
NAND Latch

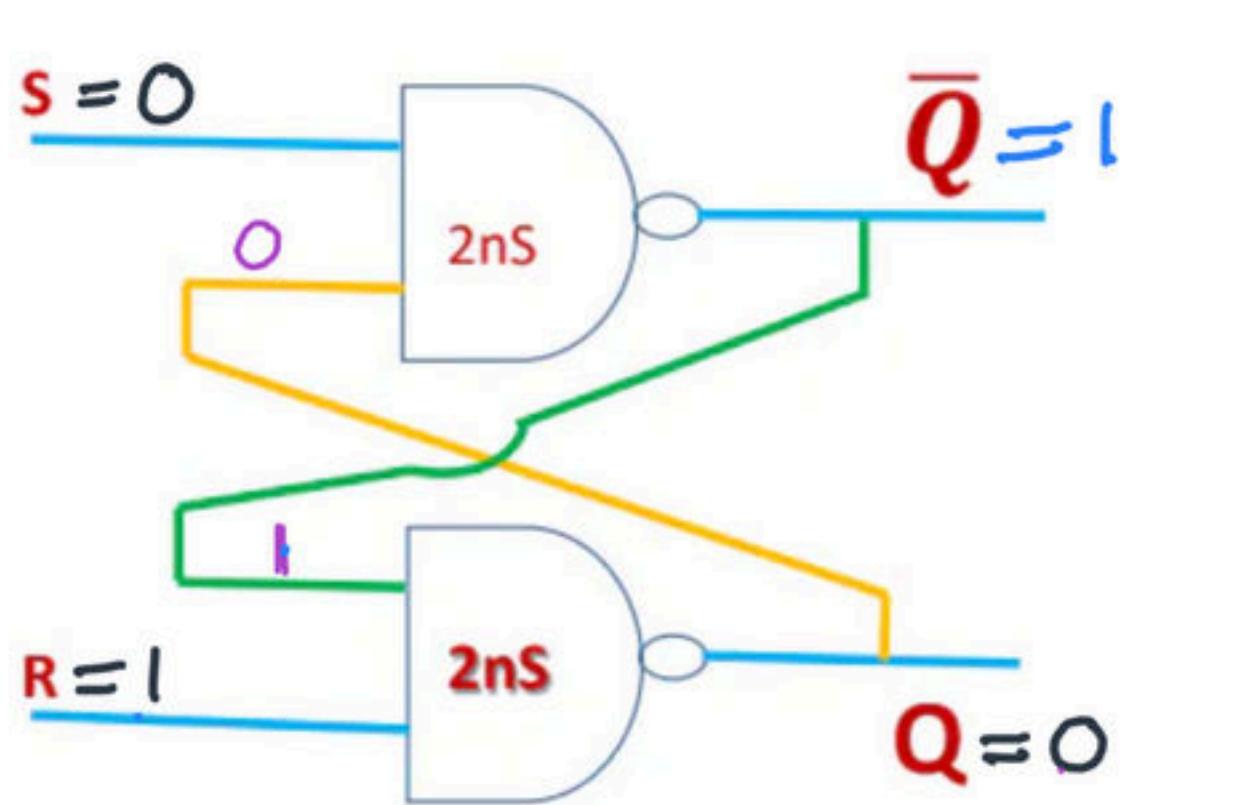


$Q_n \rightarrow$ Present O/P (Previous O/P)

$Q_{n+1} \rightarrow$ Next State O/P (present O/P)

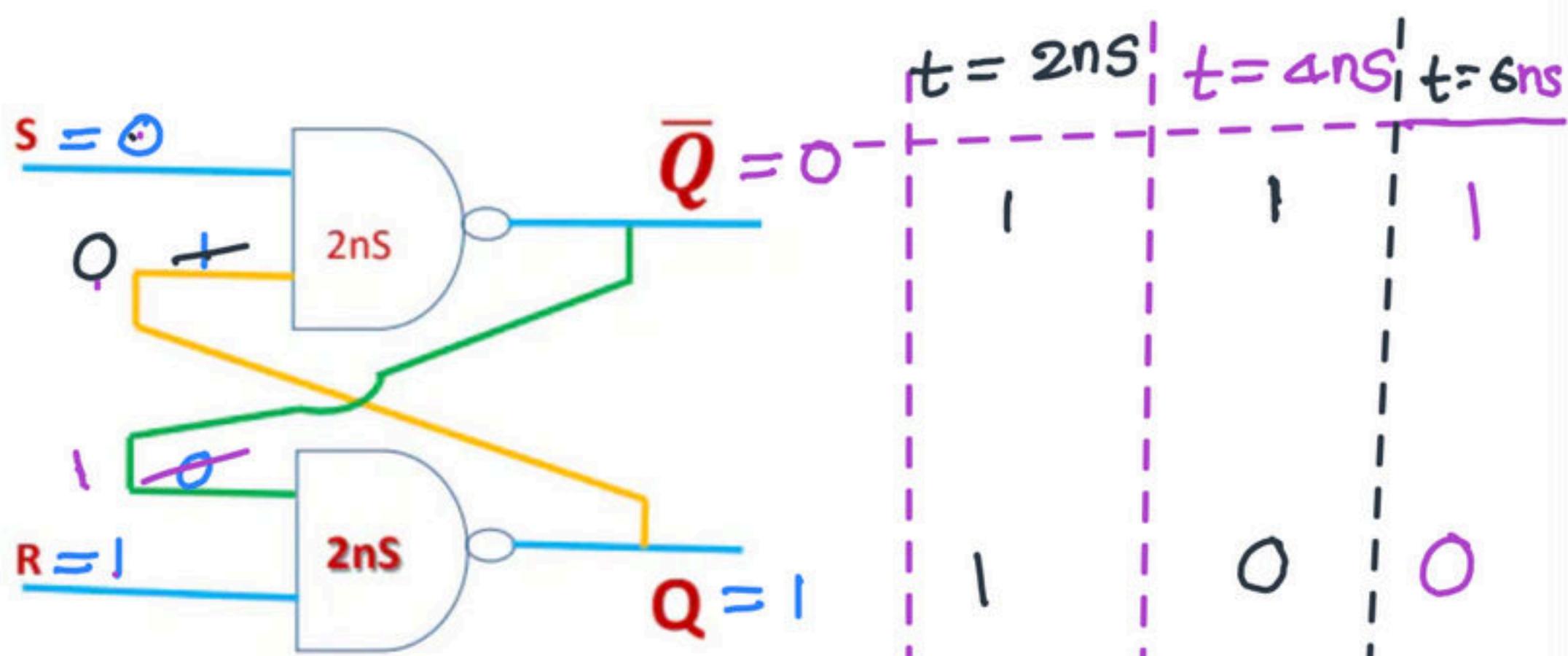
S	R	Q_n	Q_{n+1}	$\overline{Q_{n+1}}$
0	0	0	1	1
0	0	1	1	1
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0



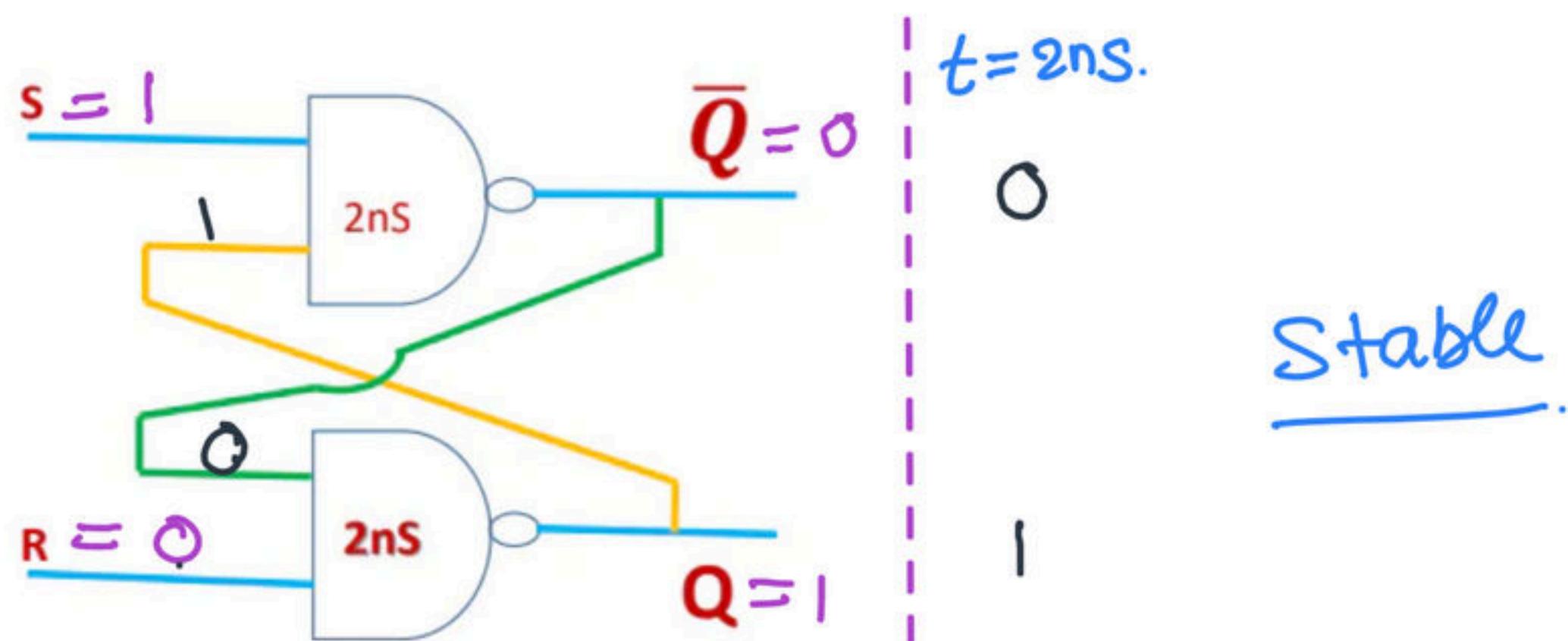
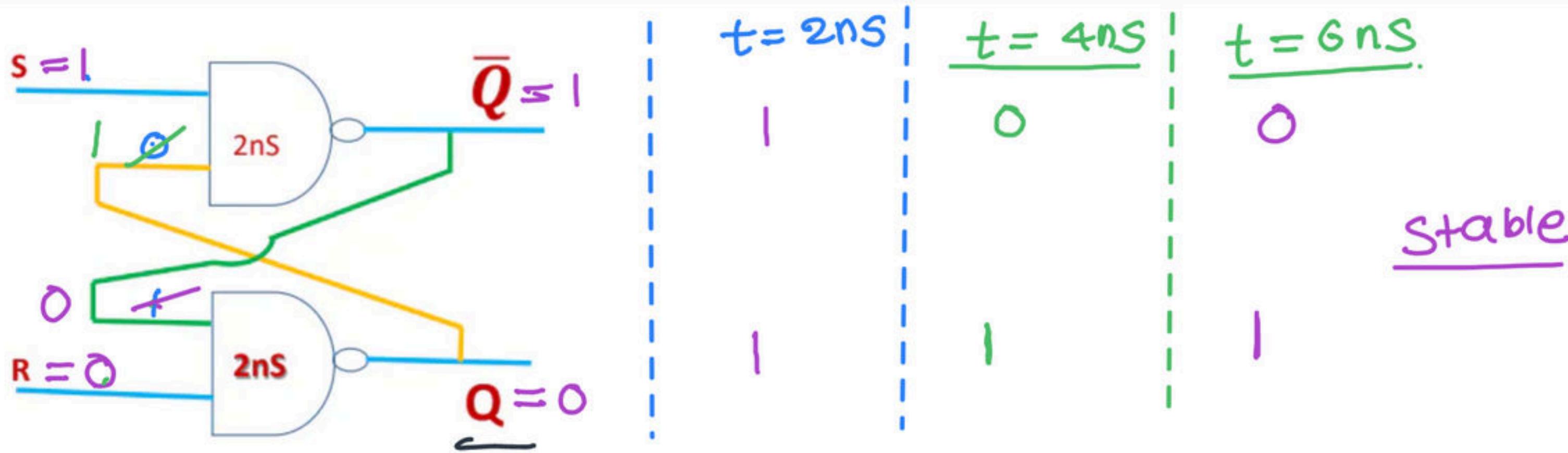


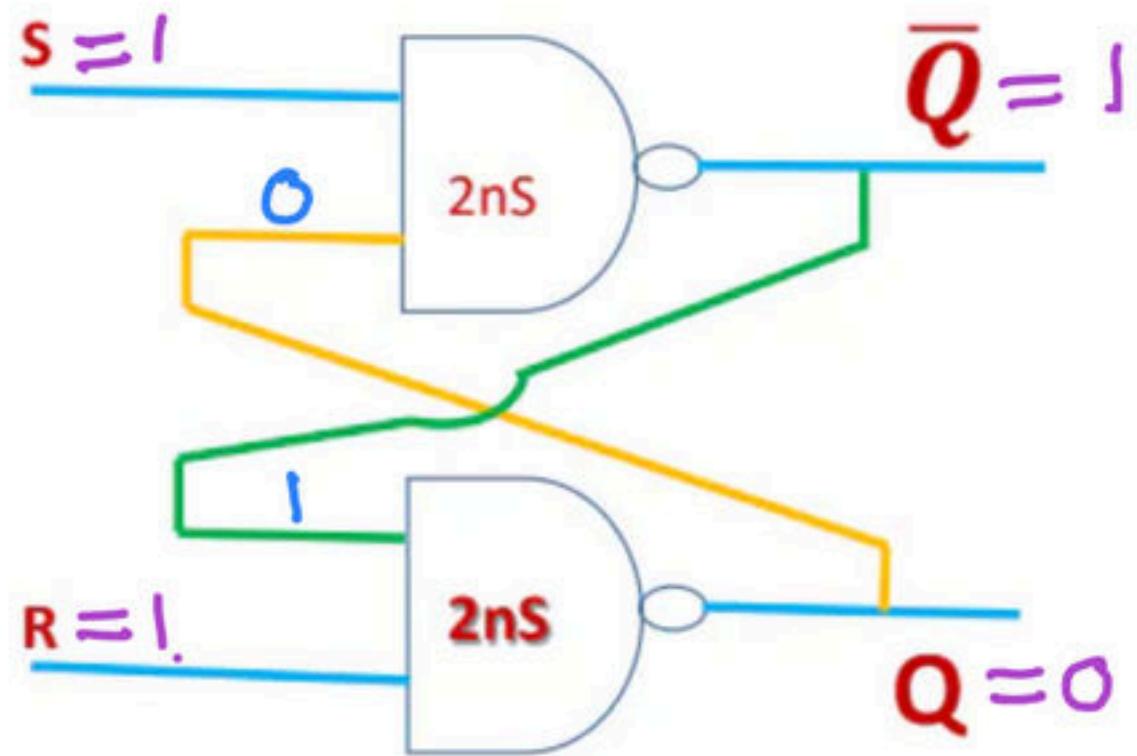
$t = 2\text{ns}$

Stable



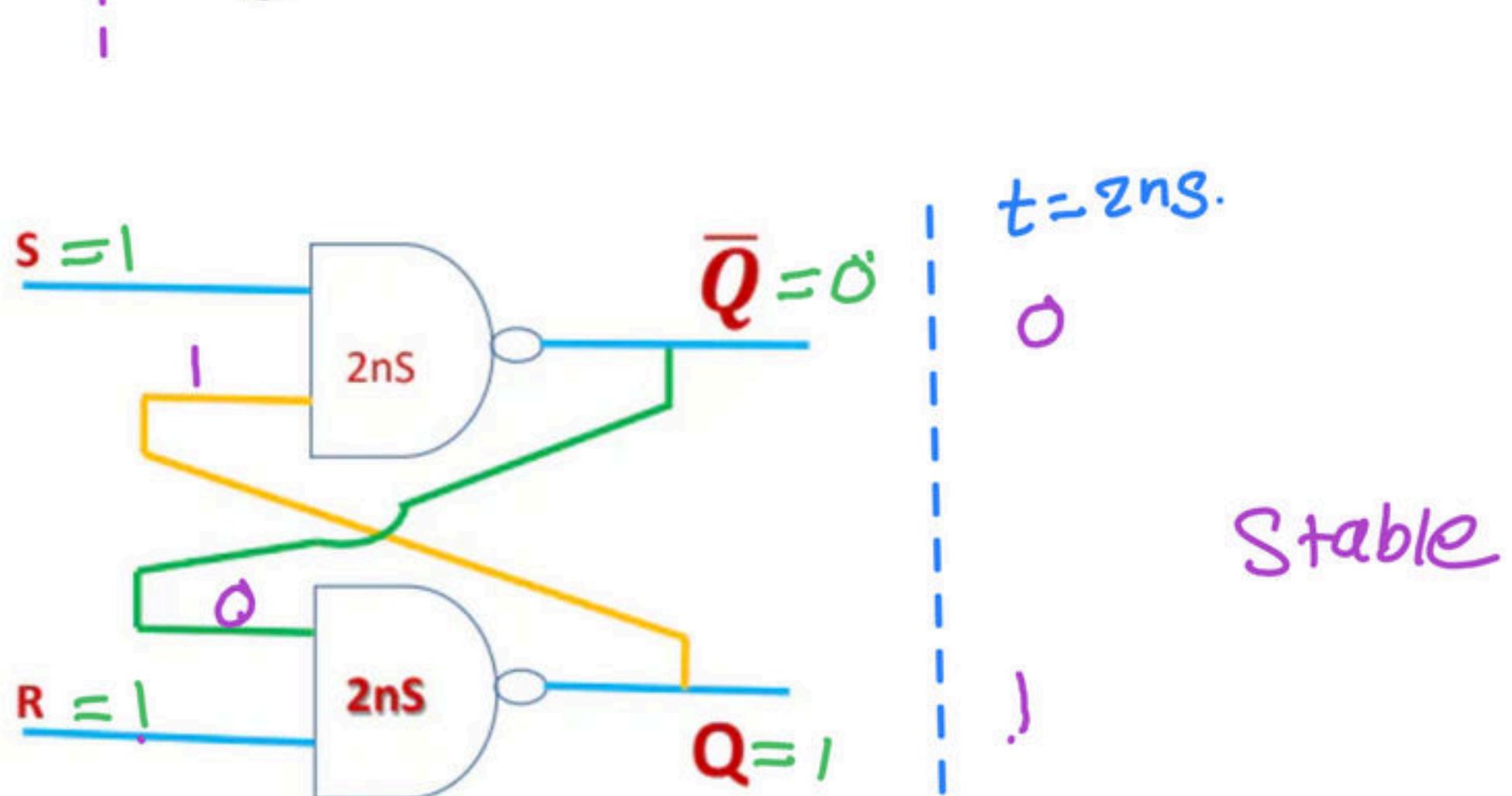
$t = 2\text{ns}$ | $t = 4\text{ns}$ | $t = 6\text{ns}$





$$t = 2nS$$

Stable



Stable

▲ 1 • Asked by Balaram

Sir result input does not satisfy exnor table of 3 input plz explain

Question 6
0.33 MARKS

Your Time Taken: 4m 33s Logic Gates

A function $f(x,y,z) = x \oplus y \oplus z = 1$. Then which one of the following is always true?

$x + yz = 1$ INCORRECT

$xyz = 1$ CORRECT ANSWER

Solution View

Question 7
0 MARKS

RCMIE X22 HS
CLIXED BY: [REDACTED]

$$\begin{array}{c} x \oplus y \oplus z = 1 \\ \hline \end{array}$$

Let $x=1$

$y=1, z=0$

$$x = y \oplus z$$

▲ 1 • Asked by Balaram

Sorry sir

Question 6
-0.33 MARKS

Your Time Taken: 4m 33s Logic Gates

A function $f(x, y, z) = x \odot y \odot z = 1$. Then which one of the following is always true?

$x + yz = 1$ ✗

$x = y \oplus z$
INCORRECT ✗.

$x.y.z = 1$ ✗.

$x = y \odot z$
CORRECT ANSWER ✗.

Solution View
CLICKED BY: RAJ

$$x = y \odot z$$

$$x = 1$$

$$x = 0$$

$$x = 1$$

$$y = 1, \quad z = 0$$

$$\underline{y = 1}, \quad \underline{z = 1}$$

$$y = 0, \quad z = 1$$

▲ 1 • Asked by Balaram

Please help me with this doubt

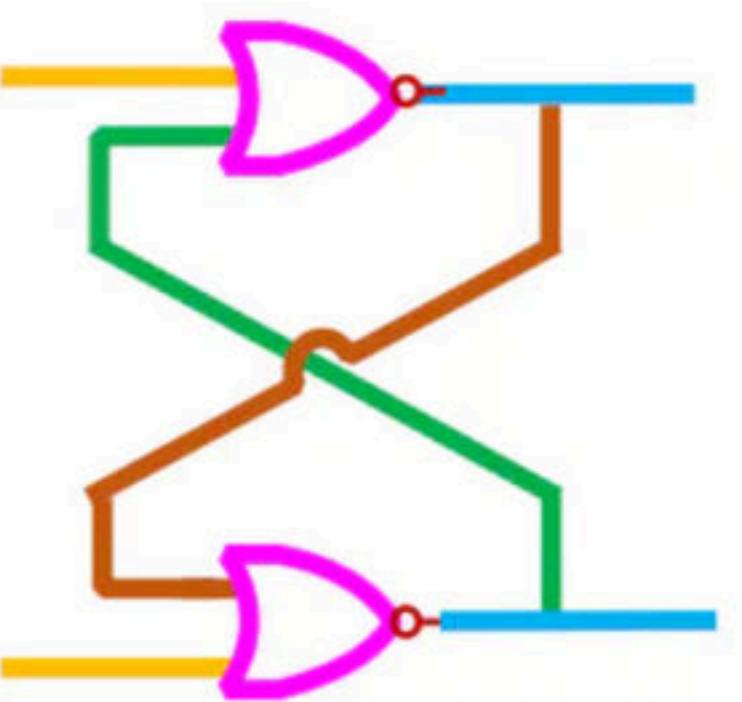
Consider the pseudo code given below-

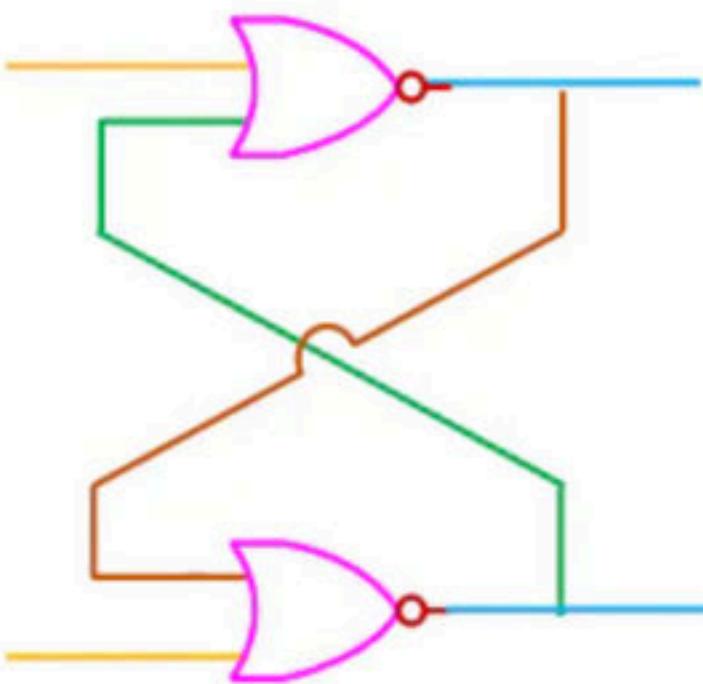
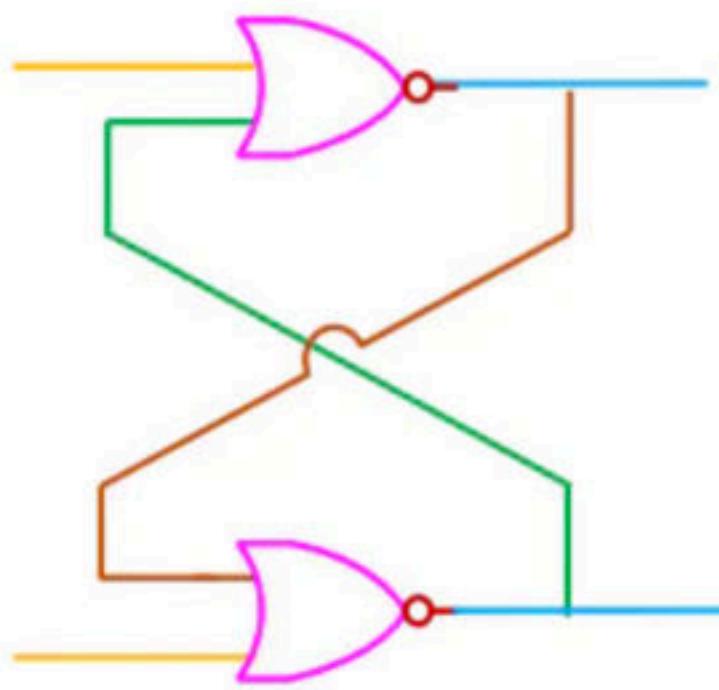
```
struct node{  
    int val;  
    struct node * next;  
}  
struct node *XYZ (struct node * H)  
{  
    struct node *prev =NULL;  
    struct node *nextNode=NULL;  
    while(cur)  
    {  
        nextNode=cur->next;  
        cur->next=prev;  
        previous=cur;  
        cur=nextNode;  
    }  
    return prev;  
}  
};
```

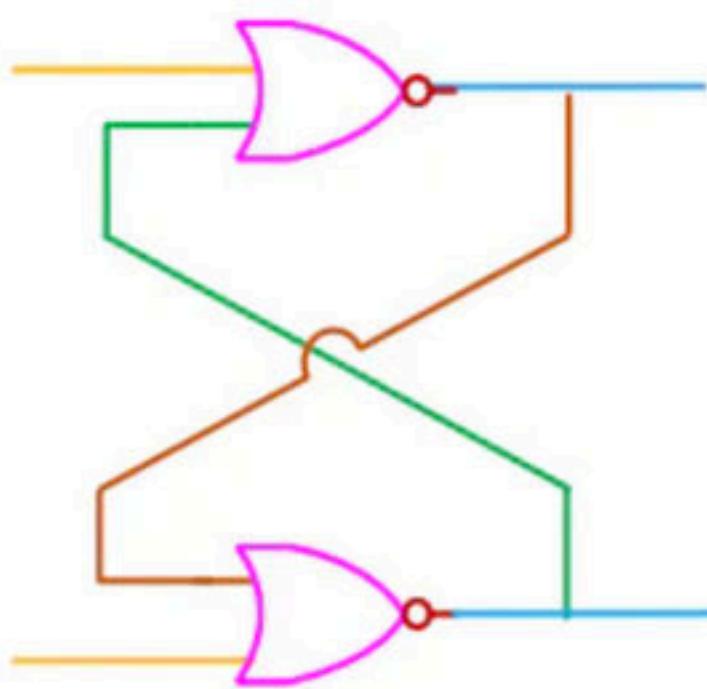
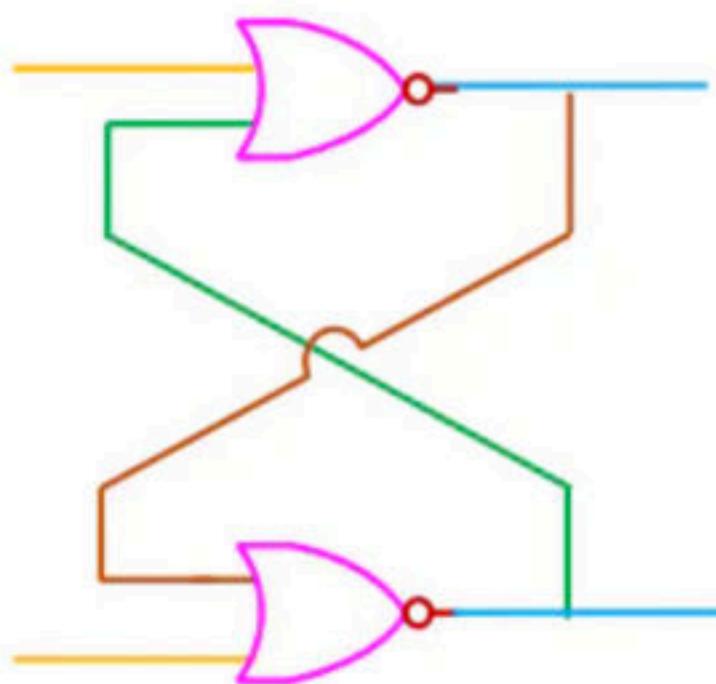
What does the function "XYZ" do if it is passed a pointer of a "Head" that points to starting node of linked list.

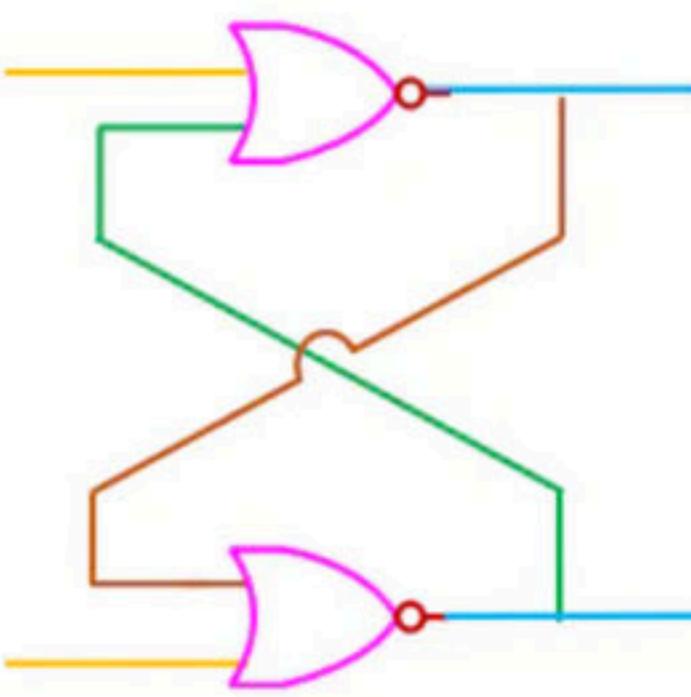
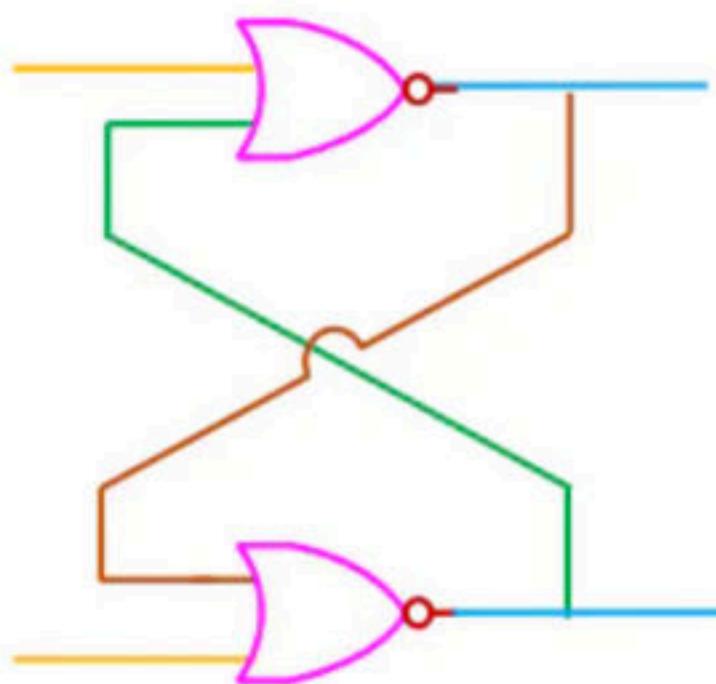
S	R	Q_{n+1}	State

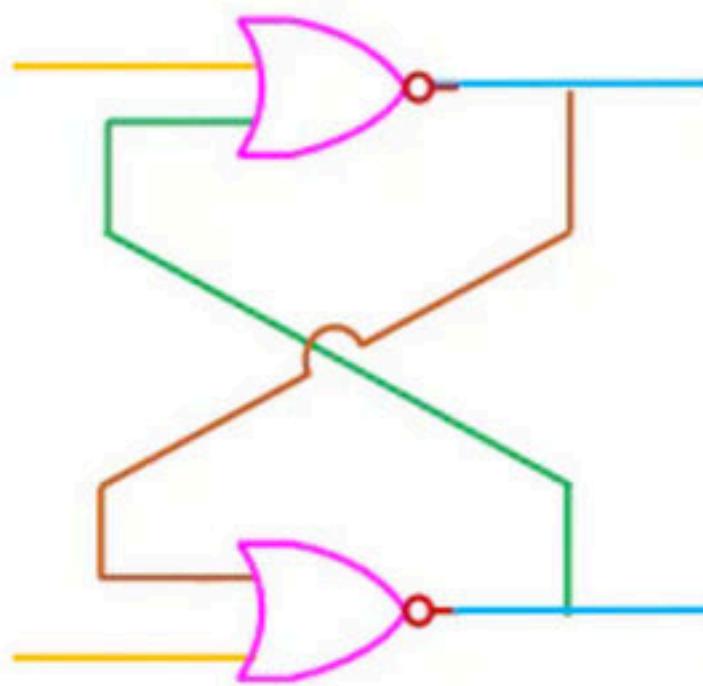
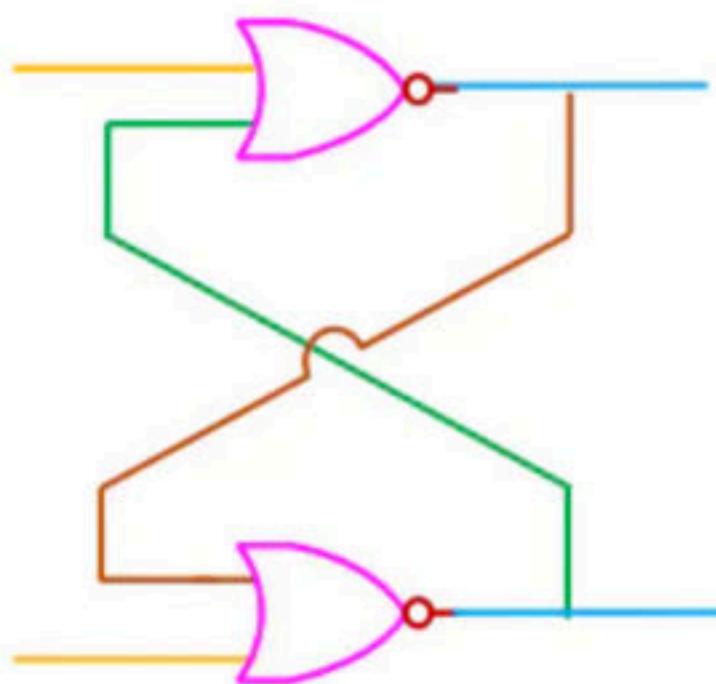
NOR Latch



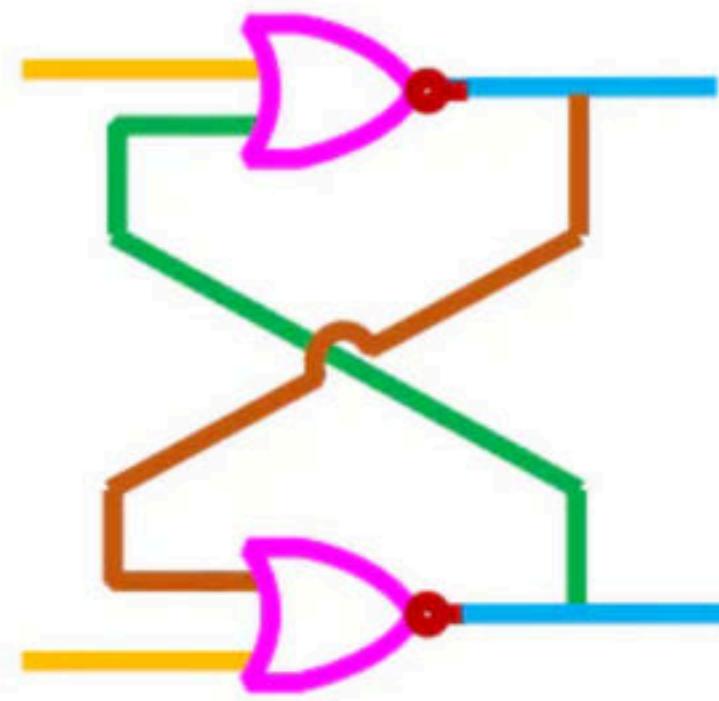
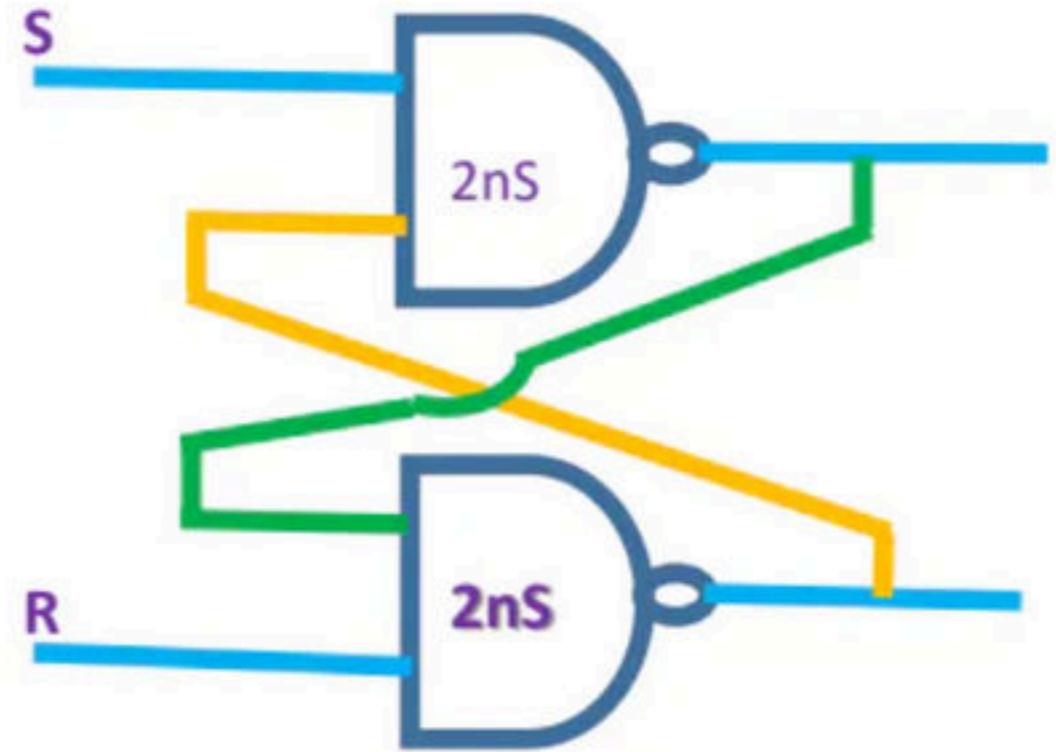






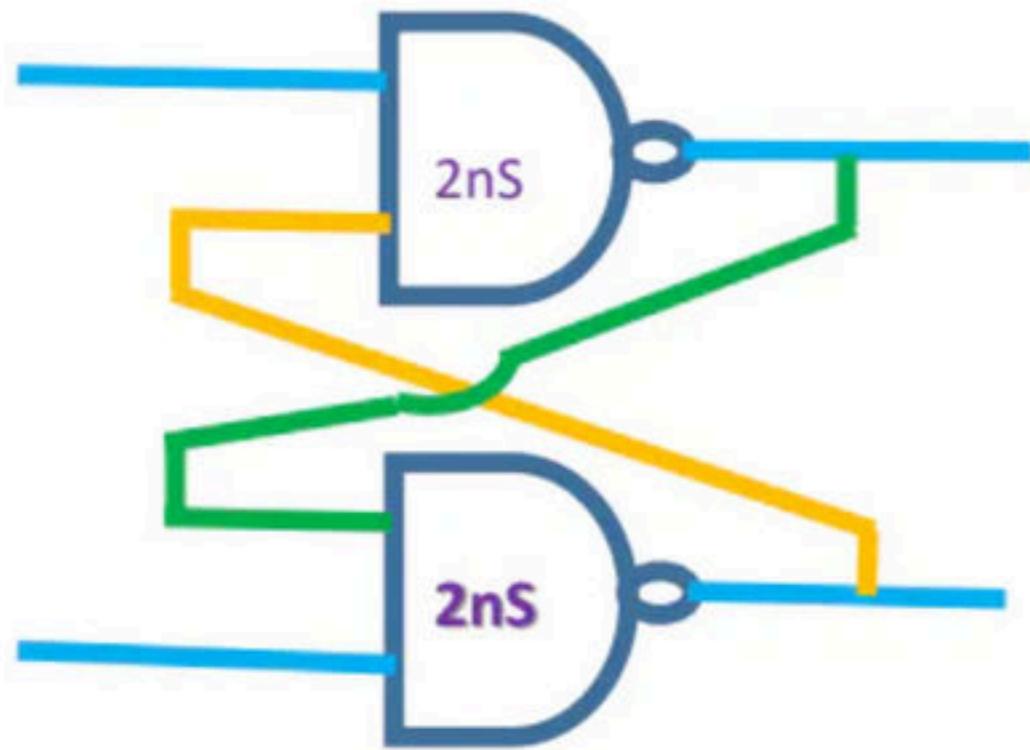


S	R	Q_{n+1}	State

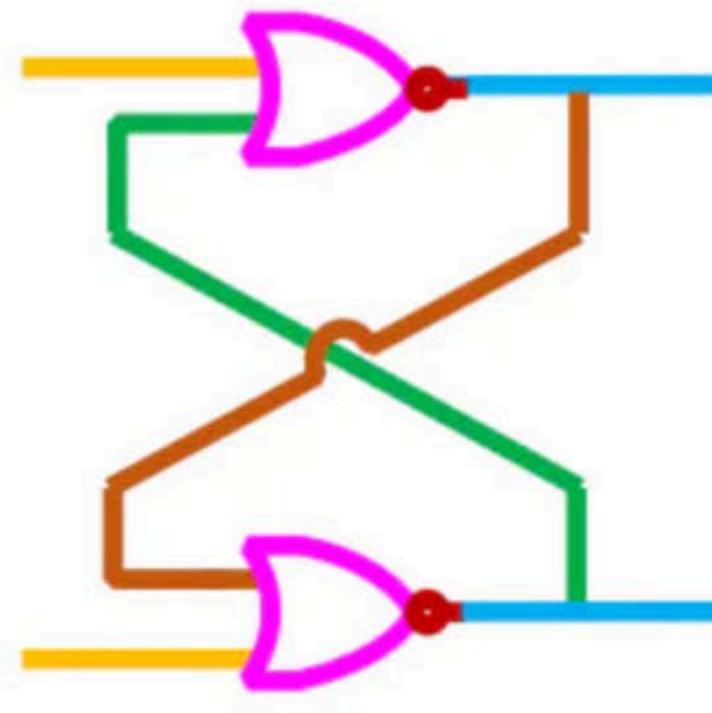


S	R	Q_{n+1}	State

S	R	Q_{n+1}	State

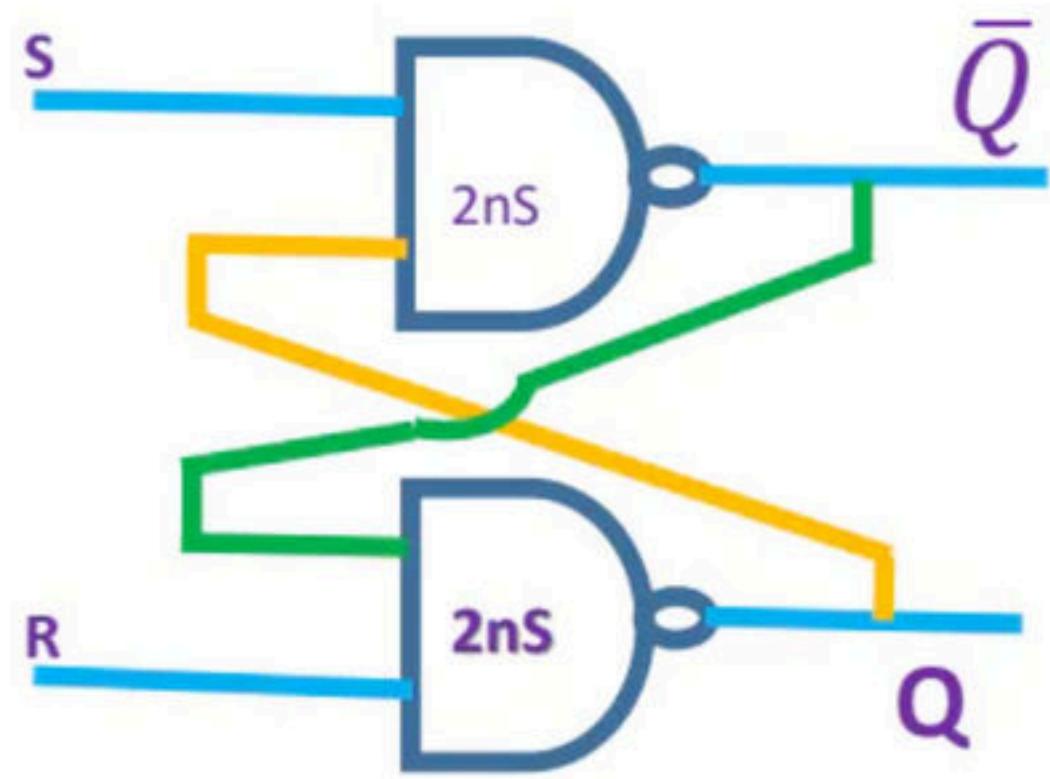


S	R	Q_{n+1}	State



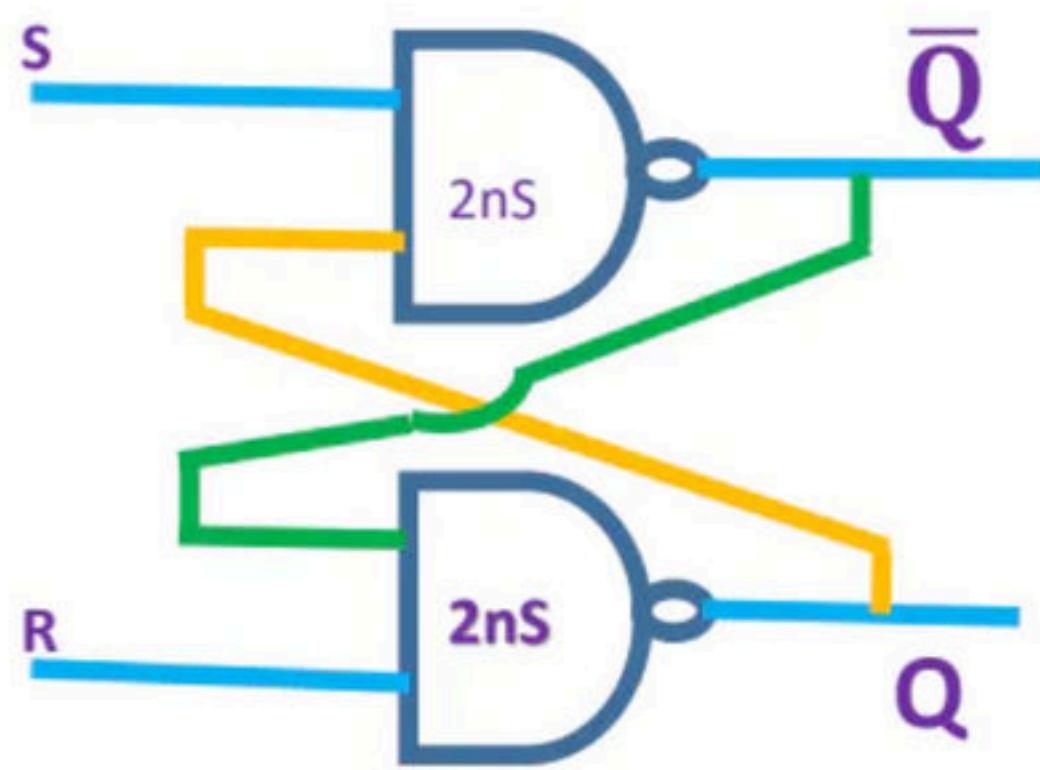
S	R	Q_{n+1}	State

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

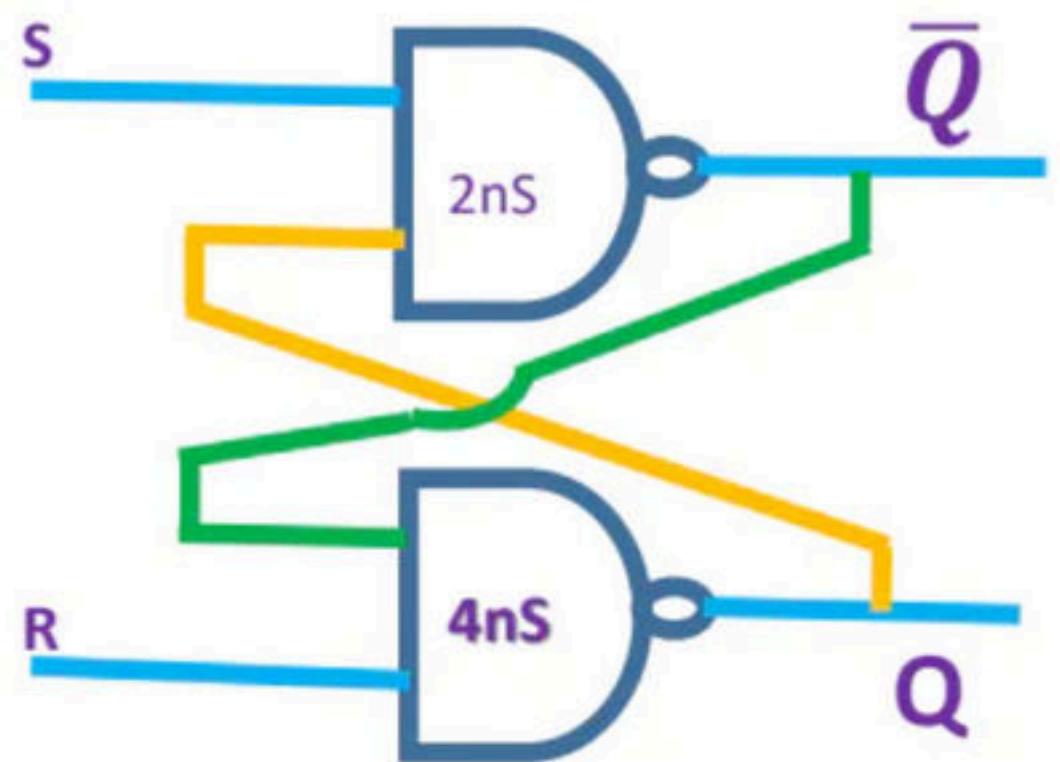


- Out put of combinational circuit depends on input combinations
- Output of sequential circuits depends on input sequence
- For unequal delay of gates also the operation is valid

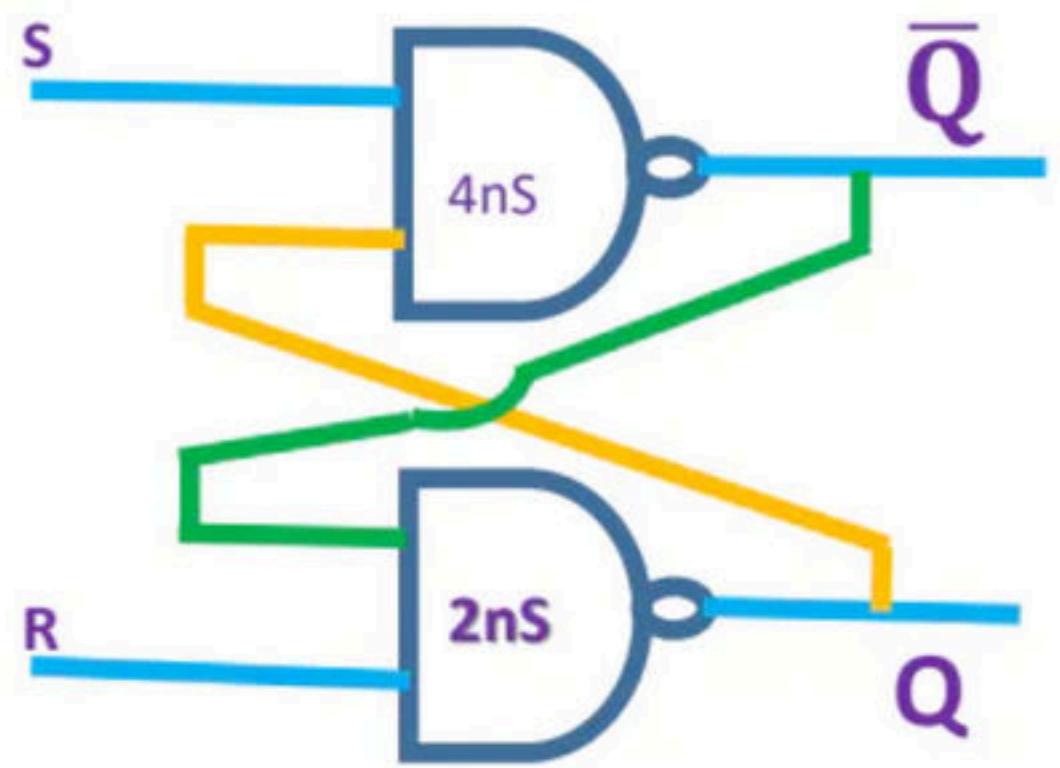
Q) Assume initially $Q=0$, for given circuit then find output for the following sequence
 $SR = 00 \rightarrow 11$, assume equal delay



Q) Assume initially $Q = 0$, for given circuit then find output for the following sequence
 $SR = 00\rightarrow 11$, unequal delay



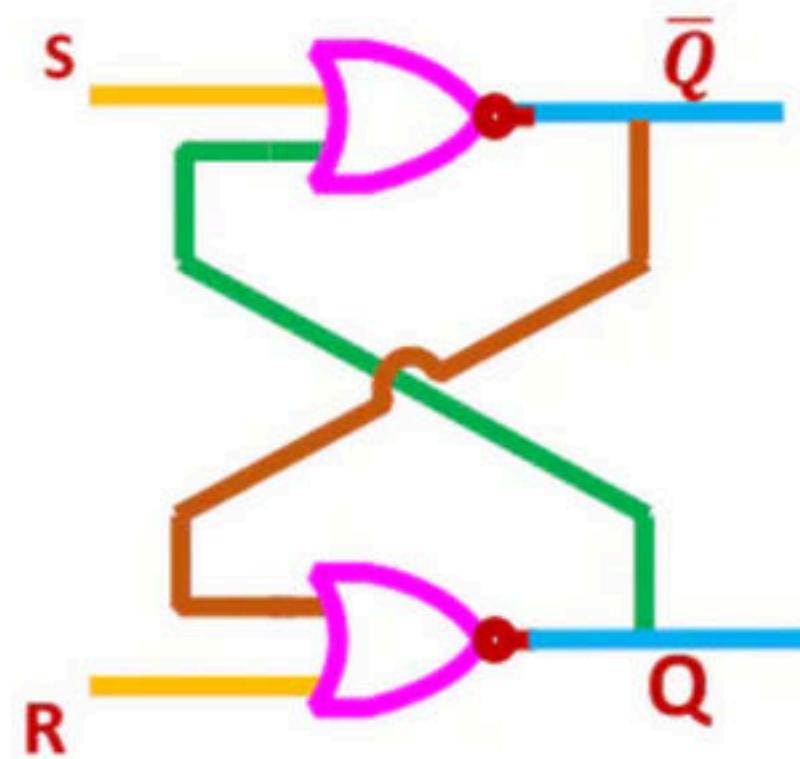
Q) Assume initially $Q = 0$, for given circuit then find output for the following sequence
 $SR = 00\rightarrow 11$, unequal delay



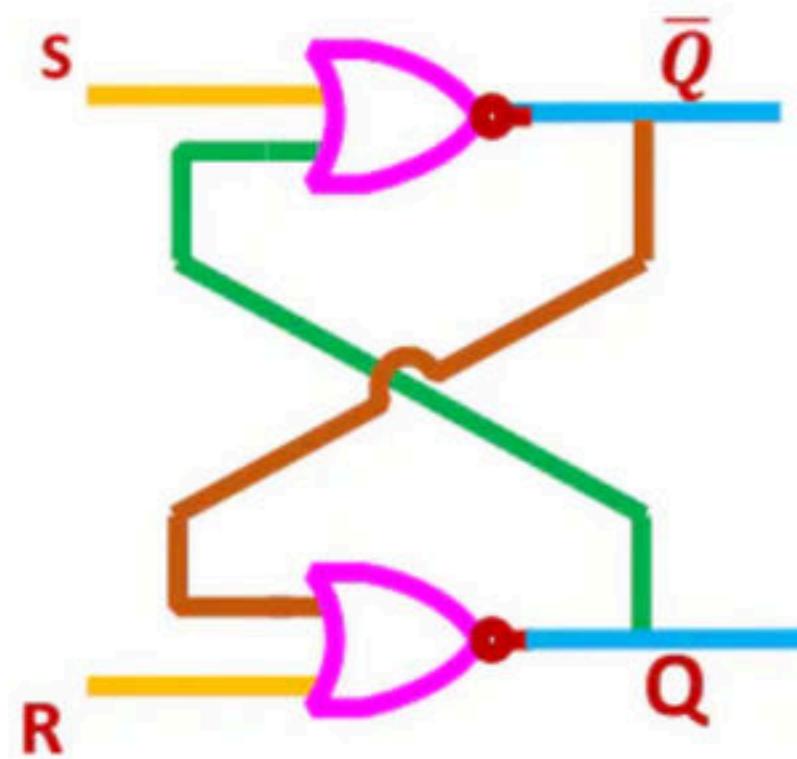
For **SR NAND** latch , if the input sequence is **00 followed by 11** , then the following cases arises

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

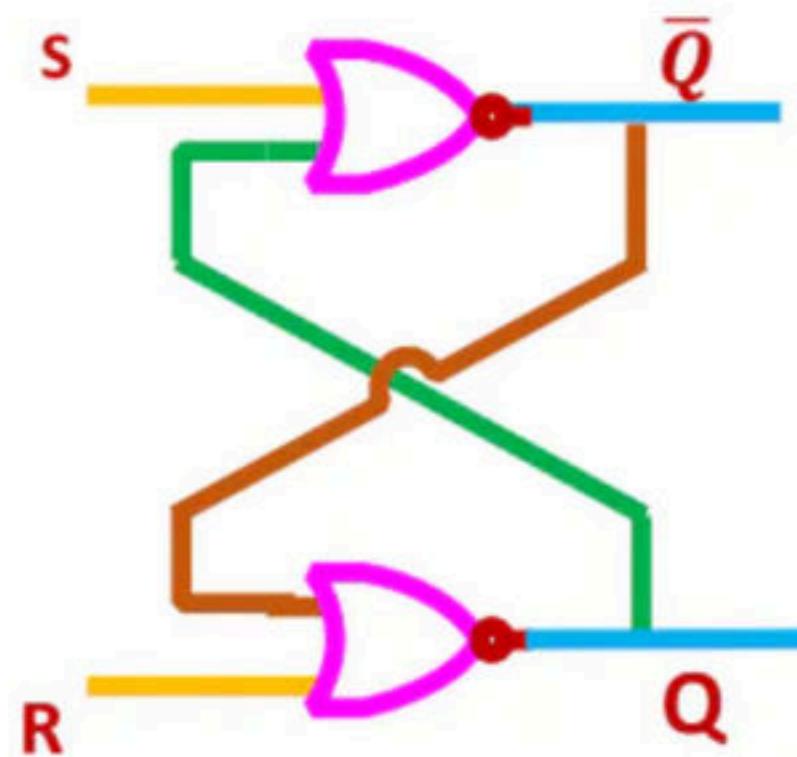
Q) Assume initially $Q=0$, for given circuit then find output for the following sequence
 $SR = 01 \rightarrow 10 \rightarrow 00 \rightarrow 11 \rightarrow 01 \rightarrow 11 \rightarrow 10$



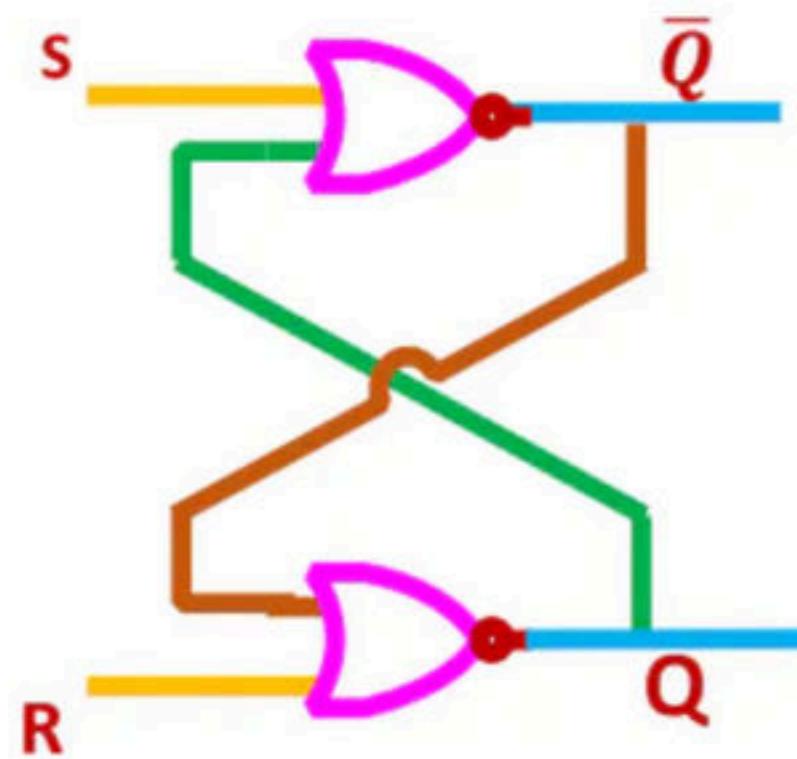
Q) Assume initially $Q=0$, for given circuit then find output for the following sequence
 $SR = 11 \rightarrow 00$, Assume equal delay



Q) Assume initially $Q=0$, for given circuit then find output for the following sequence
 $SR = 11 \rightarrow 00$, Assume unequal delay



Q) Assume initially $Q=0$, for given circuit then find output for the following sequence
 $SR = 11 \rightarrow 00$, Assume unequal delay

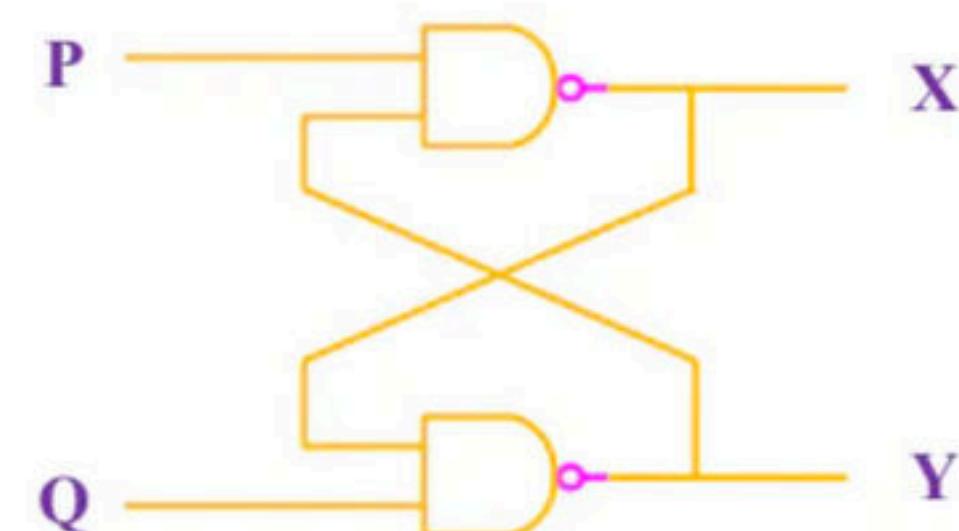


For **SR NOR** latch , if the input sequence is 11 followed by 00 , then the following cases arises

- 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

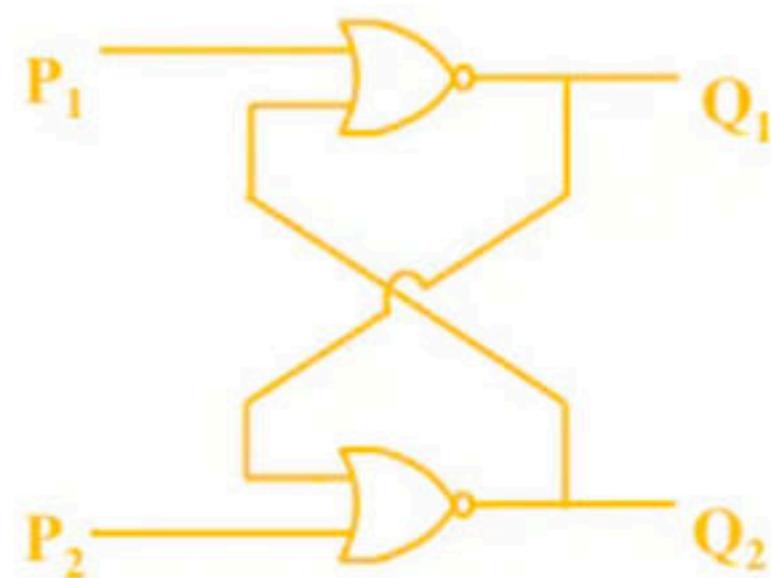
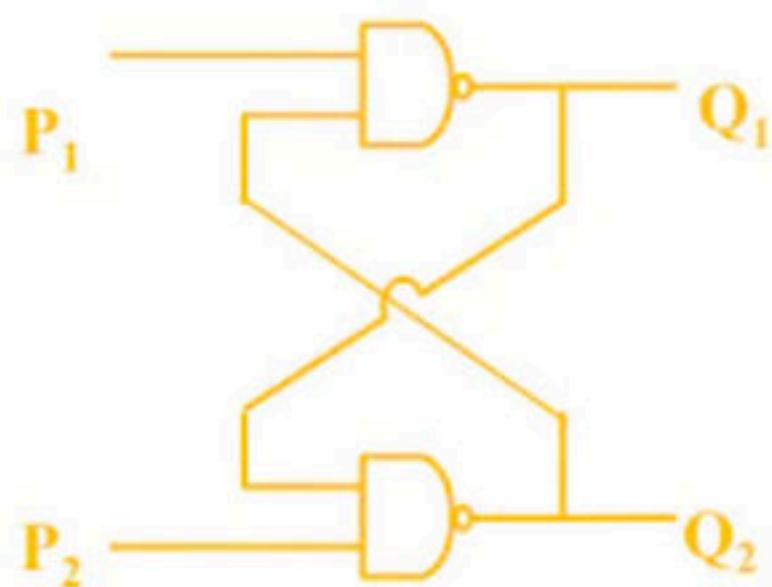
Q. In the latch circuit shown, the NAND gates have non-zero, but unequal propagation delays. The present input condition is $P = Q = '0'$. If the input condition is changed simultaneously to $P = Q = '1'$, the outputs X and Y are

- (A) $X = '1'$, $Y = '1'$
- (B) Either $X = '1'$, $Y = '0'$ or $X = '0'$, $Y = '1'$
- (C) Either $X = '1'$, $Y = '1'$ 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

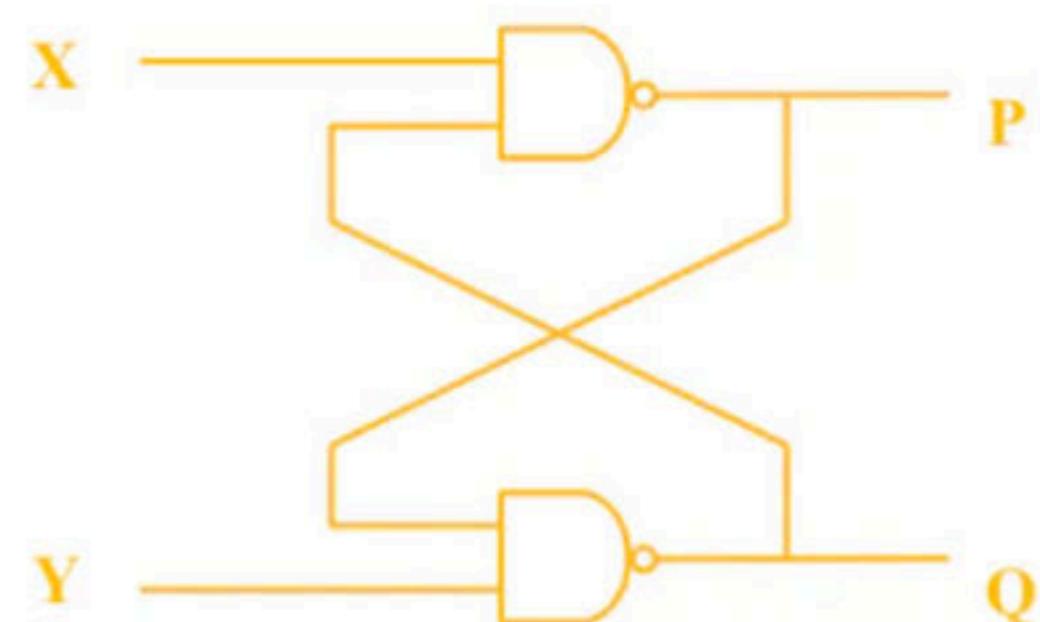
- (A)NAND: first (0,1) then (0,1) NOR: first (1,0) then (0,0)
- (B)NAND: first (1,0) then (1,0) NOR: first (1,0) then (1,0)
- (C)NAND: first (1,0) then (1,0) NOR: first (1,0) then (0,0)
- (D)NAND: first (1,0) then (1,1) NOR: first (0,1) then (0,1)



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:

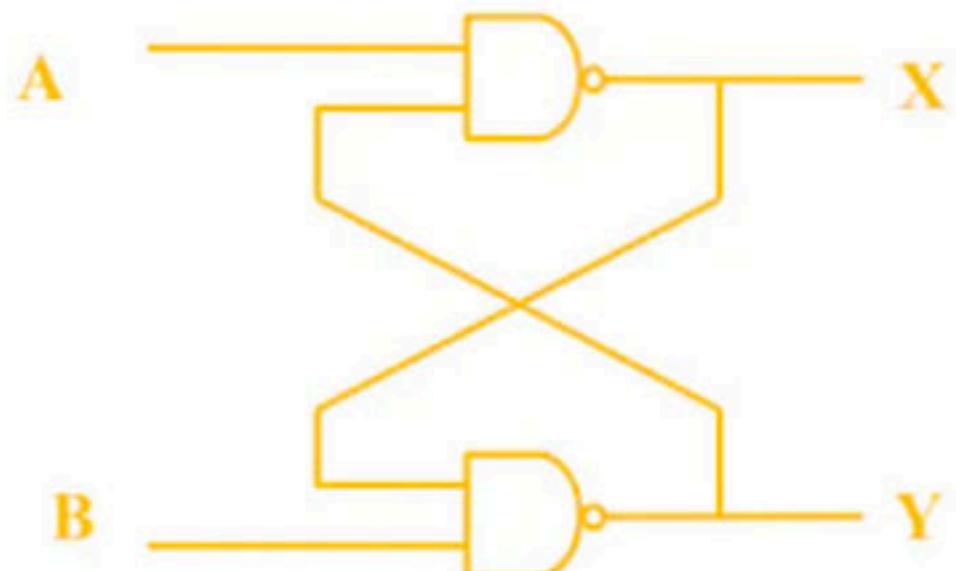
$X = 0, Y = 1$; $X = 0, Y = 0$; $X = 1, Y = 1$. The corresponding stable P, Q outputs will be:

- (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 $P = 0, Q = 1$



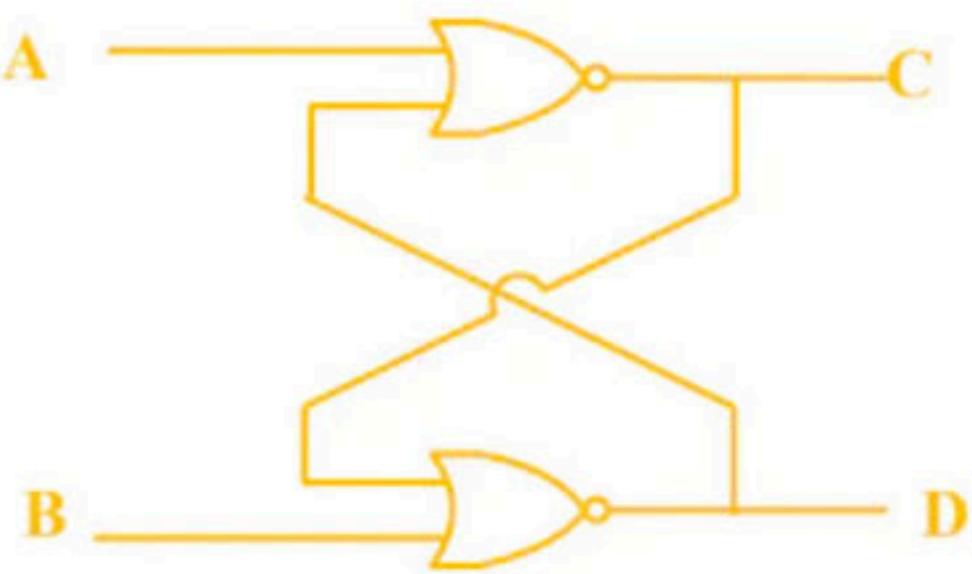
Q. The given figure, $A = 1$ and $B = 1$, the input B is now replaced by a sequence 101010....
The outputs x and y will be.

- (A) Fixed at 0 and 1, respectively.
- (B) $x = 1010 \dots$ While $y = 0101 \dots$
- (C) $x = 1010 \dots$ and $y = 0101 \dots$
- (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) $C = 0, D = 1$



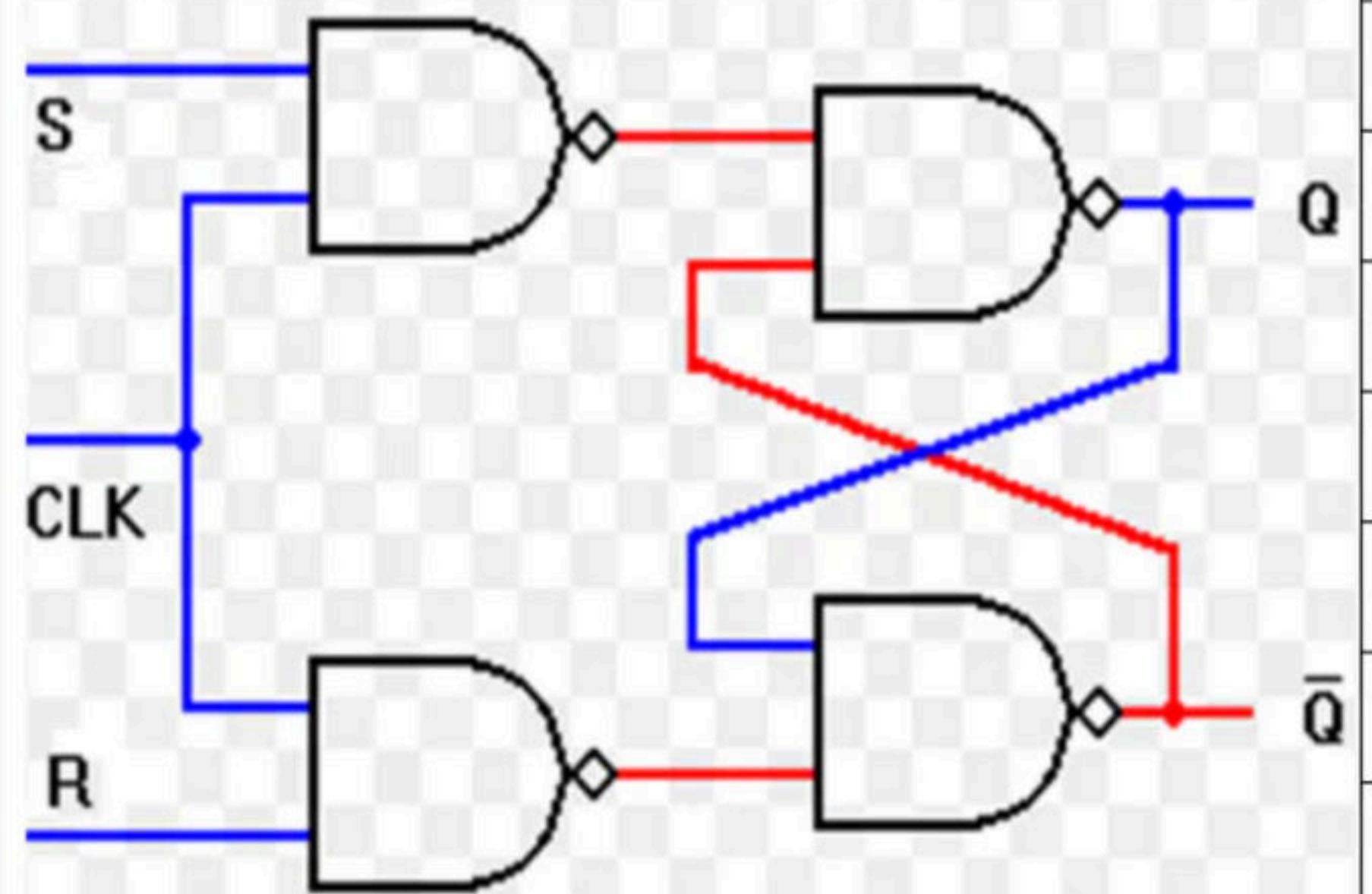
FLIP FLOP

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.

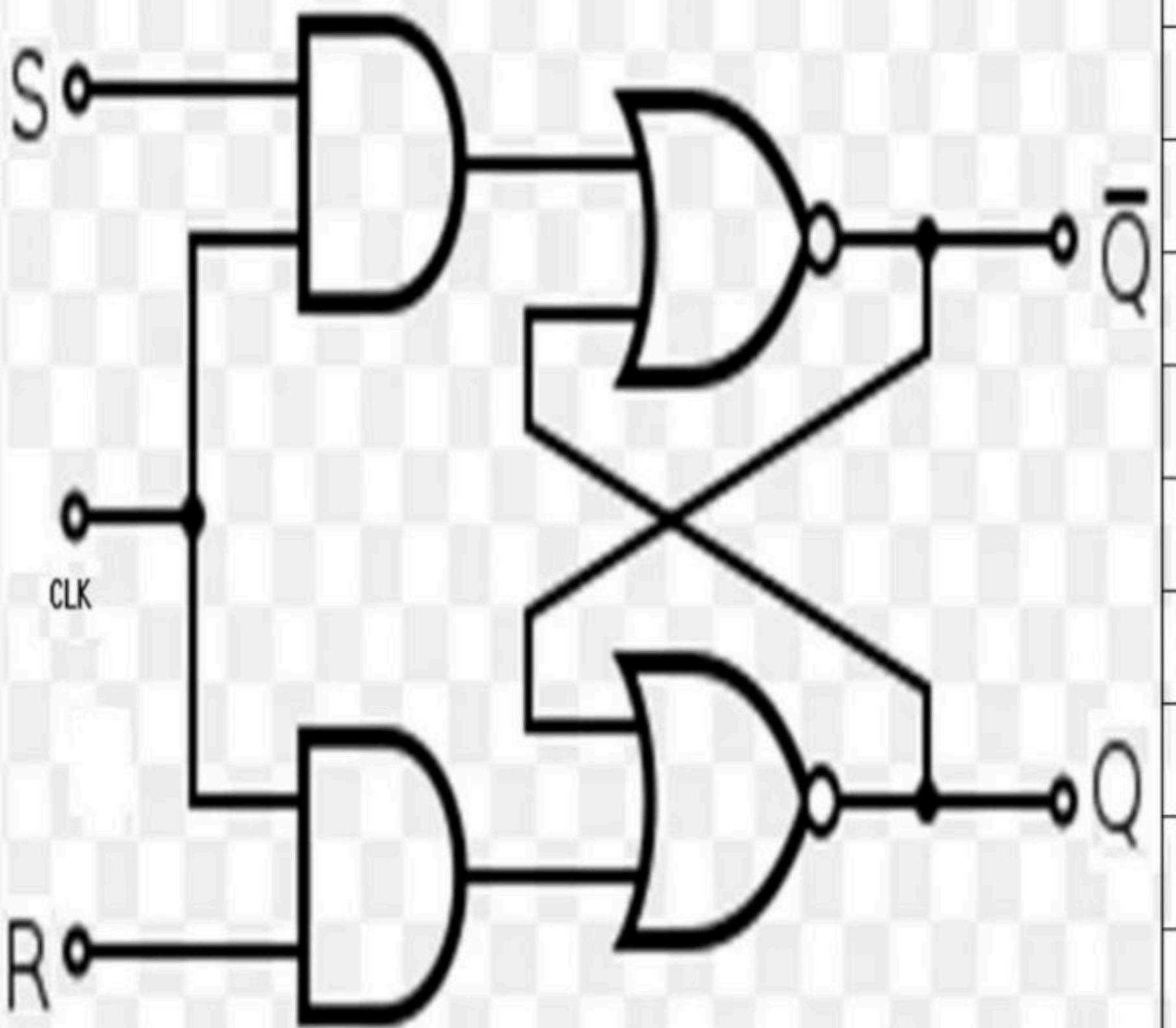
Latch + Clock = Flip Flop

SR Flip Flop

1. SR Flip Flop using NAND Latch

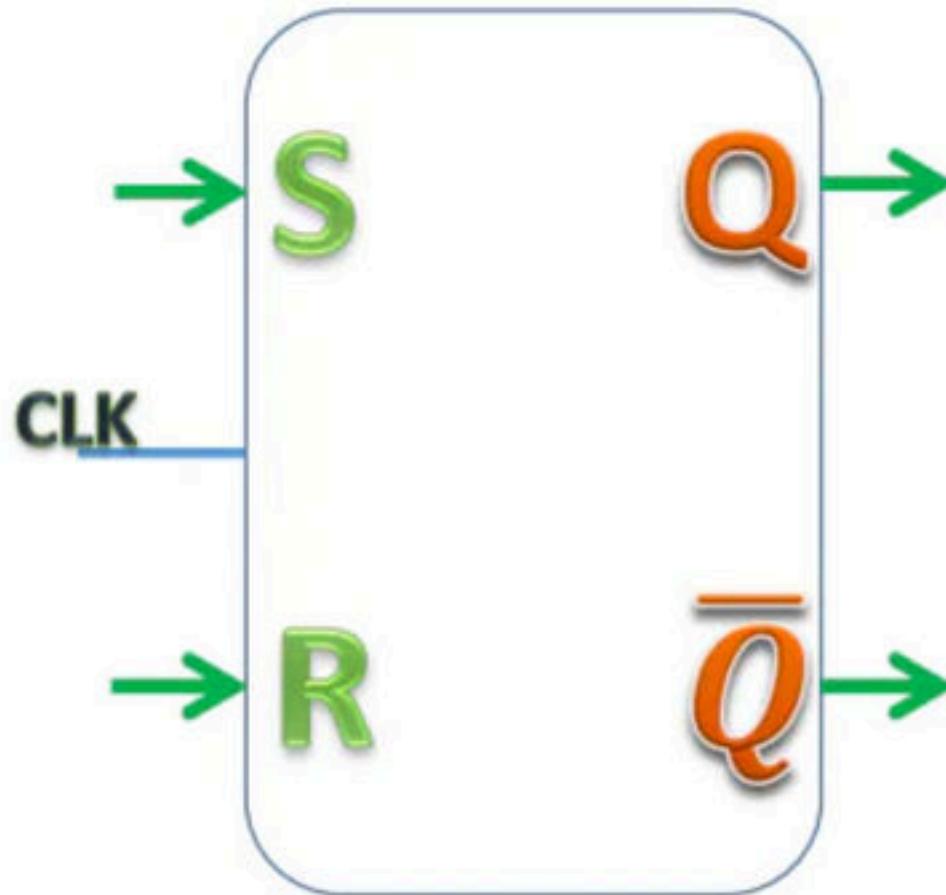


2. SR Flip Flop using NOR Latch



- 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	x	x		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Characteristic table

Characteristic Equation

Excitation table

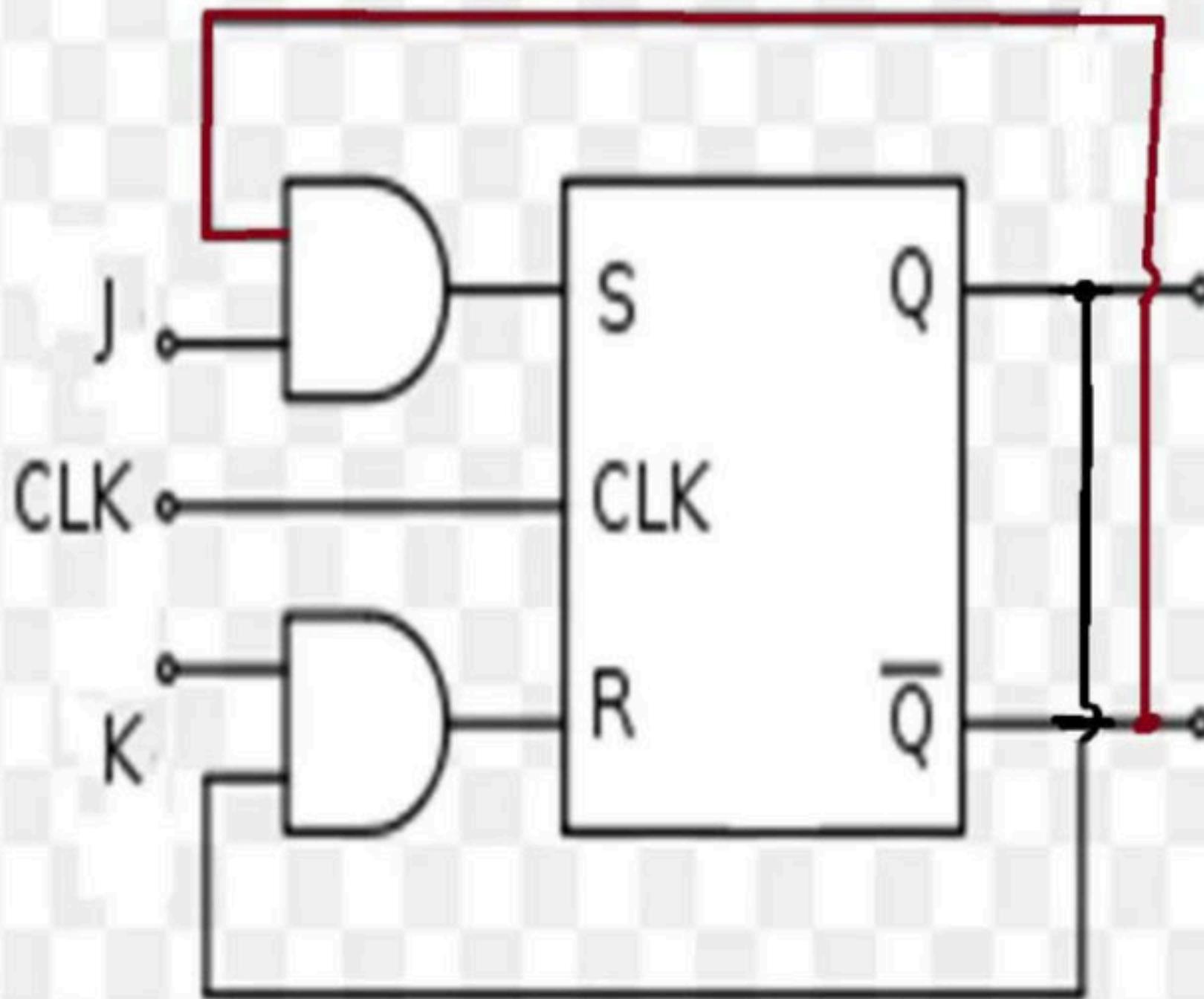
Q	Q+	S	R

State Diagram

Drawbacks

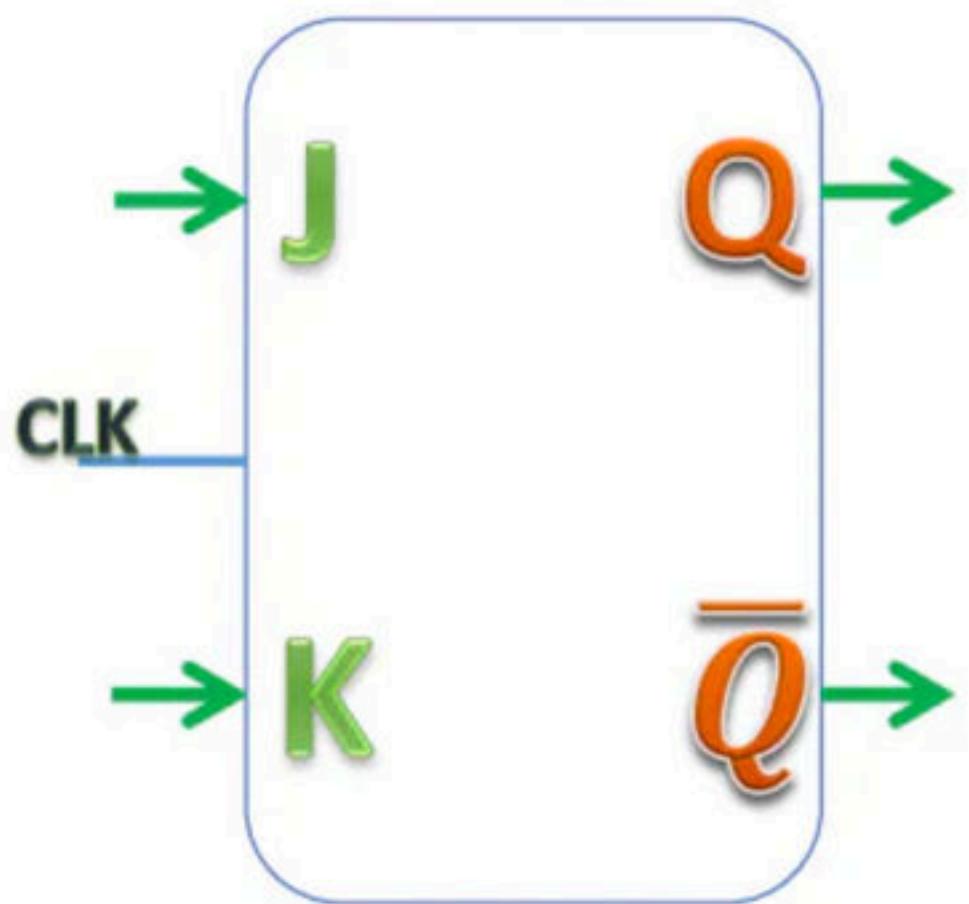
Because of presence of invalid state there is a restriction on the sequence of the applied input ,since it leads to **CRITICAL RACE**, which is undesirable .

J K Flip Flop



CLK	J	K	Q^+
0	x	x	
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

J K Flip Flop



CLK	J	K	Q+	State
0	x	x		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Characteristic table

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

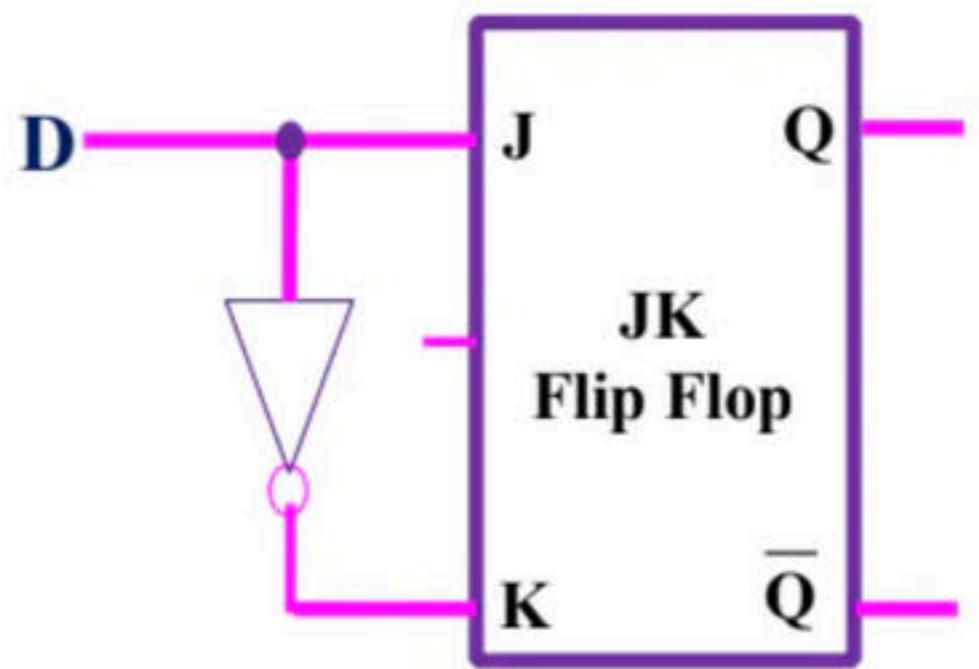
Characteristic Equation

Excitation table

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

State Diagram

D Flip Flop



CLK	D	Q+

Characteristic table

CLK	D	Q	Q+

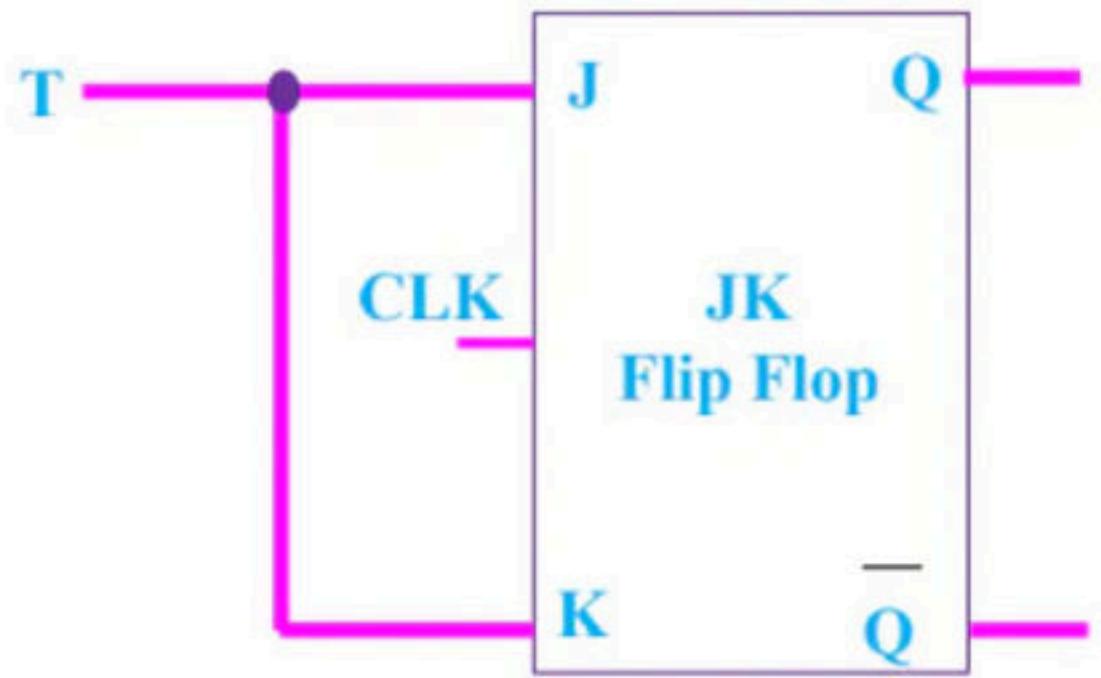
Characteristic Equation

Excitation table

Q	Q+	D

State Diagram

T Flip Flop



CLK	T	Q_+

Characteristic table

CLK	T	Q	Q+

Characteristic Equation

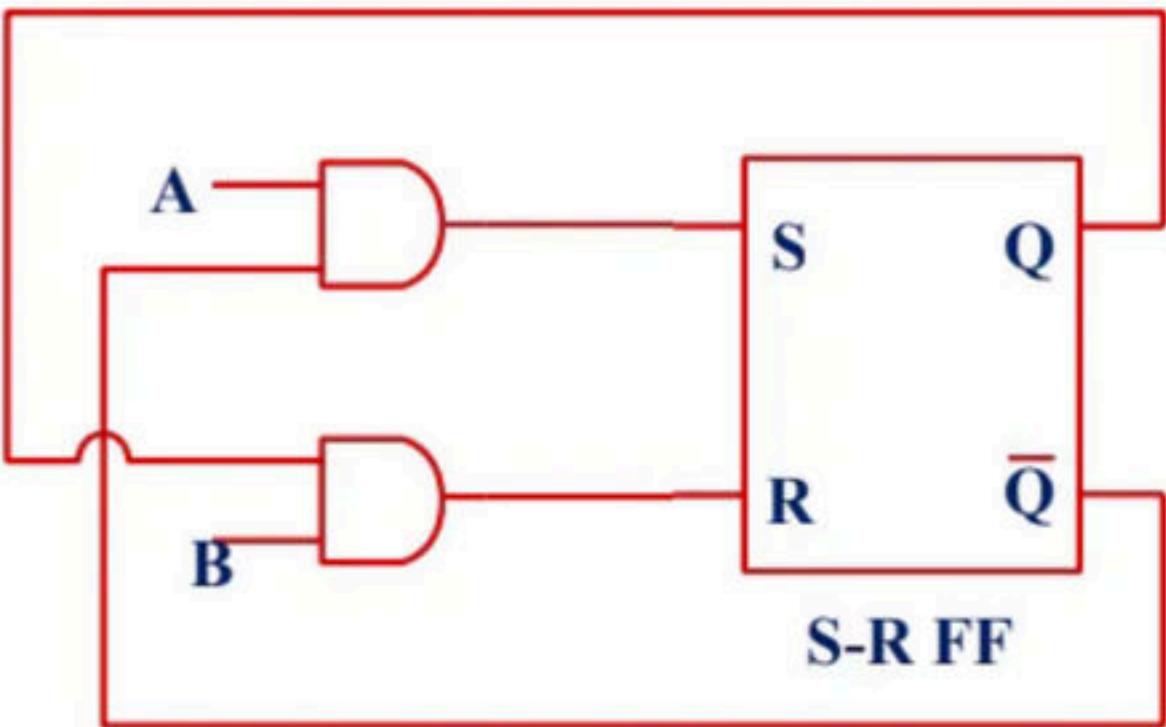
Excitation table

Q	Q+	D

State Diagram

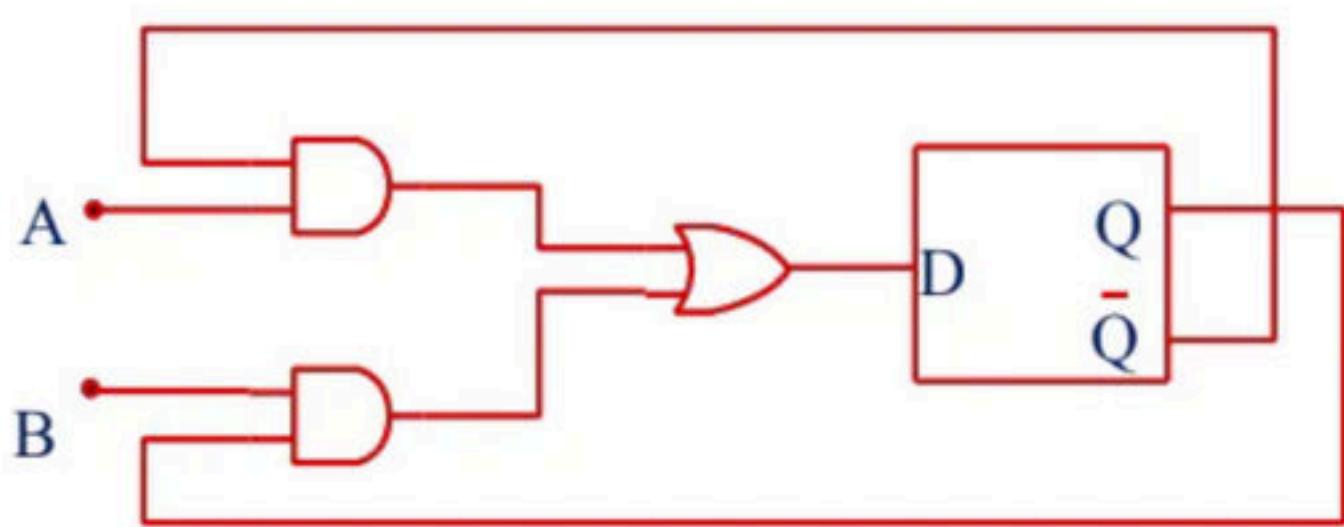
Q. The two inputs A and B are connected to an R-S FF via two AND gates as shown in the figure. If A = 1 and B = 0, the output $Q\bar{Q}$ is

- (A) 00
- (B) 10
- (C) 01
- (D) 11



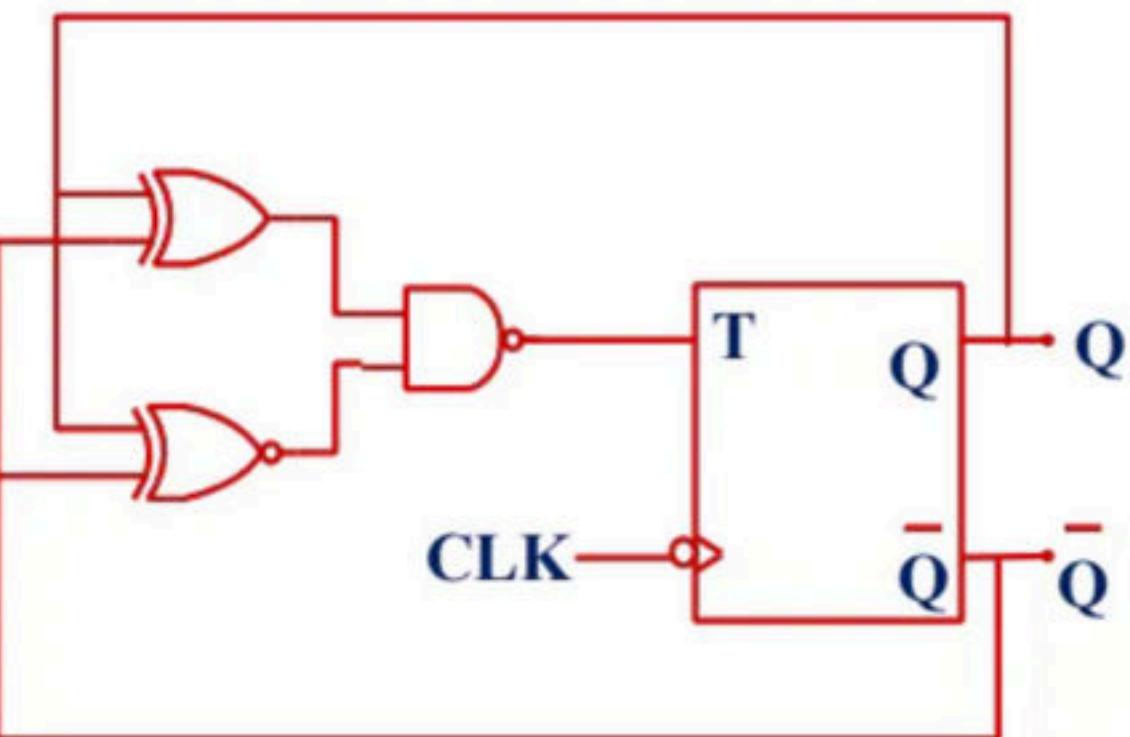
Q. What is represented by the digital circuit given above?

- (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 $B=\bar{K}$
- (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



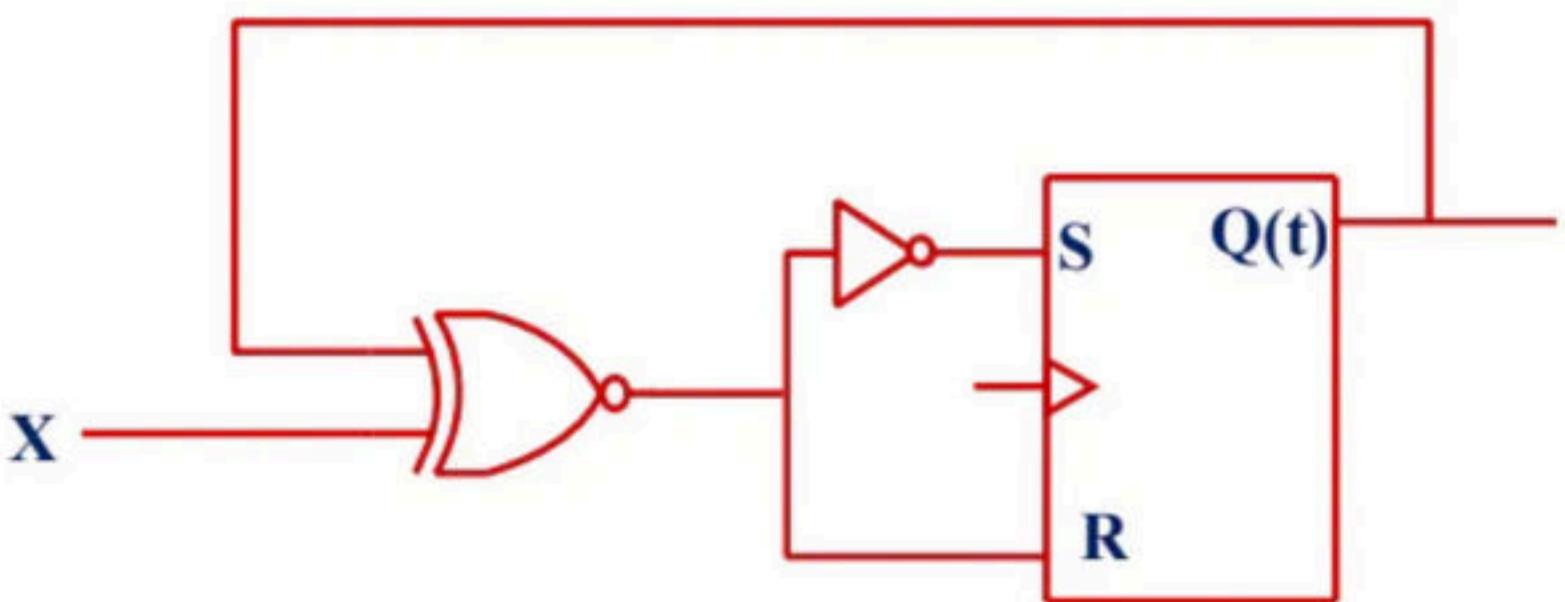
Q. Consider the circuit shown in the figure. The expression for the next state $Q(t+1)$ is

(a) $xQ(t)$

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

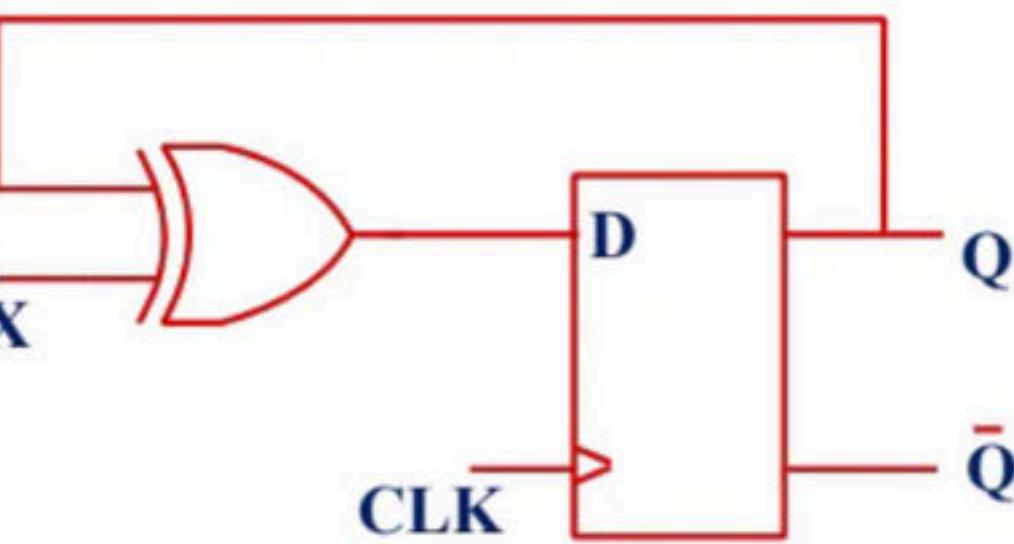
(c) $xQ(t)$

(d) $x \odot Q(t)$



Q. The digital circuit shown in figure works as a

- (A) JK flip-flop
- (B) Clocked RS flip-flop
- (C) T flip-flop
- (D) Ring counter



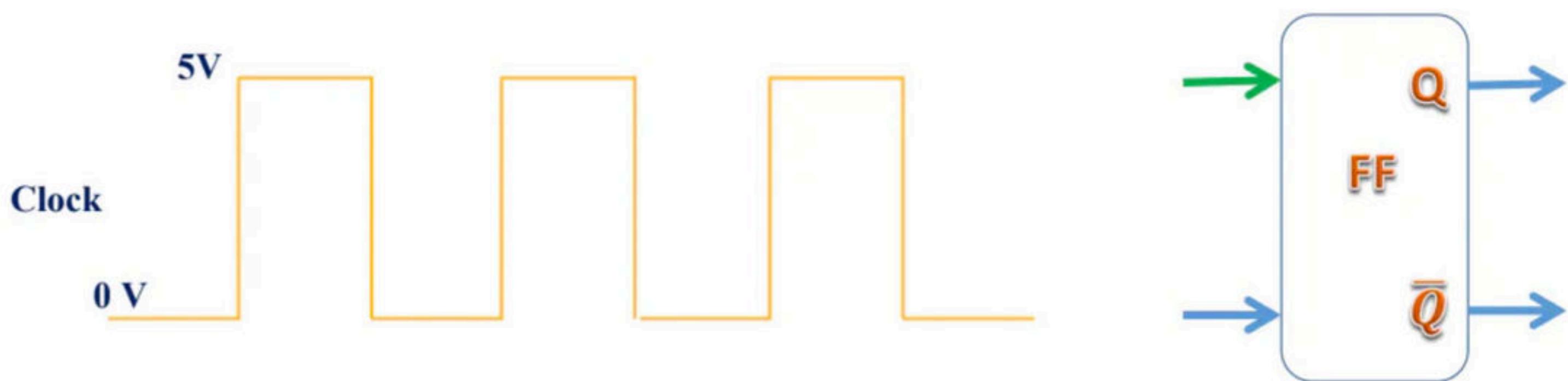
Triggering

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.

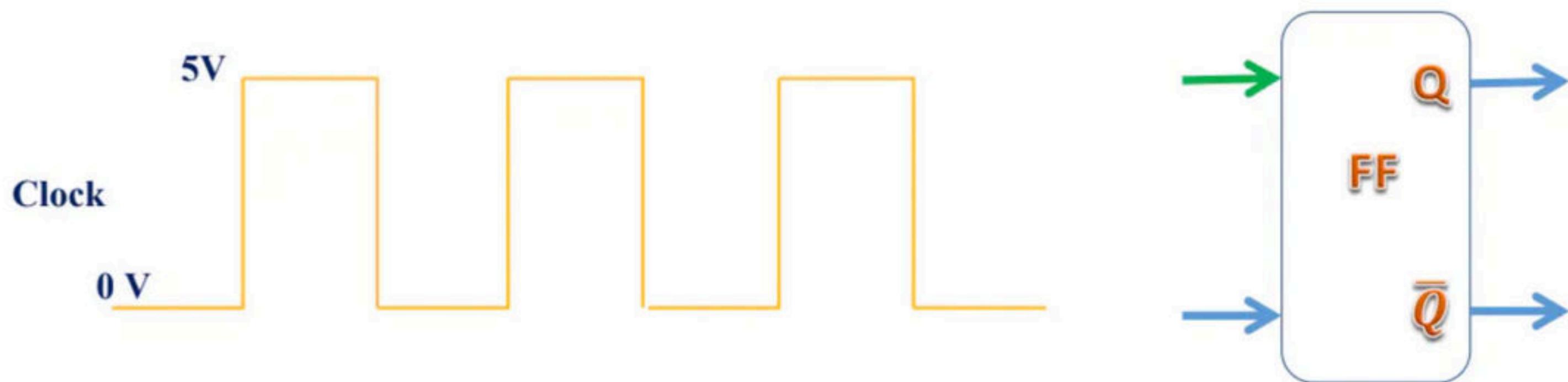
There are 4- types of triggering the flip flops .

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

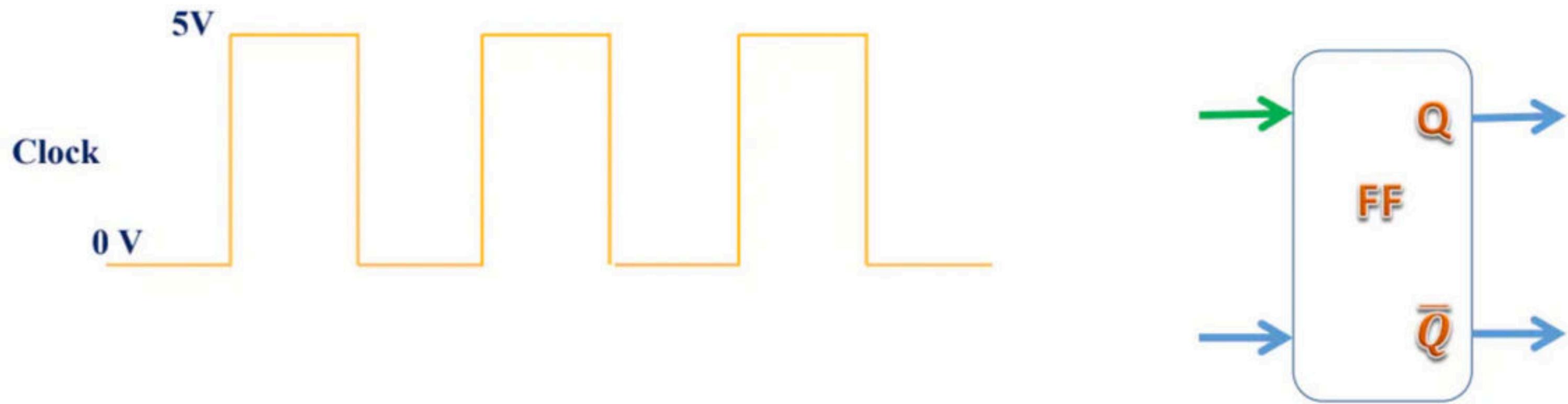
1. High (Positive) Level Triggering



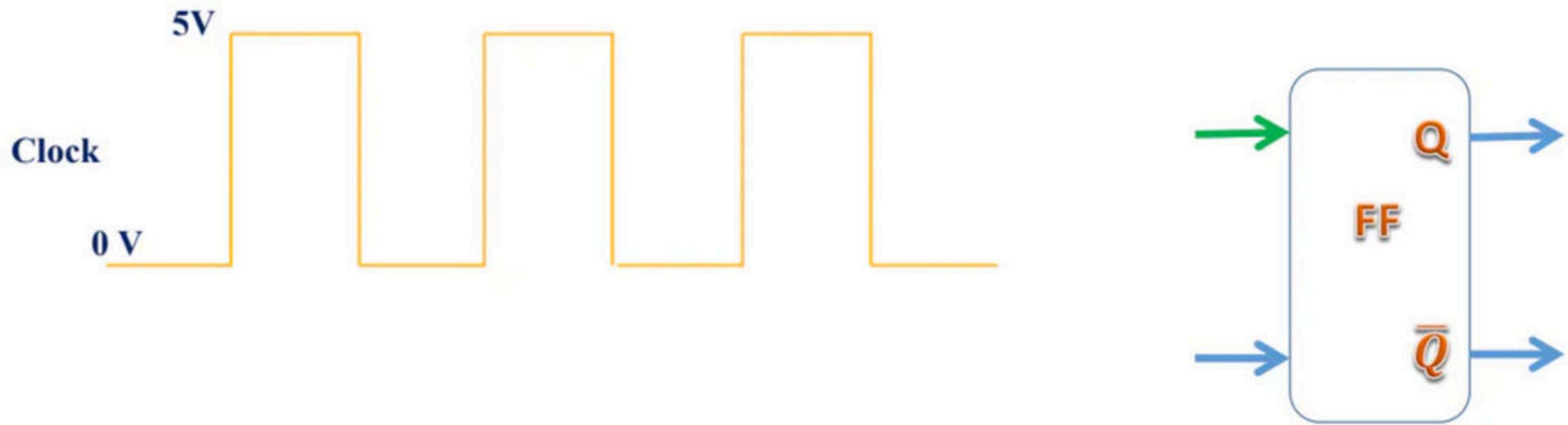
2. Low (Negative) Level Triggering



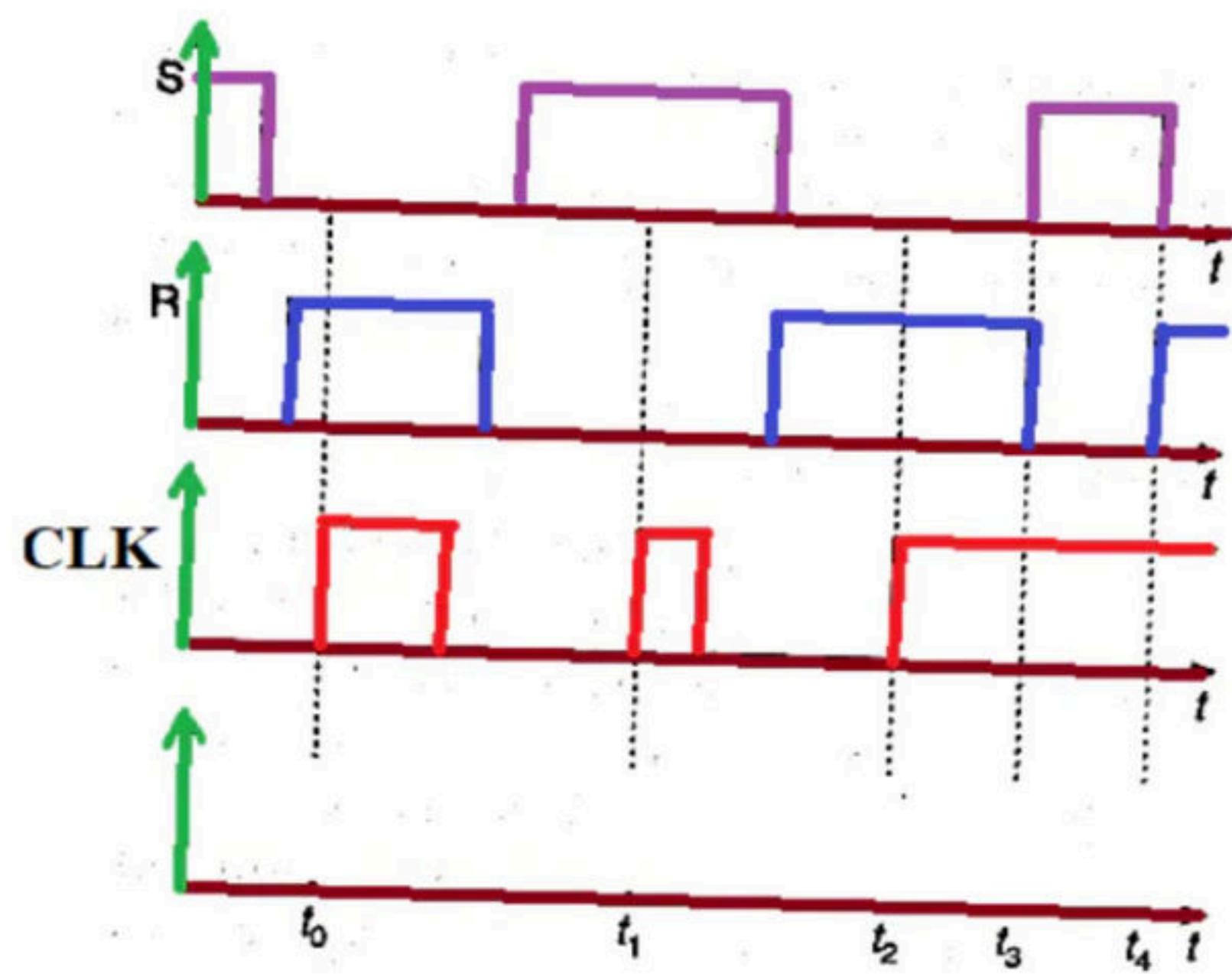
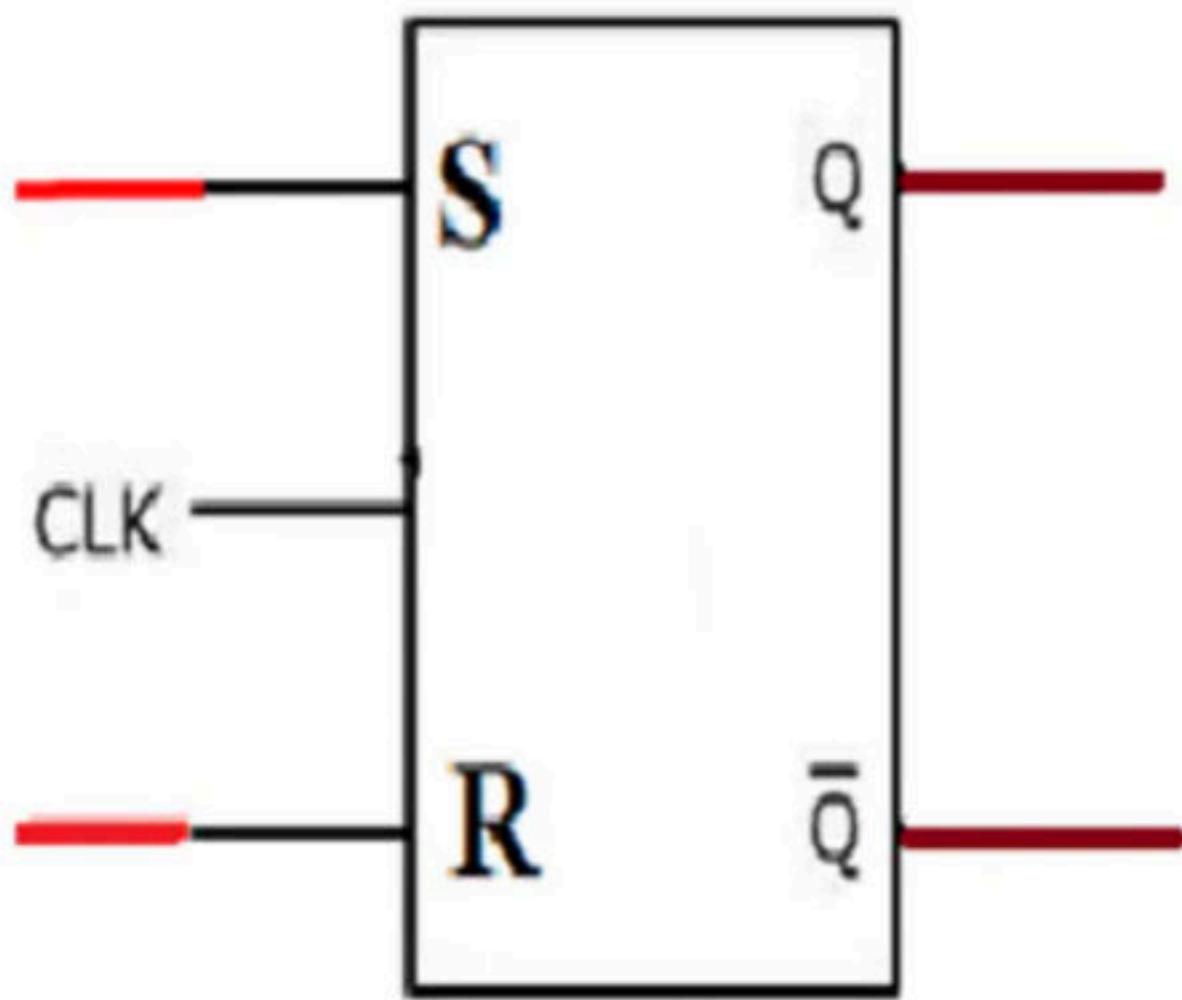
3. Positive Edge triggering



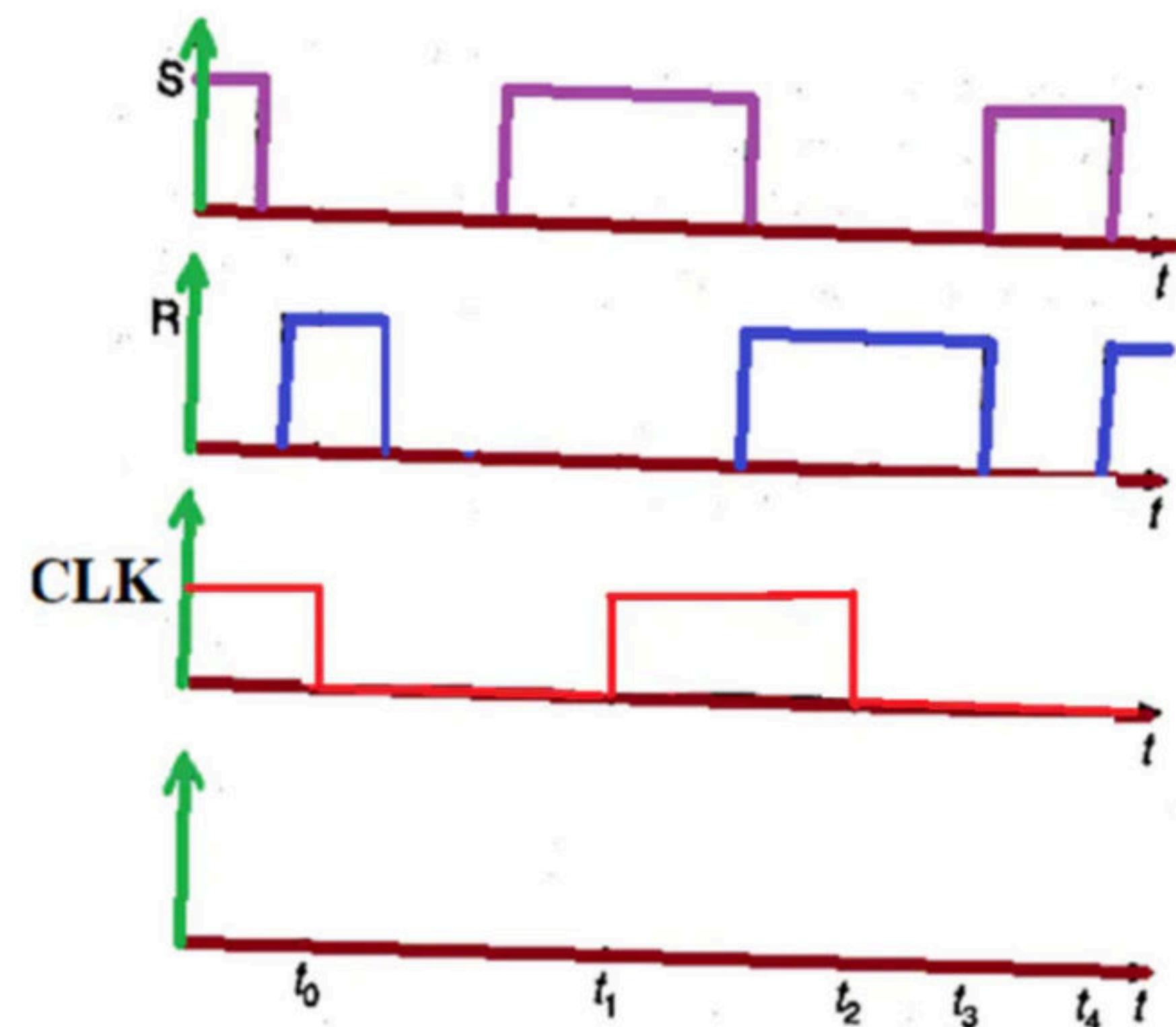
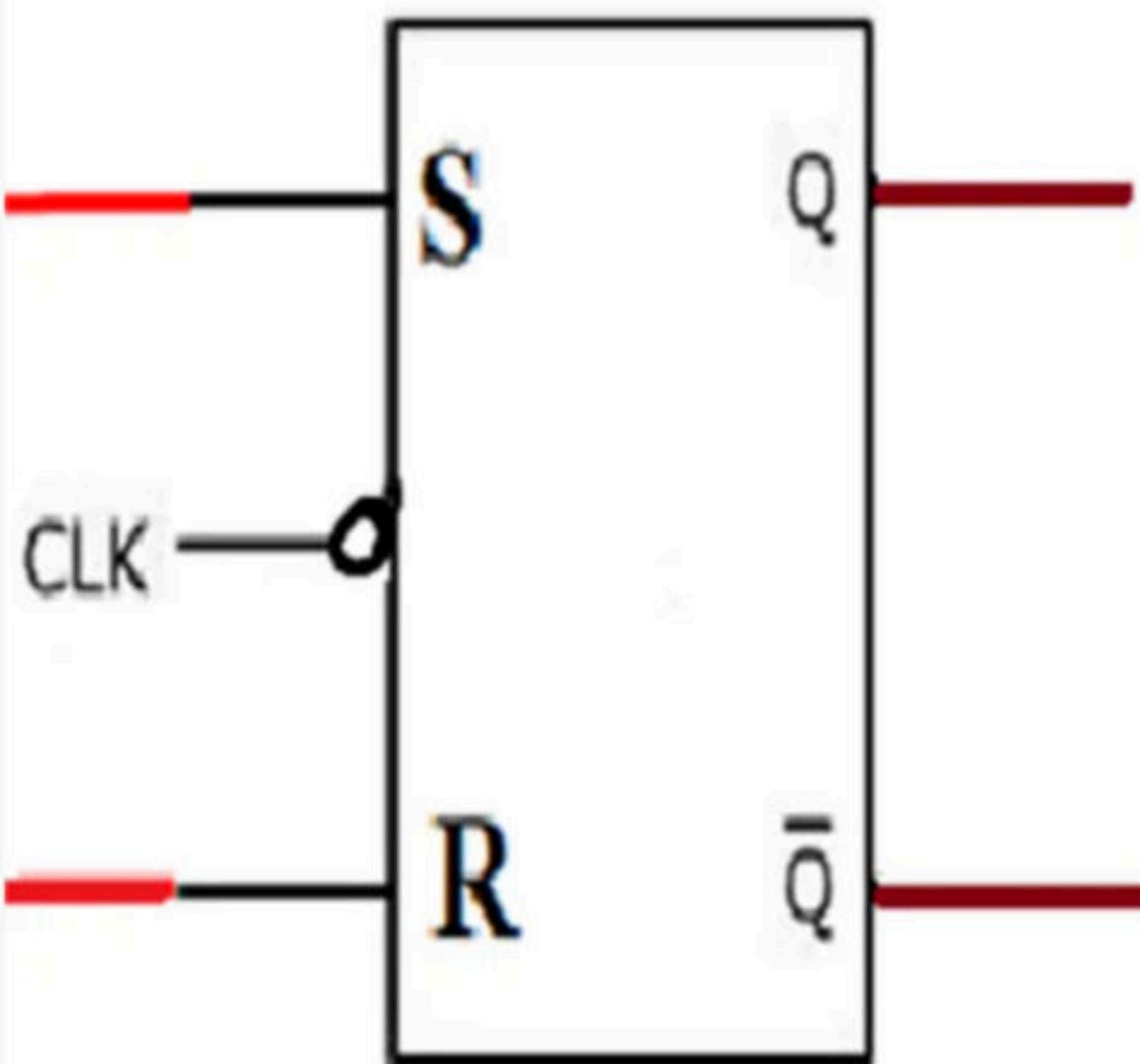
4. Negative Edge triggering



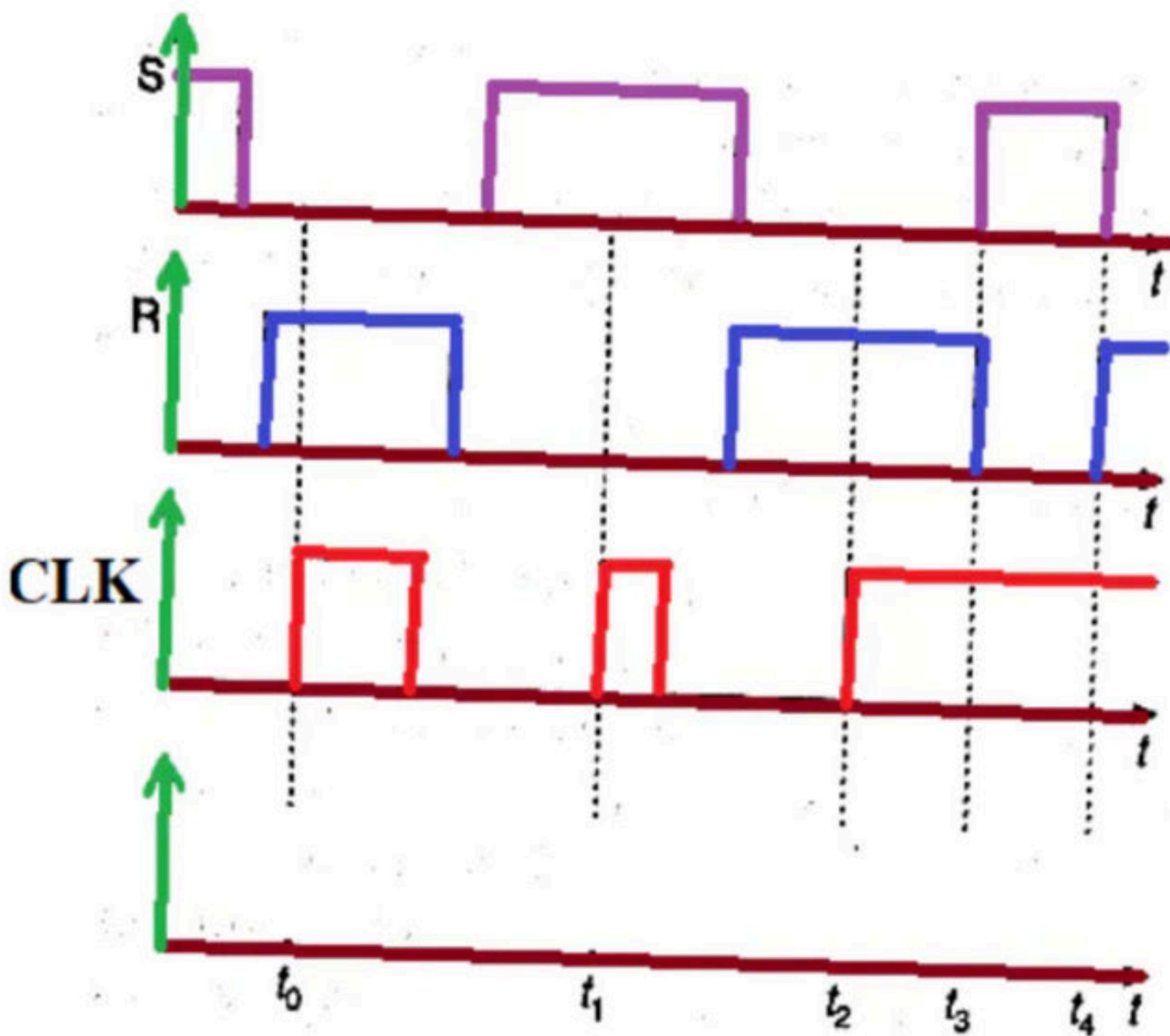
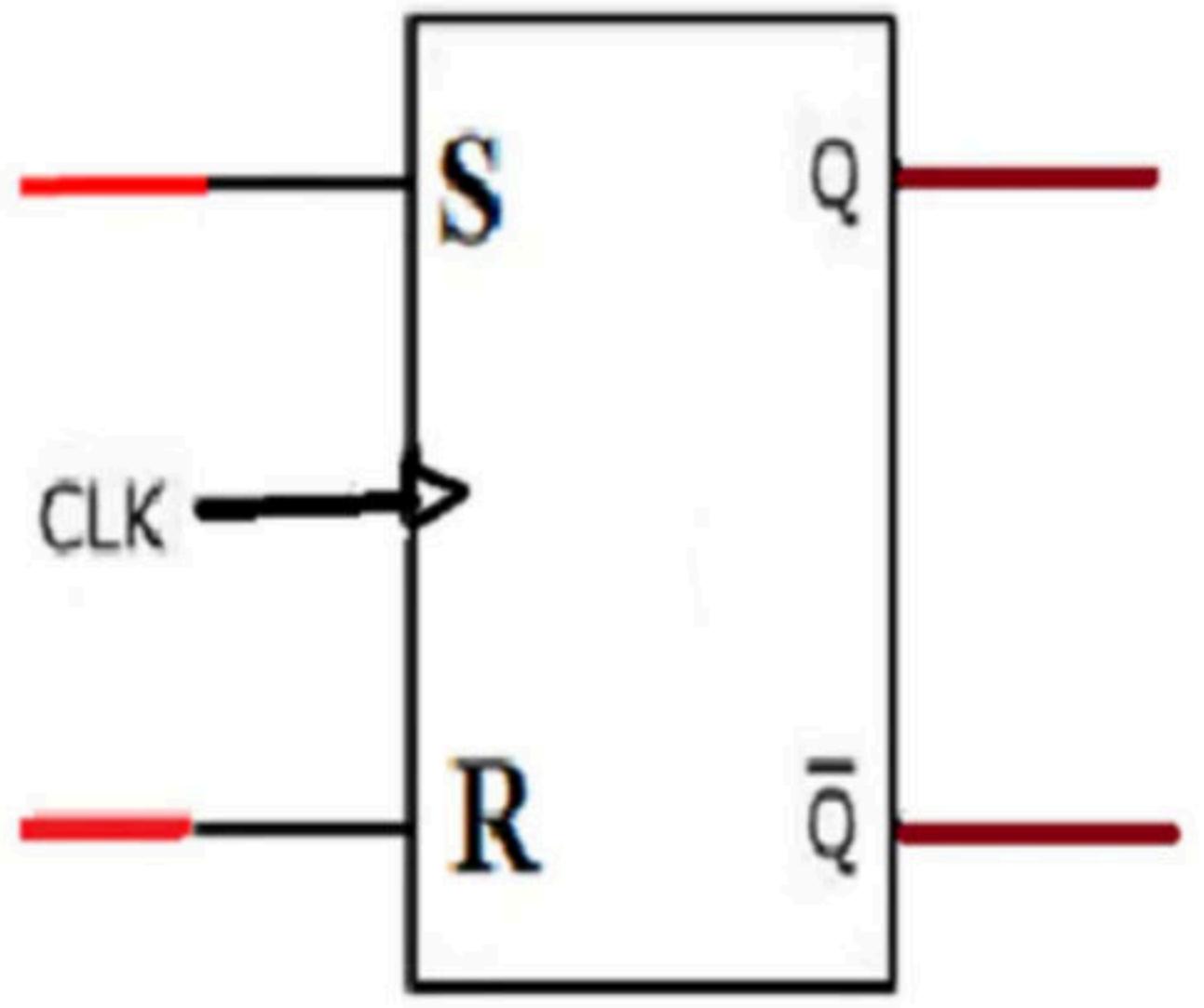
Draw the output waveform



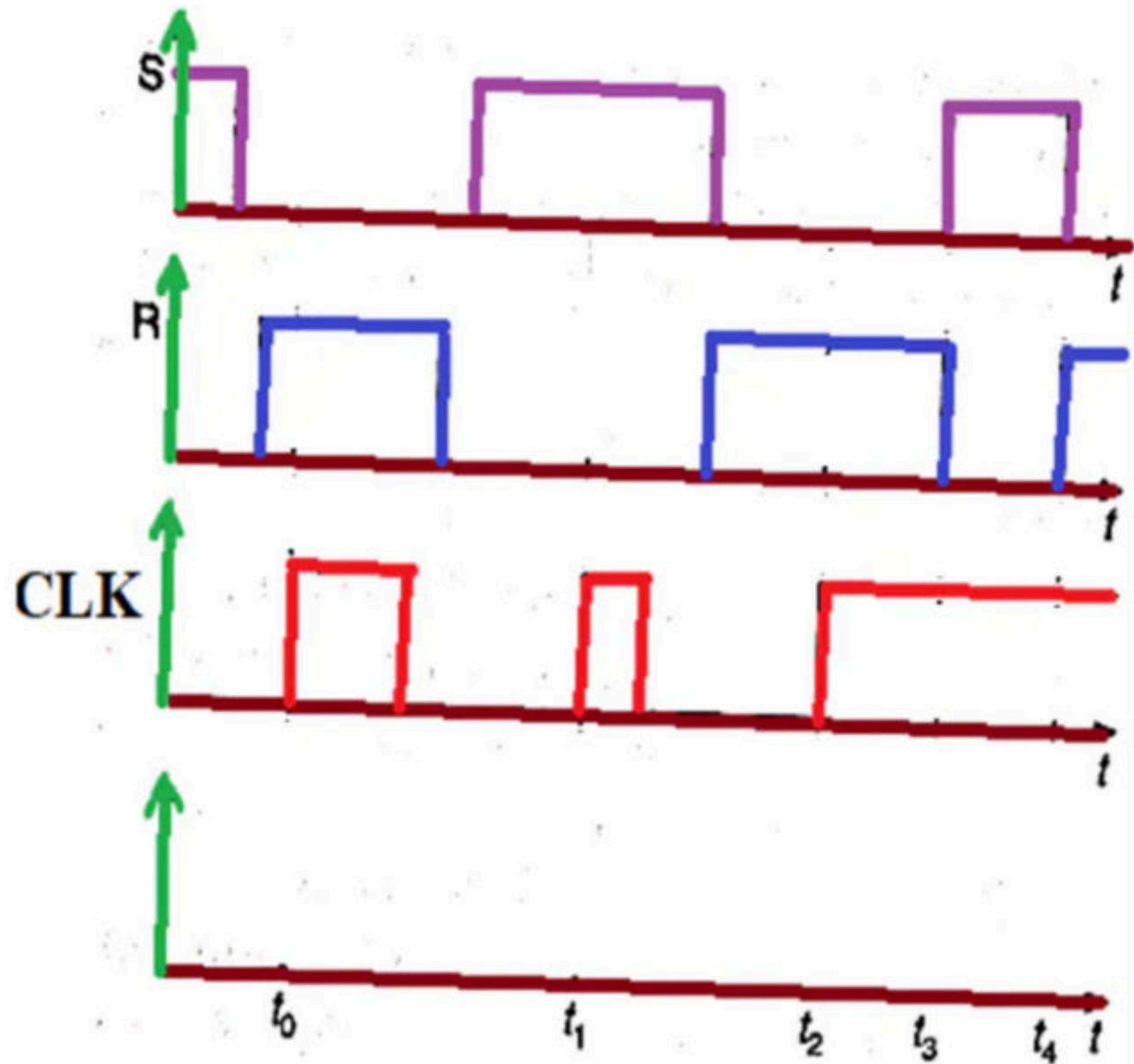
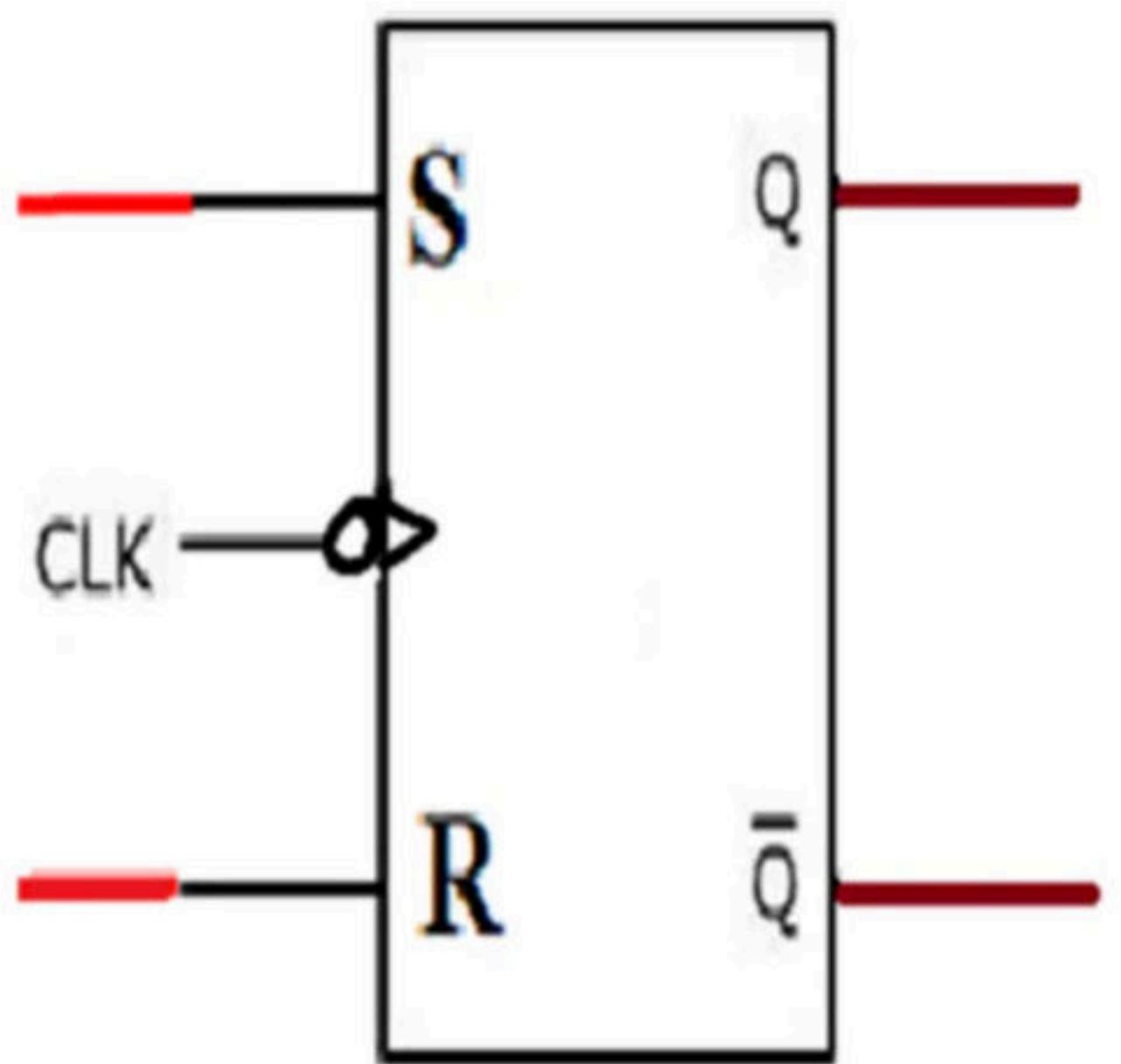
Draw the output wave form



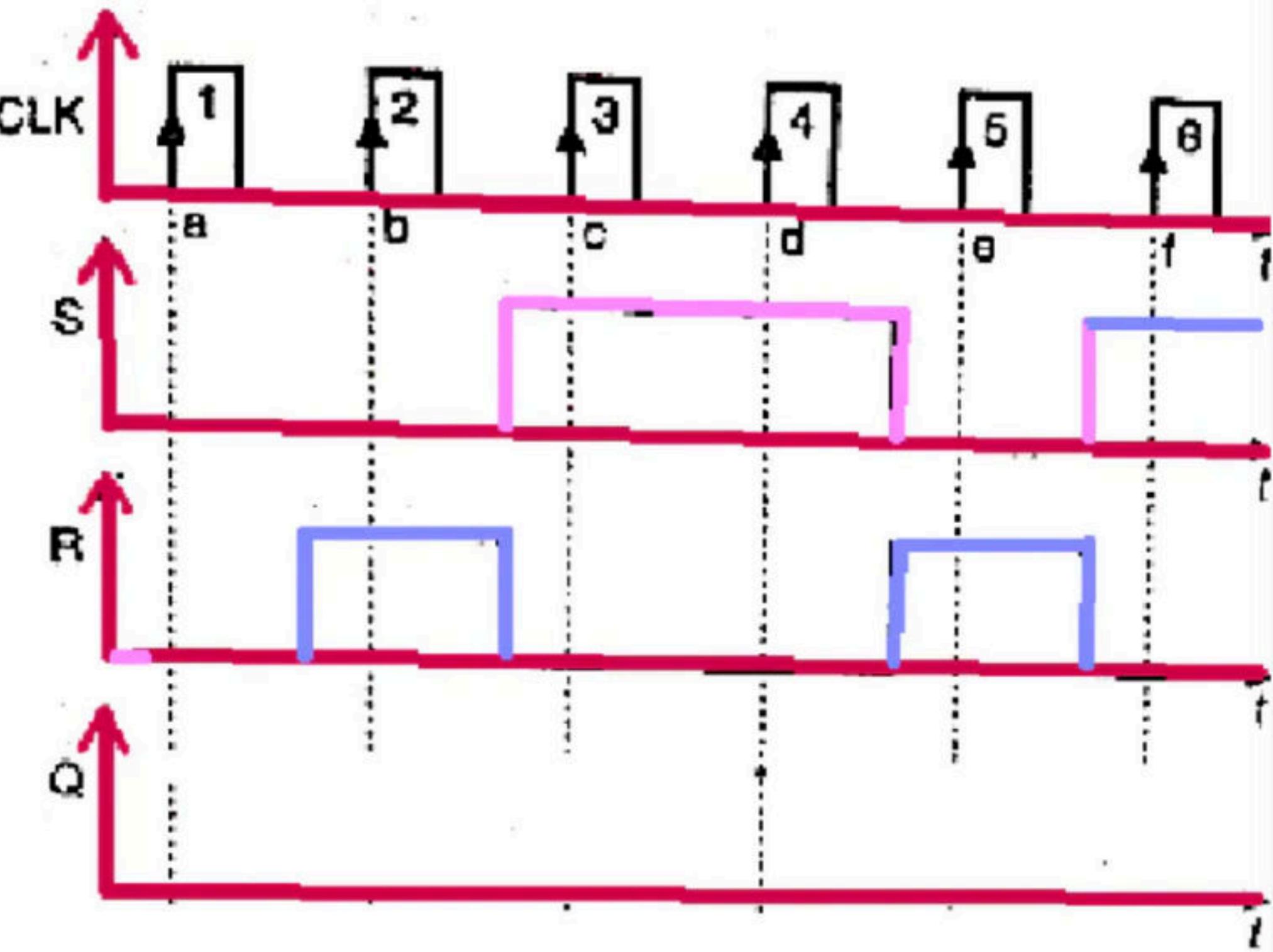
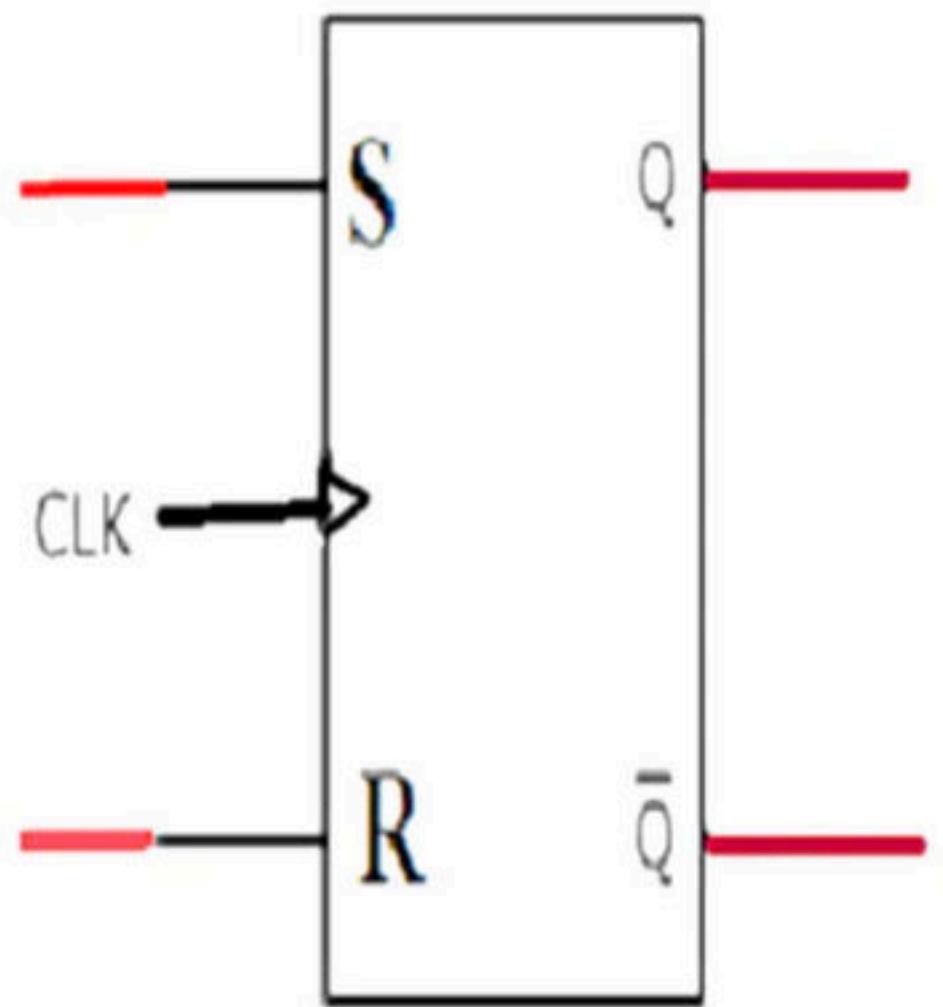
Draw the output wave form



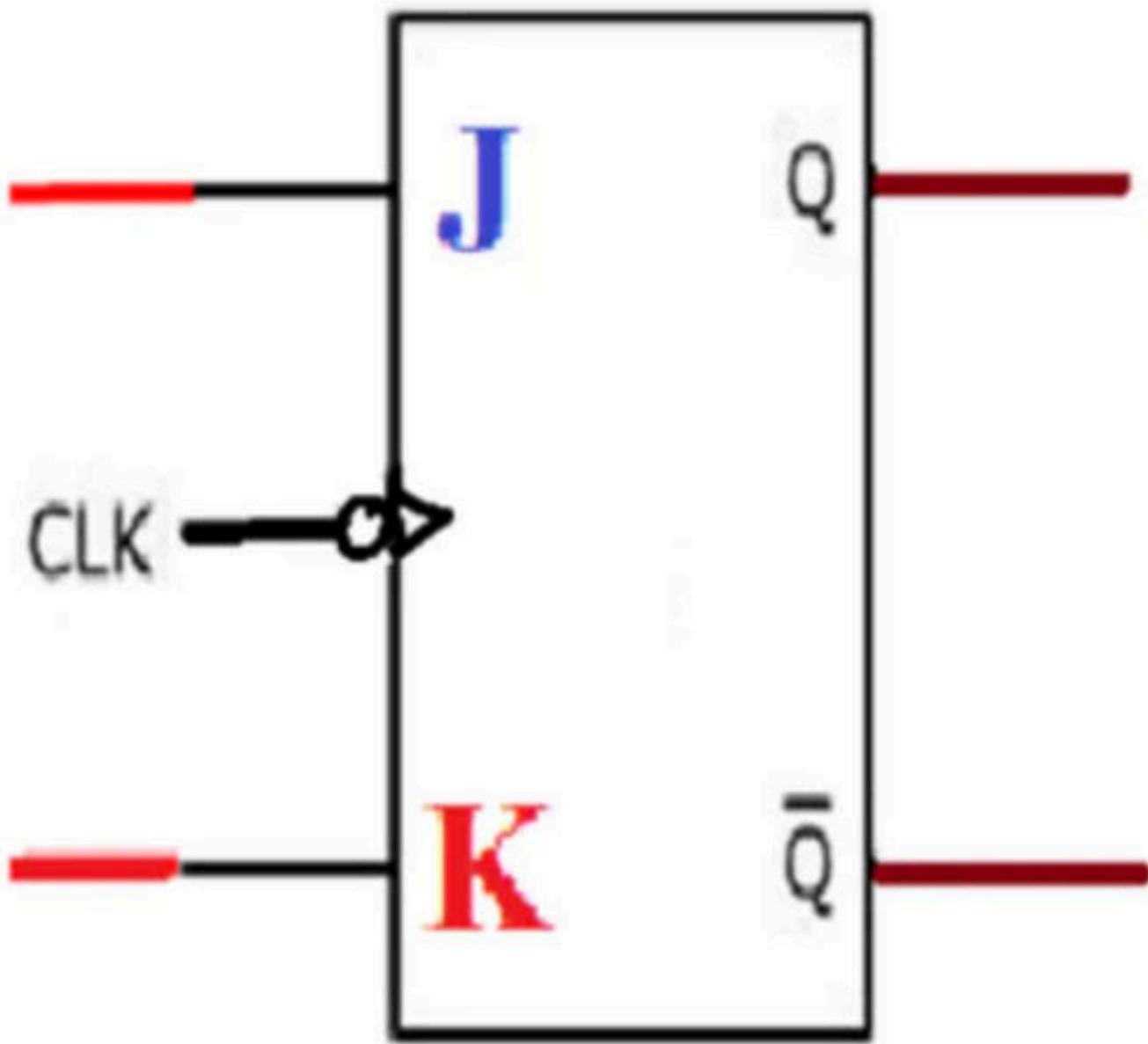
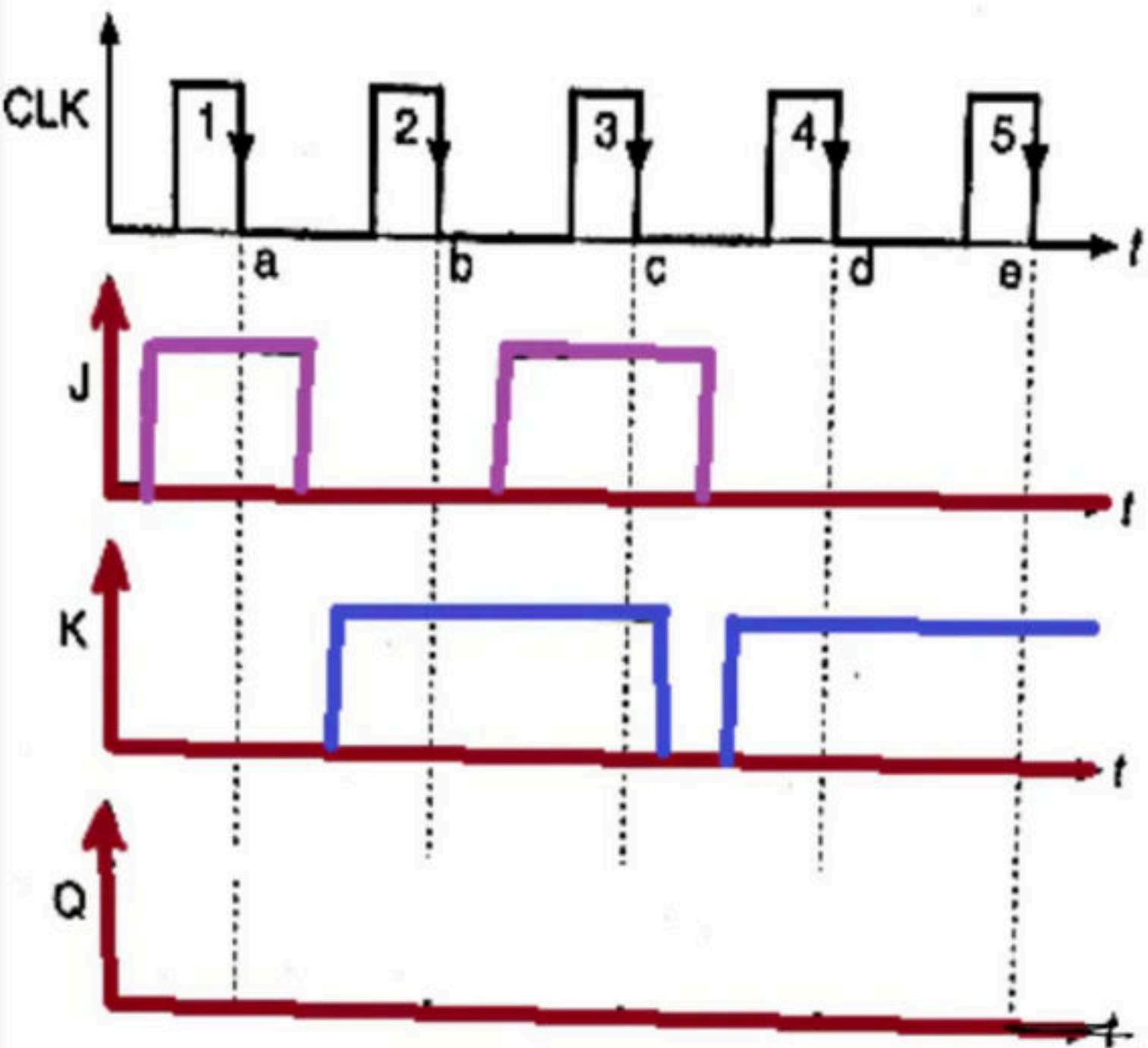
Draw the output wave form



Draw the output wave form

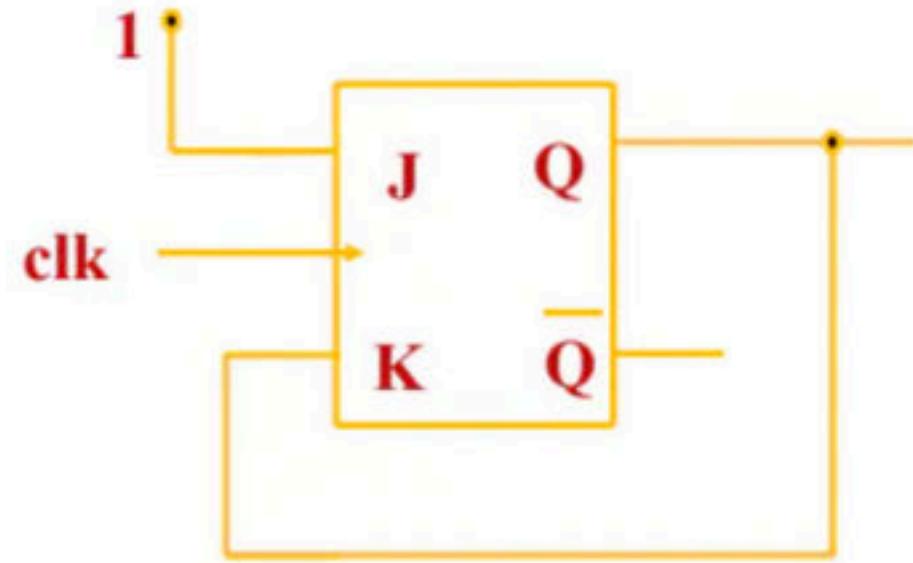


Draw the output waveform



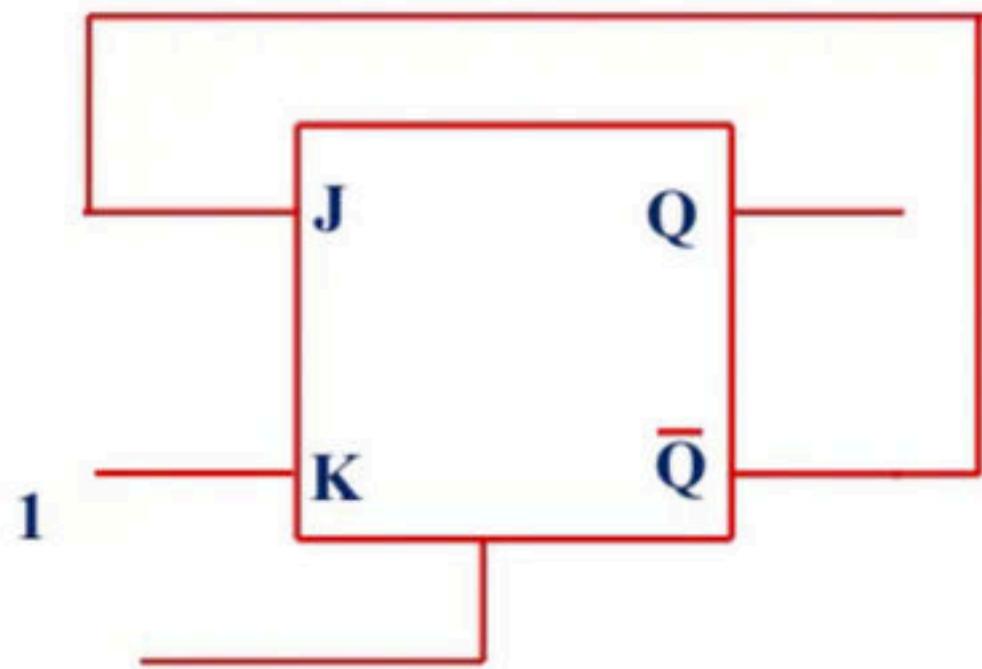
Q. In the circuit shown in the figure, $Q=0$, initially. When clock pulses are applied, the subsequent states of 'Q' will be.

- (a) 1, 0, 1, 0, 1,.....
- (b) 0, 0, 0, 0,.....
- (c) 1, 1, 1, 1,.....
- (d) 0, 1, 0, 1,.....



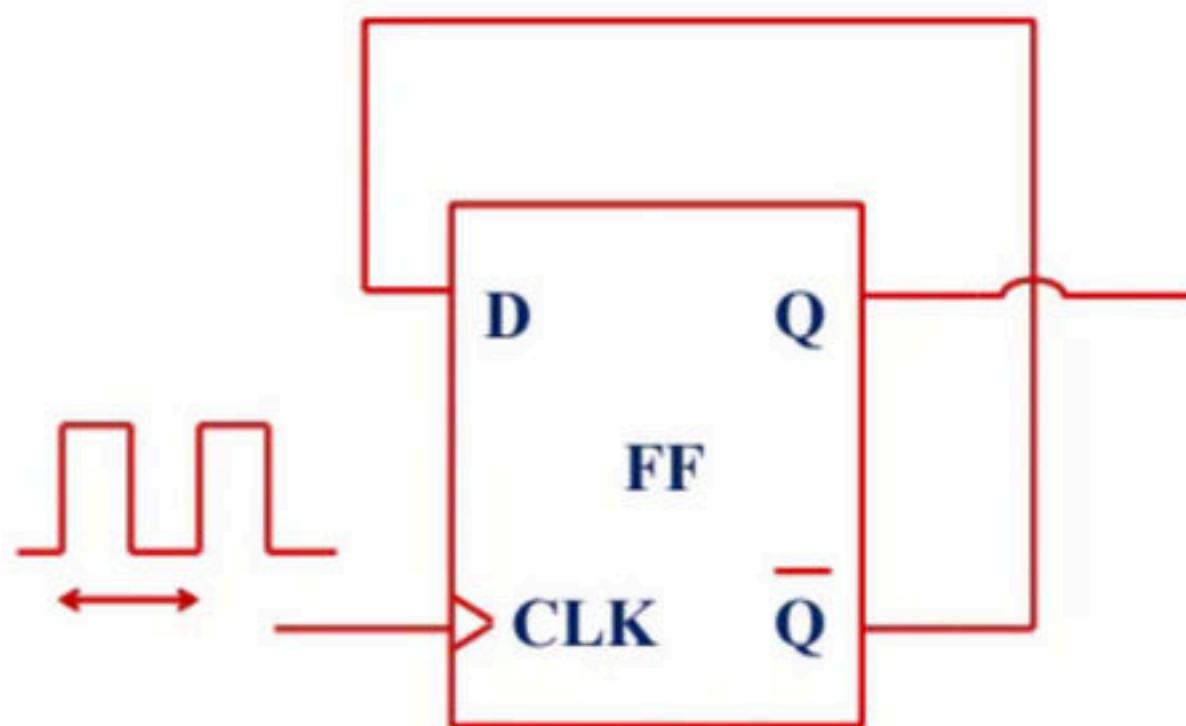
Q. Consider the following J-K flip-flop. In the above J-K flip-flop, $J = \bar{Q}$ and $K = 1$. Assume that the flip-flop was initially cleared and then clocked for 6 pulses. What is the sequence at the Q output?

- (a) 01000
- (b) 011001
- (c) 010010
- (d) 010101

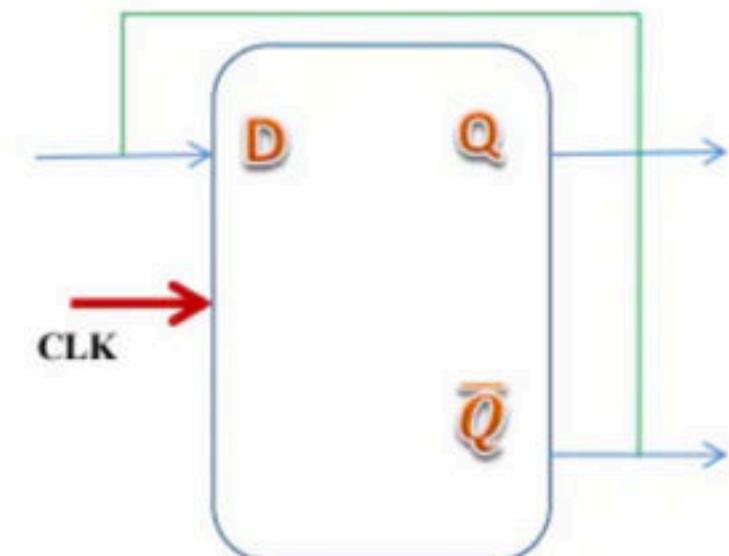
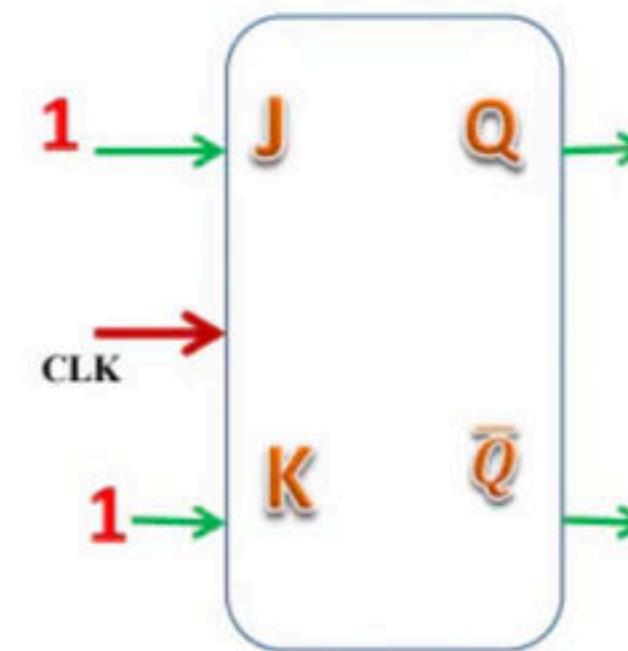
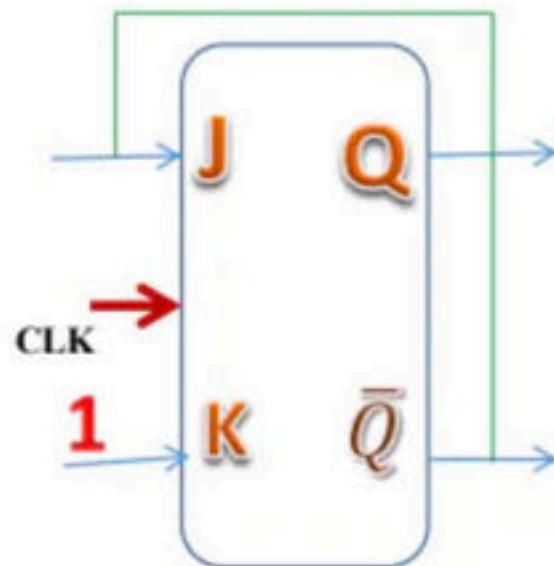
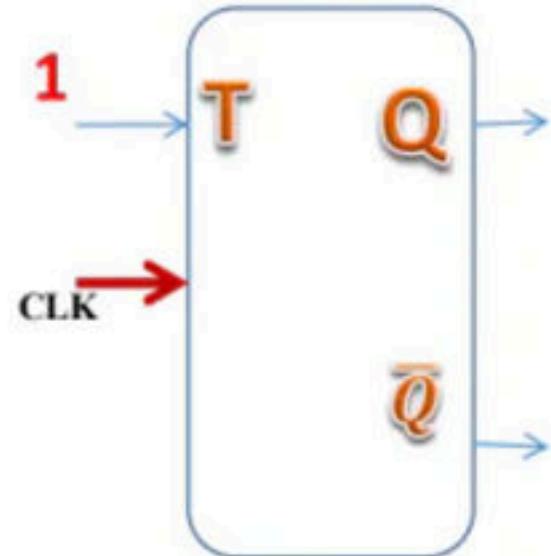
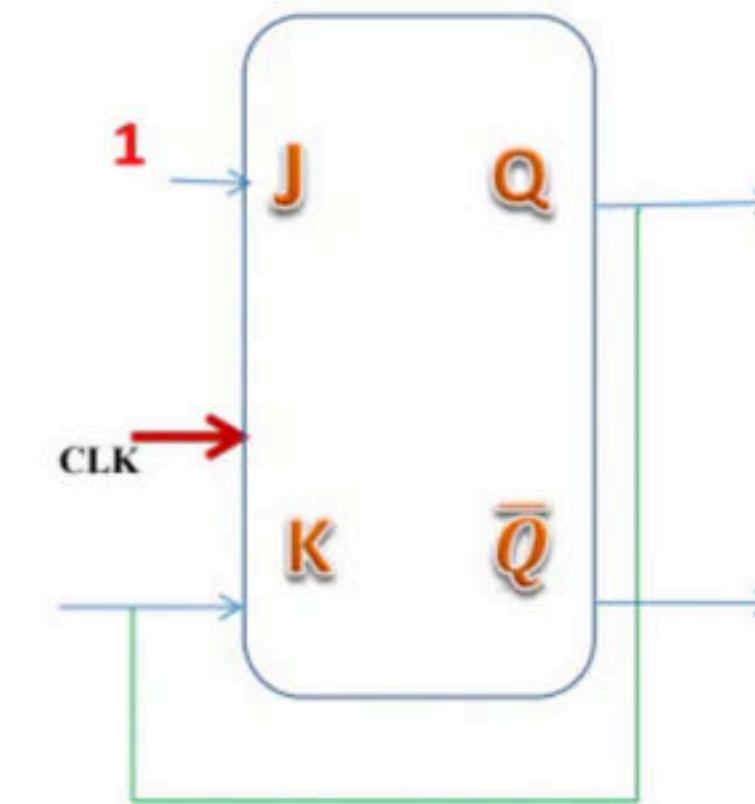
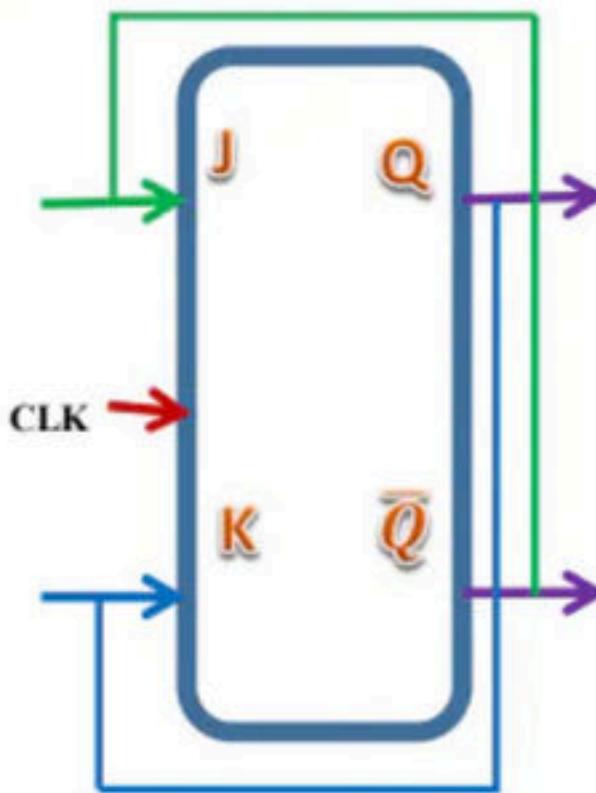
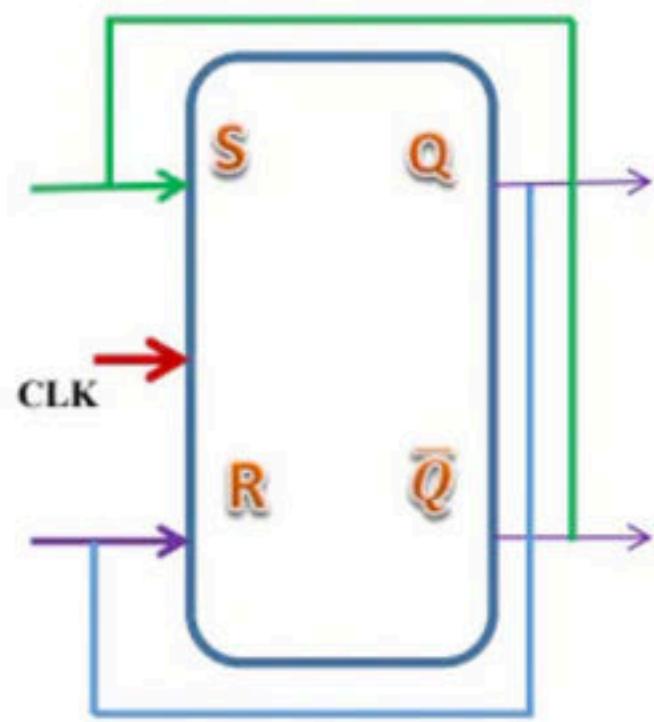


Q. The frequency of the clock signal applied to the rising edge triggered D flip-flop shown in figure is 10kHz. The frequency of the signal available at Q is.

- (A) 10 kHz
- (B) 2.5 kHz
- (C) 20 kHz
- (D) 5 kHz



Toggle Modes

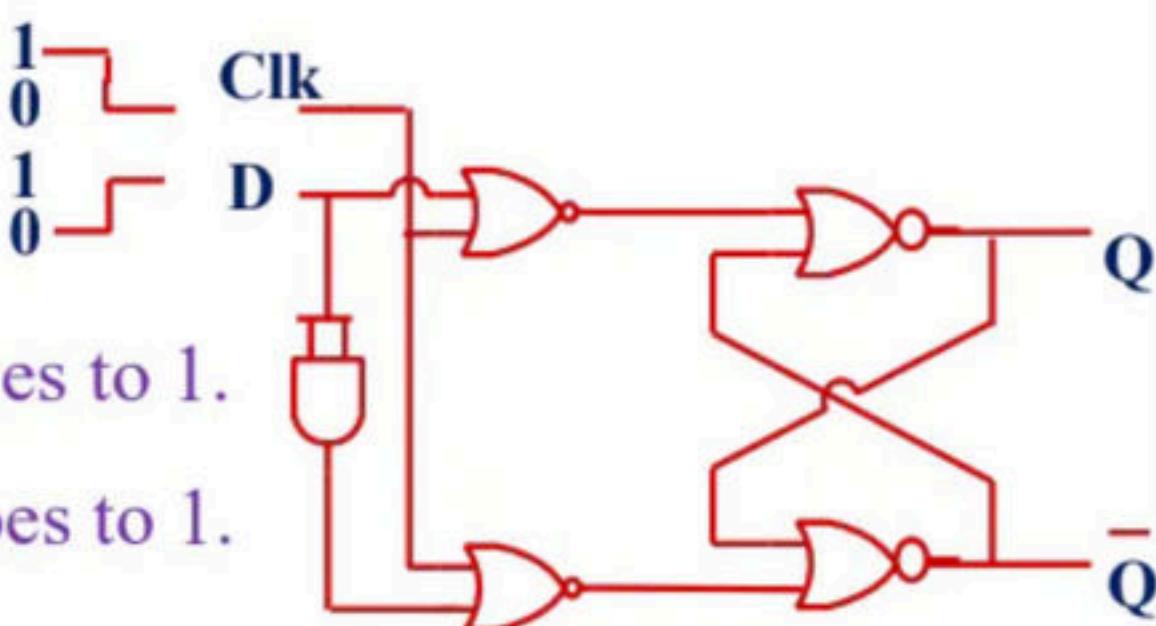


Q. The output Q_n of a J-K flip-flop is zero. It changes to 1 when a clock pulse is applied. The input J_n and K_n are respectively (\times represents don't care condition):

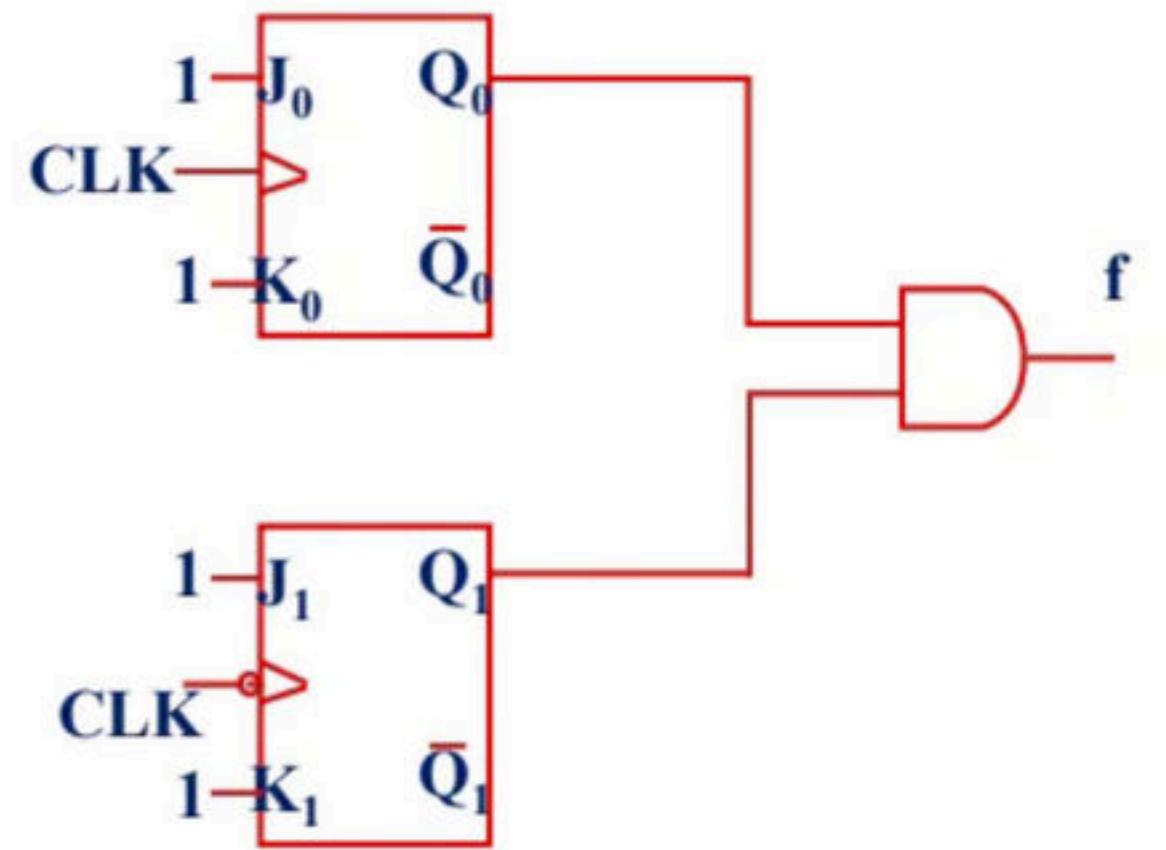
- (a) 1 and \times
- (b) 0 and \times
- (c) \times and 0
- (d) \times and 1

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.

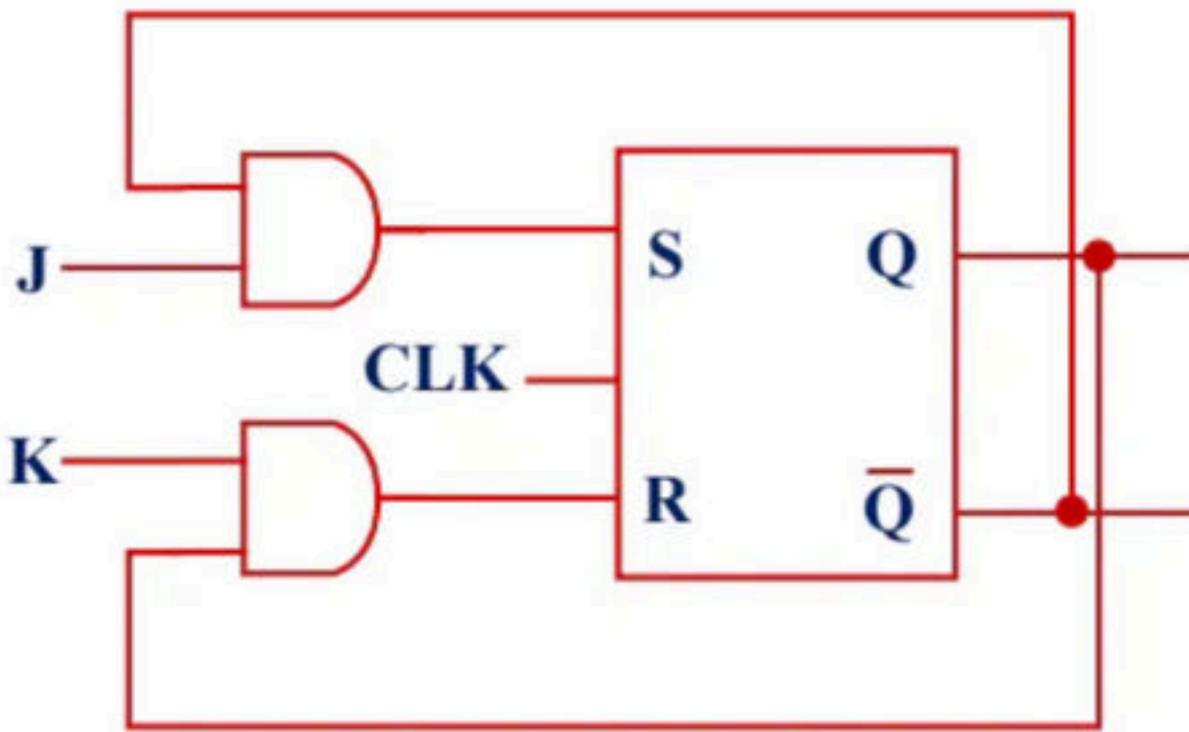


Q) Find the frequency and duty cycle of output , if the clock frequency is 10MHz



Race Around Condition

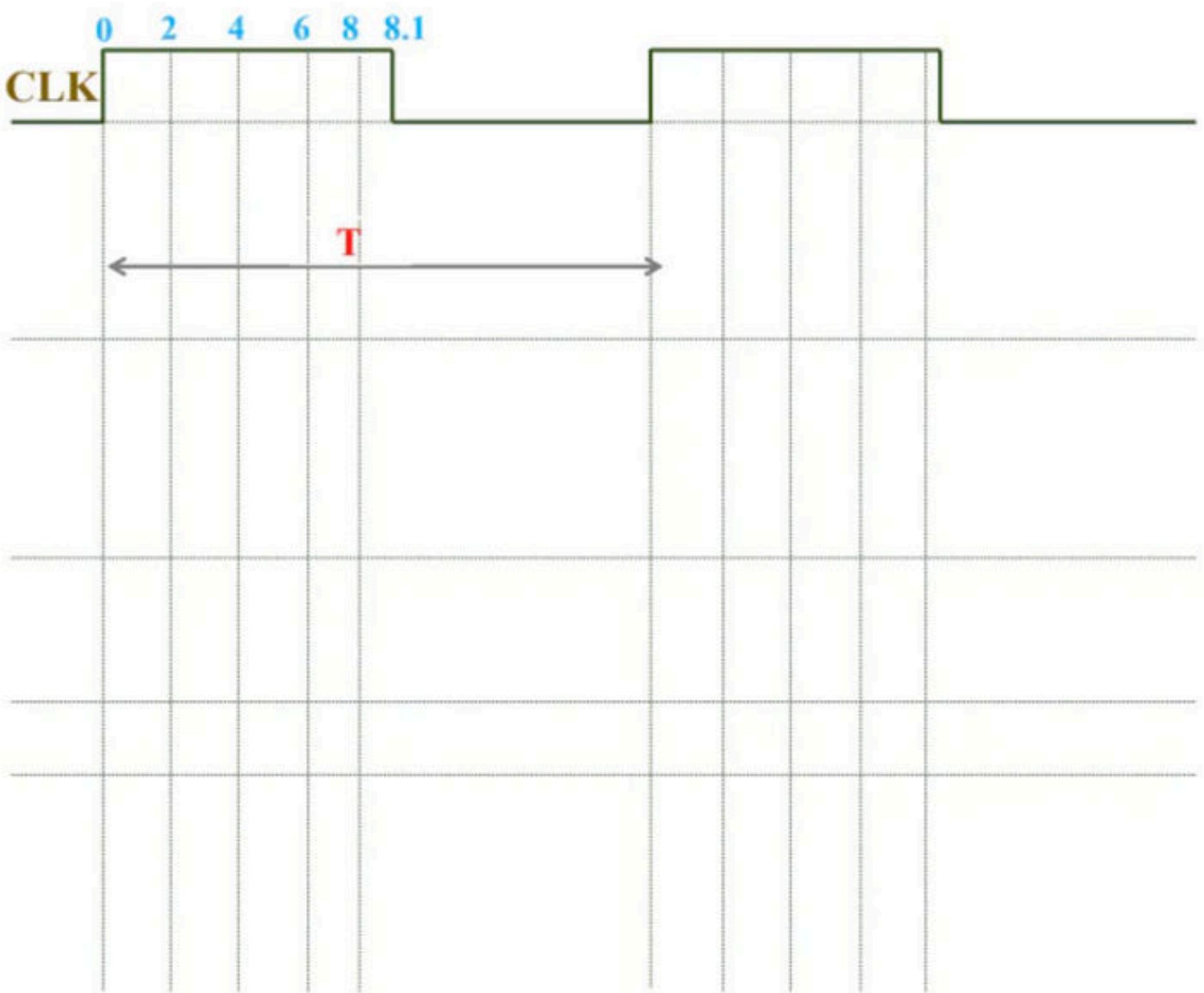
Race around condition



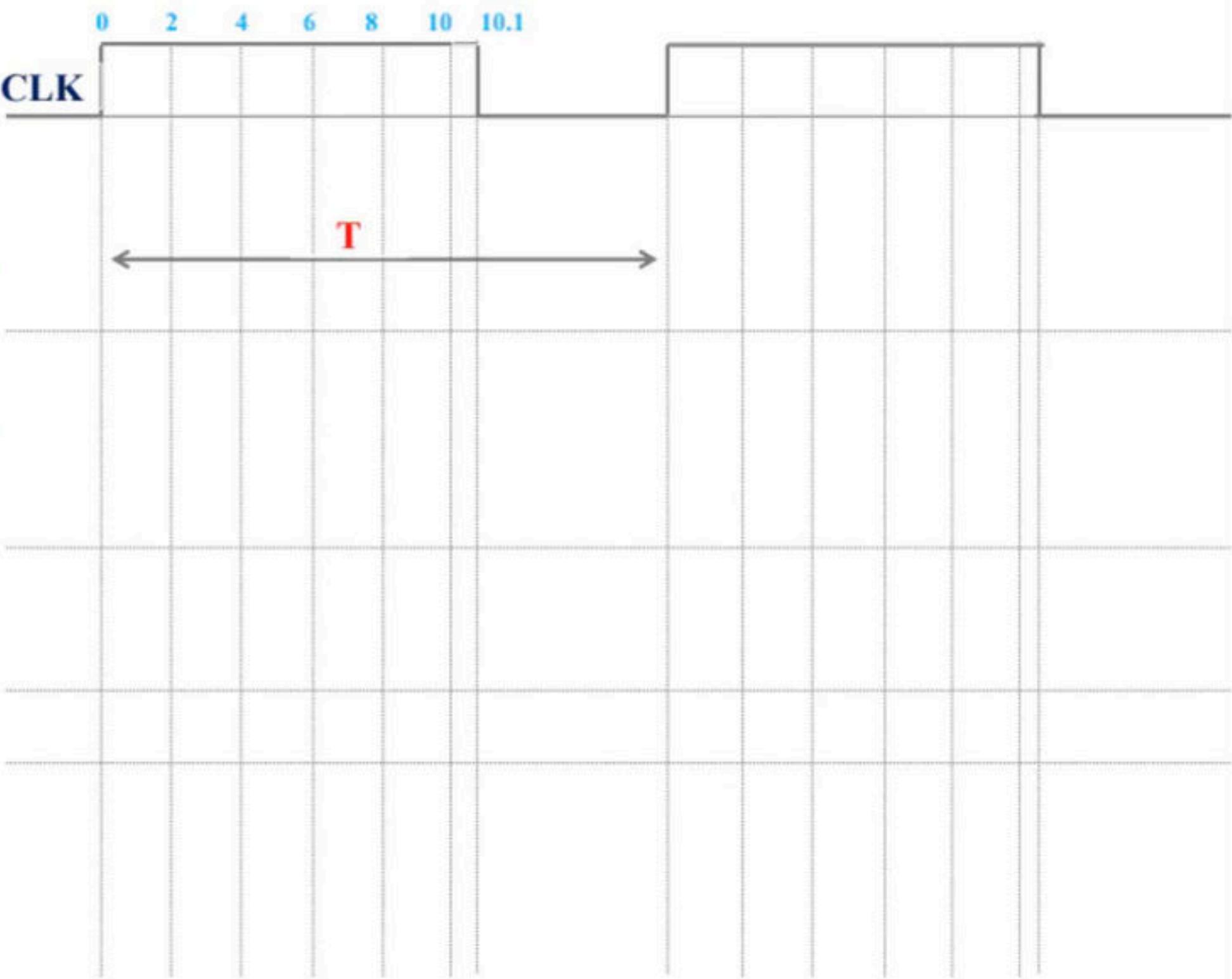
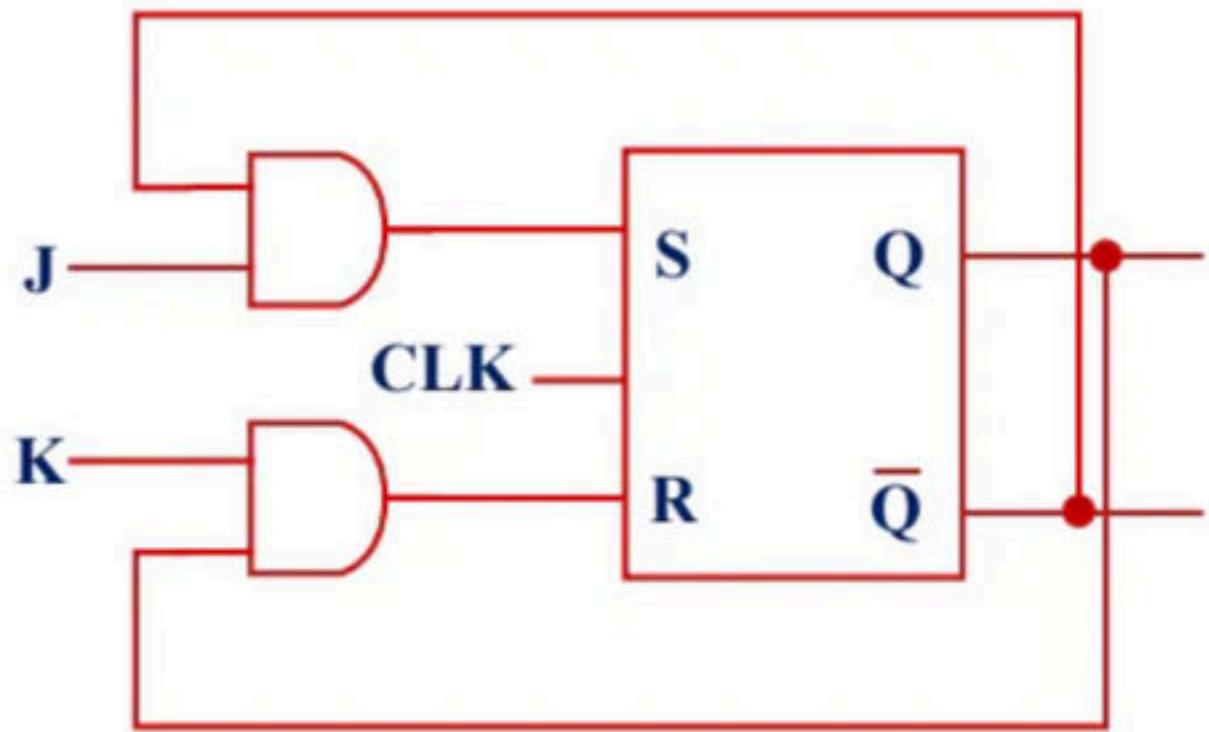
Case - 1

$$T_{CLK} = 16.2\text{ns}$$

$$t_{pd} = 2\text{ns}$$



Race around condition



Case - 2

$$T_{CLK} = 20.2\text{ns}$$

$$t_{pd} = 2\text{ns}$$

The output of the FF changes to $0 \rightarrow 1 \rightarrow 0 \dots$ 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

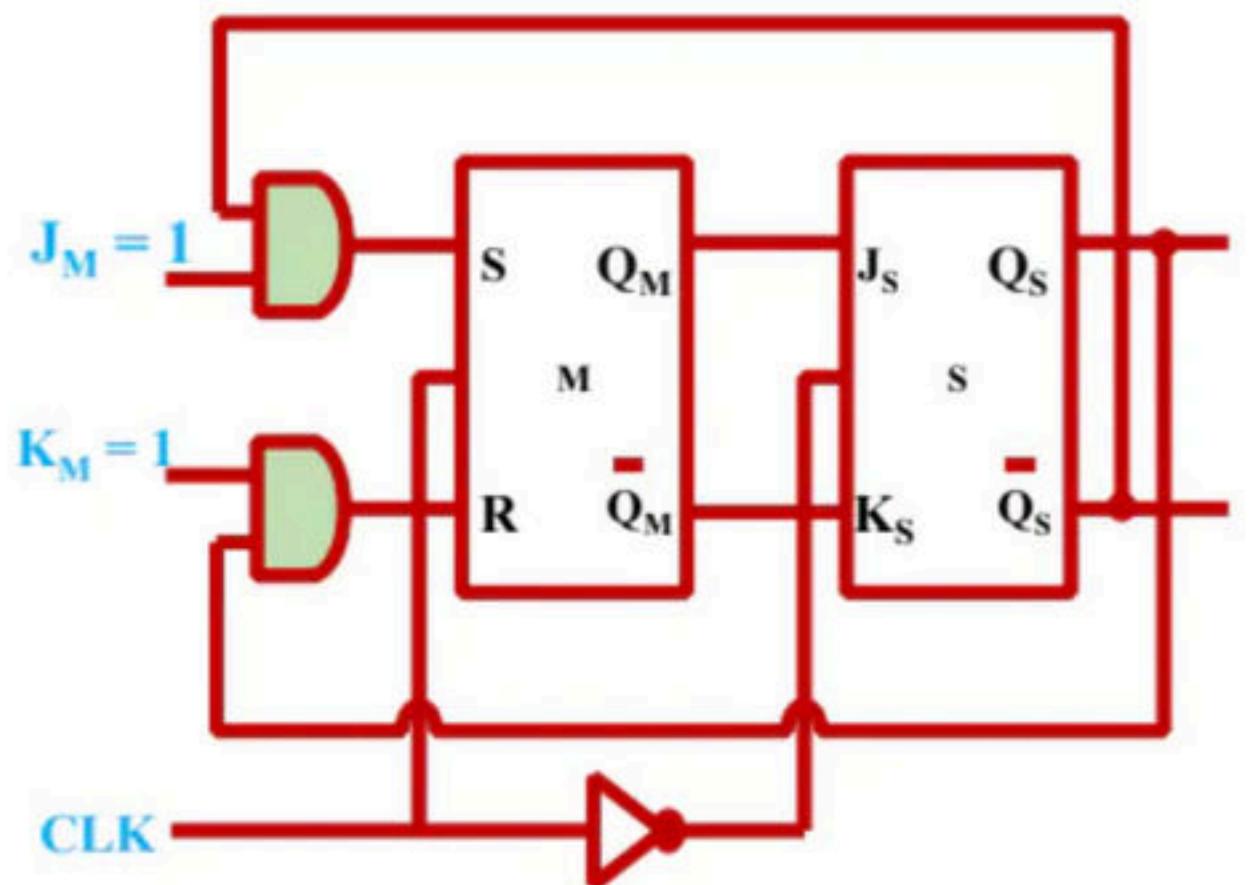
1. If the FFs are operated in level triggering
2. if $(tpd) < (Tclk)_{on}$,
3. If the FFs are operated in Toggle mode

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

$$\begin{aligned}J &= S \\K &= R\end{aligned}$$

SR to JK

$$\begin{aligned}S &= J\bar{Q} \\R &= KQ\end{aligned}$$

D to SR

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

T to SR

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

JK to D

$$\begin{aligned}J &= D \\K &= \bar{D}\end{aligned}$$

SR to D

$$\begin{aligned}S &= D \\R &= \bar{D}\end{aligned}$$

D to JK

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

T to JK

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

JK to T

$$\begin{aligned}J &= T \\K &= T\end{aligned}$$

SR to T

$$\begin{aligned}S &= T\bar{Q} \\R &= TQ\end{aligned}$$

D to T

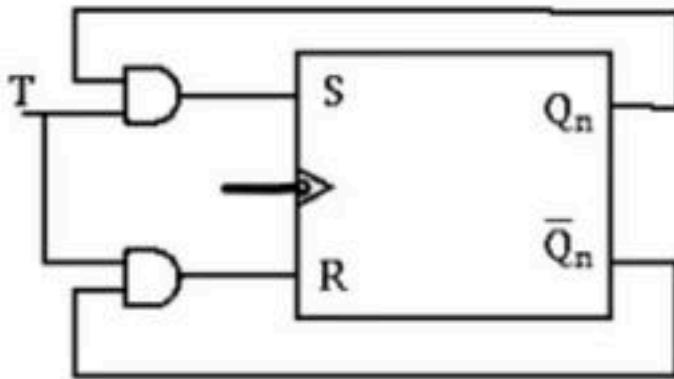
$$D = T \oplus Q$$

T to D

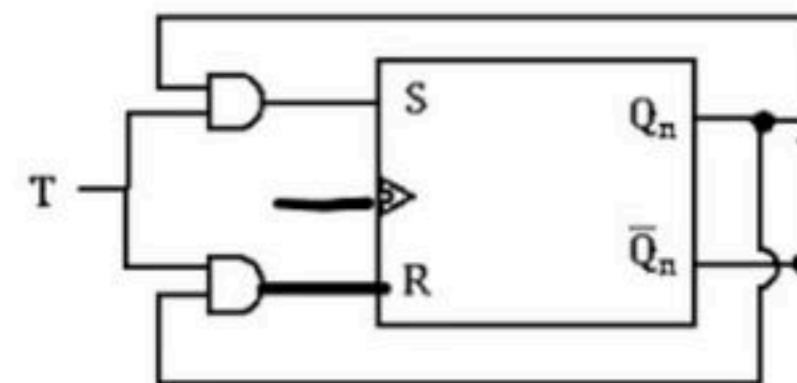
$$T = D \oplus Q$$

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

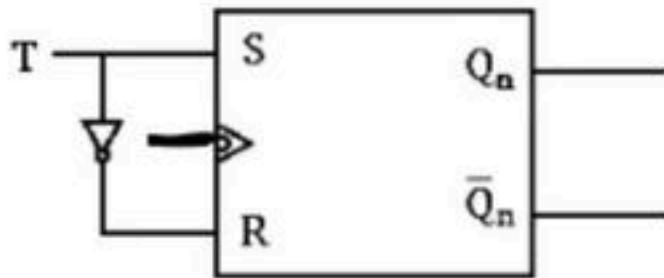
(A)



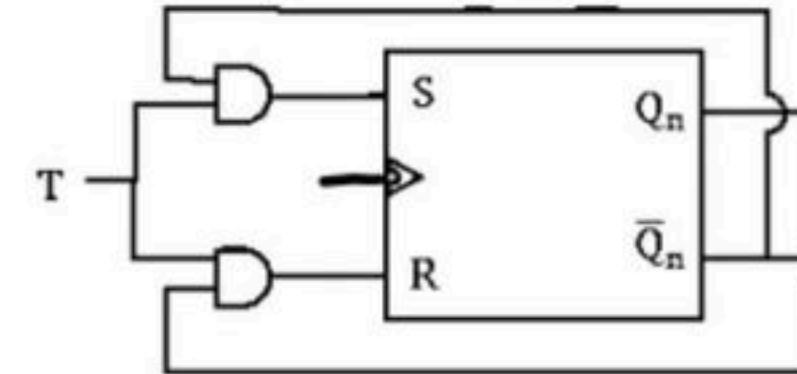
(B)



(C)

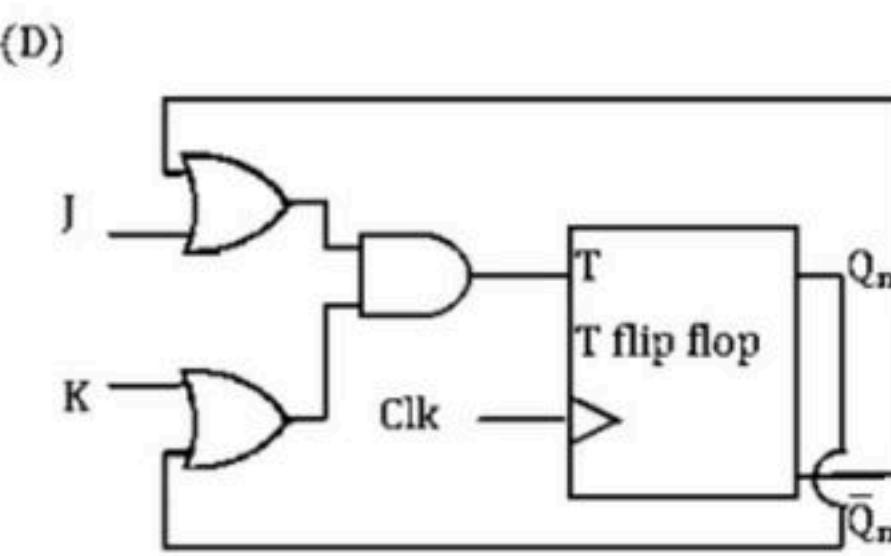
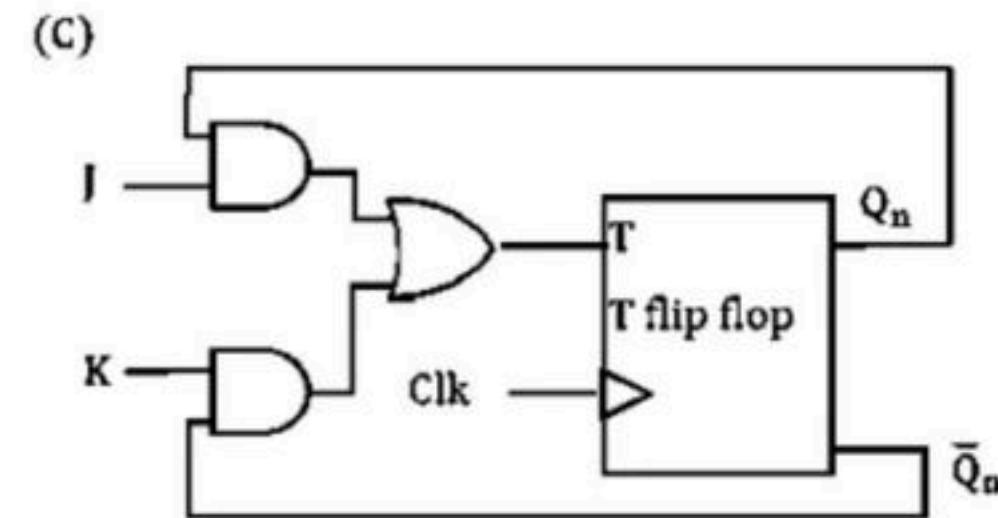
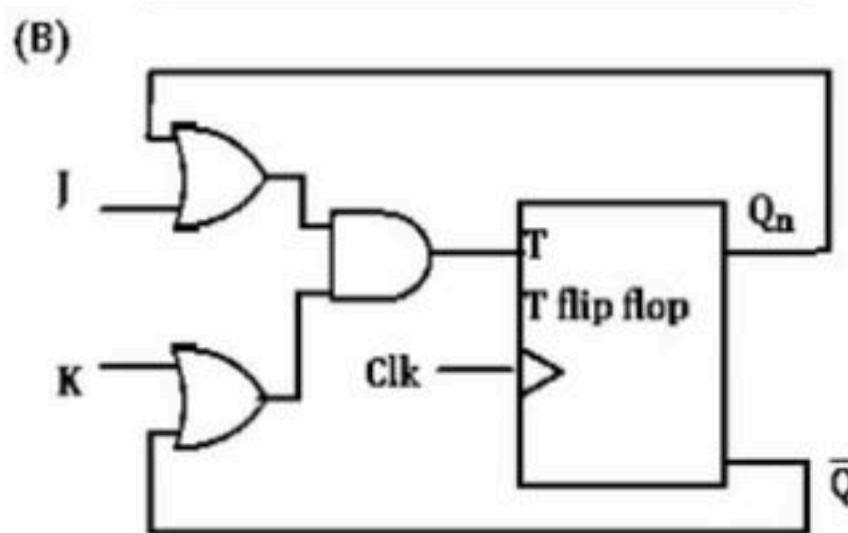
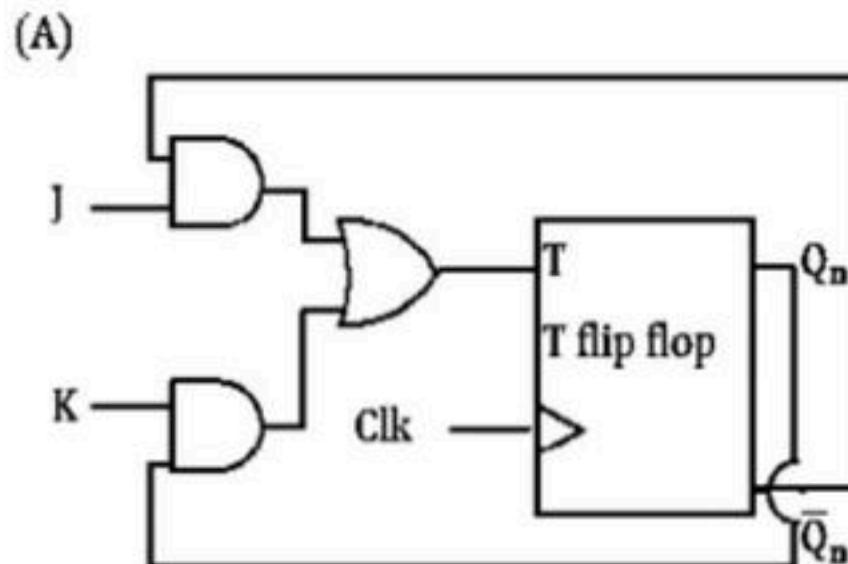


(D)



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

GATE (EE-2014)



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

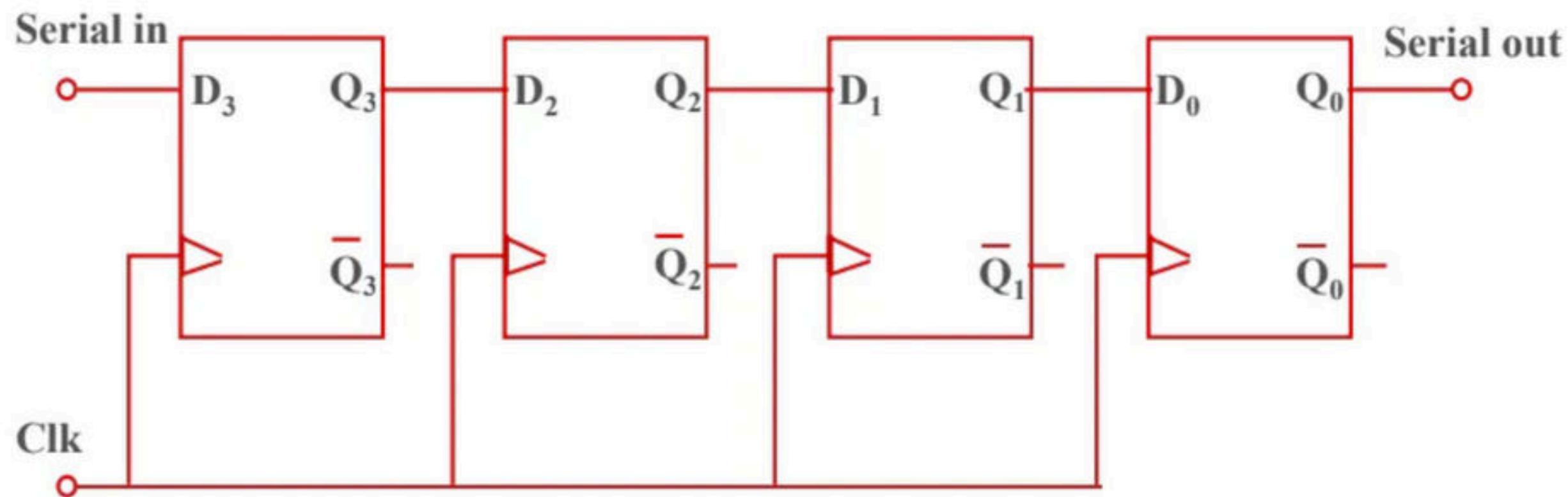
1. Serial data (Temporal code)

2. Parallel data (spatial code)

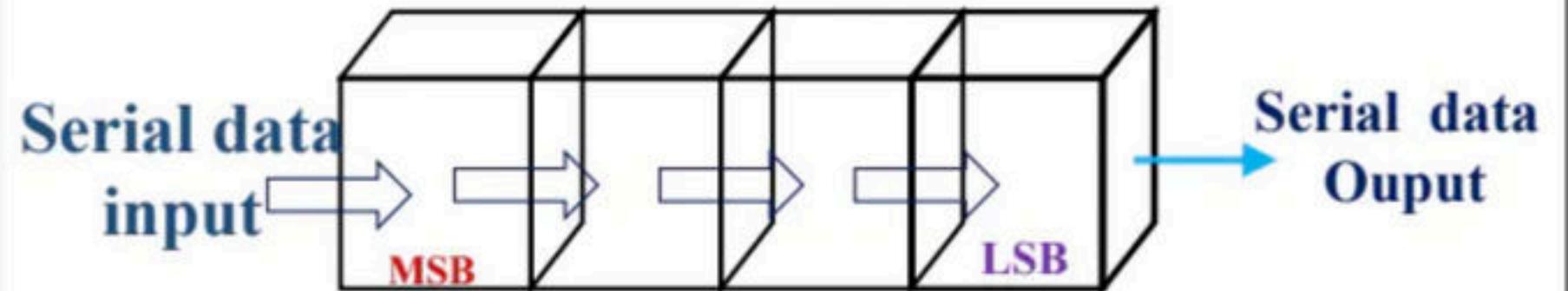
➤ Depending on i/p and o/p , registers are classified into 4 types

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



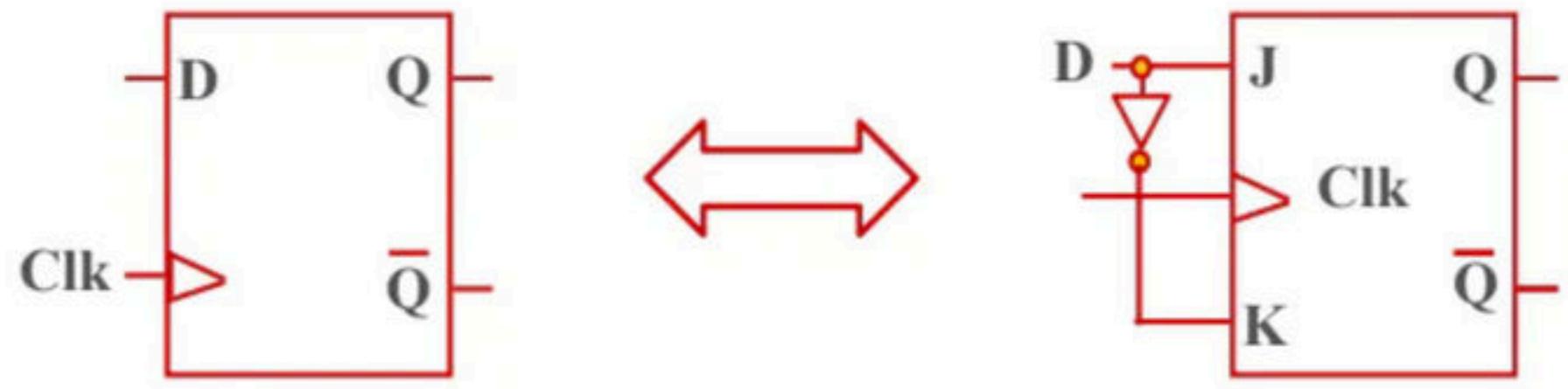
Block Diagram representation



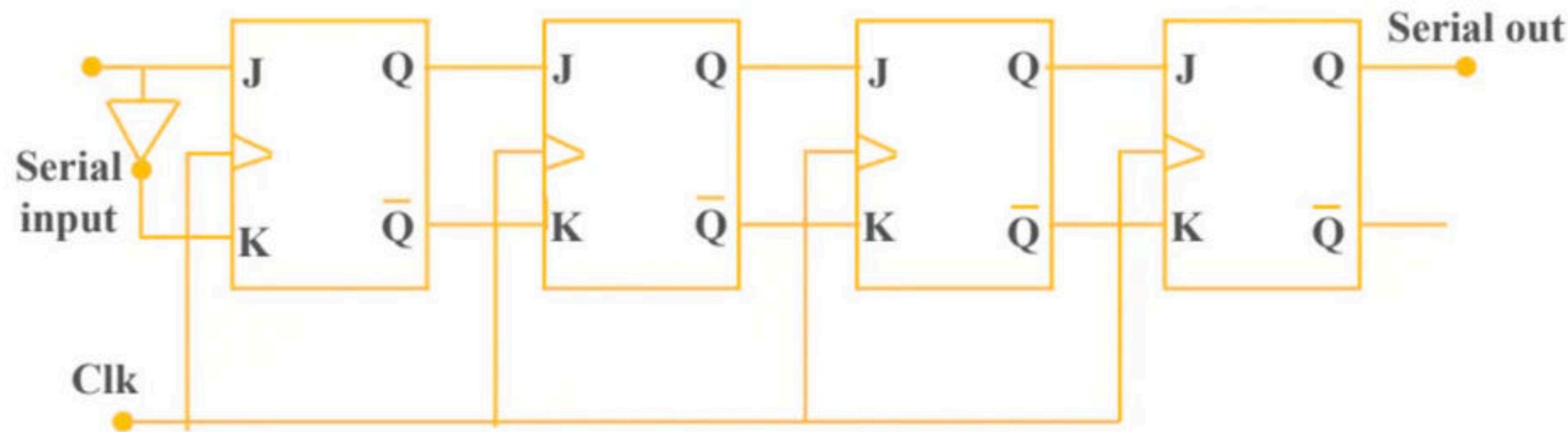
- SISO Configuration has only
 - 1- input
 - 1- output
- For SISO configuration
 - for storing = Clock pulses
 - for retrieving = clock pulses
 - Total number clock pulses =

CLK	INPUT	Q3	Q2	Q1	Q0

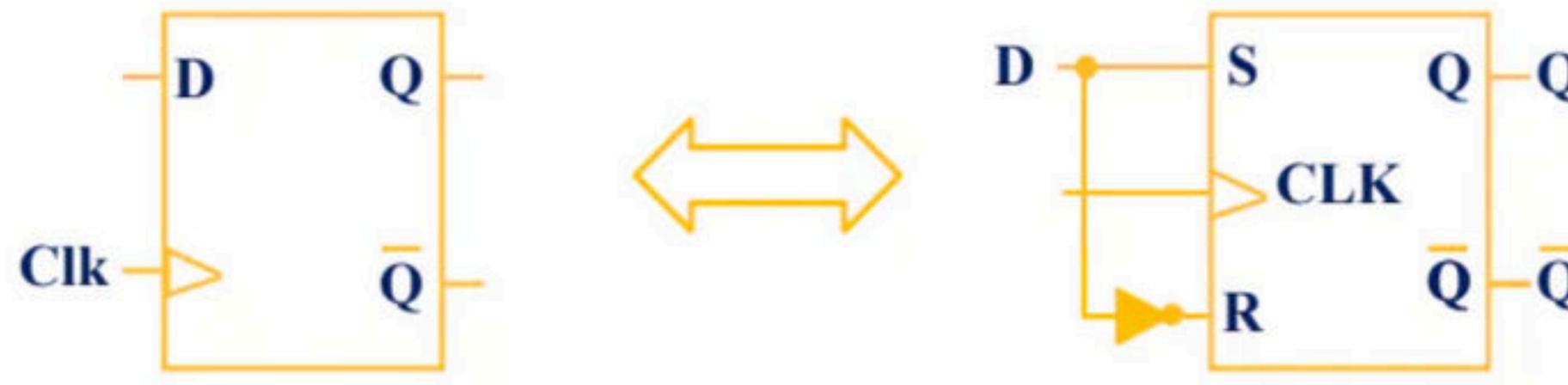
JKFF  **D FF**



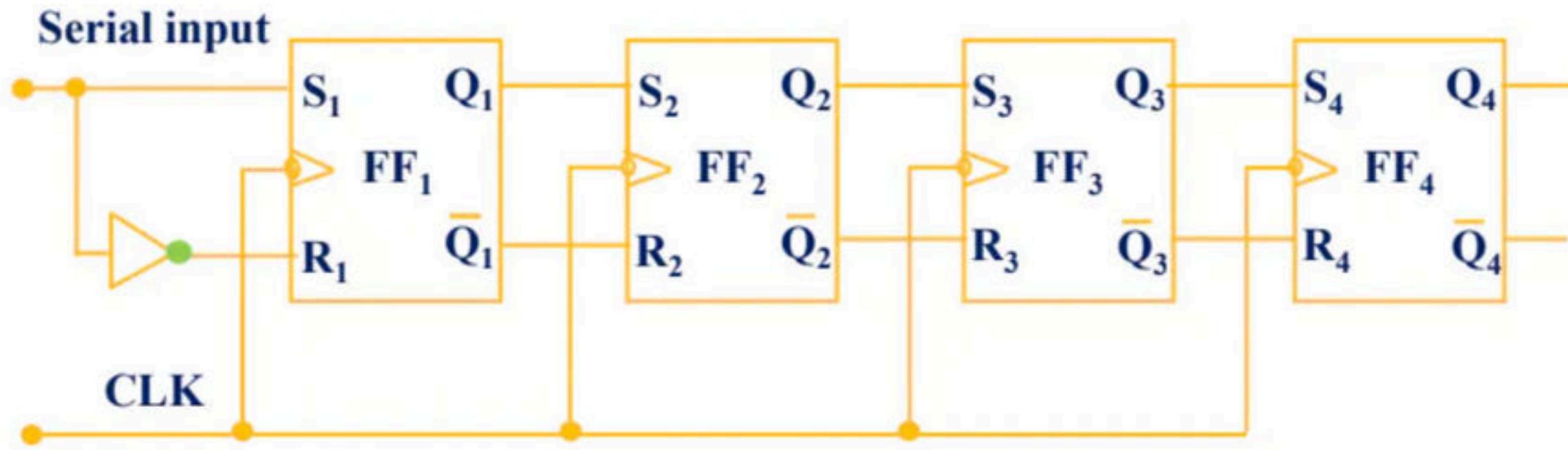
SISO Shift Register using JK FF



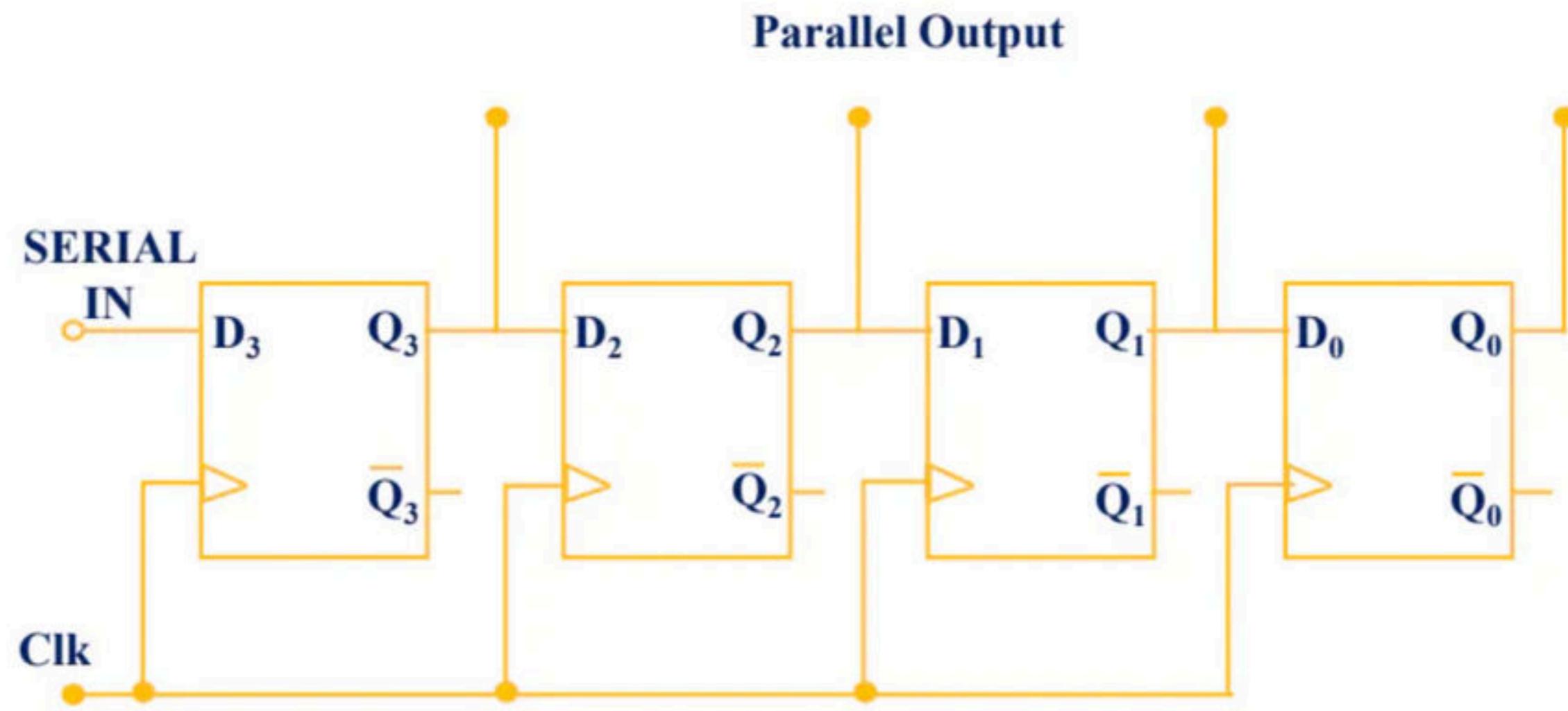
SR FF  **D FF**



SISO using SR FF



Serial In Parallel Out

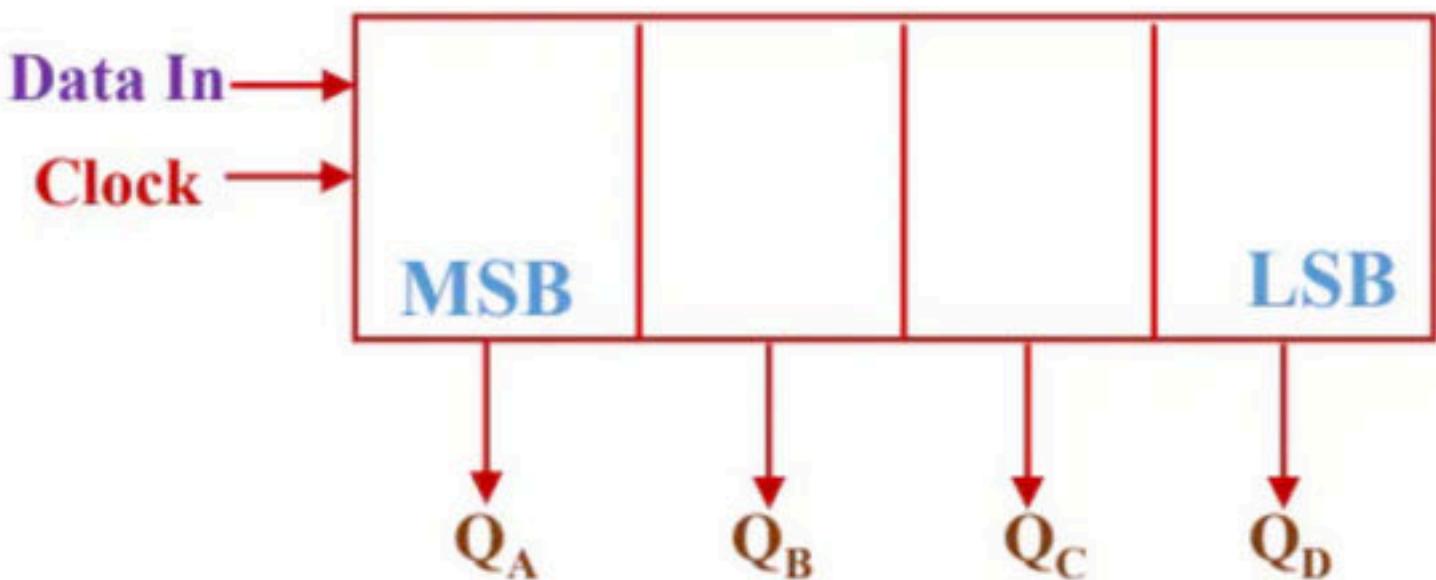


- SIPO Configuration has only
 - 1- input
 - 4- output
- For SIPO configuration
 - for storing = Clock pulses
 - for retrieving = Clock pulses

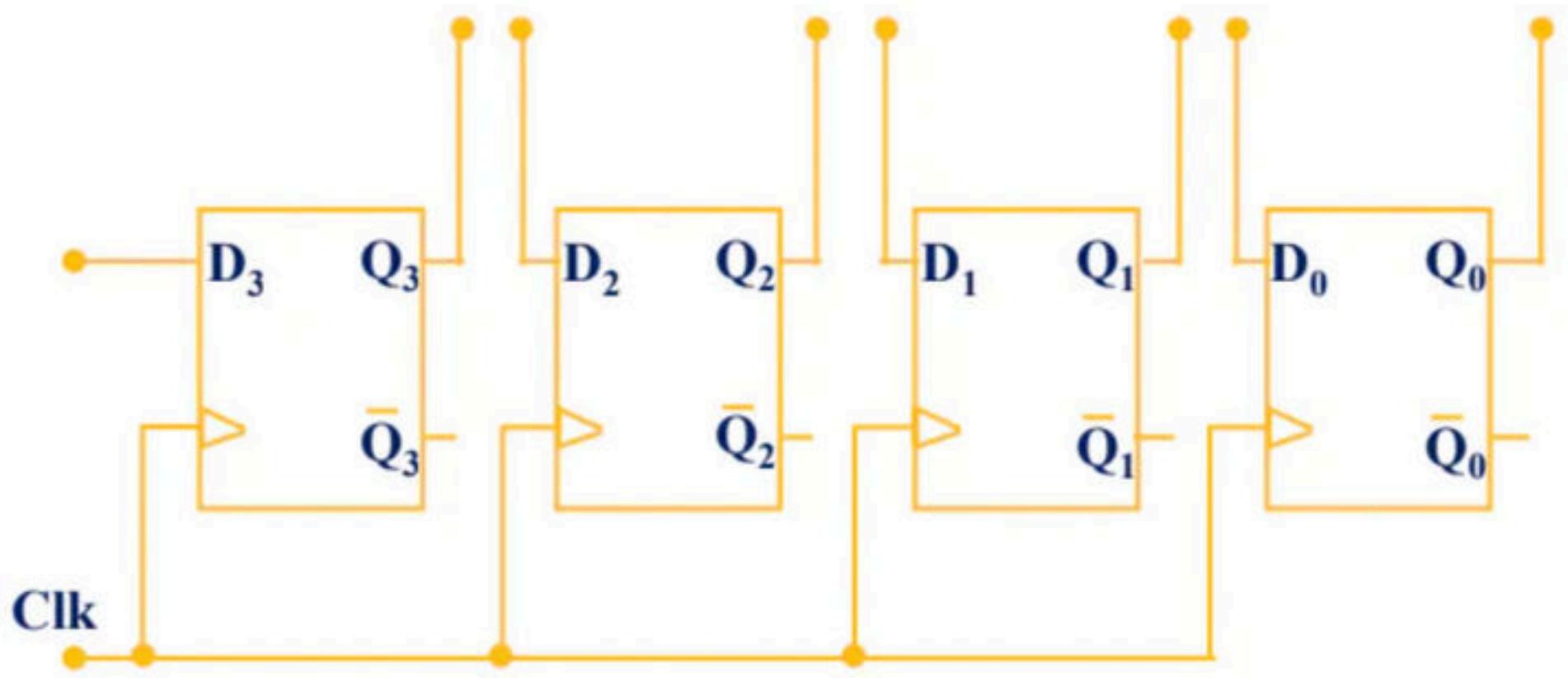
Clock pulses

Clock pulses

Total number clock pulses =



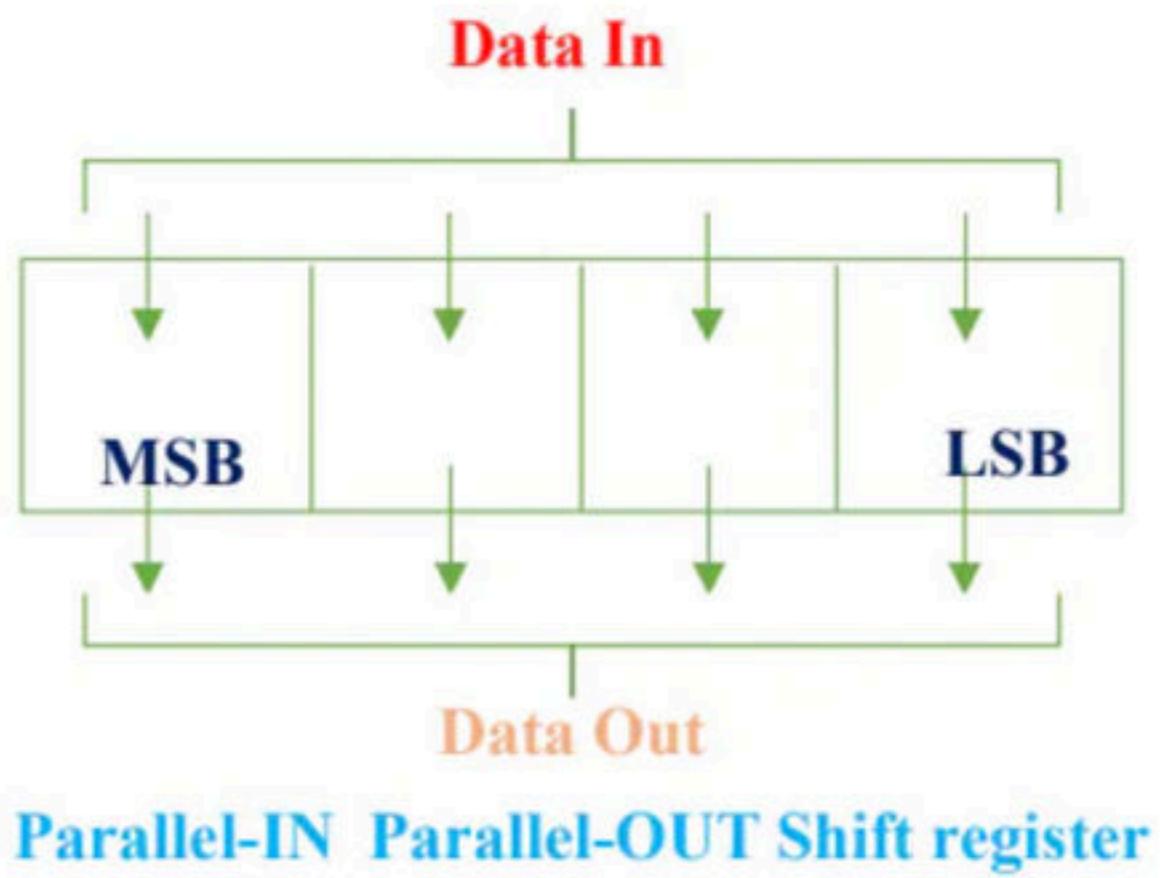
Parallel In Parallel Out



- PIPO Configuration has only
4- input
4- output

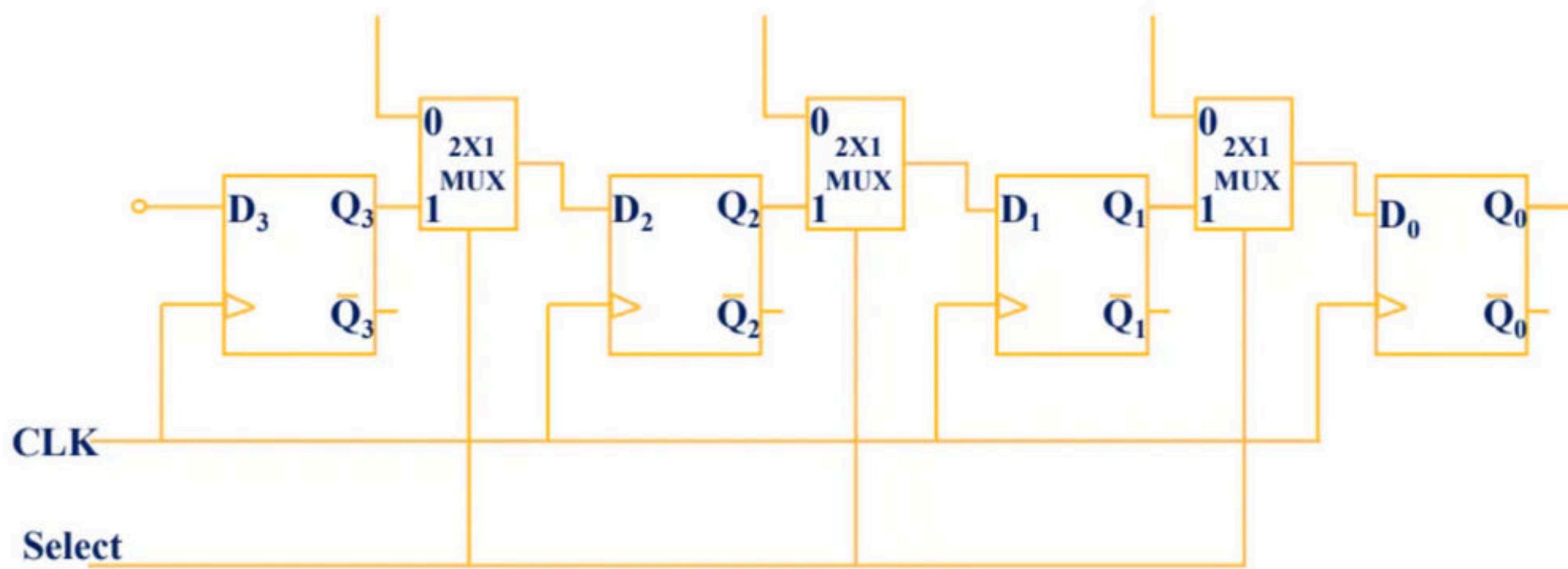
- For PIPO configuration
for storing = Clock pulses
for retrieving = Clock pulses

Total number clock pulses =

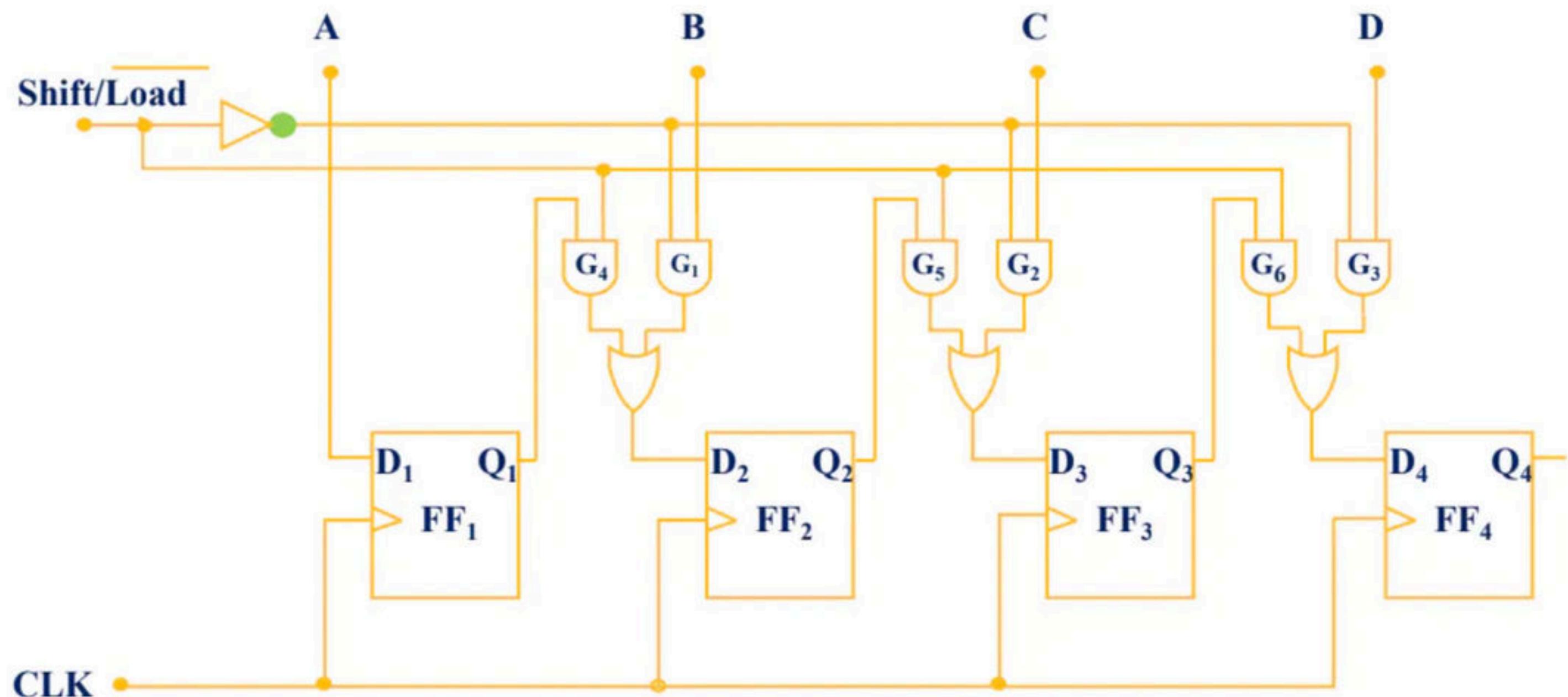


Parallel In Serial Out

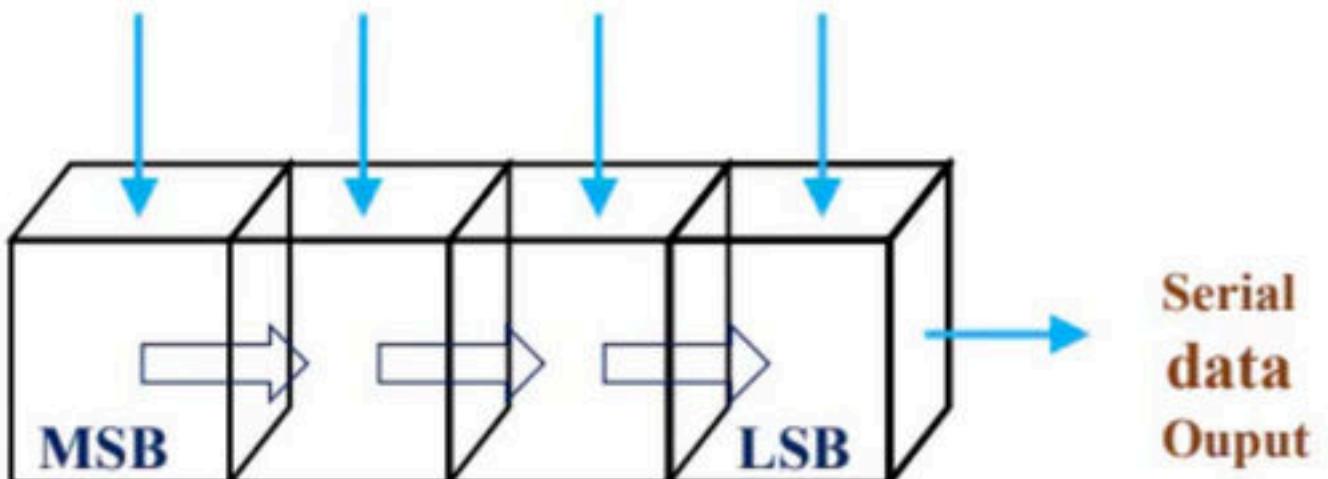
Parallel In Serial Out



Parallel In Serial Out



**Parallel
data input**



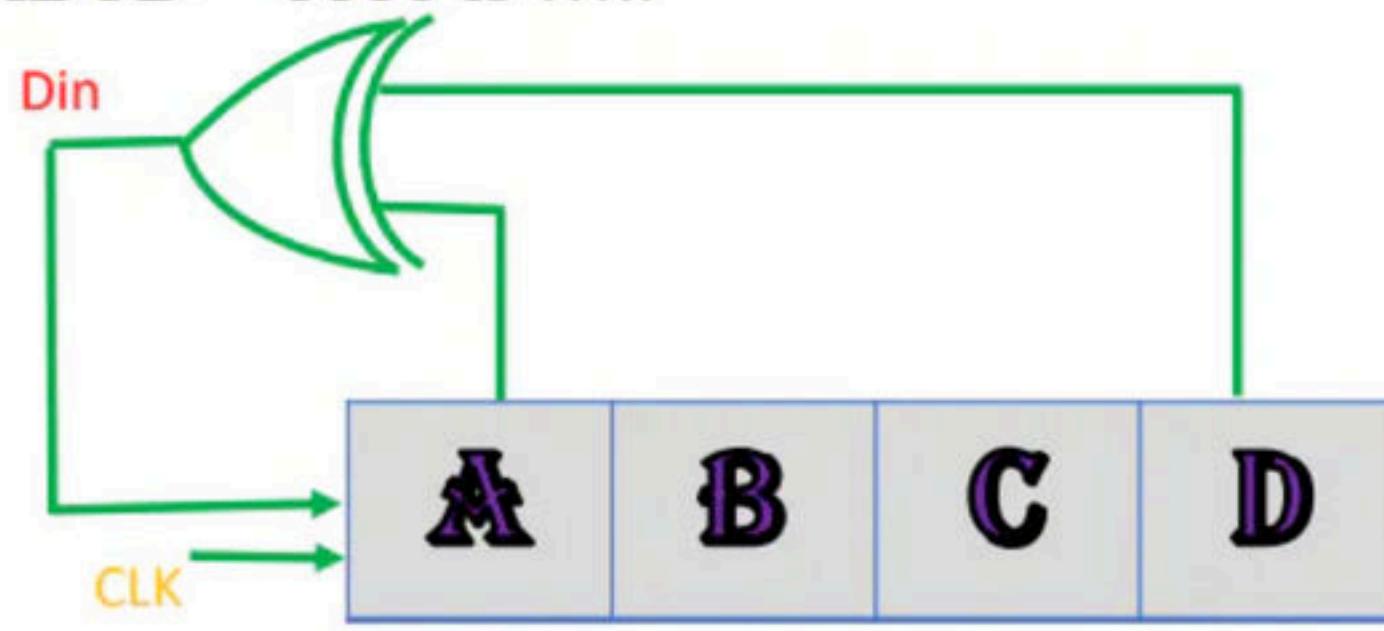
➤ PISO Configuration has only
4- input
1- output

➤ For PISO configuration
for storing = Clock pulses

for retrieving = Clock pulses

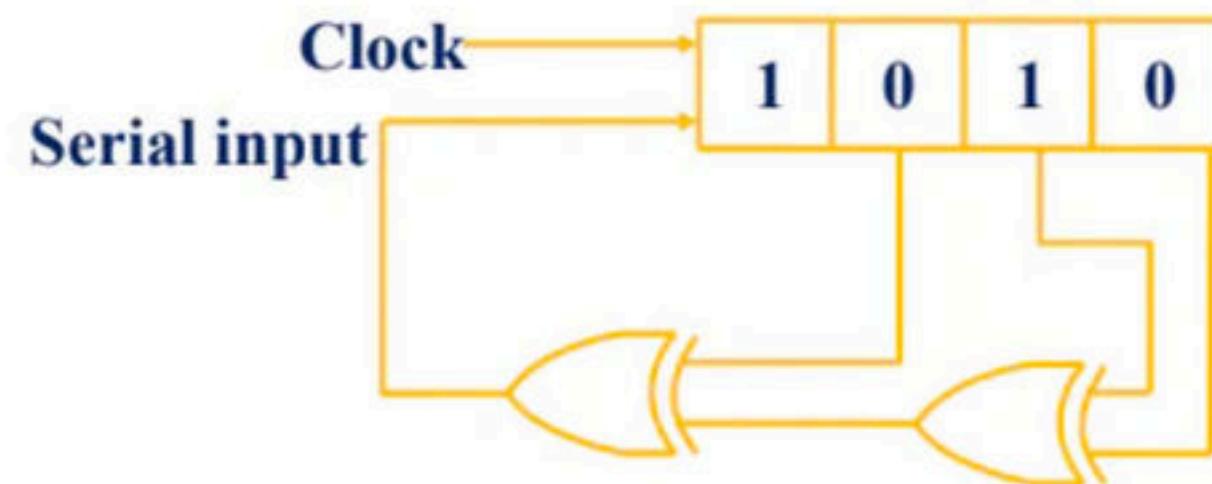
Total number clock pulses =

Q) A 4 – bit shift register circuit configured for right shift operation $Din \rightarrow A$, 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

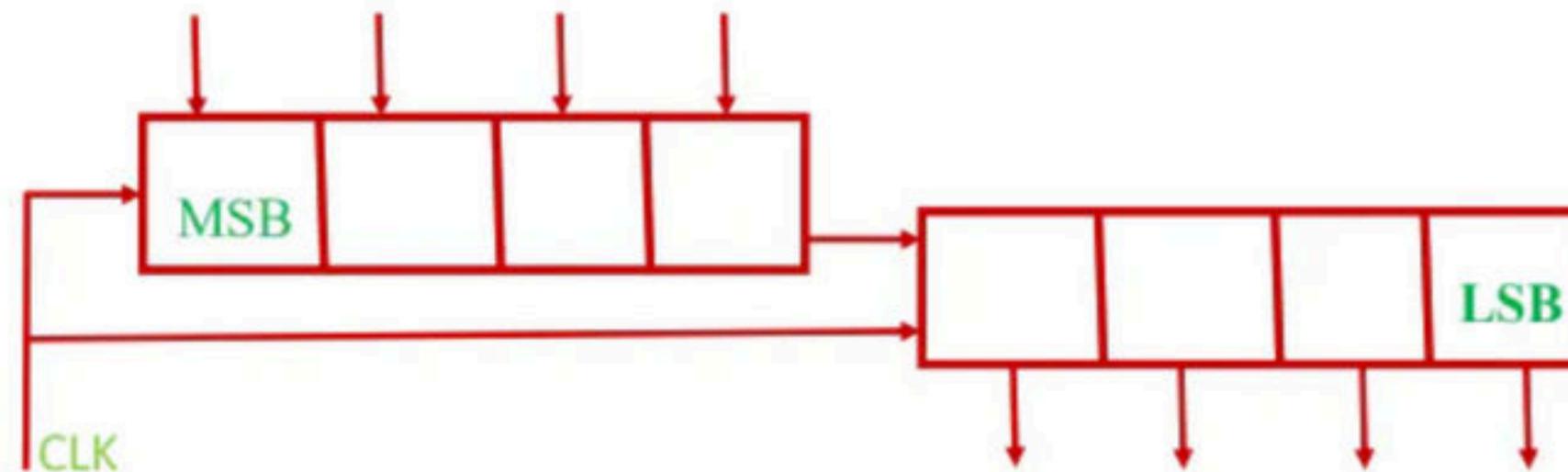


Q. The shift register shown in figure is initially loaded with the bit pattern 1010. Subsequently the shift register is clocked, and with each clock pulse the pattern gets shifted by one bit position to the right. With each shift, the bit at the serial input is pushed to the left most position (MSB).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

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

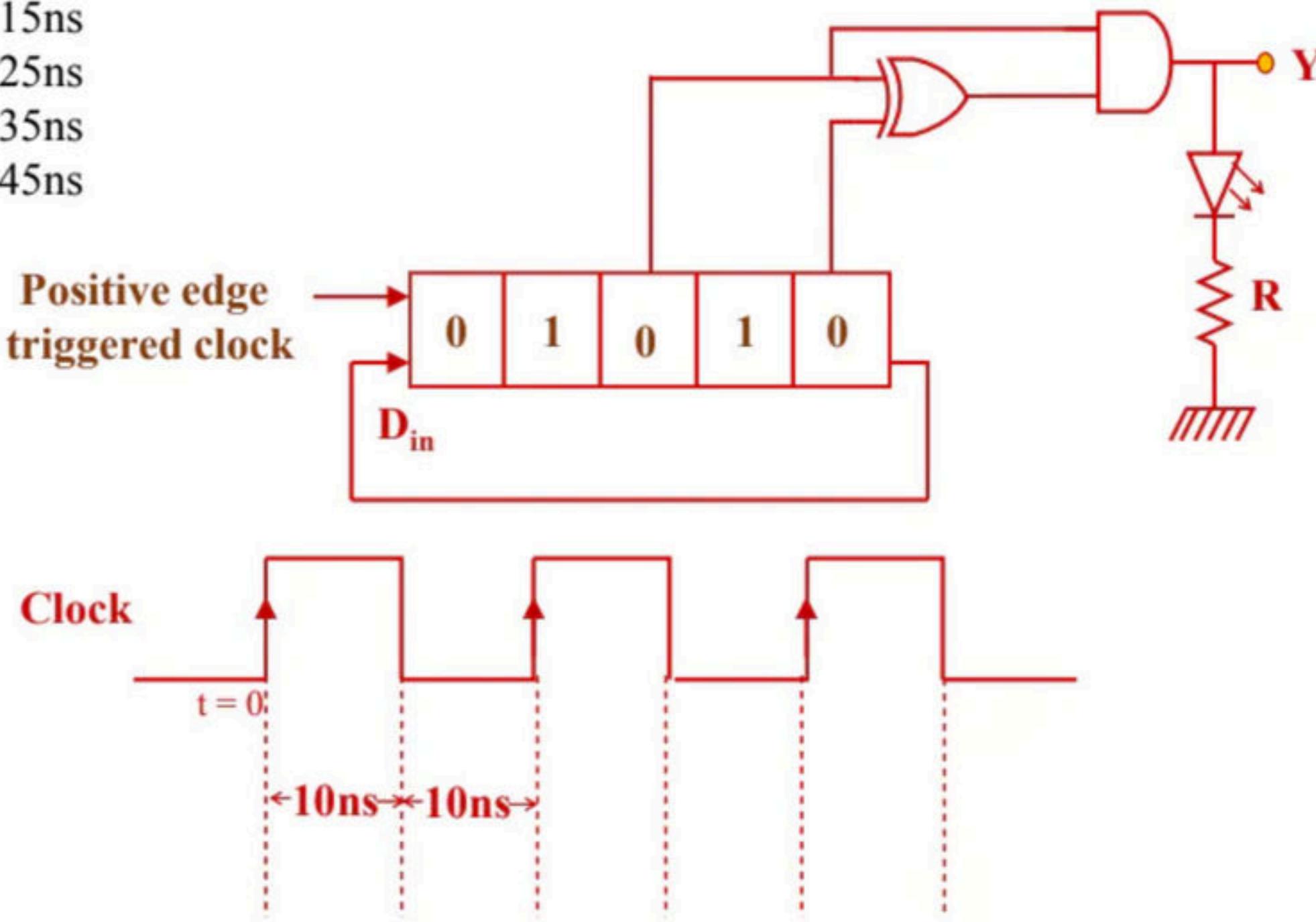


Q) An 8-bit register is made of one 4-bit PISO register (synchronous loading) cascaded with a 4-bit SIPO register as shown in the figure below , then total number of clock pulses required to perform write and read operations for one byte is -----



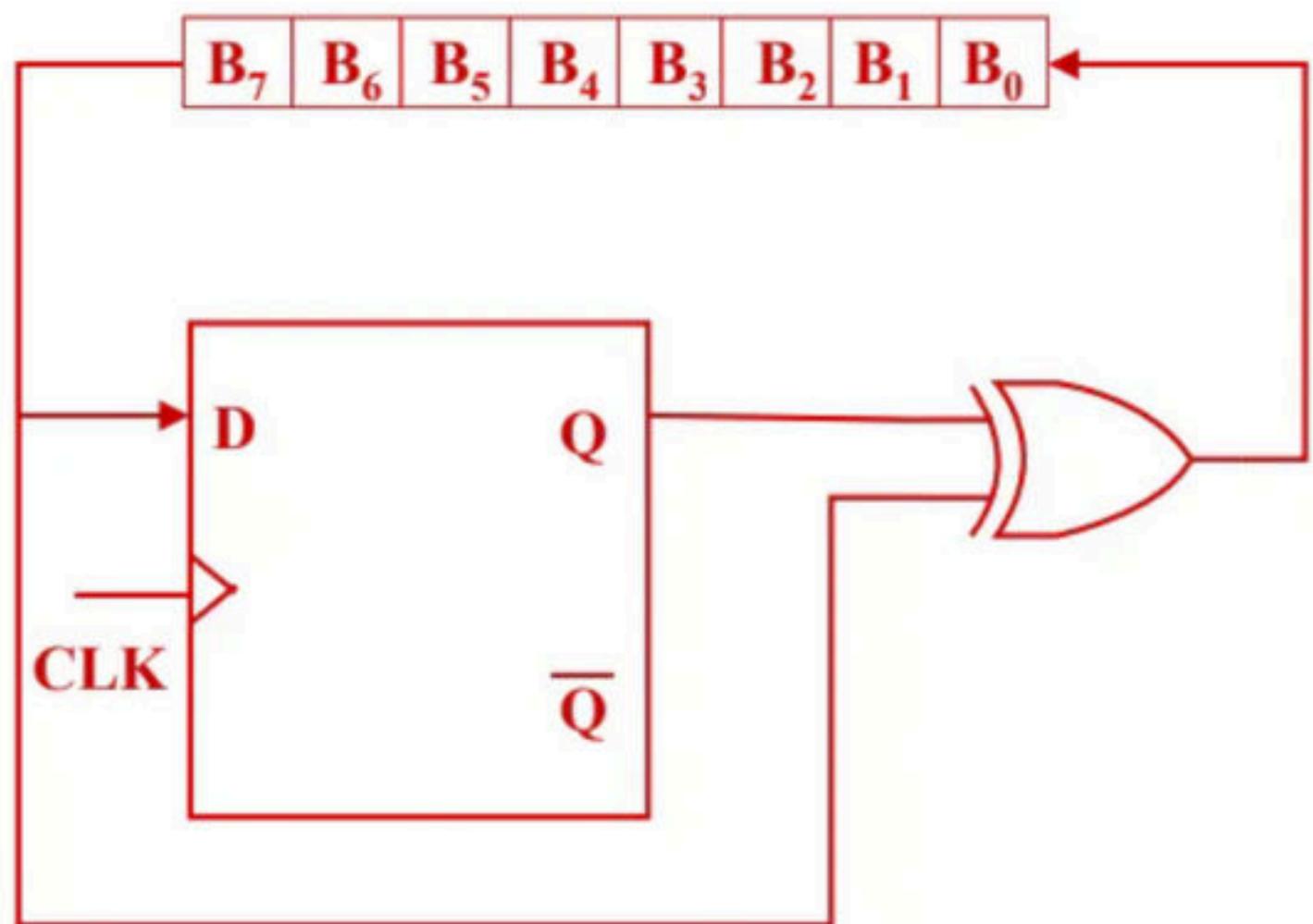
Q) Consider a serial in parallel out , right shift register circuit with initial contents as 01010 as shown in the figure . This circuit is operated with a positive edge triggered clock signal as given in figure . If +5V and 0V are used to represent logic-1 and logic-0 respectively , the at which of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

Binary

UP

Down

Non Binary

UP

Down

**Ring
counter**

Synchronous

**Johnson
Ring
counter**

**User
defined
counter**

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 =



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. \oplus ve Edge trigger and Q as a clock -----> Up counter
2. \ominus ve Edge trigger and \bar{Q} as a clock -----> Down counter
3. \oplus ve Edge trigger and Q as a clock -----> Down counter
4. \oplus ve Edge trigger and \bar{Q} as a clock -----> Up counter

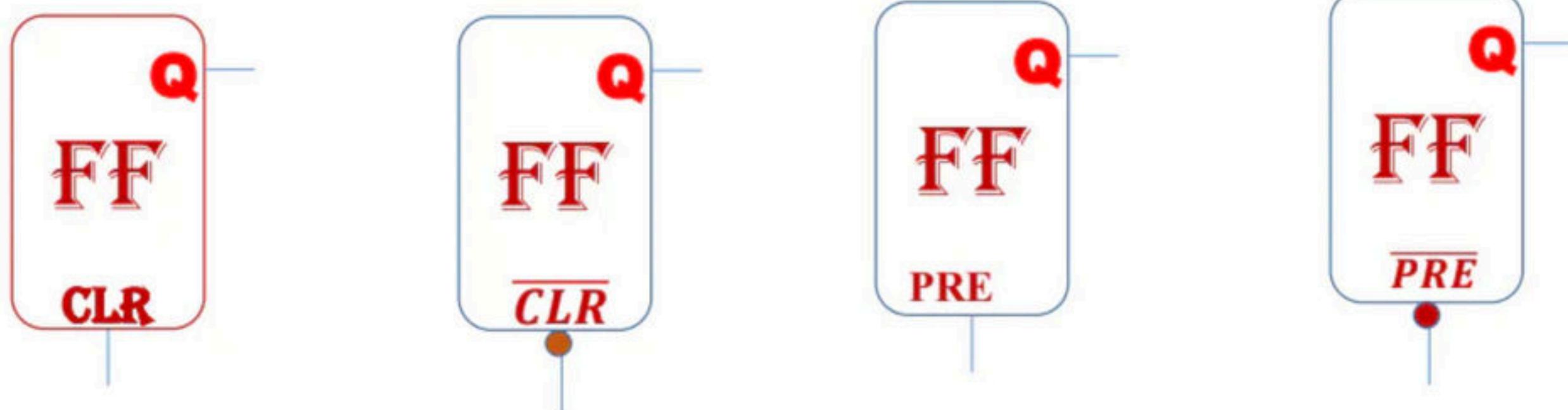
State Diagram

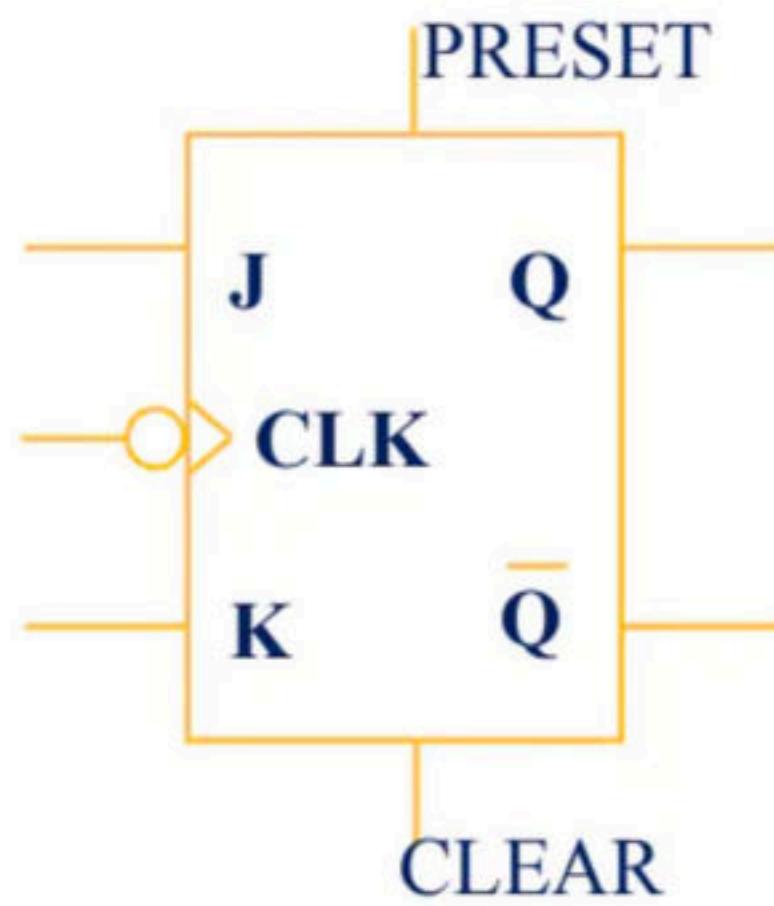
- The disadvantages of the ripple counter is that transition states are present due to delay of the FF (Decoding errors) .
- If only one FF changes its state ,then no transition states will be present , if more than one FF changes its states than transition states present.
- To avoid decoding errors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

3-Bit Ripple Up/Down counter

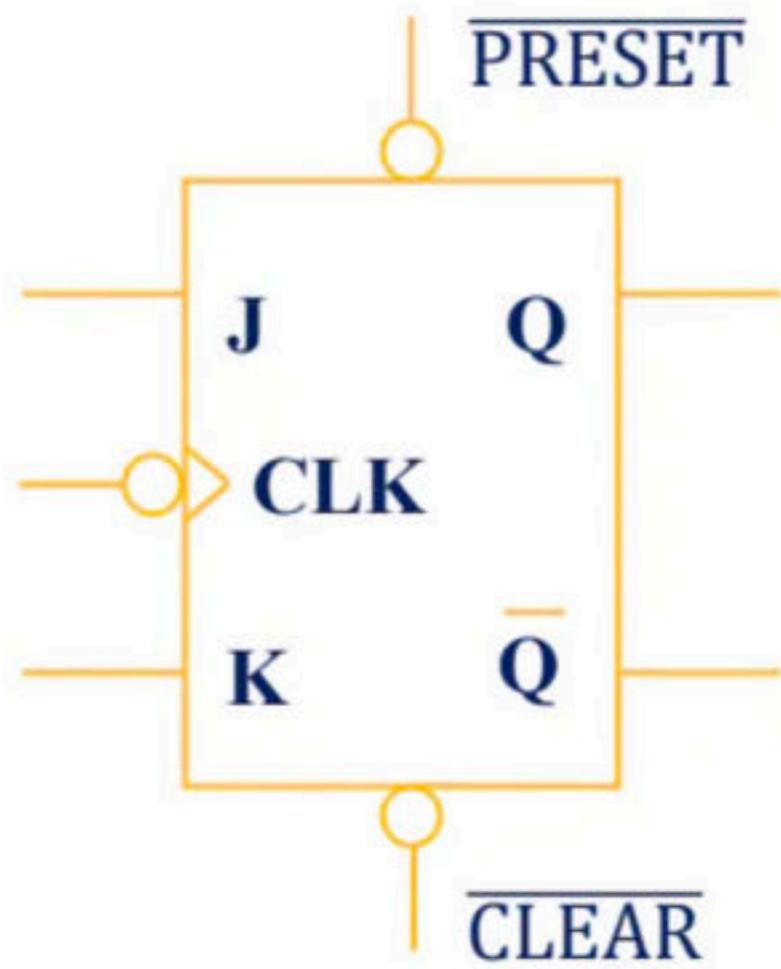
Asynchronous Clear and Asynchronous Preset

- 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 <u>PRESET</u>	Asynchronous <u>CLEAR</u>	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

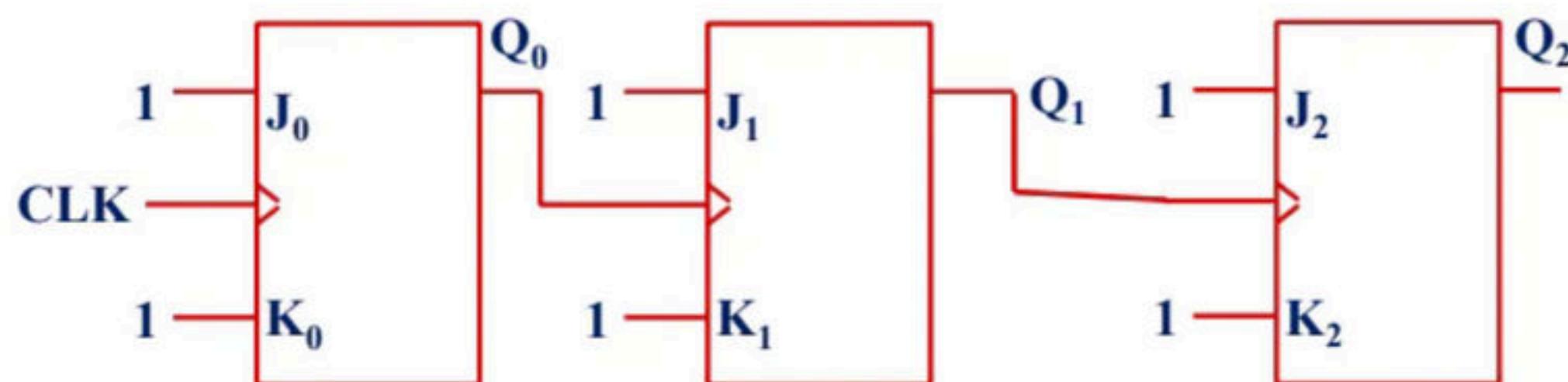
Delay analysis(Asynchronous counter)

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

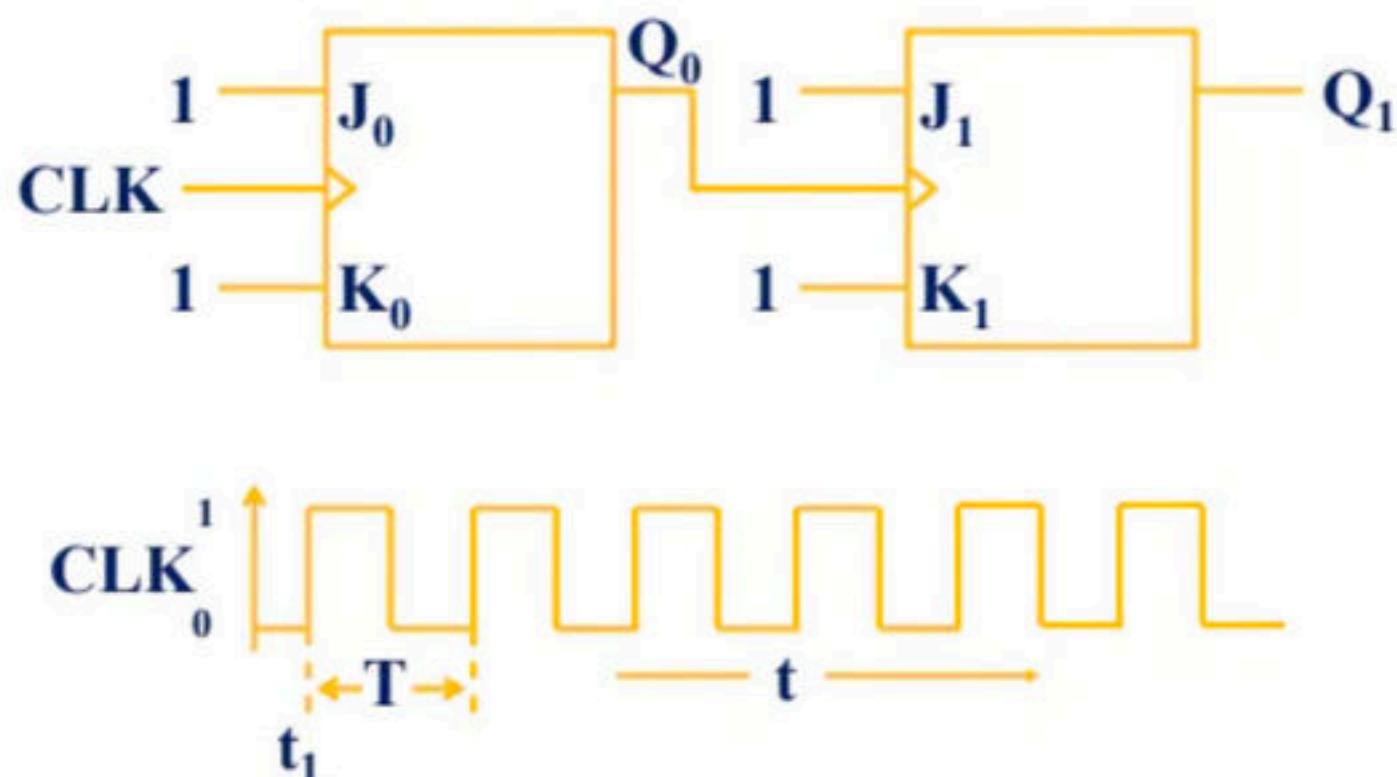
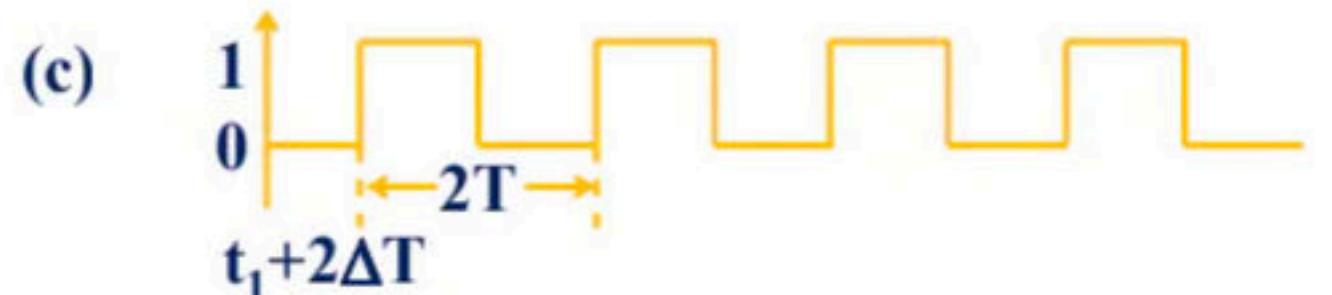
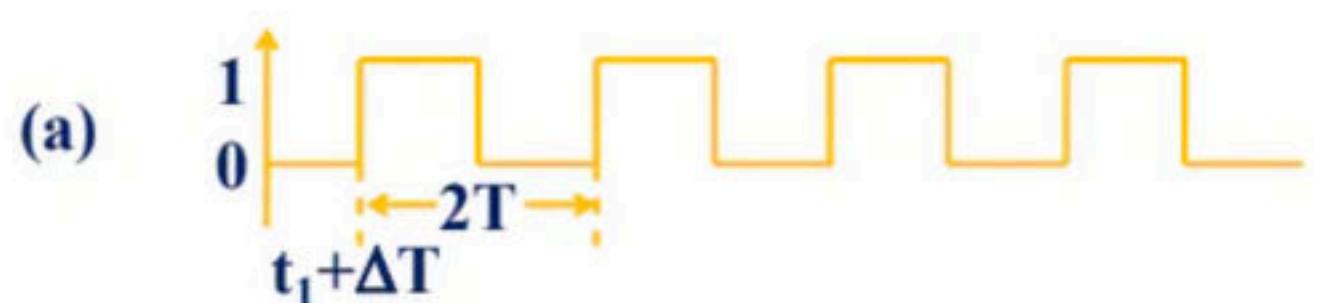
n- bit asynchronous counter is =

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 ?

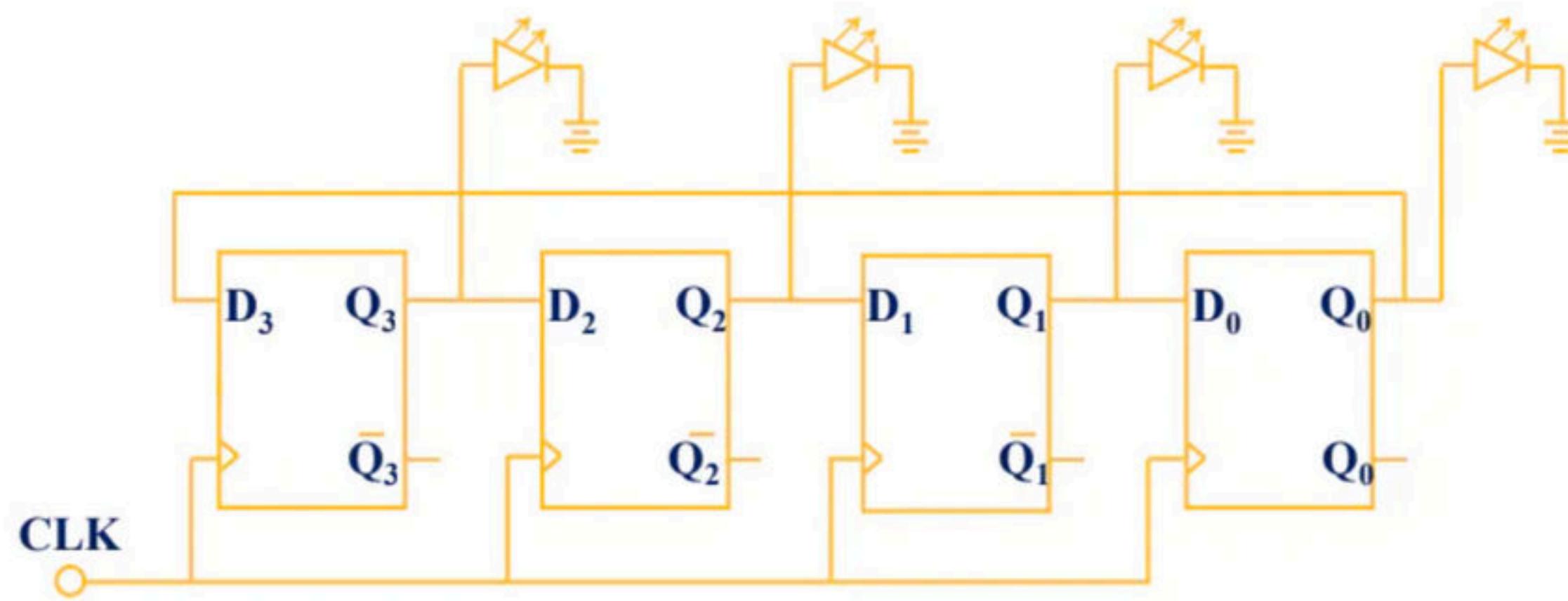


Synchronous Counter

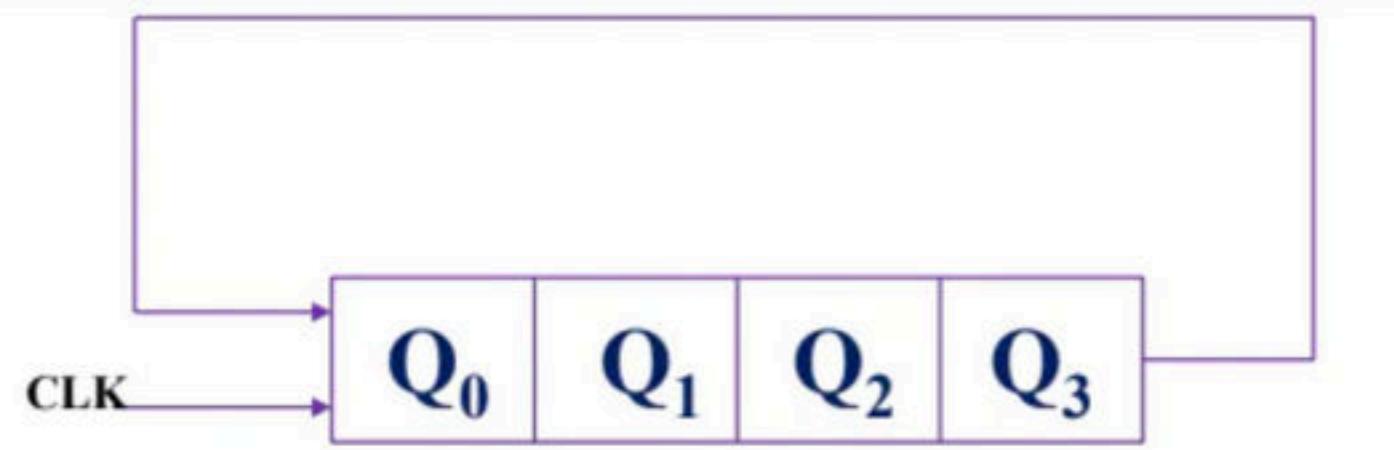
Synchronous Counter

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

4-Bit Ring Counter

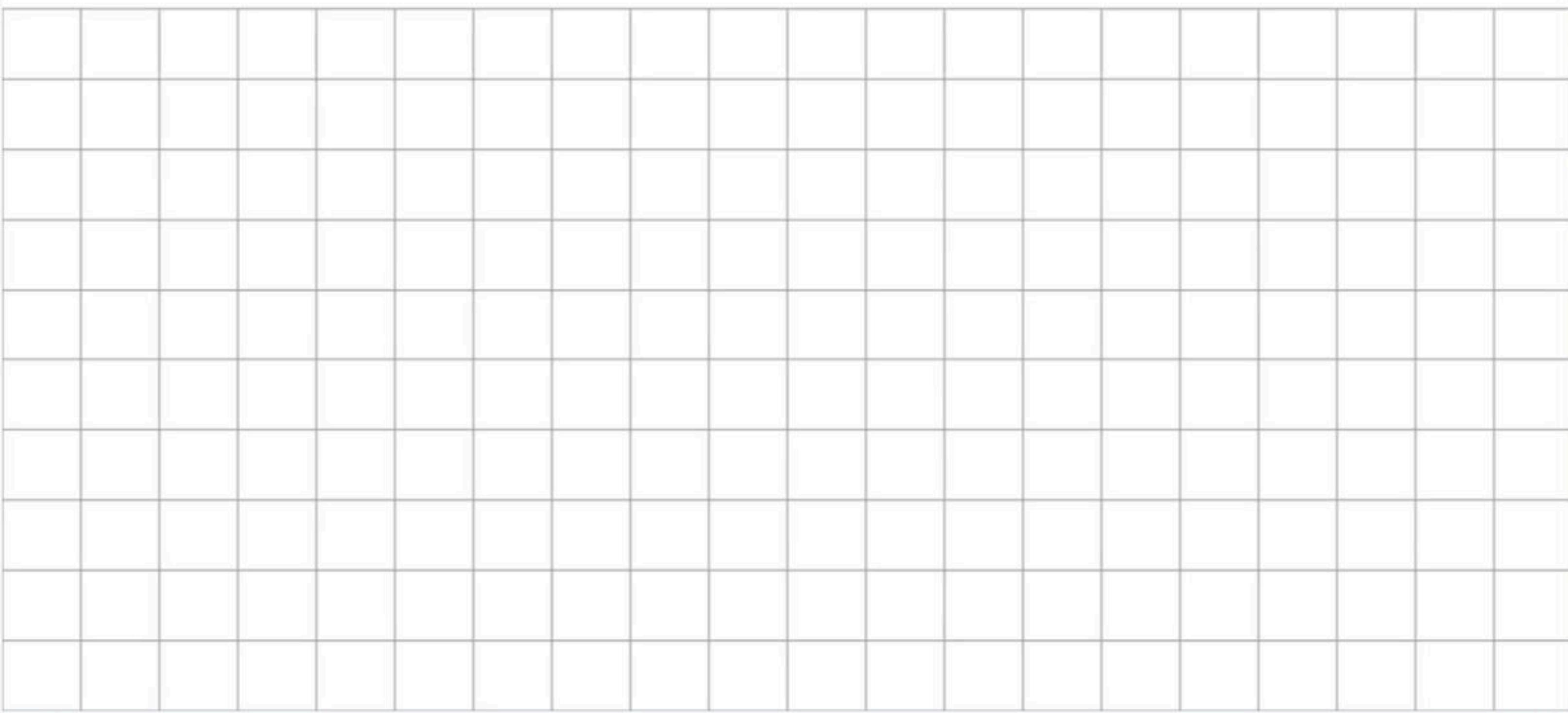


- Ring counter is a synchronous counter , it is a shift register in which last FF output is connected to the first FF input .
- Ring counter performs right shift operation .



State Diagram

Timing Diagram



Repeating pattern

Clk	Q_0

Clk	Q_1

Clk	Q_2

Clk	Q_3

$$T_{Q_0} =$$

$$f_{Q_0} =$$

$$D =$$

$$T_{Q_1} =$$

$$f_{Q_1} =$$

$$D =$$

$$T_{Q_2} =$$

$$f_2 =$$

$$D =$$

$$T_{Q_3} =$$

$$f_{Q_3} =$$

$$D =$$

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

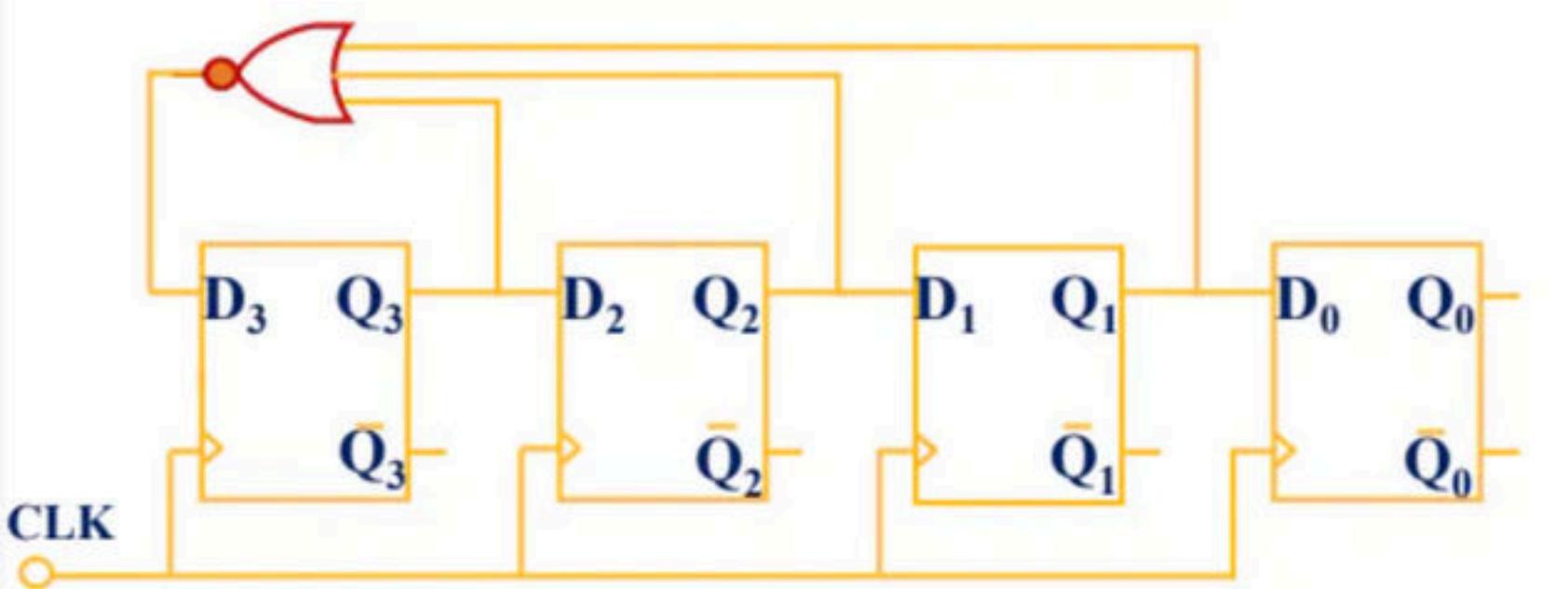
Advantage

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

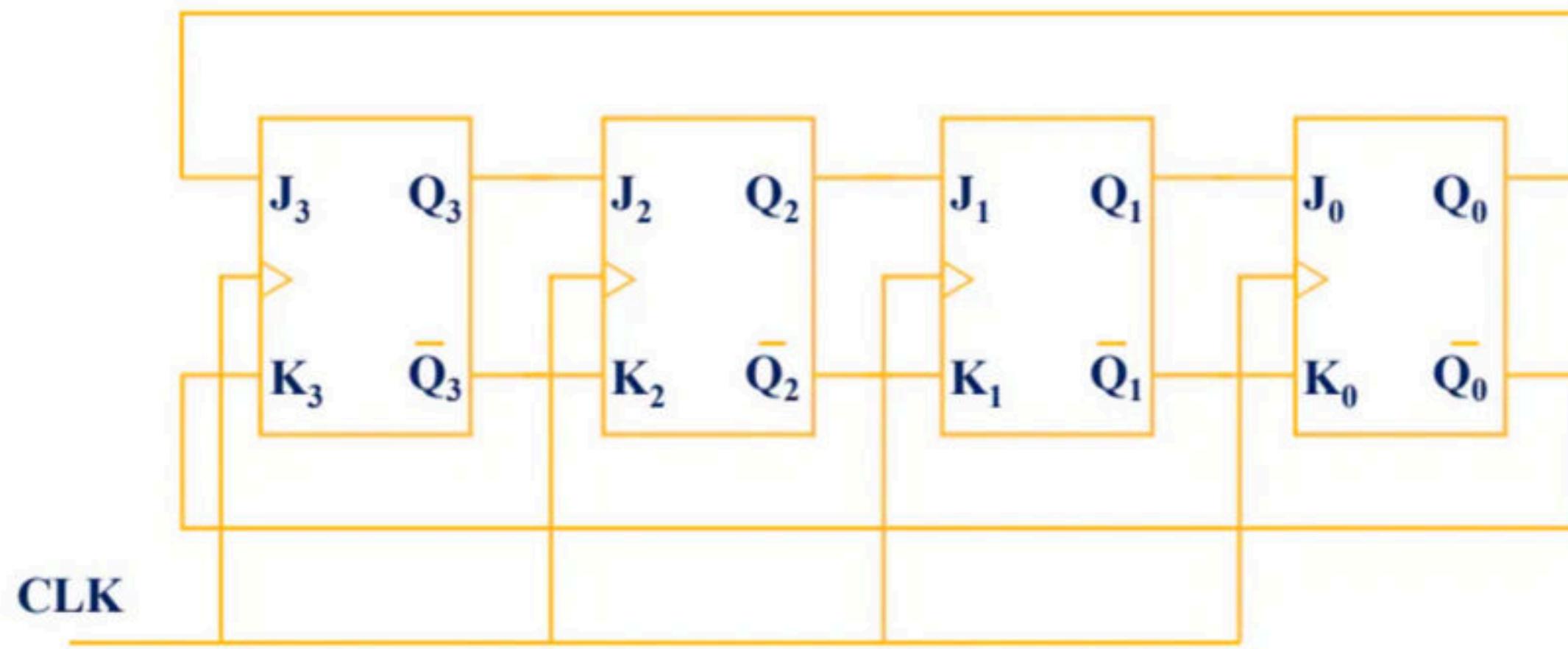
Draw back of Ring counter

- 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** .

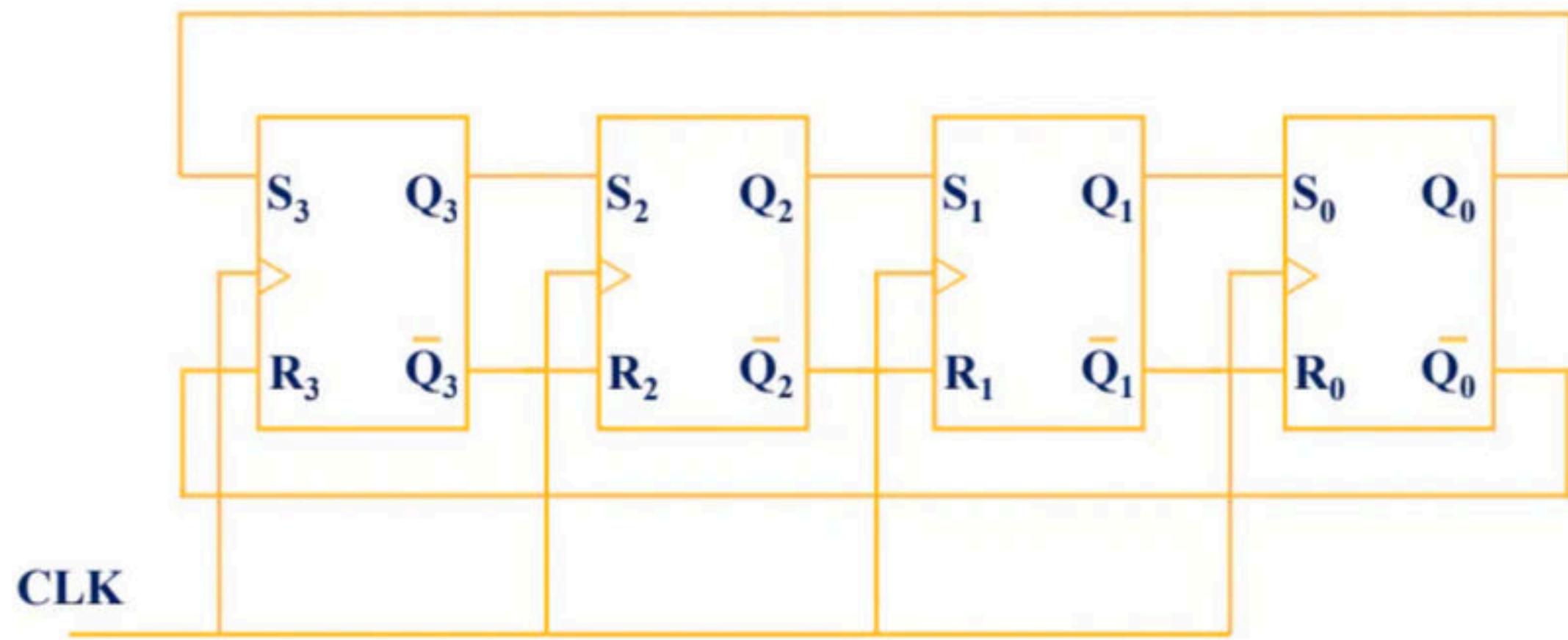
Self Starting Ring counter



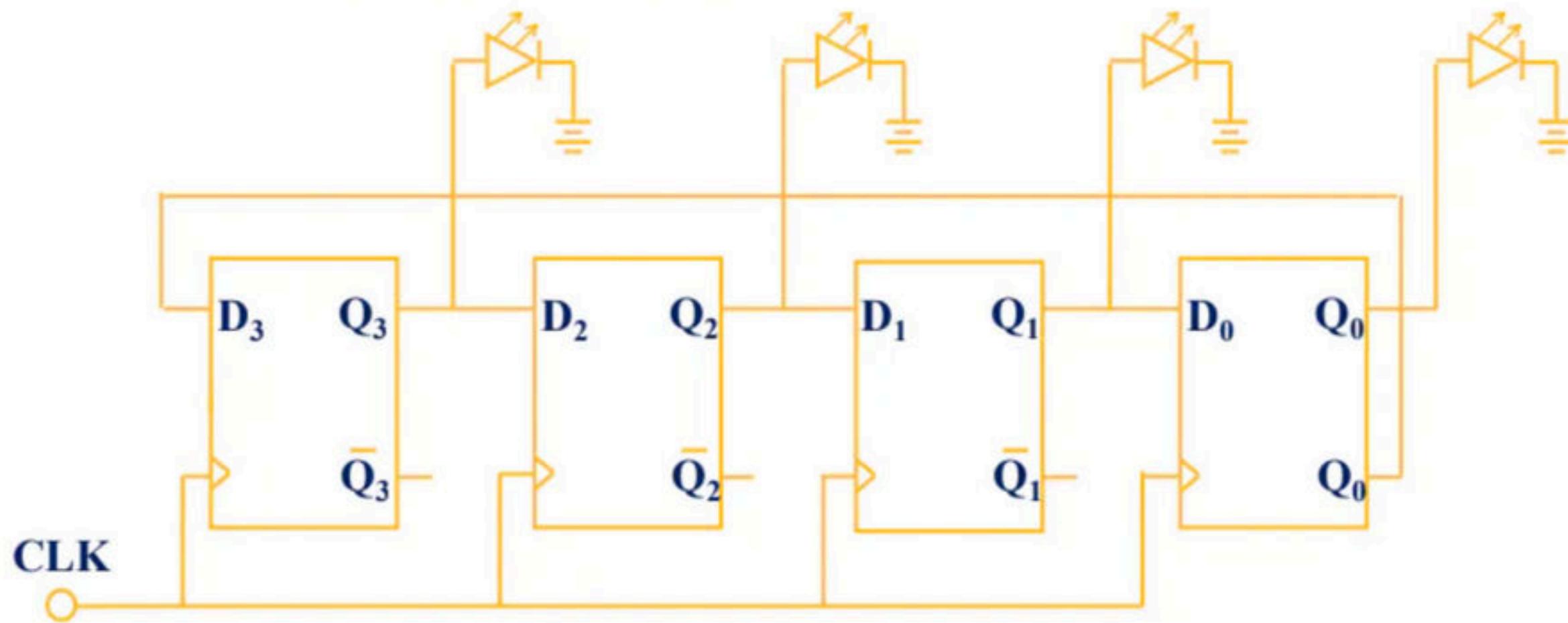
Ring counter using JK flip flop

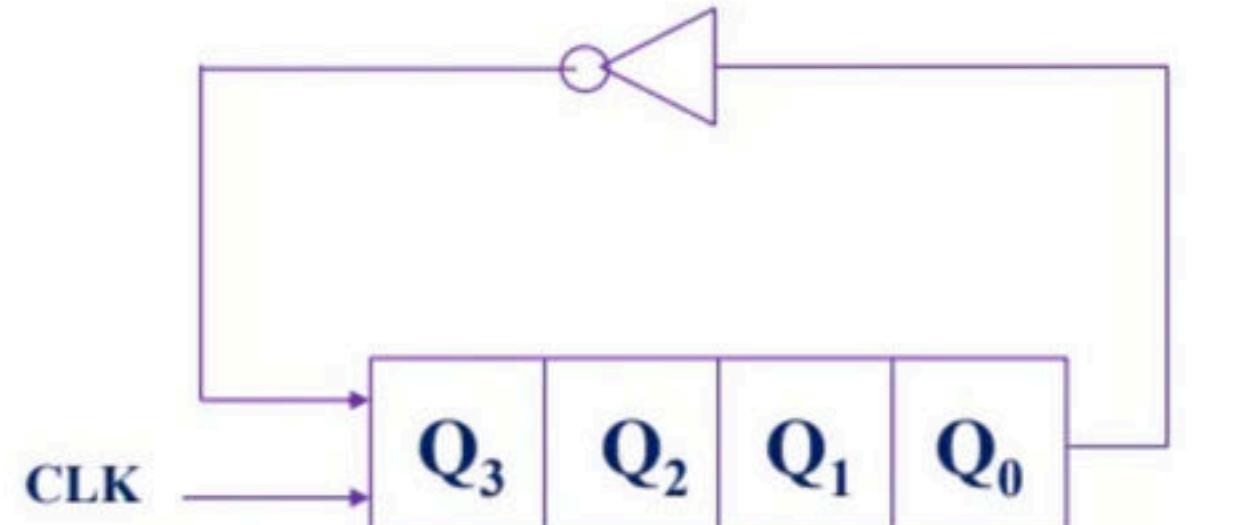


Ring counter using SR flip flop



Johnson Ring counter





Repeating pattern

Clk	Q_0

$$T_{Q_0} =$$

$$f_{Q_0} =$$

$$\mathbf{D} =$$

Clk	Q_1

$$T_{Q_1} =$$

$$f_{Q_1} =$$

$$\mathbf{D} =$$

Clk	Q_2

$$T_{Q_2} =$$

$$f_{Q_2} =$$

$$\mathbf{D} =$$

Clk	Q_3

$$T_{Q_3} =$$

$$f_{Q_3} =$$

$$\mathbf{D} =$$

For 4-bit Johnson Counter

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

For n-bit Johnson Counter

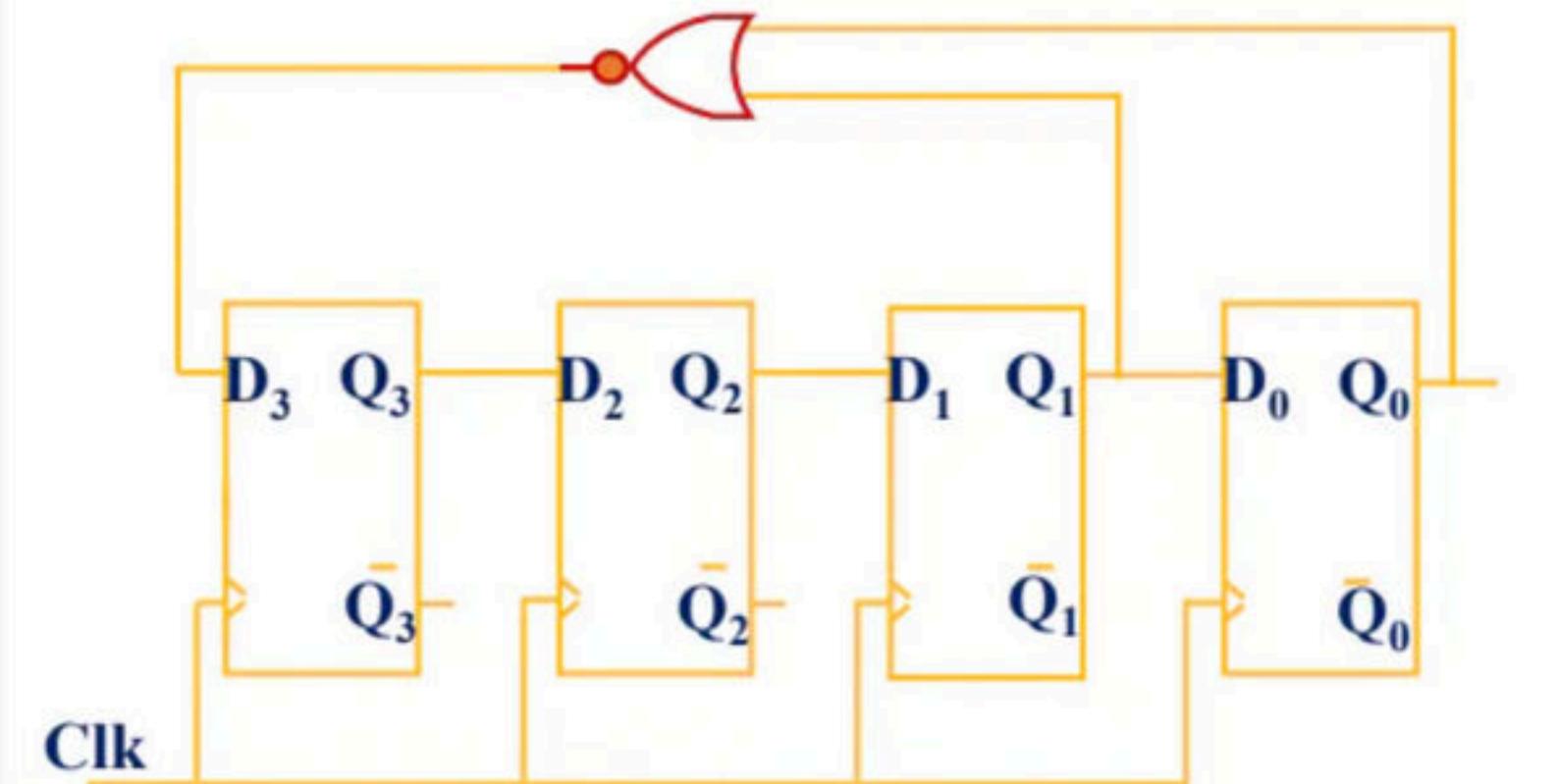
- Number of used states =
- Number of unused states =
- Mod No =
- Frequency of each FF =

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

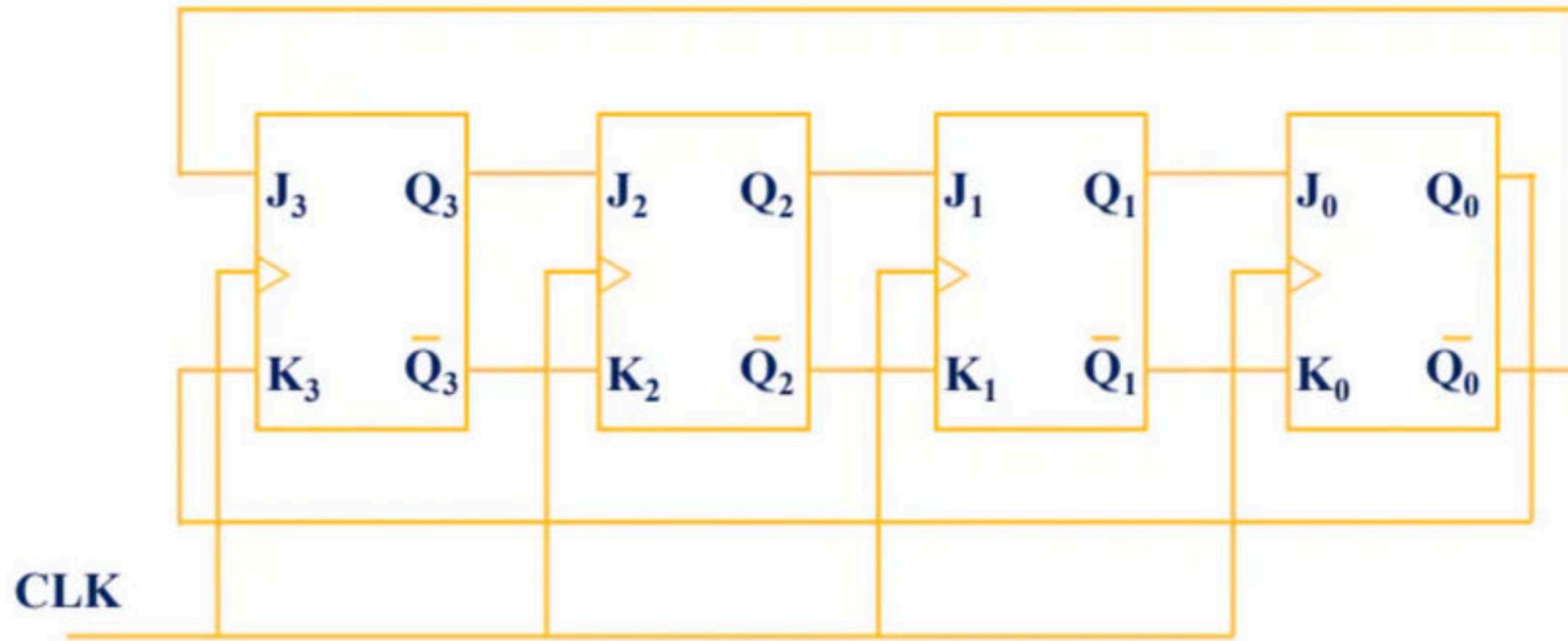
Draw back of Johnson Ring counter

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



Johnson Ring counter with JK FF



Johnson Ring counter

Twisted Ring counter

Switch tail counter

Walking Counter

Creeping counter

Mobies counter

Ring counter

1. Mod No =

2. Number of used states=

Number of unused states =

3. Time period of each FF =

4. Frequency of each FF =

5. Suffer from lock out problem

6. Decoding logic is simple

Johnson ring counter

1. Mod No =

2. Number of used states=

Number of unused states =

3. Time period of each FF =

4. Frequency of each FF =

5. Suffer from lock out problem

6. Decoding logic requires AND and NOR gates

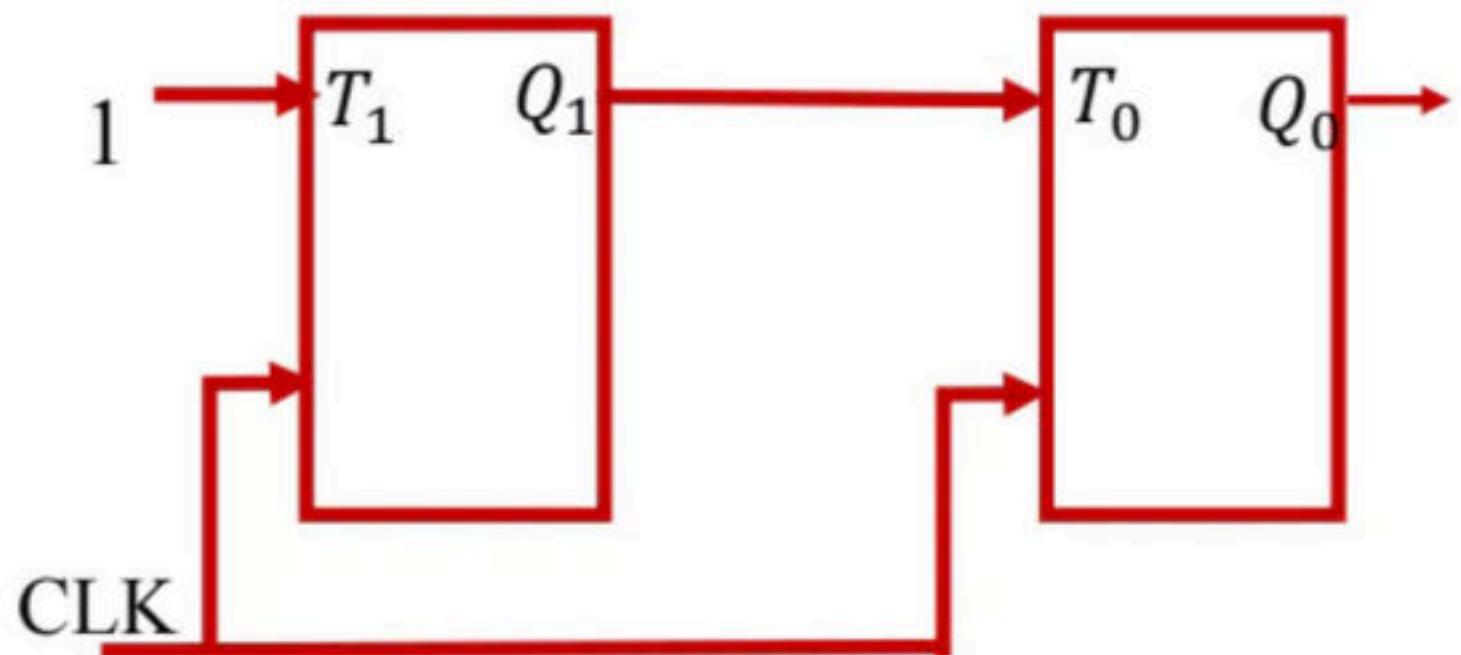
User defined counter design

Q) Design synchronous counter , whose counting sequence is

$Q_1Q_0 = 00 \rightarrow 10 \rightarrow 01 \rightarrow 11 \rightarrow 00 \rightarrow \dots$ By using T- FF

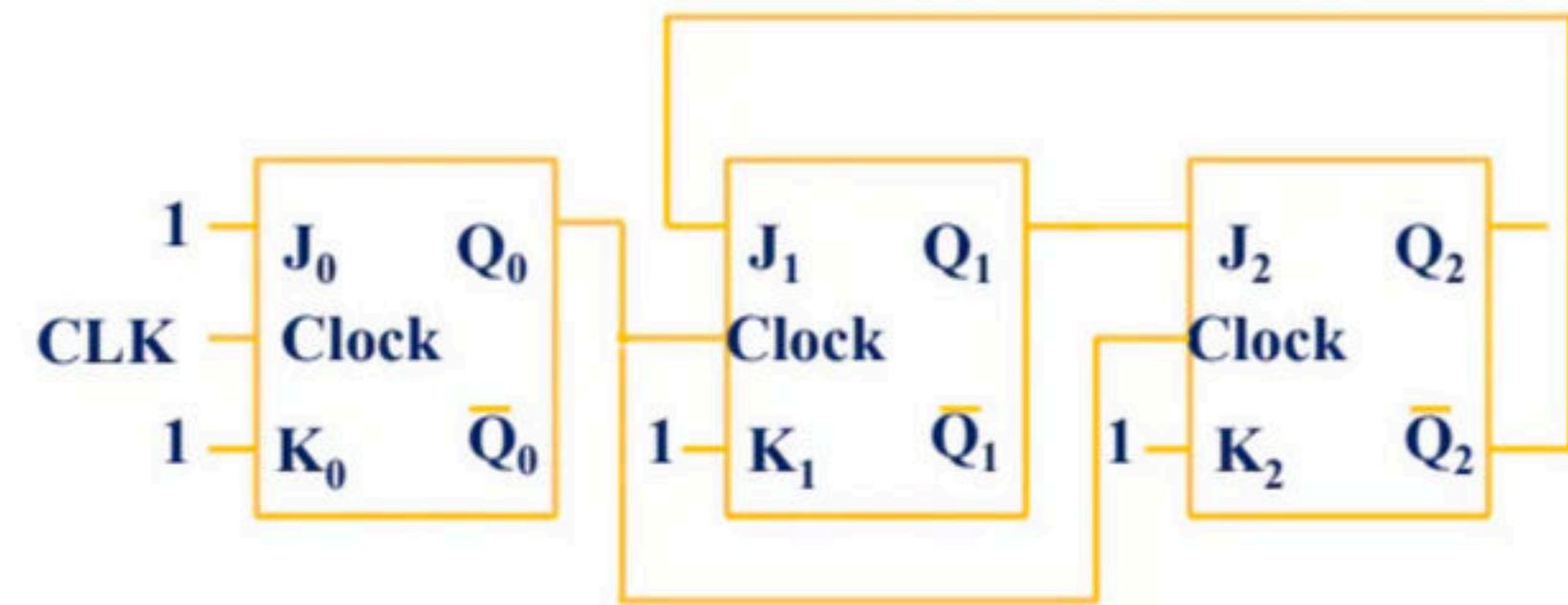
Q) Design synchronous counter , whose counting sequence is

$Q_1 Q_0 = \mathbf{00} \rightarrow \mathbf{10} \rightarrow \mathbf{01} \rightarrow \mathbf{11} \rightarrow \mathbf{00} \rightarrow \mathbf{10} \dots$ By using D- FF



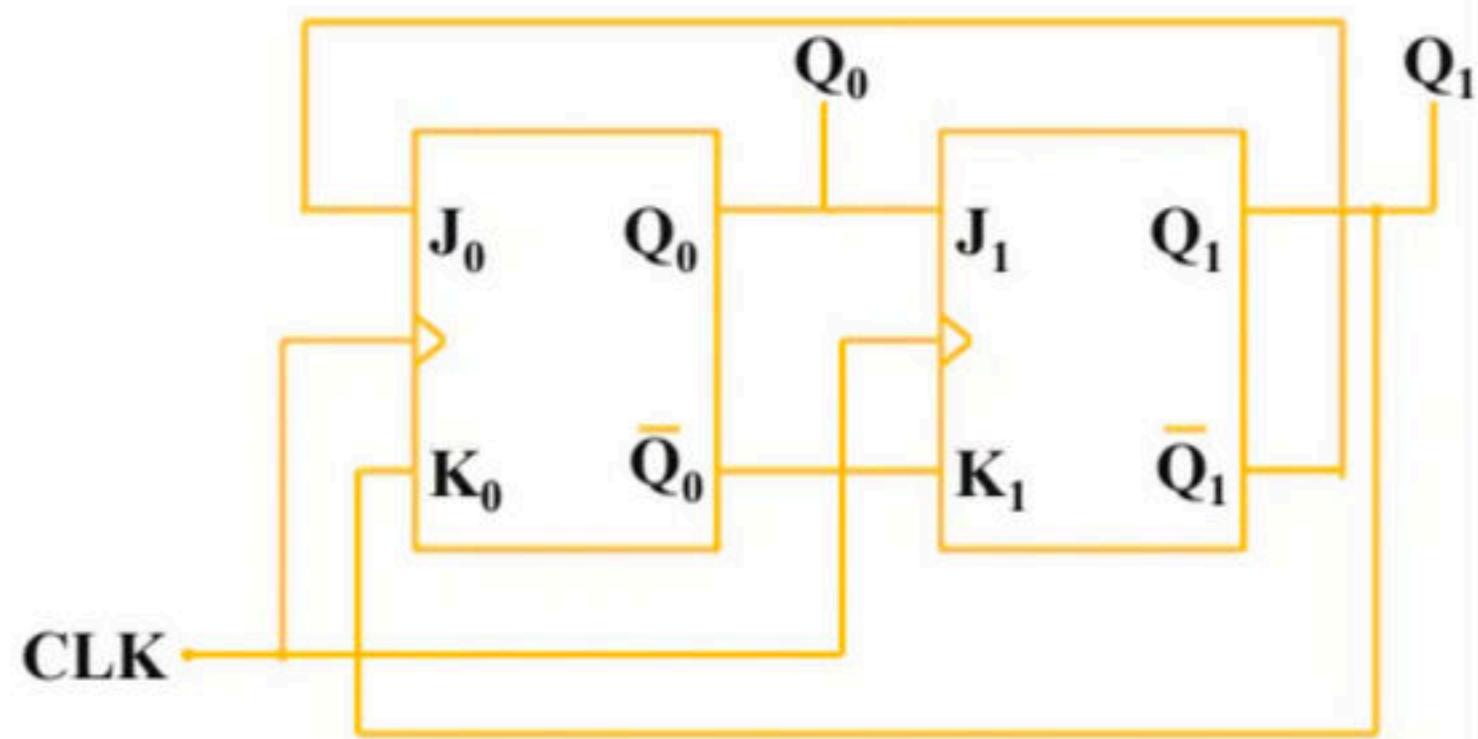
Q) Find the counting sequence of the following
If the initial state of the counter $Q_1Q_0 = 00$ then
the state of counter after
a) 236 clocks b) 251 clocks c) 333 clocks

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



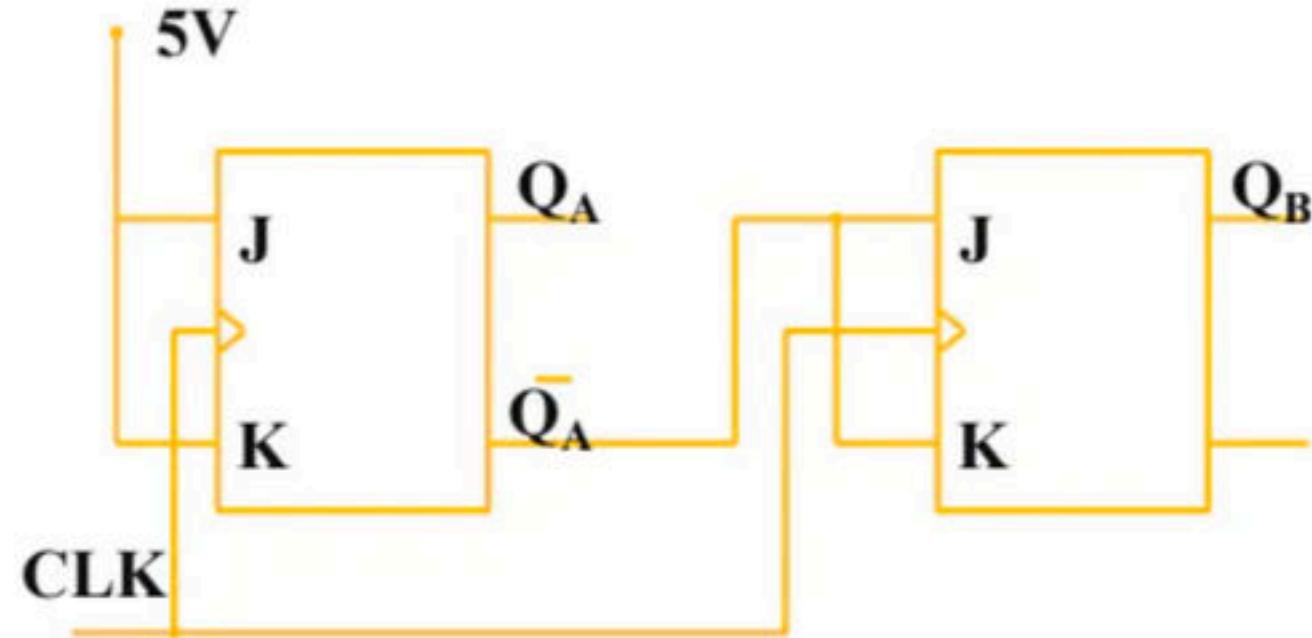
Q) In the following sequential circuit, the initial state (before the first clock pulse) of the circuit is $Q_1Q_0 = 00$. The state (Q_1Q_0), immediately after the 333rd clock pulse is.

- (A) 00
- (B) 01
- (C) 10
- (D) 11



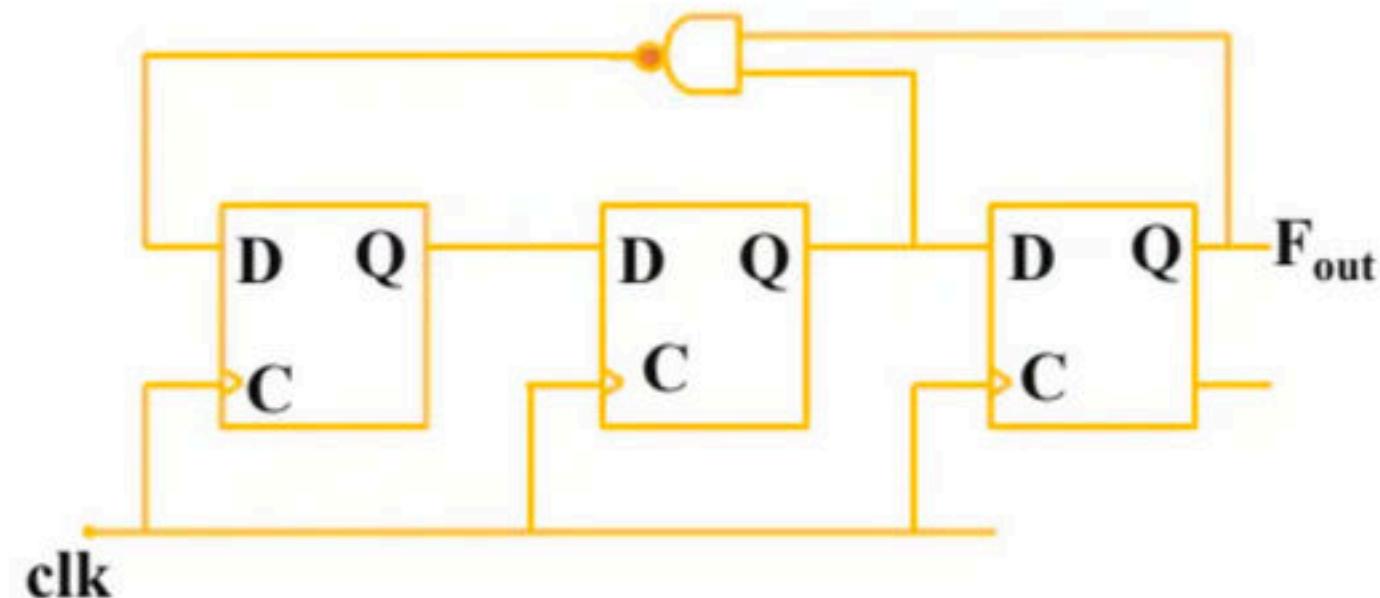
Q). The current state Q_A Q_B of a two JK flip-flop system is 00. Assume that the clock rise-time is much smaller than the delay of the JK flip-flop. The next state of system is.

- (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.

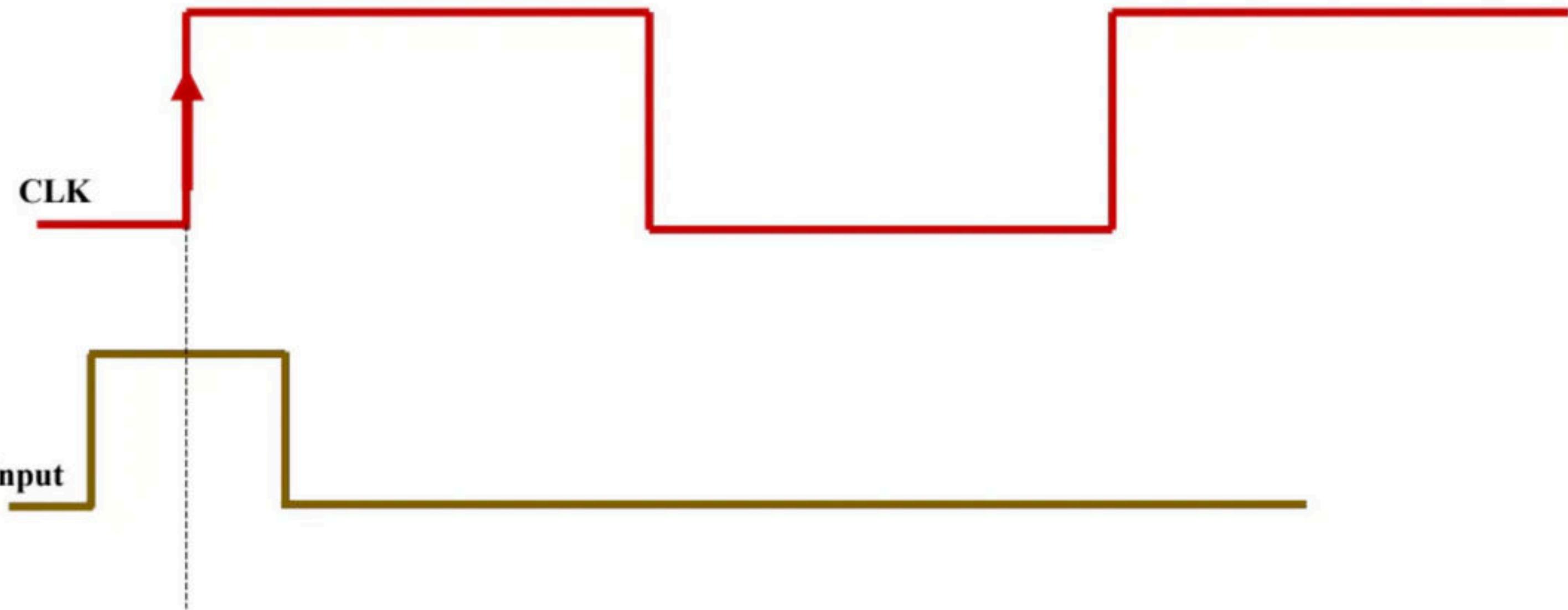


Set up time

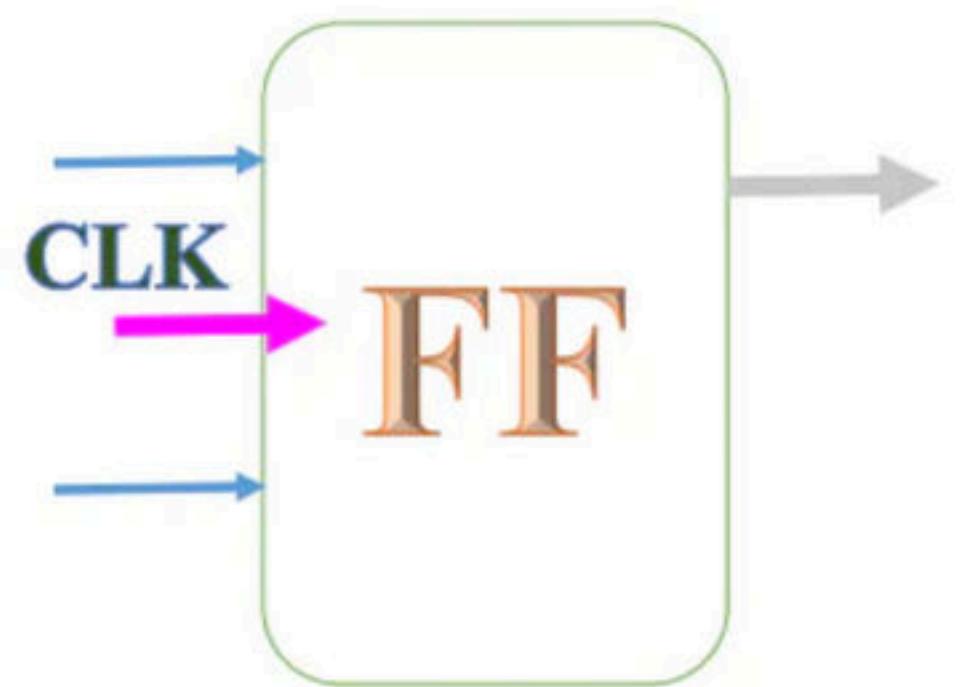
It is the minimum amount of time before the active edge of the clock, the input signal should remain constant

Hold time

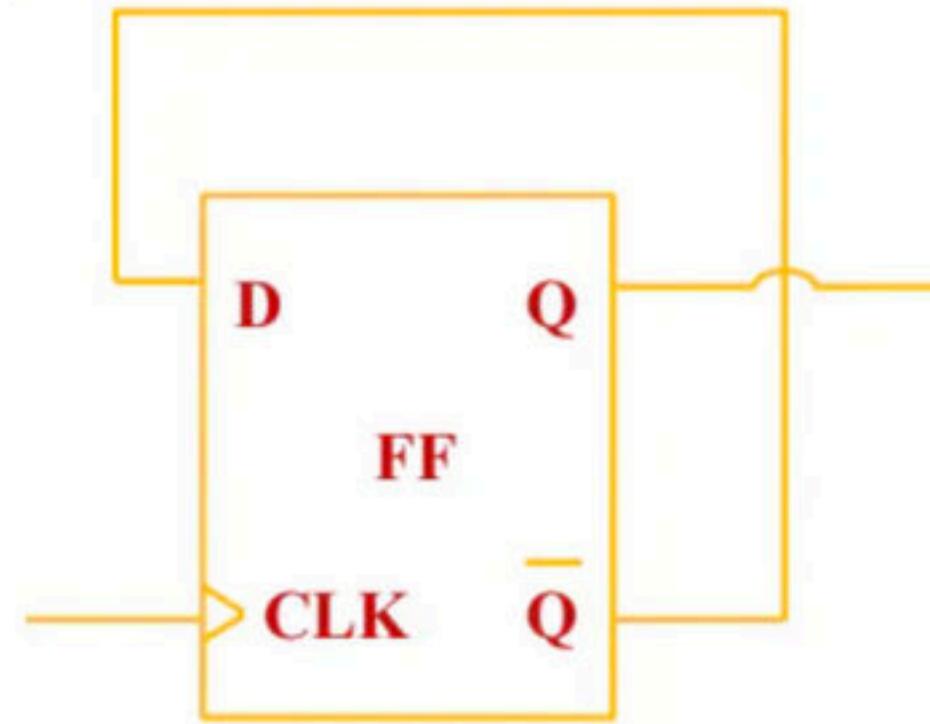
It is the minimum amount of time after the active edge of the clock, the input signal should remain constant



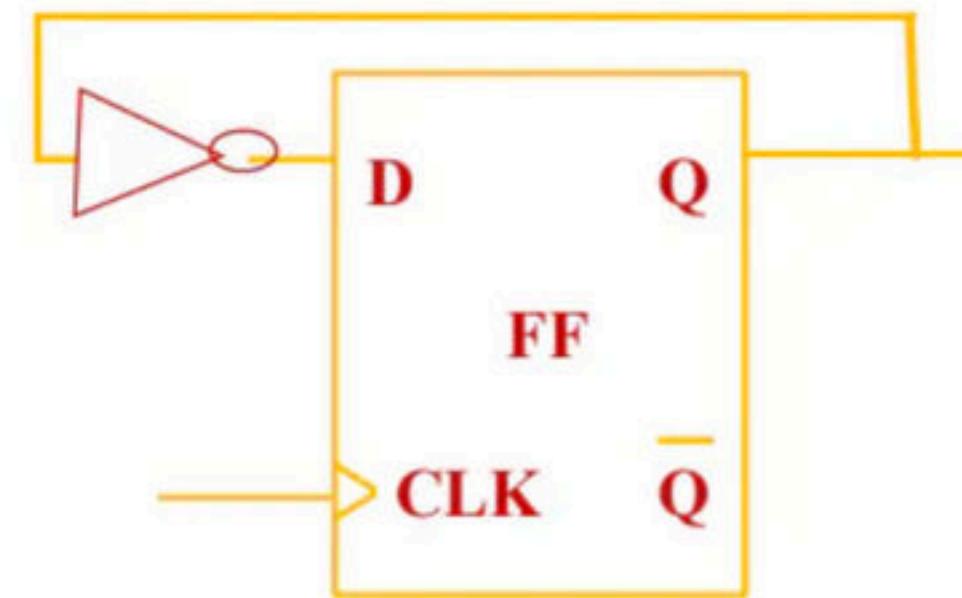
Delay in single FF



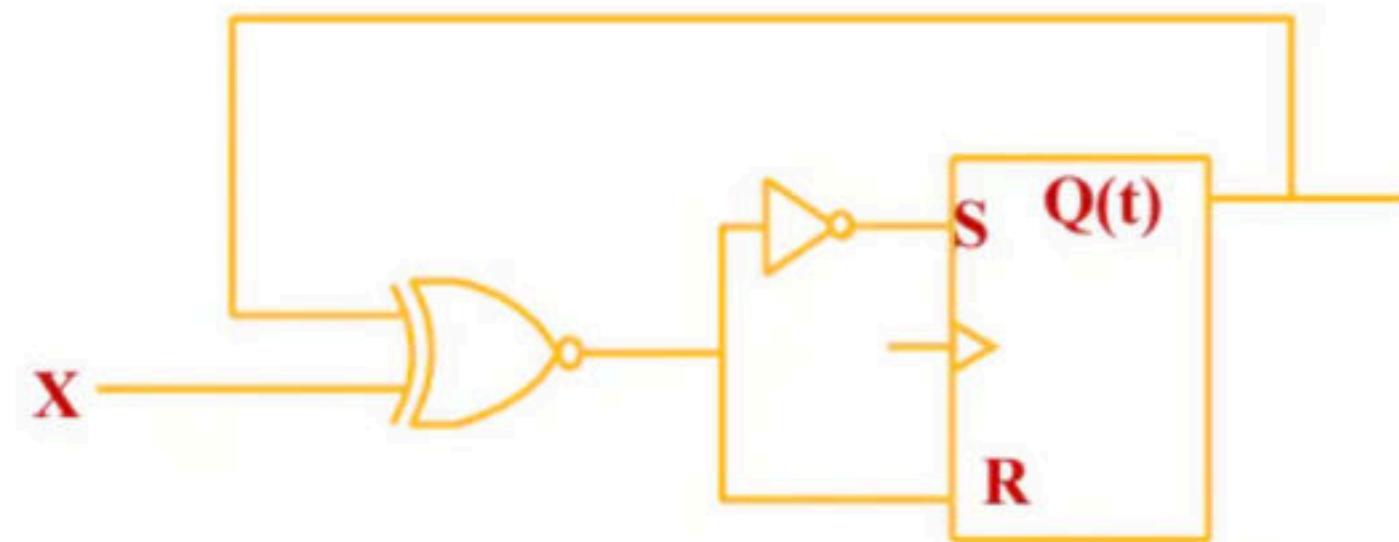
Q) Find the maximum frequency of operation for the following circuit, if the propagation delay of FF is 20ns and setup time and hold time are 10ns each .



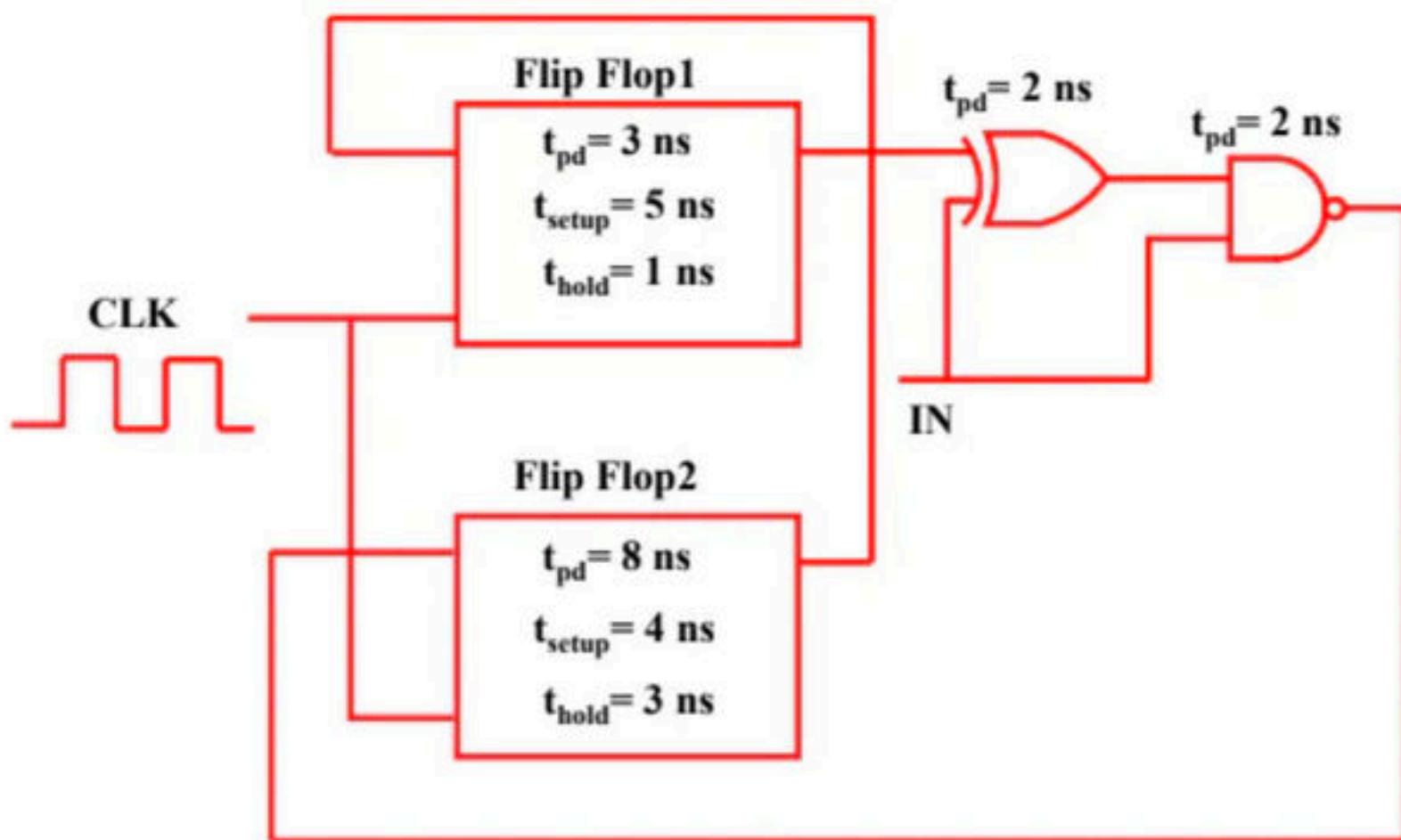
Q) Find the maximum frequency of operation for the following circuit, if the propagation delay of FF is 20ns and inverter is 5ns .



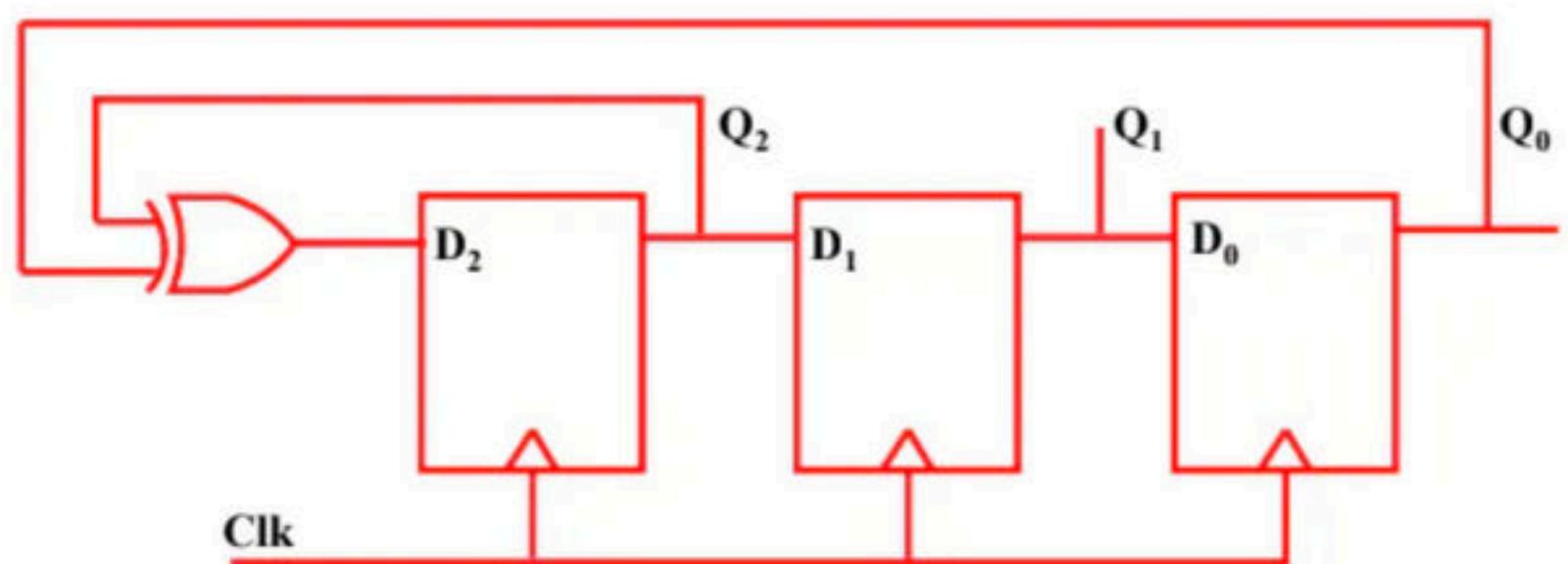
Q) Find the maximum frequency of operation for the following circuit, if the propagation delay of FF is 20ns , XNOR gate is 10ns and inverter is 5ns .



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

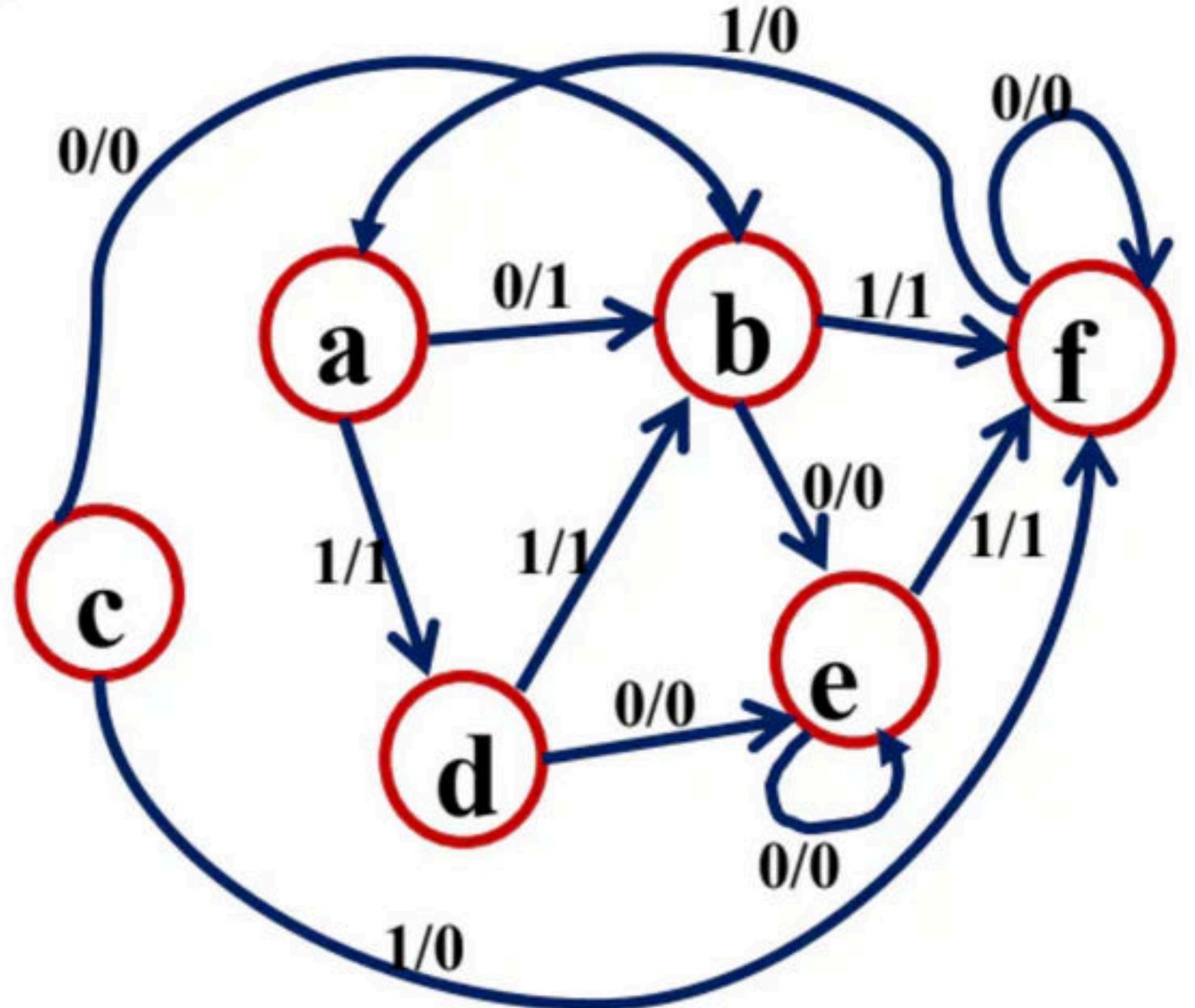
$$Z(t) = f [s(t)]$$

State Diagram

State table

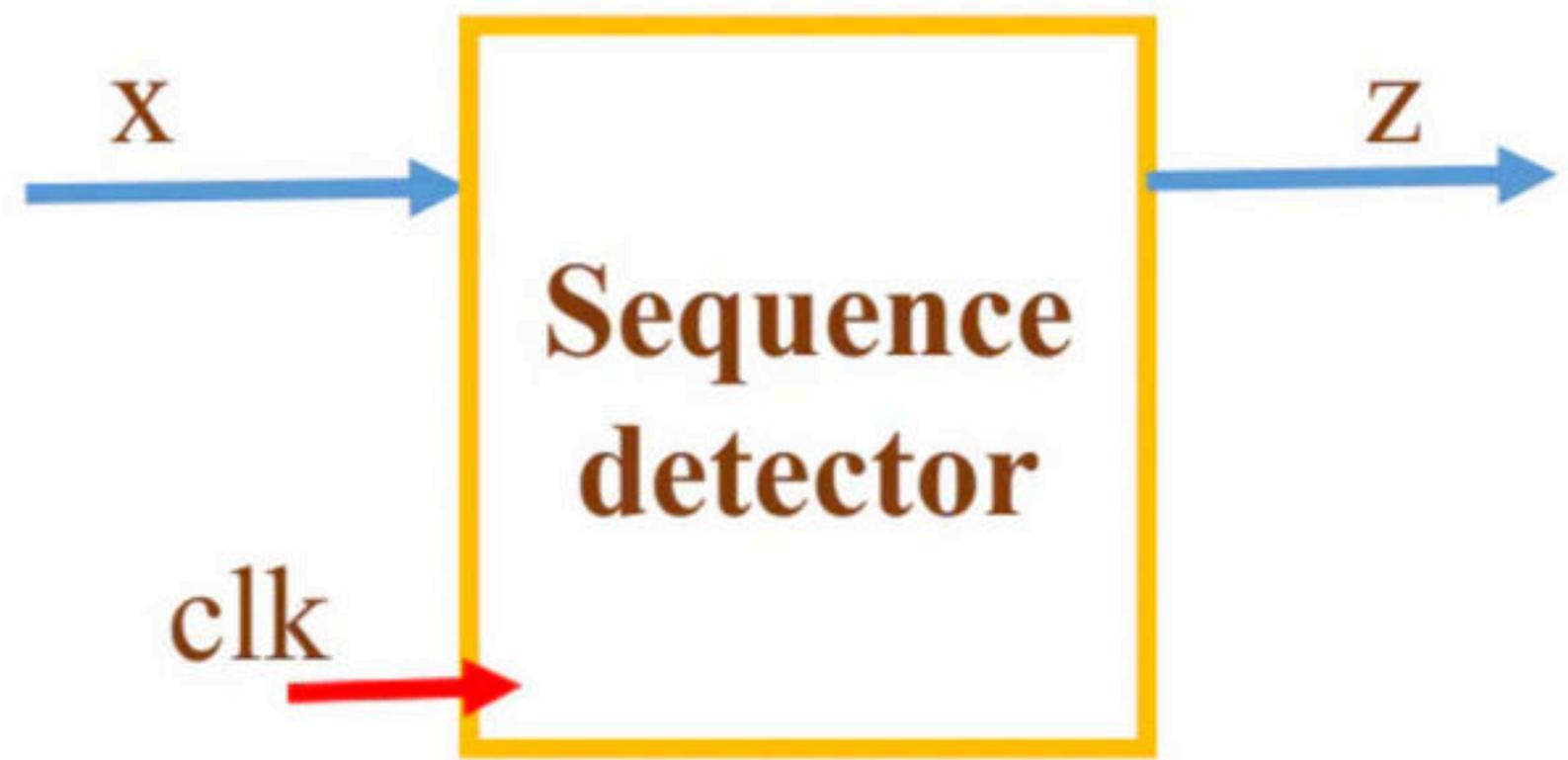
Present state	Next State		Output
	X = 0	X = 1	
a	a	b	0
b	b	c	0
c	d	c	0
d	a	d	1

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

$$X = 1101101101$$

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

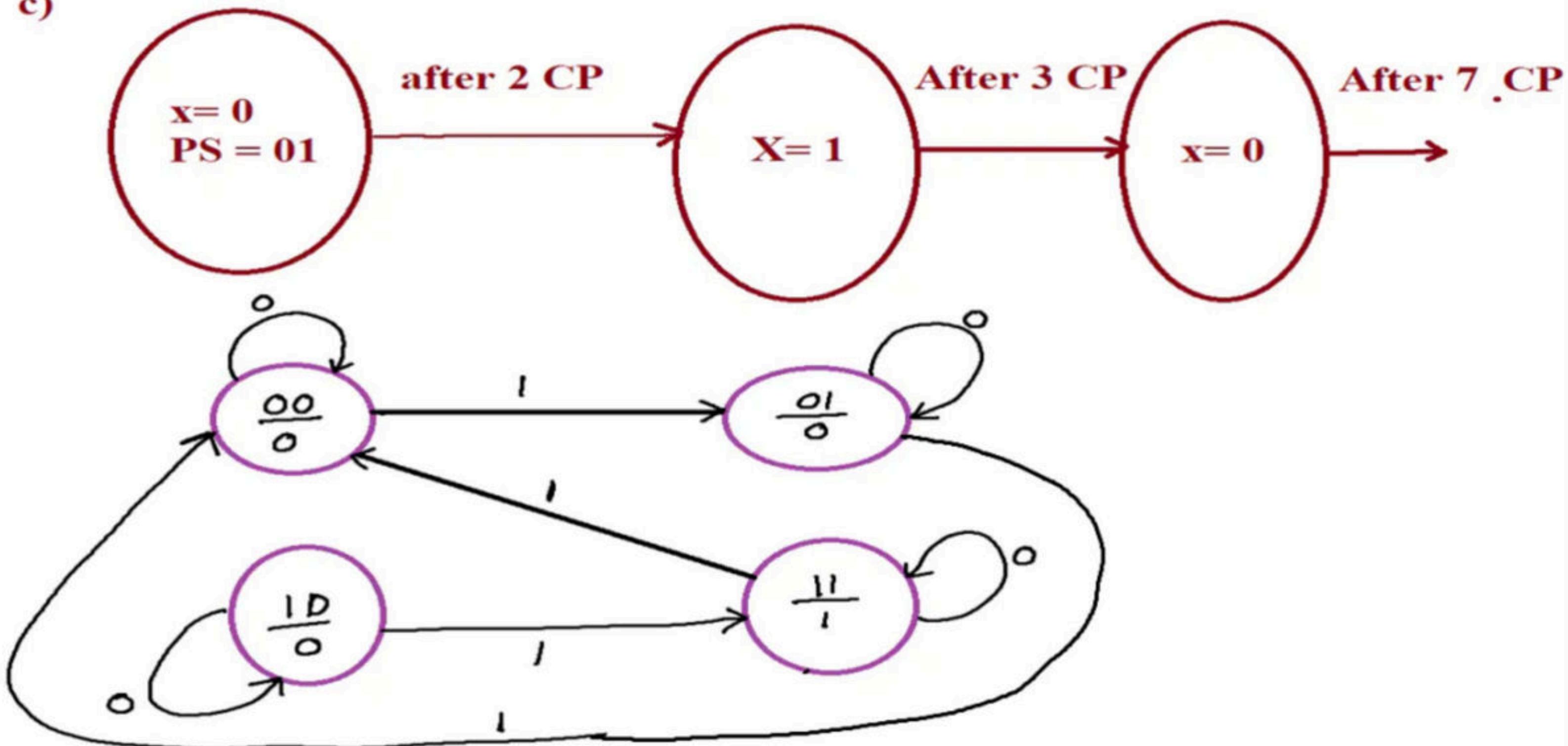
Q. Develop the non overlapping sequence detector and state table to detect 1010 by using Moore modal

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

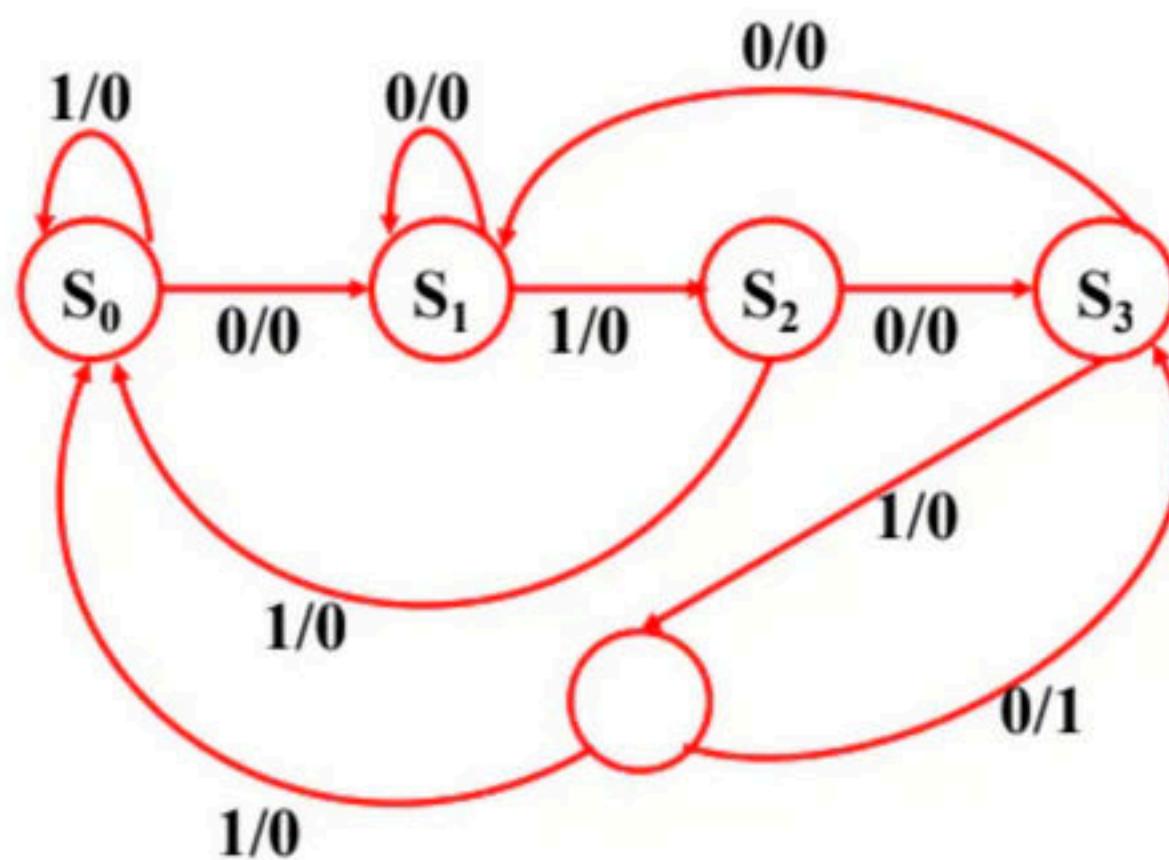
consider the state diagram find

- 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)



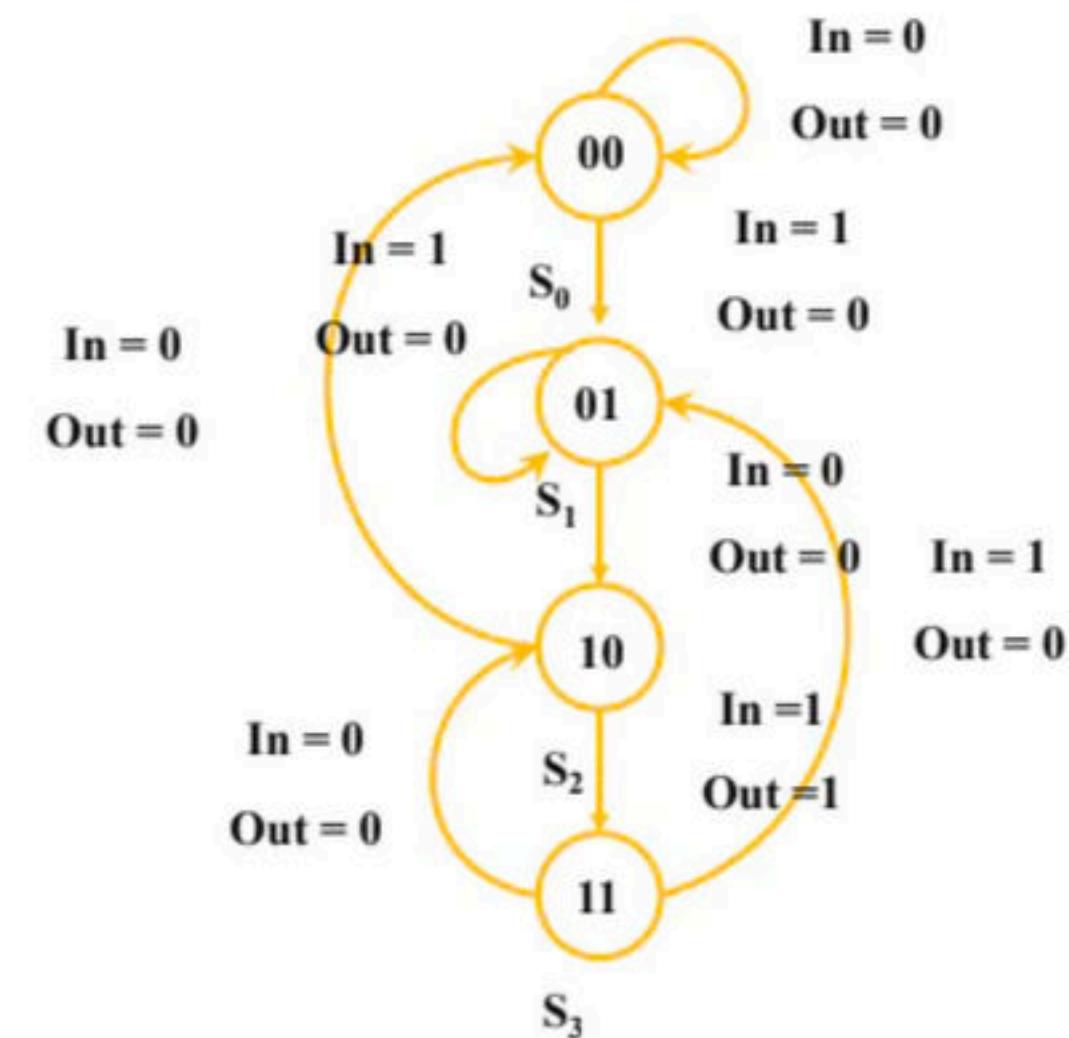
Q. The state diagram of a sequence detector is shown below. State S_0 is the initial state of the sequence detector. If the output is 1, then

- (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 10101101001101, 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

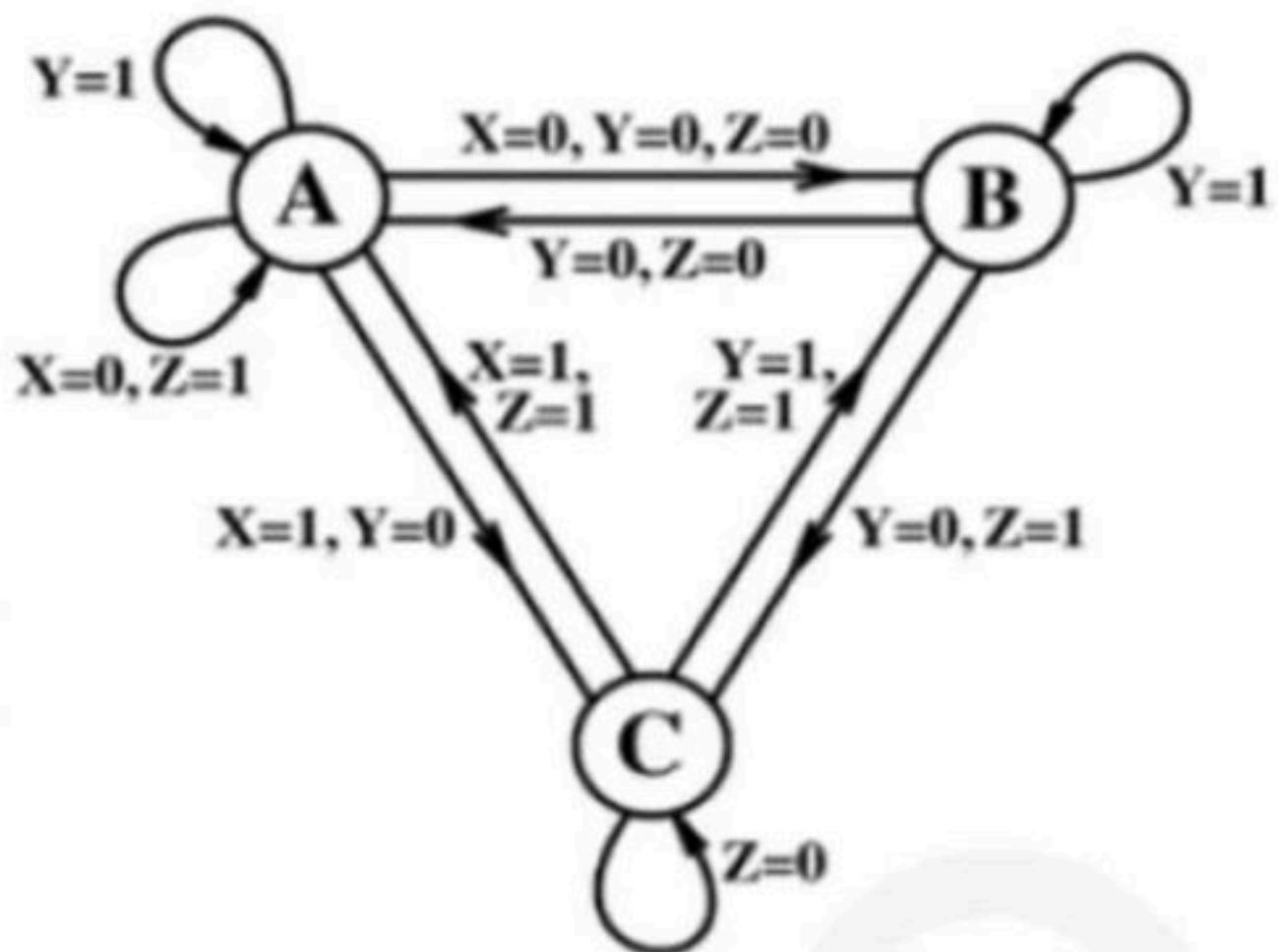
Q. Develop the overlapping sequence detector and state table to detect 0011 by using mealy modal

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

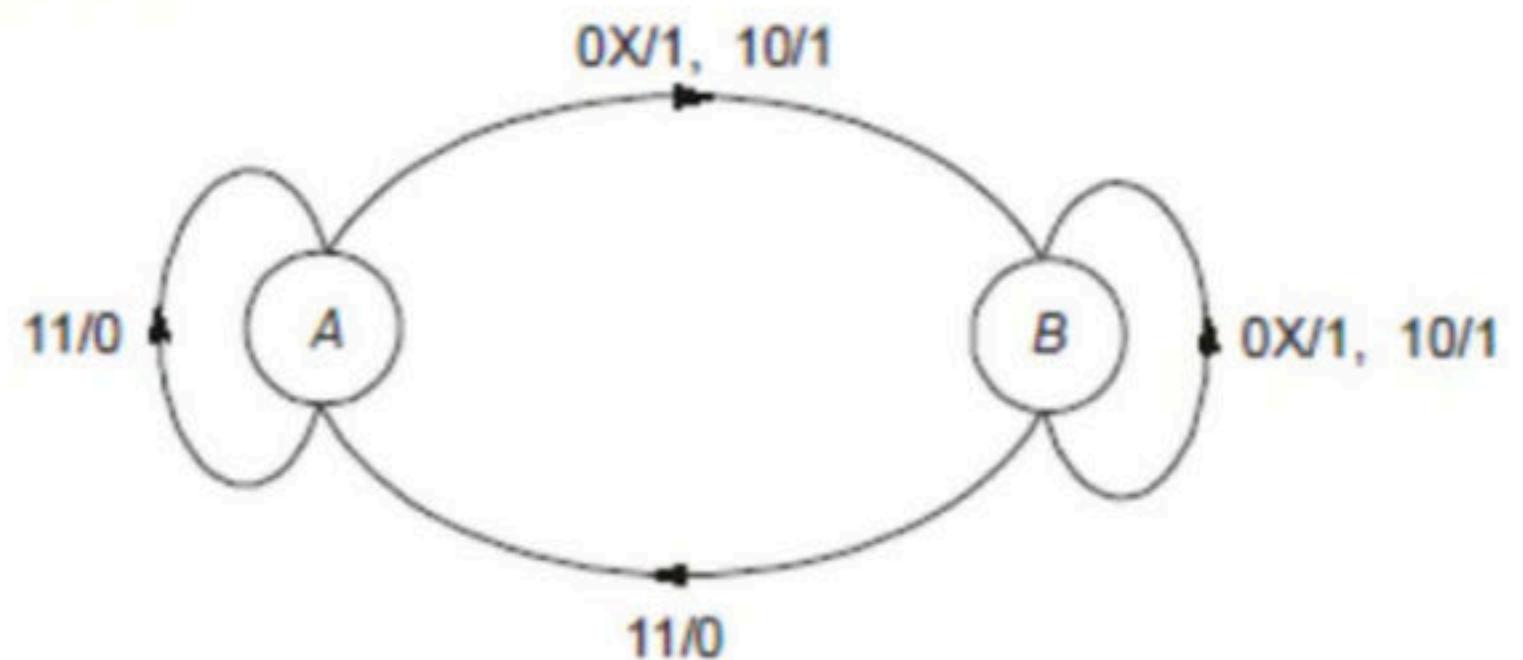
The state transition diagram for a finite state machine with states A, B and C, and binary inputs X, Y and Z, is shown in the figure.

Which one of the following statements is correct?



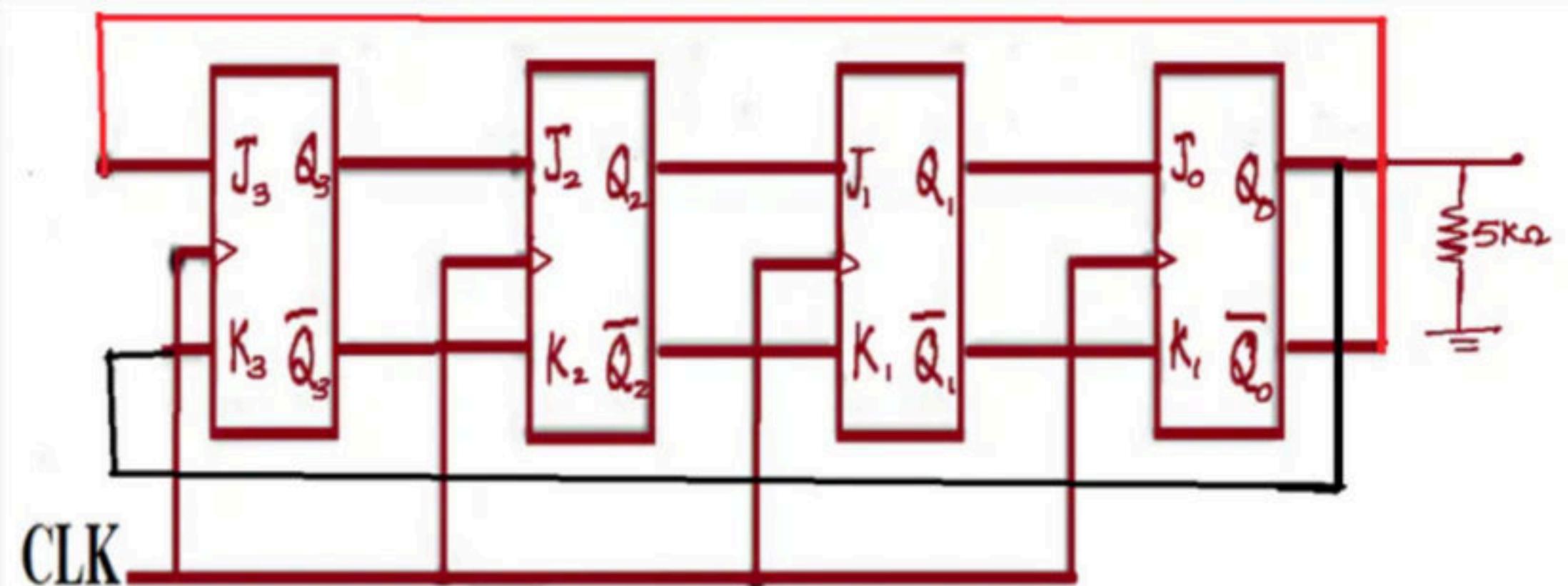
- A. Transitions from State A are ambiguously
- B. Transitions from State B are ambiguously
- C. Transitions from State Care ambiguously
- D. All of the state transitions are defined unambiguously.

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

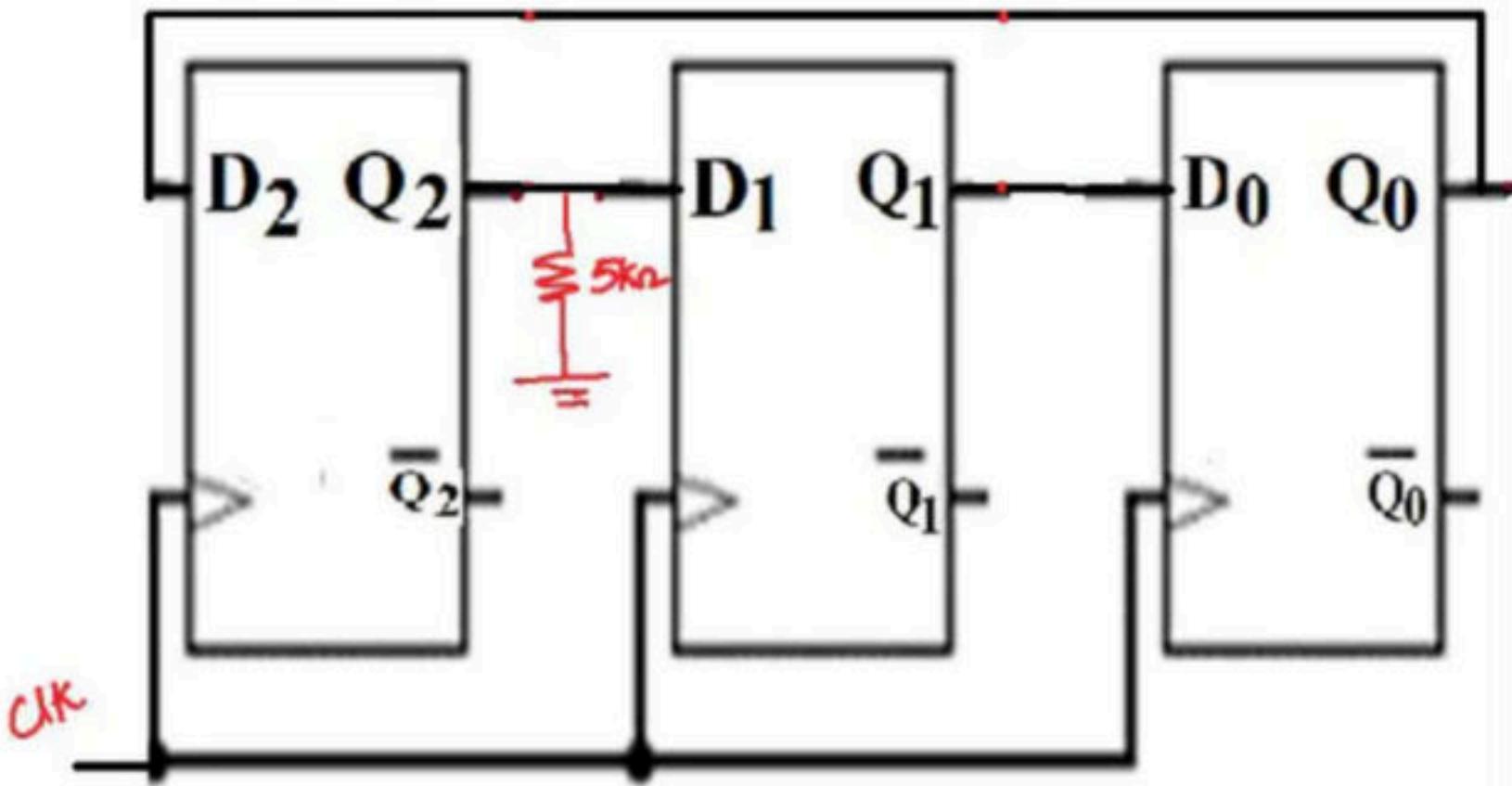


Q) Find

- power dissipation $5 \text{ K}\Omega$
- frequency of each FF
- If the transition probability of data bit at each is 0.7 , find the average value of Q_1

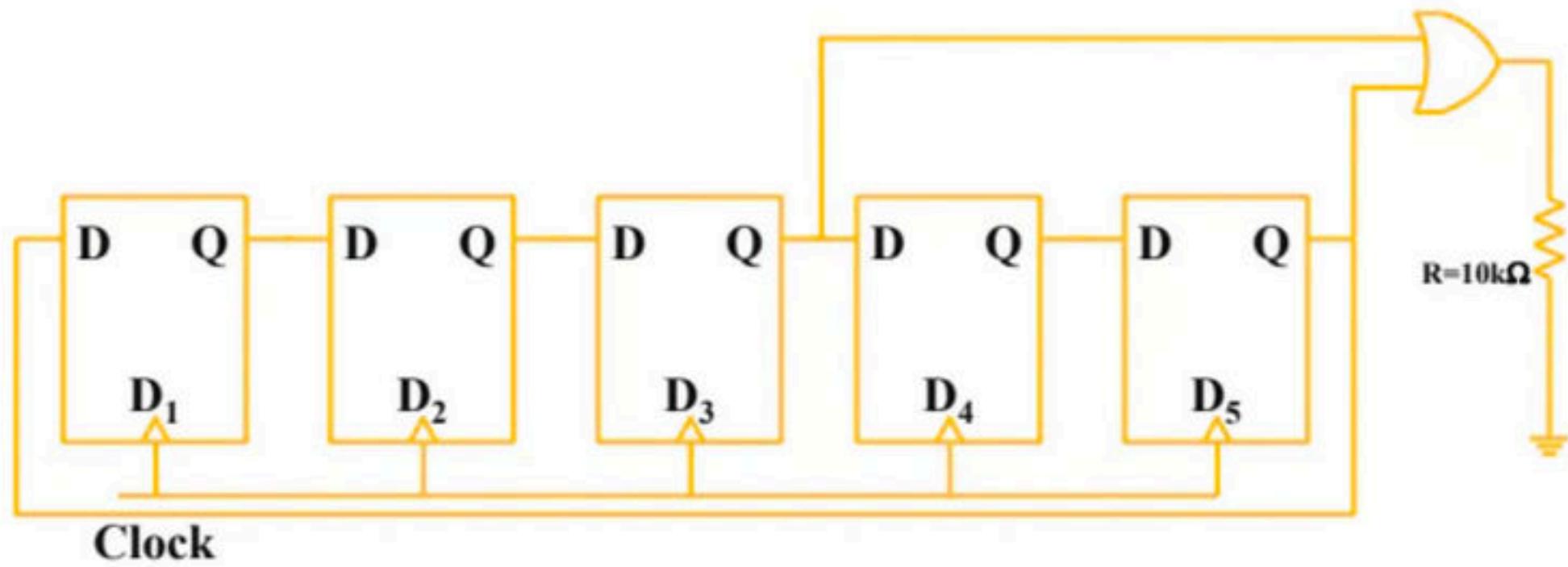
Q) If the initial state of $Q_2 Q_1 Q_0 = 110$, then find

- a) MoD No
- b) MSV of wave form of Q_1
- c) P_{diss} in $5k\Omega$
- d) Average value of wave form of Q_0
- e) Frequency of $Q_2 Q_1 Q_0$

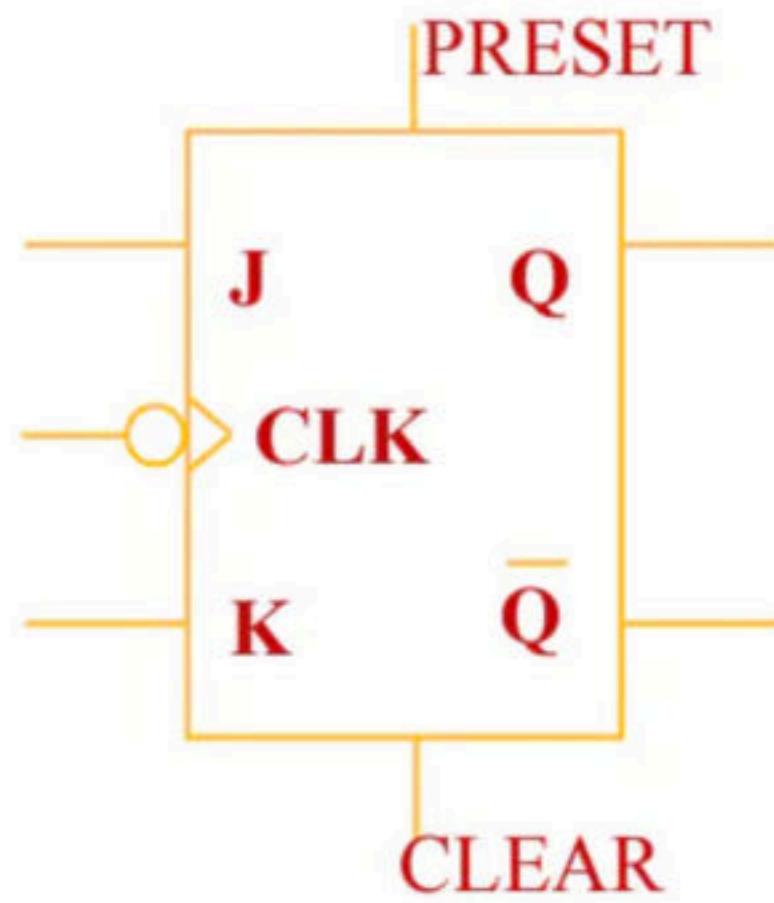


14. Assume that all the digital gates in the circuit shown in the figure are ideal, the resistor $R = 10\text{k}\Omega$ and the supply voltage is 5V.

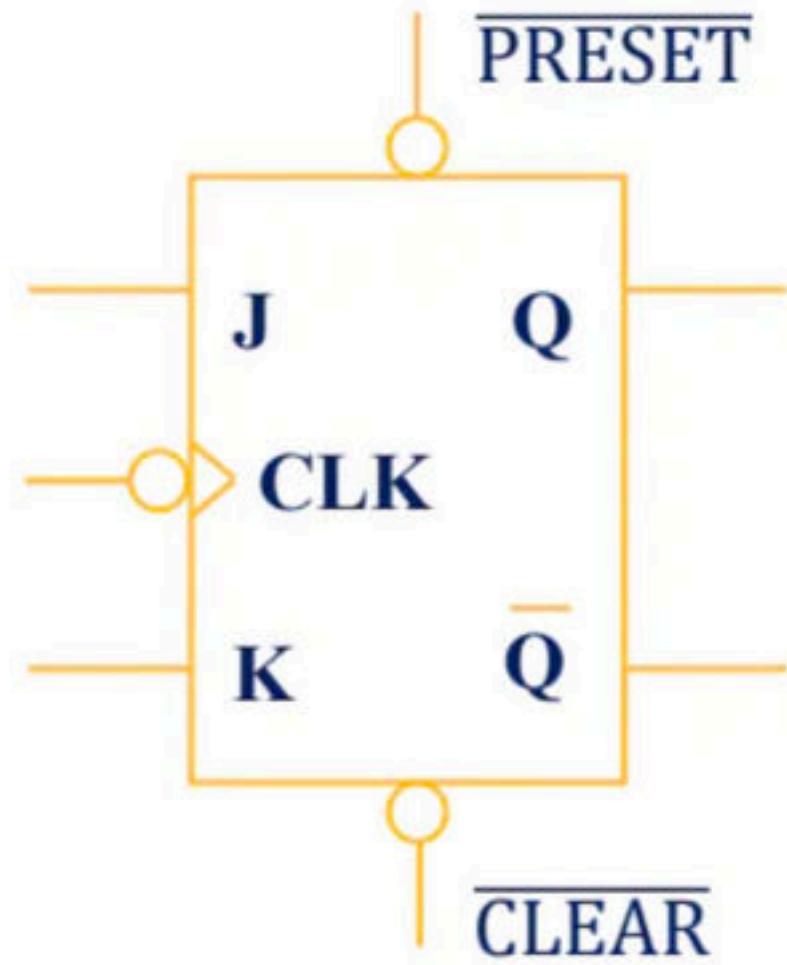
The D flip-flops D_1, D_2, D_3, D_4 and D_5 are initialized with logic values 0, 1, 0, 1 and 0, respectively. The clock has a 30% duty cycle. The average power dissipated (in m W) in the resistor R is _____.



Asynchronous Clear and Asynchronous Preset

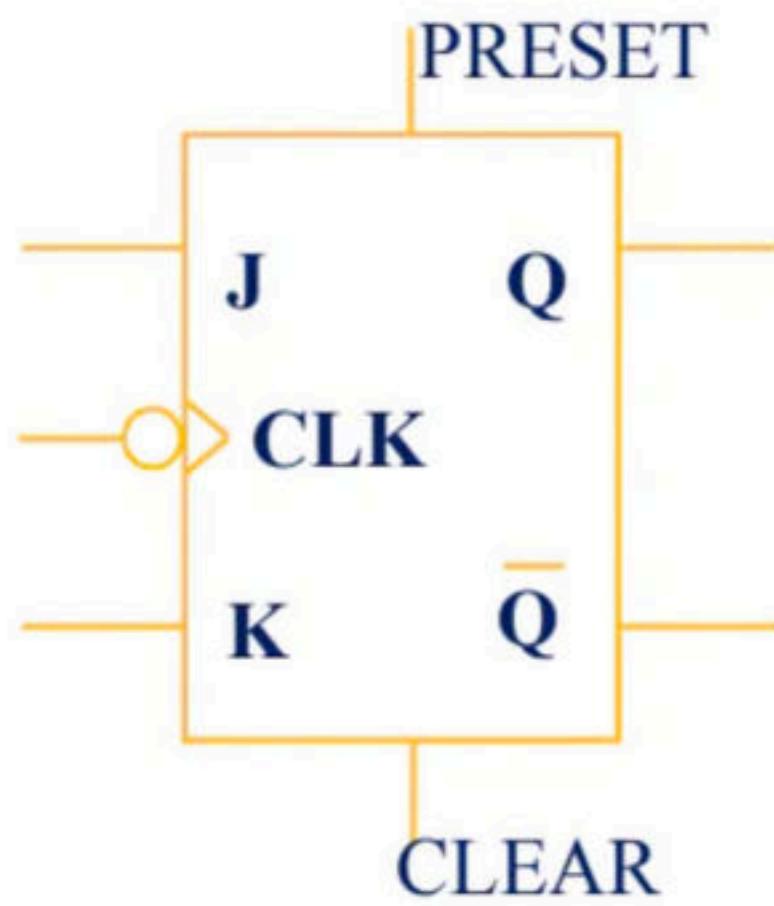


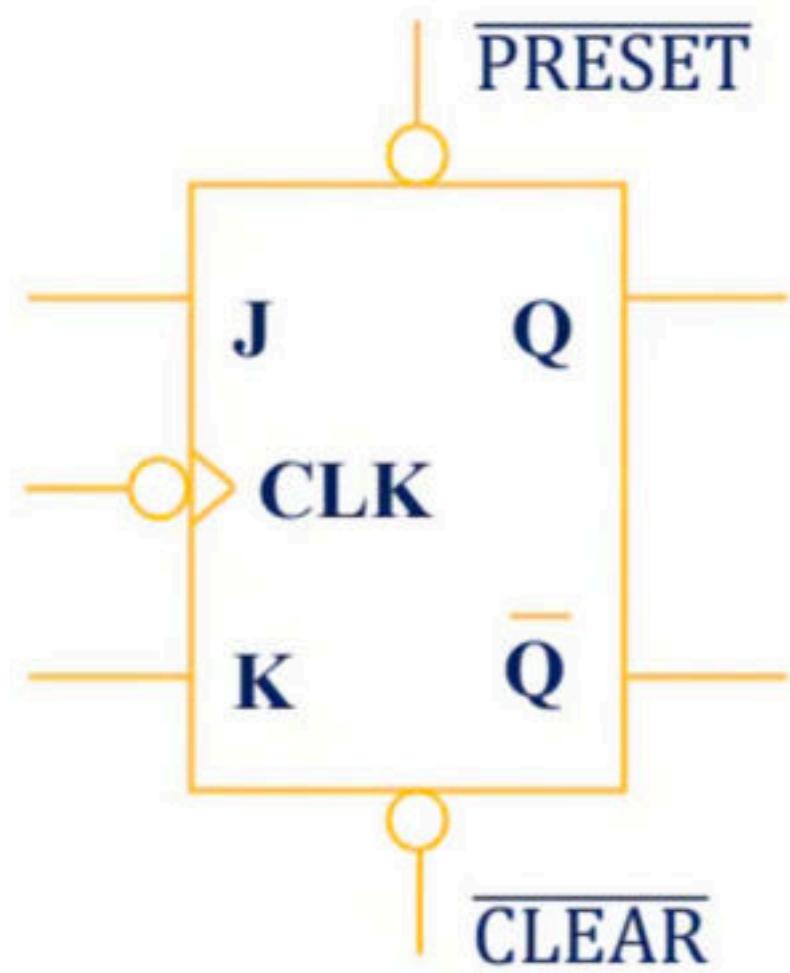
PRESET	CLEAR	FF response



PRESET	CLEAR	FF response

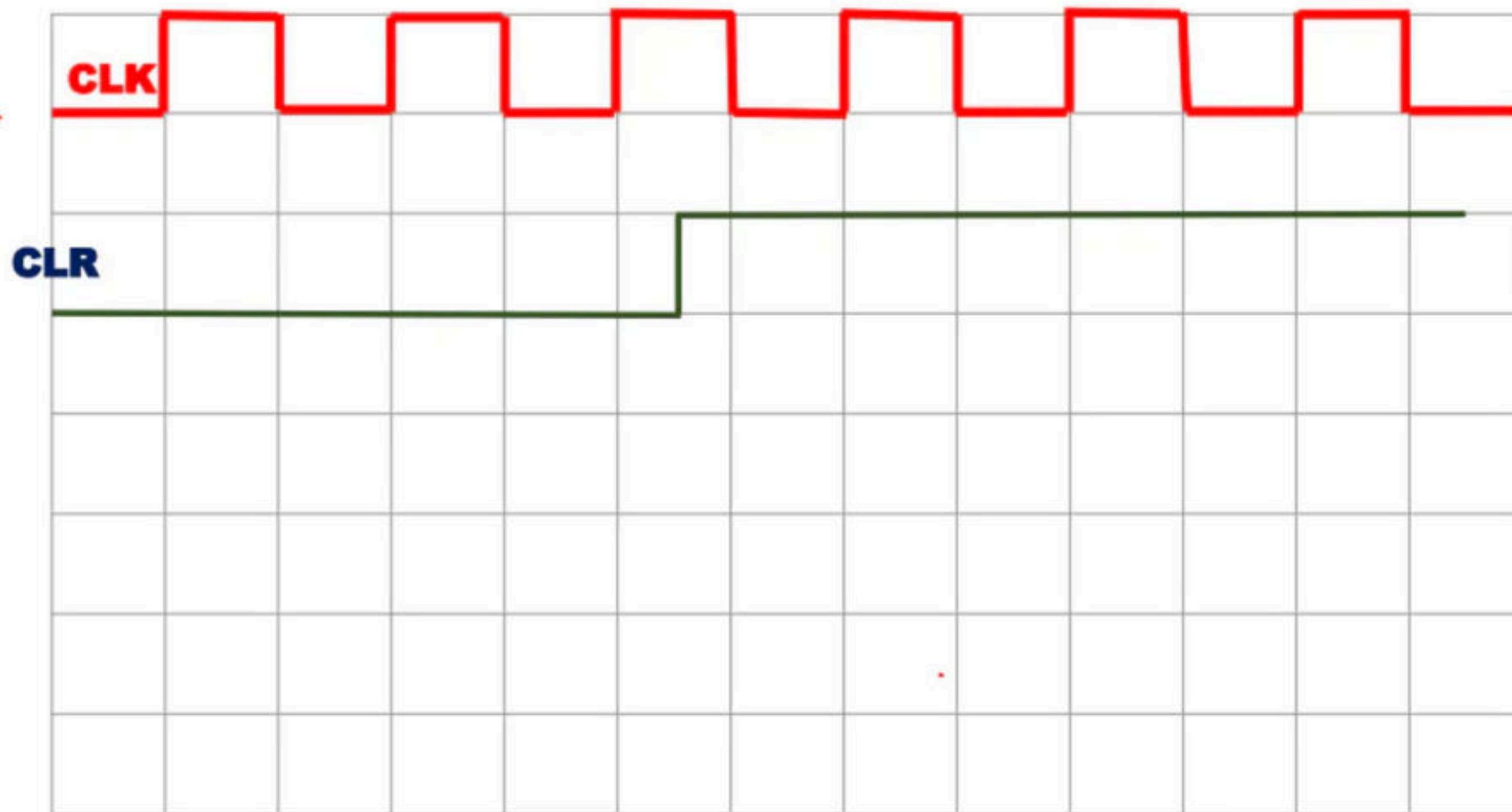
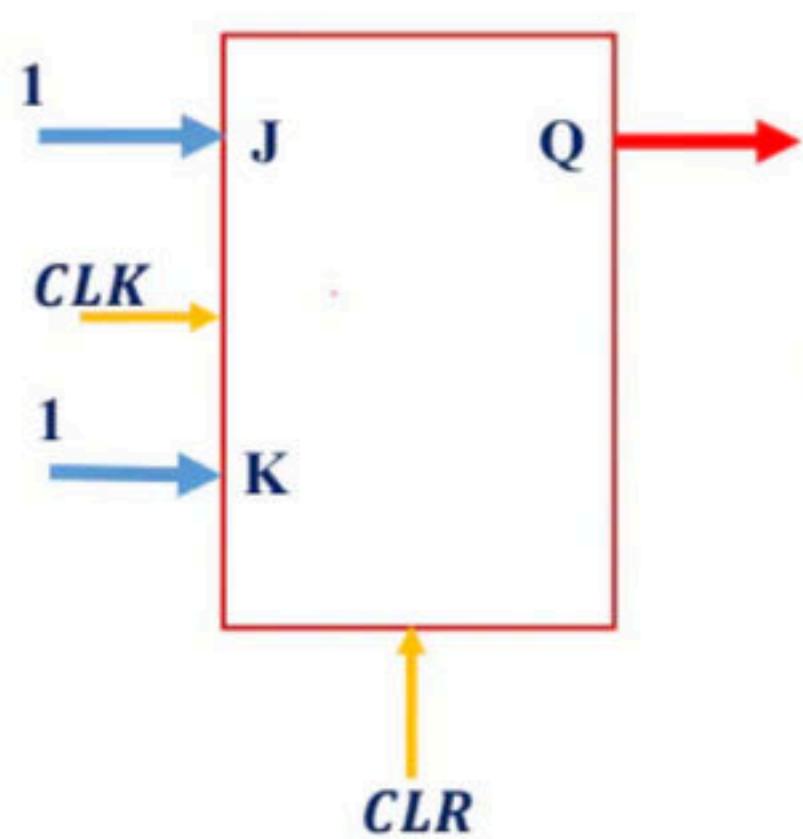
**Synchronous Clear
and
Synchronous Preset**





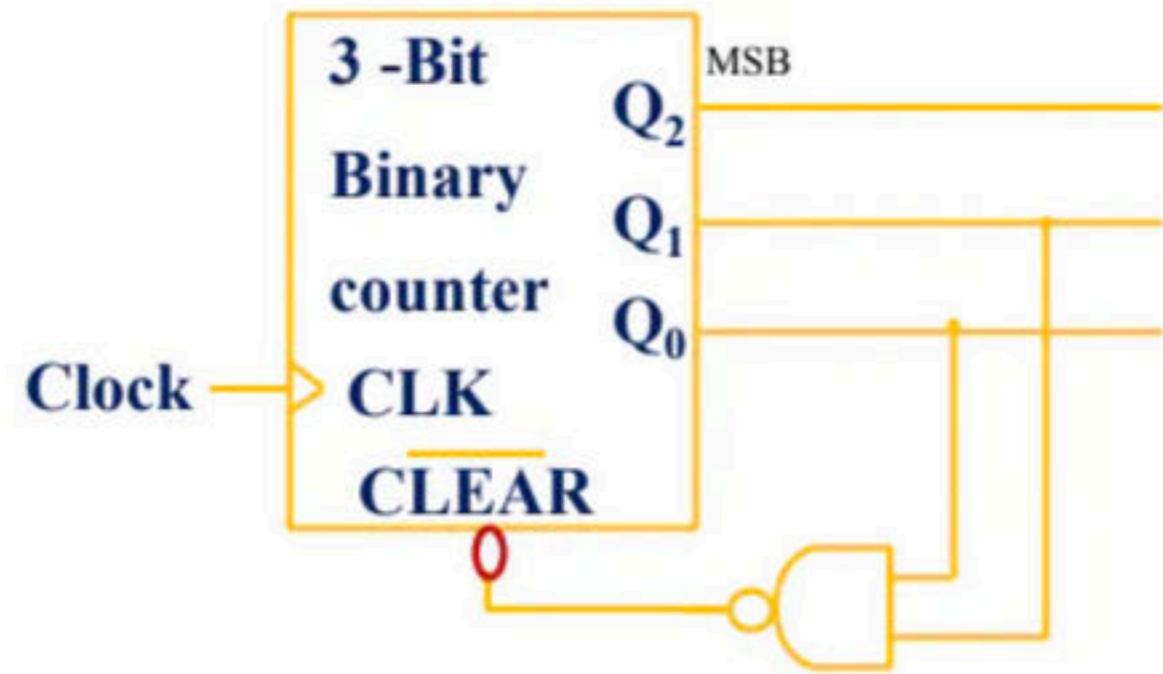
Q) Draw the output wave form of the JK FF ,

- 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 $(tpd)_{comb} \neq nTclk$

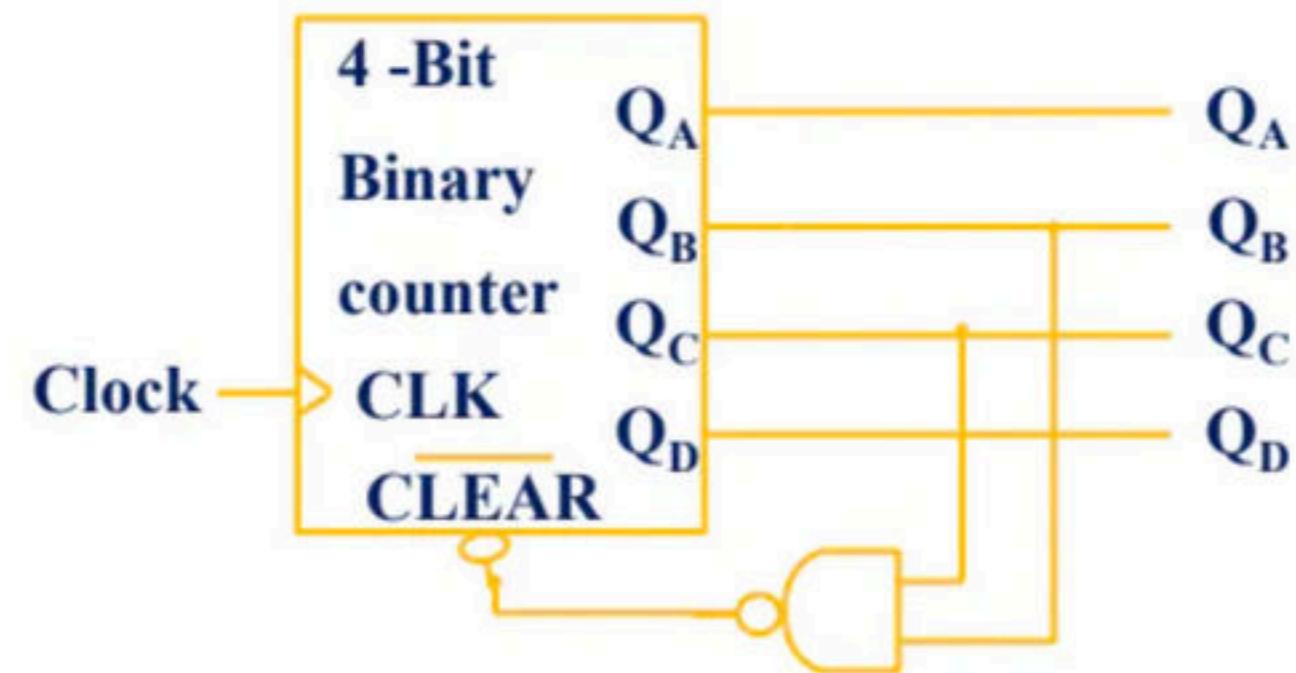
MoD No of Synchronous counter = MoD No of Asynchronous counter + 1

Assumptions

1. Consider Up counter , if not mentioned
2. $Q(\text{subscript high}) = \text{MSB}$, if not mentioned
3. Assume Asynchronous CLR , if not mentioned
4. Assume $(\text{tpd})_{\text{comb}} < \text{Tclk}$, 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 _____.

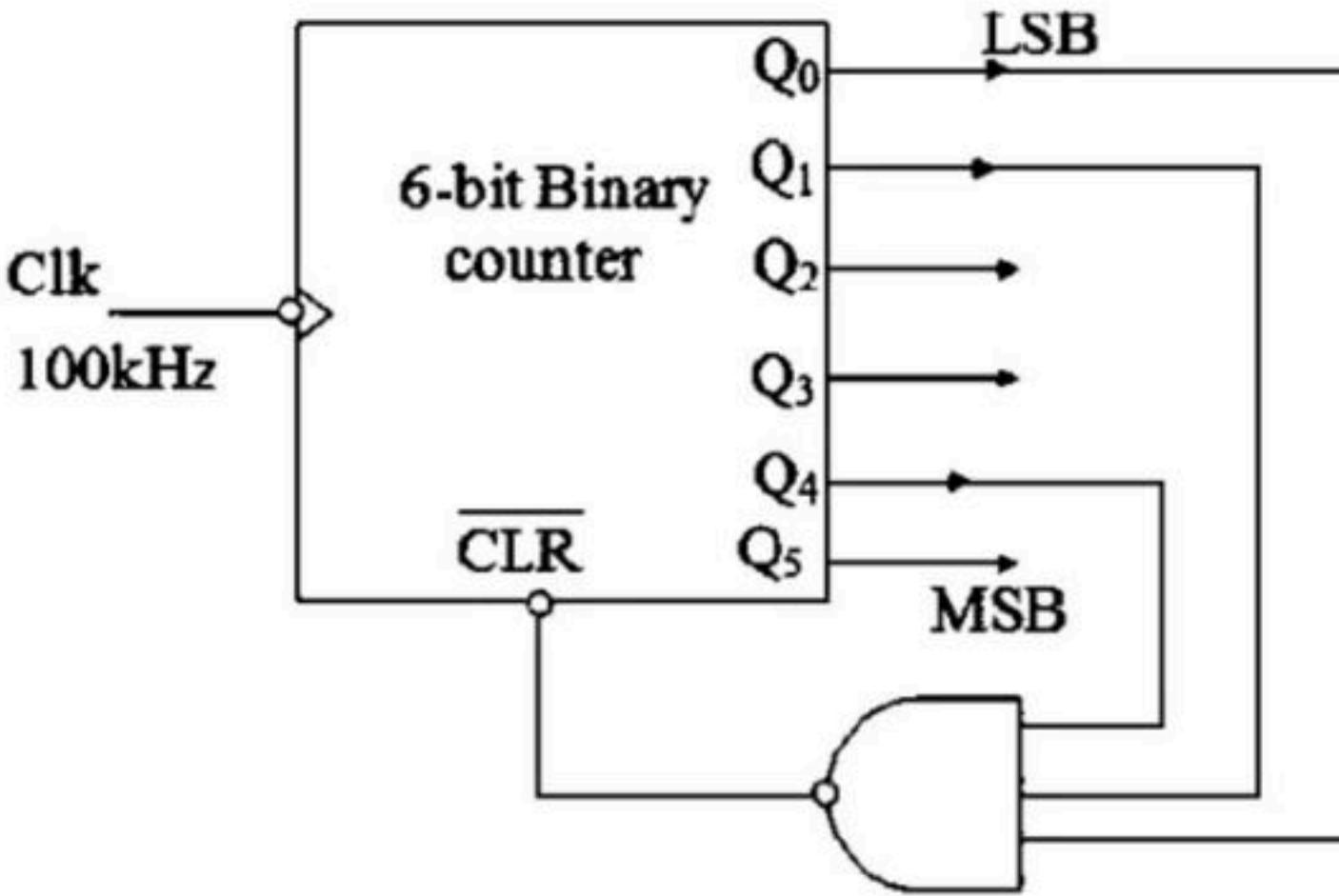


Q. The figure shows a binary counter with synchronous clear input with the decoding logic shown, the counter works as a

- (A) mod-2 counter
- (B) mod-4 counter
- (C) mod-5 counter
- (D) mod-6 counter



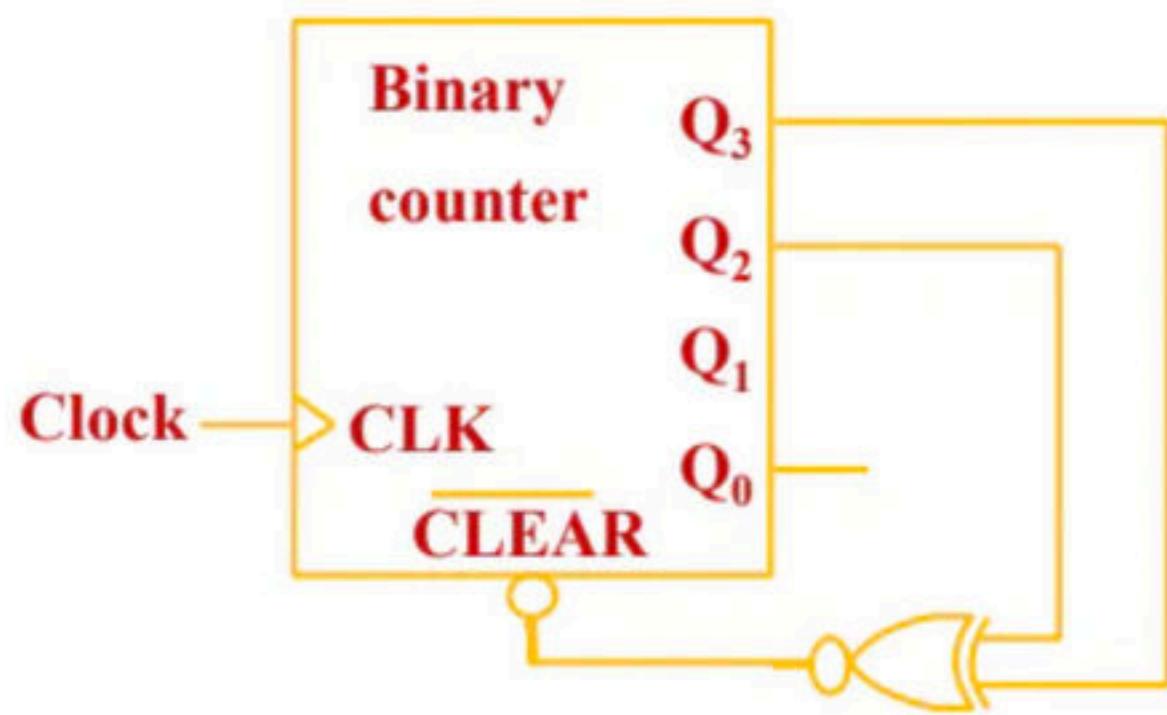
9. A mod K counter using Asynchronous Binary up counter with synchronous clear input is shown below



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

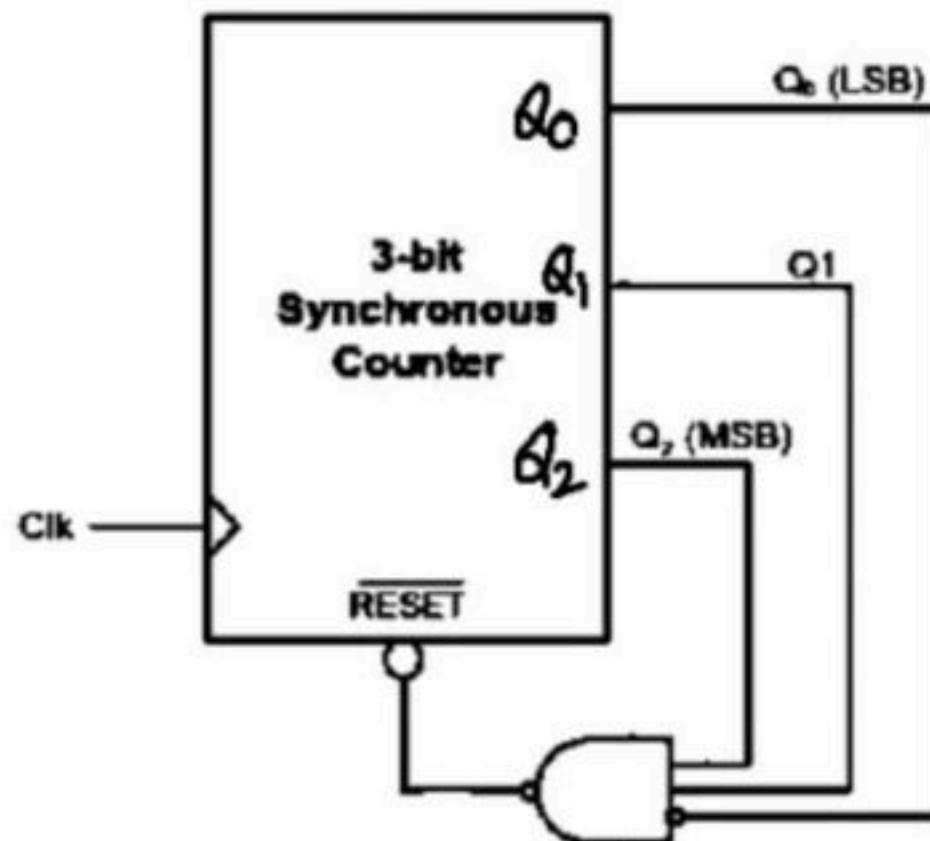
GATE (ECE-2016)

(a) mod-5 counter

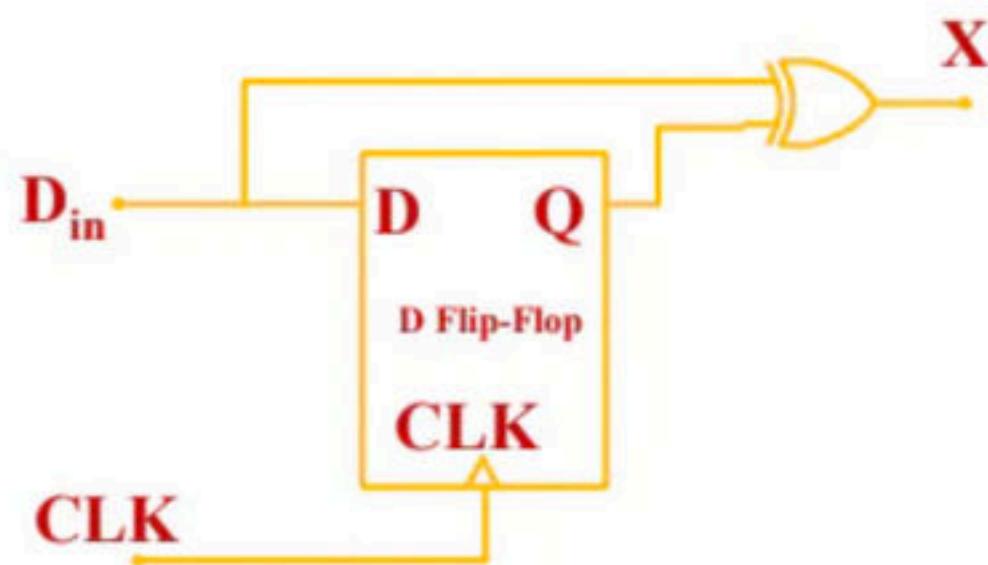
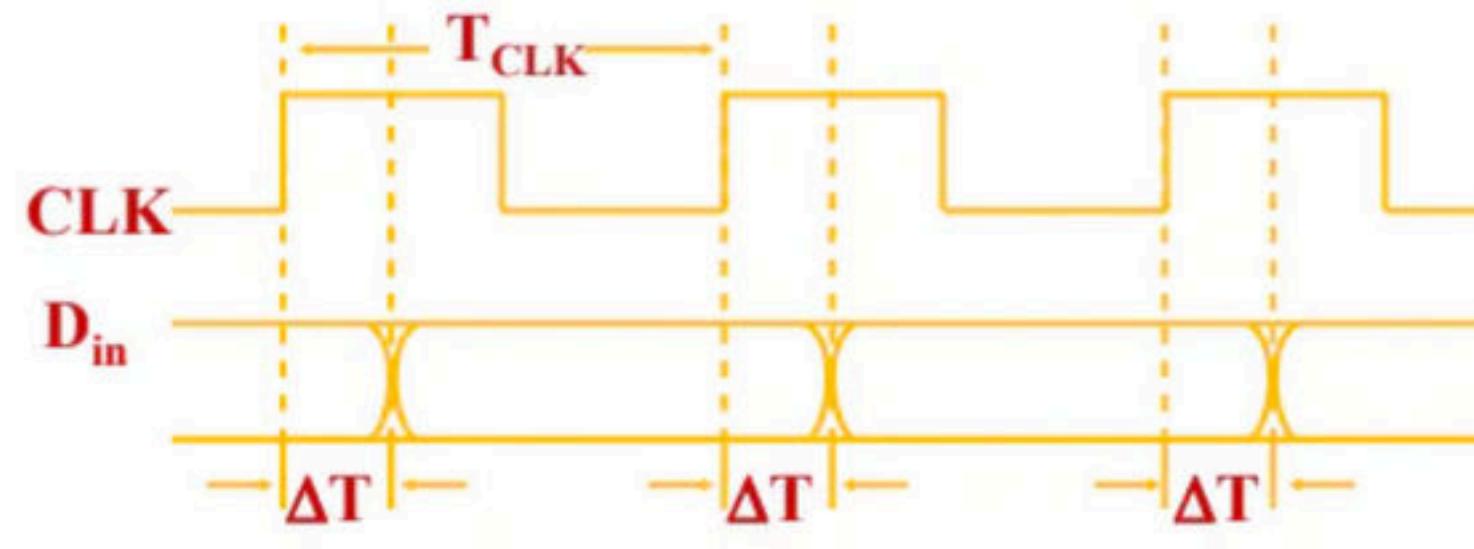
(b) mod-6 counter

(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 _____.



Q. Identify the circuit below

(a) Gray to binary converter

(b) Binary to excess 3 converter

(c) Binary to gray converter

(d) Excess-3 to binary converter

