AND GATE

1. The AND gate is a logic circuit that has two or more inputs and a single output.

2. If any of the inputs are 0s, the output is 0.

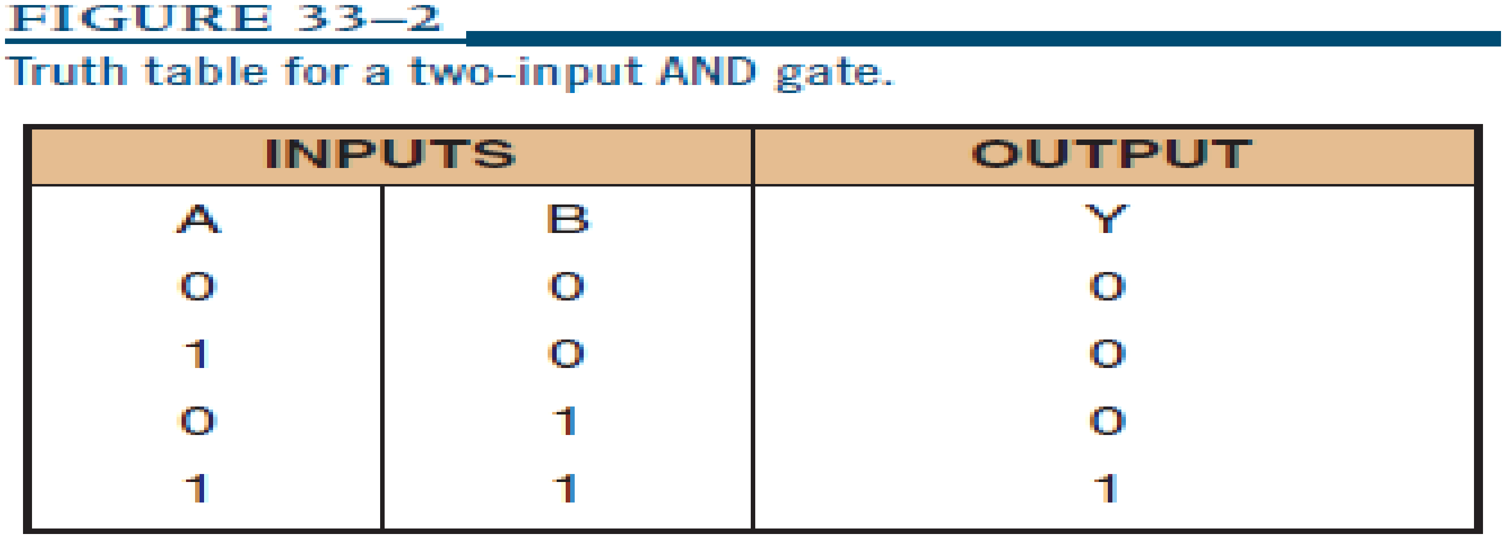


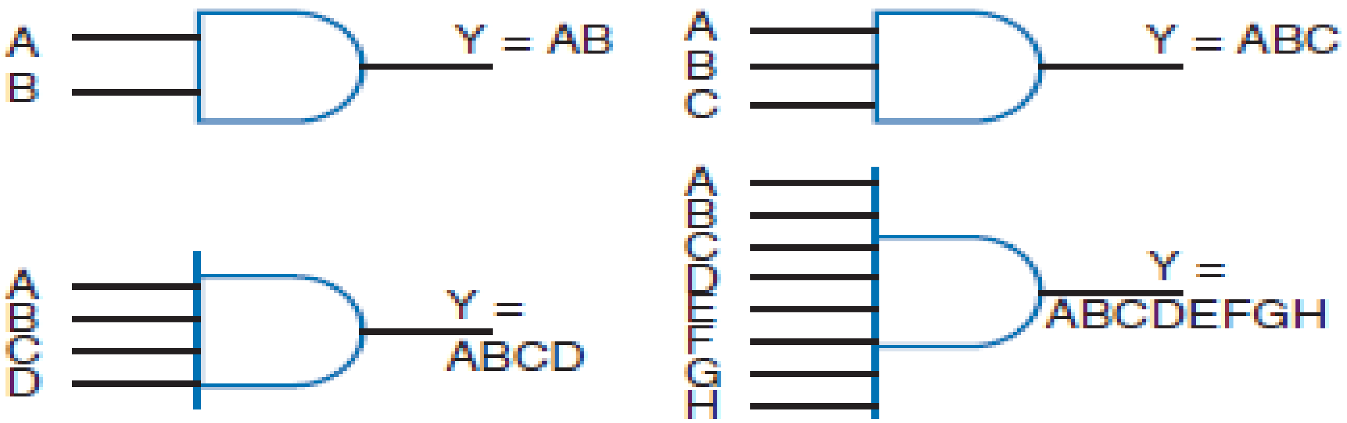
FIGURE 33–2

Truth table for a two-input AND gate.

INPUTS		OUTPUT
A	B	Y
0	0	0
1	0	0
0	1	0
1	1	1

FIGURE 33–1

Logic symbol for an AND gate.



$$Y = A \cdot B \text{ or } Y = AB.$$

OR GATE

1. An OR gate produces a 1 output if any of its inputs are 1s. The output is a 0 if all the inputs are 0s.

$$Y = A + B.$$

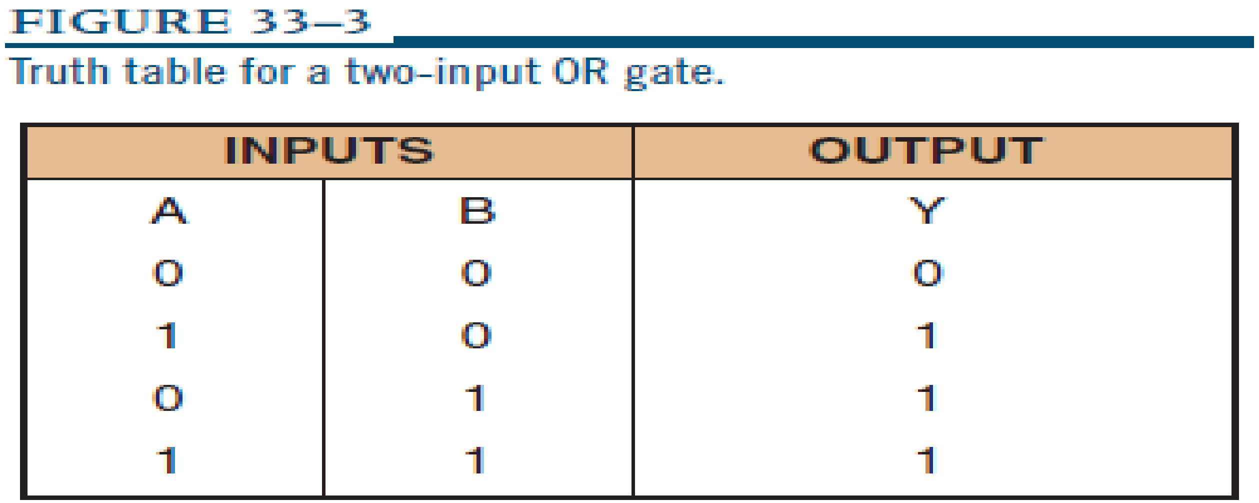


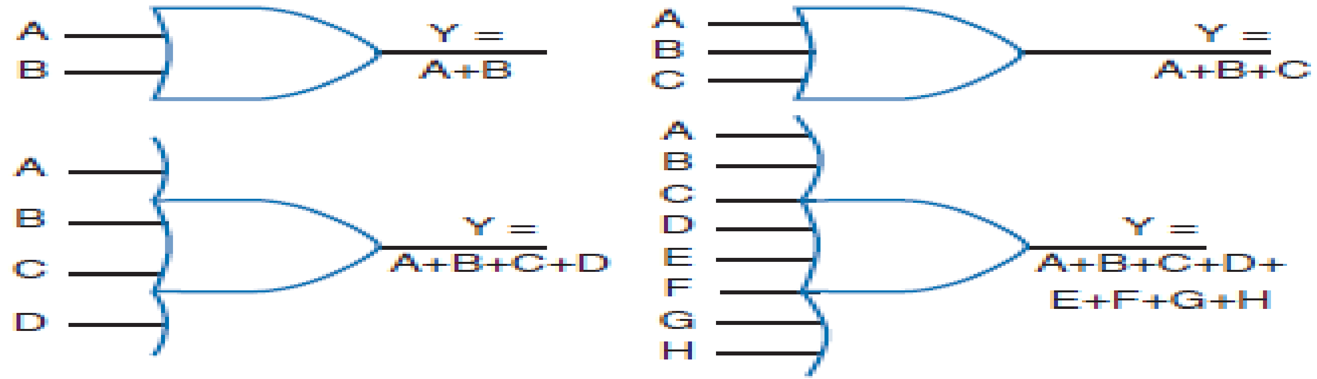
FIGURE 33–3

Truth table for a two-input OR gate.

INPUTS		OUTPUT
A	B	Y
0	0	0
1	0	1
0	1	1
1	1	1

FIGURE 33–4

Logic symbol for an OR gate.



NOT GATE

1. The simplest logic circuit is the NOT gate.
2. It performs the function called inversion, or complementation, and is commonly referred to as an Inverter.

FIGURE 33–6

Logic symbol for an inverter.

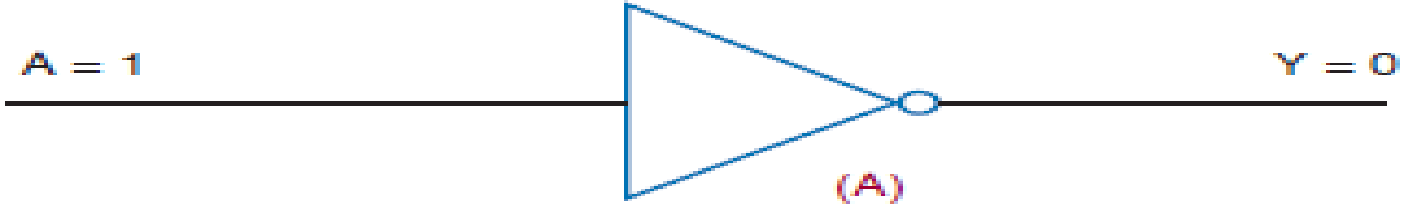
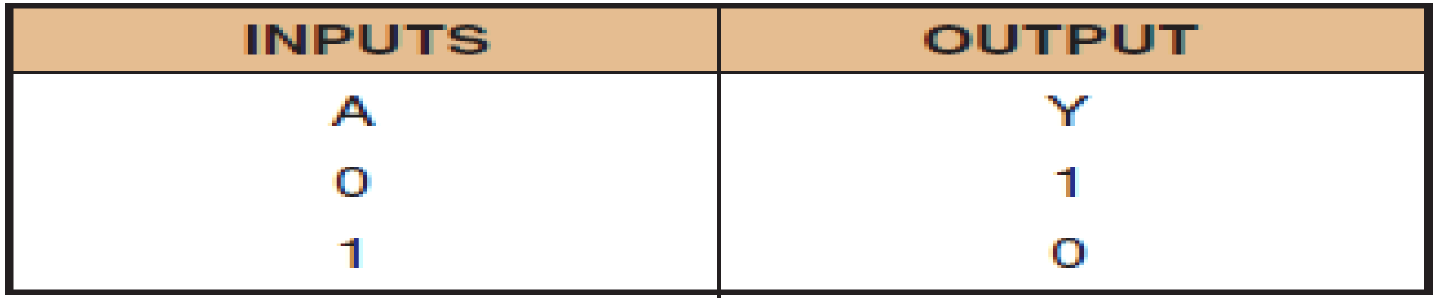


FIGURE 33–5

Truth table for an inverter.



INPUTS	OUTPUT
A	Y
0	1
1	0

- The input to an inverter is labeled A and the output is labeled $\overline{A}$ (read “A NOT” or “NOT A”).

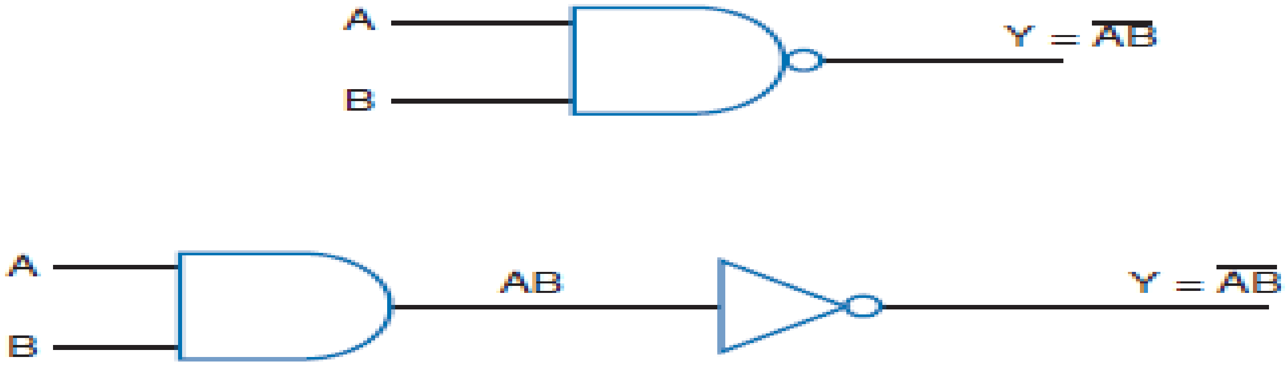
- The bar over the letter A indicates the complement of A.

NAND GATE

1. A NAND gate is a combination of an inverter and an AND gate.

FIGURE 33–7

Logic symbol for an NAND gate.



- The algebraic formula for NAND-gate output i$Y = \overline{AB}$,

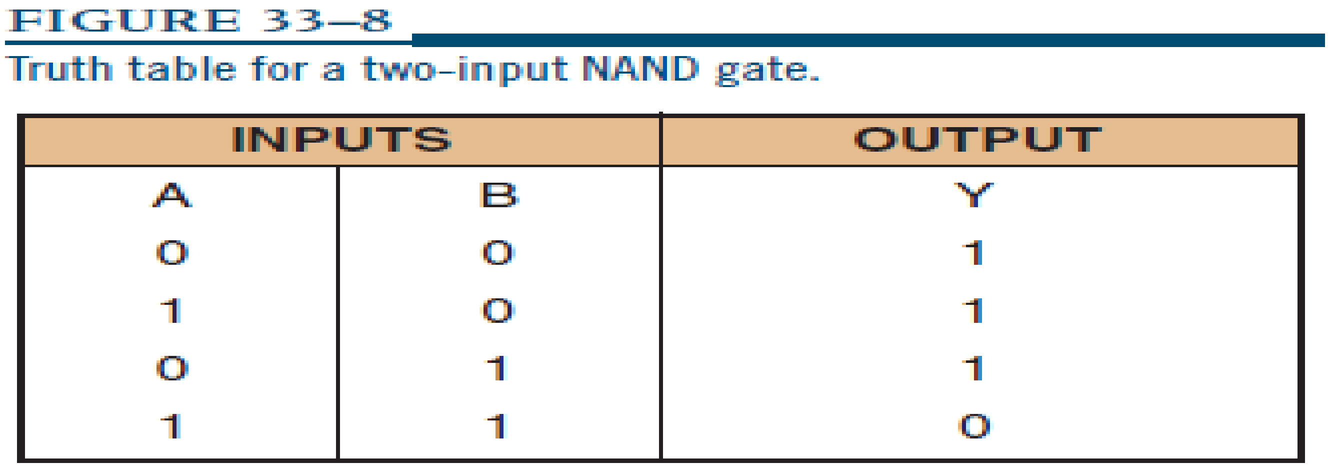


FIGURE 33–8

Truth table for a two-input NAND gate.

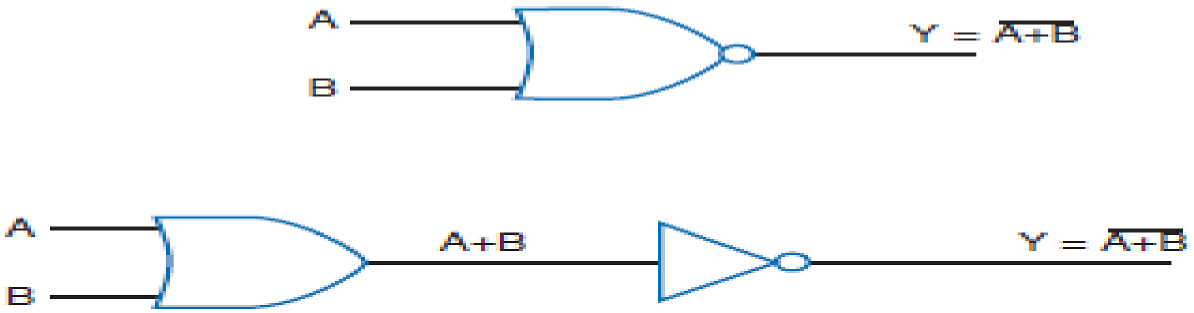
INPUTS		OUTPUT
A	B	Y
0	0	1
1	0	1
0	1	1
1	1	0

NOR GATE

1. A **NOR gate is a combination of an inverter and an** OR gate. Its name derives from its NOT-OR function.
2. Also shown is its equivalency to an OR gate and an inverter.

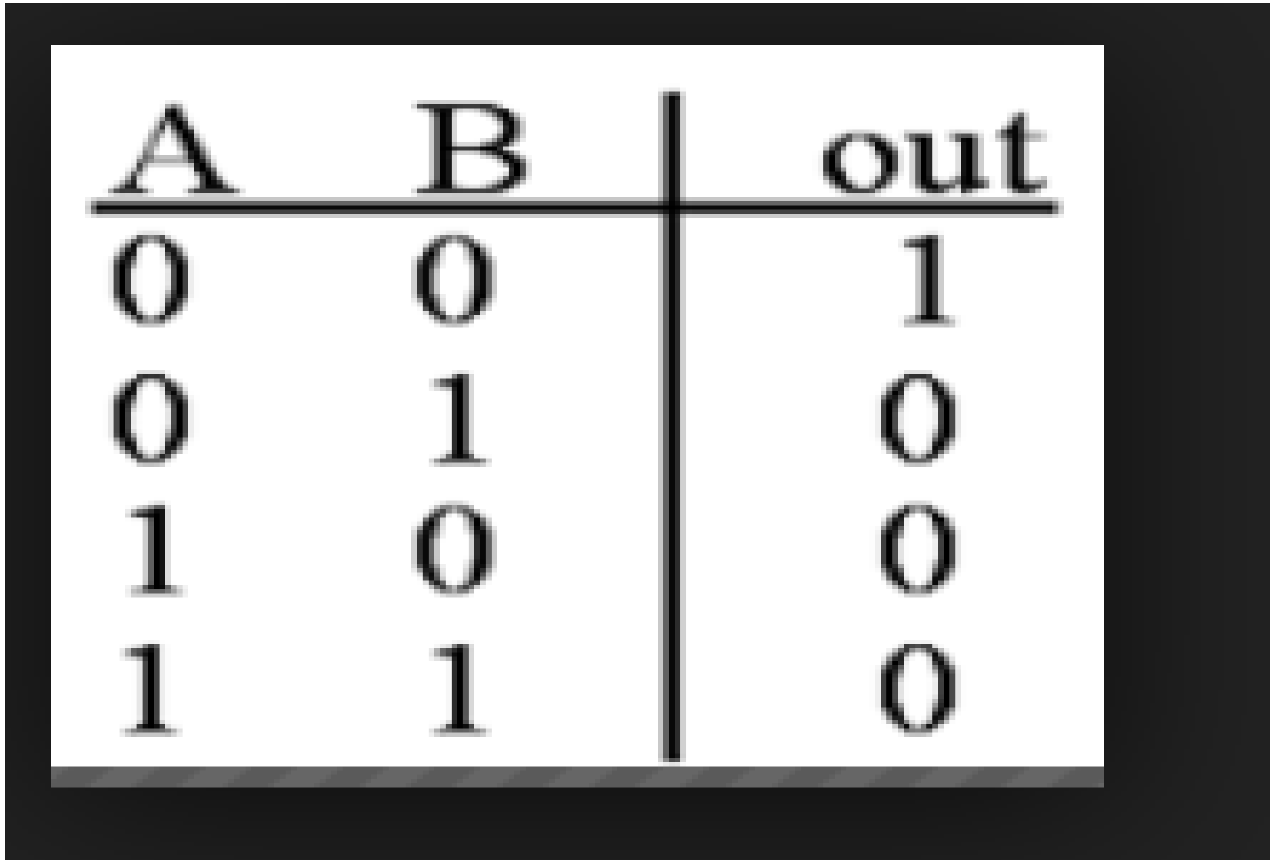
FIGURE 33–10

Logic symbol for a NOR gate.



- The algebraic expression for NOR-gate output is

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



A	B	out
0	0	1
0	1	0
1	0	0
1	1	0

EXCLUSIVE -OR

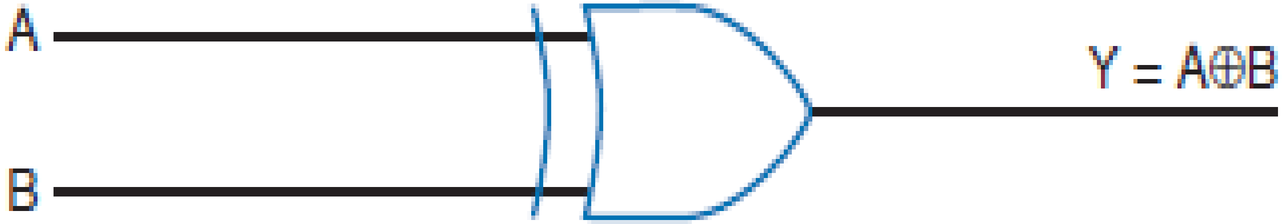
- The XOR gate is a digital logic gate that implements an exclusive or; that is, a true output (1/HIGH) results if one, and only one, of the inputs to the gate is true.

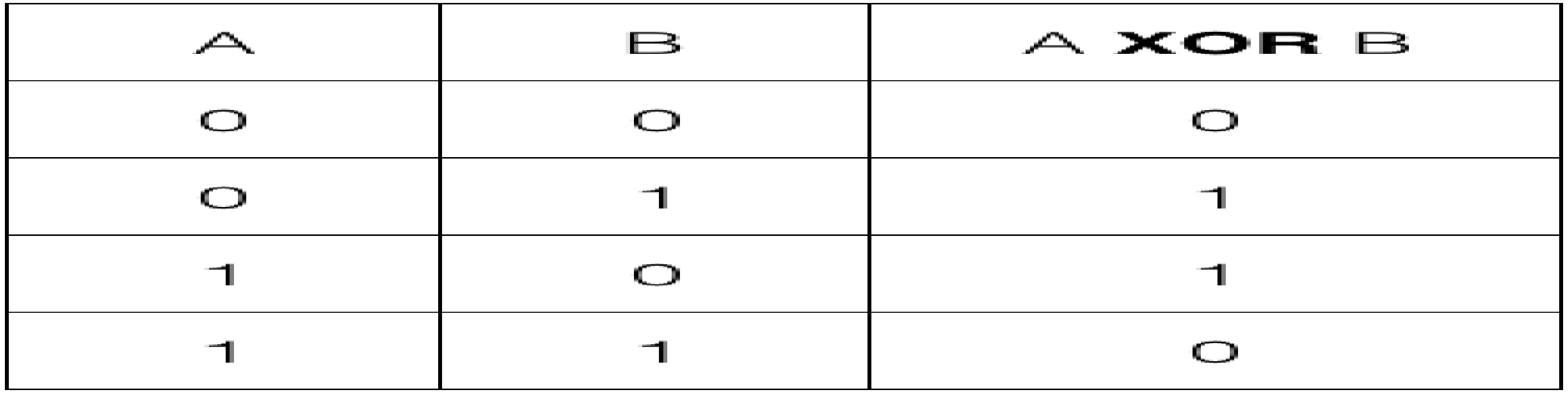
- The algebraic output is written as

$$Y = A \oplus B.$$

FIGURE 33–12

Logic symbol for an exclusive OR gate.



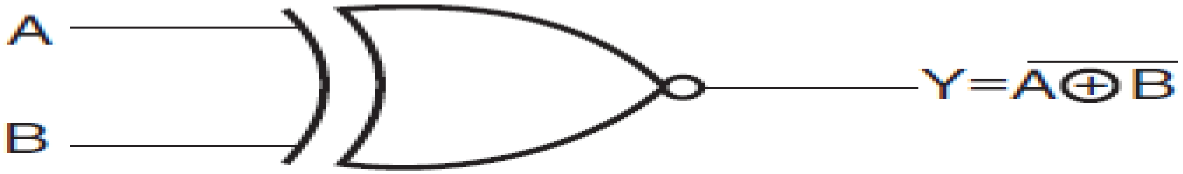


A	B	A **XOR** B
0	0	0
0	1	1
1	0	1
1	1	0

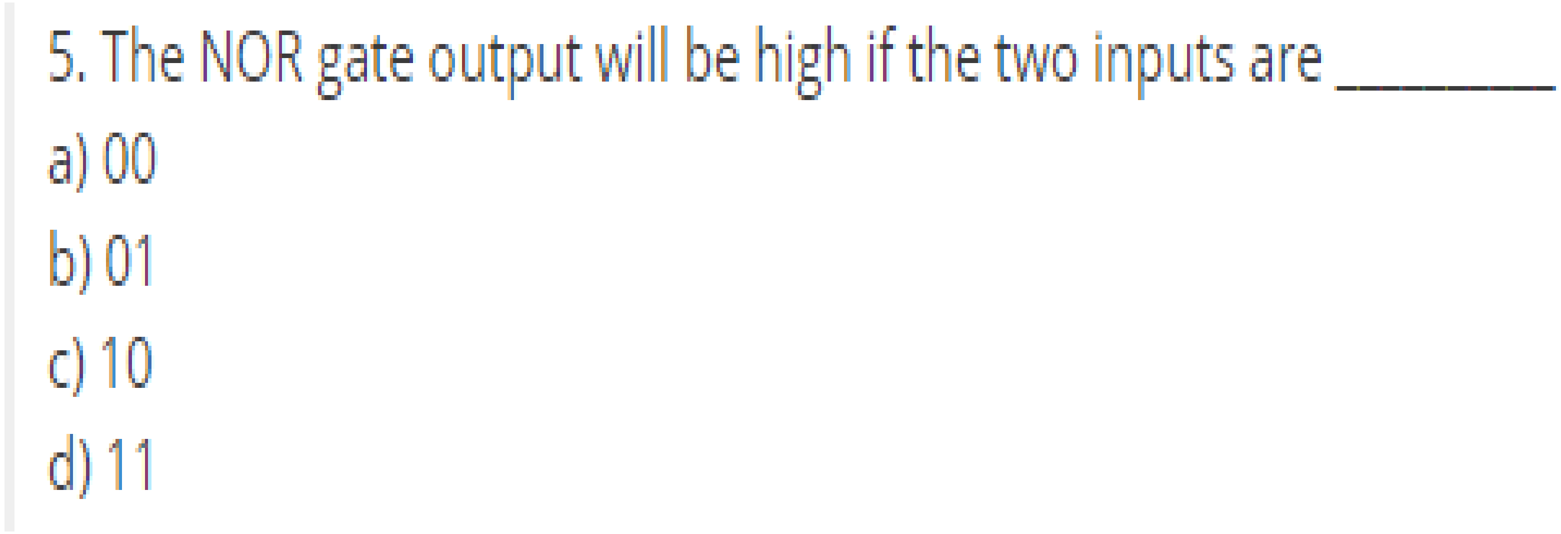
1. The complement of the XOR gate is the XNOR (**exclusive NOR) gate.**

2. **Its symbol is shown in** Figure 33–14.

3. The algebraic output is written as $Y = \overline{A \oplus B},$ read as “Y equals A exclusive nor B.”



POLL



5. The NOR gate output will be high if the two inputs are ________

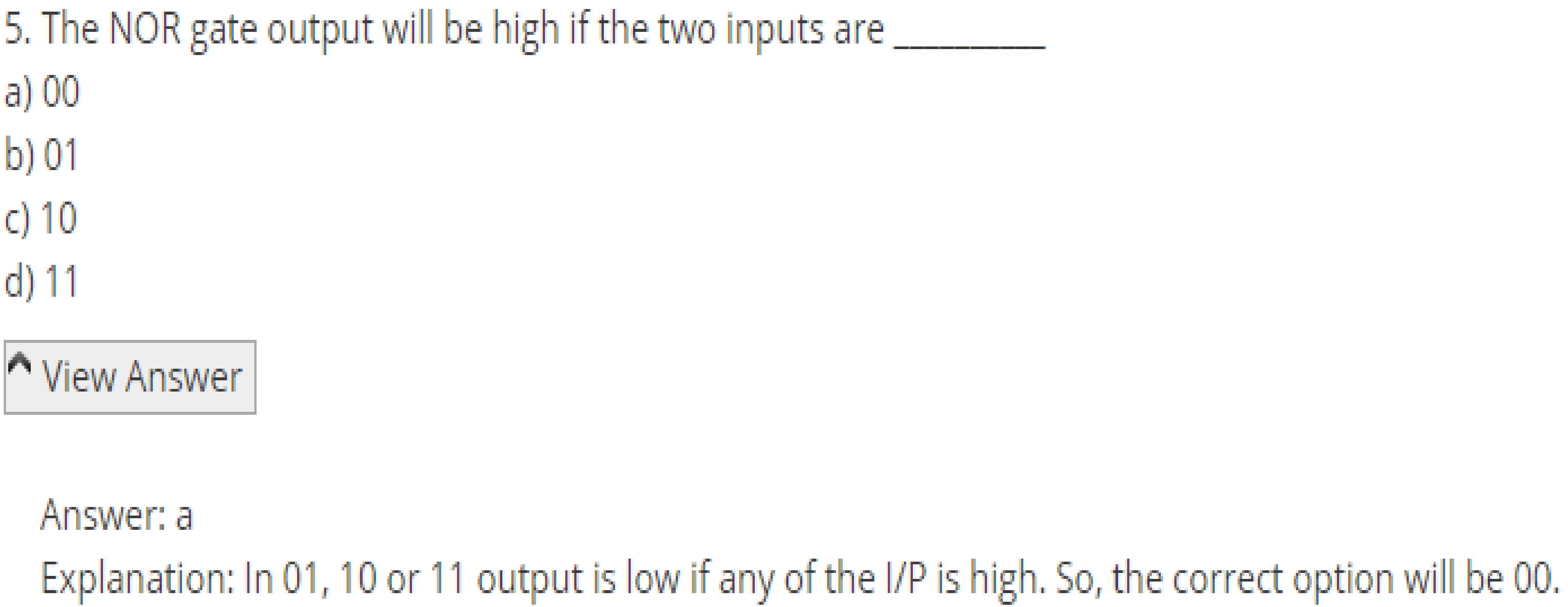
a) 00

b) 01

c) 10

d) 11

Solution



5. The NOR gate output will be high if the two inputs are ________

a) 00

b) 01

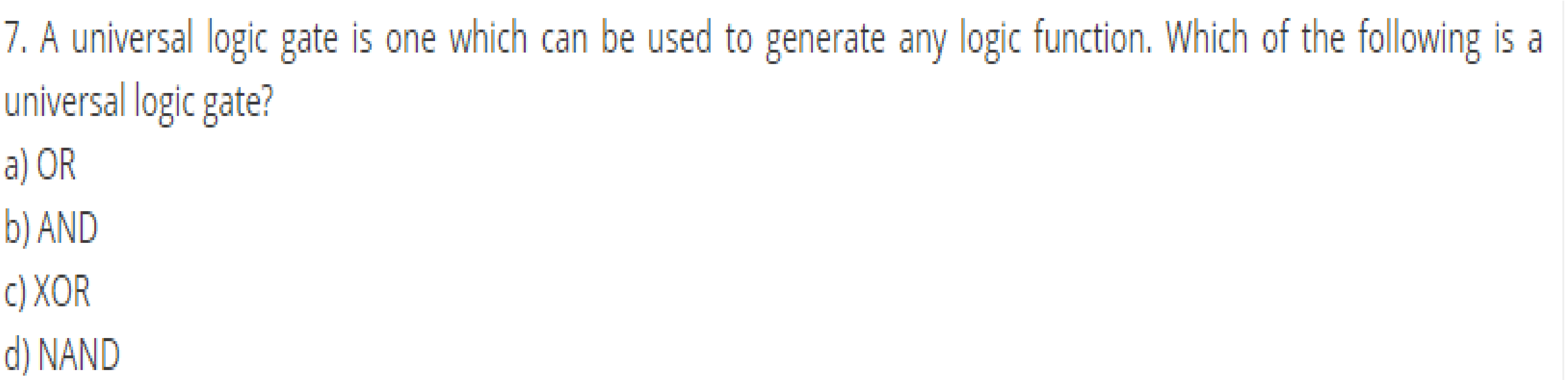
c) 10

d) 11

Answer: a

Explanation: In 01, 10 or 11 output is low if any of the I/P is high. So, the correct option will be 00.

Poll



7. A universal logic gate is one which can be used to generate any logic function. Which of the following is a universal logic gate?

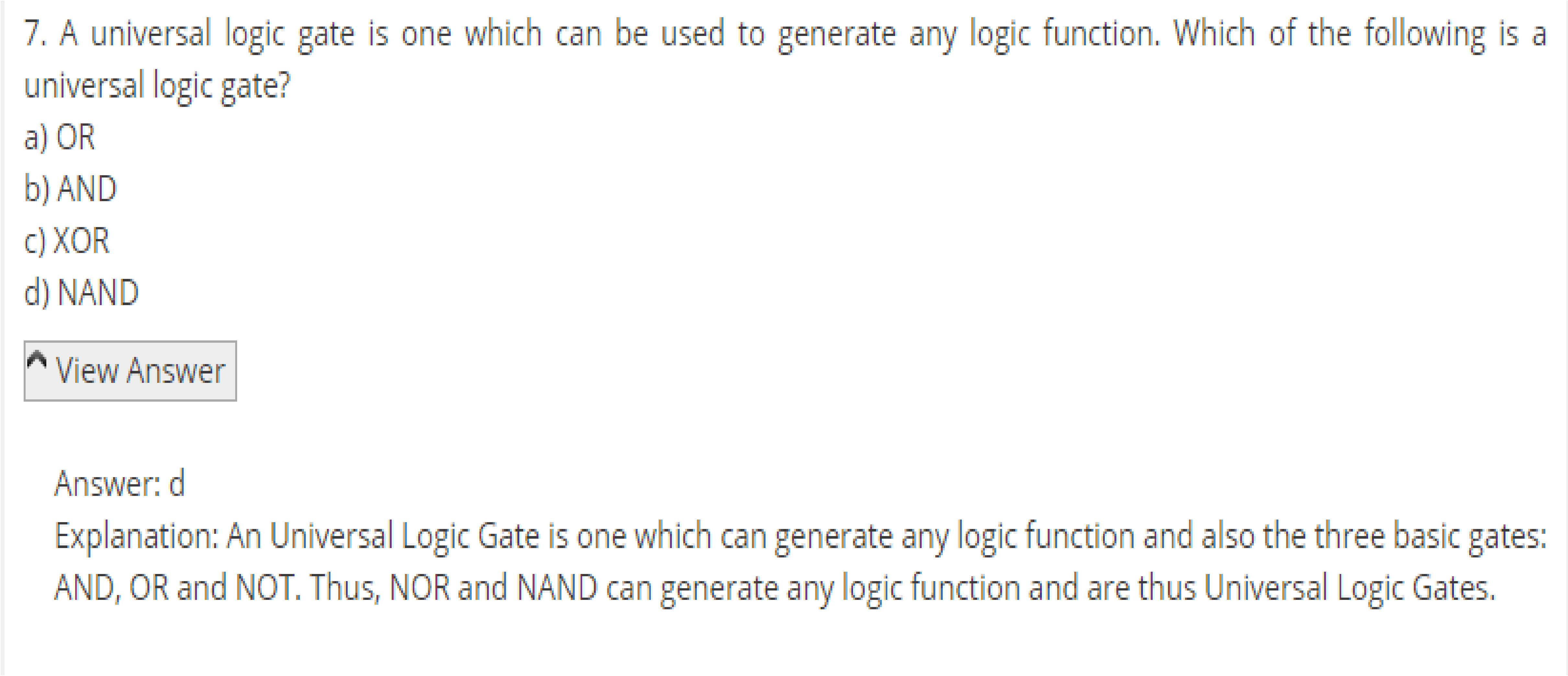
a) OR

b) AND

c) XOR

d) NAND

Solutions



7. A universal logic gate is one which can be used to generate any logic function. Which of the following is a universal logic gate?

a) OR

b) AND

c) XOR

d) NAND

Answer: d

Explanation: An Universal Logic Gate is one which can generate any logic function and also the three basic gates: AND, OR and NOT. Thus, NOR and NAND can generate any logic function and are thus Universal Logic Gates.

Law/Theorem	Law of Addition	Law of Multiplication
Identity Law	$x + 0 = x$	$x \cdot 1 = x$
Complement Law	$x + x' = 1$	$x \cdot x' = 0$
Idempotent Law	$x + x = x$	$x \cdot x = x$
Dominant Law	$x + 1 = 1$	$x \cdot 0 = 0$
Involution Law	$(x')' = x$	
Commutative Law	$x + y = y + x$	$x \cdot y = y \cdot x$
Associative Law	$x+(y+z) = (x+y)+z$	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$
Distributive Law	$x \cdot (y+z) = x \cdot y+x \cdot z$	$x+y\cdot z = (x+y) \cdot (x+z)$
Demorgan's Law	$(x+y)' = x' \cdot y'$	$(x \cdot y)' = x' + y'$
Absorption Law	$x + (x \cdot y) = x$	$x \cdot (x + y) = x$

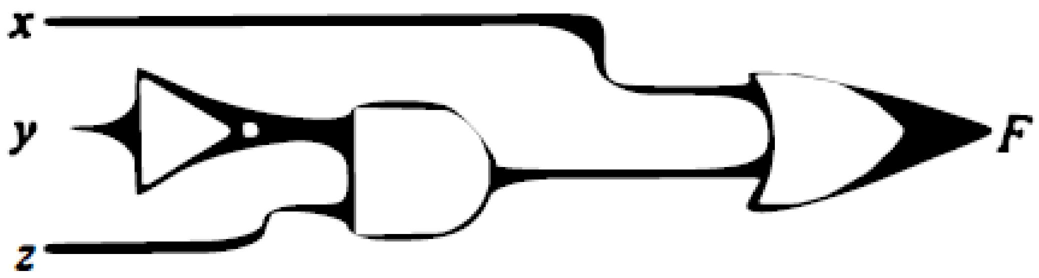
Boolean Algebra

- Boolean algebra is an algebra that deals with binary variables and logic operations.
- The variables are designated by letters such as A, B, x, and y. The three basic logic operations are AND, OR, and complement.

Figure 1-3 Truth table and logic diagram for $F = x + y'z$.

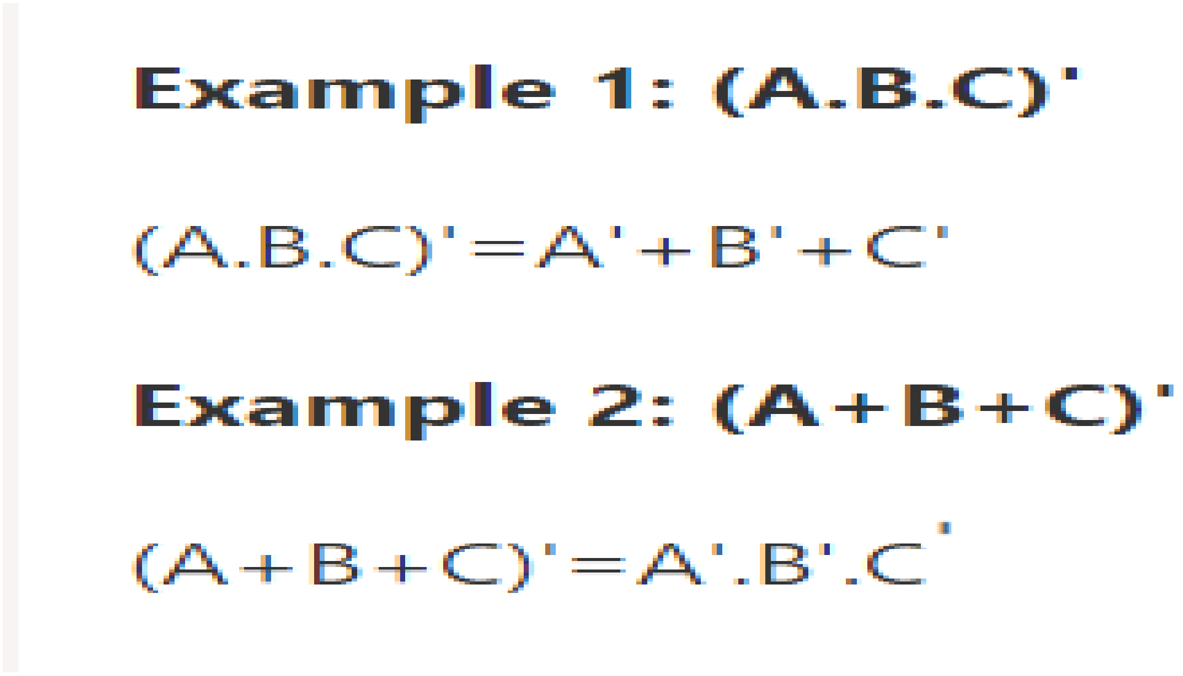
x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

(a) Truth table



(b) Logic diagram

Apply DeMorgan's theorems



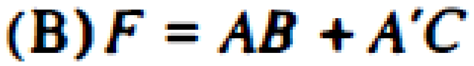
Example 1: $(A.B.C)'$

$(A.B.C)'=A'+B'+C'$

Example 2: $(A+B+C)'$

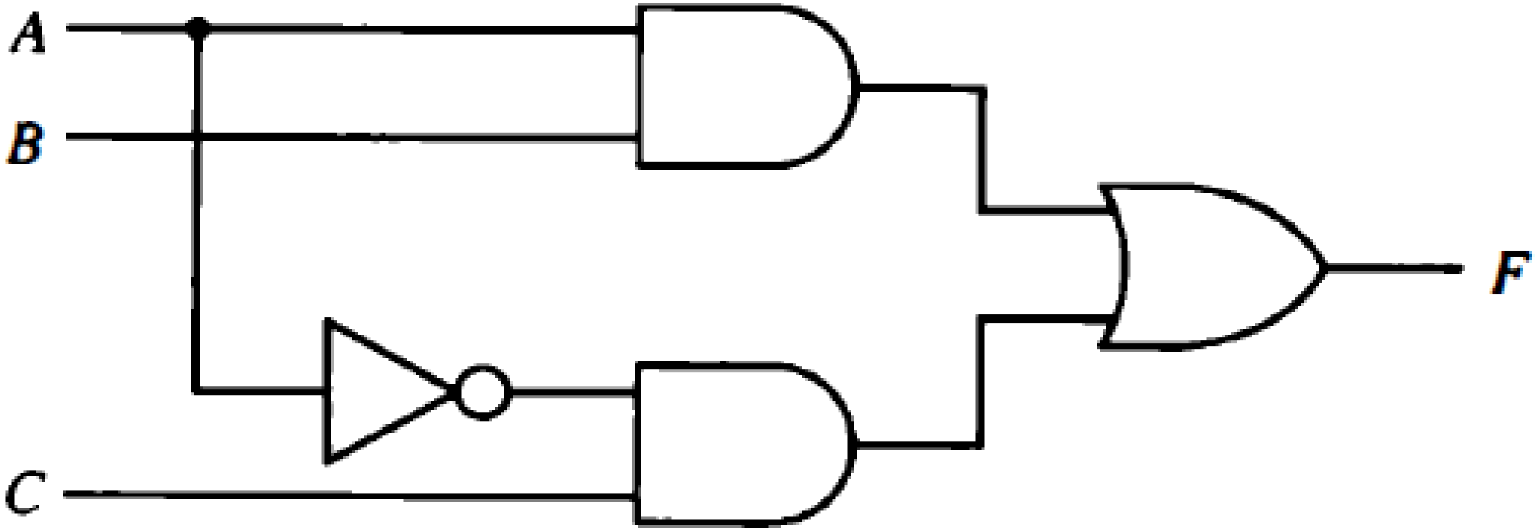
$(A+B+C)'=A'.B'.C'$

Q- Draw Diagram



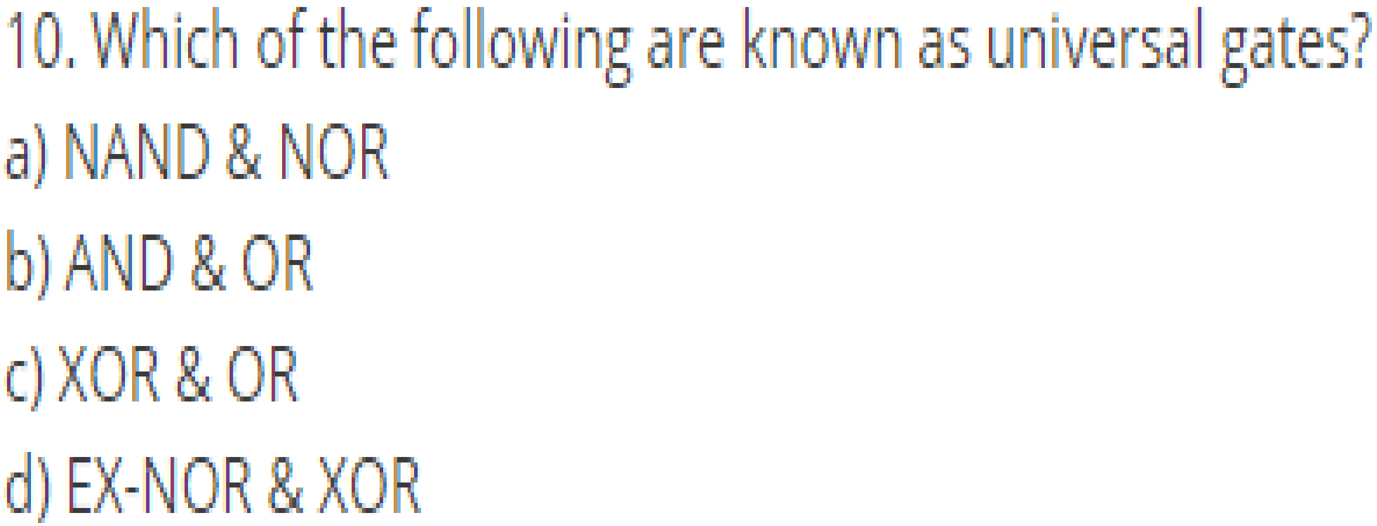
(B) $F = AB + A'C$

Q- draw diagram



(B) $F = AB + A'C$

Poll



10. Which of the following are known as universal gates?

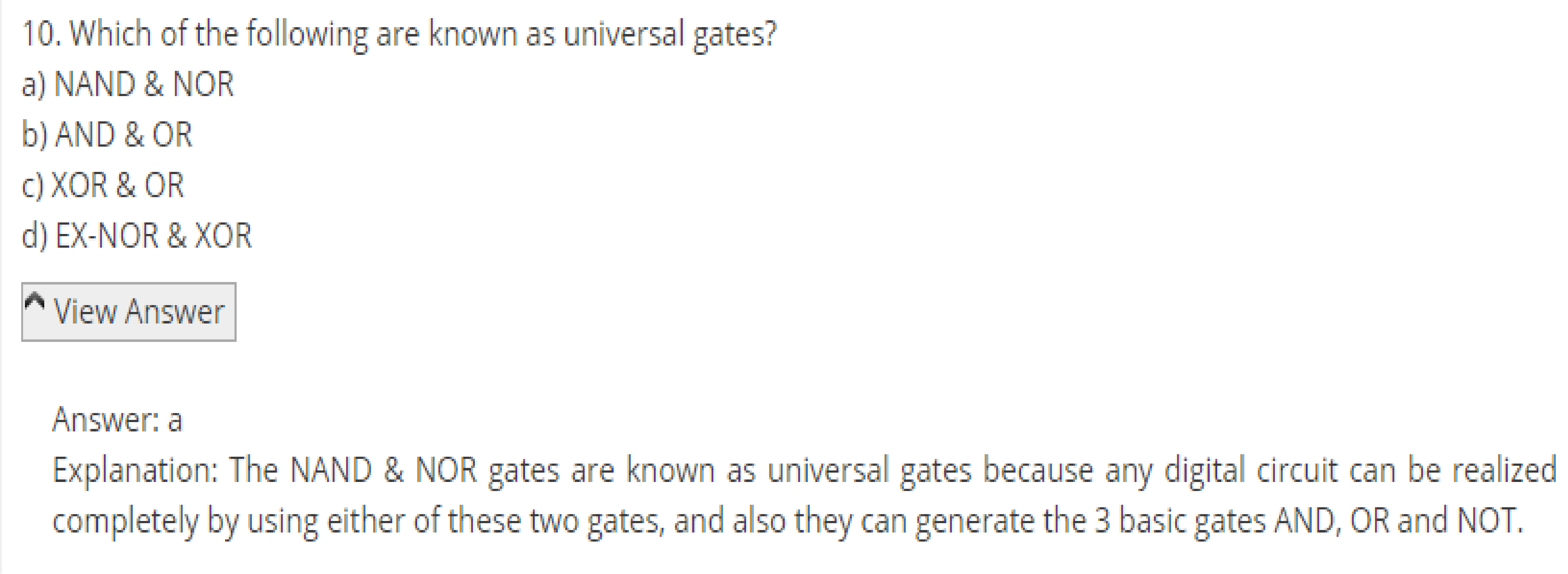
a) NAND & NOR

b) AND & OR

c) XOR & OR

d) EX-NOR & XOR

Solutions



10. Which of the following are known as universal gates?

a) NAND & NOR

b) AND & OR

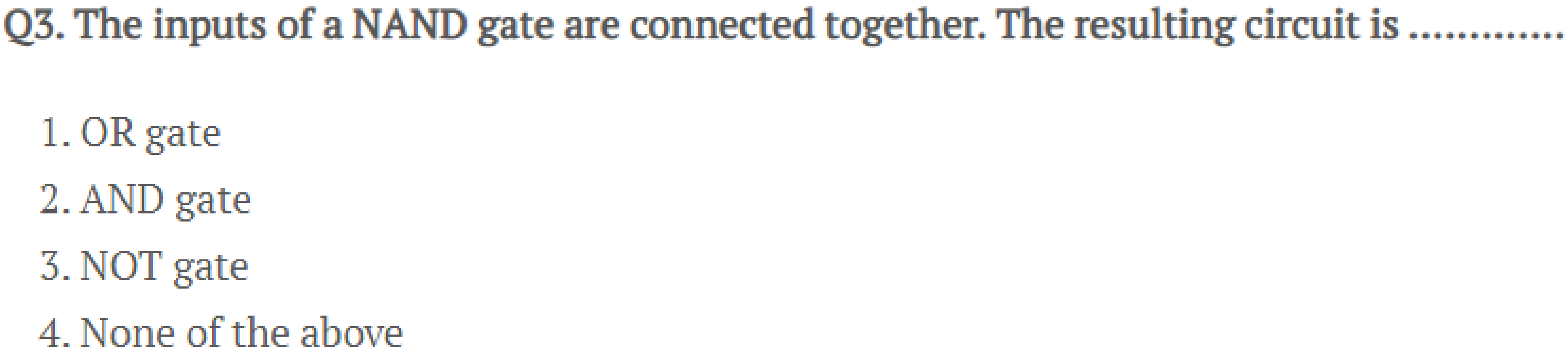
c) XOR & OR

d) EX-NOR & XOR

Answer: a

Explanation: The NAND & NOR gates are known as universal gates because any digital circuit can be realized completely by using either of these two gates, and also they can generate the 3 basic gates AND, OR and NOT.

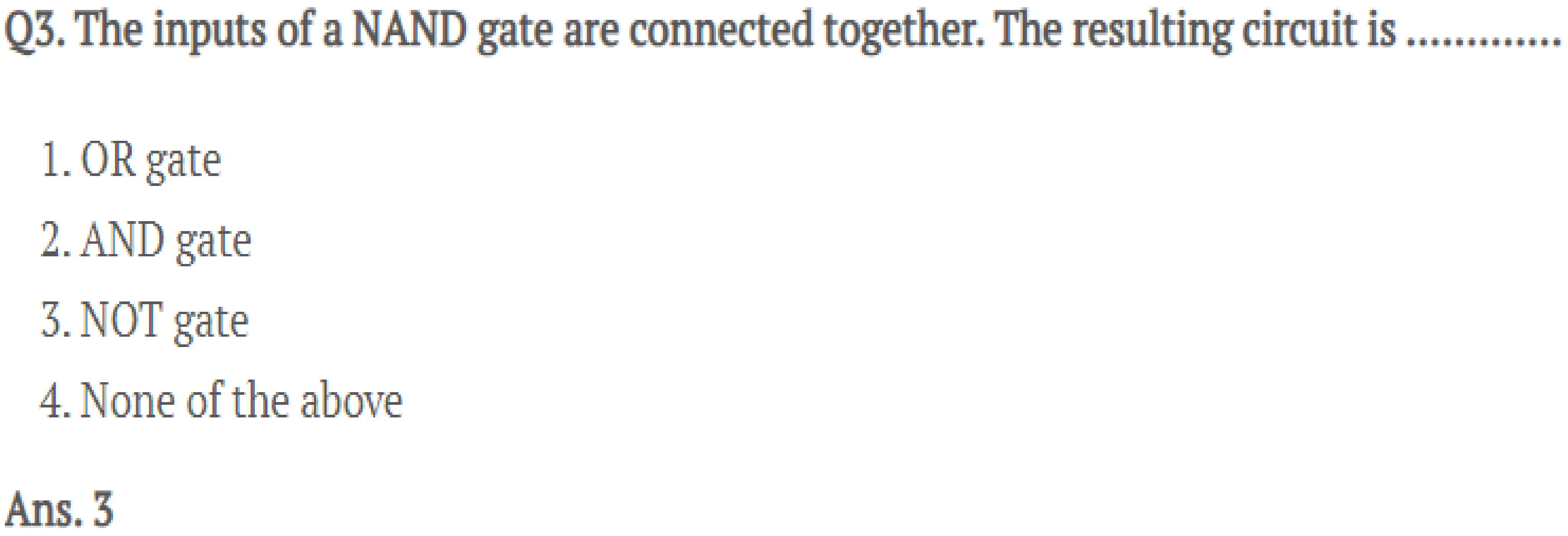
Poll



Q3. The inputs of a NAND gate are connected together. The resulting circuit is ....

1. OR gate
2. AND gate
3. NOT gate
4. None of the above

Solu

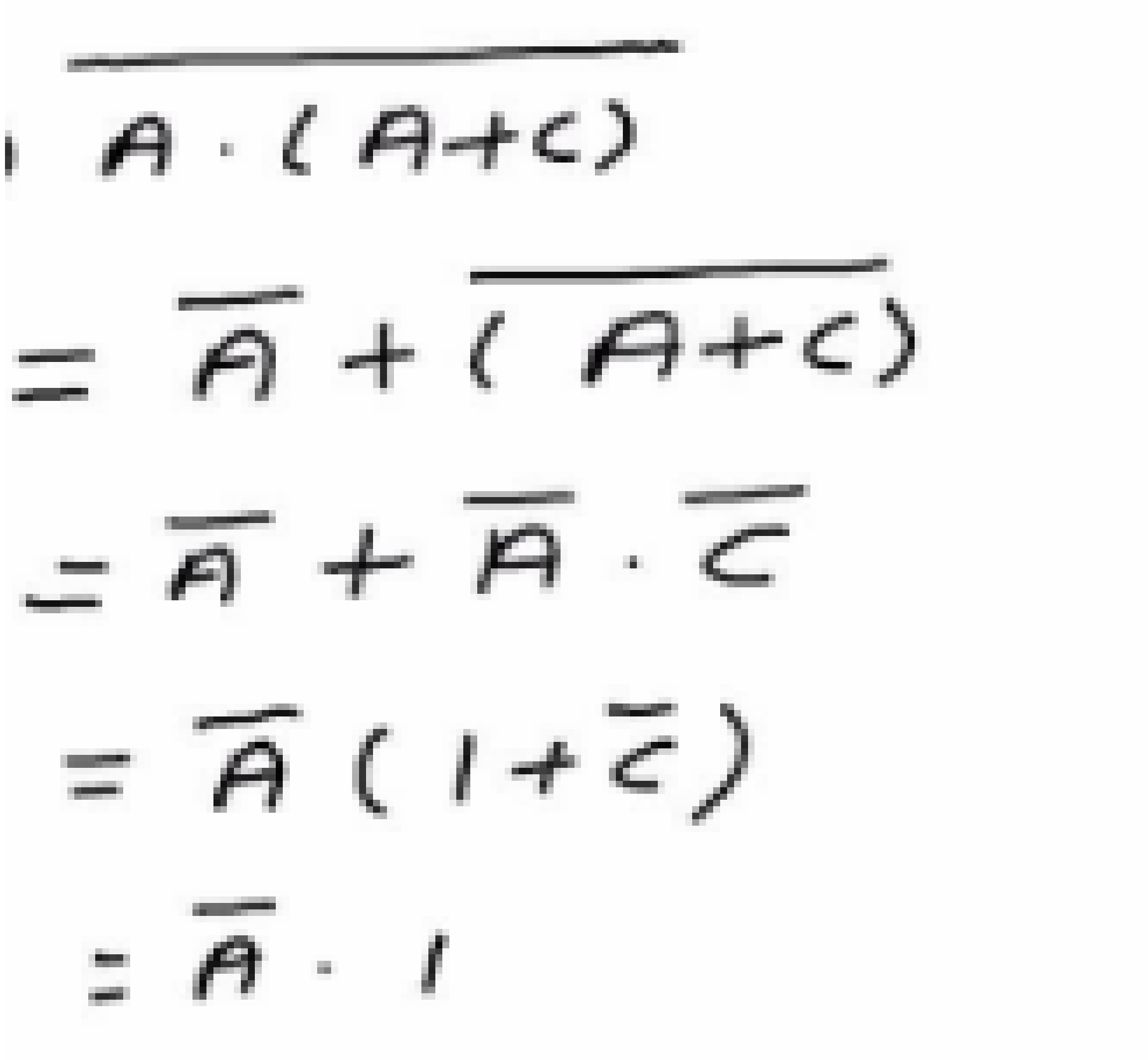


Q3. The inputs of a NAND gate are connected together. The resulting circuit is ....

1. OR gate
2. AND gate
3. NOT gate
4. None of the above

Ans. 3

Discussions



$$\overline{A\cdot(A+C)}$$

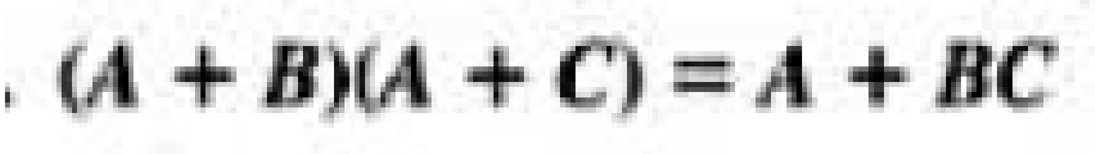
$$=\overline{A}+\overline{(A+C)}$$

$$=\overline{A}+\overline{A}\cdot\overline{C}$$

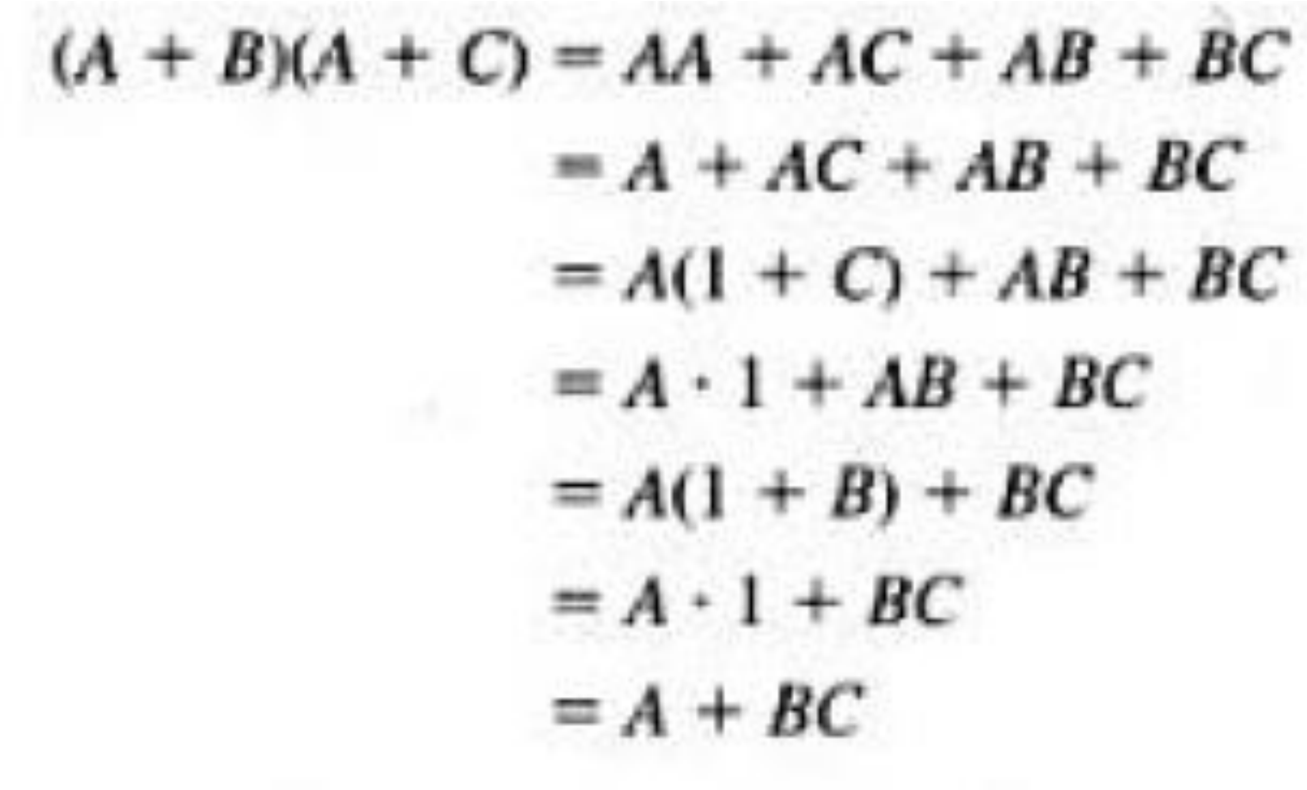
$$=\overline{A}(1+\overline{C})$$

$$=\overline{A}\cdot 1$$

Questions

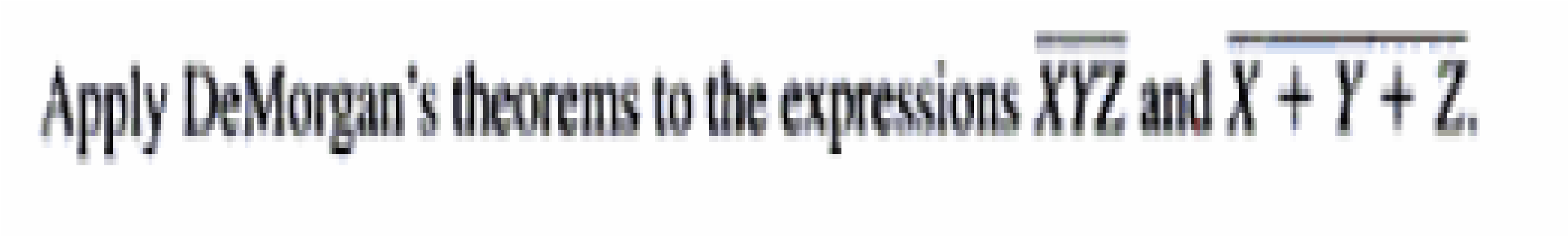


. $(A + B)(A + C) = A + BC$



$$
\begin{aligned}
(A + B)(A + C) &= AA + AC + AB + BC \\
&= A + AC + AB + BC \\
&= A(1 + C) + AB + BC \\
&= A \cdot 1 + AB + BC \\
&= A(1 + B) + BC \\
&= A \cdot 1 + BC \\
&= A + BC
\end{aligned}
$$

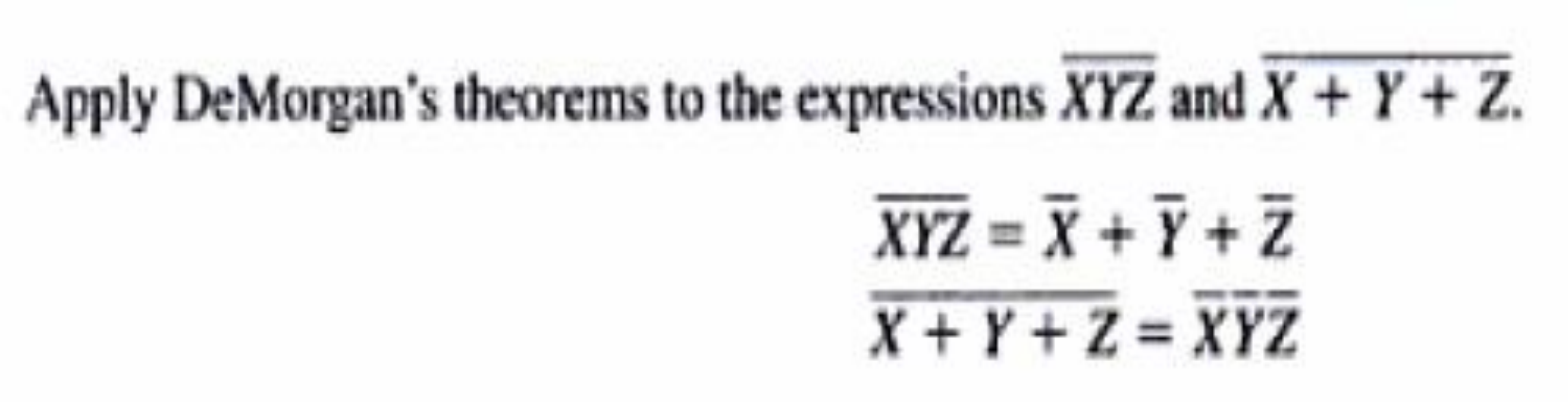
Questions



Apply DeMorgan's theorems to the expressions $\overline{XYZ}$ and $\overline{X + Y + Z}$.

Solve it

Solutions

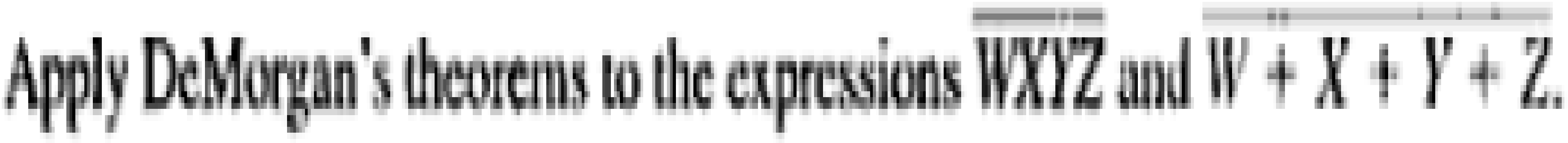


Apply DeMorgan's theorems to the expressions $\overline{XYZ}$ and $\overline{X + Y + Z}$.

$$\overline{XYZ} = \overline{X} + \overline{Y} + \overline{Z}$$

$$\overline{X + Y + Z} = \overline{X}\,\overline{Y}\,\overline{Z}$$

Questions



Apply DeMorgan's theorems to the expressions $\overline{WXYZ}$ and $\overline{W + X + Y + Z}$.

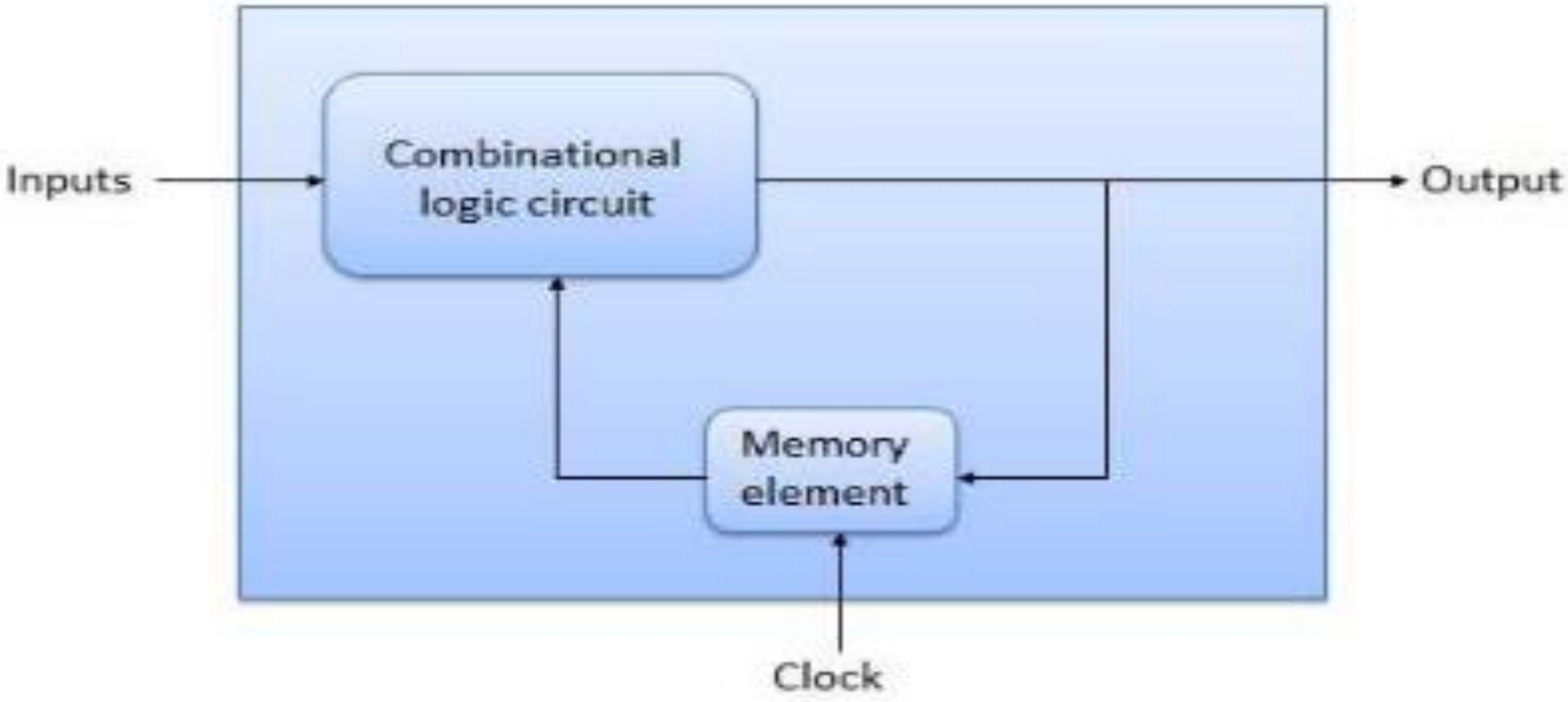
Solutions

$$\overline{WXYZ} = \overline{W} + \overline{X} + \overline{Y} + \overline{Z}$$

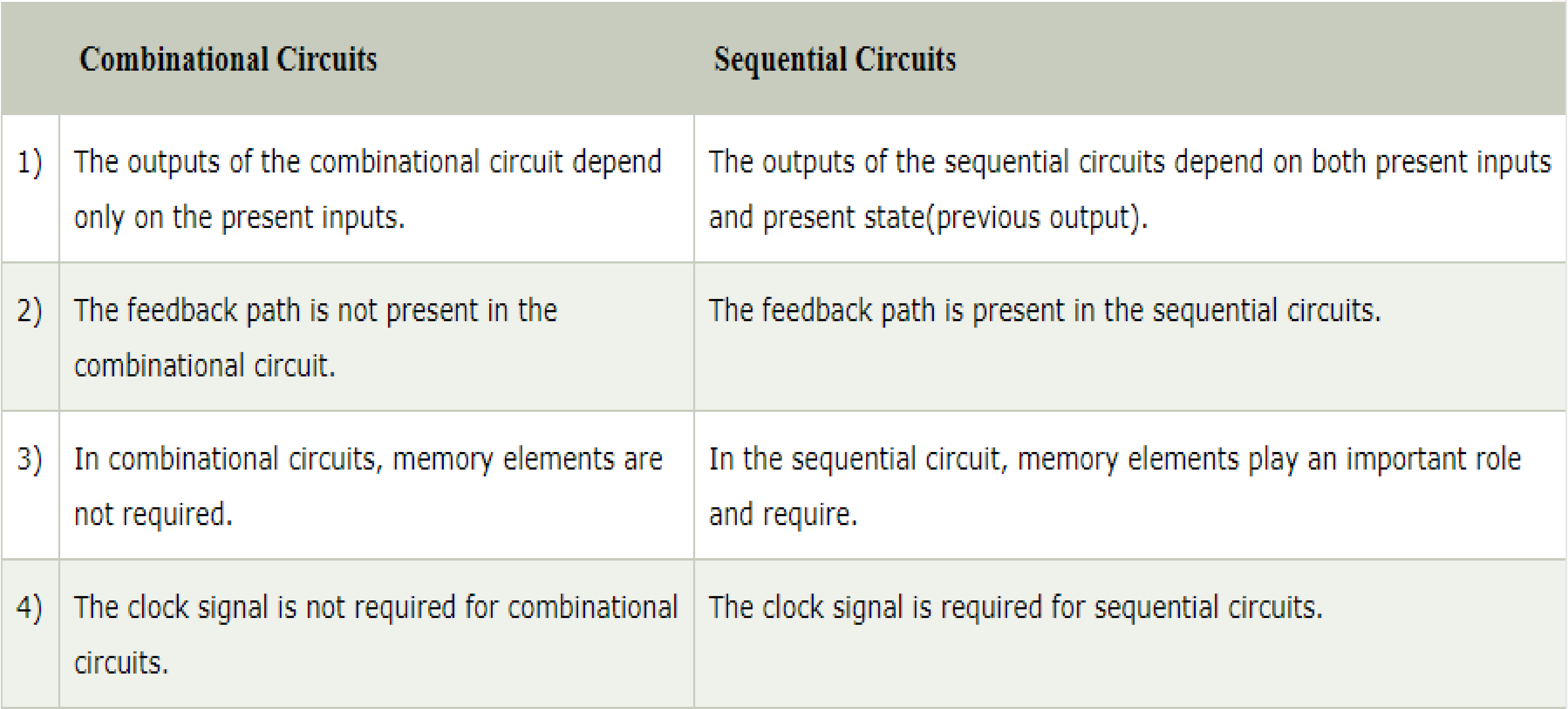
$$\overline{W + X + Y + Z} = \overline{W}\,\overline{X}\,\overline{Y}\,\overline{Z}$$

The combinational circuit does not use any memory. Hence the previous state of input does not have any effect on the present state of the circuit. But sequential circuit has memory so output can vary based on input. This type of circuits uses previous input, output, clock and a memory element.

Block diagram



Difference between the combinational circuits and sequential circuits are given below

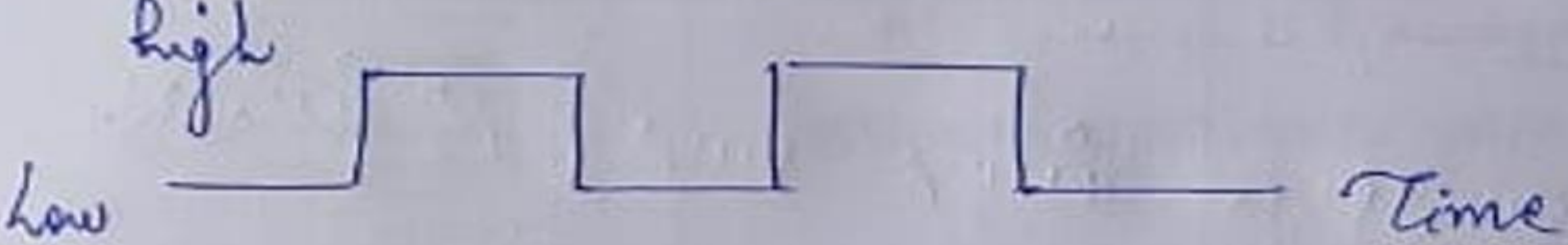


	Combinational Circuits	Sequential Circuits
1)	The outputs of the combinational circuit depend only on the present inputs.	The outputs of the sequential circuits depend on both present inputs and present state(previous output).
2)	The feedback path is not present in the combinational circuit.	The feedback path is present in the sequential circuits.
3)	In combinational circuits, memory elements are not required.	In the sequential circuit, memory elements play an important role and require.
4)	The clock signal is not required for combinational circuits.	The clock signal is required for sequential circuits.

Clock Signal

1) The clock signal is a timing signal. Every sequential circuit will have this timing signal applied.

2) clock is a rectangular signal & it repeats itsely after every T seconds.



#Triggering

Types of triggering

- Level triggering
- Edge triggering
  - a) positive edge triggering
  - b) Negative edge triggering

Types of Triggering

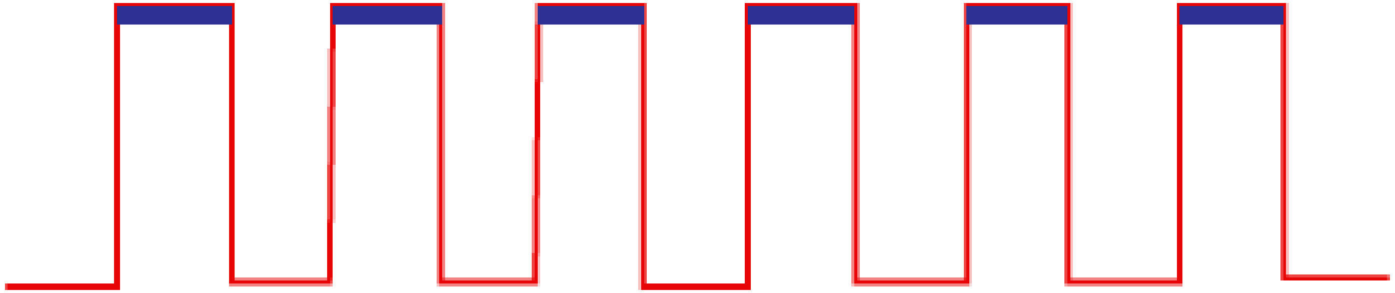
These are two types of triggering in sequential circuits:

Level triggering

The logic High and logic Low are the two levels in the clock signal. In level triggering, when the clock pulse is at a particular level, only then the circuit is activated. There are the following types of level triggering:

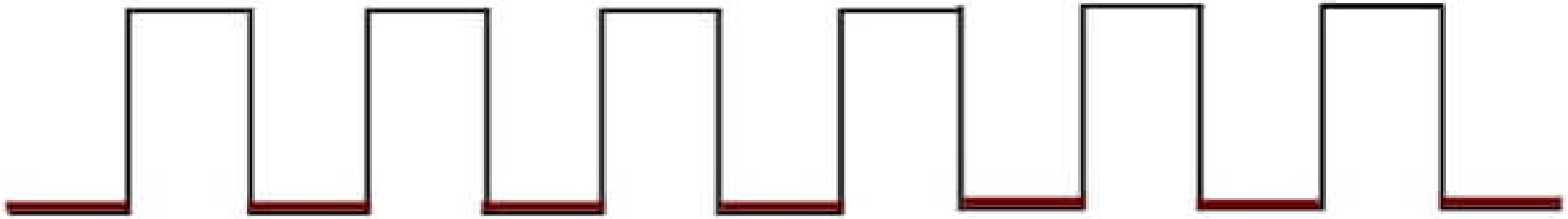
Positive level triggering

In a positive level triggering, the signal with Logic High occurs. So, in this triggering, the circuit is operated with such type of clock signal. Below is the diagram of positive level triggering:



Negative level triggering

In negative level triggering, the signal with Logic Low occurs. So, in this triggering, the circuit is operated with such type of clock signal. Below is the diagram of Negative level triggering:



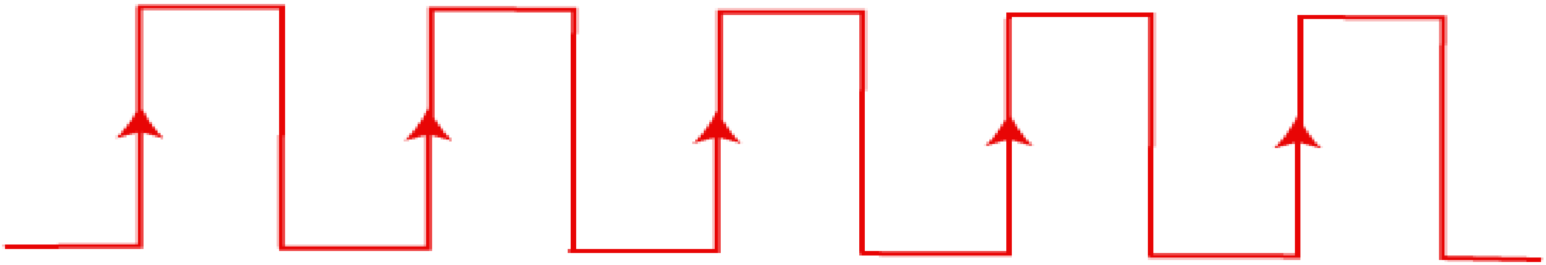
Edge triggering

In clock signal of edge triggering, two types of transitions occur, i.e., transition either from Logic Low to Logic High or Logic High to Logic Low.

Based on the transitions of the clock signal, there are the following types of edge triggering:

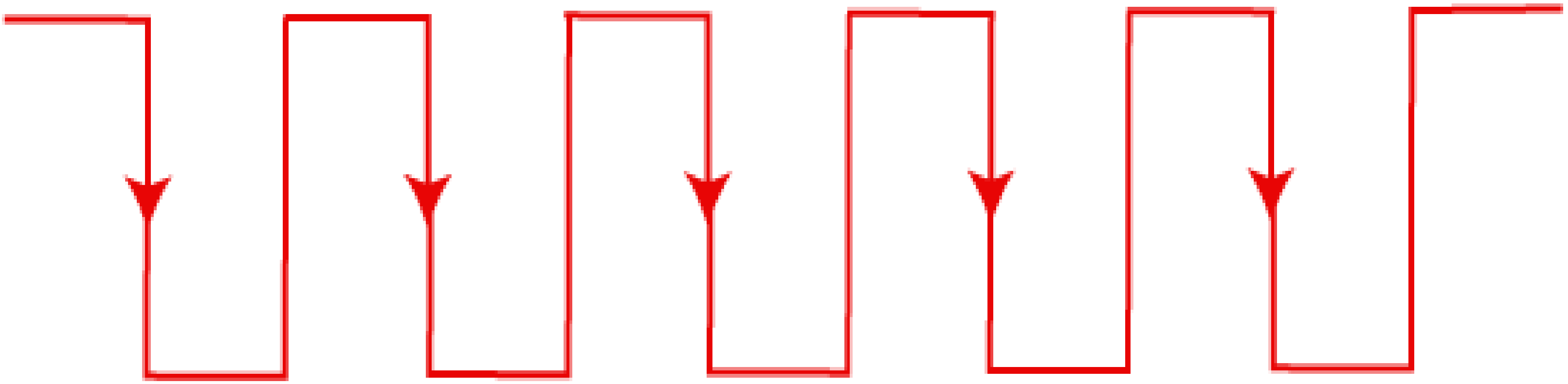
Positive edge triggering

The transition from Logic Low to Logic High occurs in the clock signal of positive edge triggering. So, in positive edge triggering, the circuit is operated with such type of clock signal. The diagram of positive edge triggering is given below.

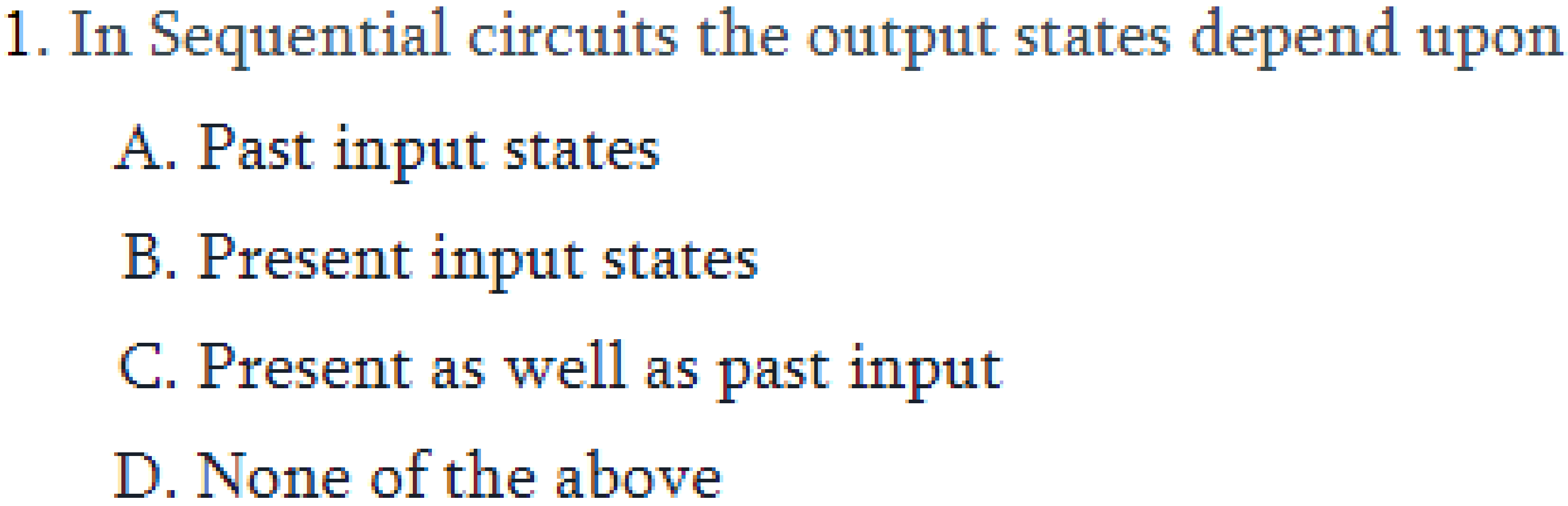


Negative edge triggering

The transition from Logic High to Logic low occurs in the clock signal of negative edge triggering. So, in negative edge triggering, the circuit is operated with such type of clock signal. The diagram of negative edge triggering is given below.



Poll



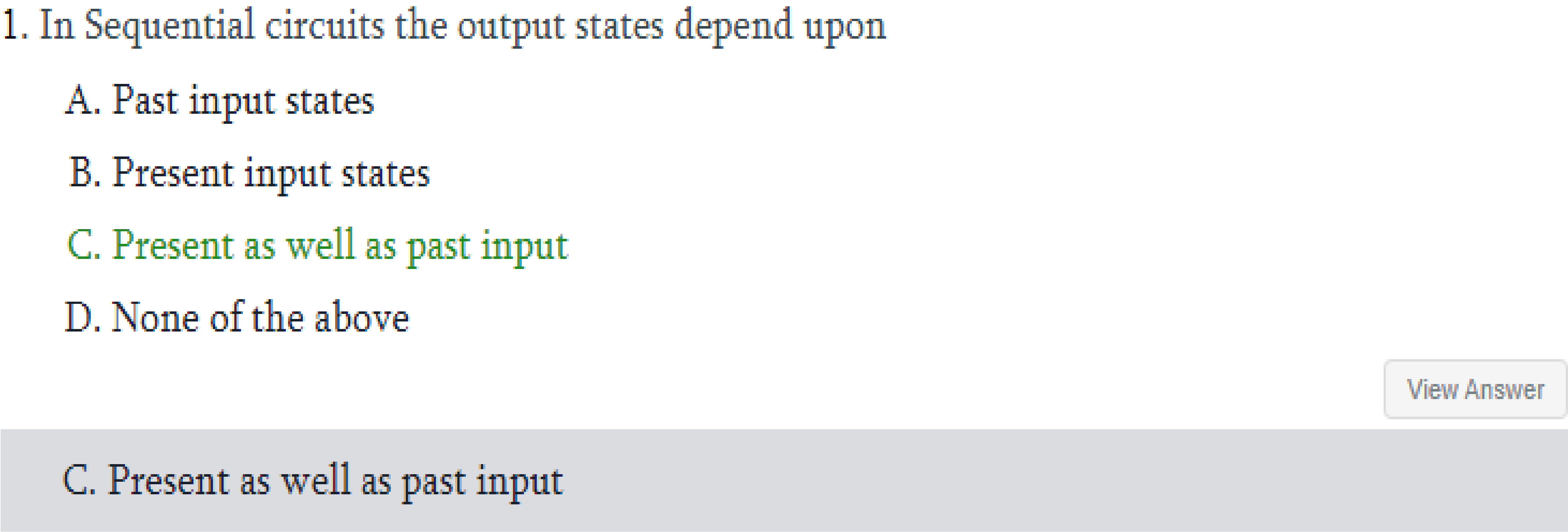
1. In Sequential circuits the output states depend upon

    A. Past input states

    B. Present input states

    C. Present as well as past input

    D. None of the above



1. In Sequential circuits the output states depend upon

   A. Past input states

   B. Present input states

   C. Present as well as past input

   D. None of the above

C. Present as well as past input

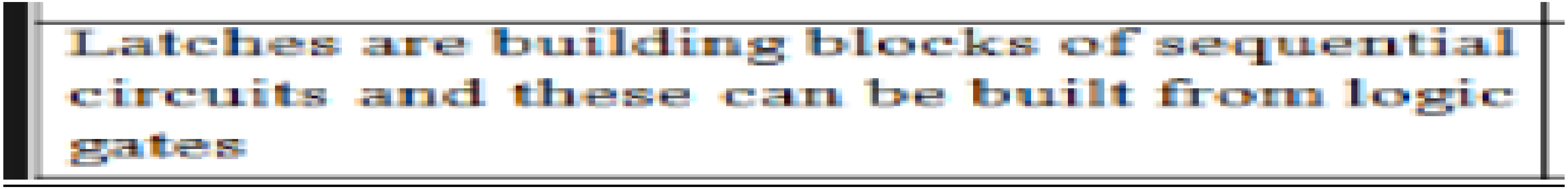
Latch

1) Latch is a sequential logic ckt which takes all it's input continuously & will change its o/p as soon as the input changes without waiting for clock i/p.

2) It is capable of storing (latching) the information:

3) [It has two output Q & $\bar{Q}$ which are complement of each other]

NOR LATCH



Latches are building blocks of sequential circuits and these can be built from logic gates

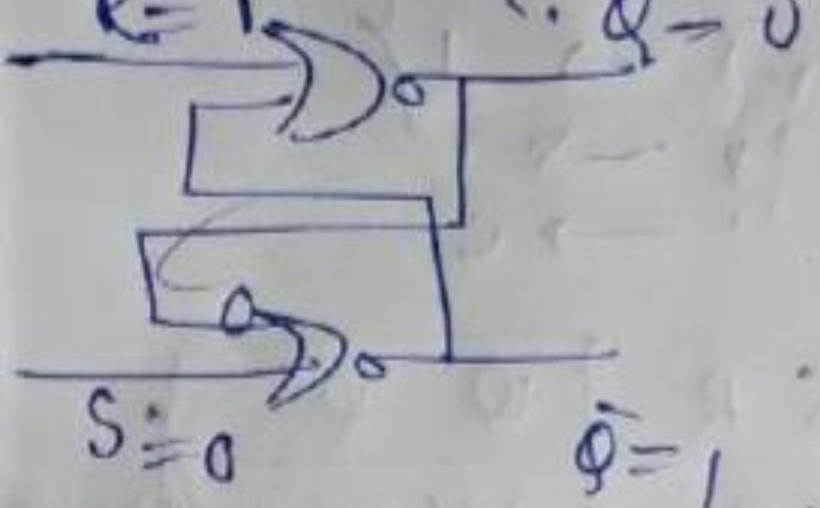
SR NOR latch

S R	Q $\bar{Q}$
00	
01	
10	

NOR

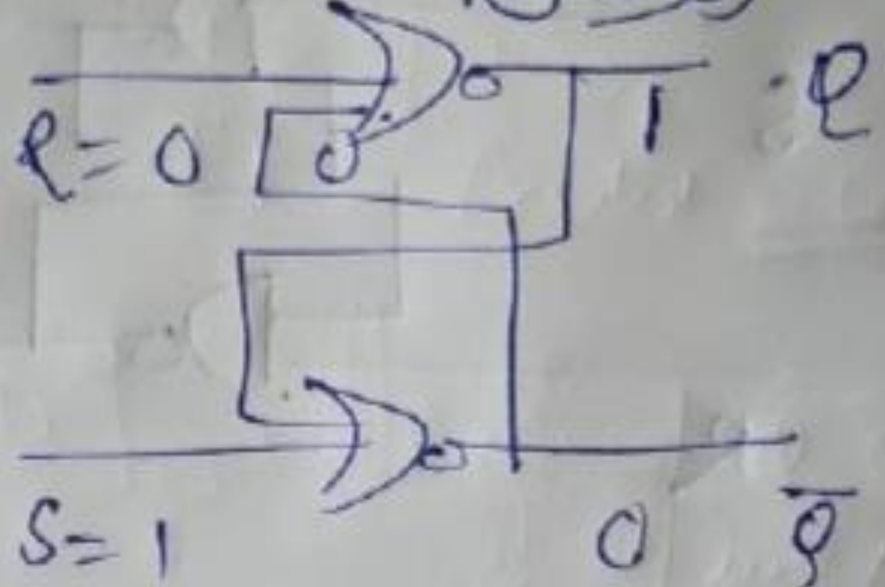
00 →	1
10 →	0
01 →	0
11 →	0

Level more



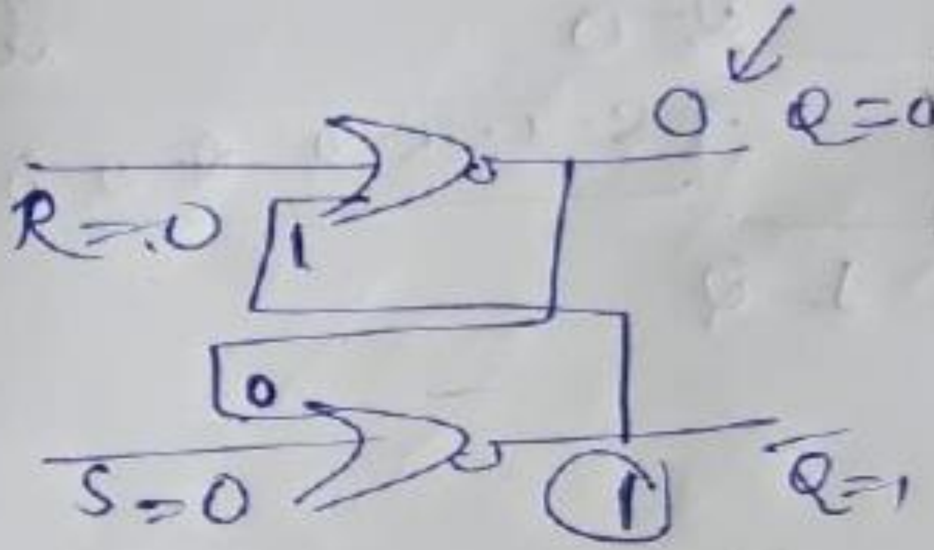
$S=0 ; R=1 ; Q=0 ; \bar{Q}=1$

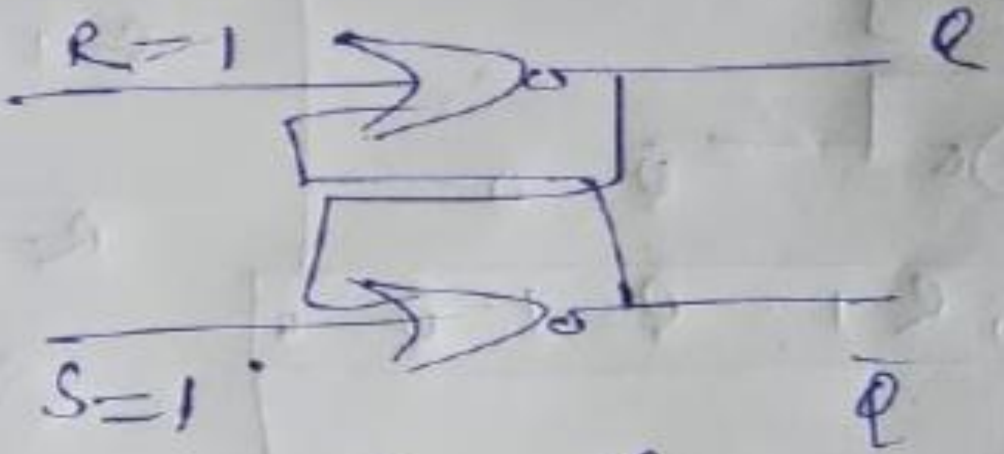
Reset



$S=1 \quad R=0 ; Q=1 ; \bar{Q}=0$

Set





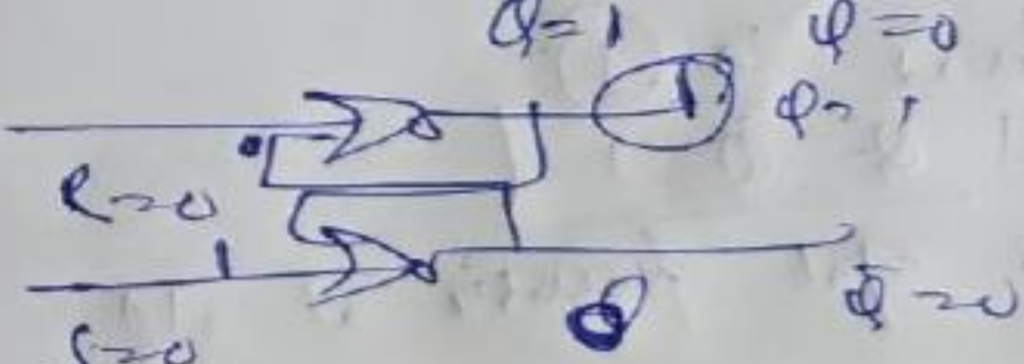
we assume previous state

$Q=0 ; \bar{Q}=1$

$Q=0 ; \bar{Q}=1$

Q=0 ; ~~$\bar{Q}$=0~~

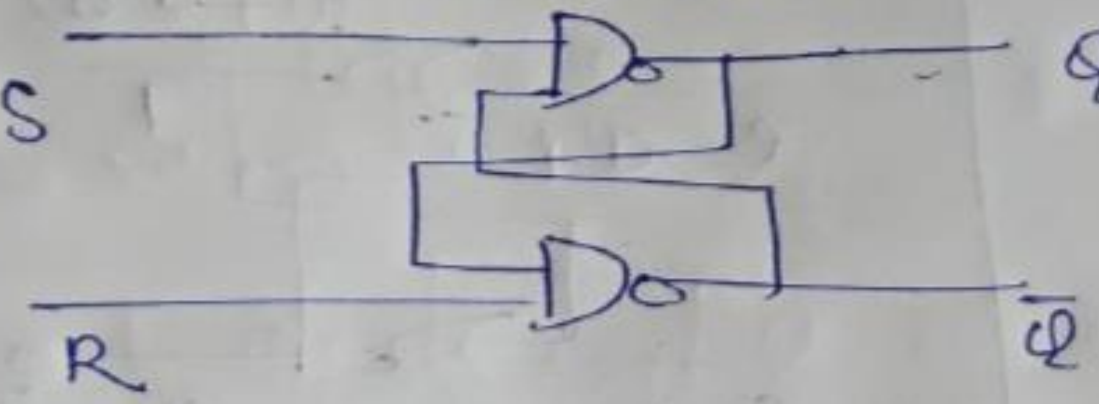
not possible



S	R	Q	$\bar{Q}$
0	0	Memory	
0	1	0	[illegible]
1	0	1	0
1	1	Invalid (Not used)	

SR NAND latch FF

'0' → bre Imore

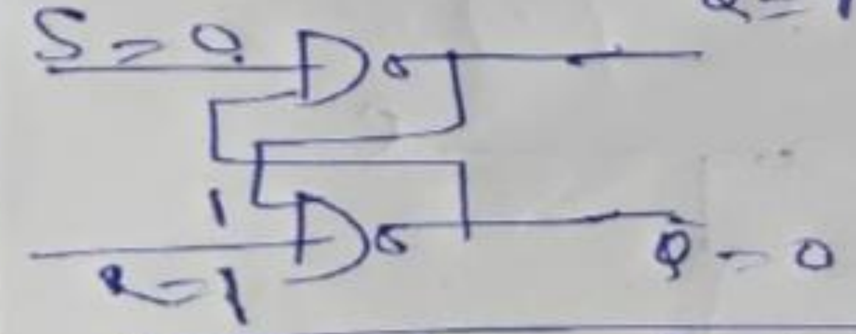


NAND

- 0 0 → 1
- 0 1 → 1
- 1 0 → 1
- 1 1 → 0

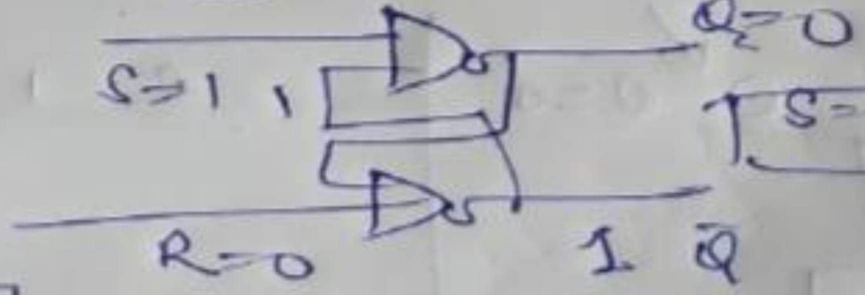
S R	Q $\bar{Q}$
0 0	Not used / invalid
0 1	1 0
1 0	0 1
1 1	Memory / No change

(1)

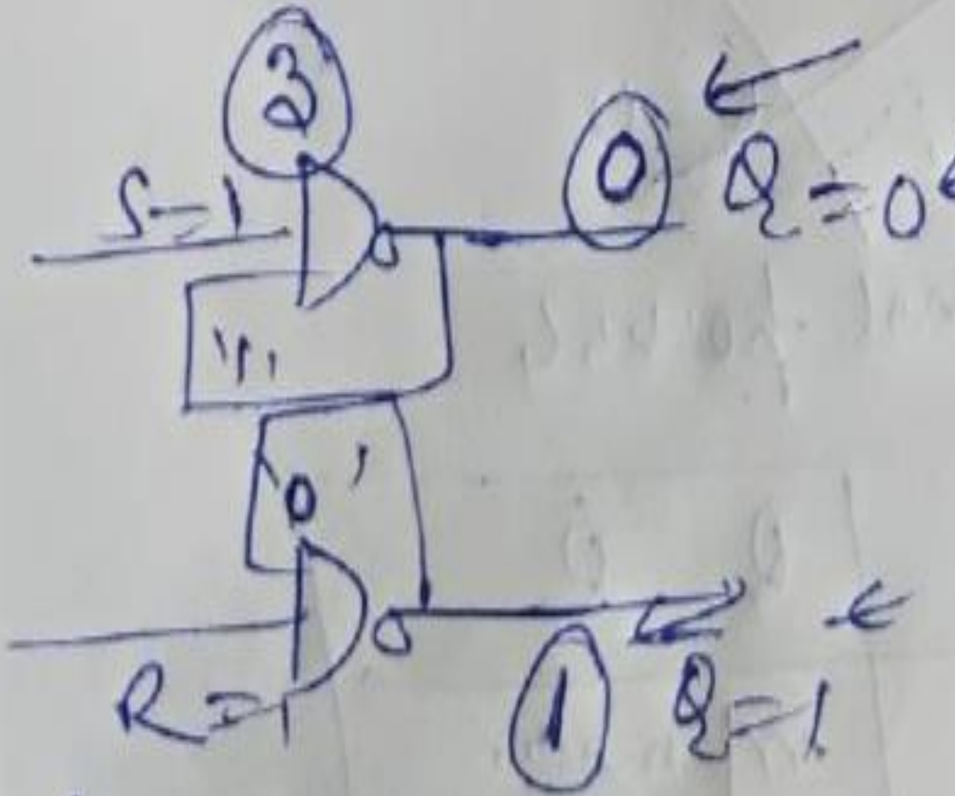


S=0; R=1; Q=1; $\bar{Q}$=0

(2)

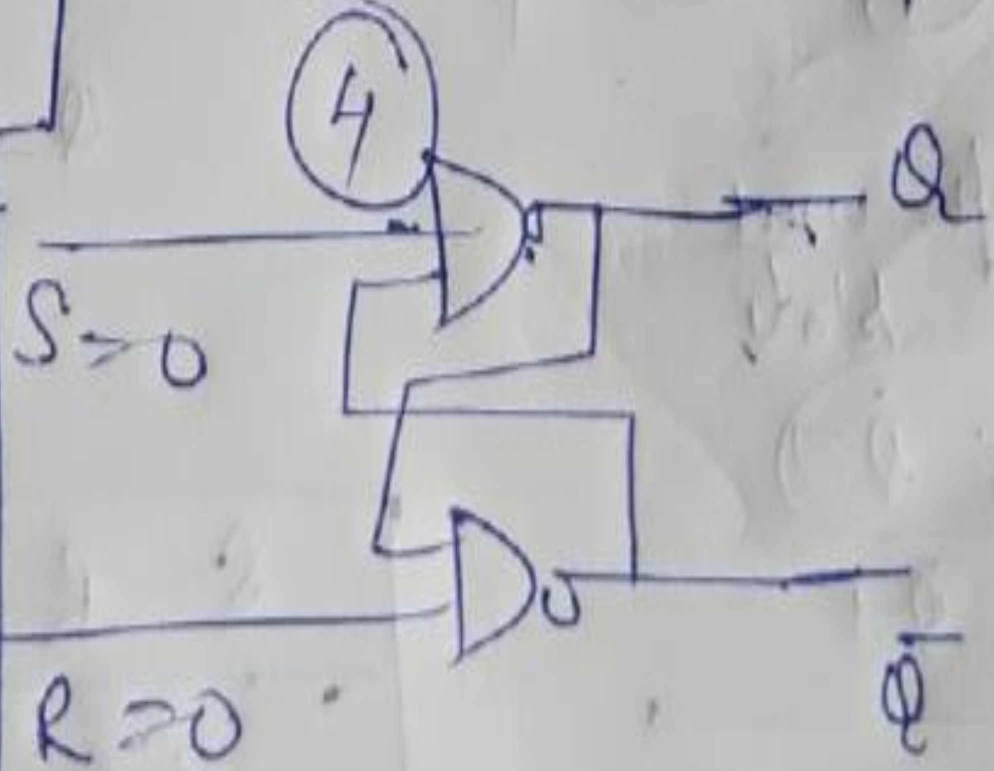


S=1; R=0; Q=0; $\bar{Q}$=



Since both '1' so assume previous state-

Because nothing is forced to 1.



Both forced to 1

$\boxed{Q = \bar{Q} = 1}$

Which is not possible at all.

Flip Flop

- Flip Flop is a sequential circuit which is used to store single bit of information at a time i.e., 0 ot 1.

- Used:-Registers

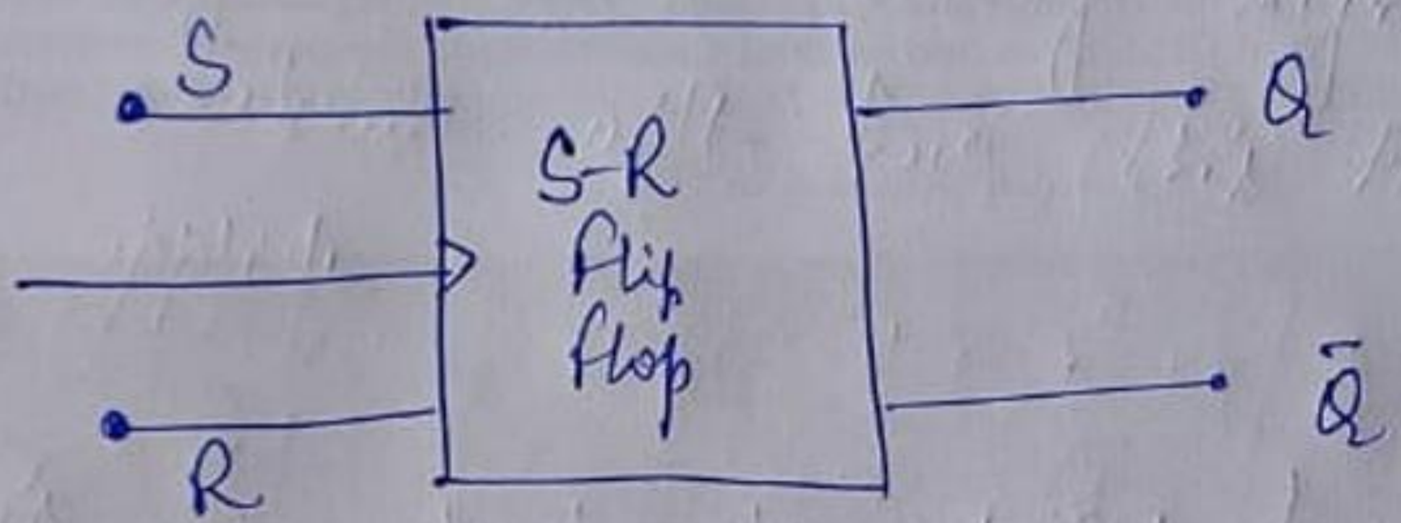
SR FLIP FLOP

- SR Flip Flop is the set reset flip flop. It consist of SR latch with clock circuit.

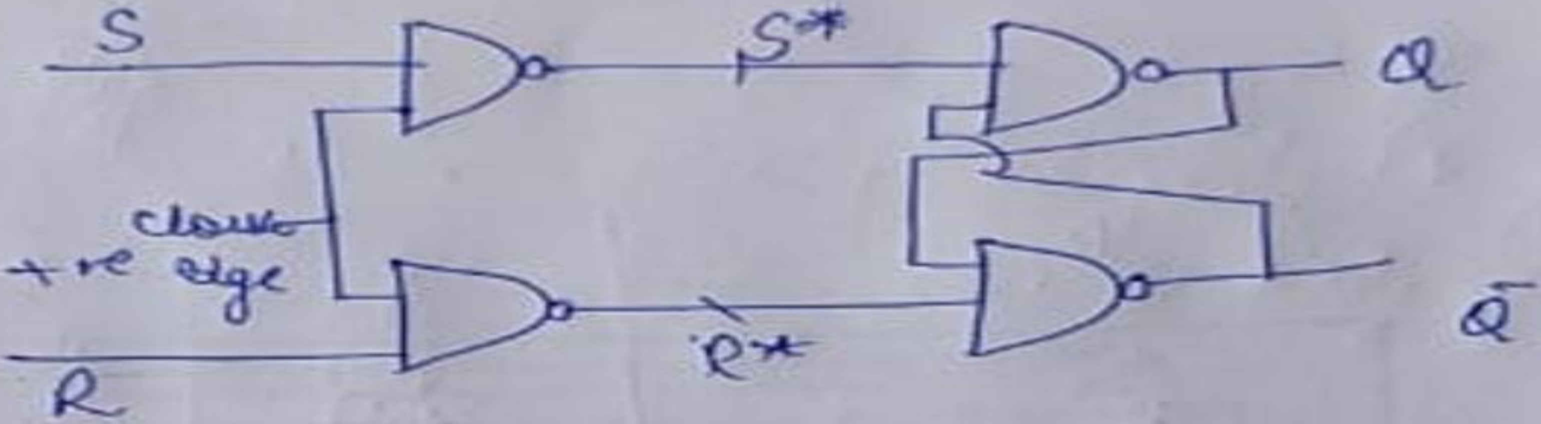
- It may be positive edge triggered or negative edge triggered.

- Triggering is the process of change of state of flop by applying input signal.

① S-R flip flop



positive edge triggered

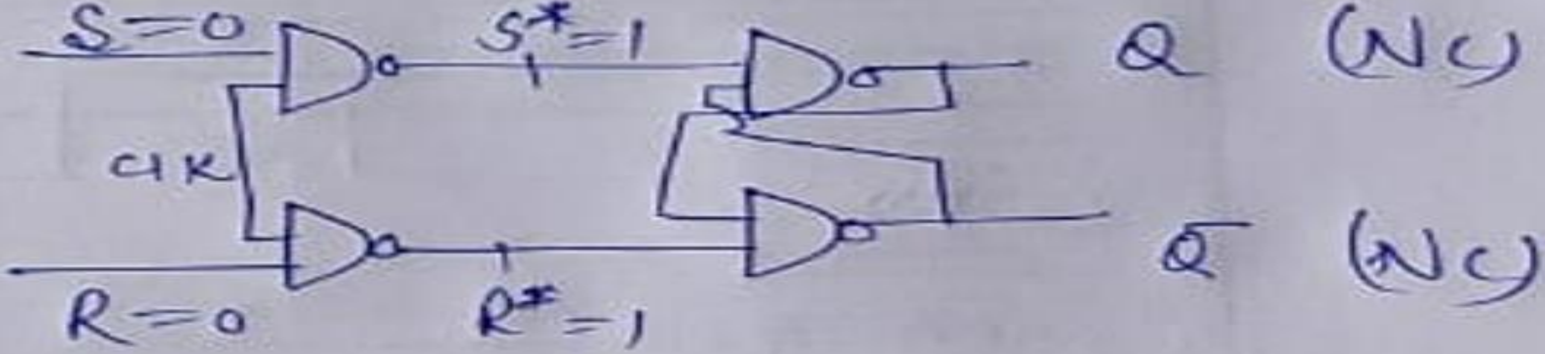


(SR NAND Latch)

S^*	R^*	Q $\bar{Q}$
0	0	Invalid
0	1	1 0
1	0	0 1
1	1	NC No change

Case I :-

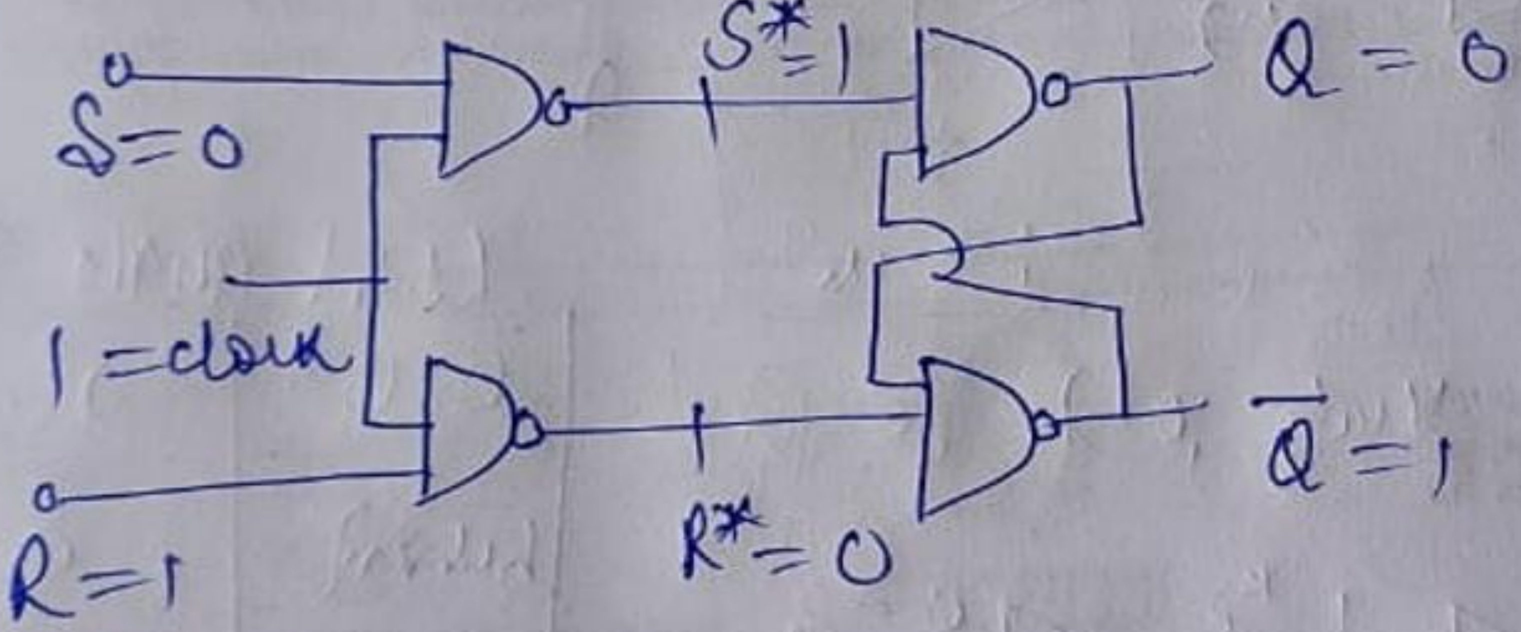
S=0, R=0, clock = ↑



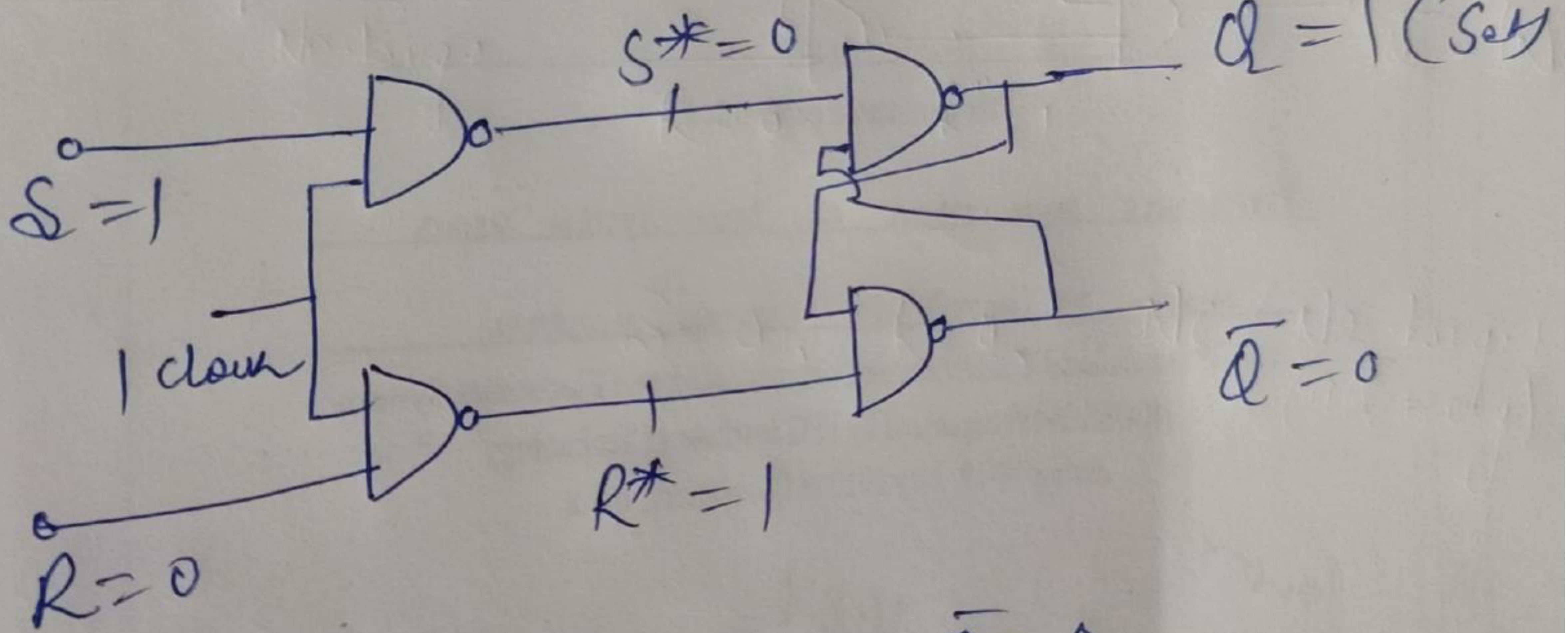
S=0, R=0, NC (No change)
(Memory)

Case II :-

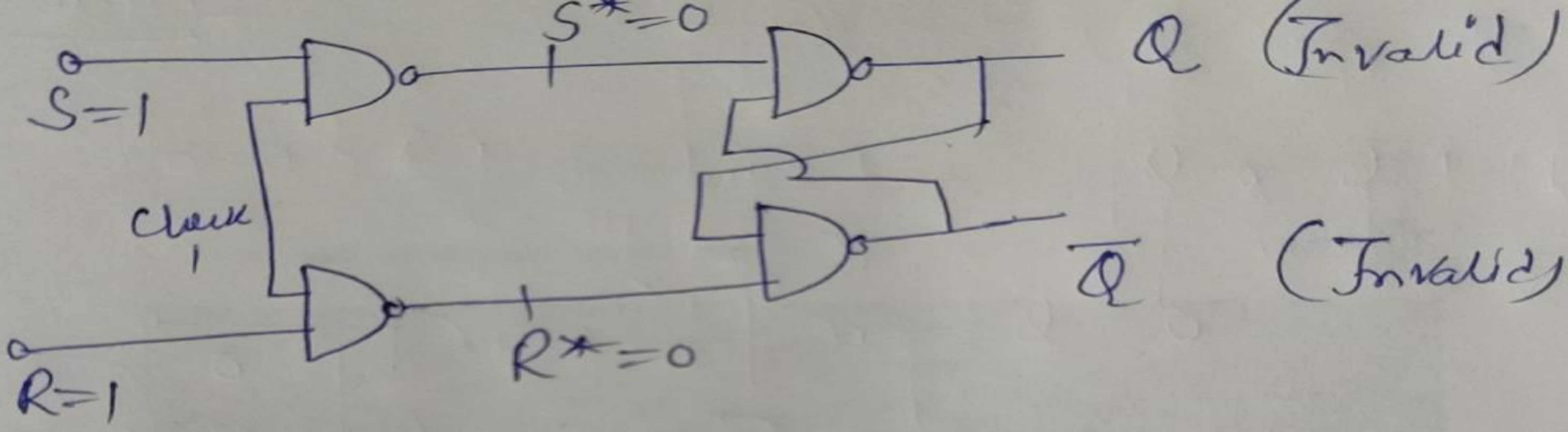
S=0 & R=1 , clock = 1   Q=0, $\bar{Q}$=1 (Reset)



Case III :-

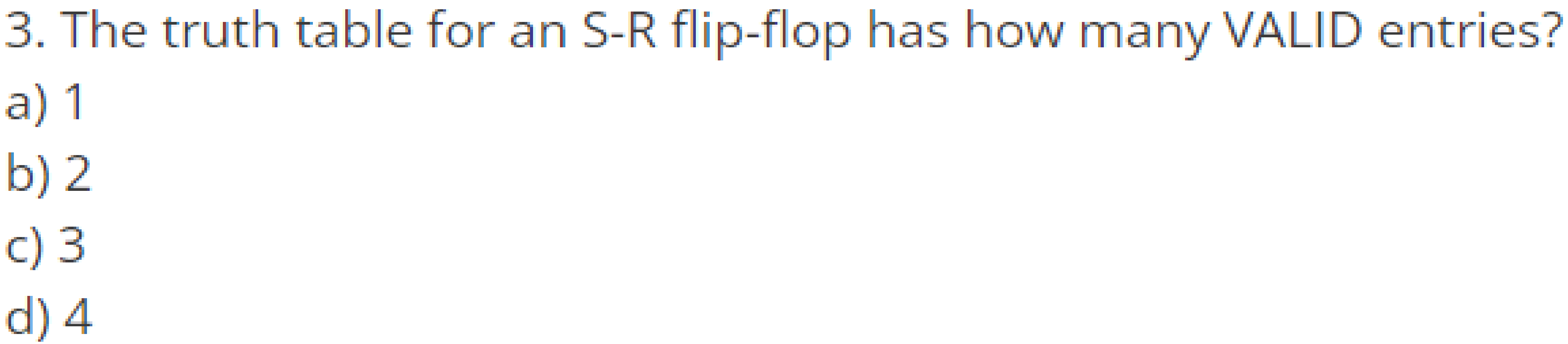


$S = 1$; $R = 0$; $Q = 1$; $\bar{Q} = 0$



clock	S R	Q	$\overline{Q}$	
↑	0 0	NC	NC	
↑	0 1	0	1	Reset
↑	1 0	1	0	Set
↑	1 1	Invalid		

MCQ



3. The truth table for an S-R flip-flop has how many VALID entries?

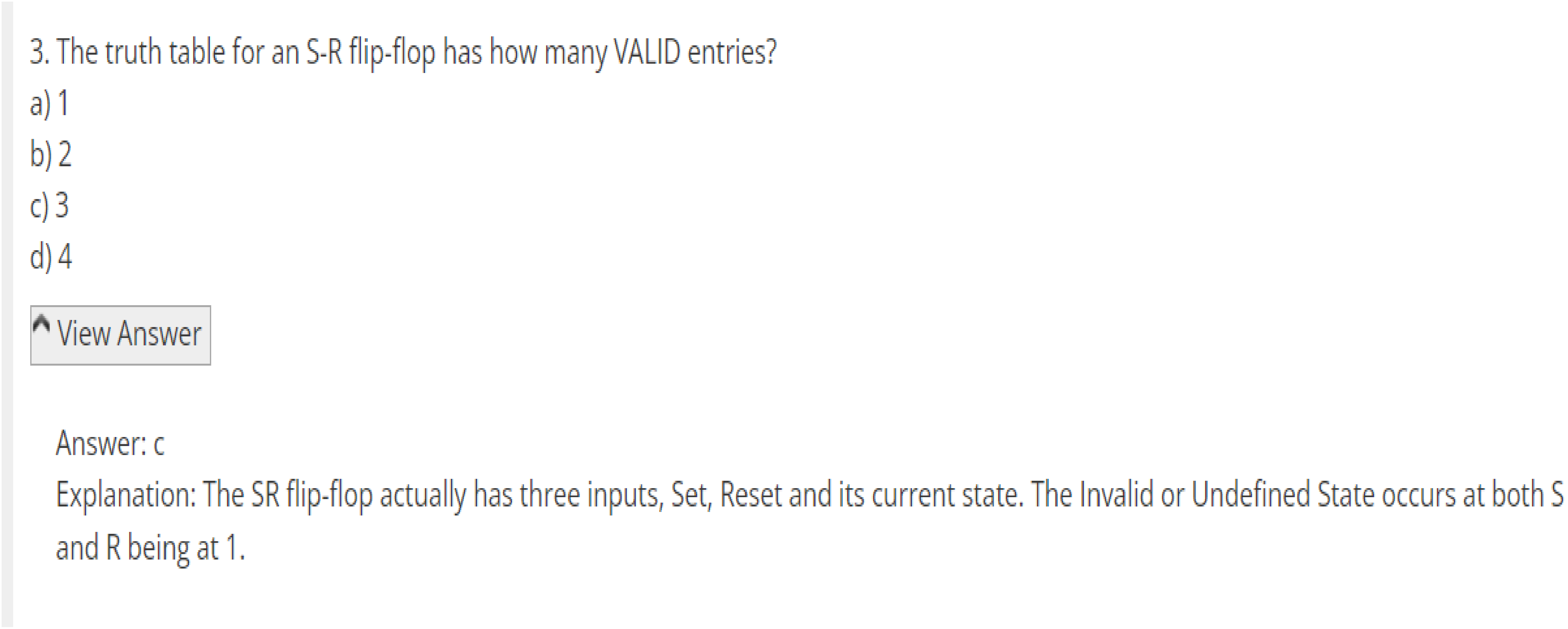
a) 1

b) 2

c) 3

d) 4

MCQ



3. The truth table for an S-R flip-flop has how many VALID entries?

a) 1

b) 2

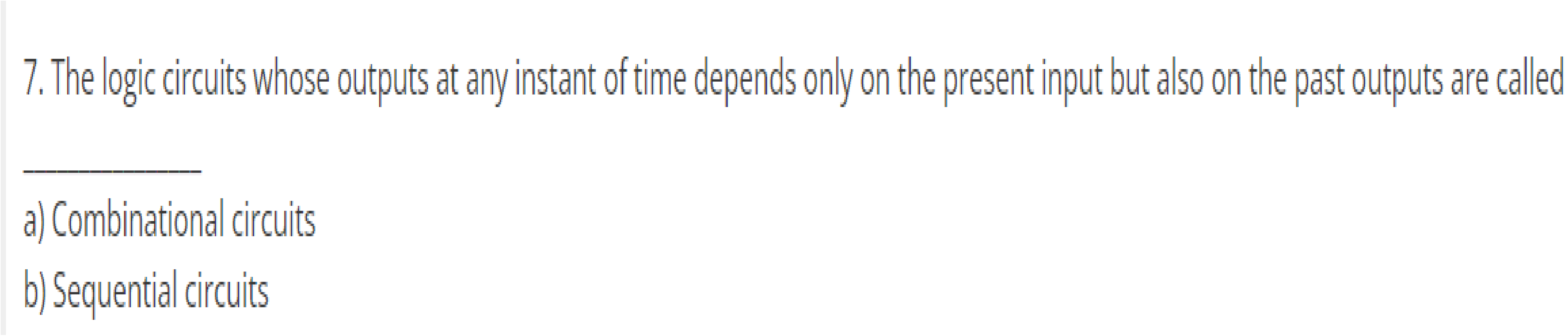
c) 3

d) 4

Answer: c

Explanation: The SR flip-flop actually has three inputs, Set, Reset and its current state. The Invalid or Undefined State occurs at both S and R being at 1.

MCQ



7. The logic circuits whose outputs at any instant of time depends only on the present input but also on the past outputs are called

____

a) Combinational circuits

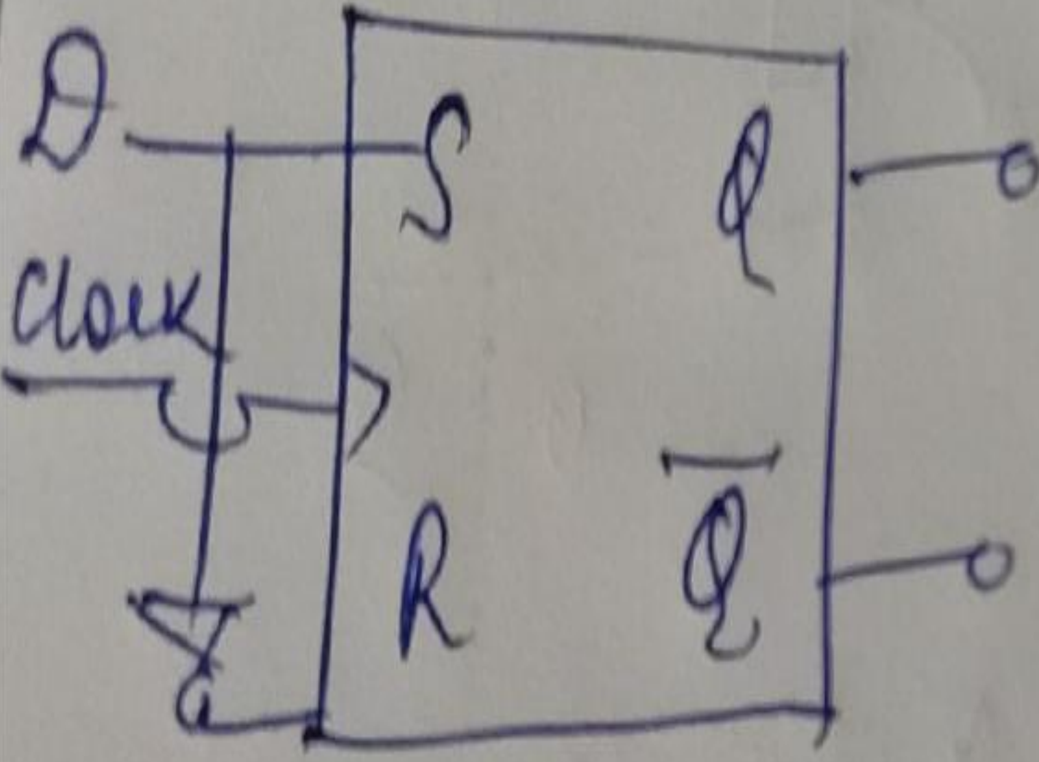
b) Sequential circuits

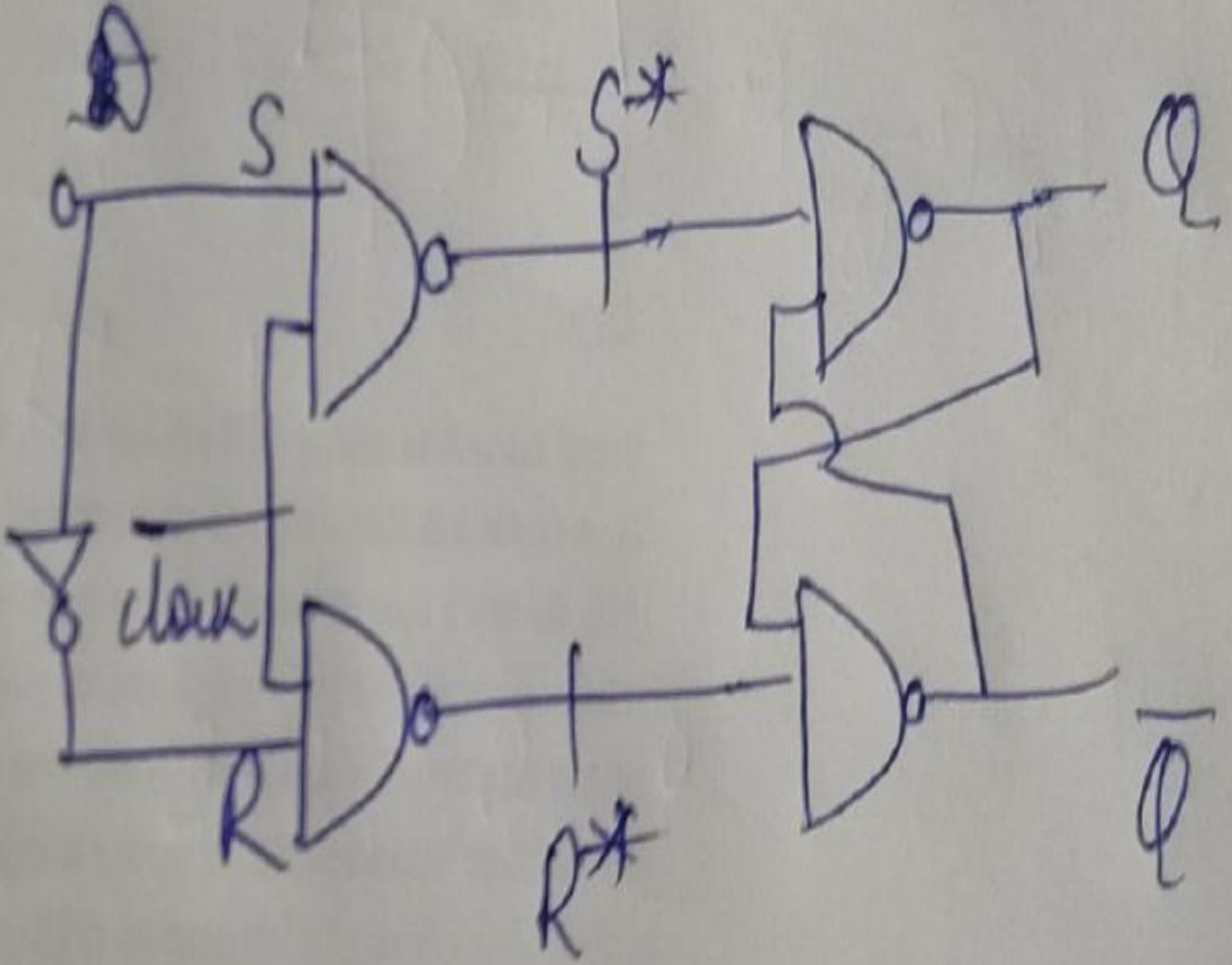
MCQ

- Answer: b
  Explanation: In sequential circuits, the output signals are fed back to the input side.

D Flip Flop (Delay Flip Flop)

- It can be designed from S-R Flip Flop by putting an inverter or NOT gate in S- R Flip Flop.


D
Clock
S
Q
R
$\overline{Q}$


D
S
S^*
Q
$\overline{clock}$
R
R^*
$\overline{Q}$

Case I → $D = 1 \Rightarrow S = 1 ; \; R = 0$

$Q = 1 ; \; \overline{Q} = 0$

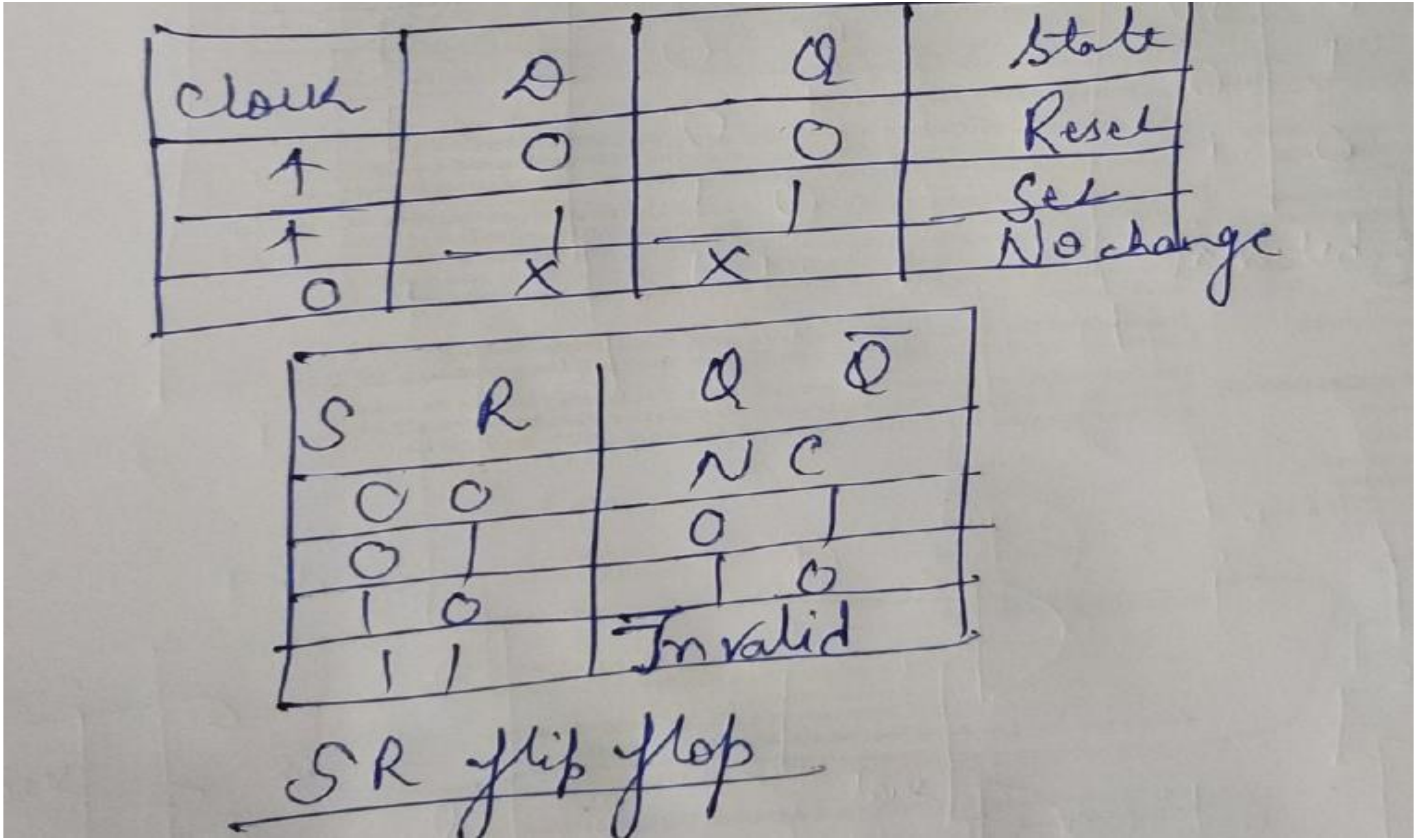
(Set)

Case II → $D = 0 \Rightarrow S = 0 ; \; R = 1$

$Q = 0 ; \; \overline{Q} = 1$

(Reset)

clock	D	Q	State
↑	0	0	Reset
↑	1	1	Set
0	X	X	No change

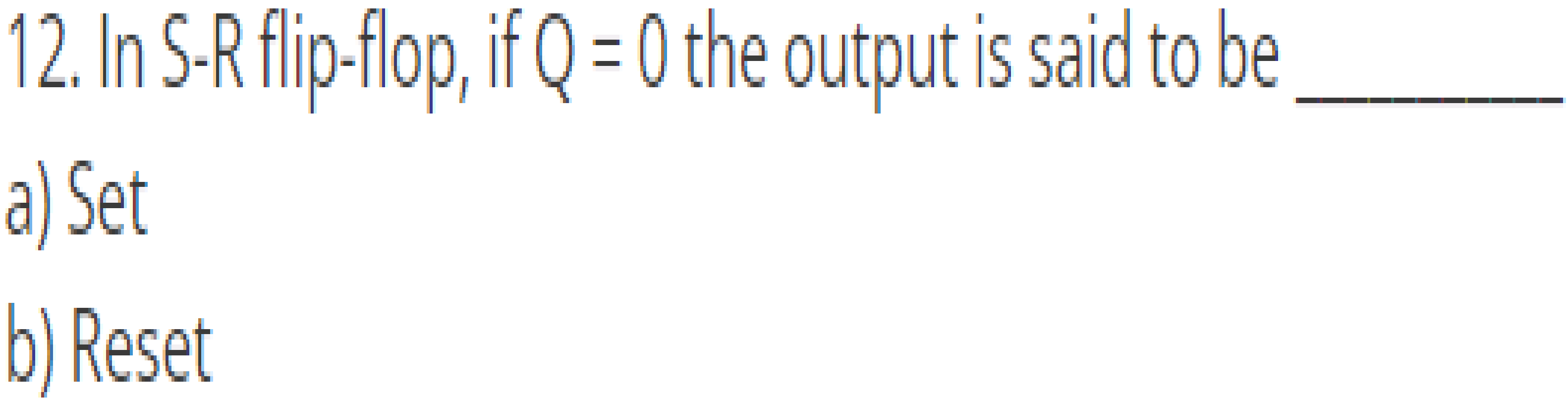


clock	D	Q	State
↑	0	0	Reset
↑	1	1	Set
0	X	X	No change

S	R	Q	$\overline{Q}$
0	0	N C	
0	1	0	1
1	0	1	0
1	1	Invalid	

SR flip flop

MCQ



12. In S-R flip-flop, if Q = 0 the output is said to be ________

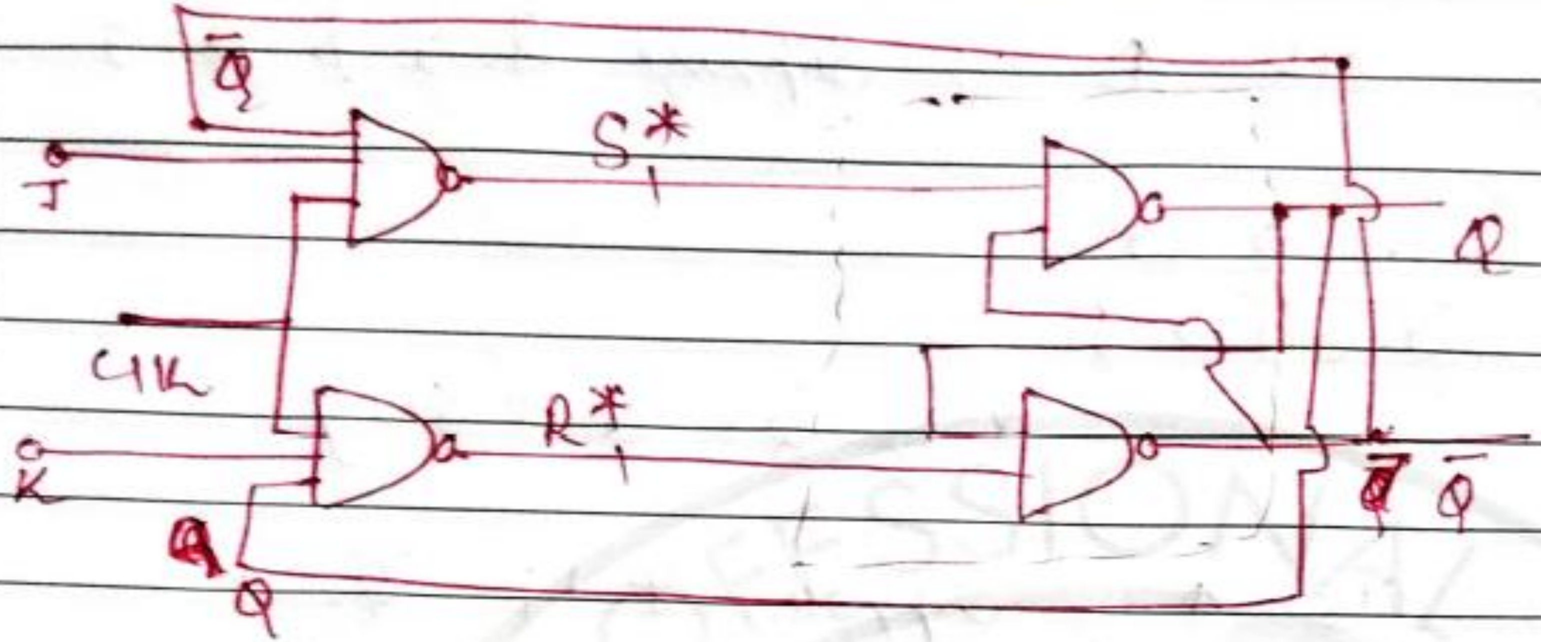
a) Set

b) Reset

MCQ

- Answer: b
- Explanation: In S-R flip-flop, if Q = 0 the output is said to be reset and set for Q = 1

J-K flip flop



SR NAND Latch

S*	R*	Q	Q̄
0	0	Invalid	
0	1	1	0
1	0	0	1
1	1	NC/memory	

Case 1:- Clk → ↑ ; J = 0 ; K = 0

$R^* = 1$ & $S^* = 1$

Case2:- Clk $\rightarrow \uparrow$; $J=0$; $K=1$; $Q=0$; $\bar{Q}=1$ (Reset)

a) If suppose $Q=0$ & $\bar{Q}=1$

$$\left.\begin{aligned} S^* &= \overline{\bar{Q}.J.\text{clk}} = 1 \\ R^* &= \overline{Q\cdot K.\text{clk}} \Rightarrow 1 \end{aligned}\right\} \text{NC state}$$

so $Q=0$ & $\bar{Q}=1$ (Reset)

(b) If suppose $Q=1$ & $\bar{Q}=0$

$$S^* = \overline{0.0.1} \Rightarrow 1$$

$$R^* = \overline{1.1.1} \Rightarrow 0 \Rightarrow \quad Q=0 \text{ \& } \bar{Q}=1 \text{ (Reset)}$$

Case III :- $S^* = \overline{\bar{Q}\cdot J\cdot Clk}$

$R^* = \overline{Q\cdot K\cdot Clk}$

previous state

A) J=1 ; K=0 ; Clk = ↑ ; Suppose $Q=0$ & $\bar{Q}=1$

$S^* = \overline{1\cdot1\cdot1} \Rightarrow 0$

$R^* = \overline{0\cdot0\cdot1} \Rightarrow 1$

$Q=1;\ \bar{Q}=0$ } Set

(B) J=1 ; K=0, Clk=↑ ; Suppose $Q=1$; $\bar{Q}=0$

$S^* = \overline{0\cdot1\cdot1} \Rightarrow 1$ } NC (No change)

$R^* = \overline{1\cdot0\cdot1} \Rightarrow 1$ } means

$Q=1;\ \bar{Q}=0$ (Remain same)

Case IV ; CLK = ↑ ; J=1, K=1

$$S^* = \overline{\bar{Q}\cdot J\cdot CLK}$$
$$R^* = \overline{Q\cdot K\cdot CLK}.$$

Previous state assume

$Q=1$; $\bar{Q}=0$

$$\left.\begin{aligned} S^* &= \overline{0.1.1} \Rightarrow 1 \\ R^* &= \overline{1.1.1} \Rightarrow 0 \end{aligned}\right\} \begin{aligned} &Q=0 \,\&\, \bar{Q}=1 \\ &\text{Reset} \end{aligned}$$

$Q_{n+1} = 0$ & $\overline{Q_{n+1}} = 1$

$Q_{n+1} = \overline{Q_n}$

$Q=0$ & $\bar{Q}=1$

$$\left.\begin{aligned} S^* &= \bar{1} = 0 \\ R^* &= 1 \end{aligned}\right\} \text{SET}$$

$Q=1$; $\bar{Q}=0$

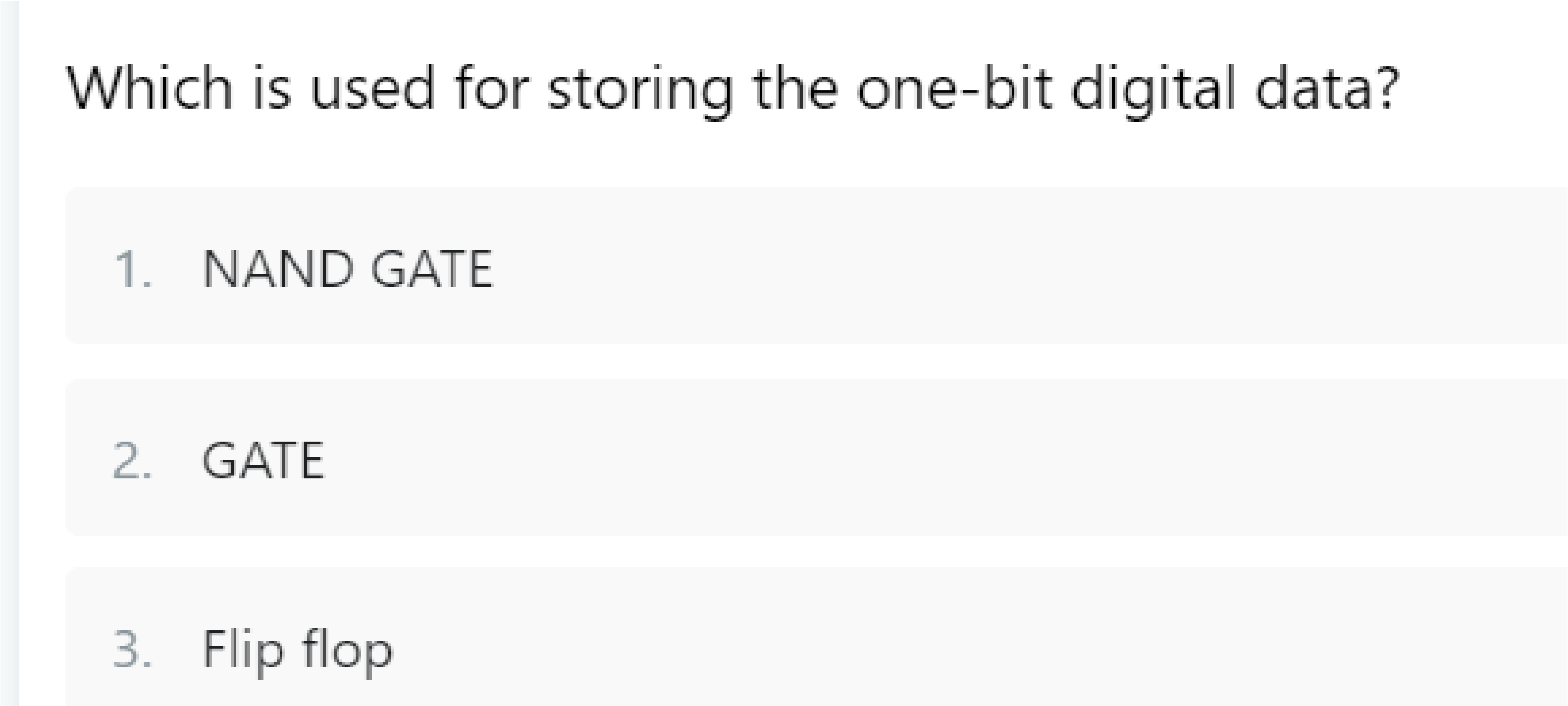
$Q_{n+1} = 1$ ($\overline{Q_n}$ }

(Toggling)

Clock	J	K	Q_{n+1}, O/P	$\bar{Q}_{n+1}$,	State
↑	0	0	NC	NC	Hold
↑	0	1	0	1	Res
↑	1	0	1	0	Set
↑	1	1	$\bar{Q}_n$	Q_n	Toggle

Thus as long as J=K=1; the o/p will keep toggling indefinitely. This multiple toggling in JK is called Race Around condition.

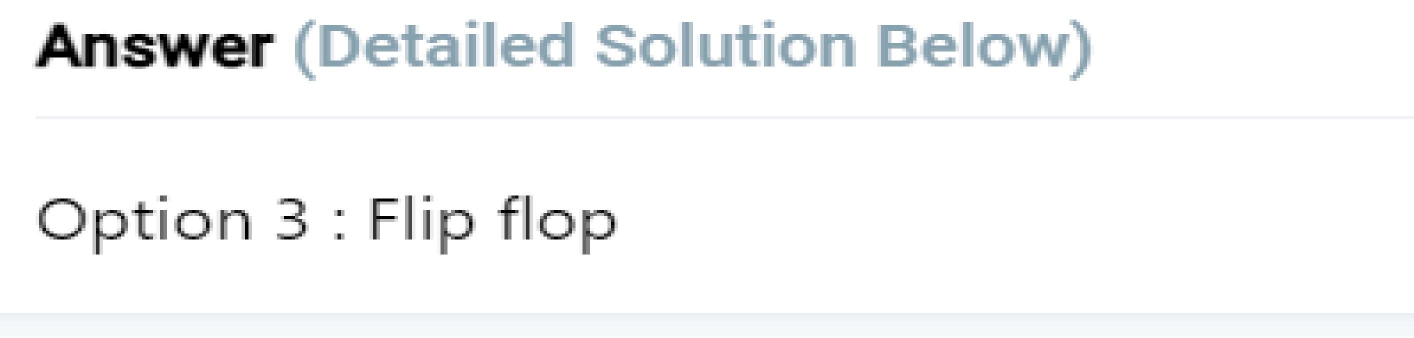
MCQ



Which is used for storing the one-bit digital data?

1. NAND GATE
2. GATE
3. Flip flop

MCQ

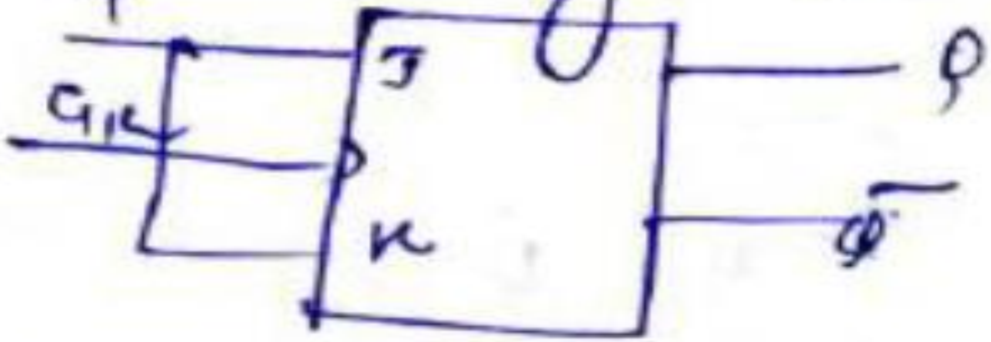


Answer (Detailed Solution Below)

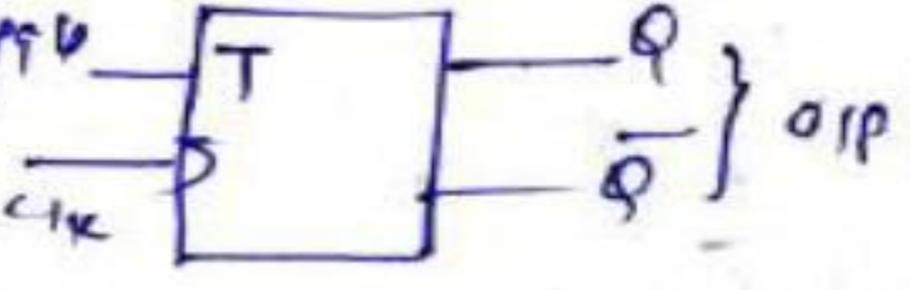
Option 3 : Flip flop

T-ff (toggle flip flop)

Toggle flip flop is basically a JK flip flop with J & K terminals permanently connected together. It has only one input T.



JK ff is conv to T flip f.



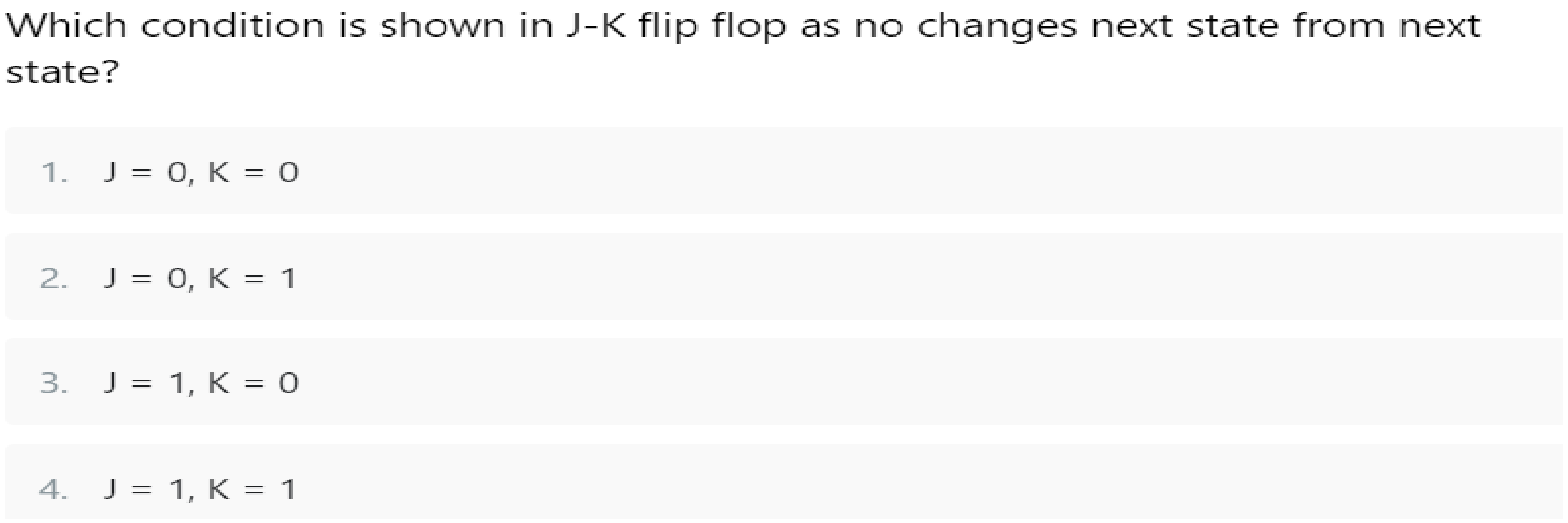
Logic symbol of positive edge triggered T flipflop triggered T flipflop

Case 1: if T=0 ; J=0 ; K=0 NC

if T=1 ; J=1 ; K=1 o/p will toggle corresponding to every leading edge of clock signal.

Clk	T	Q_{n+1}	$\bar{Q}_{n+1}$	
↑	1	$\bar{Q}_n$	Q_n	Toggle
↓	0	Q_n	$\bar{Q}_n$	NC

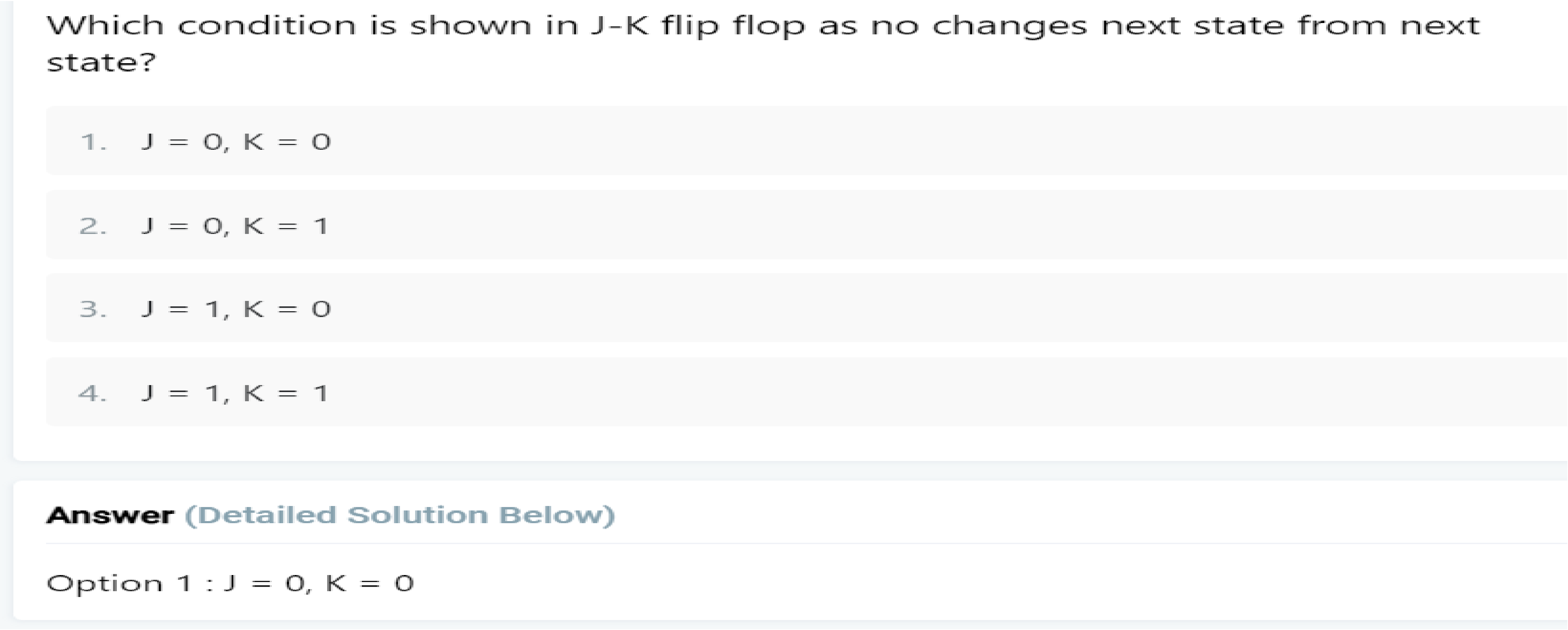
MCQ



Which condition is shown in J-K flip flop as no changes next state from next state?

1. J = 0, K = 0
2. J = 0, K = 1
3. J = 1, K = 0
4. J = 1, K = 1

MCQ



Which condition is shown in J-K flip flop as no changes next state from next state?

1. J = 0, K = 0
2. J = 0, K = 1
3. J = 1, K = 0
4. J = 1, K = 1

Answer (Detailed Solution Below)

Option 1 : J = 0, K = 0

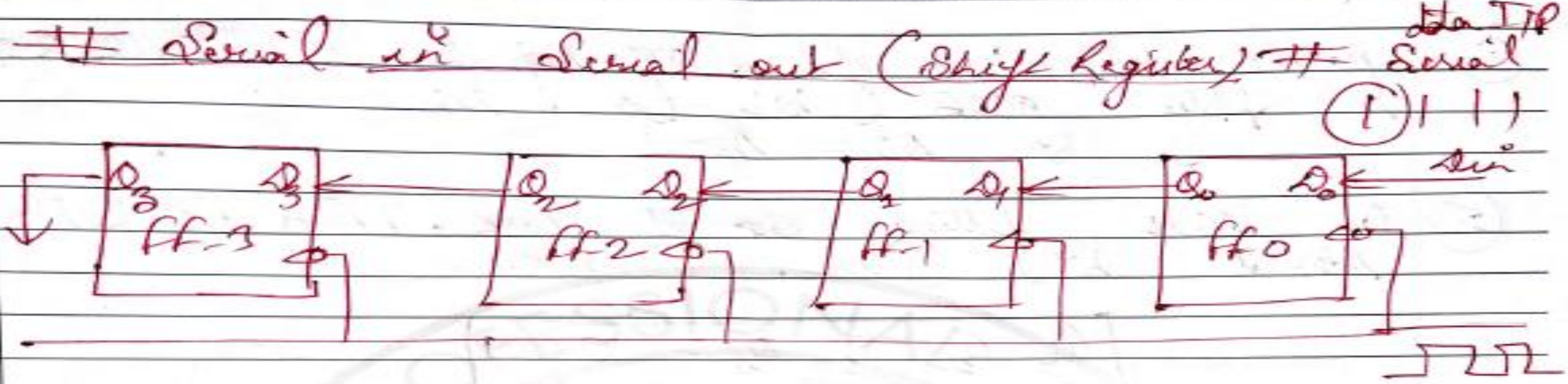
SHIFT REGISTERS

1. The binary data in a register can be moved from one flip flop to other with the application of clock pulses.

2. The registers that allow such data transfer are called as shift registers.

3. Shift registers are used for data storage, data transfer.

Serial in Serial out (Shift Register)



(1) Let all the flip flops be in the reset condition

$$\boxed{Q_3 = Q_2 = Q_1 = Q_0 = 0}$$

(2) We are going to explain the entry of a 4 bit binary no. 1111 into register.

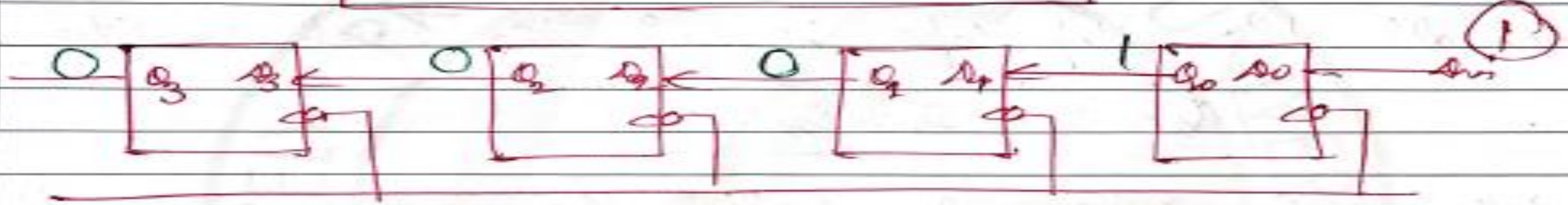
(3) MSB applied first

(4) D_0 is connected to serial data input Din. Output of ff-0 that is Q_0 is connected to the input of next flip-flop that is D_1 & soon

① Before application of clock signal let $Q_3 Q_2 Q_1 Q_0 = 0000$ & apply MSB bit to Din
so $D_{in} = D_0 = 1$

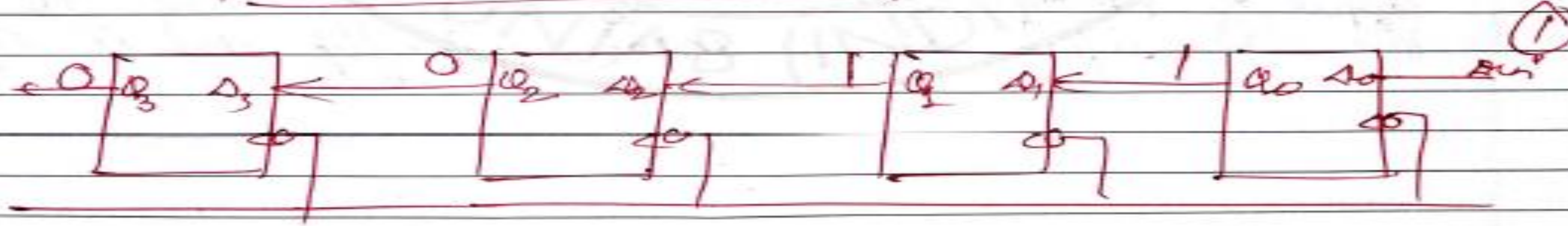
② On first falling edge of clock. FF-0 is set & stored bit in register.

$$\boxed{Q_3 \quad Q_2 \quad Q_1 \quad Q_0 = 0001}$$



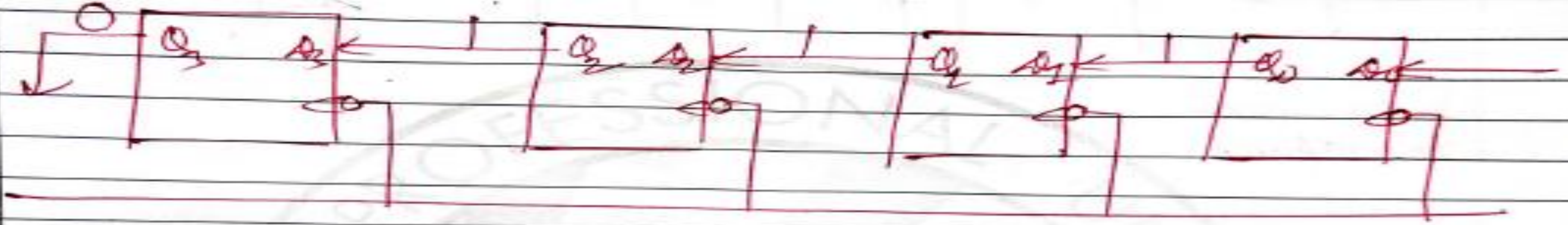
③ As soon as next negative edge of clock hits FF-1 then it set & 1 stored bit changes to

$$\boxed{Q_3 \quad Q_2 \quad Q_1 \quad Q_0 = 0011}$$



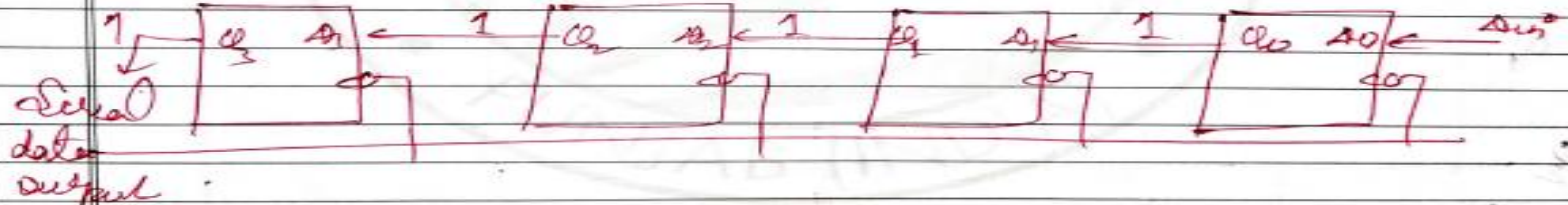
(4) As soon as, third negative clock edge hits, ff-2 bit get modified to

$$Q_3 \; Q_2 \; Q_1 \; Q_0 = 0111$$



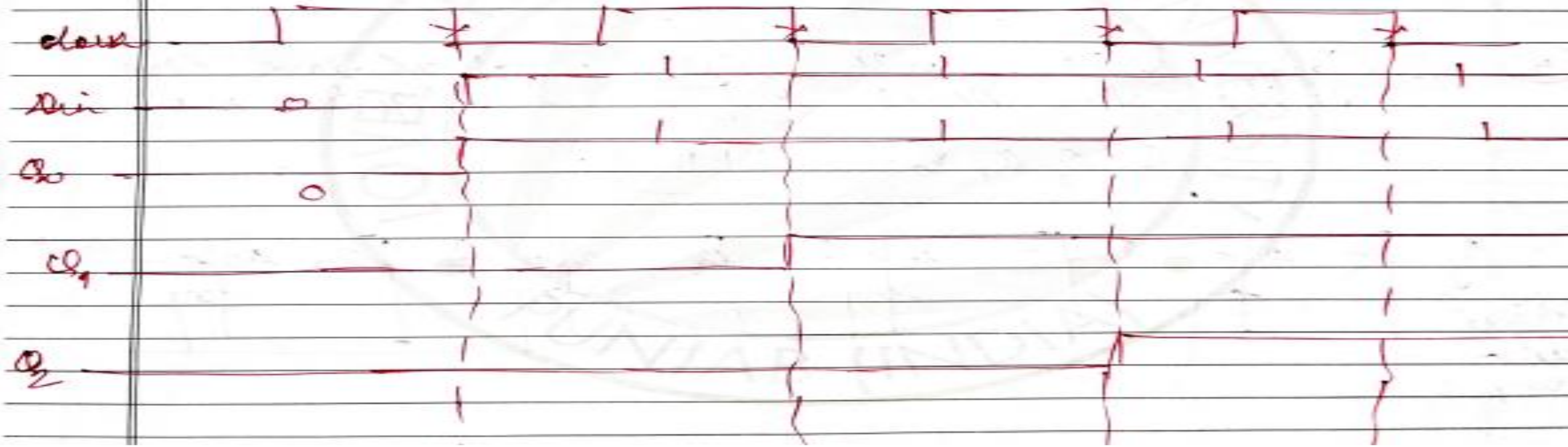
(5) Similarly Din=1 with fourth negative clock edge arriving, the stored bits in register are

$$\boxed{Q_3 \; Q_2 \; Q_1 \; Q_0 = 1111}$$

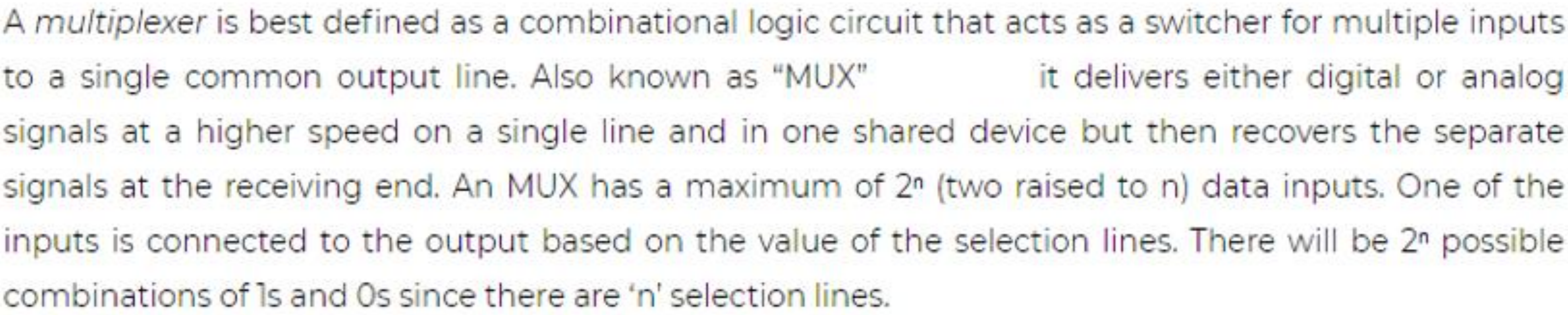


clk	Q_3	Q_2	Q_1	Q_0		D_{in} = 00
Initially	0	0	0	0		
1	0	0	0	1	←	1
1	0	0	1	1	←	1
1	0	1	1	1	←	1
1	1	1	1	1	←	1

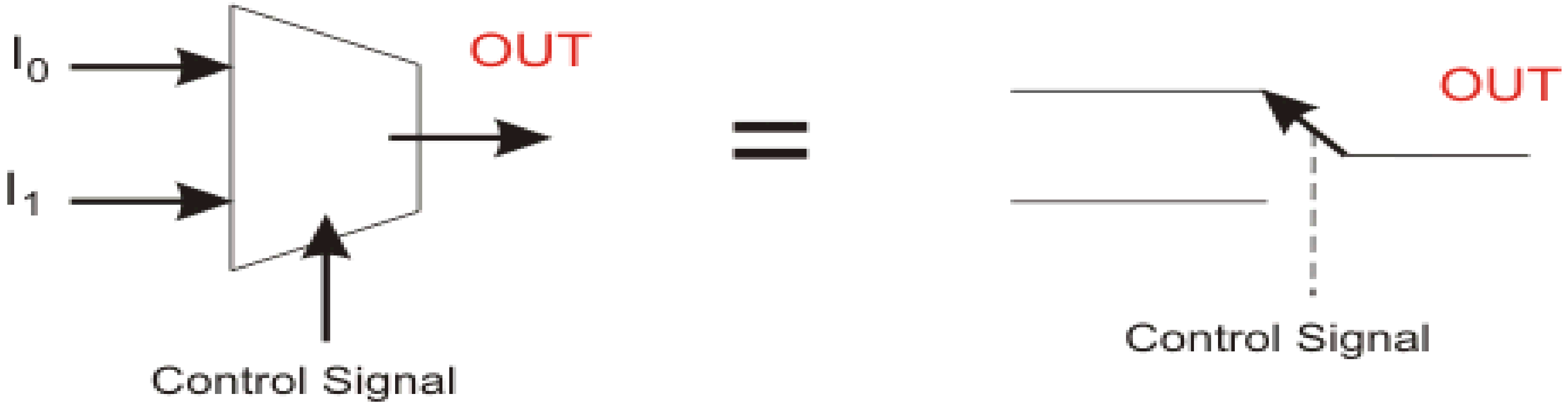
Waveforms



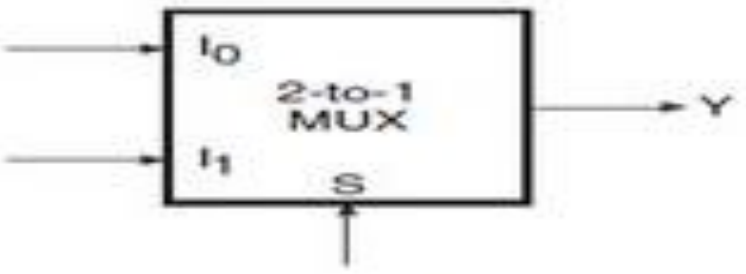
MUX



A *multiplexer* is best defined as a combinational logic circuit that acts as a switcher for multiple inputs to a single common output line. Also known as "MUX" it delivers either digital or analog signals at a higher speed on a single line and in one shared device but then recovers the separate signals at the receiving end. An MUX has a maximum of 2^n (two raised to n) data inputs. One of the inputs is connected to the output based on the value of the selection lines. There will be 2^n possible combinations of 1s and 0s since there are 'n' selection lines.



2:1 Mux



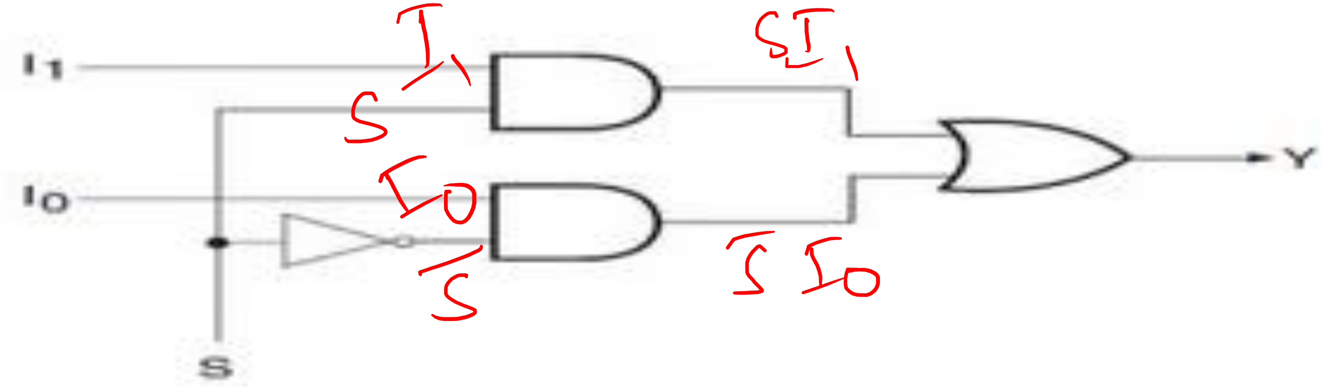
(a)

Equation (2:1) Mux

$$Y = \bar{S} I_0 + S I_1$$

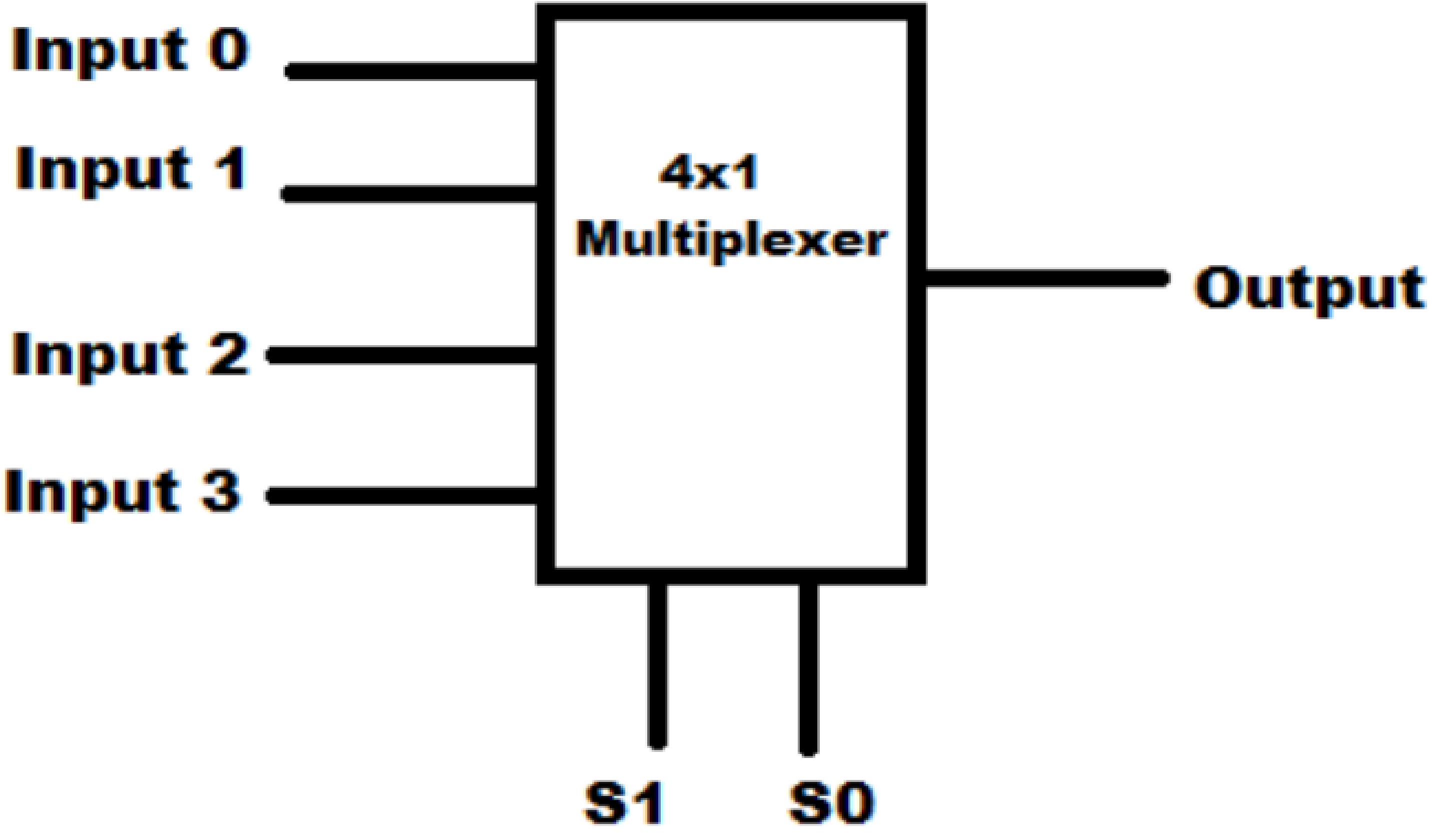
S	Y
0	I_0
1	I_1

(b)



$= \bar{S} I_0 + S I_1$

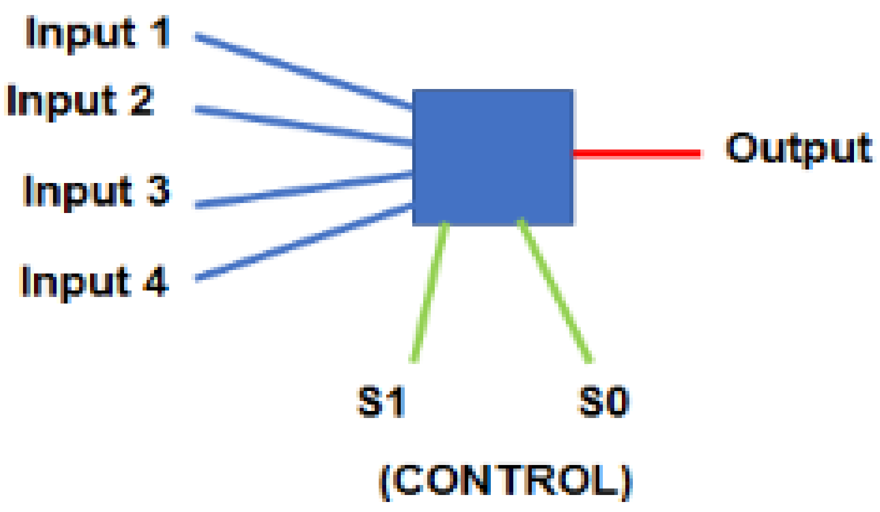
4:1 Mux



2^n :1

2^2 :1

HOW DOES A MULTIPLEXER WORK?

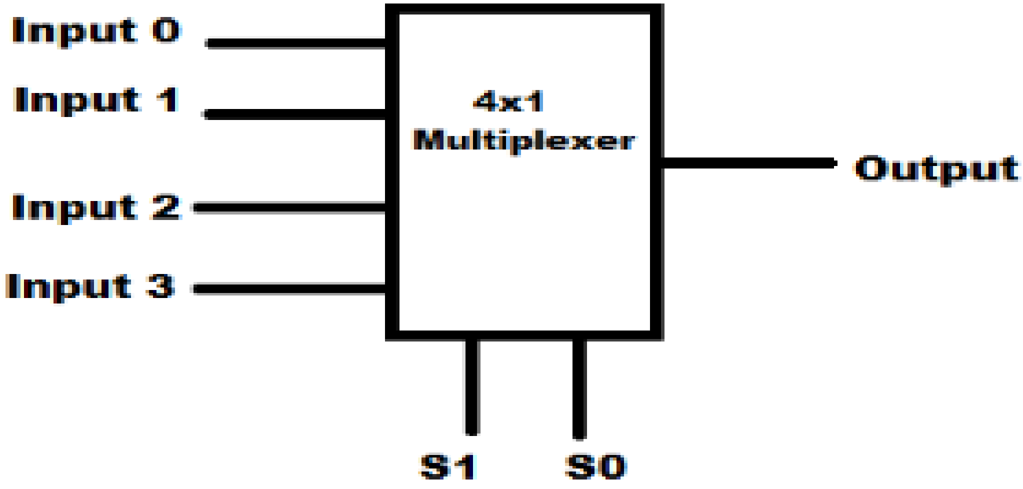


In this figure, the switch has four inputs with one output pin and you can select based on the signal given. It demonstrates the three basic parts of any multiplexer namely the input pins, output pins, and control signals.

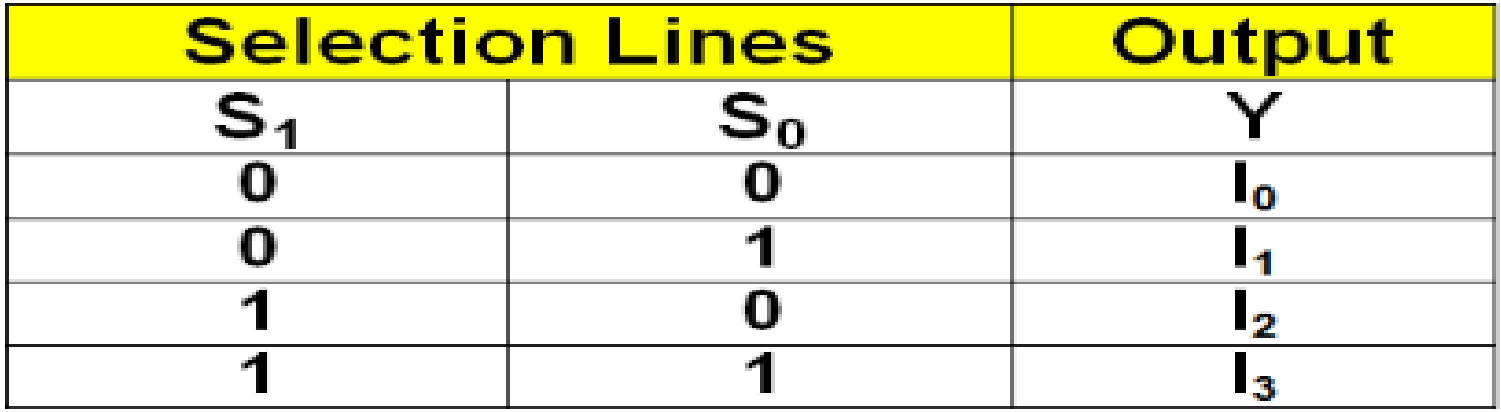
- Input Pins – these are all the available input signals from which the best required signal has to be selected. It can be analog or digital.

- Output Pin – the chosen input signal will be provided by the output pin.

- Control/Selection Pin – this selects the input pin signal. The number of control pins depends on the number of input pins.



Any of the four inputs will be connected to the output based on the combination present at these two selection lines.

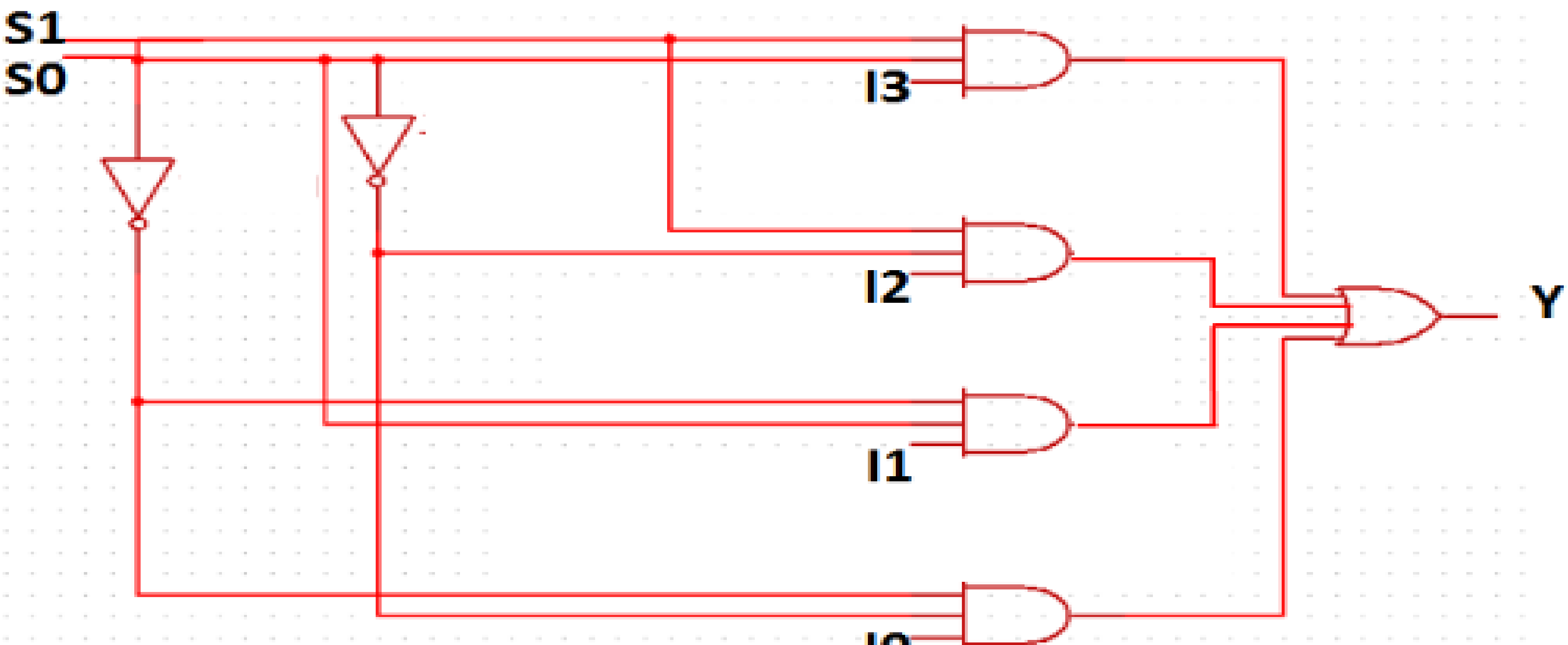


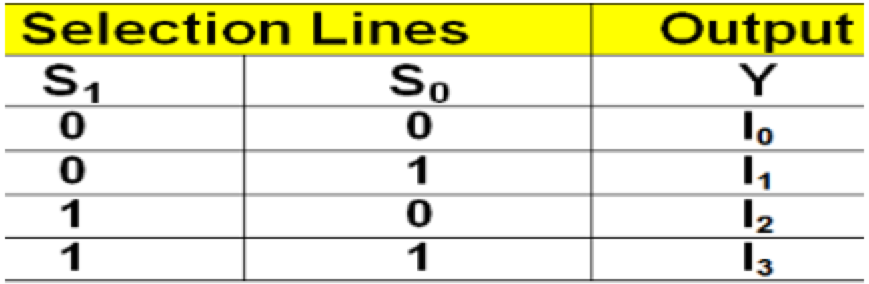
Selection Lines		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Truth Table of 4×1 Multiplexer

$$Y=S1'S0'I0+S1'S0I1+S1S0'I2+S1S0I3$$

Given the Boolean function, we can implement the 4×1 multiplexer using inverters in this circuit diagram.

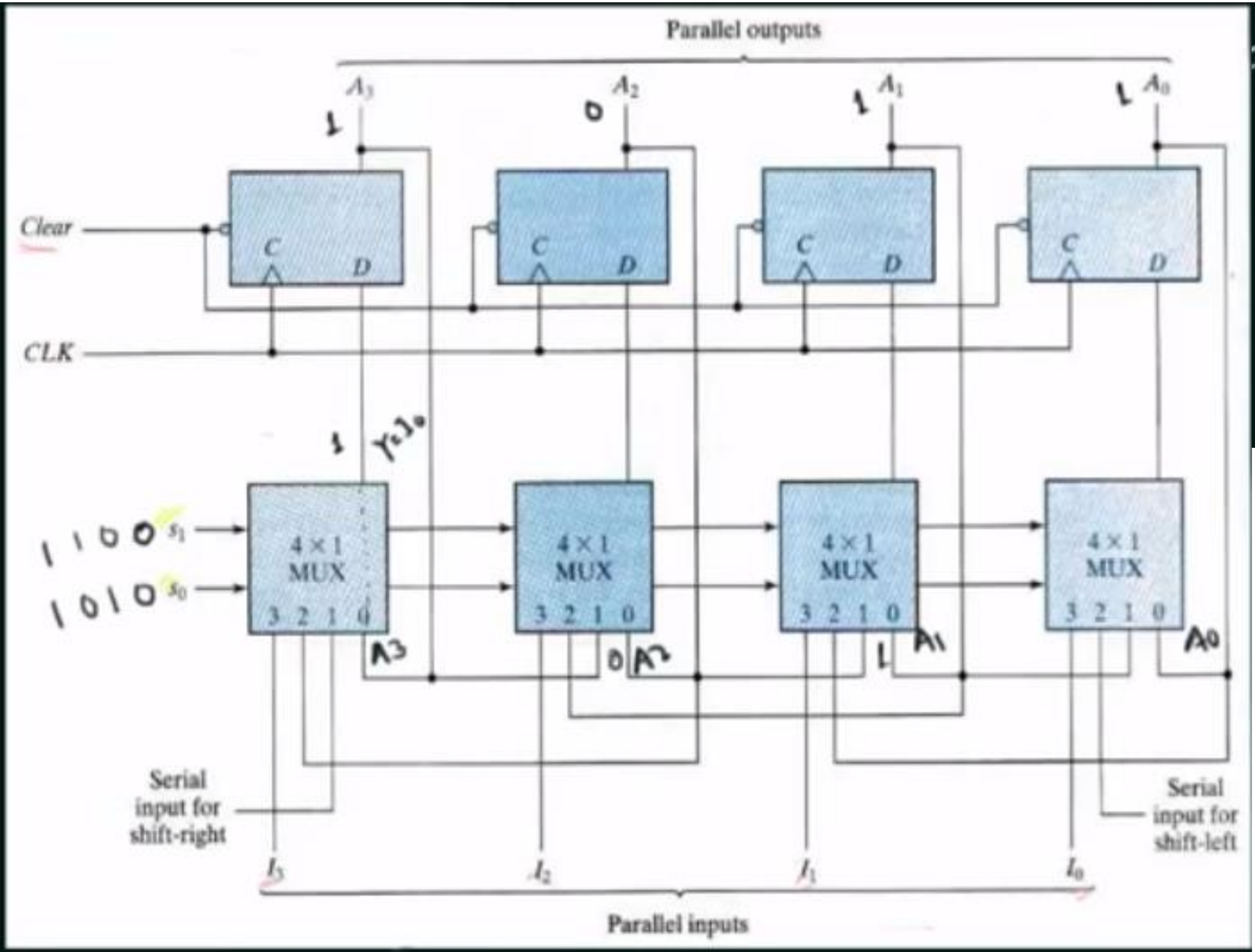


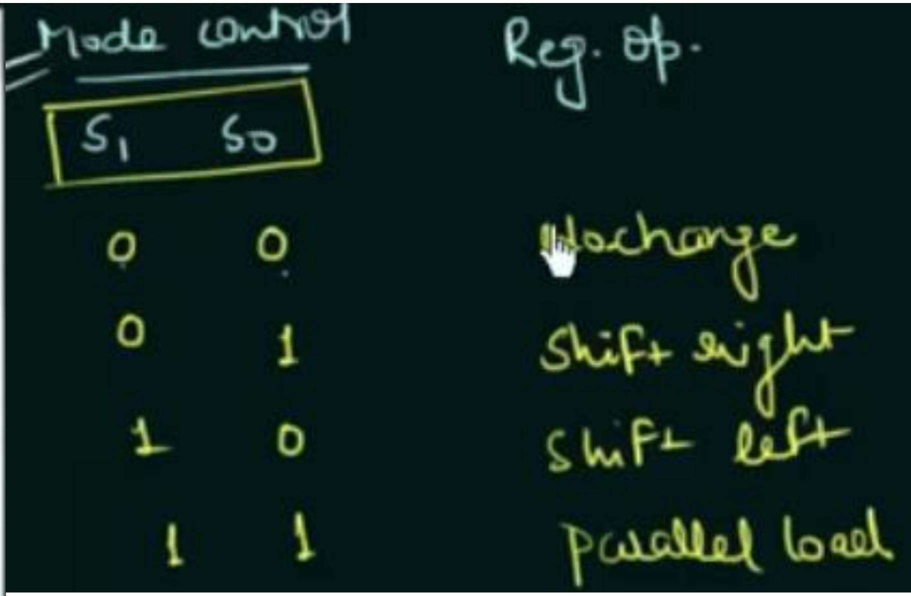


Selection Lines		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Circuit Diagram of 4×1 Multiplexer

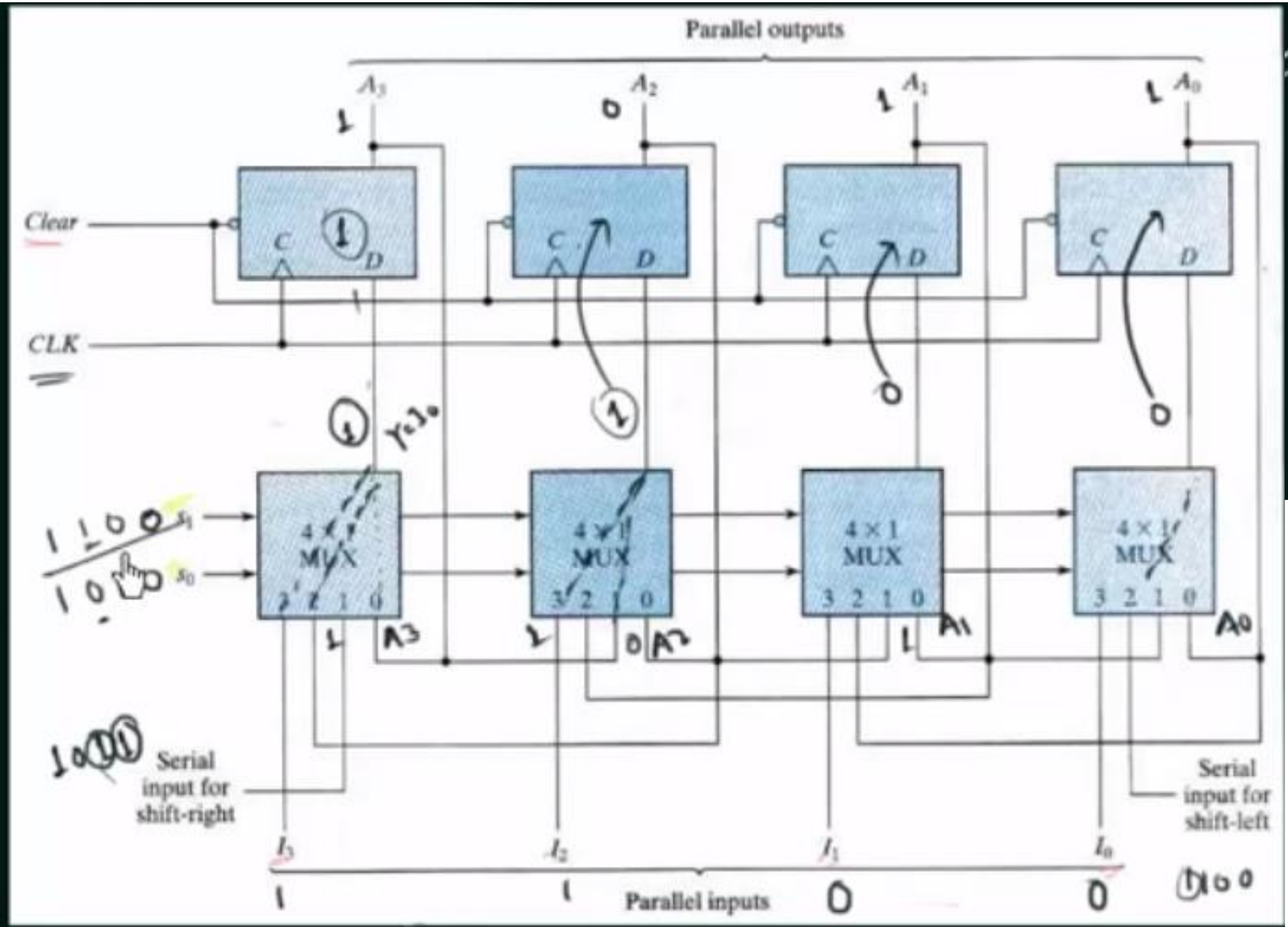
4 bits-Bidirectional Shift Register with parallel load or Universal Shift Register

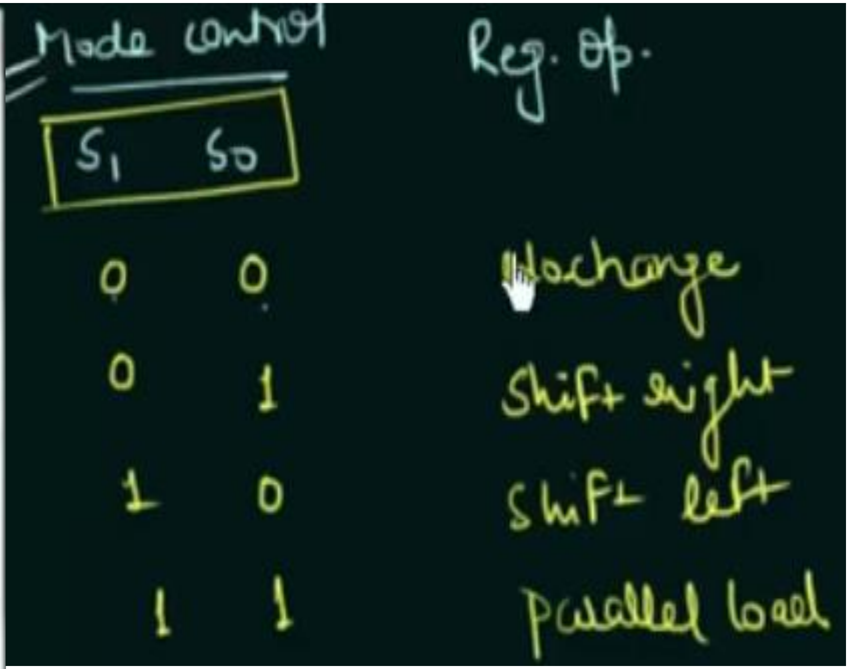




Mode control		Reg. Op.
S_1	S_0	
0	0	Nochange
0	1	Shift right
1	0	Shift left
1	1	parallel load

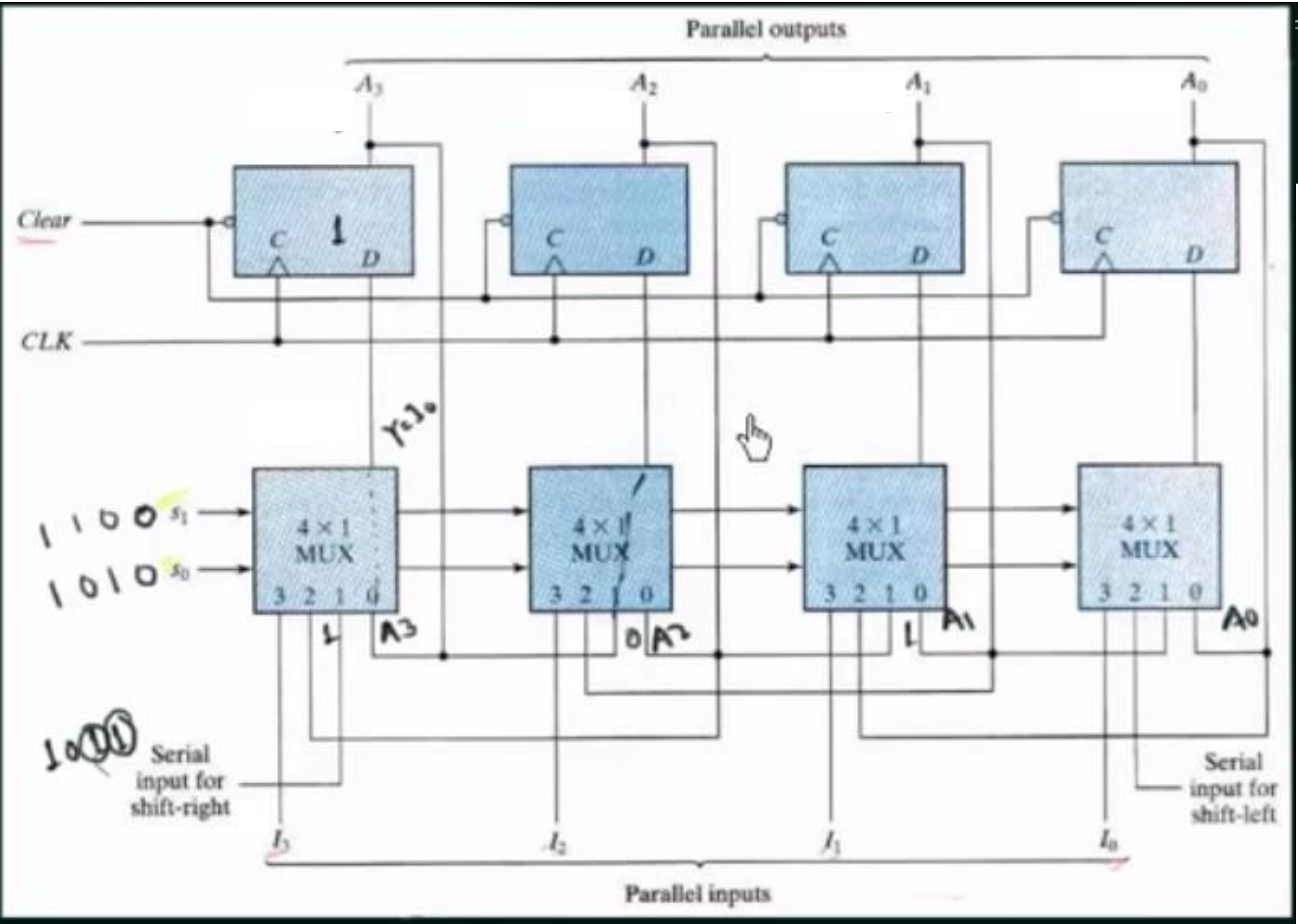
4 bits-Bidirectional Shift Register with parallel load or Universal Shift Register

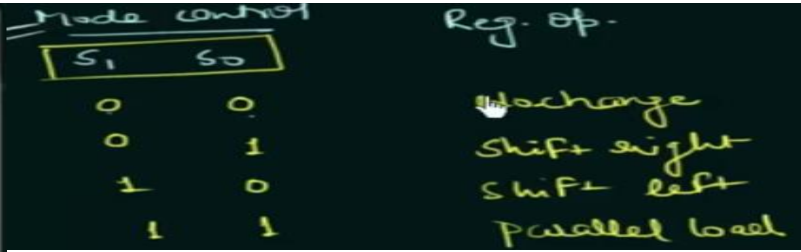




Mode control S_1	S_0	Reg. Op.
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	parallel load

4 bits-Bidirectional Shift Register with parallel load or Universal Shift Register

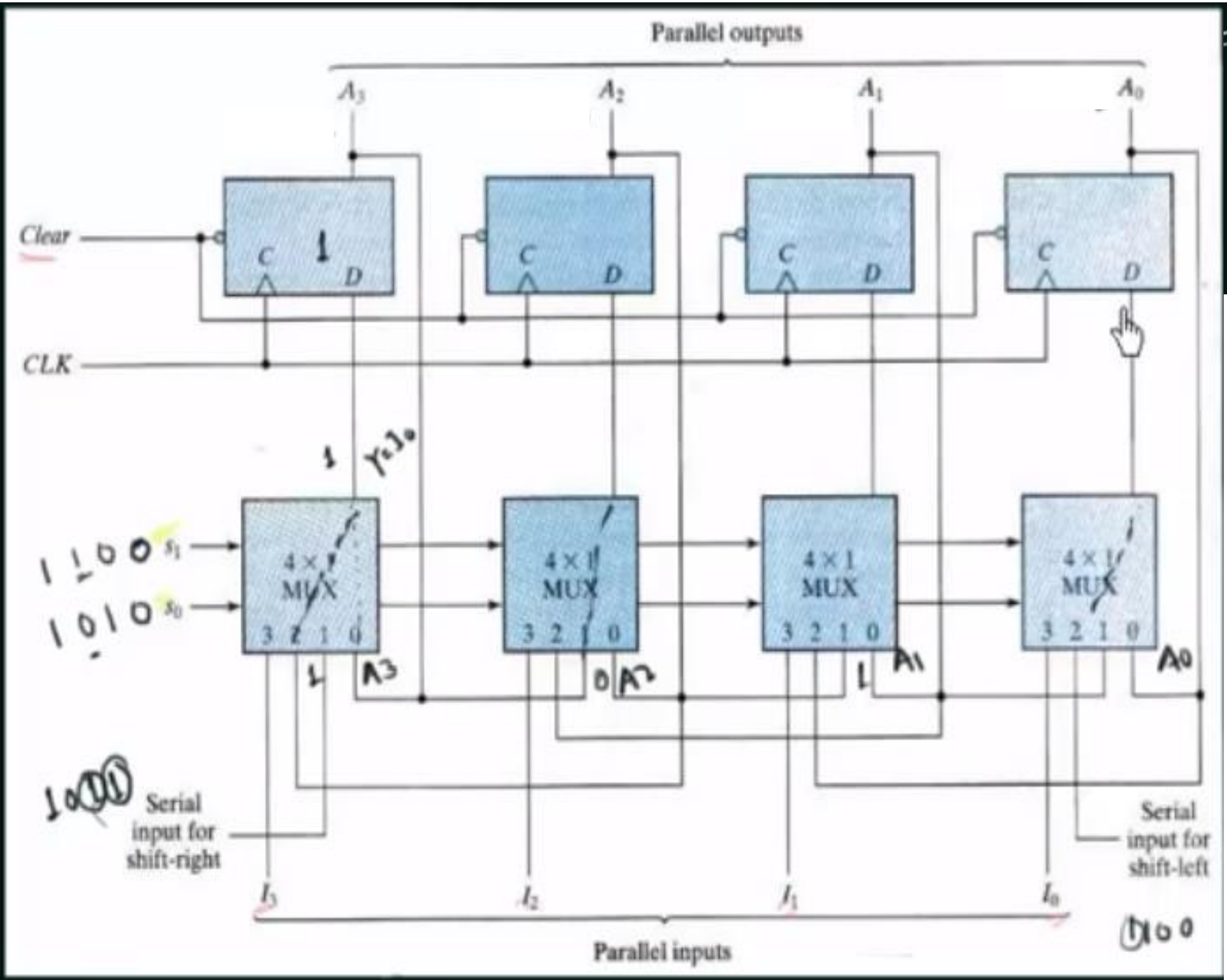


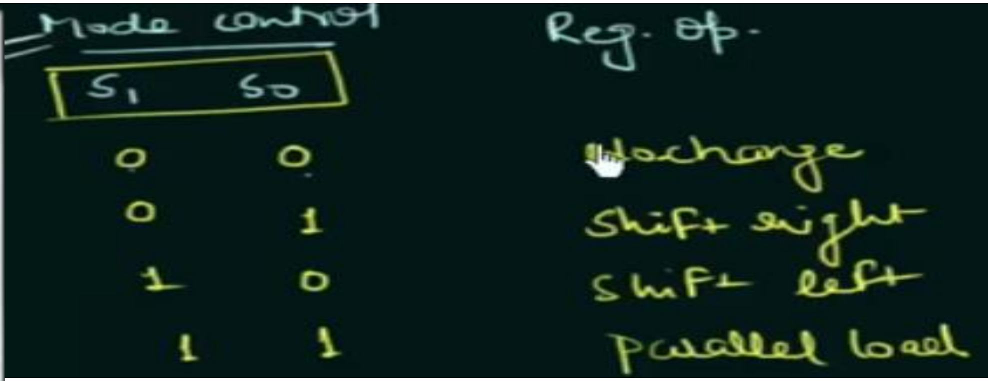


Mode control		Reg. op.
S1	S0	
0	0	Nochange
0	1	Shift right
1	0	Shift left
1	1	Parallel load

A3 A2 A1 A0
0000
1000
1100
0110
1011

4 bits-Bidirectional Shift Register with parallel load or Universal Shift Register





Mode control		Reg. op.
S_1	S_0	
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

A3 A2 A1 A0
0000
0001
0011
0110
1100