Date No

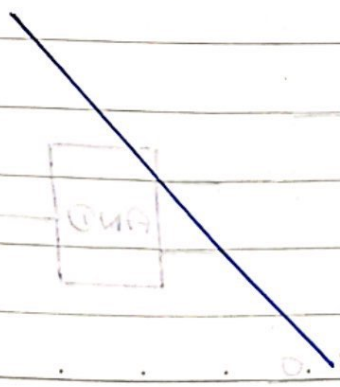
Boolean Algebra

Algebra associated with Binary Numbers is called Boolean Algebra.

Variables used in Boolean Algebra are called Boolean Variables.

The two entities 0 and 1 together with the three operations; AND, OR, NOT is called Boolean Algebra.

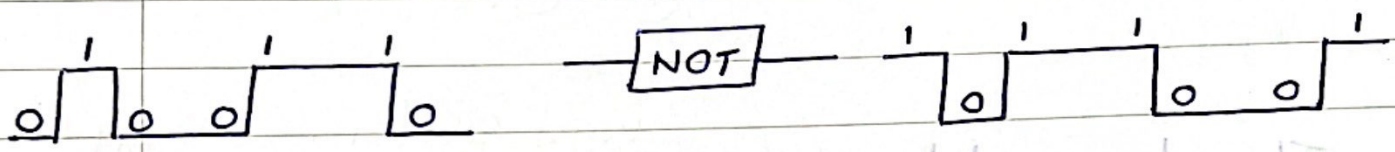
Normal Variable → any value.
Boolean Variable → 1 or 0



Date No

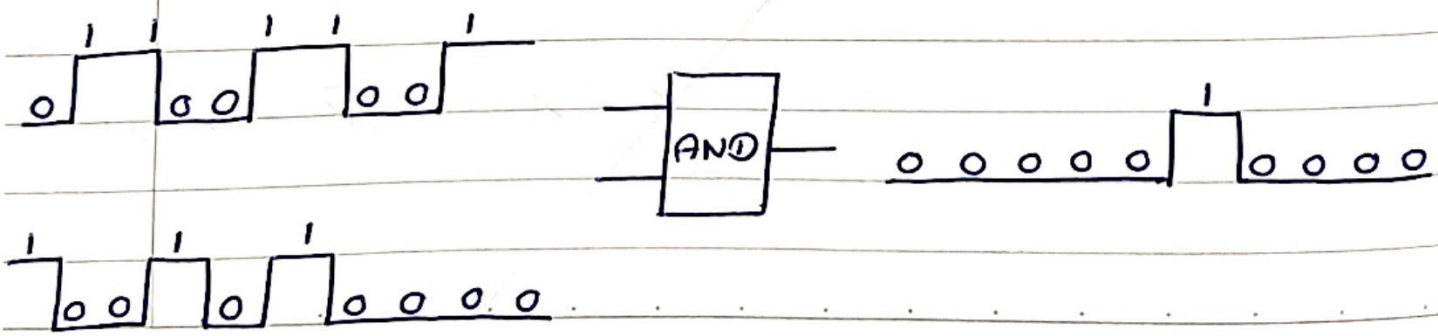
NOT Operation (Complementation)

X	NOT(X) / $\bar{X}$
0	1
1	0



AND Operation (Logical Multiplication)

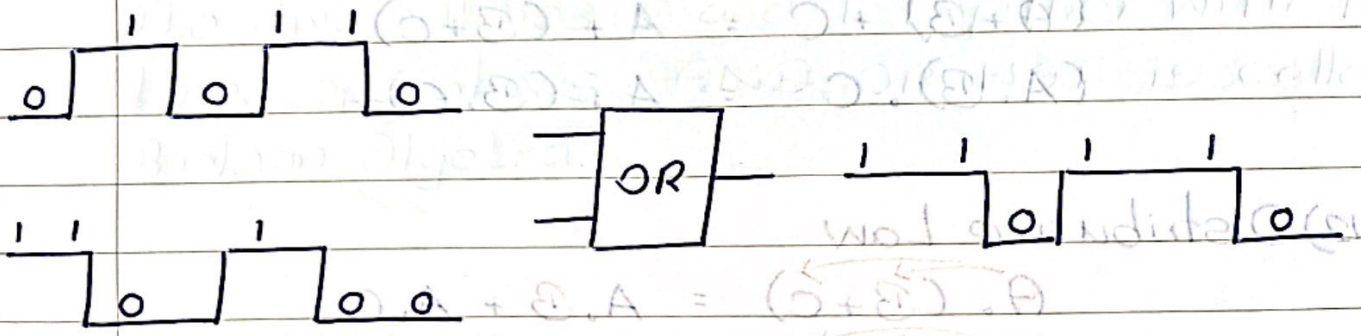
X	Y	X AND Y / (X.Y)
0	0	0
0	1	0
1	0	0
1	1	1



Date No

OR Operation (Logical Addition)

X	Y	X OR Y / (X+Y)
0	0	0
0	1	1
1	0	1
1	1	1



Boolean Algebra Rules

01.) $X + 0 = X$

02.) $X + 1 = X$

03.) $X . 0 = 0$

04.) $X . 1 = X$

05.) $X + X = X$

06.) $X . X = X$

Date No

07.) $X + \bar{X} = 1$

08.) $X \cdot \bar{X} = 0$

$\overline{(\bar{X})} = X$

10.) Commutative Law

$A + B = B + A$

$A \cdot B = B \cdot A$

11.) Associative Law

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

$(A \cdot B) \cdot C = A \cdot (B \cdot C)$

* 12.) Distributive Law

$A \cdot (B + C) = A \cdot B + A \cdot C$

$A + (B \cdot C) = (A + B)(A + C)$

13.) Redundance Law

$A + A \cdot B = A$

$A (A + B) = A$

14.) $X + \bar{X}Y = X + Y$

Date No

Prove, $A + A.B = A$

Date No

Prove, $A(A+B) = A$

Date No

Perfect Induction

Perfect induction employs a truth table, which describes the validity of the Boolean entity for all the possible value combinations of the Boolean Variables.

Prove the following using Perfect Induction.

01.) $\underbrace{X + XY}_{L.H.S} = \underbrace{X}_{R.H.S}$

X	Y	XY	X+XY
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

↑ R.H.S ↑ L.H.S

*no of variables $= 2^n$

$\therefore R.H.S = L.H.S$ //

Date No

02) $\underbrace{X(X+Y)}_{L.H.S} = \underbrace{X}_{R.H.S}$

X	Y	X + Y	X (X + Y)
0	0	0	0
0	1	1	0
1	0	1	1
1	1	1	1

↑ R.H.S

↑ L.H.S

$\therefore$ R.H.S = L.H.S

Prove the above using Boolean Variable.

01) $X + XY = X$

$L.H.S = X + XY$

$= X \underbrace{(1+Y)}_{1}$

$= X$

$= R.H.S$

Date No

02.) $X(X+Y) = X$

$$\begin{aligned} L.H.S &= X(X+Y) \\ &= X.X + X.Y \\ &= X + X.Y \\ &= X + \underbrace{(1+Y)}_{1} \\ &= X + 1 \\ &= X \\ &= R.H.S \end{aligned}$$

Using Boolean Algebra, simplify the following expressions.

01.) $\Theta = \bar{X}Y\bar{Z} + \bar{X}YZ + X\bar{Y}Z + XYZ$

$$\begin{aligned} &= \bar{X}Y\underbrace{(\bar{Z}+Z)}_{1} + XZ\underbrace{(\bar{Y}+Y)}_{1} \\ &= \bar{X}Y + XZ \end{aligned}$$

Date No

① De Morgan's Theorem.

I.) $\overline{X + Y} = \overline{X} \cdot \overline{Y}$

II.) $\overline{X \cdot Y} = \overline{X} + \overline{Y}$

Apply ① De Morgan's Theorem to the Following expressions.

01.) $Z = \overline{A + \overline{B}}$

$= \overline{A} \cdot B$

02.) $Z = \overline{AB + CD}$

$= \overline{AB} \cdot \overline{CD}$

03.) $Z = \overline{\overline{AB} \cdot \overline{CD}}$

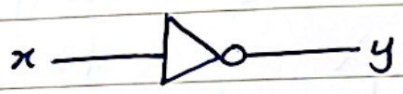
$= \overline{AB} + \overline{CD}$

$= AB + CD$

Date No

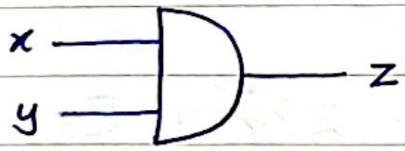
Basic Logic Gates.

(01.) NOT Gate (Inverter)



x	y
0	1
1	0

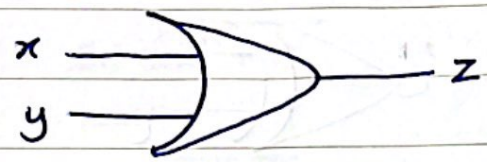
(02.) AND Gate (Logical Multiplication)



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

Date No

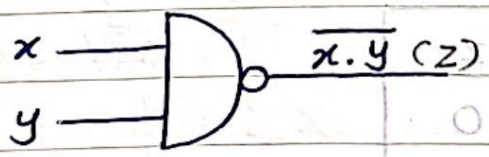
(03.) OR Gate (Addition)



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

Other Logic Gates

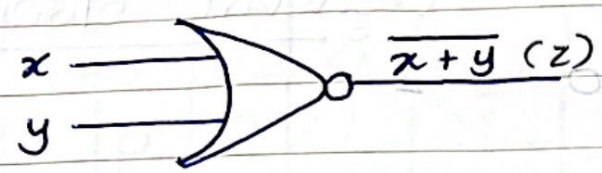
(01.) NAND Gate (Not AND)



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

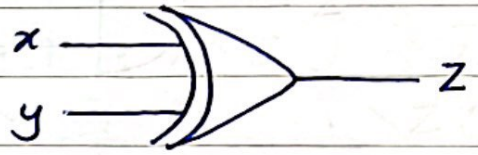
Date No

(02) NOR Gate (Not OR)



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

(03) XOR Gate (exclusive OR gate) ⊕

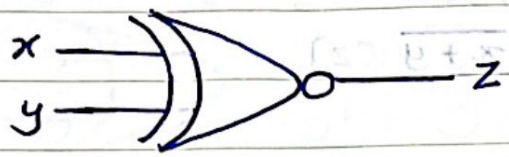


$$Z = \bar{X}Y + X\bar{Y}$$
$$= X \oplus Y$$

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

Date No

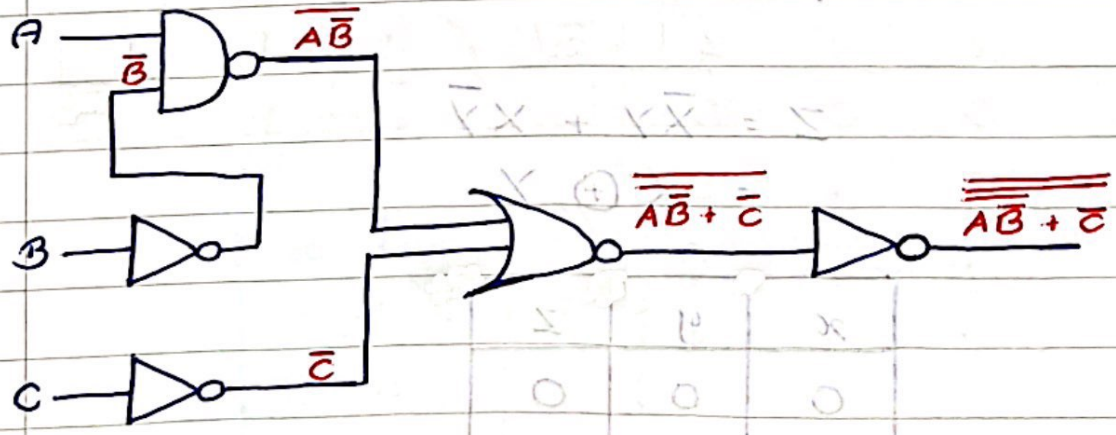
(04.) XNOR Gate (exclusive NOR gate)



$$Z = \bar{X}\bar{Y} + XY$$
$$= \overline{X \oplus Y}$$

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

Example:

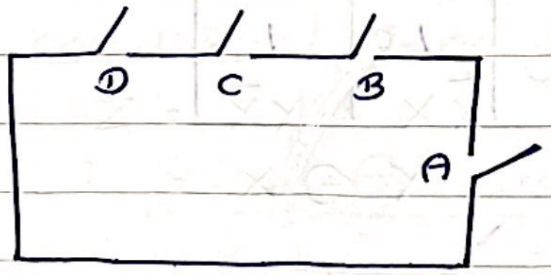


Date No

A	B	C	$\overline{B}$	$\overline{C}$	$\overline{A\overline{B}}$	$\overline{\overline{A\overline{B}} + \overline{C}}$	$\overline{\overline{A\overline{B}} + \overline{C}}$
0	0	0	1	1	1	0	1
0	0	1	1	0	1	0	1
0	1	0	0	1	1	0	1
0	1	1	0	0	1	0	1
1	0	0	1	1	0	0	1
1	0	1	1	0	0	1	0
1	1	0	0	1	1	0	1
1	1	1	0	0	1	0	1

Logic Circuit Design

Example:



- If any 2 doors are open, bulb must turn ON.
- If door A is open, without considering other doors, bulb must tum ON.

DOOR OPEN = 1

DOOR CLOSE = 0

Date No

A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Date No

Steps for Design a Logic Circuit.

01.) Construct a truth table, which describes the input / output relationship by the given specifications.

02.) Derive an algebraric expression for each of the outputs.

a) SOP method.

b) POS method.

03.) Reduce the output expression as much as possible.

a) Algebraically.

b) Using K Maps.

c) Using Tabular Method.

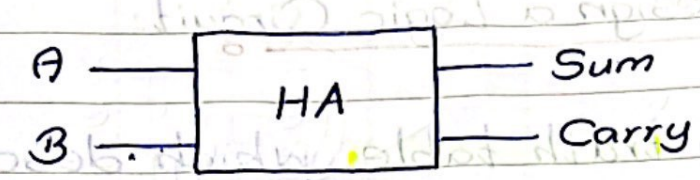
04.) Implement the circuit using gates.

Binary Adders.

Binary Half Adder.

It is a logic circuit for the addition of two 1-bit binary numbers.

Date No



Inputs

Outputs

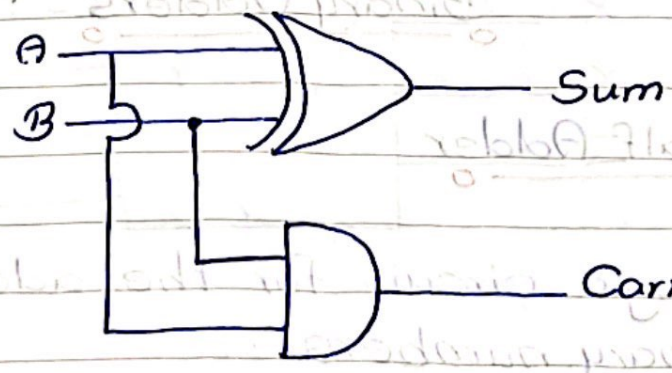
		2^1	2^0	
A	B	Carry	Sum	
0	0	0	0	0+0 → 0 → 00
0	1	0	1	0+1 → 1 → 01
1	0	0	1	1+0 → 1 → 01
1	1	1	0	1+1 → 2 → 10

Output Combinations

Input Combinations

$$Sum = \bar{A}B + A\bar{B}$$
$$= A \oplus B$$

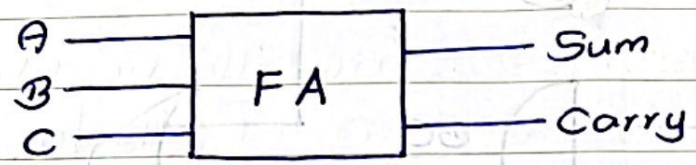
$$Carry = AB$$



Date No

Binary Full Adder

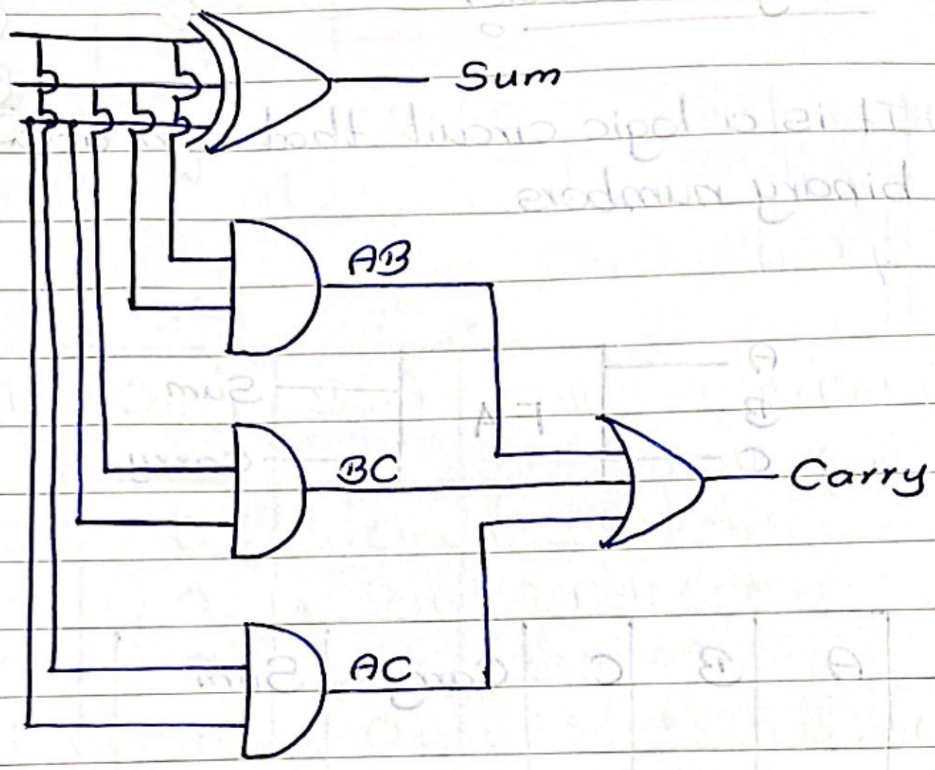
It is a logic circuit that can add three, 1-bit binary numbers.



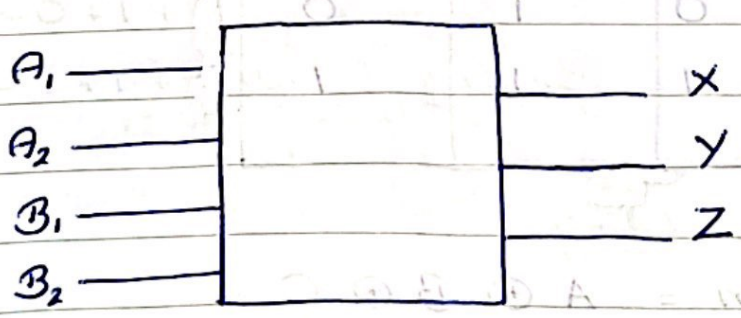
A	B	C	Carry	Sum	
0	0	0	0	0	0+0+0 → 0 → 00
0	0	1	0	1	0+0+1 → 1 → 01
0	1	0	0	1	0+1+0 → 1 → 01
0	1	1	1	0	0+1+1 → 2 → 10
1	0	0	0	1	1+0+0 → 1 → 01
1	0	1	1	0	1+0+1 → 2 → 10
1	1	0	1	0	1+1+0 → 2 → 10
1	1	1	1	1	1+1+1 → 3 → 11

$$Sum = A \oplus B \oplus C$$

$$Carry = AB + BC + AC$$



① Design a logic circuit to add two, 2-bit binary numbers. Implement the circuit using logic gates.



Date No

A_1	A_2	B_1	B_2	X	Y	Z
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

Date No

Z	Y	X	D_3	D_2	D_1	D_0
			0	0	0	0
			1	0	0	0
			0	1	0	0
			1	1	0	0
			0	0	1	0
			0	0	1	0
			0	1	1	0
			1	1	1	0
			0	0	0	1
			1	0	0	1
			0	1	0	1
			1	1	0	1
			0	0	1	1
			1	0	1	1
			0	1	1	1
			1	1	1	1

Draw the Circuit

ProMate

Date No

(01.) In a digital system, an error comparator is required which will compare two, 2-bit binary numbers, A, B and gives seperate outputs for the conditions A = B, A > B & A < B. Design a logic circuit to perform the above functions.

A_1	A_2		B_1	B_2	A = B	A > B	A < B
0	0	=	0	0	1	0	0
0	0	<	0	1	0	0	1
0	0	<	1	0	0	0	1
0	0	<	1	1	0	0	1
0	1	>	0	0	0	1	0
0	1	=	0	1	1	0	0
0	1	<	1	0	0	0	1
0	1	<	1	1	0	0	1
1	0	>	0	0	0	1	0
1	0	>	0	1	0	1	0
1	0	=	1	0	1	0	0
1	0	<	1	1	0	0	1
1	1	>	0	0	0	1	0
1	1	>	0	1	0	1	0
1	1	>	1	0	0	1	0
1	1	=	1	1	1	0	0

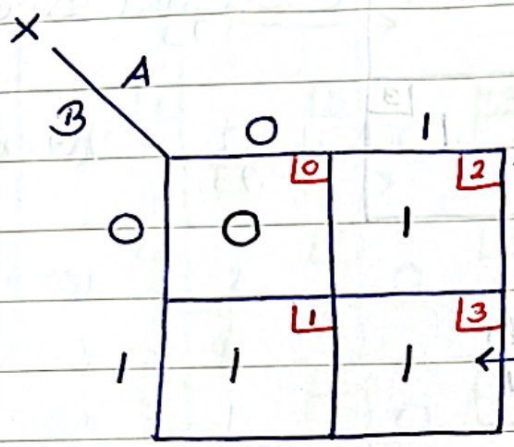
Date No

Draw the Circuit

Date No

Karnaugh Maps (K-Maps)

A boolean expression can be represented graphically using a K-Map



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

* Cells = 2^n (n = variables)

How to Find the Value.

Fill

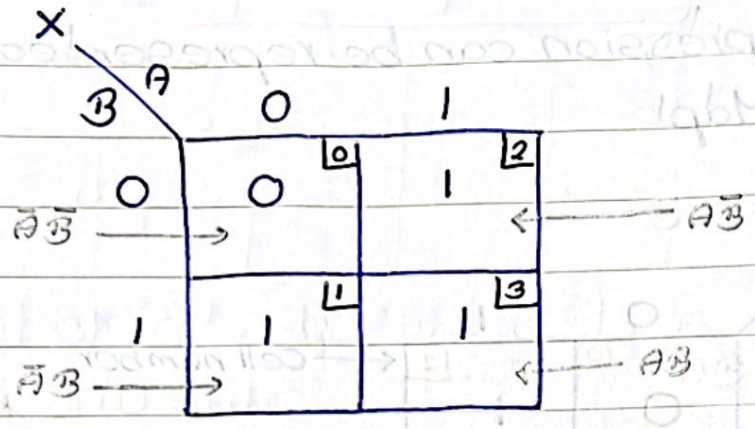
How to Find the Cell Number.

$0_A + 0_B \rightarrow 00 \rightarrow 0$
$0_A + 1_B \rightarrow 01 \rightarrow 1$
$1_A + 0_B \rightarrow 10 \rightarrow 2$
$1_A + 1_B \rightarrow 11 \rightarrow 3$

* Only one variable changing at a time.

Date No

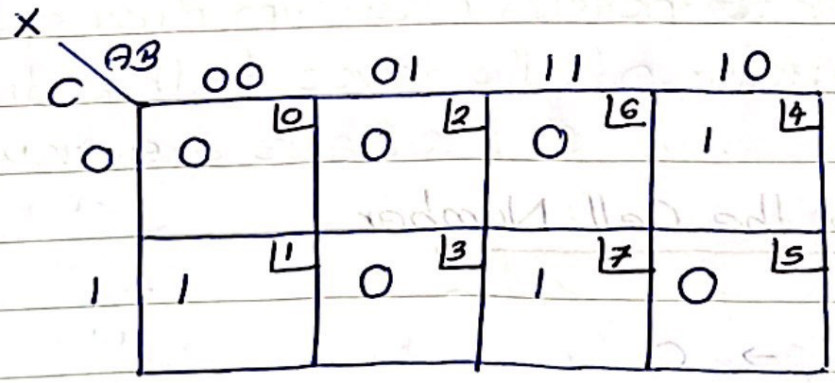
2 Variables.



$$\underset{1\,1}{A/B} + \underset{0\,1}{\bar{A}/B} + \underset{0\,1}{\bar{B}/A}$$

Put 1 for these and 0 for all others.

3 Variables.



* 11 comes before 10.

$$\underset{1\,0\,0}{A\bar{B}/\bar{C}} + \underset{0\,0\,1}{\bar{A}\bar{B}/C} + \underset{1\,1\,1}{AB/C}$$

Date No

4 Variables.

CD \ AB	00	01	11	10
00	0 [0]	0 [4]	1 [12]	0 [8]
01	1 [1]	0 [5]	0 [13]	0 [9]
11	0 [3]	1 [7]	0 [15]	0 [11]
10	0 [2]	0 [6]	0 [14]	1 [10]

$\bar{A}\bar{B}\bar{C}D + \bar{A}BCD + AB\bar{C}\bar{D} + A\bar{B}C\bar{D}$

$\quad 00\,01 \qquad 01\,11 \qquad 11\,00 \qquad 10\,10$

Ex:

C \ AB	00	01	11	10
0	0	0	0	1
1	1	1	0	1

A→0 / C→1
වෙන ඒවා

&

A→1 / B→0
වෙන ඒවා

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

$\ 0\,1 \qquad 1\,0$

* හැම variable එකක් ගත්ත වෙලාව value එක ලියන්න.

ProMate

Date No

Simplification of Boolean Expressions Using K-Map Method.

Sub Cube.

A set of 2^n adjacent cells is referred to as a sub cube.

Maximal Sub Cube.

It is the largest possible sub cube for a cell.

Minimal Map.

Map with minimum number of maximal sub cubes that cover all the miniterms of the given expression.

2^n

1 2 4 8 16 32

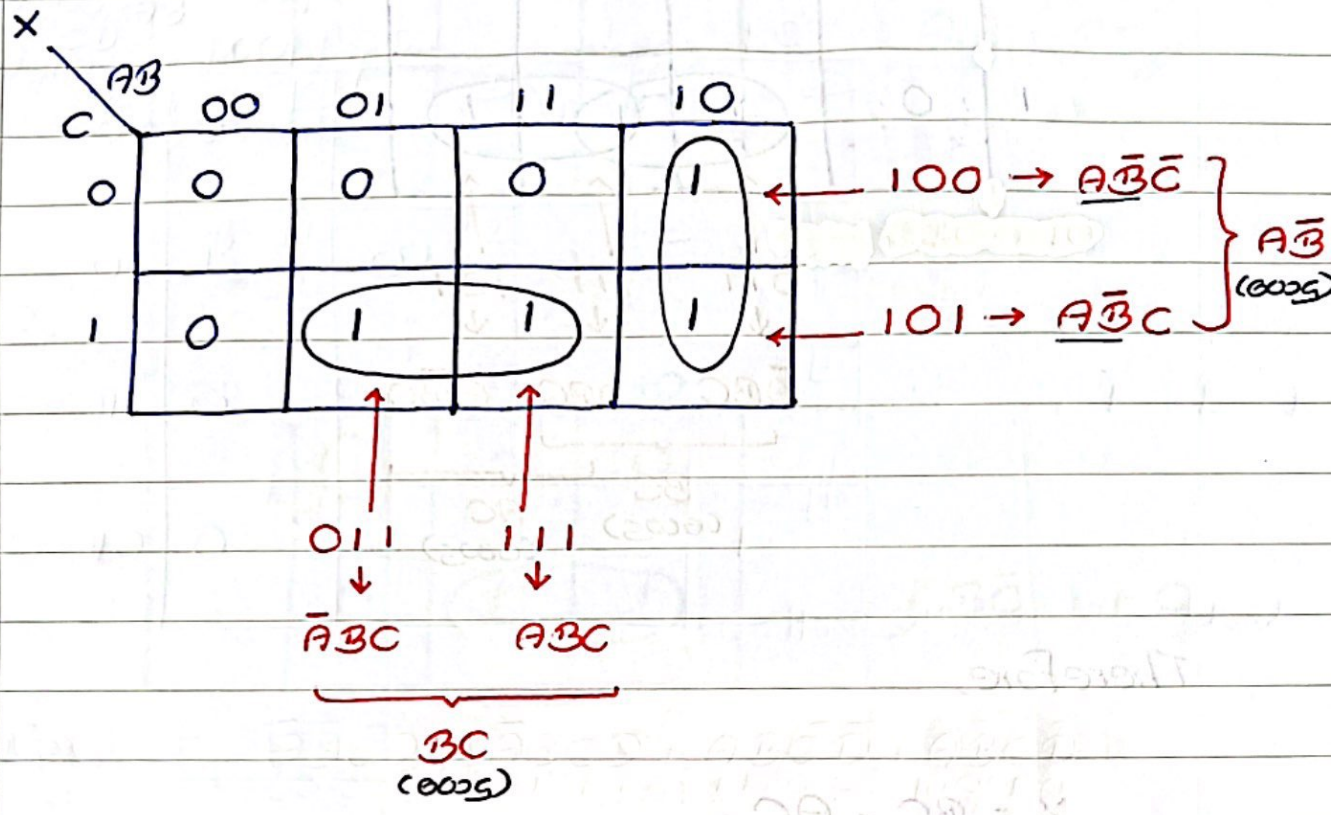
2^0 2^1 2^2 2^3 2^4 2^5

* Go for the maximum.
(If you can go for 4, can't go for 2)

Date No

Ex:

$$X = \underset{011}{\bar{A}B\bar{C}} + \underset{101}{A\bar{B}C} + \underset{100}{A\bar{B}\bar{C}} + \underset{111}{ABC}$$

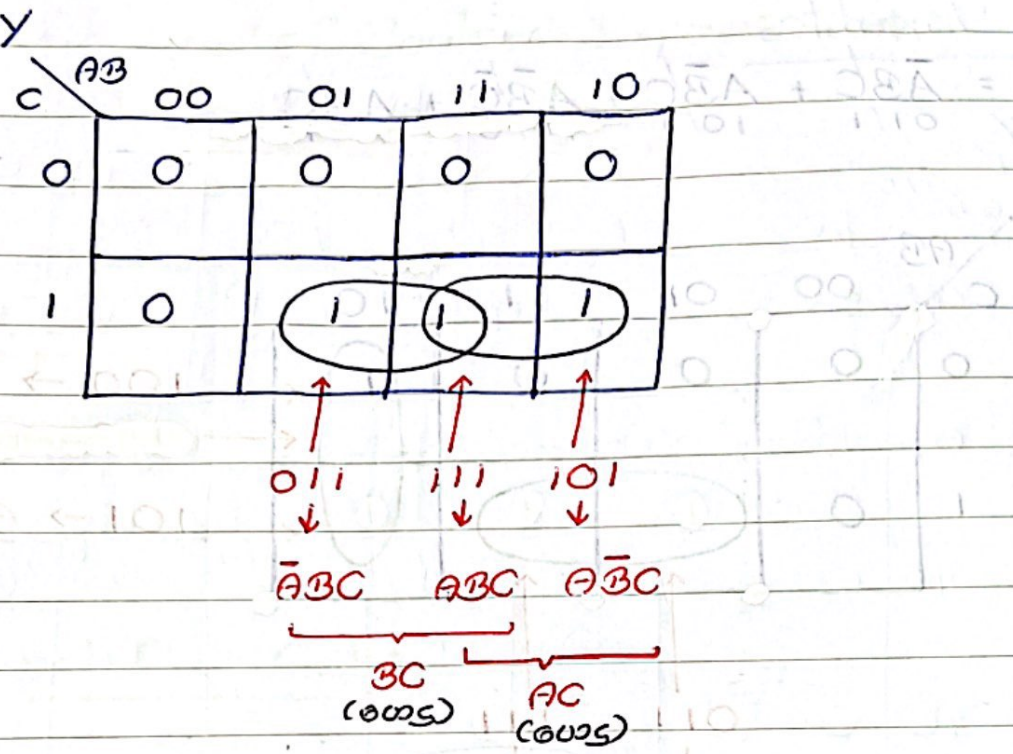


Therefore,

$$X = A\bar{B} + BC \;//$$

Date No

Ex:

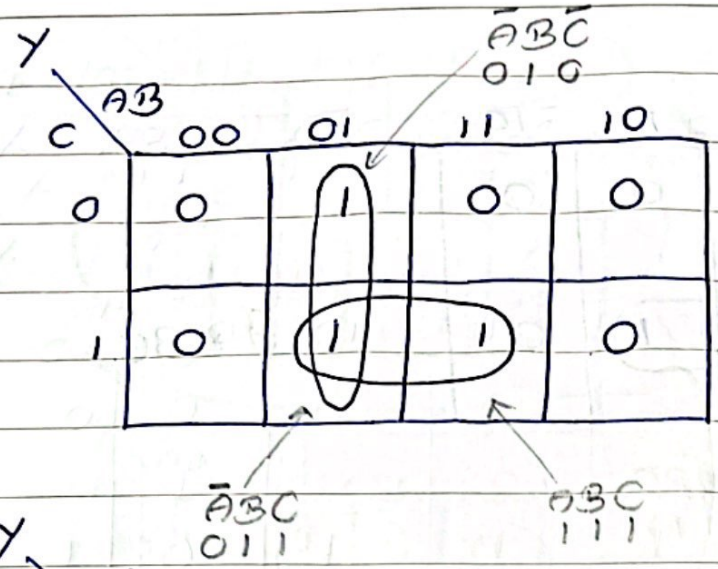


Therefore,

$$X = BC + AC //$$

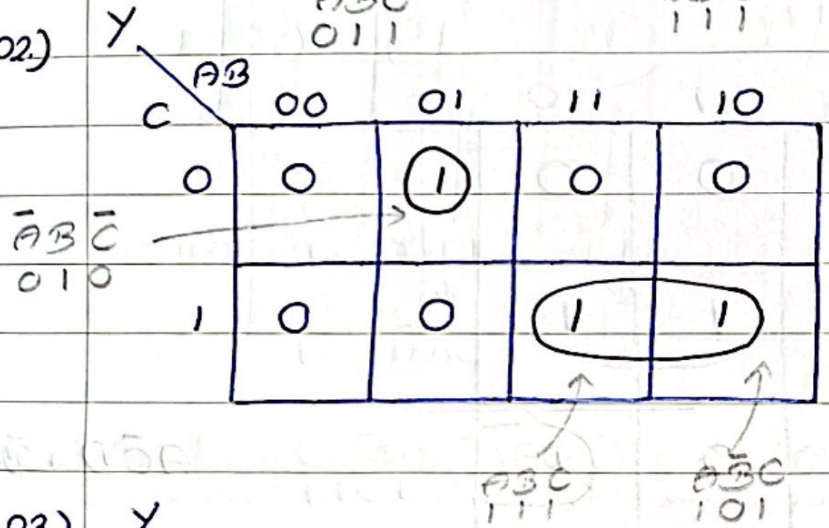
Date No

01.)



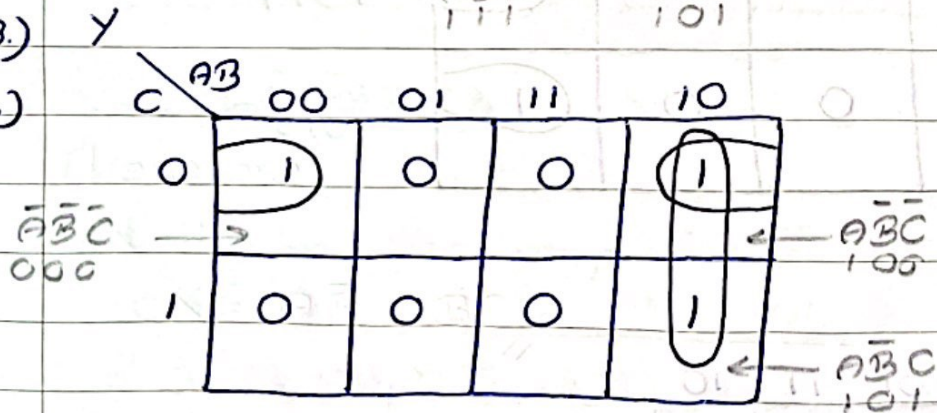
$\bar{A}B + BC$ //

02.)



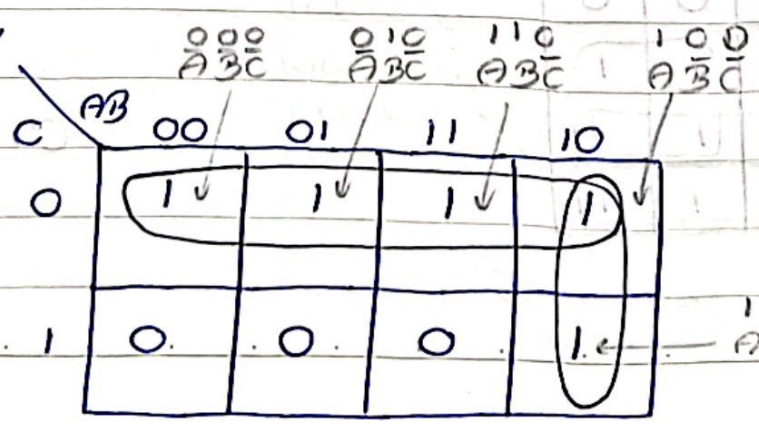
$\bar{A}B\bar{C} + AC$ //

03.)



$B\bar{C} + A\bar{B}$ //

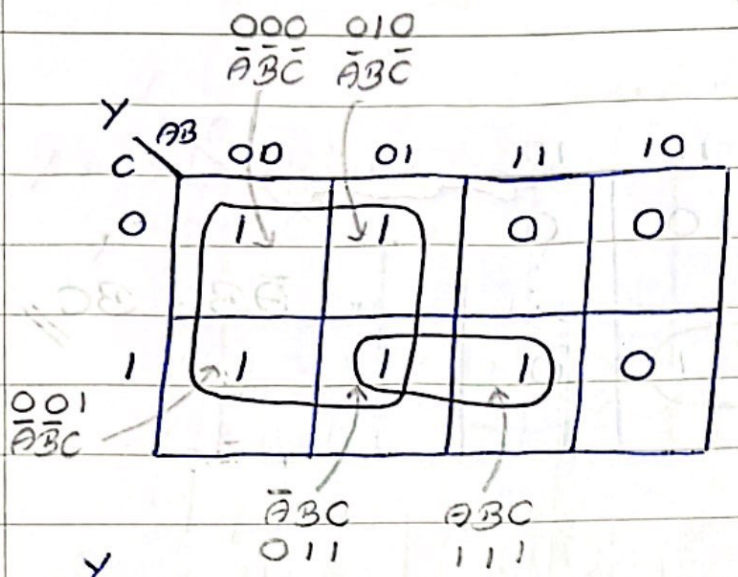
04.)



$\bar{C} + A\bar{B}$ //

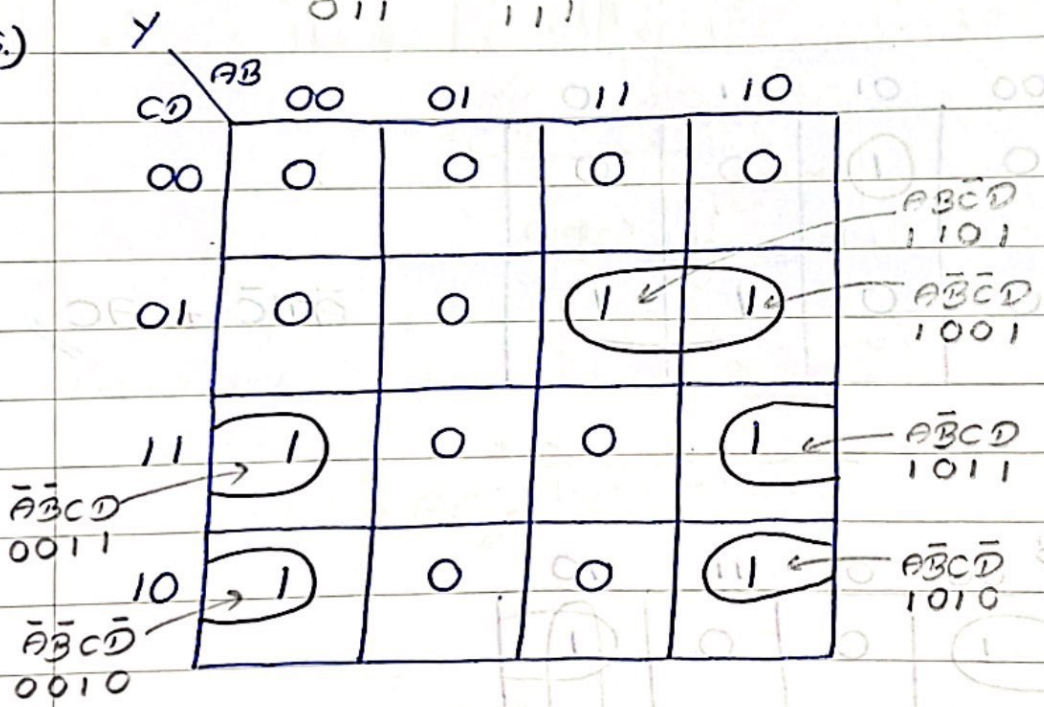
Date No

05.)



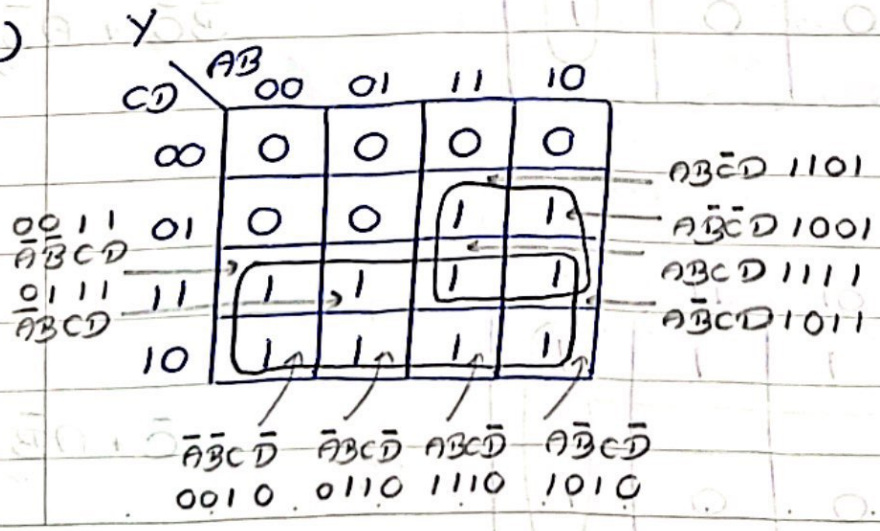
$\bar{A} + BC$ //

06.)



$A\bar{C}D + \bar{B}C$ //

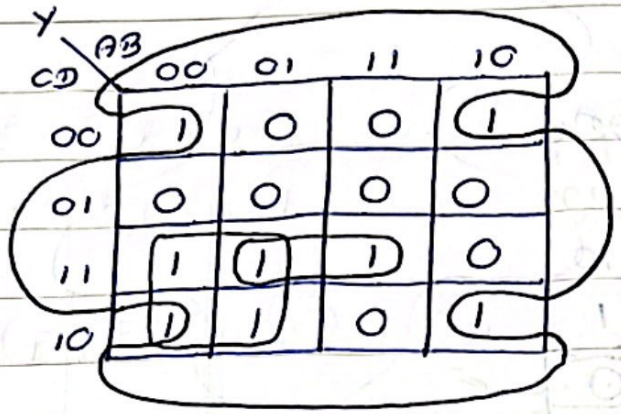
07.)



$C + AD$ //

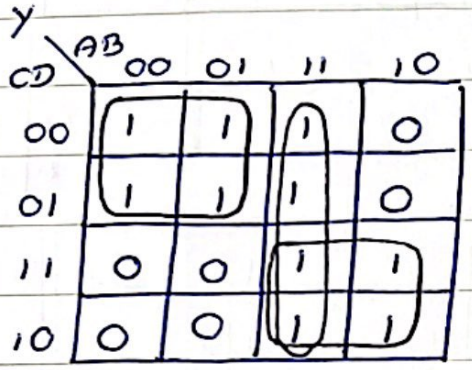
Date No

08.)



$\bar{A}C + BCD + \bar{B}\bar{D}$

09.)



$\bar{A}\bar{C} + AB + AC$

Getting the Simplified POS expression.

01.) Draw the K-Map for X.

02.) Get the minimal map for $\bar{X}$. (Cover all 0's)

03.) Get $\bar{X}$ in SOP form.

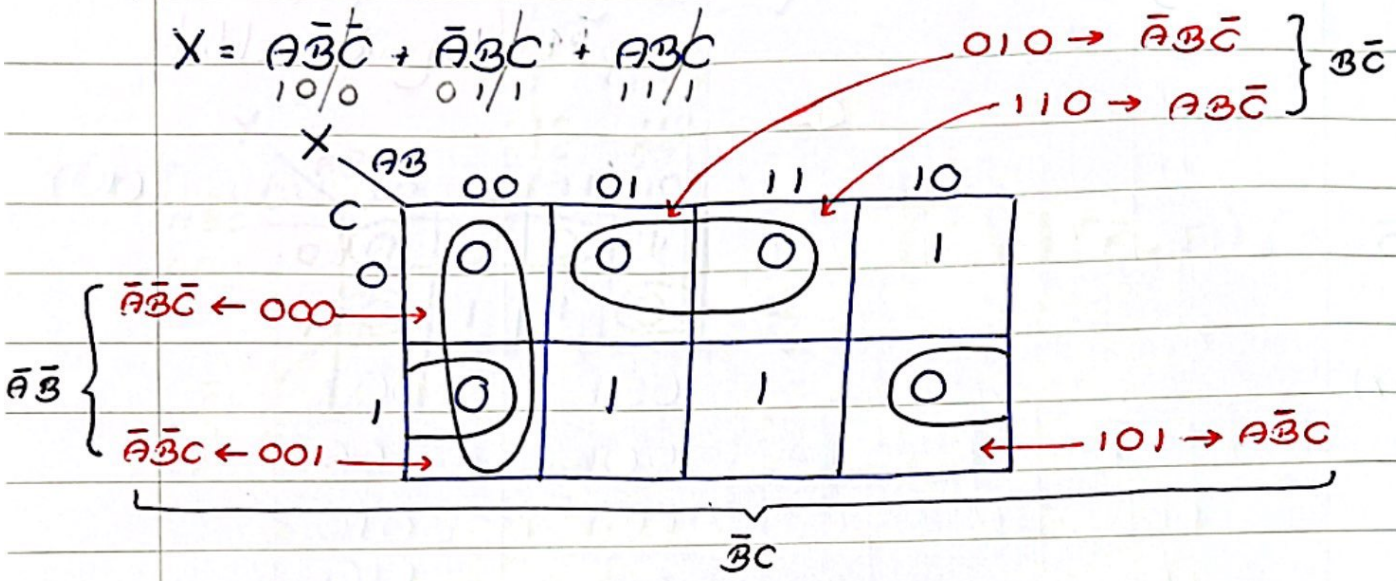
04.) Complement both sides.

Date No

Ex:

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

$$10/0 \quad 01/1 \quad 11/1$$



SOP expression:

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

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

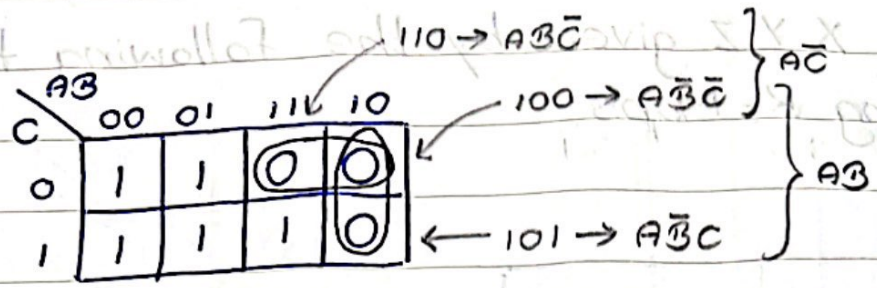
$$X = \overline{\bar{A}\bar{B}}.\overline{B\bar{C}}.\overline{\bar{B}C}$$

POS expression:

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

Date No

(01.)

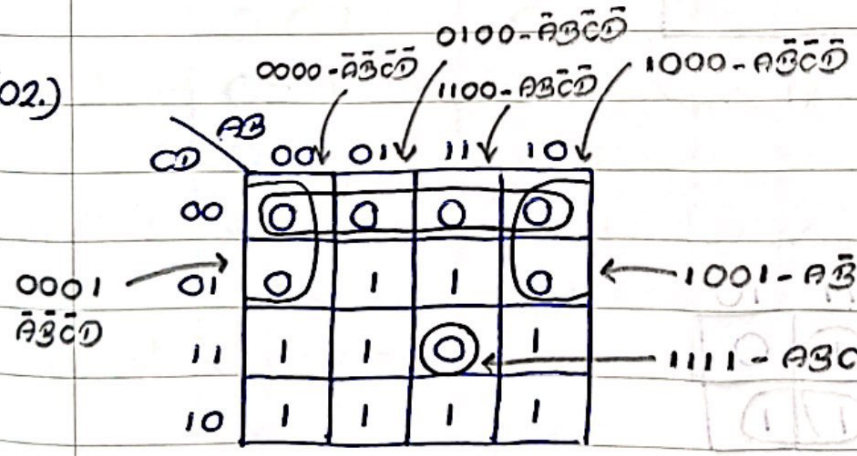


SOP expression;

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

POS expression;

(02.)



SOP expression

POS expression

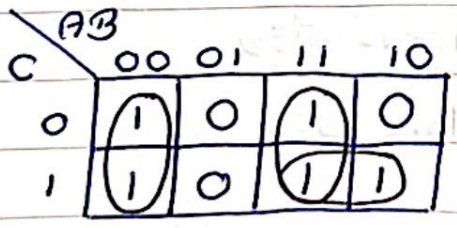
Date No

Simplify X, Y, Z given by the following truth table using K-Maps.

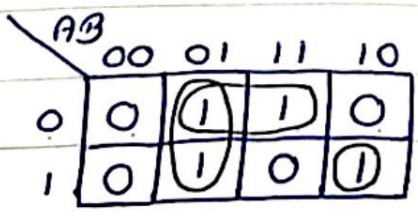
(01.)

ABC	XYZ
000	100
001	100
010	011
011	011
100	001
101	111
110	110
111	101

* X

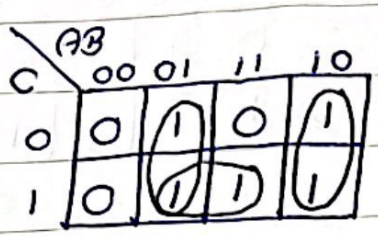


* Y



Date No

* Z

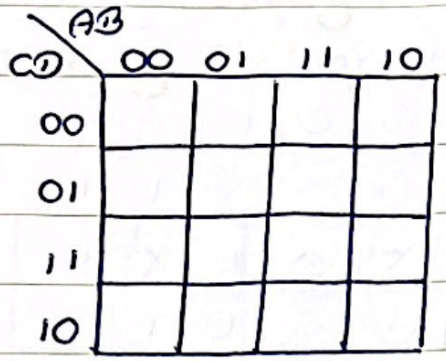


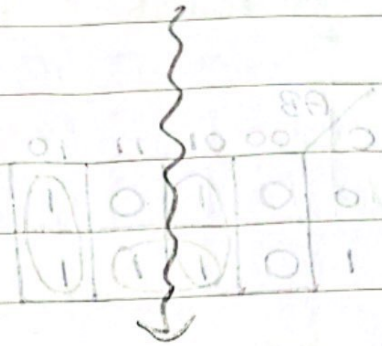
(02.)

ABCD	XYZ
0000	100
0001	001
0010	001
0011	001
0100	010
0101	100
0110	001
0111	001
1000	010
1001	010
1010	100
1011	001
1100	010
1101	010
1110	010
1111	100

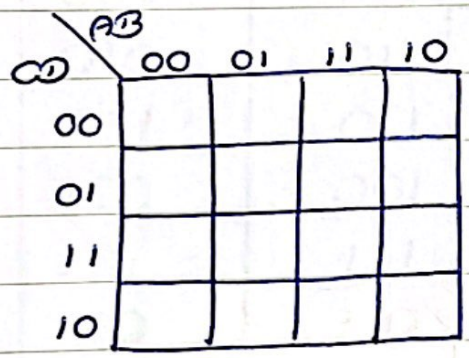
Date No

* X

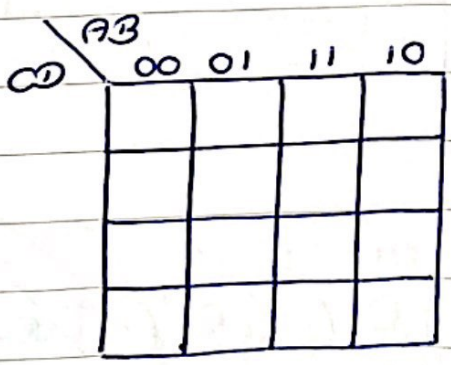




* Y



* Z



Date No

Incompletely Specified Boolean Functions.

(Don't Cares) → X

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	X
1	0	1	X
1	1	0	0
1	1	1	1

} Don't Cares

The value of the function at "don't cares" conditions, is either not significant or those A,B,C values will never occur. Therefore, don't cares can be either included or excluded in a maximal subcube so as to get the minimal map.

included in a maximal sub cube (to make $2^n = 2^2 = 4$)

excluded from a maximal sub cube.

Y	AB 00	01	11	10
C 0	1	X	X	0
1	1	1	0	0

SOP - (1)

Date No

Ex:

Z

CD \ AB	00	01	11	10
00	1	0	0	X
01	1	X	0	0
11	X	0	1	X
10	1	0	0	1

$Z = \bar{A}\bar{B} + \bar{B}C + ACD$

POS - (2)

Ex:

Z

CD \ AB	00	01	11	10
00	1	0	0	X
01	1	X	0	0
11	X	0	1	X
10	1	0	0	1

$(A+\bar{C}).(\bar{B}+\bar{B}).(B+C+D)$

(01.)

CD \ AB	00	01	11	10
00	0	1	X	0
01	0	1	X	0
11	0	0	X	0
10	1	1	X	1

$\bar{Z} = \bar{B}\bar{C} + CD$

$Z = \overline{\bar{B}\bar{C} + CD}$

$= \overline{\bar{B}\bar{C}} \cdot \overline{\bar{C}\bar{D}}$

$= (B+C).(\bar{C}+\bar{D})$

SOP POS Date No

Minterms (Σ) & Maxterms (π)

Ex:

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

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

Y / C \ AB	00	01	11	10
0	0	1	1	1
1	1	0	1	0

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

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

Y / C \ AB	00	01	11	10
0	0	1	0	1
1	1	1	0	1

Date No

Ex: $Z = f(A, B, C, D)$

$Z = 1$ for the miniterms. $(0, 2, 5, 7, 8, 13)$
$Z =$ Don't care for the miniterms. $(10, 15)$
$Z = 0$ for the remaining miniterms.

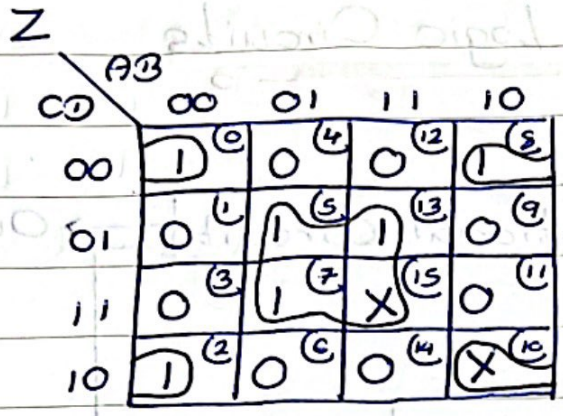
a) Draw the truth table.
b) Simplify Z using K-Map method.
c) Draw the circuit diagram.

a)

	A	B	C	D	Z
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	X
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	X

Date No

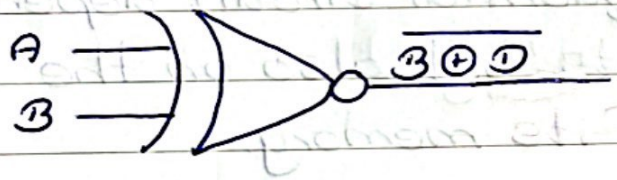
b)



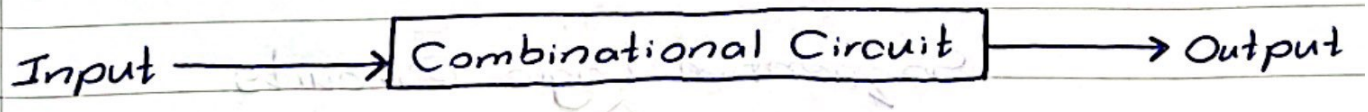
$$Z = \overline{B}\overline{D} + \overline{B}\overline{D}$$

$$= B \oplus D$$

c)



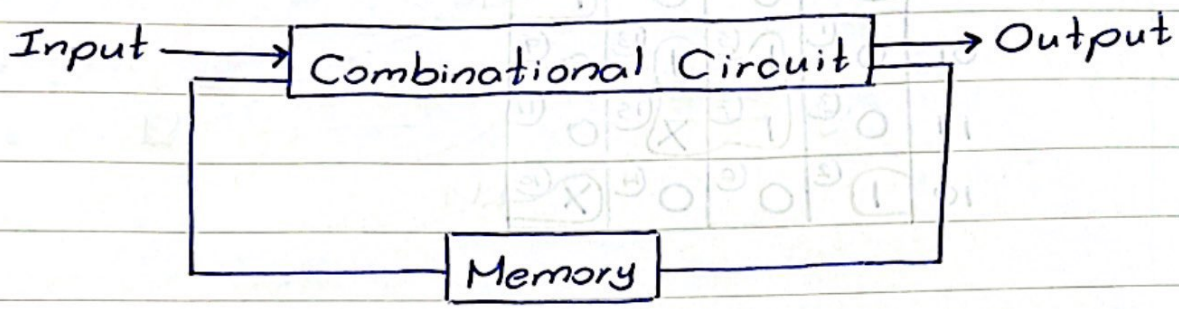
Combinational Circuits



A combinational circuit generally consists of several input signals, several output signals and an interconnection of gates. When applying the same inputs, a combinational circuit always produces the same outputs. i.e every input has a single, unique output.

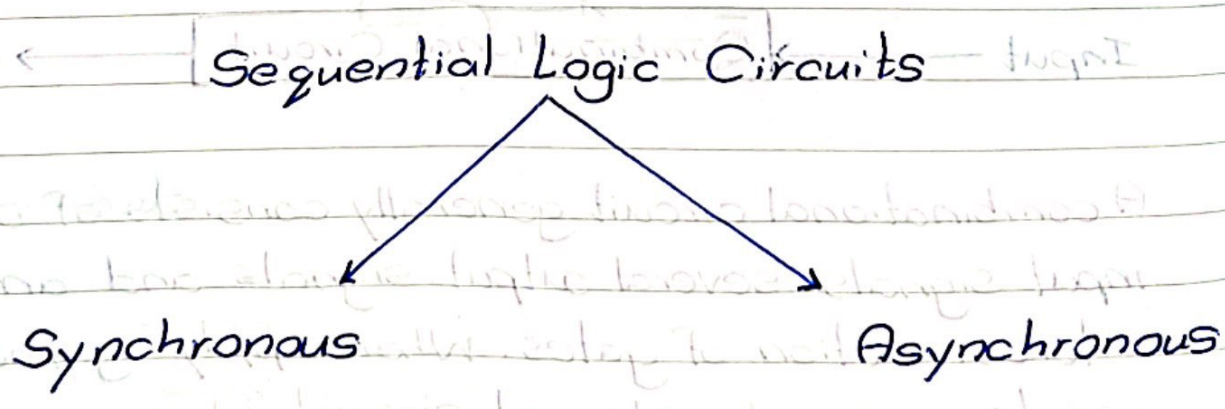
Date No

Sequential Logic Circuits



Sequential circuits are built out of memory elements and combinational logic devices. The outputs of a sequential circuit depends on not only the inputs, but also on the current contents of its memory.

Therefore, these circuits produce different outputs for the same set of inputs depending on what is stored in memory.



Date No

01.) Synchronous Sequential Logic Circuits.

Various blocks of the logic circuit are triggered by the same clock signal, called the master clock. Therefore, the output of these blocks (devices) change at the same time.

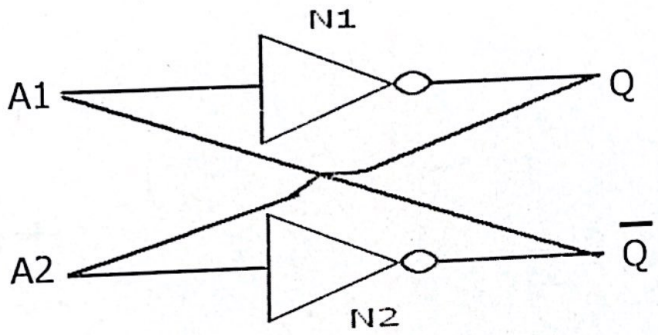
02.) Asynchronous Sequential Logic Circuits.

In this types of circuits, different parts of the circuit are triggered by different signals. Therefore, the outputs of different blocks of the circuit can change at different instances.

Latch

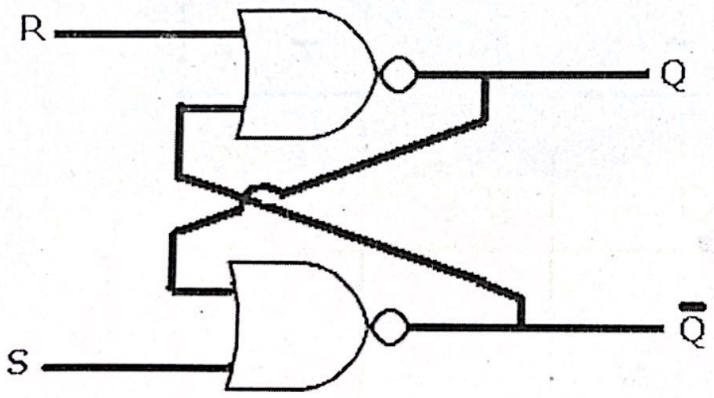
* A latch is a data storage (memory) device that can store one bit of information.

* The basic digital memory circuit is obtained by cross-coupling two NOT gates.



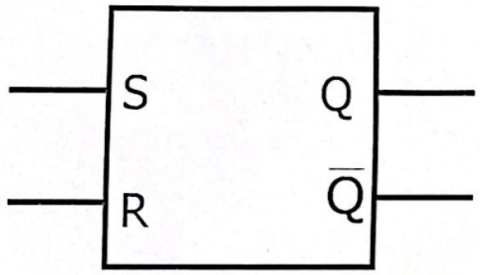
The output of each gate is connected to the input of the other and this feedback combination is called a latch.

SR Latch (Set Reset Latch)



SR Latch (Set Reset Latch)

Symbol:



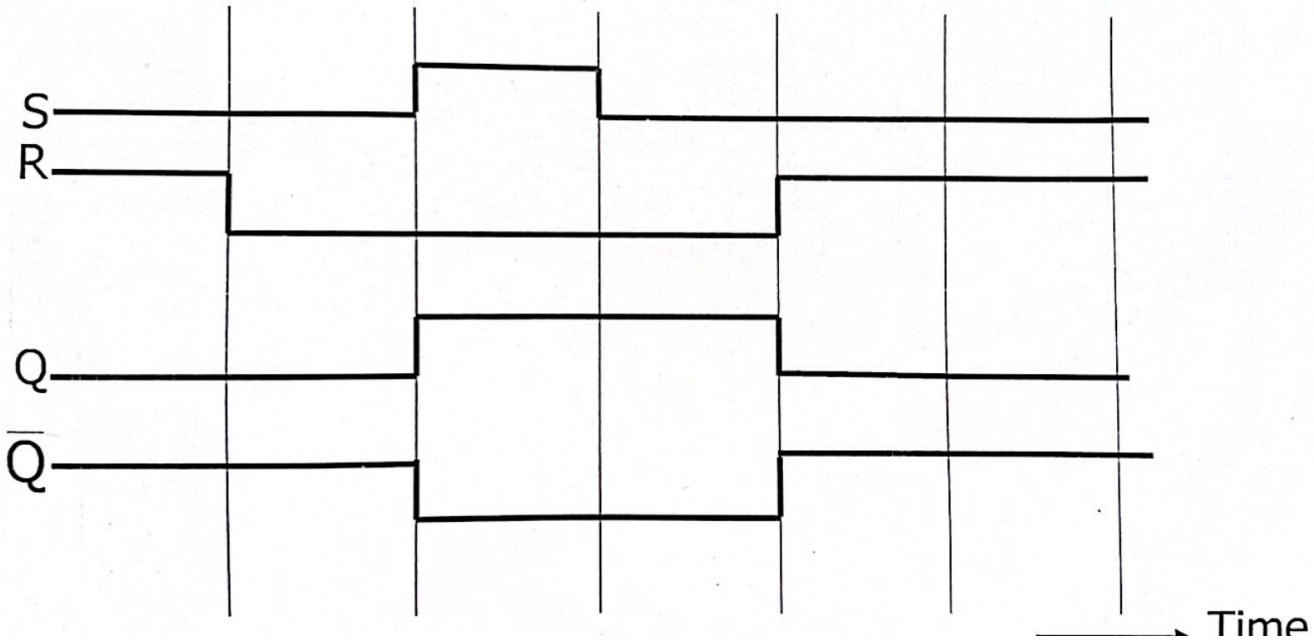
Truth table:

S	R	Qn	$\overline{Q}$n	Qn+1	$\overline{Q_{n+1}}$
0	0	0	1	0	1
		1	0	1	0
0	1	0	1	0	1
		1	0		
1	0	0	1	1	0
		1	0		
1	1	0	1	Not allowed	
		1	0		

Reduced Truth Table:

	S	R	Q_{n+1}
Hold state	0	0	Q_n
Reset state	0	1	0
Set state	1	0	1
Not allowed state	1	1	NA

Draw the timing diagram showing the operation of a SR Latch:

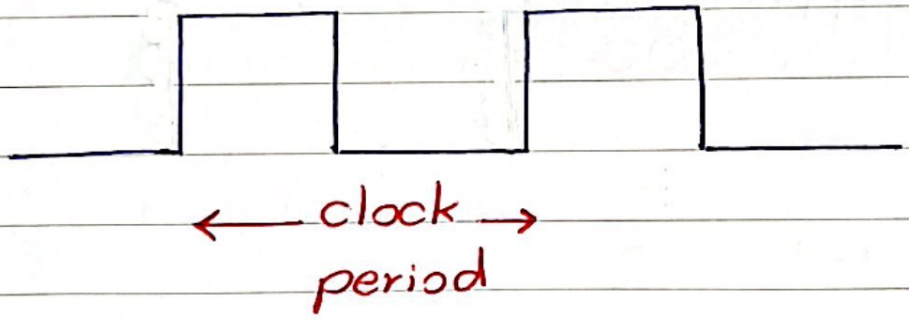


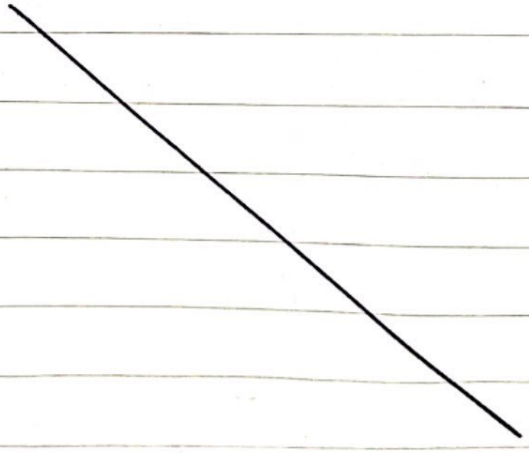
Date No

Flip Flops (Triggered Latch)

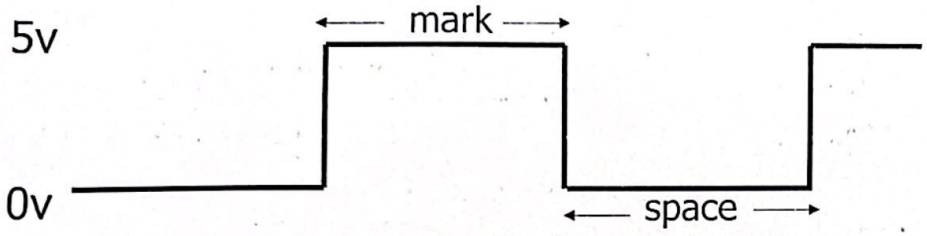
Clock

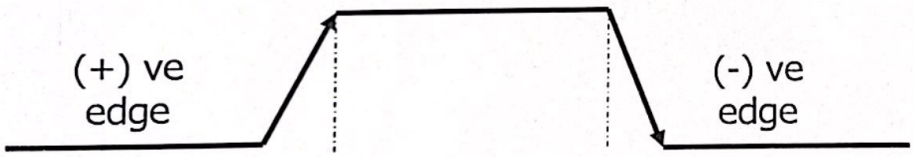
A clock is a special device that continously outputs 0's and 1's. The time it takes for the clock to change from 1 to 0 and back to 1 is called a clock period or clock cycle.



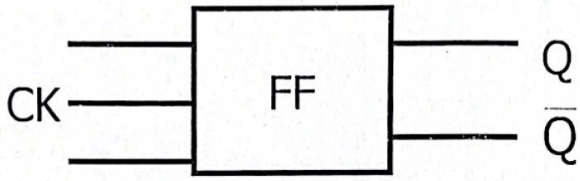


Types of triggering



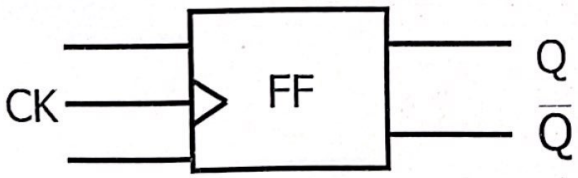


Level triggering:

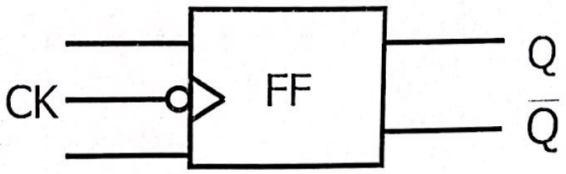


Activated by the mark

Edge triggering:



Activated by the (+)ve edge



Activated by the (-)ve edge

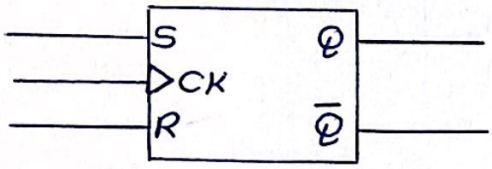
SR (Set Reset) Flip-Flop

Truth table:

CK	S	R	Q_{n+1}
0	0	0	Q_n
0	0	1	Q_n
0	1	0	Q_n
0	1	1	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	NA

When CK is low $Q_{n+1} = Q_n$

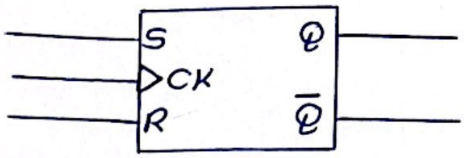
* Symbol:



0	1	0	Qn
0	1	1	Qn
1	0	0	Qn
1	0	1	0
1	1	0	1
1	1	1	NA

low $Q_{n+1} = Q_n$

*Symbol:

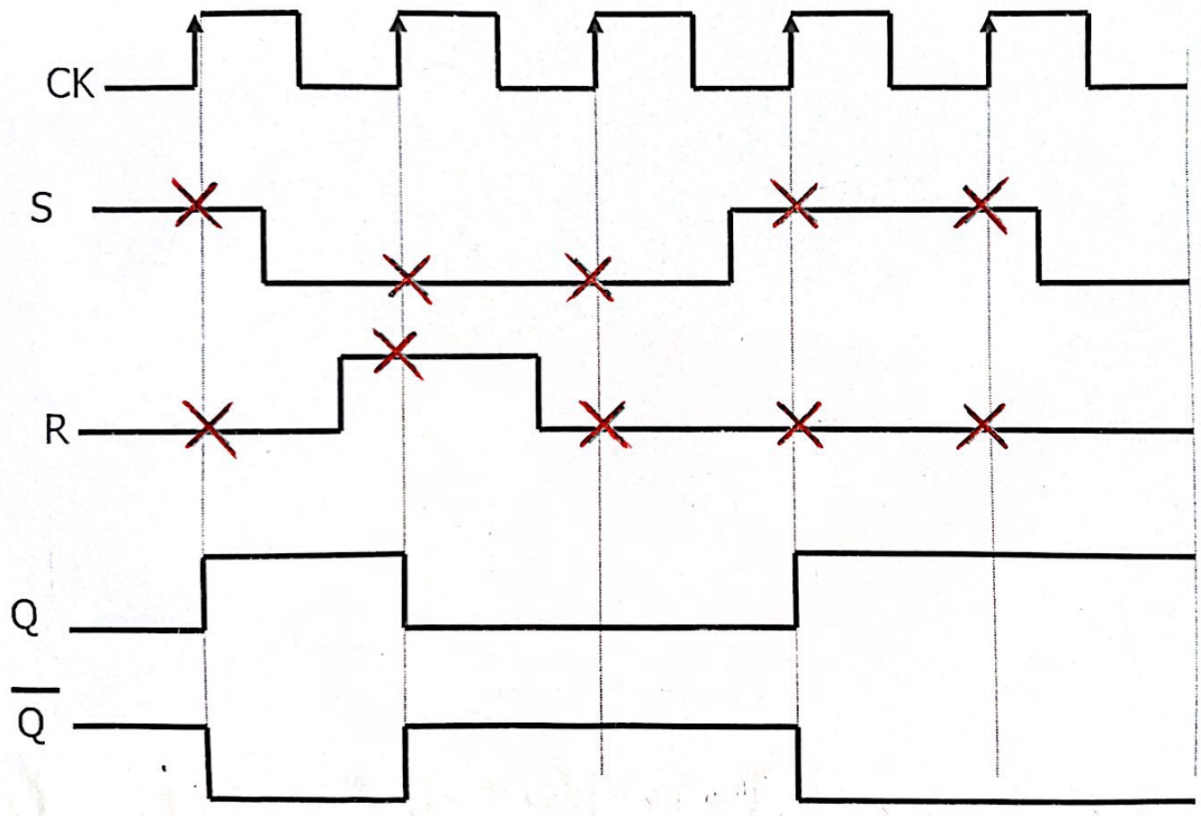


Complete Truth Table of a SR FF

When CK =1

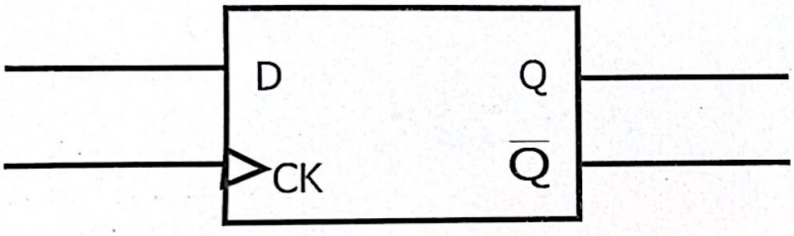
S	R	Qn	Qn+1
0	0	0	0
		1	1
0	1	0	0
		1	0
1	0	0	1
		1	1
1	1	0	NA
		1	

Draw the output waveform for the (+)ve edge triggered SR FF.



D Flip-Flop(Data FF)

Symbol:



Reduced TT:

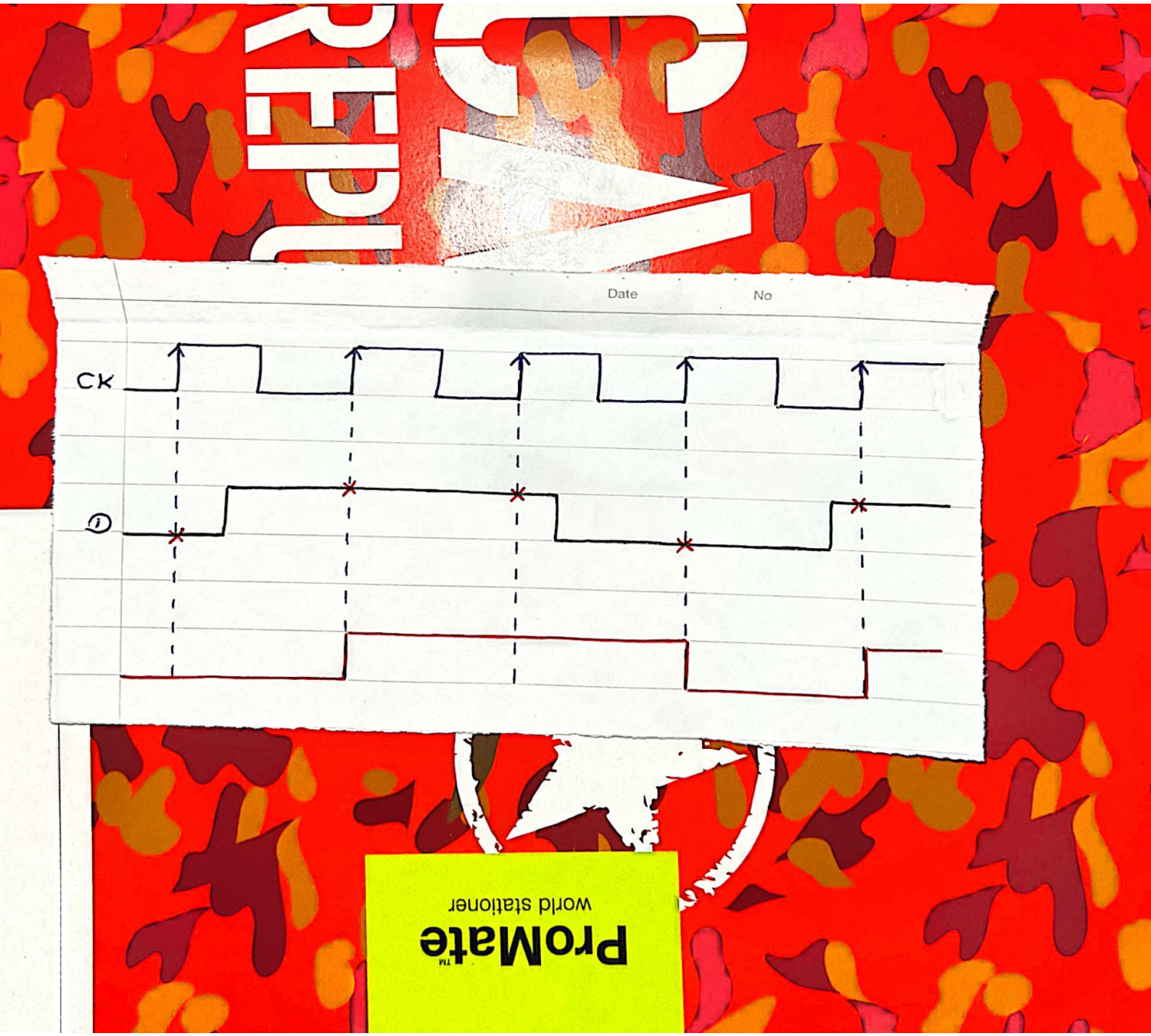
CK	D	Q_{n+1}
0	0	Q_n
0	1	Q_n
1	0	0
1	1	1

When CK is low
$Q_{n+1} = Q_n$

Complete Truth Table of a D FF

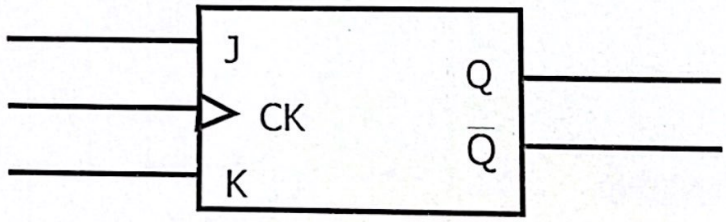
When CK =1

D	Q_n	Q_{n+1}
0	0	0
	1	0
1	0	1
	1	1



JK flip-Flop

Symbol:



Reduced TT:

When CK=1

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q}_n$

When CK = 0

$Q_{n+1} = Q_n$

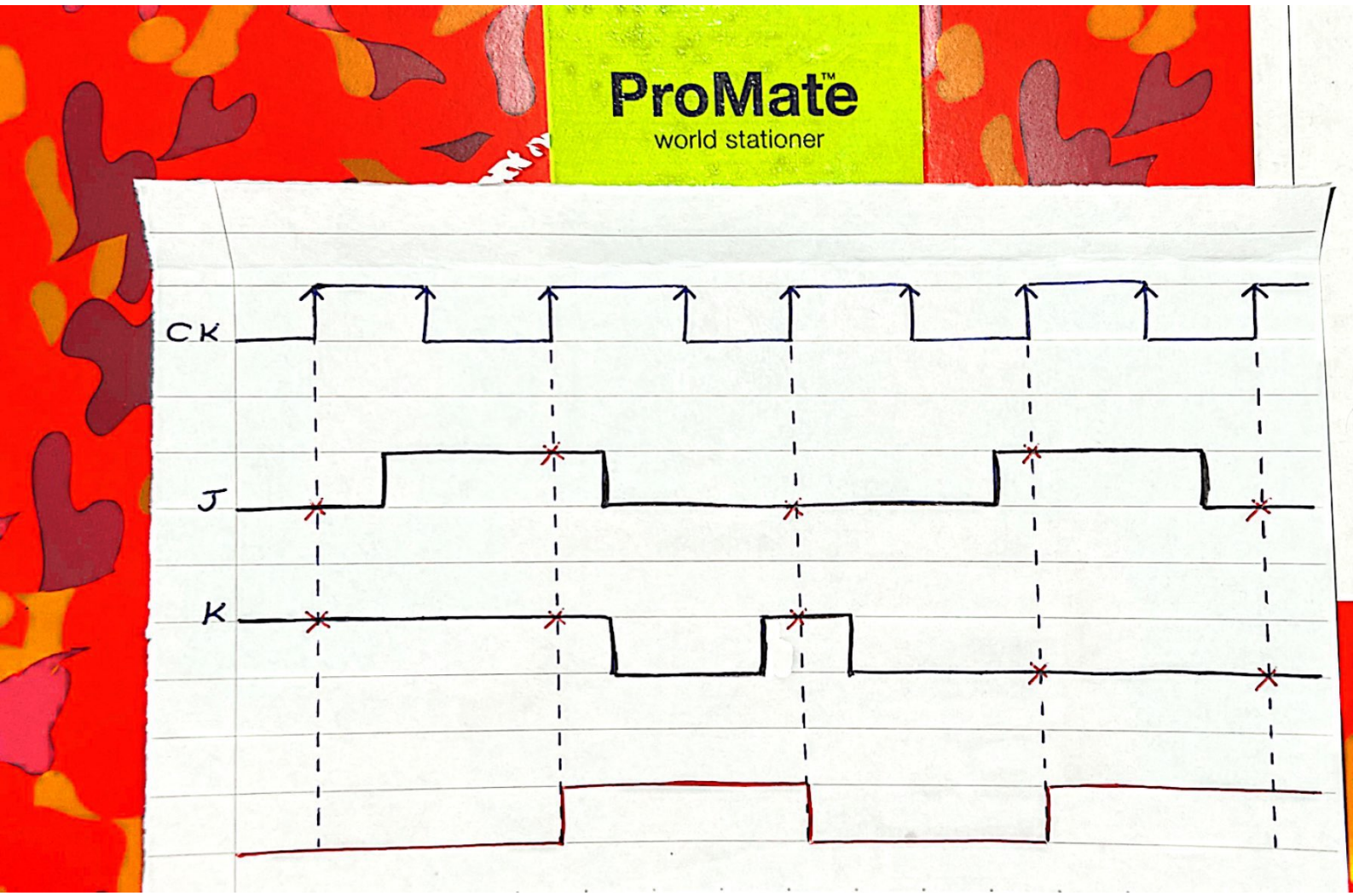
Complete Truth Table of a JK FF

When CK =1

J	K	Q_n	Q_{n+1}
0	0	0	0
		1	1
0	1	0	0
		1	0
1	0	0	1
		1	1
1	1	0	1
		1	0

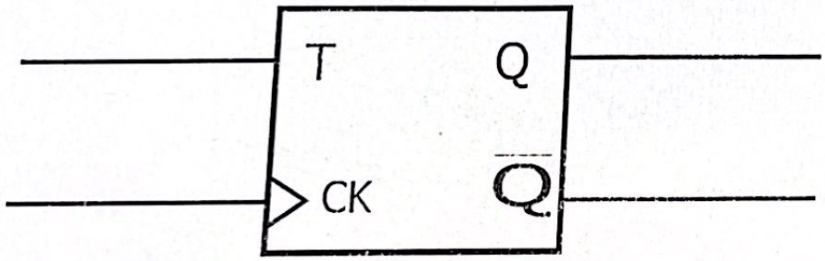
Toggle Flip-flop(T-FF)

Symbol:



Toggle Flip-flop(T-FF)

Symbol:



Reduced TT

When CK=1

T	Q_{n+1}
0	Q_n
1	$\overline{Q}_n$

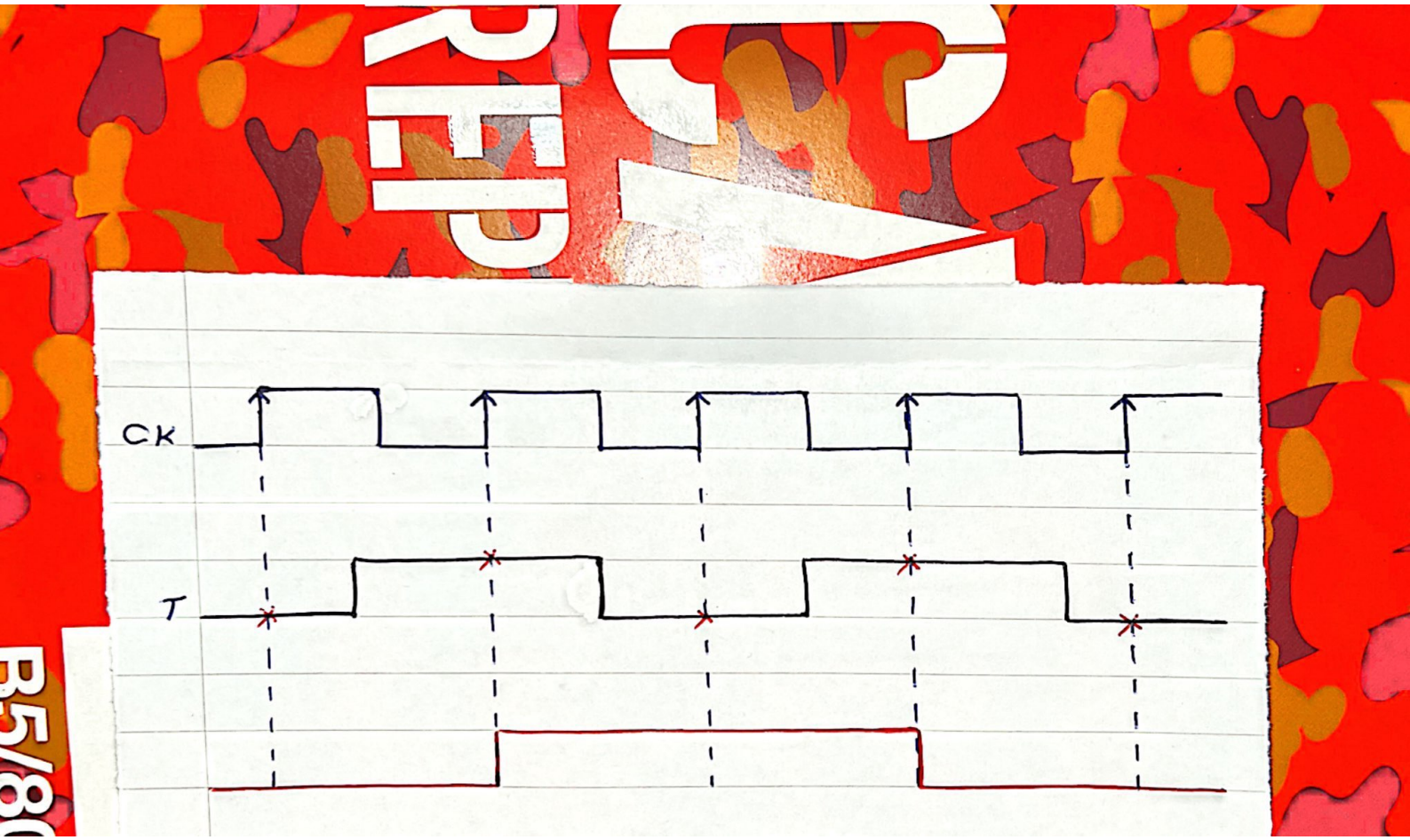
When CK = 0

$Q_{n+1} = Q_n$

Complete Truth Table of a T FF

When CK =1

T	Q_n	Q_{n+1}
0	0	0
	1	1
1	0	1
	1	0



SR Flip-flop

Truth table:

S	R	Qn	Qn+1
0	0	0	0
		1	1
0	1	0	0
		1	0
1	0	0	1
		1	1
1	1	0	NA
		1	

Excitation Table:

Qn	Qn+1	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

JK Flip Flop

Truth Table:

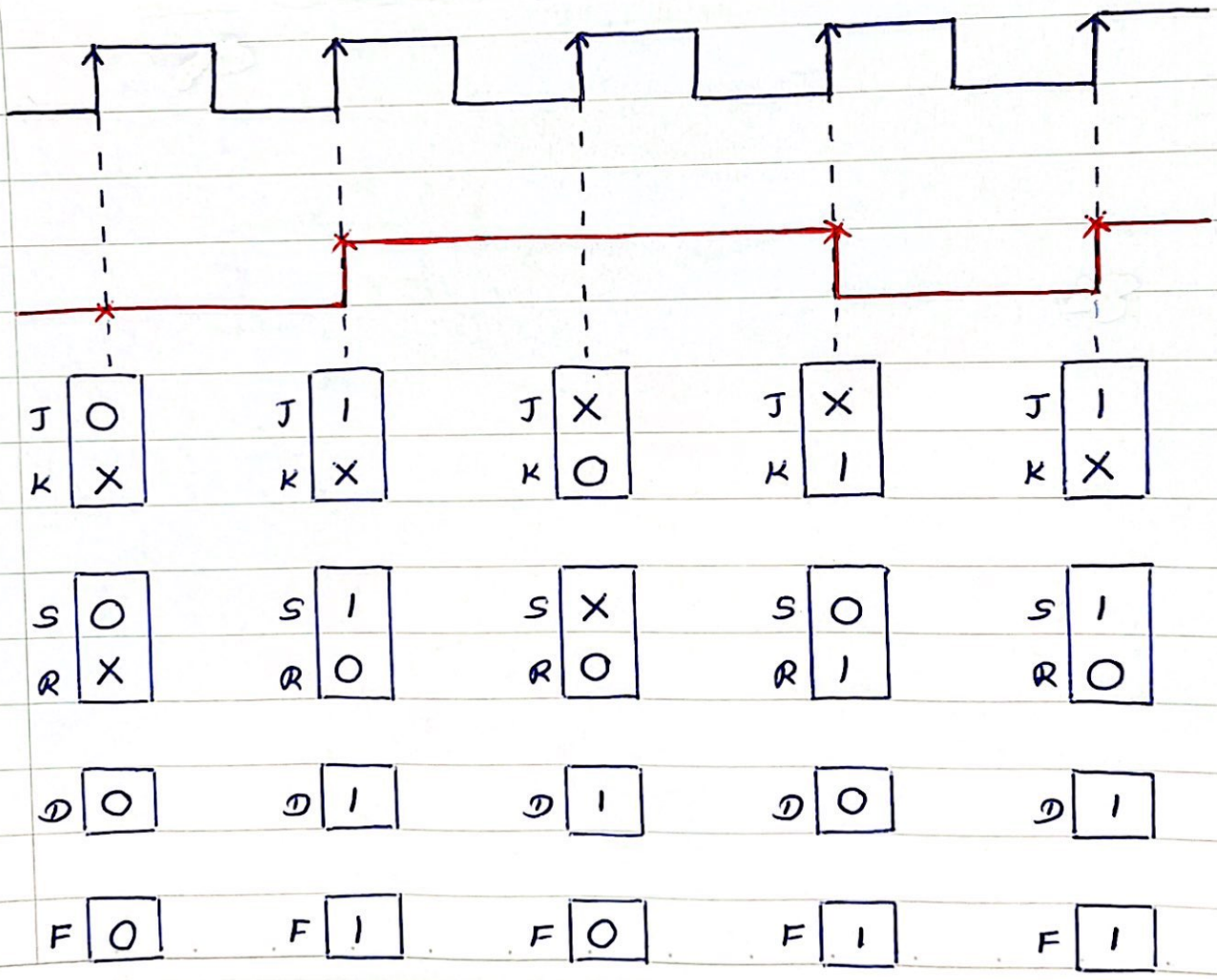
J	K	Q_n	Q_{n+1}
0	0	0	0
		1	1
0	1	0	0
		1	0
1	0	0	1
		1	1
1	1	0	1
		1	0

Excitation Table:

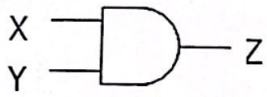
Qn	Qn+1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

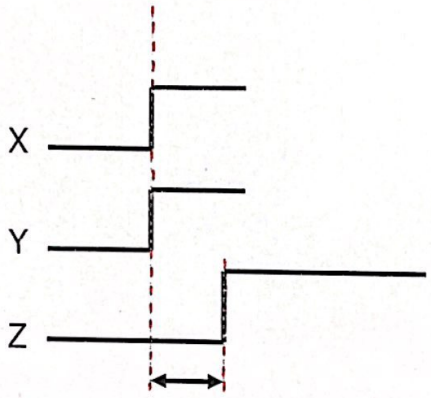
Excitation tables:

Q_n	Q_{n+1}	SR	D	JK	T
0	0	0X	0	0X	0
0	1	10	1	1X	1
1	0	01	0	X1	1
1	1	X0	1	X0	0



Propagation Delay





$t_p \approx 10^{-9}$ sec

Propagation Delay

Date

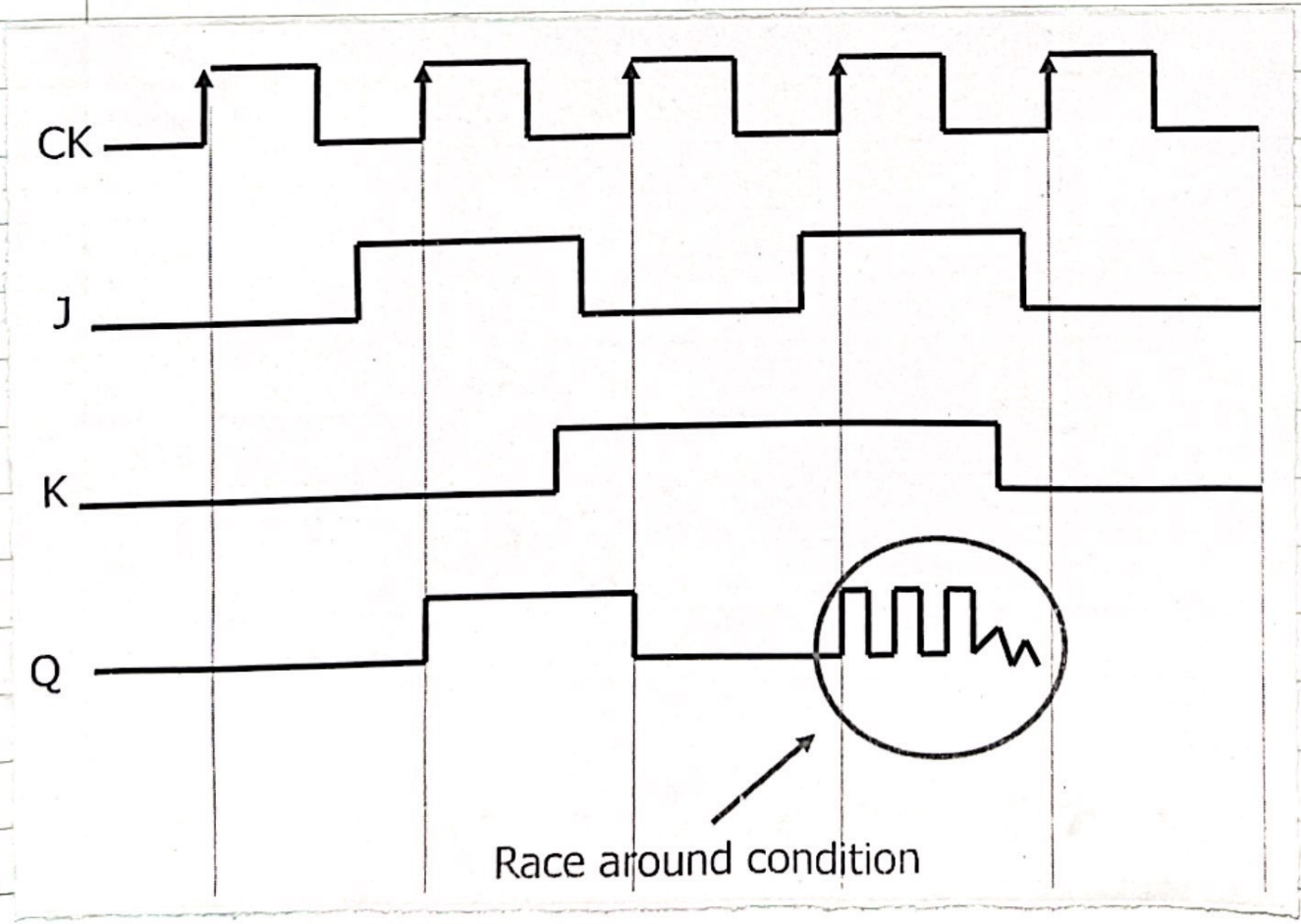
No

Date No

Race Around Condition

When, J = K = 1 and the clock is enabled, the output Q_{n+1} of a JK FF is $\bar{Q}_n$.

Therefore, the output oscillates at a frequency determinded by the propagation delay, till the clock is disabled. Therefore, at the end of the clock pulse, the value of the output cannot be determinded. The output can be either a 0 or a 1, depending on the clock duration and the frequency of oscillation.



Draw the output waveform for a:

(a) (+)ve edge triggered T FF

(b) (-)ve edge triggered T-FF

(c) (+) & (-) ve edge triggered T-FF