Saptagandaki Multiple Campus

(Affiliated to Tribhuvan University)
Bharatpur-10, Chitwan

Assignment Of Digital Logic (CACS 105)

Department of Bachelor in Computer Application
Faculty of Humanities and Social Sciences
Tribhuvan University (TU)
Kirtipur, Nepal

Submitted by

Name: *Anuj Bhurtel*

Roll number: *06*

Submitted to:

Ravi Tiwari Sir

Saptagandaki Multiple Campus
Department of Bachelor in Computer Application(BCA)

Date Of Submission:- 2076/10/26

1.1. What is Number system? Difference between analog & digital system with advantages/disadvantages & examples.

→ The system which is based on numbers is called number system. There are two types of number system.
i.e. ⓐ Non-positional. ⓑ Positional.

Difference between Digital & Analog System.

	Analog	Digital.
1.	These systems work with natural or physical values.	These systems work with digits.
2.	It works upon continious data.	It works upon discrete data.
3.	It operates by measuring and comparing.	It operates by counting and adding i.e. calculating.
4.	Its accuracy is low.	Its accuracy is high.
5.	It is special purpose in nature.	~~Results~~ It is general purpose in nature.
6.	Lower cost compared to digital system.	Higher cost compared to analog system.
7.	Normally, it can't be reprogrammed	It can be reprogrammed.
8.	e.g. Odometer, speedometer etc.	e.g. IBM PC, other digital computers or digital machines.

Advantages of Digital systems:

→ Less expensive
→ More reliable
→ Easy to manupltlate
→ Flexibility & compatibility.
→ Can be reprogrammed. etc.

Dis-advantages:

→ Consume more energy than analog circuits.
→ Quantization error during analog signal ~~sy~~ sampling.
→ Digital computer manupulates discrete elements of information by means of a binary code.

Advantages of Analog systems:

→ Cheaper as compare to digital.
→ Used for special purpose.
→ Uses less bandwidth.
→ More accurate. etc.

Disadvantages:

→ Accuracy is very low.
→ No storage facility.
→ Can't work on real time basis.
→ Can measure only natural or physical values.
→ Can't be reprogrammed. etc.

Q.2. What is IC? Write the advantages of IC.

→ An IC (Integrated Circuit) is an association or connection of various electronic devices such as resistors, capacitors, & transistors.
~~to a~~

Types of IC's

ⓐ Analog or linear ICs.
ⓑ Digital or logic ICs.

Advantages of ICs.

→ In consumer electronics, ICs have made possible the development of many new products, including personal calculators, computer and many others.

→ They have also been used to improve or lower the cost of many existing products.

~~→ The logic & arithmetic~~

→ It's more reliable

→ It has suitable for small operation.

→ It has increased the operating speed due to an absence of parasitic & capacitance effect.

→ The temperature differences between components of a circuit are small. etc.

3. Subtract: 1010.110 − 101.101 using both 1's & 2's Complement.

Solution,

1's complement of 1010.110 = 0101.001

Adding it with minuend = +101.101

$$\begin{array}{r} 0101.001 \\ +101.101 \\ \hline 1010.110 \end{array}$$

In the result there exists no carry. So to find the result doing complement of ~~the~~ (1010.110) & hence will be the required result.

∴ 1's complement of 1010.110 = 0101.001.

Hence, $(101.001)_2$ is the result when $(1010.110)_2$ is subtracted by $(101.101)_2$.

Again,

2's Complement of 1010.110 = (0101.001+1)
= 110.001

Adding it with minuend = +101.101

$$\begin{array}{r} 110.001 \\ +101.101 \\ \hline 1011.110 \end{array}$$

In the result there exists no carry, so calculating 2's Complement of the result.

∴ 2's complement of 1011.110 = (0100.001 + 1)
= (0101.001)

Hence, $(101.001)_2$ is required result after subtracting $(101.101)_2$ from $(1010.110)_2$.

14. Subtract: ~~615.6~~ 675.6 − 456.4 using 10's & 9's complement.

Solution,

9's complement of 675.6 = (999.9 − 675.6)
= 324.3

Adding it with minuend = + 456.4

780.7

In the result there exists no any carry. So calculating the 9's complement of the result.

∴ 9's complement of 780.7 = (999.9 − 780.7)
= 219.2

Hence, (219.2) is the required result after subtracting 456.4 from 675.6 #

Again,

10's complement of 675.6 = (324.3+1)
= 325.3

Adding it with minuend = +456.4

781.7

In the result there exists no carry. So calculating the 10's Complement of the result.

∴ 10's complement of 781.7 = (999.9 − 781.7 + 1)
= 219.2

Hence, (219.2) is the required result after subtracting 456.4 from 675.6. #

Q.5. What is logic gate? List universal gates. Describe different logic gates used in computer.

> A logic gate is an electronic circuit that operates on one or more inputs signals to produce an output signal.

Universal gates:-

(i) **NAND gate:-**

It is the combination of AND & NOT gate. This electronic gate produces low (0) output when all inputs are high, otherwise the output will be high.

Graphical symbol:-

Algebric expression:- $X = \overline{A \cdot B}$

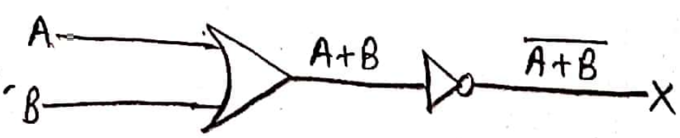
Truth table:

Inputs		A·B	Outputs
A	B		$\overline{A \cdot B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

(ii) **NOR gate:-**

It is the combination of OR & NOT gate. This electronic gate produce high voltage only when all the inputs are low, otherwise it produces low voltage. as the output will be low.

Graphical symbol:-



Algebric expression:- $X = \overline{A+B}$

Truth table:-

Inputs		$A+B$	Outputs
A	B		$(A+B)'$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

The ~~go~~logic gates used in computer are:-

(i) AND gate:-

It is an electronic circuit, which produces high output when all inputs are high. Otherwise, the output will be low.

Graphical symbol:-

Algebric expression:- $X = A \cdot B$

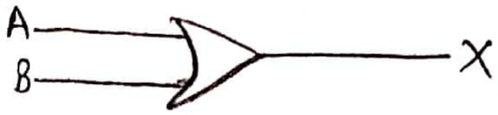
Truth table

Inputs		Output
A	B	$X = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

(ii) OR gate:-

This gate produces high output when one of the input is high. If all the inputs are low, then the output will be low.

Graphical symbol:-



Algebric expression: $X = A + B$

Truth table:

Inputs		Output
A	B	$X = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

(iii) NOT gate:-

It is an electronic circuit whose output is high when input is low and vice-versa. It is also called an inverter.

Graphical symbol:-

Algebric expression: $X = A'$

Truth table:

Input	Output
A	$X = \overline{A}$
0	1
1	0

Q.6. What is Boolean Algebra? Write the laws of Boolean algebra.

→ Boolean Algebra is algebric of logic, which deals with the study of binary variables & logical operations.

Laws of Boolean ~~The~~ Algebra

ⓐ Commutative law:-

The commutative law of Boolean algebra is expressed by.

(i) $(A+B) = (B+A)$

(ii) $A \cdot B = B \cdot A$.

ⓑ Associative law:-

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

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

ⓒ Distributive law:-

(i) $A \cdot (B+C) = AB + AC$.

(ii) $A + (B+C) = \cancel{AB}\ (A+B) + (A+C)$.

ⓓ Identity law:-

(i) $A + 0 = A$.

(ii) $A \cdot 1 = A$.

ⓔ Complement law:-

(i) $A + A' = 1$

(ii) $A \cdot A' = 0$.

1.7 State & prove the Demorgan's Theorem.

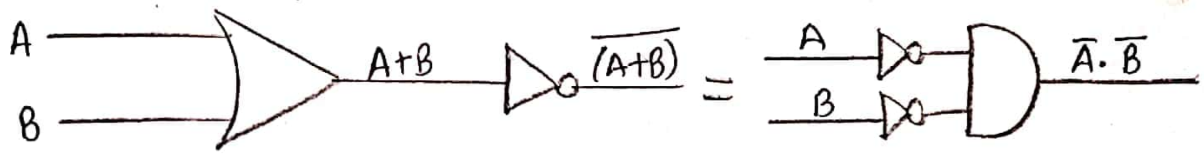
First Theorem:

The De-Morgan's first theorem states that, "The complement of a sum equals to the product of the complements."

i.e. $(\overline{A+B}) = \bar{A} \cdot \bar{B}$

Proof:

Graphical symbol:



Truth table:

Inputs			Output 1			Output 2
A	B	A+B	(A+B)'	A'	B'	A'.B'
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

Conclusion:

Comparing the values of $(\overline{A+B})$ & $\bar{A}.\bar{B}$ from the truth table, both are equal. Hence proved.

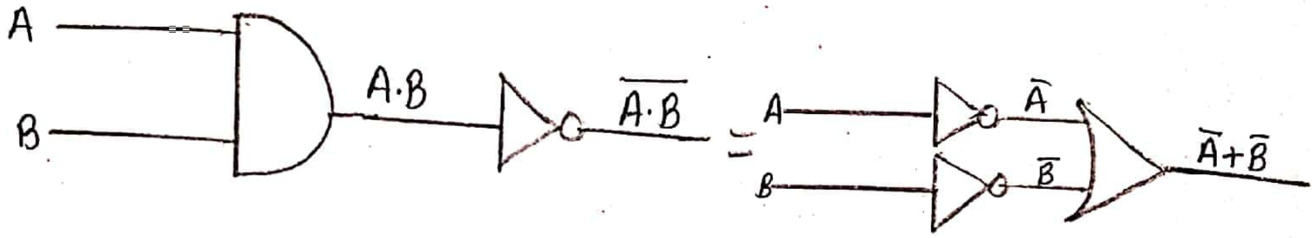
Second Theorem:

De-Morgan's second theorem states that, "The complement of a product is equals to the sum of complements."

i.e. $\overline{A \cdot B} = \bar{A} + \bar{B}$

Proof:

Graphical symbol:



Truth table:

Inputs			Output 1			Output 2
A	B	AB	$\overline{A \cdot B}$	$\bar{A}$	$\bar{B}$	$\bar{A} + \bar{B}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

Conclusion:

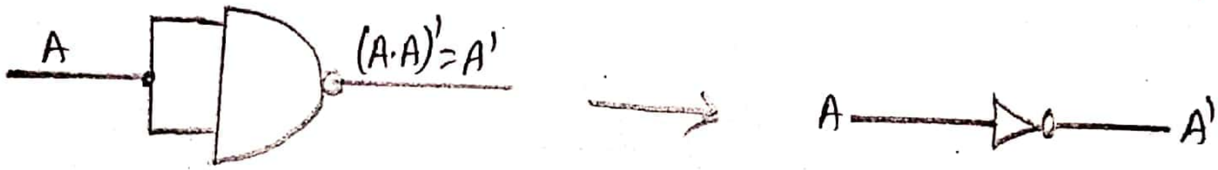
Comparing $(\overline{A \cdot B})$ & $(A' + B')$ from the truth table, both are equal. Hence proved.

1.8. What is universality of logic gate? ~~Rel~~ Realize NAND & NOR as an Universal gates.

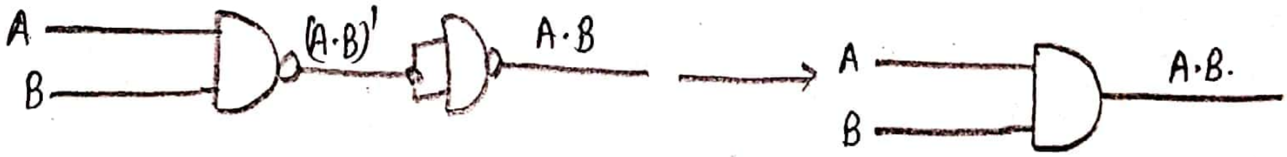
→ Universality of logic gate means to implement any logic expression using the specific gates. NAND gate and NOR gate are also known as universal gate because these gates are sufficient to implement any Boolean expression.

Universality of NAND gate.

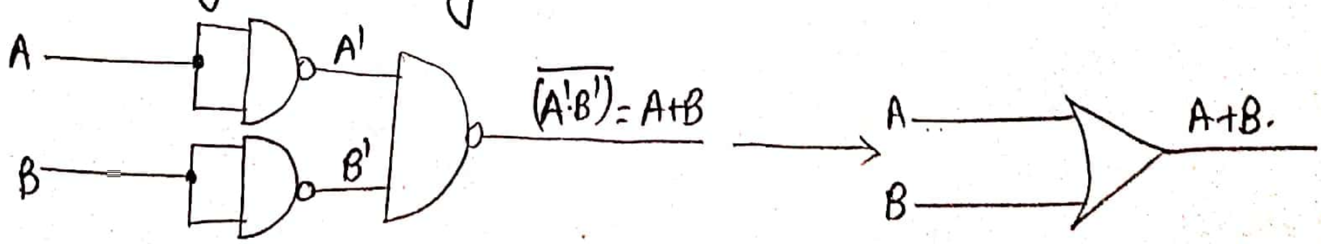
(i) Implementing an Inverter using NAND.



(ii) Implementing AND using NAND.

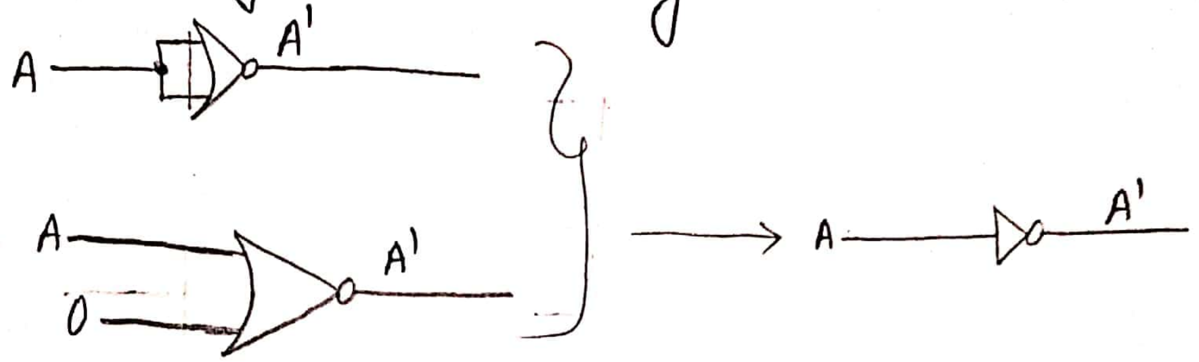


(iii) Implementing OR using NAND.

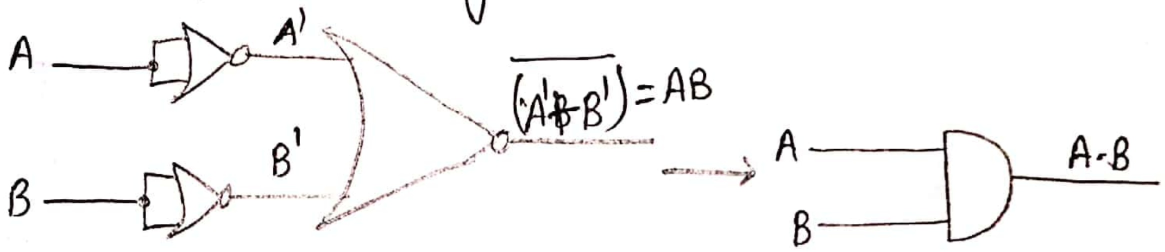


Universality of NOR gate

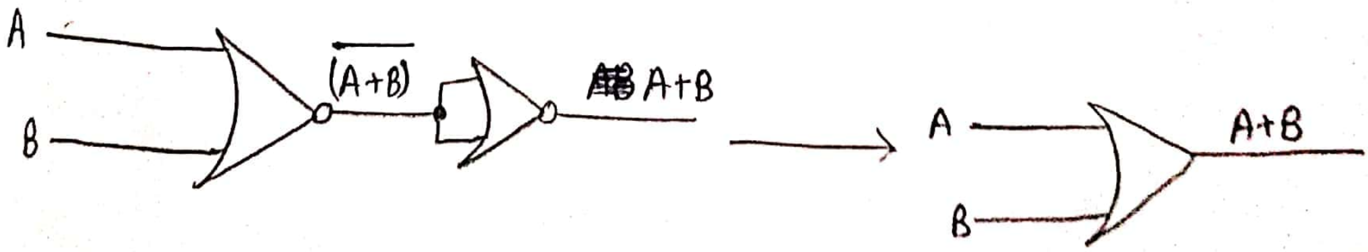
(i) Implementing an Inverter using NOR.



(ii) Implementing AND using NOR



(iii) Implementing OR using NOR.



Q.10. Simplify the following using K-Map method.

$$F(W,X,Y,Z) = \Sigma m(0,2,3,4,6,8,10,14,15) + D(7,11,12)$$

<u>Given,</u>

Here,

WX\YZ	00	01	11	10
00	1		1	1
01	1		X	1
11	X		1	1
10	1		X	1

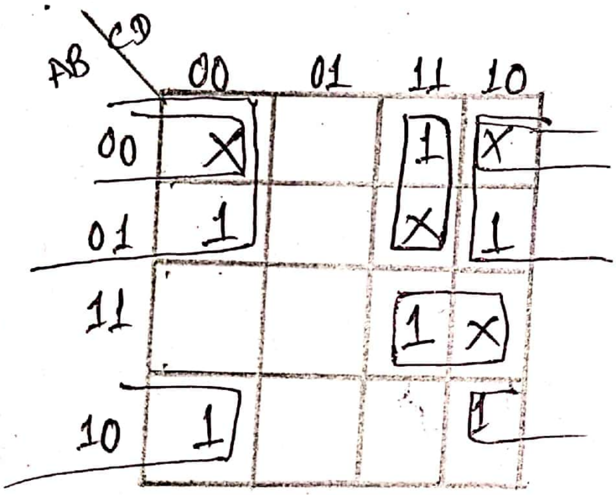
$\therefore F = Y + Z'$ is the minimum SOP. #

Q.11. Simplify the following using K-map method.

$F(A,B,C,D) = \sum m(3,4,6,8,10,15) + D(0,2,7,14).$

Given,

Here,



$\therefore F = \bar{B}\bar{D} + \bar{A}\bar{D} + \bar{A}CD + ABC$ is the minimum SOP. #

Q.12. Simplify the following (Using K-map) the Boolean function in both SOP & POS using don't care condition.

$F(A,B,C,D) = \Pi(0,1,3,7,8,12)$ & $\Pi d(5,10,13,14)$.

Here,

For POS,

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

$F = (A'+D).(A+D').(A+B+C).(A'+B'+C)$. is the required form of POS. #

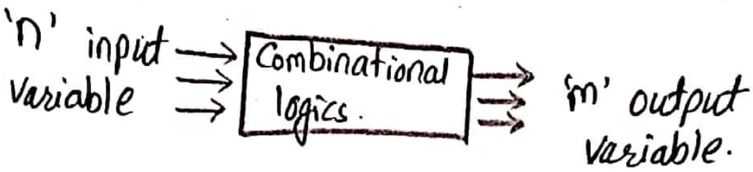
For SOP,

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

$F = CD' + AD + A'BC'$ is the minimum SOP #

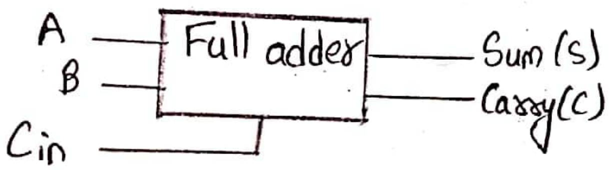
Q.13. What is Combinational circuit? Design full adder with truth tables and circuit diagram.

→ Combinational circuit is a logic circuit used to get m output variables from n input variable.



Full Adder:

→ Full adder is a combinational logic circuit used ~~to get~~ or designed to ~~get~~ add two single bit number with a carry.



Truth table:-

A	B	Cin	S	C.
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

K-map for Sum:

A \ BC	00	01	11	10
0		1		1
1	1		1	

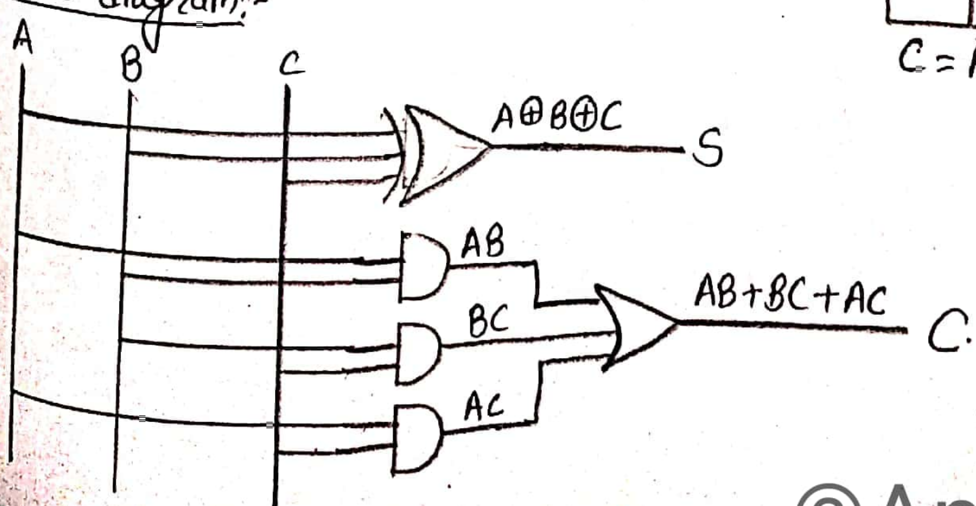
$S = A \oplus B \oplus C$

K-map for Carry:

A \ BC	00	01	11	10
0			1	
1		1	1	1

$C = AB + BC + AC$.

Circuit diagram:-



15. Define Decoder. Draw logic diagram & truth tables of 3 to 8 line decoder.

→ Decoder is a logic circuit that accept a set of input that represent a binary number and activates that output which corresponding to the input binary number.

→ The decoder has 'n' inputs & an enable line of 2^n output line.

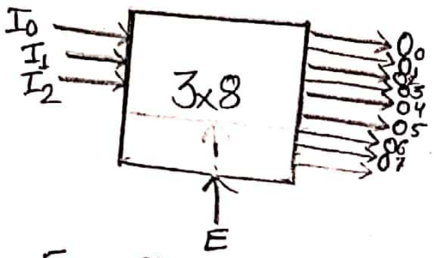


Fig:- Block diagram of 3 to 8 line decoder.

Truth table:

Inputs			Outputs							
I_0	I_1	I_2	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Logic diagram:

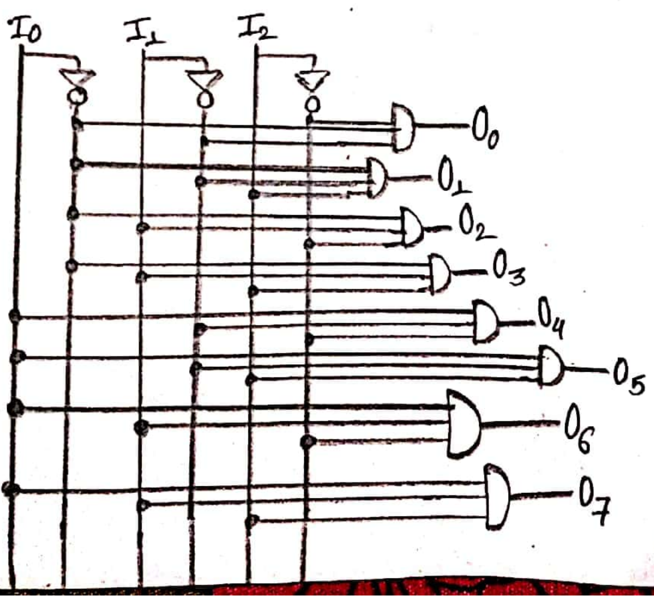


Fig:- Circuit diagram of 3x8 decoder.

16. Define Encoder. Draw logic diagram & truth table of octal to binary encoder.

→ An encoder is an digital function that produces a reverse operation from that of a decoder. An encoder has 2^n (or less) input lines & n output lines.

Octal to binary encoder

Octal-to-Binary takes 8 inputs & provides 3 outputs, thus doing the opposite of 3×8 decoder does. At any one time, only one input line has a value of 1. The figure below shows the truth table of Octal-to-Binary encoder.

Truth table:

Inputs								Outputs		
I0	I1	I2	I3	I4	I5	I6	I7	O1	O2	O3
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Logic diagram:-

$$O_0 = I1 + I3 + I5 + I7$$

$$O_2 = I2 + I3 + I6 + I7$$

$$O_3 = I4 + I5 + I6 + I7$$

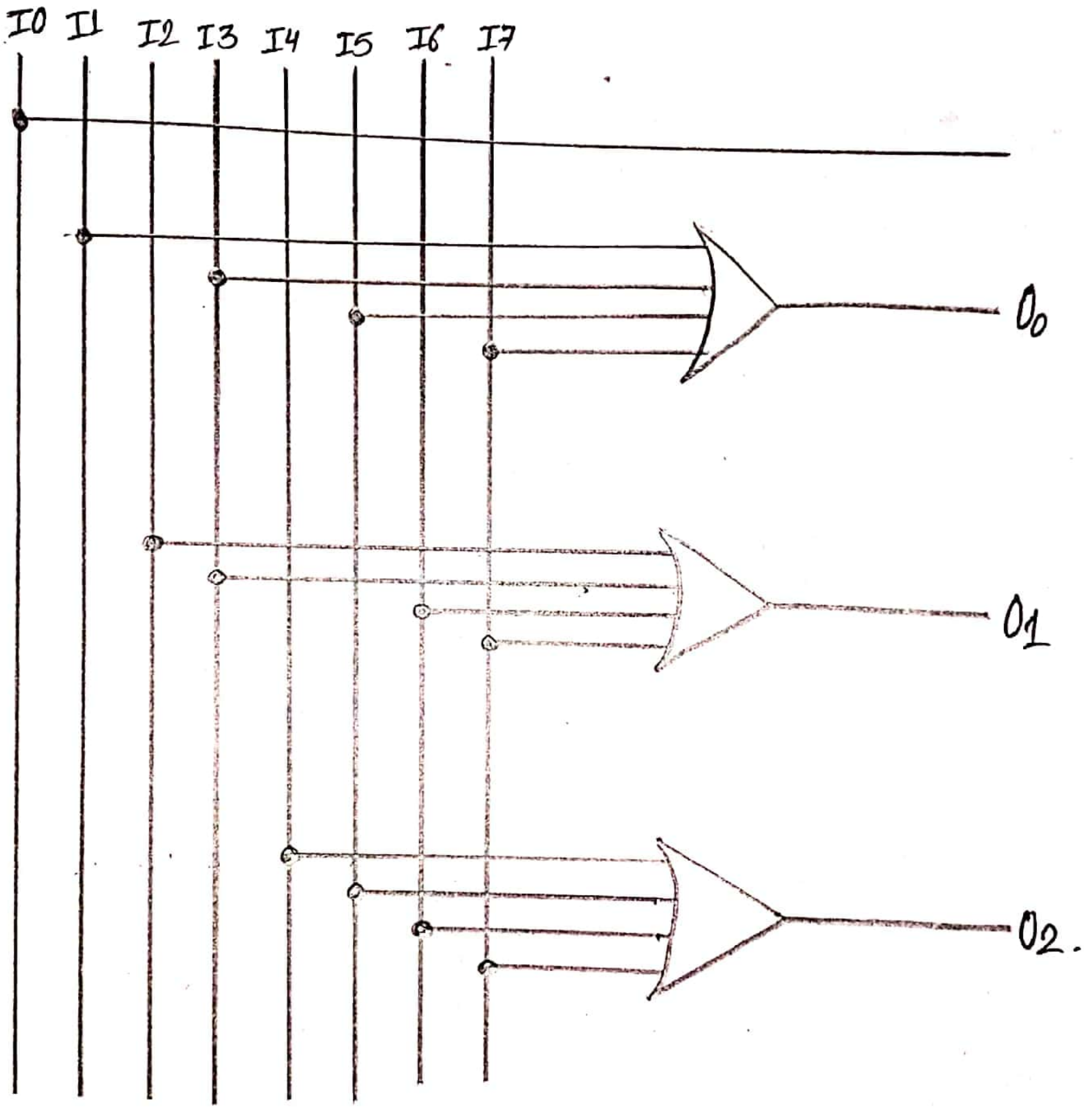


Fig:- Circuit diagram of Octal-to-Binary Encoder.

17. Simplify the following using K-map & implement it using any gates.
$F(A,B,C,D) = \Sigma(0,1,2,4,5,6)$ & $D = \Sigma(8,9,10,12,13,14)$.

Given,

Here, K-map for the given $F(A,B,C,D)$ is,

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

$F = C' + D'$

Implementing $F = C' + D'$ using gates:

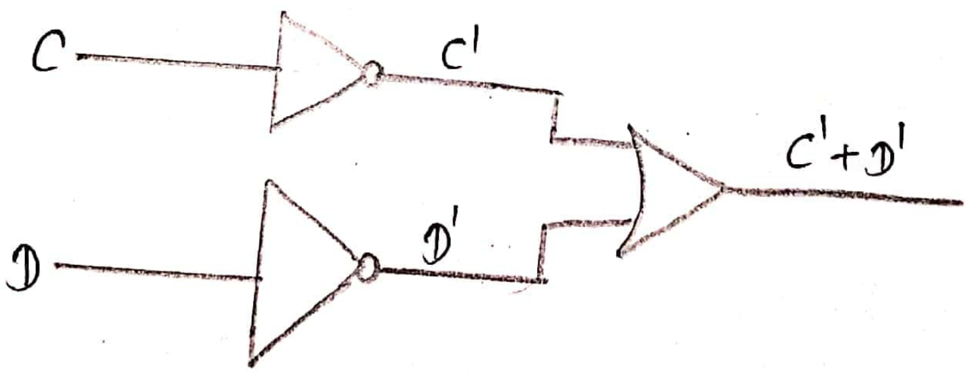


Fig:- Circuit diagram (Logic gates)

Q.18. Simplify the Boolean function $F = A'B'C' + B'CD' + A'BCD' + AB'C'$.

Given,

$$F = A'B'C' + B'CD' + A'BCD' + AB'C'.$$

Taking first term,

$$A'B'C' = A'B'C'D + A'B'C'D'$$

Second term,

$$B'CD' = AB'CD' + A'B'CD'$$

Third term,

$$A'BCD'$$

Last term,

$$AB'C' = AB'C'D + AB'C'D'$$

Now,

$$F = A'B'C'D + A'B'C'D' + AB'CD' + A'B'CD' + A'BCD' + AB'C'D + AB'C'D'$$

$$\therefore F(A,B,C,D) = \Sigma m(0,1,2,6,8,9,10). \#$$

19. Explain error detection codes with suitable examples.

→ Binary information can be transmitted from one location to another by electric wires or other communication mediums. Any external noise introduced into the physical communication medium may change some of the bits from 0 to 1 or vice-versa.

The purpose of an error-detection code is to detect such bit-reversal error. One of the most common way to achieve error detection is by means of a parity-bit. Parity bit is a extra bit included to make the total number of 1's in the resulting codes word either odd or even. A message of 4-bits & a parity bit P are shown in the table below as an example.

Odd Pariety				Even Parity			
Message	P	Message	P	Message	P	Message	P
0000	1	1000	0	0000	0	1000	1
0001	0	1001	1	0001	1	1001	0
0010	0	1010	1	0010	1	1010	0
0011	1	1011	0	0011	0	1011	1
0100	0	1100	1	0100	1	1100	0
0101	1	1101	0	0101	0	1101	1
0110	1	1110	0	0110	0	1110	1
0111	0	1111	1	0111	1	1111	0

20. Express the Boolean function $F = xy + x'z$ in a product of maxterm form.

Solution,

~~He~~ Given,

$$F = xy + x'z.$$
$$= (xy + x')\cdot(xy + z) \quad [\because \text{distributive law}]$$
$$= (x + x')\cdot(y + x')\cdot(x + z)\cdot(y + z)$$
$$= (x' + y)\cdot(x + z)\cdot(y + z)$$

Now,

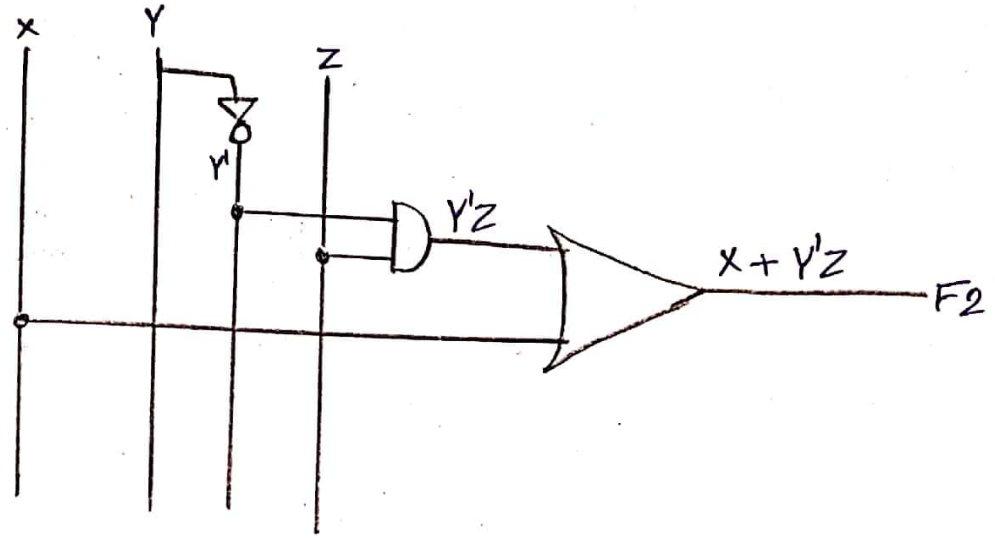
$$(\bar{x} + y) = (\bar{x} + y + z\cdot\bar{z})$$
$$= (\bar{x} + y + z)\,(\bar{x} + y + \bar{z})$$
$$(x + z) = (x + \bar{y} + z)\cdot(x + y + z)$$
$$(y + z) = (x + y + z)\,(x + \bar{y} + z).$$

$$\therefore F = (\bar{x} + y + z)\cdot(\bar{x} + y + \bar{z})\cdot(x + \bar{y} + z)\cdot(x + y + z).$$

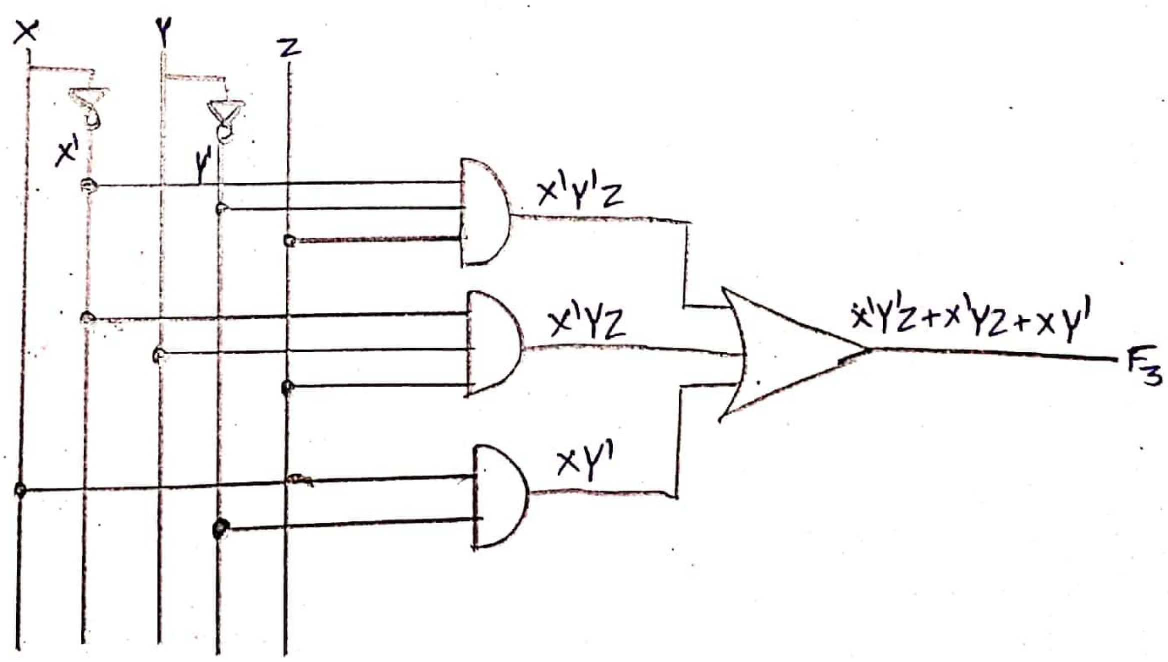
$\therefore F(X, Y, Z) = \Pi(0, 2, 4, 5)$ is ~~the~~ in the POS form #

Q.21 Draw the Logics gates of following.

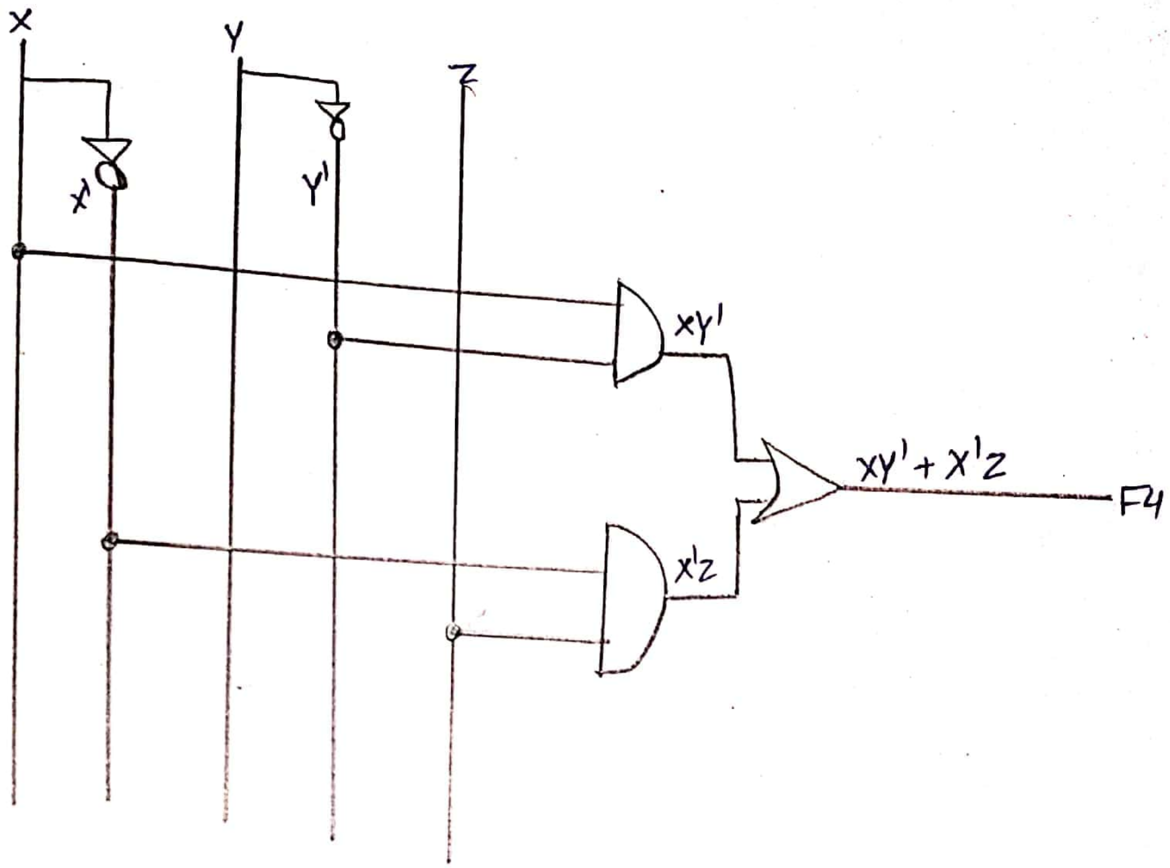
ⓐ $F_2 = x + y'z$.



ⓑ $F3 = x'y'z + x'yz + xy'$

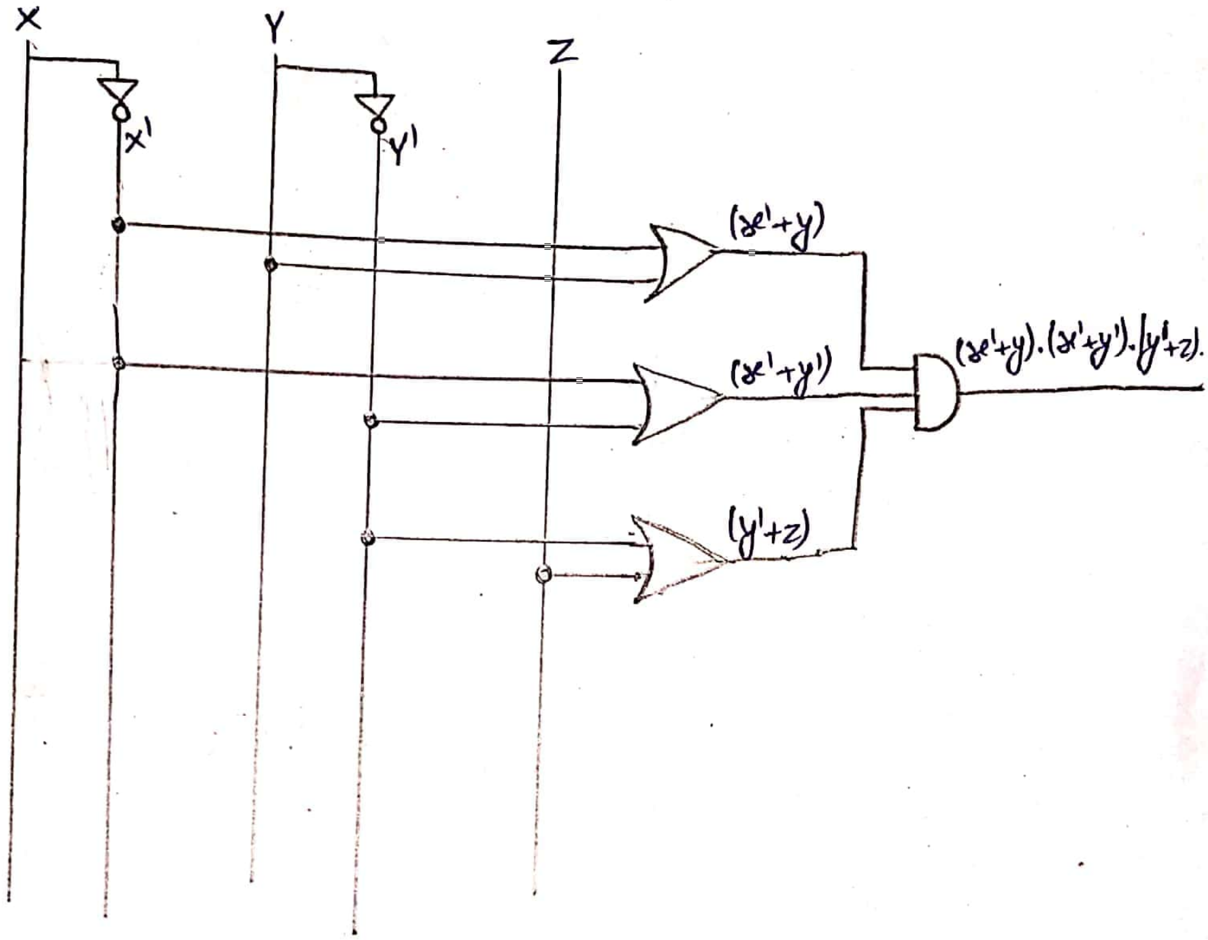


(c) $F4 = xy' + x'z$



Q.22. Draw the logics gates of following.

$(x'+y)(x'+y')(y'+z)$.



——— THE - END ———