Assignment - 01

Q-1 convert the decimal to
(i) the binary equivalent of (13.375).

(i) 13.375

2	13	
2	6	1
2	3	0
	1	1

$(13)_{10}$

$(1101)_2$

$0.375 \times 2 = 0.75$
$0.75 \times 2 = 1.5$
$0.5 \times 2 = 1$
$(0.375)_{10} = (011)_2$

Ans $(1101.011)_2$

(ii) (82.25)

2	82	
2	41	0
2	20	1
2	10	0
2	5	0
2	2	1
	1	0

$(1010010)_2$

$0.25 \times 2 = .50$ | 0
$0.5 \times 2 = 1$ | 1

Ans $= (1010010.01)_2$

16	82	
16	5	2
	0	5

$(52)_{16}$

$0.25 \times 16 = 4.00$

Ans $= (52.4)_{16}$

(2) Simplify the Boolean expression :-

$$A'BC + A'B'C + ABC + AB'C$$

$$A'C(B+B') + AC(B+B')$$

$$A'C + AC = C$$

(3) Reduce given Boolean expression

$$F(X,Y,Z) = X'Y + YZ' + YZ + XY'Z'$$

$$X'Y + YZ' + YZ + XY'Z'$$

$$= Y + X'Y + XY'Z' \qquad (Y + X'Y) = Y$$

$$= Y + XY'Z'$$

(4) equilant circuit expression for the Boolean expression

$x'y'z + yz + xz$	$z[(x'+y)(y'+y)+x]$
$x'y'z + z(x+y)$	$z[x'+y+x]$
$z(x'y' + (y+x))$	$z[1+y]$
$= z(x'y' + x + y)$	z

(5) Significance of Karnaugh maps and an example -

K-maps are used for simplying Boolean expression by minimizing the number of terms in a logic function.

they provide a visual way to find and eliminate redundant terms using grou

Ex: for a 4-bit function with the truth table -

A	B	C	D	F(A, B, C, D)
0	0	0	0	1
0	0	0	1	1
0	0	1	1	0
0	0	1	1	0
0	1	0	0	1

We can represent the function in a K-map and simplify it by grouping is to minimize the expression.

(b) Prove -

$ABCD + ABC'D' + ABCD' + ABC' + ABCDE + ABC'D'E$
$+ ABC'DE$ simplifies to AB

$ABCD + ABC'D' + ABCD' + ABC' + ABCDE + ABC'D'E + ABC'DE$

$ABC + ABC'D' + ABDE + ABC'$

$ABC + ABC' + ABDE$

$AB + ABDE$

AB

(4) Explain De-Morgan's thm applications -

De Morgan's Thm :-

- The complement of a product of variables is equal to sum of complements of variables. $\overline{A \cdot B} = \overline{A} + \overline{B}$

- The complement of sum is equal to the product of complements.

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

Applications -

De Morgan's Law are essential in digital logic design, particularly in converting complex expression with NAND and NOR gates, as these gates are often used to implement the logic functions in hardware.

(2) What is a universal gate? Discuss its properties.

Universal Gates - A universal gates is a type of logic gate that can be used to implement any Boolean function without needing to use any other gate.

(NAND or NOR)

Properties of Universal gates

(1) NAND Gates :-

Any Boolean expression can be implemented using just NAND gates. A NAND gate is the inverse of the AND gate but with proper configurations, it can produce NOT, AND, and OR functions.

(2) NOR gate - Similar to NAND, a NOR gate can implement any Boolean function by combining several NOR gates.

Both NAND and NOR gates are called "universal gates" because of their ability to perform the basic building blocks of any digital circuits.

(1) Universality

(2) • NOT using NAND

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

(3) • AND using NAND

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

• OR using NAND

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

• NOT using NOR

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

• AND using NOR

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

• OR using NOR

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

(9) Using De'morgans law, reduce the follo expression -

$$Y = (AB' \cdot (A + C))' + A'B \cdot (A + B + C')'$$

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

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

$$= \bar{A} + B + \bar{A} \cdot \bar{C} + 0$$

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

(10) Define truth table and its significance for the XOR and XNOR gate. Implement logic ckt gates for XOR and XNOR by using NAND and NOR gate.

Truth table for XOR

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

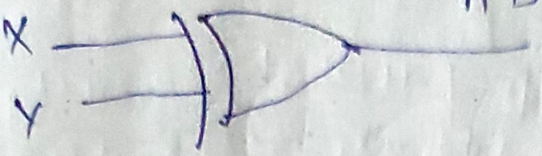
Truth table for XNOR

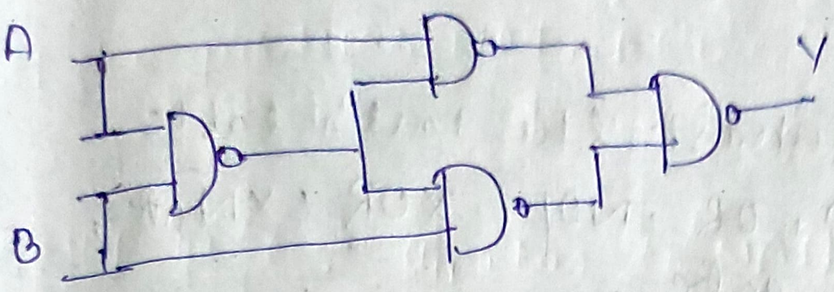
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

logic gate and significance using NAND and NOR.

XOR

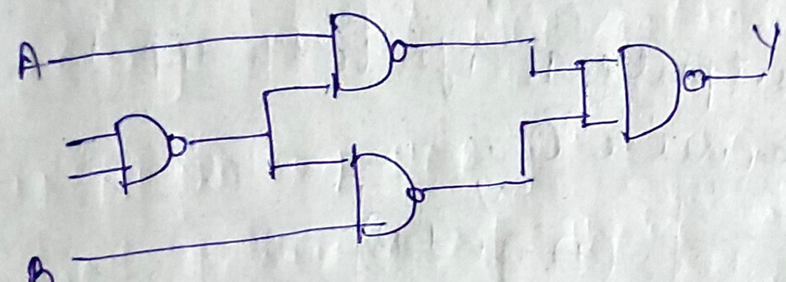
$\bar{A}B + \bar{B}A$



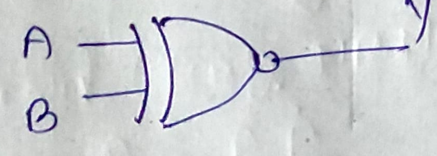


using NAND
(XOR)

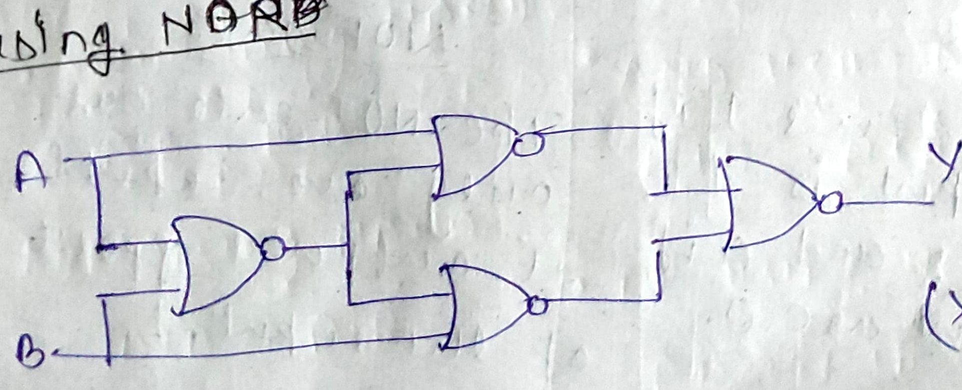
XNOR
using NAND



XNOR

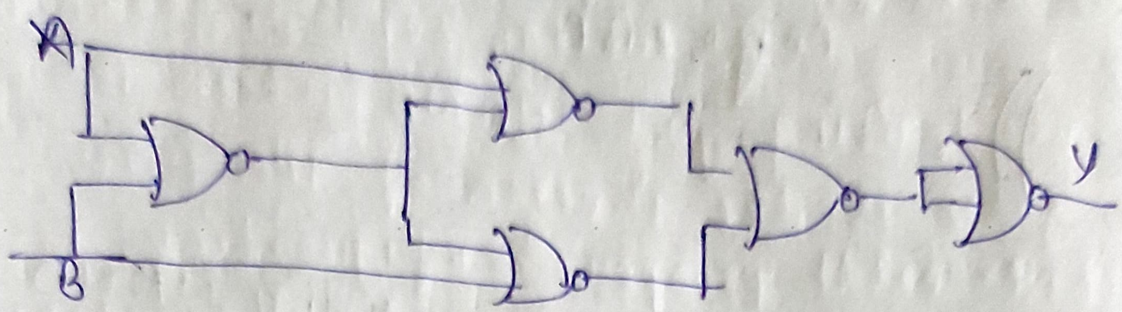


using NOR



(XOR)

using NOR → XNOR



(11) Compare and contrast the truth table functions of AND, OR, NOT, XOR, XNOR, NAND and NOR gate.

(1) AND Gate -

The AND gate outputs 1 if both inputs are 1, and otherwise 0.

A	B	Y (A and B)
0	0	0
0	1	0
1	0	0
1	1	1

(2) OR gate

OR gate outputs 1 if either input is 1 or both are 1, and 0 if both input are 0.

NOT gate

The NOT gate has only one input & it outputs opposite of the input.

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

③

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

④ XOR gate

The XOR gate outputs 1 if the I/P are different and 0, if they are the same.

⑤ XNOR gate

It is the inverse of XOR.

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

⑥ NAND Gate

The NAND gate is the inverse of the AND gate. It outputs 0 if both input are 1.

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

⑦ NOR Gate

The NOR gate is the inverse of OR gate. It O/P 1 if both are (I/P) 0.

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Differences:

(1) AND vs NAND – AND outputs 1 only when both inputs are 1, whereas NAND does not the opposite by outputting 0 when both I/P are 1.

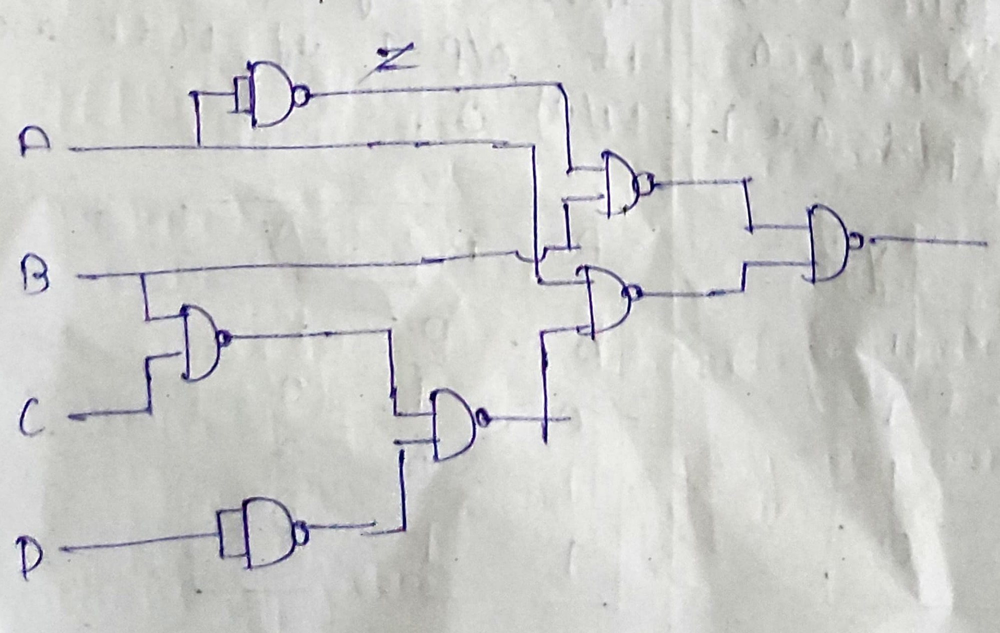
(2) OR vs NOR – OR outputs 1 when any input is 1. NOR o/p 1 only when both I/P are 0.

(3) XOR vs XNOR – XOR outputs 1 when any input is differ, while XNOR o/p 1 when inputs are the same.

(4) NOT – It is unique as it works with only one I/P, inverting its value.

(13) Implement following Boolean function by using NAND gates only.

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



(13) Define the SSI, MSI, LSI and VLSI logic circuits in DE. What it's characteristic.

SSI :- (Small Scale Integration)
SSI refers to ckt that contain a small number of logic gates (fewer than 10 gates) integrated into a single chip.

characteristics

Gate count - Typically fewer than 10 gates

Ex - Basic logic gates like AND, OR, NOT, flip-flops and decoders.

Complexity : Low

Power Consumption : Relatively low compared to larger circuits.

Applications : Used in simple combinational and sequential logic circuits.

2. MSI (Medium Scale Integration)
MSI circuits contain between 10 to 100 logic gates on a single chip.

characteristic -

Gate count - 10 to 100 gates.

Example - Multiplexer, decoders, counter and simple arithmetic ckt like adders

Complexity - Moderate

Power consumption Higher than SSI but still relatively efficient.

Application - used in slightly more complex digital system like memory decoders & arithmetic operations.

③ LSI (Large scale Integration) :- contain thousands of gates, allowing for the implementation of more complex functions -

characteristics

Gate count - 100 to several thousands gates

Ex - Microprocessors, memory chips (ROM, RAM) and simple microcontrollers.

complexity - High, capable of performing complex processing tasks.

Power consumption - Higher than SSI and MSI but optimized for a larger tasks.

Applications used in more advanced digital systems. such as computers and embedded systems.

④ VLSI (very large Scale Integration)

Gate counts - Hundred of thousands to millions of gates.

Example — Modern microprocessors, GPUs, FPGAs and complex system-on-chip (SoC) devices.

Complexity : very high, capable of executing sophisticated tasks (multi-core processing, advanced graphs rendering)

Power consumptions : optimized for performance but can be significant, especially in high-performance appl^n

Applications — used in advanced digital appl^n such as smartphones, computers, AI processors and automotive systems.

(14) find the value of x

(a) $(786.983)_{10} = (x)_{16}$

$$\begin{array}{r|l|l} 16 & 786 & \\ \hline 16 & 49 & 2 \\ \hline & 3 & 1 \end{array}$$

$(786)_{10} = (312)_{16}$

$0.983 \times 16 = 15.728$	15	F
$0.728 \times 16 = 11.648$	11	B
$0.648 \times 16 = 10.368$	10	A

~~0.368 × 16~~ =

$(0.983)_{10} \approx FBA_{16}$

Ans $= (312.FBA)_{16}$

(11) $(77.43)_{10} = (X)_{16}$

$$\begin{array}{r|l|l} 16 & 77 & \\ 16 & 4 & 13 \\ & 0 & 4 \end{array}$$

$(77)_{10} = (4D)_{16}$

F. Ans - $(4D.6E)_{16}$

$0.43 \times 16 = 6.88$

$0.88 \times 16 = 14.08$

$(0.43)_{10} \simeq 6E_{16}$

(12) Subtract the following numbers using 1's complement method -

(a) $(101)_2 - (100)_2$

(1) $A + (-B)$

$100 \xrightarrow{1's\ complement} (011)_2$

(2) Add

$$\begin{array}{r} 101 \\ 011 \\ \hline (1000)_2 \end{array}$$

ignoring the carry

Ans - $(000)_2$

$$\begin{array}{r} (000)_2 \\ +1 \\ \hline (001)_2 \end{array}$$

(b) $(111)_2 - (110)_2$

$$\begin{array}{r} 111 \\ 110 \\ \hline 001 \end{array}$$

(1) $(110)_2 \xrightarrow{1's\ com} (001)_2$

(2)

$$\begin{array}{r} 111 \\ 001 \\ \hline (\cancel{1}000)_2 \\ +1 \\ \hline (001)_2 \end{array}$$

ignoring the carry

(c) $(1111)_2 - (110)_2$

(1) $(0110)_2 \xrightarrow{\text{1's comp}} (1001)_2$

(2)

$$\begin{array}{r} 1111 \\ 1001 \\ \hline (11000)_2 \end{array}$$ ignoring carry

$$\begin{array}{r} (1000)_2 \\ +1 \\ \hline (1001)_2 \end{array}$$ Ans

(d) $(11001)_2 - (1101)_2$

(1) $(01101)_2 \xrightarrow{\text{1's comp}} (10010)_2$

$$\begin{array}{r} 11001 \\ 10010 \\ \hline (101011)_2 \end{array}$$ ignoring carry

$$\begin{array}{r} 01011 \\ +1 \\ \hline (01100)_2 \end{array}$$ Ans

(17) Add the following numbers using 2's comp method -

(a) $(42)_{10} + (-15)_{10}$ (assume using 8 bit)

(1) convert both to binary

$(42)_{10} = (00101010)_2$

$(15)_{10} = (00001111)_2$

$$\begin{array}{r|rl} 2 & 42 & \\ \hline 2 & 21 & 0 \\ 2 & 10 & 1 \\ \hline 2 & 5 & 0 \\ \hline 2 & 2 & 1 \\ \hline & 1 & \end{array}$$

11010

(2) complement of ↓

$$\begin{array}{r} 11110000 \\ +1 \\ \hline (11110001 \end{array}$$

00101010
11110001

$(1 00011011)_2 \rightarrow (27)_{10}$ Ans

discarded

(ii) $(123)_{10} + (-999)_{10}$

$(123)_{10} = (01111011)_2$ 16 bit

$(999)_{10} = 0011111100111$ (16 bit)

1's complement $(1111110000011000)_2$

+1

$(1111110000011001)_2 \rightarrow$ 2's compl

adding $(1.1.1.1.1.1.00100010)_2 = -(876)_{10}$

+1

$(0000001101101100)_2 \rightarrow$

(6) Convert the following Hexadecimal to Binary

8 4 2 1

0
1
2
3
4
5
6
7
8
9
A
1100 B
1101 C
1110 D
1111 G

(a) $(ABC \cdot 67B)_{16}$

$A \rightarrow (1010)_2$	$6 = (0110)_2$
$B \rightarrow (1011)_2$	$7 = (0111)_2$
$C \rightarrow (1100)_2$	$B = (1011)_2$

Ans $(101010111100.011001111011)_2$

(b) $(7B0.3E)_{16}$

$7 \rightarrow (0111)_2$	$3 = (0011)_2$
$B \rightarrow (1011)_2$	$E = (1110)_2$
$0 \rightarrow (0000)_2$	

Ans $(0111101100000011 1110)_2$

(c) $(473 \cdot 43A)_{16}$

$4 \rightarrow (0100)_2$	$3 = (0011)_2$	$3 = (0011)_2$
$7 \rightarrow (0111)_2$	$4 = (0100)_2$	$A = (1010)_2$

Ans $(0100011100 11.0100001 11010)_2$

(19) convert the following octal to Hexadecimal-

(a) $(3562)_8$

$(011\,101\,110\,010)_2$ An

$(0111\,0111\,0010)_{16}$

$7\quad 7\quad 2$

$(772)_{16}$ Ans

(b) $(434.67)_8$

$(100\,011\,100.110\,111)_2$ An

$\underbrace{001}_{1}\ \underbrace{0001}_{1}\ \underbrace{1100}_{B}\ .\ \underbrace{1101}_{D}\ \underbrace{1100}_{B}$

$(11B.DB)_{16}$ Ans