Page:
Date: / /

UNIT III

DIGITAL ELECTRONICS

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology
BANGALORE-560 054.

Number Systems:

Decimal Number System: The decimal number system uses the digits 0 to 9. All numbers are combination of these digits. Eg: 4362, 19.217.

Any number greater than '9' is represented by combination of these ten numbers

Eg. $(1240)_{10}$: The number '10' outside the bracket indicates the base or radix which represent the given number within the bracket is in decimal form or decimal number.

The number within the bracket, each number of symbol has positional value as well as absolute value i.e.,

3	2	1	0
1	2	4	0

$= 1\times10^3 + 2\times10^2 + 4\times10^1 + 0\times10^0$

$= 1000 + 200 + 40 + 0$

$= 1240.$

eg: 10^3 10^2 10^1 10^0 . 10^{-1}

5 6 7 8 . 9

(in power of 10) 5×10^3 6×10^2 7×10^1 8×10^0 . 9×10^{-1}

MSB . LSB

In general a number in a system of radix 'r' can be represented in the form of a polynomial in powers of 'r' as:

$$a_n a_{n-1} \ldots a_1 a_0 a_{-1} a_{-2} \ldots a_{-(m-1)} a_{-m} = a_n r^n + a_{n-1} r^{n-1} + \ldots + a_1 r^1 + a_0 r^0 + a_{-1} r^{-1} + \ldots + a_{-(m-1)} r^{-(m-1)} + a_{-m} r^{-m}$$

where a_n is the most significant digit &
a_m is the least significant digit

Gold

Binary Number System: The binary system is equivalent to electrical switch since has only two states either CLOSE or OPEN similarly binary system has only two values either '1' (logic 1) or '0' (logic 0). Hence binary number system has a radix of '2'

eg: $(1101)_2$, $(11001.0101)_2$

The binary number can be represented in polynomial form as

$$a_n a_{n-1} \cdots a_{-m} = a_n 2^n + a_{n-1} 2^{n-1} + \cdots + a_1 2 + a_0 2^0 + a_{-1} 2^{-1} + \cdots + a_{-(m-1)} 2^{-(m-1)} + a_{-m} 2^{-m}$$

eg: $(1101.0101)_2$ can be represented as.

$$(1\times2^3) + (1\times2^2) + (0\times2^1) + (1\times2^0) + (0\times2^{-1}) + (1\times2^{-2}) + (0\times2^{-3}) + (1\times2^{-4})$$

$$= 8 + 4 + 0 + 1 + 0 + \frac{1}{4} + 0 + \frac{1}{16}$$

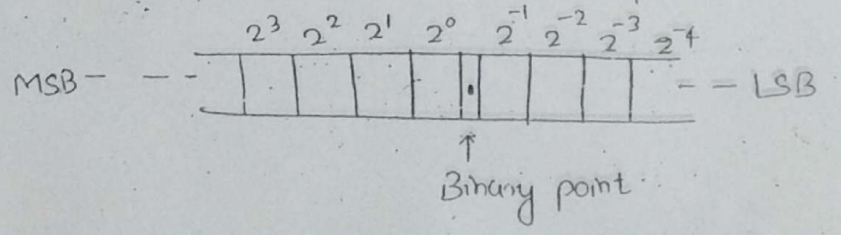
$$= 8 + 4 + 1 + 0.25 + 0.0625$$

$$= 13.3125$$

$$\therefore (1101.0101)_2 = (13.3125)_{10}.$$

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

The subscript indicates the radix or base of the number system to which the number in the parenthesis belongs.



eg:

2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}
1	1	0	1	.	1	0	1
1×2^3	1×2^2	0×2^1	1×2^0		1×2^{-1}	0×2^{-2}	1×2^{-3}

N. $1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} = (13.625)_{10}$

Scanned by CamScanner

2) $(29.3749)_{10} = (?)_2$

Solve upto four binary places.

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

2	29	
2	14	1
2	7	0
2	3	1
	1	1

$\therefore \boxed{(29)_{10} = (11101)_2}$

four binary places.

$0.3749 \times 2 = 0.7498 \quad 0$

$0.7498 \times 2 = 1.4996 \quad 1$

$0.4996 \times 2 = 0.9992 \quad 0$

$0.9992 \times 2 = 1.9984 \quad 1$

$\therefore (0.3749)_{10} = (.0101)_2$

$\therefore \boxed{(29.3749)_{10} = (11101.0101)_2}$

2) Octal to Binary

1) $(724)_8 = (?)_2$

~~724 is whole number, so divide by 8~~,

To convert from octal to binary, simply replace each Octal number by its binary equivalent from equivalent table.

$(7)_8 = (111)_2$

$(2)_8 = (010)_2$

$(4)_8 = (100)_2$

$\therefore (724)_8 = (111\ 010\ 100)_2$

$\boxed{(724)_8 = (111\ 010\ 100)_2}$

Gold

Scanned by CamScanner

2) $(2651.53)_8 = (?)_2$

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

$(2)_8 = (010)_2$

$(6)_8 = (110)_2$

$(5)_8 = (101)_2$

$(1)_8 = (001)_2$

$(3)_8 = (011)_2$

$\therefore (2651.53)_8$

$= (010\ 110\ 101\ 001.\ 101\ 011)_2$

3) Hexadecimal to Binary

This is similar to octal to binary conversion, only difference is, here the numbers are given in Hexadecimal & are represented by 4 binary digits by its binary equivalent from table.

$(ABC)_{16} = (?)_2$

$(A)_{16} = (1010)_2$

$(B)_{16} = (1011)_2$

$(C)_{16} = (1100)_2$

$\therefore (ABC)_{16} = (1010\ 1011\ 1100)_2$

2) $(.CDF)_{16} = (?)_2$

$(C)_{16} = (1100)_2$

$(D)_{16} = (1101)_2$

$(F)_{16} = (1111)_2$

$\therefore (.CDF)_{16} = (.1100\ 1101\ 1111)_2$

or $= (0.1100\ 1101\ 1111)_2$

Scanned by CamScanner

BCD Number System (Binary Coded Decimal):

The binary coded decimal is a numeric code in which each digit of decimal number is represented as a seperate group of binary bits. The BCD code is also called a weighted binary code 8-4-2-1, in which each BCD number is represented by 4-bit binary number. The 8-4-2-1 BCD ~~dod~~ code is associated with weights that is from left to right. The 3rd has a weight of 8 (2^3), the 2nd bit has a weight of 4 (2^2), first bit has a weight of 2 (2^1) & the 0th bit has a weight of 1 (2^0). The possible BCD number are tabulated below:

Decimal Number or Digit	BCD Code			
	3rd bit 8	2nd bit 4	1st bit 2	0th bit 1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Eg: 1) $(437)_{10} = (0100\ 0011\ 0111)_{BCD}$

2) $(6)_{10}$ is represented by $(\overset{8}{0}\overset{4}{1}\overset{2}{1}\overset{1}{0})_{BCD}$ i.e, we have 1's under 4 & 2 headers giving 4+2=6.

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Scanned by CamScanner

CONVERSION of ONE NUMBER SYSTEM to ANOTHER

(I) One Number to Decimal Number

1) Binary Number to Decimal Number

$2^3\ 2^2\ 2^1\ 2^0$
$a_3 a_2 a_1 a_0$

$(1011)_2 = (?)_{10}$

Using the polynomial expansion

$$a_n 2^n + a_{n-1} 2^{n-1} + \cdots + a_0 2^0 + a_{-1} 2^{-1} + \cdots + a_{-(m-1)} 2^{-(m-1)} + a_{-m} 2^{-m}$$

$n = 3,\ a_3 = 1,\ a_2 = 0,\ a_1 = 1,\ a_0 = 1$

$$(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$$

$$8 + 0 + 2 + 1$$

$$\boxed{(1011)_2 = (11)_{10}}$$

Manjunath Lakkanavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

$2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0\ 2^{-1}\ 2^{-2}\ 2^{-3}\ 2^{-4}$
$a_6 a_5 a_4 a_3 a_2 a_1 a_0\ a_{-1} a_{-2} a_{-3} a_{-4}$

$(1010110.1101)_2 = (?)_{10}$

$$(1 \times 2^6) + (0 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$$
$$+ (1 \times 2^{-1}) + (1 \times 2^{-2}) + (0 \times 2^{-3}) + (1 \times 2^{-4})$$

$$= 64 + 0 + 16 + 0 + 4 + 2 + 0 + 0.5 + 0.25 + \cancel{0.06}\ 0 + 0.0625$$

$$= 86.8125$$

$$\boxed{(1010110.1101)_2 = (86.8125)_{10}}$$

Scanned by CamScanner

OTHER

2) Octal to Decimal

$8^2\ 8^1\ 8^0$
$a_2\ a_1\ a_0$

1) $(243)_8 = (?)_{10}$

$a_n 8^n + a_{n-1} 8^{n-1} + \cdots + a_0 8^0 + a_{-1} 8^{-1} + \cdots + a_{-(m-1)} 8^{-(m-1)} + a_{-m} 8^{-m}$

$\therefore (2\times8^2) + (4\times8^1) + (3\times8^0)$

$= (2\times64) + (4\times8) + (3\times1)$

$= 128 + 32 + 3$

$= 163$

$\therefore \boxed{(243)_8 = (163)_{10}}$

$8^{-1}\ 8^{-2}\ 8^{-3}$
$a_0\ a_{-1}\ a_{-2}\ a_{-3}$

2) $(0.625)_8 = (?)_{10}$

$(0\times8^0) + (6\times8^{-1}) + (2\times8^{-2}) + (5\times8^{-3})$

$= 0 + 0.75 + 0.03125 + 0.009765625$

$= 0.7910$

$\therefore \boxed{(0.625)_8 = (0.7910)_{10}}$

3) Hexadecimal to Decimal

$16^2\ 16^1\ 16^0\ 16^{-1}\ 16^{-2}$

1) $(CAD.BF)_{16} = (?)_{10}$

$a_n 16^n + a_{n-1} 16^{n-1} + \cdots + a_0 16^0 + a_{-1} 16^{-1} + \cdots + a_{-(m-1)} 16^{-(m-1)} + a_{-m} 16^{-m}$

$\therefore (C\times16^2) + (A\times16^1) + (D\times16^0) + (B\times16^{-1}) + (F\times16^{-2})$

$= (12\times16^2) + (10\times16) + (13\times1) + (11\times16^{-1}) + (15\times16^{-2})$

$= 3245.74609375$

$\therefore \boxed{(CAD.BF)_{16} = (3245.746093)_{10}}$

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology
BANGALORE-560 054.

Scanned by CamScanner

2) $(9CF1)_{16} = (?)_{10}$

$16^3\ 16^2\ 16^1\ 16^0$

$\therefore (9\times16^3)+(C\times16^2)+(F\times16^1)+(1\times16^0)$

$= (9\times16^3)+(12\times16^2)+(15\times16^1)+(1\times16^0)$

$= 40177$

$\boxed{(9CF1)_{16} = (40177)_{10}}$

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology.
BANGALORE-560 054.

(II) One Number to Binary Number.

1) Decimal to Binary

fraction point

$(105.202)_{10} = (?)_2$

whole number

105 is before fraction point, so divide by 2

202 is after fraction point, so multiply by 2

fraction number

2	105	↓ Remainder
2	52	1
2	26	0
2	13	0
2	6	1
2	3	0
	1	1

Read the remainder at each stage upwards as

$\boxed{(105)_{10} = (1101001)_2}$

Integer

$0.202\times2 = 0.404 \quad 0$

$0.404\times2 = 0.808 \quad 0$

$0.808\times2 = 1.616 \quad 1$

$0.616\times2 = 1.232 \quad 1$

Read the integer at each stage downwards as

$\boxed{(0.202)_{10} = (.0011)_2}$

$\therefore \boxed{(105.202)_{10} = (1101001.0011)_2}$

Scanned by CamScanner

DeMorgan's Theorem

→ DeMorgan Suggested two theorems that form an important part of Boolean algebra.

1) $\overline{AB} = \overline{A} + \overline{B}$

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

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

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

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

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

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560054.

Scanned by CamScanner

1) Prove that $\overline{\overline{AB} + \overline{A} + AB} = 0$

Sol:

$\overline{\overline{AB} + \overline{A} + AB} = 0$

$= \overline{\overline{AB}} \cdot \overline{\overline{A}} \cdot \overline{AB}$ # Demorgan's thm.

$= AB \cdot A(\overline{A} + \overline{B})$

$= AB(\overline{A} + \overline{B})$ # $A \cdot A = A$.

$= AB\overline{A} + AB\overline{B}$ # $A\overline{A} = 0$, $B\overline{B} = 0$

$= 0 + 0$

$= 0 = RHS$

lify the following boolean expressions.

$\overline{AB + \overline{AC} + A\overline{B}C(AB+C)}$

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

$\overline{AC} + A\overline{B}C(AB+C)$

$B + (\overline{A} + \overline{C}) + A\overline{B}CAB + A\overline{B}C\overline{C}$

$AB + \overline{A} + \overline{C} + 0 + A\overline{B}C$ $\because B\overline{B} = 0$

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

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

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

$= AB + AC + \overline{A} + \overline{C}$

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

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

$= \overline{A} + A + B + \overline{C} = 1 + B + \overline{C} = 1$ //

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Scanned by CamScanner

(III) One Number to Octal Number

Page :
Date : / /

i) Decimal to Octal

→ The whole number i.e, before fraction point, so divide by 8

→ The fraction number i.e, after fraction point, so multiply by 8

i) $(847.951)_{10} = (?)_8$

fraction point

whole number

fraction number

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology.
BANGALORE-560 054.

		Remainder
8	847	
8	105	7
8	13	1
	1	5

$\therefore (847)_{10} = (1517)_8$

Integer

$0.951 \times 8 = 7.608$ 7

$0.608 \times 8 = 4.864$ 4

$0.864 \times 8 = 6.912$ 6

$0.912 \times 8 = 7.296$ 7

$\therefore (0.951)_{10} = (.7467)_8$

$$\boxed{(847.951)_{10} = (1517.7467)_8}$$

$(1 + \bar{A}B)AB$

AB

Gold

Scanned by CamScanner

Binary to Octal

→ The whole numbers, i.e., before fraction point has to be grouped into 3 digits from the right most bit (least significant bit)

& replace each group by the octal number equivalent.

→ The fraction number, i.e., after fraction point has to be grouped into 3 digits from left most bit (most significant bit)

& replace each group by the octal number equivalent

After grouping into 3 digits, if any group is standard (i.e., consisting of 3 digits) we have to make it standard or complete by appending 0's to it if (prefixing) group is whole number & for fraction number group, 0's to be postfixed

Manjunath Lakshmaiah
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054

1) $(101111)_2 = (?)_8$

← grouping direction

101 111

5 7

$\boxed{(101111)_2 = (57)_8}$

2) $(1010101)_2 = (?)_8$

← grouping direction

001 010 101

prefixed 0's.

1 2 5

$\boxed{(1010101)_2 = (125)_8}$

Scanned by CamScanner

(IV) One Number to Hexadecimal Number

1) Decimal to Hexadecimal

→ the whole number has to be divided by 16

→ The fraction number has to be multiplied by 16

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054

1) $(2146)_{10} = (?)_{16}$

		Remainder
16	2146	
16	134	2
	8	6

$\therefore (2146)_{10} = (862)_{16}$

2) $(2604.10546875)_{10} = (?)_{16}$

(2604 — whole number; .10546875 — fraction number; . — fraction point)

		Remainder
16	2604	
16	162	12 (C)
	10 (A)	2

$\therefore (2604)_{10} = (A2C)_{16}$

		Integer
$0.10546875 \times 16 = 1.6875$		1
$0.6875 \times 16 = 11.00$		11 (B)

$(0.10546875)_{10} = (1B)_{16}$

$(2604.10546875)_{10} = (A2C.1B)_{16}$

Scanned by CamScanner

2) Binary to Hexadecimal

→ The whole number has to be grouped into 4 digits from its LSB & replace each group by the hexadecimal number equivalent.

The fraction number has to be grouped into 4 digits from its MSB & replace each group by the hexadecimal number equivalent.

→ If after grouping into 4 digits, if any group is not standard (i.e., consisting of 4 digits) we have to make that group standard or complete by prefixing '0's to it if its a whole number & post fixing '0's to it if its a fraction number.

eg's: $(110110110.1001101)_2 = (?)_{16}$

whole number ← grouping direction.

$\underbrace{0011}_{\text{prefixed 0's}}\,0110\,1110$

$(3\ 6\ E)_{16}$

fraction number → grouping direction.

$1001\,1010$

↑ post fixed '0'

$(9\ A)_{16}$

$$\boxed{(110110110.1001101)_2 = (36E.9A)_{16}}$$

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Scanned by CamScanner

3) $\underbrace{1110}_{\text{whole number}} \cdot \underbrace{01101}_{\text{fraction number}}$ ← fraction point

Manjunath Lakshmaiah
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology
BANGALORE-560 054

$$(1110 \cdot 01101)_2 = (?)_8$$

001110 ← $\boxed{(1110)_2 = (16)_8}$

1 6

grouping direction →

011010

3 1 2 ← post fixed '0'

for fraction number $\boxed{(\cdot 01101)_2 = (\cdot 32)_8}$

$$\boxed{(1110 \cdot 01101)_2 = (16 \cdot 32)_8}$$

3) Hexadecimal to Octal

→ The hexadecimal number cannot be converted directly into octal number.

→ 1st hexadecimal number is converted into binary number

→ 2nd from binary number, we have to do binary to octal number conversion by grouping into 3 digits methods.

eg: (1) $(1D6 \cdot CA8)_{16} = (?)_8$

1st convert hexadecimal number into its binary equivalent just by replacing each number into its binary equivalent of 4 digits

$(1)_{16} = (0001)_2$	$(C)_{16} = (1100)_2$
$(D)_{16} = (1101)_2$	$(A)_{16} = (1010)_2$
$(6)_{16} = (0110)_2$	$(8)_{16} = (1000)_2$

$$(1AD6.CA8)_{16} = (0001\ 1101\ 0110.1100\ 1010\ 1000)_2$$

2nd convert this binary into octal number by grouping method.

$$(000\underbrace{1\ 1101\ 0110}_{\text{whole number}}.\underbrace{1100\ 1010\ 1000}_{\text{fraction number}})_2$$

fraction point

grouping direction

0001 1101 0110

0 7 2 6

$$(0001\ 1101\ 0110)_2 = (0726)_8$$

→ grouping direction

1100 1010 1000

6 2 5 0

$$(1100\ 1010\ 1000)_2 = (.6250)_8$$

$$\therefore (0001\ 1101\ 0110.1100\ 1010\ 1000)_2 = (726.625)_8$$

$$(1D6.CA8)_{16} = (726.625)_8$$

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Scanned by CamScanner

3) Octal to Hexadecimal

Page:
Date: / /

→ Octal number cannot be converted directly to hexadecimal.

→ 1st octal has to be converted into its binary equivalent

→ Next binary has to be converted into its hexadecimal equivalent by grouping into 4 digits method.

eg: $(615.25)_8 = (?)_{16}$

1st step: Octal has to be converted into binary just by replacing the each digit of Octal number into into its 3 bit binary equivalent.

$(615.25)_8$

$(110001\,01.010101)_2$

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

2nd step: The obtained binary number has to be divided into group of 4 digits as explained above in B to H conversion.

← grouping direction

0001|1000|1.101

prefixed 0's

(1 8 D)$_{16}$

Whole number

→ grouping direction

0101|0100

post fixed 0's

(5 4)$_{16}$

fraction number

$$\boxed{(615.25)_8 = (18D.54)_{16}}$$

Gold

Binary Arithmetic

Binary Addition:

Arithmetic addition of two binary numbers is similar to decimal arithmetic.

eg:
$$\begin{array}{r} 9 \\ +\ 7 \\ \hline 16 \end{array}$$

which results in an answer 16, where '6' is refered to as the sum and '1' is refereed to as the carry

→ Since binary numbers can have values of only 1 or 0 the four possible binary additions are as follows.

$$\begin{array}{r} 0 \\ +\ 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ +\ 1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\ 0 \\ \hline 01 \end{array} \qquad \begin{array}{r} 1 \\ +\ 1 \\ \hline 10 \end{array}$$

C S C S C S C S

S → Sum.
C → Carry.

→ The higher significant bit of this result is called Carry & lower significant bit is called Sum

→ Such a operation is known as Half addition.

→ The operation which performs addition of 3 bits (two significant bits and a previous carry) is called a full addition.

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Scanned by CamScanner

Full addition

Inputs			Outputs	
A	B	Cin	Cout	Sum (S)
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Addition Method:

1) Add bits column-wise starting from LSB with carry if any.
2) Put the sum at the bottom of the same column.
3) Put the carry, if any, on the top of next column.

Examples

1) Perform addition of $(11001100)_2$ & $(11011010)_2$

```
    1 1 1 1
+   1 1 0 0 1 1 0 0   Carry
    1 1 0 1 1 0 1 0   number 1
    ---------------   number 2
    1 1 0 1 0 0 1 1 0   Result (S)
```

2) $(10.1011)_2$ & $(11.011)_2$

```
    1 1 1 1 1
    1 0 . 1 0 1 1     (C)
+   1 1 . 0 1 1 0
    -----------
    1 1 0 . 0 0 0 1   (S)
```

Galit

3) Add $(28)_{10}$ and $(15)_{10}$ Using binary addition

Sol:

$$\begin{array}{r|l} 2 & 28 \\ \hline 2 & 14 \quad -0 \\ \hline 2 & 7 \quad -0 \\ \hline 2 & 3 \quad -1 \\ \hline & 1 \quad -1 \end{array}$$

$\therefore (28)_{10} = (11100)_2$

$$\begin{array}{r|l} 2 & 15 \\ \hline 2 & 7 \quad -1 \\ \hline 2 & 3 \quad -1 \\ \hline & 1 \quad -1 \end{array}$$

$\therefore (15)_{10} = (1111)_2$

$\therefore (11100)_2 + (1111)_2$

$$\begin{array}{r} 111 \\ 11100 \\ +\ 01111 \\ \hline 101011 \\ 2^5 2^4 2^3 2^2 2^1 2^0 \end{array}$$

(c) $(28)_{10}$
$+ (15)_{10}$

(s) $= (43)_{10}$

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Scanned by CamScanner

Complements

Page :
Date : / /

1's Complement Representation

→ The 1's complement of a binary number is the number that results when we change all 1's to zero's & the 0's to ones.

eg: 1's complement of $(11010100)_2$ is

00101011

2's Complement Representation

→ The 2's complement is the binary number that results when we add 1 to the 1's complement. It is given as

$$\boxed{\text{2's Complement} = \text{1's Complement} + 1}$$

eg: 2's complement of $(11000100)_2$

```
  11000100   ← Number
     1 1
  00111011   ← 1's Complement
+        1   ← add 1
----------
  00111100   ← 2's Complement
```

Manjunath Lakkenavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Scanned by CamScanner

Binary Subtraction Using 1's Complement Method

→ In a 1's complement subtraction, −ve number is represented in the 1's complement form & actual addition is performed to get the desired result.

eg: operation A − B is performed using following steps:

1) Take 1's complement of B.
2) Result → A + 1's Complement of B
3) If carry is generated, then the result is +ve and in the true form. Add carry to the result to get the final result.

If carry is not generated then the result is −ve in 1's complement form.

$(12)_{10} - (15)_{10}$

$(12)_{10} = (11100)_2$

$(15)_{10} = (01111)_2$

Manjunath Lakkammanavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

```
  01111      ← −ve number
  10000      ← Take 1's complement
1
+ 11100      ← retain +ve number
  10000      ← add 1's complement of −ve number
  101100
+      1     ← add carry
  01101      ← final result (Binary equivalent of (13)10)
```

carry generated

No carry generated

Verify

Hence

(2) -- Subtract $(111001)_2$ from $(101011)_2$ Using 2's complement method.

Sol^n. $(111001)_2 = (57)_{10}$.

$(101011)_2 = (43)_{10}$.

(2) $(15)_{10} - (28)_{10}$

$(15)_{10} = (01111)_2$

$(28)_{10} = (11100)_2$

Manjunath Lakkanavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology.
BANGALORE-560 054.

```
  11100   ← -ve number
  00011   ← 1's complement
  1111
+ 01111   ← retain the number
  00011   ← add 1's complement of -ve number
 -------
 10010    ← Result = Binary equivalent of (-13)10
```

No carry generated.

Verification 01101 ← 1's complement of $(13)_{10}$

(3) $10.0100 - 101.111$

```
  010.0100   ← retain
- 101.1110   ← replace by 1's complement
  1
  010.0100
+ 010.0001
 ---------
1 100.0101
```

no carry.

Hence ans is '-ve' & equal to 1's complement

Scanned by CamScanner

Let us consider additions of 3 and 6 in BCD

```
        11
        0110  → BCD of 6
+ 3     0011  → BCD of 3
  9   0 1001  → BCD of 9
      ↑
     carry
```

→ The addition is carried out as in normal binary addition & the sum is 1001 which is valid BCD code.

Case 2: Sum greater than 9 with carry 0.

addition of 6 & 8 in BCD.

```
  6     0110  → BCD of 6
+ 8     1000  → BCD of 8
 14   0 1110  → Invalid BCD number
      ↑
     carry      (1110) > 9
```

→ The sum 1110 is invalid BCD number. This was noticed because the sum of two digits exceeds 9. Whenever this occurs, the sum has to be corrected by the addition of six (0110) in the invalid BCD number, as shown below

```
+   10001   ← Add 2's complement of -ve no.
  → 101101  ← Result
```

carry generated & Ignore it

$\therefore (28)_{10} - (15)_{10} = (11100)_2 - (01111)_2$

$= (01101)_2 = (13)_{10}$.

Manjunath Lakkanavar
Assistant Professor
Dept. of Electronics and Communication
M.S. Ramaiah Institute of Technology
BANGALORE-560 054

Scanned by CamScanner

$$\begin{array}{r} 6 \\ +\,8 \\ \hline 14 \end{array} \qquad \begin{array}{r} 0110 \\ 1000 \\ \hline 1110 \end{array} \leftarrow \text{Invalid BCD}$$

$$\begin{array}{r} 1\,1\,1\, \\ 1110 \\ 0110 \\ \hline 10100 \end{array} \begin{array}{l} \leftarrow \text{Invalid BCD} \\ \leftarrow \text{add 6 (0110) to Invalid BCD.} \\ \leftarrow \text{Final result} \end{array}$$

↑ carry.

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560054.

$\underbrace{0001}_{1}\ \underbrace{0100}_{4}$ → BCD for 14

→ Decimal

→ After addition of 6, carry is produced in the second decimal position.

Case 3. Sum equals 9 or less with carry 1.

Addition of 8 & 9 in BCD.

$$\begin{array}{r} 8 \\ +\,9 \\ \hline 17 \end{array} \qquad \begin{array}{r} 1000 \\ 1001 \\ \hline 10001 \end{array} \begin{array}{l} \rightarrow \text{BCD of 8} \\ \rightarrow \text{BCD of 9} \end{array}$$

↑ carry

(0001 0001) → result

→ In this case, result (0001 0001) is valid BCD number, but it is incorrect

→ To get the correct BCD result correction factor of 6 has to be added to the LSB sum as shown below.

Scanned by CamScanner

$$
\begin{array}{rl}
\begin{array}{r} 8 \\ +\,9 \\ \hline 17 \end{array} & \begin{array}{rl}
1000 & \rightarrow \text{BCD for 8} \\
1001 & \rightarrow \text{BCD for 9} \\
\hline
0001\ 0001 & \rightarrow \text{Incorrect BCD result} \\
0000\ 0110 & \rightarrow \text{add 6 for correction} \\
\hline
\underbrace{0001}_{1}\underbrace{0111}_{7} & \rightarrow \text{BCD for 17.}
\end{array}
\end{array}
$$

Procedure of BCD addition.

Going through these three cases of BCD addition we can summarize the BCD addition procedure as follows.

1) Add two BCD numbers using ordinary binary addition.
2) If four bit sum is equal to or less than 9, no correction is needed. The sum is in proper BCD form.
3) If the four-bit sum is greater than 9 or if a carry is generated from the four bit sum, the sum is invalid.
4) To correct the invalid sum, add 6 $(0110)_2$ to the four bit sum. If a carry results from this addition, add it to the next higher-order BCD digit.

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Scanned by CamScanner

1) 34 + 27

34	BCD for 34	0011 0100
+ 27	BCD for 27	0010 0111
61	Invalid BCD	0101 1011
	Add '6' for correction	+ 0110
	BCD for 61 →	0110 0001

2) 48 + 58

48	0100	1000	
+ 58	0101	1000	
106			
	1010	1 0000	→ carry generated
greater than '9' →	0110	0110	→ adding '6' to the sum
'6' is added →	0001	0000	0110 → Corrected sum
	1	0	6

3) 589 + 299

589	0101	1000	1001
+ 299	0010	1001	1001
888			
	1000	1 0010	1 0010
+		0110	0110
	1000	1000	1000
	8	8	8

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Scanned by CamScanner

BCD Subtraction Using 9's Complement

9's Complement Representation

9's Complement of a single digit number 'n' is obtained as 9-n.

eg: 9's complement of 4 is 9-4=5

→ 9's complement of a two digit number, mn is obtained as 99-mn. for eg: 9's complement of 32 is 99-32 = 67.

→ Similarly 9's complement of a number of any size can be computed.

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

10's Complement Representation

10's Complement = 9's Complement + 1

∴ 10's Complement of 4 is 5 + 1 = 6

↑ 9's comp. of 4 (i.e, 9-4=5)

→ The 9's complement of a decimal number is found by subtracting each digit in the no. from 9.

9	9	9	9	9	9	9	9	9	9
0	1	2	3	4	5	6	7	8	9 → Decimal digit
9	8	7	6	5	4	3	2	1	0 → 9's Complement

Scanned by CamScanner

DIGITAL LOGIC

Page:
Date / /
...nath Lakkar...
...istant Professor
Dept. ... ...onics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

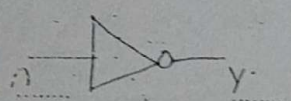
Logic Gates:

→ Logic gates are the basic elements that make up a digital system.

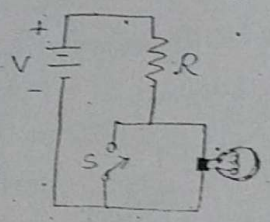
→ The electronic gate is a ckt that is able to operate on a binary i/p numbers in order to perform a particular logical operation/function.

→ The types of gates available are the NOT, AND, OR, NAND, NOR, Exclusive-OR and Exclusive-NOR.

1) NOT gate (Inverter) : The o/p is a complement of i/p.



Symbol



I/P	O/P
Switch open (Low)	Lamp ON (High)
Switch close (High)	Lamp OFF (Low)

Input	Output
A	Y
0	1
1	0

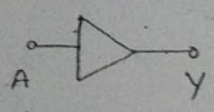
Truth Table

$$\boxed{Y = \overline{A}}$$

Boolean Expression

Application: Used to complement (Invert) digital signal.

2) Buffer : The o/p is same as i/p.



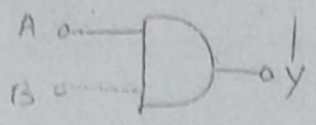
$$\boxed{Y = A}$$

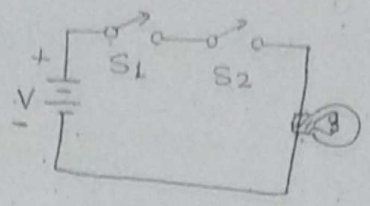
Input A	Output Y
0	0
1	0

Application: It is used to increase output driving capacity.

Scanned by CamScanner

3) AND gate: The o/p is high only when all i/ps are high

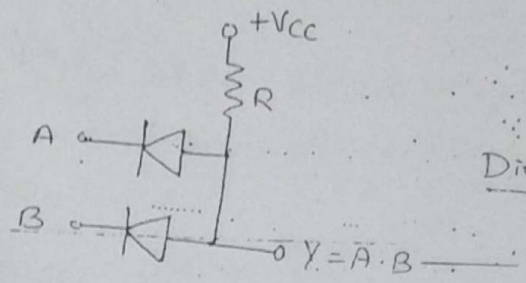




Input		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

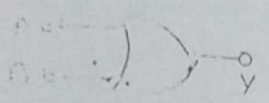
$Y = A \cdot B$

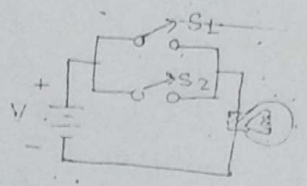
Application: Used to implement logical AND operation.



Diode Equivalent ckt

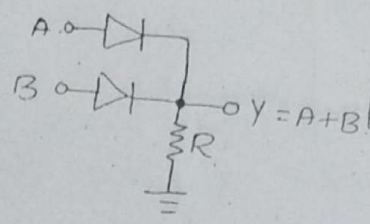
gate: The o/p is high when any of the i/ps is high





Input		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

$Y = A + B$

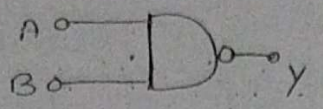


Application: Used to implement logical OR operation.

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Scanned by CamScanner

5) NAND gate: The o/p is high only when one of the i/ps is low.

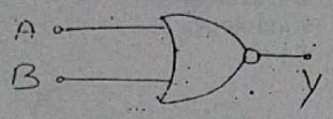


$$Y = \overline{A \cdot B}$$

Input		Output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Application: It can be used to implement any digital ckt.

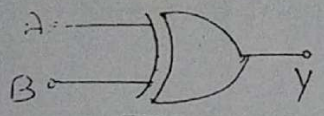
6) NOR gate: The o/p is high when all the i/ps are low



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

Input		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

7) Exclusive-OR (Ex-OR) gate: The o/p is high only when odd number of i/ps are high



XOR

$$Y = A \oplus B$$

OR

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

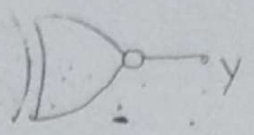
Input		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Application: It is used to implement magnitude comparator, gray code converter, adder/subtractor ckts, parity generator, modulo-2 adder etc.

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560054

Scanned by CamScanner

8) Exclusive NOR (Ex-NOR) gate: The o/p is high only when number of ones at the input or all i/ps are high.



$$Y = \overline{A \oplus B}$$

OR

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

Input		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Application: It is used to implement even parity generator, comparator, even parity checker etc.

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology.
BANGALORE-560 054.

Universal Gates

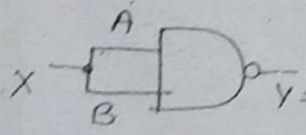
NAND & NOR gates are known as universal gates, since any logic function can be implemented using NOR gates

NAND Gate

→ The NAND Gate can be used to generate the NOT function, the AND, OR & NOR function's

NOT Function:—

→ An inverter can be made from NAND gate by connecting all of the inputs together & creating, in effect, a single common input as shown below for two i/p gate

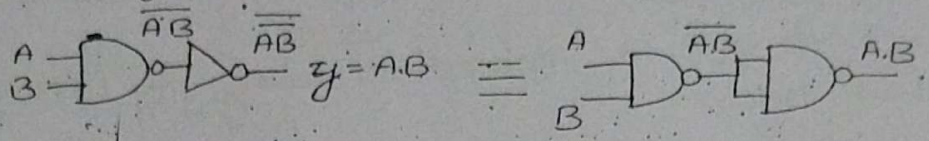


$$Y = \overline{AB} = \overline{XX} = \overline{X} + \overline{X} = \overline{X}$$

	A	B	$\overline{AB}$	
X=0	0	0	1	Y=1
	0	1	1	
	1	0	1	
X=1	1	1	0	Y=0

AND function

→ An AND function can be generated using only NAND gates. It is generated by simply inverting o/p of NAND gate i.e., $\overline{\overline{A.B}} = A.B$.



A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

≡

A	B	$\overline{A.B}$	$\overline{\overline{A.B}}$
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

OR function

Manjunath Lakkanavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

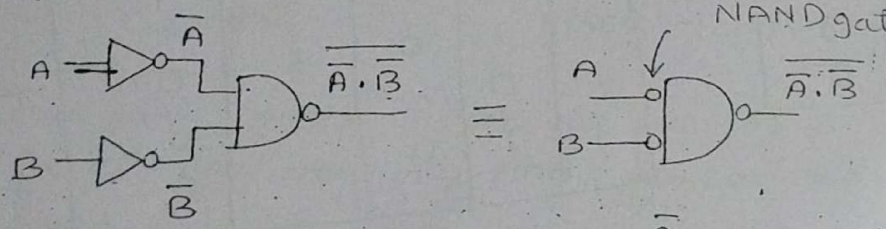
$$Y = A + B$$

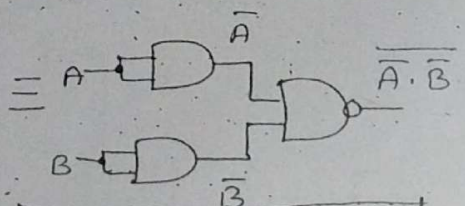
$$= \overline{\overline{A}} + \overline{\overline{B}} \qquad \because [\overline{\overline{A}} = A]$$

$$\boxed{Y = \overline{\overline{A} \cdot \overline{B}}}$$ → De Morgan's Theorem 1.

The above eqn can be implemented using only NAND gates as shown below.

Bubble at the i/p of NAND gate indicates inverted i/p.





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

$$= A + B$$

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

≡

A	B	$\overline{\overline{A} \cdot \overline{B}}$	$\overline{\overline{\overline{A} \cdot \overline{B}}}$
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	1

Gate

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

NOR function

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

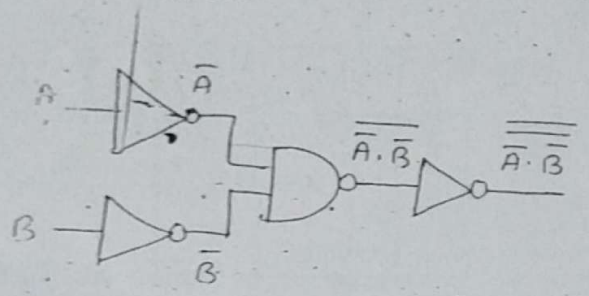
$$= \overline{A}\cdot\overline{B}$$

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

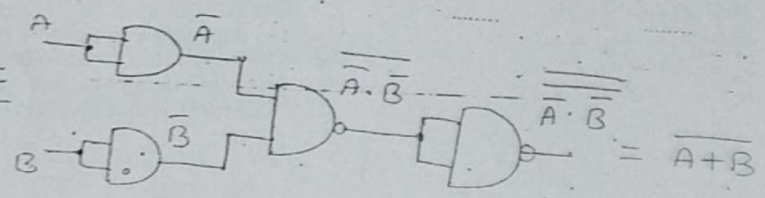
→ DeMorgan's Theorem 2

$[\overline{\overline{A}} = A]$

The above eqn is implemented using only NAND gates as shown below.



≡



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

≡

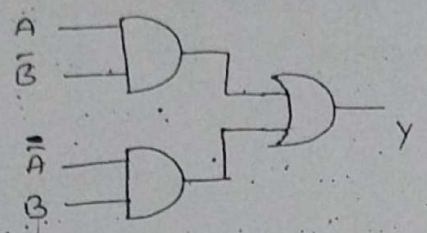
A	B	$\overline{A}\cdot\overline{B}$	$\overline{\overline{A}\cdot\overline{B}}$	$\overline{\overline{\overline{A}\cdot\overline{B}}}$
0	0	1	0	1
0	1	0	1	0
1	0	0	1	0
1	1	0	1	0

EX-OR gate Using NAND gates

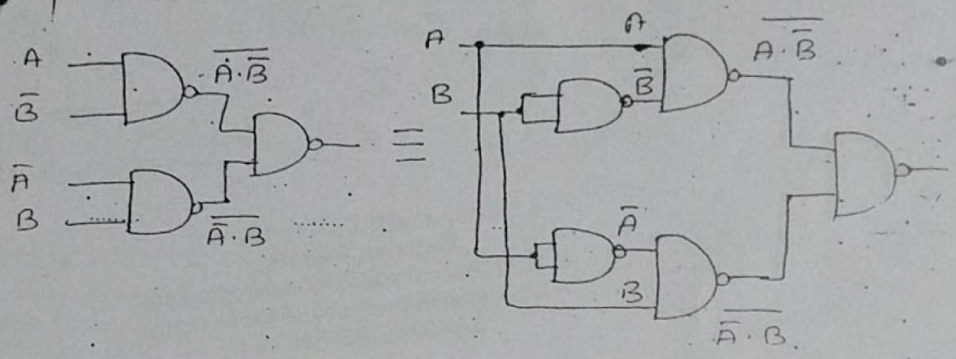
SHEKAR'S Victory
Page :
Date : / /

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

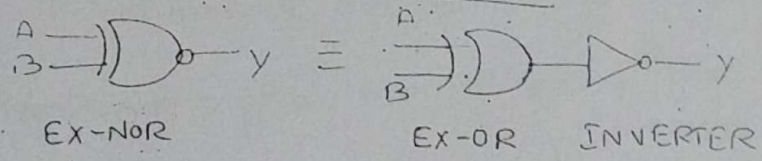
Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.



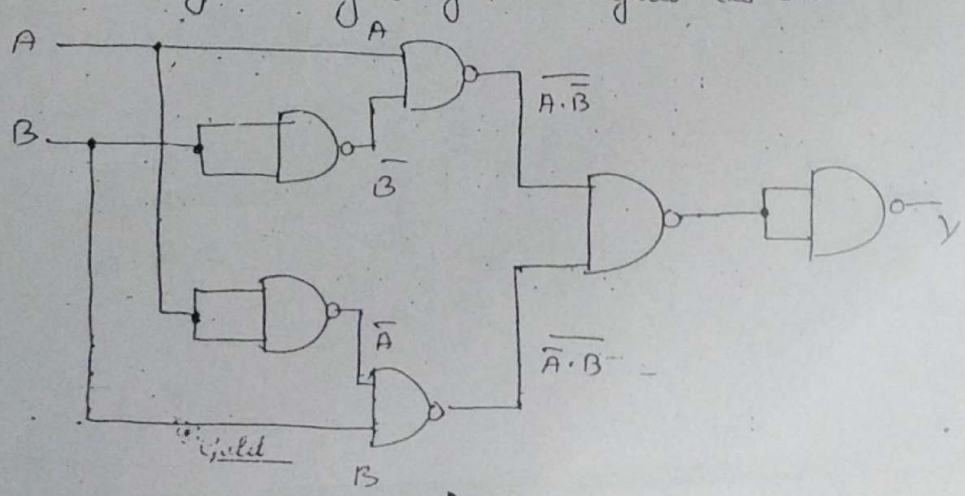
We can implement AND-OR logic by using NAND-NAND logic as shown below.



EX-NOR gate Using NAND gates.



Thus using result in previous example we have EX-NOR gate using only NAND gates as shown below.

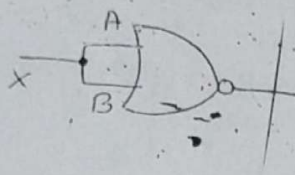


Scanned by CamScanner

(II) NOR Gate

→ Similar to NAND gate, the NOR gate is also a universal gate, since it can be used to generate the AND, OR & NAND functions.

NOT Gate -



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

$= \overline{X} \cdot \overline{X}$

$= \overline{X}$

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

X [0 0 1] Y = 1

X [1 1 0] Y = 0

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

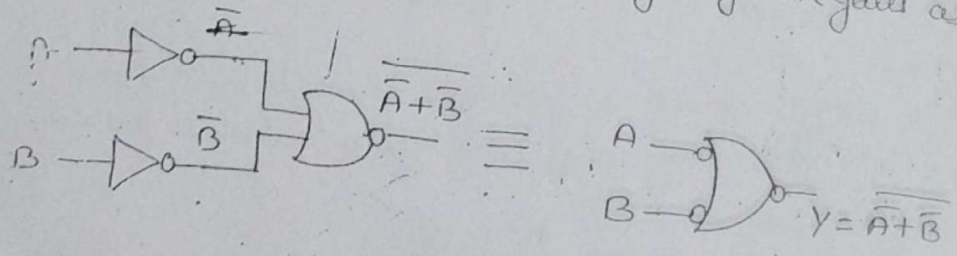
Function

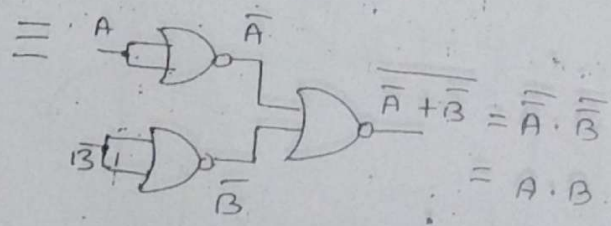
$Y = A \cdot B$

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

$Y = \overline{A + \overline{B}}$ → De-Morgan's Theorem 2.

The above eqn is implemented using only NOR gates as shown below.





Scanned by CamScanner

A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

≡

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

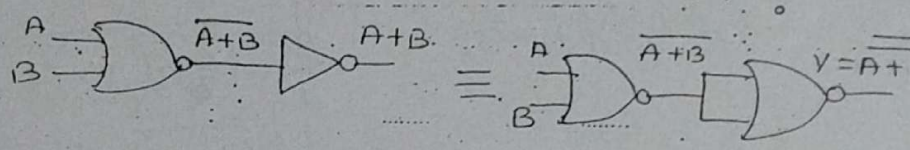
Manjunath Lakkammavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

OR function

$$Y = A+B$$

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

It is generated by simply inverting O/P of NOR gate i.e., $\overline{\overline{A+B}} = A+B$.



A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

≡

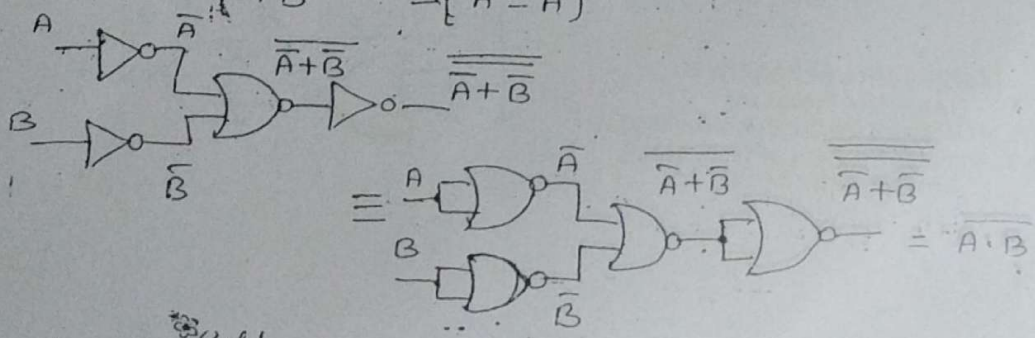
A	B	$\overline{A+B}$	$\overline{\overline{A+B}}$
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	1

NAND function

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

$$= \bar{A}+\bar{B} \quad \rightarrow \text{Demorgan's Theorem 1.}$$

$$= \overline{\overline{\bar{A}+\bar{B}}} \quad \rightarrow [\bar{\bar{A}} = A]$$



Gold

A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

$\equiv$

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

$=$

EX-NOR gate using NOR gate

$$Y = AB + \bar{A}\bar{B} = \overline{A\bar{B} + \bar{A}B}$$

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

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

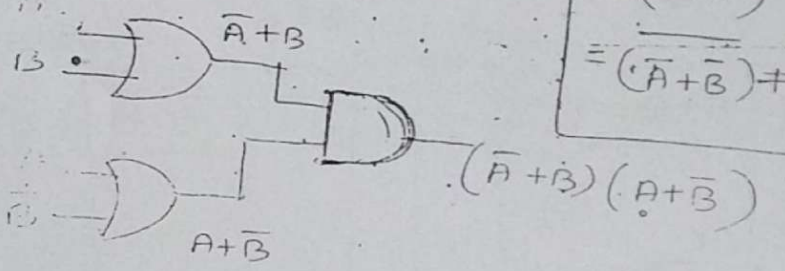
OR

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

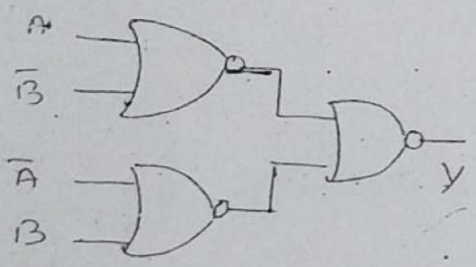
$$= (\overline{AB}) \cdot (\overline{\bar{A}\bar{B}})$$

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

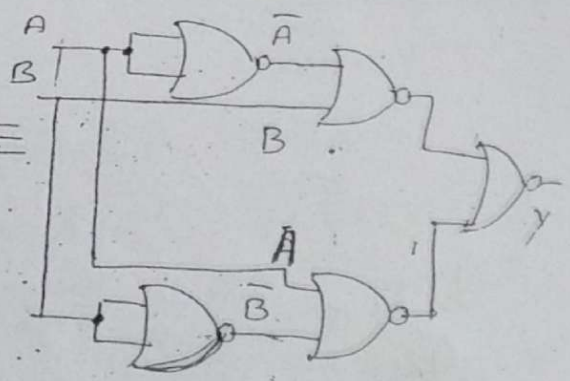
$$= \overline{(\bar{A}+\bar{B})} + \overline{(A+B)}$$



We can implement OR-AND logic by NOR-NOR logic as shown below.



$\equiv$



Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Scanned by CamScanner

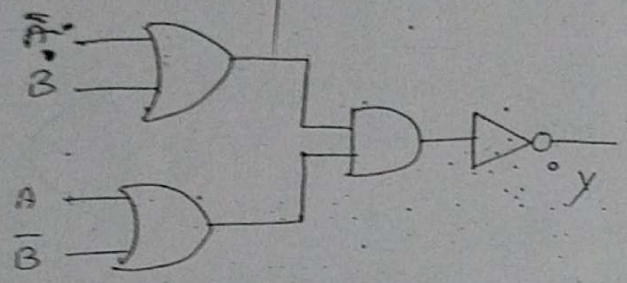
EX-OR gate using NOR gate

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

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

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

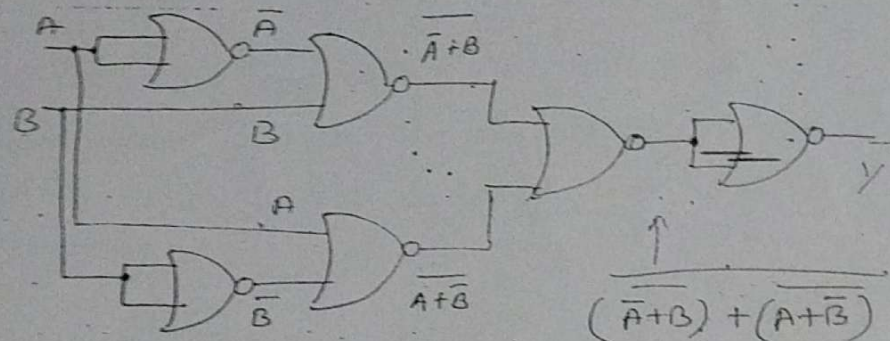
$$Y = \overline{(\bar{A}+B)(A+\bar{B})} = \overline{(\bar{A}+B)} + \overline{(A+\bar{B})}$$



$\overline{(A+B)}$

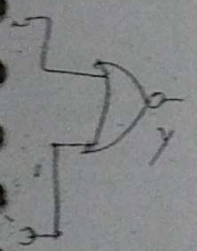
$\overline{(A+B)}$

$\equiv$



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

2 logic



Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Scanned by CamScanner

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Boolean Algebra

An algebraic system for a systematic treatment of logic is called Boolean Algebra.

→ It is a system of mathematical logic.

Variable: The symbol which represent an arbitrary elements of an boolean algebra is known as Variable.

eg: $Y = AB + C$, A, B & C are variables which can take values either 1 or 0 value. & f also can have 1 or 0 value.

Constant: In expression $Y = A + 1$, A is variable & have value 1 or 0. The second term has a value 1. So 1 is a constant here. The con term may be 1 or 0.

Complement: The complement of a variable is represent by a "bar" over the letter.

eg: Complement of A is $\overline{A}$

So if $A = 1$, $\overline{A} = 0$ & if $A = 0$, $\overline{A} = 1$

Sometimes prime symbol (') is used to denote the complement. eg: A'

Scanned by CamScanner

Boolean function: Boolean expressions are constructed by connecting the boolean constants & variables with the Boolean operations.

Boolean expressions are also called as Boolean formulas.

eg: $(A+\bar{B})C$ is used to describe the function f

$$f(A,B,C) = (A+\bar{B})C \text{ or } f = (A+\bar{B})C$$

Law of Boolean Algebra

3 of the basic laws of Boolean algebra are the same as in ordinary algebra:

- Commutative law
- Associative law
- Distributive law.

Dr. Ramanath Lalit
Associate Professor
Dept of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology
BANGALORE-560 054.

Commutative Law:

Law 1: $A+B = B+A$: States that the order in which the variable are ORed makes no difference in the o/p. (A OR B is same as B OR A).

Law 2: $A \cdot B = B \cdot A$: States that the order in which the variables are ANDed makes no difference in the o/p. (ie, A AND B is same as B AND A.)

Commutative law can be extented to any no of variables

eg: $A+B+C = B+A+C$ and $ABCD = BACD = BADC = ABDC$ so on.

Associative Law

Law 1 : $A+(B+C)=(A+B)+C$: States that ORing of several variables, the result is the same regardless of the grouping of the variables.

A ORed with B ORC = A OR B ORed with C.

Law 2 : $(AB)C = A(BC)$: States that ANDing of several variables, the result is same regardless of the grouping of variables.

A AND B ANDed with C = A ANDed with

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Distributive Law.

Law: $A(B+C) = AB+AC$: States the ORing several variables & ANDing the result with a single variable is equivalent to ANDing the result with a single variable with each of the several variables & then ORing the products.

→ It is important to note that the distributive property is often used in reverse i.e., given $AB+AC$, we replace it by its equivalent, $A(B+C)$. As in ordinary algebra, this process is called Factoring. We factored 'A' out of expression $B+C$.

Scanned by CamScanner

Rules of Boolean Algebra.

The rules of Boolean algebra are as follows:

OR Operation

Rule 1:

$$0 + \boxed{0} = \boxed{0}$$
$$0 + \boxed{1} = \boxed{1}$$
$$\Rightarrow \boxed{\begin{array}{l} 0 + A = A \\ A + 0 = A \end{array}} \text{ or}$$

Rule 2:

$$1 + \boxed{0} = 1$$
$$1 + \boxed{1} = 1$$
$$\Rightarrow \boxed{\begin{array}{l} 1 + A = 1 \\ A + 1 = 1 \end{array}} \text{ or}$$

Rule 3:

$$\boxed{0} + \boxed{0} = \boxed{0}$$
$$\boxed{1} + \boxed{1} = \boxed{1}$$
$$\Rightarrow \boxed{A + A = A}$$

Rule 4:

$$\boxed{0} + \boxed{1} = 1$$
$$\boxed{1} + \boxed{0} = 1$$
$$\Rightarrow \boxed{\begin{array}{l} A + \bar{A} = 1 \\ \bar{A} + A = 1 \end{array}} \text{ or}$$

Identities of Boolean Addition

1.) $A + 0 = A$

2) $A + 1 = 1$

3) $A + A = A$

4) $A + \bar{A} = 1$

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Scanned by CamScanner

AND Operation

Rule 5:

$$0 \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{matrix} 0 \\ 0 \end{matrix} \Rightarrow \boxed{0 \cdot A = 0 \text{ or } A \cdot 0 = 0}$$

Rule 6:

$$1 \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow \boxed{1 \cdot A = A \text{ or } A \cdot 1 = A}$$

Rule 7:

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow \boxed{A \cdot A = A}$$

Rule 8:

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{matrix} 0 \\ 0 \end{matrix} \Rightarrow \boxed{A \cdot \overline{A} = 0 \text{ or } \overline{A} \cdot A = 0}$$

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

NOT Operation

Rule 9:

$$\begin{bmatrix} \overline{\overline{0}} \\ \overline{\overline{1}} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow \boxed{\overline{\overline{A}} = A}$$

Identities of Boolean Multiplication

1) $A \cdot 0 = 0$
2) $A \cdot 1 = A$
3) $A \cdot A = A$
4) $A \cdot \overline{A} = 0$

Gold

Scanned by CamScanner

Rule 10: $\boxed{A + AB = A}$

Proof: $A + AB = A(1+B)$ Distributive law

$= A(1)$ Rule: 2

$= A$ Rule: 6

Rule 11: $\boxed{A + \bar{A}B = A + B}$

Proof: $A + \bar{A}\cdot B = A + AB + \bar{A}B$ Rule: 10

$= A + B(A+\bar{A})$ Distributive law

$= A + B(1)$ Rule: 4

$= A + B$ Rule: 6

Ru

$\boxed{(A+B)(A+C) = A + BC}$

Proof: $(A+B)(A+C)$

$= AA + AC + AB + BC$ Distributive law

$= A + AC + AB + BC$ Rule: 7

$= A(1+C+B) + BC$ Distributive law

$= A + BC$ Rule: 2

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Scanned by CamScanner

ii) $\overline{\overline{A\bar{B} + ABC} + A(B + A\bar{B})}$

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

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

$= \overline{\bar{A} + \overline{(\bar{B} + C)} + AB + AA}$

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

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

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

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

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

$= \overline{1 + B\bar{C}}$

$= \overline{1}$

$= 0$

OR :

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

$= 1 + \overline{\bar{B} + C} + AB$

$= \overline{1}$

$= 0$

Manjunath Lakkannavar
Assistant Professor
Dept of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Scanned by CamScanner

Realize the following expression using only NAND gate.

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

Sol:

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

$$= A\bar{A} + AB + AC + \bar{B}\bar{A} + B\bar{B} + \bar{B}C + \bar{A}C + BC + CC$$

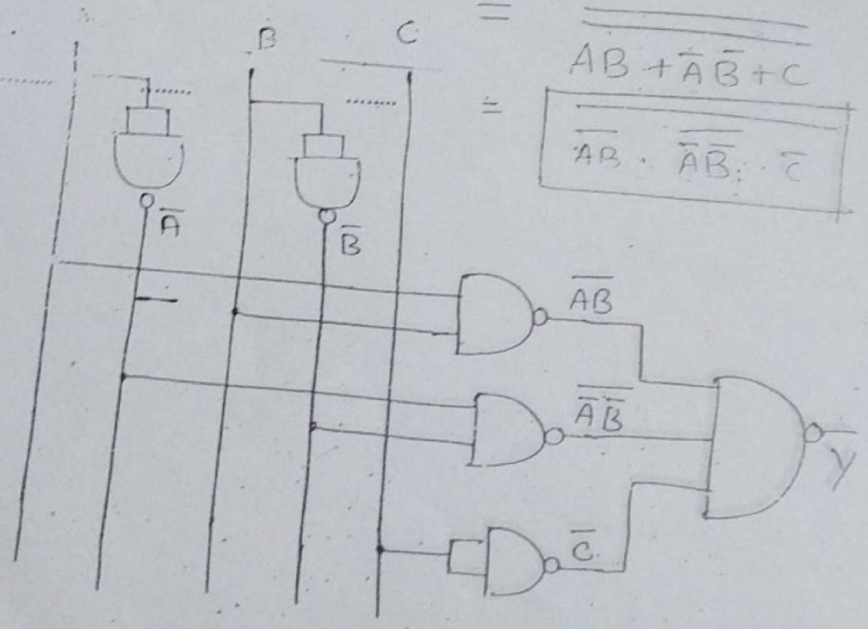
$$= 0 + AB + AC + \bar{A}\bar{B} + 0 + \bar{B}C + \bar{A}C + BC + C$$

$$= AB + \bar{A}\bar{B} + C(A+\bar{B}+\bar{A}+B+1)$$

$$Y = \boxed{AB + \bar{A}\bar{B} + C} \quad (\because A+1=1)$$

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

$$= \boxed{\overline{\overline{AB} \cdot \overline{\bar{A}\bar{B}} \cdot \bar{C}}}$$



Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology
BANGALORE-560 054.

Scanned by CamScanner

Page :
Date : / /

Simplify & realize the following using NAND gates $A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B} + \bar{A}\bar{C}$

Sol:

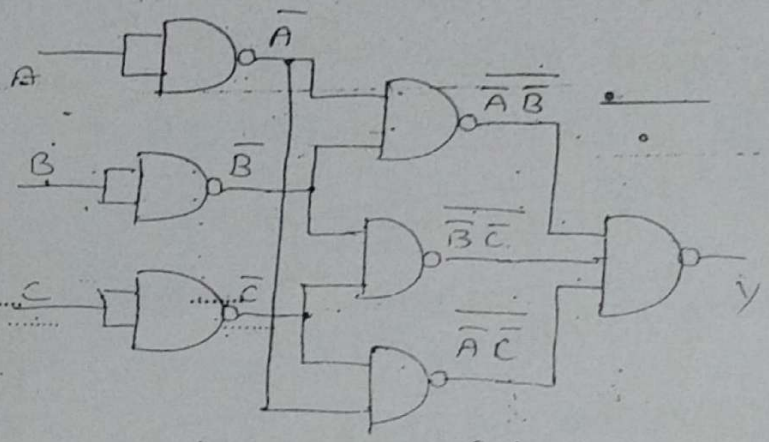
$$Y = A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B} + \bar{A}\bar{C}$$

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

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

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

$$Y = \boxed{\overline{\overline{\bar{B}\bar{C}} \cdot \overline{\bar{A}\bar{B}} \cdot \overline{\bar{A}\bar{C}}}}$$



Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology
BANGALORE-560 054.

Gold

Simplify & Realize the following Boolean expressions using logic gates.

(a) $Y = \overline{\overline{BC} + \overline{AD}(\overline{AB} + \overline{CD})}$

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

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

Manjunath L.B. Navar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology.
BANGALORE-560 054.

Sol^n

(a) $Y = \overline{\overline{BC} + \overline{AD}(\overline{AB} + \overline{CD})}$

Applying De Morgan's Thm to the RHS, treating $x = \overline{BC}$ & $y = \overline{AD}(\overline{AB} + \overline{CD})$ we have,

$$\overline{x+y} = \overline{x} \cdot \overline{y}$$

$$Y = (\overline{\overline{BC}})\left(\overline{\overline{AD}(\overline{AB} + \overline{CD})}\right)$$

$$= BC\left(\overline{\overline{AD} \cdot (\overline{\overline{AB} + \overline{CD}})}\right)$$

Applying De Morgan's Thm to $\left(\overline{\overline{AD}(\overline{AB} + \overline{CD})}\right)$ treating $x = \overline{AD}$ & $y = \overline{AB} + \overline{CD}$, we have.

$$Y = BC\left(\overline{\overline{AD}} + \overline{(\overline{AB} + \overline{CD})}\right)$$

$$= BC\left(AD + \overline{\overline{AB} + \overline{CD}}\right)$$

Now applying De Morgan's thm to $\overline{\overline{AB} + \overline{CD}}$ we have

$$Y = BC\left\{AD + (\overline{\overline{AB}})(\overline{\overline{CD}})\right\}$$

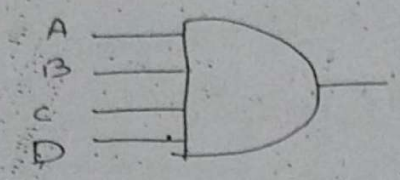
$$= BC\{AD + ABCD\}$$

Scanned by CamScanner

$\therefore Y = ABC\overline{D} + ABCD$

$$\boxed{Y = ABCD}$$

This can be implemented using a four i/p AND gate as shown below.



$Y = ABCD$

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

$= A\overline{A} + AB + A\overline{C} + \overline{A}B\overline{C} + BB\overline{C} + B\overline{C}\,\overline{C}$

$= 0 + AB + A\overline{C} + \overline{A}BC + B\overline{C} + B\overline{C}$

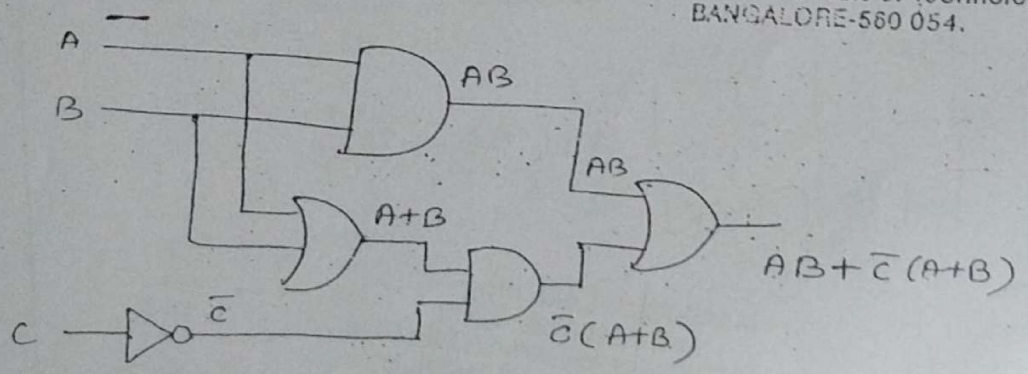
$= AB + A\overline{C} + \overline{A}B\overline{C} + B\overline{C}$

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

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

$$\boxed{Y = AB + \overline{C}(A+B)}$$

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.



Gold

Scanned by CamScanner

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

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

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

$= ABC + BBC + BCC + AC + BC + CC$

$= ABC + BC + BC + AC + BC + C$

$= ABC + BC + AC + C$

$= BC(A+1) + C(A+1)$

$= BC + C$

$= C(B+1)$

$\boxed{Y = C}$

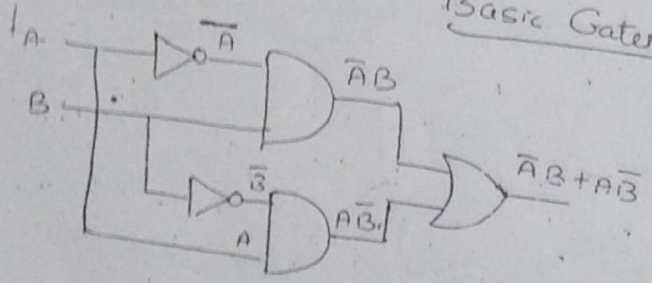
C —— $Y = C$

Realize $Y = \bar{A}B + A\bar{B}$ using (i) basic gates, (ii) NAND gates only (iii) NOR gates only

Soln

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

Basic Gates Only



Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Scanned by CamScanner

CHEKAR'S Victory
Page :
Date : / /

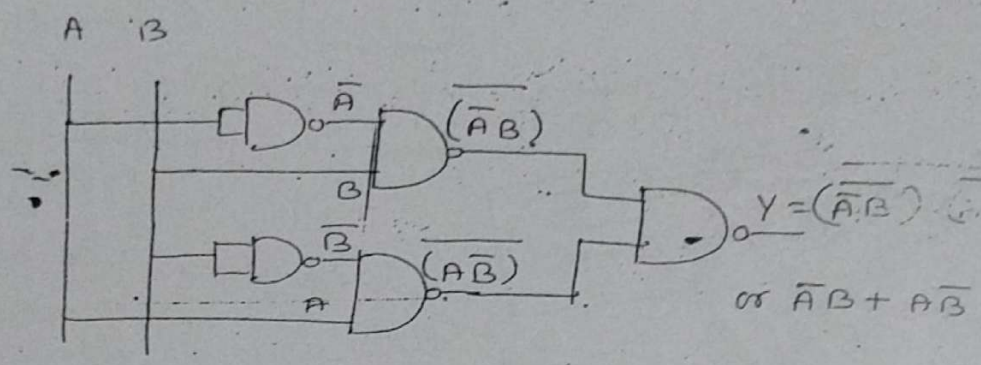
NAND gates Only

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

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

Take double complement

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



Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology.
BANGALORE-560 054.

NOR Gates Only

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

$$= \overline{\overline{\bar{A}B + A\bar{B}}}$$ → Take complement

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

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

$$= \overline{(A+\bar{B})(\bar{A}+B)}$$ → Take one more complement

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

$$Y = \boxed{\overline{\overline{(A+\bar{B}) + (\bar{A}+B)}}}$$ → Put double complement to retain the logic.

gates only

Gold

Scanned by CamScanner

OR-else

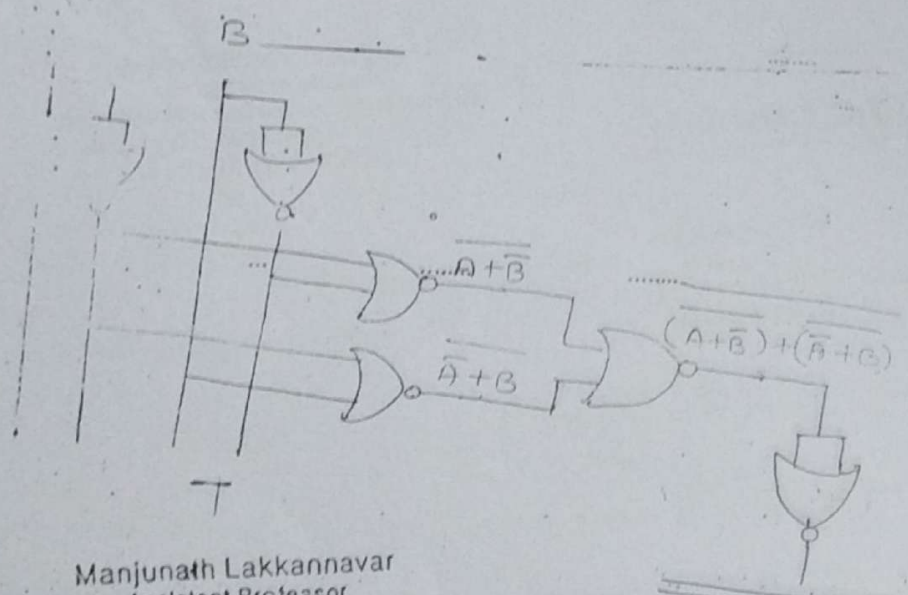
$$Y = \bar{A}B + A\bar{B}$$

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

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

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

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



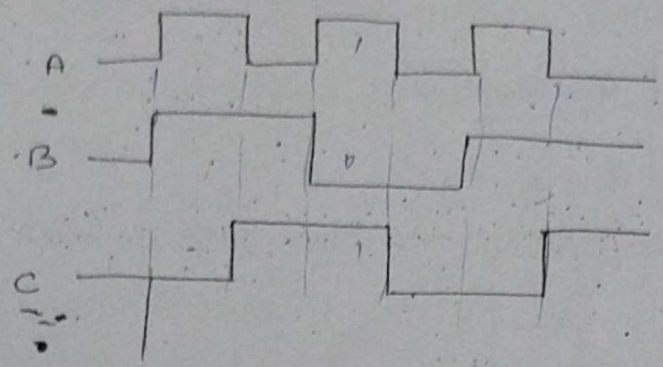
$$Y = \overline{\overline{(A+\bar{B})+(\bar{A}+B)}}$$

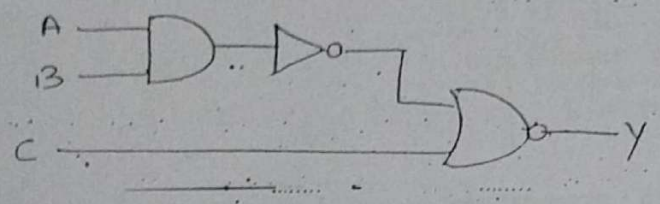
Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Scanned by CamScanner

Page :
Date : / /

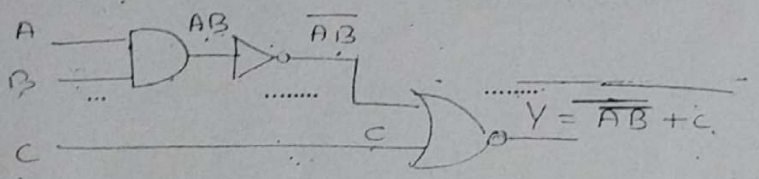
Draw the o/p waveforms of the logic ckt shown for the following i/p waveforms.





Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology
BANGALORE-560 054.

Soln



$$Y = \overline{\overline{AB} + C}$$

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

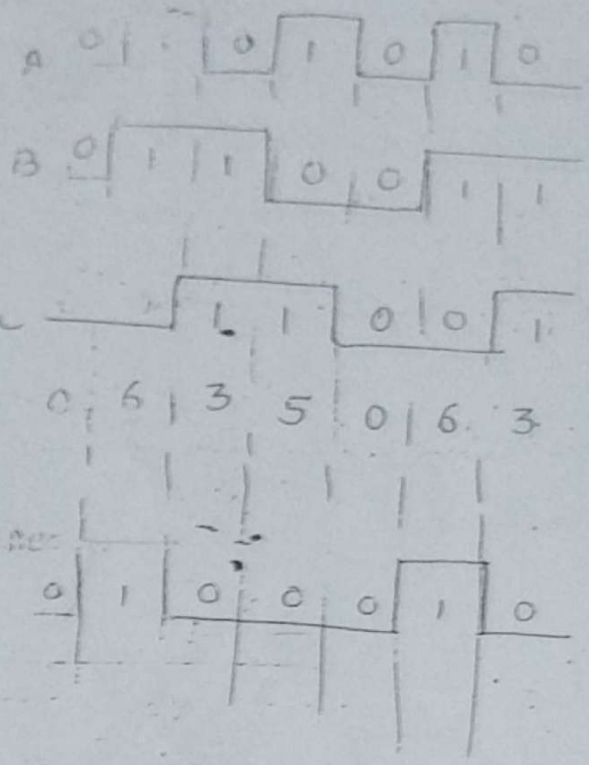
$$= \overline{\overline{A}} \cdot \overline{\overline{B}} \cdot \overline{C}$$

or $\boxed{AB\overline{C}}$

Row	A	B	C	AB	$\overline{C}$	$AB\overline{C}$
0	0	0	0	0	1	0
1	0	0	1	0	0	0
2	0	1	0	0	1	0
3	0	1	1	0	0	0
4	1	0	0	0	1	0
5	1	0	1	0	0	0
6	1	1	0	1	1	1
7	1	1	1	1	0	0

Gold

Scanned by CamScanner



Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
R. Ramaiah Institute of Technology,
BANGALORE-560 054.

→ The
One
Sum

→ Th
an

→ Th
a

→ The
bits

Scanned by CamScanner

Half Adder & Full Adder

→ Digital Computers perform various operations

→ The most basic operation is addition of two binary digits.

→ This simple addition consists of 4 possible operations

$0+0=0$

$0+1=1$

$1+0=1$

$1+1=10_2$

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

→ The 1st 3 operations produce a sum whose length one digit, but when the last operation is performed sum is two digits.

→ The higher significant bit of this is called Carry and lower significant bit is called Sum

→ The logic ckt which performs this operation is called a Half-Adder

→ The ckt which performs addition of 3 bits (two significant bits & a previous carry) is a full adder.

Gold

Scanned by CamScanner

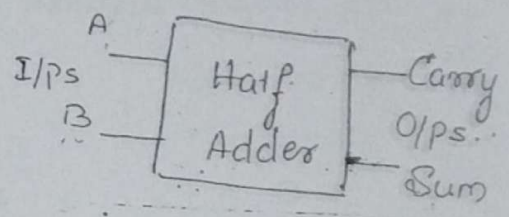
Half Adder

The half adder operation needs two binary i/ps & produce two binary o/ps.

Truth Table

Inputs		Outputs	
A	B	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Block Diagram



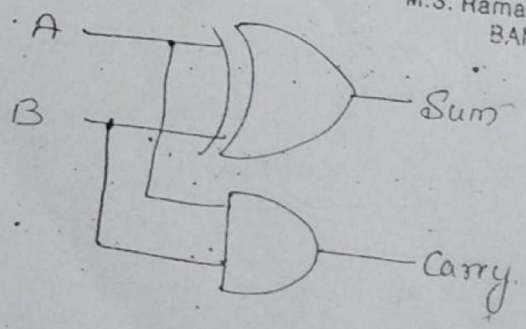
From the truth table, we have

$$Carry = A \cdot B$$

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

Logic Diagram

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.



Scanned by CamScanner

Page :
Date : / /

Half adder using NAND gates only

i/ps

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

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

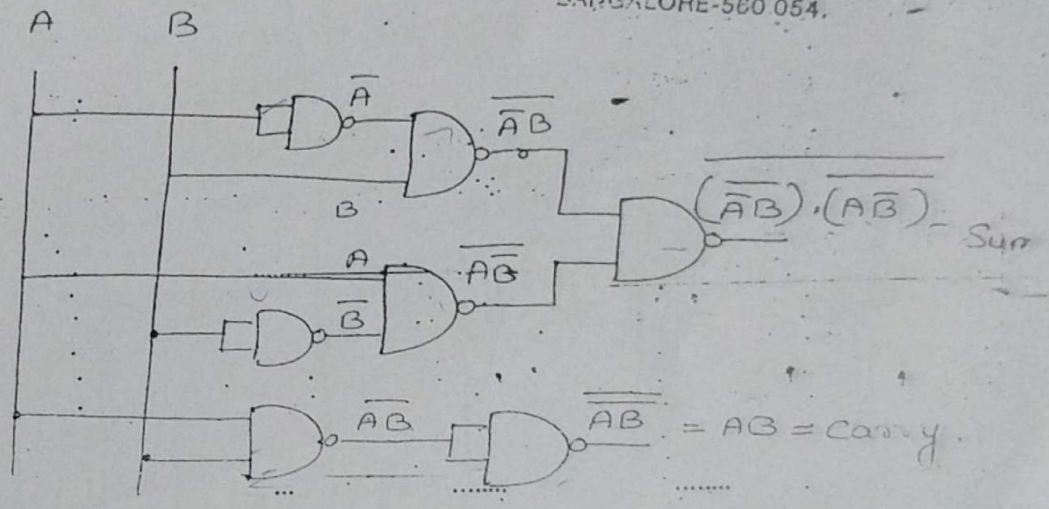
$$= \overline{\overline{(\bar{A}B)} \cdot \overline{(A\bar{B})}}$$

$$\text{Carry} = AB$$

$$= \overline{\overline{AB}}$$

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Carry o/ps
Sum



Half adder using NOR gates only

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

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

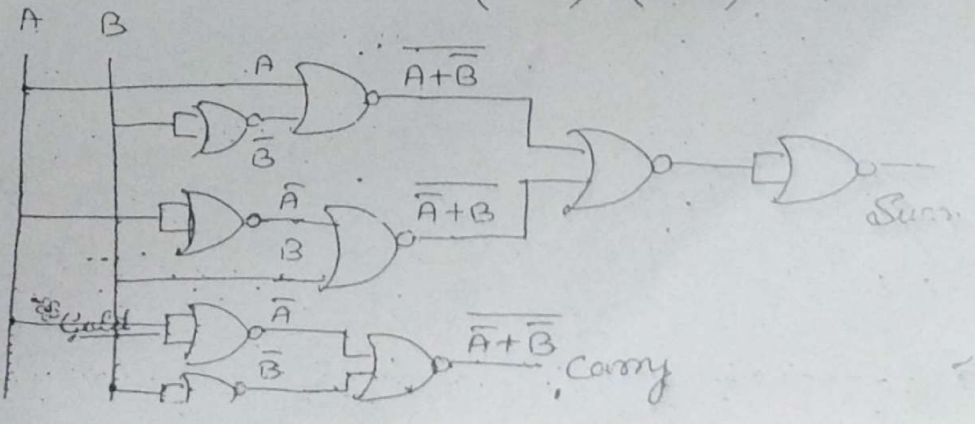
$$= \overline{(\overline{\bar{A}B}) \cdot (\overline{A\bar{B}})}$$

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

$$\text{Carry} = AB$$

$$= \overline{\overline{AB}}$$

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



Scanned by CamScanner

Full Adder

→ A full adder is a combinational ckt that forms the arithmetic sum of three i/p bits.

→ It consists of 3 i/ps & 2 o/ps.

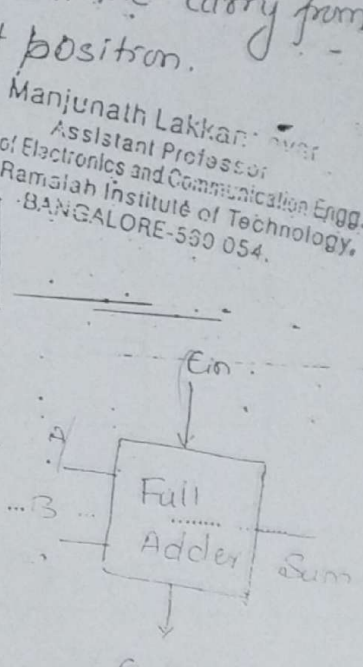
→ Two of the i/p variables, denoted by A and B, represent the two significant bits to be added.

→ The third i/p Cin, represent the carry from the previous lower significant position.

Manjunath Lakkan
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Truth Table

Inputs			Outputs	
A	B	Cin	Carry	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Block Diagram

From the truth table, we have

$$Carry = Cout = \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$$

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

$$= BC(\bar{A}+A) + AC(\bar{B}+B) + AB(\bar{C}+C)$$

$$Carry = BC + AC + AB$$

$$Sum = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

$$= \underbrace{\bar{A}\bar{B}C + ABC} + \underbrace{\bar{A}B\bar{C} + A\bar{B}\bar{C}}$$

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

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

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

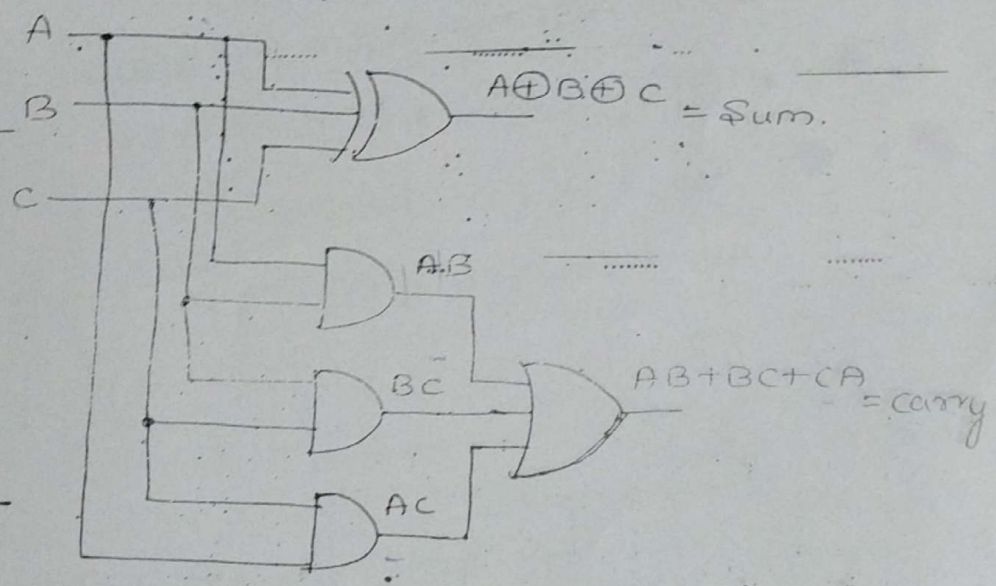
$A \oplus B = \bar{A}B + A$

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

$$\boxed{Sum = C \oplus (A \oplus B)}$$

Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology.
BANGALORE-560 054.

$A \oplus B \oplus C$



Logic Diagram

Full Adder Using Two Half Adders

→ A full adder can also be implemented with two half adders and one OR gate.

→ The sum o/p from the second half-adder is the exclusive-OR of C_{in} & the o/p of first half adder.

→ $C_{out} = AB + AC_{in} + BC_{in}$

$= AB + AC_{in}(B+\bar{B}) + BC_{in}(A+\bar{A})$

$= AB + ABC_{in} + A\bar{B}C_{in} + ABC_{in} + \bar{A}BC_{in}$

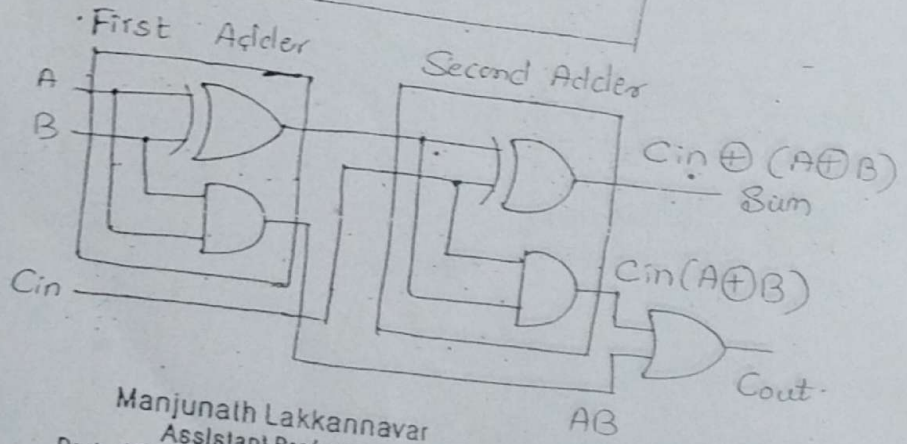
$= AB + ABC_{in} + A\bar{B}C_{in} + \bar{A}BC_{in}$

$= AB(1+C_{in}) + A\bar{B}C_{in} + \bar{A}BC_{in}$

$= AB + A\bar{B}C_{in} + \bar{A}BC_{in}$

$= AB + C_{in}(A\bar{B} + \bar{A}B)$

$$\boxed{C_{out} = AB + C_{in}(A \oplus B).}$$



Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

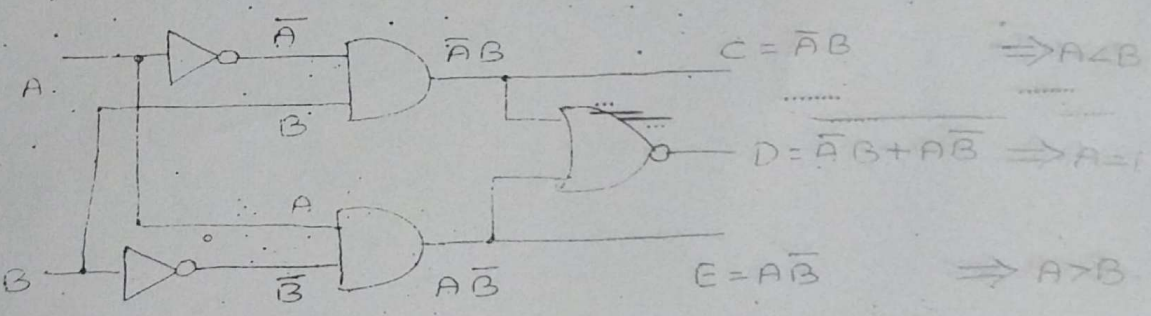
Scanned by CamScanner

1-bit Comparator

→ A digital comparator or magnitude comparator is a hardware electronic device that takes two numbers as i/p in binary form & determines whether one number is greater than, less than or equal to the other number.

→ A magnitude comparator of two 1-bits (A & B) i/ps would produce the following 3 o/p conditions when compared to each other.

$$\boxed{A > B, \quad A = B, \quad A < B}$$



Manjunath Lakkannavar
Assistant Professor
Dept. of Electronics and Communication Engg.
M.S. Ramaiah Institute of Technology,
BANGALORE-560 054.

Truth Table

Inputs		Outputs		
A	B	A > B	A = B	A < B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

Scanned by CamScanner