Scanned by CamScanner

Morris mano
Anand kumar

Digital Logic Design

Number System

Normally we use decimal system :-

$N = d_n d_{n-1} d_{n-1} \cdots\cdots d_0 d_{-1} d_{-2})_b$.

$d \Rightarrow$ digit
$b \Rightarrow$ base / radix.
$d_n \Rightarrow$ Most Significant Digit
$d_{n-2} \Rightarrow$ Least significant digit

In decimal System
$b \Rightarrow 10$.

$(325)_{10} \Rightarrow 5 \times 10^0 + 2 \times 10 + 3 \times 10^2$

~~325)~~ $(1010)_2 \Rightarrow 0 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 1 \times 2^3$
$= 0 + 2 + 0 + 8$
$= 10$.

$(127)_8 = 7 \times 8^0 + 2 \times 8^1 + 1 \times 8^2$
$= 87$.

Hexadecimal $\rightarrow$ 16 digits

$\underbrace{0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9}_{\text{normal till 9.}}$

Scanned by CamScanner

10	A
11	B
12	C
13	D
14	E
15	F

$(AF)_{16} = 15 \times 16^0 + 10 \times 16^1$
$= 175$

Q. $(12.23)_8 \longrightarrow (\quad)_{10}$

$\underleftarrow{+ve}\ \underrightarrow{-ve}$

$\Rightarrow 2 \times 8^0 + 1 \times 8^1 + 2 \times 8^{-1} + 3 \times 8^{-2}$

* **BINARY**

We use binary in digital as it comprises of 0 & 1 i.e. it follows on-off logic.

Decimal to Binary

$(17)_{10} \longrightarrow (\quad)_2$.

Method 1: Identify highest power of 2 $\leq N$.

$2^4 = 16 \leq 17$.

$4 \rightarrow 5^{th}$ bit.

1 _ _ _ _

Scanned by CamScanner

Do $17 - 2^4 = 1$
& repeat for 1
$2^0 = 1 \leq 1$

$\Rightarrow$ 1 0 0 0 1

Q. $(127)_{10} \longrightarrow (\quad)_2$

16 2^4
64 2^6

127.
$2^6 \Rightarrow 64.$ 7th.
63.
$32 \Rightarrow 2^5$ 6th.
31
$2^4 \Rightarrow 16$ 5th.
15.
$8 \Rightarrow 2^3$ 4th.
7
$2^2 \Rightarrow 4$ 3rd.
3
$2 \Rightarrow$ 2nd.
1 1st.

$\Rightarrow$ 1 1 1 1 1 1 1.

Method 2: dividend rem

2	17	
2	8	1
2	4	0
2	2	0
	~~0~~ 1	0
		1

$\boxed{10001}_2$

__/__/___

Q.) $(49)_{10} \longrightarrow (\quad)_2$

$$\begin{array}{r|l} 2 & 49 \\ \hline 2 & 24 \quad 1 \\ \hline 2 & 12 \quad 0 \\ \hline 2 & 6 \quad 0 \\ \hline 2 & 3 \quad 0 \\ \hline 2 & 1 \quad 1 \\ \hline 2 & 1 \quad 1 \\ \hline & 1 \end{array}$$

1 ~~0~~1 000 1

Q.) Conversion

i) $(10.12)_{10} \longrightarrow (\quad)_2$

for 10 its clear.

for 0.12

$0.12 \times 2 = 0.24$ 0

$0.24 \times 2 = 0.48$ 0

$0.48 \times 2 = 0.96$ 0

$0.96 \times 2 = 1.92$ 1

Do till it forms a whole number else 4-5 digits.

$\Longrightarrow (1010.0001)_2$

Q.

Q.

Q.)

MORRIS MANO

~~AN~~/ANAND KUMAR

$(32.1)_{10} \longrightarrow (\quad)_8$

$$\begin{array}{r|l} 8 & 32 \\ 8 & 4 \\ \hline & 0 \end{array}$$

0 ↑
4

$\Rightarrow$ 40

$0.1 \times 8 = 0.8 \rightarrow 0$
$0.8 \times 8 = 6.4 \rightarrow 6$
$0.4 \times 8 = 3.2 \rightarrow 3$
$0.2 \times 8 = 1.6 \rightarrow 1$
$0.6 \times 8 = 4.8 \rightarrow 4$

$\Rightarrow (40.06314)_8$

~~BIN~~

BINARY to OCTAL/HEXA

$(010\ 110\ 101 . 001\ 10)_2 \longrightarrow (\quad)_8$

$2^3 \Rightarrow 8$

so take 3 bits together

$(\underbrace{010}\ \underbrace{110}\ \underbrace{\overset{2^2 2^1 2^0}{101}} . \underbrace{\overset{2^2 2^1 2^0}{001}}\ \underbrace{100})_2$

$\longrightarrow (2\ \ 6\ \ 5\ .\ 1\ \ 4)_8$

$2^4 = 16.$

take 4 bits together

$(\underbrace{0000}\ \underbrace{1011}\ \underbrace{0101} \cdot \underbrace{0011}\ \underbrace{0000})_2$

$0 \quad B. \quad 5 \cdot 3 \quad 0.$

$\Rightarrow (B5 \cdot 30)_{16}.$

Q). $(\underbrace{011}\ \underbrace{001} \cdot \underbrace{001}\ \underbrace{101})_2$

$(3 \quad 1 \cdot 1 \quad 5)_8.$

$\underbrace{0001}\ \underbrace{1001} \cdot \underbrace{0011}\ \underbrace{0100}$

$(1 \quad 9 \cdot 3 \quad 4)_{16}.$

Octal to Hexa.

Q. $(321 \cdot 23)_8 - (\quad)_{16}$

first convert to binary.

$3 \quad 2 \quad 1 \cdot 2 \quad 3$

$(011\ 010\ 001 \cdot 010\ 011)_2$

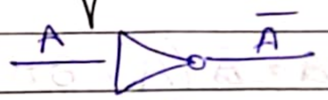
$0 \quad D \quad 1 \cdot 4C$

$(D1 \cdot 4C)_{16}.$

Scanned by CamScanner

* Logic Gates

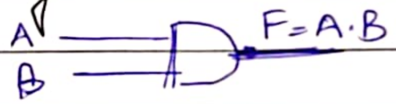
1) NOT gate



$F = \overline{A}$

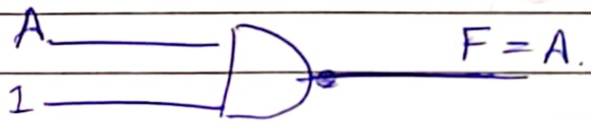
A	F
0	1
1	0

2) AND gate



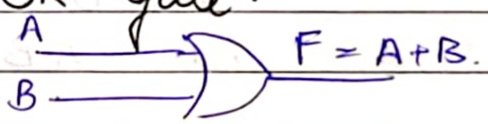
$F = A \cdot B.$

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

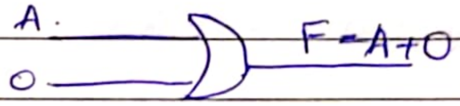


buffer

3). OR gate.

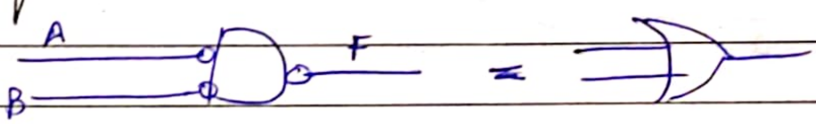


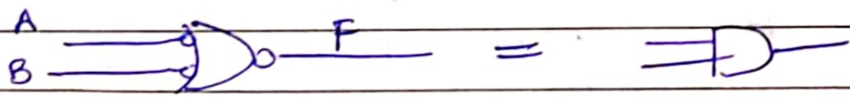
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1



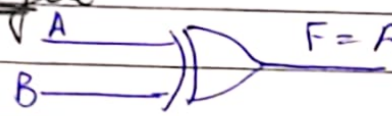
buffer.

~~4) NOR gate~~.





4) XOR gate



$F = A \oplus B$

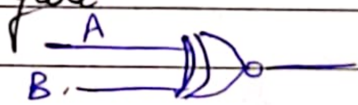
$F = A \oplus B$

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

also called inequality detector.

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

5) XNOR gate.

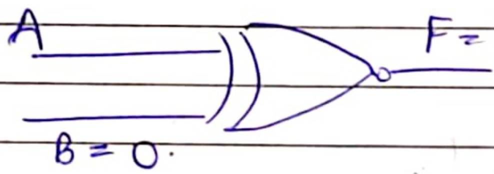


$F = A \odot B$

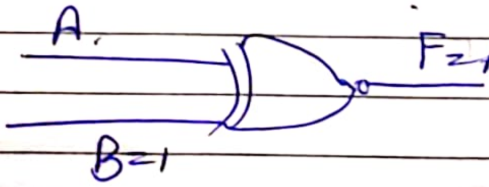
$= \overline{A \oplus B}$

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

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1



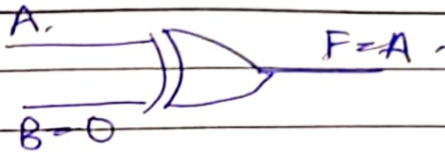
$F = \bar{A}$ - inverter



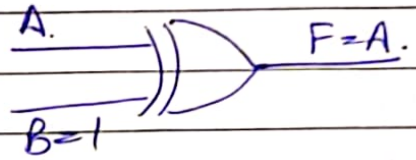
$F = A$ buffer

Scanned by CamScanner

__/__/__

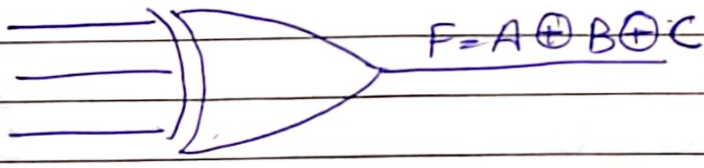


buffer.



inverter.

⇒ Triple Variable.

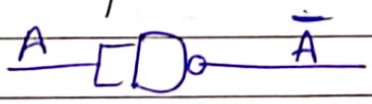


A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

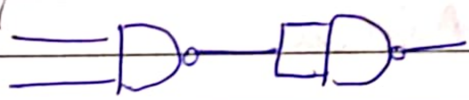
⇒ for XOR → we need odd number of '1's.

Q.) Make the following from following.

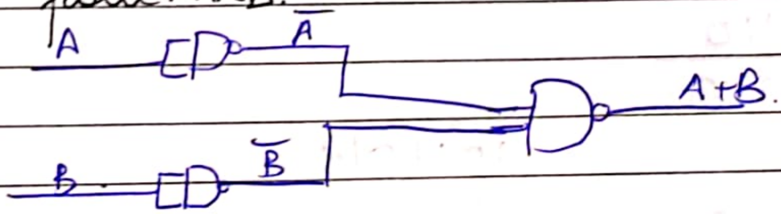
i) NOT from NAND.



ii) AND from NAND.



iii) OR from NAND.



Scanned by CamScanner

__/__/____

* Binary Algebra

Addition / Subtraction

Q.

$$\begin{array}{r} 101.11 \\ +\ 101.01 \\ \hline 1011.00 \end{array}$$

Add

0 + 0 = 0.

1 + 1 = 10 → 0 carry 1

0 + 1 = 1

1 + 0 = 1

Sub

0 − 0 = 0.

1 − 0 = 1

1 − 1 = 0.

0 − 1 = 1

1 ⇒ borrow

Q.

$$\begin{array}{r} \cancel{101.11} \\ -\ \cancel{101.01} \end{array}$$

$$\begin{array}{r} 10001 \\ -\ 01011 \\ \hline 00110 \end{array}$$

Q.

$$\begin{array}{r} 00101 \\ -\ 01010 \\ \hline \boxed{-11011} \end{array}$$

$$\begin{array}{r} 11.01 \\ -\ 0.10 \\ \hline 10.11 \end{array}$$

Multiplication & Division

Q.

$$\begin{array}{r} 111.1 \\ \times\ 110 \\ \hline 000.0 \\ 1111\ 0 \\ \hline 111100 \end{array}$$

101101.0

__/__/____

Q. 110) 101101 (111.1

$$\begin{array}{r} 101101 \\ -110 \\ \hline 1010 \\ 110 \\ \hline 1001 \\ 110 \\ \hline 0110 \\ 110 \\ \hline \times \end{array}$$

see first 3 digits. if it is smaller take 4.

* r's complemen and (r-1)'s complement.

Write the magnitude of the number but the first bit is reserved for sign.

1's complement.

$$\begin{array}{r} 11111 \\ -\ 10101 \\ \hline 01010 \end{array}$$

Either subtract from 1 or
Do 0 → 1
1 → 0] these changes.

Q) $(746)_{10}$ ⟶ 9s complement.

$$\begin{array}{r} 999 \\ -\ 746 \\ \hline \boxed{253} \end{array}$$

10 s compliment ⟶ ?
9s complement + 1

$(746) \rightarrow 253$ in 9's complement

$$\begin{array}{r} 253 \\ +1 \\ \hline \boxed{254} \end{array}$$

Q. $(101011)_2 \longrightarrow$ 1's & 2's complement

$(010100)_2 \rightarrow$ 1's comple.

$$\begin{array}{r} 010100 \\ +1 \\ \hline \boxed{010101} \end{array} \longrightarrow \text{2's complement.}$$

	Signed	1's complement	2's comp.
+5	$\boxed{0}101$	0101	0101
−5	$\boxed{1}101$	$\boxed{1}010$	$\boxed{1}011$

No change in 1's comp. & 2's compl. for +ve number.

−9	$\boxed{1}1001$	$\boxed{1}0110$	$\boxed{1}0111$
−16	$\boxed{1}10000$	$\boxed{1}01111$	$\boxed{1}10000$

★ For 2^n, we can drop sign bit. i.e. 10000 is −16 in 2's complement.

$$\begin{array}{r} 01111 \\ \hline 10000 \end{array}$$

⇒ We can write copy sign bit to LHS for any number.

eg $\boxed{1}\boxed{1}\boxed{1}\boxed{1}1001$ ✓

__/__/__

So how to recognise ?

$\boxed{1}\underbrace{1110111}$ → -ve & 2's compliment

↓ find 2's compliment.

000 1000

+1

000 $\underbrace{1001}$ → 9.

Ans = -9.

Q. 11 0001101 → ? 2's compliment.

$\boxed{1}$ 10001101 → -ve.

01110010

+1

0 $\underbrace{1110011}$ = 1+2+16+32+64.

= -115

* Overflow

When the output cannot be represented in our bit memory, it is called overflow.

2's compliment arithematics

1) 13 - 12

13 - 12 ⇒ 13 + (-12)

13 = $\boxed{0}$1101

(2's comp) -12 ⇒ $\boxed{1}$1100 → 0011

+1

$\boxed{10100}$.

__/__/____

$$
\begin{array}{r}
0\ \ 1101 \\
1\ \ 0100 \\
\hline
\cancel{1}\,0\,0001
\end{array}
$$

carry

$\boxed{00001}$

Rule 1: Discard the carry

Q2. $-13+12$

$(-13) + 12$

$12 \rightarrow 110$

$-13 \quad \boxed{1}\ 1101$

0010

+1

$\boxed{\boxed{1}\,0011}$

0000

$$
\begin{array}{rr}
 & 1\ 0011 \\
+ & 0\ 0100 \\
\hline
 & \cancel{1\,1001}
\end{array}
$$

$\boxed{11111}$

00110

+1

00111

Ans = $\boxed{1}$ 11 11

Q3. $-3-4$

$-3+(-4)$

$(-3) \rightarrow 11$

$\boxed{\boxed{1}\,01}$

$-4 \quad 100$

011

$\boxed{\boxed{1}\,100}$

00

+1

$\boxed{1}$ 01

100

011

+1

100

0110

+1

111

110

+1

$$
\begin{array}{r}
1101 \\
1100 \\
\hline
\boxed{\cancel{1}}\ 1001
\end{array}
$$

carry

$\boxed{\text{Ans} = 1001}$

~~But $\boxed{1}$ 0001 = 1~~.

__/__/__

alternate

in 2's compliment, sign bit can be copied.

so, $-3 = 101$

$-4 = 100$

$\cancel{1}\,001$ 001 = +1 which is wrong.

⇒ Take atleast 1 extra bit.

Q. 1) 127 − 54

2) 17 − 32

3) −16 − 128.

* 1's compliment Arithematics

1) 13 − 12

13 + (−12)

13 = 1101

−12 ⇒ ~~00~~ 1100

10011

~~001101~~

~~010011~~

~~$\boxed{1}$00000~~

001101

110011

$\boxed{1}$ 000000

end around.

i.e. add this carry.

000000

+1

$\boxed{0}$00001 ⇒ +1

__/__/____

Q. 12 − 13 (8 bit)

12 + (−13).

1101

[1]0010

13 ~~0000~~ 0000 1100

−12 ~~1111~~ 1 1 1 1 0 0 1 0

[1] 1 1 1 1 1 1 0

[1] 0 0 0 0 0 0 1

Q −3 − 4 (8 bit)

−3 ⇒ 1111 1100

−4 ⇒ 1111 1011

[1] 0000 0111

+1

1111 ~~0000~~ 1 0 0 0

[1]111 1 000

1000 0111 ⇒ −7

* BCD Codes. (Binary coded decimals)

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9.	1001

Scanned by CamScanner

__/__/____

But for

10 →	1	0	→	0001	0000
11 →	1	1	→	0001	0001
12 →	1	2	→	0001	~~000~~ 0010.

Excess 3 Code.

			Add 3
0	0000		0010
1	0001		0100
2	0010		0101
3	0011		⁞
4	0100		
5	0101		
6	0110		
7	0111		
8	1000	→	1011
9	1001	→	1100

eg 238.

→ 2 3 8

⇒ 0101 0110 1011

9's complement of 238 → 7 6 1

$\underbrace{1010}_{\downarrow}\ \underbrace{1001}_{\downarrow}\ \underbrace{0100}_{\downarrow \text{ 1's comp}}$

0101 0110 1011

i.e. 2 3 8.

_/_/_

* **Gray Code**

→ Used in Communication and error detection.

→ 2 consecutive numbers differ at exactly 1 bit.
i.e. we can change only 1 bit.

0 →	0	0	0	0	8 →	1	1	0	0
1 →	0	0	0	1	9 →	1	1	0	1
2 →	0	0	1	1	10 →	1	1	1	1
3 →	0	0	1	0	11 →	1	1	1	0
4 →	0	1	1	0	12 →	1	0	1	0
5 →	0	1	1	1	13 →	1	0	1	1
6 →	0	1	0	1	14 →	1	0	0	1
7 →	0	1	0	0	15 →	1	0	0	0

Binary to Gray

Q. 1 1 1 0 → ( )$_{gray}$

copy, XOR, XOR, XOR

1 0 0 0

Q. 1 0 1 1 0 0 1 0

1 1 1 0 1 0 1 1.

⇒.

__/__/____

Gray to binary

Q. 1 0 0 1 → ($\quad$)$_2$

1 $\xrightarrow{\text{copy}}$ 1, $\oplus$ 0 $\xrightarrow{\text{XOR}}$ 1, $\oplus$ 0 $\xrightarrow{\text{XOR}}$ 1, $\oplus$ 1 → 0.

⇒ 1 1 1 0.

* Sum of Product and Product of sum. and Truth

Q Write a Truth table for all number's 0 to 7 divisible by 3.

0 → 7 ⇒ atleast 3 bit -

	A	B	C	F
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

Now we can express F in 2 ways:-

Minterms / SOP (output = 1)

Maxterms / POS. (output = 0)

$\Sigma(m_0, m_3, m_6)$

$\Pi(M_1, M_2, M_4, M_5, M_7)$

Scanned by CamScanner

__/__/____

$F = \Sigma(0,3,6)$

$F = m_0 + m_3 + m_6$

$F = \bar{A}\cdot\bar{B}\cdot\bar{C} + \bar{A}\cdot BC + A\cdot B\cdot\bar{C}$

(compliment for '0')

$F = \Pi(M_1, M_2, M_4, M_5, M_7)$

$F = \Pi(1,2,4,5,7)$

$F = M_1 \cdot M_2 \cdot M_4 \cdot M_5 \cdot M_7$

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

(compliment for '1')

* K-MAP

* 3 variable

A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	000 [0]	001 [1]	011 [3]	010 [2]
A	100 [4]	101 [5]	111 [7]	110 [6]

* 4 variable

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

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0 0000	1 0001	3 0011	2 0010
$\bar{A}B$	4 0100	5 0101	7 0111	6 0110
AB	12 1100	13 1101	15 1111	14 1110
$A\bar{B}$	8 1000	9 1001	11 1011	10 1010

Scanned by CamScanner

* 5 variable

Draw 2 kMap, one with X = 0 and 1 with X = 1

X = 0

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

X = 1

AB \ CD			
16	17	19	18
20	21	23	22
28	29	31	30
24	25	27	26

X = 1

AB \ CD			
0	1	3	2
4	5	7	6
8			

NOTE.

for connection b/w 2 different ...
if there is just a 1bit they can be ...
0 → 16.

Q Draw a kMap for 3bit ~~number~~ truth table T.T.

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

A \ BC			
1	1		
	1		
1			

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

Q. $\Sigma = (0,1,2,3,4,6,7) = ?$

__/__/____

	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0	1	1	1
A	1	0	1	1

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

Product of sum

AB \ CD	$C+D$	$C+\bar{D}$	$\bar{C}+\bar{D}$	$\bar{C}+D$
$A+B$	0	1	3	2
$A+\bar{B}$	4	5	7	6
$\bar{A}+\bar{B}$	12	13	15	14
$\bar{A}+B$	8	9	11	10

Q $F = \prod(0, 2, 3, 4, 9, 13)$. Minimise the function using K Maps.

	$C+D$	$C+\bar{D}$	$\bar{C}+\bar{D}$	$\bar{C}+D$
$A+B$	0 (0)	(1)	0 (3)	0 (2)
$A+\bar{B}$	0 (4)	(5)	(7)	(6)
$\bar{A}+\bar{B}$	(12)	0 (13)	(15)	(14)
$\bar{A}+B$	(8)	0 (9)	(11)	(10)

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

Don't care

Scanned by CamScanner

Q. Number b/w 0 – 9 divisible by 3 give output 1. Find function

A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

nothing specified hence don't care.

SOP

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1		1	
$\bar{A}B$				1
AB	X	X	X	X
$A\bar{B}$		1	X	X

$\Rightarrow \bar{A}\bar{B}\bar{C}\bar{D} + \bar{B}CD + BC\bar{D} + AD.$

POS.

	$C+D$	$C+\bar{D}$	$\bar{C}+\bar{D}$	$\bar{C}+D$
$A+B$	0	0	0	0
$A+\bar{B}$	0	0	0	
$\bar{A}+\bar{B}$	X	X	X	X
$\bar{A}+B$	0		X	X

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

Scanned by CamScanner

_ / _ / _

Q. $F = \Sigma_m (0,1,2,3,4,5) + d(10,11,12,13,14,15)$
Minimize in POS.

	$C+D$	$C+\bar{D}$	$\bar{C}+\bar{D}$	$\bar{C}+D$
$A+B$	[0]	[1]	[3]	[2]
$A+\bar{B}$	[4]	[5]	0 [7]	0 [6]
$\bar{A}+\bar{B}$	X [12]	X [13]	X [15]	X [14]
$\bar{A}+B$	0 [8]	0 [9]	X [11]	X [10]

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

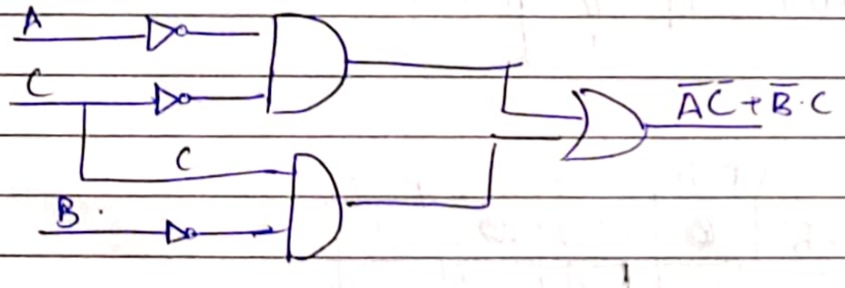
Convert SOP to POS.

$\pi_M(6,7,8,9) + d(10,11,12,13,14,15)$.

ii) SOP.

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	1	1
$\bar{A}B$	1	1		
AB	X	X	X	X
$A\bar{B}$			X	X

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



* Boolean Algebra

1) AND LAW

$A \cdot A = A$

$A \cdot 0 = 0$.

2) OR LAWS

__/__/____

$A + A = A$

$\bar{A} + A = 1$

3. Associative Law.

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

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

4) Commutative Law.

$A\cdot B = B\cdot A$

$A+B = B+A$.

5) Distributive

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

$RHS \Rightarrow A\cdot A + A\cdot C + B\cdot A + B\cdot C$

$\Rightarrow A\cdot(1+C) + B\cdot A + B\cdot C$

$= A\cdot(1+B) + BC$

$= \underline{A + BC}$.

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

6) Absorption Law.

$A + AB = A$.

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

$A + A\cdot B$.

Scanned by CamScanner

__/__/__

7). Consensus

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

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

$$\Rightarrow AB + \bar{A}C + BC\bar{A} + BCA$$

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

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

8). Transportation

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

$$= \underbrace{A\bar{A}}_{=0} + AB + C\bar{A} + CB$$

$\Rightarrow$ using consensus theorem

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

9) De Morgan's Law

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

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

Q. $\overline{\overline{AB} + \bar{A} + AB}$

$$AB + \overline{AB} = 1$$

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

$$\bar{1} = 0.$$

Hence $\underline{\underline{0}}$.

__/__/____

Q2) $A[B+\overline{C}(\overline{AB+A\overline{C}})]$

$A[B+\overline{C}\cdot(\overline{AB}\cdot\overline{A\overline{C}})]$

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

$A[B+\overline{C}(\overline{A}+\overline{B}C+\overline{A}\overline{C})]$

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

$A\cdot\overline{A} = 0$

$C\cdot\overline{C} = 0$

$\Rightarrow \underline{AB} + 0$

Q3) $(A+B)(\overline{\overline{A}(\overline{B}+\overline{C})}) + \overline{A}(B+C)$

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

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

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

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

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

$= A + C + B + BC$

$= A+B+C$

Imp

QUINNE Mccluskey (Tabular Method)

$f = \Sigma_m (0,1,6,7,8,9,13,14,15)$.

Step 1: Write binary

place of difference (2^i).

Index 0	0000 (0)	(0,1) ⇒ (1) ⇒ 000_
Index 1	0001 (1)	(0,8) = (8) ⇒ _000
	1000 (8).	(1,9) = (8) ⇒ _001
Index 2.	0110 (6)	(8,9) ⇒ (1) ⇒ 100_
	1001 (9)	(6,7) ⇒ (1) ⇒ 011_
Index 3	0111 (7)	(6,14) ⇒ (8) = _110
	1101 (13)	(9,14) ⇒ (3) _110 *
	1110 (14).	(9,13) ⇒ (4) 1_01 * R
Index 4.	1111 (15).	(7,15) ⇒ (8) _111
		(13,15) ⇒ (2) 11_1 * S
		~~(14,15) ⇒ (1) 111_~~

Step 2: Pair numbers differing in 1 bit.

Step 3: Pair for missing bit in column 3.
+ 1 bit change

C3-

0,1,8,9 (1,8) _00_ *] no pair Q.

neglect X (08,1,9) (1,8) _00_

7,6,14,15 (1,8) _11_ *.] P

neglect X (6,14,7,15) (1,8) _11_

Implicants: The pair that give answer

Prime Implicants: Optimised implicants

Scanned by CamScanner

implicants

prime implicants

NOTE: If only one pair is possible for a prime implicants become essential prime
Reverse of this is non-essential "

Step 4: Decide prime implicants (Replace - with ×)

P → × 11 × BC
Q → × 00 × $\bar{B}\bar{C}$
R → 1 × 0 1 $A\bar{C}$
S → 11 × 1 ABD.

Step 5: Write ~~[illegible]~~ 4 variable.

~~F(A,B,C,D) → BC + $\bar{B}\bar{C}$ + $A\bar{C}$ + ABD.~~

	0 ✓	1 ✓	6 ✓	7 ✓	8 ✓	9 ✓	13	14 ✓	15
Q(0,1,8,9)*	×	×			×	×			
P(6,7,14,15)*			×	×				×	×
R(9,13)						×	×		
S(13,15)							×		×

Step 6: Look each column. If a column has 1 ×
consider the variable essentially prime
Columns with multiple × are non essential
tick 9 as it is mentioned in Q
15 " " " " P.

No exception in 13 ⇒ non essential implicant

Ans: $P + Q + R$ or $P + Q + S$

$\underbrace{R \quad \text{or} \quad S}_{\text{cos of } 13}$

Q. $F = \pi M(2,3,8,12,13) + d(10,14)$.

Index 1	0+0+1+0 (2)	(2,3) ⇒ (1) → 001_ (S)	(2,10,12,14) ⇒ 1_ _0 (P)
	1+0+0+0 (8)	(2,10) ⇒ (8) – 010 (R)	(8,12,10,14) ⇒ 1_ _0
Index 2	0011 (3)	(8,12) ⇒ (4) = 1_00	
	1100 (12)	(8,10) ⇒ (2) ⇒ 10_0	
	1010 (10)	(12,13) ⇒ (1) ⇒ 110_ (Q)	
Index 3	1101 (13)	(12,14) ⇒ (2) ⇒ 11_0	
	1110 (14)	(10,14) ⇒ (4) ⇒ 1_10	

$P \Rightarrow (\bar{A} + D)$ (8,10,12,14)

$Q \Rightarrow (\bar{A} + \bar{B} + C)$ (12,13)

$R \Rightarrow (B + \bar{C} + D)$ (2,10)

$S \Rightarrow (A + B + \bar{C})$ (2,3)

	2	3	8	12	13
P (8,10,12,14)*			X	X	
Q (12,13)*				X	X
R (2,10)	X				
S (2,3)*	X	X			

10, 14 → d(10,14) not includ-ed

P, S → essential prime implicant.

$$\text{Ans} = P \cdot Q \cdot S$$
$$= (\bar{A}+D)\cdot(\bar{A}+\bar{B}+C)(A+B+\bar{C})$$

* Adder and Subtractors

carry $\boxed{1}$ $\boxed{1}$ carry$_{in}$

a = 1 0 1

b = 0 1 1

$\boxed{0}$ $\boxed{0}$ sum.

→ half adder

→ full adder

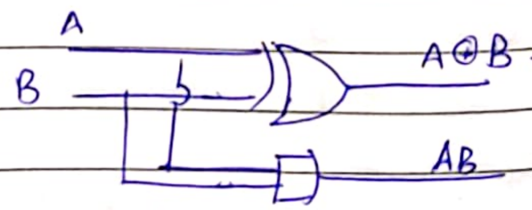
- We need a full adder, as half adder takes only 2 inpu[ts] and so we have to ignore the previous carry.

Half Adder

a	b	Sum	Cout
0	0	0	0
0	1	~~0~~ 1	0
1	0	1	0
1	1	0	1

$Sum = A \oplus B$

$C_{out} = AB.$

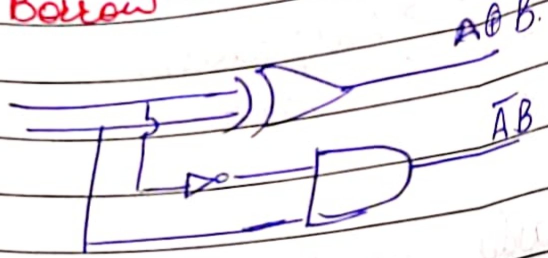


Half Subtractor.

A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$Diff = A \oplus B$.

$Borrow = \bar{A}B$



FULL ADDER

A	B	C_{in}	Sum	C_{out}
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

Sum =

A \ BC_{in}	$\bar{B}\bar{C}_{in}$	$\bar{B}C_{in}$	BC_{in}	$B\bar{C}_{in}$
$\bar{A}$		1		1
A	1		1	

$\Rightarrow \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\cdot\bar{B}\bar{C}_{in} + ABC_{in}$.

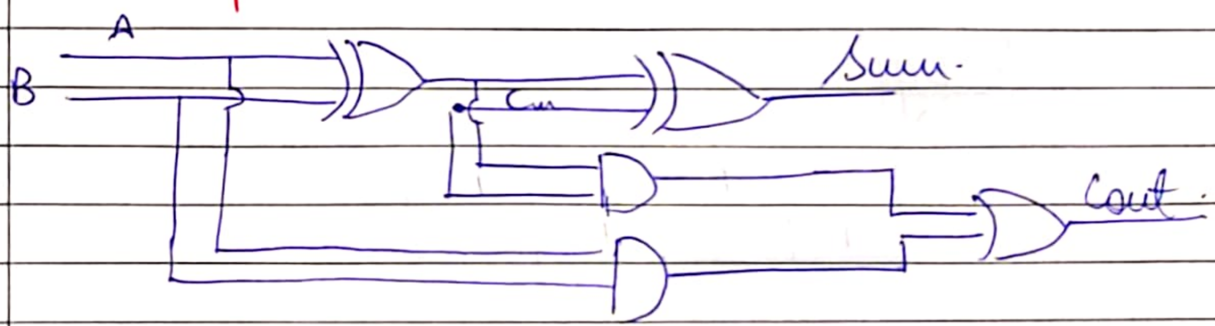
$= \bar{A}(B \oplus C_{in}) + A(\bar{B}\bar{C}_{in} + BC_{in})$

$\bar{A}(B \oplus C_{in}) + A(\overline{B \oplus C_{in}})$ (XNOR)

$= A \oplus B \oplus C$

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

Carry $= AB + BC + AC$.



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

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

$\Rightarrow B(A+\bar{A})(A+C_{in}) + A\bar{B}C_{in}$.

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

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

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

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

$= AB + BC_{in} + AC_{in}$.

2 Half Adder + 1 OR gate = Full adder.

Scanned by CamScanner

Full Subtractor

A	B	B_{in}	Diff	B_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

for Diff

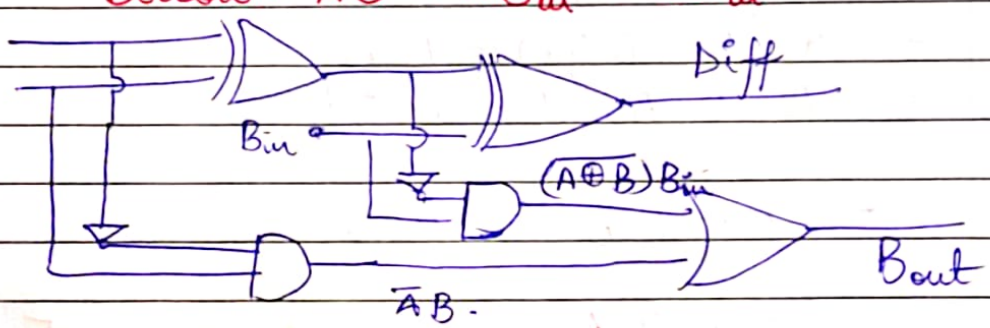
	1		1
1		1	

$$\Rightarrow \bar{A}\bar{B}B_{in} + \bar{A}B\bar{B}_{in} + A\cdot\bar{B}\bar{B}_{in} + ABB_{in}$$

$$Diff = A \oplus B \oplus B_{in}$$

for Borrow

$$Borrow = \bar{A}B + \bar{A}B_{in} + BB_{in}$$



$$B_{out} = \overline{(A \oplus B)}\, B_{in} + \bar{A} B .$$

$$= A B B_{in} + \bar{A}\bar{B} B_{in} + \bar{A} B .$$

$$= A B B_{in} + \bar{A}\left(B + \bar{B} B_{in} \right)$$

$$= A B B_{in} + \bar{A}\, \overbrace{\cancel{(B + \bar{B})}}^{=1} (B + B_{in})$$

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

$$= B (\bar{A} + A)(\bar{A} + B_{in}) + \bar{A} B_{in} .$$

$$= \bar{A} B + B_{in} B + \bar{A} B_{in} .$$

$$B_{out} = \bar{A} B + B_{in} B + \bar{A} B_{in} .$$

Scanned by CamScanner