Digital Logic Design

- Signal
  - Cont
    - Analog → Infinite amplitude levels
    - Digital → Finite amplitude levels
  - Discont
    - Analog
    - Digital

Ch-1

Number System

Base-'r' system – A number expressed in base 'r' system has co-efficients multiplied by powers of r. This system is capable of storing 'r' number of values ranging from '0' to 'r-1' where r is called the Radix / Base of a system.

Eg: $(a_3\ a_2\ a_1\ a_0)_r$

The power of the base will always increase from RHS to LHS and will always decrease from LHS to RHS.

$(a_3^2\ a_2^2\ a_1^1\ a_0^0)_r$

$= a_3 \times r^3 + a_2 \times r^2 + a_1 \times r^1 + a_0 \times r^0$

$(7936)_{10} = 7 \times 10^3 + 9 \times 10^2 + 3 \times 10^1 + 6 \times 10^0$

① Binary Number System

Here base value = 2 i.e; $r = 2$.

Hence, it will have only 2 possible values that is 0 & 1.

Eg → $(1011)_2 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$

$= 8 + 0 + 2 + 1 = (11)_{10}$

$(0.110)_2 = 1\times2^{-1} + 1\times2^{-2} \times 0\times2^{-3}$

$= 0.5 + 0.25 + 0 = (0.75)_{10}$

$(1011.110)_2 = (11.75)_{10}$

② Decimal Number System

Here base value = 10 i.e; $r = 10$.

Hence, it will have 10 no. of values that is 0 to 9.

$(1\,2\,5\,8)_{10} = 1\times10^3 + 2\times10^2 + 5\times10^1 + 8\times10^0$

③ Octal Number System

Here base value = 8 i.e; $r = 8$.

Hence, it will accomodate 8 no. of values ranging from 0 to 7.

$(5\,3\,6)_8 = 5\times8^2 + 3\times8^1 + 6\times8^0 = (350)_{10}$

8 | 8

1 → 0

base

remainder

(4) Hexadecimal Number System

Here base value = 16 i.e; $r = 16$

Hence, it will accomodate 16 values ranging from 0 to 9 then A, B, C, D, E and F.

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

$$(AC53)_{16} = A \times 16^3 + C \times 16^2 + 5 \times 16^1 + 3 \times 16^0$$

Number with different Bases :-

Decimal ($r=10$)	Binary ($r=2$)	Octal ($r=8$)	Hexadecimal ($r=16$)
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Arithmetic Operations :-

① Addition :-

a → Augend
+ b → Addend
S → Sum

Rule

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 with carry 1.

Q. a = 101101
+ b = 011010

```
  101101
+ 011010
 110111
```

② Subtraction :-

a → Minuend
− b → Subtrahend
d → Difference

Rule

0 − 0 = 0
0 − 1 = 1 with borrow 1
1 − 0 = 1
1 − 1 = 0

Q. a = 110000
− b = 01101

```
  110000
− 01101
 100011
```

③ Multiplication :-

a → Multiplicand
× b → Multiplier
P → Product

```
   1011
×   101
  1011
 0000
1011
110111
```

Number Base Conversion :-

① Binary to Decimal

$$(1011.11)_2 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 + 1\times2^{-1} + 1\times2^{-2}$$

$$= (11.75)_{10}$$

② Decimal to Binary

$(23)_{10} = (10111)_2$

$$\begin{array}{r|l} 2 & 23 \\ 2 & 11 \rightarrow 1 \\ 2 & 5 \rightarrow 1 \\ 2 & 2 \rightarrow 1 \\ & 1 \rightarrow 0 \end{array}$$

$(0.6875)_{10} \Rightarrow$

$0.6875 \times 2 = 1.375 \rightarrow 1$

$0.375 \times 2 = 0.75 \rightarrow 0$

$0.75 \times 2 = 1.5 \rightarrow 1$

$0.5 \times 2 = 1.0 \rightarrow 1$

$= (1011)_2$

$* \; 8 \quad 4 \quad 2 \quad 1 \; : \; 0.5 \quad 0.25 \quad 0.125 \quad 0.0625$

③ Octal to Decimal

$$(624.52)_8 = 6\times8^2 + 2\times8^1 + 4\times8^0 + 5\times8^{-1} + 2\times8^{-2}$$

$$=$$

④ Decimal to Octal

$(127)_{10} = (177)_8 \Rightarrow$

$$\begin{array}{r|l} 8 & 127 \\ 8 & 15 \rightarrow 7 \\ & 1 \rightarrow 7 \end{array}$$

Q. $(0.513)_{10} =$

⑤ **Binary to Octal** : Convert the given binary number into its equivalent octal number.

$(1011110 1101 . 110)_2$

$\rightarrow ()_2 \rightarrow ()_{10} \rightarrow ()_8$

→ Since $2^3 = 8$ so, we will make a group of 3 bits to obtain its equivalent octal value.

Ex: $(\underbrace{00}_{1}\,\underbrace{1011}_{3}\,\underbrace{110}_{6}\,\underbrace{1}_{1}\,.\,\underbrace{101}_{5}\,\underbrace{110}_{6}\,\underbrace{100}_{4})_2$

4 bit → 8421

3 bit → 421

$= (1355.64)_8$

⑥ **Octal to Binary** : We can express the octal number as a group of 3-bit binary values.

$(537.21)_8$

$= (101\,011\,111 . 010\,001)_2$

⑦ **Binary to Hexadecimal** : Since $2^4 = 16$ we will take a group of 4 binary bits to obtain its equivalent hexadecimal value.

$H = 2$ to $H = 16$

$2^4 = 16 \rightarrow$ 4 bits

$(0001\,1101\,1111\,1010\,0111 . 1100\,0100)_2$

$(1\ D\ F\ A\ 7 . C\ 4)_{16}$

⑧ Hexadecimal to Binary :

$(2\ A\ 5\ 1\ C\ .\ F\ F)_{16}$

$= (0010\ \ 1010\ \ 0101\ \ 0001\ \ 1100.\ 1111\ \ 1111)_2$

Find the base / radix value 'r' :-

① $(0.7)_{10} = (0.1011\ldots)_2$

$0.7 \times 2 \rightarrow 1.4 \rightarrow 1$
$0.4 \times 2 \rightarrow 0.8 \rightarrow 0$
$0.8 \times 2 \rightarrow 1.6 \rightarrow 1$
$0.6 \times 2 \rightarrow 1.2 \rightarrow 1$

② $(231.3)_4 = (\ )_7 \Rightarrow (\ )_4 = (\ )_{10} = (\ )_7$

$(\overset{2}{2}\overset{1}{3}\overset{0}{1}.\overset{-1}{3})_4 = 2 \times 4^2 + 3 \times 4^1 + 1 \times 4^0 + 3 \times 4^{-1}$

$= 32 + 12 + 1 + 0.75$

$= (45.75)_{10}$

$0.75 \times 7 = 5.25 \rightarrow 5$
$0.25 \times 7 = 1.75 \rightarrow 1$
$0.75 \times 7 = 5.25 \rightarrow 5$
$0.25 \times 7 = 1.75 \rightarrow 1$

$2 \mid 45$
$2 \mid 22 \rightarrow 1$
$2 \mid 11 \rightarrow 1$
$2 \mid 5 \rightarrow 1$
$2 \mid 2 \rightarrow 0$
1

$(0.75)_{10} = (0.5151\ldots)_7$

$(45.75)_{10} = (63.5151\ldots)_7$

③ $(16)_{10} = (100)_b \Rightarrow b = ?$

$16 = 1 \times b^2 + 0 \times b^1 + 0 \times b^0$

$\Rightarrow 16 = b^2$

$\Rightarrow b = \sqrt{16} = \pm 4$

The base value of a number can't be negative.

So, $\boxed{b = 4}$

④ $\frac{41}{3} = 13$

$\frac{(41)_b}{(3)_b} = (13)_b$

$\Rightarrow \frac{4 \times b^1 + 1 \times b^0}{3 \times b^0} = 1 \times b^1 + 3 \times b^0$

$\Rightarrow \frac{4b+1}{3} = b+3$

$\Rightarrow 4b + 1 = 3b + 9$

$\Rightarrow \boxed{b = 8}$

⑤ $\frac{33}{3} = 11 \Rightarrow$ find b ?

$\frac{(33)_b}{(3)_b} = (11)_b$

$\Rightarrow \frac{3 \times b^1 + 3 \times b^0}{3 \times b^0} = 1 \times b^1 + 1 \times b^0$

$\Rightarrow \frac{3b+3}{3} = b+1$

$\Rightarrow 3b+3 = 3b+3$ (Applicable only if LHS = RHS)

Since LHS = RHS. Hence, this holds good for all values of b.

if $b = 2$, values $= 0, 1$

if $b = 3$, values $= 0, 1, 2$

if $b = 4$, values $= 0, 1, 2, 3$.

So, $\boxed{b > 3 \text{ or } b \geq 4}$

Shortcut
find the largest number ~~of it~~ then the base will be greater than that.

(6) $1234 + 5432 = 6666$

$\boxed{b > 6}$

(7) $23 + 44 + 14 + 32 = 223 \Rightarrow x = ?$

$(23)_b + (44)_b + (14)_b + (32)_b = (223)_b$

$\Rightarrow 2\times b^1 + 3\times b^0 + 4\times b^1 + 4\times b^0 + 1\times b^1 + 4\times b^0 + 3\times b^1 + 2\times b^0 = 2\times b^2 + 2\times b^1 + 3\times b^0$

$\Rightarrow 2b + 3 + 4b + 4 + b + 4 + 3b + 2 = 2b^2 + 2b + 3$

$\Rightarrow 8b + 10 = 2b^2$

$\Rightarrow b^2 = 4b + 5$

$\Rightarrow b^2 - 4b - 5 = 0$

$\Rightarrow b - 5b + b - 5 = 0$

$\Rightarrow b(b-5) + 1(b-5) = 0$

$\Rightarrow (b+1)(b-5) = 0 \Rightarrow b = \cancel{-1}$ & $\boxed{b = 5}$

(8) $\frac{302}{20} = 12.1$

$\Rightarrow \frac{3b^2+2}{2b} = b+2+\frac{1}{b}$

$\Rightarrow \frac{3b^2+2}{2b} = \frac{b^2+2b+1}{b}$

$\Rightarrow 3b^3+2b = 2b^3+4b^2+2b$

$\Rightarrow b^3-4b^2=0$

$\Rightarrow b^2(b-4)=0$

$\Rightarrow \boxed{b=4}$

(9) $\sqrt{41} = 5$

$\sqrt{(41)_b} = (5)_b$

$\Rightarrow \sqrt{4b+1} = 5$

Expressing in decimal (mandatory to write this line)

$4b+1=25$

$4b=24$

$\Rightarrow \boxed{b=6}$

(10) $(4433)_5 = (?)_{10} = (?)$

(11) $(5654)_7 = (?)_{10} = (?)$

(12) $(1221)_3 = (?)_{10} = (?)$

If a ques like $(46)_5 = (?)_{10}$ is given then its invalid as the base can't be smaller than the digit.

*If we attend this ques we can get grace marks.

Imp

1.4: Representation of Negative Numbers

There are two ways of representing signed no.s :-

① Sign - Magnitude form

② Complement form < 1's complement / 2's complement.

① Sign - Magnitude form :-

Here an additional bit called the sign bit is placed infront of the number if the sign bit = 0 then it is a positive number. If sign bit = 1 then negative number.

Eg : +7 = 0 111 → sign-bit; 111 → magnitude bit (binary form of 7)

-3 = 1 011

+5 = 0 101

-12 = 1 1100

★ If 'n'-bits are used to represent numbers then the sign-magnitude and 1's complement represent numbers in the range $-(2^{(n-1)}-1)$ to $(+2^{(n-1)}-1)$

Here a large amount of manipulation is needed hence it is impratical and we will use the complement form.

Complement form :-

In 2's or 1's complement

(i) If the number is positive, then the magnitude is represented in its true binary form and 0 is placed infront of the MSB.

1 0 11 0.1 → Least significant Bit (LSB)

Most significant Bit (MSB) ←

(ii) If the number is negative the magnitude is represented in its 2's / 1's complement form and a sign-bit 1 is placed infront of the MSB.

1's Complement :-

There are two ways to find the 1's complement of a given binary number.

(a) Change the zeroes to ones and vice versa.

(b) The 1's complement of n-bit binary number N is given as $\boxed{\bar{N} = (2^n - 1) - N}$

Find the 1's complement of 0101

Ans:- (a) 1010

(b) $n = 4$

$$\bar{N} = (2^4 - 1) - N$$
$$= 15 - 0101$$
$$= 1111 - 0101$$
$$= (1010)$$

2's Complement :-

For the 'n'-bit number, the 2's complement will have the range between $-(2^{n-1})$ to $+(2^{(n-1)} - 1)$

$n = 4$, -2^3 to $+2^3 - 1 = -8$ to $+7$

There are three ways to find the 2's complement of a number N :-

(a) Obtain the 1's complement of a number and then add 1.

(b) Copy down the bits starting from LSB upto and including the first one and then complement the remaining bits.

(c) By subtracting the given 'n'-bit number N from 2^n that is $\boxed{N^* = 2^n - N}$

Eg: N = 10101

n = 5

(a) 01010 → 1's comp of N.

$$\begin{array}{r} 01010 \\ +1 \\ \hline 01011 \end{array}$$

(b) 01011

(c) $N^* = 2^n - N$

$= 2^5 - 10101$

$= 100000 - 10101$

$= 01011$

To represent the number in sign 2's / 1's complement form, determine the 2's / 1's complement of the magnitude of the number and then add the sign-bit.

The 2's / 1's complement operation on a signed number will change a positive number to a negative number and vice versa.

$(7)_{10} = (111)_2$ (varies in negative number)

$-7 = 1\,111 \rightarrow$ sign – magnitude

$-7 = 1\,000 \rightarrow$ sign – 1's comp

$-7 = 1\,001 \rightarrow$ sign – 2's comp

Q. Determine the decimal value in each case if they are expressed in

① sign – magnitude form

② 2's complement form

③ 1's complement form.

Ans:–

Given no.	Sign – mag	2's comp	1's comp
01101	+ 13	+13	+13
10111	– 7	–9	–8

Diff should be always 1

010111	+23	+23	+23
1 101010 (32 16 8 4 2 1)	–42	–22	–21

★ Special Case in 2's complement representation:–

Whenever a sign number has 1 in the sign-bit and all zeroes for the magnitude bits then the decimal equivalent is given as -2^n where n is the number of bits in the magnitude part.

Eg: $1000 \Rightarrow -2^n$

$\underset{3}{\underset{\downarrow}{}}$ (no. of bits) $= -2^3$

$= -8$

$10000 \Rightarrow -2^n$

$= -2^4$

$= -16$

$8 = 1000$

1's $8 = 0111$

$+ \quad 1$

1000

Characteristics of 2's complement numbers :-

The ~~the~~ 2's complement numbers have the following properties :-

(i) There is one unique zero.

(ii) The 2's complement of zero is zero.

(iii) The left most bit cannot be used to express a quantity because it is a sign-bit if it is 1, then it is a -ve number & if it is 0 then it is a +ve number.

(iv) For an n-bit word, which includes the sign-bit there are $(2^{(n-1)}-1)$ positive integers 2^{n-1} negative integers and one unique zero that is total of 2^n integers state.

(v) A negative number

Signed Binary Numbers :-

Decimal	Sign - Magnitude	Sign 1's comp	Sign 2's comp
+7	0 111	0 111	0 111
+6	0 110	0 110	0 110
+5	0 101	0 101	0 101
+4	0 100	0 100	0 100
+3	0 011	0 011	0 011
+2	0 010	0 010	0 010
+1	0 001	0 001	0 001
+0	0 000	0 000	0 000
-0	1 000	1 111	—
-1	1 001	1 110	1 111
-2	1 010	1 101	1 110
-3	1 011	1 100	1 101
-4	1 100	1 011	1 100
-5	1 101	1 010	1 011
-6	1 110	1 001	1 010
-7	1 111	1 000	1 001
-8	—	—	1 000

Addition of 2's complement numbers :-

In the addition of 'n'-bit sign-binary numbers, any carry from the sign-position is ignored. This will always produce the correct result except when the overflow occurs.

The overflow will occur for a word length of 'n'-bits if the correct representation of the sum (including sign-bit) requires more than 'n'-bits.

Let, $n = 4$

Case-1 : Addition of two positive numbers where the sum is less than 2^{n-1}.

$$\begin{array}{lll} +3 \rightarrow & 0 & 0\ 1\ 1 \\ +4 \rightarrow & 0 & 1\ 0\ 0 \\ \hline +7 & 0 & 1\ 1\ 1 \end{array}$$

$\rightarrow$ correct answer.

Case-2 : Addition of two positive numbers ; sum $> 2^{n-1}$

$$\begin{array}{lll} +5 \rightarrow & 0 & 1\ 0\ 1 \\ +6 \rightarrow & 0 & 1\ 1\ 0 \\ \hline +11 & 1 & 0\ 1\ 1 \end{array}$$

$\rightarrow$ Wrong answer because of overflow. (+11 requires 5 bits including the sign-bit)

Case-3 : Addition of a positive and negative number where -ve no. > +ve no.

$$\begin{array}{lll} +5 \rightarrow & 0 & 1\ 0\ 1 \rightarrow +5 \\ -6 \rightarrow & 1 & 0\ 1\ 0 \rightarrow (2\text{'s comp of } 6) \\ \hline -1 & 1 & 1\ 1\ 1 \end{array}$$

$\rightarrow$ Here MSB = 1, hence it's a -ve no. the result is 2's comp of 111 = 001 = (-1)

Case-4 : Addition of +ve & -ve numbers, +ve no. > -ve no.

$$\begin{array}{lll} -5 & \longrightarrow & 1011 \\ +6 & \longrightarrow & 0110 \\ \hline +1 & & \textcircled{1}0001 \end{array}$$

$\longrightarrow$ ignore the carry from the sign-bit. it is a sign-bit. It is not an overflow hence it is the correct ans.

$\downarrow$ any-carry should be ignored

Case-5 : Addition of 2 -ve nos, $|sum| \leq 2^{n-1}$

$$\begin{array}{lll} -3 & \longrightarrow & 1101 \\ -4 & \longrightarrow & 1100 \\ \hline -7 & & \textcircled{1}1001 \end{array}$$

$\longrightarrow$ ignore the carry. MSB = 1 so, its a negative number. The result is 2's complement of 001 i.e; −7. So, its a correct answer.

$\hookrightarrow$ carry

Rules for detecting overflow :—

When adding 2 'n'-bit sign-binary numbers (using 1's or 2's complement) to get the 'n'-bit sum then

(i) An overflow occurs if and only if the carry out of the sign-position is not equal to the carry into the sign-position.

(ii) An overflow occurs if adding two +ve numbers gives a negative answer or if adding two -ve numbers gives a positive answer.

Case-6 : Addition of 2-ve no.s, $|sum| > 2^{n-1}$

$-5 \rightarrow 1\ 0\ 1\ 1$

$-6 \rightarrow 1\ 0\ 1\ 0$

$-11 \quad 1\ 0\ 1\ 0\ 1 \rightarrow$ it is wrong answer because of overflow. -11 requires 5-bits including the sign-bit.

1's Complement Numbers :-

The 1's complement of a negative number is given as

① $\bar{N} = (2^n - 1) - N$

② Change 0's to 1's & vice-versa

Addition of 1's Complement Numbers :-

Here instead of discarding the last carry, it is added to the 'n'-bit sum to the right most position, this is known as End-around carry.

Let, $n = 4$

Case-1 : Addition of 2 positive numbers, sum $< 2^{n-1}$

$+3 \rightarrow 0\ 0\ 1\ 1$

$+4 \rightarrow 0\ 1\ 0\ 0$

$+7 \quad 0\ 1\ 1\ 1 \Rightarrow +7 \rightarrow$ Correct Answer.

$\downarrow$ +ve $\quad \downarrow$ 7

Case-2 : Addition of 2 positive numbers, sum $> 2^{n-1}$

$+5 \rightarrow 0\ 1\ 0\ 1$

$+6 \rightarrow 0\ 1\ 1\ 0$

$+11 \quad 1\ 0\ 1\ 1 \rightarrow$ Wrong Answer, because of overflow.

Case-3 : Addition of +ve and -ve numbers, -ve > +ve

$+5 \rightarrow 0101$

$-6 \rightarrow 1001$

$-1 \qquad 1110 \rightarrow$ Correct Answer

$1 \rightarrow$ -ve

$110 \rightarrow$ Result = 1's comp of 110 = 001

Case-4 : Addition of +ve and -ve numbers, +ve > -ve.

$-5 \rightarrow 1010$

$+6 \rightarrow 0110$

$+1 \qquad (1)0000$

$1 \rightarrow +1$

$0001 \rightarrow$ correct answer.

End-around carry

$001 \rightarrow +1$

Case-5 : Addition of 2 negative numbers where the sum is lying within the max. range, $|sum| \leq 2^{n-1}$

$-3 \rightarrow 1100$

$-4 \rightarrow 1011$

$-7 \qquad (1)0111$

$1 \rightarrow +1$

$1000 \rightarrow$ correct answer.

End-around carry.

1's $\rightarrow$ -7

Case-6 : Add of 2 -ve nos, $|sum| > 2^{n-1}$

$-5 \rightarrow 1010$

$-6 \rightarrow 1001$

$-11 \qquad (1)0011$

$1 \rightarrow +1$

End-around carry

$0100 \rightarrow$ wrong answer because of overflow.

$100 \rightarrow +4.$

Solve the following using 1's Complement :-

Let,
$n = 8$

① Add -11 & -20

② Sub 14 from 25

③ Add -25 to $+14$

④ Add -25 to -14

⑤ Add $+25$ to $+14$

⑥ Add $+25$ to -14

① $-11 \rightarrow$
$-20 \rightarrow$
-31

1.5 => Binary Codes

- These are used to modify the transmitted information before it is getting transmitted through any media.
- There are f° 2 different codes :-

① 8421

② BCD

③ Excess-3.

④ Gray Code

⑤ 6311

⑥ 2 out of 5.

Binary codes can be classified as numeric codes & alpha-numeric codes.

Numeric codes are codes which represent numeric information that is only numbers as a series of 0's and 1's.

Ex - 8421, XS-3, Gray Code, etc.

Numeric codes which is used to represent decimal digits are called binary-coded decimal that is BCD code.

Alpha-numeric codes are codes which represent alpha-numeric information i.e; letters of the alphabet and decimal numbers as a sequence of 0's & 1's.

Ex - EBCDIC (Extended binary coded decimal interchange code)

- ASCII (American standard code for information interchange)

Imp

① BCD (Binary Code Decimal) Code :-

In this code each decimal digit i.e 0 to 9 is coded by a 4-bit binary number. The main advantage of this code is its ease of conversion to and from decimal.

Eg :- ① 8 → 1000

② 7 6 → 0111 0110

③ 9 3 7 . 2 5 → 0101

1001 0011 0111 . 0010

Let us consider a 4-bit binary value i.e; n=4.

n=4.

Decimal	8421	BCD
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	10000
11	1011	10001
12	1100	10010
13	1101	10011
14	1110	10100
15	1111	10101

It is less efficient than pure binary because it requires more bits. Another disadvantage is that the arithmetic operations are more complex than they are in pure binary

* There are six illegal combinations that is 1010, 1011, 1100, 1101, 1110 and 1111

Q. Perform the BCD addition on the following :-

① 25 → 0010 0101
+ 13 0001 0011
38 0011 1000 → correct answer.
3 8

② 79 → 0111 1001
+ 43 0100 0011
122 1011 1100 → This is the invalid sum. We need to provide a correction factor of 6 that is CF = 0110 to all the sum values 12 the sum>9.
11 12
+ 0110 0110
6 ← 10010 0010
1 2 2 → correct answer

54 → 0101 0100
+ 17 0001 0111
71 0110 1011 → Invalid sum
0110 → CF of 6
0111 0001 → correct answer
7 1

(2) **Gray Code**: It is a non-weighted code. It is not suitable for arithmetic operations. It is not a BCD code. It is a cyclic code because the successive code words in this code differ in 1-bit position only i.e., it is a unit distance code. It is a reflective code.

Let $n = 4$

Decimal	8 4 2 1	Gray Code
0	0000	0000 → 0
1	0001	0001 → 1
2	0010	0011 → 3
3	0011	0010 → 2
4	0100	0110 → 6
5	0101	0111 → 7
6	0110	0101 → 5
7	0111	0100 → 4
8	1000	1100 → 12
9	1001	1101 → 13
10	1010	1111 → 15
11	1011	1110 → 14
12	1100	1010 → 10
13	1101	1011 → 11
14	1110	1001 → 9
15	1111	1000 → 8

1-bit	2-bit	3-bit	4-bit
0	00	000	0000
1	01	001	0001
	11	011	0011
	10	010	0010
		110	0110
		111	0111
		101	0101
		100	0100
			1100
			1101
			1111
			1110
			1010
			1011
			1001
			1000

axis

The 'n'-bit gray code can be obtained by reflecting the 'n-1'-bit code about an axis at the end of the code and putting the MSB of 0 above the axis & the MSB of 1 below the axis.

Convert in following binary number in Gray Code :-

Binary : 1 0 0 1

Gray : 1 1 0 1

Convert the gray code 1101 to binary code :-

1 1 0 1

1 0 0 1

③ Excess-3/XS-3 : It is a non-weighted BCD code. Here each binary code word is the corresponding 8421 code plus 0011 (3). It is a sequential code and therefore, it can be used for the arithmetic operations. It is a self-complimenting code.

For $n=4$, it has 6 no. of invalid states.

There are six invalid can i.e; 0, 1, 2, 13, 14 & 15

Find the excess-3 code of the following number :-

① $(13)_{10} = (0100\ 0110)_{XS-3}$

$$
\begin{array}{r}
0001 \quad 0011 \\
+\ 0011 \quad 0011 \\
\hline
0100 \quad 0110
\end{array}
$$

Decimal	XS-3	Binary
0	0011	0011
1	0100	0100
2	0101	0101
3	0110	0110
4	0111	0111
5	1000	1000
6	1001	1001
7	1010	1010
8	1011	1011
9	1100	1100

(4) 2 out of 5 code - It has the property that creates 2 out of 5 code as one for every valid code combination. It is a non-weighted code hence the decimal value of coded digit cannot be computed by using formula. This code is useful for error checking because any one of two bits in due to malfunction of a logic circuit then the no. of one bits is no longer exactly 2.

Decimal	2 out of 5 code
0	0 0011
1	0 0101
2	0 0110
3	01001
4	01010
5	01100
6	1 0001
7	1 0010
8	1 0100
9	1 1000

*	
0	1-0
1	2-0
2	2-1
3	3-0
4	3-1
5	3-2
6	4-0
7	4-1
8	4-2
9	4-3

(5) 6311 Codes – These are weighted codes. A 4-bit weighted code has a property that if the weights are w_3, w_2, w_1, w_0 then the code ~~are~~ $a_3\ a_2\ a_1\ a_0$ represents a decimal number :. one.

$$N = w_3 a_3 + w_2 a_2 + w_1 a_1 + w_0 a_0.$$

$w_3 = 6$
$w_2 = 3$
$w_1 = 1$
$w_0 = 0$

Decimal	8421 (BCD)	XS-3	6311	Gray Code	2 out of 5
0	0000	0011	0000	0000	00001
1	0001	0100	0001	0001	00101
2	0010	0101	0011	0011	00110
3	0011	0110	0100	0010	01001
4	0100	0111	0101	0110	01010
5	0101	1000	0111	0111	01100
6	0110	1001	1000	0101	10001
7	0111	1010	1001	0100	10010
8	1000	1011	1011	1100	10100
9	1001	1100	1100	1101	11000

Ch-2, 3 & 4

Boolean Algebra

2.1 : Introduction

Switching circuits are also called the logic circuits, gate circuits or digital circuits. Hence switching algebra or switching expressions is also called Boolean algebra or Boolean expression.

The George Bool developed this Boolean Algebra in 1847 and used it to solve problems in mathematical logic. It has many applications such as Set theory, Mathematical logic, switching circuits, etc.

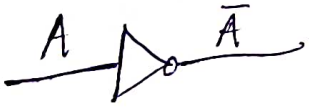
The symbols 0 & 1 used in Boolean algebra do not have a numeric value rather they represent two different states in a logic circuit i.e; they represent two switching variables

0 → Represents a range of low voltage.

1 → Represents a range of high voltage.

2.2 → Basic Operations

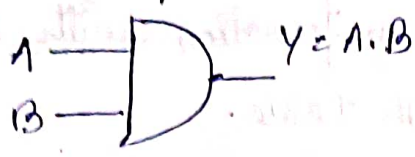
① Not / Complement / Inverse operation



T. Table

A	Y
0	1
1	0

② AND operation



T.T

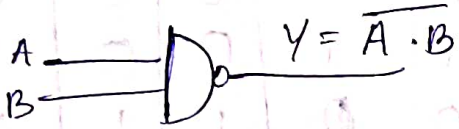
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

③ OR operation

T.T

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

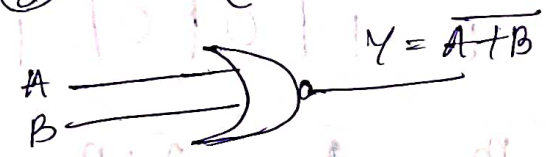
④ NAND (NOT + AND)



T.T

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

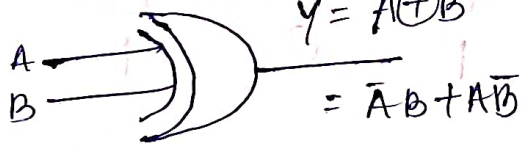
⑤ NOR (NOT + OR)



T.T

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

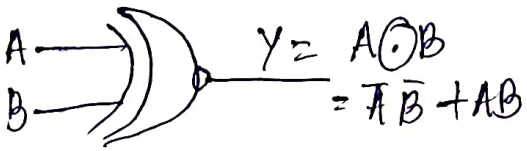
⑥ XOR / Exclusive - OR / EX - OR



T.T

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

⑦ XNOR (Exclusive - NOR / X - NOR)



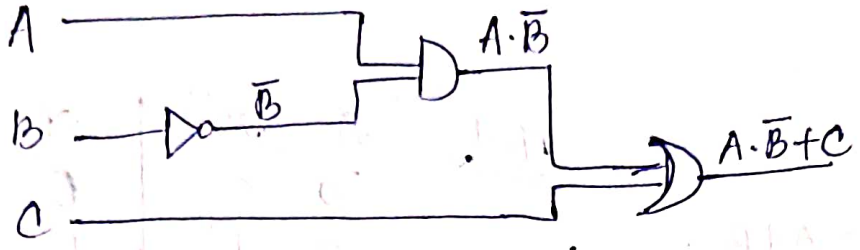
T.T

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

Boolean Expressions and Truth Tables

① Let $F = A\cdot\bar{B}+C$ implement the given function with AOI logic / Basic logic & obtain the truth table.

$+ \rightarrow$ OR Gate
$\cdot \rightarrow$ AND Gate
$- \rightarrow$



Truth Table

$F(A, B, C)$

$n \rightarrow$ no. of i/p variable

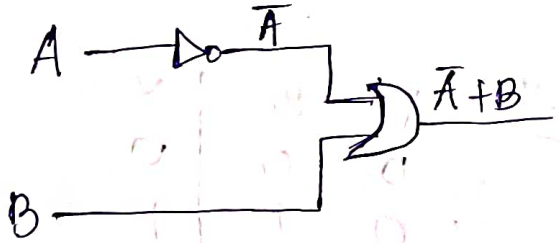
$2^n \rightarrow$ possible combinations

Range : $(0$ to $2^n-1)$

$n=3$, $2^3=8$ states (0 to 7)

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

② $F = \bar{A}+B$

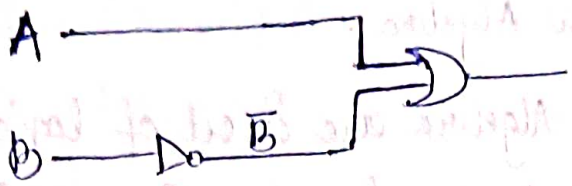


$n=2$

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

③ $F = A\bar{B}+C$

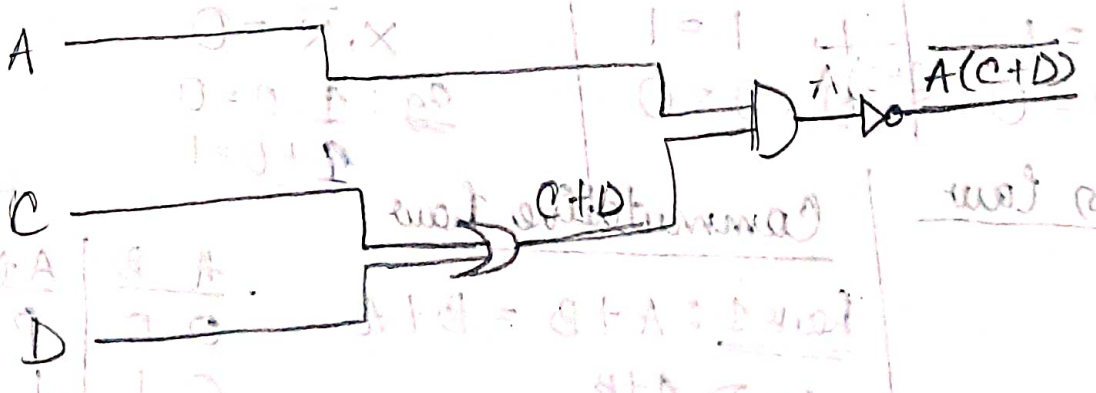
(4) $F = (A+C)(\overline{B}+C)$



C.

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

(5) $F = A(C+D) + BE$



Basic Theorems

Axioms and Laws of Boolean Algebra.

Axioms or Postulates of Boolean Algebra are a set of logical expressions that we accept without proof and using which we can build a set of useful theorems.

Operations with 0 & 1 :-

$X + 0 = X$ ⇒ '0' is additive identity

$X + 1 = 1$ (Due to law of absorption 1 is the max. limit in digital logic design which absorbs everything. So, any number added to 1 is 1.)

$X \cdot 1 = X$ ⇒ '1' is multiplicative identity

$X \cdot 0 = 0$

Idempotent Laws

$X + X = X$	$X \cdot X = 0$
$1 + 1 = 1$	$1 \cdot 1 = 1$
$0 + 0 = 0$	$0 \cdot 0 = 0$

Laws of Complementarity

$X + \overline{X} = 1$

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

Eg : $1 \cdot 0 = 0$

$1 + 0 = 1$

Involution Law

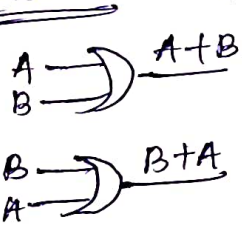
$\overline{\overline{X}} = X$

$\overline{1} = 1$

$\overline{0} = 0$

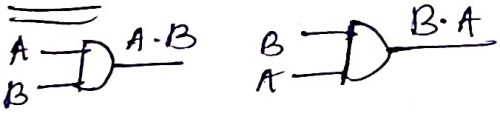
Commutative Law

Law 1 : $A + B = B + A$



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

Law 2 : $A \cdot B = B \cdot A$



AB	A·B	B·A
00	0	0
01	0	0
10	0	0
11	1	1

It is applicable for n no. of inputs also.

Also, $A + B + C = A + C + B = B + C + A = B + A + C$

Associative law

L1 : $(A+B)+C = A+(B+C)$ →

A	B	C	(A+B)	(B+C)	(A+B)+C	A+(B+C)
0	0	0				

L2 : $(A \cdot B) \cdot C = A \cdot (B \cdot C)$

↓

A	B	C	(A·B)	(B·C)	(A·B)·C	A·(B·C)

Distributive law

L1: $A \cdot (B+C) = AB + AC$

L2: $A + BC = (A+B)(A+C)$

Uniting

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

$LHS = AB + A\bar{B} = A(\cancel{B+\bar{B}})^{1} = A.$

Absorption law

L1 : $A + AB = A$

$LHS = A + AB$

$= A(\cancel{1+B})^{1}$

$= A \cdot 1 = A$

L2 : $A(A+B) = A$

$LHS = A(A+B)$

$= A \cdot A + AB$

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

$= A$

Elimination or Redundant Literal Rule (RLR) :-

L1 : $A + \bar{A}B = A + B$

$LHS = A + \bar{A}B = (\cancel{A+\bar{A}})^{1}(A+B) = (A+B)$

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

$LHS = A(\bar{A}+B) = (\cancel{A\bar{A}})^{0} + AB = AB$

Consensus Theorem or Included Factor theorem :-

Theorem 1 :-

L1 : $AB + \bar{A}C + BC = AB + \bar{A}C$

$LHS = AB + \bar{A}C + BC \cdot 1 = AB + \bar{A}C + BC(\bar{A}+A)$

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

$= AB(\cancel{1+C})^{1} + \bar{A}C(\cancel{1+B})^{1} = AB + \bar{A}C = RHS.$

Theorem 2 :-

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

Transposition Theorem

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

$RHS = (A+C)(\bar{A}+B) = \cancel{A\bar{A}}^{0} + AB + \bar{A}C + BC$

$\Rightarrow AB + \bar{A}C + BC \cdot 1$

$\Rightarrow AB + \bar{A}C + BC(A+\bar{A})$

$\Rightarrow AB + \bar{A}C + ABC + \bar{A}BC = AB(\cancel{1+C})^{1} + \bar{A}C(\cancel{1+B})^{1} = AB + \bar{A}C$

$= LHS.$

De Morgan's Law :-

L1 : $\overline{A+B} = \overline{A} \cdot \overline{B}$

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

L2 : $\overline{A \cdot B} = \overline{A} + \overline{B}$

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

Duality :-

Given a boolean algebra expression, the dual of an expression is obtained by interchanging the constants 0 & 1 & interchanging the operations of AND & OR. Variables and complements are left unchanged.

Given Expression	Dual
$\overline{0} = 1$	$\overline{1} = 0$
$0 \cdot 1 = 0$	$1 + 0 = 1$
$A \cdot 1 = A$	$A + 0 = A$
$A \cdot \overline{A} = 0$	$A + \overline{A} = 1$
$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A+B} = \overline{A} \cdot \overline{B}$
$\overline{AB} + \overline{A} + AB = 0$	$\overline{A+B} \cdot \overline{A} \cdot (A+B) = 1$
$AB = AB + \overline{A}B + A\overline{B}$	$A + B = (A+B) \cdot (\overline{A}+B) \cdot (A+\overline{B})$
$A + \overline{B}C(A + \overline{B}C) = A + \overline{B}C$	$A \cdot [(\overline{B}+C)(A \cdot (\overline{B}+C))] = A \cdot (\overline{B}+C)$

Complementing Boolean Expression :-

The inverse or complement of any boolean expression can easily be found by successively applying De Morgan's Law.

Obtain the complement of the following :-

(1) $F = A + B$

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

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

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

$\Rightarrow \bar{F} = \overline{\bar{A}+B} + \bar{\bar{C}} = (\bar{\bar{A}}\cdot\bar{B}) + C = A\bar{B} + C$ } Method - 1

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

dual of $F = (\bar{A}\cdot B) + \bar{C}$ } Method - 2

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

* Complement of a function may be obtained by finding the dual first and then w.r.t the variables (complement the variables).

$F = (A\bar{B}+C)\bar{D}+E \Rightarrow$ dual of $F = [((A+\bar{B})\cdot C)+\bar{D}]\cdot E$

Comp of $F = [((\bar{A}+B)\cdot\bar{C})+D]\cdot\bar{E}$

Simplification :-

(1) $F = A+B[AC+(B+\bar{C})D] \rightarrow F = A+B[AC+BD+D\bar{C}]$

$= A + ABC + BBD + B\bar{C}D$ [$\because B\cdot B = B$]

$= A(\cancel{1+BC}) + BD(\cancel{1+\bar{C}}) = A+BD$

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

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

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

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

$= AB\cdot 1 + AC + B\bar{C}$

$= \cancel{AB} + \cancel{AC} + \cancel{B\bar{C}}\,[\cancel{A+\bar{A}}]$

$= AB + AC +$

$= AB[C+\bar{C}] + AAC + B\bar{C}$

$= ABC + AB\bar{C} + AAC + B\bar{C}$

$= AC[B+1] + B\bar{C}[1+A]$

$= AC + B\bar{C}$

Multiplying Out & Factoring

Minterms :- Each 'AND' term is called standard product. If n is the number of input variable then we have 2^n numbers of minterm.

$n \to 2^n \to$ minterms

$\Sigma \to$ sum
$\Pi \to$ product

Let, $n = 2$

A	B	F
0	0	$\bar{A}.\bar{B} \to m_0$
0	1	$\bar{A}.B \to m_1$
1	0	$A.\bar{B} \to m_2$
1	1	$A.B \to m_3$

Minterms or Standard product

Maxterms :- Each 'OR' term is known as maxterm or a standard sum. If n is the number of input variable then we have 2^n number of maxterms.

$n \to 2^n \to$ maxterms

Let, $n = 2$

A	B	F
0	0	$A+B \to M_0$
0	1	$A+\bar{B} \to M_1$
1	0	$\bar{A}+B \to M_2$
1	1	$\bar{A}+\bar{B} \to M_3$

Maxterms or Standard sum

Note : The boolean functions expressed as a sum of its min-terms or a product of its max-terms are called canonical form else it is known as standard form.

$F = \bar{A}\bar{B} + AB \longrightarrow$ Canonical form

$F = m_0 + m_3$
$= \Sigma m(0, 3)$

$F = A + A\bar{B} \longrightarrow$ Standard form

			MinTerms		MaxTerms	
x	y	z	Terms	designation	Terms	designation
0	0	0	$\bar{x}\,\bar{y}\,\bar{z}$	m_0	$x+y+z$	M_0
0	0	1	$\bar{x}\,\bar{y}\,z$	m_1	$x+y+\bar{z}$	M_1
0	1	0	$\bar{x}\,y\,\bar{z}$	m_2	$x+\bar{y}+z$	M_2
0	1	1	$\bar{x}\,y\,z$	m_3	$x+\bar{y}+\bar{z}$	M_3
1	0	0	$x\,\bar{y}\,\bar{z}$	m_4	$\bar{x}+y+z$	M_4
1	0	1	$x\,\bar{y}\,z$	m_5	$\bar{x}+y+\bar{z}$	M_5
1	1	0	$x\,y\,\bar{z}$	m_6	$\bar{x}+\bar{y}+z$	M_6
1	1	1	$x\,y\,z$	m_7	$\bar{x}+\bar{y}+\bar{z}$	M_7

Note: The minterms and the maxterms are complement of each other that is $\boxed{m_i = \overline{M_i}}$ and vice-versa.

There are two types of standard forms:-

① Sum of Products (SOP)

or OR of AND terms

② Product of Sums (POS)

or AND of OR terms.

Express the given function as sum of minterms:-

① $F = \bar{x}\bar{y}z + x\bar{y}\bar{z} + xyz + \bar{x}yz$ → Canonical form

$= 001 \cdot 100 \quad 111 \quad \cancel{100}\ 011$

$m_1 \quad m_4 \quad m_7 \quad m_3$

$F = m_1 + m_3 + m_4 + m_7$

or, $\boxed{F = \Sigma m(1, 3, 4, 7)}$

② $F = \bar{x}yz + \bar{x}\bar{y}\bar{z} + \bar{x}y\bar{z} + x\bar{y}z \rightarrow$ Canonical form

$= 011 \quad 000 \quad 010 \quad 101$

$= m_3 \quad m_0 \quad m_2 \quad m_5$

$F = m_0 + m_2 + m_3 + m_5$

$F = \sum m(0,2,3,5)$

③ $F = x + yz \rightarrow$ Standard form

$= x \cdot 1 + yz \cdot 1$ (take a multiplicative identity i.e; 1)

$= x \cdot (y + \bar{y}) + yz(x + \bar{x})$

$= xy + x\bar{y} + xyz + \bar{x}yz$

$= xy \cdot 1 + x\bar{y} \cdot 1 + xyz + \bar{x}yz$

$= xy(z + \bar{z}) + x\bar{y}(z + \bar{z}) + xyz + \bar{x}yz$

$= xyz + xy\bar{z} + x\bar{y}z + x\bar{y}\bar{z} + xyz + \bar{x}yz$

$= 111 \quad 110 \quad 101 \quad 100 \quad 111 \quad 011$

$= m_7 \quad m_6 \quad m_5 \quad m_4 \quad m_7 \quad m_3$ (∵ If any minterm appear twice we will keep one & discard the other)

$F = m_3 + m_4 + m_5 + m_6 + m_7$

$\boxed{F = \sum m(3,4,5,6,7)}$

④ $f = a + \bar{b}c$

$= a \cdot 1 + \bar{b}c \cdot 1$

$= a(b + \bar{b}) + \bar{b}c(a + \bar{a})$

$= ab + a\bar{b} + a\bar{b}c + \bar{a}\bar{b}c$

$= ab \cdot 1 + a\bar{b} \cdot 1 + a\bar{b}c + \bar{a}\bar{b}c$

$= ab(c + \bar{c}) + a\bar{b}(c + \bar{c}) + \bar{a}\bar{b}c + \bar{a}\bar{b}\bar{c}$

$= abc + ab\bar{c} + a\bar{b}c + a\bar{b}\bar{c} + a\bar{b}c + \bar{a}\bar{b}c$

$= m_7 + m_6 + m_5 + m_4 + m_5 + m_1$

$f = m_1 + m_4 + m_5 + m_6 + m_7$

$f = \sum m(1,4,5,6,7)$

Express the following function as the product of maxterm form :-

① $F = (x+y+z)(\bar{x}+y+\bar{z})(\bar{x}+y+z) \rightarrow$ canonical form

000 101 100

M_0 M_5 M_4

$F = M_0 \cdot M_4 \cdot M_5$

or $F = \pi M(0,4,5)$

② $F = xy + \bar{x}z \rightarrow$ standard form.

$= (xy+\bar{x})(xy+z)$ (∵ Distributive law)

$= (x+\bar{x})^{1}(y+\bar{x})(x+z)(y+z)$

$= (\bar{x}+y)(x+z)(y+z) \Rightarrow$ factors of the function.

$\bar{x}+y+0 = \bar{x}+y+z\bar{z} = \underbrace{(\bar{x}+y+z)}_{M_4}(\bar{x}+y+\bar{z})_{M_5}$

$x+z+0 = x+z+y\cdot\bar{y} = \underbrace{(x+y+z)}_{M_0}(x+\bar{y}+z)_{M_2}$

$y+z+0 = y+z+x\cdot\bar{x} = \underbrace{(x+y+z)}_{M_0}(\bar{x}+y+z)_{M_4}$

$F = M_4 \cdot M_5 \cdot M_0 \cdot M_2 \cdot M_0 \cdot M_4$

$F = \pi M(0,2,4,5)$

Find the truth table of $F = xy + \bar{x}z$

Not to do in exam only for verification

x	y	z	xy	$\bar{x}$	$\bar{x}z$	$xy+\bar{x}z$
0	0	0	0	1	0	0
0	0	1	0	1	1	1
0	1	0	0	1	0	0
0	1	1	0	1	1	1
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	0	1
1	1	1	1	0	0	1

$F = \Sigma m(1,3,6,7)$

$F = \pi M(0,2,4,5)$

Q. Express the given function as the product of maxterm :-

① $F = \bar{x}y + xz$

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

$= (\bar{x} + x)(y + x) \cdots (\bar{x} + z)(y + z)$

$= (x + y + 0)(\bar{x} + 0 + z)(0 + y + z)$

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

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

$\Rightarrow F = M_0 \; M_1 \; M_4 \; M_6 \; M_0 \; M_4$

$\Rightarrow F = M_0 \cdot M_1 \; M_4 \; M_6$

Or, $\boxed{F = \Pi M(0, 1, 4, 6)}$

② $\Sigma m(1, 2, 3, 5, 7)$

$2^3 = 8$ bits so, 0 to 7.

Since minterms and maxterms are complement to each other hence the max terms are given as,

$F = \Pi M(0, 4, 6)$.

Q. $F = \Pi M(1, 3, 4, 6, 8, 9)$

$2^4 = 16$ bits so, 0 to 15

$F = \Sigma m(0, 2, 5, 7, 10, 11, 12, 13, 14, 15)$

Exclusive - OR (XOR) & Equivalence function :-

① XOR | Exclusive - OR | Ex - OR :-

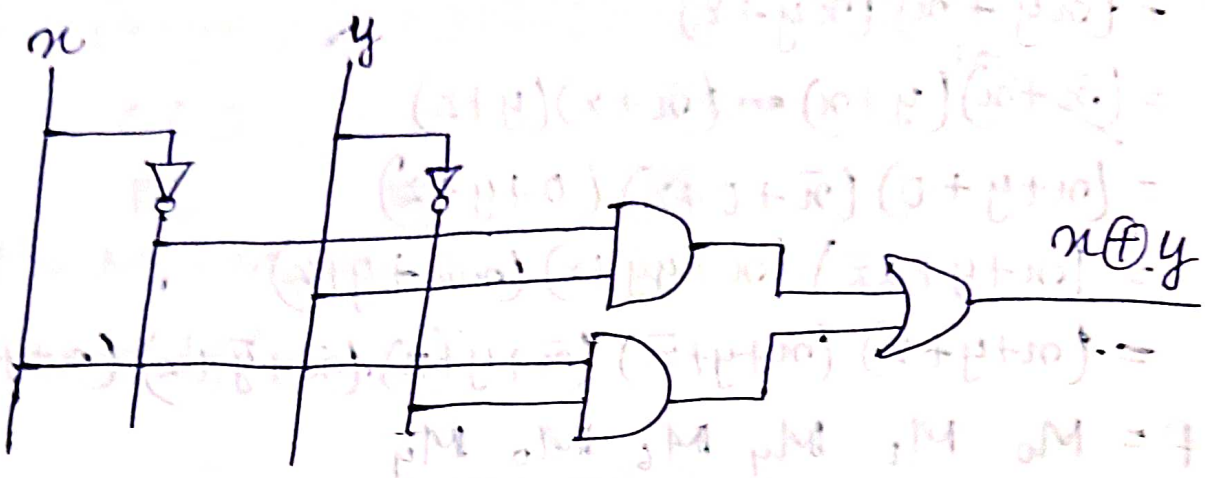
Let us consider a 2 variable XOR operation defined as follows

x	y	F
0	0	0
0	1	1
1	0	1
1	1	0

$\boxed{F = \bar{x} \cdot y + x \cdot \bar{y}}$

$\boxed{F = x \oplus y}$

Circuit Diagram :-



2) XNOR / Exclusive - NOR :-

x	y	F
0	0	1
0	1	0
1	0	0
1	1	1

$$\boxed{F = \bar{x} \cdot \bar{y} + xy}$$

$$\boxed{F = x \odot y}$$

Circuit Diagram :-

Q. Find the complement of a two-input XOR function.

$F = x \oplus y$

$\bar{F} = \overline{\bar{x}y + x\bar{y}}$

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

$= (\bar{x} + \bar{y})(\bar{x} + \bar{y})$

$= (x + \bar{y})(\bar{x} + y)$

$= \cancel{(x\bar{x})}^{\to 0} + xy + \bar{x}\bar{y} + \bar{y}y$

$= \bar{x}\bar{y} + xy = x \odot y.$

3) Let us consider a 3 input XOR function :-

x	y	z	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

$\boxed{\text{If no. of 1 is odd then 1 else 0}}$

Q. $F = \Sigma m(1, 2, 4, 7)$

$= \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$

$= \bar{x}(\bar{y}z + y\bar{z}) + x(\bar{y}\bar{z} + yz)$

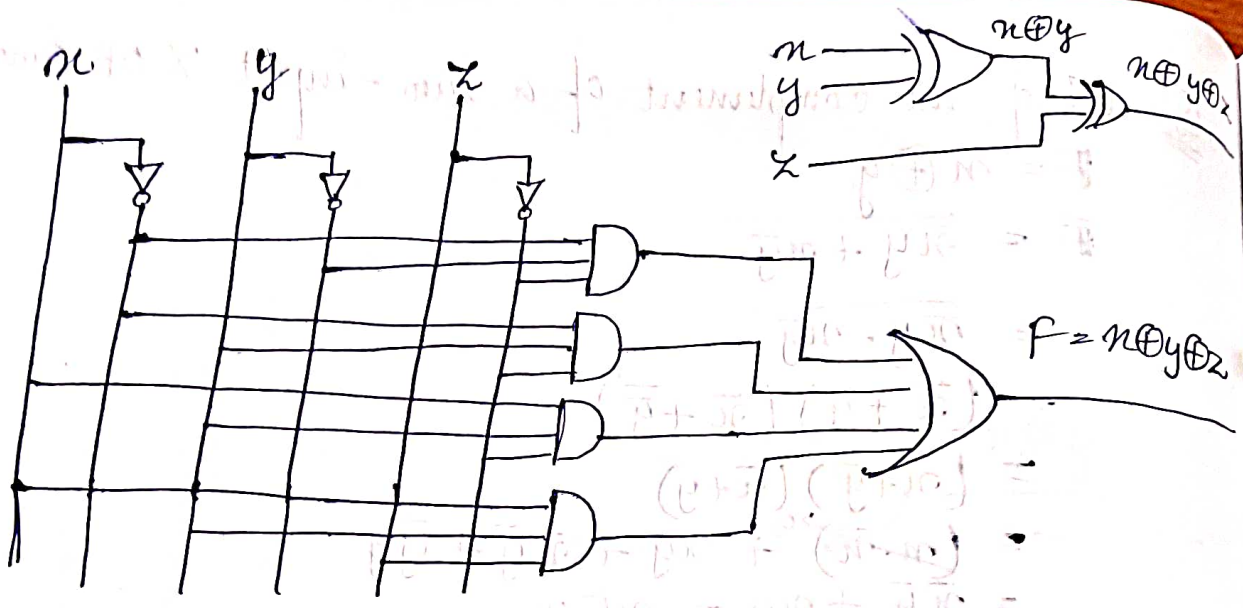
$= \bar{x}(y \oplus z) + x(y \odot z)$

$\bar{a} \quad b \qquad a \quad \bar{b}$

$= a \oplus b$

$= x \oplus y \oplus z$

$\Rightarrow \boxed{F = x \oplus y \oplus z}$


x
y
z
x
y
z
x⊕y
x⊕y⊕z
F = x⊕y⊕z

4-i/p XOR function

a	b	c	d	F
0	0	0	0	→ 0
0	0	0	1	→ 1
0	0	1	0	→ 1
0	0	1	1	→ 0
0	1	0	0	→ 1
0	1	0	1	→ 0
0	1	1	0	→ 0
0	1	1	1	→ 1
1	0	0	0	→ 1
1	0	0	1	→ 0
1	0	1	0	→ 0
1	0	1	1	→ 1
1	1	0	0	→ 0
1	1	0	1	→ 1
1	1	1	0	→ 1
1	1	1	1	→ 0

$f = \Sigma m(1,2,4,7,8,11,13,14)$

$= \bar{a}\bar{b}\bar{c}d + \bar{a}\bar{b}c\bar{d} + \bar{a}b\bar{c}\bar{d} + a\bar{b}cd + a\bar{b}\bar{c}\bar{d} + a\bar{b}\bar{c}d + ab\bar{c}d + abc\bar{d}$

$= \bar{a}\bar{b}(\bar{c}d + c\bar{d}) + \bar{a}b(\bar{c}\bar{d} + cd) + a\bar{b}(\bar{c}\bar{d} + cd) + ab(\bar{c}d + c\bar{d})$

$= a\bar{b}(c \oplus d) + \bar{a}b(c \odot d) + a\bar{b}(c \odot d) + ab(c \oplus d)$

$= (c \oplus d)(\bar{a}\bar{b} + ab) + c \odot d(\bar{a}b + a\bar{b})$

$= (c \oplus d)(a \odot b) + (c \odot d)(a \oplus b)$

$= (a \odot b)(c \oplus d) + \underbrace{(a \oplus b)}_{x}\underbrace{(c \odot d)}_{y}$

$= x \oplus y$

$= a \oplus b \oplus c \oplus d$

4-i/p XNOR 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	→ 1
0	1	1	0	→ 1
0	1	1	1	→ 0
1	0	0	0	→ 0
1	0	0	1	→ 1
1	0	1	0	→ 1
1	0	1	1	→ 0
1	1	0	0	→ 1
1	1	0	1	→ 0
1	1	1	0	→ 0
1	1	1	1	→ 1

$f = \Sigma m(0, 3, 5, 6, 9, 10, 12, 15)$

$= \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}cd + \bar{a}b\bar{c}d + \bar{a}bc\bar{d} + a\bar{b}\bar{c}d + a\bar{b}c\bar{d} + ab\bar{c}\bar{d} + abcd$

$= \bar{a}\bar{b}(\bar{c}\bar{d} + cd) + \bar{a}b(\bar{c}d + c\bar{d}) + a\bar{b}(\bar{c}d + c\bar{d}) + ab(\bar{c}\bar{d} + cd)$

$= \bar{a}\bar{b}(c \odot d) + \bar{a}b(c \oplus d) + a\bar{b}(c \oplus d) + ab(c \odot d)$

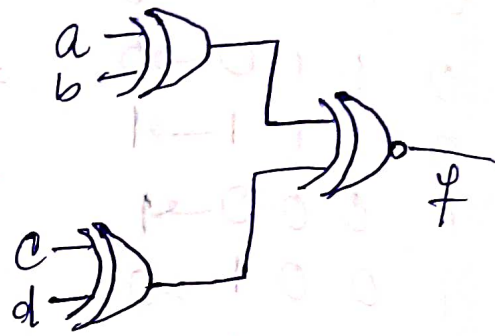
$= (c \odot d)(\bar{a}\bar{b} + ab) + (c \oplus d)(\bar{a}b + a\bar{b})$

$= (c \odot d)(a \odot b) + (c \oplus d)(a \oplus b)$

$= \underbrace{(a \oplus b)}_{x}\underbrace{(c \oplus d)}_{y} + \underbrace{(a \odot b)}_{\bar{x}}\underbrace{(c \odot d)}_{\bar{y}}$

$= x \odot y$

$$\boxed{f = (a \oplus b) \odot (c \oplus d)}$$



Find the complement of a 4 i/p XOR function

$f = \underline{a \oplus b \oplus c \oplus d}$

$\bar{f} = a \oplus b \oplus c \oplus d$

Let, $a \oplus b = x$

$c \oplus d = y$

$\bar{f} = \overline{x \oplus y}$

$= x \odot y$

$$\boxed{f = (a \oplus b) \odot (c \oplus d)}$$

Gate-Level Minimization :-

It is needed to optimize the implementation of digital circuit. Here for the optimization we are using the map method.

Map Method :-

This method was introduced by Karnaugh and hence it is known as Karnaugh-Map / K-Map.

The K-Map is a diagram made up of squares where each square represents a minterm.

If n is the number of input variable then we will have 2^n no. of squares and 2^n of Minterms.

* In K-Map the minterms are arranged according to the Gray code values.

Types of Map :-

① 2-Variable Map

x	y	f
0	0	$\bar{x}\bar{y}$
0	1	$\bar{x}y$
1	0	$x\bar{y}$
1	1	xy

x \ y	$\bar{y}$	y
$\bar{x}$	$\bar{x}\bar{y}$	$\bar{x}y$
x	$x\bar{y}$	xy

Binary	
0 0	0
0 1	1
1 0	2
1 1	3

Grey Code	
0 0	0
0 1	1
1 1	3
1 0	2

x \ y	0	1
0	00 m_0	01 m_1
1	10 m_2	11 m_3

Direction to write the gray code.

Grouping

Group of 2 minterms - pair
Group of 4 minterms - Quad
Group of 8 minterms - Octet
Group of 16 minterms - Hex

Q. Minimize the given function

$F = \Sigma m(1, 2, 3)$

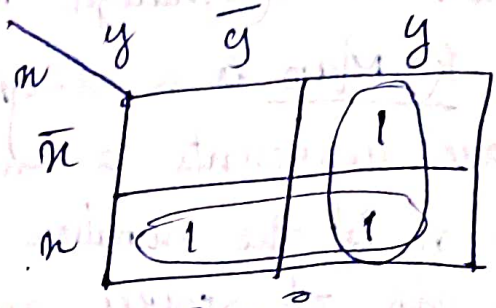
$= \bar{x}\bar{y} + x\bar{y} + xy$

$= \bar{x}y + x(\bar{y} + y)$

$= \bar{x}y + x$

$= (\bar{x} + x)(x + y)$

$= x + y$



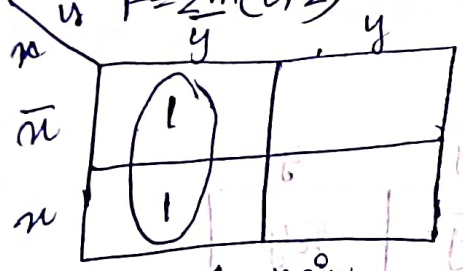
2-pair

$F = x + y$

(constant is taken)

This sign is there as there are 2 pair.

$F = \Sigma m(0, 2)$



1-pair

$\boxed{F = \bar{y}}$

Three-Variable K-Map :-

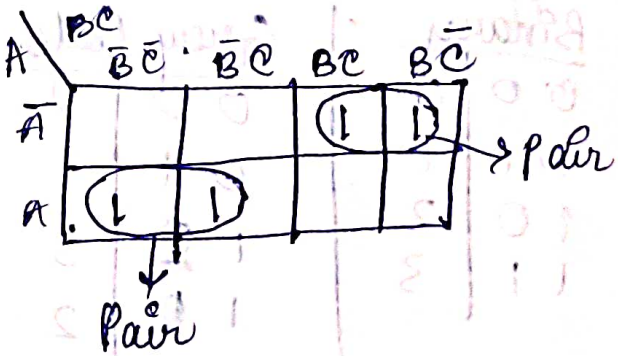
Solve the following function using 3-variable K-Map.

(i) $F = \Sigma m(2, 3, 4, 5)$

$2^3 = 8 \rightarrow$ minterms

$8 \rightarrow$ squares

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0	1	3	2
A	4	5	7	6



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

2) $F = \sum m(3,4,6,7)$

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

$F = BC + A\bar{C}$ (∵ Changing value = discard. Constant value = acceptable)

3) $F = \sum m(0,2,4,6)$

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

→ 1 Quad

$\boxed{F = \bar{C}}$

smp 7) $F = \sum m(0,1,6,7)$

a \ bc	$b\bar{c}$	$\bar{b}c$	bc	$b\bar{c}$
$\bar{a}$	1	1		
a			1	1

$\boxed{F = \bar{a}\bar{b} + ab}$ (XNOR)

4) $F = \sum m(0,1,2,3,4,5,6,7)$

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

→ octet.

$\boxed{F = 1}$

smp 8) $F = \sum m(2,3,4,5)$

a \ bc	$\bar{b}\bar{c}$	$\bar{b}c$	bc	$b\bar{c}$
$\bar{a}$			1	1
a	1	1		

$\boxed{F = \bar{a}b + a\bar{b}}$ (XOR)

5) $F = \sum m(0,2,4,5,6)$

a \ bc				
	1			1
	1	1		1

→ quad → pair

$\boxed{F = \bar{c} + a\bar{b}}$

6) $F = \sum m(1,2,3,5,7)$

a \ bc	$\bar{b}\bar{c}$	$\bar{b}c$	bc	$b\bar{c}$
		1	1	1
		1	1	

$F = C + \bar{a}b$

9) $F = \sum m(0,1,2,5)$

a \ bc	$\bar{b}\bar{c}$	$\bar{b}c$	bc	$b\bar{c}$
$\bar{a}$	1	1		1
a		1		

$\boxed{F = \bar{b}c + \bar{a}\bar{c}}$

10) $F = \Sigma m(0, 2, 3, 4, 6)$

a \ bc	$\bar{b}\bar{c}$	$\bar{b}c$	bc	$b\bar{c}$
$\bar{a}$	1		1	1
a	1			1

$F = \bar{c} + \bar{a}b$

Simplify the given function using a suitable map.

n = no. of variable $= 3$.

$2^3 =$ 8 squares

$F = \bar{a}b + abc$

a \ bc				
		1		
	1	1	1	1

$a + \bar{b}c$

a \ bc				
			1	1
			1	

$F = bc + \bar{a}b$

4-variable Map :-

Here, $n = 2^4 = 16$ squares

ab \ cd	00	01	11	10
00	0000	0001	0011	0010
01	0100	0101	0111	0110
11	1100	1101	1111	1110
10	1000	1001	1011	1010

Q1. $F = \sum m(0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)$

ab \ cd	$\bar{a}\bar{b}$	$\bar{a}b$	ab	$a\bar{b}$
$\bar{a}\bar{b}$	1 (0)	1 (1)		1 (2)
$\bar{a}b$	1 (4)	1 (5)		1 (6) → Quad
ab	1 (12)	1 (13)		1 (14) → Quad
$a\bar{b}$	1 (8)	1 (9)		

Q2. $F = \sum m(0, 1, 2, 3, 4, 6, 8, 9, 10, 11)$

1 (0)	1 (1)	1 (3)	1 (2)
1 (4)			1 (6)
1 (8)	1 (9)	1 (11)	1 (10)

Q3. $F = \sum m(3, 5, 7, 8, 10, 11, 13, 15)$

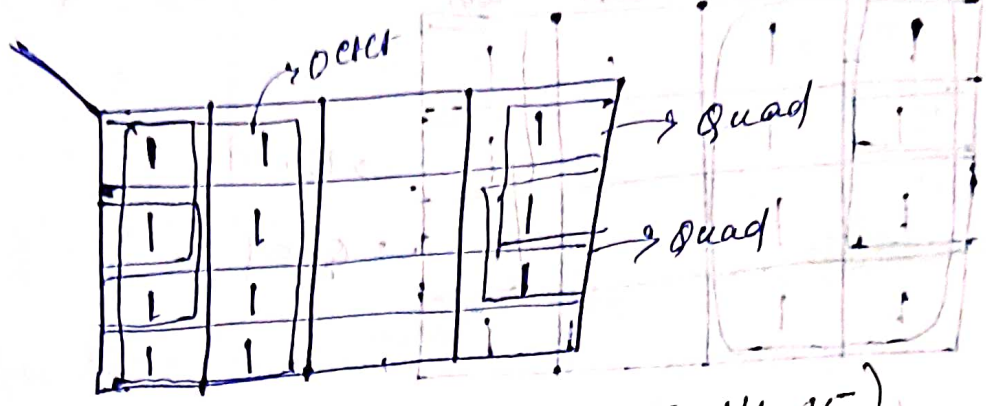
	00	01	11	10
00	0	1	1 (3)	1 (2)
01	4	1 (5)	1 (7)	6
11		1 (13)	1 (15)	
10	1 (8)		1 (11)	1 (10) → pair

Quad Quad

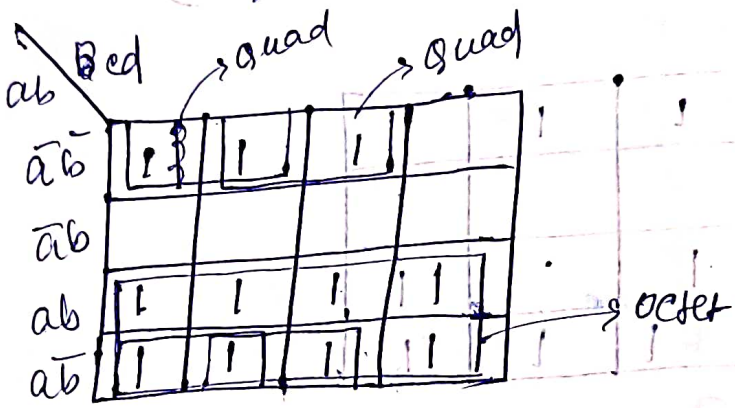
$$F = bd + cd + a\bar{b}\bar{d}$$

Solve the given functions using a K-map approach :-

Q4) $F = \Sigma m(0,1,2,4,5,6,8,9,12,13,14)$

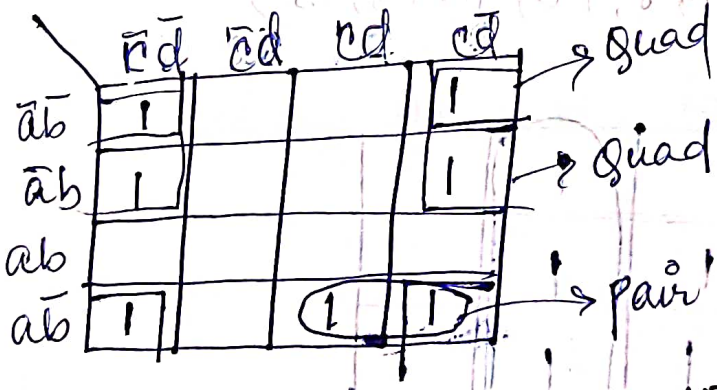


Q5) $F = \Sigma m(0,1,3,8,9,10,11,12,13,14,15)$



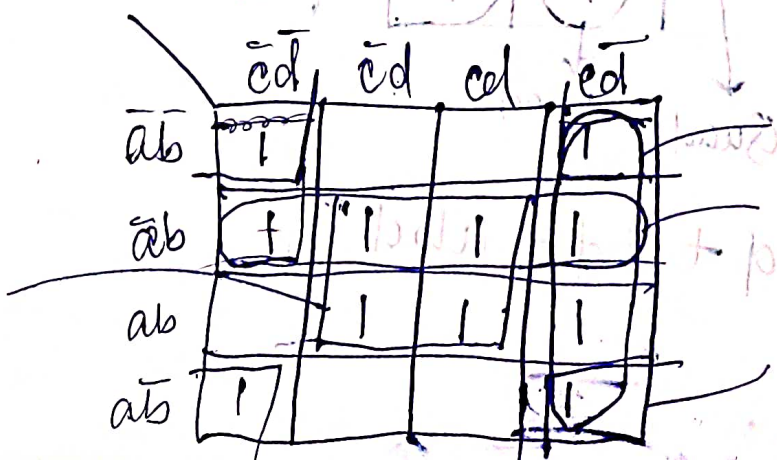
$F = a + \bar{b}\bar{c} + \bar{b}d$

Q6) $F = \Sigma m(0,2,4,6,8,10,11)$



$F = \bar{a}\bar{d} + \bar{b}\bar{d} + a\bar{b}c$

Q7. $F = \Sigma m(0,2,4,5,6,7,8,10,13,14,15)$



Q8. $F = \Sigma m(0, 2, 8, 10, 12, 13, 14)$

ab \ cd	$\bar{c}\bar{d}$	$\bar{c}d$	cd	$c\bar{d}$
$\bar{a}\bar{b}$	1			1
$\bar{a}b$				
ab	1	1		1
$a\bar{b}$	1			1

$F = \bar{b}\bar{d} + a\bar{d} + ab\bar{c}$

→ Sum of products.

Solve the following functions and express it as a product of maxterms using the K-map approach.

$F = \Pi M(1, 3, 4, 5, 6, 7, 9, 11, 15)$

ab \ cd	$c+d$	$c+\bar{d}$	$\bar{c}+\bar{d}$	$\bar{c}+d$
$a+b$		0	0	
$a+\bar{b}$	0	0	0	0
$\bar{a}+\bar{b}$			0	
$\bar{a}+b$		0	0	

$F = (a+\bar{b}) \cdot (\bar{c}+\bar{d}) \cdot (b+\bar{d})$

→ Product of sums.

★ Minterms and Maxterms are complement to each other.

Express the following function as the product of maxterm.

(i) $F = \Sigma m(0, 1, 3, 8, 9, 10, 11, 12, 13, 14, 15)$

⇒ $F = \Pi M(2, 4, 5, 6, 7)$

ab \ cd	$c+d$	$c+\bar{d}$	$\bar{c}+\bar{d}$	$\bar{c}+d$
$a+b$				0
$a+\bar{b}$	0	0	0	0
$\bar{a}+\bar{b}$				
$\bar{a}+b$				

$\boxed{F = a+\bar{b}\,(a+\bar{c}+d)}$

Don't Care :-

It is used in a map for further simplification of boolean function. Don't care min-term is a combination of whose logic values are not satisfied and it is denoted as 'X' thus, a cross inside a square in the map indicates that we don't care whether the value of '0' assigned to the function F for the particular minterm.

Solve the following Boolean function

1) $F = \Sigma m(1, 3, 7, 11, 15)$ with $d = \Sigma m(0, 2, 5)$

X	1	1	X
	X	1	
		1	
		1	

2) $F = \Sigma m(4, 5, 6, 7, 12)$
$d = \Sigma m(0, 8, 13)$

X			
1	1	1	1
1	X		
X			

Prime Implicants and Essential Prime Implicants

A Prime Implicant is a product term obtained by combining the maximum possible number of adjacent squares in the map. The Prime Implicant of the function may be obtained from the map by combining all possible maximum number of squares.

If a minterm in a square is covered by only one prime implicant then that prime implicant is said to be the essential prime implicant that is it cannot be removed from the description of a function.

Repetition → Prime Implicants

No Repetition →

Q. ~~Identify~~ Identify the prime implicant for the following function :-

① $F = \sum m(0, 2, 3, 5, 7, 8, 9, 10, 11, 13, 15)$

ab \ cd	$\bar{c}\bar{d}$	$\bar{c}d$	cd	$c\bar{d}$
$\bar{a}\bar{b}$	1		1	1
$\bar{a}b$		1	1	
ab		1	1	
$a\bar{b}$	1	1	1	1

$F = cd + a\bar{b} + bd + \bar{b}\bar{d}$

The Prime Implicants of F are $cd, a\bar{b}, bd, \bar{b}\bar{d}$

② $F = \sum m(0, 2, 5, 7, 8, 10, 13, 15)$

ab \ cd	$\bar{c}\bar{d}$	$\bar{c}d$	cd	$c\bar{d}$
$\bar{a}\bar{b}$	1			1
$\bar{a}b$		1	1	
ab		1	1	
$a\bar{b}$	1			1

$F = bd + \bar{b}\bar{d}$

Essential Prime Implicants of F are $bd + \bar{b}\bar{d}$.

8) $F = \sum m(0, 2, 4, 5, 7, 8, 10, 13, 15)$

ab \ cd	$\bar{c}\bar{d}$	$\bar{c}d$	cd	$c\bar{d}$
$\bar{a}\bar{b}$	1			1
$\bar{a}b$	1	1	1	
ab		1	1	
$a\bar{b}$	1			1

$F = bd + \bar{b}\bar{d} + \bar{a}b\bar{c}$

P.I = bd, $\bar{a}b\bar{c}$

E.P.I = $\bar{b}\bar{d}$.

NAND Implementation

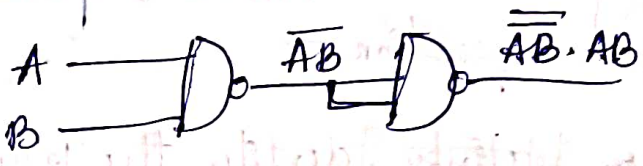
NAND

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

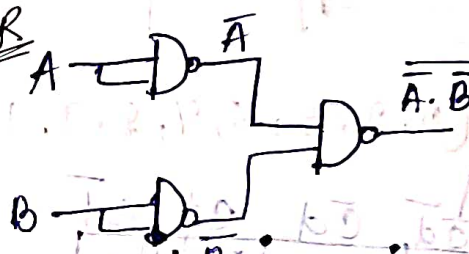
If either A or B = 0
o/p = 1
else o/p = 0.

NOT

AND



OR



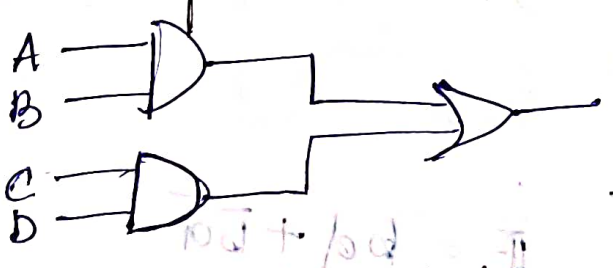
$\overline{\bar{A}.\bar{B}} = \overline{\overline{A+B}}$
$= A+B$

2-level NAND Implementation

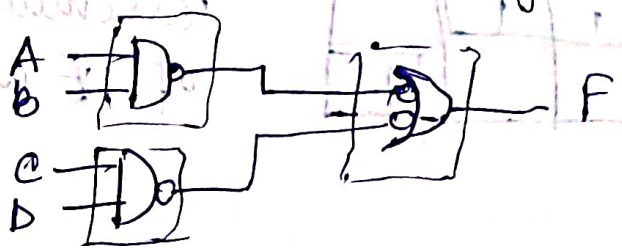
Implement the given function with NAND gates only.

$F = AB + CD$.

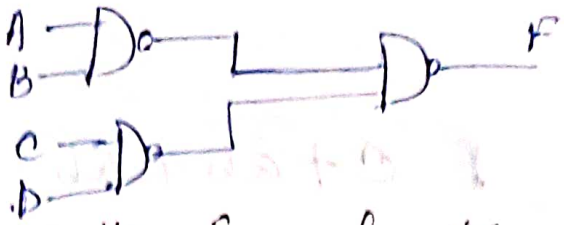
Step-1 : Implement the function using logic gates or AOI logic.



Step-2 : Insert the bubbles that is NOT gates as and when required.



Step-3 :



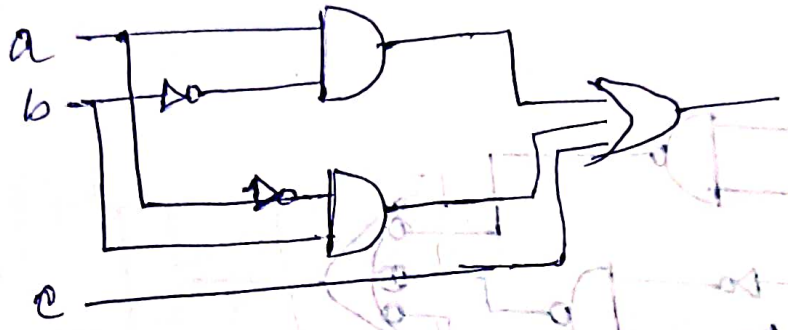
Implement the given function with NAND Gates only.

Q1. $F = \Sigma m(1, 2, 3, 4, 5, 7)$

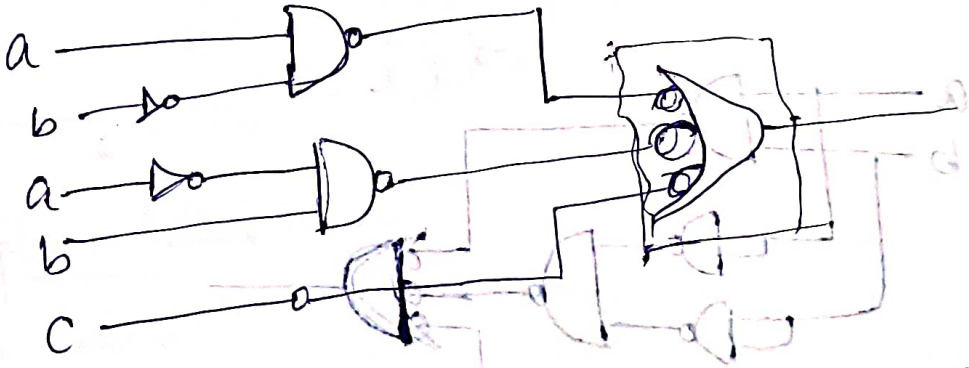
a \ bc	$\bar{b}\bar{c}$	$\bar{b}c$	bc	$b\bar{c}$
$\bar{a}$		1	1	1
a	1	1	1	

$F = \bar{c} + a\bar{b} + \bar{a}b.$

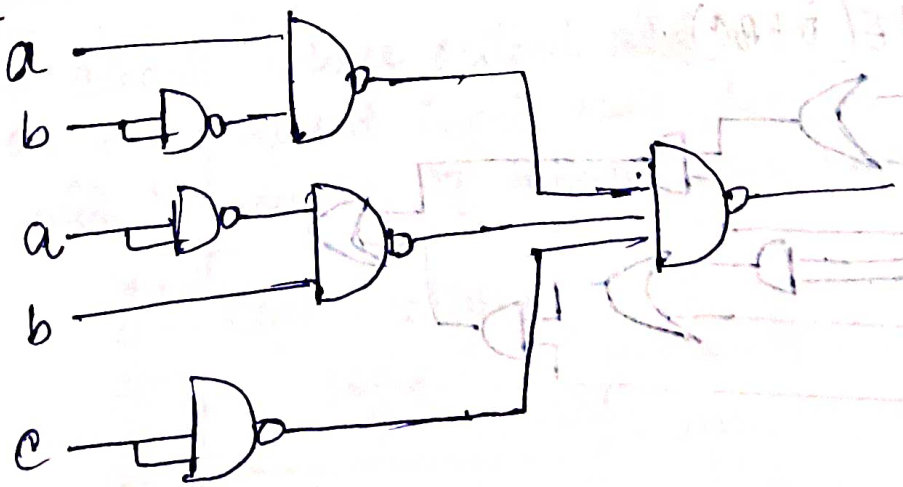
Step-1 :-



Step-2 :-



Step-3 :-

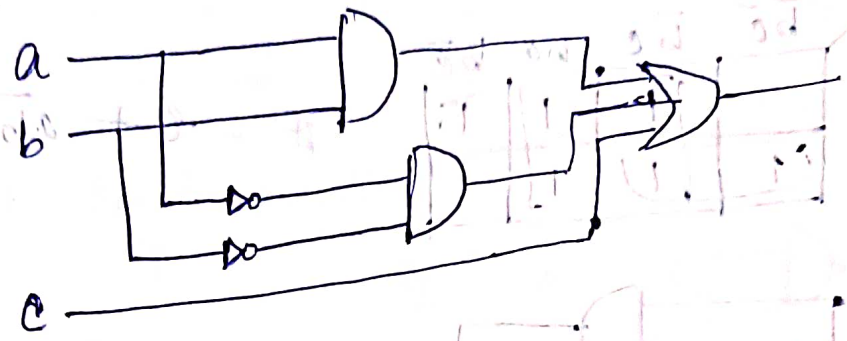


2) $F = \sum m(0,1,3,5,6,7)$

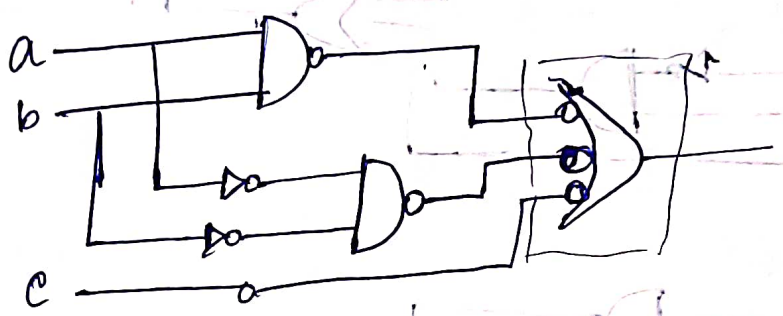
a \ bc	$\bar{b}\bar{c}$	$\bar{b}c$	bc	$b\bar{c}$
$\bar{a}$	1	1	1	
a		1	1	1

$F = c + ab + \bar{a}\bar{b}$

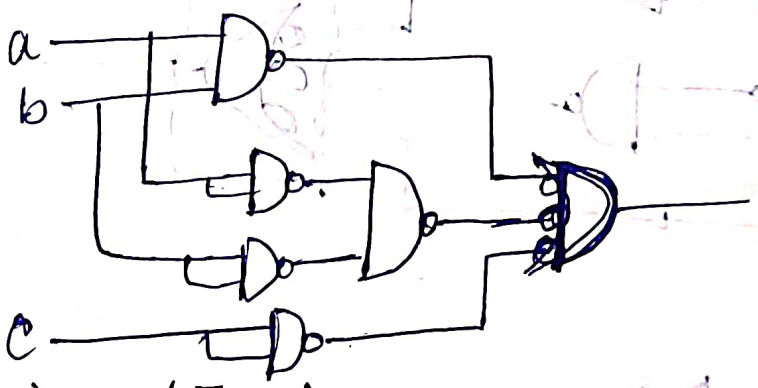
Step-1 :-



Step-2 :-

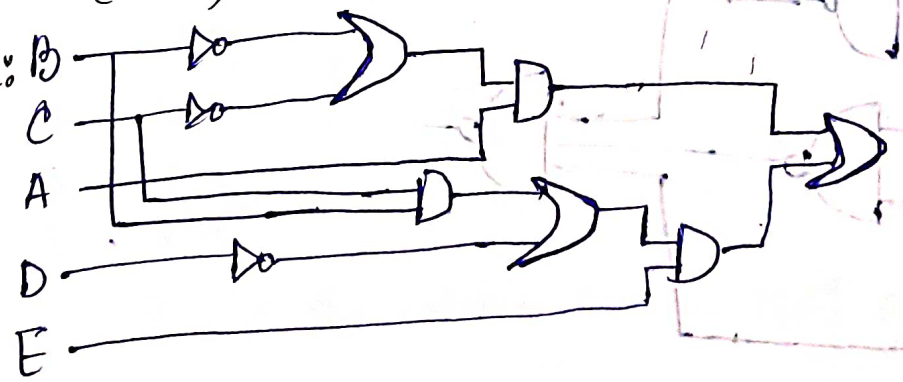


Step-3 :-

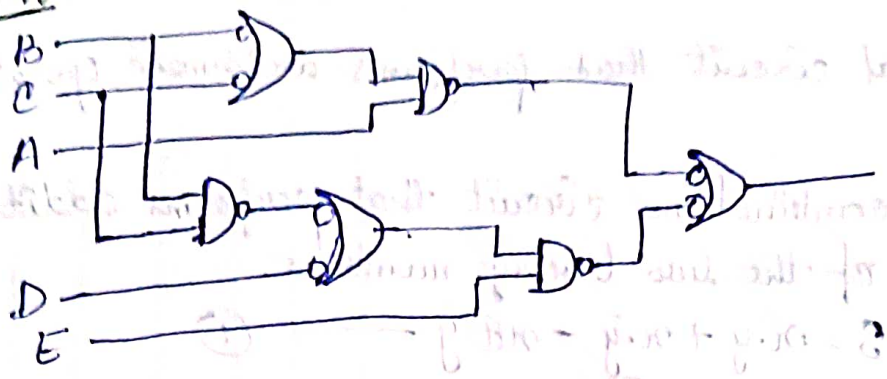


Q3. $F = (\bar{B} + \bar{C})A + E(\bar{D} + BC)$.

Step-1:

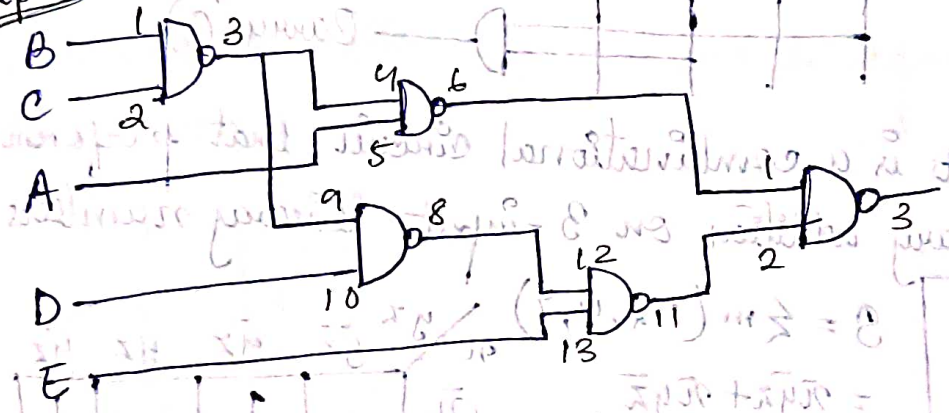


Step-2 :-



Step-3 :-

Step-4 :-

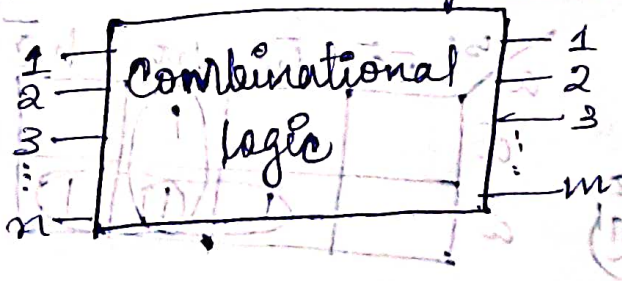


NOR Implementation of a function :-

Refer to lab 5.

Combinational logic :-

It is a circuit whose output at any instance of time will depends on the present input only. Let us consider the following circuit with 'n' number of inputs and 'm' number of outputs.



Binary Adders :-

It is a combinational circuit that performs arithmetic operations on binary numbers.

(i) **Half Adder**: It is a combinational circuit that performs addition operation of the two binary numbers.

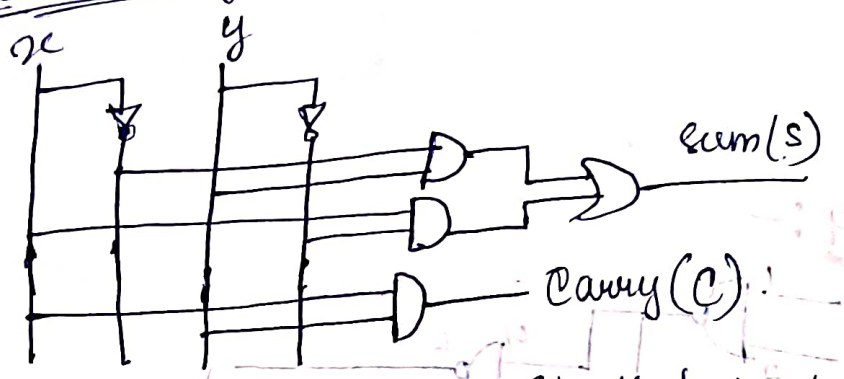
Truth table :-

x	y	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S = \bar{x}\cdot y + x\cdot\bar{y} = x \oplus y$ —— (1)

$C = xy$ —— (2)

Circuit diagram :-



(ii) **Full Adder**: It is a combinational circuit that performs binary addition on 3-input binary numbers.

Truth table :-

x	y	z	s	c
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$S = \Sigma m(1,2,4,7)$

$= \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$

$= z(\bar{x}\bar{y} + xy) + \bar{z}(\bar{x}y + x\bar{y})$

$= \underset{b}{z}(\underset{a}{x \odot y}) + \underset{b}{\bar{z}}(\underset{a}{x \oplus y})$

$= a \oplus b$

$= x \oplus y \oplus z$

$\boxed{S = x \oplus y \oplus z}$ —— (1)

$x \backslash yz$	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{x}$		1		1
x	1		1	

$C = \Sigma m(3,5,6,7)$

$= xz + xy + yz$ —— (2)

or $C = \bar{x}yz + x\bar{y}z + xy\bar{z} + xyz$

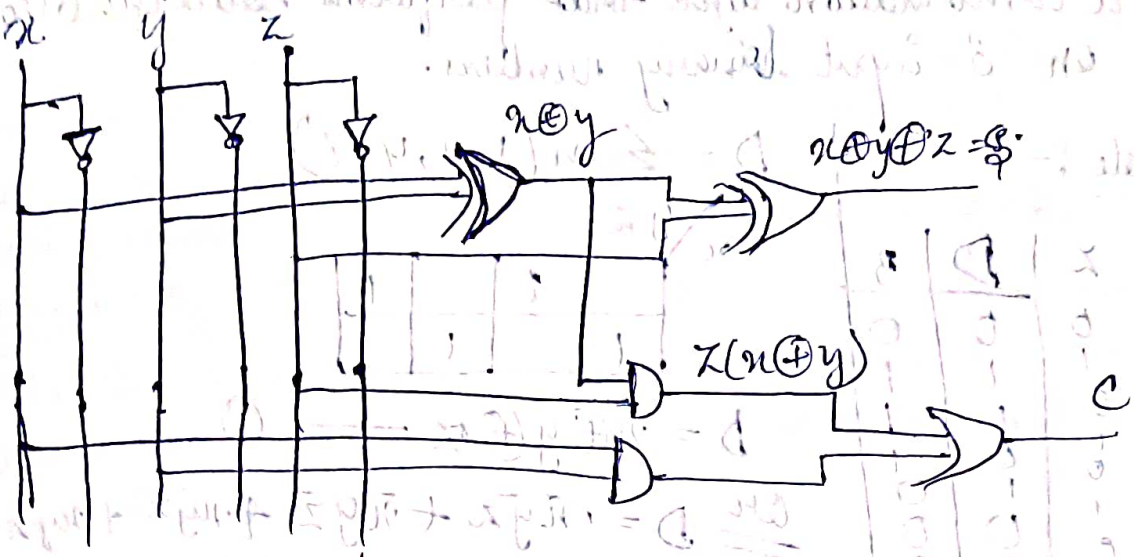
$= z(\bar{x}y + x\bar{y}) + xy(\bar{z} + z)$

$= z(x \oplus y) + xy$

$\boxed{C = z(x \oplus y) + xy}$ —— (3)

$a \backslash bc$	$\bar{b}\bar{c}$	$\bar{b}c$	bc	$b\bar{c}$
$\bar{a}$			1	
a		1	1	1

Circuit diagram :-



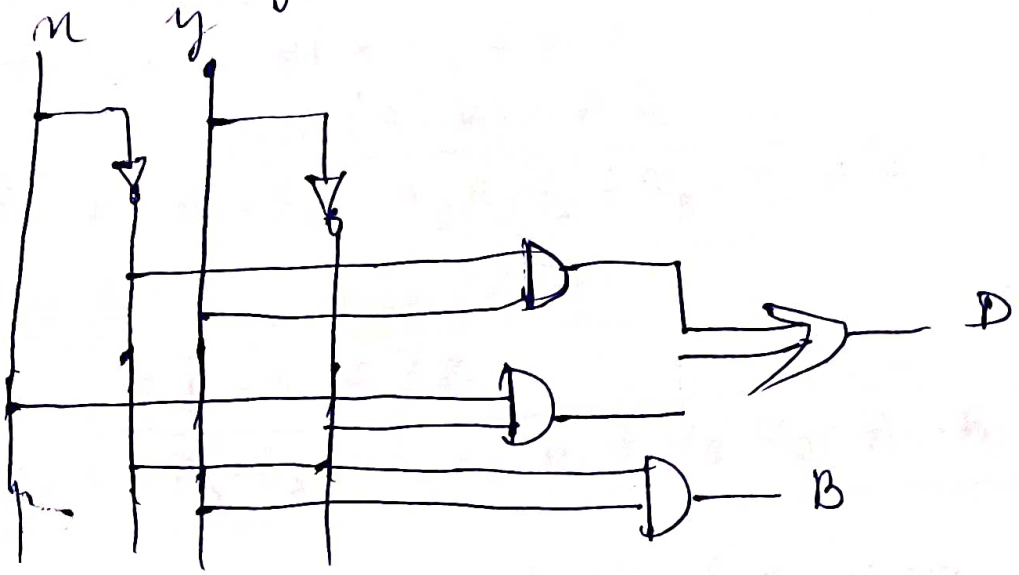
(iii) Half Subtractor :-

It is a combinal circuit that performance and instruction operation on 2 input binary numbers.

Truth Table :-

x	y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Circuit ~~diagram~~ digram :-



Full Subtraction :-

It is a combination logic that performs subtraction operation on 3-input binary numbers.

Truth Table :-

x	y	z	D	B
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

$D = \Sigma m(1,2,4,7)$

$x \backslash yz$			
	1		1
1		1	

$D = x \oplus y \oplus z$ —— (i)

or $D = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$ —

$B = \Sigma m(1,2,3)$

$x \backslash yz$	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{x}$		1	1	1
x			1	

$B = yz + \bar{x}z + \bar{x}y$ —— (3)

or $B = \bar{x}\bar{y}z + \bar{x}yz + \bar{x}\bar{y}z + xyz$ —— (4)

Circuit diagram :-

Magnitude Comparator :-

Let us consider a 4-bit magnitude comparator.
Let 'A' is a 4-bit number $A = A_3\ A_2\ A_1\ A_0$
Let 'B' is a 4-bit number $B = B_3\ B_2\ B_1\ B_0$
The circuit will compare two 4-bit numbers A and B and will provide 3 output that is $A = B$, $A > B$, $A < B$.

Case-1 : Equality Condition ($A = B$)

The two numbers are equal if all pairs of significant digits are equal that is $A_3 = B_3$ and $A_2 = B_2$ and $A_1 = B_1$ and $A_0 = B_0$. If $A = 0$ & $B = 0$ then we get $\bar{A}\bar{B}$.
OR If $A = 1$ & $B = 1$ then we get AB.

$x = \bar{A}\bar{B} + AB = A \odot B$.

$$\boxed{x_i = A_i \odot B_i} \quad \text{where } i = 0, 1, 2, 3. \qquad (1)$$

When the numbers are binary, the digits are either 0 & 1. Hence, the equality of each pair of bits can be logically expressed using an XNOR function as given by eqn (1). Hence, two 4-bits numbers 'A' and 'B' will be equal with the following condition,

$$\boxed{(A = B) = x_3 \cdot x_2 \cdot x_1 \cdot x_0} \qquad (2)$$

Case-2 : Greater than ($A > B$)

Condition : If $A = 1$ & $B = 0$
then $(A > B) = A \cdot \bar{B}$

$$\boxed{A > B = A_3 \cdot \bar{B}_3 + x_3 A_2 \bar{B}_2 + x_3 \cdot x_2 \cdot A_1 \bar{B}_1 + x_3 \cdot x_2 \cdot x_1 \cdot A_0 \bar{B}_0} \qquad (3)$$

Case-3 : Less than ($A < B$)

$A = 0$ & $B = 1 \Rightarrow \bar{A}B$

$$\boxed{A < B = \bar{A}_3 \cdot B_3 + x_3 \bar{A}_2 \cdot B_2 + x_3 \cdot x_2 \cdot \bar{A}_1 \cdot B_1 + x_3 \cdot x_2 \cdot x_1 \bar{A}_0 \cdot B_0} \qquad (4)$$

HW Draw the circuit diagram of the 4-bit magnitude comparator for $A = B$ & $A > B$ & $A < B$. (AND, NOT, OR).

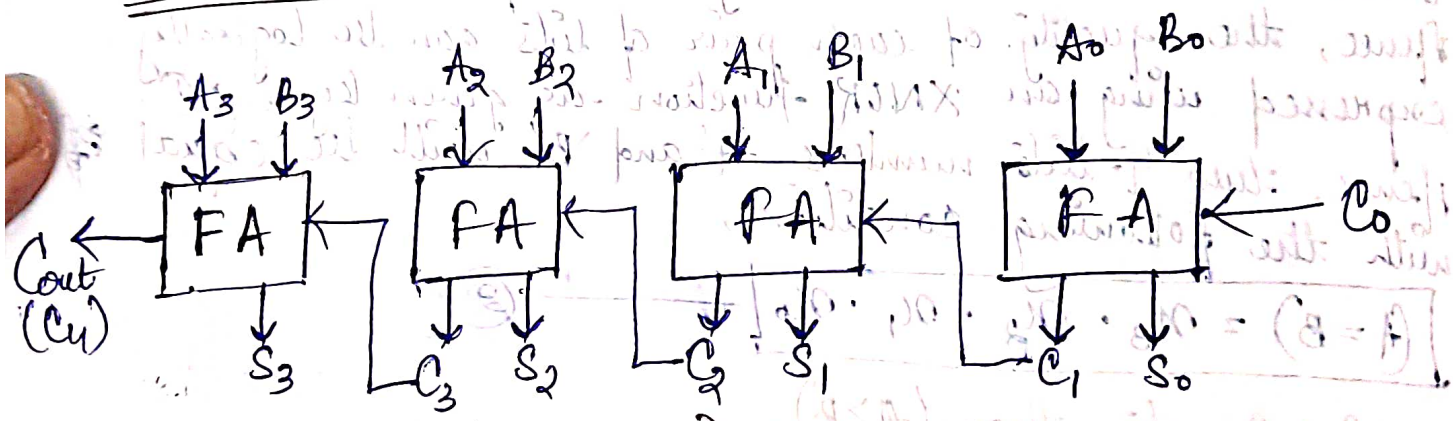
4-Bit Binary Adder (Parallel Adder) :-

For 'n'-bit parallel adder, we need n number of Full Adders

i =	3	2	1	0	
Input Carry: C_{in} =	C_3	C_2	C_1	C_0	← i/p carry
Augend: A_i =	A_3	A_2	A_1	A_0	
Addend: B_i =	B_3	B_2	B_1	B_0	
Sum: S_i =	S_3	S_2	S_1	S_0	
Carry Output: C_{out} =	C_4	C_3	C_2	C_1	

C_4 → o/p carry

Circuit



Decimal Adder (BCD Adder) :-

Deci - 10 (0 to 9)

A = 9
B = 19
19 → 10011

BCD → (0-9)

CF → 0110
↓
6

Truth Table :-

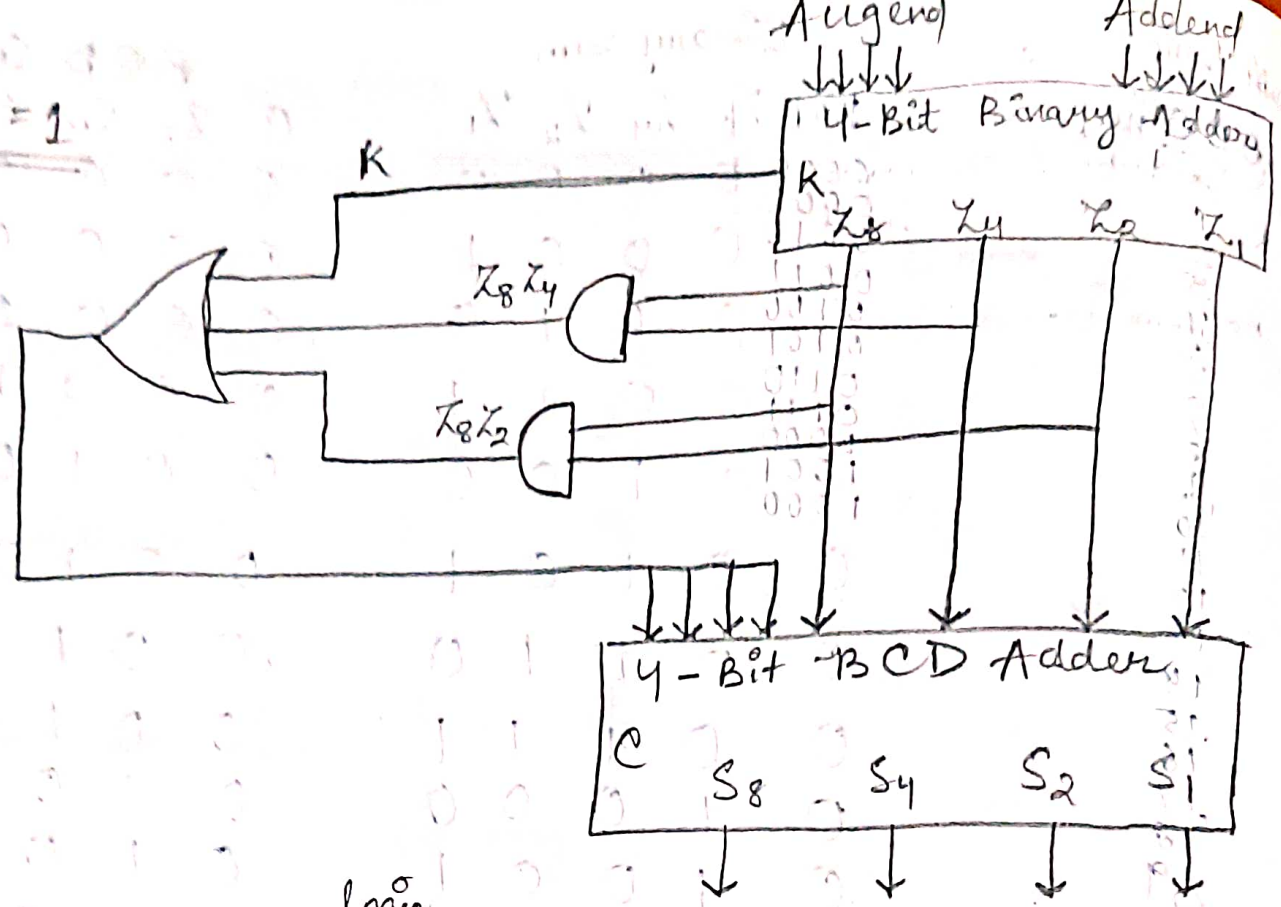
Decimal	Binary Sum K	Z_8	Z_4	Z_2	Z_1	BCD Sum C	S_8	S_4	S_2	S_1
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
2	0	0	0	1	0	0	0	0	1	0
3	0	0	0	1	1	0	0	0	1	1
4	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	1	0	0	1	0	1
6	0	0	1	1	0	0	0	1	1	0
7	0	0	1	1	1	0	0	1	1	1
8	0	1	0	0	0	0	1	0	0	0
9	0	1	0	0	1	0	1	0	0	1
10	0	1	0	1	0	1	0	0	0	0
11	0	1	0	1	1	1	0	0	0	1
12	0	1	1	0	0	1	0	0	1	0
13	0	1	1	0	1	1	0	0	1	1
14	0	1	1	1	0	1	0	1	0	0
15	0	1	1	1	1	1	0	1	0	1
16	1	0	0	0	0	1	0	1	1	0
17	1	0	0	0	1	1	0	1	1	0
18	1	0	0	1	0	1	1	0	0	0
19	1	0	0	1	1	1	1	0	0	1

$C = 0 \rightarrow$ No correction, $C = 1 \rightarrow +6$

The condition for a correction in the BCD sum is expressed through an output carry given as $C = \Sigma m(10, 11, 12, 13, 14, 15, 16, 17, 18, 19)$.

Solving it using a 5-variable K-Map will result

$$C = K + Z_8 Z_4 + Z_8 Z_2 \quad \text{——— (1)}$$



4.13: A combinational logic circuit has 4-inputs A, B, C, D and one output Z, the output is 1 if the input has 3 consecutive zeroes or 3 consecutive ones. For example, if A = 1, B = 0, C = 0 and D = 0 ⇒ Z = 1 but if A = 0, B = 1, C = 0 and D = 0 ⇒ Z = 0. Design the circuit using one 4-input OR gate and four 3-input AND gate.

Ans:-

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

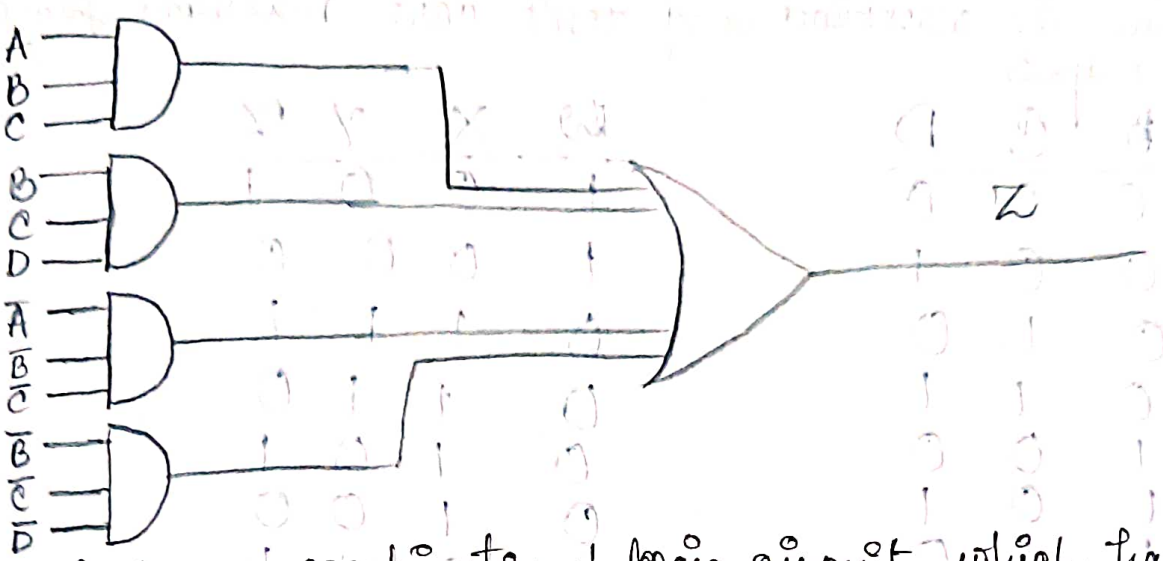
A	B	C	D	Z
1	1	1	0	1
1	1	1	1	1

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

$Z = ABC + BCD + A\bar{B}\bar{C} + \bar{B}\bar{C}\bar{D}$

$Z = \Sigma m(0, 1, 7, 8, 14, 15)$

- Assuming complementary inputs are available.



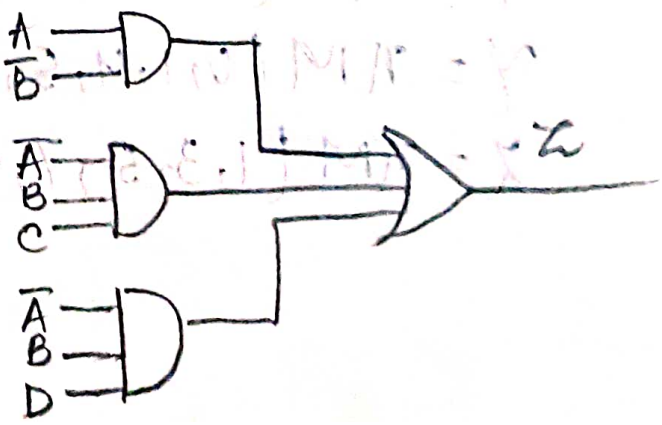
4.14: Design a combinational logic circuit which has 1 output Z and 4-bit input ABCD representing a binary number. Z should be one if the input is atleast 5 but is no greater than 11. Use 1 OR gate with 3 inputs and 3 AND gates (with no more than 3 inputs each).

Ans:-

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

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

$Z = A\bar{B} + \bar{A}BC + \bar{A}BD$



4.17: The 9's complement of decimal digit D is defined (9-d). A logic circuit produces the 9's complement of an input digit where the input and output digits are represented in BCD. The inputs are A, B, C, D and outputs are w, x, y, z.

(a) Determine the minterms and don't care minterms for each of the outputs.

(b) Determine the maxterms and don't care maxterms for each of the outputs.

Ans

A	B	C	D	w	x	y	z
0	0	0	0	1	0	0	1
0	0	0	1	1	0	0	0
0	0	1	0	0	1	1	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	0	0
0	1	1	0	0	0	1	1
0	1	1	1	0	0	1	0
1	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0

(a) $w = \Sigma m(0,1)$, $d = \Sigma m(10,11,12,13,14,15)$

$x = \Sigma m(2,3,4,5)$, $d = \Sigma m(10,11,12,13,14,15)$

$y = \Sigma m(2,3,6,7)$, $d = \Sigma m(10,11,12,13,14,15)$

$z = \Sigma m(0,2,4,6,8)$, $d = \Sigma m(10,11,12,13,14,15)$

(b) $w = \Pi M(2,3,4,5,6,7,8,9)$, $d = (10,11,12,13,14,15)$

$x = \Pi M(0,1,6,7,8,9)$, $d = \Sigma m(10,11,12,13,14,15)$

$y = \Pi M(0,1,4,5,8,9)$, $d = \Sigma m(10,11,12,13,14,15)$

$z = \Pi M(1,3,5,7,9)$, $d = \Sigma m(10,11,12,13,14,15)$

4.35 : A combinational circuit has 4 inputs A, B, C, D and 3 outputs X, Y, Z. X, Y, Z represents a binary number whose value equals the number of 1's at the input.

Ex AB CD = 1011 ⇒ XYZ = 011.

(a) Find the minterm expansion for X, Y, Z.

(b) Find the maxterm expansion for X, Y, Z.

Ans:-

A	B	C	D	X	Y	Z
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
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			

(a) $X = \Sigma m(15)$

$Y = \Sigma m(3, 5, 6, 7, 9, 10, 11, 12, 13, 14)$

$Z = \Sigma m(1, 2, 4, 7, 8, 11, 13, 14)$

(b) $X = \Pi M(0, 1, 2, 3, \ldots, 14)$

$Y = \Pi M(0, 1, 2, 4, 8, 15)$

$Z = \Pi M(0, 3, 5, 6, 9, 10, 12, 15)$

4.36: A combinational circuit has 4 inputs A, B, C, D and 4 outputs W, X, Y, Z. W, X, Y, Z represents excess-3 coded no. whose value equals the no. of 1's at the input.

Ex: ABCD = 1101 ⇒ WXYZ = 0110.

(a) Find the min-term expression for W, X, Y, Z.

(b) Find the maxterm expression for W, X, Y, Z.

Ans:-

A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	0
0	0	1	1	0	1	0	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	0
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	0	1
1	0	1	0	0	1	1	0
1	0	1	1	0	1	0	1
1	1	0	0	0	1	1	0
1	1	0	1	0	1	1	0
1	1	1	0	0	1	1	1
1	1	1	1				

(a) $W = 0$

$X = \Sigma m(1, 2, 3 \ldots, 15)$

$Y = \Sigma m(0, 7, 11, 13, 14, 15)$

$Z = \Sigma m(0, 3, 5, 6, 9, 10, 12, 15)$

(b) $W = \Pi M(0, 1, 2 \ldots 15)$

$X = \Pi M(0)$

$Y = \Pi M(1, 2, 3, 4, 5, 6, 8, 10, 12)$

$Z = \Pi M(1, 2, 4, 7, 8, 11, 13, 14)$

4.37 : A combinational circuit has 4-inputs A, B, C and D which represents a binary coded decimal digit. The circuit has 2 groups of 4 outputs that is S, T, U, V and W, X, Y, Z. Each group represents a BCD digit. The output digit represents a decimal number which is five times the input number.

Ex : ABCD = 0111 then, STUV is 0011 and WXYZ is 0101

Assume that invalid BCD digits do not occur at inputs.

(a) Construct the truth table.

(b) Write down the minimum expressions for the outputs by observing the truth table.

Ans:-

A	B	C	D	S	T	U	V	W	X	Y	Z
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	1	0	1
0	0	1	0	0	0	0	1	0	0	0	0
0	0	1	1	0	0	1	1	0	1	0	1
0	1	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	1	0	0	1	0	1
0	1	1	0	0	0	1	1	0	0	0	0
0	1	1	1	0	0	1	1	0	1	0	1
1	0	0	0	0	1	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	1	0	1

S = 0 W = 0
T = A X = D
U = B Y = 0
V = C Z = D

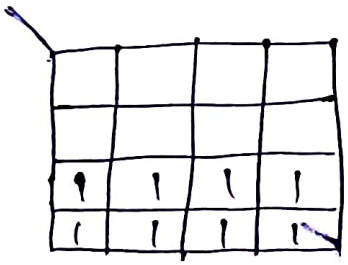
Imp Q. Design a combinational circuit to convert a 4-bit binary input to its equivalent gray code.

Ans:- Binary Inputs → B_3 B_2 B_1 B_0

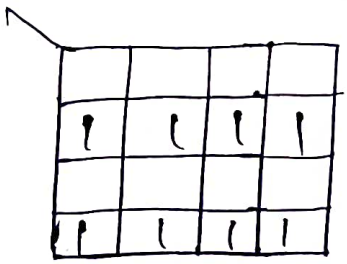
Gray Code → G_3 G_2 G_1 G_0.

B_3	B_2	B_1	B_0
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

G_3	G_2	G_1	G_0	
0	0	0	0	(0)
0	0	0	1	(1)
0	0	1	1	(3)
0	0	1	0	(2)
0	1	1	0	(6)
0	1	1	1	(7)
0	1	0	1	(5)
0	1	0	0	(4)
1	1	0	0	(12)
1	1	0	1	(13)
1	1	1	1	(15)
1	1	1	0	(14)
1	0	1	0	(10)
1	0	1	1	(11)
1	0	0	1	(9)
1	0	0	0	(8)

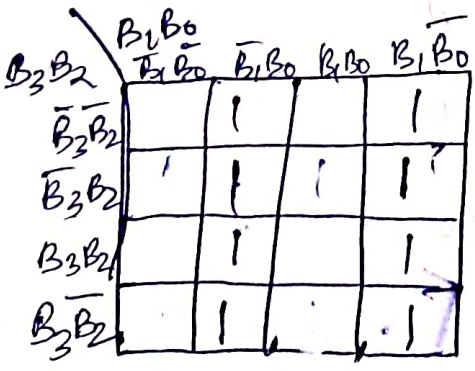


$G_3 = B_3$

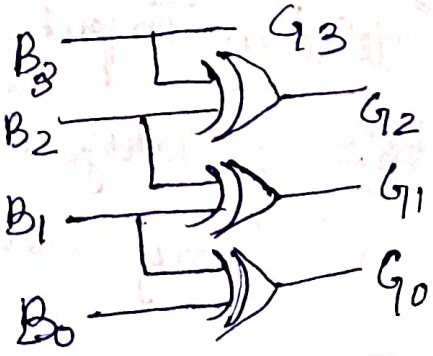


$G_2 = \bar{B}_3 B_2 + B_3 \bar{B}_2$
$= B_3 \oplus B_2$

$G_1 = B_2 \bar{B}_1 + \bar{B}_2 B_1$
$= B_2 \oplus B_1$



$G_0 = \bar{B}_1 B_0 + B_1 \bar{B}_0$
$= B_1 \oplus B_0$



4 Bit Gray to Binary Conversion :-

G_3	G_2	G_1	G_0
0	0	0	0
0	0	0	1
0	0	1	1
0	0	1	0
0	1	1	0
0	1	1	1
0	1	0	1
0	1	0	0
1	1	0	0
1	1	0	1
1	1	1	1
1	1	1	0
1	0	1	0
1	0	1	1
1	0	0	1
1	0	0	0

B_3	B_2	B_1	B_0
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

G_3G_2 \ G_1G_0	$\bar{G}_1\bar{G}_0$	$\bar{G}_1G_0$	G_1G_0	$G_1\bar{G}_0$
$\bar{G}_3\bar{G}_2$				
$\bar{G}_3G_2$				
G_3G_2	1	1	1	1
$G_3\bar{G}_2$	1	1	1	1

$B_3 = G_3$

$G_3\bar{G}_2$ \ $\bar{G}_1\bar{G}_0$	$\bar{G}_1G_0$	G_1G_0	$G_1\bar{G}_0$
$\bar{G}_3\bar{G}_2$			
$\bar{G}_3G_2$	1	1	1	1
G_3G_2			
$G_3\bar{G}_2$	1	1	1	1

$B_2 = \bar{G}_3 G_2 + G_3 \bar{G}_2$

$= G_3 \oplus G_2.$

$G_3G_2 \backslash G_1G_0$	$\bar{G}_1\bar{G}_0$	$\bar{G}_1G_0$	G_1G_0	$G_1\bar{G}_0$
$\bar{G}_3\bar{G}_2$			1	1
$\bar{G}_3G_2$	1	1		
G_3G_2			1	1
$G_3\bar{G}_2$	1	1		

$$B_1 = \bar{G}_3\bar{G}_2\cdot\bar{G}_1 + \bar{G}_3G_2\cdot\bar{G}_1 + G_3G_2\cdot G_1 + G_3\bar{G}_2\cdot\bar{G}_1$$

$$= \bar{G}_1(\bar{G}_3G_2 + G_3\bar{G}_2) + G_1(\bar{G}_3\bar{G}_2 + G_3G_2)$$

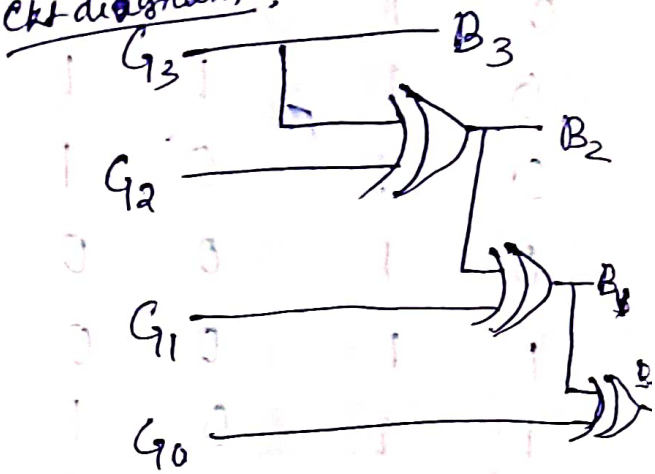
$$= \bar{G}_1(G_3 \oplus G_2) + G_1(G_3 \odot G_2)$$

$$= G_3 \oplus G_2 \oplus G_1$$

$G_3G_2 \backslash G_1G_0$				
$\bar{G}_3\bar{G}_2$		1		1
$\bar{G}_3G_2$	1		1	
G_3G_2		1		1
$G_3\bar{G}_2$	1		1	

$$B_0 = G_3 \oplus G_2 \oplus G_1 \oplus G_0$$

CKT diagram :-



Q1. Design a combinational circuit to convert the BCD to XS-3 code for decimal digits.

Q2. Design the SOP circuit to detect decimal numbers 5 to 12 in a 4-bit gray code input.

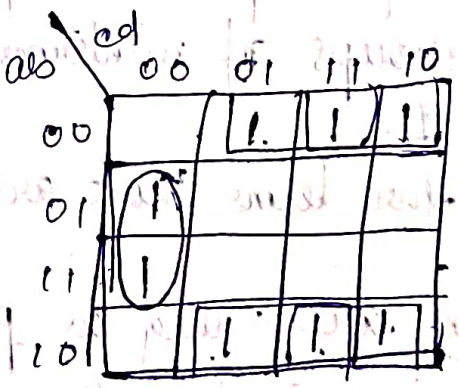
Q3. Design a circuit to detect decimal number 0, 1, 4, 6, 7, 8 in a 4-bit XS-3 code input. Input (3 to 12)

Q4. Design a combinational circuit to produce the 2's complement of a 4-bit binary number.

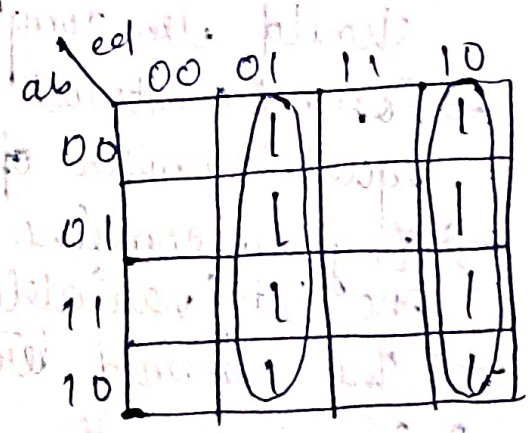
Q1. to Q3. H/W

4-bit Binary No.				9's Complement			
a	b	c	d	w	x	y	z
0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1
0	0	1	0	1	1	1	0
0	0	1	1	1	1	0	1
0	1	0	0	1	1	0	0
0	1	0	1	1	0	1	1
0	1	1	0	1	0	1	0
0	1	1	1	1	0	0	1
1	0	0	0	1	0	0	0
1	0	0	1	0	1	1	1
1	0	1	0	0	1	1	0
1	0	1	1	0	1	0	1
1	1	0	0	0	1	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	0	1	0
1	1	1	1	0	0	0	1

$w = \bar{a}b + \bar{a}d + \bar{a}c + a\bar{b}\bar{c}\bar{d}$



$x = b\bar{c}\bar{d} + \bar{b}d + \bar{b}c$



$y = \bar{c}d + c\bar{d}$

$= c \oplus d$

ab \ cd	00	01	11	10
00		1	1	
01		1	1	
11		1	1	
10		1	1	

$Z = d$

Ch-

Quine McC

→ Quine McCluskey Method reduce
function to obtain a minimum
The procedure consists of two
(i) Eliminate as many