23/3/21

Digital system – Digital system is a system that manipulates discrete elements of informatⁿ represented internally in binary form.

→ Discrete elements of informatⁿ are represented by physical quantities called signals.

→ Signals in electronic digital system uses 2 discrete values called as binary values.

→ A binary digit called bit has 2 values i.e., 0 & 1

→ Discrete elements of informatⁿ are represented with group of bits called binary codes.

→ Digital systems have prominent role in everyday life. They are used in communicatⁿ, bussiness transactⁿs, traffic control, medical treatments, weather monitoring, internet, industrial & scientific enterprises, digital telephones, digital televisⁿ & digital cameras etc.

→ The reason that commercial products are made with digital circuits are i) Most digital devices are programmable

ii) Cost reductⁿ

iii) Speed

iv) Reliability.

→ A digital system is an interconnectⁿ of digital modules. To understand the operat^n of each digital module, it is necessary to have basic information of digital circuits and their logical functions.

1 Number Systems.

1. Binary — Base 2 — (0 or 1)
2. Octal — Base 8 — (0 to 7)
3. Decimal — Base 10 — (0 to 9)
4. Hexa-decimal — Base 16. — (0 to 15)

$$\left[\begin{array}{ccccccc} 0 \text{ to } 9, & 10 & 11 & 12 & 13 & 14 & 15 \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ & A & B & C & D & E & F \end{array}\right].$$

1) Binary no. system – It is a base 2 system with two binary bits 0 & 1.

→ In binary system each binary digit, commonly known as bit has its own value & weight

→ In binary system weight is expressed as power of 2 (2^n)

ex:- 1) $\left(\begin{array}{cccc} 2^3 & 2^2 & 2^1 & 2^0 \\ 1 & 0 & 1 & 1 \end{array}\right)_2 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$

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

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

$= 8+0+2+1+0.5=(11.5)_{16}$

Representatat^n of no's

2^n

for $n=0 \Rightarrow 2^0=1 - (0 \text{ or } 1)$

$n=1 \Rightarrow 2^1=2 - (0,1,2)$

0 to 15

Decimal no	$\overset{2^3}{8}\ \overset{2^2}{4}\ \overset{2^1}{2}\ \overset{2^0}{1}$
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
10	1 0 1 0
11	1 0 1 1
12	1 1 0 0
13	1 1 0 1
14	1 1 1 0
15	1 1 1 1

Number Base Conversny

The human beings use decimal no. system which computer uses binary no. system. It is necessary to convert decimal no. into its equivalent binary no. while feeding no. into the computer and to convert binary no. into its decimal equivalent while displaying result of operatn ~~to the~~ human beings.

→ However dealing with large quantity of binary no.s of many bits is inconvincent for human beings. Therefore, octal & hexa-decimal no.'s are used as a shorthand means of expressing large binary no.s. "Digital circuits strictly work in binary".

Conversn of any base or Radix no. system ~~to decimal~~.

* Binary to decimal conversn

H/W

1) $(1101)_2 — (\ )_{10}$

$1\times2^0+0\times2^1+1\times2^2+1\times2^3$

$=1+0+4+8=(13)_{10}$

2) $(101101)_2 — (\ )_{10}$

$1\times2^0+0\times2^1+1\times2^2+1\times2^3+0\times2^4+1\times2^5$

$=1+0+4+8+0+32=(45)_{10}$

3) $(11011.01)_2 — (\ )_{10}$

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

$=1+2+0+8+16+0+1/4=27+0.25=(27.25)_{10}$

4) $(1110.0101)_2 — (\ )_{10}$

$8+0+4+2+0+0+0.25+0.0625$

$=(14.3125)_{10}$

5) $(1010.101)_2 — (10.625)_{10}$

$8+0+2+0+0.5+0+0.125$

$=(10.625)_{10}$

* Octal to decimal conversn $[(\ )_8 \rightarrow (\ )_{10}]$

1) $(24)_8 \longrightarrow (20)_{10}$

$4\times8^0+2\times8^1=4+16=20$

2) $(123)_8 \longrightarrow (83)_{10}$

$64+2\times8+3\times1=64+16+3=83$

3) $(126.5)_8 \rightarrow (\ )_{10}$

$64+2\times8+6\times1+5/8=86+0.625$

$=(86.625)_{10}$

4) $(26.21)_8 \longrightarrow (\ )_{10}$

$16+6+1/4+1/8=22+0.25+0.125$

$=(22.375)_{10}$

H/W

5) $(735)_8 \longrightarrow (477)_{10}$

$7\times64+3\times8+5=477$

6) $(246)_8 \longrightarrow (166)_{10}$

$2\times64+4\times8+6=166$.

* Hexa decimal to decimal conversn $[(\ )_{16} \rightarrow (\ )_{10}]$

1) $(129)_{16} \longrightarrow (297)_{10}$

$256+32+9=297$

2) $(AB.24)_{16} \longrightarrow (171.375)_{10}$

$11+10\times16+2/16+4/16=160+0.125+0.25+11$

$=171.375$

3) $(16.5)_{16} \longrightarrow (22.3125)_{10}$

$16+6+5/16=16+6+0.3125=22.3125$

H/W

4) $(BABA)_{16} \rightarrow (47,802)_{10}$

$A\times16^0+B\times16^1+A\times16^2+B\times16^3=(10+11\times16+10\times256+11\times256\times16)$

5) $(ABCD)_{16} \longrightarrow (44,176)_{10}$

$D\times16^0+C\times16+B\times16^2+A\times16^3$

$=(\cancel{208}+12\times16+11\times16^2+10\times256\times16)$

6) $(FAFA.B)_{16} \longrightarrow (64250.6875)_{10}$

$15\times4096+10\times256+15\times16+10\times1+11\times0.0625$

$=64250.6875$

* Base 3 / Base 4 to decimal conversion

⇒ Base 3 to decimal

1) $(21)_3 \to (7)_{10}$ ($3^1\ 3^0$)

$1\times1+2\times3=1+6=7$

2) $(112.22)_3 \to (\ )_{10}$ ($3^2\ 3^1\ 3^0\ 3^{-1}\ 3^{-2}$)

$2\times1+1\times3+1\times9+2/3+2/3^2$

$=2+3+9+0.66666667+$
0.22222222

$=(14.888889)_{10}$

⇒ Base 4 to decimal

1) $(324)_4 \to (60)_{10}$

$4\times1+2\times4+3\times16=4+8+48=60$

2) $(3122.33)_4 \to (218.9375)_{10}$

$2+8+16+3\times64+3/4+3/16$

$=(218.9375)_{10}$

3) $(22.625)_4 \to (11.703125)_{10}$

$2+8+6/4+2/16+5/64=(11.703125)_{10}$

⇒ $(100)_x \to (16)_{10}$ then $x=?$

$0(x^0)+0(x^1)+1(x^2)=16$

$x^2=16 \Rightarrow x=\sqrt{16} \Rightarrow x=4$

30/3/21

Conversion of decimal to any Base no. :-

* conversion of decimal to base 2 (binary) $[(\ )_{10} \to (\ )_2]$

1) $(24)_{10} \to (11000)_2$

2	24	= 0
2	12	= 0
2	6	= 0
2	3	= 1
	1	

$(11000)_2$

no. order

2) $(1217)_{10} \to (\ )_2$

2	1217	= 1
2	608	= 0
2	304	= 0
2	152	= 0
2	76	= 0
2	38	= 0
2	19	= 1
2	9	= 1
2	4	= 0
2	2	= 0
	1	

$(10011000001)_2$

5) $(24.625)_{10} \longrightarrow (11000.101)_2$

multiply the base atleast 3 to 4 times

$2 \underline{|24} = 0$
$2 \underline{|12} = 0$
$2 \underline{|6} = 0$
$2 \underline{|3} = 1$
1

$0.625 \times 2 = 1.25$

$1.25 \times 2 = 2.5$

$0.5 \times 2 = 1$

(11000)

$(24.625)_{10} \longrightarrow (11000.101)_2$

4) $(41.6875)_{10} \longrightarrow (\ )_2$

$2 \underline{|41} = 1$
$2 \underline{|20} = 0$
$2 \underline{|10} = 0$
$2 \underline{|5} = 1$
$2 \underline{|2} = 0$
1

$0.6875 \times 2 = 1.375$

$1.375 \times 2 = 2.75 = 0.75$

$0.75 \times 2 = 1.5 = 0.5$

$0.5 \times 2 = 1.$

(101001)

$(101001.1011)_2$

* Conversion of decimal no to octal no. $[(\ )_{10} \longrightarrow (\ )_8]$.

1) $(29)_{10} \longrightarrow (35)_8$

$8 \underline{|29} = 5$ $(35)_8$
3

2) $(153)_{10} \longrightarrow (231)_8$

$8 \underline{|153} = 1$
$8 \underline{|25} = 1$
3

$(\cancel{314})_8$ 231

3) $(0.513)_{10} \longrightarrow (0.4065)_8$

$0.513 \times 8 = 4 \times 0.4$

$0.104 \times 8 = 0.832$

$0.832 \times 8 = 6.656$
$= 0.656$

$0.656 \times 8 = 5.248$

$(0.513)_{10} \longrightarrow (0.4065)_8$

4) $(204.25)_{10} \longrightarrow (314.2)_8$

$8 \underline{|204} = 4$
$8 \underline{|25} = 1$
3

(314)

$0.25 \times 8 = 2.00$

$(204.25)_{10} \longrightarrow (314.2)_8$

* Conversⁿ of decimal no. to hexadecimal no.

$[()_{10} \rightarrow ()_{16}]$

1) $(4769)_{10} \rightarrow ()_{16}$

$$\begin{array}{r|l l} 16 & 4769 & =1 \\ 16 & 298 & =10=A \\ 16 & 18 & =2 \\ & 1 & \end{array}$$

$(12A1)_{16}$

$(4769)_{10} \rightarrow (12A1)_{16}$

2) $(422.675)_{10} \rightarrow (1A6.ACC)_{16}$

$$\begin{array}{r|l l} 16 & 422 & =6 \\ 16 & 26 & =10=A \\ & 1 & \end{array}$$

$(1A6)$

$0.675 \times 16 = 10.8 = A$

$0.8 \times 16 = 12.8 = C$

$0.8 \times 16 = 12.8 = C$

$(1A6.ACC)_{16}$

* Conversⁿ of decimal to any base (3/4/6)

$(()_{10} \rightarrow ()_3 / ()_4 / ()_6)$

1) $(24)_{10} \rightarrow (120)_4$

$$\begin{array}{r|l l} 4 & 24 & =0 \\ 4 & 6 & =2 \\ & 1 & \end{array}$$

$(120)_4$

3) $(41)_{10} \rightarrow (1112)_3$

$$\begin{array}{r|l l} 3 & 41 & =2 \\ 3 & 13 & =1 \\ 3 & 4 & =1 \\ & 1 & \end{array}$$

$(1112)_3$

2) $(164.25)_{10} - (444.13)_6$

$$\begin{array}{r|l l} 6 & 164 & =4 \\ 6 & 28 & =4 \\ & 4 & \end{array}$$

(444)

$0.25 \times 6 = 1.5$

$0.5 \times 6 = 3$

$(444.13)_6$

21/3/21

Octal and hexadecimal no's

* Binary to octal conversn $[(\ )_2 \rightarrow (\ )_8]$.

(octal no can be represented in 3 bits).

1) $(11011)_2 \longrightarrow (33)_8$

$2^2\ 2^1\ 2^0$	$2^2\ 2^1\ 2^0$
011	011
2+1 = 3	3

2) $(101011.01101)_2 \longrightarrow (53.32)_8$

$2^2\ 2^1\ 2^0$	$2^2\ 2^1\ 2^0$	$2^2\ 2^1\ 2^0$	$2^2\ 2^1\ 2^0$
101	011	011	010
5	3	3	2

3) $(1010.1010)_2 \longrightarrow (12.5)_8$

001	010	101	000
1	2	5	0

* Octal to binary conversn $[(\ )_8 \rightarrow (\ )_2]$

1) $(124)_8 \longrightarrow (\ )_2$

1	2	4
001	010	100

(By 421 table)

$(124)_8 \longrightarrow (001010100)_2$

2) $(172.613)_8 \longrightarrow (\ )_2$

1	7	2	6	1	3
001	111	010	110	001	011

$(172.613)_8 \longrightarrow (001111010.110001011)_2$

* Binary to Hexadecimal conversⁿ

1) ~~$(124)_8 \longrightarrow (\ )_2$~~ $(1101101100)_2 \longrightarrow (36C)_{16}$

~~By 8421 table~~

1	2	4
001	010	100

0011	0110	1100
3	6	12(C)

$(36C)_{16}$

2) $(101011.01101)_2 \longrightarrow (\ )_{16}$

0010	1011	.0110	1000
2	11(B)	6	8

$(2B.68)_{16}$

* Hexadecimal to binary conversⁿ

1) $(BABA)_{16} \rightarrow (\ )_2$

	8421
11	1011
10	1010

BABA → 11 10 11 10

$(BABA)_{16} \longrightarrow (1011101010111010)_2$

2) $(16.5)_{16} \longrightarrow (\ )_2$

	8421
1	0001
6	0110
5	0101

$(00010110.0101)_2$

3) $(FAFA.B)_{16} \rightarrow (\ )_2$

FAFA.B → 15 10 15 10 . 11

	8421
15	1111
10	1010
11	1011

$(FAFA.B)_{16} \rightarrow (1111101011111010.1011)_{16}$

* Octal to Hexadecimal conversⁿ

1) $(147)_8 \longrightarrow (\ )_{16}$

Octal ⟶ Binary ⟶ Hexadecimal

1	4	7
001	100	111

$(001100111)_2 \longrightarrow (067)_{16}$

0000	0110	0111
0	6	7

$(147)_8 \rightarrow (001100111)_2$ $(147)_8 \rightarrow (067)_{16}$

2) $(26.24)_8 \rightarrow (\ )_{16}$

2	6	2	4
010	110	010	100

$\Rightarrow (010110 . 010100)_2$

0001	0110	0101	0000
1	6	5	0

$(16.50)_{16}$

$(26.24)_8 \rightarrow (16.50)_{16}$

* Hexadecimal to octal conversion

1) $(16.5)_{16} \rightarrow (26.24)_8$

Hexadecimal → binary → octal

1	6	5
0001	0110	0101

$(00010110 . 0101)_2$

$(00010110 . 0101)_2$

000	010	110	010	100
0	2	6	2	4

$(26.24)_8$

2) $(68BE)_{16} \rightarrow (64276)_8$

6	8	B (11)	E (14)
0110	1000	1011	1110

$\Rightarrow (0110100010111110)_2$

000	110	100	010	111	110
0	6	4	2	7	6

$\Rightarrow (064276)_8$

3) $(27.825)_{16} \rightarrow (47.4045)_8$

2	7	. 8	2	5
0010	0111	. 1000	0010	0101

$\Rightarrow (00100111 . 100000100101)_2$

000	100	111	. 100	000	100	101
0	4	7	. 4	0	4	5

$\Rightarrow (47.4045)_8$

Binary addition

→ [n inputs are given then 2^n outputs are out]

i/p		o/p	
A +	B	sum	carry
1	0	1	0
0	1	1	0
1	1	0	1
0	0	0	0

1)
```
  1101 = 13
+ 0101 =  5
 10010 = 18
```

2)
```
  1001 = 9
+ 0110 = 6
  1111 = 15
```

3)
```
   100101
 + 110111
  1011100
```

4)
```
   110110
 + 011001
  1001111
```

5)
```
   110111
 + 111100
  1110011
```

Binary subtraction

i/p		o/p	
A −	B	diff	borrow
0	0	0	0
1	1	0	0
0	1	1	1
1	0	1	0

10 = 2, 2−1 = 1

1) 0 1 1 0 0 = 12
$-$ 0 0 1 1 0 = 6
0 0 1 1 0 = 6

2) 0 1 1 1 0 1 0 1 0
$-$ 0 1 1 0 0 1
0 1 1 0 1 1

H/W

3) 1 1 0 0 1 = 25
$-$ 0 0 1 1 0 = 6
1 0 0 1 1 = 19

4) 1 1 0 0 1 = 25
$-$ 0 1 0 1 0 = 10
0 1 1 1 1 = 15

5) 0110 = 6
$-$ 0010 = 2
0100 = 4

Octal, hexadecimal addition

Octal addition

1)
2 4
$+$ 7 3
11 7

(7 represented as 7
4 respresented as 11)

2)
7 6 3
$+$ 4 7 6
14 6 1

(9 − 11
14 − 16
12 − 14)

3)
1 4 7 7
$+$ 6 3 3 4
1 0 0 3 3

(11 − 13
8 − 10
8 − 10)

Hexadecimal addition

1)
1 A (10)
$+$ 2 5
3 F (15)

2)
4 ~~7~~ 9
$+$ 8 1 7
C 9 0

(16 − 10
9 − 9
12 − C)

H/W

3)
A B C D
$+$ B A B A
1 6 6 8 7

(22 − 16
29 − 18
22 − 16)

4)
B 4 C
$+$ 7 9 A
12 E 6

(22 − 16
14 − E
18 − 12)

Decimal	(0-8) Octal	Hexadecimal
0	0	0
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	10	8
9	11	9
10	12	A
11	13	B
12	14	C
13	15	D
14	16	E
15	17	F
16	20	10
17	21	11
18	22	12
19	23	13
20	24	14
21	25	15
22	26	16
23	27	17
24	30	18

Decimal	Octal	Hexadecimal
25	31	19
26	32	1A
27	33	1B
28	34	1C
29	35	1D
30	36	1E
31	37	1F
32	40	20
33	41	21
34	42	22
35	43	23
36	44	24
37	45	25
38	46	26
39	47	27
40	50	28
41	51	29
42	52	2A
43	53	2B
44	54	2C
45	55	2D
46	56	2E
47	57	2F
48	60	30

6/9/21.

Compliments

→ Compliments are used in digital computers to simplify the operation of substraction for logical manupulation

There are 2 types of Compliments

1) The Radix compliment (or) r's compliment.
2) The Diminished compliment (or) (r-1) compliment.

For. binary system

→ 2's compliment (r's compliment)
→ 1's Compliment ((r-1) compliment)

For decimal system

→ 10's compliment (r's compliment)
→ 9's compliment ((r-1) compliment)

* 1's Compliment

1) Perform 1's compliment of 1101 ?

In 1's compliment '1' is changed to '0' and vice versa

$1101 \xrightarrow{1's} 0010$

2) $11011001 \xrightarrow{1's} 00100110$

3) $1111110101 \xrightarrow{1's} 0000001010$.

* 2's compliment

1) Perform 2's compliment of 1101?

In 2's compliment, first the given number is converted into 1's compliment than "1" is added to it.

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

$$1101 \xrightarrow{1's} 0010$$
$$2's \rightarrow +1$$
$$0011$$

2) $$1010 \xrightarrow{1's} 0101$$
$$2's \rightarrow +1$$
$$0110$$

3) $$(11011010)_2 \xrightarrow{1's} 00100101$$
$$2's \rightarrow +1$$
$$00100110$$

* 9's compliment

1) Perform 9's compliment of 24?

In 9's compliment, the every digit should be substracted from 9

$$24 \xrightarrow{9's} 99$$
$$-24$$
$$75$$

$$(24)_{10} \xrightarrow{9's} (75)_{10}$$

2) $$789 \xrightarrow{9's} 999$$
$$-789$$
$$210$$

4) $$547600 \xrightarrow{9's} 999999$$
$$-547600$$
$$452399$$

3) $$4716 \xrightarrow{9's} 9999$$
$$-4716$$
$$5283$$

* 10's compliment

1) $24 \xrightarrow{9's}$

$$\begin{array}{r} 99 \\ -24 \\ \hline 75 \\ 2's \rightarrow \; +1 \\ \hline 76 \\ \hline \end{array}$$

$(24)_{10} \xrightarrow{10's} (76)_{10}$

2) $102398 \xrightarrow{9's}$

$$\begin{array}{r} 999999 \\ -102398 \\ \hline 897601 \\ 10's \rightarrow \; +1 \\ \hline 897602 \\ \hline \end{array}$$

3) $(246700)_{10} \xrightarrow{9's}$

$$\begin{array}{r} 999999 \\ -246700 \\ \hline 753299 \\ 10's \rightarrow \; +1 \\ \hline 753300 \\ \hline \end{array}$$

→ In 10's compliment, first 9's compliment is performed and 1 is added $\boxed{10's = 9's + 1}$

Subtraction with Compliments

* 1's Compliment subtraction

Case-1 :- subtractⁿ of smaller no. from larger no.

1) Perform 1101 − 0110

Step-1: Take 1's compliment of subtractand

$0110 \xrightarrow{1's} 1001$

Step-2: Add result to minuend

1001
+01101
10110

Step-3: Add carry to the final result by removing carry

0110
+1
0111

Verificatⁿ:- 1101 ⟶ 13
-0110 ⟶ 6
0111 ⟶ 7

2) 1010100 − 1000011

1000011 $\xrightarrow{1's}$ 0111100

0111100
+1010100
carry ① 0010000

0010000
+1
0010001

Verificatⁿ:-
1010100 = 84
-1000011 = 67
0010001 = 17

Case-2: Subtraction of larger no from smaller no.

1) Perform 0101−1010

Step-1: find 1's compliment of substrand.

1010 $\xrightarrow{1's}$ 0101

Step-2: Add the 1's compliment result to minuend

0101
+0101
1010

Step-3: The result is negative so take the 1's compliment and assign -ve symbol to result

$1010 \xrightarrow{1's} 0101$ $\quad 0101 - 1010 = -0101$

Verificatⁿ:-

$$\begin{array}{rl} -1010 & = -10 \\ 0101 & = 5 \\ \hline -0101 & = -5 \end{array}$$

2) 1010100 − 100011

$1000011 \xrightarrow{1's} 0111100$

$$\begin{array}{r} 0111100 \\ +\ 1010100 \\ \hline 1\ 0010000 \end{array}$$

(carry)

$$\begin{array}{r} 0010000 \\ +\ 1 \\ \hline 0010001 \end{array}$$

Verificatⁿ:-

$$\begin{array}{rl} 1010100 & = 84 \\ -1000011 & = 87 \\ \hline 0010001 & = 17 \end{array}$$

2) Perform 1001 − 101000

$101000 \xrightarrow{1's} 010111$

$$\begin{array}{r} 010111 \\ +\ 1001 \\ \hline 100000 \end{array}$$

$100000 \xrightarrow{1's} 011111$

(− 011111)

Verificatⁿ:-

(1001) + (−101000)

$$\begin{array}{r} -101000 \\ +\ 1001 \\ \hline 1 \end{array}$$

7/9/21

* 2's compliment subtraction

Case-1 :- subtractⁿ of smaller no. from larger no.

1) subtract $(01010)_2$ from $(11001)_2$
(substrand) (minuemand)

$(11001)_2 - (01010)_2$

Step-1: $(01010) \xrightarrow{1's} 10101$
$\xrightarrow{2's}$ +1
10110

Step-2) Add 2's compliment result to minuemand

10110
+ 11001
Carry (1) 01111

step-3: Discard the carry term

01111

verificatⁿ :-

11001 → 25
− 01010 → 10
01111 → 15

Case-2 :- subtractⁿ of larger no. from smaller no.

1) Perform 100 − 10101

1) Take 2's compliment of substrand $10101 \xrightarrow{2's} 01010$
+1
01011

2) Add 2's compliment result to minuemand
01011
+ 100
01111

3) Take 2's compliment of above result and −ve sign.

$01111 \xrightarrow{2's} 10000$
+1
10001

⇒ (−10001)

Verificatⁿ :-
100 = 4
−10101 = −21
−10001 = −17

H.W

$(10011)_2 - (10001)_2$

2)

$10001 \xrightarrow{1's} 01110$

$2's \rightarrow +1$

01111

01111

$+10011$

carry → $(1)00010 \Rightarrow (00010)$

$(10011)_2 - (10001)_2 = (00010)_2$

verificat^n :-

$10011 = 19$

$-10001 = 17$

$00010 = 02$

2) Perform $(0101100)_2 - (1000011)_2$

$1000011 \xrightarrow{2's} 0111100$

$+1$

0111101

0111101

$+0101100$

1101001

$1101001 \xrightarrow{2's} 0010110$

$+1$

0010111

$(0101100)_2 - (1000011)_2 = (-0010111)_2$

Verificat^n :-

$0101100 = 47$

$-1000011 = -67$

$-0010111 = -23$

* 9's compliment subtraction

Case-1 :- subtract^n of smaller no. from larger no.

1) Perform 52532 − 3250

1) Take 9's compliment of substrand

```
3250 --9's--> 99999
            − 03250
            -------
              96749
```

2) Add 9's compliment result to mincumand.

```
        96749
      + 52532
      -------
carry → ① 49281
```

3) Add the carry to final result

```
  49281
 +    1
 ------
  49282
```

verification :-

```
   52532
 −  3250
 -------
   49282
```

Case-2 :- subtract^n of smaller no. from larger no.

1) Perform ~~6428 − 3409~~ 3250 − 52532

1) Take 9's compliment of subtrahend.

```
~~3409 --10's--> 9999
               − 3409
               ------
                 6590
                   +1
               ------
                 6591~~
```

```
52532 ----> 99999
          − 52532
          -------
            47467
```

2) Add 9's compliment result to minumand.

```
  47467
+ 03250
  50717
```

3) Take 9's compliment of final result and add -ve sign.

```
  99999
- 50717
- 49282
```

3250 - 52532 = -49282

* 10's compliment subtractⁿ

<u>case-1</u> :- subⁿ of smaller no. from larger no.

I) Perform 6428 - 3409

1) Take 10's compliment of subtractand.

$3409 \xrightarrow{10's}$

```
  9999
- 3409
  6590
    +1
  6591
```

2) Add result to the minumand

```
   6428
 + 6591
 1 3019
```

carry → 1

3) Discard the carry term

result = 3019

<u>verification</u> :-

```
  6428
- 3409
  3019
```

Case-2 :- Sub^n ~~of~~ larger no. from smaller no.

1) Perform 125−1800

1) Perform 10's compliment on the substrand.

$$1800 \xrightarrow{9's} \begin{array}{r} 9999 \\ -1800 \\ \hline 8199 \end{array}$$

$$10's \rightarrow \begin{array}{r} 8199 \\ +1 \\ \hline 8200 \end{array}$$

2) Add 10's compliment reslt to minumand.

$$\begin{array}{r} 8200 \\ +0125 \\ \hline 8325 \end{array}$$

3) Perform 10's compliment to the above result & add −ve sign.

$$8325 \xrightarrow{10's} \begin{array}{r} 9999 \\ -8325 \\ \hline 1674 \\ +1 \\ \hline -1675 \end{array}$$

125 − 1800 = −1675

Verificat^n:-

8|4|21

Signed Binary numbers :

Positive → 0 to ∞ ; negative → −∞ to 0.

0 is put to represent the +ve sign ; 1 → −ve sign.

1) Perform sign magnitude with its sign for +9 & −9

+9 → 0 1001 → magnitude (+ve) (+10)

signbit

−9 → 1 1001 (−ve) (−10)

	8421
9	1001

When the sign is −ve then two operatns are performed

⇒ sign 1's compliment: changing all bits from 0 to 1 or 1 to 0 in magnitude

⇒ sign 2's compliment

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

−9 $\xrightarrow{1's}$ 10110

2's → +1

10111

⇒ for +ve signed no. no compliment is performed.

Decimal	signed 2's comp.	signed 1's comp.	signed magnitude
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
0	0000	0000	0000
-1	1111	1110	1001
-2	1110	1101	1010
-3	1101	1100	1011
-4	1100	1011	1100
-5	1011	1010	1101
-6	1010	1001	1110
+7	1001	1000	1111

Arithmetic Add^n^s

$$\begin{array}{r} +(\ ) \\ +(\ ) \\ \hline +(\ ) \end{array} \qquad \begin{array}{r} -(\ ) \\ -(\ ) \\ \hline -(\ ) \end{array} \qquad \begin{array}{r} +(\ ) \\ -(\ ) \\ \hline \text{larger no. sign } (\ ) \end{array}$$

1) Perform arithmetic add^n^ for +6 & +13 using 8 bits.

$$\begin{array}{rcl} +6 & \longrightarrow & 0000\ 0110 \\ +13 & \longrightarrow & 0000\ 1101 \\ \hline +19 & \longrightarrow & 0001\ 0011 \end{array}$$

2) -6 & ~~+13~~ $+13$

$-6 \Rightarrow 10000110$

$-6 \Rightarrow 11111001$

sign 1's $\Rightarrow$ $+1$

sign 2's $\Rightarrow$ 11111010

$-6 \Rightarrow 11111110$

$+13 \Rightarrow 00001101$

$1\,00000111 \Rightarrow +7$

Discard charge.

3) -6 & -13

$-13 \rightarrow 10001101$

1's comp. $\rightarrow 10110010$

$+1$

$-6 \Rightarrow 11111010$

$+13 \Rightarrow 00001101$

$+7 \quad 10000111$

H/w

4) $+6$ & -13

9/4/24

Floating Point Representatⁿ

It has 3 fields

1) sign
2) significant digits
3) Exponents

consider the no. Binary point

$1\,1\,1\,1\,0\,1\,.\,1\,0\,0\,0\,1\,1\,0$

point format

it represents that it is in floating point format

$\underbrace{1.1110110001 10}_{\text{Normalised form}} \times 2^{5}$ exponent

IEEE standard for floating

floating pt. no.'s

Standard developed by IEEE
- 32 bits ⟶ single precessⁿ format
- 64 bits ⟶ double precessⁿ format

→ So, we call ~~it~~ has IEEE stnd. or IEEE 754

representatⁿ of floating pt.

single precessn format

→ 32 bits

↓

3 fields

- sign (1bit)
- exponent (8 bit)
- mantissa (23 bits)

Format -

31	30	22	0
S	E	m	

sign bit - 0 → +ve

1 → -ve

exponent -

$E' = E + bias$

$E' = E + 127$ (bias-127)

→ It is called as excess-127 format.

Range of $E' = 0 < E' < 255$

$E \rightarrow -127 < E \leq 127$.

Double Precessn format

→ 64 bits

↓

3 bits

- sign (1 bit)
- exponent (11 bits)
- mantissa (52 bits)

Format -

63	62	51	0
S	E	m	

sign bit - 0 → +ve

1 → -ve

exponent -

$E' = E + bias$

$E' = E + 1023$

→ It is called as Excess-1023 format

Range of $E' = \cancel{0 < E < 2047}$

$0 < E < 2047$

$E \rightarrow -1022 < E \leq 1023$

1) Represent $(1259.125)_{10}$ in single & double precession.

2 | 1259 = 1
2 | 629 = 1
2 | 314 = 0
2 | 157 = 1
2 | 78 = 0
2 | 39 = 1
2 | 19 = 1
2 | 9 = 1
2 | 4 = 0
2 | 2 = 0
1

$0.125 \times 2 = 0.250$

$0.25 \times 2 = 0.5$

$0.5 \times 2 = 1.$

$(10011101011.001)_2$

Floating pt. representatn will be $1.0011101011001 \times 2^{10}$

For single precessn

sign → s = 0

exponent → E = 10

Mantissa ⇒ m = 0011101011001

$E' \Rightarrow E + 127$

$= 10 + 127$

$= (137)_{10}$

$E' = (10001001)_2$

2 | 137 1
2 | 68 0
2 | 34 0
2 | 17 1
2 | 8 0
2 | 4 0
2 | 2 0
1

31	30 – 22	0
0	10001001	00111010110010000000000
S	E	M

For double precessn

sign → s = 0

exponent → E = 10

mantissa → m = 0011101011001

$E' = E + 1023$

$= 10 + 1023 = 1033$

$E' = (10000001001)_2$

63	62 … 51	… 0
0	10000001001	0011101011001000---0
S	E	M

2) $(124.75)_{10}$

2	124	= 0
2	62	= 0
2	31	= 1
2	15	= 1
2	7	= 1
2	3	= 1
	1	

$0.75 \times 2 = 1.5$

$0.5 \times 2 = 1.$

$(1111100.11)_2$

Floating pt. representatn will be $(1.11110011 \times 2^6)_2$

for single precessn

$s = 0$

$E = 6$

$m = 11110011$

$E' = E + 127$

$E' = 6 + 127$

$E' = (133)_{10}$

$E' = (10000101)_2$

$$\begin{array}{r|l l}
2 & 133 & = 1 \\
2 & 66 & = 0 \\
2 & 33 & = 1 \\
2 & 16 & = 0 \\
2 & 8 & = 0 \\
2 & 4 & = 0 \\
2 & 2 & = 0 \\
 & 1 &
\end{array}$$

31	30 22	0
0	10000101	11110011000000000000000
S	E	M

for double precessn

$S = 0$

$E = 6$

$M = 11110011$

$E' = E + 1023 = 6 + 1023$

$E' = 1029$

$E' = (10000000101)_2$

$$\begin{array}{r|l l}
2 & 1029 & = 1 \\
2 & 514 & = 0 \\
2 & 257 & = 1 \\
2 & 128 & = 0 \\
2 & 64 & = 0 \\
2 & 32 & = 0 \\
2 & 16 & = 0 \\
2 & 8 & = 0 \\
2 & 4 & = 0 \\
2 & 2 & = 0 \\
 & 1 &
\end{array}$$

63	62 51	0
0	10000000101	1111001100000000000000000 00---0
S	E	M

12/4/21

Binary Codes

→ It is an assignment of informatn ~~to~~ with patterns

1) weighted codes (BCD or 8421, 2421, 5211)
2) Non-weighted codes (Excess-3, Gray code).
3) Sequential codes (8421, excess-3)
4) Reflectn codes (Excess-3, 2421, 3211)
5) Alphametic codes (ASCII, DIC, Hellerin)
6) Error Detecting of casting codes (Parity, Hamming code)

Weighted codes (BCD / 8421)

0 to 9 → 4 digits.

0 to 15 → Invalid status

Decimal no.	BCD digit (8421)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Q) Convert following decimal no.'s into BCD.

1) $(4)_{10}$ — $(0100)_2$

2) $(24)_{10}$ — $(00100100)_2$

3) $(1997)_{10}$ — $(0001100110010111)_2$

4) $(3805)_{10}$ — $(0011100000000101)_2$

5) $(165)_{10}$ — $(000101100101)_2$

BCD Addition

1) Perform BCD add^n for 4 & 7.

```
  4  →   0100
+ 7  →   0111
 ----   ------
 11      1011
        00110
      --------
(1)  (0001) 0001   = 11 (invalid state)
        (1)
```

→ When the value is greater than 9 add 6 of binary form gives BCD add^n.

→ Each four decimals represent the one value.

2) 6428 & 3208

```
6428 → (0110 0100 0010 1000)₂
3205 → (0011 0010 0000 0101)₂
9633    1001 0110 0010 (1101) → Invalid state
                        0110  → More than 9 added 6 to this binary digit.
        1001 0110 | 0011 | 0011
         9    6      3      3
```

H/w 3) 375 & 760

4) 4832 & 5294.

16/9/21 Octal Decimal Codes

Decimal	2421	5211	8 4 -2 -1
0	0000	0000	0000
1	0001	0001	0111
2	0010	0011	0110
3	0011	0101	0101
4	0100	0111	0100
5	1011	1000	1011
6	1100	1001	1010
7	1101	1100	1001
8	1110	1101	1000
9	1111	1111	1111

Non-weighted Codes

→ These codes are not assigned to any weight to each digit position.

→ Excess-3 & gray codes are non-weighted codes

Excess-3 codes

0 → 0000
1 → 0001
2 → 0010
13 → 1101
14 → 1110
15 → 1111

} Invalid states

0+3 → 3
1+3 → 4
2+3 → 5
3+3 → 6
4+3 → 7
5+3 → 8
6+3 → 9
7+3 → 10
8+3 → 11
9+3 → 12

→ we have 6 invalid states i.e., 0, 1, 2, 13, 14, 15.

Definitⁿ –

These code is obtained by adding 3 i.e. (0011) to each binary code that in the corresponding (8421).

→ It is a sequential & self-complementary code.

1) $5 \xrightarrow{\text{Excess-3}} 5+3 = 8$ (1000)

2) $21 \longrightarrow 21+33 = 54$ (0101 0100)

3) $4127 \longrightarrow 4127+3333 = 7460$ (0111 0100 0111 1010)

4) $24567 \longrightarrow 24567+33333 = 57890$ (0101 0111 1100 1100 1 1010)

5) $204.6 \longrightarrow 204.6+333.3 = 537.9$ (0101 0011 0111 . 1001)

Excess-3 addⁿ

1) Perform excess-3 addⁿ for 37 & 28.

$37 \xrightarrow{\text{excess-3}} 37+33 = 0011\ 0\ 1010$

$28 \longrightarrow 28+33 = 0101\ 1011$

. 11 00 0101

if carry = 1
add 3 i.e (0011)

if carry = 0
sub 3 i.e. (0011)
or
add 13, (1101)

37 + 28 → 65 $\xrightarrow{\text{excess-3}}$ 65+ = 98

```
  1100 0101
  1101 0011
 ----------
 ①1001 1000
```

↓
Discard

2) 247.6 & 359.4

```
  247.6
  359.4
 ------
  607.0
```

607.0 $\xrightarrow{\text{Excess-3}}$ 607.0 + 333.3 ⇒ 940.3

247.6 $\xrightarrow{\text{excess-3}}$ 247.6 + 333.3 = 5710.9

359.4 $\xrightarrow{\text{excess-3}}$ 359.4 + 333.3 = 6812.7

```
5710.9    0101 0111 1010 . 1001
6812.7    0110 1000 1100 . 0111
         ------------------------
          1100 0000 0111 . 0000
```

```
  1100   0000   0111 . 0000
  1101   0011   0011 . 0011
 ----------------------------
 ①1001   0011   1010 . 0011
```

↓
Discard

16/4/21 Grey cells ⟶ frank grey.

→ It is a non-weighted codes.

→ Reflected binary code.

→ unit distance code.

→ Cyclic code.

How to constract grey code

1 bit code

0
1

mirror image for 1 bit

0
1
1
0

2 bit code

0 0
0 1
1 1
1 0

mirror image for ~~2~~ bit

0 0
0 1
1 1
1 0
1 0
1 1
0 1
0 0

3 bit code

0 0 0
0 0 1
0 1 1
0 1 0
1 1 0
1 1 1
1 0 1
1 0 0

mirror image for 3 bit

0 0 0
0 0 1
0 1 1
0 1 0
1 1 0
1 1 1
1 0 1
1 0 0
1 0 0
1 0 1
1 1 1
1 1 0
0 1 0
0 1 1
0 0 1
0 0 0

0 0 0
0 0 1
0 1 1
0 1 0
1 1 0
1 1 1
1 0 1
1 0 0
1 0 0
1 0 1
1 1 1
1 1 0
0 1 0
0 1 1
0 0 1
0 0 0

~~4 bit code~~

~~0 0 0 0~~

4 bit Code

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
1 1 0 0
1 1 0 0
1 1 0 1
1 1 1 0
1 0 1 0
1 0 1 1
1 0 0 1

Grey code to binary code

$G_4 \quad G_3 \quad G_2 \quad G_1$

$1 \quad 0 \quad 0 \quad 0$

Step-1 - Take MSB of the given no. as it is

2 — In ack is resultant

3 — Repeat step-2 till end of the digits.

$$\begin{array}{cccc} G_4 & G_3 & G_2 & G_1 \\ 1 & 0 & 0 & 0 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & 1 & 1 & 1 \\ B_4 & B_3 & B_2 & B_1 \end{array}$$

$B_4 = G_4$

$B_3 = G_4 \oplus G_3$

$B_2 = G_3 \oplus G_2$

$B_1 = G_2 \oplus G_1$

1) 1110

2) 114

3) 10110

Binary to Grey code

Step-1 – Write MSB of given no. as it is

Step-2 – no XOR operation of MSB of given no with successive digit of the given no.

Step-3 – Repeat step-2 till end of the digit.

B_4	B_3	B_2	B_1	B_0
1	0	0	1	0
↓	↓	↓	↓	↓
1	1	0	1	1
G_4	G_3	G_2	G_1	G_0

$G_4 = B_4$

$G_3 = B_4 \oplus B_3$

$G_2 = B_3 \oplus B_2$

$G_1 = B_2 \oplus B_1$

$G_0 = B_1 \oplus B_0$

2) 1111

B_4	B_3	B_2	B_1
1	1	1	1
↓	$\oplus$ ↓	$\oplus$ ↓	$\oplus$ ↓
1	0	0	0
G_4	G_3	G_2	G_1

$G_4 = B_4$

$G_3 = B_4 \oplus B_3$

$G_2 = B_3 \oplus B_2$

$G_1 = B_2 \oplus B_1$

1) Convert the following into gray no.

1) $(3A7)_{16}$

$(0011\ 1010\ 0111)_2 \Rightarrow (001001110100)_G$

2) $(527)_8$

$(101010111)_2 \Rightarrow$

3) $(652)_{10}$

2) Convert grey no. 10110010 into

$(10110010)_G$

$\oplus$

$(11011100)_2$

1) Hexadecimal

$(1101|1100)_2 = (DC)_{16}$

D | C

2) Octal

$(011|011|100)$

3 | 3 | 4

$(334)_8$

3) Decimal $(01101100)_2$

$2^7 \times 1 + 2^6 \times 1 + 2^5 \times 0 + 2^4 \times 1 + 2^3 \times 1 + 2^2 \times 1 + 2^1 \times 0 + 2^0 \times 0$

$= (220)_{10}$

Alphanumeric codes

→ These are used to encourt the character

→ They are used to transmit the data b/w computers & its i/p. o/p devices such as printers, keyboards.

→ The most popular alphanumeric codes are ASCII code. & EBCDIC code.

	64	32	16	8	4	2	1
ASCII code (7 bits)	0	0	0	0	0	0	0

A $\xrightarrow{65}$ 1000001

a $\xrightarrow{97}$ 1100001

z $\xrightarrow{90}$ 1011010

space $\xrightarrow{32}$ 0100000

1) code the following into ASCII

BIRT

B $\xrightarrow{66}$ 1000 0010

I $\xrightarrow{73}$ 1001001

R $\xrightarrow{82}$ 1010010

T $\xrightarrow{84}$ 1010100

H $\xrightarrow{72}$ 1001000

EBCDIC (8bits) 0 0 0 0 0 0 0 0

128 64 32 16 8 4 2 1

→ It is Extended Binary Coded Decimal Interchange

→ It is an 8 bit alphanumerical code

→ This code uses BCD as the basic of the alphanumeric code.

19/4/24 Unit -1

Error correcting and detecting codes

→ when binary data is transmitted and processed, it is susceptible to noise that can alter or distort its contents.

→ The 1's may get changed to 0's & 0's to 1's.

Error detecting codes

→ one of the most used technique for detecting error is parity.

→ It is simplest technique.

→ Parity means adding an extra bit, it is known as parity bit.

→ we add parity bit to each word being transmitted.

⇒ there are 2 types of parity.

1) Even parity

2) Odd parity

Even Parity – for even parity, the parity bit is set 0 or 1 at the transmitter such that the total no. of 1 bits in the given code word including the parity bit is an even number.

Even parity.

	Even parity
1 0 0 1 → 9	0
~~1 0 0 0~~	+
1 0 0 0 → 8	1 even

	even parity
1001	0
1011	1
1000	1

odd parity - for odd parity, the parity bit is set to 0 or 1 at the transmitter such that the total no. of 1 bits in the given code word including the parity bit is an odd no..

	odd
0 0 0 1	0
1 0 0 1	1
1 0 1 1	0

odd & even parity in 8421 - BCD code

Decimal	8421	Parity odd	Parity even
0	0000	1	0
1	0001	0	1
2	0010	0	1
3	0011	1	0
4	0100	0	1
5	0101	1	0
6	0110	1	0
7	0111	0	1
8	1000	0	1
9	1001	1	0

1) In an even parity scheme, which of the following contain an error

1) 1010 1010

In even parity scheme, if no. of 1's are even then there is no error.

⇒ no. of 1's = 4

⇒ It is even, so no error

2) 11110110

⇒ no. of ones = 6

⇒ It is even, so no error

3) 10111001

⇒ no. of 1's = 5

It is odd; there is a error.

2) In an odd parity scheme, which of the following words contain no error

1) 10110111

⇒ no. of 1's are 6

There is a error

2) 10011010

⇒ 1's are 4

There is a error.

3) 11101010

⇒ 1's are 5

There is no error

* In block of data shown in table is to be stored on a magnetic tape. Create the row & column parity bits for the data using odd parity.

	Data	odd
	10110	0
	10001	1
	10101	0
	00010	0
	11000	1
	00000	1
	11010	0
odd	01101	0

* In block of data shown in table - - - using even parity

	data	even
	10110	1
	10001	0
	10101	1
	00010	1
~~even~~	11000	0
	00000	0
	11010	1
even	10010	0

1) Encode data bits 1101 into 7 bit even parity hamming code?

data = 1101

m = 4 bits

To find parity bit

$p=3$

Total hamming code :

$m+p$

$4+3=7$ bits

$3^{k-1} \Rightarrow 1, 2, 4$

$16 = 1, 2, 4, 8$

$(m+p)=7$

$2^p \geq (m+p) < 1$

$p=1$

$2^1 \geq 4+1+1$ (not satisfied)

$p=2$

$2^2 \geq 4+2+1$ (not satisfied)

$p=3$

$2^3 \geq 4+3+1$

$8 \geq 3$ (satisfied)

		m_1		m_2	m_3	m_4
1	2	3	4	5	6	7
001	010	011	100	101	110	111
P_1	P_2	1	P_3	1	0	1

XOR

00 = 0

10 = 1

01 = 1

11 = 0

$P_1 = XOR(3,5,7) = XOR(1,1,1) = 1$

$P_2 = XOR(3,6,7) = XOR(1,0,1) = 0$

$P_3 = XOR(5,6,7) = XOR(1,0,1) = 0$

the 7 bit even parity hamming code is 1010101.

2) given a 8 bit data word 01011011. Create a 12 bit composite hamming code that connects & detect single error

Data = 01011011

$m = 8$ bit

parity bits required

$p=4$

Total bits $\Rightarrow 8+4 = 12$ bits.

2^{k-1}

$k=1 \Rightarrow 2^0 = 1 = P_1$

$k=2 \Rightarrow 2^1 = 2 = P_2$

$k=3 \Rightarrow 2^2 = 4 = P_3$

$k=4 \Rightarrow 2^3 = 8 = P_4$

1 2 3 4 5 6 7 8 9 10 11 12

0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100

P_1 P_2 0 P_4 1 0 1 P_8 1 0 1 1

P_1 = XOR (3, 5, 6, 7, 9, 11) = (0, 1, 1, 1, 1) = 0

P_2 = XOR (3, 6, 7, 10, 11) = (0, 0, 1, 0, 1) = 0

P_4 = XOR (5, 6, 7, 12) = (1, 0, 1, 1) = 1

P_8 = XOR (4, 10, 11, 12) = (1, 0, 1, 1) = 1

The 17 bit hamming code is 000110111011.

3) A 12 bit hamming code word containing 8 bits of data and 4 parity bits is read from memory. What is the orginal 8 bit word it is the 12 bit read out as follows.

12 bits hamming code

1) 100011101010 $\longrightarrow$ m = 8
p = 4

If we remove parity bits from the hamming code then the bits are message 8 bits

12 bits $\longrightarrow$ P_1, P_2, P_4, P_8

1 0 0 0 $\longrightarrow$ remove

$\Rightarrow$ 0111 1010

2) P_1 P_2 P_4 P_8
101110000110 $\Rightarrow$ 1100 0110

22/9/21

Detect & collect errors if any in the even parity hamming code word and write the collect code.

1) 1100110

1	2	3	4	5	6	7
001	010	011	100	101	110	111
1	1	0	0	1	1	0

$e_1 = xor(1,3,5,7) = (1,0,1,0) = 0$

$e_2 = xor(2,3,6,7) = (1,0,1,0) = 0$

$e_3 = xor(4,5,6,7) = (0,1,1,0) = 0$

$\Rightarrow$ 000 . No error.

2) 0111110

1	2	3	4	5	6	7
001	010	011	100	101	110	111
0	1	1	1	1	1	0

$e_1 = xor(1,3,5,7) = (0,1,1,0) = 0$

$e_2 = xor(2,3,6,7) = (1,1,1,0) = 1$

$e_3 = xor(4,5,6,7) = (1,1,1,0) = 1$

$\Rightarrow$ 011 - error is present at 3rd bit.

Replace bit 3 with 0, 1 → 0

$\therefore$ The collect code word is 010 1110.

3) 1010111

1	2	3	4	5	6	7
001	010	011	100	101	110	111
1	0	1	0	1	1	1

$e_1 = xor(1,3,5,7) = (1,1,1,1) = 0$

$e_1 = xor(2,3,6,7) = (0,1,1,1) = 1$

$e_3 = xor(4,5,6,7) = (0,1,1,1) = 1$

$(0\,1\,1) \longrightarrow$ error is p|t at 3rd bit.

replace bit 3 with 0 $(1 \longrightarrow 0)$

$\therefore$ The correct code word is 1000111.

Boolean Algebra

→ Boolean Algebra is the algebraic structure and is defined with a set of binary elements

i.e., $B = \{0, 1\}$.

→ It uses a set operators $(+, \cdot)$ and some unproved axioms or postulates

→ In 1854, George Boole developed an algebraic system called Boolean Algebra.

→ In 1938, Shannon introduced a two values boolean algebra called switching algebra.

Axiomatic Definition of Boolean Algebra.

* Boolean algebra is an algebraic structure defined by a set of elements B, together with two binary operators '+' & '·' provided that the following Huntington postulates are satisfied.

Postulates of Boolean Algebra

	(a)	(b)	
1)	Closure property		
2)	$x+0=x$	$x\cdot 1=x$	Identity
3)	$x+y=y+x$	$x\cdot y=y\cdot x$	Commutative law
4)	$x+(y\cdot z)=(x+y)\cdot(x+z)$	$x(y+z)=(x\cdot y)+(x\cdot z)$	Distributive law
5)	$x+(x)'=1$	$x\cdot(x)'=0$	complement

Duality

Principle – It states that every algebraic exp^n deductible from postulates of boolean algebra, remains valid if the operations are interchanged.

To obtain dual of an algebraic expression.

* Interchange OR and AND operators
* Replace with 1's by 0's & 0's by 1's.

1) $x+0=x$

$x \cdot 1 = x \longrightarrow$ duality.

2) $x+x'=1$

$x \cdot x'=0$

3) $x \cdot (y+z) = x \cdot y + x \cdot z$

$x+(y \cdot z) = (x+y) \cdot (x+z)$

4) $x+y=y+x$, $x \cdot y = y \cdot x$

Theolems of Boolean Algebra.

	(a)	(b)	
1)	$x+x=x$	$x \cdot x = x$	
2)	$x+1=1$	$x \cdot 0 = 0$	
3)	$((x)')'=x$		Involution
4)	$x+(y+z)=(x+y)+z$	$x \cdot (y \cdot z)=(x \cdot y) \cdot z$	Associate
5)	$(x+y)'=(x)' \cdot (y)'$	$(x \cdot y)'=(x)'+(y)'$	Demorgan's law
6)	$x+(x \cdot y)=x$	$x \cdot (x+y)=x$	Absorption law

Theolem-1

1(a) : $x+x=x$

1(b) : $x \cdot x = x$

$x+x=x$

L.H.S $x+x$

$(x+x) \cdot 1$ (Post 3(b))

$= (x+x)\cdot(x+x')$ (Post 5(a))

$= (x)\cdot(x+x')$ (Post 4(b))

$= x\cdot x + x\cdot x'$

$= x+0$

$= x$

$x\cdot x = x$

L.H.S. $x\cdot x+0$ (Post 3(b))

$x\cdot x+ x x'$ (Post 4(a))

$x(x+x')$ (Post 5(a))

$x\cdot 1$

$= x$

23/4/21

<u>Theorem : 2</u>

2(a) : $x+1=1$

2(b) : $x\cdot 0=0$

<u>Proof</u> :- $x+1=1$

L.H.S. $= (x+1) = 1 = (x+1)(x+x')$

$x+(1\cdot x')$ (from 4(a))

$= x+x'$

$= 1$ = R.H.S.

L.H.S. = R.H.S.

∴ Hence proved.

$x \cdot 0 = 0$

$x + 1 = 1$ (duality)

L.H.S: $(x+1) \cdot 1$

$(x+1)\,(x+x')$ (from 5(a))

$= x + (1 + x')$ (from 4(a))

$= x + x' = 1$

$= RHS$

LHS = RHS ∴ Hence proved.

Theorem-3. $((x)')' = x$

x	x'	$(x')'$
0	1	0
1	0	1

RHS LHS

$2^n = 2^1 = 2$

0, 1

LHS = RHS ∴ Hence proved

Theorem-4.

4(a): $A + (B + C) = (A + B) + C$

4 A	2 B	1 C	B+C	A+(B+C)	A+B	(A+B)+C
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1

1	0	0	0	1	1	1
1	0	1	1	1	1	1
0	1	0	1	1	1	1
1	1	1	1	1	1	1
				LHS		RHS

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

L.H.S=R.H.S. ∴ Hence proved.

4(b) : $x\cdot(yz) = (xy)\cdot z$.

4 x	2 y	1 z	yz	$x(yz)$	xy	$(xy)z$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	1	0
1	1	1	1	1	1	1
				LHS		RHS

LHS = RHS

∴ Hence proved.

Theorem-5

5(b): $(A+B)' = (A)' \cdot (B)'$

A	B	A+B	(A+B)'	A'	B'	A'·B'
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

LHS RHS

LHS = RHS ∴ Hence proved.

Theorem - 6

~~A+AB~~ $A + (A+B) = A$

let $A = x$, $B = y$

$\Rightarrow x + (x+y) = x$

LHS. $x + xy$

$x(1+y) \cdot 1$

$x(1+y)(y+y')$ (5a)

$x[(y+1)(y+y')]$ (4b)

$x(y+1+y')$

$x(y+y')$

$x(1) = x =$ R.H.S

LHS = RHS ∴ Hence proved.

26/4/21

Compliment of a function:

→ Compliment of a fn. is obtained by Demorgan's law.

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

1) find the compliment of given boolean fn.

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

L.H.S = $(A+B+C)'$ let $A+B = X$

$\Rightarrow (X+C)'$ $\quad [\because (A+B)' = A'.B']$.

$\Rightarrow X'.C'$

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

$= A'.B'.C' = R.H.S.$

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

$(x')' = x$.

2) $F_1 = x'yz' + x'y'z$.

$(F_1)' = (\underbrace{x'yz'}_{A} + \underbrace{x'y'z}_{B})'$

$= (x'yz')'\cdot(x'y'z)'$

$= [(x')' + (y)' + (z')'] \cdot [(x')' + (y')' + (z)']$

$= [x+y'+z]\cdot(x+y+z']$ $\cancel{\ldots}$

3) $F_2 = x(y'z'+yz)$

$(F_2)' = [\underbrace{x}_{A}\cdot(\underbrace{y'z'+yz}_{B})]'$

$= (x)' + (y'z'+yz)'$

$= [x' + (y'z')'\cdot(yz)']$

$= \left[x' + [(y')' + (z')'] \cdot [(y)' + (z)']\right]$

$= \left(x' + (y+z) \cdot (y'+z')\right)$

$= \left[x' + yy' + yz' + zy' + zz'\right]$

$yy' = 0$

$zz' = 0$

$= x' + yz' + zy'$

(3 literals)

4) Find the compliment of the fn. by taking the dual & complementing each literal.

1) $F_1 = x'yz' + x'y'z$

duality

$+ \rightarrow \cdot$ | $0 \rightarrow 1$

$\cdot \rightarrow +$ | $1 \rightarrow 0$

Dual of $F_1 = (x'+y+z') \cdot (x'+y'+z)$

$(F_1)' = (x'+y+z')' \cdot (x'+y'+z)'$

$= ((x')' + y' + (z')') \cdot ((x')' + (y')' + z')$

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

2) $F_2 = x(y'z' + yz)$

5) Find the compliment of the following expression.

i) $(A'B+CD)E'+E$

A) $\left[(A'B+CD)E'+E\right]'$

$= \left[((A'B+CD)E')' \cdot (E)'\right]$

$= \left[((A'B)' \cdot (CD)' + E) \cdot E'\right]$

$= \left[(A+B') \cdot (C+D') + E\right]E'$.

6) Find the compliment of $F=wx+yz$ then S.T. $FF'=0$?

A) $F = wx+yz$

$F' = (wx+yz)'$

$= (wx)' \cdot (yz)'$

$F' = (w'+x') \cdot (y'+z')$

$FF' = (wx+yz) \cdot (w'+x')(y'+z')$

$= (wx+yz) \cdot (w'y'+w'z'+x'y'+x'z')$

$= wxw'y' + wxw'z' + wxx'y' + wxx'z' + yzw'y' +$

$yzw'z' + yzx'y' + yzx'z'$

$= 0+0+0+0+0+0+0+0 = 0$ $\quad (\because x \cdot x' = 0)$

27/4/21

<u>Operator precendence</u>

The operator precedence for evaluating boolean exp.n is

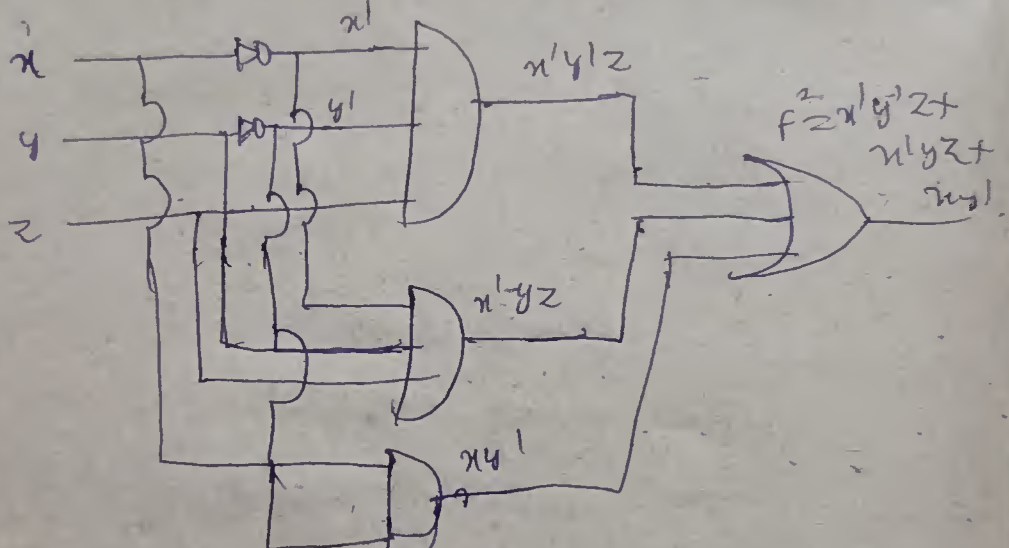
1) P $\{\}$, ( )

2) NOT (') or ($^{-}$)

3) AND (&) or ($\cdot$)

4 OR (|) or (+)

1) simplify i) $F_2 = x'y'z + x'yz + xy'$ by logic diagram?

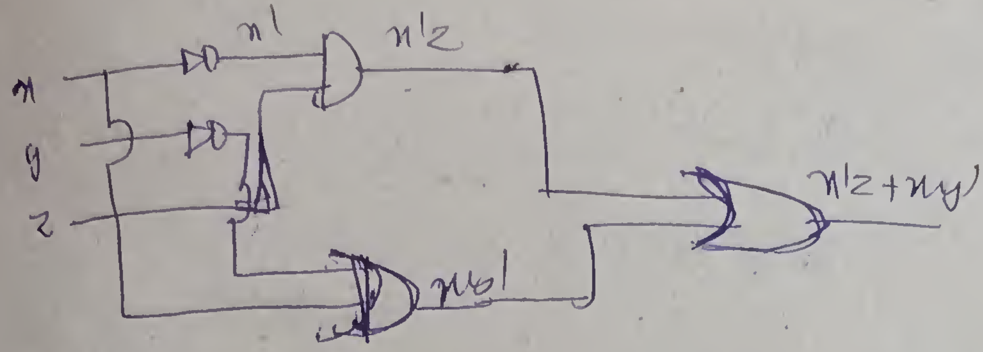


2) simply F_2 using boolean rule.

$F_2 = x'y'z + x'yz + xy'$

$= x'z(y'+y) + xy' \quad [\because y+y'=1]$

$= x'z + xy'$



3) Reduce the given exprs to min. no. of literals.

1) $x(x'+y)$

$xx'+xy$ $\left(\because xx'=0\right]$

$0+xy = xy$

2) $x+x'y$

$(x+x')(x+y)$

$(1)(x+y)$

$\Rightarrow x+y$

$\left(\begin{array}{l} x+yz=(x+y)(x+z) \\ \text{distributive law} \\ x+x'=1. \end{array}\right)$

3) $(x+y)(x+y')$

$\Rightarrow xx+xy'+yx+yy'$

$= x+xy'+xy+0$

$= x+x[y+y']$

$= x+x(1)$

$= x+x$

$\Rightarrow x$

4) $xy+x'z+yz$

$xy+x'z+yz(1)$

$xy+x'z+yz(x+x')$

$xy+x'z+xyz+x'yz$

$xy(1+z)+x'z(1+y)$

$xy(1)+x'z(1)$

$xy+x'z$ (consensus theorem)

$\left[\begin{array}{l} 1 = x+x' \\ 1+z=1 \\ 1+y=1 \end{array}\right]$

5) $xyz+x'y+xyz'$

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

$xy(1)+x'y$

$y(x+x')$

$= y$

$(z+z'=1)$

$x+x'=1$

6) $(x+y)(x'+z)(y+z) = (x+y)(x'+z)$

$\left[\because x(y+z) = x\cdot y + x\cdot z\right]$

$L.H.S. = (x+y)(x'+z)(y+z)$

$= (x+y)(x'+z)(xx'+y+z)$

$[x\cdot x = x]$

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

$= \underbrace{(x+y)}_{x}\underbrace{(x+y+z)}_{y\ \ z}\underbrace{(x'+z)}_{x}\underbrace{(x'+z+y)}_{y\ \ z}$

$= \left[(x+y)(x+y) + (x+y)(z)\right]\left[(x'+z)(x'+z) + (x'+z)y\right].$

$= \left[(x+y) + (x+y)z\right]\left[(x'+z) + (x'+z)y\right].$

$= \left[x+y(1+z)\right]\left[x'+z(1+y)\right]$

$\left[\because 1+z = 1,\ 1+y = 1\right]$

$\Rightarrow \left[(x+y)(1)\right]\left[(x'+z)(1)\right]$

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

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

$xx'+xy'+xz+yx'+yy'+yz+z'x'+z'y'+z'z$

$0+xy'+xz+yx'+0+yz+x'z'+y'z'+0$

$xy'+x(y+y')z+x'y+yz+x'z'+y'z$

$\underline{xy'}+\underline{xyz}+\underline{xy'z}+x'y+\underline{xyz}+x'z'+y'z'$

$xy'(1+z)+yz(1+x)+x'y+x'z'+y'z'$

$xy'+yz+x'y+x'z'+y'z'$

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

$\left[\because 1+z = 1,\ 1+x = 1\right.$

$\left.xx' = yy' = zz' = 0\right]$

8) $(A+B)' \quad A'+B'$

$\Rightarrow (A'\cdot B')\left((A')'\cdot(B')'\right)$

$\Rightarrow (A'\cdot B')\cdot(A\cdot B)$

$= A\cdot A'\cdot B\cdot B'$

$= 1\cdot 0 \Rightarrow 0$

Demorgan's law

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

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

$(\because AA' = B\cdot B' = 0)$.

28/1/21

1) Reduce the following boolean exp. to the indicated no. of literals.

1) $A'C'+ABC+AC'+AB'$ to 2 literals.

$\left[\begin{array}{l} \because A+A'=1 \\ C+C'=1 \\ B+B'=1 \\ x+yz=(x+y)(x+z) \end{array}\right]$

$\Rightarrow A'C'+ABC+AC'+AB'$

$\Rightarrow C'(A+A')+ABC+AB'$

$\Rightarrow C'(1)+ABC+AB'$

$\Rightarrow \underset{x}{\underline{C'}}+\underset{Y\ \ Z}{\underline{AB\,C}}+AB'$

$\Rightarrow (C'+AB)(C'+C)+AB' \Rightarrow (C'+AB)(1)+AB' \Rightarrow C'+AB+AB'$

$\Rightarrow C'+AB+AB' \Rightarrow C'+A(B+B') \Rightarrow C'+A$

2) $(x'y'+z)'+z+xy+wz$ to 3 literals

$(x'y'+z)'+z+xy+wz$

$((x'y')'\cdot(z)')+z+xy+wz$

$((x')'+(y')')\cdot z'+z+xy+wz$

$(x+y)z'+z(1+w)+xy$

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

$xz' + yz' + z + xy$

$xz' + (z+y)(z+z') + xy$

$xz' + (z+y) + xy$

$xz' + z + y + xy$

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

$(z+x)(1) + y(1+x)$

$z + x + y$

$\Rightarrow x + y + z$

$\left[\begin{array}{l} \because \text{Demorgan's law} \\ (a+b)' = a'b' \\ (a'.b')' = \bar{a} + \bar{b} \\ 1 + w = 1 \\ A + BC = (A+B)(A+C) \\ z + z' = 1 \\ 1 + x = 1 \end{array} \right]$

3) $ABCD + A'BD + ABC'D + A'D$ to 2 literals ?

$ABCD + A'BD + ABC'D + A'D$

$ABD(C+C') + A'BD + A'D$

$ABD + A'BD + A'D$

$BD[A+A'] + A'D$

$BD + A'D$

$D(B+A')$

$\left[\begin{array}{l} \because C + C' = 1 \\ A + A' = 1 \end{array} \right]$

4) Reduce the exp. $f = A\left[B + \bar{C}\left(\overline{AB + AC}\right)\right]$.

$f = A\left[B + \bar{C}\left(\overline{AB + A\bar{C}}\right)\right]$

$f = A\left(B + \bar{C}\left(\overline{AB} \cdot \overline{A\bar{C}}\right)\right)$

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

$\Rightarrow A\left[B + \bar{C}(\bar{A} + \bar{B}) \cdot (\bar{A} + C)\right]$

$\Rightarrow A\left[B + \bar{C}(\bar{A} + \bar{B}C)\right]$

$\left[\begin{array}{l} \because \text{Demorgan's law} \\ \overline{A+B} = \bar{A} \cdot \bar{B} \\ \overline{AB} = \bar{A} + \bar{B} \\ a + bc = (a+b)(a+c) \\ \text{distributive property} \\ C\bar{C} = 0 \end{array} \right]$

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

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

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

$= AB+0 = AB$

5) P.T. $A\bar{B}C+B+B\bar{D}+AB\bar{D}+\bar{A}C = B+C$

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

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

$\underbrace{A\bar{B}C}_{z\ y} + \underbrace{B}_{x} + \bar{A}C$

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

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

$\Rightarrow B+C$

∵ $1+x=1$

distributive law

$x+yz=(x+y)(x+z)$

$B+\bar{B}=1$

$A+\bar{A}=1$

Q) Find the complement & the dual of the fn. given below & then reduce it to min. no. of literals in each case.

1) $f=[(a\bar{b})a][(a\bar{b})b]$

complement $\Rightarrow \bar{f}=\overline{[(\overline{ab})a][(\overline{ab})b]}$

$\bar{f}=[\overline{\overline{ab}\,a}+\overline{\overline{ab}\,b}]$

$\Rightarrow \{\overline{(\bar{a}+\bar{b})a}+\overline{(\bar{a}+\bar{b})b}\}$

$\Rightarrow \overline{\bar{a}+\bar{b}}+\bar{a}+\overline{\bar{a}+\bar{b}}+\bar{b}$

$\Rightarrow \underline{a\cdot b}+\bar{a}+\underline{a\cdot b}+\bar{b}$

$\bar{f}\Rightarrow ab+\bar{a}+\bar{b}$

$\Rightarrow (\bar{a}+a)(\bar{a}+b)+\bar{b} \Rightarrow (\bar{a}+b)+(\bar{b}) \Rightarrow \bar{a}+b+\bar{b}$

$= \bar{a}+1$

$= 1$

∵ $(ab)' = \overline{ab} = \bar{a}+\bar{b}$

$x+yz=(x+y)(x+z)$

$\bar{a}+a=1$

$\bar{b}+b=1$

$1+x=1$

29/4/21 **CONICAL AND STANDARD FORMS**

→ minterms → AND → SOP, $(x' \cdot y' \cdot z')$

→ maxterms → OR → POS, $(x' + y' + z')$

$x \to x$, $x \to x'$

$x\,y$

$x \,\&\, y$

$2^n \to 4$ terms

$\Rightarrow n = 3 \Rightarrow 2^3 = 8$.

$x = 1$

$x' = 0$

$\left\{ \begin{array}{l} x+y \\ x+y' \\ x'+y \\ x'+y' \end{array} \right\}$ maxterms

$\left\{ \begin{array}{l} x \cdot y \\ x \cdot y' \\ x' \cdot y \\ x' \cdot y' \end{array} \right\}$ minterms

x	y	z	minterm (·) Term	Designat^n (m)	maxterm (+) Term	Designat^n (M)
0	0	0	$x'y'z'$	m_0	$x+y+z$	M_0
0	0	1	$x'y'z$	m_1	$x+y+z'$	M_1
0	1	0	$x'yz'$	m_2	$x+y'+z$	M_2
0	1	1	$x'yz$	m_3	$x+y'+z'$	M_3
1	0	0	$xy'z'$	m_4	$x'+y+z$	M_4
1	0	1	$xy'z$	m_5	$x'+y+z'$	M_5
1	1	0	xyz'	m_6	$x'+y'+z$	M_6
1	1	1	xyz	m_7	$x'+y'+z'$	M_7

1). Express the boolean fn. $F = A + B'C$ as sum of minterms or SOP

$$F = A + B'C$$

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

$$= A(BC + BC' + B'C + B'C') + AB'C + A'B'C.$$

$$= ABC + ABC' + AB'C + AB'C' + AB'C + A'B'C.$$

$$\begin{array}{ccccc} = ABC & + ABC' & + AB'C & + AB'C' & + A'B'C \\ 111 & 110 & 101 & 100 & 001 \end{array}$$

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

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

Product of maxterms (product of sum terms) (POS)

$$\prod (a+b+c) \cdot (a'+b'+c) \cdot (+ +)$$

$$[(x+yz) = (x+y)(x+z)].$$

2) Express the boolean fn. $f = xy + x'z$ as product of max terms or POS?

$$F = xy + x'z \Rightarrow (xy+x')(xy+z)$$

$$\Rightarrow (x'+x)(x'+y)(z+x)(y+z) \qquad [\because x + x' = 1].$$

$$\Rightarrow (x'+y)(x+z)(y+z)$$

$$\Rightarrow (x'+y+zz')(x+yy'+z) + (xx'+y+z)$$

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

$$\begin{array}{cccc} \Rightarrow (x'+y+z) & (x'+y+z') & (x+y+z) & (x+y'+z) \\ 100 & 101 & 000 & 010 \\ m_4 & m_5 & m_0 & m_2 \end{array}$$

$\Rightarrow$ $m_4 + m_5 + m_0 + m_2$ $\Rightarrow$ $\Sigma\Pi m(0, 2, 4, 5)$

~~$\Rightarrow f = \Sigma m\ 1, 3, 7$~~ $\Rightarrow$ $\Sigma m(1, 3, 5, 7, 9, 11, 13, 15)$.

- product of maxterm (POS) - $F(x, y, z) = (xy + z)(xz + y)$.

3) $F = (xy + z) . (xz + y)$

$= (z + x)(z + y)(y + x)(y + z)$

$= (z + x)(x + y)(z + y)$

$= (x + yy' + z)(x + y + zz')(xx' + y + z)$

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

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

0 0 0	0 1 0	0 0 1	1 0 0
m_0	m_2	m_1	m_4

$= \Sigma\Pi m(0, 1, 2, 4)$.

30/4/24

1) Convert the given exp.n from SOP to POS

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

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

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

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

1 0 0 1	1 0 0 0	0 0 1 1	0 0 0 1	1 1 1 0

$= m_9 + m_8 + m_3 + m_1 + m_{14}$

minterms or SOP $= \Sigma m(1, 3, 8, 9, 14)$.

no. of minterms $\Rightarrow 2^n = 2^4 = 16 \rightarrow$ i.e. (0 - 15)

maxterms $\prod m(0,2,4,5,6,7,8,10,11,12,13,15)$

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

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

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

2) POS $\rightarrow$ SOP $\quad F(A,B,C) = (\bar{A}+B)(B+\bar{C})$.

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

$= (\bar{A}+B+C\bar{C})(A\bar{A}+B+\bar{C}) \qquad [C\bar{C} = A\bar{A} = 0]$.

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

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

$\quad 1\ 0\ 0 \qquad 0\ 0\ 1 \qquad 1\ 0\ 1$

$\quad m_4 \qquad m_1 \qquad m_5$

$= \prod m(1,4,5)$.

no. of minterms $2^n = 2^3 = 8 \rightarrow (0-7)$.

$SOP = \sum m(0,2,3,6,7)$

<u>Binary storage</u> & <u>registers</u>

→ Digital computer ⟷ binary i.e (0,1)

data is stored

* Binary cell → capable storing bit.

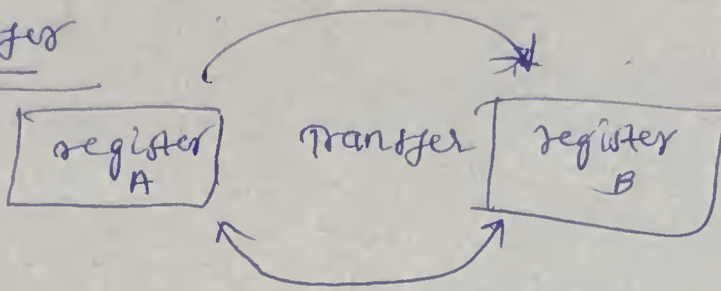
* <u>Registers</u> → group of binary cells.

'n' binary cells → 'n' bit i/p data single time

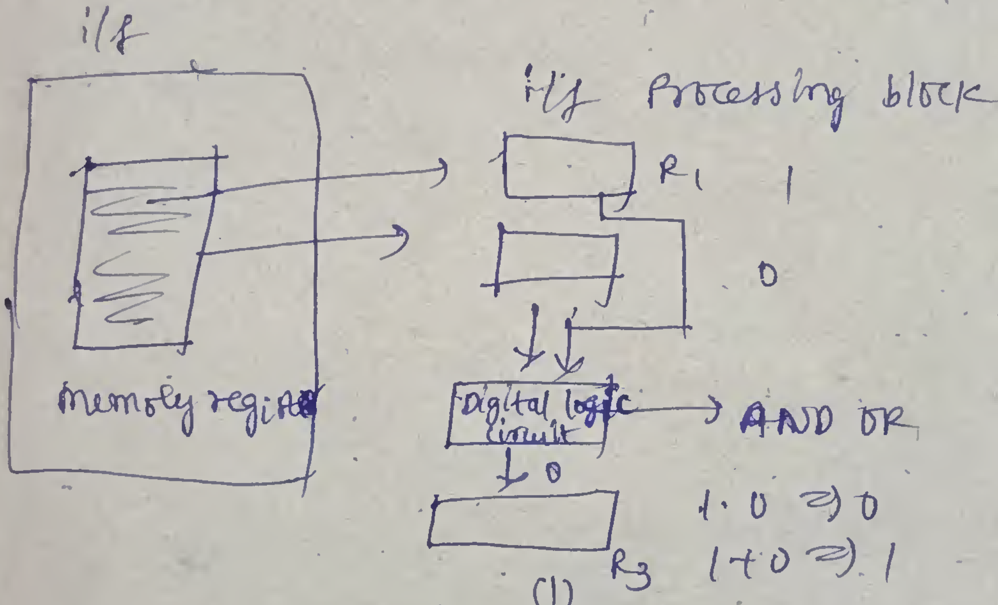
16 bit register → 2^{16} possible   store 1 bit data (0) (or) (1).

excess-3 code (3)

register Transfer



register transfer operator



4/5/24

UNIT - 2

Gate Level Minimization

K-maps:- The Karnaugh map (K-map) is a chart or a graph composed of an arrangement of adjacent cells each representing a particular combinatⁿ of variable in SOP form

→ Any boolean exp. can be expressed in standard SOP form or POS form

Two variable K-map

* A 2 variable K-map has 2^n squares $n=2 \Rightarrow 2^2 \Rightarrow 4$ squares

→ These squares are called cells.

→ Each square on the K-map represents the minterm

SOP → 0 → compl

f(A,B)

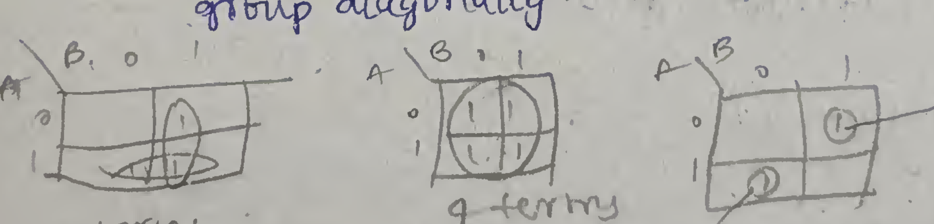
A \ B	0	1
0	A'B'	A'B
1	AB'	AB

A \ B	0	1
0	m_0	m_1
1	m_2	m_3

A \ B	0	1
0	0	1
1	2	3

⇒ any adjacent grouping is possible, can group all, but can't group diagonally

grouping



2 terms

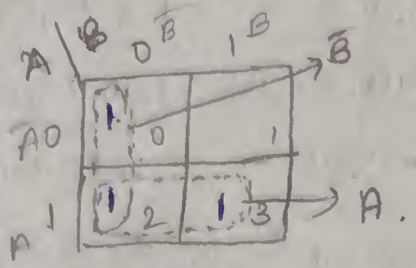
4 terms

1 term

Max terms – '0'
min terms – '1'
square

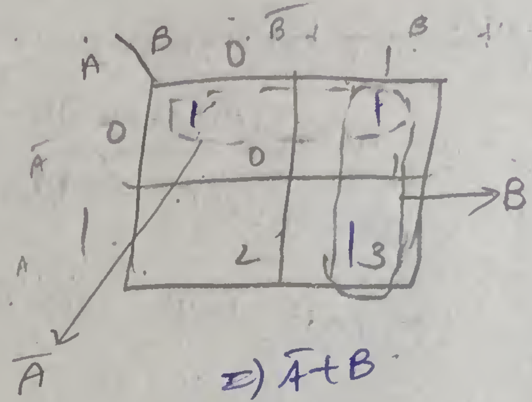
1) simplify the following boolean fun. using 2 variable k-map.

1) $F = \Sigma m(0,2,3)$



$\Rightarrow A + \bar{B}$

2) $F = \Sigma m(0,1,3)$

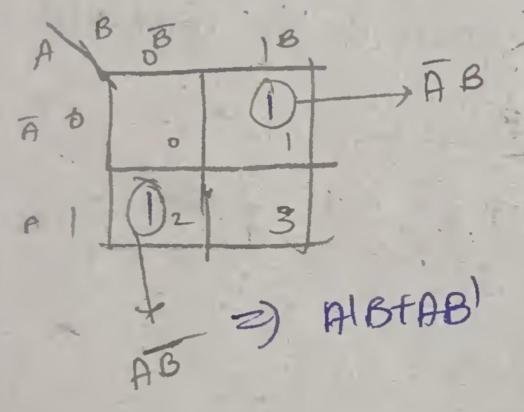


$\Rightarrow \bar{A} + B$

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

1 0 0 1

$\Rightarrow \Sigma m(2,1)$

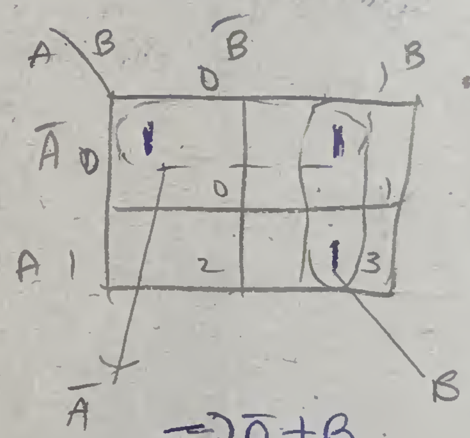


$\Rightarrow A'B + AB'$

4) $F = \bar{A}\bar{B} + \bar{A}B + AB$

0 0 0 1 1 1

$\Rightarrow \Sigma m(0,1,3)$

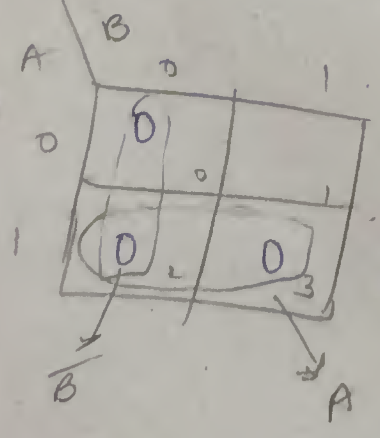


$\Rightarrow \bar{A} + B$

**

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

$\Rightarrow \Pi m(0,2,3)$

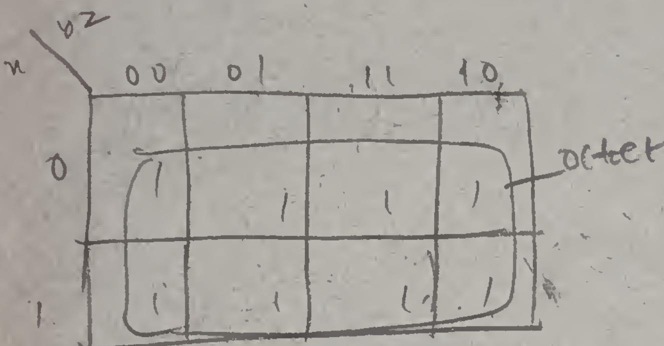


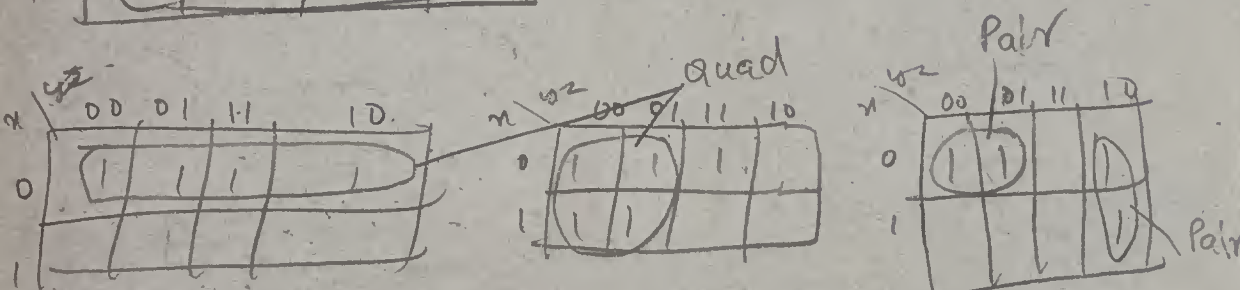
$\Rightarrow A \cdot \bar{B}$ (Pos form)

3 Variable K-map ($2^n = 2^3 = 8$)

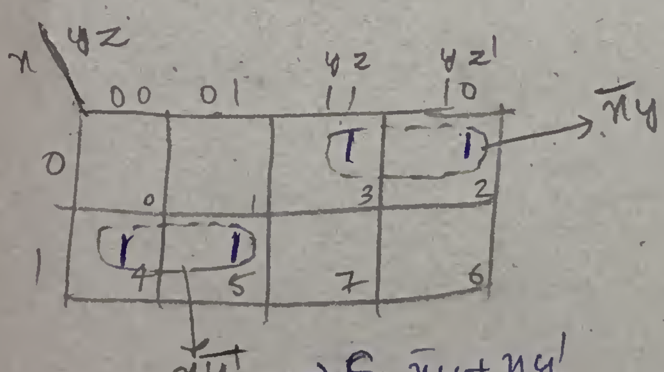
$x \backslash yz$	00	01	11	10
0	0	1	3	2
1	4	5	7	6

Grouping of
- 2 adjacent minterms — Pair
- 4 adjacent " — Quad
- 8 adjacent " — Octet





1) $F(x,y,z) = \Sigma(2,3,4,5)$



$\Rightarrow F = \bar{x}y + xy'$

2) $F(x,y,z) = \Sigma(3,4,6,7)$

$x \backslash yz$	00	01	11	10
0	0	1	1 3	2
1	1 4	5	1 7	1 6

$\rightarrow yz$

$\rightarrow x\bar{z}$

$xz' + yz$

3) $\Sigma(0,2,4,5,6)$

4) $\Sigma(1,2,4,7)$

6/5/21

1) solve using 3 variable k-map.

1) $F(x,y,z) = \Sigma(0,2,3,4,6)$

$x \backslash yz$	00	01	11	10
0	1 (0)	(1)	1 (3)	1 (2)
1	1 (4)	(5)	(7)	1 (6)

$\bar{z}$

$y\bar{x}$

$\Rightarrow \bar{x}y + \bar{z}$.

2) $F(x,y,z) = \Sigma(0,1,2,3,7)$

$x \backslash yz$	00	01	11	10
0	1 (0)	1 (1)	1 (3)	1 (2)
1	(4)	(5)	1 (7)	(6)

$\bar{x}$ yz

$\Rightarrow \bar{x} + yz$

3) $F(x,y,z) = \Sigma(3,5,6,7)$

$x \backslash yz$	00	01	11	10
0	(0)	(1)	1 (3)	(2)
1	(4)	1 (5)	1 (7)	1 (6)

yz

xy

xz.

$\Rightarrow xz + xy + yz$

4) $F(x,y,z) = \Sigma(0,1,6,7)$

$x \backslash yz$	00	01	11	10
0	1 (0)	1 (1)	(3)	(2)
1	(4)	(5)	1 (7)	1 (6)

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

$\Rightarrow \bar{x}\bar{y} + xy$.

5) $F(x,y,z) = \Sigma(1,2,\cancel{4},7)$

$x \backslash yz$	00	01	11	10
0	(0)	1 (1)	(3)	1 (2)
1	1 (4)	(5)	1 (7)	(6)

$\bar{x}y\bar{z}$

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

$\Rightarrow xy'z' + x'y'z + xyz + x'yz'$.

1) $F(x,y,z) = x'y' + yz + x'yz'$

(every literal should have 3 variables x, y & z)

Sol:-

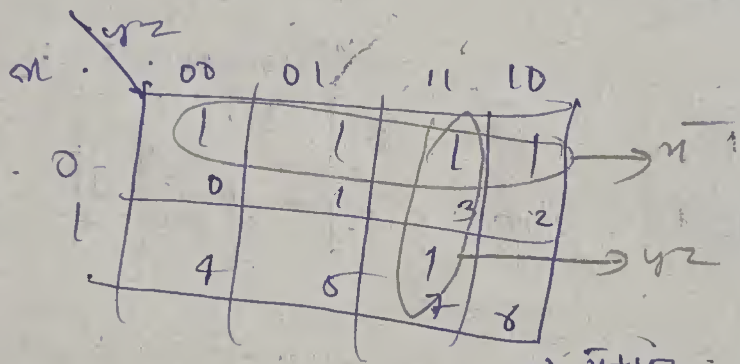
$x'y'(z+z') + (x+x')yz + x'yz'$

$\Rightarrow x'y'z + x'y'z' + xyz + x'yz + x'yz'$

001 000 111 011 101

$\Rightarrow m_1 \quad m_0 \quad m_7 \quad m_3 \quad m_2$

$\Rightarrow \Sigma m(0, 1, 2, 3, 7)$



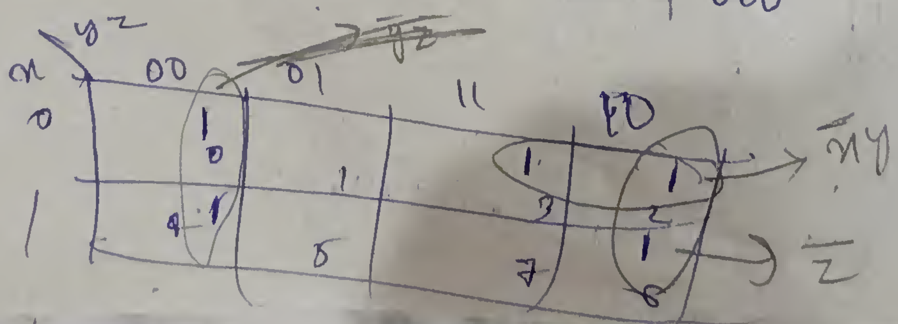
$\Rightarrow \bar{x} + yz$

2) $F(x,y,z) = x'y + yz' + y'z'$

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

$x'yz + x'yz' + xyz' + x'yz' + xy'z' + x'y'z'$.

011 010 110 010 100 , 000



$\Rightarrow \Sigma m(0, 2, 3, 4, 6)$

$\Rightarrow \bar{x}y + \bar{z}$

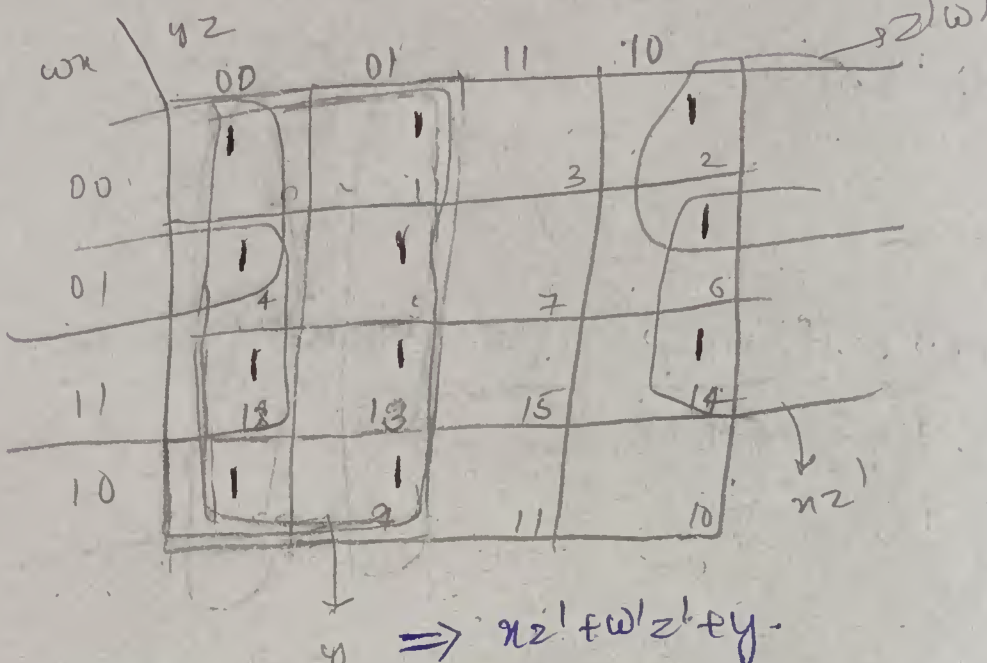
4-Variable K-map $(2^n = 2^4 = 16)$

(w, x, y, z)

wx \ yz	00	01	11	10
00	$w'x'y'z'$ m_0	$w'x'y'z$ m_1	$w'x'yz$ m_3	$w'x'yz'$ m_2
01	$w'xy'z'$ m_4	$w'xy'z$ m_5	$w'xyz$ m_7	$w'xyz'$ m_6
11	$wxy'z'$ m_{12}	$wxyz'$ m_{13}	$wxyz$ m_{15}	$wxyz'$ m_{14}
10	$wxyz'$ m_8	$wx'y'z$ m_9	$wx'yz$ m_{11}	$wx'yz'$ m_{10}

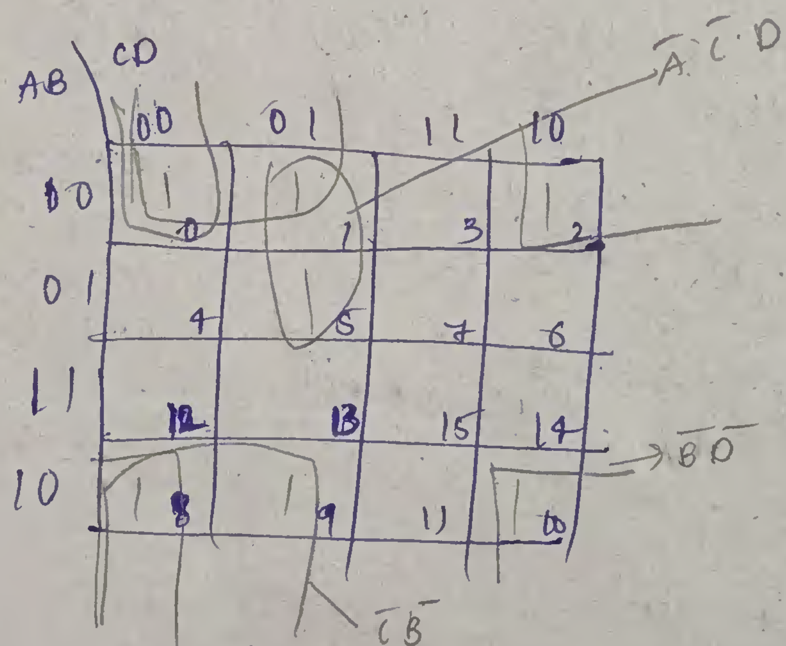
wx \ yz	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

1) $F(w,x,y,z) = \Sigma(0,1,2,4,5,6,8,9,12,13,14)$



$\Rightarrow xz' + w'z' + y$

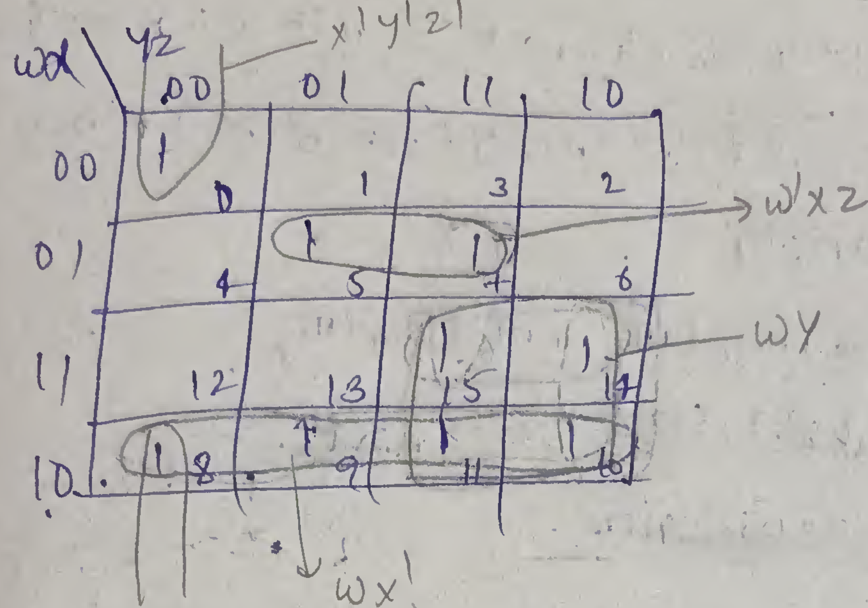
2) $F(A,B,C,D) = \Sigma(0,1,2,5,8,9,10)$



8 - octet
4 - Quad.

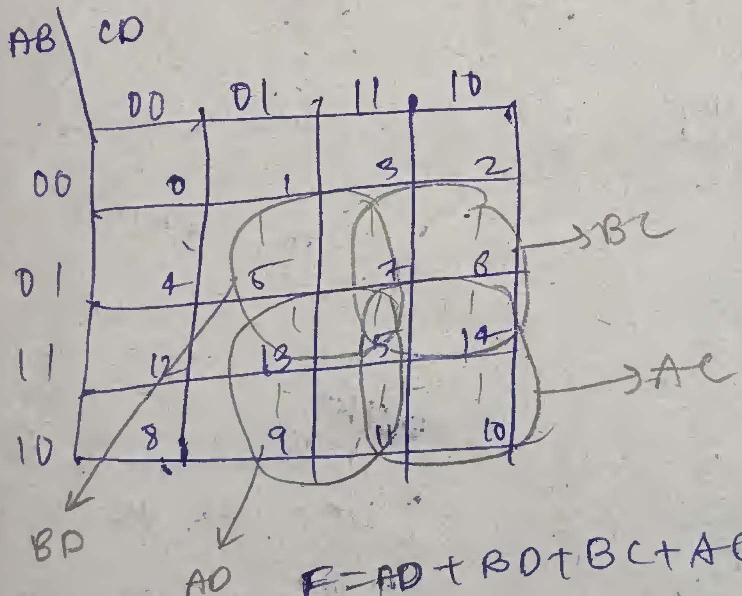
$\Rightarrow F = \bar{B}\bar{C} + \bar{B}\bar{D} + \bar{A}\bar{C}D$

3) $F(w,x,y,z) = \Sigma(0,5,7,8,9,10,11,14,15)$



$F = wx' + wY + w'xz + x'y'z'$

4) $F(A,B,C,D) = \Sigma m(5,6,7,9,10,11,13,14,15)$



$F = AD + BD + BC + AC$

5). $F = \underset{D}{A'B'C'} + \underset{A}{B'CD'} + A'BCD' + \underset{D}{AB'C'}$

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

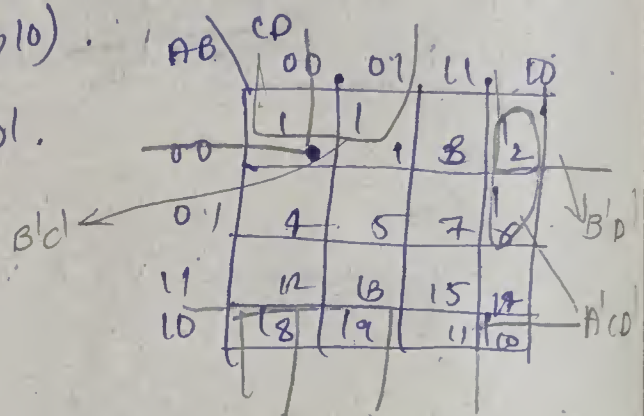
$\Rightarrow A'B'C'D + A'B'C'D' + AB'CD' + A'B'CD' + A'BCD' +$

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

$\Rightarrow m_1 + m_0 + m_{10} + m_2 + m_6 + m_9 + m_8$

$\Rightarrow \Sigma m(0,1,2,6,8,9,10)$.

$\Rightarrow F = B'C' + B'D' + A'CD'$.



5-Variable K-map ($2^n = 2^5 = 32$ cells/squares).

$\Rightarrow$ A,B,C,D,E

$m_0 = A'B'C'D'E'$

A = 0

BC \ DE	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

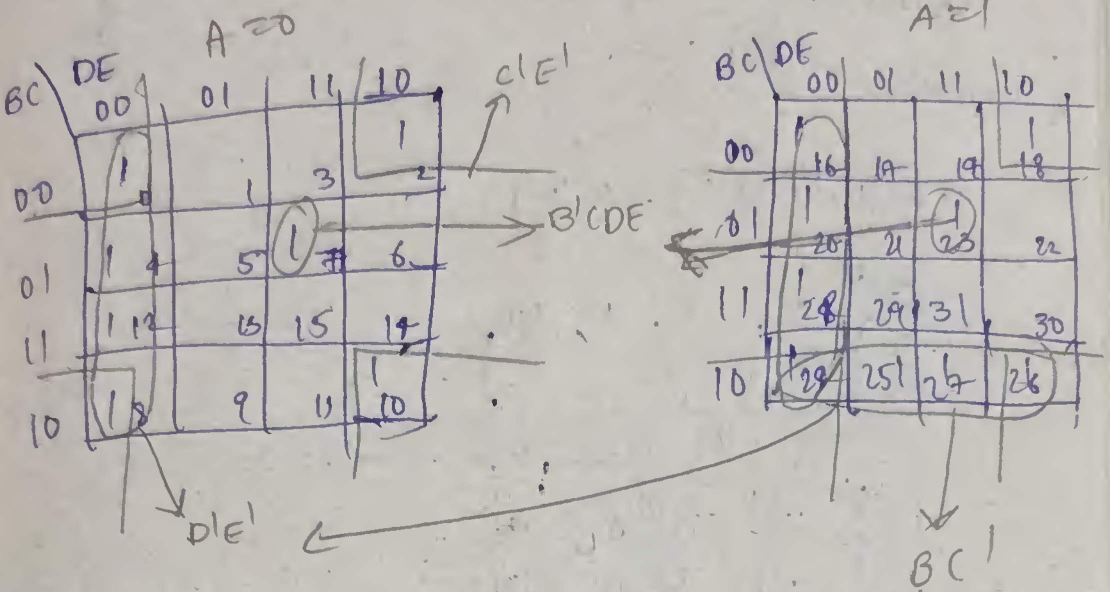
(0-15)

BC \ DE	00	01	11	10
00	m_{16}	m_{17}	m_{19}	m_{18}
01	m_{20}	m_{21}	m_{23}	m_{22}
11	m_{28}	m_{29}	m_{31}	m_{30}
10	m_{24}	m_{25}	m_{27}	m_{26}

(16-31)

32

1) $F(A,B,C,D,E) = \Sigma(0,2,4,7,8,10,12,16,18,20,23,24,25,26,27,28)$



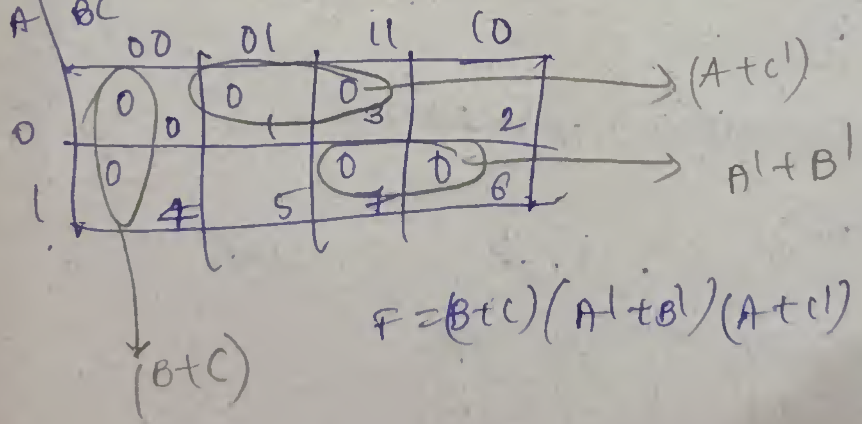
$\Rightarrow C'E' + D'E' + B'CDE + ABC'$.

Solving K-maps for POS :- 3 variable (K-map)

1) $F(A,B,C) = \underset{m_0}{(A+B+C)}\underset{m_1}{(A+B+\bar{C})}\underset{m_3}{(A+\bar{B}+\bar{C})}\underset{m_4}{(\bar{A}+B+C)}\underset{m_6}{(\bar{A}+\bar{B}+C)}$

$\underset{m_7}{(\bar{A}+\bar{B}+\bar{C})}$

$\Rightarrow \Pi m(0,1,3,4,6,7)$



$F = (B+C)(A'+B')(A+C')$

2) $F(A,B,C,D) = (A+B+C+D)(A+B+C+\bar{D})(A+B+\bar{C}+D)(A+\bar{B}+C+D)$
$(A+\bar{B}+C+\bar{D})(A+\bar{B}+\bar{C}+\bar{D})(\bar{A}+\bar{B}+\bar{C}+\bar{D})(\bar{A}+B+\bar{C}+D)$.

(4-Variable K-map)

$\Pi M(0,1,2,4,5,7,15,10)$

AB \ CD	00	01	11	10
00	0 (0)	0 (1)	3	0 (2)
01	0 (4)	0 (5)	0 (7)	6
11	12	13	0 (15)	14
10	8	9	11	0 (10)

$(A+C)$

$\rightarrow B+C'+D$

$\rightarrow B'+C'+D$

$\Rightarrow F(A,B,C,D) = (A+C)(B+C'+D)(B'+C'+D')$.

Don't care in K-maps

1) $F(A,B,C) = \Sigma m(2,3,4,5) + \Sigma d(6,7)$.

A \ BC	00	01	11	10
0	0	1	3	2
1	4	5	7	6

$\rightarrow B$

$\downarrow A$

$\Rightarrow A+B$

2) $F(A,B,C) = \Sigma m(0,2,6,7) + \Sigma d(1,3)$

A \ BC	00	01	11	10
0	0	1	3	2
1	4	5	7	6

$\bar{A} \leftarrow$

$\rightarrow B$

$\bar{A}+B$

3) $F(A,B,C,D) = \Sigma m(0,2,6,10,11,12,13) + \Sigma d(3,4,5,14,15)$

AB \ CD	00	01	11	10
00	1 (0)	(1)	X (3)	1 (2)
01	X (4)	X (5)	(7)	1 (6)
11	1 (12)	1 (13)	X (15)	X (14)
10	(8)	(9)	1 (11)	1 (10)

$\Rightarrow A'D'$

$\rightarrow AC$

BC'

$\Rightarrow BC' + AC + A'D'$.