By Assist Lecturer: Besma Nazar

Digital Systems

References:-

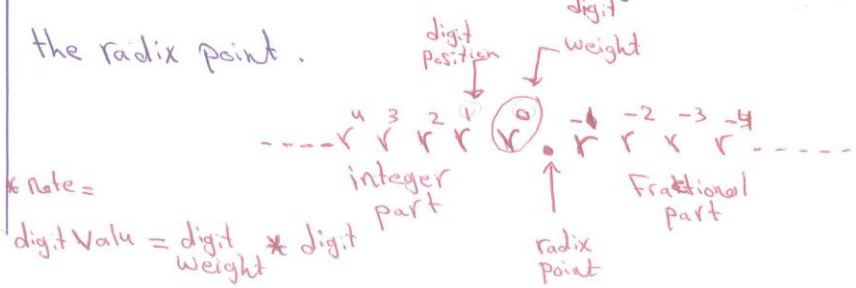
1- Digital fundamentals by Floyd.
2- Digital Design by Morris Mano.
3- Digital logic cct. analysis & design by Victor p. Nelson.
4- Digital Design principles & practice by John F. Wakerly.
5- Digital Electronics principles, Devices and Applications by Anil.

Asst. Lect. : Basma Nazar.

1- -: Number Systems :-

The study of number systems is important from the viewpoint of understanding how data are represented befor they can be processed by any digital system including a digital computer. A number system consists of an ordered set of symbols, called (digits) and the total number of digits allowed in the number system is called the (radix (r)) or (base) of the number system.

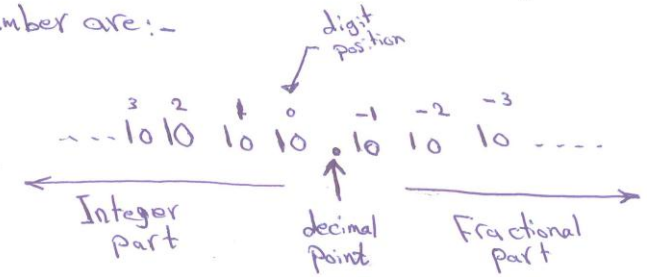
Number systems commonly used in digital system design and computer programming include decimal (r=10), binary (r=2), octal (r=8), and hexadecimal (r=16). Any number in a given system may have both an integer part and a fractional part, which are separated by a radix point (.). The place values or weights of different digits in the integer part of the number are given by r^0, r^1, r^2, r^3 and so on, starting with the digit adjacent to the radix point. For the fractional part, these are r^{-1}, r^{-2}, r^{-3} and so on, again starting with the digit next to the radix point.



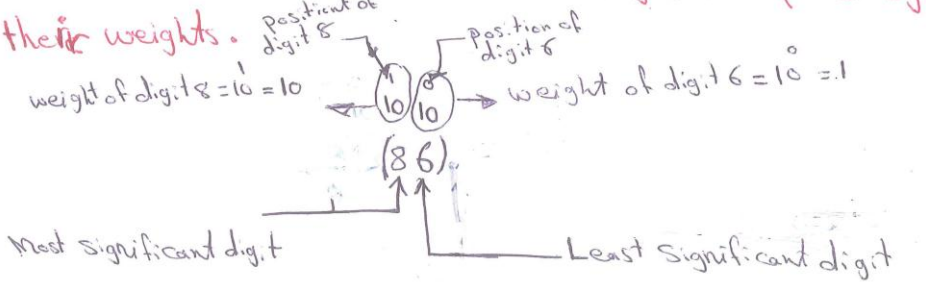
* Note =
digit Value = digit weight * digit

1- Decimal Number System: —

The decimal number system is radix (10) number system and therfor has 10 different digits or symbols. There are (0, 1, 2, 3, 4, 5, 6, 7, 8, 9). The weights of different digits in a mixed decimal number are:-



The value or magnitude of a given decimal number can be expressed as the Sum of the various digits multiplied by their weights.



Value of digit 8 = $8 * 10^1 = 8 * 10 = 80$
Value of digit 6 = $6 * 10^0 = 6 * 1 = 6$
] → 80 + 6 = 86

Ex1:- Express each of the following decimal numbers as a sum of the values of each digit.

a) $(47)_{10}$ b) $(568.23)_{10}$

Solution:-

a) 47

$$\begin{matrix} 1 & 0 \\ 10 & 10 \\ 4 & 7 \end{matrix} \quad 47 =$$

$$47 = (4*10^{1}) + (7*10^{0})$$
$$= (4*10) + (7*1)$$
$$= 40 + 7$$

b) 568.23

$$\begin{matrix} 2 & 1 & 0 & \text{decimal point} & -1 & -2 \\ 10 & 10 & 10 & \downarrow & 10 & 10 \\ 5 & 6 & 8 & . & 2 & 3 \end{matrix}$$

$$568.23 = (5*10^{2}) + (6*10^{1}) + (8*10^{0}) + (2*10^{-1}) + (3*10^{-2})$$
$$= (5*100) + (6*10) + (8*1) + (2*0.1) + (3*0.01)$$
$$= 500 + 60 + 8 + 0.2 + 0.03$$

Ex2:- What weight does the digit 7 have in each of the following numbers?

a) $(7051)_{10}$ b) $(58.72)_{10}$

Solution:-

$$\begin{matrix} 3 & 2 & 1 & 0 \\ 10 & 10 & 10 & 10 \\ 7 & 0 & 5 & 1 \end{matrix}$$

a) - the weight of 7 is $=10^{3}$
$=1000$

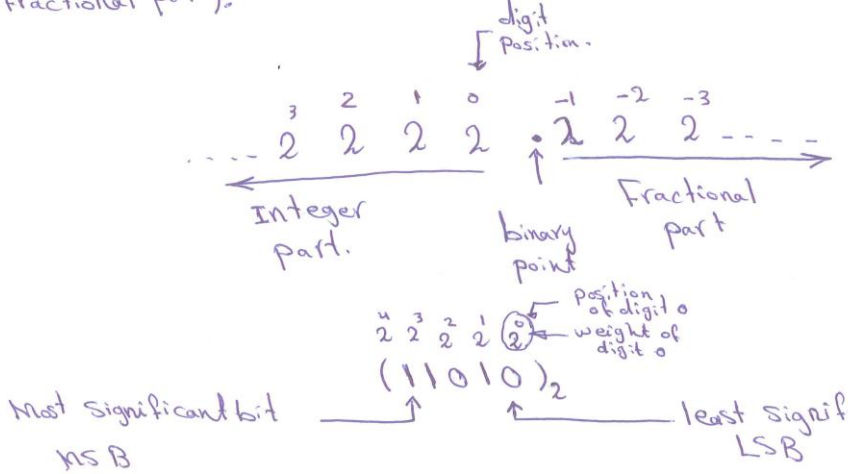
$$\begin{matrix} 1 & 0 & -1 & -2 \\ 10 & 10 & 10 & 10 \\ 5 & 8 & . & 7 & 2 \end{matrix}$$

b) the weight of 7 is 10^{-1}
$= 0.1$

Asst. Lect. : Besma Nazar

1-2) Binary Number System :-

The binary number system is a radix(2) number system with (0 and 1) as the two independent digits. A binary digit is called a bit. Starting from the binary point, the place values or weights of different bits in mixed binary number are 2^0, 2^1, 2^2 and so on (for the integer part) and 2^{-1}, 2^{-2}, 2^{-3} and so on (for the fractional part).



$$\text{Value} = (1*2^4) + (1*2^3) + (0*2^2) + (1*2^1) + (0*2^0)$$
$$= (1*16) + (1*8) + (0*4) + (1*2) + (0*1)$$

Note:- The largest decimal number that can be represented in binary with n bits equal to $2^n - 1$.

For example :- with five bits ($n=5$) the largest decimal number

$$2^5 - 1 = 31$$

and with six bits ($n=6$)

largest decimal number $= 2^6 - 1 = 63$

Ex3:- Determin the weight for the five first bits in binary system:-

Solution:-

b_5	b_4	b_3	b_2	bit1
4	3	2	1	0
2	2	2	2	2
↓	↓	↓	↓	↓
16	8	4	2	1

Ex4:- Determine the weight of the 1 in the binary number $(10000)_2$

Solution:-

$2^4\ 2^3\ 2^2\ 2^1\ 2^0$

1 0 0 0 0

The weight of 1 is $2^4 = 16$

Ex5:- Determine the weight of each bit that is a1 in the following fractional binary number $(0.1011)_2$.

Solution:-

$0.1\ 0\ 1\ 1$

$2^0 . 2^{-1}\ 2^{-2}\ 2^{-3}\ 2^{-4}$

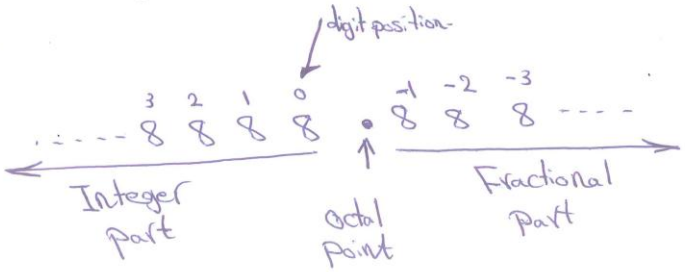
$2^{-4} = 0.0625$

$2^{-3} = 0.125$

$2^{-1} = 0.5$

(1.3) Octal Number System:-

The octel number system has a radix of (8) and therfore has eight distinct digits. The independent digits are (0, 1, 2, 3, 4, 5, 6, 7). The weights of different digits in amixed octal number are:-



8^1 (2) ← position of digit 7
8^0 (7) ← weight of digit 7

$(27)_8$

$27 = (2 * 8^1) + (7 * 8^0)$

Value $= (2 * 8) + (7 * 1)$

Note:- Coating in octal is similar to counting in decimal, except that the digits 8 and 9 are not used. To distinguish octal numbers from decimal numbers, we will use the subscript 8 to indicate an octal number For instance. $(15)_8$, $(7)_8$, $(22)_8$. Sometime you may see an "O" or a "Q" following an octal number.

(4) Hexadecimal Number System:-

The hexadecimal number system is radix(16) number system and its 16 basic digits are (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F). The first ten digits are borrowed from the decimal system. The letters A, B, C, D, E, F are used for digits 10, 11, 12, 13, 14, 15 respectively. The weights of different digits in a mixed hexadecimal number are

$\cdots\cdots 16^3\ 16^2\ 16^1\ 16^0 \;.\; 16^{-1}\ 16^{-2}\ 16^{-3} \cdots\cdots$

Integer Point ← | hexadecimal point | → Fractional Point

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

$(FD4.E)_{16}$

$= (F \times 16^2) + (D * 16^1) + (4 * 16^0) + (E * 16^{-1})$

$= (15 * 16^2) + (13 * 16^1) + (4 * 16^0) + (14 * 16^{-1})$

$= (15 * 256) + (13 * 16) + (4 * 1) + (14 * 0.0625)$

Note:- To avoid confusion hexadecimal numbers with decimal number or another numbers we will use the subscript 16 to indicide hexadecimal numbers. Sometimes you may see an "h" following a hexadecimal number.

H.w1:- What weight does the digit 0 have in each of the following numbers ?

a) $(51.03)_{10}$

b) $(11011)_2$

c) $(703.12)_8$

d) $(F2E.10A)_{16}$.

H.w2:- How many bits are required to represent the following decimal numbers ?

a) 17 b) 49 c) 114 d) 205

2 Conversion between Systems:-

1- Finding the Decimal Equivalent:-

The decimal equivalent of a given number in another number system is given by the sum of all the digits multiplied by their respective weights. The integer and fractional parts of the given number should be treated separately. Binary to decimal, octal to decimal and hexadecimal to decimal conversions are illustrated below:-

2.1 - Binary-to-Decimal Conversion:-

Ex6. Convert the following binary numbers to decimal:-

a) 11001 b) 110.001 c) 0.111

Solution:-

a) $(11001)_2 \longrightarrow (25)_{10}$

$2^4\ 2^3\ 2^2\ 2^1\ 2^0$

1 1 0 0 1

$(1*2^4)+(1*2^3)+(0*2^2)+(0*2^1)+(1*2^0)$

$(1*16)+(1*8)+(0*4)+(0*2)+(1*1)=(25)_{10}$

b) $(110.001)_2 \longrightarrow (6.125)_{10}$

$2^2\ 2^1\ 2^0\ 2^{-1}\ 2^{-2}\ 2^{-3}$

1 1 0 . 0 0 1

$(1*2^2)+(1*2^1)+(0*2^0)+(0*2^{-1})+(0*2^{-2})+(1*2^{-3})$

$(1*4)+(1*2)+(0*1)+(0*0.5)+(0*0.25)+(1*0.125)$

$=(6.125)_{10}$

c) $(0.111)_2=(0.875)_{10}$

$2^{-1}\ 2^{-2}\ 2^{-3}$

0 . 1 1 1

$(0*2^{-1})+(1*2^{-2})+(1*2^{-3})$

$=(0*1)+(1*0.5)+(1*0.25)+(1*0.125)=(0.875)_{10}$

2.2: Octal-to-Decimal Conversion:-

Ex.7:- Convert the following octal number to decimal:-

a) $(73)_8$ b) $(31.6)_8$

Solution:-

a) $(73)_8 \longrightarrow (59)_{10}$

8^1 8^0

7 3

$(7*8^1)+(3*8^0)$

$(7*8)+(3*1)=(59)_{10}$

b) $(31.6)_8 \longrightarrow (25.75)_{10}$

8^1 8^0 8^{-1}

3 1 . 6

$(3*8^1)+(1*8^0)+(6*8^{-1})$

$(3*8)+(1*1)+(6*0.125) = (25.75)_{10}$

2.3: Hexadecimal-to-Decimal Conversion:-

Ex.8:- Convert the following hexadecimal numbers to decimal:-

a) $(1C)_{16}$ b) $(F5.AB85)_{16}$

Solution:-

a) $(1C)_{16} \longrightarrow (28)_{10}$

16^1 16^0

1 C

$(1*16^1)+(C*16^0)$

$(1*16)+(12*1) = (28)_{10}$

b) $(F5.AB85)_{16} \longrightarrow (245.67)_{10}$

16^1 16^0 16^{-1} 16^{-2} 16^{-3} 16^{-4}

F 5 . A B 8 5

$(F*16^1)+(5*16^0)+(A*16^{-1})+(B*16^{-2})+(8*16^{-3})+(5*16^{-4})$

$(15*16)+(5*1)+(10*0.0625)+(11*0.00390)+(8*0.000244)+(5*0.0000153)$

$=(245.67)_{10}$

H.W:- Convert the following numbers to decimal:- (10)

a) $(1001.0101)_2$ b) $(137.21)_8$ c) $(1E0.2A)_{16}$

3-Decimal-to-Binary Conversion:-

There are two ways to convert from a decimal number to a binary number:- (3-1) Sum of weights methods
(3-2) Repeated division by 2 method.

(3-1) Sum of weights method:-

Exp:- Convert the following decimal numbers to binary:-

a) $(12)_{10}$ b) $(0.625)_{10}$ c) $(1.25)_{10}$

Solution:-

a) $(12)_{10} \rightarrow (1100)_2$

4	3	2	1	0
2	2	2	2	2
16	8	4	2	1
	1	1	0	0

$(1*8)+(1*4)=(12)_{10}$

b) $(0.625)_{10} \rightarrow (0.101)_2$

0	.	-1	-2	-3
2	.	2	2	2
0	.	0.5	0.25	0.125
0	.	1	0	1

$(1*0.5)+(1*0.125)=(0.625)_{10}$

c) $(1.25)_{10} \rightarrow (1.01)_2$

0	.	-1	-2
2	.	2	2
1	.	0.5	0.25
1	.	0	1

$(1*1)+(1*0.25)=(1.25)_{10}$

3-2:- Repeated division by 2-method:-

Ex.10:- Convert the following decimal numbers to binary:-

a) $(12)_{10}$ c.) $(0.625)_{10}$ d.) $(1.25)_{10}$ b) $(41)_{10}$

e) $(123.61)_{10}$

Solution:- a)

12	2	→ 0 LSB
6	2	→ 0
3	2	1.5 → 0.5 * 2 = 1
1	2	0.5 → 0.5 * 2 = 1 MSB
0		

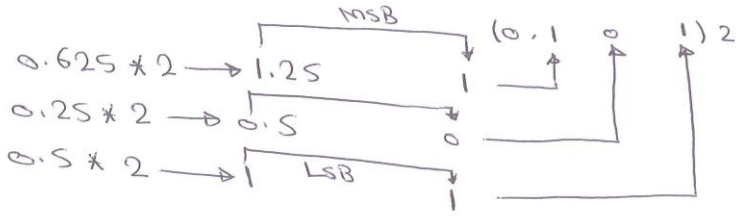
$(1\ 1\ 0\ 0)_2$

b) $(41)_{10} \rightarrow (10\ 1001)_2$

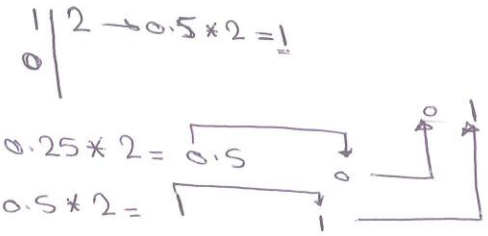
41	2	→ 20.5 → 0.5 * 2 = 1 LSB
20	2	→ 10 → 0 * 2 = 0
10	2	→ 5 → 0 * 2 = 0
5	2	→ 2.5 → 0.5 * 2 = 1
2	2	→ 1 → 0 * 2 = 0
1	2	→ 0.5 → 0.5 * 2 = 1 MSB
0		

$(1\ 0\ 1\ 0\ 0\ 1)_2$

c) $(0.625)_{10} \rightarrow (0.101)_2$

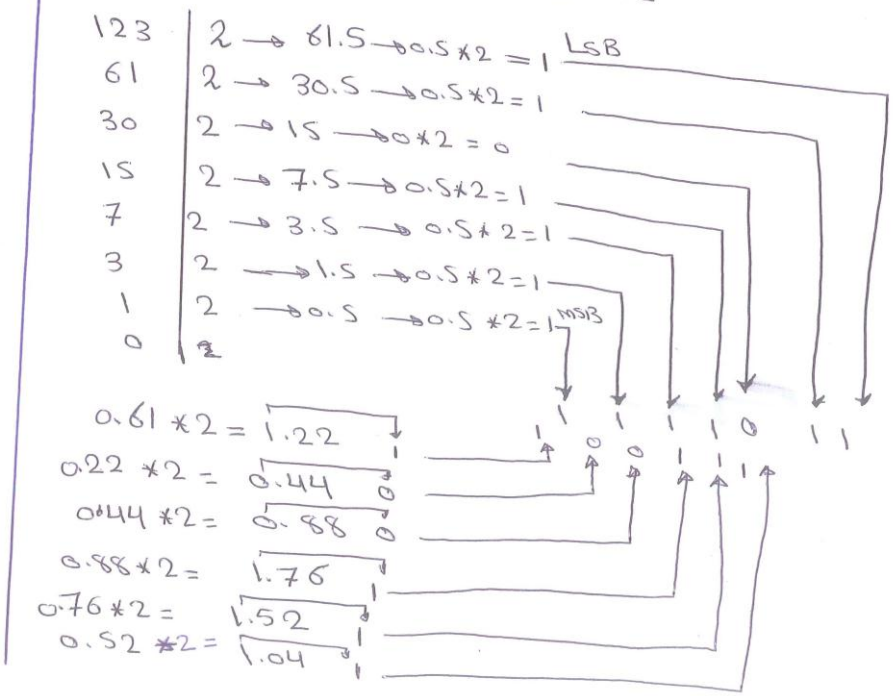


d) $(1.25)_{10} \rightarrow (1.01)_2$



note

e) $(123.61)_{10} \rightarrow (1111011.100111)_2$



4- Decimal-to-Octal Conversion:-

There are two way to convert from decimal to octal:-

(4-1) Sum of weights methods
(4-2) Repeated Division by 8.

4-1:- Sum of weights method:-

Ex11 :- Convert the following decimal numbers to octal:-

a) 14 b) 12.25

Solution:-

a) $(14)_{10} \longrightarrow (16)_8$

8^1 8^0
8 8
8 1
1 6

$(1*8)+(6*1)=(14)_{10}$

b) $(12.25)_{10} \longrightarrow (14.2)_8$

8^2 8^1 8^0 . 8^{-1} 8^{-2}
8 8 8 . 8 8
~~8~~4 8 1 . 0.125
1 4 . 2

$(1*8)+(4*1)+(2*0.125)=(12.25)_{10}$

4.2:- Repeated Division by 8:-

Ex.12: Convert the following decimal number to octal:-

a) 14 b) 12.25

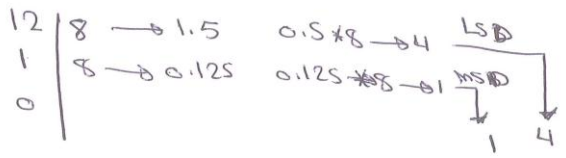
Solution:-

14	8 → 1.75	→ 0.75*8 = 6
1	8 → 0.125	→ 0.125*8 = 1
0		

LSB ↓ 6
MSB ↓ 1

$(1\ 6)_8$

b) $(12.25)_{10} \rightarrow (14.2)_8$



$0.25 * 8 = 2$

5- Decimal to Hexadecimal Conversion:-

There are two ways to convert from decimal to hexadecimal:

(5-1) Sum of weights method.

(5-2) Repeated Division by 16.

(5-1) Sum of weights method:-

Ex13:- Convert the following decimal numbers to hexadecimal:

a) $(650)_{10}$ b) $(213.0625)_{10}$

Solution:-

a) $(650)_{10} \rightarrow (28A)_{16}$

$$\begin{array}{ccc} 2 & 1 & 0 \\ 16 & 16 & 16 \\ 256 & 16 & 1 \\ (2 & 8 & A)_{16} \end{array}$$

$(2*256)+(8*16)+(A*1)$

$(2*256)+(8*16)+(10*1) = (650)_{10}$

b) $(213.0625)_{10} \rightarrow (D5.1)_{16}$

$$\begin{array}{cccc} 1 & 0 & . & -1 \\ 16 & 16 & . & 16 \\ 16 & 1 & . & 0.0625 \\ (D & 5 & . & 1)_{16} \end{array}$$

$(16*D)+(5*1)+(1*0.0625)$

$(16*13)+(5*1)+(1*0.0625) = (213.0625)_{10}$

(5-2) Repeated Division by 16 :-

(Ex.14) Convert the following decimal numbers to hexadecimal :-

a) $(650)_{10}$ b) $(245.67)_{10}$

Solution :-

a) $(650)_{10} \rightarrow (28A)_{16}$

650 | 16 → 40.625 → 0.625 × 16 → A
40 | 16 → 2.5 → 0.5 × 16 → 8
2 | 16 → 0.125 → 0.125 × 16 → 2
0

2 8 A

b) $(245.67)_{10} \rightarrow (F5.AB85)_{16}$

245 | 16 → 15.3125 → 0.3125 × 16 → F
15 | 16 → 0.9375 → 0.9375 × 16 → 5
0

5 F

0.67 × 16 → 10.72 → A
0.72 × 16 → 11.52 → B
0.52 × 16 → 8.32 → 8
0.32 × 16 → 5.12 → 5

A B 8 5

H.W :- Convert each of the following decimal numbers to binary, octal and hexadecimal numbers by using (a) sum of weight method and (b) Repeated of division

(a) 27 (b) 915 (c) 0.375
(d) 0.65 (e) 174.25 (f) 250.8

6- Binary - Octal and Octal - Binary Conversions :-

An octal number can be converted into binary equivalent by replacing each octal digit with its three-bit binary equivalent because the ($8 = 2^3$). A binary number can be converted into an equivalent octal number by splitting the integer and fractional parts into groups of three bits, starting from binary point on both sides. The 0s can be added to complete the outside groups if needed.

Ex 15 :- Convert each of the following Octal numbers to Binary

a) $(13)_8$ (b) $(7526)_8$ (c) $(1.52)_8$

Solution:-

(a) $(13)_8 \longrightarrow (1\,011)_2$

13

001 011

(b) $(7526)_8 \longrightarrow (111\,101\,010\,110)_2$

7 5 2 6

111 101 010 110

(c) $(1.52)_8 \longrightarrow (1.101\,01)_2$

1.52

001 . 101 010

Table (1)
Octal / Binary Conversion

Octal Digit	Binary (2^2 2^1 2^0)
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

Ex.16:- Convert each of the following binary number to Octal

a)$(110101)_2$ b)$(1101000100)_2$ c)$(1110100.0100111)_2$

d)$(110101.1)_2$ e) $(10101.001)_2$

Solution:— a)$(110101)_2 \longrightarrow (65)_8$

110 101
6 5

b)$(1101000100)_2 \longrightarrow (3204)_8$

011 010 000 100
3 2 0 4

c)$(1110100.0100111)_2 \longrightarrow (164.234)_8$

001 110 100 . 010 011 100
1 6 4 2 3 4

d)$(110101.1)_2 \longrightarrow (65.1)_8$

110 101 . 100
6 5 . 1

e)$(10101.001)_2 \longrightarrow (25.1)_8$

010 101 . 001
2 5 . 1

7- Hex - Binary and Binary - Hex Conversions:-

A hexadecimal number can be converted into its binary equivalent by replacing each hex digit with its four-bit binary equivalent because the $(16 = 2^4)$. A binary number can be converted into an equivalent hexadecimal number by splitting the integr and fractional parts into groups of four bits, starting from the binary point on both sides. The 0s can be added to complete the outside groups if needed.

Ex.17:- Convert each of the following hexadecimal numbers to binary:-

a) $(10A4)_{16}$ b) $(CF8E)_{16}$ c) $(2B.A)_{16}$

Solution:-

a) $(10A4)_{16} \rightarrow (0001\ 0000\ 1010\ 0100)_2$

1 0 A 4

0001 0000 1010 0100

b) $(CF8E)_{16} \rightarrow (1100\ 1111\ 1000\ 1110)_2$

C F 8 E

1100 1111 1000 1110

Table 2: Hex/binary conv.

Hex digit	Binary Digit (2 2 2 2)
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
A	1 0 1 0
B	1 0 1 1
C	1 1 0 0
D	1 1 0 1
E	1 1 1 0
F	1 1 1 1

c) $(2B.A)_{16} \rightarrow (101011.101)_2$

2 B . A

0010 1011 . 1010

Ex.18:- Convert the following binary numbers to hexadecimal:-

a) $(1100\ 1010\ 0101\ 0111)_2$ b) $(1111100010110001)_2$

c) $(1011\ 0001\ 1011.1111\ 0010)_2$

Solution:- a) $(1100\ 1010\ 0101\ 0111)_2 \rightarrow (CA57)_{16}$

b) $(1111100010110001)_2 \rightarrow (3F169)_{16}$

c) $(0010\ 1100\ 0110\ 1011.1111\ 0010)_2 \rightarrow (2C6B.F2)_{16}$

8- Hex-octal and Octal to Hex Conversions:- (19)

For hexadecimal-octal conversion, the given hex number is firstly converted into its binary equivalent which is further converted into its octal equivalent.

Hex number → Binary number → octal number

For octal-hexadecimal conversion, the octal number may first be converted into equivalent binary number and then the binary number transformed into its hex equivalent.

octal number → Binary number → Hex number

Ex.19:- Find the octal equivalent of $(2F.C4)_{16}$ and the hex equivalent of $(762.013)_8$.

Solution:- $(2F.C4)_{16} \rightarrow (0010\ 1111.1100\ 0100)_2 \rightarrow (57.61)_8$

$(762.013)_8 \rightarrow (111\ 110\ 010.000\ 001\ 011) \rightarrow (1F2.058)_{16}$

(H.W)(1):- Convert the following binary numbers to octal, hexadecimal and decimal numbers:-

a) $(1101)_2$ b) $(0.101)_2$ c) $(0.01101)_2$ d) $(10101.11)_2$

(H.W)(2):- Convert the following octal numbers to binary, hexadecimal and decimal numbers:-

a) $(65)_8$ b) $(240.51)_8$ c) $(2000)_8$ d) $(177777)_8$

(H.W)(3):- Convert each of the following hexadecimal numbers to binary, octal and decimal numbers:-

a) $(4F)_{16}$ b) $(F8.A7)_{16}$ c) $(2000)_{16}$ d) $(20b4)_{16}$

Arithmatic operation: → Addition, → Subtraction

1- Addition:-

ⓐ Binary Addition:-

The basic rules of binary addition are as follows:-

① $0+0=0$
② $0+1=1$
③ $1+0=1$
④ $1+1=0$ with a carry of 1 to the next most significant bit.
⑤ $1+1+1=1$ with a carry of 1 to the next most significant bit.

Ex:- Add the following binary numbers:-

a) $11+11$ b) $100+10$ c) $111+11$ d) $110+100$

Solution:- a) $11+11=(110)_2$

$$\begin{array}{r} {\scriptstyle 1\ \ } \\ 11 \\ 11 \\ \hline (110)_2 \end{array} +$$

b) $100+10=(110)_2$

$$\begin{array}{r} 100 \\ 010 \\ \hline (110)_2 \end{array} +$$

c) $111+11=(1010)_2$

$$\begin{array}{r} {\scriptstyle 1\ 1\ \ } \\ 111 \\ 011 \\ \hline (1010)_2 \end{array}$$

d) $110+100=(1010)_2$

$$\begin{array}{r} 110 \\ 100 \\ \hline (1010)_2 \end{array}$$

(b) Octal Addition:- 1) If the sum result ≥ 8, subtract 8 and carry 1

Ex:- Add the following numbers:-

a) $(57)_8 + (432)_8$ b) $(4163)_8 + (7520)_8$

Solution:- a) $(57)_8 + (432)_8 = (511)_8$

$$\begin{array}{r} {\scriptstyle ①\ ①} \\ 057 \\ 432 \\ \hline (511)_8 \end{array} +$$

b) $(4163)_8 + (7520)_8 = (13703)_8$

$$\begin{array}{r} {\scriptstyle ①} \\ 4163 \\ 7520 \\ \hline (13703)_8 \end{array} +$$

(c) Hexadecimal Addition:- 1) A=10, B=11, C=12, D=13, E=14, F=15
2) If the sum result ≥ 16, Subtract 16 and Carry 1

Ex:- Add the following numbers:-

a) $(58)_{16} + (4B)_{16}$ b) $(2A58)_{16} + (71D0)_{16}$

Solution:- a) $(58)_{16} + (4B)_{16} = (A3)_{16}$

$$\begin{array}{r} {\scriptstyle ①} \\ 58 \\ 4B \\ \hline (A3)_{16} \end{array} +$$

b) $(2A58)_{16} + (71D0)_{16} = (9C28)_{16}$

$$\begin{array}{r} {\scriptstyle ①} \\ (2A58) \\ (71D0) \\ \hline (9C28)_{16} \end{array} +$$

(H.w):- Add the following numbers:-

a) $(110101)_2 + (11001)_2$

b) $(175214)_8 + (152405)_8$

c) $(4F1A5)_{16} + (B85D5)_{16}$

- Complements:—

Complement are used in digital computer for simplifying the subtraction operation and for logical manipulation. There are two types of complements for each base-r system: The first is referred to as the r's complement and the second as the (r-1)'s complement.

1- Binary Number Complements:-

In the binary number system we have the 1's and 2's complements.

The 1's complement of a binary number is obtained by replacing 0s with 1s and 1s with 0s.

$0 \longrightarrow 1$
$1 \longrightarrow 0$

For example, the 1's complement of $(10010110)_2$ is $(01101001)_2$

The 2's complement of a binary number is obtained by adding '1' to it's 1's complement. or right to left copy bits through 1st logic 1, then invert.

2's Complement = 1's Complement + 1

For example, the 2's complement of $(10010110)_2$ is $(01101010)_2$

Ex:— Find the 1's complement of the following numbers:-

a) $(1011000)_2$ b) $(0101101)_2$

Solution:- a) 1011000
↓↓↓↓↓↓↓ 1's complement
$(0100111)_2$

b) 0101101
↓↓↓↓↓↓↓ 1st complement
$(1010010)_2$

Ex:- Find the 2's complement of the following numbers:-

a) $(10110101100)_2$ b) $(1101101001)_2$

Solution:- a) $(10110101100)_2$

First method 2's = 1's + 1

1st complement = 01001010011 (carries ① ①)

2's complement = $(01001010100)_2$ (1+)

Second method:- 1's complement | unchange

10110101100

01001010100

b) $(1101101001)_2$

First method = 2's = 1's + 1

1's Complement = 0010010110

2's complement = (0010010111) (1+)

Second method:- 1's complement | unchange

1101101001

$(0010010111)_2$

2- Decimal numbers Complements:-

In the Decimal number system we have the 9's and 10's Complements.

The 9's complement of a given decimal number is obtained by subtracting each digit from 9.

For example:- Find the 9's complement of $(2496)_{10}$

9999

2496 −

9's complement $(7503)_{10}$

The 10's Complement is obtained by adding '1' to the 9's complement.

10's complement = 9's complement + 1

For example: Find the 10's complement of $(2496)_{10}$

```
    9999
    2496
   ------
9's complement → 7503
                    1  +
                ------
10's complement (7504)10
```

Ex:- Find the 9's and 10's complement of the following numbers:-

a) $(546700)_{10}$ b) $(012398)_{10}$

Solution:- a) $(546700)_{10}$

```
    999999
    546700 -
   --------
9's comp.  (453299)10
                  1 +
              ------
10's comp. (453300)10
```

b) $(012398)_{10}$

```
    999999
    012398 -
   --------
9's comp.  (987601)10
                  1 +
              ------
10's comp. (987602)10
```

3- Octal number Complements:-

In the octal system, we have the 7's and 8's complements.

The 7's complement of a given octal number is obtaind by subtracting each octal digit from 7.

For example: the 7's complement of $(562)_8$ would be:-

```
    777
    562 -
   -------
7's complement (215)8
```

The 8's complement is obtained by adding '1' to the 7's complement.

8's complement = 7's complement + 1

For example: the 8's complement of $(562)_8$ would by:-

```
    777
    562 -
   -------
7's comp.  (215)8
                1 +
            -------
8's comp.  (216)8
```

Ex:- Find the 7's and 8's complement for the following number :- $(6572)_8$

Solution :-

	7 7 7 7
	6 5 7 2 -
7's complement	$(1205)_8$
	1 +
8's Complement.	$(1206)_8$

4- Hexadecimal Number Complements :-

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

The 15's and 16's complements are defined with respect to the hexa-decimal number system.

The 15's complement is obtained by subtracting each hex digit from 15. For example:- the 15's complement of $(3BF)_{16}$ would be

	15 15 15
	3 B F -
	$(C\ 4\ 0)_{16}$

The 16's complement is obtained by adding '1' to the 15's complement

16's complement = 15's complement +1

For example: the 16's complement of $(2AE)_{16}$ would be

	15 15 15
	2 A E -
15's Complement	$(D\ 5\ 1)_{16}$
	1 +
16's complement	$(D\ 5\ 2)_{16}$

Ex:- Find the 15's and 16's complement of the following number:- ABF8.

Solution:-

	15 15 15 15
	A B F 8 -
15's Complement	$(5\ 4\ 0\ 7)_{16}$
	1 +
16's complement	$(5\ 4\ 0\ 8)_{16}$

2- Subtraction:-

The direct method of subtraction thought in elementary schools uses the borrow concept. This method, works well when people perform subtraction with paper and pencil. However, when subtraction is implemented with digital H/w, the method is less efficient than the method that uses complements.

The subtraction of two n-digit numbers M-N can be done as follows:-

1- If we use $(r-1)$'s complement [1's, 9's, 7's, 15's]:-

a) For M-N, if $M \geq N$, the sum produce an end carry which can be added to the sum and this is referred to as an end around carry.

b) For M-N, if $M < N$, the sum does not produce an end carry and to obtain the answer in a familier form, take the $(r-1)$'s complement of the sum and place a negative sign infront.

2- If we use (r)'s complement [2's, 10's, 8's, 16's]:-

a) For M-N, if $M \geq N$, the sum produce an end carry which can be discarded

b) For M-N, if $M < N$, the sum does not produce an end carry and to obtain the answer in familier form, take the r's complement of the sum and place a negative sign infront.

Ex: Subtract the following binary number by use 1's and 2's Complement :-

a) $(1010100)_2 - (1000011)_2$ b) $(1000011)_2 - (1010100)_2$

Solution:-

a) $(1010100)_2 - (1000011)$

use 1's complement

```
  1010100
  1000011 -
  1010100
  0111100 +     (1's comp.)
 ---------
 10010000
cy └──────> 1 +
 ---------
```

result = $(0010001)_2$

use 2's complement

```
  1010100
  1000011 -
  1010100
  0111101 +     (2's comp.)
 ---------
 10010001
```

result = $(0010001)_2$

b) $(1000011)_2 - (1010100)_2$

use 1's complement

```
  1000011
  1010100 -
  1000011
  0101011 +     (1's comp.)
 ---------
  1101110
  0010001       (1's comp.)
 -(0010001)_2   add (-)
```

result = $-(0010001)_2$

use 2's complement

```
  1000011
  1010100 -
  1000011
  0101100 +     (2's comp.)
 ---------
  1101111
  0010001       (2's comp.)
 -(0010001)_2   add (-)
```

result = $-(0010001)_2$

Ex:- Using 9's and 10's complement subtracte:-

a) $(72532)_{10} - (3250)_{10}$ = b) $(3250)_{10} - (72532)_{10}$

Solution:—

a) $(72532)_{10} - (3250)_{10}$

9's complement

72532
03250 -
72532
9's comp. 96749 +
169281
cy → 1 +
result = $(69282)_{10}$

10's complement

72532
03250 -
72532
10's comp. 96750 +
cy 169282
result = $(69282)_{10}$

b) $(3250)_{10} - (72532)_{10}$

9's complement

03250
72532 -
03250
9's comp. 27467 +
30717
9's comp. 69282
add(-) $-(69282)_{10}$
result = $-(69282)_{10}$

10's complement

03250
72532 -
03250
10's comp. 27468 +
30718
10's comp. (69282)
add(-) $-(69282)_{10}$
result = $-(69282)_{10}$

Ex:- Subtract the following octal number by use 7's and 8's complement:-

a) $(256)_8 - (341)_8$

Solution:-

7's complement

256
341 -
256
7's comp. → 436 +
715
7's comp. → 063
add(-) → $-(63)_8$

result = $-(63)_8$

8's complement

256
341 -
256
437 +
795
8's comp. → 063
add(-) → $-(63)_8$

result = $-(63)_8$

Ex:- Subtract the following hexadecimal number by use 15's and 16's complement:-

a) $(592)_{16} - (3A5)_{16}$

Solution:-

15's complement

592
3A5 -
592
15's comp. → C5A +
1EC
→ 1 +
result = $(1ED)_{16}$

16's complement

592
3A5 -
592
C5B +
1ED

result = $(1ED)_{16}$

Ex:- Subtract the following number by use 1's and 2's complement:-

a) $(-3)_{10}+(4)_{10}$ b) $(-3)_{10}+(+4)_{10}$

Solution:-

a) $(-3)_{10}+(4)_{10}$

1's Complement

-3
4 +
binary signs
011
100
1's comp.
100
100 +
cy 1 000
1 +
result=$(001)_2$

2's Complement

-3
4 +
binary signs
011
100
2's comp.
101
100
cy ✗ 001
result=$(001)_2$

b) $(-3)_{10}+(-4)_{10}$

1's complement

-3
-4 +
signs
0011
0100
1100
1's comp.
1 0111
1 +
1000
1st comp = 0111
result = -(111)

2's Complement

-3
-4 +
0011
0100
①
1101
1100
1 1001
2's comp = 0111
result = -(111)

H.W:- Find the following:-

a) $(1001001)_2-(101110)_2$ using 1's & 2's complement.

b) $(341)_8-(256)_8$ using 7's & 8's complement.

c) $(9F1B)_{16}-(4A36)_{16}$ using 15's & 16's complement.

d) $(-18)_{10}+(-37)_{10}$ using 1's & 2's complement.

Binary Codes:-

Codes have been used for security reasons, so that others will not be able to read the message. There are many types of codes, each one of them have 4-bits, but the weight must be chosen in such a way that these sums is not greater than 15 and less than 9.

- BCD (8421) code:-

The binary coded decimal (BCD) is a type of binary code used to represent a given decimal number in an equivalent binary form. BCD-to-decimal and decimal-to-BCD conversions are very easy and straightforward. The BCD equivalent of a decimal number is written by replacing each decimal digit in integer and fractional parts with its four-bit binary equivalent. Table below gives the four bit code for one decimal digits:-

Decimal Symbol	BCD digit (8 4 2 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

Note:- ① It is important to realize that BCD numbers are decimal numbers and not binary numbers although they use bits in their representation.

② It is unable to decode binary numbers greater than 9. (The binary combination 1010 throught 1111 are not used and have no meaning in the BCD code) beause the decimal system do not have greater than 9.

Ex:- Convert each of the following decimal numbers to BCD code:

a) $(185)_{10}$ b) $(23.15)_{10}$ c) $(2469)_{10}$

Solution:- a) $(185)_{10}$

$(0001\ 1000\ 0101)_{BCD}$

b) $(23.15)_{10}$

$(0010\ 0011\ .0001\ 0101)_{BCD}$

c) $(2469)_{10}$

$(0010\ 0100\ 0110\ 1001)_{BCD}$

Ex:- Convert each of the following BCD codes to decimal:- *(H.w) and binary

a) $(0011\ 0101\ 0001)_{BCD}$ b) $(0010\ 1001.0111\ 0101)_{BCD}$

Solution:- a) $(0011\ 0101\ 0001)_{BCD} \rightarrow (351)_{10}$

(3 5 1)

b) $(0010\ 1001.0111\ 0101)_{BCD} \rightarrow (29.75)_{10}$

(2 9 . 7 5)

H.w:- Convert the following decimal number to BCD code and binary number.

a) 171.625 b) 171

-Other Decimal Codes:-

Many different codes can be formulated by arranging four bits into 16 distinct combination. The representative codes are shown in table below. Each code uses only 10 out of 16 possible 16-bit combinations that can be arranged with four bits. The other six unused combinations have no meaning and should be avoided.

Decimal digit	2421	84-2-1	7421	6311	5421	4221	Excess-3
0	0000	0000	0000	0000	0000	0000	0011
1	0001	0111	0001	0001	0001	0001	0100
2	0010	0110	0010	0010	0010	0010	0101
3	0011	0101	0011	0100	0011	0011	0110
4	0100	0100	0100	0110*	0100	1000	0111
5	1011	1011	0101	0111	1000	0111*	1000
6	1100	1010	0110	1000	1001	1100	1001
7	1101	1001	1000	1001	1010	1101	1010
8	1110	1000	1001	1011	1011	1110	1011
9	1111	1111	1010	1100	1100	1111	1100

Note:- (1) The BCD and the 2421 codes are weighted codes In weighted code, each position is assigned a weighting factor in such a way that each digit can be evaluated by adding the weights to all the 1's in the coded combination. The 84-2-1 code is an example of assigning both positive and negative weights to decimal code:-

$(6)_{10} = (0110)_{BCD} = (0*8)+(1*4)+(1*2)+(0*1)$

$(7)_{10} = (1101)_{2421} = (2*1)+(4*1)+(2*0)+(1*1)$

$(2)_{10} = (0110)_{84-2-1} = (8*0)+(4*1)+(-2*1)+(-1*0)$

(2) The Ex-3 code is a digital code obtained by adding three to each decimal digit and then converting the result to 4 bit binary. It is unweight code. It is example of self-complementing codes. The 9's complement is obtained directly by changing 1's → 0's and 0's → 1 in the code. $(395)_{10} \rightarrow (0110\ 1100\ 1000)_{Ex3}$

$(604)_{10} \rightarrow (1001\ 0011\ 0111)_{Ex3}$

Ex:- Convert the following decimal numbers:-

a) $(16)_{10} \rightarrow$ BCD code

b) $(7)_{10} \rightarrow$ 2421 code

c) $(395)_{10} \rightarrow$ Ex-3 code.

Solution:-

a) $(16)_{10} \rightarrow (0001\ 0110)_{BCD}$

b) $(7)_{10} \rightarrow (1101)_{2421}$

c) $(395)_{10} \rightarrow (0110\ 1100\ 1000)_{Ex-3}$

H.W:- Write the following decimal number in 4221 and 5421 codes ?

$(98.16)_{10}$

Gray Code:-

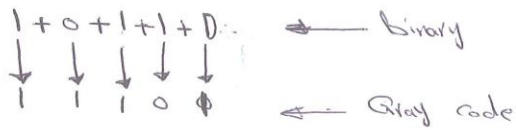
The Gray code is unweighted and is not an arithmetic code; that is, there are no specific weights assigned to the bit positions. The important feature of the Gray code is that is exhibits only a single bit change from one code word to the next in sequence. This property is important in many applications, such as shaft position encoders, where error susceptibility increases with the number of bits changes between adjacent numbers in a sequence. Like binary numbers, the Gray code can have any number of bits. In going from decimal 3 to decimal 4, the Gray code changes from 0010 to 0110 while the binary code change from 0011 to 0100, a change of three bits.

a) Binary to Gray code Conversion:-

The MSB in the Gray code is the same as the corresponding MSB in the binary number. Going from left to right, add each adjacent pair of binary code bit to get the next Gray code bit. Discard carries.

Ex:- Convert the binary number (10110) to Gray code:-

Solution:



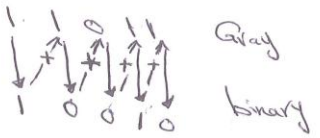
$(10110)_2 \longrightarrow (11101)_{\text{Gray code}}$.

b) Gray to Binary Conversion:-

The MSB in the binary code is the same as the corresponding bit in the Gray code. Add each binary bit generated to the Gray code bit in the next adjacent position. Discard carries.

Ex:- Convert the Gray code (11011) to binary:-

Solution:-



$(11011)_{\text{Gray}} \longrightarrow (10010)_2$

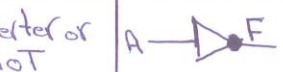
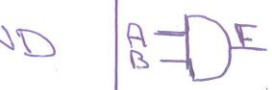
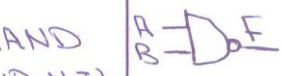
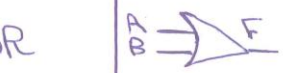
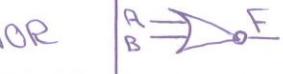
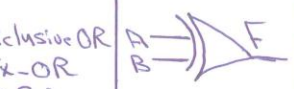
H.W(1): List the 4-bit Gray code for decimal numbers from 0 → 15.

H.W(2): (a) Convert binary 10110 to Gray code

(b) Convert Gray code 10011 to binary.

Logic Gates:-

Circuits used to process digital signals are called (logic gates). Logic symbols are used to identify these ccts. The seven gates that are the fundamental logic elements in digital system are shown below:-

Logic Function	Logic gate symbol	Truth Table	Boolean Expression
1- Inverter or NOT		input A | output F=Ā 0 | 1 1 | 0	$F = \bar{A}$
2- AND		AB | F 00 | 0 01 | 0 10 | 0 11 | 1	$F = A \cdot B$ (AND جبر)
3- NAND (AND-NOT)		AB | F 00 | 1 01 | 1 10 | 1 11 | 0	$F = \overline{A \cdot B}$
4- OR		AB | F 00 | 0 01 | 1 10 | 1 11 | 1	$F = A + B$ (OR جبر)
5- NOR (OR-NOT)		AB | F 00 | 1 01 | 0 10 | 0 11 | 0	$F = \overline{A + B}$
6- Exclusive OR Ex-OR (XOR)		AB | F 00 | 0 01 | 1 10 | 1 11 | 0	$F = A \oplus B$ (Ex-OR جبر) $= AB' + A'B$
7- Exclusive NOR Ex-NOR (Ex-OR-NOT) (XNOR)		AB | F 00 | 1 01 | 0 10 | 0 11 | 1	$F = A \odot B$ (Ex-NOR جبر) $= AB + A'B'$

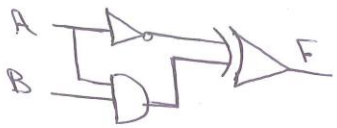
Note:- $A \oplus B = \overline{A \odot B}$

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

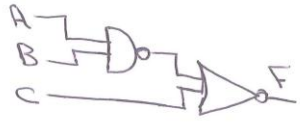
Ex:- Write the boolean algebra expression of the following cats:-

a) 

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

b) 

$F = \bar{A} \oplus (AB)$

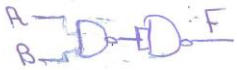
c) 

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

-Universal Gates:-

NAND and NOR gates have the property that they individually can be used to hardware-implement a logic circuit corresponding to any given Boolean expression. That is, it is possible to use either only NAND gates or only NOR gates to implement any Boolean expression

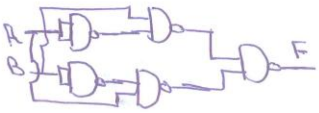
1- NOT gate:- $F = \bar{A}$ 

2- AND gate:- $F = \overline{\overline{AB}}$ 

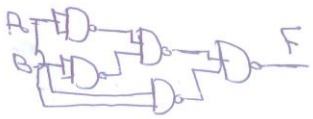
3- OR gate:- $F = \overline{\overline{A+B}} = \bar{\bar{A}} \cdot \bar{\bar{B}}$ 

4- NOR gate:- $F = \overline{\overline{A+B}} = \bar{\bar{A}} \cdot \bar{\bar{B}}$

5- XOR gate = $A \oplus B = \overline{\overline{A\bar{B} + \bar{A}B}}$

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

6- XNR-gate = $A \odot B = \overline{\overline{A\bar{B} + \bar{A}B}}$

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

Ex:- Implement the following Functions by using NAND gates only:-

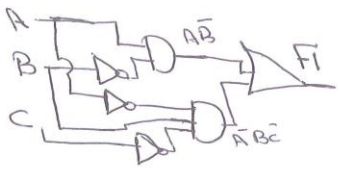
a) $F_1 = A\bar{B} + \bar{A}B\bar{C}$ b) $F_2 = AB + C\bar{D}$

Solution:-

a)

Mixed gates

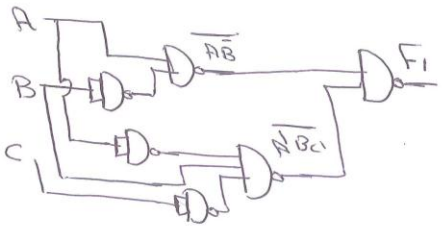
$F_1 = A\bar{B} + \bar{A}B\bar{C}$



NAND gates

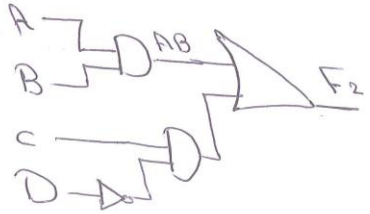
$F_1 = \overline{A\bar{B} + \bar{A}B\bar{C}}$

$F_1 = \overline{A\bar{B}} \cdot \overline{\bar{A}B\bar{C}}$



Mixed gates

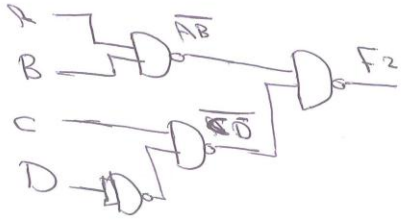
b) $F_2 = AB + C\bar{D}$



NAND gates

$F_2 = \overline{AB + C\bar{D}}$

$F_2 = \overline{AB} \cdot \overline{C\bar{D}}$



1- NOT gate $F_1 = \bar{A}$

2- AND gate $F_2 = AB$ $\overline{\bar{A} + \bar{B}}$

3- NAND gate $F_3 = \overline{AB} = \bar{A} + \bar{B}$

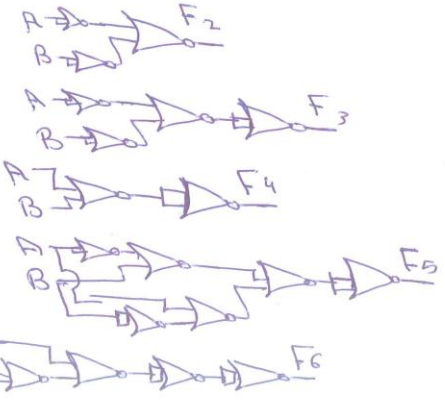
4- OR gate $F_4 = A + B$

5- EX-OR :- $F_5 = A\bar{B} + \bar{A}B$

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

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

6- EX-NOR = $\overline{F_5}$



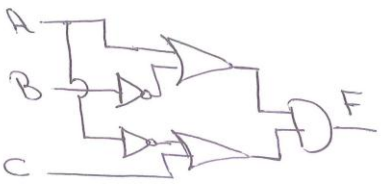
Ex:- Implement following function by using Mixed NOR gates only:-

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

Solution:-

Mixed gates

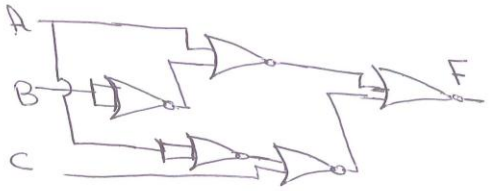
$F = (A+\bar{B}).(\bar{A}+C)$



NOR gates

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

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



Combinational logic circuits:-

Digital system are composed of combinations logic gates described by a truth table and boolean expression or logic symbol diagram. In Combination logic, the output of the cct. depends only on the inputs to the circuit.

-Demorgan's Theorem:-

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

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

Laws and Rules of boolean algebra:-

1- $A+B = B+A$
2- $AB = BA$
→ Commutative Laws

3- $A+(B+C) = (A+B)+C$
4- $A(BC) = (AB)C$
→ Associative Laws

5- $A(B+C) = AB+AC$ → Distributive Law

6- $A+0 = A$
7- $A+1 = 1$
8- $A+A = A$
9- $A+\bar{A} = 1$
10- $A \cdot 0 = 0$
11- $A \cdot 1 = A$
12- $A \cdot A = A$
13- $A \cdot \bar{A} = 0$
14- $\bar{\bar{A}} = A$
15- $A + AB = A$

Prove:- $A+AB$

$A(1+B) = A$

16- $A(A+B) = A$

Prove:- $A \cdot A + AB$

$A + AB$

$A(1+B) = A$

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

Prove:- Take the left side:-

$(A+B)(A+C) = AA + AC + AB + BC$
$= A + AC + AB + BC$
$= A(1+C) + AB + BC$
$= A + AB + BC$
$= A(1+B) + BC$
$= A + BC$

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

Prove:- $A+\bar{A}B$

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

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

Simplification Using Boolean Algebra:-

Many times in the application of Boolean algebra, you have to reduce a particular expression to its simplest form. A simplified Boolean expression uses the fewest gates possible to implement a given expression.

EX:- Using Boolean algebra techniques, simplify the following functions. expressions:

a) $AB+A(B+C)+B(B+C)$ b) $A\bar{B}+A\overline{(B+C)}+B\overline{(B+C)}$ c) $[A\bar{B}(C+BD)+\bar{A}\bar{B}]C$

Solution:- a) $AB+A(B+C)+B(B+C)$

$F = AB+AB+AC+BB+BC$

$F = AB+AC+\underline{B}+\underline{BC}$

$F = AB+AC+B(\underline{1+C})$

1

$F = \underline{AB}+AC+\underline{B}$

$F = B(\underline{A+1})+AC$

1

$F = B+AC$

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

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

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

0

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

c) $[A\bar{B}(C+BD)+\bar{A}\bar{B}]C$

$F = [A\bar{B}C+A\bar{B}\underline{BD}+\bar{A}\bar{B}]C$

0

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

$F = A\bar{B}\underline{CC}+\bar{A}\bar{B}C$

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

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

1

$F = \bar{B}C$

H.w :- Simplify the following Boolean expression:-

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

b) $ABCD + AB(\overline{C\bar{D}}) + \overline{(AB)}CD$

- Standerd form of Boolean expressions:-

All Boolean expressions, regardless of their form, can be converted in to either of two standard forms:-

1- The Sum of products (SOP)

2- The product of sum (POS)

Standardization makes the evaluation, simplification, and implementation of Boolean expressions much more systematic and easier.

1- The Sum of Products (SOP):-

SOP is a term consisting of the product of literals (variables) or their complements, when two or more product terms are summed by Boolean addition. A sum of products expression is also known as a minterm expression. It can be obtained from the truth table directly by considering those input combinations that produce a logic 1 at the output.

Ex:- $AB + \bar{A}B\bar{C} + AB \rightarrow$ SOP

Ex:- Convert the Boolean expression to SOP form:-

a) $F = (A+B)(B+C+D)$ b) $\overline{(A+B)+C}$

Solution:- a) $F = AB + AC + AD + BB + BC + BD$

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

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

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

Note:- ① Canical or Standard form of Boolean expressions is an Expanded form of Boolean expression, where each term contains all Boolean variables in their true or complemented form and is obtained by including all possible combinations of missing variables.

② In Sop to convert it to standard Sop form we AND it misses one or more variables with an expression such as $(X+\bar{X})$, where X is one of the missing variables.

Ex:- Convert the following Boolean expression in to standard Sop form:-

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

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

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

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

Ex:- put F in Sop form and simplify it using Boolean algebra and theorem and then implement this function:-

$0 \rightarrow \bar{X}$
$1 \rightarrow X$] ← in Sop

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

$F = \sum m_0, m_1, m_3$

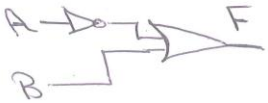
$F = \sum 0, 1, 3$

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

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

$F = \bar{A} + AB$

$F = \bar{A} + B$



Ex:- List the truth table of the function, and then implement it using NAND gate only:-

$F = AB + Bc + \bar{A}\bar{B}C$

Solution:- $F = AB + Bc + \bar{A}\bar{B}C$

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

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

$F = \underbrace{ABC}_{7} + \underbrace{AB\bar{C}}_{6} + \underbrace{\bar{A}BC}_{3} + \underbrace{\bar{A}\bar{B}C}_{1}$

ABC	F
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	0
1 1 0	1
1 1 1	1

$F = \Sigma 1, 3, 6, 7$

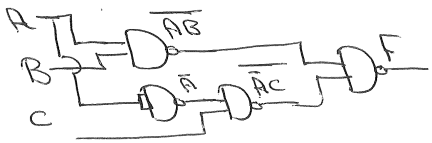
$F = ABC + AB\bar{C} + \bar{A}BC + \bar{A}\bar{B}C$

$F = AB[\underbrace{C + \bar{C}}_{1}] + \bar{A}C[\underbrace{B + \bar{B}}_{1}]$

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

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

$F = \overline{\overline{AB} \cdot \overline{\bar{A}C}}$



(H.W): Develop a truth table for the following sop expression:-

$F = \bar{X} + Y\bar{Z} + WZ + X\bar{Y}Z$

(H.W): Convert the following expressions to Standard sop forms:-

a) $AB(\bar{B}\bar{C} + BD)$ b) $A + B[AC + (B + \bar{C})D]$

2- The Product-of Sums (POS):-

POS is a term consisting of the sum (Boolean adding) of literal (variable) or their complements, when two or more sum are multiplied. A product of sum is also known as a maxterm expression. It can be obtained from the truth table by considering those input combinations that produce a logic 0 at the output.

Ex:- $F = (\bar{A}+B)(A+\bar{B}+C) \longrightarrow$ POS

Ex:- Convert the following Boolean expression to standard POS form:-

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

Note:- In POS to convert it to standard pos form we ORed it misses one or more variables with an expression such as $(X\bar{X})$, where is one of the missing variables

Solution:- $F = (A+\bar{B}+C+D\bar{D})(\bar{B}+C+\bar{D}+A\bar{A})(A\bar{B}+\bar{C}+D)$

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

Ex:- Represent F in pos form then simplify using theorem and implement it:-

	A	B	C	F	
M_0	0	0	0	0	$A+B+C$
M_1	0	0	1	1	
M_2	0	1	0	1	
M_3	0	1	1	1	
M_4	1	0	0	0	$\bar{A}+B+C$
M_5	1	0	1	1	
M_6	1	1	0	1	
M_7	1	1	1	1	

$0 \longrightarrow X$
$1 \longrightarrow \bar{X}$] in pos

$F = \Pi M_{0,}M_4$

$F = \Pi_{0,4}$

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

$F = A\bar{A} + AB + AC + \bar{A}B + B\bar{B} + BC + \bar{A}C + BC + CC$
$F = AB + AC + \bar{A}B + B + \bar{A}C + BC + C$
$F = B(A+\bar{A}+1+C) + C(A+\bar{A}+C)$
$F = B + C$

Note:- The complement of standard sop minterms equals the sum of minterms missing from the original function which are equal the standard pos maxterms.

$$\text{pos maxterm} = \overline{\text{sop minterm}}$$

$$\text{sop minterm} = \overline{\text{pos maxterm}}$$

Ex:- $F = \sum(1,4,5,6,7)$

$\bar{F} = \sum 0,2,3 = \prod 0,2,3$

Ex:- $F = \prod(0,2,4,5)$

$\bar{F} = \sum(1,3,6,7)$

Ex:- Convert the following sop expression to an equivelant pos expression:-

a) $F = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}C + ABC$

$F = \sum 0,2,3,5,7$

$F = \prod 1,4,6$

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

b) $F = w\bar{x}y + \bar{x}y\bar{z} + wx\bar{y}$

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

$F = w\bar{x}yz + w\bar{x}y\bar{z} + wx\bar{y}z + \bar{w}\bar{x}y\bar{z} + wx\bar{y}\bar{z}$

1011 (11) 1010 (10) 1010 (10) 0010 (2) 1101 (13) 1100 (12)

$F = \sum 2,10,11,12,13$

$F = \prod 1,3,4,5,6,7,8,9,14,15,0$

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

Karnaugh Maps (K-map):-

The K-map is simply another form of a truth table and is easiest method for simplifying logical expression and, if properly used, will produce the simplest SOP or POS expression possible. A logic function containing n variables requires 2^n rectangles.

Note:-

1- The main thing to remember is that only one variable can be change between adjacent rectangles when moving in a horizontal or a vertical direction. This is because that the minterms are not arranged in binary sequence but in sequence similar to Gray code and the characteristic of this sequence is that only one bit changes from 1 → 0 or from 0 → 1 in the listing sequence.

2- The logic expression obtained from K-map must be in the easiest form and cannot be simplified more.

Type of K-map:-

1- Two Variables:-

$F = (A, B)$

MSB ↑ ↑ LSB

No. of cells $= 2^n$

$= 2^2$

$= 4$

0 → 3

B \ A	$\bar{A}_0$	A_1
$\bar{B}_0$	$\bar{A}\bar{B}$ 0	$A\bar{B}$ 2
B_1	$\bar{A}B$ 1	AB 3

2- Three Variables:-

$F = (A, B, C)$

MSB ↑ ↑ LSB

No. of Celles $= 2^3$

$= 8$

0 → 7

C \ AB	$\bar{A}\bar{B}$ 00	$\bar{A}B$ 01	AB 11	$A\bar{B}$ 10
$\bar{C}$ 0	$\bar{A}\bar{B}\bar{C}$ 0	$\bar{A}B\bar{C}$ 2	$AB\bar{C}$ 6	$A\bar{B}\bar{C}$ 4
C 1	$\bar{A}\bar{B}C$ 1	$\bar{A}BC$ 3	ABC 7	$A\bar{B}C$ 5

3- Four Variables:-

$F = (A, B, C, D)$

MSB ↑ ↑ LSB

No. of Celles $= 2^4$

$= 16$

0 → 15

CD \ AB	$\bar{A}\bar{B}$ 00	$\bar{A}B$ 01	AB 11	$A\bar{B}$ 10
$\bar{C}\bar{D}$ 00	$\bar{A}\bar{B}\bar{C}\bar{D}$ 0	$\bar{A}B\bar{C}\bar{D}$ 4	$AB\bar{C}\bar{D}$ 12	$A\bar{B}\bar{C}\bar{D}$ 8
$\bar{C}D$ 01	$\bar{A}\bar{B}\bar{C}D$ 1	$\bar{A}B\bar{C}D$ 5	$AB\bar{C}D$ 13	$A\bar{B}\bar{C}D$ 9
CD 11	$\bar{A}\bar{B}CD$ 3	$\bar{A}BCD$ 7	$ABCD$ 15	$A\bar{B}CD$ 11
$C\bar{D}$ 10	$\bar{A}\bar{B}C\bar{D}$ 2	$\bar{A}BC\bar{D}$ 6	$ABC\bar{D}$ 14	$A\bar{B}C\bar{D}$ 10

4- Five Variables:-

$F = (A, B, C, D, E)$

MSB ↑ ↑ LSB

No. of celles $= 2^5$

$= 32$

0 → 31

DE \ ABC	$\bar{A}\bar{B}\bar{C}$ 000	$\bar{A}\bar{B}C$ 001	$\bar{A}BC$ 011	$\bar{A}B\bar{C}$ 010	$AB\bar{C}$ 110	ABC 111	$A\bar{B}C$ 101	$A\bar{B}\bar{C}$ 100
$\bar{D}\bar{E}$ 00	0	4	12	8	24	28	20	16
$\bar{D}E$ 01	1	5	13	9	25	29	21	17
DE 11	3	7	15	11	27	31	23	19
$D\bar{E}$ 10	2	6	14	10	26	30	22	18

5- Six Variables:-

$F = (A, B, C, D, E, F)$

MSB ↑ ↑ LSB

No. of celles $= 2^6$

$= 64$

0 → 63

DEF \ ABC	$\bar{A}\bar{B}\bar{C}$ 000	$\bar{A}\bar{B}C$ 001	$\bar{A}BC$ 011	$\bar{A}B\bar{C}$ 010	$AB\bar{C}$ 110	ABC 111	$A\bar{B}C$ 101	$A\bar{B}\bar{C}$ 100
$\bar{D}\bar{E}\bar{F}$ 000	0	8	24	16	48	56	40	32
$\bar{D}\bar{E}F$ 001	1	9	25	17	49	57	41	33
$\bar{D}EF$ 011	3	11	27	19	51	59	43	35
$\bar{D}E\bar{F}$ 010	2	10	26	18	50	58	42	34
$DE\bar{F}$ 110	6	14	30	22	54	62	46	38
DEF 111	7	15	31	23	55	63	47	39
$D\bar{E}F$ 101	5	13	29	21	53	61	45	37
$D\bar{E}\bar{F}$ 100	4	12	28	20	52	60	44	36

Ex:- Find the Boolean expression for F by using K-map?

AB	F
00	1
01	1
10	1
11	0

$F = \Sigma 0,1,2$

B \ A	$\bar{A}$	A
$\bar{B}$	1 (0)	1 (2)
B	1 (1)	0 (3)

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

Ex:- Simplify F using K-map?

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

Solution:- $F = \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D} + \bar{A}CD(B+\bar{B}) + A\bar{B}C\bar{D}$

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

0 2 8 7 3 10

$F = \Sigma 0, 2, 3, 7, 8, 10$

CD \ AB	$\bar{A}\bar{B}$ 00	$\bar{A}B$ 01	AB 11	$A\bar{B}$ 10
$\bar{C}\bar{D}$ 00	1 (0)	0 (4)	0 (12)	1 (8)
$\bar{C}D$ 01	0 (1)	0 (5)	0 (13)	0 (9)
CD 11	1 (3)	1 (7)	0 (15)	0 (11)
$C\bar{D}$ 10	1 (2)	0 (6)	0 (14)	1 (10)

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

Ex:- Simplify F using K-map:-

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

7 0 1 2 3

$F = \Sigma 0, 1, 2, 3, 7$

C \ AB	$\bar{A}\bar{B}$ 00	$\bar{A}B$ 01	AB 11	$A\bar{B}$ 10
$\bar{C}$ 0	1 (0)	1 (2)	0 (6)	0 (4)
C 1	1 (1)	1 (3)	1 (7)	0 (5)

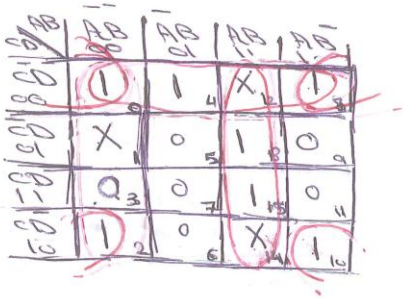
$F = \bar{A} + BC$

Don't care Conditions:-

The presence of don't cares in the design specifications gives us flexibility in selecting the output function that is to be synth-esized. When simplifying logic expressions, the Don't care terms are represented by X and may be used as 1's or 0's.

Ex:- Simplify:- $F = \sum 0,2,4,8,10,13,15 + \sum_d 1,5,12,14$, using k-map

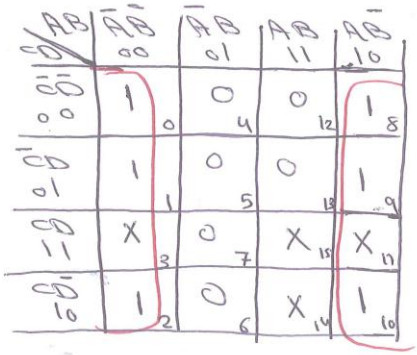
solution:-



$F = \bar{C}\bar{D} + AB + \bar{B}\bar{D}$.

Ex:- Simplify: $F = \sum 0,1,2,8,9,10 + \sum_d 3,11,14,15$ using k-map

solution:-



$F = \bar{B}$

Ex:- Design a logic cct using NAND gates only to convert BCD to Ex-3 code:-

Solution:-

ABCD	W X y Z
0000	0 0 1 1
0001	0 0 1 0
0010	0 0 1 0 1
0011	0 0 1 1 0
0100	0 0 1 1 1
0101	0 1 0 0 0
0110	0 1 0 0 1
0111	0 1 0 1 0
1000	0 1 0 1 1
1001	0 1 1 0 0
1010	X X X X
1011	X X X X
1100	X X X X
1101	X X X X
1110	X X X X
1111	X X X X

BCD count from 0 → 9
10 → 15 neglected

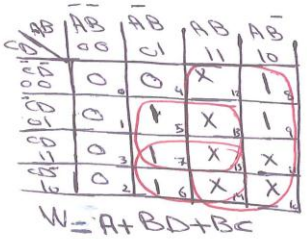
$d = \sum(10,11,12,13,14,15)$

$W = \sum(5,6,7,8,9)$

$X = \sum(1,2,3,4,9)$

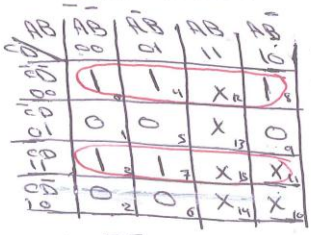
$y = \sum(0,3,4,7,8)$

$Z = \sum(0,2,4,6,8)$

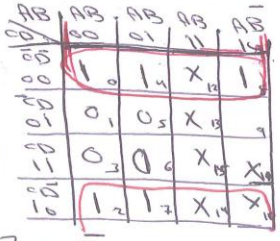


$W = A + BD + BC$

$X = B\bar{C}\bar{D} + \bar{B}D + \bar{B}C$



$y = \bar{C}\bar{D} + CD$



$Z = \bar{D}$

$W = A + BD + Bc$

$W = \overline{A \cdot \overline{BD} \cdot \overline{Bc}}$ ✓

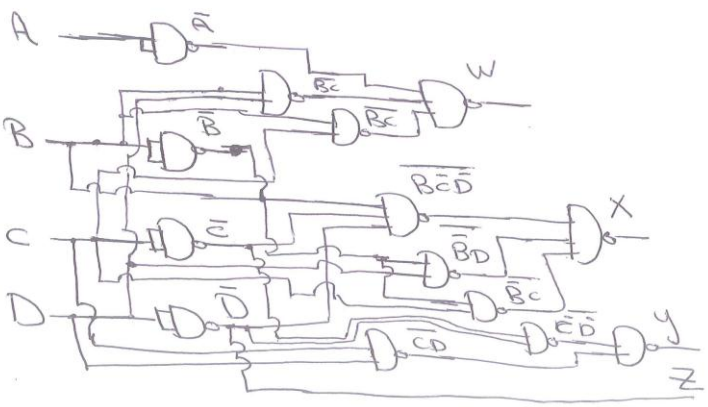
$X = B\bar{C}\bar{D} + \bar{B}D + \bar{B}C$

$X = \overline{B\bar{C}\bar{D} \cdot \overline{\bar{B}D} \cdot \overline{\bar{B}C}}$ ✓

$y = \bar{C}\bar{D} + CD$

$y = \overline{\bar{C}\bar{D} \cdot \overline{CD}}$ ✓

$Z = \bar{D}$ ✓



Ex:- Use K-map to design a logic circuit to convert from 5421 code to 8421 code using NAND gates only.

	ABCD	WXYZ
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
8	1000	1000
9	1001	1001
10	1010	1100
11	1011	1101
12	1100	1110
		1001

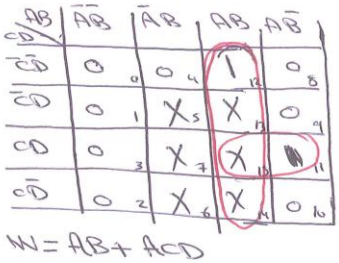
$d = \sum 5, 6, 7, 13, 14, 15$

$W = \sum 11, 12$

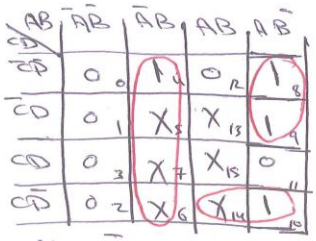
$X = \sum 4, 8, 9, 10$

$y = \sum 2, 3, 9, 10$

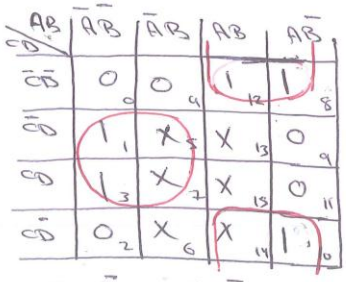
$Z = \sum 1, 3, 8, 10, 12$



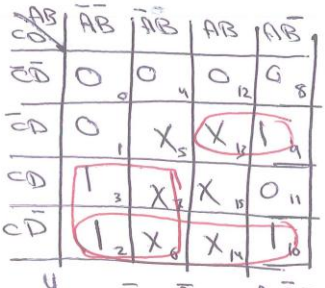
$W = AB + A\bar{C}D$



$X = \bar{A}B + A\bar{B}\bar{C} + AC\bar{D}$



$Z = \bar{A}D + A\bar{D}$



$y = C\bar{D} + \bar{A}C + A\bar{C}D$

$W = \overline{AB + A\bar{C}D} = \overline{AB} \cdot \overline{A\bar{C}D}$

$X = \overline{\bar{A}B + A\bar{B}\bar{C} + AC\bar{D}}$

$= \overline{\bar{A}B} \cdot \overline{A\bar{B}\bar{C}} \cdot \overline{AC\bar{D}}$

$y = \overline{C\bar{D} + \bar{A}C + A\bar{C}D} = \overline{C\bar{D}} \cdot \overline{\bar{A}C} \cdot \overline{A\bar{C}D}$

$Z = \overline{\bar{A}D + A\bar{D}} = \overline{\bar{A}D} \cdot \overline{A\bar{D}}$

Draw the cct as H.w

rity Generators and Checkers:-

Errors can occurs as digital codes are being transferred from one point to another within a digital system or while codes are being transmitted from one system to another. The errors take the form of undesired changes in the bits that make up the coded information; that is, a "1" can change to "0", or a "0" to "1", due to component malfunction or electrical noise.

-Parity bit Generator methods:-

Many systems, employ a parity bit as means of detecting a bit error. A word always contains either an even or an odd number of 1's. For this reason digital system employs some method for detection (sometimes correction) of errors. one of the simplest and most widely used schemes for error detection is "the parity bit method" and the two different methods used are:- 1- Even parity Generator method
2- Odd parity Generator method.

1- Even Parity Generator method:-

Even parity means attaching an extra bit to agroup of bits to produce an even number of 1's as shown in Table 1.

2- Odd parity Generator method:-

Odd parity means attaching an extra bit to a group of bits to produce an odd number of 1's. as shown in Table 2.

Table -1- Even parity Generator for three bits

A B C	Even parity P_e
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

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

$\bar{X} = \bar{A}\bar{B} + AB = \overline{A \oplus B}$

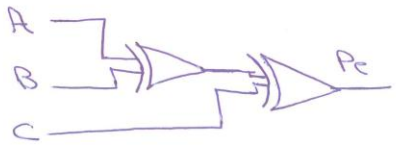
$P_e = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

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

$P_e = C\bar{X} + \bar{C}X$

$P_e = C \oplus X$

$P_e = C \oplus A \oplus B$



Logic diagram of even parity generator.

Table -2- Odd parity Generator for three bits.

A B C	Odd parity P_o
0 0 0	1
0 0 1	0
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	1
1 1 1	0

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

$\bar{X} = \bar{A}\bar{B} + AB = \overline{A \oplus B}$

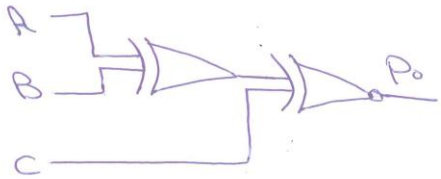
$P_o = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}C + AB\bar{C}$

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

$P_o = C\bar{X} + \bar{C}X$

$P_o = C \odot X$

$P_o = C \odot (A \oplus B)$



Logic diagram for odd parity generator.

Parity checking:-

Ex:- Show how to design a even parity checker cct for a 3-bit data:-

Solution:-

even Parity checker

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

$\bar{X} = AB + \bar{A}\bar{B} = \overline{A \oplus B}$

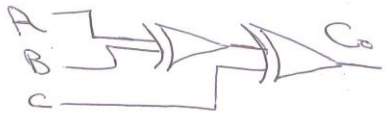
A B C	C_o
0 0 0	0
0 0 1	1 ✓
0 1 0	1 ✓
0 1 1	0
1 0 0	1 ✓
1 0 1	0
1 1 0	0
1 1 1	1 ✓

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

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

$C_o = C\bar{X} + \bar{C}X$

$C_o = C \oplus A \oplus B$



Ex:- Design a logic cct to provide the Odd parity bit for BCD code using NAND gates only:-

Solution:-

A B C D	P_o
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	✓
0 1 1 1	0
1 0 0 0	0
1 0 0 1	1

$d = \sum 10, 11, 12, 13, 14, 15$

$P_o = \sum 0, 3, 5, 6, 9$

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

$P_o = \overline{A\bar{D}} \cdot \overline{B\bar{C}D} \cdot \overline{B\bar{C}\bar{D}} \cdot \overline{\bar{A}\bar{B}\bar{C}\bar{D}} \cdot \overline{\bar{B}CD}$

H.W:- Draw the cct using NAND gates only.

H.w:— 1- Design an even/odd parity generator for 4-bit data, then implement it's cct. diagram.

2- Design a parity checker cct for a 4bit data, then implement its cct diagram.

Adders:-

Adders are important not only in computers, but in many type of digital systems in which numerical data are processed. There are two types of adders:-

1- Half adder
2- Full adder

1- Half Adder (H.A) :-

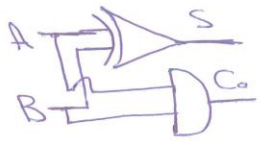
It is a digital circuit having two inputs and two outputs. The inputs are the inputs bits of the binary number and outputs are the sum of them and carry of them. The truth table and the block diagram are shown below:-

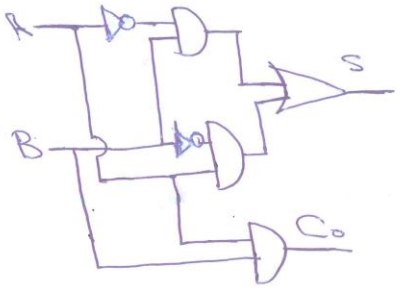
AB	SC_o
0 0	0 0
0 1	1 0
1 0	1 0
1 1	0 1

by mean of using gates, we can have different forms of H.A.

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

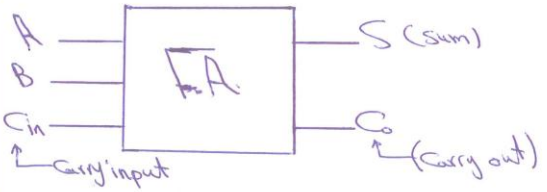
$C_o = AB$





2- Full Adder (F.A)

Application for the H.A are limited. The H.A is no sufficient for adding more than two one-bit numbers. Full Adder is a logic circuit that can add 3-bit and have two outputs. These outputs are the sum of three bits inputs and the second output is Carry of them. The truth table and the block diagram are shown below:-



A B Cin	S Co
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

From the truth table:-

$S = \sum 1,2,4,7$

$= \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$

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

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

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

$S = A \oplus B \oplus C_{in}$

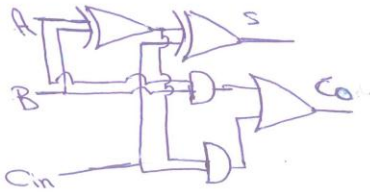
$B \odot C_{in} = \overline{B \oplus C_{in}}$

$C_o = \sum 3,5,6,7$

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

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

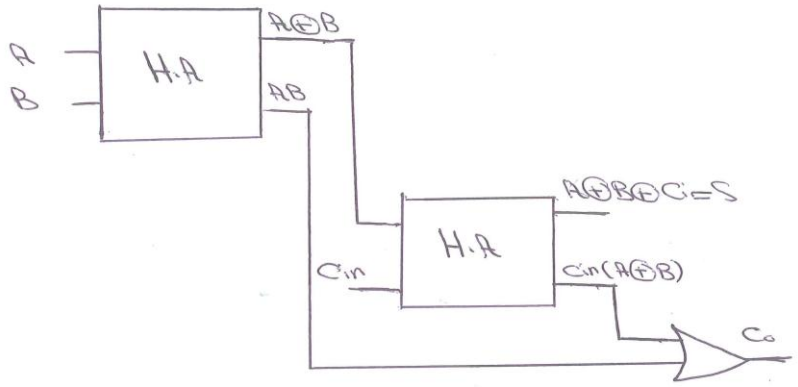
$= AB + C_{in}(A \oplus B)$



Ex:- Implement Full-Adder circuit diagram by using H.A blocks and any gate:-

Solution:- $S = A \oplus B \oplus C$
$C_o = AB + C_i(A \oplus B)$] F.A معادلات الجمع الكامل

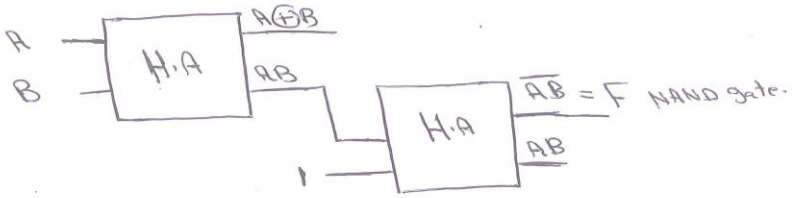
$S = A \oplus B$
$C_o = AB$] H.A معادلات نصف الجمع



Ex:- Design NAND gate by using H.A's only:-

Solution:- $F = \overline{AB}$] الجمع الكامل

$S = A \oplus B$
$C_o = AB$] معادلات نصف الجمع

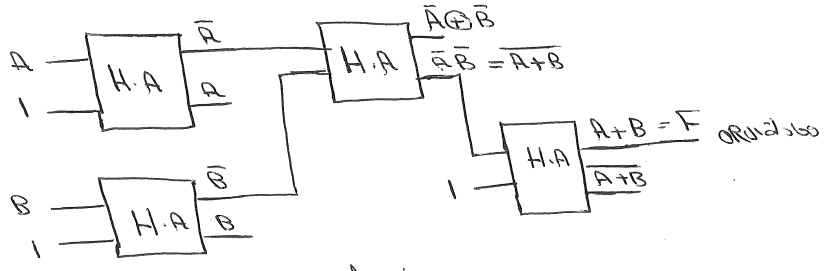


Note:- $A \oplus 1 = \overline{A}$
$A \oplus 0 = A$] Very Important.

Ex:- Design OR gate by using H.A only:-

Solution:- $F = A+B$] المعادلة المطلوب تنفيذها

$S = A \oplus B$
$C_o = AB$] المعادلات المتوفرة



Ex:- Implement the following functions below, using H.A blocks only:- boolean

$X = A \oplus B \oplus C$ $y = \bar{A}Bc + A\bar{B}c$ $Z = AB\bar{c} + c(\bar{A}+\bar{B})$

$W = ABc$

Solution:- $X = A \oplus B \oplus C$

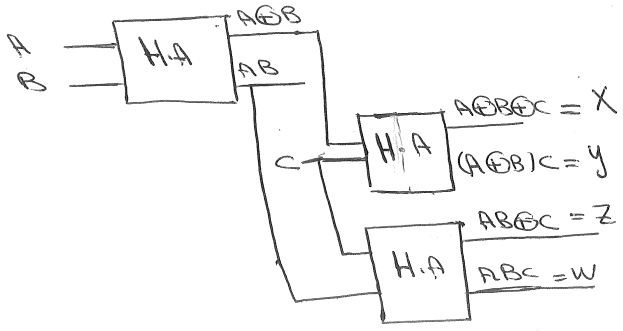
$y = \bar{A}Bc + A\bar{B}c = c(\bar{A}B + A\bar{B}) = c(A \oplus B)$

$Z = AB\bar{c} + c(\bar{A}+\bar{B}) = AB\bar{c} + c(\overline{AB})$
$= AB \oplus c$

] المعادلات المطلوب تنفيذها

$W = ABc$

$S = A \oplus B$
$C_o = AB$] المعادلات المتوفرة



Binary Parallel Adder:-

For two numbers of two bits, two adders are needed and for three bits numbers, three adders are needed and so on. The carry of each adder is connected to the carry input of the next heigher order adder.

Ex:- The two bit binary parallel adder is shown below.

Solution:- Let $A = A_1\ A_0$ (MSB → A_1, LSB → A_0)

$B = B_1\ B_0$ +

Result = $S_2\ S_1\ S_0$

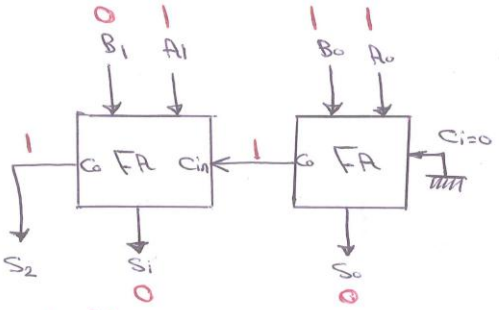
C_{in} = Carry input to LSB

C_o = Carry output to MSB

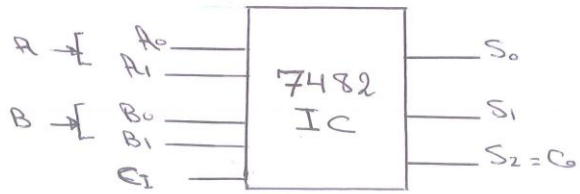
$A = 11$

$B = 01$ +

100



Note:- Two bit binary parallel adder 7482 IC



A_0, B_0 → (LSB input)

A_1, B_1 → (MSB input)

CI → (Carry input to the LSB)

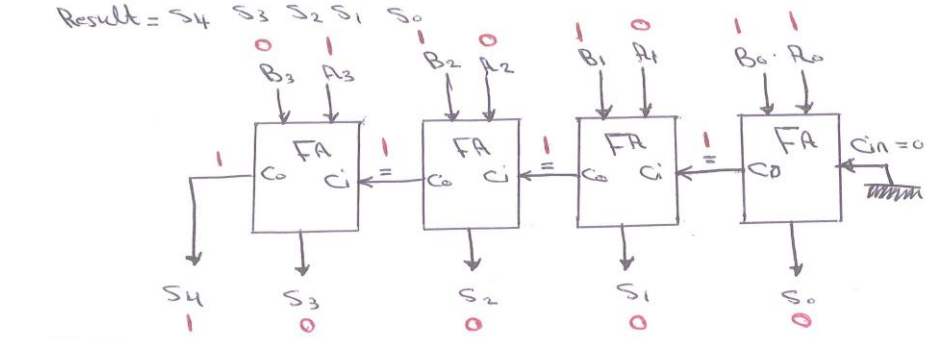
C_o → (Carry output to the MSB)

Ex:- The four bit binary parallel adder is shown below:-

Solution:- Let $A = A_3 A_2 A_1 A_0$ (MSB … LSB)

$B = B_3 B_2 B_1 B_0$ +

Result = $S_4\ S_3\ S_2\ S_1\ S_0$

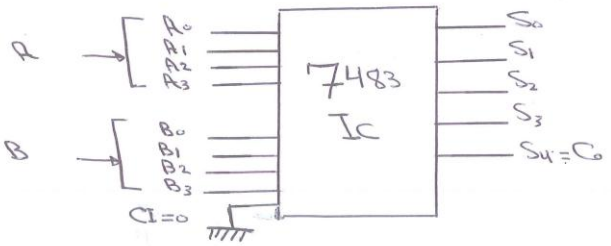


$A = 1001$

$B = 0111$ +

10000

Note:- Four bit binary parallel adder 7483 Ic



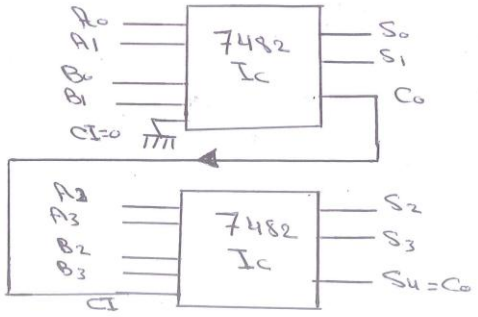
Ex:- Show how you can obtain Four bit adder from Ic of 7482 only:-

Solution:-

$A = A_3 A_2 A_1 A_0$

$B = B_3 B_2 B_1 B_0$

$S_4 S_3 S_2 S_1 S_0$



H.w:- Design 8-bits adder by using block diagram of IC 7483 (4-bit) adder.

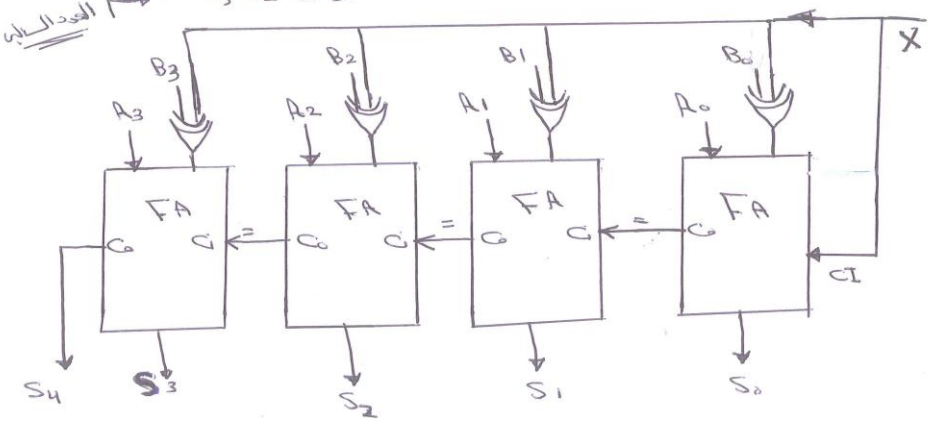
H.w:- Use Full adder block diagram of IC 7483 to convert BCD to Ex-3 code.

Adder / Subtractor:-

Ex:- Design an Adder/subtractor cct. using FA's and gates:-

Solution:- $A = A_3 A_2 A_1 A_0$

$B = B_3 B_2 B_1 B_0$ العدد الثاني



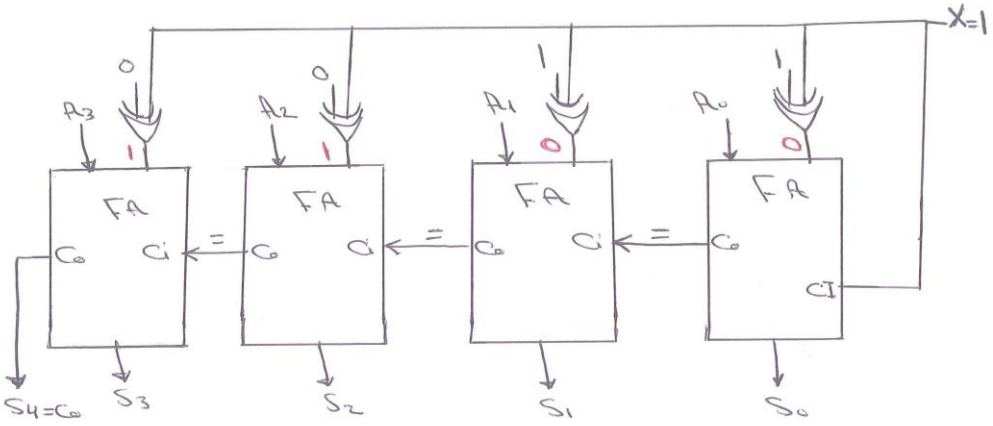
X=0 in Adding operation and output result = $S_4 S_3 S_2 S_1 S_0$

X=1 in Subtraction operation and output result equal $S_3 S_2 S_1 S_0$ and S_4 or Co will be (discarded) due to 2's complement operation.

Ex:- Using an Adder/subtractor cct. to convert from Ex-3 code to BCD code:-

Solution:- Let $A = A_3 A_2 A_1 A_0$ in Ex-3 code

Then BCD number = $A_3\ A_2\ A_1\ A_0$
$0\ 0\ 1\ 1$ -

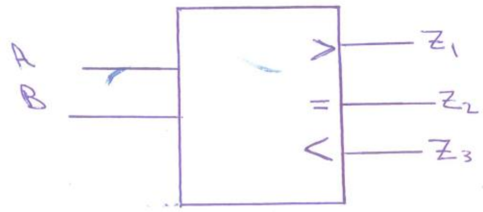


H.w:- Using an adder/subtractor cct. to convert from BCD code to Ex-3 code?

Comparators:-

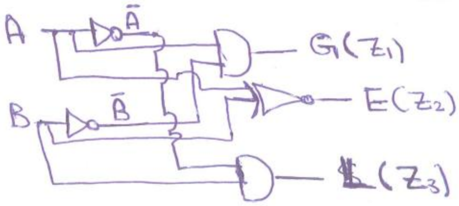
A magnitude comparator is a combinational circuit that compares two numbers (A and B), and determines their relative magnitudes. The outputs of the comparsion are specified by three binary variables that indicate whether A=B or A>B or A<B.

The block diagram for comparator is shown below:-



- For two binary numbers, each one have one bit, the truth table:-

A	B	Z_1 (A>B)	Z_2 (A=B)	Z_3 (A<B)
0	0	0	1 ✓	0
0	1	0	0	1 ✓
1	0	1 ✓	0	0
1	1	0	1 ✓	0



$Z_1(A>B) = A\bar{B}$

$Z_2(A=B) = \bar{A}\bar{B} + AB = A \odot B$

$Z_3(A<B) = \bar{A}B$

- For two binary numbers, each one have two bits (n=2)

MSB ↓ MSB ↓

$A = A_1 A_0 \quad , B = B_1 B_0$

Note: $X_0 = A_0 \odot B_0$
$X_1 = A_1 \odot B_1$

$A > B = (A_1 > B_1)$ or $(A_1 = B_1)$ and $(A_0 > B_0)$

$= (A_1\bar{B}_1) + (A_1 \odot B_1)(A_0\bar{B}_0)$

$= (A_1\bar{B}_1) + X_1 . A_0\bar{B}_0$

* المقارنة دائماً تبدأ من ال MSB

$A < B = (A_1 < B_1)$ or $(A_1 = B_1)$ and $(A_0 < B_0)$

$= \bar{A}_1 B_1 + (A_1 \odot B_1) . (\bar{A}_0 B_0)$

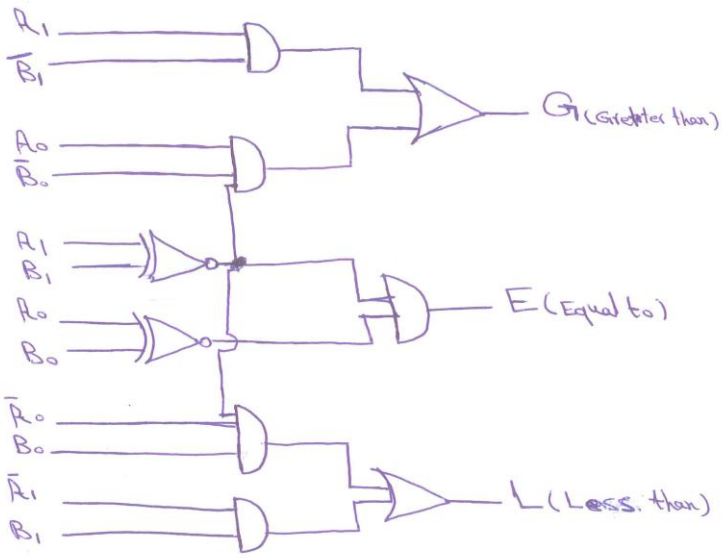
$= \bar{A}_1 B_1 + X_1 . \bar{A}_0 B_0$

$A=B \quad = (A_1 = B_1) \text{ and } (A_0 = B_0)$

$= (A_1 \odot B_1) \cdot (A_0 \odot B_0)$

$= X_1 X_0$

From the expression we may obtain the digital comparator circuit as shown below:-



Digital Comparator Circuit for 2-bits

- For two binary numbers, each one have 4-bits ($n=4$).

$A = A_3 A_2 A_1 A_0$, $B = B_3 B_2 B_1 B_0$

Note: $X_0 = A_0 \odot B_0$
$X_1 = A_1 \odot B_1$
$X_2 = A_2 \odot B_2$
$X_3 = A_3 \odot B_3$

$A > B (G_1) = A_3 > B_3$ or $A_3 = B_3$ and $A_2 > B_2$ or $A_3 = B_3$ and $A_2 = B_2$ and $A_1 > B_1$ or $A_3 = B_3$ and $A_2 = B_2$ and $A_1 = B_1$ and $A_0 > B_0$.

$= A_3 \bar{B}_3 + (A_3 \odot B_3) A_2 \bar{B}_2 + (A_3 \odot B_3)(A_2 \odot B_2) A_1 \bar{B}_1 + (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1)(A_0 \bar{B}_0)$

$= A_3 \bar{B}_3 + X_3 A_2 \bar{B}_2 + X_3 X_2 A_1 \bar{B}_1 + X_3 X_2 X_1 A_0 \bar{B}_0$

$A < B (L) = A_3 < B_3$ or $A_3 = B_3$ and $A_2 < B_2$ or $A_3 = B_3$ and $A_2 = B_2$ and $A_1 < B_1$ or $A_3 = B_3$ and $A_2 = B_2$ and $A_1 = B_1$ and $A_0 < B_0$

$$= \dot{A}_3 B_3 + (A_3 \odot B_3)(\dot{A}_2 B_2) + (A_3 \odot B_3)(A_2 \odot B_2)\dot{A}_1 B_1 + (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1)(\dot{A}_0 B_0)$$

$$= \dot{A}_3 B_3 + X_3 \dot{A}_2 B_2 + X_3 X_2 \dot{A}_1 B_1 + X_3 X_2 X_1 \dot{A}_0 B_0$$

$$A = B\ (E) = (A_3 = B_3) \text{ and } (A_2 = B_2) \text{ and } (A_1 = B_1) \text{ and } (A_0 = B_0)$$
$$= (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1)(A_0 \odot B_0)$$
$$= X_3 X_2 X_1 X_0$$

Ex:- Implement a comparator of two binary numbers each one have four bits, Let $A = {}^{\text{eu}}_{\text{so}}0101$ and $B = {}^{\text{eu}}_{\text{so}}1101$

Solution:- $A = A_3 A_2 A_1 A_0$
$B = B_3 B_2 B_1 B_0$

by substitute the values of A and B in the previous equations:-

$G = 0 + 0 + 0 + 0 = 0$

$L = 1 + 0 + 0 + 0 = 1$ $\quad \therefore A < B$

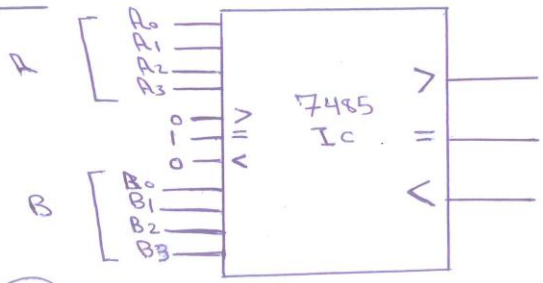
$E = 0 * 1 * 1 * 1 = 0$

H.W:- For each of the binary numbers, determine the output states for the comparator:-

A) $A_3 A_2 A_1 A_0 = 1100$
$B_3 B_2 B_1 B_0 = 1001$

B) $A_3 A_2 A_1 A_0 = 0100$
$B_3 B_2 B_1 B_0 = 0100$

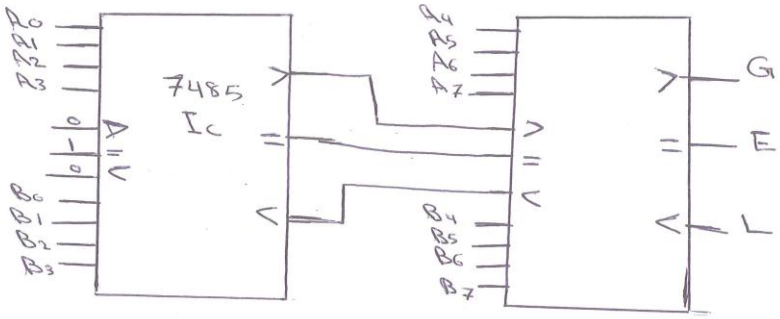
7485 IC:-



Two binary numbers of four bits comparator 7485 Ic

Ex:- Use 7485 Ic to compare two binary numbers each one have eight bits:- $A = A_7A_6A_5A_4A_3A_2A_1A_0$, $B = B_7B_6B_5B_4B_3B_2B_1B_0$

Solution:-



Ex:- Design H.A. using Comparators only:-

Solution:- H.A

$S = A\bar{B} + \bar{A}B = A \oplus B$] المخرج الأول

$C_o = AB$ المخرج الثاني

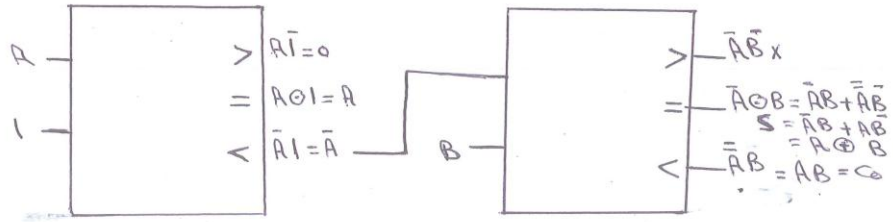
Note:-
$A \odot 1 = A$
$A \odot 0 = \bar{A}$

Comparator

$A > B\ (G) = A\bar{B}$

$A = B\ (E) = AB + \bar{A}\bar{B} = A \odot B$] المخرج الثاني

$A < B\ (L) = \bar{A}B$



H.W:- Design F.A, using Comparators only.

H.w:- Design H.A by using Comparators and any gate.

H.w:- Design F.A by using Comparator and one gate.

H.W:- Verify all logic gates using Comparator only.
AND, NOR, OR, EX-OR - EX-NOR, NAND, NOT.

Decoder and Encoder

Decoder:-

A decoder is a combinational logic circuit that converts coded information such as binary, into a recognizable form, such as decimal. There are many type of decoders such as:-

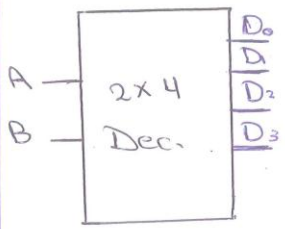
1- Binary decoder.
2- BCD to Decimal decoder.
3- BCD to 7-Segment decoder.

Note:- In general form, a decoder has n input lines to handle n bits and form one to 2^n output lines to indicate the presence of one or more n-bit combinations.

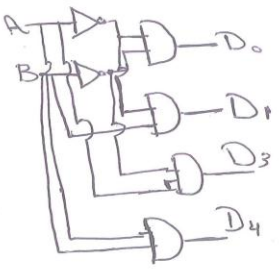
1- Binary decoder:- $(n*2^n)$ (n-to-2^n) decoder.

It converts binary coded information in to adecimal recognizable form. The output lines of this decoder is 2^n where n is the number of bits of the binary input such as 2x4 decoder, 3x8 decoder, 4x16 decoder.

* 2x4 line decoder cct. is shown below with the Truth table



AB	D_0	D_1	D_2	D_3
0 0	1	0	0	0
0 1	0	1	0	0
1 0	0	0	1	0
1 1	0	0	0	1



From the truth table:-

$D_0 = \bar{A}\bar{B}$; $D_2 = A\bar{B}$

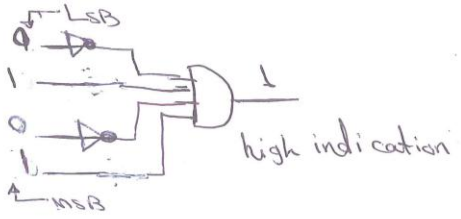
$D_1 = \bar{A}B$; $D_3 = AB$

Ex:- Show how you can indicat the No(10)16 for high or low
or Determine the logic required to decode the binary code 1010 with active high or low output.

Solution:- Active High

A AND gate is used and the output = 1

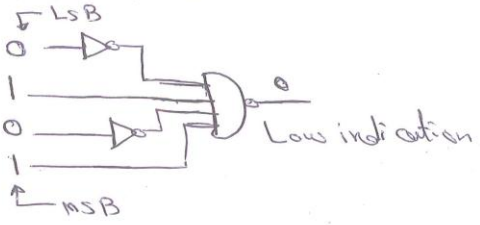
1010 = $A\bar{B}C\bar{D}$



Active Low

A NAND gate is used and the output = 0

1010 = $A\bar{B}C\bar{D}$



Implement Logic Functions Using Decoders:-

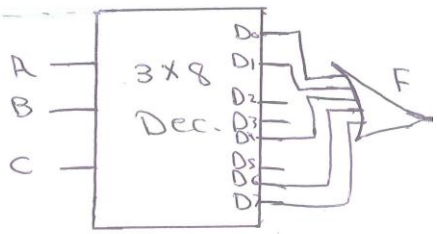
Ex:- Implement the following logic function using decoders and Logic gates:-

$F = \sum_{D_0\ D_1\ D_4\ D_6\ D_7} 0, 1, 4, 6, 7$

Solution:-

$D_0 = \bar{A}\bar{B}\bar{C}$
$D_1 = \bar{A}\bar{B}C$
$D_4 = A\bar{B}\bar{C}$
$D_6 = AB\bar{C}$
$D_7 = ABC$

ABC	F
000	1
001	1
010	0
011	1
100	1
101	0
110	1
111	1



$F = D_0 + D_1 + D_4 + D_6 + D_7$

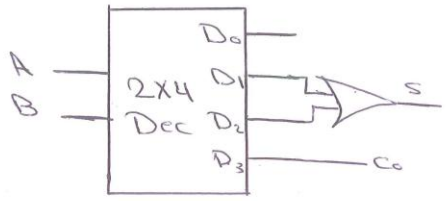
Ex:- Implement HA cct using decoder and logic gates:-

Solution:-

$S = \sum 1, 2 = \bar{A}B + A\bar{B} = D_1 + D_2$

$C_o = \sum 3 = AB = D_3$

AB	S	C_o
00	0	0
01	1	0
10	1	0
11	0	1



H.w:- By means of decoder and logic gates implement:-

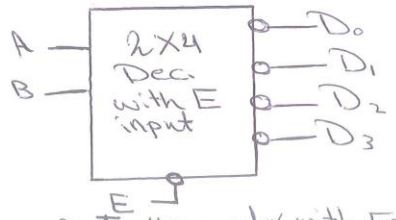
1- FA cct

2- $F = \sum o, 2, 5, 6, 7.$

Enable Control Inputs:-

Decoders include one or more enable inputs to control the circuit operation. In general, a decoder may operate with complemented or uncomplemented outputs. The enable inputs may be activated with a 0 or with a 1 signal. Some decoders have two or more enable i/p's that must satisfy a given logic condition in order to enable the cct.

A 2X4 line decoder with an enable input constructed with NAND gate is shown below:-



2-to-4 Decoder with Enable input (Active Low)

E_o	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

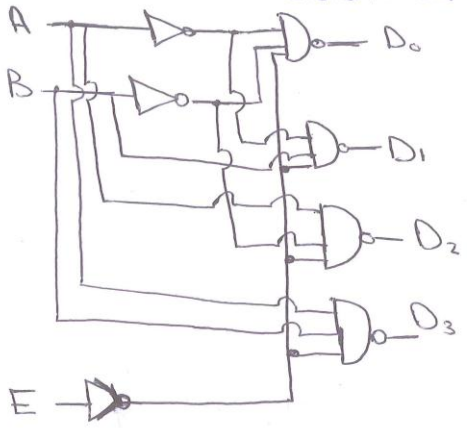
Truth Table

$D_0 = \overline{E}\overline{A}\overline{B}$

$D_1 = \overline{E}\overline{A}B$

$D_3 = \overline{E}AB$

$D_4 = \overline{E}AB$



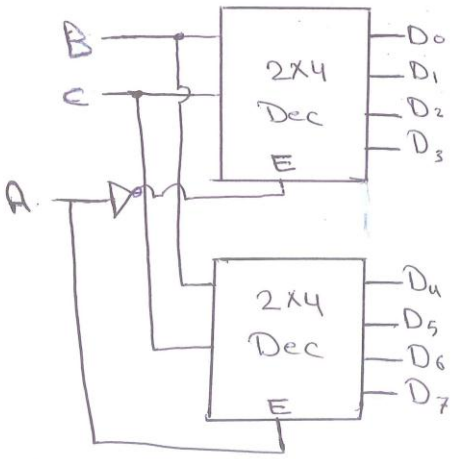
VIP

Note:- Decoders with enable inputs can be connected together to form a larger decoder circuit.

Ex:- Design 3×8 decoder using 2×4 decoders with Enable:-

Solution:-

	A B C	D_0 D_1 D_2 ... D_7
1st Dec	0 0 0	1 0 0 ... 0
	0 0 1	0 1 0 ... 0
	0 1 0	0 0 1 ... 0
	0 1 1	. . .
2nd Dec	1 0 0	. . .
	1 0 1	. . .
	1 1 0	. . .
	1 1 1	0 0 0 ... 1



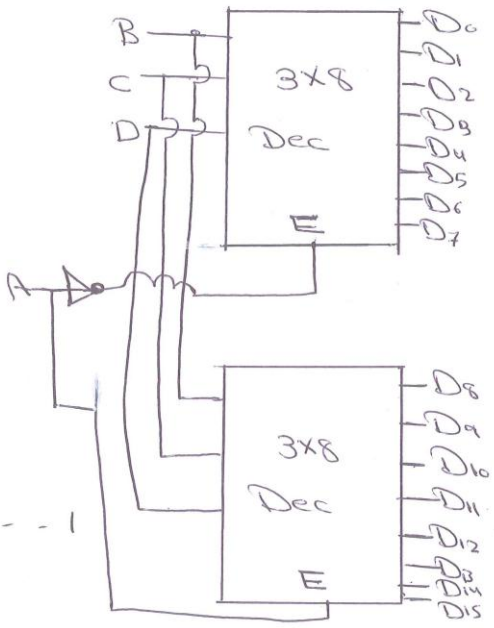
Ex:- Design 4 to-16 Decoder, using 3×8 decoders with enable.

Solution:-

A B C D	D_0 D_1 ---- D_{15}
0 0 0 0	1 0 - - - - 0
0 0 0 1	0 1 - - - - - 0
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	0 0 - - - 1

1st Dec

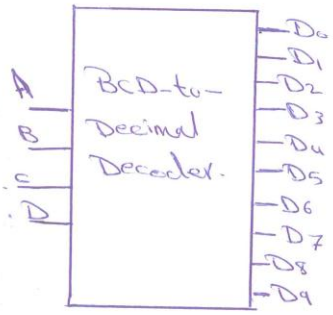
2nd Dec



H.W:- Design 4-to-16 Decoder, using 2X4 decoders with enables.

2- BCD to Decimal Decoders:-

The BCD decoder converts each BCD code (8421) into one of the possible decimal digit indications it is referred to as 4-line-to-10-line decoder or a 4-to-10 decoder. A list of the ten BCD codes and their corresponding decoding functions is given in Table below:-



ABCD	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9
0000	1	0	0	0	0	0	0	0	0	0
0001	0	1	0	0	0	0	0	0	0	0
0010	0	0	1	0	0	0	0	0	0	0
0011	0	0	0	1	0	0	0	0	0	0
0100	0	0	0	0	1	0	0	0	0	0
0101	0	0	0	0	0	1	0	0	0	0
0110	0	0	0	0	0	0	1	0	0	0
0111	0	0	0	0	0	0	0	1	0	0
1000	0	0	0	0	0	0	0	0	1	0
1001	0	0	0	0	0	0	0	0	0	1
1010	X	X	X	X	X	X	X	X	X	X
1011	X	-	-	-	-	-	-	-	-	X
1100	'	'	'	'	'	'	'	'	'	'
1101	'	'	'	'	'	'	'	'	'	'
1110	'	'	'	'	'	'	'	'	'	'
1111	X	X	X	X	X	X	X	X	X	X

$D_0 = \bar{A}\bar{B}\bar{C}\bar{D}$
$D_1 = \bar{A}\bar{B}\bar{C}D$
$D_2 = \bar{B}C\bar{D}$
$D_3 = \bar{B}CD$
$D_4 = B\bar{C}\bar{D}$
$D_5 = B\bar{C}D$
$D_6 = BC\bar{D}$
$D_7 = BCD$
$D_8 = A\bar{D}$
$D_9 = AD$

don't care $= \sum 10, 11, 12, 13, 14, 15$

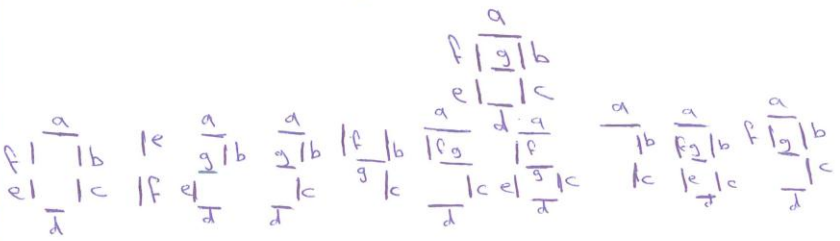
CD \ AB	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}\bar{D}$	0	4	X_{12}	8
$\bar{C}D$	1	5	X_{13}	9
CD	3	7	X_{15}	X_{11}
$C\bar{D}$	1	6	X_{14}	X_{10}

$D_2 = \bar{B}C\bar{D}$

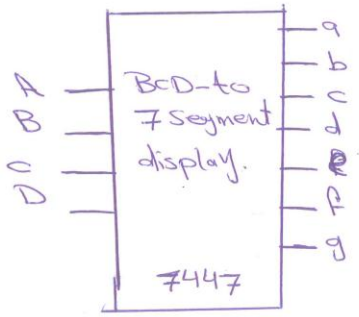
H.w:- Write the K-map for the other D. [D3 ... D9]

3- BCD to 7-Segment Decoder:-

This type of decoder accepts the BCD code on its input and provides outputs to drive 7-segment display in order to produce a decimal digital readout.



BCD	a	b	c	d	e	f	g
0000	1	1	1	1	1	1	0
0001	0	0	0	0	1	1	0
0010	1	1	0	1	1	0	1
0011	1	1	1	1	0	0	1
0100	0	1	1	0	0	1	1
0101	1	0	1	1	0	1	1
0110	1	0	1	1	1	1	1
0111	1	1	1	0	0	0	0
1000	1	1	1	1	1	1	1
1001	1	1	1	1	0	1	1
1010	—	—	—	—	—	—	—
1011	X	-	-	-	-	-	X
1100	X						
1101	X						
1110							
1111	X	-	-	-	-	-	X



$a = \sum 0, 2, 3, 5, 6, 7, 8, 9$

$b = \sum 0, 2, 3, 4, 7, 8, 9$

$c = \sum 0, 3, 4, 5, 6, 7, 8, 9$

$d = \sum 0, 2, 3, 5, 6, 8, 9$

$e = \sum 0, 1, 2, 6, 8$

$f = \sum 0, 1, 4, 5, 6, 8, 9$

$g = \sum 2, 3, 4, 5, 6, 8, 9$

don't care $= \sum 10, 11, 12, 13, 14, 15$

Encoder:-

An encoder is a digital circuit that performs the inverse operation of a decoder. An encoder has 2^n (or fewer) input lines and n output lines. An encoder accepts an active level on one of its inputs representing a digit, such as a decimal or octal digit, and converts it to a coded output such as BCD or Binary

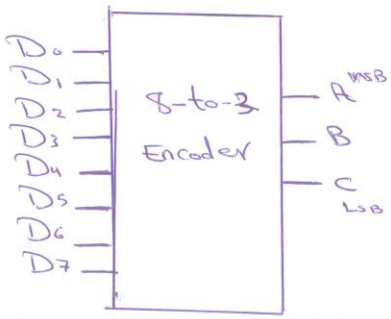
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	A	B	C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Truth Table of octal-to-binary Encoder

$A = D_4 + D_5 + D_6 + D_7$

$B = D_2 + D_3 + D_6 + D_7$

$C = D_1 + D_3 + D_5 + D_7$.



Logic symbol for octal-to-binary Encoder.

Truth table of Decimal to BCD Encoder

D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	A	B	C	D
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1

MSB

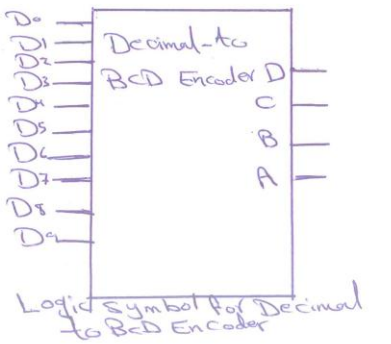
$A = D_8 + D_9$

$B = D_4 + D_5 + D_6 + D_7$

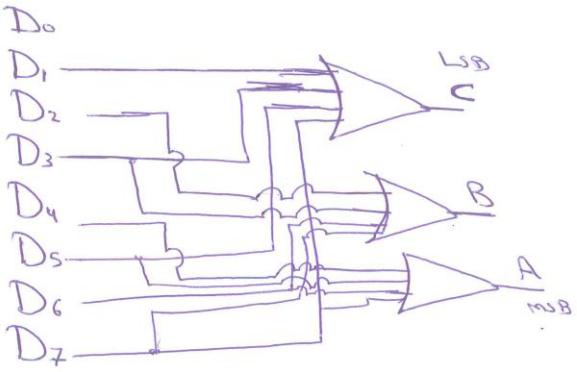
$C = D_2 + D_3 + D_6 + D_7$

$D = D_1 + D_3 + D_5 + D_7$

LSB



Logic symbol for Decimal to BCD Encoder



Basic logic diagram of a octal to Binary Encoder.

H.w :- Draw the Basic logic diagram of a decimal to BCD Encoder.

Note:- If two inputs are active simultaneously, the output produces an undefined combination. To resolve this ambiguity, encoder circuits must establish an input priority to ensure that only one input is encoded. If we establish a higher priority for i/p's with higher subscript numbers, and if both D_3 and D_6 are 1 at the same time, the output will be 110 because D_6 has higher priority than D_3.

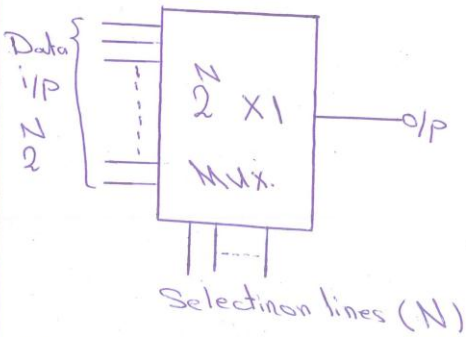
20161012

Multiplexer and Demultiplexer :-

Multiplexer :-

A multiplexer (Mux) is a device that allows digital information from several sources to be routed onto a single line for transmission over that line to a common destination. The basic multiplexer, then has several data input lines (2^N) and a single o/p line. It also has data selector inputs (N) that permit digital data on any one of the input to be switched to the o/p line.

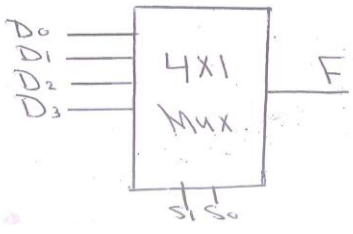
The block diagram for such multiplexer ($2^N \times 1$) is shown below:-



The multiplexer may be considered as a multi-switch.

The block diagram of 4x1 Mux. and its truth table is shown below:-

S_1	S_0	F
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

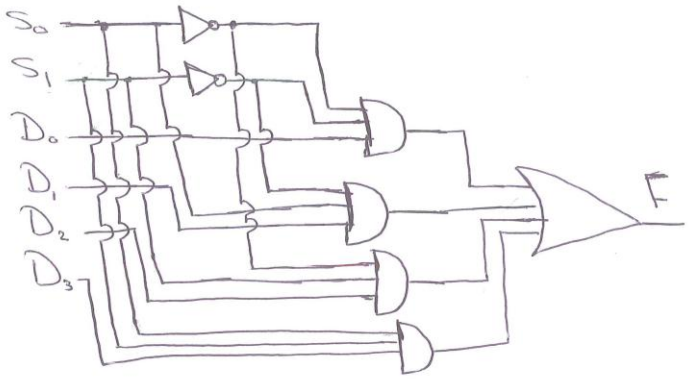


(80)

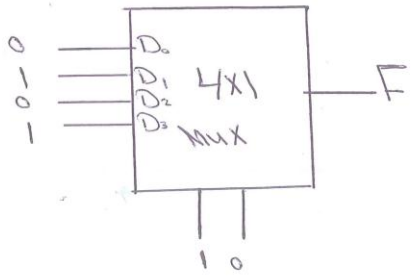
The equation of F is:-

$$F = \bar{S}_1\bar{S}_0 D_0 + \bar{S}_1 S_0 D_1 + S_1\bar{S}_0 D_2 + S_1 S_0 D_3$$

The Circuit diagram of the equation is:-



Ex :- Find the o/p of the following:-



Solution:- $S_1S_0 = \frac{10}{2}$

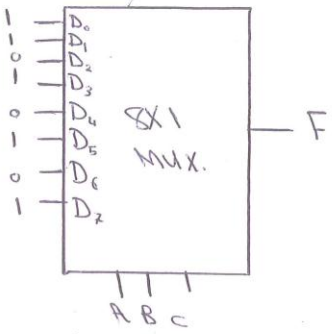
$F = D_2 = 0$

Ex:- Implement the following function be means of using the suitable Mux. $F = \Sigma 0, 1, 3, 5, 7.$

Solution:-

1) By using 8X1 MUX

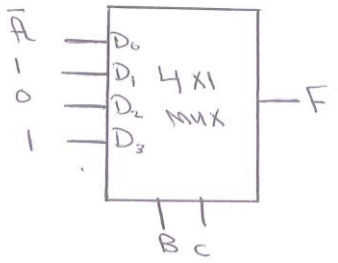
ABC	F
0 0 0	1
0 0 1	1
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	0
1 1 1	1

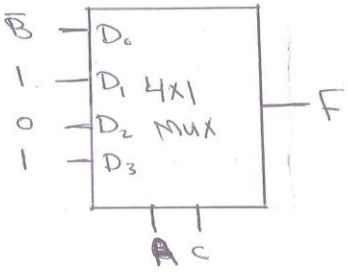


2) By using 4X1 MUX. 1 → Bc, 2 → Ac as select line.

Selection input	D_0 $\bar{B}\bar{C}$	D_1 $\bar{B}C$	D_2 $B\bar{C}$	D_3 BC
$\bar{A}$	0	1	2	3
A	4	5	6	7
result	$\bar{A}$	1	0	1

result / input	D_0 $\bar{A}\bar{C}$	D_1 $\bar{A}C$	D_2 $A\bar{C}$	D_3 AC
$\bar{B}$	0	1	4	5
B	2	3	6	7
result	$\bar{B}$	1	0	1





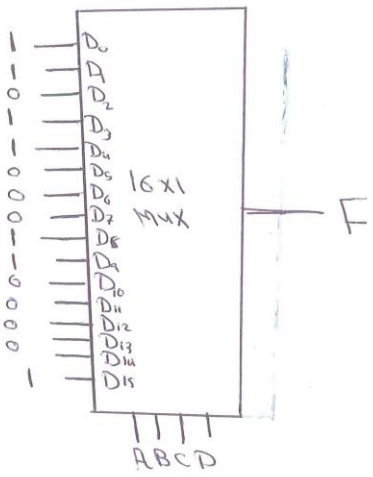
Ex:- If $F=\sum(0,1,3,4,8,9,15)$, Implement it using :-

1- 16x1 Mux ; 2- 8x1 Mux. 3- 4x1 Mux. 4. 2x1 Mux.

Solution:-

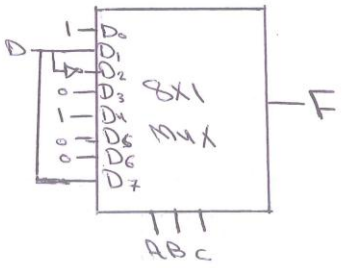
1- By using 16x1 Mux

ABCD	F
0000	1
0001	1
0010	0
0011	0
0100	1
0101	0
0110	0
0111	0
1000	1
1001	1
1010	0
1011	0
1100	0
1101	0
1110	0
1111	1



2- 1x8 Mux.

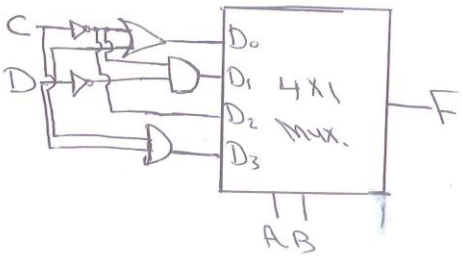
selecting input	D_0 $\bar{A}\bar{B}\bar{C}$	D_1 $\bar{A}\bar{B}C$	D_2 $\bar{A}B\bar{C}$	D_3 $\bar{A}BC$	D_4 $A\bar{B}\bar{C}$	D_5 $A\bar{B}C$	D_6 $AB\bar{C}$	D_7 ABC
$\bar{D}$	(0)	2	(4)	6	(8)	10	12	14
D	(1)	(3)	5	7	(9)	11	13	(15)
result	1	D	$\bar{D}$	0	1	0	0	D



3- 4x1 Mux.

S/I input	D_0 $\bar{A}\bar{B}$	D_1 $\bar{A}B$	D_2 $A\bar{B}$	D_3 AB
$\bar{C}\bar{D}$	(0)	(4)	(8)	12
$\bar{C}D$	(1)	5	(9)	13
$C\bar{D}$	2	6	10	14
CD	(3)	7	11	(15)
result		$\bar{C}\bar{D}$	$\bar{C}$	CD

$\bar{C}\bar{D} + \bar{C}D + CD = \bar{C} + D$



4- 2x1 Mux.

SL / input	D_0 $\bar{A}$	D_1 A
$\bar{B}\bar{C}\bar{D}$	(0)	(8)
$\bar{B}\bar{C}D$	(1)	(9)
$\bar{B}C\bar{D}$	2	10
$\bar{B}CD$	(3)	11
$B\bar{C}\bar{D}$	(4)	12
$B\bar{C}D$	5	13
$BC\bar{D}$	6	14
BCD	7	(15)
result	↓	↓

$\bar{B}\bar{C}\bar{D}+\bar{B}\bar{C}D+\bar{B}CD+B\bar{C}\bar{D}$

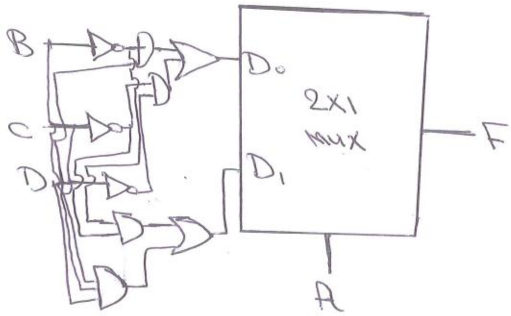
$\bar{C}\bar{D}[\bar{B}+B]+\bar{B}D[\bar{C}+C]$

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

$\bar{B}\bar{C}\bar{D}+\bar{B}\bar{C}D+BCD$

$\bar{B}\bar{C}[\bar{D}+D]+BCD$

$\bar{B}\bar{C}+BCD$



Ex:- Implement the function $F=\sum 0,2,4,6,7+\sum_d 1$

using :- 1-(4x1) Mux.

2-(2x1) multiplexers. ((2x1 Mux) استخدام قطعة واحدة من ال)

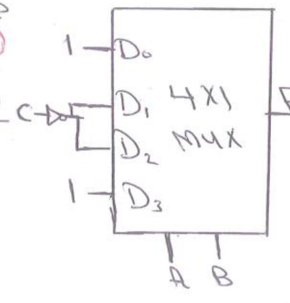
3-(2x1) Mux (As HW)

Solution:-

1- (4x1) Mux :-

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

SL / input	D_0 $\bar{A}\bar{B}$	D_1 $\bar{A}B$	D_2 $A\bar{B}$	D_3 AB
$\bar{C}$	(0)	(2)	(4)	(6)
C	(1x)	3	5	(7)
result	1	$\bar{C}$	$\bar{C}$	1



V.Imp

2-(2x1) Multiplexers:-

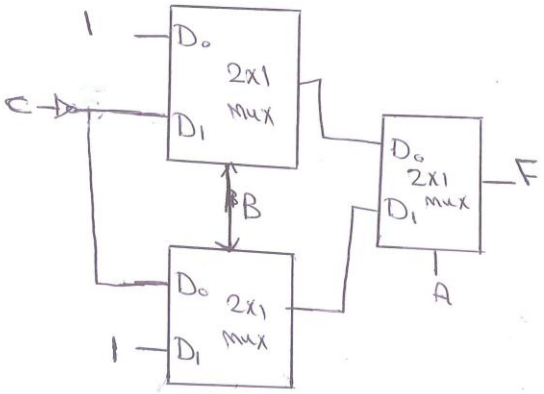
D_0 D_1

A=0 A=1 ← 2x1 mux

2x1 mux: $D_0 \rightarrow \bar{C}$, $D_1 \rightarrow C$

2x1 mux ← الاخير

S/L input	$\bar{A}\bar{B}$	$\bar{A}B$	$A\bar{B}$	AB
$\bar{C}$	0	2	4	6
C	1x	3	5	7
	1	$\bar{C}$	$\bar{C}$	1



Another way:-

A	B	$\bar{C}$	C	Result	
0	0	0	1x	1	D_0
	1	2	3	$\bar{C}$	D_1
1	0	4	5	$\bar{C}$	D_0
	1	6	7	1	D_1

Ex:- Implement HA, using:-

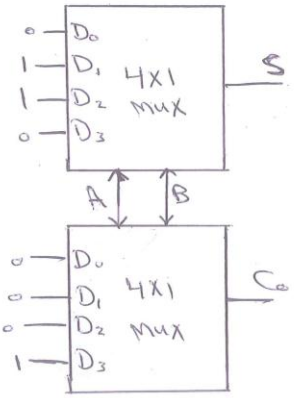
1- proper Mux.
2- 2x1 mux.
3- (2x1) multiplexers.

Solution:-

AB	S	C_o
00	0	0
01	1	0
10	1	0
11	0	1

$S = \sum 1, 2$

$C_o = \sum 3$



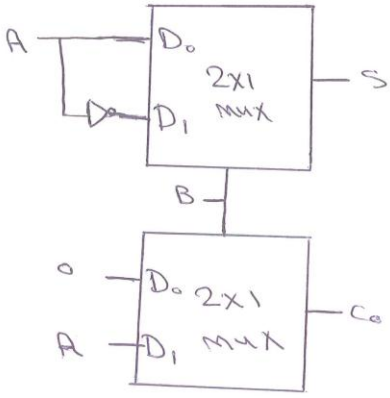
2- (2x1) mux (S and C_o) must have the same select line.

a) for $S = \sum 1,2$

input \ SL	$\bar{B}$	B
$\bar{A}$	0	(1)
A	(2)	3
result	A	$\bar{A}$

b) for $C_o = \sum 3$

input \ SL	$\bar{B}$	B
$\bar{A}$	0	1
A	2	(3)
result	0	A



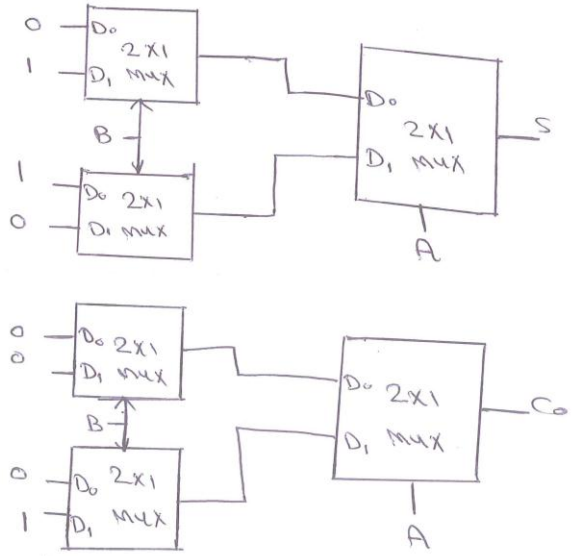
3- (2x1) multiplexers:-

a) for S

A B	S
0 0	0
0 1	1
1 0	1
1 1	0

b) for C_o

A B	C_o
0 0	0
0 1	0
1 0	0
1 1	1



H.w:- Implement FA by using:-

1- proper mux
2- 4x1 mux
3- 2x1 mux
4- (2x1) multiplexers.

Ex:- Implements the following function by using (4x1 Mux) and (2x1 Mux). $F = \sum 0,1,3,4,8,9,15$.

Solution:-

ABCD	F
0 0 0 0	1
0 0 0 1	1
0 0 1 0	0
0 0 1 1	1
0 1 0 0	1
0 1 0 1	0
0 1 1 0	0
0 1 1 1	0
1 0 0 0	1
1 0 0 1	1
1 0 1 0	0
1 0 1 1	0
1 1 0 0	0
1 1 0 1	0
1 1 1 0	0
1 1 1 1	1

4x1 mux Need 2 select line

2x1 mux Need 1 select line

∴ total No. of select line = 2+1 = 3

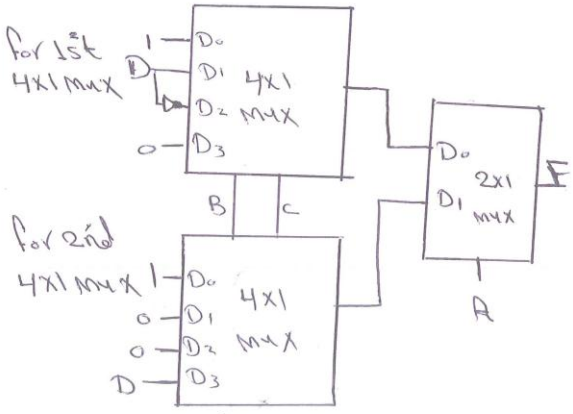
A is a select line for 2x1 Mux.

Bc are select lines for 4x1 Mux.

D is an input.

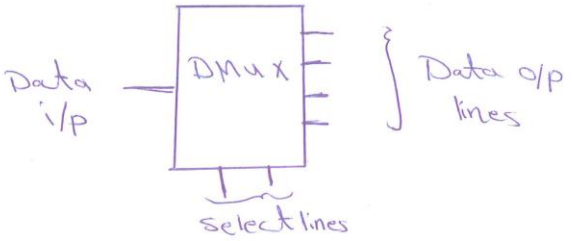
SL / input	D_0 = $\bar{B}\bar{C}$	D_1 = $\bar{B}C$	D_2 = $B\bar{C}$	D_3 = BC
$\bar{D}$	(0)	2	(4)	6
D	(1)	(3)	5	7
result	1	D	$\bar{D}$	0

SL / input	D_0 = $\bar{B}\bar{C}$	D_1 = $\bar{B}C$	D_2 = $B\bar{C}$	D_3 = BC
$\bar{D}$	(8)	10	12	14
D	(9)	11	13	(15)
result	1	0	0	D



Demultiplexer:-

A demultiplexer (DMUX) basically reverses the multiplexing function. It takes data from one line and distributes them to a given number of output lines.



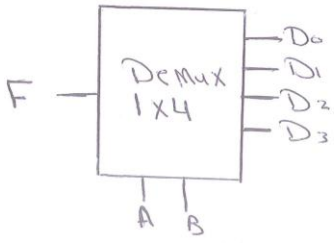
Ex:- $F = \Sigma 0, 2$

S_1 S_0	F	D_0	D_1	D_2	D_3
0 0	1	1	0	0	0
0 1	0	0	0	0	0
1 0	1	0	0	1	0
1 1	0	0	0	0	1

when $S_1S_0 = 00$

$O/P = D_0 = F = 1$

Note:- The decoder with enable input can be used as Demultiplexer which is a circuit that receives information in single line and transmit this information on the multi output.

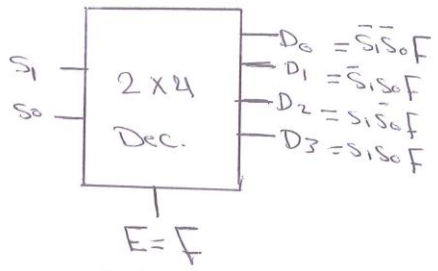


$D_0 = F\bar{A}\bar{B}$

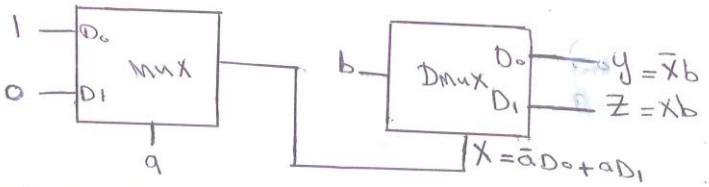
$D_1 = F\bar{A}B$

$D_2 = FA\bar{B}$

$D_3 = FAB$



Ex:- Find the value of y and z for the circuit below:-



a	b	X	y	z
0	0	1	0	0
0	1	1	0	1
1	0	0	0	0
1	1	0	1	0