Digital Logic Design :-

Boolean algebra

Set of Rules used to simplify the Given logic Expression uski functionality ko change Kare Bina.

For Exp $\quad F = A'B + BC + ABC \rightarrow (I)$

By Doing Minimization

$F = A'B + BC \rightarrow (II)$

functionality same rehni chaiee matlab, Jo (I) & (II) hai, don ki truth table mein Result same ana chaiee.

Drawbacks

(1) Or yeh sbse Bada drawback hai Boolean algebra ka - ki usme functionality same rehni chaiee!

(2) Boolean algebra ko sirf tab use kar skte hai jab variable number less ho (1, 2, 3, 4)

(3) Final results same nhi hote, because approaches different hoti hai.

$\boxed{\text{Rules}}$

(i) complement :-

A complement = $\bar{A}$ or A' or (not A)

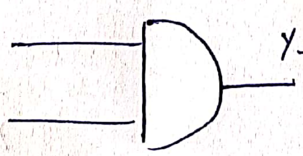
$*\ (A')' = A \qquad *\ 0' = 1 \qquad *\ 1' = 0$

(ii) AND I/P O/P

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

A

B



$Y = A \cdot B$

$$\left\{\begin{array}{l} A \cdot A = A \\ A \cdot 0 = 0 \\ A \cdot 1 = A \\ A \cdot A' = 0 \end{array}\right\}$$

(iii) OR

$A + A = A$

$A + 0 = A$

$\boxed{A + 1 = 1}$ Imp.

$A + A' = 1.$

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

(iv) Distributive law -

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

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

(v) Communitative law -

$A + B = B + A$

$A \cdot B = B \cdot A$

(vi) Asscociative Law -

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

(vii) De-Morgan's law -

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

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

Boolean algebra Example -

Ex 1 $F = AB + AB'$

$F = A[B + B']$ $\{B + B' = 1\}$

$\boxed{F = A}$

A	B	F = AB+AB'
0	0	0
0	1	0
1	0	1
1	1	1

F=A

Ex 2 $F = AB + AB'C + AB'C'$

$= A[B + B'C + B'C']$

Apply Distributive law $[A + A'B = A + B]$

$F = A[B + C + B'C']$

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

Again Distributive law

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

$F = A[B + 1]$

$= A[1]$

$F = 1.$

Ex3 $F = (A+B+C)(A+B'+C)(A+B+C')$.

Sol Let $X = A+B$

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

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

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

$F = (X+CC')(A+B'+C)$ $\boxed{CC' = 0}$

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

$= (A+B)(A+B'+C)$ {Distributive law} $\cancel{A+B(B'+C)}$

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

$= A + BB' + BC$

$= A + BC$

Ex 4 $G = (A+B)(A+B')(A'+B)(A'+B')$ {Distributive law}

$= (A+BB')(A'+BB')$

$= (A+0)(A'+0)$

$= AA'$

$= 0$.

19
-7

Binary operations

Addition

Input		output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0 (carry 1)

```
   1 1 1 1
    1 0 1 1 0   → augend
  + 1 1 0 1 1   → addent
  -----------
  1 1 0 0 0 1   → Addition
```

Subtraction

Input		output
A	B	Y
0	0	0
0	1	1 (Borrow of 1)
1	0	1
1	1	0

```
   ~~1~~0 1 0 1
  - 0 1 1 0 1
  -----------
    0 0 0 0 0
```

Multiplication

Input		output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

```
      0 1 1
    x 1 0 0
    -------
      0 0 0
    0 0 0 x
  0 1 1 x x
  ---------
  0 1 1 0 0
```

Division (NOT MUCH IMP)

X

Complement

r's complement {Radix complement}

(r-1)'s complement {Diminished Radix complement}

9's, 10's complement for Decimal number,

1's, 2's complement for Binary numbers.

9's complement -

Q → Find the 9's complement of 25678 ?

$$\begin{array}{r} 99999 \\ -25678 \\ \hline 74321 \end{array}$$

10's complement →

Q- Find the 10's complement of 25678 ?

$$\begin{array}{rr} & 99999 \\ & -25678 \\ \hline \text{9's complement} \rightarrow & 74321 \\ & +1 \\ \hline \text{10's complement} & 74322 \end{array}$$

Steps → 1) Pehle 9's complement find karenge.

2) uske baad usme 1 add kardenge

1's complement -

Q- Find the 1's complement of 10101011 ?

1 ko zero bna do, 0 ko 1.

1's complement of 10101011 = 01010100

2's complement -

Q. Find the 2's complement of 1011101 ?

1's complement of 1011101 = 0100010

+1

2's comenent 0100011

1) Pehle 1 complement nikalenge.
2) usme 1 add karenge by Binary addition.

- SUBTRACTION USING COMPLEMENTS -

Using 10's complement -

Q using 10's complement, Subtract 72532 - 3250

72532 → Minuend
- 3250 → Subtrahend

9's Complement of subtrahend

$$\begin{array}{r} 99999 \\ -\ 3250 \\ \hline 96749 \end{array}$$

10's complent

$$\begin{array}{r} 96749 \\ +1 \\ \hline 96750 \end{array}$$

Add 10's complement & minuend,

$$\begin{array}{r} 96750 \\ +\ 72532 \\ \hline 169282 \end{array}$$

If M>N (Matlab minuend bada hai subtrahend se to) to carry discard kardenge

So, Answer is 69282

2nd Rule - Agar M<N (Minuend chota hai subrahend se) to 10's complement lenge Answer ka or negetive sign laga denge.

Subraction using - 2's complement

Ques Subtract $\underbrace{1010100}_{\text{Minuent}}$ & $\underbrace{1000011}_{\text{Subtrahend}}$ using ~~xy~~ 2's complement

1's complement of 1000011 = 0111100
+ 1

2's complement = 0111101

Add 2's complement & minuend.

```
  0111101
+ 1010100
 10010001
```

Extra bit so, discard it, ~~[illegible]~~

Ques Subtract $\underbrace{1000011}_{\text{minuend}}$ & $\underbrace{1010100}_{\text{Subtrahend}}$ using 2's complement?

~~[illegible]~~

1's complement of 1010100 = 0101011
+ 1

2's complement = 0101100

Add 2's complement & minuend.

```
    0101100
    1000011
MSB 1101111
```

If no. of bits are same as minuend & subtrahend then & Most significant bit = 1 then Result is negetive and 2's complement

→ 1's Complement = 0010000

2's complement = 0010001

BOOLEAN ALGEBRA & LOGIC GATES

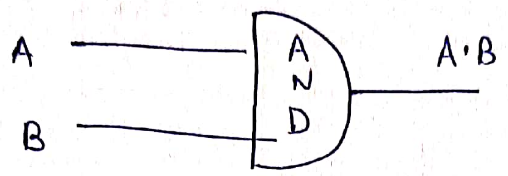
Postulates & theorems

	(a)	(b)
Postulate 2	$x+0=x$	$x\cdot 1=x$
Postulate 5	$x+x'=1$	$x\cdot x'=0$
Postulate 3	Communtative Law, $x+y=y+x$	$xy=yx$
Theorem 1	$x+x=x$	$x\cdot x=x$
Theorem 2	$x+1=1$	$x\cdot 0=0$
Theorem 3	$(x')'=x$	$(x')'=x$
Theorem 4	Associative $\cancel{x+(y+z)=(x+y)(y+z)}$ $x+(y+z)=(x+y)+z$	$x\cdot(y\cdot z)=(x\cdot y)\cdot z$
Postulate 4	Distributive $x\cdot(y+z)=xy+xz$	$x+(yz)=(x+y)(x+z)$
Theorem 5	De Morgan's Law, $(x+y)'=x'y'$	$(x\cdot y)'=x'+y'$
Theorem 6	$x+xy=x$	$x\cdot(x+y)=x$

Logic Gates

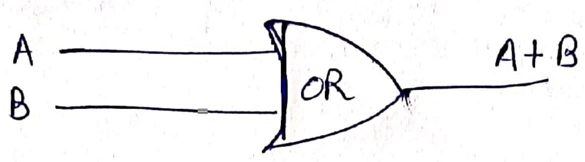
AND Gate

IC - 7808



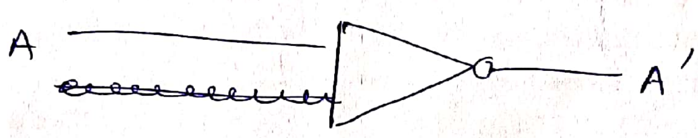
OR GATE

IC - 7432



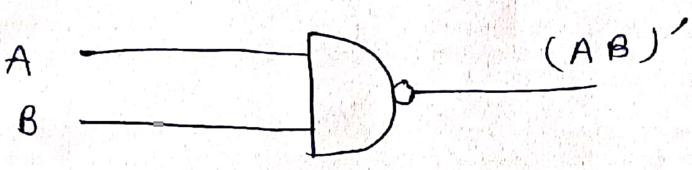
NOT Gate

IC - 7404



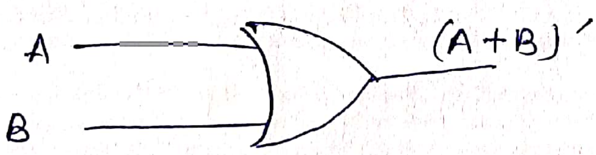
NAND GATE

IC - 7400



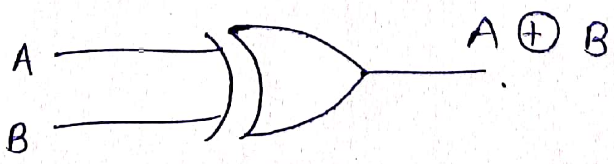
NOR GATE

IC - 7402



X-OR GATE -

IC - 7486



Number system

A set of values Repersent Quantity.

Name	Base	
Binary	2	(0, 1)
Octal	8	(0, 1, 2, 3, 4, 5, 6, 7)
Decimal	10	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
Hexadecimal	16	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)

Types of codes

- weighted
  - Binary
  - Decimal
  - octal
  - BCD
- Non weighted
  - Excess-3-code
  - Grey code

BINARY

* Binary digits (0 & 1) are called as bits.

Classification of Codes

Codes

Weighted code	Non weighted Code	Reflective codes	Sequential	Alpha numeric	Error detection & correcting Codes.
Positions ka weight hota hai	Positions ka koi weight nhi hota,	Self complementry	Ex: 8421	Ex: ASCII	Hamming code
Ex: Binary, 8421, 2421 etc	Ex - Grey - XS-3 etc	Ex - 2421 XS-3	XS-3 etc	etc	

9 → complement - 0
8 → -1
7 → -2

Binary coded Decimal {BCD} [known as 8421 code]

↳ Decimal number/digit Represented by 4 bit Binary number (0-9)

Decimal	BCD 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

BCD Addition

1) Sum ≤ 9 final carry = 0, answer → correct.
2) Sum ≤ 9 final carry = 1, answer is incorrect
   [To correct it add 0110 (6) in answer]
3) sum > 9, final carry = 0, answer is incorrect
   [To correct it add 0110 (6) in answer]

Ex 1 $(2)_{10} + (6)_{10}$ BCD addition

```
      0010
    + 0110
Sum   1000
```

Sum < 9
final carry = 0
So, answer is correct

Ex 2 $(3)_{10} + (7)_{10}$

```
      0011
    + 0111
Sum = 1010
```

Sum > 9
Final carry = 0

```
      1010
    + 0110
 0001 0000
```

$(1\ 0)_{10}$

Ex 4 $(57)_{10} + (26)_{10}$

```
      01010111  (57)
      00100110  (26)
9 >   0111 1101
           1101 > 9
          +0110
      1000 0011
         8    3
```

$(83)_{10}$

Ex 3. $(8)_{10} + (9)_{10}$

```
      ~~10000~~  1000
                 1001
Sum             ①0001
```

Final carry

Sum ≤ 9

```
      10001
    + 0110
  0001 0111
     1    7
```

$= (17)_{10}$

Excess-3 code addition -

Ex 1 $(2)_{10} + (5)_{10}$

		BCD		XS-3
$(2)_{10}$	→	0010	→	0101
$(5)_{10}$	→	0101	→	1000

```
  0101  — Excess-3
  1000  — Excess-3
  ----
× 1101  — Excess of 6.
```

So, subtract 3 from it

```
  1101
 -0011
 -----
  1000
```

Ex 2 $(27)_{10} + (39)_{10}$

		BCD		Excess-3
$(27)_{10}$	=	0010 0111	=	01011010
$(39)_{10}$	=	0011 1001	=	01101100

```
                  1 1 (1) → Final carry
Final carry → (0) 0101   [1010] — Grp 1
Grp 2 {           0110   [1100]
                  -------------
Incorrect ×       1100    0110
```

Grp 1 → final carry 1 add 0011 (3)

Grp 2 → final carry 0 ~~add~~ subtract (3)

```
  1100        0110
 -0011      + 0011
 -----      ------
  1001        1001
```

Gray code

* Reflected Binary code
* weighted code,
* cyclic code
* Minimum Error code.
* two successive values differ in only 1 bit
* Binary no. is converted into Grey code to Reduce switching operation

Decimal	Binary b_3	b_2	b_1	b_0	Gray g_3	g_2	g_1	g_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0
16	10	0	0	0	0	0	0	0

CONVERSION

Binary to decimal

Binary 1010.11

$1 \times 2^{-2} = 0.25 +$
$1 \times 2^{-1} = 0.50 +$
$0 \times 2^{0} = 0 +$
$1 \times 2^{1} = 2 +$
$0 \times 2^{2} = 0 +$
$1 \times 2^{3} = 8$

$(1010.11)_2 = (10.75)_{10}$

Binary to octal

8 4 2 1

Binary = 101010011.110100

make Group of 3's from Right

101 010 011 . 110 100

$(523.64)_8$

Binary to hexadecimal

Binary = 11111011101110010

make Group of 4's from Right

0001	1111	0111	0111	0010
1	F	7	7	2

$(1F772)_{16}$

	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
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

Decimal to Binary -

		Remainder
2	25	1
2	12	0
2	6	0
2	3	1
2	1	1
	0	

↑ Read

Binary number = 11001

$(25)_{10} = (11001)_2$

Decimal to octal -

		Remainder
8	973	5
8	121	1
8	15	7
8	1	1
	0	

$(973)_{10} = (1715)_8$

Decimal to hexadecimal

16	2297	9
16	143	15 → F (hexadecimal)
16	8	8
	0	

$(2297)_{10} = (8F9)_{16}$

Octal to Binary -

Octal = 347

011 100 111

$(347)_8 = (011100111)_2$

	4	2	1
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

Octal to Decimal -

123

$3 \times 8^0 = 3$

$2 \times 8^1 = 16$

$1 \times 8^2 = 64$

$(123)_8 = (83)_{10}$

octal to hexadecimal -

$\boxed{123}$ → octal number

↓ convert it into Binary By above table

$\boxed{001010011}$

↓ Make group of 4 bits from Right

$\boxed{0000 \quad 0101 \quad 0011}$

↓ Analyze in hexadecimal.

$\boxed{053}$

$(123)_8 = (53)_{16}$

Hexadecimal to Binary

1 A C 5

0001 1010 1100 0101

$(1AC5)_{16} = (0001101011000101)_2$

	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
(10) A	1	0	1	0
(11) B	1	0	1	1
(12) C	1	1	0	0
(13) D	1	1	0	1
(14) E	1	1	1	0
(15) F	1	1	1	1

Hexadecimal to decimal

2 C 9 B

(12) (11)

$11 \times 16^0 = 11 +$

$9 \times 16^1 = 144 +$

$12 \times 16^2 = 3072 +$

$2 \times 16^3 = 8192$

$(2C9B)_{16} = (11419)_{10}$

Hexadecimal to octal -

$\boxed{(1A)_{16}}$ Hexadecimal

↓ Convert it into Binary

$\boxed{00011010}$

↓ Make group of 3 bits

$\boxed{000\ 011\ 010}$

↓ Analyze no. from table

$\boxed{0\ 3\ 2}$

$(1A)_{16} = (032)_8$

Decimal to BCD

$(17)_{10}$

1 7

0001 0111

$(17)_{10} = (00010111)_{BCD}$

$(156)_{10}$

1 5 6

0001 0101 0110

$(156)_{10} = (000101010110)_{BCD}$

BCD to Decimal

(i) BCD = 10100

$\underbrace{0001}_{1}$ $\underbrace{0100}_{4}$

$(10100)_{BCD} = (14)_{10}$

(ii) BCD = 1001001

$\underline{0100}$ $\underline{1001}$

4 9

$(1001001)_{BCD} = (49)_{10}$

Excess-3 code (XS-3)

Decimal ⟶ 8421 code (BCD code) $\xrightarrow{\text{Add}}$ $\boxed{0011}$ $(3)_{10}$ Excess-3 (unweighted code) (self complimentry)

Excess-3 ko Excess-3 islie bolte hai kauki, Jis decimal ka Excess-3 code nikal rhe hote hai, usme add karte hai 3. to usme Excess ho jata {Bach jata hai}.

Decimal	BCD 8 4 2 1	XS-3
0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 1 0 0
2	0 0 1 0	0 1 0 1
3	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 1 1
5	0 1 0 1	1 0 0 0
6	0 1 1 0	1 0 0 1
7	0 1 1 1	1 0 1 0
8	1 0 0 0	1 0 1 1
9	1 0 0 1	1 1 0 0

$$\begin{array}{r} 0000 \\ +0011 \\ \hline 0011 \end{array}$$

XS-3 code for decimal

24

$$\begin{array}{lll} \text{BCD} & 0010 & 0100 \\ +3 & 0011 & 0011 \\ \hline & \underbrace{0101}_{5} & \underbrace{0111}_{7} \end{array}$$

Excess-3 code

Binary to Gray

Steps

1) Record the MSB as it as.
2) Add the MSB to next bit, Record the sum, neglect carry
3) Repeat the process

Q Convert $\boxed{1}$011 to Gray code

MSB (Most significant Bit)

+ + +

1 0 1 1

↓ ↓ ↓ ↓

1 1 1 0

Q

⊕ ⊕ ⊕

1 1 1 0

↓ ↓ ↓ ↓

$\boxed{1\ 0\ 0\ 1}$ Grey

Grey to Binary -

Steps

① Record the MSB as it is.

② Add MSB to next bit of Grey code, Record sum neglect carry.

③ Repeat process.

Gray 1 1 1 0

⊕ ⊕ ⊕

Binary $\boxed{1\ 0\ 1\ 1}$

Gray 1 0 0 1

⊕ ⊕ ⊕

Binary $\boxed{1\ 1\ 1\ 0}$

Sum of Products (SOP)

A	B	C	F	
0	0	0	0	m_0
0	0	1	0	m_1
0	1	0	1	m_2
0	1	1	0	m_3
1	0	0	1	m_4
1	0	1	1	m_5
1	1	0	1	m_6
1	1	1	1	m_7

total no. of combinations $= 2^n$

~~total~~

$= 2^3 = 8$

SOP mein 1 lete hai

So,

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

Standard & Canonical form (minterms)

$F(A,B,C) = m_2 + m_4 + m_5 + m_6 + m_7$

$= \Sigma m(2,4,5,6,7)$

Ques For Given truth table minimize SOP function.

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

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

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

$= B(1)$

$= B$

Ques $Y(A,B) = \Sigma m(0,2,3)$

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

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

$= \bar{B} + AB$

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

$= \bar{B} + A$

POS (Product of sum)

A	B	C	Y	
0	0	0	0	M_0
0	0	1	0	M_1
0	1	0	1	M_2
0	1	1	0	M_3
1	0	0	1	M_4
1	0	1	1	M_5
1	1	0	1	M_6
1	1	1	1	M_7

→ POS form is used when the output is "0"

$Y = \pi [M_0 M_1 M_3]$

$Y = \underbrace{(A+B+C)(A+B+\bar{C})(A+\bar{B}+\bar{C})}_{\text{Max terms.}}$

Ques For the given truth table to minimize the POS Expression-

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

$Y = \overbrace{(A+\bar{B})(\bar{A}+\bar{B})}^{\text{Max terms.}}$

Distributive law $[(A+B)(A+C) = A+BC]$

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

$= \bar{B} + 0 \quad \{A\bar{A} = 0\}$

$\boxed{Y = \bar{B}}$ — minifmal form

$Y = \pi(M_1, M_3)$

$Y = \pi[M(1,3)]$

SOP & POS form Example

SOP → Sum of Products

POS → Product of Sum

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

minterms denoted by m

maxterms denoted by M

min term ke lie sigma se denote karte hai

So, here $y = \Sigma m(0, 2, 3, 6, 7)$

Jo isme nhi hai like - 1, 4, 5 vo max terms honge

Max terms ke lie Pi se denote karte hai

$y = \Pi M(1, 4, 5)$

SOP form

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

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

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

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

$$= \bar{A}\bar{B}\bar{C} + B \qquad \text{Let } X = \bar{A}\bar{C}$$

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

$$y = X + B$$

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

Minimal SOP form

POS Form

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

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

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

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

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

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

$\boxed{Y = (B+\bar{C})(B+\bar{A})}$ Minimal POS form

****

~~Aim~~ **Minimal form to cannonical form**

$Y = A+B'C$ | By SOP (Sum of Products) |

↳ Here are 3 variables so, Each term must contain 3 variables.

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

dono ki value 1 hongi iske. Multiply karne se koi farak ni padega

isse bhi koi fark nhi padega.

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

$Y = ABC + AB\bar{C} + \boxed{A\bar{B}C} + AB\bar{C} + \boxed{\bar{B}CA} + \bar{A}\bar{B}C$

same.

Now every term has same no. of variable

$Y = \underset{(1,1,1)}{ABC} + \underset{(1,1,0)}{AB\bar{C}} + \underset{(1,0,1)}{A\bar{B}C} + \cancel{AB\bar{C}}\ \underset{(0,0,1)}{\bar{A}\bar{B}C}$

$= \Sigma M(1,5,7,8)$

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

By POS (Product of Sum)

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

↳ Here are total 3 variable so,
Every term must contain 3 variables.

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

↓

Add karne se change nhi hoga kyunki value 0 hoti hai

$= (A+B+C')(x + BB')$ Let $A'+C = x$

$= (A+B+C')(x+B)(x+B')$ ↓ Distributive Law

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

$\quad\;\; 0\,0\,1 \qquad\quad 1\,0\,0 \qquad\quad 1\,0\,1$

$= \pi(1, 4, 5)$

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

K-map {Karnaugh map}

For 1 variable

A \	0
0	

For 2-variable

$\text{cell} = 2^n \rightarrow n$ - variable number

So, cells = 4.

A \ B	0	1
0	0	1
1	2	3

For 3-variable

A \ BC	00	01	11	10
0				
1				

OR

AB \ C	0	1
00		
01		
11		
10		

For 4-variable

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

For 5-variable

1) Make 2 K-maps of 4-variable

A = 0

BC \ DE		$\bar{D}\bar{E}$ 00	$\bar{D}E$ 01	DE 11	$D\bar{E}$ 10
$\bar{B}\bar{C}$	00	0	1	3	2
$\bar{B}C$	01	4	5	7	6
BC	11	12	13	15	14
$B\bar{C}$	10	8	9	11	10

A = 1

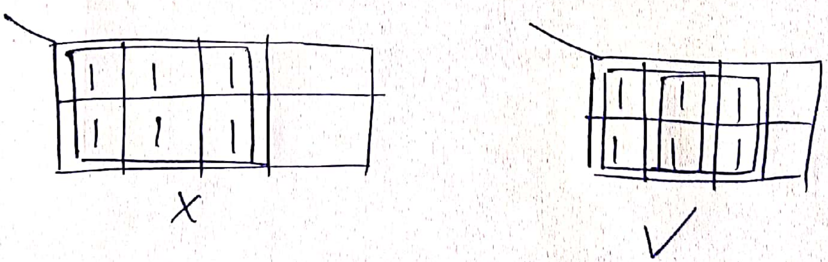
BC \ DE		$\bar{D}\bar{E}$ 00	$\bar{D}E$ 01	DE 11	$D\bar{E}$ 10
$\bar{B}\bar{C}$	00	16	17	19	18
$\bar{B}C$	01	20	21	23	22
BC	11	28	29	31	30
$B\bar{C}$	10	24	25	27	26

Rules to solve K-map -

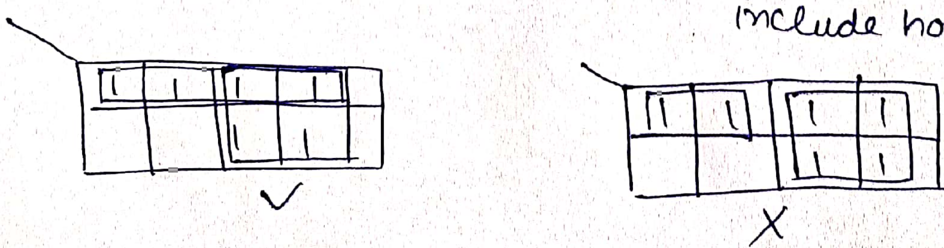
① Groups un cells ko include nhi karenge jinme value 0 hoti hai

② Groups horizontal ya vertical honge, pr diagnol nhi honge.

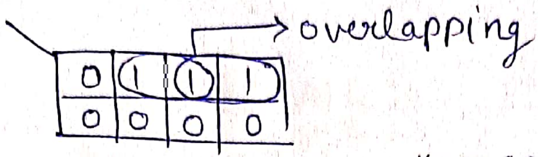
③ Groups must contain 1, 2, 4, 8, 16 .... (2^n) cells.



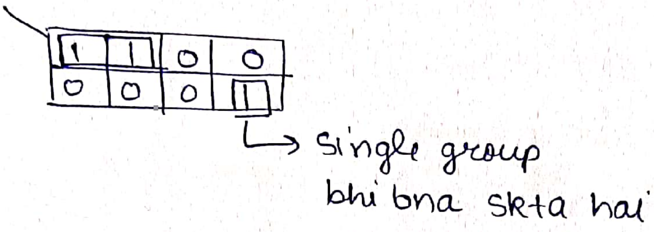
④ Har grp Bade se Bada hoga (i mean maximum cell include honge)



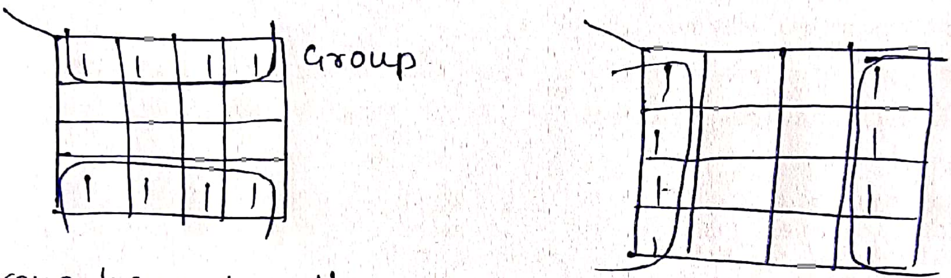
⑤ Groups overlap Kar skte hai



⑥ Har cell jisme one (1) hai Kam se Kam EK group mein zarroor hoga



⑦ Groups corners pr wrap around Kar skte hai



★★ Group hamesha 0th position se banana start Karenge.

Ques. $F(x, y, z) = \underbrace{\Sigma(3, 4, 6, 7)}$

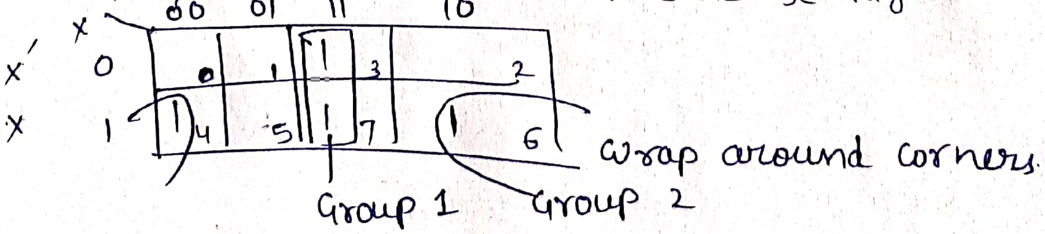
Make K-map — jo values given hai K-map mein 1 fill Kardo Baaki sb 0 Kardo

(FOR SOP'S)

Step 1

x \ yz	00	01	11	10
0	0_0	0_1	1_3	0_2
1	1_4	0_5	1_7	1_6

Step 2 — 0th position se Right ki trf jayenge. Phir neeche se Right ki trf.



Step 3

$$F = F_1 + F_2$$

(Group 1 Ka function) (Group 2 Ka function)

$$F_1 = yz \qquad F_2 = xz'$$

$$\boxed{F = yz + xz'}$$

Ques Simplify the Boolean function - using K-map -

$$F = A'B'C' + B'CD' + A'BCD' + AB'C'$$

Total variable = 4 (A, B, C, D)

So, 4 variable K-map banega

AB \ CD	C'D' 00	C'D 01	CD 11	CD' 10
A'B' 00	1 (0) F1	1 (1)	(3)	1 (2)
A'B 01	(4)	0 (5)	(7)	1 (6) F2
AB 11	(12)	0 (13)	(15)	(14)
AB' 10	1 (8)	1 (9)	(11)	1 (10) F3

~~A'B'C'~~

$F_1 = B'C'$

$F_2 = A'CD'$

$F_3 = B'CD'$

So, $\boxed{F = B'C' + A'CD' + B'CD'}$

Simplified.

Har term mein Har variable include karenge.

$$F = A'B'C'\underbrace{(D+D')}_{1} + (A+A')B'CD' + A'BCD' + AB'C'(D+D')$$

$$= \underbrace{A'B'C'D + A'B'C'D' + AB'CD' + A'B'CD' + A'BCD' + AB'C'D + AB'C'D'}_{\text{Fill 1 in K-map}}$$

POS Simplification

** POS mein hamesha 0 lenge

Ques. $F(A,B,C,D) = \pi(3,4,6,7,11,12,13,14,15)$.

↓

Sab mein zero fill kardo.

AB \ CD	(C+D) 00	(C+D') 01	(C'+D') 11	(C'+D) 10
(A+B) 00	0	1	0^{F_1} 3	2
(A+B') 01	0 4	5	0 7	0 6 F_2
(A'+B') 11	0 12	F_3 0 13	0 15	0 14
(A'+B) 10	8	9	0 11	10

$F_1 = C' + D'$ $\quad$ $F_2 = B' + D$ $\quad$ $F_3 = A' + B'$

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

Don't care conditions

Ex: $\underbrace{F(A, B, C)}_{\text{3 variable}} = \sum m(2, 3, 4, 5) + \sum d(6, 7)$

↳ don't care

A \ BC	00	01	11	10
0	$_0$	$_1$	1_3	1_2
1	1_4	1_5	X_7	X_6

Step 1 — $\sum m(2, 3, 4, 5)$ mein 1 fill karo

Step 2 — 6, 7 mein X fill karo

Step 3 : ~~D~~ agar X ko include karna zaroori hai, grp ko bada karne ke lie nhi to ignore karenge.

A \ BC	00	01	11	10
0	0_0	0_1	1_3	1_2 F_1
1	1_4	1_5 F_2	X_7	X_6

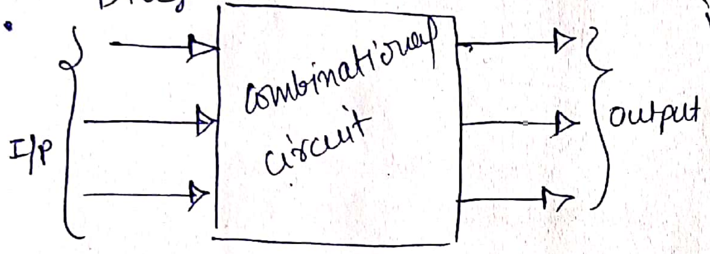
$F_1 = B$ $\qquad$ $F_2 = A$

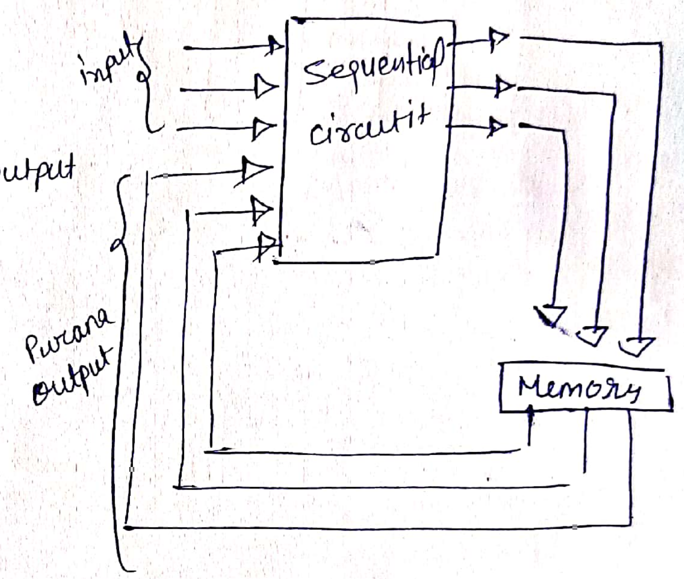
$F = F_1 + F_2 = A + B$

UNIT-3

Combinational circuits	Sequential circuits
• The output is only depends of Present Input	outputs depends on the present input as well as previous output or outputs
• Ex :- Adder, Decoder	Ex : Counter, Flip flops

• Diagram





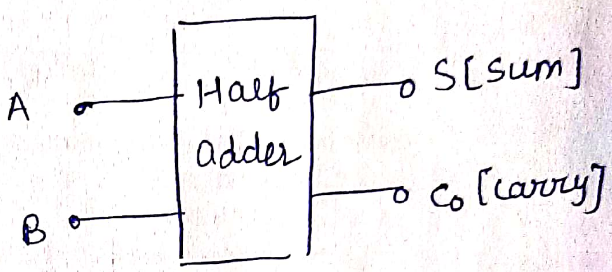
Combinational circuits	Sequential circuits
• No Feedback	Feedback loop is present
• output is independent of the previous state	output is dependent of both Previous & current state
• No memory Element	contains memory Elements (Flip flops). to store previous state

Scanned with OKEN Scanner

Half Adder -

* Used to Add single bit number
* Doesn't take carry from previous sum

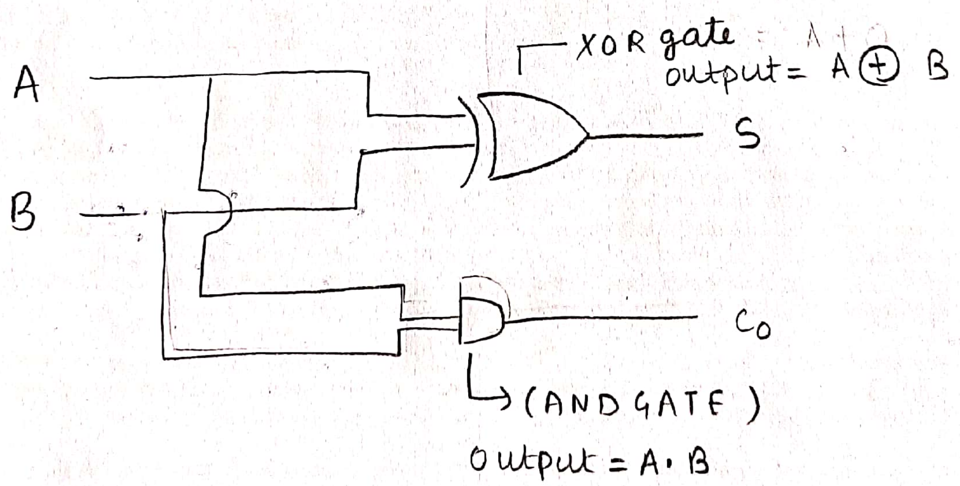
A	B	S	C_o
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



$S = A \oplus B$ → XOR

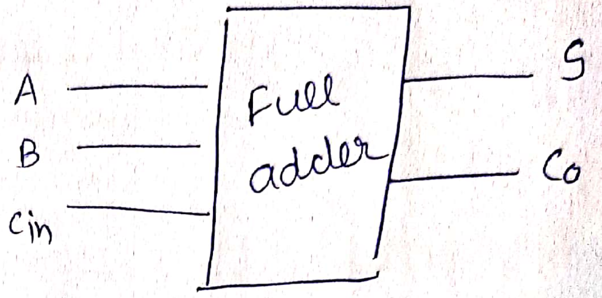
$C_o = A \cdot B$

{ $\oplus$ → XOR (isme alag bits hone pr sum 1 ata hai, or same input hone pr 0 }



FULL ADDER

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



K-map for S (sum

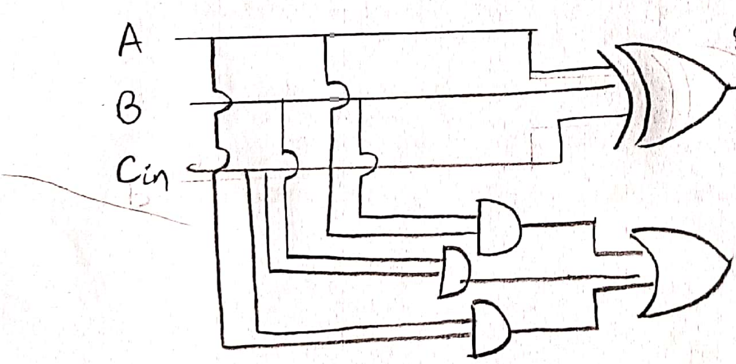
A \ BCin	00	01	11	10
0		1		1
1	1		1	

$S = A \oplus B \oplus C$

for Co (carry

A \ BCi	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$C_o = AB + BC_i + C_iA$

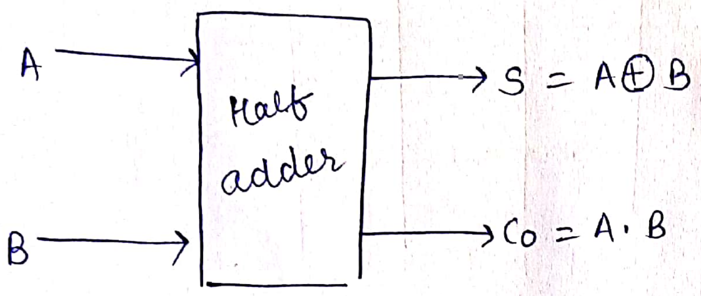


$S = A \oplus B \oplus C$

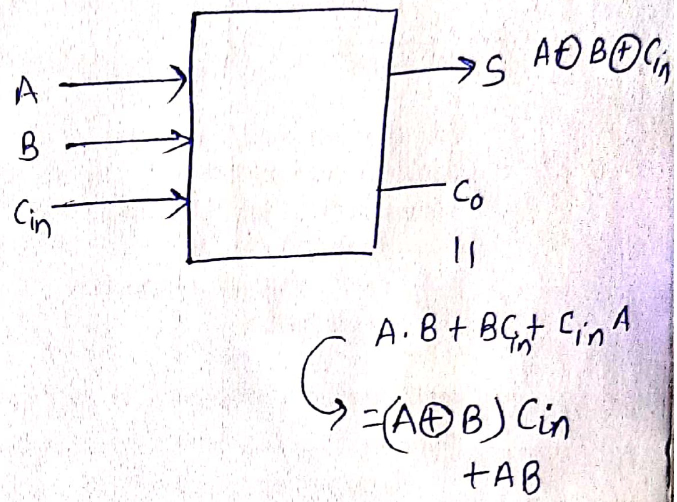
$AB + BCin + CinA$

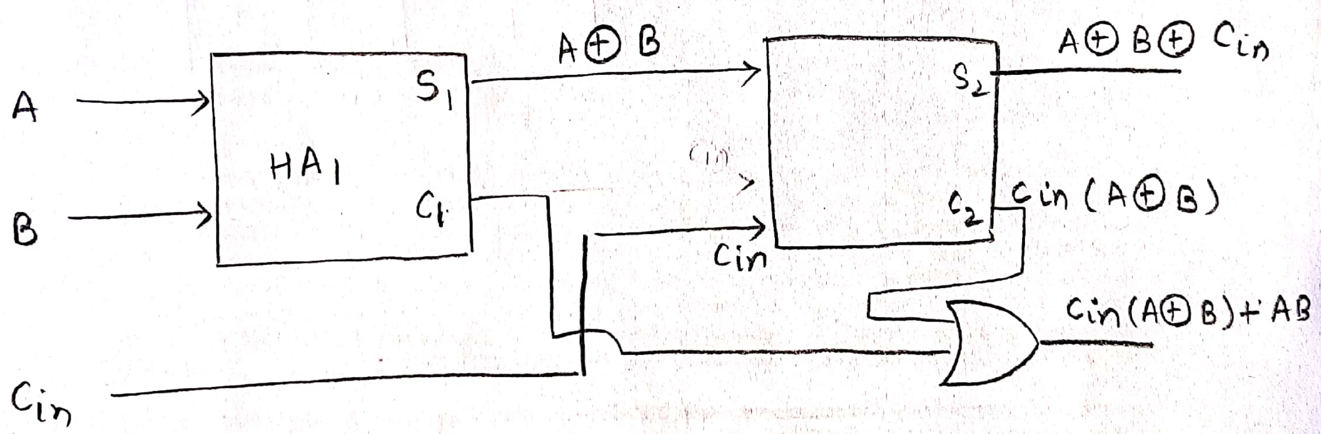
FULL ADDER USING HALF ADDER -

Half Adder



Full Adder

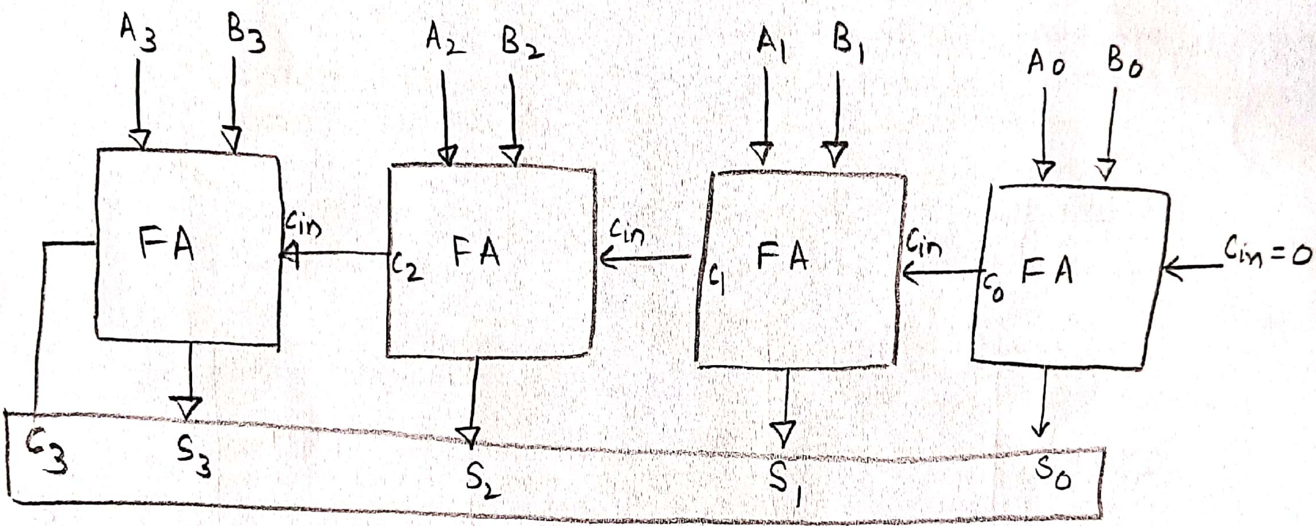




Four Bit Parllel Adder using Full adders -

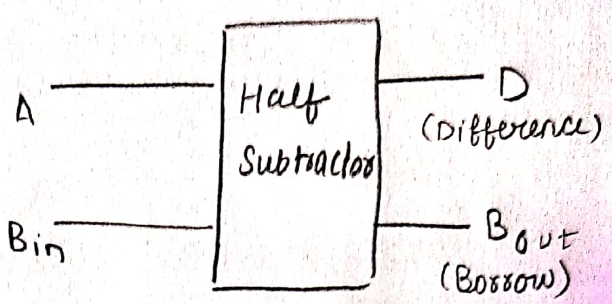
$$
\begin{array}{lcccccl}
 & C_{in} & C_{in} & C_{in} & C_{in} & & C_{in} = 0 \\
A = & 0 & A_3 & A_2 & A_1 & A_0 & \\
B = & 0 & B_3 & B_2 & B_1 & B_0 & \\
\hline
S = & \boxed{C_3} & S_3 & S_2 & S_1 & S_0 & \\
 & & C_3 & C_2 & C_1 & C_0 &
\end{array}
$$

$C_3 = 1$ or 0



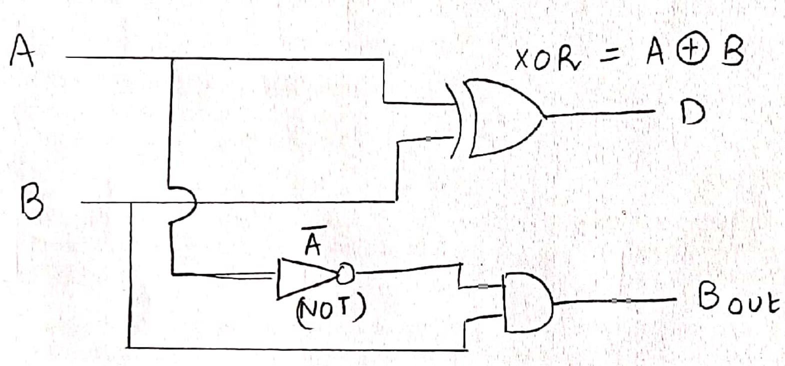
HALF SUBTRACTOR -

A	B	D	Bo
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



$$D = A \oplus B$$

$$B_o = \bar{A} B$$



Digital Logic Design

UNIT – I

★★ CONVERSIONS

Binary → Decimal, octal, Hexadecimal

Decimal → Binary, octal, Hexadecimal

Octal → Binary, Decimal, Hexadecimal

Hexadecimal → Binary, Decimal, octal

Decimal → BCD → Excess-3 ★★
Binary ⇄ Gray

Hamming code

theorem postulates of Boolean Algebra

canonical forms, SOP's, POS's.

Logic gates.

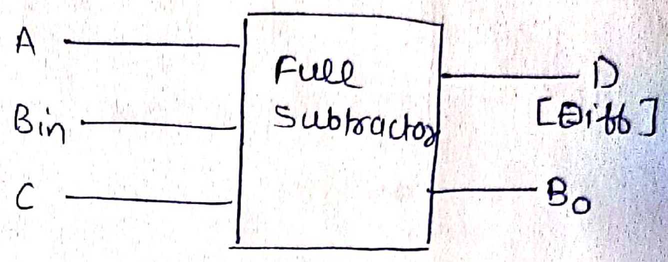
Subtraction, Addition of Binary numbers, subraction using (1, 2, 9, 10 complement)

UNIT – II

- K-map method (2, 3, 4, 5 - variable) {4 variable ★★★}
- Don't care Conditions
- Essential, Prime, Non Essential prime Implicants.

FULL SUBTRACTOR

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



A \ Bin C	00	01	11	10
0	$_0$	1_1	$_3$	1_2
1	1_4	$_5$	1_7	$_6$

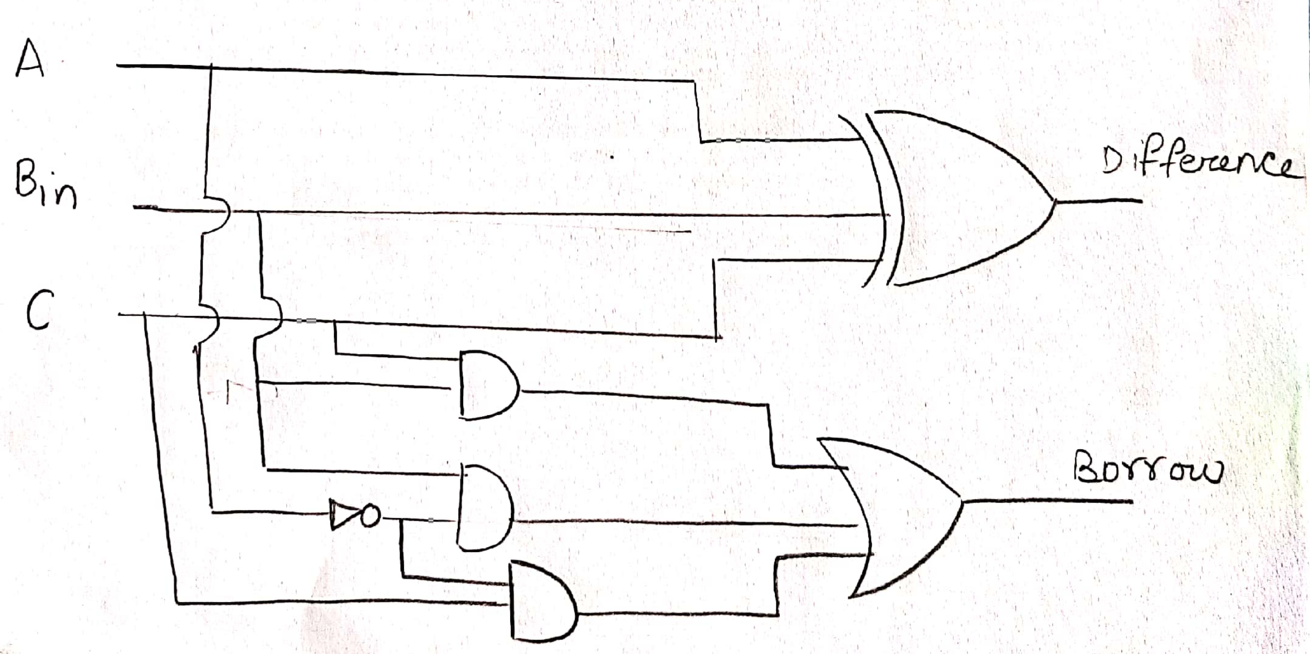
K-map for Difference (D)

$$D = A \oplus B \oplus C$$

A \ Bin C	00	01	11	00
0	0_0	1_1	1_3	1_2
1	0_4	0_5	1_7	0_6

Borrow

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

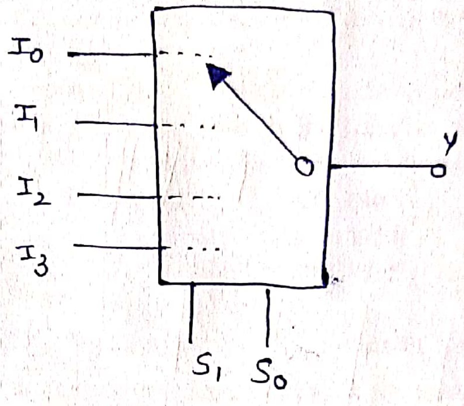


Multiplexers -

* Combinational circuit that selects Binary information. Kaafi inputs jo hote hai unme se 1 ko choose karke jo direct karta hai output lines ko.

* ~~Slowly~~ a Data selector

* Select line helps to choose the input variabe

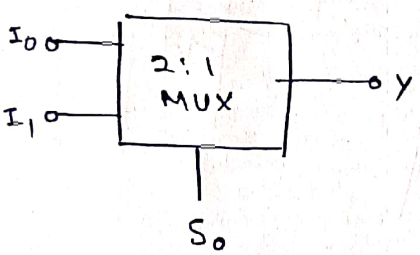


For 2:1 MUX

1 → select line

2 → Input

1 → 1 output



Relation between select lines & Inputs.

$n = 2^{m}$ → no. of select lines

n → no. of Input

$$\boxed{m = \log_2 n}$$

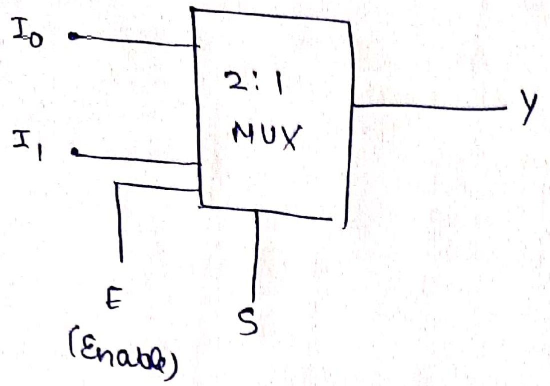
Advantages of Multiplexer

* Reduce no. of wires.
* Reduces cost & complexity of circuits.

Types of MUX

* 2:1 MUX
* 4:1 MUX
* 8:1 MUX
* 16:1 MUX
* 32:1 MUX

2:1 MUX

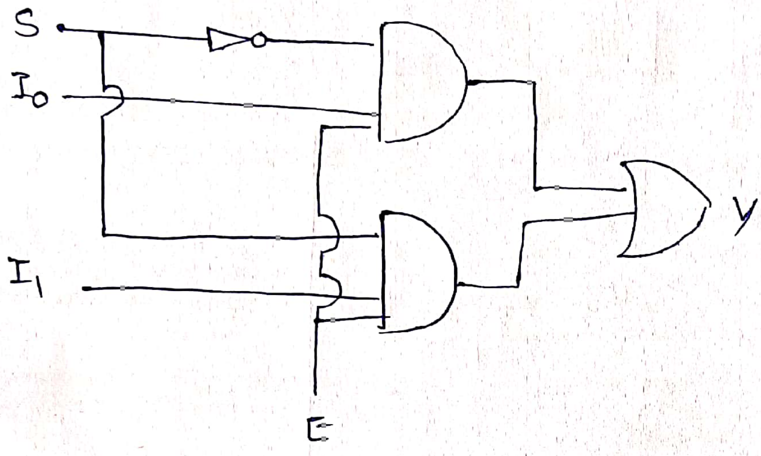


E	S	y
0	X	0
1	0	I_0
1	1	I_1

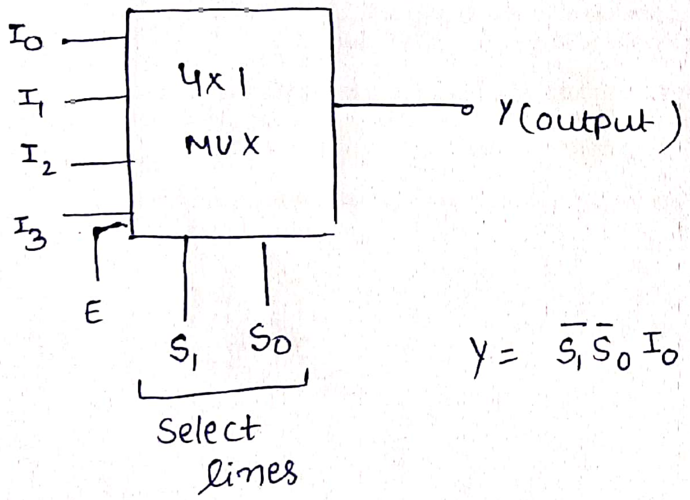
$$Y = E \cdot \bar{S} \cdot I_0 + E \cdot S \cdot I_1$$

$$Y = E(\bar{S} I_0 + S I_1)$$

Implementation

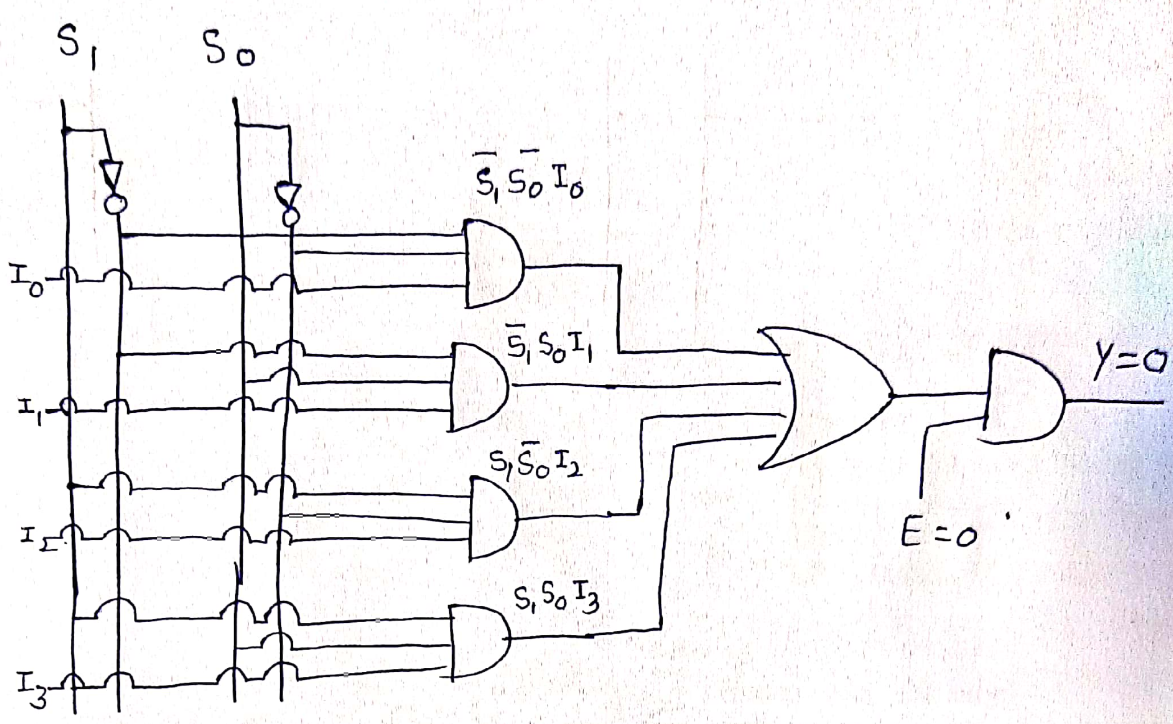


4:1 MUX

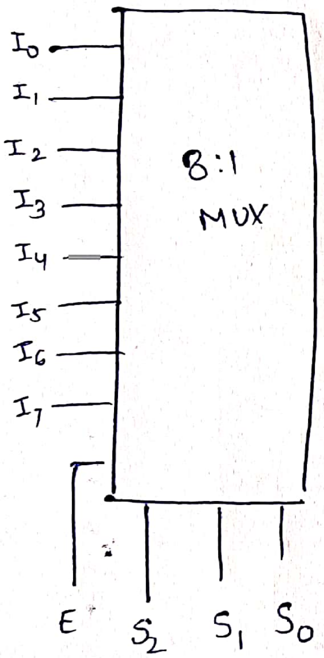


S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$$Y = \bar{S_1}\bar{S_0} I_0 + \bar{S_1} S_0 I_1 + S_1 \bar{S_0} I_2 + S_1 S_0 I_3$$



8 : 1 MUX



S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

$$Y = \bar{S}_2\bar{S}_1\bar{S}_0 I_0 + \bar{S}_2\bar{S}_1 S_0 I_1 + \bar{S}_2 S_1 \bar{S}_0 I_2 + \bar{S}_2 S_1 S_0 I_3$$
$$+ S_2\bar{S}_1\bar{S}_0 I_4 + S_2\bar{S}_1 S_0 I_5 + S_2 S_1 \bar{S}_0 I_6 + S_2 S_1 S_0 I_7$$