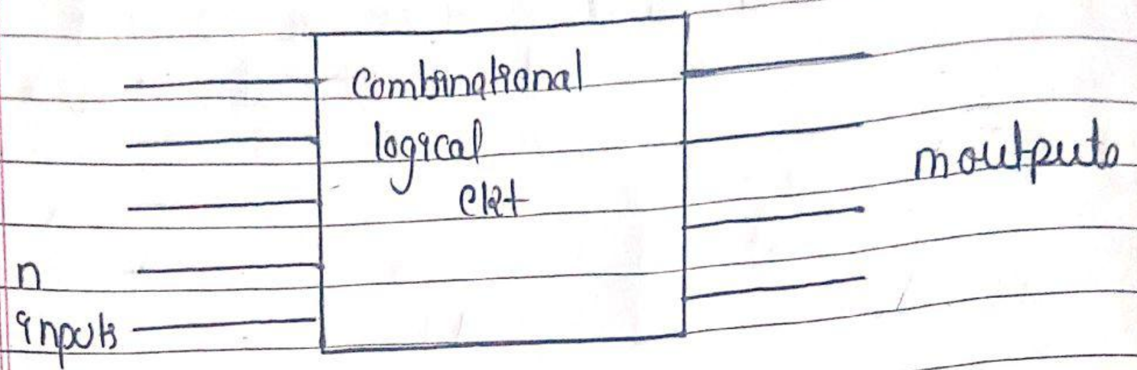
Chapter: 5

~~Combinational logic~~

Combinational logic



→ It consists of 'n' no of inputs and 'm' no of outputs.

→ At any time o/p value solely depends upon the inputs.

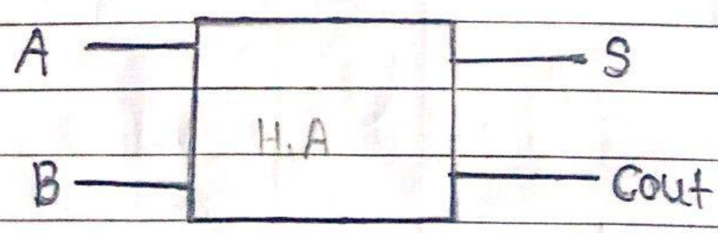
Design Procedure

1) Question is given to you.
2) From que you should find the number of i/ps 'n' and no of of outputs m.
3) Draw the truth table.
4) Simplify the outputs using K-Map.
5) Draw the logic diagram.

a) Adder

I) Half adder

It is combination ckt that is designed to add '2' bit only.



Truth table

i/p		o/p	
A	B	S	Cout (carry)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

K-map for Sum term (S)

A \ B	0	1
0		1
1	1	

for Cout

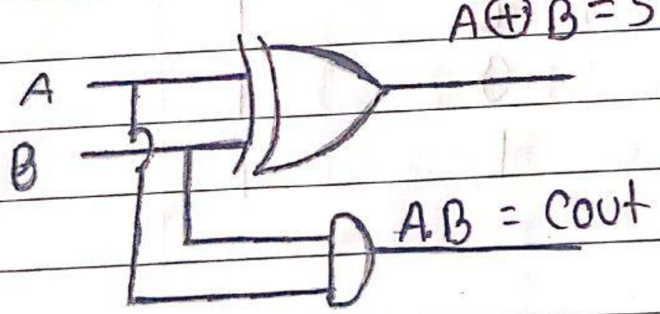
A \ B	0	1
0		
1		1

$Cout = A * B$

$S = A \oplus B$

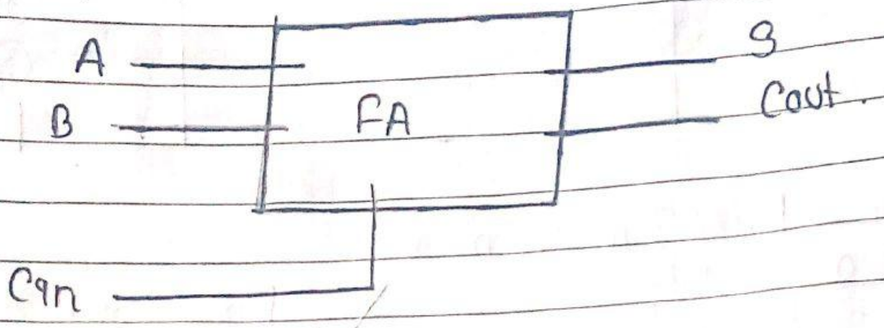
$= A\overline{B} + B\overline{A}$

Realization of Combination logic



<2> **Full adder:**

Full adder is a combinational ckt that is used to add 3 bit.



Truth table:

A	B	Cin	S	Cout
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

kMap For S

A \ B Cin	00	01	11	10
0		1		1
1	1		1	

$$S = A\overline{B}\,\overline{Cin} + \overline{A}B\,Cin + AB\,Cin + \overline{A}B\,\overline{Cin}$$

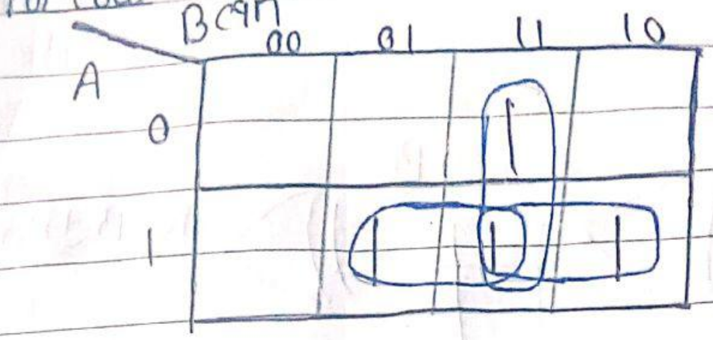
Scanned with CamScanner

$$= A(\overline{B}\,\overline{C_{in}} + BC_{in}) + \overline{A}(\overline{B}C_{in} + \overline{C_{in}}B)$$

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

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

For Cout



$$C_{out} = AC_{in} + AB + BC_{in}$$

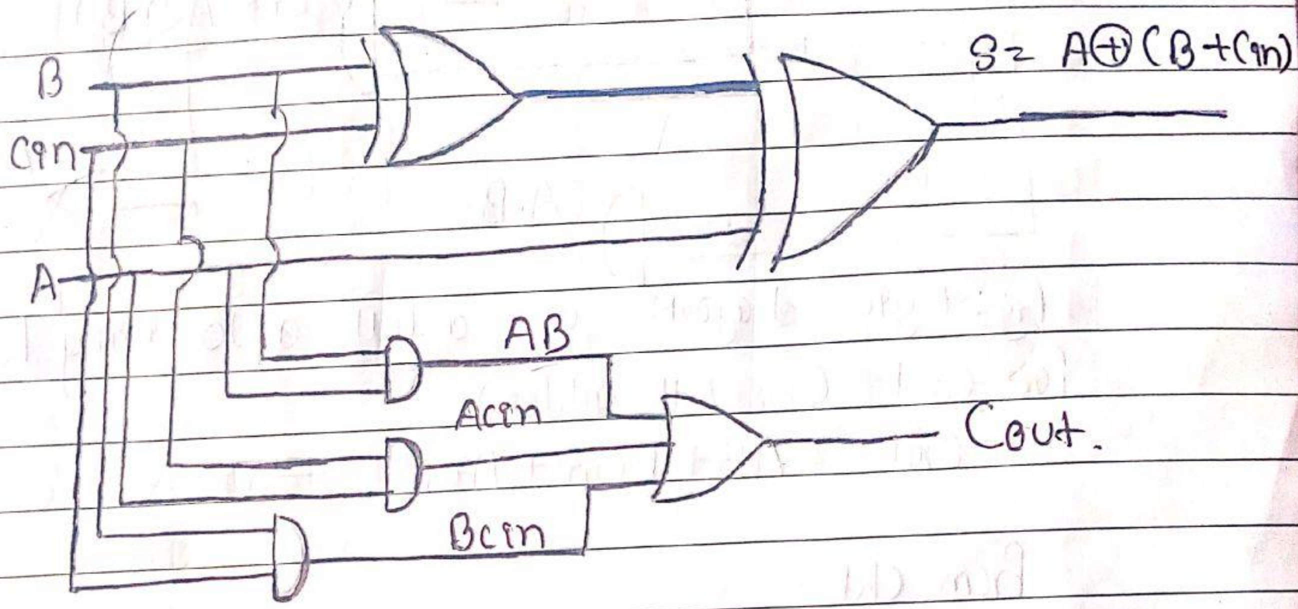


fig: Logic Diagram.

Construct a full adder using the half adder.

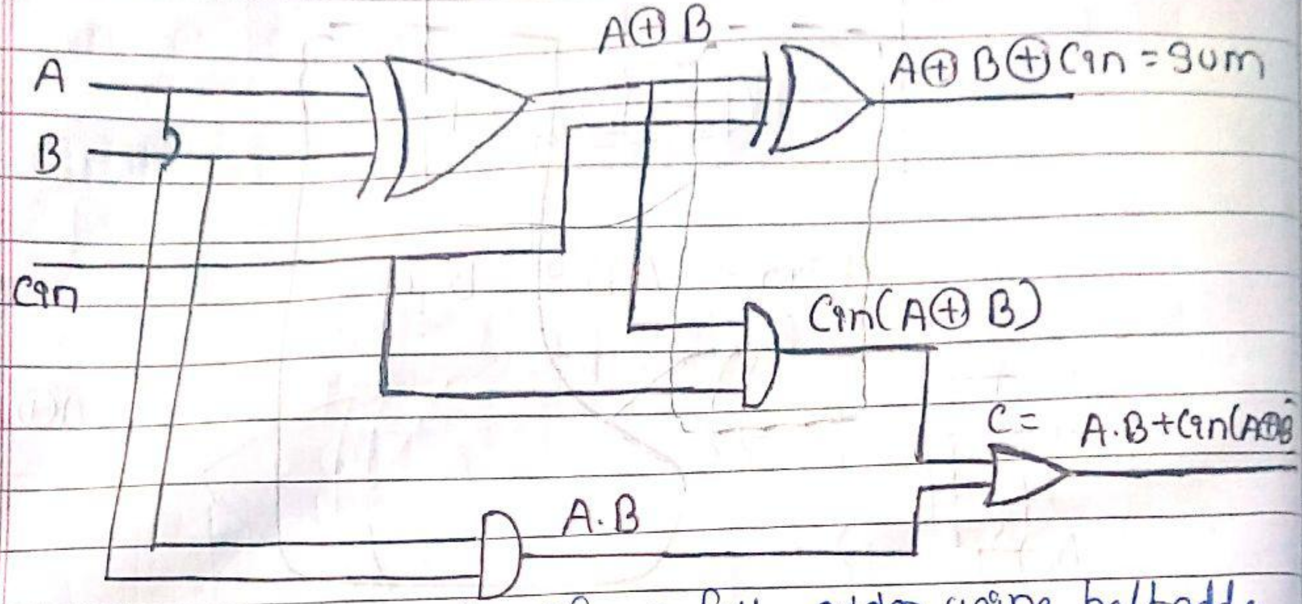


fig: logic diagram of a full adder using half adder

For Cout (for full adder)

$$Cout = AB + BCin + ACin \quad \text{---} (1)$$

From ckt,

$$Cout = A.B + Cin(A \oplus B)$$

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

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

From (i),

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

$$= AB + ABCin + \bar{A}BCin + ABCin + A\bar{B}Cin$$

$$= AB(1 + Cin) + A\bar{B}Cin + \bar{A}BCin$$

$$= AB + A\bar{B}Cin + \bar{A}BCin.$$

Hence verified

Subtracter

It is a combination of logic used to subtract two gates.

It is of two types;

1) Half ~~adder~~ Subtractor

It subtracts two bits.

Truth table

A	B	D	b
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

D = Difference
b = borrow.

K-Map for D

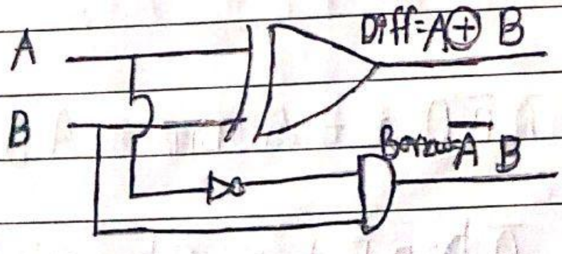
A \ B	0	1
0		1
1	1	

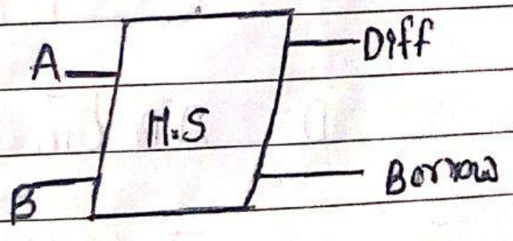
KMap for b

A \ B	0	1
0		1
1		

$D = A\overline{B} + \overline{A}B$
$= A \oplus B$

$b = \overline{A}B$





Half subtracter

2) Full Subtractor

It Subtracts 3 bits.

no of i/p = 3

no of o/p = 2

Truth table

A	B	bin	D	bout
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

Kmap for D

A \ B bin	00	01	11	10
0		1		1
1	1		1	

$$D = A\overline{B}\,\overline{b_{in}} + \overline{A}\,\overline{B}\,b_{in} + AB\,b_{in} + \overline{A}B\,\overline{b_{in}}$$

$$= \overline{B}\,(A\overline{b_{in}} + \overline{A}\,b_{in}) + B(A\,b_{in} + \overline{A}\,\overline{b_{in}})$$

$$= \overline{B}\,(A \oplus b_{in}) + B\,(\overline{A \oplus b_{in}})$$

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

$$= A \oplus B \oplus b_{in}$$

K-Map for bout

A \ B bin	00	01	11	10
0		1	1	1
1			1	

$$bout = \overline{A}\,bin + B\,bin + \overline{A}B$$

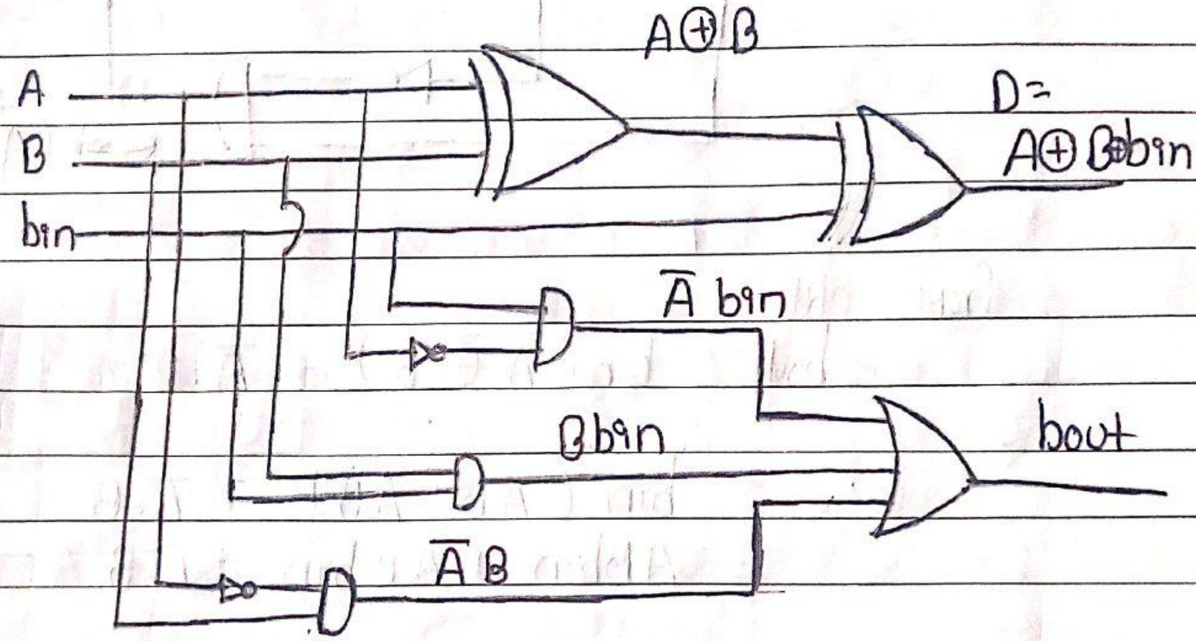
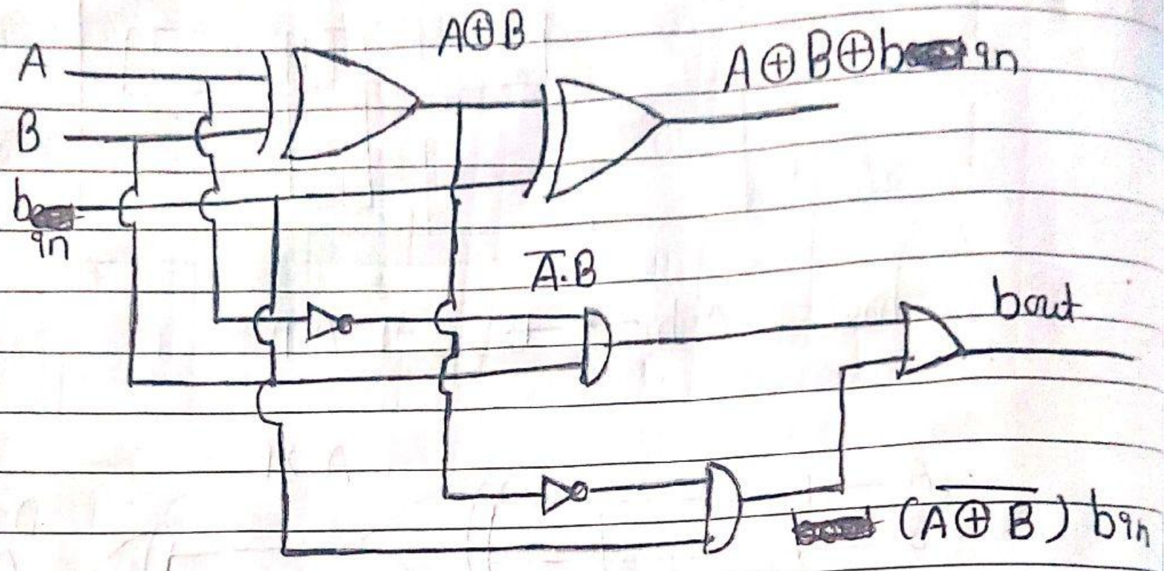


fig: Logic Diagram

Draw a full subtractor using two half subtractor & verify your answer.



From ckt,

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

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

$$= AB b_{in} + \bar{A}\bar{B} b_{in} + \bar{A}.B \quad \text{—— (i)}$$

We have,

$$b_{out} = \bar{A} b_{in} + B b_{in} + \bar{A}B$$

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

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

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

$$= \bar{A}B(1 + b_{in}) + AB b_{in} + \bar{A}\bar{B} b_{in}$$

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

Code Conversion:

17 Decimal to BCD to binary.

Decimal	Binary	BCD (PQRS)
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	XXXX	XXXX
11	XXXX	XXXX
12	XXXX	XXXX
13	XXXX	XXXX
14	XXXX	XXXX
15	XXXX	XXXX

KMap for P

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	X	X	X	X
10	1	1	X	X

KMap for Q

AB \ CD	00	01	11	10
00				
01	1	1	1	1
11	X	X	X	X
10			X	X

$Q = B$

$P = A$

K Map for 'R'

AB \ CD	00	01	11	10
00			1	1
01			1	1
11	X	X	X	X
10			X	X

$R = C$

K Map for S

AB \ CD	00	01	11	10
00		1	1	0
01		1	1	
11	X	X	X	X
10		1	X	X

$S = D$

27 Decimal or BCD to gray code;

Decimal	BCD (X Y Z W)	graycode (A B C D)
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	~~1001~~ 1101

K map for A

XY \ ZW	00	01	11	10
00				
01				
11	X	X	X	X
10	1	1	X	X

$A =$ [redacted] X

Kmap for B

XY \ ZW	00	01	11	10
00				
01	1	1	1	1
11	X	X	X	X
10	1	1	X	X

$B = X + Y$

Kmap for C

XY \ ZW	00	01	11	10
00			1	1
01	1	1		
11	X	X	X	X
10			X	X

Kmap for D

XY \ ZW	00	01	11	10
00		1		1
01		1		1
11	X	X	X	X
10		1	X	X

$C = Y\overline{Z} + \overline{Y}Z$
$= Y \oplus Z$

$D = \overline{Z}W + Z\overline{W}$
$= Z \oplus W$

Q7 Octal to BCD

Octal	binary A B C	BCD P Q R S
0	0 0 0	0 0 0 0
1	0 0 1	0 0 0 1
2	0 1 0	0 0 1 0
3	0 1 1	0 0 1 1
4	1 0 0	0 1 0 0
5	1 0 1	0 1 0 1
6	1 1 0	0 1 1 0
7	1 1 1	0 1 1 1

KMap for 'P'

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

$P = 1$

KMap for 'Q'

A \ BC	00	01	11	10
0	D	D	D	D
1	1	1	1	1

$Q = A$

KMap for R

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

$R = B$

K-Map for S

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

$S = C$

(4) 4 bit BCD to excess-3

	BCD A	B	C	D	excess-3 P	Q	R	S
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

KMap for -P

AB \ CD	00	01	11	10
00				
01		1	1	1
11	X	X	X	X
10	1	1	X	X

K-Map for Q

AB \ CD	00	01	11	10
00		1	1	1
01	1			
11	X	X	X	X
10		1	X	X

$P = A + BD + BC$

$Q = \overline{B}D + \overline{B}C + B\overline{C}\overline{D}$

Parity generator and checker

3 bit generator & 3 bit checker

For even parity

$\boxed{\text{3 bit generator}}$

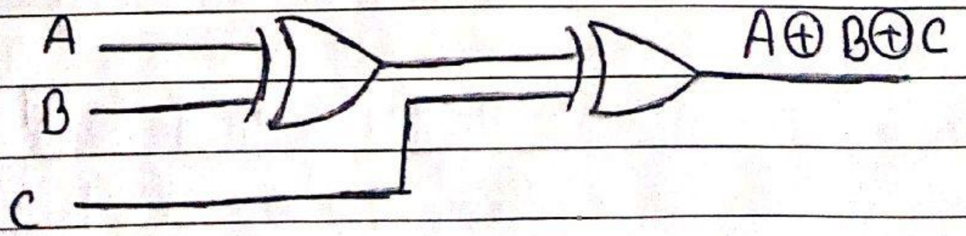
A	B	C	EP
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

K-map

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

$P = A \oplus B \oplus C$

logic ckt;

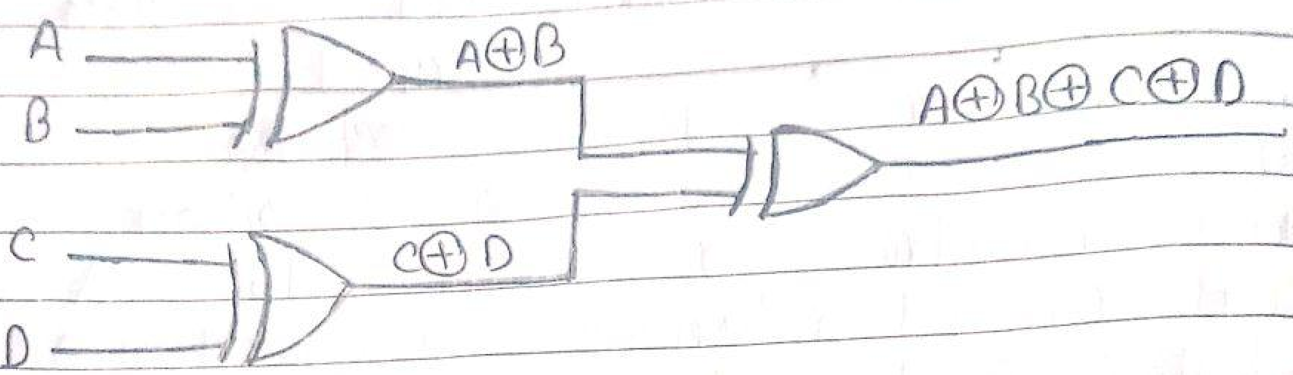


4 bit even parity checker

	A	B	C	D	Checker (ch)
0)	0	0	0	0	0
1)	0	0	0	1	1
2)	0	0	1	0	1
3)	0	0	1	1	0
4)	0	1	0	0	1
5)	0	1	0	1	0
6)	0	1	1	0	0
7)	0	1	1	1	1
8)	1	0	0	0	1
9)	1	0	0	1	0
10)	1	0	1	0	0
11)	1	0	1	1	1
12)	1	1	0	0	0
13)	1	1	0	1	1
14)	1	1	1	0	1
15)	1	1	1	1	0

AB \ CD	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

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



If o/p C is equal to 1, the receiver will know that the bits are changed during transmission i.e bits are in error if C=1.

If 'P' is connected the parity checker ckt becomes parity generator.

2(b) Design a circuit of a 3-bit parity generator and the circuit of 4-bit parity checker for odd parity.

Ans <u>For odd parity</u>

3-bit parity generator

A	B	C	P
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

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

$$P = \cancel{A \oplus B \oplus C}\ \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}C + AB\bar{C}$$

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

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

$$= \overline{A \oplus (B \oplus C)} = A \odot (B \oplus C)$$

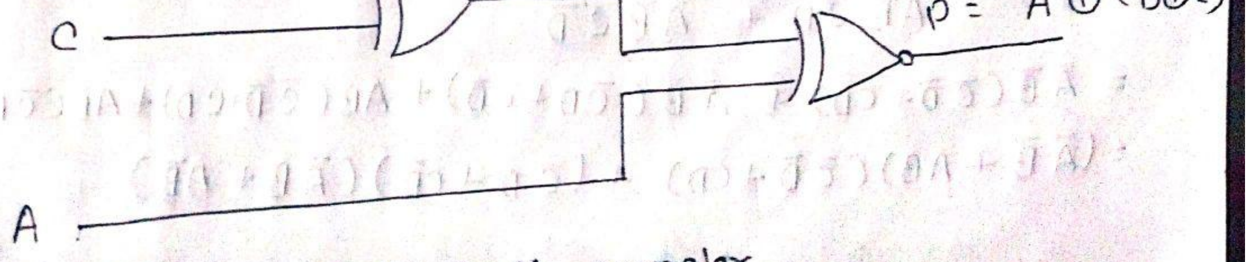


fig: 3-bit parity generator.

4-bit odd parity checker

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

AB \ CD	00	01	11	10
00	1	0	1	0
01	0	1	0	1
11	1	0	1	0
10	0	1	0	1

$$ch = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + AB\bar{C}\bar{D} + ABCD + A\bar{B}\bar{C}D + A\bar{B}C\bar{D}$$

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

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

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

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

$$= \overline{(A \oplus B) \oplus (C \oplus D)}$$

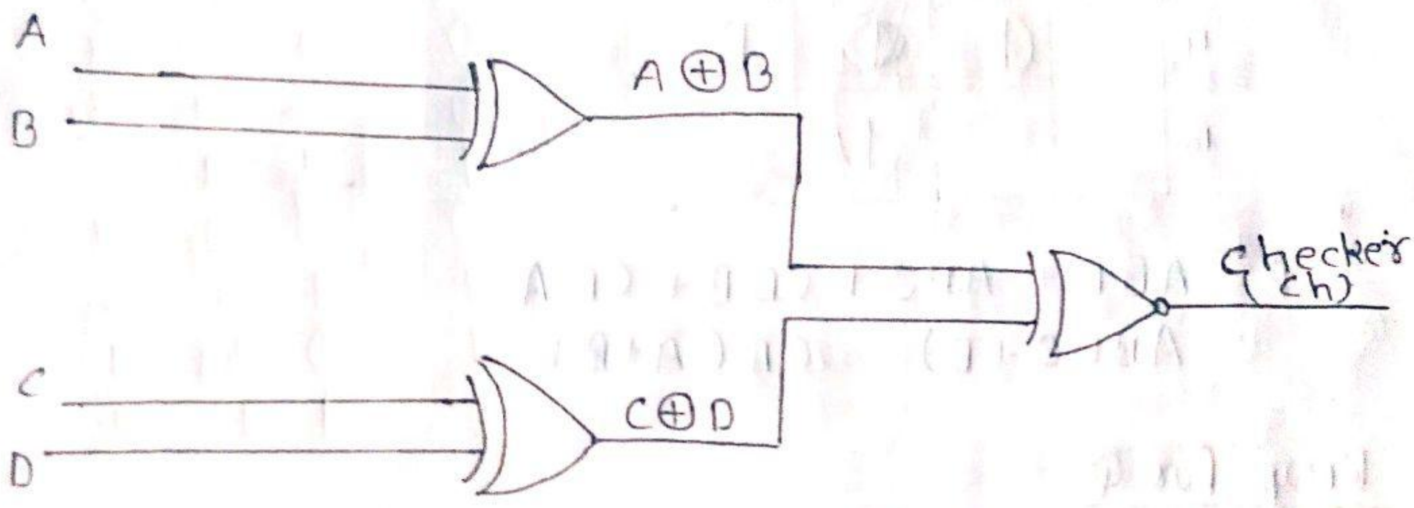


fig: circuit diagram of 4-bit odd parity checker