Combinational Circuit

The Digital system/circuits are one of the following types :-

(i)- Combinational circuits

(ii)- Sequential Circuits

→ A Combinational circuit may be defined as a logic circuit the output of which depends only upon the combination of the inputs.

→ The output does not depend on the past value of the output or inputs. Therefore combinational circuit donot needs any memory

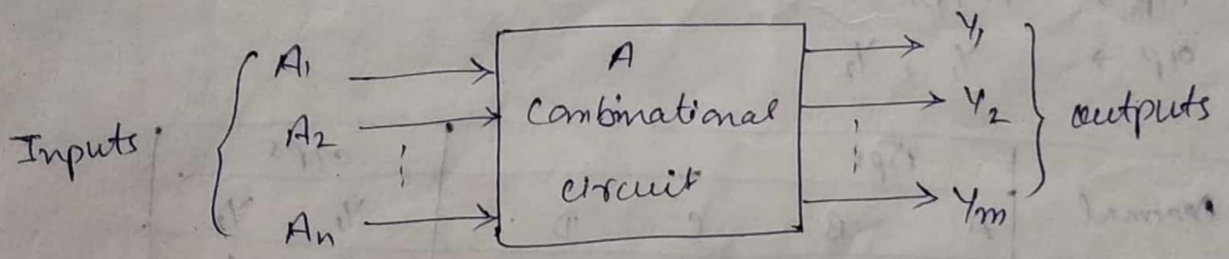


fig:- Block diagram of a Combination ckt.

→ A combinational ckt ~~many~~ can have a no of i/ps and a no of o/ps.

→ The above fig. have n i/p lines and m o/p lines. and between the i/p & o/p logic gates are connected. and hence ~~logic~~ g combinational ckts basically consists of logic gates.

Combinational Logic Design Procedure :-

(1) → Identify i/ps & o/ps.

(2) → Construct Truth Table.

(3) → write logic expression in SOP or POS.

(4) → Simplify logical expression. (using k - map)

(5) → Implement logical circuit.

Q→ A ckt has four i/ps and Two o/ps. One of the o/p is high when majority of i/ps are high. The second o/p is high only when all i/ps are same type. Design the combinational circuit.

Solⁿ→

① → i/p → A, B, C, D

o/p → Y_1, Y_2

② →

Decimal	i/ps				o/ps	
	A	B	C	D	Y_1	Y_2
0	0	0	0	0	0	1
1	0	0	0	1	0	0
2	0	0	1	0	0	0
3	0	0	1	1	0	0
4	0	1	0	0	0	0
5	0	1	0	1	0	0
6	0	1	1	0	0	0
7	0	1	1	1	1	0
8	1	0	0	0	0	0
9	1	0	0	1	0	0
10	1	0	1	0	0	0
11	1	0	1	1	1	0
12	1	1	0	0	0	0
13	1	1	0	1	1	0
14	1	1	1	0	1	0
15	1	1	1	1	1	1

(3)→ $Y_1 = \bar{A}BCD + A\bar{B}CD + AB\bar{C}D + ABC\bar{D} + ABCD$

$Y_2 = \bar{A}\bar{B}\bar{C}\bar{D} + ABCD$

(4)→ Y_1

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

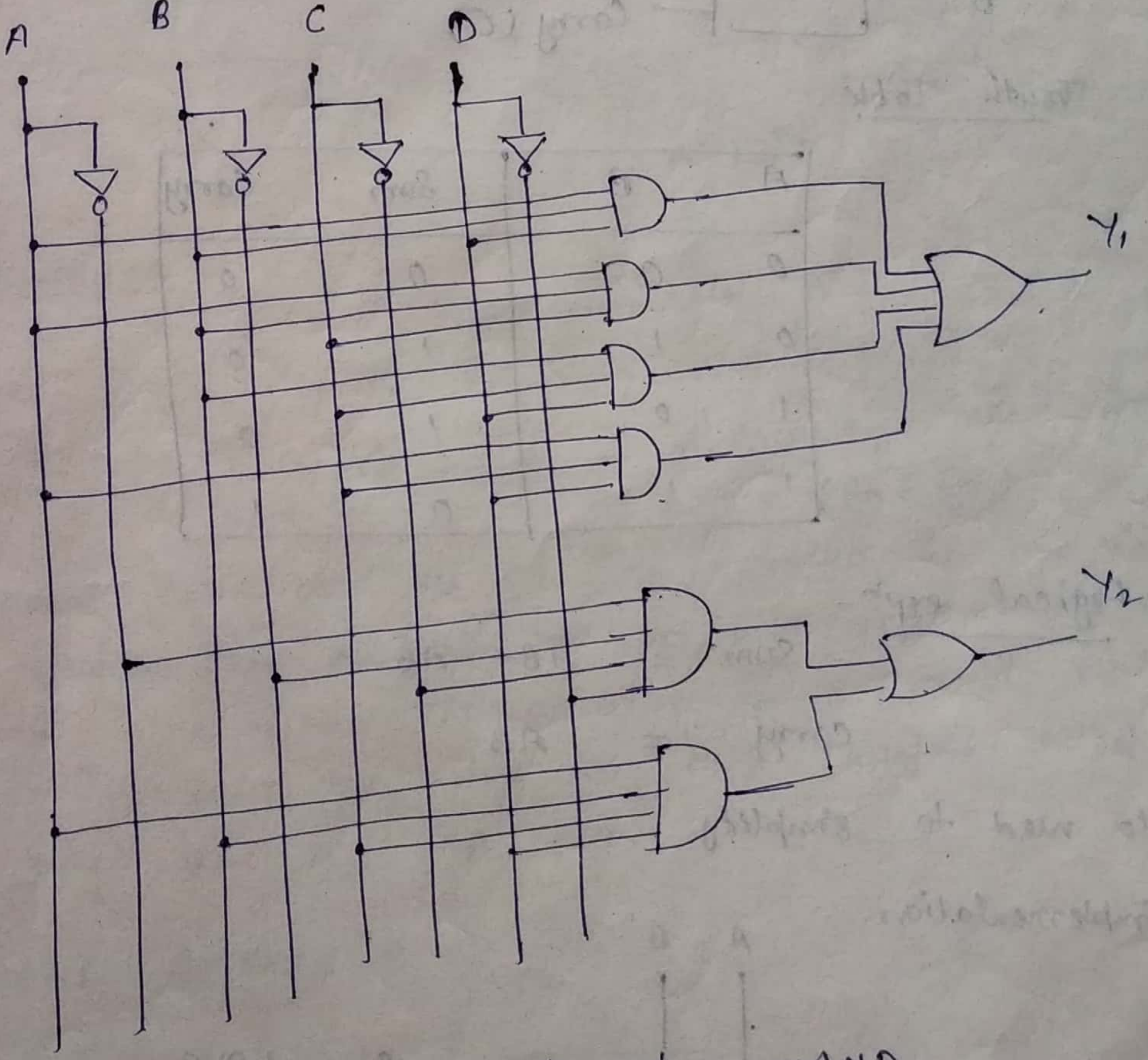
Y_2

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

$Y_2 = \bar{A}\bar{B}\bar{C}\bar{D} + ABCD$

$Y_1 = ABD + ABC + BCD + ACD$

(5)→



→ Draw This ckt using NAND

Classification of Combinational ckts :-

(i) - Binary & BCD adder ckt

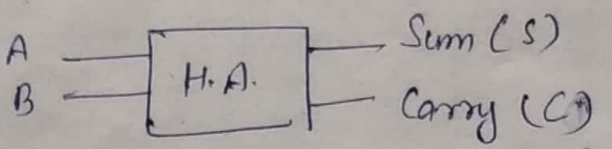
(ii) - Binary & BCD subtractor ckt.

(iii) - MUX , DeMUX, ENCODER, Decoder

(iv) Comparator & Code converter.

* Half Adder :-

Step ① → Two i/ps & Two o/ps.



② → Truth Table

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

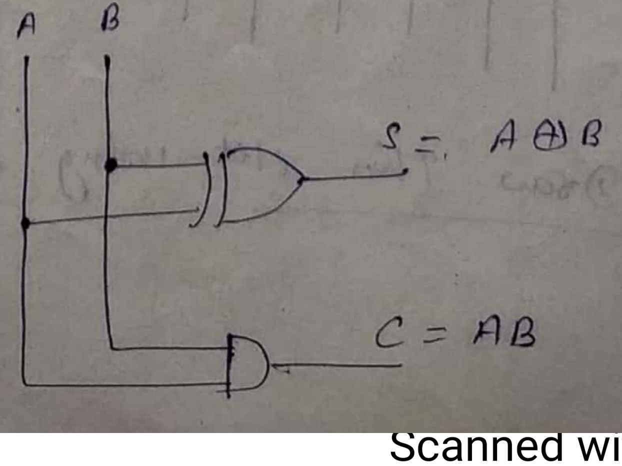
③ → Logical exp^n -

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

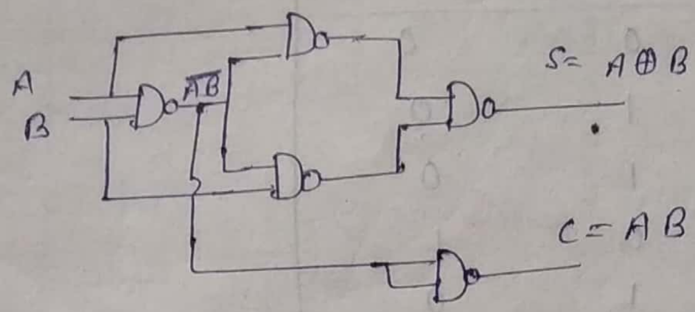
carry = AB

④ → No need to simplify.

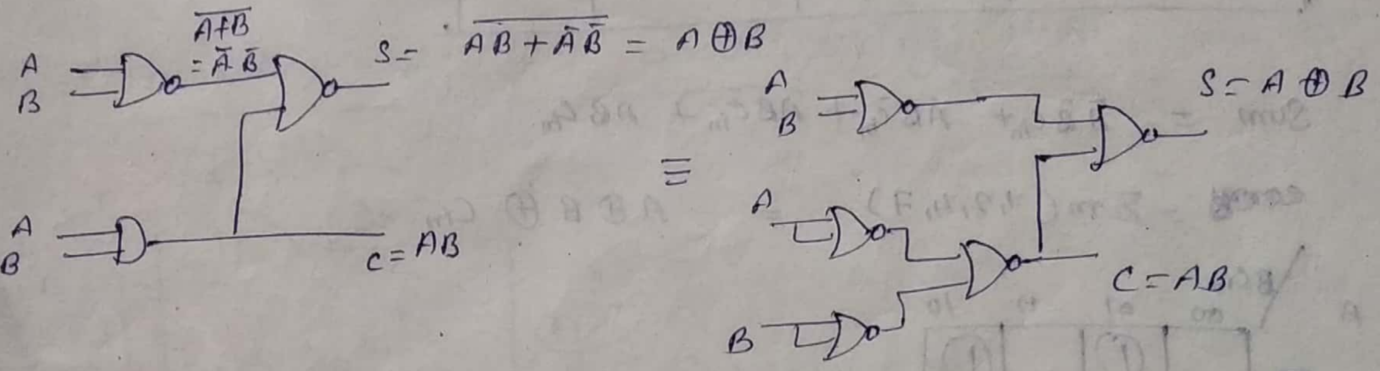
⑤ → Implementation



Implement using NAND gate :-



implement using NOR :-


A
B
$\overline{A+B} = \bar{A}\bar{B}$
S = $\overline{AB+\bar{A}\bar{B}}$ = A ⊕ B
A
B
C = AB
≡
A
B
S = A ⊕ B
A
B
C = AB

Drawback -

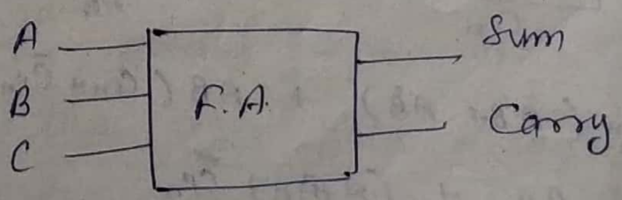
$A_1 \quad A_0 \longrightarrow$ No. A

$+ \; B_1 \quad B_0 \longrightarrow$ No. B

$+ \; C_0 \longrightarrow$ Carry generated from $(A_0 + B_0)$

→ A H.A. can add A_0 & B_0 to produce S_0 and C_0. However the addition of next bits requires the addition of A_1, B_1 & C_0. The addition of 3 Bits is not possible to perform by using an H.A. Therefore practically we don't use a H.A. ckt.

Full Adder :-



A	B	C_{in}	Sum	Carry
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

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

$= \Sigma m(1,2,4,7) = A \oplus B \oplus C_{in}$

A \ BC_{in}	00	01	11	10
0		1		1
1	1		1	

Carry = $\bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$ → write 3times

AB

A Cin

BCin

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

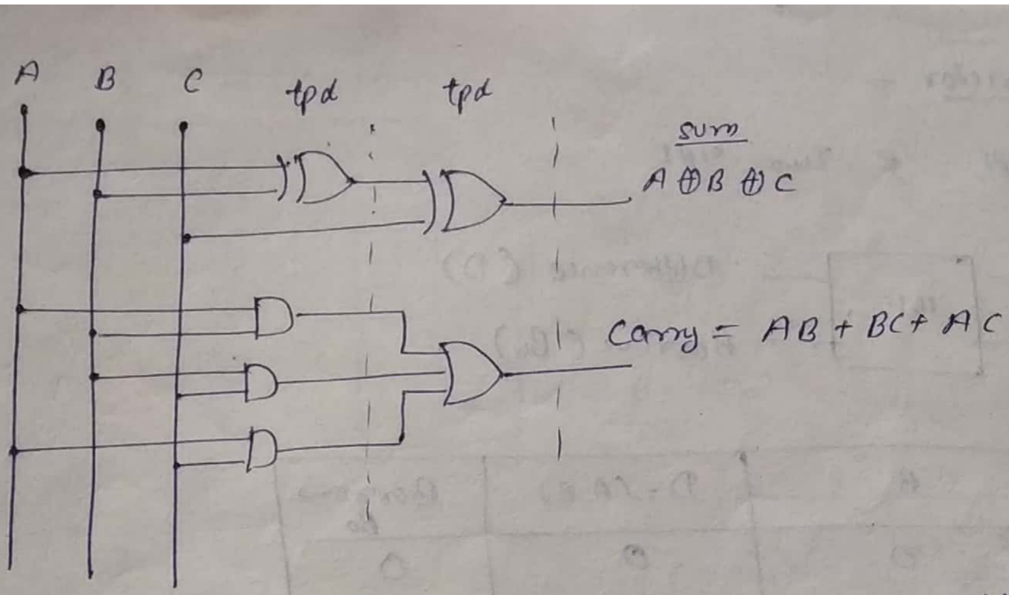
$= AB + BC_{in} + AC_{in}$

alternative for Carry -

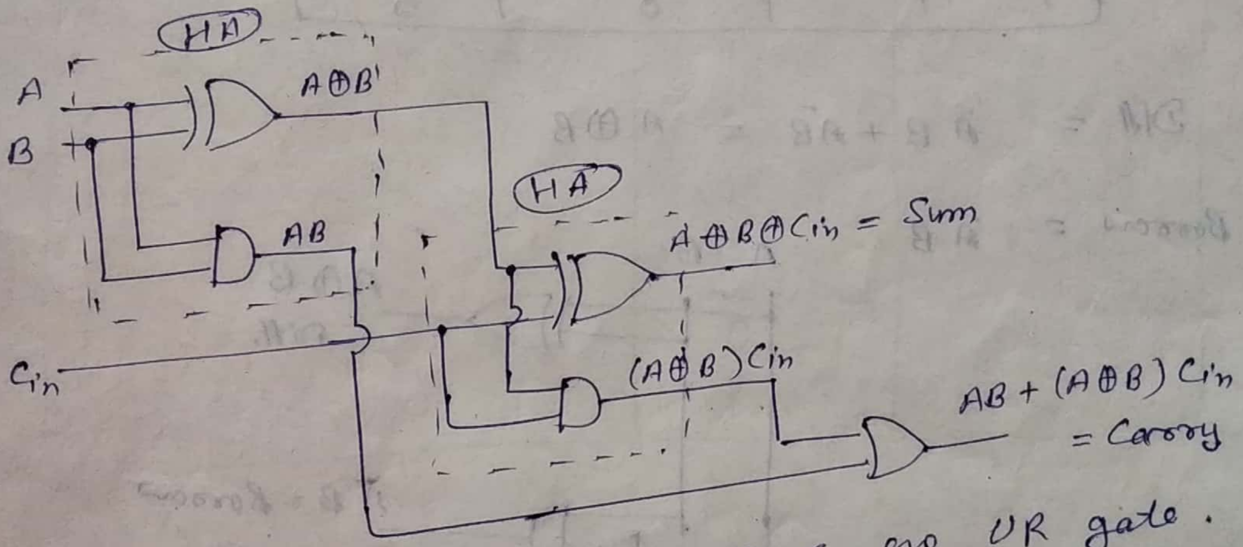
$= \underbrace{\bar{A}BC_{in} + A\bar{B}C_{in}}_{C_{in}} + \underbrace{AB\bar{C}_{in} + ABC_{in}}_{AB}$

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

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



Construction of full adder using H.A.



⇒ Full adder requires two H A & one OR gate.

Application –

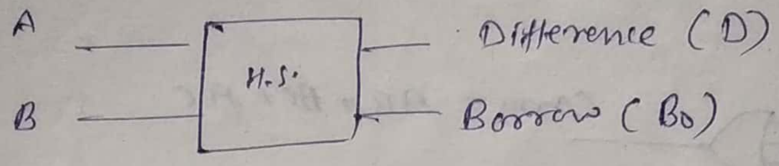
– A full adder acts as a basic building blocks

In FA if each logic gate have propagation delay of tpd. then to provide sum or carry O/P it requires min. Two logic gate delays.

delay for sum = 2 tpd
delay for carry = 2 tpd

Half Subtractor :-

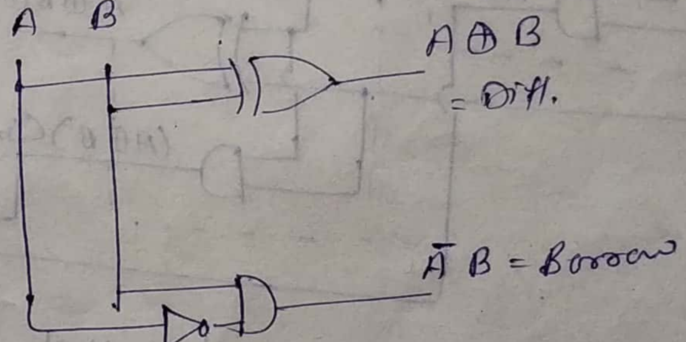
→ Two i/ps & Two o/ps



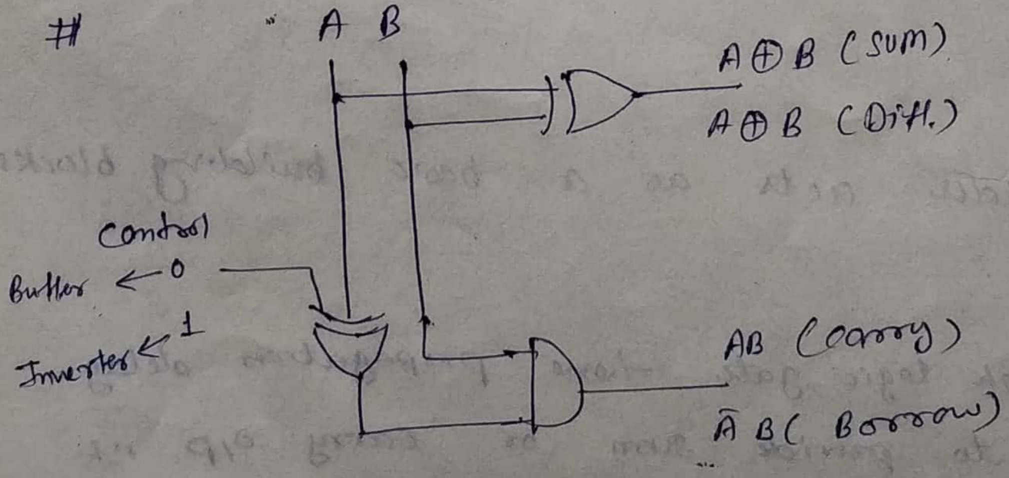
A	B	D = (A-B)	Borrow Bo
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

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

Borrow = $\bar{A}B$



#

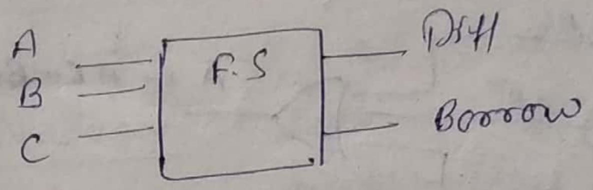


If Control is 0 → HA

if " " 1 → H.S.

full Subtractor :-

→ Three i/ps & Two o/ps.

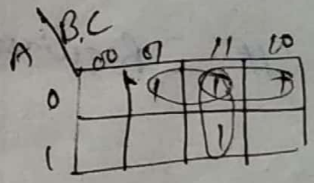


A	B	C	Diff (A-B-C)	Borrow
0	0	0	0	0
0	0	1	1.	1
0	1	0	1.	1
0	1	1	0	1
1	0	0	1.	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

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

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

$= A \oplus B \oplus C$



A \ B.C	00	01	11	10
0		1	1	1
1			1	

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

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

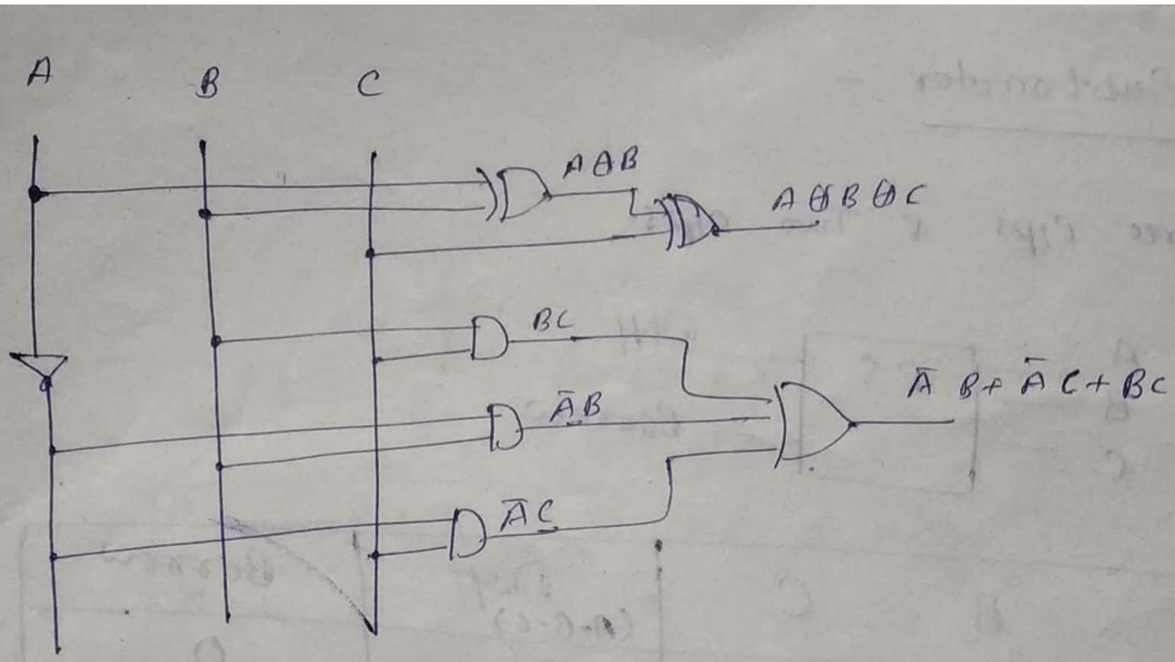
$= \bar{A}C + \bar{A}B + BC$ /

alternative

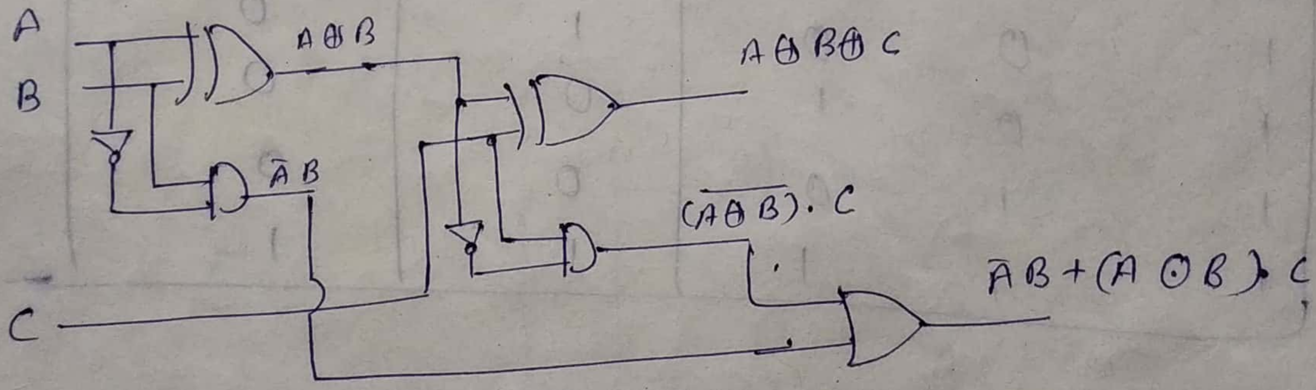
$\bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + ABC$

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

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



Full Subtractor using H.S. :-


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

Requires 2 H.S. & 1-OR gate.