Digital logic circuits

logic circuit of digital system are mainly categorised into two broad categories

- → combinational logic circuits
  eg: Adder, Subtractor, Multiplier, magnitude comparator, code converter, encoder, decoder etc.
- → Sequential logic circuits.
  eg: flip-flops (FFs), Latch, Registers, counter etc.

Combinational logic circuits :-

↳ it consists of logic gates whose output (O/P) at any time are determined from only the present combinational inputs (I/Ps)

↳ the combinational circuits performs an operation that can be specified logically by set of Boolean functions

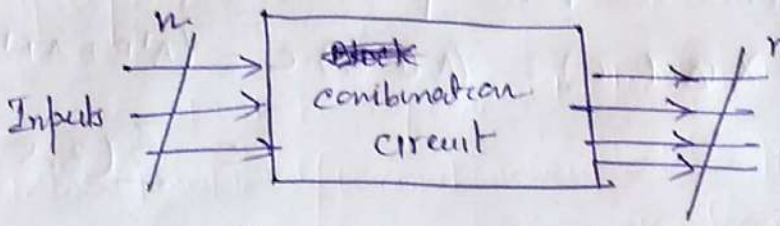
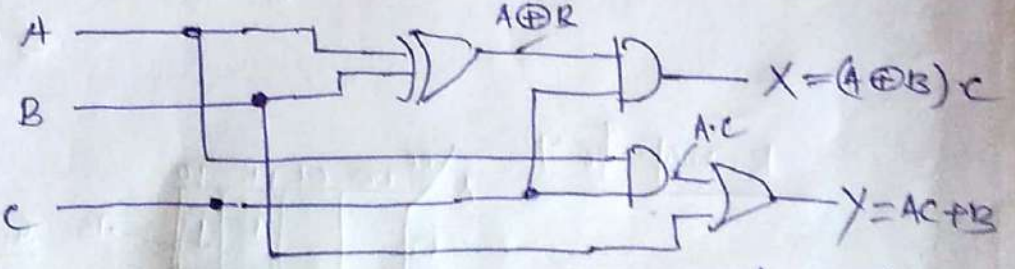


Fig: Block Diagram of combinational logic circuits.

↳ the O/Ps of combinational circuit is a function of present I/Ps only i.e. $O_{Ps} = f(I/Ps)$.



this is an example of combinational circuit whose I/Ps are A, B, C and O/Ps are X and Y. and the value of O/Ps X, Y are determined from Inputs A, B, C only.

P.no: (2)

the o/ps X and Y are a combination of Present I/p only.

$$X(A,B,C) = (A \oplus B)\cdot C = \sum(3,5) = \prod(0,1,2,4,6,7)$$

$$Y(A,B,C) = A\cdot C + B = \sum(2,3,5,6,7) = \prod(0,1,4)$$

Truth tables

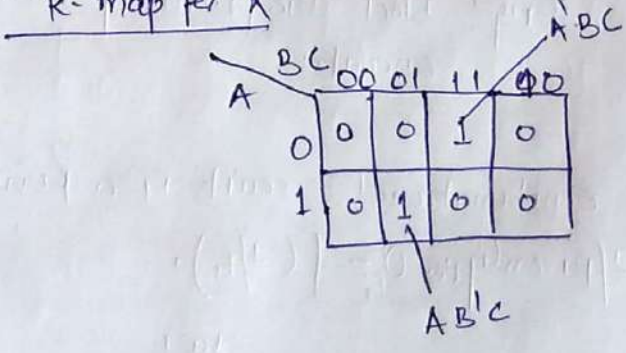
I/p's					O/Ps	
A	B	C	$A \oplus B$	$A\cdot C$	$X = (A \oplus B)\cdot C$	$Y = AC + B$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	1	0	0	1
0	1	1	1	0	1	1
1	0	0	1	0	0	0
1	0	1	1	1	1	1
1	1	0	0	0	0	1
1	1	1	0	1	0	1

the minimized form of above outputs are.

$$X(A,B,C) = (A \oplus B)\cdot C = (A'B + AB')\cdot C = A'BC + AB'C$$

$$= \sum(3,5)$$

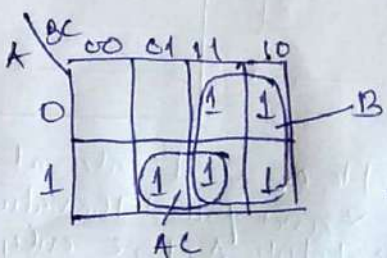
K-map for X



Since the squares whose minterm values are 1 are not adjacent to each other. Hence we not make a group of more than one squares.

⇒ the above output Remains X is already in simplified form.

K-map for Y



Simplified form of Y

$$Y(A,B,C) = B + AC$$

For Sequential circuit :-

↳ the output of a sequential circuit depend not only on Present values of I/ps, but also on past inputs.

↳ the O/Ps of sequential circuit are a function of I/Ps and the state of storage Element.

$$O = f(I/Ps, \text{Storage Elements})$$

→ past I/Ps.

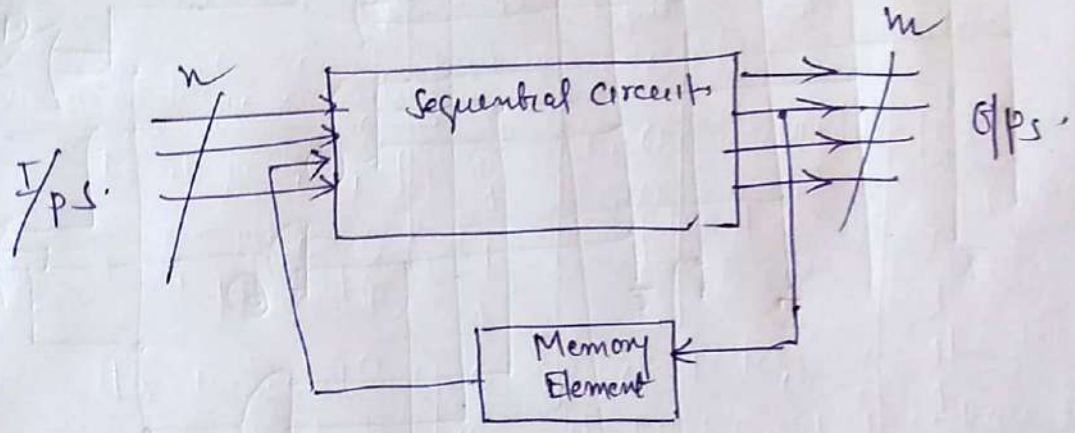
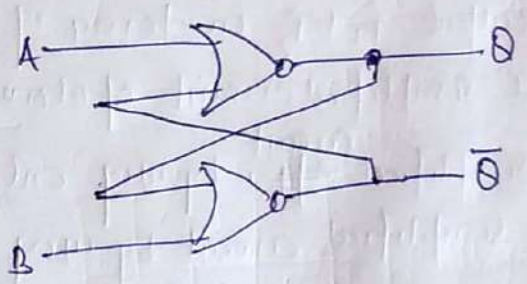


Fig: Block Diagram of sequential circuit

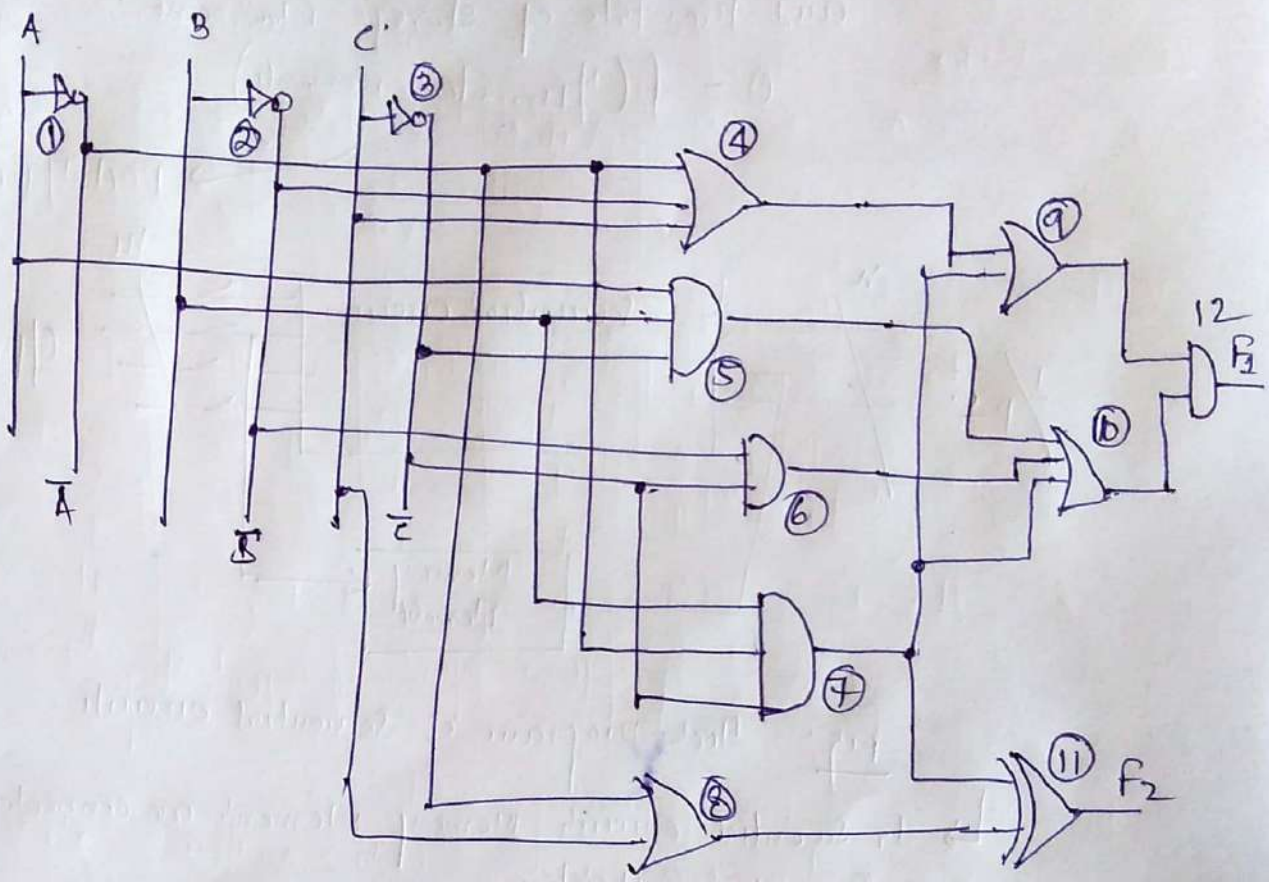
↳ In sequential circuit Memory Element are connected in the form of feedback.



In this circuit outputs Q and $\overline{Q}$ are not determined only from Present values of I/ps A & B but it also Depends on the value of past I/Ps (Stored value of Q and Q'. Hence it is an Example of sequential circuits

Analysis of combinational circuits :-

→ we determine the o/ps of circuits in terms of Boolean function that the circuit implements



determine the output F_1 & F_2 in terms of present I/ps A, B, C and also find simplified circuit of above circuit (if possible)

→ Implement simplified ~~o/p~~ circuit by using only NAND gates.

→ Implement simplified circuit by using only NOR gates.

→ Implement simplified circuit by using BASIC logic gates.

First determine the ~~output~~ Boolean function of o/ps F_1 & F_2 i.e. o/ps of gate 12 & 11. In this logic diagram total 12 logic gates are used and that are labled from 1 to 12.

Here the I/ps of circuits are A, B, C hence we determine the o/ps of all labled logic gates in terms of A, B, C.

The o/p of logic gate labled (1) is $\overline{A}$

The o/p ——— (2) is $\overline{B}$

The o/p ——— (3) is $\overline{C}$

The o/p logic gate labled as (4) is $A+\overline{B}+C$

The o/p ——— (5) is $A\,B\,\overline{C}$

The o/p ——— (6) is $\overline{B}\,\overline{C}$

The o/p ——— (7) is $A\,B\,\overline{C}$

The o/p ——— (8) is $A+C$

The o/p ——— (9) is the o/p of logic gate labelled as (4) + the o/p of logic gate labled as (7)

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

$= A(1+B\overline{C}) + \overline{B} + \overline{C} = A+\overline{B}+C$ (1)

The o/p ——— (10) = the o/p of logic gate labelled as (5) + the o/p of logic gate labelled as (6) + the o/p of logic gate labelled as (7)

$= AB\overline{C} + \overline{B}\,\overline{C} + AB\overline{C}$

$= AB\overline{C} + \overline{B}\,\overline{C} = \overline{C}(AB+\overline{B})$

$= \overline{C}[(A+\overline{B})(B+\overline{B})] = A\overline{C} + \overline{B}\,\overline{C}$ (1)

The o/p of logic gate labelled as (11) = the o/p of logic gate labelled as (8) ⊕ the o/p of logic gate labelled as (10)

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

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

$= AB\overline{C}\cdot\overline{A}\cdot\overline{C} + (\overline{A}+\overline{B}+C)\cdot(A+C)$ ($AB\overline{C}\cdot\overline{A}\cdot\overline{C} = 0$)

$= A\overline{A} + A\overline{B} + AC + \overline{A}C + \overline{B}C + C\cdot C$ ($A\overline{A} = 0$)

$= A\overline{B} + C(A+\overline{A}) + \overline{B}C + C$ ($A+\overline{A} = 1$)

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

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

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

$= A\bar{B} + C$

teo o/p of logic gate labelled as (12) = teo o/p of logic gate labelled (9) ◆ teo o/p of logic gate labelled as (10)

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

$= A \cdot A\bar{C} + \bar{B}A\bar{C} + C \cdot A\bar{C} + AB\bar{C} + \bar{B}B\bar{C} + C\bar{C}\bar{C}$ (C·AC̄ = 0, B·B̄C̄ = 0, C·C̄C̄ = 0)

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

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

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

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

$= A\bar{C}$

Sence in the given combinational circuit there are three I/ps named as A, B, C and teo o/ps named as F_1 & F_2 and teo o/ps values are determined in terms of present I/p. A, B, C are

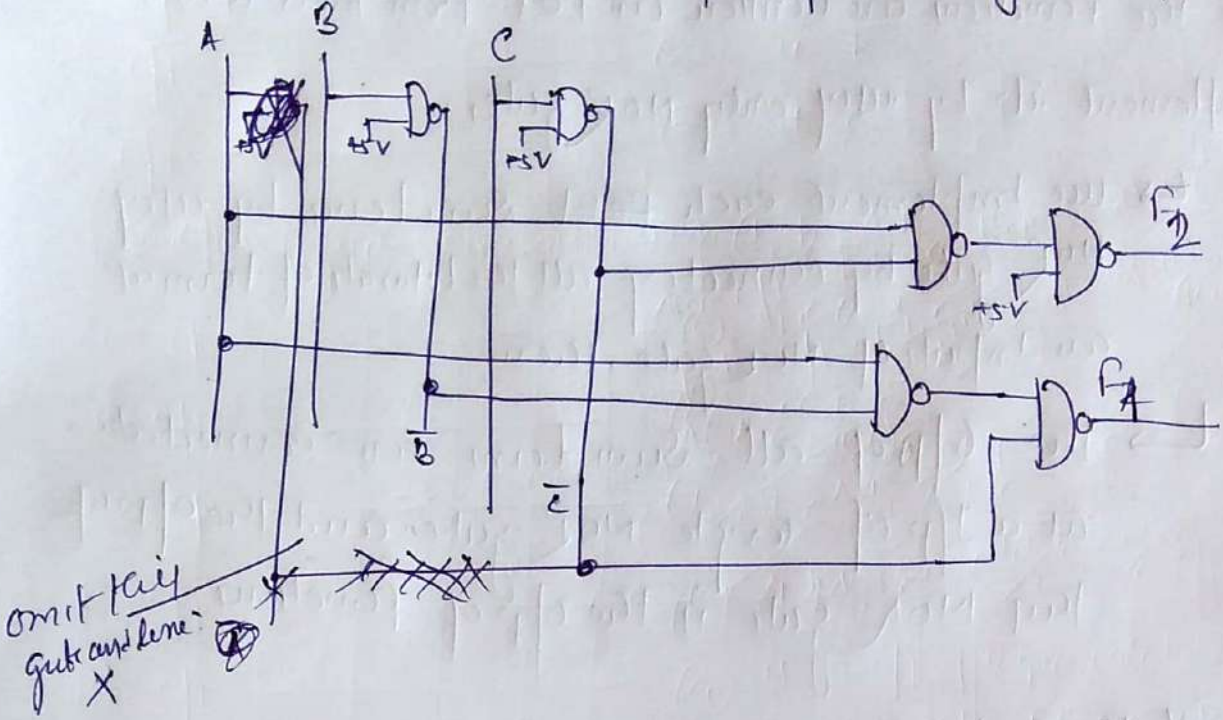
$F_2(A, B, C) = A\bar{C} = \sum(4, 6)$

$F_1(A, B, C) = A\bar{B} + C = \sum(1, 3, 4, 5, 7)$

the Truth table of F_1 & F_2 are as follows:

A	B	C	B̄	C̄	AB̄	$F_1 = A\bar{C}$	$F_2 = A\bar{B} + C$
0	0	0	1	1	0	0	0
0	0	1	1	0	0	0	1
0	1	0	0	1	0	0	0
0	1	1	0	0	0	0	1
1	0	0	1	1	1	1	1
1	0	1	1	0	1	0	1
1	1	0	0	1	0	1	0
1	1	1	0	0	0	0	1

Implementation of f_1 & f_2 by using NAND gate only



$$f_2 = \overline{\overline{A\bar{C}}} = A\bar{C}$$

$$f_1 = \overline{\overline{A\bar{B}} \cdot \overline{A}} = \overline{\overline{A\bar{B}} + \bar{\bar{C}}} = A\bar{B} + C$$

Implementation of circuits by using NOR gate only.

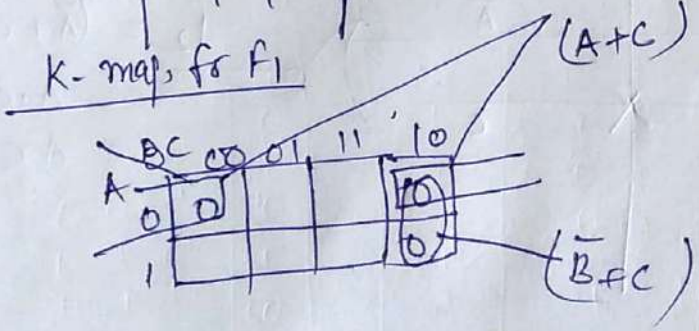
First form the o/p, function f_1 & f_2 in POS form and simplify it.

$$f_1 = A\bar{B} + C = \Sigma(1,3,4,5,7) = \Pi(0,2,6) = (A+C)(\bar{B}+C)$$

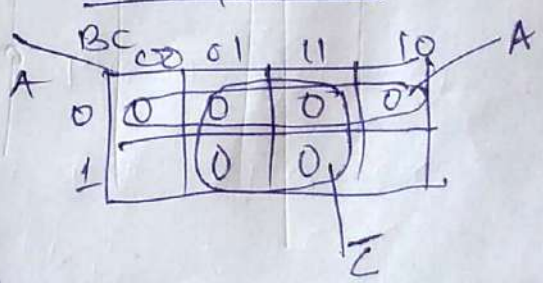
$$f_2 = A\bar{C} = \Sigma(4,6) = \Pi(0,1,2,3,5,7) = (A)(\bar{C})$$

Simplify by using K-map in POS form [i.e we form the group of square which value are 0]

K-map for f_1



K-map for f_2



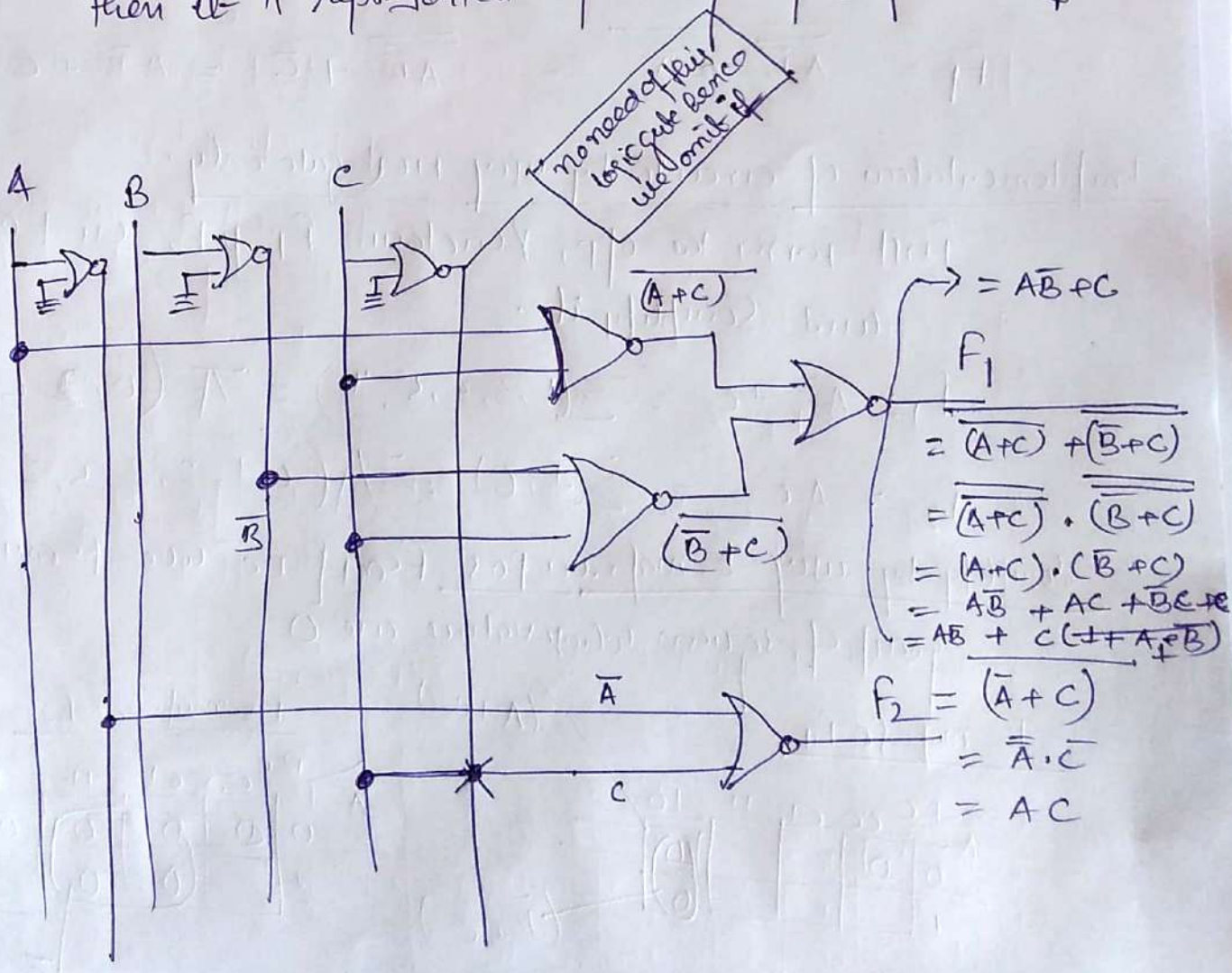
$f_1 = (A+C).(\bar{B}+C)$ [in POS form] $f_2 = (A).(\bar{C})$ in POS form

if the Function are written in POS form then we can implement it by using only NOR gate.

↳ we implement each ~~POS~~ Sum term by using NOR gate by connecting all the literals of term as an Input of NOR gate, then

↳ the O/P of all Sum term are connected as a I/P of single NOR gate. and the O/P of this NOR gate is the O/P of function.

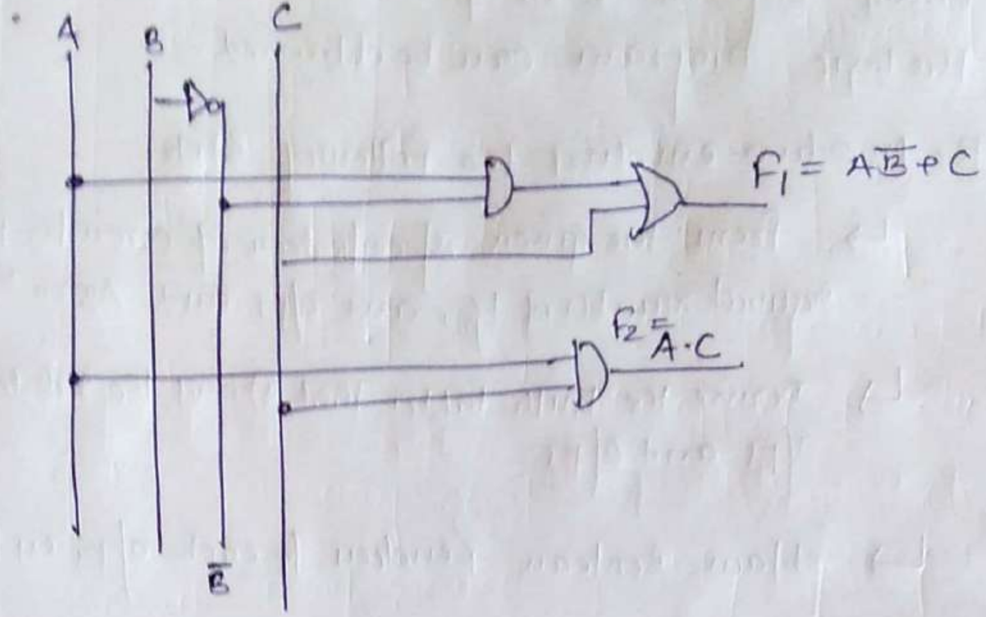
Note: if there is any complemented variable in Sum term then it is represented separately by using NOR gate.



Implementation of simplified output by using Basic logic gates.

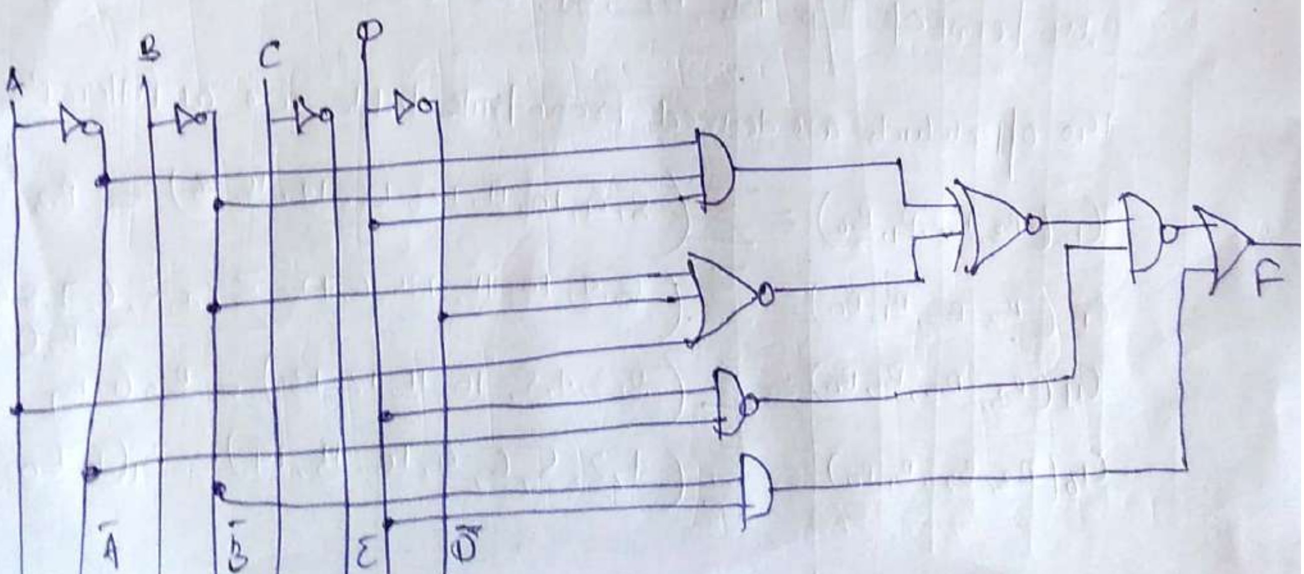
$f_1 = A\overline{B} + C$

$f_2 = A \cdot C$



Q: Analyze the given combinational circuit and find the Boolean function for their outputs and also simplify it and implement it by using
i) only NAND gates.
ii) only NOR gates.
iii) Basic logic gates.

and also find the O/p of circuits in the following input combinations and send the Answer through WhatsApp (in the form of 0 & 1)
ⓐ 0000 ⓑ 1010 ⓒ 1111 ⓓ 0111 ⓔ 1001



Design procedure of combinational logic circuits:

The Design of combinational circuits starts from the specification of Design objective and culminates with logic circuit Diagrams of set of Boolean functions from which the logic Diagram can be obtained.

→ The procedure involves the following steps:

↳ From the given specification of circuit, find the required number of I/p, and o/ps and Assign symbol to each.

↳ Derive the Truth table that shows the relation ship between I/ps and o/ps.

↳ obtain Boolean function for each o/p in terms of I/ps

↳ Draw the logic Diagram and verify its correctness (manually)

Eg: Design a logic circuit which converts a 4bit Binary Number into ~~BCD~~ 4 bit gray code.

Sol^n: Here I/ps are 4 bit binary code in the form $B_3 B_2 B_1 B_0$

$\Rightarrow$ total possible I/p combinations $= 2^4 = 16$

o/ps are 4 bit ~~binary~~ gray code in the form $G_3 G_2 G_1 G_0$

Then we determine the Boolean function for each o/p $G_3, G_2, G_1 \& G_0$ in term of I/p B_3, B_2, B_1, B_0 from the Truth table.

the o/ps which are derived from truth table are as follows:

$G_3(B_3, B_2, B_1, B_0) = \sum(8, 9, 10, 11, 12, 13, 14, 15) = B_3$

$G_2(B_3, B_2, B_1, B_0) = \sum(8, 9, 10, 11, 4, 5, 6, 7) = \sum(4, 5, 6, 7, 8, 9, 10, 11) = B_3 \oplus B_2$

$G_1(B_3, B_2, B_1, B_0) = \sum(2, 3, 4, 5, 10, 11, 12, 13) = B_2 \oplus B_1$

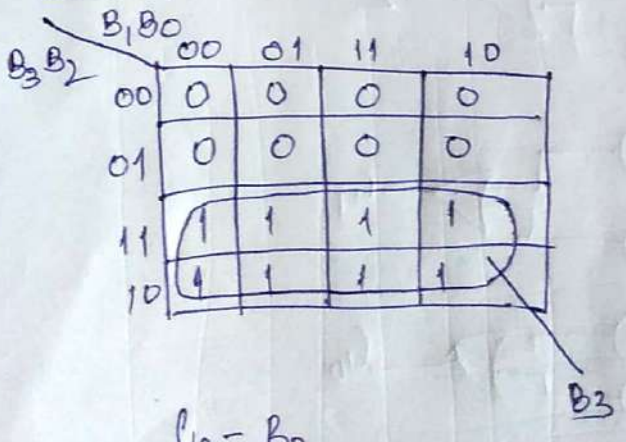
$G_0(B_3, B_2, B_1, B_0) = \sum(1, 2, 5, 6, 9, 10, 13, 14) = B_1 \oplus B_0$

Truth table for 4 bit binary to gray code:

i/ps				o/ps			
B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

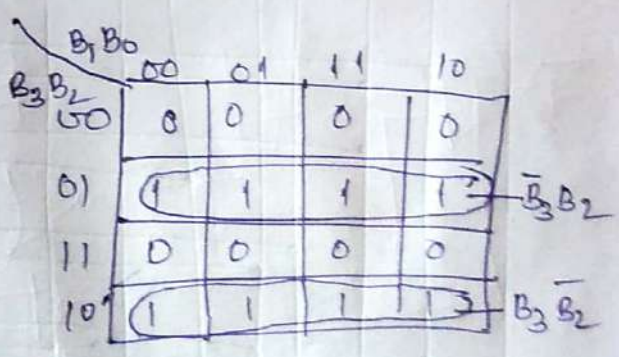
then draw two K-maps for each o/p G_3, G_2, G_1 & G_0 and find Boolean function corresponding to each o/p in terms of I/ps B_3, B_2, B_1 & B_0

K-map for G_3



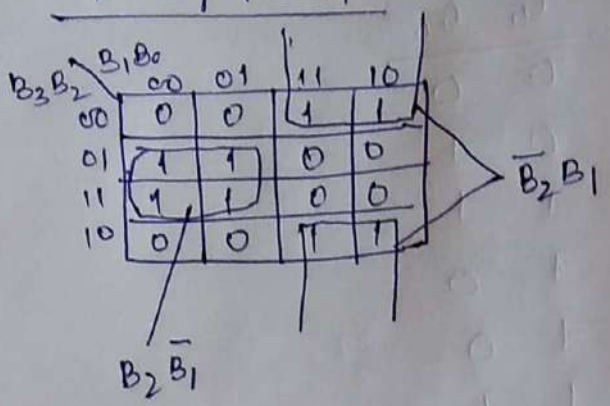
$G_3 = B_3$

K-map for G_2



$G_2 = \bar{B}_3 B_2 + B_3 \bar{B}_2$

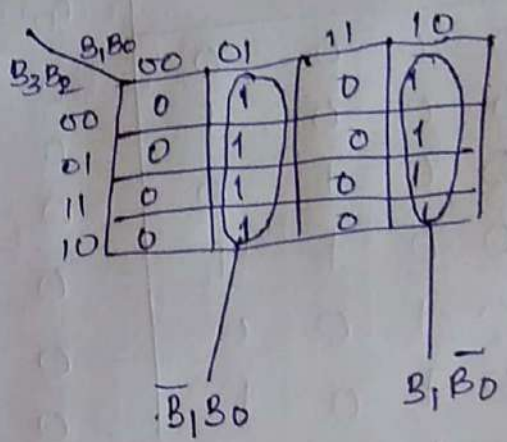
K-map for G_1



$G_1 = B_2\bar{B}_1 + \bar{B}_2 B_1$

$= B_2 \oplus B_1$

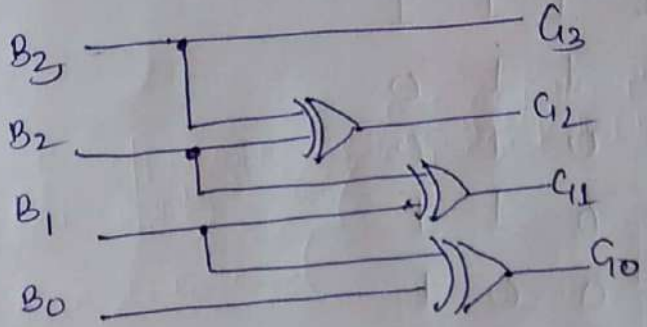
K-map for G_0



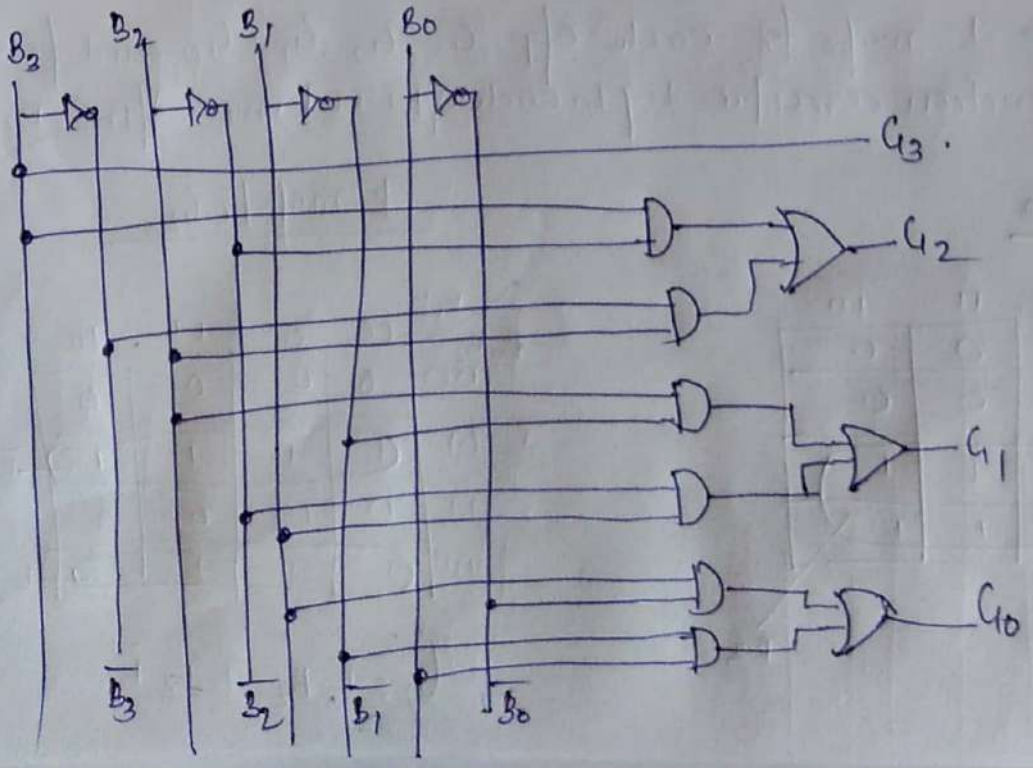
$G_0 = \bar{B}_1 B_0 + B_1 \bar{B}_0$

$= B_1 \oplus B_0$

i) Design of 4 bit Binary to Gray code converter circuit by using logic gates.



ii) Design of 4 bit Binary to Gray code converter by using Basic logic gate



Design of 4 bit Binary to Gray code converter by using NOR gate only. Same (13)

→ First we write the o/p function in SOP form for easy design of circuit through NOR gate.

$$G_3 = B_3$$

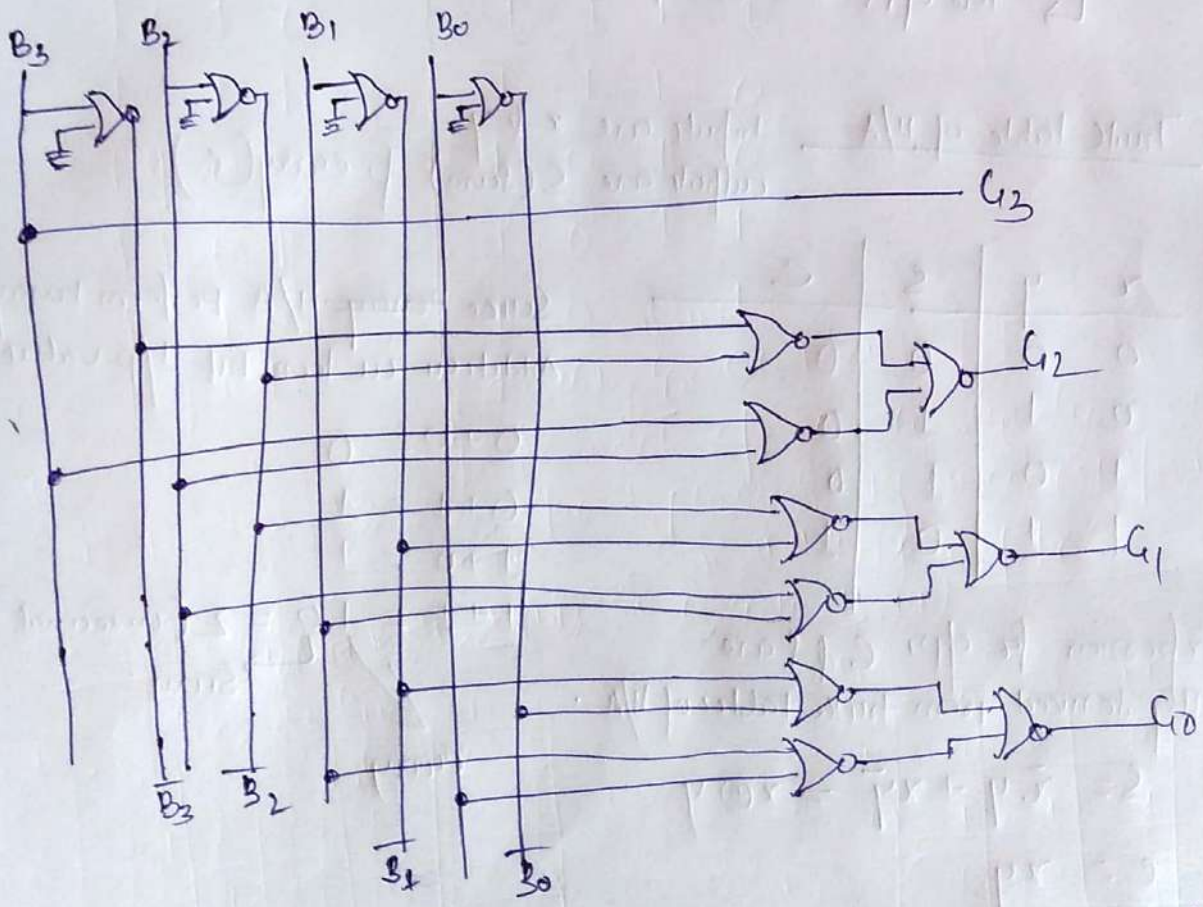
$$G_2 = (B_3\overline{B_2}) + (\overline{B_3}B_2) = (B_3\overline{B_2} + \overline{B_3}).(B_3\overline{B_2} + B_2)$$

$$= (\underbrace{B_3 + \overline{B_3}}_{1}).(\overline{B_2} + \overline{B_3}).(B_3 + B_2)(\overline{B_2} + \underbrace{B_2}_{1})$$

$$= (\overline{B_2} + \overline{B_3})(B_3 + B_2)$$

Similarly $G_1 = (\overline{B_2} + \overline{B_1})(B_2 + B_1)$

$G_0 = (\overline{B_1} + \overline{B_0})(B_1 + B_0)$



Q. Design of 4 bit Binary to gray code converter by using NAND gate only (Exercise)

Q. Design a circuit of BCD code to Excess-3 code converter (Exercise)

Binary Adder :-

↳ it is a combinational circuits that perform the arithmetic operation Addition with Binary Numbers.

- → Half Adder (H/A)
- → Full Adder (F/A)
- → connecting n-Full Adders in cascade produces a binary Adder for two - n-bit binary Numbers.

Half Adder:

↳ it is a combinational circuit.

↳ H/A needs two Binary inputs and two Binary o/ps.

↳ two Inputs are known as Augend and Addend bits;

↳ two o/Ps Variable produce their Sum & carry.

Truth table of H/A — Inputs are x & y.
outputs are S (Sum) & carry (C).

x	y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Since ~~Binary~~ H/A Perform binary Addition in two bit I/p value

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 10 = 2$ (in decimal)

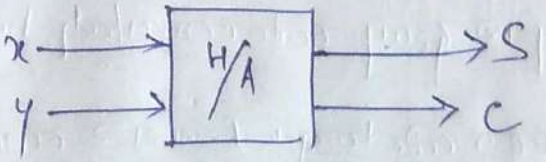
↳ Sum.

Carry

the expression for o/ps C & S are.
directly derived from truth tables of H/A.

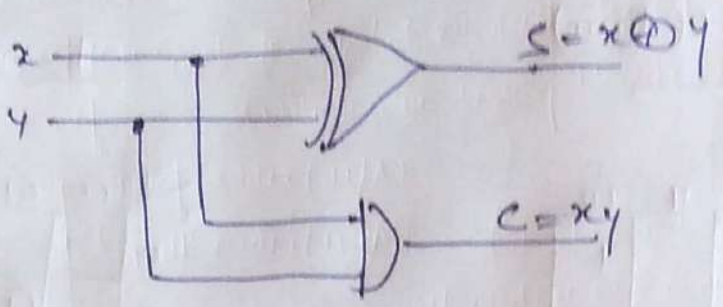
$$S = \bar{x}y + x\bar{y} = x \oplus y$$

$$C = xy$$



Block Diagram of H/A.

Design of H/A by using logic gates:



Full Adder (F/A)

↳ it is a combinational circuit
↳ it forms the arithmetic sum of three binary bits
↳ it consist of three I/p and two O/p's
↳ the two I/p variables denoted by x & y represents the two significant bits to be added and the third O/p z represents the carry from the previous lower significant position. From three I/ps variable x, y, & z, there are 8 possible mentions.
↳ two O/ps are represented by sum and carry.

Truth table of F/A

I/Ps			O/Ps	
x	y	z	S	C
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

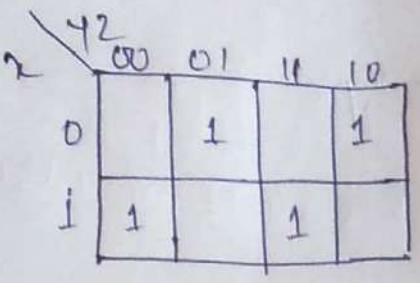
the expression from output S & C are directly derived from Truth table.

$S(x,y,z) = \sum(1,2,4,7)$
$= \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$

$C(x,y,z) = \sum(3,5,6,7)$
$= \bar{x}yz + x\bar{y}z + xy\bar{z} + xyz$

Simplify the o/ps of F/A by using k-map

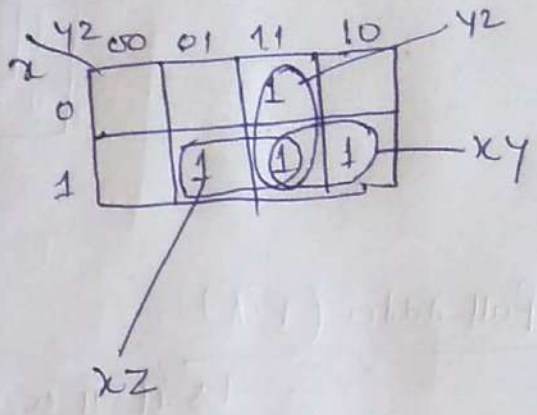
k-map for S (sum)



Since all the diagonal elements are 1 and diagonal pair form. R.H.S term o/p function is written as

$S = x \oplus y \oplus z$

k-map for c (carry)



$C(x,y,z) = xy + yz + xz$

For minimizing the number of logic gates required to implement the F/A, the o/p are represented as follows:

$S(x,y,z) = x \oplus y \oplus z$

$C(x,y,z) = \bar{x}yz + x\bar{y}z + xy\bar{z} + xyz$

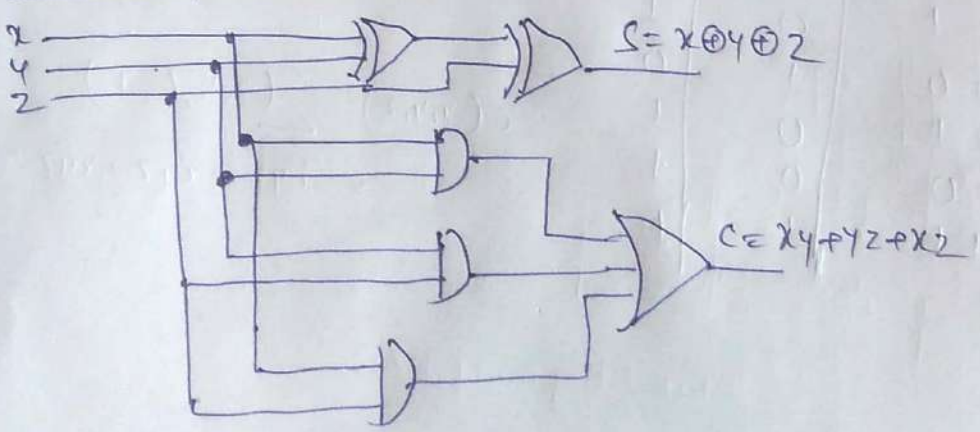
$= \bar{x}yz + x\bar{y}z + xy(\bar{z} + z)$

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

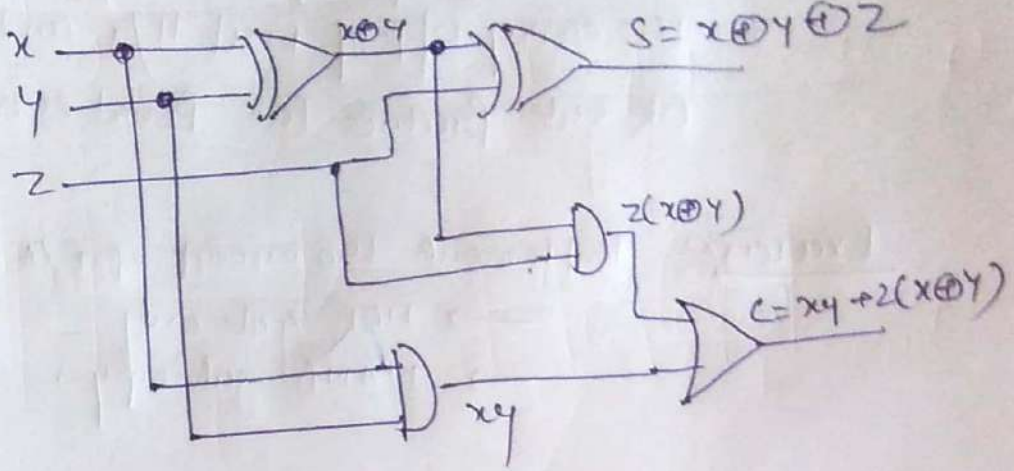
$= xy + z(\bar{x}y + x\bar{y})$

$= xy + z(x \oplus y)$

Design of Full Adder:



The above circuit is also represented as.



Implementation of F/A by using H/As

→ We know the Block Diagram of H/A.

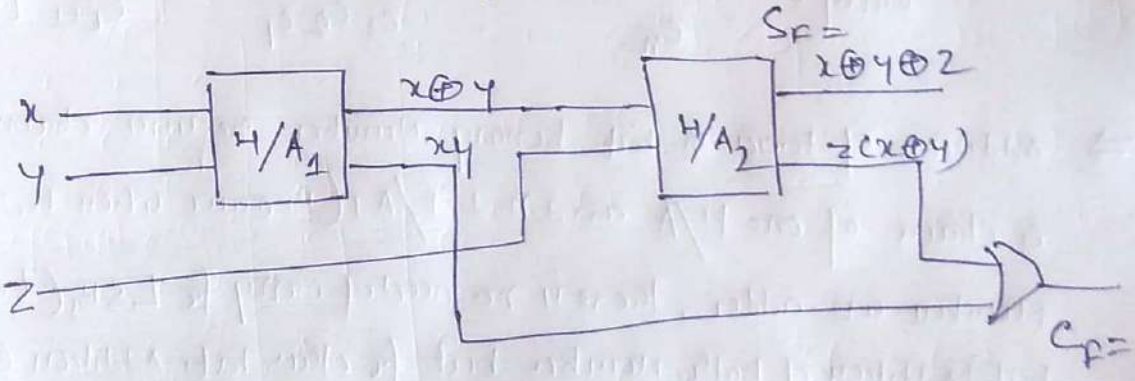
→ In a H/A

$S_H = x \oplus y$

$C_H = xy$

In a Full Adder $S_F = x \oplus y \oplus z$

$C_F = xy + z(x \oplus y)$



Explanation: For representing F/A, two H/A and single OR gate are required.

First H/A, there are two I/ps x, y and two O/ps $S_H = x \oplus y$, $C_H = xy$.

In second H/A, the first I/p is the Sum (S_H) O/p of first H/A and second I/p is third I/p variable z and the Sum O/p produce Sum (S_F) O/p of Full Adder

and carry o/p of second H/A is ORed with the carry o/p of first H/A and the o/p of full adder OR gate produce the final carry of full Adder.

Excercise: Implement the circuit of F/A & H/A by using
→ NOR gate only
→ NAND gate only.

Binary Adder: ↳ it is a combinational circuit which produce a arithmetic sum of two N-bit binary Number.
↳ it can be constructed by connecting n-Full Adder in cascade with the o/p carry of each Full Adder is connected to the I/p carry of Next Full Adder in chain

I/p carry	C_{n-1}	$C_2\ C_1\ C_0=0$	C_i	$i \leftarrow 0 \cdots n-1$
Augend (n bit)	A_{n-1}	$A_2\ A_1\ A_0$	A_i	$i \leftarrow 0 \cdots n-1$
Addend (n bit)	B_{n-1}	$B_2\ B_1\ B_0$	B_i	$i \leftarrow 0 \cdots n-1$
Sum	S_{n-1}	$S_2\ S_1\ S_0$	S_i	$i \leftarrow 0 \cdots n-1$
carry	C_n	$C_3\ C_2\ C_1$	C_{i+1}	$i \leftarrow 0 \cdots n-1$

⇒ Addition of two n-bit binary Number required chain of n F/A or a chain of one H/A and n-1 F/A s (Because when two n bit binary Number are adder, there is no need of carry for LSB (Least Significant bit) Addition of both Number but for other bits Addition carry required

eg: 1000 ← $C_0=0$ (Initial carry)

A = 1001

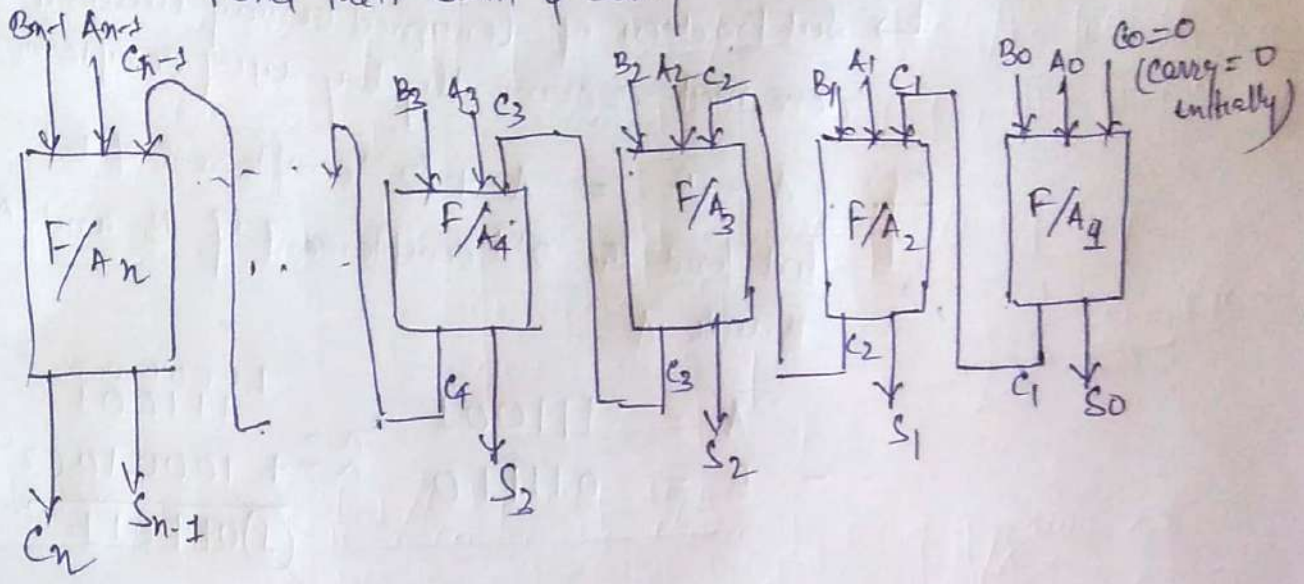
\+ B = 1110

0111 Sum ⇒ 10111

1000 carry

final carry

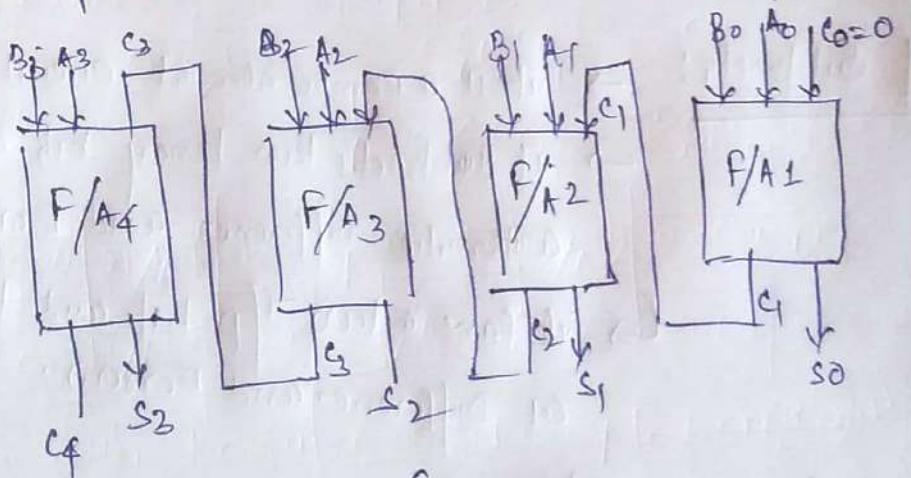
n- bit Binary Adder → it Adds two n bit binary number and Send their Sum & carry.



the arithmetic Sum of two n bits binary number is obtained in the form

$$C_n S_{n-1} \cdots S_1 S_0$$

4-Bit Binary Adder → it Adds two four Bit binary Number and Produce their arithmetic Sum & carry.



$$A = \begin{array}{r} 111 \\ 1101 \end{array} \quad \text{if } C_0 = 0$$

$$B = 1111$$

$$\overline{11100} \quad \text{Sum}$$

Final carry.

Binary Subtractor:

↳ Subtraction of unsigned binary numbers can be done most conviniently by using complements.

eg: $A - B = A + (2's \text{ complement of } B)$

first find the 2's complement of B and Add it with A

$A = 111001$

$-B = 011110$

1000000
111001
$+ 100010$ (2^{s} comp of B)
$\overline{(1)011011}$

↳ there is a carry bit
⇒ result is positive
⇒ Ignore carry bit and remaining bits represent resultant.

Half Subtractor:

→ it is a combinational circuit.
→ it Subtracts two binary bits and find their arithmetic Difference and Borrow.
↳ it have a two I/p bits and two o/ps named as Difference and Borrow.

Truth table of H/S

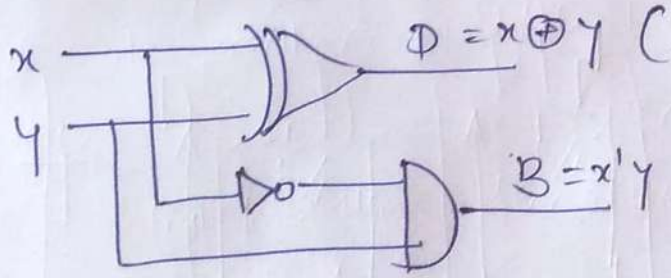
x	y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

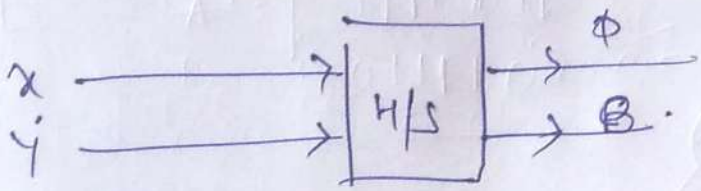
two o/ps expression for Difference(D) and Borrow(B) are directly derived from truth table

$D = x\bar{y} + \bar{x}y$

$B = \bar{x}y$

Logic Diagram of H/S.





Block Diagram of H/S.