

As we have learnt the boolean algebra, about the boolean functions, the boolean expressions as well as the different basic gates with the universal gates. These gates are used for the manufacturing of the different electronic circuits by the help of the different simplified boolean logical expressions. The logical circuits may consists of number of gates along with the binary variables as input and the outputs. The output of the given circuit completely depends upon the boolean expression. The logic circuits are classified as

- \* Combinational circuits.
- \* Sequential circuits.

### Combinational circuits:-

The logic circuits whose output at any instant completely depends on the input present at that instant are called the combinational circuits.

The combinational circuits consists of  $n$  input variables and at that instant it consists of  $m$  output variables. The output of the combinational circuit is related to its input by a combinational function which is independent of the time.

The logic diagram of the combinational circuit is as shown below -

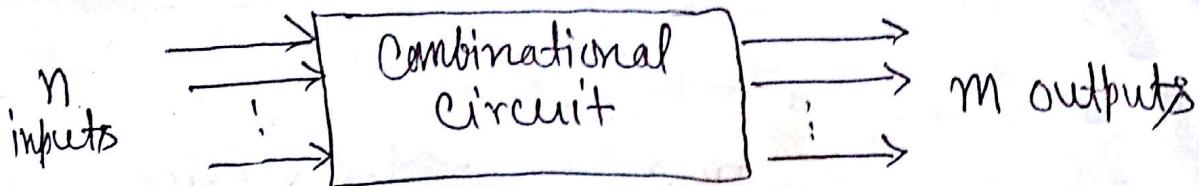


fig:- Combinational circuit

The combinational circuit do not require any memory element. Due to absence of memory this circuit is easy to design. The realization of this circuit requires more hardware component but showing fast speed nature in real practice. Due to the presence of more hardware (gates) this become more expensive in nature.

### Design Procedure of Combinational circuit

The following steps are followed for the design procedure.

- i Problem should be stated
- ii The no. of input as well as the output variables are determined.
- iii Both variables are assigned to letters or symbol
- iv According to the problem statement truth table is generate
- v Simplified boolean expression is obtained (either by using algebra or k-map)
- vi Finally draw the logic circuit diagram according to the simplified function.

## Adders

The combinational circuits which are used for performing the addition tasks are known as the adders. Depending upon the no. of bits to be added the adders are classified into two types.

- ① Half adder.
- ② Full adder.

### Half adder:-

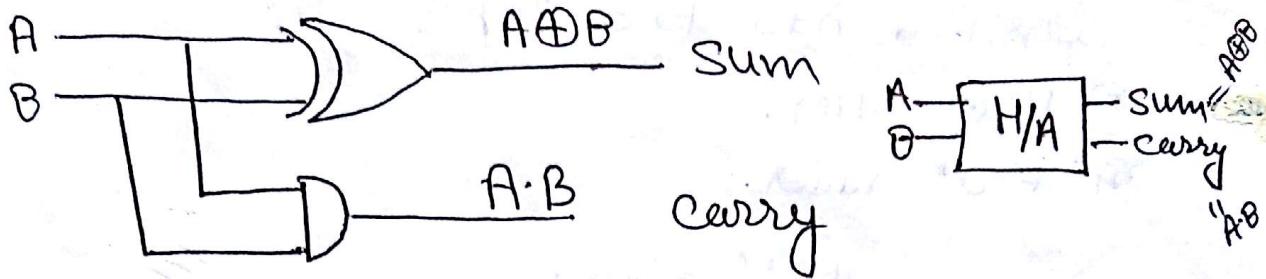
Half adder is the combinational logic circuit which is used for addition of two single bit number.

To perform the addition of two binary number we use a combinational circuit which is known as the half adder. Means that the half adder consists of the two inputs. According to the rule of general binary addition we can derive the truth table for inputs and according to the design procedure we can assign the variables by the binary values and analysing the output we create the basic circuit as follows.

x	y	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- This table shows that the sum is equivalent to XOR gate and carry is equivalent to AND

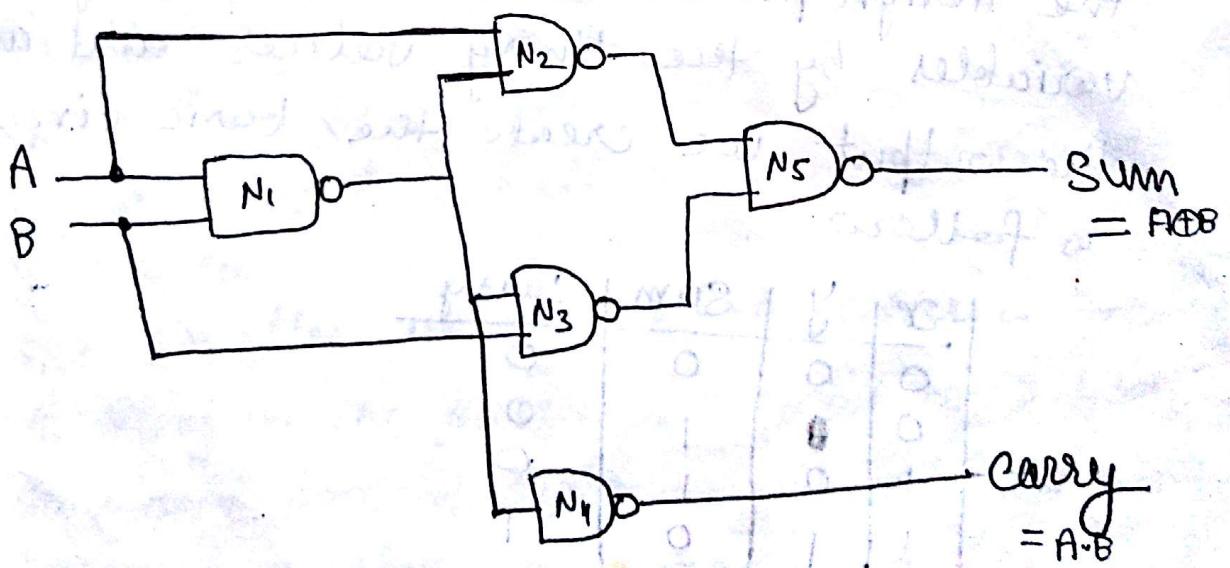
gate. Here is no need for further simplification and hence we can easily draw the logical circuit by using the gates XOR and AND.



Logic circuit for HA

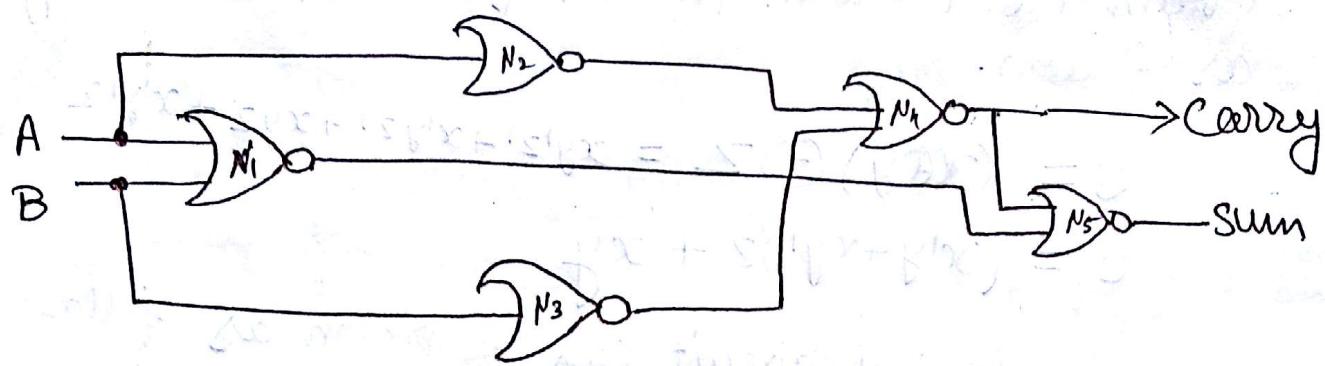
Also we know that the NAND and NOR gates are the universal gates and are considered as the expert sets for developing logic circuits and we can design the halfadder by using both of these gates as follows.

### Half adder using NAND gate



Using 5 NAND gates we can easily design the half adder circuit.

## Half adder using NOR gate



Here we use 5 NOR gates for constructing the half adder.

## Full Adder:-

The combinational circuit that is used for addition of three binary variables. Means if we have to add three binary digits x, y and z first of all two bits are added after that the third variable is added to the previous sum and the process of taking carry is taken into account from the previous case. The truth table for the full adder producing sum and carry is as shown below:

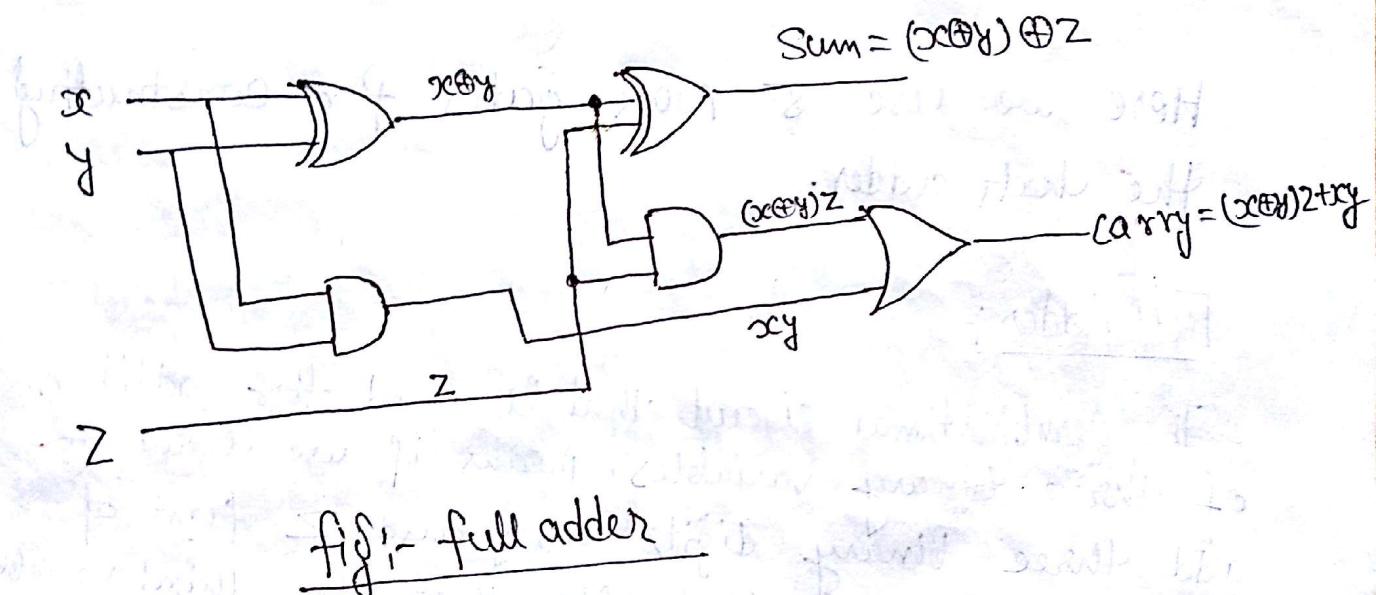
x	y	z	Sum.	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

By analyzing the truth table we can obtain the boolean function for sum and carry as follow.

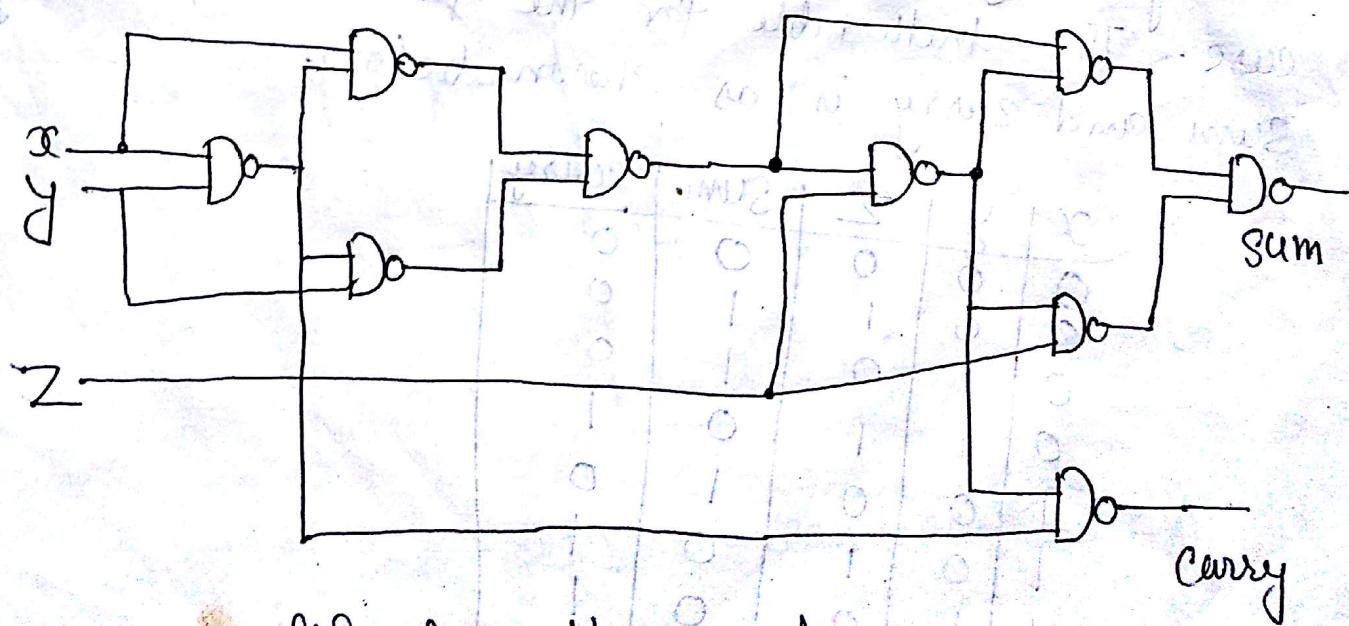
$$S = (x \oplus y) \oplus z = xy'z' + x'y'z + xyz + x'yz'$$

$$C = (x'y + xy')z + xy$$

Now, the logical circuit can be drawn as follow



We can draw the full adder using NAND gates as follow.



## Subtractors:-

The subtraction of two binary digits is done by the half subtractor. Means it is the combinational circuit used to accomplish the binary subtraction. In the binary subtraction the output is the difference. It also has the output borrow if we take it from the left most digit. If we have two inputs  $x$  &  $y$  and have to perform  $x-y$ . Here  $x$  is minuend &  $y$  is subtrahend. Consider the following truth table using the rule of binary subtraction.

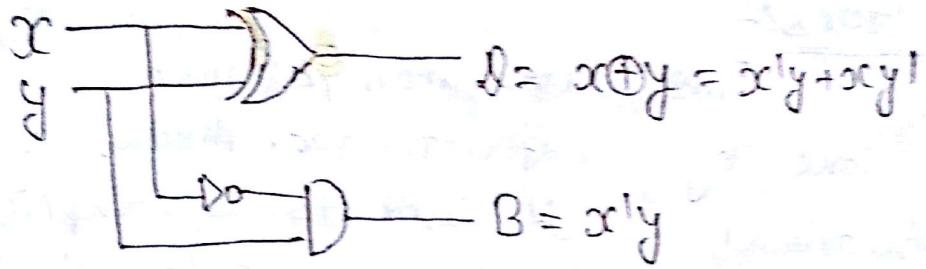
$x$	$y$	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Here the difference value is just similar as the sum value of the half adder. and the borrow value is equal to  $x'y$ . Hence the expression for difference & borrow can be given as

$$\text{Difference} = x \oplus y = xy + x'y$$

$$\text{borrow} = x'y$$

Now the logic diagram can be easily drawn for the half subtractor as follows.



We can also design the half-subtractor by using the NAND gates as follows.

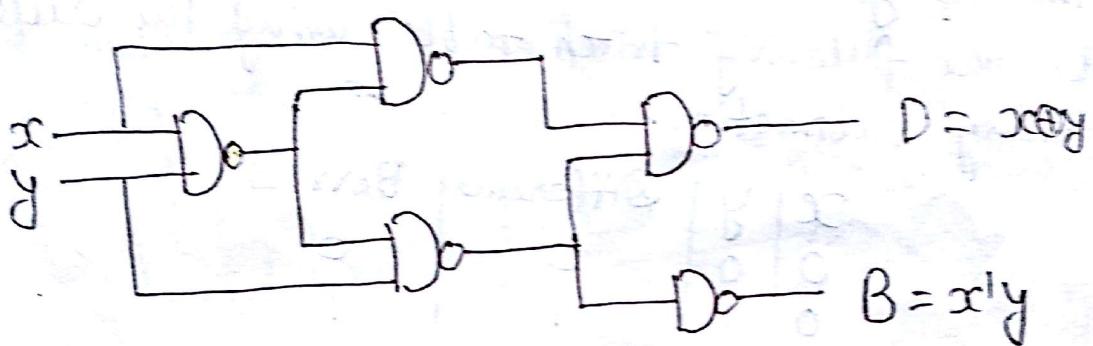


fig:- HIS using NAND gates.

Here we use 5 NAND gates.

### Half subtractor using NOR gates

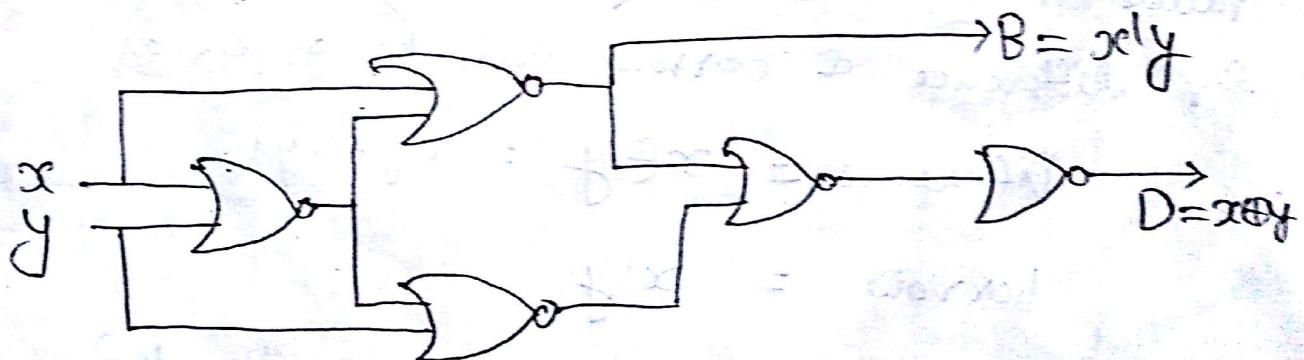


fig:- HIS using NOR gates.

Here, we use 5 NOR gates for HIS.

## Design Procedure of the Combinational Circuits

(V)

The design procedure indicate the manufacture of the IC or any other digital circuits using the minimal number of gates as well as the minimal number of inputs so that the circuit will be cheaper for manufacture. The design of any combinational circuits starts from the verbal outline of the problem and ends in a required logic circuit also the simplified boolean function that may be easily obtained from the circuit as well as the K-map simplification method. Following major steps are considered during the design of any circuits.

- State the problem
- Determine the number of input as well as the output variables.
- Assign those inputs & outputs to some alphabets
- Draw the truth table and fill it.
- Obtain the simplified boolean function
- And finally according to the boolean function draw the logic circuit.

During the design process of any combinational circuit we must keep facts in our mind as given below—

- \* Minimum number of gates are used
- \* Number of inputs are also used minimum as far as possible
- \* Propagation time of the signal is also minimum.
- \* Interconnection are also reduced
- \* Driving capabilities are also limited.

After Considering above mentioned points we can easily design any combinational circuits in the economic way.

Consider a 4-to-1 multiplexer with 2-bit inputs and 1-bit output.

Let us see its block diagram below:

$\rightarrow$  Create a table for

## Code Conversion

(11)

As the digital signals are represented in terms of the binary signals 0 & 1. Also there are variety of codes for the similar discrete elements of information and so that the different codes are used by the different systems. Some time the output from one system acts as the input for other system and during that situation we must require a converter circuit for converting the similar type of input. Hence any code converter is a circuit that makes the two systems compatible even though they uses the different codes.

Here we are considering the converter that convert the BCD codes to excess-3 codes for this consider the following truth table

Input BCD codes				Output Excess-3 codes				
A	B	C	D	w	x	y	z	
0	0	0	0	0	0	1	1	1
0	0	0	1	0	1	0	0	0
0	0	1	0	0	1	0	1	
0	0	1	1	0	1	1	0	0
0	1	0	0	0	1	1	1	1
0	1	0	1	1	0	0	0	0
0	1	1	0	1	0	0	1	
0	1	1	1	1	0	1	0	0
1	0	0	0	1	0	1	1	1
1	0	0	1	1	1	0	0	0

Now corresponding to the excess code if we draw the K-map for the values w, x, y & z using the truth table we can find the simple boolean function as - below. Here we are using the four binary variables so the K-map consists of 16 squares. And also in the truth table there are only 10 are listed due to the BCD number codes. Remaining six numbers are not available and that is the reason the don't care condition is used for the further digits for simplification.

Now the K-maps for Excess 3 codes for each variables are as follows-

ab	cd	00	01	11	10
00					
01		1	1	1	
11	X	X	X	X	
10	1	1	X	X	

ab	cd	00	01	11	10
00					
01		1			
11	X	X	X	X	
10	1		X	X	

$$w = a + bc + bd$$

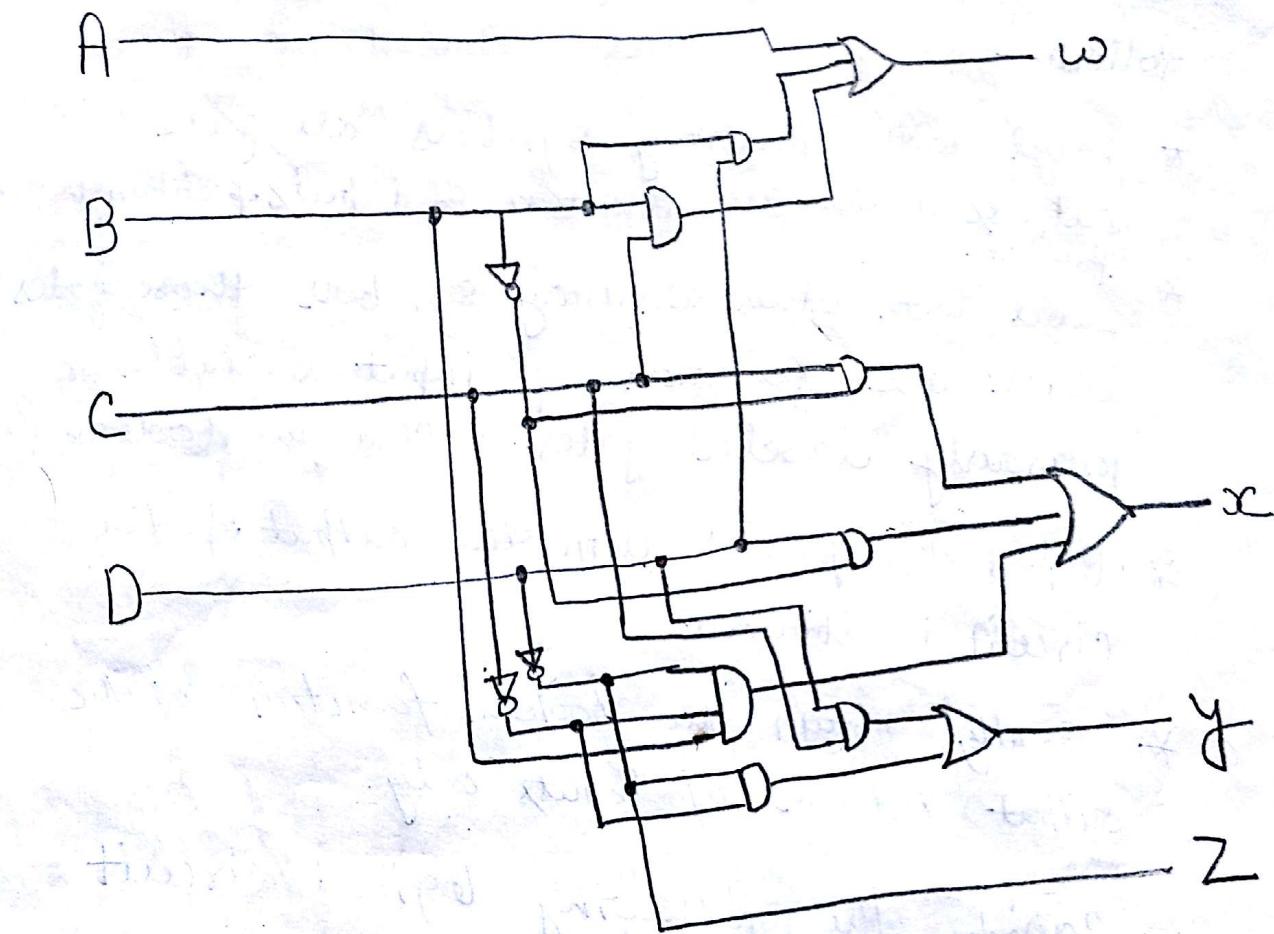
ab	cd	00	01	11	10
00					
01		1		1	
11	X	X	X	X	
10	1		X	X	

$$y = c'd' + cd$$

ab	cd	00	01	11	10
00		1		1	1
01		1			1
11	X	X	X	X	
10	1		X	X	

$$z = d'$$

The circuit diagram can be given as -



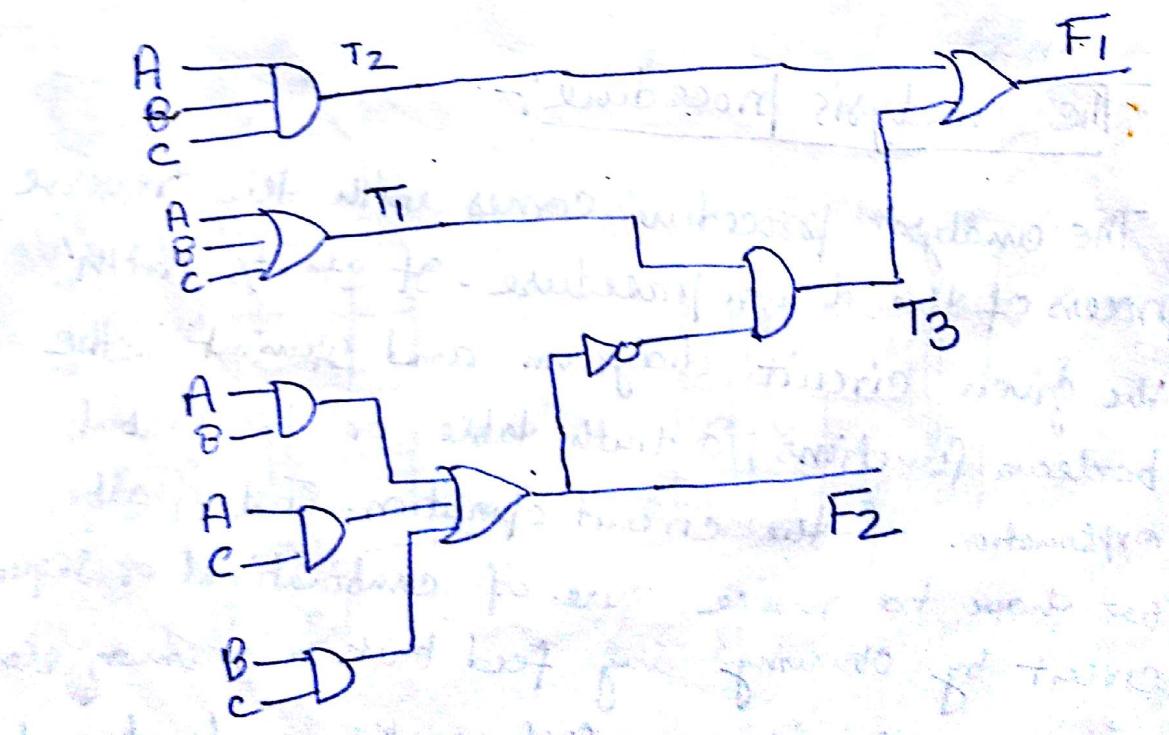
### The analysis procedure:-

The analysis procedure comes with the reverse process of the design procedure. It starts with the given circuit diagram and starts with the boolean functions, a truth table or the verbal explanation of the circuit operation. First of all we have to make sure of combinational or sequential circuit by observing any feed back or memory element. Once the circuit is verified as the combinational

circuit then the output boolean function interms of AND and OR operation as given. For obtaining the boolean function from logic diagram we proceed as follows—

- \* Label with arbitrary symbols all gate outputs which are function of inputs & obtain boolean function.
- \* Label with other arbitrary symbols those gates which are function of input variables or previously labeled gates. & find the boolean funn.
- \* Repeat the process until the output of the circuit is obtained.
- \* Finally obtain the boolean function of the output interms of inputs only.

Consider the following logical circuit



Here the given circuit has 3 inputs  $A, B, C$  and two outputs  $F_1$  &  $F_2$ . The outputs of different gates are represented by the intermediate symbols. The outputs of gates are function of input variables and are only  $F_2, T_1$  &  $T_2$ . The boolean function for those are

$$F_2 = AB + AC + BC$$

$$T_1 = A + B + C$$

$$T_2 = ABC$$

Again outputs of gates which are function of already defined symbols  $T_3$  and  $F_1$  and given by

$$T_3 = F_2' T_1$$

$$F_1 = T_3 + T_2$$

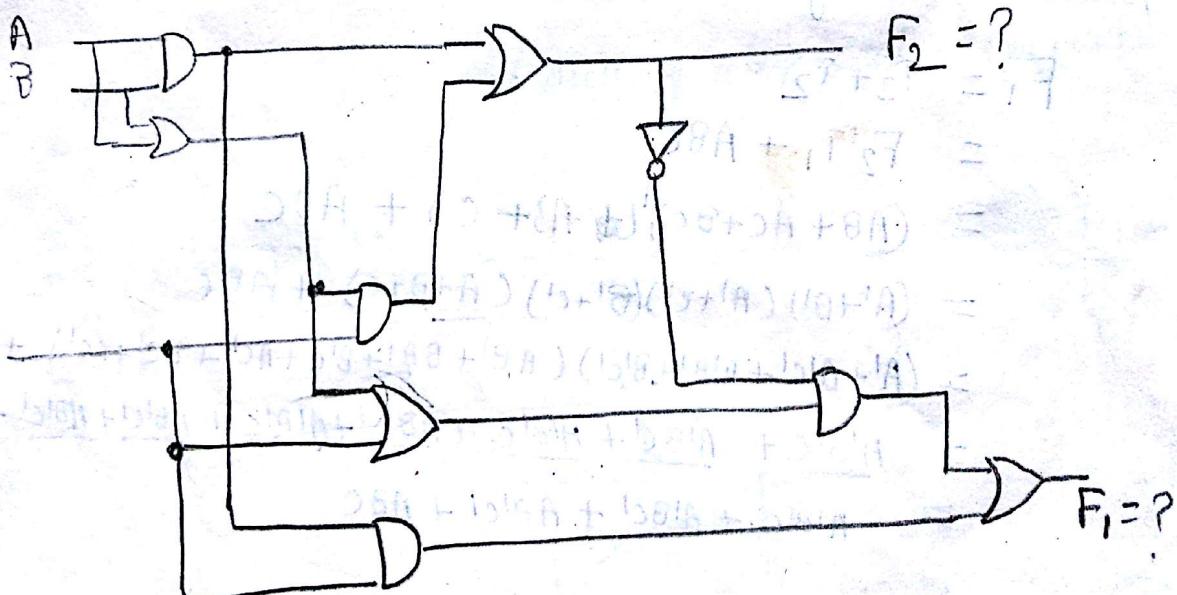
Now substituting the values

$$\begin{aligned} F_1 &= T_3 + T_2 \\ &= F_2' T_1 + ABC \\ &= (AB + AC + BC)'(A + B + C) + ABC \\ &= (A' + B')(A' + C')(B' + C')(A + B + C) + ABC \\ &= (A' + A'C' + A'B'C + B'C') (A'B + BB' + B'C + AC' + BC' + CC') + ABC \\ &= \underline{A'B'C} + \underline{A'B'C'} + \underline{A'B'C} + \underline{A'B'C} + \underline{ABC'} + \underline{ABC} + ABC \\ &= A'B'C + A'B'C' + A'B'C + A'B'C + ABC \end{aligned}$$

The truth table for the boolean function can be obtained as -

A	B	C	$F_2$	$F_2'$	$T_1 T_2 T_3$	$F_I$
0	0	0	0	1	0 0 0	0
0	0	1	0	1	1 0 1	1
0	1	0	0	1	1 0 1	1
0	1	1	1	0	1 0 0	0
1	0	0	0	1	1 0 1	1
1	0	1	1	0	1 0 0	0
1	1	0	1	0	1 0 0	0
1	1	1	1	0	1 1 0	1

In this way we can do the analysis of any circuit and can develop the boolean function as well as the truth table.

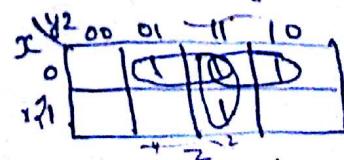
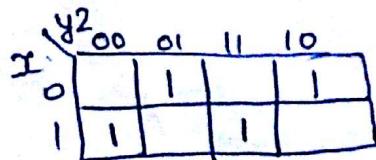


## Full Subtractor:-

Full Subtractor is a combinational circuit that performs a subtraction between two bits such that a 1 may have been borrowed by a lower significant stage. The full adder takes three inputs and two outputs. The three inputs  $x, y$  and  $z$  are minuend, subtrahend and previous borrow respectively and the two outputs are difference and borrow. The truth table for the full subtractor is as follows-

$x$	$y$	$z$	Diff.	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

Here the K-map for diff. & borrow are as follows.

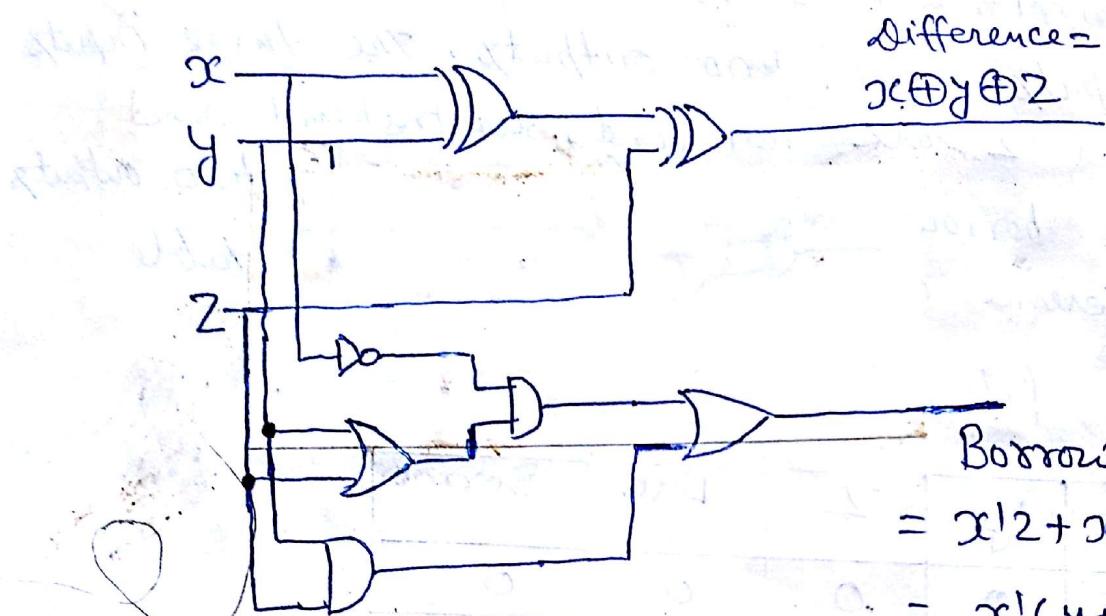


$$\text{Diff} = \bar{x}\bar{y}\bar{z} + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xy\bar{z}$$

$$\text{Borrow} = \bar{x}z + x\bar{y} + yz$$

$$= x(y+z) + yz$$

Here the boolean function for the difference is as same as the sum of the full adder and hence we can draw the logical circuit as follows -



$$\text{Difference} = x \oplus y \oplus z$$

Borrow

$$= x'z + x'y + yz \\ = x'(y+z) + yz$$

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

Ans

# Design the combinational circuit that multiplies 2 bit numbers  $a_1a_0$  &  $b_1b_0$  to produce product  $c_3c_2c_1c_0$ .

Soln

Here functional table for the circuit is given by

$a_0$	$a_1$	$b_0$	$b_1$	$c_0$	$c_1$	$c_2$	$c_3$
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Now calculating the values of  $c_0, c_1, c_2$  &  $c_3$  on the basis of minterms we get

		$b_0b_1$			
		00	01	11	10
$a_0a_1$	00	-	-	-	-
	01	-	-	-	-
11	-	-	1	-	-
10	-	-	-	-	-

$$c_0 = a_0a_1b_0b_1$$

		$b_0b_1$			
		00	01	11	10
$a_0a_1$	00	-	-	-	-
	01	-	-	-	-
11	-	-	1	1	1
10	-	-	1	1	1

$$\begin{aligned} c_1 &= a_0a_1'b_0 + a_0'b_0b_1' \\ &= a_0b_0(a_1' + b_1') \end{aligned}$$

		b <sub>0</sub> b <sub>1</sub>	c <sub>2</sub>
		00 01 11 10	00 01 11 10
		a <sub>0</sub> a <sub>1</sub>	b <sub>0</sub> b <sub>1</sub>
00			
01		1 1	1 1
11		1 1	1 1
10		1 1	1 1

		b <sub>0</sub> b <sub>1</sub>	c <sub>3</sub>
		00 01 11 10	00 01 11 10
		a <sub>0</sub> a <sub>1</sub>	b <sub>0</sub> b <sub>1</sub>
00			
01		1 1	1 1
11		1 1	1 1
10		1 1	1 1

$$C_3 = a_1 b_1$$

$$\begin{aligned}
 C_2 &= a_0 a_1' b_1 + a_0 b_0 b_1' + a_0' a_1 b_0 + a_1 b_0 b_1' \\
 &= a_0 b_1 (a_1' + b_0') + a_1 b_0 (a_0' + b_1') \\
 &= a_0 b_1 (a_1 \cdot b_0)' + a_1 b_0 (a_0 \cdot b_1)' \\
 &= a_0 b_1 \oplus a_1 b_0
 \end{aligned}$$

Hence we can draw the circuit as follow:

