MODULE 1

1. **State the base of a number system. Or Define base of a number system.**

   1 **Binary Number System** Base 2. Digits used: 0, 1

   2 **Octal Number System** Base 8. Digits used: 0 to 7

   3 **Hexa Decimal Number System** Base 16. Digits used: 0 to 9, Letters used: A- F

2. **State the number of symbols used in an octal number system.**

   Numbers from (0-7)

3. **Full form of ASCII.**
   American Standard Code for Information Interchange (ASCII).
4. **Ones complement of 1011 is <u>0100</u>**
5. **List two universal gates. Or which gates are called universal gates and why?**

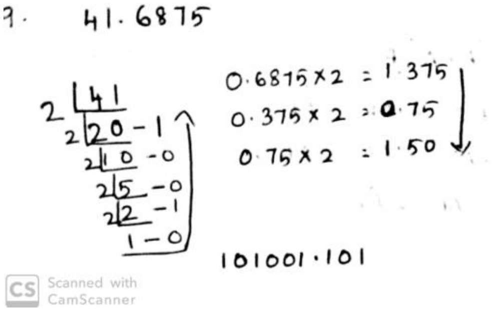
- NAND,NOR gates are universal gates because all other gates can be realized by using this

6. **Write two examples for non weighted code.**
   Excess-3 code and Gray code.
7. **What is parity bit?**

   A parity bit, or check bit, is a bit added to a string of binary code to ensure that the total number of 1-bits in the string is even or odd.[1] Parity bits are used as the simplest form of error detecting code. There are two variants of parity bits: **even parity bit** and **odd parity bit**.

8. **Define binary codes.**
   The digital data is represented, stored and transmitted as group of binary bits. This group is also called as **binary code**.

9. **Convert 41.6875$_{10}$ to binary.**



9· 41·6875

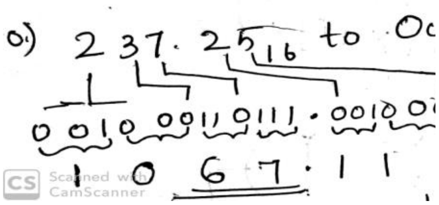
2 | 41
2 | 20 - 1
2 | 10 - 0
2 | 5 - 0
2 | 2 - 1
1 - 0

0·6875 × 2 = 1·375
0·375 × 2 = 0·75
0·75 × 2 = 1·50

101001·101

10. **Convert the following**

   I. **237.25$_{16}$ to octal**

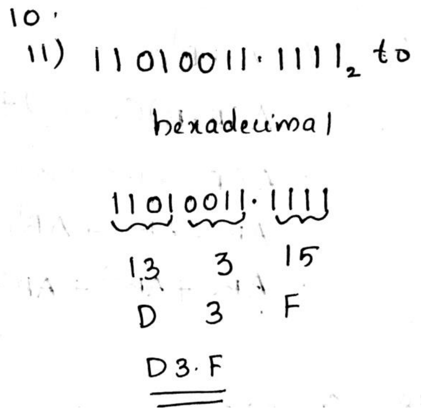


a) 237.25$_{16}$ to Oc

0 010 0011 0111 . 0010 0

1 0 6 7 . 1 1

   II. **11010011.1111$_2$ to hexadecimal**



10.

ii) 11010011.1111$_2$ to hexadecimal

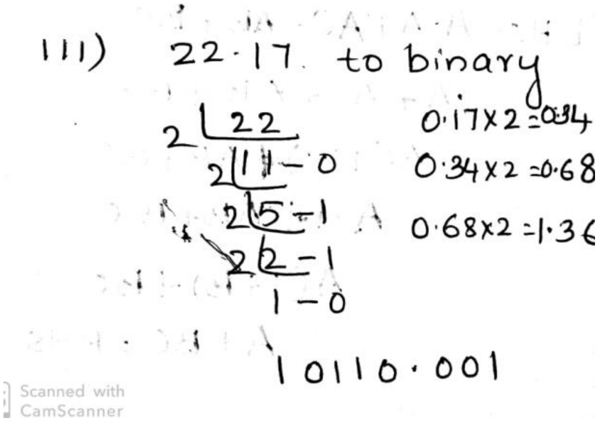
1101 0011 . 1111

13 3 15

D 3 . F

D3.F

   III. **22.17 to binary**



iii) 22.17 to binary

2 | 22
2 | 11 – 0
2 | 5 – 1
2 | 2 – 1
1 – 0

0.17×2 = 0.34
0.34×2 = 0.68
0.68×2 = 1.36

10110.001

11. **Perform the following conversions**

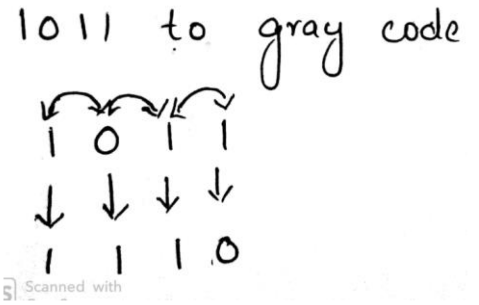
   I. **10110.0101$_2$ to hexadecimal**

   II. **F4B.11 $_{16}$ to binary**

   III. **26.24$_8$ to decimal**

   IV. **85.25 $_{10}$ to octal**

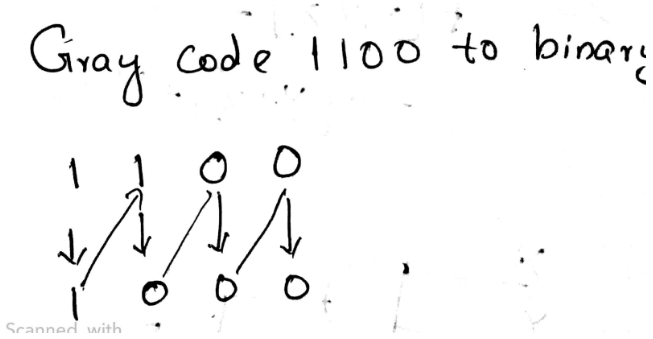
12. **Convert 1011 to gray code and gray code 1100 to binary.**



1011 to gray code

1 0 1 1

1 1 1 0

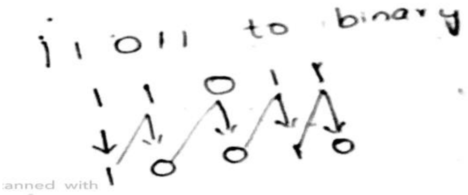


Gray code 1100 to binar

1 1 0 0

1 0 0 0

13. **Convert 10110 to gray code and 11011 to binary.**



11011 to binary

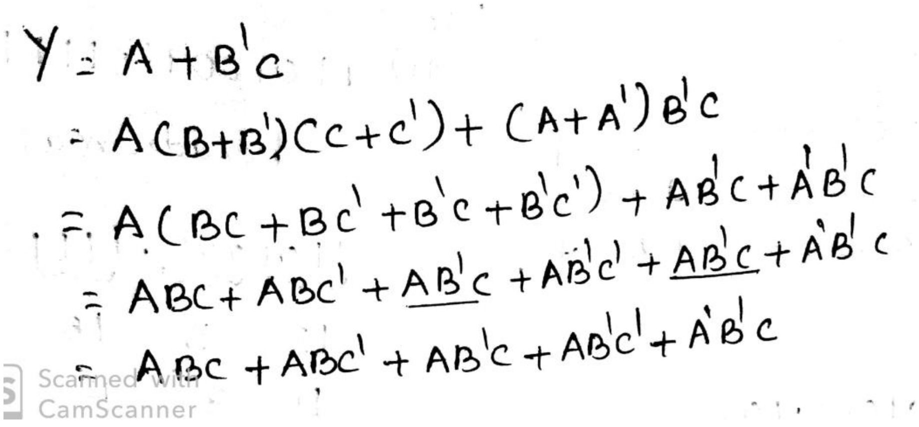
14. **Describe the 7 laws of Boolean algebra.**

Theorem	**Specific Purpose**
A+0=A	**OR OPERATION**
A+1=1	
A+A=A	
A+$\bar{A}$=1	
A.0=0	**AND OPERATION**
A.1=A	
A.A=A	
A. $\bar{A} = 0$	
$\bar{\bar{A}}$=A	**COMPLEMENT**

THEOREM	**SPECIFIC PURPOSE**
A+B=B+A	COMMUTATIVE LAW
A.B=B.A	COMMUTATIVE LAW
(A+B)+C=A+(B+C)	ASSOCIATIVE LAW
(A.B).C=A.(B.C)	ASSOCIATIVE LAW
A.(B+C)=A.B+A.C	DISTRIBUTIVE LAW
(A+B).C=A.C+B.C	DISTRIBUTIVE LAW
$\overline{A+B}$=$\bar{A}.\bar{B}$	DE MORGHAN’S LAW

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

15. Convert the following SOP into standard SOP Y=A+B’C



$Y = A + B'C$

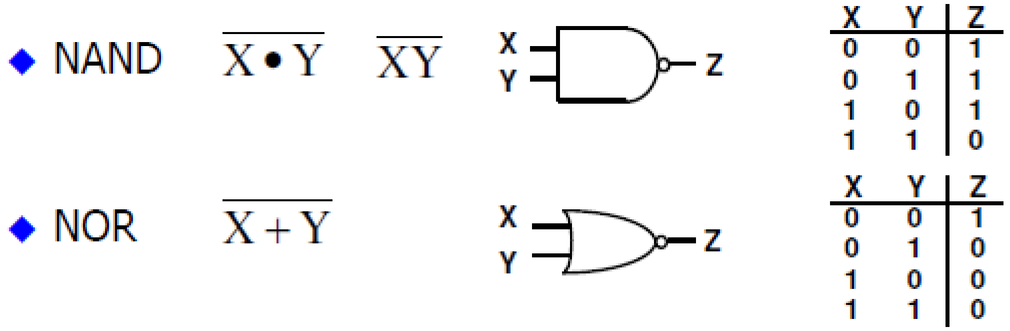
$= A(B+B')(C+C') + (A+A')B'C$

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

$= ABC + ABC' + \underline{AB'C} + AB'C' + \underline{AB'C} + A'B'C$

$= ABC + ABC' + AB'C + AB'C' + A'B'C$

16. Draw the logic symbol and truth table for universal gates.



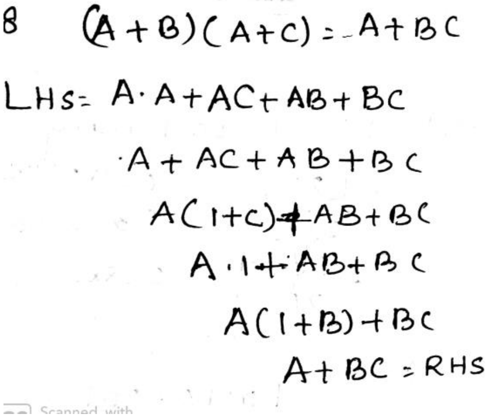
- NAND $\overline{X \bullet Y}$ $\overline{XY}$

X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

- NOR $\overline{X + Y}$

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

17. State any four theorems of Boolean algebra.Using the theorems of Boolean algebra,prove the following’(A+B)(A+C)=A+BC.



8 $(A+B)(A+C) = A+BC$

$LHS = A \cdot A + AC + AB + BC$

$A + AC + AB + BC$

$A(1+C) + AB + BC$

$A \cdot 1 + AB + BC$

$A(1+B) + BC$

$A + BC = RHS$

18. State the advantage of performing subtraction with complement method.Perform 2’S complement subtraction for the following binary numbers.

I)110000-10101 ii)1001-101000(solution is in or class note)

- The advantage of performing subtraction by the complement method is reduction in the hardware . Instead of having separate digital circuits for addition and subtraction, only adding circuits are needed.
- That is, subtraction is also performed by adders only. Instead of subtracting one number from the other, the complement of the subtrahend is added to the minuend.
- In sign-magnitude form, an additional bit called the sign bit is placed in front of the number.
- If the sign bit is a 0, the number is positive. If it is a 1, the number is negative. 28

19. Simplify the following boolean functions to a minimum number of literals.Also implement the Boolean functions with gates.

a)F(X,Y,Z)=(X+Y)(Y+Z)

20 a) $\cdot F(X,Y,Z) = (X+Y)(Y+Z)$

$= XY + XZ + Y \cdot Y + YZ$

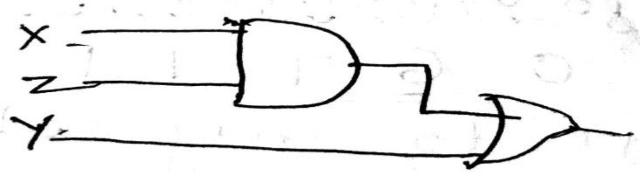
$= XY + XZ + Y + YZ$

$= XY + XZ + Y[1+Z]$

$= XY + XZ + Y \cdot 1$

$= Y[X+1] + XZ$

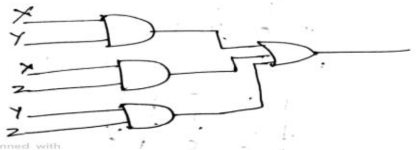
$= Y + XZ$



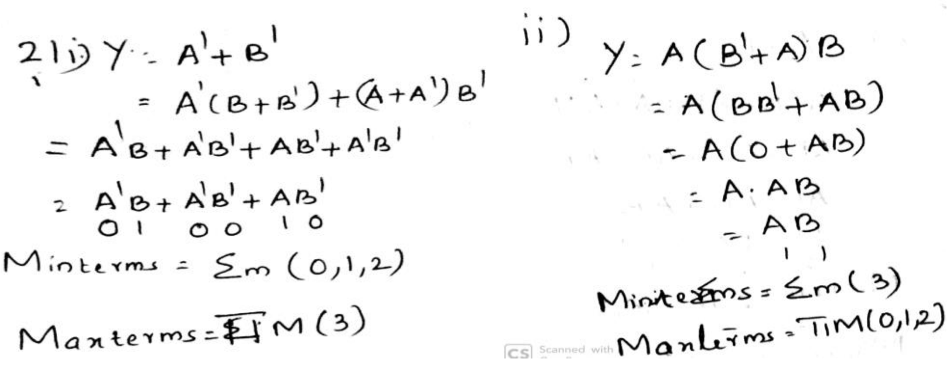
b) F(X,Y,Z)=XY+X’Z+YZ

20 b. $F[X,Y,Z] = XY + XZ + YZ$

$= X[Y+Z] + YZ$



20. Express the following Boolean expressions in minterms and maxterms.
i)A’+B’ ii)A(B’+A)B



21 i) $Y = A' + B'$

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

$= A'B + A'B' + AB' + A'B'$

$= A'B + A'B' + AB'$

$\quad 01 \quad\; 00 \quad\; 10$

Minterms $= \Sigma m(0,1,2)$

Maxterms $= \Pi M(3)$

ii) $Y = A(B'+A)B$

$= A(BB' + AB)$

$= A(0 + AB)$

$= A \cdot AB$

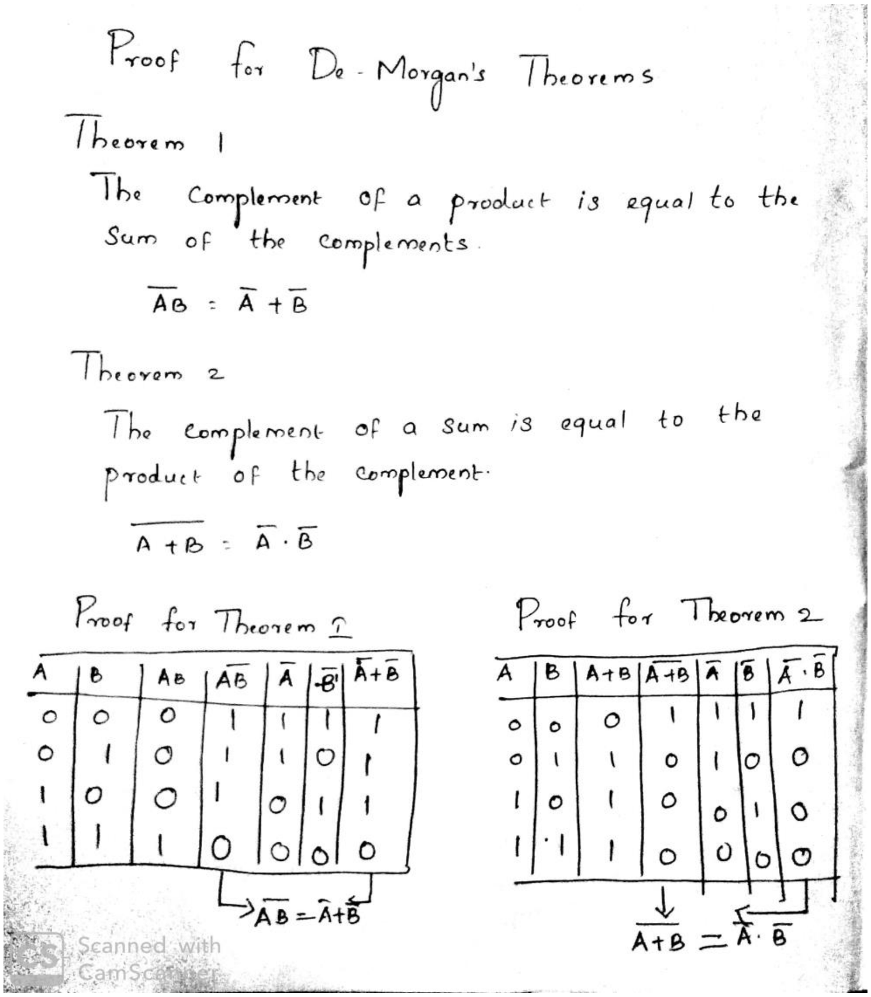
$= AB$

$\quad 1\;1$

Minterms $= \Sigma m(3)$

Maxterms $= \Pi M(0,1,2)$

21. State and prove De Morgan’s theorem.



Proof for De-Morgan's Theorems

Theorem 1

The Complement of a product is equal to the Sum of the complements.

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

Theorem 2

The complement of a sum is equal to the product of the complement.

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

Proof for Theorem I

A	B	AB	$\overline{AB}$	$\bar{A}$	$\bar{B}'$	$\bar{A}+\bar{B}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

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

Proof for Theorem 2

A	B	$A+B$	$\overline{A+B}$	$\bar{A}$	$\bar{B}$	$\bar{A}\cdot\bar{B}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

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

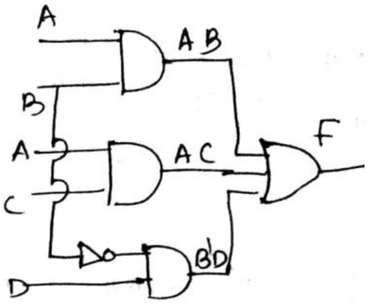
22. Reduce and draw the logic diagram for the Boolean expression F=A(B+C)+B'(B+D)

23 $F = A(B+C) + B'(B+D)$

$= AB + AC + B'B + B'D$

$= AB + AC + 0 + B'D$

$= AB + AC + B'D$



23. Reduce the expression F=A+B(AC+(B+C')D)

24 $F = A + B[AC + (B+C')D]$

$= A + ABC + BBD + BC'D$

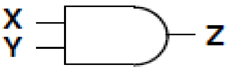
$= A + ABC + BD + BC'D$

$= A(1+BC) + BD(1+C')$

$= A + BD$

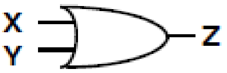
24. Explain basic gates with truth table and logic diagram.

◆ AND X•Y XY



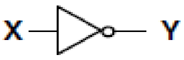
X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

◆ OR X+Y



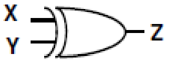
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

◆ NOT $\overline{X}$ X'



X	Y
0	1
1	0

◆ XOR $X \oplus Y$



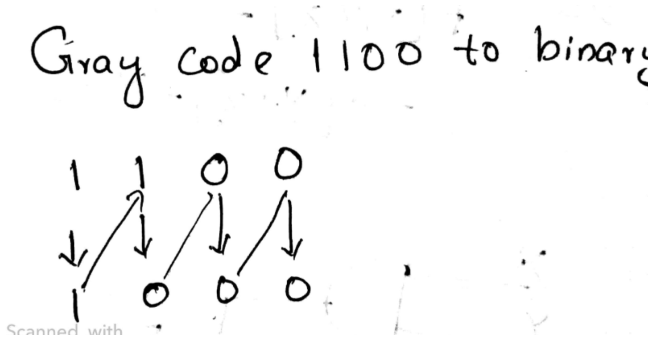
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

25. Explain the procedure of obtaining an equivalent gray code for a binary code with example.

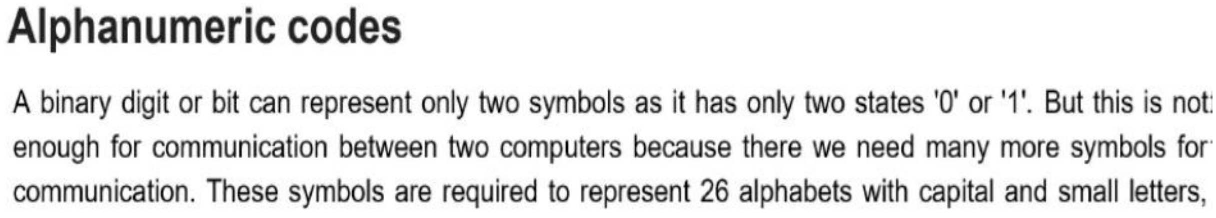
o change gray to binary code, take down the MSB digit of the gray code number, as the primary digit or the MSB of the gray code is similar to the binary digit.

To get the next straight binary bit, it uses the XOR operation among the primary bit or MSB bit of binary to the next bit of the gray code.

Similarly, to get the third straight binary bit, it uses the XOR operation among the second bit or MSB bit of binary to the third MSD bit of the gray code and so on.

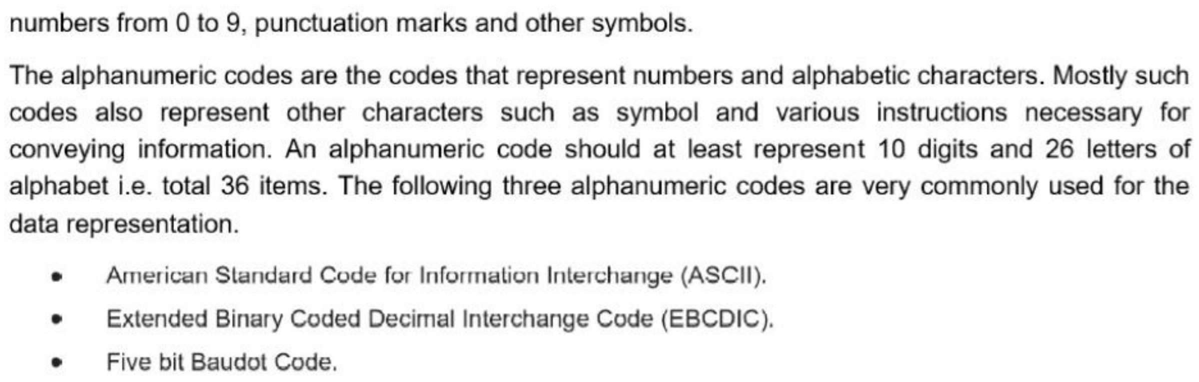


26. Explain about alpha numeric codes with examples.



Alphanumeric codes

A binary digit or bit can represent only two symbols as it has only two states '0' or '1'. But this is not enough for communication between two computers because there we need many more symbols for communication. These symbols are required to represent 26 alphabets with capital and small letters,

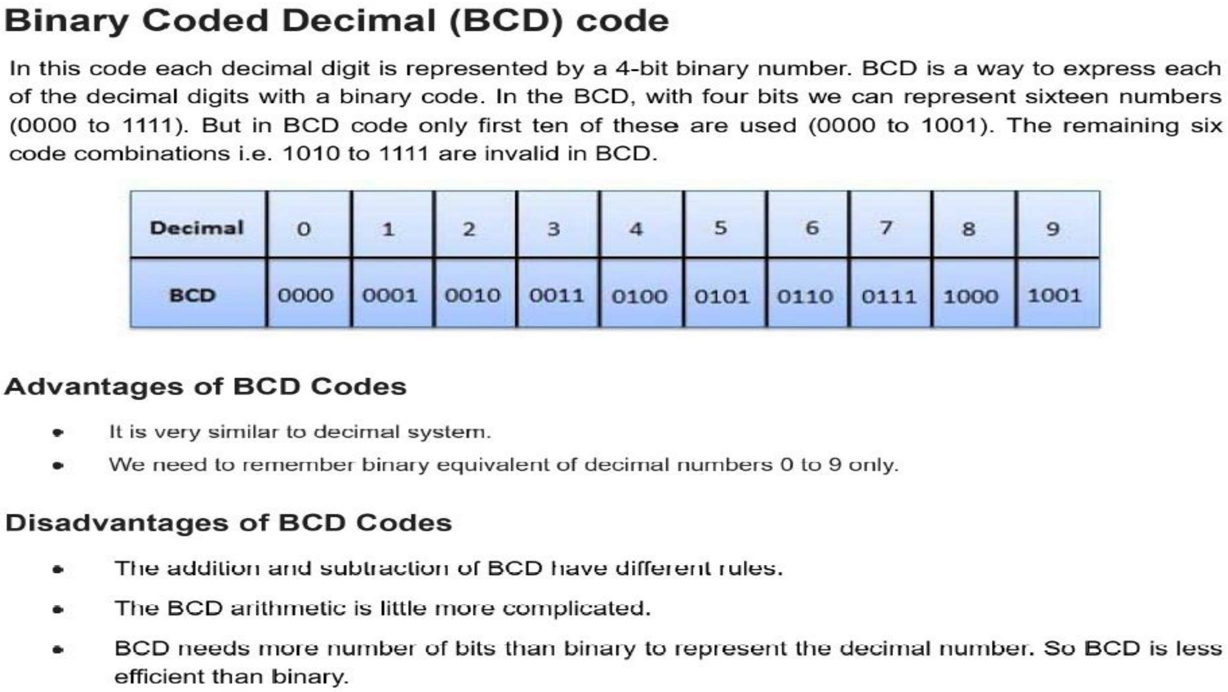


numbers from 0 to 9, punctuation marks and other symbols.

The alphanumeric codes are the codes that represent numbers and alphabetic characters. Mostly such codes also represent other characters such as symbol and various instructions necessary for conveying information. An alphanumeric code should at least represent 10 digits and 26 letters of alphabet i.e. total 36 items. The following three alphanumeric codes are very commonly used for the data representation.

- American Standard Code for Information Interchange (ASCII).
- Extended Binary Coded Decimal Interchange Code (EBCDIC).
- Five bit Baudot Code.

27. Write short notes on BCD codes.



Binary Coded Decimal (BCD) code

In this code each decimal digit is represented by a 4-bit binary number. BCD is a way to express each of the decimal digits with a binary code. In the BCD, with four bits we can represent sixteen numbers (0000 to 1111). But in BCD code only first ten of these are used (0000 to 1001). The remaining six code combinations i.e. 1010 to 1111 are invalid in BCD.

Decimal	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

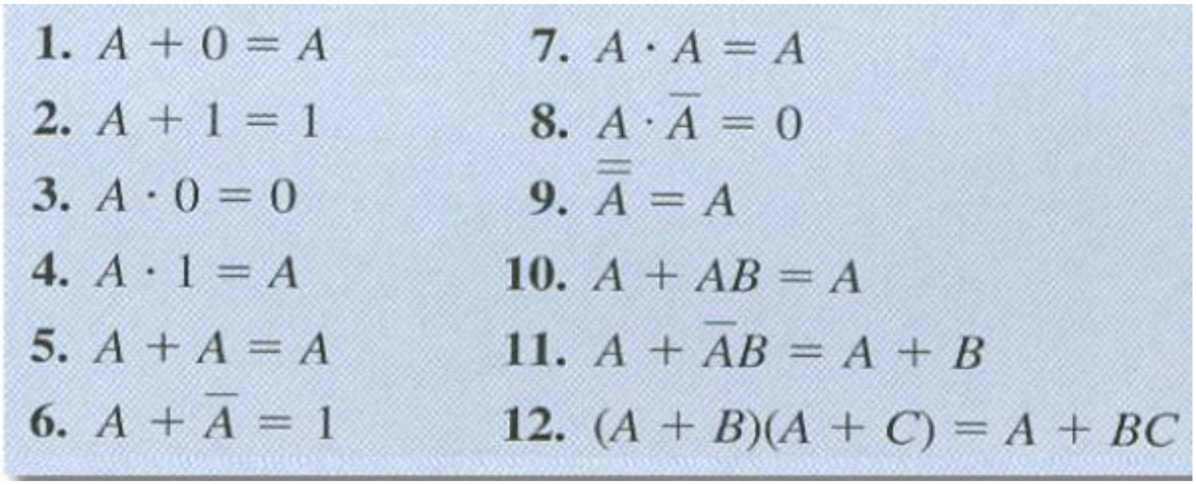
Advantages of BCD Codes

- It is very similar to decimal system.
- We need to remember binary equivalent of decimal numbers 0 to 9 only.

Disadvantages of BCD Codes

- The addition and subtraction of BCD have different rules.
- The BCD arithmetic is little more complicated.
- BCD needs more number of bits than binary to represent the decimal number. So BCD is less efficient than binary.

28. Rules of Boolean algebra

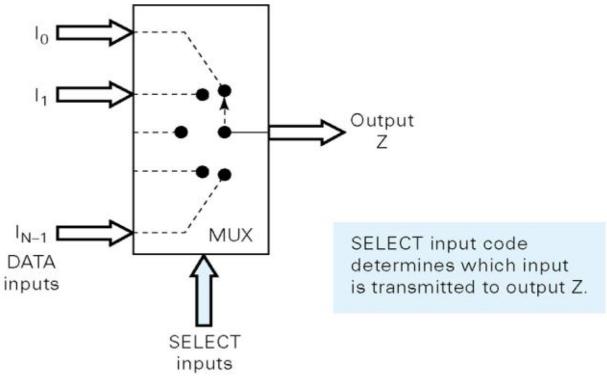


1. $A + 0 = A$
2. $A + 1 = 1$
3. $A \cdot 0 = 0$
4. $A \cdot 1 = A$
5. $A + A = A$
6. $A + \overline{A} = 1$
7. $A \cdot A = A$
8. $A \cdot \overline{A} = 0$
9. $\overline{\overline{A}} = A$
10. $A + AB = A$
11. $A + \overline{A}B = A + B$
12. $(A + B)(A + C) = A + BC$

MODULE 2

1. Define a multiplexer.

- A circuit that directs one of several digits signals to a single output depending upon the state of several select inputs.
- **Data** Inputs:The mux inputs that feed a digital signal to the output when selected.(Maximum of inputs is $2^{n)}$
- **Select** Inputs:The Mux inputs that selcts the digital input(Maximum n select lines)



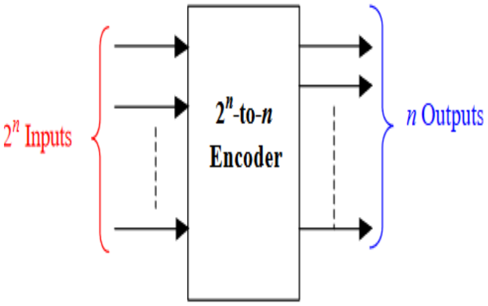
SELECT input code determines which input is transmitted to output Z.

2. What is don’t care condition?Mention its use.

Don’t cares in a Karnaugh map, or truth table, may be either **1**s or **0**s, as long as we don’t care what the output is for an input condition we never expect to see.

3. Define Encoder.

An encoder performs the inverse operation of a decoder.
oIt has 2^n inputs, and n output lines.
oOnly one input can be logic 1 at any given time (active input). All other inputs must be 0’s.
o Output lines generate the binary code corresponding to the active input.



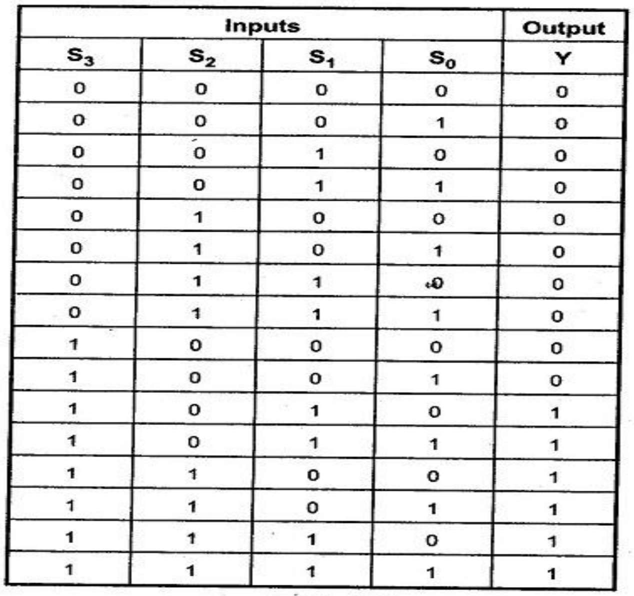
Applications of Encoder

1.Keyboard encoder for computers
2.Optical encoders –linear or rotary
3.Interfacing peripherals to microprocessors
4.Audio/video coding and transmission

4. Define minterms

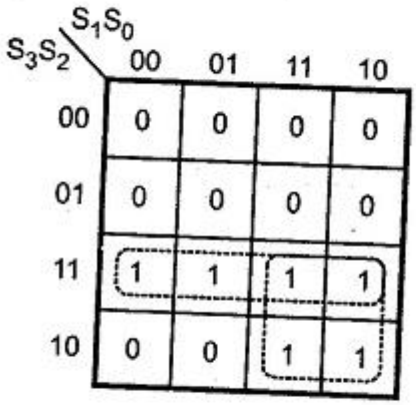
Products of boolean expression where all possible variables appear once in complement or uncomplemented variables.
A minterm is a product terms (AND term) that includes all the variables in either complemented or uncomplementedform.A two variable function has four possible combinations(2^2=4) ie.,$\bar{A}\bar{B}$,$\bar{A}B$, $A\bar{B}$, AB.These product terms are called minterm.

5. **Demonstrate a one bit comparator.**
6. **Design a 2 bit magnitude comparator .**
7. **Design and implement a 3 bit binary to gray code converter.**
   (See answers in your class note for the above questions.)
8. **Design a full adder.**
9. **Design a decimal adder with truth table and logic diagram.**



Inputs				Output
S_3	S_2	S_1	S_0	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Table 3.10



S_3S_2 \ S_1S_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	1	1

Fig. 3.31

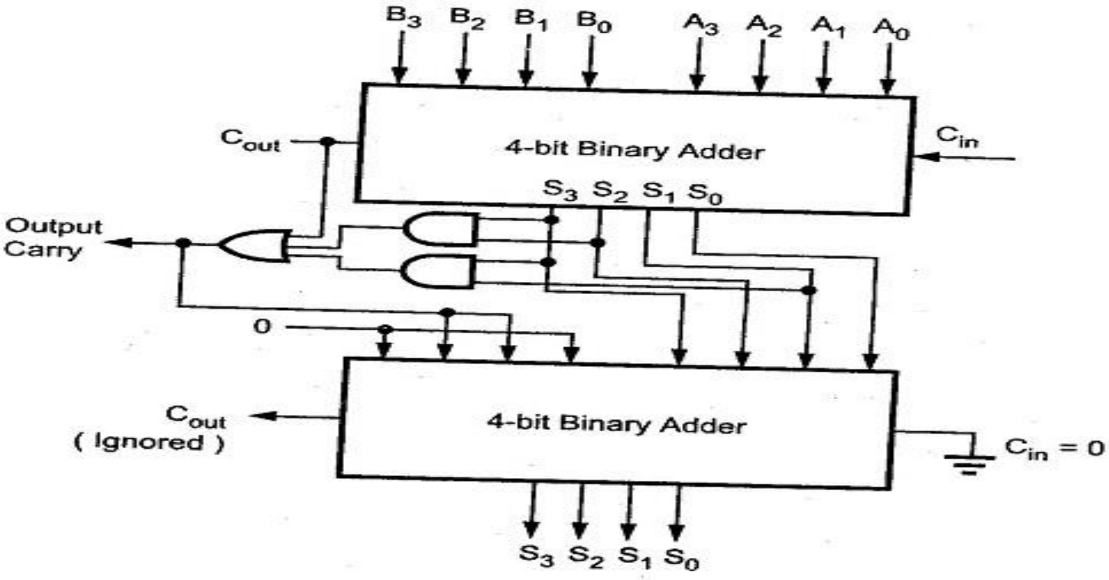
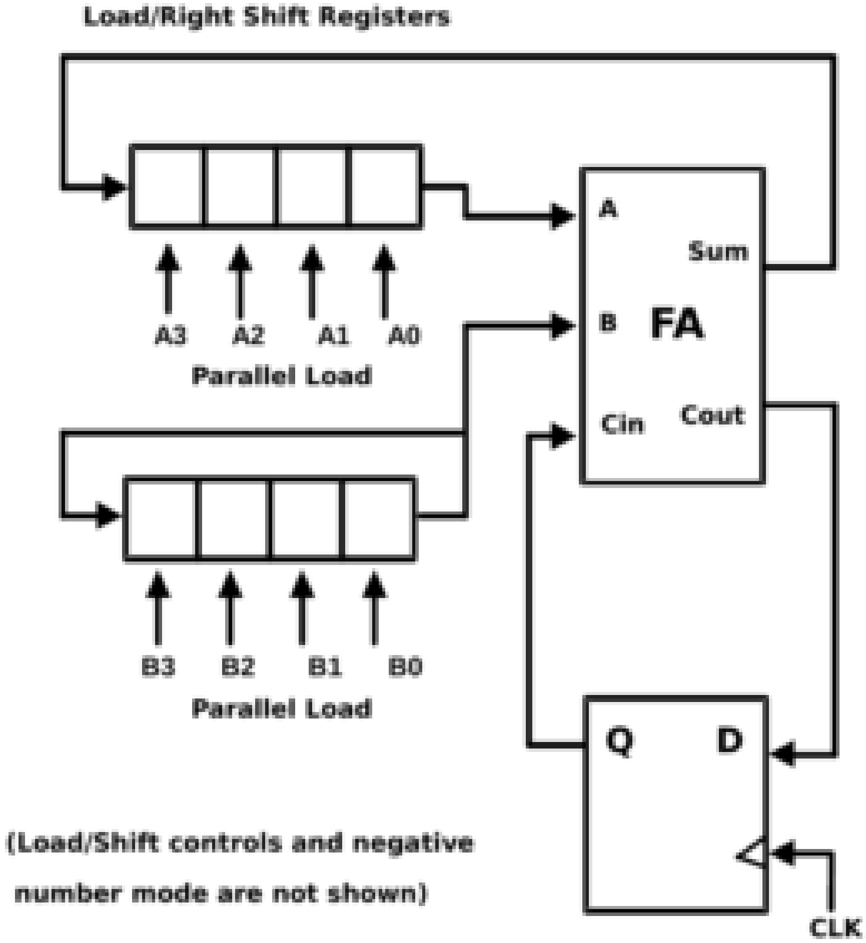


Fig. 3.32 Block diagram of BCD adder

10. Illustrate the working of a serial adder.

A 4-bit serial adder circuit consists of two 4-bit shift registers with parallel load, a full adder, and a D-type flip-flop for storing carry-out.
In order to load registers A_REG and B_REG with numbers, shift capability of the registers should be disabled and loading mode should be enabled. Loading of numbers from inputs A, B to registers A_REG, B_REG occurs in one clock cycle. After loading registers with numbers, shifting mode should be enabled to perform the arithmetic operation. The addition of numbers stored in A_REG and B_REG requires 4 cycles. Starting with the least significant bit, at each cycle one bit of number A and one bit of number B are being added. The sum is stored at the most significant bit of register A_REG. Carry-out output produced after each cycle is fed back to the full adder as a carry-in of the next significant bit. For this purpose one D-type flip-flop is used as a temporary storage element. The least significant bit of B_REG is fed to the input of the most significant bit of B_REG. Hence the circuit performs rotation operation for register B_REG.


Load/Right Shift Registers
A3
A2
A1
A0
Parallel Load
B3
B2
B1
B0
Parallel Load
A
Sum
B
FA
Cin
Cout
Q
D
CLK
(Load/Shift controls and negative
number mode are not shown)

11. Demonstrate a 4 bit adder-subtractor with suitable neat diagram.

Subtraction of binary numbers can be carried out by using the addition of 2's complement of subtrahend to the minuend. In this operation If the MSB of addition is a '0', then the answer is correct and if MSB is '1', then answer is having negative sign. Hence, by using full adders subtraction can be carried out.

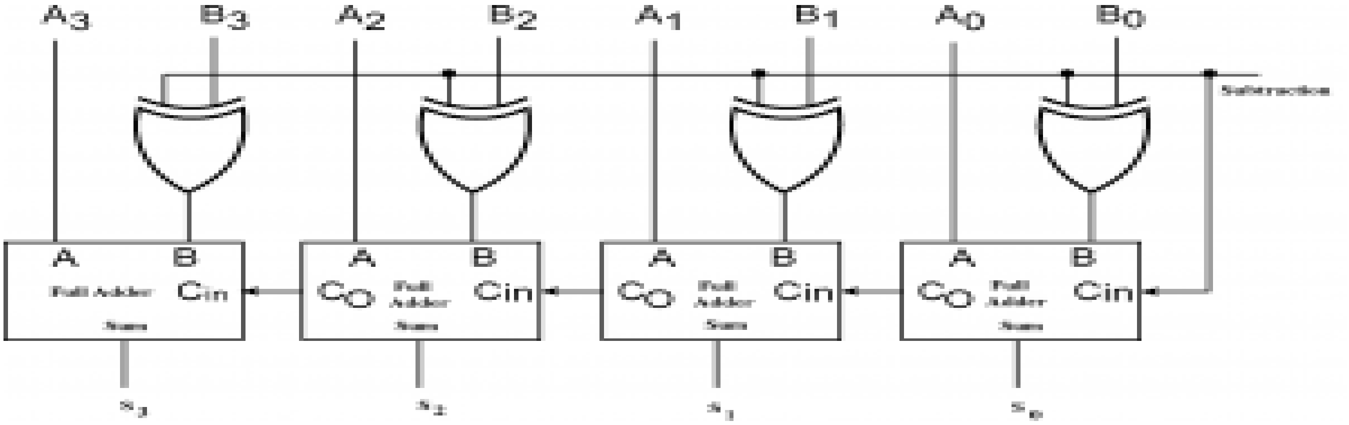


Figure above the realization of 4 bit adder-subtractor. From the figure it can be seen that, the bits of the binary numbers are given to full adder through the XOR gates. The control input is controls the addition or subtraction operation.

When the SUBTRACTION input is logic '0' , the B3 B2 B1 B0 are passed to the full adders. Hence, the output of the full adders is the addition of the two numbers.

When the SUBTRACTION input is logic '1' , the B3 B2 B1 B0 are complemented. Further, the SUBTRACTION logic '1' input is also work as Cin for the LSB full adder, due to which 2's complement addition can be carried out. Hence, the outputs of the full adders in this case is the subtraction of two numbers.

12. List the merits and demerits of K map.

MERITS

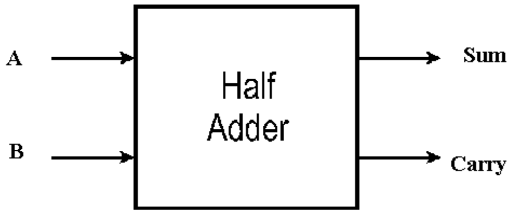
1. K-map simplification does not demand for the knowledge of Boolean algebraic theorems.
2. Usually it requires less number of steps when compared to algebraic minimization technique.

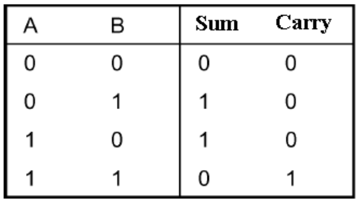
DEMERITS

1. Complexity of **K-map** simplification process increases with the increase in the number of variables
2. The minimum expression obtained might not be unique

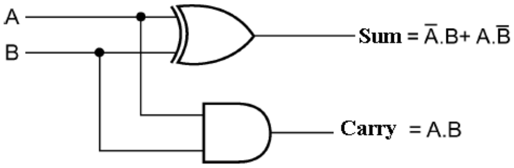
13. Design a half adder with truth table and logic diagram.

Half Adder: is a combinational circuit that performs the addition of two bits, this circuit needs two binary inputs and two binary outputs.





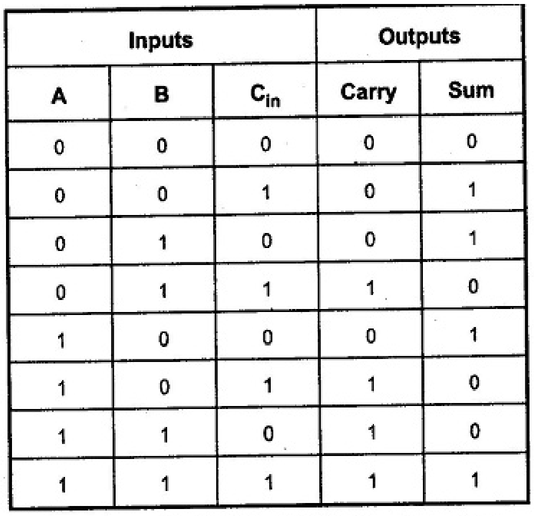
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



14. Design a full adder circuit using to half adder.Realize it using logic diagram and block diagram.

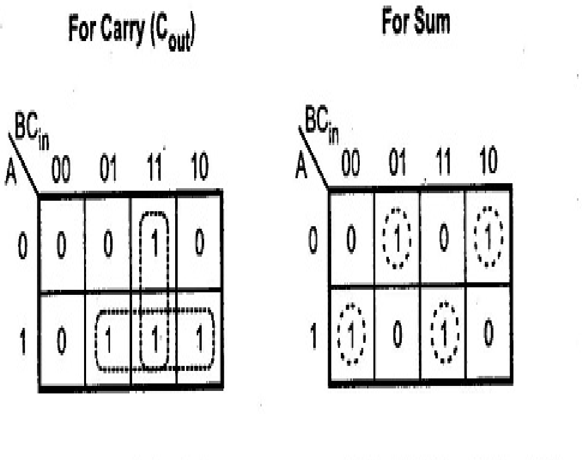
Full Adder is a combinational circuit that performs the addition of three bits (two significant bits and previous carry).

It consists of ***three inputs and two outputs***, two inputs are the bits to be added, the third input represents the carry form the previous position.



Inputs			Outputs	
A	B	C_{in}	Carry	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Table 3.7 Truth table for full-adder



For Carry (C_{out})

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

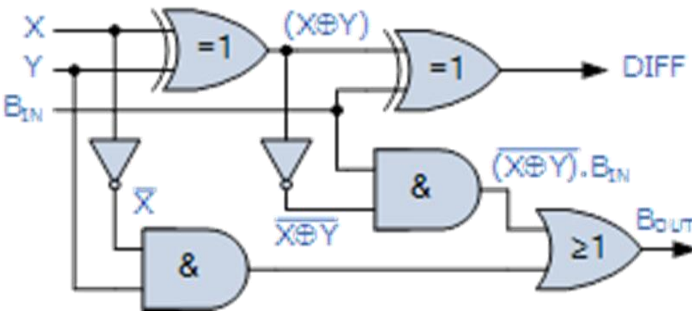
For Sum

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

$C_{out} = AB + A\,C_{in} + B\,C_{in}$

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

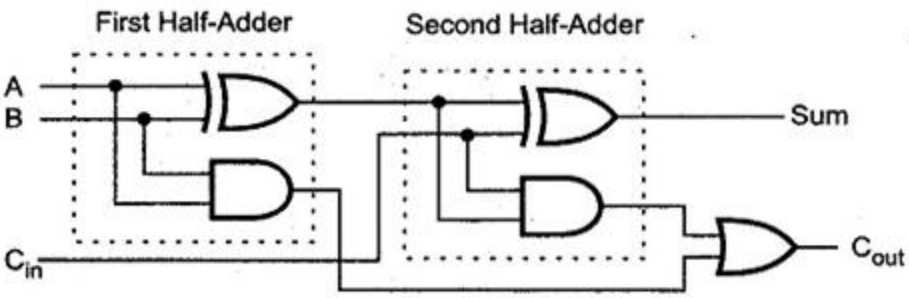
$$
\begin{aligned}
\text{Sum} &= \bar{A}\,\bar{B}\,C_{in} + \bar{A}\,B\,\bar{C}_{in} + A\,\bar{B}\,\bar{C}_{in} + A \\
&= C_{in}\,(\bar{A}\,\bar{B} + AB) + \bar{C}_{in}\,(\bar{A}\,B + A\,\bar{B} \\
&= C_{in}\,(A \odot B) + \bar{C}_{in}\,(A \oplus B) \\
&= C_{in}\,(\overline{A \oplus B}) + \bar{C}_{in}\,(A \oplus B) \\
&= C_{in} \oplus (A \oplus B)
\end{aligned}
$$



Carry= $ABC_{in} + AB\overline{C_{in}} + A\bar{B}\,C_{in} + \bar{A}\,BC_{in}$

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

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

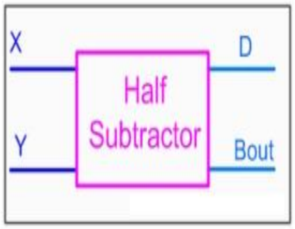


15. Design ahalfsubtractor

Half Subtractor

The half-subtractor is a combinational circuit which is used to perform subtraction of two bits. It has two inputs, X (minuend) and Y (subtrahend) and two outputs D (difference) and B (borrow). The logic symbol and truth table are shown below.

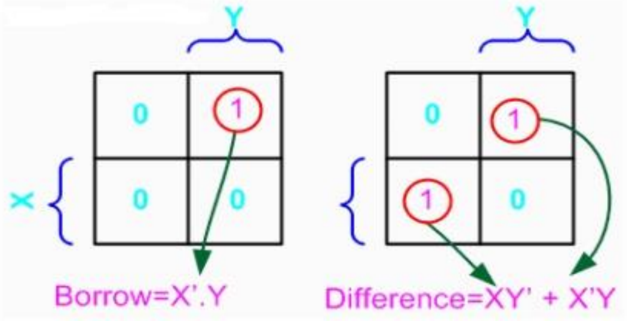
Symbol



Truth Table

X	Y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

From the above table we can draw the Kmap as shown below for "difference" and "borrow". The boolean expression for the difference and Borrow can be written.

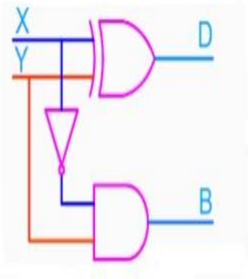


0	1
0	0

Borrow=X'.Y

0	1
1	0

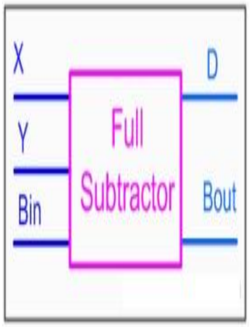
Difference=XY' + X'Y

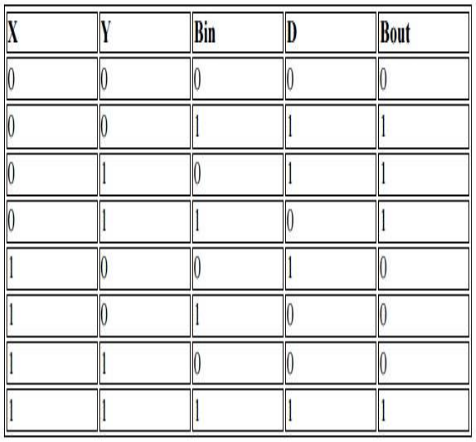


16. Design a full subtractor.

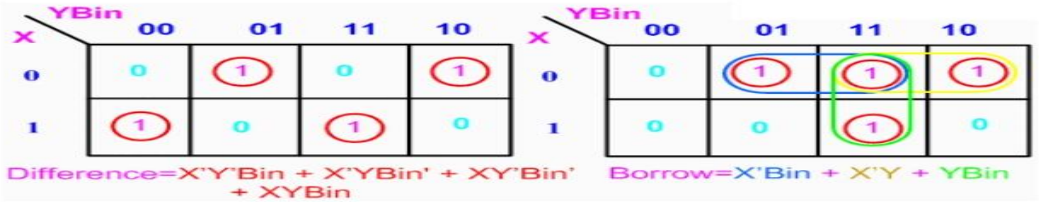
Symbol

Truth Table





X	Y	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



X \ YBin	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Difference=X'Y'Bin + X'YBin' + XY'Bin' + XYBin

X \ YBin	00	01	11	10
0	0	1	1	1
1	0	0	1	0

Borrow=X'Bin + X'Y + YBin

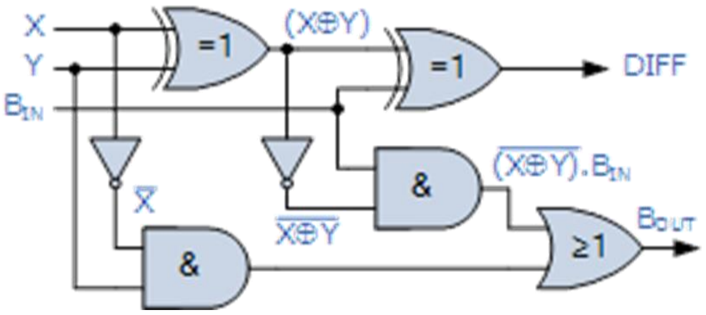
$$D = X'Y'Bin + X'YBin' + XY'Bin' + XYBin$$

$$= (X'Y' + XY)Bin + (X'Y + XY')Bin'$$

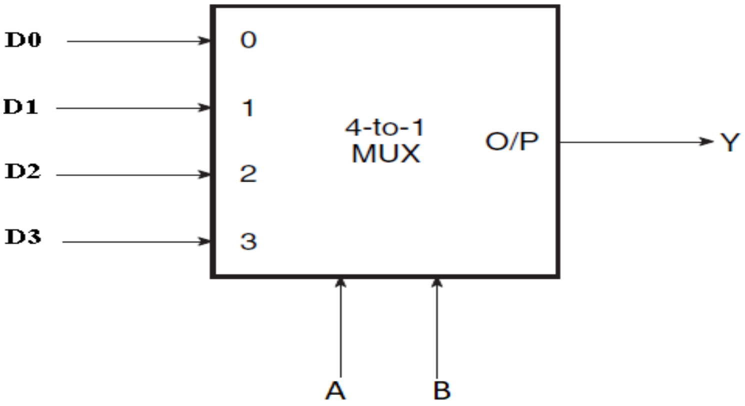
$$= (X\oplus Y)'Bin + (X\oplus Y)Bin'$$

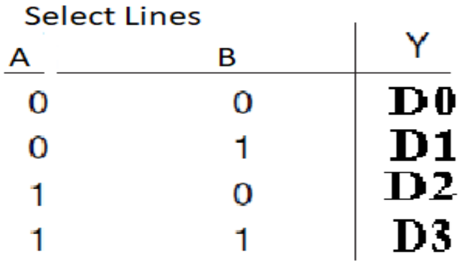
$$= X\oplus Y\oplus Bin$$

$$Bout = X'.Y + X'.Bin + Y.Bin$$



17. Design and explain the working of a 4 input multiplexer with the help of logic diagram.Write its applications.

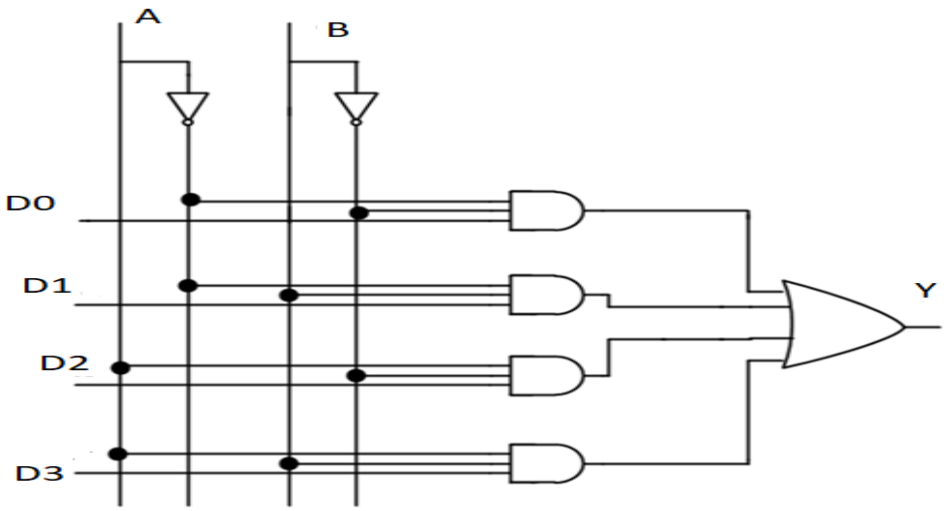




Select Lines		
A	B	Y
0	0	D0
0	1	D1
1	0	D2
1	1	D3

$$Y{=}D0\overline{A}\overline{B}+D1\overline{A}B+D2A\overline{B}+D3AB$$

- When Select lines AB = 00, the upper AND gate is enabled while all other AND gates are disabled. Therefore, data bit D0 is transmitted to the output giving Y = Do.--If the Select lines is changed to AB =11, all gates are disabled except the bottom AND gate. In this case, D3 is transmitted to the output



1. When AB=00 is applied to the select lines,the AND gate associated with D0 will have two of its inputs equal to 1 and first input connected Do.

2. The other 3 AND gates have 0 in atleast one of their inputs,which makes the output is zero.

3. Hence the OR output Y is equal to the value of D0

4. If AB=01 is applied to the select lines,the data on the input D1 appears on the data output line.

5. If AB=10 is applied to the select lines,the data on the input D2 appears on the data output

6. If AB=11 is applied to the select lines,the data on the input D3 appears on the data output.

Applications of Multiplexer:

1.Data routing

2. Data bussing

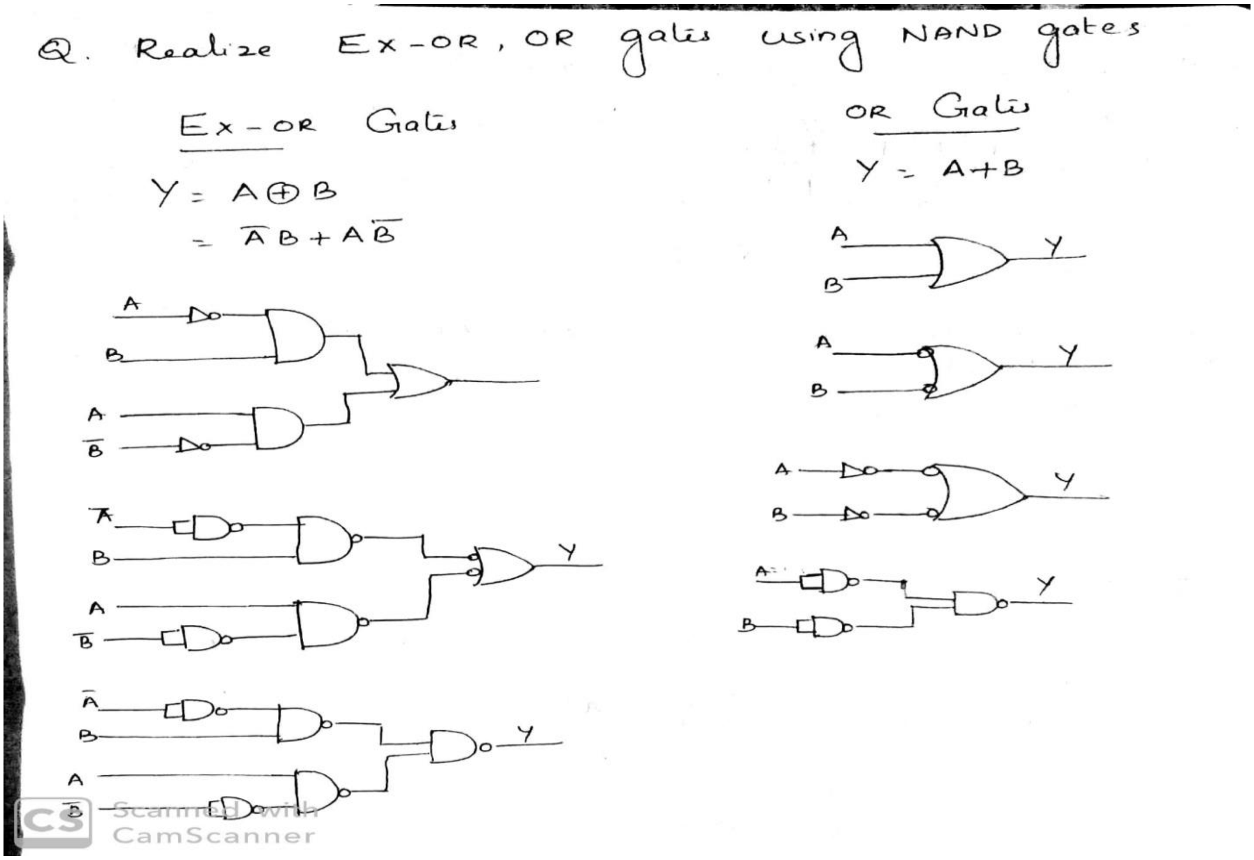
3. Switch setting comparator

4. Multiplexer as a function generator

5. Parallel to serial converter

6. Cable TV signal distribution

18. Realize X-0R ,OR gate using NAND gates.



Q. Realize Ex-OR, OR gates using NAND gates

Ex-OR Gates

$Y = A \oplus B$

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

OR Gates

$Y = A+B$

19. Write short note on SOP and POS.

Sum of Product (SOP) Form

The sum-of-products (SOP) form is a method (or form) of simplifying the Boolean expressions of logic gates. In this SOP form of Boolean function representation, the variables are operated by AND (product) to form a product term and all these product terms are ORed (summed or added) together to get the final function.

A sum-of-products form can be formed by adding (or summing) two or more product terms using a Boolean addition operation. Here the product terms are defined by using the AND operation and the sum term is defined by using OR operation.

The sum-of-products form is also called as Disjunctive Normal Form as the product terms are ORed together and Disjunction operation is logical OR. Sum-of-products form is also called as Standard SOP.

SOP form representation is most suitable to use them in FPGA (Field Programmable Gate Arrays).

Examples

AB + ABC + CDE

(AB)¯ + ABC + CD E¯

Product of Sums (POS) Form

The product of sums form is a method (or form) of simplifying the Boolean expressions of logic gates. In this POS form, all the variables are ORed, i.e. written as sums to form sum terms.

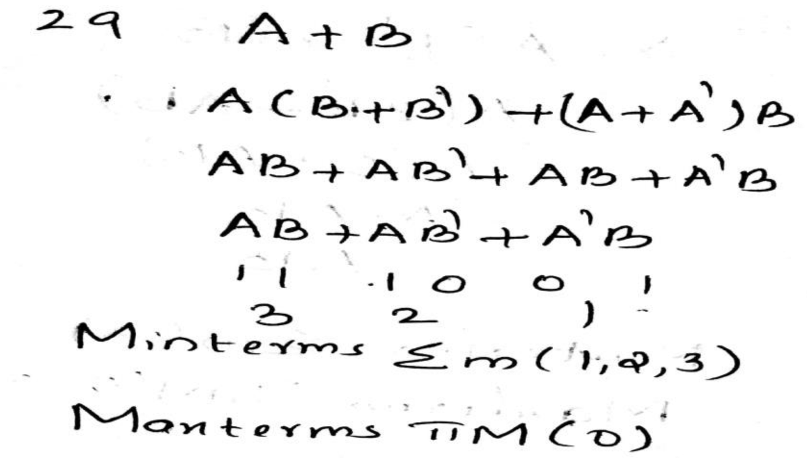
All these sum terms are ANDed (multiplied) together to get the product-of-sum form. This form is exactly opposite to the SOP form. So this can also be said as “Dual of SOP form”.

Here the sum terms are defined by using the OR operation and the product term is defined by using AND operation. When two or more sum terms are multiplied by a Boolean OR operation, the resultant output expression will be in the form of product-of-sums form or POS form.

The product-of-sums form is also called as Conjunctive Normal Form as the sum terms are ANDed together and Conjunction operation is logical AND. Product-of-sums form is also called as Standard POS.

Examples - (A+B) * (A + B + C) * (C +D)

20. **Expand A+B to minterm and maxterm.**



29 $A+B$

$A(B+B')+(A+A')B$

$AB+AB'+AB+A'B$

$AB+AB'+A'B$

$11 \quad 10 \quad 01$

$3 \quad 2 \quad 1$

Minterms $\Sigma m(1,2,3)$

Manterms $\Pi M(0)$

21. **Reduce the expression f=$\sum m(0,2,3,4,5,6)$ using K map.**
22. **Map the expression f=A’B’C+AB’C+A’BC’+ABC’+ABC.**
23. **Reduce the expression f(A,B,C)=$\sum m(0,1,2,3,4,5,6,7)$**
24. **Simplify the following function using K map and draw the logic circuit for the simplified function.F=(A,B,C,D)=$\sum 2,4,6,10,12+d(0,8,9,13)$**
25. **F=(A,B,C,D)=$\sum 4,5,7,12,14,15+d(3,8,10)$**
26. **F=(A,B,C,D)=$\sum 1,4,7,10,13+d(0,8,9,13)$**
27.