Syllabus- Module2

Karnaugh maps: minimum forms of switching functions, two and three variable Karnaugh maps, four variable karnaugh maps, determination of minimum expressions using essential prime implicants, Quine-McClusky Method: determination of prime implicants, The prime implicant chart, petricks method, simplification of incompletely specified functions, simplification using map-entered variables

Text book 1: Part B: Chapter 5 (Sections 5.1 to 5.4) Chapter 6(Sections 6.1 to 6.5)

RBT: L1, L2

Co-3

Co Po Maping

#	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO3												

Co-Pso's

#	PSO1	PSO2	PSO3
CO3			

MODULE 2

Introduction

Analog electronics is an electronics system were signal change continuously. Analog signal is a signal whose amplitude can take any value between given limits. A continuous signal. An analog circuit operates on continuous signals.

Digital electronics is a field of electronics involving the study of digital signals and the engineering of devices that use or produce them. **Digital signal** is a signal whose amplitude can have only given discrete values between defined limits. A signal that changes amplitude in discrete steps. A **digital circuits** operates on discrete signals.

Clock is a periodic, rectangular waveform used as a basic timing signal. **Duty cycle** for a periodic digital signal, the ratio of high level time to the period or the ratio of low level time to the period. A table that shows all of the input output possibilities of a logic circuit is called **truth table**.
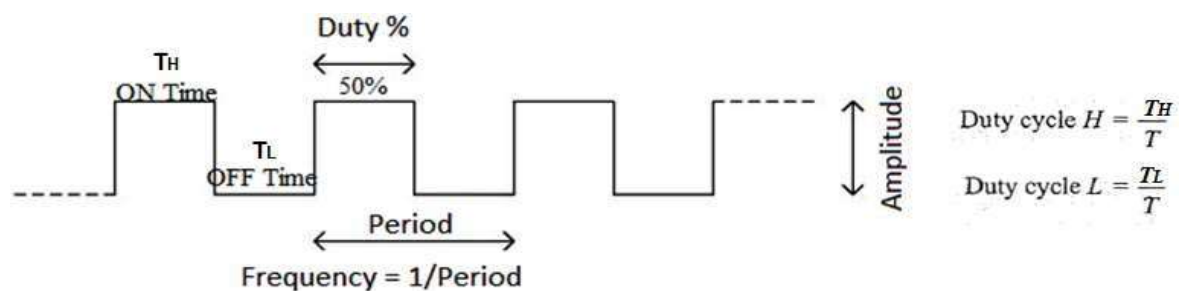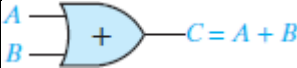

Fig: Clock Signal

A **digital circuit** having one or more input signals but only one output signal is called a gate. **Logic circuit** is a digital circuit, a switching circuit, or any kind of two-state circuit that duplicates mental processes.
The most **basic gates** are -the **NOT** gate (inverter), the **OR** gate and the **AND** gate.

NOT gate: A gate with only one input and a complemented output. $F = \overline{A}$ or $F = A'$	Truth table A | F 0 | 1 1 | 0	Logic diagram A ▷o F
OR gate: A gate with two or more inputs. The output is high when any input is high.	A B | C = A + B 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1	A B —[+]— C = A + B

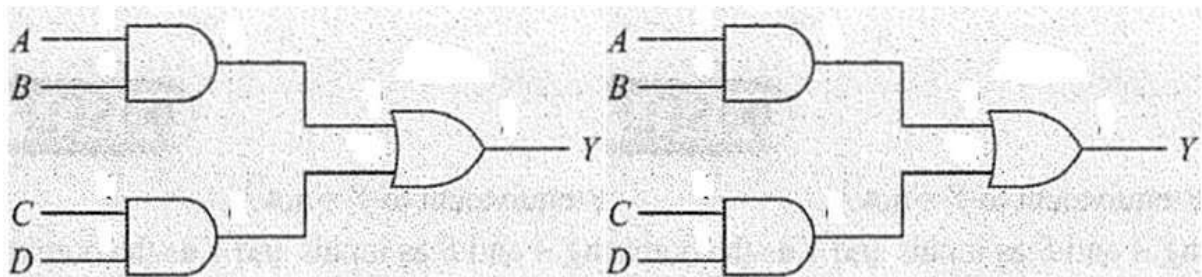|		
AND gate: A gate with 2 or more inputs.The output is high only when all inputs are high.	$A\ B$ | $C = A \cdot B$ $0\ 0$ | $\quad 0$ $0\ 1$ | $\quad 0$ $1\ 0$ | $\quad 0$ $1\ 1$ | $\quad 1$	A — B — $\boxed{\cdot}$ — $C = A \cdot B$
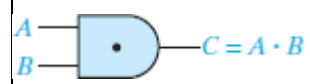
Realize following Boolean functions using basic gate.

i) Y=AB+CD   *ii) Y= (A+B)(C+D)*

Any logic function/ logic circuit can be implemented using only one kind of gate then such gates are called

universal logic gates. **NOR** gate and **NAND** gate are called universal logic gates.
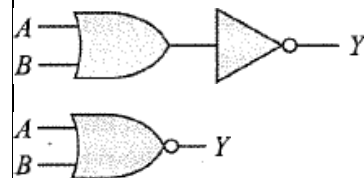
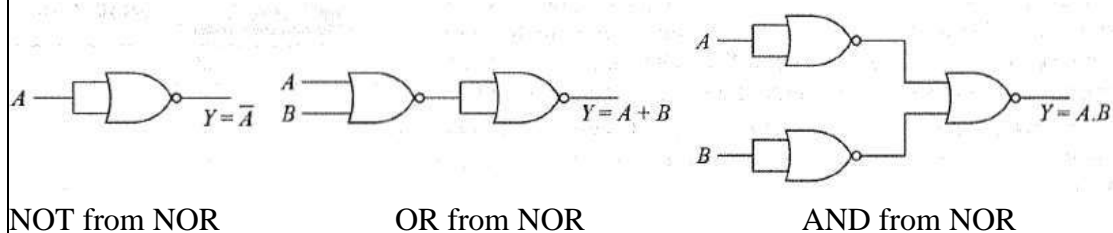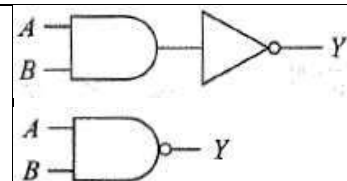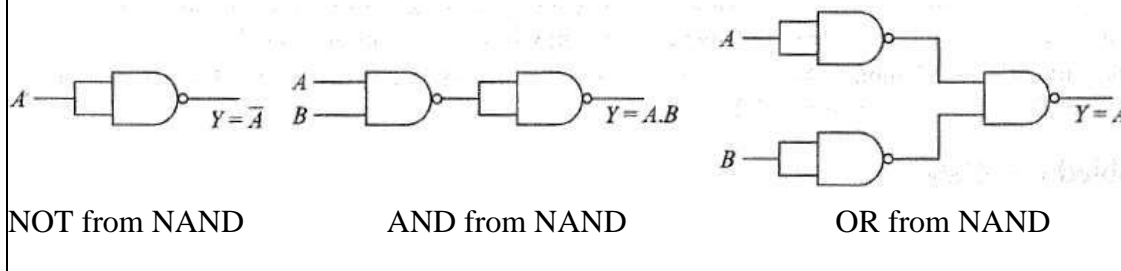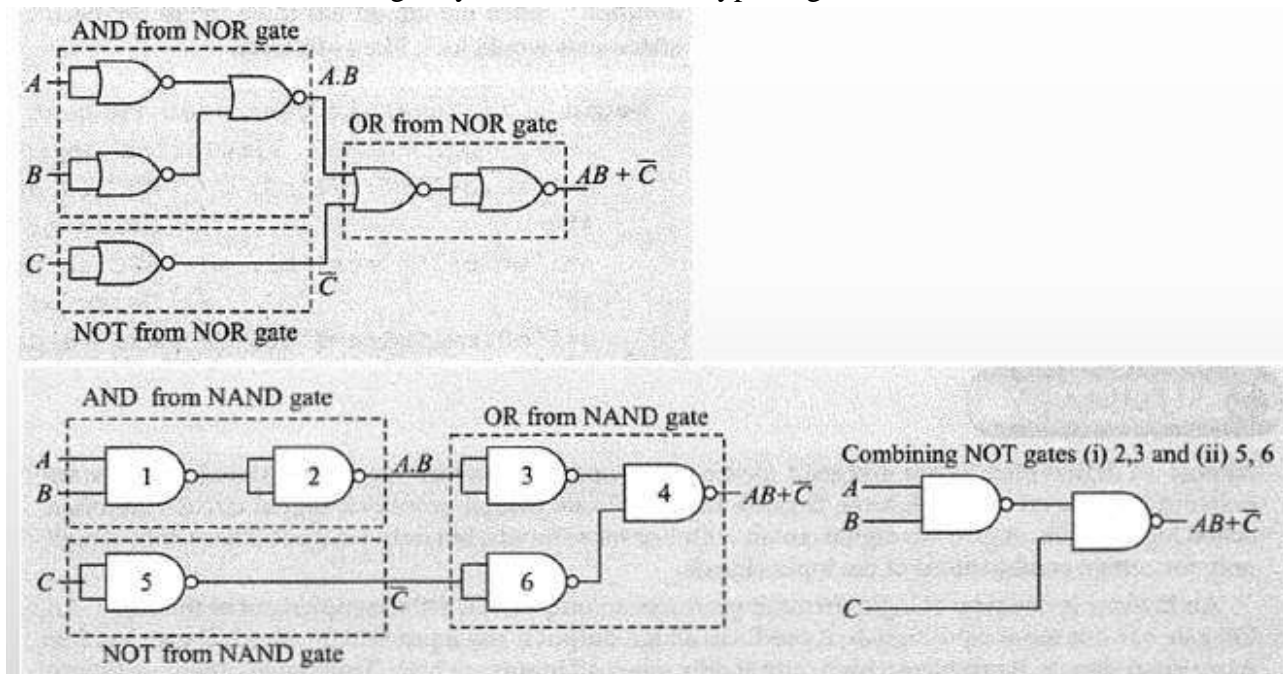|		
NOR gate: A gate with two or more inputs.The output is low when any input is high. $Y = \overline{A + B}$	Truth table A B Q 0 0 1 0 1 0 1 0 0 1 1 0	Logic diagram
Universality of NOR gate: NOT from NOR OR from NOR AND from NOR		
NAND gate: A gate with two or more inputs.The output is low when all input is high. $Y = \overline{AB}$	A B Q 0 0 1 0 1 1 1 0 1 1 1 0	

Universality of NAND gate:

NOT from NAND   AND from NAND   OR from NAND

Other important gates are XOR and XNOR gates.

Exclusive-OR gate: A gate with two or more inputs and output of is HIGH only when the number of HIGH inputs is odd.	X Y	$X \oplus Y$	
	0 0	0	
	0 1	1	
	1 0	1	
	1 1	0	
Equivalence/exclusive-NOR gate: A gate with two or more inputs and output of is HIGHonly when the number of HIGH inputs is even.	X Y	$X \equiv Y$	
	0 0	1	
	0 1	0	
	1 0	0	
	1 1	1	

Realize $Y = AB + C$ using only NOR and NAND type of gate.

Active-low refers to the concept in which a signal must be low to cause something to happen or to indicate that something has happened. **Assert** means to activate. If an input line has a bubble on it, you assert the input by making it low. If there is no bubble, you assert the input by making it high.

Laws of Boolean algebra

Operations with 0 and 1:
1. $X + 0 = X$ 1D. $X \cdot 1 = X$
2. $X + 1 = 1$ 2D. $X \cdot 0 = 0$

Idempotent laws:
3. $X + X = X$ 3D. $X \cdot X = X$

Involution law:
4. $(X')' = X$

Laws of complementarity:
5. $X + X' = 1$ 5D. $X \cdot X' = 0$

Commutative laws:
6. $X + Y = Y + X$ 6D. $XY = YX$

Associative laws:
7. $(X + Y) + Z = X + (Y + Z)$ 7D. $(XY)Z = X(YZ) = XYZ$
$= X + Y + Z$

Distributive laws:
8. $X(Y + Z) = XY + XZ$ 8D. $X + YZ = (X + Y)(X + Z)$

DeMorgan's laws:
9. $(X + Y)' = X'Y'$ 9D. $(XY)' = X' + Y'$

Minimum Forms of Switching Functions

In a **positive logic** system, binary 0 stands for low voltage and binary 1 for high voltage. In a **negative logic** system, binary 0 stands for high voltage and binary 1 for low voltage. **Assert** means to activate. If an input line has a bubble on it, you assert the input by making it low. If there is no bubble, you assert the input by making it high. **Active-low** refers to the concept in which a signal must be low to cause something to happen or activate the circuit. **Active-high** refers to the concept in which a signal must be high to cause something to happen or to activate the circuit.

Combinational Circuits are circuits made up of different types of logic gates. The output of the combinational circuit depends on the values at the input at any given time. The circuits do not make use of any memory or storage device. A **literal** is a variable or its complement.

A **minterm** of n variables is a product of n literals in which each variable appears exactly once in either true or complemented form, but not both. A **maxterm** of n variables is a sum of n literals in which each variable appearsexactly once in either true or complemented form, but not both.

Example: Minterms and Maxterms for Three Variables

Row No.	A B C	Minterms	Maxterms
0	0 0 0	$A'B'C' = m_0$	$A + B + C = M_0$
1	0 0 1	$A'B'C = m_1$	$A + B + C' = M_1$
2	0 1 0	$A'BC' = m_2$	$A + B' + C = M_2$
3	0 1 1	$A'BC = m_3$	$A + B' + C' = M_3$
4	1 0 0	$AB'C' = m_4$	$A' + B + C = M_4$
5	1 0 1	$AB'C = m_5$	$A' + B + C' = M_5$
6	1 1 0	$ABC' = m_6$	$A' + B' + C = M_6$
7	1 1 1	$ABC = m_7$	$A' + B' + C' = M_7$

Minterm expansion or a standard sum of products (SOP): A Boolean equation that is the logical sum of logical products. This type of equation applies to an AND-OR circuit.

Maxterm expansion or standard product of sums (POS): A Boolean equation that is the logical product of logical sums. This type of equation applies to an OR-AND circuit.

Steps to get SOP:
1) Locate each output 1 in truth table
2) Write the respective minterm
3) Apply OR operation to the minterms

Example:

SOP:

$$\overline{A}.\overline{B}.C + \overline{A}.B.\overline{C} + \overline{A}.B.C + A.B.\overline{C} \qquad \Sigma m(1, 2, 3, 6)$$

Steps to get POS:
1) Locate each output 0 in truth table
2) Write the respective maxterm
3) Apply AND operation to the maxterms

Example:

POS:

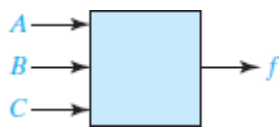$$(A+B+C).(\overline{A}+B+C).(\overline{A}+B+\overline{C}).(\overline{A}+\overline{B}+\overline{C}) = \Pi M(0, 4, 5, 7)$$

Conversion between SOP and POS:
1) Identifying complementary locations,
2) Changing mintenn to maxtenn or reverse, and finally
3) Changing summation by product or reverse

Example: $Y = F(A, B, C) = \Pi M(O, 3, 6) = \Sigma m(l, 2, 4, 5, 7)$
Example: $Y = F(A, B, C) = \Sigma m(3, 5, 6, 7) = \Pi M(O, 1, 2, 4)$

Example: Combinational Circuit with Truth Table is given write SOP and POS expressions.

A B C	f	f'
0 0 0	0	1
0 0 1	0	1
0 1 0	0	1
0 1 1	1	0
1 0 0	1	0
1 0 1	1	0
1 1 0	1	0
1 1 1	1	0

Minterms: A'BC, AB'C', AB' C, ABC', ABC

SOP:

$f = A'BC + AB'C' + AB' C + ABC' + ABC$

$f(A, B, C) = m_3 + m_4 + m_5 + m_6 + m_7$

$f(A, B, C) = \Sigma m(3, 4, 5, 6, 7)$

Maxterms: $(A + B + C), (A + B + C'), (A + B' + C)$

POS:

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

$f(A, B, C) = M0 M1 M2$

$f(A, B, C) = \Pi M(0, 1, 2)$

An **Incompletely specified function** is a Boolean function that only define output values for a subset of its inputs
- i.e. a Boolean function who's output is a don't care for at least one of its input combinations. The **X**'s in thetruth table indicate that we don't care whether the value of 0 or 1 is assigned to F.

Example: Truth Table with Don't-Cares

A B C	F
0 0 0	1
0 0 1	X
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	0
1 1 0	X
1 1 1	1

In SOP we use m to denote the required minterms and d to denote the
don't-care minterms. $F(A,B,C) = \Sigma m(0, 3, 7) + \Sigma d(1, 6)$
In POS we use M to denote the required maxterms and D to denote the
don't-care maxterms. $F(A,B,C) = \Pi M(2, 4, 5) \cdot \Pi D(1, 6)$

Write minterm and maxterm expansions for the following truth table.

A	B	C	D	Z
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	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Answer:

$Z = \Sigma\, m(0, 3, 6, 9) + \Sigma\, d(10, 11, 12, 13, 14, 15)$

$Z = \Pi\, m(1, 2, 4, 5, 7, 8) + \Sigma\, D(10, 11, 12, 13, 14, 15)$

Find the minterm expansion of $f(a, b, c, d) = a'(b' + d) + acd'$.

$$f = a'b' + a'd + acd'$$
$$= a'b'(c + c')(d + d') + a'd(b + b')(c + c') + acd'(b + b')$$
$$= a'b'c'd' + a'b'c'd + a'b'cd' + a'b'cd + \cancel{a'b'c'd} + \cancel{a'b'cd}$$
$$+ a'bc'd + a'bcd + abcd' + ab'cd'$$

Duplicate terms have been crossed out, because $X + X = X$. This expression can then be converted to decimal notation:

$$f = a'b'c'd' + a'b'c'd + a'b'cd' + a'b'cd + a'bc'd + a'bcd + abcd' + ab'cd'$$
$$\ \ \ \ 0\,0\,0\,0 \quad 0\,0\,0\,1 \quad 0\,0\,1\,0 \quad 0\,0\,1\,1 \quad 0\,1\,0\,1 \quad 0\,1\,1\,1 \quad 1\,1\,1\,0 \quad 1\,0\,1\,0$$
$$f = \Sigma\, m(0, 1, 2, 3, 5, 7, 10, 14)$$

The maxterm expansion for f can then be obtained by listing the decimal integers (in the range 0 to 15) which do not correspond to minterms of f:

$$f = \Pi\, M(4, 6, 8, 9, 11, 12, 13, 15)$$

Show that $a'c + b'c + ab = a'b' + bc + ac'$.
 We will find the minterm expansion of each side by supplying the missing variables. For the left side,

$$a'c(b + b') + b'c'(a + a') + ab(c + c')$$
$$= a'bc + a'b'c + ab'c' + a'b'c' + abc + abc'$$
$$= m_3 + m_1 + m_4 + m_0 + m_7 + m_6$$

For the right side,

$$a'b'(c + c') + bc(a + a') + ac'(b + b')$$
$$= a'b'c + a'b'c + abc + a'bc + abc' + ab'c'$$
$$= m_1 + m_0 + m_7 + m_3 + m_6 + m_4$$

Because the two minterm expansions are the same, the equation is valid.

2.1 Minimum Forms of Switching Functions

A *minimum sum of products expression* for a function is defined as a sum of product terms which has a minimum number of terms and of all those expressions which have the same minimum number of terms, has a minimum number of literals. The minimum sum of products corresponds directly

to a minimum two-level gate circuit which has a minimum number of gates and a minimum number of gate inputs. The minimum sum of products is not necessarily unique; that is, a given function may have two differentminimum sum of products forms, each with the same number of terms and the same number of literals. Given a minterm expansion, the minimum sum-of products form can often be obtained by the followingprocedure:

(i) Combine terms by using $XY' + XY = (Y'+Y)=X$. Do this repeatedly to eliminate as many literals as possible.A given term may be used more than once because X+X=X.

(ii) Eliminate redundant terms by using the theorems of Boolean Algebra.

Example: Find a minimum sum-of-products expression for

$$F(a, b, c) = \Sigma m (0, 1, 2, 5, 6, 7)$$

$$F = a'b'c' + a'b'c + a'bc' + ab'c + abc' + abc$$

$$F = a'b'(c'+c) + bc'(a'+ a) + ac(b +b')$$

$$F = a'b' + bc' + ac$$

Or

$$F = a'b'c' + a'b'c + a'bc' + ab'c + abc' + abc$$

$$F = a'b'c' + a'bc' + a'b'c + ab'c + abc' + abc$$

$$F = a'c' + b'c + ab$$

A *minimum product of sums expression* for a function is defined as a product of sum terms which has a minimum number of terms, and of all those expressions which have the same number of terms, has a minimum number of literals. Unlike the maxterm expansion, the minimum product of sums form of a function is not necessarily unique. Givena maxterm expansion, the minimum product of sums can often be obtained by a procedure similar to that used inthe minimum sum of products case, except that the theorem $(X+Y')(X+Y)= X$ is used to combine terms.

Example: $F(a, b, c,d) = \Pi M (5, 7, 6, 14, 2, 10)$

$$F = (a+b'+c +d')(a+b'+c'+d')(a+b'+c'+d)(a'+b'+c'+d)(a+b+c'+d)(a'+b+c'+d)$$

$$F = (a+b'+d')(b'+c'+d)(b+c'+d)$$

$$F = (a+b'+d')(c'+d)$$

Simplification of Boolean function reduces the gate count required to implement the circuit, the circuit works faster and circuit require less power consumption.

The various Boolean expression simplification techniques are

1) Algebraic techniques
2) Karnaugh Map/K-Map Method
3) Quine McCluskey Method
4) Entered Variable Map/ MEV/EMV Method Switching/Boolean functions can generally be simplified by using the algebraic techniques.

The disadvantagesof algebraic procedure usage are

(i)   The procedures are difficult to apply in a systematic way,

(ii)   It is difficult to tell when we have arrived at a minimum solution.

Karnaugh map/K map is a method simplifying and manipulating switching functions. K map method is fasterand easier to apply than other simplification methods.

2.2 Two and Three Variable Karnaugh Maps

Karnaugh map of a function specifies the value of the function for every combination of values of the independentvariables.

Two Variable K-Map

The number of cells in 2 variable K-map is four (2^2), since the number of variables is two. The following figureshows 2 variable K-map and location of minterms on a 2 variable K-map.
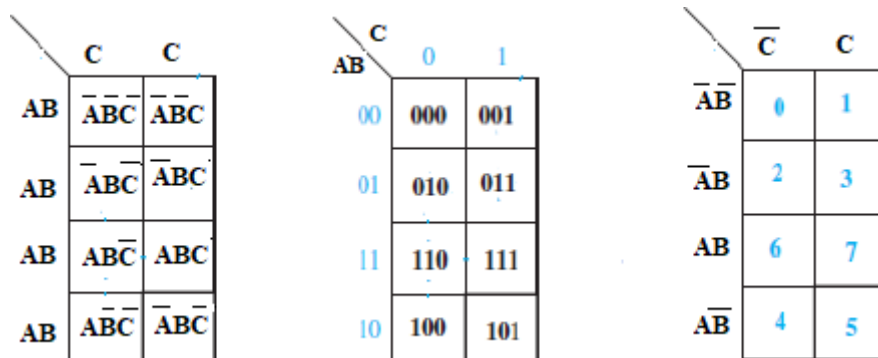

Example: Convert following
truth table into K map.Y = F(A,
B) = Σ m (2, 3)

A	B	Y
0	0	0
0	1	0
1	0	1
1	1	1
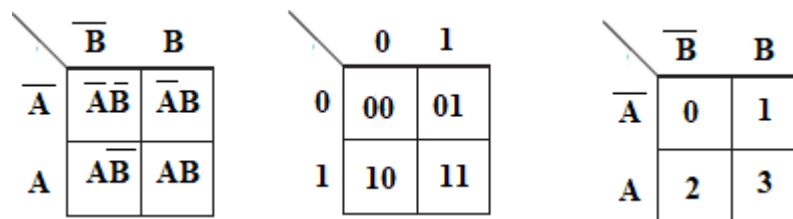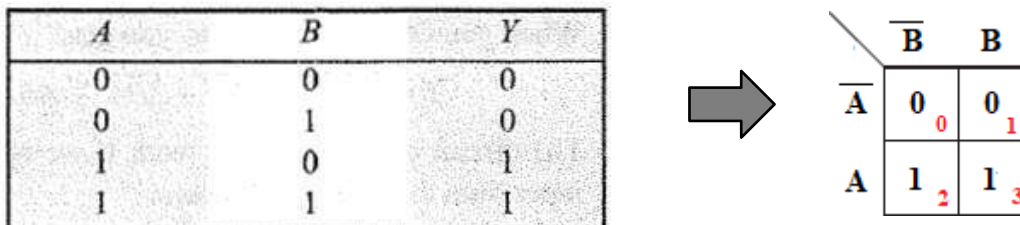
Three Variable K-Map

The number of cells in 3 variable K-map is eight (2^3), since the number of variables is 3. The following figureshows 3 variable K-map and location of minterms on a 3 variable K-map.

OR

	$\overline{B}\,\overline{C}$	$\overline{B}\,C$	$B\,C$	$B\,\overline{C}$
$\overline{A}$	$\overline{A}\,\overline{B}\,\overline{C}$	$\overline{A}\,\overline{B}\,C$	$\overline{A}\,B\,C$	$\overline{A}\,B\,\overline{C}$
A	$A\,\overline{B}\,\overline{C}$	$A\,\overline{B}\,C$	$A\,B\,C$	$A\,B\,\overline{C}$

BC / A	00	01	11	10
0	000	001	011	010
1	100	101	111	110

	$\overline{B}\,\overline{C}$	$\overline{B}\,C$	$B\,C$	$B\,\overline{C}$
$\overline{A}$	0	1	3	2
A	4	5	7	6

Example: Convert following truth table into K map. $Y = F(A, B, C) = \Sigma\, m\,(2,6,7)$

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

	$\overline{B}\,\overline{C}$	$\overline{B}\,C$	$B\,C$	$B\,\overline{C}$
$\overline{A}$	0	0	0	1
A	0	0	1	1

2.3 **Four-Variable Karnaugh Maps**

The number of cells in 4 variable K-map is sixteen (2^4), since the number of variables is 4. The following figureshows 4 variable K-map and location of minterms on a 4 variable K-map.

	$\overline{C}\,\overline{D}$	$\overline{C}\,D$	$C\,D$	$C\,\overline{D}$
$\overline{A}\,\overline{B}$	$\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$	$\overline{A}\,\overline{B}\,\overline{C}\,D$	$\overline{A}\,\overline{B}\,C\,D$	$\overline{A}\,\overline{B}\,C\,\overline{D}$
$\overline{A}\,B$	$\overline{A}\,B\,\overline{C}\,\overline{D}$	$\overline{A}\,B\,\overline{C}\,D$	$\overline{A}\,B\,C\,D$	$\overline{A}\,B\,C\,\overline{D}$
$A\,B$	$A\,B\,\overline{C}\,\overline{D}$	$A\,B\,\overline{C}\,D$	$A\,B\,C\,D$	$A\,B\,C\,\overline{D}$
$A\,\overline{B}$	$A\,\overline{B}\,\overline{C}\,\overline{D}$	$A\,\overline{B}\,\overline{C}\,D$	$A\,\overline{B}\,C\,D$	$A\,\overline{B}\,C\,\overline{D}$

	$\overline{C}\,\overline{D}$	$\overline{C}\,D$	$C\,D$	$C\,\overline{D}$
$\overline{A}\,\overline{B}$	0	1	3	2
$\overline{A}\,B$	4	5	7	6
$A\,B$	12	13	15	14
$A\,\overline{B}$	8	9	11	10

	00	01	11	10
00	0000	0001	0011	0010
01	0100	0101	0111	0110
11	1011	1101	1111	1110
10	1000	1001	1011	1010

Example: Convert following truth table into K map. $Y = F(A, B, C, D) = \Sigma\, m(1,6,7)$

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Pairs, Quads, and Octets

Two adjacent 1s in the K-map is called a *pair* and it eliminate the *variable* that changes form. Sample of pair

A *quad* is a group of four ls that are horizontally or vertically adjacent and a quad eliminates two variables andtheir complements.
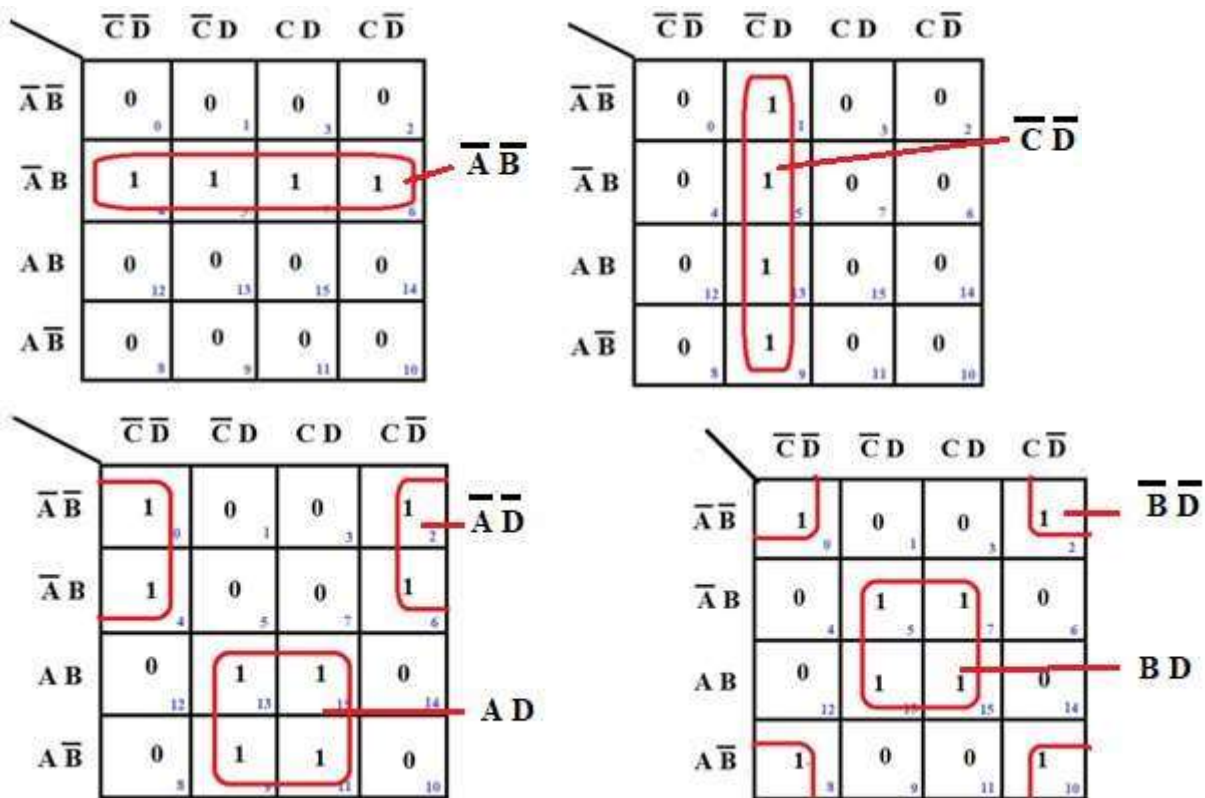
An *octet* is a group of 8 ls that are horizontally or vertically adjacent and an octet eliminates three variables andtheir complements.
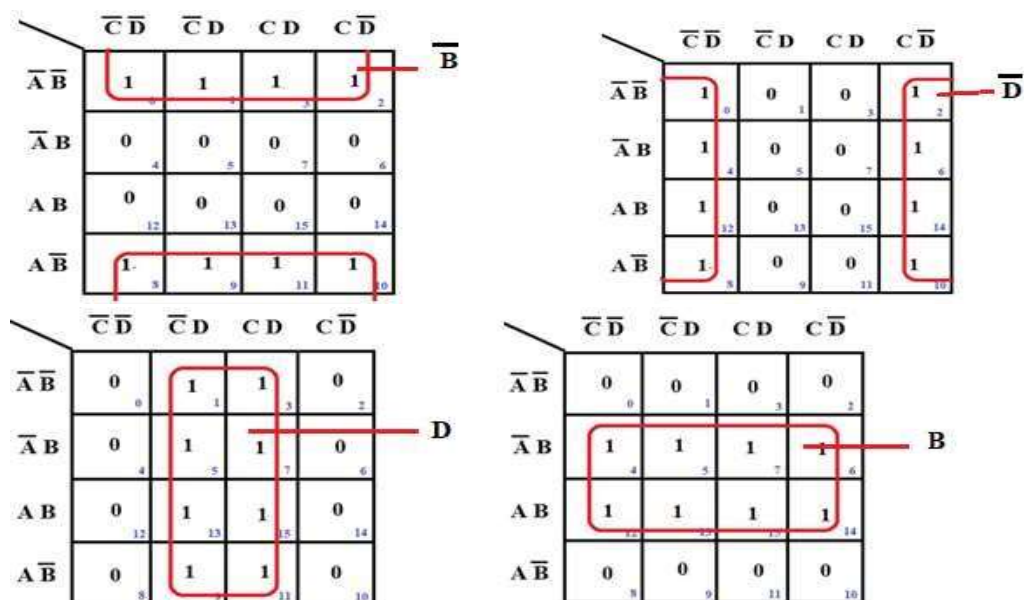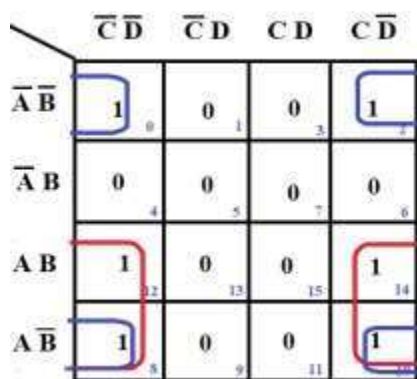
Overlapping of groups: We are allowed to use the same 1 more than once.
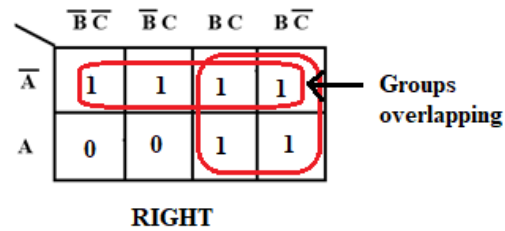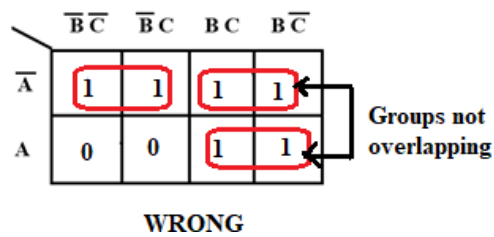
Rolling of Map: Groups may wrap around the table. The leftmost cell in a row may be grouped with the rightmostcell and the top cell in a column may be grouped with the bottom cell. Roll and overlap to get largest group.

Eliminating redundant group: A groups of 1s or 0s whose all members are overlapped by other groups is calledredundant group. After encircling all possible group, eliminate any redundant group if any. We don't consider this group while writing the simplified equations from the K-map.
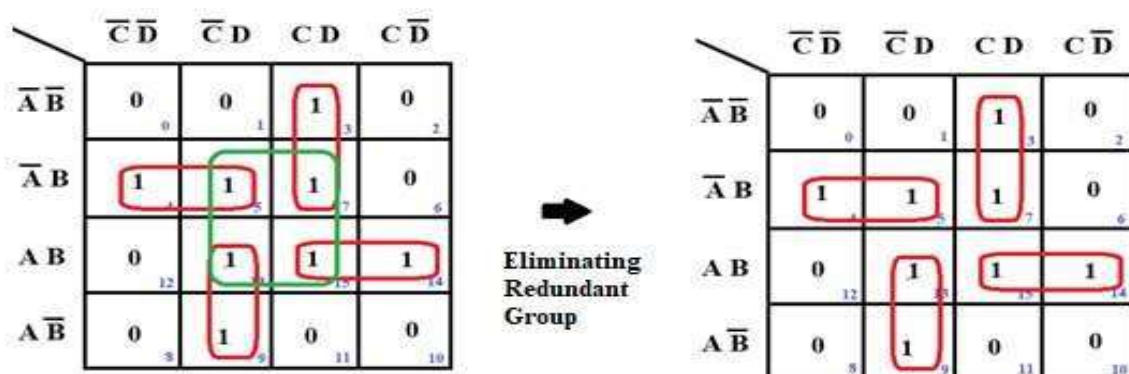
2.4 <u>Determination of Minimum Expressions using Essential Prime Implicants</u>

 Any single 1 or any group of 1's which can be combined together on a map of the function F represents a productterm which is called an implicant of F. Several implicants of F may be possible. A product term implicant is called a prime implicant if it cannot be combined with another term to eliminate a variable.

The following procedure can then be used to obtain a minimum sum of products from a Karnaugh map:

1) Choose a minterm (a 1) which has not yet been covered.
2) Find all 1's and X's adjacent to that minterm. (Check the n adjacent squares on an n-variable map.)
3) If a single term covers the minterm and all of the adjacent 1's and X's, then that term is an essential primeimplicant, so select that term. (don't-care terms are treated like 1's in steps 2 and 3 but not in step 1.)
4) Repeat steps 1, 2, and 3 until all essential prime implicants have been chosen.
5) Find a minimum set of prime implicants which cover the remaining 1's on the map. (If there is more thanone such set, choose a set with a minimum number of literals.)

The following figure shows the flowchart for determining a minimum sum of products using a Karnaugh map with an example.
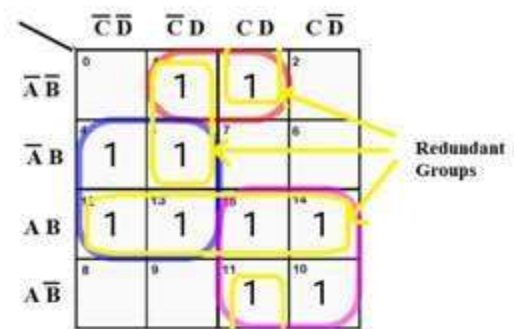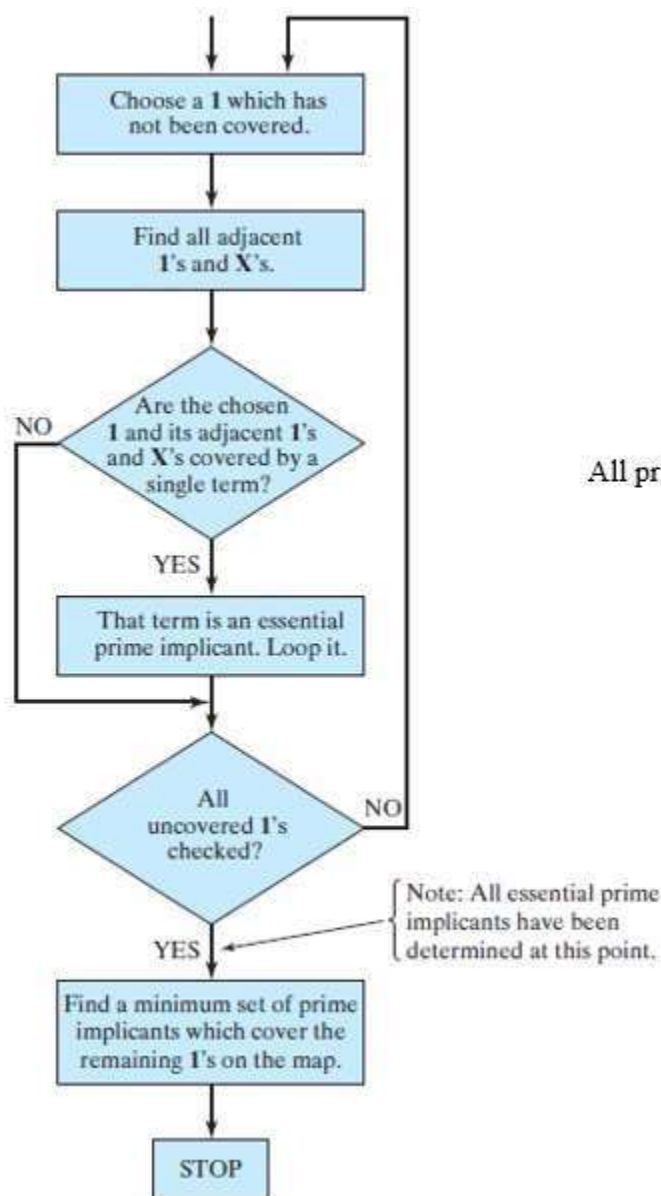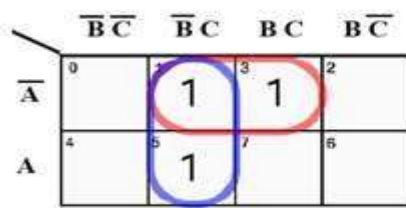

All prime implicants: $A'B'D, BC', AC, A'C'D, AB, B'CD$
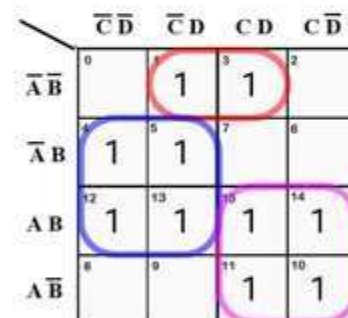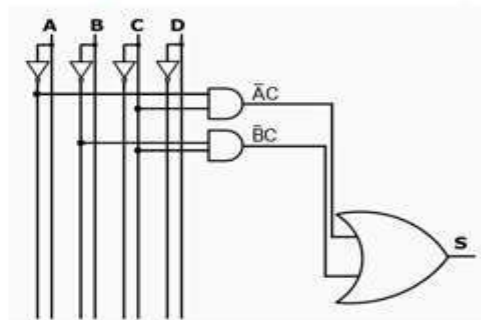
Minimum solution: $F = A'B'D + BC' + AC$

Solve S(A,B,C)=Σm(1,3,5) using K map and implement using basic gates.

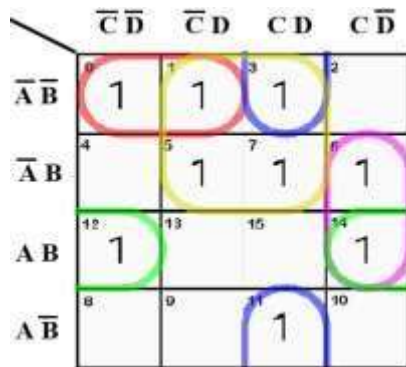$$S = \bar{A}C + \bar{B}C$$

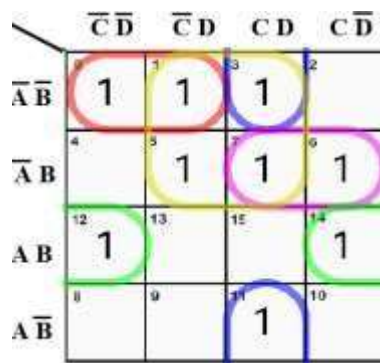Solve S= F(A,B,C)=Σ m(0, 1, 3, 5, 6, 7, 11, 12, 14) using Kmap and implement uisng basic, nand only and nor only gates.
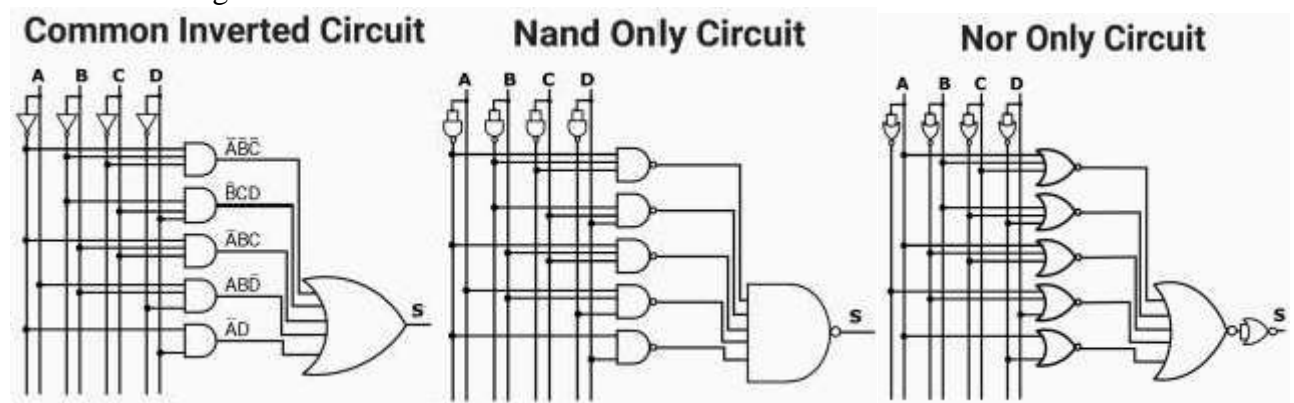

OR

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

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

Circuit diagram for $S = \bar{A}B\bar{C} + \bar{B}CD + \bar{A}BC + AB\bar{D} + \bar{A}D$

Solve S=F(A,B,C,D)=Σ m(0,1, 2, 4, 5,6, 8,9,10,12,13) using Kmap and implement uisng basic, nand only and norlygates.

$$S = \bar{A}\bar{D} + \bar{B}\bar{D} + \bar{C}$$
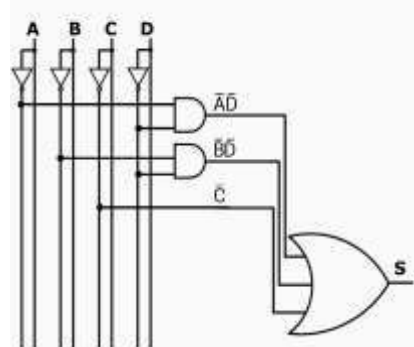
Common Inverted Circuit   **Nand Only Circuit**   **Nor Only Circuit**
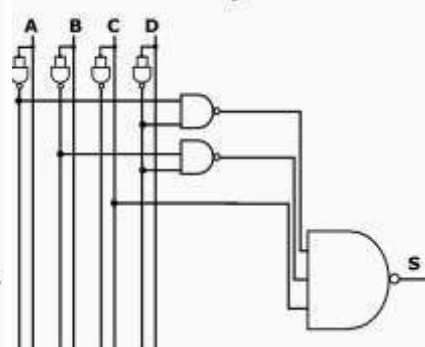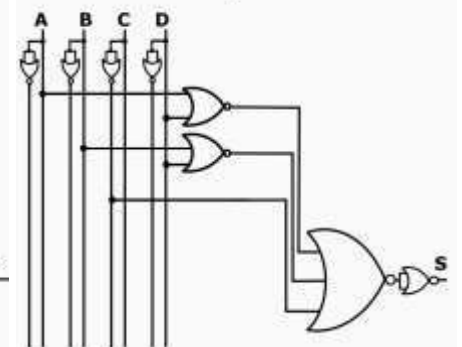
Solve S= F(A,B,C,D)=Σm(7,9,10,11,12,13,14,15) using K map to get minimum SOP expression.

$$S = BCD + AD + AC + AB$$

Solve S=F(A,B,C,D)=Σm(1,2,3,6,8,9,10,12,13,14) using K map to get minimum SOP

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

Solve S=F(A,B,C,D)=Σm(7)+d(10,11,12,13,14,15) using K map to get minimum SOP

S = BCD

Solve S =F(A,B,C,D)=Σm(2,3,5,7,10,12)+d(11,15) using K map to get minimum SOP expression.

$S = AB\bar{C}\bar{D} + \bar{A}BD + \bar{B}C$

Solve S=F(A,B,C,D)=Σm(6,7,9,10,13)+d(1,4,5,11) using K map to get minimum SOP expression.

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

Solve S= F(A,B,C,D)=Σm(0,1,2,4,5,12,14)+d(8,10) using K map to get minimum SOP expression.

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

Solve S= F(A,B,C,D)=Σm(0,1,4,8,9,10)+d(2,11) using K map to get minimum SOP expression.

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

$$S = \bar{A}\bar{C}\bar{D} + \bar{B}\bar{D} + \bar{B}\bar{C}$$
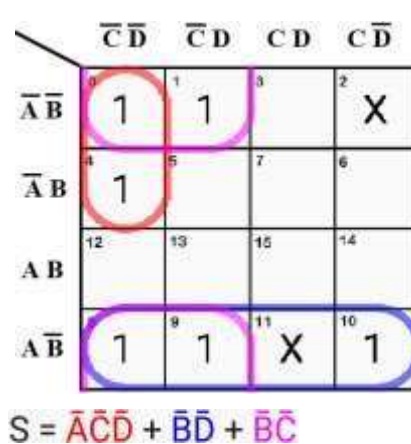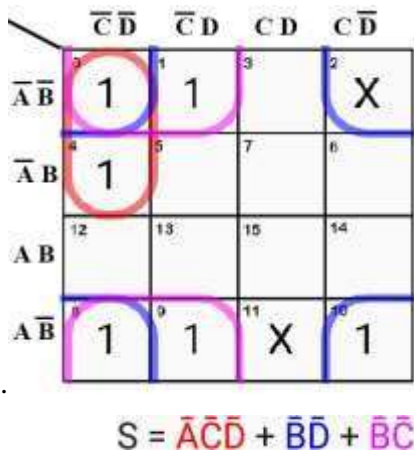
Solve S=F(A,B,C,D)=ΠM(0,1,3,4,7) using K map to get minimum POS expression.

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

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

Solve S=F(A,B,C,D)=ΠM(0,6,7,8,12,13,14,15) using K map to get minimum POS expression and implement uisngbasic, nand only and norly gates.

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

Common Inverted Circuit

Nor Only Circuit

Nand Only Circuit

Solve F(A,B,C,D)=ΠM(0,3,4,7,810,12,14).ΠD(2,6) using K map to get minimum POS expression and implementuisng basic, nand only and norly gates.

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

Solve S=F(A,B,C,D)=Σm(6,7,9,10,13)+ Σd(1,4,5,11) using K map to get minimum

POS expression.S= F(A,B,C,D)=ΠM(0,2,3,8,12,14,15).ΠD(1,4,5,11)

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

Limitations of K map:
Complexity of **K**-**map** simplification process increases with the increase in the number of variables
K map is manual technique and simplification process heavily depends on the human ability.

2.5 <u>Quine-McCluskey Method</u>

The Quine-McCluskey method provides a systematic simplification procedure which can be readily programmedfor a digital computer. The Quine-McCluskey method reduces the minterm expansion (standard sum-of-productsform) of a function to obtain a minimum sum of products.

Quine-McClusky method involves preparation of two tables; one determines prime implicants and the other selects essential prime implicants to get minimal expression. Prime implicants are expressions with least numberof literals that represents all the terms given in a truth table. Prime implicants are examined to get essential primeimplicants for a particular expression that avoids any type of duplication.

2.6 Determination of Prime Implicants

In order to apply the Quine-McCluskey method to determine a minimum sum-of-products expression for a function, the function must be given as a sum of minterms.

Step 1 − Arrange the given min terms in an ascending order and make the groups based on the number of ones present in their binary representations. So, there will be at most 'n+1' groups if there are 'n' Boolean variables in a Boolean function or 'n' bits in the binary equivalent of min terms.

Step 2 − Compare the min terms present in successive groups. If there is a change in only one-bit position, then take the pair of those two min terms. Place this symbol '_' in the differed bit position and keep the remaining bitsas it is.

$$AB'CD' + AB'CD = AB'C$$
$$\underbrace{1\ 0\ 1}_{X}\ \underbrace{0}_{Y} + \underbrace{1\ 0\ 1}_{X}\ \underbrace{1}_{Y'} = \underbrace{1\ 0\ 1}_{X}\ \_ \ \text{(the dash indicates a missing variable)}$$

$$A'BC'D + A'BCD' \ \text{(will not combine)}$$
$$0\ 1\ 0\ 1 + 0\ 1\ 1\ 0 \ \text{(will not combine)}$$

Step 3 − Repeat step2 with newly formed terms till we get all prime implicants.

Given a function F of n variables, a product term P is an implicant of F if for every combination of values of then variables for which P = 1, F is also equal to 1.

A prime implicant of a function F is a product term implicant which is no longer an implicant if any literal is deleted from it.

2.7 The Prime Implicant Chart:

The second part of the Quine-McCluskey method employs a prime implicant chart to select a minimum set of primeimplicants.

Step 4 − Formulate the prime implicant table/chart. It consists of set of rows and columns. Prime implicants canbe placed in row wise and min terms can be placed in column wise. Place '1' in the cells corresponding to the min terms that are covered in each prime implicant.

Step 5 − Find the essential prime implicants by observing each column. If the min term is covered only by one prime implicant, then it is essential prime implicant. Those essential prime implicants will be part of the simplifiedBoolean function.

Step 6 − Reduce the prime implicant table by removing the row of each essential prime implicant and the columnscorresponding to the min terms that are covered in that essential prime implicant. Repeat step 5 for Reduced prime implicant table. Stop this process when all min terms of given Boolean function are over.

Find the minimum SOP for the function f (a, b, c, d) =∑m(0, 1, 2, 5, 6, 7, 8, 9, 10, 14) using Quine McCluskeymethod.

Determination of prime implicants:

Stage 1		Stage 2		Stage 3	
Minterms	a b c d	Minterms	a b c d	Minterms	a bcd
0	0000 ✓	0, 1	000– ✓	0, 1, 8, 9	–00–
1	0001 ✓	0, 2	00–0 ✓	0, 2, 8, 10	–0–0
2	0010 ✓	0, 8	–000 ✓	0, 8, 1, 9	–00–
8	1000 ✓	1, 5	0–01	0, 8, 2, 10	–0–0
5	0101 ✓	1, 9	–001 ✓	2, 6, 10, 14	– –10
6	0110 ✓	2, 6	0–10 ✓	2, 10, 6, 14	– –10
9	1001 ✓	2, 10	–010 ✓		
10	1010 ✓	8, 9	100– ✓		
7	0111 ✓	8, 10	10–0 ✓		
14	1110 ✓	5, 7	01–1		
		6, 7	011–		
		6, 14	–110 ✓		
		10, 14	1–10 ✓		

Prime implicant chart:

$$f = b'c' + cd' + a'bd$$

Determine   the   essential   prime implicants   using QM   methodfor   the function f(A, B, C,D)=∑M(0,1,2,3,10,12,11,13,14,15)

Determination of prime implicants:

Stage 1		Stage 2		Stage 3	
ABCD	minterms	ABCD	minterms	ABCD	minterms
0 0 0 0	(0)√	0 0 0 -	(0,1)√	0 0 - -	(0,1,2,3)
		0 0 - 0	(0,2)√	0 0 - -	(0,2,1,3)
0 0 0 1	(1)√				
0 0 1 0	(2)√	0 0 - 1	(1,3)√	- 0 1 -	(2,10,3,11)
		0 0 1 -	(2,3)√		
		- 0 1 0	(2,10)√	1 - 1 -	(10,11,14,15)
0 0 1 1	(3)√			1 - 1 -	(10,14,11,15)
1 0 1 0	(10)√	- 0 1 1	(3,11)√	1 1 - -	(12,13,14,15)
1 1 0 0	(12)√	1 0 1 -	(10,11)√	1 1 - -	(12,14,13,15)
		1 - 1 0	(10,14)√		
		1 1 0 -	(12,13)√		
1 0 1 1	(11)√	1 1 - 0	(12,14)√		
1 1 0 1	(13)√				
1 1 1 0	(14)√	1 - 1 1	(11,15)√		
		1 1 - 1	(13,15)√		
1 1 1 1	(15)√	1 1 1 -	(14,15)√		

Prime implicant charts:

	0	1	2	3	10	11	12	13	14	15
A'B' (0,1,2,3)	√	√	√	√						
B'C (2,3,10,11)			√	√	√	√				
AC (10,11,14,15)					√	√			√	√
AB (12,13,14,15)							√	√	√	√

$$Y = A'B' + B'C + AB \text{ or } Y = A'B' + AC + AB$$

Find the minimum SOP for the function f (A, B, C) =$\sum$m(2, 6, 7) using Quine

McCluskey method.Determination of prime implicants:

Stage 1		Stage 2	
A B C	minterms	A B C	minterms
0 1 0	(2)√	- 1 0	(2,6)
1 1 0	(6)√	1 1 -	(6,7)
1 1 1	(7)√		

Prime implicant charts:

Prime Implicants	2	6	7
BC' (2,6)	√	√	
AB (6,7)		√	√

$$Y = AB + BC'$$

2.8 Petrick's Method

Petrick's method is a technique for determining all minimum sum-of-products solutions from a prime implicant chart. The example discussed above has two minimum solutions. As the number of variables increases, the number of prime implicants and the complexity of the prime implicant chart may increase significantly. In such cases, a large amount of trial and error may be required to find the minimum solution(s).

Petrick's method is a more systematic way of finding all minimum solutions from a prime implicant chart than the method used previously. Before applying Petrick's method, all essential prime implicants and the minterms they cover should be removed from the chart.

Steps in Petrick's method:
1. Reduce the prime implicant chart by eliminating the essential prime implicant rows and the correspondingcolumns.
2. Label the rows of the reduced prime implicant chart P_1, P_2, P_3, etc.
3. Form a logic function P which is true when all columns are covered. P consists of a product of sum terms,each sum term having the form $(P_{i0} + P_{i1} + \ ■\ ■\ ■\ )$, where P_{i0}, P_{i1} . . . represent the rows which covercolumn i.
4. Reduce P to a minimum sum of products by multiplying out and applying $X + XY = X$.

5. Each term in the result represents a solution, that is, a set of rows which covers all of the minterms in thetable. To determine the minimum solutions, find those terms which contain a minimum number of variables. Each of these terms represents a solution with a minimum number of prime implicants.

6. For each of the terms found in step 5, count the number of literals in each prime implicant and find the total number of literals. Choose the term or terms which correspond to the minimum total number of literals, and write out the corresponding sums of prime implicants.

Find the minimum SOP form for the function f(a,b,c)=$\sum m(0,1,2,5,6,7)$ using pettricks method.

Determination of prime implicants:

Stage 1		Stage 2	
minterms	abc	minterms	abc
0	000 ✓	0,1	00–
1	001 ✓	0,2	0–0
2	010 ✓	1,5	–01
5	101 ✓	2,6	–10
6	110 ✓	5,7	1–1
7	111 ✓	6,7	11–

Prime implicant charts:

Prime Imlicants			0	1	2	5	6	7
P_1	(0, 1)	a'b'	X	X				
P_2	(0, 2)	a'c'	X		X			
P_3	(1, 5)	b'c		X		X		
P_4	(2 6)	bc'			X		X	
P_5	(5, 7)	ac				X		X
P_6	(6, 7)	ab					X	X

$P = (P_1 + P_2)(P_1 + P_3)(P_2 + P_4)(P_3 + P_5)(P_4 + P_6)(P_5 + P_6) = 1$

using $(X + Y)(X + Z) = X + YZ$ and distributive law:

$P = (P_1 + P_2 P_3)(P_4 + P_2 P_6)(P_5 + P_3 P_6)$

$= (P_1 P_4 + P_1 P_2 P_6 + P_2 P_3 P_4 + P_2 P_3 P_6)(P_5 + P_3 P_6)$

$= P_1 P_4 P_5 + P_1 P_2 P_5 P_6 + P_2 P_3 P_4 P_5 + P_2 P_3 P_5 P_6 + P_1 P_3 P_4 P_6 + P_1 P_2 P_3 P_6 + P_2 P_3 P_4 P_6 + P_2 P_3 P_6$

use $X + XY = X$ to eliminate redundant terms from P.

$P = P_1 P_4 P_5 + P_1 P_2 P_5 P_6 + P_2 P_3 P_4 P_5 + P_1 P_3 P_4 P_6 + P_2 P_3 P_6$

The two solutions with the minimum number of prime implicants are obtained by choosing rows P_1, P_4, and P_5 or rows P_2, P_3, and P_6.   F= a'b' + bc' + ac   OR   F = a'c' + b'c + ab

2.9 <u>Simplification of Incompletely Specified Functions:</u>

Given an incompletely specified function, the proper assignment of values to the don't-care terms is necessary inorder to obtain a minimum form for the function. Modified Quine-McCluskey method is used to obtain a minimum solution when don't-care terms are present. In modified Quine-McCluskey method the don't-care termsare treated like required minterms when finding the prime implicants and don't-care columns are omitted when forming the prime implicant chart. If extra prime implicants are generated because of the don't-cares, the extra prime implicants are eliminated in the prime implicant chart.

Find the minimum SOP for the function F(A, B, C, D) = Σ m(2, 3, 7, 9, 11, 13) + Σ d(1, 10, 15) using Quine McCluskey method.

Determination of prime implicants:

Stage 1		Stage 2		Stage 3	
minterms	ABCD	minterms	ABCD	minterms	ABCD
1	0001 ✓	(1, 3)	00–1 ✓	(1, 3, 9, 11)	–0–1
2	0010 ✓	(1, 9)	–001 ✓	(2, 3, 10,11)	–01–
3	0011 ✓	(2, 3)	001– ✓	(3, 7, 11, 15)	--11
9	1001 ✓	(2, 10)	–010 ✓	(9, 11, 13, 15)	1--1
10	1010 ✓	(3, 7)	0–11 ✓		
7	0111 ✓	(3, 11)	–011 ✓		
11	1011 ✓	(9, 11)	10–1 ✓		
13	1101 ✓	(9, 13)	1–01 ✓		
5	1111 ✓	(10, 11)	101– ✓		
		(7, 15)	–111 ✓		
		(11, 15)	1–11 ✓		
		(13, 15)	11–1 ✓		

Prime implicant chart:

(The don't-care columns are omitted when forming the prime implicant chart)

Prime Implicants	2	3	7	9	11	13
(1, 3, 9, 11)		X̶		X̶	X̶	
(2, 3, 10, 11)	X	X			X	
(3, 7, 11, 15)		X	X		X	
(9, 11, 13, 15)				X	X	X

$$F = B'C + CD + AD$$

2.10 Simplification Using Map Entered Variables:

A technique called Map Entered Variable (MEV) or Entered Variable Map (EVM) is used to increase the effectivesize of k-map. It allows a smaller map to handle large number of variables. This is done by writing output in terms of input. It allows a smaller map to handle large number of variables. This is done by writing output in terms of input.

Rules for entering values in a Map Entered Variable K map are:

Rule No.	MEV	f	Map Entry	Comments
1	0 1	0 0	0	If function equals 0 for both values of MEV, enter 0 in appropriate cell of MEV Map.
2	0 1	1 1	1	If function equals 1 for both values of MEV, enter 1.
3	0 1	0 1	MEV	If function equals MEV, enter MEV
4	0 1	1 0	$\overline{M}$	If the function is compliment of MEV, enter $\overline{M}$
5	X X		X	If function equals X, enter X
6	0 1	0 X	0	f=0 for MEV=0 and f=X for MEV=1,enter 0
7	0 1	X 0	0	f=X for MEV=0 and f=0 for MEV=1,enter 0
8	0 1	1 X	1	f=1 for MEV=0 and f=X for MEV=1,enter 1
9	0 1	X 1	1	f=X for MEV=0 and f=1 for MEV=1,enter 1

Minimization procedure for MEV/EVM:

1) Write all the variables (original and complimented forms are treated as two different variables) in the map as0, leave 0's, minterms and don't cares as it is and obtain the SOP expression.
2) Select one variable and make all occurrences of that variable as 1, write minterms (1's) as don't cares, leave0's and don't cares as it is. Now, obtain the SOP expression by multiply the obtained SOP expression with the concerned variable.
3) Repeat step 2 for all the variables in the k-map.
4) SOP of MEV/EVM is obtained by ORing all the obtained SOP expressions.

Example – A 3-variable function can be defined as a function of 2-variables if the output is written in terms ofthird variable.
Consider a function F(A,B,C) = (0,1,2,5)

Step 1:

	$\overline{B}$	B
$\overline{A}$	1	0
A	0	0

SOP1 = $\overline{A}\,\overline{B}$

Step 2:

	$\overline{B}$	B
$\overline{A}$	X	1
A	0	0

SOP2 = $\overline{A}\,C$

Step 3:

	$\overline{B}$	B
$\overline{A}$	X	0
A	1	0

SOP3 = $\overline{B}\,C$

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

Solve S=F(A,B,C,D)=Σm(2,3,6,7,9,10,13,14) using MEV to get minimum SOP expression.

A	B	C	D	Y	MEV
0	0	0	0	0	0
0	0	0	1	0	
0	0	1	0	1	1
0	0	1	1	1	
0	1	0	0	0	0
0	1	0	1	0	
0	1	1	0	1	1
0	1	1	1	1	
1	0	0	0	0	D
1	0	0	1	1	
1	0	1	0	1	$\overline{D}$
1	0	1	1	0	
1	1	0	0	0	D
1	1	0	1	1	
1	1	1	0	1	$\overline{D}$
1	1	1	1	0	

SOP Y = $\overline{A}\,C + A\,\overline{C}\,D + C\,\overline{D}$

	$\overline{B}\,\overline{C}$	$\overline{B}\,C$	B C	$B\,\overline{C}$
$\overline{A}$	0	1	1	0
A	D	$\overline{D}$	$\overline{D}$	D

(cells numbered 0,1,3,2 / 4,5,7,6)

1) Write all the variables as 0, leave minterms, 0's and don't cares as it is and obtain the SOP expression.

	$\overline{B}\,\overline{C}$	$\overline{B}\,C$	B C	$B\,\overline{C}$
$\overline{A}$	0	1	1	0
A	0	0	0	0

SOP1: $\overline{A}\,C$

2) Replace all occurances of D with 1, all occurrences of D' with 0 and all 1's with don't care. Leave 0's and don't cares as it is.

	$\overline{B}\,\overline{C}$	$\overline{B}\,C$	B C	$B\,\overline{C}$
$\overline{A}$	0	X	X	0
A	1	0	0	1

SOP2 : $A\,\overline{C}\,D$

3) Replace all occurrences of D' with 1, all occurrences of D with 0 and all 1's with don't care. Leave 0's and don't cares as it is.

	$\overline{B}\,\overline{C}$	$\overline{B}\,C$	B C	$B\,\overline{C}$
$\overline{A}$	0	X	X	0
A	0	1	1	0

SOP3 : $C\,\overline{D}$

Solve S=F(A,B,C,D)=Σm(2,3,4,5,13,15) + Πd(8, 9, 10, 11) using MEV to get minimum SOP expression.

A	B	C	D	Y	MEV
0	0	0	0	0	0
0	0	0	1	0	
0	0	1	0	1	1
0	0	1	1	1	
0	1	0	0	1	1
0	1	0	1	1	
0	1	1	0	0	0
0	1	1	1	0	
1	0	0	0	X	X
1	0	0	1	X	
1	0	1	0	X	X
1	0	1	1	X	
1	1	0	0	0	D
1	1	0	1	1	
1	1	1	0	0	D
1	1	1	1	1	

$$SOP{:}Y = \overline{B}\,C + \overline{A}\,B\,\overline{C} + A\,D$$

1) Write all the variables as 0 , leave minterms, 0's and don't cares as it is and obtain the SOP expression.

SOP1 : $\overline{B}\,C$, $\overline{A}\,B\,\overline{C}$

2) Replace all occurances of D with 1, all occurrences of D' with 0 and all 1's with don't care. Leave 0's and don't cares as it is.

SOP2 : A D

3) Replace all occurrences of D' with 1, all occurrences of D with 0 and all 1's with don't care. Leave 0's and don't cares as it is.

NO SOP

References

Books:
1. Charles H Roth and Larry L Kinney, Analog and Digital Electronics, Cengage Learning, 2019
2. Donald P Leach, Albert Paul Malvino & Goutam Saha, Digital Principles and Applications, 8th Edition,Tata McGraw Hill, 2015.

Website Links:
1. https://www.geeksforgeeks.org/introduction-of-k-map-karnaugh-map/
2. https://www.tutorialspoint.com/digital_circuits/digital_circuits_k_map_method.htm
3. https://www.tutorialspoint.com/digital_circuits/digital_circuits_quine_mccluskey_tabular_method.htm
4. https://www.geeksforgeeks.org/variable-entrant-map-vem-in-digital-logic/
5. https://www.allaboutcircuits.com/technical-articles/everything-about-the-quine-mccluskey-method/

Video Links:
1. https://youtu.be/5c16CswEG2A
2. https://youtu.be/lo3pTQ09QGI
3. https://youtu.be/PInDAFIQQNI
4. https://youtu.be/224iz7EAEXA
5. https://youtu.be/-A2aoqWckvs
6. https://youtu.be/CoMQLMqVFTI
7. https://youtu.be/1tYF6SyKLjs
8. https://youtu.be/LgS8HaHEcWM

Question bank

1] Prove the universality of NAND and NOR Gates.

2] State Duality theorem. State and prove De Morgan's first & second theorems.

3] Determine the minimum sum-of-products for –

   a) $f1(a, b, c) = \Sigma(1, 3, 4, 5, 6, 7)$

   b) $f2(a, b, c) = \Pi(2, 4, 7)$

   c) $f3(a, b, c, d) = b'c'd' + bcd + acd' + a'b'c + a'bc'd$

4] Determine the minimum product-of-sums for –

   a) $f1(a, b, c) = \Sigma(0, 1, 2, 3, 4, 6, 7)$

   b) $f2(a, b, c) = \Pi(1, 4, 5)$

   c) $f3(a, b, c, d) = b'c'd' + bcd + acd' + a'b'c + a'bc'd$

5] Solve for the simplified Boolean expression using K-Map:

$$f1(a, b, c, d) = \bar{a}\bar{c}d + \bar{a}cd + \bar{b}\bar{c}d + ab\bar{c} + \bar{a}\bar{b}cd$$
$$f2(a, b, c, d) = (a + b + \bar{d})(\bar{a} + b + \bar{d})(a + \bar{b} + \bar{c} + d)(a + \bar{b} + c + \bar{d})(a + \bar{b} + \bar{c} + d)$$

6] Find the minimum sum-of-products for –

(a) $f1(a, b, c) = m0 + m2 + m5 + m6$ (b) $f2(d, e, f) = \Sigma m(0, 1, 2, 4)$
(c) $f3(r, s, t) = rt' + r's' + r's)$ (d) $f4(x, y, z) = M0 . M5$

7] Design a 3-input, 1-output, minimal two-level gate combinational circuit; which has an output equal to 1 when majority of its inputs are at logic 1, and has output 0 when majority of inputs are at logic 0.

8] Design a minimal sum and minimal product combinational gate circuit to generate the odd parity bit for an 8421 BCD code.

9] Design (a) Binary-to-Gray Code Converter, and (b) Gray-to-Binary Code Converter

10] A switching circuit has two control inputs (C1 and C2), two data inputs (X1 and X2), and one output (Z). The circuit performs one of the logic operations AND, OR, EQU (equivalence), or XOR (exclusive OR) on the two data inputs. The function performed depends on the control inputs:

C_1	C_2	Function Performed by Circuit
0	0	OR
0	1	XOR
1	0	AND
1	1	EQU

   (i)   Derive a truth table for Z

   (ii)   Use a Karnaugh Map to find minimum AND-OR Gate Circuit to realize Z.

11] A logic circuit realizing the function f has four inputs a, b, c, d. The three inputs a, b, and c are the binary representation of the digits 0 through 7 with a being the most significant bit. The input d is an odd-parity bit; that is, the value of d is such that a, b, c, and d always contains an odd number of 1's. (For example, the digit 1 is represented by abc = 001 and d = 0, and the digit 3 is represented by abcd = 0111.) The function f has value 1 if the input digit is a prime number. (A number is prime if it is divisible only by itself and 1; 1 is considered to be prime, and 0 is not.)

    a. Draw a Karnaugh map for f

    b. Find all prime implicants of f

    c. Find all minimum sum of products for f

    d. Find all prime implicants of f'

    e. Find all minimum product of sums for f.

12] Using Quine-McCluskey method, simplify;

    a) $f(a, b, c, d) = \sum m (3, 4, 5, 7, 10, 12, 14, 15) + \sum d (2)$

    b) $f(a, b, c, d) = \sum m (1, 5, 7, 9, 11, 12, 14, 15)$

    c) $f(a, b, c, d) = \sum m (0, 1, 3, 5, 6, 7, 8, 10, 14, 15)$

    d) $f(a, b, c, d) = \sum m (1, 3, 4, 5, 6, 7, 10, 12, 13) + \sum d (2, 9)$

    e) $f(a, b, c, d) = \sum m (9, 12, 13, 15) + \sum d (1, 4, 5, 7, 81 11, 14)$.

University Questions:

1) a. Find the minimum SOP and minimum POS expressions for the following function using K-map. $f(A, B, C, D) = \sum_m (1, 3, 4, 11) + \sum_d (2, 7, 8, 12, 14, 15)$. **(06 Marks)**

  b. What are the disadvantages of K-map method? How they are overcome in Quine Mccluskey method. Simplify following function using Q-M method
$f(A, B, C, D) = \sum_m (0, 1, 2, 5, 6, 7, 8, 9, 10, 14)$. **(08 Marks)**

  c. What is Map-Entered Variable method? Using MEV method simplify following function:
$f(A, B, C, D) = \sum_m (2, 3, 4, 5, 13, 15) + dc(8, 9, 10, 11)$. **(06 Marks)**

OR

2) a. With the help of flow chart explain how to determine minimum sum of products using Karnaugh map. **(06 Marks)**

  b. Using Q-M method simplify the following function
$F(A, B, C, D) = \sum_m(2, 3, 7, 9, 11, 13) + \sum_d(1, 10, 15)$. **(08 Marks)**

  c. With example explain Petrik's method. **(06 Marks)**