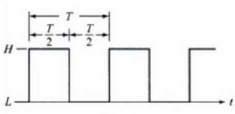
Module 2
The Combinational Logic Circuits.

INTRODUCTION:

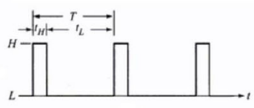
Electronic circuits and systems can be divided into two broad categories – *analog* and *digital*. Analog circuits are designed for use with small signals and are used in a linear fashion. Digital circuits are generally used with large signals and are considered nonlinear. Any quantity that changes with time can be represented as an analog signal or it can be treated as digital signal. Digital electronics involves circuits that have exactly two possible states. A system having only two states is said to be *binary*. The binary number system is widely used in digital electronics.

Hexa-Decimal	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

The operation of electronic circuits can be described in terms of its voltage levels – *high* (H) level and *low* (L) level. This could be related to the binary number system by assigning L = 0 = F (false) and H = 1 = T (true).



Symmetrical Signal



Asymmetrical Signal

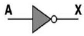
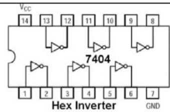
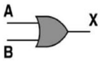
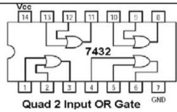
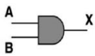
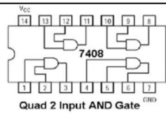
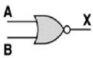
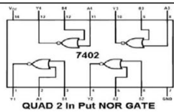
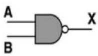
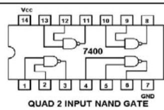
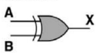
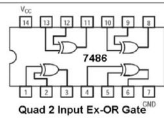
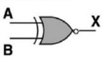
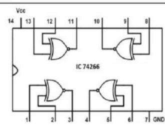
The frequency is defined as, $f = \frac{1}{T}$ where, T is the period of the signal. Duty Cycle is a convenient measure of how symmetrical or how unsymmetrical a waveform is.

$$Duty\ Cycle = \frac{T_{ON}}{T_{ON}+T_{OFF}},$$

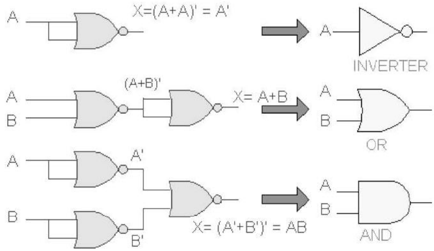
$$Duty\ Cycle, H = \frac{T_{ON}}{T_{ON}+T_{OFF}},$$

$$Duty\ Cycle, L = \frac{T_{OFF}}{T_{ON}+T_{OFF}}$$

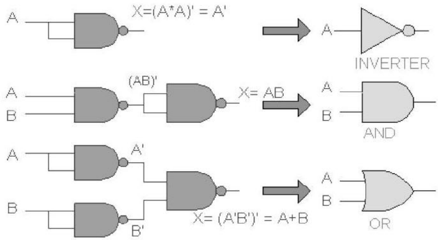
LOGIC GATES:

Circuit Symbol	Truth Table A	B	X	VHDL	IC Details
NOT Gate:  $X = \bar{A}$	0	-	1	X = ~A X <= **not** A;	
	1	-	0		
OR Gate:  $X = A + B$	0	0	0	X = A | B X <= A **or** B;	
	0	1	1		
	1	0	1		
	1	1	1		
AND Gate:  $X = A.B$	0	0	0	X = A & B X <= A **and** B;	
	0	1	0		
	1	0	0		
	1	1	1		
NOR Gate:  $X = \bar{A}.\bar{B}$	0	0	1	X = ~(A | B) X <= A **nor** B;	
	0	1	0		
	1	0	0		
	1	1	0		
NAND Gate:  $X = \bar{A} + \bar{B}$	0	0	1	X = ~(A & B) X <= A **nand** B;	
	0	1	1		
	1	0	1		
	1	1	0		
XOR Gate:  $X = A \oplus B$ $= \bar{A}B + A\bar{B}$	0	0	0	X = A ^ B X <= A **xor** B;	
	0	1	1		
	1	0	1		
	1	1	0		
XNOR Gate:  $X = A \odot B$ $= \bar{A}\bar{B} + AB$	0	0	1	X = ~(A ^ B) X <= A **xnor** B;	
	0	1	0		
	1	0	0		
	1	1	1		

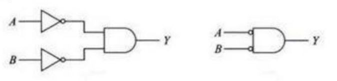
UNIVERSALITY OF NOR GATE:



UNIVERSALITY OF NAND GATE:



Bubbled AND Gate:



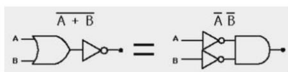
Bubbled AND gate and NOR gate are equivalent

De Morgan's First Theorem:

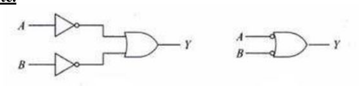
The complement of a sum equals the product of the complements. $\overline{A+B} = \overline{A}.\overline{B}$

Proof:

A	B	A+B	$\overline{A+B}$	$\overline{A}$	$\overline{B}$	$\overline{A}.\overline{B}$
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**



Bubbled OR Gate:



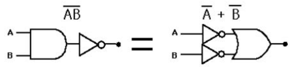
Bubbled OR gate and NAND gate are equivalent

De Morgan's Second Theorem:

The complement of a sum equals the product of the complements. $\overline{AB} = \overline{A} + \overline{B}$

Proof:

A	B	AB	$\overline{AB}$	$\overline{A}$	$\overline{B}$	$\overline{A}+\overline{B}$
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**



Duality Theorem:

Starting with a Boolean relation, you can derive another Boolean relation by –

1. Changing each OR sign to an AND sign

2. Changing each AND sign to an OR sign

3. Complementing any 0 or 1appearing in the expression.

Example:

- We say that, A+0 = A; the dual is, A.1 = A
- Consider, A(B+C) = AB + AC
  By changing the OR and AND operation, we get the dual relation:
  A + BC = (A+B)(A+C)

Laws of Boolean algebra:

The following laws are of immense use in the simplification of Boolean expressions. Note that, if A is a variable, then either A = 0 or A = 1. Also, when A = 0, A ≠ 1; and when A = 1, A ≠ 0.

De Morgan's First Theorem:-

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

$\overline{A+B} = \overline{A}.\overline{B}$ i.e., a bubbled AND gate & a NOR gate are equivalent.

De Morgan's Second Theorem:-

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

$\overline{AB} = \overline{A} + \overline{B}$ i.e., a bubbled OR gate & a NAND gate are equivalent.

1. *Commutative Law:-*
   A + B = B + A and A. B = B. A

2. *Associative Law:-*
   A + (B + C) = (A + B) + C and A . (BC) = (AB) . C

3. *Distributive Law:-*
   A(B + C) = AB + AC

4. *In relation to OR operation, the following laws hold good:-*
   A + 0 = A
   A + A = A
   A + 1 = 1 and
   $A + \bar{A} = 1$

5. *In relation to AND operation, the following laws hold good:-*
   A . 1 = A
   A . A = A
   A . 0 = 0
   $A \,.\, \bar{A} = 0$
   $\bar{\bar{A}} = A$

6. *Some more useful Boolean relations:-*
   A + AB = A
   $A + \bar{A}B = A + B$
   A (A + B) = A
   $A\,(\bar{A} + B) = AB$
   A + (B . C) = (A + B) (A + C)

Simplification of Boolean Expressions:-

The following **hints** are found to be of use, in reducing complex Boolean expressions –

1. If there are parentheses present in the given expression, they are removed first; since, multiplication should precede addition.
   E.g.:- AB + C (A + B) = AB + AC + BC

2. If there are several identical terms, all except one can be removed.
   E.g.:- A + B + C + A . 1 = A + B + C + A = A + B + C

3. If a variable repeats in a term, only one variable may be retained.
   E.g.:- A . A = A B .B . C = BC

4. If in any term, both a variable & its complement are present, that term may be removed; since, $A\bar{A} = 0$.
   E.g.:- $X\bar{X}Y = 0 \,.\, Y = 0$

5. Identify pairs of terms which contain same variables. If in a pair, a variable is absent in one term, it can be removed.
   E.g.:- ABCD + ABC = ABC (D + 1)
   = ABC . 1 since, 1 + D =1 = ABC

6. If, in a pair of terms, several variables are common, and another variable is present in one term & its complement is present in another term, this variable & its complement can be removed.
   E.g.:- $ABC + \bar{A}BC = BC\,(\bar{A} + A)$
   = BC . 1 since, $\bar{A} + A = 1$
   = BC

Module 2
Chapter 1: Karnaugh Maps.

2.1 Minimum Forms of Switching Functions:

When a function is realized using AND & OR gates, *the cost of realizing the function is directly related to the number of gates and gate inputs used*. The Karnaugh map techniques developed, lead directly to *minimum cost* two-level circuits composed of AND & OR gates. An expression consisting of a *sum-of- product* terms corresponds directly to a two-level circuit composed of a group of AND gates feeding a single OR gate (see the following Figure). Similarly, a *product-of-sums* expression corresponds to a two-level circuit composed of OR gates feeding a single AND gate. Therefore, to find minimum cost two-level AND-OR gate circuits, we must find minimum expressions in sum-of-products or product-of-sums form.

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.

Unlike the minterm expansion for a function, the minimum sum of products is not necessarily unique; that is, a given function may have two different minimum 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 following procedure:

1. Combine terms by using $X\overline{Y}+XY=(\overline{Y}+Y)=X$. Do this repeatedly to eliminate as many literals as possible. A given term may be used more than once because X+X=X.
2. Eliminate redundant terms by using the consensus theorems or other theorems.

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

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

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

None of the terms in the above expression can be eliminated by consensus. However, combining terms in a different way leads to a minimum sum of products:

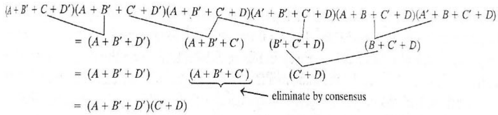
$$F = a'b'c' + a'b'c + a'bc' + ab'c + abc' + abc$$
$$= a'b' + bc' + ac$$

A *minimum product-of-sums* expression for a function is defined as a product of sum terms which

- has a minimum number of factors, and
- of all those expressions which have the same number of factors, has a minimum number of literals.

Unlike the maxterm expansion, the minimum product-of-sums form of a function is not necessarily unique. Given a maxterm expansion, the minimum product of sums can often be obtained by a procedure similar to that used in the minimum sum-of-products case, except that the theorem $(X+Y)(X+\overline{Y})= X$ is used to combine terms.

Example 2:



The uniting theorem $X\overline{Y}+ XY= X$ can be applied to minterms and products where the minterms and products are represented in algebraic notation or binary notation. The first four-variable example below illustrates this for minterms and the second for products containing three literals. The dash indicates a missing variable.

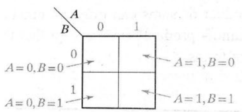
ab'cd'+ab'cd=ab'c
1010+1011=101-
Ab'c+abc=ac
101- + 111- =1-1-

Note that minterms only combine if they differ in one variable, and products only combine if they have dashes in the same position(same missing variables) and differ in one other variable. The example below do not combine.

ab'cd'+ab'c'd(will not combine)
1010 +1001
ab'c +abd(will not combine)
101- +11-1

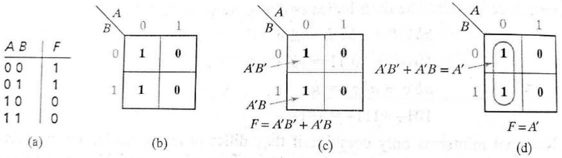
2.2 TWO AND THREE VARIABLE KARNAUGH MAPS:

Just like a truth table, the Karnaugh map of a function specifies the value of the function for every combination of values of the independent variables. The values of one variable are listed across the top of the map, and the values of the other variable are listed on the left side. Each square of the map corresponds to a pair of values for A and B as indicated.



- The figure below shows the truth table for a function F and the corresponding Karnaugh map. The value of F for A=B=0 is plotted in the upper left square, and the other map entries are plotted in a similar way in figure (b).
- Each 1 on the map corresponds to a minterm of F.
- We can read the minterms from the map just like we can read them from the truth table.
    - A 1 in square 00 for fig (c) indicates that $\overline{A}\,\overline{B}$ is a minterm of F.
    - Similarly a 1 in square 01 indicate that $\overline{A}\,B$is a minterm.
- Minterm in adjacent squares of the map can be combined since they differ in only one variable.

- Thus $\overline{A}\,\overline{B}$ and $\overline{A}B$ combine to form $\overline{A}$, and this indicated by looping the corresponding 1's on the map in fig (d).



- The following Figure shows a three-variable truth table and the corresponding Karnaugh map:
- The value of one variable (A) is listed across the top of the map, and the values of the other two variables (B, C) are listed along the side of the map.
- The rows are labelled in the sequence 00,01,11,10 so that values in adjacent rows differ in only one variable.
- For each combination of values of the variables, the value of F is read from the truth table and plotted in the appropriate map square.
- For example, for the input combination ABC=001, the values F=0 is plotted in the square for which A=0 and BC=01. For the combination ABC=110, F=1 is plotted in the A=1, BC=10 square.

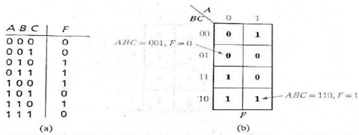
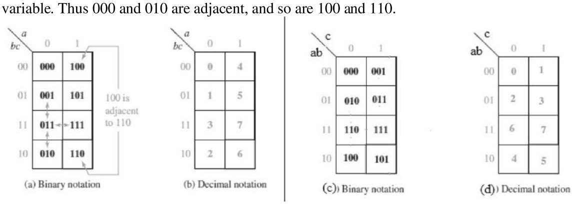


Figure below shows the location of the minterms on the three-variable map. Minterms in adjacent squares of the map differs in only one variable and therefore can be combined using the uniting theorem $X\overline{Y} + XY = X$. For example minterm 011($\bar{a}bc$) is adjacent to the three minterms with which it can be combined-001($\bar{a}\bar{b}c$), 010 ($\bar{a}b\bar{c}$), and 111(abc).
In addition to squares which are physically adjacent, the top and the bottom rows of the map are defined to be adjustment because the corresponding minterms in rows differs differ in only one variable. Thus 000 and 010 are adjacent, and so are 100 and 110.



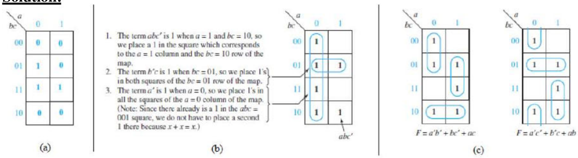
Given the minterm expansion of the function, it can be plotted on the map by placing 1's in the squares which correspond to minterms of the functions and 0's in the remaining squares.

Example:

Write the Karnaugh map for

1. $f = \sum m\left(1,3,5\right)$
2. $f\left(a,b,c\right) = ab\overline{c} + \overline{b}c + \overline{a}$
3. $f = \sum m\left(0,1,2,5,6,7\right)$

Solution:



2.3 FOUR-VARIABLE KARNAUGH MAPS

Figure shows the location of minterms on the four variable maps. Each minterm is located adjacent to the four terms with which it can combine. The definition of adjacent squares must be extended so that only are top and bottom rows adjustment as in three variable maps, but the first and the last columns are also adjustment. This requires numbering the columns in the sequence 00, 01, 11, 10 so that minterms 0 & 8, 1 & 9 etc. are adjustment squares.

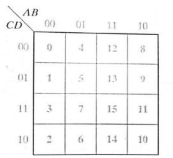
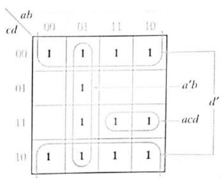


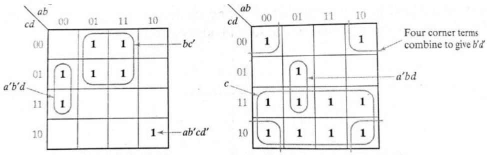
Fig: Location of Minterms on Four-Variable Karnaugh Map.

For example we now plot the following four variable expressions on the Karnaugh map.

$$f\left(a,b,c,d\right) = acd + \overline{a}b + \overline{d}$$

The first term is 1 when $a = c = d = 1$, so we place 1's in the two squares which are in the a= 1 column and cd=11 row. The term $\overline{a}b$ is 1 when ab=01, so we place four 1's in the ab=01 column. Finally, $\overline{d}$ is 1 when d =0, so we place eight 1's in the two rows for which d=0. (Duplicate 1's are not plotted because 1+1=1.)



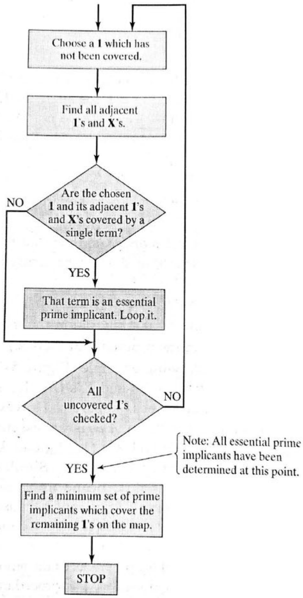


$f_1 = \Sigma m(1, 3, 4, 5, 10, 12, 13)$
$= bc' + a'b'd + ab'cd'$
(a)

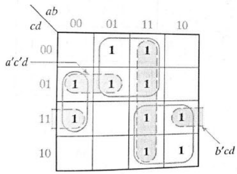
$f_2 = \Sigma m(0, 2, 3, 5, 6, 7, 8, 10, 11, 14, 15)$
$= c + b'd' + a'bd$
(b)

2.4 DETERMINATION OF MINIMUM EXPRESSIONS USING ESSENTIAL PRIME IMPLICANTS:

Any single 1 or any group of 1's which can be combined together on a map of the function F represents a product term 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 Figure shows the flowchart for determining a Minimum Sum of Products using a Karnaugh Map with an example.



Minimum solution: $F = a'b'd + bc' + ac$
All prime implicants: $a'b'd$, bc', ac, $a'c'd$, ab, $b'cd$

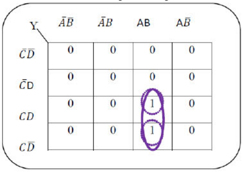


1. Choose a minterm (a 1) which has not yet been covered.
2. Find all 1's and X's adjacent to that minterm (Check the n adjacent squares on an n-variable map.
3. If a single term covers the minterm and all of the adjacent 1's and X's, then that term is an essential prime implicant, so select that term. (Note that 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 than one such set, choose a set with a minimum number of literals.)

PAIRS, QUADS, AND OCTETS:

Pairs:

The following K-map contains a pair of 1s that are horizontally adjacent. Two adjacent 1s, such as these are called a *pair*. *A pair eliminates one variable and its complement*.

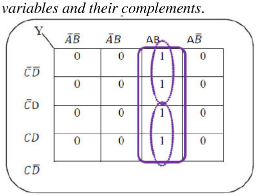


The sum-of-product equation is:

$Y = ABCD + ABCD' = ABC(D + D') = ABC$

Quad:

A *quad* is a group of four 1s that are horizontally or vertically adjacent. *A quad eliminates two variables and their complements*.

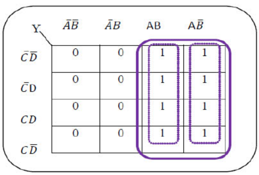


The sum-of-product equation is:

$Y = ABC' + ABC = AB(C + C') = AB$

The Octet:

The *octet* is a group of eight 1s, as shown in the following Fig. *An octet eliminates three variables and their complements*.



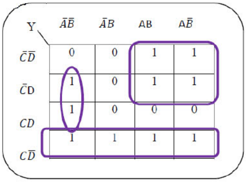
The sum-of-product equation is:

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

KARNAUGH SIMPLIFICATIONS:

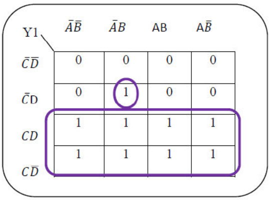
A pair eliminates one variable and its complement. A quad eliminates two variables and their complements. An octet eliminates three variables and their complements. Because of this, after drawing the K-map, first encircle the octets, then the quads, and finally the pairs, to get highest simplification.

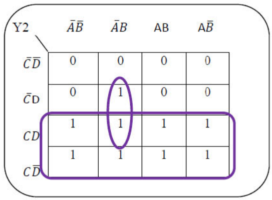
Example: *Using K-map, simplify; Y = Σm (1, 2, 3, 6, 8, 9, 10, 12, 13, 14).*
Solution:



$Y = AC' + CD' + A'B'D$

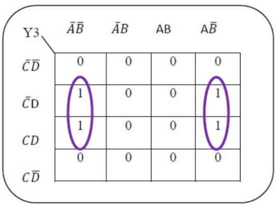
Overlapping Groups: Always overlap groups.

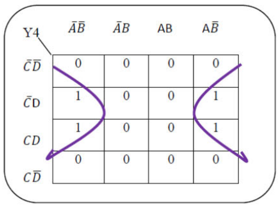




Y1 = **Y2 =**

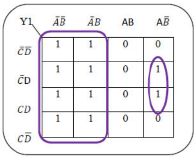
Rolling the Map:



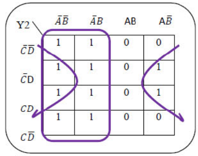


Y3 = **Y4 =**

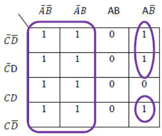
Rolling and Overlapping:



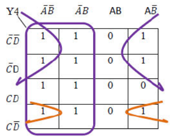
Y1 =



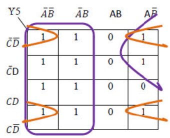
Y2 =



Y3 =

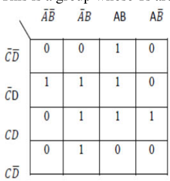


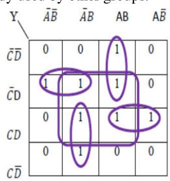
Y4 =

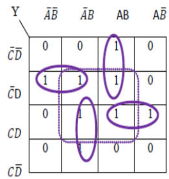


Y5 =

Eliminating Redundant Groups: After encircling groups, eliminate any *redundant groups*. This is a group whose 1s are already used by other groups.







Module 2
Chapter 2: Quine-McCluskey Method.

The Karnaugh map method is an effective way to simplify switching functions which have a small number of variables. When the number of variables is large or if several functions must be simplified, the use of a digital computer is desirable. The *Quine-McCluskey* method provides a systematic simplification procedure which can be readily programmed for a digital computer. The Quine-McCluskey method reduces the minterm expansion (standard sum-of-products form) of a function to obtain a minimum sum of products.

2.5 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.
- In the first part of the Quine-McCluskey method, all of the prime implicants of a function are systematically formed by combining minterms.
- Two minterms will combine if they differ in exactly one variable. The examples given below show both the binary notation and its algebraic equivalent.

$$AB'CD' + AB'CD = AB'C$$

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

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

$$0\ 1\ 0\ 1 + 0\ 1\ 1\ 0 \text{ (will not combine)}$$

- In order to find all of the prime implicants, all possible pairs of minterms should be compared and combined whenever possible. To reduce the required number of comparisons, the binary minterms are sorted into groups according to the number of 1's in each term.

Now, function; *f (a, b, c, d) = Σm (0, 1, 2, 5, 6, 7, 8, 9, 10, 14)* can be represented by following list of minterms:

group 0	0	0000
group 1	1	0001
	2	0010
	8	1000
group 2	5	0101
	6	0110
	9	1001
	10	1010
group 3	7	0111
	14	1110

- In this list, the term in group 0 has zero 1's, the terms in group 1 have one 1, those in group 2 have two 1's, and those in group 3 have three 1's.
- Two terms can be combined if they differ in exactly one variable. Only terms in adjacent groups must be compared.
- First, we will compare the term in group 0 with all of the terms in group 1.Terms 0000 and 0001 can be combined to eliminate the fourth variable, which yields 000– ($a'b'c'$).
- Similarly, 0 and 2 combine to form 00–0 ($a'b'd'$), and 0 and 8 combine to form –000 ($b'c'd'$). The resulting terms are listed in Column II of the following Table.
- Whenever two terms combine, the corresponding decimal numbers differ by a power of 2 (1, 2, 4, 8, etc.).
- Since the comparison of group 0 with groups 2 and 3 is unnecessary, we proceed to compare terms in groups 1 and 2. Comparing term 1 with all terms in group 2, we find that it combines with 5 and 9 but not with 6 or 10. Similarly, term 2 combines only with 6 and 10, and term 8 only with 9 and 10. The resulting terms are listed in Column 2.
- Each time a term is combined with another term, it is checked off. Also note that, a term may be used more than once. Even though two terms have already been combined with other terms, they still must be compared and combined if possible.
- At this stage, we may generate redundant terms, but these redundant terms will be eliminated later.

- We finish with Column 1 by comparing terms in groups 2 and 3. New terms are formed by combining terms 5 and 7, 6 and 7, 6 and 14, and 10 and 14.

a	b	c	d	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
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

Column 1	Column 2	Column 3
abcd	abcd	abcd

- Note that the terms in Column 2 have been divided into groups. In order to combine two terms, the terms must have the same variables, and the terms must differ in exactly one of these variables. Thus, it is necessary only to compare terms which have dashes (missing variables) in corresponding places and which differ by exactly one in the number of 1's.
- Terms in the first group in Column 2 need only be compared with terms in the second group which have dashes in the same places. Term 000– (0, 1) combines only with term 100– (8, 9) to yield –00– ($b'c'$).
- The resulting term is listed in Column 3 along with the designation 0, 1, 8, 9 to indicate that it was formed by combining minterms 0, 1, 8, and 9.
- Term (0, 2) combines only with (8, 10), and term (0, 8) combines with both (1, 9) and (2, 10).
- Again, the terms which have been combined are checked off. Comparing terms from the second and third groups in Column 2, we find that (2,6) combines with (10, 14), and (2, 10) combines with (6,14).
- Note that there are three pairs of duplicate terms in Column 3. These duplicate terms were formed in each case by combining the same set of four minterms in a different order.
- After deleting the duplicate terms, we compare terms from the two groups in Column 3. Because no further combination is possible, the process terminates.
- In general, we would keep comparing terms and forming new groups of terms and new columns until no more terms could be combined. The terms which have not been checked off because they cannot be combined with other terms are called *prime implicants*. Because every minterm has been included in at least one of the prime implicants, the function is equal to the sum of its prime implicants. In this example we have;

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

(1, 5) (5, 7) (6,7) (0, 1, 8, 9) (0, 2, 8, 10) (2, 6, 10, 14)

Definition:

- Given a function F of n variables, a product term P is an ***implicant*** of F if for every combination of values of the n 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.

Consider an Example:

$$F(a,b,c) = a'b'c' + ab'c' + ab'c + abc = b'c' + ac$$

In the above function, the implicant $a'b'c'$ is not a prime implicant because a can be eliminated, and the resulting term $b'c'$ is still an implicant of F. The implicants $b'c'$ and ac are prime implicants because if we delete a literal from either term, the term will no longer be an implicant of F.

The Quine-McCluskey method, as previously illustrated, finds all of the product term implicants of a function. The implicants which are nonprime are checked off in the process of combining terms, so that the remaining terms are prime implicants. Any nonprime term in a sum-of-products expression can thus be replaced with a prime implicant, which reduces the number of literals and simplifies the expression.

THE PRIME IMPLICANT CHART:

The second part of the Quine-McCluskey method employs a prime implicant chart to select a minimum set of prime implicants. The minterms of the function are listed across the top of the chart, and the prime implicants are listed down the side. A prime implicant is equal to a sum of minterms, and the prime implicant is said to cover these minterms. If a prime implicant covers a given minterm, an ***X*** is placed at the intersection of the corresponding row and column. The following table shows the prime implicant. All of the prime implicants (terms which have not been checked off in the above Table) are listed on the left.

-	0	1	2	5	6	7	8	9	10	14
(0, 1, 8, 9) $(b'c')$										
(0, 2, 8, 10) $(b'd')$										
(2, 6, 10, 14) (cd')										
(1, 5) $(a'c'd)$										
(5, 7) $(a'bd)$										
(6, 7) $(a'bc)$										

- In the first row, *X*'s are placed in columns 0, 1, 8, and 9, because prime implicant $b'c'$ was formed from the sum of minterms 0, 1, 8, and 9. Similarly, the all other *X*'s are placed. If a minterm is covered by only one prime implicant, then that prime implicant is called an *essential prime implicant* and must be included in the minimum sum of products. Essential prime implicants are easy to find using the prime implicant chart. If a given column contains only one *X*, then the corresponding row is an essential prime implicant. In the above Table, columns 9 and 14 each contain one X, so prime implicants $b'c'$ and cd' are essential.
- Each time a prime implicant is selected for inclusion in the minimum sum, the corresponding row should be crossed out. After doing this, the columns which correspond to all minterms covered by that prime implicant should also be crossed out.
- A minimum set of prime implicants must now be chosen to cover the remaining columns. In this example, the resulting minimum sum of products is –

- Each time a prime implicant is selected for inclusion in the minimum sum, the corresponding row should be crossed out. After doing this, the columns which correspond to all minterms covered by that prime implicant should also be crossed out.
- A minimum set of prime implicants must now be chosen to cover the remaining columns. In this example, the resulting minimum sum of products is –

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

2.6 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.

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

- First, we will label the rows of the table P_1, P_2, P_3, etc. We will form a logic function, P, which is true when all of the minterms in the chart have been covered. Let P_1 be a logic variable which is true when the prime implicant in row P_1 is included in the solution, P_2 be a logic variable which is true when the prime implicant in row P_2 is included in the solution, etc.
- Since, column 0 has *X*'s in rows P_1 and P_2, we must choose row P_1 or P_2 in order to cover minterm 0. Therefore, the expression (P_1+P_2) must be true.
- In order to cover minterm 1, we must choose row P_1 or P_3; therefore, (P_2+P_3) must be true. In order to cover minterm 2, (P_2+P_4) must be true.
- Similarly, in order to cover minterms 5, 6, and 7, the expressions (P_3+P_5), (P_4+P_6) and (P_5+P_6) must be true.
- Since we must cover all of the minterms, the following function must be true:

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

- The next step is to reduce P to a minimum sum-of-products. This is easy because there are no complements. First, we multiply out, using $(X+Y)(X+Z)=X+YZ$ and the ordinary distributive law:

$$P=(P_1+P_2 P_3)(P_4+P_2 P_6)(P_5+ P_3P_6)$$
$$=(P_1 P_4+ P_1 P_2 P_6 +P_2 P_3 P_4+ P_2 P_3 P_6)(P_5+ P_3 P_6)$$
$$= P_1 P_4 P_5+ P_1 P_2 P_5P_6+ P_2 P_3 P_4 P_5+ P_2 P_3 P_5P_6+ P_1 P_3 P_4 P_6$$
$$+ P_1 P_2 P_3 P_6+ P_2 P_3 P_4 P_6+ P_2 P_3 P_6$$

- Next, we use X+XY=X to eliminate redundant terms from P, which yields

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

- Because P must be true (P = 1) in order to cover all of the minterms, we can translate the equation back into words as follows. In order to cover all of the minterms, we must choose rows P1 and P4 and P5, or rows P1 and P2 and P5 and P6, or . . . or rows P2 and P3 and P6.

- Although there are five possible solutions, only two of these have the minimum number of rows. Thus, the two solutions with the minimum number of prime implicants are obtained by choosing rows P1, P4, and P5 or rows P2, P3, and P6.
- The first choice leads to F= $a'b'+bc'+ac$ and the second choice leads to $F=a'b'+b'c+ab$, which are the two minimum solution derived.

In summary Petrick's method is as follows number:

1. Reduce the prime implicant chart will by eliminating the essential prime implicant rows and the corresponding columns.
2. Label the rows of the 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 cover column 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 the table. 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.

The application of Petrick's method is very tedious for large charts, but it is easy to implement on a computer.

2.7 SIMPLIFICATION OF INCOMPLETELY SPECIFIED FUNCTIONS:

- Given an incompletely specific function, the proper assignment of values to the don't-care terms is necessary in order to open your minimum form for the function.
- The process of modifying the Quine McCluskey method in order to obtain a minimum solution when don't care terms are present is as follows.
- In the process of finding the prime implicants we will treat the don't-care as if they were required minterms.
- They can be combined with other minterms to eliminate as many literals as possible.
- If the extra prime implications are generated because of don't guess, this is correct because the extra prime implications will be eliminated in the next step in anyway.
- When forming the prime implicant chart, the don't-cares certain not listed at the top. This way when the prime implicant chart is solved all of the required minterms will be covered by one of the selected prime implicants.
- However, the don't care terms are not included in the final solution unless they have been used in the process of forming one of the selected prime implicants.
- The following example of simplifying an incompletely specified function should clarify the procedure.

$$F\left(A,B,C,D\right) = \Sigma m\left(2,3,7,9,11,13\right) + \Sigma d\left(1,10,15\right)$$

The don't care terms are treated like required minterms when finding the prime implicants:

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	✓		

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

	2	3	7	9	11	13
(1, 3, 9, 11)		X		X	X	
*(2, 3, 10, 11)	X	X			X	
*(3, 7, 11, 15)		X	X		X	
*(9, 11, 13, 15)				X	X	X

*Indicates an essential prime implicant.

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

- Note that although the original function was incompletely specified, the final simplified expression for F is defined for all combinations of values for A, B, C and D and is therefore completely specified. In the process of simplification we have automatically assigned values to the don't-cares in the original table for F. If we replace each term in the final expression for F by its corresponding sum of minterms, the result is

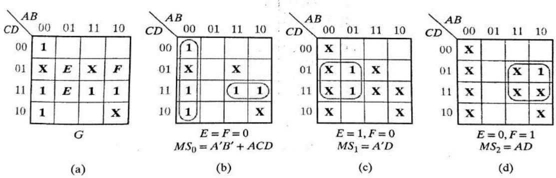
$$F = \left(m_2 + m_3 + m_{10} + m_{11}\right) + \left(m_3 + m_7 + m_{11} + m_{15}\right) + \left(m_9 + m_{11} + m_{13} + m_{15}\right)$$

Because m_{10} and m_{15} appears in this expression and m_1 does not, this implies that don't care terms in the original truth table for F have been assigned as follows
For ABCD=0001, F=0; for 1010, F=1; for 1111, F=1.

2.8 SIMPLIFICATION USING MAP-ENTERED VARIABLES:

Although the Quine-McCluskey method can be used with functions with a fairly large number of variables, it is not very efficient for functions that have many variables and relatively few terms. Some of these functions can be simplified by using a modification of the Karnaugh map method. By using map-entered variables, Karnaugh map techniques can be extended to simplify functions with more than four or five variables. The following Figure shows a four-variable map with two additional variables entered in the squares in the map.

FIGURE : Use of Map- Entered Variables



When *E* appears in a square, this means that if *E = 1*, the corresponding minterm is present in the function G, and if *E = 0*, the minterm is absent. Thus, the map represents the six-variable function;

$$G(A, B, C, D, E, F) = m_0 + m_2 + m_3 + Em_5 + Em_7 + Fm_9 + m_{11} + m_{15}$$
$$(+ \text{ don't-care terms})$$

Example: Simplify Y (A, B, C) = Σm (2, 6, 7) by using entered variable map method by taking

a) "C" as map entered variable

b) "A" as map entered variables.

Solution: Let Y = Σm (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

A	*B*	Y_C
0	0	
0	1	
1	0	
1	1	

B	*C*	Y_A
0	0	
0	1	
1	0	
1	1	

(c)	A'	A
B'	0	C
B	C'	1

$Y = AB + BC'$

(a)	A'	A
B'	0	0
B	C'	1

$Y_C = BC' + AB$

(b)	B'	B
C'	0	1
C	0	A

$Y_A = AB + BC'$

Simplification is similar to K-map method. In Fig (a), C' is grouped with 1 to get a larger group as 1 can be written ac 1 = 1 + C'. Similarly, A is grouped with 1 in Fig (b).
Now, the product term representing each group is obtained by including map entered variable (MEV) in the group as an additional ANDed term.
Hence, for Fig (a): $Y = B\bar{C} + AB$. For Fig (b): $Y = B\bar{C} + AB$.

Consider the EBM shown in Fig (c). This has only two product terms; and doesn't need a separate coverage for 1. This is because, one can write 1 = C + C', and C is included in one group and C' is included in other group.