Digital Design & Computer Organization
BCS302
Module 1

Prof Padmasree N.
Assistant Professor,
Dept. of CSE, RVITM, Bengaluru
padmasree.rvitm@rvei.edu.in

Prof Pushpa G
Assistant Professor,
Dept. of ISE, RVITM, Bengaluru
pushpag.rvitm@rvei.edu.in

Prof Krisnamurthy H
Assistant Professor,
Dept. of ISE, RVITM, Bengaluru
krishnamurthyh.rvitm@rvei.edu.in

Referred Books/Source

- Charles H Roth and Larry L Kinney, Analog and Digital Electronics, Cengage Learning,2019
- Charles H Roth and Larry L Kinney, Fundamental of Logic Design, Cengage Learning, 2017.
- Donald P Leach, Albert Paul Malvino & Goutam Saha: Digital Principles and Applications, 7th Edition, Tata McGraw Hill, 2015

Module-1

Introduction to

Digital Logic-

Karnaugh maps

&

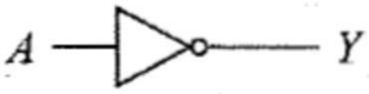
Quine-McClusky Method

Basic Gates-1

- **A *logic gate* is a digital circuit with 1 or more input voltages but only 1 output voltage.**
- Logic gates are the fundamental building blocks of digital systems.
- By connecting the different gates in different ways, we can build circuits that perform arithmetic and other functions associated with the human brain.
- **Because the circuits simulate mental processes, gates are often called *logic circuits*. NOT, OR & AND gates are the basic types of gates.**
- **The inter-connection of gates to perform a variety of logical operations is called *logic design.***
- The operation of a logic gate can be easily understood with the help of "truth table".
- **A *truth table* lists all possible combinations of inputs and the corresponding outputs.**

Basic Gates-2

- **NOT GATE (INVERTER)**
- It is a gate with only 1 input and a complemented output.

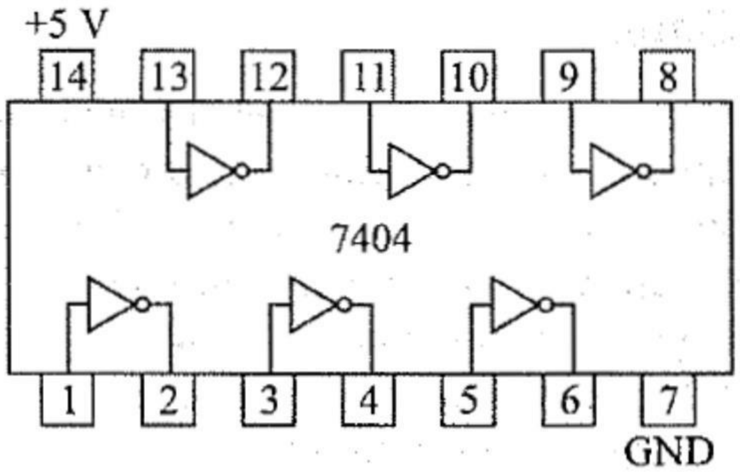


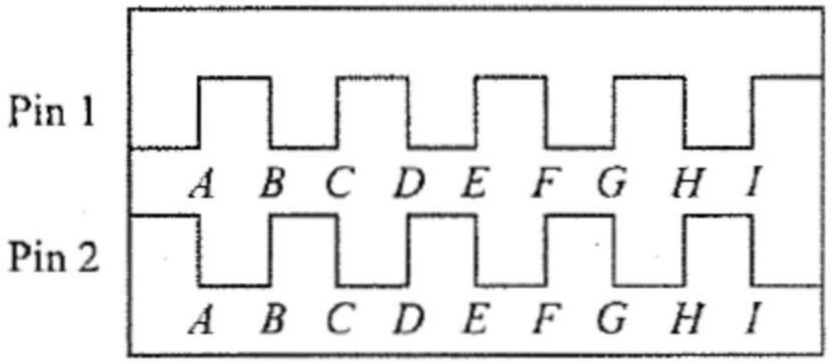
(a)

A	Y
L	H
H	L

A	Y
0	1
1	0

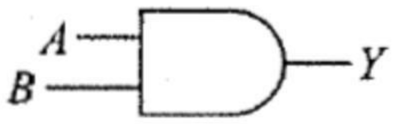
(b)





Basic Gates-3

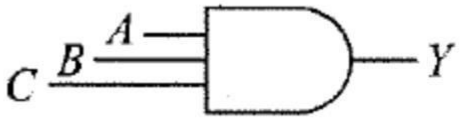
- **AND GATE**
- This is a gate with 2 or more inputs.
- The output is HIGH only when all inputs are HIGH.



(a)

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

(b)

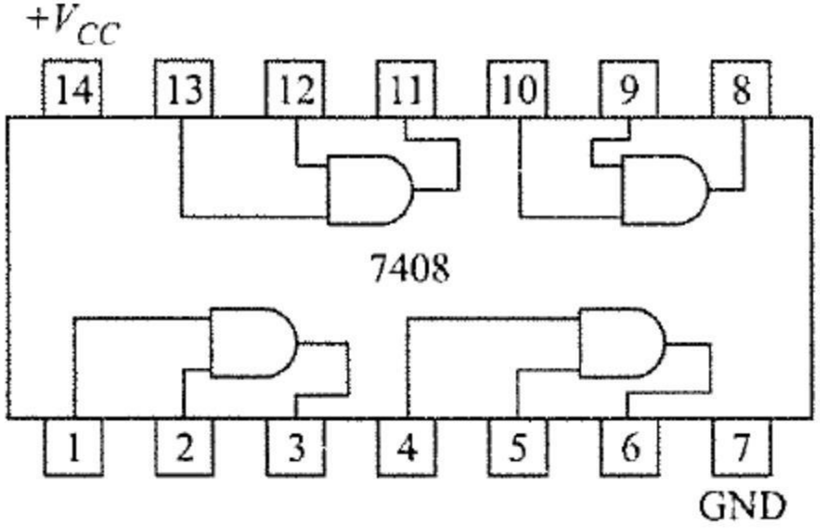


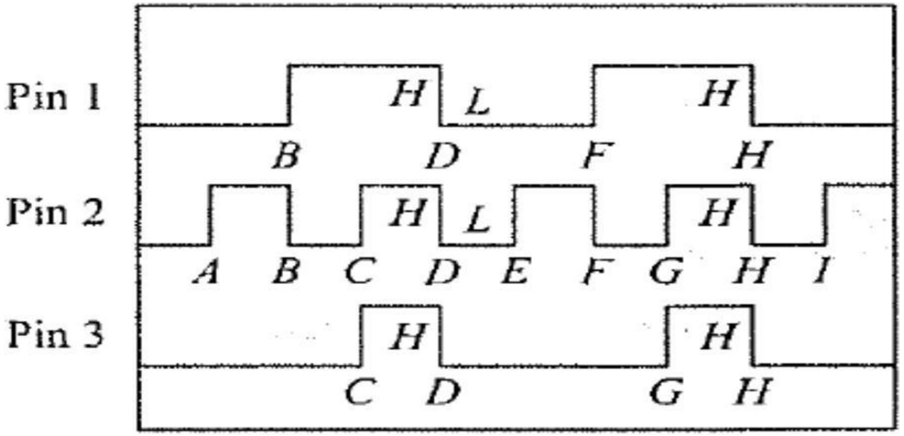
(a)

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

(b)

Basic Gates-4

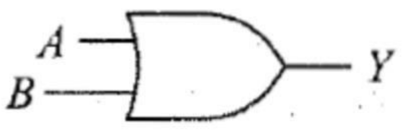




Timing diagram

Basic Gates-5

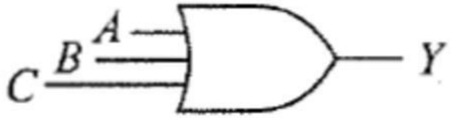
- **OR GATE**
- This is a gate with 2 or more inputs.
- The output is HIGH when any input is HIGH.



(a)

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

(b)

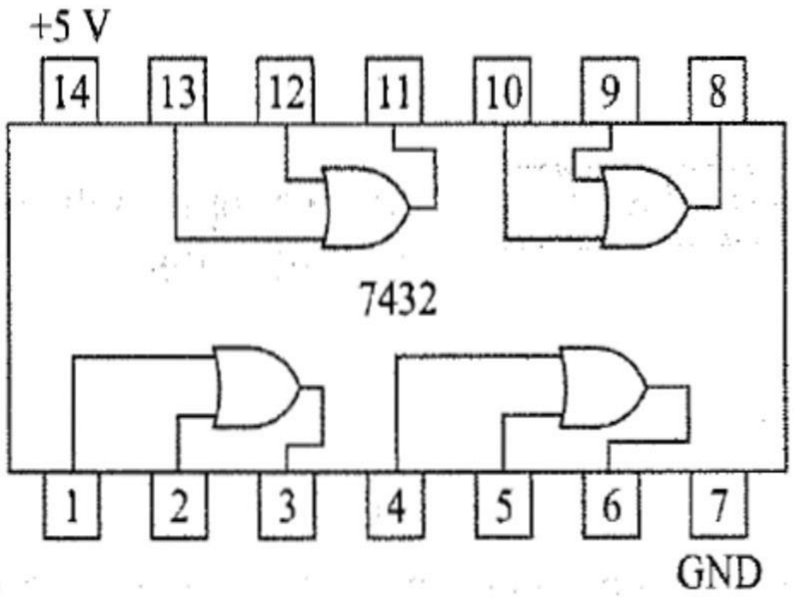


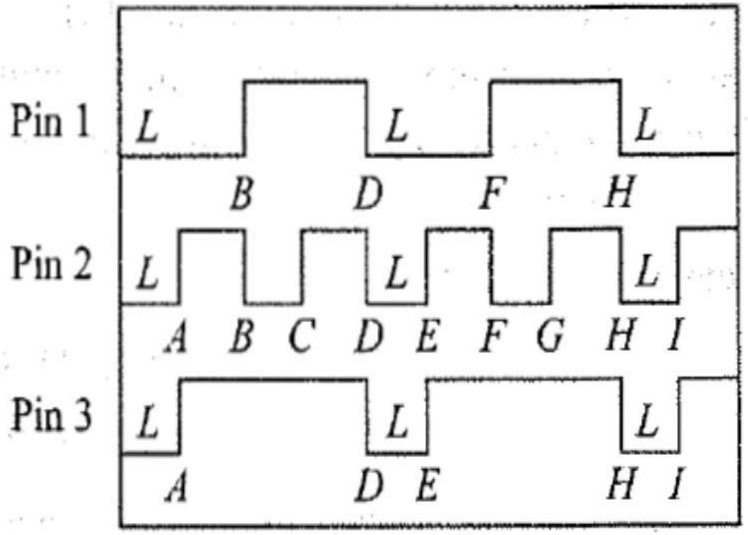
(a)

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

(b)

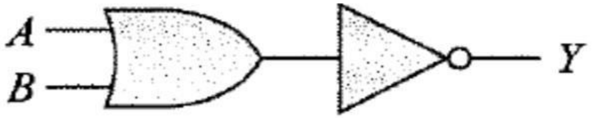
Basic Gates-6



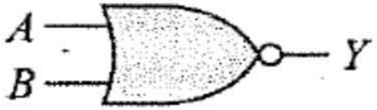


Basic Gates-9

- **NOR GATE**
- This represents an OR gate followed by an inverter.

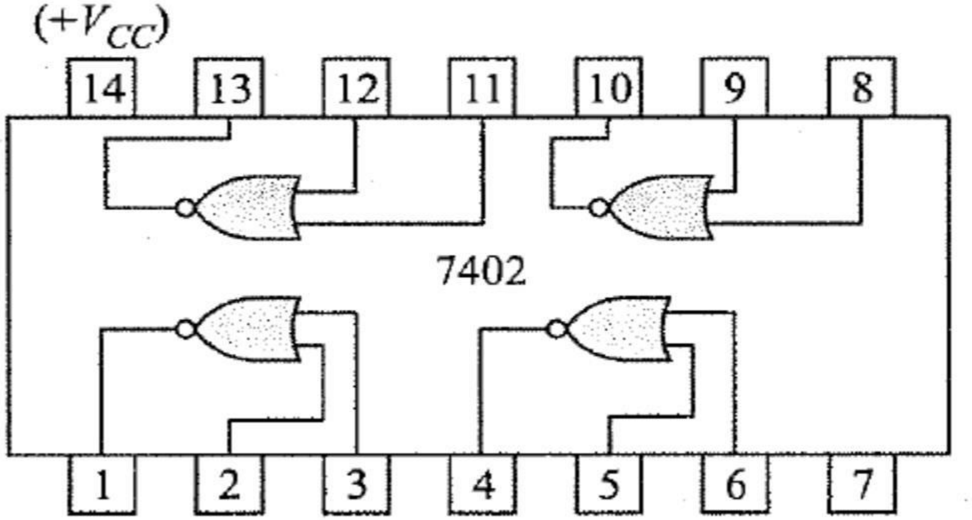


(a)



(b)

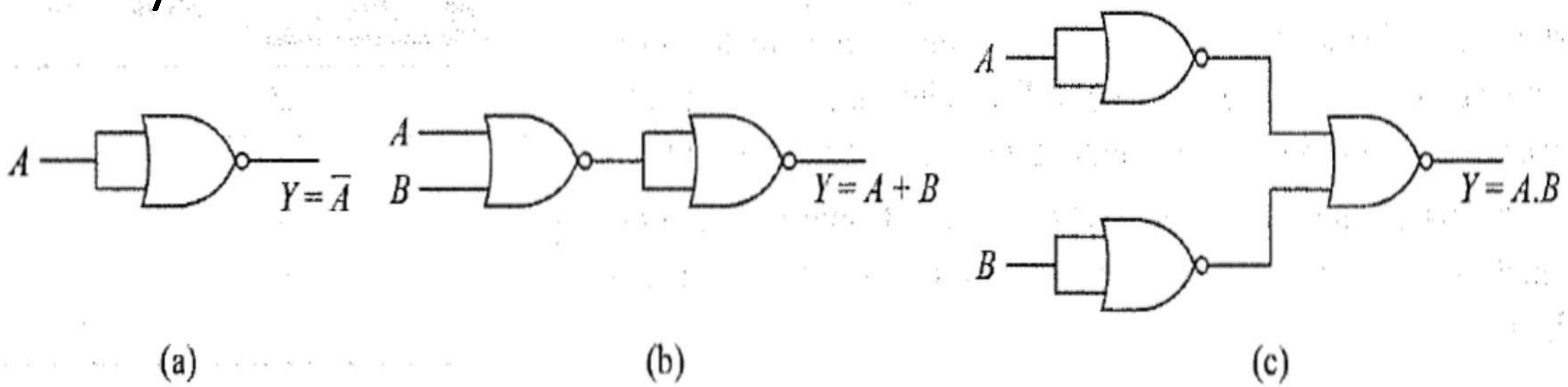
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0



(d)

Basic Gates-10

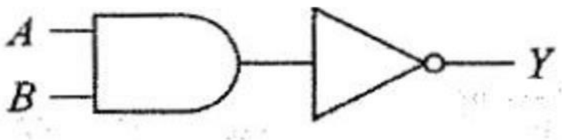
- Universality of NOR Gate



Universality of NOR gate (a) NOT from NOR, (b) OR from NOR, (c) AND from NOR

Basic Gates-11

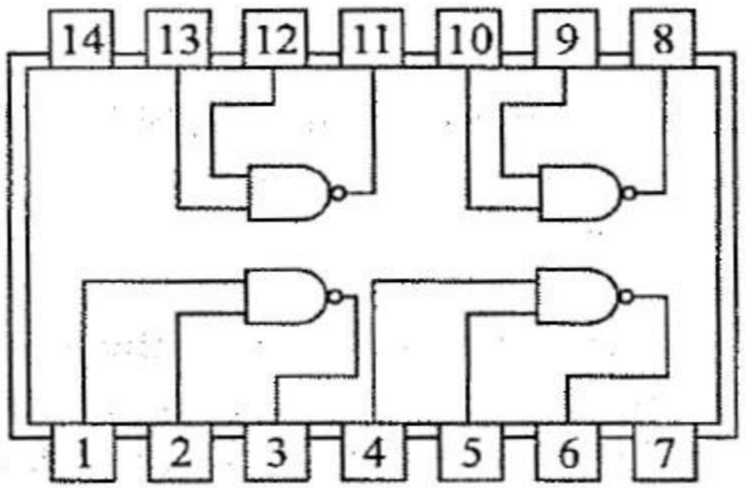
- **NAND GATE**
- This represents an AND gate followed by an inverter.



(a)

(b)

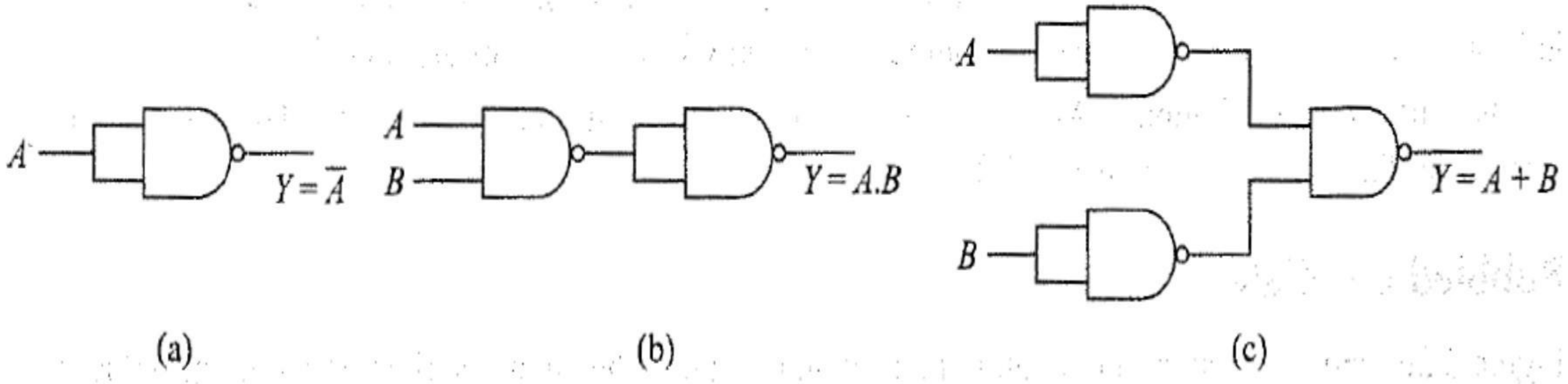
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



(d)

Basic Gates-12

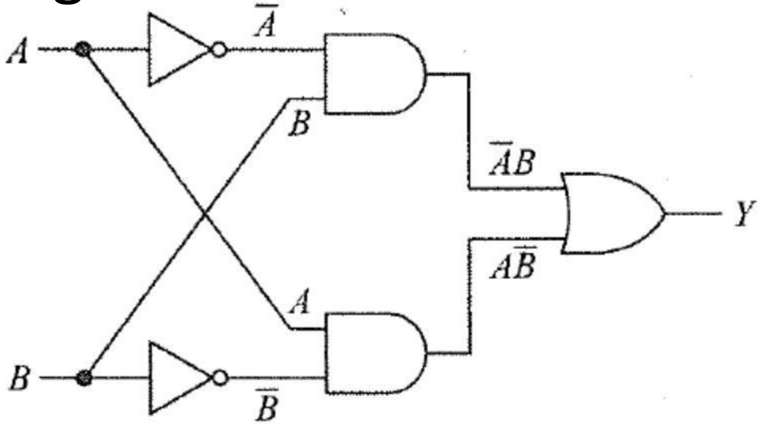
- Universality of NAND Gate



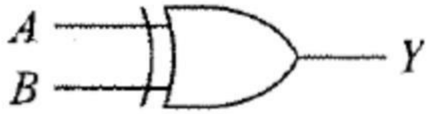
Universality of NAND gate: (a) NOT from NAND, (b) AND from NAND, (c) OR from NAND

EXCLUSIVE-OR GATES-1

- The *exclusive-OR gate* has a high output only when an odd number of inputs is high.



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

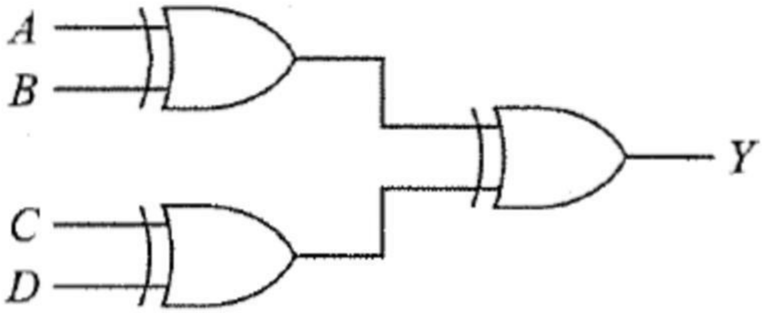


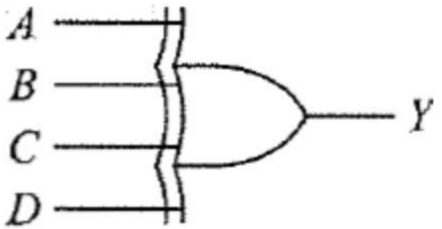
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

- OR

- Y=A⊕B

EXCLUSIVE-OR GATES-2



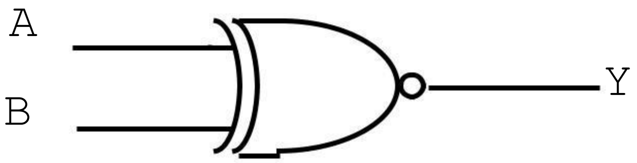


Comment	A	B	C	D	Y
Even	0	0	0	0	0
Odd	0	0	0	1	1
Odd	0	0	1	0	1
Even	0	0	1	1	0
Odd	0	1	0	0	1
Even	0	1	0	1	0
Even	0	1	1	0	0
Odd	0	1	1	1	1
Odd	1	0	0	0	1
Even	1	0	0	1	0
Even	1	0	1	0	0
Odd	1	0	1	1	1
Even	1	1	0	0	0
Odd	1	1	0	1	1
Odd	1	1	1	0	1
Even	1	1	1	1	0

EXCLUSIVE-OR GATES-2

- **EX-NOR Gate**
- **Truth Table**

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



- $Y = A \odot B$
- OR
- $Y = AB + AB$ or $Y = A \oplus B$

Introduction

- Each gate input is labelled with a **variable**.
- Each appearance of a variable or its complement in an expression will be referred to as a **literal.**
  - The following expression, which has three variables, has 10 literals
  - ab′c + a′b + a′bc′ + b′c′
- **A truth table** (also called a table of combinations) specifies the values of a Boolean expression for every possible combination of values of the variables in the expression.

Introduction

- An expression is said to be in **sum-of-products (SOP)** form when all products are the products of single variables.
  - E.g. AB′ + CD′E + AC′E′
  - A sum-of-products expression can be realized using one or more AND gates feeding a single OR gate at the circuit output.
- An expression is in **product-of-sums (POS)** form when all sums are the sums of single variables.
  - E.g. (A + B′)(C + D′ + E)(A + C′ + E′)
  - A product-of-sums expression can be realized using one or more OR gates feeding a single AND gate at the circuit output.

Introduction

Laws and Rules of Boolean Algebra

Commutative Law	$A+B=B+A$	$A\cdot B=B\cdot A$
Associative Law	$A+(B+C)=(A+B)+C$	$A\cdot(B\cdot C)=(A\cdot B)\cdot C$
Distributive Law	$A\cdot(B+C)=A\cdot C+A\cdot B$	$A+B\cdot C=(A+B)\cdot(A+C)$
Null Elements	$A+1=1$	$A\cdot 0=0$
Identity	$A+0=A$	$A\cdot 1=A$
Idempotence	$A+A=A$	$A\cdot A=A$
Complement	$A+\bar{A}=1$	$A\cdot\bar{A}=0$
Involution	$\bar{\bar{A}}=A$	
Absorption (Covering)	$A+A\cdot B=A$	$A\cdot(A+B)=A$
Simplification	$A+\bar{A}\cdot B=A+B$	$A\cdot(\bar{A}+B)=A\cdot B$
DeMorgan's Rule	$\overline{A+B}=\bar{A}\cdot\bar{B}$	$\overline{A\cdot B}=\bar{A}+\bar{B}$
Logic Adjacency (Combining)	$A\cdot B+A\cdot\bar{B}=A$	$(A+B)\cdot(A+\bar{B})=A$
Consensus	$A\cdot B+B\cdot C+\bar{A}\cdot C=A\cdot B+\bar{A}\cdot C$	$(A+B)\cdot(B+C)\cdot(\bar{A}+C)=(A+B)\cdot(\bar{A}+C)$

Introduction

- **Redundant Terms:** The Term not or no longer needed or useful.
  - (A+BC)(A+D+E)
  - =A+AD+AE+ABC+BCD+BCE
  - =A(1+D+E+BC)+BCD+BCE
  - =A+BCD+BCE

Introduction

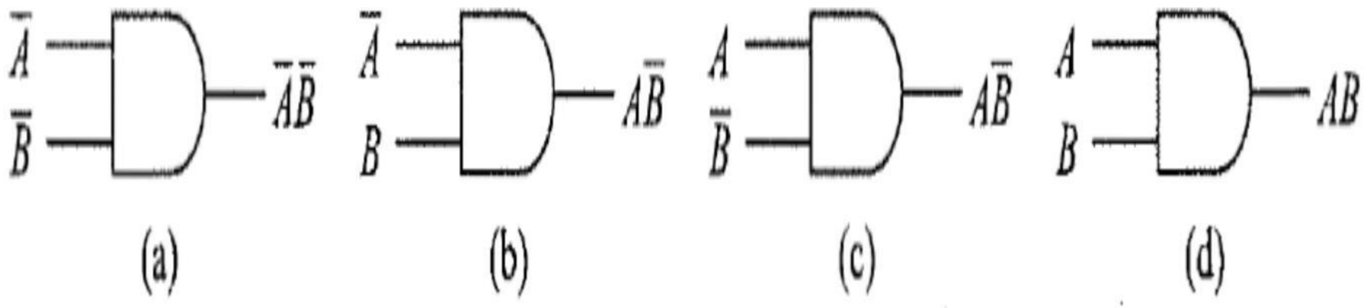
- **The Consensus Theorem**
  - Useful in simplifying Boolean expressions.
  - The consensus theorem can be used to eliminate redundant terms from Boolean expressions.
  - E.g. XY + X’Z + YZ,
    - the term YZ is redundant and can be eliminated to form the equivalent expression XY + X’Z. The term that was eliminated is referred to as the **consensus term.**
  - Given a pair of terms for which a variable appears in one term and the complement of that variable in another, the consensus term is formed by multiplying the two original terms together, leaving out the selected variable and its complement.
    - E.g.
    - The consensus of ab and a’c is bc;
    - The consensus of abd and b’de’ is (ad)(de’) = ade’.
    - The consensus of terms ab’d and a’bd’ is 0.

SUM-Of-PRODUCTS (SOP) Method-1

- Possible ways to AND two or more input signals that are in complement and uncomplement form.
- A SOP expression is two or more AND functions ORed together.

SUM-Of-PRODUCTS (SOP) Method-2

- **ANDing two variables and their complements**



A	B	*Fundamental Product*
0	0	$\bar{A}\bar{B}$
0	1	$\bar{A}B$
1	0	$A\bar{B}$
1	1	AB

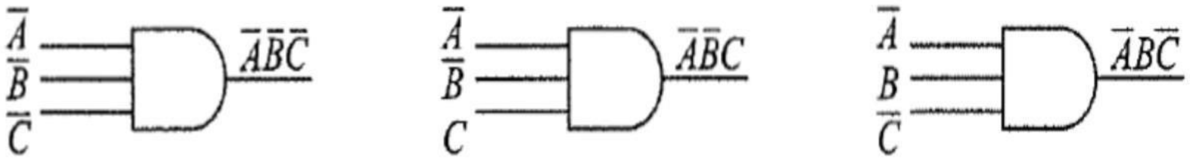
SUM-Of-PRODUCTS (SOP) Method-3

- The fundamental products are also called minterms.
- Products $\overline{A}\,\overline{B}$, AB, $A\overline{B}$, AB are represented by m0, m1, m2 and m3 respectively. The suffix i of mi comes from decimal equivalent of binary values that makes corresponding product term high.

SUM-Of-PRODUCTS (SOP) Method-4

- Example:
- Fundamental Products for Three Inputs

$\bar{A}\bar{B}\bar{C}, \bar{A}\bar{B}C, \bar{A}B\bar{C}, \bar{A}BC, A\bar{B}\bar{C}, A\bar{B}C, AB\bar{C}, ABC$



(a) (b) (c)

A	B	C	*Fundamental Products*
0	0	0	$\bar{A}\bar{B}\bar{C}$
0	0	1	$\bar{A}\bar{B}C$
0	1	0	$\bar{A}B\bar{C}$
0	1	1	$\bar{A}BC$
1	0	0	$A\bar{B}\bar{C}$
1	0	1	$A\bar{B}C$
1	1	0	$AB\bar{C}$
1	1	1	ABC

SUM-Of-PRODUCTS (SOP) Method-5

- The above three variable minterms can alternatively be represented by mo, m1, m2, m3, m4, m5, m6, and m7 respectively. Note that, for n variable problem there can be 2^n number of minterms.
- **The fundamental products by listing each one next to the input condition that results in a high output.**
- For instance, when A = 1, B = 0 and C = 0, the fundamental product results in an output of

$$Y = A\overline{B}\,\overline{C} = 1 \cdot \overline{0} \cdot \overline{0} = 1$$

Sum-of-Products Equation-1

- **Sum-of-Products Equation**
- The Sum-of-products solution, for given a truth table shown below.

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

- Write down the fundamental product for each output 1 in the truth table. For example, the first output 1 appears for an input of A = 0, B = 1, and C = 1. The corresponding fundamental product is ABC.

Sum-of-Products Equation-2

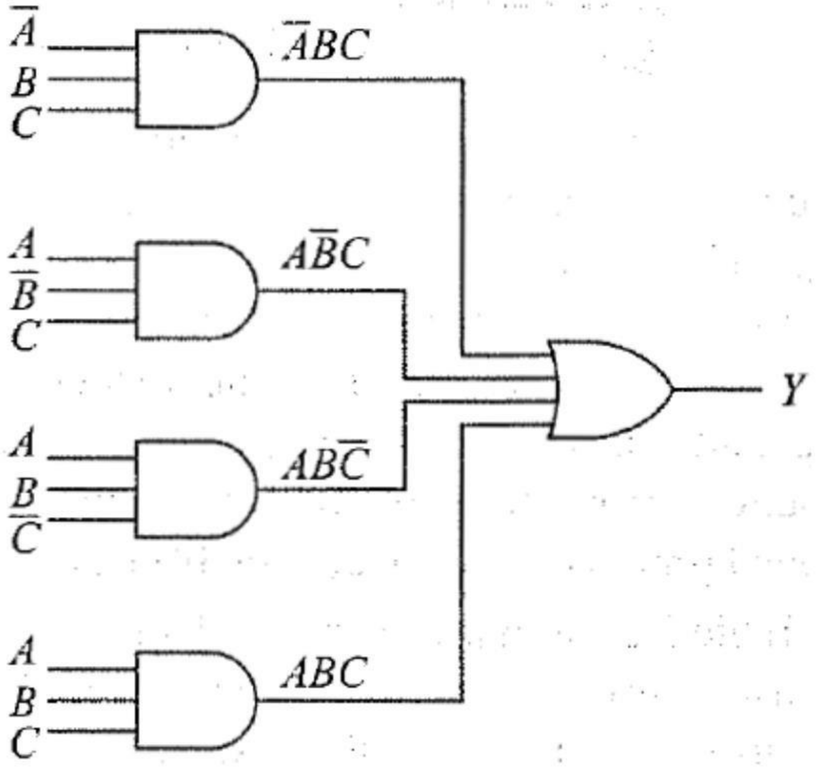
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	$1 \rightarrow \overline{A}BC$
1	0	0	0
1	0	1	$1 \rightarrow A\overline{B}C$
1	1	0	$1 \rightarrow AB\overline{C}$
1	1	1	$1 \rightarrow ABC$

- To get the sum-of-products equation, all you have to do is OR the fundamental products $Y = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$
- Alternate representation $Y = F(A, B, C) = \Sigma m\,(3, 5, 6, 7)$

Sum-of-Products Equation-3

- where '∑:' symbolizes summation or logical OR operation that is performed on corresponding minterm's and Y = F (A, B, C) means Y is a function of three Boolean variables A, B and C. This kind of representation of a truth table is also known as canonical sum form.

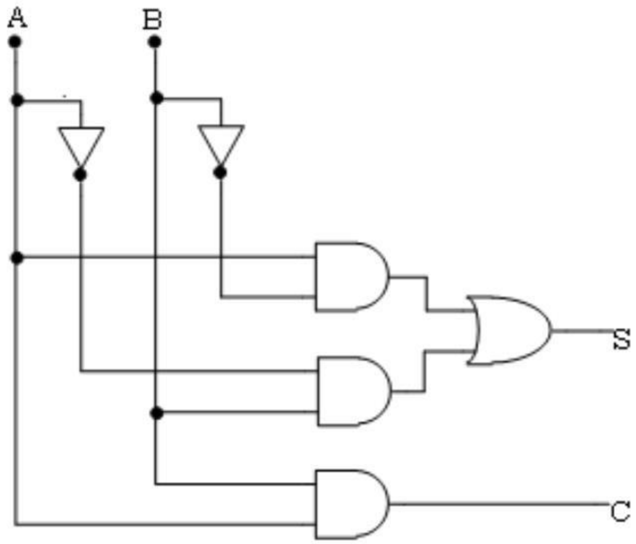
Logic Circuit



Lab Experiment

- Design and implement Half adder, Full Adder, Half Subtractor, Full Subtractor using basic gates.
  - Half Adder:
  - Truth Table for Half Adder

Input		Output	
A	B	Sum (S)	Carry (C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



  - Logic Expression
  - S=A.B+A.B
  - C=A.B

PRODUCT-Of-SUMS METHOD-1

- Given a truth table, identify the fundamental sums needed for a logic design. Then by ANDing these sums, will get the product-of-sums equation corresponding to the truth table.
- But, in the sum-of-products method, the fundamental product produces an output l for the corresponding input condition.
- But with the product of- sums method, the fundamental sum produces an output 0 for the corresponding input condition.

PRODUCT-Of-SUMS METHOD-2

- **Converting** a **Truth Table to an Equation**

A	B	C	Y	Maxterm
0	0	0	$0 \rightarrow A+B+C$	M_0
0	0	1	1	M_1
0	1	0	1	M_2
0	1	1	$0 \rightarrow A+\overline{B}+\overline{C}$	M_3
1	0	0	1	M_4
1	0	1	1	M_5
1	1	0	$0 \rightarrow \overline{A}+\overline{B}+C$	M_6
1	1	1	1	M_7

$$Y = A + B + C = 0 + 0 + 0 = 0$$

$$Y = A + \overline{B} + \overline{C} = 0 + \overline{1} + \overline{1} = 0 + 0 + 0 = 0$$

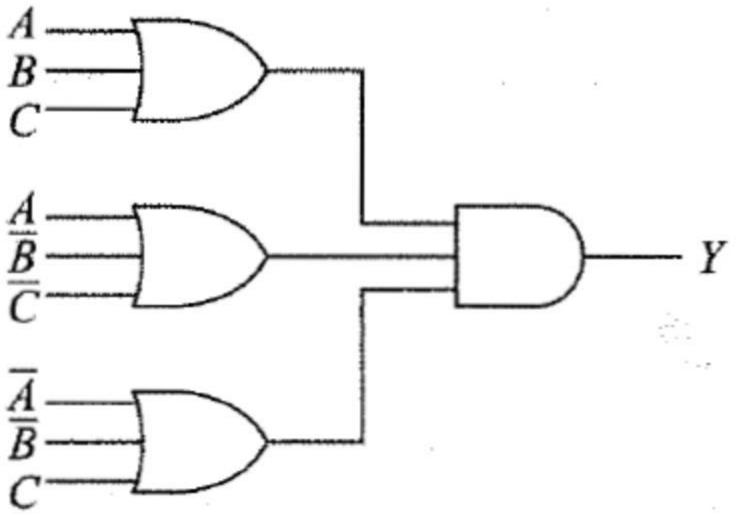
$$Y = \overline{A} + \overline{B} + C = \overline{1} + \overline{1} + 0 = 0 + 0 + 0 = 0$$

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

$$Y = F(A, B, C) = \Pi M(0, 3, 6)$$

PRODUCT-Of-SUMS METHOD-3

- **Logic Circuit**



PRODUCT-Of-SUMS METHOD-4

- **Conversion between SOP and POS**
- SOP and POS occupy complementary locations in a truth table.
  - Identifying complementary locations,
  - Changing minterm to maxterm or reverse, and
  - Changing summation by product or reverse.

$$Y = F(A, B, C) = \Pi M(0, 3, 6) = \Sigma m(1, 2, 4, 5, 7)$$

$$Y = F(A, B, C) = \Sigma m(3, 5, 6, 7) = \Pi M(0, 1, 2, 4)$$

Conversion between SOP and POS

- Mapping between canonical forms
- Minterm to maxterm conversion
  - use maxterms whose indices do not appear in minterm expansion
  - e.g. F(A,B,C) = ∑m(1,3,5,6,7) = ∏M(0,2,4)
- Maxterm to minterm conversion
  - use minterms whose indices do not appear in maxterm expansion
  - e.g. F(A,B,C) = ∏ M(0,2,4) = ∑ m(1,3,5,6,7)
- Minterm expansion of F to minterm expansion of F
  - use minterms whose indices do not appear
  - e.g., F(A,B,C) = ∑ m(1,3,5,6,7) F(A,B,C) = ∑ m(0,2,4)
- Maxterm expansion of F to maxterm expansion of F
  - use maxterms whose indices do not appear
  - e.g., F(A,B,C) = ∏ M(0,2,4) F'(A,B,C) = ∏ M(1,3,5,6,7)

DON'T-CARE CONDITIONS-1

- In digital systems, certain input conditions never occur during normal operation; therefore, the corresponding output never appears. Since the output never appears, it is indicated by an *X* in the truth table.
- The *X* is called a *don 't-care condition.* Whenever you see an X in a truth table, you can let it equal either 0 or 1, whichever produces a simpler logic circuit.

DON'T-CARE CONDITIONS-2

- **Truth Table with Don’t-Cares**
- In Minterm form
  - $F = \Sigma\, m(0, 3, 7) + \Sigma\, d(1, 6)$
- In Maxterm form
- $F = \Pi\, M(2, 4, 5) \cdot \Pi\, D(1, 6)$

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

DON'T-CARE CONDITIONS-2

- **Truth Table with Don't-Care Conditions**

A	B	C	D	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	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

- **$Y=F(A,B,C,D) = \sum m\ (9) + d(10,11,12,13,14,15)$**

Problems

- Design a logic circuit that has three inputs, A, B, and C, and whose output will be HIGH only when a majority of the inputs are HIGH.
-

Karnaugh Maps

- Switching functions can generally be simplified by using the algebraic techniques.
- However, two problems arise when algebraic procedures are used:
  - 1. The procedures are difficult to apply in a systematic way.
  - 2. It is difficult to tell when you have arrived at a minimum solution.
- The Karnaugh map method and the Quine-McCluskey procedure used to overcome these difficulties by providing systematic methods for simplifying switching functions.
- The Karnaugh map is useful tool for simplifying and manipulating switching functions of three or four variables (Literals).

Karnaugh Maps
Minimum Forms of Switching Functions

- When a function is realized using AND 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 that lead directly to minimum cost two-level circuits composed of AND 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.
- 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.

Karnaugh Maps
Minimum Forms of Switching Functions

- A minimum sum-of-products expression for a function is defined as a sum of product terms which
  - (a) has a minimum number of terms and
  - (b) of all those e xpressions 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) a minimum number of gates and
  - (b) 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.**

Karnaugh Maps
Minimum Forms of Switching Functions

- Given a minterm expansion, the minimum sum-of-products form can often be obtained by the following procedure:
  - 1. Combine terms by using the uniting theorem XY′ + XY = 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 theorem or other theorems.
- Note: Unfortunately, the result of this procedure may depend on the order in which terms are combined or eliminated so that the final expression obtained is not necessarily minimum.

Karnaugh Maps
Minimum Forms of Switching Functions

- Find a minimum sum-of-products expression for F(a, b, c) = m Σ (0, 1, 2, 5, 6, 7)

Karnaugh Maps
Minimum Forms of Switching Functions

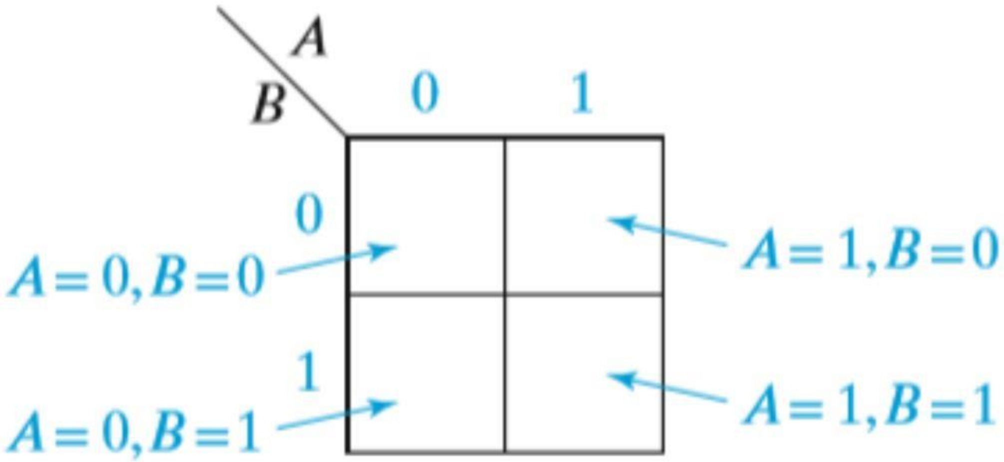
- A **minimum product-of-sums** expression for a function is defined as a product of sum terms which
  - (a) has a minimum number of terms, and
  - (b) of all those expressions which have the same number of terms, has a minimum number of literals.
- Unlike the maxterm expansion, the minimum product-of-sums form of a function is not necessarily unique.
- Minimum product of sums can often be obtained by uniting theorem $(X + Y)(X + Y') = X$ is

Karnaugh Maps
Minimum Forms of Switching Functions

- 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
  - 1 0 1– + 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.

Karnaugh Maps

Two- and Three-Variable Karnaugh Maps

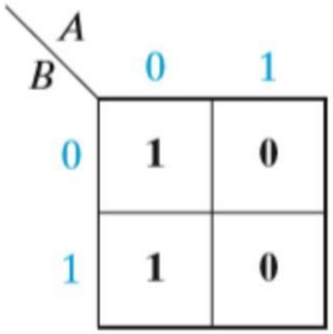


Karnaugh Maps

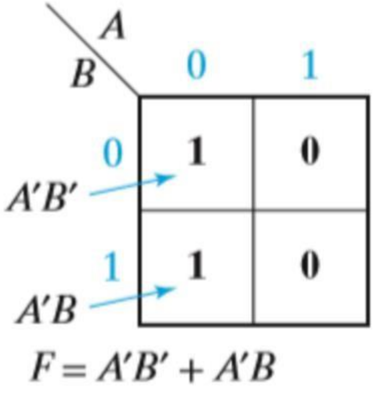
Two- and Three-Variable Karnaugh Maps

A B	F
0 0	1
0 1	1
1 0	0
1 1	0

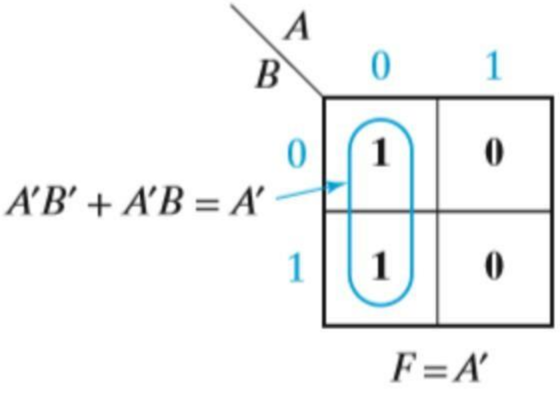
(a)



(b)



(c)



(d)

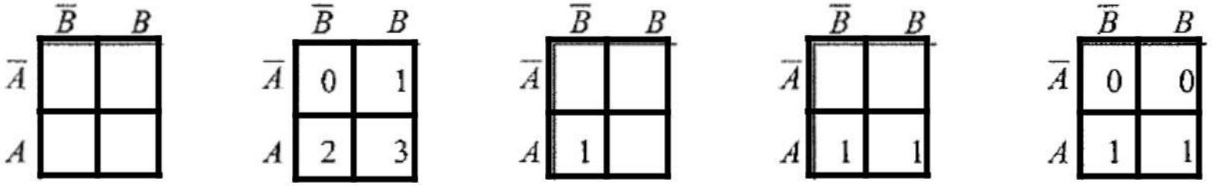
- Minterms in adjacent squares of the map can be combined since they differ in only one variable.
- Thus, A′B′ and A′B combine to form A′, and this is indicated by looping (Grouping) the corresponding 1’s on the map in Figure

Karnaugh Maps

Two- and Three-Variable Karnaugh Maps

- **Two-Variable Maps**
- Example: Y =F(A,B)= ∑m(2,3)

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

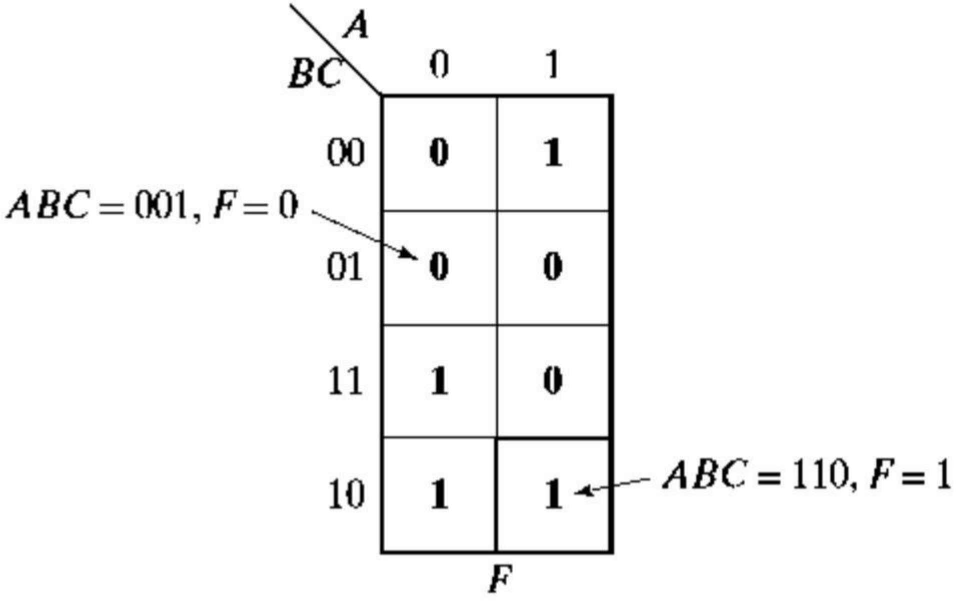


Karnaugh Maps

Two- and Three-Variable Karnaugh Maps

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

(a)



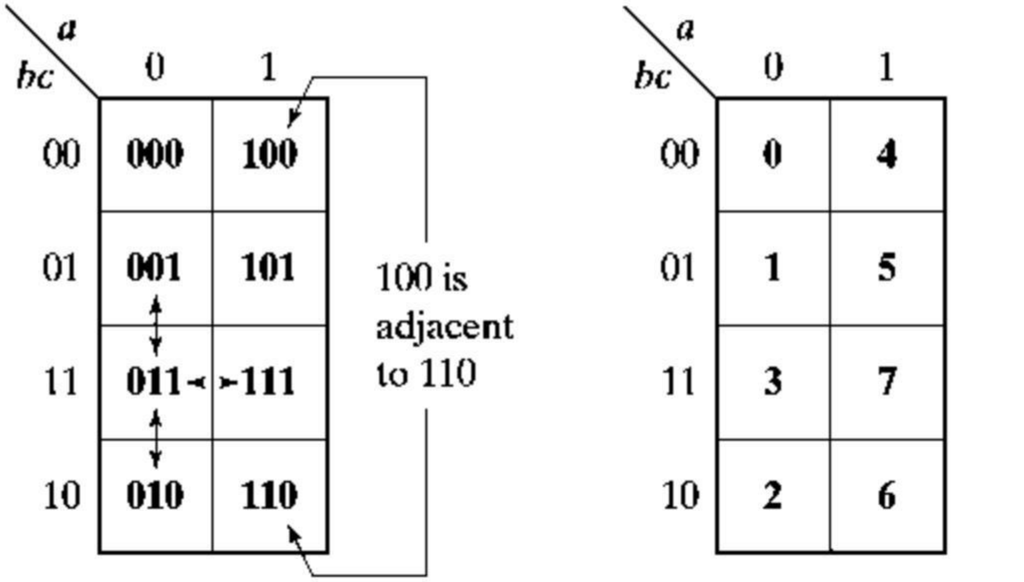
(b)

The rows are labeled in the sequence 00, 01, 11, 10 so that values in adjacent rows differ in only one variable.

Karnaugh Maps

Two- and Three-Variable Karnaugh Maps

- Location of Minterms on a Three-Variable Karnaugh Map



(a) Binary notation (b) Decimal notation

Karnaugh Maps

Two- and Three-Variable Karnaugh Maps

- **Three-Variable Maps**
- Example: Y = F(A, B, C) = ∑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

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

	$\bar{C}$	C
$\bar{A}\bar{B}$	0	1
$\bar{A}B$	2	3
AB	6	7
$A\bar{B}$	4	5

	$\bar{C}$	C
$\bar{A}\bar{B}$		
$\bar{A}B$	1	
AB	1	1
$A\bar{B}$		

	$\bar{C}$	C
$\bar{A}\bar{B}$	0	0
$\bar{A}B$	1	0
AB	1	1
$A\bar{B}$	0	0

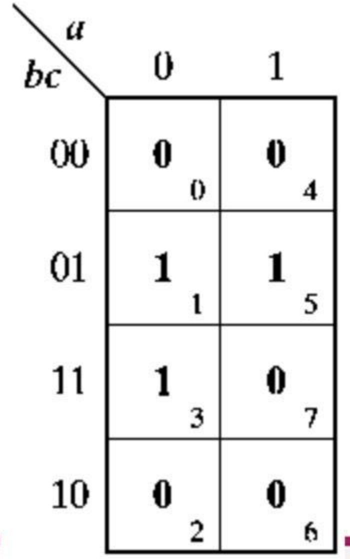
Karnaugh Maps

Two- and Three-Variable Karnaugh Maps

- Given the minterm expansion of a function, it can be plotted on a map by placing 1's in the squares which correspond to minterms of the function and 0's in the remaining squares (the 0's may be omitted if desired).
- E.g. Karnaugh Map of

$$F(a, b, c) = \Sigma\, m(1, 3, 5)$$

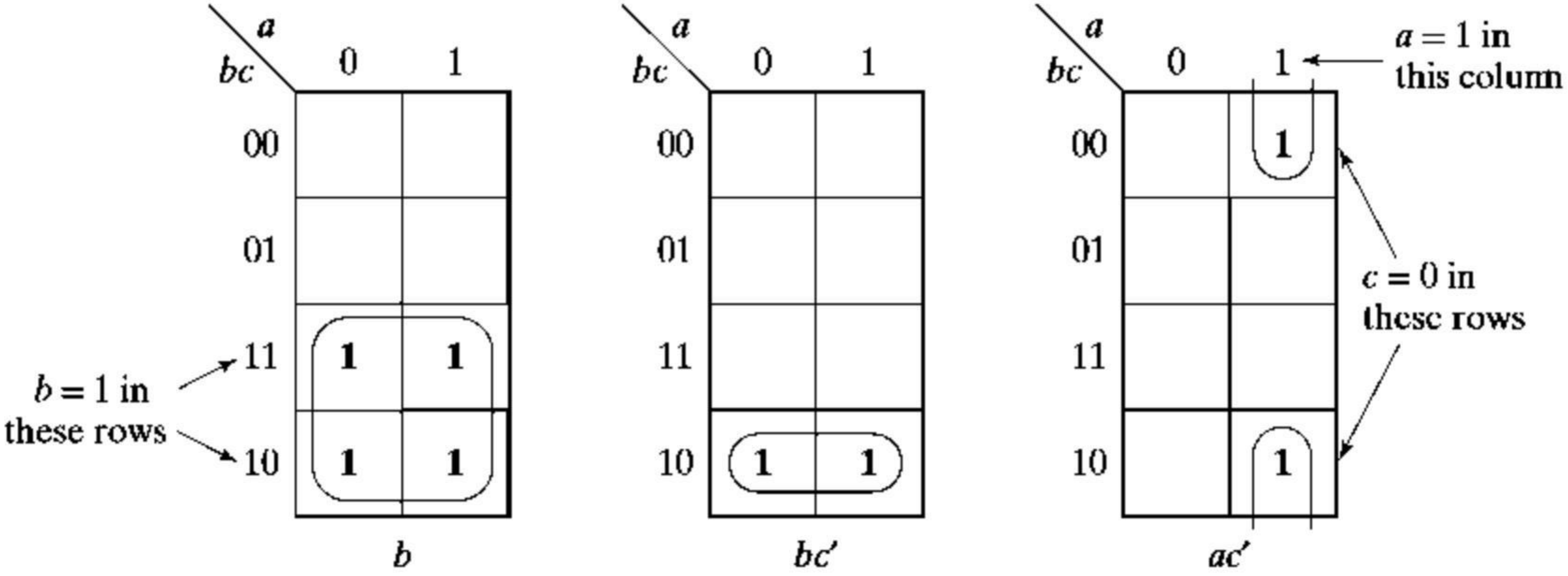
$$= \Pi\, M(0, 2, 4, 6, 7)$$



bc \ a	0	1
00	0 (0)	0 (4)
01	1 (1)	1 (5)
11	1 (3)	0 (7)
10	0 (2)	0 (6)

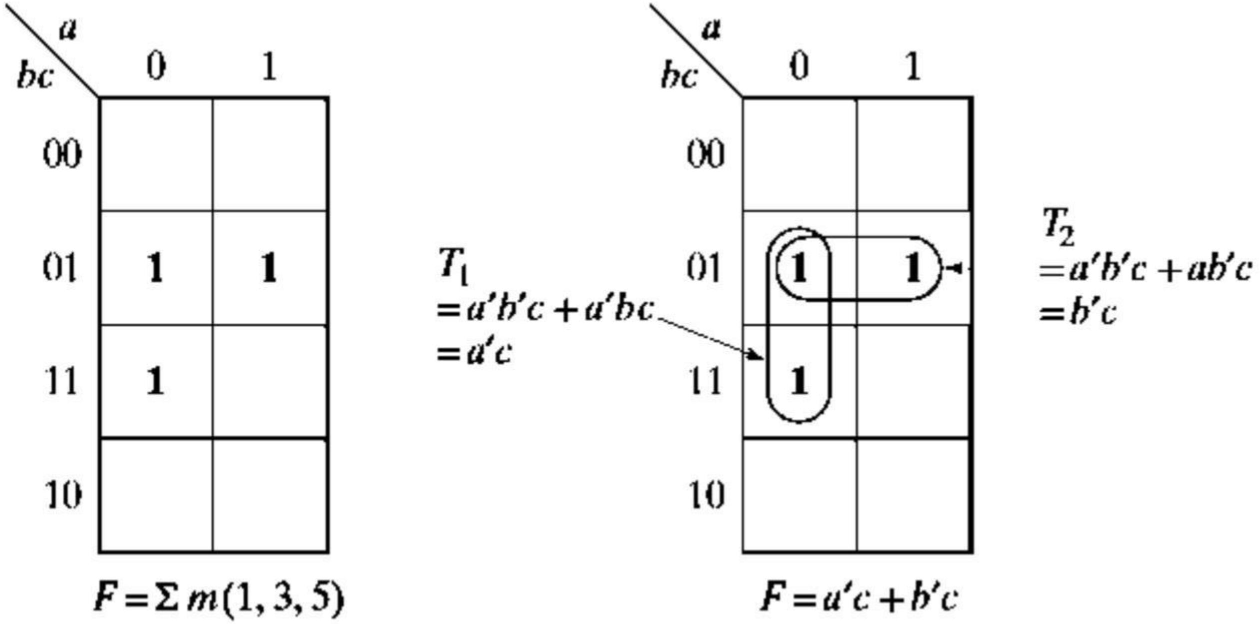
Karnaugh Maps
Two- and Three-Variable Karnaugh Maps

- Karnaugh Maps for Product Terms



Karnaugh Maps
Two- and Three-Variable Karnaugh Maps

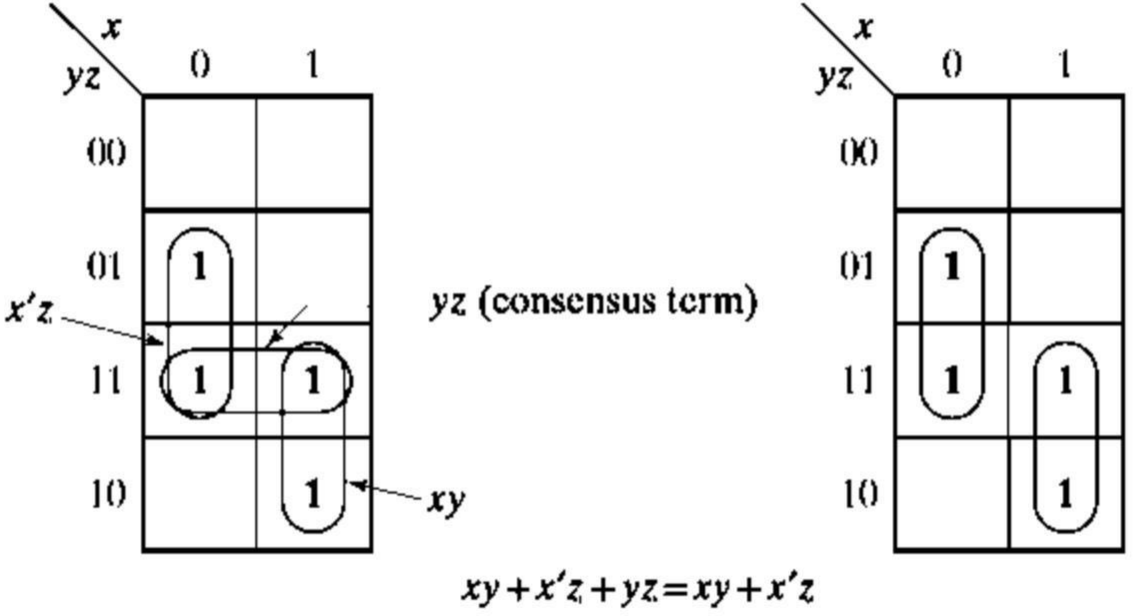
- Simplification of a Three-Variable Function
  - Terms in adjacent squares on the map differ in only one variable and combine them.
  - E.g. a’b’c and a’bc combine to form a’c, and a’b’c and ab’c combine to form b’c,



(a) Plot of minterms
(b) Simplified form of F

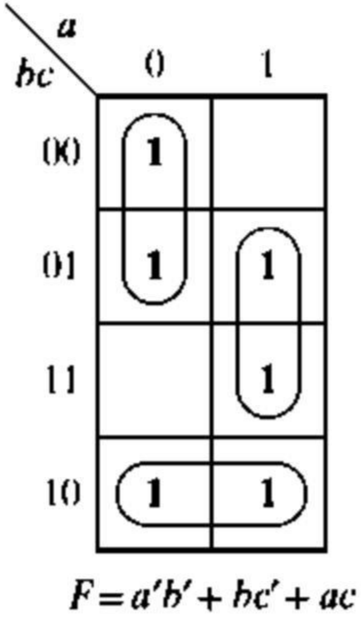
Karnaugh Maps
Two- and Three-Variable Karnaugh Maps

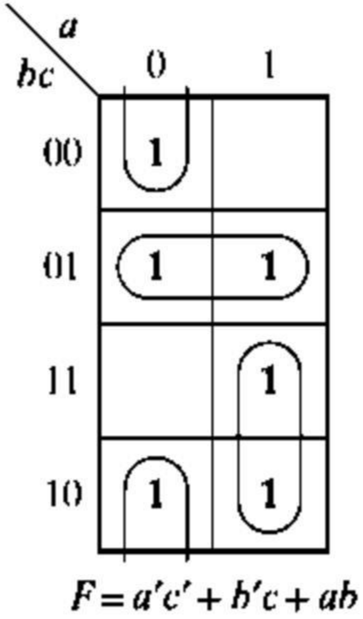
- Karnaugh Maps that Illustrate the Consensus Theorem
  - XY+X’Z+YZ=XY+X’Z
  - Note that the consensus term (YZ) is redundant because its 1’s are covered by the other two terms.



Karnaugh Maps
Two- and Three-Variable Karnaugh Maps

- Function with Two Minimum Forms
  - Two minimum solutions for F = Σ m(0, 1, 2, 5, 6, 7).





Karnaugh Maps
Two- and Three-Variable Karnaugh Maps

- The Karnaugh map method is easily extended to functions with **don't-care terms**.
- The required minterms are indicated by 1's on the map, and the don't-care minterms are indicated by **X's**.
- When choosing terms to form the minimum sum of products, all the 1's must be covered, but the X's are only used if they will simplify the resulting expression.

Karnaugh Maps
Two- and Three-Variable Karnaugh Maps

- Exercise Problems:

1. Given F1 =Σ m(0, 4, 5, 6) and F2 =Σ m(0, 3, 6, 7) find the minterm expression for F1 and F2 using K-map.

Karnaugh Maps
Two- and Three-Variable Karnaugh Maps

- Exercise Problems:

1. Work (a) and (b) with the following truth table:
   a) Find the simplest expression for F, and specify the values of the don't-cares that lead to this expression.
   b) Repeat (a) for G. (Hint: Can you make G the same as one of the inputs by properly choosing the values for the don't-care?)

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

Karnaugh Maps
Two- and Three-Variable Karnaugh Maps

- Exercise Problems:
- Find the minimum sum of products for each function using a Karnaugh map.
  - (a) f1(a, b, c) = m0 + m2 + m5 + m6
  - (b) f2(d, e, f) = Σ m(0, 1, 2, 4)
  - (c) f3(r, s, t) = rt′ + r′s′ + r′s
  - (d) f4(x, y, z) = M0 · M5

Karnaugh Maps Four-Variable Karnaugh Maps

- Location of Minterms on Four-Variable Karnaugh Map

CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Karnaugh Maps Four-Variable Karnaugh Maps

- Example: Y= F(A,B,C,D)= ∑m(1,6,7,14)

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

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$				
$\bar{A}B$				
AB				
$A\bar{B}$				

(a)

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

(b)

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

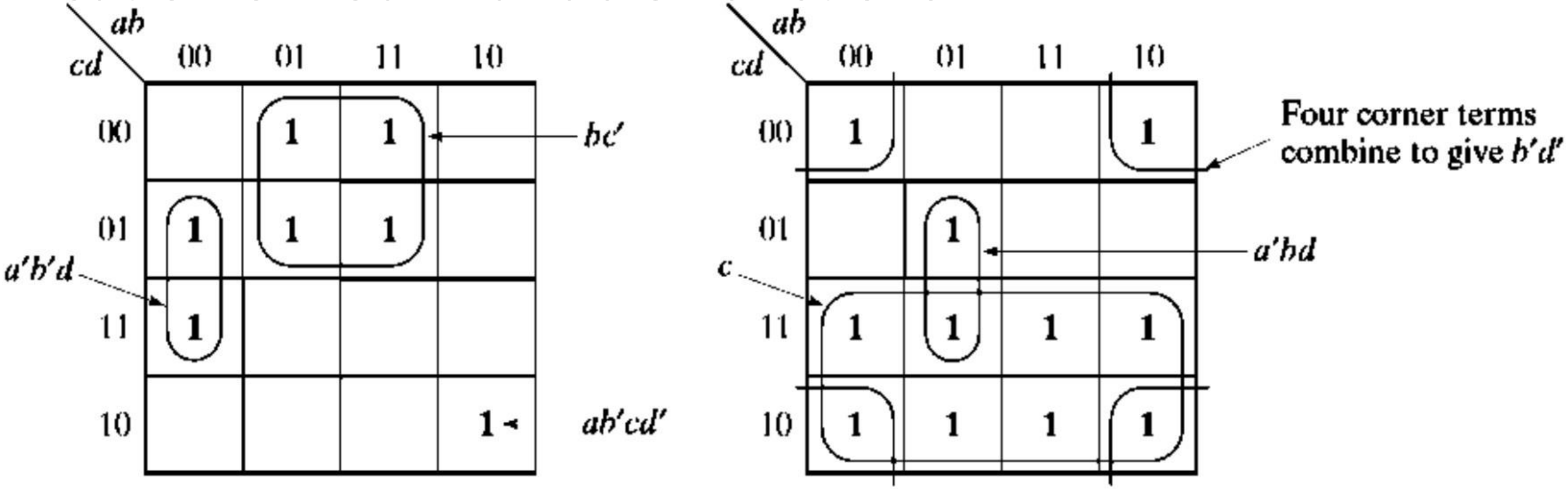
(c)

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	0	0
$\bar{A}B$	0	0	1	1
AB	0	0	0	1
$A\bar{B}$	0	0	0	0

(d)

Karnaugh Maps Four-Variable Karnaugh Maps

- Simplification of Four-Variable Functions



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

(a)

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

(b)

- Minterms can be combined in groups of two, four, or eight to eliminate one, two, or three variables, respectively.

Karnaugh Maps Four-Variable Karnaugh Maps

- Simplification of an Incompletely Specified Function

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

$$f = \Sigma\, m(1, 3, 5, 7, 9) + \Sigma\, d(6, 12, 13)$$
$$= a'd + c'd$$

Karnaugh Maps
Four-Variable Karnaugh Maps

- A minimum product of sums can also be obtained from the map.
  - Because the 0’s of f are 1’s of f’, the minimum sum of products for f’ can be determined by looping the 0’s on a map of f.
  - The complement of the minimum sum of products for f’ is then the minimum product of sums for f.
  - The following example illustrates this procedure for
    - **f = x’z’ + wyz + w’y’z’ + x’y**
  - Then, from the 0’s,
    - **f ’ = y’z + wxz’ + w’xy**
  - and the minimum product of sums for f is
    - **f = (y + z’)(w’ + x’ + z)(w + x’ + y’)**

yz \ wx	00	01	11	10
00	1	1	0	1
01	0	0	0	0
11	1	0	1	1
10	1	0	0	1

KARNAUGH SIMPLIFICATIONS-1

- The Karnaugh map uses the following rules for the simplification of expressions by *grouping* together adjacent cells containing *ones.*
- After drawing a Kamaugh map,
  - Encircle the octets first,
  - The quads second, and
  - The pairs last.

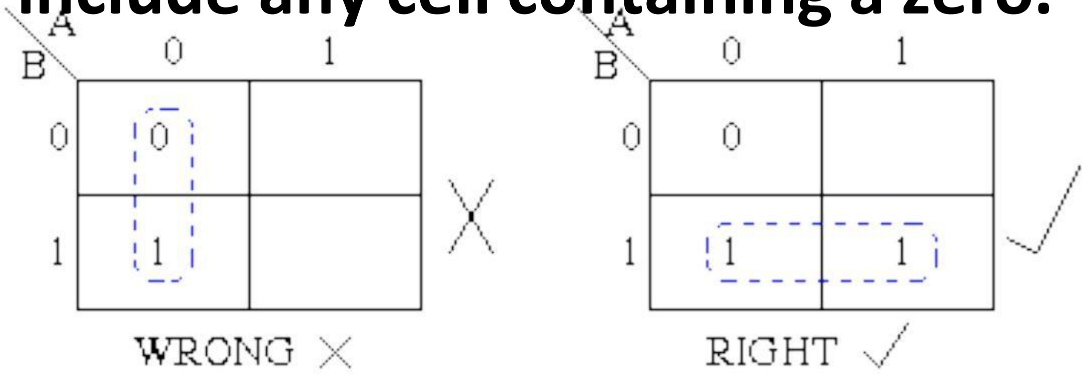
	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	1	1
$\bar{A}B$	0	0	0	1
AB	1	1	0	1
$A\bar{B}$	1	1	0	1

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	1	1
$\bar{A}B$	0	0	0	1
AB	1	1	0	1
$A\bar{B}$	1	1	0	1

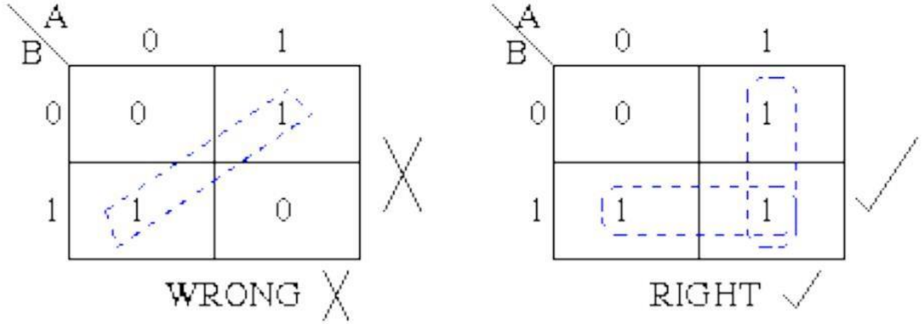
$$Y = \bar{A}\bar{B}D + A\bar{C} + C\bar{D}$$

KARNAUGH SIMPLIFICATIONS-2

- **Groups may not include any cell containing a zero.**

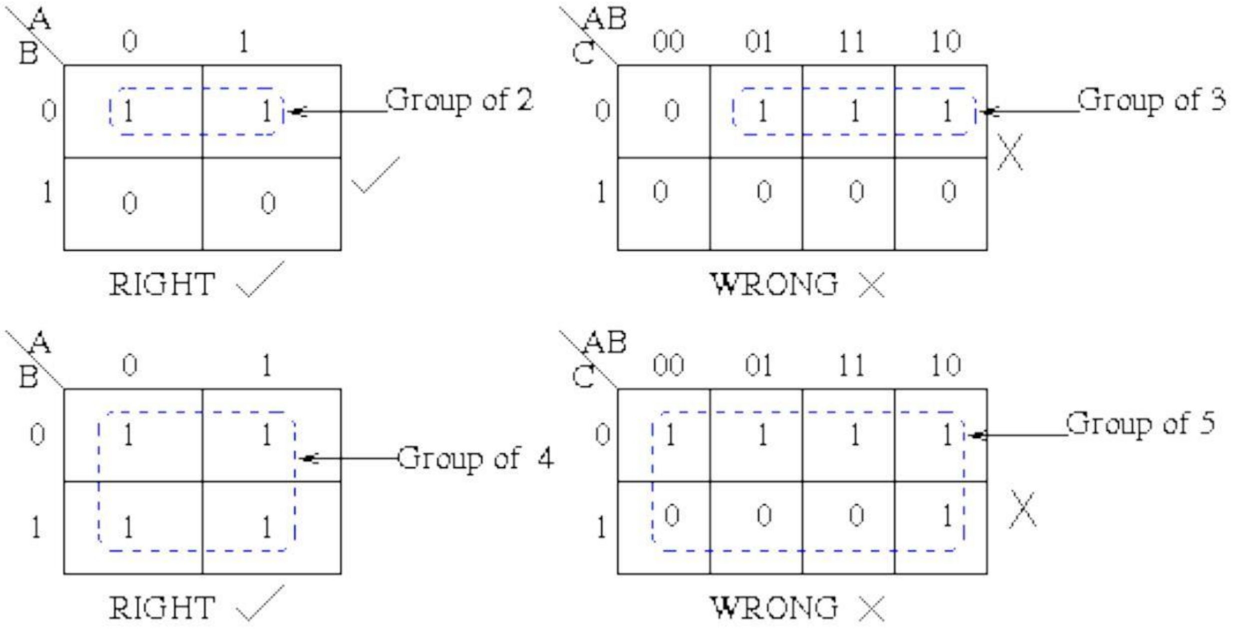


- **Groups may be horizontal or vertical, but not diagonal.**



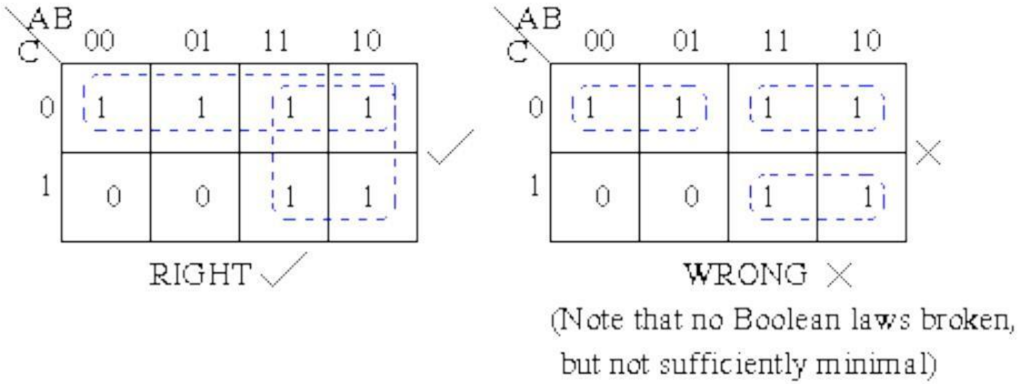
KARNAUGH SIMPLIFICATIONS-3

- **Groups must contain 1, 2, 4, 8, or in general 2^n cells.**
- **That is if n = 1, a group will contain two 1's since 2^1=2.**
- **If n = 2, a group will contain four 1's since 2^2 = 4.**

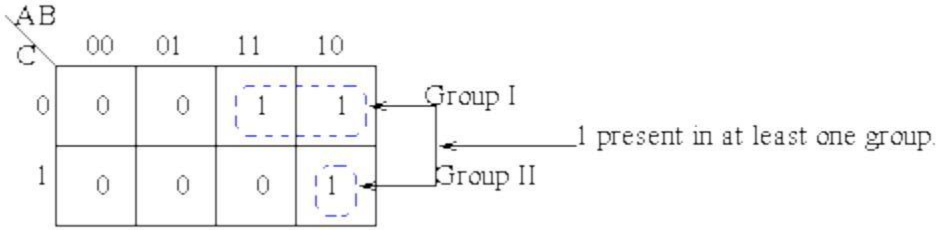


KARNAUGH SIMPLIFICATIONS-4

- **Each group should be as large as possible.**

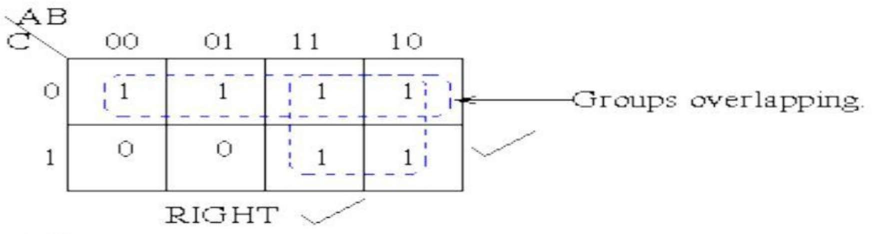


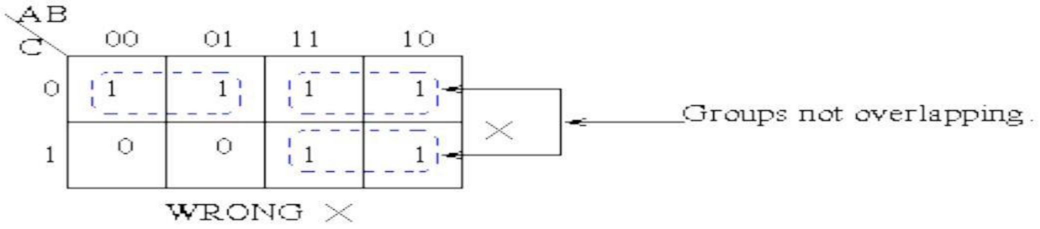
- **Each cell containing a *one* must be in at least one group.**



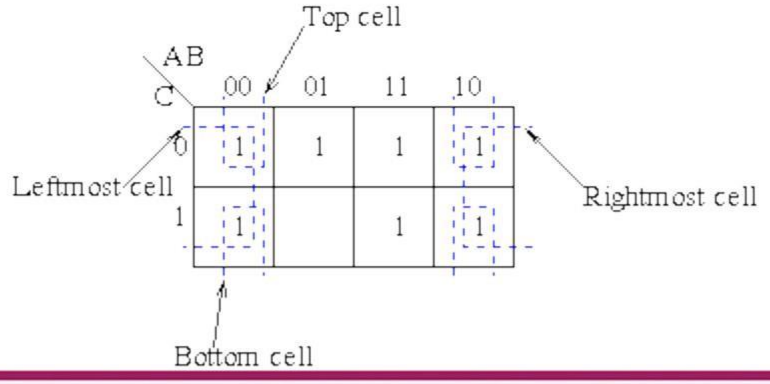
KARNAUGH SIMPLIFICATIONS-5

- **Groups may overlap.**





- **Groups may wrap (Rolling the Map) around the table. The leftmost cell in a row may be grouped with the rightmost cell and the top cell in a column may be grouped with the bottom cell.**



KARNAUGH SIMPLIFICATIONS-6

- **Rolling the Map**

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	1	0	0	1
AB	1	0	0	1
$A\bar{B}$	0	0	0	0

(a)

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	1	0	0	1
AB	1	0	0	1
$A\bar{B}$	0	0	0	0

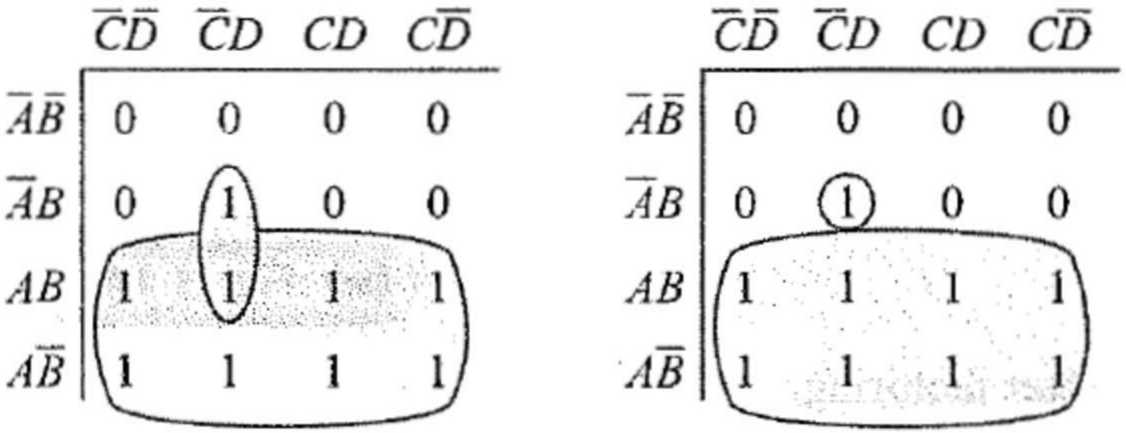
(b)

$$Y = B\bar{C}\bar{D} + BC\bar{D}$$

$$Y = B\bar{D}$$

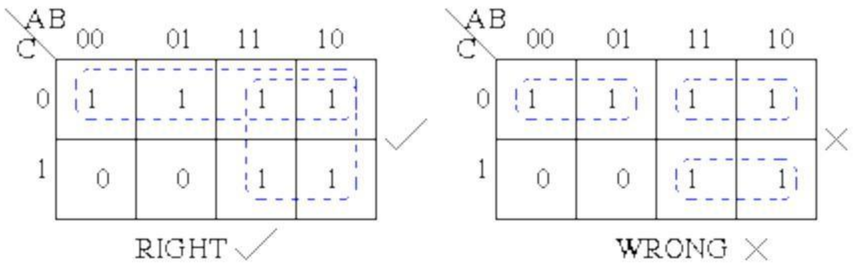
KARNAUGH SIMPLIFICATIONS-7

- **Always overlap groups if possible**
- Use the 1s more than once to get the largest groups you can.



KARNAUGH SIMPLIFICATIONS-8

- **There should be as few groups as possible, as long as this does not contradict any of the previous rules.**



KARNAUGH SIMPLIFICATIONS-11

1. No zeros allowed (Only in SOP).
2. No diagonals.
3. Only power of 2 number of cells in each group.
4. Groups should be as large as possible.
5. Every one must be in at least one group.
6. Overlapping allowed.
7. Wrap around allowed.
8. If possible roll and overlap to get the largest groups you can find.
9. Fewest number of groups possible.

K- map grouping Examples-1

- **Rolling and overlapping**

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	0	0
$\bar{A}B$	1	1	0	1
AB	1	1	0	1
$A\bar{B}$	1	1	0	0

$$Y = \bar{C} + BC\bar{D}$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	0	0
$\bar{A}B$	1	1	0	1
AB	1	1	0	1
$A\bar{B}$	1	1	0	0

$$Y = \bar{C} + B\bar{D}$$

K- map grouping Examples-2

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	0	1
$\bar{A}B$	1	1	0	1
AB	1	1	0	0
$A\bar{B}$	1	1	0	1

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

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	0	1
$\bar{A}B$	1	1	0	1
AB	1	1	0	0
$A\bar{B}$	1	1	0	1

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

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	0	1
$\bar{A}B$	1	1	0	1
AB	1	1	0	0
$A\bar{B}$	1	1	0	1

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

Eliminating Redundant Groups-2

- A group whose 1s are already used by other groups.

	$\overline{C}\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\overline{B}$	0	0	1	0
$\overline{A}B$	1	1	1	0
AB	0	1	1	1
$A\overline{B}$	0	1	0	0

Eliminating Redundant Groups-2

- A group whose 1s are already used by other groups.

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

Eliminating Redundant Groups-2

- A group whose 1s are already used by other groups.

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	1	0
$\bar{A}B$	1	1	1	0
AB	0	1	1	1
$A\bar{B}$	0	1	0	0

- All the 1 s of the quad are used by the pairs. Because of this, the quad is redundant and can be eliminated to get

Eliminating Redundant Groups-2

- A group whose 1s are already used by other groups.

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	1	0
$\bar{A}B$	1	1	1	0
AB	0	1	1	1
$A\bar{B}$	0	1	0	0

summary of the Karnaugh-map method for simplifying Boolean equations

1. Enter a 1 on the Karnaugh map for each fundamental product that produces a 1 output in the truth table. Enter 0s elsewhere.
2. Encircle the octets, quads, and pairs. Remember to roll and overlap to get the largest groups possible.
3. If any isolated 1s remain, encircle each.
4. Eliminate any redundant group.
5. Write the Boolean equation by ORing the products corresponding to the encircled groups.

Example-1

- What is the simplified Boolean equation for the following logic equation expressed by minterms? $Y=F(A,B,C,D)=\sum m(7,9,$ 10, 11, 12, 13, 14, 15)

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	1	0
AB	1	1	1	1
$A\bar{B}$	0	1	1	1

Example-1

- What is the simplified Boolean equation for the following logic equation expressed by minterms? *Y=F(A,B,C,D)=∑m(7,9,* 10, 11, 12, 13, 14, 15)

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	1	0
AB	1	1	1	1
$A\bar{B}$	0	1	1	1

Example-1

- What is the simplified Boolean equation for the following logic equation expressed by minterms? *$Y=F(A,B,C,D)=\sum m(7,9,$* 10, 11, 12, 13, 14, 15)

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	1	0
AB	1	1	1	1
$A\bar{B}$	0	1	1	1

Example-1

- What is the simplified Boolean equation for the following logic equation expressed by minterms? *Y=F(A,B,C,D)=∑m(7,9,* 10, 11, 12, 13, 14, 15)

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	1	0
AB	1	1	1	1
$A\bar{B}$	0	1	1	1

Example-1

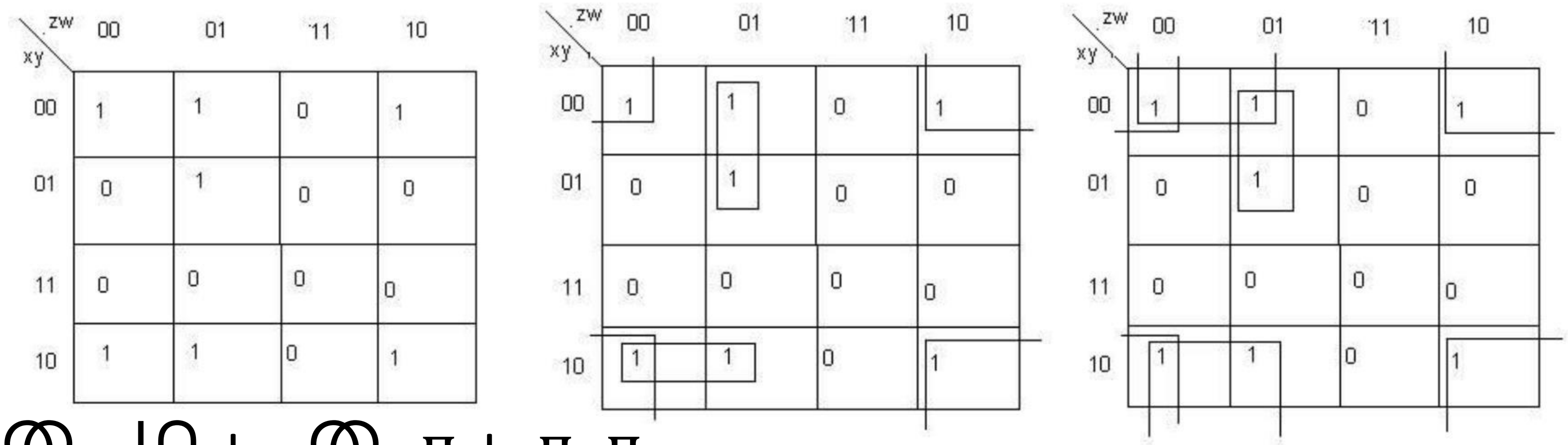
- What is the simplified Boolean equation for the following logic equation expressed by minterms? $Y=F(A,B,C,D)=\sum m(7,9,$ 10, 11, 12, 13, 14, 15)

	$\overline{C}\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\overline{B}$	0	0	0	0
$\overline{A}B$	0	0	1	0
AB	1	1	1	1
$A\overline{B}$	0	1	1	1

$$Y = AB + AC + AD + BCD$$

Example-2

- **Simplify the following Boolean function in sum of products form (SOP) F(x, y, z, w) = ∑m(0, 1, 2, 5, 8, 9, 10)**



- F= ത ഥ + ത ђ+ ђ ђ

DON'T-CARE CONDITIONS-3

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	0	0
AB	×	×	×	×
$A\bar{B}$	0	1	×	×

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	0	0
AB	×	×	×	×
$A\bar{B}$	0	1	×	×

$$Y = AD$$

DON'T-CARE CONDITIONS-4

- Ideas about don't-care conditions:

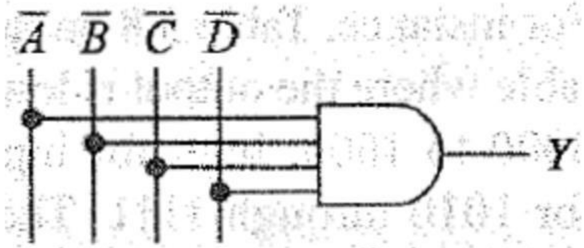
1. Given the truth table, draw a Karnaugh map with 0s, 1s, and don't-cares (X).
2. Encircle the actual 1s on the Karnaugh map in the largest groups you can find by treating the don't cares as 1s.
3. After the actual 1s have been included in groups, disregard the remaining don't cares by visualizing them as 0s.

DON'T-CARE CONDITIONS-5

- What is the simplest logic circuit for the following truth table?

A	B	C	D	Y
0	0	0	0	1
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	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

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	0	0	0
$\bar{A}B$	0	0	0	0
AB	×	×	×	×
$A\bar{B}$	0	0	×	×

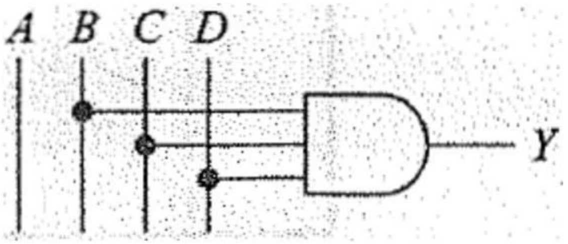


DON'T-CARE CONDITIONS-6

- Give the simplest logic circuit for following logic equation where *d* represents don't-care condition for following locations.

  $F(A, B, C, D) = \sum m(7) + d(10, 11, 12, 13, 14, 15)$

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

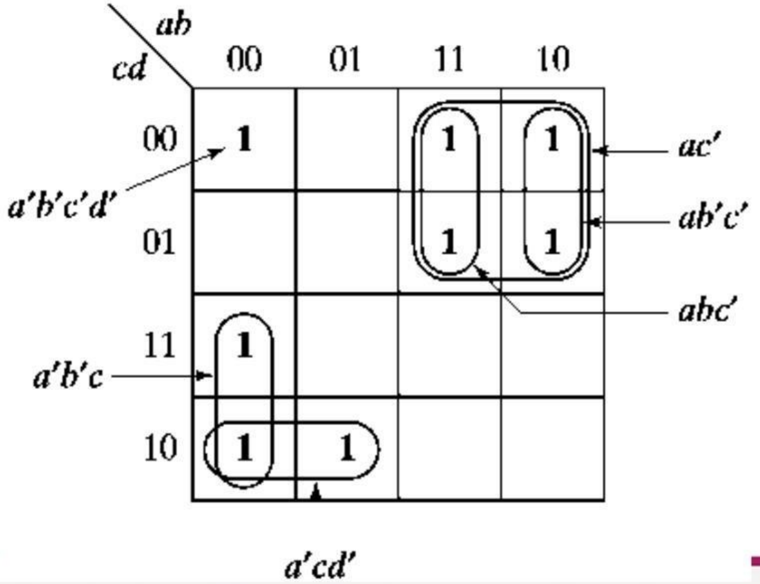


Determination of Minimum Expressions Using Essential Prime Implicants

- 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.**
- A product term implicant is called a **prime implicant** if it cannot be combined with another term to eliminate a variable.

Determination of Minimum Expressions Using Essential Prime Implicants

- In Figure, a’b’c, a’cd’, and ac’ are prime implicants because they cannot be combined with other terms to eliminate a variable.
- a’b’c’d’ is not a prime implicant because it can be combined with a’b’cd’ or ab’c’d’.
- Neither abc’, nor ab’c’ is a prime implicant because these terms can be combined together to form ac’.



Determination of Minimum Expressions Using Essential Prime Implicants

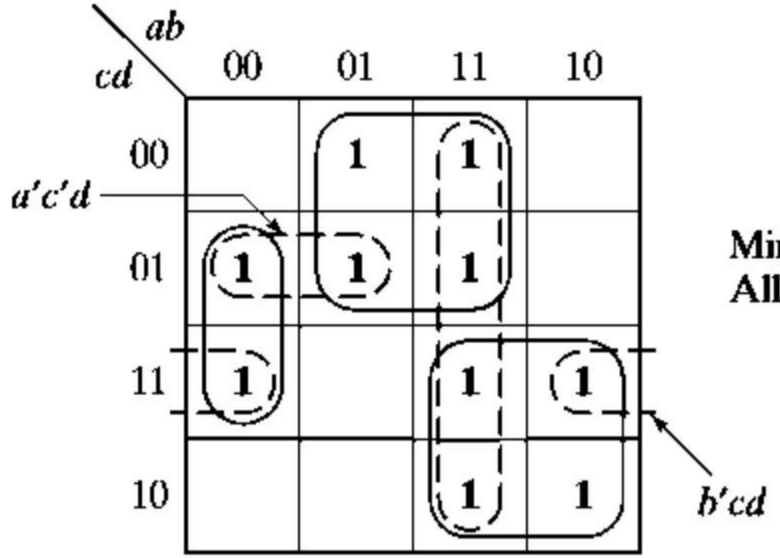
- All of the prime implicants of a function can be obtained from a Karnaugh map.
  - A single 1 on a map represents a prime implicant if it is not adjacent to any other 1's.
  - Two adjacent 1's on a map form a prime implicant if they are not contained in a group of four 1's;
  - four adjacent 1's form a prime implicant if they are not contained in a group of eight 1's, etc.

Determination of Minimum Expressions Using Essential Prime Implicants

- The minimum sum-of-products expression for a function consists of some (but not necessarily all) of the prime implicants of a function.
- In other words, a sum-of-products expression containing a term which is not a prime implicant cannot be minimum.
- This is true because if a **nonprime term** were present, the expression could be simplified by combining the nonprime term with additional minterms.

Determination of Minimum Expressions Using Essential Prime Implicants

- The function plotted in Figure has six prime implicants.
  - Three of these prime implicants cover all of the 1’s on the map, and the minimum solution is the sum of these three prime implicants.
  - The shaded loops represent prime implicants which are not part of the minimum solution.



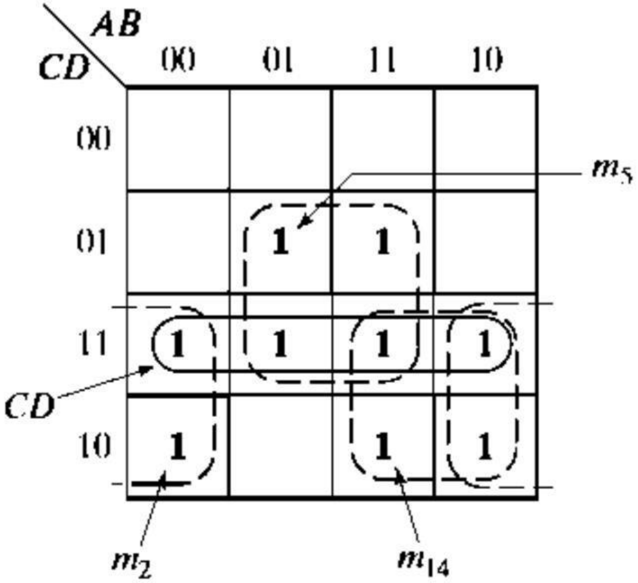
Minimum solution: $F = a'b'd + bc' + ac$
All prime implicants: $a'b'd$, bc', ac, $a'c'd$, ab, $b'cd$

Determination of Minimum Expressions Using Essential Prime Implicants

- All of the prime implicants of a function are generally not needed in forming the minimum sum of products, a systematic procedure for selecting prime implicants is needed.
- If prime implicants are selected from the map in the wrong order, a nonminimum solution may result.

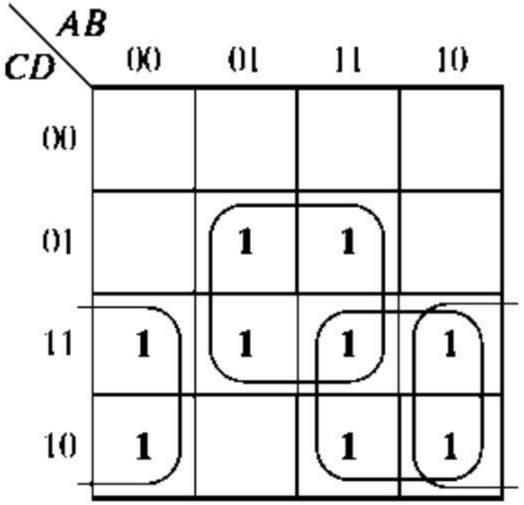
Determination of Minimum Expressions Using Essential Prime Implicants

- For example, in Figure (a), if CD is chosen first, then BD, B'C, and AC are needed to cover the remaining 1's, and the solution contains four terms.
- If the prime implicants indicated in Figure (b) are chosen first, all 1's are covered and CD is not needed.



$f = CD + BD + B'C + AC$

(a)

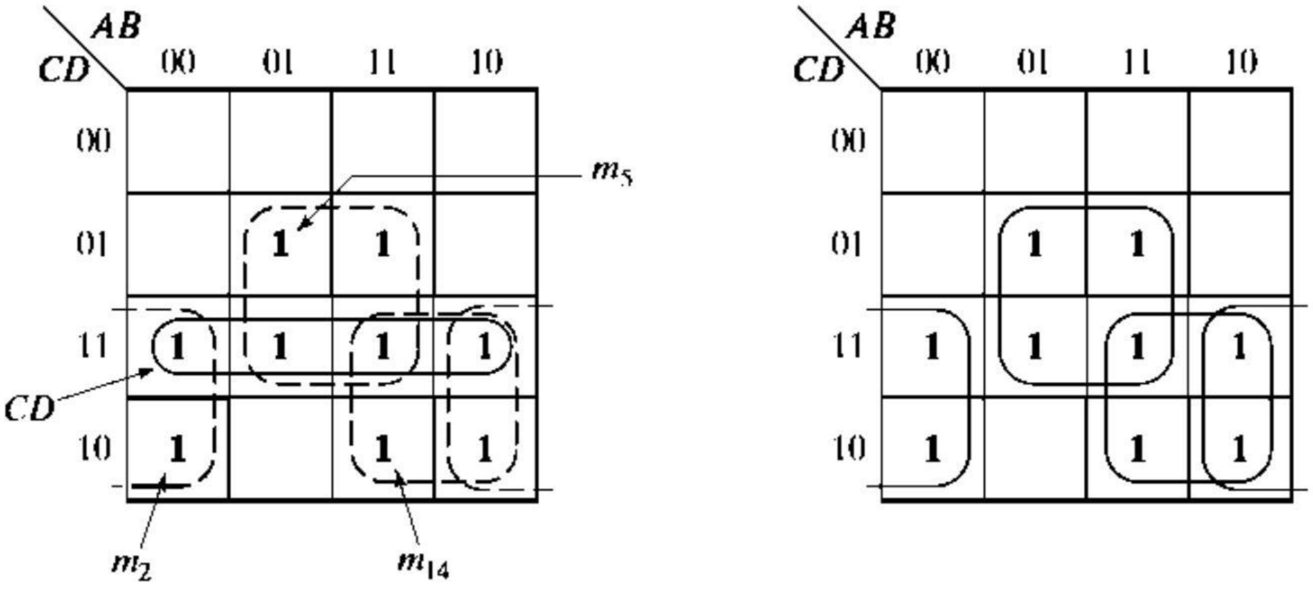


$f = BD + B'C + AC$

(b)

Determination of Minimum Expressions Using Essential Prime Implicants

- Note that some of the minterms on the map of Figure (a) can be covered by only a single prime implicant, but other minterms can be covered by two different prime implicants.
- For example, m2 is covered only by B'C, but m5 is covered by both B'C and CD.



$f = CD + BD + B'C + AC$

(a)

$f = BD + B'C + AC$

(b)

Determination of Minimum Expressions Using Essential Prime Implicants

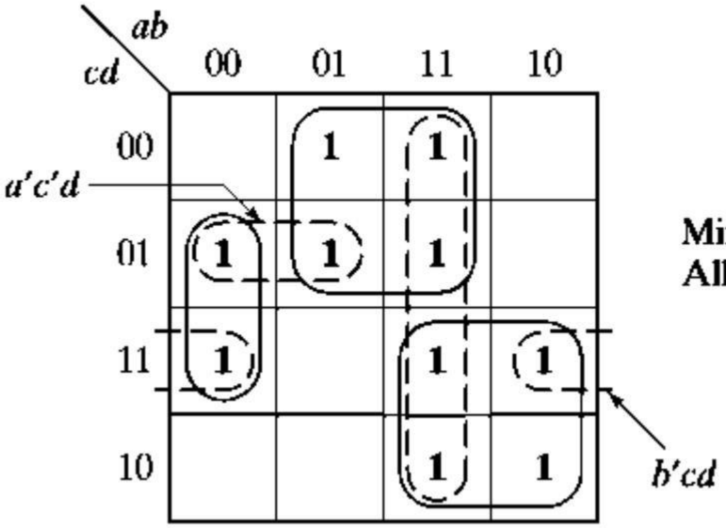
- If a minterm is covered by only one prime implicant, that prime implicant is said to be **essential prime imlicant**, and **it must be included in the minimum sum of products**.
  - B'C is an **essential prime implicant** because m2 is not covered by any other prime implicant.
  - CD is **not essential prime implicant** because each of the 1's in CD can be covered by another prime implicant.
  - The only prime implicant which covers m5 is BD, so BD is essential.
  - AC is essential because no other prime implicant covers m14.

Determination of Minimum Expressions Using Essential Prime Implicants

- In general, in order to find a minimum sum of products from a map, we should first loop all of the essential prime implicants.
- One way of finding essential prime implicants on a map is simply to look at each 1 on the map that has not already been covered, and check to see how many prime implicants cover that 1.
- If there is only one prime implicant which covers the 1, that prime implicant is essential.
- If there are two or more prime implicants which cover the 1, we cannot say whether these prime implicants are essential or not without checking the other minterms.

Determination of Minimum Expressions Using Essential Prime Implicants

- For example, in Figure, m4 is covered only by the prime implicant bc′, and m10 is covered only by the prime implicant ac.
- All other 1's on the map are covered by two prime implicants; therefore, the only essential prime implicants are bc′ and ac.

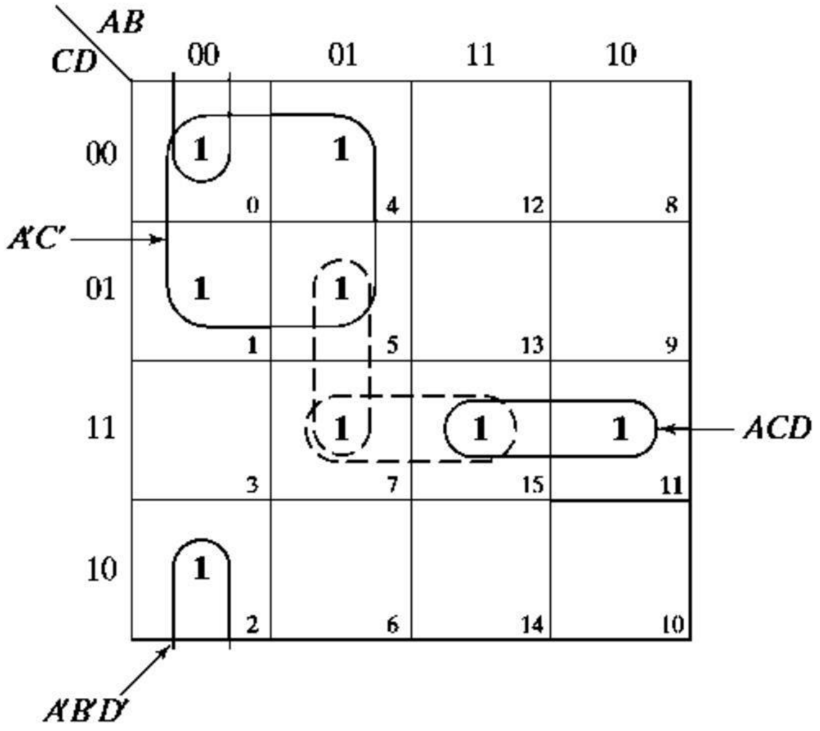


Minimum solution: $F = a'b'd + bc' + ac$
All prime implicants: $a'b'd$, bc', ac, $a'c'd$, ab, $b'cd$

Determination of Minimum Expressions Using Essential Prime Implicants

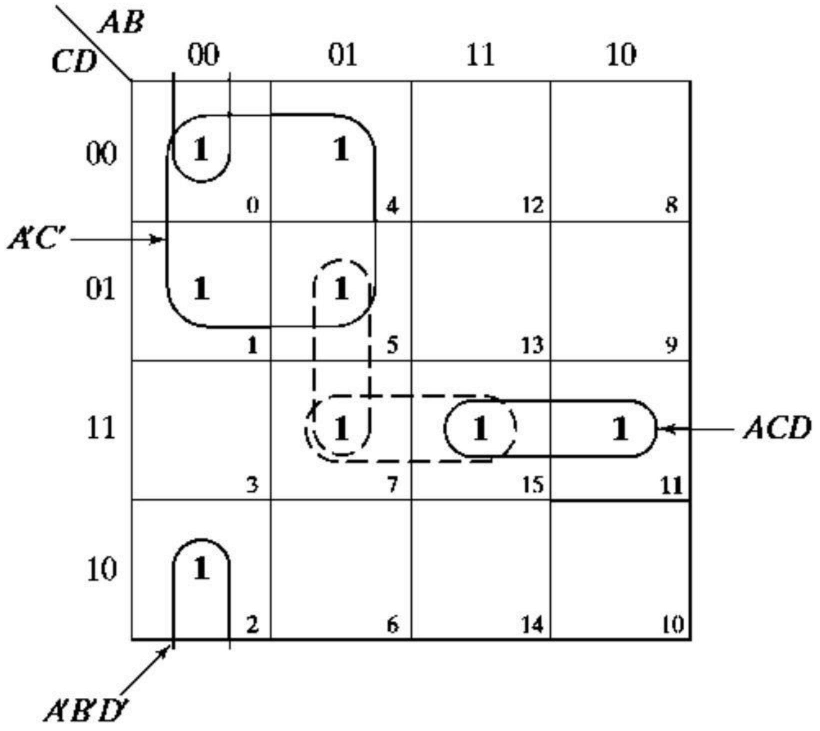
- If the given minterm and all of the 1's adjacent to it are covered by a single term, then that term is an essential prime implicant.
- If all of the 1's adjacent to a given minterm are not covered by a single term, then there are two or more prime implicants which cover that minterm, and we cannot say whether these prime implicants are essential or not without checking the other minterms.

Determination of Minimum Expressions Using Essential Prime Implicants



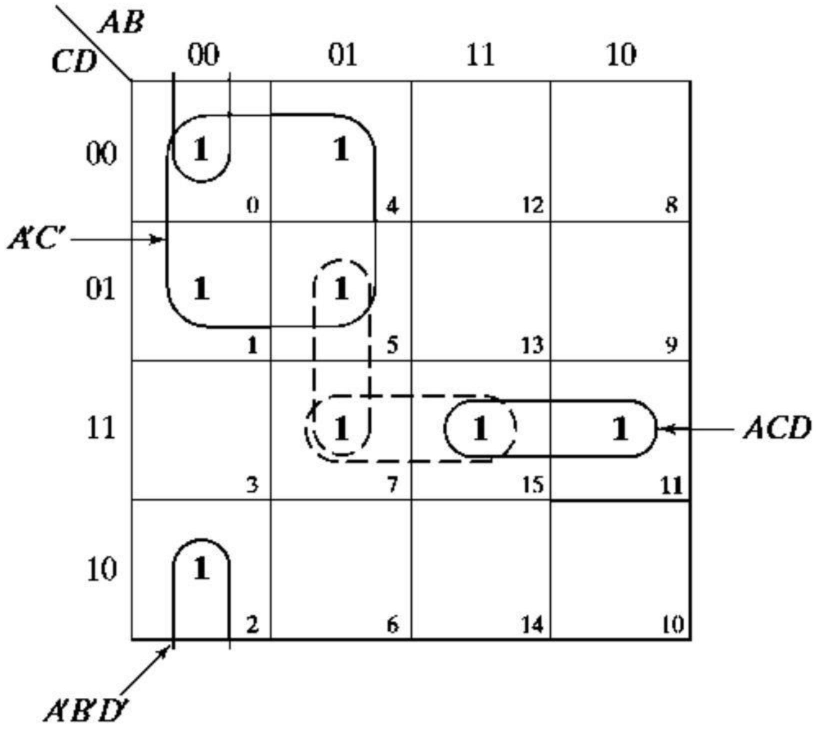
- The adjacent 1's for minterm m0 (1_0) are 1_1, 1_2, and l_4.
- Because no single term covers these four 1's, no essential prime implicant is yet apparent.
- The adjacent 1's for 1_1 are 1_0 and 1_5, so the term which covers these three 1's **(A'C')** is an essential prime implicant.

Determination of Minimum Expressions Using Essential Prime Implicants



- The only 1 adjacent to 1_2 is 1_0, **A’B’D’** is also essential.
- The only 1 adjacent to 1_{11} is 1_{15}, ACD is essential.

Determination of Minimum Expressions Using Essential Prime Implicants



- The 1's adjacent to 1_7 (1_5 and 1_{15}) are not covered by a single term, neither A'BD nor BCD is essential.
- To complete the minimum solution, one of the nonessential prime implicants is needed. So either A'BD or BCD may be selected
- The final solution is

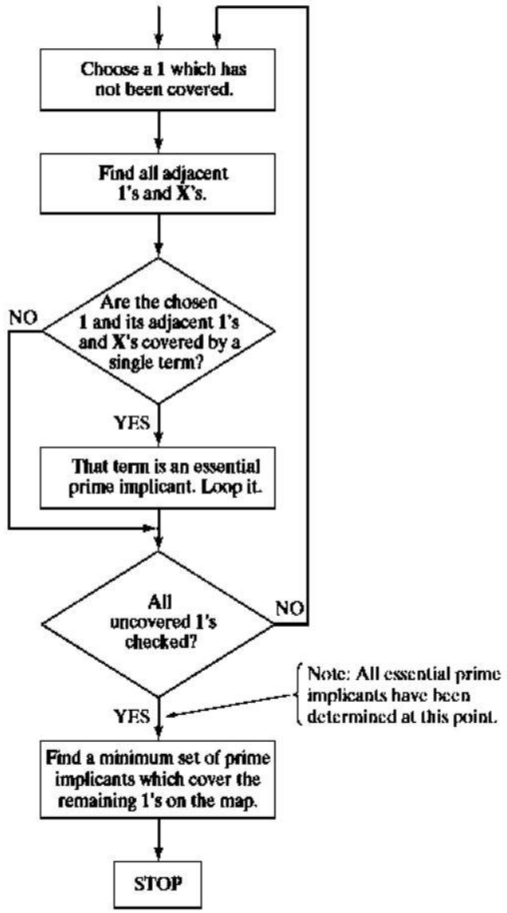
$$A'C' + A'B'D' + ACD + \begin{Bmatrix} A'BD \\ \text{or} \\ BCD \end{Bmatrix}$$

Determination of Minimum Expressions Using Essential Prime Implicants

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

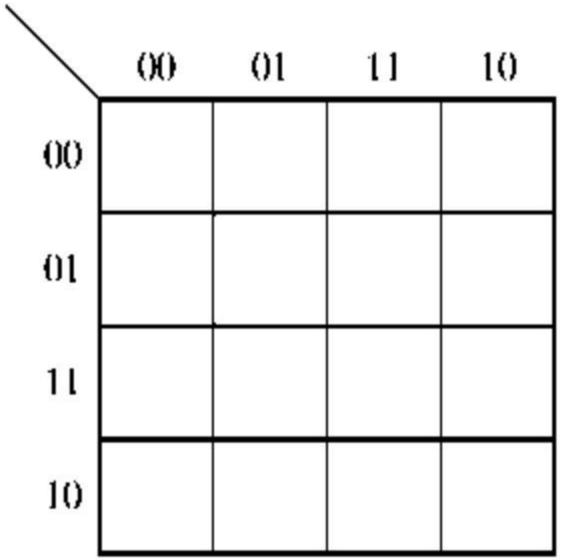
1. Choose a minterm (a 1) which has not yet been covered.
2. Find all 1’s and X’s adjacent to that minterm. (Check the n adjacent squares on an n-variable map.)
3. If a single term covers the minterm and all of the adjacent 1’s and X’s, then that term is an essential prime implicant, so select that term. (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.)

Determination of Minimum Expressions Using Essential Prime Implicants



Programmed Exercise

- Determine the minimum sum of products and minimum product of sums for
- $f = b'c'd' + bcd + acd' + a'b'c + a'bc'd$
- First, plot the map for f.



Programmed Exercise

- For problem on K-Map refer text book and VTU question papers.
- What is the simplified Boolean equation for the following logic equation expressed by minterms?
  - *Y=F(A,B,C,D)=∑m(7,9,* 10, 11, 12, 13, 14, 15)
  - F(w, x, y, z) = ∑m(0, 1, 2, 5, 8, 9, 10)
  - Y= ∑m(1, 2, 6, 7)
  - *Y=F(p,q,r,s)=∑m(1, 3, 4, 10, 12,* 13, 15)
  - *F* (*w*, *x*, *y*, *z*) = *∑m* (0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)

$$F = y' + w'z' + xz'$$

  - F (A, B, C, D) = *∑m*(0,1, 2, 5, 8, 9,10)

$$B'D' + B'C' + A'C'D$$

Programmed Exercise – Vtu Questions

- Find the minimal SOP of the following Boolean functions using K-Map:
  - $F(a,b,c,d)=\sum m(6,7,9,10,13)+d(1,4,5,11)$
  - $F(p,q,r,s)=\sum m(6,7,9,10,13)+d(0,1,8,12)$
  - $F(A,B,C,D)= \sum m(6,8,9,10,11,12,13,14,15)$
  - $F(A,B,C,D)=\sum m(1,3,5,7,8,10,12,14)$
  - $F(a,b,c,d)=\sum m(5,6,7,12,13)+d(4,9,14,15)$

Programmed Exercise

- Find the minimum sum of products for each function using a Karnaugh map.
- (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) $f4(x, y, z) = M0 \cdot M5$

Programmed Exercise

- Find the minimum sum-of-products expression for each function. Underline the essential prime implicants in your answer and tell which minterm makes each one essential.

1. $f(a, b, c, d) = \Sigma\, m(0, 1, 3, 5, 6, 7, 11, 12, 14)$
2. $f(a, b, c, d) = \Pi\, M(1, 9, 11, 12, 14)$
3. $f(a, b, c, d) = \Pi\, M(5, 7, 13, 14, 15) \cdot \Pi\, D(1, 2, 3, 9)$
4. $f(a, b, c, d) = \Sigma\, m(0, 2, 3, 4, 7, 8, 14)$
5. $f(a, b, c, d) = \Sigma\, m(1, 2, 4, 15) + \Sigma\, d(0, 3, 14)$
6. $f(a, b, c, d) = \Pi\, M(1, 2, 3, 4, 9, 15)$
7. $f(a, b, c, d) = \Pi\, M(0, 2, 4, 6, 8) \cdot \Pi\, D(1, 12, 9, 15)$

Programmed Exercise

- Find the minimum sum-of-products expression for each function. Underline the essential prime implicants in your answer and tell which minterm makes each one essential.
  1. $f(a, b, c, d) = \Sigma m\ (0, 1, 3, 5, 6, 7, 11, 12, 14)$
  2. $f(a, b, c, d) = \Pi M\ (1, 9, 11, 12, 14)$
  3. $f(a, b, c, d) = \Pi M\ (5, 7, 13, 14, 15) \cdot \Pi D\ (1, 2, 3, 9)$

Simplification Using Map-Entered Variables

- One of the input variable is placed inside Karnaugh map
- This reduces the Karnaugh map size by one degree.
  - i.e. a three variable problem that requires 23 = 8 locations in Karnaugh map will require 2(3-l) = 4 locations in entered variable map.
- This technique is particularly useful for mapping problems with more than four input variables.

Simplification Using Map-Entered Variables

- Example
- Example: Y = F(A, B, C) = ∑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

Truth Table

A	B	Y
0	0	0
0	0	
0	1	(0) c
0	1	
1	0	0
1	0	
1	1	1
1	1	

EVM Table

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

K-MAP for EVM Table

Simplification Using Map-Entered Variables

- Let's choose C as map entered variable and see how output Y varies with C for different combinations of other two variables A and B.
- For AB= 00 we find Y = 0 and is not dependent on C.
  - For AB= 01 we find Y is complement of C thus we can write Y = C.
- Similarly, for AB= 10, Y = 0 and for AB= 11,Y= 1.

Simplification Using Map-Entered Variables

- If choose A as map entered variable write the corresponding entered variable map.

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

	$\bar{C}$	C
$\bar{B}$	0	0
B	1	A

- If choose B as map entered variable write the corresponding entered variable map.

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

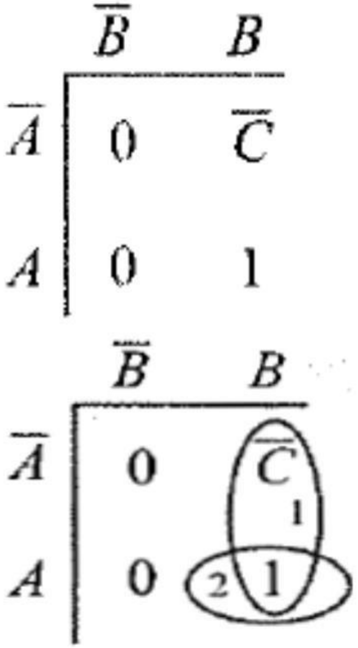
Simplification Using Map-Entered Variables

- This is similar to Karnaugh map method.

A	B	Y
0	0	0
0	0	
0	1	ŋ
0	1	
1	0	0
1	0	
1	1	1
1	1	

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

Y=BC+AB

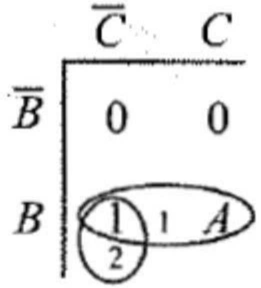


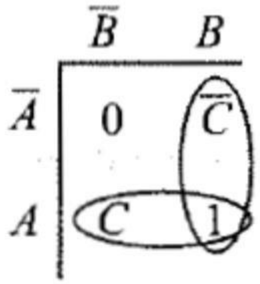
- Note that C is grouped with 1 to get a larger group as 1 can be written as 1

=1+C.

Simplification Using Map-Entered Variables

- The product term representing each group is obtained by including map entered variable in the group as an additional ANDed term.





$Y=BC^{(0)}+AB$

$Y=AC+BC^{(0)}$

Simplification Using Map-Entered Variables

- By using map-entered variables, Karnaugh map techniques can be extended to simplify functions with more than four or five variables.
- Below shows a four-variable map with two additional variables entered in the squares in the map

G(A, B, C, D, E, F) = m0 + m2 + m3 + Em5 + Em7 + Fm9 + m11 + m15 (+ don’t-care terms)

CD \ AB	00	01	11	10
00	1			
01	X	*E*	X	*F*
11	1	*E*	1	1
10	1			X

G

Simplification Using Map-Entered Variables

CD \ AB	00	01	11	10
00	1			
01	X	*E*	X	*F*
11	1	*E*	1	1
10	1			X

G

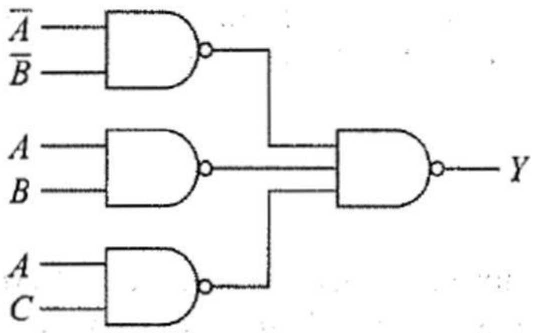
PRODUCT-Of-SUMS SIMPLIFICATION-1

- **SOP Simplification**

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

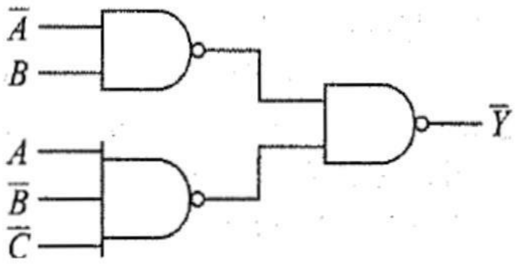
	$\overline{C}\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\overline{B}$	1	1	1	1
$\overline{A}B$	0	0	0	0
AB	1	1	1	1
$A\overline{B}$	0	0	1	1

$$Y = \overline{A}\overline{B} + AB + AC$$



- **Complementary Circuit**

	$\overline{C}\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\overline{B}$	0	0	0	0
$\overline{A}B$	1	1	1	1
AB	0	0	0	0
$A\overline{B}$	1	1	0	0



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

PRODUCT-Of-SUMS SIMPLIFICATION-2

- **Using Karnaugh map by grouping zeros**

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	0	1
AB	1	1	1	1
$A\bar{B}$	1	1	1	1

$$Y = (A + B)(A + C)(A + \overline{D}')$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	1	0
$\bar{A}B$	0	0	1	1
AB	×	×	×	1
$A\bar{B}$	×	×	×	0

$$Y = C(B + D)$$

Example-1

- Give POS form of $Y = F(A, B, C, D) = \prod M(0, 3, 4, 5, 6, 7, 11, 15)$

AB \ CD	00	01	11	10
00	0_{0}	1_{1}	0_{3}	1_{2}
01	0_{4}	0_{5}	0_{7}	0_{6}
11	1_{12}	1_{13}	0_{15}	1_{14}
10	1_{8}	1_{9}	0_{11}	1_{10}

Exam Questions

- Simplify the following using K-Map

$$Y = F(A, B, C) = \Pi M(0, 3, 6) = \Sigma m(1, 2, 4, 5, 7)$$

- By Grouping 1’s

A \ BC	00	01	11	10
0	0_{0}	1_{1}	0_{3}	1_{2}
1	1_{4}	1_{5}	1_{7}	0_{6}

By Grouping 0’s

A \ BC	00	01	11	10
0	0_{0}	1_{1}	0_{3}	1_{2}
1	1_{4}	1_{5}	1_{7}	0_{6}

- = + +AC+AB
-

= (+ +)(+ +)(+ +)

QUINE-McCLUSKY METHOD

- **Tabular Method of Minimisation**
- Karnaugh map method is very simple and intuitively appealing is somewhat subjective.
- It depends on the user's ability to identify patterns that gives largest size.
- Also the method becomes difficult to adapt for simplification of 5 or more variables.
- Quine-McClusky method involves preparation of two tables; one **determines *prime implicants*** and the other selects ***essential prime implicants*** to get minimal expression.

QUINE-McCLUSKY METHOD

- The Quine-McCluskey method reduces the minterm expansion (standard sumof-products form) of a function to obtain a minimum sum of products. The procedure consists of two main steps:
  1. Eliminate as many literals as possible from each term by systematically applying the theorem XY + XY’ = X. The resulting terms are called prime implicants.
  2. Use a prime implicant chart to select a minimum set of prime implicants which, when ORed together, are equal to the function being simplified and which contain a minimum number of literals.

QUINE-McCLUSKY METHOD

- Literal: Each appearance of a variable, either uncomplemented or complemented, is called a *literal*.
- Implicant: A product term that indicates the input valuation(s) for which a given function is equal to 1 is called an *implicant* of the function.
- **Prime Implicant**
- An implicant is called a *prime implicant* if it cannot be combined into another implicant that has fewer literals. Another way it is impossible to delete any literal in a prime implicant and still have a valid implicant.
- Prime implicants are expressions with least number of literals that represents all the terms given in a truth table. Prime implicants are examined to get essential prime implicants for a particular expression that avoids any type of duplication.

QUINE-McCLUSKY METHOD

Determination of Prime Implicants

- The function must be given as a sum of minterms. (If the function is not in maxterm form, convert to the minterm expansion)
- In the first part of the QuineMcCluskey method, all of the prime implicants of a function are systematically formed by combining minterms.

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

QUINE-McCLUSKY METHOD-4 Determination of Prime Implicants

- Stage-1, 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 put minterns in different **groups depending on how many 1s input variable combinations** *(ABCD)* have.
  - For example, first group has no l in input combination, second group has only one 1, third two 1 s, fourth three ls and fifth four 1s.

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

QUINE-McCLUSKY METHOD-5
Determination of Prime Implicants

- In Stage 2, we first try to combine first and second group of Stage1, on a member to member basis.
- The rule is to see if only one binary digit is differing between two members and we mark that position by '−'. This means corresponding variable is not required to represent those members.
- Thus (0) of first group combines with (1) of second group to form (0,1) in Stage 2 and can be represented by

  $A'B'C'$ (0 0 0 −).
- Proceed in the same manner to find rest of the combinations in successive groups of Stage 1 and table them as in figure.
- Note that, we need not look beyond successive groups to find such combinations as groups that are not adjacent, differ by more than one binary digit. Also note that each combination of Stage 2 can be represented by three literals.
- **All the members of particular stage, which finds itself in at least one combination of next stage are tick (√) (checked off) marked. This is followed for Stage 1 terms as well as terms of other stages.**

QUINE-McCLUSKY METHOD-5 Determination of Prime Implicants

- Two terms can be combined if they differ in exactly one variable.
- Comparison of terms in nonadjacent groups is unnecessary because such terms will always differ in at least two variables and cannot be combined using XY + XY′ = X.
- Similarly, the comparison of terms within a group is unnecessary because two terms with the same number of 1's must differ in at least two variables.
- Thus, only terms in adjacent groups must be compared.
- **Each time a term is combined with another term, it is checked off.**

QUINE-McCLUSKY METHOD-5
Determination of Prime Implicants

- A term may be used more than once because X + X = X.
- Even though two terms have already been combined with other terms, they still must be compared and combined if possible.
- This is necessary because the resultant term may be needed to form the minimum sum solution.
- **At this stage, we may generate redundant terms, but these redundant terms will be eliminated later.**

QUINE-McCLUSKY METHOD-6

Determination of Prime Implicants

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

QUINE-McCLUSKY METHOD Determination of Prime Implicants

- In Stage 3, we combine members of different groups of Stage 2 in a similar way. Now it will have two '–' elements in each combination. This means each combination requires two literals to represent it.
- For example (0,1,2,3) is represented by $A'B'$ (0 0 – –).
- There are three other groups in Stage 3; (2,10,3,11) represented by $B'C$, (10,14,11,15) by AC and (12,13,14,15) by AB.
- Note that, (0,2,1,3), (10,11,14,15) and (12,14,13,15) get represented by $A'B$, AC and AB respectively and do not give any new term.

QUINE-McCLUSKY METHOD
Determination of Prime Implicants

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

QUINE-McCLUSKY METHOD
Determination of Prime Implicants

- If any duplicate terms are formed in each case by combining the same set of minterms in a different order.
  - **These duplicate term must be deleting,**
- Now compare terms from the two groups.
- If no further combination is possible, the process terminates.
- There is no Stage 4 for this problem as no two members of Stage 3 has only one digit changing among them. This completes the process of determination of prime implicants.
- **The rule is all the terms that are not ticked (checked off) at any stage is treated as prime implicants.**

QUINE-McCLUSKY METHOD
Determination of 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.
- Each term has a minimum number of literals, but the number of terms is not minimum. So method of eliminating redundant prime implicants using a **prime implicant chart**.

- Given a function F of n variables, a product term P is an **implicant** of F iff 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.

QUINE-McCLUSKY METHOD
The Prime Implicant Chart

- Selection of Prime Implicants
- Once we are able to determine prime implicants that covers all the terms of a truth table we try to select essential prime implicants and remove redundancy or duplication among them.
- To find a minimum expression we construct a *prime implicant table* in which there is a row for each prime implicant and a column for each minterm that must be covered.
- **Note- don't care values are not included.**
- Then we place check marks **(**√**) or (**X**)** to indicate the minterms covered by each prime implicant.

QUINE-McCLUSKY METHOD-11

- **Selection of essential prime implicants from this table is done in the following way.**
  1. 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.
  2. Essential prime implicants are easy to find using the prime implicant chart. If a given column contains only one √ or X, then the corresponding row is an essential prime implicant. (Encircle the √ or X)
  3. Each time a prime implicant is selected for inclusion in the minimum sum, the corresponding row should be crossed out.
  4. After doing this, the columns which correspond to all minterms covered by that prime implicant should also be crossed out.
     - The essential prime implicants and the corresponding rows and columns are crossed out.
  5. A minimum set of prime implicants must now be chosen to cover the remaining columns.
     - Find minimum set of rows that cover the remaining columns **OR** Find minimum number of prime implicants that covers all the minterms
     - E.g. Find $A'B'$ and AB cover terms that are not covered by others and they are essential prime implicants. $B'C$ and AC among themselves cover 10, 11 which are not covered by others.
     - So, one of them has to be included in the list of essential prime implicants making it three.

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

QUINE-McCLUSKY METHOD-11

	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)							√	√	√	√

- **Selection of essential prime implicants from this table is done in the following way.**
  1. 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.
  2. Essential prime implicants are easy to find using the prime implicant chart. If a given column contains only one √ or X, then the corresponding row is an essential prime implicant. (Encircle the √ or X)
  3. Each time a prime implicant is selected for inclusion in the minimum sum, the corresponding row should be crossed out.
  4. After doing this, the columns which correspond to all minterms covered by that prime implicant should also be crossed out.
     - The essential prime implicants and the corresponding rows and columns are crossed out.
  5. A minimum set of prime implicants must now be chosen to cover the remaining columns.
     - Find minimum set of rows that cover the remaining columns **OR** Find minimum number of prime implicants that covers all the minterms
     - E.g. Find $A'B'$ and AB cover terms that are not covered by others and they are essential prime implicants. $B'C$ and AC among themselves cover 10, 11 which are not covered by others.
     - So, one of them has to be included in the list of essential prime implicants making it three.

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

QUINE-McCLUSKY METHOD

- **Example Problem**
- f(a, b, c, d) =Σ m(0, 1, 2, 5, 6, 7, 8, 9, 10, 14)
- Determination of Prime Implicants

Column I

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	✓

Column II

0, 1	000–	✓
0, 2	00–0	✓
0, 8	–000	✓
1, 5	0–01	
1, 9	–001	✓
2, 6	0–10	✓
2, 10	–010	✓
8, 9	100–	✓
8, 10	10–0	✓
5, 7	01–1	
6, 7	011–	
6, 14	–110	✓
10, 14	1–10	✓

Column III

0, 1, 8, 9	–00–
0, 2, 8, 10	–0–0
~~0, 8, 1, 9~~	~~–00–~~
~~0, 8, 2, 10~~	~~–0–0~~
2, 6, 10, 14	-- 10
~~2, 10, 6, 14~~	~~-- 10~~

QUINE-McCLUSKY METHOD

- Prime Implicant Chart

		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*					×	×				

- f = b′c′ + cd′ + a′bd

QUINE-McCLUSKY METHOD

- Prime Implicant Chart

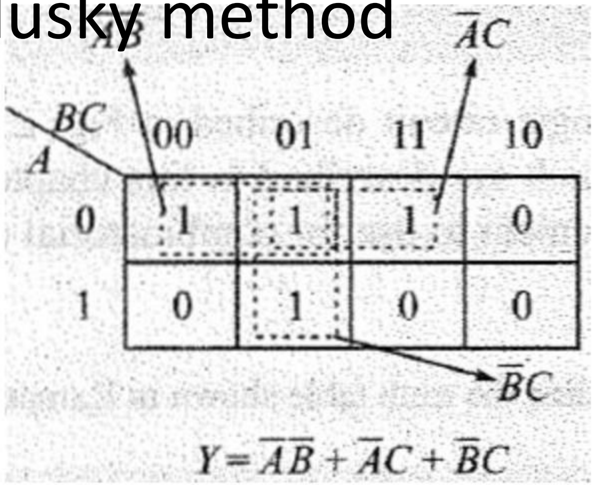
		0	1	2	5	6	7	8	9	10	14
(0, 1, 8, 9)	*b′c′*	×	×					×	×		
(0, 2, 8, 10)	*b′d′*	×		×				×		×	
(2, 6, 10, 14)	*cd′*			×		×				×	×
(1, 5)	*a′c′d*		×		×						
(5, 7)	*a′bd*				×		×				
(6, 7)	*a′bc*					×	×				

- **f = b′c′ + cd′ + a′bd**

QUINE-McCLUSKY METHOD-12

- Get a minimized expression for $Y = F(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C$ using K-map, EVM and quine mc-clusky method

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

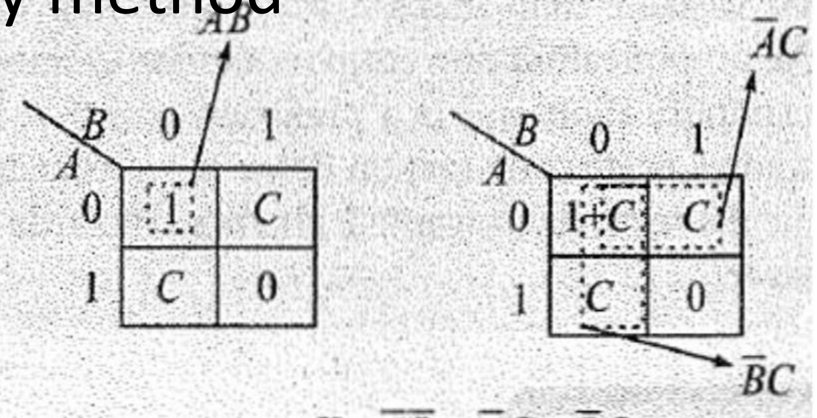


QUINE-McCLUSKY METHOD-12

- Get a minimized expression for $Y = F(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C$ using K-map, EVM and quine mc-clusky method

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

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



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

QUINE-McCLUSKY METHOD-12

- Get a minimized expression for $Y = F(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C$ using K-map, EVM and quine mc-clusky method

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

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

Stage 1			Stage 2	
ABC			*ABC*	
000	(0)	√	00–	(0, 1)
001	(1)	√	0–1	(1, 3)
			–01	(1, 5)
011	(3)	√		
101	(5)	√		

Prime implicants only from stage 2.
They are:
00–($A'B'$), 0–1 ($A'C$) and –01 ($B'C$)

	0	1	3	5
$A'B'$	√	√		
$A'C$		√	√	
$B'C$		√		√

All are essential
$Y = A'B' + A'C + B'C$

QUINE-McCLUSKY METHOD-13

- Example Problem
- F =Σ m(0, 1, 2, 5, 6, 7)

0	000	✓
1	001	✓
2	010	✓
5	101	✓
6	110	✓
7	111	✓

0, 1	00–
0, 2	0–0
1, 5	–01
2, 6	–10
5, 7	1–1
6, 7	11–

				0	1	2	5	6	7
1	→	(0, 1)	$a'b'$	X	X				
		(0, 2)	$a'c'$	X		X			
		(1, 5)	$b'c$		X		X		
2	→	(2, 6)	bc'			X		X	
3	→	(5, 7)	ac				X		X
		(6, 7)	ab					X	X

- A prime implicant chart which has two or more X's in every column is called a cyclic prime implicant chart.
- **F = a'b' + bc' + ac.**

QUINE-McCLUSKY METHOD-13

- All columns have two X's, so we will proceed by **trial and error**.
- Both (0, 1) and (0, 2) cover column 0, so we will try (0, 1).
- After crossing out row (0, 1) and columns 0 and 1, we examine column 2, which is covered by (0, 2) and (2, 6). The best choice is (2, 6) because it covers two of the remaining columns while (0, 2) covers only one of the remaining columns.
- After crossing out row (2, 6) and columns 2 and 6, we see that (5, 7) covers the remaining columns and completes the solution.
- **However, we are not guaranteed that this solution is minimum. We must go back and solve the problem over again starting with the other prime implicant that covers column 0.**

QUINE-McCLUSKY METHOD-13

		0	1	2	5	6	7
(0, 1)	*a′b′*	×	×				
(0, 2)	*a′c′*	×		×			
(1, 5)	*b′c*		×		×		
(2 6)	*bc′*			×		×	
(5, 7)	*ac*				×		×
(6, 7)	*ab*					×	×

- **F = a′c′ + b′c + ab**
- Note that in K-Map if each minterm on the map can be covered by two different loops.
- Similarly, each column of the prime implicant chart has two X’s, indicating that each minterm can be covered by two different prime implicants.

QUINE-McCLUSKY METHOD Programmed Exercise

- Get simplified expression of f(A, B, C, D, E) = Σ m (0, 2, 3, 5, 7, 9, 11, 13, 14, 16, 18, 24, 26, 28, 30) using Quine-Mc-Clusky method.

QUINE-McCLUSKY METHOD
Programmed Exercise

0	00000	✓
2	00010	✓
16	10000	✓
3	00011	✓
5	00101	✓
9	01001	✓
18	10010	✓
24	11000	✓
7	00111	✓
11	01101	✓
13	01101	✓
14	01110	✓
26	11010	✓
28	11100	✓
30	11110	✓

0, 2	000–0	✓
0, 16	–0000	
2, 3	0001–	
2, 18	–0010	
16, 18	100–0	✓
16, 24	1–000	
3, 7	00–11	
3, 11	0–011	
5, 7	001–1	
5, 13	0–101	
9, 11	010–1	
9, 13	01–01	
18, 26	1–010	
24, 26	110–0	
24, 28	11–00	
14, 30	–1110	
26, 30	11–10	
28, 30	111–0	

0, 2, 16, 18	–00–0
0, 2, 16, 18	–00–0
16, 18, 24, 26	1–0–0
24, 26, 28, 30	11--0

QUINE-McCLUSKY METHOD
Programmed Exercise

	0	2	3	5	7	9	11	13	14	16	18	24	26	28	30
(0, 2, 16, 18)	×	×								×	×				
(16, 18, 24, 26)										×	×	×	×		
(24, 26, 28, 30)												×	×	×	×
(2, 3)		×	×												
(3, 7)			×		×										
(3, 11)			×				×								
(5, 7)				×	×										
(5, 13)				×				×							
(9, 11)						×	×								
(9, 13)						×		×							
(14, 30)									×						×

QUINE-McCLUSKY METHOD Programmed Exercise

- Find a first minimum sum-of-products solution.

	0	2	3	5	7	9	11	13	14	16	18	24	26	28	30
*(0, 2, 16, 18)	⊗	X								X	X				
(16, 18, 24, 26)										X	X	X	X		
*(24, 26, 28, 30)												X	X	⊗	X
(2, 3)		X	X												
(3, 7)			X		X										
→ (3, 11)			X				X								
→ (5, 7)				X	X										
(5, 13)				X				X							
(9, 11)						X	X								
→ (9, 13)						X		X							
*(14, 30)									⊗						X

*Indicates an essential prime implicant.

- f = B′C′E′ + ABE′ + A′C′DE + A′B′CE + A′BD′E + BCDE′

QUINE-McCLUSKY METHOD Programmed Exercise

- Find a second minimum sum-of-products solution.

	0	2	3	5	7	9	11	13	14	16	18	24	26	28	30
*(0, 2, 16, 18)	⊗	×								×	×				
(16, 18, 24, 26)										×	×	×	×		
*(24, 26, 28, 30)												×	×	⊗	×
(2, 3)		×	×												
(3, 7)			×		×										
(3, 11)			×				×								
(5, 7)				×	×										
(5, 13)				×				×							
(9, 11)						×	×								
(9, 13)						×		×							
*(14, 30)									⊗						×

*Indicates an essential prime implicant.

- f = BCDE′ + B′C′E′ + ABE′ + A′B′DE + A′CD′E + A′BC′E

QUINE-McCLUSKY METHOD Programmed Exercise

- For this function, find a minimum sum-of-products solution, using the QuineMcCluskey method.
- (a) $f(a, b, c, d) = \Sigma\, m(1, 5, 7, 9, 11, 12, 14, 15)$
- (b) $f(a, b, c, d) = \Sigma\, m(0, 1, 3, 5, 6, 7, 8, 10, 14, 15)$

QUINE-McCLUSKY METHOD
Petrick's Method

- Petrick's method is a technique for determining all minimum sum-of-products solutions from a prime implicant chart.
- 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.

QUINE-McCLUSKY METHOD

Petrick's Method

Go, change the world

- Petrick's method is as follows:

1. Reduce the prime implicant chart by eliminating the essential prime implicant rows and the corresponding columns.
2. Label the rows of the reduced prime implicant chart P1, P2, P3, 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 (Pi0 + Pi1 +· · ·), where Pi0, Pi1 . . . 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.

QUINE-McCLUSKY METHOD
Petrick's Method

- The example shown in below tables has two minimum solutions.
- **F = a'b' + bc' + ac OR   F = a'c' + b'c + ab**

			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					×	×

QUINE-McCLUSKY METHOD
Petrick’s Method

- Petrick’s method, If all essential prime implicants and the minterms they cover should be removed from the chart.
- Then label the rows of the table P1, P2, P3, etc.
- We will form a logic function, P, which is true when all of the minterms in the chart have been covered.
  - Let P1 be a logic variable which is true when the prime implicant in row P1 is included in the solution, P2 be a logic variable which is true when the prime implicant in row P2 is included in the solution, etc.
- Because column 0 has X’s in rows P1 and P2, we must choose row P1 or P2 in order to cover minterm 0. Therefore, the expression **(P1 + P2)** must be true.
- In order to cover minterm 1, we must choose row P1 or P3; therefore, **(P1 + P3)** must be true.
- In order to cover minterm 2, **(P2 + P4)** must be true.
- Similarly, in order to cover minterms 5, 6, and 7, the expressions **(P3 + P5), (P4 + P6)** and **(P5 + P6)** must be true.

QUINE-McCLUSKY METHOD
Petrick's Method

- Because we must cover all of the minterms, the following function must be true:

P = (P1 + P2)(P1 + P3)(P2 + P4)(P3 + P5)(P4 + P6)(P5 + P6) = 1

- The expression for P in effect means that we must choose row P1 or P2, and row P1 or P3, and row P2 or P4, etc.

QUINE-McCLUSKY METHOD
Petrick's Method

- 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= (P1 + P2P3)(P4 + P2P6)(P5 + P3P6)

= (P1P4 + P1P2P6 + P2P3P4 + P2P3P6)(P5 + P3P6)

= P1P4P5 + P1P2P5P6 + P2P3P4P5 + <u>P2P3</u>P5<u>P6</u> + P1P3P4P6 + P1P<u>2P3P6 +</u> P<u>2P3P4</u><u>P6 </u>+ <u>P2P3P6</u>

= P1P4P5 + P1P2P5P6 + P2P3P4P5 + P1P3P4P6 + P2P3P6

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

P = P1P4P5 + P1P2P5P6 + P2P3P4P5 + P1P3P4P6 + P2P3P6

QUINE-McCLUSKY METHOD
Petrick's Method

- 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 to **F = a′c′ + b′c + ab,** which are the two minimum solutions.

QUINE-McCLUSKY METHOD

Simplification of Incompletely Specified Functions

- Given an incompletely specified function, the proper assignment of values to the don't-care terms is necessary in order to obtain a minimum form for the function.
- In this section will show how to modify the Quine-McCluskey method in order to obtain a minimum solution when don't-care terms are present.
- In the process of finding the prime implicants, we will treat the don't-care terms as if they were required minterms.
  - In this way, they can be combined with other minterms to eliminate as many literals as possible.
  - If extra prime implicants are generated because of the don't-cares, this is correct because the extra prime implicants will be eliminated in the next step anyway.
- When forming the prime implicant chart, the don't cares are 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.

QUINE-McCLUSKY METHOD

Simplification of Incompletely Specified Functions

- F(A, B, C, D) =Σ m(2, 3, 7, 9, 11, 13) +Σ d(1, 10, 15)
- The don't-care terms are treated like required minterms when finding the prime implicants:

1	0001	✓
2	0010	✓
3	0011	✓
9	1001	✓
10	1010	✓
7	0111	✓
11	1011	✓
13	1101	✓
5	1111	✓

(1, 3)	00–1	✓
(1, 9)	–001	✓
(2, 3)	001–	✓
(2, 10)	–010	✓
(3, 7)	0–11	✓
(3, 11)	–011	✓
(9, 11)	10–1	✓
(9, 13)	1–01	✓
(10, 11)	101–	✓
(7, 15)	–111	✓
(11, 15)	1–11	✓
(13, 15)	11–1	✓

(1, 3, 9, 11)	–0–1
(2, 3, 10,11)	–01–
(3, 7, 11, 15)	- - 11
(9, 11, 13, 15)	1 - - 1

QUINE-McCLUSKY METHOD

Simplification of Incompletely Specified Functions

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

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

*Indicates an essential prime implicant.

- F=B′C+CD+AD

QUINE-McCLUSKY METHOD

Simplification of Incompletely Specified Functions

- In the process of simplification, we have automatically assigned values to the don't-cares in the original truth table for F.
- If we replace each term in the final expression for F by its corresponding sum of minterms, the result is

F = (m2 + m3 + m10 + m11) + (m3 + m7 + m11 + m15) + (m9 + m11 + m13 + m15)

- Because m10 and m15 appear in this expression and m1 does not, this implies that the 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

Problems

- Find all prime implicants of the following function and then find all minimum solutions using Petrick's method:

1. $F(A, B, C, D) = \Sigma\, m(9, 12, 13, 15) + \Sigma\, d(1, 4, 5, 7, 8, 11, 14)$
2. $F(A, B, C, D) = \Sigma\, m(7, 12, 14, 15) + \Sigma\, d(1, 3, 5, 8, 10, 11, 13)$