Digital Logic Design (EL-1005)

LABORATORY MANUAL

Spring-2025

LAB 04
Simplification of Digital Circuits Using Karnaugh map

STUDENT NAME ROLL NO SEC

INSTRUCTOR SIGNATURE & DATE

MARKS AWARDED: /10

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES (NUCES), KARACHI

Lab Session 04: Simplification of Digital Circuits Using Karnaugh Map

OBJECTIVES:

The objectives of this lab is:

- To learn K-map and its usage in order to obtain cost effective circuit for implementation
- Implementation of Digital Circuits on Logisim/LogicWorks.

SOFTWARE:

- Logisim/LogicWorks

Introduction:

De-Morgan 's laws provide mathematical verification of the equivalency of the NAND and negative-OR gates and the equivalency of the NOR and negative-AND gates. The complement of a product of variables is equal to the sum of the complements of the variables. The complement of two or more AND variables is equivalent to the OR of the complements of the individual variables. The De Morgan's statements are,

Statement 1:

"The negation of conjunction is the disjunction of the negations". Or we can define that as "The compliment of the product of 2 variables is equal to the sum of the compliments of individual variables".

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

Statement 2:

"The negation of disjunction is the conjunction of the negations". Or we can define that as "The compliment of the sum of two variables is equal to the product of the compliment of each variable".

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

Figures of the about two statement shown below:

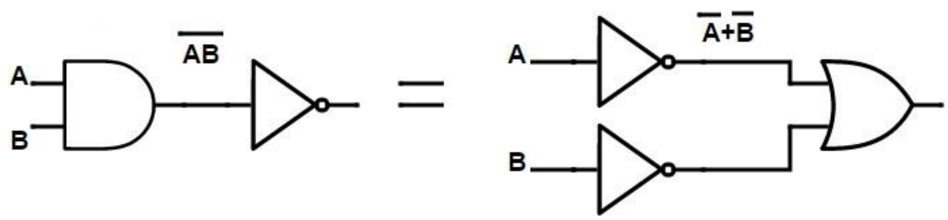


Figure 1 : NAND gate= Bubbled OR gate

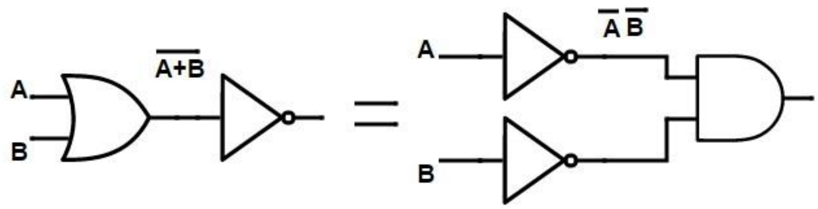
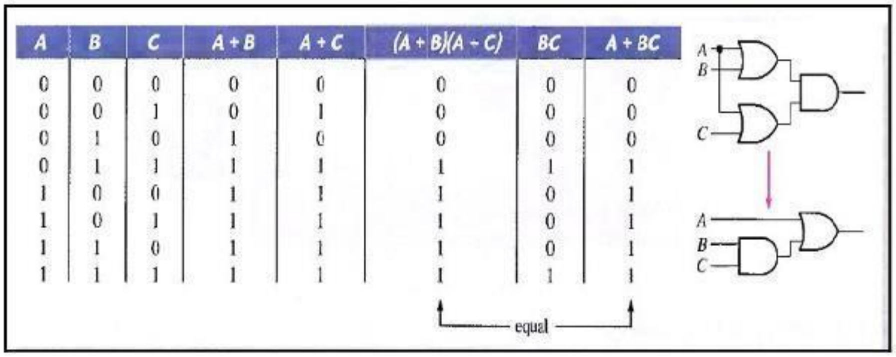


Figure 2 : NOR gate= Bubbled AND gate

Simpler expressions yield simpler hardware:

The proof is shown in table, which shows the truth table and the resulting logic circuit simplification.



A	B	C	A + B	A + C	(A + B)(A + C)	BC	A + BC
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Figure 2 : Simplification of circuit

Universality of logic Gates:

1. The NAND Gate as a Universal Logic Element

Any logic expression can be implemented using only NAND gates or only NOR gates and no other type of gate. NAND gates alone in the proper combination, can be used to perform each of the basic Boolean operations OR, AND, and INVERT.

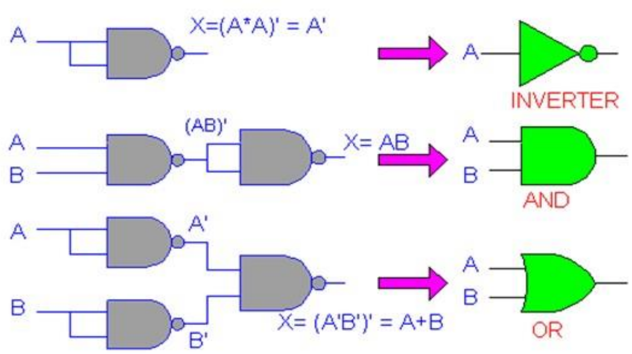


Figure 3 : NAND gate based Basic Gates

2. The NOR Gate as a Universal Logic Element

It can be shown that NOR gate can be arranged to implement any of the Boolean operations.

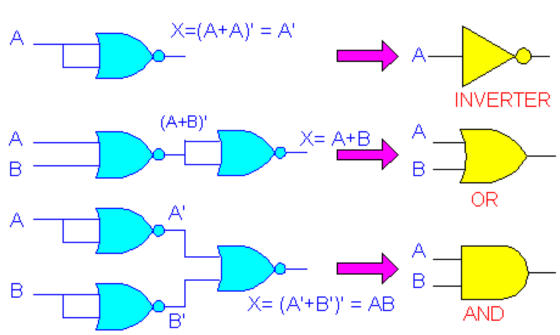


Figure 4 : NOR gate based Basic Gates

SOP AND POS Boolean Expressions:

1. The Sum-of-Products (SOP) Form:

When two or more product terms are summed by Boolean addition, the resulting expression is a sum-of-products (SOP).

Examples:

a) AB + ABC
b) ABC + CDE + BCD
c) AB + ABC + AC

2. The Product-of-Sums (POS) Form:

When two or more sum terms are multiplied, the resulting expression is a product-of-sums (POS).

Examples:

a) (A + B)(A + B + C)
b) (A + B + C)(C + D + E)(B + C + D)
c) (A + B)(A + B + C)(A + C)

Domain of a Boolean Expression:

The domain of a general Boolean expression is the set of variables contained in the expression in either complemented or uncomplemented form. For example, the domain of the expression AB + ABC is the set of variables A, B, C and the domain of the expression ABC + CDE + BCD is the set of variables A, B, C, D, E.

Standard Forms of Boolean Expressions:

1. The Standard SOP Form:

A standard SOP expression is one in which all the variables in the domain appear in each product term in the expression. We can convert any Non-standard SOP expression to a standard SOP by following the steps below:

Step 1: Multiply each nonstandard product term by a term made up of the sum of a missing variable and its complement. This results in two product terms. As you know, you can multiply anything by 1 without changing its value.

Step 2: Repeat Step 1 until all resulting product terms contain all variables in the domain in either complemented or uncomplemented form. In converting a product term to standard form, the number of product terms is doubled for each missing variable.

Example:

Convert the following Boolean expression into standard SOP form:

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

Solution

The domain of this SOP expression is A, B, C, D. Take one term at a time. The first term, $A\overline{B}C$, is missing variable D or $\overline{D}$, so multiply the first term by $D + \overline{D}$ as follows:

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

In this case, two standard product terms are the result.

The second term, $\overline{A}\,\overline{B}$, is missing variables C or $\overline{C}$ and D or $\overline{D}$, so first multiply the second term by $C + \overline{C}$ as follows:

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

The two resulting terms are missing variable D or $\overline{D}$, so multiply both terms by $D + \overline{D}$ as follows:

$$\begin{aligned}\overline{A}\,\overline{B} &= \overline{A}\,\overline{B}C + \overline{A}\,\overline{B}\,\overline{C} = \overline{A}\,\overline{B}C(D + \overline{D}) + \overline{A}\,\overline{B}\,\overline{C}(D + \overline{D})\\ &= \overline{A}\,\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}\end{aligned}$$

In this case, four standard product terms are the result.

The third term, $AB\overline{C}D$, is already in standard form. The complete standard SOP form of the original expression is as follows:

$$A\overline{B}C + \overline{A}\,\overline{B} + AB\overline{C}D = A\overline{B}CD + A\overline{B}C\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + AB\overline{C}D$$

2. The Standard POS Form

A standard POS expression is one in which all the variables in the domain appear in each sum term in the expression. We can convert any Non-standard POS expression to a standard POS by following the steps below:

Step 1: Add to each nonstandard product term a term made up of the product of the missing variable and its complement. This results in two sum terms. As you know, you can add 0 to anything without changing its value.

Step 2: Apply rule: A + BC = (A + B)(A + C)

Step 3: Repeat Step 1 until all resulting sum terms contain all variables in the domain in either complemented or uncomplemented form.

Example:

Convert the following Boolean expression into standard POS form:

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

Solution

The domain of this POS expression is A, B, C, D. Take one term at a time. The first term, $A + \overline{B} + C$, is missing variable D or $\overline{D}$, so add $D\overline{D}$ and apply rule 12 as follows:

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

The second term, $\overline{B} + C + \overline{D}$, is missing variable A or $\overline{A}$, so add $A\overline{A}$ and apply rule 12 as follows:

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

The third term, $A + \overline{B} + \overline{C} + D$, is already in standard form. The standard POS form of the original expression is as follows:

$$\begin{aligned}&(A + \overline{B} + C)(\overline{B} + C + \overline{D})(A + \overline{B} + \overline{C} + D) =\\ &\qquad (A + \overline{B} + C + D)(A + \overline{B} + C + \overline{D})(A + \overline{B} + C + \overline{D})(\overline{A} + \overline{B} + C + \overline{D})(A + \overline{B} + \overline{C} + D)\end{aligned}$$

K-MAP:

A Karnaugh map provides a systematic method for simplifying Boolean expressions and, if properly used, will produce the simplest SOP or POS expression possible, known as the minimum expression. As you have seen, the effectiveness of algebraic simplification depends on your familiarity with all the laws, rules, and theorems of Boolean algebra and on your ability to apply them. The Karnaugh map, on the other hand, provides a "cookbook" method for simplification. Karnaugh maps can be used for expressions with two, three, four, and five variables, but we will discuss only 3-variable and 4-variable situations to illustrate the principles. The number of cells in a Karnaugh map, as well as the number of rows in a truth table, is equal to the total number of possible input variable combinations. For three variables, the number of cells is 2^3 = 8. For four variables, the number of cells is 2^4 = 16.

Karnaugh map is used to obtain optimized logic representation so that it can be implemented using a minimum number of logic gates. The sum-of-product form can always be implemented using AND gates feeding into an OR gate, and a product-of-sum form leads to OR gates feeding an AND gate.

SOP Expression Mapping on K-Map (3-Variables):

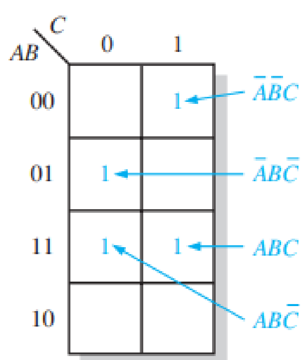
Map the following standard SOP expression on a Karnaugh map:

$$\overline{A}\,\overline{B}C + \overline{A}B\overline{C} + AB\overline{C} + ABC$$

Solution

Evaluate the expression as shown below. Place a 1 on the 3-variable Karnaugh map in Figure 4–29 for each standard product term in the expression.

$$\underset{001}{\overline{A}\,\overline{B}C} + \underset{010}{\overline{A}B\overline{C}} + \underset{110}{AB\overline{C}} + \underset{111}{ABC}$$



POS Expression Mapping on K-Map (4-Variables):

Map the following standard POS expression on a Karnaugh map:

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

Solution

Evaluate the expression as shown below and place a 0 on the 4-variable Karnaugh map in Figure 4–44 for each standard sum term in the expression.

$$\underbrace{(\overline{A} + \overline{B} + C + D)}_{1100}\underbrace{(\overline{A} + B + \overline{C} + \overline{D})}_{1011}\underbrace{(A + B + \overline{C} + D)}_{0010}\underbrace{(\overline{A} + \overline{B} + \overline{C} + \overline{D})}_{1111}\underbrace{(A + B + \overline{C} + \overline{D})}_{0011}$$

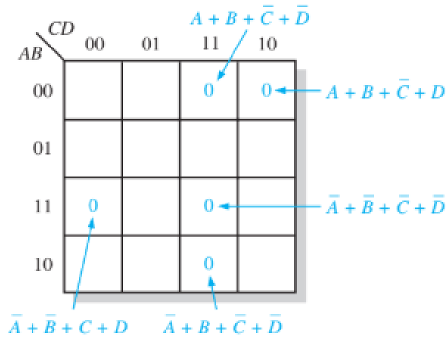


FIGURE 4–44

Karnaugh Map Simplification of SOP Expressions:

A minimum SOP expression is obtained by **grouping the 1s** and **determining the minimum SOP expression** from the map.

1. Grouping the 1s

You can group 1s on the Karnaugh map according to the following rules by enclosing those adjacent cells containing 1s. The goal is to maximize the size of the groups and to minimize the number of groups.

1. A group must contain either 1, 2, 4, 8, or 16 cells, which are all powers of two. In the case of a 3-variable map, $2^3 = 8$ cells is the maximum group.
2. Each cell in a group must be adjacent to one or more cells in that same group, but all cells in the group do not have to be adjacent to each other.
3. Always include the largest possible number of 1s in a group in accordance with rule 1.
4. Each 1 on the map must be included in at least one group. The 1s already in a group can be included in another group as long as the overlapping groups include noncommon 1s.

2. Determining the Minimum SOP Expression from the Map

1. Group the cells that have 1s. Each group of cells containing 1s creates one product term composed of all variables that occur in only one form (either uncomplemented or complemented) within the group. Variables that occur both uncomplemented and complemented within the group are eliminated. These are called *contradictory variables.*
2. Determine the minimum product term for each group.
   (a) For a 3-variable map:
      (1) A 1-cell group yields a 3-variable product term
      (2) A 2-cell group yields a 2-variable product term
      (3) A 4-cell group yields a 1-variable term
      (4) An 8-cell group yields a value of 1 for the expression
   (b) For a 4-variable map:
      (1) A 1-cell group yields a 4-variable product term
      (2) A 2-cell group yields a 3-variable product term
      (3) A 4-cell group yields a 2-variable product term
      (4) An 8-cell group yields a 1-variable term
      (5) A 16-cell group yields a value of 1 for the expression
3. When all the minimum product terms are derived from the Karnaugh map, they are summed to form the minimum SOP expression.

<u>Example:</u>

Use a Karnaugh map to minimize the following standard SOP expression:

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

Solution

The binary values of the expression are

$$101 + 011 + 001 + 000 + 100$$

Map the standard SOP expression and group the cells as shown in Figure 4–37.

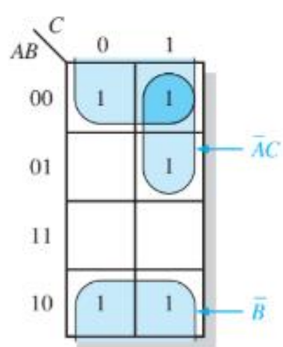


FIGURE 4–37

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

Keep in mind that this minimum expression is equivalent to the original standard expression.

Karnaugh Map Simplification of POS Expressions:

The process for minimizing a POS expression is basically the same as for an SOP expression except that you group 0s to produce minimum sum terms instead of grouping 1s to produce minimum product terms. The rules for grouping the 0s are the same as those for grouping the 1s

Example:

Use a Karnaugh map to minimize the following standard POS expression:

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

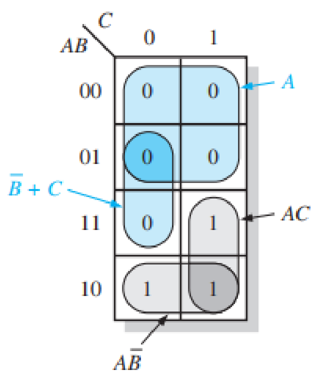
Also, derive the equivalent SOP expression.

Solution

The combinations of binary values of the expression are

$$(0 + 0 + 0)(0 + 0 + 1)(0 + 1 + 0)(0 + 1 + 1)(1 + 1 + 0)$$

Map the standard POS expression and group the cells as shown in Figure 4–45.



Notice how the 0 in the 110 cell is included into a 2-cell group by utilizing the 0 in the 4-cell group. The sum term for each blue group is shown in the figure and the resulting minimum POS expression is

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

Keep in mind that this minimum POS expression is equivalent to the original standard POS expression.

Grouping the 1s as shown by the gray areas yields an SOP expression that is equivalent to grouping the 0s.

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

Lab Task:

Question 01:

Using K-MAP, write down minimum SOP and actual expression for each case. Create Truth Table of Actual expression and implement Both actual and reduced expression on Logisim.

1.

AB \ C	0	1
00	1	
01		1
11	1	1
10		

2.

AB \ C	0	1
00	1	1
01	1	
11		1
10	1	1

Question 02:

Make a POS Simplified Expression of the following. Note the Blank spaces represent 0.

1.

AB \ CD	00	01	11	10
00	1	1		
01	1	1	1	1
11				
10		1	1	

2.

AB \ CD	00	01	11	10
00	1			1
01	1	1		1
11	1	1		1
10	1		1	1

Question: 3 Use K-MAP to minimize the given SOP expression. Implement the minimized SOP on Logisim and Complete Truth Table.

$$A\bar{B}C + \bar{A}BC + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C}$$

Question: 04 Use K-MAP to minimize the given SOP expression. Implement the minimized SOP on Logisim and Complete Truth Table. Also verify if and POS expression can be made using the derived K-Map if yes Built one and its circuit on the software.

$$\overline{B}\,\overline{C}\,\overline{D} + \overline{A}B\overline{C}\,\overline{D} + AB\overline{C}\,\overline{D} + \overline{A}\,\overline{B}CD + A\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + \overline{A}BC\overline{D} + ABC\overline{D} + A\overline{B}C\overline{D}$$

INSTRUCTIONS FOR SUBMISSION

1. **Create a Word file, having screenshots of circuits given as Lab task.**
2. **Upload World file and .CRIC file of Logisim on Google Classroom.**