Digital Logic Design (EL-1005)

LABORATORY MANUAL

Spring-2022

LAB 05
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

Date: 28th Feb 2022

Lab Session 05: 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.

SOFTWARE:

- Logisim – Logic Works

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:

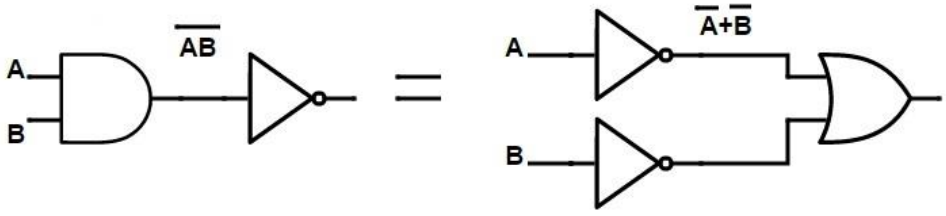


Figure 1 : NAND gate= Bubbled OR gate

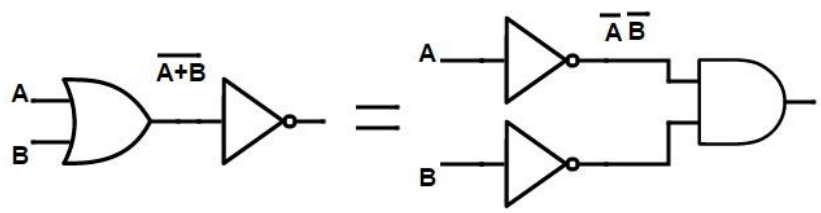
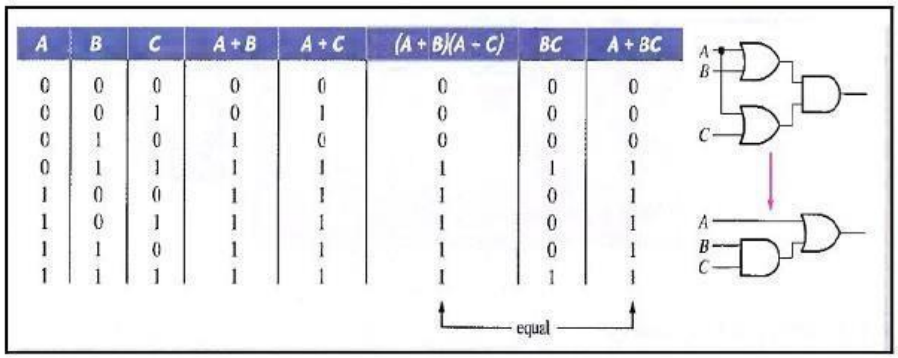


Figure 2 : NOR gate= Bubbled AND gate

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

K-MAP:

The Karnaugh map (K-map) is a method used to simplify Boolean expressions. K-Map is a grid- like representation of a truth table that gives more insight. The required Boolean results are transferred from a truth table onto a two-dimensional grid where the cells are ordered in gray code and each cell position represents one combination of input conditions, while each cell value represents the corresponding output value. Optimal groups of 1s or 0s are identified, which represent the terms of a canonical form of the logic in the original truth table. These terms can be used to write a minimal Boolean expression representing the required logic.

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.

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.

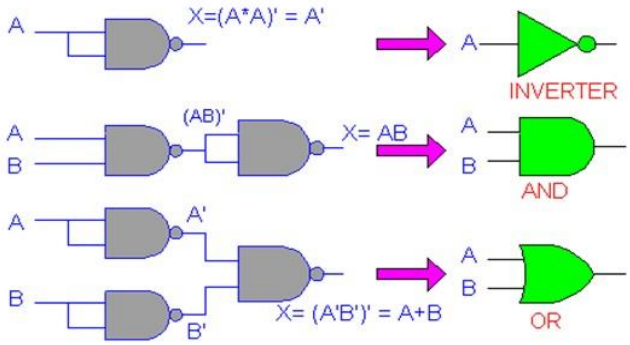


Figure 3 : NAND gate based Basic Gates

2. The NOR Gate as a Universal Logic Element

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

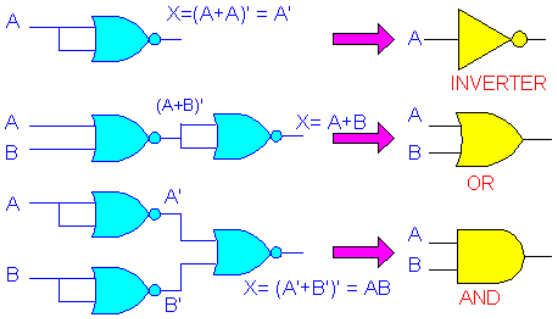


Figure 4 : NOR gate based Basic Gates

Report Experiment 05

Question: 01 Design circuit diagram in Logic Works given expressions by using NAND gates only

1. **ABC + D' + E'**
2. **ABC + DE**

Question: 02 simplify the following Boolean expression: and design circuit in Logisim/Logic Works.

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

Basic rules of Boolean algebra.

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

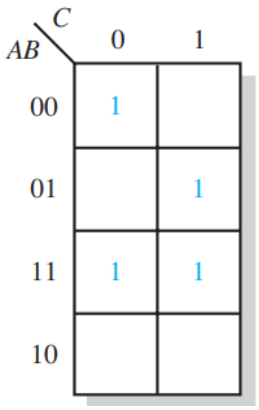
A, B, or C can represent a single variable or a combination of variables.

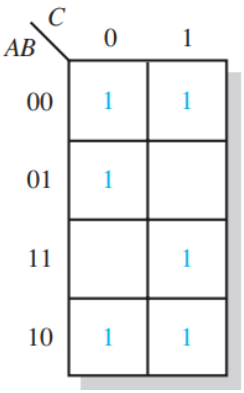
Question: 03

a. **Group the 1s in each of the Karnaugh maps in Figures**

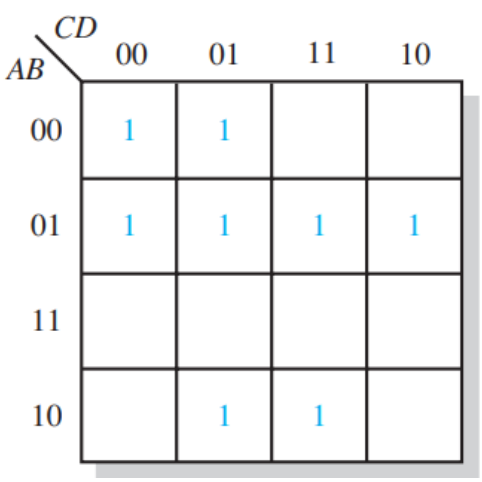
b. **Determine the product terms for each of the Karnaugh maps in Figures and write the resulting minimum SOP expression on Logisim/Logic Works.**

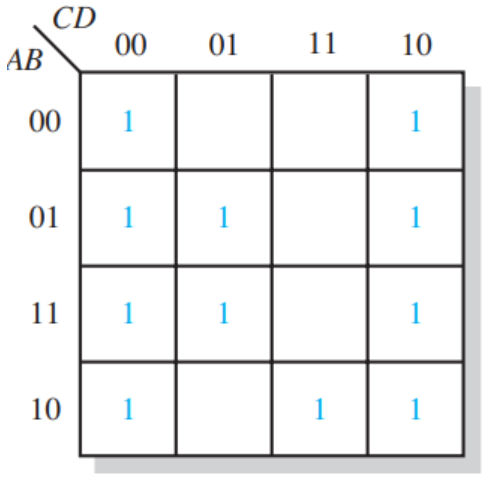
1.





2.





Question: 04 Use a Karnaugh map to minimize the following SOP expression Implement the minimized SOP on Logisim and Complete Truth Table.

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

Question: 05 Use a Karnaugh map to minimize the following POS expression Implement the minimized POS on Logisim and Complete Truth Table.

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

Question: 06 Convert the following Boolean expression into standard SOP form and design circuit diagram in Logic Works /Logisim:

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

Question: 07 Convert the following Boolean expression into standard POS form and design circuit diagram in Logic Works/Logisim:

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

INSTRUCTIONS FOR SUBMISSION

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