

Digital Logic Design (EL-1005) LABORATORY MANUAL Spring-2024



LAB 05 Simplification of Digital Circuits Using Karnaugh Map Hardware Implementations

STUDENT NAME

ROLL NO

SEC

INSTRUCTOR SIGNATURE & DATE

MARKS AWARDED: /10

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

Prepared By: Muhammad Nouman Hanif

Lab Session 05: Simplification of Digital Circuits Using Karnaugh Map Hardware Implementations

OBJECTIVES:

The objectives of this lab are:

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

APPARATUS:

- Logic Works
- Logic Probe
- Logic Trainer

COMPONENTS:

ICs 74LS86 (XOR), 74LS266 (XNOR), 74LS02 (NOR), 74LS00 (NAND), 74LS08 (AND), 74LS32 (OR), 74LS04 (NOT)

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 complement 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 complement of the sum of two variables is equal to the product of the complement of each variable".

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

Figures of the about two statements shown below:

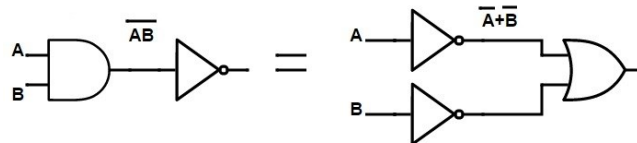


Figure 1 : NAND gate= Bubbled OR gate

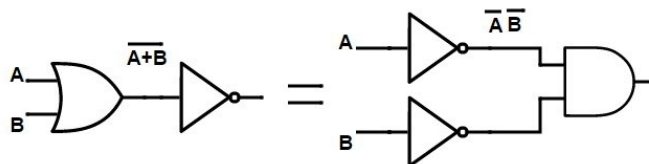


Figure 2 : NOR gate= Bubbled AND gate

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\bar{A} = 0$	$A + \bar{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$

Figure 2: Boolean Algebra Simplification Table

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.

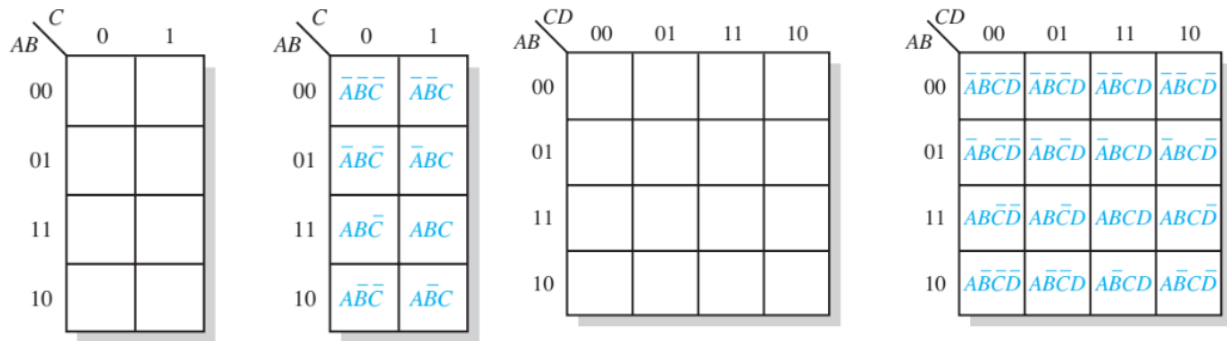


Figure 3: A 3-variable and 4-variable Karnaugh map showing Boolean product terms for each cell

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.

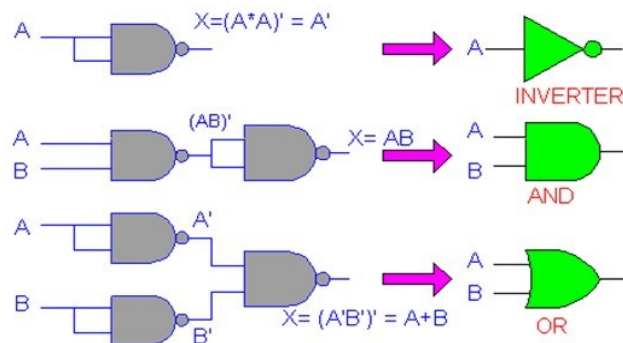


Figure 3 : NAND gate based Basic Gates

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.

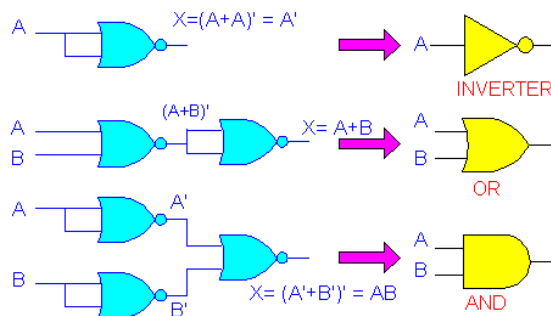


Figure 4 : NOR gate based Basic Gates