**National University of Computer and Emerging Sciences**

**Digital Logic Design Lab 02**

****

Fast School of Computing

FAST-NU, Lahore, Pakistan

Page **2** of **4**

**Objectives**

• Simulate, analyze, and integrate digital circuits with LogicWorks

• Laws of Boolean Algebra

• Minterms & Maxterms

• SOP & POS forms

Contents

1. Introduction to Boolean Algebra ......................................................................................................2 2. Laws of Boolean Algrbra ...................................................................................................................3 ACTIVITY 1.................................................................................................................................................3 ACTIVITY 2.................................................................................................................................................3 ACTIVITY 3.................................................................................................................................................3 ACTIVITY 4.................................................................................................................................................4 REFERENCES..............................................................................................................................................4

**1. Introduction to Boolean Algebra**

Boolean Algebra and Switching Algebra are two mathematical disciplines that deal with the manipulation of binary values. They are used in digital circuit design to simplify expressions and represent the behavior of digital circuits.

**Boolean Algebra** was first introduced by George Boole in 1854, and it deals with binary values, i.e., 0 and 1. It uses logical operators, such as AND, OR, NOT, etc., to manipulate binary values. The rules and laws of Boolean Algebra are used to simplify Boolean expressions and make the design of digital circuits easier.

**Switching Algebra**, on the other hand, is a specialized form of Boolean Algebra that deals specifically with the behavior of digital circuits. It uses the same binary values and logical operators as Boolean Algebra but goes a step further by defining the behavior of switches in digital circuits. This discipline is used to simplify expressions, analyze the behavior of digital circuits, and to study the performance of digital circuits.

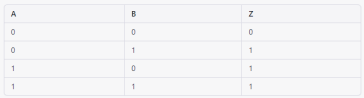
A **truth table** is a tabular representation of all possible combinations of inputs and the corresponding outputs of a logic function. It is used to show the behavior of digital circuits and to check the correctness of digital logic design.

For example, consider a logic function with two inputs A and B and one output Z, described as Z = A + B. The truth table for this function is as follows:

Page **2** of **4**

© Hazoor Ahmad FAST NU LHR

Page **3** of **4**

****

The first two columns represent the inputs A and B, and the third column represents the output Z. The rows of the table represent all possible combinations of inputs. Each row in the table shows the inputs and the corresponding output for that combination. In this example, the output Z is 1 if either A or B is 1. If both inputs are 0, then the output is 0.

**2. Laws of Boolean Algrbra**

**Idempotent Law:** A + A = A and A . A = A.

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

**Associative Law:** A + (B + C) = (A + B) + C and A . (B . C) = (A . B) . C

**Distributive Law:** A . (B + C) = (A . B) + (A . C) and A + (B . C) = (A + B) . (A + C) **Identity Law:** A + 0 = A and A . 1 = A

**Domination Law**: A + 1 = 1 and A . 0 = 0

**DeMorgan’s law**: (A + B)’ = A’ . B’ and (A . B) ’ = A’ + B’

**Uniting Law:** AB + ABʹ = A and (A + B)(A + Bʹ) = A

**Absorption Law**: A + AB = A and A(A + B) = A

**Elimination Law**: A + AʹB = A + B and A(Aʹ + B) = AB

**Consensus Law**: AB + AʹC + BC = AB + AʹC and (A + B)(Aʹ + C)(B + C) = (A + B)(Aʹ + C) **ACTIVITY 1**

With the help of other laws prove the following laws:

∙ Demorgan’s Law

∙ Uniting Law

∙ Absorption Law

∙ Elimination Law

∙ Consensus Law

**ACTIVITY 2**

Convert the following to **POS** form by using the **Distributive law,** implement on **Logic Works** and verify through truth table**.**

��′ + ��′

**ACTIVITY 3**

For the Boolean Functions **E** and **F**:

a) List min-term and max-terms of each function.

Page **3** of **4**

© Hazoor Ahmad FAST NU LHR

Page **4** of **4**

b) Express in sum-of-min-terms algebraic form.

c) Implement on **Logic works.**

| **X** | **Y** | **Z** | **E** | **F** |
| --- | --- | --- | --- | --- |
| **0** | **0** | **0** | **0** | **1** |
| **0** | **0** | **1** | **1** | **0** |
| **0** | **1** | **0** | **1** | **1** |
| **0** | **1** | **1** | **0** | **0** |
| **1** | **0** | **0** | **1** | **1** |
| **1** | **0** | **1** | **0** | **0** |
| **1** | **1** | **0** | **1** | **0** |
| **1** | **1** | **1** | **0** | **1** |

.

**ACTIVITY 4**

Build a circuit for the Boolean function **F = AB + (A XOR B) C + B NAND D** and find the truth table by using LogicWorks. Implement **F** in Logic Works, and determine input & output waveforms.

**REFERENCES**

• https://www.math

drills.com/multiop/mixed\_adding\_subtracting\_base\_system\_08\_octal\_4digit\_001.php • https://miniwebtool.com/octal-calculator/

• https://www.geeksforgeeks.org/arithmetic-operations-of-hexadecimal-numbers/ • https://www.cs.uregina.ca/Links/class-info/201/LW5/lecture.html

• http://site.iugaza.edu.ps/ahamada/fall-2018/combinational-digital-design-lab/

Page **4** of **4**

© Hazoor Ahmad FAST NU LHR