DIGITAL SYSTEM

Boolean Operation

Mardziah Mohd Shaari
Subri Abu Kassim

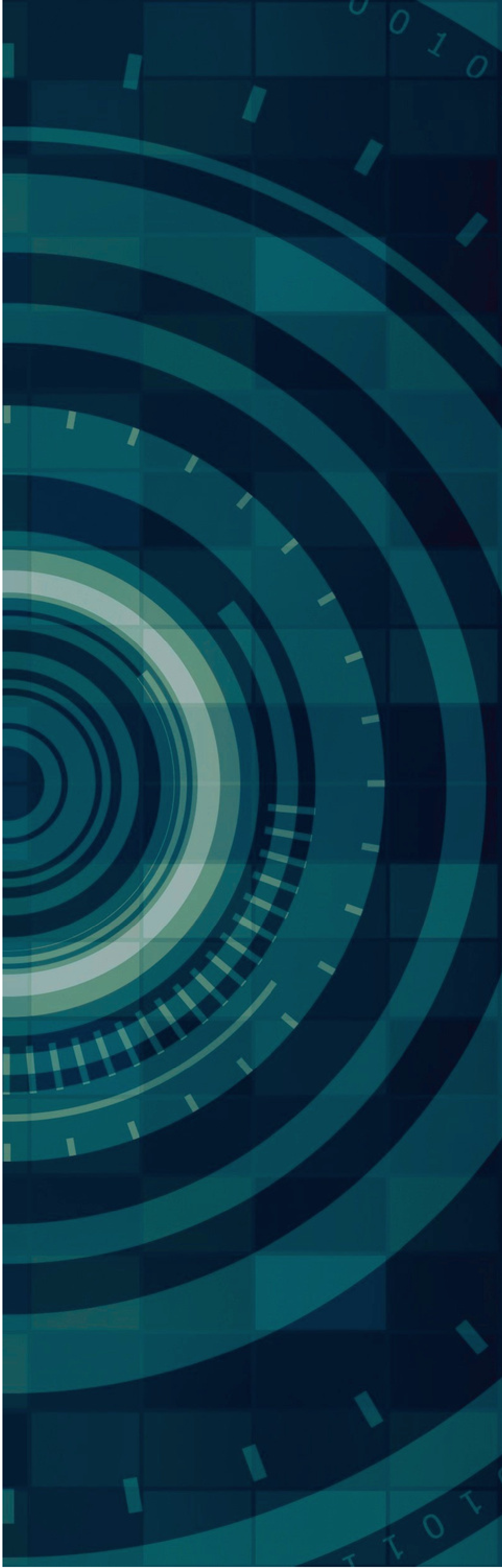
2022

Digital System

Boolean Operations

Written By:

Mardziah Mohd Shaari
Subri Abu Kassim



First published 2022

e ISBN 978-967-2421-54-2

Editor:
Zuraini Ghani

Writer:
Mardziah Binti Mohd Shaari
Subri Bin Abu Kassim

Published by:
Politeknik Ungku Omar,
Jalan Raja Musa Mahadi, 31400 Ipoh, Perak.
Tel: 05-5457656 Fax: 05-5471162

Digital System

Boolean Operations

WRITER:

MARDZIAH MOHD SHAARI
MECHANICAL ENGINEERING DEPARTMENT

SUBRI ABU KASSIM
KOLEJ VOKASIONAL LEBOH CATOR

EDITOR:

ZURAINI GHANI

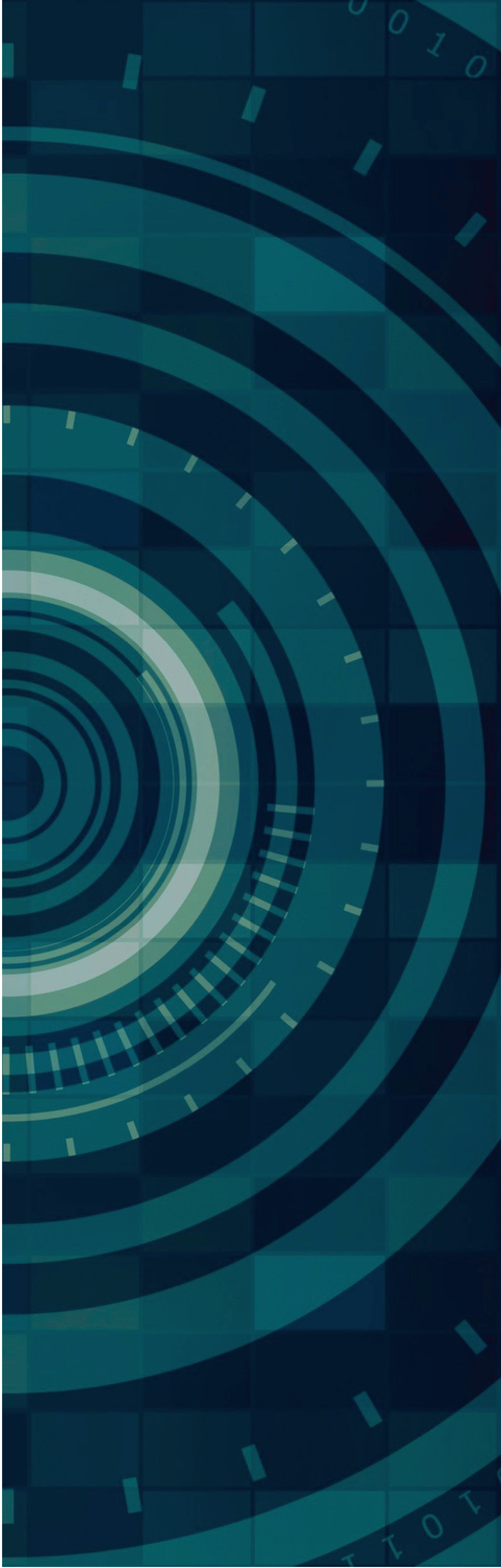
2022 EDITION

DEDICATION

To all students -
Hope this eBook will help to enhance your knowledge and understanding of Boolean Operations.
Make reading pleasure in your learning and life.

PREFACE

This eBook is the study of the principles and techniques of modern digital systems. It teaches the fundamental principles of Boolean algebra. This topic covers the symbols, operations and functions of logic gates. The content of this e-book contains an explanation of gate logic functions and truth tables. Students can illustrate the basic concepts of Boolean Algebra and use them in logic circuits analysis and design.



ACKNOWLEDGEMENT

Alhamdulillah, finally we are able to finish this eBook. We are very grateful to our colleagues who gave us a chance to work on this project.

We would like to thank them for giving us valuable suggestions and ideas, as well as the polytechnic that provided all of the resources needed in order for it to be successful.

Our family always encouraged us through every rough patch with their support. Last but not least, thank you to everyone involved rather directly or indirectly.

ABOUT THE AUTHOR

Mardziah Binti Mohd Shaari

Course Coordinator of Digital System in Mechanical Engineering Department, Politeknik Ungku Omar.

Subri Bin Abu Kassim

Senior lecturer in Kolej Vokasional Leboh Cator, Ipoh.

CONTENS

2 Course Learning Outcome

3 Boolean Constant and Variables

4 The Basic Gates: 1. AND Gate

6 The Basic Gates: 2. OR Gate

8 The Basic Gates: 3. NOT Gate

10 Universal Gate

12 Boolean Algebra

13 Laws of Boolean Algebra

14 Rules of Boolean Algebra

15 Q & A

17 Logic Expression

18 What You Should Do?

19 Karnaugh Map

20 Steps to Solve Expression Using K-Map

21 Problems

23 What You Should Do

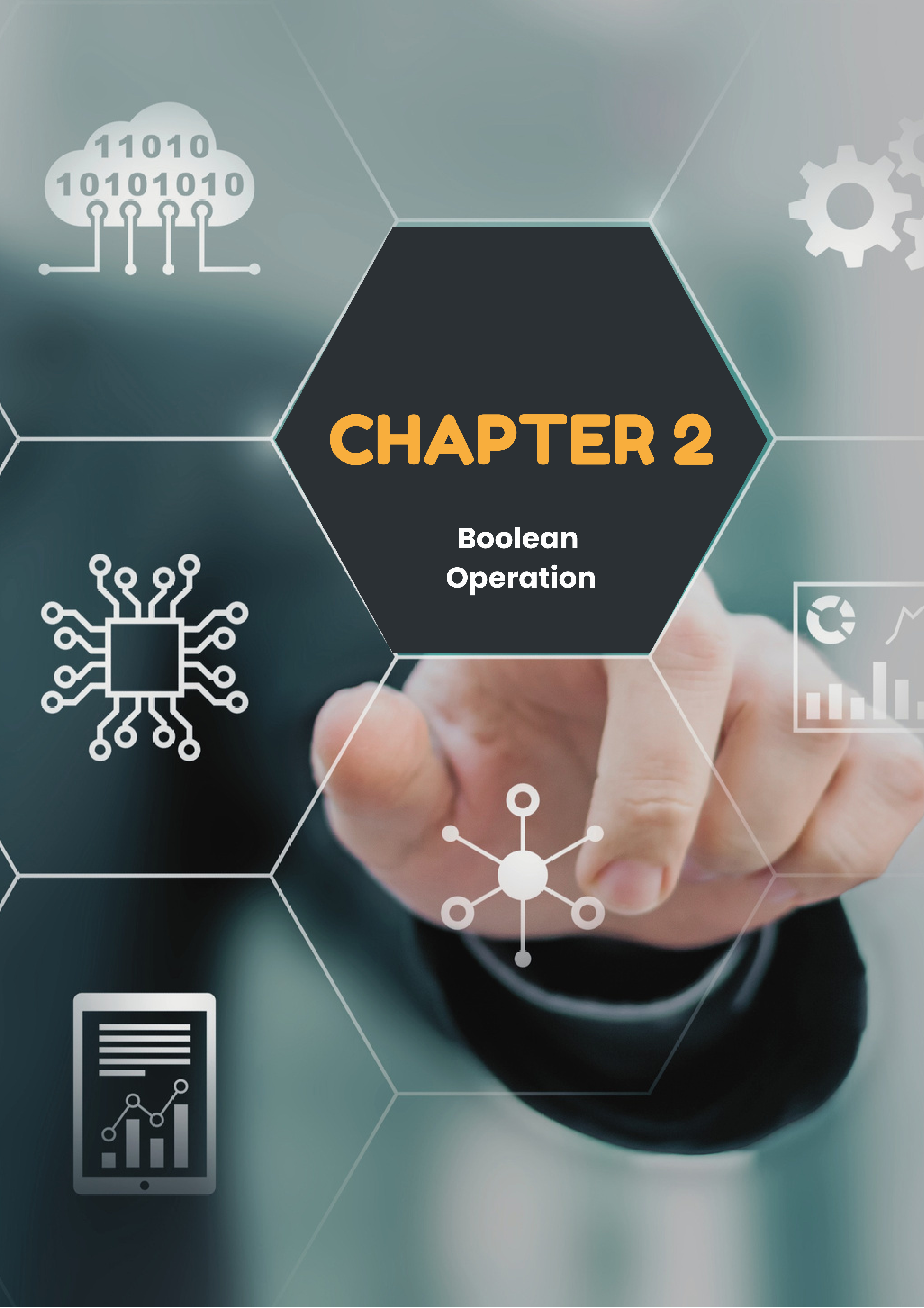
24 Mapping A Standard POS Expression

24 Review Question

25 References

CHAPTER 2

Boolean Operation



COURSE LEARNING OUTCOMES (CLO1)

Distinguish the characteristics and operations of various digital circuit (C4, PLO1)

This presentation is about the basic concepts of Digital Electronics:

- BASIC LOGIC CONCEPTS
- BASIC LOGIC FUNCTIONS
- LOGIC SYMBOLS
- TRUTH TABLE
- SUM OF PRODUCT
- PRODUCT OF SUM

BOOLEAN CONSTANTS AND VARIABLES

Logic 0	Logic 1
False	True
Off	On
Low	High
No	Yes
Open switch	Closed switch

BOOLEAN CONSTANTS AND VARIABLES

Boolean 0 and 1 do not represent actual numbers but instead represent the state of a voltage variable, or what is called its logic level.

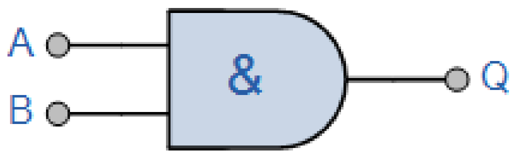
A voltage in a digital circuit is said to be at the logic 0 level or the logic 1 level, depending on its actual numerical value. In digital logic, several other terms are used synonymously with 0 and 1.

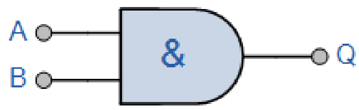
Some of the more common ones are shown in. We will use the 0/1 and LOW/HIGH designations most of the time.

Basic Digital logic is based on 3 primary functions (the basic gates):

1. AND GATE
2. OR GATE
3. NOT GATE

THE BASIC GATES:
1. AND GATE



Symbol	Truth Table		
 2-input AND Gate	B	A	Q
	0	0	0
	0	1	0
	1	0	0
	1	1	1
Boolean Expression **Q = A.B**	Read as A **AND** B gives Q		

The output state of a digital logic AND gate only returns "LOW" again when ANY of its inputs are at a logic level "0".

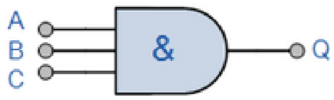
In other words for a logic AND gate, any LOW input will give a LOW output.
Boolean expression of:
A.B = Q.

THE BASIC GATES:
1. AND GATE

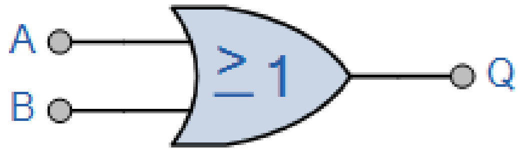
The 3-input Logic AND Gate

Because the Boolean expression for the logic AND function is defined as (.), which is a binary operation, AND gates can be cascaded together to form any number of individual inputs.

However, commercial available AND gate IC's are only available in standard 2, 3, or 4-input packages.
If additional inputs are required, then standard AND gates will need to be cascaded together to obtain the required input value.

Symbol	Truth Table			
 3-input AND Gate	C	B	A	Q
	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
Boolean Expression Q = A.B.C	Read as A AND B AND C gives Q			

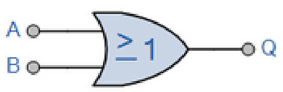
THE BASIC GATES:
2. OR GATE



The output, Q of a "Logic OR Gate" only returns "LOW" again when ALL of its inputs are at a logic level "0". In other words for a logic OR gate, any "HIGH" input will give a "HIGH", logic level "1" output.

The logic or Boolean expression given for a digital logic OR gate is that for Logical Addition which is denoted by a plus sign, (+) giving us the Boolean expression of: A+B = Q.

The 2-input Logic OR Gate

Symbol	Truth Table		
	B	A	Q
 2-input OR Gate	0	0	0
	0	1	1
	1	0	1
	1	1	1
Boolean Expression Q = A+B	Read as A OR B gives Q		

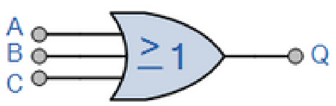
The OR operation produces a result (output) of 1 whenever any input is a 1. Otherwise the output is 0

THE BASIC GATES:
2. OR GATE

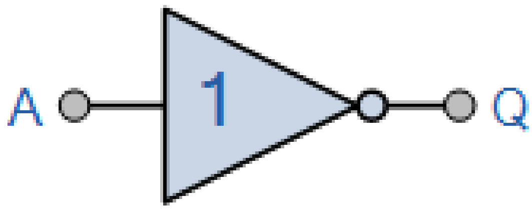
The 3-input Logic OR Gate

Like the AND gate, the OR function can have any number of individual inputs.

However, commercial available OR gates are available in 2, 3, or 4 inputs types. Additional inputs will require gates to be cascaded together.

Symbol	Truth Table			
	C	B	A	Q
 3-input OR Gate	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
Boolean Expression Q = A+B+C	Read as A OR B OR C gives Q			

THE BASIC GATES:
3. NOT GATE



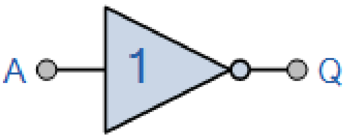
"If A is NOT true, then Q is true"

Inverting NOT gates are single input devicse which have an output level that is normally at logic level "1" and goes "LOW" to a logic level "0" when its single input is at logic level "1", in other words it "inverts" (complements) its input signal. The output from a NOT gate only returns "HIGH" again when its input is at logic level "0" giving us the Boolean expression of:

A = Q.

Then we can define the operation of a single input digital logic NOT gate as being:

THE BASIC GATES:
3. NOT GATE

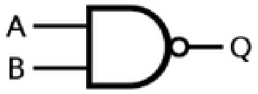
Symbol	Truth Table	
 Inverter or NOT Gate	A	Q
	0	1
	1	0
Boolean Expression Q = not A or $\overline{A}$	Read as inverse of A gives Q	

The “bubble” (o) present at the end of the NOT gate symbol above denotes a signal inversion (complementation) of the output signal. But this bubble can also be present at the gates input to indicate an active-LOW input.

This inversion of the input signal is not restricted to the NOT gate only but can be used on any digital circuit or gate as shown with the operation of inversion being exactly the same whether on the input or output terminal.

UNIVERSAL
GATE

NAND GATE

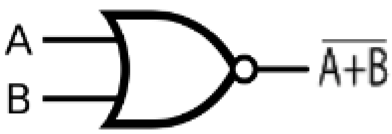


Q = A NAND B

Truth Table

Input A	Input B		Output Q
0	0		1
0	1		1
1	0		1
1	1		0

NOR GATE

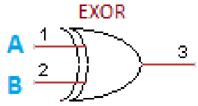


A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

UNIVERSAL
GATE

EX OR GATE

EXOR



$Y = A \oplus B = \bar{A}.B + A.\bar{B}$

Two Input XOR gate		
A	B	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

If any input is HIGH, the output is HIGH
If all inputs are SAME, the output is LOW

EX NOR GATE

Inputs		Output $X = \overline{A \oplus B}$
A	B	
0	0	1
0	1	0
1	0	0
1	1	1

If any input is HIGH, the output is LOW
If all inputs are SAME, the output is HIGH

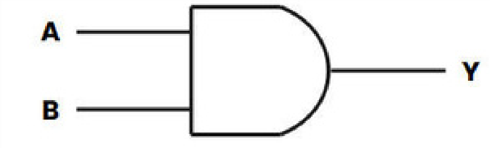
BOOLEAN ALGEBRA

BOOLEAN ALGEBRA

Boolean algebra finds its most practical use in the simplification of logic circuits.
After finding the circuit inputs and outputs, we can come up with either an expression or a truth table to describe what the circuit does.

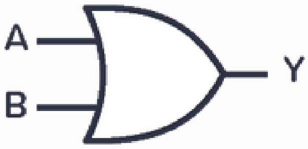
- Normal mathematical expressions can be simplified using the laws of algebra.
- For binary system, we use BOOLEAN ALGEBRA to simplify BOOLEANS EXPRESSIONS

AND GATE



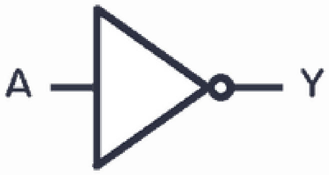
a) $X . 0 = 0$
b) $X . 1 = X$
c) $X . X = X$
d) $X . \overline{X} = 0$

OR GATE



a) $X + 0 = X$
b) $X + 1 = 1$
c) $X + X = X$
d) $X + \overline{X} = 1$

NOT GATE



a) If i/p A = 1, o/p Y = 0
b) If i/p A = 0, o/p Y = 1

LAWS OF BOOLEAN ALGEBRA

- Commutative Law
- Associative Law
- Distributive Law
- Auxiliary Law
- De Morgan's Law

Commutative Law

- These laws indicate that the order in which we OR(+) or AND(x) two variables is unimportant; the result is the same.

$$X + Y = Y + X$$
$$X . Y = Y . X$$

Associative Law

- Which state we can group the variables in an AND(x) expression or OR(+) expression any way we want.

$$X . (Y . Z) . Z = XYZ$$
$$X + (Y + Z) = (X + Y) + Z = X + Y + Z$$

Distributive Law

- Which state that an expression can be expended by multiplying term by term just the same as in ordinary algebra.

$$(W + X)(Y + Z) = WY + XY + WZ + XZ$$
$$X . (Y + Z) = (X . Y) + (X . Z)$$

Auxiliary Law

- Do not have any counterparts in an ordinary algebra.

a) $X + XY = X$

b) $X + \overline{X}Y = X + Y$

c) $X (X + Y) = X$

d) $X (\overline{X} + Y) = XY$

De Morgan's Theorem

- Useful in simplifying expression in which product (AND) and sum (OR) of variables is inverted. The two theorems are:

$$\overline{(X + Y)} = \overline{X} . \overline{Y}$$
$$\overline{(X . Y)} = \overline{X} + \overline{Y}$$

RULES OF BOOLEAN ALGEBRA

1.) $A + 0 = A$

2.) $A + 1 = 1$

3.) $A . 0 = 0$

4.) $A . 1 = A$

5.) $A + A = A$

6.) $A + \overline{A} = 1$

7.) $A . A = A$

8.) $A . \overline{A} = 0$

9.) $\overline{\overline{A}} = A$

10.) $A + AB = A$

11.) $A + \overline{A}B = A + B$

12.) $(A + B)(A + C) = A + BC$

Q & A

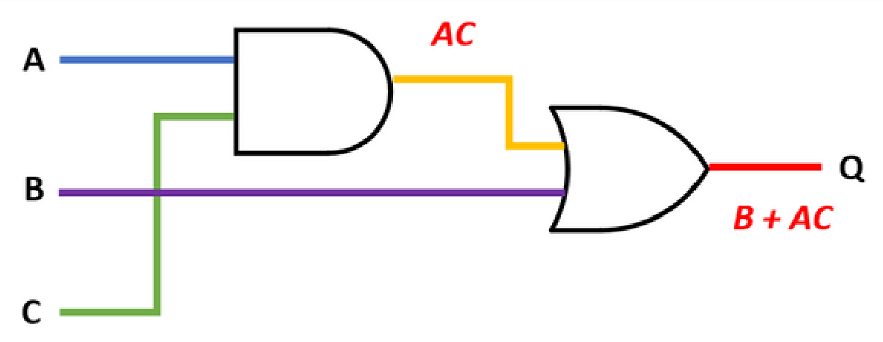
Simplify the Boolean equation below and draw the Logic Circuit:

1. Q = AB + A (B + C) + B (B + C)

Solution:

$$Q = (AB + AB) + AC + (BB) + BC$$
$$= AB + AC + (B + BC)$$
$$= AB + AC + (B (1 + C))$$
$$= AB + AC + B$$
$$= (AB + B) + AC$$
$$= (B (A + 1)) + AC$$
$$= B + AC$$

Logic Circuit:



Q & A

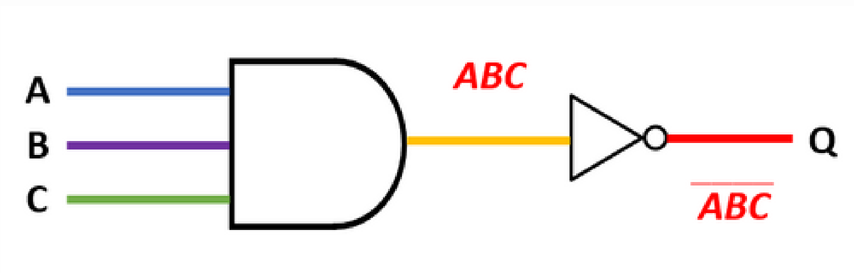
Simplify the Boolean equation below and draw the Logic Circuit:

2. $Q = \overline{AB} + \overline{ABC}$

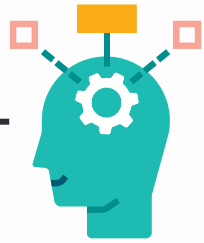
Solution:

$$
\begin{aligned}
Q &= \overline{AB} + \overline{ABC} \\
&= \bar{A} + \bar{B} + \bar{A} + \bar{B} + \bar{C} \\
&= (\bar{A} + \bar{A}) + (\bar{B} + \bar{B}) + \bar{C} \\
&= \bar{A} + \bar{B} + \bar{C} \\
&= \overline{ABC}
\end{aligned}
$$

Logic Circuit:



LOGIC EXPRESSION



THE SIGNIFICANCE OF MINTERMS AND MAXTERMS

Minterms and Maxterms used to define the two standard forms for logic expression.

SUM OF PRODUCT (SOP) - sum of minterms

PRODUCT OF SUM (POS) - sum of maxterms

SUM OF PRODUCT (SOP)

- The SOP expression is the equation of the logic function as read off the truth table to specify the input combinations when the output is a ***logical 1***
- When the SOP have 2 or more product terms are summed by Boolean addition, example;

i. $AB + ABC$

ii. $AB\overline{C} + C + B\overline{C}D$

PRODUCT OF SUM (POS)

- The POS expression is the equation of the logic function as read off the truth table to specify the input combinations when the output is a ***logical 0***
- When the POS have 2 or more product terms are multiply by Boolean addition, example;

i. $(A + B) + (A + B + C)$

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

What You Should Do?

Logic Expression - SOP
Example:

ROW	INPUT			OUTPUT
NUMBER	A	B	C	X
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

Consider this truth table using sum-of-product (SOP):

1) Find where output is **1**

2) Write down each product of inputs which create a **1**

3) Sum all of the products **(+)**

4) Draw the Karnaugh map

5) Find the Boolean Equation

KARNAUGH **MAP**

Karnaugh map or a K-map refers to a pictorial method that is utilised to minimise various Boolean expressions without using the Boolean algebra theorems along with the equation manipulations.

A Karnaugh map can be a special version of the truth table. We can easily minimise various expressions that have 2 to 4 variables using a K-map

- **The diagram below illustrates the correspondence between the Karnaugh map and the truth table for the general case of a two variable problem.**

A	B	F
0	0	a
0	1	b
1	0	c
1	1	d

Truth Table

B \ A	0	1
0	a	b
1	c	d

F

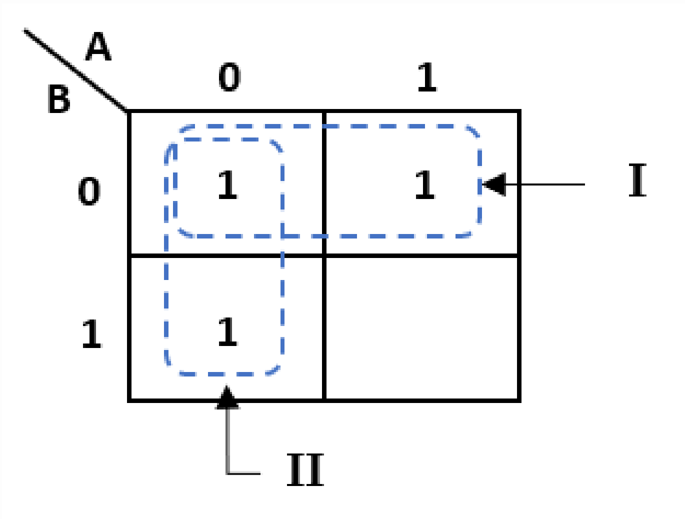
STEPS TO SOLVE EXPRESSION USING K-MAP

1. Select K-map according to the number of variables.
2. Identify minterms or maxterms as given in problem.
3. For SOP put 1's in blocks of K-map respective to the minterms (0's elsewhere).
4. For POS put 0's in blocks of K-map respective to the maxterms(1's elsewhere).
5. Make rectangular groups containing total terms in power of two like 2,4,8 ..(except 1) and try to cover as many elements as you can in one group.
6. From the groups made in step 5 find the product terms and sum them up for SOP form.

PROBLEMS

Example 1

Consider the expression $Z = f(A,B) = \bar{A}\bar{B} + A\bar{B} + \bar{A}B$ plotted on Karnaugh Map:



- Pairs of 1's are grouped as shown above, and the simplified answer is obtained.

Hence the simplified answer is $Z = \bar{A} + \bar{B}$

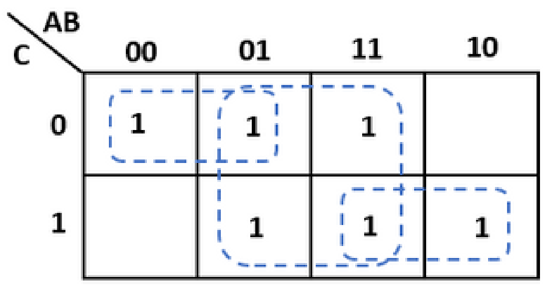
PROBLEMS

Example 2

Minimize the following problems using the Karnaugh maps method.

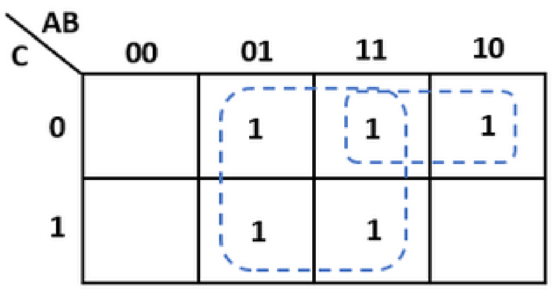
1. $Z = f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}B + AB\bar{C} + AC$
2. $Z = f(A,B,C) = \bar{A}B + B\bar{C} + BC + A\bar{B}\bar{C}$

1. $Z = f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}B + AB\bar{C} + AC$



- The minimized result obtained is $= \mathbf{B + AC + \bar{A}\bar{C}}$

2. $Z = f(A,B,C) = \bar{A}B + B\bar{C} + BC + A\bar{B}\bar{C}$



- The minimized result obtained is $= \mathbf{B + A\bar{C}}$

What You Should Do?

Logic Expression - POS
Example:

ROW	INPUT			OUTPUT
NUMBER	A	B	C	X
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Consider this truth table using product-of-sum (POS):

1) Find where output is **0**

2) Write down each product of inputs which create a **0**

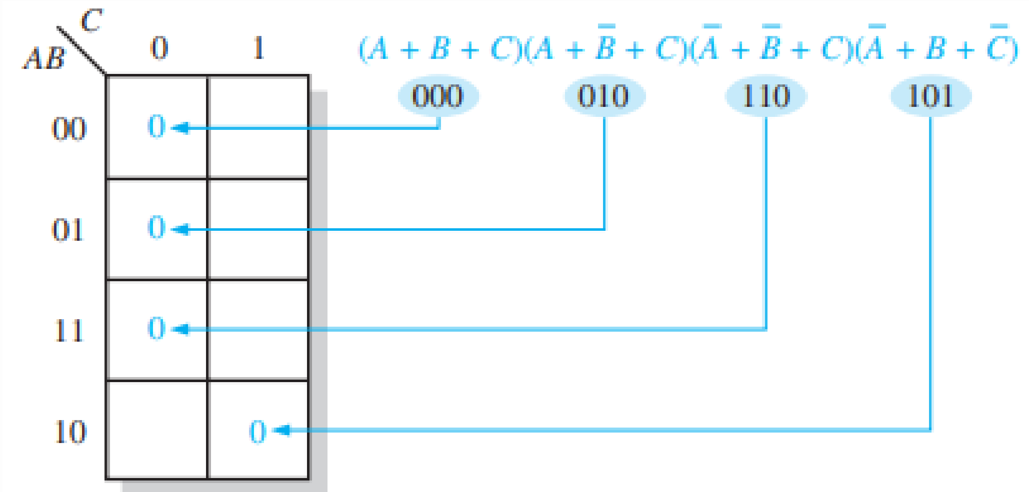
3) Sum all of the products **(x @ .)**

4) Draw the Karnaugh map

5) Find the Boolean Equation

MAPPING A STANDARD POS EXPRESSION

1. Determine the binary value of each sum term in the standard POS expression. This is the binary value that makes the term equal to 0.
2. As each sum term is evaluated, place a 0 on the Karnaugh map in the corresponding cell.



REVIEW QUESTIONS

What Do You Think ?

1) Which of the following expressions is in SOP form?

(a) $AB + CD + E$
(b) $AB(C + D)$
(c) $(A + B)(C + D + F)$
(d) $\overline{MN} + PQ$

2) Repeat Question 1 for the POS form

REFERENCES

Tocci, R. J. (1991). Digital systems: principles and applications. Pearson Education India.

Gate, B. N., Gate, G. N., Upper, U. I. L. I. O., Gate, O. R., Gate, N. O. R., Gate, X. O. R., & Gate, X. N. O. R. (1993). Introduction to digital logic design.

Sharma, D. S. (2017). Digital Electronics And Logic Design. India:S.K. Kataria & Sons.

Floyd, T. L. (2003). Digital Fundamentals (8th Edition). Prentice Hall.

Hanf, W. (1975). The Boolean algebra of logic. Bulletin of the American Mathematical Society, 81(3. P1), 587-589.
Chicago

Maini, A. K. (2007). Digital electronics: principles, devices and applications. John Wiley & Sons.