Minimisation Techniques

LEARNING OBJECTIVES

At the end of this lesson, you should be able to:

- Explain the importance of efficient Boolean simplification.
- Simplify Boolean expressions effectively.
- Apply Boolean algebra theorems to optimise logic circuits.
- Derive logic circuits from Boolean expressions.
- Construct Karnaugh maps for logic circuits.

Purpose of studying Boolean theorems and Karnaugh maps

- Helps in presenting more efficient representation of boolean functions
- Simplifies the process of designing circuits
- Provides solutions to boolean functions, that are more flexible and optimised.

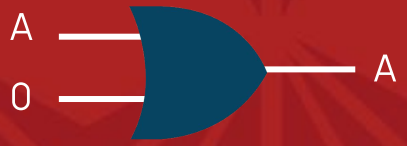
INTRODUCTION

- Given that boolean algebra helps with critical thinking and problem solving, the minimisation of boolean expression helps develop the efficiency in problem solving.
- This unit is focused on simplifying logical expressions and optimising circuits.

BASIC RULES OF BOOLEAN ALGEBRA

RULE 1: IDENTITY ELEMENT (OR)

$$A + 0 = A$$

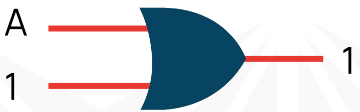


Truth table:

A	0	Output
0	0	0
1	0	1

RULE 2: NULL ELEMENT (OR)

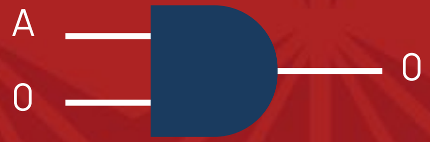
$$A + 1 = 1$$



A	1	Output
0	1	1
1	1	1

RULE 3: NULL ELEMENT (AND)

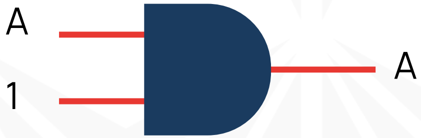
$$A \cdot 0 = 0$$



A	0	Output
0	0	0
1	0	0

RULE 4: IDENTITY ELEMENT (AND)

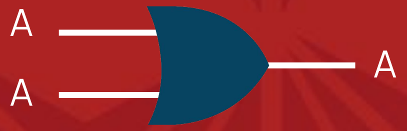
$$A \cdot 1 = A$$



A	1	Output
0	1	0
1	1	1

RULE 5: IDEMPOTENT LAW

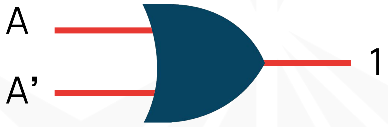
$$A + A = A$$



A	A	Output
0	0	0
1	1	1

RULE 6:

$$A + A' = 1$$



A	A'	Output
0	1	1
1	0	1

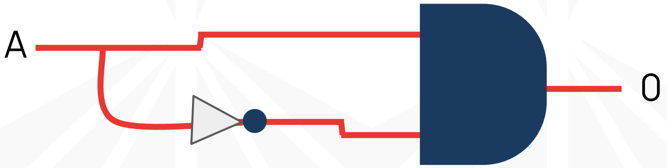
RULE 7:

$$A \cdot A = A$$

A	A	Output
0	0	0
1	1	1

RULE 8:

$$A \cdot A' = 0$$



A	A'	Output
0	1	0
1	0	0

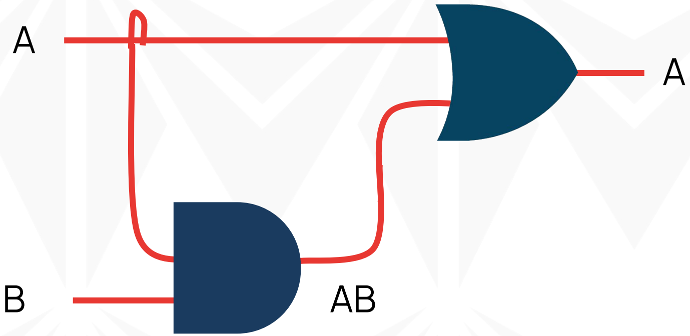
RULE 9:

$$A'' = A$$

A	A'	A''
0	1	0
1	0	1

RULE 10:

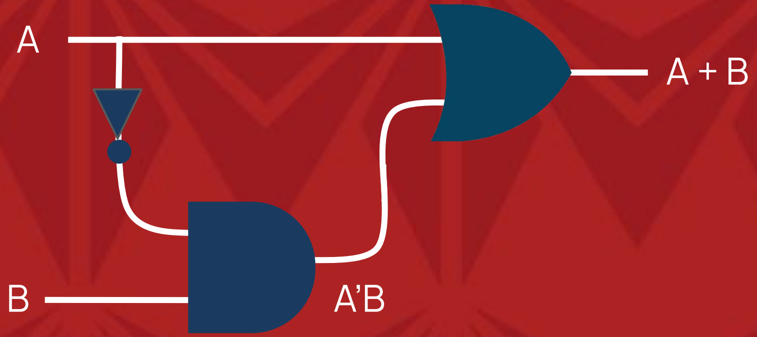
$$A + AB = A$$



A	B	AB	Output
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

RULE 11:

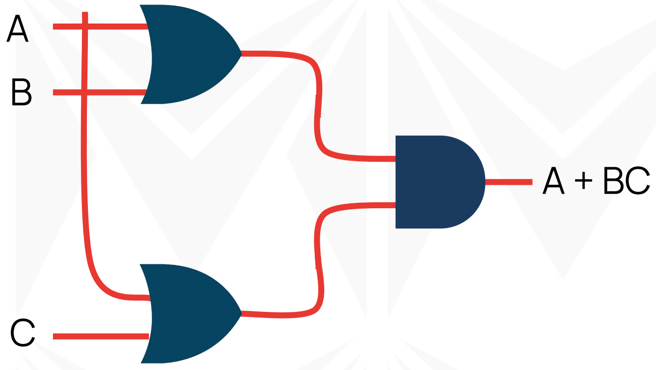
$$A + A'B = A + B$$



A	A'	B	A'B	Output
0	1	0	0	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	1

RULE 12: (A + B) (A + C) = A + BC

A	B	C	X = A+B	Y = A+C	X.Y	A+BC
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	1	0	0	0
0	1	1	1	1	1	1
1	0	0	1	1	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1



SIMPLIFICATION EXAMPLES

Using the Boolean algebra techniques discussed above, simplify the expressions below:

1. $AB + A(B + C) + B(B + C)$

Solution

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

2. $(A\overline{B}\,(C + BD) + \overline{A}\,\overline{B}\,)\,C$

<u>Solution</u>

$(A\,\overline{B}C \;+\; A\overline{B}BD + \overline{A}\,\overline{B})C$ (Distributive law applied)

$A\,\overline{B}CC \;+\; A\overline{B}BDC + \overline{A}\,\overline{B}C$ (Distributive law applied)

$A\,\overline{B}C \;+\; \overline{A}\,\overline{B}C$ (Recall, $CC = C$ and $\overline{B}B = 0$)

$\overline{B}C(A + \overline{A})$ (Recall, $A + \overline{A} = 1$)

$\overline{B}C \cdot 1$ (Recall, $\overline{B}C \cdot 1 = \overline{B}C$)

$\overline{B}C$

MID-LESSON ASSESSMENT

Simplify the following boolean expressions

1. $[A\overline{B}(C + BD) + \overline{A}\,\overline{B}]C$

2. $\overline{A}BC + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}C + ABC$

3. $AB\overline{C} + \overline{A}\,\overline{B}C + \overline{A}BC + \overline{A}\,\overline{B}\,\overline{C}$

4. Apply DeMorgan's theorems to the following:

$$\overline{\overline{\overline{(ABC)}\,\overline{(EFG)}} + \overline{\overline{(HIJ)}\,\overline{(KLM)}}}$$

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

DEMORGAN'S THEOREM

1. $\overline{A + B} = \overline{A}\,\overline{B}$

A	A'	B	B'	A'B'	A + B	(A+B)'
0	1	0	1	1	0	1
0	1	1	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	0	1	0

DEMORGAN'S THEOREM

2. $\overline{AB} = \overline{A} + \overline{B}$

A	A'	B	B'	A' + B'	AB	(AB)'
0	1	0	1	1	0	1
0	1	1	0	1	0	1
1	0	0	1	1	0	1
1	0	1	0	0	1	0

EXAMPLES

Using DeMorgan's theorem and other algebra techniques discussed above, simplify the expression below:

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

EXAMPLES

Using the Boolean algebra techniques discussed above, simplify the expression below:

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

KARNAUGH MAPS (K-MAPS) INTRODUCTION

A Karnaugh map is a graphical tool for simplifying Boolean expressions.

It is a visual representation of truth tables in the form of an array. It presents all the possible values of input variables and the resulting output for each input combination.

K-Maps are represented in cells. The number of cells is equal to the total number of input variables combination; for 3 variables, number of cells = 2^3

While they can be used to represent expressions with 2, 3, 4 or 5 variables, they are mostly used to represent expressions with 3 or 4 variables.

For ease of representation, $\bar{A} \equiv A'$

K-MAPS STRUCTURE

3-variable K-Map:

AB \ C	0	1
00	A'B'C'	A'B'C
01	A'BC'	A'BC
11	ABC'	ABC
10	AB'C'	AB'C

OR

AB \ C	0	1
00	000	001
01	010	011
11	110	111
10	100	101

K-MAPS STEPS

When a K-Map is fully representing a boolean expression, there will be a number of 1s equal to the number of product terms in the expression.

The mapping process can be summarised as follows:

Step 1: Reduce the expression to a sum of products, and determine the binary value of each product term in the expression.

Step 2: As each product term is evaluated, place a 1 on the Karnaugh map in the cell representing the product term combination.

Step 3: Group evenly adjacent 1s. That is, 2, 4, 8,...

K-MAPS STRUCTURE

4-variable K-Map:

AB \ CD	00	01	11	10
00	0000	0001	0011	0010
01	0100	0101	0111	0110
11	1100	1101	1111	1110
10	1000	1001	1011	1010

AB \ CD	00	01	11	10
00	A'B'C'D'	A'B'C'D	A'B'CD	A'B'CD'
01	A'BC'D'	A'BC'D	A'BCD	A'BCD'
11	ABC'D'	ABC'D	ABCD	ABCD'
10	AB'C'D'	AB'C'D	AB'CD	AB'CD'

EXAMPLES

A

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

B

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

C

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

EXAMPLES

AB

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

AC

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

BC

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

EXAMPLES

A + B

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

A’B

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

B + C’

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

SIMPLIFICATION EXAMPLES

1. $AB + A(B + C) + B(B + C)$

   $AB + AB + AC + BB + BC$

   $AB + AC + B + BC$

A	B	C	Out
0	0	0	
0	0	1	
0	1	0	1
0	1	1	1
1	0	0	
1	0	1	1
1	1	0	1
1	1	1	1

K-MAP REPRESENTATION

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

$= B + AC$

EXAMPLES

2. Map the following standard Sum of Product expression on a Karnaugh map.

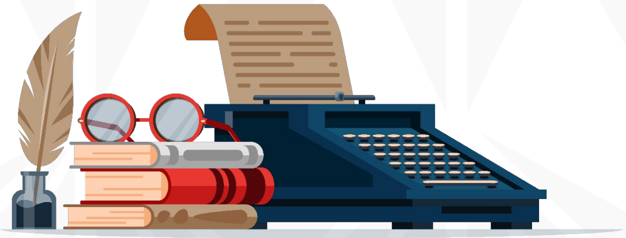
A'B'C + A'BC' + ABC' + ABC

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

= BC' + AB + A'B'C

SUMMARY AND APPLICATIONS

- Complex Logic circuits can be represented with boolean algebra, and simplified by following the rules of boolean algebra minimisation.
- Boolean algebra minimisation shows that complex circuits can sometimes be represented with a simple combination of a few gates.
- Karnaugh maps are used to represent sum boolean expressions as 2-dimensional array of 1s and 0s.
- K-Maps help simplify boolean expressions faster.



REFERENCES

1. Hill, F. J., Peterson, G. R. (2013). *Introduction to Switching Theory and Logic Design* (3rd ed.). John Wiley & Sons Inc.
2. Floyd, T. L. (2013). *Digital Fundamentals: A Systems Approach*. Pearson.

Thank You