CLO1-Week3-Simplify Boolean Expression Using Algebra Axiom
Outline

• Understand the Axiom of Boolean Algebra
• Knowing the Boolean Algebra Theorem
• Understand the Concept of Duality in Boolean Algebra
• Understand how to solve Boolean Algebra Expression by Truth Table
• Understand how to simplify Boolean Expression by Using Boolean Algebra
Why should be “Simple”?

• Digital computers contain circuits that implement Boolean functions.
• The simpler that we can make a Boolean function, the smaller the circuit that will result.
  – Simpler circuits are cheaper to build, consume less power, and run faster than complex circuits.
• With this in mind, we always want to reduce our Boolean functions to their simplest form.
• There are a number of Boolean identities that help us to do this.
Axiom in Boolean Algebra

1a. 0 0 = 0
1b. 1 + 1 = 1

2a. 1 1 = 1
2b. 0 + 0 = 0

3a. 0 1 = 1 0 = 0
3b. 1 + 0 = 0 + 1 = 1

4a. If x = 0, then \( \overline{x} = 1 \)
4b. If x = 1, then \( \overline{x} = 0 \)
Single Variable Theorem

5a. \( x \cdot 0 = 0 \)
5b. \( x + 1 = 1 \)

6a. \( x \cdot 1 = x \)
6b. \( x + 0 = x \)

7a. \( x \cdot x = x \)
7b. \( x + x = x \)

8a. \( x \cdot \overline{x} = 0 \)
8b. \( x + \overline{x} = 1 \)

9. \( \overline{\overline{x}} = x \)

- Null Law
- Identity Law
- Idempotent Law
- Inverse Law
Duality Concept

- Duality of Boolean Expression is obtained by replacing the AND operator with the equivalent of an OR operator, and the operator OR by equivalent operators AND, bit '0' with the equivalent bit '1' and bit '1' to bits '0'
- This principle will be useful in the manipulation of Boolean algebra in a logic circuit simplification
- To simplify the Boolean Expression, you need to use Boolean Algebra Law and also its Characteristic
Two or Three Variable-Characteristic in Algebra

10a. \( x \cdot y = y \cdot x \)  
10b. \( x + y = y + x \) \}
      \text{Commutative}

11a. \( x \cdot (y \cdot z) = (x \cdot y) \cdot z \)  
11b. \( x + (y + z) = (x + y) + z \) \}
      \text{Associative}

12a. \( x \cdot (y + z) = x \cdot y + x \cdot z \)  
12b. \( x + (y + z) = (x + y) + (x + z) \) \}
      \text{Distributive}

13a. \( x + x \cdot y = x \)  
13b. \( x \cdot (x + y) = x \) \}
      \text{Absorptive}
Two or Three Variable-Characteristic in Algebra

14a. \( x \cdot y + x \cdot \bar{y} = x \) 
14b. \( (x + y) \cdot (x + \bar{y}) = x \) 

\[ \text{Combining} \]

15a. \( \bar{x} \cdot \bar{y} = \bar{x} + \bar{y} \) 
15b. \( \bar{x} + \bar{y} = \bar{x} \cdot \bar{y} \) 

\[ \text{Theorem of DeMorgan} \]

\[ \bar{X} = X', \quad \bar{X} = (X')' \] 

\[ \text{Symbol Equivalence} \]
For Example

By Using Boolean Theorem, Proof that:

1. \( X \cdot Y + X \cdot Y' = X \)
2. \( X + X \cdot Y = X \)
Examples: Proof Theorem of Algebra Boole By Using Axiom and Single Variable Theorem:

| 1. Proof Theorem Distributive | :  | X • Y + X • Y' = X |
| : |  | X • Y + X • Y' = X • (Y + Y') |
|  | Complement : | X • (Y + Y') = X • (1) |
|  | Identity : | X • (1) = X |
| 2. Proof Theorem Identity | :  | X + X • Y = X |
| : |  | X + X • Y = X • 1 + X • Y |
|  | Distributive | X • 1 + X • Y = X • (1 + Y) |
|  | Identity | X • (1 + Y) = X • (1) |
|  | Identity | X • (1) = X |
THEOREM of DeMorgan

\[(X + Y)' = X' \cdot Y'\]

\[\text{NOR Gate equivalent with AND Gate with complemented input}\]

\[(X \cdot Y)' = X' + Y'\]

\[\text{NAND Gate equivalent with OR Gate with complemented input}\]

DeMorgan theorem can be used to convert statement AND/OR be a statement OR/AND

Example:

\[Z = A' B' C + A' B C + A B' C + A B C'\]

\[Z' = (A + B + C') \cdot (A + B' + C') \cdot (A' + B + C') \cdot (A' + B' + C)\]
A Truth Table can be expressed as a Boolean function

A truth table can be expressed in two forms that are equivalent Boolean function

The functions of the equation obtained from a truth table called a canonical form.

*Sum of Products Form*

Also called disjunctive normal form, an expansion

Part minterm truth table

\[
\begin{array}{cccccccc}
A & B & C & F & F' \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 0 \\
\end{array}
\]

\[
F = A' B' C' + A B' C' + A B C' + A B C
\]

\[
F' = A' B' C' + A' B' C + A B C'
\]
CANONICAL SUM of PRODUCT (SoP)

Sum of Products

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Minterms</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>( \overline{A} \overline{B} \overline{C} = m_0 )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>( \overline{A} \overline{B} C = m_1 )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>( A \overline{B} \overline{C} = m_2 )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>( A \overline{B} C = m_3 )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>( A B \overline{C} = m_4 )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>( A B C = m_5 )</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>( A \overline{B} \overline{C} = m_6 )</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>( A B \overline{C} = m_7 )</td>
</tr>
</tbody>
</table>

\( F \) in form of SoP:

\[
F(A,B,C) = \Sigma m(3,4,5,6,7) = m_2 + m_3 + m_4 + m_6 + m_7
\]

\[
= A^3 B C + A^5 B C' + A B^7 C + A B C' + A B C
\]

Simplify by Using Boolean Algebra Theorem:

\[
F = A B' (C + C') + A' B C + A B (C' + C)
\]

\[
= A B' + A' B C + A B
\]

\[
= A (B' + B) + A' B C
\]

\[
= A + A' B C
\]

\[
= A + B C
\]

Realization Results Simplification

SoP expression
**Product of Sums / Conjunctive Normal Form / Maxterm Expansion**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Maxterms</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>( A + B + C = M_0 )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>( A + B + C = M_1 )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>( A + B + C = M_2 )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>( A + B + C = M_3 )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>( A + B + C = M_4 )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>( A + B + C = M_5 )</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>( A + B + C = M_6 )</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>( A + B + C = M_7 )</td>
</tr>
</tbody>
</table>

Boolean function PoS readings based on the truth table:

\[
F(A,B,C) = \Pi M(0,1,2)
\]

\[
= (A + B + C) (A + B + C') (A + B' + C)
\]

\[
F'(A,B,C) = \Pi M(3,4,5,6,7)
\]

\[
= (A + B' + C') (A' + B + C) (A' + B + C') (A' + B' + C) (A' + B' + C')
\]
COMPARISON OF IMPLEMENTATION

Canonical Sum of Products

Minimized Sum of Products

Canonical Products of Sums
learning