Boolean Operator Precedence

- The order of evaluation in Boolean Expressions is:
  - Parentheses
  - NOT
  - AND
  - OR

- Because AND takes precedence over OR, parentheses must be placed around the OR operator more frequently.
Useful Theorems

- $x \cdot y + \overline{x} \cdot y = y$  $(x + y)(\overline{x} + y) = y$  Minimization
- $x + x \cdot y = x$  $x \cdot (x + y) = x$  Absorption
- $x + \overline{x} \cdot y = x + y$  $x \cdot (x + y) = x \cdot y$  Simplification
- $x \cdot y + \overline{x} \cdot z + y \cdot z = x \cdot y + \overline{x} \cdot z$  Consensus
  $(x + y) \cdot (\overline{x} + z) \cdot (y + z) = (x + y) \cdot (\overline{x} + z)$
- $x + y = \overline{x} \cdot \overline{y}$  $x \cdot \overline{y} = \overline{x} + \overline{y}$  DeMorgan's Laws

Proof of Simplification

$x \cdot y + \overline{x} \cdot y = y$  $(x + y)(\overline{x} + y) = y$
Proof of Consensus

\[ x \cdot y + \overline{x} \cdot z + y \cdot z = x \cdot y + \overline{x} \cdot z \]

Proof of DeMorgan’s Law

\[ \overline{x + y} = \overline{x} \cdot \overline{y} \quad \overline{x \cdot y} = \overline{x} + \overline{y} \]
Boolean Function Evaluation

<table>
<thead>
<tr>
<th>F1</th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
</tr>
</thead>
<tbody>
<tr>
<td>x y z</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>x + y z</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>x y z + x y z + x y</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>x y + x z</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Expression Simplification

- Simplify to contain the smallest number of literals (complemented and uncomplemented variables):
  \[A B + \overline{A} C D + \overline{A} B D + A C \overline{D} + A B C D = \]
Complementing Functions

- Use DeMorgan's Theorem to complement a function:
  1. Interchange AND and OR operators
  2. Complement each literal

- Example: Complement \( xyz + xyz \)

Canonical Forms

- It is useful to specify Boolean functions in a form that:
  - Allows comparison for equality.
  - Has a correspondence the truth tables

- Canonical Forms are in common usage:
  - Sum of Minterms (SOM)
  - Product of Maxterms (POM)
Minterms

- **Minterms** are AND terms with every variable in true or complemented form.
- Given that each binary variable may appear normal (e.g., \(x\)) or complemented (e.g., \(\overline{x}\)), there are \(2^n\) minterms for \(n\) variables.
- **EXAMPLE:** Two variables, combined with an AND operator, \(X \cdot Y\) have \(2^2 = 4\) combinations:
  - \(XY\) (both normal)
  - \(\overline{X}Y\) (\(X\) normal, \(Y\) complemented)
  - \(X\overline{Y}\) (\(X\) complemented, \(Y\) normal)
  - \(\overline{X}\overline{Y}\) (both complemented)
- Thus there are **four minterms** of two variables.

Maxterms

- **Maxterms** are OR terms with every variable in true or complemented form.
- Given that each binary variable may appear normal (e.g., \(x\)) or complemented (e.g., \(\overline{x}\)), there are \(2^n\) maxterms for \(n\) variables.
- **Two variables**, combined with an OR operator, \(X + Y\), have \(2^2 = 4\) combinations:
  - \(X + Y\) (both normal)
  - \(X + \overline{Y}\) (\(x\) normal, \(y\) complemented)
  - \(\overline{X} + Y\) (\(x\) complemented, \(y\) normal)
  - \(\overline{X} + \overline{Y}\) (both complemented)
Maxterms and Minterms

- **Examples: Two variable minterms and maxterms.**

<table>
<thead>
<tr>
<th>Index</th>
<th>Maxterm</th>
<th>Minterm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x + y</td>
<td>x \ y</td>
</tr>
<tr>
<td>1</td>
<td>x + y</td>
<td>x \ y</td>
</tr>
<tr>
<td>2</td>
<td>x + y</td>
<td>x \ y</td>
</tr>
<tr>
<td>3</td>
<td>x + y</td>
<td>x \ y</td>
</tr>
</tbody>
</table>

- The index above is important for describing which variables in the terms are true and which are complemented.

Order of variables

- Minterms and maxterms are designated with a subscript
- The subscript is a number, corresponding to a binary pattern
- The bits in the pattern represent the complemented or normal state of each variable listed in a given order.
- All variables will be present in a minterm or maxterm and will be listed in the **same order** (usually alphabetically)
- **Example:** For variables a, b, c:
  - Maxterms: (a + b + c), (a + b + c)
  - Minterms: a b c, a b c, a b c
  - Terms: (b + a + c), a \ b c, and (c + b + a) are NOT in particular order.
  - Terms: (a + c), b c, and (a + b) do not contain all variables
The index for the minterm or maxterm, expressed as a binary number, is used to determine whether the variable is shown in the true form or complemented form.

For Minterms:
- “1” means the variable is “Not Complemented” and
- “0” means the variable is “Complemented”.

For Maxterms:
- “0” means the variable is “Not Complemented” and
- “1” means the variable is “Complemented”.

Example: (for three variables)
Assume the variables are called X, Y, and Z.
The standard order is X, then Y, then Z.
The index 0 (base 10) = 000 (base 2 to three digits) so all three variables are complemented for minterm 0 (X, Y, Z) and no variables are complemented for maxterm 0 (X, Y, Z).

Minterm 0, called m_0 is XYZ.
Maxterm 0, called M_0 is (X + Y + Z).