How would we prove that all three representations are equivalent?

Outline
- Review of three representations for combinational logic:
  - truth tables,
  - graphical (logic gates), and
  - algebraic equations
- Relationship among the three
- Adder example
- Formal description of Boolean algebra
- Laws of Boolean algebra

CL Block Example #1

Boolean Equation:

\[ y_0 = (x_0 \text{ AND } x_1) \text{ OR } (x_0 \text{ AND } x_2) \]

Truth Table Description:

<table>
<thead>
<tr>
<th>x0</th>
<th>x1</th>
<th>x2</th>
<th>y0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Gate Representation:

\[ y_0 = x_0 \cdot (x_1 \oplus x_2) \]

How do we convert from one to the other?

Theorem: Any Boolean function that can be expressed as a truth table can be written as an expression in Boolean Algebra using AND, OR, NOT.
In general: 2n rows for n inputs. 256 rows.

Is there a more efficient (compact) way to specify this function?

Spring 2002 EEC5010 - Lec6-bool1 Page 8
Logic Functions

A Boolean expression is valid if it satisfies the following axioms:

1. Commutative Law: $0 + 1 = 1 + 0$
2. Associative Law: $x + (y + z) = (x + y) + z$
3. Distributive Law: $x(y + z) = xy + xz$
4. Idempotent Law: $x + x = x$, $x \cdot x = x$
5. Involution Law: $(x')' = x$
6. Complement Law: $x + x' = 1$, $x \cdot x' = 0$
7. Complement Theorem: $x + xy = x$, $x(1 + y) = x$
8. Consensus Theorem: $x + y + z = x + y$ if $x'z = y'$
9. DeMorgan's Law: $(x + y)' = x' \cdot y'$, $(x \cdot y)' = x' + y'$

Other logic functions of 2 variables $(x, y)$

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$f_0$</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
<th>$f_4$</th>
<th>$f_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</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>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Look at NOR and NAND:

- Theorem: Any Boolean function that can be expressed as a truth table can be expressed using NAND and NOR.
- Proof sketch:
  - $\overline{x} = \overline{x}$
  - $x = \overline{x}$
- How would you show that NAND or NOR is sufficient?

Laws of Boolean Algebra

Duality: A dual of a Boolean expression is derived by interchanging OR and AND operations, and 0s and 1s (literals are left unchanged).

- $x + y = (x')'$
- $x \cdot y = (x')'$

Any law that is true for an expression is also true for its dual.

- Operations with 0 and 1:
  - $x + 0 = x$
  - $x \cdot 1 = x$
- Idempotent Law:
  - $x + x = x$
- Involution Law:
  - $(x')' = x$
- Complement Law:
  - $x + x' = 1$
- Commutative Law:
  - $x + y = y + x$

Proving Theorems via axioms of Boolean Algebra

Ex: prove the theorem: $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$

- Distributive Law:
  - $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
- Associative Law:
  - $(x \cdot y) \cdot z = x \cdot (y \cdot z)$

DeMorgan's Law

- $(x + y)' = x' \cdot y'$
- $(x \cdot y)' = x' + y'$

DeMorgan's Law can be used to convert AND/OR expressions to OR/AND expressions.

Example: $x' + y' + z = (x'y')z'$
Algebraic Simplification

Ex: full adder (FA) carry out function

\[ C_{ou} = a'b'c + ab'c + abc' + abc \]

\[ \begin{align*}
  &= a'b'c + ab'c + abc' + abc \\
  &= (a' + a)bc + ab'c + abc' + abc \\
  &= bc + ab'c + abc' + abc \\
  &= bc + a(b' + b)c \\
  &= bc + ac + ab \\
\end{align*} \]