LECTURE #2: Intro to Logic Design and Boolean Algebra
EEL 3701: Digital Logic and Computer Systems
Based on lecture notes by Dr. Eric M. Schwartz

Logically Analyzing a Paragraph:

When I’m hungry and I don’t eat, I get a headache. I also get a headache when I repeatedly bang my head against the wall without wearing a helmet.

Output/Result: Getting a headache (Headache)

Inputs/Conditions:
1st Sentence:
- I’m hungry (Hungry)
- I don’t eat (NoEat)

2nd Sentence:
- I repeatedly band my head (BangHead)
- Without wearing a helmet (NoHelmet)

Equation: Headache = (Hungry AND NoEat) OR (BangHead AND NoHelmet)

There are 2 possible values for each variable: True or False
Logic abbreviations:
  True: T or 1
  False: F or 0

---

Logical **AND** Operator:

**Representation:**
- \( Z = A \times B \)
- \( Z = A \ast B \)
- \( Z = A \bullet B \)
- \( Z = A \land B \)
- \( Z = AB \)

**Logic Gate Symbol:**

[Diagram of logic gate]

**Truth Table:**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>( Z = AB )</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

AKA: Conjunction
Logical **OR** Operator:

**Representation:**

\[
Z = A + B \\
Z = A \lor B
\]

**Truth Table:**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Z = A+B</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

**Logic Gate Symbol:**

AKA: Disjunction

**Original equation becomes:** Headache = (Hungry * NoEat) + (BangHead * NoHelmet)

Draw the equation using the logic gates AND and OR:

---

Logical **NOT** Operator:

**Representation:**

\[
Z = /A \\
Z = \overline{A} \\
Z = A' \\
Z = \sim A
\]

**Truth Table:**

<table>
<thead>
<tr>
<th>A</th>
<th>Z = ~A</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

**Logic Gate Symbol:**

AKA: Complement (operation)  
AKA: Inverter (device)  
AKA: Level Shifter (device)

**Original equation becomes:** Headache = (Hungry * ~Eat) + (BangHead * ~Helmet)

Redraw the equation with NOT gates:

---
**Boolean Algebra:**
- Developed by George Boole in 1847
- Applied to switching circuits (two-state devices) by Claude Shannon in 1939
- Is the basic mathematics used for designing digital systems

For our purposes, variables can only have one of two values.
- Known as switching algebra
- Examples of variables: A, B, X, Y
  - Represent system Inputs and Outputs
  - (See paragraph example above)
- Variables can assume the value 0 or 1
  - 0 and 1 are not numbers, they are states
  - States may represent any two states
    - Example: High and Low voltage
    - Example: True and False

Basic Boolean operations:
- AND (discussed above)
- OR (discussed above)
- NOT (discussed above)

Order of operations in Boolean expressions:
1) NOT (on individual variables)
2) AND
3) OR
4) NOT (on entire expressions)

Note: Parentheses are used to modify priority.

Definitions:
- Terms: The objects of the universe of discourse
  - i.e. constants, variables, functions
- Literal: A variable or its complement
  - Ex: \( Z = ABC + AB' + A'BC' + B'C' \)
    - Has 4 variables (A, B, C, Z)
    - Has 11 literals
- Equivalence: Two Boolean expressions are equivalent iff they have the same values for every possible combination of the variables. Since a Truth Table is an exhaustive tabulation of the input variables, identical columns imply equivalent expressions.
Two more logical operators for completeness:

**Logical XOR/EOR (Exclusive Or) Operator:**

**Representation:**

\[ Z = A \oplus B \]

**Truth Table:**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Z = A \oplus B</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

**Logic Gate Symbol:**

AKA: Exclusive Disjunction

**Logical XNOR/EQUIV (Exclusive Nor/Equivalence) Operator:**

**Representation:**

\[ Z = A \odot B \]
\[ Z = \overline{A \oplus B} \]

**Truth Table:**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Z = A \odot B</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

**Logic Gate Symbol:**

AKA: Exclusive Conjunction

Note: Both of these can be created from basic logic gates.

\[ A \oplus B = \overline{A} \overline{B} + AB \]
\[ A \odot B = AB + \overline{A} \overline{B} = \overline{A \overline{B} + AB} \]