Basic Logic Gates

Truth Table

- The most basic representation of a logic function
- Lists the output for all possible input combinations
- How many rows of the truth table needed?

Truth Table: Inverter

- Inverted signals are denoted with an overbar
- Or with a prime symbol
  - $A'$

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>A B ...</td>
<td>X Y ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Y = A'</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Truth Table: AND Gate

- The result of an AND operation is 1 if and only if all inputs are 1
- Depict AND by the multiplication symbol
  - $A \cdot B$
- Or by lumping the signals together
  - $A B$
- We don't really build these gates...

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>$Y = A \cdot B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Truth Table: OR Gate

- The result of an OR operation is 1 if and only if any inputs are 1
- Depict OR by the addition symbol
  - $A + B$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>$Y = A + B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

About the Little Circle...

- The little circle is what inverts

Sum of Products

- How do you get from a truth table to a logic expression?
- Sum of products is standard way of synthesizing simple circuits
- Procedure:
  1. Find the rows with the ‘1’ output
  2. Write the product-form expression for the inputs in that row (0=inverted, 1=normal)
  3. Combine the products in step 2 into a sum (OR the results of step 2)
**Sum of Products**

1. Find the rows with the ‘1’ output
2. Write the product-form expression for the inputs in that row (0=inverted, 1=normal)
3. Combine the products in step 2 into a sum (OR the results of step 2)

\[ \overline{A} \cdot B + AB = Y \]

**De Morgan’s Laws**

- "Break the line, change the sign"
- Two laws:
  - \[ A' + B' = (AB)' \]
  - \[ A' \cdot B' = (A+B)' \]

\[ \overline{A + B} = \overline{A} \cdot \overline{B} \]

\[ \overline{A \cdot B} = \overline{A} + \overline{B} \]
**De Morgan's Laws**

\[ A + B = \overline{A \cdot B} \]

- In other words...
  \[ \overline{A} + \overline{B} = \overline{A \cdot B} \]
- Push the bubbles through!

**De Morgan's Laws and SOP**

\[ \overline{A} \cdot \overline{B} + \overline{A} \cdot \overline{B} = 1 \]

- Generate equivalent circuits
  - NAND/NAND
  - NOR/NOR
- We prefer NAND/NAND circuits
  - Same transistor count as NOR
  - NANDs are faster

**Masking**

- Want to look only at certain bits of a binary word
- Use a mask to remove the uninteresting bits
- Example:
  \[ A = \overline{1011} \ 0110 \]
  \[ B = 0000 \ 1111 \]
  \[ C = 0000 \ 0110 \]
  \[ A = \overline{1011} \ 0110 \]
  \[ B = 0110 \ 0000 \]
  \[ C = 1111 \ 0110 \]

**Axioms of Boolean Algebra**

- \[ 0 \cdot 0 = 0 \]
- \[ 1 + 1 = 1 \]
- \[ 1 \cdot 1 = 1 \]
- \[ 0 + 0 = 0 \]
- \[ 0 \cdot 1 = 1 \cdot 0 = 0 \]
- \[ 1 + 0 = 0 + 1 = 1 \]
- \[ 1 + 0 = 0 + 1 = 1 \]
- \[ \text{if} \ x = 0 \ \text{then} \ x' = 1 \]
- \[ \text{if} \ x = 1 \ \text{then} \ x' = 0 \]
Single-Variable Theorems

- $x \cdot 0 = 0$
- $x + 1 = 1$
- $x \cdot 1 = x$
- $x + 0 = x$
- $x \cdot x = x$
- $x + x = 1$
- $(x')' = \overline{\overline{x}} = x$

Properties of Boolean Algebra

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

Properties of Boolean Algebra

- Absorption
  - $x + x \cdot y = x$
  - $x \cdot (x + y) = x$
- Combining
  - $x \cdot y + x \cdot y' = x$
  - $(x + y) \cdot (x + y') = x$
- De Morgan's Laws
  - $(x \cdot y)' = x + y$
  - $(x + y)' = x \cdot y$

Logic Minimization

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Y</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>