LOGICAL COMPLETENESS AND 2-LEVEL DESIGN

So far

NOT, AND, OR are logically complete: we can implement any Boolean function with them.

Question: are there any other sets of logically complete functions?
**NAND function**

Truth table:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>(AB)'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</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>0</td>
</tr>
</tbody>
</table>

Symbol:

To show NAND function is **logically complete**, we will show it can implement NOT, AND, OR.
* NOT function:

* AND function:
\texttt{OR function:}
* NANO implementation of SOP form

Example:

\[ A \quad B \quad C \quad F = ? \]

Using NANO implementation for AND, OR:

Using involution property we can simplify:
NOR function

Truth table:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>(A+B)′</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</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>0</td>
</tr>
</tbody>
</table>

* NOT function: \( x′ = (x+x)′ \) ( ? )
\* \text{AND function:} \quad x \cdot y = \left[ (x \cdot y) \right]' \\
= \left[ x' + y' \right]' \\

\* \text{OR function:} \quad x + y = \left[ (x + y) \right]'
* NOR implementation of POS form

Example:

\[ \begin{array}{c}
\text{x} \\
\text{y} \\
\text{z}
\end{array} \quad \text{G} = ? \]
A circuit is 2-level if there are at most 2 gates between any input and the output.

Every function can be represented in SOP form, thus it can be implemented with a 2-level \((\text{AND-OR})\) network or 2-level \(\text{NAND}\) network.

Similarly, every function can be represented in POS form, so it can be implemented with a 2-level \((\text{OR-AND})\) network or a 2-level \(\text{NOR}\) network.
**Exclusive-OR (XOR)**

For a 2-input XOR, output is 1 if inputs are **different**.

For an n-input XOR, output is 1 if inputs have **odd number of 1's**.

XOR can be used as **controlled inverter**:

\[ X \quad \text{XOR} \quad 1 \quad \text{Output} \]

\[ X \quad \text{XOR} \quad 0 \quad \text{Output} \]