Exclusive OR/ Exclusive NOR

- The exclusive OR (XOR) function is an important Boolean function used extensively in logic circuits. Uses for the XOR gate include:
  - Adders/subtractors
  - Parity generators/checkers
  - Signature analyzers
  - Pseudo-random sequence generators

Definitions
- The XOR function is: \( X \oplus Y = XY + \overline{X} \overline{Y} \)
- The exclusive NOR (XNOR) function, otherwise known as Equivalence is: \( X \odot Y = XY + XY \)

Tables for XOR/XNOR

- Operator Rules: XOR  
  \[
  \begin{array}{c|cc}
    X & Y & X \oplus Y \\
    
    0 & 0 & 0 \\
    0 & 1 & 1 \\
    1 & 0 & 1 \\
    1 & 1 & 0 \\
  \end{array}
  \]

- XOR Implementation (Cont.)

- The AND-OR implementation is the SOP form for the XOR function:
  \[
  X \odot Y = X Y + \overline{X} \overline{Y}
  \]
- The multiple-level NAND implementation can be derived by combining inversions as follows:
  \[
  X \odot Y = (X + \overline{Y}) \overline{(X + Y)} = X(X + Y) + \overline{X} \overline{Y}
  \]
Odd Function

- The modulo 2 sum function for n variables
  - Contains an odd number of 1's, and
  - Is therefore called the odd function.
- The inverse of the modulo 2 sum function for n variables
  - Contains an even number of 1's, and
  - Is therefore called the even function.
- Implementation of even and odd functions for greater than 4 variables as a single gate is difficult, so “trees” of 2 to 4 input XOR or XNORs are used.

Parity Generators/Checkers

- A parity tree for n data bits generates a parity bit that is appended to the data bits to form an n + 1-bit codeword
  - Example: 3-bit even parity generator
- A parity tree for n + 1 bits checks the codeword for correct parity:
  - C=0 if the codeword parity is correct
  - C=1 if the codeword parity is correct
  - Example: 4-bit even parity checker

Example: Odd Function Implementation

- Three-Input Odd Function:
- Four Input Odd Function:

Product terms of Odd and Even

- For an n-bit even or odd function, there will be \( (2^n)/2 \) or \( 2^{n-1} \) product terms of n variables (minterms)!

Integrated Circuit Parameters

- Logic device families are characterized by the following parameters:
  - Fan-in – the number of inputs available on a gate
  - Fan-out – the number of inputs the output of a gate can drive
  - Logic Levels – the signal value ranges defining 1 and 0
  - Propagation Delay – The time for an input signal change to propagate to an output
  - Noise Margin – the amount of noise a logic signal can tolerate without error
  - Power Supply – the voltage(s) required to allow the circuit to operate
  - Power Dissipation – the amount of power a circuit consumes

Propagation Delay

- Propagation delay is the time for a change in the input of a gate to propagate to the output.
- Delay is usually measured from the 50% of the logic level voltage reference points.
- High-to-low \( (t_{PHL}) \) and low-to-high \( (t_{PLH}) \) output signal changes may have different propagation delays.
- High-to-low \( (t_{H}) \) and low-to-high \( (t_{L}) \) transitions are defined with respect to the output, not the input.
- An HI input transition causes:
  - an HI output transition if the gate inverts and
  - an HL output transition if the gate does not invert.
Propagation Delay Example

- What is the delay for:
  - a string of inverters?
  - a string of buffers?

Positive and Negative Logic

- The same physical gate has different logical meanings depending on interpretation of the signal levels.
  - Positive Logic
    - Logic 1 is set to HIGH (more positive) signal levels
    - Logic 0 is set to LOW (less positive) signal levels
  - Negative Logic
    - Logic 1 is set to LOW (more negative) signal levels
    - Logic 0 is set to HIGH (less negative) signal levels
- A gate which implements a Positive Logic AND function will implement a Negative Logic OR function.

Positive and Negative Logic (Cont.)

- Rearranging the negative logic terms to be in proper function table order we get:

Positive Logic | Negative Logic
---|---
0 0 | 0 0 |
0 1 | 0 1 |
1 0 | 1 0 |
1 1 | 1 1 |

Positive logic "OR", Negative Logic "AND"

Logic Symbol Conventions

- Symbols:

Positive Logic: $X \land Y \land Z$
Negative Logic: $X \lor Y \lor Z$

Positive and Negative Logic (Cont.)

- Given this signal level table:

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>X Y</td>
<td>L L</td>
</tr>
<tr>
<td>L L</td>
<td>L L</td>
</tr>
<tr>
<td>L H</td>
<td>H L</td>
</tr>
<tr>
<td>H L</td>
<td>H H</td>
</tr>
</tbody>
</table>

- What logic function is implemented?

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>X Y</td>
<td>0 0</td>
</tr>
<tr>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>0 1</td>
<td>0 1</td>
</tr>
<tr>
<td>1 0</td>
<td>0 0</td>
</tr>
<tr>
<td>1 1</td>
<td>0 0</td>
</tr>
</tbody>
</table>