XOR Revisited

XOR is also called modulo-2 addition.

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

\[ A \oplus B = 1 \text{ only when there are an odd number of } 1\text{'s in } (A,B). \text{ The same is true for } A \oplus B \oplus C \text{ also.} \]

\[ 1 \oplus A = \bar{A} \quad \text{Why?} \]

\[ 0 \oplus A = A \]
Logic Design Examples

Half Adder

$$S = A \oplus B$$

$$C = A \cdot B$$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>S</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</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>
</tbody>
</table>
Full Adder

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = A.B + B.C_{in} + A.C_{in}$$

1. Design a full adder using two half-adders (and a few gates if necessary).

2. Can you design a 1-bit subtractor?
Decoders

A typical decoder has \( n \) inputs and \( 2^n \) outputs.

A 2-to-4 decoder and its truth table

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

A 2-to-4 decoder and its truth table

\( D3 = A.B \)
\( D2 = A\overline{B} \)
\( D1 = \overline{A.B} \)
\( D0 = \overline{A.B} \)

Draw the circuit of this decoder.

The decoder works per specs when (Enable = 1). When Enable = 0, all the outputs are 0.

Exercise. Design a 3-to-8 decoder.

Question. Where are decoders used?

Can you design a 2-4 decoder using 1-2 decoders?
Encoders

A typical encoder has $2^n$ inputs and $n$ outputs.

\[
\begin{array}{cccc}
D_0 & D_1 & D_2 & D_3 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{array}
\begin{array}{cc}
A & B \\
0 & 0 \\
0 & 1 \\
1 & 0 \\
1 & 1 \\
\end{array}
\]

A 4-to-2 encoder and its truth table

\[
\begin{align*}
A &= D_1 + D_3 \\
B &= D_2 + D_3
\end{align*}
\]
**Multiplexor**

It is a *many-to-one switch*, also called a *selector*.

\[ A \xrightarrow{S = 0} F = A \]
\[ B \xrightarrow{S = 1} F = B \]

### Specifications of the mux

A 2-to-1 mux

\[ F = \overline{S}. A + S. B \]

**Exercise.** Design a 4-to-1 multiplexor using two 2-to-1 multiplexors.
Demultiplexors

A demux is a one-to-many switch.

A 1-to-2 demux and its specification

So, $X = \overline{S}. A$, and $Y = S. A$

Exercise. Design a 1-4 demux using 1-2 demux.