Activity

Implement an inverting 2-input multiplexer.

Assume that S and \( \overline{S} \) are available.

\[
\begin{align*}
 f &= (S \cdot D1) + (\overline{S} \cdot DO) \\
 PD &= (S \cdot D1) + (\overline{S} \cdot DO) \\
 PU &= (S \cdot D1) + (\overline{S} \cdot DO) \\
 &= (S \cdot D1) \cdot (\overline{S} \cdot DO) \\
 &= (S + D1) \cdot (\overline{S} + DO)
\end{align*}
\]
### IMPLEMENTATION 1

<table>
<thead>
<tr>
<th>$D_0$</th>
<th>$D_1$</th>
<th>$S$</th>
<th>$\overline{Y}$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**Y = \overline{D_0D_1S} + \overline{D_0D_1S} + \overline{D_0D_1S} + \overline{D_0D_1S}**

**Sum of Minterms**

**K-MAP**

**Need equation of form $F = \overline{\text{something}}$ since static CMOS is inverting Renegar's**

**Follows static CMOS duality rules**
**IMPLEMENTATION 2**

<table>
<thead>
<tr>
<th>$D_0$</th>
<th>$D_1$</th>
<th>$S$</th>
<th>$Y$</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>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$$Y = \overline{D_0 D_1 S} + D_0 \overline{D_1 S} + D_0 \overline{D_1 S} + D_0 D_1 S$$  
**SUM OF MINTERMS**

$$= D_0 \overline{S} + D_1 S + D_0 D_1$$  
**RENDUANT**  

$$= D_0 \overline{S} + D_1 S$$  
**RELUANT WANT TO IMPLEMENT $Y$ NOT $\overline{Y}$**

$$Y = \frac{D_0 \overline{S} + D_1 S}{2}$$

---

**K-Map**

**Follows static CMOS**

**Duality rules**
Implementation 2

\[ \text{mix pull-up front \& pull-down from back} \]

\[ \text{not drawn to each other} \]

Implementation 4

\[ \text{mix pull-up front \& pull-down from back} \]

\[ \text{not drawn to each other} \]
Implementation y is essentially a tri-state mux impl.

Tri-State Buffer

\[ A \rightarrow B^{\text{en}} \]

Tri-State Mux Impl

\[ D_0, D_1 \rightarrow Y \]

This is IMP y!
ACTIVITY

Implement \( F = ABC \) (3 input AND gate) using pass transistor logic.

Pass transistors can only "pass" 0 if \( A, B, C \) or their complement or a constant \( 0 \). Don't use pass transistors to "pass" a constant \( VDD \).

Ensure that output is always driven to \( 0 \) or \( VDD \) and is never floating.

Start with 2 input AND gate

\[
\begin{align*}
A & \quad F = AB \\
B & \quad \text{But if } B = 0, \text{ output is floating} \\
\phi
\end{align*}
\]

\[
\begin{align*}
A & \quad F = ABC \\
B & \quad \text{Now output is always driven high or low} \\
\phi
\end{align*}
\]

3 input AND gate

\[
\begin{align*}
A & \quad F = ABC \\
B & \quad \text{But if } B = 0, \text{ output is floating} \\
\phi
\end{align*}
\]