Programmable Logic Array (PLA)

Implementing combinational circuits
  Boolean functions ---> gates

Programmable logic array (PLA)
  More general implementation

Start with all inputs and their negations:

X_2

X_1

X_0
Programmable Logic Array (PLA)

Add a set of AND gates with perpendicular connections (3 in this case):

X₂
X₁
X₀

\[ \begin{array}{c}
X₂ \\
X₁ \\
X₀ \\
\end{array} \]
Programmable Logic Array (PLA)

Implement a truth table:

<table>
<thead>
<tr>
<th>$x_2$</th>
<th>$x_1$</th>
<th>$x_0$</th>
<th>$z_1$</th>
<th>$z_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<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>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</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>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Connect the necessary inputs to each AND gate in order to construct the minterms:

Dots indicate connections

Minterms:
- $\neg x_2 \neg x_1 x_0$
- $\neg x_2 x_1 x_0$
- $x_2 \neg x_1 x_0$
- $x_2 x_1 x_0$
This can be simplified by combining the inputs for each AND gate:

Now add OR gates to combine the minterms:
The vertical lines are called the AND plane
The horizontal lines are called the OR plane
We can simplify the picture even further by eliminating the gates themselves:

Programming the PLA:
Make the connections represented by the black dots