3. PAL Implementation

The following four Boolean functions are to be implemented using a PAL that has 4 inputs, four outputs, and a three-wide AND-OR structure (each output OR gate has three inputs). Complete the chart and PAL connection map below. For the chart, use a “1” to indicate an uncomplemented variable and a “0” to indicate a complemented variable, and a “-” to indicate no connection. For the connection map, label each output of the OR gates, and mark each required connection with “X”.

\[
X = \overline{A}C + BC + BC + BD \\
Y = AC + AD + \overline{A}D + \overline{B}C \\
Z = AB + \overline{A}B
\]

These AND gates are unused. Hook up the complemented and uncomplemented input to feed a logic 0 to the OR gate. \((A \cdot \overline{A} = 0, A + 0 = A)\)

<table>
<thead>
<tr>
<th>AND Inputs</th>
<th>Output Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
</tr>
</tbody>
</table>

\[
X = \overline{A}C + BC + BC(C+D) \\
Y = A(C+D) + \overline{A}D + \overline{B}C \\
Z = AB + \overline{A}B \\
S = C + D
\]
X indicates all inputs are connected. This produces a zero from the AND gate (A \cdot \bar{A} = 0)
4. PLA Implementation

\[ F(A, B, C) \]
\[
\begin{array}{ccc}
A & B & C \\
0 & 1 & 1 \\
1 & 0 & 0 \\
\end{array}
\]

\[ G(A, B, C) \]
\[
\begin{array}{ccc}
A & B & C \\
1 & 0 & 0 \\
0 & 1 & 0 \\
\end{array}
\]

The problem says to minimize the number of AND terms needed. This means we need to look for shared implicants among both K-maps.

Remember, a PLA can implement complements of functions too. This means we need to compare shared implicants among four combinations: \( F + G \), \( \overline{F} + \overline{G} \), \( \overline{F} + G \), \( F + \overline{G} \).

For this problem, notice that \( \overline{G} \) contains all of the minterms in \( F \) and \( F \) contains all of the minterms in \( G \). This means we should implement either \( F + \overline{G} \) or \( \overline{F} + G \) for a minimum solution. In fact, both give minimum solutions.
Solution 1: \( F + \overline{G} \)

\[
F(A, B, C) = \overline{A} \overline{B} C + AC
\]

\[
G(A, B, C) = \overline{A} \overline{B} C + AC + \overline{A} B
\]

Solution 2: \( \overline{F} + G \)

\[
\overline{F}(A, B, C) = A \overline{B} C + \overline{A} C + AB
\]

\[
G(A, B, C) = \overline{A} \overline{B} C + AC
\]

Compare K-maps and try to find shared implicants (not necessarily prime implicants.)
4. PLA Implementation
Implement the following two Boolean functions using a PLA. The objective is to minimize the number of product terms needed. Give your answer by filling in the PLA programming table and connection map below. For the chart, use a "1" to indicate an uncomplemented variable a "0" to indicate a complemented variable, and a "--" to indicate no connection. You should also specify whether the outputs need to be Complemented (C) or True (T) in the table. For the connection map, label each output of the AND gates, and mark each required connection with "x". Indicate connections with an 'x'. Also, label the outputs of the XOR gates. Note, that you should not need more than six product terms!

\[
F = \overline{A}BC + AC \\
\overline{G} = \overline{A}BC + AC + \overline{AB}
\]

\[F(A,B,C) = \sum m(0,5,7)\]
\[G(A,B,C) = \sum m(1,4,6)\]

<table>
<thead>
<tr>
<th>Product Term</th>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>A B C</td>
<td>F G</td>
<td></td>
</tr>
<tr>
<td>1 [\overline{A}BC]</td>
<td>0 0 0</td>
<td>1 1</td>
</tr>
<tr>
<td>2 AC</td>
<td>1 - 1</td>
<td>1 1</td>
</tr>
<tr>
<td>3 [\overline{A}B]</td>
<td>0 1 -</td>
<td>- -</td>
</tr>
<tr>
<td>4 -</td>
<td>- -</td>
<td>- -</td>
</tr>
</tbody>
</table>

Enter T or C:
4. PLA Implementation

Implement the following two Boolean functions using a PLA. The objective is to minimize the number of product terms needed. Give your answer by filling in the PLA programming table and connection map below. For the chart, use a "1" to indicate an uncomplemented variable a "0" to indicate a complemented variable, and a "-" to indicate no connection. You should also specify whether the outputs need to be Complemented (C) or True (T) in the table. For the connection map, label each output of the AND gates, and mark each required connection with "x". Indicate connections with an 'x'. Also, label the outputs of the XOR gates. Note, that you should not need more than six product terms!

\[
F(A,B,C) = \sum m(0,5,7) \\
G(A,B,C) = \sum m(1,4,6)
\]

\[F = \overline{A}BC + \overline{A}C + \overline{A}B\]

\[G = \overline{A}BC + \overline{A}C\]

<table>
<thead>
<tr>
<th>Product Term</th>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>1</td>
<td>\overline{A}BC</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>\overline{A}C</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>\overline{A}B</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Enter T or C: C T