\[ f = x_1x_2 + x_2\overline{x}_3 + \overline{x}_1\overline{x}_2 \]

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</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>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>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ f = x_1x_2 + x_2\overline{x}_3 + \overline{x}_1\overline{x}_2 = (x_1x_2 + \overline{x}_1\overline{x}_2) + (x_2\overline{x}_3) \]

**Diagram:**

- **Diagram 1:**
  - \( f_1 = x_1x_2 + \overline{x}_1\overline{x}_2 \)
  - \( x_1 \) -> \( 1 \)
  - \( x_2 \) -> \( 0 \)
  - \( x_3 \) -> \( 0 \)

- **Diagram 2:**
  - \( f_2 = x_2\overline{x}_3 \)
  - \( x_1 \) -> \( 0 \)
  - \( x_2 \) -> \( 1 \)
  - \( x_3 \) -> \( 0 \)

- **Diagram 3:**
  - \( f = f_1 + f_2 \)
  - \( x_1 \) -> \( 1 \)
  - \( x_2 \) -> \( 1 \)
  - \( x_3 \) -> \( 1 \)
  - \( f_1 \) -> \( 1 \)
  - \( f_2 \) -> \( 1 \)
  - \( f \) -> \( 0 \)
Mapping and Packing

Verilog and Schematics

Translation

Gates and FFs

Netlist of Blocks

LUTs and FFs

Packing
Placement and Routing

Tightly coupled: routing needs locations, but placement needs to know about routing congestion. Traditionally they are separated, with limited feedback, to make the task tractable.

Placement Problem

Location of I/O blocks may or may not be fixed

An ideal placement
1. is easily routable
2. has minimum delay
3. has low resource utilization
4. etc.

Accurate measurement of these parameters requires routing the placement
Routing Problem

Given locations of the blocks, realize **ALL** of their interconnections