Programmable Logic Array

Liangzhen Lai
Outline

• Introduction

• Anti-fuse

• 2-level logic decomposition

• Multi-valued Applications

• PLA Complexity
Introduction

- Programmable Logic Arrays (PLAs)
- A set of AND gates linking to OR gates
- Implement Boolean expression using sum-of-product (SOP)
- The figure shows how to implement an adder using PLAs

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

\[ C+ = \overline{ABC} + A\overline{BC} + AB\overline{C} + ABC \]
Anti-fuse

- Anti-fuse employs a thin barrier of non-conducting amorphous silicon between two metal conductors.
- Usually in mesh structure
- When a sufficiently high voltage is applied across the amorphous silicon it is turned into a polycrystalline silicon-metal alloy with a low resistance, which is conductive.
2-level Logic Decomposition

• Every Boolean logic can be decomposed into product-of-sum (POS) or sum-of-product by Karnaugh map (k-map)

\[ S = A \oplus B \oplus C = \overline{A}BC + \overline{A}B\overline{C} + A\overline{B}C + ABC \]

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

<table>
<thead>
<tr>
<th></th>
<th>A'B'</th>
<th>A'B</th>
<th>AB'</th>
<th>AB</th>
</tr>
</thead>
<tbody>
<tr>
<td>C'</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Multiple-valued Application

• The operation of PLAs can be extended to multiple-valued functions

• PLA can perform the following three functions:

1) MIN: \( f(x_1, x_2) = x_1 x_2 \) (= MIN \((x_1, x_2)\)),
2) MAX: \( f(x_1, x_2) = x_1 + x_2 \) (= MAX \((x_1, x_2)\)), and
3) literal: \( f(x_1) = a x_1^b \) (= \( r - 1 \) if \( a \leq x_1 \leq b \) and = 0, otherwise).

\[
f(x_1, x_2) = (1 \ x_1^2 \ x_2^3) + (1 \ x_1^3 \ x_2^3) + (2 \ x_1^4 \ x_2) + (3 \ x_1^2 \ x_2^3).
\]
Multiple-valued Application

- The input and output signals has 4 values
- Internal signals have only 2 values
- Special generator/encoder is required
PLA Complexity

- Total number of product terms: $2^n$
- The required product terms can be much less
- The required product terms depends on how many 1's in realized function

![Graph showing upper and lower bounds on the average number of product terms required in the minimal realization of 8-input binary functions versus the number of 1's in the function.](image)
Reference

[1] E.A. Bender and J. T. Bulter, "On the Size of PLA's Required to Realize Binary and Multiple-valued Functions"