Programmable Logic Array

Liangzhen Lai

Outline

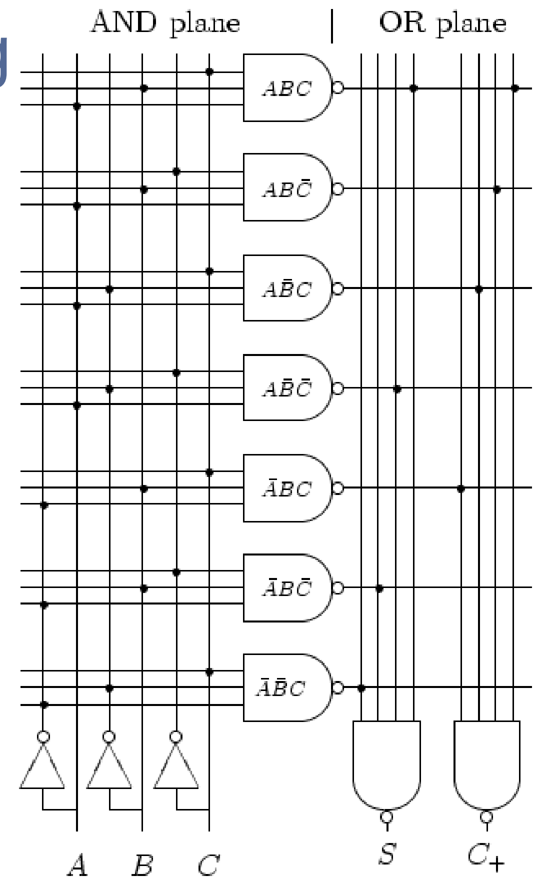
- Introduction
- Anti-fuse
- 2-level logic decomposition
- Multi-valued Applications
- PLA Complexity

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

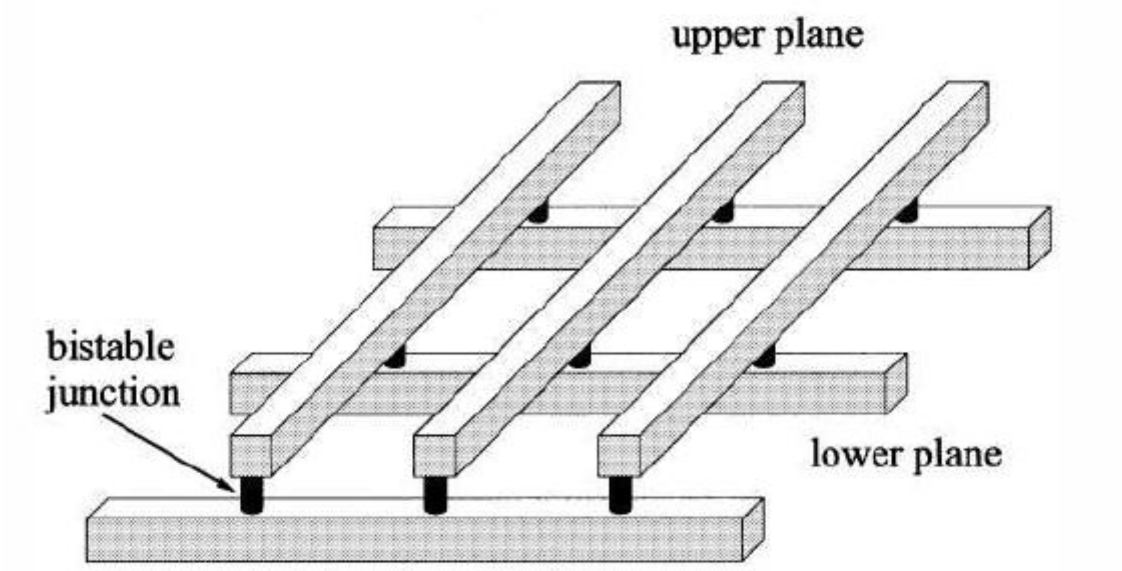
$$S = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

$$C+ = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$



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)

$$S = A \oplus B \oplus C = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C} + ABC$$

$$= (A + B + \overline{C})(A + \overline{B} + C)(\overline{A} + B + C)(\overline{A} + \overline{B} + \overline{C})$$

	A'B'	A'B	AB'	AB
C'	0	1	1	0
C	1	0	0	1

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 $a \leqslant x_1 \leqslant b$ and $= 0$, otherwise).

x_2 \ x_1	0	1	2	3
0	0	0	0	0
1	0	1	3	3
2	0	2	1	0
3	0	2	1	1

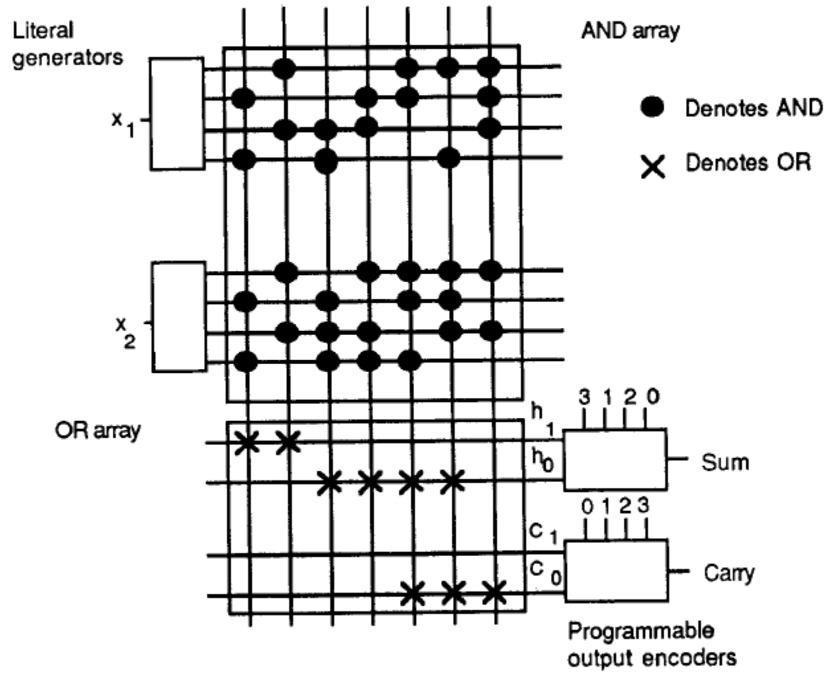
$$f(x_1, x_2) = (1\ {}^1x_1^2\ {}^1x_2^3) + (1\ {}^1x_1^3\ {}^3x_2^3) + (2\ {}^1x_1^1\ {}^2x_2^3) + (3\ {}^2x_1^3\ {}^1x_2^1).$$

Multiple-valued Application

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

Literal generator.

Four-Valued Signal	Two-Valued Signals			
0	0	3	3	3
1	3	0	3	3
2	3	3	0	3
3	3	3	3	0



Four-valued PLA adder.

Output encoding for Sum

Four-Valued Signal	Two-Valued Signals	
0	3	3
1	0	3
2	3	0
3	0	0

Output encoding for Carry.

Four-Valued Signal	Two-Valued Signals	
0	0	0
1	0	3
2	3	0
3	3	3

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

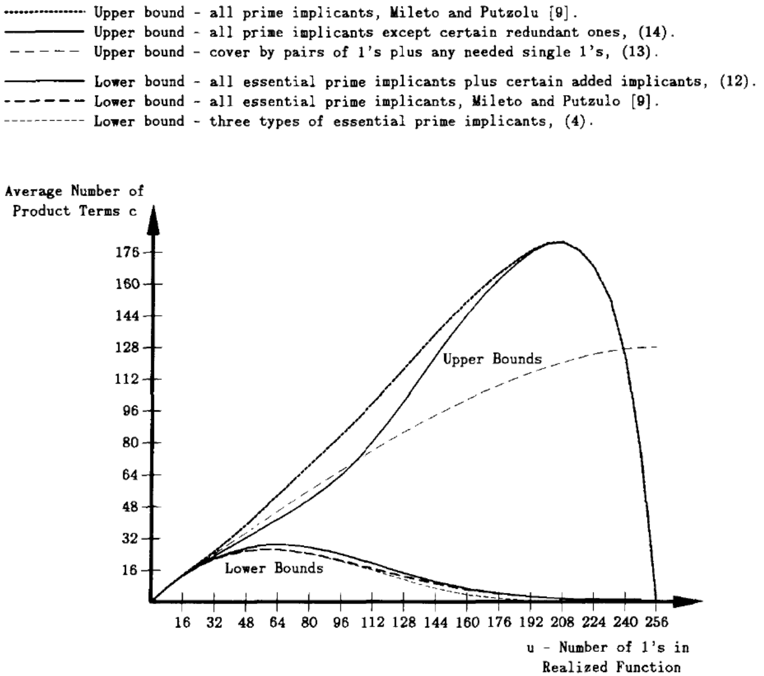


Fig. 2. Upper and lower bounds on the average number of product terms required in the minimal realization of 8-input binary functions versus the number of 1's in the function.

Reference

[1] E.A. Bender and J. T. Bulter, ”On the Size of PLA’s Required to Realize Binary and Multiple-valued Functions”
[2] T. Sasao, “Multiple-Valued Logic and Optimization of Programmable Logic Array”