1989-01

On the size of PLA's required to realize binary and multiple-valued functions

Bender, Edward A.

On the size of PLA's required to realize binary and multiple-valued functions," IEEE Transactions on Computers, C-38, Jan. 1989, pp. 82-98 1988

On the Size of PLA's Required to Realize Binary and Multiple-Valued Functions

EDWARD A. BENDER AND JON T. BUTLER, FELLOW, IEEE

Abstract—**While the use of programmable logic arrays in modern logic design is common, little is known about what PLA size provides reasonable coverage in typical applications. We address this question by showing upper and lower bounds on the average number of product terms required in the minimal realization of binary and multiple-valued functions as a function of the number of nonzero output values. When the number of such values is small, the bounds are nearly the same, and accurate values for the average are obtained.**

In addition, an upper bound is derived for the variance of the distribution of the number of product terms required in minimal realizations of binary functions. When the number of nonzero values is small, we find that the variance is small, and it follows that most functions require nearly the average number of product terms.

The variance, in addition to the upper and lower bounds, allow conclusions to be made about how PLA size determines the set of realizable functions. Although the bounds are most accurate when there are few nonzero values, they are adequate for analyzing commercially available PLA's, which we do in this paper. Most such PLA's are small enough that our results can be applied. For example, when the number of nonzero values exceeds some threshold σ_T, determined by the PLA size, only a small fraction of the functions can be realized. Our analysis shows that for all but one commercially available PLA, the number of nonzero values is a statistically meaningful criteria for determining whether or not a given function is likely to be realized.

Index Terms—**Complexity of logic circuits, enumerative analysis, logic design, multiple-valued logic, PLA, programmable logic arrays.**

I. Introduction

A PROBLEM which has remained unsolved for many years is how the number of functions realized by programmable logic arrays depends on PLA size. In the 1950's and 1960's, this problem was couched as the number of functions requiring c or fewer product terms in its minimal sum-of-products expression. Mileto and Putzolu [9], in 1964, derived expressions for the average number of prime implicants and essential prime implicants for n-variable binary functions with a fixed number of minterms. These quantities represent upper and lower bounds on the average number of product terms in a minimal sum-of-products expression. The same bounds were derived in Mileto and Putzolu [10] for binary functions with multiple outputs. Glagolev [6] obtained results similar to those in [9] over the set of all n-variable functions.

The problem is especially important now that PLA's are commercially available and are commonly used as part of VLSI circuits. For example, consider a commercially available PLA having 16 inputs, 48 product terms, and 8 outputs [7]. While the range on the number of product terms required for one output 16-input functions extends from 1 to 2^{15}, there is no analytic method for determining what fraction of such functions are realized with 48 or fewer product terms. Recently, the problem has become important for higher radices, as well, for example in the multiple-valued CCD PLA implementation of Kerkhoff and Butler [3], [8] and in the PLA proposed by Sasao [16].

Sasao and Terada [18] have shown that the analysis and design of binary PLA's with p-bit decoders at the input can be performed using functions with 2^p-valued inputs and a binary output. A calculation is shown for the number of prime implicants in functions with n r-valued inputs and a binary output. This is extended in Sasao and Terada [19], where approximations to upper and lower bounds on the number of product terms in minimal realizations of functions with r-valued inputs and a binary output, for $r \geqslant 2$, are used to approximate the average number of product terms in such functions. Other issues in the analysis and design of PLA's are considered in Sasao [13]–[17] and Chan [5].

In this paper, we derive upper and lower bounds on the average number of product terms required in PLA's where *both* the inputs and output are r-valued for $r \geqslant 2$. The bounds are dependent on the number of nonzero output values and on the distribution of those values. For the special case of $r = 2$, upper and lower bounds are derived which are tighter than any previous bounds. In addition, we show an upper bound on the variance of the distribution of the number of product terms required in the minimal realization of binary functions.

For specific cases, we show the derived results by graphs. For example, the improved bounds for binary functions on 8 inputs are compared to previously calculated bounds in a plot of the number of PLA product terms versus the number of minterms. For 8- and 12-input functions, similar plots are used to compare the derived bounds to statistically generated values of the average number of product terms needed in minimal realizations, as well as the standard deviation.

Although the upper and lower bounds are close only when

Manuscript received August 8, 1986; revised February 25, 1987 and May 18, 1987. E. A. Bender was supported by NSF Grant MCS83-00414 and ONR Contract N00014-85-K-0495. J. T. Butler was supported by NATO Grant 423/84 and by a NAVELEX Chair Professorship tenured at the Naval Postgraduate School.

E. A. Bender is with the Department of Mathematics, University of California-San Diego, La Jolla, CA 92093.

J. T. Butler is with the Department of Electrical and Computer Engineering, Naval Postgraduate School, Monterey, CA 93943.

IEEE Log Number 8823533.

0018-9340/89/0100-0082$01.00

$x_2 \backslash x_1$	0	1	2	3
0	0	0	0	0
1	0	1	3	3
2	0	2	1	0
3	0	2	1	1

Fig. 1. Example of a 4-valued 2-variable function.

the number of nonzero values is small, we find that almost all commercially available PLA's can be analyzed using the results of this paper. We consider five such PLA's. For all but one, the number σ of minterms in a given function f with few minterms is a statistically meaningful parameter in the determination of whether f is likely to be realized. That is, if all functions are equally probable, then a given function is likely to be realized if σ is less than some threshold σ_T and is unlikely to be realized if it is more. The one exception is an 8-input 32 product term PLA, where there is a wide range of σ for which such a statement cannot be made. For some PLA's, σ_T is close to the number of product terms of the PLA. In this case, very few minterms combine into larger product terms, and such PLA's resemble content-addressable memories, where σ_T is the address space size.

For 4- and 8-input 4-valued functions, upper and lower bounds on the average number of product terms needed in a minimal realization are plotted versus the number of nonzero values for various distributions of the nonzero values. For 4-input functions, the bounds are sensitive to the distribution, while for 8-input functions they are not.

The paper is organized as follows. Section II presents background information. Sections III and IV show the derivations for the upper and lower bounds, respectively, on the average number of product terms required in minimal realizations. The derivation of an upper bound on the variance is demonstrated in Section V. Our results are plotted in Section VI. The casual reader may want to consider only Sections II and VI and the concluding remarks, Section VII.

II. Background

Let $R = \{0, 1, \cdots, r-1\}$ be a set of r logic values, where $r \geqslant 2$, and let $X = \{x_1, x_2, \cdots, x_n\}$ be a set of n variables, where x_i takes on values from R. A function $f(X)$ is a mapping $f: R^n \to R$. It is convenient to visualize $f(X)$ as that shown in Fig. 1. An assignment of values to variables in X is represented by a vector $\boldsymbol{v}$. The value of $f(X)$ for that assignment is $f(\boldsymbol{v})$. If $f(\boldsymbol{v}) = k$, $\boldsymbol{v}$ is called a *k-cell* of $f(X)$. In Fig. 1, there are eight 0-cells, four 1-cells, two 2-cells, and two 3-cells.

Functions realized by PLA's considered in this paper are composed by 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).

In binary, the MIN, MAX, and literal functions correspond to AND, OR, and x^*, where $x^* \in \{x, \bar{x}\}$. Both the MAX and MIN functions can be extended to three or more variables. Furthermore, constants and literals can occur as operands. For example, for $r = 4$, $f(x_1, x_2) = 2\ {}^1x_1^1\ {}^2x_2^3$ is a function which is 2 when x_1 is 1 and x_2 is 2 or 3 and is 0 otherwise. Functions of this type are called *product terms*. Any function $f(X)$ can be expressed as the MAX of product terms. For example, the function in Fig. 1 can be expressed as

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

It is convenient to use the term *sum-of-products* to describe such an expression, with the recognition that sum refers to MAX and product to MIN. The PLA's considered in this paper realize such expressions, and the design is one of finding an expression for a given function which has no more than the number of product terms allowed by the PLA. A sum-of-products expression is *minimal* if there is no other expression for $f(X)$ with fewer product terms. The expression in (1) is minimal, since three or fewer terms are impossible due to the necessity of realizing the two nonzero logic values 2 and 3, with at least one term each and the necessity of realizing 1's with at least two product terms.

An *implicant for k* of a given function $f(X)$ is a product term $I(X)$ such that $f(X) \geqslant I(X)$ and there is at least one k-cell of $f(X)$ which is a k-cell of $I(X)$. A *prime implicant for k* of $f(X)$ is an implicant $I(X)$ of $f(x)$ such that there is no other implicant $I'(X)$ of $f(X)$, where $I(X)' \geqslant I(X)$. For example, $1\ {}^2x_1^2\ {}^2x_2^3$ is an implicant of the function in Fig. 1. However, it is not a prime implicant, that status being held by $1\ {}^1x_1^2\ {}^1x_2^3$. An *essential prime implicant for k*, is a prime implicant $I(X)$ for k such that there is a k-cell of $I(X)$ which is not a k-cell for any other prime implicant. For example, $1\ {}^1x_1^2\ {}^1x_2^3$ is an essential prime implicant for the function in Fig. 1, by virtue of the 1-cell at (2, 2), which is not a 1-cell in any other prime implicant. A k-cell in a prime implicant is *(k − 1)-bounded* if all cells adjacent to it but not in the prime implicant contain values at most equal to $k - 1$ (two cells are *adjacent* if they differ by a unit vector). For example, the 1-cell at (2, 2) is 0-bounded, while the 1-cell at (2, 3) is not.

III. Lower Bounds on the Average Number of Product Terms in Binary and Multiple-Valued PLA's

Mileto and Putzolu [9] derive expressions for the average number of essential prime implicants in n-variable 2-valued functions with u 1's. This is a lower bound on the average number of AND terms in a minimal sum-of-products expression.

In Sasao and Terada [19], the excessive computer time required to evaluate the expressions derived in [9] is avoided by enumerating only a subclass of essential prime implicants. However, this class is large enough to include most essential prime implicants. An inclusion/exclusion sum is generated and all terms are approximated. The result is an *approximation* to the lower bound.

Our approach to *r-valued* functions is similar. Instead of enumerating all essential prime implicants, only a subclass is enumerated. Again, this class is large enough to include most essential prime implicants. However, in order to compute a *provable* lower bound, no approximations are made. In spite of this, reasonable computation times are achieved.

A. Lower Bounds for r-Valued Functions Derived from Three Types of Essential Prime Implicants

1) Method of Approach: Given an r-valued function, a 2-valued function can be obtained by converting all values less than k to 0 and the rest to 1. Any 2-valued function so derived corresponds to many r-valued functions. We approach the problem of computing lower bounds on PLA size by enumerating a binary form of the function and then converting to the r-valued form.

Let $f(x_1, x_2, \cdots, x_n)$ be an r-valued function, and let m_k be the number of k-cells for $0 \leqslant k \leqslant r-1$. For the example function in Fig. 1, $m_0 = 8$, $m_1 = 4$, $m_2 = 2$, and $m_3 = 2$. Let $M = m_0 + m_1 + \cdots + m_{r-1}$. It follows that $M = r^n$. For a given distribution of values m_i, the number of functions with that distribution is

$$\binom{M}{m_0, m_1, \cdots, m_{r-1}} = \frac{M!}{m_0! m_0! \cdots m_{r-1}!}.$$

The form of the expressions for the lower bounds on the average number of product terms is

$$\text{LB} = \frac{1}{\binom{M}{m_0, m_1, \cdots, m_{r-1}}} \sum_f c(f) \tag{2}$$

where $c(f)$ is a lower bound on the number of product terms needed in a minimal realization of f and where the sum is over all functions with m_0 0's, m_1 1's, $\cdots$, and m_{r-1} $r-1$'s. $c(f)$ is derived by counting three categories of essential prime implicants used in any minimal realization of f, 1) single cells, 2) single lines, and 3) planar 2×2 squares. Our approach to evaluating the sum of (2) is to enumerate these essential prime implicants and to sum over the functions containing them. Let N_r be the total contribution to this sum by a specified r-valued essential prime implicant.

In converting from a binary to an r-valued function, we recognize four types of cells of the r-valued function, according to the logic value in the cell,

1) less than k,
2) equal to k,
3) greater than or equal to k, and
4) DON'T CARE.

For example, an *isolated* k-cell is a single cell v that has all neighbors in category 1, while all nonneighbors are in category 4. v is represented in a minimal sum-of-products expression by an essential prime implicant covering just that cell, larger implicants being precluded by neighbor cells with values less than k. Nonneighbors are DON'T CARE, since their value has no effect on the implicant covering v. We count the corresponding 2-valued functions according to $0 \leftrightarrow$ less than k, $1 \leftrightarrow$ equal to k, and $1 \leftrightarrow$ greater than or equal to k, and note the number E of cells in the *equal to k* category. For a given category of essential prime implicants, if N_2 is the number of binary functions containing such implicants, and N_r is the corresponding number of r-valued functions, then

$$N_r = \binom{m_0 + m_1 + \cdots + m_{k-1}}{m_0, m_1, \cdots, m_{k-1}} \cdot \binom{(m_k - E) + m_{k+1} + \cdots + m_{r-1}}{m_k - E, m_{k+1}, \cdots, m_{r-1}} N_2.$$

That is, N_r is the product of,

1) the number of ways to associate $0, 1, \cdots,$ and $k-1$ with a binary 0,

2) the number of ways to associate $k, k+1, \cdots,$ and $r-1$ with a binary 1, and

3) the number of 2-valued functions.

It follows that the contribution R_r to the expression for LB from each category of implicant is

$$R_r = \frac{N_r}{\binom{M}{m_0, m_1, \cdots, m_{r-1}}} = \frac{m_k!}{(m_k - E)!} \frac{(u-E)!}{u!} \frac{N_2}{\binom{M}{u}} = \left(\prod_{i=0}^{E-1} \frac{m_k - i}{u - i} \right) R_2 \tag{3}$$

where $u = m_k + m_{k+1} + \cdots + m_{r-1}$. For functions with few nonzero values, E is small, and so this expression is simple.

Since we consider three categories of essential prime implicants, our lower bound is a sum over three terms,

$$\text{LB} = \sum_{A \in \{I, L, S\}} R_r(A), \tag{4}$$

where I, L, and S represent isolated single cell implicants, single line implicants, and planar 2×2 implicants, respectively.

2) Category I—Isolated Single Cell Essential Prime Implicants: In translating this to the binary problem, we observe that the relevant cell v must have value k, while all adjacent cells have $k-1$ or less. Thus, $E = 1$ in (3). Suppose i of the coordinates in v are interior, i.e., have values strictly between 0 and $r-1$. Then $2i + (n-i)$ adjacent cells must be 0 with the rest DON'T CARES. For the binary case,

$$R_2(I) = \frac{1}{\binom{M}{u}} \sum_{i=0}^{n} \eta(i) \beta(i) \tag{5}$$

where η is the number of ways to choose v and β is the number of ways to fill in other logic values. We have

$$\eta(i) = \binom{n}{i} 2^{n-i} (r-2)^i,$$

since there are $\binom{n}{i}$ ways for i of the n coordinates to be interior, 2^{n-i} ways for each of the two boundary values, 0 and $r-1$, to occur, and $(r-2)^i$ ways for the interior logic values to occur. Furthermore,

$$\beta(i)=\binom{M-(1+2i+(n-i))}{u-1},$$

since from among the $M - (1 + 2i + (n - i))$ cells other than v and its neighbors, $u - 1$ of these must be chosen to be 1. Using (3), we have, for the count of essential prime implicants consisting of a single isolated k-cell,

$$R_r(I)=\frac{m_k}{u}R_2(I). \tag{6}$$

3) Category L—Single Line Essential Prime Implicants: Essential prime implicants in this category consist of two or more cells aligned along one of the n variables x_i. Let the cells in this implicant be indexed starting with 1 for the cell with the smallest value of x_i, and let j be the smallest index corresponding to a k-cell which is $k-1$-bounded. Since this is the only cell which must contain k, $E = 1$. There are two cases of single line implicants, $L_j(\text{bdy})$ where the first cell of the implicant has $x_i = 0$ and $L_j(\text{int})$ where the first cell has $x_i \neq 0$. The position of the implicant specifies all remaining coordinates. Let t be the number of coordinates not on a boundary. We have

$$R_r(L_j(*))=\frac{m_k}{u}R_2(L_j(*)) \tag{7}$$

and

$$R_2(L_j(*))=n\sum_{t=0}^{n-1}\binom{n-1}{t}2^{n-1-t}(p-2)^tB_j(*)\frac{C_j(t,*)}{\binom{M}{u}} \tag{8}$$

where the number of ways to choose x_i for the first cell is

$$B_j(*)=\begin{cases}1 & \text{if } *=\text{bdy}\\ p-2 & \text{if } *=\text{int and } j=1,\\ p-j & \text{if } *=\text{int and } j>1\end{cases}$$

and $C_j(t, *)$ is the number of ways to assign function values given the first location and alignment of the implicant. The factor n in (8) counts the number of alignments. $\binom{n-1}{t}$ is the number of ways t coordinates of an implicant can be internal, while 2^{n-1-t} and $(p - 2)^t$ count the ways border and nonborder coordinates can occur.

Consider $C_j(t, *)$. If $j = 1$, then the first cell is 0-bounded. Since this is a line implicant, two 1's are forced. Also,

$$2t+(n-1-t)+\begin{cases}0 & \text{if } *=\text{bdy}\\ 1 & \text{if } *=\text{int}\end{cases}$$

0's are forced. Thus,

$$C_1(t,\text{bdy})=\binom{M-(n+t+1)}{u-2} \text{ and }$$

$$C_1(t,\text{int})=\binom{M-(n+t+2)}{u-2}.$$

The case for $j > 1$ and $* = \text{bdy}$ is more complex. Let $X \subseteq \{1, 2, \cdots, j\} = J$, $N_{\geqslant}(X)$ = the number of ways function values can be assigned given the position of the first cell and implicant alignments so that positions in X are 0-bounded. Define $N_{=}(X)$ in the same way except that *none* of the positions in $J - X$ are 0-bounded. Then, $C_j(t, \text{bdy}) = N_{=}(\{j\})$. By the principle of inclusion/exclusion (Bender and Goldman [2]),

$$N_{=}(\{j\})=\sum_{j\in X\subseteq J}(-1)^{|X|-1}N_{\geqslant}(X). \tag{9}$$

In deriving $N_{\geqslant}(X)$, we note there are j forced 1's and $|X|(2t + (n - 1 - t))$ forced 0's. Letting $x = |X|$, we have from (9)

$$C_j(t,\text{bdy})=\sum_{x=1}^{j}\binom{j-1}{x-1}(-1)^{x-1}\binom{M-x(n+t-1)-j}{u-j}.$$

The derivation for $C_j(t, \text{int})$ is the same except that an additional 0 is forced. Thus,

$$C_j(t,\text{int})=\sum_{x=1}^{j}\binom{j-1}{x-1}(-1)^{x-1} \cdot\binom{M-x(n+t-1)-j-1}{u-j}.$$

4) Category S—Planar Prime Implicants: An essential prime implicant in this class is 2×2 and has the property that in the two coordinates where the implicant values vary, it is bounded by cells at most $k - 1$ or by the boundary. This involves no cells where $r = 2$, 4 cells when $r = 3$, and 4, 6, or 8 cells when $r > 3$ depending on whether $b = 2$, 1, or 0 sides of the implicant are on the boundary. Suppose exactly j cells of the implicant contain k and are $k - 1$-bounded.

From (3), we have

$$R_r(S_j(b))=\prod_{i=0}^{j-1}\frac{m_k-i}{u-i}R_2(S_j(b)). \tag{10}$$

Similar to (8) for the single line prime implicant, we have

$$R_2(S_j(b))=\binom{n}{2}\sum_{t=0}^{j-1}\binom{n-2}{t}2^{n-2-t}(r-2)^tB\frac{D_j(t,b)}{\binom{M}{u}} \tag{11}$$

where

$$B=\begin{cases}1 & \text{if } r=2\\ \binom{2}{b}2^b(r-3)^{2-b} & \text{if } r\geqslant 3.\end{cases}$$

Here b is the number of the two varying coordinates that are adjacent to a border. As in (8), B represents the number of ways of positioning the implicant. When $r = 2$, the position is forced. When $r \geqslant 3$, $\binom{2}{b}$ is the number of ways b of the two variables involve a border, 2^b is the number of ways the border can be chosen (0 or $r - 1$), and $(r - 3)^{2-b}$ is the number of ways nonborder values can be chosen.

Let F denote the set of four positions in the 2×2 implicant. Using inclusion/exclusion to solve for $D_j(t, b)$, we have

$$D_j(t, b) = \sum_{j=|Y|} N_=(Y) = \binom{4}{j} \sum_{J \subseteq X \subseteq F} (-1)^{|X|-j} N_{\geqslant}(X)$$

$$= \binom{4}{j} \sum_{x=j}^{4} (-1)^{x-j} \binom{4-j}{x-j} \binom{M-z-4}{u-4}$$

where

$$z = x(2t + (n-2-t)) + \begin{cases} 0 & \text{if } r = 2 \\ 8-2b & \text{if } r \geqslant 3. \end{cases}$$

Thus, the lower bound associated with three types of essential prime implicants is found by substituting (6), (7), and (10) into (4).

B. Improved Lower Bounds for Binary Functions Derived by Counting Certain Nonessential Prime Implicants

All known lower bounds on the average number of product terms in minimal sum-of-products expressions for binary functions count essential prime implicants only. The best bounds are those which count all essential prime implicants [9]. If LB is the average number of essential prime implicants in n-variable binary functions, then an *improved* bound LB$'$ is

$$\text{LB}' = \text{LB} + M(n, u) \tag{12}$$

where $M(n, u)$ is the average number of certain nonessential prime implicants. We derive $M(n, u)$ as follows.

Let ξ_i be the ith unit vector in R^n, and let $v \oplus \xi_i$ be a cell whose components are the Exclusive OR of the corresponding components of v and ξ_i. Since ξ_i is a unit vector, v and ξ_i are adjacent. Let f be a binary function, and let v be a cell with the properties

1) $f(v) = 1$,

2) $f(v \oplus \xi_i) = f(v \oplus \xi_j) = 1$, for $i, j \in \{0, 1, \cdots, n\}$,

3) $f(v \oplus \xi_k) = 0$, if $k \neq i, j$, and

4) $f(v \oplus \xi_i \oplus \xi_j) = 0$.

It follows that the two pairs of 1-cells $(v, v \oplus \xi_i)$ and $(v, v \oplus \xi_j)$ each belong to a prime implicant of f. If

5) $(v, v \oplus \xi_i)$ and $(v, v \oplus \xi_j)$ belong to *nonessential* prime implicants,

then the lower bound calculation using only essential prime implicants does not count any implicant which covers v. Therefore, we can derive an improved lower bound by counting the number of v which satisfies the five properties above.

We can guarantee that both pairs of 1-cells are nonessential with the proviso that both $v \oplus \xi_i$ and $v \oplus \xi_j$ are adjacent to at least one 1-cell other than v. We proceed in two steps.

First, we count the ways to choose ξ_i, ξ_j, and v. In order to avoid overcounting which can occur, for example, when v, $v' = v \oplus \xi_i$, and $v' \oplus \xi_j$ form a triple of 1-cells satisfying the proviso, we restrict the ith and jth component of v to be 0. Thus, the number of choices is $\binom{n}{2} 2^{n-2}$, since there are $\binom{n}{2}$ ways to choose i and j and 2^{n-2} ways to choose all other components of v.

Second, we count the ways to complete the pattern. There are three forced 1's and $n - 1$ forced 0's. Of the

$$\binom{2^n - (n+2)}{u-3}$$

ways to complete the functions, some correspond to essential prime implicants at either $v \oplus \xi_i$ or $v \oplus \xi_j$ or both. These correspond to the case where one or both $v \oplus \xi_i$ and $v \oplus \xi_j$ have all other neighboring cells as 0 cells. Using inclusion/exclusion, we have for the number of completions

$$C(n, u) = \binom{2^n - (n+2)}{u-3} - 2\binom{2^n - (n+2) - (n-2)}{u-3} + \binom{2^n - (n+2) - 2(n-2)}{u-3}.$$

$M(n, u)$, the average number of implicants, is then $C(n, u)/\binom{2^n}{u}$.

IV. Upper Bounds on the Average Number of Product Terms in Binary and Multiple-Valued PLA's

Since a minimal sum-of-products expression can be derived by forming the MAX of an appropriate choice of prime implicants, the average number of prime implicants is an upper bound on the average number of product terms in minimal sum-of-products expressions. This is the approach chosen by Mileto and Putzolu [9] in their analysis of binary functions. Sasao and Terada [19] avoid the problem of excessive computation time needed for the calculation of the upper bounds of [9] by forming approximations to the average number of prime implicants. The result is an *approximation* to the upper bound.

Our approach to *r-valued* functions is different. Nonzero cells are covered with implicants consisting of pairs of identical adjacent cells, starting with pairs aligned in the x_1 direction, then the x_2 direction, etc. Nonzero cells which remain uncovered are covered with single cell implicants. We avoid overcounting which occurs, for example, when a square of cells is covered by four pairs when two will suffice, by the ordering of pairs according to the alignment with the axis defined by the input variables. For a given function, the number of terms used in any covering is an upper bound on the number required in a minimal covering. Thus, the average number in the covering described above is a *provable* upper bound on the average number required in a minimal covering.

A. Upper Bounds for r-Valued Functions Derived from Pair and Single Cell Implicants

Given $f: R^n \rightarrow R$ and $v, v + \xi_j \in R^n$, $(v, v + \xi_j)$ is a k *special pair* in f if

1) $f(v) = f(v + \xi_j) = k \neq 0$,
2) $f(v \pm \xi_i) \neq k$ for $1 \leqslant i < j$, if $v \pm \xi_i \in R^n$, and
3) $f(v + \xi_j \pm \xi_i) \neq k$ for $1 \leqslant i < j$ if $v + \xi_j \pm \xi_i \in R^n$,

where $\pm$ denotes that two statements are valid, one with $+$ and the other with $-$. We have the following theorem.

Theorem 1: f can be covered with $u - s$ implicants, where u is the number of nonzero cells in f and s is the number of special pairs in f.

Proof: Let S be the set of special pairs and P be the set of cells covered by S. Two special pairs which overlap, must be colinear, i.e., differ in the same input variable. Otherwise, there is a contradiction associated with the specification of the special pair aligned in the coordinate with the larger index. Thus, S can be partitioned into sets covering nonoverlapping sets of colinear points. If L is the largest set of special pairs covering a set G of colinear k-cells, L covers $|L| + 1 (\leqslant r)$ cells; however, a single line implicant covers G. Thus, the number of implicants needed to cover P is at most $|P| - |S|$. The nonzero cells not in P can be covered by $u - |P|$ single point implicants. Hence, $u - |P| + |P| - |S| = u - s$ implicants suffice. Q.E.D.

Thus, an upper bound on the average number of prime implicants is

$$\mathrm{UB} = \frac{1}{\binom{M}{m_0, m_1, \cdots, m_{r-1}}} \sum_f (u - s) \tag{13}$$

and so we need the sum of $u - s$ over all functions. The sum of u over all functions is $m_k \binom{M}{m_k}$. The sum of s over all functions is calculated as follows. The number of v with $v, v + \xi_j \in R^n$ and t coordinates $<j$ not on a boundary is

$$\left(\binom{j-1}{t}(r-2)^t 2^{j-1-t}\right)(r-1)(r^{n-j})$$

where the three factors enumerate ways to pick components with coordinates $<j$, $=j$, and $>j$.

Let $f_k(v) = 1$ if $f(v) = k$ and 0 otherwise. The number of forced 1's in f_k is 2 and the number of forced 0's is $2(2t + (j - 1 - t))$. Thus, an upper bound for the number of implicants summed over all $\binom{M}{m_k}$ different f_k is

$$Q_k = m_k \binom{M}{m_k} - \sum_{j=1}^{n} \sum_{t=0}^{j-1} \binom{j-1}{t} \cdot (r-2)^t 2^{j-1-t} (r-1) r^{n-j} \binom{M - 2(j+t)}{m_k - 2}.$$

Since each binary f_k corresponds to precisely

$$\binom{M - m_k}{m_1, \cdots, m_{k-1}, m_{k+1}, \cdots, m_k}$$

distinct r-valued f's, an upper bound for the average number of implicants is

$$\sum_{k=1}^{r-1} Q_k \binom{M - m_k}{m_1, \cdots, m_{k-1}, m_{k+1}, \cdots, m_r} \Bigg/ \binom{M}{m_0, m_1, \cdots, m_{r-1}} = \sum_{k=1}^{r-1} \frac{Q_k}{\binom{M}{m_k}}.$$

In the binary case ($r = 2$), this upper bound for the average number of prime implicants reduces to

$$u - \frac{M}{2\binom{M}{u}} \sum_{j=1}^{n} \binom{M - 2j}{u - 2}.$$

B. Improved Upper Bounds for Binary Functions Derived by Eliminating Redundant Implicants

All known upper bounds on the average number of product terms in minimal sum-of-products expressions for binary functions count prime implicants exclusively. Since not all prime implicants are used in a minimal sum-of-products expression, an improved upper bound can be obtained by eliminating certain redundant prime implicants. Specifically, consider three overlapping implicant pairs.

$$1)\ \boldsymbol{a}\ \boldsymbol{b},\quad 2)\ \boldsymbol{b}\ \boldsymbol{c},\quad \text{and}\quad 3)\ \boldsymbol{c}\ \boldsymbol{d},$$

where $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, and $\boldsymbol{d}$ are 1-cells, and 2 is a prime implicant. In a count of prime implicants, the inclusion of implicants containing 1 and 3 makes it unnecessary to include 2. The latter is a special case of *absolutely eliminable prime implicants* (Muroga [12]). Let implicants 1, 2, and 3 be aligned with variables x_{i_1}, x_{i_2}, and x_{i_3}; that is, $\boldsymbol{a}$ and $\boldsymbol{b}$ differ in coordinate x_{i_1}, $\boldsymbol{b}$ and $\boldsymbol{c}$ differ in coordinate x_{i_2}, and $\boldsymbol{c}$ and $\boldsymbol{d}$ differ in coordinate x_{i_3}.

We can assume, without restricting instances, that $i_1 < i_3$ provided we distinguish $\boldsymbol{b}\ \boldsymbol{c}$ and $\boldsymbol{c}\ \boldsymbol{b}$. Thus, we have three cases labeled by the relative value of i_2,

$$\text{LOW: } i_2 < i_1 < i_3,$$

$$\text{MID: } i_1 < i_2 < i_3, \text{ and}$$

$$\text{HIGH: } i_1 < i_3 < i_2.$$

For a given $\boldsymbol{b}\ \boldsymbol{c}$, assume that i_1 and i_3 are in the earliest possible direction; that is, there are no other implicant pairs, $\boldsymbol{a}'\ \boldsymbol{b}$ and $\boldsymbol{c}\ \boldsymbol{d}'$, such that either or both $\boldsymbol{a}'\ \boldsymbol{b}$ and $\boldsymbol{c}\ \boldsymbol{d}'$ are aligned along an axis by an input of index lower than that of $\boldsymbol{a}\ \boldsymbol{b}$ and $\boldsymbol{c}\ \boldsymbol{d}$, respectively. With i_1 in the earliest possible direction, $i_1 - 1$ 0's are forced for MID and HIGH. For LOW, the restriction $i_2 < i_1$ forces a 1 in a direction earlier than i_1, and so $i_1 - 2$ 0's are forced. Similarly, with i_3 in the earliest direction, the restriction $i_2 < i_3$ for LOW and MID forces a 1 in a direction earlier than i_3, and so $i_3 - 2$ 0's are forced. For HIGH, neither i_1 nor i_2 impose a forced 1 and so $i_3 - 1$ 0's are forced.

To arrive at an improved upper bound, we take the upper bound derived by counting prime implicants and subtract the overcounting just discussed. We may have a problem if, for example, $\boldsymbol{a\ b}$ is used to eliminate $\boldsymbol{b\ c}$, and it is also eliminated using, say $\boldsymbol{r\ a}$ and $\boldsymbol{b\ s}$. If $\boldsymbol{b\ s}$ were not eliminated, it would still be correct to eliminate $\boldsymbol{b\ c}$. We can guarantee this by insisting that any implicant in a direction later than the direction of a prime implicant it is used to eliminate be an essential prime implicant. This forces $n - 1$ 0's adjacent to $\boldsymbol{d}$ in LOW and MID and $n - 2$ 0's adjacent to $\boldsymbol{a}$ in LOW (not $n - 1$ because of double counting due to a 0 forced by $i_3 > i_2$ is as early as possible).

We count the ways $\boldsymbol{b\ c}$ is a prime implicant by using inclusion/exclusion. Suppose i is such that the cells at $\boldsymbol{b} \oplus \xi_i$ and $\boldsymbol{c} \oplus \xi_i$ are 1-cells. Since i_3 is in the earliest possible direction, we see from $\boldsymbol{c} \oplus \xi_i$ that $i \geqslant i_3$. Since $\boldsymbol{c\ d}$ is an essential prime implicant when $i_3 > i_2$, we cannot have $i = i_3$ for LOW or MID. If we force $\boldsymbol{c} \oplus \xi_{i_3}$ to be a 1-cell for HIGH, we have $i > i_3$ for all cases. Let X be a set of directions i such that $\boldsymbol{b} \oplus \xi_i$ and $\boldsymbol{c} \oplus \xi_i$ are 1-cells. By the previous discussion, $i > i_3$. Also, $i_2 \notin X$. Thus, if $x = |X|$, the number of ways to choose X is $\binom{n-i_3-1}{x}$ for HIGH and $\binom{n-i_3}{x}$ for MID and LOW. Each X forces $2x$ 1's. There are 2^n ways to choose the ordered pair $\boldsymbol{b\ c}$ given i_2; that is, there 2^{n-1} ways to position $\boldsymbol{b\ c}$, and there are two ways to pick the value of x_i for $\boldsymbol{b}$ of $\boldsymbol{b\ c}$, 0 or 1. Thus, $N_{=}(\phi)$ for the three cases is

$$\text{LOW: } \mathrm{RI}_{\mathrm{low}} = 2^n \sum_{i_2<i_1<i_3} \sum_x (-1)^x \binom{n-i_3}{x} \cdot \binom{2^n-(2x+4)-(i_1+i_3+2n-7)}{u-(2x+4)}$$

$$\text{MID: } \mathrm{RI}_{\mathrm{mid}} = 2^n \sum_{i_1<i_2<i_3} \sum_x (-1)^x \binom{n-i_3}{x} \cdot \binom{2^n-(2x+4)-(i_1+i_3+n-4)}{u-(2x+4)}$$

$$\text{HIGH: } \mathrm{RI}_{\mathrm{high}} = 2^n \sum_{i_1<i_3<i_2} \sum_x (-1)^x \binom{n-i_3-1}{x} \cdot \binom{2^n-(2x+4)-(i_1+i_3-1)}{u-(2x+4)}.$$

In all cases, i_2 occurs only in the summation. Thus, the summations on i_2 can be replaced by the following factors:

$$\text{LOW: } (i_1-1)$$

$$\text{MID: } (i_3-i_1-1).$$

$$\text{HIGH: } (n-i_3).$$

If UB is the upper bound derived by counting all prime implicants, the improved bound UB$'$ is

$$\mathrm{UB}' = \mathrm{UB} - \frac{1}{\binom{2^n}{u}} (\mathrm{RI}_{\mathrm{low}} + \mathrm{RI}_{\mathrm{mid}} + \mathrm{RI}_{\mathrm{high}}). \quad (14)$$

V. An Upper Bound on the Variance of the Number of PLA Product Terms for Binary Functions

Glagolev [6] calculated the variance of the distribution associated with the number of functions containing exactly m implicants (not necessarily prime) with 2^j 1-cells for $0 \leqslant j \leqslant n$. Mileto and Putzolu [11] performed a similar calculation with u, the number of minterms in the function, as a parameter. This is compared to statistical data on the number of prime implicants required in minimal realizations, and is shown to be comparable. The calculation of the variance on prime implicants is more difficult, and of the variance of prime implicants in minimal realizations even more so. Our approach is to calculate an upper bound for the latter.

Let X, Y, and Z be random variables bounded as follows, $0 \leqslant X \leqslant Y \leqslant Z$. The variance of Y, $\sigma^2(Y)$, is related to expected values of Y as

$$\sigma^2(Y) = E(Y^2) - E^2(Y) \leqslant E(Z^2) - E^2(X).$$

Since $\sigma^2(Z) = E(Z^2) - E^2(Z)$, we have

$$\sigma^2(Y) \leqslant \sigma^2(Z) + E^2(Z) - E^2(X)$$

$$\leqslant \sigma^2(Z) + (E(Z) - E(X))(E(Z) + E(X)). \quad (15)$$

If we identify Y with the number of product terms in a PLA and X and Z with lower and upper bounds, we have an upper bound on the variance of the number of product terms as a function of the variance of the upper bound variable and expected values of the upper and lower bound variables. However, an improved bound can be obtained using *deficiencies*.

Associate X, Y, and Z with $u - c_u$, $u - c$, and $u - c_l$, respectively, where u is the number of 1's in the function, and where c_u, c, and c_l are the number of product terms associated with the upper bound, average, and lower bound, respectively. Because we are enumerating functions with few 1's, c, c_u, and c_l are small, and $u - c_u$, $u - c$, and $u - c_l$ are close to 0. Thus, the upper bound on $\sigma(Y)$ is reasonably tight. For fixed u, $\sigma(Y) = \sigma(c)$ and $\sigma(Z) = \sigma(c_l)$. From (15), we have

$$\sigma^2(c) \leqslant \sigma^2(c_l) + (2u - E(c_l) - E(c_u))(E(c_u) - E(c_l)).$$

We use for $E(c_l)$ the lower bound obtained from all essential prime implicants as calculated in Mileto and Putzolu [9], and for $E(c_u)$, the upper bound as calculated from all prime implicants except certain redundant ones. Consider the calculation of $\sigma^2(c_l)$. Let X be a random variable whose value is the number of essential prime implicants in a function f, and let EPI (f) be the set of essential prime implicants of f. We want

$$\binom{M}{u} E(X^2) = \sum_f \left(\sum_\pi \Psi(\pi \in \mathrm{EPI}\ (f)) \right)^2$$

where we define $\Psi(\pi \in \mathrm{EPI}\ (f)) = 1$ if π is an essential prime implicant of f and $=0$ otherwise. Expanding the square and rearranging yields

$$\binom{M}{u} E(X^2) = \sum_{\pi_1,\pi_2} \sum_{f} \Psi(\pi_1,\ \pi_2 \in \mathrm{EPI}\ (f))$$

$$= \binom{M}{u} E(X) + \sum_{\pi_1 \neq \pi_2} N_{\pi_1,\ \pi_2} \tag{16}$$

where $N_{\pi_1,\ \pi_2}$ is the number of functions with both π_1 and π_2 as essential prime implicants. The last sum can be calculated by choosing two distinct implicants and by determining the number of ways to complete the function so that both are essential prime implicants.

An implicant π can be represented by an element of $\{0, 1, *\}^n$, and is a function which is 1 if and only if x_i is restricted to be 0 or 1 when the ith position is 0 or 1, respectively, and is unrestricted when the ith position is $*$. For example, 01*0*1 represents the implicant $\bar{x}_1 x_2 \bar{x}_4 x_6$. Let π_1 and π_2 be two prime implicants. Let

$n_{\substack{00\\11}}$ = number of coordinates where π_1 is 0 or 1 and π_2 is 0 or 1,

$n_{\substack{*0\\*1}}$ = number of coordinates where π_1 is $*$ and π_2 is 0 or 1,

$n_{\substack{0*\\1*}}$ = number of coordinates where π_1 is 0 or 1 and π_2 is $*$, and

n_{**} = number of coordinates where π_1 is $*$ and π_2 is $*$.

For example, for $\pi_1 = 01{*}0{*}1$ and $\pi_2 = 0{*}{*}1{*}0$, $n_{\substack{00\\11}} = 3$, $n_{\substack{*0\\*1}} = 0$, $n_{\substack{0*\\1*}} = 1$, and $n_{**} = 2$. Let d, the number of *disagreements*, be the number of components in which π_1 is 0 and π_2 is 1 or vice versa. For example, with π_1 and π_2, $d = 2$. In evaluating the last sum of (16), there are four cases to consider according to the number of disagreements. That is,

$$\binom{M}{u} E(X^2) = \binom{M}{u} E(X) + E_0 + E_1 + E_2 + G_2$$

where E_d is the sum over all pairs of essential prime implicants of the number of functions, where d is the number of disagreements, and G_d is the number with more than d disagreements. The evaluation of E_d and G_2 proceeds in two parts,

1) count the ways π_1 and π_2 can be chosen given $\boldsymbol{n} = (n_{\substack{00\\11}}, n_{\substack{0*\\1*}}, n_{\substack{*0\\*1}}, n_{**})$, and

2) count the ways f can be chosen given π_1, π_2, and $\boldsymbol{n}$.

Evaluating E_d for 1) yields

$$\binom{n}{\boldsymbol{n}} 2^{n_{\substack{00\\11}} + n_{\substack{0*\\1*}}} \binom{n_{\substack{00\\11}}}{d} 2^{n_{\substack{*0\\*1}}}$$

$$= \frac{n!}{n_{\substack{00\\11}}!\, n_{\substack{*0\\*1}}!\, n_{\substack{0*\\1*}}!\, n_{**}!} \binom{n_{\substack{00\\11}}}{d} M 2^{-n_{**}}.$$

Evaluating G_2 for 1) yields

$$\binom{n}{\boldsymbol{n}} M 2^{-n_{**}} \left(2^{n_{\substack{00\\11}}} - \binom{n_{\substack{00\\11}}}{0} - \binom{n_{\substack{00\\11}}}{1} - \binom{n_{\substack{00\\11}}}{2} \right)$$

$$= \frac{n!}{n_{\substack{00\\11}}!\, n_{\substack{*0\\*1}}!\, n_{\substack{0*\\1*}}!\, n_{**}!} M 2^{-n_{**}} \left(2^{n_{\substack{00\\11}}} - \frac{n_{\substack{00\\11}}(n_{\substack{00\\11}}+1)}{2} - 1 \right).$$

For 2), we need the following observation: π is an essential prime implicant for f if and only if

a) $f(v) = 1$ for all $v \in \pi$, and, for at least one $v \in \pi$,

b) for every unit vector ξ_j, $f(v \oplus \xi_j) = 0$, whenever $v \oplus \xi_j \notin \pi$.

We use inclusion/exclusion. For $S_i \subseteq \pi_i$, where $i \in \{1, 2\}$, define $N(\geqslant S_1, \geqslant S_2)$ to be the number of functions f such that $f = 1$ on $\pi_1 \cup \pi_2$, and every $v_i \in S_i$ satisfies b), for $1 \leqslant i \leqslant 2$. We want $N(\neq \phi, \neq \phi)$, the number of functions such that $f = 1$ on $\pi_1 \cup \pi_2$ and at least one $v_i \in S_i$ for $i = 1, 2$ satisfies b). By inclusion/exclusion on the second argument, we obtain for the number of functions where every $v_1 \in S_i$ satisfies b), and at least one $v_2 \in S_2$ exists which satisfies b),

$$N(\geqslant S_1,\ \neq \phi) = \sum_{\phi \neq S_2 \subseteq \pi_2} (-1)^{|S_2|-1} N(\geqslant S_1,\ \geqslant S_2).$$

Including the first argument yields

$$N(\neq \phi,\ \neq \phi) = \sum_{S_1 \neq \phi} \sum_{S_2 \neq \phi} (-1)^{|S_1|+|S_2|} N(\geqslant S_1,\ \geqslant S_2). \tag{17}$$

$N(\geqslant S_1, \geqslant S_2)$ can be calculated as follows. If there are $z(\pi_1, \pi_2, S_1, S_2)$ forced 0's in the functions counted in $N(\geqslant S_1, \geqslant S_2)$ and $w(\pi_1, \pi_2)$ forced 1's, then

$$N(\geqslant S_1,\ \geqslant S_2) = \binom{M-z-w}{u-w}.$$

Evaluation of (17) is time consuming because of the many terms in the two sums. Alternatively, we can find an approximation to (17) by retaining only the first terms. Using Bonferroni's inequalities for inclusion/exclusion, restricting the terms to less than a fixed magnitude gives an over- or underestimate, depending on whether the first neglected term is negative or positive. The program which implements this, in fact, terminates evaluation if the magnitude of the terms falls below a threshold.

We have for the number of forced 1's

$$w(\pi_1,\ \pi_2) = \begin{cases} 2^{n_{\substack{*0\\*1}} + n_{**}} + 2^{n_{\substack{0*\\1*}} + n_{**}} - 2^{n_{**}} & \text{for } E_0 \\ 2^{n_{\substack{*0\\*1}} + n_{**}} + 2^{n_{\substack{0*\\1*}} + n_{**}} & \text{otherwise.} \end{cases}$$

Here $2^{n_{\substack{*0\\*1}} + n_{**}}$ and $2^{n_{\substack{0*\\1*}} + n_{**}}$ represent the number of 1's in implicant π_1 and π_2. In the case of 0 disagreements in components counted under $n_{\substack{00\\11}}$, there is overlap in the essential prime implicants amounting to $2^{n_{**}}$ 1's, which must be deducted. However, when there is at least one disagreement, the implicants are disjoint, and the deduction is unnecessary.

The number of 0's forced in b) by $v \in S_1$ is $n_{\substack{00\\11}} + n_{\substack{0*\\1*}}$. Thus for S_1, there are $|S_1|\,(n_{\substack{00\\11}} + n_{\substack{0*\\1*}})$ forced 0's, and for S_2, $|S_2|\,(n_{\substack{00\\11}} + n_{\substack{*0\\*1}})$ forced 0's. However, among these two sets, there may be common 0's. Also, a forced 0 of one implicant may coincide with a 1 of the other, in which case, there are no functions satisfying these contradictory requirements. Consider the enumeration by cases.

G_2: With more than two disagreements in the values of $n_{\substack{00\\11}}$, π_1 and π_2 are disjoint and so are all adjacent 0's. Thus,

$$z=|S_1|(n_{\substack{00\\11}}+n_{\substack{0*\\1*}})+|S_2|(n_{\substack{00\\11}}+n_{\substack{*0\\*1}}).$$

From (17)

$$N(\neq\phi,\ \neq\phi)=\sum_{\substack{s_1\geq 1\\ s_2\geq 1}}(-1)^{s_1+s_2}\binom{2^{n_{\substack{*0\\*1}}+n_{**}}}{s_1}\binom{2^{n_{\substack{0*\\1*}}+n_{**}}}{s_2}$$

$$\cdot\binom{M-2^{n_{\substack{*0\\*1}}+n_{**}}-2^{n_{\substack{0*\\1*}}+n_{**}}-s_1(n_{\substack{00\\11}}+n_{\substack{0*\\1*}})-s_2(n_{\substack{00\\11}}+n_{\substack{*0\\*1}})}{u-2^{n_{\substack{*0\\*1}}+n_{**}}-2^{n_{\substack{0*\\1*}}+n_{**}}}. \tag{18}$$

E_1: If $v \in S_1$ and v agrees with π_2 in the components counted in $n_{\substack{*0\\*1}}$, then $v \oplus \xi_j \in \pi_2$, where j is the component which caused $d = 1$. This is a contradictory requirement, since $f(v \oplus \xi_j) = 0$ by b), and $= 1$ since $v \oplus \xi_j \in \pi_2$. Hence, we may assume S_1 contains no vectors that agree with π_2 in those $n_{\substack{*0\\*1}}$ components. This is a set of size $2^{n_{**}}$. For such S_1 and similar S_2, w and z as before, the expression for $N(\neq\phi, \neq\phi)$ is the same as (17) except that

$$\binom{2^{n_{\substack{*0\\*1}}+n_{**}}}{s_1}$$

is replaced by

$$\binom{2^{n_{\substack{*0\\*1}}+n_{**}}-2^{n_{**}}}{s_1}$$

and

$$\binom{2^{n_{\substack{0*\\1*}}+n_{**}}}{s_2}$$

is replaced by

$$\binom{2^{n_{\substack{*0\\*1}}+n_{**}}-2^{n_{**}}}{s_2}.$$

E_2: For this case, there are no 1's in π_1 adjacent to 1's in π_2, and so no contradictions occur. Let $T_1 \subseteq S_1$ be those vectors v that differ from a vector in S_2 in precisely two places. Since $d=2$, these two places are the two disagreements among the $n_{\substack{00\\11}}$ variables where π_1 is 0 and 1, and π_2 is 0 and 1. Let i and j be the coordinates where the disagreements occur. Then,

$$z=|S_1|(n_{\substack{00\\11}}+n_{\substack{0*\\1*}})+|S_2|(n_{\substack{00\\11}}+n_{\substack{*0\\*1}})-2|T_1|,$$

because, for each $v \in T_1$, both $v \oplus \xi_i$ and $v \oplus \xi_j$ are forced 0's by S_1 and S_2. Thus, when we collect terms in (16), we have a triple sum over $t = |T_1|$, $s_1 = |S_1|$, and $s_2 = |S_2|$.

The number of triples (T_1, S_1, S_2) which gives values (t_1, s_1, s_2) is calculated as follows. A choice for $T_1 \subseteq S_1$ specifies T_2, those vectors in S_2 which differ from vectors in S_1 in two places ($T_2 = \{v \mid$ there is a $u \in T_1$, such that $v = u \oplus \xi_i \oplus \xi_j\}$). Of the $2^{n_{\substack{*0\\*1}}+n_{**}}$ choices for S_1, $2^{n_{**}}$ of them agree with π_2 in the coordinates counted by $n_{\substack{*0\\*1}}$, and are thus possible elements of T_1. Thus, there are $\binom{2^{n_{**}}}{t}$ choices for T_1. If j additional elements from that set are to be in S_1, we have

$$\binom{2^{n_{**}}-t}{j}\binom{2^{n_{\substack{*0\\*1}}}-2^{n_{**}}}{s_1-t-j}\binom{2^{n_{\substack{0*\\1*}}+n_{**}}-t-j}{s_2-t}$$

choices for S_1 and S_2 given T_1, where the factors above count, from left to right, then the number of ways to make a choice of elements, the number of ways to complete the choice of S_1, and the number of ways to complete the choice of S_2. Combining all this yields

$$N(\neq\phi,\ \neq\phi)=\sum_{s_1,s_2>0}\sum_{t\geq 0}\sum_{j\geq 0}(-1)^{s_1+s_2}$$

$$\cdot\binom{2^{n_{**}}}{t}\binom{2^{n_{**}}-t}{j}\binom{2^{n_{\substack{*0\\*1}}+n_{**}}-2^{n_{**}}}{s_1-t-j}$$

$$\cdot\binom{2^{n_{\substack{0*\\1*}}+n_{**}}-t-j}{s_2-t}\binom{M-w-z}{u-w}$$

where

$w=2^{n_{\substack{*0\\*1}}+n_{**}}+2^{n_{\substack{0*\\1*}}+n_{**}}$ and

$$z=(n_{\substack{00\\11}}+n_{\substack{0*\\1*}})s_1+(n_{\substack{00\\11}}+n_{\substack{*0\\*1}})s_1-2t.$$

E_0: Since $d = 0$ and π_1 and π_2 are *distinct prime* implicants, it must be that $n_1 \neq 0$ and $n_2 \neq 0$. Furthermore, $S_1 \cap \pi_2 \neq \phi$. On the contrary, for $v \in S_1 \cap \pi_2$ with $v \oplus \xi_k$

TABLE I
COMMERCIALLY AVAILABLE PROGRAMMABLE LOGIC ARRAYS

Number of Inputs	Number of Product Terms	Number of Outputs	Type of PLA	Manufacturer Comments
8	8	8	Field Progr. Erasable	Altera EP-300
8	32	10	Field Progr.	Signetics PLS153 10 add'l i/o lines
12	32	10	Field Progr.	Signetics PLS173 10 add'l i/o lines
12	48	8	Field Progr.	Signetics PLS161 /179
16	48	8	Field Progr.	Signetics, Motorola, and Silicon Sys.

$\in \pi_2$, we have $f(v \oplus \xi_k) = 1$. However, since $v \in S_1$, by b), $f(v \oplus \xi_k) = 0$, a contradiction. Thus, $v_1 \in S_1$, must disagree with π_2 in $p > 0$ positions in the $n_{*\substack{0\\1}}$ range. Suppose $p = 1$, and the coordinate is i. Let j be a coordinate in the $n_{\substack{0\\1}*}$ range. Then, $v_2 = v \oplus \xi_i \oplus \xi_j$ is a possible element of S_2, and $v_1 \oplus \xi_j = v_2 \oplus \xi_i$ is an overlap of forced 0's.

Thus, we have a more complex situation than before. Let

$$S_i = D_i \cup \left(\bigcup_{\alpha \in \{0,1\}^{n_{**}}} S_i(\alpha) \right)$$

where all $v \in D_i$ have $p > 1$ and all $v \in S_i(\alpha)$ have $p = 1$ and equal α on the n_{**} range. Then,

$$z = |S_1|(n_{\substack{00\\11}} + n_{\substack{0\\1}*}) + |S_2|(n_{\substack{00\\11}} + n_{*\substack{0\\1}}) - \sum_{\alpha} |S_1(\alpha)| - |S_2(\alpha)|.$$

There are

$$\binom{2^{n_{**}}(2^{n_{*\substack{0\\1}}} - 1 - n_{*\substack{0\\1}})}{d_1}$$

different D_1 with $|D_1| = d_1$ and

$$\binom{2^{n_{**}}(2^{n_{\substack{0\\1}*}} - 1 - n_{\substack{0\\1}*})}{d_2}$$

different D_2 with $|D_2| = d_2$. There are $\binom{n_{*\substack{0\\1}}}{\sigma}$ choices for $S_1(\alpha)$ with $|S_1(\alpha)| = \sigma$ and $\binom{n_{\substack{0\\1}*}}{\sigma}$ choices for $S_2(\alpha)$ with $|S_2(\alpha)| = \sigma$, given α. A *composition* of vector (m_1, m_2) with k parts is a sequence of k vectors $(\lambda_{1j}, \lambda_{2j})$, $(1 \leqslant j \leqslant k)$ with $\lambda_{ij} \geqslant 0$ such that both λ_{1j} and λ_{2j} are not 0. Let $C(m, k)$ be the set of such compositions. With any such composition, we can associate a sequence $(|S_1(\alpha)|, |S_2(\alpha)|)$ $(\alpha \in \{0, 1\}^{n_{**}})$ by specifying the k terms in the latter sequence that differ from $(0, 0)$. This can be done in $\binom{2^{n_{**}}}{k}$ ways. Putting this all together

$$N(\neq \phi, \neq \phi) = \sum_{d_1, d_2} (-1)^{d_1 + d_2} \cdot \binom{2^{n_{**}}(2^{n_{*\substack{0\\1}}} - 1 - n_{*\substack{0\\1}})}{d_1} \binom{2^{n_{**}}(2^{n_{*\substack{0\\1}}} - 1 - n_{\substack{0\\1}*})}{d_2}$$

$$\sum_{m_1, m_2} (-1)^{m_1 + m_2} \sum_k \binom{2^{n_{**}}}{k} \cdot \sum_{\lambda \in C(m,k)} \prod_{j=1}^{k} \binom{n_{*\substack{0\\1}}}{\lambda_{1j}} \binom{n_{\substack{0\\1}*}}{\lambda_{2j}} \binom{M - z - w}{u - w}$$

where the sum over m_1 and m_2 is restricted so that $d_i + m_i > 0$, since it equals s_i, and where

$$w = 2^{n_{*\substack{0\\1}} + n_{**}} + 2^{n_{\substack{0\\1}*} + n_{**}} - 2^{n_{**}}$$

and

$$z = (n_{\substack{00\\11}} + n_{\substack{0\\1}*})(d_1 + m_1) + (n_{\substack{00\\11}} + n_{*\substack{0\\1}})(d_2 + m_2) - \sum_{j=1}^{k} \lambda_{1j}\lambda_{2j}.$$

VI. RESULTS

In this section, we show how the results derived in previous sections can be used to predict whether a given function is realized by a PLA. We consider the five commercially available PLA's [7] shown in Table I. In this set, the number of input variables ranges from 8 to 16, while the number of product terms ranges from 8 to 48. The number of outputs is 8 or 10. Our comparison is based on single-output PLA's. A comparison involving more than one output must account for product term sharing, which is not covered by the analysis of this paper.

We begin by comparing the accuracy of the various upper and lower bounds.

A. Comparison Among Various Bounds

A program was written to solve for

1) lower bounds for r-valued functions—(4),
2) improved lower bounds for binary functions—(12),
3) upper bounds for r-valued functions—(13), and
4) improved upper bounds for binary functions—(14).

The results for binary functions with $n = 8$ inputs and u 1's, where $0 \leqslant u \leqslant 256$ are shown in Fig. 2, together with the upper and lower bounds derived by Mileto and Putzolu [9].

The highest curve in Fig. 2 is the upper bound derived by Mileto and Putzolu [9] and is the average number of prime

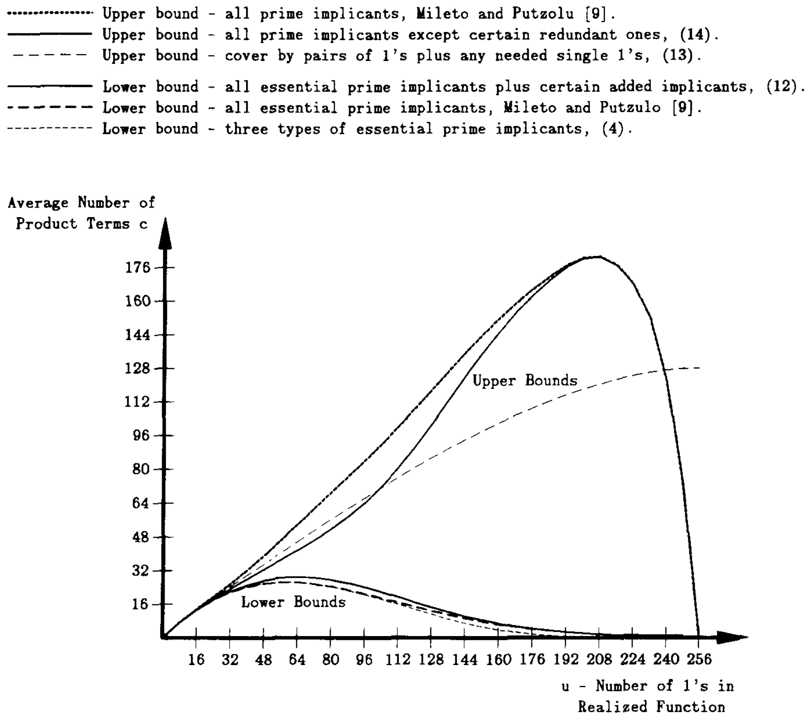


Fig. 2. Upper and lower bounds on the average number of product terms required in the minimal realization of 8-input binary functions versus the number of 1's in the function.

implicants over the set of all functions with u 1's, for $0 \leqslant u \leqslant 256$. The solid curve just below corresponds to the average number of prime implicants less certain redundant prime implicants (derived in Section IV-B). The dashed curve lying mostly below this is an upper bound derived by covering all 1's with implicants consisting of pairs of 1's and single 1's (derived in Section IV-A). When $u = 256$, all cells are 1, and such a covering requires 128 implicants, where one will do. Thus, for this and nearby values of u, this bound is poor. However, for smaller values of u, it is better than the bounds derived from all prime implicants, because of the large number of prime implicants associated with functions where there are approximately as many 1's as 0's. But then, for even smaller numbers of 1's, the restriction to implicants of size two or one in the covering is a disadvantage compared to the two bounds derived by counting unrestricted prime implicants. Therefore, for this case, the latter bounds are better.

Of the three lower bounds, the best is derived by counting essential prime implicants and certain nonessential prime implicants (derived in Section III-B). This is shown as a solid line. The wide dotted line just below it corresponds to essential prime implicants only as derived by Mileto and Putzolu [9]. The thin dotted line below this corresponds to three types of essential prime implicants. There is very little difference between the three lower bounds.

Among *all* bounds for small u, there is also very little difference. It is in this range that the average value can be determined accurately, which we do in the next sections.

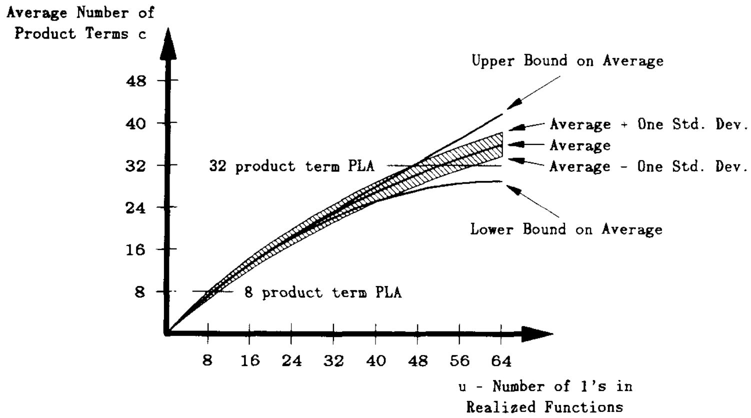
In the following analysis, we use the best bounds possible. For binary functions with small u, the best upper bound is based on a count of all prime implicants less certain redundant prime implicants, while the best lower bound is based on a count of essential prime implicants plus certain nonessential ones. These are indicated by solid lines in Fig. 2, as well as subsequent figures. For r-valued functions with $r > 2$, we use for the upper bound the bound derived by covering nonzero cells with pair and single cells, while, for the lower bound, the bound derived from three types of essential prime implicants.

B. Comparison of Calculated Bounds with Statistically Derived Values

Fig. 3(a) shows the best bounds of Fig. 2 for $0 \leqslant u \leqslant 64$, as well as statistically derived averages. Each point in the latter curve is produced from the average number of product terms required in the minimal realization of 1000 random functions with a fixed number of 1's for $u = 2i$, where $1 \leqslant i \leqslant 32$. The minimal realization was found by a program producing the absolute minimal sum-of-products expression for each function. For each u, the standard deviation was also calculated. The curves corresponding to the average plus and minus one standard deviation are shown in Fig. 3(a), and the area between them is shown by hatching. For each $u = 2i$, $20 \leqslant i \leqslant 32$, at least one random sample was not used because the minimal realization was not resolved. However, the number of unresolved functions was never more than 3.1 percent of the total and was neglected.

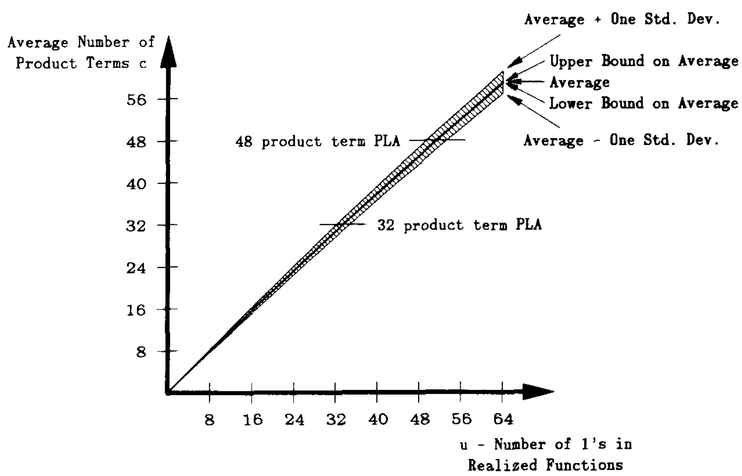
Fig. 3(b) shows the same information for 12-input functions. Unlike the upper and lower bounds for 8-input functions, the

Shaded area represents statistical data from 1000 random functions per point. Shown is the average number of product terms + and - one standard deviation.



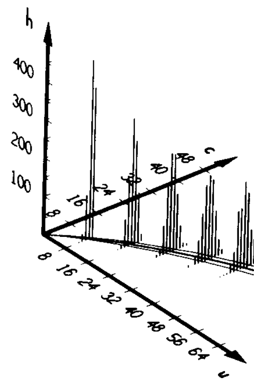
(a)

Shaded area represents statistical data from 1000 random functions per point. Shown is the average number of product terms + and - one standard deviation.

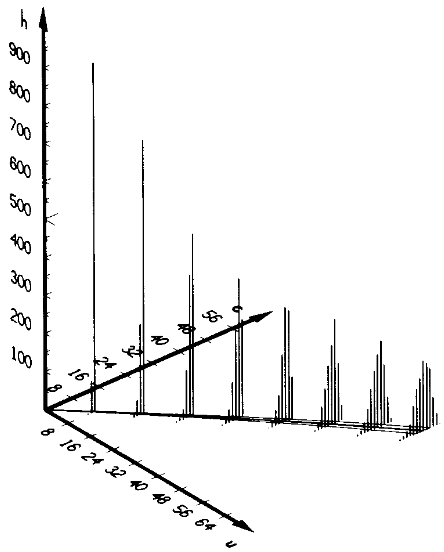


(b)

Fig. 3. Average number of product terms c required in the minimal realization of 8- and 12-input binary functions versus u, the number of 1's in the functions. Shown are upper and lower bounds and statistically derived averages.



(a)



(b)

Fig. 4. Distribution of the number of product terms required in the minimal realization of 8- and 12-input binary functions. The c–u plane shows the average number of product terms c required versus the number of 1's u in the function $\pm$ one standard deviation. The h axis shows the number of sample functions with u 1's requiring c product terms, where each sample set has 1000 functions.

corresponding bounds for 12-input functions are very close to each other over the full range, $0 \leqslant u \leqslant 64$. In fact, they fall within the hatched area bounded by the average plus and minus one standard deviation. Because of the closeness of the bounds, the average, in this case is accurately known. The statistical data in Fig. 3(b) were also generated by sample sets of size 1000. However, in all cases, there were no unresolved functions.

Fig. 4(a) and (b) shows three-dimensional plots of the statistical data. The c–u (horizontal) plane contains the average and the average plus and minus one standard deviation. The vertical axis shows h, the number of samples in each sample set with the corresponding values of c and u. We show a set of h' functions having u' 1's and requiring c' product terms in its minimal realization as a line from $(u, c, h) = (u', c', 0)$ to (u', c', h').

From the data, it can be seen that, when the average number of product terms required in the minimal realization is sufficiently smaller than the number of 1's in the function, the distribution is approximately symmetric about the average. However, for functions with very few 1's, the distribution is skewed, with many functions requiring the maximum number of product terms, while the remaining functions trail off as c decreases.

Fig. 5 shows the plot of the variance derived from the sample set as well as the upper bound derived in Section V for 8- and 12-input binary functions. For 8-input functions, the bound is higher except for a small range of u. However, for 12-input functions, the statistically derived values are consistently higher than the upper bound. It is believed that this is due at least in part to a small sample size. The graininess in the statistical data is thought to be due to the small sample set size, while the graininess in the upper bound curve is thought to be due to truncations of the inclusion/exclusion sums.

Fig. 6 shows upper and lower bounds on the number of product terms in the minimal realization of binary functions on 16 inputs. Computer storage and time restrictions precluded the generation of statistical data for these cases. Also shown are the plots corresponding to the average of the upper and lower bounds plus and minus a value that corresponds to the upper bound on the standard deviation (calculated in Section V).

C. Comparison of the Number of Functions Realized by Commercially Available PLA's

All of the five PLA's listed in Table I are represented by horizontal lines through hatched regions in Figs. 3 and 6. These regions represent areas of concentration of functions in the plot of the number of product terms required in minimal realizations versus the number of minterms. A line corresponding to each PLA divides functions with few 1's into two subsets, those which are realized (below the line) and those which are not (above the line). With the exception of the 8-input 32 product term PLA, the hatched region at the point of intersection is small (because of small variance). Therefore, the number of minterms u in a random function f with few 1's is a statistically strong indicator of the probability that f will be realized. That is, if u is sufficiently larger than σ_T, the abscissa at the point of intersection, it is unlikely that f will be realized. Conversely, if u is significantly smaller, the converse is true. Only in the region near σ_T, does the probability deviate from the extremes. Since the region is small for most PLA's, the threshold between realizability and nonrealizability is sharp. The only exception is the 8-input 32 product term PLA, where a large variance makes u a weak indication of realizability. It should be noted that this analysis does not consider functions with many 1's that are realized by c or fewer (mostly large) product terms.

The small variance is especially notable for 16-input 48 product term PLA's. The region between the upper and lower bounds and between the standard deviation lines closely approximates a single line of slope 45 degrees. Thus, for most functions with (small) values of u that make realization likely, the minimal realization consists of minterms which cannot be

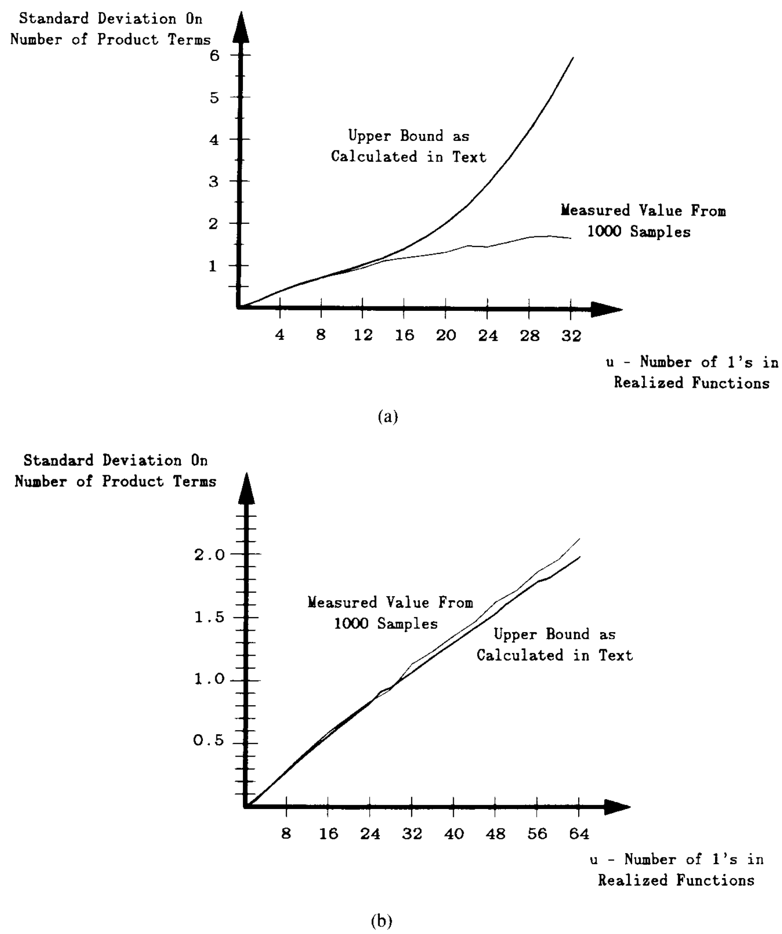


Fig. 5. Upper bound on the standard deviation for the distribution of the number of product terms required in the minimal realization of 8- and 12-input binary functions versus the number of 1's in the function. Shown also is the standard deviation obtained experimentally from 1000 samples per point.

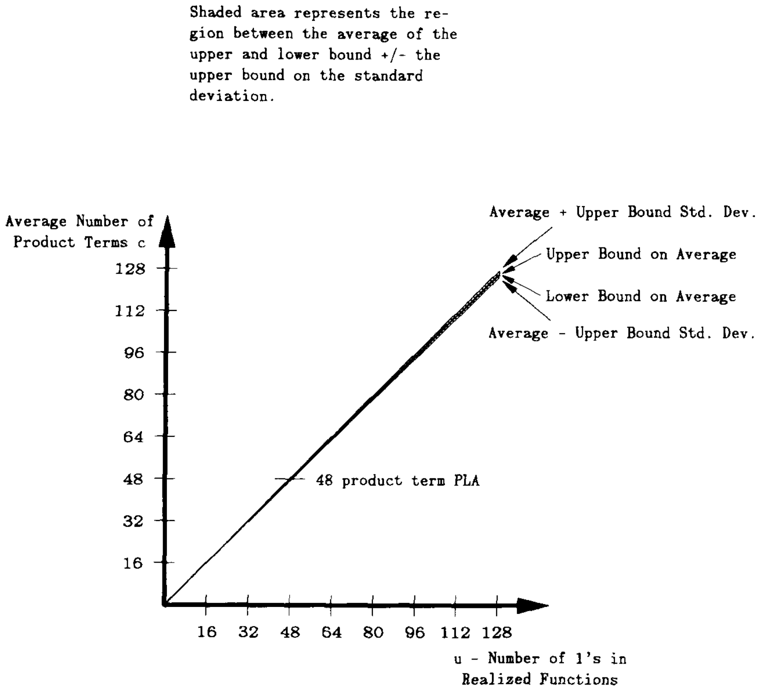


Fig. 6. Average number of product terms c required in the minimal realization of 16-input binary functions versus u, the number of 1's in the functions. Shown are the upper and lower bounds on the average number of product terms and the average of these bounds plus and minus the upper bound on one standard deviation. The area between is hatched.

combined with any other minterm. In this case, *a c product term PLA is, in effect, a content addressable memory, where the stored pattern is the minterm specification and where the number of stored addresses is c.*

Although a PLA may realize only a small fraction of functions with σ_T or more 1's, it may still realize a large number of such functions. For example, the 48 product term PLA on 16 inputs shown in Fig. 6 realizes

$$\sum_{j=1}^{48} \binom{2^{16}}{j} \approx 10^{170}$$

functions with 16 or fewer 1's. However, from Sasao [16], such a PLA realizes at least $3^{48(16-6)} \approx 10^{229}$ different functions. The large difference is due to the fact that there are many more product terms (3^n) than there are product terms involving all n variables (2^n), of which the latter, almost exclusively, are involved in the realization of functions with 48 or fewer 1's.

D. Comparison of Bounds for 4-Valued Functions with Various Distributions of Nonzero Values

Fig. 7 shows the plot of upper and lower bounds on the number of product terms required in the minimal realization of 4-valued PLA's with 4 and 8 inputs. The plots for four distributions of nonzero values are shown below

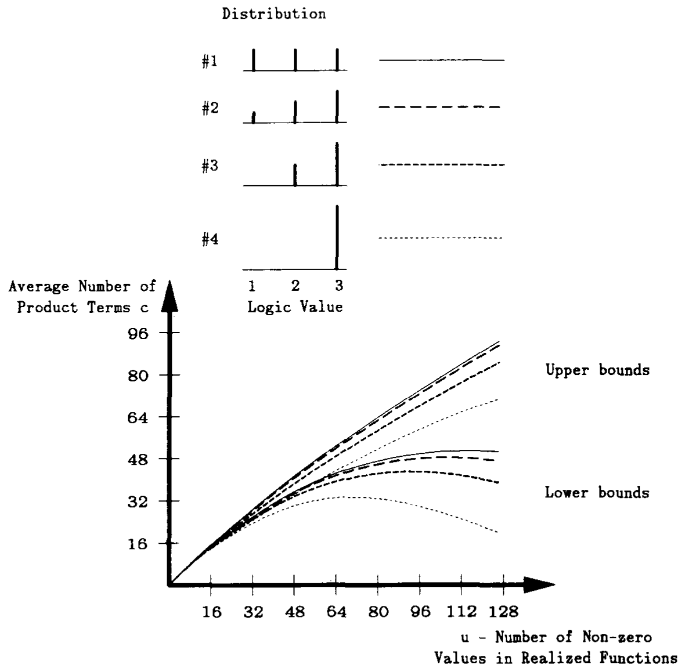
1) $n_3 = n_2 = n_1$ $\quad n_1 = 2i \quad 0 \leqslant i \leqslant 11$,
2) $n_3 = 3n_1,\ n_2 = 2n_1$ $\quad n_1 = i \quad 0 \leqslant i \leqslant 11$,
3) $n_3 = 2n_2,\ n_1 = 0$ $\quad n_2 = 2i \quad 0 \leqslant i \leqslant 11$, and
4) $n_3 = u,\ n_2 = n_1 = 0$ $\quad n_3 = 6i \quad 0 \leqslant i \leqslant 11$.

The plots show that, as the distributions move from skewed to uniform, the upper and lower bounds increase. We would expect this, since skewed distributions have a larger fraction of cells with one nonzero logic value which can be combined with similar cells.

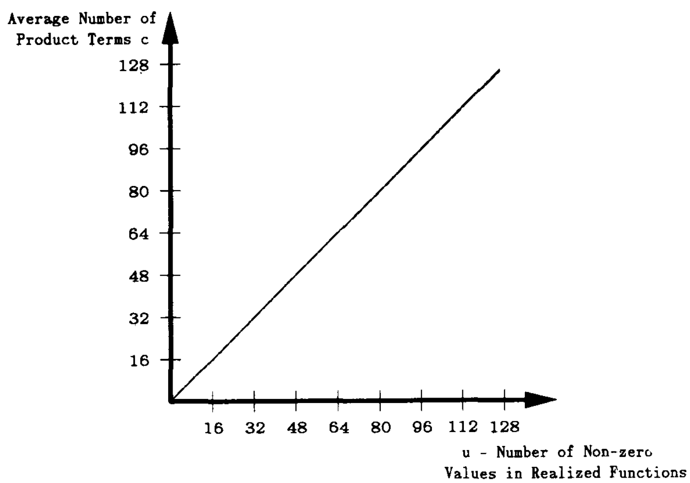
VII. Concluding Remarks

Our approach to the problem of enumerating binary functions realized by programmable logic arrays is to derive upper and lower bounds, as was done in Mileto and Putzolu [9] and Sasao and Terada [19], and to observe that for functions with few 1's, the two bounds are close to each other. However, we extend their results in two ways. First, our bounds are more accurate. Second, we derive bounds which are valid for PLA's where both the inputs and outputs are r-valued, for $r \geqslant 2$. Thus, the results apply to nonbinary logic, where new PLA's are being proposed [16] and implemented [3], [8].

In addition, we derive an upper bound on the variance of the distribution of functions with u 1's over the number of product terms needed in a minimal realization. This, in addition to the average value information, allows an analysis of binary functions with few 1's that are realized by commercially available PLA's.



(a)



(b)

Fig. 7. Upper and lower bounds on the average number of product terms required in the minimal realization of 4-valued functions with PLA's of 4 and 8 inputs.

In spite of the fact that the bounds are most accurate for functions with few 1's, our analysis yields an interesting result for almost all commercially available PLA's. Because of the small variance, we can make the following statements about functions with few 1's. There is a threshold σ_T, dependent on the PLA, such that, if an arbitrary function f has more 1's than σ_T, it is unlikely to be realized by the PLA. Conversely, if f has fewer than σ_T 1's, it is likely to be realized. For all but one PLA, the threshold is sharp, in the sense that there is only a narrow range around σ_T for which such a strong statement cannot be made. For PLA's with many inputs, σ_T is close to the number of product terms. Thus, if a function has more than σ_T minterms, it is unlikely to be realized, while, if the function has no more than σ_T minterms, it is unlikely that the minterms will combine. The PLA is, in effect, a content addressable memory.

Acknowledgment

The authors are grateful for the comments provided by the referees.

References

[1] E. A. Bender, J. T. Butler, and H. G. Kerkhoff, "Comparing the SUM with the MAX for use in four valued PLA's," in *Proc. Int. Symp. Multiple-Valued Logic,* Kingston, Ont., Canada, May 1985, pp. 30-35.

[2] E. A. Bender and J. R. Goldman, "On the application of Möbius inversion in combinatorial analysis," *Amer. Math. Monthly,* vol. 82, pp. 789-803, Oct. 1975.

[3] J. T. Butler and H. G. Kerkhoff, "Analysis of input and output configurations for use in four-valued programmable logic arrays," *IEE Proc.*, vol. 134, pt. E, pp. 168-176, July 1987.

[4] R. K. Brayton, G. D. Hachtel, C. T. McMullen, and A. Sangiovanni-Vincentelli, *Logic Minimization Algorithms for VLSI Synthesis.* Boston, MA: Kluwer Academic, 1984.

[5] A. H. Chan, "Using decision trees to derive the complement of a binary function with multiple-valued inputs," *IEEE Trans. Comput.,* vol. C-36, pp. 212-214, Feb. 1987.

[6] V. V. Glagolev, "Some bounds for disjunctive normal forms of the algebra of logic," *Problemi Kibernetiki,* vol. 19, pp. 74-93, 1970 (English translation, Systems Theory Research, Consultants Bureau, NY).

[7] *IC Master,* New York: United Technical Pubs., 1986, pp. 4473-4474.

[8] H. G. Kerkhoff and J. T. Butler, "Design of a high-radix Programmable logic array using profiled peristaltic charge-coupled devices," in *Proc. Int. Symp. Multiple-Valued Logic,* Blacksburg, VA, May 1986, pp. 128-136.

[9] F. Mileto and G. Putzolu, "Average values of quantities appearing in Boolean function minimization," *IEEE Trans. Electron. Comput.,* vol. EC-13, pp. 87-92, Apr. 1964.

[10] ——, "Average values of quantities appearing in multiple output Boolean minimization," *IEEE Trans. Electron. Comput.,* vol. EC-14, pp. 542-552, Aug. 1965.

[11] ——, "Statistical complexity of algorithms for Boolean function minimization," *J.ACM,* vol. 12, pp. 364-375, July 1965.

[12] S. Muroga, *Logic Design and Switching Theory.* New York: Wiley, 1979.

[13] T. Sasao, "Multiple-valued decomposition of generalized Boolean functions and the complexity of programmable logic arrays," *IEEE Trans. Comput.,* vol. C-30, pp. 635-643, Sept. 1981.

[14] ——, "Input variable assignment and output phase optimization of PLA's," *IEEE Trans. Comput.,* vol. C-33, pp. 879-894, Oct. 1984 (also Corrections and additions to ... by T. Sasao, Dep. Elec. Eng., Osaka Unit., Osaka 565, Japan).

[15] ——, "An algorithm to derive the complement of a binary function with multiple-valued inputs," *IEEE Trans. Comput.,* vol. C-34, pp. 131-140, Feb. 1985.

[16] ——, "On the optimal design of multiple-valued PLA's," in *Proc. Int. Symp. Multiple-Valued Logic,* Blacksburg, VA, May 1986, pp. 214-223.

[17] ——, *Programmable Logic Arrays: How to Make and How to Use* ISBN4-526-02026-5 C3054, Nikkan Kogyou 1986 (in Japanese).

[18] T. Sasao and H. Terada, "Multiple-valued logic and the design of programmable logic arrays with decoders," in *Proc. Int. Symp. Multiple-Valued Logic,* Bath, England, May 1979, pp. 27-37.

[19] ——, "On the complexity of shallow logic functions and the estimation of programmable logic array size," in *Proc. Int. Symp. Multiple-Valued Logic,* Evanston, IL, May 1980, pp. 65-73.

Edward A. Bender was born in Brooklyn, NY, in 1942. He received the B.S. degree in mathematics with honors from the California Institute of Technology, Pasadena, in 1963 and the Ph.D. degree in mathematics in 1966, also from Caltech.

He was an instructor at Harvard and a research mathematician at the Communications Research Division of the Institute for Defense Analyses. In 1974, he joined the Department of Mathematics of the University of California, San Diego, where he is currently a Professor. His primary research interest is enumerative combinatorial analysis and its applications in computer science, but he also dabbles in algebraic matrix theory and ecology. Much of his recent work has dealt with the asymptotic enumeration of graphs and maps. He is the author of *An Introduction to Mathematical Modeling* and has published over 70 technical papers.

Jon T. Butler (S'67–M'67–SM'82–F'89) was born on December 26, 1943 in Baltimore, MD. He received the B.E.E. and M.Engr. (E.E.) degrees from Rensselaer Polytechnic Institute, Troy, NY in 1966 and 1967, respectively, and the Ph.D. degree from The Ohio State University, Columbus, in 1973.

Since 1987, he has been a Professor at the Naval Postgaduate School, Monterey, CA. From 1974 to 1987, he was at Northwestern University, Evanston, IL. During that time he served two periods of leave at the Naval Postgraduate School, first as a National Research Council Senior Postdoctoral Associate (1980-1981) and second as the NAVELEX (former name) Chair Professor (1985-1987). His research interests include multiple-valued logic and reliable multiprocessing systems.

Dr. Butler was the Chairman of the 1980 International Symposium on Multiple-Valued Logic and a co-guest editor of a special issue on multiple-valued logic for the IEEE TRANSACTIONS ON COMPUTERS. He was the first Chairman of the Computer Society's Technical Committee on Multiple-Valued Logic and an Editor of the IEEE TRANSACTIONS ON COMPUTERS. He served as the Guest Editor of a special issue on multiple-valued logic in COMPUTER. He received the Award of Excellence and the Outstanding Contributed Paper Award for papers presented at the International Symposium on Multiple-Valued Logic.