

**A New Approach to Realize Partially
Symmetric Functions**

by

Joseph Ja'Ja' and Sau-Mou Wu

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 1986		2. REPORT TYPE		3. DATES COVERED 00-00-1986 to 00-00-1986	
4. TITLE AND SUBTITLE A New Approach to Realize Partially Symmetric Functions				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Maryland,Electrical Engineering Department,College Park,MD,20742				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT see report					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 24	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

A New Approach to Realize Partially Symmetric Functions ¹

Joseph Ja'Ja'

Department of Electrical Engineering
and
Institute For Advanced Computer Studies
University of Maryland
College Park, Md 20740

Sau-Mou Wu

Department of Electrical Engineering
University of Maryland
College Park, Md 20740

Abstract

In this paper, we consider the class of partially symmetric functions and outline a method to realize them. Each such function can be expressed as a sum of totally symmetric functions such that a circuit can be designed whose complexity depends on the size of such symmetric cover. We compare the sizes of symmetric and sum-of-product covers and show that the symmetric cover will be substantially smaller for this class of functions. We also establish bounds on the area required to realize these circuits in a reasonable layout model of VLSI. Our results show that these layouts will be considerably smaller than the corresponding PLA's for the partially symmetric functions.

¹Supported in part by NSA Contract No. MDA_904_85H_0015, NSF Grant No. DCR_86_00378 and by the Systems Research Center Contract No. OIR_85_00108.

1 Introduction

The problem of the automated design of good circuits for a given set of Boolean functions has been studied extensively in the literature (e.g. [1–5]). Researchers have attempted to reach optimal solutions under some cost measures associated with different technologies. In spite of these efforts, optimal circuits are known only for either specific functions such as those arising in integer addition and integer multiplication, or few specialized classes of functions. This has led people to develop heuristic techniques [1,5–8] that work reasonably well in practice without achieving any optimality criterion.

In this paper, we study the properties of *partially symmetric functions* [9–11] and describe a new method to realize them. We try to express each such function as a sum of *totally symmetric functions* and show that a circuit can be designed with the complexity dependent on the size of such *symmetric cover*. We compare the sizes of symmetric covers and sum-of-product covers and show that the symmetric cover will be substantially smaller if the functions have a reasonable degree of symmetry. If the given functions have no symmetry, the symmetric cover reduces to a sum-of-product cover. We also establish bounds on the *area* required to realize these circuits in a reasonable layout model of VLSI [12,13] and show that these layouts will be considerably smaller than the corresponding PLA's if the functions have a moderate degree of symmetry. We use the PLA's to

compare our results because they are widely used in practice.

The rest of the paper is organized as follows. In the next section, we introduce the class of partially symmetric functions and describe some of their properties. The class of totally symmetric functions are considered as a special case. A comparison between the sizes of symmetric covers and sum-of-product covers is made in section 3, while section 4 is reserved for the layout scheme for our circuits and for establishing bounds on the corresponding areas.

2 Basic Properties and Realization Scheme

We introduce in this section the class of the partially symmetric functions [9–11] and establish some of their properties. We later show how to take advantage of the symmetric properties when realizing these functions. First of all, we start by reviewing some well-known definitions from switching theory [14].

Let $f(x_1, \dots, x_n)$ and $g(x_1, \dots, x_n)$ be n -variable Boolean functions. f is said to *cover* g if $f = 1$ whenever $g = 1$. Let h be a product term of the literals of a subset of $\{x_1, \dots, x_n\}$ where by literal we mean the appearance of a complemented or uncomplemented variable. h is said to be an *implicant* of f if f covers h ; h is a *prime implicant* of f if h is an implicant of f such that the deletion of any literal from it results in a new product which is not covered by f . A *cover* C of f is set of implicants of f such that C covers f ;

C is said to be a *minimum cover* of f if all the implicants it contains are prime implicants and the number of these prime implicants is minimum.

2.1 Partially symmetric functions

Let $f(x_1, x_2, \dots, x_n)$ be a Boolean function and let $\rho = \{X_1, X_2, \dots, X_s\}$ be a partition of $\{x_1, x_2, \dots, x_n\}$ such that $|X_i| = n_i$, $1 \leq i \leq s$ and $\sum_{i=1}^s n_i = n$. f is called ρ -symmetric if

$$f(X_1, X_2, \dots, X_s) = f(X'_1, X'_2, \dots, X'_s)$$

where $X'_i = \Pi_i(X_i)$, and Π_i is an arbitrary permutation on X_i .

Let $w(X)$ be the weight function (counting the number of 1's) over vector X . It is obvious that f can be determined by $w(X_i) = k_i$, $1 \leq i \leq s$. Define $C(f)$ to be

$$C(f) = \{ (k_1, k_2, \dots, k_s) \mid f = 1 \text{ for } w(X_i) = k_i, 1 \leq i \leq s \}$$

Let $(k_1, k_2, \dots, k_s) \in C(f)$ and let

$$\sigma_{(k_1, k_2, \dots, k_s)}(X) = \sigma_{k_1}(X_1) \sigma_{k_2}(X_2) \dots \sigma_{k_s}(X_s)$$

where $\sigma_{k_i}(X_i)$ is the k_i th elementary symmetric function on X_i , which equals to 1 when $w(X_i) = k_i$. Clearly,

$$f(x_1, x_2, \dots, x_n) = \sum_{\underline{k}=(k_1, k_2, \dots, k_s) \in C(f)} \sigma_{\underline{k}}(X)$$

After manipulating the above equation, we can rewrite f as follows:

$$f(x_1, x_2, \dots, x_n) = \sum f_{i_1}(X_1) f_{i_2}(X_2) \dots f_{i_s}(X_s) \quad (1)$$

where

$$f_{i_j}(X_j) = \sum_k \sigma_k(X_j) \text{ for certain value of } k. \quad (2)$$

Generalizing the concepts of prime implicants and minimum realization, we define the following terms.

Definition 2.1 *The product term $P = f_{i_1}(X_1) \dots f_{i_s}(X_s)$ in equation(1) will be called a symmetric implicant of f if $f = 1$ whenever $P = 1$. Let each $f_{i_j}(X_j)$ in P be called an symmetric component. Then P will be called a symmetric prime implicant if P is a symmetric implicant and there is no other symmetric implicant P' with fewer symmetric components such that $P' = 1$ whenever $P = 1$.*

Definition 2.2 *A symmetric cover C_s of a function f is a set of symmetric implicants such that C_s covers f ; C_s is said to be a minimum symmetric cover of f if all the symmetric implicants it contains are symmetric prime implicants and the number of these symmetric prime implicants is minimum.*

Definition 2.3 *Equation(1) will be called a minimum symmetric realization if the product terms in it form a minimum symmetric cover.*

It is not hard to show the following lemma.

Lemma 2.1 *Let f be a partially symmetric function with respect to the symmetric partition $\{X_1, \dots, X_s\}$. Then there exists a minimum symmet-*

ric realization of f such that each of the products is a symmetric prime implicant.

Without loss of generality, assume $n_i \geq 2$ for $1 \leq i \leq t$, $t \leq s$ and $n_j = 1$ for $t + 1 \leq j \leq s$. Then equation(1) can be rewritten as follows.

$$f(x_1, x_2, \dots, x_n) = \sum f_{i_1}(X_1) f_{i_2}(X_2) \dots f_{i_t}(X_t) x_{i_{t+1}}^{(k_{t+1})} \dots x_{i_s}^{(k_s)}. \quad (3)$$

Let $Y_i = f_{i_1}(X_1) f_{i_2}(X_2) \dots f_{i_t}(X_t)$.

$$f(x_1, x_2, \dots, x_n) = \sum Y_i \cdot x_{i_{t+1}}^{(k_{t+1})} \dots x_{i_s}^{(k_s)}$$

It follows that equation(3) can be realized by a structure that has three main parts (figure(1)): (i) a set of ES blocks which generate all the elementary symmetric functions required by $f_{i_j}(X_j)$'s, (ii) a NAND plane which realizes Y_i 's, and (iii) a PLA handling those single variable partitions (AND plane) and generating the outputs of the given functions (OR plane). Figure(2) shows the circuit diagram for the NAND plane; the realization scheme for the ES blocks will be discussed later.

2.2 Totally Symmetric Functions

As a special case of partially symmetric functions, a function is said to be totally symmetric if in equation(1), $s = 1$, i.e. there is only one symmetric partition which contains all the variables. Formally,

Definition 2.4 A Boolean function $f(x_1, \dots, x_n)$ is symmetric if for any permutation $\pi \in S_n$, $f(x_1, \dots, x_n) = f(x_{\pi(1)}, \dots, x_{\pi(n)})$.

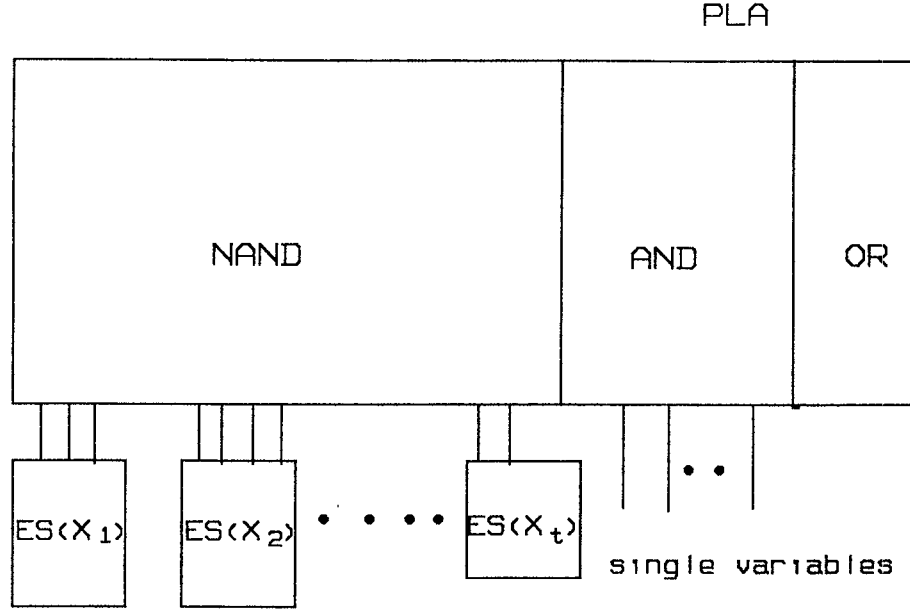


Figure 1: Realization Scheme for Partially Symmetric Functions

Let $X = (x_1, \dots, x_n)$ be a binary n -tuple and let the weight $w(X)$ of X be the number of 1's in X . From the definition, we can see that the value of a symmetric function $f(X)$ depends on the weight $w(X)$; in other words, f has the same value on all binary n -tuples that have the same weight. Therefore, if $f(X)$ has value a_i , $a_i \in \{0, 1\}$ on n -tuples X of weight i , for $0 \leq i \leq n$, then $f(X)$ can be written in the form

$$f(X) = \sum_{i=0}^n a_i \bullet \sigma_i(X)$$

where $\sum$ and $\bullet$ denote AND and OR respectively, and the i -th elementary symmetric function, $\sigma_i(x_1, \dots, x_n)$ equals to 1 when $w(x_1, \dots, x_n) = i$.

Let $F = \{f_i(x_1, \dots, x_n)\}_{i=1}^t$ be a set of symmetric Boolean functions and $I \subseteq \{0, \dots, n\}$ be such that $a_i = 1$ for all $i \in I$. Then as shown

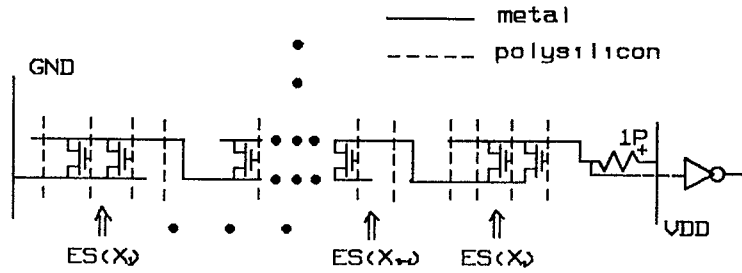


Figure 2: Circuit diagram of NAND plane

in figure(3) a realization of F will consist of two parts: (1) a weight(or counting) function $f_w(x_1, \dots, x_n)$ that counts the number of 1's among the x 's and outputs its binary expansion ; (2) a weight detector that takes the output of the weight function and produces the set F corresponding to I .

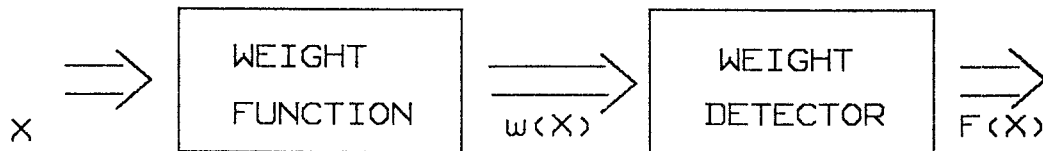


Figure 3: Scheme to realize symmetric fucntions

3 Comparison of the symmetric and the sum-of-product covers

As discussed in the previous section, any partially symmetric function can be manipulated into the form of equation(1) whose product terms are *symmetric prime implicants* which form a *minimum symmetric cover*. In this section, we compare the size of this realization with that of the minimum sum-of-product realization.

Let $c_s(f)$ be the number of symmetric prime implicants in the *minimum symmetric realization* of equation (1) and let $c(f)$ be the number of prime implicants in a minimum sum-of-product cover of f . We will make a comparison of $c_s(f)$ with $c(f)$ and show that $c_s(f)$ will be much smaller than $c(f)$ whenever f has a reasonable degree of symmetry.

we first introduce the definitions and establish some properties of convex blocks and prime convex blocks [11]

3.1 Convex Blocks

Definition 3.1 A convex block, denoted as $w^{[p,q]}(X)$ is a Boolean function defined as follows:

$$\omega^{[p,q]}(X) = \begin{cases} 1, & \text{if } \text{weight}(X) \in [p, q]; \\ 0, & \text{otherwise} \end{cases}$$

$W_i = \omega^{[p_i, q_i]}(X)$ is a prime convex block of a function f if there is no other convex block $W_j = \omega^{[p_j, q_j]}(X)$ such that $W_i = 1 \implies W_j = 1 \implies f = 1$.

Lemma 3.1 *All the implicants of an n -variable convex block $\omega^{[p,q]}(X)$ has u uncomplemented literals and v complemented literals where $u \in [p, q]$ and $n - q \leq v \leq n - u$.*

Proof: The bound for u follows directly from the definition. The upper bound of v is also trivial. Now we check the lower bound of v .

Suppose in an implicant P , $v < n - q$. Then there are $(n - u - v)$ out of the n variables not appearing in P . When we expand P into its minterms, each of the $(n - u - v)$ new variables may be either 0 or 1 and their weights may range from 0 to $n - u - v$. Then the resulting weight of minterms of P may be $u + (n - u - v) = n - v > q$, a contradiction. Thus $v \geq n - q$. $\diamond$

Lemma 3.2 *All the prime implicants in the convex block, $\omega^{[p,q]}(X)$, contain p uncomplemented literals and $n - q$ complemented literals. Moreover, every one of them covers 2^{q-p} minterms.*

Proof: Let P be the set of product terms containing p uncomplemented and $n - q$ complemented literals. From previous lemma, we see that P is a subset of implicants of $\omega^{[p,q]}(X)$. Since every element of P has the least number of literals, it is clear that no two of them could be combined to reduce the number of literals. On the other hand, it is not hard to check that any implicant of $\omega^{[p,q]}(X)$ not in P must be covered by at least one element of P . Thus P is the set of all the prime implicants.

To prove the bound on the number of minterms covered by a prime implicant of $\omega^{[p,q]}(X)$, notice that since every prime implicant contains p uncomplemented and $n - q$ complemented literals, there are $(q - p)$ free variables. Hence 2^{q-p} minterms can be deduced from a prime implicant. $\diamond$

Theorem 3.1 *Given a convex block $\omega^{[p,q]}(X)$, the size of the minimum cover, $C(\omega^{[p,q]}(X))$, satisfies*

$$C(\omega^{[p,q]}(X)) \geq 2^{p-q} \sum_{i=p}^q \binom{n}{i}$$

Proof: It is clear that there are totally $\sum_{i=p}^q \binom{n}{i}$ minterms covered by $\omega^{[p,q]}(X)$ and from lemma 3.2, a prime implicant covers 2^{q-p} minterms. Therefore,

$$C(\omega^{[p,q]}(X)) \geq \sum_{i=p}^q \binom{n}{i} / 2^{q-p} \quad \diamond$$

3.2 Comparison

Recall the minimum symmetric realization of equation(1).

$$f(x_1, \dots, x_n) = \sum_{i=1}^{c_s(f)} f_{i_1}(X_1), \dots, f_{i_s}(X_s)$$

It is obvious that $f_{i_j}(X_j)$ can be expressed in form of convex blocks and f can be rewritten as

$$f(X_1, X_2, \dots, X_s) = \sum_{i=1}^{c'_s(f)} W_i \tag{4}$$

where

$$W_i = \prod_{j=1}^s \omega_j^{[v_{ij}, t_{ij}]}(X_j)$$

It is clear that $c_s(f) \leq c'_s(f)$. Now we show the following.

Lemma 3.3 *Suppose in the equation(4) there exist at least one pair of intervals $[v_{ik}, t_{ik}]$ and $[v_{jk}, t_{jk}]$ in W_i and W_j respectively, for all $i \neq j$, such that they are nonoverlapping and non-adjacent, then the W_i 's are prime convex blocks of f and $\text{Minterm}(W_i) \cap \text{Minterm}(W_j) = \emptyset$, where $\text{Minterm}(W_i)$ is the set of all the minterms of W_i .*

Proof: Suppose not. Let m be a minterm covered by both W_i and W_j . Then for some k , $[v_{ik}, t_{ik}]$ and $[v_{jk}, t_{jk}]$ are not adjacent and nonoverlapping. Assume, without loss of generality, that $v_{jk} > t_{ik}$. Since $m \in \text{Minterm}(W_i)$, m has at least $n_i - t_{ik}$ complemented variables from X_k . On the other hand, $m \in \text{Minterm}(W_j)$ implies that at most $n_i - v_{jk}$ complemented variables from X_k . But $n_i - v_{jk} < n_i - t_{ik}$, a contradiction and hence the claim follows. $\diamond$

Theorem 3.2 *Let f be a function partially symmetric over $\{X_1, \dots, X_s\}$ with $|X_i| = n_i$, such that there is at least one j such that $n_j \geq 2$, for $1 \leq i, j \leq s$. Suppose that f is expressed in form of equation(4) and that there exist at least a pair of intervals $[v_{ik}, t_{ik}]$ and $[v_{jk}, t_{jk}]$ which are not adjacent and nonoverlapping for every i and j , $i \neq j$. If we replace each $\omega^{[v_{ij}, t_{ij}]}(X_j)$ with its minimum cover and apply the distributive law,*

the products obtained are all prime implicants of f . Moreover, they form a minimum cover.

Proof: We start by observing the following fact. Let $\{p_{1i}\}$ and $\{p_{2j}\}$ be the minimum covers of $f_1(X_1)$ and $f_2(X_2)$ respectively, where $X_1 \cap X_2 = \emptyset$. Then $\{p_{1i}, p_{2j}\}$ is a minimum cover of prime implicants of $f_1(X_1)f_2(X_2)$.

It follows that if we replace each ω with its minimum cover, the products obtained from $\prod_{j=1}^s \omega^{[v_{ij}, t_{ij}]}(X_j)$ form a minimum cover of W_i , $1 \leq i \leq c'_s(f)$. Moreover, from the given conditions of disjointness and nonoverlappingness, lemma 3.3 ensures W_i and W_j are prime convex blocks and $Minterm(W_i) \cap Minterm(W_j) = \emptyset$, $\forall i \neq j$.

We now show that each element p of the minimum cover of W_i is a prime implicant of f . Suppose not. Then there exists an implicant q of f such that either $p = 1 \Rightarrow q = 1 \Rightarrow f = 1$ or $p + q$ is an implicant of f . In the first case, q covers some minterms of f . By the disjointness property, q can only cover minterm belonging to one W_j , for some j . It is then easy to check that the first case can not happen. The second case implies that W_i and W_j are adjacent, which is impossible.

Using the above claim, it is easy to check that the resulting prime implicants form a minimum cover. $\diamond$

Corollary 3.1 *Let $c(f)$ be the size of the minimum cover of f and let $c_s(f)$ be the size of the minimum symmetric cover. Then*

$$c(f) \geq \sum_{i=1}^{c_s(f)} \prod_{j=1}^s \left[2^{v_{ij}-t_{ij}} \sum_{k=v_{ij}}^{t_{ij}} \binom{n_j}{k} \right]$$

Example 1: Suppose $f(x_1, \dots, x_{12})$ is partially symmetric over the symmetric partition $(X_1, X_2, X_3, x_{10}, x_{11}, x_{12})$, where $\|X_1\| = 3$, $\|X_2\| = 2$ and $\|X_3\| = 4$. Let f be given by:

$$\begin{aligned} f(X_1, X_2, X_3, x_{10}, x_{11}, x_{12}) = & \omega^{[0,2]}(X_1)\omega^{[0,1]}(X_2)\omega^{[2,3]}(X_3)\bar{x}_{10}x_{11}x_{12} \\ & + \omega^{[2,3]}(X_1)\omega^{[1,2]}(X_2)\omega^{[1,3]}(X_3)x_{10}\bar{x}_{11}x_{12} \\ & + \omega^{[1,3]}(X_1)\omega^{[2,2]}(X_2)\omega^{[0,2]}(X_3)\bar{x}_{10}x_{11}\bar{x}_{12} \end{aligned}$$

Here, $c_s(f) = 3$ and by the above corollary,

$$\begin{aligned} c(f) \geq & 2^{0-2} \left[\binom{3}{0} + \binom{3}{1} + \binom{3}{2} \right] \cdot 2^{0-1} \left[\binom{2}{0} + \binom{2}{1} \right] \cdot 2^{2-3} \left[\binom{4}{2} + \binom{4}{3} \right] \\ & + 2^{2-3} \left[\binom{3}{2} + \binom{3}{3} \right] \cdot 2^{1-2} \left[\binom{2}{1} + \binom{2}{2} \right] \cdot 2^{1-3} \left[\binom{4}{1} + \binom{4}{2} + \binom{4}{3} \right] \\ & + 2^{1-3} \left[\binom{3}{1} + \binom{3}{2} + \binom{3}{3} \right] \cdot 2^{2-2} \left[\binom{2}{2} \right] \cdot 2^{0-2} \left[\binom{4}{0} + \binom{4}{1} + \binom{4}{2} \right] \end{aligned}$$

$$\Rightarrow c(f) \geq 30 \quad \diamond$$

Corollary 3.2 *Let f be a totally symmetric function which takes the value 1 over the maximal and nonoverlapping intervals $[a_1, b_1], \dots, [a_t, b_t]$. Then $c_s(f) = 1$ and*

$$c(f) \geq \sum_{i=1}^t \left[2^{a_i-b_i} \sum_{k=a_i}^{b_i} \binom{n}{k} \right]$$

Example 2: Let f be the parity function on n variables. Then $a_1 = b_1 = 1$, $a_2 = b_2 = 3 \dots$. Hence

$$c(f) \geq \sum_{i \geq 0} \binom{n}{2i+1}$$

If n is odd, the above bound gives $c(f) \geq 2^{n-1}$ (actually an exact bound can be derived for this function). $\diamond$

Example 3: Let n be even and let f be the following symmetric function

$$f(x_1, \dots, x_n) = \begin{cases} 1 & \text{if the number of 1's is } \leq \frac{n}{2} \\ 0 & \text{otherwise} \end{cases}$$

Then,

$$\begin{aligned} c(f) &\geq 2^{-\frac{n}{2}} \sum_{k=0}^{\frac{n}{2}} \binom{n}{k} \\ &\geq 2^{-\frac{n}{2}} \cdot 2^{n-1} = 2^{\frac{n}{2}-1} \end{aligned}$$

As a conclusion, we observe the following facts:

1. The size of the symmetric minimum cover is always less than or equal to that of sum-of-product minimum cover.
2. The size of the minimum symmetric cover will be substantially smaller than that of the minimum cover if the function has a high degree of symmetries (i.e. s is small).
3. When there is no symmetry at all, $c_s(f) = c(f)$.

4 Layout of Partially Symmetric Functions

In this section we will describe how to lay out partially symmetric functions based on the realization schemes proposed in the previous sections. We consider the grid model [12] where the processing elements are mapped onto the grid points and wires can only run along grid lines such that no two wires overlap.

4.1 Layout for totally symmetric functions

We recall that any set of totally symmetric functions can be realized by two main parts, namely, weight function tree and weight detector (figure(3)). In what follows, we will discuss how to lay them out efficiently.

4.1.1 Weight function

The weight function, $f_w : \{0, 1\}^n \rightarrow \{0, 1\}^{\lceil \log(n+1) \rceil}$, counts the number of 1's among the input variables and outputs its binary expansion. For simplicity, assume $n = 2^k - 1$. It is well-known that f_w can be realized by a divide-and-conquer strategy as follows.

$$f_w^{(2^k-1)}(x_1, \dots, x_n) = f_w^{(2^{k-1}-1)}(x_1, \dots, x_{2^{k-1}-1}) + f_w^{(2^{k-1})}(x_{2^{k-1}}, \dots, x_{2^k-1}) \quad (5)$$

If we take full (or half) full adder as base operator, then equation(5) can be implemented as in figure(4).

To lay out this tree, we require that all the inputs and outputs be on

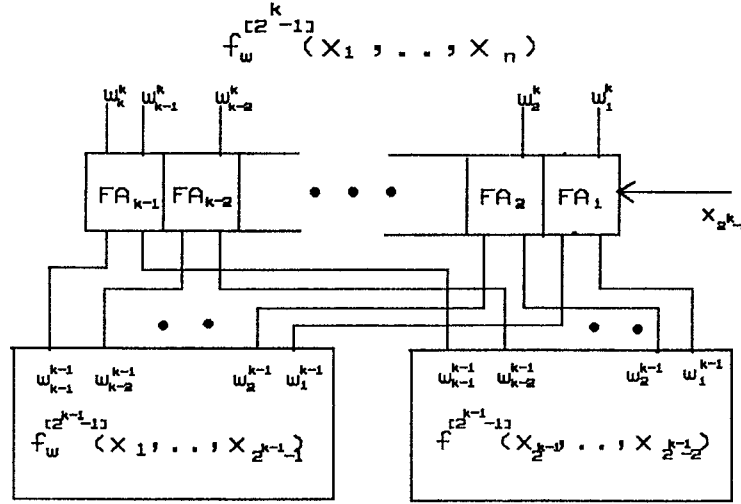


Figure 4: Realization of weight function tree

the boundary and that the enclosing rectangle be as close to a square as possible.

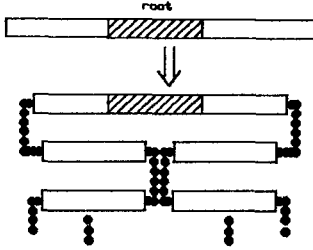
Theorem 4.1 *It is possible to lay out the tree of the weight function of n variables in a square of size $O(\sqrt{n} \log n)$ such that all the inputs and outputs lie on the boundary.*

Proof: We first show how to lay out the tree linearly in an area of $O(n \log^2 n)$ and then show how to turn it into square.

Lay out the tree of (say) $(k-1)$ -adjacent adders in the middle. Recusively, lay out the left subtree on the left side of the root and the right subtree on the right side. The length of the layout is clearly $O(n)$. Its height $H(k)$, i.e. the number of horizontal tracks, is given by the recurrence

$$H(k) \leq H(k-1) + k - 1 \quad \text{and} \quad H(2) = 1$$

It then follows that $H(k) = O(k^2) = O(\log^2 n)$.



Next, as illustrated on the left, we fold the linear layout into a square with the root on the top row. This folding procedure does not increase the area by more than a factor of 2 [12]. Thus we can make the layout into a square with the desired properties. $\diamond$

4.1.2 Weight detector

Receiving the outputs of the weight function tree, the weight detector generates the values of the given functions. Let $m = \lceil \log_2(n+1) \rceil$ and $I \subseteq \{0, 1, \dots, n\}$, such that $a_i = 1$ for all $i \in I$. Then $f_p : \{0, 1\}^{\lceil \log_2(n+1) \rceil} \rightarrow \{0, 1\}$, can be written in the following form.

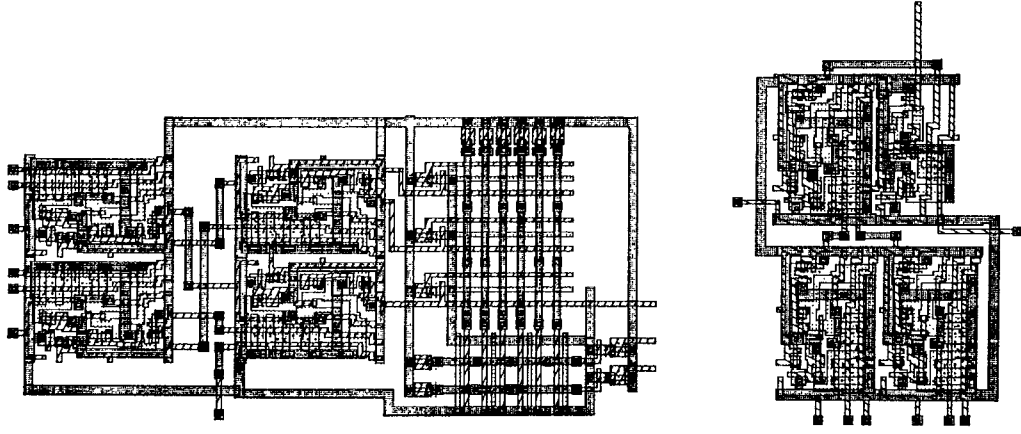
$$f_p(y_1, \dots, y_m) = \sum_{i \in I} \delta_i(y_1, \dots, y_m)$$

where δ_i equals to 1 only when the binary expansion of i is equivalent to $(y_1, \dots, y_m)$.

It is very easy to use PLA to implement f_p once I is known. Since $I \subseteq \{0, 1, \dots, n\}$, the PLA implementation requires at most $n+1$ rows on the AND/OR planes. Hence, the corresponding area is $O(n \log n)$.

Theorem 4.2 *Given m n -variable symmetric functions, the above scheme could lay them out in $O(n \log^2 n + mn)$ area with $O(\log n)$ time delay.*

Based on this method, a program SYMMETRIC has been written which first checks whether the given set of Boolean function are symmetric



(a) $n = 7$

(b) 8 bits parity tree

Figure 5: Layouts of totally symmetric functions

and in the affirmative produces layout for the subset that are symmetric. Figure(5) shows two layouts generated by this program.

4.2 Layout for partially symmetric functions

Extending the realization scheme in figure(1), we determine the floor plan of the layout to be the one shown in figure(6) where NAND and PLA parts are as described previously, and ES_BLOCK is composed of all the elementary symmetric blocks, $ES(X_i)$, which can be laid out by the method described in the previous subsection. Now, we show the following theorem.

Theorem 4.3 *Let f be partially symmetric with respect to the symmetric partition $\{X_1, X_2, \dots, X_s\}$ such that $|X_i| = n_i$, $1 \leq i \leq s$, and let $c_s(f)$ be the number of terms in a minimum symmetric realization of f . Suppose $|n_i| \geq 2$ for $1 \leq i \leq t$, $t \leq s$ and $|n_j| = 1$ for $t + 1 \leq j \leq s$. Then we can*

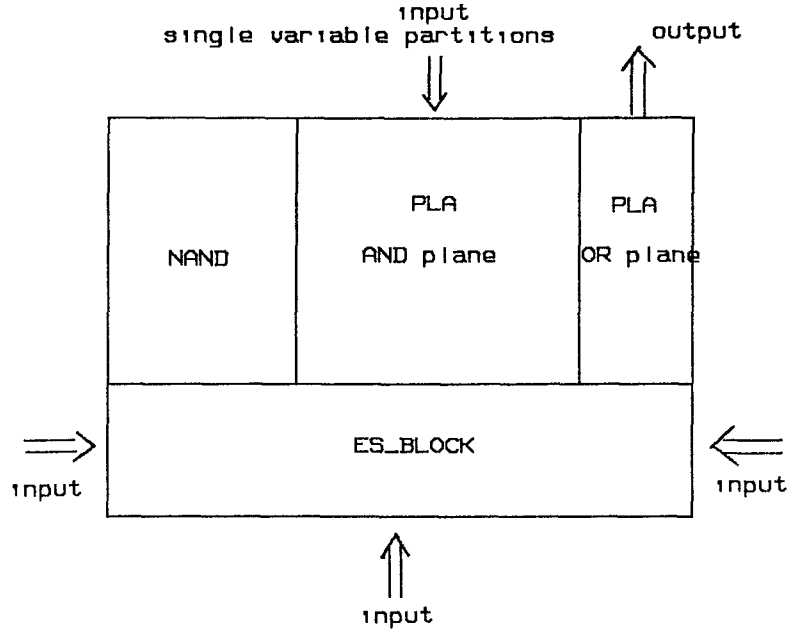


Figure 6: Floor plan to lay out Partially Symmetric Functions

layout f in an area of order

$$O\left(\sum_{i=1}^t n_i \log^2 n_i + nc_s(f)\right)$$

Proof: We have already seen how to layout $\omega(X_i)$ in an area of $O(n_i \log^2 n_i)$. It is easy to see that we can get all the elementary symmetric functions on X_i in essentially the same area. Hence let's assume that we have cells $ES(X_i)$ to compute the required elementary symmetric functions of X_i for all i . From the floor plan in figure(6), we can lay out the partially symmetric functions in the stated area. $\diamond$

Comparing this area with that required by an optimal PLA implementation, we note that if $c(f)$ denotes the size of minimum cover, then the

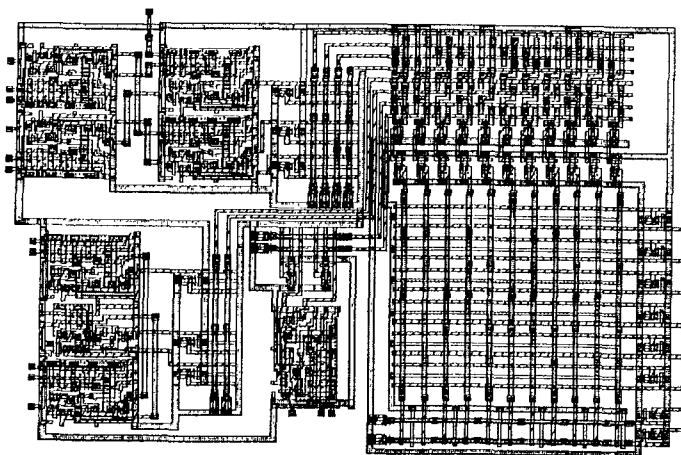


Figure 7: Layout of partially symmetric functions by SYMBL

area of an optimal PLA is $O(nc(f))$. Since the area $nc_s(f)$ in the minimum symmetric realization tends to be almost always the dominant term, using corollary 3.1, we can see that if the function has a good degree of symmetry, substantial reduction in area will result by using the above method. Shown in figure(7) is the layout of two 21-variable functions generate by the system SYMBL [15] based on this method.

References

- [1] R. Brayton, G.D. Hachtel, C.T. McMullen & A.L. Sangiovanni-Vincentelli, *Logic minimization algorithm for VLSI synthesis*, Kluwer Academic Publishers, Reading, 1984.
- [2] L. Trevillyan, W. Joyner & L. Berman, "Global flow analysis in automatic logic design," *IEEE T. on Computer* C-35 (Jan. 1986), 77–81.
- [3] C. Rowen & J.L. Hennessy, "SWAMI: a flexible logic implementation system," *Proc. 22nd Design Automation Conference* (1985).

- [4] B. Serlet, "ALPS: A generator of static CMOS layout from Boolean functions," *Proc. 4-th MIT Conference* (1985).
- [5] S.J. Hong, R.G. Cain & D.L. Ostapko, "MINI: a heuristic approach for logic minimization," *IBM J. of Res. and Dev.* 18 (September 1974), 443–458.
- [6] R. Brayton, G.D. Hachtel, L. Hemachandra, A.R. Newton & A.L. Sangiovanni-Vincentelli, "A comparison of logic minimization strategies using ESPRESSO, an PLA program package for partitioned logic minimization," *Proc. Int. Symp. on Cir. and Syst.* (May 1982).
- [7] E. Morreale, "Recursive operators for prime Implicant and irredundant normal form determination," *IEEE T. Computer* C-19, No.6 (June 1970), 504–509.
- [8] R.E. Bryant, "Graph-based algorithms for Boolean function manipulation," *T. on Computers* Vol C-35, No. 8 (August 1986), 677–691.
- [9] Das & Sheng, "On detecting total or partial symmetry of switching functions," *IEEE T. Computer* C-20 (1971), 352–355.
- [10] Yao & Tang, "On identification of redundancy and symmetry of switching functions," *IEEE T. Computer* C-20 (1971b), 1609–1613.
- [11] M. Davio, J. Deschamps & M. Thayse, *Discrete and switching functions*, McGraw-Hill, Reading, 1978.

- [12] C. Leiserson, "Area efficient layouts," *21st Symp. on Fund. of Comp. Science IEEE* (1980).
- [13] C. Mead & L. Conway, *Introduction to VLSI systems*, Addison-Wiley, 1980.
- [14] S. Muroga, *Logic design and switching theory*, Wiley-Interscience, Reading, 1979.
- [15] J. Ja'Ja' & S.M. Wu, "SYMBL: An optimized layout system for partially symmetric function," *Int'l Conference on Computer-Aided Design* (1986).