

A Fast Method to Derive Minimum SOPs for Decomposable Functions

Tsutomu Sasao

Center for Microelectronic Systems
and Department of Computer Science and Electronics
Kyushu Institute of Technology
Iizuka, Fukuoka, 820-8502 JAPAN
e-mail: sasao@cse.kyutech.ac.jp

Jon T. Butler

Department of Electrical
and Computer Eng.
Naval Postgraduate School
Monterey, CA 93943-5121 U.S.A.
e-mail: butler@cs.nps.navy.mil

Abstract - This paper shows that divide-and-conquer derives a minimum sum-of-products expression (MSOP) of functions that have an AND bi-decomposition when at least one of the sub-functions is orthodox. This extends a previous result showing that divide-and-conquer derives the MSOP of the AND bi-decomposition of two orthodox functions. We show that divide-and-conquer does *not* always produce an MSOP when neither function is orthodox. However, our experimental results show that, in this case, it derives a near minimal SOP. At the same time, our approach significantly reduces the time needed to find an MSOP or near minimal SOP. Also, we extend our results to functions that have a tri-decomposition.

I. INTRODUCTION

Let function f have the decomposition $f(X_1, X_2, \dots, X_m) = g(h_1(X_1), h_2(X_2), \dots, h_m(X_m))$, where X_i and X_j ($i \neq j$) are disjoint. A sum-of-products expression (SOP) for f can be derived by substituting a minimum sum-of-products expression (MSOP) for h_i ($i = 1, 2, \dots, m$) and its complement, as appropriate, into an MSOP for g , and then applying the distributive law. This process is denoted as **divide-and-conquer**. In this paper, we seek conditions under which the resulting SOP is an MSOP. Fig. 1.1 shows the circuit associated with divide-and-conquer.

A specific type of decomposition holds special interest. Function f has an *AND (OR) bi-decomposition* if f can be written as $f(X_1, X_2) = h_1(X_1) \diamond h_2(X_2)$, where X_1 and X_2 are disjoint sets of variables and \diamond is the AND (OR) operation. Many benchmark functions have either an AND or OR bi-decomposition [12].

Functions with an OR bi-decomposition are especially interesting, since a minimum sum-of-products expression (MSOP) for $f(X_1, X_2) = h_1(X_1) \vee h_2(X_2)$ is obtained as $H_1(X_1) \vee H_2(X_2)$, where $H_1(X_1)$ and $H_2(X_2)$ are the MSOP's for $h_1(X_1)$ and $h_2(X_2)$, respectively. In this case, divide-and-conquer always produces an MSOP. As a result, the computation time is short. That is, if $|X_1|$ is nearly the same as $|X_2|$, the time to find $H_1(X_1)$ and $H_2(X_2)$ separately is typically much less than the time to find the MSOP of $f(X_1, X_2) = h_1(X_1) \vee h_2(X_2)$. This follows from the fact that the time to optimize an SOP of an n -variable function whose MSOP has t products is at least $O(nt^2)$.

If f has an AND bi-decomposition, $f = h_1(X_1)h_2(X_2)$, it is tempting to believe that an MSOP for f is obtained by ANDing the MSOPs for $h_1(X_1)$ and $h_2(X_2)$ followed by the application of the distributive law. However, this is only known to hold for

the case where $h_1(X_1)$ and $h_2(X_2)$ are both *orthodox* functions. Namely, an *orthodox* function f has the property that

$$\tau(f) = \eta(f),$$

where $\tau(f)$ is the number of product terms in an MSOP of f and $\eta(f)$ is the number of true minterms of f in the maximum independent set of f ; i.e. a largest set *MIS* of minterms with the property that no prime implicant of f covers two minterms in *MIS*. If $h_1(X_1)$ and $h_2(X_2)$ are both orthodox, then a fast minimization exists for $h_1(X_1)h_2(X_2)$, namely minimize $h_1(X_1)$ and $h_2(X_2)$ separately and apply the distributive law to derive an MSOP for $h_1(X_1)h_2(X_2)$. This property and its relation to fast minimization was first shown in [11]. It has been incorporated into the MUSASHI logic minimizer [6], which successfully derives exact minimum SOPs for functions where ESPRESSO fails due to memory overflow. In this paper, we extend this by showing that an MSOP for a function with an AND bi-decomposition can be obtained in the same way if *exactly one* of $h_1(X_1)$ and $h_2(X_2)$ is orthodox. Further, we show that no definitive statement can be made when both are non-orthodox, although experiments show that, for most cases, the distributive law is *not* sufficient for deriving an MSOP of $h_1(X_1)h_2(X_2)$.

We also extend the theory to derive the MSOPs for tri-decomposable functions. This paper is organized as follows. Section II outlines the notation used. The section after that shows the results cited above. The next section discusses new results on tri-decomposition. Section IV shows experimental results, and Section V gives concluding remarks.

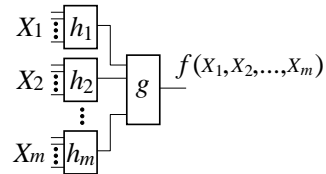


Fig. 1.1. Function decomposition associated with divide-and-conquer.

II. NOTATION

Definition 2.1 x and \bar{x} are literals of variable x . The AND of literals is an implicant I of f , if f is 1 when I is 1.

Definition 2.2 A prime implicant *PI* of a function f is an implicant of f , such that the deletion of any literal in *PI* causes it not to be an implicant of f .

Definition 2.3 An irredundant sum-of-products expression (ISOP) $F(X)$ of function $f(X)$ is the OR of PIs of $f(X)$, such that $f(X) = F(X)$ and no PI can be deleted from $F(X)$ without changing the function represented by $F(X)$.

Definition 2.4 Among the ISOPs for $f(X)$, one with the fewest PIs is a minimum SOP or MSOP.

Example 2.1 $\bar{x}_1x_3 \vee x_2\bar{x}_3 \vee x_1\bar{x}_2$ and $\bar{x}_1x_3 \vee \bar{x}_1x_2 \vee x_1\bar{x}_3 \vee x_1\bar{x}_2$ are both ISOP's of the same function, but only the former is an MSOP. (End of Example)

Definition 2.5 Let $\tau(f)$ be the number of PIs in an MSOP for f .

Example 2.2 For $f = \bar{x}_1x_3 \vee x_2\bar{x}_3 \vee x_1\bar{x}_2$, $\tau(f) = 3$. (End of Example)

The examples we use to illustrate the concepts include symmetric functions.

Definition 2.6 S_A^n , a (totally) symmetric function, is 1 if m of its n variables are 1, where $m \in A$ and is 0 otherwise.

Example 2.3 The AND and OR functions on n variables are symmetric and represented by $S_{\{n\}}^n$ and $S_{\{1,2,\dots,n\}}^n$, respectively. Also, $\bar{x}_1x_3 \vee x_2\bar{x}_3 \vee x_1\bar{x}_2 = S_{1,2}^3(x_1, x_2, x_3)$. The majority function $MAJ(x_1, x_2, x_3)$ is given as $x_1x_2 \vee x_1x_3 \vee x_2x_3 = S_{2,3}^3(x_1, x_2, x_3)$. (End of Example)

III. ORTHODOX FUNCTIONS

As discussed in the previous section, minimization of an SOP is easier when a function f has an AND bi-decomposition, and an MSOP is formed by applying the distributive law to the SOPs of the component functions.

A. Independent Sets of Minterms

Definition 3.1 Given a function $f(X)$, let $M(f)$ be the set of true minterms for f . Then, $IND(f) \subseteq M(f)$ is an **independent set of minterms** of f iff no PI of f covers more than one minterm in $IND(f)$.

Definition 3.2 Given a function f , $MIS(f)$ denotes a **maximum independent set** of minterms. $\eta(f)$ is the number of elements in $MIS(f)$.

Example 3.1 Symmetric function $f = S_{\{1,2\}}^3(x_1, x_2, x_3)$ has two maximum independent sets of minterms $\{\bar{x}_1\bar{x}_2x_3, \bar{x}_1x_2\bar{x}_3, x_1\bar{x}_2\bar{x}_3\}$ and $\{x_1x_2\bar{x}_3, x_1\bar{x}_2x_3, \bar{x}_1x_2x_3\}$. Thus, $\eta(f) = 3$. (End of Example)

Definition 3.3 Given a function $f(X)$, let $M(f)$ be the set of true minterms for f . Then, $DIS(f) \subseteq M(f)$ is a **set of distinguished minterms** if exactly one PI of f covers each minterm in $DIS(f)$.

Example 3.2 Symmetric function $S_{\{1,2\}}^3(x_1, x_2, x_3)$ has no distinguished minterms because every minterm is covered by two PIs. However, $S_{\{1,2,3\}}^3(x_1, x_2, x_3) = x_1 \vee x_2 \vee x_3$ has three distinguished minterms, $\bar{x}_1\bar{x}_2x_3, \bar{x}_1x_2\bar{x}_3, x_1\bar{x}_2\bar{x}_3$. (End of Example)

For a function f covered only by essential PIs, $\eta(f) = |DIS(f)|$. In general, $\eta(f) \geq |DIS(f)|$.

B. Relationship Between MSOPs and Independent Sets of Minterms

Lemma 3.1 [11]

$$\tau(f) \geq \eta(f). \quad (1)$$

Lemma 3.2 [11] Let $IND(h_1)$ be an independent set of minterms of $h_1(X_1)$ and $IND(h_2)$ be an independent set of minterms of $h_2(X_2)$. Then, $IND(h_1) \times IND(h_2)$ is an independent set of minterms for $h_1(X_1)h_2(X_2)$, where $IND(h_1) \times IND(h_2)$ is the AND of all minterms in $IND(h_1)$ with all minterms in $IND(h_2)$.

Definition 3.4 A function f is **orthodox** iff

$$\tau(f) = \eta(f).$$

Otherwise, f is **non-orthodox**.

Example 3.3 Symmetric function $f = S_{\{1,2\}}^3(x_1, x_2, x_3) = x_1\bar{x}_2 \vee x_2\bar{x}_3 \vee x_3\bar{x}_1$ is orthodox, since

$$\tau(f) = \eta(f) = 3.$$

(End of Example)

Theorem 3.1 [11] Let $h_1(X_1)$ and $h_2(X_2)$ be orthodox, where X_1 and X_2 are disjoint sets of variables. Then, $h_1(X_1)h_2(X_2)$ is orthodox.

Next, we derive a new theorem that extends this. Specifically, we show that a minimal SOP is achievable using only the distributive law even if we replace the requirement that both functions be orthodox with the less restrictive requirement that at least one be orthodox. This suggests that the fast algorithm is extended to a much larger class of functions, since experimental results show that orthodox functions represent a vanishingly small fraction of all functions as n increases [11].

Theorem 3.2 Let $H_1(X_1)$ be the MSOP of a nontrivial orthodox function $h_1(X_1)$, and let $H_2(X_2)$ be the MSOP of a nontrivial non-orthodox function $h_2(X_2)$, where X_1 and X_2 are disjoint sets of variables. Then, an MSOP for $h_1(X_1)h_2(X_2)$ is obtained by applying the distributive law to $H_1(X_1)H_2(X_2)$.

Proof On the contrary, suppose that there exists an MSOP $MFG(X_1, X_2)$ for h_1h_2 , where $\tau(h_1h_2) < \tau(h_1)\tau(h_2)$. Let $MIS(h_1)$ be a maximum independent set of h_1 . Let \bar{a}_i be an assignment that corresponds to a minterm in $MIS(h_1)$. Then, $MFG(\bar{a}_i, X_2)$ represents the function $h_2(X_2)$. This is because $h_1(\bar{a}_i) = 1$ and the $MFG(X_1, X_2)$ represents $h_1(X_1)h_2(X_2)$. Let the number of products in $MFG(\bar{a}_i, X_2)$ be t_i .

There are $\eta(h_1)$ different ways to assign \bar{a}_i . Note that for different \bar{a}_i , different products in $MFG(X_1, X_2)$ remain, since $h_1(X_1)$ is non-trivial (not a constant 1). Further, $\sum_{i=1}^{\eta(h_1)} t_i = \tau(MFG(X_1, X_2))$. Note that, $\tau(MFG(X_1, X_2)) < \tau(h_1)\tau(h_2)$. Also, $\tau(h_1) = \eta(h_1)$, since h_1 is orthodox.

By the Pigeonhole Principle [4], there exists at least one assignment \bar{a}_i such that $t_i < \tau(h_2(X_2))$. But, this is a contradiction, since such an assignment produces an SOP for $h_2(X_2)$ with fewer products than $\tau(h_2(X_2))$. ■

Corollary 3.1 Let exactly one of $h_1(X_1)$ or $h_2(X_2)$ be orthodox, where X_1 and X_2 are disjoint sets of variables. Then,

$$\tau(h_1(X_1)h_2(X_2)) = \tau(h_1(X_1))\tau(h_2(X_2)).$$

We can extend our result to bi-decompositions besides AND and OR.

Theorem 3.3 Suppose that the given function f has a bi-decomposition $f(X_1, X_2) = g(h_1(X_1), h_2(X_2))$, where X_1 and X_2 are disjoint sets of variables. If h_1 , h_2 , and their complements are orthodox, then an MSOP for f can be derived from the MSOPs for g , and h_1 , h_2 and their complements using the distributive law only.

This shows that divide-and-conquer succeeds for all two-variable functions g when the functions h_i and their complements are orthodox.

C. Classes of Functions That Are Orthodox

The following summarizes functions that are orthodox [11].

TABLE 3.1
SUMMARY OF FUNCTIONS THAT ARE ORTHODOX.

Function
Unate (includes threshold)
Symmetric
All functions on less than four variables

D. Further Results

Definition 3.5 Let \vec{a} and \vec{b} be assignments of values to variables X . Function $f(X)$ is positive iff $f(\vec{a}) \geq f(\vec{b})$, whenever $\vec{a} \geq \vec{b}$.

Theorem 3.4 Let f be represented as $f(X_1, X_2, \dots, X_m) = g(h_1(X_1), h_2(X_2), \dots, h_m(X_m))$, where h_i are orthodox, $g(y_1, y_2, \dots, y_m)$ is positive, and X_1, X_2, \dots , and X_m are pairwise disjoint. Then, $g(h_1(X_1), h_2(X_2), \dots, h_m(X_m))$ is orthodox.

Proof Let $P = y_{i_1}y_{i_2}\dots y_{i_k}$ be a prime implicant of g . Then there exist $\eta(h_{i_1})\eta(h_{i_2})\dots\eta(h_{i_k})$ independent minterms for f . A method to find such minterms is shown in Example 3.5. Let t be the total number of such minterms for all the PIs for g . Note that they form an independent set of minterms of f . Thus, an MSOP for f requires at least t PIs.

On the other hand, f can be represented by the SOP with t PIs. This is because substituting MSOPs for h_i into the MSOP for g , and then applying the distributive law produces an SOP with t products, since $\tau(h_i) = \eta(h_i)$. This implies that $\tau(f) = \eta(f)$. ■

Corollary 3.2 Let f be represented as $f(X_1, X_2, \dots, X_m) = g(h_1(X_1), h_2(X_2), \dots, h_m(X_m))$, where h_i are orthodox, g is positive, and X_1, X_2, \dots , and X_m are pairwise disjoint. Then, an MSOP for f can be obtained by substituting the MSOPs for $h_i(X_i)$ into $G(y_1, y_2, \dots, y_m)$, and applying the distributive law.

Example 3.4 Consider the case where $g(y_1, y_2, y_3) = y_1y_2 \vee y_2y_3 \vee y_3y_1$. Here, $f(X_1, X_2, X_3) = g(h_1(X_1), h_2(X_2), h_3(X_3))$ contains $\eta(h_1)\eta(h_2) + \eta(h_2)\eta(h_3) + \eta(h_3)\eta(h_1)$ independent minterms. Also, it is clear that substituting MSOPs for h_1 , h_2 and h_3 into the MSOP for $g(y_1, y_2, y_3)$ and then applying the distributive law produces an SOP with $\tau(h_1)\tau(h_2) + \tau(h_2)\tau(h_3) + \tau(h_3)\tau(h_1)$ products. Note that $\tau(h_i) = \eta(h_i)$. This means that the MSOP for $g(h_1(X_1), h_2(X_2), h_3(X_3))$ has $\tau(h_1)\tau(h_2) + \tau(h_2)\tau(h_3) + \tau(h_3)\tau(h_1)$ products. (End of Example)

Example 3.5 Let $g(y_1, y_2, y_3) = y_1y_2 \vee y_2y_3 \vee y_3y_1$, and let $h_1(x_1, x_2, x_3) = h_2(x_1, x_2, x_3) = h_3(x_1, x_2, x_3) = h(x_1, x_2, x_3) = x_1\bar{x}_2 \vee x_2\bar{x}_3 \vee x_3\bar{x}_1$. In this case, $\tau(h) = \eta(h) = 3$. An MIS for $h_1(x_1, x_2, x_3)$ is $\{\bar{x}_1x_2x_3, x_1\bar{x}_2x_3, x_1x_2\bar{x}_3\}$. For simplicity, we denote this by $\{011, 101, 110\}$. Consider the function $f(X_1, X_2, X_3) = g(h_1(X_1), h_2(X_2), h_3(X_3))$. Then, a set of independent minterms is

111-011-011	111-011-101	111-011-110
111-101-011	111-101-101	111-101-110
111-110-011	111-110-101	111-110-110
011-111-011	011-111-101	011-111-110
101-111-011	101-111-101	101-111-110
110-111-011	110-111-101	110-111-110
011-011-111	011-101-111	011-110-111
101-011-111	101-101-111	101-110-111
110-011-111	110-101-111	110-110-111

There are 27 minterms in the set. Note that they are mutually independent, as follows.

A set of all the prime implicants (PIs) of the positive function $g(y_1, y_2, y_3)$ forms an MSOP. Note that the literals of a PI of g are uncomplemented. If a variable y_i appears as an uncomplemented literal in a PI, assign a minterm of the MIS of $h_i(X_i)$. If a variable y_i is missing in a PI, assign any minterm in the complement function of $h_i(X_i)$.

First consider a PI of g : y_2y_3 . For x_1, x_2 and x_3 , we assign a minterm 111, since y_1 is missing in y_2y_3 . For x_4, x_5 and x_6 , we assign a minterm 011 or 101 or 110, since y_2 appears as a positive literal in y_2y_3 . For x_7, x_8 and x_9 , we assign a minterm 011 or 101 or 110, since y_3 appears as a positive literal in P . Thus, for a PI y_2y_3 , we have 9 different assignments. They are minterms of f . Also, they are independent each other. For example, 111-011-011 and 111-011-101 are independent. On the contrary, suppose that there is a PI that covers both minterms, then it covers 111-011-**1, where * denotes a don't care, while - is a delimiter. This implies 111-011-111 is a minterm of f , which contradicts the definition of f , since $g(0,1,0)=0$.

Next, consider two PIs of g : y_2y_3 and y_1y_2 . Also consider two independent sets of minterms for them. Select one minterm from y_2y_3 and another minterm from y_1y_2 . Suppose that they are 111-011-011 and 011-111-011. We claim that these two minterms are mutually independent. On the contrary, suppose

TABLE 3.2
POSSIBLE VARIABLES IN A PRODUCT TERM OF THE MAJ FUNCTION.

X_1	X_2	X_3	Comment
			Impossible
		✓	Impossible
	✓		Impossible
	✓	✓	
✓			Impossible
✓		✓	
✓	✓		
✓	✓	✓	Impossible

that there is a PI that covers both minterms, then it covers *11-11-011. This implies 111-111-011 is a minterm of f , which contradicts the definition of f , since $g(0, 0, 1) = 0$.

In this way, we can show that these 27 minterms are mutually independent. (End of Example)

We can extend Theorem 3.4 by relaxing the requirement that all subfunctions must be orthodox in the case of a specific positive function, $MAJ(y_1, y_2, y_3)$.

Lemma 3.3 Let $H_1(X_1)$, $H_2(X_2)$, and $H_3(X_3)$ be the MSOP of functions $h_1(X_1)$, $h_2(X_2)$, and $h_3(X_3)$, respectively, where X_1 , X_2 , and X_3 are pairwise disjoint sets of variables. Assume at least two are orthodox. Then, an MSOP for $MAJ(h_1(X_1), h_2(X_2), h_3(X_3))$ is derived from $H_1(X_1)H_2(X_2) \vee H_1(X_1)H_3(X_3) \vee H_2(X_2)H_3(X_3)$, using the distributive law. Further, $\tau(MAJ(h_1(X_1), h_2(X_2), h_3(X_3))) = \tau(h_1(X_1))\tau(h_2(X_2)) + \tau(h_1(X_1))\tau(h_3(X_3)) + \tau(h_2(X_2))\tau(h_3(X_3))$.

Proof Table 3.2 shows the situation with respect to variables in a prime implicant of M of the MSOP of $MAJ(h_1(X_1), h_2(X_2), h_3(X_3))$. Since the AND of two functions, at least one of which is orthodox, on disjoint sets of variables are orthodox, we can conclude that MSOPs for $H_1(X_1)H_2(X_2)$, $H_1(X_1)H_3(X_3)$, and $H_2(X_2)H_3(X_3)$ can be obtained by the distributive law. It remains to show that an MSOP for $MAJ(h_1(X_1), h_2(X_2), h_3(X_3))$ is derived as $H_1(X_1)H_2(X_2) \vee H_1(X_1)H_3(X_3) \vee H_2(X_2)H_3(X_3)$. On the contrary, assume it is not, and there is another ISOP, M , that is an MSOP.

Consider a prime implicant in M . A check in Table 3.2 indicates that at least one variable exists in M from the group designated by the heading. The first row, containing no check, corresponds to a product term with no variables. Thus, this is an "Impossible" situation, as indicated by the comment in the fourth column. The case in which only variables from X_3 occur, the second row, is also Impossible. This is so, because, for this product term, variables in X_1 and X_2 can be chosen so that $h_1(X_1)$ and $h_2(X_2)$ are 0. Thus, no matter what variables are included from X_3 , the function is 0 for some assignments of values such that the product term is 1. It follows that the product term is not an implicant of the function. Similarly, it is impossible to have variables just from X_1 and just from X_2 .

The last row of Table 3.2 corresponds to a product term that has at least one variable from each set X_1 , X_2 , and X_3 . For this product term, it must be that at least two of the three functions,

$h_1(X_1)$, $h_2(X_2)$, and $h_3(X_3)$ are 1; otherwise the function is not 1. Suppose, $h_1(X_1)$ and $h_2(X_2)$ are 1. Then, we can eliminate variables from h_3 and the function is still 1. It follows that such variables are redundant, and the product term is not a prime implicant. Thus, this case is impossible.

Of the remaining four possibilities, consider a term that has variables or their complements from X_2 and X_3 , and none from X_1 . It must be that values which make the product term 1, also cause $h_2(X_2)$ and $h_3(X_3)$ to be 1. On the contrary, if either $H_2(X_2)$ or $H_3(X_3)$ are 0, variables from X_1 can be chosen so that the function is 0. Thus, the term is not a product term. It follows that a product term of the function containing variables from X_2 and X_3 only is also a product term of $H_2(X_2)H_3(X_3)$. A similar argument applies when the product term has variables only from X_1 and X_2 or from X_1 and X_3 . This proves that an implicant of the function must be an implicant of exactly one of $H_1(X_1)H_2(X_2)$, $H_1(X_1)H_3(X_3)$, and $H_2(X_2)H_3(X_3)$. It follows that if the SOP derived from $H_1(X_1)H_2(X_2) \vee H_1(X_1)H_3(X_3) \vee H_2(X_2)H_3(X_3)$ is not an MSOP, then at least one of $H_1(X_1)H_2(X_2)$, $H_1(X_1)H_3(X_3)$, and $H_2(X_2)H_3(X_3)$ is not an MSOP, a contradiction. $\tau(MAJ(h_1(X_1), h_2(X_2), h_3(X_3))) = \tau(h_1(X_1))\tau(h_2(X_2)) + \tau(h_1(X_1))\tau(h_3(X_3)) + \tau(h_2(X_2))\tau(h_3(X_3))$ follows directly from this. ■

Note that we don't actually require $h_1(X_1)$, $h_2(X_2)$, and $h_3(X_3)$ to be orthodox. Rather, we require only that the SOPs derived from $H_1(X_1)H_2(X_2)$, $H_1(X_1)H_3(X_3)$, and $H_2(X_2)H_3(X_3)$ be MSOPs. Thus, Lemma 3.3 applies to a larger class of functions.

Definition 3.6 f is a positive cascade function if f can be represented as

$$f(x_1, x_2, \dots, x_n) = x_1 \diamond_1 (x_2 \diamond_2 (\dots (x_{n-1} \diamond_{n-1} x_n) \dots)), \quad (2)$$

where \diamond_i is either the OR (\vee) or AND (\wedge) function.

We can conclude

Lemma 3.4 The MSOP of a positive cascade function can be obtained by recursive substitution into (2) and the distributive law only.

Definition 3.7 A function $f(X)$ is a **positive fanout-free function** iff it can be realized by a circuit where each gate realizes either AND or OR and where each gate output drives at most one gate input or the circuit output.

Note that $MAJ(y_1, y_2, y_3)$ is not a positive fanout-free function.

Lemma 3.5 The MSOP of a positive fanout-free function can be obtained by recursive substitution and the distributive law only in the expression for the fanout-free circuit.

Definition 3.8 A function $f(X)$ is **unate** iff it is a constant or is representable as an SOP in which each variable appears only complemented or only uncomplemented.

Lemma 3.5 can be extended to unate fanout-free functions. A unate function is formed from a positive function by complementing zero, one, or more variables and/or the function itself. For example, \bar{x}_1 , $x_1 \vee \bar{x}_2$, x_1 , and $x_1 \vee x_2$ are unate functions. The last two functions are positive. From Lemma 2 of [2], any

unate fanout-free circuit realizes a function that is also realized by a positive fanout-free circuit with zero, one, or more variables complemented. Thus, the MSOP of a unate fanout-free function $f(X)$ can be derived from the underlying positive fanout-free function $f(X)$, as specified in Lemma 3.5. Thus, we can state

Lemma 3.6 *The MSOP of a unate fanout-free function can be obtained by recursive substitution and the distributive law only in the expression for the fanout-free circuit, where inverters can appear only on the input.*

Lemma 3.7 *Let $H_1(X_1)$ and $H_2(X_2)$ be the MSOPs of non-constant functions $h_1(X_1)$ and $h_2(X_2)$, respectively, where X_1 and X_2 are disjoint sets of variables. Let $m(y, X_1, X_2)$ be a MUX function such that $m = h_1(X_1)\bar{y} \vee h_2(X_2)y$. Then, an MSOP for h is obtained by substituting $H_1(X_1)$ into f and $H_2(X_2)$ into g and applying the distributive law.*

Proof A prime implicant P of $m = h_1(X_1)\bar{y} \vee h_2(X_2)y$ is a prime implicant of $h_1(X_1)\bar{y}$ or $h_2(X_2)y$ and must contain the literal \bar{y} or y , respectively. Removing the literal y yields a prime implicant P' of $h_1(X_1)$ or $h_2(X_2)$. From this, it follows that the MSOP has at least $\tau(h_1(X_1)) + \tau(h_2(X_2))$ PIs. ■

IV. TRI-DECOMPOSITION

Definition 4.1 *f is a tri-decomposable iff f is represented as*

$$f(X_1, X_2, X_3) = g(h_1(X_1), h_2(X_2), h_3(X_3)). \quad (3)$$

A natural extension to bi-decomposition, tri-decomposition has special merit because certain choices for g occur often in logic design, for example, the majority function and the multiplexer.

Theorem 4.1 *Let f be tri-decomposed as*

$$f(X_1, X_2, X_3) = g(h_1(X_1), h_2(X_2), h_3(X_3)). \quad (4)$$

Let h_i and \bar{h}_i ($i = 1, 2, 3$) be orthodox.

- 1) If $g(y_1, y_2, y_3) = \bar{y}_1\bar{y}_2\bar{y}_3 \vee y_1y_2y_3$ then $\tau(f) = \tau(\bar{h}_1)\tau(\bar{h}_2)\tau(\bar{h}_3) + \tau(h_1)\tau(h_2)\tau(h_3)$. Also, f is orthodox.
- 2) If $g(y_1, y_2, y_3) = \bar{y}_1y_2 \vee y_1y_3$ then $\tau(f) = \tau(\bar{h}_1)\tau(h_2) + \tau(h_1)\tau(h_3)$. Also, f is orthodox.
- 3) If $g(y_1, y_2, y_3) = \bar{y}_1y_2y_3 \vee y_1\bar{y}_2y_3 \vee y_1y_2\bar{y}_3$ then $\tau(f) = \tau(\bar{h}_1)\tau(h_2)\tau(h_3) + \tau(h_1)\tau(\bar{h}_2)\tau(h_3) + \tau(h_1)\tau(h_2)\tau(\bar{h}_3)$. Also, f is orthodox.
- 4) If $g(y_1, y_2, y_3) = y_1\bar{y}_2\bar{y}_3 \vee y_2y_3$ then $\tau(f) = \tau(h_1)\tau(\bar{h}_2)\tau(\bar{h}_3) + \tau(h_2)\tau(h_3)$. Also, f is orthodox.

In all cases, MSOPs for f are derived by substituting MSOPs for $h_i(X_i)$ and $\bar{h}_i(X_i)$ into y_i and \bar{y}_i and applying the distributive law.

Suppose that the given function f has a tri-decomposition $f(X_1, X_2, X_3) = g(h_1(X_1), h_2(X_2), h_3(X_3))$, where X_1, X_2 and X_3 are mutually disjoint. Also assume that h_1, h_2, h_3 and their complements are all orthodox.

In this case, Theorem 4.1 shows that for many functions g , an MSOP for f can be derived from the MSOPs for g , and h_1, h_2, h_3 and their complements using the distributive law only.

There are $2^8 = 256$ different functions of three variables. Among them, 38 functions are degenerate, that is, depend on two or fewer variables. Also 9 are positive functions, and 64 are unate fanout-free as well as unate cascade functions. The number of three-variable functions that are neither, positive, fanout-free, nor P-equivalent to the functions g in Theorem 4.1 is only 76. Thus, in many functions, divide-and-conquer produces an MSOP.

V. EXPERIMENTAL RESULTS

An experiment was conducted to determine the extent to which divide-and-conquer minimizes functions with an AND bi-decomposition. Namely, we have shown that it is an attractive method when at least one function is orthodox. On the other hand, we know that the distributive law fails to produce an MSOP for specific non-orthodox functions [11]. The extent of this failure has been an open question up to this point.

To investigate this, we formed a set S of 67 randomly generated 6-variable *nonorthodox* functions. For each, we found the MSOP, using the Quine-McCluskey (QM) method. Then, we produced 2278 12-variable functions of the form $f(X_1, X_2) = f_1(X_1)f_2(X_2)$, where $f_1(X_1)$ and $f_2(X_2)$ are chosen from S . Specifically, each function $f(X_1, X_2)$ occurs as one of the $\binom{67}{2} + 67 = 2278$ ways to choose an unordered pair from S . Then, we found the MSOP of each $f(X_1, X_2)$, using the QM algorithm and compared it with the sum-of-products expression obtained by applying the law of distributivity to the product of the MSOPs for $f(X_1)$ and $f(X_2)$. For each pair, we compute $\delta(f)$, which is the absolute error using the divide-and-conquer method, i.e., the difference between the number of products in an MSOP and the SOP derived by divide-and-conquer. For example, a value $\delta(f(X_1, X_2)) = 0$, means that the divide-and-conquer method yields an MSOP, while $\delta(f(X_1, X_2)) = 1$ means the divide-and-conquer method yields a sum-of-products expression with one more product term than the MSOP.

Among the 2278 pairs of functions, we found that $0 \leq \delta \leq 4$. The number of pairs where $\delta = 4, 3, 2, 1$, and 0 is 2, 9, 198, 1053, and 994, respectively. For 22 of the functions, excessive computation time needed for the derivation of the MSOP for the composite function $f(X_1, X_2) = f_1(X_1)f_2(X_2)$ precluded the computation of δ .

Fig. 5.1 shows the value of δ for all pairs of functions in S . The two horizontal axes each represent indices on the 67 random functions, while the vertical axis shows the value of δ when the function indexed by one horizontal axis is ANDed with the function indexed by the other horizontal axis. The diagonal extending from (1,1) to (67,67) represents functions in S ANDed with themselves. Because AND is commutative, a function above this diagonal is identical to one below it, and the resulting δ values are symmetric about this diagonal. Some values of δ are missing because of prohibitive computation

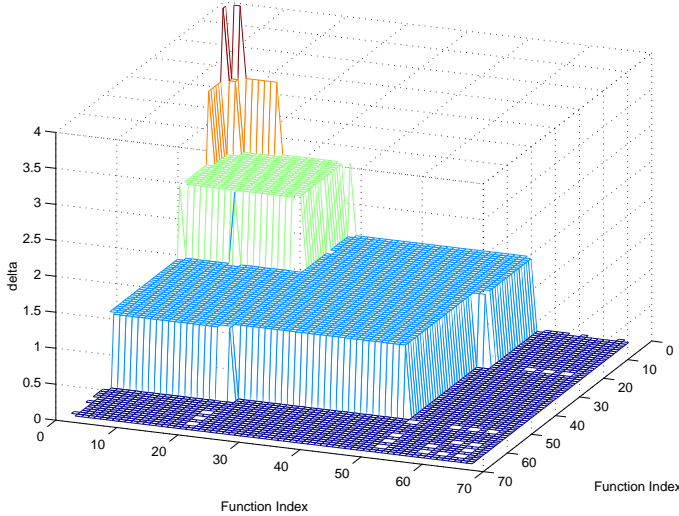


Fig. 5.1. Distribution of excess product terms over MSOP when divide-and-conquer is applied to non-orthodox functions.

time, as mentioned above.

The indices of the functions were chosen so that functions that tended to produce smaller δ had a larger index. With this choice, it is clear that there appear to be functions that tend to produce $\delta = 0$ when combined in a bi-decomposition with any other function. This suggests the existence of functions in addition to orthodox functions that yield an MSOP when combined in a bi-decomposition with any other function. Within the 67 randomly chosen functions, 16 produce $\delta = 0$ in a bi-decomposition with all 67 functions.

VI. CONCLUDING REMARKS

In this paper, we have shown that divide-and-conquer is a good heuristic. This follows directly from $\delta(f) = 0$ when at most one of the two functions is orthodox, and from the small δ when neither is orthodox.

Our technique can significantly reduce the time needed to find an MSOP. The concepts discussed have been used to improve MUSASHI logic minimizer [6], which can successfully derive exact minimum SOPs for functions on which ESPRESSO fails due to memory overflow. For example, ESPRESSO [1] attempts to derive the MSOP of $f = x_1x_2x_3 \vee x_4x_5x_6 \vee \dots \vee x_{97}x_{98}x_{99}$ by first finding the complement of f . It consists of 3^{33} implicants and is impossible to derive; i.e., ESPRESSO fails. On the other hand, MUSASHI just returns the input SOP as an MSOP. It is a sum of disjoint support functions. So, each component function can be minimized independently. Also, consider the function $g = (x_1 \vee x_2)(x_3 \vee x_4)(x_5 \vee x_6) \dots (x_{39} \vee x_{40})$. ESPRESSO fails since the number of products is 2^{20} and requires too much computation time, while MUSASHI generates its MSOP easily.

There is an interesting open question. Namely, what char-

acterizes those functions for which divide-and-conquer succeeds in finding the MSOP of functions with an AND bi-decomposition. From prior experiments [11], it appears that orthodox functions form a vanishingly small subset of all n -variable functions as $n \rightarrow \infty$. Thus, if the characteristic is tied to the orthodox property, this suggests that a vanishingly small number of AND bi-decomposable functions can use divide-and-conquer to find an MSOP. However, from the experiments presented here, divide-and-conquer sometimes succeeds when both functions are non-orthodox.

ACKNOWLEDGEMENTS

This research is partly supported by a Grant in Aid for Scientific Research from the Japan Society for the Promotion of Science (JSPS), and funds from MEXT via the Kitakyushu innovative cluster project. Mr. M. Matsuura did experiments.

REFERENCES

- [1] R. K. Brayton, G. D. Hachtel, C. T. McMullen, and A. Sangiovanni-Vincentelli, *Logic Minimization Algorithms for VLSI Synthesis*, Norwall, MA, Kluwer Academic Publishers, April 1984.
- [2] J. T. Butler, "Analysis and design of fanout-free networks of positive symmetric gates," *Jour. of the ACM*, pp. 481-498, July 1978.
- [3] J. P. Hayes, "The fanout structure of switching functions," *Jour. of the ACM*, pp. 551-571, October 1975.
- [4] C. L. Liu, *Elements of discrete mathematics*, McGraw-Hill, 1977.
- [5] R. S. Michalski and Z. Kulpa, "A system of programs for the synthesis of switching circuits using the method of disjoint stars," *Proceedings of IFIP Congress*, pp. 61-65, April 1971.
- [6] A. Mischenko and T. Sasao, "Large-scale SOP minimization using decomposition and functional properties," *40th Design Automation Conference*, Anaheim, CA, pp. 471-477, June 2-6, 2003.
- [7] A. Odlyzko, "On covering a product of sets with products of subsets," *Discrete Mathematics*, pp. 373-380, 1973.
- [8] W. J. Paul, "Realizing Boolean functions on disjoint set of variables," *Theoretical Computer Science 2*, pp. 383-396, 1976.
- [9] R. L. Rudell and A. Sangiovanni-Vincentelli, "Multiple-valued minimization for PLA optimization," *IEEE Trans. Computer-Aided Design*, Vol. CAD-6, No. 5 pp. 727-749, September 1987.
- [10] T. Sasao, *Switching Theory for Logic Synthesis*, Kluwer Academic Pub., p. 95, Boston, London and Dordrecht, 1999.
- [11] T. Sasao and J. T. Butler, "On the minimization of SOPs for bi-decomposable functions," *Asia and South Pacific Design Automation Conference, (ASP-DAC'2001)*, Yokohama, Japan, pp. 219-224, Jan. 30-Feb. 2, 2000.
- [12] T. Sasao and M. Matsuura, "DECOMPOS: An integrated system for functional decomposition," *1998 International Workshop on Logic Synthesis*, Lake Tahoe, pp. 471-477, June 1998.
- [13] B. Voight and I. Wegener, "A remark on minimal polynomials of Boolean functions," *CSL'88, 2nd Workshop on Computer Science Logic Proceedings*, pp. 372-383, 1989.