Derivation of the Complete Sum of a Switching Function with the Aid of the Variable-entered Karnaugh Map

Ali M. Rushdi[PN1] and Husain A. Al-Yahya

Department of Electrical and Computer Engineering,
King Abdulaziz University, P.O. Box 9027, Jeddah, 21413,
Saudi Arabia.

(Received 19 August 1998; accepted for publication 2 August 2000)

Abstract. The complete sum (*CS*) of a switching function f is defined as a sum-of-products formula whose products constitute all, and nothing but, the prime implicants of f. It has many useful applications including the simplification and minimization of switching functions, proving equivalence or independence, solution of Boolean equations, and transient hazard analysis. This paper presents two novel techniques for deriving the *CS* with the aid of the variable-entered Karnaugh map (*VEKM*); a map that enjoys several pictorial advantages and a doubled variable-handling capability, and hence is recommended when the number of variables ranges from 7 to 12 or even more. Only completely specified switching functions (*CSSFs*) are considered herein, simply because the *CS* of an incompletely specified function f is that of the *CSSF* that represents the upper bound for f. Our first technique uses the *VEKM* for obtaining any product-of-sums expression for the function which can be multiplied out to produce the CS after deletion of any absorbable terms. The second technique starts with a *VEKM* of *CS* entries, and after repeated folding of the *VEKM* ends up with the required *CS* provided necessary absorptions are implemented after each folding. Both techniques gain much from the use of a novel multiplication matrix that restricts the number of term comparisons needed for implementing absorptions. This matrix can be used to advantage also with some purely algebraic techniques such as Tison method. However, algebraic techniques, even after improvement, might remain inferior to *VEKM* techniques, obviously since the latter can combine most merits of map and algebra. Dual versions of the *VEKM* techniques considered can be used to obtain the dual of the complete sum, viz., the complete product.

Introduction

The complete sum of a switching function f, to be denoted by $CS(f)$, is the all-prime-implicant disjunction that expresses f, i.e., it is a sum-of-products (s-o-p) formula whose products are all the prime implicants of f. The complete sum is also known in the literature [1,2] as the Blake *canonical form* of the switching function. Since all the prime implicants of f are present in $CS(f)$, it is obviously unique and hence stands for a canonical representation of the switching function. The complete sum for the incompletely-specified switching function (*ISSF*) $f = g \vee d(h)$ is that of the associated completely-specified switching function $(CSSF)$ $f_1 = g \vee h$. This means that a study of the complete sum always involves a *CSSF* and does not really involve an *ISSF*. Henceforth, when we refer to a switching function *f*, we understand it is a *CSSF*.

The concept of the complete sum of a switching function *f* is closely related to that of a syllogistic formula for *f* [1,2]. However, while *CS* (*f*) is unique and canonical, there are infinitely many syllogistic formulas for *f*. A syllogistic formula of *f* can be defined as an s-o-p formula whose terms include, but are not necessarily excluded to, all the prime implicants of *f*, i.e., it is the complete sum of *f* disjuncted (possibly) with terms each of which subsumes some prime implicant of *f*. Each of the following formulas are syllogistic formulas:

(i) a complete-sum formula,
(b) an alterm (a disjunction of single literals),
(c) an s-o-p formula of monoform literals only,
(d) an s-o-p formula such that no two terms in it have a consensus that does not appear in the formula.

If we compare the definition of a syllogistic formula for *f* to that of its complete sum *CS* (*f*), we note that *CS* (*f*) is minimal within the class of syllogistic formulas for *f* , i.e., the set of terms in any syllogistic formula for *f* is a superset of the set of terms in *CS* (*f*). Hence *CS* (*f*) can be represented by *ABS*(*F*), where *F* is any syllogistic formula for *f* and *ABS* (*F*) denotes an equivalent absorptive formula of *F*, i.e., a formula obtained from *F* by successive deletion of terms absorbed in other terms of *F*.

Obtaining the complete sum is the first important step in deriving the irredundant disjunctive forms and, in particular, a minimal sum of a given switching function, and consequently constructing economical networks for the function. Since the complete sum is a canonical representation, it is useful in proving the equivalence of two switching expressions. Also, it is useful in simplifying switching expressions or detecting if a function independent of some variables [3]. The complete sum is very crucial in transient hazard analysis [4]. Its evaluation is the basic instrumental tool of Boolean reasoning and the solution of Boolean equations [1]. In fact, a systematic method for finding all prime implicants of a switching function can enable logicians to ferret out hidden implicants and therefore, hidden logical conclusions from a given set of premises [2].

In view of our definition of *CS* (*f*) as *ABS*(*F*), it is obvious that *CS* (*f*) may be generated by the following two-step procedure: (a) Find a syllogistic formula *F* for *f* and (b) Delete absorbed terms to obtain *ABS*(*F*). Many techniques exist in the literature for carrying out step (a). These are categorized by Brown [1] into the three basic approaches of exhaustion of implicants, iterative consensus and multiplication. In the following two sections we present novel versions of the multiplication and iterative consensus approaches, which are implemented through the use of the variable-entered Karnaugh map (*VEKM*) [3-15] and are hence referred to as *VEKM* multiplication and *VEKM* folding, respectively. The *VEKM* is an extended version of the conventional Karnaugh map (*CKM*) which retains most of its pictorial merits while improving its capability to handle larger numbers of variables and/or functions whose intrinsic structure is more complex. We also introduce a novel multiplication matrix that has the distinct advantage that it minimizes the number of comparisons needed in the cumbersome step (b) above of deleting absorbed terms. An offshoot outcome of this paper is given in section IV wherein our matrix multiplication is used to improve Tison method, which is a classical purely algebraic method for obtaining the complete sum. This achievement is self rewarding since we rely on Tison method for securing *CS* formulas as *VEKM* entries. Even after improvement, Tison method remains inferior to our *VEKM* techniques, provided the number of variables used is not very small. Final comments on the paper are given in its concluding section (section V) This paper contains some specialized, albeit standard, terminology and concepts of switching theory. A useful reference on these is the excellent and lucid text by Muroga [3]. To make the paper more self-contained and easier to read, we supplement its main text with 2 appendices covering non original material. Appendix A is a brief introductory tutorial on the *VEKM*. Appendix B introduces Tison method and illustrates it with an example that enables the reader to appreciate the improvement proposed herein.

VEKM Multiplication

The present section discusses a *VEKM* implementation of the multiplication method for obtaining the complete sum of a switching function. According to Brown [1], the credit of this method should go to C. S. Peirce, though it is now frequently attributed to Nelson [16]. The essence of this method is given in Theorem 1 to follow. For a proof of this theorem, the interested reader may refer to [1].

Theorem 1: Let a switching function f be expressed as a conjunction of syllogistic formulas and multiply out to obtain a sum-of-products (s-o-p) formula using the distributive laws. This s-o-p formula is a syllogistic formula for f. Then, use idempotency of AND to drop duplicate literals and use absorption to delete any term that subsumes another. Now, the final formula is the complete sum for f.

It is clear from Theorem 1 that the multiplication technique guarantees the generation of all the prime implicants of f, i.e., it is a substitute for consensus generation. However, the technique is still burdened with the two tasks of (a) multiplying out formulas, and (b) implementing absorption. Note that, in particular, the expansion of a conjunctive form into a disjunctive one is usually very-time consuming. Use of certain identities, which might be labeled as intelligent multiplication, is usually helpful [1, p.85]. Of paramount importance is the identity

$$(X \vee Y)(X \vee Z) = X \vee Y Z, \tag{1}$$

which is useful not only in *VEKM* multiplication, but also in *VEKM* folding, as we will see later. Other useful rules are also given in [3, pp. 182-183]. There are faster means for achieving the purpose of multiplication (viz., the derivation of all prime implicants) without really having to expand a conjunctive form into a disjunctive one. One such means advocated by Muroga [3] is that of the semantic tree method.

As a special case of Theorem 1, a syllogistic formula may be produced by multiplying out a conjunction of alterms, i.e., a product-of-sum (p-o-s) formula. In the present section, we make use of this special case wherein a *VEKM* is utilized to obtain the required p-o-s formula. Note that such a formula need not necessarily be minimal, though the simpler it is the better. Therefore, we can implement the *VEKM* dual procedure in [11] without worrying too much about minor details that are needed for exact minimality.

Whenever we multiply only two sums of products at a time, it is convenient to construct a multiplication table or matrix. The horizontal keys of this table are the terms of one sum and its vertical keys are the terms of the other sum, while its entries are the terms resulting from multiplying out the two sums. Figure1 shows typical keys and entries of such a table, where we use the symbol $\{P_iT_j\}$ to denote the product of the two terms P_i and T_j after deleting any repeated literals (thanks to the idempotency of AND). Of course, if the terms P_i and T_j have at least one opposition, i.e., one literal that appears complemented in one of them and uncomplemented in the other, then $\{P_iT_j\}$is 0.

Now, we observe and prove a new interesting and useful property of the proposed multiplication matrix. Suppose that the product $\{P_rT_k\}$ subsumes (and hence is absorbed by) the product $\{P_iT_j\}$ which lies in a different row $(i \neq r)$ and a different column $(j \neq k)$. This means that the set of literals of $\{P_rT_k\}$ is a superset of the set of literals of $\{P_iT_j\}$ and hence it is a superset of each of the set of literals of P_i and that of T_j, and hence $\{P_rT_k\}$ subsumes both P_i and T_j. By construction, $\{P_rT_k\}$ subsumes both P_r and T_k. Now, since $\{P_rT_k\}$subsumes the four terms P_i, T_j, P_r and T_k, it subsumes each of the two products $\{P_iT_k\}$ (which lies in the same column as $\{P_rT_k\}$) and $\{P_rT_j\}$(which shares the same row as $\{P_rT_k\}$). In conclusion, if a general product$\{P_rT_k\}$ is to be ever absorbed by another product in the table, then we can find an absorbing product for it either in the same row *r* or in the same column *k*. To change the disjunction of products in the table into an absorptive formula, there is no need to compare every product with all other products in the table. Instead, every remaining nonzero product in the table is manipulated as follows: Either this product is absorbed in another in the same row or column or it stays unabsorbed. The number of comparisons needed to implement the *ABS* (.) operation is limited in the worst case to that of comparing each nonzero product to the nonzero products in its row and column.

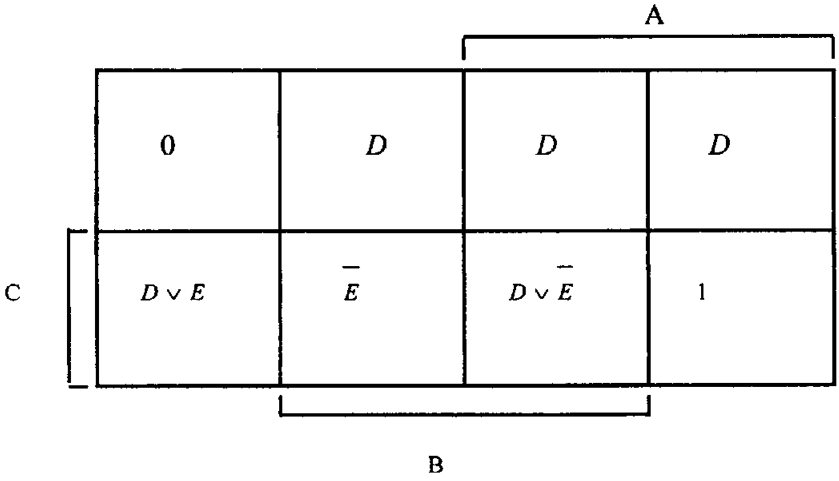
		T_j	$\cdots$	T_k	$\cdots$
.		.		.	
.		.		.	
.		.		.	
P_i	$\cdots$	$\{P_iT_j\}$	$\cdots$	$\{P_iT_k\}$	$\cdots$
.		.		.	
.		.		.	
.		.		.	
P_r		$\{P_rT_j\}$	$\cdots$	$\{P_rT_k\}$	$\cdots$
.		.		.	
.		.		.	
.		.		.	

Fig. 1. The general layout of the multiplication table of two sums of products.

Example 1:

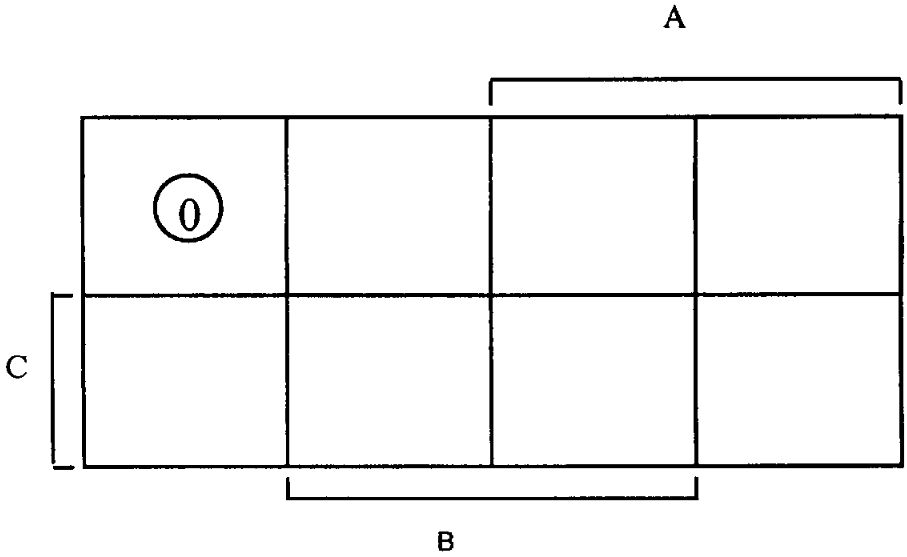
To give a simple illustration of the above procedure, we use a *CSSF* of 5 variables only, though *VEKM* techniques are usually more competitive for a larger number of variables. Let this *CSSF* be represented by the *VEKM* in Fig. 2. A standard *VEKM* procedure (see Appendix A and Fig. 3) produces thc following minimal p-o-s formula for *f*

$$\begin{aligned} f = &(A \vee B \vee C)(C \vee D)(\bar{B} \vee D \vee \bar{E})(A \vee \bar{B} \vee \bar{C} \vee \bar{E}) \\ &(A \vee B \vee D \vee E). \end{aligned} \qquad (2)$$



0	D	D	D
$D \vee E$	$\bar{E}$	$D \vee \bar{E}$	1

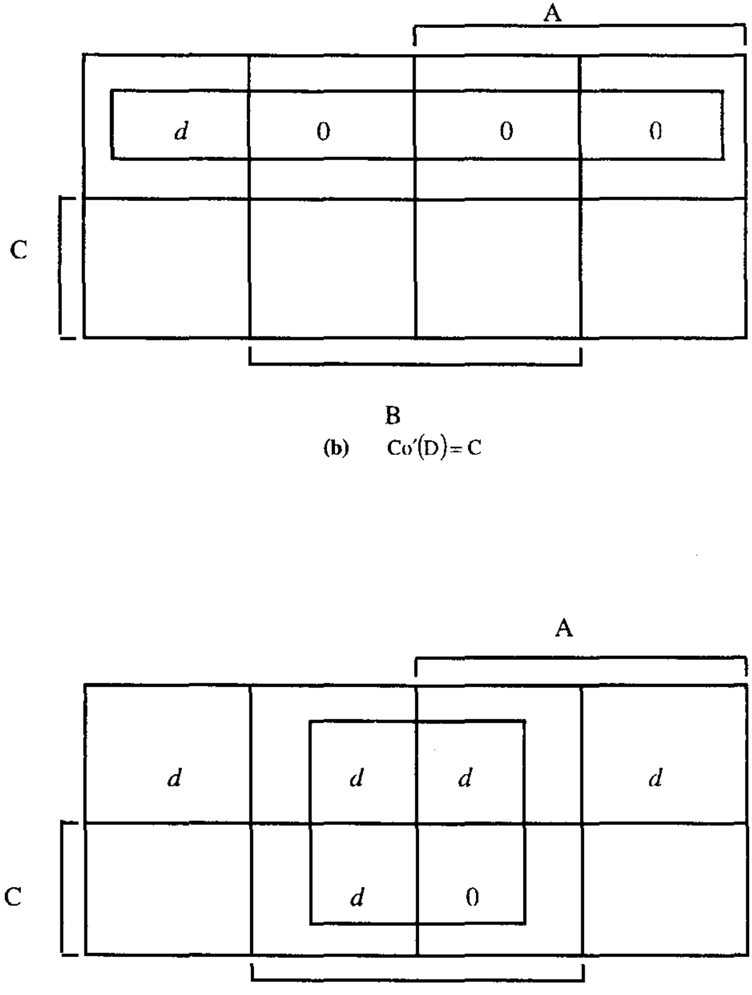
Fig. 2. A VEKM representation of a CSSF f (A,B,C,D,E).



⓪			

(a) $Co'(0) = A \vee B \vee C$

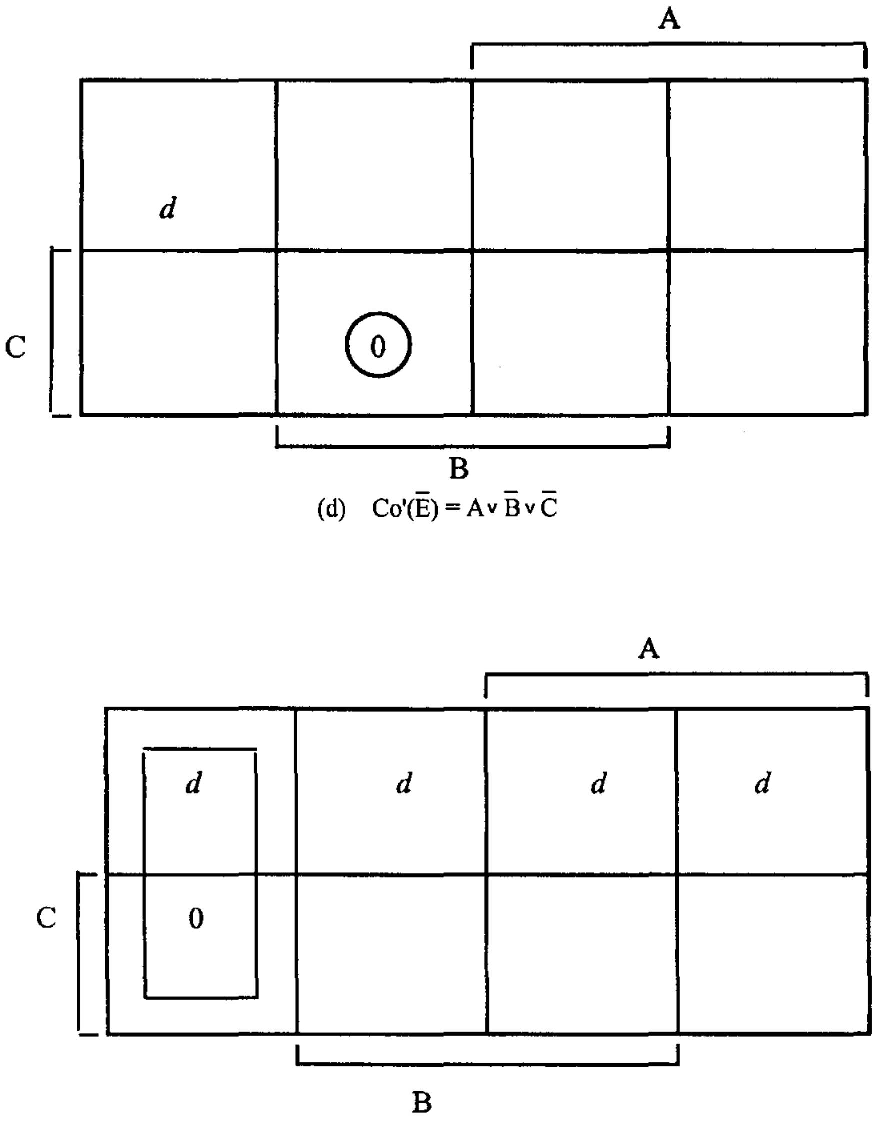
Fig. 3(a). The dual contribution of the entered alterms for the function in Fig. 2.



(b) $Co'(D) = C$

(c) $Co'(D \vee \overline{E}) = \overline{B}$

Fig. 3(b,c). The dual contribution of the entered alterms for the function in Fig. 2.



(d) $\mathrm{Co}'(\bar{E}) = A \vee \bar{B} \vee \bar{C}$

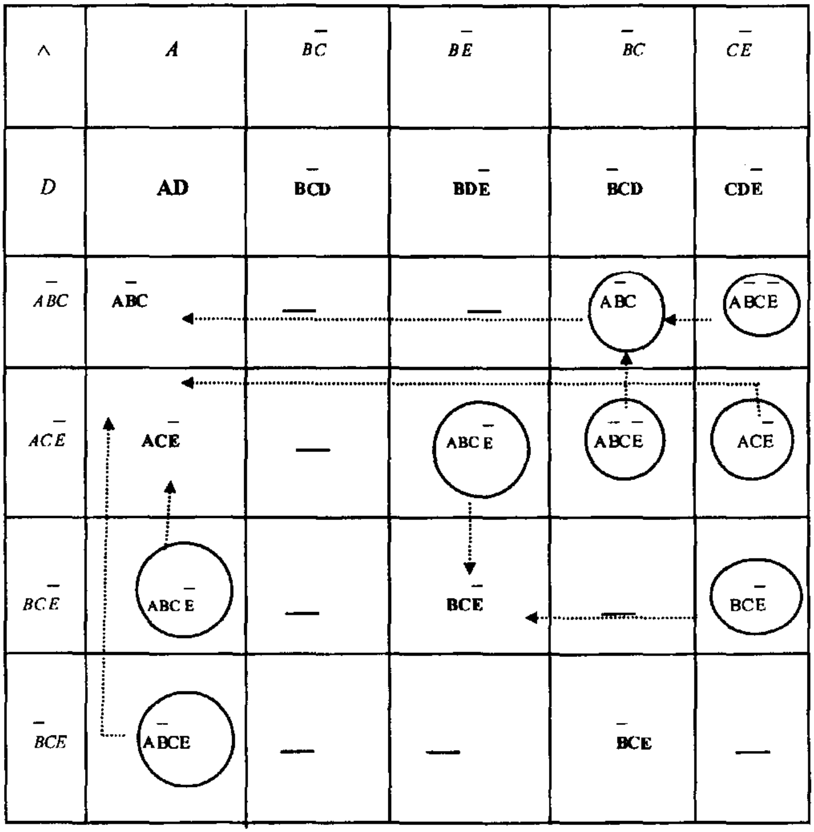
(e) $\mathrm{Co}'(D \vee E) = A \vee B$

Fig. 3(d,e). The dual contributions of the entered alterms for the function in Fig. 2.

Using (1) to multiply out the second, third and fifth alterms (which have the literal D common to all of them), and multiply out the first and fourth alterms (which have the literal A common to both of them), we have

$$
\begin{aligned}
f &= \left(D \vee C\left(\bar{B} \vee \bar{E}\right)\left(A \vee B \vee E\right)\right)\left(A \vee \left(B \vee C\right)\left(\bar{B} \vee \bar{C} \vee \bar{E}\right)\right) \\
&= \left(D \vee A\bar{B}C \vee AC\bar{E} \vee BC\bar{E} \vee \bar{B}CE\right) \\
&\quad \left(A \vee B\bar{C} \vee B\bar{E} \vee \bar{B}C \vee C\bar{E}\right)
\end{aligned}
\tag{3}
$$

Now, we use the matrix in Fig. 4 to implement the multiplication in (3). Note that we use a (__) whenever cancellation occurs, i.e., whenever the product of two terms is 0.



$\wedge$	A	$B\bar{C}$	$B\bar{E}$	$\bar{B}C$	$C\bar{E}$
D	**AD**	$B\bar{C}D$	$BD\bar{E}$	$\bar{B}CD$	$CD\bar{E}$
$A\bar{B}C$	$A\bar{B}C$	—	—	$A\bar{B}C$	$A\bar{B}C\bar{E}$
$AC\bar{E}$	$AC\bar{E}$	—	$ABC\bar{E}$	$A\bar{B}C\bar{E}$	$AC\bar{E}$
$BC\bar{E}$	$ABC\bar{E}$	—	$BC\bar{E}$	—	$BC\bar{E}$
$\bar{B}CE$	$A\bar{B}CE$	—	—	$\bar{B}CE$	—

Fig. 4. Matrix multiplication of two sums of products.

As pointed out earlier, the matrix multiplication allows an easy tracking of absorptions because of the fact that if a term is to be ever absorbed, then one of its absorbing terms will belong to either its row or to its column. Therefore, we restrict our search for an absorbing term to the same row or column as that of the product which is a candidate for being absorbed. When an absorption relation is discovered it is indicated by an arrow from the absorbed product to the absorbing product. As shown in Fig.4, such an arrow should be either horizontal or vertical. The absorbed term is now circled but it is not excluded from further consideration as a possible absorbent of other terms. After all absorption possibilities are exhausted, 9 products remain in Fig. 4 which we set in bold. These constitute all *PIs* of *f*, i.e., their disjunction is *CS* (*f*).

VEKM Folding

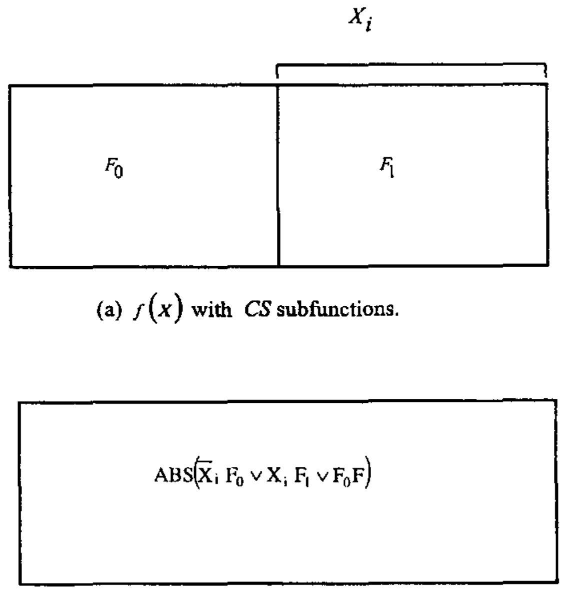
The basic idea of *VEKM* folding is derived from the Boole-Shannon expansion of a switching function $f(X)$ about a single variable X_i [1,11], namely

$$\begin{aligned} f(X) &= (\overline{X}_i \vee f_1)(X_i \vee f_0) \\ &= \overline{X}_i\ f_0 \vee X_i\ f_1 \vee f_0\, f_1, \end{aligned} \tag{4}$$

where $f_0 = f/\overline{X}_i$ and $f_1 = f/X_i$ are the subfunctions or ratios of f obtained by restricting the variable X_i in it to 0 and 1, respectively. To obtain *CS*(f), let f_0, f_1 in (4) be given by the *CS* formulas F_0 and F_1, and reduce (4) to the following form

$$\mathrm{CS}(f) = \mathrm{ABS}(\overline{X}_i\ F_0 \vee X_i\ F_1 \vee F_0\, F_1) \tag{5}$$

A formal proof of equation (5) is given by Reusch [17]. This equation has the *VEKM* interpretation depicted by Fig. 5. The *VEKM* in Fig. 5(a) has 2 cells with *CS* entries and the corresponding *VEKM* in Fig. 5(b) has a single cell with a *CS* entry. Now, we may view the function f in the previous discussion as a subfunction of some other function, which leads us to suggest the following repeated folding. If the n-variable *CSSF* $f(X)$ is given by a *VEKM* of m map variables, $0 \le m \le n$, with *CS* entries, then we can eliminate the map variables, one by one, by folding the *VEKM* with respect to the boundary of the map variable to be eliminated, X_i say. In such folding, the number of map cells is halved while each pair of map cells with opposite values of X_i and common values of the remaining map variables is replaced in accordance with Fig.5 by a single map cell of these remaining map variables. Since the starting *VEKM* has *CS* entries in all its cells, the resulting *VEKM* also has *CS* entries in all its cells. The procedure terminates when we obtain a *VEKM* of no map variables which is an algebraic expression of the *CS*.



(a) $f(x)$ with *CS* subfunctions.

(b) $f(x)$ in *CS* form.

Fig. 5. The 2-cell *VEKM* in (a) with *CS* entries is folded into a single cell in (b) with a *CS* entry.

There are three classes of terms in (5), namely those in $\bar{X}_i F_0$, $X_i F_1$ and $F_0 F_1$, where the *CS* formulas F_0 and F_1 are independent of X_i. Table 1 explores the possibility that an absorbing term and an absorbed term may belong to a certain combination of these classes. It is not possible for any term in ($\bar{X}_i F_0$) or ($X_i F_1$) to absorb any other term. This observation was earlier noted by Brown [1, p. 82]. In other words, it says that absorbing terms belong only to ($F_0 F_1$) and can cause the absorption in terms belonging to either ($\bar{X}_i F_0$) or ($X_i F_1$) or ($F_0 F_1$) itself. In view of this fact and of (4), we propose the construction of the multiplication matrix in Fig. 6 to represent (5) in the form

$$\mathrm{CS}(f) = \mathrm{ABS}\left(\left(F_1 \vee \bar{X}_i\right) \wedge \left(F_0 \vee X_i\right)\right) \tag{5a}$$

Table 1. On the possibility that an absorbing (subsumed) term and an absorbed (subsuming) term may belong to a certain pair of classes of terms

Absorbing term in Absorbed term in	$\overline{X}_i F_0$	$X_i F_1$	$F_0 F_1$
$\overline{X}_i F_0$	Impossible; F_0 is a *CS*	Impossible; A term with the X_i literal does not subsume one with the X_i literal	possible
$X_i F_1$	Impossible; A term with the X_i literal does not subsume another with the X_i literal	impossible; F_1 is a *CS*	possible
$F_0 F_1$	impossible; A term without the X_i literal does not subsume a term having it	impossible; A term without the X_i literal does not subsume a term having it	possible

The main task in Fig. 6 is to find the terms in the matrix ($F_0\ F_1$) first, and then delete absorbed terms among them, making use of the fact that if a term can be ever absorbed, then it can be absorbed by a term belonging to the same row or column. Next, we consider terms of the column vector $\overline{X}_i\ F_0$ and see whether either of them is to be absorbed by any of the terms in the same row belonging to $F_0\ F_1$. Similarly, each term in the row vector $X_i\ F_1$ is checked for possible absorption against a term in ($F_0\ F_1$) that lies in the same column.

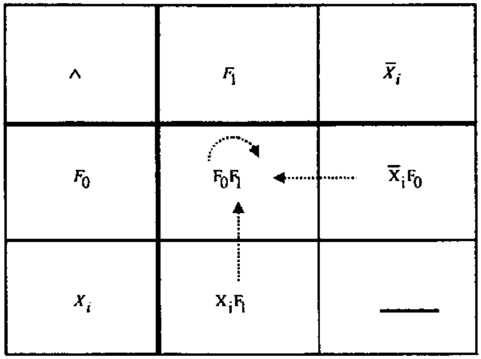


Fig. 6. A graphical illustration of Eq. (5a). the dotted arrows imply possible termwise absorptions.

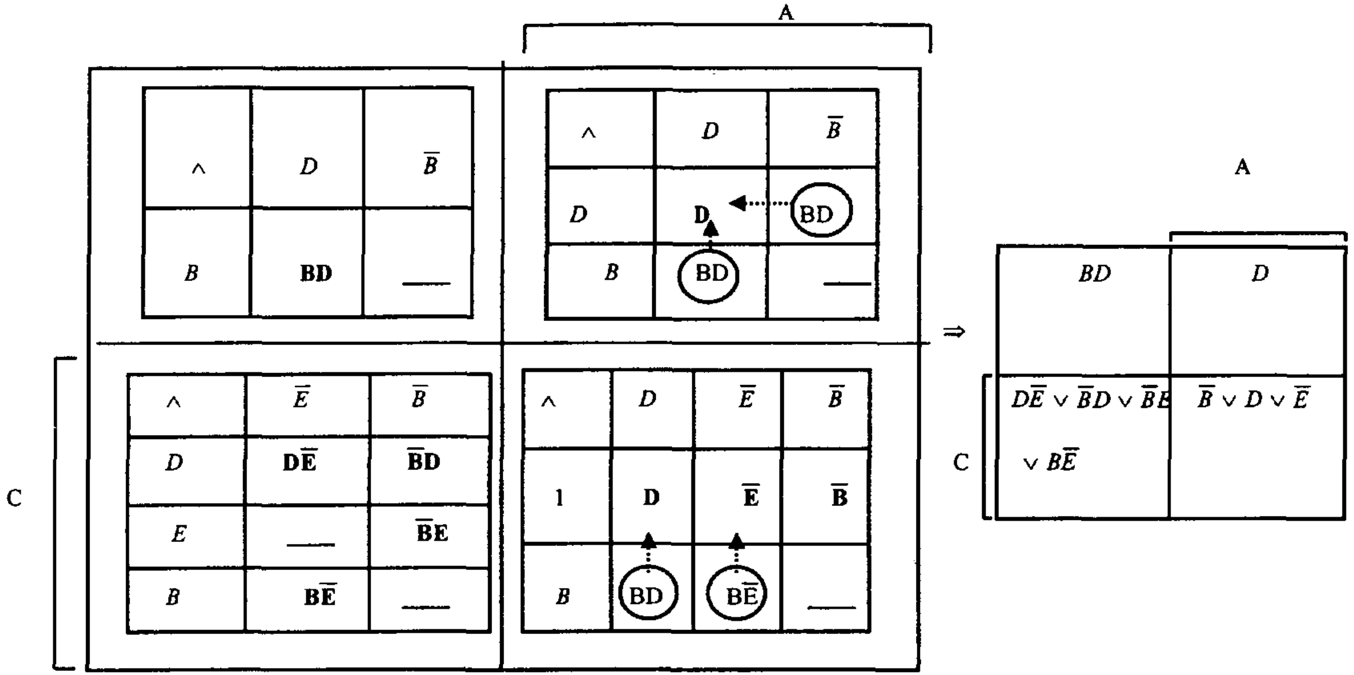


Fig. 7(a). Repeated folding of the *VEKM* in Fig. 2.

(a)

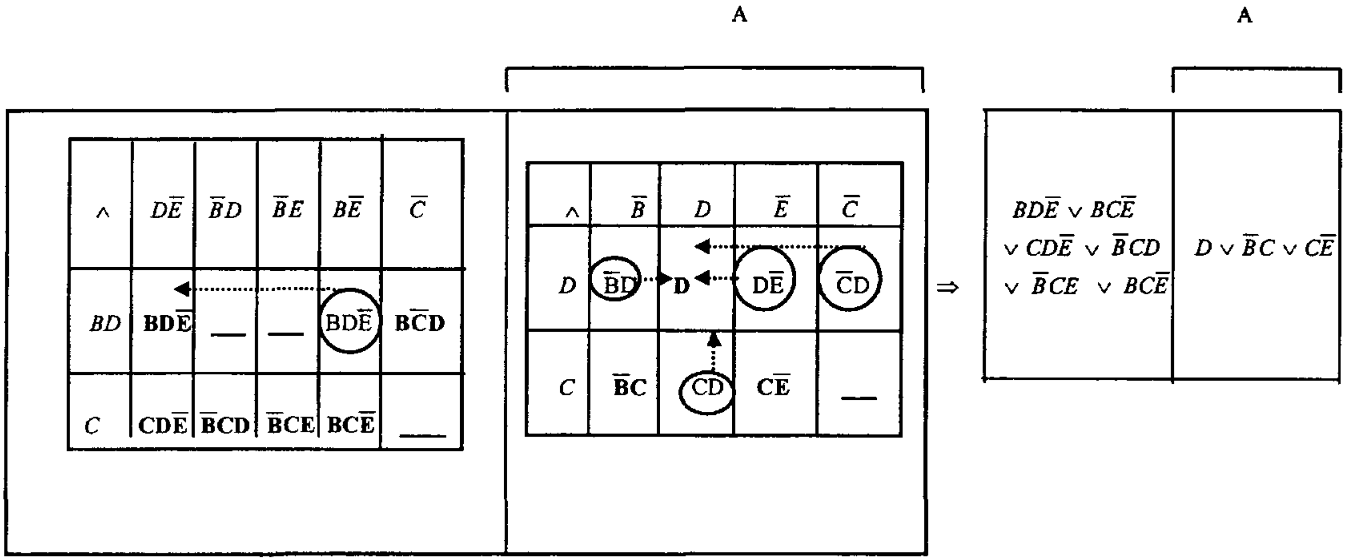
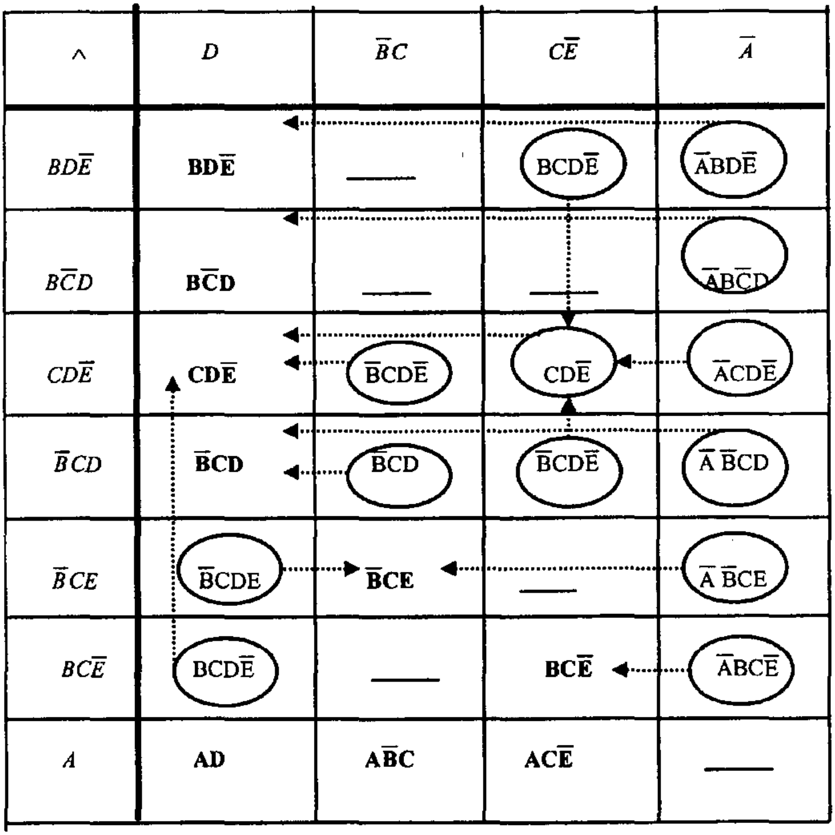


Fig. 7(b). Repeated folding of the *VEKM* in Fig. 2.

(b)



$\wedge$	D	$\overline{B}C$	$C\overline{E}$	$\overline{A}$
$BD\overline{E}$	$\mathbf{BD\overline{E}}$	—	$BCD\overline{E}$	$\overline{A}BD\overline{E}$
$B\overline{C}D$	$\mathbf{B\overline{C}D}$	—	—	$\overline{A}B\overline{C}D$
$CD\overline{E}$	$\mathbf{CD\overline{E}}$	$\overline{B}CD\overline{E}$	$CD\overline{E}$	$\overline{A}CD\overline{E}$
$\overline{B}CD$	$\mathbf{\overline{B}CD}$	$\overline{B}CD$	$\overline{B}CD\overline{E}$	$\overline{A}\,\overline{B}CD$
$\overline{B}CE$	$\overline{B}CDE$	$\mathbf{\overline{B}CE}$	—	$\overline{A}\,\overline{B}CE$
$BC\overline{E}$	$BCD\overline{E}$	—	$\mathbf{BC\overline{E}}$	$\overline{A}BC\overline{E}$
A	$\mathbf{AD}$	$\mathbf{A\overline{B}C}$	$\mathbf{AC\overline{E}}$	—

(c)

Fig. 7(c). Repeated folding of the *VEKM* in Fig. 2.

Example 2 (Example 1 revisited)

The *VEKM* in Fig. 2 has *CS* entries. It is folded in Fig. 7(a) with respect to the map variable B, then in Fig. 7(b) with respect to C, and finally in Fig. 7(c) with respect to A. For instructive purposes, we are writing the pertinent subfunctions as products of a matrix multiplication in accordance with Fig. 1. Absorbed terms are circled and deleted. Of course, it is not necessary to actually construct all the multiplication matrices for the earlier simple and numerous subfunctions in Figs.7(a) and (b) where algebraic manipulation can be straightforward. The use of a multiplication matrix in Fig. 7(c) is, however, a saving from tedious and cumbersome manipulations. The remaining products in Fig. 7(c) are set in bold. They are exactly the same 9 *PIs* obtained previously in Fig. 4.

Now, we consider an important simplification for the rules of *VEKM* folding. Let the two *CS* formulas F_0 and F_1 in (5) or (5a) have some common terms that are given by an s-o-p formula *T*, namely

$$F_0 = T \vee G_0 , \tag{6a}$$

$$F_1 = T \vee G_1 , \tag{6b}$$

where G_0 and G_1 are the s-o-p formulas that remain after excluding the common terms T from F_0 and F_1, respectively. Note that since the terms of T are a subset of those of the CS s F_o and F_1, the formula T must be absorptive. In fact, it is the sum of all the prime implicants of F that are independent of X_i. Now, Eq. (5a) is replaced in accordance with (1) by

$$\begin{aligned} CS(f) &= ABS\left(\left(T \vee G_1 \vee \overline{X}_i\right) \wedge \left(T \vee G_0 \vee X_i\right)\right) \\ &= ABS\left(T \vee \left(G_1 \vee \overline{X}_i\right) \wedge \left(G_0 \vee X_i\right)\right), \end{aligned} \tag{7a}$$

which can be further simplified to

$$CS(f) = T \vee ABS\left(G_0 G_1 \vee \overline{X}_i G_0 \vee X_i G_1\right). \tag{7b}$$

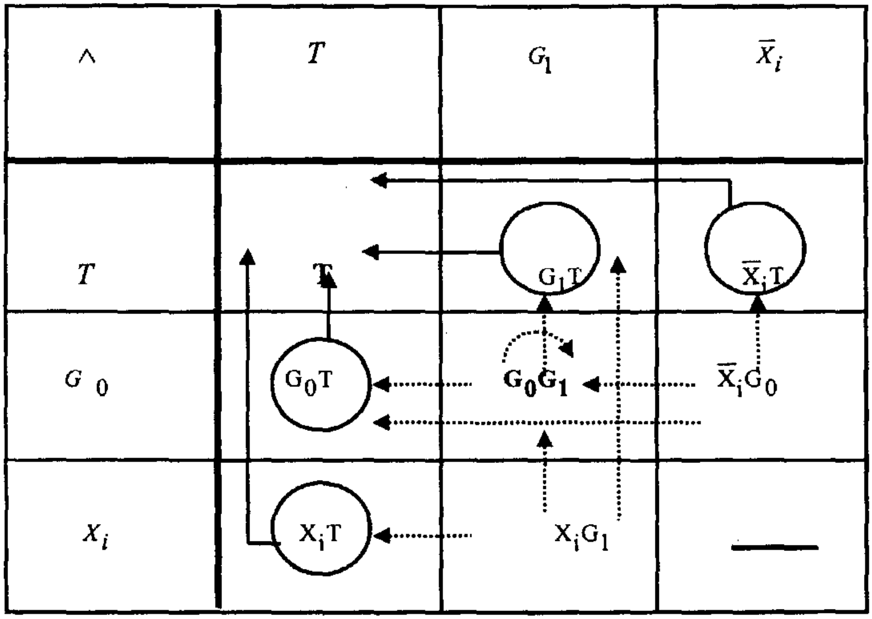
Equation (7a) is represented by the multiplication matrix in Fig. 8 while Eq. (7b) is represented by its simplified version of Fig. 9. In Fig. 8, solid arrows denote actual termwise absorptions by terms in the same row (for horizontal arrows) or by terms in the same column (for vertical arrows), while dotted arrows indicate possible absorptions. The following comparisons suffice for implementing absorptions, wherein comparison continues till the compared term is absorbed or till the set of terms to which it is compared is exhausted.

a) Compare every nonzero term in the column vector $\overline{X}_i G_0$ to nonzero terms in the same row in the matrix $G_0 G_1$ and to terms of T.

b) Compare every nonzero term in the row vector $X_i G_1$ to nonzero terms in the same column in the matrix $G_0 G_1$ and to terms of T.

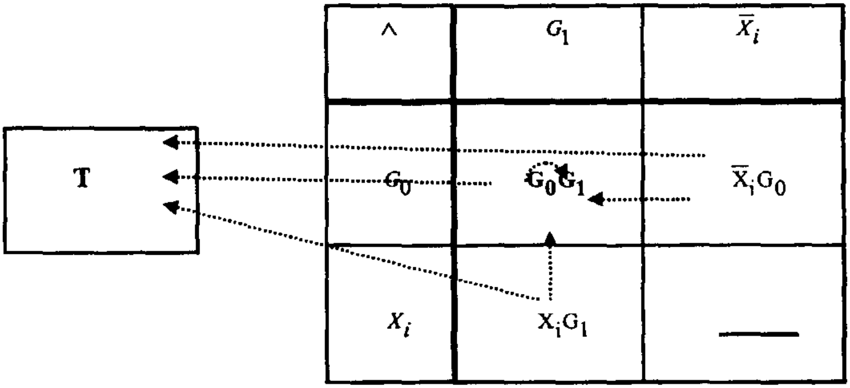
c) Compare every nonzero term in the matrix $G_0 G_1$ to nonzero terms in the same row or the same column of the matrix $G_0 G_1$ and to terms of T.

Equation (7b) can be used in its simplified graphical form of Fig. 9 to simplify the multiplication matrix in cell $A\overline{C}$ in Fig. 7(a) and that in cell A in Fig. 7(b).



$\wedge$	T	G_1	$\overline{X}_i$
T	T	G_1T	$\overline{X}_iT$
G_0	G_0T	G_0G_1	$\overline{X}_iG_0$
X_i	X_iT	X_iG_1	—

Fig. 8. Illustration of the multiplication in Eq. (7).



$\wedge$	G_1	$\overline{X}_i$
G_0	G_0G_1	$\overline{X}_iG_0$
X_i	X_iG_1	—

Fig. 9. A simplification of Fig. 8.

An Improved Tison Method

It is instructive to note that our result (7b) represents a typical step in Tison method for obtaining the *CS* of a *CSSF* f (see Appendix B). The only difference is that Tison method (in any step other than the last one) neither necessitates nor guarantees the use of *CS* expressions F_0, F_1 but might use other expressions for the functions f_0 and f_1. That is why it does not yield a *CS* for f in a single step, but has to repeat its typical step for all biform variables X_i. In fact, the typical step in Tison method starts by arranging a given expression for f with respect to a biform variable X_i in the form

$$f = g_0\overline{X}_i \vee g_1 X_i \vee t, \tag{8}$$

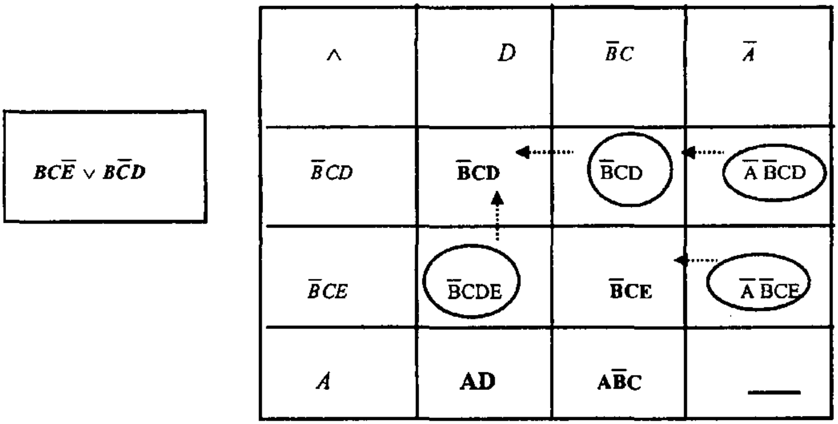
where g_0, g_1, and t are formulas that are independent of X_i. Next, f is augmented by all consensi between terms in $g_0\overline{X}_i$ and g_1X_i (which turn out to be the nonzero products in $g_0\,g_1$), and then subsuming terms are absorbed. The method repeats this typical step for all biform variables ending with the *CS* of f after the last step. If in Fig. 9 we use the lower case letters g_0, g_1, and t to indicate general s-o-p formulas and not necessarily the *CS* formulas G_0, G_1, and T, then Fig. 9 suggests an economic layout for implementing Tison method with a restricted number for the comparisons needed for implementing absorptions. In such a layout arrows should be added from t to g_0g_1 and to t itself since t (unlike T) is not guaranteed to consist of prime implicants only. However, the need to consider possible absorptions of terms of t as implied by these arrows gradually diminishes and ceases to exist for the last step at which t becomes a disjunction of prime implicants only This observation is particularly useful for our present techniques, since we usually rely on Tison method for securing initial *VEKM* entries in *CS* forms. In the following example, we demonstrate our matrix implementation of Tison method. We observe that this improved version of Tison method (though much better than the original version) remains inferior to *VEKM* techniques, mainly because it handles large matrices right from the beginning and it requires a cumbersome step of rearrangement of terms at each step. This observation confirms our belief that *VEKM* techniques which combine both merits of map and algebra cannot be surpassed by purely algebraic methods such as Tison method.

Example 3 (Example 1 revisited)

The function represented by the *VEKM* in Fig. 2 has the s-o-p formula

$$f = AD \vee A\overline{B}C \vee BC\overline{E} \vee B\overline{C}D \vee \overline{A}\overline{B}CD \vee \overline{A}\overline{B}CE\ . \tag{9}$$

Figure 10 illustrates our improved implementation of the typical step of Tison method generating consensi with respect to the 4 biform variables *A*, *B*, *C* and *E*, respectively. The starting expression for each step (as indicated below its matrix implementation) is obtained as a rearrangement of the result of the previous step. The final outcome from the last step in Fig. 10(d) is exactly the same 9 PIs obtained previously in each of Figs. 4 and 7(c).



$BC\bar{E} \vee B\bar{C}D$

$\wedge$	D	$\bar{B}C$	$\bar{A}$
$\bar{B}CD$	$\mathbf{\bar{B}CD}$	$\bar{B}CD$	$\bar{A}\,\bar{B}CD$
$\bar{B}CE$	$\bar{B}CDE$	$\mathbf{\bar{B}CE}$	$\bar{A}\,\bar{B}CE$
A	**AD**	$\mathbf{A\bar{B}C}$	———

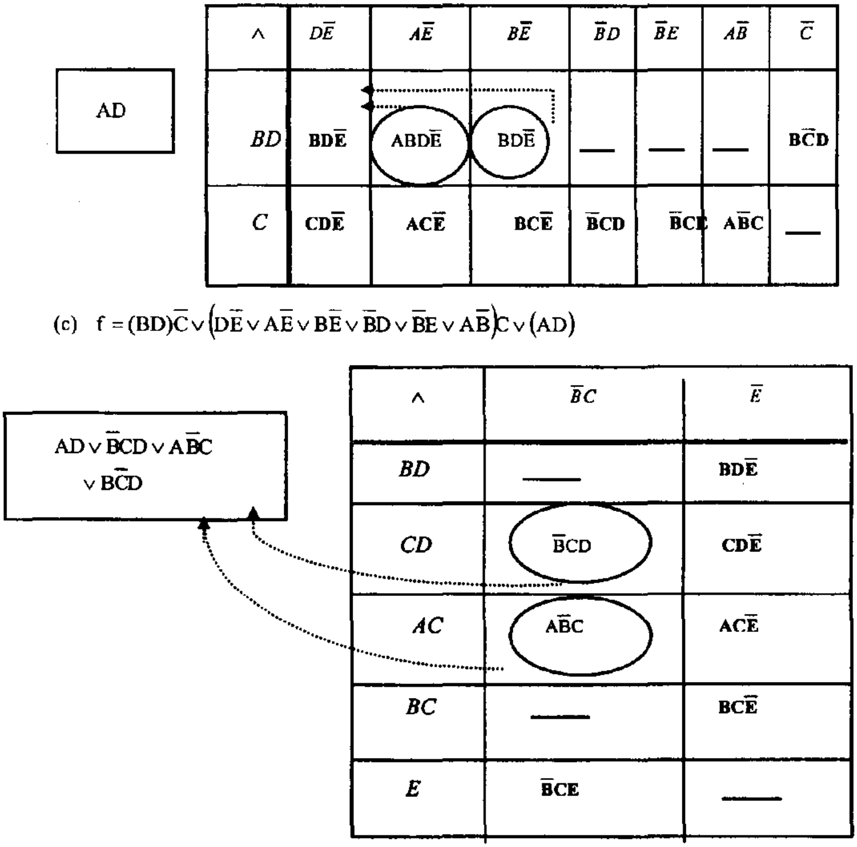
(a) $f = (\bar{B}CD \vee \bar{B}CE)\bar{A} \vee (D \vee \bar{B}C)A \vee (BC\bar{E} \vee B\bar{C}D)$

AD

$\wedge$	$C\bar{E}$	$\bar{C}D$	$\bar{B}$
CD	$\mathbf{CD\bar{E}}$	———	$\mathbf{\bar{B}CD}$
CE	———	———	$\mathbf{\bar{B}CE}$
AC	$\mathbf{AC\bar{E}}$	———	$\mathbf{A\bar{B}C}$
B	$\mathbf{BC\bar{E}}$	$\mathbf{B\bar{C}D}$	———

(b) $f = (CD \vee CE \vee AC)\bar{B} \vee (C\bar{E} \vee \bar{C}D)B \vee (AD)$

Fig. 10. (Matrix implementation of Tison method (a,b).



AD

$\wedge$	$D\overline{E}$	$A\overline{E}$	$B\overline{E}$	$\overline{B}D$	$\overline{B}E$	$A\overline{B}$	$\overline{C}$
BD	$BD\overline{E}$	$ABD\overline{E}$	$BD\overline{E}$	—	—	—	$B\overline{C}D$
C	$CD\overline{E}$	$AC\overline{E}$	$BC\overline{E}$	$\overline{B}CD$	$\overline{B}CE$	$A\overline{B}C$	—

(c) $f = (BD)\overline{C} \vee \left(D\overline{E} \vee A\overline{E} \vee B\overline{E} \vee \overline{B}D \vee \overline{B}E \vee A\overline{B}\right)C \vee (AD)$

$AD \vee \overline{B}CD \vee A\overline{B}C$
$\vee B\overline{C}D$

$\wedge$	$\overline{B}C$	$\overline{E}$
BD	—	$BD\overline{E}$
CD	$\overline{B}CD$	$CD\overline{E}$
AC	$A\overline{B}C$	$AC\overline{E}$
BC	—	$BC\overline{E}$
E	$\overline{B}CE$	—

(d) $f = (BD \vee CD \vee AC \vee BC)\overline{E} \vee \left(\overline{B}C\right)E \vee \left(AD \vee \overline{B}CD \vee A\overline{B}C \vee B\overline{C}D\right)$

Fig. 10. (Matrix implementation of Tison method.

Conclusion

Two original procedures of *VEKM* multiplication and folding are introduced which serve to extract all the prime implicants of a switching function and hence obtain its complete sum. Dual versions of these procedures should extract all the prime implicates of the function and use them to construct its complete product. Details of these dual versions can be worked out if one follows the duality rules in [11]. All procedures are efficient for manual use with medium-sized problems of up to 12 variables or even more. They combine the pictorial visualization of the map with the simplicity of straightforward algebraic manipulations.

Moreover, the use of a novel multiplication matrix or table is proposed. The entries of the matrix are shown to enjoy a very useful property, namely, that if an entry is to be ever absorbed by another, then it must be absorbable by an entry in the same row or in the same column. Therefore, the number of term comparisons needed for implementing absorptions can be dramatically decreased. The multiplication matrix is used to advantage with the new procedures as well as with the classical algebraic method of Tison.

In contrast with *VEKM* multiplication which relies on map heuristics, *VEKM* folding is algorithmic in nature. Moreover, the rules of VEKM folding are easily simplified to utilize the existence of some common terms for the CS's of the pertinent subfurctions. Such a simplification leads to an improved version of *VEKM* folding, which inherently uses intelligent multiplication to reduce the number of term comparisons needed. This version is truly promising as a basis for an efficient computer code for extracting all the prime implicants and prime implicates of a *CSSF.*

Obtaining the complete sum/product can be an end in its own right as justified by the numerous applications cited in the introduction. Alternatively, it may be only a part of the minimization process. To complete the job in this latter case, a sequel to this work appears in a forthcoming paper [18] that uses *VEKM* or *VEKM* related methods to obtain an exactly minimal sum/product starting with the complete sum/product.

References

[1] Brown, F.M .*Boolean Reasoning: The Logic of Boolean Equations.* Boston: Kluwer Academic Publishers, 1990.

[2] Gregg, J.R. *Ones and Zeros: Understanding Boolean Algebra, Digital Circuits, and the Logic of Sets.* New York: IEEE Press, 1998.

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

[4] Fletcher, W.I .*An Engineering Approach to Digital design* Englewood- Cliffs, NJ: Prentice Hall,1980.

[5] Clare, C. R. *Designing Logic Systems Using State Machines* .New York: McGraw-Hill, 1973.

[6] Rushdi, A. M. "Symbolic Reliability Analysis with the aid of Variable- Entered Karnaugh Maps." *IEEE Transactions on Reliability,* R 32, No. 2 (1983), 134-139.

[7] Rushdi, A. M. "Map Derivation of the Minimal Sum of a Switching Function from That of Its Complemen." *Microelectronics and Reliability,* 25, No. 6 (1985), 1055-1065.

[8] Ercegovac, M.D. and Lang, T. *Digital Systems and Hardware/ Firmware Algorithms* .New York: Wiley, 1985.

[9] Roth, Jr., C.H. *Fundamentals of Logic Design,* 3rd ed. (St. Paul, MN: West Publishing, 1995.

[10] Rushdi, A.M. "Map Differentiation of Switching Functions." *Microelectronics and Reliability,* 26, No. 5 (1986), 891-908.

[11] Rushdi, A.M. "Improved Variable-Entered Karnaugh Map Procedures." *Computers and Electrical Engineering,* 13, No. 1 (1987), 41-52.

[12] Rushdi, A.M. "Logic Design of NAND (NOR) Circuits by the Entered-Map-Factoring Method." *Microelectronics and Reliability,* 27, No. 4 (1987), 693-701.

[13] Brown, F.M. and Distler, R.J. "An Improved State Diagram." *IEEE Transactions on Education,* 34, No. 2 (1991), 199-203.

[14] Unger, S. H. *The Essence of Logic Circuits*. (2nd *ed.*). New York: IEEE Press, 1997.

[15] Rushdi, A.M. "Karnaugh Map." *Encyclopaedia of Mathematics*, Vol. II, Hazewinkel, M. (Ed.) Boston: Kluwer Academic Publishers, 1996.

[16] Nelson, R. J. "Simplest Normal Truth Functions." *J. Symbolic Logic*. 20(1955), 105-108.

[17] Reusch, B. "Generation of Prime Implicants from Subfunctions and a Unifying Approach to the Covering Problem." *IEEE Transaction on Computers*, C-24, No. 9 (1975), 924-930.

[18] Rushdi, A. M. and Al-Yahya, H. A. "Variable-entered Karnaugh Map Procedures for Obtaining the Irredundant Disjunctive Forms of a Switching Function from Its Complete Sum." *Journal of King Saud University* (Engineering Sciences). Accepted for publication in Vol. 14 No. (1).

[19] Tison, P. "Generalization of Consensus Theory and Application to the Minimization of Boolean Functions." *IEEE Transactions on Electronic Computers*, EC-16, No. 8 (1967), 446-456.

[20] Cutler, R.B., Kinoshita, K. and Muroga, S. *Exposition of Tison's Method to Derive all Prime Implicants and all Irredundant Disjunctive Forms for a Given Switching Function* .Report No. UIUCDCS-R-79-993. Department of Computer Science, University of Illinois, Urbana, 1979.

APPENDIX A
On the Variable-entered Karnaugh Map

Classically, the variable-entered Karnaugh map (*VEKM*) has been developed to double the variable-handling capability of the conventional Karnaugh map (*CKM*). The *VEKM* is also more useful in handling functions with complex intrinsic structure. For example, given the 4-variable *CSSF* of Fig. 11, one may find its *PIs* too many to be visualized easily on the *CKM* [3, p.186]. However, either of the equivalent *VEKM* representations in Fig. 12 can easily be used to extract all *PIs* of the given *CSSF*. Moreover, the *VEKM* can be used to represent Boolean functions of the form f: $B^{m} \to B^{n-m}$, where $B = \{0,1\}$, or equivalently of the form f: $B^{k} \to B$, where $B = \{0,1\}^{j}$, $j > 1$, is the Boolean carrier of 2^{j} elements. This is a direct consequence of the fact that a Boolean function is completely defined by the 0, 1 assignments for each of its arguments and not necessarily by all the 2^{j} possible assignments of an argument [1].

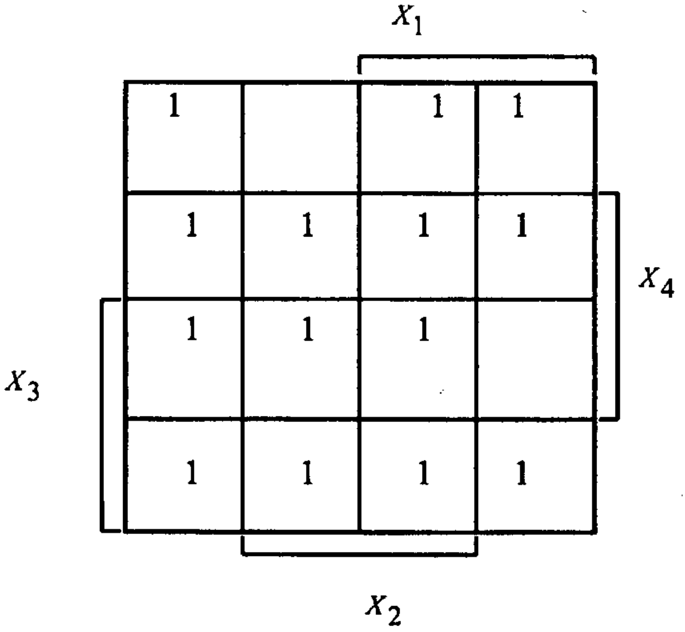
The construction of a *VEKM* is based on the general Boole-Shannon expansion of Boolean functions [1,11], a statement of which in p-o-s form is given below. If f: $B^{n} \to B$ is a Boolean function of n variables, with B denoting a Boolean carrier such as $\{0,1\}$ or $\{0,1,a,\overline{a}\}$, etc., then the function can be expanded about m of its n variables, $0 \le m \le n$, as follows. Let X_p, X_e, X denote respectively the m-tuple, the $(n$-$m)$-tuple and the n-tuple

$$X_p = [X_1, X_2, ..., X_{m-1}, X_m], \tag{A.1}$$

$$X_e = [X_{m+1}, X_{m+2}, ..., X_n], \tag{A.2}$$

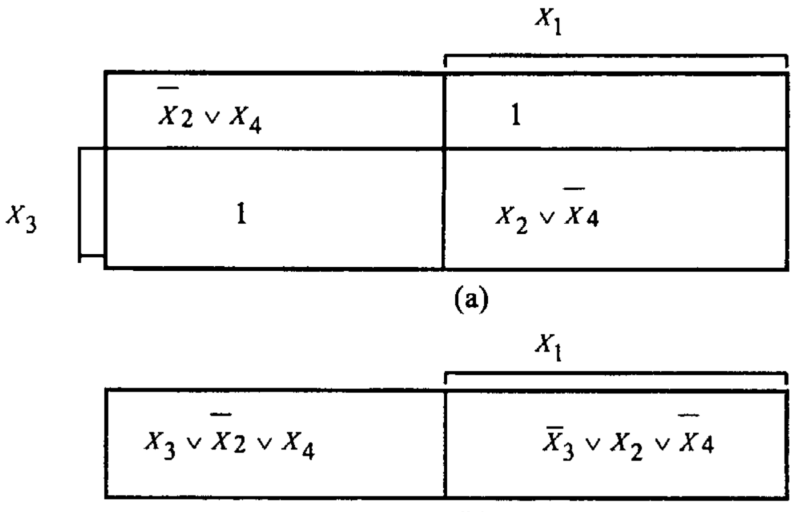
$$X = [X_p, X_e], \tag{A.3}$$

then the Boolean function $f(X) = f(X_p, X_e)$ can be written in the form



1		1	1
1	1	1	1
1	1	1	
1	1	1	1

Fig. 11. A 4-variable *CKM* representing a *CSSF* with unusually numerous *PIs*.



$\overline{X}_2 \vee X_4$	1
1	$X_2 \vee \overline{X}_4$

(a)

$X_3 \vee \overline{X}_2 \vee X_4$	$\overline{X}_3 \vee X_2 \vee \overline{X}_4$

(b)

Fig. 12. Two VEKM representations of the CSSF in Fig. 11.

$$f(X_1, X_2, ..., X_{m-1}, X_m, X_e) = (X_1 \vee X_2 \vee ... \vee X_{m-1} \vee X_m \vee f_0) \wedge \\ (X_1 \vee X_2 \vee ... \vee X_{m-1} \vee \overline{X}_m \vee f_1) \wedge \\ .... \wedge \left(\overline{X}_1 \vee \overline{X}_2 \vee ... \vee \overline{X}_{m-1} \vee \overline{X}_m \vee f_{(2^m - 1)}\right) \qquad (A.4)$$

where

$$f_0 = f(0, 0, ..., 0, 0, X_e),$$
$$f_1 = f(0, 0, ..., 0, 1, X_e),$$
$$.$$
$$.$$
$$.$$
$$f_{(2^m - 1)} = f(1, 1, ..., 1, 1, X_e). \qquad (A.5)$$

are called subfunctions, restrictions, residuals or ratios of the original function $f(X)$. Each of them is obtained from $f(X)$ through certain assignments or specifications to its m expansion variables. Therefore, each subfunction is a function of the remaining $(n\text{-}m)$ variables X_e only, though, not necessarily a genuine function of all of them. The expansion (A.4) can be used to represent $f(X)$ in terms of a map of m map variables, namely $X_1, X_2, ..., X_m$. Such a map is called a variable-entered Karnaugh map (*VEKM*) since the entries of its 2^m cells are formulas representing the (generally) variable subfunctions $f_0, f_1, ..., f_{(2^m - 1)}$. For a map cell in which the sequence $X_1 X_2 ... X_{m-1} X_m$ is of a value equal to the binary representation for the integer i, the entry of the cell is f_i. When the expansion (A.4) is associated with a *VEKM* representation the m expansion variables X_p are called keystone or map variables while the e=(m-n) remaining variables are called entered variables.

Without loss of generality, the variables $X_1, X_2, ..., X_n$ of the function f are assumed to be arranged such that the first m among them are the map variables. Usually, we choose these m variables as the most frequently used variables, i.e., as the variables that appear most in a typical algebraic expression of the function. For a given number m where $0 \le m \le n$, there is $\binom{n}{m}$ possible choices for the set of map variables. Therefore, there are $\binom{n}{m}$ types of expansions or *VEKM* representations for the Boolean function $f(X)$. The total number of such *VEKM* representations is

$$N = \sum_{m=0}^{n} \binom{n}{m} = 2^n. \tag{A.6}$$

Note that a *VEKM* representation is unique only as far as the choice of its map and entered variables is concerned, but there are usually many possible formulas for its subfunctions. Out of the 2^n *VEKM* representations, the 2 extreme ones are not genuine *VEKMs*. For m=0 and e=n, a *VEKM* degenerates into a purely algebraic expression, while for m=n and e=0 a *VEKM* degenerates into a conventional or classical Karnaugh map (*CKM*), i.e., to a map of constant rather than variable entries. Genuine *VEKMs* serve as intermediaries between these two extreme cases of pure algebra and pure mapping.

Ways of transformations among various *VEKM* representations are discussed in [11]. If the number of map variables is to be increased from m_1 to $m_2 > m_1$ say, then each of the 2^{m_1} subfunctions of the original *VEKM* is expanded about the new $(m_2 - m_1)$ map variables according to (A.4) to yield $2^{(m_2 - m_1)}$ subfunctions of its own, with a number of 2^{m_2} subfunctions emerging for the new *VEKM*. On the other hand, if the number of map variables is to be decreased from m_2 to m_1, then the original *VEKM* is divided into 2^{m_1} disjoint partitions each of which consists of $2^{(m_2 - m_1)}$ of the original *VEKM* cells which have common values for the $(m_2 - m_1)$original map variables that are to be switched into entered variables. These partitions represent *VEKM* representations for the subfunctions of the new *VEKM* and hence can be used to produce algebraic expressions for them.

If the entries of the *VEKM* representing a *CSSF* f are set in minimal p-o-s forms, then an irredundant conjunctive form f is given by [11]

$$f = \bigwedge_{r} (I_r \vee Co'(\, I_r \,)), \tag{A.7}$$

where I_r is an alterm that appears in at least one *VEKM* cell (i.e., it is a prime implicate of the corresponding subfunction(s) of f), while $Co'(\, I_r \,)$ (which is called the dual contribution of I_r) is to be set in minimal p-o-s form through its *CKM* representation which is deduced from the original VEKM through the rules [11]:

1. Regard as a d-cell any cell containing
   i. an alterm strictly subsumed by I_r, or
   ii. two or more alterms whose consensus is subsumed by I_r.

2. Regard as a 0-cell any cell containing I_r. However, if such a cell is already covered in each of the *CKMs* representing the dual contributions of alterms whose consensus is subsumed by I_r, then regard this cell as a *d*-cell.
3. Regard the remaining cells as 1-cells (leave them blank)

Example 4 (a prelude to Example 1)

A 5-variable CSSF is represented by the *VEKM* of Fig. 2 whose entries have been already set in minimal p-o-s forms. Five alterms appear in the cells of this *VEKM*, namely 0, D, $\left(D \vee \bar{E}\right)$, , $\left(D \vee E\right)$. Note, in particular, that the 1 entry in cell $A\bar{B}C$ is not an alterm. The minimal p-o-s dual contributions of these alterms are obtained from the *CKMs* in Fig. 3(a)-(e). We explain the construction of these *CKMs* by discussing the case of Fig. 3 (c) in detail. Here, we have a *CKM* for $Co'\left(D \vee \bar{\mathrm{E}}\right)$. Cell *ABC* is a 0-cell since the alterm $\left(D \vee \bar{\mathrm{E}}\right)$ appears therein (Rule 2). Since the alterm $\left(D \vee \bar{\mathrm{E}}\right)$ strictly subsumes the alterms D, $\bar{E}$, 0, each of the 5 cells containing either one of these latter alterms is a *d*-cell (Rule 1). The remaining 2 cells are 1-cells and are left blank. We use standard *CKM* covering methods to produce a minimal p-o-s cover for $Co'\left(D \vee \bar{\mathrm{E}}\right)$ in Fig. 3(c), which consists of the single alterm loop $\bar{B}$. According to (A.7), a prime implicate $\left(\left(D \vee \bar{\mathrm{E}}\right) \vee Co'\left(D \vee \bar{\mathrm{E}}\right)\right) = \left(D \vee \bar{E} \vee \bar{B}\right)$ should appear in (2). Other prime implicates in (2) are produced similarly.

APPENDIX B
Tison Method for Obtaining *CS* (*F*)

Tison method for obtaining all the prime implicants of a switching function f (i.e., obtaining *CS*(f)) is a systematic streamlined version of the iterative-consensus technique. The original work of Tison appeared in [19], but a more readable exposition can be found in [20] or in [3, pp. 103-105]. The method is sometimes called "Tison method" for short, though its lengthier name serves to differentiate it from another Tison method, namely, that for the derivation of all irredundant disjunctive forms [3, pp. 198-211]. The essence of the present Tison method is summarized in Theorem 2.

Theorem 2:

Start with a set of $n(0)$ products $S_0 = \{T_1^{(0)}, T_2^{(0)}, \cdots, T_{n(0)}^{(0)}\}$ with m biform variables $X_1, X_2, \cdots, X_m$ and a switching function f that is the disjunction of the products in S_0. For $1 \le i \le m$ repeat the following 2-part step that replaces a set of products S_{i-1} by another S_i:

I. For $1 \le j \le k \le n(i-1)$, if X_i appears complemented in one of the two products $T_j^{(i-1)}$ and $T_k^{(i-1)}$ and appears uncomplemented in the other such that they have no other opposition, then they have a consensus with respect to X_i. Form that consensus and add it to S_{i-1}. Finally, S_{i-1} is replaced by a superset S'_{i-1} of l (i-1) elements, where $l(i-1) \ge n(i-1)$.

II. Consider every pair $\{T_j^{(i-1)}, T_k^{(i-1)}, \cdots, j \ne k\}$ of (so far remaining) products in S'_{i-1}. If $T_j^{(i-1)}$ subsumes $T_k^{(i-1)}$, then delete $T_j^{(i-1)}$. Otherwise, if $T_j^{(i-1)}$ is subsumed by $T_k^{(i-1)}$ then delete $T_k^{(i-1)}$. Whenever all subsumptions (and subsequent deletions) are exhausted, let the remaining set be $S_i = \{T_1^{(i)}, T_2^{(i)}, \cdots, T_{n(i)}^{(i)}\}$.

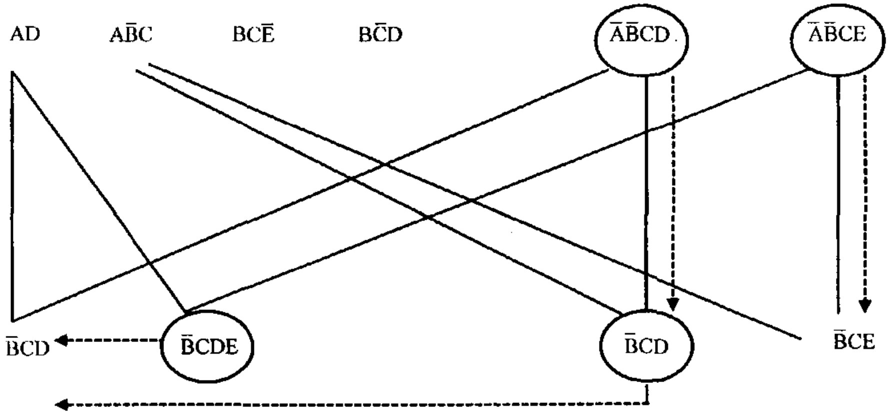
The disjunction of products in any of the sets S_i, $1 \le i \le m$ is an expression of f, and the final set S_m consists of all prime implicants of f.

Example 5 (a prelude to Example 3)

The function represented by (9) is a disjunction of the 6 products in the set

$$S_0 = \left(AD, A\bar{B}C, BC\bar{E}, B\bar{C}D, \bar{A}\,\bar{B}CD, \bar{A}\,\bar{B}CE\right), \qquad (B.1)$$

which has 4 biform variables that we arbitrarily chose to order them as A, B, C and D. Every term in S_0 is compared to its successors to form consensi with respect to the biform variable A. Fifteen comparisons are needed to produce 4 such consensi as shown by the solid lines in Fig. 13(a) which mimics the graphical representation of Muroga [3]. These consensi are written in the bottom line in Fig. 13(a), and when added to the original 6 products of form a new set S'_0 of 10 products. Now possible subsumptions among every pair of remaining products are considered. The number of subsuming products detected and deleted is 4. These are shown circled in Fig. 13(a) wherein a subsumption relation is indicated by a dotted line. The remaining (uncircled) products in Fig. 13(a) constitute set S_1. Similar work is needed to obtain sets S_2, S_3 and S_4 (by forming consensi with respect to B, C and D respectively and subsequently deleting subsuming terms), The final set S_4 consists of all *PIs* of *f*.



(a)

(a) Mur

Fig. 13

Actual comparisons, (C) Reduced comparisons.

In Fig. 13(b) we show an equivalent representation for the first step in Tison method (the one used in producing consensi with respect to A). We use the lower triangular part of a matrix whose keys are the 6 products in (B.1). Now it is clear that $\binom{6}{2} = 15$ comparisons are needed. To minimize the number of unnecessary comparisons, we use a reduced matrix whose horizontal and vertical keys are the products in (B.1) containing A and $\bar{A}$ respectively and from which other products (those independent of A) are excluded. Now, we have a representation essentially equivalent to our improved one in Fig. 10(a). However, in Fig. 10(a) we use simple multiplication rather than consensus generation.

$A\bar{B}C$	—				
$BC\bar{E}$	—	—			
$B\bar{C}D$	—	—	—		
$\bar{A}\bar{B}CD$	$\bar{B}CD$	$\bar{B}CD$	—	—	
$\bar{A}\bar{B}CE$	$\bar{B}CDE$	$\bar{B}CE$	—	—	—
	AD	$A\bar{B}C$	$BC\bar{E}$	$B\bar{C}D$	$\bar{A}\bar{B}CD$

(b)

(b) Actual comparisons

$BC\bar{E} \vee B\bar{C}D$	AD	$A\bar{B}C$
$\bar{A}\bar{B}CD$	$\bar{B}CD$	$\bar{B}CD$
$\bar{A}\bar{B}CE$	$\bar{B}CDE$	$\bar{B}CE$

(c)

(c) Reduced comparisons

اشتقاق المجموع الكامل لدالة تبديلية بالاستعانة بخريطة كارنوه متغيرة المحتويات

علي محمد رشدي و حسين عبد الله آل يحيى

قسم الهندسة الكهربائية والحاسبية، جامعة الملك عبد العزيز، ص.ب ٩٠٢٧،

جدة ٢١٤١٣، المملكة العربية السعودية

(استلم في ١٩/٨/١٩٩٨م؛ وقبل للنشر في ٢/٨/٢٠٠٠م)

ملخص البحث. يُعرَّف المجموع الكامل لدالة تبديلية بأنه صيغة مجموع مضروبات تشكل مضروباتها جميع (ولا شيء سوى) الضامنات الأولية لهذه الدالة. وللمجموع الكامل تطبيقات مفيدة عديدة تشمل التبسيط والتصغير الأعظمي للدوال التبديلية، البرهنة على التكافؤ أو الاستقلال، حل المعادلات البولانية، وتحليل المخاطر العابرة. تقدم ورقة البحث هذه أسلوبين مبتكرين لاشتقاق المجموع الكامل بالاستعانة بخريطة كارنوه متغيرة المحتويات، وهي الخريطة التي تتمتع بمزايا تصويرية عديدة وبمضاعفة عدد المتغيرات التي تتعامل معها. وتقتصر الدراسة على اعتبار الدوال التبديلية كاملة التحديد، وذلك ببساطة لأن المجموع الكامل لدالة تبديلية غير كاملة التحديد ما هو إلا ذلك المجموع للدالة كاملة التحديد التي تمثل الحد الأعلى للدالة الأصلية. إن أسلوبنا الأول يستعمل الخريطة المذكورة للحصول على تعبير للدالة في صورة مضروب مجموعات يمكن تنفيذ الضرب فيه ومن ثم إيجاد المجموع الكامل وذلك بعد حذف الحدود القابلة للامتصاص. أما الأسلوب الثاني فيبدأ بالخريطة المذكورة وقد جعلت مدخلاتها في صورة مجموعات كاملة، ثم بعد طيات متعددة للخريطة ينتهي إلى المجموع الكامل المطلوب وذلك شريطة إنجاز الامتصاصات اللازمة عقب كل طية. وينتفع الأسلوبان كثيرا من استخدام مصفوفة ضرب مبتكرة تحد من عدد المقارنات بين الحدود التي تلزم لإنجاز الامتصاصات. إن من الممكن استغلال هذه المصفوفة لتحسين بعض الطرائق الجبرية الصرفة مثل طريقة تايسون لإيجاد المجموع الكامل. إلا أن هذه الطرائق الجبرية، حتى بعد تحسينها، تظل أضعف من طرائق الخريطة متغيرة المدخلات، ذلك لأن تلك الطرائق الأخيرة تجمع بين مزايا الخريطة والجبر معا. ويمكن استعمال الصورة المزاوجة لأسلوبي الخريطة المذكورين وذلك لإيجاد الكمية المزاوجة للمجموع الكامل وهي الكمية التي تعرف باسم المضروب الكامل.