On a New Boolean Function with Applications

Fabrizio Luccio, *Fellow*, *IEEE*, and Linda Pagli

Abstract—Consider a hypercube of 2^n points described by n Boolean variables and a subcube of 2^m points, $m \leq n$. As is well-known, the Boolean function with value 1 in the points of the subcube can be expressed as the product (AND) of $n-m$ variables. The standard synthesis of arbitrary functions exploits this property. We extend the concept of subcube to the more powerful *pseudocube*. The basic set is still composed of 2^m points, but has a more general form. The function with value 1 in a pseudocube, called *pseudoproduct*, is expressed as the AND of $n-m$ EXOR-factors, each containing at most $m+1$ variables. Subcubes are special cases of pseudocubes and their corresponding pseudoproducts reduce to standard products. An arbitrary Boolean function can be expressed as a sum of pseudoproducts (SPP). This expression is in general much shorter than the standard sum of products, as demonstrated on some known benchmarks. The logical network of an n-bit adder is designed in SPP, as a relevant example of application of this new technique. A class of symmetric functions is also defined, particularly suitable for SPP representation.

Index Terms—Pseudocube, pseudoproduct, EXOR-factor, Boolean function, algebraic expression, logical design.

1 Introduction

This is a contribution to Boolean functions and logical design, centered on the new concepts of *pseudocube* and *pseudoproduct*.

Consider a hypercube B^n described by the Boolean variables $x_0, ..., x_{n-1}$. The sets of 2^m points lying on subcubes $B^m \subseteq B^n$, $m \leq n$, are of paramount importance in Boolean algebra. In fact, the *characteristic function* of B^m (i.e., the function with value 1 in the points of the subcube, and value 0 elsewhere) can be expressed as the *product* (AND) of the $n-m$ variables having constant value in B^m, either in direct or complemented form. The synthesis of arbitrary functions exploits this property.

In this paper, we generalize the concept of subcube to a more powerful one. The basic set, still composed of 2^m points, is now called *pseudocube* of degree m. Subcubes are special cases of pseudocubes. The interesting fact is that pseudocubes have simple algebraic expressions using Exclusive OR (EXOR). To get an intuition of how pseudocubes look, we can put the following initial definition:

1. any single point is a pseudocube of degree 0;
2. any pair of points is a pseudocube of degree 1;
3. a subset P of 2^m points is a pseudocube of degree m if P can be divided into two disjoint pseudocubes P_1, P_2 of degree $m-1$, and there exists a subset α of variables such that P_2 can be derived from P_1 by complementing the variables of α in each of the points of P_1.

For example, consider the set P of 2^3 points in B^4 shown in the Karnaugh map of Fig. 1. This set can be divided into two subsets P_1, P_2 lying on the subcubes with $x_0 = 0$ and $x_0 = 1$, respectively, such that P_2 can be derived from P_1 by complementing the value of x_0 in each of the points of P_1. In turn, P_1 can be divided into two subsets P_{11}, P_{12}, corresponding to $x_1 = 0$ and $x_1 = 1$ (first and second column of the map), and P_{12} can be derived from P_{11} by complementing the values of x_1 and x_3. A similar decomposition clearly holds for P_2. Now, P_{11} and P_{12} are pseudocubes of degree 1 by Item 2 above, therefore, P_1 is a pseudocube of degree 2 by Item 3. Similarly, P_2 is a pseudocube of degree 2. Therefore, P is a pseudocube of degree 3. An algebraic expression for P can be directly extracted from the set of its points. As we shall see, this expression is $x_1 \oplus x_2 \oplus \bar{x}_3$.

Pseudocubes will be represented as Boolean matrices of 2^m rows (the points) and n columns (the variables). The definition given above will be substituted by another, based on the properties of such matrices. The paper is organized as follows:

Section 2 contains a discussion on Boolean matrices as a prerequisite to proving the results of the following sections. In particular, a matrix M is called *balanced* if its columns contain half 0s and half 1s, and this property repeats recursively on any submatrix obtained as the restriction of M to all the rows where an arbitrary variable has constant value. In Section 3, we formally define pseudocubes as sets of points whose matrix is balanced, and prove that pseudocubes exhibit several nice properties. For a given pseudocube P of degree m, we define the class $\Gamma(P)$ of the pseudocubes obtained from P by complementing a given arbitrary subset of variables and show that the elements of $\Gamma(P)$ are disjoint and tessellate B^n. Furthermore, the union of any two pseudocubes of the same class Γ is a pseudocube, and the intersection of two arbitrary pseudocubes is either empty or is a pseudocube.

In Section 4, we introduce the *pseudoproduct* of degree m as the characteristic Boolean function of a pseudocube P of the same degree. Pseudoproducts have a canonical algebraic expression indicated as CEX(P) consisting of the AND of $n-m$ *EXOR-factors*, each containing at most $m+1$ literals. CEX(P) can be built in linear time from the points of P. If P is a subcube, the EXOR-factors reduce to single variables and CEX(P) reduces to the standard product

• *The authors are with the Dipartimento di Informatica, Università di Pisa, Corso Italia, 40, I-56125 Pisa, Italy. E-mail: {luccio, pagli}@di.unipi.it.*

Manuscript received 29 Sept. 1997; revised 28 May 1998.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 105725.

0018-9340/99/$10.00 © 1999 IEEE

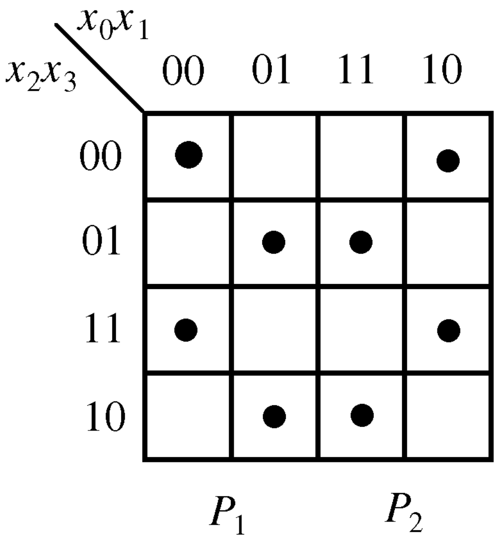


Fig. 1. A pseudocube P of degree 3 (black dots) composed of two pseudocubes P_1, P_2 of degree 2.

expression. In the example of Fig. 1, we have $n = 4$ and $m = 3$ and the canonical expression CEX(P) = $x_1 \oplus x_2 \oplus \bar{x}_3$ consists of one EXOR-factor. Given two pseudoproducts P_1, P_2 belonging to the same class Γ, we can construct CEX($P_1 \cup P_2$) from CEX(P_1) and CEX(P_2) in linear time.

We then show how an arbitrary Boolean function f can be expressed as a sum (OR) of pseudoproducts and compare this new form (SPP) with the standard sum of products (SP) and with binary decision diagrams (BDD). In fact, SP is a particular case of SPP, but the latter is much shorter in general. As relevant examples of application of this new technique, we design the logical network of an n-bit adder (Section 5), and derive the SPP expressions for some known benchmark Boolean functions (Section 6). We finally define a class of symmetric functions insensitive to the complementation of subsets of variables (Section 7) and show that they are particularly suitable for SPP representation.

A	c_0	c_1	c_2	c_3	c_4	c_5
r_0	0	1	0	1	0	1
r_1	0	1	0	1	1	0
r_2	0	1	1	0	0	1
r_3	0	1	1	0	1	0
r_4	1	1	0	0	0	0
r_5	1	1	0	0	1	1
r_6	1	1	1	1	0	0
r_7	1	1	1	1	1	1

Fig. 2. A canonical matrix for $m = 3$, $n = 6$, with canonical columns $\mathbf{c}_0, \mathbf{c}_2, \mathbf{c}_4$.

To the best of our knowledge, this approach is completely new, except for some preliminary results that we have presented at a conference [10]. We regard this work as basic theory, mainly directed to the comprehension and algebraic representation of Boolean functions.

2 Some Properties of Boolean Matrices

Boolean matrix theory is well-established (e.g., see [2]). We introduce here some new definitions and properties of these matrices, with the sole purpose of proving the results reported in the following. The proofs of the formal statements of this section have been moved to an appendix, since they are not crucial to follow the development of our theory.

Given a binary vector $\mathbf{u}$, its *complement* $\bar{\mathbf{u}}$ is the elementwise Boolean complementation of $\mathbf{u}$. The symbol $\hat{\mathbf{u}}$ denotes $\mathbf{u}$ or $\bar{\mathbf{u}}$. $\mathbf{0}$ and $\mathbf{1}$ denote vectors of all 0s, or all 1s, respectively. In the following, we shall always refer to binary vectors and binary matrices.

Definition 1. *A vector* $\mathbf{u}$ *of* 2^m *elements,* $m \geq 0$, *is* normal *if*

1. $m = 0$ *or*
2. $m > 0$

and $\mathbf{u} = \mathbf{v}\hat{\mathbf{v}}$ *with* $\mathbf{v}$ *(hence,* $\hat{\mathbf{v}}$*) normal. (*$\mathbf{v}\hat{\mathbf{v}}$ *is the concatenation of* $\mathbf{v}$ *and* $\hat{\mathbf{v}}$*).*

Definition 2. *A vector* $\mathbf{u}$ *is* constant *if* $\mathbf{u} = \mathbf{0}$ *or* $\mathbf{u} = \mathbf{1}$*; otherwise* $\mathbf{u}$ *is* divided. $\mathbf{u}$ *is* balanced *if it is constant or half of its elements are* 0 *and half are* 1. *Two balanced vectors* $\mathbf{u}$, $\mathbf{v}$ *are* concordant *if* $\mathbf{u} = \mathbf{v}$ *or* $\mathbf{u} = \bar{\mathbf{v}}$*; otherwise* $\mathbf{u}$, $\mathbf{v}$ *are* discordant.

Clearly, any vector of one or two elements is normal and any normal vector is balanced. For example, $00, 0101, 01101001$ are normal (and balanced) vectors. By induction on m we can easily prove:

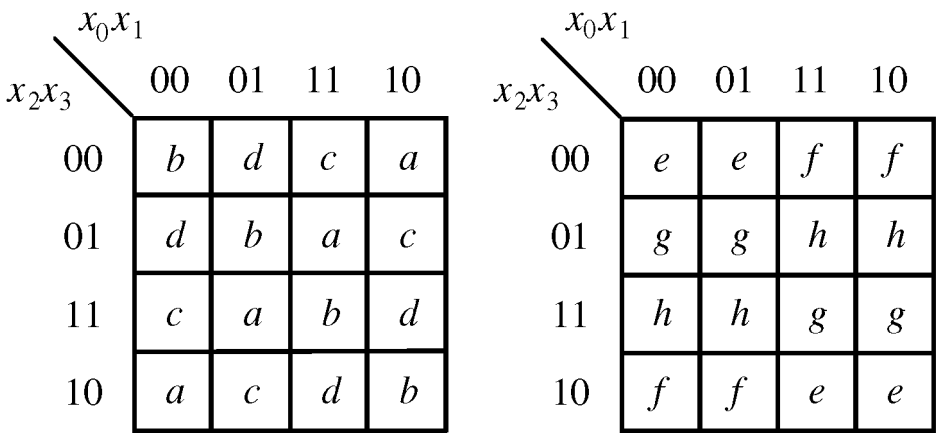
Proposition 1. *Let a vector* $\mathbf{u}$ *of* 2^m *elements be normal. Then,* $\mathbf{u} = \mathbf{v}_0 \ldots \mathbf{v}_{2^{m-k}-1}, 0 \leq k \leq m$, *where each* $\mathbf{v}_\mathrm{i}$ *consists of* 2^k *equal elements (i.e.,* $\mathbf{v}_\mathrm{i} = \mathbf{0}$ *or* $\mathbf{v}_\mathrm{i} = \mathbf{1}$*). For* $k \leq m - 1$*, we also have* $\mathbf{v}_0 = \bar{\mathbf{v}}_1$.

Definition 3. *A normal vector* $\mathbf{u} = \mathbf{v}_0 \ldots \mathbf{v}_{2^{m-k}-1}$ *is* k-normal, *with* k *defined as in Proposition 1. A k-normal vector is* k-canonical *if* $\mathbf{v}_i = \mathbf{0}$ *for* i *even and* $\mathbf{v}_i = \mathbf{1}$ *for* i *odd.*

For example, the vector 1100001100111100 is 1-normal, with $m = 4$. We have $\mathbf{v}_0 = 11$, $\mathbf{v}_1 = 00$, $\mathbf{v}_2 = 00$, and so on. The vector 0000111100001111 is 2-canonical. We now extend the above concepts from vectors to matrices. Unless differently specified, a matrix will always have 2^m rows $\mathbf{r}_0, \ldots, \mathbf{r}_{2^m-1}$ and n columns $\mathbf{c}_0, \ldots, \mathbf{c}_{n-1}$, $n \geq 1$ and $0 \leq m \leq n$.

Definition 4. *A matrix* M *is* normal *if* $\mathbf{r}_i \neq \mathbf{r}_j$ *for* $i \neq j$ *and all the columns are normal. A normal matrix is* canonical *if its rows, interpreted as binary numbers, are arranged in increasing order.*

The matrix shown in Fig. 2 is canonical. An important property of canonical matrices is the following (the proof is in Appendix A).



$x_2x_3 \backslash x_0x_1$	00	01	11	10
00	*b*	*d*	*c*	*a*
01	*d*	*b*	*a*	*c*
11	*c*	*a*	*b*	*d*
10	*a*	*c*	*d*	*b*

$x_2x_3 \backslash x_0x_1$	00	01	11	10
00	*e*	*e*	*f*	*f*
01	*g*	*g*	*h*	*h*
11	*h*	*h*	*g*	*g*
10	*f*	*f*	*e*	*e*

Fig. 3. Eight pseudocubes in B^4: the points of a pseudocube are marked with the same letter. Each map contains a class Γ.

Proposition 2. *A canonical matrix M contains m columns $\mathbf{c}_{i_0}, \ldots, \mathbf{c}_{i_{m-1}}$ of increasing indices, such that $\mathbf{c}_{i_j}$ is $(m-j-1)$-canonical for $0 \leq j \leq m-1$.*

In a canonical matrix M, the columns $\mathbf{c}_{i_0}, \ldots, \mathbf{c}_{i_{m-1}}$ of Proposition 2 are called *canonical columns* of M (if several $(m-j-1)$-canonical columns exist, $0 \leq j \leq m-1$, $\mathbf{c}_{i_j}$ is the one of smallest index among them). The other columns are the *noncanonical* ones. See Fig. 2. We now extend the concept of balanced vectors to balanced matrices.

Definition 5. *The restriction of a matrix M, composed of the rows whose elements in column $\mathbf{c}_i$ are equal to 1 (respectively, equal to 0) is denoted by M_i (respectively, $M_{\bar{i}}$). M is* balanced *if all its rows are pairwise different, all its columns are balanced, and each M_i (hence, $M_{\bar{i}}$) is balanced, $0 \leq i \leq n-1$.*

The notation of Definition 5 is extended by writing $M_{\hat{i}}$ for M_i or $M_{\bar{i}}$. Further restrictions $((M_{\hat{i}})_{\hat{j}})_{\hat{k}}\ldots$ are indicated as $M_{\hat{i},\hat{j},\hat{k}\ldots}$.

For a given matrix, the property of being balanced is obviously not affected by row permutations. In addition, particular row permutations preserve normality. The following Propositions 3 and 4 show that normal and balanced matrices are the same up to row permutations (the proofs are in Appendix A).

Proposition 3. *Any normal (in particular, canonical) matrix M is balanced.*

Proposition 4. *Any balanced matrix M can be transformed into the corresponding canonical matrix by rearranging the rows, interpreted as binary numbers, in increasing order.*

From a computational point of view, we can test whether a matrix M is balanced in $\Theta(2^m \times n)$ time (i.e., in time linear with the size of the input matrix). The algorithm is as follows:

Algorithm 1(check balancing of a matrix M)
1. check that all the columns of M are balanced;
2. sort the rows of M as they were binary numbers;
3. check that all the columns thus obtained are normal.

Phase 1 can be trivially executed in $\Theta(2^m \times n)$. Phase 2 can be executed in equal time using Radix-Sort. In Phase 3, the control of normality of each column $\mathbf{c}_h$ is done in $\Theta(2^m)$ time, recursively checking the normality of $\mathbf{c}_h[0 : 2^{m-1}-1]$, and then checking that $\mathbf{c}_h[0 : 2^{m-1}-1] = \mathbf{c}_h[2^{m-1} : 2^m - 1]$, or $\mathbf{c}_h[0 : 2^{m-1}-1] = \bar{\mathbf{c}}_h[2^{m-1} : 2^m - 1]$. If Phase 1 or Phase 3 fail, the algorithm stops declaring M nonbalanced.

3 Pseudocubes

Consider a set $S = \{s_0, \ldots, s_{2^m-1}\}$ of 2^m points of B^n, $m \leq n$. We can obviously represent S as a Boolean matrix, with the rows associated to the points $s_0, \ldots, s_{2^m-1}$ and the columns associated to the variables $x_0, \ldots, x_{n-1}$. We shall indifferently refer to points or rows, and to variables or columns. We formally pose:

Definition 6. *A* pseudocube *of degree m is a set of 2^m points whose matrix is balanced or, equivalently, it is canonical up to a permutation of rows (Propositions 3 and 4).*

Note that any set of one or two points is a pseudocube. The reader may check that the sets of points marked in the two Karnaugh maps of Fig. 3 are pseudocubes. For example, the set marked with b has matrix

$$\begin{array}{cccc} \mathbf{x}_0 & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{array}$$

which is balanced and becomes canonical, permuting the third and fourth rows. Algorithm 1 can be directly applied to check if a set $S \subseteq B^n$ is a pseudocube.

A cube $B^{n-k} \subseteq B^n$, $0 \leq k \leq n$, is a subset of 2^{n-k} points, whose matrix has k constant columns and $n-k$ divided columns. Sorting the rows as increasing binary numbers, we obtain a canonical matrix. That is, a cube is a special case of pseudocube where only the canonical columns are divided. We shall see that pseudocubes maintain some basic properties generally ascribed to cubes.

Definition 7. *For a point $s \in B^n$ and a subset of variables α, the* transformed point *$\alpha(s)$ is obtained from s by complementing the variables in α. For a set of points S, the* transformed set *$\alpha(S)$ is the set $\{\alpha(s) : s \in S\}$.*

Proposition 5. *Let P be a pseudocube of degree m and α be a subset of variables. Then, $\alpha(P)$ is a pseudocube of degree m, with $\alpha(P) = P$ or $\alpha(P) \cap P = \emptyset$.*

Proof Without loss of generality, represent P by the canonical matrix M and assume that the variables $x_0, \ldots, x_{m-1}$ and $x_m, \ldots, x_{n-1}$, respectively, indicate the canonical and the noncanonical columns of M. Denote by C and NC the submatrices formed by such columns. First, consider the effect of complementing variables in NC, that is, let $\alpha \subseteq \{x_m, \ldots, x_{n-1}\}$. The matrix M' thus obtained, associated with $\alpha(P)$, is still canonical, with the canonical part $C' = C$ and a remaining part NC'. In fact, all the rows in M' are distinct, because the subrows in C' are distinct; and all the columns in NC' are normal because variable complementation preserves normality. Furthermore, the rows of M' are sorted in increasing order since ordering is induced by $C' = C$. Then, $\alpha(P)$ is a pseudocube of degree m. Moreover, for any two points $u \in P$, $v \in \alpha(P)$, either some of the entries in C, C' are different or all these entries are equal, but some entries in NC, NC' are different by the effect of α. Then, $\alpha(P) \cap P = \emptyset$. Consider now complementing a variable x_i in C. This has the same effect as complementing all the variables x_j in NC such that $(x_j)_i = \overline{(x_j)_{\bar{i}}}$, where $(x_j)_k$ denotes the restriction of the column x_j to M_k. If no such variable x_j exists, then complementing x_i amounts to a permutation of the rows of M, bringing P onto itself. Therefore, for $\alpha \subseteq \{x_0, \ldots, x_{m-1}\}$ there is a (possibly empty) subset $\beta \subseteq \{x_m, \ldots, x_{n-1}\}$ such that $\alpha(P) = \beta(P)$, and the previous analysis shows that $\alpha(P) \cap P = \emptyset$ for $\beta(P) \neq \emptyset$, $\alpha(P) = P$ for $\beta(P) = \emptyset$. Finally, if α has variables in C and in NC, the effect of the two subsets can be analyzed separately. The previous analysis again shows that $\alpha(P) \cap P = \emptyset$, or $\alpha(P) = P$. □

The proof of Proposition 5 contains the seeds of a more general result. In fact, distinct subsets of variables α, β corresponding to noncanonical columns generate disjoint pseudocubes $\alpha(P)$, $\beta(P)$. Since there are 2^{n-m} such subsets, there are at least 2^{n-m} disjoint pseudocubes generated from P. Moreover, Proposition 5 also states that no two pseudocubes generated from P may partially overlap. We then have:

Theorem 1. *For any pseudocube P of degree m, there are exactly 2^{n-m} disjoint pseudocubes (including P) of degree m obtained from P by complementation of variables. These pseudocubes are exactly the ones obtained by complementing the noncanonical variables of the canonical matrix in all possible ways. These pseudocubes form a class, denoted by $\Gamma(P)$, that tessellates B^n.*

Note that $\Gamma(P) = \Gamma(\alpha(P))$ for any α. Two families Γ in B^4 are shown in Fig. 3. x_0 and x_1 are the canonical variables for both families. In the first map, the pseudocube marked with b derives from the one marked with a by complementing the noncanonical variable x_2, or the canonical variable x_0. Then, complementing x_0 and x_2 brings the pseudocube onto itself. In the second map, the pseudocube marked with f derives from the one marked with g by complementing x_2 and x_3. Complementing x_1 brings the pseudocube onto itself. A relevant result is the following:

Theorem 2. *Let P, Q be pseudocubes of degree m, $P \cap Q = \emptyset$. $P \cup Q$ is a pseudocube (of degree $m+1$) if and only if $Q \in \Gamma(P)$.*

Proof. As in the proof of Proposition 5, represent P by a canonical matrix M with canonical columns associated to $x_0, \ldots, x_{m-1}$. Also, let $R = P \cup Q$.

If part. Let $Q = \alpha(P)$, with $\alpha \subseteq \{x_m, \ldots, x_{n-1}\}$: We prove that R is a pseudocube. Q can be represented by a canonical matrix N, obtained from M complementing the columns of α. Then, R can be represented by the matrix Z obtained attaching N to M. The columns of Z belonging to α have the structure $\mathbf{v}\bar{\mathbf{v}}$, and the other columns have the structure $\mathbf{vv}$, with $\mathbf{v}$ normal since it comes from M. That is, all the columns of Z are normal, hence, R is a pseudocube by Proposition 3.

Only if part. Let R be a pseudocube: We prove that $Q \in \Gamma(P)$. Choose the canonical matrix N to represent Q, and attach N to M to form a matrix Z representing R. Since the columns of Z must be balanced, any variable x_i divided in M (in particular, $x_0, \ldots, x_{m-1}$) must be also divided in N. Take any pair of variables $x_i, x_j \in \{x_0, \ldots, x_{m-1}\}$ and consider Z_i (the restriction of Z to the rows with $x_i = 1$). We have that x_j is divided in M_i and is balanced in Z_i, therefore, it is divided in N_i. That is, x_i and x_j are discordant in N. We conclude that $x_0, \ldots, x_{m-1}$ are divided and pairwise discordant in N, and form the canonical part of N. The proof now proceeds by induction on m. The basis $m = 1$ holds as a consequence of the fact that any balanced vector of four elements is normal. Let the assertion hold for $m - 1$. Note that M_0, $M_{\bar{0}}$, N_0, $N_{\bar{0}}$ correspond to pseudocubes of degree $m - 1$ and that $M_0 \cup M_{\bar{0}}$, $N_0 \cup N_{\bar{0}}$, $M_0 \cup N_0 = Z_0$ correspond to pseudocubes. Then, by induction, M_0, $M_{\bar{0}}$, N_0, $N_{\bar{0}}$ are all in the same class Γ. Therefore, each column of Z has the form $\hat{\mathbf{u}}\hat{\mathbf{u}}\hat{\mathbf{u}}\hat{\mathbf{u}}$. Consider now the submatrix $W = (..(Z_{\bar{1}})_{\bar{2}}..)_{\overline{m-1}}$ composed of the four starting rows of $M_{\bar{0}}$, M_0, $N_{\bar{0}}$, N_0. In each column, W contains the starting elements of the four vectors $\hat{\mathbf{u}}$ above. The values of these elements determine if the corresponding $\hat{\mathbf{u}}$ has the actual configuration $\mathbf{u}$ or $\bar{\mathbf{u}}$. Being a restriction of Z, W is the matrix of a pseudocube of degree 2, therefore, its columns are balanced (hence, normal). It is immediate to verify that all the corresponding arrangements of $\hat{\mathbf{u}}\hat{\mathbf{u}}\hat{\mathbf{u}}\hat{\mathbf{u}}$ can be rewritten as $\mathbf{w}\hat{\mathbf{w}}$. That is, P and Q belong to the same class Γ. □

Let us now see how a pseudocube of degree m can be extended to one of degree $m + 1$.

Definition 8. *Given a pseudocube P of degree $m \geq 1$ and a point $u \notin P$, the* extension *$EXT(P, u)$ is a set of points such that $u \in EXT(P, u)$, and $P \cup EXT(P, u)$ is a pseudocube of degree $m + 1$.*

Proposition 6. *For any pseudocube P of degree $m \geq 1$ and any point $u \notin P$, $EXT(P, u)$ exists and is uniquely determined, and $EXT(P, u) \in \Gamma(P)$.*

Proof. Take an arbitrary point $v \in P$. Consider the subset of variables α such that $\alpha(v) = u$ and denote $\alpha(P) = P'$. By Proposition 5 and Theorem 2, we have that P' is a pseudocube of degree m and $P \cup P'$ is a pseudocube of degree $m + 1$. That is, $EXT(P, u) = P'$ and $EXT(P, u) \in \Gamma(P)$. To prove that $EXT(P, u)$ is unique, assume by contradiction that another completion P'' exists, with $P' \neq P''$, and $u \in (P' \cap P'')$. By assumption, $P \cup P''$ is a pseudocube, hence, $P'' \in \Gamma(P)$ by Theorem 2, which is impossible because also $P' \in \Gamma(P)$ and $P' \cap P'' \neq \emptyset$. □

From the proof of Proposition 6, we can make the following:

Observation 1. *Given three points u, v, w, there exists exactly one point z such that $\{u, v, w, z\}$ is a pseudocube.*

In this simple case there is an immediate construction for z since the value of each variable in z can be determined by balancing the corresponding column. For example:

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	...
u	0	0	0	1	...
v	1	0	1	0	...
w	1	0	0	1	...
z	0	0	1	0	...

The concept of extension is needed to prove some further results. We have:

Proposition 7. *Let R be a pseudocube of degree r and P be a pseudocube of degree p, with $r \geq 2$, $1 \leq p < r$, and $P \subset R$. For any point $u \in R - P$ we have $EXT(P, u) \subset R$.*

Proof. 1. We first prove the proposition for $p = 1$ by induction on r. Let $P = \{v, w\}$.

Basis. $r = 2$. Immediate from Observation 1.

Inductive step. $r \geq 2$. Assuming that the hypothesis holds for r, we prove that it holds for $r + 1$. Represent the matrix M of R in normal form, and partition M from the top into four consecutive normal submatrices A, B, C, D, each composed of 2^{r-2} rows. AB and CD will denote the top and bottom halves of M.

a. If v, w, u are all in AB, or all in CD, the proposition holds by the inductive hypothesis;

b. Let v, w be in AB and u be in CD (all other cases are similar). Use the permutations *PERM1*, *PERM2*, *PERM3* given in Appendix A that preserve the normality of M. It is easy to see that, with some permutations where we set $k = m - 3$, v and w can be brought into A, leaving u in CD. Then, with one permutation where we set $k = m - 2$, u is brought into B, leaving v, w in A. We now have v, w, u in AB, and the inductive hypothesis applies.

2. We now prove the proposition for any $p > 1$. Take an arbitrary point $v \in P$. Consider the subset of variables α such that $\alpha(v) = u$. We have $EXT(P, u) = \alpha(P)$. Take an arbitrary point $z \neq u$ in $EXT(P, u)$. We prove that $z \in R$, hence, $EXT(P, u) \subset R$. Consider the point w such that $\alpha(w) = z$. Note that $w \in P$. Since $\{v, w\}$ is a pseudocube, and $\alpha\{v, w\} = \{u, z\} \in \Gamma(\{v, w\})$, then $\{v, w, u, z\}$ is a pseudocube by Theorem 2. Therefore, $\{z\} = EXT(\{v, w\}, u)$, that is $z \in R$ by point 1. □

The preceding results show how a pseudocube of degree m, and a point external to it, determine a pseudocube of degree $m + 1$ containing the two. Indeed, a stronger result holds:

Theorem 3. *Given k points $s_0, \ldots, s_{k-1}$, the pseudocube T of minimal degree t containing $s_0, \ldots, s_{k-1}$ is uniquely determined, and we have $\lceil \log_2 k \rceil \leq t \leq k - 1$.*

Proof. The lower bound is trivial since a pseudocube of degree $< \lceil \log_2 k \rceil$ contains less than k points. To prove the upper bound, consider the pseudocubes S^i, $0 \leq i \leq k - 1$, built as follows:

$$S^0 = \{s_0\}$$

$$S^1 = \{s_0, s_1\}$$

$$S^i = \begin{cases} S^{i-1} & \text{if} \quad s_i \in S^{i-1} \\ S^{i-1} \cup EXT(S^{i-1}, s_i) & \text{if} \quad s_i \notin S^{i-1}. \end{cases}$$

Note that all the S^i are uniquely determined by Proposition 6. We now prove by induction on i that, for any pseudocube $S \ni s_0, \ldots, s_i$, we have $S^i \subseteq S$.

Basis. $i = 0, 1$. Trivial.

Inductive step. $2 \leq i \leq k - 1$. Assuming that the hypothesis holds for $i - 1$, we prove that it holds for i. For any pseudocube $S \ni s_0, \ldots, s_i$ we trivially have $S \ni s_0, \ldots, s_{i-1}$, hence, $S^{i-1} \subseteq S$ by induction. If $s_i \in S^{i-1}$, we have $S^i = S^{i-1}$ and the inductive hypothesis obviously holds for i. If $s_i \notin S^{i-1}$, we have $S^i = S^{i-1} \cup EXT(S^{i-1}, s_i)$ and the hypothesis holds by Proposition 7. As a consequence, $S^{k-1} \subseteq$ any subset containing $s_0, \ldots, s_{k-1}$, that is, $T = S^{k-1}$. By construction, this set is uniquely determined and we have $t =$ (degree of $S^{k-1} \leq k - 1$), where equality applies if $s_i \notin S^{i-1}$ in the construction of all S^i. □

We finally study how pseudocubes intersect.

Theorem 4. *Let P and R be pseudocubes of degree p, r, respectively. Then, either $P \cap R = \emptyset$ or $P \cap R$ is a pseudocube of degree $\geq p + r - n$.*

Proof. The case $P \cap R = \emptyset$ is trivial. If $P \cap R = \{s_0, \ldots, s_{k-1}\}$, $k \geq 1$, then both P and R must contain the minimum pseudocube T containing $s_0, \ldots, s_{k-1}$, as shown in the proof of Theorem 3. This implies that $\{s_0, \ldots, s_{k-1}\} = T$, that is, $P \cap R$ is a pseudocube. We now prove a lower bound for $|T|$. Consider the set $\{R^1, \ldots, R^h\}$ of the elements of $\Gamma(R)$ that have nonempty intersection with P (R is one of these elements). $P \cap R^i$, $1 \leq i \leq h$, is a pseudocube as proven above, with $|P \cap R^i| = 2^{r_i}$ for proper r_i. For any pair R^j, R^k, we have that $R^j \cup R^k$ is a pseudocube by Theorem 2, therefore, $Q = P \cap (R^j \cup R^k)$ is a pseudocube and $|Q| = 2^q$ for proper q. We then have $2^{r_j} + 2^{r_k} = 2^q$, that implies $r_j = r_k$. Then, $|P \cap R^1| = \ldots = |P \cap R^h| = 2^a$, with $2^p/h = 2^a$. The value of a is minimum if all the elements of $\Gamma(R)$ have nonempty intersection with P, that is, $h = 2^{n-r}$. We then have $2^p/2^{n-r} \leq 2^a$, that is, $a \geq p + r - n$. □

At the beginning of this section, we saw that any cube $B^k \subseteq B^n$, $0 \leq k \leq n$, is a pseudocube. Let us now discuss how the main properties of pseudocubes are interpreted for cubes. Theorem 1 applies directly to cubes, implicitly defining the tessellation $\Gamma(C)$ for a given cube C, whose members are cubes. Theorem 2 holds only in its "only if" part. That is, if C and D are cubes of degree m, $C \cap D = \emptyset$, the fact that $C \cup D$ is a cube implies $D \in \Gamma(C)$, but the union of two cubes of the same family Γ is not necessarily a cube (but, obviously, is a pseudocube). For a cube C of degree $m \geq 1$ and a point $u \notin C$, we can sharpen Definition 8 by letting $EXT(C,u)$ be a set of points containing u such that $C \cup EXT(C,u)$ is a cube of degree $m+1$. The existence of $EXT(C,u)$ is not guaranteed in this case, however, if $EXT(C,u)$ exists, then it is uniquely determined and $EXT(C,u) \in \Gamma(C)$. Theorem 4 holds unchanged for cubes. Finally, note that no useful adaptation can be done for Theorem 3 because the smallest cube containing two given points can be as large as B^n. This occurs, for example, for the points $x_1, x_2, \ldots, x_n = (0, 0, \ldots, 0)$ and $x_1, x_2, \ldots, x_n = (1, 1, \ldots, 1)$.

4 Pseudoproducts

Our first concern is now to derive a concise algebraic expression of the characteristic function of a pseudocube. We use the Boolean operators OR (symbol +), AND (symbol $\cdot$), and EXOR (symbol $\oplus$). Since these operators are commutative and associative, they are naturally applied to any number of variables. In particular, we define an *EXOR-factor* as a single variable or as a string of variables connected by EXOR in any order. The following lemma contains two equalities that can be easily proven by induction:

Lemma 1. *Let $y_1, ..., y_k$ be Boolean variables, $k \geq 2$. We have:*

1. $y_1 \oplus y_2 \oplus \ldots \oplus \bar{y}_k = \overline{(y_1 \oplus y_2 \oplus \ldots \oplus y_k)}$;
2. $$y_1 \cdot y_2 \cdot \ldots \cdot y_k + \overline{y}_1 \cdot \overline{y}_2 \cdot \ldots \cdot \overline{y}_k = (\overline{y}_1 \oplus y_2) \cdot (\overline{y}_1 \oplus y_3) \cdot \ldots \cdot (\overline{y}_1 \oplus y_k).$$

Lemma 1.1, combined with the commutativity of EXOR, shows that complementing any variable in an EXOR-factor amounts to complementing the whole factor. This also immediately implies:

Lemma 2. *$\hat{y}_1 \oplus \hat{y}_2 \oplus ... \oplus \hat{y}_k = y_1 \oplus y_2 \oplus ... \oplus y_k$, if the number of complementations in the left-hand side is even; $\hat{y}_1 \oplus \hat{y}_2 \oplus ... \oplus \hat{y}_k = y_1 \oplus y_2 \oplus ... \oplus \bar{y}_k$, if the number of complementations in the left-hand side is odd.*

We now pose:

Definition 9. *Let P be a pseudocube of degree m in B^n; let M be the canonical matrix of P; and let $x_{p_0}, \ldots, x_{p_{m-1}}$ and $x_{p_m}, \ldots, x_{p_{n-1}}$ be the canonical and noncanonical variables, respectively, with these two sets ordered for increasing values of the indices. The* canonical expression *associated with P, denoted by $CEX(P)$, is given by $f_0 \cdot f_1 \cdot \ldots \cdot f_{n-m-1}$, where each f_i, $0 \leq i \leq n-m-1$, is an EXOR-factor containing the following variables:*

1. *the canonical variables x_{p_j}, $0 \leq j \leq m-1$ such that $M[0, p_{m+i}] \neq M[2^{m-j-1}, p_{m+i}]$; these variables are ordered for increasing indexes;*
2. *the noncanonical variable $x_{p_{m+i}}$ if $M[0, p_{m+i}] = 1$ or $\bar{x}_{p_{m+i}}$ if $M[0, p_{m+i}] = 0$.*

Note that $CEX(P)$ contains an EXOR-factor for each noncanonical variable. For the pseudocube P_1 whose matrix M^1 is shown in Fig. 4 we have:

$$\begin{aligned} CEX(P_1) = &(x_1 \oplus x_2) \cdot (x_0 \oplus x_1 \oplus x_3) \cdot \\ &(x_0 \oplus x_1 \oplus x_4 \oplus \bar{x}_5) \cdot (x_0 \oplus \bar{x}_6). \end{aligned} \tag{1}$$

Note that the canonical matrix of a pseudocube P can be built in $\Theta(2^m \times n)$ (linear) time from the set of points of P by applying Algorithm 1 that, in fact, builds the canonical matrix M. Once this matrix is known, $CEX(P)$ can be built in $\Theta(m \times n)$ time by first determining the canonical variables of M and, then, applying Definition 9. The length of $CEX(P)$ is $O((n-m) \times m)$.

In Boolean algebra, the characteristic function of a subcube $B^m \subseteq B^n$ is called a *product* and is expressed as the AND of $n-m$ variables in direct or complemented form. Similarly, we call *pseudoproduct of degree m* the characteristic function of a pseudocube of degree m. This function can be expressed as the AND of $n-m$ EXOR-factors. In fact, we have:

Theorem 5. *For a pseudocube P, the pseudoproduct can be expressed as $CEX(P)$.*

Proof By induction on m.

Basis.$m = 0$. P consists of one point. All the variables are noncanonical. $CEX(P)$ reduces to a product (minterm) $\hat{x}_{p_0} \cdot \ldots \cdot \hat{x}_{p_{n-1}}$, built according to Definition 9.2.

Inductive step. $m > 1$. Denote by P_1, P_2 the pseudocubes of degree $m-1$ whose canonical matrices are $M_{\bar{p}_0}$ and M_{p_0}, respectively. Inductively assume that $CEX(P_1)$ and $CEX(P_2)$ are valid expressions for P_1, P_2. Note that the canonical variables of P_1 and P_2 are $x_{p_1}, \ldots, x_{p_{m-1}}$, while x_{p_0} is constant in P_1 and P_2, hence, is a noncanonical variable for the two pseudocubes. Applying Definition 9.2 to the column p_0 of $M_{\bar{p}_0}$ and M_{p_0}, we have $CEX(P_1) = \bar{x}_{p_0} \cdot f^1{}_1 \cdot \ldots \cdot f^1{}_{n-m-1}$ and $CEX(P_2) = x_{p_0} \cdot f^2{}_1 \cdot \ldots \cdot f^2{}_{n-m-1}$. To compare $f^1{}_i$ with $f^2{}_i$, $1 \leq i \leq n-m-1$, note that the column p_{m+i} of M is normal. Then, by Definition 9.1, $f^1{}_i$ and $f^2{}_i$ contain the same canonical variables from among $x_{p_1}, \ldots, x_{p_{m-1}}$. Moreover, depending on the values of $M[0, p_{m+i}]$ and $M[2^{m-1}, p_{m+i}]$, both $f^1{}_i$ and $f^2{}_i$ contain $x_{p_{m+i}}$, or $\bar{x}_{p_{m+i}}$, or one contains $x_{p_{m+i}}$ and the other contains $\bar{x}_{p_{m+i}}$. Therefore, we have $f^1{}_i = f^2{}_i$, or $f^1{}_i = \bar{f}^2{}_i$ by Lemma 1.1. Letting $f^1{}_{i_1} = f^2{}_{i_1}, ..., f^1{}_{i_k} = f^2{}_{i_k}$ and $f^1{}_{i_{k+1}} = \bar{f}^2{}_{i_{k+1}}, ..., f^1{}_{i_{n-m-1}} = \bar{f}^2{}_{i_{n-m-1}}$, we have, by Lemma 1.2:

$$\begin{array}{ccccccc}
x_0 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\
0 & 0 & \underline{1} & \underline{1} & 0 & \underline{0} & \underline{0} \\
0 & 0 & \underline{1} & \underline{1} & 1 & \underline{1} & \underline{0} \\
0 & 1 & \underline{0} & \underline{0} & 0 & \underline{1} & \underline{0} \\
0 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & \underline{1} & \underline{0} & 0 & \underline{1} & \underline{1} \\
1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 1
\end{array}
\qquad
\begin{array}{ccccccc}
x_0 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 1 & 0
\end{array}$$

$$M^1 \qquad\qquad\qquad M^2$$

Fig. 4. The canonical matrices M^1 and M^2 of two pseudocubes P_1 and P_2 of degree 3 in B^7. For both matrices, x_0, x_1, x_4 and x_2, x_3, x_5, x_6 are the canonical and noncanonical variables, respectively. The underlined entries in M^1 are the ones to be checked in Definition 9. We have $P_2 = \alpha(P_1)$ with $\alpha = \{x_2, x_3, x_6\}$.

$$\begin{aligned}
CEX(P_1) + CEX(P_2) &= f^1{}_{i_1} \cdot \ldots \cdot f^1{}_{i_k} \\
&\quad \cdot (\bar{x}_{p_0} \cdot f^1{}_{i_{k+1}} \cdot \ldots \cdot f^1{}_{i_{n-m-1}} + x_{p_0} \\
&\quad \cdot \bar{f}^1{}_{i_{k+1}} \cdot \ldots \cdot \bar{f}^1{}_{i_{n-m-1}}) \\
&= f^1{}_{i_1} \cdot \ldots \cdot f^1{}_{i_k} \cdot (x_{p_0} \oplus \bar{f}^1{}_{i_{k+1}}) \cdot \ldots \\
&\quad \cdot (x_{p_0} \oplus \bar{f}^1{}_{i_{n-m-1}})
\end{aligned}$$

and this expression coincides with CEX(P). Since $CEX(P_1) + CEX(P_2)$ is an expression for the characteristic function of P, the theorem immediately follows. □

Since a subcube B^m is a particular case of pseudocube of degree m, a product is a particular case of pseudoproduct. The AND expression of a product is a limit case of CEX expression. In fact, the $n-m$ variables appearing in the product have constant values in the corresponding subcube and constitute the noncanonical variables of the pseudocube. In general, the CEX expression of a pseudoproduct of degree m contains $n-m$ EXOR-factors (Definition 9), hence, at most $(n-m)(m+1)$ literals.

Unlike for the limit case of a product, the CEX expression of a pseudoproduct depends, in general, on the ordering of the variables because the canonical variables of the pseudocube are chosen according to the (conventional) values of their indices. For the matrix M^1 of Fig. 4, the new ordering $x_0x_4x_5x_1x_2x_3x_6$ would imply that x_0, x_4, x_5 are the canonical variables, yielding the new CEX expression:

$$\begin{aligned}
CEX(P_1) = (x_0 \oplus x_4 \oplus x_5 \oplus \bar{x}_1) \cdot (x_0 \oplus x_4 \oplus x_5 \oplus x_2) \\
\cdot (x_4 \oplus x_5 \oplus x_3) \cdot (x_0 \oplus \bar{x}_6).
\end{aligned} \tag{2}$$

Note that this expression is different from (1) and contains more literals (but, obviously, the same number of EXOR-factors).

From Theorem 5, we have that any pseudocube of degree $n-1$ has a canonical expression consisting of one EXOR-factor. Conversely, given an EXOR-factor f, there exists a pseudocube P of degree $n-1$ whose pseudoproduct can be expressed by f (not necessarily in CEX form). To prove this point, first apply Lemma 2 to transform f into an equivalent EXOR-factor f' with the same variables of f, all in direct form, or with the last variable complemented. Using Definition 9, we can then reconstruct the unique pseudocube P such that $CEX(P) = f'$.

Given two pseudocubes P_1, P_2 of degree m such that there exists a subset α of variables with $P_2 = \alpha(P_1)$, we have that $P = P_1 \cup P_2$ is a pseudocube of degree $m+1$ (Theorem 2). P_1 and P_2 have the same canonical and noncanonical variables and α may be limited to noncanonical variables (proof of Proposition 5). We now study how to derive $CEX(P)$ from $CEX(P_1)$ and $CEX(P_2)$. Note that P will have the same canonical variables of P_1, P_2, plus a new one taken from the noncanonical variables of P_1, P_2.

Let $\{x_{i_0}, \ldots, x_{i_{n-m-1}}\}$ be the set of noncanonical variables of P_1 and P_2 and let $\alpha = \{x_{i_0}, \ldots, x_{i_k}\}$, $k \le n-m-1$. We have $CEX(P_1) = f_{i_0} \cdot \ldots \cdot f_{i_k} \cdot f_{i_{k+1}} \cdot \ldots \cdot f_{i_{n-m-1}}$, where f_{i_j} contains the noncanonical variable x_{i_j}. By Lemma 1.1, we then have $CEX(P_2) = \bar{f}_{i_0} \cdot \ldots \cdot \bar{f}_{i_k} \cdot f_{i_{k+1}} \cdot \ldots \cdot f_{i_{n-m-1}}$. That is,

$$\begin{aligned}
CEX(P_1) + CEX(P_2) &= f_{i_{k+1}} \cdot \ldots \cdot f_{i_{n-m-1}} \\
&\quad \cdot (f_{i_0} \cdot \ldots \cdot f_{i_k} + \bar{f}_{i_0} \cdot \ldots \cdot \bar{f}_{i_k}).
\end{aligned}$$

For $|\alpha| = 1$ (i.e., $\alpha = \{x_{i_0}\}$), we have $CEX(P_1) + CEX(P_2) = f_{i_{k+1}} \cdot \ldots \cdot f_{i_{n-m-1}}$, which is the expression $CEX(P)$. x_{i_0} is the new canonical variable of P, and does not appear in that expression (in fact, P_1 and P_2 are identical in the subexpressions with $x_{i_0} = 0$ and $x_{i_0} = 1$). For $|\alpha| > 1$, we have, by Lemma 1.2:

$$\begin{aligned}
CEX(P_1) + CEX(P_2) &= f_{i_{k+1}} \cdot \ldots \cdot f_{i_{n-m-1}} \\
&\quad \cdot (\bar{f}_{i_0} \oplus f_{i_1})(\bar{f}_{i_0} \oplus f_{i_2}) \cdot \ldots \\
&\quad \cdot (\bar{f}_{i_0} \oplus f_{i_k}).
\end{aligned} \tag{3}$$

We transform this expression into $CEX(P)$ by the following rule:

Rule 1. *In (3), each subexpression $(\bar{f}_{i_0} \oplus f_{i_j})$, $1 \le j \le k$, is an EXOR-factor that is rearranged as follows*

1. *The complementation over f_{i_0} is assigned to its variable x_{i_0} (Lemma 1.1), that is, f_{i_0} is changed to ${f^1}_{i_0}$ with all the canonical variables in direct form; the new expression $({f^1}_{i_0} \oplus f_{i_j})$ is simplified by eliminating each canonical variable y, if any, appearing in both ${f^1}_{i_0}$ and f_{i_j} (in fact, $y \oplus y = 0$ and $0 \oplus f = f$) to yield the new expression ${f^2}_{i_j}$;*
2. *If ${f^2}_{i_j}$ contains $\bar{x}_{i_0}$, the complementation is moved from this variable to the last variable x_{i_j} of ${f^2}_{i_j}$ to obtain the expression ${f^3}_{i_j}$; note that ${f^3}_{i_j}$ is an EXOR-factor containing some canonical variables, and the noncanonical variables x_{i_0} and $\hat{x}_{i_j}$; include x_{i_0} among the canonical variables, and reorder this set for increasing values of the indices, to obtain the expression g_{i_j} ($\hat{x}_{i_j}$ remains in the last position);*
3. *Expression (3) has been transformed into the form $f_{i_{k+1}} \cdot \ldots \cdot f_{i_{n-m-1}} \cdot g_{i_1} \cdot \ldots \cdot g_{i_k}$ composed of $n - m - 1$ EXOR-factors, each containing a noncanonical variable: Sort these factors for increasing order of the indices of such variables.*

The expression obtained by Rule 1 is in CEX form. In fact, this expression is $CEX(P)$ and includes the new canonical variable x_{i_0} (first variable of α). For example, consider the canonical matrices of the two pseudoproducts P_1, $P_2 = \alpha(P_1)$ shown in Fig. 4. The canonical variables are x_0, x_1, x_4, and $\alpha = \{x_2, x_3, x_6\}$. We have:

$$CEX(P_1) = (x_1 \oplus x_2) \cdot (x_0 \oplus x_1 \oplus x_3) \cdot (x_0 \oplus x_1 \oplus x_4 \oplus \bar{x}_5) \\ \cdot (x_0 \oplus \bar{x}_6)$$

$$CEX(P_2) = (x_1 \oplus \bar{x}_2) \cdot (x_0 \oplus x_1 \oplus \bar{x}_3) \cdot (x_0 \oplus x_1 \oplus x_4 \oplus \bar{x}_5) \\ \cdot (x_0 \oplus x_6)$$

and, applying Rule 1:

$$\begin{aligned} CEX(P) &= (x_0 \oplus x_1 \oplus x_4 \oplus \bar{x}_5) \cdot (\overline{(x_1 \oplus x_2)} \oplus (x_0 \oplus x_1 \oplus x_3)) \\ &\quad \cdot (\overline{(x_1 \oplus x_2)} \oplus (x_0 \oplus \bar{x}_6)) \\ &= (\text{by rule 1.1})(x_0 \oplus x_1 \oplus x_4 \oplus \bar{x}_5) \cdot (\bar{x}_2 \oplus x_0 \oplus x_3) \\ &\qquad \cdot (x_1 \oplus \bar{x}_2 \oplus x_0 \oplus \bar{x}_6) \\ &= (\text{by rule 1.2})(x_0 \oplus x_1 \oplus x_4 \oplus \bar{x}_5) \cdot (x_0 \oplus x_2 \oplus \bar{x}_3) \\ &\qquad \cdot (x_0 \oplus x_1 \oplus x_2 \oplus x_6) \\ &= (\text{by rule 1.3})(x_0 \oplus x_2 \oplus \bar{x}_3) \cdot (x_0 \oplus x_1 \oplus x_4 \oplus \bar{x}_5) \\ &\qquad \cdot (x_0 \oplus x_1 \oplus x_2 \oplus x_6), \end{aligned}$$

where x_2 is the new canonical variable.

In summary, for a pseudocube $P = P_1 \cup P_2$, $CEX(P)$ can be built from $CEX(P_1)$ and $CEX(P_2)$ with the following Algorithm 2. The time is linear with the size of the input expressions.

Algorithm 2 (build $CEX(P)$ from $CEX(P_1)$, $CEX(P_2)$)
1. compare $CEX(P_1)$ with $CEX(P_2)$ to determine if P_1 and P_2 belong to the same class; this occurs when the two expressions are represented by two identical strings, except, possibly, for the last variable of each EXOR-factor that may appear in direct or complemented form. If this is the case, derive α;

2. **if**$|\alpha| = 1$ (i.e., $\alpha = \{x_{i_0}\}$
 then build $CEX(P)$ by suppressing from $CEX(P_1)$ the EXOR-factor containing x_{i_0}
 else derive (3) and apply Rule 1.

Algorithm 2 is the basis of a synthesis method to be discussed in Section 6.

5 Expressing Boolean Functions: The Adder

The algebraic representation of Boolean functions is a crucial tool in digital design and symbolic manipulation [7]. After the initial development of Switching Theory, in the fifties and sixties, there has been a steady interest in the applications of synthesis techniques that are now directed to VLSI design [4], [11]. For this purpose, Boolean functions are represented by algebraic expressions or binary decision diagrams (BDD) [1], [3]. Let us discuss the role of pseudoproducts in this area, showing how they can be used to derive the expression of an arbitrary function. A comparison will be made with the algebraic representation in disjunctive form, also called "sum of products" (SP), and with BDDs in some significant cases.

The simplest case is the one of a function that is a pseudoproduct, hence, is directly expressed in CEX form. Since such a function is defined on the 2^m points of a pseudocube P, its specification (truth table) has size $\Theta(2^m \times n)$. For $m = \Theta(n)$, the time to build CEX(P) is exponential in n, although the length of CEX(P) is $O(n^2)$. This is clearly unavoidable when building any algebraic expression for P, since the whole specification of P must be examined.

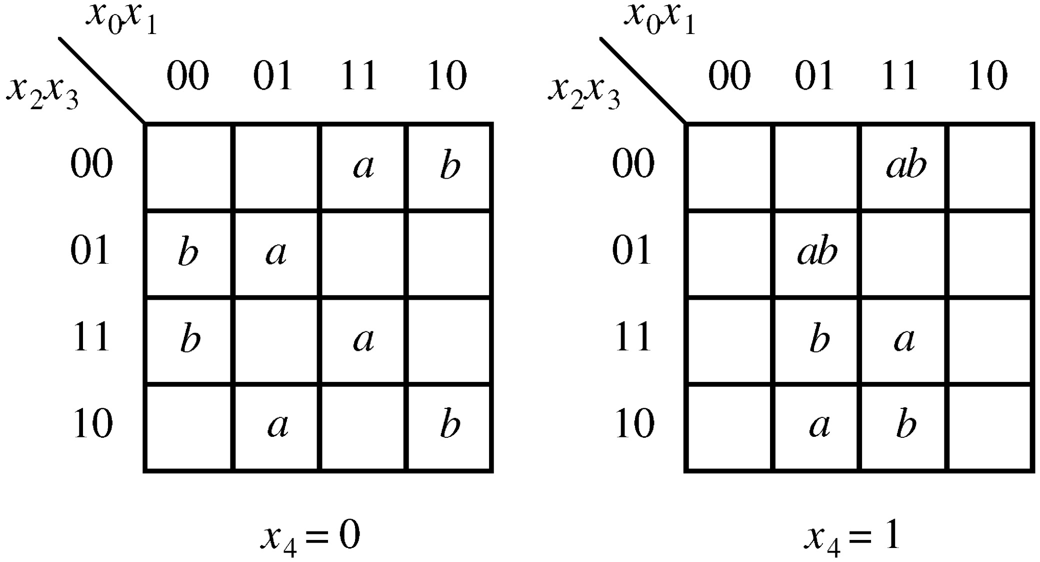
An obvious example of pseudoproduct is the parity function in B^n that has value 1 exactly in the points of a pseudocube of degree $n - 1$. This function is expressed in CEX form as a single EXOR-factor containing n literals (in fact, the EXOR of all the variables, one of which in complemented form) versus a minimal SP containing $n2^{n-1}$ literals and a reduced BDD containing $2n$ nonterminal nodes. Some examples in B^4 are the eight functions with value 1 in the pseudocubes of Fig. 3. It can be easily verified that all of them have CEX expressions shorter than the corresponding minimal SP forms and a number of literals smaller than the number of nonterminal nodes of the reduced BDDs.

In general, an arbitrary function can be expressed as a disjunction of pseudoproducts, giving rise to a "sum of pseudoproducts" (SPP) form. For example, the set of points of B^5 marked with a and/or b in Fig. 5 is the union of two partially overlapping pseudocubes P_a, P_b. The function with value 1 in the points of $P_a \cup P_b$ is then expressed as CEX(P_a) + CEX(P_b), that is:

$$x_1 \cdot (x_0 \oplus x_2 \oplus x_3) + (x_0 \oplus x_3) \cdot (x_1 \oplus \overline{x}_4),$$

while the minimal SP for the same function contains 27 literals, and the reduced BDD contains nine nonterminal nodes. If a function has to be implemented as a digital circuit, the role of SPP crucially relies on the use of EXOR gates (see [8], [13], [22] for their realization). Moreover, passing from SP to SPP amounts to pass from a two-level to a three-level circuit. These features have always to be taken into account, and will not be further repeated.

Important examples of SPP forms are encountered in the design of an n-bit adder. First, consider a stage of a carry-ripple adder with three inputs a, b (local digits), and c



$x_2x_3 \backslash x_0x_1$	00	01	11	10
00			a	b
01	b	a		
11	b		a	
10		a		b

$x_4 = 0$

$x_2x_3 \backslash x_0x_1$	00	01	11	10
00			ab	
01		ab		
11		b	a	
10		a	b	

$x_4 = 1$

Fig. 5. The union of two pseudocubes.

(previous carry), and two outputs s (local sum) and c^* (carry to the next stage). We have:

SP: $s = abc + a\bar{b}\bar{c} + \bar{a}b\bar{c} + \bar{a}\bar{b}c$,
SPP: $s = a \oplus b \oplus c$; (one pseudoproduct of degree 2)
SP: $c^* = ab + bc + ac$,
SPP: $c^* = bc + a(b \oplus c)$; (two pseudoproducts of degree 1),

where the SPP forms are much more compact.

A more interesting circuit is a fully parallel n-bit adder, with inputs $A = a_{n-1}a_{n-2}\dots a_0$, $B = b_{n-1}b_{n-2}\dots b_0$, and output $S = s_n s_{n-1}\dots s_0$. Each output s_i will be obtained as a function of the inputs $a_i, \dots, a_0, b_i, \dots, b_0$, in SP, or SPP, form. For studying these functions, we make use of the function carry c_i relative to the sum of $a_{i-1}\dots a_0$ plus $b_{i-1}\dots b_0$, although this carry will not be explicitly generated by the circuit. For SP, define s_i through the relations:

$$\begin{aligned} s_0 &= a_0\bar{b}_0 + \bar{a}_0 b_0, \\ s_i &= a_i b_i c_i + a_i\bar{b}_i\bar{c}_i + \bar{a}_i b_i \bar{c}_i + \bar{a}_i\bar{b}_i c_i, \quad i > 0, \end{aligned} \tag{4}$$

where c_i and $\bar{c}_i$ are recursively defined as:

$$\begin{aligned} c_1 &= a_0 b_0, \\ c_i &= a_{i-1}b_{i-1} + b_{i-1}c_{i-1} + a_{i-1}c_{i-1}, \quad i > 1, \end{aligned} \tag{5.1}$$

$$\begin{aligned} \overline{c_1} &= \bar{a}_0 + \bar{b}_0 \\ \bar{c}_i &= \bar{a}_{i-1}\bar{b}_{i-1} + \bar{b}_{i-1}\bar{c}_{i-1} + \bar{a}_{i-1}\bar{c}_{i-1}, \quad i > 1. \end{aligned} \tag{5.2}$$

These relations can be used to derive the minimal SP forms for all the s_i. In fact, the expressions for s_i and c_i cannot be further reduced, independently of the fact that, in the right-hand side of the relations, a_i, b_i are variables and c_i is a function. For example, we have:

$$s_1 = a_1b_1a_0b_0 + a_1\bar{b}_1\bar{a}_0 + a_1\bar{b}_1\bar{b}_0 + \bar{a}_1b_1\bar{a}_0 + \bar{a}_1b_1\bar{b}_0 + \bar{a}_1\bar{b}_1a_0b_0.$$

Instead of giving the minimal SP expression for each s_i, we evaluate the complexity of this expression in terms of the number Π_i of its products and the number Λ_i of its literals. For this purpose, we need to evaluate the numbers π_i, λ_i, and $\bar{\pi}_i$, $\bar{\lambda}_i$ of products and literals of c_i and $\bar{c}_i$, respectively. From (5.1) and (5.2), we derive the recurrences:

$$\begin{aligned} &\pi_1 = 1,\ \pi_i = 2\pi_{i-1} + 1, \\ &\lambda_1 = 2,\ \lambda_i = 2\lambda_{i-1} + 2\pi_{i-1} + 2, \\ &\bar{\pi}_1 = 2,\ \bar{\pi}_i = 2\bar{\pi}_{i-1} + 1, \\ &\bar{\lambda}_1 = 2,\ \bar{\lambda}_i = 2\bar{\lambda}_{i-1} + 2\bar{\pi}_{i-1} + 2, \end{aligned}$$

that have solutions:

$$\begin{aligned} \pi_i &= 2^i - 1, \\ \lambda_i &= i2^i, \\ \bar{\pi}_i &= 2^i + 2^{i-1} - 1, \\ \bar{\lambda}_i &= i2^i + (i-1)2^{i-1}. \end{aligned}$$

From (4), we then have:

$$\Pi_i = 2\pi_i + 2\bar{\pi}_i = 2^{i+2} + 2^i - 4. \tag{6}$$

To compute the number of literals of s_i, note that, when each term $\hat{a}_i\hat{b}_i\hat{c}_i$ of (4) is expanded by substituting the expression for $\hat{c}_i$, each product appearing in $\hat{c}_i$ is multiplied by two new variables. We then have:

$$\begin{aligned} \Lambda_i &= 2\lambda_i + 2\bar{\lambda}_i + 4\pi_i + 4\bar{\pi}_i \\ &= 2^{i+3} + i2^{i+2} + 2^{i+1} + (i-1)2^i - 8. \end{aligned} \tag{7}$$

For example, from (6) and (7), we have $\Pi_1 = 6$, $\Lambda_1 = 20$, that are the numbers of products and literals of the minimal SP form of s_1, as found in the algebraic expression for s_1 above. As a conclusion, we note that both Π_i and Λ_i grow exponentially with i.

To express s_i in SPP, we represent our functions in the recursive maps of Fig. 6 (not necessarily Karnaugh maps) that are easily constructed from the definition of sum and carry. Each submap containing a function (e.g., the submap of s_i for $a_ib_i = 00$, containing c_i) recursively stands for the map of that function (i.e., the map of c_i labeled with

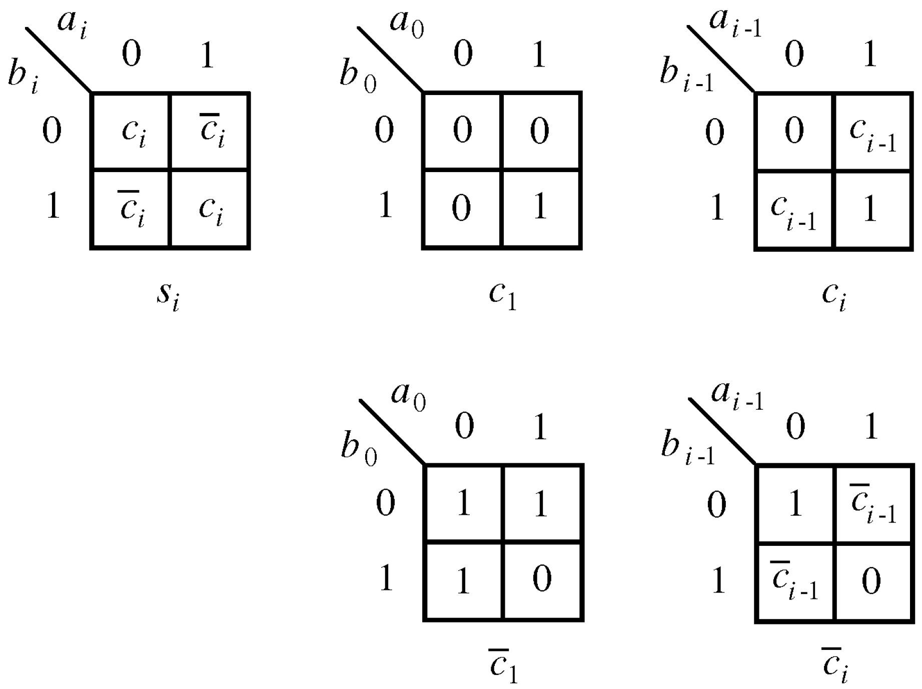


Fig. 6. Map representation of sum and carry in a fully parallel adder.

a_{i-1}, b_{i-1}). Consider c_i. Each pseudocube P of c_{i-1}, contained in the submap for $a_{i-1}b_{i-1} = 01$, can be transformed into a pseudocube Q in the submap for $a_{i-1}b_{i-1} = 10$ by complementing a_{i-1} and b_{i-1}. Therefore, P and Q belong to the same class Γ and can be joined to form a pseudocube $R = P \cup Q$ (Theorem 2). The corresponding pseudoproducts can be combined, to easily obtain: $CEX(R) = (a_{i-1} \oplus b_{i-1})CEX(P)$, where $CEX(P)$ is recursively computed on $a_{i-2}, \ldots, a_0, b_{i-2}, \ldots, b_0$. In $CEX(R)$, a_{i-1} is a new canonical variable and b_{i-1} is a new noncanonical variable. To form a (not necessarily minimal) SPP expression for c_i, we consider all the pseudoproducts of c_{i-1}, pairwise combined as above, plus the pseudoproduct a_ib_i corresponding to the submap for $a_{i-1}b_{i-1} = 11$, which contains all 1s. We inductively assume that c_{i-1} is expressed in SPP, to obtain:

$$\begin{aligned}
c_1 =& a_0b_0,\\
c_i =& a_{i-1}b_{i-1} + (a_{i-1} \oplus b_{i-1})c_{i-1}\\
=& a_{i-1}b_{i-1} + (a_{i-1} \oplus b_{i-1})a_{i-2}b_{i-2}\\
&+ (a_{i-1} \oplus b_{i-1})(a_{i-2} \oplus b_{i-2})c_{i-2}\\
=& \cdots\\
=& a_{i-1}b_{i-1} + (a_{i-1} \oplus b_{i-1})a_{i-2}b_{i-2}\\
&+ (a_{i-1} \oplus b_{i-1})(a_{i-2} \oplus b_{i-2})a_{i-3}b_{i-3} + \cdots\\
&+ (a_{i-1} \oplus b_{i-1}) \cdots (a_1 \oplus b_1)a_0b_0.
\end{aligned} \tag{8}$$

Similarly, we have:

$$\begin{aligned}
\overline{c}_1 =& \overline{a}_0\overline{b}_0 + (a_0 \oplus b_0),\\
\overline{c}_i =& \overline{a}_{i-1}\overline{b}_{i-1} + (a_{i-1} \oplus b_{i-1})\overline{c}_{i-1}\\
=& \cdots\\
=& \overline{a}_{i-1}\overline{b}_{i-1} + (a_{i-1} \oplus b_{i-1})\overline{a}_{i-2}\overline{b}_{i-2}\\
&+ (a_{i-1} \oplus b_{i-1})(a_{i-2} \oplus b_{i-2})\overline{a}_{i-3}\overline{b}_{i-3} + \cdots\\
&+ (a_{i-1} \oplus b_{i-1}) \cdots (a_1 \oplus b_1)\overline{a}_0\overline{b}_0\\
&+ (a_{i-1} \oplus b_{i-1}) \cdots (a_1 \oplus b_1)(a_0 \oplus b_0).
\end{aligned} \tag{9}$$

Consider now s_i. We can combine the pseudocubes in the two submaps containing c_i and the ones in the two submaps containing $\overline{c}_i$, to obtain the expression:

$$s_i = (a_i \oplus b_i)\overline{c}_i + \overline{(a_i \oplus b_i)}c_i = (a_i \oplus b_i)\overline{c}_i + (a_i \oplus \overline{b}_i)c_i. \tag{10}$$

Combining (8), (9), and (10), we directly obtain an SPP form for s_i that can be easily simplified to the following:

$$\begin{aligned}
s_i =& (a_i \oplus a_{i-1} \oplus b_i)(a_{i-1} \oplus \overline{b}_{i-1})\\
&+ (a_i \oplus a_{i-2} \oplus b_i)(a_{i-1} \oplus b_{i-1})(a_{i-2} \oplus \overline{b}_{i-2})\\
&+ \cdots\\
&+ (a_i \oplus a_0 \oplus b_i)(a_{i-1} \oplus b_{i-1})(a_{i-2} \oplus b_{i-2}) \cdots (a_0 \oplus \overline{b}_0)\\
&+ (a_i \oplus b_i)(a_{i-1} \oplus b_{i-1})(a_{i-2} \oplus b_{i-2}) \cdots (a_0 \oplus b_0).
\end{aligned} \tag{11}$$

It must be noted that all the pseudoproducts appearing in (11) correspond to disjoint pseudocubes, because any two of them respectively contain the disjoint factors $(a_j \oplus \overline{b}_j)$ and $(a_j \oplus b_j)$. The last pseudoproduct represents the bottom-left to top-right diagonal of the map of s_i, which contains all 1s.

To compare the complexity of this new expression with the one of a minimal SP form for s_i, let us evaluate the number Ψ_i of its pseudoproducts, the number Υ_i of its

EXOR-factors, and the number Δ_i of its literals. By inspection of (11) we immediately have:

$$\Psi_i = i + 1, \tag{12}$$

$$\Upsilon_i = \frac{(i+1)(i+2)}{2} + i = \frac{1}{2}i^2 + \frac{5}{2}i + 1, \tag{13}$$

$$\Delta_i = \sum_{j=0}^{i-1}(5 + 2j) + 2i + 2 = i^2 + 6i + 2. \tag{14}$$

Comparing the values of Ψ_i and Υ_i with Π_i of (6) may not be particularly significant, due to the different structure of the two forms. The superiority of SPP over SP is clearly proven by examining the number of literals. Note the quadratic growth of Δ_i versus the exponential growth of Λ_i shown in (7).

Many other circuits have been proposed for the adder and is out of our scope to compare them all. We simply recall that a three-level n-bit adder based on PLAs with input decoders has been proposed in [16]. This circuit has $n^2 + 1$ PLA columns (products), hence, a cubic number of literals.

6 Minimization and Benchmark Results

In the previous section, we have shown that the SPP forms for addition are very compact. Indeed, this may not be surprising, due to the presence of EXOR operators in the CEX expressions of the pseudoproducts and to the particular distribution of the 1s in the functions to be represented. However, we speculate on the possibility of using SPP to express arbitrary functions economically, hence, to get small circuits.

An important approach based on an AND–EXOR form was devised long ago by Reed and Muller to represent arbitrary functions [5], [12]. This form is extensively used for function classification [19], [20] and has originated several methods for circuit synthesis. In particular, heuristics for generating short AND–EXOR forms, and Decode–AND–EXOR forms have been, respectively, proposed in [18], [23] and [15]. Other three-level forms using EXOR are proposed in [9], [14], [17]. None of these techniques, however, has reached the maturity of SP minimization. Our SPP method is a direct generalization of the one for SP where pseudoproducts are substituted for products and benefits of a well-established structure.

A basic concept related to SP forms is the one of *prime implicant* of a function f, that is, a product implying (i.e., covered by) f and not implying any other product implying f. As is well-known, a minimal SP form for f can be built as a disjunction of prime implicants only. This concept immediately extends to the one of *prime pseudoproduct*, that is, a pseudoproduct implying f and not implying any other pseudoproduct of f.

Prime pseudoproducts can be built with successive pairwise combinations of pseudocubes of smaller order belonging to the same class Γ, using Algorithm 2 of Section 4. This suggests an immediate extension of the classical SP minimization methods to construct an SPP form with minimal number of pseudoproducts, composed of prime pseudoproducts only. For example, the well-known method of Quine-McCluskey [7] can be immediately reformulated as follows:

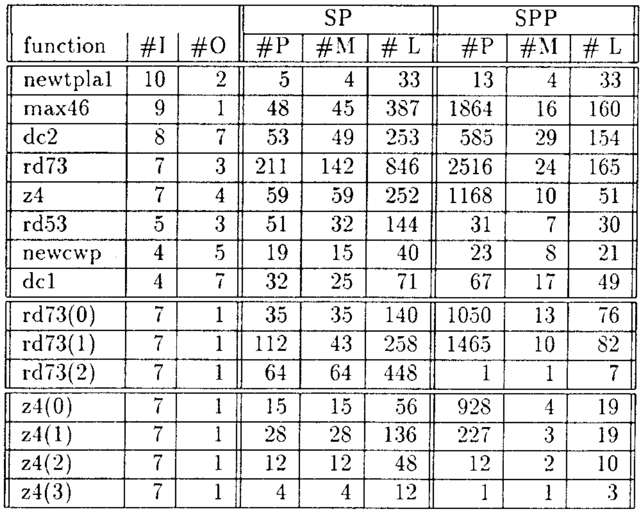
Algorithm 3. (build an SPP with minimal number of pseudoproducts)
1. Start from the CEX expressions of all the pseudoproducts of degree $k = 1$ (i.e., CEX of all the pairs of points);
2. For increasing k, generate the CEX expressions of the pseudoproducts of degree $k + 1$ from pairs of pseudoproducts of degree k by applying Algorithm 2 to the corresponding pseudocubes; for each k, retain only the pseudoproducts corresponding to pseudocubes that cannot be combined into larger ones, as the prime pseudoproducts of degree k;
3. Make a minimal selection of prime pseudoproducts covering the whole function; the OR of their CEX expressions gives a minimal SPP.

The CEX expression of a pseudoproduct built by composition of two pseudoproducts of smaller degree contains one EXOR-factor less than the ones of the two components. Then, an SPP form with minimal number of EXOR-factors can also be built with prime pseudoproducts only. Algorithm 3 can be easily transformed for this purpose with an immediate adaptation of Step 3. Deriving an SPP form with *minimal number of literals* is a more complex task. First, the number of literals in a CEX expression may depend on the ordering of the variables, as noted in Section 4. That is, the minimization holds relative to a chosen ordering. Second, when two CEX expressions are combined by Algorithm 2, the number of literals in the new expression not necessarily decreases. For $P = P_1 \cup P_2$, we have, from Rule 1, that $CEX(P)$ and $CEX(P_1)$ (or, similarly, $CEX(P_2)$) differ for a portion of the formula, respectively, $(\bar{f}_{i_0} \oplus f_{i_1}) \cdot \ldots \cdot (\bar{f}_{i_0} \oplus f_{i_k})$ simplified by Rule 1.1 and $f_{i_0} \cdot \ldots \cdot f_{i_k}$. Although the former portion contains, in general, fewer literals than the latter, due to the simplification introduced by the rule, this is not necessarily true. The most elementary instances of this fact are found in B^3. The points 000 and 111 constitute two pseudocubes P_1, P_2 of degree 0. Their union is a pseudocube P of degree 1. We have $CEX(P_1) = \overline{x}_0\overline{x}_1\overline{x}_2$, $CEX(P_2) = x_0x_1x_2$, $CEX(P) = (x_0 \oplus \overline{x}_1)(x_0 \oplus \overline{x}_2)$. The expressions for P_1, P_2 have three EXOR-factors (reduced to single variables) and three literals, while the expression for P has two EXOR-factors and four literals. As a consequence, the construction of an SPP with minimum number of literals cannot be limited to prime pseudoproducts. Indeed, in such a construction a pseudoproduct can be excluded from consideration only if is covered by another pseudoproduct whose expression contains fewer literals.

The above property is implemented in an easy modification of Algorithm 3. We have:

Algorithm 4 (build an SPP with minimal number of literals, under a chosen ordering of variables)
1. Start from the CEX expressions of all the pseudoproducts of degree $k = 0$ (single points);
2. For increasing k, generate the CEX expressions of the pseudoproducts of degree $k + 1$ from pairs of pseudoproducts of degree k; discard a pseudoproduct of degree k if its CEX expression contains h literals and is combined into one

TABLE 1
Comparison of Benchmark Results



			SP			SPP		
function	#I	#O	#P	#M	# L	#P	#M	# L
newtpla1	10	2	5	4	33	13	4	33
max46	9	1	48	45	387	1864	16	160
dc2	8	7	53	49	253	585	29	154
rd73	7	3	211	142	846	2516	24	165
z4	7	4	59	59	252	1168	10	51
rd53	5	3	51	32	144	31	7	30
newcwp	4	5	19	15	40	23	8	21
dc1	4	7	32	25	71	67	17	49
rd73(0)	7	1	35	35	140	1050	13	76
rd73(1)	7	1	112	43	258	1465	10	82
rd73(2)	7	1	64	64	448	1	1	7
z4(0)	7	1	15	15	56	928	4	19
z4(1)	7	1	28	28	136	227	3	19
z4(2)	7	1	12	12	48	12	2	10
z4(3)	7	1	4	4	12	1	1	3

#I: number of input variables
#O: number of output variables
#P: total number of prime implicants (or prime pseudoproducts)
#M: number of prime implicants (or prime pseudoproducts) in the minimal expression.
#L: number of literals in the minimal expression

of degree $k+1$ whose expression contains $\leq h$ literals;
3. Make a selection with minimal number of literals from among the pseudoproducts retained in Step 2 (with extended terminology, call them also "prime pseudoproducts" when referring to Algorithm 4).

Depending on the function, the number of literals in a minimal SPP expression is less than or equal to the number of literals in a minimal SP expression for the same function. If the two numbers are equal, Algorithm 4 may, in fact, generate an SP expression as a special case of SPP.

Comparing Algorithms 3 and 4, we see that the selection in Step 3 is made from among a larger number of pseudoproducts in Algorithm 4 than in Algorithm 3. This does not increase the asymptotic worst-case complexity of the method, since the minimization problem is NP-hard for SP [6] and for BDD [3], and remains clearly NP-hard for SPP in its two variants. In other words, all minimization techniques are, in theory, very hard. In practice, SP minimization is often rather easy, although some functions of many variables may force us to derive a suboptimal expression heuristically, due to the high complexity of the computation required to attain a minimal form [4], [11]. In SPP minimization, this situation is much more likely to occur due to the possibly huge number of prime pseudoproducts of the function. On the other hand, we may expect that most functions have an SPP expression much shorter than the corresponding minimal SP, since there are many more pseudoproducts than products. The results obtained with some benchmarks confirm these observations, as we now outline.

We have adopted the benchmarks functions of [21] that are widely used in the evaluation of synthesis methods (e.g., see [4], [15]). The results for eight such functions are shown in the upper section of Table 1, as samples of the potentiality of our method (Algorithm 4 has been used for SPP). All of them have a few input variables so that the minimal expressions were determined. The different outputs of each function have been minimized separately and, for two of them, the results relative to the single outputs are reported in the two lower sections of the table. The function $newtpla1$ yields identical results in SP and SPP, meaning that the prime pseudoproducts in the minimal form are, in fact, prime implicants. For the functions $max46$, $dc2$, $newcwp$, and $dc1$, the size of the minimal PSS expression is about one half of the corresponding SP. The three functions $rd73$, $z4$, and $rd53$ show a much greater improvement of SPP over SP. Among the results for the single outputs of $rd73$ and $z4$, we find that $rd73(2)$ and $z4(3)$ are, indeed, single pseudoproducts, with enormous advantage, in particular for $rd73(2)$. Single outputs consisting of single pseudoproducts are encountered in five other cases (not shown) for the functions studied in Table 1. On the negative side, we note a generalized greater effort to produce the SPP minimal forms, due to the highest number of prime pseudoproducts over prime implicants.

7 A Special Class of Functions

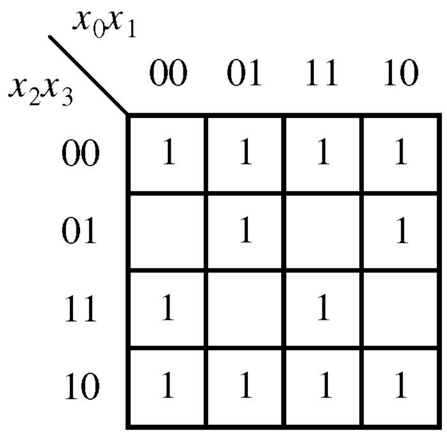
To conclude our discussion on SPP forms, let us now characterize a class of functions suitable for efficient SPP representation, based on special symmetries. It is known that the class of symmetric Boolean functions (i.e., functions insensitive to variable permutations) is particularly suitable for BDD representation [3]. For SPP, we define a new type of symmetry. Given a function f with value 1 in $S \subseteq B^n$ and a subset of variables α, denoted by $\alpha(f)$, the function with value 1 in the points of $\alpha(S)$ (see Definition 7).

Definition 10. *A function f is auto-symmetric in α if $f = \alpha(f)$.*

In the function f of Fig. 7, complementing x_0 and x_1 amounts to moving each 1 onto another 1 without changing the function. That is, f is auto-symmetric in $\{x_0, x_1\}$ (but not in $\{x_0\}$ or in $\{x_1\}$). Other examples are the function with value 1 in the points marked a or b in Fig. 5, which is auto-symmetric in $\{x_0, x_3\}$; and the parity function, which is auto-symmetric in any subset of variables of even cardinality. If a function is auto-symmetric in two different subsets α, β, it is clearly auto-symmetric also in $\alpha \oplus \beta$. The function of Fig. 7 is auto-symmetric in $\{x_0, x_1\}$, $\{x_0, x_2\}$, and $\{x_1, x_2\} = \{x_0, x_1\} \oplus \{x_0, x_2\}$. For the parity function, note that if two sets α, β have even cardinality, the set $\alpha \oplus \beta$ also has even cardinality.

A product function p is auto-symmetric in any subset of the variables which do not appear in the AND expression of p (i.e., in any subset of canonical variables). This is not necessarily true for pseudoproducts. In fact, if p is a pseudoproduct auto-symmetric in α, then α must contain at least one canonical variable (otherwise, $\alpha(p) \neq p$ by Theorem 1). For an arbitrary function, we have:

Proposition 8. *Let f be auto-symmetric in α and p be a prime pseudoproduct of f. Then, $\forall x_i \in \alpha$, the pseudocube P associated with p is composed of two pseudocubes lying in the subspaces B^{n-1} with $x_i = 0$ and $x_i = 1$, respectively.*



x_2x_3 \ x_0x_1	00	01	11	10
00	1	1	1	1
01		1		1
11	1		1	
10	1	1	1	1

Fig. 7. A function auto-symmetric in $\{x_0, x_1\}$, $\{x_0, x_2\}$, $\{x_1, x_2\}$.

Proof. Let B_0, B_1 denote the two subspaces. By definition of pseudocube, P lies in B_0, or in B_1, or half in B_0 and half in B_1. If P is in B_0, then $\alpha(P)$ is in B_1 and $P \cup \alpha(P)$ is a pseudocube of greater degree against the hypothesis that p is prime. Similarly, P cannot lie in B_1. □

Proposition 8 is the basis of a stronger result, relating auto-symmetry to primality. We have:

Theorem 6. *Let f be auto-symmetric in $\alpha_1, \ldots, \alpha_k$. Choose the variables $x_{j_1}, \ldots, x_{j_k}$ such that $x_{j_i} \in \alpha_i$, $1 \leq i \leq k$, and $x_{j_i} \notin \alpha_h$, $1 \leq i < h \leq k$. Let f_k be the restriction of f to the subspace B^{n-k} with $x_{j_1} \ldots x_{j_k} = \mathbf{v}$, where $\mathbf{v}$ is an arbitrary binary vector of k elements. Then, f_k and f have the same number of prime pseudoproducts.*

Proof. We prove that each pseudocube associated to a prime pseudoproduct of f is divided into 2^k equal pseudocubes, lying in the subspaces where $x_{j_1}, \ldots, x_{j_k}$ assume all the possible sets of values. For $k = 1$ this thesis follows from Proposition 8. Then, proceed by induction. Let the thesis hold for the restriction f_{k-1} of f to the subspace B^{n-k-1} with $x_{j_1} \ldots x_{j_{k-1}} = \mathbf{v}$, where $\mathbf{v}$ is any binary vector of $k-1$ elements. Since $x_{j_1}, \ldots, x_{j_{k-1}} \notin \alpha_k$, the variations induced on f by complementing the variables in α_k move the 1s of f_{k-1} onto themselves. That is, f_{k-1} is auto-symmetric in α_k. Therefore, by Proposition 8, each pseudocube associated with a prime pseudoproduct of f_{k-1} is composed of two pseudocubes, lying in the subspaces with $x_{j_1} \ldots x_{j_{k-1}} x_{j_k} = \mathbf{v}0$ and $x_{j_1} \ldots x_{j_{k-1}} x_{j_k} = \mathbf{v}1$. The theorem immediately follows. □

The function f of Fig. 7 has two prime pseudoproducts of degree $n - 1 = 3$, with CEX expressions $\overline{x}_3$ and $x_0 \oplus x_1 \oplus x_2$, respectively. The three sets of auto-symmetry of f cannot be chosen together, to fulfill the conditions of Theorem 6. Groups of two sets do. We may choose, for example:

$$\alpha_1 = \{x_0, x_1\},\ \alpha_2 = \{x_0, x_2\},\ x_{j_1} = x_1,\ x_{j_2} = x_0,$$

which fulfill the theorem. Let f_k be the restriction of f to the subspace with $x_0 x_1 = 00$. Both f and f_k have two prime pseudoproducts, with each of the latter covering one fourth of the 1s of one prime pseudoproduct of f.

Auto-symmetric functions are then suitable for SPP representation, in particular when the minimization of the number of pseudoproducts, or of EXOR-factors, is sought for. In this case, only prime pseudoproducts are considered and the study of f can be carried out on the restriction f_k. The algebraic expresson of f is then easily found from the one of f_k. Large values of k in Theorem 6 generally imply SPP forms composed of a few prime pseudoproducts. If a function consists of the union of a few auto-symmetric functions, it is also suitable for SPP representation. In particular, if f has value 1 in 2^m points and $k = m$, f_k reduces to a single point, hence, f is a pseudoproduct. If each α_i contains only one variable, f is a product.

Significant examples of auto-symmetry are exhibited by the functions s_i (sum) and c_i (carry) of an n-bit adder, discussed in Section 5. Refer to the definition of s_i and c_i as presented in the maps of Fig. 6. Consider a map labeled a_j, b_j. After the recursive substitutions of the submaps, the columns (respectively, the rows) of the map become labeled with vectors $a_j a_{j-1} \ldots a_0$ (respectively, $b_j b_{j-1} \ldots b_0$), arranged in increasing binary order. (This arrangement is not the one of a Karnaugh map and facilitates the following). On this map, consider the portion D composed of the cells lying on the bottom-left to top-right diagonal; and the upper-left and lower-right triangular portions U and L separated by D. Denote by d_j, u_j, ℓ_j the functions with all 1s in D, U, L, respectively. Note that $u_j = \alpha(\ell_j)$, $\alpha = \{a_j, b_j, a_{j-1}, b_{j-1}, \ldots, a_0, b_0\}$. It can be easily proven by induction on i that $c_i = \ell_{i-1}$, hence, $\overline{c}_i = u_{i-1} + d_{i-1}$. From the definition of c_i in Fig. 6, we have:

$$c_i = \ell_{i-1} = a_{i-1} b_{i-1} + r_{i-1}, \tag{15}$$

where r_{i-1} is defined in Fig. 8. Similarly, we have:

$$s_i = d_i + t_i, \tag{16}$$

where t_i is also defined in Fig. 8.

The function s_i is defined in B^{2i}. Its components d_i and t_i are auto-symmetric. In fact, d_i is auto-symmetric in $\{a_i, b_i\}, \{a_{i-1}, b_{i-1}\}, \ldots, \{a_0, b_0\}$. Therefore, by Theorem 6, d_i can be studied in the subspace B^i with $a_i a_{i-1} \ldots a_0 = 00 \ldots 0$, where the restriction consists of one prime pseudoproduct with a single 1. d_i is then expressed by one prime prime pseudoproduct, as already found in (11).

Consider now t_i. Recalling that $u_{i-1} = \alpha(\ell_{i-1})$, $\alpha = \{a_{i-1}, b_{i-1}, \ldots, a_0, b_0\}$ from Fig. 8, we see that t_i is auto-symmetric in $\{a_i, b_i\}$ and in $\{a_i, a_{i-1}, b_{i-1}, \ldots, a_0, b_0\}$. Selecting b_i from the first set and a_i from the second, we have, from Theorem 6, that t_i can be studied in the subspace B^{2i-2} with $a_i b_i = 00$. That is, we can restrict our study to the prime pseudoproducts of the function ℓ_{i-1} that is, in turn, given in (15). This function is composed of the product $a_{i-1} b_{i-1}$ and the function r_{i-1}, which is auto-symmetric in $\{a_{i-1}, b_{i-1}\}$ (Fig. 8). r_{i-1} is then studied in the subspace B^{2i-3} with $a_{i-1} = 0$. Indeed, the function has its 1s only in $a_{i-1} b_{i-1} = 01$, therefore, the examination can be restricted to this subspace only. Here, we find ℓ_{i-2} and the reasoning repeats recursively until the function reduces to a single point. Since, at each step, we add a new prime pseudopro-

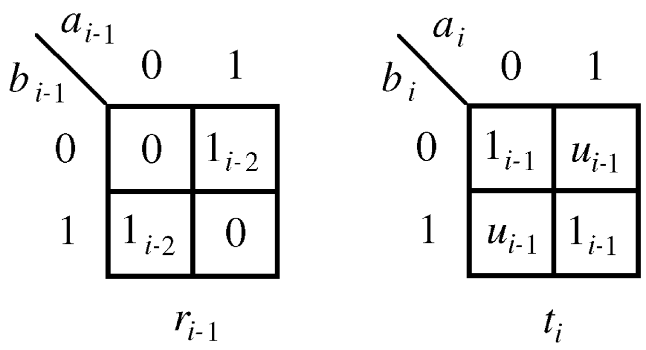


Fig. 8. Map representation of the functions r_{i-1} and t_i.

duct a_ib_j, the function ℓ_{i-1} is expressed by the OR of i prime pseudoproducts, as already found in (8).

Combining all these results, s_i is expressed by the OR of $i+1$ prime pseudoproducts that are exactly the ones included in (11).

8 Concluding Remarks

In this work, we have introduced the new concept of pseudocube in B^n, as an extension of the concept of subcube. Pseudocubes exhibit several properties already known for subcubes, plus some new strong ones. We have studied the characteristic function of a pseudocube, called pseudoproduct, and its algebraic expression in EXOR-AND form: This is our "new Boolean function". The expression of a pseudoproduct is a basic tool for the algebraic representation of arbitrary Boolean functions, in the form of sum of pseudoproducts (SPP). Compared with other forms based on EXOR, SPP is completely new.

We have developed this work to contribute to the comprehension, classification, and algebraic representation of Boolean functions. However, we have also argued that SPP may be an economic form to express functions, leading to small combinatorial circuits. For this purpose, we have determined the SPP expressions for the output functions of a parallel adder, as a significant example of the advantages of our approach. Experiments on popular benchmarks for logic synthesis have also given encouraging results.

As in all initial works, much effort is still needed to prove the versatility of pseudoproducts and their role in Boolean algebra and circuit design. A challenging issue is to study the relations between SPP forms and binary decision diagrams (BDDs) and, in particular, the use of BDDs to build SPP forms, as done in [4] for SP. On the circuit side, other important logical networks, like the multiplier, should be designed in SPP form to assess the usefulness of the new technique.

Appendix

Proof of Proposition 2. Examine the columns of M for increasing values of the indices. At least m columns must be divided, otherwise the rows of M could not be all distinct. Let C be the set of such columns. Since the rows are sorted as binary numbers, the first column $\mathbf{c}$ of C is $(m-1)$-canonical, hence, $\mathbf{c} = \mathbf{c}_{i_0}$. Inductively assume that, among the first h columns of C, we have discovered the first j canonical columns $\mathbf{c}_{i_0}, \ldots, \mathbf{c}_{i_{j-1}}$, $j \leq h < m$. This implies that the first h columns of C are all $(m-s)$-normal (or canonical), $1 \leq s \leq j-2$. Therefore, there must be at least one column among the remaining columns of C that is k-normal, $k < m-j-2$, otherwise, the rows of M cannot be all distinct. The leftmost column $\mathbf{c}$ with such a property is $(m-j-3)$-canonical by the structure of the binary numbers. That is, $\mathbf{c}$ is the $(j+1)$th canonical column $\mathbf{c}_{i_j}$. □

Row permutations that preserve normality. For $m \geq 2$ and an arbitrary $k \leq m-2$, let the row indices of a normal matrix M be divided in 2^{m-k} consecutive groups $g_0, g_1, \ldots, g_{2^{m-k}-1}$, each containing 2^k consecutive indices. Divide the groups in consecutive quartets, each composed of $g_{4i}, g_{4i+1}, g_{4i+2}, g_{4i+3}$, $0 \leq i \leq 2^{m-k-2}-1$. The following permutations of rows, indicated by permutations of the corresponding indices, preserve the normality of any column, hence, the normality of M:

PERM1: $\forall i$, exchange g_{4i} with g_{4i+1};
PERM2: $\forall i$, exchange g_{4i+1} with g_{4i+2};
PERM3: $\forall i$, exchange g_{4i+1} with g_{4i+3}.

Proof of Proposition 3. We prove that M is balanced, by induction on m. This is trivial for $m = 0$ and $m = 1$. Let the property hold for $m-1$, $m > 1$. Recall that all the columns of a normal matrix are balanced. Taking an arbitrary divided column $\mathbf{c}_j$, we execute proper row permutations to get $\mathbf{c}_j = \mathbf{01}$, preserving the normality of M. If $\mathbf{c}_j$ is $(m-1)$-normal and $\mathbf{c}_j = \mathbf{01}$, we are done. If $\mathbf{c}_j = \mathbf{10}$, permute the upper and lower 2^{m-1} rows of M to get $\mathbf{c}_j = \mathbf{01}$. This operation obviously preserves the normality of M. If $\mathbf{c}_j$ is $(k < m-1)$-normal, we have $\mathbf{c}_j = \mathbf{v}_0\mathbf{v}_1 \ldots \mathbf{v}_{2^{m-k}-1}$, with each subcolumn $\mathbf{v}_i$ consisting of 2^k equal elements. Consider the starting quartet $\mathbf{v}_0\mathbf{v}_1\mathbf{v}_2\mathbf{v}_3$ of subcolumns. If $\mathbf{v}_0 = \mathbf{1}$ (hence, $\mathbf{v}_1 = \mathbf{0}$), apply PERM1. We now have: $\mathbf{v}_0 = \mathbf{0}$, $\mathbf{v}_1 = \mathbf{1}$. If $\mathbf{v}_2 = \mathbf{0}$ (hence, $\mathbf{v}_3 = \mathbf{1}$), apply PERM2; otherwise, $\mathbf{v}_2 = \mathbf{1}$, $\mathbf{v}_3 = \mathbf{0}$, apply PERM3. We now have: $\mathbf{v}_0\mathbf{v}_1 = \mathbf{00}$, $\mathbf{v}_2\mathbf{v}_3 = \mathbf{11}$, and all the other quartets have been similarly permuted. $\mathbf{c}_j$ has then become $(k+1)$-normal, and the construction repeats until $\mathbf{c}_j$ becomes $(m-1)$-normal. We then have $\mathbf{c}_j = \mathbf{01}$. Since the performed permutations have preserved the normality of the matrix, M_j and $M_{\bar{j}}$ are also normal, hence, they are balanced by induction. The above construction can be repeated for any divided column $\mathbf{c}_j$, hence, M is balanced. □

Proof of Proposition 4. Let R be the matrix obtained by rearranging the rows of M. We must prove that R is normal. For $0 \leq m \leq 2$, this holds because any balanced column of one, two, or four elements is normal, as can be easily verified by inspection of all cases. For $m > 2$, scan the columns of R from left to right, to find the first three columns $\mathbf{c}_r, \mathbf{c}_s, \mathbf{c}_t$ such that $\mathbf{c}_r = \mathbf{01}$, $\mathbf{c}_s = \mathbf{0101}$, and $\mathbf{c}_t = \mathbf{01010101}$. It can be easily verified that these columns exist, since R is balanced and sorted. We now prove, by induction on m, that all the other columns $\mathbf{c}_j$, $j \neq r, s, t$, are normal. The basis $m \leq 2$ has been proven above. For $m > 2$, we inductively assume that any balanced matrix of 2^h rows arranged in increasing order is normal, for $h \leq m-1$. In particular, this occurs for all the restrictions of R. Denote by $(\mathbf{c}_j)_k$ the restriction of $\mathbf{c}_j$ to R_k. Use the notation $\tilde{\mathbf{u}}$ to indicate that a vector $\mathbf{u}$ is equal to itself or to its complement and that the same choice is maintained within an expression (note the difference with $\hat{\mathbf{u}}$ that may indicate $\mathbf{u}$ and $\bar{\mathbf{u}}$ in two occurrences inside the same expression). By the inductive hypothesis, we have $(\mathbf{c}_j)_{\bar{r}} = \mathbf{u}\tilde{\mathbf{u}}$, where $(\mathbf{c}_j)_{\bar{r},\bar{s}} = \mathbf{u} = \mathbf{v}\mathbf{w}$ (with $\mathbf{v} = \mathbf{w}$ or $\mathbf{v} = \bar{\mathbf{w}}$). We now prove that $\mathbf{c}_j$ is normal. One of the following two cases must be inductively verified:

1. $(\mathbf{c}_j)_{\bar{s}} = (\mathbf{c}_j)_{\bar{r},\bar{s}}(\mathbf{c}_j)_{r,\bar{s}} = \mathbf{u}\mathbf{u}$. This implies:

   $$(\mathbf{c}_j)_{r,\bar{s},\bar{t}} = (\mathbf{c}_j)_{\bar{r},\bar{s},\bar{t}} = \mathbf{v},$$

   hence, $(\mathbf{c}_j)_{\bar{t}} = \mathbf{v}\tilde{\mathbf{v}}\mathbf{v}\mathbf{x} = \mathbf{v}\tilde{\mathbf{v}}\mathbf{v}\tilde{\mathbf{v}}$, hence,

   $$(\mathbf{c}_j)_r = (\mathbf{c}_j)_{r,\bar{s}}(\mathbf{c}_j)_{r,s,\bar{t}}(\mathbf{c}_j)_{r,s,t} = \mathbf{u}\tilde{\mathbf{v}}\mathbf{y} = \mathbf{u}\tilde{\mathbf{v}}\tilde{\mathbf{w}} = \mathbf{u}\tilde{\mathbf{u}}.$$

   Therefore, $\mathbf{c}_j = (\mathbf{c}_j)_{\bar{r}}(\mathbf{c}_j)_r = \mathbf{u}\tilde{\mathbf{u}}\mathbf{u}\tilde{\mathbf{u}} = \mathbf{z}\mathbf{z}$ normal.

2. $(\mathbf{c}_j)_{\bar{s}} = (\mathbf{c}_j)_{\bar{r},\bar{s}}(\mathbf{c}_j)_{r,\bar{s}} = \mathbf{u}\bar{\mathbf{u}}$. This implies: $(\mathbf{c}_j)_{r,\bar{s},\bar{t}} = \bar{\mathbf{v}}$, hence,

   $$(\mathbf{c}_j)_{\bar{t}} = \mathbf{v}\tilde{\mathbf{v}}\bar{\mathbf{v}}\mathbf{x} = \mathbf{v}\tilde{\mathbf{v}}\bar{\mathbf{v}}\bar{\tilde{\mathbf{v}}},$$

   hence,

   $$(\mathbf{c}_j)_r = (\mathbf{c}_j)_{r,\bar{s}}(\mathbf{c}_j)_{r,s,\bar{t}}(\mathbf{c}_j)_{r,s,t} = \bar{\mathbf{u}}\bar{\tilde{\mathbf{v}}}\mathbf{y} = \bar{\mathbf{u}}\bar{\tilde{\mathbf{v}}}\bar{\tilde{\mathbf{w}}} = \bar{\mathbf{u}}\bar{\tilde{\mathbf{u}}}.$$

   Therefore, $\mathbf{c}_j = (\mathbf{c}_j)_{\bar{r}}(c_j)_r = \mathbf{u}\tilde{\mathbf{u}}\bar{\mathbf{u}}\bar{\tilde{\mathbf{u}}} = \mathbf{z}\bar{\mathbf{z}}$ normal. □

ACKNOWLEDGMENTS

We are in debt to our student Luigi Palpacelli who carried out the benchmark experimentation. This work was supported in part by MURST of Italy.

REFERENCES

[1] S.B. Akers, "Binary Decision Diagrams," *IEEE Trans. Computers,* vol. 27, no. 6, pp. 509-516, June 1978.

[2] R.A. Brualdi and H.J. Ryser, *Combinatorial Matrix Theory.* Cambridge: Cambridge Univ. Press, 1991.

[3] R.E. Bryant, "Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams," *ACM Computing Surveys,* vol. 24, no. 3, pp. 293-318, 1992.

[4] O. Coudert, "Doing Two Level Logic Minimization 100 Times Faster," *Proc. ACM-SIAM Symp. Discrete Algorithms,* pp. 112-121, San Francisco, 1995.

[5] M. Davio, J.P. Dechamps and A. Thayse, *Discrete and Switching Functions.* New York: McGraw-Hill, 1978.

[6] M.R. Garey and D.S. Johnson, *Computers and Intractability.* San Francisco: Freeman, 1979.

[7] Z. Kohavi, *Switching and Finite Automata Theory.* New York: McGraw-Hill, 1970.

[8] H.C. Lai and S. Muroga, "Logic Networks with Minimum Number of NOR (NAND) Gates for Parity Functions of n Variables," *IEEE Trans. Computers,* vol. 36, no. 2, pp. 157-166, Feb. 1987.

[9] G. Lakuhani, "Minimization of Switching Functions for a Multi-Level EXOR Realization," *Proc. IFIP WG 10.5 Workshop Applications of the Reed-Muller Expansion in Circuit Design,* pp. 185-190, 1995.

[10] F. Luccio and L. Pagli, "Normal Matrices, Pseudo-Cubes and Pseudo-Products," *Congressus Numerantium,* vol. 127, pp. 33-56, 1997.

[11] P.C. McGeer, J. Sanghavi, R.K. Brayton and A.L. Sangiovanni-Vincentelli, "ESPRESSO-SIGNATURE: A New Exact Minimizer for Logic Functions," *IEEE Trans. VLSI,* vol. 1, no. 4, pp. 432-440, 1993.

[12] D.E. Muller, "Application of Boolean Algebra to Switching Circuit Design and to Error Detection," *IRE Trans. Electronic Computers,* vol. 3, pp. 6-12, 1954.

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

[14] T. Sasao, "Logic Synthesis with EXOR Logic Gates," *Logic Synthesis and Optimization,* T. Sasao, ed. Kluwer Academic, 1993.

[15] T. Sasao, "EXMIN2: A Simplification Algorithm for Exclusive-OR-Sum-of-Products Expressions for Multiple-Valued Input Two-Valued Output Functions," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* vol. 12, pp. 621-632, 1993.

[16] T. Sasao, "Input Variable Assignment and Output Phase Optimization of PLA's," *IEEE Trans. Computers,* vol. 33, no. 10, pp. 879-894, Oct. 1984.

[17] T. Sasao, "Representation of Logic Functions Using EXOR Operators," *Representation of Discrete Functions,* T. Sasao and M. Fujita, eds. Kluwer Academic, 1996.

[18] A. Tran, "Graphical Method for the Convertion of Minterms to Reed-Muller Coefficients and the Minimization of Exclusive-OR Switching Functions," *IEE Proc. Part E,* vol. 134, pp. 93-99, 1987.

[19] C.C. Tsai and M. Marek-Sadovska, "Generalized Reed-Muller Forms as a Tool to Detect Symmetries," *IEEE Trans. Computers,* vol. 45, no. 1, pp. 33-40, Jan. 1996.

[20] C.C. Tsai and M. Marek-Sadovska, "Boolean Function Classification via Fixed Polarity Reed-Muller Forms," *IEEE Trans. Computers,* vol. 46, no. 2, pp. 173-186, Feb. 1997.

[21] S. Yang, *Logic Synthesis and Optimization Benchmarks,* User Guide. Microelectronic Center of North Carolina, 1991. Benchmarks available at: ftp://ftp.mcnc.org/pub/benchmark/Benchmark-dirs/LGSynth93/.

[22] K. Yano, Y. Sasaki, K. Rikino, and K. Seki, "Top-Down Pass Transistor Logic Design," *IEEE J. Solid State and Circuits,* vol. 31, no. 6, pp. 792-803, 1996.

[23] K. Yasuoka, "A Generation Method for Exor-Sum-of-Products Expressions Using Shared Binary Decision Diagrams," *Logic Synthesis and Optimization,* T. Saso, ed. Kluwer Academic, 1993.

Fabrizio Luccio received the Dr.Ing. degree in electrical engineering from the Politecnico di Milano, Italy, in 1962, and the Libera Docenza in electronic computers from the Italian university system in 1968. He is currently a professor of computer science at the University of Pisa, Italy. After an industrial experience with Olivetti, he joined the Politecnico di Milano, starting his research activity in logical design and in programming language translation. In 1966, he moved to M.I.T. as a staff member at Project MAC, working in compiling techniques. He then became a professor at the University of Southern California and, then, at New York University, pursuing research in theoretical and algorithmic aspects of logical network synthesis. In 1971, he permanently returned to Italy as a lecturer and, later, a professor of informatics at the University of Pisa. He spent several sabbatical periods as a visiting professor at UCLA, the University of Illinois, the National University of Singapore, the University of Hawaii, and Carleton University in Ottawa, Canada. He has also been a visiting scientist at the IBM T.J. Watson Research Center and at the NTT LSI Laboratories in Morinosato, Japan. His current research interests are in algorithm design and in the relationship between abstract computational models and realistic computers and circuits. Professor Luccio is a fellow of the IEEE and a member of the ACM.

Linda Pagli received the Laurea in information sciences from the University of Pisa, Italy, in 1973. She is currently a professor of computer science at the University of Pisa. After receiving her degree, she was associated as a researcher with the Department of Informatics of the University of Pisa, starting her research activity in data structures and, then, in VLSI computation. In 1987, she was appointed a professor of computer science at the University of Salerno, Italy, returning to Pisa three years later. She has been a visiting professor at Carleton University in Ottawa, Canada. Her current research interests are in the bases of computation and in design and analysis of sequential and parallel algorithms. Professor Pagli has pursued intense activities in higher education in favor of developing countries. In this framework, she has been a professor at the National University of Somalia for an extensive period of time. She is a member of the ACM.