TWMS J. App. Eng. Math. V.10, N.2, 2020, pp. 459-470

NOVEL TECHNIQUE FOR DISJOINTED SUM OF PRODUCTS

YAVUZ CAN, §

ABSTRACT. A classical problem of Boolean theory is to derive a disjointed Sum of Products. This work introduces a novel approach for converting Sum of Products into disjointed Sum of Products which is based on a novel, generally valid, combining technique of 'orthogonalizing difference-building $\ominus$'. Postulates and rules for this linking technique are defined which have to be considered getting correct results. The benefit of the novel approach is that the result contains fewer number of product terms which has significant advantages for further calculations as the Boolean Differential Calculus.

Keywords: Sum of products, disjointed sum of products, minimization, disjointed exclusive-or sum of products, combinational circuit.

AMS Subject Classification: 03B99, 03G05, 03G25, 94C10

1. INTRODUCTION

The calculation of test patterns, which are used to detect feasible faults in combinational circuits regarding the $stuck-at\ fault\ model$, are done by Boolean Differential Calculus (BDC) [3–5, 18]. This procedure can easily be performed in the Quaternary-Vector-List (QVL) arithmetic [10], which has benefits in terms of computation time and memory usage. QVL is a kind of matrix representation of Boolean functions and also the extended form of Ternary-Vector-List (TVL) [5, 6, 9, 12, 13, 15]. Faults caused by defective transistors of a gate, broken or shorted wires between gates affects the output of a logic circuit. Each changing behavior of the output can be described by the BDC. Firstly, the corresponding Boolean expression of the combinational circuit is transformed ($TRANS$) in an equivalent Sum of Products (SOP) in QVL-arithmetic as shown in Figure 1. Next this SOP is orthogonalized ($ORTH\_DF$). Thus, we get a disjointed Sum of Products (dSOP) which is subsequently transformed into a disjointed Exclusive-Or Sum of Products (dESOP) [4, 6, 8, 14, 20]. Now, the BDC is performed at this dESOP. After another orthogonalization step ($ORTH\_AF$) of the result of BDC it is transformed into a dSOP. Finally, this dSOP contains all possible test patterns. Additionally, the number of containing product terms after each activity is given around the flow diagram in Figure 1. The number of the product terms increases with each activity that effects the number of test patterns. The

Institute for Electronics Engineering, Friedrich-Alexander-University Erlangen-Nuremberg; 91058 Erlangen, Germany.
e-mail: yavuz.can@fau.de; ORCID: https://orcid.org/0000-0001-6052-2351.

§ Manuscript received: August 6, 2018; accepted: November 20, 2018.

noval approach of orthogonalization reduces the number of product terms N_{orth} which consequently reduces the number of test patterns in the end of the calculation step.

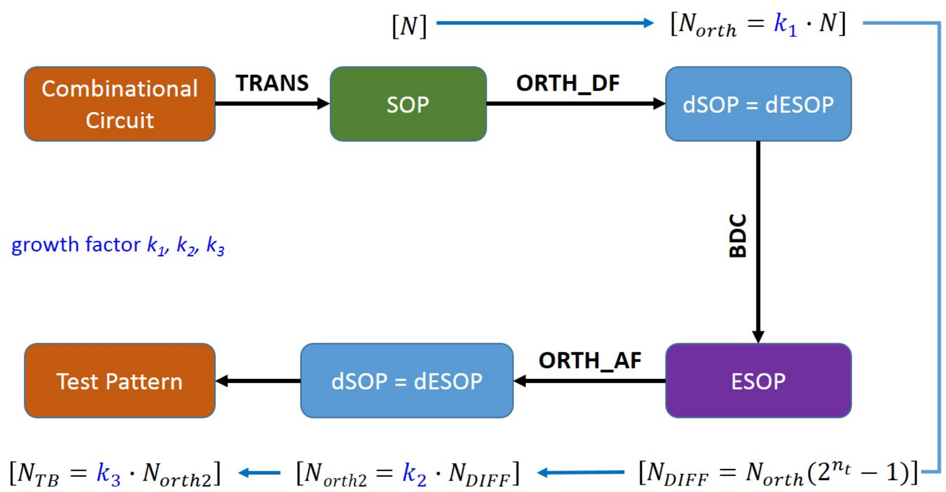


FIGURE 1. Calculation steps for test patterns

2. PRELIMENARIES

A Boolean function is defined as the mapping $f(\underline{x}) : \{0,1\}^n \to \{0,1\}$. There exists four normal forms of Boolean functions [4, 19]: disjunctive normal form DNF, conjunctive normal form CNF, antivalence normal form ANF and equivalence normal form ENF which consist of either product terms

$$p_i(\underline{x}) = \bigwedge_{i=1}^{n} = x_1 \wedge .. \wedge x_n \quad \text{with} \quad n \in \mathbb{N} \tag{1}$$

or clauses

$$c_i(\underline{x}) = \bigvee_{i=1}^{n} = x_1 \vee .. \vee x_n \quad \text{with} \quad n \in \mathbb{N} \tag{2}$$

(with $n \geq 1$ as the lenght of the variables; dimension) in which variables are either negated $\bar{x}_i$ or not-negated x_i. The aversion to them is their large number of literals. By using the associated form of a normal form this aversion can be reduced to disjunctive form DF, conjunctive form CF, antivalence form AF and equivalence form EF. The disjunctive form DF [4, 19] is also considered as Sum of Products SOP (Eq. (3)) and the antivalence form AF [4, 19] as an Exclusive-Or Sum of Products ESOP (Eq. (4)). With $N > 1$ as the number of product terms [2, 17] it applies:

$$SOP(\underline{x}) = \bigvee_{i=1}^{N} p_i(\underline{x}) \tag{3}$$

$$ESOP(\underline{x}) = \bigoplus_{i=1}^{N} p_i(\underline{x}) \tag{4}$$

The orthogonality is a special attribute of Boolean functions. A SOP or ESOP is orthogonal if its product terms are disjointed to one another in pairs at least in one variable by x_i

and $\bar{x}_i$ [5, 6, 11]. Consequently, these product terms have no common covering after their logical conjunction, $(p_i(\underline{x}) \wedge p_j(\underline{x}) = 0)$. The orthogonal representation of a disjointed Sum of Products $dSOP(\underline{x})$ [1] is equal to the orthogonal form of a disjointed Exclusive-Or Sum of Products $dESOP(\underline{x})$. It applies $dSOP(\underline{x}) = dESOP(\underline{x})$ [5, 6, 11, 14]. That means, that dSOP is equivalent to dESOP containing the same product terms. This relationship can be explained well with the following definition in [20], if both product terms $p_i(\underline{x})$ and $p_j(\underline{x})$ are disjointed to each other:

$$p_i(\underline{x}) \vee p_j(\underline{x}) = p_i(\underline{x}) \oplus p_j(\underline{x}) \oplus \underbrace{(p_i(\underline{x}) \wedge p_j(\underline{x}))}_{=0} \tag{5}$$

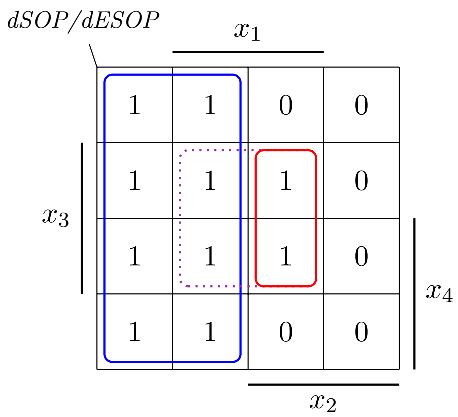
With this relation a SOP of two product terms can be transformed in an ESOP of these both product terms. For the special case of orthogonality the conjunction of both products terms results to 0. Thus, because of $x_i \oplus 0 = x_i$, it follows:

$$p_i(\underline{x}) \vee p_j(\underline{x}) = p_i(\underline{x}) \oplus p_j(\underline{x})$$

In this case, the left side is equal to the right side. That means, that a dSOP is equivalent to dESOP, $dSOP(\underline{x}) = dESOP(\underline{x})$.

$$\bigvee_{i=1}^{N} p_i(\underline{x}) = \bigoplus_{i=1}^{N} p_i(\underline{x})$$

Example 2.1. *It is given a* $dSOP(\underline{x}) = \bar{x}_2 \vee x_1x_2x_3$ *and a* $dESOP(\underline{x}) = \bar{x}_2 \oplus x_1x_2x_3$ *which are visualized in the following K-map and are characterized by non-overlapping cubes.*



Special calculations can be more easily solved in another form. For example, building the complement of a SOP is a complex procedure. However, by orthogonalization of SOP the dSOP is transformed in a dESOP at which the complement by linking with $\oplus 1$ is subsequently determined, it follows $\overline{dESOP}$. In a further step of orthogonalization of $\overline{dESOP}$ the **re**-transforming back to dSOP is gained. Thus, the complement $\overline{dSOP}$ is calculated. Since an orthogonal function can be transformed in another form, it simplifies the handling for further calculations in applications of electrical engineering, e.g. calculation of suitable test patterns for combinational circuits for verifying feasible logical faults which can be mathematically determined by Boolean Differential Calculus (BDC) [3–5, 18].

3. Methodology of Orthogonalization

3.1. **Orthogonalizing Difference-Building.** Orthogonalizing difference-building $\ominus$ is the composition of two calculation steps - the usual difference-building out of the set theory and the subsequent orthogonalization as shown in Figure 2. This method $\ominus$ is generally valid and equivalent to the usual method of difference-building [6]. The orthogonalizing difference-building $p_m(\underline{x}) \ominus p_s(\underline{x})$ between a minuend product term $p_m(\underline{x})$ and a subtrahend product term $p_s(\underline{x})$ corresponds to the removal of the intersection, which is formed of both product terms, from the minuend product term $p_m(\underline{x})$, which means $p_m(\underline{x}) - (p_m(\underline{x}) \wedge p_s(\underline{x}))$. However, the result consists of several product terms which are pairwise disjointed to each other. Equation (6) applies; with $n, n' \in \mathbb{N}$ as the dimension or the number of variables of $p_m(\underline{x})$ and $p_s(\underline{x})$. This equation specifies the order in which the variables of the given product terms has to be linked.

$$p_m(\underline{x}) \ominus p_s(\underline{x}) = \bigwedge_{m=1}^{n} x_m \ominus \bigwedge_{s=1}^{n'} x_s := \bigwedge_{m=1}^{n} x_m \wedge \bigvee_{s=1}^{n'_j} \bar{x}_{s_j} = \tag{6}$$
$$= \big(x_1 \cdot \ldots \cdot x_{n-1} x_n\big)_m \wedge \big(\bar{x}_{1_j} \vee x_{1_j}\bar{x}_{2_j} \vee \ldots \vee x_{1_j} \cdot \ldots \cdot x_{(n'-1)_j}\bar{x}_{n'_j}\big)_s$$

In this case, the formula

$$\bigvee_{i=1}^{n'_j} \bar{x}_i = \bar{x}_{1_j} \vee x_{1_j}\bar{x}_{2_j} \vee \ldots \vee x_{1_j}x_{2_j} \cdot \ldots \cdot \bar{x}_{n'_j} \tag{7}$$

out of [11] is used to describe the orthogonalizing difference-building in a mathematically easier way. The method of orthogonalizing difference-building $\ominus$ is demonstrated by the following Example 3.1.

Example 3.1. *A subtrahend $p_s(\underline{x}) = x_2x_3x_4$ is subtracted from a minuend $p_m(\underline{x}) = x_1$ by the method of $\ominus$.*

$$x_1 \ominus x_2x_3x_4 = x_1\bar{x}_2 \vee x_1x_2\bar{x}_3 \vee x_1x_2x_3\bar{x}_4$$

The following points explain the application of Eq. (6) by using that example:

- The first literal of the subtrahend, here x_2, is taken complement and build the intersection with the minuend, here x_1. Consequently, the first term in the result is $x_1\bar{x}_2$.
- Then the second literal, here x_3, is taken complement and build the intersection with the minuend and the first literal x_2 of the subtrahend. Therefore, the second term is $x_1x_2\bar{x}_3$.
- Following the next literal, here x_4, is taken complement and build the intersection with the minuend and the first literal x_2 and second literal x_3 of the subtrahend. Thus, the third term of the difference is $x_1x_2x_3\bar{x}_4$.
- This process is continued until all literals of the subtrahend are singly complemented and linked by building the intersection with the minuend in a separate term.

Following rules that must be followed to get correct results for the application of $\ominus$:

1. If the subtrahend is already orthogonal to the minuend ($p_s(\underline{x}) \perp p_m(\underline{x})$) the result corresponds to the minuend :

$$p_m(\underline{x}) \ominus p_s(\underline{x}) = p_m(\underline{x}) \mid_{p_s(\underline{x}) \perp p_m(\underline{x})} \tag{8}$$

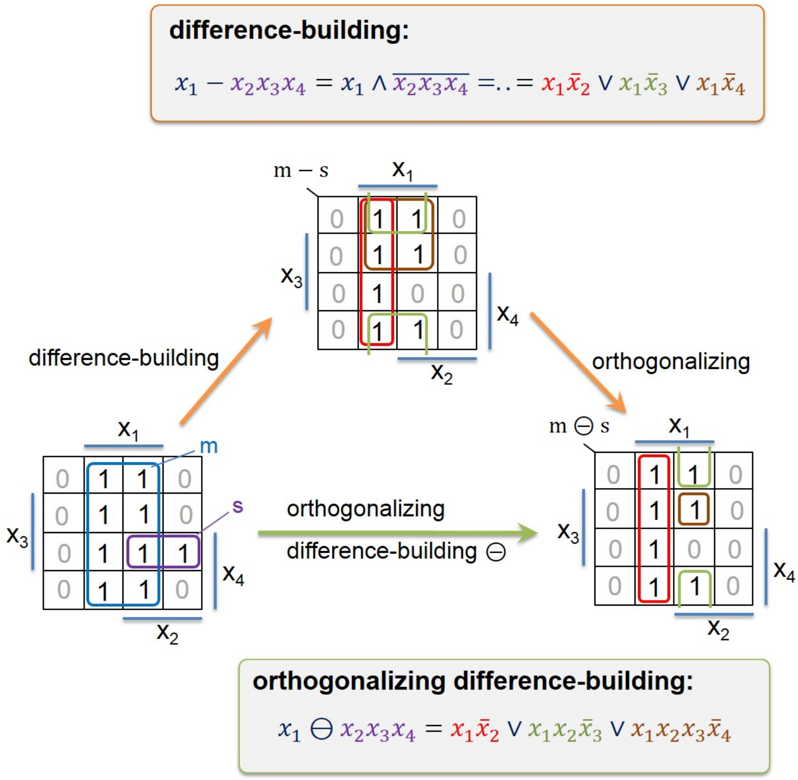


FIGURE 2. ⊖: Two procedures in one step

Example 3.2. *A subtrahend* $p_s(\underline{x}) = x_1x_2$ *is subtracted from a minuend* $p_m(\underline{x}) = \bar{x}_2x_3$ *whereby there are already orthogonal.*

$$\bar{x}_2x_3 \ominus x_1x_2 = \bar{x}_1\bar{x}_2x_3 \vee x_1\bar{x}_2x_3 = \bar{x}_2x_3\underbrace{(\bar{x}_1 \vee x_1)}_{=1} = \bar{x}_2x_3$$

2. The orthogonal difference between 0 and the subtrahend is the subtrahend itself:

$$0 \ominus p_s(\underline{x}) = p_s(\underline{x}) \tag{9}$$

Example 3.3. *A subtrahend* $p_s(\underline{x}) = x_1x_2$ *is subtracted from 0.*

$$0 \ominus x_1x_2 = x_1x_2$$

3. The result between 1 and subtrahend is the complementary of the subtrahend which results in a disjointed Sum of Products:

$$1 \ominus p_s(\underline{x}) = dSOP(\underline{x}) \mid_{\overline{p_s(\underline{x})}} \tag{10}$$

Example 3.4. *A subtrahend* $p_s(\underline{x}) = x_1x_2$ *is subtracted from 1.*

$$1 \ominus x_1x_2 = \bar{x}_1 \vee x_1\bar{x}_2$$

4. Thereby, the symbol of subset $\subseteq$ out of the set theory is transferred to the switching algebra. The result between subtrahend and minuend is empty if the minuend product term is already completely contained in the subtrahend product term $(p_m(\underline{x}) \subseteq p_s(\underline{x}))$:

$$p_m(\underline{x}) \ominus p_s(\underline{x}) = 0 \mid_{p_m(\underline{x}) \subseteq p_s(\underline{x})} \tag{11}$$

Example 3.5. *A subtrahend* $p_s(\underline{x}) = x_1$ *is subtracted from a minuend* $p_m(\underline{x}) = x_1\bar{x}_2x_3$.

$$x_1\bar{x}_2x_3 \ominus x_1 = 0$$

Two calculation procedures - difference-building and the subsequent orthogonalization - can be performed in one step by the use of this new method $\ominus$ as shown in Figure 2. The result out of $\ominus$ is orthogonal in contrast to the result out of the method difference-building. However, both results are different in their representations but homogenous in their covering of 1s. They only differ in their form of coverage, whereas the novel technique constitutes the solution in already orthogonal form.

3.2. **Distributivity of $\ominus$.** The distributive property of an operation allows the exclusion of the same term. That means, that a term can be factored out. In this case, it applies the distributive law for $\ominus$ for left and right side.

$$p_1(\underline{x}) \cdot \big(p_2(\underline{x}) \ominus p_3(\underline{x})\big) = \big(p_1(\underline{x}) \cdot p_2(\underline{x})\big) \ominus \big(p_1(\underline{x}) \cdot p_3(\underline{x})\big) \tag{12}$$

The validity of the distributive property is given by the following proof:

$$\begin{aligned}
p_1(\underline{x}) \cdot \big(p_2(\underline{x}) - p_2(\underline{x})p_3(\underline{x})\big) &= p_1(\underline{x}) \cdot p_2(\underline{x}) - p_1(\underline{x}) \cdot p_2(\underline{x}) \cdot p_3(\underline{x}) \\
p_1(\underline{x}) \cdot \big(p_2(\underline{x}) \wedge (\overline{p_2(\underline{x})p_3(\underline{x})})\big) &= p_1(\underline{x}) \cdot p_2(\underline{x}) \wedge \big(\overline{p_1(\underline{x}) \cdot p_2(\underline{x}) \cdot p_3(\underline{x})}\big) \\
p_1(\underline{x}) \cdot \big(p_2(\underline{x}) \wedge (\overline{p_2(\underline{x})} \vee \overline{p_3(\underline{x})})\big) &= p_1(\underline{x}) \cdot p_2(\underline{x}) \wedge \big(\overline{p_1(\underline{x})} \vee \overline{p_2(\underline{x})} \vee \overline{p_3(\underline{x})}\big) \\
p_1(\underline{x}) \cdot \big(p_2(\underline{x}) \cdot \overline{p_3(\underline{x})}\big) &= p_1(\underline{x}) \cdot p_2(\underline{x}) \cdot \overline{p_3(\underline{x})} \\
p_1(\underline{x}) \cdot p_2(\underline{x}) \cdot \overline{p_3(\underline{x})} &= p_1(\underline{x}) \cdot p_2(\underline{x}) \cdot \overline{p_3(\underline{x})}
\end{aligned}$$

Both sides are equivalent. This charasteristic of distributivity is demonstrated by the following Example 3.6 whereby both sides result to the same term.

Example 3.6.

$$\begin{aligned}
x_1 \cdot (x_2\bar{x}_3 \ominus \bar{x}_1x_2) &= (x_1 \cdot x_2\bar{x}_3) \ominus (x_1 \cdot \bar{x}_1x_2) \\
x_1 \cdot (x_1x_2\bar{x}_3) &= x_1x_2\bar{x}_3 \\
x_1x_2\bar{x}_3 &= x_1x_2\bar{x}_3
\end{aligned}$$

3.3. **Orthogonalization of SOP.** The orthogonalization of every $SOP(\underline{x})$ consisting of at least two product terms ($N > 1$) can be performed by Eq. (13) which is based on the combination technique of $\ominus$ [6]. By the mathematical induction the general validity of Eq.(13) is given in Proof 1.

$$\begin{aligned}
dSOP(\underline{x}) := \bigvee_{k=0}^{N-1} \Bigg(\bigominus_{i=k+1}^{N} p_i(\underline{x}) \Bigg) = \\
= \big(p_1(\underline{x}) \ominus p_2(\underline{x}) \ominus \ldots \ominus p_N(\underline{x})\big) \vee \ldots \vee \big(p_{N-1}(\underline{x}) \ominus p_N(\underline{x})\big) \vee p_N(\underline{x})
\end{aligned} \tag{13}$$

Proof. 1

- $\forall N \in \mathbb{N}, \quad N \geq N_0$ applies $A(N)$:

$$\bigvee_{k=0}^{N-1} \Bigg(\bigominus_{i=k+1}^{N} p_i(\underline{x}) \Bigg) = \big(p_1(\underline{x}) \ominus p_2(\underline{x}) \ominus \ldots \ominus p_N(\underline{x})\big) \vee \ldots \vee \big(p_{N-1}(\underline{x}) \ominus p_N(\underline{x})\big) \vee p_N(\underline{x})$$

- if $A(N_0) \wedge \big(\forall N \in \mathbb{N},\ N \geq N_0 : A(N) \rightarrow A(N+1)\big) \Rightarrow \forall N \in \mathbb{N},\ N \geq N_0 : A(N)$

  Basis $A(N_0)$: $N_0 = 3$

$$\bigvee_{k=0}^{3-1}\left(\mathop{\ominus}_{i=k+1}^{3} p_i(\underline{x})\right) = \big(p_1(\underline{x}) \ominus p_2(\underline{x}) \ominus p_3(\underline{x})\big) \vee \big(p_2(\underline{x}) \ominus p_3(\underline{x})\big) \vee p_3(\underline{x})$$

  Inductive step $A(N) \rightarrow A(N+1)$: $N \rightarrow N+1$

$$\bigvee_{k=0}^{(N+1)-1}\left(\mathop{\ominus}_{i=k+1}^{N+1} p_i(\underline{x})\right) = \big(p_1(\underline{x}) \ominus \ldots \ominus p_N(\underline{x}) \ominus p_{N+1}(\underline{x})\big) \vee \ldots \vee \big(p_N(\underline{x}) \ominus p_{N+1}(\underline{x})\big) \vee p_{N+1}(\underline{x})$$

$$\bigvee_{k=0}^{N}\left(\mathop{\ominus}_{i=k+1}^{N+1} p_i(\underline{x})\right) = \big(p_1(\underline{x}) \ominus \ldots \ominus p_N(\underline{x}) \ominus p_{N+1}(\underline{x})\big) \vee \ldots \vee \big(p_N(\underline{x}) \ominus p_{N+1}(\underline{x})\big) \vee p_{N+1}(\underline{x})$$

$$\bigvee_{k=0}^{N}\left(\mathop{\ominus}_{i=k+1}^{N} p_i(\underline{x}) \ominus p_{N+1}(\underline{x})\right) = \big(p_1(\underline{x}) \ominus \ldots \ominus p_N(\underline{x}) \ominus p_{N+1}(\underline{x})\big) \vee \ldots \vee \big(p_N(\underline{x}) \ominus p_{N+1}(\underline{x})\big) \vee p_{N+1}(\underline{x})$$

- Analysis for N=2:

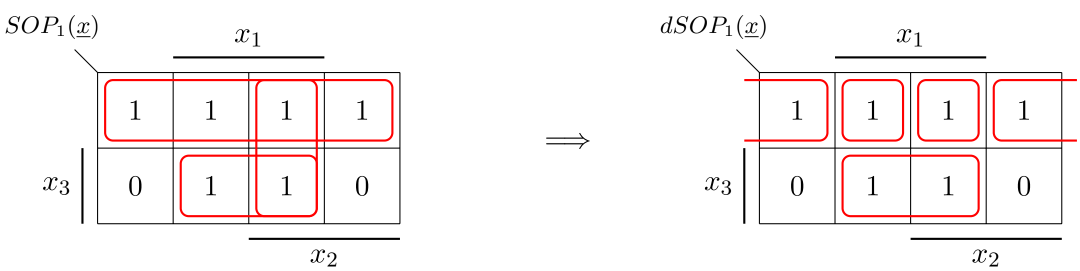
$$\bigvee_{k=0}^{2}\left(\mathop{\ominus}_{i=k+1}^{2} p_i(\underline{x}) \ominus p_3(\underline{x})\right) = \big(p_1(\underline{x}) \ominus p_2(\underline{x}) \ominus p_3(\underline{x})\big) \vee \big(p_2(\underline{x}) \ominus p_3(\underline{x})\big) \vee p_3(\underline{x})$$

□

The order of the calculation is important. That means, the first two product terms must be calculated and then the third product term must be calculated with the result of them, and so on. The result of $dSOP(\underline{x})$ can diversify depending on the starting product term. As a SOP has the characteristic of being commutative the order of their product terms can be changed to get results with fewer number of disjointed product terms N_{orth}. Result with fewer number of N_{orth} is often reached by ordering from higher number of variables to fewer number of variables. Following Example 3.7 gives an overview about the procedure orthogonalizing by Eq. (13) and afterwards Example 3.8 with an additional process of sorting.

Example 3.7. *Function $SOP_1(\underline{x}) = \bar{x}_3 \vee x_1x_2 \vee x_1x_3$ has to be orthogonalized by Eq.* (13) *and the result is visualized in a K-map.*

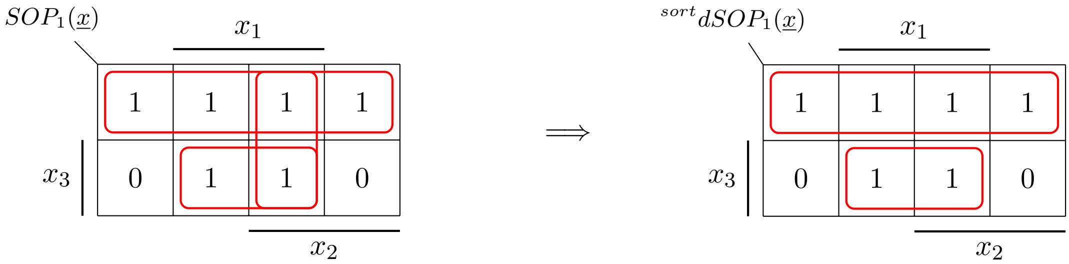
$$dSOP_1(\underline{x}) = \big(\bar{x}_3 \ominus x_1x_2 \ominus x_1x_3\big) \vee \big(x_1x_2 \ominus x_1x_3\big) \vee x_1x_3 =$$
$$= \Big(\underbrace{\big(\bar{x}_1\bar{x}_3 \vee x_1\bar{x}_2\bar{x}_3\big) \ominus x_1x_3}_{Eq.\ (8)}\Big) \vee x_1x_2\bar{x}_3 \vee x_1x_3 = \bar{x}_1\bar{x}_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1x_2\bar{x}_3 \vee x_1x_3$$



Function $dSOP_1(\underline{x})$ consists of four disjointed product terms ($N_{orth} = 4$) and is the orthogonalized form of $SOP_1(\underline{x})$. Both are equivalent. They only differ in their form of coverage which are illustrated in the K-map.

Example 3.8. *Now, the sorted form of* $dSOP_1(\underline{x})$ *here* ${}^{sort}dSOP_1(\underline{x}) = x_1x_2 \vee x_1x_3 \vee \bar{x}_3$ *has to be orthogonalized by Eq.* (13).

$$
\begin{aligned}
{}^{sort}dSOP_1(\underline{x}) &= \left(x_1x_2 \ominus x_1x_3 \ominus \bar{x}_3\right) \vee \big(\underbrace{x_1x_3 \ominus \bar{x}_3}_{Eq.\ (8)}\big) \vee \bar{x}_3 = \\
&= \big(\underbrace{x_1x_2\bar{x}_3 \ominus \bar{x}_3}_{=0\ Eq.\ (11)}\big) \vee x_1x_3 \vee \bar{x}_3 = x_1x_3 \vee \bar{x}_3
\end{aligned}
$$



Function ${}^{sort}dSOP_1(\underline{x})$ is another equivalent orthogonal form of $SOP_1(\underline{x})$ which consists of two disjointed product terms ($N_{orth} = 2$) that is also illustrated in a K-map. 1s are covered by two cubes. By sorting a minimized dSOP can be reached. The results are homogenous. They only differ in their form of coverage. The additional step of sorting brings the benefit of reducing the number of product terms N_{orth} in the result.

4. Comparisons and Measurements

4.1. **Comparison of N_{orth} of dSOP before and after sorting.** However, to make a statement about a optimized form, the optimum minimization would have to be defined, which has not yet been clarified. Table 1 illustrates the percentage reduction of terms by the use of subsequent procedure of sorting. N_{orth} and ${}^{sort}N_{orth}$ in respect to N and x_n - as the length of the given tuple (dimension) - are compared. It applies ${}^{sort}N_{orth}(N, x_n) < N_{orth}(N, x_n)$. That means, that the procedure of sorting brings a benefit for gaining minimized dSOP. Firstly, a list of ten non-orthogonal SOPs in respect to $N = \{5, 10, 15\}$ and dimension $x_n = \{5, 6, .., 49, 50\}$ is created. Consequently, per each N has produced 450 different non-orthogonal SOPs. Subsequently, each SOP was orthogonalized according to the novel approach before and after sorting. The resulting number of product terms N_{orth} in dSOP and ${}^{sort}N_{orth}$ in ${}^{\text{sort}}$dSOP in respect to N and x_n were determined as shown in Figure 3. Out of these values an average value was calculated for each dimension x_n. Thereby, the results of the quotients of the average number of disjointed product terms were obtained. Furthermore, quotients of N_{orth} and ${}^{sort}N_{orth}$ in percentage value were gained as shown in Table 1. These values give the information about the percentage deviation of N_{orth} to ${}^{sort}N_{orth}$. However, in rare cases negative values occur because of the better result due to N_{orth} in comparison to ${}^{sort}N_{orth}$. In those cases, the number of terms are fewer in unsorted cases. Finally, a total average percentage value per each N was determined out of these average values. Consequently, an additional procedure of sorting leads to dSOP with fewer number of product terms. Minimization of approximately 17% till 28% are obtained in comparison to a dSOP which was not sorted before.

4.2. **Comparison of other methods in N_{orth}.** In Figure 4 the number of product terms in the result of dSOP is shown in comparison to the methods $dSOP^{m2}$ out of [16] and $dSOP^{m1}$ out of [4]. In this case, the average number of product terms N_{orth} in respect to N and x_n in dSOP is compared. The average value is formed of 100 calculated tasks for

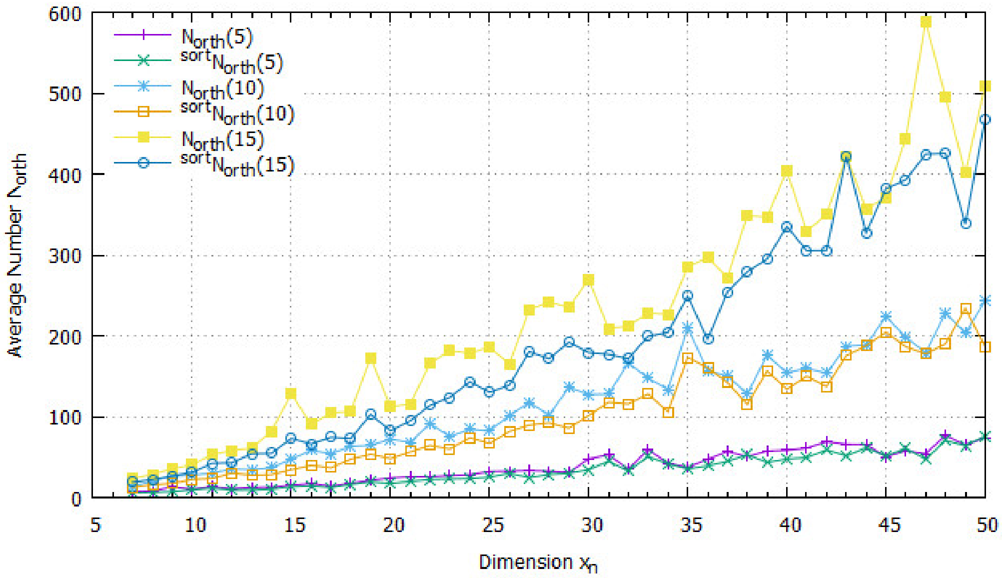


FIGURE 3. Average number of N_{orth} and ${}^{sort}N_{orth}$ in respect to N={5, 10, 15}

	Percentage Value [%]		
x_n	$\frac{N_{orth}}{{}^{sort}N_{orth}}(5)$	$\frac{N_{orth}}{{}^{sort}N_{orth}}(10)$	$\frac{N_{orth}}{{}^{sort}N_{orth}}(15)$
5	0.0	8.6	-
6	23.6	29.0	-
7	28.8	11.8	24.6
8	30.2	27.0	28.7
9	73.3	52.0	32.8
10	6.1	24.6	29.6
⋮	⋮	⋮	⋮
28	13.6	10.5	40.9
29	2.7	58.6	22.3
30	38.6	25.8	50.6
31	17.8	8.5	18.0
⋮	⋮	⋮	⋮
45	-1.6	9.4	-2.8
46	-5.4	6.5	13.1
47	14.8	0.6	38.4
48	8.8	20.2	16.4
49	1.2	-13.1	18.6
50	-2.5	30.6	8.8
average	**17.3**	**21.7**	**27.6**

TABLE 1. Percentage value of $\frac{N_{orth}}{{}^{sort}N_{orth}}$

each dimension. These charts illustrate that the novel method $dSOP^{c1}$ offers better results than the two other methods $dSOP^{m2}$ and $dSOP^{m1}$. The number of product terms N_{orth} in the orthogonalized SOP by the method $dSOP^{c1}$ is fewer than by the methods $dSOP^{m2}$

and $dSOP^{m1}$. It applies $N_{orth}^{dSOP^{c1}} < N_{orth}^{dSOP^{m1,m2}}$. The corresponding average values are partially shown in Table 2. The number of product terms is reduced approximately by a factor of 2 by SOP^{c1}: $N_{orth}^{dSOP^{c1}} \approx \frac{N_{orth}^{dSOP^{m1}}}{1.87}$ and $N_{orth}^{dSOP^{c1}} \approx \frac{N_{orth}^{dSOP^{m2}}}{2.34}$. This attribute is important because a further calculation of dSOP needs fewer operations and is carried out more quickly. That means, the respective computation time is reduced. In this case, a further calculation of a dSOP such as the Boolean Differential Calculus BDC is performed with a fewer number of product terms and thus reduces the number of further operations. By the use of this new method with additional procedure of sorting the size of the set of test patterns will be get smaller. This reduction after $ORTH\_DF$ leads to the operation of BDC at fewer number product terms which consequently will also minimized its results. By a further orthogonalization $ORTH\_AF$ of a reduced result of BDC will acquired to a smaller number of terms in dSOP which provides a set of test patterns that enable a fully verification of a combinational circuit in respect to the $stuck-at\ fault$ model in the final.

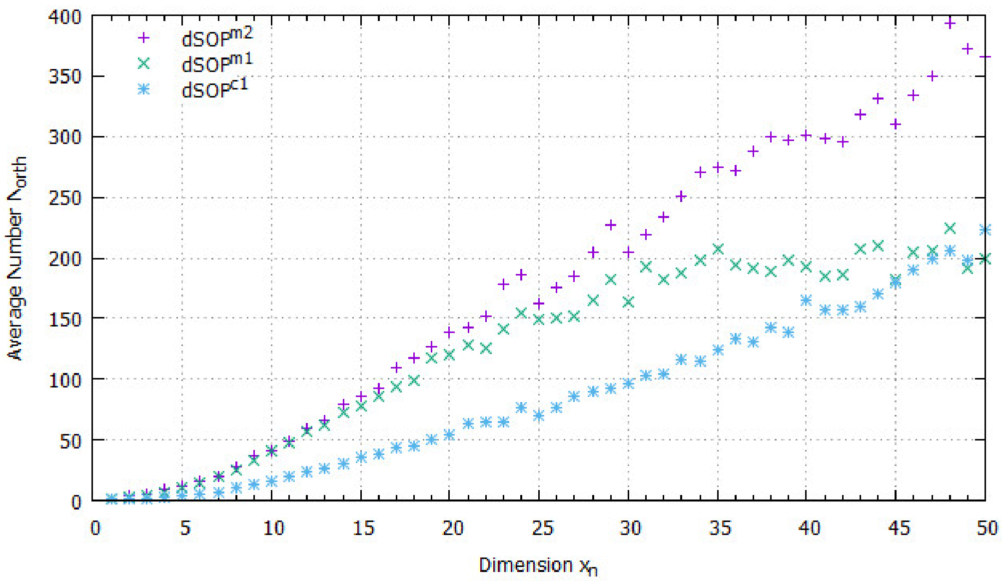


FIGURE 4. Average number of N_{orth} with input size N=10

5. Conclusion

This work shew a novel approach for building a disjointed Sum of Products of a given Sum of Product which is based on the combining technique of 'orthogonalizing difference-building $\ominus$'. This linking technique replaces two calculation steps - building the difference and the subsequent orthogonalization - by one step. Additionally, postulates regarding to distributivity and axioms for $\ominus$ are defined which have to be considered getting correct results. Furthermore, every Sum of Products can easily be orthogonalized mathematically by a corresponding equation which bases on $\ominus$. By this orthogonalization it will be reached disjointed Sum of Products in a simpler way. The general validity was also proven by the mathematical induction. An additional step of sorting before the step of orthogonalization achieves a reduction of approximately 17% till 28% of the number of product terms in the disjointed Sum of Products. This feature was illustrated by a measurement whereby the orthogonalization took place before and after sorting. Furthermore, the corresponding Algorithm $dSOP^{c1}$ was compared with two other algorithms $dSOP^{m2}$ [16] and $dSOP^{m1}$ [4]

x_n	$dSOP^{m2}$	$dSOP^{m1}$	$dSOP^{c1}$	$\frac{dSOP^{m2}}{dSOP^{c1}}$	$\frac{dSOP^{m1}}{dSOP^{c1}}$
1	1.99	1.72	1.00	1.99	1.72
2	3.62	2.86	1.06	3.42	2.70
3	5.66	4.57	1.78	3.18	2.57
4	8.67	7.19	2.84	3.05	2.53
5	12.36	10.27	4.03	3.07	2.55
⋮	⋮	⋮	⋮	⋮	⋮
24	186.81	153.97	76.37	2.45	2.02
25	163.02	148.84	69.95	2.33	2.13
26	175.69	150.73	76.41	2.30	1.97
27	184.53	151.81	86.20	2.14	1.76
⋮	⋮	⋮	⋮	⋮	⋮
46	333.92	205.03	190.46	1.75	1.08
47	349.92	205.40	199.80	1.75	1.03
48	393.22	224.77	206.27	1.91	1.09
49	371.80	191.73	198.06	1.88	0.97
50	365.35	199.73	222.69	1.64	0.90
average factor				**2.34**	**1.87**

TABLE 2. Average value of N_{orth} in respect to $N = 10$

in their number of product terms in the obtained dSOP. $dSOP^{c1}$ determines fewer number of product terms in disjointed Sum of Product in contrast to the methods $dSOP^{m2}$ and $dSOP^{m1}$. This reduction is about 50% by $dSOP^{c1}$. In the case of dimension x_n higher 50, this reduction is theoretically to be expected. Consequently, a further calculation of a dSOP such as the Boolean Differential Calculus is performed with fewer number of product terms and thus reduces operation steps. Therefore, at the end of the calculation procedure minimizing of the set of test patterns is expected.

References

[1] Bernasconi, A. and Ciriani, V. and Luccio, F. and Pagli, L., (2008), New Heuristic for DSOP Minimization, In 8th International Workshop on Boolean Problems (IWSBP), 18-19 September 2008, Freiberg (Sachs.), Germany.

[2] Bernasconi, A. and Ciriani, V. and Trucco, G. and Villa, T., (2013), Minimization of EP-SOPs via Boolean relations, In 21st International Conference on Very Large Scale Integration (VLSI-SoC), 7-9 October 2013, Istanbul, Turkey.

[3] Bochmann, D. and Posthoff, Ch., (1981), Binäre dynamische Systeme, Akademie-Verlag, Berlin.

[4] Bochmann, D., (2006), Binäre Systeme - Ein Boolean Buch, LiLoLe-Verlag, Hagen.

[5] Bochmann, D. and Zakrevskij, A.D. and Posthoff, Ch., (1984), Boolesche Gleichungen. Theorie - Anwendungen - Algorithmen, VEB Verlag Technik, Berlin.

[6] Can, Y., (2016), Neue Boolesche Operative Orthogonalisierende Methoden und Gleichungen, FAU University Press, Erlangen.

[7] Can, Y. and Steinbach, B., (2018), Orthogonalization of a TVL in Disjunctive or Conjunctive Form, In: Steinbach B. Further Improvements in the Boolean Domain, Newcastle upon Tyne, Cambridge Scholars Publishing, pp. 156-172.

[8] Can, Y. and Kassim, H. and Fischer, G., (2016), New Boolean Equation for Orthogonalizing of Disjunctive Normal Form based on the Method of Orthogonalizing Difference-Building, Journal of Electronic Testing. Theory and Applicaton (JETTA), 32: 197-208.
[9] Can, Y. and Kassim, H. and Fischer, G., (2016), Orthogonalization of DNF in TVL-Arithmetic, In 12th International Workshop on Boolean Problems (IWSBP), 22-23 September 2016, Freiberg (Sachs.), Germany.
[10] Can, Y., (2016), Quaternary-Vector-List for totally computational treating of switching algebraic tasks, In 26. International Conference on Information, Communication and Automation Technologies (ICAT/IEEE), 22-23 September 2016, Sarajevo, Bosnia & Herzogovina.
[11] Crama, Y. and Hammer, P.L., (2011), Boolean Functions. Theory, Algorithms, and Applications, Cambridge University Press, New York.
[12] Dorotska, Ch. and Steinbach, B., (2003), Orthogonal Block Change & Block Building Using Ordered Lists of Ternary Vectors, In the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM). Proceedings of the 7th International Conference, 18-22 Februar 2003, pp. 441-444.
[13] Kempe, G., (2003), Tupel von TVL als Datenstruktur für Boolesche Funktionen, PhD-Thesis, Technical University Bergakademie, Freiberg.
[14] Posthoff, Ch. and Bochmann, D. and Haubold, K., (1986), Diskrete Mathematik, BSB Teubner, Leibzig.
[15] Posthoff, Ch. and Steinbach, B., (1991), Logikentwurf mit XBOOLE. Algorithmen und Programme, Verlag Technik GmbH, Berlin.
[16] Posthoff, Ch. and Steinbach, B., (1979), Binäre Gleichungen - Algorithmen und Programme, Technische University of Karl-Marx-Stadt, Chemnitz.
[17] Sasao, T., (1993), EXMIN2: A Simplification Algorithm for Exclusive-OR-Sum-of-Products Expression for Multiple-Valued-Input Two-Valued-Output Functions, IEEE Trans. on Computer Aided Design, 12: 621-632.
[18] Steinbach, B., (2009), The Boolean Differential Calculus - Introduction and Examples, Proceedings - Reed-Muller Workshop, 23-24 May 2009, pp. 107-117.
[19] Steinbach, B. and Posthoff, Ch., (2007), An Extended Theory of Boolean Normal Forms, Proceedings of the 6th Annual Hawaii International Conference on Statistics, Mathematics and Related Fields, 22-23 September 2007, pp.1124-1139.
[20] Zander, H.J., (1989), Logischer Entwurf binärer Systeme, Verlag Technik, Berlin.

Dr.-Ing. Yavuz Can was born in Erlangen, Germany, in 1979. He received his Diploma degree (Dipl.-Ing.) in Mechatronics in 2010 and his degree in Doctor of Engineering (Dr.-Ing) in 2016 from Friedrich-Alexander-University of Erlangen-Nuremberg, Germany. He is currently a guest researcher at the Institute for Electronics Engineering at Friedrich-Alexander University in Erlangen. His research mainly deals with the theory and the applications of binary function and algorithms for their utilization. Furthermore, he want to combine Discrete Mathematics and computer-based methods. He is also interested in the theme of orthogonalization of normal forms of Boolean functions and in the field of Ternary-Vector-List arithmetic.