An Extended Theory of Boolean Normal Forms

Bernd Steinbach
Freiberg University of Mining and Technology
Institute of Computer Science
D-09596 Freiberg, Germany
email:steinb@informatik.tu-freiberg.de

Christian Posthoff
The University of The West Indies
St. Augustine Campus
Trinidad & Tobago
email:cposthoff@fsa.uwi.tt

Abstract

Normal forms of Boolean functions allow to check whether two given Boolean expressions describe the same Boolean function. The disjunctive normal form (DNF) as well as the conjunctive normal form (CNF) are widely used. Sometimes an algebraic normal form is taken into account in order to compare or to evaluate given Boolean functions. The basic property of each normal form is the unique representation of the given Boolean function. The aim of this paper is to study whether there are other normal forms of Boolean functions which allow a more detailed exploration of properties of Boolean Functions. In order to do this we extend the theory of Boolean normal forms in several directions.

1 Introduction

A Boolean expression is defined by Boolean variables and constants, Boolean operations, and parentheses. If we assign values 0 or 1 to these Boolean variables and evaluate the Boolean operations according to their priority or controlled by parentheses, we get a single Boolean value 0 or 1. Thus, a correct Boolean expression defines a Boolean function $f : B^k \to B$.

There are many Boolean operations. We restrict our considerations in this paper to the operations NOT ($\overline{a}$), *negation*, AND ($a \wedge b, a \cdot b, a\,b$), *conjunction*, OR ($a \vee b$), *disjunction*, XAND ($a \odot b$), *equivalence*, and XOR ($a \oplus b$), *antivalence*.

The number of of Boolean expressions which represent the same Boolean function is infinite. Thus, it was an important problem to find special types of Boolean expressions that allow to decide whether two Boolean expressions describe the same Boolean function. Based on Shannon's decomposition [7], the disjunctive normal form (DNF) was found. By using de Morgan's law, the conjunctive normal form (CNF) could be derived. In both normal forms NOT-operations are only applied to single variables *(literals)*. Such literals are connected by nonlinear AND- or OR-operations. The DNF and CNF are widely used because these nonlinear operations are well understood by human beings.

Later the linear XOR-operation was used to define an algebraic normal form with the structure of a polynomial of conjunctions. This normal form is widely called *Reed-Muller expansion* [4], [3]. The basic decomposition of this normal form, however, was given by Davio [2]. Using

Table 1: Boolean Operations

a	b	$\overline{a}$	$\overline{b}$	$a \wedge b$	$a \vee b$	$a \odot b$	$a \oplus b$
0	0	1	1	0	0	1	0
0	1	1	0	0	1	0	1
1	0	0	1	0	1	0	1
1	1	0	0	1	1	1	0

the positive and the negative Davio decomposition, two fixed polarity and $2^k - 2$ mixed polarity Reed-Muller normal forms can be specified. It should be mentioned that Zhegalkin [11] suggested such a polynomial.

Usually a Boolean function is uniquely defined by 2^k constants of the DNF, the CNF or each polynomial of the class of Reed-Muller normal forms. In the cases of DNF and CNF these constants are directly the values of the function. The constants of a Reed-Muller normal form can be calculated by suitable transformations or by operations of the Boolean Differential Calculus [1], [5].

Over a long period of time no new concepts for the theory of Boolean normal forms were published. It seemed that this theory was completely known. The DNF was strongly connected with the sum-of-product (SOP) representation of Boolean functions, a research topic for more than half a century. It is known from [6] that the exclusive-or-sum-of-product (ESOP) representation of Boolean functions is typically more compact than the SOP representation. In [9], [10] Steinbach detected and described a new normal form based on ESOPs. In contrast to Reed-Muller normal forms this new *Specialized Normal Form (SFN)* does not fix the literals. By using the SNF, many properties of Boolean functions and particularly their ESOP representations could be found until now. It is to be expected that the theory of Boolean functions can be extended using the SNF.

The rest of this paper is organized as follows. Section 2 will give some basic definitions. The theory of classical normal forms is summarized in Section 3. A first simple extension in Section 4 introduces linear operations in normal forms. A second extension in Section 5 introduces the polynomials as normal forms. The newest extension of the theory of normal forms will be given in Section 6. There the SNF will be introduced, and some properties will be explained. Section 7 concludes the paper.

2 Preliminaries

Table 1 defines the Boolean operations used in this paper. The sign of the AND-operation ($\wedge$) will sometimes be replaced by ($\cdot$) or often even omitted.

The orthogonality can be used as a bridge between the nonlinear operations AND / OR and the associated linear operations XAND / XOR.

Definition 1 ***- Orthogonality -***

1. *If it holds for conjunctions C_i and C_j that*

$$C_i \wedge C_j \equiv 0 \quad \forall\ i \neq j, \tag{1}$$

then

$$\bigvee_{k=1}^{n} C_k = \bigoplus_{k=1}^{n} C_k. \tag{2}$$

2. If it holds for disjunctions D_i and D_j that

$$D_i \vee D_j \equiv 1 \ \ \forall\, i \neq j, \tag{3}$$

then

$$\bigwedge_{k=1}^{n} D_k = \bigodot_{k=1}^{n} D_k. \tag{4}$$

Table 1 can help to understand the orthogonality. Condition (1) selects the first three lines of Table 1 where the columns of OR and XOR are equivalent. For the second case of orthogonality condition (3) selects the last three lines of Table 1 where the columns of AND and XAND are equivalent.

The derivative of the Boolean function $f(x_i, \mathbf{x}_1)$ with regard to the variable x_i is a Boolean function $f(\mathbf{x}_1)$ that is equal to 1 if a change of the value of x_i causes a change of the value of the given function. In [5] it was proven that (5) is equivalent to (6). Equation (6) emphasizes that the derivative of the Boolean function $f(x_i, \mathbf{x}_1)$ with regard to the variable x_i does not depend on x_i.

Definition 2 ***- Simple Derivative -***
Let $f(\mathbf{x}) = f(x_1, \ldots, x_i, \ldots, x_n)$ be a Boolean function of n variables, then

$$\frac{\partial f(\mathbf{x})}{\partial x_i} = f(x_1, \ldots, x_i, \ldots, x_n) \oplus f(x_1, \ldots, \overline{x}_i, \ldots, x_n) \tag{5}$$

is the (simple) derivative of the Boolean function $f(\mathbf{x})$ with regard to the variable x_i.

This definition is equivalent to

$$\frac{\partial f(\mathbf{x})}{\partial x_i} = f(x_1, \ldots, x_i = 0, \ldots, x_n) \oplus f(x_1, \ldots, x_i = 1, \ldots, x_n). \tag{6}$$

Several variables of a Boolean function can change their values simultaneously. The vectorial derivative of the Boolean function $f(\mathbf{x}_0, \mathbf{x}_1)$ with regard to the variables of $\mathbf{x}_0$ is a Boolean function which depends generally on all variables $(\mathbf{x}_0, \mathbf{x}_1)$. A function value 1 of the vectorial derivative indicates that the corresponding value of the given function will change if all variables of $\mathbf{x}_0$ change their values simultaneously. Simple derivatives can be regarded as special case of vectorial derivatives, where the vector $\mathbf{x}_0$ is restricted to a single variable x_i.

Definition 3 ***- Vectorial Derivative -***
Let $\mathbf{x}_0 = (x_1, x_2, ..., x_k)$, $\mathbf{x}_1 = (x_{k+1}, x_{k+2}, ..., x_n)$ be two disjoint sets of Boolean variables, and $f(\mathbf{x}_0, \mathbf{x}_1) = f(x_1, x_2, ..., x_n) = f(\mathbf{x})$ a Boolean function of n variables, then

$$\frac{\partial f(\mathbf{x}_0, \mathbf{x}_1)}{\partial \mathbf{x}_0} = f(\mathbf{x}_0, \mathbf{x}_1) \oplus f(\overline{\mathbf{x}}_0, \mathbf{x}_1) \tag{7}$$

is the vectorial derivative of the Boolean function $f(\mathbf{x}_0, \mathbf{x}_1)$ with regard to the variables of $\mathbf{x}_0$.

Simple derivatives are again Boolean functions so that further simple derivatives with regard to other variables can be calculated. The m-fold derivative of a Boolean function $f(\mathbf{x}_0, \mathbf{x}_1)$ with regard to $\mathbf{x}_0$ is a Boolean function $f(\mathbf{x}_1)$ that is equal to 1 for such subspaces $\mathbf{x}_1 = const$ where the function $f(\mathbf{x}_0, \mathbf{x}_1)$ has an odd number of function values 1. Simple derivatives can be regarded as a special case of m-fold derivatives, too.

Definition 4 - *m-fold Derivative* -
Let $\mathbf{x}_0 = (x_1, x_2, ..., x_m)$, $\mathbf{x}_1 = (x_{m+1}, x_{m+2}, ..., x_n)$ be two disjoint sets of Boolean variables, and $f(\mathbf{x}_0, \mathbf{x}_1) = f(x_1, x_2, ..., x_n) = f(\mathbf{x})$ a Boolean function of n variables, then

$$\frac{\partial^m f(\mathbf{x}_0, \mathbf{x}_1)}{\partial x_1 \partial x_2 \ldots \partial x_m} = \frac{\partial^m f(\mathbf{x}_0, \mathbf{x}_1)}{\partial \mathbf{x}_0} = \frac{\partial}{\partial x_m}\left(\cdots\left(\frac{\partial}{\partial x_2}\left(\frac{\partial f(\mathbf{x}_0, \mathbf{x}_1)}{\partial x_1}\right)\right)\cdots\right) \tag{8}$$

is the m-fold derivative of the Boolean function $f(\mathbf{x}_0, \mathbf{x}_1)$ with regard to the subset of variables $\mathbf{x}_0$.

There is a helpful relationship between a certain set of vectorial derivatives and an associated m-fold derivative. Equation (9) was proven in [1] and implies that the XOR-operation of all vectorial derivatives with regard to all nonempty subsets of variables $\mathbf{x}_0$ leads to the m-fold derivative of $f(\mathbf{x}_0, \mathbf{x}_1)$ with regard to the same set $\mathbf{x}_0$.

Theorem 1 - *Relation between vectorial and m-fold derivatives* -
It holds for $f(\mathbf{x}_0, \mathbf{x}_1) = f(x_1, x_2, ..., x_n) = f(\mathbf{x})$, $\mathbf{x}_0 = (x_1, x_2, ..., x_m)$, $\mathbf{x}_1 = (x_{m+1}, x_{m+2}, ..., x_n)$, $X^* = P(\mathbf{x}_0) \setminus \{\emptyset\}$ that

$$\frac{\partial^m f(\mathbf{x}_0, \mathbf{x}_1)}{\partial x_1 \partial x_2 \ldots \partial x_m} = \bigoplus_{\mathbf{y} \in X^*} \frac{\partial f(\mathbf{x})}{\partial \mathbf{y}}. \tag{9}$$

Shannon published in [7] the well-known basic formula (10) that allows to create disjunctive normal forms.

Definition 5 - *Shannon decomposition* -
Any Boolean function $f(x_i, \mathbf{x}_1)$ can be expressed as

$$f(x_i, \mathbf{x}_1) = \overline{x}_i f(x_i = 0, \mathbf{x}_1) \vee x_i f(x_i = 1, \mathbf{x}_1). \tag{10}$$

The positive (11) and the negative Davio decomposition (12) can be derived from the Shannon decomposition (10).

Theorem 2 - *Davio decomposition* -
Any Boolean function $f(x_i, \mathbf{x_0})$ can be expressed as a positive Davio decomposition

$$f(x_i, \mathbf{x}_1) = f(x_i = 0, \mathbf{x}_1) \oplus x_i \frac{\partial f(x_i, \mathbf{x}_1)}{\partial x_i} \tag{11}$$

or as a negative Davio decomposition

$$f(x_i, \mathbf{x}_1) = f(x_i = 1, \mathbf{x}_1) \oplus \overline{x}_i \frac{\partial f(x_i, \mathbf{x}_1)}{\partial x_i}. \tag{12}$$

Proof. The right-hand side of (10) satisfies (1) so that (by using (2)) equation (10) can be written as

$$f(x_i, \mathbf{x}_1) = \overline{x}_i f(x_i = 0, \mathbf{x}_1) \oplus x_i f(x_i = 1, \mathbf{x}_1) \tag{13}$$

and transformed by

$$\begin{aligned}
f(x_i, \mathbf{x}_1) &= (x_i \oplus 1) f(x_i = 0, \mathbf{x}_1) \oplus x_i f(x_i = 1, \mathbf{x}_1) \\
f(x_i, \mathbf{x}_1) &= f(x_i = 0, \mathbf{x}_1) \oplus x_i f(x_i = 0, \mathbf{x}_1) \oplus x_i f(x_i = 1, \mathbf{x}_1) \\
f(x_i, \mathbf{x}_1) &= f(x_i = 0, \mathbf{x}_1) \oplus x_i (f(x_i = 0, \mathbf{x}_1) \oplus f(x_i = 1, \mathbf{x}_1)) \\
f(x_i, \mathbf{x}_1) &= f(x_i = 0, \mathbf{x}_1) \oplus x_i \frac{\partial f(x_i, \mathbf{x}_1)}{\partial x_i}
\end{aligned}$$

into the positive Davio decomposition. From (13) the negative Davio decomposition can be found by

$$\begin{aligned}
f(xi, \mathbf{x}_1) &= \overline{x}_i f(x_i = 0, \mathbf{x}_1) \oplus (\overline{x}_i \oplus 1) f(x_i = 1, \mathbf{x}_1) \\
f(x_i, \mathbf{x}_1) &= f(x_i = 1, \mathbf{x}_1) \oplus \overline{x}_i f(x_i = 0, \mathbf{x}_1) \oplus \overline{x}_i f(x_i = 1, \mathbf{x}_1) \\
f(x_i, \mathbf{x}_1) &= f(x_i = 1, \mathbf{x}_1) \oplus \overline{x}_i (f(x_i = 0, \mathbf{x}_1) \oplus f(x_i = 1, \mathbf{x}_1)) \\
f(x_i, \mathbf{x}_1) &= f(x_i = 1, \mathbf{x}_1) \oplus \overline{x}_i \frac{\partial f(x_i, \mathbf{x}_1)}{\partial x_i}.
\end{aligned}$$

3 Classical Normal Forms

The disjunctive normal form (DNF) can be calculated iteratively starting from any Boolean expression. The application of Shannon's decomposition (10) to the first variable of the function leads to equation (14):

$$f(x_1, x_2, x_3, \dots, x_k) = \overline{x}_1 f(x_1 = 0, x_2, x_3, \dots, x_k) \vee x_1 f(x_1 = 1, x_2, x_3, \dots, x_k). \tag{14}$$

The application of Shannon's decomposition (10) to the second variable of the function and subsequent transformations using distributive and commutative laws create representation (15) of the function:

$$\begin{aligned}
f(x_1, x_2, \dots, x_k) &= \overline{x}_2(\overline{x}_1 f(x_1 = 0, x_2 = 0, \dots, x_k) \vee x_1 f(x_1 = 1, x_2 = 0, \dots, x_k)) \vee \\
&\quad x_2(\overline{x}_1 f(x_1 = 0, x_2 = 1, , \dots, x_k) \vee x_1 f(x_1 = 1, x_2 = 1, \dots, x_k)) \\
&= \overline{x}_1\overline{x}_2 f(x_1 = 0, x_2 = 0, \dots, x_k) \vee x_1\overline{x}_2 f(x_1 = 1, x_2 = 0, \dots, x_k) \vee \\
&\quad \overline{x}_1 x_2 f(x_1 = 0, x_2 = 1, \dots, x_k) \vee x_1 x_2 f(x_1 = 1, x_2 = 1, \dots, x_k).
\end{aligned} \tag{15}$$

The disjunctive normal form (DNF) (16) has been completely created, if Shannon's decomposition (10) and the above transformations have been applied to all variables of the function:

$$f(\mathbf{x}) = \bigvee_{\mathbf{c} \in B^k} [f(\mathbf{c})(x_1 \oplus \overline{c}_1)(x_2 \oplus \overline{c}_2) \dots (x_k \oplus \overline{c}_k)]. \tag{16}$$

The DNF consists of minterms, selected by constants of function values, and connected by OR-operations. Each minterm includes all variables connected by AND-operations. Due to the NOT-operation, no minterm occurs twice in the DNF. The general DNF takes all existing minterms into account. 2^k function values define the function uniquely. The DNF of a concrete function includes such minterms that are selected by function values 1.

The conjunctive normal form (CNF) of the Boolean function $f(\mathbf{x})$ can be calculated starting with the DNF of its complement $\overline{f}(\mathbf{x})$. The DNF of $\overline{f}(\mathbf{x})$ can be created easily from the DNF of $f(\mathbf{x})$, if all coefficients $f(\mathbf{c})$ are replaced by $\overline{f}(\mathbf{c})$ (17):

$$\overline{f}(\mathbf{x}) = \bigvee_{\mathbf{c} \in B^k} [\overline{f}(\mathbf{c})(x_1 \oplus \overline{c}_1)(x_2 \oplus \overline{c}_2) \dots (x_k \oplus \overline{c}_k)] \tag{17}$$

The solution of equation (17) will not change, if the complement on both sides of the equation is calculated (18). Two consecutive NOT-operations on the left-hand side can be removed, and the NOT-operation on the right-hand side can be transformed by de Morgan's law so that (19) is built:

$$\overline{\overline{f}}(\mathbf{x}) = \overline{\bigvee_{\mathbf{c}\in B^k} [\overline{f}(\mathbf{c})(x_1 \oplus \overline{c}_1)(x_2 \oplus \overline{c}_2)\ldots(x_k \oplus \overline{c}_k)]}. \tag{18}$$

$$f(\mathbf{x}) = \bigwedge_{\mathbf{c}\in B^k} [\overline{\overline{f}}(\mathbf{c}) \vee \overline{(x_1 \oplus \overline{c}_1)} \vee \overline{(x_2 \oplus \overline{c}_2)} \vee \ldots \vee \overline{(x_k \oplus \overline{c}_k)}] \tag{19}$$

Using $\overline{(a \oplus \overline{b})} = (a \oplus \overline{b} \oplus 1) = (a \oplus b \oplus 1 \oplus 1) = (a \oplus b \oplus 0) = (a \oplus b)$ and removing again two consecutive NOT-operations, the conjunctive normal form (CNF) (20) can be created:

$$f(\mathbf{x}) = \bigwedge_{\mathbf{c}\in B^k} [f(\mathbf{c}) \vee f(x_1 \oplus c_1) \vee (x_2 \oplus c_2) \vee \ldots \vee (x_k \oplus c_k)]. \tag{20}$$

The CNF consist of maxterms, selected by constants of function values, and connected by AND-operations. Parentheses are used because the priority of the AND-operation is higher than the priority of the OR-operation. Each maxterm includes all variables connected by OR-operations. Due to the NOT-operation, no maxterm occurs twice in the CNF. The general CNF takes all existing minterms into account. 2^k function values define the function uniquely. The CNF of a concrete function includes such maxterms that are selected by function values 0.

Summarily there are two classical normal forms:

1. *Disjunctive Normal Form* (DNF)

$$f(\mathbf{x}) = \bigvee_{\mathbf{c}\in B^k} [f(\mathbf{c})(x_1 \oplus \overline{c}_1)(x_2 \oplus \overline{c}_2)\ldots(x_k \oplus \overline{c}_k)],$$

2. *Conjunctive Normal Form* (CNF)

$$f(\mathbf{x}) = \bigwedge_{\mathbf{c}\in B^k} [f(\mathbf{c}) \vee f(x_1 \oplus c_1) \vee (x_2 \oplus c_2) \vee \ldots \vee (x_k \oplus c_k)].$$

4 First Extension - Linear Operation

The classical normal forms use, in addition to the NOT-operation, only the nonlinear operations AND and OR. The XOR-operation in the general representation of these normal forms are auxiliaries to express the type of a literal: x or $\overline{x}$, respectively.

New types of normal forms originate if the linear operations XOR and XAND are used instead of the outer nonlinear operations OR and AND of the DFN or CNF. The minterms of a DNF satisfy the first orthogonality condition (1) so that the DNF (16) can be converted into the XOR normal form (21) due to (2). Analogously, the maxterms of a CNF satisfy the second orthogonality condition (3) so that the DNF (20) can be transformed into the XOR normal form (22) due to (4).

Consequently, there are two orthogonal linear normal forms:

1. *XOR Normal Form* (XORNF)

$$f(\mathbf{x}) = \bigoplus_{\mathbf{c}\in B^k} [f(\mathbf{c})(x_1 \oplus \overline{c}_1)(x_2 \oplus \overline{c}_2)\ldots(x_k \oplus \overline{c}_k)], \tag{21}$$

2. *XAND Normal Form* (XANDNF)

$$f(\mathbf{x}) = \bigodot_{\mathbf{c}\in B^k} [f(\mathbf{c}) \vee f(x_1 \oplus c_1) \vee (x_2 \oplus c_2) \vee \ldots (x_k \oplus c_k)]. \tag{22}$$

The XOR-operation allows to create a nonorthogonal normal form. The required steps are explained on the level of two Shannon expansions in a partially built XORNF (23). Due to $a = a \oplus 0$ and $0 = a \oplus a$, the value of a function expressed by a XOR form will not change if the same term is added twice using XOR. In (24) such an extension was done without the minterm that includes only negated variables. Next in (25), the terms are reordered such that once the function $f(x_1 = 0, x_2 = 0, \ldots, x_k)$ can be separated, and in the rest the minterms can be separated. In the last step the minterm can be reduced to the constant 1, and remaining function pairs describe simple or vectorial derivatives (26). The function values of these derivatives coincide for the associated minterm and the fixed position $(x_1 = 0, x_2 = 0)$ so that in (26) this fixed position is chosen.

$$\begin{aligned} f(x_1, x_2, \ldots, x_k) \quad = \quad & \overline{x}_1\overline{x}_2 f(x_1 = 0, x_2 = 0, \ldots, x_k) \oplus \\ & x_1\overline{x}_2 f(x_1 = 1, x_2 = 0, \ldots, x_k) \oplus \\ & \overline{x}_1 x_2 f(x_1 = 0, x_2 = 1, \ldots, x_k) \oplus \\ & x_1 x_2 f(x_1 = 1, x_2 = 1, \ldots, x_k) \end{aligned} \tag{23}$$

$$\begin{aligned} f(x_1, x_2, \ldots, x_k) \quad = \quad & \overline{x}_1\overline{x}_2 f(x_1 = 0, x_2 = 0, \ldots, x_k) \oplus \\ & x_1\overline{x}_2 f(x_1 = 1, x_2 = 0, \ldots, x_k) \oplus \\ & \overline{x}_1 x_2 f(x_1 = 0, x_2 = 1, \ldots, x_k) \oplus \\ & x_1 x_2 f(x_1 = 1, x_2 = 1, \ldots, x_k) \oplus \\ & x_1\overline{x}_2 f(x_1 = 0, x_2 = 0, \ldots, x_k) \oplus x_1\overline{x}_2 f(x_1 = 0, x_2 = 0, \ldots, x_k) \oplus \\ & \overline{x}_1 x_2 f(x_1 = 0, x_2 = 0, \ldots, x_k) \oplus \overline{x}_1 x_2 f(x_1 = 0, x_2 = 0, \ldots, x_k) \oplus \\ & x_1 x_2 f(x_1 = 0, x_2 = 0, \ldots, x_k) \oplus x_1 x_2 f(x_1 = 0, x_2 = 0, \ldots, x_k) \end{aligned} \tag{24}$$

$$\begin{aligned} f(x_1, x_2, \ldots, x_k) \quad = \quad & (\overline{x}_1\overline{x}_2 \oplus x_1\overline{x}_2 \oplus \overline{x}_1 x_2 \oplus x_1 x_2) f(x_1 = 0, x_2 = 0, \ldots, x_k) \oplus \\ & x_1\overline{x}_2 (f(x_1 = 1, x_2 = 0, \ldots, x_k) \oplus f(x_1 = 0, x_2 = 0, \ldots, x_k)) \oplus \\ & \overline{x}_1 x_2 (f(x_1 = 0, x_2 = 1, \ldots, x_k) \oplus f(x_1 = 0, x_2 = 0, \ldots, x_k)) \oplus \\ & x_1 x_2 (f(x_1 = 1, x_2 = 1, \ldots, x_k) \oplus f(x_1 = 0, x_2 = 0, \ldots, x_k)) \end{aligned} \tag{25}$$

$$\begin{aligned} f(x_1, x_2, \ldots, x_k) \quad = \quad & f(x_1 = 0, x_2 = 0, \ldots, x_k) \oplus x_1\overline{x}_2 \left[\frac{\partial f(x_1, x_2, \ldots, x_k)}{\partial x_1}\right]_{x_1=0, x_2=0} \oplus \\ & \overline{x}_1 x_2 \left[\frac{\partial f(x_1, x_2, \ldots, x_k)}{\partial x_2}\right]_{x_1=0, x_2=0} \oplus \\ & x_1 x_2 \left[\frac{\partial f(x_1, x_2, \ldots, x_k)}{\partial (x_1, x_2)}\right]_{x_1=0, x_2=0} \end{aligned} \tag{26}$$

The application of these steps to all variables lead to the nonorthogonal XOR normal form (NOXORNF):

$$f(\mathbf{x}) = f(\mathbf{x} = \mathbf{0}) \oplus \bigoplus_{\mathbf{y} \in P(\mathbf{x}) \setminus \emptyset} \left[\left[\frac{\partial f(x_1, x_2, \ldots, x_k)}{\partial \mathbf{y}}\right]_{\mathbf{x}=\mathbf{0}} \wedge (x_1 \oplus \overline{c}_1)(x_2 \oplus \overline{c}_2) \ldots (x_k \oplus \overline{c}_k) \right] \tag{27}$$

where $\mathbf{c} \in B^k, c_i = \begin{cases} 1 \Leftrightarrow y_i \in \mathbf{y} \\ 0 \Leftrightarrow y_i \notin \mathbf{y} \end{cases}$.

The NOXORNF shows that a Boolean function is uniquely defined by the single function value $f(\mathbf{0})$ and the values of all vectorial derivatives for $\mathbf{x} = \mathbf{0}$.

Generally in an expansion such as (24) $2^k - 1$ pairs of terms are added. The term that is excluded in this procedure determines the fixpoint of the NOXORNF. Consequently, there are 2^k different NOXORNF.

5 Second Extension - Polynomials

The terms of normal forms introduced until now include both negated and unnegated variables. A requirement for polynomials is that each variable occurs either negated or unnegated. The linear XOR-operation allows to solve this problem. The general structure of algebraic normal forms (ANF) is specified in (28).

$$\begin{aligned} f(\mathbf{x}) &= f(x_1, x_2, \ldots, x_k) \\ &= a_0 \oplus a_1(x_1 \oplus c_1) \oplus \ldots \oplus a_n(x_n \oplus c_n) \\ &\quad \oplus a_{1,2}(x_1 \oplus c_1)(x_2 \oplus c_2) \oplus \ldots \oplus a_{k-1,k}(x_{k-1} \oplus c_{k-1})(x_k \oplus c_k) \oplus \ldots \\ &\quad \oplus a_{1,2,\ldots,k}(x_1 \oplus c_1)(x_2 \oplus c_2) \ldots (x_k \oplus c_k) \end{aligned} \tag{28}$$

The 2^k constants a_i specify the Boolean function $f(x_1, x_2, \ldots, x_k)$ uniquely. Thus, ANFs can be used to check whether two given Boolean expressions describe the same Boolean function. A further benefit of ANFs is that certain properties of the represented Boolean function can be detected easily. If, for instance, all coefficients $a_{1,2} \ldots a_{1,2,\ldots,k}$ are equal to 0, then a linear function is given.

If $\mathbf{c} = \mathbf{0}$, then only nonnegated variables appear in (28), and the ANF is called *positive polarity Reed-Muller normal form (PPRMNF)*. If $\mathbf{c} = \mathbf{1}$, then only negated variables appear (28), and the ANF is called *negative polarity Reed-Muller normal form (NPRMNF)*. In all the other cases the ANF is called *mixed polarity Reed-Muller normal form*.

There are different possibilities to find the coefficients of (28). One of them is the algebraic transformation. The following example shows that this method is only suitable for small functions.

$$\begin{aligned} f(a, b, c) &= a \vee \bar{b}c \\ &= a \oplus \bar{b}c \oplus a\bar{b}c \\ &= a \oplus (b \oplus 1)c \oplus a(b \oplus 1)c \\ &= a \oplus bc \oplus c \oplus abc \oplus ac \\ &= a \oplus c \oplus ac \oplus bc \oplus abc \end{aligned}$$

The coefficient of the positive polarity Reed-Muller normal form can be found iteratively using the positive Davio decomposition (11) and the Boolean Differential Calculus [1], [5]. The application of the positive Davio decomposition (11) to the first variable of the function leads to the equation (29). The derivation in (29) does not depend on x_1 so that a restriction to $x_1 = 0$ is allowed (29):

$$f(x_1, x_2, \ldots, x_k) = f(x_1 = 0, x_2, \ldots, x_k) \oplus x_1 \frac{\partial f(x_1, x_2, \ldots, x_k)}{\partial x_1}. \tag{29}$$

The derivation in (29) does not depend on x_1 so that a restriction to $x_1 = 0$ is allowed (30):

$$f(x_1, x_2, \ldots, x_k) = f(x_1 = 0, x_2, \ldots, x_k) \oplus x_1 \left. \frac{\partial f(x_1, x_2, \ldots, x_k)}{\partial x_1} \right|_{x_1 = 0}. \tag{30}$$

The application of the positive Davio decomposition (11) to the second variable of the function and successive transformations create the representation (31) of the function:

$$\begin{aligned} f(x_1,x_2,\ldots,x_k) &= \left[f(x_1=0,x_2,\ldots,x_k) \oplus x_1 \frac{\partial f(x_1,x_2,\ldots,x_k)}{\partial x_1}\right]_{x_1=0,x_2=0} \\ &\quad \oplus x_2 \left[\frac{\partial f(x_1=0,x_2,\ldots,x_k)}{\partial x_2} \oplus x_1 \frac{\partial^2 f(x_1,x_2,\ldots,x_k)}{\partial x_1 \partial x_2}\right]_{x_1=0,x_2=0} \\ &= f(x_1=0,x_2=0,\ldots,x_k) \\ &\quad \oplus x_1 \left[\frac{\partial f(x_1,x_2,\ldots,x_k)}{\partial x_1}\right]_{x_1=0,x_2=0} \\ &\quad \oplus x_2 \left[\frac{\partial f(x_1,x_2,\ldots,x_k)}{\partial x_2}\right]_{x_1=0,x_2=0} \\ &\quad \oplus x_1 x_2 \left[\frac{\partial^2 f(x_1,x_2,\ldots,x_k)}{\partial x_1 \partial x_2}\right]_{x_1=0,x_2=0} \end{aligned} \tag{31}$$

The positive polarity Reed-Muller normal form (32) is completely created, if the positive Davio decomposition (10) and the above transformations are applied to all variables of the function:

$$\begin{aligned} f(x_1,x_2,\ldots,x_k) &= f(x_1=0,x_2=0,\ldots,x_k=0) \\ &\quad \oplus x_1 \left[\frac{\partial f(x_1,x_2,\ldots,x_k)}{\partial x_1}\right]_{x_1=0,x_2=0,\ldots,x_k=0} \oplus \ldots \\ &\quad \oplus x_k \left[\frac{\partial f(x_1,x_2,\ldots,x_k)}{\partial x_k}\right]_{x_1=0,x_2=0,\ldots,x_k=0} \\ &\quad \oplus x_1 x_2 \left[\frac{\partial^2 f(x_1,x_2,\ldots,x_k)}{\partial x_1 \partial x_2}\right]_{x_1=0,x_2=0,\ldots,x_k=0} \oplus \ldots \\ &\quad \oplus x_{k-1} x_k \left[\frac{\partial^2 f(x_1,x_2,\ldots,x_k)}{\partial x_{k-1} \partial x_k}\right]_{x_1=0,x_2=0,\ldots,x_k=0} \oplus \ldots \\ &\quad \oplus x_1 x_2 \ldots x_k \left[\frac{\partial^k f(x_1,x_2,\ldots,x_k)}{\partial x_1 \partial x_2 \ldots \partial x_k}\right]_{x_1=0,x_2=0,\ldots,x_k=0} \end{aligned} \tag{32}$$

The PPRMNF (32) shows that a Boolean function is uniquely defined by the single function value $f(\mathbf{0})$ and the values of all m-fold derivatives for $\mathbf{x} = \mathbf{0}$.

By using the negative Davio decomposition (12) in this approach the negative polarity Reed-Muller normal form (33) is created:

$$
\begin{aligned}
f(x_1,x_2,\ldots,x_k) \quad = \quad & f(x_1=1,x_2=1,\ldots,x_k=1) \\
& \oplus \overline{x}_1 \left[\frac{\partial f(x_1,x_2,\ldots,x_k)}{\partial x_1}\right]_{x_1=1,x_2=1,\ldots,x_k=1} \oplus \ldots \\
& \oplus \overline{x}_k \left[\frac{\partial f(x_1,x_2,\ldots,x_k)}{\partial x_k}\right]_{x_1=1,x_2=1,\ldots,x_k=1} \\
& \oplus \overline{x}_1\overline{x}_2 \left[\frac{\partial^2 f(x_1,x_2,\ldots,x_k)}{\partial x_1 \partial x_2}\right]_{x_1=1,x_2=1,\ldots,x_k=1} \oplus \ldots \\
& \oplus \overline{x}_{k-1}\overline{x}_k \left[\frac{\partial^2 f(x_1,x_2,\ldots,x_k)}{\partial x_{k-1} \partial x_k}\right]_{x_1=1,x_2=1,\ldots,x_k=1} \oplus \ldots \\
& \oplus \overline{x}_1\overline{x}_2\ldots\overline{x}_k \left[\frac{\partial^k f(x_1,x_2,\ldots,x_k)}{\partial x_1 \partial x_2 \ldots \partial x_k}\right]_{x_1=1,x_2=1,\ldots,x_k=1} \qquad (33)
\end{aligned}
$$

In order to create a Reed-Muller normal form, the type of the Davio decomposition can be chosen for each variable without any restrictions. Consequently, in addition to PPRMNF and NPRMNF of $f(x_1,x_2,\ldots,x_k)$, there are 2^k-2 mixed polarity Reed-Muller normal forms. The general Reed-Muller normal form (GRMNF) is given in (34) where the constants in the vector **c** specify the chosen polarities:

$$
\begin{aligned}
f(x_1,x_2,\ldots,x_k) \quad = \quad & f(x_1=c_1,x_2=c_2,\ldots,x_k=c_k) \\
& \oplus (x_1 \oplus c_1) \left[\frac{\partial f(x_1,x_2,\ldots,x_k)}{\partial x_1}\right]_{\mathbf{x}=\mathbf{c}} \oplus \ldots \\
& \oplus (x_k \oplus c_k) \left[\frac{\partial f(x_1,x_2,\ldots,x_k)}{\partial x_k}\right]_{\mathbf{x}=\mathbf{c}} \\
& \oplus (x_1 \oplus c_1)(x_2 \oplus c_2) \left[\frac{\partial^2 f(x_1,x_2,\ldots,x_k)}{\partial x_1 \partial x_2}\right]_{\mathbf{x}=\mathbf{c}} \oplus \ldots \\
& \oplus (x_{k-1} \oplus c_{k-1})(x_k \oplus c_k) \left[\frac{\partial^2 f(x_1,x_2,\ldots,x_k)}{\partial x_{k-1} \partial x_k}\right]_{\mathbf{x}=\mathbf{c}} \oplus \ldots \\
& \oplus (x_1 \oplus c_1)(x_2 \oplus c_2)\ldots(x_k \oplus c_k) \left[\frac{\partial^k f(x_1,x_2,\ldots,x_k)}{\partial x_1 \partial x_2 \ldots \partial x_k}\right]_{\mathbf{x}=\mathbf{c}} \qquad (34)
\end{aligned}
$$

An alternative source of the general Reed-Muller normal form is the nonorthogonal XOR normal form (27) which can be as in (35):

$$
\begin{aligned}
f(x_1,x_2,\ldots,x_k) \quad = \quad & f(\mathbf{x}=\mathbf{0}) \\
& \oplus x_1\overline{x}_2\ldots\overline{x}_k \left[\frac{\partial f(x_1,x_2,\ldots,x_k)}{\partial x_1}\right]_{\mathbf{x}=\mathbf{0}} \oplus \\
& \oplus \overline{x}_1 x_2\ldots\overline{x}_k \left[\frac{\partial f(x_1,x_2,\ldots,x_k)}{\partial x_2}\right]_{\mathbf{x}=\mathbf{0}} \oplus \ldots \\
& \oplus x_1 x_2\ldots\overline{x}_k \left[\frac{\partial f(x_1,x_2,\ldots,x_k)}{\partial (x_1,x_2)}\right]_{\mathbf{x}=\mathbf{0}} \oplus \ldots \\
& \oplus x_1 x_2\ldots x_k \left[\frac{\partial f(x_1,x_2,\ldots,x_k)}{\partial (x_1,x_2\ldots x_k)}\right]_{\mathbf{x}=\mathbf{0}} \qquad (35)
\end{aligned}
$$

The minterms in (35) can be replaced by polynomials (36):

$$
\begin{aligned}
f(x_1, x_2, \dots, x_k) \quad = \quad & f(\mathbf{x} = \mathbf{0}) \\
& \oplus (x_1 \oplus x_1 x_2 \oplus \dots \oplus x_1 x_k \oplus \dots \oplus x_1 x_2 \dots x_k) \left[\frac{\partial f(\mathbf{x})}{\partial x_1} \right]_{\mathbf{x}=\mathbf{0}} \oplus \\
& \oplus (x_2 \oplus x_1 x_2 \oplus \dots \oplus x_2 x_k \oplus \dots \oplus x_1 x_2 \dots x_k) \left[\frac{\partial f(\mathbf{x})}{\partial x_2} \right]_{\mathbf{x}=\mathbf{0}} \oplus \dots \\
& \oplus (x_1 x_2 \oplus \dots \oplus x_1 x_2 \dots x_k) \left[\frac{\partial f(\mathbf{x})}{\partial (x_1, x_2)} \right]_{\mathbf{x}=\mathbf{0}} \oplus \dots \\
& \oplus x_1 x_2 \dots x_k \left[\frac{\partial f(\mathbf{x})}{\partial (x_1, x_2 \dots x_k)} \right]_{\mathbf{x}=\mathbf{0}} \qquad (36)
\end{aligned}
$$

By using distributive and commutative laws, equation (36) can be transformed into (37):

$$
\begin{aligned}
f(x_1, x_2, \dots, x_k) \quad = \quad & f(\mathbf{x} = \mathbf{0}) \\
& \oplus x_1 \left[\frac{\partial f(\mathbf{x})}{\partial x_1} \right]_{\mathbf{x}=\mathbf{0}} \oplus \\
& \oplus x_2 \left[\frac{\partial f(\mathbf{x})}{\partial x_2} \right]_{\mathbf{x}=\mathbf{0}} \oplus \dots \\
& \oplus x_1 x_2 \left[\frac{\partial f(\mathbf{x})}{\partial x_1} \oplus \frac{\partial f(\mathbf{x})}{\partial x_2} \oplus \frac{\partial f(\mathbf{x})}{\partial (x_1, x_2)} \right]_{\mathbf{x}=\mathbf{0}} \oplus \dots \\
& \oplus x_1 x_2 \dots x_k \left[\frac{\partial f(\mathbf{x})}{\partial x_1} \oplus \frac{\partial f(\mathbf{x})}{\partial x_2} \oplus \frac{\partial f(\mathbf{x})}{\partial (x_1, x_2)} \oplus \dots \oplus \frac{\partial f(\mathbf{x})}{\partial (x_1, x_2 \dots x_k)} \right]_{\mathbf{x}=\mathbf{0}} (37)
\end{aligned}
$$

By means of theorem (9) XORs of vectorial derivatives can be substituted by m-fold derivatives so that the positive polarity Reed-Muller normal form (32) originates. Analogue transformations exist between NOXORNFs of all 2^k fixed points and associated GRMNFs. Due to the linearity of the used operation inverse transformations from GRMNFs to NOXORNFs can be executed as well.

6 Third Extension - New Special Normal Form: SNF

An exclusive-or-sum-of-product (ESOP) consists of cubes connected by XOR-operations. In a cube negated or nonnegated Boolean variables are connected by AND operations. A reduced ESOP of $f(x_1, x_2, \dots, x_k)$ includes each cube at most once such that it may contain up to 3^k cubes. All normal forms considered until now restrict this set of cubes to 2^k. XORNFs take cubes of minterms into account, NOXORNFs replace one of these minterms by a constant function value, and GRMNFs allow only one polarity for each variable. Does there exist a normal form in the ESOP domain that overcomes theses restrictions? A positive answer to this question is given in this section. This generalization allows to study properties of Boolean functions more in detail.

An algebraic property of the XOR-operation and the Boolean variable x is visible in the following formulas:

$$
\begin{aligned}
x &= \overline{x} \oplus 1 \qquad (38) \\
\overline{x} &= 1 \oplus x \qquad (39) \\
1 &= x \oplus \overline{x} \qquad (40)
\end{aligned}
$$

These three formulas show that each element of the set $\{x, \overline{x}, 1\}$ has isomorphic properties. For each variable in the support of the Boolean function f, exactly one left-hand side element of (38), (39) or (40) is included in each cube of an ESOP of function f. An application of these formulas from the left to the right doubles the number of cubes and is called expansion. The reverse application of these formulas from the right to the left halves the number of cubes and is called compaction.

A second important property of the exclusive-or operation for a Boolean function f and a cube C is shown in the following formulas:

$$f = f \oplus 0 \tag{41}$$
$$0 = C \oplus C \tag{42}$$
$$f = f \oplus C \oplus C \tag{43}$$

It follows from these formulas follows that two identical cubes can be added to or removed from any ESOP without changing the represented function. The SNF can be defined using two simple algorithms based on the properties mentioned above. These algorithms define the functions `Exp(`f`)` and `R(`f`)`, respectively. The function `Exp(`f`)` of Algorithm 1 distributes the information about each given cube to 2^k cubes, similar to the creation of a hologram of an object.

Algorithm 1 Calculate Exp(f)

Require: any ESOP of a Boolean function f
Ensure: complete expansion of the Boolean function f w.r.t. all variables of its support
1: **for all** variables V_i of the support of f **do**
2:     **for all** cubes C_j of f **do**
3:         $\langle C_{n1}, C_{n2}\rangle \leftarrow$ `expand`(C_j, V_i)
4:         replace C_j by $\langle C_{n1}, C_{n2}\rangle$
5:     **end for**
6: **end for**

The `expand()` function in line 3 expands the cube C_j with respect to the variable V_i into the cubes C_{n1} and C_{n2} based on the appropriate formula (38), (39) or (40).

The function `R(`f`)` of Algorithm 2 removes pairs of cubes such that the remaining ESOP of the Boolean function f includes each cube at most once. This algorithm is based on formula (41), (42), (43).

Algorithm 2 Calculate R(f)

a

Require: any ESOP of a Boolean function f containing n cubes
Ensure: reduced ESOP of f containing no cube more than once
for $i \leftarrow 0$ to $n-2$ **do**
    for $j \leftarrow i+1$ to $n-1$ **do**
        if $C_i = C_j$ **then**
            $C_i \leftarrow C_{n-1}$
            $C_j \leftarrow C_{n-2}$
            $n \leftarrow n-2$
            $j \leftarrow i$
        end if
    end for
end for

Using the algorithms `Exp(`f`)` and `R(`f`)` it is possible to create a special ESOP. Generally all 3^k cubes may occur in such a special ESOP. Thus 3^k constants c_i define which one of 2^k Boolean functions is expressed by the ESOP given by the SNF(f) (44).

Definition 6 ***- Specialized Normal Form: SNF(f) -***
Take any ESOP of a Boolean function f. The resulting ESOP of

$$SNF(f) = R(Exp(f)) \tag{44}$$

is called Specialized Normal Form (SNF) of the Boolean function.

In contrast to the normal forms introduced above not each set of 3^k constants c_i of the SNF describes an ESOP in normal form. Thus the constants c_i of the SNF(f) do not express the function f only, but also further information about this function f. The SNF (44) shows several remarkable properties. Some of them are summarized in the following.

Theorem 3 ***- Normal Form: SNF -***
The SNF(f) (44) is unique and consequently a normal form.

Theorem 4 ***- Fixed Number of Different Reduced ESOPs -***
Every Boolean function has exactly $2^{(3^k-2^k)}$ different reduced ESOPs.

Theorem 5 ***- Weak Lower Bound on the Size of the Minimum ESOP -***
No ESOP representation of a Boolean function f contains less than

$$\left\lceil \frac{|SNF(f)|}{2^k} \right\rceil \tag{45}$$

cubes (the smallest integer number greater than the number of cubes in the SNF(f) divided by 2^k).

Theorem 6 ***- Construct Minimal ESOP -***
Adding the smallest number of pairs of the identical cubes to SNF(f) in such a way that the complete expansions of cubes are created, leads to the exact minimum ESOP of the given function.

Theorem 7 ***- Overlapping of Expanded Cubes -***
Let two cubes, C_1 and C_2, belonging to the ESOP of $f(x_1, x_2, \ldots, x_k)$, have the distance D. Then their expansions, Exp(C_1) and Exp(C_2), overlap in 2^{k-D} cubes.

Proofs. Theorems 3, ..., 7 have been proven in [9].

The number of cubes in the SNF(f) is a measure of the represented function f. The more cubes in the SNF(f), the more complex is the function f. The number of cubes in the SNF, marked by `# SNF` in Figure 1, is a simple criterion to classify Boolean functions. Such a classification allows to study the properties of large Boolean functions, because the number of functions to be analyzed can be restricted to one representative of a class.

Figure 1 shows the results of a complete evaluation of all 65536 Boolean functions of four variables. The number of functions of each class is marked by `# BF`. It can be seen in Figure 1 that some classes include ESOPs having different numbers of cubes in their exact minimal ESOP marked by `# EMIN`. This value can be taken to define subclasses of functions.

Figure 1 documents both the benefit of a simple specification of classes of Boolean functions by the number of cubes in their SNF and the requirement to specify more detailed subclasses. One possibility to characterize the subclasses may be the adjacency graph of a SNF. This graph emphasizes implicit knowledge of the SNF such that edges of the graph describe structural relationships between the cubes of the SNF which are used as labels of the vertices.

```
# ALLBF = 65536
# SNF = 0        # EMIN = 0   # BF 1
# SNF = 16       # EMIN = 1   # BF 81
# SNF = 24       # EMIN = 2   # BF 324
# SNF = 28       # EMIN = 2   # BF 1296
# SNF = 30       # EMIN = 2   # BF 648
# SNF = 32       # EMIN = 3   # BF 648
# SNF = 34       # EMIN = 3   # BF 3888
# SNF = 36       # EMIN = 3   # BF 6624
# SNF = 36       # EMIN = 4   # BF 108
# SNF = 38       # EMIN = 3   # BF 7776
# SNF = 40       # EMIN = 3   # BF 2592
# SNF = 40       # EMIN = 4   # BF 6642
# SNF = 42       # EMIN = 3   # BF 216
# SNF = 42       # EMIN = 4   # BF 14256
# SNF = 44       # EMIN = 4   # BF 12636
# SNF = 46       # EMIN = 4   # BF 3888
# SNF = 46       # EMIN = 5   # BF 1296
# SNF = 48       # EMIN = 5   # BF 1944
# SNF = 50       # EMIN = 5   # BF 648
# SNF = 54       # EMIN = 6   # BF 24
```

Figure 1: Classes of all exact minimal ESOPs of 4 variables.

Definition 7 ***- Adjacency Graph*** $AG^{SNF(f)}$***(V,E) of the SNF(f) -***
The vertices V of the adjacency graph $AG^{SNF(f)}(V,E)$ *correspond to the cubes of the SNF(f). Each vertex carries the ternary vector of the associated cube as label. Two vertices V of* $AG^{SNF(f)}(V,E)$ *are connected by an edge, if the associated labels have a distance equal to one that means they differ exactly in one position of the ternary vectors.*

An interesting property of any adjacency graph is given in the following theorem.

Theorem 8 ***-*** $AG^{SNF(f)}(V,E)$ ***is a*** k***-regular graph -***
The degree of any vertex in the adjacency graph $AG^{SNF(f)}(V,E)$ *of the SNF(*$f(x_1,x_2,\ldots,x_k)$*) is equal to* k*.*

Proof. Theorem 8 was proven in [9].

The degree of a vertex is the number of vertices connected with this vertex. It follows from Theorem 8 that the adjacency graph $AG^{SNF(f)}(V,E)$ for a Boolean function $f : B^k \to B$ is a *k-regular graph*.

A criterion to define the required subclasses of Boolean functions is that their adjacency graphs are isomorphic to each other.

The adjacency graph $AG^{SNF(f)}(V,E)$ may by the basic instrument to find the associated exact minimal ESOP. In [8] an extended adjacency graph $EAG^{SNF(f)}(V,E)$ of the SNF(f) was defined which indicates the cubes that must be added in order to find the cube of an exact minimal ESOP.

Figure 2 illustrates for a very simple example how exact minimal ESOPs can be constructed from a given SNF. In this example one pair of cubes is added to the SNF. Note, adding of a pair of cubes does not change the function. Each of the created lattices can be collapsed into a single cube of the minimal ESOP. There are three minimal ESOPs created from the graphs of Figure 2 b), c), and d).

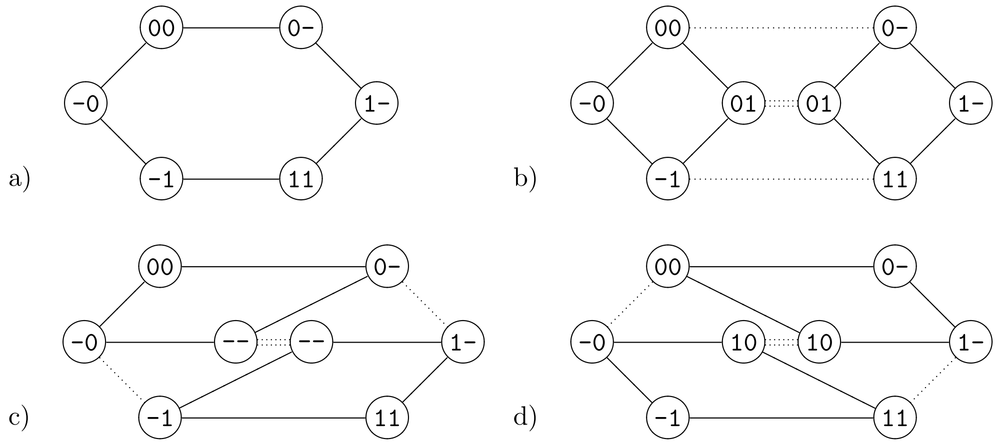

Figure 2: Reconstruction of the minimal ESOP of $SNF(f) = \overline{x}_1\overline{x}_2 \oplus \overline{x}_1 \oplus x_1 \oplus x_1x_2 \oplus x_2 \oplus \overline{x}_2$: a) Adjacency graph $AG^{SNF(f)}(V, E)$, b)Adding the pair $(\overline{x}_1x_2, \overline{x}_1x_2)$ - minimal ESOP $f(x_1, x_2) = x_1 \oplus \overline{x}_2$, c)Adding the pair $(0, 0)$ - minimal ESOP $f = x_1x_2 \oplus \overline{x}_1\overline{x}_2$, d)Adding the pair $(x_1\overline{x}_2, x_1\overline{x}_2)$ - minimal ESOP $f = \overline{x}_1 \oplus x_2$.

7 Conclusions

Basically normal forms were introduced in order to check whether two expressions describe the same Boolean function. Widely used are the DNF and the CNF which reflect directly the ON-set and the OFF-set of the function. This basic normal forms use, in addition to the NOT-operation, only the nonlinear AND- and OR-operations.

The orthogonality allows to introduce the linear XOR- and XAND-operation in the XORNF and the XANDNF. Due to the orthogonality, the semantic of these normal forms is not comparable to the DNF and the CNF. A first step into an alternative model of normal forms was done by the NOXORNF. Instead of function values this normal form requires only one function value and uses $2^k - 1$ values of vectorial derivatives.

The NOXORNF can be transformed into a normal form having a polynomial structure. It is characteristic for the GRMNF, in contrast to XORNF, that the cubes connected by XOR include zero to k variables. The unique representation of the GRMNF is ensured by a fixed polarity of each variable. As in the NOXORNF the GRMNF can be defined for 2^k basic points. The 2^k coefficients of the GRMNF can be calculated by means of m-fold derivatives. Alternatively the positive or negative Davio decomposition can be used to create each one of the GRMNFs, particularly the PPRMNF and the NPRMNF.

In the same way as the normal forms mentioned above, the SNF describes a given Boolean function in a unique way. In contrast to NOXORNF or GRMNF, each cube may occur in the ESOP of the SNF. While the 2^k coefficients of all previous normal forms can be chosen without any restriction in order to define a Boolean function, the 3^k coefficients of the SNF are not independent on each other. In addition to the check whether two expressions describe the same Boolean function, the SNF can be used to find an exact minimal ESOP without time-consuming search, it can classify Boolean functions and evaluate their properties.

The theory of Boolean normal forms is almost complete for DNF, CNF, XORNF, XANDNF, NOXORNF, and GRMNF. An extension of the theory of Boolean normal forms is given by the SNF. Several properties of Boolean functions, particularly in the ESOP domain, were detected by the SNF itself, the associated adjacency graph $AG^{SNF(f)}(V, E)$, and the extended adjacency graph $EAG^{SNF(f)}(V, E)$. It can be expected that by means of the SNF further knowledge about Boolean functions will be achieved in the near future.

References

[1] D. Bochmann, Ch. Posthoff: Binäre dynamische Systeme. Oldenbourg Verlag, München, Germany, 1981.

[2] P. Davio, J.P. Deschamps, and A. Thayse: Discrete and Switching Functions. New York: McGraw-Hill, 1978.

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

[4] I.S. Reed: A Class of Multiple-Error-Correcting Codes and Their Decoding Scheme. IRE Trans. Information Theory, vol. 4, 1954, pp. 38-42.

[5] Ch. Posthoff, B. Steinbach: Logic Function and Equation - Binary Models for Computer Science. Springer, 2004.

[6] T. Sasao, M. Fujita: Representation of Discrete Functions, Kluwer Academic Publishers, 1996.

[7] C.E. Shannon, A Symbolic Analysis of Relay and Switching Circuits. Transactions of the AIEE, Vol. 57, 1938, pp. 713-723.

[8] B. Steinbach: Adjacency Graph of the SNF as Source of Information. in: Proceedings of the 7th International Workshops on Boolean Problems, September 21-22 September 2006, Freiberg University of Mining and Technology, Germany, 2006, pp. 19-27.

[9] B. Steinbach, A. Mishchenko: SNF: A Special Normal Form for ESOPs. in: Proceedings of the 5th International Workshop on Application of the Reed-Muller Expansion in Circuit Design (RM 2001), August 10 - 11, 2001. Mississippi State University, Starkville (Mississippi) USA, pp 66 - 81.

[10] B. Steinbach, V. Yanchurkin, M. Lukac: On SNF Optimization: a functional comparison of methods. in: Proceedings of the 6th International Symposium on Representations and Methodology of Future Computing Technology (RM 2003), March 10 - 11, 2003. University of Trier, Germany, 2003, pp 11 - 18.

[11] I.I. Zhegalkin: On the Technique of Calculation the Sentences in Symbolic Logic. Matem. Sbornik, vol. 34, pp. 9-28, 1927 (in Russian).