Properties of Autocorrelation Coefficients for Single-Output Switching Functions

J. E. Rice, Member, IEEE, J. C. Muzio, Senior Member, IEEE, N. A. Anderson and R. Jansen

Abstract—A variety of mathematical transforms have traditionally been used in various logic synthesis applications. This paper investigates the use of the autocorrelation transform:

\[ C(\tau) = \sum_{v=0}^{2^n-1} f(v) \cdot f(v \oplus \tau) \]

Properties of the coefficient resulting from the application of this transform to switching functions are examined and detailed, including properties to identify symmetries and decompositions. The potential uses in logic synthesis of these properties and other observations based on the autocorrelation coefficients are explored, with emphasis on proofs as mathematical justification of the theorems relating the observed properties of the coefficients to properties of the underlying switching functions.

Index Terms—high level synthesis, logic synthesis, transforms, Boolean logic, digital logic, function representations.

I. INTRODUCTION

The autocorrelation transform has been used in various areas including optimization and synthesis of combinational logic [1], variable ordering for Binary Decision Diagrams [2], and to compute the estimate \( C(f) \) of a function’s complexity [1, 3]. Use of the autocorrelation transform, however, has been limited. This may be due to either the fact that their computation is not trivial, or that little has been known of the transform’s properties. To address this first problem new computation methods have recently been introduced by Rice, Muzio and Serra [4, 5] as well as by Stankovic and Karpovsky [6]. To address the second we devote this work to an explanation of the theoretical use of the autocorrelation transform in the identification of properties that may be useful in activities such as logic synthesis of Boolean functions. Effort is made to provide proofs to both justify and explain how we propose these coefficients be used, and although this work does not yet include experimental results based on these theories, future work in this direction is currently underway.

We first present the definition and an explanation of the autocorrelation transform. We then introduce several theorems relating the values of the resulting autocorrelation coefficients to properties of the underlying switching function. A number of potential applications for these theorems are presented, and future directions for this work are discussed.

II. BACKGROUND

The autocorrelation transform is a special case of the correlation transform, which is defined in [3] as follows:

\[ B^k(\tau) = \sum_{v=0}^{2^n-1} f(v) \cdot g(v \oplus \tau). \]  \hspace{1cm} (1)

If \( f \) and \( g \) are the same function then this becomes the autocorrelation transform, also called the cross-correlation, or convolution function. The superscript is generally omitted when referring to the autocorrelation transform. By convention \( B(\tau) \) is evaluated with \( f \) in the usual Boolean domain of \( \{0,1\} \). If \{+1,-1\} encoding is used then the resulting autocorrelation coefficients are denoted as \( C(\tau) \):

\[ C(\tau) = \sum_{v=0}^{2^n-1} f(v) \cdot f(v \oplus \tau). \]  \hspace{1cm} (2)

It is straightforward to derive the relationship between \( B \) and \( C \), namely:

\[ C(\tau) = 2^n - 4k + 4B(\tau). \]  \hspace{1cm} (3)

In this equation, \( k = B(0) \), which is also the number of true minterms in the function. We should also point out that the \( + \) operator is used to indicate the OR operator when used in logical expressions, and to indicate arithmetic addition when used in arithmetic equations such as (3) or summations.

The derivation is based on this relationship between \{0,1\} encoded outputs, labeled as \( z \), and \{+1,-1\} encoded outputs, labeled as \( y \): \( y_i = -2z_i + 1 \).

Although the same information is present in both \( C(\tau) \) and \( B(\tau) \) there are some patterns that are more easily identifiable when using \{+1,-1\} encoding, and vice versa. Thus it is useful to be able to use either encoding, particularly for analysis.

It is useful to present some additional notation to aid in the understanding of this paper.

\( \tau \) and \( \tau' \) indicate values ranging from \( 0 \) to \( 2^n - 1 \). \( \tau \), is used to indicate one such value. These are usually expressed as a binary expansion.
\[ \sum_{\tau=0}^{2^n-1} B(\tau) = k^2. \] 

The proof of this theorem relies on the following Lemma:

**Lemma 1**

\[ \sum_{\tau=1}^{2^n-1} B(\tau) = 2 \binom{k}{2}. \]

\( \binom{k}{2} \) is the number of pairings of the minterms as computed in the summation of the autocorrelation coefficients. This is then multiplied by two to produce all possible pairings in the form \( i, j \) and \( j, i \).

**Proof:** Using Lemma 1 the sum of all of the \( \{0,1\} \) autocorrelation coefficients is as follows:

\[
\sum_{\tau=0}^{2^n-1} B(\tau) = B(0) + 2 \binom{k}{2} \\
= k + 2 \frac{k(k-1)}{2} \\
= k^2.
\]

**Theorem 2**

\[ C(\tau) = 2^n - 4m \quad \forall \tau \]

where \( m \leq k, k-2, \ldots, 0 \) for even values of \( k \) and \( m \leq k, k-2, \ldots, 1 \) for odd values of \( k \), \( k \) being the number of true minterms in the function or the number of false minterms in the function, whichever is fewer.

**Proof:** The largest possible number of mismatch pairs, that is, negative contributions to the total coefficient value is \(-2k\). The remaining pairs, which of necessity result in positive contributions to the coefficient value is \( 2(2^n - k) \). Thus the total value for the coefficient is

\[ C(\tau) = -2k + 2(2^n - k) \\
= -2k + 2^n - 2k \\
= 2^n - 4k.
\]

However, this assumes that all positive minterms will pair with false minterms and vice versa. This is not the case; for some coefficients a subset of false minterms may pair with other false minterms. Each time a false minterm is paired with another false minterm the number of negative contributions is reduced by 2, leading to the equation in Theorem 2.

**IV. General Properties**

This section introduces theorems that relate particular patterns in the autocorrelation coefficients to underlying properties of the switching function. We propose in future work to utilize these patterns in identifying properties in the switching functions that may be useful in logic synthesis. For example, identification of variables of which the function is independent may reduce the problem size to something more manageable, while determining the possible existence of symmetries is known to be a useful technique in logic synthesis [7].
A. Trivial Functions

A trivial function is one in which all output values of the function are 1, or all output values of the function are 0, assuming \([0,1]\) encoding.

**Theorem 3** \(C(\tau) = C(\tau') \forall \tau \text{ and } \tau' \in \{0,\ldots, 2^n - 1\}\) and if and only if \(f(X) = 1\) or \(f(X) = 0\).

**Proof:** If all the coefficients are equal, they must all have the value \(2^n\) as the coefficient \(C(0)\) always has this value. Based on this, if all of the coefficients have equal value, then this implies that the function matches itself at every value of \(\tau\). This can only occur if the function consists entirely of true minterms, or entirely of false minterms.

B. Identifying Redundant Variables

The following two theorems may be applied to identify redundant variables in a function. Theorem 4 describes the situation that occurs when a function does not depend on one of the input variables. Theorem 5 describes the situation when a function may depend on only one of the input variables.

**Theorem 4** A function \(f(X)\) is independent of input variable \(x_i\) if and only if \(C(\tau_i) = 2^n\).

**Proof:** Without loss of generality let us define a function \(f(X)\) that is independent of \(x_n\). By definition, \(f(0, x_{i_1}, \ldots, x_{i_l}) = f(1, x_{i_1}, \ldots, x_{i_l})\). Then

\[
C(\tau_n) = \sum_{\tau = 0}^{2^n - 1} f(\tau) \times f(\tau \oplus \tau_n)
\]

\[
= \sum_{\tau = 0}^{2^n - 1} f(\tau) \times f(\tau \oplus \tau_n)
\]

\[
+ \sum_{\tau = 2^n}^{2^n - 1} f(\tau) \times f(\tau \oplus \tau_n)
\]

Let us define the range 0 to \(2^{n-1}\) as A and \(2^{n-1}\) to \(2^n - 1\) as B. Then \(\tau \in A \Rightarrow \tau \oplus \tau_n \in B\) and \(\tau \in B \Rightarrow \tau \oplus \tau_n \in A\) . Since the function is defined to have \(f(A) = f(B)\) then

\[
C(\tau_n) = \sum_{\tau = 0}^{2^n - 1} f(\tau) \times f(\tau \oplus \tau_n)
\]

\[
+ \sum_{\tau = 2^n}^{2^n - 1} f(\tau) \times f(\tau \oplus \tau_n)
\]

\[
= 1 + 1
\]

\[
= 2^n.
\]

To prove the second part of the theorem we define (without loss of generality) a function \(f(X)\) for which \(C(\tau_0) = 2^n\). This is only possible if \(f(\tau) = f(\tau \oplus \tau_0) \forall \tau\). This implies that \(f(1, x_{i_1}, \ldots, x_{i_l}) = f(0, x_{i_1}, \ldots, x_{i_l})\), indicating that \(f(X)\) is not dependent on \(x_n\).

**Theorem 5** A function \(f(X)\) has \(2^{n-1}\) autocorrelation coefficients \(C(\tau) = 2^n\) (including \(C(0)\)) and the remaining \(2^{n-1}\) coefficients \(C(\tau') = 2^n\) if and only if the function is dependent on only one of its input variables OR is related to such a function through the application of one or more invariance operations.

A function that is dependent on only one of its input variables has exactly \(2^{n-1}\) true minterms. However, this is not the only situation where a function can have exactly \(2^{n-1}\) true minterms. As discussed in [8] any function that is related to this type of function through the application of one of four invariance operations will have \(2^{n-1}\) true minterms. The theorem above and the proof following refer to any of these types of functions.

**Proof:** Without loss of generality let us define \(f(X) = x_1\), where \(x_1\) is the lowest order bit of the input \(X\). Then if \(\tau\) is an odd number the binary expansion of \(\tau\) contains a 1 in the lowest order bit, and then by definition \(f(\tau) = f(\tau \oplus \tau)\) where the bar indicates the Boolean not operator. Then

\[
C(\tau) = \sum_{\tau = 0}^{2^n - 1} f(\tau) - 1 = -2^n.
\]

Similarly if \(\tau'\) is an even number, then the binary expansion contains a 0 in the lowest order bit and by definition \(f(\tau) = f(\tau \oplus \tau')\). Then

\[
C(\tau') = \sum_{\tau = 0}^{2^n - 1} f(\tau) = 2^n.
\]

Given autocorrelation coefficients of the pattern described above the function must be dependent on only one of the input variables (or related to such a function). Without loss of generality we assume that \(C(\tau') = 2^n\) where \(\tau'\) is even and \(C(\tau) = 2^n\) where \(\tau\) is odd. \(C(\tau') = 2^n\) where \(\tau'\) is even indicates that the function matches up two false or two true minterms for every product in the summation. Additionally every product being computed is comparing two inputs for which \(x_1\) remains unchanged. Moreover, 
\(C(\tau) = 2^n\) where \(\tau\) is odd indicates that the function matches a false minterm with a true minterm for every product in the summation, and that every product is matching a pair of inputs for which \(x_1\) varies. Based on this we can determine that the function must be dependent only on \(x_n\).

C. Dissimilar Minterms

The following are three theorems that allow a designer to identify a sparse (or the inverse) function from the values of the function’s autocorrelation coefficients. A sparse function is one in which a majority of the input values result in a particular output, e.g. \(0\), and the remaining minority (possibly only one) result in the other possible output value. The first two theorems detail two specific cases: functions that possess one and only one true minterm (or the inverse) and functions that possess only two true minterms (or the inverse).

**Theorem 6** A function \(f(X)\) has exactly one dissimilar minterm if and only if \(C(\tau) = 2^{n-1} \forall \tau \neq 0\).

The proof is given in the Appendix.

The corollary for the \([0,1]\) encoding can be shown by applying (3) to the theorem above. The general result is as follows:

**Corollary 1** A function \(f(X)\) has exactly one dissimilar minterm if and only if \(B(\tau) = k - 1\).
It should be pointed out that this general $B(\tau)$ are quite limited. This is because for a function to have exactly one dissimilar minterm then either $k=2^a-1$, in which case $B(\tau)=2^{a-2} \forall \tau \neq 0$, or $k=1$, which results in $B(\tau)=0 \forall \tau \neq 0$.

This type of analysis of the division of true and false minterms in the function can be extended to the situation with $d$ dissimilar minterms. Initially we look at the case of $d=2$ before giving the general result.

**Theorem 7** A function $f(X)$ has exactly two dissimilar minterms if and only if

$C(0) = 2^a$,

$C(\tau) = 2^a - 2^{a-2} \forall \tau \neq 0$,

$C(\tau) = 0$ \( \forall \tau = 0 \).

The proof is given in the Appendix.

**Corollary 2** A function $f(X)$ has exactly two dissimilar minterms if and only if

$B(0) = B(\tau_0) = k$ and

$B(\tau) = k - 2 \forall \tau, \tau_0 \neq 0$ and $\tau \neq \tau_0$.

Again, although Corollary 2 states a general result, in practice the values are limited to the following:

• $B(0) = B(\tau_0) = 2$ and $B(\tau) = 0$, or

• $B(0) = B(\tau_0) = 2^a-2$ and $B(\tau) = 2^{a-2}$.

It should also be noted that this pattern of coefficients indicates that the function is either itself degenerate or is related through the application of the autocorrelation invariance operators [8] to a degenerate function.

**Theorem 8** A function $f(X)$ has $d$ dissimilar minterms if and only if the autocorrelation coefficients have the following properties:

• $C(0) = 2^a$,

• for $p \in 2, 4, 6, \ldots, d$ (or $2, 4, 6, d-1$ if $d$ is odd) $C(\tau) = 2^a - 4d$,

• for the remaining coefficients, $C(\tau) = 2^a - 4d$.

Again, the proof is given in the Appendix.

**D. Identification of Exclusive-OR Logic**

In some approaches to logic synthesis it is useful to identify a decomposition of the function that utilizes the Boolean $\oplus$ (exclusive-or) operator [9].

**Theorem 9** $C(\tau) = 2^a$ if and only if the function $f(X)$ has a decomposition $f(X) = h(X) \oplus x_i$ where $h(X)$ is independent of $x_i$.

If no first order coefficients meet the requirements for the presence of this decomposition, we then go on to include the second order coefficients in the examination. This is described in Theorem 10.

**Theorem 10** $C(\tau_j) = C(\tau_j) = C(\tau_j) = 0, \text{ if } j \text{ if and only if the function } f(X) \text{ can be decomposed into } h(X) \oplus g(X)$ where $g(X) = x_i \lambda_j \lambda_{j k} \tau, \text{ and } h(X) \text{ is independent of both } x_i \text{ and } x_j$.

$C(\tau_j)$ can be further extended to functions where $g(X) = \vee(x_j \ldots x_{m+n})$ or $g(X) = \wedge(x_j \ldots x_{m+n})$, $i \in \{1 \ldots n\}$ and $j + m \leq n$.

**Theorem 11** If $f(X)$ can be decomposed into $h(X) \oplus g(X)$ where $X \cup X_2=X$ and $X_1 \cap X_2 = \emptyset$ then $C(\tau) = C(\tau)$.

This theorem describes a situation in which a function $f(X)$ is known to have a decomposition of the format $h(X) \oplus g(X)$ where $h(X)$ is independent of all the variables in $g$. Thus $\phi = \{\emptyset\}$. In this case autocorrelation coefficients that are related to $g$’s variables for both functions $f$ and $g$ will then be equivalent. Since it is possible to construct examples such that $f(X) = h(X) \oplus g(X)$, $C(\tau_2) = C(\tau_2)$, and $X_1 \cap X_2 \neq \emptyset$, the presence of such a pattern $C(\tau_2) = C(\tau_2)$ is not strong enough to uniquely identify all exclusive-or based decompositions of this type. However, given two functions $f(X)$ and $g(X)$ it is always possible to construct $h(X)$ such that $g(X) \oplus h(X)$ since $\oplus$ is reversible and $h(X)$ can be determined by finding $f(X) \oplus g(X)$. Thus using a known library it would be possible to perform a fast determination, using the autocorrelation coefficients, of whether a mutually exclusive decomposition was likely, and then use the above to construct the second function of the decomposition. Proofs for each of these theorems are given in the Appendix.

V. USES

The above properties have been put to a variety of uses, including determining three-level decompositions [5], variable ordering and optimization of binary decision diagrams (BDDs) [2, 10], and in classifying Boolean logic functions [8]. Further work is progressing on additional uses, such as developing heuristics for Kronecker decision diagram (KDD) decomposition selections and for determining whether a BDD, KDD or functional decision diagram (FDD) is a better representation for a function [11]. Below we investigate the use of autocorrelation coefficients in identifying properties such as symmetries, linearity and self-duality of functions.

A. Totally Symmetric Functions

There are a number of different types of symmetries. We begin with the most restrictive symmetry. A Boolean function is said to be totally symmetric if the output is unchanged by any permutation of the inputs to the function. For example, $f = x_1 x_2 x_3$ is totally symmetric, as is the majority function $f = x_1 x_2 x_3 + x_1 x_2 x_3$. We also discuss a recently introduced type of symmetry termed antisymmetries [12]. An antisymmetry occurs when permuting all or a subset of variables results in the exact inverse of the original function.

**Theorem 12** If a function $f(X)$ is totally (anti)symmetric then all $\{+1,-1\}$ autocorrelation coefficients for any given order will be equal within the order. This may be written as $C(\tau) = C(\tau^{'}) \forall \tau, \tau^{'} \text{ such that } |\tau| = |\tau^{'}|$. 

**Proof:** Work in [8] showed that permuting any two variables $j$ and $k$ results in exchanging the values of the coefficients $C(\tau_0)$ and $C(\tau_0)$. Since a function symmetric in two variables $j$ and $k$ by definition will not change if $j$ and $k$ are permuted then the autocorrelation coefficients will also not change — the function remains the same. Thus for $C(\tau_0)$ and $C(\tau_0)$ to be exchanged and yet no change to occur, we must have $C(\tau_0) = C(\tau_0)$. A function that is totally symmetric will not change for any permutation of its
variables, so \( C(\tau_{a0}) = C(\tau_{b0}) = \ldots = C(\tau_{n0}) \), assuming the
variables are numbered from 1 to \( n \). We can express this as
\( C(\tau) = C(\tau^*) \) since \( |\tau| = |\tau^*| \) if \( \alpha \) can take on any value and
the property still holds.

We note that this implies that there are non-totally-symmetric functions with coefficients of this
pattern. An example of this is given below in Subsection V-B. However, such a function must be related through the
application of one or more invariance operators to some
totally symmetric function. We expand upon this in Section VI. We note also that the above theorem includes totally
antisymmetric functions, since the negation of a function

\[
\begin{array}{|c|c|}
\hline
\text{Symmetry} & \text{Definition} \\
\hline
E[x_1, x_2] & f(x_1, x_2, 0, 1) = f(x_1, x_2, 1, 0) \\
N[x_1, x_2] & f(x_1, x_2, 0, 1) = f(x_1, x_2, 1, 0) \\
S[x_1, x_2] & f(x_1, x_2, 0, 1) = f(x_1, x_2, 1, 0) \\
S[x_1, x_2] & f(x_1, x_2, 0, 1) = f(x_1, x_2, 1, 0) \\
\hline
\end{array}
\]

does not affect the \{+1, -1\} autocorrelation coefficients.

### B. Partially Symmetric Functions

A slightly less restrictive form of symmetry is that of
partial symmetry. A Boolean function is said to be partially
symmetric, or possess a partial symmetry if the output is
unchanged by any permutation of some subset of the inputs to
the function.

**Theorem 13** If a function \( f(X) \) is partially symmetric in a
subset of its input variables \( x_{i_1}, \ldots, x_{i_m} \) then the
autocorrelation coefficients \( C(\tau_{i_1}) \) will have equal values for
all \( \tau \in \{i_1, \ldots, i_m\} \).

The same reasoning as used above for Theorem 12 can be
used here. Permuting any \( m \) variables \( x_{i_1} \) through \( x_{i_m} \) results in
exchanging the values of the coefficients \( C(\tau_{i_1}) \) through \( C(\tau_{i_m}) \).
However, the function does not change, by definition, and so coefficients \( C(\tau_{i_1}) \) through \( C(\tau_{i_m}) \) must be
equal. For example, the function

\[
f(X) = x_1 x_2 + x_3 x_4 + x_1 x_2 x_3 + x_1 x_2 x_3 x_4
\]
is partially symmetric in \( x_1, x_2, x_3, x_4 \). Although it appears that the product
\( x_1 x_2 \) is missing it is unnecessary as it is covered by the other
products. The autocorrelation coefficients for this function are
given in Table I. Of note are the sets of values (for \( \tau \)) 1000, 0100 and 0010, and 1001, 0101 and 0011 which illustrate the
theorem above.

### C. Functions with Symmetries of Degree Two

A third type of symmetry is a symmetry of degree two. This
is a partial symmetry in which two sub-functions of the
original function are identical and also are independent of
two of the function's variables. Symmetries of degree two are
identified by finding patterns where \( f(x_1, \ldots, a, \ldots, b, \ldots, x_d) = \)
\( f(x_1, \ldots, c, \ldots, d, \ldots, x_d) \), \( a, b, c, d \in \{0, 1\} \). Equivalence (E),
non-equivalence (N) and single-variable (S) symmetries as
defined by Hurst, Miller and Muzzio are all types of
symmetries of degree two, and are defined in Table II [13].
Without loss of generality these definitions label the two
variables of interest as \( n \) and \( n-1 \). \( x^* \) refers to the the inverse
of \( x \).

Antisymmetries can also be extended to the symmetries of
degree two. For instance, an anti-equivalence symmetry is
usually denoted \( \overline{E} \{x_{n-1}, x_n\} \).

**Theorem 14** A function \( f(X) \) with some type of
(anti)symmetry of degree two will have autocorrelation
coefficient values as follows:

\[
E\{x_i, x_j\} \quad \text{or} \quad N\{x_i, x_j\} \rightarrow C(\tau_{ij}) = C(\tau_{ji}) ,
\]

\[
S\{x_i \mid x_j\} \quad \text{or} \quad S\{x_i \mid x^{-1}_j\} \rightarrow C(\tau_{ij}) = C(\tau_{ji}) , \quad \text{and}
\]

\[
S\{x_i \mid x_j\} \quad \text{or} \quad S\{x_i \mid x^{-1}_j\} \rightarrow C(\tau_{ij}) = C(\tau_{ji}) .
\]

Proofs for these are given in [14].

\[
\begin{array}{|c|c|c|c|c|c|c|}
\hline
\tau & C(\tau) & \tau & C(\tau) & \tau & C(\tau) & \tau & C(\tau) \\
\hline
000 & 0 & 16 & 0 & 100 & 0 & 4 & 0 \\
000 & 4 & 010 & 1 & 4 & 0 & 100 & 1 \\
000 & 4 & 011 & 12 & 010 & 0 & 12 & 0 \\
000 & 4 & 011 & 1 & 4 & 101 & 1 & 4 & 1
\hline
\end{array}
\]

**Table I:** \{+1, -1\} AC coefficients for the partially symmetric
function \( f(X) = \overline{x} \overline{x} x + \overline{x} x_4 + x_3 x_4 + x_1 x_2 + x_1 x_3 \).

**Table II:** Definitions and notation for equivalence, non-equivalence, and single-variable symmetries.

### D. Is it Possible to Determine Symmetries from the
Autocorrelation Coefficients?

Hurst, Miller and Muzzio provide tests based on a function's
spectral coefficients that will ascertain whether or not the
function possesses a particular symmetry [13]. However, as
indicated by the example in Table I, the autocorrelation
coefficients cannot be used in the same way. This can be
explained by examining the spectral symmetry tests, as
described in Table III. The notation used in this table is as
follows:

- \( S^4 \) includes all spectral coefficients that involve neither
  \( x_i \) or \( x_j \),
- \( S^4 \) includes all spectral coefficients that involve \( x_i \) but not
  \( x_j \),
- \( S^4 \) includes all spectral coefficients that involve \( x_j \) but not
  \( x_i \), and
- \( S^4 \) includes all spectral coefficients that involve both \( x_i \)
  and \( x_j \).

The spectral coefficients are computed using

\[
T^m \cdot X = S .
\]

For example, for a \( n=3 \) Boolean function,
$T^n = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & 1 & -1 \end{bmatrix},$  

$Y$ is the output vector of the function, for example  

$Y = \begin{bmatrix} 1 & y_0 \\ -1 & y_1 \\ -1 & y_2 \\ 1 & y_3 \\ 1 & y_4 \\ 1 & y_5 \\ 1 & y_6 \\ -1 & y_7 \end{bmatrix},$  

and $S$ is the resulting spectral coefficients. Using the sample function from above, the coefficients would be  

$S = \begin{bmatrix} 2 & s_0 \\ 2 & s_1 \\ 2 & s_2 \\ 2 & s_{12} \\ -2 & s_3 \\ -2 & s_{13} \\ -2 & s_{23} \\ 6 & s_{123} \end{bmatrix}.$  

Examination of the spectral symmetry tests for three variables illustrates that if $\begin{bmatrix} s_2 \\ s_{12} \end{bmatrix} = \begin{bmatrix} s_3 \\ s_{13} \end{bmatrix}$ then the function must possess $N(x_2, x_3).$ Similarly, if $\begin{bmatrix} s_2 \\ s_{12} \end{bmatrix} = \begin{bmatrix} -s_3 \\ -s_{13} \end{bmatrix}$ then the function must possess $E(x_2, x_3).$ The notation used here for labeling of coefficients is as illustrated in the example above.  

In the autocorrelation coefficients, this distinction is lost. This brings to question the following situation. If $\begin{bmatrix} s_2 \\ s_{12} \end{bmatrix} = \begin{bmatrix} -s_3 \\ -s_{13} \end{bmatrix}$ then the autocorrelation coefficients will still be equal; however, the symmetries do not exist. The same holds true if $s_2 = s_3$ and $s_{12} = -s_{13}.$ Therefore it is not possible to determine if a function has a particular equivalence, nonequivalence or single variable symmetry solely by examining the autocorrelation coefficients. The same holds true for totally and partially symmetric functions.

**E. Self-Dual & Self-Anti-Dual Functions**

**Definition 5.1** The dual of a function $f(x_1, x_2, ..., x_n)$ is $\overline{f}(\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$ and is denoted by $f^d.$

$f^d$ is obtained first by replacing each literal $x_i$ with $\overline{x}_i$ and then by complementing the function. A self-dual function is a function such that $f = f^d.$ There are $2^{2n-1}$ self-dual functions of $n$ variables. A self-anti-dual function is a function such that $f = f(\overline{x}, \overline{x}, ..., \overline{x}).$

**Theorem 15** A function $f(X)$ will have $C(2^n-1) = -2^n$ if and only if it is a self-dual function. Similarly, a function will have $C(2^n-1) = 2^n$ if and only if it is a self-anti-dual function.

**Proof:** If a function is self-dual, then by definition $f(X) = \overline{f}(\overline{X})$, which can be rewritten as $f(X) = \overline{f}(X \oplus 2^{n-1})$. Using $\{+1,-1\}$ notation $f(X) \overline{f}(X) = -1.$ Then by definition

$$C(2^n-1) = \sum_{v=0}^{2^n-1} f(v) \cdot f(v \oplus 2^n-1)$$

and thus

$$= \sum_{v=0}^{2^n-1} f(v) \cdot \overline{f}(v) = -2^n.$$

**Table III:** Spectral symmetry tests for symmetries in $(x_0, x_1, x_2)$ that refers to the inverse of $x_3.$

<table>
<thead>
<tr>
<th>Symmetry</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S[x_0, x_1]$</td>
<td>$S^t + S^t = 0$</td>
</tr>
<tr>
<td>$S[x_0, x_1]$</td>
<td>$S^t + S^t = 0$</td>
</tr>
<tr>
<td>$E[x_0, x_1]$</td>
<td>$S^t + S^t = 0$</td>
</tr>
<tr>
<td>$N[x_0, x_1]$</td>
<td>$S^t - S^t = 0$</td>
</tr>
<tr>
<td>$S[x_0, x_1]$</td>
<td>$S^t - S^t = 0$</td>
</tr>
<tr>
<td>$S[x_0, x_1]$</td>
<td>$S^t - S^t = 0$</td>
</tr>
</tbody>
</table>

Similarly, for self-anti-dual functions, by definition $f(X) = f(\overline{X})$, which can be rewritten $f(X) = f(X \oplus 2^{n-1})$ and so again, by definition

$$C(2^n-1) = \sum_{v=0}^{2^n-1} f(v) \cdot \overline{f}(v \oplus 2^n-1)$$

and thus

$$= \sum_{v=0}^{2^n-1} f(v) \cdot \overline{f}(v) = 2^n.$$  

If $C(2^n-1) = -2^n$ then every pair of minterms $f(v)$ and $f(v \oplus 2^n-1)$ in the summation $\sum_{v=0}^{2^n-1} f(v) \cdot f(v \oplus 2^n-1)$ must result in a -1 when multiplied and thus must have inverse values of each other. So $f(v) = \overline{\overline{f}(v \oplus 2^n-1)}$, or, $f(v) = f(v)$, which is the definition of a self-dual function. Similarly, if $C(2^n-1) = 2^n$ then every pair of minterms $f(v)$ and $f(v \oplus 2^n-1)$ in the summation $\sum_{v=0}^{2^n-1} f(v) \cdot f(v \oplus 2^n-1)$ must result in a 1 when multiplied and thus must have identical values. So $f(v) = f(v \oplus 2^n-1)$, or, $f(v) = f(v)$, which is the definition of a self-anti-dual function.
F. Linear Functions

Definition 5.2 If a logic function \( f \) is represented as
\[ f = a_0 \oplus a_1 x_1 \oplus a_2 x_2 \oplus \ldots \oplus a_n x_n \] where \( a_i = 0 \) or \( 1 \) then \( f \) is said to be a linear function [9].

It is interesting to note that a linear function is either a self-dual or self-anti-dual function. The proof is given in [9]. There are \( 2^{n+1} \) linear functions of \( n \) variables, and a linear function that is obtained by assigning a linear function to an arbitrary variable of a linear function is also a linear function.

Theorem 16 A function \( f(X) \) is linear if and only if all of its coefficients \( C(\tau) = -2^\tau \), such that the weight of \( \tau \) \( |\tau| = 1 \).

This theorem follows directly from Theorem 9. We extend this theorem further to specify that if all values of \( \tau \) with a single one in the binary expansion result in \( C(\tau) = -2^\tau \) then the function must be decomposable for all variables in the fashion described above.

VI. DISCUSSION AND FUTURE WORK

As noted in Sections V-A and V-B, identifying where a function has equal coefficients within a given order, or in a subset of that order, is not sufficient to identify a symmetric function. However, other work has identified that a function does not have a symmetry but whose autocorrelation coefficients reflect this property must be in the same autocorrelation class as some totally/partially symmetric function [8]. Thus it may be possible to identify the necessary operations to apply in order to transform the subject function into a symmetric function, thus making it possible to leverage the advantages inherent in symmetries. Future work will address tools to make this determination.

A comment on the suitability of the autocorrelation transform as an analysis tool is appropriate; the authors have found that properties defined on the outputs of a function are better suited to analysis with autocorrelation coefficients than are properties defined based on the structure of a function. For instance, the properties of self-duality and self-anti-duality lend themselves very nicely to identification through autocorrelation coefficients, while on the other hand monotone functions are much more difficult to identify.

VII. CONCLUSION

There are many existing techniques for the identification of properties such as symmetries, including [15, 16] and [17]. Rather than competing with these, this paper concentrates instead on the theoretical aspects of the autocorrelation transform as an analysis tool. We can conclude from this work that the autocorrelation transform can identify if a function does not possess a symmetry, but that the autocorrelation coefficients resulting from the transform do not provide a sufficient condition for the existence of symmetries. Ongoing work in this area includes implementation of our technique in order that we may compare it with existing techniques, as well as the various directions described in Section 6. An extension of the analysis led to necessary and sufficient conditions for the identification of self-dual/self-anti-dual and linear functions. Future work will include implementations for these properties as well.

This paper presents an exploration of the properties inherent to the autocorrelation transform as applied to single-output completely specified boolean functions. Various uses have been suggested in other publications. The ultimate goal of this work is to develop a preprocessing tool which will be used to aid chip designers in making choices prior to or during the design process. For instance, if one can quickly identify that a function cannot result in a non-exponential BDD then much optimization time will be saved by beginning work with a KDD representation. The properties described here are being used in the development of such a tool.

Other avenues for future work include extending this research to the incompletely specified and multiple-output cases.

APPENDIX -- PROOFS

Theorem 6: A function \( f(X) \) has exactly one dissimilar minterm if and only if
\[ C(\tau) = 2^{-4} \forall \tau \neq 0. \]

Proof: Without loss of generality let us define a function \( f \) such that \( f(v)=1 \) when \( v = 0, \ldots, 2^n \), and \( f(v) = -1 \) when \( v = 2^n - 1 \).

Then
\[ C(\tau) = \sum_{v=0}^{2^n-1} f(v) \times f(v \oplus \tau) \]
\[ = \left( \sum_{v=0}^{2^n-1} f(v) \times f(v \oplus \tau) \right) \]
\[ + f(2^n - 1) \times f(2^n - 1 \oplus \tau) \]
\[ = \left( \sum_{v=0}^{2^n-1} 1 \times f(v \oplus \tau) \right) + (-1) \times 1 \]
\[ = (2^n - 2 - 1) - 1 \]
\[ = 2^n - 4 \forall \tau \neq 0. \]

Thus if \( f(X) \) has exactly one true minterm then all of the coefficients \( C(\tau) = 2^{-4}, \tau \neq 0 \).

For the second part of this proof, if all that is known of the function is the coefficients of this pattern, then it can be shown as follows that the function must have either exactly one true or exactly one false minterm.

For a coefficient \( C(\tau) \) let us define \( q \) as the number of positive pairs in the summation, and \( r \) as the number of negative pairs in the summation. A pair in this case is a combination of two minterms \( i, j \), and a positive pair results when both minterms are true or when both are false. It should be noted that in the summation for the autocorrelation equation each pair is encountered twice. Then \( 2q-2r = 2^{n-4} \) and \( 2q+2r = 2^n \).

These equations can be solved to show that \( r = 1 \). If there is only one negative pair in the summation then there is only one pair combining a true and a false minterm; all other pairs must combine either two true minterms or two false minterms. If there is only one coefficient \( C(\tau) \) for which this
holds, then there can be any number of combinations of true and false minterms to meet these requirements. However,
there are \( 2^n \cdot i \) coefficients that have only one negative pair; therefore there can be only one dissimilar minterm in the
function.

**Theorem 7**: A function has exactly two dissimilar minterms if and only if \( C(0) = 2^n \), \( C(\tau_0) = 2^n \), and \( C(\tau) = 2^n + 8 \)
\( \forall \tau, \tau_0 \neq 0 \) and \( \tau \neq \tau_0 \).

**Proof**: We approach this proof by first demonstrating that
if there is one coefficient \( C(\tau_0) = 2^n \), \( \tau \neq 0 \) and the remaining
\( 2^n - 2 \) coefficients \( C(\tau) = 2^n - 8 \), then the function has exactly
two dissimilar minterms. Let us define a function \( f \) such that
\( f(v) = 1 \) when \( v \in \{0, \ldots, 2^n\} \), \( v \neq i, j \) and \( f(v) = 1 \) when \( v = i, j \).
Without loss of generality let \( i = 0 \) and \( j = 1 \). Then

\[
C(\tau) = \sum_{i=0}^{2^n-1} f(v) \times f(v \oplus \tau)
\]

\[
= f(i) \times f(i \oplus \tau) + f(j) \times f(j \oplus \tau)
\]

\[
= \sum_{i=0}^{2^n-1} f(i) \times f(i \oplus \tau) + \sum_{i=0}^{2^n-1} f(j) \times f(j \oplus \tau)
\]

Then if \( i \oplus \tau = j \) and \( j \oplus \tau = i \), \( C(\tau) = 2^n \). Otherwise \( C(\tau) =
2^n + (2^n - 2) = 2^n - 8 \). Because of the nature of the \( \oplus \) operator,
\( i \oplus \tau = j \iff j \oplus \tau = i \), and so there is only one assignment of \( \tau \)
for which this can occur.

A similar process to that shown in the proof of Theorem 6 can be used to prove that the pattern of coefficients can only result in a function with exactly two dissimilar minterms. **Theorem 8**: A function has \( d \) dissimilar minterms if and only if the autocorrelation coefficients have the following properties:

- \( C(0) = 2^n \),
- for \( \binom{d}{p} \), \( p \in \{2, 4, 6, \ldots, d \} \), \( \left( \begin{array}{c} 
2 \n \end{array} \right) \), \( \binom{d}{d-1} \) if \( d \) is odd \( C(\tau) = 2^n - 4d + 4p \), and
- for the remaining coefficients, \( C(\tau) = 2^n - 4d \).

The proof is similar to those for Theorems 6 and 7.

**Proof**: Let us define a function \( f(X) \) for which there are \( d \)
dissimilar minterms. Without loss of generality we assume that
\( f(v) = 1 \) when \( v \in \{0, \ldots, d-1\} \) and \( f(v) = 1 \) when \( v \in \{d, \ldots, 2^n-1\} \). Then there are \( d-p \) mod 2 ways (resulting in
\( \binom{d}{2} \), \( \binom{d}{4} \), \( \binom{d}{6} \), \( \ldots \) or \( \binom{d}{2} \), \( \binom{d}{4} \), \( \binom{d}{6} \), \( \ldots \) ) in which pairs of dissimilar minterms may match up, resulting in

\[
C(\tau) = 2p - 2d + \sum_{i=0}^{2^n-1} f(v) \times f(v \oplus \tau)
\]

where the first term \( 2p \) is the result of the sum of the matching dissimilar minterms, the second term \( 2(d-p) \) is the result of the sum of the non-matching dissimilar minterms, and the final term is the sum of the remaining minterms which are all similar.

There are also coefficients resulting from the situation in which none of the dissimilar coefficients match in the summation:

\[
C(\tau) = -2d + \sum_{i=0}^{2^n-1} f(v) \times f(v \oplus \tau)
\]

\[
= 2^n - 4d
\]

Again, using a similar technique to that shown in the proof of
Theorem 6, if \( q \) is the number of positive pairs and \( r \) is the number of negative pairs then \( 2q + 2r = 2^n \) and \( 2q - 2r = 2^n - 4d \)
which results in \( r = d \).

**Theorem 9**: \( C(\tau_n) = -2^n \) if and only if the function \( f(X) \) has a decomposition \( f(X) = f^*(X) \oplus x \), where \( f^*(X) \) is independent of \( x \).

**Proof**: We first determine that a function with the decomposition \( f(X) = f^*(X) \oplus x \) has a first order autocorrelation coefficient \( C(\tau) = -2^n \).

Without loss of generality let \( i = n \). Then

\[
C(\tau) = \sum_{i=0}^{2^n-1} f(v) \times f(v \oplus \tau)
\]

\[
= \sum_{i=0}^{2^n-1} [f(v \oplus x) \times f(v \oplus (v \oplus \tau) \oplus (x \oplus \tau))]
\]

\[
= \sum_{i=0}^{2^n-1} (f(v \oplus 0) \times (f(v \oplus \tau) \oplus (0 \oplus \tau))
\]

\[
= \sum_{i=0}^{2^n-1} (f(v \oplus 1) \times (f(v \oplus \tau) \oplus (1 \oplus \tau))
\]

\[
= \sum_{i=0}^{2^n-1} f(v) \times (-f(v \oplus \tau))
\]

\[
= -2^n.
\]
function is the inverse of the other half, which can be achieved by defining a function \( f(X) \) as \( f(X) = \phi(X) \otimes x_i \).

**Theorem 10:** \( C(\tau_i) = C(\tau_j) = C(\tau_k) = 0 \) if \( i \neq j \) and only if the function \( f(X) \) can be decomposed into \( h(X) \otimes g(X) \) where \( g(X) = x_i \otimes x_j, \) \( i \in \{ \land, \lor \} \), and \( h(X) \) is independent of both \( x_i \) and \( x_j \).

**Proof:** Let us define a function \( f(X) = f^a(X) \otimes g(X) \) where \( g(X) = x_i \lor x_j \) and \( f^a(X) \) is independent of \( x_i \) and \( x_j \), and let us assume without loss of generality that \( i = n \) and \( j = n-1 \). Then

\[
C(\tau) = \sum_{v \in \{ 0, 1 \}} f(v_i) \times f(v \oplus \tau)
= \sum_{v} f(v_i) \times f(v_1 \oplus \tau) + \sum_{v} f(v_2) \times f(v_2 \oplus \tau)
+ \sum_{v} f(v_3) \times f(v_3 \oplus \tau) + \sum_{v} f(v_4) \times f(v_4 \oplus \tau)
\]

where

- A: \( v_1 = 0 \) to \( 2^{n-2} \) - 1 (0000 ... 0011),
- B: \( v_2 = 2^{n-2} \) to \( 2^{n-1} \) - 1 (0100 ... 0111),
- C: \( v_3 = 2^{n-1} \) to \( 2^n - 2^{n-2} - 1 \) (1000 ... 1011), and
- D: \( v_4 = 2^n - 2^{n-2} \) to \( 2^n - 1 \) (1100 ... 1111).

Then \( C(\tau_{n-1}) = \sum_{v} [(f^a(v_i) \otimes x_n \land x_{n-1})]
\]

\[
\times (f^a(v_1 \oplus \tau_{n-1}) \otimes (x_n \land x_{n-1} \lor \tau_{n-1}))
\]

\[
+ \sum_{v} [(f^a(v_2) \otimes x_n \land x_{n-1})
\times (f^a(v_2 \oplus \tau_{n-1}) \otimes (x_n \land x_{n-1} \lor \tau_{n-1}))]
\]

\[
+ \sum_{v} [(f^a(v_3) \otimes x_n \land x_{n-1})
\times (f^a(v_3 \oplus \tau_{n-1}) \otimes (x_n \land x_{n-1} \lor \tau_{n-1}))]
\]

\[
+ \sum_{v} [(f^a(v_4) \otimes x_n \land x_{n-1})
\times (f^a(v_4 \oplus \tau_{n-1}) \otimes (x_n \land x_{n-1} \lor \tau_{n-1}))]
\]

\[
= \sum_{v} [(f^a(v_i) \otimes 0 \land 0)
\times (f^a(v_1 \oplus \tau_{n-1}) \otimes (0 \land 0 \lor \tau_{n-1}))]
\]

\[
+ \sum_{v} [(f^a(v_2) \otimes 0 \land 1)
\times (f^a(v_2 \oplus \tau_{n-1}) \otimes (0 \land 1 \lor \tau_{n-1}))]
\]

\[
+ \sum_{v} [(f^a(v_3) \otimes 1 \land 0)
\times (f^a(v_3 \oplus \tau_{n-1}) \otimes (1 \land 0 \lor \tau_{n-1}))]
\]

\[
+ \sum_{v} [(f^a(v_4) \otimes 1 \land 1)
\times (f^a(v_4 \oplus \tau_{n-1}) \otimes (1 \land 1 \lor \tau_{n-1}))]
\]

\[
= \sum_{v} [(f^a(v_i) \otimes 0) \land 0)
\times (f^a(v_1 \oplus \tau_{n-1}) \otimes (0 \land 0 \lor \tau_{n-1}))]
\]

\[
+ \sum_{v} [(f^a(v_2) \otimes 0 \land 1)
\times (f^a(v_2 \oplus \tau_{n-1}) \otimes (0 \land 1 \lor \tau_{n-1}))]
\]

\[
+ \sum_{v} [(f^a(v_3) \otimes 1 \land 0)
\times (f^a(v_3 \oplus \tau_{n-1}) \otimes (1 \land 0 \lor \tau_{n-1}))]
\]

\[
+ \sum_{v} [(f^a(v_4) \otimes 1 \land 1)
\times (f^a(v_4 \oplus \tau_{n-1}) \otimes (1 \land 1 \lor \tau_{n-1}))]
\]

\[
= \sum_{v} f(v_i) \times f(v_i \oplus \tau_{n-1})
+ \sum_{v} f(v_2) \times f(v_2 \oplus \tau_{n-1})
+ \sum_{v} f(v_3) \times f(v_3 \oplus \tau_{n-1})
+ \sum_{v} f(v_4) \times f(v_4 \oplus \tau_{n-1})
\]

\[
= 2^{n-2} + 2^{n-2} - 2^{n-2} - 2^{n-2} = 0
\]

and similarly for \( C(\tau_{n-2}) \) and \( C(\tau_{n-11}) \).

If \( C(\tau_0) = C(\tau_{n-1}) = C(\tau_{n-19}) = 0 \) then each of the summations may be broken down into \( C(\tau) = 2^{n-2} + 2^{n-2} - 2^{n-2} - 2^{n-2} \). Let us assume there exists some variable ordering such that

\[
\sum_{v} f(v_i) \times f(v_i \oplus \tau_{n-1}) = 2^{n-2}
+ \sum_{v} f(v_2) \times f(v_2 \oplus \tau_{n-1}) = 2^{n-2}
+ \sum_{v} f(v_3) \times f(v_3 \oplus \tau_{n-1}) = -2^{n-2}
+ \sum_{v} f(v_4) \times f(v_4 \oplus \tau_{n-1}) = -2^{n-2}
\]

Then the first two summations tell us that for part of the function \( f(v) \) is independent of variable \( x_{n-1} \) and the second two indicate that for part of the function \( f(v) \) contains \( \otimes x_{n-1} \). This indicates that the solution must be of the form \( f(X) = f^a(X) \otimes g(X) \) where \( f^a(X) \) is independent of \( x_{n-1} \) and \( g(X) \) contains \( x_{n-1} \). The same process is then applied to the other known coefficients, \( C(\tau_{n-2}) = C(\tau_{n-19}) = 0 \). There are two possible solutions:

**Solution 1**

\[
\tau_{n-1} \quad \tau_n \quad \tau_{n-11}
\]

\[
\sum_{v} f(v_i) \times f(v_i \oplus \tau) = 2^{n-2} \quad 2^{n-2} \quad -2^{n-2}
\]

\[
\sum_{v} f(v_2) \times f(v_2 \oplus \tau) = 2^{n-2} \quad -2^{n-2} \quad 2^{n-2}
\]

\[
\sum_{v} f(v_3) \times f(v_3 \oplus \tau) = -2^{n-2} \quad 2^{n-2} \quad 2^{n-2}
\]

\[
\sum_{v} f(v_4) \times f(v_4 \oplus \tau) = -2^{n-2} \quad -2^{n-2} \quad -2^{n-2}
\]

The above is obtained for \( g(X) = x_i \land x_2 \).

**Solution 2**

\[
\tau_{n-1} \quad \tau_n \quad \tau_{n-11}
\]

\[
\sum_{v} f(v_i) \times f(v_i \oplus \tau) = -2^{n-2} \quad -2^{n-2} \quad 2^{n-2}
\]

\[
\sum_{v} f(v_2) \times f(v_2 \oplus \tau) = -2^{n-2} \quad 2^{n-2} \quad 2^{n-2}
\]

\[
\sum_{v} f(v_3) \times f(v_3 \oplus \tau) = 2^{n-2} \quad -2^{n-2} \quad 2^{n-2}
\]

\[
\sum_{v} f(v_4) \times f(v_4 \oplus \tau) = 2^{n-2} \quad 2^{n-2} \quad 2^{n-2}
\]

The above is obtained for \( g(X) = x_i \lor x_2 \).

The proof is easily extended to any number of variables in

---

1. Four variable expansions are given for the sake of clarity only. This does not limit the proof to four variables.
g since there is always only one input combination for which 
\( g(X) = x_i \land x_{i+1} \land \ldots \land x_{i+m} = 1 \) and similarly where 
\( g(X) = x_i \lor x_{i+1} \lor \ldots \lor x_{i+m} = 0 \).

Theorem 11: If \( f(X) \) can be decomposed into \( h(X1) \oplus g(X2) \) 
where \( X1 \cup X2 = X \) and \( X1 \cap X2 = \emptyset \) then 
\( C^h(\tau_{X2}) = C^h(\tau_{X1}) \).

Proof: Without loss of generality, let us assume that \( X2 \) consists of some 
combination of the variables \( x_i, x_{i+1}, \ldots, x_{i+m} \) and \( X1 \) consists of 
the remaining variables from \( X \). Then given a function \( f(X) \) that is 
decomposable into \( h(X1) \oplus g(X2) \) as described in the theorem above, by definition we have \( h(X) = h(X \oplus X2) \); in other words, \( h \) is not affected 
if any of the variables in \( X2 \) are changed. Then, noting that in \{+1,-1\} 
notation performing an \( \oplus \) operation is the same as the 
mathematical multiplication operation, the following holds:

\[
C^h(\tau_{X2}) = \sum_{i=0}^{2^n-1} f(v) \times f(v \oplus \tau_{X2}) \\
= \sum_{i=0}^{2^n-1} [h(v) \oplus g(v)] \times [h(v \oplus \tau_{X2}) \oplus g(v \oplus \tau_{X2})] \\
= \sum_{i=0}^{2^n-1} h(v) \times h(v \oplus \tau_{X2}) \times g(v) \times g(v \oplus \tau_{X2}) \\
= \sum_{i=0}^{2^n-1} (h(v))^2 \times g(v) \times g(v \oplus \tau_{X2}) \\
= \sum_{i=0}^{2^n-1} 1 \times g(v) \times g(v \oplus \tau_{X2}) \\
= C^0(\tau_{X2})
\]

REFERENCES

[1] R. Tomczuk, Autocorrelation and decomposition methods in 

for variable ordering for ROBDDs, in: Proceedings of the 4th 
International Workshop on Applications of the Reed-Müller Expansion in 


in: Proceedings of the 4th International Workshop on 

of three-level decompositions, in: Proceedings of the International 

[6] R. S. Stankovic, M. G. Karpovsky, Remarks on calculation of 
aucorrelation on finite dyadic groups by local transformations of 
decision diagrams, in: R. Moreno-Daz, F. Pichler, A. Quesada-Arencibia 
(Eds.), EUROCAST, Vol. 3643 of Lecture Notes in Computer Science, 


[8] J. E. Rice, J. C. Muzio, Use of the autocorrelation function in the 
classification of switching functions, in: Proceedings of the Euromicro 
Symposium on Digital System Design: Architectures, Methods and Tools 
(DSD), 2002, pp. 244–251.


for BDD optimizing by using autocorrelation methods, in: Proceedings of 

[11] J. E. Rice, Making a choice between FDDs and BDDs, in: 
Proceedings of the International Workshop on Logic Synthesis (IWLS), 

functions, in: Proceedings of the International Symposium on Circuits and 


[14] J. E. Rice, Autocorrelation coefficients in the representation and 
classification of switching functions, Ph.D. thesis, University of Victoria 
(2003).

functions represented by ROBDDs, in: Proceedings of the International 

variable symmetry in Boolean functions through Walsh transform, in: 
Proceedings of the International Symposium on Circuits and Systems 
(ISCAS), 2002, pp. 745–748.

variable ordering of decision diagrams, in: Proceedings of International 