Forward and Inverse Transformations Between Haar Spectra and Ordered Binary Decision Diagrams of Boolean Functions

Bogdan J. Falkowski, *Senior Member, IEEE*, and
Chip-Hong Chang, *Student Member, IEEE*

Abstract—Unnormalized Haar spectra and Ordered Binary Decision Diagrams (OBDDs) are two standard representations of Boolean functions used in logic design. In this article, mutual relationships between those two representations have been derived. The method of calculating the Haar spectrum from OBDD has been presented. The decomposition of the Haar spectrum, in terms of the cofactors of Boolean functions, has been introduced. Based on the above decomposition, another method to synthesize OBDD directly from the Haar spectrum has been presented.

Index Terms—Boolean functions, Haar spectrum, Haar transform, ordered binary decision diagram, Shannon decomposition, spectral techniques.

1 Introduction

THE various transformations which map a data vector (truth table) onto another vector have been used in digital logic design for more than 30 years. Spectral techniques have been applied to Boolean function classification, disjoint decomposition, parallel and serial linear decomposition, spectral translation synthesis (extraction of linear pre- and post-filters), multiplexer synthesis, prime implicant extraction, threshold logic synthesis, state assignment, and testing and evaluation of logic complexity [14], [16], [22], [30], [31], [34]. The renewed interest in applications of spectral methods in design of VLSI digital circuits is caused by their excellent design for testability properties and the possibility of performing the decomposition with gates other than the ones used in most classical approaches.

There are at least two transforms which are based on square-wave-like functions that are well suitable for Boolean functions: Haar and Walsh transforms. All but two basis functions in the Haar transform consist of a square wave pulse located on an otherwise zero amplitude interval. When applied to logic design, an unnormalized Haar transform [16], [17], [30], [34] is usually used. An extension of the Haar transform to deal with an incompletely specified Boolean function, where each spectral coefficient has an easy interpretation in the terms of basic logic gates, has recently been introduced [8]. The Walsh functions are global, like the Fourier functions, and consist of a set of irregular rectangular waveforms with only two amplitude values +1 and −1 [1], [13], [14], [15], [16], [33], [34]. Walsh spectral coefficients of Boolean functions have easy interpretation, and efficient methods of calculation of such spectra directly from the reduced representation of Boolean functions, in the form of disjoint cube representation for different Walsh orderings, have recently been introduced [7]. Computation of the fast Haar transform (FHT) requires order N (N is a number of spectral coefficients) additions and subtractions, which makes it much faster than the fast Walsh transform (FWT) [1], [4], [25], [30], [32], [33], [34]. Hardware-based fast Haar chips have been developed [4]. Due to its low computing requirements, the Haar transform has been used mainly for pattern recognition and image processing [33], [34]. Such a transform is also well suited in communication technology for data coding, multiplexing, and digital filtering [17], [29], [34]. The Haar system is a prototype for wavelets and has many, but not all, of the properties of orthogonal wavelets [32]. The advantages of computational and memory requirements of the Haar transform make it of big interest to VLSI designers as well. For example, the authors of [26], [27] presented a set of CAD tools to perform a switch-level fault detection and diagnosis of physical faults for practical MOS digital circuits using a reduced Haar spectrum analysis. In their system, the unnormalized reduced Haar binary spectrum was used as a means not only for diagnosing digital MOS ICs as a tool external to the circuit, but also as a possibility for a self-test strategy. The use of this set of CAD tools allowed the derivation of strategies for testing MOS circuits when memory states were encountered as a consequence of some fault types. The advantage of using Haar functions instead of Walsh functions in CAD systems based on spectral methods for some classes of Boolean functions was shown in [16], [34]. For example, the analysis in [16] shows that the spectral complexity of conjunction and disjunction increases with the number of variables, exponentially for the Walsh functions and only linearly for the Haar functions. The circuit of the spectral multifunctional logical module [16], [17], [34] to generate arbitrary Boolean functions is shown in Fig. 1. It consists of a generator of basis functions, an adder, a multiplier, and the memory to store spectral coefficients. The module can be reprogrammed by changing dynamically its memory content. Such a behavior of the module is useful in real-time adaptive control systems [17], [34]. Karpovsky [16] noticed that the size of the memory block can be optimized only when the Haar basis is used. It is due to the fact that the number of nonvanishing Haar coefficients is reduced with input permutation of variables—the situation which does not apply to the Walsh basis. It should be noted that the realization of a permutation requires no special hardware [16]. Another advantage of the Haar spectrum in this application is the smallest number of required arithmetic operations, as there are many zero entries in the Haar transform matrix, and the number of nonvanishing Haar coefficients is reduced. The basic module from Fig. 1 can be modified to cater for the synthesis of sequential circuits [17].

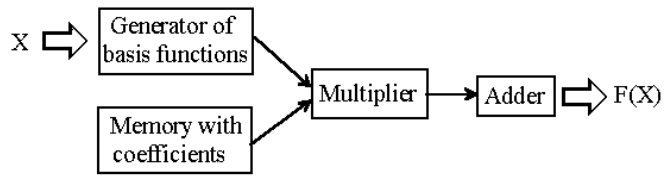

Fig. 1. Basic structure of a spectral multifunctional module.

The recent article [6] caused even more interest in the Walsh spectrum by showing an efficient method of calculating this spectrum directly from its recursive definition rather than from the properties of the transform itself, as the earlier method [23], [24]. In the method [6], both the original data and resulting spectrum are stored in the form of Decision Diagrams. This method was later extended to the calculation of the Chrestenson Transform, which is a generalization of Walsh functions based on complex numbers, that is frequently used for the representation of multiple-valued functions [16], [20]. Unfortunately, both mentioned methods, though general, can be applied only to such transformation

- *B.J. Falkowski is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Ave., Singapore 639798. E-mail: efalkowski@ntu.edu.sg.*
- *C.-H. Chang was previously with Nanyang Technological University. Currently, he is with Nanyang Polytechnic, French Singapore Institute, 12 Science Centre Road, Singapore 609080.*

Manuscript received 21 Mar. 1994.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 104962.

matrices that can be represented by a single recursive equation based on a standard Kronecker product structure. For example, they can be used for the Reed-Muller transform [24]. These methods are not suitable for the cases when the transformation matrix is not recursive, which was first noticed by [31]. Also, these methods do not apply to the case of Haar transform, as such a transform cannot be represented recursively by a standard Kronecker product. Instead, the recursive definitions of the Haar transform are based either on new matrix operators and standard Kronecker products applied to some block submatrices from the transform matrix [33], or can be represented by a "Generalized" Kronecker product [30]. Since the methods [6], [20] are not applicable to the Haar transform, the new methods for such a case have to be sought for. Such a new method should take into account the unique properties of the transformation matrix in question, as well as be optimized in terms of number of required operations and memory space.

Although the properties of Haar spectra have considerable interest and attraction, the majority of publications to date have employed the Walsh rather than the Haar transform in their considerations. It is mainly due to the fact that, up to now, there has been no efficient method of calculating Haar spectra directly from reduced representations of Boolean functions, such as Decision Diagrams, and vice versa. The present article addresses this important issue of operating only on reduced representations of both Haar spectra and original Boolean functions, and presents the mutual relationship between them. It is the first time that such a relationship between Haar spectra of Boolean functions and second common representation of such functions (so-called Binary Decision Diagrams discussed in more detail in Section 4) has been established. All the presented derivations are valid not only for completely specified Boolean functions, but also for incompletely specified ones (i.e., functions that have don't cares as some of their logical values) as well. It is very important, since, in most practical engineering design problems, the incompletely specified Boolean functions have to be dealt with. Introduced algorithms allow more efficient manipulation of different representations of Boolean functions during the synthesis process, since both spectral and OBDD representations of such functions are available to the designer, and either of them can be used interchangeably, dependent on the requirements of the design process.

2 Basic Definitions and Properties

An n-variable Boolean function $F(x_1, x_2, \ldots, x_n)$ is a mapping F: $\{0, 1\}^n \rightarrow \{0, 1, -\}^k$ where the symbol "–" means a nonspecified value (a don't care) and k is the number of outputs. A Boolean function is *completely specified* if all its outputs contain only the set {0, 1}, and *incompletely specified* if any of its output is a nonspecified one.

A *literal*, ($\dot{x}_i$) is a variable of a Boolean function in either affirmation (x_i) or negation ($\overline{x}_i$). A *minterm* or *normal term* of an n-variable Boolean function is an AND term of exactly n *different* literals. A *false (OFF) minterm* is a minterm for which the value of the function is zero, *a true (ON) minterm* is a minterm for which the value of the function is one, and *a don't care (DC) minterm* is a minterm for which the value of the function is either zero or one. An n-variable minterm can be represented by an n-bit integer, the *minterm number*. In the minterm number, a variable in affirmation is replaced by one and in negation by zero. Two minterms are said to be *adjacent* when the Hamming distance between their minterm numbers is equal to one.

An n-bit string is a vertex of an object called a *0-cube*. An n-variable Boolean function is represented as an n-dimensional space (n-hypercube), in which each vertex represents a minterm. A collection of 2^i, $i \in \{0, 1, ..., n\}$ adjacent minterms is called an *i-cube*. A cube can be represented by an n-string of 0, 1, and –, where 0 corresponds to the complemented value of the variable, 1 to the affirmative value, and – to the missing variable in the cube. The *ON, OFF,* and *DC cubes* are cubes corresponding to the product term of ON, OFF, and DC minterms, respectively. The sets of ON, OFF, and DC cubes are called *ON, OFF,* and *DC arrays,* respectively. Two cubes are *disjoint* if they do not have any minterm in common. Otherwise, when they share some minterms, they are *nondisjoint*.

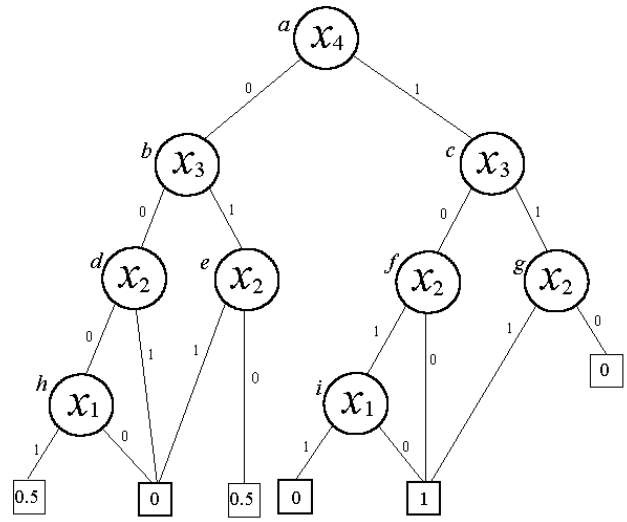

Fig. 2. OBDD of an incompletely specified function, ON(F) = $\{x_4x_2\overline{x}_1, x_4x_3x_2, x_4\overline{x}_3\overline{x}_2\}$, DC(F) = $\{\overline{x}_4x_3\overline{x}_2, \overline{x}_4\overline{x}_2x_1\}$.

The *Shannon's decomposition* of a Boolean function around the variable x_i is [28]:

$$F(X) = x_i F_{x_i} + \overline{x}_i F_{\overline{x}_i} \tag{1}$$

where F_{x_i} is a cofactor of $F(X)$, with respect to x_i, and $F_{\overline{x}_i}$ is a cofactor of $F(X)$, with respect to $\overline{x}_i$.

A Binary Decision Diagram (BDD) [2], [3], [5], [10], [18], [19], [21], [23], [24] is a Rooted Directed Acyclic Graph representation with *Node* (or *Vertex*) *Set V* and *Edge Set E*. The Node Set consists of two types of nodes: the *nonterminal* and *terminal nodes*. A nonterminal node $v \in V$ has as attributes an *index*, denoted by *index*(v), to identify an input variable of a function, and two children (or successors), *low*(v) and *high*(v) $\in V$. A terminal node $u \in V$ has no child, and it has a value, denoted by *value*(u). *value*(u) = 0, 1, or 0.5 for the functional value of logical zero, one, or don't care, respectively. The Edge Set consists of two types of edges. A *0-edge* is a link from a node v to its low child *low*(v), and a *1-edge* is one that connects v to *high*(v). A *root* is the topmost or the first nonterminal node in the BDD. A path is a set of nodes and edges traversed from the root to a terminal node. An *Ordered Binary Decision Diagram* (*OBDD*) is a BDD where the input variables in all paths appear in a fixed order, and each variable in a path appears, at most, once. In an OBDD, an ordering vector for the input variables is maintained, such that *index*(*low*(v)) < *index*(v) and *index*(*high*(v)) < *index*(v) for all nonterminal nodes $v \in V$.

Fig. 2 shows an OBDD for a four-variable incompletely specified Boolean function consisting of an *ON* array, $ON(F) = \{x_4x_2\overline{x}_1, x_4x_3x_2, x_4\overline{x}_3\overline{x}_2\}$, and a *DC* array, $DC(F) = \{\overline{x}_4x_3\overline{x}_2, \overline{x}_4\overline{x}_2x_1\}$. For the purpose of an illustration, each nonterminal node of an OBDD is labeled with a unique alphabet. An edge-connecting node a to node b is denoted by the symbol η_{ab}. η_{ab} is also called the *output edge* of node a or the *input edge* of node b.

PROPERTY 1. *A path with k nodes represents an* $(n - k)$*-cube, where* $k = 1, 2, ..., n$*, since an absent node corresponds to a redundant variable in a term or "–" in a cube notation. For each node v in a path*

η, if its 1-edge is also contained in η, then the variable x_i is present in the cube where i = index(v). Otherwise, the variable $\bar{x}_i$ is present. The logical value of the cube follows the functional value ε of the terminal node in the path. The cubes obtained from any two paths of an OBDD are disjoint.

The orthogonal discrete Haar functions can be formulated as [12], [15]:

$$H_{dc}(x) = 1 \quad for \quad 0 \le x < 1$$

$$H_l^{(k)}(x) = \begin{cases} \sqrt{2^l} & for \ \frac{k}{2^l} \le x < \frac{2k+1}{2^{l+1}} \\ -\sqrt{2^l} & for \ \frac{2k+1}{2^{l+1}} \le x < \frac{k+1}{2^l} \\ 0 & 0 \le x < \frac{k}{2^l} \ or \ \frac{k+1}{2^l} \le x < 1 \end{cases}$$

where x is a continuous interval [0, 1); $l = 0, 1, 2, ..., n-1$ is a degree of Haar function describing the number of zero crossings; and $k = 0, 1, ..., 2^l - 1$ is an order of Haar function describing the position of the subset l within a function. Since $H_{dc}(x)$ is a constant function, it is also called a *direct current* function.

The discrete Haar matrix $[T_n]$ is a $2^n \times 2^n$ orthogonal matrix, formed by a discrete sampling of the set of Haar functions at $\frac{1}{2^n}$ division in the interval [0, 1).The first two rows of $[T_n]$ are global basis functions $H_{dc}(x)$ and $H_0^{(0)}(x)$, respectively. All subsequent rows are constituted by local basis functions $H_l^{(k)}(x)$ in an ascending order of l and k.

In digital logic design, an *unnormalized discrete* Haar transform is used instead [8], [9], [14], [16], [34]. The entries in the unnormalized discrete Haar matrix contain only the values of 1, –1, and 0 that are obtained by taking the signs of all the nonzero entries in the discrete Haar matrix $[T_n]$. For simplicity, the same symbols $H_l^{(k)}$ and $[T_n]$ are used to denote an unnormalized discrete Haar function and the matrix, respectively. From now on, we will refer to them as Haar functions and the Haar matrix, without the words *unnormalized* and *discrete*.

For an n-variable Boolean function $F(x_1, x_2, \ldots, x_n)$ the Haar spectrum is given by [14], [16], [17], [34]: $R = [T_n][F]$, where R is the Haar spectrum (a column vector of dimension $2^n \times 1$) and $[F]$ is a $2^n \times 1$ column vector of the logical values of the function $F(X)$ (a truth table or minterm vector). Two types of coding are used for the minterm vector $[F]$ of a Boolean function before its spectrum is computed [7], [8], [9], [14], [16], [34]. The truth vector for R coding is coded by its original values: 0 for false minterms, 1 for true minterms, and 0.5 for don't care (DC) minterms. In the S coding, false minterms are represented by 1, true minterms by –1, and DC minterms by 0. Since there exists a linear relationship between the Haar spectra for both R and S codings for completely and incompletely specified functions [8], this article will use only R coding.

Each spectral coefficient of R spectrum can be derived by multiplying the coded minterm vector of the Boolean function $[F]$ by its corresponding Haar function. Besides the first two spectral coefficients r_{dc} (the so-called *dc coefficient* corresponding to the *dc function*) and $r_0^{(0)}$,which are globally sensitive to $F(X)$, the remaining $2^n - 2$ Haar spectral coefficients are only locally sensitive to the cofactors resulted from the repeated applications of Shannon's decomposition of $F(X)$, with respect to some variable x_i, $i = 1, 2, ..., n$. Similar to Haar functions, $H_l^{(k)}$ spectral coefficients $r_l^{(k)}$ are characterized by their degrees l and orders k.

PROPERTY 2. *For a Haar spectrum of an n-variable Boolean function F, there are 2^{n-i} spectral coefficients of degree $n - i$.* Each measures a correlation of a different set of 2^i *neighboring minterms, where $i = 1, 2, ..., n$. The value of the dc coefficient r_{dc}, is equal to the number of minterms of the Boolean function, and the coefficient $r_0^{(0)}$ describes the difference between the number of minterms of the cofactors decomposed around the variable $\bar{x}_n$ and x_n, respectively.*

DEFINITION 1. *A* standard trivial function, *u_I, $I = 2^l + k$ and $I \in \{0, 1, ..., 2^n - 1\}$, associated with each Haar function $H_l^{(k)}$ describes a Boolean space of 2^{n-l} neighboring minterms (an $(n-l)$-cube) on a Karnaugh map that has an influence on the value of a spectral coefficient $r_l^{(k)}$, where $l = 0, 1, 2, ..., n-1$ and $k = 0, 1, ..., 2^l - 1$. For each index I of u_I, there exists a unique value of l and k. Formally, u_I can be expressed as a product term*:

$$u_0 = u_1 = 1 \text{ and } u_I = \prod_{i=1}^{l} x_{n-l+i}^{k_i} \ \forall l, k \in Z;$$
$$1 \le l \le n-1 \text{ and } 0 \le k \le 2^l - 1 \qquad (2)$$

where Z is the set of integers.

DEFINITION 2. *A literal $\dot{x}_{n-l}$, $l = 0, 1, ..., n-1$ is called an* extended literal *of the standard trivial function u_I ($I = 1, 2, ..., 2^n - 1$). An extended literal $\dot{x}_{n-l}$ can be either in affirmation or negation and will be called an* extended variable *x_{n-l} or $\bar{x}_{n-l}$ accordingly.*

PROPERTY 3. *An extended literal $\dot{x}_{n-l}$ divides the corresponding standard trivial function, u_I ($I = 1, 2, ..., 2^n - 1$) into two symmetrical halves, equivalent to the cofactors of the Shannon's decomposition of the standard trivial function with respect to x_{n-l} and $\bar{x}_{n-l}$ accordingly. Consequently, $H_{dc} = u_0 = 1$, $H_l^{(k)} = u_1(\bar{x}_{n-l} - x_{n-l})$ $\forall l$ and k, where $I = 2^l + k$.*

PROPERTY 4. *The degree l of the Haar function indicates the number of literals present in a standard trivial function u_I ($I = 0, 1, ..., 2^n - 1$). All 2^l Haar functions of degree l have the same extended literal $\dot{x}_{n-l}$.*

PROPERTY 5. *The order k of a Haar function $H_l^{(k)}$ indicates the polarities of the literals present in the standard trivial function u_I. The order k can be expressed as a binary l-tuple by writing a 1 or 0 for each variable in u_I, according to whether this literal appears in affirmation or negation, with the most significant bit corresponds to the literal $\dot{x}_n$ and the least significant bit corresponds to the literal $\dot{x}_{n-l+1}$.*

EXAMPLE 1. For a four variable Boolean function, the Haar coefficient $r_3^{(1)}$ has the standard trivial function u_9. Since $l = 3$, k can be expressed as a binary 3-tuple 001. From (2), $u_9 = \bar{x}_4\bar{x}_3x_2$, and the extended literal is $\dot{x}_{4-3} = \dot{x}_1$.

3 ALGORITHM FOR COMPUTATION OF HAAR SPECTRUM FROM OBDD

The OBDD is canonical if there exists no nonterminal node v with *index*(*low*(v)) = *index*(*high*(v)). In this paper, we consider only canonical or reduced OBDD.

DEFINITION 3. *The total number of truth, false, or don't care minterms covered by the term $u_I\dot{x}_{n-l}$ is denoted by $\sum M_\varepsilon(u_I, \dot{x}_{n-l})$ where $\varepsilon \in \{0, 1, 0.5\}$, u_I is the standard trivial function u_I and $\dot{x}_{n-l}$ is its corresponding extended literal.*

$\sum M_\varepsilon(u_I, \dot{x}_{n-l})$ can be evaluated by a *matching* process. Each path is either selected or rejected based on the outcome of the comparison of every output edge with the affirmative (logical 1) or

negative (logical 0) value of the associated literal in u_I. Starting from the root, a preorder traversal is performed. If the top variable x_i ($i = index(v)$) of a nonterminal node v is present in u_I, only one of its two children will be traversed depending on the polarity of the corresponding literal in u_I. If the literal is complemented, $low(v)$ will be visited, otherwise, $high(v)$ will be visited. If the top variable of v is the extended literal (i.e., $index(v) = n - l$) or is absent in u_I, both children of v will be traversed. However, an exception occurs in the above matching process. Let v be the present node and $u \in \{low(v), high(v)\}$ be the next node to be visited according to the above rule. Then, u will not be visited if $index(v) > n - l$ and $index(u) < n - l$. In other words, any path that does not contain the extended literal will be pruned early, before the terminal node is reached. During the recursive preorder traversal, whenever a terminal node is encountered, a path with terminal value $\varepsilon \in \{0, 1, 0.5\}$ is selected. The selected path represents a disjoint $(n-p)$-cube if it consists of p nonterminal nodes. If j out of all l literals in u_I are missing in the path, $M_\varepsilon(u_I, \dot{x}_{n-1}) = 2^{n-p-j}$. $\sum M_\varepsilon(u_I, \overline{x}_{n-l})$ (or $\sum M_\varepsilon(u_I, x_{n-l})$) is obtained by accumulating the minterm contributions by each selected path that terminates in $F = \varepsilon$ and contains the extended variable $\overline{x}_{n-l}$ (or x_{n-l}).

In the algorithm computing the Haar spectrum, either the paths that terminate into $F = 1$ and 0.5 or $F = 0$ and 0.5 are to be considered, but not both. The set of paths to be selected depends on the type of the Boolean function, and the condition ruling the selection is given later, as a remark following Algorithm 1.

Algorithm 1: Calculation of Haar spectrum for completely and incompletely specified Boolean functions.

1) Computation of r_{dc} .
   The first spectral coefficient, r_{dc} is given by :
   $$r_{dc} = \sum M_1(F,1) + \tfrac{1}{2}\sum M_{\frac{1}{2}}(F,1) \tag{3}$$
   where $\sum M_1(F,1)$ is the sum of the number of true minterms of the function $F(X)$ and $\sum M_{\frac{1}{2}}(F,1)$ is the sum of the number of DC minterms of the function.
2) Computation of all but r_{dc} spectral coefficients.
   All other spectral coefficients, $r_l^{(k)}$, $l \geq 0$ can be computed by the following procedure:
   (i) Carry out the matching process described previously to select the desired paths.
   (ii) For all selected paths that terminate in 1, add the number of minterms contributed by paths containing the top variables $\overline{x}_{n-l}$ to obtain $\sum M_1(u_I, \overline{x}_{n-l})$ and those containing x_{n-l} to obtain $\sum M_1(u_I, x_{n-l})$, respectively.
   (iii) For all selected paths that terminate in 0.5, add the number of minterms contributed by paths containing the top variable $\overline{x}_{n-l}$ to obtain $\sum M_{\frac{1}{2}}(u_I, \overline{x}_{n-l})$ and those containing x_{n-l} to obtain $\sum M_{\frac{1}{2}}(u_I, x_{n-l})$.
   (iv) The spectral coefficient, $r_l^{(k)}$, is given by:
   $$r_l^{(k)} = \sum M_1(u_I, \overline{x}_{n-l}) - \sum M_1(u_I, x_{n-l}) + \tfrac{1}{2}\left[\sum M_{\frac{1}{2}}(u_I, \overline{x}_{n-l}) - \sum M_{\frac{1}{2}}(u_I, x_{n-l})\right]. \tag{4}$$

It is obvious that, for a completely specified Boolean function, there is no DC minterms and (3), (4) do not have terms with $\sum M_{\frac{1}{2}}$.

For the OBDD that has more terminal values 1 than 0, the spectral coefficients are calculated more efficiently by considering the paths that terminate in $F = 0$ instead of $F = 1$. For such cases, $\sum M_1$ in (3) and (4) are replaced by $\sum M_0$. The calculated spectrum is the spectrum of the complemented function $\overline{F(X)}$, denoted by $\overline{R}$. The actual R spectrum of $F(X)$ can be calculated from $\overline{R}$ by the following set of equations [9]:

$$r_{dc} = 2^n - \overline{r}_{dc} \text{ and } r_l^{(k)} = -\overline{r}_l^{(k)} \ \forall l,$$
$$k \in Z;\ 0 \leq l \leq n-1 \text{ and } 0 \leq k \leq 2^l - 1. \tag{5}$$

EXAMPLE 2. Consider the incompletely specified Boolean function represented by the OBDD in Fig. 2. A sample calculation of the first three spectral coefficients is given as follows.

For all paths that terminate in 1 and 0.5, we have: $M_{\frac{1}{2}}(F,1) = 1$ for $\eta_{abdh}(0.5)$, $M_{\frac{1}{2}}(F,1) = 2$ for $\eta_{abe}(0.5)$, $M_1(F,1) = 1$ for $\eta_{acfi}(1)$, and $M_1(F,1) = 2$ for $\eta_{acf}(1)$ and $\eta_{acg}(1)$. From (4), $r_{dc} = (1 + 2 \times 2) + 0.5(1 + 2) = 6.5$. $M_1(u_1, x_4) = 1$ for $\eta_{acfi}(1)$, $M_1(u_1, x_4) = 2$ for $\eta_{acf}(1)$ and $\eta_{acg}(1)$, $M_{\frac{1}{2}}(u_1, \overline{x}_4) = 1$ for $\eta_{abdh}(0.5)$ and $M_{\frac{1}{2}}(u_1, \overline{x}_4) = 2$ for $\eta_{abe}(0.5)$. Since $\sum M_1(u_1, \overline{x}_4) = \sum M_{\frac{1}{2}}(u_2, x_4) = 0$, from (5), $r_0^{(0)} = 0 - (1 + 2 \times 2) + 0.5[(1 + 2) - 0] = -3.5$. $M_{\frac{1}{2}}(u_2, x_3) = 2$ for $\eta_{abe}(0.5)$ and $M_{\frac{1}{2}}(u_2, \overline{x}_3) = 1$ for $\eta_{abdh}(0.5)$. Since $\sum M_1(u_2, \overline{x}_3) = \sum M_1(u_2, x_3) = 0$, from (5), $r_1^{(0)} = 0 + 0.5(1 - 2) = -0.5$.

All other higher degree spectral coefficients are computed similarly. The other nonvanishing coefficients are calculated to be: $r_1^{(1)} = r_2^{(1)} = r_2^{(2)} = r_3^{(5)} = 1, r_2^{(0)} = 0.5, r_2^{(3)} = -2$, and $r_3^{(0)} = -0.5$.

It should be noted that the Haar spectrum for a different variable ordering as that of the initial OBDD is also possible by Algorithm 1. In this case, the literals in u_I may not appear in adjacent order in the OBDD. Pruning of paths that does not contain the extended literals has to be delayed until the path is selected based on the matching process. The modified algorithm can also be extended to calculate the Haar spectrum from Free Binary Decision Diagram (FBDD) [5], [11], where each variable appears, at most, once in each path, but the variables encountered in any two paths may appear in different order.

4 DECOMPOSITION OF HAAR SPECTRAL COEFFICIENTS

When an n- variable Boolean function $F(X)$ is decomposed with respect to $\overline{x}_n$ or x_n, the *residue Haar spectra* R' of the cofactor $F_{\overline{x}_n}$ and R'' of the cofactor F_{x_n} can be computed by:

$$R' = \left[T_{n-1}\right]F_{\overline{x}_n} \text{ and } R'' = \left[T_{n-1}\right]F_{x_n}. \tag{6}$$

LEMMA 1. *The relationships expressing the 2^{n-1} residue Haar spectral coefficients of the cofactors $F_{\overline{x}_n}$ and F_{x_n} in terms of the Haar spectral coefficients of n-variable Boolean function $F(X)$ can be derived as follows*:

$$r'_{dc} = \tfrac{1}{2}\left(r_{dc} + r_0^{(0)}\right) \text{ and } r_l^{(k)'} = r_{l+1}^{(k)} \ \forall\, l,\ k \in Z;$$
$$0 \leq l \leq n-2 \text{ and } 0 \leq k \leq 2^l - 1 \tag{7}$$
$$r''_{dc} = \tfrac{1}{2}\left(r_{dc} - r_0^{(0)}\right) \text{ and } r_l^{(k)''} = r_{l+1}^{(2^l+k)} \ \forall\, l,\ k \in Z;$$
$$0 \leq l \leq n-2 \text{ and } 0 \leq k \leq 2^l - 1 \tag{8}$$

where the symbols with single and double prime superscripts denote the spectral coefficients of the cofactors $F_{\overline{x}_n}$ and F_{x_n}, respectively.

PROOF. Since the cofactor with respect to $\overline{x}_n$ can be obtained by considering only the first 2^{n-1} minterms of $F(X)$, the local basis Haar functions of $\left[T_n\right]$ in the time interval [0.5, 1) have no direct contribution to the residue spectrum of the cofactor. When the basis functions of $\left[T_n\right]$ in the interval [0, 0.5)

TABLE 1
HAAR SPECTRA FOR *n*-VARIABLE ELEMENTARY FUNCTIONS

S/N	ON(F)	DC(F)	Values of non-vanishing coefficients
1(a)	x_n	$\varnothing$	$r_{dc} = 2^{n-1}, r_D^{(0)} = -2^{n-1}$
1(b)	$\bar{x}_n$	$\varnothing$	$r_{dc} = r_D^{(0)} = 2^{n-1}$
1(c)	$\varnothing$	x_n	$r_{dc} = 2^{n-2}, r_D^{(0)} = -2^{n-2}$
1(d)	$\varnothing$	$\bar{x}_n$	$r_{dc} = r_D^{(0)} = 2^{n-2}$
1(e)	x_n	$\bar{x}_n$	$r_{dc} = 3\times 2^{n-2}, r_D^{(0)} = -2^{n-2}$
1(f)	$\bar{x}_n$	x_n	$r_{dc} = 3\times 2^{n-2}, r_D^{(0)} = 2^{n-2}$
2(a)	OR	$\varnothing$	$r_{dc} = 2^n - 1, r_l^{(0)} = -1 \ \forall\, l$
2(b)	NOR	$\varnothing$	$r_{dc} = r_l^{(0)} = 1 \ \forall\, l$
2(c)	$\varnothing$	OR	$r_{dc} = 2^{n-1} - 0.5, r_l^{(0)} = -0.5 \ \forall\, l$
2(d)	$\varnothing$	NOR	$r_{dc} = r_l^{(0)} = 0.5 \ \forall\, l$
2(e)	OR	NOR	$r_{dc} = 2^n - 0.5, r_l^{(0)} = -0.5 \ \forall\, l$
2(f)	NOR	OR	$r_{dc} = 2^{n-1} + 0.5, r_l^{(0)} = 0.5 \ \forall\, l$
3(a)	AND	$\varnothing$	$r_{dc} = 1, r_l^{(2^l-1)} = -1 \quad \forall\, l$
3(b)	NAND	$\varnothing$	$r_{dc} = 2^n - 1, r_l^{(2^l-1)} = 1 \quad \forall\, l$
3(c)	$\varnothing$	AND	$r_{dc} = 0.5, r_l^{(2^l-1)} = -0.5 \quad \forall\, l$
3(d)	$\varnothing$	NAND	$r_{dc} = 2^{n-1} - 0.5, r_l^{(2^l-1)} = 0.5 \quad \forall\, l$
3(e)	AND	NAND	$r_{dc} = 2^{n-1} + 0.5, r_l^{(2^l-1)} = -0.5 \quad \forall\, l$
3(f)	NAND	AND	$r_{dc} = 2^n - 0.5, r_l^{(2^l-1)} = 0.5 \quad \forall\, l$
4(a)	XOR	$\varnothing$	$r_{dc} = 2^{n-1}, r_{n-1}^{(k)} = (-1)^{g+1} \quad \forall\, 0 \le k < 2^{n-1}$
4(b)	XNOR	$\varnothing$	$r_{dc} = 2^{n-1}, r_{n-1}^{(k)} = (-1)^{g} \quad \forall\, 0 \le k < 2^{n-1}$
4(c)	$\varnothing$	XOR	$r_{dc} = 2^{n-2}, r_{n-1}^{(k)} = 0.5(-1)^{g+1} \quad \forall\, 0 \le k < 2^{n-1}$
4(d)	$\varnothing$	XNOR	$r_{dc} = 2^{n-2}, r_{n-1}^{(k)} = 0.5(-1)^{g} \quad \forall\, 0 \le k < 2^{n-1}$
4(e)	XOR	XNOR	$r_{dc} = 3\times 2^{n-2}, r_{n-1}^{(k)} = 0.5(-1)^{g+1} \quad \forall\, 0 \le k < 2^{n-1}$
4(f)	XNOR	XOR	$r_{dc} = 3\times 2^{n-2}, r_{n-1}^{(k)} = 0.5(-1)^{g} \quad \forall\, 0 \le k < 2^{n-1}$

are compared with those of $[T_{n-1}]$, the following relationships are obtained: $H'_{dc} = \frac{1}{2}\left(H_{dc} + H_0^{(0)}\right)$ and $H_l^{(k)'} = H_{l+1}^{(k)} \ \forall l,\ k \in Z \mid 0 \le l \le n-2$ and $0 \le k \le 2^l - 1$, where $H_l^{(k)'}$ denotes the Haar functions of $[T_{n-1}]$.

Since $r_{dc} = H_{dc}[F]$ and $r_l^{(k)} = H_l^{(k)}[F]$, all symbols H in the above relations can be substituted by corresponding to them symbols r for all concerned degrees and orders of the Haar functions. Hence, (7) is derived.

Equation (8) can be similarly proven by comparing the basis Haar functions of $[T_n]$ in the time interval [0.5, 1) with that of $[T_{n-1}]$ and performing some arithmetic operations on the basis functions of $[T_n]$. □

Equations (7) and (8) closely parallel the definition of Shannon's decomposition from (1), except that the decision variable x_i must be evaluated in a descending order of the index i ($i = n, n-1, \ldots, 1$). The size of the resulting spectrum R' or R'' is halved through each iteration. After n iterations, only a single spectral coefficient is left. The value of this coefficient is either 0, 0.5 or 1, representing the R coded functional value of the minterm formed by the conjunction of all decision variables along the path.

PROPERTY 6. *When the dc coefficient of a residue spectrum is equal to* 0, *the logical function describing the cofactor is* 0.

The following Properties 7 to 10 and Lemmas 2 and 3 are given for the case when the Boolean function is always dependent on the variable x_n or the decomposition of the function starts with the variable x_n. Such formulation of properties is useful in the following algorithm. However, it should be noticed that the more general case of the functional dependence on a variable x_i or the decomposition with any variable x_i, $1 \le i \le n$ is possible. Similar properties can be given for the decomposition according to any cofactor $F_{\bar{x}_i g(\dot{X})}$ or $F_{x_i g(\dot{X})}$ where $g(\dot{X}) = g(\dot{x}_{i+1}, \dot{x}_{i+2}, \ldots, \dot{x}_n)$ and the corresponding residue spectra R' and R'' of 2^{i-1} spectral coefficients.

PROPERTY 7. *When all but the dc coefficients of a spectrum are zero, the n-variable function is either a tautology or all its space is full of don't cares. Coefficient* r_{dc} *has the maximum value* 2^n *for the tautology and* 2^{n-1} *for the second case.*

PROPERTY 8. *When an n-variable function or a cofactor is dependent only on the most significant variable* x_n, *it can be represented by a BDD with only one nonterminal node v where index*(*v*) = *n and low*(*v*) *and high*(*v*) *are both terminal nodes. The six possible combinations of ON-, OFF-, and DC-terminal values associated with low*(*v*) *and high*(*v*) *are shown in rows 1a to 1f of Table 1.*

PROPERTY 9. *When an n-variable Boolean function is independent of the most significant variable* x_n, *its spectrum is characterized by* $r_0^{(0)} = 0$ *and* $r_{l+1}^{(k)} = r_{l+1}^{(2^l+k)}\ \forall l, k \in Z; 0 \le l \le n-2$ and $0 \le k \le 2^l - 1$.

Property 9 is derived by equating (7) with (8) since the cofactors of such a function decomposed about the variable x_n are identical. From Lemma 1, whenever this condition occurs in the synthesis process, the value of r_{dc} is halved and the degrees of the remaining coefficients are decremented by 1 while their values remain unchanged.

PROPERTY 10. *Similar to Property 9, there are six possible combinations of elementary functions generated from each basic logical operation on n nonredundant variables. The nonvanishing spectral coefficients for these elementary functions are given in rows 2a-2f to 4a-4f of Table 1. In Table 1,* OR = $x_1 \vee x_2 \vee \ldots \vee x_n$, AND = $x_1 \wedge x_2 \wedge \ldots \wedge x_n$, XOR = $x_1 \otimes x_2 \otimes \ldots \otimes x_n$, *and z is the Hamming weight of the order k, i.e., the number of ones in the binary n-tuple of the decimal number k. The BDDs of these elementary functions can be easily derived.*

LEMMA 2. *Let OBDD(F) be an OBDD of an n-variable completely specified function F and OBDD(G) be an OBDD of an incompletely specified function G obtained from OBDD(F) by changing the terminal node with terminal value* 0 *to* 0.5. *If superscripts f and g are used to denote the spectral coefficients of F and G, respectively, then*

$$r_l^{g(k)} = \tfrac{1}{2}\left(r_{dc}^f + 2^n\right) \text{ and } r_l^{g(k)} = \tfrac{1}{2} r_l^{f(k)}\ \forall l, k \in Z;\quad 0 \le l \le n-2 \text{ and } 0 \le k \le 2^l - 1 \tag{9}$$

PROOF. Let t_0 and t_1 be the number of false and truth minterms for F. $t_0 = 2^n - t_1$ and $r_{dc}^f = t_1$. Since the number of don't care minterms in $G = t_0$, $r_{dc}^g = t_1 + 0.5t_0 = 0.5\left(t_1 + 2^n\right) = 0.5\left(r_{dc}^f + 2^n\right)$. From (4), since F is completely specified, $r_l^{f(k)} = \sum M_1\left(u_I, \bar{x}_{n-l}\right) - \sum M_1\left(u_I, x_{n-l}\right)$. In addition, for any spectral coefficient of degree l of G, $\sum M_{\frac{1}{2}}\left(u_I, \bar{x}_{n-l}\right) = 2^l - \sum M_1\left(u_I, \bar{x}_{n-l}\right)$ and $\sum M_{\frac{1}{2}}\left(u_I, x_{n-l}\right) = 2^l - \sum M_1\left(u_I, x_{n-l}\right)$. From (4),

$$r_l^{g(k)} = \sum M_1\left(u_I, \bar{x}_{n-l}\right) - \sum M_1\left(u_I, x_{n-l}\right) + \tfrac{1}{2}\left[2^l - \sum M_1\left(u_I, \bar{x}_{n-l}\right) - 2^l + \sum M_1\left(u_I, x_{n-l}\right)\right] = \tfrac{1}{2}\left[\sum M_1\left(u_I, \bar{x}_{n-l}\right) - \sum M_1\left(u_I, x_{n-l}\right)\right] = \tfrac{1}{2} r_l^{f(k)}.$$ □

LEMMA 3. *If OBDD(G) in Lemma 2 is obtained from OBDD(F) by changing the terminal node with terminal value* 1 *to* 0.5, *then* $R^g = 0.5R^f$.

PROOF. Similar to proof of Lemma 2. □

Similar to the output complement attribute edge [3], the replacement of the terminal node with value 0 by that with value 0.5 and the terminal node with value 1 by that with value 0.5 of a BDD can be regarded as two additional new attribute edges *op0* and *op1* respectively. The introduction of *op0* and *op1* not only maximizes the space efficiency but also increases the hit ratio of the hash-based cache compute table. From Lemmas 2 and 3, it is obvious that halving the summation of all higher coefficients constitutes a simple and effective hash function for both *op0* and *op1*. Given the above preliminaries, the algorithm for constructing OBDD from Haar spectrum is based on the successive applications of (7) and (8). The decision variable is in the descending order of the index i ($i = n, n-1, \ldots, 1$).

Algorithm 2: Synthesis of OBDD from Haar spectrum for completely and incompletely specified Boolean functions

1) Using Properties 9 and 10, verify if the spectrum describes a single variable function or an elementary function. If it is the case, replace the spectrum by the corresponding OBDD and exit.
2) Set the index $i = n$, where n is the number of variables of the function in the first iteration and the number of variables of the considered cofactors subsequently. Assign the variable x_i to the root of OBDD with two output edges.
3) Using (7) and (8), compute two residue Haar spectra, each of 2^{i-1} coefficients for the cofactors $F_{\bar{x}_i}$ and F_{x_i} corresponding to the low and high children. Reduce the index i by 1.
4) When any edge has a *dc* coefficient (either r'_{dc} or r''_{dc}) equal to 0, connect it to the 0-valued terminal node.
5) When any edge has a *dc* coefficient equal to the maximum value 2^i, connect it to the 1-valued terminal node. Skip Step 6 if the function is completely specified.
6) When any edge has a *dc* coefficient equal to 2^{i-1} and all other coefficients are zero, connect it to a 0.5-valued (don't care) terminal node.
7) When any edge has a residue spectrum such that the zero degree coefficient is equal to 0 and $r_{l+1}^{(k)} = r_{l+1}^{(2^l+k)}$ for $\forall l, k \in Z$; $0 \le l \le n-2$ and $0 \le k \le 2^l - 1$, delete the coefficients $r_0^{(0)}$ and $r_{l+1}^{(2^l+k)}$. The value of r_{dc} is halved and the degrees l of the remaining coefficients $r_l^{(k)}$ are decreased by one while their values remain unchanged.
8) When some spectra corresponding to two or more output edges from some node x_i are identical, direct all the output branches to only one copy of such a spectrum.
9) For any nonterminal branches, repeat Steps 1 to 9 until all edges are terminated.

EXAMPLE 3. For the set of Haar coefficients calculated in Example 2, the partial OBDDs resulting from each iteration of the above Algorithm are shown in Figs. 3a-3c. Different numbered steps of Algorithm are indicated by arrows pointing to either terminal values or subtrees in Fig. 3. The final OBDD is identical to that shown in Fig. 2.

5 EXPERIMENTAL RESULTS

The presented algorithm has been implemented in C. Table 2 shows the results of the calculation of complete Haar spectra for selected MCNC benchmark functions obtained from their OBDD representations. The ordering of variables is derived from the circuit topology. To give an impression of size of the considered circuits and resulting Haar spectra, some information on the MCNC benchmark functions is provided in Table 2. The columns "Inputs" and "Outputs" show the number of input and output variables of the benchmark functions respectively, and the column "Number" represents the number of essential nonvanishing Haar spectral coefficients, while the column "Time" is the system execution time in seconds required to calculate the Haar spectrum on a HP Apollo Series 715 workstation. It should be noted that the algorithm can accept a more general FBDD representation.

6 CONCLUSION

The essential relationships between classical (OBDDs) and spectral (Haar spectra) representations of Boolean functions used in the design of VLSI digital circuits have been stated. The fundamental

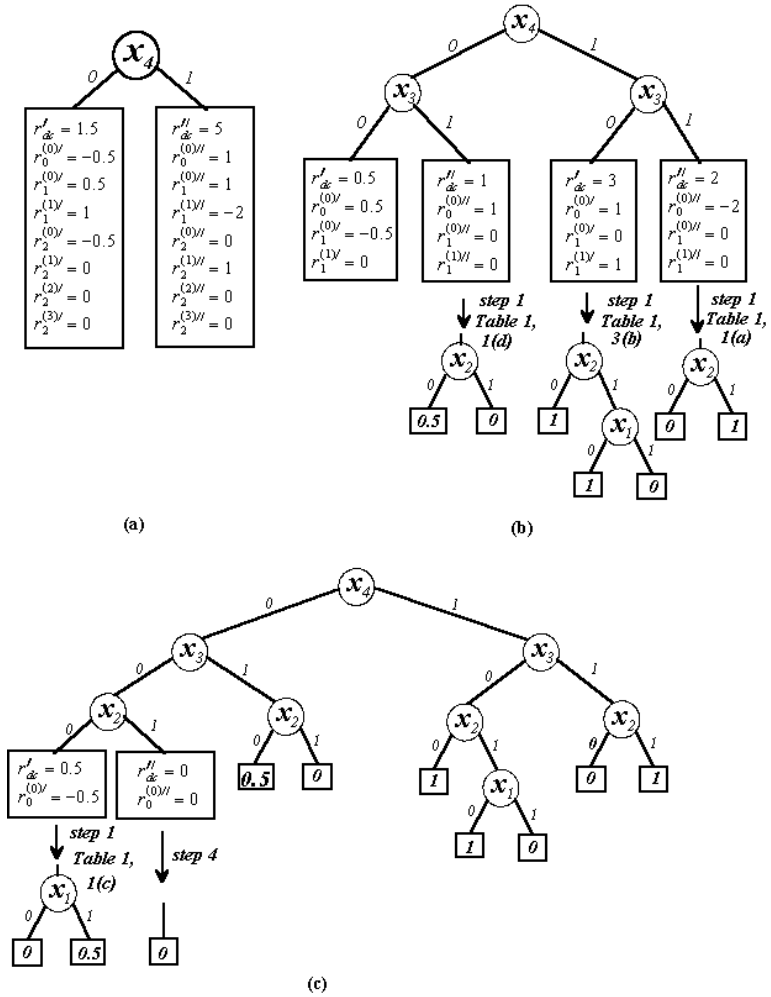

Fig. 3. Partial OBDDs resulting from Algorithm 2.

formulas presented in Sections 3 and 4 are very useful, since either representation can be more convenient in different stages of the VLSI design process. The stated formulas give us the working tool to translate in both directions the spectral and classical knowledge about the underlying Boolean function. The research summarized here will have not only impact on the more efficient applications of both representations of Boolean functions in the design process, but also gives the insight onto the links between computer and communication technologies: two areas that use extensively Haar spectra in many applications. In both technologies, Haar spectra are the most promising approaches dealing with the problems of test generation and response data analysis and compression [26], [27]. A major advantage of the approach presented here to mutual Haar spectrum / OBDD conversion process is its convenience for computer implementation, and, by using reduced representations for both original data and corresponding spectra, its ability to yield solutions to problems of very large dimensions. A similar approach can also be applied to calculate other spectra that do not have recursive transformation from Decision Diagrams, for example, the spectra of different Discrete Wavelet Transforms [32].

REFERENCES

[1] N. Ahmed and K.R. Rao, *Orthogonal Transforms for Digital Signal Processing*. Berlin: Springer-Verlag, 1975.

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

[3] R.E. Bryant, "Graph-Based Algorithm for Boolean Function Manipulation," *IEEE Trans. Computers*, vol. 35, no. 8, pp. 677-691, Aug. 1986.

[4] A.M. Buron, J.A. Michell, and J.M. Solana, "Single Chip Fast Haar Transform at Megahertz Rates," *Theory and Applications of Spectral Techniques*, C. Moraga, ed., pp. 8-17. Univ. Dortmund Press, Oct. 1988.

[5] S. Chakravarty, "A Characterization of Binary Decision Diagrams," *IEEE Trans. Computers*, vol. 42, no. 2, pp. 129-137, Feb. 1993.

[6] E.M. Clarke, K.L. McMillian, X. Zhao, M. Fujita, and J. Yang, "Spectral Transforms for Large Boolean Functions with Applications to Technology Mapping," *Proc. 30th ACM/IEEE Design Automation Conf.*, pp. 54-60, June 1993.

[7] B.J. Falkowski, I. Schaefer, and M.A. Perkowski, "Effective Computer Methods for the Calculation of Rademacher-Walsh Spectrum for Completely and Incompletely Specified Boolean Functions," *IEEE Trans. Computer-Aided Design*, vol. 11, no. 10, pp. 1,207-1,226, Oct. 1992.

[8] B.J. Falkowski and C-H Chang, "A Novel Paired Haar Based Transform: Algorithms and Interpretations in Boolean Domain," *Proc. 36th Midwest Symp. Circuits and Systems*, Detroit, pp. 1,101-1,104, Aug. 1993.

[9] ----, "Efficient Algorithms for the Forward and Inverse Transformations Between Haar Spectrum and Binary Decision Diagram," *Proc. 13th IEEE Int'l Phoenix Conf. Computers and Communications*, pp. 497-503, Phoenix, Apr. 1994.

[10] S.J. Friedman and K.J. Supowit, "Finding the Optimal Variable Ordering for Binary Decision Diagrams," *IEEE Trans. Computers*, vol. 39, no. 5, pp. 710-713, May 1990.

[11] J. Gergov and C. Meinel, "Efficient Boolean Manipulation with OBDDs Can Be Extended to FBDDs," *IEEE Trans. Computers*, vol. 43, no. 10, pp. 1,197-1,209, Oct. 1994.

TABLE 2
EXPERIMENTAL RESULTS

MCNC	Inputs	Outputs	Number	Time (sec.)
alu2	10	6	441	0.03
alu4	14	8	6,185	0.06
9symml	9	1	214	0.05
x2	10	7	97	0.04
parity	16	1	32,768	0.19
mux	21	1	35,490	0.11
f51m	8	8	82	0.03
cmb	16	1	224	0.05
Z4ml	7	4	51	0.02
cu	14	11	59	0.05
pm1	16	13	15,709	0.09
cc	21	20	17,626	0.07
ttt2	24	21	839,126	5.00
pcler8	27	17	163,840	0.51
comp	32	3	262,136	0.69
pcle	19	9	65,792	0.21
sct	19	15	26,912	0.08

[12] A. Haar, "Zur Theorie der othorgonalen Funktionsysteme," *Math. Ann*., vol. 69, pp. 331-371, 1910.

[13] H.F. Harmuth, *Transmission of Information by Orthogonal Functions*. Berlin: Springer-Verlag, 1972.

[14] S.L. Hurst, D.M. Miller, and J.C. Muzio, *Spectral Techniques in Digital Logic*. London: Academic Press, 1985.

[15] S. Kaczmarz and H. Steinhaus, *Theory of Orthogonal Series* (in German). New York: Chelsea, 1951.

[16] M.G. Karpovsky, *Finite Orthogonal Series in the Design of Digital Devices*. New York: John Wiley, 1976.

[17] W. Kulesza, *Systems of Spectral Analysis of Digital Data* (in Polish). Warsaw: WKL, 1984.

[18] C.Y. Lee, "Representation of Switching Circuits by Binary-Decision Diagrams," *Bell System Technical J*., vol. 38, pp. 985-999, July 1959.

[19] M.R. Mercer, R. Kapur and D.E. Ross, "Functional Approaches to Generating Orderings for Efficient Symbolic Representations," *Proc. 29th ACM/IEEE Design Automation Conf*., pp. 624-627, June 1992.

[20] D.M. Miller, "Spectral Transformation of Multiple-Valued Decision Diagrams," *Proc. IEEE 24th Int'l Symp. Multiple-Valued Logic*, pp. 89-96, Boston, May 1994.

[21] B.M.E. Moret, "Decision Trees and Diagrams," *ACM Computing Surveys*, vol. 14, no. 4, pp. 593-622, Dec. 1982.

[22] J.C. Muzio and S.L. Hurst, "The Computation of Complete and Reduced Sets of Orthogonal Spectral Coefficients for Logic Design and Pattern Recognition Purpose," *Int'l J. Computers and Electrical Eng*., vol. 5, pp. 231-249, 1978.

[23] S. Purwar and A.K. Susskind, "Computation of Walsh Spectrum from Binary Decision Diagram and Binary Decision Diagram from Walsh Spectrum," *Int'l J. Computers and Electrical Eng*., vol. 15, no. 2, pp. 59-65, 1989.

[24] S. Purwar, "An Efficient Method of Computing Generalized Reed-Muller Expansion from Binary Decision Diagram," *IEEE Trans. Computers*, vol. 40, no. 11, pp. 1,298-1,301, Nov. 1991.

[25] P.R. Roeser and M.E. Jernigan, "Fast Haar Transform Algorithms," *IEEE Trans. Computers*, vol. 31, no. 2, pp. 175-177, 1982.

[26] G. Ruiz, J.A. Michell, and A. Buron, "Fault Detection and Diagnosis for MOS Circuits from Haar and Walsh Spectrum Analysis: On the Fault Coverage of Haar Reduced Analysis," *Theory and Applications of Spectral Techniques*, C. Moraga, ed., pp. 97-106. Univ. Dortmund Press, Oct. 1988.

[27] G. Ruiz, J.A. Michell, and A. Buron, "Switch-Level Fault Detection and Diagnosis Environment for MOS Digital Circuits Using Spectral Techniques," *IEE Proc.*, Part E, vol. 139, no. 4, pp. 293-307, July 1992.

[28] C.E. Shannon, "A Symbolic Analysis of Relay and Switching Circuits," *Collected Papers of Claude Elwood Shannon*, N.J.A. Sloane and A.D. Wyner, eds., pp. 471-495. IEEE CS Press, 1992.

[29] J.E. Shore, "On the Applications of Haar Functions," *IEEE Trans. Comm*., vol. 21, pp. 206-216, 1973.

[30] M.R. Stojic, M.S. Stankovic, and R.S. Stankovic, *Discrete Transforms and Their Applications* (in Serbian). Beograd: Nauka, 1993.

[31] M.A. Thornton and V.S.S. Nair, "Efficient Calculation of Spectral Coefficients and Their Applications," *IEEE Trans. Computer-Aided Design Integrated Circuits and Systems*, vol. 4, no. 11, pp. 1,328-1,341, Nov. 1995.

[32] G.G. Walter, *Wavelets and Other Orthogonal Systems with Applications*. Boca Raton, Fla.: CRC Press, 1994.

[33] L.P. Yaroslavsky, *Digital Picture Processing*. Berlin: Springer-Verlag, 1985.

[34] L.A. Zalmanzon, *Fourier, Walsh and Haar Transforms and Their Application in Control, Communication and Other Fields* (in Russian). Moscow: Nauka, 1989.