Generalized Haar Spectral Representations and Their Applications

Bogdan J. Falkowski
Nanyang Technological University,
School of Electrical and Electronic Engineering,
Blk S1, Nanyang Avenue, Singapore 639798.
Email : efalkowski@ntu.edu.sg
Phone : (65)790-4521
Fax : (65)791-2687

Abstract

Haar transform is known to have the smallest computational requirement and has been used mainly for pattern recognition and image processing. Although the properties of Haar spectra of Boolean functions have considerable interest and attraction, the majority of publications to date have employed the Walsh rather than Haar transform in their considerations. It is mainly due to the fact that up to recently there was no efficient method of calculating Haar spectra directly from reduced representations of Boolean functions such as decision diagrams and cubes. Recently, efficient methods based on Decision Diagrams and cubical representation for the computation of Haar spectra have been developed. Two methods based on decision diagrams and a new data structure called the "Haar Spectral Diagram" is discussed. The method to calculate Haar spectra from disjoint cubes of Boolean functions is also presented. A concept of paired Haar transform for representation and efficient optimization of systems of incompletely specified Boolean functions will be discussed. Finally another form of Haar transform, so called "Sign Haar Transform" is discussed and basic properties of Boolean functions in its spectral domain are shown. Various applications of Haar transform in logic design are also mentioned.

1. Introduction

There are at least two transforms which are based on square-wave like functions that are well suitable for Boolean functions: Walsh and Haar transforms. 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, 2, 20, 23-27, 39, 42, 43]. Each but two basis functions in Haar transform consists of a square wave pulse located on an otherwise zero amplitude interval. 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, 2, 32]. Due to its low computing requirements, Haar transform has been used mainly for pattern recognition and image processing [2, 3, 9, 27, 42, 43]. Such a transform is also well suited in communication technology for data coding, multiplexing and digital filtering [1, 26, 36]. The advantages of computational and memory requirements of the Haar transform make it of big interest to VLSI designers as well [24, 34, 35].

Local property of Haar transform makes it of interest in those applications in computer-aided design systems where there are Boolean functions of many variables that have most of its values grouped locally. Such weakly specified and local functions frequently occur in logic design and machine learning [8, 28, 29] and can be extremely well described by

few spectral coefficients from Haar transform while the application of Walsh, global transform would be quite cumbersome in such cases and the locally grouped minterms would be spread throughout the Walsh spectrum. In most engineering design problems, incompletely specified functions have to be dealt with. To better deal with the mentioned cases, we have introduced a novel concept of *Paired Haar transform* [11, 13, 15]. In Paired Haar Transform, all the information about true and don't care minterms is kept separately, by what it is available in different stages of CAD process. In efficient synthesis of incompletely specified Boolean functions there is a need for filling don't care minterms of the original function by '0's and '1's in such a way that the resulting completely specified Boolean function will be easily implemented by available basic gate structures and Programmable Logic Devices (PLDs). Different complexity criteria are used in order to make such an allocation of don't minterms optimal in the sense that the final SOP consists of the least number of product terms or literals. In order to fulfill the above requirement by spectral approach, instead of operating on a single spectrum from the *R*-coded vector, a *Paired Haar transform* has been introduced. The definitions and properties of *Paired Haar transforms* are discussed here.

Finding the minimal realizations for logic functions is usually associated with the problems of optimizing their reduced representations. An array of cubes provides a straight forward implementation of a disjunctive sum-of-products (SOP) expression for a given function in two-levels [10]. Generation of a minimal SOP expression for an incompletely specified Boolean function is equivalent to the well known set covering problem which is NP-hard. To avoid duplicating effects due to common minterms among cubes, an array of disjoint cubes corresponding to the disjoint sum-of-products (DSOP) representation has often been used as an initial representation or preprocessing tool in many logic synthesizers [19, 20, 28]. For large circuits, Free Binary Decision Diagram (FBDD) [6] is a more succinct representation than the cubical representations of SOP and DSOP. A special subset of FBDD is the Ordered Binary Decision Diagram (OBDD) [12, 14, 22, 38, 40] which is a canonical representation of Boolean function with a given ordering of variables. All these four reduced Boolean domain representations have widespread applications in logic design and can be directly mapped into two-level realizations or used as initial formulations for further refinement and synthesis in various forms of multilevel digital circuits. The way of calculating Haar spectra directly from various decision diagrams are discussed.

In the next section, a new data structure called the “Haar Spectral Diagram” is also discussed. The way of calculating *Paired Haar* transform directly from disjoint cubes is also presented. Section 6 discusses non-linear quantized transforms. First, a non-linear transform, called the "Sign Haar Transform" is introduced. The transform is unique and converts binary/ternary vectors into ternary spectral domain. Walsh functions served as the underlying basis functions of Sign Walsh Transform with restricted coefficient values. In this section, unnormalized Haar functions are used as the basis of new quantized transform called "Sign Haar Transform". Similarly to the case when the sign function is applied to Walsh functions, two versions of Sign Haar Transform are possible. First, when the signs of normalized Haar spectral coefficients are used, second when the quantized functions at each stage of the flow graph of fast, normalized Haar Transforms are applied. The properties of both spectra obtained according to the above definitions are similar to those obtained by the Sign Walsh Transform. The final section lists different applications of Haar transform in logic design.

2. Unnormalized Haar Transform

The orthogonal discrete Haar functions introduced in [23] can be formulated according to [26, 39, 42] as :

$$H_{dc}(x) = 1 \qquad \text{for} \qquad 0 \le x < 1,$$

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

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 crossing; k = 0, 1, ..., 2^l–1 is an order of Haar function describing the position of the subset l within a function.

The discrete Haar matrix T_N [1, 2, 3, 5, 24, 32, 36, 43] is an N x N ($N = 2^n$) 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. The discrete Haar functions represented by the rows of a ($2^n \times 2^n$) matrix may be ordered in the sequential and natural ordering.

$$T_{8S} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

Figure 1: Sequency ordered Haar functions for n = 3.

$$T_{8N} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

Figure 2: Naturally ordered Haar functions for n = 3.

Definition 2: Discrete Haar functions of order N represented by the ($2^n \times 2^n$) matrix $T_s(n)$, in the sequential ordering are given by the following recurrence relation [39]:

$$T_s(n) = \begin{bmatrix} T_s(n-1) \otimes [1 \quad 1] \\ 2^{\frac{(n-2)}{2}} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix} \otimes I_{2^{k-2}} \otimes \begin{bmatrix} 1 & -1 \end{bmatrix} \end{bmatrix} \quad \text{with } T_s(1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad (2)$$

In digital logic design, an *unnormalized discrete* Haar transform is used instead [11-15, 24-26, 39, 40, 43]. 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 non-zero 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, they will be referred to as Haar function and Haar matrix without the words *unnormalized* and *discrete*.

Definition 3 : The Haar transform matrix T_N of order $N = 2^n$ can be defined recursively as [1, 2, 36, 39, 42] :

$$T_N = \begin{bmatrix} T_{\frac{N}{2}} \otimes \begin{bmatrix} 1 & 1 \end{bmatrix} \\ \mathbf{I}_{\frac{N}{2}} \otimes \begin{bmatrix} 1 & -1 \end{bmatrix} \end{bmatrix} \text{and } T_1 = 1, \qquad (3)$$

where $\mathbf{I}_{\frac{N}{2}}$ is an identity matrix of order $N/2$.

Besides the first two spectral coefficients r_{dc} 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, $i = 1, 2, ..., n$. Similarly to the Haar functions $H_l^{(k)}$, spectral coefficients $r_l^{(k)}$ are characterized by their degrees l and orders k.

Property 1 : 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 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 literals $\bar{x}_n$ and x_n, respectively.

Definition 4 : 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 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 :

$$u_0 = u_1 = 1 \text{ and } u_I = \prod_{i=1}^{l} x_{n-l+i}^{k_i} \ \forall\ l, k \in \mathbf{Z}\ ;\ 1 \le l \le n-1 \text{ and } 0 \le k \le 2^l - 1, \qquad (4)$$

where $\mathbf{Z}$ is the set of integers.

Definition 5 : A literal $\dot{x}_{n-l}$, $l = 0, 1, ..., n-1$ is called an *extended literal* of the standard trivial function u_I ($I = 1, 2, ..., 2^n-1$).

Property 2 : An extended literal $\dot{x}_{n-l}$ divides the corresponding standard trivial function, u_I ($I = 1, 2,..., 2^n-1$) into a *positive* and a *negative standard trivial function* (PSTF and NSTF), 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, $H_{dc} = u_0 = 1$, $H_l^{(k)} = u_I\,(\bar{x}_{n-l} - x_{n-l})\ \forall\ l$ and k where $I = 2^l + k$.

Property 3 : The *degree l* of a Haar function indicates the number of literals present in its standard trivial function u_I ($I = 1, 2, 3,.., 2^n-1$). All 2^l Haar functions of degree l have the same extended literal $\dot{x}_{n-l}$.

Property 4 : The maximum and minimum values of any Haar coefficient of degree l are equal to 2^{n-l-1} and -2^{n-l-1} respectively.

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 (4), $u_9 = \bar{x}_4\bar{x}_3x_2$ and the extended literal is $\dot{x}_{4-3} = \dot{x}_1$.

3. Definition and Properties of Paired Haar Spectrum

Different non-polynomial spectral expansions for completely specified Boolean functions have been known [1, 25]. Based on the recursive definition of unnormalized Haar transform in (3), a non-polynomial Haar expansion of an n-variable Boolean function F can be derived [13].

Theorem 1 :

$$F(X) = \frac{1}{2^n}\left\{ r_{dc} + (-1)^{x_n} r_0^{(0)} + \sum_{l=1}^{n-1} 2^l (-1)^{x_{n-l}} \sum_{k=0}^{2^l-1} r_l^{(k)} \prod_{i=n-l+1}^{n} x_i^{k_{i-n+l}} \right\}, \qquad (5)$$

where $k_i \in \{0,1\}$ is the i-th bit in the binary l-tuple of the order k ; $x_i^j = x_i$ if $j = 1$ and $x_i^j = \bar{x}_i$ if $j = 0$.

Proof : The inverse unnormalized Haar transform T_N^{-1} of order of $N = 2^n$ can be defined in a similar recursive form as in (3) :

$$T_N^{-1} = \frac{1}{2^n} G_N^{\mathrm{T}}, \qquad (6)$$

where $G_N = \begin{bmatrix} G_{\frac{N}{2}} \otimes \begin{bmatrix} 1 & 1 \end{bmatrix} \\ \frac{N}{2}\mathbf{I}_{\frac{N}{2}} \otimes \begin{bmatrix} 1 & -1 \end{bmatrix} \end{bmatrix}$ and $G_{1=1.}$

Comparing (3) and (6), G_N is generated from T_N by incorporating a scaling factor of $N/2$. For $n \geq 1$, since each iteration of (6) generates an additional degree of inverse Haar functions, the scaling factor of 2^l can be applied to every forward Haar functions of the same degree l to obtain the corresponding row in G_N.

Hence, $T^{-1}(0) = \frac{1}{2^n} T^{\mathrm{T}}(0)$ and $T^{-1}(I) = \frac{1}{2^n} \times 2^l \times T^{\mathrm{T}}(I)$ where $I = 1, 2, ..., 2^n-1$, the superscript T denotes matrix transpose and $T(I)$ is the row vector corresponding to row I of the matrix T and $I = 2^l + k$.

$$F(X) = T^{-1} \times R = [T^{-1}(0)]^{\mathrm{T}} r_{dc} + [T^{-1}(1)]^{\mathrm{T}} r_0^{(0)} + [T^{-1}(2)]^{\mathrm{T}} r_1^{(0)} + [T^{-1}(3)]^{\mathrm{T}} r_1^{(1)} + ... + [T^{-1}(2^n-1)]^{\mathrm{T}} r_{n-1}^{(2^{n-1}-1)} = \frac{1}{2^n}\left\{ T(0) r_{dc} + \sum_{l=0}^{n-1} \sum_{k=0}^{2^l-1} 2^l T(2^l+k) r_l^{(k)} \right\}.$$

The forward Haar functions representing each row of the transform T are given by : $T(0) = 1$, $T(1) = (\bar{x}_{n-l} - x_{n-l})$ and $T(I) = T(2^l + k) = u_I(\bar{x}_{n-l} - x_{n-l})$, where u_I is the standard trivial function corresponding to the Haar function $T(I)$. Since $\bar{x}_{n-l} - x_{n-l} = 1$ if $x_{n-l} = 0$ and -1 if $x_{n-l} = 1$, $\bar{x}_{n-l} - x_{n-l} = (-1)^{x_{n-l}}$. The standard trivial function u_I can be represented by the product $\prod_{i=n-l+1}^{n} x_i^{k_{i-n+l}}$, where $k_{i-n+l} \in \{0, 1\}$ is the $(i-n+l)$-th bit in the binary l-tuple of the order k and $x_i^{k_{i-n+l}} = x_i$ if $k_{i-n+1} = 1$ and $x_i^{k_{i-n+l}} = \bar{x}_i$ if $k_{i-n+1} = 0$. Thus,

$$F(X) = \frac{1}{2^n}\left\{ r_{dc} + (-1)^{x_n} r_0^{(0)} + \sum_{l=1}^{n-1} \sum_{k=0}^{2^l-1} 2^l (-1)^{x_{n-l}} r_l^{(k)} \prod_{i=n-l+1}^{n} x_i^{k_{i-n+l}} \right\}$$

$$= \frac{1}{2^n}\left\{ r_{dc} + (-1)^{x_n} r_0^{(0)} + \sum_{l=1}^{n-1} 2^l (-1)^{x_{n-l}} \sum_{k=0}^{2^l-1} r_l^{(k)} \prod_{i=n-l+1}^{n} x_i^{k_{i-n+l}} \right\}. \qquad \square$$

Example 2 : Consider the four-variable incompletely specified Boolean function $F(X) = \Sigma_{ON}(8, 9, 10, 14, 15) + \Sigma_{DC}(1, 4, 5)$ where the numbers enclosed in $\Sigma_{ON}(...)$ and $\Sigma_{DC}(...)$ indicate the truth and don't care minterms, respectively. The Haar spectrum calculated from the *R*-coded vector of *F* is given by :

$R = [r_{dc} \quad r_0^{(0)} \quad r_1^{(0)} \quad r_1^{(1)} \quad r_2^{(0)} \quad \cdots \quad r_2^{(3)} \quad r_3^{(0)} \quad \cdots \quad r_3^{(7)}]^T = [6.5 \ -3.5 \ -0.5 \ 1 \ 0.5 \ 1 \ 1 \ -2 \ -0.5 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0]^T$.

From Theorem 1, the Haar expansion of this function is given by :

$F(X) = \frac{1}{16}\{6.5 - (-1)^{x_4}(3.5) + (-1)^{x_3}(-\bar{x}_4 + 2x_4) + (-1)^{x_2}(2\bar{x}_4\bar{x}_3 + 4\bar{x}_4x_3 + 4x_4\bar{x}_3 - 8x_4x_3) + (-1)^{x_1}(-4\bar{x}_4\bar{x}_3\bar{x}_2 + 8x_4\bar{x}_3x_2)\}$.

Consider the input assignment $X = 5 = 0101_2$, i.e., $x_4 = x_2 = 0$ and $x_3 = x_1 = 1$. The *R*-coded value of *F* under this input assignment can be calculated from the above expansion. $F(5) = \frac{1}{16}(6.5 - 3.5 + 1 + 4 + 0) = 0.5$.

Definition 6 : A *Paired Haar transform* (*PHT*) for an incompletely specified *n*-variable Boolean function *F* is a mapping $\chi : (F_{ON}, F_{DC}) \rightarrow (R_{ON}, R_{DC})$, where $R_{ON} = T \times F_{ON}$ and $R_{DC} = T \times F_{DC}$. F_{ON} is obtained by replacing all don't care outputs of *F* by 0s, and F_{DC} is obtained from *F* by replacing all true outputs by 0s and don't care outputs by 1s. *T* is the unnormalized Haar transform defined in (3). The tuple (R_{ON}, R_{DC}) is known as the *Paired Haar spectrum*. Spectral coefficients from spectra R_{ON} and R_{DC} are indicated by lower case letters accordingly.

In *R* coding, the unnormalized Haar spectrum is related to the Paired Haar spectrum as follows :

$$R = R_{ON} + 0.5 \times R_{DC}. \qquad (7)$$

The Paired Haar spectrum for an incompletely specified *n*-variable Boolean function *F* is composed of 2^n vectors, each having two elements. The elements in the first vector are denoted by $(r_{ON})_{dc}$ and $(r_{DC})_{dc}$, and in the remaining vectors by $(r_{ON})_l^{(k)}$ and $(r_{DC})_l^{(k)}$, where $0 \le l \le n-1$, $0 \le k \le 2^l-1$. Each coefficient can be interpreted as the cardinality of the interception of true and don't care sets of the function with the appropriate PSTF and NSTF. Let $a_l^{(k)}$ be the number of true minterms of *F* in NSTF, $b_l^{(k)}$ the number of true minterms of *F* in PSTF, $c_l^{(k)}$ the number of don't care minterms of *F* in NSTF and $d_l^{(k)}$ is the number of don't care minterms of *F* in PSTF. Let a_{dc} and c_{dc} represents the total number of true and don't care minterms, respectively. Then, $(r_{ON})_{dc} = a_{dc}$, $(r_{DC})_{dc} = c_{dc}$, $(r_{ON})_l^k = a_l^{(k)} - b_l^{(k)}$ and $(r_{DC})_l^{(k)} = c_l^{(k)} - d_l^{(k)}$.

For completely specified function, Paired Haar spectrum becomes the unnormalized Haar spectrum since for completely specified Boolean functions $c_l^{(k)}$ and $d_l^{(k)}$ are always 0.

Example 3 : Figure 3 shows the Karnaugh maps of the standard trivial functions for unnormalized Haar transform of order 16. The NSTF and PSTF are filled with triangles and circles, respectively, on each map. For the four-variable incompletely specified Boolean function $F(X) = \Sigma_{ON}(8, 9, 10, 14, 15) + \Sigma_{DC}(1, 4, 5)$ from Example 2, the values $a_l^{(k)}, b_l^{(k)}, c_l^{(k)}$ and $d_l^{(k)}$ are listed below each Karnaugh map. From Figure 3, the Paired Haar spectrum $(R_{ON}, R_{DC}) = [((r_{ON})_{dc}, (r_{DC})_{dc}) \quad ((r_{ON})_0^{(0)}, (r_{DC})_0^{(0)}) \quad ((r_{ON})_1^{(0)}, (r_{DC})_1^{(0)}) \ ... \ ((r_{ON})_3^{(7)}, (r_{DC})_3^{(7)})]^T = [(5, 3) \ (-5, 3) \ (0, -1) \ (1, 0) \ (0, 1) \ (0, 2) \ (1, 0) \ (-2, 0) \ (0, -1) \ (0, 0) \ (0, 0) \ (0, 0) \ (0, 0) \ (1, 0) \ (0, 0) \ (0, 0)]^T$.

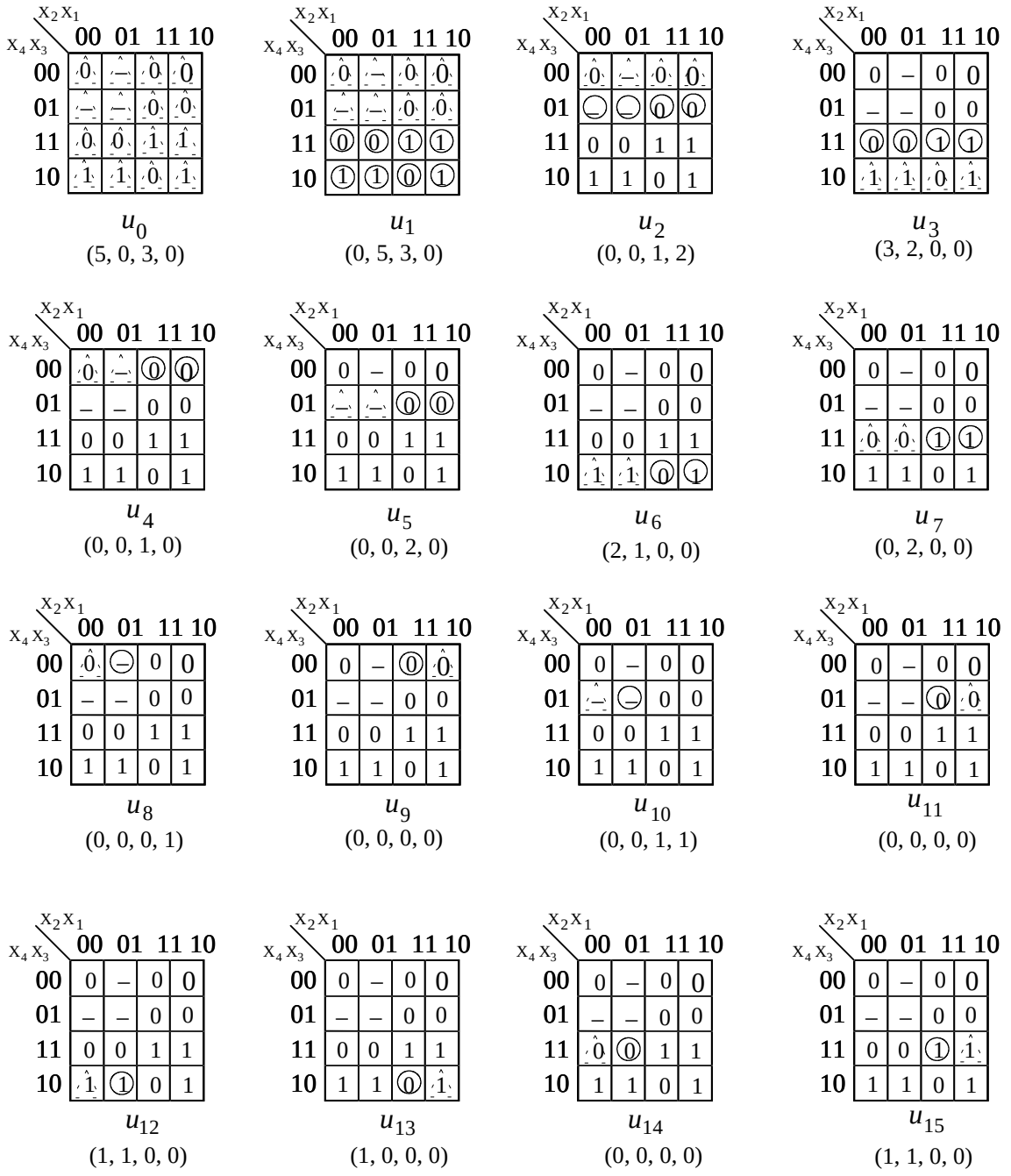

Figure 3: Number of minterms in PSTF and NSTF of unnormalized Haar transform of an four-variable incompletely specified Boolean function.

Table 1 shows the Paired Haar spectra for some completely and incompletely *n*-variable elementary functions. In Table 1, *z* is the number of '1s' in the binary *n*-tuple of the order *k*. The abbreviations OR, NOR, AND, NAND, XOR and XNOR are used for the Boolean expressions $x_1 \vee x_2 \vee ... \vee x_n$, $\overline{x_1 \vee x_2 \vee \cdots \vee x_n}$, $x_1 \wedge x_2 \wedge ... \wedge x_n$, $\overline{x_1 \wedge x_2 \wedge \cdots \wedge x_n}$, $x_1 \oplus x_2 \oplus ... \oplus x_n$ and $\overline{x_1 \oplus x_2 \oplus \cdots \oplus x_n}$, respectively. For completely specified elementary functions, $DC(F) = \varnothing$. The incompletely specified elementary functions that are considered are the functions whose ON and DC arrays can be described by the above elementary functions or the functions whose ON array is empty and DC array is described by an elementary function. The

latter will not happen in the specification of the original Boolean function but may appear in some subfunctions in the process of decomposition. Unless otherwise specified, the values of l and k in Table 1 are integers in the following ranges : $l = 0, 1, ..., n-1$, $k = 0, 1, ..., 2^l-1$.

Property 6 : The maximum value of any parameter $a_l^{(k)}$, $b_l^{(k)}$, $c_l^{(k)}$ or $d_l^{(k)}$ associated with the Paired Haar coefficient of degree l is equal to 2^{n-l-1}.

Property 7 : The sum of all parameters $a_l^{(k)}$and $b_l^{(k)}$ with the same degree l is equal to a_{dc} and the sum of all parameters $c_l^{(k)}$and $d_l^{(k)}$ with the same degree l is equal to c_{dc}. i.e.,

$$\sum_{k=0}^{2^l-1}(a_l^{(k)} + b_l^{(k)}) = a_{dc}, \quad (8)$$

$$\sum_{k=0}^{2^l-1}(c_l^{(k)} + d_l^{(k)}) = c_{dc}. \quad (9)$$

Property 8 : If $(r_{ON})_l^{(k)} + (r_{DC})_l^{(k)} = 2^{n-l-1}$ for some l and k, there exists an ON $(n-l-1)$-cube equal to the corresponding NSTF with total assignment of all don't care minterms in the NSTF. Similarly, if $(r_{ON})_l^{(k)} + (r_{DC})_l^{(k)} = -2^{n-l-1}$ for some l and k, there exists an ON $(n-l-1)$-cube equal to the corresponding PSTF with total assignment of all don't care minterms in the PSTF.

Lemma 1 : With the consideration of all possible assignments of don't care minterms in an incompletely specified Boolean function, the cardinality of the largest prime implicant p, has the upper and lower bounds given by :

$$2^{n-lmax-1} \le p \le 2^{\log_2\lfloor (r_{ON})_{dc} + (r_{DC})_{dc} \rfloor}, \quad (10)$$

where $\lfloor x \rfloor$ means the largest integer no greater than x. $lmax$ is the maximum degree among all degrees l that has an order k such that $(r_{ON})_l^{(k)} + (r_{DC})_l^{(k)} = 2^{n-l-1}$ or -2^{n-l-1}.

Proof : The upper bound of any implicant can not exceed the total number of true and don't care minterms which is given by the sum $a_{dc} + c_{dc} = (r_{ON})_{dc} + (r_{DC})_{dc}$. Since the cardinality of any implicant is an integer exponent of base 2, the upper bound of (10) is proven. Since the cardinality of the largest prime implicant is always larger than or equal to the smallest implicant, the lower bound is trivially obtained from Property 8. □

Lemma 2 : The Paired Haar spectrum for the complement $(\overline{F})$ of Boolean function F is given by :

$$((r_{ON})'_{dc}, (r_{DC})'_{dc}) = (2^n - (r_{ON})_{dc} - (r_{DC})_{dc}, (r_{DC})_{dc}), \text{ and}$$
$$((r_{ON})_l^{(k)\prime}, (r_{DC})_l^{(k)\prime}) = (-(r_{ON})_l^{(k)} - (r_{DC})_l^{(k)}, (r_{DC})_l^{(k)}), \quad (11)$$

where prime superscripts $'$ are used to indicate the spectral coefficients of the complement function $\overline{F}$. $l = 0, 1, ..., n-1$ and $k = 0, 1, ... , 2^l-1$.

Proof : Complementing a Boolean function changes the ON minterms to OFF minterms and vice versa but leaves the DC minterms unchanged. Thus, $(r_{DC})'_{dc} = (r_{DC})_{dc}$ and $(r_{DC})_l^{(k)\prime} = (r_{DC})_l^{(k)}$. $(r_{ON})'_{dc}$ = The number of ON minterms of $\overline{F}$ = number of OFF minterms in $F = 2^n$– number of ON and DC minterms of $F = 2^n - (r_{ON})_{dc} - (r_{DC})_{dc}$. The number of ON minterms of $\overline{F}$ in NSTF (or PSTF) = number of OFF minterms of F in NSTF (or PSTF) = number of cells in NSTF (or PSTF) – number of ON and DC minterms of F in NSTF (or PSTF). For any degree l, the number of cells in the NSTF and PSTF are both equal to 2^{n-l-1}. By definition, number of ON and DC minterms of F in NSTF = $a_l^{(k)} + c_l^{(k)}$ and number of ON and DC minterms in PSTF = $b_l^{(k)} + d_l^{(k)}$. Hence, $(r_{ON})_l^{(k)\prime} = 2^{n-l-1} - a_l^{(k)} - c_l^{(k)} - (2^{n-l-1} - b_l^{(k)} - d_l^{(k)}) = b_l^{(k)} - a_l^{(k)} + d_l^{(k)} - c_l^{(k)} = -(r_{ON})_l^{(k)} - (r_{DC})_l^{(k)}$. □

Lemma 3 : If an n-variable Boolean function is independent of the variable x_i , then

$$\sum_{k=0}^{2^{n-i}-1} (r_{ON})_{n-i}^{(k)} = 0. \quad (12)$$

Proof : If an n-variable Boolean function is independent of the variable x_i, then $\sum_{k=0}^{2^{n-i}-1} a_{n-i}^{(k)} = \sum_{k=0}^{2^{n-i}-1} b_{n-i}^{(k)}$.

The value of r_{DC} is not important as the values of the don't care minterms covered by x_i can be set to 0 to make x_i redundant if required. Since $r_{ON} = a_{ON} - b_{ON}$ for any degree l and order k, (12) is obtained. □

Table 1 Paired Haar spectra for elementary functions.

ON(F)	DC(F)	Non-vanishing coefficients of Paired Haar Spectrum
OR	∅	$(r_{ON})_{dc} = 2^n - 1, (r_{ON})_l^{(0)} = -1 \; \forall l$
NOR	∅	$(r_{ON})_{dc} = 1, (r_{ON})_l^{(0)} = 1 \; \forall l$
∅	OR	$(r_{DC})_{dc} = 2^n - 1, (r_{DC})_l^{(0)} = -1 \; \forall l$
∅	NOR	$(r_{DC})_{dc} = 1, (r_{ON})_l^{(0)} = 1 \; \forall l$
OR	NOR	$(r_{ON})_{dc} = 2^n - 1, (r_{ON})_l^{(0)} = -1 \; \forall l; (r_{DC})_{dc} = 1, (r_{DC})_l^{(0)} = 1 \; \forall l$
NOR	OR	$(r_{ON})_{dc} = 1, (r_{ON})_l^{(0)} = 1 \; \forall l; (r_{DC})_{dc} = 2^n - 1, (r_{DC})_l^{(0)} = -1 \; \forall l$
AND	∅	$(r_{ON})_{dc} = 1, (r_{ON})_l^{(2^l-1)} = -1 \; \forall l$
NAND	∅	$(r_{ON})_{dc} = 2^n - 1, (r_{ON})_l^{(2^l-1)} = 1 \; \forall l$
∅	AND	$(r_{DC})_{dc} = 1, (r_{DC})_l^{(2^l-1)} = -1 \; \forall l$
∅	NAND	$(r_{DC})_{dc} = 2^n - 1, (r_{DC})_l^{(2^l-1)} = 1 \; \forall l$
AND	NAND	$(r_{ON})_{dc} = 1, (r_{ON})_l^{(2^l-1)} = -1 \; \forall l; (r_{DC})_{dc} = 2^n - 1, (r_{DC})_l^{(k)} = 1 \; \forall l$
NAND	AND	$(r_{ON})_{dc} = 2^n - 1, (r_{ON})_l^{(2^l-1)} = 1 \; \forall l; (r_{DC})_{dc} = 1, (r_{DC})_l^{(2^l-1)} = -1 \; \forall l$
XOR	∅	$(r_{ON})_{dc} = 2^{n-1}, (r_{ON})_{n-1}^{(k)} = (-1)^{z+1} \; \forall k$
XNOR	∅	$(r_{ON})_{dc} = 2^{n-1}, (r_{ON})_{n-1}^{(k)} = (-1)^{z} \; \forall k$
∅	XOR	$(r_{DC})_{dc} = 2^{n-1}, (r_{DC})_{n-1}^{(k)} = (-1)^{z+1} \; \forall k$
∅	XNOR	$(r_{DC})_{dc} = 2^{n-1}, (r_{DC})_{n-1}^{(k)} = (-1)^{z} \; \forall k$
XOR	XNOR	$(r_{ON})_{dc} = 2^{n-1}, (r_{ON})_{n-1}^{(k)} = (-1)^{z+1} \; \forall k; (r_{DC})_{dc} = 2^{n-1}, (r_{DC})_{n-1}^{(k)} = (-1)^{z} \; \forall k$
XNOR	XOR	$(r_{ON})_{dc} = 2^{n-1}, (r_{ON})_{n-1}^{(k)} = (-1)^{z} \; \forall k; (r_{DC})_{dc} = 2^{n-1}, (r_{DC})_{n-1}^{(k)} = (-1)^{z+1} \; \forall k$

4. Calculation of Paired Haar Spectrum from Disjoint Cubes

Although hardware based fast Haar chips exist [3, 5], very few algorithms are available for calculating discrete unnormalized Haar spectrum that can be incorporated directly into computer-aided design tools of VLSI digital circuits. In order to enhance the effectiveness and efficiency of the newly introduced Paired Haar transform in CAD and software applications, it is important to develop fast algorithm for the calculation of their spectra. Here the Boolean function from which the Paired Haar spectrum is calculated is represented by an array of disjoint cubes [8, 19, 20]. The advantages of the presented algorithm are that it allows the independent calculation of only some selected coefficients, and the partial spectral coefficients contributed by each disjoint ON or DC cube can be executed concurrently in parallel dedicated processors.

Definition 7 : The *partial spectral coefficient* of an ON or a DC *p*-cube of a Boolean function *F* is equal to the value of the spectral coefficient that corresponds to the contribution of this cube to the full *n*-space spectrum of the Boolean function *F*. The number of partial spectral coefficients *npsc* describing the Boolean function *F* is equal to the number of ON and DC cubes describing this function.

Property 9: The *partial dc coefficient* $((r_{ON})_{dc}, (r_{DC})_{dc})$ contributed by a *p*-cube *C* of a Boolean function *F* is equal to $(2^p, 0)$ if *C* is an ON cube and equal to $(0, 2^p)$ if *C* is a DC cube.

Property 10 : Each ON(or DC) cube contributes a partial Paired Haar spectral coefficient $(r_{ON})_l^{(k)}$(or $(r_{DC})_l^{(k)}$) of degree *l* and order *k* depending on the logical value of the literal x_{n-l} ($0 \le l \le n-1$, x_n is the MSB and x_1 is the LSB). Each literal x_i ($1 \le i \le n$) of a *p*-cube *C* contributes a value *v* to the Paired Haar coefficient, $(r_{ON})_{n-i}^{(k)}$ if *C* is an ON cube and to $(r_{DC})_{n-i}^{(k)}$ if *C* is a DC cube. Depending on the literal x_i and the order *k* of the spectral coefficient, the value *v* is given by :

$$v = \begin{cases} 2^{p-q} & \text{if } k \subseteq \rho_i(C) \text{ and } x_i = \text{'0'} \\ -2^{p-q} & \text{if } k \subseteq \rho_i(C) \text{ and } x_i = \text{'1'} \\ 0 & \text{Otherwise} \end{cases}, \qquad (13)$$

where $\rho_i(C)$ is the cube obtained by shifting the cube *C i* bits to the right, and *q* is the number of '–' in the cube $\rho_i(C)$, i.e., $q = \log_2 |\rho_i(C)|$. $v = 0$ if x_i = '–' or the binary representation of the order *k* is not covered by the cube $\rho_i(C)$.

Based on Property 10, the procedure to calculate the partial Paired Haar spectral coefficient $((r_{ON})_l^{(k)}, (r_{DC})_l^{(k)})$ contributed by a *p*-cube of an *n*-variable Boolean function *F* is given in Figure 4.

```
Procedure partial_coeft (Paired Haar spectrum PHS, cube C, degree l)
{
    order_list = {k∈Z | k ⊆ ρ_{n-l}(C)};
    for (each integer k in order_list) {
      p = number of '–' in C ; q = number of '–' in ρ_{n-l}(C);
      if (bit x_{n-l} of C = 0) v = 2^{p-q};
      else if (bit x_{n-l} of C = 1) v = -2^{p-q};
      if (lookup(PHS, l, k, (r_ON)_l^(k), (r_DC)_l^(k)) = 0) create((r_ON)_l^(k), (r_DC)_l^(k));
      if (C is an ON cube) (r_ON)_l^(k) += v ;
      else if (C is a DC cube) (r_DC)_l^(k) += v ;
      if ((r_ON)_l^(k) = 0 and (r_DC)_l^(k) = 0) remove(PHS, l, k);
      else insert(PHS, l, k, (r_ON)_l^(k), (r_DC)_l^(k));
    }
}
```

Figure 4 : Procedure for calculating the partial spectrum contributed by a *p*-cube.

In Figure 4, order_list is an array of integers representing the minterms covered by the cube $\rho_{n-l}(C)$. *PHS* is a link list of non-zero valued Paired Haar spectral coefficients sorted in ascending order of degree *l* and order *k*. The routine **lookup** searches from *PHS* for any non-zero Paired Haar coefficient of degree *l* and order *k*. If found, it returns the coefficient in the tuple $((r_{ON})_l^{(k)}, (r_{DC})_l^{(k)})$. Otherwise, the routine **create** is called to allocate new Paired Haar coefficient of degree *l* and order *k*. If the computed values of $(r_{ON})_l^{(k)}$ and $(r_{DC})_l^{(k)}$ are both equal to zero, the routine **remove** is called to remove the Paired Haar coefficient of

degree l and order k from *PHS*. Otherwise, the routine **insert** is called to insert the non-zero coefficient in *PHS* according to l and k. The partial dc coefficient can be easily computed from the cardinality of the cube C by Property 9. By summing up the respective partial coefficients contributed by all disjoint cubes, the full Paired Haar spectrum for the n-variable Boolean function F is obtained. The algorithm in Figure 5 describes the procedure of calculating the complete Paired Haar spectrum.

```
Procedure Paired_Haar(Array of disjoint ON and DC cubes D)
{
    Initialize(PHS);
    foreach (cube C_j ∈ D, j = 1 to npsc) {
        p = number of '–' in C_j ;
        if (C_j is an ON cube) (r_ON)_dc += 2^p ;
        else if (C_j is a DC cube) (r_DC)_dc += 2^p ;
        for (l = 0 to n–1) partial_coeft (PHS, C_j, l);
    }
    return PHS ;
}
```

Figure 5: Algorithm to calculate the complete Paired Haar spectrum.

In Figure 5, the routine **Initialize** sets up the link list *PHS* and initializes the dc coefficient $(r_{ON})_{dc}$ and $(r_{DC})_{dc}$ to 0. The number of partial spectral coefficients *npsc* is equal to the number of disjoint ON and DC cubes. To conserve disk space, it is sufficient to store only the non vanishing Paired Haar coefficients. The Procedure **Paired_Haar** can be modified to include options to just calculate a selected Paired Haar coefficient or only spectral coefficients for a complete degree. In the former case, the Procedure **partial_coef** can be simplified to accept the desired degree l and order k as arguments. In the latter case, the degree l is supplied as an additional input argument to Procedure **Paired_Haar** and the for loop involving l is omitted.

Example 4 : An example for calculating the Paired Haar spectrum by Procedure **Paired_Haar** is shown in Table 2. The four-variable incompletely specified Boolean function used in this example is $F(X) = \Sigma_{ON}(1, 2, 3, 5, 6, 7, 10, 11, 12, 13, 14, 15) + \Sigma_{DC}(0)$. The disjoint ON and DC cubes describing F are given in the first row of Table 2. Since there are four disjoint ON and DC cubes, *npsc* = 4. The column under each disjoint cube shows its corresponding partial spectral coefficients. The total spectrum obtained by summing all partial coefficients is given in the last column.

The calculation of one of the Paired Haar spectral coefficients $((r_{ON})_1^{(0)}, (r_{DC})_1^{(0)})$ is demonstrated as follows :

Since $l = 1$, $n–l = 3$. For the ON cube ––1–, $p = 3$, ρ_3(––1–) = 000–, $q = 1$. Since x_3 = '–', $(r_{ON})_1^{(0)} = 0$. For the ON cube 110–, ρ_3(110–) = 0001. Since $k = 0 = 00002 \not\subset 0001$, $(r_{ON})_1^{(0)} = 0$. For the ON cube 0–01, $p = 1$, ρ_3(0–01) = 0000, $q = 0$. Since x_3 = '–', $(r_{ON})_1^{(0)} = 0$. For the DC cube 0000, ρ_3(0000) = 0000, $p = q = 0$. Since $k = 0 = 0000_2 \subseteq 0000$ and x_3 = '0', $(r_{DC})_1^{(0)} = 2^{0-0} = 1$. Hence, $((r_{ON})_1^{(0)}, (r_{DC})_1^{(0)}) = (0, 1)$.

TABLE 2 Calculation of Paired Haar spectrum from an array of disjoint cubes.

Cube		– –1–	110 –	0–01	0000	Total spectrum
l	k	$(r_{ON})_l^{(k)}$	$(r_{ON})_l^{(k)}$	$(r_{ON})_l^{(k)}$	$(r_{DC})_l^{(k)}$	$((r_{ON})_l^{(k)}, (r_{DC})_l^{(k)})$
dc coeficient		8	2	2	1	(12, 1)
0	0	0	-2	2	1	(0, 1)
1	0	0	0	0	1	(0, 1)
1	1	0	-2	0	0	(-2, 0)
2	0	-2	0	1	1	(-1, 1)
2	1	-2	0	1	0	(-1, 0)
2	2	-2	0	0	0	(-2, 0)
2	3	-2	2	0	0	(0, 0)
3	0	0	0	-1	1	(-1, 1)
3	1	0	0	0	0	(0, 0)
3	2	0	0	-1	0	(-1, 0)
3	3	0	0	0	0	(0, 0)
3	4	0	0	0	0	(0, 0)
3	5	0	0	0	0	(0, 0)
3	6	0	0	0	0	(0, 0)
3	7	0	0	0	0	(0, 0)

The algorithm **Ordered_Paired_Haar** is implemented in C, and the computation time and space requirement of the Paired Haar spectra for some MCNC benchmark functions are given in Table 3. The MCNC benchmark functions in PLA format are preprocessed by the disjoint cube algorithm [19, 20] before the test. The number of disjoint cubes is given in the fourth column labeled #disjoint in Table 3. The number of input and output variables of each function are also given in the second and third columns, respectively. The column labeled #coefficients is the number of non-vanishing Paired Haar coefficients and the column labeled Time is the system execution time in seconds on a HP Apollo Series 735 workstation.

TABLE 3 Benchmark results for **Ordered_Paried_Haar**.

Functions	#inputs	#outputs	#disjoint	#coefficients	Time (s)
9sym	9	1	145	211	0.01
Z9sym	9	1	185	211	0.03
5xp1	7	10	75	128	0.02
Z5xp1	7	10	128	128	0.04
alu4	14	8	1043	12008	0.16
sao2	10	4	96	102	0.03
apex4	9	19	523	511	0.07
bw	5	28	106	29	0.03
clip	9	5	176	504	0.02
con1	7	2	11	85	0.03
inc	7	9	33	128	0.01
misex1	8	7	32	232	0.03
misex3	14	14	164	3168	0.15
misex3c	14	14	2945	5404	0.17
table3	14	14	179	8992	0.02
table5	17	15	166	78011	0.32
sqrt8	8	4	40	255	0.03
t481	16	1	887	28231	0.27
b12	15	9	70	28880	0.15
ex1010	10	10	1017	1021	0.05
squar5	5	8	32	32	0.02
xor5	5	1	16	17	0.04
rd53	5	3	32	32	0.02
rd73	7	3	141	128	0.04
rd84	8	4	256	256	0.02

5. Calculation of Haar Spectra from Binary Decision Diagrams

Due to the importance of both spectral and BDD representations in different applications of logic design, there is a growing research interest in finding the mutual relations between them. Recently, we have investigated the links between OBDDs and Haar spectra of Boolean functions [6, 12, 14]. The following definitions associate the standard characteristic set of a Haar spectral coefficient with paths in the FBDD representation.

Definition 8 : A *complete path* is a path that contains vertices of all variables present in the standard characteristic set of a spectral coefficient being calculated.

Definition 9 : An *indexed edge* is an edge of a vertex whose top variable is present in the standard characteristic set of a spectral coefficient being calculated. Conversely, a *non*

indexed edge is an edge of a vertex whose top variable is absent in the standard characteristic set.

For any spectrum used in digital logic design, a set of positive and negative standard trivial functions (PSTFs and NSTFs) can always be determined from the positions of the '1' and '−1' entries in its transformation matrix. From the difference between the number of minterms covered by the interception of the cover of a Boolean function and PSTFs and NSTFs, an unified and systematic approach to the computation of the spectral coefficients for any transform from FBDD is derived. The general principle is illustrated in Figure 6.

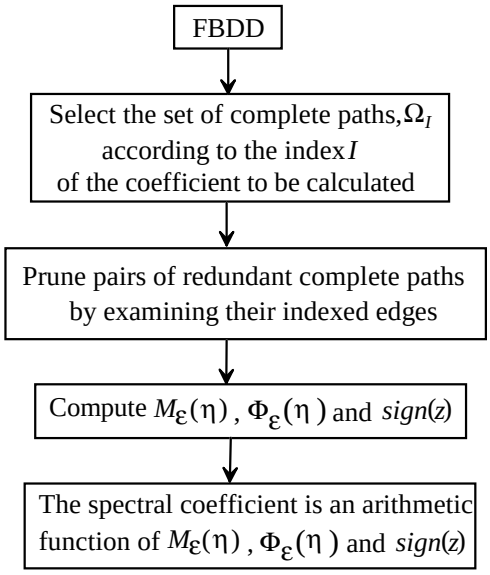

Figure 6: General principle for calculating any spectral coefficient from FBDD.

In Figure 6, $M_\varepsilon(\eta)$ represents the number of minterms covered by each path η in Ω_I, $\Phi_\varepsilon(\eta)$ represents the value assigned to the path η in Ω_I according to whether it contains an odd or even number of indexed edges, and $sign(z) = 1$ or -1 depending on whether the order z of the coefficient being calculated is odd or even. Since the PSTF is a tautology for the dc coefficient of most transforms, the dc coefficient is usually calculated separately. In the sequel, specific procedures applying the principle from Figure 6 to the calculation of the Haar spectra will be shown. Either the set of ON and DC paths or the set of OFF and DC paths are considered, preferably the set with the fewer number of paths is selected. If the set of OFF and DC paths is selected, the negation in spectral domain is required.

Computation of Haar spectrum from FBDD

Unlike Hadamard-Walsh transform, Haar transform is local and does not possess any recursive standard Kronecker product structure in any ordering. Due to the local property of Haar transform, apart from the first two coefficients, the union of PSTF and NSTF does not occupy the complete Boolean *n*-space. Hence, the definition of the complete path must correlate with the reduced space of the standard trivial functions.

Definition 10 : A *local complete path* is a path that fulfills all the following requirements :

1. it contains vertices of one or more variables present in the standard characteristic set of the spectral coefficient being calculated.
2. all indexed edge values follow the polarities of the variables present in the standard trivial function, i.e., 0 if complemented, 1 if affirmative.
3. it contains a vertex of the extended variable x_{n-l}.

In the above definition, it should be noted that both edges of the vertex containing the extended variable are considered as non-indexed edges.

The local complete paths can be selected by a *matching* process. Each path is either selected or rejected based on the outcome of the comparison of every indexed edge value with the affirmative (logical 1) or negative (logical 0) value of the associated literal in the standard trivial function u_I. Starting from the root, a preorder traversal is performed. If the top variable *xi* (*i = index(v)*) of a non terminal vertex *v* is present in *uI* , only one of its two children will be traversed depending on the polarity of the corresponding literal in *uI* . If the literal is complemented, *low*(*v*) will be visited, otherwise *high*(*v*) will be visited. When a match occurs, the number of indexed edges is also incremented. If the top variable of *v* is the extended variable (i.e., *index(v) = n–l*) or is absent in *uI*, both children of *v* will be traversed. Along the traversal, the top variable of each vertex and its edge value are recorded. During the recursive preorder traversal, whenever a terminus is encountered, a path η is selected. η is a local complete path if the number of indexed edges is greater than 0 and the extended variable is contained in the recorded list. Otherwise, the path is pruned. For the special case of OBDD, the matching process can be sped up by pruning paths that do not contain the extended literal early before the terminus is reached. Let *v* be the present vertex of the OBDD and $u \in \{low(v), high(v)\}$ be the next vertex to be visited according to the above rule. Then, *u* will not be visited if *index(v) > n–l* and *index(u) < n–l*.

The following procedure calculates the Haar spectrum for completely and incompletely specified Boolean functions from FBDD.

Procedure 1 : Calculation of Haar spectrum from FBDD

A. Calculation of dc coefficient, r_{dc}

1. The dc Haar coefficient is calculated as follows :

$$r_{dc} = \sum_{\eta \in \eta(1)} 2^{n-|\eta|} + \tfrac{1}{2} \sum_{\eta \in \eta(-)} 2^{n-|\eta|}, \qquad (14)$$

where η(1) and η(–) are the set of ON- and DC- paths respectively, and *n* is the number of variables of the Boolean function *F*.

B. Calculation of other coefficients, $r_l^{(k)}$

1. Select only the ON- and DC- local complete paths using the matching process described above. The set of such paths is denoted by Ω (*l*, *k*), where *l* and *k* are the degree and order of the Haar coefficient to be calculated.

2. The value of the Haar spectral coefficient $r_l^{(k)}$ is given by :

$$r_l^{(k)} = \sum_{\eta(1) \in \Omega(l,k)} 2^{n-l-\kappa(\eta(1))-1} \Phi(\eta(1)) + \tfrac{1}{2} \sum_{\eta(-) \in \Omega(l,k)} 2^{n-l-\kappa(\eta(-))-1} \Phi(\eta(-)), \qquad (15)$$

where η(1) and η(–) are the ON- and DC- local complete paths respectively. κ(η) = number of non-indexed edges in the path η − 1 (minus 1 since the non-indexed edge of the extended variable is excluded). The sign Φ(η) is equal to "+1" if the edge value of the extended variable x_{n-l} in the local complete path η is 0, and equal to "–1" if the edge value of the extended variable is 1. It is apparent that when *l* = *n*–1, all but an edge of the extended variable in a local complete path are indexed, and hence κ(η) = 0.

If the spectral coefficients are calculated by considering the OFF- local complete paths instead of the ON- local complete paths, η(1) in (14) and (15) are replaced by η(0). The calculated spectrum is the spectrum of the complemented function, $\overline{R}$. The actual spectrum *R* can be calculated from $\overline{R}$ by the following set of equations [12, 14] :

$r_{dc=2^n} - \bar{r}_{dc}$, (16)

$r_l^{(k)} = -\bar{r}_l^{(k)}$ for $l = 0,1, ..., n-1$; $k = 0, 1, ..., 2^l-1$. (17)

Example 5 : As an example, the complete set of Haar coefficients for an incompletely specified Boolean function ON(F) = $\{\bar{x}_4\bar{x}_2x_1, x_4x_2\bar{x}_1, x_4x_3x_2\}$, DC($F$) = $\{x_4\bar{x}_3\bar{x}_2\}$represented by the FBDD in Figure 7 is computed.

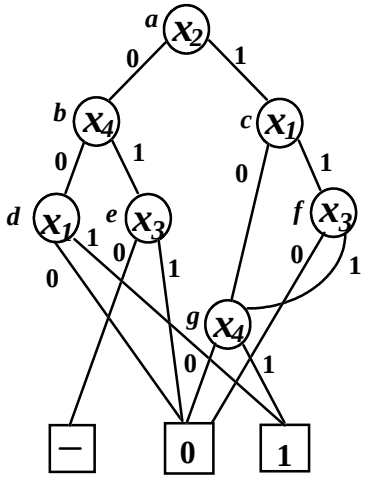

Figure 7: FBDD for the incompletely specified Boolean function, ON(F) = $\{\bar{x}_4\bar{x}_2x_1, x_4x_2\bar{x}_1, x_4x_3x_2\}$ and DC(F) = $\{x_4\bar{x}_3\bar{x}_2\}$.

The ON-paths are $\eta_{abd}(1)$, $\eta_{acg}(1)$ and $\eta_{acfg}(1)$. The only DC-path is $\eta_{abe}(-)$. $|\eta_{abe}(1)| = |\eta_{acg}(1)| = |\eta_{abe}(-)| = 3$ and $|\eta_{acfg}(1)| = 4$. From (14), $r_{dc} = (2+2+1) + 0.5(2) = 6$.

For a sample calculation of $r_0^{(0)}$, $\Omega(0, 0) = \{\eta_{abd}(1), \eta_{ac}g(1), \eta_{acfg}(1), \eta_{abe}(-)\}$.

$\Phi(\eta_{abd}(1)) = 1$, $\Phi(\eta_{acg}(1)) = \Phi(\eta_{acfg}(1)) = \Phi(\eta_{abe}(-)) = -1$.

$\kappa(\eta_{abd}(1)) = \kappa(\eta_{acg}(1)) = \kappa(\eta_{abe}(-)) = 2$ and $\kappa(\eta_{acfg}(1)) = 3$.

From (15), $r_0^{(0)} = 2^{4-0-2-1}(1) + 2^{4-0-2-1}(-1) + 2^{4-0-3-1}(-1) + 0.5\{2^{4-0-2-1}(-1)\} = 2 - 2 - 1 - 1 = -2$.

Since $\Omega(1, 0) = \varnothing$, $r_1^{(0)} = 0$.

All higher degree spectral coefficients are calculated similarly and the results are given in Table 4.

TABLE 4 Calculation of Haar spectral coefficients from FBDD.

l	k	$\Omega(l, k)$	$\Phi(\eta)$	$\kappa(\eta)$	$r_l^{(k)}$
0	0	$\eta_{abd(1)}$	1	2	−2
		$\eta_{acg(1)}$	−1	2	
		$\eta_{acfg(1)}$	−1	3	
		$\eta_{abe(-)}$	−1	2	
1	0	∅	–	–	0
1	1	$\eta_{acfg(1)}$	−1	2	0
		$\eta_{abe(-)}$	1	1	
2	0	$\eta_{abd(1)}$	1	1	1
2	1	$\eta_{acd(1)}$	1	1	1
2	2	$\eta_{acg(1)}$	−1	1	0
		$\eta_{abe(-)}$	1	0	

l	k	$\Omega(l, k)$	$\Phi(\eta)$	$\kappa(\eta)$	$r_l^{(k)}$
¨ 2.00	3	$\eta_{acg(1)}$	−1	1	−2
		$\eta_{acfg(1)}$	−1	1	
3	0	$\eta_{abd(1)}$	−1	0	−1
3	1	∅	–	–	0
3	2	$\eta_{abd(1)}$	−1	0	−1
3	3	∅	–	–	0
3	4	∅	–	–	0
3	5	$\eta_{acg(1)}$	1	0	1
3	6	∅	–	–	0
3	7	$\eta_{acfg(1)}$	−1	0	0
		$\eta_{acg(1)}$	1	0	

TABLE 5 Experimental results for the calculation of Haar spectra.

MCNC	Inputs	Outputs	Number	Time (sec.)
alu2	10	6	441	0.03
alu4	14	8	6185	0.06
9symml	9	1	214	0.05
x2	10	7	97	0.04
parity	16	1	32768	0.19
mux	21	1	35490	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	15709	0.09
cc	21	20	17626	0.07
ttt2	24	21	839126	5.00
pcler8	27	17	163840	0.51
comp	32	3	262136	0.69
pcle	19	9	65792	0.21
sct	19	15	26912	0.08

Table 5 shows the results of the calculation of Haar spectra by Procedure 1. The BDD representations of the benchmark functions are generated before applying our algorithm. The

variables are ordered according to the given circuit topology. In both tables, the columns 'Inputs' and 'Outputs' are the number of inputs and outputs of the benchmark functions respectively. In Table 5, the column 'Number' represents the number of essential non-vanishing Haar spectral coefficients, while the column 'Time' is the system execution time in seconds required to calculate the Haar spectrum on an HP Apollo Series 715 workstation.

5.2 Calculation of the Haar Spectrum in Natural Ordering on Decision Diagrams

In [38] the Haar spectrum calculation based on decision diagrams was proposed. The presented algorithm have taken the advantages of the properties of the Haar functions in the natural ordering. However, this algoritm can be also adapted to the sequency ordering through rearrangement of the calculation and applies to both normalized and unnormalized Haar spectra.

Thanks to the relation (2), the procedure for the Haar spectrum calculation is based on following rule [38] that is performed recursively at all the nodes in the DD, starting from the root node.

$$\boldsymbol{Q}(N,k) = (\boldsymbol{Q}(N_0,\ k-1) \stackrel{\wedge}{+} \boldsymbol{Q}(N_1,k-1) \diamond (\sqrt{2^{n-k}} \stackrel{\wedge}{*} (\boldsymbol{Q}(N_0,k-1) \stackrel{\wedge}{-} \boldsymbol{Q}(N_1,k-1)) \qquad (18)$$

$\boldsymbol{Q}(N,0) = v_N$ if N is the terminal node

where $\diamond$ denotes concatenation of vectors, k is the node level, v_N is the value of the terminal node N, n is the number of variables and $\stackrel{\wedge}{+}$, $\stackrel{\wedge}{-}$ and $\stackrel{\wedge}{*}$ are applied only for the first elements in vectors and they denote addition, subtraction and multiplication, respectively.
Finally the Haar spectrum is determined by:

$$\boldsymbol{Y}_f(n) = \boldsymbol{Q}(root, n). \qquad (19)$$

Relations (18) can be easily modified as follows to calculate the spectrum of the unnormalized Haar transform.

$$\boldsymbol{Q}(N,k) = (\boldsymbol{Q}(N_0,\ k-1) \stackrel{\wedge}{+} \boldsymbol{Q}(N_1,k-1)) \diamond (\boldsymbol{Q}(N_0,k-1) \stackrel{\wedge}{-} \boldsymbol{Q}(N_1,k-1)) \qquad (20)$$

Example 5: By using the procedure the Haar spectrum of the integered-valued three-variable discrete function f = [1, 1, 2, 0, 2, 0,2, 0] is calculated as follows:

$$Q(c,1) = ([2] \stackrel{\wedge}{+} [0]) \diamond 2 \stackrel{\wedge}{*} ([2] \stackrel{\wedge}{-} [0]) = \begin{bmatrix} 2 & 4 \end{bmatrix}^T$$

$$Q(c',1) = ([1] \stackrel{\wedge}{+} [1]) \diamond 2 \stackrel{\wedge}{*} ([1] \stackrel{\wedge}{-} [1]) = \begin{bmatrix} 2 & 0 \end{bmatrix}^T$$

$$Q(b,2) = \left(\begin{bmatrix} 2 \\ 0 \end{bmatrix} \stackrel{\wedge}{+} \begin{bmatrix} 2 \\ 4 \end{bmatrix}\right) \diamond \left(\sqrt{2} \stackrel{\wedge}{*} \left(\begin{bmatrix} 2 \\ 0 \end{bmatrix} \stackrel{\wedge}{-} \begin{bmatrix} 2 \\ 4 \end{bmatrix}\right)\right) = \begin{bmatrix} 4 & 0 & 0 & 4 \end{bmatrix}^T$$

$$Q(b',2) = \left(\begin{bmatrix} 2 \\ 4 \end{bmatrix} \stackrel{\wedge}{+} \begin{bmatrix} 2 \\ 4 \end{bmatrix}\right) \diamond \left(\sqrt{2} \stackrel{\wedge}{*} \left(\begin{bmatrix} 2 \\ 4 \end{bmatrix} \stackrel{\wedge}{-} \begin{bmatrix} 2 \\ 4 \end{bmatrix}\right)\right) = \begin{bmatrix} 4 & 4 & 0 & 4 \end{bmatrix}^T$$

$$Q(a,3) = \left(\begin{bmatrix} 4 \\ 0 \\ 0 \\ 4 \end{bmatrix} \stackrel{\wedge}{+} \begin{bmatrix} 4 \\ 4 \\ 0 \\ 4 \end{bmatrix}\right) \diamond \left(\begin{bmatrix} 4 \\ 0 \\ 0 \\ 4 \end{bmatrix} \stackrel{\wedge}{-} \begin{bmatrix} 4 \\ 4 \\ 0 \\ 4 \end{bmatrix}\right) = \begin{bmatrix} 8 & 0 & 0 & 4 & 0 & 4 & 0 & 4 \end{bmatrix}^T$$

Unlike the other discrete transform and the corresponding DDs based calculation procedures, in the calculation of the Haar spectrum through the recurrence relation (18), the operations $\hat{+}, \hat{-}$ and $\hat{*}$ are applied to the first elements in the sub vectors represented by the nodes in the DD. The other elements of these sub vectors are not involved in the calculation procedure. This feature is due to the local property of Haar transform and results in a considerable simplification of the calculation procedure for the Haar spectrum.

5.3 The Haar Spectral Diagram and Its Properties

In [22], a new data structure called "Haar Spectral Diagram" (or HSD) and its properties have been analyzed. The natural ordered unnormalized Haar functions are used to represent the Haar transform in terms of a Kronecker product yielding a natural decision-diagram based representation. The resulting graph is a point decomposition of the Haar spectrum using edge values. It was shown in [22] that the Haar spectrum represented as an HSD requires no more nodes than the ROBDD for the same function, and for incompletely specified functions, the HSD is isomorphic to the ROBDD.

The natural ordered unnormalized Haar transform can be represented as follows [22]. We first define the $2^n \times 2^n$ "DC" matrix D^n in which the first row is all ones, and all the other elements are zero, then, we separate the Haar transform matrix T^n into the DC part and the remaining "AC" part A^n as:

$$T^n = A^n + D^n \tag{21}$$

We now define the matrix A^n for the "AC" part to be:

$$A^n = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \circledast A^{n-1} + \begin{bmatrix} 0 & 0 \\ 1 & -1 \end{bmatrix} \circledast D^{n-1} \tag{22}$$

where $A^0 = 0$. For $n=3$, this modified Haar Transform matrix has the form as follows, i.e. represents the unnormalized Haar matrix in natural ordering:

$$T^3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \tag{23}$$

Using the above modified Haar matrix, the Haar transform of an n-input Boolean function f can be written as:

$$S^n(f) = A^n Y^n(f) + D^n Y^n(f) \tag{24}$$

Since the first row of A^n is all zeros, and only the first row of D^n is non-zero, $\tilde{S} = AY$ represents the AC components of the spectrum. By (22) we can separate this vector into its low half and high half as:

$$L(\tilde{S}) = A^{n-1} L(Y) \tag{25}$$

$$H(\tilde{S}) = A^{n-1} H(Y) + D^{n-1}(L(Y) - H(Y)) \tag{26}$$

These equations are the basis for a decision diagram which is called the Haar-Spectral Diagram(HSD) [22]. While an HSD can be used to represent any vector, unless stated otherwise, it is assumed that it is being used to represent the Haar transform of some Boolean function. An HSD is an acyclic directed graph having a single terminal node, and

with all other nodes labeled with a variable and having two edges leaving the node labeled "0" and "1". Each of these variable node split the represented vector into a low and a high half. The 1-edge of each non-terminal node, and the top edge of the graph have a "0-element" value. The edge value represents the first element of the vector to which it is applied. That is, an edge value is only applied when the "0" branch is taken for all nodes below it on a path, including any skipped variables or "cross-points". In practice this implies that no more than one non-zero value can be applied on any path. Figure 8 shows an HSD node.

The HSD is the ideal representation for the Haar transform of a Boolean function since the equations for the low half (25) and high half (26) of the spectrum fit naturally onto the definition of the HSD. The right-hand side of the addition in (26) is a vector in which only the first element is non-zero mapping naturally onto the HSD concept of a 0-element edge. Similarly, the right-hand side of (21) is also a vector in which only the first element is non-zero which also maps to the 0-element edge concept for the top node.

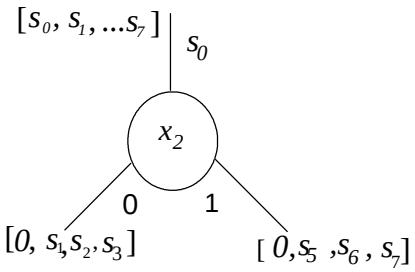

Figure 8: An HSD Node

Two acyclic graphs are said to be isomorphic when they have the same structure, or more precisely(**var**(g), $L^*(g)$ and $H^*(g)$ are respectively the variable index, the low and high edges of node g):

Definition 11 **:** ROBDD graph G_R with nodes r_i and HSD graph G_H with nodes h_j are isomorphic ($G_R \simeq G_H$) if and only if there is mapping function σ such that for any terminal node r_i of G_R,$\sigma(r_i)$ maps to the unique terminal node of G_H, and for any pair of non-terminal nodes the mapping is one-to-one and $h_j = \sigma(r_i)$ holds if and only if **var**(h_j) = **var**(r_i), $L^*(h_j) = \sigma(L^*(r_i))$ and $H^*(h_j) = \sigma(H^*(r_i))$.

Theorem 2 **:** [22] For any completely specified function *f* having a truth table represented by ROBDD $G_R(f)$ and Haar spectrum represented by HSD $G_H(f)$ each with the same variable ordering, $G_R(f)$ and $G_H(f)$ are isomorphic.

Example 7: An example HSD/ROBDD for the function: $x_4x_3\overline{x_2}x_1 + x_4\overline{x_2}x_0 + \overline{x_4}x_3\overline{x_2}\overline{x_1}\overline{x_0} + \overline{x_4}\overline{x_3}x_2\overline{x_1}\overline{x_0}$ is shown in Figure 9. Since the ROBDD is isomorphic to the HSD, the same graph can be interpreted as either an HSD or an ROBDD. The terminal nodes are marked to correspond to the ROBDD terminal values, but are ignored in the HSD interpretation. The "1" edges are indicated by marking them with their HSD edge value. For completely specified functions, the terminal node values for the ROBDD can also be recovered from the HSD edge values. When the "1" edge value of a node is negative, any terminal node from the "1" edge is logic-0, and any terminal node from the "0" edge is a

logic-1. When the “1” edge value is positive, the reverse is true. Furthermore, it can be shown that if the “1” edge value of a node is 0, then that node cannot have a terminal node as a child, since any such node would be removed by the deletion rule.

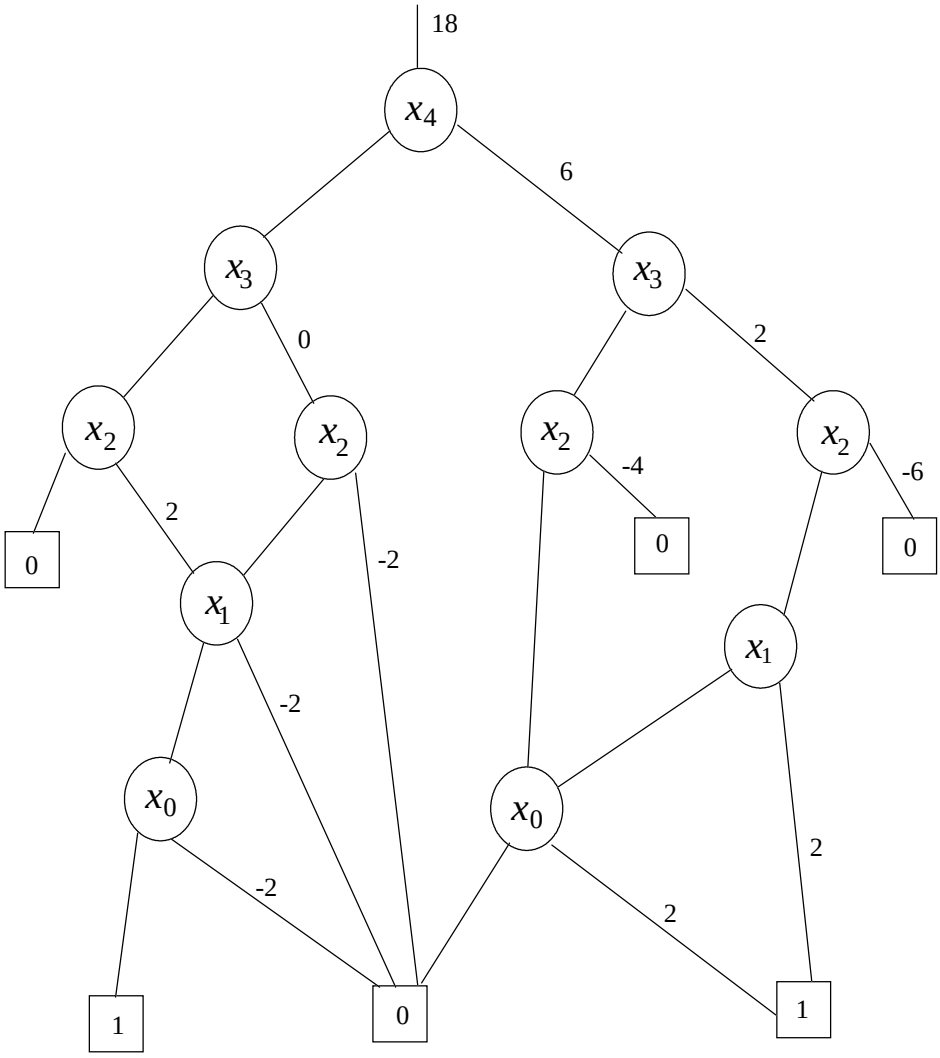

Figure 9: HSD/ROBDD for a five input function.

6. Sign Haar Transform

Various generalizations of Haar transform are known. They were introduced to adapt the original system of Haar functions to some practical applications as well as to extend their applications to different classes of signals. The recent review of some selected generalizations of Haar functions is available [37]. Some of these generalizations include the slant Haar transform [21], Watari transform [41], complex Haar transform [31] and the real multiple-valued Haar transform [39]. In this section another form of Haar transform is considered known as “Sign Haar Transform” [16-18, 30] . The number of different properties of Sign Haar Transform are investigated and presented in this section. The computational advantages of traditional Haar versus Walsh spectra still apply for their corresponding quantized transforms. It is therefore advantageous from the computational points of view to use Sign Haar Transform where Sign Walsh Transform have been used i.e., for Boolean function decomposition and testing of logical circuits [4]. Besides applications in logic

design, the new transform can be used when there is a need for a unique coding of binary/ternary vectors into the spectral domain of the same dimensions. One possible application would be security coding in cryptographic systems using the Sign Haar-χ Walsh-γ Transform [18] and ternary communication systems with the Sign Haar-χ Transform [30].

Besides calculation of both forward and inverse Sign Haar Transforms by using fast flow diagrams which is similar to those of fast Haar Transforms, Sign Haar spectra can be calculated directly from recursive definitions that involve data and transform domain variables.

Definition 12 : An invertible forward Sign Haar Transform $\underline{h}$ is:

$$\underline{h}\left(\overrightarrow{O_n} \oplus_d \omega_1\right) = sign \sum_{x_n=0}^{1}\left[sign \sum_{x_{n-1}=0}^{1}\left[...sign \sum_{x_1=0}^{1}\left\{(-1)^{x_n\omega_1} f\left(\overrightarrow{x_n}\right)\right\}...\right]\right] \tag{27}$$

and

$$\underline{h}\left(\overrightarrow{O_n} \oplus_d \overrightarrow{\omega_p} \oplus_d 2^p\right) = sign \sum_{x_{n-p}=0}^{1}\left[sign \sum_{x_{n-p-1}=0}^{1}\right.$$
$$\left[..sign \sum_{x_1=0}^{1}\left\{(-1)^{x_{n-1}} f\left\{\left[\left(\overrightarrow{O}_n \oplus_d \overrightarrow{\omega}_p\right) \oplus_c (n-p)\right] \oplus_d \overrightarrow{x_{n-p}}\right\}\right\}...\right]\Bigg]. \tag{28}$$

The inverse Transform is:

$$f\left(\overrightarrow{x_n}\right) = sign\left[(-1)^{x_1}\underline{h}\left\{\left[\left(\overrightarrow{O_1} \wedge \overrightarrow{x_n}\right) \oplus_c 1\right] \oplus_d 2^{n-1}\right\} + \right.$$
$$sign\left[(-1)^{x_2}\underline{h}\left\{\left[\left(\overrightarrow{O_2} \wedge \overrightarrow{x_n}\right) \oplus_c 2\right] \oplus_d 2^{n-2}\right\} + ...\right.$$
$$... + sign\left[(-1)^{x_p}\underline{h}\left\{\left[\left(\overrightarrow{O_p} \wedge \overrightarrow{x_n}\right) \oplus_c p\right] \oplus_d 2^{n-p}\right\} + ... + \right.$$
$$sign\left[(-1)^{x_{n-1}}\underline{h}\left\{\left[\left(\overrightarrow{O_{n-1}} \wedge \overrightarrow{x_n}\right) \oplus_c (n-1)\right] \oplus_d 2\right\} + \right.$$
$$\left.\left.\left. sign\left[\sum_{\omega_1=0}^{1}(-1)^{x_n\omega_1}\underline{h}\left(\overrightarrow{O_n} \oplus_d \omega_1\right)\right]\right]...\right]...\right] \Bigg]. \tag{29}$$

In (28), $1 \le p < n$ and in (29), $1 \le p \le n$.

Mutual relations in the definitions of forward and inverse Sign Haar Transforms $\underline{h}$ for p-th ω_p and first transform variable ω_1 are shown. For the forward transform when $p \to 0$, (28) yields $\underline{h}\left(\overrightarrow{O_n} \oplus_d \overrightarrow{\omega_p} \oplus_d 2^p\right) = \underline{h}\left(\overrightarrow{O_n} \oplus_d \overrightarrow{\omega_0} \oplus_d 2^0\right) = \underline{h}\left(\overrightarrow{O_n} \oplus_d 1\right) = \underline{h}\left(\overrightarrow{O_n} \oplus_d \omega_1\right)$ when $\omega_1 = 1$.

Hence for this condition, (27) has been derived. For the inverse transform, the p-th element is $sign\{(-1)^{x_p}\underline{h}\left\{\left[\left(\overrightarrow{O_p} \wedge \overrightarrow{x_n}\right) \oplus_c p\right] \oplus_d 2^{n-p}\right\}$, and when $p \to n$, the equation approaches

$sign\{(-1)^{x_n}\underline{h}\left\{\left[\left(\overrightarrow{O_n} \wedge \overrightarrow{x_n}\right) \oplus_c n\right] \oplus_d 2^{n-n}\right\} = sign\{(-1)^{x_n}\underline{h}\left\{\left[\overrightarrow{O_n} \oplus_c n\right] \oplus_d 1\right\}$

$= sign\{(-1)^{x_n}\underline{h}\left[\overrightarrow{O_n} \oplus_d 1\right] = sign(-1)^{x_n\omega_1}\underline{h}\left(\overrightarrow{O_n} \oplus_d \omega_1\right)$ when $\omega_1 = 1$.

Hence, the n-th element of the recursive definition in (29) has been derived.

In this part, the Sign Haar spectra for common logic functions and the major properties of the transform are presented. There is no direct relationship between the Sign Haar spectra calculated for *S*- and *R*-coded Boolean functions, which differs from other transforms used in logic design (i.e., Walsh, Haar).

In the following presentation of the properties, let f be the original data function and $\underline{h}$ its Sign Haar transform. Let a and b be ternary variables, where $a, b \in \{-1, 0, 1\}$.

Property 11: For arbitrary ternary variables a and b :

$$sign[sign\{a+b\}+sign\{a-b\}] = a \tag{30}$$

and

$$sign[sign\{a+b\}-sign\{a-b\}] = b. \tag{31}$$

Property 12: Let function $f(\overrightarrow{x_n})$ be a constant, such that its ternary vector $\overrightarrow{F}$ has all the coefficients equal and $F_j (0 \le j < N)$; $Fj \in \{-, 0, +\}$. Then,

$$f(\overrightarrow{x_n}) = 0 \;\Leftrightarrow\; \underline{h}(\overrightarrow{\omega_n}) = 0, \qquad x_p, \omega_p \in \{0, 1\} \text{ and } 1 \le p \le n. \tag{32}$$

Also, $$f(\overrightarrow{x_n}) = \pm 1 \;\Leftrightarrow\; \underline{h}(\overrightarrow{\omega_n}) = \pm \prod_{p=1}^{n} (1 - \omega_{p-1}). \tag{33}$$

Property 13 : When an *S*-coded *n*-variable function is functionally dependent on a single Boolean variable in affirmation, i.e.,

$$f(\overrightarrow{x_n}) = f(x_n, ..., x_1) = x_p\ , \quad p \in \{1, ..., n\}, \quad x_p \in \{+1, -1\},$$

the corresponding Sign Haar Transform,

$$\underline{h}(\overrightarrow{\omega_n}) = +1 \times \left(\omega_{n-p+1} \wedge \left\{ \bigwedge_{k=n-p+2}^{n} \overline{\omega}_k \right\} \right), \tag{34}$$

where $\omega_k \in \{0, 1\}$ and the logic AND operation in brackets ( ) will yield the value 1 or 0. If for some *p*, $k > n$, by definition the expression $\bigwedge_{k=n-p+2}^{n} \overline{\omega}_k$ is equal to 1, otherwise the symbol $\overline{\omega}_k$ represents the logical inversion of the transform variable ω_k. The meaning of the symbol $\overline{\omega}_k$ and the restriction on the value of $\bigwedge_{k=n-p+2}^{n} \overline{\omega}_k$ for some *p*, when $k > n$ is also the same as above for Properties 14 - 16.

Property 14 : When an *S*-coded *n*-variable function is functionally dependent on a single Boolean variable in negation, i.e.,

$$f(\overrightarrow{x_n}) = f(x_n, ..., x_1) = \overline{x}_p\ , \quad p \in \{1, ..., n\}, \quad x_p \in \{+1, -1\},$$

its Sign Haar Transform is,

$$\underline{h}(\overrightarrow{\omega_n}) = -1 \times \left(\omega_{n-p+1} \wedge \left\{ \bigwedge_{k=n-p+2}^{n} \overline{\omega}_k \right\} \right). \tag{35}$$

Property 15 : For an *S*-coded *n*-variable Boolean function $f(\overrightarrow{x_n})$ where the Sign Haar spectrum is $\underline{h}(\overrightarrow{\omega_n})$, the spectrum of the negated function is derived simply by inverting all the signs of the original spectra. Hence, when

$$\underline{f(\overrightarrow{x_n})} \Leftrightarrow \underline{h}(\overrightarrow{\omega_n}),$$

then $$\overline{f(\overrightarrow{x_n})} \Leftrightarrow -\underline{h}(\overrightarrow{\omega_n}). \tag{36}$$

Property 16 : When an *R*-coded *n*-variable function is functionally dependent on a single Boolean variable in affirmation, i.e.,

$$f(\overrightarrow{x_n}) = x_p \text{ where } p \in \{1, 2,..., n\} \text{ and } x_p \in \{0,1\},$$

its Sign Haar Transform,

$$\underline{h}(\overrightarrow{\omega_n}) = \bigwedge_{k=1}^{n} \overline{\omega}_k - \left(\omega_{n-p+1} \wedge \left\{ \bigwedge_{k=n-p+2}^{n} \overline{\omega}_k \right\} \right), \tag{37}$$

where $\omega_k \in \{0,1\}$ and the logical AND operation in the bracket ( ) will yield value 1 or 0.

$$\underline{h}_3(\overrightarrow{\omega_4}) = [+, 0, 0, 0, -, -, -, -, 0, 0, 0, 0, 0, 0, 0, 0].$$

Property 17 : When an *R*-coded *n*-variable function is functionally dependent on a single Boolean variable in negation, i.e.,

$$f(\overrightarrow{x_n}) = \overline{x}_p \text{ where } p \in \{1, 2,..., n\} \text{ and } x_p \in \{0,1\},$$

then the Sign Haar Transform

$$\underline{h}(\overrightarrow{\omega_n}) = \bigwedge_{k=1}^{n} \overline{\omega}_k + \left(\omega_{n-p+1} \wedge \left\{ \bigwedge_{k=n-p+2}^{n} \overline{\omega}_k \right\} \right). \tag{38}$$

7. Application of Haar Transform in Logic Design

The advantages of computational and memory requirements of the Haar transform make it of big interest to VLSI designers. For example, the authors of [33-35] 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 means not only for diagnosing digital MOS IC's 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 to derive strategies for testing MOS circuits when memory states were encountered as a consequence of some fault types. The advantage to use Haar functions instead of Walsh functions in CAD systems based on spectral methods for some classes of Boolean functions was shown in [24, 26, 43]. For example, the analysis in [24] 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 spectral multifunctional logical module [24, 26] to generate arbitrary Boolean functions 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 [26, 43]. Karpovsky [24] 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 non vanishing Haar coefficients is reduced with input permutation of variables - the situation which does not apply to Walsh basis. It should be noted that the realization of a permutation requires no special hardware [24]. 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 non vanishing Haar coefficients is reduced.

In many practical problems of logic design and machine learning, weakly specified Boolean functions are frequently encountered [8, 29]. These functions are efficiently represented by the arrays of ON and OFF terms since a majority of their functional domain are don't cares. The local property of the Haar transform makes it of interest in those applications in computer-aided design systems where there are Boolean functions of many variables that have most of their ON-minterms grouped locally. Such weakly specified and local functions can be extremely well described by few spectral coefficients from Haar transform while the application of Walsh, global transform would be quite cumbersome in such cases and the locally grouped minterms would be spread throughout the Walsh spectrum. In most engineering design problems, incompletely specified functions have to be dealt with. The don't care sets derived from circuit structures represent an additional degree of freedom and their effective utilization often results in highly economical circuits. To better deal with the mentioned cases, the concept of *Paired Haar transform* was introduced [11, 13, 15]. In Paired Haar Transform, all the information about true and don't care minterms is kept separately, by what it is available in different stages of CAD process. Useful properties and applications of Paired Haar spectra in logic design, for example, minimization of mixed polarity Reed-Muller expansion, generation of quasi-optimal FBDDs and multiplexer synthesis for incompletely specified Boolean functions, have been demonstrated in [6, 7, 15].
An unified entropy approach operating on Paired Haar spectrum for their heuristic optimization with effective utilization of the don't care sets for incompletely specified Boolean functions have been developed in [7]. For FBDD and OBDD minimization, there is no need to generate an initial BDD with an arbitrary variable ordering followed by improving

the variable ordering with local search or simulated annealing in two steps. The algorithm for the FBDD minimization can be used for multiplexer universal logic module network synthesis in tree type realization by treating each vertex as a set of control variables with multiple children. The extension of the FBDD minimization algorithm to multiplexer synthesis permits mixed control variables within each level if it leads to early termination of more paths with constants or single variables.

8. Conclusion

This paper is a short review of some recent developments in the area of Haar transform in logic design. This survey presented the basic definitions and properties of paired Haar transform and its efficient calculation from disjoint cubes. Various methods of calculation of Haar spectra through decision diagrams were also shown. A novel data structure so called Haar decision diagram was discussed. Essential properties of Sign Haar spectra of Boolean functions were presented. It should be noticed that as in all reviews not all the recent work could be included and discussed. For example in [40] the relationship between the conditional output probabilities of Boolean functions and the Haar transform is discussed. Also the applications of Haar spectra were only mentioned and the interested readers can look into the references for more details. Due to unique local property of Haar spectra there should be more problems in the analysis and synthesis of Boolean and discrete functions that could be possibly solved more efficiently by exploiting the information present in the Haar spectrum.

References

[1] AGAIAN, S., ASTOLA, J., and EGIAZARIAN, K.: ‘Binary polynomial transforms and nonlinear digital filters’ (Marcel Dekker, New York, 1995)

[2] AHMED, N., and RAO, K. R.: 'Orthogonal transforms for digital signal processing' (Springer-Verlag, Berlin,1975)

[3] ALBANESI, M. G., and FERRETTI, M.: ‘A high speed Haar transform implementation’, Journal of Circuits, Systems and Computers, 1992, vol. 2, no. 3, pp. 207-226

[4] BESSLICH, P. W., and TRACHTENBERG, E.A.: ‘A three-valued quasi-linear transformation for logic synthesis’, in ‘Spectral Techniques : Theory and Applications’. C. MORAGA and R. CREUTZBURG, Ed., North Holland: Elsevier, 1992.

[5] BURON, A. M., MICHELL, J. A., and SOLANA, J. M.: 'Single chip fast Haar transform at megahertz rates', in 'Theory and applications of spectral techniques', MORAGA, C. (Ed.) (University Dortmund Press, 1988) pp. 8-17

[6] CHANG, C. H., and FALKOWSKI, B. J.: ‘Generation of Quasi-optimal FBDDs through Paired Haar Spectra’, Proceedings of the 31st IEEE International Symposium on Circuits and Systems (ISCAS'98), Monterey, California, USA, vol. VI, Jun. 1998, pp. 167-170

[7] CHANG, C. H., and FALKOWSKI, B. J.: ‘Haar spectra based entropy approach to quasi-minimization of FBDDs’, *IEE Proc. - Comput. Digit. Tech.* (in press)

[8] CIESIELSKI, M. J., YANG, S., and PERKOWSKI, M. A.: 'Multiple-valued minimization based on graph coloring', Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers & Processors, Oct. 1989, pp. 262-265

[9] CREUTZBURG, R.: ‘Recent results in image compression using the wavelet transform’, in ‘Advances in Spectral Techniques’, MORAGA, C. (Ed) (University Dortmund Press, 1998) pp. 65-68

[10] DE MICHELI, G.: 'Synthesis and optimization of digital circuits' (McGraw-Hill, New York, 1994)

[11] FALKOWSKI, B. J. and CHANG, C.H.: ‘A novel paired Haar based transform: algorithms and interpretations in Boolean domain’, Proceedings 36th Midwest Symp. on Circuits and Systems, Detroit, Michigan, Aug. 1993, pp. 1101-1104

[12] FALKOWSKI, B.J. and CHANG, C. H.: 'Efficient algorithms for the forward and inverse transformations between Haar spectrum and binary decision diagram', Proceedings 13th IEEE Int. Phoenix Conf. on Computers and Communications, Phoenix, Arizona, Apr. 1994, pp. 497-503
[13] FALKOWSKI, B. J., and CHANG, C. H.: 'Properties and applications of Paired Haar transform', Proceedings of the 1st IEEE Int. Conf. on Information, Communications and Signal Processing (ICICS'97), Singapore, Sep. 1997, vol 1, pp. 48-51
[14] FALKOWSKI, B. J., and CHANG, C. H.: 'Forward and inverse transformations between Haar spectra and ordered binary decision diagrams of Boolean functions', *IEEE Trans.*, Nov. 1997, C-46, (11), pp. 1272-1279
[15] FALKOWSKI, B. J., and CHANG, C. H.: 'Calculation of Paired Haar spectra for systems of incompletely specified Boolean functions', Proceedings of the 31st IEEE Int. Symp. on Circuits and Systems (ISCAS'98), Monterey, California, USA, Jun. 1998, vol. VI, pp. 171-174
[16] FALKOWSKI, B.J., and RAHARDJA, S.: `Sign Haar Transform', Proceedings of IEEE International Symposium on Circuits and Systems (27th ISCAS), London, United Kingdom, May 1994, vol. 2, pp. 161-164
[17] FALKOWSKI, B.J., and RAHARDJA, S.: `Properties of Boolean functions in spectral domain of Sign Haar Transform', Proceedings of IEEE International Conference on Information, Communications and Signal Processing (1st ICICS), Singapore, September 1997, vol. 1, pp. 64-68
[18] FALKOWSKI, B.J., and RAHARDJA, S.: `Recursive Relations and Fast Flow Diagram for Non-Linear Quantized Transform', *IEE Proceedings, Computers and Digital Techniques*, March 1998, vol. 145, no. 2, pp. 97-108
[19] FALKOWSKI, B. J., SCHAEFER, I., and CHANG, C. H.: 'An efficient computer algorithm for the calculation of disjoint cubes representation of Boolean functions', Proceedings of the 36th IEEE Midwest Symposium on Circuits and Systems, Detroit, Michigan, Aug. 1993, pp. 1308-1311
[20] FALKOWSKI, B. J., SCHAEFER, I., and PERKOWSKI, M. A.: 'Effective computer methods for the calculation of Rademacher-Walsh spectrum for completely and incompletely specified Boolean functions', *IEEE Trans.*, Oct. 1992, **CAD-11**, (10), pp. 1207-1226
[21] FINO, R. J., and ALGAZI, V. R.: 'Slant Haar transform', *IEE Proc.*, 1974, Vol. 62, pp. 653-654
[22] HANSEN, P, and SEKINE, M: 'Decision diagram based techniques for the Haar wavelet transform', Proceedings of the 1st International Conference on Information, Communications and Signal Processing, Singapore, Sep. 1997, vol 1, pp 59-63
[23] HAAR, A.: 'Zur Theorie der othorgonalen Funktionsysteme', *Math. Ann.*, vol. 69, pp.331-371, 1910.
[24] KARPOVSKY, M. G.: 'Finite orthogonal series in the design of digital devices' (John Wiley, New York, 1976)
[25] KUKHAREV, G. A., SHMERKO, V. P., and YANUSHKEVICH, S. N.: 'Technique of binary data parallel processing for VLSI' (University Press, Minsk, 1991) (in Russian)
[26] KULESZA, W.: 'Systems of spectral analysis of digital data' (WKL, Warsaw, 1984) (in Polish)
[27] MOHARIR, P. S.: 'Pattern-Recognition Transforms' (John Wiley, New York, 1992)
[28] NEWTON, A. R. (Ed.): 'Selected papers on logic synthesis for integrated circuit design' (IEEE Press, New York, 1987)
[29] PERKOWSKI, M. A., ROSS, T., GADD, D., GOLDMAN, J. A., and SONG, N.: 'Application of ESOP minimization in machine learning and knowledge discovery', Proceedings of the IFIP WG10.5 Workshop on Applications of the Reed-Muller Expansion in Circuit Design, Makuhari, Chiba, Japan, Sep. 1995, pp. 102-109
[30] RAHARDJA, S., and FALKOWSKI, B.J.: `Application of Sign Haar Transform in Ternary Communication System', *International Journal of Electronics*, November 1995, vol. 79, no. 5, pp. 551-559
[31] RAO, K. R., REVULURI, K., NARASIMHAN, M.A., and AHMED, N.: 'Complex Haar transform', *IEEE Trans. on Acoustics, Speech and Signal Processing*, Feb. 1976, pp. 102-104
[32] ROESER, P. R., and JERNIGAN, M. E.: 'Fast Haar transform algorithms', *IEEE Trans.*, 1982, **C-31**, (2), pp. 175-177
[33] RUIZ, G., BURON, A., and MICHELL, J. A.: 'NMOS and CMOS physical faults diagnosis by spectral techniques: simulation and associated tools', Proceedings of the IEEE Workshop on Languages for Automation, Vienna, Austria, Aug. 1987, pp. 191-194
[34] RUIZ, G., MICHELL, J. A., and BURON, A.: 'Fault detection and diagnosis for MOS circuits from Haar and Walsh spectrum analysis: on the fault coverage of Haar reduced analysis', in 'Theory and Applications of Spectral Techniques', MORAGA, C. (Ed.) (University Dortmund Press, 1988) pp. 97-106

[35] RUIZ, G., MICHELL, J. A., and BURON, A.: 'Switch-level fault detection and diagnosis environment for MOS digital circuits using spectral techniques', *IEE Proc. E. Comput. Digit. Tech.*, Jul. 1992, 139, (4), pp. 293-307

[36] SHORE, J. E.: 'On the applications of Haar functions', *IEEE Trans.*, 1973, COM-21, pp. 206-216

[37] STANKOVIC, R. S., and FALKOWSKI, B. J.: 'Haar functions and transforms and their generalizations', Proceedings of the 1st IEEE International Conference on Information, Communications and Signal Processing, Singapore (ICICS'97), Sep. 1997, vol 4, pp 1-5

[38] STANKOVIC, M, JANKOVIC, D, and STANKOVIC, R. S.: 'Efficient algorithm for Haar spectrum calculation', Proceedings of the 1st IEEE International Conference on Information, Communications and Signal Processing (ICICS'97), Singapore, Sep. 1997, vol 4, pp. 6-10

[39] STOJIC, M.R., STANKOVIC, M.S., and STANKOVIC, R. S.: 'Discrete transforms and their applications' (Beograd, Nanka, 1993) (in Serbian)

[40] THORNTON, M.A.; 'Modified Haar transform calculation using digital circuit output probabilities', Proceedings of the 1st IEEE International Conference on Information, Communications and Signal Processing (ICICS'97), Singapore, Sep. 1997, vol 1, pp. 52-58

[41] WATARI, C.: 'Generalization of Haar functions', *Tohoku Math. Journal*, 1956, vol. 8, pp. 286-290

[42] YAROSLAVSKY, L. P.: 'Digital picture processing' (Springer-Verlag, Berlin, 1985)

[43] ZALMANZON, L. A.: 'Fourier, Walsh and Haar Transforms and their application in control, communication and other fields' (Nauka, Moscow, 1989) (in Russian)