Paired Haar spectra computation through operations on disjoint cubes

B.J.Falkowski and C.-H.Chang

Abstract: Paired Haar spectra for systems of incompletely specified boolean functions are calculated from arrays of disjoint cubes. The method is based mainly on two basic cube operations of shifting and addition that can be efficiently implemented by computer. The method can calculate only a few selected coefficients or all of them in parallel. To further reduce computational requirement only nonvanishing coefficients are stored.

1 Introduction

Spectral techniques have been applied to boolean function classification, disjoint decomposition, parallel and serial linear decomposition, spectral translation synthesis, multiplexer synthesis, prime implicant extraction by spectral summation, threshold logic synthesis, logic complexity, state assignment, testing and prefiltering in technology mapping [1–9]. In contrast to most traditional CAD systems (MIS-II, BOLD, and Synopsys) based on the unate paradigm [10], spectral systems are very efficient for strongly nonunate functions such as parity, addition, or multiplication that frequently occur in real designs. In addition, spectral techniques alone perform efficiently parallel and serial linear decomposition (extraction of linear pre- and postfilters). There are at least two transforms based on squarewave-like functions that are suitable as boolean functions: Haar and Walsh transforms. All but two basis functions in the Haar transform consists of a square-wave pulse located on an otherwise zero-amplitude interval. When applied to logic design, a nonnormalised Haar transform [1–3, 7, 8] is usually used. The Walsh functions are global like Fourier functions and consist of a set of irregular rectangular waveforms with only two amplitude values +1 and −1 [1, 3, 7, 8, 11–13]. Walsh spectral coefficients of boolean functions have an easy interpretation and efficient methods of calculation of such spectra directly from reduced representation of boolean functions in the form of disjoint cube representation have been introduced in [35]. Computation of the fast Haar transform (FHT) requires order N (N is the number of spectral coefficients) additions and subtractions, which makes it much faster than the fast Walsh transform (FWT) [3, 7, 8, 11, 13–15]. Hardware-based fast Haar chips have been developed [14].

Owing to its low computing requirement, the Haar transform has been used mainly for pattern recognition and image processing [8, 13, 16–18]. Such a transform is also well-suited in communication technology for data coding, multiplexing and digital filtering [8, 16]. The advantages of computational and memory requirements of the Haar transform make it of high interest to VLSI designers as well. For example, the authors of [4–6] presented a set of CAD tools to perform switch-level fault detection and diagnosis of physical faults for practical MOS digital circuits using a reduced Haar spectrum analysis. In their system the nonnormalised 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 Haar functions instead of Walsh functions in CAD systems based on spectral methods for some classes of boolean function was shown in [1, 8]. For example, the analysis in [1] 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 [1, 3] 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 behaviour of the module is useful in real-time adaptive control systems [3, 8]. Karpovsky [1] noticed that the size of the memory block can be optimised only when the Haar basis is used. This is due to the fact that the number of nonvanishing Haar coefficients is reduced with the input permutation of variables, a situation which does not apply to the Walsh basis. The realisation of a permutation requires no special hardware [1]. Another advantage of the Haar spectrum in this application is the smaller 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.

In many practical problems of logic design and machine learning, weakly specified boolean functions are frequently encountered [19–23]. 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

IEE Proceedings online no. 19990327

DOI: 10.1049/ip-cds:19990327

Paper first received 10th August 1998 and in revised form 1st February 1999

The authors are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Blk S1, Nanyang Avenue, Singapore 639798

their ON minterms grouped locally. Such weakly specified and local functions can be extremely well described by a few spectral coefficients from the Haar transform, while the application of the 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 utilisation often results in highly economical circuits. To better deal with the mentioned cases, the concept of the paired Haar transform was introduced [24–28]. In the paired Haar transform, all the information about true and don't- care minterms is kept separately, by what is available in different stages of CAD process. Useful properties and applications of paired Haar spectra in logic design, for example, the minimisation of mixed polarity Reed–Muller expansion, and generation of quasioptimal FBDDs and multiplexer synthesis for incompletely specified boolean functions, have been demonstrated [24, 25, 27].

Although the properties of Haar spectra have considerable interest and attraction, the majority of publications to date have considered the Walsh rather than Haar transform. This is mainly due to the fact that up to now there is no efficient method of calculating Haar spectra directly from reduced representations of boolean functions. Recently, efficient symbolic methods based on the binary decision diagram representation for the computation of nonnormalised Haar spectra have been developed [18, 29, 30]. These methods can be used efficiently in various CAD systems and the decision diagrams can represent both the original boolean functions and their spectra. Binary decision diagrams have proved to be very convenient data structures for the majority of discrete function representations, permitting manipulation and calculation with large discrete functions efficiently in terms of space and time. Therefore they are frequently used to represent data structures in modern CAD VLSI systems [31]. However, some of such systems are based on cubical representation rather than decision diagrams and this article solves the problem of efficient calculation of paired Haar spectrum for such CAD systems [9, 10, 12, 32].

The algorithm presented has overcome the inefficiency of calculation of both spectra directly from the definition of the transforms by matrix multiplication. Even if the more efficient fast transform method of calculating the Haar spectrum was used, it requires the conversion of the cubical representation to the full minterm vector. The new algorithm has the advantage over the last-mentioned methods of operating directly on reduced representation such as cubes. It is especially beneficial for functions of many variables because the number of cubes maybe linearly proportional to the number of variables, and the number of entries in the minterm vector is always exponential to the number of variables. By representing the boolean function in the form of an array of disjoint cubes instead of minterms, the spectral coefficients can be computed more rapidly from such a reduced representation with smaller required memory, while the ability to calculate only partial spectra is still preserved. Hence, the new algorithm has allowed practical applications of paired Haar transforms for CAD Systems using cubical rather than graph based representations of discrete functions.

To use boolean functions that are represented as minterms or arrays of nondisjoint cubes, the input data are preprocessed by a fast algorithm that generates an array of disjoint ON-cubes (in the case of completely specified boolean functions) or disjoint ON- and DC-cubes (in the case of incompletely specified functions). The algorithm that generates such an array and its implementation is described in [33]. For each disjoint cube, the appropriate partial spectral coefficients are calculated. The final paired Haar spectrum is found by adding all the corresponding partial coefficients contributed by the complete array of disjoint cubes.

2 Basic definitions

An n-variable boolean function $F(x_1, x_2, ..., x_n)$ 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 elements of the set $\{0, 1\}$ and incompletely specified if any of its outputs is a nonspecified one.

Definition 1: An n-bit string is a vertex of an object called a 0-cube. An n-variable boolean function is represented as a 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 [12, 31]. A cube can be represented by an n-string of symbols 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 terms of ON, OFF and DC minterms, respectively. An *ON* array of cubes of a boolean function F, denoted by *ON(F)*, is defined as a set of cubes for which $F = 1$, an *OFF* array, denoted by *OFF(F)*, is a set of cubes for which $F = 0$, and a *DC* array, denoted by *DC(F)*, is a set of cubes for which $F = -$.

Definition 2: Two cubes are disjoint if they do not have any minterm in common. Otherwise, when they share some minterms, they are nondisjoint.

Definition 3: A nonnormalised Haar transform is defined in term of the nonnormalised Haar functions, which are represented as rows of a nonnormalised Haar matrix, T_N of order $N = 2^n$ [1, 7, 8]

$$T_N = \begin{bmatrix} T_{\frac{N}{2}} \otimes [1 \quad 1] \\ I_{\frac{N}{2}} \otimes [1 \quad -1] \end{bmatrix} \text{ and } T_1 = 1 \qquad (1)$$

where $I_{N/2}$ is an identity matrix of order $N/2$ and the symbol $\otimes$ denotes the right-hand Kronecker product. It can be easily seen that the nonnormalised Haar matrix so defined is not orthogonal since $T_N T_N^T \neq kI_N$ where k is a constant. Hence to recover a boolean function from its nonnormalised Haar spectrum, various methods have been developed [24, 25, 27–29].

For an n-variable boolean function $F(x_1, x_2, ..., x_n)$ the Haar spectrum is given by $R = [T_N] F$ where R is the Haar spectrum (a column vector of dimension $2^n \times 1$) and F is the R-coded truth vector of the boolean function $F(X)$ [3, 12, 35]. In R coding, the false minterms are coded as 0, true minterms as 1 and don't care (DC) minterms as 0.5.

Besides the first two Haar spectral coefficients r_{dc} (so called *dc* coefficient corresponding to *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. A spectral coefficient $r_l^{(k)}$ is characterised by its degree l and order k.

Definition 4: A standard trivial function (STF), denoted by u_I, $I \in \{1, 2, ..., 2^n - 1\}$, associated with each Haar spectral coefficient r_{dc} or $r_l^{(k)}$ describes some set of $(n - l)$ cubes that

has an influence on the value of a spectral coefficient r_{dc} or $r_l^{(k)}$ where $0 \le l \le n-1$ and $0 \le k \le 2^l - 1$. 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

$$u_0 = u_1 = 1 \text{ and } u_I = \prod_{r=1}^{l} x_{n-l+r}^{k_r}$$
$$\text{for } 1 \le l \le n-1 \text{ and } 0 \le k \le 2^l - 1 \tag{2}$$

where k_r is the rth bit of bin(k). The operation bin(k) gives the binary representation of a natural number k.

Property 1: For a Haar spectrum of an n-variable boolean function F, there are 2^l spectral coefficients of degree l, each gives a correlation between the boolean function and a standard trivial function corresponding to the coefficient. The value of r_{dc} is equal to the number of minterms of F and the coefficient $r_0^{(0)}$ describes the difference between the number of minterms in the functions $\bar{x}_n$ and x_n.

Property 2: The degree l of Haar coefficient indicates the number of literals present in a STF u_I for $I = 1, 2, \ldots, 2^n - 1$.

Property 3: The order k of Haar spectral coefficient $r_l^{(k)}$ is the decimal equivalence of the binary l-tuple formed by writing a 1 or 0 for each variable in a STF u_I ($I = 2, 3, \ldots, 2^n - 1$) according to whether this literal appears in affirmation or negation. When k is expressed as a binary l-tuple, the most significant bit (MSB) corresponds to the literal $\dot{x}_n$ and the least significant bit (LSB) corresponds to the literal $\dot{x}_{n-l+1}$.

For each index I of a STF u_I, there exist unique values l and k such that $I = 2^l + k$.

Recently, a paired Haar transform has been introduced [24–28] to efficiently allocate don't-care minterms in the optimisation of free binary decision diagrams (FBDDs) and multiplexer synthesis of incompletely specified boolean functions.

Definition 5: A paired Haar transform (PHT) for an incompletely specified n-variable boolean function F is a mapping χ: $(F_{ON}, F_{DC}) \rightarrow (R_{ON}, R_{DC})$, 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 nonnormalised Haar transform. 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 nonnormalised Haar spectrum is related to the paired Haar spectrum as follows:

$$R = R_{ON} + 0.5 \times R_{DC} \tag{3}$$

Example 1: For the four-variable incompletely specified boolean function $F(X) = \Sigma_{ON}(8, 9, 10, 14, 15) + \Sigma_{DC}(1, 4, 5)$, the paired Haar spectrum $(R_{ON}, R_{DC}) = [((r_{ON})_{dc}, (r_{DC})_{dc})\ ((r_{ON})_0^{(0)}, (r_{DC})_0^{(0)})\ ((r_{ON})_1^{(0)}, (r_{DC})_1^{(0)}) \ldots ((r_{ON})_3^{(7)}, (r_{DC})_3^{(7)})]^T = [(5, 3)\ (-5, 3)\ (0, -1)\ (1, 0)\ (0, 1)\ (0, 2)\ (1, 0)\ (-2, 0)\ (0, -1)\ (0, 0)\ (0, 0)\ (0, 0)\ (0, 0)\ (1, 0)\ (0, 0)\ (0, 0)]^T$.

3 Calculation of paired Haar spectrum for single output function

In this Section the boolean function from which the paired Haar spectrum is calculated is represented by an array of disjoint ON and DC cubes.

Definition 6: 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.

The following two properties are new and form the basis of our procedure to calculate paired Haar spectra of incompletely specified boolean functions.

Property 4: 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 5: 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

$$v = \begin{cases} 2^{p-q} & \text{if bin}(k) \subseteq \rho_i(C) \text{ and } x_i = \text{'0'} \\ -2^{p-q} & \text{if bin}(k) \subseteq \rho_i(C) \text{ and } x_i = \text{'1'} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $\rho_i(C)$ is the cube obtained by a logical right shift of the cube C by i bits, and q is the number of '–' in the cube $\rho_i(C)$, i.e. $q = \log_2|\rho_i(C)|$. The operation bin(k) gives the binary representation of a natural number k. $v = 0$ if x_i = '–' or the binary representation of the order k is not covered by the cube $\rho_i(C)$. A logical right shift by i bits means that the i least significant bits will be deleted and i zeros will be fed from the left to cover the new i most significant positions. From eqn. 2, the standard trivial function $u_I = \Pi_{r=1}^{l} x_{n-l+r}^{k_r}$. It follows that the value of the rth bit in bin(k) determines the polarity of the variable x_{n-l+r} in the intersection of the cube C and u_I. Hence, shifting the cube C with the logical right shift operation by $i = n - l$ positions reveals the number of minterms of the cube covered by the standard trivial function corresponding to the coefficient.

```
Procedure partial_coeft (Paired Haar spectrum PHS, cube C, degree l)
{
    order_list = {k ∈ N | bin(k) ⊆ ρ_{n-l}(C)};
    Initialise(PHS);
    for (each 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) = (r_ON)_l^(k) + v ;
      else if (C is a DC cube) (r_DC)_l^(k) = (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));
    }
}
```

Fig. 1 *Procedure for calculating partial spectrum contributed by a p-cube*

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 based on the foregoing properties is given in Fig. 1. In Fig. 1, *order_list* is an array of integers representing the minterms covered by the cube $\rho_{n-l}(C)$. *PHS* is a link list of nonzero valued paired Haar spectral coefficients sorted in ascending order of degree l and order k. The routine **lookup** searches from *PHS* for any nonzero 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 $(r_{ON})_l^{(k)}$, $(r_{DC})_l^{(k)}$ 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 nonzero 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 4. By summing 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 Fig. 2 describes the procedure of calculating the complete paired Haar spectrum. In Figs. 1 and 2 the routine **Initialise** sets up the link list *PHS* and initialises all the Haar coefficients to 0. The number of partial spectral coefficients *npsc* is equal to the number of disjoint ON and DC cubes. To conserve disc space, it is sufficient to store only the nonvanishing 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.

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

Fig. 2 *Algorithm to calculate complete paired Haar spectrum*

Table 1: Calculation of paired Haar spectrum from 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 coefficient		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)

Example 2: An example for calculating the paired Haar spectrum by procedure **Paired_Haar** is shown in Table 1. 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 1. 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 sample calculation of one of the paired Haar spectral coefficients $((r_{ON})_1^{(0)}, (r_{DC})_1^{(0)})$ using procedure **partial_coeft** in Fig. 1 is demonstrated as follows:

Since $l = 1$, $n - l = 3$. For the ON cube $C_1 = $ —1–, $p = 3$, $\rho_{n-l}(C_1) = \rho_3($—1–$) = 000$–, $q = 1$. Since x_3 of C_1 is '–', $(r_{ON})_1^{(0)} = 0$. For the ON cube $C_2 = 110$–, $\rho_{n-l}(C_2) = \rho_3(110$–$) = 0001$. Since $k = 0 = 0000_2 \not\subset \rho_{n-l}(C_2) = 0001$, $(r_{ON})_1^{(0)} = 0$. For the ON cube $C_3 = 0$–01, $p = 1$, $\rho_{n-l}(C_3) = \rho_3(0$–$01) = 0000$, $q = 0$. Since x_3 of C_3 is '–', $(r_{ON})_1^{(0)} = 0$. For the DC cube $C_4 = 0000$, $\rho_{n-l}(C_4) = \rho_3(0000) = 0000$, $p = q = 0$. Since $k = 0 = 0000_2 \subseteq \rho_{n-l}(C_4) = 0000$ and x_3 of C_4 is '0', $(r_{DC})_1^{(0)} = 2^{0-0} = 1$. Hence, $((r_{ON})_1^{(0)}, (r_{DC})_1^{(0)}) = (0, 1)$.

4 Paired Haar spectrum for system of incompletely specified boolean functions

In this Section an efficient method for the calculation of paired Haar spectrum of a system of incompletely specified boolean functions that can have any number of functions and arbitrary locations of don't care minterms in each of the functions of the system is presented. Consider a system of t incompletely specified functions. By ordering the system of t functions to form a binary t-tuple $F_{t-1}\ F_{t-2}\ \ldots\ F_0$, where F_{t-1} is the MSB, a single multivalued output function F is obtained [1]. Furthermore, let F_{jON} be the truth vector obtained from F_j by replacing its don't care outputs by 1, and F_{jDC} be the truth vector obtained from F_j by replacing its true outputs by 0s and its don't care outputs by 1s. The functions F_{ON} and F_{DC} can be written as a weighted sum of each individual function F_j as follows:

$$F_{ON} = \sum_{j=0}^{t-1} 2^j F_{jON} \text{ and } F_{DC} = \sum_{j=0}^{t-1} 2^j F_{jDC} \quad (5)$$

Applying paired Haar transform to both sides of the expression,

$$R_{ON} = 2^j R_{jON} \text{ and } R_{DC} = 2^j R_{jDC} \quad (6)$$

where the tuples (R_{ON}, R_{DC}) and (R_{jON}, R_{jDC}) are the paired Haar spectra of the multiple output function F and its jth output F_j, respectively. The total spectrum (R_{ON}, R_{DC}) is called the ordered paired Haar spectrum since it is sensitive to the relative position of each output function within the system. Since the weighted sums F_{ON} and F_{DC} are formed from super increasing sequence, it is trivial to show that the ordered paired Haar spectrum obtained in this way is unique.

One way to calculate the ordered paired Haar spectrum (R_{ON}, R_{DC}) is by applying the algorithm **Paired_Haar** in Fig. 2 to each individual output F_j $(0 \le j \le t-1)$. Then, the t resulting paired Haar spectra (R_{jON}, R_{jDC}) are weighted by 2^j and summed according to eqn. 6. However, the ordered paired Haar spectrum obtained by direct application of eqn. 6 is very inefficient when the number of functions t is large as a large number of disjoint ON and DC cubes are duplicated in some arrays of F_j. Fortunately, the algorithm in [33] can generate a single array of near optimal number of disjoint cubes for a system of t functions. Each disjoint

cube C consists of an input part ($x_n\ x_{n-1}\ ...\ x_1$) and an output part ($y_{t-1}\ y_{t-2}\ ...\ y_0$), where the input variable $x_i = 0$, 1 or – ($1 \le i \le n$) depending on whether x_i appears as complemented or affirmative form or does not appear in the product term represented by C, and the output variable $y_j = 0$, 1 or – ($0 \le j \le t - 1$) depending on whether the cube represented by the input part of C is a OFF, ON or DC cube of the function F_j. Instead of operating on t arrays of disjoint ON and DC cubes separately, procedure **Paired_Haar** in Fig. 2 can be modified to produce the paired Haar spectrum for a system of incompletely specified functions by accepting a single array of disjoint cubes. The new procedure **Ordered_Paired_Haar** is given in Fig. 3. The global variable *ONweight* and *DCweight* accumulates the weight factors contributed by all ON and DC outputs of each cube C_j. Hence the partial coefficient is calculated once for each cube C_j as opposed to s times in the previous method if the input part of C_j appears s times in t arrays. The if and else if statements on lines 7 and 8 of procedure **partial_coeft** (Fig. 1) have to be modified to $(r_{ON})_l^{(k)} = (r_{ON})_l^{(k)} + v \times$ *ONweight* and $(r_{DC})_l^{(k)} = (r_{DC})_l^{(k)} + v \times$ *DCweight* accordingly. Procedure **Ordered_Paired_Haar** in Fig. 3 also allows independent calculation of either some selected coefficients or a complete degree of spectral coefficients by inclusion of additional arguments and removal of the inner for loops as described.

Example 3: Consider the system of incompletely specified functions F_0 and F_1 taken from examples 1 and 2, respectively. The ordered paired Haar spectrum is calculated by procedure **Ordered_Paired_Haar** in Table 2. The array of disjoint cubes describing the system of functions is given by {—1– 10, 110– 10, 0–01 1–, 111– 01, 10–0 01, 0000 –0, 0100 0–, 1001 01}. The input part $x_4\ x_3\ x_2\ x_1$ and the output part $y_1\ y_0$ of the cubes are listed in the first two rows of Table 2. The *ONweight* and *DCweight* are given, respectively, in the third and four rows under each cube. The partial coefficients contributed by each cube are given in columns 2 to 9 and are summed to give the total spectrum for the system of functions in the last column.

```
Procedure Ordered_Paired_Haar(Array of disjoint cubes D)
{
    Initialise(PHS);
    foreach (cube C_j ∈ D, j = 1 to npsc) {
        p = number of '–' in C_j ;
        ONweight = DCweight = 0 ;
        foreach (output variable y_i of C_j , i = 0 to t–1 ) {
            if ( y_i = 1) ONweight = ONweight + 2^i ;
            else if ( y_i = – ) DCweight = DCweight + 2^i ;
        }
        (r_ON)_dc = (r_ON)_dc + 2^p × ONweight;
        (r_DC)_dc = (r_DC)_dc + 2^p × DCweight;
        /* Note : Subroutine partial_coeft is modified by multiplying the partial value v by ONweight */
        /* for (r_ON)_l^(k) and DCweight for (r_DC)_l^(k) */
        for (l = 0 to n–1) partial_coeft (PHS, C_j, l);
    }
    return PHS ;
}
```

Fig. 3 *Algorithm for calculating paired Haar spectrum for system of functions*

Table 2: Calculation of ordered paired haar spectrum for system of boolean functions

$x_4x_3x_2x_1$	–-1–	110–	0–01	111–	10–0	0000	0100	1001	
y_1y_0	10	10	1–	01	01	–0	0–	01	Total spectrum
Onweight	2	2	2	1	1	0	0	1	
DCweight	0	0	1	0	0	2	1	0	
$((r_{ON})_{dc}, (r_{DC})_{dc})$	(16, 0)	(4, 0)	(4, 2)	(2, 0)	(2, 0)	(0, 2)	(0, 1)	(1, 0)	(29, 5)
$((r_{ON})_0^{(0)}, (r_{DC})_0^{(0)})$	(0, 0)	(–4, 0)	(4, 2)	(–2, 0)	(–2, 0)	(0, 2)	(0, 1)	(–1, 0)	(–5, 5)
$((r_{ON})_1^{(0)}, (r_{DC})_1^{(0)})$	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 2)	(0, –1)	(0, 0)	(0, 1)
$((r_{ON})_1^{(1)}, (r_{DC})_1^{(1)})$	(0, 0)	(–4, 0)	(0, 0)	(–2, 0)	(2, 0)	(0, 0)	(0, 0)	(1, 0)	(–3, 0)
$((r_{ON})_2^{(0)}, (r_{DC})_2^{(0)})$	(–4, 0)	(0, 0)	(2, 1)	(0, 0)	(0, 0)	(0, 2)	(0, 0)	(0, 0)	(–2, 3)
$((r_{ON})_2^{(1)}, (r_{DC})_2^{(1)})$	(–4, 0)	(0, 0)	(2, 1)	(0, 0)	(0, 0)	(0, 0)	(0, 1)	(0, 0)	(–2, 2)
$((r_{ON})_2^{(2)}, (r_{DC})_2^{(2)})$	(–4, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(1, 0)	(–3, 0)
$((r_{ON})_2^{(3)}, (r_{DC})_2^{(3)})$	(–4, 0)	(4, 0)	(0, 0)	(–2, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(–2, 0)
$((r_{ON})_3^{(0)}, (r_{DC})_3^{(0)})$	(0, 0)	(0, 0)	(–2, –1)	(0, 0)	(0, 0)	(0, 2)	(0, 0)	(0, 0)	(–2, 1)
$((r_{ON})_3^{(1)}, (r_{DC})_3^{(1)})$	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
$((r_{ON})_3^{(2)}, (r_{DC})_3^{(2)})$	(0, 0)	(0, 0)	(–2, –1)	(0, 0)	(0, 0)	(0, 0)	(0, 1)	(0, 0)	(–2, 0)
$((r_{ON})_3^{(3)}, (r_{DC})_3^{(3)})$	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
$((r_{ON})_3^{(4)}, (r_{DC})_3^{(4)})$	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(1, 0)	(0, 0)	(0, 0)	(–1, 0)	(0, 0)
$((r_{ON})_3^{(5)}, (r_{DC})_3^{(5)})$	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(1, 0)	(0, 0)	(0, 0)	(0, 0)	(1, 0)
$((r_{ON})_3^{(6)}, (r_{DC})_3^{(6)})$	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
$((r_{ON})_3^{(7)}, (r_{DC})_3^{(7)})$	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)

Table 3: Experimental results

Function	Inputs	Outputs	Disjoint	Coefficients	Time (OPH)	Time (FHT)	Size (SBDD)	Time (SBDD)
9sym	9	1	145	211	0.01	0.03	33	50
Z9sym	9	1	185	211	0.03	0.01	–	–
5xp1	7	10	75	128	0.02	0.02	88	140
Z5xp1	7	10	128	128	0.04	0.02	–	–
alu4	14	8	1043	12008	0.07	0.07	1352	2420
sao2	10	4	96	102	0.02	0.03	154	220
apex4	9	19	523	511	0.04	0.02	1021	1800
bw	5	28	106	29	0.03	0.04	138	190
clip	9	5	176	504	0.02	0.02	254	370
con1	7	2	11	85	0.03	0.03	18	20
inc	7	9	33	128	0.01	0.04	–	–
misex1	8	7	32	232	0.03	0.02	47	70
misex3	14	14	1641	3168	0.13	0.08	1301	2290
misex3c	14	14	2630	4100	0.14	0.08	1275	2150
table3	14	14	179	8992	0.02	0.06	941	1540
table5	17	15	166	78011	0.13	0.31	–	–
sqrt8	8	4	40	255	0.02	0.04	42	50
t481	16	1	887	28231	0.08	0.15	32	50
b12	15	9	58	28872	0.06	0.14	–	–
ex1010	10	10	1017	1021	0.05	0.04	–	–
squar5	5	8	32	32	0.02	0.03	–	–
xor5	5	1	16	17	0.03	0.02	9	20
rd53	5	3	32	32	0.02	0.02	23	30
rd73	7	3	141	128	0.04	0.04	–	–
rd84	8	4	256	256	0.02	0.06	59	80

'–' indicates data not available

5 Experimental results

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 [34] are given in Table 3. The MCNC benchmark functions in PLA format are preprocessed by the disjoint cube algorithm [33, 35] before the test. The number of disjoint cubes is given in the fourth column labelled '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 labelled 'coefficients' is the number of nonvanishing paired Haar coefficients and the column labelled 'Time (OPH)' is the system execution time in seconds on a HP Apollo Series 735 workstation. For comparison, the computation of paired Haar spectra for the same set of benchmark functions by the fast Haar transform method [3, 7, 8, 11, 13, 14] is implemented and run on the same workstation. The execution times in seconds for the fast Haar transform algorithm are shown in the column labelled 'Time (FHT)'. As MCNC benchmark functions have to be converted into full minterm vector before application of the fast Haar transform, preprocessing to generate true and don't care minterms is done but this computation time is not included in the column 'Time (FHT)'. Although the execution times of both methods are comparable, our method requires considerably less memory than the fast Haar transform method, particularly when the number of disjoint cubes is small. Besides the difference in the dimensions of the inputs (i.e. disjoint cubes against minterms) in the algorithms, an amount of storage proportional to 2^n where n is the number of input variables is required for the intermediate and final results using the fast Haar transform method. By contrast, the memory for the spectral coefficients calculated by our algorithm can be dynamically allocated and freed when necessary during the computational process.

Recently, an algorithm to compute Haar spectrum by symbolic methods was presented in [18]. The same algorithm was modified to calculate the paired Haar spectrum and the results obtained from R.S. Stankovic are added as columns labelled 'Size (SBDD)' and 'Time (SBDD)'. The columns 'Size (SBDD)' and 'Time (SBDD)' are the sizes of the shared binary decision diagrams (SBDDS) for the paired Haar spectra and the execution time in seconds for the computation of paired Haar spectra from shared binary decision diagrams of the benchmark functions, respectively. The calculations are performed on a 133MHz pentium PC with 32Mbyte of RAM. In SBDD, two bytes are required for each node to store the calculated spectral coefficient values while each nonvanishing paired Haar coefficient in our implementation requires one integer storage for the index and two integers storage for the values. Moreover, each node in SBDD has, according to R.S. Stankovic [private communication], the following data structure:

```
struct node = record;
    low, high: pointer to node;
    index: 1 ... n+1;
    left, right: integer;
    id: integer;
end;
```

Hence, from the results it may be concluded that the SBDD-based algorithm has similar space complexity as

ours. However, it does not allow direct computation of selected coefficients as easily as ours and also requires preprocessing of original benchmark functions by converting them into reduced decision diagrams. As with other results presented the results for 'Time (SBDD)' do not include this preprocessing time.

6 Conclusion

A new algorithm that generates paired Haar spectrum for system of incompletely specified boolean functions from the disjoint cube representation has been shown. Since the number of such cubes can be considerably smaller than the number of minterms, the memory requirement can be reduced significantly. The advantages of this kind of representation used frequently in modern CAD VLSI systems [9, 10, 12], especially that for practical functions the number of disjoint cubes is much smaller than the number of minterms, has been manifested in [32]. Recently, the concept of entropy and equivocation is formulated through paired Haar spectra of incompletely specified boolean functions and applied to quasiminimisation of free binary decision diagrams [24, 25]. The ability to calculate only some spectral coefficients made possible by this research is very important since there are many spectral methods in digital logic design for which the values of only selected spectral coefficients are needed [1, 3].

The fundamental advantage of the algorithm is the usage of a reduced representation of boolean functions in the form of disjoint cubes as the internal data from which the algorithm calculates the spectra. Such an approach gives the algorithm the ability to yield solutions to problems of very high dimensions and is applicable to these CAD systems which use cubical representation for discrete functions. The algorithm is very well suited for systolic VLSI realisations, and may be implemented as a hardware coprocessor in a manner similar to those used for other binary expansions [2].

7 Acknowledgment

The authors wish to thank the referees for their numerous helpful comments.

8 References

1 KARPOVSKY, M.G.: 'Finite orthogonal series in the design of digital devices' (John Wiley, New York, 1976)
2 KUKHAREV, G.A., SHMERKO, V.P., and YANUSHKEVICH, S.N.: 'Technique of binary data parallel processing for VLSI' (University Press, Minsk, 1991) (in Russian)
3 KULESZA, W.: 'Systems of spectral analysis of digital data' (WKL, Warsaw, 1984) (in Polish)
4 RUIZ, G., BURON, A., and MICHELL, J.A.: 'NMOS and CMOS physical faults diagnosis by spectral techniques: simulation and associated tools'. Proceedings of IEEE workshop on *Languages for automation*, Vienna, Austria, Aug. 1987, pp. 191–194
5 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* MORAGA, C. (Ed.): 'Theory and applications of spectral techniques' (University Dortmund Press, 1988), pp. 97–106
6 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.*, 1992, **139**, (4), pp. 293–307
7 STOJIC, M.R., STANKOVIC, M.S., and STANKOVIC, R.S.: 'Discrete transforms and their applications' (Beograd, Nanka, 1993) (in Serbian)
8 ZALMANZON, L.A.: 'Fourier, Walsh and Haar Transforms and their application in control, communication and other fields' (Nauka, Moscow, 1989) (in Russian)
9 ZHINTELIS, G.B., KARCHIAUSKAS, E.K., and MACHIKENAS, E.K.: 'Automation of designing microprogrammable structures' (Mashinostrojeniye, Leningrad, 1985) (in Russian)
10 NEWTON, A.R. (Ed.): 'Selected papers on logic synthesis for integrated circuit design' (IEEE Press, New York, 1987)
11 AHMED, N., and RAO, K.R.: 'Orthogonal transforms for digital signal processing' (Springer-Verlag, Berlin, 1975)
12 KINOSITA, K., ASADA, K., and KARAZU, O.: 'Logic design for VLSI' (Iwanami Shoten, Tokyo, 1985) (in Japanese)
13 YAROSLAVSKY, L.P.: 'Digital picture processing' (Springer-Verlag, Berlin, 1985)
14 BURON, A.M., MICHELL, J.A., and SOLANA, J.M.: 'Single chip fast Haar transform at megahertz rates' *in* MORAGA, C. (Ed.): 'Theory and applications of spectral techniques' (University Dortmund Press, 1988), pp. 8–17
15 ROESER, P.R., and JERNIGAN, M.E.: 'Fast Haar transform algorithms', *IEEE Trans. Comput.*, 1982, **31**, (2), pp. 175–177
16 SHORE, J.E.: 'On the applications of Haar functions', *IEEE Trans.*, 1973, **COM–21**, pp. 206–216
17 STANKOVIC, R.S., and FALKOWSKI, B.J.: 'Haar functions and transforms and their generalisations'. Proceedings of the 1st IEEE international conference on *Information, communications and signal processing* (ICICS'97), Singapore, Sep. 1997, Vol. 4, pp. 1–5
18 STANKOVIC, M., JANKOVIC, D., and STANKOVIC, R.S.: 'Efficient algorithm for Haar spectrum calculation'. Proceedings of 1st IEEE international conference on *Information, communications and signal processing* (ICICS'97), Singapore, Sep. 1997, Vol. 4, pp. 6–10
19 PERKOWSKI, M.A.: 'A new representation of strongly unspecified switching functions and its application to multi-level AND/OR/EXOR synthesis'. Proceedings of IFIP WG10.5 workshop on *Applications of the Reed-Muller expansion in circuit design* (Reed-Muller'95), Makuhari, Chiba, Japan, Sep. 1995, pp. 143–151
20 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 IFIP WG10.5 workshop on *Applications of the Reed-Muller expansion in circuit design* (Reed-Muller'95), Makuhari, Chiba, Japan, Sep. 1995, pp. 102–109
21 PERKOWSKI, M.A., RYDZEWSKI, A., and MISIUREWICZ, P.: 'Theory of logic circuits' (Technical University of Warsaw Publisher, Warsaw, 1977) (in Polish)
22 ZAKREVSKIJ, A.: 'Search space reducing: a super-fast algorithm for minimum AND/EXOR implementation of systems of weakly specified Boolean functions'. Proceedings of 4th international conference on *Pattern recognition and information processing* (PRIP'97), Minsk, Belarus, 1997, Vol. 1, pp. 327–331
23 ZAKREVSKIJ, A., and TOROPOV, N.: 'Optimizing polynomial implementation of incompletely specified Boolean functions'. Proceedings of international conference on *Computer aided design*, Minsk, Belarus, 1995, Vol. 1, pp. 93–98
24 CHANG, C.H., and FALKOWSKI, B.J.: 'Generation of quasi-optimal FBDDs through paired Haar spectra'. Proceedings of 31st IEEE international symposium on *Circuits and systems* (ISCAS'98), Monterey, CA, USA, Jun. 1998, Vol. VI, pp. 167–170
25 CHANG, C.H., and FALKOWSKI, B.J.: 'Haar spectra based entropy approach to quasi-minimization of FBDDs', *IEE Proc. - Comput. Digit. Tech.*, 1999, **146**, (1), pp. 41–49
26 FALKOWSKI, B.J., and CHANG, C.H.: 'A novel paired Haar based transform: algorithms and interpretations in Boolean domain'. Proceedings of 36th IEEE midwest symposium on *Circuits and systems* (MWSCAS'93), Detroit, MI, Aug. 1993, pp. 1101–1104
27 FALKOWSKI, B.J., and CHANG, C.H.: 'Properties and applications of paired Haar transform'. Proceedings of 1st IEEE international conference on *Information, communications and signal processing* (ICICS'97), Singapore, Sep. 1997, Vol. 1, pp. 48–51
28 FALKOWSKI, B.J., and CHANG, C.H.: 'Calculation of paired Haar spectra for systems of incompletely specified Boolean functions'. Proceedings of 31st IEEE international symposium on *Circuits and systems* (ISCAS'98), Monterey, CA, USA, Jun. 1998, Vol. VI, pp. 171–174
29 FALKOWSKI, B.J., and CHANG, C.H.: 'Forward and inverse transformations between Haar spectra and ordered binary decision diagrams of Boolean functions', *IEEE Trans. Comput.*, 1997, **46**, (11), pp. 1272–1279
30 HANSEN, P., and SEKINE, M.: 'Decision diagram based techniques for the Haar wavelet transform'. Proceedings of 1st international conference on *Information, communications and signal processing* (ICICS'97), Singapore, Sep. 1997, Vol. 1, pp. 59–63
31 DE MICHELI, G.: 'Synthesis and optimisation of digital circuits' (McGraw-Hill, New York, 1994)
32 CIESIELSKI, M.J., YANG, S., and PERKOWSKI, M.A.: 'Multiple-valued minimization based on graph coloring'. Proceedings of IEEE international conference on *Computer design: VLSI in computers and processors*, Oct. 1989, pp. 262–265
33 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* (MWSCAS'93), Detroit, MI, Aug. 1993, pp. 1308–1311
34 YANG, S.: 'Logic synthesis and optimization benchmarks user guide' (Microelectronics Centre of North Carolina, Jan. 1991)
35 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. Comput.-Aided Des. Integr. Circuits Syst.*, 1992, **11**, (10), pp. 1207–1226