PERGAMON

Computers and Electrical Engineering 29 (2003) 13–24

Computers and Electrical Engineering

www.elsevier.com/locate/compeleceng

Calculation of the paired Haar transform through shared binary decision diagrams

Milena Stanković [a], Bogdan J. Falkowski [b,*], Dragan Janković [a], Radomir S. Stanković [a]

[a] Department of Computer Science, Faculty of Electronics, 18000 Niš, Yugoslavia

[b] School of Electrical and Electronic Engineering, Nanyang Technological University, Block S1, Nanyang Avenue, Singapore 639798, Singapore

Received 24 October 2000; accepted 8 January 2001

Abstract

A concept of paired Haar transform (PHT) for representation and efficient optimization of systems of incompletely Boolean functions has recently been introduced. In this article, a method to calculate PHT for incompletely specified switching functions through shared binary decision diagrams (SBDDs) is presented. The algorithm converts switching functions in the form of SBDDs into their paired Haar spectra and can operate on functions with many variables.

Keywords: Discrete transform; Haar transform; Paired transform; Decision diagrams; Switching functions

1. Introduction

Discrete Haar transform is the simplest example of wavelet expansion and attracts much attention in engineering practice for its peculiar properties [2,3,7–18,27–32].

The advantages of computational and memory requirements of the Haar transform make it of a big interest to VLSI designers. For example, the authors of [21,22] present 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 diagnosis 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

* Corresponding author. Tel.: +65-790-4521; fax: +65-791-2687.
E-mail address: efalkowski@ntu.edu.sg (B.J. Falkowski).

PII: S0045-7906(01)00022-2

to derive strategies for testing MOS circuits when memory states were encountered as a consequence of some fault types. Testing through spectra analysis is to look for a determined input/output behavior for some input sequence but not observing the output values directly but rather their spectrum, and more precisely some selected coefficients. This has a potential advantage that the amount of information in which to look for some error is severely diminished at the expense of computing the spectrum coefficients, something that is a routine computation and in some cases could be worth to do it in hardware as "built-in test". 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 [17,18,32]. For example, the analysis in Ref. [17] 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 [9,18] to generate arbitrary Boolean functions consists of a generator of basis functions, an adder, a multiplier, and the memory content. Such a behavior of the module is useful in real time adaptive control systems [18,32]. Karpovsky [17] 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 [17]. 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 [5,19,20]. These functions are efficiently represented by the arrays of ON and OFF terms, since a majority of their functional domain are do not 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 transform, which is a global transform, would be quite cumbersome in such cases, since 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 (PHT) was introduced for incompletely specified switching functions [11]. Intended applications of PHT concern functions with large number of variables. In PHT, all the information about true and do not 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 free binary decision diagrams (FBDDs) and multiplexer synthesis for incompletely specified Boolean functions, have been demonstrated in Refs. [4,10–13]. A 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 [4]. For FBDD and ordered binary decision diagram (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.

In view of the above applications for PHT, it is necessary to provide efficient calculation methods for PHT through reduced representations of switching functions such as disjoint cubes [13] or decision diagrams [4,9,23]. In this article, we extend the method presented in [25,26] to multi-output switching functions that are represented by shared binary decision diagrams (SBDDs) having a separate root node for each output [23]. We use this method to calculate PHT by taking advantages of peculiar properties of PHT [9,12].

2. Basic concepts

2.1. Switching functions

Let $B = \{0, 1\}$. The mapping f: $B^n \to B$ is an n-variable switching function. If the output of f for a combination of logic values for input variables is not specified, then f is an incompletely specified switching function. Thus, an n-variable incompletely specified switching function is the mapping f: $B^n \to B \cup \{*\}$, where $*$ denotes a non-specified value (do not care).

Definition 1 ([7,10]). A pair of functions $(f_{\mathrm{ON}}, f_{\mathrm{DC}})$ is assigned to each incompletely specified switching function f. f_{ON} is defined as a function obtained from f by replacing all the do not care outputs of f by 0. f_{DC} is obtained by replacing all the true outputs by 0 and do not care outputs by 1.

2.2. Binary decision diagrams

Binary decision diagrams (BDDs) are a data structure convenient to represent switching functions of a large number of variables [23]. BDDs are derived by the reduction of binary decision trees (BDTs).

Definition 2. BDT representing a switching function f is a rooted acyclic diagram $D = (V, E)$ with the edge set E and the node set V consisting of the root node, non-terminal nodes, and constant nodes. A variable x_i is assigned to each non-terminal node $v \in V$ and is called the decision variable for v. All the nodes assigned to the same variable x_i form the ith level in the BDT. The constant nodes represent the values of f.

BDTs for switching functions of a given number of variables can be derived by the recursive application of the Shannon decomposition rule $f = \bar{x}_i f_0 \oplus x_i f_1$, where $f_0 = f(x_i = 0)$ and $f_1 = f(x_i = 1)$ to all the variables in f. In a BDT, the values of constant nodes are logic values 0 and 1. Multi-terminal BDTs (MTBDTs) are defined by allowing integers or complex numbers as the values of constant nodes [6]. Multi-terminal BDDs (MTBDDs) are derived by the reduction of MTBDTs [6].

Multiple-output switching functions are represented by shared BDDs (SBDDs) or shared MTBDDs [23] having a separate root node for each output. Thus, SBDDs are obtained by sharing isomorphic subtrees in BDDs for outputs of f, considered as separate particular switching functions.

Example 1. Consider a two-output incompletely specified switching function $f = f_1 \star f_0$ given by the truth-vectors

$$\mathbf{F}_1 = [*, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1]^{\mathrm{T}},$$
$$\mathbf{F}_0 = [0, *, 0, 0, *, *, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1]^{\mathrm{T}}.$$

Thus, f_{ON} and f_{DC} associated to this function are given by the vectors

$$\mathbf{F}_{\mathrm{ON}_1} = [0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1]^{\mathrm{T}},$$
$$\mathbf{F}_{\mathrm{DC}_1} = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^{\mathrm{T}},$$
$$\mathbf{F}_{\mathrm{ON}_0} = [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1]^{\mathrm{T}},$$
$$\mathbf{F}_{\mathrm{DC}_0} = [0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^{\mathrm{T}}.$$

Example 2. Fig. 1 shows BDT for the function f_1 given by the truth-vector $\mathbf{F}_1$ in Example 1. Fig. 2 shows SBDD for f in this example.

As shown in the above example, a SBDDs with two root nodes can be used to represent the functions f_{ON}, and f_{DC} associated to an incompletely specified switching function f. A SBDD with m pairs of the root nodes can be used to represent f_{ON} and f_{DC} functions assigned to an m-output switching function f.

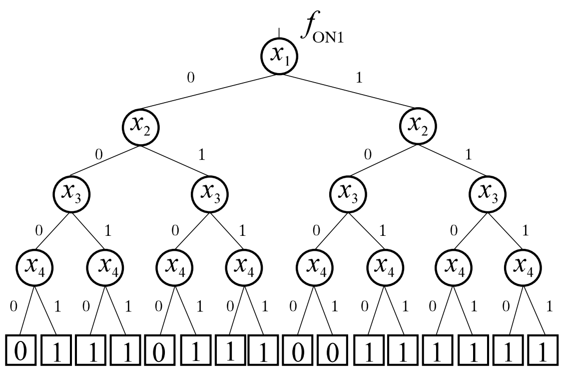

Fig. 1. BDT for f_1 in Example 1.

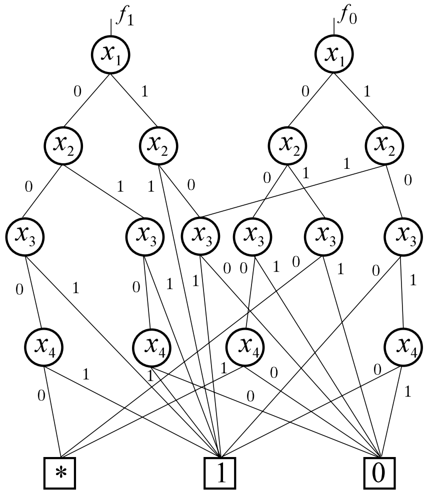

Fig. 2. SBDD for f in Example 1.

3. Discrete Haar transform

Discrete Haar transform is defined in terms of the discrete Haar functions, which are conveniently represented as rows of an $(2^n \times 2^n)$, $n \in N$, matrix $\mathbf{T}(n)$ denoted as the Haar matrix [16,17,29,32].

In this paper, we consider the non-normalized Haar transform defined as follows.

Definition 3. Discrete Haar functions of order n represented by $(2^n \times 2^n)$ matrix $\mathbf{T}(n)$, in the sequency ordering are given by the following recurrence relation:

$$\mathbf{T}(n) = \begin{bmatrix} \mathbf{T}(n-1) \otimes [1 \quad 1] \\ \mathbf{I}_{(n-1)} \otimes [1 \quad -1] \end{bmatrix},$$

where $\otimes$ denotes the Kronecker product,

$$\mathbf{T}(1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

and $\mathbf{I}_q$ is the identity matrix of order q.

Definition 4. For a switching function f represented by the truth-vector $\mathbf{F}(n) = [f(0), \ldots, f(2^n - 1)]^{\mathrm{T}}$, the Haar spectrum $\mathbf{Y}_f(n) = [Y(0), \ldots, Y(2^n - 1)]^{\mathrm{T}}$ is given by:

$$\mathbf{Y}_f(n) = \mathbf{T}(n)\mathbf{F}(n), \tag{1}$$

$$\mathbf{F}(n) = \mathbf{T}(n)^{-1}\mathbf{Y}_f(n), \tag{2}$$

where $\mathbf{T}(n)$ is the Haar matrix in the corresponding ordering and $\mathbf{T}^{-1}(n)$ is its inverse over the complex field C. The inverse is equal to the $\mathbf{T}^{\mathrm{T}}(n)$, where $\mathbf{T}^{\mathrm{T}}$ denotes the transpose of $\mathbf{T}$.

3.1. Paired Haar transform

Definition 5 ([7,10]). A PHT for an incompletely specified n-variable switching function f is a mapping χ: $(f_{\mathrm{ON}}, f_{\mathrm{DC}}) \rightarrow (\mathbf{R}_{\mathrm{ON}}, \mathbf{R}_{\mathrm{DC}})$, where $\mathbf{R}_{\mathrm{ON}} = \mathbf{TF}_{\mathrm{ON}}$, and $\mathbf{R}_{\mathrm{DC}} = \mathbf{TF}_{\mathrm{DC}}$. The pair $(\mathbf{R}_{\mathrm{ON}}, \mathbf{R}_{\mathrm{DC}})$ is the PHT-spectrum for f.

Example 3. PHT-spectrum for f in Example 1 is given by the vectors

$$(\mathbf{R}_{\mathrm{ON}_1}, \mathbf{R}_{\mathrm{DC}_1})_1 = [(12,1),(0,1),(0,1),(-2,0),(-1,1),(-1,0),(-2,0),\\ (0,0),(-1,1),(0,0),(-1,0),(0,0),(0,0),(0,0),(0,0),(0,0)]^{\mathrm{T}},$$

$$(\mathbf{R}_{\mathrm{ON}_0}, \mathbf{R}_{\mathrm{DC}_0})_1 = [(5,3),(-5,3),(0,-1),(1,0),(0,1),(0,2),(1,0),(-2,0),(0,-1),\\ (0,0),(0,0),(0,0),(0,0),(1,0),(0,0),(0,0)]^{\mathrm{T}}.$$

Fig. 3 shows SBDD for the PHT-spectrum for the multiple-output function f in Example 1.

Definition of PHT proved useful in solving some problems in logic design, as for example, minimization of mixed polarity Reed–Muller expressions, generation of quasi-optimal FBDDs, and multiplexer synthesis [3,10,11].

4. Calculation of PHT through SBDD

Calculation procedure for PHT-spectrum is derived as a modification of the procedure for calculation of the Haar spectrum of integer-valued or complex-valued functions in Ref. [25], see also Ref. [26]. The PHT-spectrum is calculated by processing the nodes of SBDD. The PHT-spectrum is stored at the SBDD of f in the 2 bytes fields assigned to each node.

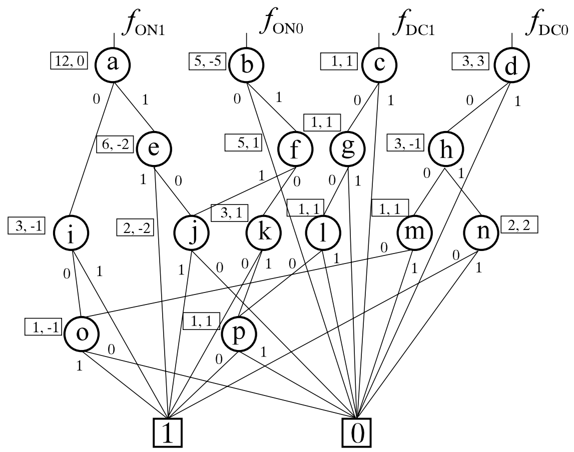

Fig. 3. SBDD for PHT-spectrum for f in Example 1.

In a BDD, each node is related to two subtrees. These two subtrees are rooted at the nodes to which point the outgoing edges of the considered node. In the matrix notation, these subtrees represent some subvectors in the truth-vector for f.

A discussion of the structure of the Haar matrix and properties of the Haar transform, permits derivation of some recurrence relations useful in calculation of the Haar spectrum [25,26]. These relations are performed at each node in the MTBDD or BDD by starting from the nodes assigned to the variable x_n. The inputs are the values of constant nodes. These relations reduce calculations to the operations over first elements in the subvectors represented by the subtrees related to the processed node. The values calculated by processing the node are stored in a 2 bytes field. The left part of this field is the value of a Haar transform coefficient. The other part is the input in the processing of the nodes at the upper level in the DD. In this way, the Haar spectrum is stored in the fields assigned to the nodes of the MTBDD or BDD representing f whose transform is calculated. We extend this method to SBDDs, and use it to calculate the PHT-spectra of incompletely specified multiple-output switching functions.

It is assumed that in the SBDD for a given function $f(x_1, \ldots, x_n)$, the variable x_1 is assigned to the root node. The other variables are assigned to the levels in the SBDD in the increasing order. Each node in the SBDD is represented by the following data structure

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

Fig. 4 shows the procedure *CalcTransform* used to calculate the PHT-spectrum. It calls a procedure *CalcNode* that processes the nodes in the SBDDs. Therefore, the calculation procedure for PHT-spectrum consists of the following steps:

1. Generate SBDD for functions F_{ON} and F_{DC}.
2. Apply the procedure CalcTransform to the nodes in the SBDD.
3. Read the spectrum from the fields assigned to the nodes in the SBDD and prepare the output file containing the spectrum.

The PHT spectrum is read by using standard procedures for traversing a DT. Inorder traversal and level order traversal procedure are used to read PHT spectrum in natural and sequency ordering, respectively [24]. Inorder traversal is illustrated in Fig. 5 for reading the PHT-spectrum for $n = 3$.

The method permits to read particular spectral coefficients independently of other coefficients, the feature that is important in many applications of the Haar spectra [3,4,10,12,17,21,22].

Since the algorithm for calculation of PHT-spectrum consists of processing of nodes in the SBDDs, its complexity in terms of both space and time is proportional to O(s), where s is the size of the SBDD. The space overhead reduces to the 2 byte field assigned to each node.

Procedure CalcNode(u:node)

begin

$DiffLevel0 = u.index - u.low.index - 1$

$DiffLevel1 = u.index - u.high.index - 1$

$u.left = 2^{DiffLevel0} * u.low.left + 2^{DiffLevel1} * u.high.left$

$u.right = 2^{DiffLevel0} * u.low.left - 2^{DiffLevel1} * u.high.left$

end

procedure CalcTransform()

begin

for $i = n, 1, -1$

for each node u at the i-th level

Calc-node(u)

end

Fig. 4. Procedure for calculation of PHT-spectrum.

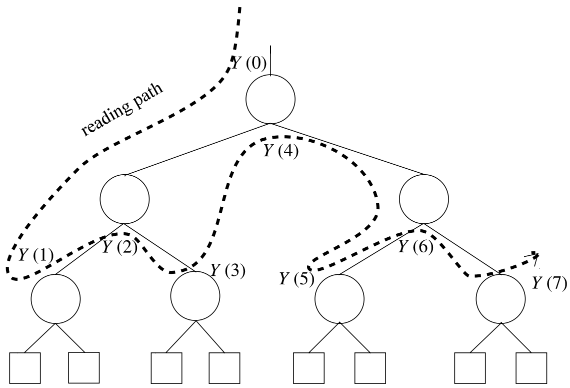

Fig. 5. Inorder traversal of a BDT.

Example 4. Fig. 3 shows SBDD for functions f_{ON} and f_{DC} assigned to f in Example 1. The result of the processing nodes is shown in the fields assigned to the nodes. We descend the SBDD and read the PHT-spectrum for f. Table 1 explains the calculations performed at the nodes of SBDD.

5. Experimental results

Table 2 shows complexity of SBDDs for some mcnc benchmark functions and CPU-times for calculation of the PHT. The number of inputs (i), outputs (o), and size (s) of SBDDs are shown. Calculation time is given in milliseconds. Calculations are performed on a 133 MHz Pentium PC

Table 1
Calculations at the nodes in SBDD

Node	Field	
	Left	Right
a	$2\times 3+6=12$	$2\times(-1)-(-2)=0$
b	$8\times 0+5=5$	$8\times 0-5=-5$
c	$1+8\times 0=1$	$1-8\times 0=1$
d	$3+8\times 0=3$	$3-8\times 0=3$
e	$2+4\times 1=6$	$2-4\times 1=-2$
f	$3+2=5$	$3-2=1$
g	$1+4\times 0=1$	$1-4\times 0=1$
h	$1+2=3$	$1-2=-1$
i	$1+2\times 1=3$	$1-2\times 1=-1$
j	$2\times 0+2\times 1=2$	$2\times 0-2\times 1=-2$
k	$2\times 1+1=3$	$2\times 1-1=1$
l	$1+2\times 0=1$	$1-2\times 0=1$
m	$1+2\times 0=1$	$1-2\times 0=1$
n	$2\times 1+2\times 0=2$	$2\times 1-2\times 0=2$
o	$0+1=1$	$0-1=-1$
p	$1+0=1$	$1-0=1$

Table 2
Complexity of *f* and CPU-times

f	*i*	*o*	*s*	*d*	*t* (ms)	*t-cubes* (ms)
9sym	9	1	33	145	0.05	10
5xp1	7	10	88	75	0.14	20
alu4	14	8	1352	1043	2.42	70
sao2	10	4	154	96	0.22	20
apex4	9	19	1021	523	1.8	40
bw	5	28	138	106	0.19	30
clip	9	5	254	176	0.37	20
con1	7	2	18	11	0.02	30
misex1	8	7	47	32	0.07	30
misex3	14	14	1301	1641	2.29	130
misex3c	14	14	1275	2630	2.15	140
xor5	5	1	9	16	0.02	20
rd53	5	3	23	32	0.03	20
rd84	8	4	59	256	0.08	20
sqrt8	8	4	42	40	0.05	20
t481	16	1	32	887	0.05	80
table3	14	14	941	179	1.54	20

with 32 Mbytes of RAM. In these experiments, we used SBDD package developed by us. However, is it written by an analogy to the existing DDs packages by using the recommendations for programming of DDs in Refs. [1,14]. As it is noted above, the complexity of the algorithm is proportional to the size of the SBDD for a given *f*. This is the explanation why the calculation

times are large for relatively small benchmark functions in terms of the number of inputs and outputs, however, having SBDDs of a large size.

For a comparison, the column denoted by *t–cubes* shows the time for calculation of the PHT-spectrum through the disjoint cube representations taken from [13]. The number of disjoint cubes for each function is given in the column (d). These results are given for calculations performed on a HP Apollo Series 735 workstation.

6. Closing remarks

The proposed algorithm permits efficient calculation of PHT-spectra for the functions f_{ON} and f_{DC} assigned to an m-output n-variable switching function f and represented by a SBDD with m pairs of root nodes. Then, PHT-spectrum for f is calculated by processing the nodes in the SBDD. The result is stored in a 2 byte field assigned to each node. Thus, the PHT-spectrum is represented in the thus modified SBDD for f.

The procedure permits efficient calculation of PHT-spectrum for incompletely specified multi-output switching functions with large number of variables and large number of outputs. Memory requirements and time complexity of the algorithm approximate the size of SBDD for f.

The ability to calculate only some spectral coefficients made possible by this research is very important, since there are many application of the Haar spectra in digital logic design for which the values of only selected spectral coefficients are needed [3,4,10,17,21,22].

References

[1] Brace KS, Rudell RL, Bryant RE. Efficient implementation of a BDD package. Proceedings of IEEE Design Automation Conference, June 1990. p. 40–5.
[2] Burrus CS, Gohinut RA, Guo N. Introduction to wavelets and wavelet transforms. Upper Saddle River: Prentice-Hall; 1998.
[3] Chang CH, Falkowski BJ. Generation of quasi-optimal FBDDs though paired Haar spectra. Proceedings of IEEE International Symposium on Circuits and Systems (31st ISCAS), vol. 6. Monterey, California, USA, June 1998. p. 167–70.
[4] Chang CH, Falkowski BJ. Haar spectra based entropy approach to quasi-minimization of FBDDs. IEE Proc Comput Digital Tech 1999;146(1):41–9.
[5] Cieselski MJ, Yang S, Perkowski MA. Multiple-valued minimization based on graph coloring. Proceedings of International Conference on Computer Design: VLSI in Computers and Processors, October 1989. p. 262–5.
[6] Clarke EM, Fujita M, Zhao X. Multi-terminal decision diagrams and hybrid decision diagrams. In: Sasao T, Fujita M, editors. Representations of discrete functions. Boston: Kluwer Academic Publishers; 1996. p. 93–108.
[7] Falkowski BJ, Chang CH. A novel paired Haar transform: algorithms and interpretations in Boolean domain. Proceedings of IEEE Midwest Symposium on Circuits and Systems (36th MWSCAS). Detroit, Michigan, USA, August 1993. p. 1101–4.
[8] Falkowski BJ, Chang CH. Efficient algorithm for forward and inverse transformations between Haar spectrum and binary decision diagrams. Proceedings of IEEE International Phoenix Conference on Computers and Communication (13th IPCCC). Phoenix, Arizona, USA, April 1994, p. 497–503.
[9] Falkowski BJ, Chang CH. Forward and inverse transformations between Haar spectra and ordered binary decision diagrams of Boolean functions. IEEE Trans Comput 1997;46(11):1272–9.

[10] Falkowski BJ, Chang CH. Properties and applications of paired Haar transform. Proceedings of IEEE International Conference on Information, Communications and Signal Processing (1st ICICS), Singapore, vol. 1, September 1997. p. 48–51.
[11] Falkowski BJ, Chang CH. Calculation of paired Haar spectra for systems of incompletely specified Boolean functions. Proceedings of IEEE International Symposium on Circuits and Systems (31st ISCAS), Monterey, California, USA, vol. 6, June 1998. p. 171–4.
[12] Falkowski BJ, Chang CH. Properties and calculation of paired Haar transform. J Approx Theory Appl (ATA) 1999;15(2):1–14.
[13] Falkowski BJ, Chang CH. Paired Haar spectra computation through operations on disjoint cubes. IEE Proc Circuits Devices Syst 1999;146(3):117–23.
[14] Hachtel G, Somenzi F. Logic synthesis and verification algorithms. Boston: Kluwer Academic Publishers; 1996.
[15] Hansen JP, Sekine M. Decision diagrams based techniques for the Haar wavelet transform. Proceedings of IEEE International Conference on Information, Communications and Signal Processing (1st ICICS), Singapore, vol. 1, September 1997. p. 59–63.
[16] Hurst SL. The Haar transform in digital network synthesis. Proceedings of IEEE 11th International Symposium on Multiple-Valued Logic, May 1981. p. 10–18.
[17] Karpovsky MG. Finite orthogonal series in the design of digital devices. New York: Wiley; 1976.
[18] Kulesza W. Systems of spectral analysis of digital data. Warsaw: WKL; 1984 [in Polish].
[19] Newton AR, editor. Selected papers on logic synthesis for integrated circuit design. New York: IEEE Press; 1987.
[20] Perkowski MA, Ross T, Gadd D, Goldman JA, 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. Makuhari, Chiba, Japan, September 1995. p. 102–9.
[21] Ruiz G, Michell JA, 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, editor. Theory and Application of Spectral Techniques. University Dortmund Press, October 1988, p. 97–106.
[22] Ruiz G, Michell JA, Buron A. Switch-level fault detection and diagnosis environment for MOS digital circuits using spectral techniques. IEE Proc Part E 1992;139(4):293–307.
[23] Sasao T, Fujita M, editors. Representations of discrete functions. Boston: Kluwer Academic Publishers; 1996.
[24] Sedgewick R. Algorithms in C. New York: Addison-Wesley; 1990.
[25] Stanković M, Janković D, Stanković RS. Efficient algorithm for Haar spectrum calculation. Sci Rev 1996; 21/22:171–82.
[26] Stanković M, Janković D, Stanković RS. Efficient algorithm for Haar spectrum calculation. Proceedings of IEEE International Conference on Information, Communications and Signal Processing (1st ICICS), Singapore, vol. 4, September 1997. p. 6–10.
[27] Stanković RS, Falkowski BJ. Haar functions and transforms and their generalizations. Proceedings of IEEE International Conference on Information, Communications and Signal Processing (1st ICICS), vol. 4, Singapore, September 1997. p. 1–5.
[28] Stanković RS, Stojić MR, Stanković MS. Recent developments in abstract harmonic analysis with applications in signal processing. Belgrade and Niš: Nauka and Elektronski fakultet; 1996.
[29] Stojić MR, Stanković MS, Stanković RS. Discrete transforms in application. 2nd ed. Belgrade: Nauka; 1993 [in Serbian].
[30] Thornton MA. Modified Haar transform calculation using digital circuit output probabilities. Proceedings of IEEE International Conference on Information, Communications and Signal Processing (1st ICICS), vol. 1, Singapore, September 1997. p. 52–8.
[31] Vetterli M, Kovačević J. Wavelets and subband coding. Upper Saddle River: Prentice-Hall; 1995.
[32] Zalmanzon LA. Fourier, Walsh, and Haart transforms and their applications in control, communication and Other Fields. Moscow: Nauka; 1989 [in Russian].

Milena Stanković received B.E. degree in Electronic Engineering in 1976, and M.Sc., and Ph.D. degrees in Computer Science in 1982 and 1988 from Faculty of Electronics, University of Niš. She was with High School of Electrotechnic, Niš, from 1976 to 1978. From 1978 to date she is with Faculty of Electronics, Niš. Presently, she is a Professor teaching Programming Languages and Compilers. Her

research interests include programming languages, spectral techniques, and signal processing. She was an author and co-author of a couple of monographs and textbooks in computer science.

Dragan Janković received the B.E. and M.Sc. degree in Computer Science from Faculty of Electronic Engineering, University of Nis, in 1991 and 1995, respectively. He is currently working towards the Ph.D. degree and as an Assistant for Programming and Logic Design at the same University. His research interests include decision diagrams, multiple-valued logic, spectral techniques in logic design, and programming.

Radomir S. Stanković received B.E. degree in Electronic Engineering from Faculty of Electronics, University of Niš, in 1976, and M.Sc., and Ph.D. degrees in Applied Mathematics from Faculty of Electrical Engineering, University of Belgrade, in 1984, and 1986, respectively. He was with High School of Electrotechnic, Niš, from 1976 to 1987. From 1987 to date he is with Faculty of Electronic, Niš. Presently, he is a Professor teaching Logic Design. His research interests include switching theory and multiple-valued logic, signal processing and spectral techniques. He served as the co-editor and editor of two editorials and the author of couple of monographs in spectral techniques.

Bogdan J. Falkowski received the M.S.E.E. degree from the Technical University of Warsaw, Poland, and the Ph.D. degree form Portland State University, Oregon, USA. His industrial experience includes research and development positions at several companies from 1978 to 1986. He then joined the Electrical Engineering Department at Portland State University. Currently, he is an Associate Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore which he joined in 1992. His research interests include VLSI systems and design, synthesis and optimization of switching circuits, multiple-valued systems, design of algorithms, design automation, digital signal and image processing. He has published three book chapters and over 150 articles in the refereed journals and conferences. Dr. Falkowski is a Senior Member of IEEE and a Member of IEEE Computer Society and IEEE Circuits and Systems Society. He is a member of Eta Kappa Nu and Pi Beta Upsilon.