Automation and Remote Control, Vol. 63, No. 1, 2002, pp. 125–138. Translated from Avtomatika i Telemekhanika, No. 1, 2002, pp. 140–153.

A Unifying Approach to Edge-valued and Arithmetic Transform Decision Diagrams

C. Moraga*, T. Sasao, and R. Stanković*****

* *University of Dortmund, Dortmund, Germany*
** *Kyushu Institute of Technology, Iizuka, Japan*
*** *University of Niš, Niš, Yugoslavia*
Received May 27, 2000

Abstract—This paper shows that binary decision diagrams (BDDs) and their generalizations are not only representations of switching and integer-valued functions, but also Fourier-like series expansions of them. Furthermore, it shows that edge-valued binary decision diagrams (EVBDDs) are related to arithmetic transform decision diagrams (ACDDs), which are the integer counterparts of the functional decision diagrams (FDDs). Finally, it shows that the complexity of multi-terminal binary decision diagrams (MTBDDs), EVBDDs and ACDDs of a function f depends on the structure of the truth-vector of f, partial arithmetic transform spectra of f and the arithmetic transform spectrum of f, respectively.

1. INTRODUCTION

The basic concepts which we now denote as decision trees (DTs) were introduced in [18] for discrete sets representations, and exploited in representations of switching functions in [19]. Some other authors used the same way of representation of switching and multiple-valued functions [1, 21, 37, 27]. Due to that pioneering work of a few authors, DTs and decision diagrams (DDs), derived by the reduction of DTs by sharing isomorphic subtrees, have been accepted as a data structure for representation of discrete functions, the mappings between discrete sets.

Since the publication of the paper by Bryant [5] over a decade ago, much work has been done to better understand, to develop and to generalize binary decision diagrams (BDDs) [31]. In this same period of time, work has been done to apply methods of abstract harmonic analysis to logic design and pattern analysis [13, 15, 16, 26]. This activity has been referred to as Spectral Techniques in the literature. Spectral techniques are known to suffer under the "curse of dimensionality," since operations with matrices that are larger than can be supported with todays computers may be required. A breakthrough was achieved with the works of E. Clarke and his colleagues [9, 10], where decision diagrams are shown to be efficient data structures to alleviate the dimensionality burden in spectral techniques. This in turn has motivated new works developing a spectral view of decision diagrams (DDs) (Chapters 3, 4, 5 and 6 of [31]). The present contribution falls in this last category. The rest of the paper is organized as follows: Section 2 introduces the series expansion of switching functions and its relation to decision diagrams. This section is a tutorial and is included for the sake of completeness. Readers already familiar with decision diagrams may move directly to Section 3. Arithmetic transform decision diagrams (ACDDs) are discussed in Section 3, where the concept of partial spectrum is introduced and a functional decomposition based on ACDDs is deduced. The main contribution of the paper is presented in Section 4, where a comparative analysis of edge-valued decisions diagrams (EVBDDs) and ACDDs is conducted and their strong relationship is proven.

0005-1179/02/6301-0125$27.00 © 2002 MAIK "Nauka/Interperiodica"

2. SERIES EXPANSIONS OF SWITCHING FUNCTIONS

Switching functions can be regarded as functions from the finite dyadic group into the Galois field $GF(2)$. The finite dyadic group D of order 2^n consists of the set of n-tuples $(x_1, \ldots, x_n)$, where x_i takes the values 0 or 1, the operation is the componentwise modulo 2 addition, usually denoted as EXOR (short for exclusive OR). This group can be represented as the direct product of the basic cyclic groups of order 2, $C_2 = (\{0, 1\}, \oplus)$. The set of all switching functions of a given number of vriables expresses the structure of the vector space with operations EXOR and AND applied componentwise over the truth-vectors of the functions. This space is denoted by $GF(C_2^n)$. In that setting, various AND-EXOR expressions [30] can be considered as a Fourier series-like expansion [16] in terms of some complete function sets $Q = \{\varphi_i(\cdot)\}, i = 0, \ldots, 2^n - 1$, that are bases in $GF(C_2^n)$.

For example, the positive polarity Reed–Muller (PPRM) expression [30], is a Fourier-like series expansion in terms of the Reed–Muller functions described by the elementary products of switching variables. In matrix notation, they are represented by the columns of the matrix

$$\mathbf{R}(n) = \bigotimes_{i=1}^{n} \mathbf{R}(1),$$

where the basic Reed–Muller matrix $\mathbf{R}(1)$ is given by

$$\mathbf{R}(1) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

Since $\mathbf{R}(1)$ is self-inverse in $GF(2)$, $\mathbf{R}(n)$ is also self-inverse in $GF(2)$.

$\mathbf{R}(n)$ as defined above is called the Reed–Muller matrix in the Hadamard ordering [14, 15]. This matrix is used to compute the coefficients of the PPRM of a given switching function, as is illustrated below.

$[\mathbf{R}(n)]^{-1}$ is given by the vector expression

$$[\mathbf{R}(n)]^{-1} = \mathbf{R}(n) = \bigotimes_{i=1}^{n} [1 - x_i].$$

$\mathbf{R}(n)$ generates the product terms appearing in PPRM, corresponding to the Reed–Muller functions. Since the columns of $\mathbf{R}(n)$ form a basis in $G(C_2^n)$, the coefficients c_i, $i = 0, \ldots, 2^n - 1$ of the PPRM of a given switching function f are defined as the scalar products (in $G(C_2^n)$) of the Reed–Muller functions represented by $\mathbf{R}(n)$ with the truth vector $\mathbf{F}$ corresponding to f.

The PPRM is obtained by computing

$$f(x_1, \ldots, x_n) = (\mathbf{R}(n))^{-1} \mathbf{R}(n) \mathbf{F},$$

where $(\mathbf{R}(n))^{-1}$ is the vector of the symbolic expression, $\mathbf{R}(n)$ is the binary matrix and $\mathbf{F}$ is the column truth vector of f. The vector resulting from $\mathbf{R}(n)\mathbf{F}$ is called the Reed–Muller spectrum of f.

Truth table of Example 1

x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	0
1	0	1
1	1	1

Example 1. Let $f : \{0,1\}^2 \to \{0,1\}$ be a switching function shown in Table,

$$(\mathbf{R}(2))^{-1} = \bigotimes_{i=1}^{2} [1 \ x_i] = [1 \ x_2 \ \ x_1 \ \ x_1x_2]$$

and

$$\mathbf{R}(2) = \bigotimes_{i=1}^{2} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

The coefficients c_0, c_1, c_2 and c_3 are given by

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} [1\ 0\ 1\ 1]^T = [c_1 \ \ c_2 \ \ c_3 \ \ c_4]^T = [1\ 1\ 0\ 1]^T.$$

This leads to the following PPRM

$$f(x_1, x_2) = 1 \oplus x_2 \oplus x_1x_2,$$

where $\oplus$ denotes the EXOR.

This spectral transform approach to the PPRM of a switching function is easily extended to other expressions. The disjunctive normal form of f, for instance, is (formally) given as a Fourier-like series expansion in terms of the trivial basis

$$E = \{e_{r,j}\}, \quad r, j \in \{0, \dots, 2^n - 1\}, \tag{1}$$

where if $r = j$ then $e_{r,j} = 1$ and if $r \neq j$ then $e_{r,j} = 0$.

In analogy to the former development, define:

$$\mathbf{I}(n) = \bigotimes_{i=1}^{n} \mathbf{I}(1) = \bigotimes_{i=1}^{n} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

with the associated symbolic expression

$$[\mathbf{I}(n)]^{-1} = \mathbf{I}(n) = \bigotimes_{i=1}^{n} [\overline{x}_i \ \ x_i].$$

For the former example, this leads to

$$(\mathbf{I}(2))^{-1} = \bigotimes_{i=1}^{2} [\overline{x}_i \ \ x_i] = [\overline{x}_1\overline{x}_2 \ \ \overline{x}_1x_2 \ \ x_1\overline{x}_2 \ \ x_1x_2]$$

and since

$$\mathbf{I}(2)[1\ 0\ 1\ 1]^T = [1\ 0\ 1\ 1]^T,$$

it follows that

$$f(x_1, x_2) = \overline{x}_1\overline{x}_2 \oplus \overline{x}_1x_2 \oplus x_1x_2.$$

Besides the PPRM and the disjunctive normal form, the spectral interpretation is extended to other AND-EXOR representations of switching functions as, for example, fixed polarity Reed–Muller expressions (FPRMs), Kronecker and pseudo-Kronecker expressions [31] in terms of some suitably defined bases in $G(C_2^n)$ [36]. Moreover, it has been shown [31] that these AND-EXOR expansions can be generated by decision trees. In the spectral interpretation, these decision trees are graphical representations of the Fourier-like series expansions with respect to the corresponding bases. In a decision tree, each path from the root node up to a constant node is associated to a basic function φ_j from the related basis. The constant nodes of this tree contain the values of the corresponding expansion coefficients $q_i = <\varphi_i, f>$, where $<\cdot,\cdot>$ denotes the scalar product in $G(C_2^n)$. After the application of suitably defined reduction rules [36], the data structure of a tree transfers into the structure of an acyclic graph called decision diagram [5, 28].

For example, BDDs [5] are defined from the trees associated to the disjunctive normal form through the Shannon expansion theorem. Therefore, BDDs are graphical representations of Fourier-like series expansions with respect to the trivial basis (1), described by minterms. Functional decision diagrams, FDDs are defined with respect to the trees associated to Reed–Muller expressions through the Davio expansions. They are, therefore, graphical representations of a series expansion in terms of the Reed–Muller functions. Kronecker decisions diagrams, KDDs [31], are derived from the trees whose basic functions φ_i are defined through the Kronecker product of arbitrary combinations of elementary matrices of the form $[\overline{x}_i \; x_i]$, $[1 \; x_i]$ and $[1 \; \overline{x}_i]$ corresponding to the Shannon, positive Davio and negative Davio expansions, respectively [36]. Pseudo Kronecker decision diagrams, PSKDDs, are derived from the trees whose basic functions are combinations of the basic functions of different KDDs, provided that they are linearly independent [31]. Such trees are used for an alternative definition of various AND-EXOR expressions. A systematization of AND-EXOR expressions and their corresponding decision diagrams is given in [30].

3. ARITHMETIC TRANSFORM DECISION DIAGRAMS

In the spectral transform approach, switching functions are considered as elements of the complex vector space $C(C_2^n)$ of functions from D into the complex field C by interpreting the logical 0 and 1 as integers. Furthermore, the operations AND and EXOR are replaced by multiplication and addition in C, respectively.

Under this assumption, the trivial basis (1) is also a basis in $C(C_2^n)$. This allows defining multiterminal decision diagrams (MTBDDs) as a direct extension of BDDs, and using them for the representation of integer-valued functions, particularly the Walsh spectrum of switching functions [7, 8, 10]. Moreover, it is possible to develop procedures for the calculation of Walsh spectra from OBDDs and synthesis of OBDDs from the Walsh spectrum [11]. As noted in [36], MTBDDs are also called algebraic decision diagrams (ADDs) [4].

Taking in account the above assumptions on the interpretation of logical constants as integers and the replacement of the corresponding operations, Reed–Muller functions can be considered to build an integer function basis in $C(C_2^n)$ that permits the definition of arithmetic polynomials or, equivalently, the arithmetic transform in $C(C_2^n)$ [13, 22, 25, 26, 34].

The coefficients of these arithmetic expressions for a given function f are defined as the scalar products of the columns of the arithmetic transform matrix $\mathbf{A}(n)$ with $\mathbf{F}$. $\mathbf{A}(n)$ is defined as the inverse of $\mathbf{R}(n)$ over C

$$\mathbf{A}(n) = \bigotimes_{i=1}^{n} \mathbf{A}(1),$$

where

$$\mathbf{A}(1) = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} = [\mathbf{R}(1)]^{-1} \quad \text{over} \quad C.$$

The associated symbolic expression for $\mathbf{A}(n)^{-1}$ is given by

$$[\mathbf{A}(n)]^{-1} = \mathbf{R}(n) = \bigotimes_{i=1}^{n} [1\ x_i].$$

With respect to the former example, this would lead to

$$f(x_1, x_2) = [\mathbf{A}(2)]^{-1}\mathbf{A}(2)\mathbf{F},$$

$$f(x_1, x_2) = ([1\ x_1] \otimes [1\ x_2]) \left(\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \right) [1\ 0\ 1\ 1]^T$$

$$= [1\ x_2\ x_1\ x_1x_2] \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & -1 & -1 & 1 \end{bmatrix} [1\ 0\ 1\ 1]^T$$

$$= [1\ \ x_2\ \ x_1\ \ x_1x_2][1\ \ -1\ 0\ 1]^T = 1 - x_2 + x_1x_2.$$

From the theory of fast Fourier transforms [16], the matrix $\mathbf{A}(n)$ may be factorized as follows:

$$\mathbf{A}(n) = \bigotimes_{j=0}^{n-1} \Big((\mathbf{I}(1))^j \mathbf{A}(1) (\mathbf{I}(1))^{n-j-1} \Big)$$

$$= \Big(\mathbf{A}(1)(\mathbf{I}(1))^{n-1}) \otimes (\mathbf{I}(1)\mathbf{A}(1)(\mathbf{I}(1))^{n-2}) \otimes \cdots \otimes ((\mathbf{I}(1)^{n-1})\mathbf{A}(1) \Big)$$

$$= (\mathbf{A}(1) \otimes \mathbf{I}(1) \otimes \cdots \otimes \mathbf{I}(1)) \, (\mathbf{I}(1) \otimes \mathbf{A}(1) \otimes \mathbf{I}(1) \otimes \cdots \otimes \mathbf{I}(1)) \cdots (\mathbf{I}(1) \otimes \cdots \otimes \mathbf{I}(1) \otimes \mathbf{A}(1))$$

$$= \prod_{j=1}^{n} \mathbf{U}_j(n),$$

where

$$\mathbf{U}_j(n) = \bigotimes_{i=1}^{n} \mathbf{u}_{ij}(1)$$

and $\mathbf{u}_{ij} = \mathbf{A}(1)$ if $i = j$, and $\mathbf{u}_{ij} = \mathbf{I}(1)$ otherwise.

The matrix $\mathbf{U}_j$ defines the partial arithmetic transform of f with respect to the j-th variable x_j.

Definition 1. For a function f with a truth-vector $\mathbf{F} = [f(0), \ldots, f(2^n - 1)]^T$, the partial arithmetic spectrum with respect to the i-th variable $\mathbf{S}_{if}$, is given by

$$\mathbf{S}_{if} = [S_{if}(0), \ldots, S_{if}(2^n - 1)]^T = \mathbf{U}_i(n)\mathbf{F}.$$

The corresponding (partial) series expansions are

$$f = [1\ x_i]\mathbf{S}_{if}, = [1\ x_i]\mathbf{U}_i(n)\mathbf{F} \quad (i = 1, 2, \ldots, n).$$

For a given function f, let f_{i0} and f_{i1} denote the subfunctions obtained by fixing x_i to 0 and 1 respectively. Then, the vector $[\mathbf{F}_{i0}\ \mathbf{F}_{i1}]^T$ represents the permutation of $\mathbf{F}$ that is obtained by

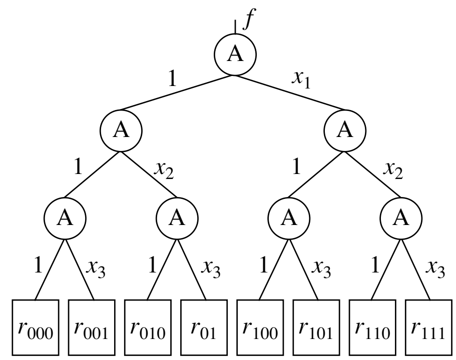

Fig. 1. ACDT for $n = 3$.

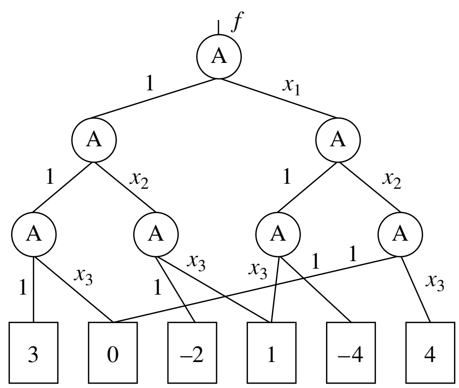

Fig. 2. ACDD of f in Example 2.

reordering the arguments such that x_i takes the first position and all other arguments keep their relative ordering. This leads to the following series expansion:

$$\begin{aligned} f &= [1\ x_i](\mathbf{A}(1) \otimes \mathbf{I}(1) \otimes \cdots \otimes \mathbf{I}(1))[\mathbf{F}_{i0}\ \mathbf{F}_{i1}]^T \\ &= [1\ x_i](\mathbf{A}(1) \otimes \mathbf{I}(1)^{[n-1]})[\mathbf{F}_{i0}\ \mathbf{F}_{i1}]^T, \end{aligned}$$

where $\mathbf{I}(1)^{[n-1]}$ denotes the $(n-1)$-fold Kronecker product of 1(1) with itself. Then

$$\begin{aligned} f &= [1\ x_i]\begin{bmatrix} \mathbf{I}(1)^{[n-1]} & 0 \\ -\mathbf{I}(1)^{[n-1]} & \mathbf{I}(1)^{[n-1]} \end{bmatrix}[\mathbf{F}_{i0}\ \mathbf{F}_{i1}]^T \\ &= [1\ x_i][\mathbf{F}_{i0}\ (\mathbf{F}_{i1} - \mathbf{F}_{i0})]^T = 1 \times \mathbf{F}_{i0} + x_i(\mathbf{F}_{i1} - \mathbf{F}_{i0}). \end{aligned}$$

When no confusion arises, the "i"-indices of the truth-vectors are omitted. This leads to the more familiar notation found in the literature:

$$f = 1 \times f_0 + x(f_1 - f_0). \tag{2}$$

Similar to the functional decision diagrams (FDDs), a tree can be associated to each arithmetic expression for a given discrete function f. In that tree, each path from the root up to the constant nodes corresponds to a basis function and the constant nodes contain the coefficients of the arithmetic spectrum of f [30]. Binary moment decision diagrams (BMDs) [5] are similar to ACDDs, but derived by using different reduction rules.

Example 2. Figure 1 shows the ACDT for functions of $n = 3$ varibales. Figure 2 shows the reduced ACDD for

$$f(x_1, x_2, x_3) = 3 - 4x_1 + 4x_1x_2 + x_1x_3 - 2x_2 + x_2x_3,$$

taken from Example 1 in [20]. The constant nodes represent the arithmetic spectrum of f given by $\mathbf{A}_f = [3, 0, -2, 1, -4, 1, 4, 0]^T$. This ACDD was drawn by using the drawing rule defined by expansion (2).

4. EDGE-VALUED DECISION DIAGRAMS

EVBDDs are defined by using the expression [20]

$$x(v_l + f_l) + (1 - x)(v_r + f_r), \tag{3}$$

instead the Shannon expansion. The next example introduces the concept.

Example 3. Figure 3 [20] shows the EVBDD for the function $f = 3 - 4x_1 + 4x_1x_2 + x_1x_3 - 2x_2 + x_2x_3$ in Example 2. Note that the reduction is done by using the same reduction rules as with MTBDDs and ACDDs. These rules are a generalization of BDD reduction rules [36].

In EVBDDs, the values of constant nodes are set to 0, and the calculation procedure is not done by nodes, but is transferred to edges. That is achieved by associating the corresponding value v_l to the left outgoing edges, while v_r is enforced to be 0 to provide canonical representation [20]. That can be interpreted as follows.

In the ACDDs defined by arithmetic rules (2) and (3), as the drawing and reading rules, the constant nodes represent the arithmetic spectrum A_f of f. To get the zero at the constant nodes in the EVBDD, these values should be subtracted, and just that was actually done recursively at the edges of the EVBDD by the additive correction v_l in (3). Note that in [20] the inverse notation is used in which the left outgoing edge corresponds to the logical 1 and the right outgoing branch corresponds to the logical 0, respectively.

Written as $f = xv_l + f_0 + x(f_1 - f_0)$, since $v_r = 0$, the drawing rule (3) for the EVBDDs, equals that of the ACDDs, except for the additive constant xv_l introduced to transfer the calculation procedure to the edges and to set the values of the constant nodes to zero. More precisely, in each node of the ACDD, the arithmetic transform is being implemented with respect to a particular

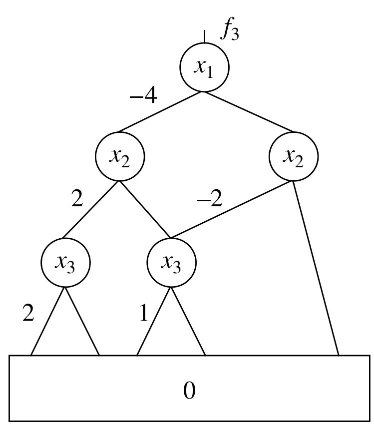

Fig. 3. EVBDD of f in Example 2.

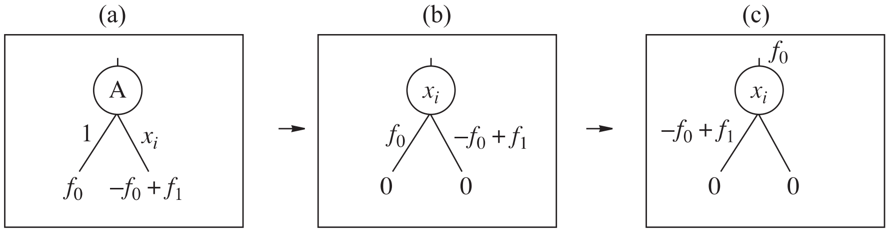

Fig. 4. Relationship among the nodes of ACDDs and EVBDDs.

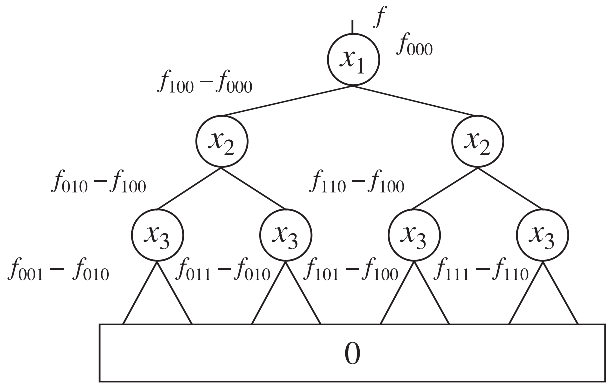

Fig. 5. Complete EVBDD for $n=3$ and the partial arithmetic transform coefficients.

variable, which, by definition of the arithmetic transform, results into the values f_0 for $x=0$ and f_1-f_0 for $x=1$. These values should be subtracted at the right and left outgoing edges in the EVBDD to achieve the zero at the constant nodes. This is the reason that the value at the right outgoing edge is always determined as $v_r=f_0-f_0=0$, while the value at the left outgoing edge is calculated as $v_l=f_1-f_0$. Note that the value at the entering edge at the root node is $f(0\cdots 0)$ which is by definition $A_f(0\cdots 0)$. Therefore, if the constant nodes are set to zero and the calculation procedure is done at the edges as that was done in the EVBDDs, an ACDD node shown in Fig. 4a translates into the node at Fig. 4b, and further into the EVBDD node in Fig. 4c for the inverse notation used in the EVBDDs.

In that way, the calculation procedure for the determination of labels in the EVBDD goes from the right to the left side of the DD, as shown in the Fig. 4.

The explanation becomes obvious if the calculation of the values v_l at a complete edge-valued decision tree (EVBDT) is considered.

Example 4. Figure 5 shows the complete EVBDD, i.e., EVBDT for $n=3$, and the calculations of the values v_l. Relationship of the values at the edges of this tree to the values of the spectra of the corresponding partial arithmetic transforms is obvious from the matrix relations for $\mathbf{S}_{if}$ in Definition 1. From this definition,

$$\begin{aligned}
\mathbf{S}_{1f}(100) &= f_{100}-f_{000},\\
\mathbf{S}_{2f}(010) &= f_{010}-f_{000},\\
\mathbf{S}_{2f}(110) &= f_{110}-f_{100},\\
\mathbf{S}_{3f}(001) &= f_{001}-f_{000},\\
\mathbf{S}_{3f}(011) &= f_{011}-f_{010},\\
\mathbf{S}_{3f}(101) &= f_{101}-f_{100},\\
\mathbf{S}_{3f}(111) &= f_{111}-f_{110}.
\end{aligned}$$

Theorem 1. *An EVBDD represent f in the form of the arithmetic polynomial for f.*

Proof. The proof is obvious, since the decomposition rule applied at the nodes in EVBDD corresponds to the symbolic notation for the basic arithmetic transform matrix $\mathbf{A}(1)$, and it is applied recursively, level by level, through the decision tree to all the variables in f.

Theorem 2. *Weighting coefficients at the edges at the i-th level in an edge-valued binary decision tree (EVBDT) for f are the partial arithmetic transform coefficients of f with respect to the variable x_i at the positions 2^{n-i} in $\mathbf{S}_{if}$.*

Proof. The proof follows from the definition of the expansion rules applied at the nodes of EVBDTs and definition of the partial arithmetic transforms.

As in the case of ACDDs, a given EVBDD of f represents at the same time the arithmetic spectrum A_f of f, which can be read from the EVBDD by using the drawing rule as the reading rule. To show the relationship to the ACDDs, the reading rule for determination of the algebraic spectrum A_f from the EVBDD of f can be written formally as $A_f = 1 \times 0 + A_{f_0} + x_i(0 - 0 + A_{f_1} - A_{f_0})$, or $A_f = 1 \times 0 + v_r + x_i(0 - 0 + v_l)$, since by definition, $v_0 = A_{f_0}$ and $v_l = A_{f_1} - A_{f_0}$. Note that, as with the ACDDs, A_f are used instead of f in the formulation, since the determination of an integer-valued function that is the polynomial representing the arithmetic spectrum of another integer-valued function is under consideration.

The determination procedure goes from the bottom to the top, as in the case of ACDDs, but from the left to the right nodes, for the inverse notation used in [20]. The constant nodes are passed first. It is assumed, from the definition of EVBDDs, that after the nodes at the level i are passed, they become the constant nodes for the level $i-1$ and, therefore, should be set to zero. In that way, the calculation always concerns the values at the edges.

Theorem 3. *An EVBDD for f represents at the same time f and the arithmetic spectrum S_f of f.*

Proof. The proof follows from spectral interpretation of DDs, since coefficients at the edges are values of partial arithmetic transform coefficients, and the recursive structure of decision trees, as well as the Kronecker product structure of the arithmetic transform matrix.

By using the inverse arithmetic transform $\mathbf{A}(1)^{-1}$ at the nodes in EVBDD, we read the arithmetic spectrum for f.

Therefore, the EVBDDs correspond to ACDDs with the modified notation to achieve the savings in the storage of the constant nodes at the price of the storage of the values attached to the left outgoing edges. However, it remains to estimate whether the average number of nodes is greater or lower than the number of values attached to edges for functions of a considerable number of variables. As is noted in [20], for functions where the number of distinct terminal nodes is small, MTBDDs, may be the more space efficient. The following general comment may be given.

In a multi-terminal binary decision tree, the values of constant nodes are the function values of f. The reduction is possible if there are some constant or equal subvectors of orders 2^k, $k \leq n-1$ the truth-vector of f. Therefore, the size of a MTBDD, defined as the total number of nodes including the constant nodes, depends upon the function f, i.e., upon the structure of its truth-vector $\mathbf{F}$. That will be interpreted as the representation complexity of f, which is determined by the number of different values of f, and the eventual periodicity of the function values in $\mathbf{F}$. If a value repeats in $\mathbf{F}$, and if a sequence of 2^k elements appears in $\mathbf{F}$ periodically, then the representation complexity of f decreases and the corresponding MTBDD is simpler.

In an arithmetic transform decision tree, the values of constant nodes are the arithmetic transform coefficients. Therefore, the size of ACDDs depends upon the structure of the arithmetic spectrum A_f of f in the same way as the size of MTBDD depends on the structure of the truth-vector of f. In that respect, the function f in Example 1 is neither convenient for the representation by ACDDs nor MTBDDs, since there are no many constant or equal subvectors in both truth-vector for f or its arithmetic spectrum.

The size of EVBDDs depends in the same way upon the complexity of the truth-vectors of the partial arithmetic transforms, since the values v_l in the EVBDD take the corresponding values of these truth-vectors of the partial arithmetic transforms as is shown in Fig. 5.

Remark 1. Reduction at the i-th level in a DD is performed by deleting or sharing nodes representing isomorphic sub-trees depending on the equal or otherwise assignment of the values of variables at the $(z-1)$-th level in the DD.

1. In MTBDDs, possibility to delete or share a node depends on the function values of the represented functions f.

2. In ACDDs, possibility to delete or share a node depends on the values of arithmetic transform coefficients of f.

3. In EVBDDs, possibility to delete or share a node depends on the values of partial arithmetic transform coefficients of f.

Proof. Proof follows from the spectral interpretation of the values of constant nodes in MTBDDs and ACDDs, and weighting coefficients at the edges in EVBDDs.

Unlike EVBDDs, besides the optimization by choosing the best suited order of variables, ACDDs offer the possibility of optimization by using the negative arithmetic expansions $f = 1 \times f_1 + \overline{x}_i(f_0 - f_1)$ in the same way as that was done by using the negative Davio expansions in the case of fixed polarity Reed–Muller expressions (FPRMs) [29].

By using the nodes corresponding to different expansion rules in the complex vector space, a variety of polynomial representation and the corresponding decision diagrams can be defined corresponding to the Kronecker and pseudo Kronecker decision diagrams [30].

Thus, ACDDs are a concept defined as a particular example of a general theory originating in the Fourier series representations of signals [14–16, 32, 34, 38].

The reading rule to determine the function f represented by a given EVBDD is expressed by the function (3) $x(v+l) + (1-x)r$, which can be written as $x(v+l) + \overline{x}r$, where $\overline{x}$ denotes the logical complement of x, since x is a switching variable.

Example 5. Consider the tree in Fig. 5. This tree represents the function

$$\begin{aligned} f = f_{000} + x_1(f_{100} - f_{000} + x_2(f_{110} - f_{100} + x_3(f_{111} - f_{110} + 0) + \overline{x}_3 \times 0) \\ + \overline{x}_2(x_3(f_{101} - f_{100} + 0) + \overline{x}_3 \times 0)) \\ + \overline{x}_1(x_2(f_{010} - f_{000} + x_3(f_{011} - f_{010} + 0) + \overline{x}_3 \times 0) \\ + \overline{x}_2(x_3(f_{001} - f_{000} + 0) + \overline{x}_3)) \\ = f_{000} + x_1(f_{100} - f_{000}) + \overline{x}_1x_2(f_{010} - f_{000}) + x_1x_2(f_{110} - f_{100}) \\ + \overline{x}_1\overline{x}_2x_3(f_{001} - f_{000}) + x_1\overline{x}_2x_3(f_{101} - f_{100}) \\ + \overline{x}_1x_2x_3(f_{011} - f_{010}) + x_1x_2x_3(f_{111} - f_{110}). \end{aligned} \tag{4}$$

This expression is considered as the expansion of f relative to the basic functions $\varphi_0 = 1$, $\varphi_1 = x_1$, $\varphi_2 = \overline{x}_1x_2$, $\varphi_3 = x_1x_2$, $\varphi_4 = \overline{x}_1\overline{x}_2x_3$, $\varphi_5 = x_1\overline{x}_2x_3$, $\varphi_6 = \overline{x}_1x_2x_3$, $\varphi_7 = x_1x_2x_3$, which are represented by the columns of the matrix

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The inverse matrix $\mathbf{Q}^{-1}$ is

$$\mathbf{Q}^{-l} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

$\mathbf{Q}^{-1}$ is the transform matrix to determine the coefficients in (4).

The tree in Fig. 6 represents this expansion. The constant nodes contain spectral coefficients S_f of f defined by

$$\mathbf{S}_f = \mathbf{Q}^{-1}\mathbf{F},$$

where $\mathbf{F} = [f_{000}, f_{001}, f_{010}, f_{011}, f_{100}, f_{101}, f_{110}, f_{111}]^T$.

Note that this is a particular pseudo-Kronecker tree [30]. Therefore, an EVBDD is considered as a Kronecker DD with respect to a particular basis.

Theorem 4. *An EVBDT is isomprphic to a pseudo-Kronecker tree with the positive Davio node as the root node and as the leftmost nodes at each level, while the other nodes are Shannon nodes, and with variables taken in the reversed order.*

Proof. Proof follows from the interpretation of $(1 - x)$ as an equivalent of $\overline{x}$ analysis of the calculation procedure for determination of f from EVBDD and relationship between the nodes in EVBDDs and ACDDs.

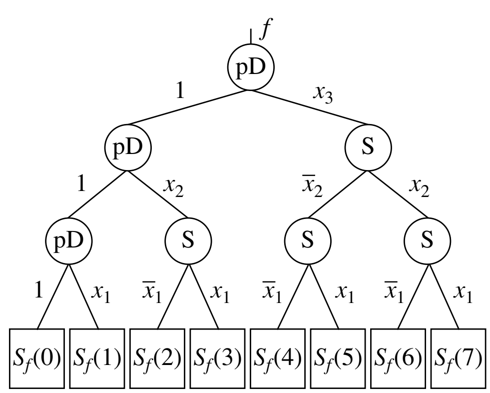

Fig. 6. Pseudo-Kronecker DT corresponding to the EVBDT for $n = 3$.

5. CLOSING REMARKS

Decision trees are graphical representations of Fourier-like series expansions with respect to various basic functions. Decision diagrams (DDs) are derived from these decision trees by using the reduction rules. DDs for integer-valued functions, such as MTBDDs, EVBDDs, ACDDs, and WDDs, are integer counterparts of the corresponding decision diagrams associated to some AND-EXOR expressions. They are derived with respect to the same basic sets of functions, but considered over the complex field in the direct or $\{0,1\} \rightarrow \{-1,1\}$ coding.

EVBDDs are a different representation of integer pseudo-Kronecker DDs. They are related to the partial arithmetic transforms. Their complexity depends on the structure of vectors representing partial arithmetic spectra of f.

DDs are also considered as graphical representations of spectral transforms expansions of switching and integer valued functions. This is an explanation why they are efficient in spectral transforms computations, and in the manipulations and calculations in expressible through matrix relations [9, 24].

The use of some other spectral transforms, possibly non-linear, but invertible, as for example, the sign transform [3, 38], could permit the derivation of some new classes of spectral transforms decision diagrams that do not have the proper counterparts in AND-EXOR related DDs.

EVBDDs require integer weights even for representation of single-output switching functions. That is a disadvantage, since the represented functions are two-valued. This disadvantage is overcome by defining the edge-valued functional binary DDs (EVFBDDs) [35, 33]. EVFBDDs are defined in terms of the partial Reed–Muller transforms in the same way as the EVBDDs are defined in terms of the partial arithmetic transforms.

REFERENCES

1. Akers, S.B., Binary Decision Diagrams, *IEEE Trans. Computers*, 1978, vol. C-27, no. 6, pp. 509–516.
2. Becker, B., Drechsler, R., and Theobald, M., On the Implementation of a Package for Efficient Representation and Manipulation of Functional Decision Diagrams, *Proc. IFIP WG 10.5 Workshop on Applications of the Reed–Muller Expansion in Circuit Design*, Kebschull, U., Schubert, E., and Rosenstiel, W., Eds., Hamburg, Germany, 1993, pp. 162–169.
3. Besslich, Ph.W. and Trachtenberg, E.A., Binary Input/Ternary Output Switching Circuits Designed via the Sign Transform, *Proc. 22nd Int. Symp. on Multiple-Valued Logic*, Sendai, Japan, 1992, pp. 348–354.
4. Bahar, R.I., Frohm, E.A., Gaona, C.M., Hachtel, G.D., Macii, E., Pardo, A., and Somenzi, F., Algebraic Decision Diagrams and Their Applications, *Int. Conf. on CAD*, 1993, pp. 188–191.
5. Bryant, R.E., Graph-Based Algorithms for Boolean Functions Manipulation, *IEEE Trans. Comput.*, 1986, vol. C-35, no. 8, pp. 667–691.
6. Bryant, R.E. and Chen, Y-A., Verification of Arithmetic Functions with Binary Moment Decision Diagrams, unpublished paper, 1994, CMU-CS-94-160.
7. Clarke, E.M., McMillan, K.L., Zhao, X., Fujita, M., and Yang, J., Spectral Transforms for Large Boolean Functions with Application to Technology Mapping, *30th A CM/IEEE Design. Autom. Conf.*, 1993.
8. Clarke, E.M., Zhao, X., Fujita, M., Matsunaga, Y., and McGeer, R., Fast Walsh Transform Computation with Binary Decision Diagram, *Proc. IFIP WG 10.5 Workshop on Applications of the Reed–Muller Expansion in Circuit Design*, Kebschull, U., Schubert, E., and Rosenstiel, W., Eds., Hamburg, Germany, 1993, pp. 82–85.
9. Clarke, E.M., McMillan, K.L., Zhao, X., and Fujita, M., Spectral Transforms for Extremely Large Boolean Functions, *Proc. IFIP WG 10.5 Workshop on Applications of the Reed–Muller Expansion in Circuit Design*, Kebschull, U., Schubert, E., and Rosenstiel, W., Eds., Hamburg, Germany, 1993, pp. 86–90.

10. Clarke, E.M., Fujita, M., McGeer, P.O., McMillan, K.L., and Yang, J.C.-Y., Multi-Terminal Binary Decision Diagrams: An Efficient Data Structure for Matrix Representation, *Int. Workshop on Logic Synthesis*, 1993.

11. Falkowski, B.J. and Chang, C.H., Efficient Algorithms for the Calculation of Walsh Spectrum from OBDD and Synthesis of OBDD from Walsh Spectrum for Incompletely Specified Boolean Functions, *37th Midwest Symp. on Circuits and Systems*, Lafayette, Louisiana, U.S.A., 1994.

12. Falkowski, B.J. and Chang, C.H., Efficient Algorithm for the Calculation of Arithmetic Spectrum from OBDD and Synthesis of OBDD from Arithmetic Spectrum for Incompletely Specified Boolean Functions, *IEEE Int. Symp. on Circuits and Systems ISCAS94*, U.S.A., 1994.

13. Heidtmann, K.D., Arithmetic Spectrum Applied to Fault Detection for Combinatorial Networks, *IEEE Trans. Comput.*, 1991, vol. 40, no. 3, pp. 320–324.

14. Hurst, S.L., *Logical Processing of Digital Signals*, London: Crane Russak and Edward Arnold, 1978.

15. Hurst, S.L., Miller, D.M., and Muzio, J.C., *Spectral Techniques in Digital Logic*, Bristol: Academic, 1985.

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

17. Kukharev, G.A., Shmerko, V.P., and Yanushkievich, S.N., *Technique of Binary Data Parallel Processing for VLSI*, Minsk: Vysheyshaja Shcola, 1991.

18. Kurepa, Dj.R., *Ensembles ordonnés et ramifiès*, Paris, 1935.

19. Kurepa, Dj.R., Sets–Logics–Machines, *Proc. Int. Symp. Theory of Switching*, Harvard University, Cambridge, 1957, part. 1, pp. 137–146.

20. Lai, Y.F., Pedram, M., and Vrudhula, S.B.K., EVBDD-Based Algorithms for Integer Linear Programming, Spectral Transformation, and Functional Decomposition, *IEEE Trans. Computer-Aided Design*, 1994, vol. 13, no. 8, pp. 959–975.

21. Lee, C.Y., Representation of Switching Circuits by Binary Decision Diagrams, *Bell Syst. Tech. J.*, 1959, vol. 38, pp. 985–999.

22. Malyugin, V.D., On a Polynomial Realization of Cortege of the Boolean Functions, *Vestn. Akad. Nauk SSSR*, 1982, vol. 265, no. 6.

23. Malyugin, V.D. and Sokolov, V.V., Intensive Logical Calculations, *Avtom. Telemekh.*, 1993, no. 4, pp. 160–167.

24. Malyugin, V.D., Stanković, R.S., and Stanković, M., Calculations of the Coefficients of Polynomial Representations of Switching Functions Through Binary Decision Diagrams, *Proc. Preventive Engineering and Information Technologies*, Niš, Yugoslavia, 1994, pp. 10-1–10-4.

25. Malyugin, V.D. and Veits, A.V., Intensive Calculations in Parallel Logic, *Proc. 5th Int. Workshop on Spectral Techniques*, 1994, Beijing, China, pp. 63–64.

26. Muzio, J.C., Stuck Fault Sensitivity of Reed–Muller and Arithmetic Coefficients, in *Theory and Applications of Spectral Techniques*, Moraga, C., Ed., Dortmund, 1989, pp. 36–45.

27. Picard, C.F., *Theorie des questionnaires*, Paris: Gauthier-Villars, 1965.

28. *Logic Synthesis and Optimization*, Sasao, T., Ed., Boston: Kluwer, 1993.

29. Sasao, T., AND-EXOR Expressions and Their Optimizations, in *Logic Synthesis and Optimization*, Sasao, T., Ed., Boston: Kluwer, 1993, pp. 287–312.

30. Sasao, T. and Debnath, D., An Exact Minimization Algorithm for Generalized Reed–Muller Expansions, *IEICE Trans. Fundamentals*, 1996, vol. E79-A, no. 12, pp. 2123–2130.

31. Sasao, T., Representations of Logic Functions by Using EXOR Operators, in *Representations of Discrete Functions*, Sasao, T. and Fujita, M., Eds., Boston: Kluwer, 1996, pp. 29–54.

32. Stanković, R.S., A Note on the Relation between Reed–Muller Expansions and Walsh Transform, *IEEE Trans. Electromagnetic Compatibility*, 1982, vol. EMC-24, no. 1, pp. 68–70.

33. Stanković, R.S., Edge-Valued Decision Diagrams Based on Partial Reed–Muller Transforms, *Proc. Reed–Muller Colloquium UK'95*, Bristol, England, 1995, pp. 9-1–9-13.

34. Stanković, R.S. and Moraga, C., Some Remarks about the Relationship among the Reed–Muller, Algebraic and Walsh Transform of Switching Functions, *Yugoslav Conference for ETAN*, Belgrade, Yugoslavia, 1993.

35. Stanković, R.S. and Moraga, C., Edge-Valued Decision Diagrams for Switching Functions Based Upon the Partial Reed–Muller Transforms, *Proc. 2nd Int. Conf. Application of Computer Systems Szczecin*, Poland, 1995, pp. 7–16.

36. Stankovic, R.S., Sasao, T., and Moraga, C., Spectral Transform Decision Diagrams, in *Representations of Discrete Functions*, Sasao, T. and Fujita, M., Ed., Boston: Kluwer, 1996, pp. 55–92.

37. Thayse, A., Davio, M., and Deschamps, J.-P., Optimization of Multiple-Valued Decision Diagrams, *ISMVL-8*, 1978, pp. 171–177.

38. Varma, D. and Trachtenberg, E.A., Efficient Spectral Techniques for Logic Synthesis, in *Logic Synthesis and Optimization*, Sasao, T., Ed., Boston: Kluwer, 1993, pp. 215–232.

This paper was recommended for publication by V.A. Lototskii, a member of the Editorial Board