Some Remarks about Spectral Transform Interpretation of MTBDDs and EVBDDs

Radomir S. Stanković
Braće Taskovića 17/29
18 000 Niš
Yugoslavia

Abstract— In this paper we give a spectral transform interpretation of AND-EXOR representations of switching functions and related decision diagrams in the vector space over $GF(2)$. The consideration is uniformly extended to the Fourier series-like expressions of functions in the complex vector space and the decision diagrams for integer-valued functions. It is shown that the multi-terminal decision diagrams, MTBDDs, and edge-valued decision diagrams, EVBDDs, for integer-valued functions are derived by using the same sets of basic functions already applied for the decision diagrams attached to some AND-EXOR expressions, but considered over the complex field. The algebraic transform decision diagrams, ATDDs, are considered as the integer counterparts of the functional decision diagrams, FDDs, attached to the algebraic transform in the same way as the FDDs are attached to the Reed-Muller expressions. It is shown that the EVBDDs are the ATDDs in different notation.

I. Introduction

It may be said that the motif and interest for the introduction of the decision diagrams, DDs, for integer-valued functions closely relates to the present renewed interest in spectral techniques for logic design [6], reinitiated in [11].

For that reason, in this paper we first give a spectral transforms interpretation of the DDs attached to various AND-EXOR representations of switching functions. Then, we extend this group-theoretic approach to Fourier analysis to provide an uniform interpretation of the multi-terminal binary decision diagrams, MTBDDs, [3] and the edge-valued binary decision diagrams, EVBDDs, [7]. It is shown that EVBDDs are nothing, but a different notation of the DDs attached to the algebraic transform of switching functions [8], [1] which is, further, nothing, but the integer counterpart of the Reed-Muller transform. Thus, the EVBDDs actually are the integer counterparts of the positive polarity Reed-Muller decision diagrams with the expansion procedure associated to branches instead to the nodes.

[1] The algebraic transform decision diagrams, ATDDs, [10].

Fig. 1. Reading of f from the Q-tree.

II. AND-EXOR Expressions

The application of the discrete Fourier-like transforms in the great majority of various tasks in signal processing, switching theory and logic design reduces to the calculation of the matrix relation

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

where $\mathbf{F} = [f(0), \ldots, f(g-1)]^T$ is the truth-vector of a signal f on a finite discrete set of the cardinality g, $\mathbf{Q}$ is a $(g \times g)$ invertible transform matrix, and $\mathbf{S}_f = [S_f(0), \ldots, S_f(g-1)]^T$ is the set of the corresponding Fourier-like spectral coefficients.

If the switching functions are considered as elements of the function space on the finite dyadic group [2] over the Galois field $GF(2)$, then various AND-EXOR expressions, including the complete disjunctive form, can be interpreted as the Fourier series-like expansions in this vector space with respect to different bases. The same interpretation extends to the DDs regarded as the graphical representations attached to some AND-EXOR representations of switching functions [9].

III. Decision Diagrams for Switching Functions

An uniform interpretation of various AND-EXOR expansions, systematized in the Sasao's theorem [9], and the corresponding trees, can be given and their relationship to spectral representations established if we note that in a tree labels at branches in each path from the root node up to a constant node form a product $\varphi_i = a_1 \ldots a_n$, where $a_i \in \{1, x_i, \overline{x}_i\}$. Relationship of the tree representations to the spectral transforms is direct and rather simple. The products φ_i describes a set of functions in terms of products of switching variables. These functions can be understood as columns of a $(2^n \times 2^n)$ transform matrix $\mathbf{Q}$. The constant nodes of a tree representing a function f given by the truth-vector $\mathbf{F}$, contain the values of Fourier-like spectrum $\mathbf{S}_f$ defined by (1).

Example 1 In the case of the Shannon tree the basic functions are described by minterms, i.e., for $n = 3$ they are given by $\varphi_0 = \overline{x}_1\overline{x}_2\overline{x}_3$, $\varphi_1 = \overline{x}_1\overline{x}_2 x_3$, $\varphi_2 = \overline{x}_1 x_2 \overline{x}_3$, $\varphi_3 = \overline{x}_1 x_2 x_3$, $\varphi_4 = x_1\overline{x}_2\overline{x}_3$, $\varphi_5 = x_1\overline{x}_2 x_3$, $\varphi_6 = x_1 x_2 \overline{x}_3$, $\varphi_7 = x_1 x_2 x_3$. Therefore, the used basis is the trivial basis whose elements can be identified as columns of the identity matrix.

For the positive Davio tree the basic functions are the Reed-Muller functions given for $n = 3$ by $\varphi_0 = 1$, $\varphi_1 = x_1$, $\varphi_2 = x_2$, $\varphi_3 = x_2 x_3$, $\varphi_4 = x_1$, $\varphi_5 = x_1 x_3$, $\varphi_6 = x_1 x_2$, $\varphi_7 = x_1 x_2 x_3$, representing the columns of the Reed-Muller matrix.

¿From the above discussion, a direct discrete transform Q is related to the nodes in a tree and, thus, the constant nodes contain the values of the Q-spectrum of f, while the inverse mapping is related to the branches of the DD. Therefore, given a tree, in reading f it represents by using the corresponding reading rule, [3] we actually perform the inverse mapping Q^{-1} and, thus, we get f.

In a tree, if we change the reading rule by the drawing rule and in that way perform the Q-mapping in reading f, we will nutralize the impact of the Q^{-1} related to the branches of the DD, since $QQ^{-1} = I$, and we will read the Q-spectrum of f. The statement is illustrated in Fig.1 and 2.

Threfore, each tree drawn with respect to a transform Q defined though the corresponding basis $\{\varphi_i\}$, representing a function f, can be considered as the Shannon tree representing the Q-spectrum of f.

For example, the positive Davio tree of f is the Shannon tree of the Reed-Muller spectra S_f of f and vice versa.

A reduced ordered decision diagram is obtained from the corresponding tree by using some simplification rules [9]. For example, binary decision diagrams, BDDs, are derived from the Shannon tree by using the following reduction rules

1. Identify two nodes v and v' in the DD, where the sub-DDs rooted in v and v' are isomorphic.

2. Delete a node v whose two outgoing branches point to the same node and connect the incoming branches of the deleted node to the corresponding successor.

Functional decision diagrams, FDDs, are derived from the positive Davio tree by using another appropriate set of

Fig. 2. Reading of S_f of f from the Q-tree.

Fig. 3. ATDD of f from Example 2.

[2] The finite dyadic group of order 2^n, C_{2^n} consists of the set of binary n-tuples $(x_1, \ldots, x_n), x_i \in \{0, 1\}$, under the operation of componentwise modulo 2 addition , usually denoted as EXOR. This group can be represented as the direct product of the basic cyclic groups of order 2, $C_2 = (\{0, 1\}, \oplus)$.

[3] In paractice, we follow the labels at branches.

reduction rules [4]. Compared to the reduced FDDs, the positive Davio tree reduced by the above set of reduction rules produces the quasi-reduced FDDs, QRFDDs. Thus, RFDD are derived form QRFDDs by a further reduction. Since these reduction rules adopted to FDDs can not be used in the reduction of the BDDs [4], we will consider in what follows the DDs derived from the corresponding trees by using the above mentioned reduction rules for an uniform interpretation. These rules are applicable to other DDs in the same sense as noted for FDDs.

The above given statement applies to the DDs derived from the corresponding trees through above mentioned reduction rules. [4]

IV. DDs for Integer-valued Functions

A. Multi-Terminal Binary Decision Diagrams

With the above given spectral interpretation of the DDs attached to AND-EXOR representations, the extension of the concept of DDs to the representation of integer-valued functions is straightforward. The Walsh spectrum of a switching function is an example of the integer-valued functions on the finite dyadic groups. The trivial basis used with the Shannon tree and, thus, BDDs, is a basis also in the space of integer, or even complex functions on the finite dyadic group, if the values 0 and 1 of switching variables are considered as the integers and the logical AND is formally replaced by the multiplication in the complex field. Thanks to that, the MTBDDs were introduced for the representation of the Walsh spectra as a generalization of the BDDs by permitting the integers as constant nodes [3]. The MTBDDs are considered in [1] under the name algebraic decision diagrams, ADDs.

B. Algebraic Decision Diagrams

The BDDs were used in the calculation of the algebraic transform [5]. Thanks to the above discussion, we will do just the opposite. We will use the algebraic transform to define a class of DDs for the representation of the integer-valued functions including the switching functions as a particular example.

Consider the functions Reed-Muller functions representing the basis for the PPRMs and FPRMs expressions, as the integer-valued functions. This set of functions is a basis in the complex vector space over the finite dyadic groups which permits the definition of the so-called algebraic expressions or the algebraic transform [8].

As in the case of PPRM expressions, the coefficients of these algebraic polynomials are defined by a relation corresponding to (1)

$$\mathbf{A}_f(n) = \mathbf{S}(n)\mathbf{F}, \tag{2}$$

[4] The DDs are the canonical representations.

Fig. 4. EVBDD of f from Example 2.

where

$$\mathbf{S}(n) = \bigotimes_{i=1}^{n} \mathbf{S}_i(1), \quad \mathbf{S}_i(1) = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, \tag{3}$$

since $\mathbf{S}_i(1)$ is the inverse of $\mathbf{R}_i(1)$ over the complex field. From there, it follows that the algebraic tree can be defined thanks to the function expansions

$$f = 1 \cdot f_0 + x_i f_1, \tag{4}$$
$$f = 1 \cdot f_0 + x_i(f_1 - f_0), \tag{5}$$

derived easily from the matrix definition of the algebraic transform.

These expansions should be used as reading and drawing rules for the algebraic DDs, ATDDs, in the same way as that was done above in the case of other DDs. The ATDDs are introduced in [2] from a different point of view under the name binary momnt decision diagrams BMDs. As in the case of MTBDDs, the reduction of ATDDs can be carried out by using the Shannon reduction rules.

Example 2 Fig.3 shows the reduced ATDDs representing the function $f(x_1, x_2, x_3) = 3 - 4x_1 + 4x_1x_2 + x_1x_3 - 2x_2 + x_2x_3$ taken from Example 1 in [7].

The constant nodes contain the algebraic spectra of f given as a vector by $\mathbf{A}_f = [3\ 0\ -2\ 1\ -4\ 1\ 4\ 0]^T$. The ATDD was drawn by using the drawing rule (5). The reading rule is given by (4) and, thus, this ATDD represent the function

$$\begin{aligned} f &= 1 \cdot (1 \cdot (1 \cdot 3 + x_3 \cdot 0) + x_2 \cdot (1 \cdot (-2) + x_3 \cdot 1)) \\ &\quad + x_1(1 \cdot (1 \cdot (-4) + x_3 \cdot 1) + x_2 \cdot (1 \cdot 4 + x_3 \cdot 0)) \\ &\quad = 3 - 2x_2 + x_2x_3 - 4x_1 + x_1x_3 + 4x_1x_2. \end{aligned}$$

The same ATDD represent the algebraic spectrum A_f of f, which can be determined if the drawing rule is used as the reading rule. Thus, if the reading rule is $A_f = 1 \cdot A_{f_0} + x_i(A_{f_1} - A_{f_0})$, the ATDD in Fig.3 represents the function which is the algebraic polynomial representing the algebraic spectra A_f of f

$$\begin{aligned} A_f &= 1\cdot(1\cdot(1\cdot 3 + x_3\cdot(0-3)) \\ &\quad +x_2\cdot(1\cdot(-2)+x_3\cdot(1-(-2)) \\ &\quad -(1\cdot 3 + x_3\cdot(0-3)))) \\ &\quad +(1\cdot(1\cdot(-4)+x_3\cdot(1-(-4))) \\ &\quad +x_2\cdot(1\cdot(-4)+x_3\cdot(1-(-4)) \\ &\quad -(1\cdot 4 + x_3\cdot(0-4))) \\ &\quad -1\cdot(1\cdot 3 + x_3\cdot(0-3)) \\ &\quad +x_2\cdot(1\cdot(-2)+x_3\cdot(1-(-2)) \\ &\quad -(1\cdot 3 + x_3\cdot(0-3)))) \\ &= 3 - 3x_3 - 5x_2 + 6x_2x_3 - 7x_1 + 8x_1x_3 \\ &\quad +13x_1x_2 - 15x_1x_2x_3. \end{aligned}$$

C. Edge-valued Decision Diagrams

As is stated in [7], the EVBDDs are defined by using the algebraic function

$$x(v_l + f_l) + (1-x)(v_r + f_r), \qquad (6)$$

instead the Shannon expansion. The concept will be introduced though the example.

Example 3 Fig. 4 taken form [7] shows the EVBDD for the function $f = 3 - 4x_1 + 4x_1x_2 + x_1x_3 - 2x_2 + x_2x_3$ from Example 2.

Note that, as is defined in [7], the reduction is carried out by using the same set of the reduction rules as with MTBDDs and ATDDs.

The values of constant nodes in EVBDDs are set to 0 and the calculation procedure is not related to the nodes, which are denoted by switching variables, but is transferred to branches (edges) [5] by taking the advantage from the recursive structure of the algebraic transform matrix. That is achieved by attaching the corresponding value v_l to the left outgoing branches, while the v_r is enforced to be 0 to provide the canonical representation as was said in [7]. That can be interpreted as follows.

In the ATDDs defined by using the algebraic expansion rules (4) and (5), as the reading and drawing rules for ATDDs, the constant nodes contain the values of the algebraic 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 branches of the EVBDD by the additive correction xv_l in (6). Note that in [7] the inverse notation is used in which the left outgoing branch corresponds to the logical 1 and,

[5] The terms branches is usually used in the consideration of FFT flow-graphs. Since we are giving the spectral interpretation of DDs, we will continue to use this term instead of the edge.

Fig. 5. Relationship among the nodes of ATDDs and EVBDDs.

Fig. 6. Complete EVBDD of f from Example 2.

respectively, the right outgoing branch corresponds to the logical 0.

Written as $f = xv_1 + f_0 + x_i(f_1 - f_0)$, since $v_r = 0$, the drawing rule (6) for the EVBDDs, equals that of the ATDDs, except for the additive constant xv_1 introduced to transfer the calculation procedure to the branches and to set the values of the constant nodes to zero. More precisely, in each node of the ATDD we are implementing the algebraic transform with respect to a particular variable, which, by definition of the algebraic 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 branch in the EVBDD to achieve the zero at the constant nodes. This is the reason that the value at the right outgoing branch is always determined as $v_r = f_0 - f_0 = 0$, while the value at the left outgoing branch is calculated as $v_l = f_1 - f_0$. Note that the value at the dinging

Fig. 7. Partial algebraic transform matrices

branch 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 related to the branches as that was done in the EVBDDs, an ATDD node shown in Fig.5a) translates into the node at Fig.5b), and further into the EVBDD node in Fig.5c) 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 we show by the shadows in the Fig.5.

The explanation becomes obvious if we consider the calculation of the values v_l at a complete EVBDD.

Example 4 Fig.6 shows the complete EVBDD of the function from Example 1 and the calculations of the values v_l.

Relation to the values of the spectra of the corresponding partial algebraic transforms is shown by using the matrix representations of these transforms given in Fig.7.

Thus, by using the rule $f = x_i v_1 + f_0 + x_i(f_1 - f_0)$, the EVBDD in Fig. 6 represents the function

$$\begin{aligned} f \;=\; & f_{000} + x_1(f_{100} - f_{000}) + x_2(f_{010} - f_{000}) \\ & + x_2x_3(f_{011} - f_{010} - f_{001} + f_{000}) \\ & + x_1(x_2(f_{110} - f_{100}) + x_3(f_{101} - f_{100}) \\ & + x_2x_3(f_{111} - f_{110} - f_{101} + f_{100}) \\ & - x_2(f_{010} - f_{000}) - x_3(f_{001} - f_{000}) \\ & - x_2x_3(f_{011} - f_{010} - f_{001} + f_{000})) \end{aligned}$$

which is, by definition, the algebraic polynomial of f.

As in the case of ATDDs, a given EVBDD of f represents at the same the algebraic 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 ATDDs, the reading rule for determination of the algebraic spectrum A_f from the EVBDD of f can be written formally as $A_f = 1 \cdot 0 + A_{f_0} + x_i \cdot (0 - 0 + A_{f_1} - A_{f_0})$, or $A_f = 1 \cdot 0 + v_0 + x_i \cdot (0 - 0 + v_1)$, since by definition, $v_0 = A_{f_0}$ and $v_1 = A_{f_1} - A_{f_0}$. Note that, as with ATDDs, we use A_f instead of f in the formulation, since we are considering the determination of an integer-valued function which is the algebraic polynomial representing the algebraic spectrum of another integer-valued function.

If we want to continue to follow the analogy to the ATDDs, we can say that the determination procedure goes from the bottom to the top, as in the case of ATDDs, but from the left to the right nodes, for the inverse notation used in [7]. The constant nodes are passed [6] first. It is assumed, from the definition of EVBDDs, that after the nodes at the level i are passed, they becomes 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 branches. Therefore, it may be said that the EVBDDs are nothing, but the ATDDs with the modified notation in order 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 branches. However, it remains to estimate whether the average number of nodes is greater or lower than the number of values attached to branches for functions of a considerable number of variables. As is noted in [7], in functions where the number of distinct terminal nodes is small, the MTBDDs, may be the space more efficient. The following general comment may be given.

The complexity of a MTBDD depends upon the function f, i.e., upon the structure of its truth-vector $\mathbf{F}$. We will denote that as the representation complexity of f, which is determined by the number of different values of f in the field of integers or the complex field, 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 repeatly appears in $\mathbf{F}$ with same period, then the representation complexity of f decrease and the corresponding MTBDD is simpler.

The complexity of ATDDs depends upon the representation complexity of the algebraic spectrum A_f of f in the same way. In that respect the function f from Example 2 is not convenient for the representation by the ATDDs nor MTBDDs.

However, the complexity of EVBDDs depends in the same way upon the complexity of the truth-vectors of the partial algebraic transforms appearing as the inter-

[6] In the ATDDs we should say processed, but here the constant nodes are set to zero by definition and we should simply pass them and calculate with the values at edges.

mediate truth-vectors in the calculation of the algebraic transform of f through the FFT-like algorithms, since the values v_1 in the EVBDDs take the corresponding values of these truth-vectors of the partial algebraic transforms as is shown in Fig.7.

The ATDDs offer a possibility for the optimization by using the negative algebraic expansion $f = 1 \cdot f_0 + \overline{x}_i(f_0 - f_1)$ in the same way as that was done by using the negative Davio expansions in the case of FPRMs representations in the vector space over $GF(2)$.

By using the nodes corresponding to different possible expansion rules in the complex vector space a variety of polynomial representation and the corresponding decision diagrams can be defined.

The ATDDs are a concept defined as a particular example of a general theory originating in the Fourier series representations of signals.

Regarding a particular application, the representation of Walsh spectra of switching functions, the Walsh decision diagrams, WDDs, [10], offer some advantages compared to the both MTBDDs and EVBDDs providing that the $\{0,1\} \rightarrow \{-1,1\}$ coding of f is performed.

Note that the WDDs are defined with respect to the Reed-Muller functions considered over the complex field, as in the case of MTBDDs, EVBDDs and ATDDs, but in the $\{0,1\} \rightarrow \{-1,1\}$ coding. Just that property was the mathematical base permitting the writing of the procedures for the calculation of Walsh spectra from OBDDs and synthesis of OBDDs from the Walsh spectra.

Regarding the mentioned application, it is important to note that there are some restrictions on the sets where the Walsh coefficients can take their values. For example, the Walsh coefficients of the three-variable switching functions can take 9 different values $\{0, \pm 2, \pm 4, \pm 6, \pm 8\}$ [6]. Moreover, there are allowed three different sets of the Walsh spectral coefficients $\{0, \pm 4\}, \{0, \pm 8\}, \{\pm 2, \pm 6\}$, and it is requested that the sum of Walsh coefficients of a function must be equal ± 8 [6]. Therefore, there are at most three different constant nodes in the WDD representing a three-variable switching function and there are no other values to be stored.

V. Closing Remarks

The decision diagrams for integer-valued functions, as for example the MTBDDs, EVBDDs, ATDDs, WDDs, are nothing, but the integer counterparts of the corresponding decision diagrams attached to some AND-EXOR expressions, since 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.

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

References

[1] , Bohar, R.I., Frohm, E.A., Gaona, C.M., Hachtel, G.D., Maci, E., Pardo, A., Somenzi, F., "Algebraic decision diagrams and their applications", *Int. Conf. on CAD*, November 1993, 188-191.

[2] Bryant, R.E., Chen, Y-A., "Verification of arithmetic functions with binary moment decision diagrams", unpublished paper, May 31, 1994, CMU-CS-94-160.

[3] Clarke, E.M., Zhao, X., Fujita, M., Matsunaga, Y., McGeer, R., "Fast Walsh transform computation with Binary Decision Diagram", in: Kebschull, U., Schubert, E., Rosentiel, W., Eds., *Proc. IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expansion in Circuit Design*, 16.-17.9.1993, Hamburg, Germany, 82-85.

[4] Drechsler, R., Sarabi, A., Theobald, M., Becker, B., Perkowski, M.A., "Efficient representation and manipulation of switching functions based on ordered Kronecker functional decision diagrams", *Proc. DAC'94*.

[5] Falkowski, B.J., 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", *Proc. IEEE Int. Symp. on Circuits and Systems ISCAS94*, U.S.A., 1994.

[6] Hurst, S.L., Miller, D.M., Muzio, J.C., *Spectral Techniques in Digital Logic*, Academic Press, Bristol, 1985.

[7] Lai, Y.F., Pedram, M., Vrudhula, S.B.K., "EVBDD-based algorithms for integer linear programming, spectral transformation, and functional decomposition", *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, Vol.13, No.8, 1994, 959-975.

[8] Malyugin, V.D., "On a polynomial realization of cortege of the Boolean functions", *Repts. of the USSR Academy of Sciences*, Vol.265, No.6, 1982.

[9] Sasao, T., Debnath, D., "An exact minimization algorithm for generalized Reed-Muller expansions", *Proc. IEEE Asia-Pacific Conference on Circuits and Systems*, 5.-8.12.1994, Taipei, Taiwan.

[10] Stanković, R.S., Sasao, T., Moraga, C., "Spectral transforms decision diagrams", submitted to the Reed-Muller'95 Workshop.

[11] Yang, J., De Micheli, G., "Spectral transforms for technology mapping", Technical Report CSL-TR-91-498, Stanford University, December 1991.

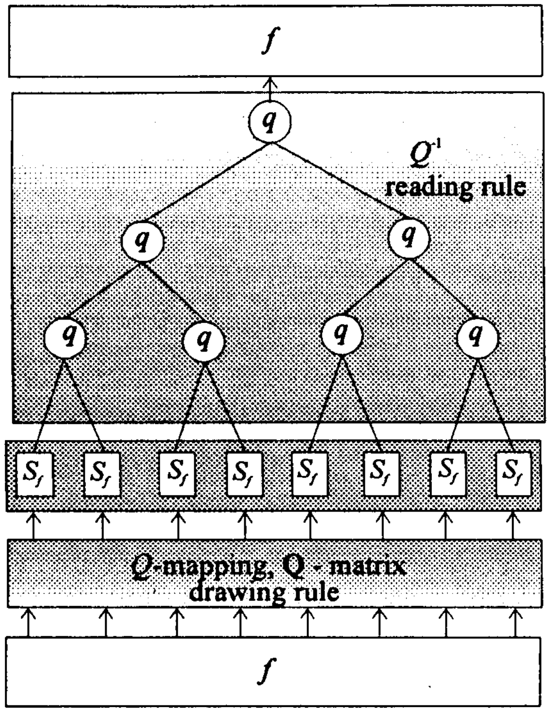

Fig. 1. Reading of f from the Q-tree.

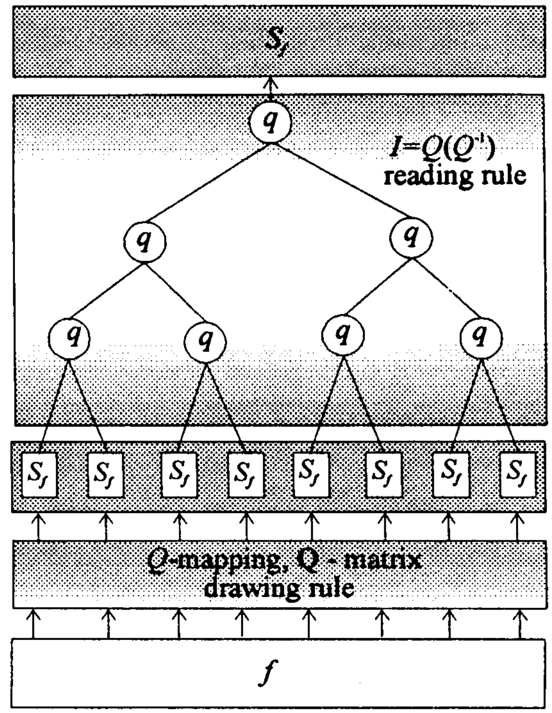

Fig. 2. Reading of S_f of f from the Q-tree.

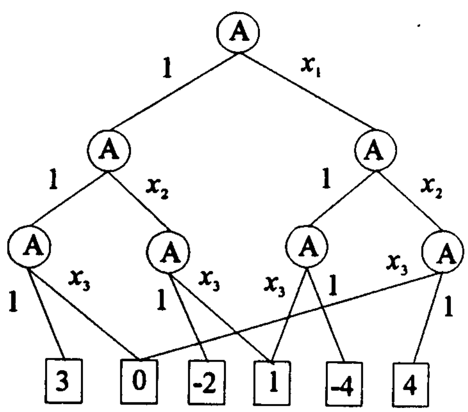

Fig. 3. ATDD of f from Example 2.

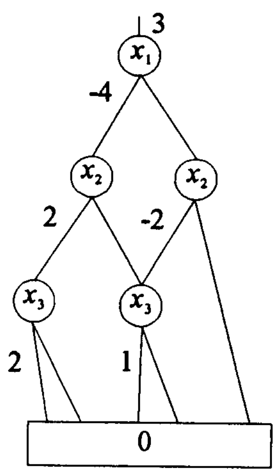

Fig. 4. EVBDD of f from Example 2.

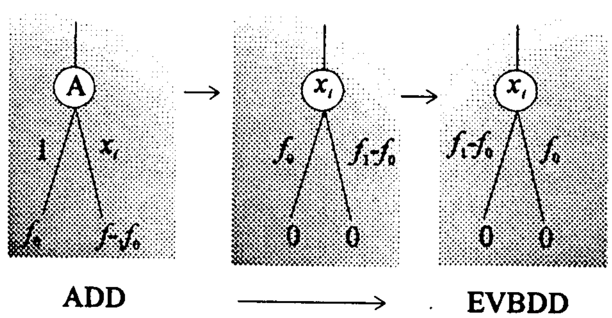

Fig. 5. Relationship among the nodes of ATDDs and EVBDDs.

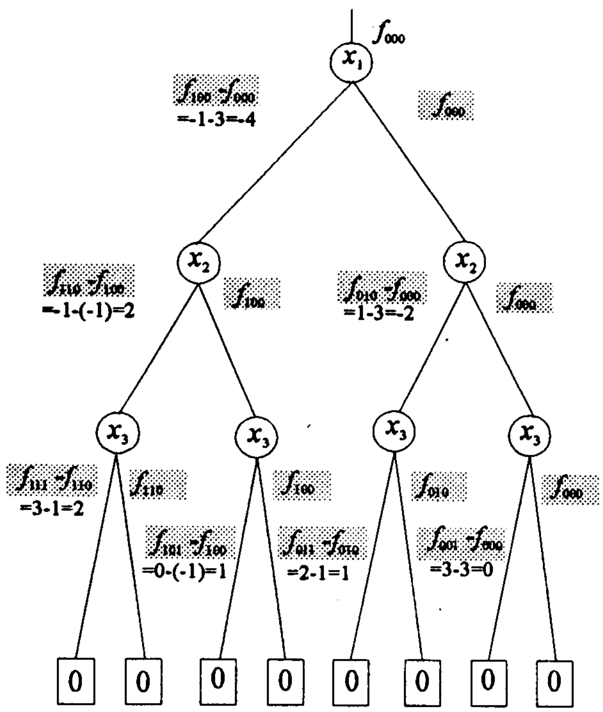

Fig. 6. Complete EVBDD of f from Example 2.

$$\begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ -1&0&0&0&1&0&0&0\\ 0&-1&0&0&0&1&0&0\\ 0&0&-1&0&0&0&1&0\\ 0&0&0&-1&0&0&0&1 \end{bmatrix} \begin{bmatrix} 3\\3\\1\\2\\-4\\3\\0\\1 \end{bmatrix} = \begin{bmatrix} 3\\3\\1\\2\\-4\\-3\\0\\1 \end{bmatrix} \begin{array}{l} f(000)\\ \\ \\ \\ f(100)\text{-}f(000)\\ \\ \\ \\ \end{array}$$

$$\begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ -1&0&1&0&0&0&0&0\\ 0&-1&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&-1&0&1&0\\ 0&0&0&0&0&-1&0&1 \end{bmatrix} \begin{bmatrix} 3\\3\\1\\2\\-4\\3\\0\\1 \end{bmatrix} = \begin{bmatrix} 3\\3\\-2\\-1\\-1\\0\\2\\3 \end{bmatrix} \begin{array}{l} f(000)\\ \\ f(010)\text{-}f(000)\\ \\ f(100)\\ \\ f(110)\text{-}f(100)\\ \\ \end{array}$$

$$\begin{bmatrix} 1&0&0&0&0&0&0&0\\ -1&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&-1&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&-1&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&-1&1 \end{bmatrix} \begin{bmatrix} 3\\3\\1\\2\\-4\\3\\0\\1 \end{bmatrix} = \begin{bmatrix} 3\\0\\1\\1\\-1\\1\\1\\2 \end{bmatrix} \begin{array}{l} f(000)\\ f(001)\text{-}f(000)\\ f(010)\\ f(011)\text{-}f(010)\\ f(100)\\ f(101)\text{-}f(100)\\ f(110)\\ f(111)\text{-}f(110) \end{array}$$

Fig. 7. Partial algebraic transform matrices