FACTA UNIVERSITATIS (NIŠ)
Ser. Math. Inform. 20 (2005), 137–148

REDUCTION OF DECISION DIAGRAMS BY DISJUNCTIVE SPECTRAL TRANSLATION IN THE WALSH–HADAMARD DOMAIN

Milena Stanković and Suzana Stojković

Abstract. In this paper, we define a heuristic method for reduction of the number of nodes in Binary decision diagrams representing the switching function (BDDs). The method uses the disjunctive spectral translation in the Walsh–Hadamard spectral domain to read information about possible reduction of nodes in the original domain. It is based on the following assumption related to the density of switching functions defined as the difference between the number of zero and one values. In a BDD for a given function f, the reduction is possible if in the truth-vector $\mathbf{F}$ for f there are some constant or mutually equal subvectors of orders 2^k, $k < n-1$, where n is the number of variables. It is natural to assume that for a function with high density, it is possible to reduce a larger number of non-terminal nodes, since many equal values, 0 or 1, would produce some constant subvectors in $\mathbf{F}$. Thus, for such functions we can derive a BDD with fewer non-terminal nodes. We determine the transformation of function in original domain from the Walsh–Hadamard spectrum for f. Some experimental results shows that for some classes of functions, the proposed method provides a considerable reduction of nodes in BDDs.

1. Introduction

Binary decision diagrams (BDDs) are an efficient way for representation of discrete functions, which has many applications in CAD systems for VLSI design, signal processing, and related areas. BDDs are especially efficient in representations and calculations with functions of a large number of variables, and thanks to that, they considerably extend applicability of those method whose calculation complexity is a limiting factor in applications. In

Received February 19, 2004.
2000 *Mathematics Subject Classification.* Primary 94C10; Secondary 43A32.

particular, BDDs permit efficient calculation of spectral transforms of large functions [8]. Efficiency of methods based on BDDs considerably depends on the size of BDDs, defined as the number of non-terminal nodes. Thus, reduction of the size of BDDs, which we denote as the minimization of BDDs, is a very important task greatly discussed in many publications, see for example, [1, 6].

The minimization of BDDs is usually performed by reordering variables in the represented function f. However, this is an NP-complete problem, and its exact solution is a very space and time demanding task. Therefore, many heuristic algorithms are proposed for the reduction of the size of BDDs, see for example a discussion of that problem in [2]. However, this is still an open and challenging task.

In this paper, we propose a slightly different method. The BDD for a given f is transferred into a smaller size BDD for another function f' derived from f by some simple operations. Thus derived BDD for f' has reduced size and preserves the complete information about the original function f.

2. BDDs and WDDs

A BDD for a given f is a directed acyclic graph derived by the reduction of the Binary decision tree (BDT) for f.

Example 2.1. Fig. 2.1 shows the BDT for a switching function $f(x_1, x_2, x_3, x_4)$ given by the truth-vector $\mathbf{F} = [1001011010100101]^T$, and Fig. 2.2 shows the corresponding BDD for f.

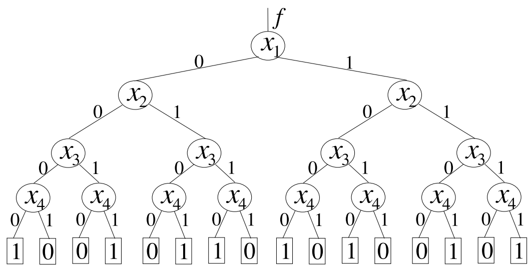

Fig. 2.1: BDT for f in Example 2.1.

Definition 2.1. (Levels) In a BDD for f, the i-th level consists of nodes to which the same variable x_i in f is assigned. The node at the first level is the root node. The level $n+1$ consists of the constant nodes.

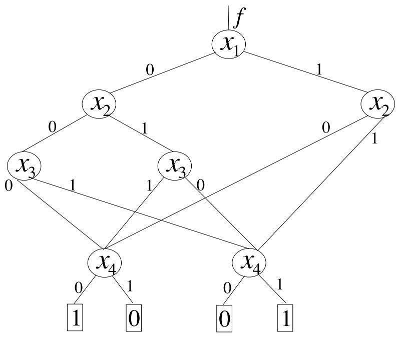

FIG. 2.2: BDD for f in Example 2.1.

Definition 2.2. (Paths in the BDT) A path in the BDT $p = p_1 p_2 \cdots p_{i-1}$ from the root node to a node at the i-th level is a binary sequence of order $i-1$ consisting of labels at the edges in the considered path.

Definition 2.3. (Paths in the BDD) In the BDD, a path from the root node to a node at the i-th level is a sequence $p = p_1 p_2 \cdots p_{i-1}$, where p_k equals the label of the incoming edge to the node at the level k included in the path, and $p_k = -$ if there is no node at the level k, the path is passing down.

Definition 2.4. (Subtrees) *rm Each path $p = p_1 p_2 \cdots p_{i-1}$ determines a subtree rooted at the node at the level i where point this path.*

In the application of the Walsh–Hadamard transform in switching theory, the logic values 0 and 1 of a switching function f are often replaced by integers 1, and -1, respectively.

Definition 2.5. (Walsh–Hadamard spectrum) For a function f given by the truth-vector $\mathbf{F} = [f(0), \ldots, f(2^{n-1})]^T$, in the $(1, -1)$ coding, the Walsh–Hadamard spectrum is defined as the vector $\mathbf{R}(n) = [R(0), \ldots, R(2^n - 1)]$ determined by:

$$\mathbf{R}(n) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}^{\otimes n} \mathbf{F},$$

where $\otimes$ denotes the Kronecker product.

The elements of $\mathbf{R}(n)$ are the Walsh spectral coefficients. They are conveniently denoted by the indices of basic (Rademacher) Walsh functions whose product determines the Walsh function in respect to which a Walsh coefficient is calculated.

Example 2.2. The Walsh–Hadamard spectrum for f in Example 2.1 is given by:

$$\mathbf{R}(4)=[R_0, R_4, R_3, R_{34}, R_2, R_{24}, R_{23}, R_{234}, R_1, R_{14}, R_{13}, R_{134}, R_{12}, R_{124}, R_{123}, R_{1234}]^T$$
$$=[0, 0, 0, 0, 0, -8, 0, -8, 0, 0, 0, 0, 0, 8, 0, 8]^T.$$

For a function f represented by a BDD, we calculate the Walsh-Hadamard spectrum by performing at each node the addition and subtraction of the subtrees rooted at the nodes where point the outgoing edges of the processed node. In this way, the BDD for f is transformed into the WDD for f [7].

Example 2.3. Fig. 2.3 shows WDT for f in Example 2.1 and Fig. 2.4 shows corresponding WDD.

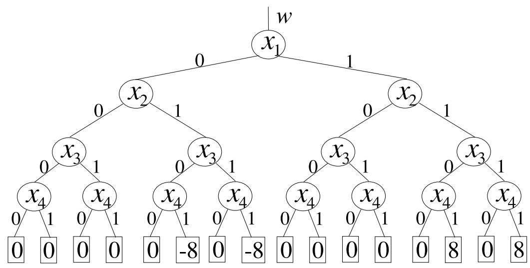

FIG. 2.3: WDT for f in Example 2.1.

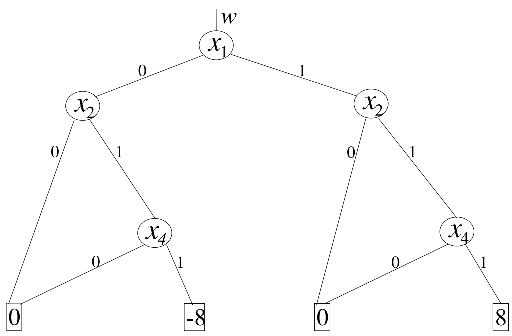

FIG. 2.4: WDD for f in Example 2.1.

3. Spectral Disjoint Translation

For a given f, the value of the Walsh coefficient R_0 is equal to the density of f, i.e., to the difference between the number of zero and one values in

F. It follows, that for functions with the equal number of zeros and ones, $R_0 = 0$. The negative value of R_0 shows that f has greater number of non-zero elements, and conversely for the positive value of R_0. This coefficient is a measure of the correlation between f and the constants 1 and 0. Coefficients of the first order $(R_1, R_2, \ldots R_n)$ shows the correlation of f with the switching variables $x_1, x_2, \ldots, x_n$, while the coefficients of the higher orders $R_{i,j,\ldots k}$ show the correlation between f and the modulo 2 sum (EXOR) of related variables $x_i, x_j, \ldots x_k$.

It is known that some operations over f, denoted as spectral invariant operations, change the order, but not the values of Walsh–Hadamard coefficients [3], [4]. The invariant operations are efficiently used in several tasks in switching theory and applications, as for example in classification and circuit realization of switching functions [3],[4]. In this paper, we use the disjunctive spectral translation which is an spectral invariant operation defined as follows.

Definition 3.1. (Disjunctive spectral translation) Disjunctive spectral translation of switching function f with respect to the index is defined as a mapping $f \to f'$ given by

$$f'(x_1, x_2, \ldots, x_n) = x_i \oplus f(x_1, x_n, \ldots, x_n).$$

In spectral domain, the spectrum $\mathbf{R}_{f'}(n)$ for f' is generated by the permutation of spectral coefficients in $R_f(n)$, such that

$$\begin{array}{lcl} R'_i = R_0 & \text{and} & R'_0 = R_i \\ R'_{ij} = R_j & \text{and} & R'_j = R_{ij}, \end{array}$$

for each index j different from i.

In a WDD, the disjunctive spectral translation with respect to variable x_i permutes the pairs of subtrees determined by the paths $p_1p_2 \cdots p_{i-1}0$ and $p_1p_2 \cdots p_{i-1}1$. In the BDD, the EXOR operation $(x_i \oplus f)$ complements the values in the constant nodes in the all subtrees determined by the paths $p_1p_2 \cdots p_{i-1}1$. It follows that we can generate the BDD for f' from the BDD for f by complementing values in the corresponding constant nodes.

Example 3.1. Fig. 3.1 shows transformation of the BDT for f in Example 1 into the BDT for f' determined by the disjunctive spectral translation with respect to the variable x_2. The transformation consists of the complementation of the subtrees determined by the paths 01 and 11.

Fig. 3.2 shows WDT for the function $f' = x_2 \oplus f$.

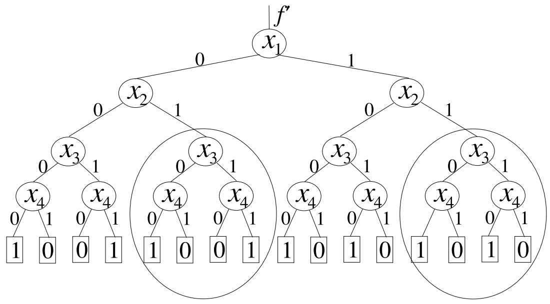

FIG. 3.1: BDT for $f' = x_2 \oplus f$.

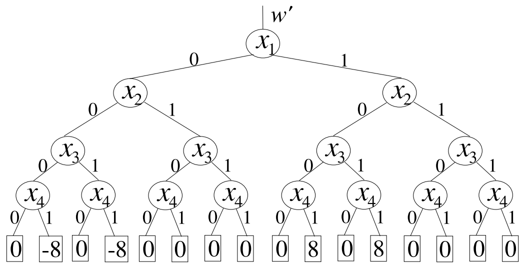

FIG. 3.2: WDT for $f' = x_2 \oplus f$.

It is obvious, that by a repeated spectral disjunctive translation, each Walsh coefficient can be translated into the position of R_0.

4. Minimization of BDDs

In this section, we present a heuristic method for reduction of the size of BDDs, which is based on the following properties and an assumption.

1. **Property 1.** Since R_0 shows the density of f, it follows that for $R_0 > 0$, f has greater number of ones than zeros. The conversely is true for $R_0 < 0$.

2. **Property 2.** The Walsh coefficient with the largest absolute value may be permuted with R_0 by a repeated application of the disjunctive spectral translation performed over the WDD for f.

3. **Property 3.** The number of disjoint spectral translations for this permutation is smaller if in the WDD for f, the path from the root

node to the constant node showing the coefficient with the largest absolute value consists of the smaller number of 1−edges.

4. **Assumption 1.** Given two functions f and g by the BDT(f) and BDT(g). In many cases, if $R_0(f) > R_0(g)$, then we can perform a larger number of node reductions in BDT(f) than in the BDT(g).

From these considerations, the following procedure for minimization of BDDs can be formulated.

Procedure for reduction of BDDs

1° Generate BDD for f.

2° Generate WDD for f by performing the Walsh-Hadamard transform over the BDD for f.

3° In WDD for f, determine the path $p = p_1p_2 \cdots p_{n-1}p_n$ with the minimum number of ones, to the constant node showing the Walsh coefficient with the largest absolute value.

4° For each i, determined by $p_i = 1$ in the path p, perform in the BDD for f the complementation of constant nodes in the all subtrees determined by paths $s_1s_2 \cdots s_{i-1}1$, where $s_1, s_2, \cdots, s_{i-1} \in 0, 1$.

The proposed procedure will be explained and illustrated by the following example.

Example 4.1. We have:

1° For f, in Example 2.1, we generate a BDD. That BDD is shown in Fig. 2.1.

2° We generate the WDD as shown at Fig. 2.3. The corresponding spectrum is:

$$\mathbf{R} = [0, 0, 0, 0, 0, -8, 0, -8, 0, 0, 0, 0, 0, 8, 0, 8]^T.$$

3° In WDD for f, the constant node with the value -8 shows the Walsh coefficient with the largest absolute value. The path $p = 01 - 1$ is the path with the minimum number of 1−edges from the root node to the constant nodes with the value -8.

4° In the path p, the values 1 are at the positions 2 and 4, and thus, we perform the spectral disjoint translation with respect to the variables x_2 and x_4. It follows that in the original domain, we generate the BDDs for functions derived from f by this spectral operation. Fig. 3.1 shows BDD for the function $f' = x_2 \oplus f$. After that translation the spectrum $\mathbf{R}$ is transformed in:

$$\mathbf{R}' = [0, -8, 0, -8, 0, 0, 0, 0, 0, 8, 0, 8, 0, 0, 0, 0]^T.$$

Fig. 3.2 shows BDD for $f'' = x_4 \oplus f' = x_4 \oplus x_2 \oplus f$.
After that translation the spectrum $\mathbf{R}'$ is transformed in:

$$\mathbf{R}'' = [-8, 0, -8, 0, 0, 0, 0, 0, 8, 0, 8, 0, 0, 0, 0, 0]^T.$$

From the BDD for f'', we easily read f by the EXOR with x_2 and x_4. It may be noted that the BDD for f (Fig. 2.1) has 7 non-terminal nodes, while BDD for f'' (Fig. 3.2) has just two non-terminal nodes. Thus, in this example, the proposed algorithm performed a considerable reduction of the size of the BDD.

5. Experimental Results

We have developed a programming package in C consisting of a library of programming modules performing particular steps in the proposed procedure for minimization of BDDs. Then, we have performed a series of experiments intended to estimate the rate of reduction of the BDDs which we can achieve by using this procedure.

Table 5.1 shows some experimental results for optimization of BDDs for some mcnc benchmark functions. For multi-output functions, each output is considered as a separate function, and the Table 5.1 shows the average value of the achieved reduction over all the outputs. In this table, N_{in} denotes the number of input variables, and N_{out} denotes the number of outputs.

Table 5.2 shows the results of the minimization of BDDs for all the outputs in $5xp1$. In this table, N is the number of non-terminal nodes in the BDD before the transformation, p is the path in the WDD for $5xp1$ with the minimum number of ones to the constant node showing the Walsh coefficient with the largest absolute value, N_i is the number of non-constant nodes in the BDD after the i-th spectral translation.

The performed experiments show that for the most of functions, we get a considerable reduction of more than 40% of nodes in the BDD for f. However, for Alu4, the procedure produced the BDD with the increased number of non-terminal nodes. It is believed that this is a consequence of removing some nodes at the upper levels which in the BDD for f may be shared, while permutation by disjoint spectral translation may destroy the corresponding isomorphic subtrees. We believe that a combination of

Table 5.1: Reduction of BDDs by disjoint spectral translation.

Function	N_{in}	N_{out}	Reduction ratio (in %)
2of4	4	1	12.5
5xp1	7	10	41.5
Add2	4	3	44.9
Alu4	14	8	-8.7
Bw	5	28	6.2
Duke2	22	29	0
Rd84	8	4	33.3
Rd53	5	3	25

spectral disjoint translation and linear spectral translation may be useful to overcome this problem. The further investigations will be devoted to such considerations.

6. Function Realization from Minimized BDDs

The proposed method for minimization of BDDs may be useful in function realization from BDDs. It is known that a BDD for f can be directly mapped into a multiplexer network. Since the proposed procedure provides a minimized BDD for f' assigned to f by the disjoint spectral translation in a unique way, the realization of f through the BDD for f' is possible. The price for using BDD for f' instead the BDD for f is small a consists of additional EXOR circuits to get f from f'. However, the saving of multiplexer modules may be considerable. It is equal to achieved reduction of the number of nodes in the BDD for f and f'. The same consideration applies to the small dept circuit synthesis from BDDs [5]. The method will be illustrated by the following example.

Example 6.1. Fig. 6. shows the multiplexer network for realization of f in Example 2.1 derived from the BDD for f. Fig. 6. shows the network for realization of f derived from the BDD for f''. The effects of the minimization are obvious.

Table 5.2: Reduction of BDD for $5xp1$

output	N	p	N_1	N_2	N_3
1	14	0000110	11	11	
2	22	0000011	18	19	
3	23	1000101	23	16	16
4	16	1100010	14	14	11
5	11	0110001	8	8	6
6	9	1011000	5	3	3
7	5	-10-000	2		
8	3	–11000	1	0	
9	1	—1000	0		
10	9	0000000			

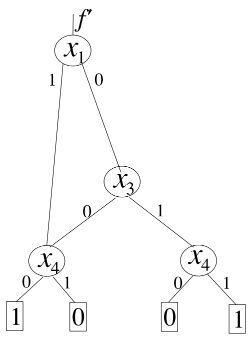

FIG. 6.1: BDD for $f' = x_2 \oplus f$.

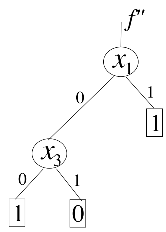

FIG. 6.2: BDD for $f'' = x_4 \oplus f'$.

7. Closing Remarks

In this paper, we have proposed a heuristic method for reduction of the size of BDDs by using the disjoint spectral translation in the Walsh–Hadamard domain. A given function is considered in the spectral domain, and we perform a permutation of spectral coefficients such that the Walsh–Hadamard coefficient with the largest absolute value is shifted into the position of the first Walsh–Hadamard coefficient. This permutation in the spectral domain, assigns to f a function f' whose BDD has fewer nodes than the BDD for f. The relation between f and f' is simple and it is expressed in terms of EXOR with switching variables involved into the performed spectral disjoint translation over f. Thus, we can easily determine

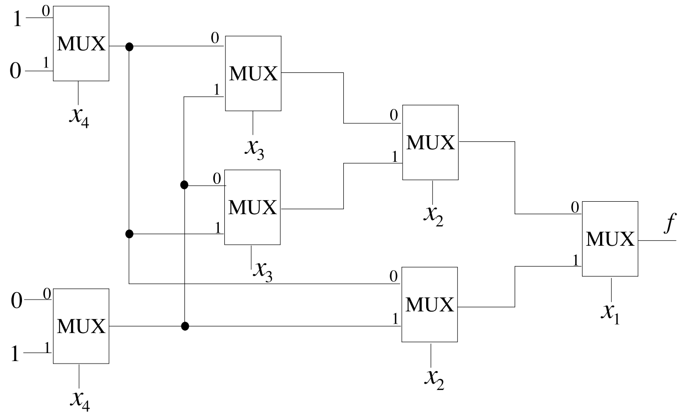

FIG. 6.3: Realization of f by MUX(2×1) from BDD for f.

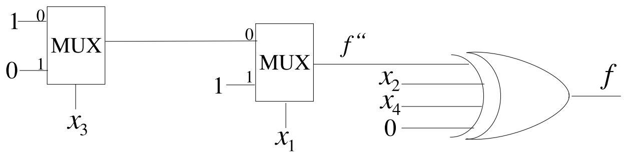

FIG. 6.4: Realization of f form BDD for f''.

f from the BDD for f'.

The experimental results show that the proposed method produces considerable reductions for many functions used in practice. Thus, the procedure may be a good basis for the design methods from BDDs.

REFERENCES

1. R.E. Bryant: *Graph-based algorithms for Boolean functions manipulation.*, IEEE Trans. Computers, Vol. C-35, No. 8 (1986), 667-691.
2. R. Drechsler, B. Becker: *Binary Decision Diagrams, Theory and Implementation.* Kluwer Academic Publishers, 1998.
3. C.R. Edwards: *The application of the Rademacher-Walsh transform to Boolean functions classification and threshold logic synthesis.* IEEE Trans, C-24 (1975), 48-62.
4. S.L. Hurst: *The Logical Processing of Digital Signals.* Crane, Russak and Edwards Arnold, 1978.
5. N. Ishiura: *Synthesis of multi-level logic circuits from binary decision diagrams* SASIMI'92, 74-83.

6. J. Rice, M. Serra, J. Muzio: *The use of autocorrelation coefficients for variable ordering for ROBDDs.* Proc. 4th Int. Workshop on application of Reed-Muller Expression in circuit Design, August 20-21 (1999), 185-196.
7. R.S. Stanković, T. Sasao, C. Moraga: *Spectral transform decision diagrams.* In: Sasao, T., Fujita, M., (eds.), Representations of Discrete Functions, Kluwer (1996), 55-92.
8. R.S. Stanković, M.Stanković, D. Janković: *Spectral Transforms in Switching Theory: Definitions and Calculations.* Nauka, Belgrade, 1998.

Dept. of Computer Science
Faculty of Electronics
University of Niš
Beogradska 14,
18 000 Niš, Serbia
e-mail: mstankovic@elfak.ni.ac.yu
suza@elfak.ni.ac.yu