FACTA UNIVERSITATIS
Series: **Automatic Control and Robotics** Vol. 8, N° 1, 2009, pp. 89 - 97

CALCULATION OF DYADIC CONVOLUTION TROUGH BYNARY DECISION DIAGRAM

UDC 519.115.4   681.3.042

Miloš M. Radmanović

Faculty of Electronic Engineering Niš, Serbia

Abstract. *Calculation of a function's dyadic convolution coefficients is usually done by brute force method of the defining equation. This method requires time exponential in the number of function's inputs to calculate each coefficient. In this paper, a method for more efficient calculation of a function's convolution coefficients is presented. The method is based on Reduced Ordered Shared Binary Decision Diagram (ROSBDD) representations of the functions. Advantages and disadvantages of both methods are experimentally tested and discussed.*

Key words: *Boolean function, dyadic convolution, ROBDD, BDD package*

1. INTRODUCTION

Convolution is an important operation in signal processing and systems theory. Boolean functions are functions for which the inputs and outputs are restricted to the Boolean domain. If a function such as the convolution function is applied to the output vector of the function, the result is a representation of the function in a non-Boolean domain. This representation is generally referred to as the convolution coefficients of the function.

Much work has been performed in applying transforms to Boolean functions in order to achieve a more global view of the function. Transforms such as the Walsh and their applications in digital logic are well researched [4]. Also, there is a lot of software support. There is far less work, however, on the use of other transforms such as the convolution transform.

The convolution coefficients provide a measure of the similarity between two Boolean functions where one function is shifted by a given amount. This is also called cross-correlation function. The convolution coefficients have been used in various areas of digital logic. However, their use has been limited. Calculation of a function's convolution coefficients is usually done by brute force application of the defining equation. This method for computing the convolution coefficients is exponential in the number of inputs to the functions.

Received May 20, 2010

Autocorrelation function is the special case of convolution function. New binary decision diagram (BDD) based methods for computation of autocorrelation coefficients have been introduced by [1], [2] and [3]. Previous work on the analysis of BDD based methods for the calculation of autocorrelation coefficients has shown that decision diagram based methods can outperform standard methods. Inspired by previous work on computation of autocorrelation coefficients, in this paper, we present a method for more efficient calculation of the function's convolution coefficients. The method is based on Reduced Ordered Shared BDD (ROSBDD) representations of the functions. The advantages and disadvantages of both methods are experimentally tested and discussed.

The following section provides background on related research including dyadic convolution coefficients and ROSBDD representation of the function. Section III describes BDD-based methods for calculating convolution coefficients. Section IV outlines the experimental results and presents tests of calculation of a function's convolution coefficients by brute force method and BDD-based method. It should be noted that both methods have been tested to multiple-output functions, but are currently limited to calculation of the first order coefficients. Some concluding remarks end the paper.

2. Background Theory

2.1. Dyadic Convolution Coefficients

The convolution coefficients of a switching function are another representation in the spectral domain. There is no loss of data in the transformation. The convolution coefficients of a function are calculated using the convolution function.

The general convolution (cross-correlation) function between two given switching functions $f(X)$ and $g(X)$ where $X=x_{n-1}x_{n-2}...x_1$ at the distance u is defined as:

$$C(u)=\sum_{v=0}^{2^n-1} f(v)g(v\oplus u) \quad (1)$$

When $f(X) = g(X)$, the resulting equation gives the convolution of a function with itself, translated by u. The resulting coefficients are referred to as the autocorrelation coefficients of the function. The autocorrelation function is defined as:

$$B(u)=\sum_{v=0}^{2^n-1} f(v)f(v\oplus u) \quad (2)$$

For multiple-output functions a second step must be performed to combine the convolution function for each of the individual functions into the total convolution function. The total convolution function [5] where m is the number of outputs, the multiple output function F consists of $f_0 f_1 ... f_{m-}$ and G consists of $g_0 g_1 ... g_m$ is defined as:

$$C(u)=\sum_{i=0}^{m-1} C_i(u)=\sum_{i=0}^{m-1}\sum_{v=0}^{2^n-1} f_i(v)\cdot g_i(v\oplus u) \quad (3)$$

2.2. Reduced Ordered Shared Binary Decision Diagrams

BDDs have become widely used for a variety of CAD applications, including symbolic simulation, verification of combinational logic and verification of sequential circuits.

Binary decision diagrams (BDDs) are a data structure convenient to represent switching functions of a large number of variables [6]. BDDs are derived by the reduction of binary decision trees (BDTs). The reduction is performed by sharing the isomorphic subtrees and deleting the redundant information in the decision trees using the suitably defined reduction rules. The original representation BDTs were presented by Akers [7] and are simply a graphical implementation of the Shannon decomposition of the function. The Shannon decomposition of the function is defined as:

$$f(X) = \overline{x}\, f_0 + x\, f_1 \quad (8)$$

where f_0 is the function f where $x = 0$ and f_1 is the function f where $x = 1$.

Restrictions to create reduced ordered BDDs were added by Bryant [8]. BDT and ROBDD representation of function $f(x_1, x_2, x_3) = x_1 + x_2\overline{x}_3$ are shown in Fig. 1 and Fig 2.

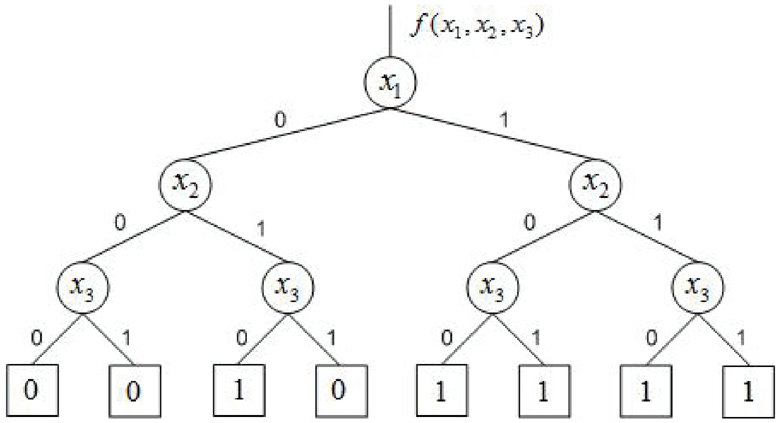

Fig. 1 The BDT representation of function $f(x_1, x_2, x_3) = x_1 + x_2\overline{x}_3$

Multiple-output switching functions are represented by shared ROBDDs (ROSBDDs) [6] having a separate root node for each output. Thus, ROSBDDs are obtained by sharing isomorphic subtrees in ROBDDs for outputs of function, considered as separate particular switching functions. ROSBDD representation of functions $f_1(x_1, x_2, x_3) = x_1 + x_2\overline{x}_3$ and $f_2(x_1, x_2, x_3) = x_1x_2 + x_2\overline{x}_3$ is shown in Fig. 3.

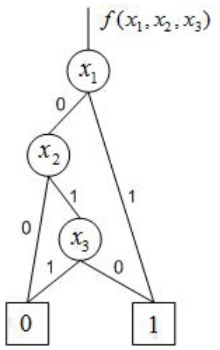

Fig. 2 The ROBDD representation of function $f(x_1,x_2,x_3) = x_1 + x_2\overline{x}_3$

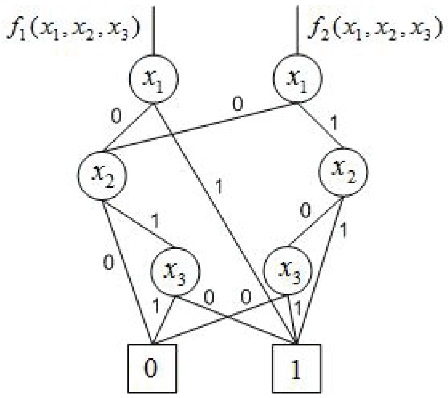

Fig. 3 The ROSBDD representation of functions $f_1(x_1,x_2,x_3) = x_1 + x_2\overline{x}_3$ and $f_2(x_1,x_2,x_3) = x_1x_2 + x_2\overline{x}_3$

3. BDD Based Method For The Calculation of Convolution Coefficients

Using Equation (3) to calculate convolution coefficients for a function requires $O(m2^{2n})$ operation. Implementation of Equation (3) also requires an exponential number of calculations for each coefficient if it is applied to each of the 2^n outputs of the function. More sophisticated BDD-based method of calculating the convolution coefficients makes use of representation that does not require all of the possible input combinations to be described.

In this method, we use the property that the BDD representation of function $f(v \oplus u)$ is equal to the BDD representation of function f with permuted labels at the edges for all the nodes for the variable v_i corresponding to $u_i = 1$, where u_i is the i-th bit in the binary representation for $u = (u_1,...,u_n)$.

ROSBDD representations of functions, Equation (3) and previous BDD property give the following procedure:

1) For each coefficient u, create ROSBDD by taking function f and ROSBDD by taking function g with permuted label at the edges defined by u.
2) Perform multiplication of ROSBDD($f(v)$) and ROSBDD($g(v \oplus u)$) using the standard procedure for functions represented by decision diagram [6] and call the result ROSBDD($f(v)g(v \oplus u)$).
3) Count the minters (terminal nodes) in ROSBDD($f(v)g(v \oplus u)$).

The first step of procedure for calculation of 3-rd convolution coefficient is illustrated in Fig 4.

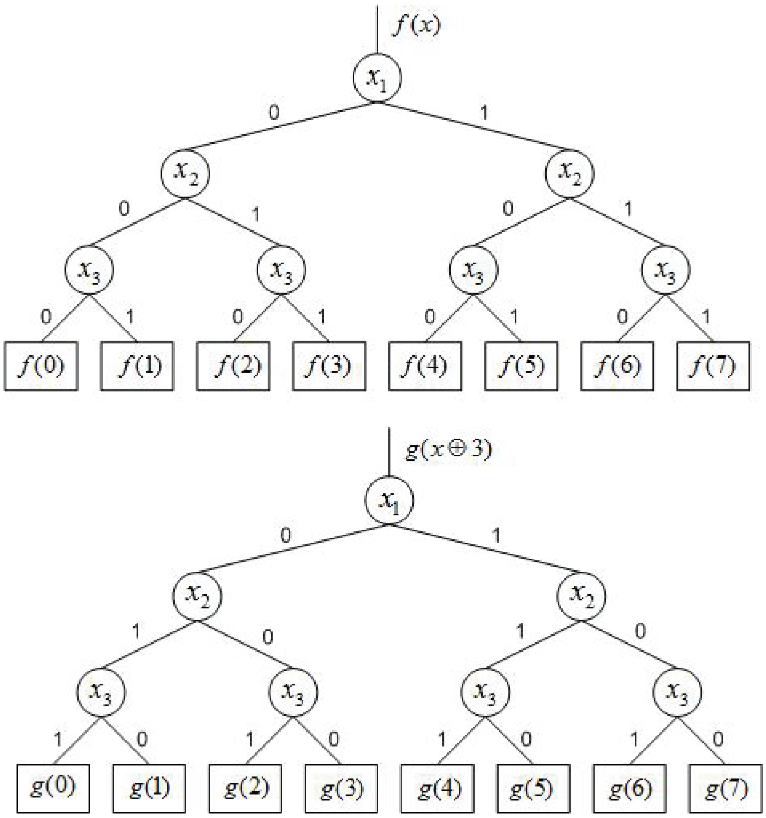

Fig. 4 The first step of BDD based method for the calculation of convolution coefficients

Corresponding labels at the edges should be permuted. For example, $u = (0,1,1)$ and the labels at the edges corresponding to x_2 and x_3 should be permuted.

The second step of procedure for calculation of 3-rd convolution coefficient is illustrated in Fig 5.

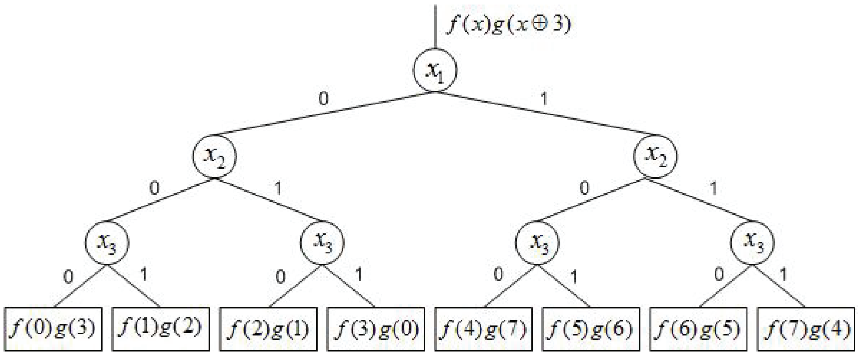

Fig. 5 The second step of BDD based method for the calculation of convolution coefficients

The third step of procedure for calculation of 3-rd convolution coefficient is illustrated in Equation 9.

$$C_{fg}(3) = f(0)g(3) + f(1)g(2) + f(2)g(1) + f(3)g(0) + \\ + f(4)g(7) + f(5)g(6) + f(6)g(5) + f(7)g(4) \quad (9)$$

In the previous figures, we represented *C*(3) by a BDT. In practice, the calculations are always performed over ROBDDs and advantages are taken from the compactness of the ROBDDs compared to BDTs.

An example of calculating the 3-rd convolution coefficient for the functions $f_1(x_1,x_2,x_3) = x_1 + x_2\overline{x}_3$ and $g(x_1,x_2,x_3) = x_1x_2 + x_2\overline{x}_3$ through ROBDDs is shown in the following figure. The sum of all values of the truth vector, that is represented through the ROBDD($f(x)g(x \oplus 3)$) gives the 3-rd convolution coefficient (see Equation 10).

$$C_{fg}(3) = 0 + 0 + 0 + 0 + 1 + 1 + 1 + 0 = 3 \quad (10)$$

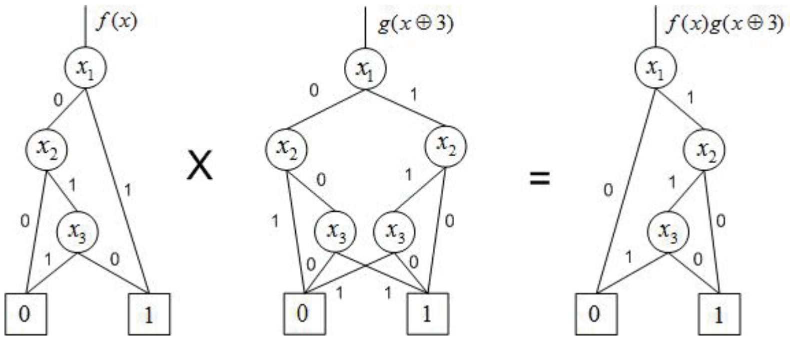

Fig. 6 Calculating the 3-rd convolution coefficient for the functions $f_1(x_1,x_2,x_3) = x_1 + x_2\overline{x}_3$ and $g(x_1,x_2,x_3) = x_1x_2 + x_2\overline{x}_3$ through ROBDDs

This method is convenient for calculation of a separate convolution coefficient. The number of operations using this method to calculate the convolution coefficients for a functions *f* and *g* requires *O*(*m***k***l*) operations, where *m* is the number of outputs of functions f and g, k denotes the number of nodes in ROSBDD(f), and l denotes the number of nodes in ROSBDD(g).

4. EXPERIMENTAL RESULTS

In order to investigate the performance of the BDD-based method, I have developed a tool for calculation of first order convolution coefficients. The brute force method of defining equation was implemented mainly for comparison and verification of BDD-based method.

Both methods were implemented in C++ using MS Windows platform (32-bit). BDD-based method was implemented using BDD package “MyBDD 1.1 Library“ [9]. Both methods were tested on 33 benchmarks [10] and timings as generated on a PC Pentium IV on 2,66 GHz with 4 GB of RAM. The memory usage for all tests was limited to 2 GB. Timing statistics of BDD-based method include building ROSBDDs. All benchmarks were used in the Espresso-mv or pla format [11].

Table 1 gives the complete list of experimental results. All times are given in seconds. Blank entries indicate that the method failed to complete for that benchmark. Reasons for not completing included running out of memory. The 5-th column lists the timings of brute force method of defining equation. The brute force method was failed (ran out of memory) in 22 of 33 cases. Successful benchmarks for that method were completed in lower time than benchmarks for BDD-based method. The 6-th column lists the timings of BDD-based method. All benchmarks for that method were completed within 5 seconds of time.

For comparison, the brute force method failed in all cases where the number of input variables is greater than 22.

5. CONCLUSIONS

I have implemented and compared two methods for computing a multiple-output function’s first-order convolution coefficients. The brute force method of defining equation is not feasible for large functions. This method requires time and memory exponential in the number of function’s inputs to calculate each coefficient. In this paper, a method for more efficient calculation of a function’s convolution coefficients is presented. The method is based on ROSBDD representations of the functions. Besides permitting processing of multi-output functions of a larger number of variables with a restricted memory, binary decision diagrams offer a considerable flexibility in calculations of a single coefficient or subsets of convolution coefficients. Especially, calculations by manipulating with labels at the edges of binary decision diagrams are promising. Further work will be devoted to deeper exploiting these possibilities as well as exploiting different decision diagrams.

Table 1 Timing statistics of brute force method and BDD-based method for computing the first order convolution coefficients

Function name (*f*)	Inputs / outputs / cubes (*f*)	Function name (*g*)	Inputs / outputs / cubes (*g*)	Brute force method [s]	BDD-based method [s]
ex1010	10 / 10 / 1081	rnd_10_10_100	10 / 10 / 100	0.3	0.1
clpl	11 / 5 / 20	rnd_11_5_100	11 / 5 / 100	0.9	0.1
alu1	12 / 8 / 20	rnd_12_8_100	12 / 8 / 100	1.2	0.1
misex3	14 / 14 / 1703	rnd_14_14_100	14 / 14 / 100	1.2	0.1
b12	15 / 9 / 431	rnd_15_9_100	15 / 9 / 100	1.5	0.1
al2	16 / 47 / 103	rnd_16_47_100	16 / 47 / 100	2.9	0.2
table5	17 / 15 / 158	rnd_17_15_100	17 / 15 / 100	2.2	0.2
in2	19 / 10 / 137	rnd_19_10_100	19 / 10 / 100	3.4	0.3
mark1	20 / 31 / 129	rnd_20_31_200	20 / 31 / 200	30.1	0.3
cc	21 / 20 / 45	rnd_21_20_200	21 / 20 / 200	33.1	0.3
duke2	22 / 29 / 87	rnd_22_29_200	22 / 29 / 200	84.2	0.4
cps	24 / 109 / 654	rnd_24_109_200	24 / 109 / 200		0.5
alupla	25 / 5 / 2144	rnd_25_5_200	25 / 5 / 200		0.3
bc0	26 / 11 / 479	rnd_26_11_200	26 / 11 / 200		0.3
x9dn	27 / 7 / 120	rnd_27_7_200	27 / 7 / 200		0.3
c8	28 / 18 / 79	rnd_28_18_200	28 / 18 / 200		0.4
chkn	29 / 7 / 154	rnd_29_7_200	29 / 7 / 200		0.3
exep	30 / 63 / 175	rnd_30_63_400	30 / 63 / 400		0.6
b3	32 / 20 / 234	rnd_32_20_400	32 / 20 / 400		0.5
b4	33 / 23 / 54	rnd_33_23_400	33 / 23 / 400		0.6
in3	35 / 29 / 75	rnd_35_29_400	35 / 29 / 400		0.7
jbp	36 / 57 / 166	rnd_36_57_400	36 / 57 / 400		0.7
signet	39 / 8 / 124	rnd_39_8_400	39 / 8 / 400		0.6
seq	41 / 35 / 336	rnd_41_35_400	41 / 35 / 400		0.8
ti	47 / 72 / 271	rnd_47_72_400	47 / 72 / 400		1.1
ibm	48 / 17 / 173	rnd_48_17_400	48 / 17 / 400		0.8
misg	56 / 23 / 75	rnd_56_23_800	56 / 23 / 800		1.2
e64	65 / 65 / 65	rnd_65_65_1600	65 / 65 / 1600		2.0
x7dn	66 / 15 / 622	rnd_66_15_1600	66 / 15 / 1600		1.3
x2dn	82 / 56 / 112	rnd_82_56_3200	82 / 56 / 3200		2.7
soar	83 / 94 / 529	rnd_83_94_3200	83 / 94 / 3200		2.6
mish	94 / 43 / 91	rnd_94_43_6400	94 / 43 / 3200		3.4
ex4	128 / 28 / 620	rnd_128_28_6400	128 / 28 / 3200		4.8

REFERENCES

1. Stanković, R., S., Bhattacharaya, M., Astola, J., T., *Calulation of Dyadic Autocorrelation through Decision Diagrams*, Proc. European Conf. Circuit Theory and Design, pp. 337-340, Aug. 2001.
2. Rice, J.E., Muzio, J.C., *Methods for Calculating Autocorrelation Coefficients*, Proceedings of the 4th Workshop on Boolean problems, (IWSBP2000), 69-76, 2000.
3. R. Stankovi¶ć and M. Karpovsky, 'Remarks on Calculation of Autocorrelation on Finite Dyadic Groups by Local Transformations of Decision Diagrams", Computer Aided Systems Theory -- EUROCAST 2005.
4. S. L. Hurst, D. M. Miller, and J. C. Muzio, "Spectral Techniques in Digital Logic", Orlando, Florida: Academic Press, Inc., 1985.
5. Karpovsky M.G., Finite Orthogonal Series in the Design of Digital Devices, John Wiley 1976.

6. Sasao, T., Fujita, M., (eds.), *Representation of Discrete Functions*, Kluwer, 1996.
7. Ackers S., Binary Decision Diagrams, IEEE Transaction on Computers, vol. C-27(6), pp. 509-516, 1978.
8. Bryant, R. E., *Graph-based Algorithms for Boolean Function Manipulation* IEEE Transaction on Computers C-35(8), 1986.
9. M. Radmanović, "BDDMinPath - implemented using the MyBDD 1.1 Libary", available at http://stgroup.elfak.edu.rs/home/node/5
10. C. Albrecht, "IWLS 2005 Benchmarks," 2005, available at http://iwls.org/iwls2005/benchmark_presentation.pdf
11. R. Rudell, "Espresso Misc. Reference Manual Pages", University of California, Berkeley, California, USA, available at http://embedded.eecs.berkeley.edu/pubs/downloads/espresso/index.htm

RAČUNANJE DIJADIČKE KONVOLUCIJE PREKO BINARNIH DIJAGRAMA ODLUČIVANJA

Miloš M. Radmanović

Za računanje diadičkih konvolucionih koeficijenata funkcija se obično koristi metoda iscrpljivanja korišćenjem definicije konvolucije. Za računanje jednog koeficijenta, vreme izvršavanja ove metode je eksponencijalno zavisno od broja ulaznih promenljivih funkcije. U ovom radu je predstavljen metod za efikasnije računanje konvucijonih koeficijenata funkcija. Metod je zasnovan na predstavljanju funkcije preko redukovanih uređenih razdeljenih binarnih dijagramima odlučivanja (ROSBDD). Prednosti i nedostaci oba metoda su eksperimentalno testirane i razmotrene.

Ključne reči: *Bulove funkcije, diadička konvolucija, ROBDD, BDD paket.*