Table 3: Results of the Probability Based Heuristic Method for DD Reordering

Circuit Name	Initial Size	Resultant Size	CPU Time
c432.isc	31172	1691	986
c499.isc	53866	34432	248
c880.isc	10348	5816	54
c1908.isc	13934	8762	78
c3540.isc	72858	43564	727
5xp1.pla	74	42	< 1
9sym.pla	25	25	< 1
alu4.pla	1197	567	3
bw.pla	108	101	< 1
con1.pla	18	16	< 1
duke2.pla	973	361	1
misex1.pla	41	37	< 1
misex2.pla	136	96	< 1
misex3.pla	1301	504	3

symmetry detection and decision diagram variable reordering. Although some good results have been obtained thusfar, these research efforts are ongoing.

Acknowledgment

The authors would like to thank Professor D. M. Miller for discussions concerning the probability calculation algorithms using DDs.

References

[1] P. W. Besslich. *Spectral Techniques and Fault Detection, (M. G. Karpovsky, editor)*. Academic Press, Boston, Massachusetts, 1985.

[2] B. Bollig and I. Wegener. Improving Variable Ordering is *np*-Complete. *IEEE Transactions on Computers*, vol. 45, no. 9:993–1002, September 1996.

[3] S. Chakravarty. On the Complexity of Using BDDs for the Synthesis and Analysis of Boolean Circuits. *Proceedings of the 27th Annual Allerton Conference on Communication, Control, and Computing*, pages 730–739, 1989.

[4] C. K. Chow. On the Characterization of Threshold Functions. *IEEE Special Publication S.134*, pages 34–38, 1961.

[5] D. Green. *Modern Logic Design*. Addison-Wesley, Reading, Massachusetts, 1986.

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

[7] S. Panda, F. Somenzi, and B. F. Plesier. Symmetry Detection and Dynamic Variable Ordering of Decision Diagrams. *Proceedings of the International Conference on Computer Aided Design*, pages 628–631, November 1994.

[8] K. P. Parker and E. J. McCluskey. Probabilistic Treatment of General Combinational Networks. *IEEE Transactions on Computers*, vol. c-24:668–670, June 1975.

[9] M. Pedram. Power Minimization in IC Design: Principles and Applications. *ACM Transactions on Design Automation of Electronic Systems*, vol. 1, no. 1, 1996.

[10] R. Rudell. Ordering for Ordered Binary Decision Diagrams. *Proceedings of the International Conference on Computer Aided Design*, pages 42–47, November 1993.

[11] S. C. Seth, L. Pan, and V. D. Agrawal. PREDICT-Probabilistic Estimation of Digital Circuit Testability. *Proceedings of the Fault Tolerant Computing Symposium*, pages 220–225, 1985.

[12] M. A. Thornton. Modified Haar Transform Calculation Using Digital Circuit Output Probabilities. *To Appear International Conference on Information, Communications and Signal Processing*, 1997.

[13] M. A. Thornton and V. S. S. Nair. BDD Based Spectral Approach for Reed-Muller Circuit Realisation. *IEE Proceedings, Computers and Digital Techniques*, vol. 143, no. 2:145–150, 1996.

[14] M. A. Thornton and V. S. S. Nair. Efficient Calculation of Spectral Coefficients and Their Applications. *IEEE Trans. on CAD/ICAS*, vol. 14, no. 11:1328–1341, November 1995.

been developed. The tool utilizes an OBDD representation of the logic function as input and iteratively decomposes the OBDD as the resulting logic circuit is constructed. Although a single spectral coefficient may be computed in a reasonable amount of CPU time, an entire spectral description of a circuit requires an exponential number of spectral coefficients (2^n), for this reason a subset of the Walsh coefficients, the first order spectral coefficients referred to as the Chow parameters [4] are computed only. Depending on the Chow parameter magnitudes, a series of heuristic rules are invoked that govern the OBDD decomposition/logic realization step. Table 2 contains data regarding the synthesis of some common benchmark circuits and a comparison using the *misII* tool from Berkeley with the "rugged script".

Table 2: Comparison of Spectral Based Heuristic Method with *misII*

Circuit Name	Output Name	Number Inputs	*mis*II Size	Spectral Size
$5xp1$	*output* 1	7	14	25
$9sym$	*output* 1	9	262	194
$alu4$	*output* 2	14	21	13
bw	*output* 1	5	65	14
$con1$	*output* 1	6	8	11
$rd53$	*output* 1	5	18	9
$rd73$	*output* 1	7	114	125
$rd84$	*output* 1	8	98	253
$vg2$	*output* 2	25	35	46
$apex1$	*output* 12	45	96	38
$misex3c$	*output* 1	14	34	34
$xor5$	*output* 1	5	4	4
$clip$	*output* 4	9	81	170

5 Boolean Function Variable Symmetry

Symmetry detection in Boolean functions is important because it may lead to simpler circuits that implement the function and is also useful in finding good variable orderings for DDs [7]. General symmetries may be detected by means of a transformation approach that checks for function invariance under all possible transformations. Unfortunately the number of such tests can be exponentially large reducing the practicallity of the approach.

Fortunately, the number of variables over which to search for symmetry can be reduced (sometimes substantially so) by using Theorem 1.

Theorem 1 *A necessary condition for symmetry under a circular permutation of k variables is:*
$\wp\{x_i \cdot f\} = \wp\{x_j \cdot f\} \forall x_i, x_j \in \{x_1, x_2, \ldots x_k\}$

6 Variable Reordering in Decision Diagrams

A proper variable ordering is crucial for the usefulness of DDs since poor orderings can result in an structure that is huge in size. The problem of finding the optimal ordering is NP-complete [2]. Fortunately, a suboptimal ordering will suffice in many cases which provides the motivation for the development of heuristic methods to determine "good" variable orderings.

The use of "sifting" algorithms as proposed in [10] is a popular method for reordering a given DD. The sifting technique finds the best ordering of n^2 different possible orderings from the entire set of $n!$ orderings. Our method is to modify the DD variable ordering prior to the application of the sifting algorithm. Therefore, we essentially change the subset of ordering solutions over shich the sifting algorithm will search. We utilize probability measures to arrive at the initial ordering. By sorting the variables in ascending order based on the values of $\wp\{C_{x_i}(f)\}$ where the consensus function, $C_{x_i}(f) = f_{\overline{x}_i} \cdot f_{x_i}$, is based on the Shannon co-factors of the function, f, has yielded the following experimental results given in Table 3. These computations were performed using negative edge-attributed SBDDs on a 100 MHz SPARCstation 20 with 160 MB of RAM.

7 Conclusion

Boolean function output probabilities can be computed through a simple traversal of a decision diagram structure. Therefore, the probability information information is applied to several design and analysis problems in the combinational logic realm. Specifically, research results were surveyed for the problems of spectra computation, logic synthesis, variable

$$W_f(0) = 2^n(1 - 2\wp\{f\}) \tag{2}$$

$$W_f(f_c) = 2^n(4\wp\{f \cdot f_c\} - 1) + W_f(0) \tag{3}$$

$$W_f(f_c) = 2^n(3 - 4\wp\{f + f_c\}) - W_f(0) \tag{4}$$

3.2 Reed-Muller Spectrum

The Reed-Muller family of spectra consist of 2^n distinct transformations. These are generally classified according to a *polarity number*. The polarity number is used to indicate if a literal is present in complemented or non-complemented form in the generalized Reed-Muller Boolean algebraic expression. In terms of the Reed-Muller spectra, the polarity number can be considered to uniquely define the transformation matrix. The generalized Reed-Muller spectra have been studied and used extensively in the past. Reference [5] provides in depth background material.

In relating output probabilities to the Reed-Muller spectra considerations must be made due to the fact that the RM spectrum is computed over Galois field 2 and the output probabilities are real quantities in the interval [0,1]. An isomorphic relation is used to address this problem. Like the Walsh spectrum, each of the rows of the RM transformation matrix may be viewed as constituent function output vectors. The constituent functions turn out to be all possible products of literals (with complementation or lack thereof) determined by the polarity number. Therefore, any arbitrary RM spectral coefficient may be computed as given in Equation 5.

$$R_f(f_c) = [2^n\wp\{f \cdot f_c\}](\mathrm{mod}2) \tag{5}$$

3.3 Haar Spectrum

This section will describe how output probabilities can be used to compute the modified Haar spectral coefficients directly. Each of the constituent Boolean functions contain Shannon co-factor terms (with the exception of the first two matrix rows). A mathematical definition of the modified Haar transform can be found in [6].

In order to determine the total number of matching outputs between f and f_c, it is necessary to determine when both simultaneously evaluate to a logic-0 level as well as a logic-1 level. We denote the percentage of the total number of matches of logic-0 between some f and f_c as p_{m0} and likewise for the logic-1 levels, p_{m1}. With this viewpoint, f_c expressions can be constructed that utilize co-factors of the function to be transformed to restrict the range space and to dictate where the relative location of the valid output of the f_c function occurs in the 2^n row vector components. Each of the co-factors is then ANDed with an appropriate co-factor to provide sign information (or alternatively, Boolean constant values) for the matrix row vector components.

The computation can now be viewed as finding the p_{m0} and p_{m1} values. This is true because each individual Haar spectral coefficient can be computed using the relationship in Equation 6 where $p_m = p_{m0} + p_{m1}$.

$$H_k = 2^{n-i}[2^{i+1}p_m - 1] \tag{6}$$

Where n is the dimension of the range space of the function to be transformed, f, and i is the dimension of the range space of a particular Shannon co-factor of f. Table 1 contains symbols for each of the Haar spectral coefficients, H_i, values that indicate the size of the co-factor function range, i, and probability expressions that evaluate whether the function to be transformed and the constituent function simultaneously evaluate to logic-0 (denoted as p_{m0}), or evaluate to logic-1 (denoted as p_{m1}). Note that the expression p_m is used and is defined as $p_m = p_{m0} + p_{m1}$.

Table 1: Relationship of the Haar Spectrum and Equivalence Functions

SYMBOL	i	$n-i$	p_m
H_0	0	3	$\wp\{\overline{f \oplus 0}\} = 1 - \wp\{f\}$
H_1	0	3	$\wp\{\overline{f \oplus x_1}\}$
H_2	1	2	$\wp\{\overline{x}_1 \cdot \overline{f \oplus x_2}\}$
H_3	1	2	$\wp\{x_1 \cdot \overline{f \oplus x_2}\}$
H_4	2	1	$\wp\{\overline{x}_1 \cdot \overline{x}_2 \cdot \overline{f \oplus x_3}\}$
H_5	2	1	$\wp\{\overline{x}_1 \cdot x_2 \cdot \overline{f \oplus x_3}\}$
H_6	2	1	$\wp\{x_1 \cdot \overline{x}_2 \cdot \overline{f \oplus x_3}\}$
H_7	2	1	$\wp\{x_1 \cdot x_2 \cdot \overline{f \oplus x_3}\}$

4 Probability Based Logic Synthesis

Since spectral values are easily computed from output probability values, a prototype combinational logic synthesis tool based on spectral coefficients has

2 Probability Computation Using Decision Diagrams

Circuit output or signal probabilities provide very useful information regarding the behavior of a digital logic circuit. In the past these quantities have been used to evaluate the effectiveness of random testing [8], to compute spectral coefficients [14] [13] [12], and have recently been used quite extensively in the design and analysis of low power circuitry [9].

Several exact and approximate methods for the determination of circuit output probabilities have been developed. Algebraic and circuit topology based methods were proposed in the original paper [8] and later a method using ordered binary decision diagrams (OBDDs) was proposed in [3]. Some of the estimation methods developed include those discussed in [11].

The OBDD based method in [3] can be used to compute the output probabilities in a very efficient manner if the candidate circuit is described using a compact OBDD. It is proven that the asymptotic complexity of this method is $O(N)$ where N is the number of vertices in the OBDD. This methodology is a based upon a breadth-first traversal of the OBDD with intermediate computations representing the probability that an internal node is reached during some arbitrary path traversal from the initial node to a terminal node. The input to this algorithm is the OBDD and the probability distribution functions for the n dependent variables of the function f.

A new decision diagram (DD) based algorithm has recently been developed by D. M. Miller. This technique has been implemented and also has linear complexity but offers some advantages over that in [3]. The new method is based upon a modified depth-first traversal from the terminal vertex to the initial vertex. A unique aspect of this approach is that all internal vertices are assigned subtree probabilities. That is, internally computed values correpsond to the output probability of a subtree as if the particular internal node were an initial node.

3 Spectra Computation Using Probabilities

By using Boolean relations between the function to be transformed, f, and various other constituent functions, f_c, output probabilities may be computed and used to calculate various spectral coefficients through simple algebraic relations. The advantage of this approach is that the spectral coefficients are computed individually allowing complete storage of 2^n spectral coefficients to be avoided.

3.1 Walsh Spectrum

The Walsh spectral values form several different spectra. The chief difference in the various spectra depend only upon the order that coefficients appear [6]. Various common orders include the Hadamard-Walsh, the Walsh, the Paley-Walsh, and, the Rademacher-Walsh spectra [1] [6].

Since the relationships described here apply only to a single Walsh coefficient, the particular coefficient orderings are not important. Each coefficient is described by the particular f_c that corresponds to a transformation matrix row vector regardless of the actual location of the row vector in the matrix. The Walsh coefficient is denoted by $W_f(f_c)$ where f denotes that the spectral coefficient is with respect to function f, and, f_c denotes that the coefficient is dependent on the row vector corresponding to constituent function f_c.

It is generally the case that the Walsh spectral values are computed by replacing all occurrences of logic-1 with the integer, -1, and all occurrences of logic-0 with the integer, +1. The actual calculation of the coefficient is then carried out by using integer arithmetic. In terms of computing an inner-product of a transformation row vector and an output vector of a function to be transformed, it is easy to see that each scalar product to be accumulated is either +1 or -1 in value. Furthermore, a scalar product of -1 will only occur when the functions f and f_c have different output values for the same set of input variable assignments.

If we consider the equivalence function given by the exclusive-NOR operation (XNOR), a function can be formed whose output is logic-1 if and only if both f and f_c are at logic-1, $\overline{f \oplus f_c}$. Furthermore, if the output probability of this function is computed, we have the percentage of outputs where both f and f_c simultaneously output the same value, $\wp\{\overline{f \oplus f_c}\}$. The corresponding Walsh spectral coefficient can then be obtained by scaling the output probability value by 2^n, where n is the number of variables in f. This is given in Equation 1.

$$W_f(f_c) = 2^n[1 - \wp\{f \oplus f_c\}] \tag{1}$$

Using properties of Boolean algebra and elementary probability theory, alternative expressions for the Walsh coefficients may be computed as given in Equations 2, 3 and 4.

Applications of Circuit Probability Computation Using Decision Diagrams *

M. A. Thornton, R. P. Moore

J. C. Cordova

Department of Computer Systems Engineering

Department of Electrical Engineering

University of Arkansas

Fayetteville, AR 72701-1201

Abstract

Digital circuit output probabilities provide meaningful information regarding the behavior of combinational logic circuits. The computation of the probability values can be computationally expensive when Boolean equations or netlists are used. When the circuits are represented by compact decision diagrams (DD), efficient algorithms exist for determining the probability values. The use of the probabilities based on decision diagram representations are examined here. Specifically, research results of investigations into spectra computation, synthesis, symmetry detection and DD reordering are surveyed.

1 Introduction

The circuit output probability for a combinational logic circuit can be defined as the probability that a given output will yield a logical-1 value given the distributions of the inputs. These probability values have proven to be useful in areas such as analysis of random testing and estimation of signal switching for low power circuit design. Our research involves the use of output probabilities in circuit synthesis, function spectra calculation, input symmetry detection, and the reordering of decision diagrams. A survey of our current research results is presented.

Decision Diagrams in their various forms (OBDDs, SBDDs, and Attributed Edge BDDs) have been shown to efficiently represent a large class of Boolean functions due to their compactness. Furthermore, algorithms with polynomial complexity (in terms of DD vertices) exist for the manipulation of decision diagrams. The usefulness of decision diagrams is extended by showing that circuit output probabilities can be calculated directly from the diagram with a reasonable computation time.

The spectrum of a Boolean function provides a unique description and has been used for a variety of analysis and synthesis tasks. However, the calculation of spectral values typically comes with a high cost in terms of computational complexity. Here, we describe a close relationship between spectral coefficients and circuit output probabilities given by simple algebraic relationships. Methods to calculate the Walsh, Reed-Muller, and Haar spectra are described.

Traditional methods for combinational logic synthesis utilize cube lists or PLA type descriptions as input to a rule or heuristic based area or delay minimizer followed by a technology mapping algorithm. We have developed a prototype synthesis system that converts a decision diagram representation into a multi-level circuit suitable for input to more traditional optimizers.

The detection of symmetrical variables in a behavioral description of a digital circuit is very useful for subsequent design and analysis tasks. Particularly, symmetries that are circular in nature and that involve more than two variables can be difficult to detect. In general, there are “n choose k” possible k-circular symmetries for a function of n dependent variables. We show that this upper bound can usually be tightened allowing for fewer possible symmetries to be considered using circuit output probabilities which are obtained from decision diagram representations.

Decision Diagrams, specifically Ordered Binary Decision Diagrams (OBDD) as defined by Bryant, can only efficiently represent Boolean functions if the variables are properly ordered . Thus a portion of our research has focused on choosing proper variable orderings.

*This research was supported by the National Science Foundation under contract MIP-9633085.