A Method for Approximate Equivalence Checking*

Mitch Thornton

Mississippi State University
Mississippi State, Mississippi
mitch@ece.msstate.edu

Rolf Drechsler   Wolfgang Günther

Albert-Ludwigs University
Freiburg, Germany
{drechsle,guenther}@informatik.uni-freiburg.de

Abstract

An approximate equivalence checking method is developed based on the use of partial Haar spectral diagrams (HSDs). Partial HSDs are defined and used to represent a subset of the Haar spectral coefficients for two functions. Due to the uniqueness properties of the Haar transform, a necessary condition for equivalence is that the individual coefficients must have the same value. The probability that two functions are equivalent is then computed based on the number of observed, same-valued, Haar coefficients. The method described here can be useful for the case where two candidate functions require extreme amounts of computational resources for exact equivalence checking. For simplicity, the technique is explained for the binary case first and extensions to Multiple Valued Logic *(MVL) are shown afterwards. Experimental results are provided to validate the effectiveness of this approach.*

1. Introduction

The equivalence checking problem for two logic functions of n variables, $f(\underline{X})$ and $g(\underline{Y})$, is addressed in this work. Here, we assume that the correspondence between the vectors of variables, $\underline{X}$ and $\underline{Y}$ is known. Although this problem is easily solved when f and g can be completely represented in *Binary Decision Diagram* (BDD) or *Multi-Valued Decision Diagram* (MDD) form, problems can arise for some functions whose corresponding *Decision Diagram* (DD) representations are too large. Thus, we are motivated to formulate a technique for equivalence checking based on partial representations of f and g. The incorporation of the Haar spectral coefficients in our approach allows for further information about the two candidate functions to be exploited.

This problem has applications in logic synthesis and is also of concern in verification systems where two representations of a function are compared [2, 3, 10, 13]. Two abstractions of a circuit resulting from different optimization phases of a logic synthesis system (e.g. $f(\underline{X})$ and $g(\underline{Y})$) may need to be checked to determine if $f(\underline{X}) = g(\underline{Y})$. This is applicable for methods that express state machines as BDDs or MDDs as well as for the verification of purely combinational logic.

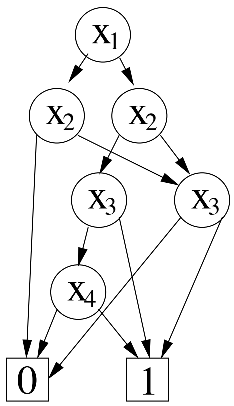
In many cases, this problem can be easily solved by building a BDD [1, 4] or an MDD [12], respectively, representing f and g according to a common variable order. When this is possible, the determination of equivalence is accomplished by simply comparing two pointer values. However, some classes of functions result in DDs with an exponential number of vertices regardless of the variable order.

The technique described here allows for the equivalence checking problem to be formulated in terms of a subset of Haar spectral coefficients [7, 8]. Given a set of Haar spectral coefficients, we examine the probability that $f(\underline{X}) = g(\underline{Y})$. This allows the equivalence checking problem to be iteratively refined in terms of possible error by accounting for the existence of more matching coefficients. Thus, techniques that provide subsets of Haar spectral coefficients [5, 6, 14] for representations of f and g can be used for non-tautology checking of the $f \oplus g$ function. A similar approach using an arithmetic transform and a decision diagram structure known as an *Interleaved BDD* (IBDD) has also been proposed [9]. The technique described here differs due to the fact that we utilize partial HSDs versus IBDDs allowing us to make use of the multi-resolution, modified Haar wavelet transform [7, 8] rather than the arithmetic transform (note the term 'modified' is used to indicate that the basis functions are normalized as is described in [8]). This allows for the advantage of partially representing the functions under consideration and to obtain the Haar spectral coefficients directly from a traversal of the HSD without performing additional spectral computations. Furthermore, the multi-resolution nature of the Haar transform offers advantages in the probability calculations since higher ordered coefficients can represent disjoint portions of the function of interest.

In this approach, we adapt the method reported in [6] that allows the Haar spectral coefficients to be represented as a HSD with the concept of the partial BDD as given in [11]. This allows for a partial function representation to be used for quickly computing subsets of Haar spectral coefficients avoiding problems that may arise for functions that result in very large DDs when represented in their fully specified form. Once the subsets of Haar spectral coefficients are found to be equivalent for two candidate functions, f and g, we compute the probability that f and g are equivalent. If any two same-ordered Haar spectral coefficients are found that have different values, we can declare that $f \neq g$ and halt the process.

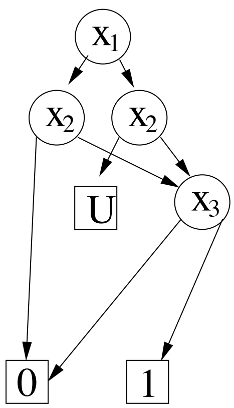
A discussion of the background of partial decision diagrams and HSDs is reviewed followed by a section on the mathematical basis of our technique. The mathematical basis includes a review of relevant aspects of the Haar transform and contains the derivations for the probability computations followed by a simple binary example. Then extension to MVL is discussed. We present the results of some preliminary experiments that indicate the effectiveness of using matching Haar coefficients for statistical verification. Finally, a section containing conclusions is given.

*This work was supported in part by NSF grants CCR-9633085, SBE-9815371 and



	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	0	1	1	1
10	0	1	1	1

Figure 1. Complete BDD



	00	01	11	10
00	0	0	0	U
01	0	0	0	U
11	0	1	1	U
10	0	1	1	U

Figure 2. First incomplete BDD

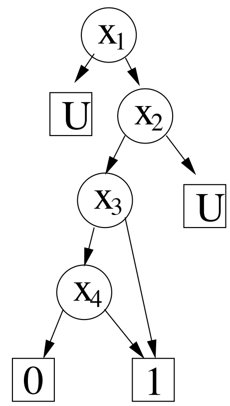
2. Decision diagrams

For simplicity of the presentation, without loss of generality we restrict ourselves to the binary case first. Extensions to the MVL case are straightforward by considering MDDs instead of BDDs.

Boolean variables can assume values from $\mathbf{B} := \{0, 1\}$. In the following, we consider Boolean functions $f : \mathbf{B}^n \rightarrow \mathbf{B}^m$ over the variables specified by the vector $\underline{X} = (x_1, \ldots, x_n)$. As is well-known, each Boolean function $f : \mathbf{B}^n \rightarrow \mathbf{B}$ can be represented by a *Binary Decision Diagram* (BDD), which is a directed acyclic graph where a Shannon decomposition is carried out in each node. In the following discussion, only reduced, ordered BDDs as defined in [1] are considered and for briefness these graphs are referred to as BDDs.

2.1. Incomplete construction

As long as symbolic simulation can be carried out completely, the verification process succeeds. But problems arise if BDDs do not fit in the main memory of a computer. This might be due to several reasons. The first (and simplest) reason is that a "bad" variable ordering has been chosen. In the past, several techniques have been proposed for BDD minimization (for an overview see [4]). Furthermore, the ordering in which the operands are combined during the creation of a BDD is very important.



	00	01	11	10
00	U	U	U	0
01	U	U	U	1
11	U	U	U	1
10	U	U	U	1

Figure 3. Second incomplete BDD

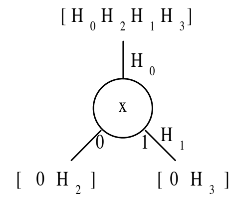


Figure 4. Non-terminal in HSD

2.2. Haar spectral diagrams

In [6], a directed graph referred to as a *Haar Spectral Diagram* (HSD) is defined that represents the Haar spectrum of a Boolean function. HSDs are isomorphic to BDDs (with the exception that all BDD terminal vertices are "mapped" to a common HSD terminal vertex). This allows the BDD representation of a function to double as a representation of the Haar spectrum with extra memory storage required only in the form of an additional edge-attribute value. The additional storage is needed because all 1-edges in the HSD have a Haar spectral coefficient as an attribute.

The enabling observation for defining the HSD is that the Haar transformation matrix can be expressed in terms of Kronecker products if the natural order of the coefficients is permuted. The n-dimensional transformation matrix that produces the coefficients in the permuted order, T^n, can be represented as a sum of matrices denoted as A^n and D^n as given in Equation 1.

$$T^n = A^n + D^n \tag{1}$$

By using this observation and viewing a non-terminal node of a BDD as pointing to two disjoint subfunctions, we can represent the spectrum of the subfunctions (the subfunction spectra are actually scaled by a constant in this case) as two different portions of the entire vector due to T^n. Figure 4 is similar to the diagram originally appearing in [6] and illustrates this relationship. Using these observations, it is possible to represent the Haar spectrum of a function by annotating all 1-edges of the graph (and the pointer to the initial node) with Haar spectral coefficients.

3. Mathematical basis and derivation

In this section, the notation used throughout the remainder of the paper is defined and relations between probabilistic events and Haar spectral coefficients are derived.

3.1. Notation

The following notation is used:

- $H_i(f)$ represents the individual i^{th} Haar spectral coefficient of the Boolean function, $f(\underline{X})$, where $\underline{H}^T = (H_0, H_1, \ldots, H_{2^n-1})$.
- $P[A]$ is the discrete probability that some event, A, occurs and $P[f]$ is the output probability of a Boolean function, f, which is the likelihood that $f = 1$ given the distribution of the dependent variables in $\underline{X}$.
- S_i is the event that $H_i(f) = H_i(g)$, that is, the i^{th} Haar spectral coefficients of f and g are equal in value.
- E is the event that $f(\underline{X}) = g(\underline{Y})$, that is, the functions f and g are functionally equivalent.

3.2. Haar spectrum

This section will summarize the ideas about how output probabilities can be used to compute the modified Haar spectral coefficients directly as given in [14]. In the approach of [14], it is shown that a "composite function", f_c, may be created in BDD form and that through BDD based manipulations of f_c, a Haar spectral coefficient may be computed.

In order to determine the total number of matching values between f and a row-function, 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 a row-function as p_{m0} and likewise for the logic-1 levels, p_{m1}. With this viewpoint, the composite 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.

Given these observations, we see that the k^{th} modified Haar spectral coefficient can be calculated as:

$$H_k = 2^{n-i}[2(p_{m0} + p_{m1}) - 1] \tag{2}$$

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. The result of Equation 2 reduces the computation of a single modified Haar spectral coefficient to that of finding matching percentages of identical similar outputs of f and a transformation matrix row-function. This can be accomplished by applying the output probability computation algorithm to a BDD representation of the f_c functions. Using the result of Baye's theorem, the co-factor output probabilities can be computed by ANDing various cubes with the original function f and dividing the result by the output probability of the cube itself, which is a constant.

Example 1 *The divisor for the p_{m0} and p_{m1} expressions, $P[\overline{x}_1 x_2]$ is a constant equal to $\frac{1}{2^i}$ and thus may be factored out resulting in Equation 2 being rewritten as:*

$$H_k = 2^{n-i}[2^{i+1}(p_{m0} + p_{m1}) - 1] \tag{3}$$

Since the Boolean expressions $f \cdot \overline{x}_1 \cdot x_2 \cdot x_3$ and $\overline{f} \cdot \overline{x}_1 \cdot x_2 \cdot \overline{x}_3$ are disjoint, the overall probability may be computed as the sum of the individual probabilities, or alternatively, as the probability of the inclusive-OR of the functions. This is true because it is easy to see that $P[g + h] = P[g] + P[h]$ for g and h that are covered by disjoint cube sets. Combining the Boolean arguments and simplifying:

$$f \cdot \overline{x}_1 \cdot x_2 \cdot x_3 + \overline{f} \cdot \overline{x}_1 \cdot x_2 \cdot \overline{x}_3 = \overline{x}_1 x_2 (\overline{x_3 \oplus f}) \tag{4}$$

Therefore, we can rewrite Equation 3 as:

$$H_k = 2^{n-i}[2^{i+1} P[\overline{x}_1 x_2 (\overline{x_3 \oplus f})] - 1] \tag{5}$$

The manipulations used in Example 1 may be applied to all of the modified Haar spectrum coefficients. This leads to the interesting result that the modified Haar coefficients depend on the set of $n + 1$ Boolean relations, $\{\overline{f \oplus 0}, \overline{f \oplus x_1}, \overline{f \oplus x_2}, \cdots, \overline{f \oplus x_n}\}$, which describe the equivalence of a particular dependent variable, x_i, and the function to be transformed, f. We refer to this set of functions as the *characteristic equivalence relations*. Higher ordered coefficients are based on disjoint partitions of the range space of these equivalence functions. The partitioning is accomplished by ANDing the equivalence functions with various cubes of other dependent variables of f referred to as the *characteristic cubes*. The specific co-factor that p_m is computed from is given by the inherent order of the dependent variables describing f.

3.3. Probabilistic equivalence checking

By the definition of event E and the assumption that all functions of n variables are equally likely to arise (uniform distribution), it is easy to see that:

$$P[E] = \frac{1}{2^{2^n}} \tag{6}$$

Since the Modified Haar spectrum for a given fully specified Boolean function is unique [8], $P[S_i|E] = 1$ also holds. This may be generalized for the occurrence of any subset of q events, $\{S_i\}$, to that shown in Equation 7.

$$P[\bigcap_{i=1}^{q} Si|E] = 1 \tag{7}$$

Also we see that $P[S_i]$ is the ratio of all possible functions that yield the coefficient, $H_i(f)$, divided by the total population of 2^{2^n}. We define a counting function, $k(H_i)$, that is integer valued and yields the number of fully specified Boolean functions for which the i^{th} Haar spectral coefficient is H_i. Thus we can express this relationship as shown in Equation 8.

$$P[S_i] = \frac{k(H_i)}{2^{2^n}} \tag{8}$$

From probability theory, we know that Equation 9 holds.

$$P[E \bigcap S_i] = P[S_i|E]P[E] = P[E|S_i]P[S_i] \tag{9}$$

Using the relationships in Equations 9, 7 and 6, we see that the conditional probability becomes:

$$P[E|S_i] = \frac{P[E]}{P[S_i]} = \frac{1}{k(H_i)} \tag{10}$$

In general, for any subset of events, $\{S_i\}$, we have the expression as given in Equation 11.

$$P[E|\bigcap_{i=1}^{q} S_i] = \frac{P[E\bigcap(\bigcap_{i=1}^{q} S_i)]}{P[\bigcap_{i=1}^{q} S_i]}$$

$$= \left(\frac{1}{2^{2^n}}\right)\left(\frac{1}{P[\bigcap_{i=1}^{q} S_i]}\right) = \frac{1}{2^{2^n} P[\bigcap_{i=1}^{q} S_i]} \qquad (11)$$

Equation 11 is the governing expression for the probabilistic equivalence checking technique described in this paper. We see that given a subset of matching Haar spectral coefficients for two functions, f and g, (or alternatively, a subset of events, $\{S_i\}$), the probability that f and g are indeed equivalent may be computed. By obtaining the information that a new event S_i has occurred, we may update the value $P[\bigcap_{i=1}^{q} S_i]$ thereby increasing the value $P[E|\bigcap_{i=1}^{q} S_i]$.

3.4. Relation of Haar coefficients to probabilistic events

This section will derive the relationship between the probabilistic events, S_i, and their dependence upon the corresponding Haar spectral coefficients, $H_i(f)$ and $H_i(g)$. The Haar spectral coefficients may be obtained through the use of any efficient method such as those in [5, 6, 14].

For this probabilistic scheme to be practically useful, we need to determine the joint distribution, $P[\bigcap_{i}^{j<2^n-1} S_i]$, as a function of the corresponding subset of Haar spectral coefficients. We first consider the simple case of determining a function for $P[S_i]$ that depends on the single Haar spectral coefficient, H_i. For a single matching coefficient, we are interested in finding, $P[E|S_i]$. Since it is known that $P[E \cap S_i] = P[S_i]P[E|S_i]$, we can express the conditional probability as given in Equation 12 since $P[S_i] \neq 0$.

$$P[E|S_i] = \frac{P[E \cap S_i]}{P[S_i]} \qquad (12)$$

The numerator of Equation 12 is the percentage of functions f and g that have a common Haar coefficient, H_i. Since all equivalent functions have the same Haar spectra by the uniqueness property of the transform, we see that $P[E \cap S_i] = 1/2^{2^n}$. The denominator of Equation 12 is the percentage of functions that have a common H_i value. In general, many different functions can have common H_i values. For example, 6 out of 16 possible functions of $n = 2$ variables have $H_0 = 0$. Based on the definition of the counting function, $k(H_i)$, we can then express $P[S_i] = k(H_i)/2^{2^n}$ and Equation 12 is rewritten as Equation 13.

$$P[E|S_i] = \frac{1}{k(H_i)} \qquad (13)$$

The relationship between the characteristic equivalence functions and the Haar spectral coefficients is established in the following results.

Lemma 1 *Two Boolean functions, $f(x_1, x_2, \ldots, x_n)$ and $g(x_1, x_2, \ldots, x_n)$ can not be equivalent if it is true that $P[\overline{f \oplus x_i}] \neq P[\overline{g \oplus x_i}]$.*

Corollary 1 *Two co-factors about the same cube of $\overline{f \oplus x_i}$ and $\overline{g \oplus x_i}$ have identical output probabilities.*

Table 1. All Possible Boolean Functions for $n = 2$ and their Haar Spectra

Function, f				H_0	H_1	H_2	H_3
0	0	0	0	4	0	0	0
0	0	0	1	2	2	0	2
0	0	1	0	2	2	0	-2
0	0	1	1	0	4	0	0
0	1	0	0	2	-2	2	0
0	1	0	1	0	0	2	2
0	1	1	0	0	0	2	-2
0	1	1	1	-2	2	2	0
1	0	0	0	2	-2	-2	0
1	0	0	1	0	0	-2	2
1	0	1	0	0	0	-2	-2
1	0	1	1	-2	2	-2	0
1	1	0	0	0	-4	0	0
1	1	0	1	-2	-2	0	2
1	1	1	0	-2	-2	0	-2
1	1	1	1	-4	0	0	0

Not all events, $\{S_i\}$, are statistically independent. As an example, H_0 and H_1 are dependent since an intersection of the co-factors of the characteristic equivalence functions of H_0 and H_1 exists and is non-null. In order to find the value $P[S_1 \bigcap S_0]$, we generalize our definition of the counting function to $k(H_0, H_1)$ which will denote the number of possible Boolean functions that may have both H_0 and H_1 as Haar spectral coefficients. Given this quantity, we may then express the desired joint probability as given in Equation 14.

$$P[S_0 \bigcap S_1] = \frac{k(H_0, H_1)}{2^{2^n}} \qquad (14)$$

In general, we have Equation 15 resulting in Equation 16.

$$P[\bigcap_{i=0}^{q} S_i] = \frac{k(H_i, H_{i+1}, \ldots, H_q)}{2^{2^n}} \qquad (15)$$

$$P[E|\bigcap_{i=0}^{q} S_i] = \frac{1}{2^{2^n} P[\bigcap_{i=1}^{q} S_i]} = \frac{1}{k(H_i, H_{i+1}, \ldots, H_q)} \qquad (16)$$

To compute this joint probability distribution, we must have some information concerning the dependent relationship between individual $k(H_i)$ and $k(H_m)$ values.

4. Example calculation

As an example, consider Table 1 which contains the Haar spectral vectors for all possible functions of $n = 2$ variables. We will assume that we are dealing with two functions, $f(x_1, x_2)$ and $g(x_1, x_2)$ such that f and g are equivalent to the function represented in the third row of Table 1. Thus, the corresponding Haar spectral vector is $\underline{H}^T(f) = \underline{H}^T(g) = (H_0, H_1, H_2, H_3) = (2, 2, 0, -2)$. Figure 5 contains the Karnaugh maps and corresponding partial and complete BDDs for the function f (or g). Note that the BDDs are also interpreted as HSDs with the $1 - edges$ having an attribute equal to a Haar spectral coefficient value. The coefficient attributes are shown on the HSD/BDDs with an "*" indicating that the exact coefficient could not be computed. From the center partial HSD/BDD, we see that

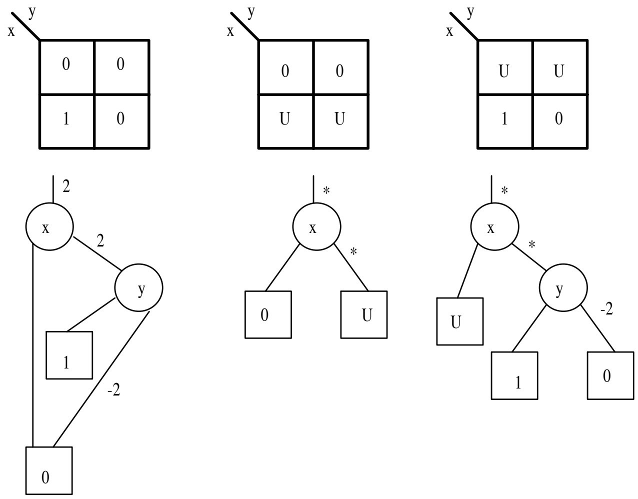


Figure 5. Karnaugh Maps and HSD/BDDs of Complete and Partial Functions

$H_2 = 0$ and from the rightmost partial HSD/BDD we see that $H_3 = -2$.

From the partial BDDs, it is seen that only two Haar spectral coefficients can be obtained, H_2 and H_3. This is due to the fact that H_0 and H_1 require a completely specified HSD since the corresponding transform matrix rows have no 0-valued entries. For more practical cases with much larger values of n, we obtain a larger fraction of the total number of Haar coefficients than the 50% obtained from this small example.

Using the previously derived equations, we have $k(H_2) = 8$, $k(H_3) = 4$ and $k(H_2, H_3) = 2$. These values result in the probability values $P[E|S_2] = \frac{1}{8}$, $P[E|S_3] = \frac{1}{4}$ and $P[E|(S_2 \cap S_3)] = \frac{1}{2}$. Furthermore, we note that $P[S_2 \cap S_3] = P[S_2]P[S_3] = (\frac{1}{2})(\frac{1}{4}) = \frac{1}{8}$ in this case since S_2 and S_3 are statistically independent. The independence arose from the fact that the two partial BDDs represent disjoint segments of the range space of the function. If this is ensured during the construction of all partial BDDs, the joint computation of $k(H_2, H_3)$ may be avoided and $P[E|(S_2 \cap S_3)]$ may be computed as given in Equation 17.

$$P[E|(S_2 \cap S_3)] = \frac{1}{2^{2^n} P[S_2 \cap S_3]}$$

$$= \frac{1}{2^{2^n} P[S_2]P[S_3]} = \frac{1}{16\frac{1}{8}} = \frac{1}{2} \qquad (17)$$

This result shows that there are only 2 possible functions out of the population of $2^{2^n} = 16$ that have $H_2 = 0$ and $H_3 = -2$.

5. Extension to MVL networks

The development of the technique has been applied to binary valued logic functions thus far for simplicity in explanation. However these methods are easily extended to MVL through the use of partial *Multi-valued Decision Diagrams* (MDDs) [12] and the use of the Haar transform for discrete valued functions.

The formulas for computation of the equivalence probabilities change in that the appearance of the term 2^n is replaced by p^n for p-valued logic.

6. Experimental results

Some preliminary experiments were formulated to investigate the effectiveness of using Haar spectral coefficients for equivalence checking. Our initial experiments were run to observe the average number of Haar coefficients needed before a mismatch in value was found for two functions known to be slightly different. These results also give an indication of how different errors between two versions of a circuit affect the number of required Haar coefficients for a mismatch to be found.

The initial set-up for this experiment involved choosing a single output from a binary combinational benchmark function and randomly inserting a single inverter in the netlist. Next, HSDs were formed for the circuit with the inverter and without. To ensure the two HSDs did indeed represent different functions, a graphical equivalence checker was used. The experiment consisted of randomly extracting pairs of same-order Haar coefficients from the two representations until two were found that differed in value. For each given circuit error (that is, each given inverter insertion) 1024 trials were made.

Table 2 contains the results for 10 benchmark functions, each with 10 different inverter errors. The column labeled *Inp* contains the number of distinct variables that the function depends on and the row labeled *avg* is the average number of Haar coefficients (over the 1024 trials) that were required before a mismatch occurred. Likewise, the row labeled *dev* contains the standard deviation of the number of required Haar coefficients. It is apparent that the standard deviation is approximately the same value as the mean in all cases. This is a result of the fact that the subset of Haar coefficients was chosen randomly with the assumption that each was equally likely for two designs that are known to differ (ie., a geometric distribution resulted in terms of the average number of coefficients before a mismatch occurred). Although this observation is largely an artifact of our experimental setup, another result is the large range in value of the required number of coefficients in order to detect the differences in the two circuits. As an example, we see that the benchmark *frg1* has differences in the averages that are as great as four orders of magnitude (eg. 70.1 versus 104225.8).

The data presented in Table 3 was computed in order to compare the Haar coefficient matching scheme to random simulations. These results compare the average number of required Haar coefficients to the number of random simulations that must be performed before a difference in the two circuits is detected. The simulations were performed using equally likely, randomly generated test vectors. The averages were formed over the 10 circuit modifications described above with 1024 trials each. In terms of comparing just the number of simulations to required Haar coefficients, we see that each technique is approximately equal since of the 21 benchmark functions in Table 3, 13 required fewer coefficients than random simulations.

7. Conclusion

A method for probabilistically determining the equivalence of two functions has been developed and presented. We have combined the use of two notions; partial DDs [11] and the computation of Haar spectral coefficients using a BDD as a HSD [6]. The probabilistic framework has been

Table 2. Effect of Different Errors on Haar Coefficient Matching

Circuit	Inp		Inverter Error (10 Random Trials)									
c432	36	avg	58.0	24.8	474.1	15.9	58.3	9.6	8.2	8.4	97.5	56.4
		dev	61.3	24.5	471.2	15.4	55.3	9.2	8.2	8.4	97.5	56.4
c499	41	avg	69.3	65.3	61.4	68.3	66.1	165.6	59.2	59.5	61.4	66.4
		dev	70.4	66.6	60.1	66.9	65.3	163.4	58.9	58.9	60.1	61.1
c880	42	avg	669.3	393.4	36.6	160.7	768.6	134.7	69.4	75.5	46.8	74.2
		dev	681.1	385.9	36.7	161.4	763.9	125.9	68.3	72.1	46.3	72.5
c2670	78	avg	29.9	9.2	10.5	128.1	7.5	62.8	9.7	5.5	18.3	117.6
		dev	28.5	9.3	10.0	129.0	7.2	60.8	9.0	5.0	18.2	110.7
cm151a	12	avg	5.3	16.1	6.4	6.5	15.9	372.5	5.4	5.3	110.3	4.2
		dev	4.7	16.3	5.9	6.1	15.8	403.2	5.1	4.7	109.0	3.5
cu	13	avg	22.5	16.1	48.5	80.2	162.6	253.3	182.1	30.7	225.8	11.1
		dev	22.2	15.9	47.6	85.0	162.1	248.6	190.1	31.2	246.0	10.2
misex3	14	avg	326.6	28.1	579.5	337.1	303.1	234.0	47.0	54.5	131.4	79.4
		dev	336.0	27.1	576.0	324.3	287.3	243.1	45.9	54.4	127.2	74.4
frg1	25	avg	39866.3	70.1	471.6	104225.8	1709.9	989.2	3025.0	12890.1	38287.1	1956.8
		dev	38715.4	70.9	456.9	104031.3	1740.4	1018.6	2954.8	12954.8	38180.4	1992.8
too_large	36	avg	636.5	2559.8	104685.6	1169.4	640.6	1302.6	614.6	738.6	711.5	7888.4
		dev	616.7	2667.7	103407.8	1176.7	609.8	1272.8	611.5	731.0	658.5	7798.7
t481	16	avg	810.6	503.9	415.2	53.3	3.1	1164.9	23.0	383.2	646.1	427.1
		dev	760.4	508.1	425.4	52.0	2.6	1088.7	22.4	364.2	674.5	440.8

Table 3. Average Number of Haar Coefficients Before a Mismatch Occurs

Circuit	Inp	Avg Number Coefficients	Avg Number Simulations
9sym-hdl	*9*	*2.7*	*5.9*
c2670.329	78	39.9	29.1
c432.432GAT	36	81.1	43.2
c499.OD31	*41*	*74.2*	*252.4*
c880.880GAT	42	242.9	119.1
cc.l0	7	16.0	3.8
cm150a	21	7373.9	55.1
cm151a.m	12	54.8	22.1
cm162a.r	11	33.5	27.3
cu.v	*13*	*103.3*	*223.5*
dalu.O7	*57*	*3133.2*	*3584.3*
frg1.d0	*25*	*20349.2*	*26958.6*
misex3.l2	*14*	*212.1*	*1586.0*
mux	21	29810.1	93.1
pcler8.q0	*13*	*641.4*	*1567.7*
pm1.c0	*9*	*25.4*	*90.5*
rd53-hdl.out<2>	*5*	*6.8*	*9.8*
t481	*16*	*443.0*	*2214.2*
too_large.n0	*36*	*12094.8*	*94946.1*
x2.p	*10*	*15.9*	*31.3*
z4ml.24	*7*	*7.4*	*32.1*

derived for the equivalence checking problem. An extension to MVL has been outlined.

Preliminary experimental results indicate that this approach may be a viable alternative for equivalence checking of functions that are difficult to represent completely. Our experiments also indicate that this approach may be better in terms of required computational resources as compared to a repeated simulation approach.

References

[1] R.E. Bryant. Graph - based algorithms for Boolean function manipulation. *IEEE Trans. on Comp.*, 35(8):677–691, 1986.

[2] P. Camurati, P. Prinetto, and P. di Torino. Formal verification of hardware correctness: Introduction and survey of current research. *IEEE Computer*, Jul.:8–19, 1988.

[3] S. Devadas, Hi-Keung Tony Ma, and R. Newton. On the verification of sequential machines at differing levels of abstraction. *IEEE Trans. on Comp.*, 7(6):713–722, 1988.

[4] R. Drechsler and B. Becker. *Binary Decision Diagrams - Theory and Implementation*. Kluwer Academic Publishers, 1998.

[5] B.J. Falkowski and C.-C. Chang. Efficient algorithms for forward and inverse transformations between haar spectrum and binary decision diagram. In *International Phoenix Conference on Computers and Communications*, pages 497–503, 1994.

[6] J.P. Hansen and M. Sekine. Decision diagram based techniques for the haar wavelet transform. In *International Conference on Information, Communication & Signal Processing*, pages 59–63, 1997.

[7] S.L. Hurst. The haar transform in digital network synthesis. In *Int'l Symp. on Multi-Valued Logic*, pages 10–18, 1981.

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

[9] J. Jain, J. Bitner, D. Fussell, and J. Abraham. Probabilistic verification of Boolean functions. *Formal Methods in System Design: An International Journal*, 1(1):63–118, 1992.

[10] S. Malik, A.R. Wang, R.K. Brayton, and A.L. Sangiovanni-Vincentelli. Logic verification using binary decision diagrams in a logic synthesis environment. In *Int'l Conf. on CAD*, pages 6–9, 1988.

[11] D.E. Ross, K.M. Butler, R. Kapur, and M.R. Mercer. Fast functional evaluation of candidate OBDD variable ordering. In *European Conf. on Design Automation*, pages 4–9, 1991.

[12] A. Srinivasan, T. Kam, S. Malik, and R.E. Brayton. Algorithms for discrete function manipulation. In *Int'l Conf. on CAD*, pages 92–95, 1990.

[13] V. Stavridou, H. Barringer, and D.A. Edwards. Formal specification and verification of hardware: A comparative case study. In *Design Automation Conf.*, pages 197–204, 1988.

[14] M.A. Thornton. Modified haar transform calculation using digital circuit output probabilities. In *International Conference on Information, Communication & Signal Processing*, pages 52–58, 1997.