Logic Circuit Equivalence Checking Using Haar Spectral Coefficients and Partial BDDs*

M. A. Thornton R. Drechsler W. Günther

Mississippi State University

Mississippi State, Mississippi

mitch@ece.msstate.edu

Albert-Ludwigs-University

Freiburg, Germany

{drechsle,guenther}@informatik.uni-freiburg.de

A probabilistic 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 Haar spectral coefficients for two Boolean functions. The resulting coefficients are then used to compute and to iteratively refine the probability that two functions are equivalent. This problem has applications in both logic synthesis and verification. The method described here can be useful for the case where two candidate functions require extreme amounts of memory for a complete BDD representation. Experimental results are provided to validate the effectiveness of this approach.

*This work was supported in part by NSF grants CCR-9633085, SBE-9815371 and DAAD grant 315/PPP/gü-ab.

1 INTRODUCTION

The equivalence checking problem for two Boolean 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 BDD form, problems can arise for some functions whose corresponding BDD representations are too large. Thus, we have motivation 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 in terms of the library binding stage where a technologically independent sub-function, f, must be "mapped" to a technologically dependent "library cell" represented functionally by g such that $f(\underline{X}) = g(\underline{Y})$ [13]. Typically a subset of g_i cells satisfy this equivalence and the logic synthesis system chooses a specific g_i based on some optimization constraint such as area minimization, shape factor, speed, etc. Determining the appropriate set of "library cells", $\{g_i\}$, can be accomplished via the application of an equivalence checking technique.

The equivalence checking function is also of concern in verification systems where two representations of a function are compared [3] [5] [14] [15] [17]. Two abstractions of a circuit resulting from different optimization phases of a logic synthesis system (eg. $f(\underline{X})$ 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 as well as for the verification of purely combinational logic.

In many cases, this problem can be easily solved by building an *Ordered Binary Decision Diagram* (OBDD) [1] [6] 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 OBDDs with an exponential number of vertices regardless of the variable order [2] [4].

The technique described here allows for the equivalence checking problem to be formulated in terms of a subset of Haar spectral coefficients [10] [11]. 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 [8] [9] [18] for representations of f and g can be used for non-tautology checking. A similar approach using an arithmetic transform and a decision diagram structure known as an *Interleaved BDD* (IBDD) has also been proposed [12]. 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 [10] [11] rather than the arithmetic transform. 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. As is shown in a later section of the paper, this characteristic allows for some of the required marginal probability density functions to be easily computed due to the statistical independence among the higher ordered Haar coefficients. This is the main reason that the Haar spectra was chosen for this application rather than a global transform such as the Walsh.

In this approach, we adapt the method reported in [9] that allows the Haar spectral coefficients to be represented as a HSD with the concept of the partial BDD as given in [16].

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 BDDs 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 BDDs 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. Next, we present a simple example and 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.

2 BINARY DECISION DIAGRAMS

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) [1], which is a directed acyclic graph where a Shannon decomposition

$$f = \overline{x}_i f_{x_i=0} + x_i f_{x_i=1} \quad (1 \leq i \leq n)$$

is carried out in each node.

A BDD is called *ordered* if each variable is encountered at most once on each path from the root to a terminal node and if the variables are encountered in the same order on all

such paths. A BDD is called *reduced* if it does not contain multiple isomorphic sub-graphs or any instances of both edges from a single vertex pointing to the same node. Reduced and ordered BDDs are unique since each distinct Boolean function and a given variable ordering result in a canonical representation.

BDDs are defined analogously for multi-output functions $f : \mathbf{B}^n \rightarrow \mathbf{B}^m$ as for the case of single-output functions: A BDD G_j for each component function f_j $(1 \leq j \leq m)$ is used for the *shared* BDD representation G for f. The order of the variables is fixed over all G_js.

For functions represented by reduced, ordered BDDs, efficient manipulation algorithms may be formulated [1]. In the following discussion, only reduced, ordered BDDs 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 [6]). Furthermore, the ordering in which the operands are combined is very important, as can be seen by the following simple example:

Example 1 Let $\mathcal{F}$ be an AND gate with three inputs f, g and h that occurs during symbolic simulation of a circuit. BDD packages based on recursive synthesis have to compute:

$$(f \cdot g) \cdot h,\ f \cdot (g \cdot h) \text{ or } (f \cdot h) \cdot g$$

The order in which the calculation is performed largely influences the number of nodes that are needed during the computation, e.g. if $f \cdot g$ is computed first, but $h = 0$. In this case the result $f \cdot g$ (which might be large) is computed first even though the results of the AND gate is 0.

Some first steps for finding good orderings involve traversing gates in circuit representations as described in [7]. However, there also exist functions for which the corresponding BDD size becomes exponential (independent of the variable ordering). The most popular example is the multiplier [2].

In [16] an approach based on *partial information* has been proposed: If the BDD size becomes too large some parts can be projected to a new "terminal" node, called U for undefined. The drawback of this method is that the complete functionality of the represented circuit is no longer present and complete verification is not possible.

Fortunately, the resulting partial DD gives enough information to compute at least some Haar coefficients. This fact will allow us to formulate the equivalence checking technique based on matching subsets of Haar coefficients using only the partial BDDs.

To give a better understanding of partial BDDs, including the value U, we consider the function from [16] given by the table in Figure 1. As can be seen, the BDD for this function requires 6 non-terminal nodes (in the following, we fix the variable ordering). We assume that the memory of the BDD package is limited to 4 non-terminal nodes. Thus, complete construction is not possible. However, if two runs are made using partial information and the rest is projected to an undefined value, U (see Figure 2 and 3), we see that the complete function can be obtained using two partial BDDs.

2.2 Haar Spectral Diagrams

In [9], 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 HSD edges corresponding to a BDD edge annotated with a logic-1 value (these edges are referred to as 1-edges in the following) 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.

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

A^n can now be defined by the Kronecker product relation (denoted by the $\otimes$ operator) as:

$$A^n = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes A^{n-1} + \begin{bmatrix} 0 & 0 \\ 1 & -1 \end{bmatrix} \otimes D^{n-1} \tag{2}$$

The initial cases are $A^0 = 0$ and $D^0 = 1$. It is observed that the first row of A^n is all zeros and only the first row of D^n is non-zero, thus the spectral vector due to T^n can be represented by the two vectors due to A^n and D^n separately. Intuitively, it is seen that the

matrix D^n contributes to the computation of the 0^{th} ordered Haar coefficient only while the A^n matrix contributes to all other other coefficients. 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 [9] and illustrates this relationship. The figure shows the relationship between Equation 2 and a node in a HSD.

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.

As an example, consider the Boolean function, $f = x_1x_2 + \overline{x}_3$. Equation 3 gives the Haar spectrum for this function. If the T^3 matrix were used, the resulting permuted spectral vector would become $[H_0, H_4, H_2, H_5, H_1, H_6, H_3, H_7] = [-2, -2, 0, -2, 2, -2, 2, 0]$. The permutations arise due to the constant and same variable ordering characteristic of the HSD. These coefficients can be used as annotation values on the Shannon decision tree representing the example function as shown in Figure 5. As is well known, the corresponding BDD can be formed by removing all isomorphic subgraphs and redundant nodes from the decision tree representation. If these reductions are carried and the edge annotations are retained, the tree in Figure 5 results in the diagram shown in Figure 6.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \\ 0 \\ 2 \\ -2 \\ -2 \\ -2 \\ 0 \end{bmatrix} \tag{3}$$

The entire spectrum can be recovered through traversals of the HSD and using several properties that arise due to the BDD reduction rules. As an example, any 1-edge that is annotated by a negative value must point to either a non-terminal or the logic-0 terminal node in the HSD. Likewise, a positive valued 1-edge attribute implies the reverse of this rule. Furthermore, if a 1-edge attribute is 0-valued, it cannot point to a terminal node since it would be removed by the BDD reduction rules. With knowledge of these properties and a given HSD, the entire Haar spectrum can easily be reconstructed.

Using the example HSD in Figure 6, it is immediately apparent that $H_0 = -2$, $H_1 = 2$ and $H_3 = 2$ since these values are explicit 1-edge attributes. It is also inferred that $H_4 = H_5 = -2$ since all traversals from the x_1 0-edge do not encounter any x_2 node. $H_6 = -2$ by the attribute value on the 1-edge from the x_3 node also since this edge is traversed for the path where $x_1 = 1$, $x_2 = 0$ and $x_3 = 1$. Also, $H_2 = 0$, since the leftmost x_2 node was removed from the Shannon decision tree due to its' redundancy. Likewise, $H_7 = 0$ since the rightmost x_3 node was removed from the Shannon tree due to its' redundancy.

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:

- $f(x_1, x_2, \ldots, x_n)$ represents a fully specified Boolean function of n variables that may also be represented by the vector, $\underline{X} = (x_1, x_2, \ldots, x_n)$.
- $\underline{H}^T$ represents the transpose of the modified Haar spectral coefficient vector representing some function, $f(\underline{X})$.
- $H_i(f)$ represents the individual i^{th} Haar spectral coefficient of the Boolean function, $f(\underline{X})$, where $\underline{H}^T = (H_0, H_1, \ldots, H_{2^n-1})$. H_i is also represented as H_s^o in some of the literature where o is the order of the spectral coefficient and s is the s^{th} Haar function [11].
- $P[A]$ is the discrete probability that some event, A, occurs.
- $P[f]$ 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 [18]. The idea was developed by making observations about the structure of the transformation matrix.

Each transformation matrix row consists of the integer elements -1, +1 and 0. An integer -1 represents the Boolean 1 constant, an integer +1 represents the Boolean 0 constant, and an integer 0 indicates the absence of a Boolean constant. Each row represents a particular modified Haar function, f_c, dependent upon n or fewer variables where n is the number of variables of f, the function to be transformed.

Figure 7 contains the modified Haar transformation matrix for any function of $n = 3$ variables. It is noted that higher ordered coefficients are computed from matrix row functions with a decreasing range space dimension. In fact, this decrease in the dimension of the range space corresponds directly to various Shannon co-factors of the function to be transformed.

The output vector of the function to be transformed generally contains integers with -1 representing logic-1 and +1 representing logic-0. With this viewpoint, we can define the number of matches between a particular transformation matrix row vector as the number of times the row vector and function vector components are simultaneously equal to -1 or +1. Since some of the rows represent functions that are masked by co-factors, the row-function space is less than 2^3 in size and the presence of a 0 value corresponds to the "masked" portion of the function represented by a row.

The presence of co-factors in the Haar constituent functions can be accounted for by using Bayes' theorem to represent these quantities as output probabilities of the AND of the function to be transformed with its respective dependent literals. These functions are shown

to the left of the transformation matrix in Figure 7.

In order to determine the total number of matching outputs 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 (shown to the left of the transformation matrix in Figure 7) 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. Note that the variable ordering used in the construction of the matrix in Figure 7 is $x_1 \prec x_2 \prec x_3$.

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

$$H_k = 2^{n-i}[2(p_{m0} + p_{m1}) - 1] \tag{4}$$

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. If N_m represents the number of times an intermediate product value of $+1$ occurs in the computation of a particular modified Haar spectral coefficient (corresponding to 1×1 and -1×-1 products) and N_{mm} corresponds to the number of times a product value of -1 occurs (corresponding to 1×-1 products), then the k^{th} modified Haar spectral coefficient is given as:

$$H_k = N_m - N_{mm} \tag{5}$$

It is noted that the sum of N_m and N_{mm} must necessarily equal 2^{n-i} where i indicates the

number of variables about which co-factors have been taken. Substituting this observation into Equation 5 yields:

$$H_k = 2N_m - 2^{n-i} \tag{6}$$

We define p_m to be the total percentage of times that a matching output between the f and f_c functions occur, therefore $p_m = 2^{n-i} \times N_m$. Furthermore, $p_m = p_{m0} + p_{m1}$. Substituting these definitions into Equation 6 yields the result as givenin Equation 4

The result of Equation 4 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 an OBDD representation of the f_c functions. Using the result of Bayes' 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.

The following table 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 row function simultaneously evaluate to logic-0 (denoted as p_{m0}), or evaluate to logic-1 (denoted as p_{m1}).

By observing that the p_{m0} and p_{m1} expressions for a given H_k in Table 1 are statistically independent, the individual computations may be combined into a compact form. As an example, consider H_5.

Example 2 *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 4 being rewritten as:

$$H_k = 2^{n-i}[2^{i+1}(p_{m0} + p_{m1}) - 1] \tag{7}$$

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:

$$f \cdot \overline{x}_1 \cdot x_2 \cdot x_3 + \overline{f} \cdot \overline{x}_1 \cdot x_2 \cdot \overline{x}_3 = \overline{x}_1 x_2 (\overline{x_3 \oplus f}) \tag{8}$$

Therefore, we can rewrite Equation 7 as:

$$H_k = 2^{n-i}[2^{i+1} P[\overline{x}_1 x_2 (\overline{x_3 \oplus f})] - 1] \tag{9}$$

□

The manipulations used in Example 2 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.

Table 2 contains the probability functions for an $n = 3$ variable transformation in terms of the characteristic equivalence relations using the variable order $x_1 \prec x_2 \prec x_3$. Using this table, each coefficient can be computed using Equations 10 and 11.

$$H_i = 2^{n-j}[2^{j+1}p_m - 1] \tag{10}$$

$$j = \begin{cases} 0, & i = 0 \\ \lfloor log_2(i) \rfloor, & i > 0 \end{cases} \tag{11}$$

We can also compute the total number of possible different valued coefficients for a particular i (or equivalently, a particular j). We note that the Haar coefficients range in value as given by $\{-2^{n-j}, -2^{n-j}+2, -2^{n-j}+4, \ldots, -2, 0, +2, \ldots, 2^{n-j}-4, 2^{n-j}-2, 2^{n-j}\}$. Thus, the total number of possible different valued coefficients (denoted by N_j) is given in Equation 12.

$$N_j = 2^{n-j} + 1 \tag{12}$$

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:

$$P[E] = \frac{1}{2^{2^n}} \tag{13}$$

Since the Modified Haar spectrum for a given fully specified Boolean function is unique [11], Equation 14 also holds.

$$P[S_i|E] = 1 \tag{14}$$

Equation 14 may be generalized for the occurrence of any subset of q events, $\{S_i\}$, to that shown in Equation 15.

$$P[\bigcap_{i=1}^{q} Si|E] = 1 \tag{15}$$

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 16.

$$P[S_i] = \frac{k(H_i)}{2^{2^n}} \tag{16}$$

From probability theory, we know that Equation 17 holds.

$$P[E \bigcap S_i] = P[S_i|E]P[E] = P[E|S_i]P[S_i] \tag{17}$$

Using the relationships in Equations 17, 15 and 13, we see that the conditional probability becomes:

$$P[E|S_i] = \frac{P[E]}{P[S_i]} = \frac{1}{k(H_i)} \tag{18}$$

In general, for any subset of events, $\{S_i\}$, we have the expression as given in Equation 19.

$$P[E|\bigcap_{i=1}^{q} S_i] = \frac{P[E \bigcap (\bigcap_{i=1}^{q} S_i)]}{P[\bigcap_{i=1}^{q} S_i]} = \left(\frac{1}{2^{2^n}}\right)\left(\frac{1}{P[\bigcap_{i=1}^{q} S_i]}\right) = \frac{1}{2^{2^n} P[\bigcap_{i=1}^{q} S_i]} \qquad (19)$$

Equation 19 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 [8] [9] [18].

We note that given the i^{th} Haar spectral coefficient for a function, f, and a function, g, there appear to be four possibilities as given in Table 3. It is seen that as soon as $H_i(f) \neq H_i(g)$ occurs, it is possible to declare $f \neq g$ and to terminate the process of equivalence checking. When $H_i(f) = H_i(g)$, it is not known whether $f = g$ or $f \neq g$ unless all possible H_i are found to be equivalent. However, it is possible to successively refine the $P[E|\bigcap_{i=1}^{q} S_i]$ value using Equation 19.

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 20 since $P[S_i] \neq 0$.

$$P[E|S_i] = \frac{P[E \cap S_i]}{P[S_i]} \tag{20}$$

The numerator of Equation 20 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 20 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 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 20 is rewritten as Equation 21.

$$P[E|S_i] = \frac{1}{k(H_i)} \tag{21}$$

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 $P[\overline{f \oplus x_i}] \neq P[\overline{g \oplus x_i}]$.*

Proof: Assume the contradiction of the lemma, that is $P[\overline{f \oplus x_i}] \neq P[\overline{g \oplus x_i}]$, but $f = g$. Since $f = g$, then it must be true that $\overline{f \oplus x_i} = \overline{g \oplus x_i}$ and that $P[\overline{f \oplus x_i}] = m_f/2^n$ and $P[\overline{g \oplus x_i}] = m_g/2^n$ where m_f is the number of distinct 1-values in the truth vector of $\overline{f \oplus x_i}$ and m_g is the number of distinct 1-values in the truth vector of $\overline{g \oplus x_i}$. But since $P[f] = P[g]$

and $\overline{f \oplus x_i} = \overline{g \oplus x_i}$, it must be the case that $m_f = m_g$, thus contradicting the assumption 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.*

We denote f_{ci} as the function that is formed as the intersection of some cube and i^{th} characteristic equivalence function. Thus, f_{ci} depends upon all n variables and $p_m = P[f_{ci}]$. The total number of functions with a common H_i value (denoted as $k(H_i)$), can be computed as the total number of different f_{ci} functions that have a common $P[f_{ci}]$ due to Corollary 1. Therefore, $k(H_i)$ is the different number of ways that a function with a range space of size 2^{n-j} can have $2^n p_m$ logic-1 values. This combinatorial quantity must be scaled by a constant to account for the decreasing magnitude of the H_i that are distributed over the entire population of 2^{2^n} functions as i increases. This scaling factor is seen to be $2^{N_0 - N_j}$ where N_i is defined in Equation 12. Given these observations, $k(H_i)$ can be expressed as:

$$k(H_i) = 2^{N_0 - N_j} \left(\begin{array}{c} 2^{n-j} \\ \\ 2^n p_m \end{array} \right) \tag{22}$$

Using the fact that $2^{N_0 - N_j} = 2^{2^n - 2^{n-j}}$ and that $p_m = \frac{H_i + 2^{n-j}}{2^{n+1}}$, we can reduce Equation 22 to Equation 23.

$$k(H_i) = 2^{2^n - 2^{n-j}} \left(\begin{array}{c} 2^{n-j} \\ \\ \frac{H_i + 2^{n-j}}{2} \end{array} \right) \tag{23}$$

Thus, we can rewrite Equation 21 as shown in Equation 24.

$$P[E|S_i] = \left(\frac{1}{2^{2^n-2^{n-j}}}\right)\begin{pmatrix} 2^{n-j} \\ \frac{H_i+2^{n-j}}{2} \end{pmatrix}^{-1} \tag{24}$$

As an example, consider the case where $H_0(f) = H_0(g) = 2$ for $n = 2$ variables. To compute $P[E|S_0]$, we use the relationship in Equation 24 resulting in $P[E|S_0] = \frac{1}{4}$.

To successively improve the $P[E|S_i]$ value, it must be updated with each subsequent event, S_i. For a practical implementation, this means that $P[\bigcap_i^{j<2^n-1} S_i]$ must be computed as a function of the corresponding Haar coefficients, H_i. However, this value cannot be computed as a simple product of individual $P[S_i]$ values since the multiple S_i events are not necessarily statistically independent.

For the general case, we must re-compute $P[\bigcap_{i=1}^{q} S_i]$ for each new event, S_i, in order to update the $P[E|\bigcap_{i=1}^{q} S_i]$ value. Some events are statistically dependent while other subsets are not. Recall that the values, H_i, depend on co-factor functions of various characteristic equivalence functions about some cube. A subset of events, $\{S_i\}$, are all statistically independent if they result from a corresponding subset of matching Haar spectral coefficients, $\{H_i\}$, that are formed based on Shannon co-factors with respect to mutually disjoint cubes. As an example, H_2 and H_3, result from the functions $\overline{x}_1 \cdot \overline{f \oplus x_2}$ and $x_1 \cdot \overline{f \oplus x_2}$ which are disjoint. Thus, $P[S_2 \bigcap S_3] = P[S_2]P[S_3]$ since $\overline{x}_1$ and x_1 are disjoint characteristic cubes.

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 25.

$$P[S_0 \bigcap S_1] = \frac{k(H_0, H_1)}{2^{2^n}} \tag{25}$$

In general, we have Equation 26 resulting in Equation 27.

$$P[\bigcap_{i=0}^{q} S_i] = \frac{k(H_i, H_{i+1}, \ldots, H_q)}{2^{2^n}} \tag{26}$$

$$P[E|\bigcap_{i=0}^{q} S_i] = \frac{1}{2^{2^n} P[\bigcap_{i=1}^{q} S_i]} = \frac{1}{k(H_i, H_{i+1}, \ldots, H_q)} \tag{27}$$

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. As an example of this dependence, we will derive the relationship between H_0 and H_1. For $i = 0, 1$, the integer j is zero valued yielding the relationships as shown in Equations 28 and 29.

$$H_0 = 2^n[2p_{m0} - 1] \tag{28}$$

$$H_1 = 2^n[2p_{m1} - 1] \tag{29}$$

For these two coefficients, we have the output probabilities p_{m0} and p_{m1} that may be expressed as:

$$p_{m0} = P[\overline{0} \cdot \overline{f \oplus 0}] = P[\overline{f}] = \frac{1}{2}\{P[\overline{f}_{\overline{x}_1}] + P[\overline{f}_{x_1}]\} \tag{30}$$

$$p_{m1} = P[\overline{0} \cdot \overline{f \oplus x_1}] = P[\overline{f \oplus x_1}] = \frac{1}{2}\{P[f_{x_1}] + P[\overline{f}_{\overline{x}_1}]\} \tag{31}$$

Thus, the corresponding Haar spectral coefficients become:

$$H_0 = 2^n\{P[\overline{f}_{\overline{x}_1}] + P[\overline{f}_{x_1}] - 1\} \tag{32}$$

$$H_1 = 2^n\{P[\overline{f}_{\overline{x}_1}] + P[f_{x_1}] - 1\} \tag{33}$$

Equating the $P[\overline{f}_{\overline{x}_1}]$ values results in the relationship between H_0 and H_1 as given in Equation 34.

$$H_1 = H_0 + 2^n\{P[f_{x_1}] - P[\overline{f}_{x_1}]\} \tag{34}$$

Using the result of Equations 23 and 34, we have:

$$k(H_1) = \begin{pmatrix} 2^n \\ \frac{H_1+2^n}{2} \end{pmatrix} = \begin{pmatrix} 2^n \\ \frac{H_0+2^n\{P[f_{x_1}]-P[\overline{f}_{x_1}]\}+2^n}{2} \end{pmatrix} = \begin{pmatrix} 2^n \\ \frac{H_0+2^n}{2} + \frac{H_1-H_0}{2} \end{pmatrix} \tag{35}$$

Using the identity relation:

$$\begin{pmatrix} n \\ k \end{pmatrix}\begin{pmatrix} n-k \\ m-k \end{pmatrix} = \begin{pmatrix} n \\ m \end{pmatrix}\begin{pmatrix} m \\ k \end{pmatrix} \tag{36}$$

We rewrite Equation 35 as:

$$\begin{pmatrix} 2^n \\ \frac{H_0+2^n}{2} \end{pmatrix}\begin{pmatrix} \frac{H_0+2^n}{2} \\ \frac{H_1+2^n}{2} \end{pmatrix} = \begin{pmatrix} 2^n - \frac{H_1+2^n}{2} \\ \frac{H_0+2^n}{2} - \frac{H_1+2^n}{2} \end{pmatrix}\begin{pmatrix} 2^n \\ \frac{H_1+2^n}{2} \end{pmatrix} \tag{37}$$

Equation 37 reduces to Equation 38 and we define the resulting quantity as the value $A_{0,1}$.

$$A_{0,1} = \frac{k(H_0)}{k(H_1)} = \begin{pmatrix} 2^n - \frac{H_1+2^n}{2} \\ \frac{H_0+2^n}{2} - \frac{H_1+2^n}{2} \end{pmatrix}\begin{pmatrix} \frac{H_0+2^n}{2} \\ \frac{H_1+2^n}{2} \end{pmatrix}^{-1} \tag{38}$$

Thus, we have the result that the values $k(H_0)$ and $k(H_1)$ are deterministically related as $k(H_0) = A_{0,1}k(H_1)$. Using this fact, we devise a means for computing the desired joint quantity, $k(H_0, H_1)$. We know that for a given H_0 and H_1 to exist for a single function, the corresponding $k(H_0) - A_{0,1}k(H_1) = 0$. Thus, it is possible to check all possible $k(H_1)$ values for $H_1 = \{-2^n, -2^n + 2, \ldots, -2, 0, 2, \ldots, 2^n - 2, 2^n\}$ and where the relationship is satisfied, we increment the value of $k(H_0, H_1)$. This may be expressed in closed form through the use of the unit-step function, $u(\tau)$, as defined Equation 39.

$$u(\tau) = \begin{cases} 1, & \tau = 0 \\ 0, & otherwise \end{cases} \tag{39}$$

Equation 40 then expresses the desired relationship.

$$k(H_0, H_1) = \sum_{i=0}^{2^n+1} u[k(H_0) - A_{0,1}k(H_i)] \tag{40}$$

The upper bound of the summation is the total number of possible different valued H_1 Haar spectral coefficients, 2^n. Although the complexity of this approach is prohibitive for low-ordered coefficients, Equation 40 can be algorithmically stated as follows:

1. Compute $A_{0,1}$ based on the value of the Haar spectral coefficients.

2. Initialize $k(H_0, H_1) = 0$.

3. Loop over all possible $2^n + 1$ values of H_1 and evaluate `if` $(k(H_0) - A_{0,1}k(H_1) == 0)$ `then` $k(H_0, H_1)++$

4. $P[S_0 \bigcap S_1] = \frac{k(H_0,H_1)}{2^{2^n}}$, or, $P[E|(S_0 \bigcap S_1)] = \frac{1}{k(H_0,H_1)}$.

This procedure can be generalized by deriving and computing new $A_{i,j,\ldots,k}$ values each time a new event, S_k, occurs and updating the $P[\bigcap_{i=1}^{q} S_i]$ and hence, the $P[E|\bigcap_{i=1}^{q} S_i]$ values.

In general, this procedure leads to exponential complexity for updating all new k values. However, we note that the multi-resolution nature of the Haar transform allows us to determine subsets of coefficients that are statistically independent, thus avoiding the computation of large joint distributions. This can also be coupled with the construction of the partial BDDs by constraining them to represent mutually disjoint portions of the functions under consideration. We also note that the decreased range space dimension of high-ordered coefficients, H_i, can allow the algorithm to run in reasonable time for those H_i.

4 EXAMPLE CALCULATION

As an example, consider Table 4 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 4. 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 8 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 $H_2 = 0$ 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 41.

$$P[E|(S_2 \cap S_3)] = \frac{1}{2^{2^n} P[S_2 \cap S_3]} = \frac{1}{2^{2^n} P[S_2]P[S_3]} = \frac{1}{16\frac{1}{8}} = \frac{1}{2} \tag{41}$$

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 $H_3 = -2$ since $P[E|S_2] = \frac{1}{8}$ narrows the candidate set of functions to a size of 8 and $P[E|S_3] = \frac{1}{4}$ further reduces the set to 2 functions.

5 EXPERIMENTAL RESULTS

Experiments were formulated to investigate the effectiveness of using Haar spectral coefficients for equivalence checking. These 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. The 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 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, the entire BDD for each was constructed and compared for equivalence. Obviously, if this were always possible, the probabilistic approach would not be needed. 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 5 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 6 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 6, 13 required fewer coefficients than random simulations.

However, we must point out the very important fact that the computational overhead required to obtain a single Haar coefficient is, in general, not equal to that for performing a simulation. Furthermore, the assumption that the Haar coefficients are equally likely to occur also biases these results to some degree since the subset of coefficients resulting from a specific partial HSD will have mutual dependence due to the definition of the transform. Nevertheless, we can conclude that the use of Haar coefficients does appear to yield as much information as random simulations over this sample of benchmark functions. The importance of this result is that schemes based on the computation of Haar coefficients which are more efficient than a netlist simulation can be used to increase the effectiveness of statistical verification in terms of requiring fewer computational resources.

6 CONCLUSION

A method for probabilistically determining the equivalence of two Boolean functions has been developed and presented. We have combined the use of two notions; partial BDDs [16] and the computation of Haar spectral coefficients using a BDD as a HSD [9]. The probabilistic framework has been derived for the equivalence checking problem.

Experimental results indicate that this approach can be a viable alternative for equiva-

lence checking of functions that are difficult to represent completely. The experiments also indicate that this approach can 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] R.E. Bryant. On the complexity of VLSI implementations and graph representations of Boolean functions with application to integer multiplication. *IEEE Trans. on Comp.*, 40:205–213, 1991.

[3] 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.

[4] S. Devadas. Comparing two-level and ordered binary decision diagram representations of logic functions. *IEEE Trans. on CAD*, 12(5):722–723, 1993.

[5] 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.

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

[7] R. Drechsler, A. Hett, and B. Becker. Symbolic simulation using decision diagrams. *Electronic Letters*, 33(8):665–667, 1997.

[8] B.J. Falkowski and C.-C. Chang. Forward and inverse transformations between haar spectra and ordered binary decision diagrams of boolean functions. *IEEE Trans. on Comp.*, 46:1272–1279, 1997.

[9] 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.

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

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

[12] 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.

[13] K. Keutzer. Dagon: Technology binding and local optimization by dag matching. In *Design Automation Conf.*, pages 341–347, 1987.

[14] J.C. Madre and J.-P. Billon. Proving circuit correctness using formal comparison between expected and extracted behaviour. In *Design Automation Conf.*, pages 205–209, 1988.

[15] 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.

[16] 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.

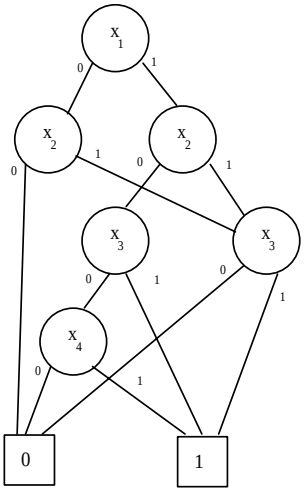
[17] V. Stavridou, H. Barringer, and D.A. Edwards. Formal specification and verification of hardware: A comparative case study. In *Design Automation Conf.*, pages 197–204, 1988.

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

Mitch Thornton is an Associate Professor in the Electrical and Computer Engineering Department at Mississippi State University. He has served on the faculty at the University of Arkansas for four years and has five years of industrial experience. His research interests include system synthesis and verification, computer architecture and arithmetic. He is a member of the IEEE and the IEEE Computer Society.

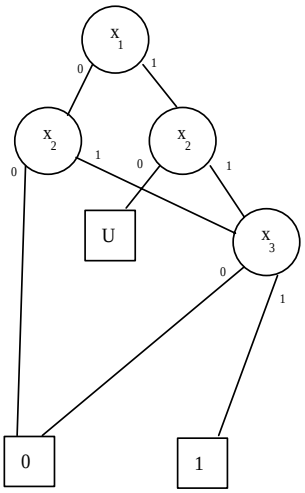
Rolf Drechsler received his diploma and Ph.D. degree in computer science from the J.W. Goethe-University in Frankfurt am Main, Germany, in 1992 and 1995, respectively. He is currently working at the Institute of Computer Science at the Albert-Ludwigs-University of Freiburg im Breisgau, Germany. He published two books with Kluwer Academic Publishers, one on BDD techniques co-authored by Bernd Becker and one on using evolutionary algorithms for VLSI CAD. His research interests include verification, logic synthesis, and evolutionary algorithms.

Wolfgang Günther received his diploma in computer science from the Albert-Ludwigs-University of Freiburg im Breisgau, Germany in 1998. He is currently working at the same institute as a Ph.D. student in the group of Bernd Becker. His research interests include formal verification, logic synthesis, and evolutionary algorithms.



$x_3 x_4$ \ $x_1 x_2$	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



$x_3 x_4$ \ $x_1 x_2$	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

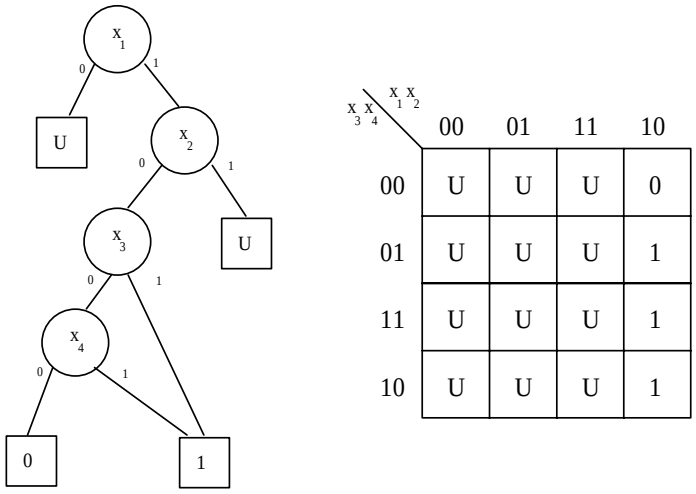


Figure 3: Second incomplete BDD

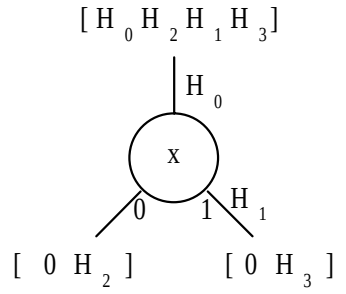


Figure 4: Non-terminal in HSD

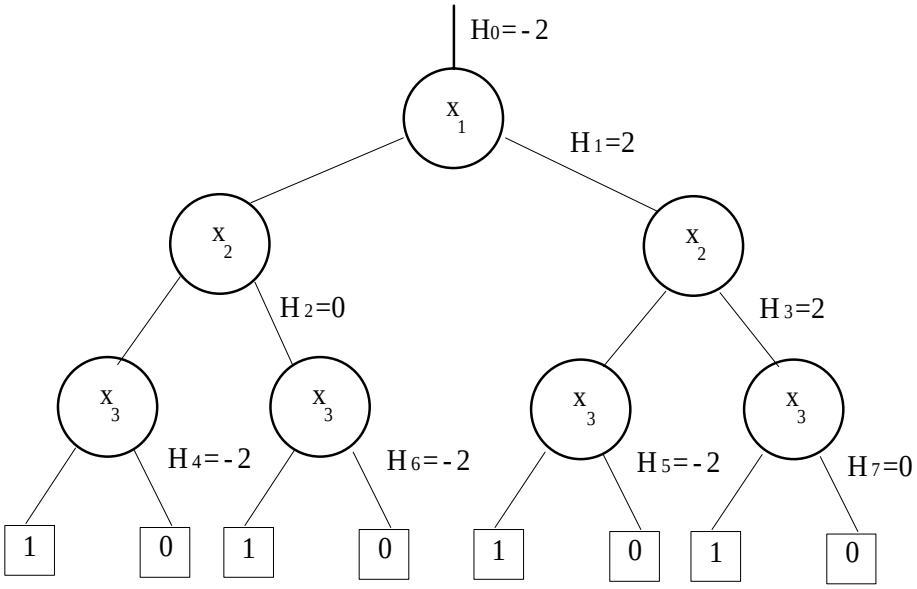


Figure 5: Shannon Decision Tree with Haar Coefficient 1-edge Annotations

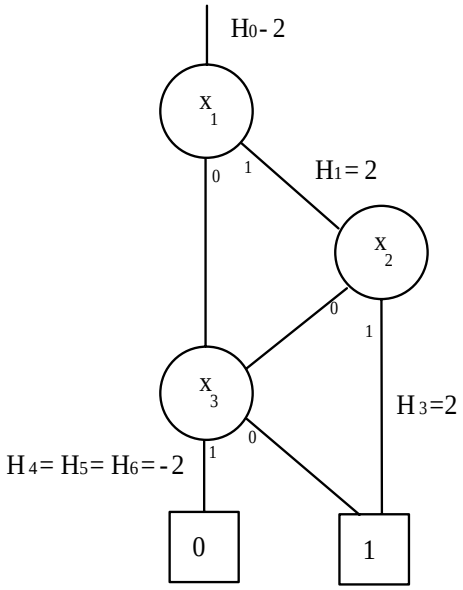


Figure 6: Example HSD

$$
\begin{array}{r}
f \\
x_1 \\
x_2 \cdot f_{\overline{x}_1} \\
x_2 \cdot f_{x_1} \\
x_3 \cdot f_{\overline{x}_1 \overline{x}_2} \\
x_3 \cdot f_{\overline{x}_1 x_2} \\
x_3 \cdot f_{x_1 \overline{x}_2} \\
x_3 \cdot f_{x_1 x_2}
\end{array}
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
\end{bmatrix}
$$

Figure 7: Example of Modified Haar Transformation Matrix for $n = 3$

Table 1: Relationship of the Haar Spectrum and Output Probabilities

SYMBOL	i	$n-i$	p_{m1}	p_{m0}
H_0	0	3	$P[f \cdot 0]$	$P[\overline{f} \cdot \overline{0}]$
H_1	0	3	$P[f \cdot x_1]$	$P[\overline{f} \cdot \overline{x}_1]$
H_2	1	2	$\frac{P[f \cdot \overline{x}_1 \cdot x_2]}{P[\overline{x}_1]}$	$\frac{P[\overline{f} \cdot \overline{x}_1 \cdot \overline{x}_2]}{P[\overline{x}_1]}$
H_3	1	2	$\frac{P[f \cdot x_1 \cdot x_2]}{P[x_1]}$	$\frac{P[\overline{f} \cdot x_1 \cdot \overline{x}_2]}{P[x_1]}$
H_4	2	1	$\frac{P[f \cdot \overline{x}_1 \cdot \overline{x}_2 \cdot x_3]}{P[\overline{x}_1 \cdot \overline{x}_2]}$	$\frac{P[\overline{f} \cdot \overline{x}_1 \cdot \overline{x}_2 \cdot \overline{x}_3]}{P[\overline{x}_1 \cdot \overline{x}_2]}$
H_5	2	1	$\frac{P[f \cdot \overline{x}_1 \cdot x_2 \cdot x_3]}{P[\overline{x}_1 \cdot x_2]}$	$\frac{P[\overline{f} \cdot \overline{x}_1 \cdot x_2 \cdot \overline{x}_3]}{P[\overline{x}_1 \cdot x_2]}$
H_6	2	1	$\frac{P[f \cdot x_1 \cdot \overline{x}_2 \cdot x_3]}{P[x_1 \cdot \overline{x}_2]}$	$\frac{P[\overline{f} \cdot x_1 \cdot \overline{x}_2 \cdot \overline{x}_3]}{P[x_1 \cdot \overline{x}_2]}$
H_7	2	1	$\frac{P[f \cdot x_1 \cdot x_2 \cdot x_3]}{P[x_1 \cdot x_2]}$	$\frac{P[\overline{f} \cdot x_1 \cdot x_2 \cdot \overline{x}_3]}{P[x_1 \cdot x_2]}$

Table 2: Relationship of the Haar Spectrum and Characteristic Equivalence Functions

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

Table 3: Apparent Possibilities Given f, g, $H_i(f)$ and $H_i(g)$

Function Relation	H_i Relation	Observation
$f = g$	$H_i(f) = H_i(g)$	Possible $f = g$
$f = g$	$H_i(f) \neq H_i(g)$	Not Possible
$f \neq g$	$H_i(f) = H_i(g)$	Possible $f = g$
$f \neq g$	$H_i(f) \neq H_i(g)$	$f \neq g$

Table 4: All Possible Boolean Functions for $n = 2$ and their Haar Spectra

Function, f				H_0	H_1	H_2	H_3	Expression
0	0	0	0	4	0	0	0	0
0	0	0	1	2	2	0	2	xy
0	0	1	0	2	2	0	-2	$x\overline{y}$
0	0	1	1	0	4	0	0	x
0	1	0	0	2	-2	2	0	$\overline{x}y$
0	1	0	1	0	0	2	2	y
0	1	1	0	0	0	2	-2	$x \oplus y$
0	1	1	1	-2	2	2	0	$x + y$
1	0	0	0	2	-2	-2	0	$\overline{xy}$
1	0	0	1	0	0	-2	2	$\overline{x \oplus y}$
1	0	1	0	0	0	-2	-2	$\overline{y}$
1	0	1	1	-2	2	-2	0	$x + \overline{y}$
1	1	0	0	0	-4	0	0	$\overline{x}$
1	1	0	1	-2	-2	0	2	$\overline{x} + y$
1	1	1	0	-2	-2	0	-2	$\overline{x} + \overline{y}$
1	1	1	1	-4	0	0	0	1

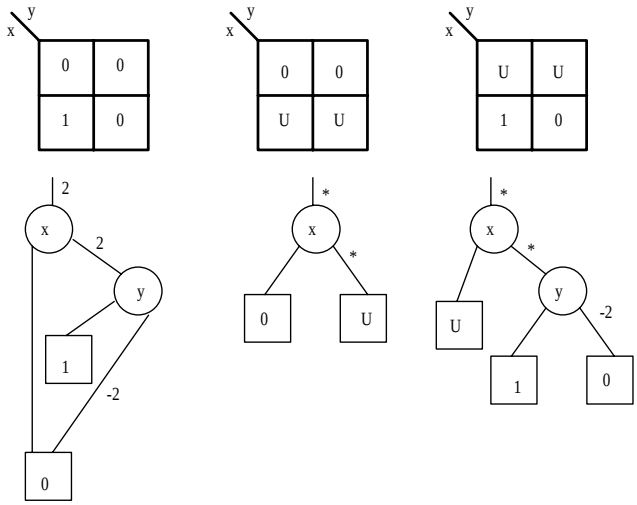


Figure 8: Karnaugh Maps and HSD/BDDs of Complete and Partial Functions

Table 5: 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 6: 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*