Principle Investigator/Project Director: Mitchell A. Thornton

Institution: Southern Methodist University

Award Number: CCR-0243365

Program: CISE

Project Title: Evolutionary Algorithms for Symbolic FSM Equivalence Checking

Research Results in Equivalence Checking

Mitchell A. Thornton, Associate Professor

Aditya Mukherjee, Research Assistant Professor

Department of Computer Science and Engineering

Southern Methodist University, Dallas, TX 75275

ABSTRACT: Formal methods for the verification of *Integrated Circuits* (ICs) are a collection of techniques used to ensure the correctness of a design before fabrication. Formal methods have been investigated recently and continue to be an area of active research in the *Computer Aided Design* (CAD) for *Electronic Design Automation* (EDA) community. While many important results and CAD tools have resulted, the verification problem continues to be difficult due to the high complexity of the underlying algorithms and the circuits that they are intended to validate for correctness. One of the more successful approaches for formal verification is *Equivalence Checking* (EC), both for combinational and synchronous sequential circuits. In general EC is a method that requires two different models of a circuit that are used as input, typically at different levels of abstraction, and determines the functional equivalence (or non-equivalence) at some level. In this research, we are investigating methods to increase the effectiveness of EC. Specifically, we are experimenting with new data structures that are tuned specifically for use in EC algorithms.

1. INTRODUCTION

Design verification is increasingly becoming a bottleneck in time-to-market for *Application Specific Integrated Circuits* (ASIC) and *System On a Chip* (SOC) design ventures. The 2001 *International Technology Roadmap for Semiconductors* (ITRS) design technology document [1] identifies some of the key challenges in design verification. It points out that, unlike technologies such as synthesis, verification is still an art and relies heavily on manual tasks.

One of the more successful verification techniques in use today is *Equivalence Checking* (EC). In general, EC techniques compare two models of a circuit and determine the equivalence of some property; typically, the property of functional equivalence. Functional equivalence is the property that for all given input stimuli in some set, both models produce the same output responses. Two important approaches for the implementation of functional equivalence checkers are to reason about the models using SAT (satisfiability) solving algorithms or to transform each model into some canonical form and then check for the existence of an isomorphic relationship between them. Recently, researchers have developed hybrid methods that encompass both of these approaches [2]. In the research described here, we discuss results achieved from the investigation of methods to increase the effectiveness of EC. Specifically, we discuss new and efficient data structures that are tuned specifically for use in EC algorithms.

2. DISCUSSION OF THE INVESTIGATION

Among the established techniques for showing the equivalence between two candidate circuits are: the use of *Binary Decision Diagrams* (BDDs) [3], the use of *Automatic Test Pattern Generation* (ATPG) techniques, SAT-solvers, and, implication. Each of these approaches has advantages and disadvantages for different classes of circuits. We are investigating the use of these techniques for different classes of circuits and the integration of these techniques in order to study the problem of determining better data structures for these approaches. In particular, we are investigating the use of these methods in the spectral domain to determine if aspects of these approaches may be more efficiently computed based on spectral arguments.

One way to perform EC is to represent two models of a circuit in a canonic form and then check the canonic forms for equivalence. For this approach to be successful, the following constraints must be satisfied:

- Conversion of the circuit model(s) into a canonic form must be efficient
- The canonic form must be compact and not require an excessive amount of memory
- The test determining if two canonic forms are the same must be very fast
- If the two models are not equivalent, a means for identifying at least one test vector that shows model differentiation must be present and be efficient

As an example, BDDs [3] may be used as canonic forms that are compact in size for many functions of interest, relatively fast to construct, easy to compare due to the implementation of a "unique table" data structure [4], and allow for a test vector to be computed in the case of non-equivalence. Problems can occur in this straightforward approach when the size of the BDD becomes too large. This has motivated considerable interest in dynamic variable reordering methods [4, 5] and other techniques for reducing the size of a BDD as it is constructed.

There are some circuits, most notably arithmetic circuits, which can result in exponentially large BDD representations regardless of the variable ordering. This fact has prompted researchers to search for other more compact decision diagram-like structures. One of these is the "Binary Moment Diagram" (BMD) [6]. This structure was shown to be linear in size for integer multiplier circuits that are exponential in size when represented as BDDs. Other decision diagram structures have been proposed leading to further reductions in size for arithmetic circuits such as *BMDs and K*BMDs [7] and decision diagrams adapted for floating point (*PHDDs) [8]. This proliferation of different types of decision diagrams has lead to a virtual "alphabet-soup" of differing decision diagrams and can be very confusing.

Recently, it was shown that these varying types of decision diagrams can be classified very conveniently as graphical representations of discrete spectra of the function of interest [9]. In fact, the BMD proposed by Bryant in 1995 to efficiently represent multipliers [7] is nothing more than a graphical representation of the *arithmetic spectrum*. The arithmetic spectrum has been known since at least [10] but has not seen much use (as have any spectral methods) since efficient means of computing them were not known until recently [11]. Likewise, the so-called "Functional Decision Diagrams" (FDDs) [12] are the same as a decision diagram representation of the Reed-Muller spectrum. Many other decision diagrams are merely hybrid spectral transformations (i.e. spectral transforms whose transformation matrices are products of other transforms), or, are negated and edge-valued versions. The fact that the discrete spectral representations of functions can be computed efficiently and represented as canonic decision diagram forms allows for investigation of new classes of DDs for circuit canonic representation particularly spectral decision diagrams that are hybrid transforms.

3. EXPERIMENTAL RESULTS

A set of experimental results are presented in this section to demonstrate the potential viability of the equivalence checking research outlined above. This set of experimental results demonstrates that large combinational logic functions can be represented in the spectral domain efficiently. Several benchmark circuits (in netlist form) are parsed into a BDD structure and transformed to the spectral domain using the algorithms in [9]. Circuits were chosen that are not trivial cases and that had embedded arithmetic functions. Table 3.0-1 contains these results.

In Table 3.0-1 the column labeled BDD contains the number of vertices in the initial BDD. The columns labeled Walsh and Haar contain the number of nodes in the decision diagrams representing the Walsh and Haar spectra of each of these functions, and T-read, T-Walsh, and T-Haar contain the amount of time (in CPU seconds) to read the netlist into a BDD, transform a BDD into a Walsh decision diagram, and, transform a BDD into a Haar decision diagram. These results were obtained on a 500 MHz PC with 64 MB RAM running Windows 98. The implementation used the CUDD software package [13] with ADDs being used to represent the SDDs. It is interesting to note that in several cases it took longer to synthesize the BDD from the netlist (i.e. to parse the netlist and build the initial BDD representation) than to transform it into the spectral domain. These are preliminary results only and could be improved upon by incorporating negated edges on the spectral decision diagrams. This proposed improvement would allow for further sharing of vertices and would reduce the decision diagram sizes further.

Table 3.0-1: Results Showing the Viability of Spectral Representation

Circuit	In/Out	BDD	Walsh	Haar	T-Read	T-Walsh	T-Haar
alu4	14/8	804	8005	2827	0.16	0.17	0.01
pdc	16/40	695	10581	2992	0.66	0.05	0.01
soar	83/94	482	3778	2321	0.44	0.06	0.01
apex3	54/50	851	34879	15109	0.60	0.46	0.16
des	256/245	3038	27892	16664	2.15	0.38	0.16
dalu	75/16	1037	26692	11399	1.10	0.55	0.28
pair	173/137	3747	memout	42385	4.23	-	0.33
rot	135/107	5922	memout	116370	1.53	-	0.99
c3540	50/22	23851	memout	316666	23.84	-	5.82
c5315	178/123	2197	77554	58037	1.81	3.40	1.15
c7552	207/108	9485	memout	62684	10.94	-	1.95

4. CONCLUSIONS

In summary of the experimental results section, the following point is made. Many past decision diagram structures (BMDs, *BMDs, FDDs, KBMDs, K*BMDs, *PHDDs) that have been developed for arithmetic circuits have been shown to be nothing more than spectral representations of the functions [9]. Because we can represent spectral decision diagrams efficiently, we believe there are many additional structures possible for specialized arithmetic circuits including hybrid combinations of various transforms. Therefore, there is more research needed for finding canonic forms that are useful for circuit verification.

Acknowledgements

The authors are pleased to acknowledge the financial support for this research by the National Science Foundation (CCR-0243365). We also thank Dr. Alan Mishchenko for his invaluable help with the results in Table 3.0-1.

5. REFERENCES

[1] 2001 International Technology Roadmap for Semiconductors, Design Technology Document, http://public.itrs.net/Files/2002Update/2001ITRS/Design.pdf.

[2] A. Kuehlmann, M. Ganai and V. Paruthi, Circuit-based Boolean Reasoning, *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 232-237, 2001.

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

[4] K.S. Brace, R.L. Rudell, and R.E. Bryant, Efficient Implementation of a BDD Package, *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 40-45, 1990.

[5] R. Rudell, Dynamic Variable Ordering for Ordered Binary Decision Diagrams, *Proceedings of the IEEE/ACM International Conference on CAD*, pp. 42-47, 1993.

[6] R.E. Bryant and Y.-A. Chen, Verification of Arithmetic Functions with Binary Moment Diagrams, *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 535-541, 1995.

[7] R. Drechsler, Pseudo-Kronecker Expressions for Symmetric Functions, *Proceedings of the IEEE VLSI Design Conference*, pp. 511-513, 1997.

[8] Y.-A. Chen and R.E. Bryant, *PHDD: An Efficient Graph Representation for Floating Point Circuit Verification, *Proceedings of the IEEE/ACM International Conference on CAD*, pp. 2-7, 1997.

[9] M.A. Thornton, R. Drechlser and D.M. Miller, **Spectral Techniques in VLSI CAD**, Kluwer Academic Publishers, Boston, MA, ISBN 0-7923-7433-9, July 2001.

[10] S.K. Kumar and M.A. Breuer, Probabilistic Aspects of Boolean Switching Functions via a New Transform, *Journal of the ACM*, 28(3):503-520, 1981.

[11] M.A. Thornton and R. Drechsler, Spectral Decision Diagrams Using Graph Transformations, *Proceedings of the IEEE/ACM Conference on Design, Automation and Test in Europe*, pp. 713-717, 2001.

[12] U. Kebschull and W. Rosenstiel, Efficient Graph-Based Computation and Manipulation of Functional Decision Diagrams, *Proceedings of the IEEE/ACM European Conference on Design Automation*, pp. 278-282, 1993.

[13] F. Somenzi, CUDD: CU Decision Diagram Package, Release 2.1, University of Colorado, Boulder, CO.