2010 2nd International Conference on Education Technology and Computer (ICETC)

Comparing Simulations and Graphical Representations of Complexities of Benchmark and Large-Variable Circuits

Prasad P.W.C.
Charles Sturt University,
Sydney Study Centre,
NSW, Australia
c.withana@sga.edu.au

Azam Beg
College of Information Technology
UAE University, Al-Ain,
United Arab Emirates
abeg@uaeu.ac.ae

Ashutosh Kumar Singh
Department of Electrical and Computer Engineering
Curtain University of Technology
Sarawak Campus, Malaysia
ashutosh.s@curtin.edu.my

Abstract: **In this work, we analyzes the relationship between randomly generated Boolean function complexity and the number of nodes in benchmark circuits using the Binary Decision Diagrams (BDD). We generated BDDs for several ISCAS benchmark circuits and derived the area complexity measure in terms of number of nodes. We demonstrate that the benchmarks and randomly generated Boolean functions behave similarly in terms of area complexity. The experiments were extended to a large number of variables to verify the complexity behavior. It was confirmed that the rise of the complexity graph is only important to calculate the circuit complexities.**

Keywords-Binary Decision diagram, Benchmark circuits, Area Complexity

I. INTRODUCTION AND LITERATURE REVIEW

BDD and its derivatives based on Boolean decomposition such as Davio [1], Shannon [2], Read-Muller[3], Kronecker [4] etc., require the inputs and outputs to be in terms of bit levels. Therefore, these representations can be quite time consuming. However, representation of multiple output functions has important applications in areas such as logic simulation and testing [5]. As the circuit sizes continue to grow, the need for fast evaluation becomes even more significant. The continuous increase of integration level of modern digital circuits imposes high and growing requirements for methods and algorithms used in VLSICAD design verification and testing [5] [6]. According to Moore's law [7], the number of transistors on a single chip doubles every year, and it has withstood the test of time since Gordon Moore made this observation in 1965. Boolean function representation has a direct influence on the computation time and space requirements of digital circuits and most of the problems in VLSI/CAD designs can be formulated in terms of Boolean functions. The efficiency of any method used depends on the complexity of Boolean functions [8] . Research on the complexity of Boolean functions in non-uniform computation models is now part of one of the most interesting and important areas in theoretical computer science [8]- [10]. Rapid increases in the design complexity and the need to reduce time-to-market have resulted in a need for computer-aided design (CAD) tools that could help make important design decisions *early* in the design process. Area complexity is one of the most important criterion that has to be taken into account while making these decisions. However, to be able to make these decisions early, there is a need for methods to estimate the area complexity and power consumption from a design description of the circuit at a high level of abstraction [11].

In 1949, Shannon [2], studied area complexity, measured in terms of the number of relay elements used in building a Boolean function (switch-count). In that paper, Shannon proved that the *asymptotic* complexity of Boolean functions is *exponential* to the number of inputs, and that for large, *almost every* Boolean function is exponentially complex. In 1956, Muller demonstrated the same result for Boolean functions implemented [3] using logic gates (gate-count measure). A key result of his work is that a measure of complexity based on gate-count is independent of the nature of the library used for implementing the function. Several researchers have also reported results on the relationship between area complexity and entropy of a Boolean function. In 1990 Cheng et al., empirically [12] demonstrated the relation between entropy and area complexity, with area complexity measured as literal count. They showed that *randomly generated* Boolean functions have a complexity *exponential* in, and proposed to use that model as an area predictor for logic circuits. However, the circuits tested were very small, typically having less than ten inputs. As one tries to apply that model to realistic very large scale integration (VLSI) circuits, it quickly breaks down due to the exponential dependence, leading to unrealistically large predictions of circuit area. For example, when applied to a circuit with 32 inputs (having been tuned to inputs), this model predicts an area of million gates, whereas the circuit can in reality be implemented with only 84 gates. [11]. In 1999, Nemani and Najm, proposed an area and power estimation capability, given only a *functional* view of the design, such as when a circuit is described only with Boolean equations. In this case, no structural information is known and the lower level (gate-level or lower) description of this function is not available. The methods proposed in Wu [13] and Kurdahi [14] both make use of the sum-of-products (SOP) representation of a function, and estimate the area based on the total number of AND and OR gates required in this representation. Typically, the actual number of gates required will be much smaller than this number after optimization.

978-1-4244-6370-1/$26.00 © 2010 IEEE

The use of logic verification and optimization algorithms in VLSI CAD systems requires efficient representation and manipulation of Boolean functions [5]. During the last two decades, BDDs have gained great popularity as successful method for the representation of Boolean functions [6], [15]. The ever-increasing complexity of circuit designs is directly related to the complexity of parameters that describe the Boolean function. Over the years, the number of nodes in a BDD became a major concern since it is proportional to the complexity of the Boolean circuit [16]. Over the past two decades most of the problems in the synthesis, design and testing of combinational circuits, have been solved using various mathematical methods [17], [18]. Researchers in this area are actively involved in developing mathematical models that predict the number of nodes in a BDD in order to predict the complexity of the design in terms of the time needed to optimize it and verify its logic.

The main objective of this paper is to extend the work done by the same authors on the BDD complexity for the benchmark circuits and analyze its overall behavior. The remaining of this paper is divided as follows: in the second section, we review the previous work done on estimation by the authors. Section three provides the comparison for the ISCAS benchmark [19] results for BDD area complexity derived from Colorado University Decision Diagram (CUDD) [20] and the results extracted from the complexity graphs. The complexity behavior for higher number of variables and how to exploit those results for ISCAS benchmark circuit complexity verification were explained in section four, followed by the conclusions.

II. PREVIOUS WORK BY THE AUTHORS

In this section, we briefly describe authors' previous work completed and results achieved in the area of the estimation of BDD complexity.

The complexity of the ROBDD mainly depends on the number of their nodes. Simulations have been performed in [21], [22] to analyze the complexity variation in ROBDDs i.e. the relation between the number of product terms and the number of nodes for any number of variables.

We carried out experiments using CUDD [20] package to analyze the exact complexity variation of an ROBDD, i.e., the relation between the number of product terms and the number of nodes for any number of variables. For each variable count n between 1 and 14 inclusive and for each term count between 1 and 2^n-1, 100 SOP terms were randomly generated and the CUDD) package was used to determine the area complexity in number of nodes. This process was repeated until the average size of the area complexity (i.e. number of nodes) became 1. Then the graphs for area complexity (Fig. 1) were plotted against the product term count for each number of variables.

The Fig. 1 graph indicates that the BDD complexity in general increases as the number of product terms

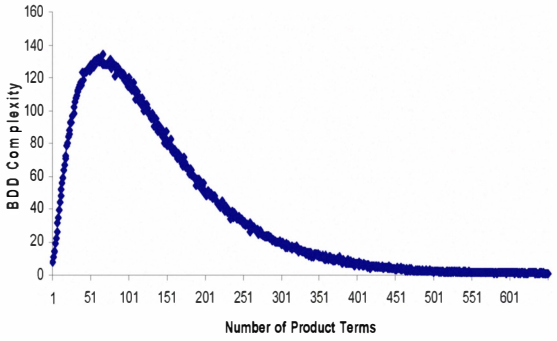

Fig 1. BDD complexity variation results for 10 variables

increases. This is clear from the rising edge of the curve. At the end of the rising edge, the graph reaches a maximum complexity. Apart from that the peak also specifies the number of product terms (critical limit) of a Boolean function that leads to the maximum BDD complexity for any Boolean function with 10 variables. If the number of product terms increases above the critical limit, as expected, the product terms starts to simplify and the BDD complexity will reduce. The BDD complexity graph shown in Fig. 2 indicates that as the number of product terms increases the complexity of the BDD decreases at a slower rate and ultimately reaches 1 node.

III. VALIDATION OF THE GRAPHICAL REPRESENTATION OF BOOLEAN FUNCTION COMPLEXITIES

The graphical predictions for BDD complexity for selected ISCAS benchmark circuits are tabulated in Table 1. The actual results for ISCAS benchmark circuits were obtained on an X86 PC running on Linux environment.

The 1st column indicates the ISCAS benchmark circuit name and the 2nd and 3rd columns are for the input variables and number of outputs for the respective benchmark circuit. In column 4, the actual BDD complexity for the benchmark circuits have been calculated using CUDD package. For the graphical prediction calculation, each benchmark circuits were extracted to find the total number of SOPs. These SOP terms consist of different number of variables and product terms. Therefore the calculations were done from different variable graphs. The graphical predictions for benchmark circuits are tabulated in column 5.

Although the benchmark circuits considered had up to 47 inputs, no output depended on more than 14 of those inputs. The circuits for all outputs were measured. It was observed that the term-variable count combinations were almost all to the left of the peak complexity, and thus still in region of logarithmic complexity. So, empirically the most important part of the model is the logarithmic rise, and it was this part that has been validly tested by the benchmark circuit analysis. This part of the model also has the strongest theoretical grounding.

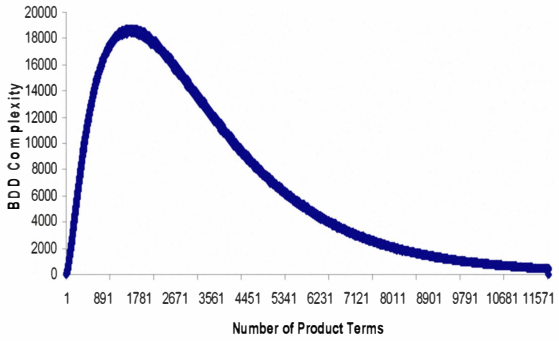

Fig 2. BDD complexity variation results for 18 variables

IV. GRAPHICAL REPRESENTATION FOR HIGHER NUMBER OF VARIABLES

Fig. 3 illustrates the results for higher number of variables. Note that only the left part of the full graph is shown; it is because of the long simulation times. Since this curve represents a small number of data points, it is hard to conclude whether the general behavior of the BDD complexity for these variables remains the same or changes. We are in the process of collecting more data points in order to get the average of the complexity for a given number of product terms, which will make it easy to compare all the graphs behaviors on the same scale. It is obvious that curves are more difficult to generate for larger number of variables because of the higher number of SOP terms it generates as random Boolean functions. As the number of variable increases, the random generated Boolean function complexity of the SOP functions increases exponentially (2^n). Therefore, the hardware resources used for actual benchmark complexity calculation will not be sufficient to complete the simulations for higher number of SOP. This can be the main cause for not getting the complete graph for higher number of variables. However it cannot be a critical factor for the proposed graph prediction method as discussed in previous paragraph.

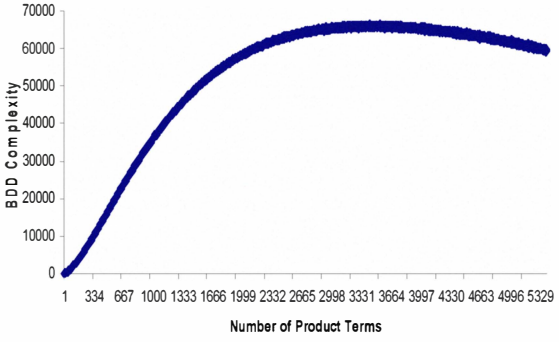

Fig 3. BDD complexity variation results for 20 variables

From Fig 3, it can be inferred that the difference between this graph and previous were only due to lack of samples in order to get more simplified data points. According to this graph for 20 variables, those data points will come only after 5400 product terms. However those product terms could not produce the output due to complexities of those randomly generated product terms. Figs 4-6 illustrate the graphical representation of BDD complexity for 28, 45 and 60 variables, respectively.

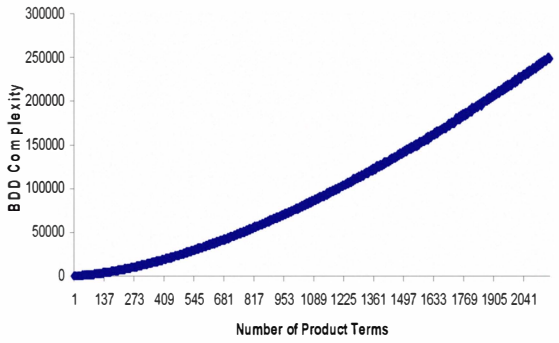

Fig 4. BDD complexity variation results for 28 variables

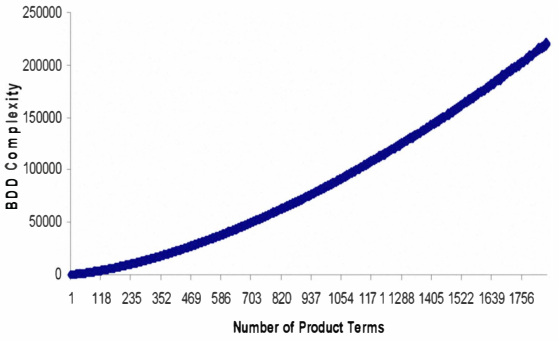

Fig 5. BDD complexity variation results for 45 variables

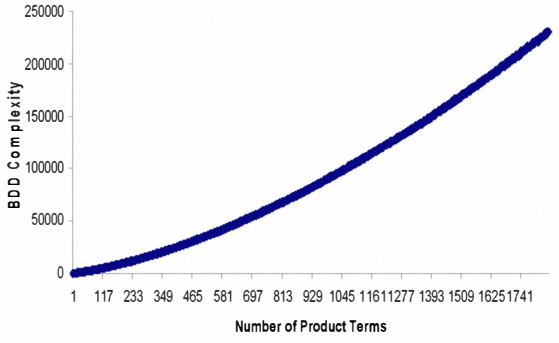

Fig 6. BDD complexity variation results for 60 variables

V. CONCLUSION

In this work, we have analyzed the relationship between actual results and the results calculated from graphical representation for a larger selection of ISCAS benchmark circuits. An advantage of this model is that it is a single integrated model for different number of variables and number of product terms.

Table 1
Complexity estimation results for ISCAS benchmark circuits

Circuit	Inputs	Outputs	BDD	Complexity
			Actual	Graphical
Apex4	9	19	1452	1286
apex7	10	8	325	344
b1	3	2	12	12
B12	16	9	94	97
B9	42	21	225	230
C1355	41	32	64075	67437
C17	5	2	10	10
C1908	33	25	18062	19681
C432	36	7	2470	2493
C499	41	32	60434	62923
C8	28	18	139	149
Cc	21	20	124	105
cht	47	36	181	192
clip	9	45	368	394
cm138a	6	6	48	56
cm150a	21	1	31	34
cm152a	11	8	23	28
Cm162a	14	5	47	56
cm163a	9	8	49	59
cm42a	4	10	50	50
cm82a	5	12	16	19
cm85a	10	16	41	42
cmb	16	4	54	59
Comp	32	3	294	312
con1	5	5	17	18
count	35	16	216	229
cu	14	11	79	89
decod	5	1	87	96
F51m	14	8	66	74
I1	26	13	74	81
i6	5	6	408	413
i7	6	5	510	493
Misex1	8	7	87	89
misex1	6	6	66	69
Misex2	25	18	177	171
Mux	21	1	64	69
My_adder	33	17	681	693
Pm1	16	13	77	80
rd53	5	16	23	24
rd73	7	64	33	38
rd84	8	162	51	54
S5378	36	49	85	89
Sao2	10	4	113	131
sct	19	15	169	177
Sqrt8	8	4	30	43
Squar5	5	8	56	57
t481	16	16	31	31
term1	34	10	53	57
ttt2	24	21	117	117
Vg2	25	8	190	194
X2	10	7	60	68
Z4ml	7	4	36	32

The illustrated results show that the graphical calculation of the benchmark complexity was very closer to the actual results and also the importance of the graphical data up to the peak of the graph. We also concluded that the graphical representations for higher number of variables are computationally harder and they will only be useful for justifying the BDD complexity behavior and not much importance on the calculation of the benchmark complexities.

ACKNOWLEDGMENT

The work of this paper is partially funded by Curtin Sarawak Research Fund, Curtin University of Technology, Malaysia.

REFERENCES

[1] Thayse, A. and Davio, M., "Boolean Differential Calculus and Its Application to Switching Theory", IEEE Trans. Computers, 1973, vol. 22, no. 4, pp. 409-420.

[2] Shannon, C. E. (1949). "The synthesis of two-terminal switching circuits", Bell System Technical Journal, vol. 28 (1), pp. 59-98.

[3] Muller, D. E. (1956). "Complexity in electronic switching circuits", IRE Transactions on Electronic Computers, vol. 5, pp. 15-19.

[4] Drechsler, R., Sarabi, A. Theobald, M., Becker, B., and Perkowski, M.A., "Efficient Representation and Manipulation of Switching Functions Based on Order Kronecker Function Decision Diagrams", Proc. Design Automation Conf. (DAC), 1994, pp. 415-419.

[5] Priyank, "VLSI Logic Test, Validation and Verification, Properties & Applications of BDDS", *Lecture Notes,* Department of Electrical and Computer Engineering University of Utah, UT 84112, 1997.

[6] Bryant, R. E., "Graph–Based Algorithm for Boolean Function Manipulation", IEEE Trans. Computers, 1986, vol. 35, pp. 677-691.

[7]Moore, G. E., "Progress in Digital Integrated Electronics", *IEEE IEDM*, 1975, pp.11-13.

[8] Wegener, I., "The Complexity of Boolean Functions", John Wiley and Sons Ltd, 1987.

[9] Meinel, C., and Slobodova, A., "On the Complexity of constructing Optimal Ordered Binary Decision diagrams", Proc. of 19th Inter. Symposium on Mathematical Foundation of Computer Science, 1994, pp. 515-524.

[10]Tani, S., Hamaguchi, K. and Yajima, S., "The Complexity of the Optimal Variable Ordering Problems of a Shared Binary Decision Diagram", IEICE Transactions on Information and Systems, 1996, vol. 4, pp. 271-281.

[11] Nemani, M.,and Najm, F. N., "High-Level Area and Power Estimation for VLSI Circuits", IEEE Tran. on CAD of Integrated Circuits and Sys., 1999, vol.18 (6), pp. 697-713.

[12] Cheng, K. T., and Agrawal, V. "An entropy measure for the complexity of multi-output boolean functions", Proceedings of DAC, 1990, 302-305.

[13] Wu, A. C. H., "Layout-area models for high-level synthesis", Proceedings of. International Conf. Computer-Aided Design (ICCAD), 1991, pp. 34–37.

[14] Kurdahi, F. J., "Linking register-transfer and physical levels of design", IEICE Transactions on Information and Systems, 1993, vol. 76(9), pp. 991–1002.

[15] Akers, S. B., "Binary Decision Diagram", IEEE Trans. Computers, 1978, vol. 27, pp.509-516,

[16] Drechsler, R., and Sieling, D., "Binary Decision Diagrams in Theory and Practice", Springer-Verlag Transaction, 2001, pp.112-136,

[17]Jain, J., Moundanos, D., Mohanram, K., Lu, Y., and Touba, N. A., "Data Structures for Partial Verification", Proceedings of World Multiconference on Systemics, Cybernetics, and Informatics (SCI), 2000, vol. 8.

[18] Van Eijk, C. A. J., "Formal Methods for the Verification of Digital Circuits", PhD thesis, Eindhoven University of Technology, Netherlands, 1997.

[19] Hansen, M., Yalcin, H., and Hayes, J. P., "Unveiling the ISCAS-85 Benchmarks: A Case Study in Reverse Engineering", IEEE Transaction on Design and Test, 1999, vol. 16, pp. 72-80.

[20] Somenzi, F., "*CUDD: Decision Diagram Package*". *ftp://vlsi.colorado.edu/ pub/*, 2003.

[21] Raseen, M., Prasad, P.W.C., and Assi, A., "An Efficient Estimation of the ROBDD's Complexity", Integration - the VLSI journal, 2005, vol.39, pp. 211-228.

[22] Raseen, M., Prasad, P.W.C., and Senanayake, S.M.N.A., "XOR/XNOR Functional Behaviour on ROBDD Representation", Proceedings of the 14th IASTED Conference on Applied simulation and Modelling (ASM), 2005, pp. 115-119.