International Journal of Modern Electronics and Communication Engineering (IJMECE) ISSN: 2321-2152
Volume No.-6, Issue No.-6, November, 2018

A Hybrid-Sift Algorithm for BDD Based Area Optimization of MIMO Adder Circuits

M. Balal Siddiqui1st
Department of Electronics & Comm. Engg.
Jamia Millia Islamia
New Delhi, India
E-mail: balalsid@gmail.com

M. T. Beg2nd
Department of Electronics & Comm. Engg.
Jamia Millia Islamia
New Delhi, India

S. N. Ahmad3rd
Department of Electronics & Comm. Engg.
Jamia Millia Islamia
New Delhi, India

Abstract: Adder circuits are basic building block of today's electronic circuits. They are important part of all the modern digital Integrated Circuits. Today the increased size of data tends to increase in IC complexity, which results in more area consumption for different units including adders. The overall area minimization is the prime concern of IC design today. Binary decision diagrams are technique use for implementation of digital circuits in various computer aided design tools of VLSI design which finally implemented to physical structure. In this work our prime focus is on minimization of various BDD represented adder circuits. We have proposed a Hybrid-Sift algorithm technique which uses an improved initial variable ordering with the existing technique of generation variable order for optimization of BDD circuits. The results have compared with the different other existing methods implemented for adder circuits. The improved result shows that this proposed work has a good improvement of overall area of adder circuits of different bit sizes**.**

Keywords: BDD, Variable Ordering, Decision Diagram, Adders Optimization

I. INTRODUCTION

Adders are the important building block of the any digital circuits. The main works of adders are for arithmetic and mathematical operations. With the increasing complexity of today's electronic circuits, the chip area consumption of adders is also increasing. Area optimization of the different units of an electronic circuit is the prime concern of todays integrated chip designers. Different data structures are used in modern VLSI CAD tools to represent the digital hardware and finally to reduce (or optimize) the overall area using different design constraints. Binary Decision Diagram is one such data structure used for implementation and representation of different digital units in VLSI CAD tools [1]. The adder's representation using Binary Decision Diagrams are quite easy and require less implementation time. Binary Decision Diagrams are data structures which are acyclic, tree like in nature [2]. In Implementation, Binary Decision Diagrams consist of one root node, various internal nodes which called intermediate nodes and two end nodes which called leaf nodes [3]. There are different works proposed by different authors for minimization of Binary Decision Diagram based electronic circuits [4]. The variable ordering of input variables in Binary Decision Diagram is NP-Complete problem [5-6].

In this paper, a method based on improved initial ordering combined with sift algorithm for variable ordering of adder circuit is implemented.. A comparison of this proposed method with the different results obtained from different algorithmic methods is done for adder circuits of different bit counts.

This organization of this paper is as follows: Section 1 is introduction, Section 2 is basics about Binary Decision Diagrams, Section 3 is about adders' implementation using Binary Decision Diagram and the proposed method, Section 4 discussed results and comparison with other works, Section 5 is conclusion of the work.

II. BINARY DECISION DIAGRAMS AND ORDERING OF VARIABLES

A. *Binary Decision Diagrams*

Binary Decision Diagrams are data structures which are acyclic, tree like in nature. In Implementation, Binary Decision Diagrams have only one root node, various internal nodes which called intermediate nodes and two end nodes which called leaf nodes [7]. The Binary Decision Diagram based hardware implementation has one extra advantage that the final BDD structure can be easily implemented using 2:1 MUXes in which each node of the BDD represented by a single 2:1 MUX [8]. Also the Reduced and Ordered BDDs (ROBDD) using different rules propose by Bryant in his work [1] results in a canonical representation of the Adder circuits. In our work, the term BDD represents the Reduced and Ordered Binary Decision Diagram (ROBDD). The term 'Ordered' means that the ordering of each input variable is same in each branch of

the decision diagram. The term 'Reduced' means the basic structure is reduced using some rules, which are:

Rule 1: Merging of leaf nodes i.e. duplicate high and low terminal nodes removal (merging all Low-leaf nodes to a single Low-leaf node and merging each high-leaf nodes to a single high-leaf node)

Rule 2: Duplicate intermediate nodes removal

Rule 3: Redundant nodes removal.

Fig. 1 shows the rule 1, duplicate high and low terminal nodes removal for a decision diagram of a digital circuit.

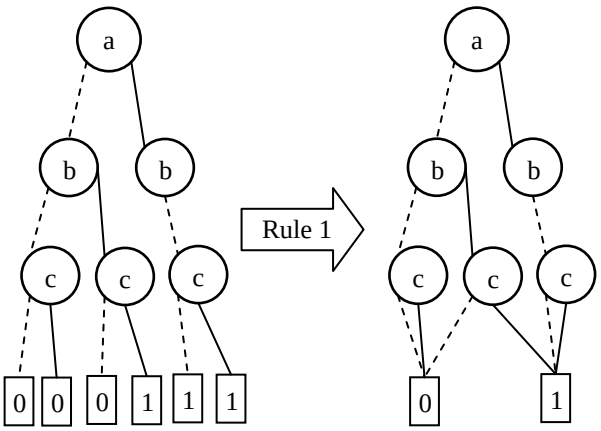

Figure 1. Merging of high leaf nodes and low leaf nodes.

In Fig.1 all the high (ones) terminal node is merged to a single high terminal node whereas all the low (zeros) terminal nodes are merged to a single low terminal node. In rule 2, if there are two same nodes pointing to same set of nodes and pointed by same set of nodes then one node can be easily removed and that removal have no effect on the overall functionality of the system. In rule 3 if any node is pointed by single node by both high and low branch and pointing to a single node by its both high and low branches then the node can be easily removed and this removal also have no effect on the overall functionality of the system. This is shown in Fig. 2.

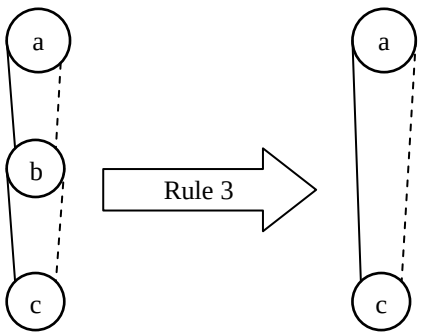

Figure 2. Example of redundant nodes removal.

B. *Ordering of Input Variables*

The ordering of input variables has very strong effect on the size of the Binary Decision Diagrams. The sizes of the Diagrams are the total number of internal nodes. The total numbers of internal nodes are very sensitive for input variable ordering. Fig. 3 shows the effect of variable ordering on the number of internal nodes of a function. In Fig. 3, both the BDD diagram represents the same Boolean function but the different ordering of input variables results in the different number of internal nodes. So we can easily observe the effect of variable ordering on the total size of the Decision Diagrams.

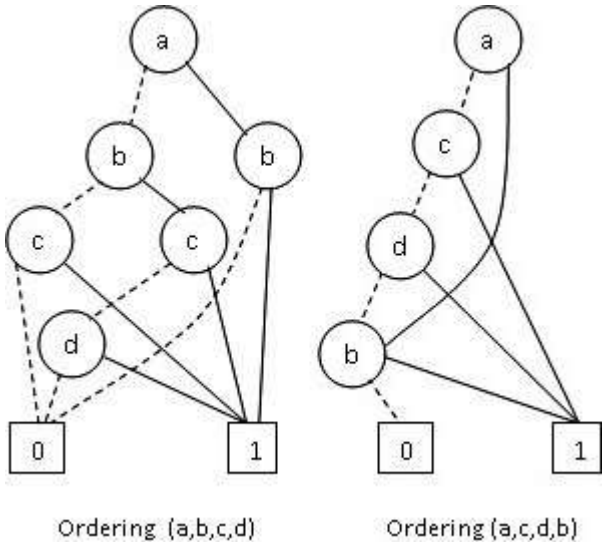

Figure3. BDD of the same function with different variable ordering.

For Adders of higher bit size in which the internal nodes size is in terms of hundreds, the input ordering of the adder inputs have very strong effect on the overall chip area of the adders.

III. PROPOSED METHOD AND IMPLEMENTATION ON ADDER CIRCUITS

A. *Proposed Method*

This proposed work is implementation of an improved initial variable ordering, which work as first order for sift algorithm technique for BDD variable reordering of adder circuits. Implementation of the proposed work is shown in the flow diagram in Fig. 4 and also in Algorithm shown in Fig 5. A random variable number is first chosen and that is moved to some position according to the scheme shown in the figure. The number next to it is also moved to all the positions except the position(s) locked by the previous number(s). In this way, using the scheme shown in the flow diagram of the Fig. 4, we have calculated the optimized position for each variables and the final positioning of all the variables are used as initial variable order for the *Sift_algorithm* method , hence the name *hybrid-sift_Algorithm.*

B. *Implementation on Adder Circuit for Area Optimisation*

The scheme for calculation of area is shown in Fig. 5. In Fig. 5 an algorithm to calculate area is shown. The '*.pla*' based adder circuits from different benchmark suits are used for

implementation of BDD based structure. The final BDD structure is converted to MUX based structure in which each internal node of the BDD is converted to one 2:1 MUX. The total area consumption is calculated by multiplying the total number of MUXes to the total area of a single MUX. We have used the area unit 2:1 MUX as 14.87 sq. units for 180 nm technology calculated using synopsis tool [9].

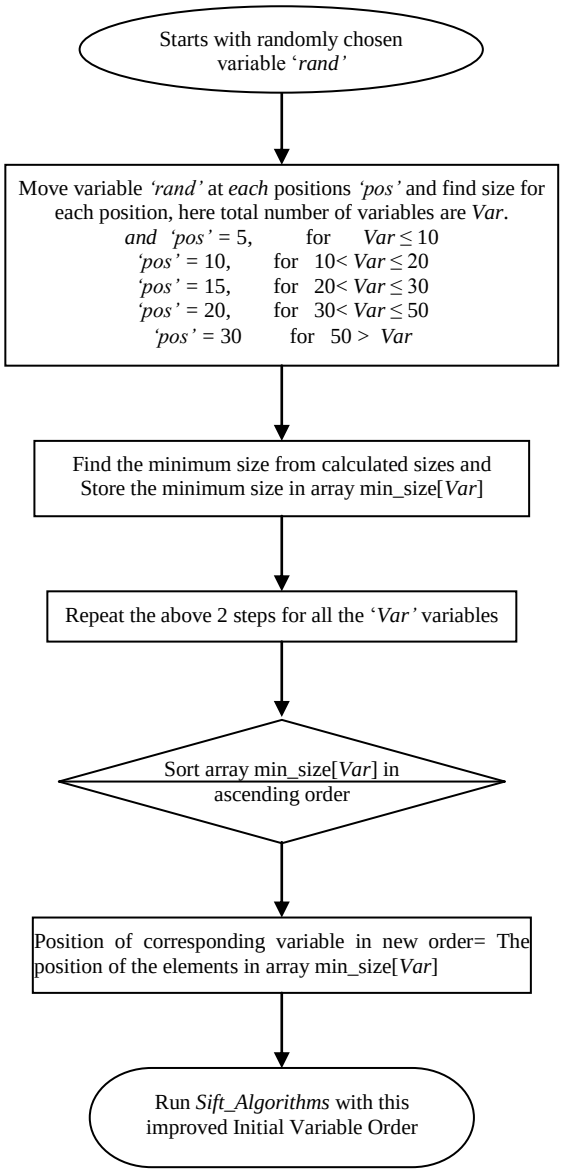

Figure 4. Flow diagram for proposed initial variable order calculation.

IV. RESULTS

The whole scheme is implemented in C language on an Ubuntu desktop OS on UNIX platform. Buddy-2.4, which is a UNIX based BDD manipulation package, which is proposed by Nielsen [10], is used for calculation of different BDD related parameter. The adder circuits in '*.pla*' format from different benchmark suits are first converted to '*C*' language based expressions and then implemented as input in program. The program for area calculation using algorithm shown in Fig. 5 is also written in '*C*'. A detail comparison of the result obtained from proposed hybrid_sift_Algorithm based approach is calculated and compared with the area calculated using existing techniques available in BuDDy-2.4, a BDD manipulation tool. The adders of different length (bit sizes 1 to 7) are used in our work. The comparison of area is shown in Table 1. The results shown in Table 1 are also shown in bar diagram in Fig. 6. From bar diagram of Fig. 6 and from Table 1 we can easily observe that our proposed method takes minimum area compared to other existing methods. In Table 1, we have compared the increased area, with reference to our proposed method. From Table 1 we can easily observed that the existing methods of BuDDy-2.4 package namely '*INITIAL_NODE_COUNT*', '*WIN2*', '*WIN2ite*', '*WIN3*' and '*RANDOM*' techniques have 3.04, 3.07, 3.05, 1.04 and 1.68 times more area consumption respectively than our proposed Hybrid-Sift method, Which is scaled down to 1.

Algorithm: find minimum Area

Require: Minimum Area
1: determine step size S;
2: start with first variable;
3: **for** v = 1 to N **do**
4: pos_vector = S;
5: **for** p = 0 to N **do**
6: $Var[v] \Leftrightarrow Var$[pos _vector];
7: calculate Optimization Parameter (Area);
8: store the result in a Array: weightage[ ];
9: pos_vector = pos_vector + 'S';
10: **end for**
11: V_best = best value in Array: weightage[ ];
12: best position for variable Var[v]← Position of V_best in Array: weightage[ ] ;
13: fix the position of Var[v];
14: **end for**
15: lock position for all variables:
16: use this ordering as seed ordering;
17: apply sift_Algorithm with this seed ordering;
18: find minimum area;

Figure 5. Algorithm for find minimum area for different adder circuits.

V. CONCLUSION

In this work, we have successfully implemented the proposed hybrid-sift algorithm technique. The results obtained using this proposed technique shown quiet a good area reduction when compared to other existing techniques. The adders of up to 7 bit size have been implemented. Higher bit adders have not implemented due to high execution time and system memory limitation, but for bigger circuits also this method seems to give good results. For future work, this proposed technique can be implemented with other existing techniques also.

Table1. Comparison of area calculated with proposed technique with other existing techniques

Sl. No.	Adder circuits	#i	#o	Area Calculation (in sq. units)					
				Existing Methods of BuDDy-2.4					Proposed Hybrid-Sift Method
				Initial node count	WIN2	WIN2ite	WIN3	RANDOM	
1.	1-adder	3	2	118.96	118.96	118.96	118.96	118.96	118.96
2.	2-adder	5	3	252.79	252.79	252.79	252.79	252.79	252.79
3.	3-adder	7	4	475.84	475.84	505.58	475.84	505.58	386.62
4.	4-adder	9	5	936.81	966.55	936.81	684.02	907.07	684.02
5.	5-adder	11	6	1873.62	1903.36	1873.62	907.07	1159.86	817.85
6.	6-adder	13	7	3762.11	3791.85	3762.11	1263.95	1382.91	1263.95
7.	7-adder	15	8	7553.96	7583.7	7553.96	1397.78	3925.68	1397.78
Area increment compared to proposed method				**3.04**	**3.07**	**3.05**	**1.04**	**1.68**	**1**

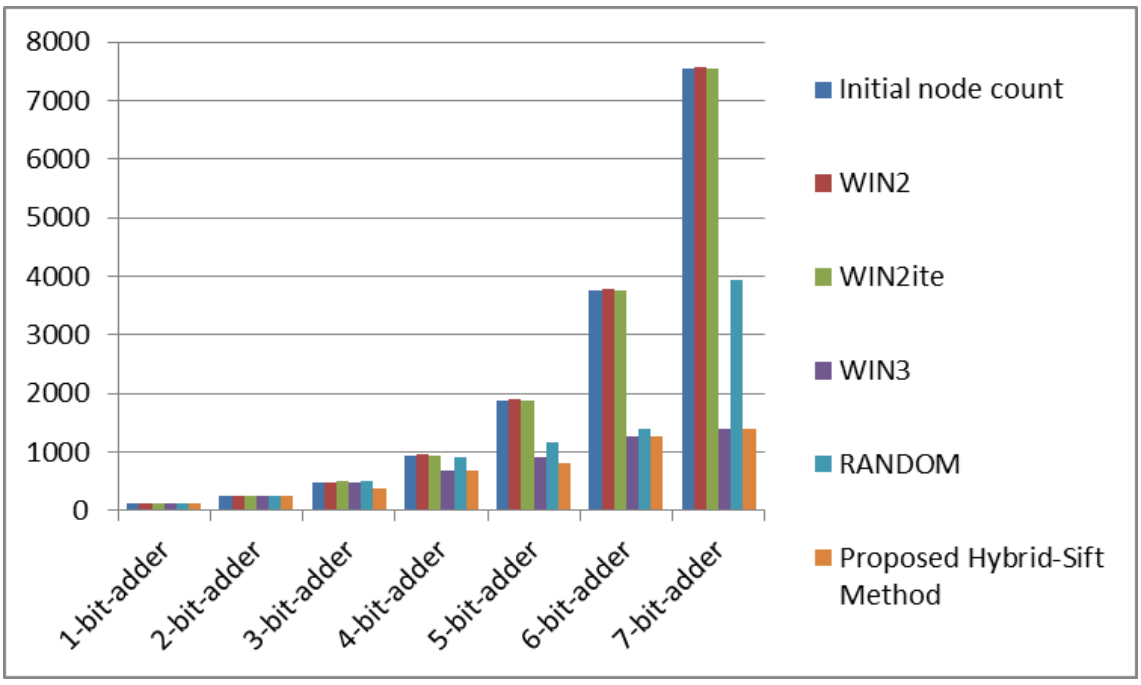

Figure 6. Comparison of different existing methods with proposed method for adders of different size..

REFERENCES

[1] Bryant, R.E, "Graph-based algorithms for Boolean function manipulations," IEEE Trans. on Computers, C-35: pp. 677-691, 1986.

[2] C. Y. Lee, "Representation of switching circuits by Binary-Decision Programs," Bell Systems Technical Journal, 38: pp. 985–89, 1959.

[3] R. Rudell, "Dynamic variable ordering for Ordered Binary Decision Diagrams," Proceedings of the 1993 International Conference on Computer Aided Design (ICCAD), Santa Clara, CA, USA, pp. 42-47, 1993.

[4] M. B. Siddiqui, S. N. Ahmad and M. T. Beg, "Variable ordering of BDD mapped multi-input multi-output adders using modified genetic algorithm," 2017 International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT), Aligarh, pp. 218-221, 2017.

[5] S. I. Minato, "Binary Decision Diagrams and Applications for VLSI CAD," Book, Norwell, Massachusetts: Kluwer; 1995.

[6] D. Bergman, A. A. Cire, W. J. Van Hoeve et al., "Decision diagrams for optimization," Switzerland: Springer; 2016.

[7] M. B. Siddiqui, S. N. Ahmad and M. T. Beg, "Modified GA method for variable ordering in BDD for MIMO digital circuits," In: Proceedings of the 2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT); 2016, Dec 2-3; Pune, India. pp. 378-82, 2016.

[8] P. K. Sharma, N. K. Singh, "BDD based area and power efficient digital circuit design using 2T and 4T MUX at 90 nm technology," 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kanyakumari, India, pp. 7–11, 2014.

[9] Preeti Singh, "Synthesis and optimization of a 4-bit magnitude comparator circuit using BDD and pre-computation based strategy for low power," [M.Tech thesis]. Patiala, India: Thapar University, Available from: Thapar University Library E-Reserve (Online Thesis), 2012.

[10] BuDDy: A Binary Decision Diagram Package By Jorn Lind-Nielsen, http://sourc eforge.net/projects/buddy (online)

AUTHOR'S BIOGRAPHIES

M Balal Siddiqui received the B.E. degree in Electronics and Communication Engineering from Visvesvaraya Technology University, Belgaum and M.Tech. degree in VLSI Design & CAD from Thapar University, Patiala in 2009 and 2013, respectively. He is now with Jamia Millia Islamia as a Research Scholar.

M Tariq Beg received the B.Sc. degree in Electrical from Aligarh Muslim University, M.Tech. degree in Microwave Electronics from Delhi University and Doctorate degree from Jamia Millia Islamia in 1985,1987 and 2003, respectively. He is now with Jamia Millia Islamia as a Professor..

S Naseem Ahmad received the B.Sc. degree in Electrical from Aligarh Muslim University, M.Tech. degree in Microwave Electronics from Delhi University and Doctorate degree from Jamia Millia Islamia in 1985,1987 and 2003, respectively. He is now with Jamia Millia Islamia as a Professor.