

BDD Ordering and Minimization Using Various Crossover Operators in Genetic Algorithm

Rupinder Kaur¹, Manu Bansal²

Thapar University, Patiala, India¹

Assistant Professor, ECED, Thapar University, Patiala, India²

Abstract: Binary Decision Diagram (BDD) is a data structure which is extensively used for compact representation of Boolean functions. On a more abstract level, BDDs can be considered as a compressed representation of sets or relations. BDDs are extensively used in CAD software to synthesize circuits (logic synthesis) and in formal verification. Ordering of BDDs play a major role in reduction of nodes and hence the area. In this paper, genetic algorithm with three crossover operators namely order, cycle and partially mapped has been proposed for minimization of shared ordered BDDs. The results have been compared using these three operators for Multi-input Adder Benchmark Circuits.

Keywords: Genetic, Optimization, Variable Ordering, BDDs, Multi-input Adders.

I. INTRODUCTION

Binary Decision Diagrams (BDDs) [1-2-3] represent data structures which are extensively used for compact representation of Boolean functions. The need of the time is that we need to optimize chip area usage in order to be able to fit more functionality into a given chip size. A good variable order is needed that minimizes the size. In this paper, an evolutionary algorithm i.e. genetic algorithm with three versions of the crossover operator has been used for BDD minimization: Order Crossover, Partially Mapped Crossover and Cycle Crossover.

This paper is organized as follows. Section 2 provides an introduction to variable ordering problem Section 3, discusses genetic algorithm, various crossover operators used and its implementation using C++ for BDD ordering and Optimization.

Simulation results are given that demonstrates the efficiency of these operators when compared with each other and other existing variable ordering methods. Final conclusions have been arrived at Section 4.

II. BINARY DECISION DIAGRAMS

BDD is a finite directed acyclic graph (DAG) with a unique initial node as shown in figure 1. All non-terminal nodes are labeled with a Boolean variable. Each non-terminal node has exactly two edges from that node to others, labeled 0 and 1. Shannon decomposition is carried out in each node thus, node F is represented as

$$F = x_i G + x_i' H,$$

where G is the positive cofactor of F with respect to x_i ($G = F_{x_i}$), and H is the negative cofactor of F with respect to x_i ($H = F_{x_i'}$). The node F thus represents the function $F = x_i G + x_i' H$. G is also known as the THEN node and H is also known as the ELSE node. The sink nodes represent the constant functions 0 and 1.

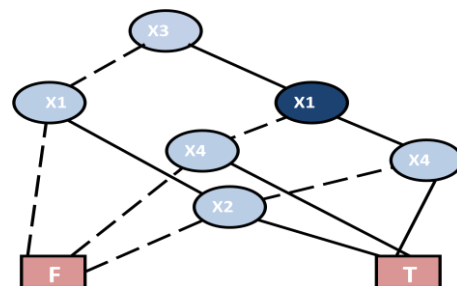


Fig.1 Binary Decision Diagram for the function $F=x_1x_2+x_3x_4$

Ordered binary decision diagram (OBDD), was proposed by Bryant. The OBDD [2] is a DAG in which different variables appear in same order on all paths from the root. If all the equivalent and isomorphic nodes are merged and all the redundant nodes are removed from the OBDD, the resultant BDD is called Reduced ordered BDD. One such structure for the function is shown in figure 2.

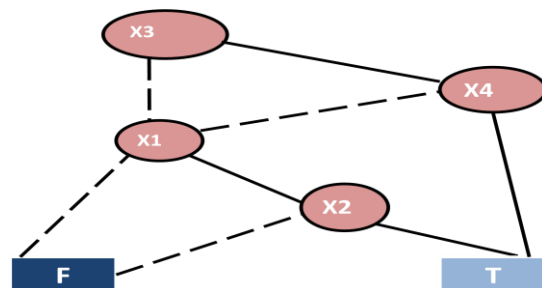


Fig.2 Reduced Binary Decision Diagram for the function $F=x_1x_2+x_3x_4$

A BDD is reduced if it contains no isomorphic sub graphs, and every vertex has distinct children. Reduced, ordered BDDs [3] are canonical representations of Boolean functions for a fixed variable order, two functions are equivalent if and only if their BDDs are identical. An OBDD is converted to an ROBDD using the following reduction rules as shown in figure 3:

- a. Merging equivalent nodes
- b. Merging isomorphic nodes
- c. Eliminating redundant tests

BDD sizes are very sensitive to the order chosen on input variables. Various Heuristics tend to find near optimal variable ordering for BDDs, as finding the best variable ordering is NP-hard.

This paper is focused on BDD minimization [11] using Genetic Algorithm, its implementation and advantages in comparison with the other existing algorithms. Three crossover operators namely Order, Cycle and Partially Mapped Crossover have been used to obtain optimal solutions.

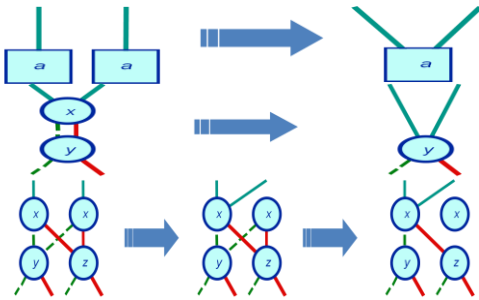


Fig.3 Reduction rules for BDD.

III. BDD MINIMIZATION BASED ON GENETIC ALGORITHM

A. Genetic Algorithm

Genetic Algorithms GA, [9] are a family of excellent optimization techniques based on principle of natural selection and human genetics. The objective of GA is to find an optimal solution to a problem. GAs work by evolving a population of individuals over a number of generations. A fitness value is assigned to each individual depending on the application. For each generation, individuals are selected from existing population for reproduction, the individuals are crossed to generate new individuals, and the new individuals are mutated to introduce new characteristics. The solution with best fitness value will be the most optimum solution. The formulation of Genetic Algorithm for any problem involves the careful and efficient encoding of the solutions to form chromosomes, crossover and mutation operators and a cost function measuring the fitness of the chromosomes in a population. GAs use two basic processes from evolution: inheritance, or passing of features from one generation to other, and competition, or the survival of the fittest, which results in weeding out bad features from individuals in the population.

/*Algorithm GA */

- Formulate initial population
- Randomly initialize population
- Repeat
 - Evaluate objective function
 - Find fitness function
 - Apply genetic operators
 - Reproduction
 - Crossover
 - Mutation
- Until stopping criteria

Fig.4 Basic Genetic Algorithm [8]

Crossover is the primary method of optimization in the genetic algorithm. The performance of the genetic algorithm depends, to a great extent, on the performance of the crossover operator used. Three versions of the crossover operator have been used for BDD minimization: Order Crossover, Partially Mapped Crossover and Cycle Crossover. The performances of these operators have been compared.

A flowchart for GA based approach used for BDD Variable Ordering is shown in figure 5.

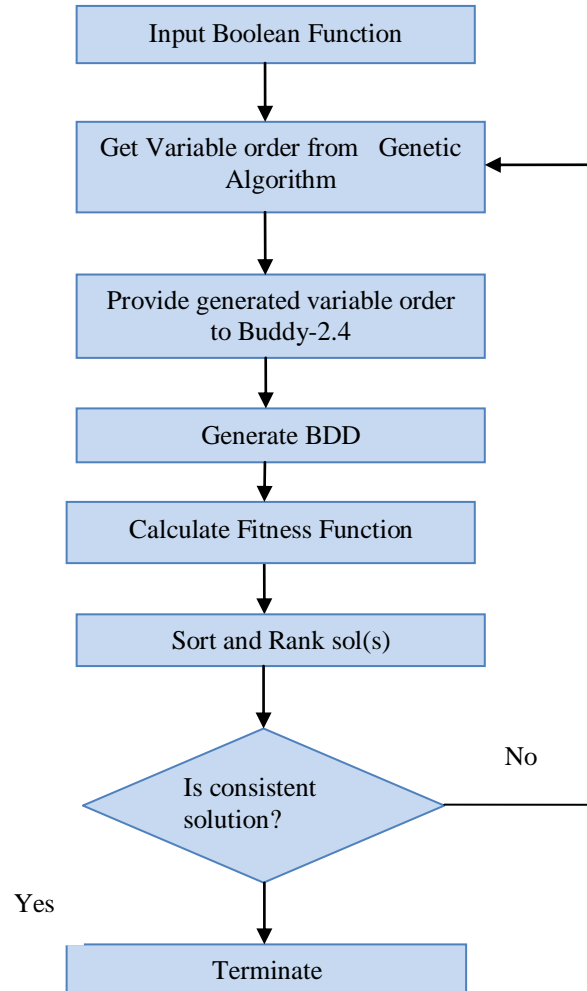


Fig.5 Flow-chart of the Genetic Algorithm based technique used for BDD Variable Ordering [6]

B. Simulation Results

In this section, the result of simulation for Multi-input adder benchmark circuits has been presented. The evolutionary algorithms, using three types of crossover operators, are implemented with C++ codes and simulated using BUDDY 2.4 package on Ubuntu 12.04. The available approaches for BDD ordering and node minimization are compared along with the proposed three versions of crossover operator of the genetic algorithm for BDD minimization on the set of Boolean functions. The proposed approaches give minimal nodes for every function when compared with all other approaches. The simulation results for various multi-input adder benchmark circuits have been displayed in Table 1. The proposed algorithm and approaches provide the optimum variable

order with minimum number of nodes in minimum iterations and number of runs for every function.

TABLE 1
Comparison of GA results with existing algorithms for adders

Multi-input Adder Benchmark Circuits	I/O	Initial node count	WIN2	WIN2ite	WIN 3	SIFT	RANDOM	Proposed GA with order crossover		Upto % node reduction n w.r.t. initial node count	Proposed GA with cycle crossover			Proposed GA with PMX crossover		
								Node Count	CPU Time (sec)		Node Count	CPU Time (sec)	%reduction	Node Count	CPU Time (sec)	%reduction
1-adder	3/2	8	8	8	8	8	8	8	0.01	0	8	0.01	0	8	0.01	0
2-adder	5/3	17	17	17	17	17	17	17	0.02	0	17	0.02	0	17	0.03	0
3-adder	7/4	32	32	34	32	26	34	25	0.1	21.875	32	0.17	0	26	0.22	18.75
4-adder	9/5	63	65	63	46	46	61	35	0.2	44.44	45	0.31	28.57	35	0.52	44.44
5-adder	11/6	126	128	126	61	55	78	49	0.42	61.11	82	0.62	34.92	44	1.78	65.07
6-adder	13/7	253	255	253	85	85	93	79	1.11	68.77	132	1.47	47.82	58	5.94	77.07
7-adder	15/8	508	510	508	94	94	264	135	2.67	73.4	165	3.86	67.51	72	8.38	85.82

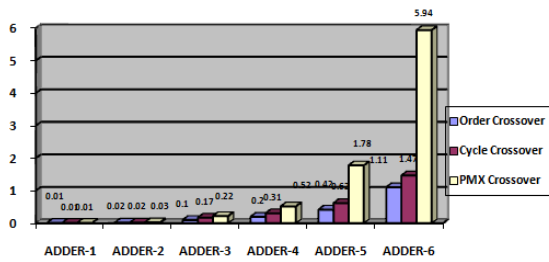


Fig6. Comparison of computation time for various benchmark circuits using different Crossover Operators in GA

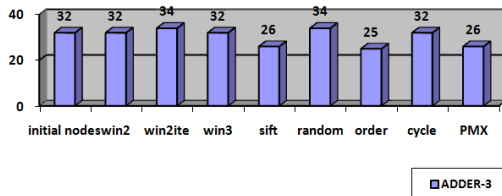


Fig7. Comparison of node count for ADDER-1 circuit using different algorithms.

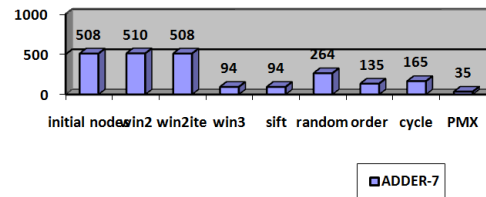
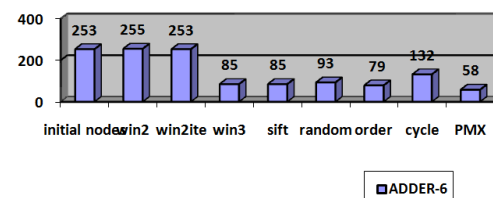
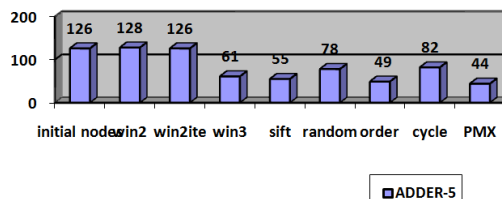
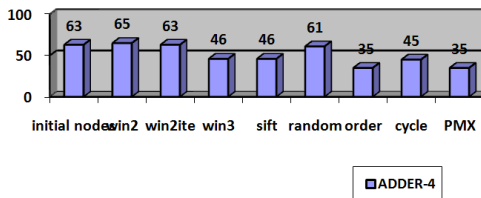


Fig8. Comparison of node count for different multi-input adder circuits using different algorithms.

IV. CONCLUSION

In the present study, three types of crossover operators for the Genetic Algorithm have been proposed and it has been implemented using C++ for ordering and minimization of BDD nodes. The simulations have been done with multi-input adder benchmark circuits. It has been observed that this technique gives minimal number of nodes when compared with other existing algorithms. Also, when the operators are compared, Partially Mapped Crossover gives minimum nodes, while the Cycle Crossover gives minimum computation time for most of the circuits. The order crossover also gives optimum results. So, Genetic Algorithm is best suited for VLSI circuits.

ACKNOWLEDGMENT

With the biggest contribution to this paper, I would like to thank Mrs. Manu Bansal had given me her full support in guiding me with stimulating suggestions and encouragement to go ahead in all the time of the completion of the paper.

REFERENCES

- [1] Sheldon B. Akers, "Binary Decision Diagrams," IEEE Transactions on Computers, vol. C-27, no. 6, June (1978)
- [2] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," IEEE Transactions on Computers, vol. 35, August (1986)
- [3] F. Towhidi, A.H. Lashkari, R.S. Hosseini, "Binary Decision Diagram (BDD)", International Conference on Future Computer and Communication, pg496-499,2009
- [4] William N.N. Hung, Xiaoyu Song, El Mostapha Aboulhamid, and Michael A. Driscoll, "BDD Minimization by Scatter Search," IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol. 21, no. 8, August (2006)
- [5] N. Zhuang, M.S.T. Bente, and P.Y.K Cheung, "Improved Variable Ordering of BDDs with Novel Genetic Algorithm," IEEE

- International Symposium on Circuits and Systems, vol. 3, 12-15 May (1996)
- [6] Saurabh Chaudhary, and Anirban Dutta, “Algorithmic Optimization of BDDs and Performance Evaluation for Multi-level Logic Circuits with Area and Power Trade-offs,” IEEE International Conference on Electronics, Circuits and Systems, July (2011)
- [7] Misagh Takapoo, and M.B Ghaznavi-Ghoushchi, “IDGBDD: The novel use of ID3 to improve Genetic algorithm in BDD reordering,” International Conference on Electrical Engineering / ElectronicComputer Telecommunication and Information Technology,19-21 May (2010)
- [8] Merz P and Freisleben B, “A genetic local search approach to the quadratic assignment problem,” Proceedings of the 7th International Conference on Genetic Algorithms, (1997)
- [9] Tom V Mathew, “Genetic Algorithm,” Report submitted at IIT Bombay.
- [10] P. W. C. Prasad, A. Assi, A. Harb and V. C. Prasad, “Binary Decision Diagrams: An Improved Variable Ordering using Graph Representation of Boolean Functions,” International Journal of Computer Science, Vol. 1, no. 1, pp. 1-7, 2006.
- [11] R. Ebendt, F. Gorschwini and R. Drechsler, “Advanced BDD Minimization”, Springer, New York, 2005.

BIOGRAPHY



Rupinder Kaur currently pursuing M.Tech in VLSI Design from Thapar University, Patiala. I have got my paper “Ordering and Reduction of BDDs Using Various Crossover Operators in GA” published in 3rd National Conference on Advances in Metrology (ADMET-2014). I got the best paper award for the same.