Implementation of GA with Position Based Crossover-PX Technique for Size Optimization of BDD Mapped Adder Circuits

M. Balal Siddiqui, M. T. Beg, S. N. Ahmad

***Abstract*: Binary Decision Diagrams or BDD are data structure used to represent single and multi-output digital circuits. BDD mapped adder circuits are used to represent different adder functions in a digital system. Optimization of adder circuits are done by optimizing the corresponding BDDs. In this work the optimization of BDD Mapped adder circuits are proposed by using genetic algorithm with position-based crossover-PX technique. The main feature of position-based crossover technique is that it is suitable for order-based solution formation. We compared our result with other existing variable order method available in BDD manipulation tool BuDDy-2.4. The result is obtained for Full Adder circuits of 1 to 8-bit size. Experimental results show the improvement of the proposed work over other techniques. The result is quite significant for large circuits i.e. full adder circuit having larger bit size.**

Keywords*: *Adder, BDD, Binary Decision Diagram, Optimization, Variable Ordering.

I. INTRODUCTION

In today's electronic circuit design, adders are one of the important circuit elements. It is an integral part of various arithmetic and mathematical operation unit. In today design environment, as the circuit complexity and data size are continuously increasing, the effective hardware implementation of adders become more area consuming and power hungry. So, the optimization of both, power as well as area, is highly desirable for all the circuit parts. As adders are one of the most used circuit elements, optimization of adder effective area is the prime goal of the circuit design part. For design purpose, the different electronic CAD tools use different data structure to represent the behavioral function of different circuit elements. BDDs are one of the most used data structures for Boolean circuit representation in various electronic design tools [1]. The behavioral function of adders is easily represented by Binary Decision Diagrams [2]. The BDDs are highly compact in nature and can be easily represented using multiplexers [3]. Binary Decision Diagrams are part of decision diagram family having one root node, which represent the output function, two terminal nodes, one represents the Boolean 'high' denoted as '1' and other represents Boolean 'low' denoted as '0'. Binary Decision Diagrams are acyclic in nature and having different internal nodes, which represents different sub-function of the main function and represented by different input variables. Bryant shows that the BDD representation of switching circuits are compact in nature and he proposed some reduction rules by which the BDD can be "Reduced" and "Ordered" and hence the final obtained diagram is known as Reduced and Ordered Binary decision Diagram [4]. The term 'Ordered' means the input variables ordering is same along all the path of the BDDs. The ordering of input variables has very exponential effect on total size of the decision diagram. Finding a good order of input variable is NP-Complete problem. Different authors have implemented various techniques to find an optimal variable order for Binary Decision Diagrams. Genetic algorithm is widely used to find solution for NP-Complete problems. Genetic algorithms are known for using different biological techniques including crossover and mutations. There are different crossover techniques proposed by many authors for optimization of BDD mapped adder circuits [5]. In this paper, a position-based crossover (PX) technique is implemented for optimization of BDD mapped adder circuits of size 1 to 8-bit. The result obtained using proposed crossover technique is compared with the other existing Techniques available in BDD manipulation tool [6].

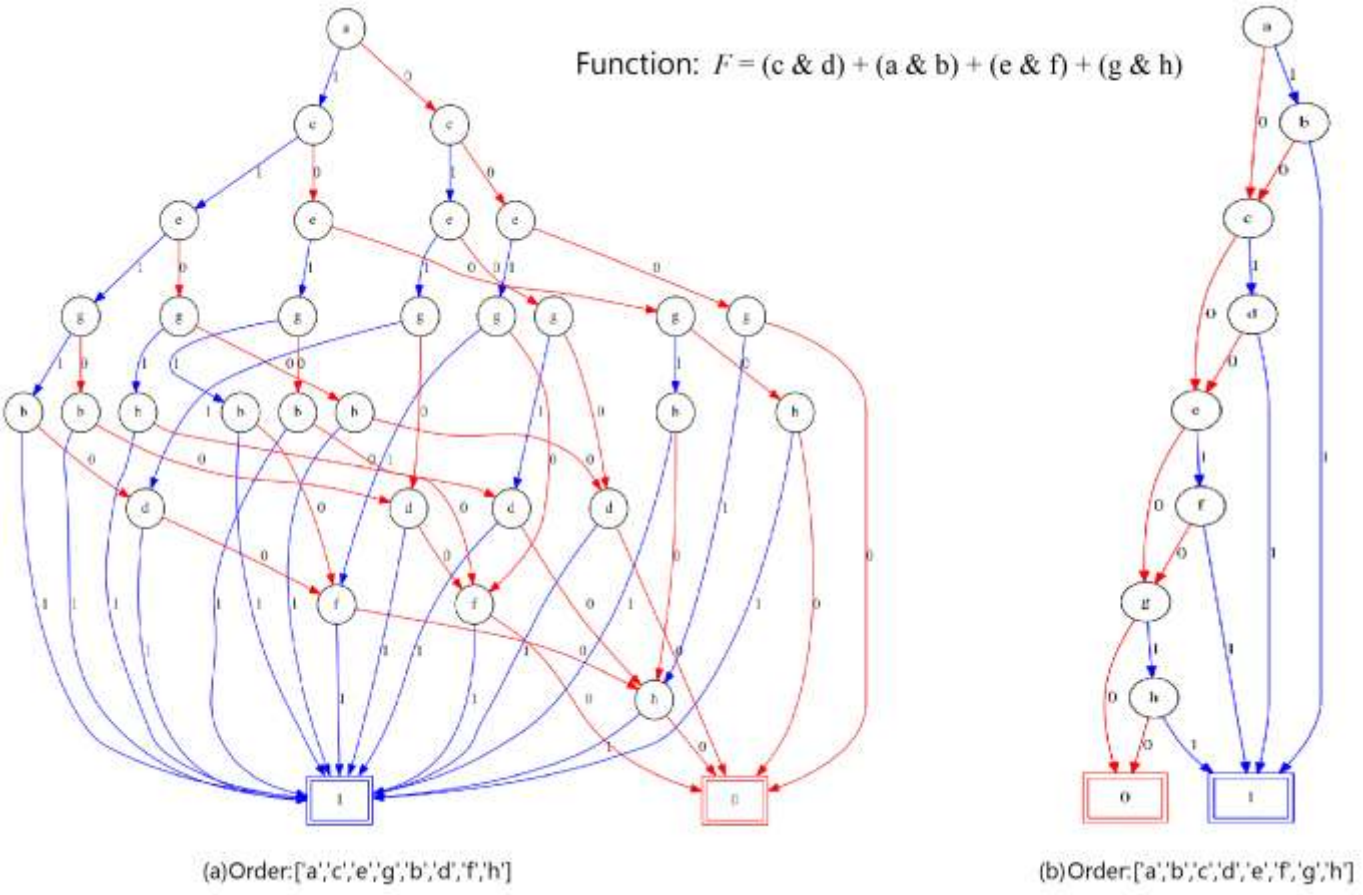
The complete organization of this paper is as follows: Introduction part is in section 1, in Section 2, Binary Decision Diagrams are introduced, Section 3 is about GA method with use of position based crossover technique and problem formulation, Section 4 is about results and implementation, Section 5 is the conclusion of the proposed work.

II. BINARY DECISION DIAGRAMS

A. Binary Decision Diagram

Binary Decision Diagrams are data structure, which are acyclic in nature. As data structure, they are commonly used in different electronics design tools to represent switching circuits. Based on Lee's Binary Decision Tree [7], Bryant has proposed the Ordered and Reduced Decision Tree and termed as ROBDD. Bryant has given three reduction steps, which gives a canonical representation of the switching function. These steps are: 1- Duplicate terminal nodes

Revised Manuscript Received on February 29, 2020.
* Correspondence Author
Md Balal Siddiqui*, Department of Electronics & Communication Engineering, Jamia Millia Islamia, New Delhi, India. Email: balalsid@gmail.com
M. T. Beg, Department of Electronics & Communication Engineering, Jamia Millia Islamia, New Delhi, India.
S. N. Ahmad, Department of Electronics & Communication Engineering, Jamia Millia Islamia, New Delhi, India.



removal (i.e. merging all terminal nodes to either of two terminal nodes (i.e. high node and low node), 2- Duplicate intermediate

Fig. 1. BDD constructed for function, F = (c & d) + (a & b) + (e & f) + (g & h), shown in (1) with two different variable orders (a): ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'] and (b): ['a', 'c', 'e', 'g', 'b', 'd', 'f', 'h']

node removal and 3- redundant nodes removal. [8].

The final obtained decision diagram after application of all three rules is in Reduced and Ordered form and known as Reduced and Ordered BDD (ROBDD) which is simply represented as BDD.

B. Impact of Variable Order on BDD

The size of Binary Decision Diagram depends on the order of input variables [9-12]. Choosing a bad variable order results in an exponential size growth in the decision diagram. Fig 2. shows the BDDs of a function '*F*' shown in (1) with two different variable order ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'] and ['a', 'c', 'e', 'g', 'b', 'd', 'f', 'h']:

$$F = (c \,\&\, d) + (a \,\&\, b) + (e \,\&\, f) + (g \,\&\, h) \quad (1)$$

The function '*F*' represented in (1) is a Boolean function having eight input variable 'a', 'b', 'c', 'd', 'e', 'f', 'g' and 'h'. in Fig 1., the same function '*F*' is represented with two different variable order. The BDD size corresponding to the variable order ['a', 'c', 'e', 'g', 'b', 'd', 'f', 'h'] is 8, while the BDD size corresponding to the variable order ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'] is 32. The effect of variable order become more and more significant with the increase in number of variables.

III. PROPOSED WORK

A. Genetic Algorithm and Problem Formation

Genetic Algorithms are class of optimization algorithms which uses the biological process of offspring generation from parents [13]. For the purpose of generation of offspring, the genetic algorithms use famously two biological operators: crossover and mutation. Crossover generates the offspring from the parent genes by keeping the some part of the characteristic of both the parent genes in the offspring while mutation operator is used for creating a new set of offspring with the some or all part of the offspring as entirely different characteristic which are not present in either of the parent's genes. The set of solutions are represented as biological chromosome, while these chromosomes in a generation acts as parent for the next generation. In this way, in every generation, new offspring are created, and they survive based on their fitness values, which is defined by the corresponding objective function. Fig. 2. shows a chromosome representing the possible solution for Binary Decision Diagram.

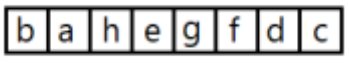


Fig. 2. A Chromosome representing the solution

In Fig 2. The variable order ['b',

'a', 'h', 'e', 'g', 'f', 'd', 'c'] is represented using a chromosome. The position of the variable in the chromosome is the order of the corresponding variable in BDD construction. In this way number of different chromosomes is constructed in a population for each generation. The conventional cyclic mutation is used in this work for the Genetic Algorithm. The working of Genetic Algorithm method is shown using flow chart in Fig 3. Fitness function used in the flow chart is the number of nodes in the BDD, which is the size of the BDD constructed.

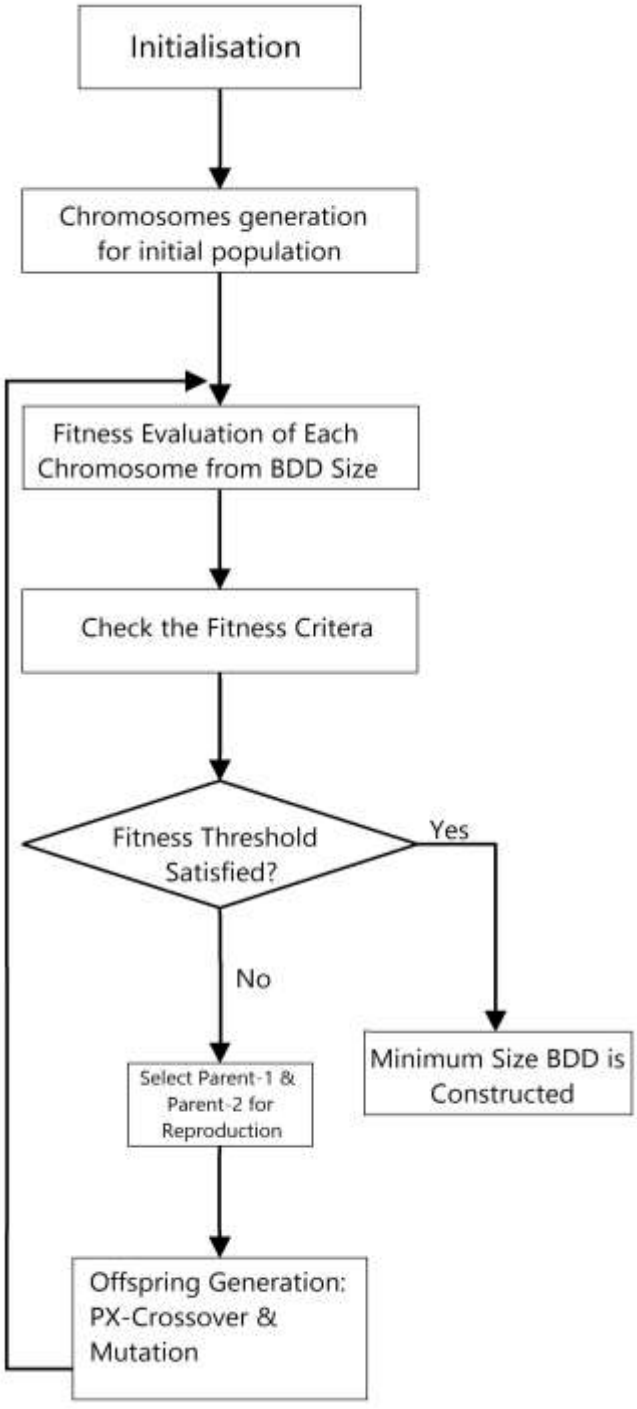


Fig. 3. Flowchart of GA method with Proposed PX-Crossover Operator.

B. Position Crossover (PX):

Position Crossover (PX) is one of the most efficient and fast technique of offspring generation. As compared to other crossover techniques, this crossover technique takes less time and is greedy in nature. It starts by generating the offspring by selecting a random order from the order set of parent-1 and then the remaining vacant variable order position are taken from parent-2 in an offspring. In this way the position crossover generates the new child chromosome from parent chromosomes.

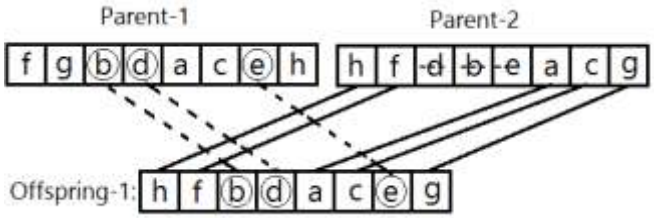


Fig. 4.Position Crossover (PX) Scheme for offspring generation from two parents.

Fig 4. Shows the implementation of the position crossover (PX) method for generation of an offspring chromosome from two parent chromosomes. From Fig 4., we can see that Parent-1 is a chromosome representing a variable order ['f', 'g', 'b', 'd', 'a', 'c', 'e', 'h'] and the Parent-2 is a chromosome representing the variable order ['h', 'f', 'd', 'b', 'e', 'a', 'c', 'g']. As explained, we have randomly selected here three position from Parent-1: position 3rd, 4th and 6th. The variable from these positions from Parent-1 is copied at the same position and in same order in the offspring-1. These variables are highlighted in circles in Fig 4. Now from Parent-2 all the variables, except the variables chosen from Parent-1, are copied to the offspring-1 at the empty positions, keeping the order of the variables same as in Parent-2, while not breaking any rule of constraints. In this way a completely new child/offspring is generated, which will act as a parent for the next generation.

IV. RESULTS AND IMPLEMENTATION

The whole Genetic Algorithm using position crossover is written in C code. We have used BDD manipulation tool, BuDDy-2.4, available at [6]. g++ C compiler is used for programing on an Ubuntu based machine. Result is shown in Table 1. We have implemented this GA method with PX crossover on full adder circuits with bit size 1-to-8. The result is compared with different implemented existing techniques available in BuDDy-2.4 tool. From table 1, we can see that the other methods, *Win2*, *Win2ITE*, *Win3*, *SIFT* and *Random* have 3.5, 3.5, 1.1, 1.1 and 2.4 times respectively more number of nodes compared to the proposed GA with Position Crossover-OX technique.

V. CONCLUSION AND FUTURE SCOPE

This proposed position crossover technique gives better results compared to other inbuild techniques available in BDD manipulation tool

BuDDy-2.4. from Table 1, it is observed that for smaller full adder circuits, like 1-bit FA, 2-bit FA, 3-bit FA and 4-bit FA, there is less possibility of exploring large solution space as the input variable size itself is very low. The improvement is more significant for full adder circuit having large bit size. It is observed that the different other new crossover techniques can also be implemented for BDD size and the possibility of exploring new method for Binary Decision Diagram size reduction is always open.

Sl. No.	Adder circuits	Input/Output	BDD Size					
			Existing Methods [From BuDDy-2.4 BDD Manipulation Tool]					Proposed GA with PX Crossover Method
			WIN2	*WIN2ite*	*WIN3*	*SIFT*	*RANDOM*	
1.	1-bit FA	3/2	8	8	8	8	8	8
2.	2-bit FA	5/3	17	17	17	17	17	17
3.	3-bit FA	7/4	32	34	32	26	34	25
4.	4-bit FA	9/5	65	63	46	46	61	45
5.	5-bit FA	11/6	128	126	61	55	78	49
6.	6-bit FA	13/7	255	253	85	85	93	63
7.	7-bit FA	15/8	510	508	94	94	264	97
8.	8-bit FA	17/9	1001	1001	287	283	845	273
Size increased in other existing methods compare to proposed method			**3.5**	**3.5**	**1.1**	**1.1**	**2.4**	**1**

Table- I: Result Showing Proposed GA with PX Crossover compared with different existing methods

REFERENCES

1. S. Chaudhury and A. Dutta, "Algorithmic optimization of BDDs and performance evaluation for multi-level logic circuits with area and power trade-offs," Scientific Research, vol. 2, pp. 217-224, 2011.
2. 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, 2017, pp. 218-221.
3. M. B. Siddiqui, S. N. Ahmad and M. T. Beg, "Modified GA method for variable ordering in BDD for MIMO digital circuits," 2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT), Pune, 2016, pp. 378-382.
4. R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," in IEEE Transactions on Computers, vol. C-35, no. 8, pp. 677-691, Aug. 1986.
5. H. Sakanashi, T. Higuchi, H. Iba and Y. Kakazu, "Evolution of Binary Decision Diagrams for digital circuit design using genetic programming," ICES, vol 1259, pp. 470-481,1996
6. BuDDy:A Binary Decision Diagram Package By Jorn Lind-Nielsen, http://sourceforge.net/projects/buddy(online).
7. C. Y. Lee, "Representation of switching circuits by Binary decision programs," in The Bell System Technical Journal, vol. 38, no. 4, pp. 985-999, July 1959.
8. M. B. Siddiqui and M. Bansal, " BDD ordering: A method to minimize BDD size by using improved initial order," 2013 International Journal of VLSI and Embedded Systems (IJVES), vol. 04, issue. 03, pp. 127-130, May 2013.
9. M. B. Siddiqui M. T. Beg and S. N. Ahmad, "A Global Search Algorithm Combined with SIFT Algorithm for Optimization of MUX BDD Switching Circuits," International Journal of Scientific & Technology Research, vol. 1, issue 1, pp. 2293-99, Nov 2019.
10. M. B. Siddiqui, "Implementation of an improved initial order in various dynamic variable ordering techniques for BDDs," M.Tech. dissertation, Dept. Elect & Comm. Eng., Thapar Univ., Patiala, Punjab, 2013
11. M. B. Siddiqui and M. Bansal, " BDD ordering: A method to minimize BDD size by using improved initial order," 2013 International Journal of VLSI and Embedded Systems (IJVES), vol. 04, issue. 03, pp. 127-130, May 2013.
12. R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," in IEEE Transactions on Computers, vol. C-35, no. 8, pp. 677-691, Aug. 1986.
13. O. Brudaru, R. Ebendt and I. Furdu, "Optimizing variable ordering of BDDs with double hybridized embryonic genetic algorithm," IEEE International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, pp. 167-173, 2010.

AUTHORS PROFILE

M Balal Siddiqui received B.E. degree in E&C Engg, from VTU, Karnataka, in 2009, M.Tech. degree in VLSI Design & CAD from Thapar University, Patiala in 2013. He is now with Jamia Millia Islamia, New Delhi as a research scholar. His current research interest includes Digital VLSI circuits, VLSI architecture and low power algorithms.

Mirza Tariq Beg received B.Sc. (Engineering) Degree in Electronics and Communication Engineering from AMU, Aligarh, in 1985, M.Tech. degree in Microwave Electronics from Delhi University, N. Delhi in 1987, and Ph.D. degree from Jamia Millia Islamia, N. Delhi, in 2003. He is currently professor and head of the Deptt of E&C Engg, Jamia Millia Islamia. He has over 50 publications in international journals/conferences. His research interests include data communication, wireless communication, and computer networks.

Syed Naseem Ahmad received B.Sc. (Engineering) degree in Electrical and M.Sc. (Engineering) degree in Control and Instrumentation from AMU, Aligarh, and Doctorate degree in E&C Engg from Jamia Millia Islamia,
New Delhi in 1975,1978 and 2005, respectively. He recently retired as a professor from Deptt of E & C Engg, Jamia Millia Islamia, N Delhi. He has over 90 publications in international journals/conferences. His research interests include Digital Image Processing, Wireless Networks and VLSI Design.