

## Algorithmic Reduction And Optimization Of Logic Circuit In Area And Power Tradeoffs' With The Help Of Binary Decision Diagram

Sukhmanjeet Kaur

Department of ECE,  
Thapar University  
Patiala, 147004, India

Manu Bansal

Department of ECE,  
Thapar University,  
Patiala, 147004, India

### Abstract

*This paper presents an algorithmic technique based on hybridizing Symbolic Manipulation Techniques based on BDDs with more traditional explicit solving algorithms. Boolean functions can be graphically manipulated to reduce the number of nodes, hence the area, when implemented as Binary decision diagrams. So here, ordering of BDD nodes plays a very important role. Most of the algorithms for variable ordering of OBDD have focus on area minimization. Hence, for minimizing the power consumption, suitable input variable ordering is required. So, to find an optimal variable, three algorithms have been used namely genetic algorithm based technique, a branch and bound algorithm and a scatter search algorithm in this paper. Experimental results show a substantial reduction in area and power. Moreover, a comparison is made between all the above techniques.*

**Keywords:** Binary decision diagram (BDD), variable order, (GA) genetic algorithm, branch&bound algorithm (B&B), variable ordering, Scatter-search, area and power tradeoffs.

### 1.Introduction

Binary Decision diagrams (BDD) are the data structures that are used to represent Boolean functions and also in the area of logic synthesis, verification and testing. The success of this

approach is due to the fact that theoretical trees can be transformed into rooted, directed acyclic graph. The BDD is said to be 'reduced' if the following two rules have been applied to its graph: Merge any isomorphic sub graphs and eliminate any node whose two children are isomorphic.

This reduction results in much simpler circuit decomposition structure that still represents the initial Boolean function. If the original BDD is ordered, then only the reduction works, ie., if the different Boolean variables appear in the same order on all the paths going from the roots to the leaves. A BDD can be applied in either reduced order (ROBDD), in specific order (obdd) or in canonical form. The OBDD is said to be ROBDD, if all the redundant nodes and all identical nodes are shared. The BDD originated in logic studies for manipulation and computations of logical expressions were used very early in the domain of switching circuit design. Initial efforts were those of Lee and Akers, followed by those of Bryant [1] who emphasized the use of BDD as a fundamental circuit decomposition tree.

In this context, a number of algorithms exist for the problem of variable ordering. These variable ordering algorithms are broadly divided into- static variable ordering, dynamic variable ordering and

evolutionary algorithms. Using BDD, logic verification for larger set of networks has been carried out and from the concept of circuit topology, most of the variable ordering algorithms are based on depth first traverse through a circuit from primary outputs to primary inputs [2, 3]. A dynamic variable ordering technique for ordered BDD is described by R.Rudell [4]. He proposed two OBDD minimization algorithms called sifting algorithm and window permutation algorithm, both beneficial in reducing the size of OBDD. Basically, there is an enhancement of an OBDD package, where OBDD package maintains the order of the variables. A linear sifting algorithm, for the optimization of decision diagrams is proposed [6]. The algorithm tells the efficiency of sifting and the power of linear transformation and also useful to extract a linear filter and achieve the necessary decomposition. A genetic algorithm (GA) is implemented to find a variable ordering that reduces the size of ordered binary decision diagrams [7]. This paper shows that GA performs very well and is a practical alternative algorithm for variable ordering. Nagisa ishiura and hiroshi sawada [8] describes a new algorithm where the optimum order is found by the exchange of variables of BDD and gradual improvement methods for minimizing the binary decision diagrams (BDDs). Minimization of BDD by scatter search has been presented in [9]. An improved branch and bound algorithm for exact BDD minimization is given by Rudiger Ebendt and Wolfgang Gunther [11] which minimizes the computations. Here, we have observed that the variable ordering algorithm not only reduces the size of BDD but also reduces the power.

## 2. Problem statement

The problem of finding an optimal variable ordering for Binary Decision Diagrams (BDD) or Multi-Valued Decision Diagrams (MDD) is widely

known to be NP-Complete. This paper presents a survey of static heuristic techniques applied to ordering the variables of the BDD/MDD under construction in order to minimize the overall size of the resulting decision diagram.

Much research has been carried out on devising heuristic and meta-heuristic approaches for establishing near-optimal variable ordering for BDD/MDD construction. This section presents some of these techniques, grouped into subsections based on their general approach. Our problem will involve finding a clustering technique for BDD so that which take into account the events of the function in correlation with the actual variables seem to be generally more effective in reducing overall sizes of resulting decision diagrams, as the event span metrics seem to promote a more holistic summarization of the properties of the circuits/functions. Additional future metrics that take into consideration not only the clustering of events and variables, but also the positioning of those variables in the resulting ordering

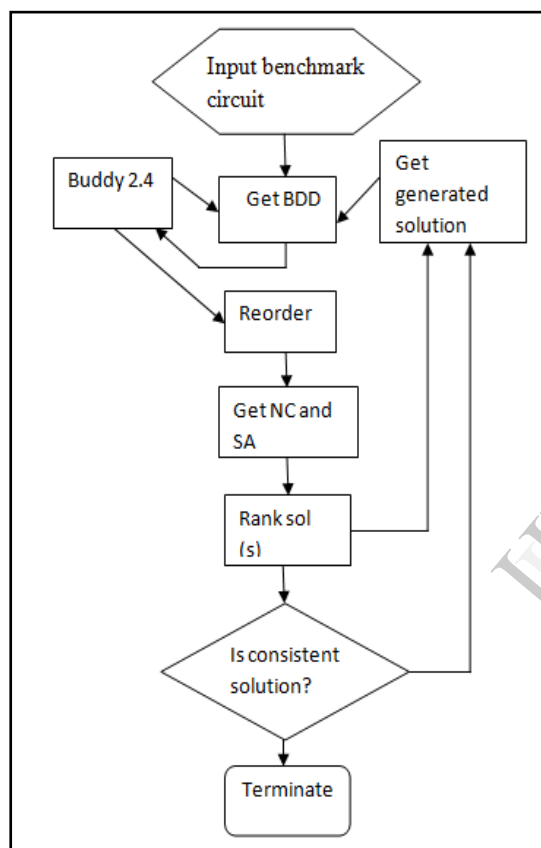
## 3. Approaches used

We have used different algorithmic approaches for efficient ordering of variables in OBDD, namely, the genetic algorithm which is an optimization technique, a branch and bound approach, scatter search technique and dynamic variable ordering approach. At starting, variable ordering problem has been put together in the framework of GA and constitute a GA-based program to obtain the best possible order decided by the minimal node count and power consumption of the resulting BDD. This will be followed by the experimentation with a number of benchmark circuits. Then, we will go for the same variable ordering problem by using a branch and bound (BB) based algorithmic approach which is also an excellent optimization technique for multi-objective problems and has a finite but

usually very large number of feasible solutions. A BB algorithm searches the complete space of solutions (exact method) for a given problem for optimum solution. [19]

### 3.1 Genetic Algorithm Based Approach

Genetic Algorithm (GA) is an optimization technique based on principle of natural selection and natural genetics. Flow chart for genetic-based technique is drawn in the figure 1.



**Figure1. Flow chat for Genetic-based technique [19]**

They start with an initial population (solution space) consisting of a set of randomly generated solutions. Based on some reproductive plan especially, the cross- over and mutation, they are allowed to evolve over a number of generations. After each generation, the chromosomes are evaluated based on some fitness criteria. Depending upon the selection policy and fitness value, the set of chromosomes for next generation

are selected. Finally, the algorithm terminates when there is no improvement in solution over a fixed number of generations. The best solution at that generation has been accepted as the solution produced by GA [19]. The proposed GA-based scheduler takes advantage of the parallelism of decomposable data grid applications to achieve the desired performance level. We have evaluated the proposed approach comparing with other algorithms. Simulation results show that the proposed GA-based approach can be a competitive choice for scheduling large data grid applications in terms of both scheduling overhead and the relative solution quality as compared to other algorithms.

### 3.2 Branch and Bound Approach

In this section, we will apply branch and bound algorithm for the current variable ordering problem and optimal trade-off between area and power. A B&B algorithm searches the complete space of solutions for a given problem for the best solution. However, explicit enumeration is normally impossible due to exponentially increasing number of potential solution. The use of bounds for the function to be optimized combined with the value of the current best solution enables the algorithm to search parts of solution space only implicitly. The explored subspaces are represented as nodes in a dynamically generated search tree, which initially only contains the root, and each iteration of a classical B&B algorithm processes one such node. The iteration has three main components: Selection of the node to process, Bound calculation and Branching.

The sequence of these may vary according to the strategy chosen for selecting the next node to process. If the selection of next subproblem is based on the bound value of the subproblem, then the first operation of iteration after choosing the node is branching iteration. For each of these, it is

checked whether the subspace consists of single solution, in which case it is compared to the current best solution keeping the best of those.

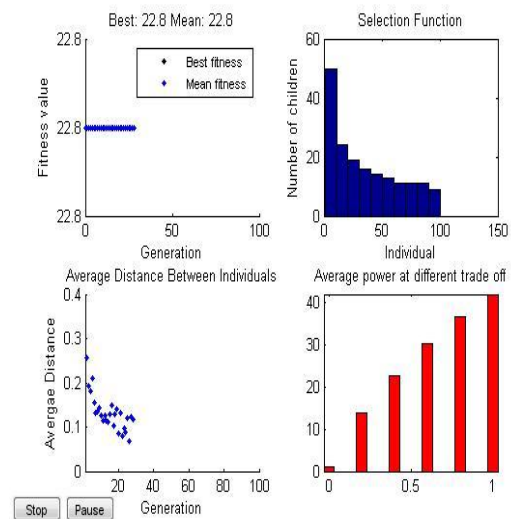
### 3.3 Scatter Search

Scatter search operates on a set of solutions, the reference set, by combining these solutions to create new ones. The main mechanism for combining solutions is such that a new solution is created from the linear combination of two other solutions. Unlike a “population” in genetic algorithms, the reference set of solutions in scatter search tends to be small. In genetic algorithms, two solutions are randomly chosen from the population and a “crossover” or combination mechanism is applied to generate one or more offspring. A typical population size in a genetic algorithm consists of 100 elements, which are randomly sampled to create combinations. In contrast, scatter search chooses two or more elements of the reference set in a systematic way with the purpose of creating new solutions. Since the combination process considers at least all pairs of solutions in the reference set, there is a practical need for keeping the cardinality of the set small.

## 4. Experimental Results

An optimal variable order has a greater impact on power minimization also, as because, node switching and leakage is dependent on the number of BDD nodes and its order. Majority of the heuristic techniques discussed here has stressed only upon the size or complexity of the resulting BDD. However, power is considered to be one of the critical design issues especially when there is drastic device scaling and increasing use of portable, battery operated digital devices in recent times. In this paper, due weight age is given to both the area (complexity) and power consumption of

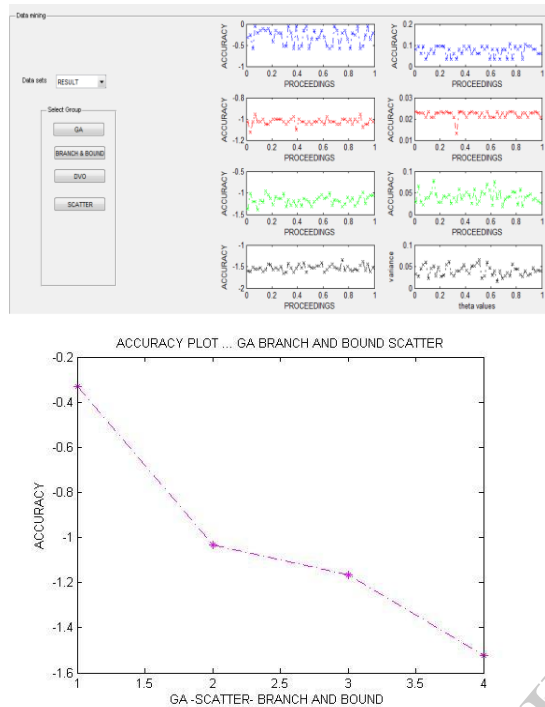
the resulting BDD after optimization and here figure 1 shows the average power reduction at different trade offs.



**Figure2. Graph of average power at different trade off and fitness value**

All the programs has been run with matlab codes. In this paper, firstly we build an n-bit adder circuit with  $N$ (number of bits)= 6, created BDD for the same and after the results, the actual number of nodes created were 63, the number of path variables were 126. Then created ROBDD for the circuit and we found reduced nodes to be 21 and reduced paths to be 41. Thus, new number of nodes were 42 and new number of paths were 85. Finally, after reducing the number of nodes in the circuit and calculating switching activity of the circuit, genetic algorithm is implemented on the circuit for feature optimization. Switching activity for the circuit found was 16. Switching activity is calculated by :  $2(* \text{ minterm} * \text{ maxterm})$ . Average power reduction obtained was 24.50. Best mean for fitness value calculated is 22.8 as shown in figure 2. After this, we compared different algorithms like genetic algorithm, branch & bound algorithm, scatter search and dynamic variable ordering. Best accuracy in terms of power and area found was of genetic algorithm.

Comparison is made between all the above discussed techniques and results of those are shown in figure 3. It shows GA attains highest accuracy in terms of power and area as compare to other techniques and scatter search has less accuracy.



**Figure3. Graph showing accuracy of GA, Branch and bound, scatter search**

## 5. Conclusion

ROBDD is applied on the n-bit adder circuit and we have calculated new number for nodes, new number of paths and the switching activity of the circuit. Then GA has been applied on the circuit to calculate the average reduction power that is found to be 24.5. Finally, the comparison with other established techniques such as, branch & bound, scatter search technique and dynamic variable ordering has been done and found that the proposed genetic algorithm technique is superior compared to others in fulfilling the objectives.

Exhaustive experimentation has been done with ISCAS93 benchmark circuits to see the

effectiveness of the proposed techniques for area and power optimization using BDD.

## 6. References

- [1] S. B. Akers, "Binary Decision Diagram," IEEE Trans. Computers, vol. 27, 1978.
- [2] S. Malik, A. R. Wang, R. K. Brayton and A. Sangiovanni-Vincentelli, "Logic Verification Using Binary Decision Diagrams in a Logic Synthesis Environment," Proceedings of International Conference on Computer-Aided Design, Santa Clara, 7-10 November 1988, pp. 6-9.
- [3] M. Fujita, H. Fujisawa and Y. Matsunaga, "Variable Ordering Algorithms for Ordered Binary Decision Diagrams and Their Evaluation," IEEE Transactions on Computer-Aided Design, Vol. 12, No. 1, 1993, pp. 6-12.
- [4] R. Rudell, "Dynamic Variable Ordering for Ordered Binary Decision Diagrams," Proceedings of the 1993 IEEE/ACM International Conference on Computer-Aided Design, Santa Clara, 7-11 November 1993, pp. 42-47.
- [5] H. Fujii, G. Ootomo and C. Hori, "Interleaving Based Variable Ordering Methods for Ordered Binary Decision Diagrams," Proceedings of the 1993 IEEE/ACM International Conference on Computer-Aided Design, Santa Clara, 7-11 November 1993, pp. 38-41.
- [6] C. Meinel and F. Somenzi, "Linear Sifting of Decision Diagrams," Proceedings of the 34th Annual Automation Conference, Anaheim, 9-13 June 1997, pp. 202-207.
- [7] R. Drechsler and N. Göckel, "Minimization of BDDs by Evolutionary Algorithms," International Workshop on Logic Synthesis (IWLS), Lake Tahoe, 1997.
- [8] N. Ishiura, H. Sawada and S. Yajima, "Minimization of Binary Decision Diagrams Based on Exchange of Variables," 1991 IEEE International Conference on Computer-Aided Design, Santa Clara, 11-14 November 1991, pp. 472-475.
- [9] William N.N. Hung, X. Song, "BDD Minimization by scatter search," IEEE Transaction on Computer-Aided design of integrated circuits and systems. vol.21, no. 8, August 2002.

- [10] R. Drechsler, B. Becker, N. Göckel, "Learning Heuristics for OBDD Minimization by Evolutionary Algorithms," Lecture Notes in Computer Science, vol. 1141, 1996,
- [11] W. Günther and R. Drechsler, "Improving EAs for Se-quencing Problems," Proceedings of the Genetic and Evolutionary Computation Conference, 2000.
- [12] B. Beate, L. Martin and W. Ingo, "Simulated Annealing to Improve Variable Orderings for OBDDs," Proceedings of the International Workshop on Logic Synthesis, May 1995, pp. (5-1)-(5-10).
- [13] R. Drechsler, M. Kerttu, P. Lindgren and M. Thornton, "Low Power Optimization Techniques for BDD Mapped Circuits Using Temporal Correlation," Canadian Journal of ECE, vol. 27, no. 4, 2002, pp. 159-164.
- [14] R. Ebdet, F. Gorschwim and R. Drechsler, "Advanced BDD Minimization", Springer, New York, 2005.
- [15] 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, 2006, pp. 1-7.
- [16] S. Chaudhury and S. Chattopadhyay, "Output Phase Assignment for Multilevel Multi-output Logic Synthesis with Area and Power Trade-offs," 2006 Annual IEEE India Conference, New Delhi, 15-17 September 2006, pp. 1-4.
- [17] M. Rice and S. Kulhari, "A Survey of Static Variable Ordering Heuristics for Efficient BDD/MDD construction, Technical Report ,2008.
- [18] O. Brudaru, R. Ebdet and I. Furdu, "Optimizing Variable Ordering of BDDs with Double Hybridized Embryonic Genetic Algorithm," Proceedings of 12th IEEE International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, 23-26 September 2010, pp. 167-173.
- [19] S. Chaudhury, A. Dutta, "Algorithmic optimization of BDDs and Performance evaluation for multi-level logic circuits with area and power trade-offs", Circuits and Systems, vol.2, pp. 217-224, 2011.