ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 5 may, 2014 Page No. 6132-6139

Algorithmic Reduction and Optimization of Logic Circuit in area and Power Tradeoffs' with the Help of BDD

Gaurav Sharma
Research Scholar, Thapar University

Abstract:

The design complexity and increasing speed of very-large-scale integration (VLSI) chips implies a significant increase in the power consumption. So, many different design approaches have been developed by researchers to reduce the power. This paper presents an algorithmic technique based on hybridizing Symbolic Manipulation Techniques based on BDDs with more traditional explicit solving algorithms. To validate the approach, the graph colouring problem has been selected as a hard-to-solve problem, and an optimized solution based on hybrid techniques has been implemented. Experimental results on a set of benchmarks derived from the CAD for VLSI area show the applicability of the approach to graphs with millions of vertices in a limited CPU time. 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. Also, the switching activity of the circuit is calculated. Moreover, a comparison is made between all the above techniques.

Keywords: BDDs, variable order, genetic algorithm, branch& bound algorithm, variable ordering, 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.
- 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.

Another variable ordering algorithm is based on cover patterns and selects most binate variable first for two-level circuits and depth traverse of circuits for multi-level circuits [4, 5]. As the result depends on initial variable order, so this approach is quite effective. Hence, the variable ordering methods to find good initial variable from the network topology are very important. 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. Minimizing the number of one- path of BDD is accomplished by evolutionary algorithm (EA) in [12]. 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 is 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. However, in the current variable ordering problem for optimizing area and power, in combinational logic circuits realizes ad BDD.

3.1 Genetic Algorithm Based Approach

Genetic Algorithms (GA) are stochastic optimization based on principle of natural selection and natural genetics. 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 is accepted as the solution produced by GA. Data grid technology promises geographically distributed scientists to access and share physically distributed resources such as compute resource, networks, storage, and

most importantly data collections for large-scale data intensive problems. Because of the massive size and distributed nature of these datasets, scheduling data grid applications must consider communication and computation simultaneously to achieve high performance. In many data grid applications, data can be decomposed into multiple independent sub datasets and distributed for parallel execution and analysis. We exploit this property and propose a novel genetic algorithm based approach that automatically decomposes data onto communication and computation resources. The proposed GA-based scheduler takes advantage of the parallelism of decomposable data grid applications to achieve the desired performance level. We evaluate 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. Flow chart for genetic-based technique is drawn in the table 3.1.1

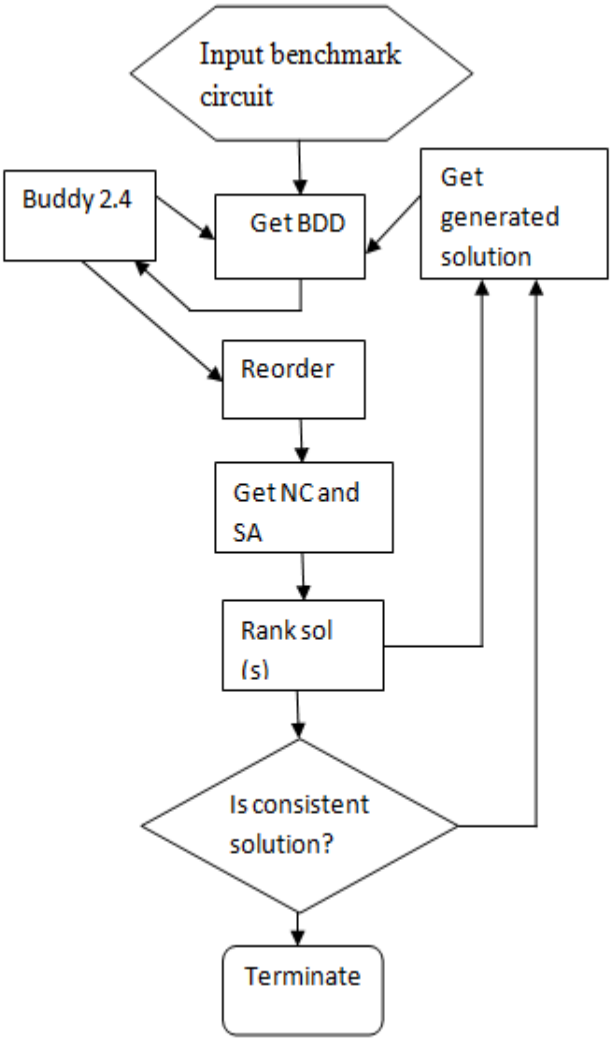

Figure1. Flow chat for Genetic- based technique

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 sub problem is based on the bound value of the sub problem, then the first operation of a iteration after choosing the node is branching. i.e. subdivision of the solution space of the node into two or more subspaces to be investigated in a subsequent 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. Otherwise the bounding function for the subspace is calculated and compared to the current best solution.

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 Result:

The GA based program as defined above is implemented with matlab codes and experimented by running on a Pentium core-2 duo processor having 1GB of RAM with a number of nodes. The choice of the task which is to be serviced next is done at the run-time. The algorithms performing such scheduling differ in the assumptions about the complexity of task and task behaviour.After reducing the number of nodes in the ciruit nad calculating switching activity of the circuit,genetic algorithm is implemented on the circuit for feature optimization. Average power at different trade off is calculated and shown in figure 2. Best mean for fitness value calculated is 22.8. 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.

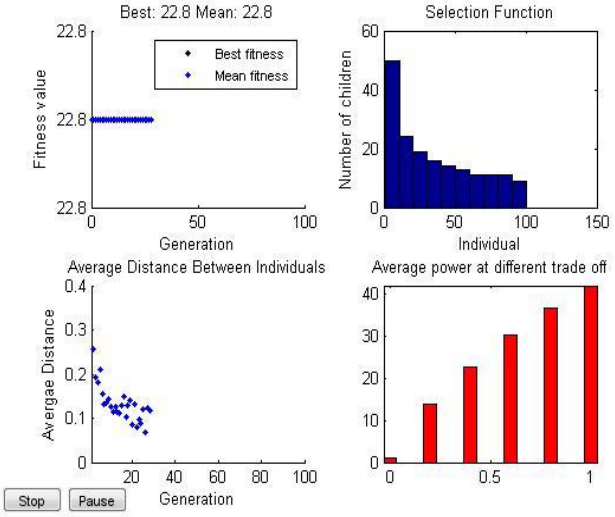

Figure 2. Graph of average power at different trade off and fitness value

Improving the variable ordering of BDDs is NP-Complete and finding the best order is NP-hard. How-ever, the most tedious job in case of OBDDs is to find an optimal variable order. 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. 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.

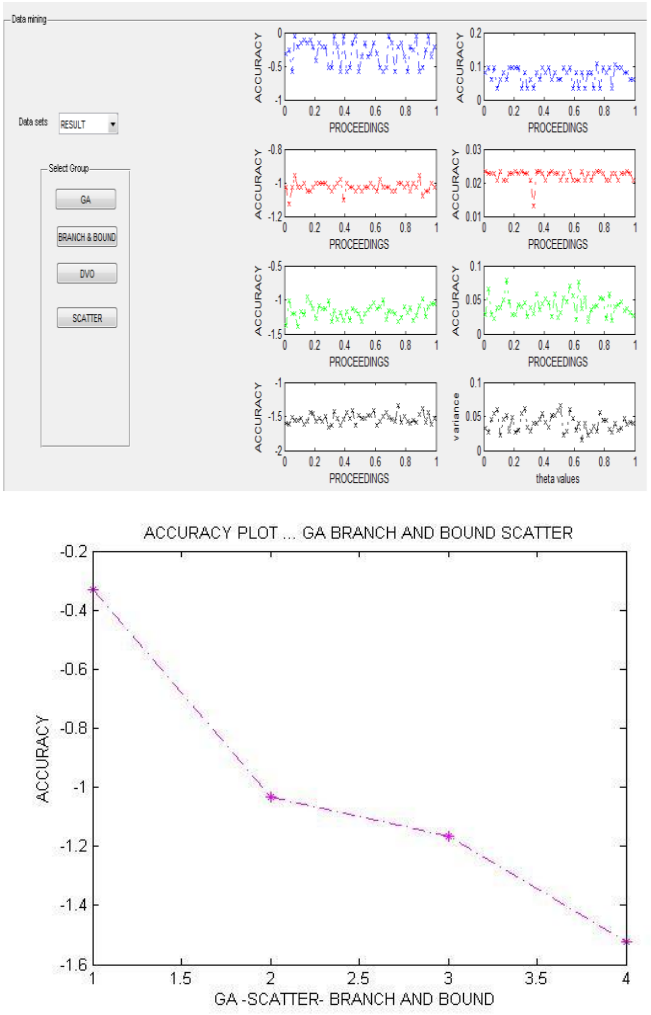

Figure3. Graph showing accuracy of GA, Branch and bound, scatter search

5. Conclusion

Presented here two techniques for BDD optimization namely, GA based optimization and Branch and Bound based Greedy optimization. Exhaustive experimentation has been done with ISCAS93 benchmark circuits to see the effectiveness of the proposed two techniques for area and power optimization. Finally, the comparison with other established techniques such as, scatter search technique and dynamic variable ordering have been done and found that the proposed two techniques are superior compared to others in fulfilling the objectives.

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. Sangio-varmi-Vincentelli, "Logic Verification Using Binary Deci-sion 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. Matsnnaga, "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 Bi-nary 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 intergrated 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] M.Hilgemeier, N.Drechsler, R.Drechsler, "Minimizing the number of one-paths in BDDs by an Evolutionary algorithm,"

[13] M. A. Thornton, J. P. Williams, R. Drechsler, N. Drechsler and D. M. Wessels, "BDD Variable Reordering Based on Probabilistic and Evolutionary Algorithms," 1999 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, 22-24 August 1999, pp. 381-387.

[14] 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).

[15] M. Fujita, Y. Matsmraga and T. Kakuda, "On Variable Ordering of Binary Decision Diagrams for the Applica-tion of Multi-level Logic Synthesis," Proceedings of European Design Automation Conference, Amsterdam, 25-28 February 1991, pp. 50-54.

[16] 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.

[17] 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.

[18] P. W. C. Prasad, M. Raseen, A. Assi and S. M. N. A. Senanayake, "BDD Path Length Minimization Based on Initial Variable Ordering", Journal of Computer Science, vol. 1, no. 4, 2005, pp. 520-528.

[19] M. Rice and S. Kulhari, "A Survey of Static Variable Ordering Heuristics for Efficient BDD/MDD construction, Technical Report ,2008.

[20] 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.

[21] O. Brudaru, R. Ebendt 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.

[22] R. Ebendt, F. Gorschwin and R. Drechsler, "Advanced BDD Minimization", Springer, New York, 2005.