ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 3, March 2014

Effect of various Crossover operators in Memetic algorithm on Multi-input adders

Anamika[1], Mrs. Manu Bansal[2]

Electronics & Communication Department, Thapar University, Patiala, India[1]

Assistant Professor, Electronics & Communication Department, Thapar University, Patiala, India[2]

Abstract: Binary Decision Diagram (BDD) is an important data structure play a major role in reduction of nodes and computation time. In this paper, Memetic algorithm with various crossover operators has been proposed for reduction of BDD node count and the execution time. The results have been compared using various Crossover operators (namely ordered, partially mapped and cyclic) for multi-input Adder Benchmark Circuits.

Keywords: Memetic Algorithm, Optimization, Crossover operators, BDDs, Multi-input Adder benchmark circuits.

I. INTRODUCTION

Binary decision diagrams (BDDs) are the most memory-efficient canonical data structure for Boolean functions known so far [1]. The fact that the size of BDDs is highly sensitive to variable ordering has spurred a large body of research on heuristic variable ordering. In 1993, Rudell gave a powerful reordering algorithm called sifting and showed that there is an efficient way to change variable ordering dynamically while BDDs are constructed, which significantly enlarged the applicability of BDDs. Low power consumption has emerged as key design parameter for digital VLSI systems. The trend has been to develop methodologies and techniques which maintain a circuit's throughput and area constraints while achieving some desired level of power efficiency. Probability-based power analysis tool depend heavily on BDDs to determine signal activity, while historic uses of BDDs have been in digital circuit design areas of synthesis, verification, and testing. As early as in 1959, Lee introduced the concept of Binary Decision Programs and set of rules to transform these programs into switching circuits. Bryant demonstrated that BDDs have two very useful properties. First, BDDs are canonical; given the BDDs for two circuits are equivalent in behavior if BDDs are identical. Second, BDDs are effective at representing combinatorially large sets, which is useful in FSM equivalence checking and logic minimization. Evolutionary algorithms (EAs) are powerful search algorithms [2]. Unlike conventional search approaches, they simultaneously consider many points in the search space so as to increase the chance of global convergence Therefore, EAs exhibit a good potential of global exploration, optimization problems such as highly nonlinear, non-differentiable and multi-modal optimization problems [2-3]. However, EAs are frequently incapable of finding a precise solution in default of local search mechanisms. In recent years, Memetic algorithms (MAs) have been receiving increasing attention from the evolutionary computation community. MAs are hybrid EAs that combine genetic operators with local search methods. The local search methods are used to perform the local refinement procedures [4]. Therefore, they can be regarded as the integration between populations based EAs and local search methods. With global exploration and local exploitation in search space, MAs are capable of obtaining more high-quality solutions [4].

II. BDD OPTIMIZATION

A binary decision diagram is a rooted, directed acyclic graph with one or two terminal nodes of out-degree zero labeled 0 or 1 (or residue variable in positive or negative form), and a set of variable nodes *u* of out-degree two. BDD has only one node with no parent called the root node.

Figure 1 represents a simple BDD of function $F = x_1.x_2' + x_1'x_2$ of two input variables x_1,x_2 with a vector of function y= 0101

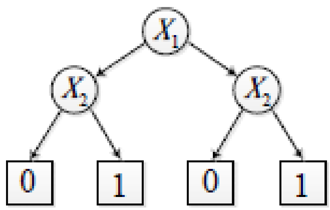


Figure 1. BDD example

By evaluating each term of the function, a complete binary vector y of the function *f* can be determined. This procedure can also be applied similarly in the reversed order, thus it is possible to build a BDD from a binary vector of a function [5].

BDD can be optimized in many ways. BDD optimization method can be divided into two main categories [7]:

(1) BDD ordering - results in Ordered Binary Decision Diagram (OBDD), which respects a given order of input variables. The size of the BDD depends heavily on its input variable ordering.

(2) BDD reduction - when applied on OBDD results in Reduced OBDD (ROBDD). ROBDD respects these two rules:

a Uniqueness: no two distinct nodes u and v represents the same variable and have the same left and right successor, i.e.: *var(u)= var(v), left(u)=left(v), right(u)=right(v)* what implies u=v

b Non-redundancy: no variable node u has the identical left and right successor, i.e. :*left(u) = right(u)*

BDD variable reordering problem is used in the proposed design of comparing the effect of different crossover operators on node count and CPU computation time.

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 3, March 2014

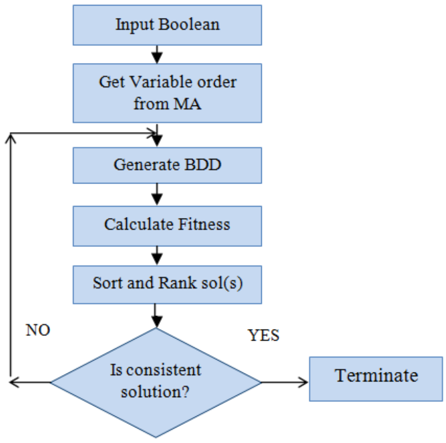
III. BDD MINIMIZATION BASED ON MEMETIC ALGORITHM

Memetic algorithms (MA) are inspired by Dawkins' notion of a meme [7], [8]. They are population based optimization algorithms in which the solutions are subjected to processes of competition and mutual cooperation in a way that resembles the behavioral patterns of the living beings from the same species.

They might be implemented as local search heuristics techniques, approximation algorithms or, sometimes, even (partial) exact methods. The hybridization is meant to either accelerate the discovery of good solutions, for which evolution alone would take too long to discover, or to reach solutions that would otherwise be unreachable by evolution or a local method alone.

MA is similar to GA, but the elements forming a chromosome are memes, and not genes. Meme is the basic unit of cultural or imitation passed on by non-genetic operators. The unique aspect of MA is that all chromosomes in a generation are allowed to gain some experience, through a local search [9], before being involved in the Evolutionary process. On a randomly created initial population, a local search is performed on each chromosome in order to improve its experience and thus obtain a population of local optimum solutions. The process then undergoes a basic genetic operation to produce new generation, which is again subjected to local search. Various approaches have been proposed [9] to perform the local search in Memetic algorithm. The performance of each chromosome is evaluated again after performing local search. The change is retained if the performance improves else it is reversed [10]. Main parameters influencing the performance of MA are population size, number of generations, crossover rate, and mutation rate along with the local search mechanisms. 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 types of the crossover operator have been used for BDD minimization: Order Crossover, Partially Mapped Crossover and Cycle Crossover. The performance of ordered crossover operator has been compared.

A flowchart for MA based approach used for BDD Variable Ordering is shown in figure below.



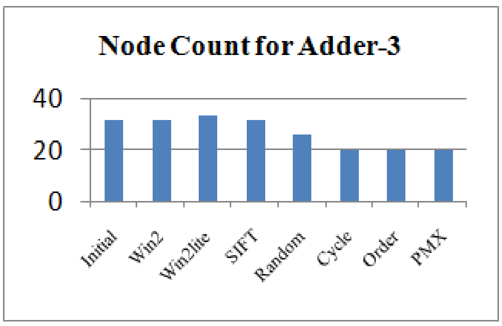
IV. SIMULATION RESULTS

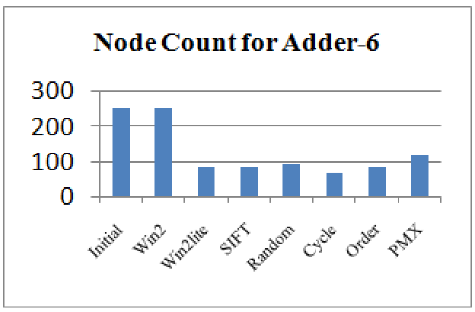
The proposed approach using ordered, cyclic and partially mapped crossover operators are being implemented with C++ on Ubuntu 11.04. It gives minimal nodes and reduced execution time for every function when compared with all other approaches. The simulation results for multi-input Adder circuits have been displayed in Table 1.

TABLE 1: Comparison of MA results

Multi-input Adder Bench mark Circuits	I/O	Initial node count	WIN2	WIN2ite	WIN 3	SIFT	RAND OM	Proposed MA with cycle crossover		% node reduction w.r.t. initial node count	Proposed MA with order crossover			Proposed MA 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	0.08	0	8	0.13	0	8	0.14	0
2-adder	5/3	17	17	17	17	17	17	14	0.11	17.6	14	0.17	17.6	14	0.13	17.6
3-adder	7/4	32	32	34	32	26	34	20	0.35	41.17	20	0.17	41.17	20	0.36	41.17
4-adder	9/5	63	65	63	46	46	61	30	7.91	52.3	38	7.09	39.6	32	6.76	49.2
5-adder	11/6	126	128	126	61	55	78	32	44.5	74.6	32	27.7	74.6	32	43.9	74.6
6-adder	13/7	253	255	253	85	85	93	68	23.83	73.12	88	175.6	65.2	118	186.5	53.35

Fig2. Comparison of node count for different multi-input adder circuits using different algorithms.





ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526
INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 3, March 2014

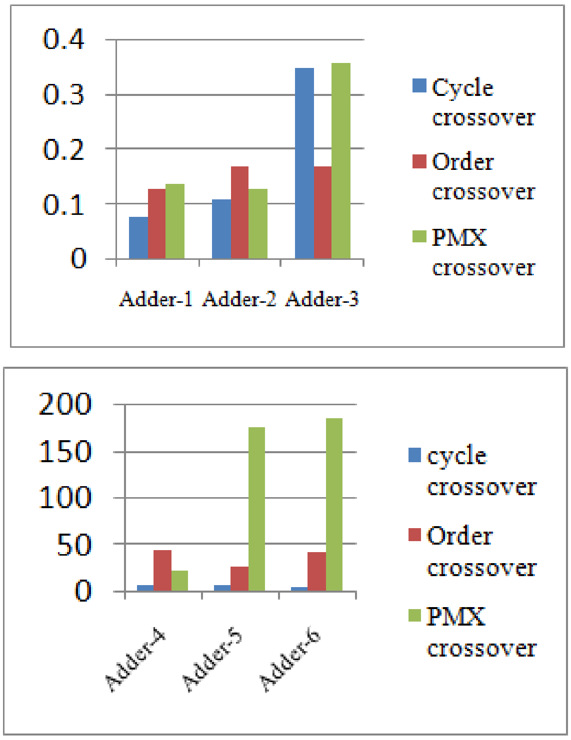


Fig3. Comparison of computation time for various benchmark circuits using different Crossover Operators in MA.

V. CONCLUSIONS

The result of Memetic algorithm using the various crossover operators on the n-bit adder circuits are presented and these results demonstrate that MA algorithm implemented using cycle crossover operator results in less computation time and reduced node count.

REFERENCES

[1] W.Gunther and R.Drechsler, "Efficient Minimization and manipulation of linearly transformed Binary Decision Diagrams", IEEE Transactions on Computers, pp. 1196-1209, 2003.
[2] Z. Michalewicz, "Genetic Algorithm + Data Structure = Evolution Programs", Springer-Verlag, 1994.
[3] A. Vargas-Martinez and L. E. Garza-Castanon, "Combining artificial intelligence and advanced techniques in fault-tolerant control", Journal of Applied Research and Technology, vol. 9, no. 2, pp. 202–226, 2011.
[4] W. E. Hart, N. Krasnogor and J. E. Smith, "Recent Advances in Memetic Algorithms", Springer-Verlag, 2005.
[5] M. Maruniak and P. Pistek, "Binary Decision Diagram Optimization Method Based on Multiplexer Reduction Method", IEEE International Conference on System Science and Engineering, pp.395-399, 2013.
[6] R. Ebendt, G.Fey, R. Drechsler, "Advanced BDD Optimization", Springer 2005, 222p, ISBN 978-0-387-25453-1.
[7] Wikipedia search, "Memetic Algorithm", http://en.wikipedia.org/wiki/Memetic_algorithm
[8] M. Rojas and C. Coella, "A Memetic algorithm with simplex crossover for solving Constrained Optimization Problems", World Automation Congress 2012, pp.1-6
[9] P. Merz and B. Freisleben, "A Genetic local search approach to the quadratic assignment problem", Proc. Of seventh International conference Genetic Algorithms,1997, pp.465-472
[10] S. Rehan and M. Bansal, "Performance Comparison among different Evolutionary Algorithms in terms of Node Count Reduction in BDDs", International Journal of VLSI and Embedded Systems, 2013, pp.491-496.
[11] S.Chaudhary and A. Dutta, "Algorithmic Optimization of BDDs and Performance Evaluation for Multi-level Logic Circuits with Area and Power Trade-offs", Circuits and Systems, 2011, pp. 217-224.
[12] N. Frištacký, M. Kolesár, J. Kolenička, "Optimization of multiplexer trees using modified truth table," International Conference on Applied Electronics (AE), Pilsen: IEEE, 2010, pp. 265-268.
[13] A. Shreshtha , "Decision Diagram Based Methods and Complexity Analysis for Multi-State Systems", IEEE transactions on reliability 2010, pp. 145-161

BIOGRAPHY

Anamika received the B.Tech in Electronics & Communication from LIT, Phagwara. Pursuing M.Tech in VLSI Design from Thapar University, Patiala. Her research interests include FPGA implementation using HDLs (VHDL, Verilog), programming using Scripting languages (Perl, TCL/TK).