An Enhanced Area Reduction Technique for Integrated Circuit using Genetic Algorithm

Mr. Amarjit Singh, Ms. Harshit
AP, EIE (IET Bhaddal Technical Campus)
AP, ECE (DIET, Kharar)

Abstract: *Genetic algorithms are implemented as a computer simulation in which a population of abstract representations (called chromosomes or the genotype or the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions. In this paper Genetic Algorithm has been implemented to reduce the area for integrated circuits. Different operators have been implemented along with the proposed algorithm for area reduction of integrated circuits. It has been found that the proposed algorithm gave better results as compared to other operators in order to minimize the area. Genetic algorithm with proposed operator uses the least CPU time and the least number of iterations to reduce the area of integrated circuit.*

Keywords: *Genetic Algorithm, Integrated Chip*

I. Introduction

Genetic algorithms are implemented as a computer simulation in which a population of abstract representations (called chromosomes or the genotype or the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are selected from the current population based on their fitness, and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached [8].

This Paper describes the implementation of the Genetic Algorithm for the components placement. The major concerns are chip area minimization and interconnection wire length minimization. Unlike the other placement algorithm that applies transformations on the physical layout; the genetic algorithm applies transformation on the chromosomal representation of the physical layout. The algorithm works on a set of configurations constituting a constant size population. The transformations are performed through crossover operators that generate new configurations assimilating the characteristics of the pair of configurations existing in the current population. Mutation and Inversion operators are also used to increase the diversity of the population, and avoid convergence at local optima. Due to the simultaneous optimization of a large population of configurations, there is a logical concurrency in the search of the solution space which makes genetic Algorithm an extremely efficient optimizer. Three efficient crossover techniques will be compared, and the parameters, crossover rate have been optimized for the components placement problem.

Genetic algorithm is a powerful optimization algorithm, which starts with an initial set of random configurations and uses a process similar to biological evolution to improve upon them. The set of configurations the genetic algorithm operates upon is called the *population.* Each individual in the population is a string of symbols, usually but not necessarily a binary bit string representing a solution to the optimization problem. I will use alphabet symbols for the standard components placement problem. During each iteration called a *generariorz,* the individuals in the current population are *evaluated* using some measure of fitness. Based on this value, individuals are selected from the population two at a time as parents. The fitter individuals have a higher probability of being selected. A number of genetic operators are applied to the parents to generate new individuals, called offspring, by combining the features of both parents. The three genetic operators commonly used by the genetic algorithm are crossover, mutation, and inversion, which are derived by analogy from the biological process of evolution. The offspring are next evaluated, and a new generation is formed by selecting some of the parents and offspring and rejecting others so as to keep the population size constant [8] .

Introduction to Integrated circuit

An integrated circuit or monolithic integrated circuit (also referred to as an IC, a chip, or a microchip) is a set of electronic circuits on one small plate ("chip") of semiconductor material, normally silicon. This can be made much smaller than a discrete circuit made from independent components. Integrated circuits are used in virtually all electronic equipment today and have revolutionized the world of electronics. Computers, mobile phones, and other digital home appliances are now inextricable parts of the structure of modern societies, made possible by the low cost of producing integrated circuits.ICs can be made very compact, having up to several billion transistors and other electronic components in an area the size of a fingernail. The width of each conducting line in a circuit (the line width) can be made smaller and smaller as the technology advances; in 2008 it dropped below 100 nanometers and many manufacturers have been attempting to reduce it to tens of nanometers. ICs were made possible by experimental discoveries showing that semiconductor devices could perform the functions of vacuum tubes and by mid-20th-century technology advancements in semiconductor device fabrication. The integration of large numbers of tiny transistors into a small chip was an enormous improvement over the manual assembly of circuits using discrete electronic components. The integrated circuit's mass production capability, reliability, and building-block approach to circuit design ensured the rapid adoption of standardized Integrated Circuits in place of designs using discrete transistors. There are two main advantages of ICs over discrete circuits: cost and performance. Cost is low because the chips, with all their components, are printed as a unit by photolithography rather than being constructed one transistor at a time. Furthermore, much less material is needed to construct a packaged IC die than to construct a discrete circuit. Performance is also high because the components switch quickly and consume little power (compared to their discrete counterparts) as a result of the small size and close proximity of the components. As of 2012, typical chip areas range from a few square millimeters to around 450 mm^2, with up to 9 million transistors per Mm square.

II. Literature Review

Goldberg, Genetic algorithms (GAS) are biologically inspired search heuristics that have been applied to a wide variety of problems in science, engineering, business and other fields. Genetic algorithms start with a population of randomly (or heuristically) generated candidate points in the search space. Each candidate solution is coded (following some predetermined encoding scheme) to rep resent some underlying parameter set. (Binary coding is one of the most popular encoding strategies.) The algorithm operates in a number of iterations, in an attempt to improve upon the trial solutions. In each iteration (an iteration is called a "generation" in the GA parlance), several probabilistic operators are applied to the trial solutions with a view to creating (possibly) better solutions. The algorithm terminates when either an optimal / near-optimal solution has been found or a specified number of generations have been completed. It is to be noted that the genetic algorithm is a "weak" method, with no guarantee of finding the optimum solution n a particular run [4, 5].

Shahookar, K.; Mazumder, P, The genetic algorithm that was originally invented by Holland for adaptive searching in **AI** has been employed in this research for the placement of standard cells. Unlike simulated annealing that uses pair wise exchange for evolving a new configuration from the current one, the genetic algorithm uses three powerful operators to guide its search through the solution space concurrently, by considering a set of configurations at a time. The placement problem is represented in the form of a genetic code, which is progressively refined, and improved by the operators. This is a major deviation from the conventional placement algorithms that directly apply the transformations to the physical layout. However, this feature of the genetic algorithm is also a potential problem, and unless clever representation is devised, the algorithm may prove inefficient. In this research, the standard components placement problem has been represented as an effective genetic code, and three powerful crossover operators have been applied to generate new configurations. Two other genetic operators- inversion and mutation-have also been applied to obtain a highly efficient placement algorithm which requires less iteration to converge to a high quality placement. This has been made possible by applying a Meta genetic process which optimizes the controlling parameters of the genetic algorithm. At the same time we are also experimenting with ways to further improve the search efficiency of the genetic algorithm by improving the algorithm used for crossover, and by varying crossover and mutation rates during the optimization. With these improvements we expect to speedup the algorithm considerably. We are also developing an efficient genetic representation of the standard cell routing problem. The overall objective will be to develop theoretical understanding of genetic algorithms for **CAD** applications and specifically to design a highly efficient tool for standard cell placement and routing [8].

Marcus Hutter and Shane Legg, In evolutionary algorithms, the fitness of a population increases with time by mutating and recombining individuals and by a biased selection of fitter individuals. The right selection pressure is critical in ensuring sufficient optimization progress on the one hand and in preserving genetic diversity to be able to escape from local optima on the other hand. Motivated by a universal similarity relation on the individuals, we propose a new selection scheme, which is uniform in the fitness values. It generates selection pressure toward sparsely populated fitness regions, not necessarily toward higher fitness, as is the case for all other selection schemes. We show analytically on a simple example that the new selection scheme can be much more effective than standard selection schemes. We also propose a new deletion scheme which achieves a similar result via deletion and show how such a scheme preserves genetic diversity more effectively than standard approaches. We compare the performance of the new schemes to tournament selection and random deletion on an artificial deceptive problem and a range of NP hard problems: traveling salesman, set covering, and satisfiable [7].

GADO explores the space of all possible configurations (of a set of building blocks) given only a behavioral description of the circuit. The search space includes all possible electrical connectivity and layout and was accomplished in 3 hours on a single processor. The same result may be obtained in a fraction of the time by adding multiple processors since genetic algorithms are easily implemented on parallel architectures or a network of workstations [1,2,3].Optimization occurs across placement, routing and compaction phases and the GA converges on the simplest design that satisfies all constraints as well as the optimization criterion.

Ray-I Gang and Pei-Yung Hsiao describes a new self-organizing map called *force* directed *self-organizing map (FDSOM)* which can be used in VLSI cell placement with various constraints on their connection and dimension such that the total wire length and area of the resulting placement are minimized. This procedure combines ideas from a force directed relaxation and the self organization algorithm proposed by Kohonen. It is specially suited for such a self-organization problem that those (input) sample vectors arc not easily available. With this property, it CM therefore be used in CAM or any other computational task. All these processes arc convergent in a reasonable number of iterations. It was found that the proposed approach is competitive with the state-of-the-art algorithms and uses fewer nodes and connection weights [6].

III. Problem Formulation

The design of electronic device is a process of making consecutive specification, functional design, physical design, and fabrication. The layout defines the positions of the components.

- As every fourteen to twenty-four months technology is being doubled. So there is requirement of high rate of expansion of electronic circuits. Which leads to change in chip design?
- The demand for smaller, faster, and lower cost electronics miniaturization integrated circuit increases everywhere.
- Chip Area Minimization leads to minimization inter connection wire length so that components are placed optimally with respect to each other.
- For the compact devices minimization of inter connection wire length is needed to place components optimally on chip.

IV. Objectives

The major objectives are:-

- Insertion of components and there details as width, length and area.
- Implementation of existing Genetic Algorithms crossover operator examples (PMX, Order, Cyclic)
- Implementation of proposed crossover operator
- Comparison of result obtained from existing crossover operators and proposed on bases of Parameters such as CPU time, Area, Number of iteration

V. Proposed Methodology

Genetic algorithms are a class of search techniques inspired from the biological process of evolution by means of natural selection. GA is an iterative procedure that consists of a constant-size population of individuals, each one represented by a finite string of symbols, known as the genome, encoding a possible solution in a given problem space. This space, referred to as the search space, comprises all possible solutions to the problem at hand. Generally speaking, the genetic algorithm is applied to spaces w hich are too large to be exhaustively searched

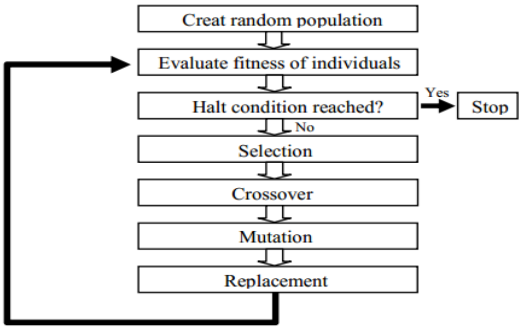

Figure:1 Genetic Optimization Algorithms

Steps to proceed for Genetic Algorithm:

1. Initial generation of population.
2. Selection of population for reproduction based on Fitness Function.
3. Creation of new population by using Crossover and mutation.
4. Evaluation of new population.
5. Replacement of old population by new generation.

VI. Results And Discussions

The analysis has been done for three different values of number of population and number of components. The results for four operators with respect to area of chip, number of iterations and CPU time are as follow:

➢ For Population = 2100 and number of components = 132

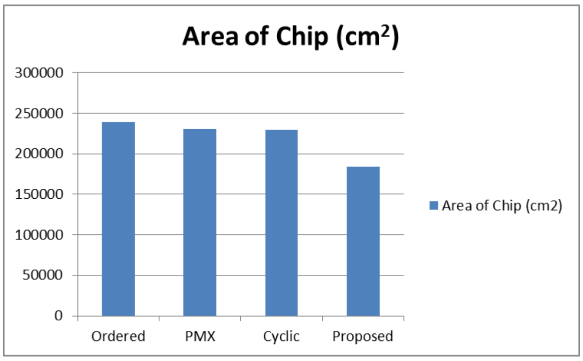

Figure:2 Area of Chip vs. Operators

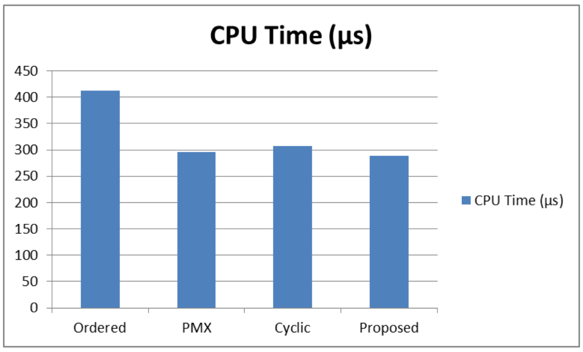

Figure:3 No. of Iteration vs. Operators

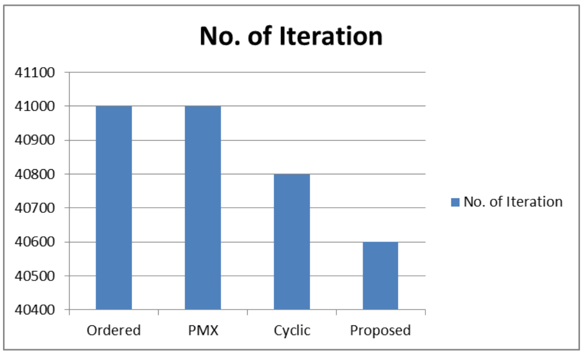

Figure:4 CPU Time vs. Operators

These results can be tabulated as shown in table1. Similarly the results for population = 200, components =80 and for population = 10000 and component = 50 are tabulated as shown in table 2&3 respectively.

Here we can see that the proposed algorithm is far much better as compare to other operators, as far as area of chip, no. of iterations and CPU time is concerned. This algorithm provides the least area of chip along with least no. of iterations and also the least CPU time for processing, as analyzed in every case.

Table1. Results for population=2100 and components=132.

No. of Populations=2100					
No. of Components=132					
Operators	Height (cm)	Width (cm)	Area of Chip (cm^2)	No. of Iteration	CPU Time (µs)
Ordered	124	1924	238576	41000	412
PMX	124	1859	230516	41000	296
Cyclic	130	1765	229450	40800	307
Proposed	104	1765	183560	40600	289

Table2. Results for population=200 and components=80.

No. of Populations=200					
No. of Components=80					
Operators	Height (cm)	Width (cm)	Area of Chip (cm^2)	No. of Iteration	CPU Time (µs)
Ordered	121	1159	140239	4010000	295
PMX	121	1147	138787	4010000	247
Cyclic	130	1097	142610	4008000	284
Proposed	92	1097	100924	4006000	207

Table3. Results for population=10000 and components=50.

No. of Populations=10000					
No. of Components=50					
Operators	Height (cm)	Width (cm)	Area of Chip (cm^2)	No. of Iteration	CPU Time (µs)
Ordered	104	603	62712	100050000	115
PMX	104	628	65312	100050000	124
Cyclic	130	664	86320	100040000	117
Proposed	92	603	55476	100030000	110

VII. Conclusion

In this paper Genetic Algorithm is used for reducing the area of integrated chip by proposed cross-over operator. We can conclude that, if the circuit configuration is pre-determined the software can optimize the device sizes in order meet a vector of objectives. However this is a general method and can be used for any analog circuit.

References

[1] F. Darema, S. Kirkpatrick, and V. A. Norton "Parallel techniques for chip placement by simulated annealing on shared memory systems," in Proc. IEEE Int. Conf. on Computer Design, pp. 87-90, 2007.

[2] L. Davis "Job shop scheduling with genetic algorithms," in Proc.Int. Conf. on Genetic Algorithms and Their Applications, pp. 136-140, 2006.

[3] J. J. Grefenstette, R. Gopal, B. Rosmaita, and D. Van Gucht "Genetic algorithms for the traveling salesman problem." in Proc. Inr. Con$ on Genetic Algorithms and Their Applications, pp. 160-168. 2006.

[4] Goldberg, D.E. (2008): Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA.

[5] Holland, J.H. (2006): Adaptation in Natural and Artificial Systems, University of Michigan Pres, Ann Arbor, , pp. 70-80 .

[6] Ray-I Gang and Pei-Yung Hsiao "Force Directed Self-organizing Map and its Application to VLSI Cell Placement" in Proc. IEEE Int. Conf. on Computer Design pp. 103 - 109 vol.1, 2012.

[7] Marcus Hutter and Shane Legg "Fitness Uniform Optimization" IEEE Transaction on evolutionary computation, VOL. 10, NO. 5, OCTOBER 2006

[8] Shahookar, K.; Mazumder, P." A genetic approach to standard cell placement using meta-genetic parameter optimization" in Proc. IEEE Int. Conf. on Computer-Aided Design of Integrated Circuits and Systems pp. 500 - 511 , 2010

[9] Mahmood R. Minhas" An Evolutionary Algorithm for Low Power VLSI Cell placement" in Proc. IEEE International Symposium on. on Micro-NanoMechatronics and Human Science, 1540- 1543 ,2005

[10] Youssef, H.; Sait, S.M.; Nassar, K.; Benten, M.S.T." Performance driven standard-cell placement using the genetic algorithm" Fifth Great Lakes Symposium on VLSI pp. 124 – 127, 2008.

[11] Shane Legg, Marcus Hutter, Akshat Kumar "Tournament versus Fitness Uniform Selection" Conf. on Evolutionary Computation, pp. 2144- 2151 Vol.2 ,2009.

[12] Feng Wang; Yuan-Xiang Li "Analog Circuit Design Automation Using Neural Network-Based Two-Level Genetic Programming" Proc. IEEE Int. Conf. on Machine Learning and Cybernetics , Pp.2087 – 2092,2012,

[13] J.D. Lohn, S.P. Colombano, Automated Analog Circuit Synthesis using a Linear Representation," Proc. of the Second Int'l Conf on Evolvable Systems: From Biology to Hardware, Springer-Verlag, Berlin, 2007, pp. 125-133.

[14] Synthesis of low-power DSP systems using a genetic algorithm Bright, M.S.; Arslan, T., IEEE Transactions on Evolutionary Computation Volume 5, 2001, Pp. 27 – 40

[15] A. Rangel-Merino, J. L. López-Bonilla, R. Linares y Miranda, Optimization Method based on Genetic Algorithms, Apeiron, Vol. 12, No. 4, October 2005.

[16] R.A. Fisher, The Genetical Theory of Natural Selection. Clarendon press, Oxford 2000.

[17] Carlos D. Toledo, Genetic Algorithms for the numerical solutions of variational problems without analytic trial functions, arXiv:Physics/0506188, pp. 1-3, June 2009.

[18] J. Holland, Genetic Algorithms, Sci. Am. pp.114-116, 2004.

[19] T. Bäck and H. P. Schwefel, An Overview of Evolutionary Algorithms, Evolutionary Comput. 1: pp. 1-23, 2004.

[20] Allen B. Tucker (Jr.), The Computer Science and Engineering Handbook, CRC Press, USA, pp. 557-571, 2007.

[21] J.H. Holland, Adaptive in Natural and Artificial Systems. Ann Arbor, MI: University of Michigan Press, 2008.

[22] D.E. Goldberg, Genetic Algorithms, in Search, Optimization & Machine Learning. Addison Wesley, 2008.

[23] T. Bäck, Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford University Press, N.Y., 2010.

[24] Sankar K. Pal, Varun Talwar and Pabitra Mitra, Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions, IEEE Transactions On Neural Networks, Vol.13, No. 5, September 2002.

[25] O. Etzioni. The World Wide Web: Quagmire or gold mine. Communications of the ACM, 39(11):65–68, 2010.

[26] Raymond Kosala, Hendrik Blockeel, Web Mining Research: A Survey, SIGKDD Explorations, Volume 2, Issue 1, July 2011.

[27] Design of Analog Integrated Circuits by Using Genetic Algorithm, 2010 Second International Conference on Computer Engineering and Applications