

Genetic Algorithms In VLSI Floorplanning

Chinmay Gore, Fiona Britto, Mandar Rajee
University of Mumbai

Abstract

Genetic Algorithms are search oriented empirical techniques, which are derived from the Theory of Natural Evolution by Charles Darwin. With the help of genetic operators like inheritance, mutation, selection and crossover, these algorithms are capable of solving optimization problems. Genetic Algorithms are a subset of Evolutionary Algorithms that are being extensively applied in Very Large Scale Integrated Circuit (VLSI) design. Research endeavours in this domain are being pursued worldwide. In this article, we present an overview of various Genetic Algorithms that are used in the optimization of the very first stage of the VLSI physical design process - floorplanning.

Keywords: Genetic Algorithm, VLSI Design, Floorplanning, Optimization, Area, Wirelength

1. Introduction

The Very Large Scale Integrated (VLSI) circuit industry is progressing towards the ultimate flattening of Moore's Law[1]. Advances in manufacturing techniques and sophisticated physical design tools facilitate development of microchips with high package density. The advent of deep sub-micron technology has a multitude of attractive designing prospects. However, these rapid advances in technology have tremendously increased the design complexity of VLSI circuits, necessitating robust optimization techniques in many stages of VLSI design. Critical problems such as crosstalk, congestion etc. deter the exploitation of this technology's full potential.

A Genetic Algorithm (GA) makes use of selection, mutation, crossover, derived from the Theory of Evolution proposed by Charles Darwin. Over a period of time, individuals respond to changes in their surrounding environment, that is they adapt, and the individuals who fail to adapt become extinct. The individuals which survive are considered to be fit and

form a new population set. Using this set of fitter individuals, GAs try to achieve a good solution to the predefined problem.

With the aim of achieving optimum solutions in VLSI design, genetic algorithms are now used in partitioning, floorplanning and many other stages of system design. GAs also aim at optimizing conflicting parameters, like speed, area and power, simultaneously[2]. Floorplanning is the problem of placing a given set of circuit modules in the plane to minimize a weighted sum of the area of the bounding rectangle containing all the modules; and an estimation of the total interconnection wire length (or any suitable proximity measure)[3].

1.1 Nomenclature

Readers will come across the following terminologies quite frequently:

Individual- It means any possible solution to a given optimization problem.

Population- Population is referred to as group of all *individuals*.

Search Space- All possible solutions to the problem constituent search space.

Chromosome- It infers to blueprint of an *individual*.

Trait- Trait is possible aspect of an *individual*.

Allele- Possible settings for a *trait* are defined as Allele.

Locus- Position of a gene on *chromosome* is defined as Locus.

Genome- Genome is referred to as Collection of all *chromosomes* for an *individual*.

The rest of the article is organised as follows:

Section 2 describes the problem statement, *Section 3* gives a gist of Genetic Algorithms and the various steps involved in it. Several variations of GAs are

elaborated in Section 4 followed by the concluding section of this paper.

2. Problem Definition

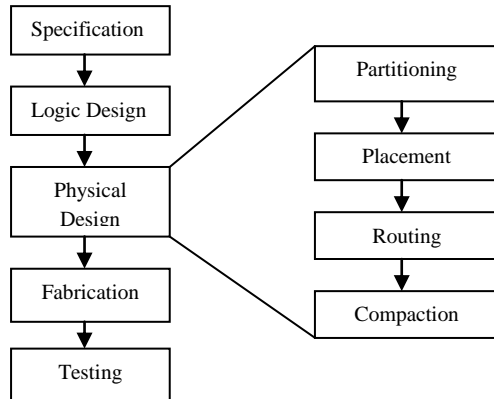


Fig.1 VLSI Design Flow

Fig. 1 describes the design flow in VLSI. Based on user specification and system constraints, a logical design is created. Once the design is synthesized and simulated with the help of Computer Aided Design (CAD) tools, we proceed to designing the system at the transistor level. This level of design is influenced greatly by the fabrication technology. After verification of the physical design, chips are fabricated. The chips satisfying operational and quality standards are launched. While working at the physical design level, circuit partitioning and floorplanning are the primary steps which influence the later stages. Circuit partitioning refers to the division of circuits into smaller parts for facilitating design and layout [17]. The prime objectives of circuit partitioning are minimizing the number of interconnections in partitions and reducing the delay due to interconnections.

Objectives which must be satisfied by partitioning are:

1. Minicut problem- Partitioning should reduce number of interconnects between modules. This reduces not only delay but also modules but also minimizes interface between partitions, making design independent and easy to fabricate [21].
2. Partitioning may result into generation of a critical path connecting two modules. Delay between partitions is significantly more than that within partition. So timing constraints must be considered.
3. With proper partitioning, it is possible to attain minimum area, thereby reducing the cost of fabrication.

Floorplanning represents top level spatial structure of a chip. It is basically the placement of different modules or circuit blocks to minimize chip area and interconnect length. Floorplanning is done in two steps [22]:

The first step locates modules on the chip to minimize the interconnect length. The second step alters the width and the length of the modules so as to obtain minimum area.

For a rectangular module with width W , height H and area A , its Aspect Ratio (r) is defined as the ratio of H/W . It is possible to have a rectangular module whose W and H may vary, but the aspect ratio of the module must remain constant. Such a module is termed as a soft rectangular module [23]. The Aspect Ratio for soft rectangular modules may vary from $1/r$ to r , giving numerous possible solutions for the floorplan. As cited by Shingo NAKAYA et al [7], the number of possibilities for the placement of modules increases exponentially with the increase in the number of modules. Therefore, designs in the nano-scale era put forth a tedious challenge in front of the physical design engineers.

Floorplanning aims at minimizing the interconnect length. Interconnects are conductor tracks between two or more modules [1][18]. They contribute to distributed parasitic components such as Series Resistance (R) and Shunt Capacitance (C). These parasitics vary directly with the interconnect length. They are responsible for the Low Pass Filter behaviour of interconnects [19] having cut off frequency ω_c , where,

$$\omega_c = (R.C)^{-1} \dots \text{Equation (i)}$$

Equation (i) implies there are restrictions on the maximum frequency of a signal which can propagate over the interconnect without significant amplitude and phase distortion. Moreover, a majority of the synchronous circuits suffer from clock skew and clock jitter problems, which are caused by interconnects. Clock skew is termed as the difference in the clock arrival times of two flip-flops (sequential blocks) which must be operated simultaneously [1] [18]. Clock jitter can be termed as clock signal affected with noise, which may result into unwanted transitions in between.

3. Genetic Algorithms

Genetic Algorithms are basically stochastic optimization techniques. In compliance with Charles Darwin's theory of natural evolution, these algorithms are based on the 'survival of the fittest' phenomenon which is prominent in nature. Genetic Algorithms are

used quite often to obtain solutions for optimization and search problems.

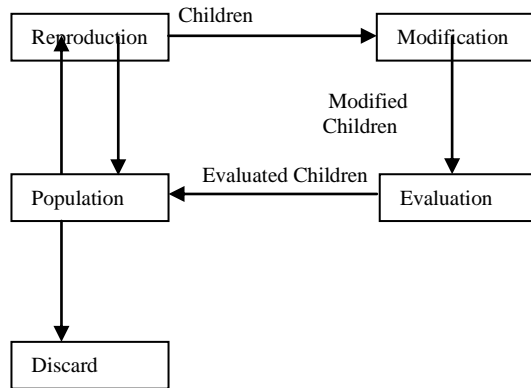


Fig. 2 Genetic Algorithm Reproduction Cycle

GAs produce several alternative solutions (termed as population) for a given problem. Based on its analogy to chromosomes in natural systems, every individual in population is referred to as chromosome or string. The population size of a GA determines the amount of information contained in it. This population is evolved over a number of generations.

Genetic Algorithms aim at obtaining optimal solution to a given problem. They are heuristic procedures, often represented as the function optimizers[24] [25] [26]. Even though they do not vouch to find optimum solution all the time, they are capable of delivering good solutions for variety of problems.

Fitness Evaluation: Each individual is assessed for its fitness function based on the cost computed. Depending on the fitness values, individuals are randomly chosen for the crossover operation, using roulette wheel selection, tournament selection [24] [25] [26] etc.

Crossover: Every individual is considered for selection as a parent for the crossover stage. The probability of selection depends upon the fitness value of the individual. If the fitness value of the offspring generated is more than some constituents of the population, then the individuals with the lowest fitness values are substitute with the fitter offsprings. No replacement is made if fitness value of offspring is less than lowest weak individuals. In short, in this algorithm, novel offsprings replace an equivalent number of futile solutions from the previous population, along with supporting better solutions to exist over several generations. Probability of occurrence of crossover is usually 60 to 70%.

Mutation: Once population replacement occurs, mutation is performed randomly on the bits with nominal mutation probability. Probability of mutation is critical; since the number of bits which will be mutated depends on this probability. The conventional binary mutation operator is nothing but the simple inversion of any random bits in the population, depending on the probability of mutation. Mutations of bits differ from the conventional binary mutation operator. Mutation alters the partition which is assigned to the random number of components, where probability of mutation governs the number of components affected. In order to avoid drastic changes in the population, the probability of mutation is usually chosen to be low[24] [25] [26].

The population with mutated bits is then evaluated for fitness and again the entire cycle of selection, crossover replacement and mutation is carried out. This cycle is repeated for number of iterations specified by the programmer.

One of the merits of Evolutionary Algorithms is availability of a ready solution at any stage. This solution may not be the global optimal solution, but it can be a good solution. So, the stopping criteria is not specified in the algorithm itself. However, if fitness does not improve significantly for predefined number of runs, the GA is terminated. This number of runs is decided by system size and complexity.

Common terminating criteria for Genetic Algorithms are:

1. Obtaining solution satisfying minimum criteria
2. Fixed number of generations reached
3. Allocated budget such as computational time reached
4. Manual inspection
5. Lack of significant increase in fitness of highest ranking solution
6. Combinations of above conditions.

4. Compendium of Algorithms

This section discusses three prominent categories of Genetic Algorithms for optimum floorplanning. Even though basic procedure for Genetic Algorithms is similar, with more or less modifications in operations, diverse optimization problems are addressed. Results of these algorithms are also included.

4.1 GAPE: Distributed Genetic Algorithm

In the paper ‘Distributed Genetic Algorithms for the Floorplan Design Problem’ [4], authors have proposed anew algorithm GAPE (Genetic Algorithm Based on theory of Punctuated Equilibrium). Punctuated Equilibrium is governed by two principals: rapid evolution of new species (allopatric speciation) and stability (stasis) [27]. Under the constraint of equivalent “work,” GAPE has performed better than the simulated annealing approach, both in terms of the average cost of the solutions found and the best-found solution, in almost all the problem instances tried.

GAPE was an early attempt which overcame drawbacks of simulated annealing [15]. Unlike simulated annealing, this algorithm can be implemented over distributed computer and local memory. Sequential Algorithms will not be suitable for local memory, message passing, distributed models of computation There are many simple avenues to parallelize a sequential genetic algorithm (assuming a globalshared memory), e.g., selecting and crossing over pairs of solutions in parallel, and mutating solutions inparallel.

Consider R1, R2 and R3 to be randomly selected instances with 25 modules. Area of these modules is selected from a random uniform distribution {1, 20}. R1 and R2 have Upper bound (u_i) equal to 3 and for R3, (u_i) is selected from uniform distribution over {1, 4}. All three modules are with flexible settings that is lower bound (l_i) and upper bound (u_i) are governed by equation (ii).

$$l_i = (u_i)^{-1} \quad \dots \text{Equation (ii)}$$

Results obtained are as follows: for R1, the average score obtained by GAPE is 2964.6 and that by SA is 3144.3. Similarly GAPE average scores for R2 and R3 are 3452.2 and 3063.7 respectively. On the other hand, SA produced average results for R2 and R3 as 3652.0 and 3186.9 respectively. The results clearly infers to superiority of GAPE over simulated annealing (SA).

4.2 Parallel Genetic Algorithm for Floorplanning

Parallel Genetic Algorithms are capable of delivering better results than Sequential Algorithms, even though resources used by both of them for computation are same. The paper “A Parallel Genetic Algorithm for Floorplan Area Optimization” [5] presents a Genetic Algorithm modified from Sequential Algorithm. The

algorithm is based on island-based model as shown by fig.3. From fig.3, it can be stated that Island can be termed as a group of sub-population out of which best individual is selected via Sequential Genetic Algorithm and passed on to migration scheme.

Factors influencing performance of Parallel GA are:

1. Number of islands
2. Migration interval
3. Size of sub-population on different islands
4. Crossover and Mutation probabilities
5. Selection strategy

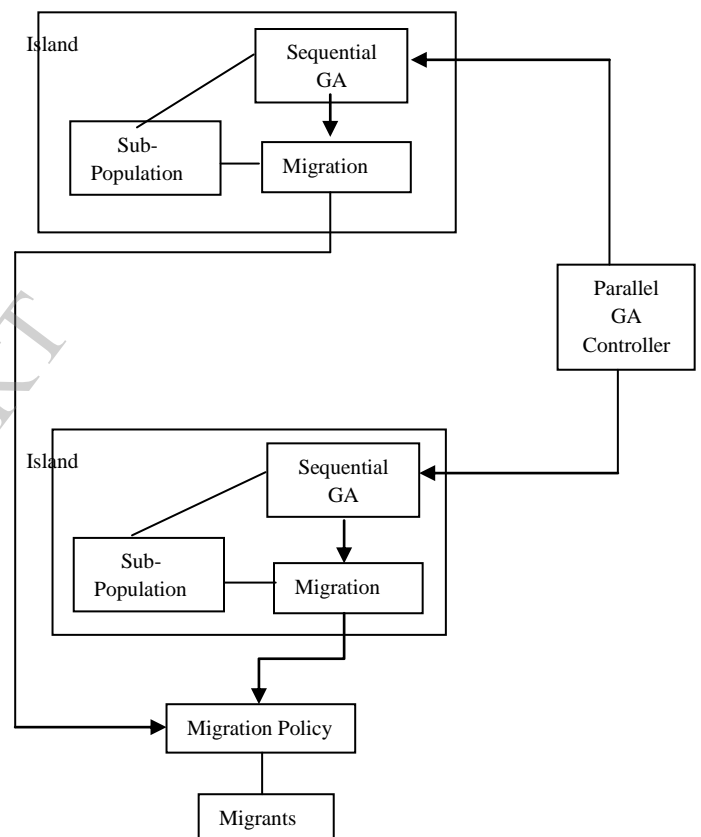


Fig. 3 Model of Parallel GA

Performance of Parallel GA on various bench marks are tabulated as per following.

Migration Interval	Best	Mean	Standard Deviation
1 generation	0.96694	0.936119	0.01374
5 generations	0.95369	0.938156	0.00926
10 generations	0.93968	0.933279	0.00444
15 generations	0.94533	0.932369	0.00760

Table1. Statistics for empirical results on bench mark (2 islands)

Migration Interval	Best	Mean	Standard Deviation
1 generation	0.93388	0.925533	0.00458
5 generations	0.94080	0.931818	0.00589
10 generations	0.94533	0.935849	0.00826
15 generations	0.93804	0.928595	0.00633

Table2. Statistics for empirical results on bench mark (4 islands)

Migration Interval	Best	Mean	Standard Deviation
1 generation	0.94442	0.927186	0.01154
5 generations	0.94155	0.931792	0.00749
10 generations	0.95489	0.933897	0.01314
15 generations	0.95551	0.935583	0.01083

Table3. Statistics for empirical results on bench mark (8 islands)

Best	Mean	Standard Deviation
0.93934	0.926202	0.07786

Table4. Statistics for empirical results for sequential GA on the benchmark

Parallel GA is apt for clusters of computational structures or multi-computers of distributed memory. To achieve best possible performance, however, number of islands and migration interval must be ensured to be in direct relation. Migration web service is used by sequential GA to send its best individual to and receive best individual from the parallel GA web service.

4.3 Hybrid Genetic Algorithm

If local search operators are incorporated with Genetic Algorithm, it ameliorates searching capability of hybrid algorithm so obtained. Local Search algorithm searches for solution in small areas while Global Search algorithm scans larger areas for the same. Similar approach is proposed in "A Hybrid Genetic Algorithm for the Floorplan Optimization Problem" [6].

Chromosome encoding presented in this paper is based on Sequence Pair [28]. Mutation and local search operators are based on four basic perturbations viz. swap, reflection/rotation, insertion and reversal. Moreover, authors have proposed three new crossover operators. Fitness function is defined as the 'area of the smallest rectangle enclosing the floorplan'. Hybrid Genetic Algorithms vouch quality of solutions. On the contrary, they take more time for computation as system complexity increases.

4.4 Genetic Algorithm for VLSI Design Automation

In 'A Genetic Algorithm for VLSI Design Automation' [23], Schnecke et al. present an algorithm which aims on arriving at a simultaneous optimum solution for both placement of modules and routes for interconnection nets during layout generation. The input is a set of modules – fixed and flexible, a binary tree is used to characterize the placement of these modules. Due to problem specific genotype encoding as a binary tree, it is possible to compute the detailed routing at the time of module placement. The algorithm proposes two mutation operators- one which change the partial layouts and the other which changes the structure of the binary tree. A suggestion to implement a new type of recombination operator is made which can replace the existing crossover operators and operate on a pool of fitter individuals. The thesis stresses on using new and problem specific genetic operators in order to arrive at the global optima.

5. Conclusion and Future Scope

Combining the merits of various Genetic Algorithms, it is possible to achieve highly optimum solution which can resolve conflicting parameters simultaneously, leading to circuit designs that would be far better than those available today. There has been a trade-off between achieving optimized area and runtime of an algorithm. Implementing these algorithms collectively will give a better chance to resolve this trade off. Distributed Genetic Algorithm provides the GAPE method which proves its efficacy over simulated annealing, Hybrid algorithm can efficiently optimize the floorplan design due to its local search operator. On the similar lines, Parallel Genetic Algorithms are proven for faster computation having advantages over Sequential Algorithms. Thus in future, we can combine these stellar advantages of these individual algorithms and implement a Genetic Algorithm which will take

care of resolving these parameters simultaneously. This will enable us to implement area optimized designs along with shorter run time of an algorithm and higher efficiency. The work presented in this paper provides a compendium of advantages of these algorithms. Collective implementation and investigation of novel Genetic Algorithms will be a promising prospect for obtaining optimized solutions for complex floorplanning problems.

6. References

- [1] Neil H.E Weste and Kamran Eshraghian, *Principles of CMOS VLSI Design*, Pearson Education, Delhi, 2005.
- [2] Srilata Raman, L.M. Patnail, "Performance Driven MCM Partitioning through an Adaptive Genetic Algorithm", IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 4, NO. 4, DECEMBER 1996
- [3] D. F. Wong and C. L. Liu, "Floorplan Design of VLSI circuits", Algorithmica, Springer, New York, 1989.
- [4] James P. Cohoon, Shailesh U. Hegde, Worthy N. Martin and Dana S. Richards, "Distributed Genetic Algorithms for the Floorplan Design Problem", IEEE transactions on computer-aided design, VOL IO, NO. 4. APRIL 1991.
- [5] Maolin Tang and Raymond Y. K. Lau, "A Parallel Genetic Algorithm for Floorplan Area Optimization", Seventh International Conference on Intelligent Systems Design and Application, IEEE, 2007.
- [6] Lin-Yu Tseng and Tuan-Yung Han, "A Hybrid Genetic Algorithm for the Floorplan Optimization Problem",
- [7] Shingo Nakaya, Tetsushi Koide and Shin'ichi Wakabayashi, "AN ADAPTIVE GENETIC ALGORITHM FOR VLSI FLOORPLANNING BASED ON SEQUENCE-PAIR", ISCAS-2000, IEEE International Symposium on Circuits and Systems, Geneva, 28-30 May 2000
- [8] S. Kirkpatrick, C. D. Gelatt, and M.P. Vecchi, "Optimization by simulated annealing," *Sci. 220*, pp. 671-680, May 13, 1983.
- [9] D. P. LaPotin and S. W. Director, "Mason: A global floorplanning tool," in Proc. IEEE Int. Conf. on Computer-Aided Design, Santa Clara, CA, 1985, pp. 143-145.
- [10] R. H. J. M. Otten, "Automatic floorplan design," in Proc. 19th ACM-IEEE Design Automation Conf., Minneapolis, MN, 1982
- [11] D.F. Wong, P.S. Sakhamuri "Efficient floorplan optimization," in Proc. IEEE Int. Conf. on Computer Design, Port Chester, NY, 1983, pp. 499-502.
- [12] R. H. J. M. Otten and L. P. P. P. van Ginneken, "Floorplandesign using simulated annealing," in Proc. IEEE Int. Conf. on Computer-Aided Design, Santa Clara, CA, 1984, pp. 96-98.
- [13] C. Sechen and A. Sangiovanni-Vincentelli, "The Timbenvolf placement and routing package," IEEE J. Solid-state Circuits, vol. SC-20, pp. 510-522, 1985.
- [14] D. F. Wong and C. L. Liu, "A new algorithm for floorplan design," in Proc. 23rd ACM-IEEE Design Automation Conf., Las Vegas, NV, 1986, pp. 101-107.
- [15] D. F. Wong, H. W. Leong, and C. L. Liu, "Simulated Annealing for VLSI Design."
- [16] L. S. Woo, C. K. Wong, and D. T. Tang, "Pioneer: A macrobased floorplanning design system," VLSI Systems Design
- [17] I.H. Shanavas, R.K. Gnanamurthy and T.S. Thangaraj, "A Novel Approach to find the best fit for VLSI Partitioning – Physical Design", International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom), 2010
- [18] John Uyemura, *Introduction to VLSI circuits and systems*, Wiley Publication, New Delhi
- [19] N.R. Shanbhag, "A Mathematical Basis for Power-Reduction in Digital VLSI Systems", IEEE Transactions on Circuits and Systems II : Analog and Digital Signal Processing, Vol : 44 , Issue : 11, November 1997.
- [20] Zamin Ali Khan, S. M. Aqil Burney, Jawed Naseem, Kashif Rizwan, " Optimization of Power Consumption in VLSI Circuit " , IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011
- [21] Sandeep Singh Gill, Dr. Rajeevan Chandel, Dr. Ashwani Chandel, "Genetic Algorithm Based Approach To Circuit Partitioning", International Journal of Computer and Electrical Engineering, Vol. 2, No. 2, April, 2010 1793-8163
- [22] Jae-Gon Kim and Yeong-Dae Kim, Member, "A Linear Programming-Based Algorithm for Floorplanning in VLSI Design", IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, VOL.22, NO. 5, MAY 2003
- [23] Volker Schnecke, Oliver Vornberger, "A GENETIC ALGORITHM FOR VLSI PHYSICAL DESIGN AUTOMATION",
- [24] Shivanandam and Deepa, Principles of Soft Computing, New Age Publications.
- [25] Satish Kumar, *Neural Networks and Fuzzy Logics*, Prentice Hall of India.
- [26] Timothy, *Fuzzy Sets and Fuzzy Logics*, Prentice Hall of India
- [27] James P. Cohoon, Shailesh U. Hegde, Worthy N. Martin and Dana S. Richards, "Distributed Genetic Algorithms for the Floorplan Design Problem", IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN, VOL IO, NO. 4. APRIL 1991 483
- [28] H. Murata, K. Fujiyoshi. S. Nakalake. and Y. Kajitani. "VLSI module placement based on rectangle-packing by the sequence-pair", IEEE Transactions on Computer-Aided Design, Vol. IS, pp. 1518-1524, Dec. 1996

IJERT