Optimization of VLSI Floorplanning Using Genetic Algorithm

S.Venkatraman[*], M.Sundhararajan

Department of Electronics and Communication Engineering, Research scholar, Bharath University, Chennai, India

***Corresponding author: E-Mail: apece2008@gmail.com**

ABSTRACT

Floor planning is that the terribly central stage in VLSI physical style for class conscious building module style methodology. Floorplanning affords early response that evaluates architectural choices, approximation of chip space, estimates delay, interconnect length and congestion caused by wiring. As technology advances, style complexness is increasing and therefore the circuit size is obtaining larger. Thus space of the circuit gets increase and tougher to minimizing the interconnect length. The VLSI floorplanning is that the NP onerous downside. So it's horribly troublesome to seek out the best solution. During this paper we tend to take into account, a multi-objective genetic algorithmic program primarily based floorplanning has been developed with novel crossover operators to handle the multi-objective floorplanning for Very Large Scale Integration application specific integrated circuits (VLSI ASICs). The Genetic algorithmic program (GA) approach is employed for minimizing the whole space and interconnects length.

KEY WORDS: VLSI floor planning, Genetic Algorithm, Area and Interconnect length.

1. INTRODUCTION

Floorplanning may be an essential step, because it sets up the bottom work for an honest layout. However, it's computationally quite onerous fairly often the task of floor planning is finished by a design engineer, instead of a CAD tool. The method of determinant block shapes and positions with space minimisation objective and ratio demand is mentioned as Floorplanning. A standard strategy for blocks floorplanning is to work out, within the 1st part, the relative location of the blocks to every different supported connection-cost criteria. Within the second step, block size is performed with the goal of minimizing the general chip space and therefore the location of every block is finalized [Tung-Chieh Chen, 2005]. Throughout floorplanning, each location and orientation of a collection of rectangular blocks area unit set specified no blocks overlap. The interconnect is set by the add of the lengths of all the nets. Usually half perimeter inter connect length (HPICL) is employed as an estimate of the whole inter connect length.

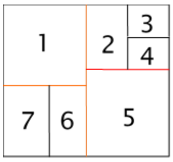

Figure.1. Slicing floorplan

The slicing floorplan is solely explains that the kids of all composite cells may be obtained by bisecting the cell horizontal or vertical direction, the floorplan is named slicing floorplan. So, during a slicing floorplan a composite cell is formed by combining its youngster's horizontal or vertical (putting subsequent to each other from left to right or stacking the on prime of each other).A natural way to represent a slicing floorplan is by means that of a slicing tree. The composite cell is nothing however cell that area unit made of leaf cells area unit known as composite cells. The leaf is that the lowest-level cells area unit known as leaf cells.

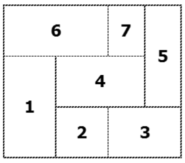

Figure.2. Non-Slicing Floorplan

The non-slicing floorplan is solely explains that the kids of all composite cells can't be obtained by bisecting the cell horizontal or vertical direction, the floorplan is named non- slicing floorplan. Not all floorplans area unit slicing for instance, within the wheel or spiral floorplan. Here the kids of the given cell cannot be obtained by division. The composite cell must be composed of a minimum of five cells so as to not be slicing (Chang-Tzu Lin, 2002).

Irrespective of the hardware implementation alternative, optimisation is needed at all the design levels to get the simplest implementation of the target application. Several of the floorplanning optimisation issues within the design stages are NP-complete issues. Unsatisfiable heuristics is accustomed quickly acquire an answer for the NP-complete issues. But unsatisfiable heuristics don't own any mechanism to escape out of regionally best solutions. Due to this reason, they'll end in solutions of terribly poor quality as they'll bog down in native optima. Stochastic optimisation methods like simulated annealing, Particle swarm optimisation and genetic algorithms create use of random moves to escape out of native optima. In particular time, the random optimisation strategies usually have the power to achieve the globally best resolution (Gracia Nirmala Rani 2016).

Although the necessity for an honest global optimization technique within the hardware style method is obvious, the criticality of the improvement techniques can vary with the planning levels and also the form of hardware implementation. In VLSI ASIC design, the circuit synthesis and physical design stages square measure the foremost crucial stages as they play a big role in getting the most effective system implementation. In FPGA based mostly hardware design, design flexibility and improvement opportunities square measure restricted compared to full custom ASICs because the placement of logic cells and interconnections is mounted. Hence, the FPGA style cycle is extremely sensitive to the high level model of the system. Thus, manufacturing associate degree correct and concise behavioral model of the system is extremely important in FPGA based mostly hardware style. This re-iterates the requirement for elegant IP modules which will be re-used by the designer throughout high level modeling. The availability of an IP module library reduces the complexness of high level modeling. With entire systems being enforced on constant FPGA, several FPGA styles need random improvement cores as a part of the appliance itself (Fernando, Pradeep Ruben, Sherwani, 2009). Since the necessities of the applications square measure widespread, a sturdy stochastic improvement technique that's straightforward to implement in hardware is required.

A genetic algorithm (GA) is a stochastic optimization technique modeled on the theory of evolution in nature. It has been successfully employed for a wide variety of problems including NP-complete problems such as the Traveling Salesman problem (Grefenstette, 1998), real-time problems such as reconfiguration of evolvable hardware (Grefenstette, 1998) and other optimization problems that have complex constraints. Genetic algorithms are easy to implement in software and hardware. They can also be easily adapted to a wide variety of problems. They can make use of existing knowledge about a problem by incorporating successful existing operators into their optimization framework. They are a multi-agent optimization technique and hence can be easily modified for multi-objective optimization.

Problem Description: The floorplanning drawback thought of during this study is that the drawback of putting a group of circuit blocks on a semiconductor chip with the target of minimizing chip size and total wire length. Here, the chip size is defined because the space of the smallest VLSI floorplanning may be a well-studied drawback that a range of optimisation techniques are applied together with simulated annealing (SA), mathematical programming, and genetic algorithms (Hameem Shanavas, 2011). Early floorplanners prohibited space optimisation alone. However with the arrival of the deep sub-micron regime, floorplanners shifted their focus to optimizing interconnect. However if interconnect is that the solely objective to be optimized, the ensuing floorplan can have plenty of unused space. Hence, some floorplanners tried to optimize each space and interconnect. Within the single normalized weighted sum (SNWS) approach to multi-objective optimisation, concurrent optimisation of two objectives implies that the optimizer uses equal weights to multiply the normalized objectives before adding them along to get the one normalized weighted sum. Classical (outline-free) floorplanners supported Simulated annealing [Chen, 2005] use the one normalized weighted sum approach to optimize the two objectives, specifically space and total interconnect. These SA-based floorplanners take issue solely within the data structure, Sequence pair [Chen, 2005] or transitive Closure Graph, used to represent the floorplans. Such floorplanners kind one scalar objective perform mistreatment the two normalized objectives and also the user outlined weights for every objective as shown in Equation below.

$$F=W_1 * \frac{\text{Area}}{\text{Area}_{\max}} + W_2 * \frac{\text{Wirelength}}{\text{Wirelength}_{\max}}$$

Genetic algorithms were first planned for circuit placement by Cohoon and Paris. The primary genetic rule for floorplanning was planned by Cohoon et al and used Normalized Polish expressions to represent floorplans. Later, several different genetic floorplanners were planned that developed novel crossover techniques for various floorplan representation schemes. Hatta et al proposed the primary genetic floorplanner supported the Sequence pair representation that optimized floorplan space. Their genetic floorplanner used two new crossover operators, specifically One-point partially Matched Crossover (OPX) and Uniform partially Matched Crossover (UPX). Hatta et al combined the well-known partially Matched Crossover (meant for permutation-based chromosomes) with two binary crossover operators to formulate two new crossover operators (OPX and UPX) that would handle the Sequence pair floorplan representation. OPX may be a combination of the one purpose crossover and therefore the part Matched Crossover. On the opposite hand, UPX may be a combination of the uniform crossover and therefore the partially Matched Crossover operators.

Modern VLSI floorplanning, as outlined by Kahng, focuses on interconnect optimisation within a fixed chip outline. With the chip complexness increasing with the rising integration technology, hierarchical design ways became imperative. In a hierarchical design flow, floorplanning at the topmost-level may need a versatile chip define. However the floorplans for the modules of the upper levels can fix the floorplan define for the lower level sub-modules. This has leads to an exaggerated importance for the modern fixed outline floorplanning problem. It’s to be noted that in trendy floorplanning, interconnect is that the primary objective whereas space isn't any longer an objective however rather a constraint.

Adya and markov,2003, planned a completely unique simulated annealing primarily based hybrid floorplanning tool known as Parquet that's capable of each mounted and versatile define floorplanning. Parquet uses the one normalized weighted total approach throughout simulated annealing primarily based optimisation however additionally has some heuristic operators that drive the optimisation engine towards solutions that conform the fixed outline constraint. They planned the notion of slack of a module in Sequence pair primarily based floorplanning. Generally, all the modules in an exceedingly floorplan are compacted to the bottom-left corner of the space that they will occupy. To calculate slack, all the modules are in addition compacted to the top-right corners within the floorplan. The slacks of a module i are computed as:

X- Slack (i) =xc_topRight (i)-xc_botleft (i)
Y- Slack (i) =yc_topRight (i)-yc_botleft (i)

Where xc_topRight (i), yc_topRight (i) are the x- and y-coordinates of the lower left corner of module i when the floorplan (i.e., all its modules) is compacted to the top-right corner, and xc_botLeft (i), yc_botLeft (i) are the x- and y-coordinates of the lower left corner of module i when the floorplan is compacted to the bottom-left corner. Multi-objective genetic algorithms and non-domination based solution ranking concepts have been successfully used for various problems belonging to different domains

Andrew Kahng, Jens Leinig, Igor L. Markov, Jin Hu, ―VLSI Physical Design: From Graph Partitioning to timing Closure,‖ Springer, 2011 during this paper, proposed design and optimisation of integrated circuits are essential to the creation of latest semiconductor chips, and physical optimizations are getting a lot of outstanding as a results of semiconductor scaling. Introduces and compares algorithms that are used throughout the physical design section of integrated-circuit design, whereby a geometrical chip layout is created ranging from associate degree abstract circuit design. The emphasis is on essential and basic techniques, starting from hyper graph partitioning and circuit placement to timing closure.

The proposed floorplanner works expressly on coincident improvement of space and interconnect using the selective Non-dominated Sorting based Genetic rule. It's to be noted that the projected floorplanner are often simply extended to perform fastened outline floorplanning by incorporating a penalty perform or by employing a changed fitness assignment.

2. PROPOSED METHOD

The proposed floorplanner is an exclusive non-dominated sorting based multi-objective genetic algorithm employing two novel crossover operators, a set of mutation operators, and a local optimization operator. Proposed Multi-Objective Genetic Floorplanner (cx_rate, mut_rate, N-generations, pop Size, tourney Size).

```
{
    • Population ← Generate random Initial Population (pop Size);
    • For gen in N_generations do {
    • EliteSet ← Non-Dominated Sort (population, Area, Wire);
    • Mate Pool ← Select Mating Pool (population);
    • For i in 1 to cx_rate do {
    • (P1, P2) ← Select Parents (Mate Pool, tourney Size);
    • (Off1, Off2) ← Crossover (P1, P2);}
    • For i in 1 to mut_rate do {
    • MutIndex ← Select Non-Elite Individual (population);
    • Mutate (MutIndex);
    • Update Population (population, MutIndex);}
    • Update Population (population, offspring);}
    • Return (Elite Set (population));
}
```

Figure 3. Pseudo-code of the proposed hybrid Genetic Algorithm

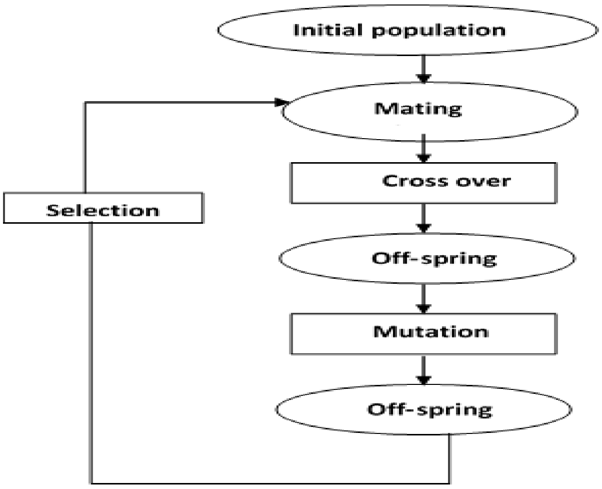

Figure.4. Process flow of Genetic algorithm

Proposed Genetic Algorithm: The proposed genetic floor planner starts with an initial population of every which way generated Sequence Pairs and orientation vectors with a training set of benchmark circuits from both the MCNC and GSRC benchmark suites were conducted to search out the values for the GA parameters that offer the most effective results supported the results of the experimental studies, Equation 1 has been derived to see the GA population size supported the problem size, i.e., range of modules (n). This equation was then applied to size the population by the proposed floorplanner for all the benchmarks. To the most effective of the Author's data, this is often the primary genetic floorplanner that varies the population size in line with the quantity of modules (n) present within the benchmark circuit.

Population size = 10*n ---------- (1)

Sorting of the Population: The proposed floor planner kinds the whole population into varied non-domination levels in terms of space and total interconnect at the start of each generation (line 3 of procedure in Figure 2). All the people within the current population square measure allotted a non-domination rank ranging from zero. The fittest individuals have the smallest amount non-domination ranks**.**

Best Individuals: The number of elite individuals within the current population of the projected GA varies with every generation. All the individuals with very cheap no domination rank (zero) type this set of elite individuals. These individuals don't seem to be subject to mutation. Thus, individuals containing genetic data contributory to reduced space or interconnect are going to be preserved for the longer term generations.

Mating Pool Selection: In each generation the proposed floorplanner selects a pool of the objectives thought of within the proposed GA, the crowding distance of an answer is measured within the objective area. Within the planned GA, all the individuals within the population area unit appointed a crowding distance additionally to the non-domination ranks. The crowding distance of a personal may be a live of the proximity of neighboring solutions within the current population. In multi-objective genetic algorithms, early convergence of a non-dominated front to a little section of the particular front should be prevented. This may be achieved by protective solutions that don't have close neighboring solutions within the individuals to use as parents for the crossover operations in this generation (line four of procedure in Figure.3).

The scale of this pool is ready to half the population size as suggested in. If the quantity of elite individuals is bigger than this size, then the best individuals with the most important crowding distance area unit picked to make the mating pool to confirm that a various set of individual's area unit maintained within the population.

Crowding Distance: Crowding Distance (di) of a solution, si is defined because the distance between resolution, si-1 and si+1 belonging to constant non-domination front because the solution si and are immediate neighbors to the solution si. Crowding Distance may be measured in either the solution encoding space or the target perform area. The target area during this work is two-dimensional as floorplan space and total interconnect are the objectives thought of. Within the planned GA, all the people within the population are appointed a crowding distance additionally to the non-domination ranks. The crowding distance of a personal could be a measure of the proximity of neighboring solutions within the current population. In multi-objective genetic algorithms, premature convergence of a non-dominated front to a small section of the particular front should be prevented. This may be achieved by protective solutions that don't have close neighboring solutions within the current population.

Parent Selection: The proposed GA uses crowded tournament selection to select the two parents for crossover from the mating pool of individuals. In tournament selection, a group of parent candidates is selected randomly from the mating pool. A tournament is played between these candidate individuals to determine the fittest two individuals of the group.

Crossover: The crossover operator (lines 5-8 of the procedure in Figure 3) is employed by genetic algorithms to mix sensible traits from the parents to make extremely work offspring. A good crossover operator should additionally make sure that the offspring doesn't closely match either parent. These properties make sure that the crossover operator explores completely different promising areas within the resolution area. The crossover rate (cx_rate) limits the quantity of crossover performed in any generation

Mutation Operators: The mutation operator (lines 9-13 of the procedure in Figure.3) is employed by genetic algorithms to provide diversity within the population. It conjointly helps in avoiding a fast convergence to local optima. Three mutation operators are employed in the planned genetic floorplanner. A non-elite individual is randomly chosen from the population and one in all the three mutation operators is applied to the individual. Every of the mutation operators has an equal choice likelihood. The mutation rate (mut_rate) limits the quantity of mutation operations performed in any generation. A mutation rate of 0.1 gave the most effective leads to empirical studies and was maintained for all the experiments. The three mutation operators employed in the proposed floorplanner are described below

- Complement a chain (vertical-horizontal).
- Exchange a sub tree and a leaf.
- Rotate a module (M) or sub floorplan.

Population Update: The new population for successive generation shaped by selecting the specified variety of individuals from the combined pool of this population and also the freshly formed offspring population. The combined pool is sorted into non-dominated fronts and every one the individuals belonging to a specific non-domination front area unit derived into the new generation ranging from the non-domination front with rank zero. If the addition of all the individuals during a bound non-domination front leads to violation of the population size, then the specified variety of individual's area unit chosen using the crowded selection operator in order that a uniformly distributed front, with respect to the crowding distance, is obtained.

3. RESULTS & CONCLUSION

The proposed genetic floor planner was applied using C++ and assembled the experiments; the proposed genetic floorplanner was run with a crossover rate of 1.0 and mutation rate of 0.1. These GA parameter values were empirically determined all the experiments were run on a Linux machine with a 3.2GHz Intel Pentium 4 processor and 2GB RAM. All the experiments were run on a Linux machine with a 3.2GHz circuits Intel Pentium 4 processor and 2GB RAM.

Table I. Output result for proposed method (GA) for GSRC benchmark

Benchmark circuit	Area (A)	Interconnect length (W)	Runtime in (s)
N10	229578	32169	0.47
N30	225692	89066	2.17
N50	218964	119878	3.49
N100	200455	195489	9.18
N200	203658	35850	21.82
N300	318999	51246	647.75

Table.2. Area and interconnect comparisons with existing method using MCNC benchmark circuit (A=area in mm^2, W= interconnect length in mm)

Benchmark circuit	SA based Floorplanning		Adapt GA Floorplanning		Proposed Genetic Algorithm	
	A (mm^2)	W (mm)	A (mm^2)	W (mm)	A (mm^2)	W (mm)
apte	46.56	390.57	48.85	400.25	47.30	360.58
Hp	22.22	492.00	30.58	490.69	10.05	246.94
Xerox	10.28	164.54	25.59	195.45	20.56	200.78
Ami33	1.327	57.72	1.69	45.58	1.21	31.33
Ami49	40.66	803.89	37.16	971.3	37.81	677.9

VLSI Floor planning has reworked in to a multi-objective improvement problem with the recent advances in integration technology. Genetic algorithms are extensively employed in completely different forms to resolve numerous multi-objective improvement issues. During this work, the NSGA-II multi-objective genetic algorithmic program has been applied to tackle the VLSI floorplanning problem Considering the floorplan space and total Interconnect length objectives. The hybridized multi-objective floorplanner achieves superb results for the MCNC and GSRC benchmark suites as compared to alternative floorplanners that perform synchronal improvement of space and Interconnect length. Thus, genetic algorithms are often used effectively for multi-objective improvement in VLSI design.

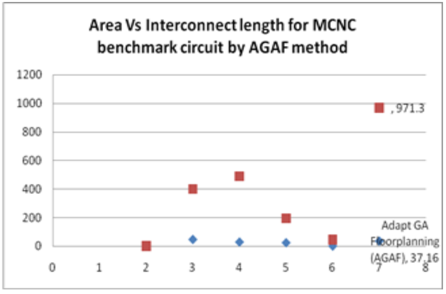

Figure.5. Area and interconnect length optimization for MCNC benchmark circuit using Simulated Annealing floorplanning

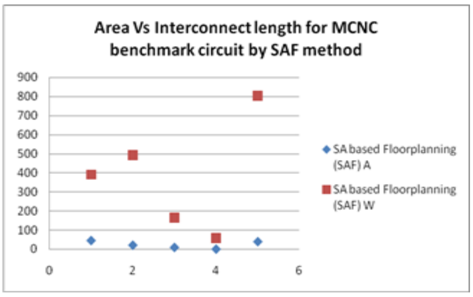

Figure.6. Area and interconnect length optimization for MCNC benchmark circuit using Adaptive GA floorplanning.

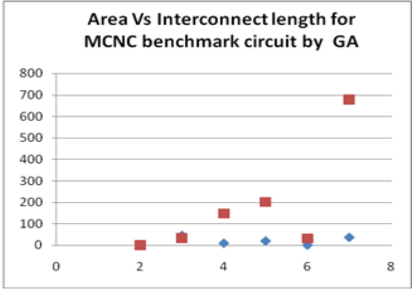

Figure.7. Area and interconnect length optimization for MCNC benchmark circuit using Adaptive GA floorplanning.

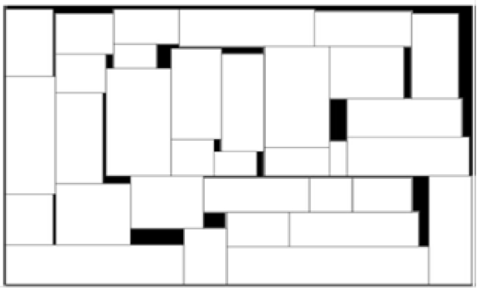

Figure.8. Floorplan of ami33 benchmark with area = 1.21mm2 and total Interconnect length = 31.33mm

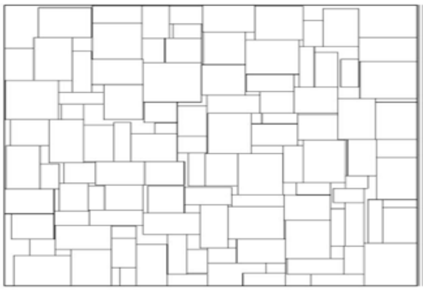

Figure.9. Floorplan of n100 GSRC benchmark with area = 20.0455mm^2 and interconnect length= 19.5489mm.

REFERENCES

Adya S.N and Markov I.L, Fixed-outline Floor planning, Enabling Hierarchical Design, IEEE Trans. on VLSI Systems, vol, 11 (6), 2003, 1120-1135.

Andrew Kahng, Jens Leinig, Igor L. Markov, Jin Hu, VLSI Physical Design, From Graph Partitioning to Timing Closure, Springer, 2011

Chang-Tzu Lin, De-Sheng Chen, Yi-Wen Wang, Hsin-Hsien Ho, Modern Floor planning with Abutment and Fixed-Outline Constraints, IEEE transactions on systems, man, and cybernetics—part b, cybernetics, 39 (1), 2002.

Chang-Tzu Lin, De-Sheng Chen, Yi-Wen Wang, Hsin-Hsien Ho, Modern Floor planning with Abutment and Fixed-Outline Constraints, IEEE transactions on systems, man, and cybernetics—part b, cybernetics, 39 (1), 2002.

Chen T.C and Chang Y.W, Modern floor planning based on fast simulated annealing, In Proc, ACM Int Symp, Physical Design, San Francisco, CA, 2005, 104–112**.**

Cohoon J.P, Hegde S, Martin W, and Richards D, Distributed genetic algorithms for the floorplan design problem, IEEE transactions on Computer-Aided Design of Integrated Circuits and Systems, 10 (4), 1991, 483-492.

Fernando, Pradeep Ruben, Genetic algorithm based design and optimization of VLSI ASICs and reconfigurable hardware, Graduate Theses and Dissertations, 2009.

Gracia Nirmala Rani, Rajaram S, Analysis & Design of VLSI Floor planning Algorithms for Nano-Circuits, IntJAdv Engg Tech, 7 (1), 2016, 527-532.

Grefenstette J.J, Gopal R, Rosmaita B.J, and Van Gucht D, Genetic Algorithms for the Traveling Salesman Problem, Proceedings of the 1st International Conference on Genetic Algorithms, 1998, 160-168, 1998.

Hameem Shanavas I and Ramaswamy Kannan Gnanamurthy, Wirelength Minimization in Partitioning and Floorplanning Using Evolutionary Algorithms, VLSI Design, 2011, 9.

Hatta K, Wakabayashi S, and Koide T, Solving the rectangular packing problem by an adaptive GA based on sequence-pair, in Proceedings of ASPDAC, 1999, 181-184.

Lin J.M and Chang Y.W, TCG, A Transitive Closure Graph-Based Representation for Non-Slicing Floorplans, in Design Automation Conference, 2001.

Sherwani N, Algorithms for VLSI Physical Design Automation, Kluwer Academic Publishers, Boston, Mass, USA, 1999.

Tang M and Yao X, A memetic algorithm for VLSI floor planning, IEEE Transactions on Systems, Man, and Cyber Net, 37 (1), 2007.

Tung-Chieh Chen, and Yao-Wen Chang, Modern floor planning based on fast simulated annealing, Proceedings of the 2005 international symposium on Physical design, San Francisco, California, USA, 2005.