

A Lobal Optimization Algorithm for VLSI Floor planning Problems

Mr.S.Venkatraman¹, P.Joshua Steve Allan², V.Manikandan³, S.Sanjeev⁴, R. Vivek Narayan⁶

^{1,2,3,4,5,6}Department of Electronics and Communication –Vel Tech, Chennai-60006

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

Abstract

Floorplanning is an important step for building a module design methodology. Floorplanning give early feedback that evaluates field of study choices, estimates chip space ,delay and congestion caused by wiring. As technology advances, style complexness are increasing and therefore the circuit size is obtaining larger. To deal with the increasing style complexness, hierarchic style and holding modules are widely used. This makes floorplanning coming up with way more essential to the standard of a really massive Scale Integration (VLSI) style, for several years, floorplanning comes up with could be an essential step, because it sets up the bottom work for a layout. However, the method of crucial blocks, shapes and positions with space step-down objective and ratio demand is observed as floorplanning. Common strategy for blocks floorplanning is to see within the 1st part so the relative location of the blocks to every different supported connection-cost criteria, within the second step, block filler is performed with the goal of minimizing the chip space and therefore the location of every block is finalized. From the machine purpose of read, VLSI floorplanning is NP-hard. The answer area can increase exponentially with the expansion of circuit scale, so it's tough to seek out the best answer by exploring the world answer area. To handle this complexness swarm based optimisation technique has opted during this projected work. A generalize answer has developed to require care of space likewise as interconnection wire length. to realize this weighted objective perform has outlined. the benefits of PSO like simplicity in implementation, and the Stimulated Annealing is used to reduce the hotspots in the circuits and much more effective way Genetic Algorithm is used obtain better optimization techniques

Keywords: Floorplanning, ,Genetic Algorithm ,Stimulated Annealing, PSO ,Area Optimization

1. Introduction

Floor planning is a crucial step in physical style. Floor-planning helps to supply tentative location of IC building blocks. it's vital result helps to see VLSI chips .The goal of floorplanning is to optimize a predefined value , like the realm of a ensuing floor planning given by the minimum bounding parallelogram of the layout region. The floor planning space is directly associated with the chip atomic number 14 value. The larger the realm, the upper the atomic number 14 value. The free area within the floorplanning bounding parallelogram that isn't lined by any module is named white area or dead area. In floorplanning step a style will contain 2 varieties of blocks: onerous and soft blocks. a tough block could be a circuit module with space and ratio fastened. whereas a soft block has fastened space however variable ratio. This paper presents a Satisfiability Modulo Theory primarily based approach for floorplanning with onerous, soft and rotating basic blocks[16-18].

1.1. Slicing Representation:

A slicing floorplan is a rectangular area that is sliced recursively by a horizontal or vertical slicing line into a set of rectangular regions, called rooms, to accommodate a set of circuit modules M. Slicing floorplans are generally represented by slicing trees (or equivalently, by Polish expressions.)

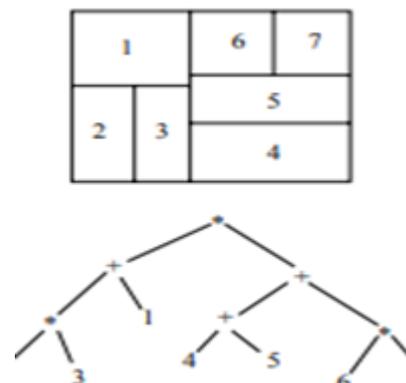


Figure1:Slicing represnetation

the leaf nodes in a slicing tree represent the modules. Each internal node of a slicing tree is labeled by operator or operator +, corresponding to a vertical or horizontal slicing line, respectively. Every subtree rooted at an internal node represents a supermodule consisting of one or more component modules and/or supermodules. In a slicing floorplan, each supermodule has a rectangular shape. The Polish expression for a slicing floorplan is obtained by traversing the slicing tree in postorder

1.2. Non slicing representation:

An O-tree is induced from an admissible placement defined in [Fig1]. A placement is said to be admissible if and only if no module can shift left or down with other modules being fixed; i.e.,

all modules are compacted in both L and N directions. See Figure 2(a) for an admissible placement. An O-tree is an ordered tree structure with an arbitrary number of branches (children) for each node. (Figure 2(b) shows the Otree for the placement shown in Figure 2(a).) As shown in Figure 2(b), the branches are irregular; for example, node % has only one child while T has four. The irregular structure complicates tree operations and incurs sequence encoding. For example, the O-tree

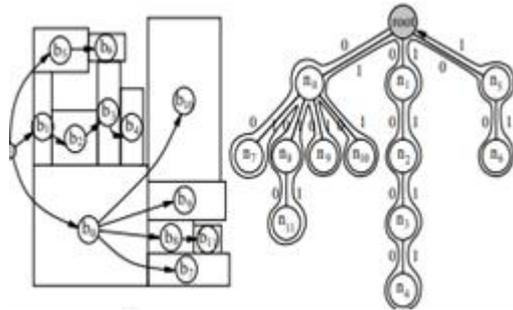


Figure2: Non slicing representation

2. Objective

In this work we used to combine three methods, simulated annealing (SA), genetic algorithm (GA) and particle swarm optimization (PSO), to make a comparison of their performances. Each possible viewpoint is evaluated with a score table representing the wall segments visible from each viewpoint based on scanning geometry constraints. The goal is to find a minimum number of viewpoints that can obtain complete coverage of all wall segments with a minimal sum of incidence angles. The different methods have been implemented and compared in terms of the quality of the solutions, runtime and repeatability. The results obtained in this research show that PSO and GA provide similar solutions while SA doesn't guarantee an optimal solution within limited iterations. Overall, GA is considered as the best choice for this problem based on its capability of providing an optimal solution and fewer parameters to tune.

3. Existing system:

Sutanthavibul [5] developed the floor planning drawback as a mixed number linear program. the strategy projected divides the given drawback into varied sub problems, every sub drawback is solved mistreatment 0-1 mixed number programming technique. The best answer for every sub problem is found and therefore the final answer is achieved mistreatment sequential addition of latest components to Associate in Nursing already existing partial solution. Chen [2] have solved the floor planning drawback in particle swarm optimisation that is population primarily based biological process formula. Their objective is additionally to attenuate the realm of onerous blocks. They derived a technique to handle every kind of placement constraints at constant time as well as preplace constraints, vary constraints, boundary constraints, alignment constraint, abatement, clump etc, during this work they simulated hardening with sequence is employed. Vertical and Horizontal constraint graph is employed to cipher the position of every module. Evangeline and Young [1] used twin binary trees to represent twin binary trees, they begin from the module at the lower left corner and travel upward just in case of left sub tree and to the proper in case of right sub tree. On reaching the lower left corner of another module a node is inserted within the tree. This method is recurrent till all modules are visited. An analogous method is finished ranging from the higher right corner and traveling downward. Then a binary sequence and a mosaic floorplan could be a initiated hence it has zero areas. Then a twin binary sequence is made.

3.1. Simulated Annealing Algorithm:

The simulated Annealing technique was initially developed by Metropolis et al. (1953). It simulates the transcription of particles in an exceedingly body to crystalline state amid the decrease of temperature. The particles of a body move freely at intervals a spread with associate amplitude determined by the blood heat. Provided the cooling is slow enough, the particles will organize themselves in states of progressively lower energy, leading eventually to the state of lowest energy, i.e., the crystalline state (Baselga, 2011). the thought of Storm Troops follow the Monte-Carlo unvaried technique (Berne, 2004):

1) Initial answer x_0 . associate impulsive initial answer x_0 and its objective perform $f(x_0)$ area unit generated. during this paper, x represents a scanning arrange with a group of scanner locations, $f(x)$ is that the quality of this scanning arrange, which is able to be additional processed in segment three.3.

2) Improvement Δx . the advance Δx is generated by a random distribution perform, that reflects the free movement of the particles. one in every of the foremost appropriate performs is that the distribution with density function of:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/2\sigma^2} \quad (2)$$

where σ^2 is that the variance that defines the movement amplitude in every iteration and is set by the present temperature $T(i)$:

$$\sigma_i = \sigma_0(i)^{-1}, 0 < \beta < 1$$

where σ_i is the standard deviation that defines the movement amplitude in each iteration and is determined by the current temperature $T(i)$:

$$\sigma_i = \sigma_0(i)^{-1}, 0 < \beta < 1 \quad (3)$$

where σ_0 = the initial standard deviation

β = the cooling factor

For the temperature in each iteration, some widely accepted cooling schemes are (Baselga, 2011):

$$T(i) = T_0 \log(i+1) \quad (i) = T_0(1+i)^{-\alpha} \quad T(i) = T_0 \alpha i, 0 < \alpha < 1 \quad (4)$$

where T_0 = sufficiently high initial temperature, e.g., 10000°C, so the particles move widely in the body

α = cooling rate

For the application in this paper, σ_0 is determined based on the size of the room so that the candidate solutions can move freely within the entire room. The initial temperature T_0 and cooling factors α and β are empirical values that largely effect the algorithm performance. β is usually set as α^2 to reduce the undefined parameters.

3) Acceptance criteria Equation 5 is used to prevent the solution from falling into a local minimum.

$$x_{i+1} = \begin{cases} x_i + \Delta x & \text{if } f(x_i + \Delta x) < f(x_i) \\ x_i & \text{otherwise with a probability } p > 0 \end{cases} \quad (5)$$

$$\text{where } p = e^{-f(x_i + \Delta x) - f(x_i)} / T(i) \quad (6)$$

4) Repeat step 2 and 3 until the stop criterion is reached. In this paper, the stop criterion for all methods is the maximum number of iterations.

4. Proposed system:

4.1. Genetic Algorithm

Developed originally by Holland (1975), the genetic algorithm is based on Darwinian evolutionary theory of survival of the fittest. A new population (i.e., a group of candidate solutions) is generated from old ones based on some genetic rules. Each solution is evaluated by its fitness until the best solution is found. The basic principle of GA is outlined as follows:

1) Initial population. Generate a random population with i chromosomes, x_0 , and their objective functions, $f(x_0)$. The chromosomes x_i in this paper are i scanning plans with different sets of scanner locations and (x) represent the quality of each plan.

2) Generate a new population. A new population is created based on Darwinian evolutionary theory with three genetic operators as shown in Figure 2:

- **Selection:** Select two chromosomes from a population with a probability based on their objective functions;
- **Crossover:** Elements of two parent chromosomes are crossed over based on a certain rule to create two children chromosomes;
- **Mutation:** Elements in an arbitrary chromosome is mutated with a mutation probability.

These three operations are repeated until each chromosome in the population have been modified (i.e., a new population is generated).

3) Keep generating new population until the stop criterion is reached and the chromosome with the minimum objective function is considered as the optima.

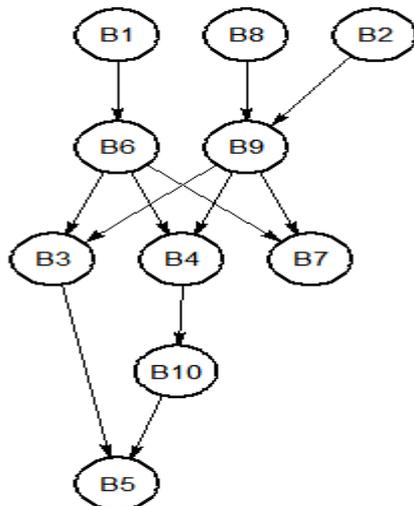


Figure3: Node representation of GA

4.2. Particle Swarm Optimization:

The particle swarm optimization (Kennedy, 2011) is based on the movement of a group of birds (i.e., particles). Each particle flies in a defined search space to discover its best solution, and adjusts its movement based on its own flying experience as well as the flying experience of other particles (Doma and Sedeek, 2014). The PSO algorithm has four main steps:

1) Initial particles. Generate i particles with random positions x_0 within the search domain D , random velocities v_0 and objective functions $f(x_0)$. Similar to GA, particles x_i are i scanning plans with different sets of scanner locations, and (x) are the quality of each plan.

2) Velocity update. The velocity of each particle v_k is updated based on the local optimum position p_i of this particle over time, and the optima of all particles pg :

$$v_{k+1} = wv_k + c_1 \text{rand}(p_i - x) \Delta t + c_2 \text{rand}(p_g - x_k) \Delta t \quad (7)$$

where w = inertial factor

c_1 = self-confidence factor

c_2 = swarm confidence factor

The first two terms represent the influence of current motion and previous optimal motions of this particle; the third term is the influence of the optimal particles. Factors w , c_1 and c_2 are empirical values .

Position update. The positions are updated based on their velocities with Equation 8:

$$x_{k+1} = x_k + v_{k+1} \Delta t \quad (8)$$

The idea of PSO algorithm is depicted

4) Termination criterion. Step 2 and 3 are repeated until the stop criterion is met. Finally, the position with the minimum object function will be saved as the optimal solution

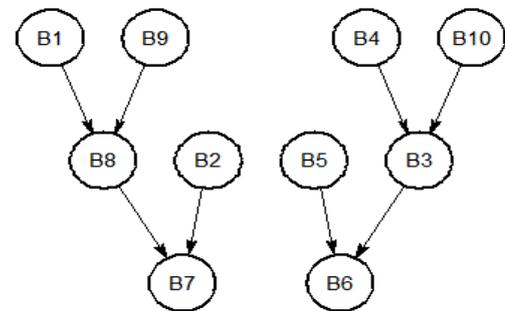


Figure4: Node representation of PSO

To solve the problem of the increasing VLSI design complexity, hierarchical design method is widely used. In this work, the proposed method is to apply simplified model with hierarchical design method together to speed up floorplanning and the whole flow. The traditional hierarchical design process is shown in Figure 1, a 10M design need 10h for quick placement, 3h for trial route, 3h for timing analysis, 10h for timing budget, and then partition, block implementation and top implementation, assemble and optimization before timing signoff, finally the runtime of this method was more than 26h. If adopting proposed method in hierarchical design as shown in Figure 2, although model creation took 5h, it could dramatically save runtime of the next series of processes, for example the quick placement took 2h and timing budget took 1h, so the whole flow runtime could be significantly reduced with the proposed method .Units are the special measure which simulate the location and area of removed logical units to remain original utilization and area for more accurate floorplanning. In addition, there are other advantages of flex filler: Firstly, a flex filler's area is dozens larger than standard modules, so it can relatively reduce the number of modules in net list to reduce design scale by simplifying internal logics. Secondly, flex filler itself does not have any logical.

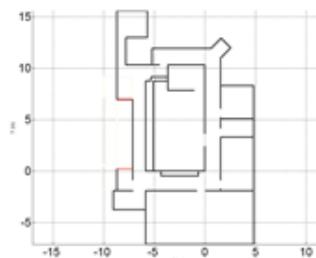


Figure5: Module Representation

Also, floorplanning is supposed to adjust by results f P&R, so the reduction will speed up timing analysis and evaluation of floorplanning while reducing P&R runtime, finally overall floorplanning runtime can be reduced to a large extent. Figure 5 shows the flex fillers in EDA tool’s physical view, selected rectangular box is flex filler whose size is much larger than the maintained interface logics in lower left corner.

4.3. Experimental results and analysis:

This work was implemented in EDA physical style code “Innovus” (the original name is Encounter), experimented on a two.1M automotive electronic chip style and five alternative chips.[11] Figure 6 half-dozen is that the chip with none physical style. The simplified model supported ART technology could be a phyOsical level simulation model with reduced net list, replace inactive logics with flex fillers and stay active logics .The experiment outlined eighty nine modules as simplified models and gave a close model creation report in Figure seven. The initial instances is 2086769, about 2.1M, after modelling, it declined to 128695+71829=0.2M, therefore this methodology reduced style size from two.1M to 0.2M, decreasing to nine.6% of original net list. With reduced net list, fast floorplanning methodology was easier to implement temporal arrangement analysis, fast placement, prewrote, improvement and budgeting.

To prove it, we have a tendency to recorded the runtime of every step before partition step in stratified style flow and compared the

runtime between style flow while not model and with model. As shown in Table 1, while not simplified model, total flow runtime before partition step was 807.95 minutes, however with simplified model, total flow runtime before partition step was a hundred forty five.08 minutes. Therefore runtime reduced by five.57 times, and peak memory small by two.22 times from 20G to 9G. Besides, in Figure eight, it's the chip floorplanning result with simplified model, and therefore the simplified models was marked with colour squares. Actually, the tactic not solely will dramatically speed up stratified physical style flow, it'd conjointly keep smart temporal arrangement and congestion results on one-pass flow. it used RC extraction engine to report temporal arrangement once aggregation style, there have been all half-dozen.87e+05 methods, and that we centred on worst negative slack (WNS) of “reg2reg” - 0.048ns, that was cheap for one -pass flow after we evaluated design’s time closure. The congestion reports showed the horizontal congestion was zero.10% and vertical congestion was zero.35% that was acceptable during a two.1M design.

It gives the better area optimization result (2.27%) compared to the other optimization algorithms. When compared the wire length metric, it provides the best optimization (87.71%) related to the SA algorithm only. Table 2 exhibit the runtime comparison between the optimization algorithms. It clearly shows that the DE algorithm has a less computation time with other algorithms. Because the DE algorithm has a few control parameters to find an optimal function

Table1:Runtime analysis

MCNC Benchmarks		Previous research			Proposed method
Circuits	Modules#	Simulated Annealing	Fast SA	Hybrid Simulated Annealing (sec)	Proposed method
apte	9	7	2	3.95	0.48
xerox	10	25	5	0.57	1.09
Hp	11	55	20	0.27	0.58
ami33	33	34.17	19	5.1	2.08
ami49	49	47.52	28	6.78	5.05

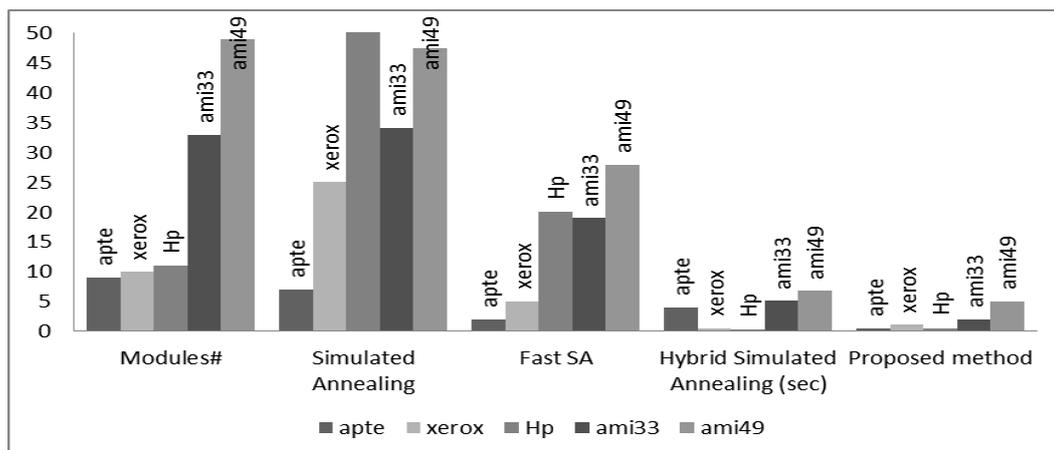


Figure 6:Runtime analysis

Table2:Area Optimization

MCNC Benchmarks		Previous algorithms				
Circuits	Modules	Simulated annealing area (mm2)	EA simulated annealing (mm2)	Hybrid simulated annealing (mm2)	genetic algorithm (mm2)	Proposed Method(mm2)
apte	9	46.92	49.43	47.12	46.9	46.63
xerox	10	19.83	20.78	20.89	20.03	19.69
hp	11	8.95	9.364	9.47	9.08	9.29
ami33	33	1.27	1.253	1.21	1.19	1.22
ami49	49	36.8	36.74	37.8	37.49	36.22

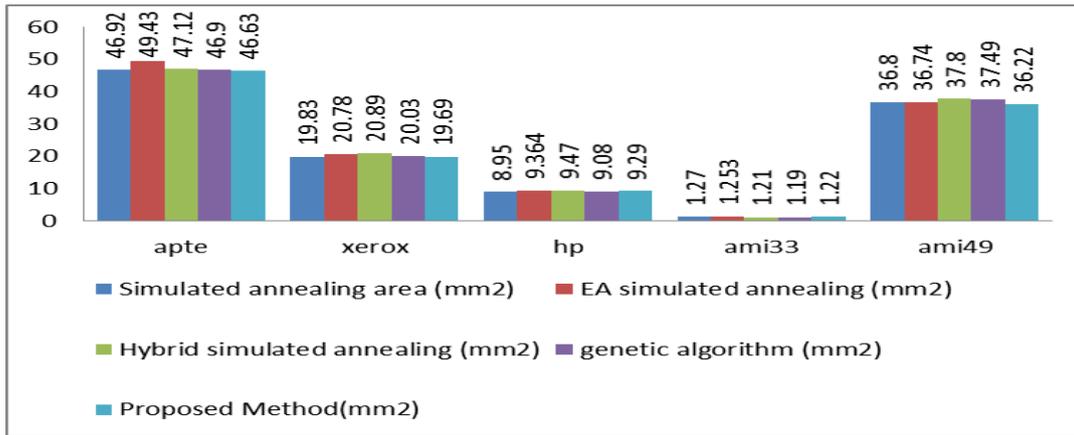


Figure 7:Area Optimization

The proposed algorithm with alignment and performance constraints have compared with SA framework as shown in Table 1&2. The proposed approach adopts a DE algorithm without resorting to floor plan representations, and the placement constraints of the buses are satisfied during the block-packing process. The following notations will be used in the benchmark circuits: xerox-1 and hp-1 contains four alignment blocks only. Xerox-2 and hp-2 contains four alignment blocks and two performance blocks. ami33-1 and ami49 -1 consists of four and five alignment blocks ami33-2 and ami49-2 contains four alignments and three performance blocks. Table 6 and 7 compares area (2.05%) and runtime (43.99%) between SA and DE with alignment and performance constraints. Table 8 represents the wire length estimation of the algorithm. When the performance blocks are used, it reduces the critical net delay and it leads to decrease in the total wire length of the module In order to further verify the effectiveness of proposed method, we experimented on several different huge size of designs to compare traditional method and proposed method in hierarchical physical design flow. In Table 3, the quick floorplanning method improve runtime by 6.2 times and reduce memory by 2.8 times on average. It provides the foundation for smooth implementation of P&R in hierarchical physical design and enhance the performance of EDA tool in VLSI physical design and traditional method flow comparison shown in fig 7-8.

5. Conclusion

EDA tools are used in more effective way to meet speed and quality requirement of IC chip design. Floorplanning is the basis for P&R in physical design, but it usually accounts for about one-third runtime in whole flow to find a suitable floor plan, so shortening the floorplanning and evaluation runtime is an effective way to speed up chip's physical design. Using the logical reduction method to create model for floorplanning in top-down

hierarchical physical design flow can effectively reduce design size by reducing irrelevant information of this step in net list, then it can quickly predict if design meets timing and congestion requirement after physical design implementation. It proved the method can drastically reduce the runtime by 6.2 times and memory by 2.8 times .The comparison of these three algorithms prevents the better optimization between them .The best value is take out for further improvements

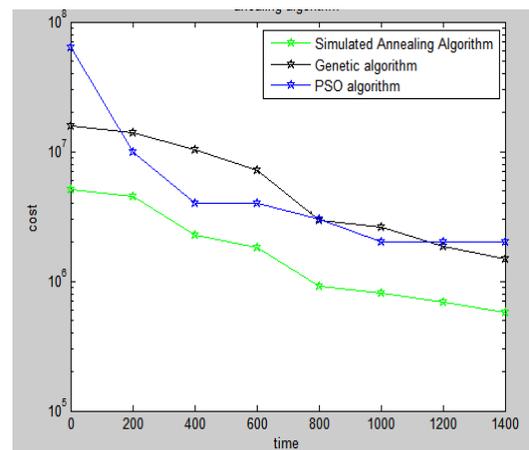


Figure8:Comparative Cost Analysis with Different Algorithm.

References:

- [1] M. Tang and X. Yao, "A Memetic Algorithm for VLSI Floorplanning," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 37, no. 1, pp. 62-69, Feb. 2017
- [2] Chang-Tzu Lin, De-Sheng Chen, Yiwen. Wang ,”An Efficient Genetic Algorithm for Slicing Floorplan Area Optimization,”

- Proceedings of the International Symposium on Circuits And Systems, pp. 879-882, 2016.
- [3] Valenzuela C.L. and Wang P.Y., "VLSI placement and area optimization Valenzuela C.L. and Wang P.Y., "VLSI placement and area optimization using a genetic algorithm to breed normalized postfix expressions," IEEE Transactions on Evolutionary Computation, pp. 390-401, 2017
- [4] R. B. Singh, A. S. Baghel and A. Agarwal, "A review on VLSI floorplanning optimization using metaheuristic algorithms," 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, 2016, pp. 4198-4202
- [5] B. Gwee and M. Lim, "A GA with heuristic based decode for IC floor planning," Proceedings of the International Symposium on Circuits And Systems, VLSI J., vol. 28, no. 2, pp. 157-172, 2013
- [6] P. Moscato, "On evolution, search, optimization, genetic algorithms a Towards memetic algorithms," International Conference on Electrical, Electronics, and Optimization Techniques .Pasadena, Tech. Rep. 826, 2017
- [7] S. Nag and K. Chaudhary, "Post-Placement ResidualOverlap Removal with Minimal Movement", Transactions on Systems, Man, and Cybernetics, pp. 581-586, 2018
- [8] H. Murata, K. Fujiyoshi, S. Nakatake and Y. Kajitani, "VLSI module placement based on rectangle-packing by the sequence pair", IEEE Transactions on Evolutionary Computation. on CAD, vol 15(12), pp.1518-1524, 2015
- [9] Isao Tazawa, Seiichi Koakutsu, and Hironori Hirata, "An Immunity based Genetic Algorithm and its Application to the VLSI Floorplan Design Problem," Proceedings of IEEE International Conference on Evolutionary Computation, pp. 417-421, 2018.
- [10] Ho, and X.-I. Chang, "An efficient generalized multiobjective evolutionary algorithm," Proceedings of the Genetic and Evolutionary Computation Conference, pp. 871-878, 2014.
- [11] Yun-Chih Chang; Yao-Wen Chang; Guang-Ming Wu; Shu-Wei Wu, "B*-trees: A New Representation for Non-slicing Floorplans," Proceedings of the Design Automation Conference, pp. 458-463, 2000.
- [13] D. F. Wong, and C. L. Liu, "A New Algorithm for Floorplan Design," Proceedings of the Design Automation Conference, pp.101-107, 2016.
- [14] Valenzuela C.L. and Wang P.Y., "VLSI placement and area optimization using a genetic algorithm to breed normalized postfix expressions," IEEE Transactions on Evolutionary Computation, pp. 390-401, 2017.
- [15] Chang-Tzu Lin, De-Sheng Chen, Yiwen.Wang, "An Efficient Genetic Algorithm for Slicing Floorplan Area Optimization," Proceedings of the International Symposium on Circuits And Systems, pp. 879-882, 2014.
- [16] E.K. Burke and G. Kendall, "Evaluation of Two Dimensional Bin Packing Problem using the No Fit Polygon", Proceedings of the 26th International Conference on Computers and Industrial Engineering, Melbourne, Australighjina, 15-17 December 2015, pp 286-291
- [17] Dr. AntoBennet, M, Sankar Babu G, Suresh R, Mohammed Sulaiman S, Sheriff M, Janakiraman G ,Natarajan S, "Design & Testing of Tcam Faults Using TH Algorithm", Middle-East Journal of Scientific Research 23(08): 1921-1929, August 2015 .
- [18] Dr. AntoBennet, M "Power Optimization Techniques for sequential elements using pulse triggered flipflops", International Journal of Computer & Modern Technology , Issue 01 ,Volume01 ,pp 29-40, June 2015.
- [19] Dr. AntoBennet, M,Manimaraboopathy M,P. Maragathavalli P,Dinesh Kumar T R, "Low Complexity Multiplier For Gf(2m) Based All One Polynomial", Middle-East Journal of Scientific Research 21 (11): 2064-2071, October 2014.