SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V9I12P111

Volume 9 Issue 12, 126-133, December 2022

Original Article

Partitioning of VLSI Circuits on the basis of Standard Genetic Algorithm and Comparative Analysis of Partitioning Algorithms

P. Rajeswari[1], Theodore S Chandra[2]

[1,2]*Department of ECE, Dayananda Sagar University, Bangalore, India.*

[1]*Corresponding Author : prajeswarisugans@gmail.com*

Received: 20 October 2022 Revised: 27 November 2022 Accepted: 10 December 2022 Published: 25 December 2022

Abstract *- Circuit segmentation or partitioning is one of the important issues in the VLSI physical design scheme. It appears at certain stages in the VLSI design scheme, such as the logic and physical design schemes. The circuit dividing issue is remarkably difficult. The potential of genetic algorithms has been harnessed to take care of many computationally difficult issues on the grounds that current conventional techniques cannot make the expected forward leaps related to complexity, time, and cost. This paper presents and deals with the issue of segmentation of a circuit using a genetic algorithm. The programme provides a number of vertices that are closely related to each other but exceptionally distinct from other divisions. Minimizing the reduction in VLSI circuit segmentation is the highest priority. Other than this, minimum deductions are also included for upgrading various constraints like power, delay, and area. In any case, due to the continuous advancement of semiconductor advancements, a VLSI chip can contain too many semiconductors, and subsequently, the size of the circuit segmentation issue becomes too large. Large segmentation strategies can certainly affect the presentation and cost of a VLSI chip*.

Keywords *- VLSI physical design flow, Circuit partitioning, DYPSO, Genetic Algorithm, Chromosomes.*

1. Introduction

Partitioning is a methodology to separate a circuit or structure into a grouping of additional littlest components. [1] It is an arrangement task to break a gigantic framework into parts completed on discrete interfacing parts. Afterwards, it is again an algorithmic strategy to settle different and complex combinatorial upgrade issues, as in reasoning or configuration mix. The principal clarification that distributing transformed into a central and a portion of the time fundamental arrangement task today is the huge addition of multifaceted structure design previously and the typical further advances of microelectronic system plan and production. This study aims to create a class of iterative computations for multiobjective VLSI partitioning such that the circuit delay and interconnect the desirable prerequisite constraints cut set. With the progress of VLSI development, the amount of circuit parts executed on the VLSI chip is extending bit by bit. A considerable number of semiconductors close by their different interconnections can be placed on a lone chip today. With such endless interconnections running over the chip district, it makes a big difference to find approaches to decreasing the overall length of the running wires across the chip locale since this can impact various limits of the interconnects like time delay in the signal, causing power usage in the interconnections, area of chip got by the interconnections. Circuit separating is a critical stage in VLSI real arrangement. This incorporates the partition of a circuit into additional unassuming parts for the straightforwardness of plan, plan and testability. The issue includes isolating the circuit net summary into two subsets, and a piece of the edges are moreover cut. The amount of edges in a location with two remarkable parts is the cost of a bundle. The objective ability gets the interconnection information, and the splitting game plan is improved regarding the interconnection between the parts with the necessity of outlining changed portions.

2. VLSI Circuit Partitioning

It is the task of separating a circuit into smaller parts. VLSI design is a difficult process, and in this way, it is divided into no intermediate levels. Partitioning is also considered an NP-Complete problem, indicating that polynomial-time computation can solve the issue. The real planning stage must include VLSI circuit partitioning. The essence of circuit partitioning is to divide a circuit into various sub-circuits with the fewest possible connections between them. This can be achieved by recursively dividing a circuit into two sections until we arrive at the wanted degree of intricacy. Therefore, three-way partitioning is a fundamental issue in circuit partitioning, which can be described as [2]. The goal is to partition the circuit into sub-parts to such an extent that the size of the sub-parts is inside recommended ranges, and the quantity of interconnections between the parts is limited. Different divisions bring about various circuit executions. It proposes an alternative way to deal with taking care of the circuit dividing issue. In this work, an evolutionary

calculation approach is utilized. As a result, proper partitioning can significantly improve circuit performance while also lowering design costs.

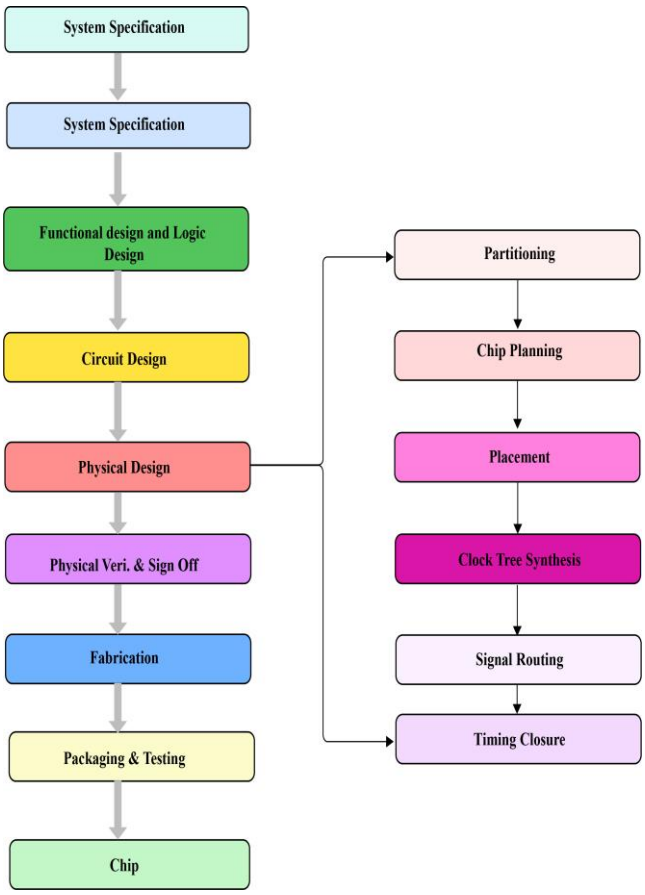

Fig. 1 Block Diagram of Physical Design Flow

3. Literature Survey

In the past, scientists have presented parcel-circuit with their hypotheses. To solve the problem of neighborhood maxima/minima, the work of [1] offered equipment hereditary calculation by developing GA and local inquiry processors that make use of external memory. The possibility of selecting a chromosome has been expressed by [1] as having the capacity of both the best and the most appallingly bad chromosome. Two GA have been proposed by [2], one considering 0-1 encoding and the other considering number encoding. The research in [3] led to the development of a flexible technique for circuit partitioning in which execution performance is improved by varying the population size, hybrid rate, and transformation rate. The problem of dividing can be viewed as a problem of chart partitioning, where each module (such as doors) is considered a vertex, and the association between them deals with the edges between the hubs [4]. Every vertex in the diagram should be treated as a space that can address a rational entryway for the computation, and an edge should be used to address an association [5]. The transformative calculation (EA) and neighborhood search are combined in the memetic computation (pp. 6, 7). (LS). The EAs are used to consider the global ideal. The LS used in this case will increase the assembly speed for the EA. The grouping technique put forth by [8] can reduce the size of large-scale apportioning problems without sacrificing the quality of the partitioning arrangement. The suggested bunching calculation's display is evaluated using the ISPD98 benchmark suite, a common arrangement of dividing benchmarks. In [9], multiobjective hypergraph partitioning calculations have been proposed in light of the staggering worldview. These calculations are suitable for producing arrangements in which the cut and the greatest subdomain degree are simultaneously constrained. The simplification of VLSI interconnection (netlist) bipartition has been proposed in [26] using a discrete Molecule Multitude Enhancement (DPSO) calculation. A transformative calculation known as Memetic Calculation (Mama) includes at least one neighborhood search step as part of its development cycle. Mama has submitted an application to a local search for improving the VLSI division in [11]. An alternative method that was previously presented has been introduced in [12]. A method employing the contiguous framework of a chart for the layer-task issue has been suggested in [13]. As the first stage in automating VLSI real plans, [27] has presented a multitude-based heuristic methodology for addressing adjusted min-cut circuit partitioning. [15] are aware of the various heuristic approaches to the problem of circuit partitioning. They have also conducted a comparative analysis of several computations that have been suggested. 3D-coordinated circuits (3D-ICs) are a new technology that has enormous promise. Little impact area and vertical interconnections between different bits of dust are appreciated by 3D-ICs, allowing shorter wire lengths between doors. They then exhibit lower connection delays and lower power consumption. The 3D Partitioning and Layer task is one of many advancements included in the planned stream for 3D integrated circuits. This action is crucial since the outcome will affect how future developments are displayed. Like previous division problems, this one is also an NP-hard problem. The nature of partitioning is continuously characterized by parameters like layer task, area of I/O terminals, TSV minimization, and region adjusting. To achieve these objectives, Forbidden Search and Recreated Tempering has been used [16]. [17] has described a way of obtaining a base cut using the Discrete Molecule Multitude Improvement (DPSO) calculation and the Discrete Fire Fly Algorithm, a multitude-based heuristic methodology (DFFA). A 3D floorplan partitioning calculation has been proposed by [28]. The suggested method combined an expense-based heuristic and a power-coordinated computation, putting the hubs into consideration for both enticing power and horrifying power to address the long-net issue. Numerous important uses of hypergraph division exist, such as in VLSI design or logical registering. The staggered memetic calculation has been suggested by [19] as a solution to the problem, focusing on arrangement quality. New, potent staggered recombination and transformation jobs that produced a lot of diversity were essential components of commitment. Another diagram partitioning problem is introduced in [30] and transformed into a Profundity-limited Levelized

Chart Dividing (DLGP) problem, which is optimally resolved by means of a potent programming computation. According to an example application, DLGP can create timing-correct circuit solutions for Single Motion Quantum (SFQ) theory, a lovely heartbeat-based, door-level pipelined superconductive registering texture. A staggered circuit division calculation in light of the superior KL calculation has been proposed in [21], combining the possibilities of better KL calculation and staggered partitioning calculation. The fair bipartition hypothesis of diagrams, which has been developed by providing a reasonable introductory packet, is based on this premise.

4. Dypso Overview

DYPSO is the same as transformative calculation strategies in that a population of likely answers for the issue is utilized to test the hunt space. In PSO, however, each individual in the population has a variable speed (position change) at which they move in the chase space. Moreover, every individual has a memory, reviewing the best spots of the pursuit space they have ever visited. Subsequently, its improvement is a gathered speed increment towards its best recently visited position and towards the best individual in a topological area. Suppose that the hunt space is D-layered; then the A D-layered vector, Xi = [xi1, xi2...xiD], can be used to address the I-th molecule in the multiplicity. Another D-layered vector, Vi = [vi1, vi2,... viD], can be used to address the speed (position change) of this molecule. Pi = [pi1, pi2,... piD] is intended to represent the I-th molecule's most recently visited position. Defining "g" as the file of the best molecule between a variety of molecules (i.e., the g-th molecule is the best), "n" is the best seen by that specific molecule, and let the superscripts mean the cycle number. Afterwards, the multitude is controlled by (1) and (2).

$$V(n+1)id=[w.Vnid+C1r1(Pnid-Xnid)+C2r2(Pngd-Xnid)] \quad \ldots\ldots (1)$$

$$Xnid+V(n+1)id=Xnid+Xnid+V(n+1)id \ldots\ldots (2)$$

w is known as the inertial weight; c1 and c2 are two positive constants; c1 is known as the mental boundary, and c2 is known as the social boundary. The task of the inertial weight w in Condition (1) is considered significant for the union way of behaving of the PSO. Inertial loads utilize to control the impact of the previous history of speed on the present. In like manner, the boundary w characterizes the compromise between the worldwide (expansive) and neighborhood (close by) investigation capacities of the multitude of controls. An enormous inertial burden works with the worldwide investigation (disclosure of new regions), while a small one works with neighborhood investigation. A fitting incentive for the latency load generally gives harmony between worldwide and neighborhood investigation capacities, and subsequently, ideal arrangement discovery diminishes the quantity of cycles required. The boundaries c1 and c2 are not vital for the combination of the PSO. Be that as it may, legitimate adjusting can bring about quick union and decrease neighborhood minima. The boundaries r1 and r2 are used to keep up with the variety of the populace, and they are equitably circulated over the reach [0, 1]. PSO exhibitions are exceptionally dependent on the quality of boundaries selected. However, it is impossible to predict these boundaries' ideal qualities. Hence, the ideal choice is to foster a versatile climate so that, as per the prerequisite, it can pick the appropriate value. In this paper, three distinct environmental advancements are thought of. (3). Set the idleness to a high value at first to speed up the global search of the inquiry space, and then slowly lower it to get more modern solutions.

$$W=Mxw-n.(Mxw-Mxw)/(|0.75*Mxn|) \ldots. (3)$$

Where Mxw and Mnw are the greatest and least weight values, I is the redundancy number, and MXn is the most extreme number of reiterations.

5. GA Highlights

Charles Darwin's theory of natural evolution and his idea of "survival of the fittest" served as the basis for developing genetic algorithms, a well-known class of evolutionary algorithms. Genetic algorithms acquired a lot of traction after it was demonstrated that they could be used to solve a variety of complex issues more quickly and efficiently, including optimization problems like the travelling salesperson problem, searching difficulties, learning problems, scheduling problems, placing and routing challenges, etc. Even after roughly 50 years since their invention by John Holland and his students, genetic algorithms are still a common solution for many optimization issues. It provides a small indication of these algorithms' potential. In contrast to other algorithms, genetic algorithms take a directed approach and can do multi-dimensional searches in addition to a partially random method [25]. However, randomization is still maintained with the expectation that the global maximum will be attained by adding a "mutation" operator. Genetic algorithms have a unique property that gives them more realistic answers than other modern algorithms while running quickly.

5.1. Gene-based Algorithms Terminology

5.1.1. Chromosomes

It is necessary to provide a representation of the problem's solutions that the computer can understand before applying these algorithms to a particular problem. These images are referred to as "chromosomes." Fields on each chromosome stand in for different aspects of a problem. Genes are the names of these fields. The genetic algorithm makes these chromosomes randomly and then looks for the best one to use as the solution to the problem.

5.1.2. Crossover

For each iteration, the genetic algorithm aims to produce better generations. At a specific "crossover point", chromosomes must "cross over." This indicates

that specific locations on two or more chromosomes, referred to as "parent chromosomes", are chosen, and the genes at those locations are switched out to create "child chromosomes." The resultant offspring, therefore, carries both parents' DNA (properties of both the parent chromosomes). The terms "crossover" and "reproduction" describe this process. There is no guarantee that the crossover will happen just once for each pair of chromosomes.

The total number of times a pair of chromosomes is utilised for reproduction can be determined by various methods. Rationing and ranking are two often utilised techniques [8]. • In rationing, a chromosome's fitness determines how quickly it reproduces. Higher fitness chromosomes procreate and contribute to more children. By doing this, the algorithm may quickly achieve a population of fit people. An idea known as the "Roulette Wheel" is presented to facilitate this. This genetic operator is based on However; there is a drawback to this approach. We might only obtain a local maximum if the algorithm identifies a dominant individual too early and prevents the chromosomes from replicating[10]. The ranking approach, which sorts chromosomes according to their fitness and assigns them a certain rank, is an alternative to the rationing method. Then, according to their rank, they reproduce.

Higher-ranked individuals have a higher chance of reproducing, while lower-ranked individuals have a reduced chance of doing so. Once more, this uses a slightly modified version of the "Roulette Wheel" method. Figure 2 depicts a visual representation of the crossing. As shown in Fig. 2, the genes from both parents are passed on to the children made from the chromosomes, giving them new and unique traits.

The offspring produced from the chromosomes share the genes of both parents to inherit new and unique characteristics.

5.1.3. Mutation

A form of "hill climbing" algorithm, genetic algorithms can occasionally become trapped at a local maximum. Like in biology, a gene on a chromosome can change randomly due to mutation. This is occasionally injected into the chromosomes to ensure that the solution we arrive at is a global maximum rather than a local maximum. Additionally, some crucial genes will probably be lost along with the chromosomes chosen to create the next population during the crossover phase. It may cause the application to take little time to complete the task or altogether alter the final outcome. So, to stop this from happening, it is sometimes necessary to add helpful genes (information) to the chromosome.

5.1.4. Fitness Function

Simply put, a "fitness function" is a mathematical model depicting a chromosome's survival capacity in a given setting.

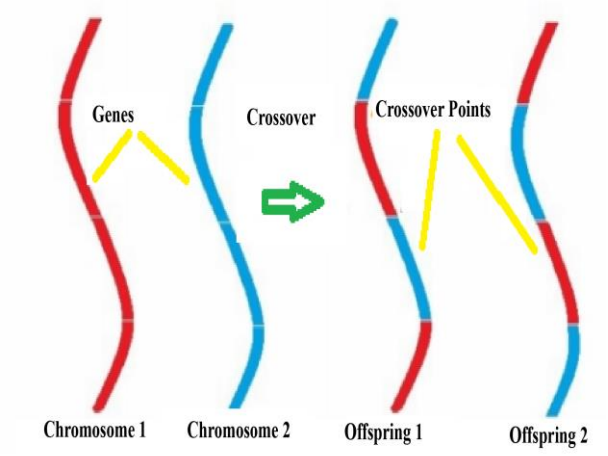

Fig. 2 Illustration of a Genetic Crossover

Every problem is unique, and each solution is created to hold all the essential components of the system to which the genetic algorithm is applied. Only the fitness function can determine how good the final solution will be. Genetic algorithms' benefits Since genetic algorithms were created in the 1960s. They have grown in popularity since they have many advantages over other algorithms of a similar nature. Among the benefits are • Any challenging issue can be handled using genetic algorithms as long as we can encode system attributes into a fitness function and represent the qualities in the form of a chromosome. • Compared to previous algorithms, we can search within a vast search space rather quickly while also greatly compressing them. This is because these algorithms employ a distinctive method that involves selecting a wide range of points from a search space, conducting a directed search within that space, and then progressively convergent towards the ideal answer [8]. A hill climbing algorithm is a genetic algorithm. Infrequent mutation operations do not become trapped at the local optima values because of the infrequent mutation operations, in contrast to conventional hill-climbing algorithms. They ensure that the final output is a global optimum and not a local optimum, even if this step adds a tiny bit of time to the overall solution time. The fact that the issue search space does not have to be continuous is another significant benefit of genetic algorithms. Even though it is explicitly described, better solutions can be created by genetically modifying the intermediate solutions (solutions that were originally not defined in the search space). As a result, these algorithms significantly aid in solving issues where there is a lack of information about the search field. If configured appropriately, the user may manage the genetic algorithms and select when to stop the flow, which makes them particularly useful for issues with several solutions. If there are multiple solutions to a problem and we keep running the algorithm after discovering the first one, it may also find the remaining solutions.

6. Proposed Method

At first, a populace of characterized size has been created through the uniform irregular cycle. Every part conveys an irregular stage of number from 1 to mxm (where mxm is the complete number of modules in the thought-about circuit); the wellness of individuals in the populace has gotten. The two guardians must first choose from the ongoing populace to acquire posterity. The determinations of the two guardians have been characterized through the wellness-based competition choice. In this cycle n part of the populace has thought about haphazardly, and among n part, the fittest part has considered as a parent. The upside of this cycle is the fair opportunity of that multitude of individuals having relative wellness higher than most fragile (n-1) individuals. Through a two-point get-over, posterity has been made, and the transformation administrator gives an irregular change starting with one module and then onto the next one. The course of getting over and change might make posterity infeasible. Recorrect administrator has finished the change from infeasible to the attainable arrangement. The functioning guideline of the recorrector administrator has displayed in the segment. The ongoing guardian and posterity populace structure a joined pool from where score-based competition choice interaction is applied to characterize the cutting-edge populace. In the score-based competition, every part faces a quantity of irregular adversaries and relies on their wellness and the score of the competition chosen. The last score of part is the all-out got to score against their rivals. Individuals with a higher worth competition score are considered part of the conceivable cutting-edge populace. Over this populace, elimination and broadening processes have been applied. The subtleties of the termination and enhancement process have been examined in the segment. The result of broadening the process has been considered the last cutting-edge populace. On relies upon the ending standards, either this cutting edge will become as the ongoing populace or fittest part has considered as definite arrangement.

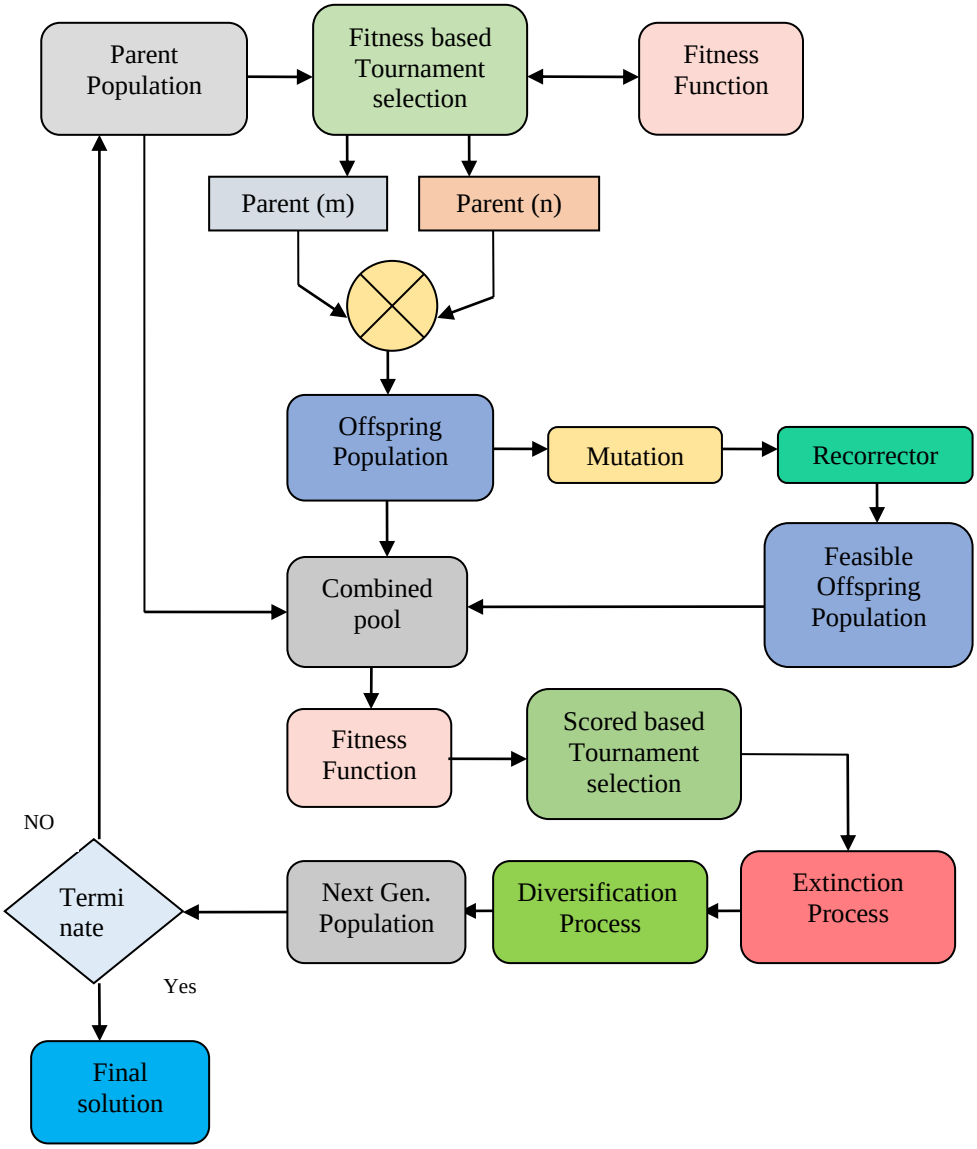

Fig. 3 Functional block diagram of the proposed method

7. Results

The below combinational circuit is considered to provide the experimental conformation of the proposed model. Over the circuit, the 2 different algorithms DYPSO, SGATS have been applied for 10 independent trials; the performance of each algorithm has been estimated in terms of the mean value of satisfying the percentage desired area of partitions, the need for the total number of interconnections among all available partitioning and objective function value.

The standard deviation of objective function values has also been estimated to obtain the algorithm's robustness. Along with numeric comparison among the two different algorithms, the convergence graphs have also been analyzed to get an idea about the speed of finding the solution. Circuit partitioning has also been presented for the best-achieved partition under different trials, assigning the same color for all modules that come under the same partition. The whole process has been simulated under a MATLAB environment. The circuit represented a graph where nodes represented the circuit module, and the connection between nodes was defined according to connections among the circuit modules.

The population for each algorithm has been maintained as 50, and the allowed number of iterations under a trial is 100. For DYPSO, inertia weight has reduced from 1.2 to 0.1, while cognition and constant social values have been kept at 0.5. The extrinsic factor of 0.72 also has been applied. In SGATS, the size of tournament has held at 10% of the population size. Two-point crossover operator has been applied, and mutation probability has been maintained as 0.1. The value maximum limit is 40% of the population size.

Table 1. Comparison of DYPSO and SGATS Algorithms

Circuit	Partitions Area constraint satisfaction (%)	Total no. of connections among partitions	Objective Function value
DYPSO	72.2	10.10	12.76 (0.31)
SGATS	72.2	10.00	12.66 (0.00)

Table 2. Three Level partitioning details of Fig.4.

1st partition	Partition Area	Total Area violation	No.of Connection between
P1 {1, 4, 6, } P2{2, 3, 5, 8} P3 {9, 10 }	4 4 2	2.6667	11.0000

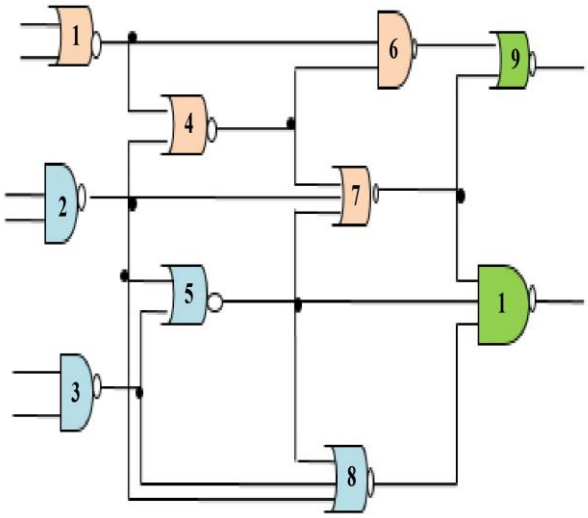

Fig. 4 Circuit Diagram Considered for partitioning

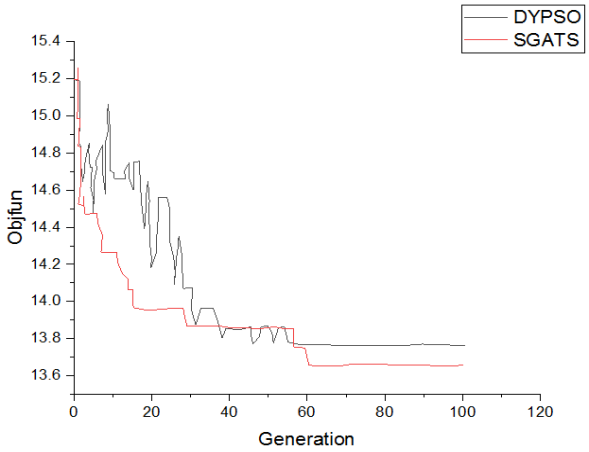

Fig. 5 Comparision graph of DYPSO and SGATS Algorithms

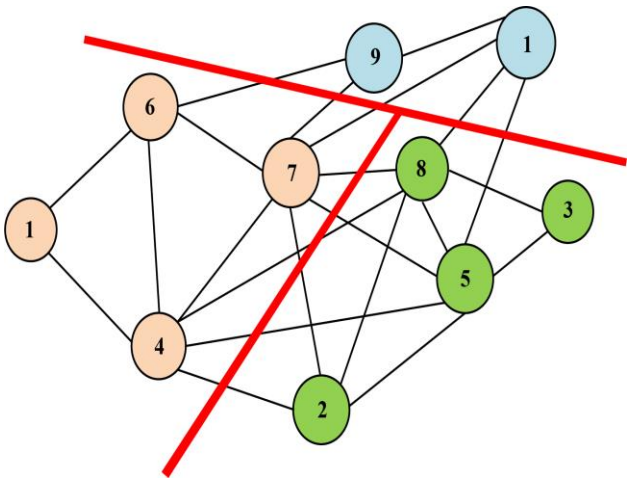

Fig. 6 Graph partitioning of the Circuit Diagram

For the circuit Fig.4, the three-level partitioning requirement has been resolved by two different algorithms and obtained mean performances over 10 independent trials have shown in Table.1&2. As there was a fraction area allocated for each partition which is impossible to achieve practically, there is an error in the demanded area

compared to the desired one. The performance of DYPSO was the poorest, and the SGATS algorithm converged early by the proposed method, as shown in Fig.5. and Fig.6.

8. Conclusion

The suggested method gives an idea for breaking the circuit into smaller sub-circuits to restrict the number of interconnects in the circuit, which leads to a reduction in cost and chip area. To achieve this, in this work, the division must be designed by applying a standard Genetic Algorithm so that the number of interconnects connecting the isolated groups is kept to a minimum, thereby shortening the overall length of the circuit interconnects.

References

[1] Y. Yodtean, and P. Chantngarm, "Hybrid Algorithms for Circuit Segmentation," *Proceedings of the IEEE Space Ten Conference*, vol. 4, 2004.

[2] G. F. Nan, and M. Q. Lee, "Two Novel Coding Ways for Circuit Segmentation Primarily Based Genetic Algorithms," *Proceedings of the Third IEEE International Conference on Machine Learning*, vol. 4, pp. 2182–2188, 2005.

[3] G. C. Sipakoulis, I. Karafyllidis, and A. Thanelakis, "Genetic Segmentation and Placement for VLSI Circuits," *Proceedings of The 6th IEEE International Conference on Electronics, Circuits and Systems*, vol. 3, pp. 1647-1650, 1999.

[4] C. J. Augeri, and H. H. Ali, "New Graph-Based Algorithms for Segmentation of VLSI Circuits," *2004 IEEE International Symposium on Circuits and Systems*, pp. V-521-V-524, 2004. *Crossref*, https://doi.org/10.1109/ISCAS.2004.1329055

[5] A. Cincotti, V. Cuttelo, and M. Pavone, "Graph Segmentation of Genetic Algorithms with OPDX," *Proceedings of the IEEE World Congress on process Intelligence*, pp. 402–406, 2002.

[6] Shanavas, Hameem, and R. K. Gnanamurthy, "Evolutionary Algorithmical Approach for VLSI Floorplanning Problem," *International Journal of Computer Theory and Engineering*, vol. 1, no. 4, pp. 461-464, 2009.

[7] N. Krasnogor, and J. Smith, "A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 5, pp. 474-488, 2005. *Crossref*, https://doi.org/10.1109/TEVC.2005.850260

[8] Jianhua Li, and L. Behjat, "A Connectivity Based Clustering Algorithm with Application to VLSI Circuit Partitioning," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 5, pp. 384-388, 2006. *Crossref*, https://doi.org/10.1109/TCSII.2005.862174

[9] N. Selvakkumaran, and G. Karypis, "Multiobjective Hypergraph-Partitioning Algorithms for Cut and Maximum Subdomain-Degree Minimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 3, pp. 504-517, 2006. *Crossref*, https://doi.org/10.1109/TCAD.2005.854637

[10] Roba khega, Kamal Mahmoud Afisa, and Mohammed Yassin Subaih, "Comparison Between the Performance of the Simulated Annealing and Genetic Algorithms in Physical Conductor Orientation within FPGA," *SSRG International Journal of VLSI & Signal Processing*, vol. 6, no. 3, pp. 14-17, 2019. *Crossref*, https://doi.org/10.14445/23942584/IJVSP-V6I3P104

[11] I. H. Shanavas, R. K. Gnanamurthy, and T. S. Thangaraj, "A Novel Approach to Find the Best Fit for VLSI Partitioning - Physical Design," *2010 International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 330-332, 2010. *Crossref*, https://doi.org/10.1109/ARTCom.2010.93.

[12] R. Bazylevych, and L. Bazylevych, "The Methodology and Algorithms for Solving the Very Large-Scale Physical Design Automation Problems: Partitioning, Packaging, Placement and Routing," *2013 2nd Mediterranean Conference on Embedded Computing*, pp. 1-2, 2013. *Crossref*, https://doi.org/10.1109/MECO.2013.6601386

[13] K. Khan et al., "A New Efficient Layer Assignment Algorithm for Partitioning in 3D VLSI Physical Design," *2013 1st International Conference on Emerging Trends and Applications in Computer Science*, pp. 203-207, 2013. *Crossref*, https://doi.org/10.1109/ICETACS.2013.6691423

[14] J. Hinay shelly, and B. Craige Shreen, "D flip flops for Linear Response Shift Register in CMOS technology," *SSRG International Journal of VLSI & Signal Processing*, vol. 4, no. 3, pp. 16-20, 2017. Crossref, https://doi.org/10.14445/23942584/IJVSP-V4I5P104

[15] B. Sinha et al., "Heuristics in Physical Design Partitioning: A Review," *2015 International Conference on Innovations in Information, Embedded and Communication Systems*, pp. 1-5, 2015. *Crossref*, https://doi.org/10.1109/ICIIECS.2015.7192900

[16] Sadiq M. Sait, Feras Chikh Oughali, and Mohammed Al-Asli, "Style Layer Assignment and Partitioning for 3d Integrated Circuits Mistreatment Tabu Search and Simulated Annealing," *Journal of Applied Analysis and Technology*, vol. 14, no. 1, pp. 67-76, 2016.

[17] P. Rajeswari, and S. T. Chandra, "A Survey on an Optimal Solution for VLSI Circuit Partitioning in physical design using DPSO & DFFA Algorithms," *2017 International Conference on Intelligent Sustainable Systems*, pp. 868-872, 2017. *Crossref*, https://doi.org/10.1109/ISS1.2017.8389301

[18] Sreelekshmi .S, and Pooja S. Mohan, "Area Efficient Architecture for TCAM using Hybrid Partitioned SRAM," *SSRG International Journal of Electronics and Communication Engineering*, vol. 2, no. 7, pp. 26-29, 2015. *Crossref*, https://doi.org/10.14445/23488549/IJECE-V2I7P109

[19] Robin Andre, Sebastian Schlag, and Christian Schulz, "Memetic Multilevel Hypergraph Partitioning," *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 347-354, 2018. *Crossref*, https://doi.org/10.1145/3205455.3205475

[20] Sumitha Manoj, and R. Surendiran, "Investigation of Duty Cycle Distortion in Clock Channels with Infinisim Clockedge Technology," *International Journal of Engineering Trends and Technology*, vol. 70, no. 4, pp.457-464, 2022. *Crossref*, https://doi.org/10.14445/22315381/IJETT-V70I4P238.

[21] X. Lei et al., "A New Multilevel Circuit Partitioning Algorithm Based on the Improved Kl Algorithm," *2019 IEEE 5th Intl Conference on Big Data Security on Cloud, IEEE International Conference on High Performance and Smart Computing, and IEEE International Conference on Intelligent Data and Security*, pp. 178-182, 2019. *Crossref*, https://doi.org/10.1109/BigDataSecurity-HPSC-IDS.2019.00041

[22] Shikha Arora et al., "Hybrid Algorithm PSO and SA in Achieving Partitioning Optimization for VLSI Applications," *International Journal of P2P Network Trends and Technology*, vol. 2, no. 1, pp. 1-4, 2012.

[23] Rajeswari, P, and S Theodore Chandra, "A Survey on an Optimal Solution for VLSI Circuit Partitioning in Physical Design using DPSO & DFFA algorithms," *2017 International Conference on Intelligent Sustainable Systems*, pp. 868-872, 2017. *Crossref*, https://doi.org/10.1109/ISS1.2017.8389301

[24] Rajeswari. P, Theodore S. Chandra, and Amith Kiran Kumar, "Synthesis of VLSI Structural Cell Partitioning Using Genetic Algorithm," *ICT Systems and Sustainability, Springer*, pp. 279-287, 2021. *Crossref*, https://doi.org/10.1007/978-981-15-8289-9_26

[25] S. Roy, and S. Banerjee, "An Efficient Genetic Algorithm Based Multi-Objective Optimization Technique for VLSI Circuit Partitioning with Reduced Power Consumption," *2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques*, pp. 741-745, 2021. *Crossref*, https://doi.org/10.1109/ICEECCOT52851.2021.9708020

[26] Peng, Shaojun et al., "A Discrete PSO for Partitioning in VLSI Circuit," *2009 International Conference on Computational Intelligence and Software Engineering*, pp. 1-4, 2009. *Crossref*, https://doi.org/10.1109/CISE.2009.5364339

[27] Pradip Kumar Sharma, and M. Kaur, "A Discrete FireFly Algorithm for VLSI Circuit Partitioning," *2014 International Conference on Electronics and Communication Systems*, pp. 1-4, 2014. *Crossref*, https://doi.org/10.1109/ECS.2014.6892764

[28] L. Lyu, and T. Yoshimura, "A Force Directed Partitioning Algorithm for 3D Floorplanning," *2017 IEEE 12th International Conference on ASIC*, pp. 718-721, 2017. *Crossref*, https://doi.org/10.1109/ASICON.2017.8252576

[29] Rajeswari P et al., "Associate Degree Investigation on Basic Ideas of Particle Swarm Optimisation Algorithmic Program for VLSI Design," *International Journal of Engineering and Techniques*, 2018.

[30] Ghasem Pasandi, and Massoud Pedram,"A Graph Partitioning Algorithmic Program with Application in Synthesizing Single Flux Quantum Logic Circuits," *arXiv.org,Xiv:1810.00134*, 2018. *Crossref*, https://doi.org/10.48550/arXiv.1810.00134