Grenze International Journal of Engineering and Technology, January Issue

Optimizing VLSI Layouts through Genetic Algorithm: A Path to Automated Physical Design

Dr. Meenakshi Gupta[1], Mr. Aryan Singh Chauhan[2], Dr. Niharika Thakur[3], Dr. Bhanu Pratap Chaudhary[4] and Mr. Harshil Aron[5]
[1-5]Electronic and Communication Engineering Manav Rachna University, Faridabad, India
Email: meenakshigupta@mru.edu.in, aryansinghchauhan000@gmail.com, niharika@mru.edu.in, bhanu@edu.in, harshilaron27@gmail.com

***Abstract*— This paper presents a Genetic Algorithm (GA)-based approach for optimizing VLSI layouts to advance automation in physical design. Traditional VLSI design techniques often face challenges with complex discrete optimization problems, resulting in inefficient layouts and extended design cycles. The proposed methodology utilizes a binary slicing-tree representation to model module placement and routing, integrated with hybrid pipelining and hardware/software co-design to enhance performance. Mutation and crossover operations are strategically applied to minimize wire length and layout area while improving routing efficiency. Experimental evaluations on industry-standard benchmarks demonstrate that the proposed GA-based layout optimization significantly enhances efficiency and throughput compared to conventional methods. The results affirm that Genetic Algorithms provide a promising pathway toward automated, intelligent, and high-performance VLSI physical design.**

***Keywords*— Mutation, Crossover operations, Optimization process Automated VLSI layout optimization, Physical design processes.**

I. INTRODUCTION

In the realm of optimization problems, genetic algorithms offer a unique approach by focusing not on the problems themselves [1] but on coded representations of potential solutions. Rather than directly navigating the solution space, genetic algorithms explore space of the encoded solutions [2]. For continual parameter optimization, like those encountered in various real-world scenarios, a common approach involves representing solutions as strings of genes, typically floating-point numbers. Each gene within these strings corresponds to a specific resource [3] of the solution vector. However, when it deals with discrete optimization problems, the landscape becomes more complex. Not every string represents a valid solution, and simply combining two individuals through crossover operations doesn't always yield viable offspring [4]. Repair mechanisms become necessary to ensure offspring correctness, complicating the correlation between parents and offspring [5]. to tackle real-world discrete optimization problems effectively, bespoke genetic algorithms must employ application- specific encoding schemes and intelligent operators. These operators generate offspring while adhering to problem- specific constraints, leveraging domain knowledge to produce high-quality solutions. Unlike in function optimization, where undesirable genes are easily identified, discrete optimization requires proactive strategies to avoid incorporating unfit genes into the population.

Grenze ID: 01.GIJET.12.1.8_3

This might involve employing hill-climbing techniques during the initial population construction or operator application. Furthermore, in contrast to identifying bad genes, recognizing the building blocks of an optimal solution is inherently challenging. Thus, genetic algorithm designers play an important role in providing a pool of potentially beneficial genes from which the algorithm can draw.

II. Literature Review

Genetic Algorithms (GAs) have proven effective for VLSI layout optimization, evolving from early implementations that utilized simple binary string encoding to more advanced methods like slicing trees and hierarchical representations [6], which have significantly improved results. GAs hold a promising future in VLSI layout generation, especially when hybrid approaches are employed, combining GAs with other techniques for enhanced performance [7]. Moreover, multi-objective GAs have been developed to manage conflicting objectives, and recent research has focused on addressing constraints and variability, further advancing the capabilities of GAs in VLSI design [8].

III. Methodology

- Layout Generation: Generating layouts for modern VLSI microchips, each containing millions of transistors, is a complex actions involving several sequential steps [9]. These steps include system specification, physical design, logic design, circuit design, and functional design. The physical design phase transforms the circuit description into the chip layout required for fabrication [10]. This layout entails geometric descriptions of circuit components, interconnection routes, and pads for chip I/O connections [11]. The aim of layout or design generation is to reduce the chip's area and produce efficient routing with short wire lengths [12], especially for critical connections. Given its complexity, physical design is typically broken down into consecutive sub-steps. Firstly, the circuit is partitioned into modules (macro cells), which are then placed in chip during floor planning. Ample space must be reserved during placement to accommodate all interconnections. The procedure phase involves connecting pins on module borders, which occurs in two stages: global routing to determine initial routes and detailed routing to compute exact paths between modules. The final step is layout compaction, where the layout is compressed to reduce total area.
- A Genetic Algorithm layout Generation: This proposed for the generation of layouts, wherein a layout encompasses module positions, flexible modules, and interconnection routes [13]. This intricate phenotype cannot be directly shown by a simple genetic string of elementary data- types. Instead, binary slicing tree is advocated as a viable method to characterize module placement. Within this framework, each leaf represents a module, while inner nodes denote meta- blocks [14], delineating arrangements and routing information for sets of modules. By delineating arrangements and routing information at this meta-block level, the binary slicing tree affords a more holistic perspective on layout generation [15].

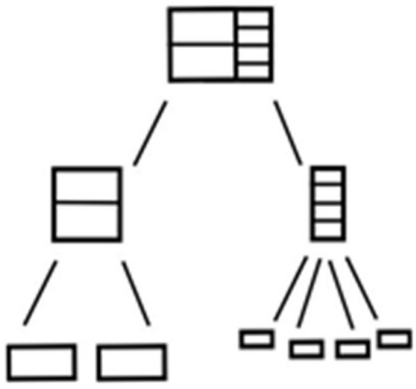

Fig. 1. Representation of Genotype

The inclusion of design functionality for all the nodes of tree facilitates the consideration of multiple implementations for flexible blocks, thereby ensuring the representation of diverse layouts within a single individual. Through the mixed of modules into meta-blocks, the placement and turning of the composite blocks are predetermined, leading to the aggregation of their respective shape- functions. Notably, the proliferation of execution for meta-blocks at top levels of the tree is mitigated by redundancy reduction techniques, resulting in

only two distinct implementations for resulting meta- blocks. Furthermore, a high bound for routing space within meta-blocks is figured and incorporated into their shape- functions, enhancing the efficiency of the improved process. Moreover, the process of combining modules into meta-blocks introduces an additional layer of complexity and optimization potential. By predefining the positioning and rotation of composite blocks within meta-blocks, the algorithm can leverage prior knowledge and heuristics to streamline the search for optimal layouts. This predetermined structuring not only expedites the improved process but also enables the aggregation of shape-functions, consolidating the representation of complex layout configurations into more manageable and interpretable units.

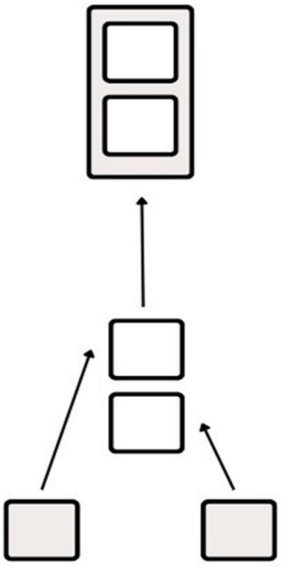

Fig.2. Combining two flexible blocks into a meta block and incorporating routing space

- Hybrid creation of individuals: The slicing-tree for individual within the hybrid creation process, we start by building a slicing-tree from the ground up. We employ a special technique called iterated matching to form building blocks that define top-notch partial layouts [16]. Imagine a complete chart where each node represents a block, and the edges are weighted based on the quality of combining two adjacent blocks into a meta- block. Through a series of iterations, we match nodes to form pairs, focusing on maximizing the sum of edge weights, which indicates the grade of the resulting meta block in a terms of shared connections [17]. By pairing highly inter linked blocks together, we effectively minimize wire length and overall layout area [18]. To ensure diversity among individuals, we introduce randomness in computing edge mass for some of the relevant graphs

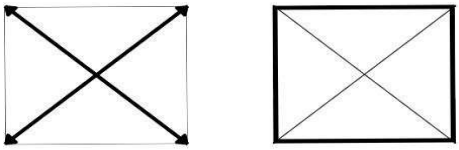

Fig. 3. Showcasing the three potential matches and highlighting the maximum mass match

- Multiple Genes : In the intricate world of design, where the canvas spans from website layouts to architectural blueprints, the introduction of adaptable blocks adds layers of complexity, veiling the path to the perfect arrangement in a shroud of ambiguity. This uncertainty, though daunting, invites a methodical dance—an intricate choreography wherein every conceivable permutation of these versatile elements is meticulously cataloged and preserved. Imagine this archive as a grand library, each page chronicling a different iteration, a unique dance of form and function. Every twist and turn, every subtle shift, captured and preserved for posterity. It's a testament to the human spirit of exploration and innovation, where each recorded configuration represents not just a design, but a journey—a story of creativity and discovery. And

in this vast repository of design possibilities, lies a treasure trove of insights and revelations. Like a wise sage, it whispers secrets of balance and harmony, guiding designers through the labyrinth of choices towards the elusive ideal. With each page turned, designers draw upon the collective wisdom of past endeavors, finding inspiration in the trials and triumphs of those who came before. But it's not just about finding the perfect layout—it's about the journey, the exploration, the joy of discovery. Every design permutation is a celebration of human ingenuity, a testament to our ability to envision and create. And as designers sift through this wealth of alternatives, they are not just finding a solution—they are weaving a tapestry of innovation, stitching together the threads of creativity and craftsmanship into a masterpiece of design.

- Optimization Process: These It begins after creating initial individuals with various components. The genetic algorithm tweaks individuals through mutation and combines building blocks through crossover. Mutation involves changing block arrangements and the shape of slicing tree. Crossover merges two individuals to create offspring, with new elements added to conclude the layout. It's crucial to ensure that the genetic working can lead to reaching the best solution regardless of the starting individuals. In simpler terms, the algorithm needs to reach the best outcome from any starting point by mutating or combining elements effectively. Mutation, a fundamental genetic operator, acts as a catalyst for change within individuals, instigating alterations in both the spatial arrangements of blocks and the overarching shape of the slicing tree. This process imbues the algorithm with the ability to explore new avenues and traverse uncharted territories within the design space. Through strategic modifications to block arrangements and tree structures, mutation injects novelty and variation into the population, thereby fostering the exploration of alternative layout configurations and design paradigms. In parallel, crossover emerges as a synergistic force, orchestrating the fusion of genetic components from two parent individually to engender offspring endowed with the best traits of both progenitors. This process of genetic recombination entails the merging of building blocks and the incorporation of new elements to complete the layout puzzle. By leveraging the collective wisdom encoded within the parent individuals, crossover engenders offspring that inherit favorable characteristics while also introducing novel combinations and permutations, thus enriching the genetic diversity of the population. Certain to the efficacy of the optimization process is the imperative to ensure that genetic operators exert their transformative influence with utmost efficacy and discernment, irrespective of the initial configurations of individuals. In essence, the algorithm must possess the innate capacity to navigate the design space with dexterity and resilience, adeptly steering towards the best possible outcomes through judicious application of mutation and crossover. This necessitates a delicate balance between exploration and exploitation, wherein the algorithm seeks to harness the creative potential of mutation to explore new design territories while also capitalizing on the consolidating power of crossover to exploit promising avenues of optimization.

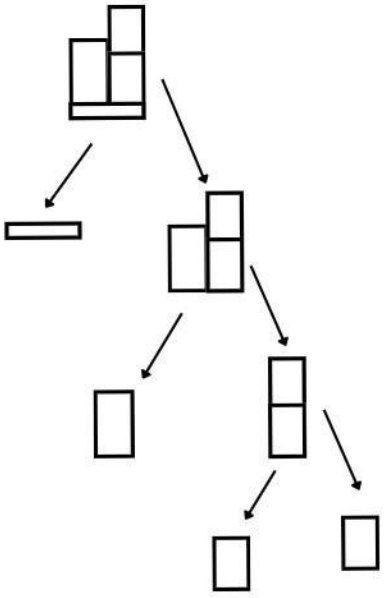

Fig.4. Mutations occur through block exchanges (top) and alterations to the slicing tree structure (bottom)

IV. CONCLUSION

This In conclusion, the proposed genetic algorithm for VLSI layout generation represents a robust and sophisticated approach meticulously designed to confront the multifaceted challenges inherent in optimizing the intricate layouts demanded by contemporary microchips. By harnessing the ability of a binary slicing tree representation, this algorithm adeptly encapsulates the complex phenotype of module placement and interconnection routes, thereby furnishing a remarkably adaptable framework capable of accommodating a myriad of layout configurations within a single individual. The hybrid creation process, ingeniously integrating iterated matching techniques, stands as a testament to the ingenuity underlying this approach, ensuring the generation of layouts of unparalleled quality characterized by minimized wire length and optimized overall area utilization Through the judicious application of mutation and crossover operations, the optimization process unfolds iteratively [19], steadily refining individuals towards the pursuit of the optimal solution, irrespective of their initial configurations [21]. In essence, this approach not only represent the inherent potential of genetic algorithms in addressing discrete optimization problems but also serve as a beacon illuminating a promising pathway towards the automation of VLSI layout optimization. By contribute to the on-going advancements in physical design processes for microchip fabrication, it heralds a new era of efficiency and efficacy in the realm of semiconductor technology, poised to revolutionize the landscape of microchip development and deployment.

ACKNOWLEDGMENT

The I would like to express my sincere gratitude to Professor/Dr. Meenaskshi Gupta, my supervisor, for their invaluable guidance, encouragement, and insightful suggestions throughout the course of this research. Their continuous support and expertise were essential in overcoming various challenges. I also extend my appreciation to Manav Rachna University for providing the necessary resources and a conducive environment for conducting this research

REFERENCES

[1] R.Vanalakshmi,S.Maragathasundari,K.S. Dhanalakshmi, Queuing system in VLSI physical design, Mathematics and Computers in Simulation, 2022.
[2] Ping Yang, Haiying Yang, Wei Qiu, Shuting Wang, Chuanquan Li, Optimal approach on net routing for
[3] Deepthi Amuru, Andleeb Zahra, Harsha V. Vudumula, Pavan K. Cherupally, Sushanth R. Gurram, Amir Ahmad, Zia Abbas, AI/ML algorithms and applications in VLSI design and technology, Integration, 2023.
[4] C.P. Ravikumar, Solving VLSI physical design problems on a vector machine, Computer-Aided Design, 1993.
[5] Nithiyanantham Janakiraman, Palanisamy Nirmal Kumar, Multi- objective module partitioning design for dynamic and partial reconfigurable system-on-chip using genetic algorithm, Journal of Systems Architecture, 2014.
[6] Rubiyah Yusof, Marzuki Khalid, Gan Teck Hui, Syafawati Md Yusof, Mohd Fauzi Othman, Solving job shop scheduling problem using a hybrid parallel micro genetic algorithm, Applied Soft Computing, 2011.
[7] Z. Zhu, D.J. Mulvaney, V.A. Chouliaras, Hardware implementation of a novel genetic algorithm, Neurocomputing, 2007.
[8] Lin Li, Yici Cai, Qiang Zhou, A survey on machine learning-based routing for VLSI physical design, Integration, 2022.
[9] Dimitrios Stathis, Panagiotis Chaourani, Syed M.A.H. Jafri, Ahmed Hemani, Clock tree generation by abutment in synchoros VLSI design, Microprocessors and Microsystems, 2023.
[10] Uwe Schulz, Hierarchical physical design system for VLSI-/370 microprocessor, Microprocessing and Microprogramming, 1988.
[11] Lei Xie, Hao Cai, Chao Wang, Jun Yang, Towards an automated design flow for memristor based VLSI circuits, Integration, 2020.
[12] J. Snehalatha, Chintaiah Nannepaga, Sivasubramanyam Medasani, VLSI physical design based on Tabu- ant colonies modeling, Applied Soft Computing, 2014.
[13] Bujjibabu Nannepaga, Katepogu Rajkumar, Conditional spatial transition reduction data encoding technique for VLSI interconnects, e- Prime - Advances in Electrical Engineering, Electronics and Energy, 2024.
[14] Chávez-Bracamontes Ramón, García- López Reyna Itzel, Gurrola-Navarro Marco Antonio, Bandala-Sánchez Manuel, VLSI Design with Alliance Free CAD Tools: an Implementation Example, Ingeniería, Investigación y Tecnología, 2015.
[15] Akash Agrawal, Prosenjit Gupta, Incremental analysis of large VLSI Layouts, Integration, 2009.
[16] Sriparna Saha, Susmita Sur-Kolay, Parthasarathi Dasgupta, Sanghamitra Bandyopadhyay, MAkE:Multiobjective algorithm fork-way equipartitioning of a point set,Applied Soft Computing, 2009.
[17] Steve T.W. Lai, Evangeline F.Y. Young, Chris C.N. Chu, Handling routability in floorplan design with twin binary trees, Integration, Volume 42, Issue 4, 2009.

[18] N. Vijaya Bala, Dr. R. Shanmughalakshmi, High speed VLSI architectures for DWT in biometric image compression: A study, Procedia Computer Science, Volume 2, 2010.
[19] Lucanus J. Simonson, Industrial strength polygon clipping: A novel algorithm with applications in VLSI CAD, Computer-Aided Design, Volume 42, Issue 12,2010.
[20] Alexandre Linhares, Horacio Hideki Yanasse, Connections between cutting-pattern sequencing, VLSI design, and flexible machines, Computers & Operations Research, Volume 29, Issue 12,2002.
[21] Alberto Isidori, Ruth Bars, Jean- Michel Dion, Sebastian Engell, Torkel Glad, DESIGN METHODS, IFAC Proceedings Volumes, Volume 35, Issue 1, 2002
[22] Tony Kenyon, 5 - Physical topology design, Editor(s): Tony Kenyon, High Performance Data Network Design, Butterworth- Heinemann, 2002.