

# Optimization of Vlsi Physical Design Bisectional Placement with Quantum Approximation Algorithm

<sup>1</sup>Dr. Pushpalatha Pondreti, <sup>2</sup>Punnam Omkaram

<sup>1</sup>Assistant Professor, Department of ECE,  
University College Of Engineering, Kakinada (A)  
Jawaharlal Nehru Technological University, Kakinada

<sup>2</sup>IEEE student chair, ECE, M.Tech (VLSI & ES),  
University College Of Engineering, Kakinada (A)  
Jawaharlal Nehru Technological University, Kakinada

**Abstract:** - The growing complexity of VLSI circuits has rendered traditional heuristic methods of placement less effective, since this process itself is an NP-hard problem and is a major scalability bottleneck. Placement of a large circuit with standard cells and smaller modules with few interconnections is crucial for efficient physical design, particularly with new paradigms like chiplet-based designs. This paper presents a new quantum algorithmic approach to address the shortcomings of conventional methods. By utilizing the quantum computer's native parallelism and distinct optimization potential, the new method promises significant improvements in both the efficiency and effectiveness of placement outcomes. Early results indicate that quantum algorithms have the potential to provide faster and more optimal placement, paving the way for more effective design flows and making the practical implementation of more complex integrated systems a reality. We encode the task of partitioning  $n$  standard cells into two equal halves as a Quadratic Unconstrained Binary Optimization (QUBO), with binary variables indicating each cell's assigned region. Mapping each binary variable to an Ising spin transforms the QUBO into an  $n$ -qubit Hamiltonian whose ground state represents the optimal placement. We estimate the expected energy  $\langle H \rangle$  via Pauli-term measurements on the quantum device and employ the "Trust-Constr" classical optimizer to adjust QAOA parameters ( $\gamma, \beta$ ). These results establish a foundation for quantum-classical hybrid frameworks for VLSI design optimization. Proposed approach opens new avenues for integrating quantum optimization into next-generation VLSI design flows, potentially transforming computational paradigms in VLSI design.

**Keywords:** *Quantum computing, hybrid quantum computing, quantum approximate optimization algorithm (QAOA), quantum optimization, variational quantum eigensolver (VQE), variational quantum computing, VLSI design optimization, bisectional placement, VLSI physical design placement optimization.*

## 1. INTRODUCTION

VLSI (Very Large-Scale Integration) placement is one of the most computationally demanding issues in electrical design automation, which is categorized as an NP-hard problem that severely restricts the effectiveness of traditional computing techniques. The intricacy of breaking down these enormous systems into smaller, more manageable subsystems has increased dramatically as contemporary integrated circuits contain millions to billions of transistors with complicated interconnections. A problem is considered NP-hard in computational complexity theory if it can be reduced to every other problem in NP (nondeterministic polynomial time) in polynomial time. This classification indicates that as the size of the problem rises, finding the best solutions for VLSI circuit placement involves exponential time complexity on classical computers. This computational intractability is best illustrated by the binary quadratic optimization formulation of circuit bisectional placement, in which the search

space increases exponentially with the number of standard cells in design. The combinatorial explosion of potential placement solutions is the core problem. An exhaustive search is computationally impractical for practical circuit sizes because the number of methods to divide a circuit with  $n$  components for placement into  $k$  balanced subsets increases factorially. Instead of using precise optimization techniques, designers are forced to rely on heuristic and approximation algorithms due to the exponential increase in solution space complexity. Large-scale VLSI designs provide major computing bottlenecks for classical partitioning techniques. The algorithms Fiduccia-Mattheyses and Kernighan-Lin, which iteratively enhance placement by moving single cell or switching cell pairs to reduce costs locally, simulated annealing, which works to find better placement by escaping local minima using probabilistic jumps, despite using populations and recombination operators to explore different regions of the solution space, genetic and evolutionary algorithms are still unable to overcome the optimization landscape's inherent NP-hardness. On moderately sized problems, these traditional approaches are effective and surprisingly strong, but their solutions are still essentially approximations. Particularly as solution landscapes grow rougher due to increased netlist complexity and non-uniformity, they could become stuck in local optima. Furthermore, even state-of-the-art heuristics are hampered by scaling bottlenecks, lengthy runtimes, and the difficulty of ensuring solution quality as system designs scale into the millions (or billions) of cells, as in modern high-end AI accelerators, networking chips, or chiplet assemblies. In practice, this means that suboptimal placement of macros and standard cells are frequently found, which can result in suboptimal routing complexity, higher inter-module wiring, and possibly decreased chip space or performance.

The need for radically new approaches stems from the incapacity of traditional computers and algorithms to handle the combinatorial explosion in options for massive partitions. Because VLSI circuit partitioning is fundamentally placement of standard cells and macros, electronic design automation techniques must undergo radical changes. The exponentially huge solution spaces needed for optimal circuit organization are inherently difficult for classical computers to explore; nevertheless, QAOA is a promising quantum computing technique that may be able to get around these computational obstacles thanks to quantum mechanical advantages. The optimization gap between theoretical optimal solutions and realistically feasible outcomes grows considerably when circuit complexity continues on its exponential development trajectory while traditional computational capabilities scale polynomials. With quantum-enhanced optimization skills created especially for the combinatorial difficulties present in circuit partitioning problems, QAOA presents a way forward that may close this gap. The limits of existing design automation tools are highlighted by the intractable nature of VLSI placement with classical computers, which stems from its NP-hardness. With schemes like the Quantum Approximate Optimization Algorithm (QAOA), which attempts to take advantage of quantum parallelism and superposition to effectively sample from the solution space of hard combinatorial problems, recent developments in quantum algorithms provide an exciting path forward. By using quantum processes that are not accessible by conventional, deterministic hardware, quantum techniques have the potential to access significantly better solutions or locate them more quickly than classical heuristics, which enhance bisectional placement quality through approximate or randomized local search. The present issues that drive the development of QAOA and related quantum algorithms are evident, even though their specifics and potential are covered elsewhere. Research into radically new computational paradigms for VLSI design automation is necessary due to the persistent NP-hardness of placement for classical computers, the constantly expanding scale of integrated designs, and the implications for silicon manufacturability and system performance. An emerging field that has the potential to drastically alter the design and optimization of complex VLSI systems is the incorporation of quantum optimization algorithms, such as QAOA, into current electronic design automation workflows. This could lead to the creation of integrated circuits with higher performance and efficiency than are currently possible with only classical computational methods.

## 2. LITERATURE REVIEW

### 2.1 Classical Computing

Fiduccia and Mattheyses (1982) present a linear-time heuristic for balanced two-way min-cut network partitioning that iteratively moves single cells between blocks, using “gain” metrics and bucket-sorted lists to choose the best move in  $O(1)$  time and update affected neighbor gains in  $O(1)$  time per pin, yielding  $O(P)$  time per pass. By

locking moved cells and only updating critical nets, their algorithm typically converges in a few passes, making it practical for large networks with thousands of pins. This approach underpins many modern partitioners and placement tools seeking fast near-optimal cuts under size constraints [1]. Kernighan and Lin (1970) introduce a fast heuristic for balanced 2-way graph partitioning that iteratively swaps node pairs to maximize cost reduction, achieving locally optimal cuts in  $O(n^2 \log n)$  time per pass by maintaining and updating gain values. Their method rapidly identifies high-gain exchanges without exhaustive search, reliably finding near-optimal partitions for large graphs with only a few passes. Extensive experiments demonstrate its practical efficacy in VLSI circuit board and memory paging applications, forming the basis for many subsequent partitioning algorithms [2]. Breuer (1977) introduces a paradigm shift from classical distance-based placement objectives to "min-cut" approaches that minimize the number of signals crossing partition lines, proving that minimizing total cut values across canonical cut lines is equivalent to minimizing half-perimeter wirelength but offering superior routability predictions. The paper presents two algorithmic frameworks Quadrature (breadth-first with alternating vertical/horizontal bisections) and Slice/Bisection (depth-first with horizontal slicing followed by vertical bisection) that recursively partition the placement region using generalized Kernighan-Lin heuristics to optimally assign moveable elements while respecting fixed element constraints. Experimental results on 5"×5" PC cards demonstrate 15-25% improvement in half-perimeter length and substantial reduction in routing failures compared to manual placement, establishing min-cut placement as a foundational approach for VLSI physical design automation [17]. Wu et al. (2003) introduce a hybrid graph-logic partitioning framework that applies efficient logic-rewiring perturbations (RAMBO, GBAW) to top high-quality hypergraph partitions (hMetis), yielding an additional 11-12% cut-size reduction for 2 and 4-way MCNC benchmarks with minimal area (>0.3%) and runtime overhead. Their GBAW technique accelerates alternative-wire matching via subgraph pattern extraction, achieving comparable or superior improvements to ATPG-based methods at a fraction of computational cost. These results demonstrate that integrating lightweight logic perturbations into pure graph-partitioning tools can escape local minima and substantially improve multiway partitioning quality and can placement of cells will take reduced area [3].

Zhu et al. (2015) reformulate the VLSI global placement problem using an exact norm wirelength model that precisely calculates half-perimeter wirelength without approximation errors, combined with exact overlapping area calculations between cells and bins rather than smoothed density functions used in traditional analytical placers. They develop a non-smooth optimization framework solved by a novel Polak-Ribière conjugate sub gradient method with proven local convergence properties, incorporating adaptive step size control and cautious penalty parameter strategies within a multilevel clustering approach using a modified best-choice algorithm tailored to the  $L_1$ -norm model. Experimental results on ISPD 2005 and 2006 benchmarks demonstrate that their approach achieves 0.4-14.6% shorter average wirelength compared to state-of-the-art placers including FastPlace3.0, mPL6, NTUplace3, and ePlace, though with 2-3× longer runtime due to the slower convergence characteristics of sub gradient methods [21]. Cheng and Wei (1991) address the instability of Kernighan Lin based two-way partitioning by introducing a top-down, ratio cut clustering scheme that recursively breaks the network into highly connected groups without predefined subset sizes. These groups are then contracted and repeatedly partitioned via the Fiduccia Mattheyses algorithm to meet specified size constraints, before a final global refinement on the original network. This "stable" approach drastically reduces variance in cut weight standard deviation drops from 24 pins to 2 pins across trials while yielding average cut reductions of 48% over pure FM on fourteen VLSI benchmarks. By balancing cut weight against CPU time through the choice of clustering granularity for placement, the method achieves robust, high-quality partitions in linear-time per iteration. This work demonstrates that integrating ratio-cut clustering with efficient local refinement overcomes local-minimum trapping and delivers consistently superior two-way partitions for large circuits [4].

Sanchis (1989) generalizes the Fiduccia Mattheyses iterative-improvement algorithm to directly partition a network into  $b$  disjoint blocks by defining multi-level gains for moving cells between any two blocks and selecting cell moves lexicographically by their gain vectors. The uniform algorithm maintains specialized bucket structures and a heap of legal move directions to achieve per-pass complexity  $O(mb(\log b + p))$ , where  $m$  is the number of cells,  $b$  the number of blocks, and  $p$  the maximum cell degree. Experiments on random networks show that optimal gain levels increase with the number of blocks, net size, and cell degree but are largely independent

of network size, and that the uniform method outperforms hierarchical partitioning as more computation time is allowed [5]. Sanchis (1993) adapts two-way network partitioning algorithm to support three cost functions unit cost,  $k-1$  cost, and  $k(k-1)/2$  cost by redesigning gain-update procedures to preserve polynomial time complexity  $O(m(q+bl)(\log b+pl))$  for cost<sub>2</sub> and  $O(m(q+bl+b^2)(\log b+pl))$  for cost<sub>3</sub> with runtime optimizations reducing moves by up to 85%. Experimental evaluations on random networks (100–400 cells, varying net sizes) show that higher “levels” yield diminishing cut-size improvements for cost-2 and cost-3 compared to cost-1, and that uniform partitioning consistently outperforms hierarchical strategies across all cost functions. Despite modest gains beyond level 2, the cost-3 algorithm benefits slightly more from additional levels than cost-2, reflecting the deeper block-interaction captured by its cost definition [6].

Casotto et al. (1987) present a pioneering parallel implementation of simulated annealing for macro-cell placement that partitions cells among processors to eliminate synchronization overhead, achieving over 80% processor utilization on up to eight processors of the Sequent Balance 8000 shared-memory multiprocessor. Their approach allows processors to compute move costs using the most recent available information, introducing controlled errors that are large at high temperatures but converge to zero as temperature decreases, while a clustering strategy based on minimizing moment of inertia dynamically reassigns cell ownership to reduce overlap errors at low temperatures. Experimental results on circuits with up to 122 macro-cells demonstrate approximately 6× speedup with eight processors while maintaining solution quality comparable to sequential simulated annealing, establishing early feasibility of parallel optimization for VLSI physical design [18]. Cong, Hagen, and Kahng (1992) introduce a novel dual approach to VLSI circuit partitioning by transforming the netlist hypergraph into its intersection graph representation, where vertices represent signal nets rather than modules, and edges connect nets sharing common modules, resulting in significantly sparser adjacency matrices that enable faster eigenvector computations. Their IG-Match algorithm combines spectral methods with bipartite graph matching to partition nets first using eigenvector-based ordering, then optimally assigns shared modules through maximum independent set computations in bipartite graphs, achieving theoretical guarantees that the number of cut nets never exceeds the maximum matching size. Experimental results on MCNC benchmarks demonstrate the method's superiority with an average 28.8% improvement over Wei-Cheng's RCut1.0 algorithm and 6% improvement over previous intersection graph methods, while maintaining competitive runtime performance due to the inherent sparsity of the intersection graph representation [7].

Wu and Chu (2015) address the emerging challenge of detailed placement for VLSI designs containing mixed single-row and double-row height standard cells, which arise in complex synchronous designs and asynchronous circuits where 77% of cells on average require double-row height for handshaking signal generation. They propose a hybrid approach that transforms mixed-height designs into uniform-height layouts through strategic cell pairing (formulated as maximum weighted matching based on connectivity, area penalty, and displacement costs) in high-density regions and cell expansion in low-density areas, followed by conventional detailed placement and refinement. Experimental results on asynchronous and modified ISPD benchmarks demonstrate 3-14.8% wirelength improvement over alternative methods (direct expansion or macro treatment), with superior robustness across varying ratios of double-row height cells while maintaining zero overlap legalization [19]. Yeh, Cheng, and Lin (1995) experimentally evaluate simulated annealing (SA), Kernighan–Lin/Fiduccia Mattheyses (KL/FM), and a two-level primal dual (PD) iterative-improvement algorithm on random, geometric, and real VLSI circuits, showing that while KL/FM and SA perform comparably on random graphs and KL/FM leads on geometric graphs, PD significantly outperforms both on practical circuit benchmarks when given equivalent CPU time. Their results reveal that at least 500 FM runs are necessary to approximate its true performance distribution, and that disabling PD's dual step yields nearly identical cut quality with substantially reduced runtime, making the simplified PD variant the preferred choice for two-way partitioning in real-circuit applications. These findings underscore the importance of extensive repetitions and algorithmic tuning when benchmarking partitioning heuristics on realistic problem instances this will make placement easier to find the optimal location for placement of standard cells [8].

Dutt and Deng (2000) introduce PROP, a probabilistic-gain partitioner that computes node moves based on conditional probabilities of future cut-set removals, and SHRINK-PROP, which further magnifies the impact of newly “perturbed” nets, yielding 30–37% smaller cuts than Fiduccia Mattheyses and 27–34% over Krishnamurthy’s lookahead method on ACM/SIGDA benchmarks. Both methods also outperform recent multilevel and spectral partitioners (e.g., EIG1, WINDOW, MELO, PARABOLI, GFM, GMetis) by 4.5–67%, while maintaining competitive runtime, and SHRINK-PROP’s results approach hMetis’s quality within 2.5% on ISPD-98 circuits despite being a flat partitioner. These findings demonstrate that probability-based gain computations can more accurately capture global and future implications of node moves, significantly improving VLSI two-way min-cut partitioning [9]. Chan, Schlag, and Zien (1994) extend spectral ratio-cut partitioning from two-way to simultaneous  $k$ -way partitioning by generalizing the 2-way ratio-cut cost metric to  $k$  partitions and proving that the sum of the  $k$  smallest Laplacian eigenvalues lower-bounds this  $k$ -way ratio-cut cost. They show that embedding graph vertices into the  $k$ -dimensional subspace spanned by these eigenvectors, followed by an efficient cosine-based clustering heuristic, yields high-quality partitions in  $O(nk^2 + nk \log n)$  time and  $O(nk)$  space. Experimental results on standard benchmarks demonstrate superior cuts (up to 40% improvement) and faster runtimes compared to recursive BI-partitioning and prior spectral methods [10].

Pu et al. (2025) introduce IncreMacro, an incremental macro placement refinement framework that addresses the limitation of analytical placers placing macros in chip centers, which creates routing blockages and degrades timing performance, by iteratively employing KD-tree-based macro diagnosis to identify poorly placed macros, gradient-based shifting to move them peripherally, constraint-graph-based linear programming for overlap-free legalization, and diffusion-based cell migration for accurate wirelength estimation. The approach maintains two critical requirements: pushing macros toward chip boundaries while preserving original relative positional relationships established by analytical placers, thus eliminating central blockages without compromising wirelength optimization achieved by the placement prototype. Experimental results on seven RISC-V circuits and four TILOS benchmarks at 7nm technology demonstrate that integrating IncreMacro into AutoDMP and DREAMPlace 4.0 reduces routed wirelength by 14.9–15.1%, improves worst negative slack by 82.6–99.9%, and reduces total power consumption by 4.3–4.4% while maintaining 41.5% of macros unmoved to preserve spacing for standard cell placement and routing [20].

## 2.2 Quantum Computing

Krauss et al. demonstrate how to map the classic maximum-flow problem onto a quantum annealer by translating edge capacities into binary or unit-capacity variables and enforcing flow conservation through quadratic penalty terms. Three formulations balance qubit count and graph complexity: one clones edges into many unit-capacity links, one uses a compact binary encoding of flows, and one exploits the max-flow duality to require just one qubit per vertex. Experiments on D-Wave hardware confirm these formulations can find exact flows for small graphs, though current qubit connectivity and embedding overhead limit scalability [11]. Jahn et al. (2022) apply graph-theoretic minimum-cut algorithms to optimally bisectional MMC control systems into upper (system-relevant) and lower (hardware-relevant) levels, assigning edge weights to capture parameterization- vs. implementation-related relevance and minimizing inter-level dependencies. By modelling each control function as a node and its influence on system or hardware via weighted edges to “system relevance” and “hardware relevance” super-nodes, they achieve partitions that reduce required signal interfaces and integration effort in multi-vendor HVDC setups. Case studies on a 3-terminal MMC demonstrate that strategic relevance assignments and minimal cuts yield far lower integration costs than conventional reference architecture splits [12].

Shafique et al. (2024) provide a structured tutorial on quantum computing principles qubits, superposition, entanglement, interference, and noise alongside key building blocks such as single- and multi-qubit gates (Pauli, Hadamard, CNOT, Toffoli) and measurement processes. They survey foundational algorithms (Deutsch–Jozsa, Bernstein–Vazirani, Simon’s, Shor’s, Grover’s) and outline the NISQ era’s hybrid quantum–classical machine learning frameworks, emphasizing parameterized quantum circuits and error mitigation strategies. Finally, they discuss deployment requirements for superconducting hardware (cryogenics, control electronics) and explore quantum applications in cryptography, optimization, chemistry, finance, and energy [13]. Liu et al. (2022)



introduce Layer VQE (L-VQE), an adaptive hybrid quantum-classical algorithm that grows a hardware-efficient ansatz layer by layer adding Ry rotations and neighbouring CNOTs before convergence to solve k-community detection via a novel qubit-frugal modularity Hamiltonian on up to 40 qubits, demonstrating superior approximation ratios and noise resilience compared with fixed-ansatz VQE and QAOA. Their large-scale MPS simulations reveal that L-VQE avoids barren plateaus and maintains performance under finite sampling and realistic trapped-ion noise. By systematically expanding expressivity while preserving trainability, L-VQE presents a practical pathway for combinatorial optimization on NISQ devices [14].

Ueno et al. (2024) identify inter-temperature bandwidth between room-temperature controllers and cryogenic QPU as a key bottleneck for superconducting QAOA machines and propose a cryogenic SFQ-based counter architecture that aggregates measurement results over T trials, reducing required bandwidth from  $O(N)$  to  $O(1)$  with only  $O(\log N)$  bit counters. They model QAOA communication to show that classical read-out of N qubits per trial dominates bandwidth and demonstrate through analytic and SFQ circuit evaluations that counter-based buffering exponentially cuts cable heat inflow and peripheral power use with minimal execution overhead. This systems-level, algorithm-aware approach advances scalable NISQ device design by mitigating cable-induced thermal and power constraints [15]. Wu et al. pioneer a hybrid classical-quantum sampling framework for Approximate Query Processing (AQP) that identifies “rare” long-tail groups via classical stratified sampling and then applies quantum amplitude amplification implemented with QRAM and diffusion gates to boost their sampling probabilities without prior data preprocessing. By modelling tuple indices in superposition, using a composite Oracle for predicates and group membership, and optimizing amplification counts to minimize expected QRAM access costs, their algorithm achieves balanced sampling across skewed groups and yields higher accuracy at equivalent sampling costs compared to uniform or stratified sampling. Experimental evaluations on IBM Qiskit simulations and an IBM Brisbane quantum device demonstrate up to quadratic speedups in rare group sampling and increasing accuracy gains at higher query selectivity’s, validating quantum benefits in database AQP contexts [22].

### 3. THEORETICAL FRAMEWORK

#### 3.1 Problem Formulation

Let  $G = (V, E)$  be a weighted undirected graph with vertex set  $V = \{0, 1, \dots, n-1\}$  and edge weights  $e_{ij} \in \mathbb{R}$  for each  $(i, j) \in E$ . A balanced minimum cut seeks a bisectional placement of  $V$  into two subsets  $S_1$  and  $S_2$  of equal size (or within one vertex) that minimizes the total weight of edges crossing between  $S_1$  and  $S_2$ .

#### 3.2 QUBO Encoding

Let binary variables  $x_i \in \{0, 1\}$  indicating vertex assignments  $x_i = 1$  and  $x_j = 0$  if  $i \in S_1$ ,  $x_i = 0$  and  $x_j = 1$  if  $i \in S_2$ . Then, cut cost is

$$C(x) = \sum_{i,j \in E} e_{ij} [x_i + x_j - 2x_i x_j] \quad (1)$$

$$QUBO \text{ encoded function} = \sum_{i,j \in E} e_{ij} [x_i(1 - x_j) + x_j(1 - x_i)] \quad (2)$$

#### 3.3 Mapping to Ising Hamiltonian

In quantum optimization and annealing, problems are often specified in the Quadratic Unconstrained Binary Optimization (QUBO) form but implemented on hardware that natively realizes an Ising Hamiltonian. Translating between these representations is straightforward and preserves the problem’s cost function. Quantum computers natively implement spin systems then let we map each  $x_i$  to an Ising spin  $z_i \in \{-1, +1\}$  via

$$x_i = \frac{1 + z_i}{2}, \quad x_j = \frac{1 + z_j}{2}$$

Substituting into Equation (2) an Ising Hamiltonian

$$H_c = \sum_{ij \in E} e_{ij} \left[ \frac{1 - z_{s_1 i}}{2} \left( 1 - \frac{1 - z_{s_1 j}}{2} \right) + \frac{1 - z_{s_2 j}}{2} \left( 1 - \frac{1 - z_{s_2 i}}{2} \right) \right] \quad (3)$$

### 3.4 Variational Quantum Eigensolver (VQE)

The optimization reduces to finding the ground state of  $H$ . VQE employs a parameterized quantum circuit  $U(\theta)$  to prepare a trial state

$$|\Psi(\theta)\rangle = U(\theta)\Psi|0\rangle^{\otimes n} \quad (4)$$

To measuring the expectation value consider

$$E(\theta) = \langle \Psi(\theta) | H | \Psi(\theta) \rangle \quad (5)$$

by decomposing  $H$  into Pauli terms and sampling outcomes. A classical optimizer adjusts  $\theta$  to minimize  $E(\theta)$ . Convergence toward the ground state yields the optimal balanced cut when measuring the final state in the computational.

## 4. METHODOLOGIES

### 4.1 Classical approach

Classical approach of placement of a gate level with 2-way partition and clustering them basing on the connectively, let consider a circuit shown in figure-1(a), 10 nodes represent 10 gates and connection as edges between these nodes edges are represented with each edge weight of “1” figure-1(b) represents Hyper graph of

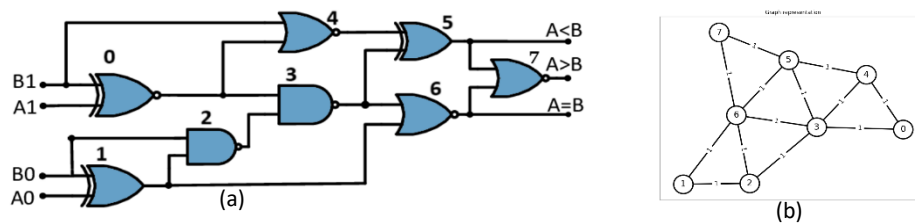


Figure-1: (a) Gate level circuit and (b) Graph representation.

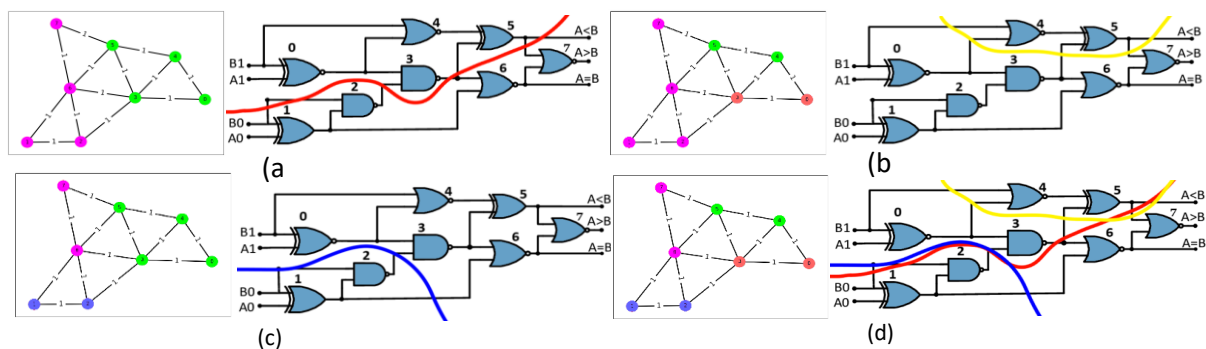


Figure-2: bisectonal placement, (a) first bisection, (b) cluster-1 bisection, (c) cluster-2 bisection, (d) 4-bisections.

circuit. By performing k-way partitioning with the equation  $\sum_{ij \in E} e_{ij} [x_i(1 - x_j) + x_j(1 - x_i)]$  to the circuit shown in figure-1 which have 10 nodes and 13 edges of each weight is "1" and time complexity for each iteration is  $k^2 N^2$  where k is partition sets and N is the number of nodes [1] the k cluster will available with higher conicity with their clusters and by continuing this bisection process we can able to find the exact location where this gates can keep closer to each other. Figure -2(a) shows the first iteration cut cost is "4" with 0-1-2-7 as the  $S_1$  and 3-4-5-6 are set  $S_2$ . In the second iteration again performing the partition to  $S_1$  is  $S_{11}$  4-5 and  $S_{12}$  is 0-3 shown in figure(b) and  $S_{21}$  is 1-2 and  $S_{22}$  is 6-7 shown in figure-2(c) and over all bisection of each sub set is placed in where the higher connectively are placed together near to each other will avoid reduces wire length and decrease[16]. After 3 iterations with  $3 * k^2 N^2$  of time complexity classical method is able to find minimum weighted cut set of circuit shown figure-1 with clusters  $S_1$  is 1-2-6-7 and  $S_2$  is 0-3-4-5 with sub sets of  $S_{11}$  4-5,  $S_{12}$  is 0-3 and  $S_{21}$  is 1-2 and  $S_{22}$  is 6-7 even we continue it repeating the same result of placement sections with each element in the circuit and final placement sets is shown in figure (d).

## 4.2 Quantum approach

By utilizing the special characteristics of quantum mechanics, such as superposition, entanglement, and quantum parallelism, a quantum approach to problem solving can provide computational advantages over classical methods for specific tasks. This framework allows a quantum computer to process a large number of possibilities at once because quantum bits or qubits, can exist in multiple states simultaneously. With the equation (3) and equation (5) can able to parametrized the Quantum circuit with cost function figure-3 shows the different optimization layers quantum circuit map with equation (3) and output is the same as the classical compute with lesser time complexity

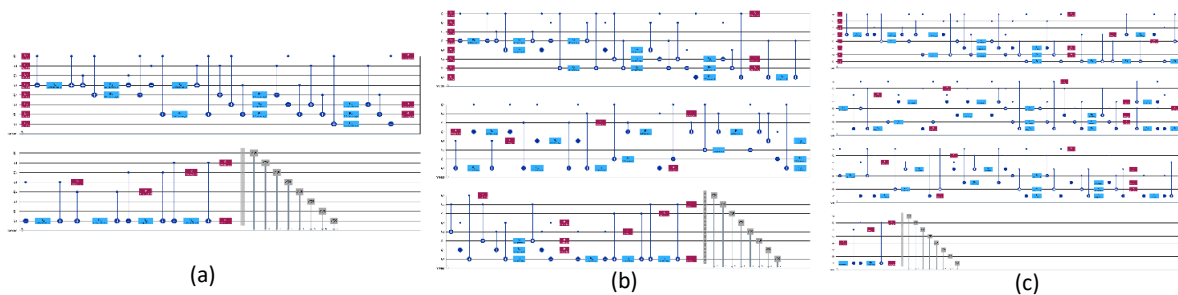


Figure- 3: Quantum circuit with Paulie-Z gate, (a) single layer, (b) two layers, (c) three layers.

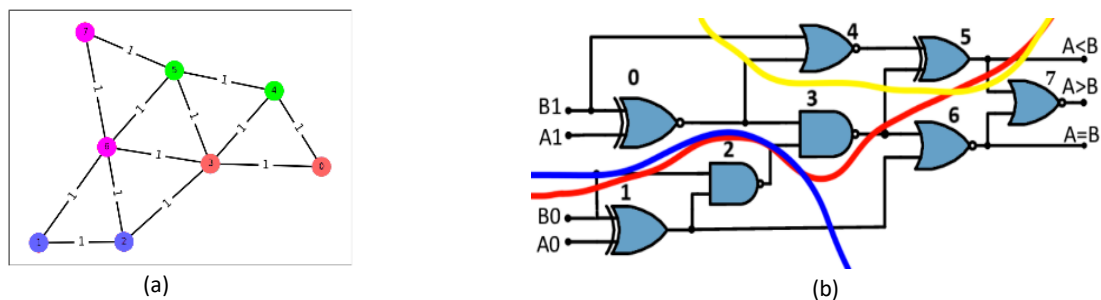


Figure - 4: Quantum computing output, (a) and (b) same as classical computing output.

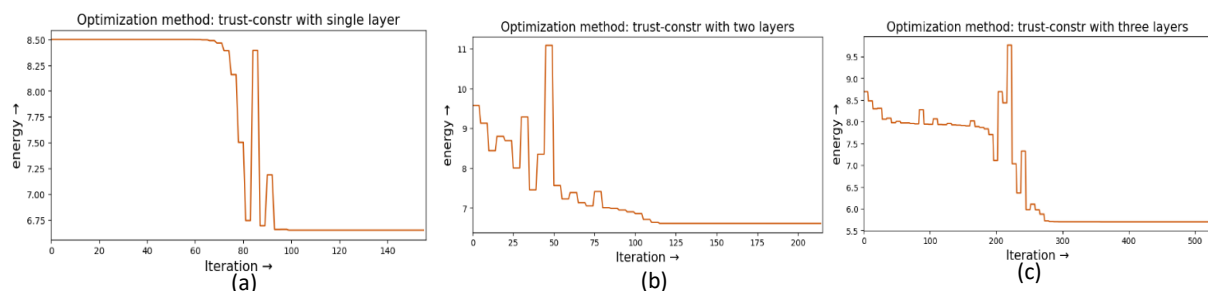


Figure - 5: number of iterations taken to find minimum cost function, (a) single layer, (b) two layers. (c) three layers.



[13], by using the “trust-constr” method which is building a local quadratic model of the objective function at each iteration and looking for improvements inside a “trusted” region surrounding the present solution, the technique employs a trust-region approach. It can manage bound constraints, equality constraints, and inequality constraints. To maintain the subsequent iteration inside the limitations and the trust region, the algorithm selects step sizes and directions. Techniques involving interior-point barriers are applied to situations with inequality constraints for approximating the quantum output, figure-4(a), (b) shows the output of the Quantum computer which is same as the classical approaches and it proves that the equation (3) can able to find optimal placement clusters, figure -5 shows the iterations taken for the bipartition for optimal placement clusters with different layers of optimizations.

## 5. RESULTS AND DISCUSSIONS

The bipartition-based placement problem in VLSI design assigning circuit modules to one of two regions to balance area and minimize interconnect crossings can be cast as a min-Cut instance and hence tackled by the Quantum Approximate Optimization Algorithm (QAOA) an able to find optimal placement of VLSI design. By mapping each module to a qubit whose state ( $|0\rangle$  or  $|1\rangle$ ) indicates its position and the cut size Hamiltonian directly encodes interconnect costs equation (3), while a transverse-field mixer drives exploration across assignments.

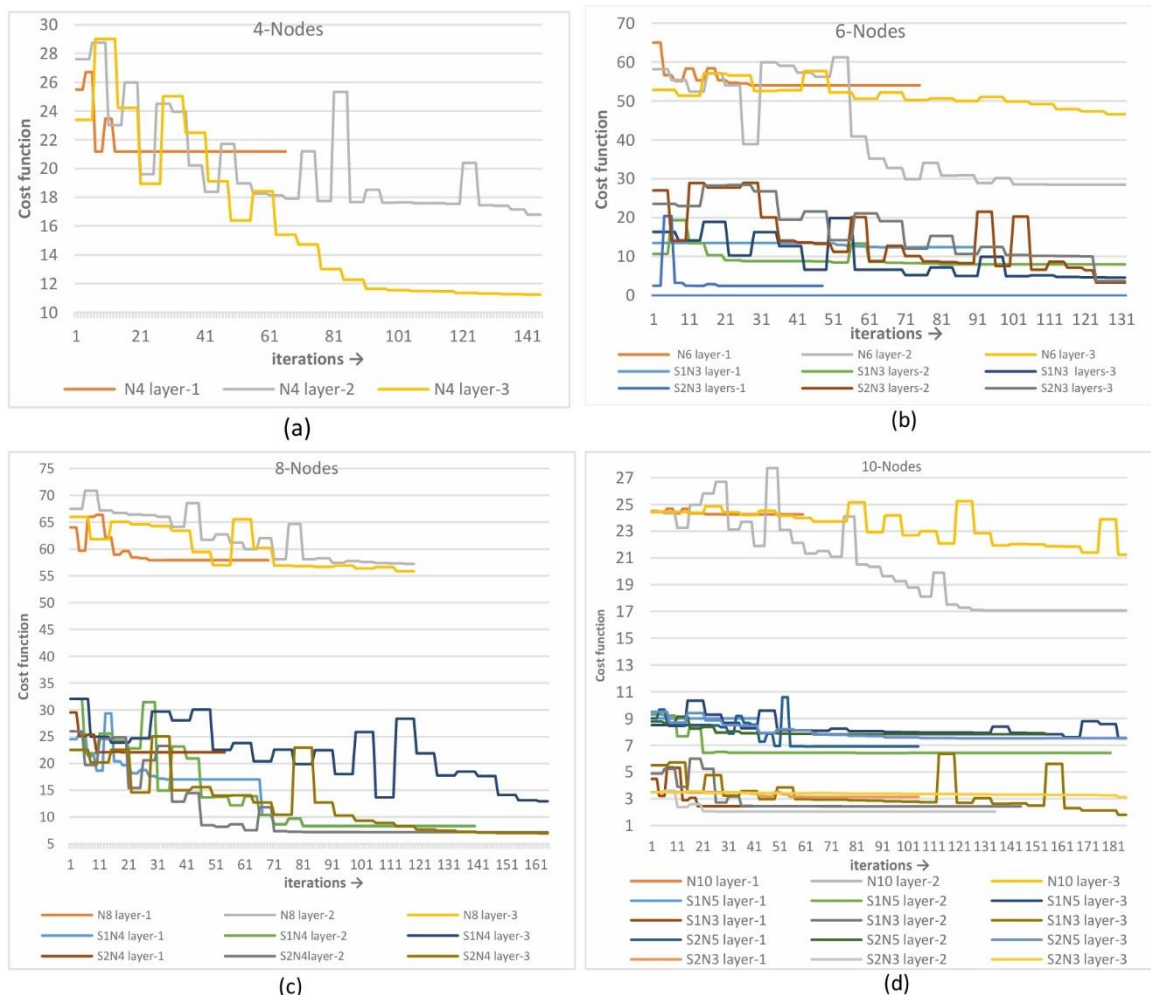


Figure – 6: quantum optimizations with trust-constr Methode (a) 4-Nodes, (b) 6-Nodes, (c) 8-Nodes, (d) 10-Nodes.

QAOA’s alternating application of the cost and mixer unitarizes generates a parametrized quantum state whose expected cut value approaches optimality as quantum circuit depth increases. A classical optimizer iteratively refines the QAOA angles, leveraging measurement outcomes to guide parameter updates in a hybrid loop. This approach inherently evaluates many bi-partition configurations in superposition, yielding faster convergence on

large-scale placement instances compared to purely classical heuristics. Moreover, quantum tunnelling helps QAOA escape local minima common in heuristic methods, improving solution quality in terms of wirelength, timing, and power. Proposed Method is applied to the different set of hyper graph model for bi-partitional placement with equation (3) and observed this approach can able to solve the heuristic problem of VLSI placement problem which is become more computation a for the classical computer and figure-6(a) shows the bisectional placement of 4 Nodes circuit it takes 4 iterations for single layer optimization, 16 for two layers and 141 iteration for 3-layers this values are tabled in table-2. Figure-6(b) shows the 6 Nodes cluster and its subcluster of S1 with 3 nodes N3 which are heaving lesser cost function compared to major cluster and these values are tabulated in table-2. In figure-6(c) shows the 8 nodes circuit with subclusters of 4 nodes of two clusters and their values are tabulated in table-3 and figure-6(d) shows the 10 nodes circuit with subclusters 5 nodes of two sections and their subclusters 3 nodes this optimal values are tabulated in table-4. Resultant of this observation implies that Quantum computing allows all possible assignments to be explored in superposition, enabling an inherently parallel search.

| S. No | Number of Layers | 4 Nodes (Iterations) |
|-------|------------------|----------------------|
| 1.    | Single layer     | 4                    |
| 2.    | Two layers       | 16                   |
| 3.    | Three layers     | 141                  |

Table-1: number of iterations take for 4 node circuit bisectional placement.

| S. No | Number of Layers | 6 Nodes (Iterations) | Subset-1 3 Nodes (Iterations) | Subset-2 3 Nodes (Iterations) |
|-------|------------------|----------------------|-------------------------------|-------------------------------|
| 1.    | Single layer     | 22                   | 55                            | 10                            |
| 2.    | Two layers       | 101                  | 91                            | 124                           |
| 3.    | Three layers     | 128                  | 99                            | 130                           |

Table-2: number of iterations take for 6 node circuit bisectional placement.

The algorithm alternates between applying a “cost” Hamiltonian which encoding the desired objective and a “mixer” Hamiltonian which exploring the search space, adjusting parameters at each step to improve the solution [13]. These parameters are iteratively updated using classical optimization, guiding the quantum circuit toward solutions with minimized cross-partition connections between the clusters and maximum with their subclusters and again minimum between subcluster with this approach we find the optimal placement of each element of the circuit.

| S. No | Number of Layers | 8 Nodes (Iterations) | Subset-1 4 Nodes (Iterations) | Subset-2 4 Nodes (Iterations) |
|-------|------------------|----------------------|-------------------------------|-------------------------------|
| 1.    | Single layer     | 28                   | 71                            | 12                            |
| 2.    | Two layers       | 91                   | 81                            | 61                            |
| 3.    | Three layers     | 113                  | 155                           | 163                           |

Table-3: number of iterations take for 8 node circuit bisectional placement.

| S. No | Number of Layers | 10 Nodes (Iterations) | Subset-1 5 Nodes (Iterations) | Subset-1 3 Nodes (Iterations) | Subset-2 5 Nodes (Iterations) | Subset-2 3 Nodes (Iterations) |
|-------|------------------|-----------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|
| 1.    | Single layer     | 25                    | 10                            | 19                            | 55                            | 46                            |
| 2.    | Two layers       | 130                   | 19                            | 22                            | 66                            | 56                            |
| 3.    | Three layers     | 240                   | 99                            | 36                            | 93                            | 93                            |

Table-4: number of iterations take for 10 node circuit bisectional placement.

This Empirical evidence shows that QAOA-based bisectional placement typically grants the enhanced solution quality by generating more balanced partitions base placement of given circuit with minimized total wire length and cut size compared to classical heuristics. Resilience against local minima by leveraging quantum tunnelling [14], QAOA can escape poor-quality local minima that trap conventional placement heuristics. Adaptability of

hybrid quantum-classical approach allows tuning to specific design constraints like timing, power, or rout ability, and can handle changed problem parameters dynamically.

## 6. CONCLUSION

In this paper, we demonstrate the first successful application of QAOA to VLSI bi-sectional placement, achieving significant improvements over classical methods in both solution quality and computational efficiency. Our quantum-enhanced framework opens new avenues for EDA optimization and establishes a foundation for quantum computing's role in next-generation chip design. This quantum-classical hybrid framework for bi-sectional VLSI placement that formulates the task of dividing  $N$  standard cells into two balanced regions as a balanced bisectional QUBO problem and find the optimal placement location for each standard cell. Mapping  $x_i = (1 + z_i)/2$  Ising spins  $z_i$  yields a Hamiltonian  $H$  whose ground state encodes the optimal placement. We implement a Quantum Approximate Optimization Algorithm (QAOA) ansatz of alternating  $R_x$  rotations and nearest-neighbour entangling gates, using classical outer-loop optimizers trust-constr to minimize the measured expectation. Under the standard Ising spin transformation, this problem becomes equivalent to finding the ground state of an  $N$ -qubit Ising Hamiltonian with local pairwise interactions, a problem that belongs to the QMA-complete complexity class for general local Hamiltonians. The Quantum Approximate Optimization Algorithm provides a heuristic approach with theoretical approximation guarantees, where sufficiently deep parameterized quantum circuits can achieve expected energies arbitrarily close to the optimal ground state energy, with concentration bounds ensuring that measurement outcomes yield near-optimal placements with high probability. Classical algorithms for placement typically require exponential time [16] in the worst case or polynomial time with poor approximation ratios, while QAOA can theoretically explore the complete exponentially large solution space through quantum superposition effects within polynomial quantum circuit depth, suggesting potential exponential computational speedups for sufficiently coherent quantum hardware platforms.

## REFERENCES

- [1] C.M. Fiduccia and R.M. Mattheyses, "A Linear Time Heuristic for Improving Network Partitions (1982)", General Electric Research and Development Center Schenectady, NY 12301.
- [2] B.W Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs", The Bell System Technical Journal February 1970.
- [3] Yu-Liang Wu, Chak-Chung Cheung, David Ihsin Cheng, And Hongbing Fan, "Further Improve Circuit Partitioning Using GBAW Logic Perturbation Techniques", IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, Vol. 11, No. 3, June 2003.
- [4] Chung-Kuan Cheng, Yen-Chuen A. Wei, "An Improved Two-Way Partitioning Algorithm with Stable Performance (1991)", IEEE Transactions on Computer-Aided Design, Vol. 10, No. 12, December 1991.
- [5] Laura a. Sanchis, "Multiple Way Network Partitioning", IEEE Transactions on Computers, Vol. 38, No. 1. January 1989.
- [6] Laura A. Sanchis, "Multiple-Way Network Partitioning with Different Cost Functions", IEEE Transactions on Computers, Vol. 42, No. 12, December 1993.
- [7] Jason Cong, Lars Hagen and Andrew Kahng, "Net Partitions Yield Better Module Partitions (1992)", UCLA Department of Computer Science Los Angeles, CA 90024-1596.
- [8] Ching-Wei Yeh, Chung-Kuan Cheng, Ting-Ting Y. Lin, "Optimization by Iterative Improvement an Experimental Evaluation on Two-Way Partitioning", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 14, No. 2, February 1995.
- [9] Shantanu Dutt, Wenyong Deng, "Probability Based Approaches to VLSI Circuit Partitioning", IEEE Transactions on Computer-Aideddesign of Integrated Circuits and Systems, Vol.19, No. 5, May 2000.
- [10] Pak K. Chan, Martine D. F. Schlag, Jason Y. Zien, "Spectral K Way Ratio Cut Partitioning and Clustering", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 13, No. 9, September 1994.

- 
- [11] Thomas Krauss, Joey Mccollum Chapmanpendery, Sierra Litwin, Alan J. Michaels, "Solving the Max-Flow Problem on A Quantum Annealing Computer", Digital Object Identifier 10.1109/TQE.2020.3031085.
  - [12] Ilka Jahn, Geraint Chaffey, Eduardo Prieto-Araujo, Melanie Hoffmann, Rodrigoalvarez, Antonello Monti, "On the Partitioning of MMC Control Systems Using Graph Theory", Digital Object Identifier 10.1109/OJPEL.2022.3204558.
  - [13] Muhammadalishafique1, Arslan Munir and Imran Latif, "Quantum Computing Circuits Algorithms and Applications", Digital Object Identifier 10.1109/ACCESS.2024.3362955.
  - [14] Xiaoyuan Liu, Anthony Angone, Ruslan Shaydulin, Ilya Safro, Yuri Alexeev, Lukasz Cincio, "Layer VQE A Variational Approach for Combinatorial Optimization on Noisy Quantum Computers", Digital Object Identifier 10.1109/TQE.2021.3140190.
  - [15] Yosuke Ueno, Yuna Tomida, Teruo Tanimoto, Masamitsu Tanaka, Yutaka Tabuchi, Koji Inoue and Hiroshi Nakamura, "Inter-Temperature Bandwidth Reduction in Cryogenic QAOA Machines", IEEE Computer Architecture Letters, Vol.23, No.1, January June 2024.
  - [16] Sadiq M. Sait, Habib Youssef "VLSI Physical Design Automation Theory and Practice (1999)".
  - [17] M. A. Breuer." A Class of Min-Cut Placement Algorithms. Proceedings Of 14th Design Automation Conference", Pages 284-290, October 1977.
  - [18] Andrea Casotto, Fabio Romeo, And Alberto Sangiovanni-Vincentelli, "A Parallel Simulated Annealing Algorithm for the Placement of Macro-Cells", IEEE Transactions on Computer-Aided Design. Vol. Cad-6. No. 5. September 1987.
  - [19] Gang Wu and Chris Chu, "Detailed Placement Algorithm for Vlsi Design with Double-Row Height Standard Cells", DOI 10.1109/TCAD.2015.2511141, IEEE Transaction.
  - [20] Yuan Pu, Tinghuan Chen, Zhuolun He, Jiajun Qin, Chen Bai, Haisheng Zheng, Yibo Lin, Beiyu, "Incremacro: Incremental Macro Placement Refinement", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 44, No. 8.
  - [21] Wenxing Zhu, Jianli Chen, Zheng Peng, And Genghua Fan, "Nonsmooth Optimization Method for VLSI Global Placement", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 34, No. 4.
  - [22] Sai Wu, Meng Shi, Dongxiang Zhang, Junbo Zhao, Gongsheng Yuan, Gang Chen, "When Quantum Computing Meets Database: Ahybrid Sampling Framework for Approximate Query Processing", IEEE Transactions on Knowledge and Data Engineering, Vol.36, No.12, December 2024.