

## P<sup>2</sup>CTS-3D IC: An Efficient Placement Aware Partitioning and Clock Tree Synthesis in 3D-integrated Circuits

Radeep Krishna Radhakrishnan Nair and Sivakumar Pothiraj\*

Department of Electronics and Communication Engineering, School of Electronics and Electrical Technology, Kalasalingam Academy of Research and Education, Krishnankoil, Tamilnadu, India

Rajini Nagarajan

Department of Mechanical Engineering, School of Automotive and Mechanical Engineering, Kalasalingam Academy of Research and Education, Krishnankoil, Tamilnadu, India

\* Corresponding author. E-mail: sivapothi@gmail.com DOI: 10.14416/j.asep.2020.09.001

Received: 29 April 2020; Revised: 9 July 2020; Accepted: 1 September 2020; Published online: 21 September 2020

© 2020 King Mongkut's University of Technology North Bangkok. All Rights Reserved.

### Abstract

The three dimensional integration of electronic circuits (3D-ICs) is one of the most promising approaches to encounter eternally increasing demands of functionality, performance, and power consumption. However, there exist many challenges in the minimization of power dissipation, skew, latency, wirelength, and obstacle avoidance in 3D-ICs. To overcome the discussed problems, an efficient placement aware partitioning and synthesis by clock tree have been proposed in 3D-ICs. In the proposed framework, the first process begins with the partition of netlist into multiple layers using hybrid Chemical Reaction Optimization (CRO) and K-Medoid algorithm and the placement process has been executed using Grid Warping Technique (GWT). Then, the routing process has been performed using the Line Search (LS) algorithm. The clock tree synthesis process gets accomplished with a Density Sorting (DS) algorithm that considers multiple TSVs. Further, the Clock tree topology has been implemented using the algorithm of Merging with Defer and Embedding (DME) and the algorithm of Graph with Neighbor at the Nearest. Consequently, the buffering and the embedding processes have been carried out to measure the total power dissipation and timing parameters. The proposed work minimizes the skew value compared to the other existing methods like GAP, EP, and SEA and it also provides efficient clock tree synthesis with Defer Merging and Embedding (DME) and Nearest Neighbor Graph (NNG) algorithm based methods as well as it reduces the skew optimally. Finally, the performance of the proposed work has been evaluated and proved to be better by considering the following metrics such as wirelength, power, skew, latency, and area.

**Keywords:** Three dimensional integrated circuits, Partitioning, Placement, Routing, Synthesis using clock tree, Through silicon via

### 1 Introduction

The advancements in VLSI technology lead to an enhancement in 3D-IC fabrication [1] emerging from 2D-IC. The difficulties surrounded with these enhancement include the die yields with its impact on

the 3D Integration and the increment in the component densities [2], [3]. The 2D-ICs have multiple problems related to the performances, power consumption, and interconnected delays. 3D-ICs have evolved as an approach to overwhelm problems in 2D-ICs. 3D-ICs play a significant role in portable and smart device



applications such as mobile phones and 5G wireless devices. In the physical design of 3D-IC, partitioning, placement, and routing processes are essential. The information-sharing graph-based partitioning approach [4] has been intended for 3D-IC fabrication. Computation of Graph with Information leads to appropriate partitioning. A multi-objective circuit partitioning technique has been introduced [5] in 3D-IC. Here, an algorithm with searching especially the Tabu search is utilized. Also, a simulated Annealing Algorithm has been used. The above algorithms are considered for area balancing and Through Silicon Via (TSV) count minimization, respectively for active partitioning. Kernighan Lin (KL) graph bi-partitioning algorithm has offered [6] useful partitioning of netlist into multiple layers for 3D-IC. KL partitioning method generates more number of clusters and each possesses multiple cores. A 3D-IC physical design strategy includes partitioning, placement, and routing processes. The placement process is essential for 3D-IC fabrication and that allocates the optimum position for each component. The placement based on non-uniform scaling gets intended [7] to optimize the consumption of power in multi-tier 3D ICs at the monolithic gate level.

In this paper, the core objective is to minimize the wirelength, power dissipation, latency, and skew. The proposed framework's pivotal outcome contains four phases for practical physical design fabrication of 3D-IC, including partitioning, routing, placement, and clock tree synthesis. In the partitioning phase, Chemical Reaction Optimization (CRO) algorithm [8] combined K-Medoid hybrid algorithm [9] has been introduced to partition the netlist into multiple layers. The most exceptional medoid point is selected using Chemical Reaction Optimization (CRO) algorithm and that improves the K-Medoid algorithm partitioning process. The congestion avoidance gets achieved using Grid Warping Technique (GWT) [10] based placement. In Grid Warping Technique (GWT) placement, macros are static, and the layers are dynamic warped to locate macros without any congestion. The wirelength is optimally reduced and Line Searching [11] algorithm based routing has been proposed. LS algorithm forms a path between components by generating horizontal and vertical lines between source and destination terminals. The skew is minimized and the clock tree synthesis process gets implemented. Density Sorting

[12] algorithm has been utilized to cluster high-density sink nodes and clock tree topology gets implemented using Defer Merging and Embedding (DME) [13] and Nearest Neighbor Graph (NNG) [14] algorithm. To measure the total power consumption and timing parameters, buffering, and the embedding processes are executed.

The rest of the paper has been organized as follows: Section 2 details the works that are performed previously. Section 3 illustrates the problem description and section 4 presents the explanation method. Section 5 illustrates the results and discussion and Section 6 depicts the conclusion.

## 2 Related Works

The previous works that are conducted on 3D-IC fabrication have been portrayed in this section. Four processes, such as partitioning, placement, routing, and clock synthesis have been investigated. Sivaranjani *et al.* [15] have proposed Fast Force Directed Simulated Annealing (FFSA) algorithm. During the process of annealing, the force as a new factor gets considered in the algorithm of FFSA. It modifies every move as a force-directed move. Banerjee *et al.* [16] have proposed partitioning based on the graph in the 3D-IC partitioning technique. The proposed approach consists of two partitioning parts such as initial partitioning and refined partitioning. The algorithm of Breadth-First Search (BFS) is utilized in the initial partitioning part. The Algorithm of BFS gives initial solution and further partitioning process is performed. Jang *et al.* [17] have proposed Min-Cut partitioning methodology. It contains five stages. Shanavas *et al.* [18] have proposed a hybrid evolutionary algorithm-based placement in 3D-IC. The hybrid algorithm combines both the genetic algorithm and the SA algorithm. Caleb Serafy *et al.* [19] have proposed coupling aware global 3D-IC placement. The Fixed cell algorithm is used to place components that adopt cell placement with a fixed standard. Athikulwongse *et al.* [20] have proposed thermal coupling in placement with 3D-IC. The thermal coupling approach utilizes two level processes for 3D-IC placement. Firstly, the Local Power density is minimized in each die where the extension of TSV is done. Chen *et al.* [21] have proposed a Topology Aware Adaptive Routing (TAAR) scheme in 3D-IC systems.

The TAAR, which gets introduced, contains three routing stages and they are dynamically modified concerning the topology status of routing paths. The TAAR increases routing flexibility. Ghosal *et al.* [22] have proposed obstacle aware Steiner routing in 3D-ICs. It is the Recti-linear Minimum Steiner Tree (RMST) algorithm that gets executed for routing. Ebrahimi *et al.* [23] have completed Minimal Adaptive Routing for 3D-IC fabrication. The Hamiltonian path is utilized in the MAR algorithm. The MAR algorithm discovers the routes and forwards the packets by finding the neighbor of the current switch. Cai *et al.* [24] have proposed efficient insertion of buffering.

For that, the clock tree synthesis with slew constrained needs has been utilized. The clock tree topology is constructed and the algorithm with an obstacle awareness has been utilized. Using the NGSPICE Simulation, the lookup table gets built. It ensures the attainment of slew rate and buffer delay. Liu *et al.* [25] have quoted for 3D clock tree synthesis, the TSV aware topology generation is ideal. In the generation process of the clock tree topology, the TSV overhead is estimated by the TSV Equivalent Wirelength (TEWL). This method computes the clock network over heading by combining with horizontal wirelength and it is traditionally used. Liu *et al.* [26] have proposed 3D clock tree synthesis with whitespace awareness. For the tree synthesis with 3D Clock, the algorithm for TSV arrangement has been implemented for the white space awareness.

### 3 Problem Definition

In this section, the problems that are present in the existing 3D-IC fabrication framework have been described. The genetic algorithm based partitioning approach is used [27] for partitioning the netlist into multiple layers in 3D-IC and the proposed genetic algorithm chooses the initial population from the given netlist. The genetic algorithm contains three significant phases to partition the netlist and they are selection, mutation, and cross over. The fitness function gets computed to select the partition points. In the selection process, two chromosomes get selected for each time. This chromosome selection is done based on the computed fitness values. The highest fitness function value chromosome gets selected for the parent each time. The genetic algorithm is time-consuming

and sensitive to the initial parameters, and it cannot guarantee an optimal solution. Electrostatics based placement is pursued [28] in 3D-ICs fabrication. Here, a mixed size placement algorithm has been used for electrostatics based placement in 3D-ICs. Density 3D function is introduced to provide global smoothness. In 3D density function, every placement objects (cells) gets converted to a positively charges cuboid. 3D numerical solutions have been obtained using Fast Fourier Transform (FFT) technique. In a 3D numerical solution, spectral methods are used for solving the 3D-Poisson equation using frequency decomposition. A non-linear 3D pre-conditioner has been utilized to equalize all moving entities from an optimization perspective. The 3D non-linear preconditioning method enhances the convergence rate rather than the solution quality. In this work, the electrostatics placement approach has been proposed and it is not suitable for large macros. The maze routing algorithm is utilized [29] for 3D-IC routing process. The stochastic algorithm has been used to deliver the Ripping-up and Re-routing (R&R) processes. Maze routing algorithm establishes the shortest path using the BFS algorithm, and the R&R process minimizes congestion and wirelength. Stochastic evaluation algorithm controls R&R, and it also adjusts the parameters of Maze routing and framing. The stochastic evaluation algorithm performs net selection in R&R. Two steps are majorly available in the router and they are globally proposed. With the routing of the nets, the congestion is identified and eradicated towards the end of the first step. Maze routing takes more amount of memory to store routing information. BFS algorithm has been used to estimate the shortest path and it consumes more time. The synthesis of the clock tree has been introduced [30] as an awareness in terms of power and obstacle for 3D IC. Here, 3D clock tree synthesis follows three significant processes. 1) Single TSV is used for generating 3D abstract clock tree topology. 2) Obstacles are avoided in the routing process. 3) Clock skew and slew control get accomplished using buffering and embedding. Here, the sinks with a clock are multiple times located in dies where the connection is completed by a single tree. Dies get connected through the TSV, where clock signal travels from one die to another. Buffer insertion algorithms are finally utilized in the clock tree construction network. The buffering process continues in a top-down manner.

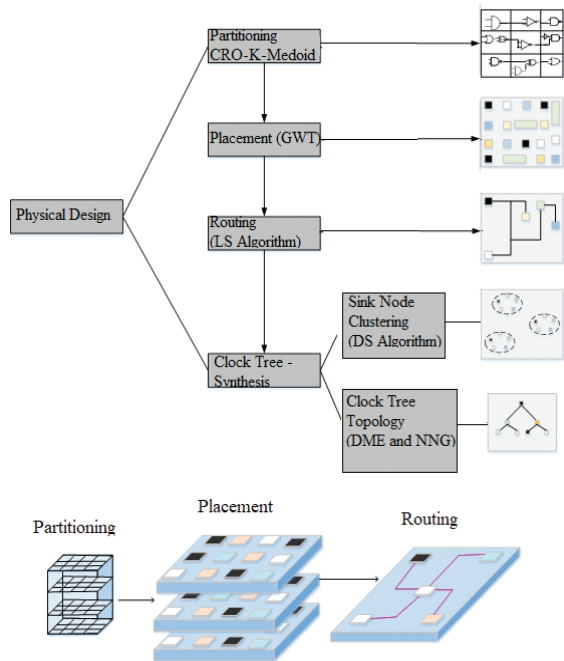
Buffer location is frequently updated in a 3D clock tree. In this paper, multiple TSVs are not considered for clock tree synthesis.

## 4 Methods

In this section, the overview of the proposed work and the 3D-IC fabrication processes containing partitioning, placement, routing, and clock tree synthesis are demonstrated.

### 4.1 System overview

The intended framework mainly concentrates on four stages, such as partitioning, placement, routing, and clock tree synthesis. Figure 1 demonstrates the overall process of the proposed framework. In the first stage, the partitioning process minimizes the complexity by dividing larger netlist into various layers. A hybrid algorithm based partitioning approach has been pursued for partition the netlist. The proposed hybrid algorithm is a combination of Chemical Reaction Optimization (CRO) algorithm and K-Medoid algorithms [31]. Here, Chemical Reaction Optimization (CRO) algorithm is used to explore the optimum medoid point, and further the process has been carried out in K-Medoid algorithm. In the second stage, the placement process gets accomplished using Grid Warping Technique (GWT) that warps layer until better macro placement gets achieved. In the third stage, routing has been executed using the Line Search algorithm to determine optimum paths for communication between two macros. In the fourth stage, clock tree synthesis has been performed and it considers multiple TSVs. The high-density sink nodes are clustered with the algorithm of Density Sorting applied. Clock tree topology is constructed using Defer Merging and Embedding (DME) and Nearest Neighbor Graph (NNG) algorithm. The Deferred-Merge Embedding (DME) algorithm embeds internal nodes of the topology  $G$  via a two-phase process. A bottom-up phase constructs a tree of line segments representing the loci of possible placements of the internal nodes. A top-down phase then resolves the exact locations of all internal nodes. DME embeds efficiently the wire of a given topology. The Nearest Neighbors Graph (NNG) is a directed graph connecting each element with its nearest neighbors. Since the Defer Merging and Embedding (DME)



**Figure 1:** Physical Design Strategy of the proposed work.

algorithm only optimizes a prescribed topology, it cannot achieve all possible improvements of the clock tree construction. As a result, Neighbour Graph (NNG) algorithm, which gives promising results, has been utilized. Finally, the buffering and embedding processes are carried out to measure the total power dissipation and timing parameters.

### 4.2 Partitioning

In partitioning, a hybrid algorithm that is a combination of Chemical Reaction Optimization (CRO) algorithm and K-Medoid algorithms has been pursued. Here, the K-Medoid algorithm is utilized to partition the netlist into multiple layers. In the K-Medoid algorithm, an initial number of clusters is selected to partition the larger netlist. In the K-Medoid algorithm, initial medoid selection is critical. Hence, Chemical Reaction Optimization (CRO) algorithm has been employed to select medoid points in the K-Medoid algorithm. Chemical Reaction Optimization (CRO) algorithm is a meta-heuristic algorithm motivated by the nature of chemical reactions. Chemical Reaction Optimization

(CRO) algorithm starts with an initial number of solutions and performs a sequence of collisions as well as it finds the final stable condition. In the Chemical Reaction Optimization (CRO) algorithm, the initial object is considered as a molecule and it computes potential energy for each molecule. Chemical Reaction Optimization (CRO) algorithm changes its population in each iteration and this factor differentiates Chemical Reaction Optimization (CRO) algorithm from other evolutionary algorithms. The potential energy of each molecule considers the fitness function computation of each molecule. In the proposed approach, potential energy has been considered as a minimum distance. Molecoll is the threshold value given to Chemical Reaction Optimization (CRO) algorithm as an input to perform uni-object reaction or inter-object reaction.

After selecting the medoid point using Chemical Reaction Optimization (CRO) algorithm, further processes are completed with the algorithm of K-Medoid. The algorithm of K-Medoid is based on Partitioning Around Medoid (PAM) i.e. works on PAM [32] to partition the netlist. The algorithm of K-Medoid is a classical partitioning technique of clustering where data of 'n' subjects are converted into clusters with 'k'. The K-Medoid algorithm is robust to noise and reliable to outliers. K-Medoid algorithm computes total swapping cost in each iteration and the minimum total cost pairs are selected in each iteration. The hybrid Chemical Reaction Optimization (CRO) algorithm and K-Medoid algorithm steps are given below.

#### Step 1:

In the first step, the number of clusters, number of objects, object position, object size and Molecoll are initialized.

#### Step 2:

In this step, fitness function for each object is computed in Chemical Reaction Optimization (CRO) algorithm based on the distance ( $d$ ) between each objects. Fitness function can be expressed as follows [Equation (1)]:

$$F = f(d) \quad (1)$$

Where  $f(d)$  denotes fitness function of each object in terms of distance.

#### Step 3:

If randomly selected number 'r' is greater than Molecoll, then uni-object reaction is started or else inter-object reaction is started. Reaction Selection ( $RS$ ) can be expressed as below Equation (2):

$$RS = \begin{cases} r > Molecoll; U_{OR} \\ r < Molecoll; I_{OR} \end{cases} \quad (2)$$

Where,  $U_{OR}$  represents uni-object reaction and  $I_{OR}$  represents inter-object reaction.

#### Step 4:

If uni-object reaction is selected, there will be a threshold value to guide whether to go wall ineffective collision or decomposition. If the total collision number parameter is greater than the allocated threshold, then it goes to the decomposition state or else it goes to the wall ineffective collision.

Wall ineffective collision is uni-object reaction with only one object. Here, object  $\tau$  is allowed to change to another one  $\tau'$ , if their values are consensus with following inequity [Equation (3)]:

$$F(\tau) + K(\tau) \geq F(\tau') \quad (3)$$

Where,  $F(\tau)$  denotes fitness function of object  $\tau$ ,  $F(\tau')$  represents fitness function of object  $\tau'$  and  $K(\tau)$  denotes kinetic value which is a non-negative number used to escape from local optimum.

In decomposition, object  $\tau$  may decompose into two new objects  $\tau_1'$  and  $\tau_2'$ , if their values are consensus with the following inequity [Equation (4)]:

$$F(\tau) + K(\tau) \geq F(\tau_1') + F(\tau_2') \quad (4)$$

Where,  $F(\tau_1')$  denotes fitness function of object  $\tau_1'$  and  $F(\tau_2')$  denotes fitness function of object  $\tau_2'$

#### Step 5:

If inter-object reaction is selected, there will be a threshold value to guide whether to go synthesis or inter-object ineffective collision. If the kinetic value of two objects is less than the threshold, then it will go to synthesis or else go to the inter-object ineffective collision.

Synthesis is inter-object reaction and here, two objects are combined to form new object  $\tau'$ , if their

values are according to the following expression [Equation (5)]:

$$F(\tau_1) + F(\tau_2) + K(\tau_1) + K(\tau_2) \geq F(\tau') \quad (5)$$

Where,  $K(\tau_1)$  denotes kinetic value of object  $\tau_1$  and  $K(\tau_2)$  denotes kinetic value of object  $\tau_2$ .

In inter-object collision, objects  $\tau_1$  and  $\tau_2$  may decompose into two new objects  $\tau_1'$  and  $\tau_2'$ , if their values are satisfied with the following expression [Equation (6)]:

$$F(\tau_1) + F(\tau_2) + K(\tau_1) + K(\tau_2) \geq F(\tau_1') + F(\tau_2') \quad (6)$$

If the stopping criterion is satisfied, then CRO terminates its process. Selected medoid from above steps is used for further process in K-Medoid algorithm.

#### Step 6:

Based on the computed medoid from the previous steps, K-Medoid algorithm groups the objects. First, it computes total swapping cost to group the objects. The total swapping cost of medoid as well as the object is expressed as follows [Equation (7)]:

$$T_C = C_{m,o} \quad (7)$$

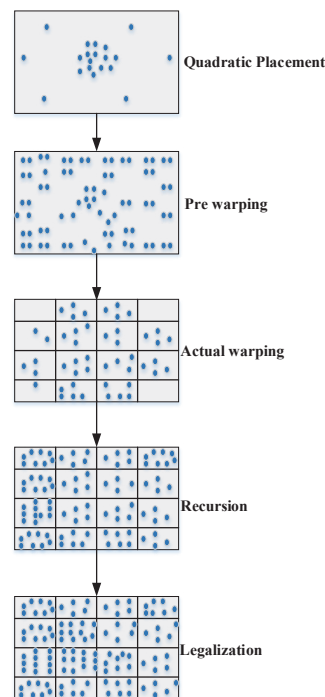
Where,  $C_{m,o}$  denotes total cost to group the medoid and the objects.

#### Step 7:

If the computed total cost is less than zero, then a new object is replaced with old medoid. This process is repeated until it finds no change between objects. Finally, the given netlist is partitioned into multiple layers.

### 4.3 Placement

The proposed placement procedure makes use of Grid Warping Technique (GWT) for effective placement in 3D-IC physical strategy [33]. In Grid Warping Technique (GWT), surfaces on which macros get placed move to optimize macros location. The other placement techniques move macros and they increase the cost function, whereas the present proposed approach moves the surfaces that minimize the cost function. Figure 2, demonstrates the proposed Grid



**Figure 2:** Grid warping technique based placement.

Warping Technique (GWT) based placement in 3D-IC fabrication. This Grid Warping Technique (GWT) placement begins with the conventional quadratic analytical placement. Here, each macro gets placed in dimensionless point connected to the group of weighted two-point wires. Using Grid Warping Technique (GWT), overall squared Euclidean wirelength is minimized. An identical  $n \times n$  matrix grid is placed on the placement surface and in that, each grid intersection points is known as the control point. The warping method elastically moves these points and they get controlled with grid deformation approximation. The proposed Grid Warping Technique (GWT) depends on decomposition with recursive nature in each region and it decomposes inside with the confinement of cells. A solver at the layer of recursive in each layer runs with quadratic and global nature and enforces constraint of gravity based on the center in the sub-region for each level. It is a starting point and serves for the placement with each sub-region that completes warping. A local enhancement step is utilized to improve the wirelength in each level and that is known as re-warping. The Grid Warping Technique (GWT) follows five significant steps for accurate

placing of macros in the partitioned layer and they are shown in Figure 2. The following steps are executed in Grid Warping Technique (GWT) placement.

#### 4.3.1 Quadric placement

At first, initial placement has been achieved by placing macros in partitioned layers randomly.

#### 4.3.2 Pre-warping

Pre-warping is performed after initial placement. Here, problems are geometrically conditioned for each more straightforward solution.

#### 4.3.3 Actual warping

After pre-warping, actual warping gets implemented and that moves the control points to the deformation grid in the grid matrix.

#### 4.3.4 Recursion

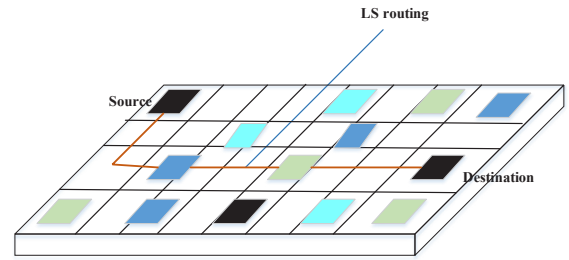
Recursion is performed after the completion of actual warping. This process is executed to locate the warp in the optimum position without any congestion. This process is repeated until all macros get correctly placed in the layer surface.

#### 4.3.5 Legalization

Finally, the final legalization process is accomplished and it is the most critical process in grid warping. Hence, it alleviates all the problems in placement. Legalization reforms the overlapped macros and minimizes the displacement in the partitioned area.

### 4.4 Routing

In proposed framework, routing is established using the LS algorithm. Hightower’s Line Search algorithm which is one of the types in the Line Search (LS) algorithms, is pursued [34]. The proposed Hightower LS router runs faster than the Maze routing algorithm. Hightower Line Search (LS) algorithm stores only feature that gets already located in the routing surface. This behavior of the algorithm leads to the minimization of memory needs.



**Figure 3:** Hightower’s LS routing.

Figure 3, illustrates Hightower’s Line Search (LS) algorithm based routing in 3D-fabrication. The proposed algorithm generates each sub-paths that originate from two mid-points, since it provides a line search approach. The proposed Hightower’s Line Search (LS) algorithm is a vector router that allocates two line segments in orthogonal directions from the source and the destination until an obstacle gets stretched. This process is repeated until the line segment originated from the source intersects the line segment originated from the destination.

The basic idea of the proposed algorithm is to assign line segments from a point and prolonging them until either an obstacle or the edge of the routing area gets met. The developed line segments are either horizontal or vertical in orientation. The point lies on the line segment and it is known as an escape point.

Hightower Line Search (LS) algorithm adds only one new line segment to each present line segment. The Hightower Line Search (LS) algorithm describes three escape processes to permit the router to establish a path around different types of obstacles. For instance, if a line segment parallels a wiring segment, then the new line segment is located beyond the segment's endpoint. The Hightower Line Search (LS) routing algorithm generates fewer but escape line than the Mikami algorithm.

For a path containing ‘1’ bends, at most 4l line segments would be obtained. Hence, the total time for consideration is given below [Equation (8)]:

$$\sum_{i=1}^{4l} (n + i) = O(nl + l^2) \quad (8)$$

Given a set of vertical segments, look for the horizontal slice line with the condition that the number of line segment endpoints in the upper and lower bisector is balanced. For a set ‘s’ of horizontal line segments

and a vertical line segments query  $h$ , all line segments are discovered in 's' or the minimum line segment that intersects the 'h'. Hence, if 'K' intersection exists, the total time needed is represented as  $O(\log^2 n + K)$ .

#### 4.5 Synthesis with clock Tree

The process synthesis of clock tree gets done. Here, the distribution of clock is consistently checked. It gets done to all elements of 3D-IC physical design consistently. The main objective of clock tree synthesis includes latency and skew minimizations. The synthesis with a clock tree contains two processes: tree building and tree balancing with the clock. The proposed clock tree synthesis process considers multiple TSVs. In the present framework, the Density Sorting algorithm [35] has been used to cluster high-density sink nodes in the clock tree. Clock tree topology has been generated by implementing the Defer Merging and Embedding (DME) and Nearest Neighbor Graph (NNG) algorithms. Finally, the buffering and embedding processes are carried out to measure the total power dissipation and timing parameters.

##### 4.5.1. Sink node clustering

Density Sorting (DS) algorithm has been proposed to achieve a high-density sink node with clustering by clock tree. This DS algorithm-based clustering approach achieves a balanced distribution of TSVs. Hence, multiple TSVs are used and they improve the reliability of the 3D-IC design.

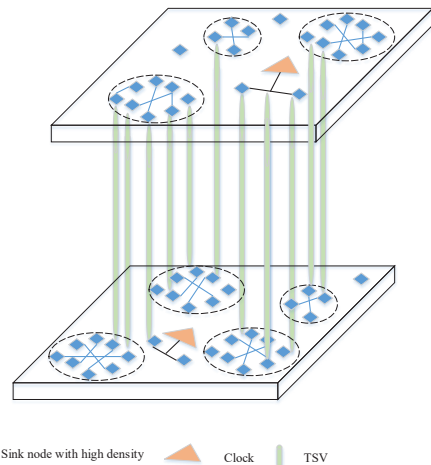
Figure 4, depicts the sink node clustering with high density using DS algorithm. The proposed DS algorithm contains four stages to cluster the sink nodes.

##### Step 1:

In the first step, the calculation of  $R$ , the Manhattan radius, has been calculated with the number of neighboring sink node. As the center, a sink node among one is selected. The condition is that under Manhattan radius of a certain level, the neighbor of two or more does likely not insert the TSV with the nodes and its neighbor. The neighboring nodes are labeled to cluster with appropriate marks.

##### Step 2:

The cluster marked contains the selection of



**Figure 4:** Sink node clustering.

nodes. Here, the center set nodes, which are more than neighbors, are in proximity.

##### Step 3:

A cluster can get sorted as one with all proximity sink nodes. The breaking of iteration takes place, if the cluster finds unsuitable with the addition of any sink node. It allows the returning out of all sink nodes within-cluster

##### Step 4:

Once the 'n' iteration is over, all useful clusterings take place for all the sink nodes with high-density area.

##### 4.5.2 Clock tree topology

Clock tree topology has been constructed using DME and NNG algorithms. The running of DME [36] is fast and time is linear with the synchronization of elements. DME yields optimal (i.e., minimum wirelength) total wirelength, that in turn reduces the size of the chip. Nearest Neighbor Graph (NNG) algorithm is a graph and in that, two vertices 'f,' and an edge connect 'b', if the distance between the two vertices is in a small range. Using Nearest Neighbor Graph (NNG) algorithm, Defer Merging and Embedding (DME) selects the nearest neighbor based on the minimum distance.

The DME algorithm sets up to internal nodes embedding deferment for a long possible time within a given topology.

1) There is a locus of feasible locations for subtrees



recursively combined roots which are computed with the bottom-up phase.

2) A top-down phase decides to embed with accurate clock tree internal nodes

### **Bottom up phase**

The concept of bottom-up phase is described here: A possible placement set of ‘j’ represented with segment is merged by ‘G’ with each node ‘j’. Depending on segment merging of two children, the nodes merging segment takes place and it follows the order of bottom-up process for the connection topology—the case of Each edge G is assigned with the length for the merging segments in building the tree. For the embedding of G, the retaining of length is done.

### **Top-down phase**

With the manner of a top-down approach, internal nodes are embedded exactly with the construction of Tree of segments.

In topology ‘G’ containing the node ‘j,’ the cases arise 1) in the merged segment; any point is chosen, when ‘j’ is the root node. 2) In merge segment (j) with a distance or less placement of (j) ie, parent p, a placement is selected as any point, if other than the root internal node ‘v’ gets attributed. Finally, Tree with internal nodes attributes the exact location within this phase.

From the above two phases and Nearest Neighbor Graph (NNG), an efficient clock tree topology has been constructed for synthesis purposes.

#### **4.5.3 Buffering and embedding process**

The final process of the proposed framework is buffering and embedding processes. Buffering and embedding processes are used to measure the total power consumption and timing parameters. In this phase, thereby, for all nodes, the exact location is determined along with TSV [37] for the buffers. The tree topology T and sink location sets are considered for embedding. The TSV number [38] and the wirelength in total are minimized. Further, their weighted sum and a clock tree construction are needed for T and G.i.e. the quantity is minimized [Equation (9)].

$$w_1 \cdot nv(T) + w_2 \cdot \sum_{v_i} |e_i| \quad (9)$$

Where, and represent the weight vectors. From the buffering and embedding processes, the total power dissipation and timing parameters are found in the proposed 3D-IC fabrication.

## **5 Results and Discussion**

In this section, the experimental results of the proposed work are demonstrated. The performance of the proposed work has been evaluated with the existing methods in terms of wirelength [39]–[41], area, delay, power, and skew. It is clear from the comparison results that the proposed work has achieved better performance compared to the existing approaches like Genetic Algorithm based Partitioning (GAP), Electrostatic Placement (EP) and Stochastic Evaluation Algorithm (SEA). This section has further been divided into four sub-sections: simulation environment, performance metrics, simulation results, and result comparison.

### **5.1 Simulation environment**

This section describes a simulation environment where the scalability and the functionality of the proposed work are estimated. The proposed work has been done in MATLAB R2019b tool. The IBM dataset has been used for 3D IC fabrication. The dataset used consists of three files and shown as .net, .netD and .are files. The Matlab program is coded with coding lines and the input is received from the benchmark file. It is followed by extracting the macros, nets, area, and other details. These extracted features are formed into matrix and saved. The macros get scattered in relative space. It demonstrates that the macros distribute one another. Other parameters are plotted to depict the position of the blocks and they are placed along with determining the congestion of the block arrangement. The White Space allocation is done in unused areas between macroblocks, and they are identified and plotted as well as the Through Silicon Vias has been placed and positioned. The Latency buffering rate has been calculated and the number of Skew Cells has been plotted in x-direction and y-directions.

#### **5.1.1 Dataset description**

In this section, dataset parameters that are used in the

implementation of the proposed work are explained. The IBM dataset has been derived from the ISPD06 HB benchmark set. The benchmarks are with a defined range from 500 to 2000 blocks in size along with soft and hard macros. Here, each benchmark in the content of IBM-HB has been converted as follows to reproduce difficult instances: 100% inflation of the largest macro and thus, leads to reducing the total area of the cell of other remaining macros. The data set of the proposed work contains movable modules: cell and macro, nets, largest area, and the ratio of the largest and the smallest areas (Table 1).

**Table 1:** Dataset description

Circuit Name	Movable Modules		Nets	Area (l)	Area(l)/Area (S)
	Cell	Macro			
ibm-HB 01	0	911	5829	6.4	8416
ibm-HB 02	0	1471	8508	11.3	3004.3
ibm-HB 03	0	1289	10279	10.8	33088
ibm-HB 04	0	1584	12456	9.2	13296.5
ibm-HB 05	0	749	9963	13.6	18173.8
ibm-HB 06	0	1120	15047	4.8	399.5
ibm-HB 07	0	1269	16075	12.1	50880
ibm-HB 08	0	1113	18913	5.4	29707
ibm-HB 09	0	1595	27508	4.8	71299
ibm-HB 10	0	1497	27477	4.5	9902.3
ibm-HB 11	0	1233	26320	6.4	74256
ibm-HB 12	0	954	27011	4.2	33088
ibm-HB 13	0	1635	43062	2.0	17860
ibm-HB 14	0	1412	52779	11.0	62781.3
ibm-HB 15	0	1091	47821	1.9	31093
ibm-HB 16	0	1442	56517	0.9	12441
ibm-HB 17	0	943	42200	1.0	3384

## 5.2 Performance metrics

In this section, the performance metrics that are used to evaluate the performance of the proposed work are described and the performance metrics are wirelength, latency, area, power and skew.

### 5.2.1 Wirelength

In 3D-IC physical design, wirelength is one of the important parameters and it is described as the length of the wire on electronic chip connected between the components of the fabricated circuit. Wirelength ' $w_l$ ' is expressed as follows [Equation (10)]:

$$w_l = \sum_{i=1}^n \frac{a_{ci}}{w_{ci}} \quad (10)$$

Where ' $w_l$ ' denotes area of the circuit and ' $w_l$ ' denotes width of the circuit.

### 5.2.2 Area

To decide the 3D-IC size, area parameter is very essential to be considered, during the physical design strategy of chip fabrication. Area of the designed chip is measured using length and width of the circuit. Area is expressed as follows [Equation (11)]:

$$H_a = \sum_{k=1}^n (H_{lk} \times H_{wk}) \quad (11)$$

Where ' $a_{ci}$ ' is denoted as length of the 3D-IC and ' $w_{ci}$ ' is represented as width of the 3D-IC.

### 5.2.3 Latency

Latency is the major parameter to be considered in 3D-IC fabrication. Hence, increase in signal travel from sender to receiver leads to decrease the efficiency of fabricated chip. Latency can be described as the time taken to signal travel from source to the destination. Latency ' $L_c$ ' of the chip is defined as follows [Equation (12)]:

$$L_c = P_d + S_d \quad (12)$$

Where,  $P_d$  represents the propagation delay and  $S_d$  represents serialization delay.

### 5.2.4 Power

In 3D-IC design, reduction of power consumption plays the major factor. Power parameter is demonstrated as the amount of energy spent at particular function. Power (P) can be expressed as follows [Equation (13)]:

$$P = \frac{e_c}{T} \quad (13)$$

Where,  $e_c$  denotes energy spent at each function and  $T$  represents time.

### 5.2.5 Skew

In chip fabrication, clock tree synthesis process is

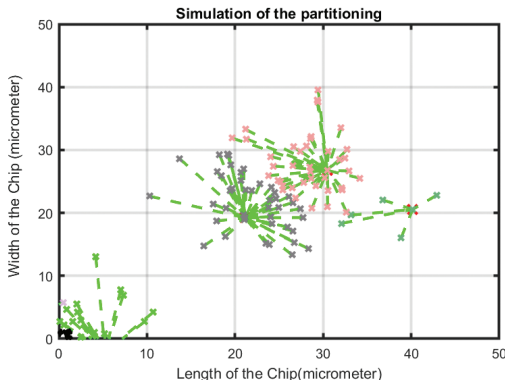


Figure 5: Simulation result of partitioning.

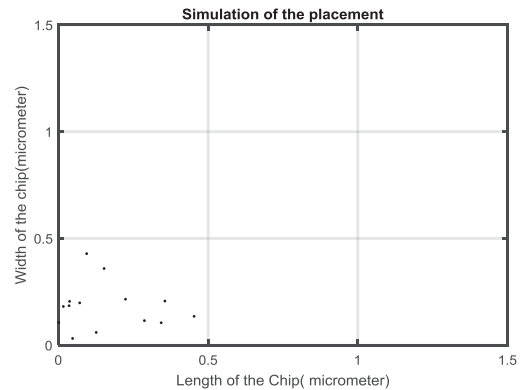


Figure 6: Simulation result of placement.

essential during physical design [42]–[46]. Hence, it reduces the skew of the generated chip. Skew factor is described as the phenomenon in which the same source signal arrives at different components at different time.

### 5.3 Simulation results

This section elaborates the simulation results of physical design strategy of 3D-IC fabrication processes which include partitioning, placement, routing, clock tree topology and buffering and embedding process.

#### 5.3.1 Partitioning

Partitioning is the first process in the simulation environment [47], [48]. Partitioning is implemented using hybrid algorithm that is the combination of Chemical Reaction Optimization (CRO) and K-Medoid algorithms. IBM dataset has been provided as input to the simulation tool.

The Figure 5, depicts the simulation result of the proposed hybrid algorithm based partitioning approach that partitions the netlist effectively using the combination of Chemical Reaction Optimization (CRO) algorithm and K-Medoid algorithms.

#### 5.3.2 Placement

After implementing partitioning, placement process has been executed. In placement, the macros are placed without any congestion using the proposed technique Grid Warping Technique (GWT).

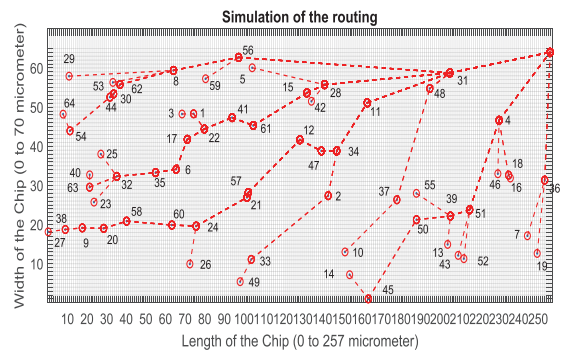


Figure 7: Simulation result of routing.

The Figure 6, depicts the simulation result of proposed grid warping placement. It portrays the macros that are placed in the partitioned area and it locates macros without any overlapping.

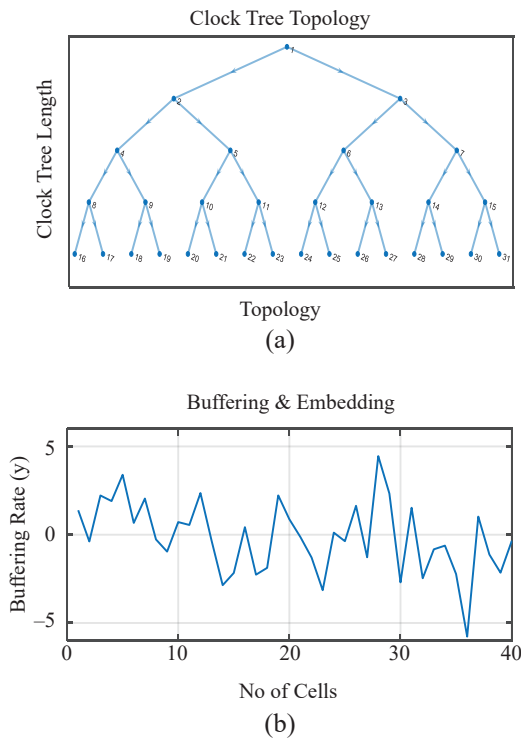
#### 5.3.3 Routing

Routing is executed, after the completion of placement process and it reduces the wirelength optimally using the proposed Line Search algorithm.

Figure 7, illustrates the simulation result of Line Search algorithm based routing. The proposed Line Search based routing discovers paths without increasing wirelength parameter and power consumption.

#### 5.3.4 Clock tree synthesis

Clock tree synthesis process is important to minimize skew and latency of the proposed 3D-IC physical



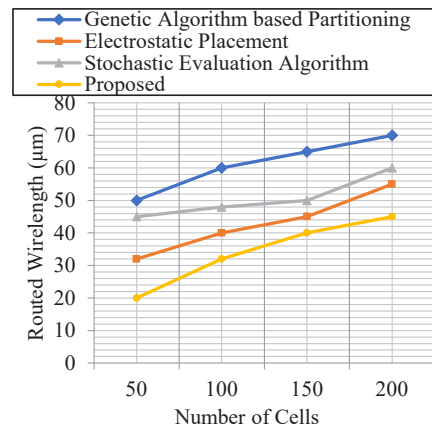
**Figure 8:** (a) Simulation result of clock tree topology  
(b) Simulation result of buffering and embedding.

design. The clock tree synthesis of the present work first clusters the sink nodes that contain high density and builds the clock tree topology.

The Figure 8(a) and (b) illustrates the clock tree topology and buffering and embedding processes of the proposed work. The clock tree topology is constructed using DME and NNG algorithms and they play vital role in the reduction of skew and latency. Buffering and embedding processes measure the total power consumption and the timing parameter.

#### 5.4 Results comparison

In this section, the comparison of the proposed work with the existing works, which include Genetic Algorithm based Partitioning (GAP), Electrostatic Placement (EP) and Stochastic Evaluation Algorithm (SEA), has been expressed. From this comparison, it can be noted that the proposed framework is more efficient than the existing approaches like Genetic Algorithm based Partitioning (GAP), Electrostatic Placement (EP) and Stochastic Evaluation Algorithm (SEA).



**Figure 9:** Comparison of wirelength.

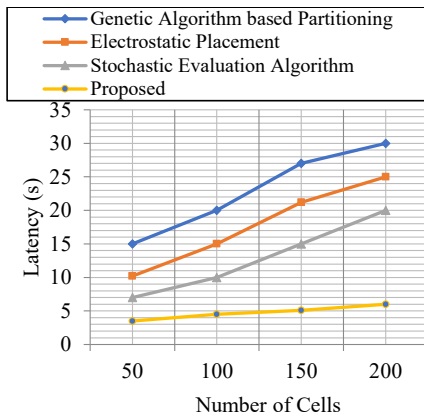
##### 5.4.1 Comparison of wirelength

Wirelength parameter plays a vital role in the size of 3D-IC fabrication. If the wirelength is increased, then automatically 3D-IC size also increases. Hence, wirelength parameter is essential in 3D-IC fabrication. The wirelength parameter of the proposed work is compared with the existing approaches like Genetic Algorithm based Partitioning (GAP), Electrostatic Placement (EP) and Stochastic Evaluation Algorithm (SEA).

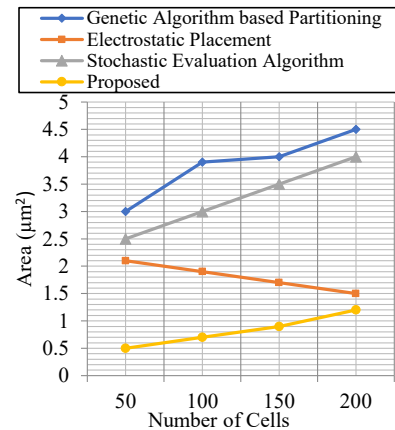
From Figure 9, comparison graph, it can be elaborated that the wirelength of the proposed work is less compared to other existing approaches like Genetic Algorithm based Partitioning (GAP), Electrostatic Placement (EP) and Stochastic Evaluation Algorithm (SEA). Hence, GWT based placement and LS based routing have been proposed. The Grid warping technique (GWT) based placement places macros without congestion and overlapping whereas LS routing explores the path with the minimum distance. Consequently, the utilized algorithms work efficiently to minimize the wirelength and in turn, reduces the size of the 3D-IC whereas in the existing methods like Genetic Algorithm based Partitioning (GAP), Electrostatic Placement (EP) and Stochastic Evaluation Algorithm (SEA), the wirelength increases, due to their implemented framework.

##### 5.4.2 Comparison of latency

Latency is a necessary metric to measure the physical



**Figure 10:** Comparison of latency.



**Figure 11:** Comparison of area.

design strategy of 3D-IC fabrication. The time taken to signal travel from transmitter to receiver is known as latency. It is measured using the addition of propagation delay and serialization delay.

From the Figure 10, it can be portrayed that the proposed work achieves minimum latency in signal transformation between the components of 3D-IC. The maximum delay of the proposed work is 5.2 s and the minimum delay is 3.0 s. It has happened, since the algorithm based routing has been proposed. It finds the minimum path between two components and in turn, reduces the delay in 3D-IC. In order to overcome the problem of time and amount of memory, line search algorithm has been proposed and it reduces the time and the amount of memory for storing the information in routing. In line search algorithm, the source and the destination are routed by generating horizontal and vertical lines for both source and destination terminals. If the horizontal and vertical lines of source and destination intersect, then the path is found and the routing is done between the source and the destination. The existing approaches like Genetic Algorithm based Partitioning (GAP), Electrostatic Placement (EP) and Stochastic Evaluation Algorithm (SEA) have the largest delay compared to the proposed work, due to their physical design implementations. The EP method has maximum delay in signal transformation between components in 3D-IC, due to its poor routing and placement.

#### 5.4.2 Comparison of area

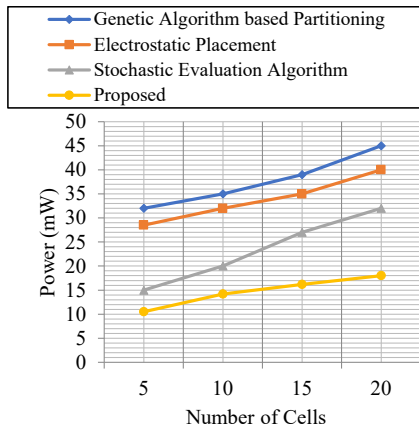
In the physical design of 3D-IC, area parameter is an

essential factor, since area defines the size of the chip. Moreover, the compact size of the 3D-IC is the major aspect to be considered in all applications like portable devices. Area metric is computed using the length and width of the fabricated chip.

From the comparison of Figure 11, it can be depicted that the proposed framework contains less area compared to the other existing methods like Genetic Algorithm based Partitioning (GAP), Electrostatic Placement (EP) and Stochastic Evaluation Algorithm (SEA). Since Grid warping technique (GWT) based placement has been introduced, it effectively reduces the area of the circuit; because it moves only warping surface not the macros. Line Search algorithm based routing minimizes the wirelength while discovering minimum path between components. In the above comparison, it is evident that Genetic Algorithm based Partitioning (GAP) method has high area compared to other techniques like Electrostatic Placement (EP) and Stochastic Evaluation Algorithm (SEA). These methods have not concerned about wirelength factor and hence, their area metric increases dramatically.

#### 5.4.4 Comparison of power

Power consumption is the major issue in physical design strategy of 3D-IC manufacturing. The increase in power feature may lead to decrease the working efficiency of the chip and generate poor communication between components. In turn, the overall process of the 3D-IC is affected.



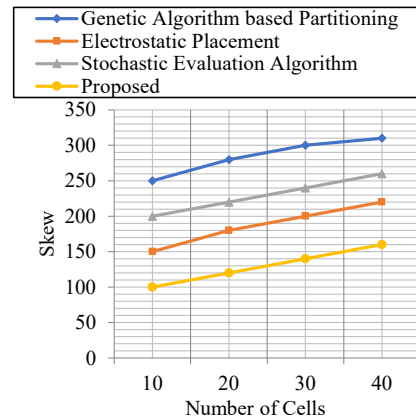
**Figure 12:** Comparison of power.

The comparison shown in Figure 12, elaborates that the power consumption of the proposed work is less compared to the other existing methods including Genetic Algorithm based Partitioning (GAP), Electrostatic Placement (EP) and Stochastic Evaluation Algorithm (SEA). The maximum power consumption of the proposed work is 18 W and the minimum power consumption of the proposed work is 10.5W. The power consumption has been reduced because of the proposed Density Sorting (DS) based clustering and Line Search based routing. Density Sorting (DS) based clustering method clusters the sink nodes with high density to minimize the power dissipation. Line Search based routing finds the minimum path between the sender and the receiver to minimize the power consumption and wirelength. In the above comparison, it can be seen that Genetic Algorithm based Partitioning (GAP) method consumes more power compared to others, due to its routing and it has not concerned about clock tree synthesis.

#### 5.4.5 Skew

Clock tree synthesis process has been implemented to minimize the skew and latency, since these factors play major roles in the flexibility of 3D-IC. Skew is a significant parameter in 3D-IC fabrication and it reduces the output signal quality.

The comparison of Figure 13, shows that the proposed work minimizes the skew value compared to the other existing methods like Genetic Algorithm based Partitioning (GAP), Electrostatic Placement (EP)



**Figure 13:** Comparison of skew.

and Stochastic Evaluation Algorithm (SEA), since an efficient clock tree synthesis with Defer Merging and Embedding (DME) and Nearest Neighbor Graph (NNG) algorithm based methods has been proposed and it reduces the skew optimally. The high-density sink nodes are clustered using DS algorithm which decreases the skew and latency in the proposed work. From the above comparison, it can be concluded that Genetic Algorithm based Partitioning (GAP) method has the highest skew compared to other methods like Electrostatic Placement (EP) and Stochastic Evaluation Algorithm (SEA), since Genetic Algorithm based Partitioning (GAP) method has not considered the clock tree synthesis process in physical design of 3D-IC fabrication.

## 6 Conclusions

3D-IC physical design strategy is one of the emerging technologies that create large amount of smart and portable devices. This present work has proposed 3D-IC fabrication with reduced size, wirelength, skew and power consumption. The proposed framework consists of four major parts in 3D-IC fabrication and it includes partitioning, placement, routing and clock tree synthesis. Partitioning process is executed using hybrid algorithm that combines Chemical Reaction Optimization (CRO) and K-Medoid algorithm. Using the hybrid algorithm, the proposed approach partitions the netlist into multiple layers effectively. In the placement process, Grid Warping Technique (GWT) is pursued to locate macros effectively in

particular area. There are multiple stages to place macros in specific area without any congestion and overlapping. In routing, LS algorithm is utilized and that effectively finds path with reduced wirelength. In clock tree synthesis, multiple TSV has been considered and it contains three stages that include sink node clustering, clock tree topology and buffering and embedding. In the first stage, sink node with high density has been clustered using Density Sorting (DS) algorithm. After completing clustering, clock tree topology is constructed using Defer Merging and Embedding (DME) and Nearest Neighbor Graph (NNG) methods that construct tree effectively. In the final stage of the proposed work is buffering and embedding that is used to measure the total power dissipation and timing parameter. The proposed work has been compared with the existing works by considering metrics such as wirelength, latency, area, power and skew. From the comparison, it is concluded that the proposed framework works better than the existing works like Genetic Algorithm based Partitioning (GAP), Electrostatic Placement (EP) and Stochastic Evaluation Algorithm (SEA).

In future, the works may be directed towards addressing the tradeoffs for total wirelength, whitespace area and aspect ratio. Further the work has been planned to improve clock synthesis networks with multiple corners and multiple modes using clock scheduling.

## References

- [1] R. K. R. Nair, S. Pothiraj, and T. R. R. Nair “A novel optimization approach for partitioning-based place and route in 3D integrated circuits,” *The International Journal of Electrical Engineering & Education*, Jun. 2020, doi: 0020720920930346.
- [2] O. Adekomaya, T. Jamiru, E. R. Sadiku, and A. A. Adediran, “Sustainability of high temperature polymeric materials for electronic packaging applications,” *Applied Science and Engineering Progress*, vol. 11, no. 3, pp. 217–224, 2018.
- [3] Z. Wang, S. Kanwal, L. Wang, and A. Chattopadhyay, “Automated high-level modeling of power, temperature and timing variation for microprocessor,” *Applied Science and Engineering Progress*, vol. 10, no. 3, pp. 163–175, 2017.
- [4] J. Barreiro-Gomez, C. Ocampo-Martinez, and N. Quijano, “Partitioning for large-scale systems: A sequential distributed MPC design,” *IFAC Papersonline*, vol. 50, no.1, pp. 8838–8843, 2017.
- [5] S. M. Sait, F. C. Oughali, and M. Al-Asli, “Design partitioning and layer assignment for 3D integrated circuits using tabu search and simulated annealing,” *Journal of Applied Research and Technology*, vol. 14, no. 1, pp. 67–76, 2016.
- [6] K. Manna, V. S. S. Teja, S. Chattopadhyay, and I. Sengupta, “TSV placement and core mapping for 3D mesh based network-on-chip design using extended Kernighan-Lin partitioning,” in *2015 IEEE Computer Society Annual Symposium on VLSI*, 2015, pp. 392–397.
- [7] S. D. Lin and D. H. Kim, “Detailed-placement-enabled dynamic power optimization of multi-tier gate-level monolithic 3-D ICs,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 7, pp. 845–854, 2018.
- [8] S. Bechikh, A. Chaabani, and L. B. Said, “An efficient chemical reaction optimization algorithm for multiobjective optimization,” *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2051–2064, Oct. 2015.
- [9] M. T. Islam, P. K. Basak, P. Bhowmik, and M. Khan, “Data clustering using hybrid genetic algorithm with k-means and k-medoids algorithms,” in *23rd International Computer Science and Engineering Conference (ICSEC)*, 2019, pp. 123–128.
- [10] M. Wang, G. Yang, J. Lin, S. Zhang, A. Shamir, S. Lu, and S. Hu, “Deep online video stabilization with multi-grid warping transformation learning,” *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2283–2292, May 2019.
- [11] J. Zhang, “On the non-monotone armijo-type line search algorithm,” in *Third International Conference on Information and Computing, Wuxi*, 2010, pp. 308–311.
- [12] R. Zhang, X. Wei, and T. Watanabe, “A sorting-based IO connection assignment for flip-chip designs,” in *IEEE 10th International Conference on ASIC*, 2013, pp. 1–4, doi: 10.1109/ASICON.2013.6811927.
- [13] H. Huang, W. Luk, W. Zhao, and X. Zeng, “DME-based clock routing in the presence of obstacles,” in *2007 7th International Conference on ASIC*,

- 2007, pp. 1225–1228.
- [14] M. R. Trad, A. Joly, and N. Boujemaa, “Large scale KNN-graph approximation,” in *IEEE 12th International Conference on Data Mining Workshops, Brussels*, 2012, pp. 439–448.
- [15] P. Sivaranjani and G. Srividhya, “3D IC partitioning by using fast force directed simulated annealing algorithm for reducing through-silicon vias,” *International Journal of Advanced Trends in Engineering and Technology*, vol. 1 no. 2, pp. 6–11, 2016.
- [16] S. Banerjee, S. Majumder, and B. B. Bhattacharya, “A graph-based 3D IC partitioning technique,” presented at 2014 *IEEE Computer Society Annual Symposium on VLSI*, Florida, USA, Jul. 9–11, 2014.
- [17] C. Jang and J. Chong, “Thermal-aware floor planning with min-cut die partition for 3D ICs,” *ETRI Journal*, vol. 36, no. 4, pp. 635–642, 2014
- [18] H. Shanavas and R. K. Gnanamurthy, “Optimal solution for vlsi physical design automation using hybrid genetic algorithm,” *Mathematical Problems in Engineering*, vol. 2014, pp. 1–15, 2014.
- [19] C. Serafy and A. Srivastava, “TSV replacement and shield insertion for TSV–TSV coupling reduction in 3-D global placement,” *IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 4, pp. 554–562, 2015.
- [20] K. Athikulwongse and S. K. Lim, “Exploiting die-to-die thermal coupling in 3-D IC placement,” *IEEE Transactions On Very Large Scale Integration (Vlsi) Systems*, vol. 22, no. 10, pp. 2145–2155, 2014.
- [21] K.-C. Chen, S.-Y. Lin, H.-S. Hung, and A.-Y. Wu, “Topology-aware adaptive routing for non-stationary irregular mesh in throttled 3D NoC systems,” *IEEE Transactions On Parallel And Distributed Systems*, vol. 24, no. 10, pp. 2109–2120, 2017.
- [22] P. Ghosal, S. Das, and A. Das, “A new class of obstacle aware steiner routing in 3D integrated circuits,” *Advances in Computing and Information Technology*, vol. 178, pp. 697–706, 2013.
- [23] M. Ebrahimi, M. Daneshalab, P. Liljeberg, J. Plosila, J. Flich, and H. Tenhunen, “Path-based partitioning methods for 3D networks-on-chip with minimal adaptive routing,” *IEEE Transactions on Computers*, vol. 63, no. 3, pp. 718–733, 2014.
- [24] Y. Cai, C. Deng, Q. Zhou, H. Yao, F. Niu, and C. N. Sze, “Obstacle-avoiding and slew-constrained clocktree synthesis with efficient buffer insertion,” *IEEE Transactions on Very Large Scale Integration (Vlsi) Systems*, vol. 23, no. 1, pp. 142–155, 2014.
- [25] W. Liul, H. Dul, Y. Wangl, Y. Ma, Y. Xie, J. Quan, and H. Yangl, “TSV-aware topology generation for 3D clock tree synthesis,” in *14th Int'l Symposium on Quality Electronic Design*, 2013, pp. 300–307.
- [26] W. Liu, H. Dul, Y. Wang, Y. Ma, Y. Xie, J. Quan, and H. Yang, “Whitespace-aware TSV arrangement in 3-D clock tree synthesis,” *IEEE Transactions on Very Large Scale Integration (Vlsi) Systems*, vol. 23, no. 9, pp. 1842–1853, 2015.
- [27] K. Maragos, K. Siozios, and D. Soudris, “A genetic algorithm based partitioning for 3-D reconfigurable architectures,” presented at *HiPEAC Workshop on Reconfigurable Computing (WRC)*, Netherlands, 2015.
- [28] J. Lu, H. Zhuang, and I. Kang, “ePlace-3D: Electrostatics based placement for 3D-ICs,” in *ISPD '16 Proceedings of the 2016 on International Symposium on Physical Design*, 2016, pp. 11–18.
- [29] S. M. Sait and U. F. Siddiqi, “A stochastic evolution algorithm based 2D VLSI global router” *Journal Integration, the VLSI Journal archive*, vol. 53, pp. 115–125, 2016.
- [30] L. Chandrakar, S. Ravi, and H. M. Kittur, “Power and obstacle aware 3D clock tree synthesis,” *Electrical and Electronic Engineering*, pp. 1–13, 2017, doi: 10.20944/preprints201705.0056.v1.
- [31] S. S. Gill, R. Chandel, and A. Chandel, “Genetic algorithm based approach to circuit partitioning,” *International Journal of Computer and Electrical Engineering*, vol. 2, no. 2, pp. 196–202, 2010.
- [32] N. Y. Aung, K. K. San, and S. Z. Hlaing, “Hybrid partition around medoids algorithm for large volume of data,” in *2019 International Conference on Advanced Information Technologies (ICAIT)*, 2019, pp. 280–285.
- [33] X. Zhong, J. D. Ma, S. M. Fowler, and R. A. Rutenbar, “Large-scale placement by grid-warping,” in *Proceedings of the 41st annual Design Automation Conference*, pp. 351–356. 2004.
- [34] J. Hightower and G. Borriello, “Location sensing techniques,” *IEEE Computer*, vol. 34, no. 8,



- pp. 57–66, 2001.
- [35] A. Jayadev, A. Jafarpour, A. Orlitsky, and A. T. Suresh, “Sorting with adversarial comparators and application to density estimation,” in *2014 IEEE International Symposium on Information Theory*, 2014, pp. 1682–1686.
- [36] X. Zhao, J. Minz, and S. K. Lim, “Low-power and reliable clock network design for through-silicon via (TSV) based 3D ICs,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 1, no. 2, pp. 247–259, 2010.
- [37] S. Osmolovskiy, J. Knechtel, I. L. Markov, and J. Lienig, “Optimal die placement for interposer-based 3D ICs,” in *23rd Asia and South Pacific Design Automation Conference*, pp. 513–520, Jan. 2018.
- [38] R. K. Dash, J. L. Risco-Martin, and A. K. Turuk, “A thermal driven genetic algorithm for three dimensional network-on-chip systems,” in *SCSC’16 Proceedings of the Summer Computer Simulation Conference*, 2016, vol. 47, pp. 1–8.
- [39] É. Lepercq, Y. Blaquièrre, and Y. Savaria, “A pattern-based routing algorithm for a novel electronic system prototyping platform,” *Integration*, vol. 62, pp. 224–237, 2018.
- [40] Y.-H. Fan, “Fast routing verification with complexity effect for SOC,” *Applied Computing and Informatics*, vol. 15, no. 2, pp. 144–152, 2019.
- [41] S. R. Jain and K. Okabe, “Training a fully convolutional neural network to route integrated Circuits,” *Computer Vision and Pattern Recognition*, 2017, doi: arXiv:1706.08948.
- [42] R. Ewetz and C.-K. Koh, “Fast clock scheduling and an application to clock tree synthesis,” *Integration the VLSI Journal*, vol. 56, pp. 115–127, 2016.
- [43] K. S. Kumar and J. Reuben, “Minimal buffer insertion based low power clock tree synthesis for 3D integrated circuits,” *Journal of Circuits, Systems, and Computers*, vol. 25, no. 11, 2016, doi: 10.1142/0218126616501425.
- [44] F.-W. Chen and T. Hwang, “Clock-tree synthesis with methodology of reuse in 3D-IC,” *ACM Journal on Emerging Technologies in Computing Systems*, vol. 10, no. 3, pp. 1–22, Apr. 2014.
- [45] T. Lu and A. Srivastava, “Gated low-power clock tree synthesis for 3D-ICs,” in *Proceedings of the 2014 International Symposium on Low Power Electronics and Design*, pp. 319–322, 2014.
- [46] H. Huang, W.-S. Luk, W. Zhao, and X. Zeng, “DME-based clock routing in the presence of obstacles,” in *7th International Conference on ASIC*, 2007, pp. 1225–1228.
- [47] W. Jakawat and R. Makkhongkaew, “Graph clustering with K-nearest neighbor constraints,” in *16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 309–313, 2019.
- [48] S. S. Gill, R. Chandel, A. Chandel, “Genetic algorithm based approach to circuit partitioning,” *International Journal of Computer and Electrical Engineering*, vol. 2, no. 2, pp. 196–202, 2010.