Monolithic 3D (M3D) is an emerging heterogeneous integration technology that overcomes the limitations of the conventional through-silicon-via (TSV) and provides significant performance uplift and power reduction. However, the ultra-dense 3D interconnects impose significant challenges during physical design on how to best utilize them. Besides, the unique low-temperature fabrication process of M3D requires dedicated design-for-test mechanisms to verify the reliability of the chip. In this article, we provide an in-depth analysis on these design and test challenges in M3D. We also provide a comprehensive survey of the state-of-the-art solutions presented in the literature. This article encompasses all key steps on M3D physical design, including partitioning, placement, clock routing, and thermal analysis and optimization. In addition, we provide an in-depth analysis of various fault mechanisms, including M3D manufacturing defects, delay faults, and MIV (monolithic inter-tier via) faults. Our design-for-test solutions include test pattern generation for pre/post-bond testing, built-in-self-test, and test access architectures targeting M3D.

CCS Concepts: • Hardware → 3D integrated circuits; Physical design (EDA); Design for testability;

Additional Key Words and Phrases: Monolithic 3D IC, design and simulation, design-for-test

1 INTRODUCTION AND MOTIVATION

Today’s conventional two-dimensional (2D) integrated circuits (ICs) are confronted with challenges associated with high costs, scalability, performance, and high power consumption. The scaling of Complementary Metal Oxide Semiconductor (CMOS) technology is approaching the physical limits as predicted by Moore’s Law [68], which leads to high cost in integrating more transistors in a given 2D floorplan. In addition, machine learning and other multi-threading applications require the integration of more logic cores in a single chip, but the reticle size limits the
maximum area of a 2D IC that can be fabricated. Moreover, long signal routing wires in 2D ICs can cause timing degradation and an increase in switching power. Emerging 3D IC technologies provide an alternative solution to address these challenges and implement the next-generation ICs.

Various 3D integration technologies have been proposed in recent years, including Through-Silicon-Via (TSV) based 3D [57], Face-to-Face (F2F) bonded 3D [60], and Monolithic 3D (M3D) [6]. Among them, the TSV-based 3D ICs provide a reliable solution to integrate multiple dies in 3D, but TSVs also introduce high silicon area overhead and large parasitic resistance and capacitance (RC) due to their large size (usually 3–10 μm). F2F-bonded 3D ICs utilize Cu-Cu direct bonding to establish vertical connections between the back-end-of-line (BEOL) layers of two dies [50]. The F2F bonding structure provides a small vertical connection pitch (around 1 μm [51]), but it allows only two-die integration, and the electrical and thermal coupling issues also become exacerbated in F2F 3D ICs due to the bonded BEOL layers [78, 79, 84]. M3D ICs, however, offer the highest connection density and flexibility compared with the other 3D integration technologies. An M3D IC uses Monolithic Inter-tier Vias (MIVs) for vertical connections, whose size is typically less than 100 nm [21], similar to regular vias in the BEOL layers. These small MIVs introduce negligible RC delays and silicon area overhead and enable very high 3D connection density (up to 100 million/mm² in a 14 nm technology node) [93]. Therefore, M3D technology offers the greatest potential to exploit the third dimension for IC design, with potential cost reduction compared with the 2D baseline [38]. We will focus on this monolithic 3D technology in the article.

To implement commercial-grade monolithic 3D ICs, we need a comprehensive Electrical Design Automation (EDA) tool flow. Several major challenges need to be resolved in developing such a tool flow: (1) Most of the existing commercial EDA tools are designed for 2D ICs and do not support 3D IC implementation natively. (2) Placement and clock tree design need to be done with an awareness of the 3D partitioning. (3) Sign-off verification, including timing, power, and thermal analysis should be performed on the entire 3D system to verify the reliability. In this article, we summarize the state-of-the-art techniques in EDA development for M3D ICs.

The first step in M3D fabrication involves a standard high-temperature process to integrate the transistors and interconnects in the bottom tier. A thin inter-layer dielectric is then created over the bottom tier, and low-temperature molecular bonding of the silicon-on-insulator (SOI) substrate is used to obtain the top tier [7, 99]. The MIVs are finally fabricated to connect the top and bottom tiers. The above steps are repeated for the fabrication of additional tiers.

Dense M3D integration can lead to routing congestion in the bottom tier, increasing wire length and making thermal hotspots more likely [72]. As the device tiers are sequentially fabricated in situ, care must be taken so that the high-temperature fabrication steps (e.g., dopant activation) of the thin upper layer do not damage the underlying interconnects and degrade the bottom-tier transistors [14, 22]. Therefore, different tiers of an M3D IC can exhibit various types of defects or process variations. Tier-level fault localization is required for providing feedback to the foundry to optimize the fabrication steps and enable yield learning.

High integration density and aggressive scaling of the inter-layer dielectric make MIVs especially prone to defects such as shorts and opens [54, 55]. MIV testing is therefore needed to ensure effective defect screening and quality assurance. While MIVs can conceivably be tested together with the M3D logic/memory tiers, fault localization, defect isolation, and yield learning require a test solution that can exclusively target the MIVs in an M3D IC. Design-for-testability (DfT) for MIVs is therefore a promising step towards this direction. In this article, we summarize the state-of-the-art methods for testing M3D designs and highlight important considerations to be taken into account for future research in DfT for M3D ICs.

This article is a comprehensive tutorial on M3D IC design and test. We provide an in-depth analysis on the challenges of physical implementation and design-for-test in M3D ICs. We present
an overview of the state-of-the-art physical design methodologies and DfT mechanisms. We look into each stage of the design flow and highlight the challenges for M3D ICs. After comparing the existing approaches and demonstrating their pros and cons, we propose open problems for future studies on EDA development for M3D ICs. The article is organized as follows:

In Section 2, we present challenges and solutions for physical design of M3Ds in four main stages: partitioning, placement, clock network design, and thermal analysis. We first compare gate-level vs. block-level partitioning approaches and show how the design information is taken into consideration in the partitioning stage. Next, we look into the development of 3D placement techniques, including pseudo-3D placers and true 3D placers. Next, we discuss the challenges of 3D clock delivery network (CDN) design and provide a survey of existing 3D CDN design tools. Last, we present thermal analysis and optimization approaches for M3D.

Section 4 highlights the different sources of non-linear process-parameter variations in M3D ICs and discusses the impact of inter-tier coupling and void defects on the testing of small-delay defects. Section 5 lists the fault models and specialized low-cost DfT architectures for the targeted testing of MIVs and different tiers in the M3D design.

Section 6 discusses research opportunities towards the development of efficient EDA tools for the design and test of commercial-quality M3D ICs. We conclude in Section 7.

2 EDA SOLUTIONS FOR MONOLITHIC 3D ICS

2.1 Tier Partitioning

Tier partitioning is a critical step in M3D IC implementations, because it has huge impacts on the following procedures and the performance of the final design. In this step, we separate the components in the original design into different tiers. There are three flavors of partitioning schemes: transistor-level, gate-level, and block-level. Transistor-level partitioning means partitioning each standard cell and dividing PMOS and NMOS into separated tiers, which does not require a specialized tier-partitioning algorithm. Therefore, we focus on the gate-level and block-level partitioning methods in this article.

The block-level methods regard the functional blocks as a basic unit during partitioning, which allow the blocks to be separated into different tiers, but require the standard cells within each block to remain on the same tier. These methods have lower computation resource requirement due to the small number of functional blocks and preserve the design hierarchy information during partitioning, but they tend to result in a low number of 3D nets (those nets with pins on different tiers) and might not fully utilize the capacity of high-density 3D technology like M3D. However, the gate-level methods allow the standard cells in the design to be freely allocated to different tiers, which increase the flexibility in partitioning, but propose challenges in limiting the number of 3D nets and determining the connection locations. In the following subsections, we will first look at the impact of 3D connectivity on system performance and then discuss a few state-of-the-art gate-level and block-level partitioning methods and compare their pros and cons. Previous studies on 3D tier partitioning and floorplanning include References [8, 42, 88], and so on.

2.1.1 Recursive Partitioning and the Impact of 3D Connectivity. The M3D technology provides a sufficiently high 3D interconnect density, which removes the limitation of 3D connectivity from the fabrication perspective. It becomes important to understand the impact of 3D net numbers on the system performance and to optimize the 3D nets for specific designs. The partitioning solutions have significant impacts on the total wirelength in a 3D system, which in turn affect the RC delay of the interconnects and the system performance. Therefore, we use the total wirelength as a matrix to evaluate the partitioning solutions.

Recursive partitioning is a method widely used for 2D IC placement, while we can also utilize it for 3D tier partitioning with minor modifications [75]. The original method uses V-cuts (cut
Fig. 1. The concept of the bin-based tier-partitioning method [72]. The left figure shows a projected 2D placement on half the area. The right figure shows the min-cut partitioning is being done within each rectangular bin.

along the y-axis) and H-cuts (cut along the x-axis) recursively to divide a 2D design into smaller sub-regions. For two-tier 3D ICs, we add a Z-cut to separate the cells in each sub-region to different tiers. We use a sequence of cuts to partition the entire design, which may look like \(H, V, H, V, \ldots, Z, \ldots, V\). The order of the cut sequence has an impact on the 3D net number, because the number of H-cuts and V-cuts before the Z-cut determine the size of the sub-region and the number of nets to be divided in the Z direction. Therefore, this method allows us to control the 3D net number.

The authors in Reference [75] use the recursive partitioning method to implement 3D ICs with five benchmark circuits. Although they consider the TSV-based 3D technology, the conclusions could be generalized to M3D ICs. The number of TSVs (3D nets) is adjusted by changing the order of H-cuts, V-cuts, and Z-cut. After partitioning, placement, routing, and timing optimization are performed with commercial tools. They analyze the wirelength and longest path delay in the 5 benchmark circuits to study the impact of partitioning and 3D net number. Results show that as the number of 3D nets increases, the wirelength and longest path delay of the circuits first decrease, but then increase. That is because adding more 3D nets helps remove long 2D interconnects when the 3D net number is small. In TSV-based 3D designs, when the 3D net number is too large, inserting more 3D nets will force tightly coupled cells to be separated into different tiers and introduces extra RC parasitics. Therefore, the trend of performance with increasing 3D connectivity is similar to a U-curve, which means that an optimal point of 3D net number exists to maximize the system performance. The objective of the partitioning algorithm is to approach this optimal point. However, when we apply this method to M3D ICs, the optimal 3D net number will be much larger compared with TSV-based 3D, since the RC parasitics introduced by MIVs are negligible. Also, the recursive partitioning method is not flexible enough to control the 3D net number within each sub-region. Therefore, we need more partitioning algorithms dedicated to 3D tier partitioning to overcome these issues.

2.1.2 Bin-based Min-cut Partitioning. The bin-based min-cut partitioning [72] is a method that allows to control the 3D net number and fulfill the area balance constraint. This method is also part of the Shrunk-2D physical design flow. First, we divide the design into rectangular bins with the given column and row numbers. The bin size is adjustable and has a great impact on the quality of the solution, because the partitioning will be done within each bin.

After that, we use the Fiduccia–Mattheyses (FM) [34] algorithm to perform the min-cut partition for each bin, as shown in Figure 1. The cells in each bin will be separated into two tiers. The difference of cell area in each tier needs to be within a certain limit, which is called the area-balance
constraint. We first create an initial solution that satisfies the constraint by arbitrarily assigning the cells to two tiers. We define the cutsize as the inter-tier connection number, which is equal to the 3D net number. Then the gain of moving one cell from one tier to another is defined as the cutsize decrease. Such a movement is only allowed when moving the cell does not break the area balance in the specific bin. In the beginning, all the cell gains are calculated and stored in sorted buckets. Also, all the cells are marked as “unlocked.” Then, the cell with the highest gain is moved to the other tier, marked as “locked,” and only the gain of its neighboring cells will be updated. This step is repeated until all the cells are locked. The whole process is called a pass and will be repeated multiple times until cutsize cannot be further reduced. Because of the bucket structure and the neighboring gain updating method, the computation complexity of the algorithm is only $O(C)$, where $C$ represents the number of cells. Therefore, this method could provide a partitioning solution efficiently with controllable bin size.

According to the experimental results, the authors in Reference [72] find that the MIV number always decreases with the increasing bin size, because the partitioner has more freedom to minimize the cutsize in a larger bin. However, the bin size has a mixed impact on wirelength and performance. When the bin size is too small, the partitioner has too many area-balance constraints to fulfill and thus tends to result in higher cutsize, which affect the MIV placement and design quality. When the bin size is too large, the partitioning result tends to cause cell displacement and cannot fully utilize the 3D connectivity provided by M3D. The optimal bin size varies with designs, but the results show that a bin size of 10–20 $\mu$m works well across a large range of benchmark designs. The trend is also similar to the U-curve described in Section 2.1.1.

However, the bin-based min-cut partitioning is a gate-level method that works only in a flattened netlist. The design hierarchy and micro-architecture information is completely ignored by the algorithm. Therefore, it could provide sub-optimal solutions for designs with complicated hierarchical structures, such as commercial microprocessors.

2.1.3 Design-aware Partitioning. A design-aware partitioning approach, Cascade-2D, is proposed in Reference [19] and also enhanced in Reference [97]. Unlike the previous methods, the Cascade-2D approach performs the partitioning on the Register-Transfer Level (RTL) designs rather than the gate-level netlists. Therefore, this approach can handle both block-level and gate-level M3D, also consider the RTL-level constraints. The flexibility of the approach allows the users to incorporate any partitioning algorithm and also to manually adjust the tier assignment of any cells or blocks. There are two major ways to perform partitioning using Cascade-2D: (1) by analyzing the connectivity information in existing 2D implementation; (2) based on the design micro-architecture.

The first way utilizes the design information in a 2D implementation to guide the partitioning. Initially, we need to implement the design with the conventional 2D flow using commercial tools. With the 2D implementation database, we perform timing analysis and quantify the number of timing paths between each pair of functional blocks. This number is defined as the degree of connectivity of the block pair, which represents how close the two blocks are coupled from the timing perspective. The cell area of each functional block is also extracted.

Later on, we need to partition the functional blocks into two tiers, with the objective to maximize the degree of connectivity between the two tiers and the constraint to balance the cell area of the two groups. The objective helps separate the pair of blocks with a high degree of connectivity into different tiers. By doing this, it reduces the distance between the functional blocks by replacing the long 2D interconnects with the 3D nets. The maximum degree of connectivity in block-level partitioning is usually much smaller than the one in gate-level partitioning, because it is limited by the number of functional blocks. Therefore, the 3D net number will not exceed the optimal point.
in the *U-curve* of Section 2.1.1, which means that we can optimize the wirelength and timing by maximizing the degree of connectivity.

Figure 2 shows an example of the design-aware partitioning process. For the critical functional block pair like A, B, we pre-partition them into separated tiers and fix the tier assignment. Then, we divide the rest of the blocks into two tiers to maximize the degree of connectivity while satisfies the area-balance constraint. This step can be automatized with the FM algorithm described in Section 2.1.2, while the gain of cell movement needs to be defined as the increase of cutsize. Experimental results [19] show that the Cascade-2D partitioner results in a much smaller MIV number (an order of magnitude lower) compared with the result of the bin-based min-cut partitioner. This tends to lead to a slightly higher wirelength, but it helps reduce the total capacitance of MIVs and improve the power saving in 3D. As a result, the Cascade-2D shows better power reduction (up to 18.1%) compared with Shrink-2D (less than 10%) on a 32-bit processor design.

In Reference [97], the authors use the Cascade-2D design-aware partitioner on a commercial microprocessor. With a comprehensive understanding of the micro-architecture of the processor, they partition the functional blocks with tight timing coupling into separated tiers, which mitigate the long interconnects on the xy-plane.

In summary, the design-aware partitioning approach allows us to take the design hierarchy and micro-architecture information into consideration, which tends to result in a small 3D net number, but has benefits in power reduction, especially for processor designs. However, this approach does not consider the heterogeneity among the functional blocks. Also, the understanding of micro-architecture is hard to obtain and to utilize if the RTL design is obtained from a third party. Therefore, more intelligent methods such as neural networks need to be used to analyze the design information and provide better partitioning solutions.

### 2.1.4 Logic-on-memory Heterogeneous Partitioning

Logic-on-memory partitioning is a straightforward partitioning strategy to separate the logic and memory components in a design into different tiers. In 2D CPU designs, cache has become an important part of the chips and often occupies more than 50% of the die area in modern multi-core systems. However, the structure of the cache is usually different from the logic modules. The memory blocks consist of high-density bit-cell arrays (e.g., the six-transistor SRAM cell) and peripheral circuits instead of standard cells, which are usually obtained as independent physical IPs and instantiated as hard-macro during IC implementation. The large size of the memory blocks propose challenges for floorplanning and can easily lead to long logic-to-memory interconnects in 2D. Previous studies have shown that M3D integration of L1 caches can provide up to 40% shorter L1 access time and 10% performance improvement compared with 2D [39]. Moreover, the emerging Non-Volatile Memory provides more alternatives for cache implementation, which also introduces more heterogeneity.
between the memory and logic components. Logic-on-memory partitioning allows us to exploit the heterogeneity in 3D ICs.

In Reference [5], the authors show that the macro-on-logic partitioning allows to implement the F2F 3D ICs with the 2D commercial tools and achieve 20.4%–28.2% performance improvement compared with the 2D ICs. The majority of the memory blocks are first pre-placed on the macro die and then projected to the logic die with correct macro pin locations. The remaining memory blocks and standard cells are then placed on the logic die. The performance benefits stem from the shorter vertical logic-to-memory distance, smaller footprint area, and better clock tree quality compared with the 2D baseline. The authors further exploit the heterogeneity of the 3D ICs by removing two BEOL layers from the macro die, because the signal routing density is lower on the macro die than that on the logic die. This has only a slight impact on the performance gain, but introduces significant benefits in terms of manufacturing cost.

The authors of Reference [81] use a similar method in M3D ICs, while the logic die is placed on top of the memory die, which results in the logic-on-memory 3D ICs. This is because if we place the memory tier on the top, then it becomes hard to optimize the MIV placement because memory blocks are large in size and MIVs are not allowed to penetrate them, which also makes it difficult to access the memory pins. Figure 3 shows two examples of the partitioning on a RISC-V processor with different memory capacity. Experimental results show the frequency boost of the logic-on-memory M3D ICs is larger than 35% compared with the 2D baseline, while the improvement is higher with the larger memory capacity. Apart from the advantages of F2F 3D ICs, the performance of logic-on-memory M3D ICs also benefits from the optimized signal routing. The higher BEOL layers of the memory die are less crowded than the lower layers of the logic die but also close to the logic gates, which allow them to act as a "highway" for signal routing. That is, the router utilizes the routing resources on the BEOL layer of the memory die to optimize the critical path and improve the timing.

In Reference [103], the authors propose three guidelines to perform the logic-on-memory heterogeneous partitioning for processor designs. First, keep the 3D floorplan size around 50% of the 2D floorplan size, which ensures the area balance between the memory die and logic die. Second, move as many low-level cache blocks and large-size cache blocks to the memory die. This
is because the low-level caches tend to have high degrees of connectivity with logic modules, while the large-size memory blocks create obstructions for signal routing. Therefore, moving them to the other die helps reduce the logic-to-memory routing wirelength. Third, separate memory block pairs with long 2D interconnects to different tiers. This is to reduce the memory-to-memory routing wirelength. Using these guidelines, the authors implement a logic-on-memory M3D IC with the Spin Torque Transfer Magnetic RAM (STT-MRAM) and achieve up to 32.8% wirelength reduction and 13.9% energy saving compared with the 2D baseline using only SRAM.

In summary, the logic-on-memory partitioning is a straightforward strategy that exploits the heterogeneity of M3D ICs and provides significant performance and power benefits, while it only applies to the designs with more than 50% area occupied by memory.

2.1.5 Graph Neural Network for Tier Partitioning. A graph neural network (GNN) framework, TP-GNN, is proposed in Reference [65] for tier partitioning in two-tier M3D ICs. The framework considers multiple features and constraints to generate the optimized partitioning solution, including the gate-level timing and topological information as well as the block-level design hierarchy information to address the timing and placement quality degradation issues in previous partitioning methods. The framework takes the projected 2D design (similar to the Shrunk-2D stage in Reference [73]) as an input and generates the tier assignment result for each cell as the partitioning solution.

The tier partitioning process using TP-GNN is divided into three major steps: (1) graph contraction and feature aggregation; (2) unsupervised GNN learning; (3) weighted k-means clustering. Figure 4 shows an example of the first step. We first create an undirected clique-based graph based on the projected 2D design. Then, we perform a hierarchy-aware edge contraction algorithm to transform the graph, which contracts the nodes within the same hierarchical module and updates the weights iteratively. The purpose of graph contraction is to preserve the design hierarchy information and improve the efficiency of the flow. Since the cell pairs with the shortest distance in each module are contracted into supernodes, they will not be separated into different tiers in the following steps. This helps avoid cutting short 2D connections and introducing 3D routing overhead, and thus mitigates the placement quality degradation issues in the bin-based and design-aware partitioning methods. The reduced node number also improves the runtime of the flow. After the contraction, the feature of each node is aggregated with the features from its neighboring node. Among the seven features, one represents the block-level design hierarchy; three are related to the cell timing; the other three represent the topology and geometry of the 2D
The combination of these features provides a comprehensive representation of the circuit for tier partitioning and is easy to use in neural networks, while the design hierarchy and cell timing information are hard to integrate with other analytical partitioning methods. And by aggregating the features of neighboring nodes, it considers the various timing and logical dependence of each individual cell on its neighbors, which also helps improve the partitioning quality.

The aggregated features are used in the unsupervised GNN learning process. The loss function is defined based on the cross-entropy between each node and its neighboring nodes. The neighboring nodes are selected via random walking to avoid over-fitting. We use a gradient descent optimizer to minimize the loss function and update the parameters. After that, the weighted k-means clustering algorithm is performed to generate the partitioning results, as shown in Figure 5. The algorithm clusters the nodes into two groups based on the closeness of their feature parameters, while the weight of each node is the total cell area it represents. This approach also ensures all the timing and design hierarchy information are considered to generate the partitioning. Experimental results show that the design quality is improved significantly with the TP-GNN partitioner. For two RISC-V-based processor designs, the total wirelength is further reduced by around 7% compared to the bin-based min-cut partitioner, while the reduction on the critical path wirelength reaches 50%. This is because the TP-GNN approach makes sure the cells with similar node representation and tight coupling are partitioned to the same tier and utilizes the design hierarchy and cell timing information to mitigate the extra 3D routing overhead.

In conclusion, the graph-learning based approach, TP-GNN, combines the benefits of the gate-level and block-level partitioning methods. It allows flexible tier assignment for each individual cell, but also considers the block-level design hierarchy information. The innovation of this approach is that multiple levels of design information are integrated through the neural network and are leveraged to mitigate the placement and timing degradation problem. Therefore, TP-GNN represents the state-of-the-art technique for tier partitioning in 3D ICs.

However, the partitioning approach can be further improved in future studies. First, power-related information should be considered as part of the features, including the internal power of each cell, the toggling rate of each net, and so on. This will help reduce the power consumption in the 3D ICs, especially the switching power, by mitigating the extra switching power consumed on
the 3D nets. In addition, the preferences for power delivery and thermal dissipation should also be considered during partitioning. For example, we prefer to place the current-intensive cells on the bottom tier due to the short distance to the power bumps, while we tend to move some cells to the top tier to avoid the hot spots because it is closer to the heat sink. Moreover, multi-tier 3D ICs and heterogeneous technology propose new challenges for partitioning. To balance the tradeoff of performance, power, and manufacturing cost, consider physical limits of power delivery and temperature, leverage the potentials of more advanced 3D technology, more studies need to be done on tier partitioning.

2.2 3D Placement

Placement was the main focus of early development of 3D IC physical design tools. In the 2000s, several research groups from academia have presented placement algorithms mainly targeting TSV-based 3D ICs. Goplen et al. proposed an extension of the force-directed method to handle 3D placement \cite{40} and the recursive bisection method by adding z-axis cuts to the algorithm \cite{41}. Cong et al. \cite{27} proposed transformation techniques such as folding a 2D layout into \( \frac{1}{n} \) and splitting each square bin into \( n \) different tiers.

In the early 2010s, more aggressive approaches have been presented to solve 3D placement problem, still targeting TSV-based 3D ICs. These works extend the existing 2D analytic placers \cite{18, 24, 63, 90} by considering z-dimension as the new freedom to further improve wirelength and congestion while minimizing overlap among cells and TSVs \cite{28, 44, 49, 64}.

Starting in the mid 2010s, another approach called pseudo-3D design flows made its debut \cite{5, 19, 56, 73}. Pseudo-3D approaches utilize commercial 2D CAD engines to produce 3D IC designs. Unlike the previous academic 3D IC placers (true-3D), these approaches handle routing and timing closure as well. A common theme among these pseudo-3D approaches is that they first build 2D physical design using commercial 2D IC physical design tools. Next, they transform this intermediate 2D layout into 3D layout using various techniques. Pseudo-3D approaches concentrate on seamless utilization of 2D commercial tools to produce 3D IC design, which is the main reason their power, performance, and area (PPA) qualities outperform those of commercial 2D counterparts.

In this section, we start by discussing four true-3D placers \cite{28, 44, 49, 64} and then move on to discuss four pseudo-3D placers \cite{5, 19, 56, 73}.

### 2.2.1 Force-3D Placement

Force-3D \cite{49} is a force-directed partition-first 3D placement approach. In Force-3D flow, a 2D netlist is first partitioned and the cells in the circuit are assigned to different tiers. The partitioning is performed using either FM min-cut \cite{34} or loose-net removal (LR) algorithms \cite{26}. Following the tier assignment, the nets connecting different cells and the I/O pins are treated as springs of different spring constants, which is based on the endpoints of the nets. The equilibrium force acting on this system of springs is calculated and is solved to obtain the initial placement of the cells. Due to the inherent nature of springs, the initial placement solution has all the cells clustered in certain regions. So, a force is required to spread the cells across the entire die to achieve a uniform density distribution across the die. This force is called the move force. However, due to the elastic nature of springs, they try to return to their equilibrium position when a force is applied on them. Hence, a hold force is required to prevent the cells from moving back to the previous position. On solving the system of these three equations, a global placement solution with cells uniformly distributed across the dies is obtained. The global placement obtained is imported into commercial placement tools and then tier-wise detail placement is performed. Figure 6 shows the overall force-3D placement flow.
2.2.2 Non Linear-3D Placement. Non Linear-3D (NL3D) [28] placement flow uses an analytical approach to solve 3D global placement problem using non-linear optimization methods. In NL3D, a 3D floorplanning is first performed on a 2D netlist. The cells in the circuit are clustered into blocks; the blocks are assigned to multiple tiers; and then the blocks are declustered to generate an initial placement. The aim of NL3D global placement tool is to minimize the overall wirelength and TSV usage in the design, subject to non-overlapping constraints. The overall design is divided into several bins and the cell density in each bin is restricted to achieve the non-overlapping constraints. This placement problem is modelled into an objective function combining the wirelength and non-overlap constraints and then solved analytically to obtain the global placement solution. The non-overlap constraints are transformed into a density penalty function using Huber approximation and Helmholtz global smoothing techniques, making it easier to solve the objective function analytically. Detail placement and legalization are then performed on each tier using a custom legalizer. However, the legalized results may sometimes contain overlaps and require a commercial legalizer to obtain the final detail placement solution. Figure 7 shows the overall NL3D placement flow.

2.2.3 ePlace-3D Placement. ePlace-3D [64] is another flat, analytical, non-linear placement tool that focuses on wirelength minimization and density equalization. Similar to NL3D, in ePlace-3D, an objective function is analytically solved to obtain a 3D global placement solution for a given 2D netlist. Unlike using mathematical approximations to model the non-overlap and density equalization constraints, an electrostatic force-based density model called eDensity-3D is used by ePlace-3D. The overall placement region is modelled as an electrostatic field, and each placement object is treated as a positively charged cuboid. The cells are distributed away from high-density regions due to the repulsive nature of like charges, thereby achieving a uniform placement density. The overall objective function modelling the wirelength constraint and electrostatic density function is solved in an iterative manner, till all the cells in the circuit are evenly distributed across all tiers. After obtaining the global placement solution, commercial tools are used to perform detailed placement and legalization. Though, ePlace-3D achieves better density equalization, it requires more number of tiers to place all the cells compared with other true3D placement approaches. Figure 8 shows the overall ePlace-3D placement flow.

2.2.4 NTU-3D Placement. NTU-3D [44] is a TSV-aware analytical 3D placement approach that takes the area of TSVs into account while modelling the placement objective function. NTU-3D is
a pure placement engine and requires an initial placement. Similar to NL3D, the whole placement region is divided into multiple bins and subjected to maximum bin density constraints. However, unlike NL3D, areas of both cells and TSVs in each bin are considered while calculating the bin density. As a result, white space is reserved for TSV placement during 3D global placement stage. On obtaining the global placement solution, the cells are legalized tier-by-tier, followed by TSV insertion and legalization. Figure 9 shows the overall NTU-3D placement flow.

2.2.5 Shrink-2D Placement. Shrink-2D [73] uses a commercial 2D CAD tool to generate an intermediate result in which all cells and wires are shrunk in half and placed onto a 2D die with a half footprint. This guarantees the full utilization of a commercial 2D placement tool. Following the placement and routing with shrink standard and hard macro cells, the cells and wires are resized to their original sizes; partitioning is performed, and the cells are assigned to specific tiers and the MIV locations are fixed on the 3D nets.

Figure 10 shows a conceptual view of the Shrink-2D flow. In the intermediate Shrink-2D P&R result from the commercial 2D CAD tool, all standard cells and wires are scaled down to 50% and its total footprint has 50% area of its 2D design counterpart. All components are expanded to their original size afterwards, which creates many overlaps in the design (Figure 10(2)). The chip is then divided into several square bins (Figure 10(3)), and the Fiduccia-Mattheyses (FM) min-cut
partitioning algorithm [34] is applied to each bin to evenly partition the cells into both tiers (Figure 10(4)). Remaining cell overlaps in both tiers are removed during tier-by-tier legalization (Figure 10(5)). Shrunk-2D performs tier partitioning after the full P&R of an intermediate 2D design with a 2D CAD tool. This improves the partitioning quality to maintain the planar placement result from the commercial tool. Figure 11 shows the steps in the Shrunk-2D design flow. Except for the tier partitioning step, the Shrunk-2D design flow fully utilizes the commercial 2D tool in 3D design and generates an efficient monolithic 3D design output.

2.2.6 Compact-2D Placement. Shrunk-2D has risks of shrinking cells and wires into an even smaller technology than the commercial tool can handle. It also has inaccurate RC values for shrunk interconnects. This affects the quality of final 3D design results. The Compact-2D [56]
design flow resolves this problem by using similar concepts from Shrink-2D while not shrinking cells and wires. Instead, it scales only the RC parasitics ($1 \rightarrow 0.707$) by a factor of 0.707 to make the CAD tool treat the interconnect much faster than that of the original 2D design. Therefore, a Compact-2D P&R result has smaller timing path delays compared to the same path length in a 2D design, which requires fewer resources (e.g., buffers, inverters, ...) for timing closure. When compressing the whole design into 50%, the interconnects return to their original RC value and the timing closure resources become sufficient. This flow removes a great amount of potential design rule violations that come from shrinking elements beyond the coverage of CAD tools. The conceptual view of the Compact-2D flow is shown in Figure 12. The intermediate Compact-2D P&R result has the same area and cell size as a 2D design. In the next stage, it contracts the chip area to 50% and generates a placement result with high overlaps, which is similar to the intermediate stage of Shrink-2D. Since all interconnects are scaled down and the unit RC parasitics are scaled up to the original size at the same time in an identical proportion, the overall interconnect RC parasitics remain consistent. The rest of the steps are the same as those in the Shrink-2D flow. Figure 13 recites the Compact-2D design flow. It is conceptually similar to the Shrink-2D design flow, except that Compact-2D only adjusts the RC parasitics scaling factor, instead of the whole technology files.

2.2.7 Cascade-2D Placement. The main motivation of Cascade-2D [19] is to place all tiers simultaneously, rather than performing tier-by-tier designs. Since existing commercial 2D CAD tools...
cannot restore all tiers of a 3D design at the same time, Cascade-2D uses a 2D structure that represents a 3D design in a vertically long floorplan (W:H = 1:2) where the top and bottom half represent the top and bottom tier correspondingly. Each inter-tier connection is represented with two dummy anchor cells at the MIV position in both tiers and a zero-parasitic wire that connects the anchor cells. Figure 14 shows the overall design flow of Cascade-2D. It partitions the 2D netlist into 3D netlists with two tiers (say, top and bottom). The 3D nets are represented as I/O pins in each tier, and anchor cells are attached right next to each pin. Then, the MIV locations are determined by anchor cell locations from the sequential 2D placement of the top and bottom tier.

During the top tier placement, we locate the driving MIVs (top to bottom) at the anchor cell’s output ports. The bottom tier placement is performed while considering the MIV locations from the top tier placement, which guides the receiving cells to be placed near their driving MIVs. Then the remaining anchor cells and MIVs (bottom to top) are placed in a similar way. MIV locations are then restored to the Cascade-2D floorplan of 1:2 aspect ratio and routed with zero parasitic dummy wires. Dummy wires represent the instant connection between both tiers through MIVs. A hard horizontal fence across the center forbids cross-half (=cross-tier) cell movement during the entire flow. After all design steps, the Cascade-2D design result is transformed into a monolithic 3D design by laying the upper half (top tier) upon the lower half (bottom tier) and replacing each dummy wire with a vertical MIV. Cascade-2D fully utilizes the efficiency of commercial 2D CAD tools, which leads to a better timing-closed design compared with the tier-by-tier design.
2.2.8 Macro-3D Placement. Macro-3D [5] is a memory-on-logic 3D placement tool that uses commercial tools to generate placement results. Unlike prior works such as Reference [82], Macro-3D allows 2D tools to complete M3D placement with a real 3D metal stack and exact memory pin locations. In this flow, the memory blocks are treated as hard macros; assigned to the memory tier; and their placement is fixed. Then, the hard macros are shrunk to the minimum possible technology dimension, called the site size, and placed on the logic tier, retaining their pin locations at the memory tier. This way the hard macros do not cause any placement congestion and routing is performed in a 3D fashion. After fixing the hard macros placement, a commercial 2D placement engine is used to place the logic cells and MIVs, followed by legalization. Figure 15 shows the overall macro-3D placement flow. As the name suggests, this flow is best suited when one of the tiers is completely filled with hard macros.
2.3 3D Clock Delivery Network

The clock delivery network plays a major role in 3D Integrated Circuit performance as well as the main contributor to dynamic power from higher target performance. In the beginning, the 3D clock tree network goal is to minimize and balance the skew from thermal gradients during real-time operation. Later, the fabrication and manufacture of 3D IC become possible with F2F stacked vias, pre-bond test-ability of the clock tree becomes useful to prevent bonding defected dies together. With the invention of Through-silicon-via, coupling effects and geometry issues become significant due to the impacts of TSVs on the reliability of the clock signal and performance. Building a 3D clock tree with constrained locations called “TSV-array” prevents reliability issues. Recently, there are some studies in building 3D ICs by utilizing 2D commercial EDA tools, which bring about the optimization of the clock tree. Therefore, we present the solution to 3D Clock Delivery Network as follows:

2.3.1 BURITO 3D Clock Routing. The Buffered Clock Tree with Thermal Optimization (BURITO) [69] is the first 3D clock tree synthesis method that considers the impacts of thermal gradient on skew. To cope with thermal-inducing skew, our BURITO minimizes and balances clock skew between two distinct thermal profiles using the proposed Balanced Skew Theorem. Since the thermal distribution is changing during the real-time operation, there are many possible distributions where we could not target for all of them. Instead, we bound the skew between the uniform distribution and the worst-case scenario. Therefore, the thermal-induced skews are handled between these two bounds.

To calculate the skew gradient in 3D, we compute the interconnection delay under the non-uniform thermal profile with the following steps: We segment the interconnection, compute the thermal variant delay for each segment, and calculate the total delay along the path. Since the path contains a merging point between die, a delay formulation is proposed where the connection between two clock sink (p and q) has a merging point u. In Figure 18, there are two examples of merging point for non-buffered clock tree in Figure 18(a) and buffered clock tree merging in Figure 18(b).

With the skew gradients, we calculate the delay for each clock sink to utilize it in the balanced skew theorem. Note that, using average timing profile instead of two at corresponding locations made solving step faster than calculating each in one timing profile separately.

BURITO Clock routing algorithm consists of four main steps: 3D abstract tree generation, clock tree embedding and buffering, and thermal-aware optimization. The entire process is illustrated in Figure 16.

(1) 3D Abstract Tree The abstract tree generation is the first step to build the clock tree by assigning through-via under the through-via bound while minimizing the wirelength. Our approach extends the MMM algorithm [46] to generate the abstract tree in a top-down fashion with additional settings, as shown in Figure 17. The process starts from divide the clock sinks into two subsets recursively until each set contains only one sink. Next, we start from the leaf node in a bottom-up fashion to assign the tier location based on the through-via bound. We consider the following rules during this step: (1) If via bound is one, then each sink set is partitioned in the way that each subset belongs in the same tier. (2) If the bound is larger than one, then the set is flattened and partitioned by vertical and horizontal lines. At the end of the partitioning, we decide the number of through-via bound for each subset as follows: (i) estimate the number of through-via required for each set; (ii) divide given bound with the ratio of estimated vias. And the cut direction is selected based on balanced via bound in both subsets. Timing complexity in the abstract tree generation is $O(n \log(n))$, where n is the number of clock sinks.

Fig. 16. Four main steps of our BURITO [69]: (a) 3D abstract tree generation, (b) embedding and buffering, (c) thermal analysis, (d) thermal-aware optimization.

Fig. 17. Our BURITO abstract tree under various through-via bounds [69].

Fig. 18. Our BURITO feasible merging region for buffer insertion [69].
(2) **Buffering and Embedding** In this stage, clock sinks are placed and buffers are inserted to meet the zero skew constraint under uniform thermal distribution. Our approach extends the DME algorithm \[10\] to handle the topology for the 3D abstract tree. For buffer insertion, we extend Reference [94] to support 3D environment. The merging steps for both unbuffered and buffered tree are illustrated in Figures 18(a) and 18(b), respectively. The line and polygon are the feasible region for unbuffered tree and buffered tree.

(3) **Thermal Optimization** The final step is to perform the thermal analysis under thermal variations. We relocate clock sinks and buffers to minimize and balance skew with Balanced Skew Theorem. We extend Reference [25] to merge clock nodes in different dies.

2.3.2 **Pre-bond Testable Clock Tree Router.** In 3D stacked ICs, each die is fabricated separately. Thus, the testing in each dies before bonding is an important process to avoid bonding with defective dies. The main challenge is the clock tree needs to be completed for pre-bond testing, while the 3D clock tree is composed of nets in both the top and bottom dies. The straightforward approach is to have two complete clock tree on each die with one Through-Silicon-Via in between. However, the overall wirelength is long and consumes more power, because the benefits of wirelength saving from 3D routing are not fully utilized. In this study, we proposed the methodology to perform pre-bond testing on a 3D clock tree.

Our algorithm implements a pre-bond testable clock tree for the two-die stack. In this study, the top die contains the clock source. The algorithm consists of two main steps: (1) 3D tree construction and (2) Redundant tree routing. In 3D tree construction, we generate the 3D clock tree such that the top die contains a fully connected 2D tree, and wirelength is minimized with a zero skew. We extend 3D-MMM algorithm \[69\] to construct the 3D clock tree with additional elements for pre-bond testing capability. TSV buffer and transmission gate are inserted in the process with redundant tree technique to complete the clock tree for each die for testing purposes.

Since the top die contains a complete clock tree with a located clock source, the testing can be performed without any additional modification. However, the clock skew is not zero, because some branches are located on another die after the separation process. To avoid the skew degradation from the pre-bond testing for top die, we employ the TSV buffer to be inserted at the inter-die location right before the TSV. As a result, the 3D clock tree is constructed with TSV buffer such that a zero skew is obtained and wirelength is minimized. With TSV buffer, skew degradation is resolved as TSV buffer hides the downstream capacitance. Figure 19 illustrates the comparison between impact of TSV-buffer insertion on skew calculation on the CT1 branch.

For bottom die, the pre-bond testing requires a fully connected clock tree, while only the forests with missing subtrees are formed from the 3D clock tree routing algorithm. To complete the subtrees into a complete clock tree, we add an additional tree called “redundant tree” that connects the root of the subtree while maintaining the zero skew. The **transmission gates (TGs)** are used to connect and disconnect the redundant tree, as we only need the fully connected clock tree on the bottom die for pre-bond testing. The redundant tree routing is performed by first inserting the TGs at roots of the bottom die subtree in a top-down fashion. Next, we use DME algorithm \[46\] to embed and buffer the abstract tree under zero-skew and minimum wirelength constraints. Lastly, we connect the control wire to TGs while minimizing the wirelength using RMST-pack \[48\] to minimize the overhead. The impacts of TSV bound on wirelength and clock power of our 3D pre-bound clock tree are shown in Figure 20. The results show that the wirelength constantly reduces with the increasing TSV numbers.

Once the clock tree on each die is a completed clock tree using our algorithm, we perform per-bond testing on each die. In the bottom die, we turn on the TGs to connect the redundant tree to subtrees. In top die, there is no further action required, since the clock tree is competed by itself.
Fig. 19. TSV-buffer insertion approach [100].

Fig. 20. Pre-bond clock power under various TSV bound [100].
along with TSV-buffer to hide out downstream capacitance. After pre-bond testing is completed, All TGs are turned off and the design is ready for post-bond testing.

2.3.3 Clock TSV Array Utilization. 3D ICs become popular from the benefits of lower power, higher performance, and smaller footprint. However, the Through-silicon vias (TSVs) may impact the reliability and cost issue in the manufacturing process. The main issue occurs from the deformation of TSV locations. TSVs can squeeze and stretch adjacent transistors and interconnects that cause mechanical issues such as open hold, short, or crack. Another issue is the coupling effect between TSVs themselves or with other devices affect timing and waveform. Therefore, the main idea of TSV array design is to determine TSV locations that are manufacturable and mitigate TSV-related reliability issues simultaneously. As a result, we propose the work on 3D clock routing for TSV array utilization in [101]. To optimize the skew and power, we present our decision-tree-based clock synthesis (DTCS) to efficiently explore the entire solution space of TSV array utilization with the minimized skew and power. We integrated our DTCS with existing 3D clock synthesis to utilize TSV array. Our DTCS constraints three main steps: (1) Decision Tree Construction; (2) Clock Tree Construction; (3) Clock Tree refinement.

Our decision tree is a binary tree that allows us to visualize the entire solution space of TSV array utilization, as illustrated in Figure 21. Each tree node represents a state where sink sets are partitioned into two groups with one TSV in between. Two sink sets are children of a tree node that allow the sink set to be recursively partitioned to all possibilities. We estimate the power consumption of each tree node for the later stage to find the minimum node to construct the clock tree. Each step is performed by using 3D-MMM [69] with the TSV bound of 1 and later insert the buffer using the sDMBE algorithm [102].

The decision tree in the previous stage is parsed to construct the clock tree. Only leaf nodes contain their X-Y cut power value while the internal node’s power a Z cut. Thus, this step is to find the missing information by visiting a decision tree in a bottom-up fashion, which is described as follows: The cut direction for X-Y in each node is the minimum value among four possibilities:
Z+Z, Z+XY, XY+Z, XY+XY. Once the traversal is completed, we obtain all number of cut and cut direction of each cut.

From the clock tree result in the previous stage, the TSV bound may not be satisfied, since the cut decision is based on the minimum power. Therefore, we develop a Clock Tree Refinement algorithm to reduce the TSV count under bound constraints while maintaining the power. The main approach is to convert X-Y cut in subtrees to Z-cut, as the X-Y cut requires more TSV numbers. We develop the binary-integer-linear-programming (BILP)-based algorithm to choose the optimal set of decision nodes.

2.3.4 3D Clock Optimization. In recent years, 3D IC has shown a good potential to continue Moore’s law prediction. There are many studies that attempt to develop the 3D IC Place-And-Route (PnR) tools. However, the tools are not a complete flow or very limited for in-house usage. The Pseudo-3D approach, where specialized PnR flows are utilizing 2D commercial EDA tools with tricks to implement industrial-quality grade, is more popular. The clock tree in shrunk2D [73] is not fully optimized and causes timing degradation after the tier-partitioning stage. In this study, we proposed the optimization algorithm that utilized 2D commercial EDA tools. Our flow extends the shrunk2D flow by optimizing its clock tree using clock clustering and FF relocation methods.

Our M3D Clock tree optimization extends Shrunk-2D flow, as shown in Figure 22. Two additional information is required as an input for the algorithm: (1) Clock Tree of Pseudo-3D of S2D to perform clock clustering based on clock hierarchy to minimize the inter-tier overhead delay, (2) Critical Path of S2D M3D to analyze which Flip Flop to be moved to improve the clock skew by moving it closer to its root.

As a first step of the M3D clock tree optimization as shown in Figure 23, we obtain the clock tree hierarchy from the Pseudo-3D stage where the clock tree is constructed and buffered using 2D commercial tools by shrinking cells to fit in projected 3D footprint. Note that in baseline (Shrunk2D), once Pseudo-3D PnR is completed, cells are partitioned into two die by using FM Bin-based min-cut [34]. Next, locations of all cells are legalized to fit the cells within the placement rows. As the clock sinks are partitioned randomly with the goal of minimum wirelength, routing overhead in the clock tree from MIVs is introduced by placing two sinks in different die. With the clock tree hierarchy from the Pseudo-3D stage, the goal is to group the clock sinks based on the subtrees, as shown in Figure 24.

With the clock clustering group, we use a simple but effective algorithm to assign the clocks within the same cluster at the same tier. The straightforward method is to obtain the baseline tier partition result of all the clock sinks and use the majority vote within clusters to assign the tier location. If there is a tie, then we assign to the tier that balances the number of clock cells between tier. This will remove the unnecessary MIVs on the clock path, which introduces additional clock delay and clock skew within that cluster.
Since high clock skew can cause the timing violations, moving the Flip-flop closer to the center of its subtree reduces the clock latency, which helps improve the clock skew and the timing. The concept of FF relocation is shown in Figure 25 where FF1 and FF2 are in the same cluster, and moving FF2 closer reduces the induced wirelength, resulting in small clock latency. However, choosing the FF to relocate has to be careful, because it may introduce additional combinational delay paths and hurt the timing on other paths. We select the FFs in the critical paths where the clock skew is higher than the average skew within the same cluster to be move.

2.4 3D Thermal Analysis and Optimization

One of the major challenges with respect to 3D IC design is to deal with thermal issues. In general, 3D ICs face more thermal issues than 2D ICs. 3D ICs have several device layers stacked on top of each other and pack a lot of devices in a smaller area, leading to constricted paths for thermal dissipation. Several techniques have been proposed to address the thermal issues in different stages of physical design by using different thermal analysis and optimization techniques over the past two decades. This section includes six such noteworthy thermal analysis and optimization techniques that are used to address thermal issues in 3D ICs with efficient ways during floorplanning, placement, or routing stages. Previous studies on M3D thermal analysis and optimization include References [45, 58].

2.4.1 3D Floorplanning with Thermal Vias. In 3D ICs, thermal dissipation becomes extremely difficult, as the devices are packed together on top of each other over a small area. The high packing density of 3D ICs makes it difficult for the heat generated by devices at the center of the bulk to dissipate to the ambient. One of the efficient techniques to tackle this issue is to insert thermal vias through all the device layers, thereby providing an unrestricted path for heat dissipation from the bulk of the IC. Reference [32] proposes a thermal via insertion technique during the floorplanning stage to address the thermal issues in 3D ICs. Given (i) a set of blocks, (ii) the width, height, and average power density of each block, (iii) a netlist that specifies how the blocks are connected, and (iv) the number of device layers in the 3D structure, the thermal-aware 3D floorplanner in Reference [32] finds a location for each block in the floorplan and plans the location of the thermal vias in such a way that the cost function shown in Equation (1) is minimized. $A_{tot}$ is the total circuit area including the thermal vias, $wl_n$ is the wirelength of net $n$, and $T_{max}$ is the maximum temperature of the substrate and $w_1$, $w_2$, and $w_3$ are the corresponding weights.

$$w_1 \cdot A_{tot} + w_2 \cdot \sum wl_n + w_3 \cdot T_{max}$$  \hspace{1cm} (1)
There are three different approaches to thermal-aware floorplanning using thermal vias. The first approach is to use the traditional simulated annealing floorplanning technique to optimize the overall area and wirelength without considering the thermal component initially, followed by inserting thermal vias at thermal hotspots. This approach is called Area Wirelength-driven Floorplanning (AWF). The second approach is to reduce the overall temperature of the chip considering the area, wirelength, and chip temperature simultaneously, followed by inserting thermal vias at hotspots. This approach is called Thermal-Driven Floorplanning (TDF). The third approach is integrating the thermal via inserting stage with every iteration of the simulated annealing technique, so the effect of thermal vias is also considered. This approach is called Integrated thermal Via Floorplanning (IVF). A 3D thermal resistance mesh model, as shown in Figure 26, is used to calculate the temperature of the entire chip. In this mesh model, each node represents a small volume of the 3D die stack, including the TSVs, substrate, heat sink, dielectric, metal, and transistors, and each edge denotes the thermal conductivity between two adjacent regions. Then, the thermal conductivity and power dissipation of the 3D mesh are calculated and a thermal resistivity matrix \( R \) and a power vector \( p \) are generated. The temperature of the chip \( t \) is calculated using Equation (2):

\[
t = R \cdot p.
\]  

In designs where the thermal conductivity does not change, the fast matrix approach of directly using Equation (2) is efficient, and in designs where the thermal conductivity changes a random walk approach, which is iterative and recalculates the thermal conductivity, gives better results.

2.4.2 Thermal-aware 3D Steiner Routing. A 3D IC Steiner Routing (3DSR) technique for multi-pin net routing in 3D stacked ICs is proposed in Reference [70]. It considers all dies simultaneously and constructs performance-oriented Steiner trees under the given thermal profile while determining the locations for the related TSVs. The goal of 3DSR is to reduce both Elmore delay among all pins in each net and the maximum temperature of all the nodes in a given thermal grid. The 3DSR algorithm consists of two phases. The first phase, called the construction phase, creates 3D Steiner trees without considering any congestion. The second phase, called the refinement phase, removes congestion by rip-up-and-route. Following the Steiner routing, a TSV relocation
Fig. 27. Fast thermal model, where $R_b$ denotes the thermal resistance to the heat sink, $P_1, \ldots, P_5$ is the power consumption of each layer, and $R_1, \ldots, R_5$ represents the vertical thermal resistance on each layer. [70].

A technique is used to move as many TSVs into thermal hotspots as possible while preserving the original tree topology obtained during the construction step. The main objective of TSV relocation is to minimize the maximum temperature on the chip by providing a vertical heat dissipation path without violating the timing constraints. A fast thermal model [29] as shown in Figure 27 is used to perform thermal analysis and thereby optimize the TSV location. The fast thermal model-based thermal analysis is performed before and after TSV relocation to ensure the TSV relocation reduces the overall chip temperature. The relocation is performed using an iterative linear programming technique (ILP), as described in Reference [70]. On average, this ILP-based 3DSR TSV-relocation technique reduces the chip temperature by 10%–20%.

### 2.4.3 Nonlinear Regression-based Thermal-aware 3D Floorplanner

In a thermal-aware floorplanner, the thermal analysis stage plays a crucial role in making decisions that reduce the overall operating temperature of the 3D ICs in an effective way. Samal et al. propose a unique nonlinear regression-based fast thermal analysis technique in Reference [87], which can be incorporated with existing thermal-aware 3D floorplanners to achieve better thermal behavior in 3D ICs. The fast thermal analysis approach models the temperature based on different aspects of monolithic 3D ICs that affect the circuit temperature drastically. In this model, an entire chip is divided into several 3D meshes, as shown in Figure 28(b), and each mesh is assigned a power density value. The size of the 3D mesh can be tuned for better accuracy. The model also considers conventional cooling methods such as heat spreader and heat sink modules attached to the ICs, as shown in Figure 28(a). A full-chip thermal analysis is run using ANSYS Fluent and few supporting scripts.

Based on the thermal analysis, it is identified that the following factors affect the operating temperature of an IC: (1) Power of objective tile, (2) Total Power of rest of the tiles in the same tier, (3) Lumped sum of power of all tiles exactly above the objective, (4) Lumped sum of power of rest of tiles of the above tiers (excluding the ones directly above), (5) Lumped sum of power of all tiles exactly below the objective, (6) Lumped sum of power of rest of tiles of the tiers below (excluding the ones directly below), (7) Distance of the tile from each of the four 2D boundaries (four variables), and (8) Distance from vertical boundaries (3D location). For practical designs, this model accurately determines the temperature with an average error less than 5%. This thermal model is then incorporated with a thermal-aware floorplanner to optimize the chip temperature, by including temperature as one of the parameters in the cost function. The temperature parameter
Fig. 28. (a) 3D IC packaging structure for cooling with heat sink. (b) Thermal analysis model structure with an objective tile (yellow) and rest of the tiles. [87].

Fig. 29. TSV spread and TSV align forces added to force3D placement flow for better thermal behavior of 3D ICs [4]. The orange rectangles are the TSVs, and the red rectangles represent the regions with high-power dissipation, while the green rectangles denote the regions with low-power dissipation. The yellow covering on the green rectangles in (b) shows that the heat generated by the high-power regions on the bottom die can be transferred to the top die through the TSVs and increase the temperature in the green regions, which could be mitigated with TSV alignment.

is tuned only after a certain area constraint is achieved. This is done to ensure that the chip does not blow up in size by a very large extent to reduce the operating temperature significantly. On average, fast thermal analysis model incorporated 3D floorplanner helps in reducing the maximum operating temperature by up to 22% with a little performance and area overhead.

2.4.4 Thermal Coupling-aware Force-3D Placer. A TSV spread and alignment method that exploits the thermal properties of TSVs is presented in Reference [4]. The TSVs are evenly distributed to reduce power density in local power hotspots and TSVs of adjacent tiers are vertically aligned to provide a direct path from every die to the heat sink attached to the top of the IC. Using ANSYS Fluent, a part of the bulk of silicon is simulated with and without TSVs and the corresponding thermal conductivity values are computed. These values are then used to guide a force3D placer, mentioned in Section 2.2.2.1, to perform thermal-aware 3D placement that reduces the overall chip operating temperature. A TSV spread force is used to spread the TSV across each die, thereby reducing intra-die thermal variation. Then, a TSV align force is applied on the TSVs to align the TSV across all the dies. The illustration of these two forces is shown in Figure 29.
2.4.5 Tile-based Thermal Analysis Flow. The tile-based thermal analysis flow [104] is an efficient and accurate method to simulate and generate the thermal profile of an entire 3D IC considering inter-die thermal coupling. The overview of the flow is shown in Figure

In this flow, the authors first divide a 3D IC into square tiles on the xy-plane and then divide each tile into cubes along the fabrication layers on the z-axis. Then, the power map of the device layer is generated using the vector-based power analysis results. This tile-based power map reflects the worst-case heat generation in the 3D IC and acts as a heat source during thermal analysis.

After that, the authors analyze the thermal conductivity of each cube based on the material properties and density, as shown in Figure 31(a). Within each cube, the vertical thermal resistors are in parallel, since they consist of different materials and the heat flow is distributed throughout the materials. However, the lateral thermal resistors are in series. The thermal conductivity map is created by putting all the cubes together, as shown in Figure 30. With both the power map and thermal conductivity map, the authors perform the thermal analysis using a commercial thermal solver. Figure 31(b) shows the generated temperature map of a F2F 3D IC with the AES-128 benchmark design.

This approach can be applied to M3D ICs with minor modifications. The 3D stack in M3D ICs is face-to-back bonded, but it can still be divided into thermal cubes along the z-axis in a similar manner. Also, MIV density affects the vertical thermal conductivity of the interface cubes with MIV insertion, and thus the heat dissipation through MIVs can be taken into consideration.

In summary, the tile-based flow provides an efficient and reliable solution for thermal analysis in M3D ICs, since the runtime of the flow depends on the tile size and the resolution is highly

---

Fig. 30. An overview of the tile-based thermal analysis scheme: (1) generate 3D thermal conductivity map by analyzing the lateral and vertical thermal conductivity of each cube; (2) create tile maps for each active layer based on the power distribution. The colors on each map show the various thermal conductivity, temperature, and power throughout the 3D IC, respectively.
Fig. 31. Thermal conductivity calculation (a) in the tile-based analysis flow and the generated temperature maps (b) in a F2F 3D IC with the AES-128 benchmark design [104].

adjustable. The analysis results can be used in thermal-aware timing/power analysis and design optimization.

2.5 Summary of EDA Challenges and Solutions for M3D

In this section, we provide comparisons between M3D and conventional 3D ICs (TSV-based 3D and F2F 3D) from an EDA perspective and discuss how we can apply the physical design methods presented in the previous sections to M3D ICs. We first summarize the differences between 2D, conventional 3D, and M3D at each design stage in Table 1.

For tier partitioning, M3D provides much higher connectivity compared with conventional 3D, and the RC parasitics introduced by MIVs are negligible. As a result, the optimal partitioning solution can have a large number of 3D nets in M3D, while the tier partitioning methods presented in Section 2.1 can still be applied to M3D ICs.
Table 1. Comparisons between 2D, Conventional 3D, and M3D ICs in Terms of Physical Design

<table>
<thead>
<tr>
<th>Design stage</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tier partitioning</td>
<td>N/A</td>
</tr>
<tr>
<td>Power delivery</td>
<td>Regular 2D PDN</td>
</tr>
<tr>
<td>Placement</td>
<td>2D placement</td>
</tr>
<tr>
<td>Clock delivery</td>
<td>2D CDN</td>
</tr>
<tr>
<td>Thermal analysis</td>
<td>2D thermal analysis</td>
</tr>
</tbody>
</table>

For power delivery, M3D ICs are faced with unique challenges, because the M3D PDN may consist of a large number of MIVs and power MIV placement can be irregular [20]. Building a reliable M3D PDN is an interesting topic for future studies.

For placement, the algorithms need to consider a large number of 3D nets in M3D ICs to optimize the solution. The Pseudo-3D flows (Shrunk-2D, Compact-2D, Macro-3D, etc.) can provide optimized placement solutions using 2D tools, but true-3D placer is more desirable with M3D.

For clock delivery, the algorithms mentioned in Section 2.3 can be adjusted to the large 3D net number and applied to M3D.

For thermal analysis, M3D ICs require a different thermal model for MIVs and packages. With the tile-based thermal analysis flow presented in Section 2.4.5, we can model the M3D ICs and perform thermal analysis for the entire 3D stack.

3 SYNERGY BETWEEN PHYSICAL DESIGN AND TESTING

With the continued process scaling and the impact of fault diagnosis on volume production, testing has become a critical part of the IC manufacturing flow [16]. Emerging technologies like M3D ICs at the nanometer regime are especially prone to random fabrication process variations and manufacturing defects. These imperfections often lead to parametric faults that require attention during manufacturing test. Additionally, M3D ICs are vulnerable to aging-related effects such as thermo-mechanical stress and power-supply noise. These necessitate low-cost built-in self-test (BIST) architectures to ensure in-field reliability.

Efficient DfT methodologies must offer high fault coverage at a low area and test time overhead. Towards this end, the physical design and testing aspects in the proposed M3D design flow are closely integrated. To highlight the challenges in M3D testing, we first explore the different sources of imperfections in the M3D fabrication process with emphasis on parametric delay faults that arise due to resistive opens and bridges [36]. Delay fault detection is also affected due to inter-tier coupling and voids in the ILD. We show that the effects of these defects vary based on several physical design considerations, including tier partitioning and die stacking. We then present multiple DfT solutions for tier-level fault localization and MIV testing that have been integrated with the physical design flow.

Section 5 presents a background on small delay defects (SDDs) and how testing for these defects is affected due to various defects. Defects in MIVs are likely to have a significant impact on performance; therefore, Section 5 discusses MIV fault models, explores the applicability of TSV-testing methods to MIVs. We also present a built-in self test structure that can be easily integrated with the physical design flow in a non-intrusive fashion.
4 DELAY TESTING FOR MONOLITHIC 3D ICS

4.1 Background

4.1.1 Delay Fault Models. Process variations and manufacturing defects result in parametric faults, increased propagation delay, and slower signal transitions. The gate delay fault model targets defects that affect the propagation delay of gates [9]. The path delay fault model targets the accumulation of delays, including small delays, along a path. Of particular interest are long paths, i.e., paths with small timing slack, while often only a subset of long paths are tested.

The transition fault model is a special case of the gate delay fault model; it assumes that the delay fault at a node exceeds the clock period and is catastrophic enough to cause logic failure [83]. A transition fault can affect the timing slack of all paths through the affected cell. However, a fault can be detected only on paths with a small enough timing slack such that, in the presence of the fault, a signal transition is not captured within the rated clock period. Therefore, to detect small-delay defects, it is critical that the longest (minimum-slack) paths through a fault site are selected. The transition fault at a node can be classified into “no effect,” “limited effect,” or “gross effect,” based on the size of the defect [9]. Defects of the type “no effect” have negligible impact on the gate delay, and as such, do not have an impact on circuit operation. “Limited Effect” and “gross effect” faults result in timing failure in some and all of the paths through the affected cells, respectively.

Delay faults at a site are typically detected using an ordered pair of test patterns, where the first and second patterns are called the launch and the capture cycle, respectively. In the fault-free case, the launch and capture cycles ensure that a signal transition is justified at the fault site and propagated to a capture flop within the rated clock period. If a delay fault is present, then the transition in at least one path through the fault site is not detected within the rated clock period. Process variations typically result in a small change in the propagation delay; thus, for increasing the probability of these small delay defects, the ATPG tool tries to propagate the transition via the longest path through the fault site.

Resistive opens and bridges in vias and interconnects often result in small delay defects (SDDs) in nanometer circuits [36]. SDDs can also originate due to variations in device parameters that typically have a parametric impact on the propagation delay of affected logic gates [91]. Such parameter variations can be caused by imperfections in the fabrication flow, thermo-mechanical stress, crosstalk, power-supply noise, and aging [98]. Traditionally, burn-in tests have been used to identify dies with such reliability concerns [91]. However, burn-in is an expensive process and has been shown to damage dies due to extreme stress conditions [95]. This motivates the use of SDD tests as a low-cost alternative. To ensure efficient SDD detection, delay faults must be tested via long paths. Commercial EDA tools attempt to ensure this by using a “constrained” transition delay fault model—only the paths having a nominal timing slack lower than a predefined margin are used for testing.

The Statistical Delay Quality Level (SDQL) is a commonly used surrogate metric for SDD coverage and test pattern grading [98]; a lower value of SDQL signifies higher effectiveness of the SDD test patterns. To simulate the effectiveness of test patterns under process variations, a delay-defect distribution function $F(s)$ is considered, where $s$ is the delay fault size. Therefore, $F(s)$ denotes the probability that either a rising or a falling transition delay fault of size $s$ exists at a random node. Note that $F(s)$ is typically obtained from empirical data or a process test chip; some examples of such distributions can be found in Reference [89]. SDD detection becomes more relevant when more defects are of small size, i.e., $F(s)$ decreases rapidly with increasing $s$.

Consider an SDD fault $X$ and a corresponding test pattern $TP_X$ for it. Let the timing margin of the longest (minimum-slack) path through $X$ in the absence of any parameter variations be
given by $T_m^\star$. Similarly, let the nominal slack through the path sensitized by $TP_X$ be $T_d^\star$. Therefore, faults of size $s < T_m^\star$ are redundant, whereas all faults of size $s > T_d^\star$ can be detected by $TP_X$. The probability that an irredundant fault of size $s$ at $X$ remains undetected is therefore given by

$$
P_X^\star = \int_{T_m^\star}^{T_d^\star} F(s)ds.
$$

The nominal SDQL for a test pattern set is then given by

$$
\theta^\star = \frac{2N}{\sum_{j=1}^{2N} P_{X_j}^\star} = \frac{2N}{\sum_{j=1}^{2N} \int_{T_m^\star}^{T_d^\star} F(s)ds}, \quad (3)
$$

where the number of nodes (delay faults) in the netlist is $N(2N)$. The probability of SDD escape decreases with decreasing $\theta^\star$. Conventional SDD ATPG tools attempt to generate a test-pattern set that minimizes $\theta^\star$.

Next, we will explore how variations in process, supply voltage, and temperature affect the overall yield in conventional 3DICs. There are some key differences between the fabrication process of TSV-based conventional 3DICs and that of monolithic 3DICs. However, the challenges that arise due to die-stacking in conventional 3DICs are also observed in monolithic integration. This includes the die-to-die and the via-to-via coupling effects. Understanding the impact of these defects on delay fault testing in TSV-based ICs is critical, as they will also arise for M3DICs.

### 4.1.2 Delay Faults in Conventional 3DICs

3DICs allow the creation of novel systems that are not feasible by planar fabrication technology—this includes the integration of seemingly dissimilar components like memory, logic, optics, RF, and analog circuits [33]. Additionally, in 3DIC integration, the global 2D interconnects are replaced by short TSV-based vertical interconnects. This, in turn, leads to less capacitive and resistive loading, thus improving the circuit timing.

The impact of systematic and random variations in process, supply voltage, and temperature in 2DICs has been explored in detail in the literature. Reference [11] shows that variations in transistor parameters can result in as much as $20\times$ variation in the chip leakage current and a 30% variation in the operating frequency. In a 180 nm CMOS logic technology, the threshold voltage has been shown to be normally distributed with $\sigma = 30mV$ [11]. The critical paths can also vary from chip-to-chip. Additionally, variations in switching activity across the die often result in uneven supply voltage distribution and temperature hotspots. Within-die temperature fluctuations may cause performance mismatches, which lead to logic or functional failures.

In contrast, even though fabrication imperfections are considered as bottlenecks that need to be overcome to bring 3DIC technology from the lab to the market, the problem of yield improvement of 3DICs has not been explored in detail. The problem of finding the optimal integration of dies into 3DICs is electrically and combinatorially challenging. It can be shown that for three or more integrated die, maximizing the parametric yield for 3DICs is NP-hard [33].

A more tractable version of this problem, with two dies, is explored in a case study in Reference [33]. Here, the authors consider a 3DIC, in which the upper wafer holds a memory die while the substrate is the Central Processing Unit (CPU) wafer. To determine the optimal integration that maximizes the parametric yield, the impact of stand-alone variations in the two die is first observed. Under random process variation scenarios, it is shown that the length of the critical path in the CPU is affected. Due to this, the maximum possible clock frequency of each CPU die varies. Similarly, the memory (L2 cache) access times are also shown to vary in a Gaussian-like distribution. Following a hierarchical approach, the impact of simultaneous variations in the two constituent die on the performance of the 3DIC is subsequently observed. It is shown that due to large variation in the CPU operating frequency and the memory access time under fabrication imperfections, the processor efficiency (calculated in terms of millions of instructions per second) deviates from its nominal value. The authors also point out that the impact of process variation-
induced delay faults often depends on the particular application executing on the processor. For example, a memory-intensive application will be heavily impacted by process variations in the L2 cache die.

The key takeaway here is that, similar to 2D ICs, fabrication imperfections in 3D ICs predominantly affect the propagation delay and can therefore be modeled as SDDs. As discussed above, SDDs are often hard to detect, as they need to be sensitized on a sufficiently long path (with low slack). The quality of an SDD test pattern set is determined by the SDQL metric, which takes the slack on the sensitized paths into consideration. However, note that the propagation delay (and therefore, the slack) of a path can vary under process variations. This necessitates the use of a variation-aware delay fault testing methodology to prevent test escape under process variations.

### 4.2 Sources of Imperfections in the Monolithic 3D Fabrication Process

Although monolithic integration offers significant improvements in terms of design flexibility and energy efficiency, it involves fabrication-related challenges. In M3D ICs, multiple device layers are grown on the same layer in a serial manner [59]; this, in turn, necessitates the following requirement to be satisfied: a low-temperature top-tier annealing process to protect the transistors in the bottom-tier [85].

These requirements have an adverse impact on the transistors in the top tier and interconnects in the bottom tier. Therefore, if these effects are not considered, then it is likely that the performance and energy efficiency of monolithic 3D integration is overestimated.

Contrary to conventional die stacking, epitaxy-based M3D integration cannot reuse the conventional 2D process. Additionally, it is difficult to achieve good crystallized layers of silicon with any known deposition techniques when fabricating monolithic 3D architectures. This results in poor electrical characteristics of transistors and prevents them from achieving high-performance operation [59].

Low-temperature wafer bonding is a key processing step in the monolithic 3D integration flow. The condition of the bonding surfaces plays a crucial role in achieving a defect-free bond [54]. Hydroxyl (OH) groups typically lead to high bond strengths; due to this, oxide layers are the prime candidates for bonding surfaces. As these oxide layers also serve as the inter-layer dielectric, defects introduced during the wafer-bonding step can impact the top layer transistors as well as the MIVs. The sizes of these defects can be in the millimeter to nanometer range, and their presence is often attributed to voids, delaminations, and foreign particles. Due to the increasing process variations at nano-scale technology nodes, it is especially difficult to control the occurrence of nanometer size voids. Thus, understanding the impact of these defects on the functionality and performance of an M3D IC is critical.

The presence of voids at the bond interface reduces the effective back-gate dielectric capacitance, which in turn affects the threshold voltage of the top-layer transistors [54]. Figure 32 shows the different possible alignments of an air-filled void with the channel of a top-layer transistor. Consider the case in Figure 32(a), where the void is perfectly aligned with the top-layer channel. The effective back-gate dielectric capacitance is then given by:

\[
\frac{1}{C_{\text{effective}}} = \frac{t_{\text{box}}'}{\epsilon_{\text{ox}}} + \frac{t_{\text{void}}}{\epsilon_{\text{void}}} + \frac{t_{\text{box}}''}{\epsilon_{\text{ox}}},
\]

where \(t_{\text{box}}\) is the thickness of the back-gate dielectric in the absence of a void; \(t_{\text{box}}'\) and \(t_{\text{box}}''\) are the thicknesses of the dielectric layers above and below the void, respectively. \(t_{\text{void}}\) is the thickness of the void while \(\epsilon_{\text{ox}} = 3.9\) and \(\epsilon_{\text{void}} = 1\) denote the dielectric constants of the inter-layer dielectric (ILD) and void, respectively. Note that as \(\epsilon_{\text{void}} < \epsilon_{\text{ox}}\) the effective back-gate dielectric capacitance will be less than the back-gate capacitance in the absence of the void. For the cases with positive
Fig. 32. Alignment of a void with the channel of a top-layer transistor: (a) perfect alignment, (b) positive misalignment, and (c) negative misalignment [54].

Table 2. Threshold-voltage Values of a Double-gated Device for Transistor-level M3D Integration [54]

<table>
<thead>
<tr>
<th>ILD Thickness</th>
<th>Channel Type</th>
<th>Defect-Free Case</th>
<th>Bubble Thickness</th>
</tr>
</thead>
<tbody>
<tr>
<td>23 nm</td>
<td>P</td>
<td>−0.109 V</td>
<td>−0.101 V</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>−0.067 V</td>
<td>−0.066 V</td>
</tr>
<tr>
<td>100 nm</td>
<td>P</td>
<td>−0.066 V</td>
<td>−0.065 V</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>−0.124 V</td>
<td>−0.123 V</td>
</tr>
</tbody>
</table>

and negative misalignments (Figure 32(a) and Figure 32(b), respectively), the effective back-gate dielectric capacitance will lie between that for the perfectly aligned case and the defect-free case.

Device-level simulations to study the dependence of the drain current ($I_D$) on the front-gate voltage $V_{FG}$ for double gate transistors in the presence of a void in the back-gate dielectric has been performed in Reference [54]. The authors have shown that a significant shift in the threshold voltage is encountered for both P- and N-type devices, especially when a low ILD thickness is used. Table 2 highlights this dependency for a double-gated device for transistor-level M3D integration; corresponding results for gate-level integration can be found in Reference [54]. Note that, as expected, the shift in the threshold voltage increases as the ILD thickness (void thickness) decreases (increases).

4.3 Coupling in M3D ICs

Electrostatic coupling is typically observed in SOI transistors where the front and back gates compete to induce charge in the silicon body; as a result, the electrical state of the silicon film depends on the bias on both these gates [54]. Transistors located in the top layer of an M3D IC can also be considered as SOI transistors, as the top layer is separated from the bottom layer by a dielectric. The impact of coupling on the performance depends on the design partitioning method. In the transistor-level design partitioning approach, the P-channel and the N-channel transistors are placed on different layers. A schematic structure of an M3D IC partitioned at the transistor-level is shown in Figure 33(a). In such an M3D IC, the front gate of the transistor in the bottom layer acts as the bottom gate of the transistor in the top layer [54]. Note also that the front and back gates of the top-layer transistor will be connected in CMOS circuits, and as such, these transistors can be regarded as double-gate SOI transistors. The double gate transistor is asymmetric in nature, as the
front- and back-gate work functions and oxide thicknesses are different. As the back gate controls the electrical state of the transistors in the top and bottom layer, they are coupled to one another.

However, in gate-level partitioning, all transistors in a standard cell are fabricated in one layer. Here, the uppermost metal line from the BEOL of the bottom layer acts as the bottom gate for the top-layer transistor and the ILD acts as the back-gate dielectric. Multiple back-gate configurations are possible in this case; they can be broadly classified into the following types: (i) a metal line from the BEOL of the bottom layer that completely covers the channel, (ii) a metal line from the BEOL of the bottom layer that partially covers the channel, and (iii) a metal line from the BEOL of the bottom layer that overlaps with the channel and is far away. In the first two cases, the voltage on the metal line impacts the electrical state of the channel in a top-layer transistor, thereby coupling the top and bottom device layers.

In Reference [54], the authors have performed device-level simulations to evaluate the impact of coupling on the threshold voltage of a top-layer transistor. For an N-type (P-type) transistor in the top-tier, an increase (decrease) in the threshold voltage is observed when coupling effects are considered. Additionally, there can be some changes in the threshold voltage of a top-layer transistor due to misalignment between the front and back gates. The change in threshold voltage due to a change in the voltage on the metal line decreases when the metal line does not completely overlap with the transistor [54].

4.4 Impact of Coupling and Void Defects on Path Delays

The critical path delay in an M3D IC can vary due to coupling between device layers and on the nature and the size of the void defects arising in the wafer bonding step [54]. Here, the authors consider an example design with 18 two-input gates are considered and simulations are performed after randomly injecting defects on the gates. For transistor-level partitioning, four defects are injected; these include voids in the top-layer (N-type) transistors and at most one MIV defect. It is observed that even in the presence of defects, the path delays do not change significantly when a
MIV thickness of 100 nm is considered. This is in agreement with the results in Table 2, where we observed that the impact of voids on the threshold voltage is insignificant for an ILD thickness of 100 nm. However, notable changes in the path delays are reported for an ILD thickness of 23 nm. This impact is magnified when MIV defects are considered.

For gate-level design partitioning, it is assumed that a void impacts a gate in its entirety; this is done by considering the same defect size for both N- and P-channel transistors in a gate. The back gate voltage of the top-layer transistor is also varied to simulate the impact of coupling on path delays in a gate-level partitioned design. Similar to the transistor-level partitioning case, the average change in the path delay is significant for a low ILD thickness. The impact of manufacturing defects on the path delay depends on the instances in which such defects are inserted. For example, if all defects are inserted in the instances on the same path, or if MIV defects are considered, then path delay is significantly affected. From the results, we can conclude that coupling and wafer-bonding defects can cause a change in the path delays; this, in turn, affects the timing of the circuit, especially when the ILD thickness is less than 100 nm.

4.5 Delay Fault Testing in the Presence of Coupling, Void Defects, and Parameter Variations

Coupling and void defects are typically manifested as SDDs at nano-scale technology nodes. SDDs are difficult to test, as they are detected only when the long paths through the fault site are sensitized. Therefore, commercial SDD ATPG flows use static timing analysis to generate patterns only for paths with a slack below a pre-defined threshold. However, the slack on a path can vary in the presence of fabrication imperfections; therefore, it is necessary to consider these effects while generating test patterns.

The importance of considering coupling, void defects, and parameter variations during SDD ATPG is highlighted using simulation results in Reference [54]. Using the extracted slack data from the static timing analysis tool, test patterns are generated by the timing-aware transition fault ATPG. The effectiveness of the test patterns is then calculated using the SDQL metric for two cases: (i) fault-free circuit: the nominal slack data and the test pattern set used as inputs during fault simulation; and (ii) faulty circuit: the slack data for a defective M3D instance and the test pattern set used as inputs during fault simulation. Over multiple defective instances, it is observed that the SDQL values obtained in the faulty cases are greater than those obtained using the nominal (fault-free) timing library. Representative results for three IWLS benchmarks are shown in Figure 34.

The increase in SDQL shows that fabrication imperfections increase the range and size of defects that are not tested by SDD ATPG techniques that do not take such imperfections into account. Note that the impact of these defects is magnified as the ILD thickness decreases. This is expected, as the degree of coupling and the effect of voids on the back-gate capacitance increases with decreasing ILD thickness. Note also that the propagation delay of a path can either increase or decrease due to coupling and wafer-bonding defects. Therefore, the slack on the path being sensitized by the test patterns can also decrease under fabrication imperfections. Due to these variations in the slack, a decrease in SDQL from its nominal value is observed in some defective scenarios.

Various methods have been explored to model the impact of process variations; however, only a few of them are computationally efficient. For example, statistical static timing analysis (SSTA)-based ATPG methods take variability-aware delay data into consideration to generate efficient test patterns to detect SDDs in the presence of process variations. However, SSTA-based methods, like the one proposed in Reference [23], require multiple dynamic timing analysis runs. The Gaussian pin-to-pin propagation delay distributions are provided as input to the timing analysis. The extremely high test data volume and the large runtime render SSTA-based ATPG methods inefficient for modern designs.
A timing-unaware ATPG tool can be used to activate SDDs on long paths by constraining the set of available endpoints (scan flip-flops) for capture. The paths in the design are first classified into three groups—Long Path, Intermediate Path, and Short Path—based on their length and the minimum sizes of the delay defect detected through the path. During ATPG, only the scan flip-flops on the long paths are considered as observation points while the other flops are masked. This forces the ATPG tool to activate the SDDs through the long paths. A multiple-detect (in this case, a 15-detect) technique is used to further increase the probability of the activation of the long paths. Note that the classification of the paths based on their length is critical to the efficiency of this method; however, this is performed based only on the nominal path delays. However, parameter variations have significant impacts on the propagation delay. Due to this, it is possible that a path that is not classified as a “long path” based on its nominal delay, can be critical under a process variation scenario. Therefore, using this method can result in SDD test escape.

Many process-variation aware delay fault ATPG methods only consider variations that have a linear impact on the propagation delay \cite{12, 66}. In Reference \cite{12}, this linear dependence allows the use of a response surface method technique to achieve a 40x speed-up in simulation over SPICE-based approaches. Similarly in Reference \cite{66}, the path delay is expressed as a linear function of the parameters affected by process variation. The long-path selection problem is then mapped to the polynomial-time feasibility problem in linear programming. However, the assumption of linear dependence limits the applicability of these methods.

5 DESIGN-FOR-TEST SOLUTIONS FOR MONOLITHIC 3D ICs

5.1 MIV Fault Models

The target fault models for MIVs are the same as those for interconnects in an IC with one active (device) layer, because the dimensions of a MIV are comparable to that of vias in today’s ICs \cite{62}. During fabrication, the MIVs are treated as back-end-of-line (BEOL) vias that are susceptible to open and short defects \cite{43}. Therefore, the typical fault models for a MIV are shorts, opens, and SAFs \cite{53, 54}. The MIVs are more likely to be placed in the critical logic paths of the M3D design than conventional BEOL vias. This is because the objective of M3D partitioning is to reduce wirelength and improve timing by inserting a high density of MIVs. As a result, a defect in a MIV is likely to have a higher impact on the chip functionality compared to a defect of the same size.
in a BEOL via. MIV faults can be further classified into hard and resistive categories based on the type and size of the underlying defects. Hard shorts can occur due to imperfect design and circuit synthesis, or particle contamination during fabrication [15]. A resistive short has a resistance in the intermediate range, typically ranging from a few KΩs to several hundred KΩs [76]. Resistive shorts can occur when the MIV metal diffuses through the ILD to make partial contact with another nearby MIV [37], or due to defects at the interface between two tiers after vertical integration of the top tier’s device layer [54].

A hard open occurs when a MIV fails to land on a contact pad. This gap in connection leads to a very high open resistance, typically in the order of MΩs [54, 76]. A resistive open is of relatively smaller size and has a resistance in the intermediate range, typically ranging from a few KΩs to several hundred KΩs [76]. Resistive opens can occur due to bonding defects [54], mechanical stress-induced striations on the underside of a MIV [61], air-voids inside a MIV due to imperfect electro-chemical deposition of metal [35], hairline cracks, and pinhole defects [92].

5.2 Automatic Test Pattern Generation for MIVs

The testing of M3D ICs in general, and MIVs in particular, has remained largely unexplored thus far. Due to the high MIV integration density, retrofitting of conventional interconnect BIST approaches can introduce significant overhead. Methods such as References [77, 86] use dedicated scan elements (test points) for test access. However, these solutions require large test application time, since the number of test patterns required for high fault coverage can become prohibitively large for high MIV density [47]. Moreover, the number of required test points is directly proportional to the MIV count. ATPG-based interconnect test methods, such as Reference [31], are likely be less effective for MIV testing, because I/O pins are available only on one layer in an M3D IC; either test data or test responses—or both in the case of MIVs that do not land on the bottom tier—must be propagated through multiple tiers and the associated MIVs. This requirement adds significantly to the propagation constraints for ATPG. Even if tests can be found by an ATPG tool, additional MIV faults on test paths, which is a likely scenario due to high MIV density, will impede testability. Commercial ATPG tools tend to target single faults for test-pattern generation. However, multiple faults are likely for dense MIV layouts; hence, test escapes might occur if tests are generated under the single-fault assumption.

5.3 Applicability of Pre-bond and Post-bond TSV-Testing Methods

A potential test solution is to extend pre-bond and post-bond testing methods for 3D-stacked ICs to M3D ICs. However, pre-bond TSV testing methods such as Reference [71] are not applicable to MIVs, because bare MIVs cannot be exposed and the current wafer-probe technology cannot support the pitch requirement for MIVs (100 nm to 200 nm) [55].

While post-bond testing techniques for 3D-stacked designs can be extended to post-assembly M3D testing, recently proposed methods such as IEEE Std. P1838 [67] need a die-wrapper register cell on both ends of the MIV for providing test access to the different tiers. This, in turn, significantly increases the associated area overhead due to the large MIV count, which is typically an order of magnitude higher than the number of TSVs in conventional 3D-stacked designs.

5.4 Design-for-test Solutions for M3D ICs

5.4.1 Test Isolation of Bottom Tier. According to Reference [3], the bottom tier’s transistors and metal layers are typically fabricated at the most advanced technology node. As a result, the bottom tier is more susceptible to defects than the top tier and must be tested in isolation to guarantee defect localization and yield ramp-up. Unlike conventional 3D-stacked ICs where pre-bond testing is possible, the bottom tier cannot be tested a priori during the intermediate stages of
M3D integration, as MIVs are not fabricated until the top tier has been processed. In References [17, 96], the authors propose an efficient technique to isolate the bottom tier from the rest of the M3D design to specifically test for defects arising in the bottom tier.

A bypass structure based on programmable e-fuses is used to isolate the bottom tier; Figure 35 illustrates the bottom-tier isolation with an e-fuse. A copper-based e-fuse connects two MIVs—one of them is an input of the bottom tier (downward-going) while the other is the output (upward-going). In this way, pair-wise input and output MIVs of the bottom tier are connected via e-fuses. A transmission gate connected to the input MIV is switched off to prevent signal transmission from the top tier when the bottom tier is being tested. At the same time, the e-fuse is programmed to be highly conductive to prevent test responses from leaving the bottom tier in the test mode. As a result, the e-fuse re-directs the signal from point A in Figure 35 to point B in the test mode. In the functional mode, the e-fuse is disconnected and the transmission gate is switched on. The area overhead of the proposed isolation method is negligible for billion-gate designs. To minimize the wirelength of the pair-wise MIV connections for preventing signal degradation, a maximum-weighted matching is obtained for a bipartite graph where the vertices represent MIVs and edges represent connections between input and output MIVs. The edge weights represent the physical distance between the corresponding input and output MIVs.

In case of unequal number of input and output MIVs, dummy vertices are added to balance the bipartite graph. Connection between a MIV and a dummy vertex is assigned a very large weight (larger than the maximum distance between a pair of input and output MIVs). The maximum-weighted bipartite matching returns the pair-wise MIVs such that the added wirelength is minimized. If a MIV is paired with a dummy vertex, then the MIV is considered to be an independent MIV. To prevent signal degradation, an additional constraint is imposed on MIV pairing, which states that the distance between paired MIVs must not exceed a pre-determined threshold $T$. Therefore, all paired MIVs, whose inter-MIV distance exceeds $T$, are added to a set $H$. Then, the MIVs in $H$ are re-paired such that new pairs are created with distance less than $T$. The remaining unpaired MIVs are considered as independent MIVs. Evaluation on ITC’02 SoC benchmarks (considered as the bottom tiers of a hypothetical M3D design) shows that the wirelength obtained by the proposed MIV-pairing method is 7 to 40 times smaller than the wirelength obtained through random MIV pairing.

The authors propose three alternative DfT structures for observing an independent output MIV of the bottom tier by re-routing its input signal to: (i) an existing ("reused") I/O pad in the top tier via an added MIV, (ii) an added ("new") I/O pad in the top tier via an added MIV, and (iii) a dedicated scan flop in the bottom tier. For the independent input MIVs, their output ends are re-routed to dedicated scan flops in the bottom tier that, in turn, control the corresponding logic gates in the bottom tier. Each of the four proposed DfT structures has a hardware cost. Hence, an optimal combination of these structures needs to be determined to address the problem of
independent MIVs to minimize the total hardware overhead of bottom-tier isolation. An ILP-based optimization problem is formulated with the objective of minimizing the total hardware cost of the added DfT structures for the independent MIVs. The constraints for the ILP solver are: (i) signals corresponding to only independent output MIVs can be re-routed to reused or new I/O pads, (ii) the number of reused pads cannot exceed the number of existing pads, and (iii) the total number of MIVs and scan flops in a subarea of the bottom tier cannot exceed pre-determined thresholds.

5.4.2 Test Method Based on Dedicated Test Layers. In Reference [55], a DfT solution based on dedicated test layers is presented for testing M3D ICs, as illustrated in Figure 36(b). The dedicated test layer is located between two functional tiers of the M3D IC and provides additional controllability and observability for stand-alone MIV testing as well as enable tier isolation for effective fault diagnosis. An interface register, consisting of interface-scan cells, is located in the dedicated test layer for controlling and observing the interface of two adjacent functional tiers. Conventional test structures such as scan chains, test-compression architecture, and IEEE Std. Wrappers are inserted in the functional tiers. Test structures for inter-layer DfT are inserted in the test layer. As different tiers in M3D integration may be fabricated on different technologies, the authors recommend using one test layer per functional tier, except the topmost tier where the I/O pins are located, so different functional tiers can be tested in an isolated manner with a high coverage.

Dedicated probe pads are inserted in all test layers for enabling pre-bond or partial-assembly test using low-force vertical probe cards [13]. The probe pads in the test layer are covered by a passivation layer before depositing the next functional tier. The proposed DfT architecture assumes a package interface that is compliant with IEEE Std. 1149.1 to allow board-level post-assembly testing of a packaged M3D IC. The test layer supports several operating modes such as partial-assembly test via probe pads, post-assembly test via external I/O pins, isolated serial and parallel test of different functional tiers, stand-alone MIV testing, and bypassing the test layer itself.

A comprehensive cost model is also presented in Reference [55] to estimate the cost-per-die for a given DfT solution. The cost model is based on three assumptions: (i) the same rate of occurrence
of spot defects in all the tiers, (ii) spot defects only affect the tier they occur in, and (iii) spot defects arising during wafer-bonding affect MIVs only. The cost model considers the wafer-bonding cost, MIV-processing cost, tier-processing cost (including front-end-of-line and back-end-of-line metal stack), yield of a single tier, MIV process yield, and the number of dies in a wafer to estimate the cost-per-die. The number of dies $N_{\text{die}}$ in a wafer is given by Reference [30]:

$$N_{\text{die}} = \frac{\pi r_{\text{wafer}}^2}{A_{\text{die}}} - \frac{2\pi r_{\text{wafer}}}{\sqrt{2A_{\text{die}}}},$$

where $r_{\text{wafer}}$ is the radius of the wafer and $A_{\text{die}}$ is the area of the die. The proposed DfT solution based on dedicated test layers achieves significant savings in the cost-per-die for several M3D benchmarks and different values of MIV yield compared to the P1838-based method.

Test scheduling is essential for M3D ICs due to their limited power budget and the presence of multiple functional tiers, each containing multiple cores. A test-scheduling methodology using **Integer Linear Programming (ILP)** is proposed in Reference [55] to minimize the wafer-level testing time of M3D ICs for a given power budget and test-access mechanism (TAM) width. The objective is to optimally schedule testing of IEEE 1,500-wrapped cores in the M3D design such that the total test time is minimized while the power dissipation during concurrent testing of cores does not exceed a pre-determined threshold.

5.5 Built-in Self-test of MIVs

5.5.1 Inter-layer BIST Architecture. In Reference [52], an inter-layer BIST architecture is presented for testing MIVs. It is based on the DfT architecture of Reference [55] that mandates a dedicated test layer between the two functional M3D tiers. The dedicated test layer contains interface-register cells whose outputs and inputs are tied to MIVs. Essentially, an upward-going MIV (from bottom to top tier) or downward-going MIV (from top to bottom tier) is divided into two segments—inbound MIV and outbound MIV—which are tied to the input and output of an interface scan cell, respectively. The interface cells, together with their inbound and outbound MIVs, are stitched into a **twisted-ring counter (TRC)**; see Figure 37. In a TRC, the inverted output of the last interface cell is connected to the input of the first cell. The pattern in an $N$-bit TRC repeats every $2N$ clock cycles; here, $N$ is the number of interface cells in the TRC that is also equal to the number of MIVs.
The TRC is clocked at the functional clock frequency for the test pattern generation to enable at-speed testing of resistive defects in MIVs. To observe the test response, the TRC contents are scanned out after a finite number of functional clock cycles. It can be shown that the sequence of test patterns produced by the TRC detects all shorts and opens in the MIVs if the TRC contents are shifted out and observed after $N$ and $2N$ clock cycles.

The stitching of interface cells into a TRC is possible if and only if there exist equal numbers of upward-going and downward-going MIVs. However, this may not be the case in real M3D designs. If the number of upward-going (downward-going) MIVs exceeds downward-going (upward-going) MIVs, then the upward-going (downward-going) MIVs are clustered so the number of clusters of upward-going (downward-going) MIVs equals the number of downward-going (upward-going) MIVs. Clustering is done based on the Manhattan distance between the MIVs; this ensures that MIVs in the same cluster are located in close proximity and have a higher likelihood of the occurrence of shorts. A cluster of MIVs is then visualized as a graph where a MIV represents a vertex and an edge indicates a possible short between two MIVs in the cluster. Graph coloring is then performed with two colors to deliver complementary inputs to MIVs, which are likely to get shorted, for detecting shorts. The output of a driver interface cell is inverted accordingly when connecting to a MIV in the cluster. The outputs of the MIVs in a cluster are aggregated using an AND gate and fed to a load interface cell.

To minimize the impact of the added test structures on circuit power and timing, wirelength minimization is required while stitching the interface cells into a TRC. The authors map the wirelength minimization problem during TRC stitching to finding a minimum-cost Hamiltonian circuit in a complete bipartite graph, where a vertex represents a MIV (or MIV cluster) and an edge represents a connection between upward-going and downward-going MIVs (or MIV clusters). As the problem is NP-Complete, the authors propose a heuristic that outperforms greedy TRC stitching on several benchmarks.

The authors further present a machine learning–based framework for the diagnosis of defect size in MIVs. A three-layer feedforward neural network is trained to predict the defect-size bin of a resistive defect that has been detected by the TRC. The features used for training are Hamming distance between fault-free and faulty TRC contents after $N$ and $2N$ cycles and the functional clock frequency. The proposed method achieves an accuracy of more than 90%.

5.5.2 XOR-BIST Architecture. The proposed BIST architecture for testing shorts, opens, and SAFs in MIVs is shown in Figure 38. On the output side of the MIVs, we insert 2-input XOR gates.
between adjacent MIVs. There are \((N - 1)\) XOR gates for a set of \(N\) MIVs. The XOR outputs feed a space compactor that is an optimally balanced AND tree with \((N - 1)\) inputs and a 1-bit output signature \(Y_1\). We can determine whether there is a fault in the given set of MIVs by observing \(Y_1\). Test data is fed to the MIV inputs from an input source \(V_{in}\), which provides complementary signals to adjacent MIVs in the test mode via an inverter chain. A 2:1 multiplexer is present at MIV’s input to switch between functional input (FI) and test mode based on the Launch signal.

The MIVs are tested in two clock cycles by switching \(V_{in}\). The test patterns to the MIVs are “101...” in the first cycle and “010...” in the second cycle. It can be shown that a set of MIVs contains no hard faults if and only if \(Y_1 = 1\) in both clock cycles. The only scenario where aliasing occurs is when all MIVs are alternately stuck at 0 and 1, leading to masking of the MIV faults; however, the probability of such a scenario is \(\frac{1}{2^n}\) for \(n\) MIVs. The manner in which the MIVs are driven in the test mode leads to a deterministic hard-short behavior; this is illustrated in the inset of Figure 38. If two MIVs are shorted, then the MIV appearing first (pre-MIV) in the path of the incoming test signal from \(V_{in}\) will drive the other MIV (post-MIV) because of the lower resistance path through the hard short.

5.5.3 Dual-BIST Architecture. The BIST design of Section 3.2 is also susceptible to SAFs, which may mask MIV fault(s). To minimize the masking probability, a parallel propagation path is added from the MIV outputs to a second 1-bit signature \(Y_2\). The physical structure of this parallel path to \(Y_2\) (BIST-B) is identical to that of the path from the XOR inputs to \(Y_1\) (BIST-A). The XOR and AND gates in BIST-B are replaced with their logical dual counterparts (XNOR and OR, respectively) in BIST-A, as shown in Figure 39. The MIVs under test and the “dual-BIST” engine are fault-free if and only if \(Y_1 = 1\) and \(Y_2 = 0\) for both test vectors. With this “dual-BIST” architecture, it can be shown that MIV fault(s) cannot be masked by a single BIST fault. Moreover, the masking probability due to multiple BIST faults is very low.

The M3D design flow Shrink-2D [73] is used to synthesize the block-level partitioned benchmarks listed in Table 3. The dual-BIST is inserted tier-wise to generate 3D BISTed designs. Table 4 presents the power consumption and area overheads of the BIST-inserted Rocketcore design (BI) relative to the non BIST-inserted design (N-BI). The impact of dual-BIST on the circuit’s PPA is observed to be minimal.

5.5.4 Automation of Non-intrusive Dual-BIST Insertion. Our custom in-house tool for BIST insertion, implemented using Python, takes as inputs: (1) tier-partitioned gate-level netlists of the M3D tiers, (2) target ILVs to be BIST-inserted, and (3) target number of scan chains to insert in the full M3D design. Note that one dual-BIST engine tests a group of eight ILVs and accordingly, using (2), the tool determines the number of dual-BIST engines to be inserted in the M3D design.
Table 3. M3D Benchmarks Used for PPA Comparison

<table>
<thead>
<tr>
<th>Name</th>
<th>Footprint (μm × μm)</th>
<th>Cell count</th>
<th>MIV count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rocketcore [2]</td>
<td>700 × 700</td>
<td>301,219</td>
<td>1,200</td>
</tr>
<tr>
<td>(f_m = 350)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVC-Nova [1]</td>
<td>420 × 420</td>
<td>134,547</td>
<td>317</td>
</tr>
<tr>
<td>(f_m = 366)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AES-128 (I) [1]</td>
<td>350 × 350</td>
<td>102,278</td>
<td>425</td>
</tr>
<tr>
<td>(f_m = 2273)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AES-128 (II) [1]</td>
<td>350 × 350</td>
<td>90,233</td>
<td>426</td>
</tr>
<tr>
<td>(f_m = 1250)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Impact of Dual-BIST on PPA

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Metric</th>
<th>N-BIST</th>
<th>BI</th>
<th>Overhead (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rocketcore</td>
<td>Cell area (μm²)</td>
<td>382,037.6</td>
<td>389,785.6</td>
<td>2.03</td>
</tr>
<tr>
<td></td>
<td>Wire-length (m)</td>
<td>7.03</td>
<td>7.17</td>
<td>1.97</td>
</tr>
<tr>
<td></td>
<td>Power (mW)</td>
<td>227,3607</td>
<td>232,5544</td>
<td>2.28</td>
</tr>
<tr>
<td>AVC-Nova</td>
<td>Cell area (μm²)</td>
<td>196,841.5</td>
<td>199,038.7</td>
<td>1.12</td>
</tr>
<tr>
<td></td>
<td>Wire-length (m)</td>
<td>3.02</td>
<td>3.04</td>
<td>0.61</td>
</tr>
<tr>
<td></td>
<td>Power (mW)</td>
<td>87,46578</td>
<td>90,79729</td>
<td>3.81</td>
</tr>
<tr>
<td>AES-128</td>
<td>Cell area (μm²)</td>
<td>132,049.9</td>
<td>134,771.2</td>
<td>2.06</td>
</tr>
<tr>
<td>(I)</td>
<td>Wire-length (m)</td>
<td>1.68</td>
<td>1.7</td>
<td>1.24</td>
</tr>
<tr>
<td></td>
<td>Power (mW)</td>
<td>178,5411</td>
<td>194,3433</td>
<td>8.85</td>
</tr>
<tr>
<td>AES-128</td>
<td>Cell area (μm²)</td>
<td>103,632.4</td>
<td>106,324</td>
<td>2.6</td>
</tr>
<tr>
<td>(II)</td>
<td>Wire-length (m)</td>
<td>1.61</td>
<td>1.64</td>
<td>1.35</td>
</tr>
<tr>
<td></td>
<td>Power (mW)</td>
<td>110,96</td>
<td>119.42</td>
<td>7.63</td>
</tr>
</tbody>
</table>

Using (1) and (2), dual-BIST insertion is carried out in a non-intrusive manner; Figure 40 illustrates the dual-BIST insertion method. Using (3), the tool generates a dofile for scan-chain insertion that follows the BIST-insertion stage. Using the generated dofile, scan chains are inserted in the BIST-inserted M3D design using Mentor Tessent.

6 OPEN PROBLEMS AND OPPORTUNITIES FOR RESEARCH

Tier partitioning approaches can be further improved by considering power consumption, voltage-drop, and thermal coupling in heterogeneous multi-tier M3Ds. For example, power consumption should be considered in future studies as a factor for tier partitioning.

Power delivery for M3D ICs is also challenging because of the large power MIV number and irregular MIV placement caused by standard cell blockage [20]. Therefore, developing EDA approach to building reliable M3D PDNs remains an open problem for future studies.

For 3D placement, mitigating timing degradation remains a major challenge for both true-3D and pseudo-3D placers. In pseudo-3D methods, timing degradation is introduced when converting the intermediate 2D placement into 3D, which is primarily due to the errors of pin location and RC extraction. In the case of true-3D placers, none of them are yet to be integrated with commercial back-end flow including routing and timing closure. This hinders their applicabilities in real-world designs and power, performance, area (PPA) optimizations.

To better understand the thermal properties of M3D ICs, more studies need to be done to quantify the heat dissipation rate and the impact of MIV density, and also compare these properties with conventional 3D ICs.

More efforts are needed to develop “true-3D clock routers that are tightly integrated in commercial EDA tool flow.” Most existing routers presented in academia still remain as exploratory work with potential but never validated with real-world designs. Those clock routers found in
pseudo-3D flows such as Shrunk-2D, Compact-2D, or Macro-3D [5, 56, 73] are still pseudo-3D: An intermediate 2D clock tree built with a commercial 2D clock router is transformed into a 3D clock tree using partitioning. A post-partitioning refinement is used to improve the quality, but this pseudo-3D approach may not be as good as true-3D clock trees, the ones created from scratch with multiple tiers of FFs in mind.

Design optimization for thermal hotspot mitigation still needs more studies and innovations. Thermal-aware floorplanning that avoids vertical overlaps among high-power consumption modules seems to work reasonably well. But they come with degradation in PPA. Dummy thermal vias also provide some limited reliefs, but whitespace may not be available at the places we desire. Micro-fluidic channels are promising, but their practicality and cost issues remain unsolved. Thermal mitigation calls for strong collaboration among chip designers, package designers, material scientists, and foundries to come up with cross-disciplinary and holistic solutions.

For early detection and diagnosis of novel M3D-specific failure modes in the field, significant test effort is required for improving the reliability of M3D designs. Variation-aware test-pattern generation needs to be explored for detecting small-delay defects in the presence of inter-tier coupling. Investigating the coupling between MIVs and active devices will further contribute to the development of advanced fault models and M3D-aware test generation. For high-resolution diagnostics, the granularity of fault localization can be enhanced via strategic insertion of low-cost test structures in different M3D tiers; this will enable efficient failure analysis of tier-specific defects and yield learning.
7 CONCLUSIONS

We have witnessed significant development and wide applications of emerging 3D technology in recent years. For monolithic ICs (M3D), the exceptionally high 3D-interconnect density and the unique low-temperature fabrication process propose numerous challenges for physical implementation and design-for-test. In this article, we presented a comprehensive survey of the state-of-the-art EDA solutions for tier partitioning, placement, clock delivery network design, and thermal analysis for M3D. In addition, we discussed the state-of-the-art DfT solutions as well as the need for rigorous approaches to delay fault testing for M3D designs. This enables the implementation of reliable commercial-grade M3D ICs, and therefore paves the road to commercialization of the M3D technology.

REFERENCES


Received October 2020; revised April 2021; accepted June 2021