PowerSynth Design Automation Flow for Hierarchical and Heterogeneous 2.5-D Multichip Power Modules

Imam Al Razi, *Graduate Student Member, IEEE*, Quang Le, *Student Member, IEEE*, Tristan M. Evans, *Member, IEEE*, Shilpi Mukherjee, H. Alan Mantooth, *Fellow, IEEE*, and Yarui Peng, *Member, IEEE*

***Abstract*—As a critical energy-conversion system component, power semiconductor modules and their layout optimization has been identified as a crucial step in achieving the maximum performance and density for wide bandgap technologies (i.e., GaN and SiC). New packaging technologies are also introduced to produce reliable and efficient multichip power module (MCPM) designs to push the current limits. The complexity of the MCPM layout is surpassing the capability of a manual, iterative design process to produce an optimum design with agile development requirements. An electronic design automation tool called PowerSynth has been introduced with on-going research toward enhanced capabilities to speed up the optimized MCPM layout design process. As a part of this continuing research, in PowerSynth v1.9, a constraint-aware layout engine has been developed, which enables integrating heterogeneous components, handling complex geometry, exploring a larger solution space, improved success rate, and providing options for multiobjective optimization algorithms. The layout engine is generic, scalable, and efficient in performing electro-thermal optimizations on both 2-D and 2.5-D power modules. To validate these enhanced design capabilities, a 2.5-D full-bridge power module layout is designed, optimized, fabricated, and tested with measurement results matching closely with model prediction. This result closes the loop in the power electronics design process with an experimentally validated module design automation flow.**

***Index Terms*—Corner stitch and constraint graph, hierarchical and heterogeneous, layout optimization, multichip power module (MCPM), PowerSynth.**

Manuscript received May 15, 2020; revised August 28, 2020, November 5, 2020, and December 10, 2020; accepted December 22, 2020. Date of publication January 8, 2021; date of current version May 5, 2021. This work was supported in part by the National Science Foundation under Grant EEC-1 449 548 and in part by the Army Research Lab under Contract W911NF1820087. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the National Science Foundation or Army Research Lab. Recommended for publication by Associate Editor K. Sheng. *(Corresponding author: Imam AlRazi.)*

Imam Al Razi and Yarui Peng are with the Department of Computer Science and Computer engineering, University of Arkansas, Fayetteville, AR 72701 USA (e-mail: ialrazi@uark.edu; yrpeng@uark.edu).

Quang Le, Tristan M. Evans, and H. Alan Mantooth are with the Department of Electrical Engineering, University of Arkansas, Fayetteville, AR 72701 USA (e-mail: qmle@uark.edu; tmevans@uark.edu; mantooth@uark.edu).

Shilpi Mukherjee is with the Microelectronics-Photonics, University of Arkansas, Fayetteville, AR 72701 USA (e-mail: sxm063@uark.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPEL.2021.3049776.

Digital Object Identifier 10.1109/TPEL.2021.3049776

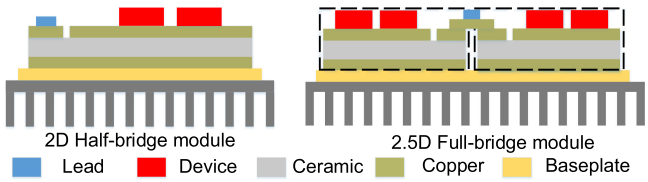


Fig. 1. Cross-sectional view of the 2-D half-bridge power module (left) and 2.5-D full-bridge power module (right).

I. Introduction

RECENT advancements in electric automobiles, aircraft, smart grid, and consumer devices all require next-generation power conversion circuits and systems with extremely high density, efficiency, and reliability [1], [2]. These unprecedented demands are accelerating the design and packaging evolution of power modules by integrating wide bandgap (WBG) devices (e.g., SiC/GaN) and passive components [3], [4]. Moreover, multichip power module (MCPM) layouts need to be optimized carefully to push the power density with smaller filter size, and increased switching frequency [5]–[7]. Novel packaging and design techniques are introduced to minimize parasitics, improve cooling, increase power density, and reduce voltage overshoot [8], [9]. These techniques include integrating multiple substrates into a single package, flip-chip wirebondless modules, and packaging heterogeneous components like gate drivers, heat sinks, thermal sensors and switches, decoupling capacitors along with power devices.

Among these advanced techniques, 2.5-D power modules represent the latest power packaging technique with an agile product development concept. Though 2.5-D design has been adapted in very large scale integrated (VLSI) circuit and package design, 2.5-D packaging is a relatively new concept in power electronics. 2.5-D package refers to a planar substrate consisting of multiple active substrates connected with additional routing resources. For example, two 2-D half-bridge power modules can be placed on a single baseplate to form a 2.5-D full-bridge power module (shown in Fig. 1). A direct bonded copper (DBC) or a low temperature cofired ceramic (LTCC) interposer can be used to form an electrical reliable connection bridge between these two substrates [10], [11]. On the other hand, 2-D layout consists of a single routing and device layer as shown in Fig. 1(left).

Currently, the power module design process is a manual, repetitive, tedious task that requires expertise on several tools and software [12], [13]. To come up with a satisfactory solution, the designer requires several iterations of finite-element analysis (FEA) to account for changes in the trace orientation, device spacing, etc. Due to the conflicting nature of electrical, thermal, and mechanical aspects of a power module, one trial to improve one aspect affects the other one. Therefore, it generally takes at least four to six iterations in the industry for an expert to come up with a satisfactory design ready for fabrication [14]. Currently, there is no standard computer-aided design (CAD) tool available to verify real-world performance before manufacture. Once the module is fabricated and if the module does not perform according to expectation, a costly redesign is required. It is evident that with the ever-growing complexity of the MCPM layouts, this manual and iterative design process is not capable of generating high-performance designs that are both cost and time-effective. Also, the solution space is often restricted due to limitations in engineering time. As such, recently, design automation tools with multiobjective optimization have been investigated by the power electronics society to explore a large variety of designs, reduce cost and computational effort, and alleviate design complexity [5], [6], [15]–[20].

In [5], the authors proposed an automatic layout generation method to consider a legitimate tradeoff between electromagnetic compatibility (EMC) and thermal constraints. A simplified EMC and a thermal model are applied individually to generate a "Thermal-EMC" plane with Pareto-front solutions. Though the proposed method may determine a theoretical optimum solution for a given technology, it is hindered by limited solution space and design rule violations. In [20], a sequentially coupled approach is proposed for optimizing power module layouts by integrating a few FEA modeling tools (e.g., ANSYS). These tools are coupled in a multidisciplinary design optimization framework to interact with each other while generating solution layouts. Though the methodology has been proven superior to the traditional manual design flow, some limitations like time-consuming finite-element method (FEM) for electro-thermal evaluation, few design variables in the solutions, and a single variable in each iteration of the optimization makes this methodology infeasible for 2.5-D layout optimization. Another research also adapted the placement and routing concept from the VLSI design automation area and implemented in power electronics design automation [6]. The components like die, wire bonds, connected traces are merged into a single rectangle to simplify the problem. Relative position, component orientation, and the gap among components are translated into a binary string, which is manipulated in the optimization phase to generate new solutions. Two-folded optimization is performed by a genetic algorithm, where the outer loop aims at placement optimization via the sequence pair method [21], and the inner loop focuses on routing. The optimization cost function involves only footprint area and electrical parasitics, which raises concern about the thermal reliability of the solutions. Also, simplified representation saves computation time by sacrificing accuracy in calculating parasitics. Though the methodology has been extended to 3-D layout optimization [22], no hardware-validation has been presented even with a simplified technology.

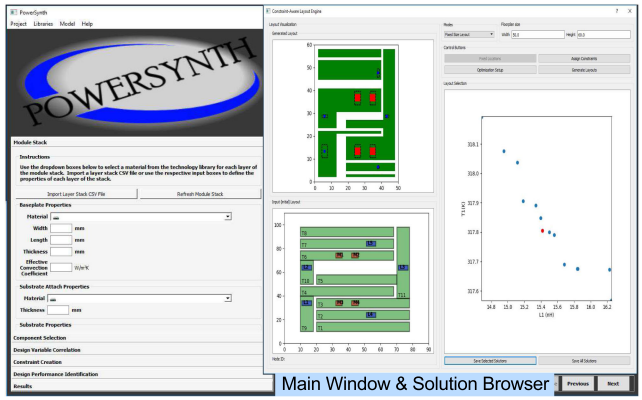


Fig. 2. PowerSynth graphical user interface.

PowerSynth Evolution: The initial work in [23] has introduced a simultaneous electro-thermal optimization methodology named "Power-CAD" for design, analysis, and optimization of a discrete power module. Power-CAD has been proven to be both cost and time-effective for the power module design industry by significantly reducing the number of design cycles. After about a decades of research and development effort, a complete EDA tool "PowerSynth" for MCPM layout optimization was presented in [15]. PowerSynth integrates fast, accurate, hardware-validated electrical, and thermal models for MCPMs within a multiobjective optimization framework. This published version uses a symbolic layout consisting of lines and points to represent a power module. It has a built-in technology library and manufacturer design kit (MDK) to account for a wide range of materials and design rules from the manufacturer. Another important capability is the back-annotation of layout-extracted parasitics to the original circuit schematic, which enables round-trip engineering before fabrication. Also, exporting 3-D layouts to several commercial FEA tools is another must-have feature for detailed analysis. PowerSynth is the first tool that focuses on MCPM layouts, and research has been continuing to enhance its capabilities and make it more generic, efficient, and scalable. In [16], a planar representation of the power modules based on a customized corner stitch [24] data structure with constraint graph [25] evaluation methodology has been proved to successfully overcome limitations with the symbolic layout method. Further research has proved that hierarchical corner stitch and constraint graph data structure and algorithms are more effective for power module layout synthesis and optimization [17]. These recent efforts have enabled PowerSynth to process more complex geometry and lead toward 3-D power module layout optimization. The graphical user interface (GUI) of the PowerSynth is shown in Fig. 2.

The following key contributions and new features are highlighted in this article.

1) A new layout representation format that replaces lines and points with rectangles and hence is capable of representing all planar configurations. This new format can represent any Manhattan geometry with an arbitrary number of components, which makes it more scalable and generic.
2) A generic, scalable, and efficient layout engine based on the hierarchical corner stitch data structure with constraint

graph evaluation methodology, suppressing the previous matrix-based layout engine. It can generate not only fixed-floorplan sized solutions but also minimum-sized, and variable-sized solutions. These new capabilities provide more design options with a larger solution space.

3) Constraint-aware layout generation methods that remove the necessity of the design rule checking (DRC) and always produce DRC-clean solutions. Design rules consideration in the layout generation phase accelerates the solution generation process and guarantees 100% manufacturable solutions. The design rules can be easily modified by the user through an exposed interactive interface.
4) Generic connection handling algorithms, for both intralayer and interlayer connections. The interconnect handling algorithms are implemented in a generic way so that the algorithms can handle different types of connections.
5) A hardware-validated experimental design study showing a complete PowerSynth-enabled 2.5-D MCPM design flow. Finally, a 2.5-D power module design is chosen and optimized hierarchically using the proposed CAD flow. The optimization results are verified against physical measurements. The previous work on PowerSynth [15] focused on validating the models developed for it and their application in a multiobjective optimization routine. In that work, test vehicles are used to validate the models and a full optimization of a 2-D half-bridge module is presented without final verification of an optimized solution module through fabrication. However, this work goes further by not only validating the optimization results but also with major updates in the layout handling capability.

Section II of the article introduces a brief overview of PowerSynth's latest framework. Section III describes the layout generation methodology of the new layout engine. Section IV provides a brief description of the electrical and thermal model, along with optimization algorithms. Section V shows some sample cases designed by PowerSynth and optimization results for a 2.5-D full-bridge power module design. Section VI presents the validation of the optimization results through fabrication and measurements. Finally, Section VII concludes the article and describes the future work.

II. PowerSynth Architecture

Since the very first version of PowerSynth [15], some new models were developed to incorporate new objectives for optimization, and some features were added to enable more capabilities over time. The latest architecture in the v1.9 release (installation package can be downloaded from [26]) is shown in Fig. 3.

1) Manufacturer Design Kit: In the integrated circuit (IC) design industry, the process design kit is one of the most important pieces of intellectual property from foundries to enable the design process. Similarly, for power module fabrication, a certain rule set has to be followed from the beginning of the layout design. This ruleset includes materials, devices, leads, wire bond information, and design constraints with a technology layer stack. To ensure fabrication-ready layouts, an MDK has been integrated with PowerSynth that contains an interactive material library, design constraints interface, and technology library. All the libraries are interfaced with graphics so that the user can edit the libraries, if necessary. A PowerSynth MDK contains all the required information that an MCPM designer needs to design a power module with ensured manufacturability, improved reliability, and maximized efficiency.

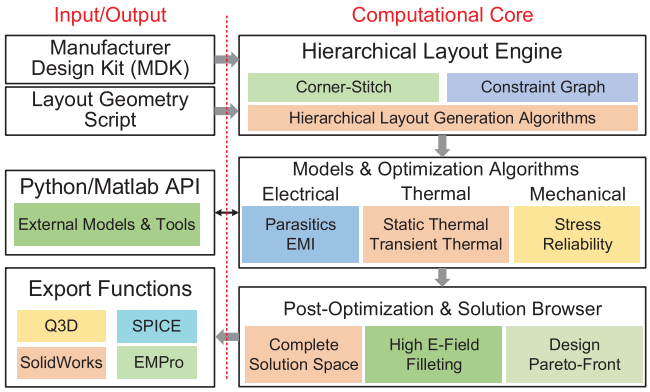


Fig. 3. PowerSynth architecture (v1.9).

TABLE I
DRC-Clean Solution Generation Comparison

Devices #	% of DRC-Clean Layouts		Average Time(s) (per DRC-clean solution)		
	Old	New	Old	New	Speedup
4	15.75	100	8.8	0.161	×54.65
6	20.47	100	5.49	0.207	×26.52
8	15.75	100	3.94	0.272	×14.46
10	12.71	100	2.6	0.349	×7.45
12	15.17	100	1.99	0.447	×4.45

2) Hierarchical Layout Engine: The methodology of generating a layout solution is the backbone of the tool. In the latest version, the matrix-based methodology has been replaced by a more generic, efficient, and scalable one using the hierarchical corner stitch data structure with constraint graph evaluation techniques. This layout engine takes design constraints from the MDK together with an initial geometry script from the user as input to process the layout. With this methodology, an arbitrary number of components can be handled by applying generic and time-efficient algorithms. Also, this approach eliminates the design rule check (DRC) step. The efficiency comparison between the corner-stitched data structure with the constraint graph methodology and the traditional matrix-based methodology has been shown in [16]. The previous layout engine generated solutions are not always DRC-clean, which shows a variable efficiency with varying number of devices in the same layout. To compare the percentage of DRC passed solutions between two methods, a half-bridge power module layout is chosen, and the number of devices are varied from 4 to 12 with an increment of 2. For each case, total 15 015 solutions are generated. The comparison result is shown in Table I. For each case, the new layout engine generates 100% DRC-clean solutions, and among all cases, the old one only generates a maximum of 20.47% DRC-clean solutions. Therefore, the new layout engine has enhanced the capabilities of PowerSynth to a great extent. A detailed description is provided in Section III.

3) Performance Evaluation Models: Though PowerSynth has been performing electro-thermal optimizations since the first release, a few more models are developed recently to consider more objectives like EMI, stress, and partial discharge. Application of the EMI model for power module layout evaluation and optimization is presented in [27]. Besides, an updated electrical model [28] can consider both self and mutual inductance, resistance, and capacitance. Further discussion on the electrical model is provided in Section IV. The latest PowerSynth architecture is based on a modular approach so that the optimizer is flexible in accepting various models through application programming interfaces (APIs) while evaluating performance metrics. For example, the Army Research Lab (ARL) has developed a 3-D thermal and stress evaluation tool called ParaPower [29]. To incorporate their thermal and stress model, a PowerSynth-ParaPower API has been developed to allow PowerSynth to use ParaPower for thermal and stress evaluations.

4) Optimization Algorithms: In the first version of PowerSynth, only a nondominated sorting genetic algorithm (NSGA-II) [30] has been used for optimization since the matrix-based methodology is not compatible with some other optimization algorithms. Though NSGA-II can converge faster, it has certain limitations: Many generations are required before convergence; the distribution of the weights to the objectives is complicated; generated solutions are not guaranteed to be globally optimum; solution space is not large enough. These limitations initiated a study toward assessing the viability of other optimization algorithms.

5) Postoptimization and Solution Export: In the latest version, along with the Pareto-front solutions shown in an interactive solution browser, an entire solution space is also reported after optimization. Moreover, for each solution, the layout geometry is also exposed in a CSV file containing each component's coordinates, width, and length. This information helps a designer to regenerate the geometry script for the solution layout. Users can also perform filleting to the sharp corners to reduce electric field concentration as well as partial discharge problems.

To perform a detailed analysis, the user can export PowerSynth solutions to commercial FEA tools like ANSYS Q3D, EMPro, and SolidWorks. PowerSynth can generate a script that creates a 3-D model of the module in these tools automatically. Then, the user can perform detailed FEA analysis for thermal and parasitics. The exported SPICE netlist has been updated with mutual inductance values and can be back-annotated for subsequent analysis in a circuit simulator.

III. Constraint-Aware Layout Engine

The new constraint-aware layout engine has enabled PowerSynth to generate 100% DRC-clean solutions for more complicated 2-D and 2.5-D modules. The hierarchical layout generation is generic and scalable enough to expand its capability to handle multilayer 3-D layouts.

A. Layout Geometry Script

The layout engine takes the initial layout as input through the geometry script, which is one of the fundamental upgrades in the new version of PowerSynth. Previously, a symbolic layout was used to represent the placement and routing of the power module components. This leads to some severe limitations. E.g., in the symbolic layout representation, traces and wire bonds are represented as lines and the devices, and the leads are represented as points. Also, wire bonds are always perpendicular to their landing traces. This representation technique limits PowerSynth to process simple geometries only. Since symbolic layout uses 1-D lines to present traces, it cannot handle geometries with planar traces containing multiple noncollinear devices easily. With the updated layout description script, a layout is described in a hierarchical manner, where each component is considered as a group of rectangles. This rectilinear representation ensures to process any kind of Manhattan structure, and the hierarchical representation ensures to place an arbitrary grouping depth of any components. Hierarchical placement refers to the order of component insertion. For example, in a power module, a trace typically encloses devices, and each device encloses pads. So, traces need to be declared before the devices and devices before the pads. A sample 2.5-D power module layout, and the corresponding tree structure are shown in Fig. 4(a), and (c), respectively. Since this 2.5-D full-bridge module layout is symmetrical, it can be split into two subdesigns. The geometry script shown in Fig. 4(b) represents the highlighted half-bridge power module in Fig. 4(a). Here, L1, L2, L3, L4, L5, L6, and L7 represent DC_1-, DC_1+, Out_1, KH, GH, GL, and KL, respectively. In this example script, the indentation refers to the hierarchy level. Here, T, L, D, and C stand for trace, lead terminal, device, and capacitor, respectively. R270 means the device is rotated by 270°. The directly connected rectangular traces are inserted in a group-wise manner. Each new group starts with a "+" character while a "-" character represents the continuity of that group. Each trace component is represented by six fields: name, type, lower-left X,Y coordinates, width, and length. Each terminal or device representation has four or five fields: name, type, bottom-left corner X, Y coordinates, and an optional orientation field. The technology library contains the terminal and device dimensions. In the layout geometry script, the device type (i.e., MOS, IGBT, Diode) and terminal type (i.e., power lead, signal lead) are directly considered. This type information is used to find the appropriate device and terminal from the technology library. Once the specific component is found, the corresponding dimensions, material information are loaded automatically. These information are then used while inserting the components in the appropriate hierarchical plane.

B. Data Structure

Since the corner stitching data structure and the constraint graph evaluation technique has been adapted from VLSI and customized for power modules, a brief overview of the basic corner stitch and constraint graph is introduced here.

Corner Stitch: The basic corner stitch data structure was introduced by John Ousterhout [24] and widely used in VLSI CAD tools. In this data structure, there are two types of nonoverlapping rectangular tiles: solid and empty. Four pointers are used in each tile to traverse the layout area efficiently. The planar corner stitch

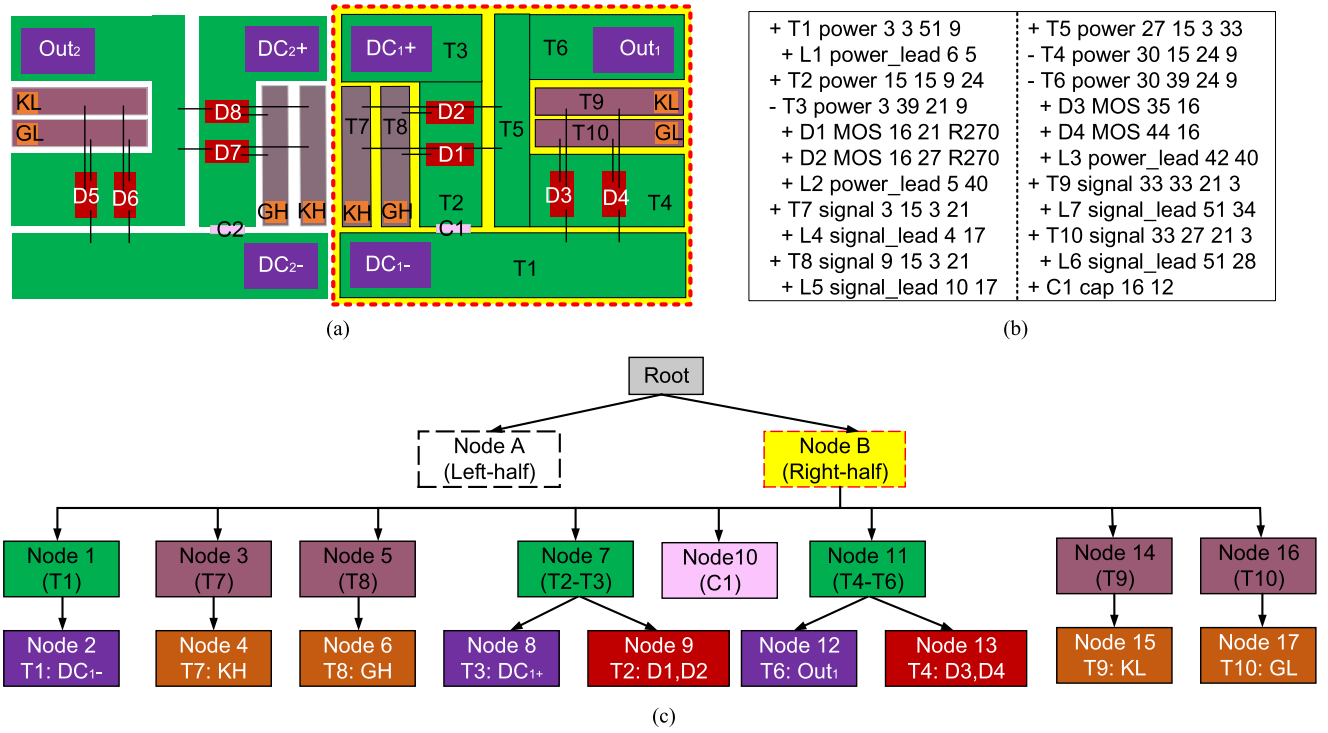


Fig. 4. (a) Two half-bridge modules of a 2.5-D full-bridge power module, (b) input geometry script for the right half module, (c) tree structure of the full-bridge module.

has two orientation types: horizontal corner stitch (HCS) and vertical corner stitch (VCS). The basic rules for creating HCS (and VCS) are: (1) each tile must be as wide (tall) as possible. (2) after satisfying rule (1), each tile must be as tall (wide) as possible. Due to the linear time complexity of associated algorithms (e.g., insert, merge, search) to create corner stitch data structure and the convenience of obtaining necessary design constraints, this data structure has been extended to represent power module layouts in PowerSynth. Horizontal and vertical corner stitch planes of a sample layout are shown in Fig. 5.

Constraint Graph: Constraint graph is a computation technique for a set of inequalities. In this graph, if a vertex A should always maintain a minimum distance of W from another vertex B, then the relationship between A and B can be expressed as

$$B - A \geq W. \tag{1}$$

Here, A is the origin, and B is the destination. To maintain the minimum design constraints among all components, two types of constraint graphs (CGs) are created: horizontal (HCG) and vertical (VCG). HCG maintains the relative location among components horizontally, and VCG vertically. For each corner-stitched plane, two constraint graphs are used to maintain technology constraints, where coordinates are mapped into vertices, and the constraints are mapped into edges. Sample HCG and VCG are shown in Fig. 5.

Hierarchical Corner Stitch Data Structure: The original corner stitch data structure is a representation of a 2-D plane, in which tile overlapping is not allowed. This data structure is modified to represent power module layouts. Generally in power layouts, die and leads are placed on top of traces while pins and

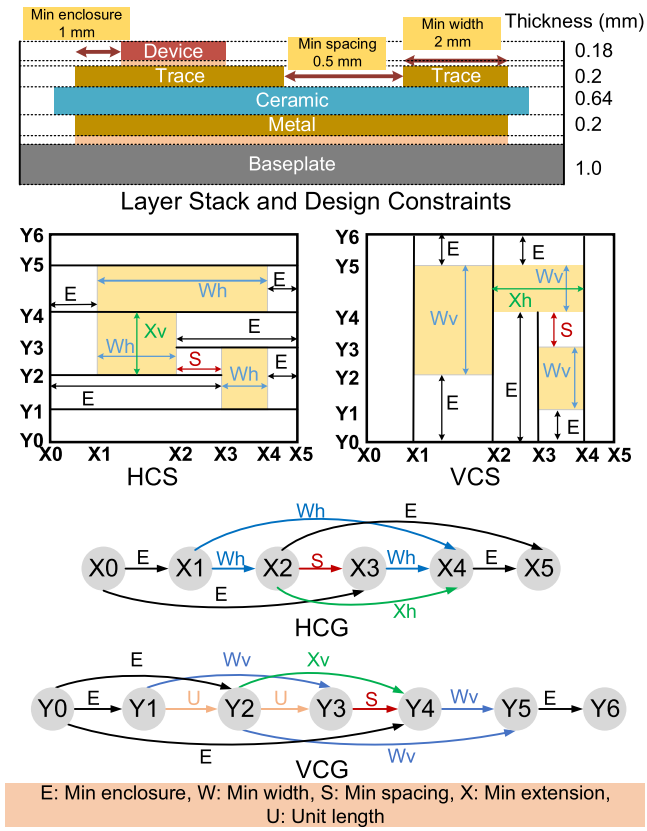


Fig. 5. Layer stack and design constraints illustration, corner-stitched planes (HCS/VCS) and corresponding constraint graphs (HCG/VCG).

wire bonds are placed on top of die. To handle hierarchical placement of components in the layout, a tree structure is maintained in the layout engine, and each node of the tree is a corner-stitched plane. Due to the hierarchical representation, two additional tile types are considered: foreground and background. When a tile is placed on top of another, the former one is treated as the background and the latter one foreground. For the symmetrical example in Fig. 4, no additional computation is required for the left-half. As T2 and T3 are connected, they are in the same node (Node 7). D1 and D2 are inserted on top of T2, which makes D1, and D2 foreground tiles and T2 background tile in Node 9. To generate layout solutions, first, the layout is drafted hierarchically and converted into the layout geometry script format. All components need to be declared before using them. A high-level workflow of the layout generation methodology is shown in Algorithm 1. The geometry script is read-in and the hierarchical tree is created based on the input script. A root node is created with the empty tile having the outline of the floorplan. For each symmetrical part, a new node is created as a child node of the root. Then, for one child node, the complete sub-tree [as shown in Fig. 4(c)] is created using the input geometry information. For each node in the sub-tree, corner stitch algorithms have been applied to create HCS and VCS planes. Pseudocode for corner stitch plane creation is summarized in Algorithm 2. Here, for each node, the insert operation is applied to create HCS and VCS planes. Since the connected components are in the same node, the inserted tile searches for such tile in its direct neighbor list and merge horizontally (in HCS plane) or vertically (in VCS plane) upon satisfying the corner stitch merging criteria. Then the merged tile traverses all neighbors to merge all possible empty tiles, which are split during tile insertion. This process is referred to as shadow rectification in the corner stitch algorithm.

Algorithm 1: Layout Generation Workflow.

```
Read input geometry script
Create a root node with initial floorplan rectangle
for each symmetrical module do
    Create a child node of the root
Choose one child node and build the sub-tree
for each child node in the sub-tree do
    CreateCornerStitch(child node)
    CreateConstraintGraph(HCS/VCS plane)
    Perform bottom-up constraint propagation
Evaluate the root node using the longest path algorithm
Perform top-down location propagation
```

Algorithm 2: CS and CG Construction.

```
Function CreateCornerStitch(Node):
    foreach tile in Node do
        Perform corner stitch tile insertion operation
        H_neighbors=list of east/west neighbors
        V_neighbors=list of north/south neighbors
        foreach neighbor in H(V)_neighbors do
            if neighbor.type==tile.type then
                Perform corner stitch merge operation
            Perform shadow rectification for the new
             tile
    return
Function CreateConstraintGraph(CS plane):
    H_coordinates=list of all x-coordinates of the plane
    V_coordinates=list of all y-coordinates of the plane
    foreach index in the H(V)_coordinates do
        Create a vertex in HCG/VCG
    foreach tile in the plane do
        Find proper constraint type and value
        Create an edge in HCG/VCG
    foreach pair of vertices in HCG/VCG do
        if no edge exists then
            Add a missing edge
    return
```

C. Layout Generation

The engine considers all design constraints through constraint graphs. Two types of constraints have been considered in this article: technology constraints and high voltage-and-current dependent constraints. The technology constraints are minimum spacing, minimum width, minimum length, minimum extension, and minimum enclosure. An illustration of each type of constraint is illustrated in Fig. 5. Here, constraint value U is for maintaining relative location between components. These edges can not be found directly from corner stitch plane. For high-voltage and current applications, voltage-dependent spacing and current-dependent width are considered as reliability constraints. The reliability constraints application result is shown in [16]. An interactive constraint table is maintained so that the users can modify the minimum constraint values according to their necessity.

1) Hierarchical Constraint Graph Creation: Once both HCS and VCS planes are created, the corresponding horizontal constraint graph (HCG) and vertical constraint graph (VCG) are created by applying all design constraints. High-level sample pseudocode for constraint graph creation is shown in Algorithm 2. In this step, all X/Y coordinates from HCS/VCS plane are mapped to the HCG/VCG vertices. The associated design constraint is mapped as an edge in constraint graphs. Once all design constraints are mapped, a thorough check is performed on the constraint graph to make sure there is an edge in between every pair of vertices. If no edge is found in between any two vertices, an edge is created with a minimum database unit value to preserve the relative location. This edge is referred to as a missing edge in the algorithm. After creating both constraint graphs, a bottom-up constraint propagation algorithm [17] has been performed to propagate minimum design constraints from the child to the parent node. Once the second for loop in Algorithm 1 ends, the root node contains all the propagated minimum constraints. This node is then evaluated using the longest path algorithm, and a top–down location propagation algorithm [17] is applied to generate solutions. Constraint graph evaluation is described in the latter part of this section.

For each node in the tree, the foreground tiles are used to find minimum width and length constraints, whereas the background tiles provide the minimum enclosure and spacing constraints. The graph creation starts with leaves in the tree. This article refers ‘node’ to a corner stitch tree and ‘vertex’ to a constraint

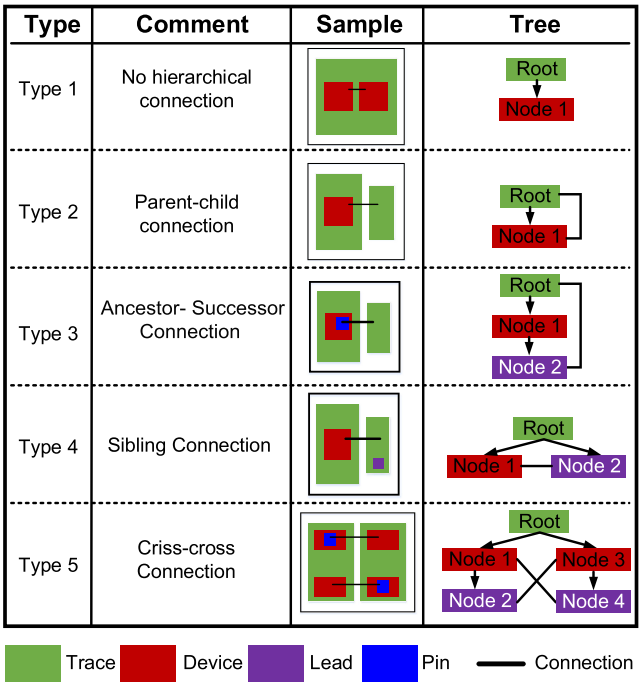


Fig. 6. Five types of rigid wire bond connections.

graph to distinguish between hierarchical trees and constraint graphs. Each graph is evaluated by the longest path algorithm. After the evaluation of each child node, the longest distance from the source vertex to sink vertex is propagated to the parent node graph as a propagated edge between the corresponding vertices. Each tile is mapped as two vertices with an edge in the constraint graph. For each corner-stitched plane, the lowest coordinate is considered as the source vertex, and the highest coordinate is considered as the sink vertex. In between, there may exist independent and dependent vertices. If there is a fixed dimension component (i.e., devices, leads) in the corner-stitched plane, that component width (length) is mapped with a fixed edge in the HCG (VCG). Any destination vertex of a fixed edge is a dependent vertex. The dependent vertex always maintains a fixed distance from the corresponding independent vertex. All other vertices and edges are independent and nonfixed.

2) Interconnection Handling: In this article, two types of wire bonds are considered: flexible and rigid. For flexible wire bonds, no alignment between the connected group is maintained. Therefore, each group can move independently without violating constraints, resulting in the excessive length of wire bonds. The rigid wire bonds are treated as point connections and have source and destination coordinates. Source and destination vertices either have the same Y or X coordinate for horizontal and vertical wire bonds, respectively. Depending on the hierarchy, five possible types of connections are considered. A set of horizontal rigid connections is shown in Fig. 6. Except for Type-1 connections, all other types of connection handling require vertices propagation from the child node to the parent node and can also introduce a backward edge in the constraint graph with both fixed edge and connection points. The connection processing algorithms are generic to all these types. A detailed illustration

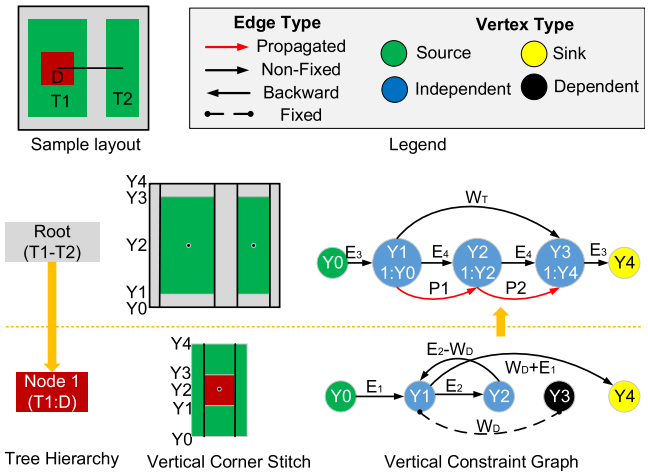


Fig. 7. Illustration of edges, vertices, and constraint propagation for a sample layout.

of minimum constraint propagation from the child to parent for a Type-2 connection is shown in Fig. 7. In the sample layout, a device (D) placed on top of a trace (T1) is connected with another trace (T2) by a horizontal rigid wire bond. The corresponding tree is shown on the bottom-left. In the child node (Node 1), the trace is the background tile and the device is the foreground tile with a wire bond terminal point. The parent node (Root) includes an outline of the layout with both traces and wire terminal points. The left terminal point of the wire has been propagated from the child node as the connection is only visible at the root node. Since the VCS of Node 1 has five Y coordinates, there are five vertices in the corresponding VCG. Here, E_1, E_2, and W_D stand for the minimum enclosure of the trace to the device, the minimum enclosure of the device to the wire bond, and the minimum length of the device, respectively. There is a nonfixed edge from Y2 to Y3 with a weight of E_2. Since Y3 is a dependent vertex, and Y1 is its reference point, the incoming edge from Y2 can be referred to Y1. The math expressions are shown in (2) and (3). By substituting Y3, (4) can be derived, which is represented as a backward edge in the VCG. This backward edge imposes a maximum constraint, whereas the forward edge imposes the minimum constraint. As a result, the edge from Y3 to Y4 is also redirected to the reference vertex Y1 with a weight of $W_D + E_1$. Both the dependent vertex and connection point vertex are handled with this algorithm.

$$Y3 - Y1 = W_D \tag{2}$$

$$Y3 - Y2 \geq E_2 \tag{3}$$

$$Y1 - Y2 \geq E_2 - W_D. \tag{4}$$

As Y0, Y2, and Y4 vertices are propagated from Node 1 to the root node, they are labeled in Root VCG with a prefix "1:". In the root node, the design constraints are E_3, E_4, W_T, which stands for minimum enclosure of the substrate to the trace, minimum enclosure of the trace to the wire bond, and the minimum length of the trace, respectively. Here, there are two propagated edges P1 and P2 from the child node (Node 1). P1 is the minimum distance from Y0 to Y2, and P2 is the minimum distance from

Y2 to Y4 in Node 1. Both P1 and P2 are equal to E_1+E_2. In this way, constraint propagation is performed throughout the tree to reserve a minimum space for child components in the parent node.

3) Hierarchical Constraint Graph Evaluation: The constraint graphs are evaluated using the longest path algorithm [25] in two phases. In the first phase, the minimum constraints are evaluated and propagated from the leaf to the root of the tree. The bottom-up constraint propagation algorithm is described in [17]. Once all minimum constraints are propagated to the root, the constraint graphs are ready for the second phase of evaluation, where the solutions are generated. Based on the user's choice, the following three types of solutions can be generated.

1) Minimum-sized solution: To generate the most-compact solution, the minimum constraint graphs are evaluated. First, the root node is evaluated, and all vertices minimum locations are determined. These locations are then propagated from top to bottom in the tree, and the associated algorithm is described in [17]. This solution reflects the theoretical maximum power density of the layout though it is not reliable due to the high temperature. This option is called "Mode 0" in PowerSynth.
2) Variable-sized solutions: If the designer has no prior requirement of floorplan size, this option (called "Mode 1") can help the user to choose a starting point. This mode can generate an arbitrary number of solutions. Once the number of solutions is selected, the layout generation algorithm [16] randomizes the edge weights of the root node constraint graphs. Then the locations are propagated towards leaf nodes. Without the upper limit of constraint values, this mode provides the largest solution space. By applying the optimization algorithm, a preferred floorplan size can be determined in this mode.
3) Fixed-sized solutions: This option is called "Mode 2." If the user has a pre-defined floorplan size, the layout generation algorithm [16] can also create an arbitrary number of solutions. Also, the user can use the optimum floorplan size found in Mode 1 as the fixed floorplan size. Due to the fixed upper bound, each vertex in the CGs has both minimum and maximum locations. Once the movable range is determined, the location of each vertex can be randomized to create new layouts.

Since all solutions will maintain the same relative locations as in the initial layout, all components in the same node are correlated. If multiple components in the same node share the same coordinate in the initial layout, the correlation remains throughout all the solutions. The current layout solutions are highly input geometry script dependent. For better results, the user needs to set up the script carefully to maintain a reasonable layout hierarchy.

IV. Models and Optimization Algorithms

A. Performance Evaluation Models

PowerSynth developers have been considering not only new models to include new objectives for optimization, but also they are working on improving existing models. The first version of the PowerSynth has already validated the first-order linear approximation technique through the Laplacian matrix model [15] with physical measurements. Since this methodology does not consider current density and mutual coupling among components, for some 2-D layouts with planar traces, it overestimates the loop inductance. Mutual coupling consideration is necessary, especially for high-density, multilayer layouts. A partial element equivalent circuit (PEEC)-based model [28] has been developed to address these limitations. This model constructs a controlled mesh structure and produces a detailed voltage-current distribution map. In addition, PowerSynth thermal model [15] can estimate the maximum junction temperature of the layout within a good accuracy. Besides, ARL ParaPower API has enabled PowerSynth users to use a 3-D thermal model and stress evaluation method. For the 2.5-D power module, optimization objectives include minimizing loop inductance and reducing the maximum temperature of the module. The electrical and thermal models used for power loop inductance and temperature evaluation are summarized in this section.

1) Electrical Model: The PEEC-based distributed electrical model with an adaptive mesh derived from the corner stitch data structure from the new layout engine is used for power loop inductance evaluation. A planar data structure with rectilinear objects is used to pass the layout information to the model. The mesh nodes are generated from the corner stitch data structure by filtering the intersection points from both HCS and VCS planes. Initially, all mesh nodes are distributed into two groups: internal and boundary, where the boundary nodes are nodes located on the connected trace group perimeter, and the rest of the nodes are treated as internal. The internal nodes always have four neighbors, as opposed to the boundary nodes. Each node is tagged with a unique index, and possible edges are created from the node to connect the surrounding neighbors. This edge stores the width and length information of each trace and may be either internal or boundary type. A good approximation of the skin effect is achieved as boundary edges are always smaller than internal edges. The response surface model from [15] is used to evaluate the self-inductance of every edge and while the mutual inductance between every pair of edges is evaluated using the equation in [31]. A hierarchical approach is implemented to handle the connections between device terminals and traces. This hierarchical representation reduces the total number of meshes, improves computational efficiency, and handles multilayer structures. Experimental verification of this model has been performed in [28]. This updated model can report distributed parasitic netlist, including *R*, *L*, *C*, and *M* components. This distributed netlist can be back-annotated to the circuit for validation. Since in a layout, the number of mutual inductance components (M) is very large, these values cannot be directly used in the time domain simulation. Therefore, a weighted distributed parasitic netlist based on the loop computation from the PEEC model can be used for transient simulation.

2) Thermal Model: Thermal performance is also one of the most important factors for MCPM layout optimization. In this article, the hardware-validated, fast, reduced-order thermal model [15] is used for 2.5-D power module optimization. The thermal model has already proven to accurately estimate

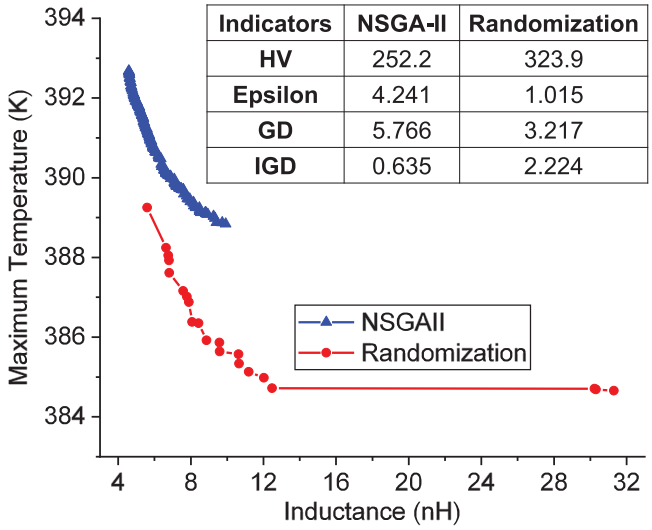
the steady-state maximum temperature of a power module. The model takes the layer stack material properties and dimensions, heat transfer coefficient, and steady-state power dissipation of each device as input. Then, a FEA simulation is performed using Gmsh [32] and Elmer [33] for each device. The temperature and flux values at the top metal layer of the stack are mapped to a regular grid, which is used to construct multiple rectangular contours for both temperature and heat flux magnitude. This format enables faster computation at contour intersections during the superposition calculation, since it approximates the thermal coupling effect among devices. The thermal resistance network is then extracted, and an impedance matrix is built using the 1-D heat transfer circuit topology. Later this matrix is used to quickly evaluate the steady-state thermal performance of each solution. This model has been proved to predict the result correctly with less than 10% error compared to the FEA simulations and approximately 10 000 times faster.

B. Optimization Algorithms

For optimizing a power module layout, the first version of PowerSynth and many other studies have used a genetic algorithm. Though a genetic algorithm like NSGA-II has been proven to be beneficial for multiobjective optimization in general, to have a legitimate tradeoff among the objectives in case of the power module, a properly tuned algorithm is required. Also, for heterogeneous power module layouts, the layout complexity and number of design variables grow exponentially, which makes it difficult due to dependency between variables. The following two optimization algorithms are considered in the latest version.

1) Genetic Algorithm (NSGA-II): This is a multiobjective optimization algorithm that generates new solutions as a next offspring by performing crossover, and mutation on the current population [30]. It takes a set of design variables and the number of generations as initial input and generates initial solutions as a starting point. These solutions are evaluated by the performance models and ranked based on the objectives. A nondominated sorting is performed on each generation to select the solutions on the Pareto-front. The procedure runs until it reaches the maximum number of generations. This algorithm can generally reach the Pareto-front within a few thousand solutions, thus yields a fast computational time.

2) Randomization: This is the built-in solution generation algorithm [16] for PowerSynth latest version. In this algorithm, the edge weights of the constraint graphs are randomized within the floorplan upper bound depending on the layout generation mode. This algorithm can generate an arbitrary number of solutions by varying edge values in the constraint graphs. This algorithm can explore a larger and more distributed solution space than that of the NSGA-II at the cost of longer run time. Since NSGA-II has better guidance towards optimization based on the objectives than the randomization, NSGA-II can find an optimized solution space quickly. For example, to achieve the Pareto-fronts (shown in Fig. 8), the runtime for NSGA-II and randomization are 2304 and 2698 s, respectively. To reach the Pareto-fronts, randomization and NSGA-II require 30 000 and 20 000 solutions, respectively. Therefore, it is evident that to achieve a comparable Pareto-front solution set, randomization requires a larger solution space as well as longer runtime than that of NSGA-II.



Indicators	NSGA-II	Randomization
HV	252.2	323.9
Epsilon	4.241	1.015
GD	5.766	3.217
IGD	0.635	2.224

Fig. 8. Pareto-front solution set comparison and tabular representation of the performance indicator values. Here, HV reference point is (33, 397).

Several well-established performance metrics [34], [35] are considered to compare the efficiency of the algorithms. These include hypervolume (HV), epsilon-indicator, generational distance (GD), inverted generational distance (IGD), etc. HV reflects the accuracy and diversity of the solutions and is measured against a reference point. The higher HV value represents a better solution space. Epsilon indicator measures the accuracy, diversity, and cardinality of the solution space. GD is an indicator of the accuracy and the inverted GD indicates both accuracy and diversity. These three indicators are preferred to be lower values for better-optimized solution space. A study has been performed for a quantitative comparison of the solution spaces from both of the algorithms. For a sample power module, a nondominated sorting is performed on 20 000, and 30 000 solutions for NSGA-II, and randomization, respectively. The Pareto-front solution space is shown in Fig. 8. A reference Pareto-front is generated from these two Pareto-solution sets, which is used to calculate the above-mentioned quantitative indicators and the result illustrated in a table (shown in Fig. 8).

From the table, it is evident that NSGA-II is better in terms of IGD, whereas randomization is better in case of other three indicators for the particular case. Based on our study, it can be said that no single generic optimization algorithm is enough for the best performance, and customized algorithms are in great need of expanding design capability without sacrificing flexibility and efficiency.

V. Example Designs Using PowerSynth

A. High Current, Complexity, and Density Layouts

The new layout engine is more flexible, efficient, and capable of handling modules with more complex geometry and higher power density compared to the old engine. To demonstrate these

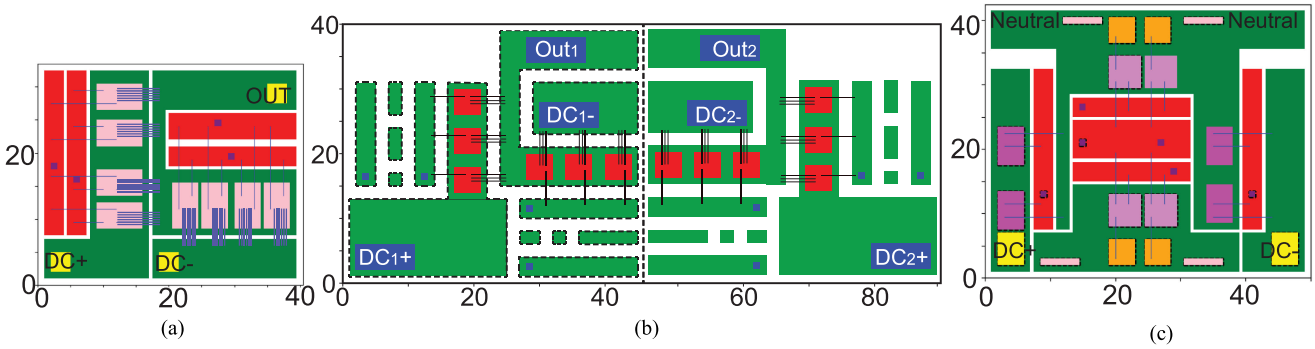


Fig. 9. Three minimum-sized solution layouts. (a) High-current rated half-bridge module (40.5 mm × 33.5 mm), (b) 2.5-D full-bridge module (94 mm × 40 mm), and (c) phase-leg module (50 mm × 42.5 mm).

capabilities three sample layouts are chosen, which are not possible to handle using the old layout engine. The minimum-sized solutions are shown in Fig. 9. Each case represents the maximum power density for individual initial input layout. A brief description about each solution is provided below.

Fig. 9(a) represents a high-current rated half-bridge power module layout with four devices (CPM3-1200-0016 A: 1200V-112 A) per switching position. The evaluated current rating for this module is 353 A at junction, and case temperature of 150°C, and 90°C, respectively. Since this module has a higher current rating (353 A versus 300 A) and smaller power loop inductance (7.12 nH versus 10.2 nH) compared to commercial modules rated with similar testing condition but larger footprint, this case shows the capability of producing high-density power module using PowerSynth.

In Fig. 9(b), a 2.5-D full-bridge module is shown, which has two symmetrical half-bridge modules. Each half-bridge module has three SiC MOSFETs per switching position. This case proves the benefit of hierarchical approach. Optimizing part-by-part has been proven to be computationally less (12 times) expensive compared to planar 2.5-D module optimization.

Finally, in the Fig. 9(c), there is a phase-leg module consisting of SiC MOSFETs (dark pink), IGBTs (light pink), and diodes (amber). This layout is an example of the geometrical complexity handling capability and also the hierarchical benefits as the optimization results can be reused to optimize a three phase inverter.

All these three modules initial layouts are taken as input using the corresponding geometry script and a set of standard minimum design rules is applied to generate the minimum-sized solutions. Each of the cases can be considered for further optimization with standard floorplan size as well. Depending on the cost, geometry complexity as well as fabrication complexity, a simpler 2.5-D full-bridge module design [shown in Fig. 4(a)] is selected for the optimization study and result validation.

B. Layout Optimization on the 2.5-D Module

This new version of PowerSynth can consider hierarchical optimization, which enables the tool to optimize part-by-part and reuse the optimization result. The old version cannot perform this divide-and-conquer strategy and hence requires longer

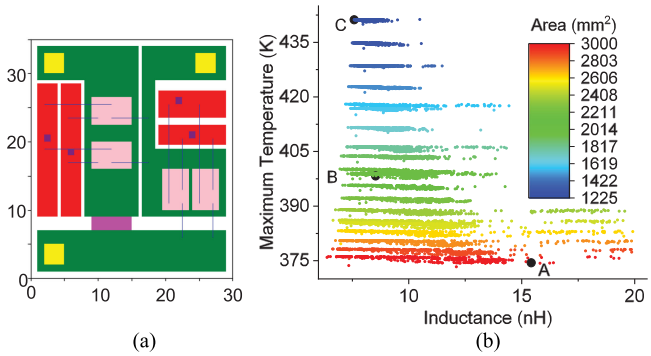


Fig. 10. (a) Minimum-sized solution (30 mm × 35 mm) and (b) complete solution space.

computation time than the current one. The hierarchical approach has enabled optimizing the 2.5-D module in two parts: First, optimize each symmetrical part, and second, perform an evaluation of the complete module. Therefore, the sample 2.5-D power module [shown in Fig. 4(a)] layout is chosen in such a way that the advantage of the hierarchical approach can be demonstrated and validated through physical measurements in both cost-and-time effective way. Since the 2.5-D full-bridge power module has two symmetrical half-bridge modules, optimization results from one half can be mapped to the other one. The hierarchical optimization approach allows the reuse of the optimization results and hence reduces the computational effort by half [17]. Therefore, to optimize the 2.5-D full-bridge power module shown in Fig. 4(a), the right-half module is taken as the input using the layout description script. A standard set of minimum design constraints are applied to generate the minimum-sized solution shown in Fig. 10(a). The solution layout is then evaluated using electrical and thermal models described earlier in this section. Since a decoupling capacitor is inserted into the power module, the power loop is considered from the capacitor (C1) top terminal to the capacitor (C1) bottom terminal, and the evaluated loop inductance is 7.986 nH. A layer stack with a baseplate and a DBC substrate is characterized using Gmsh [32] and Elmer [33] to evaluate the maximum temperature. The ambient temperature is set to 300 K, and the heat transfer coefficient is set to 150 W/m^2.K. The maximum temperature for this half-bridge is found 457.954 K. Since this is

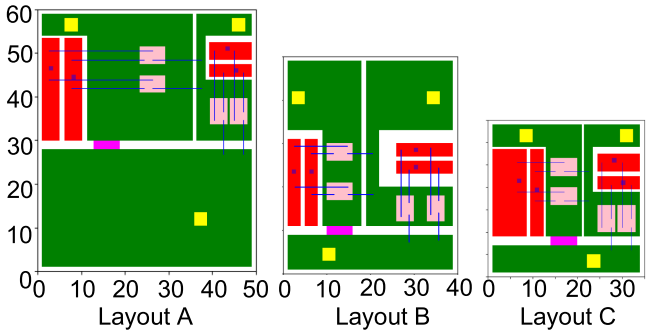


Fig. 11. Selected three layouts from Fig. 10(b).

TABLE II
PERFORMANCE TRADEOFF OF THREE SELECTED SOLUTIONS

Layout ID	Inductance (nH)	Max Temperature (K)	Size (mm×mm)
A	15.426	374.401	50×60
B	8.538	398.278	40×50
C	7.582	441.159	35×35

the minimum-sized solution, it reflects the theoretical maximum power density of the layout family. However, this design is not reliable in terms of thermal performance due to such high temperature, though the loop inductance is quite low.

In this article, the optimization target is to minimize power loop inductance and maximum temperature of the module by varying the placement and route of the components. To optimize the layout and come up with a balanced electro-thermal solution, about 10 000 solutions are generated with varying floorplan sizes. Since the minimum-sized solution is 30 mm × 35 mm, the floorplan sizes are chosen to be greater than the minimum size. Floorplan sizes are varied from 35 mm × 35 mm to 50 mm × 60 mm to explore a large solution space. For each floorplan size, around 500 solutions are generated and evaluated. The complete solution space is shown in Fig. 10 (b). Here, three solutions are labeled to show the tradeoff between the electrical and thermal objectives, and their corresponding layouts are shown in Fig. 11. Generally, the tool suggests a Pareto-front solution set and the user can choose from the solution set based on the tradeoff requirement. However, in this case, the complete solution space is shown for all evaluated design cases.

In Table II, the performance of each selected solution is provided. The table shows that Layout C has a lower inductance value with the highest temperature due to spacing constraints. On the other hand, Layout A has a better thermal result with a worse electrical, which is reasonable as it has the largest floorplan area. Among the solutions, Layout B holds a reasonable tradeoff between the two extreme choices. Therefore, out of all 10 000 solutions, Layout B with a floorplan size of 40 mm × 50 mm and balanced design objectives, is chosen for fabrication and testing. Though the solutions with larger floorplan sizes may provide both better thermal and electrical performance, those solutions are not chosen since the floorplan utilization is low, and the fabrication cost increases proportionally with the floorplan size. Also, there are few solutions with the same floorplan size but have lower inductance values. However, those solutions are not chosen due to some limitations. E.g., devices or traces are too close, wide signal traces, and long wire bonds, etc. These solutions can be unreliable and difficult to implement through the manual fabrication process carried out in a lab environment. Therefore, Layout B is chosen since it is easier to fabricate and also very close to the Pareto-optimal solution set.

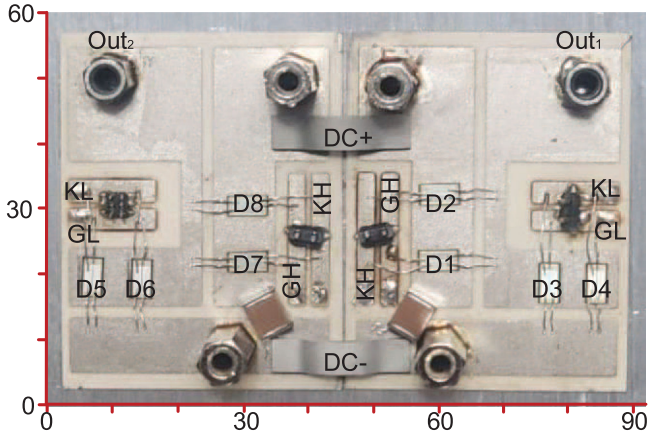


Fig. 12. Fabricated 2.5-D full-bridge power module (92 mm × 60 mm).

VI. EXPERIMENTAL VERIFICATION

A. Fabrication

The High-Density Electronics Center (HiDEC) of the University of Arkansas has complete processing, packaging, and assembly facilities for developing state-of-the-art power electronic module packages. To leverage the facilities, the selected power module has been fabricated and assembled in-house with a standard SiC-based MCPM fabrication flow. The selected layout information is exported into standard CAD files for manufacturing using the PowerSynth export feature. Then the DBC is prepared for attaching the devices and terminals through the chemical etching process. 1.2-kV SiC devices (CPM2-1200-0040B) from CREE are attached, and the wire bonding is performed by an automatic machine with 12-mil aluminum wires. Then, the terminals and the capacitor are attached to complete the half-bridge module fabrication. The room for the capacitor is not enough on the dc- trace since the octagon terminals in the fabricated module are different from the square terminals in the PowerSynth solution. Therefore, the capacitor needs to be placed diagonally. The substrate has been diced into two half-bridge modules to form a modular 2.5-D full-bridge module. Both of the modules dc+ and dc− traces are connected through metal bridges for intersubstrate connections. Finally, the base plate and heatsink are attached for heat dissipation. The fabricated power module is shown in Fig. 12.

The fabricated full-bridge module size is 92 mm × 60 mm and the current rating is 76 A. To determine the current rating of the full-bridge power module, a continuous test simulation is performed in LTSPICE using CREE provided SPICE model for CPM2-1200-0040B device. The full-bridge module has been simulated as an inverter at 800 V with a 3Ω resistive load and an 870 μH inductive load in series. In this simulation, PowerSynth extracted parasitics (shown in Fig. 17) has been incorporated.

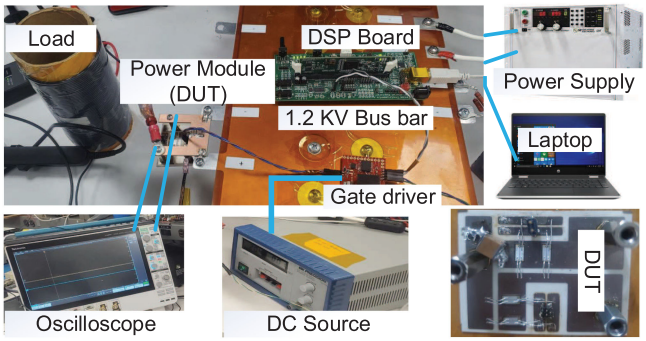


Fig. 13. Double pulse test setup.

The switching frequency is 50 kHz and the output current is 76.64 A with 1 kHz frequency. The calculated efficiency is 99.67%. The external gate resistance is 10 Ω. Since the devices have a maximum of 175 °C junction temperature tolerance, the junction temperature is set to 150 °C. Average switching loss is measured from the simulation. The Rdson value for each device is found 76 mΩ at 150 °C from the device datasheet, which is used to calculate conduction loss. The total power loss values have been used in ANSYS static thermal simulation to verify the junction temperature. We found a maximum junction temperature of 158.42 °C (5.6% mismatch with the assumption) with a case temperature of 90 °C. Two 12-mils aluminum bonding wires are used in each device, which limits the current through each device to 40 A. So, for each leg, the maximum of 80 A current is allowed as there are two devices in each position. The half-bridge module dimensions are comparable with commercial modules from CREE [36]. A good number of larger modules (e.g., 105 mm × 62 mm, 80 mm × 53 mm, 106 mm × 62 mm, 108 mm × 45 mm, etc.) than the optimized solution are available in the market. Since fabricated module (shown in Fig. 12) size is smaller, the current rating is also much smaller compared to the commercial modules. However, the full-bridge module is a value added by not requiring two separate half-bridge modules.

B. Double Pulse Testing

To verify the PowerSynth electrical model, a double pulse test is performed with the fabricated power module. The test schematic and experiment setup is shown in Figs. 13 and 14(a) , respectively. An FR4 small bus has been used to connect the module (DUT) terminals with the 1.2-kV laminated bus bar terminals. Due to the custom interface, a jumper wire has been used to connect the gate driver with the gate and Kelvin source terminals of the DUT. The load (shown in Fig. 13) has an inductance of 131 μH. As the fabricated module has no encapsulating gel, both of the modules are tested under 400 V/15 A rating for safety consideration. The Ids and Vds waveforms are shown in Fig. 14 (b), and (c), respectively.

C. Thermal Measurement

To verify the optimization result, a maximum temperature measurement experiment is performed in an access-controlled

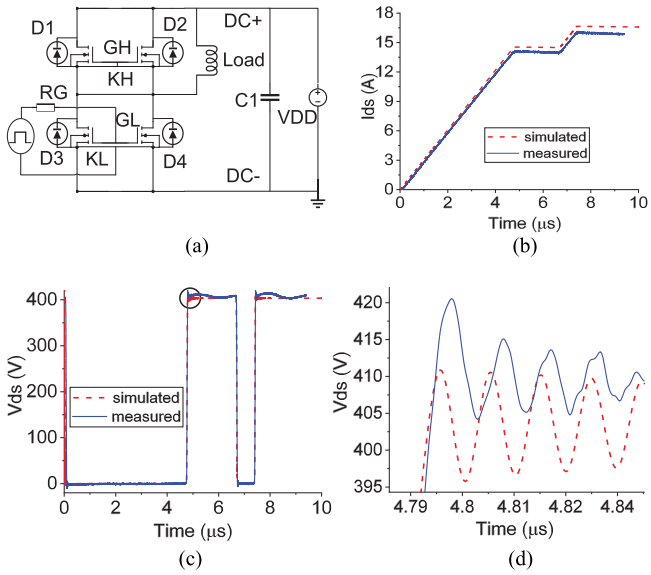


Fig. 14. (a) DPT schematic and measurement results for (b) Ids, (c) Vds, and (d) zoom-in shot of (c).

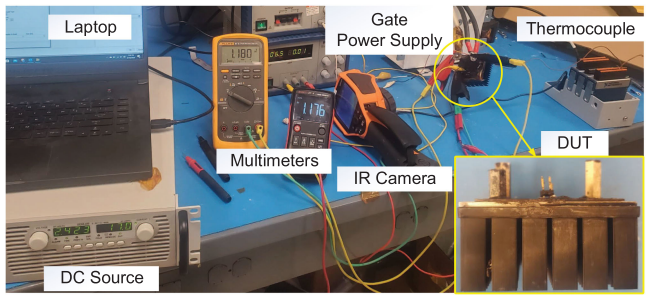


Fig. 15. Thermal measurement experimental setup.

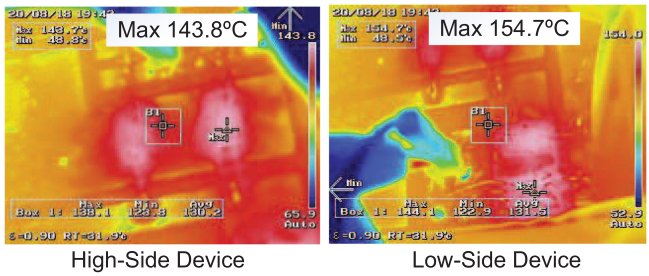


Fig. 16. Measurement results.

lab environment. The KEYSIGHT U5855 A TrueIR Thermal Imager has been used to measure the temperature of the devices and heat sink. A J-type thermocouple from National Instruments is used to measure ambient temperature. While fabricating the module, an aluminum baseplate is used as copper baseplate was not available. So, silver epoxy is used to attach the aluminum baseplate (50 mm × 60 mm) with the substrate. A 60 mm × 60 mm commercial heat sink (shown in Fig. 15) is attached with the baseplate using thermal grease material (thermal conductivity 4.5 W/m-K) to achieve a close match with the assumed heat transfer coefficient in the optimization

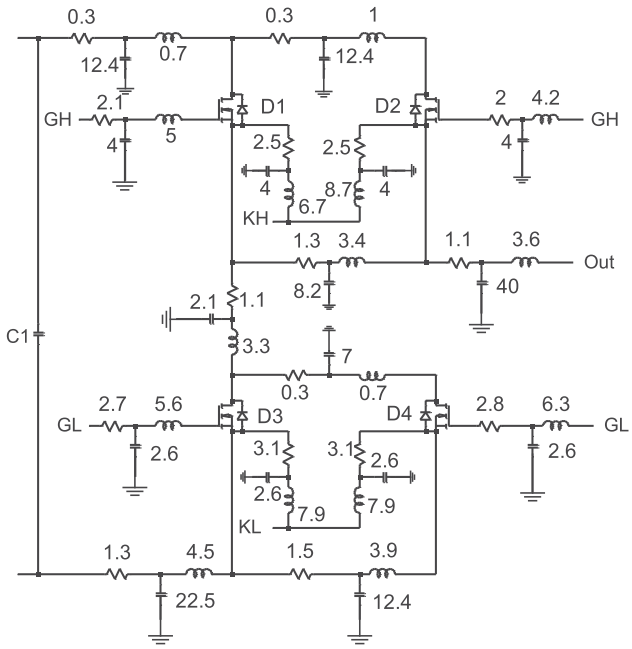


Fig. 17. Schematic of the extracted parasitic netlist. *R*, *L*, and *C* are in mΩ, nH, and pF, respectively.

case. However, based on the methodology from [15], the calculated equivalent heat transfer coefficient of natural convection is approximately 123 W/m^2-K. Here, the heat sink surface area is 4.83E-02 m^2 and temperature is 407 K. The measurement setup is shown in Fig. 15. A high-current dc power supply is used to supply the necessary current to the devices so that the power dissipation for each parallely connected devices pair is approximately 20 W as we assumed 10 W heat dissipation for each die while performing the optimization. Two multimeters are used to measure the voltage drop across each pair of parallel-connected devices. From electrical measurements, the supply current is found 17 A to achieve a total power dissipation of 40.052 W for the module. In the measurement case, exactly same assumed boundary conditions as in the optimization case cannot be achieved due to several obvious factors like defect in insulation, heat transfer coefficient mismatch, manufacturing defects, material mismatch of the baseplate, etc.

D. Validation Results

1) Electrical Performance: PowerSynth extracted distributed netlist for the Layout B (shown in Fig. 11), has been used for time-domain simulations in LTSPICE. The schematic of the extracted netlist is shown in Fig. 17. Here, the capacitance value of C1 is 1.27 μF. The parasitic values are extracted at 100 MHz, and a gate pulse with -5 to $+20$ V is applied with a gate ON and OFF resistance of 15 and 5Ω, respectively. Since only low-side devices are switched, the high-side switches are turned OFF. A 400 V dc voltage is applied to the circuit, while Vds and Ids are measured for low-side switches. The resultant plots are shown in Fig. 14. From the Vds and Ids comparison result, it is clear that the simulation has a close match with the measurement. In the case of Ids, the difference is around 3.71% with 16.49 A in simulation versus 15.90 A in measurement. The measured Vds has a higher overshoot of about 10 V and higher noises than the simulated one due to the additional jumper wire inductance and noises from the gate signal in the measurement setup. In the case of measurement, the turn-OFF ringing frequency is found 89.43 MHz, while the simulation shows 86 MHz as illustrated in Fig. 14(d). The extracted loop inductance from PowerSynth is within 3.84% accuracy.

TABLE III
COMPARISON BETWEEN THERMAL MEASUREMENT, ANSYS, AND POWERSYNTH

Case	Maximum Temperature (K)				Error in Temp. Rise
	D1	D2	D3	D4	
Measurement	416.8	416.5	427	427.7	-
ANSYS	418.9	416.8	422.4	422.9	-3.57%
PowerSynth	418	417.9	418.3	418.5	-7.15%

2) Thermal Performance: To have a fair comparison between the measurement result and PowerSynth predicted result, the thermal performance is re-evaluated for the same layout in both ANSYS Workbench and PowerSynth by introducing all possible changes in the boundary conditions from the measurement. The thermal camera images are shown in Fig. 16. The result comparison is shown in Table III. In simulations, for each SiC device, the heat dissipation is set to 10 W (2.31 W/mm^3). The heat transfer coefficient of the baseplate backside is 123 W/m^2-K, and the ambient temperature is 299 K.

From the results in Table III, it is evident that the maximum temperature rise difference of PowerSynth compared to the measurement and ANSYS result is within 10%, and 5%, respectively. There are a few factors like measurement equipment (power supply, multimeters, thermocouple, *IR* camera) tolerances, defects in the structure itself, heat dissipation through radiation are present in the measurement case, which cannot be considered in both ANSYS and PowerSynth cases. The IR camera tolerance is within $+2$ °C and -2 °C. The thermocouple reading accuracy is within $\pm 0.1\%$. The natural convection air cooling is applied and heat sink temperature is measured using the IR camera. The total power dissipation is not completely across the structure and a part of the heat is dissipated through radiation. Therefore, the equivalent heat transfer coefficient value estimation is not 100% accurate. As the baseplate attachment process was manual, some voids are found between the substrate and the baseplate from the scanning acoustic microscope images. All these unavoidable factors are causing the discrepancy between the measurement and simulation result.

VII. CONCLUSION

PowerSynth core architecture has been updated from the planar approach to a modular, and hierarchical one to optimize more complex, and high density power modules. The hierarchical corner stitch with constraint propagation has outperformed the traditional matrix-based layout generation methodology in efficiency and success rate. The updated constraint-aware layout engine is more efficient, scalable, and generic compared with the previous approach. This layout engine is demonstrated with the capability of processing complex geometry with

heterogeneous components, exploring a broader solution space, and adapting to different optimization algorithms. Updated electrical and thermal models with hierarchical layout description script enable PowerSynth to optimize not only 2-D but also 2.5-D power module layouts efficiently. This hierarchical approach reveals a promising path to extend PowerSynth to 3-D power modules as well. A full-bridge 2.5-D power module layout is optimized using the latest PowerSynth v1.9, and a well-balanced solution is fabricated and tested. The measurement and FEM simulation show a close agreement with PowerSynth-predicted electrical and thermal results. A completed round-trip PowerSynth-assisted design flow is demonstrated to achieve better productivity over the traditional manual design approach. In this iteration, thermal and electrical mutual coupling within each symmetrical part are considered and the coupling among different symmetrical parts will be considered in future. Since the tool is still a subject of on-going research, there are limitations and scope for updating features to produce better solutions. Two worth mentioning limitations are: 1) dependency on the initial layout from the user and maintaining the relative position of the components throughout the solutions; 2) inability to handle non-Manhattan routing. Along with addressing these limitations, next step will be demonstrating 3-D MCPM layout optimization using PowerSynth updated CAD-flow. Existing models will be updated, and validated for 3-D power module cases. Also, the architecture will be upgraded to support parallel computing, and machine-learning models. A new layout description language with object-based layout representation will also be introduced in the new version. Incorporating these new features, PowerSynth v2.0 will be released with the necessary documentation, and test cases for the users.

ACKNOWLEDGMENT

The authors would like to thank A. Wallace, D. G. Castillo, H. Chen for their help throughout the fabrication, Md. H. Mahmud, Y. Wu, F. Diao, Dr. Y. Zhao, A. Imran Emon, Dr. F. Luo, Md M. Hossain, A. Ur Rashid for their help throughout the electrical testing, and A. Iradukunda, B. Md. Nafis, M. Montazeri, and Dr. D. Huitink for their help with CAD simulation and thermal measurements.

REFERENCES

[1] X. Zhao *et al.*, "Performance optimization of a 1.2 kV SiC high density half bridge power module in 3D package," in *Proc. IEEE Appl. Power Electron. Conf. Expo.*, Mar. 2018, pp. 1266–1271.

[2] C. Chen *et al.*, "Flexible PCB-Based 3-D integrated SiC half-bridge power module with three-sided cooling using ultralow inductive hybrid packaging structure," *IEEE Trans. Power Electron.*, vol. 34, no. 6, pp. 5579–5593, Jun. 2019.

[3] H. A. Mantooth and S. S. Ang, "Packaging architectures for silicon carbide power electronic modules," in *Proc. Int. Power Electron. Conf.*, May. 2018, pp. 153–156.

[4] F. Yang, Z. Wang, Z. Liang, and F. Wang, "Electrical performance advancement in SiC power module package design with Kelvin drain connection and low parasitic inductance," *IEEE J. Emerg. Sel. Top. Power Electron.*, vol. 7, no. 1, pp. 84–98, Mar. 2019.

[5] S. Mandray *et al.*, "Automatic layout optimization of a double sided power module regarding thermal and EMC constraints," in *Proc. IEEE Energy Convers. Congr. Expo.*, Sep. 2009, pp. 1046–1051.

[6] P. Ning, F. Wang, and K. D. T. Ngo, "Automatic layout design for power module," *IEEE Trans. Power Electron.*, vol. 28, no. 1, pp. 481–487, Jan. 2013.

[7] H. Li, S. Munk-Nielsen, S. Bczkowski, and X. Wang, "A novel DBC layout for current imbalance mitigation in SiC MOSFET multichip power modules," *IEEE Trans. Power Electron.*, vol. 31, no. 12, pp. 8042–8045, Dec. 2016.

[8] M. Chinthavali *et al.*, "50-kW 1 kV DC bus air-cooled inverter with 1.7 kV SiC MOSFETs and 3D-Printed novel power module packaging structure for grid applications," in *Proc. IEEE Appl. Power Electron. Conf. Expo.*, Mar. 2018, pp. 133–140.

[9] A. Dutta and S. S. Ang, "A module-level spring-interconnected stack power module," *IEEE Trans. Compon., Packag. Manuf. Technol.*, vol. 9, no. 1, pp. 88–95, Jan. 2019.

[10] S. Huang, Z. Xu, F. Yu, and S. S. Ang, "Impact of nano-diamond composites on low-temperature co-fired ceramic interposer for wide bandgap power electronic module packages," in *Proc. IEEE Workshop Wide Bandgap Power Devices Appl.*, Nov. 2016, pp. 314–318.

[11] C. Chen, F. Luo, and Y. Kang, "A review of SiC power module packaging: Layout, material system and integration," *CPSS Trans. Power Electron. Appl.*, vol. 2, no. 3, pp. 170–186, Sep. 2017.

[12] C. Martin, J. M. Guichon, J. L. Schanen, and R. Pasterczyk, "Gate circuit layout optimization of power module regarding transient current imbalance," *IEEE Trans. Power Electron.*, vol. 21, no. 5, pp. 1176–1184, Sep. 2006.

[13] Y. Peng *et al.*, "PowerSynth progression on layout optimization for reliability and signal integrity," *IEICE Nonlinear Theory Appl.*, vol. 11, no. 2, pp. 124–144, Apr. 2020.

[14] K. Hermanns, Y. Peng, and H. A. Mantooth, "The increasing role of design automation in power electronics: Gathering what is needed," *IEEE Power Electron. Mag.*, vol. 7, no. 1, pp. 46–50, Mar. 2020.

[15] T. M. Evans *et al.*, "PowerSynth: A power module layout generation tool," *IEEE Trans. Power Electron.*, vol. 34, no. 6, pp. 5063–5078, Jun. 2019.

[16] I. Al Razi, Q. Le, H. A. Mantooth, and Y. Peng, "Constraint-aware algorithms for heterogeneous power module layout synthesis and optimization in PowerSynth," in *Proc. IEEE Workshop Wide Bandgap Power Devices Appl.*, Oct. 2018, pp. 323–330.

[17] I. Al Razi, Q. Le, H. A. Mantooth, and Y. Peng, "Hierarchical layout synthesis and design automation for 2.5D heterogeneous multi-chip power modules," in *IEEE Energy Convers. Congr. Expo.*, Sep. 2019, pp. 2257–2263.

[18] I. Laird, X. Yuan, J. Scoltock, and A. J. Forsyth, "A design optimization tool for maximizing the power density of 3-Phase DC-AC converters using silicon carbide (SiC) devices," *IEEE Trans. Power Electron.*, vol. 33, no. 4, pp. 2913–2932, Apr. 2018.

[19] D. Zhang, T. Fan, P. Ning, and X. Wen, "An automatic EMI filter design methodology for electric vehicle application," in *Proc. IEEE Energy Convers. Congr. Expo*, Oct. 2017, pp. 4497–4503.

[20] M. Hammadi *et al.*, "Layout optimization of power modules using a sequentially coupled approach," *Int. J. Simul. Model.*, vol. 10, pp. 122–132, Sep. 2011.

[21] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "VLSI module placement based on rectangle-packing by the sequence-pair," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 15, no. 12, pp. 1518-1524, Dec. 1996.

[22] H. Cao, P. Ning, X. Wen, and T. Yuan, "A genetic algorithm based motor controller system automatic layout method," in *Proc. Int. Conf. Power Electron. IEEE Energy Convers. Congr. Expo Asia*, May. 2019, pp. 1499–1504.

[23] N. Hingora *et al.*, "Power-CAD: A novel methodology for design, analysis and optimization of power electronic module layouts," in *Proc. IEEE Energy Convers. Congr. Expo.*, Sep. 2010, pp. 2692–2699.

[24] J. K. Ousterhout, "Corner stitching: A data-structuring technique for VLSI layout tools," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 3, no. 1, pp. 87–100, Jan. 1984.

[25] Y. Z. Liao and C. K. Wong, "An algorithm to compact a VLSI symbolic layout with mixed constraints," in *Proc. ACM Des. Autom. Conf.*, Jun. 1983, pp. 107–112.

[26] "PowerSynth releases," Accessed: Jan. 21, 2021. [Online]. Available: https://e3da.csce.uark.edu/release/PowerSynth/

[27] T. Evans *et al.*, "Development of EDA techniques for power module EMI modeling and layout optimization," in *Proc. IMAPS Int. Symp. Microelectron.*, Oct. 2019, pp. 193–198.

[28] Q. Le, T. Evans, Y. Peng, and H. A. Mantooth, "PEEC method and hierarchical approach towards 3D multichip power module (MCPM) layout optimization," in *Proc. IEEE Int. Workshop Integr. Power Packag.*, Apr. 2019, pp. 131–136.
[29] "ARL ParaPower," Accessed: Jan. 21, 2021. [Online]. Available: https://github.com/USArmyResearchLab/ParaPower
[30] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[31] C. Hoer and C. Love, "Exact inductance equations for rectangular conductors with applications to more complicated geometries," *J. Res. Nat. Bur. Standards-C. Eng. Instrum.*, vol. 69C, pp. 127–137, Jan. 1965.
[32] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities," *Int. J. Numer. Methods Eng.*, vol. 79, pp. 1309–1331, Sep. 2009.
[33] M. Malinen and P. Råback, *Elmer Finite Element Solver for Multiphysics and Multiscale Problems*. Jlich, Germany: Forschungszentrum Juelich, Sep. 2013, pp. 101–113.
[34] N. Riquelme, C. Von Lücken, and B. Baran, "Performance metrics in multi-objective optimization," in *Proc. Latin Amer. Comput. Conf.*, Oct. 2015, pp. 1–11.
[35] E. Zitzler *et al.*, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.
[36] "CREE modules." Accessed: Jan. 21, 2021. [Online]. Available: https://www.wolfspeed.com/power/products/sic-power-modules

Imam Al Razi (Graduate Student Member, IEEE) received the B.Sc. in electrical and electronic engineering from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 2016. He is currently working toward the Ph.D. degree with the Computer Science and Computer Engineering Department, University of Arkansas, Fayetteville, AR, USA.

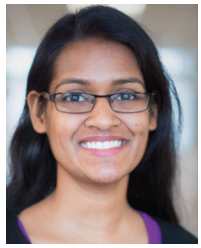
He is working as a National Science Foundation's Engineering Research Center for Power Optimization of Electro-Thermal Systems (POETS) Student on automated layout synthesis and optimization tool for multichip power modules (MCPMs). His research interests include computer-aided design (CAD) tool development for power electronics and VLSI.

Quang Le (Student Member, IEEE) received the B.S. (*summa cum laude*) degree in electrical engineering in 2015 from the University of Arkansas, Fayetteville, AR, USA, where he is currently working toward the Ph.D. degree in electrical engineering.

He is currently a Research Assistant at the University of Arkansas. His current research focuses on modeling methods for electrical parasitic extraction for power electronics modules.

Tristan M. Evans (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering in 2009 and 2011, respectively, from the University of Arkansas, Fayetteville, AR, USA, where he is currently working toward the Ph.D. degree in electrical engineering.

He then joined the R&D department of ROHM Semiconductor at their headquarters in Kyoto, Japan, as an Assistant Researcher focusing on new die attach technology for SiC power devices and thermal modeling of SiC power modules. In 2016, he was awarded the Distinguished Doctoral Fellowship from the University of Arkansas. He is a Research Assistant with the University of Arkansas. His current research interests include developing design automation tools for power electronics modules. His current research focuses on developing design automation tools for power electronics modules.



Shilpi Mukherjee (Member, IEEE) received the B.Eng. degree in electrical and electronics from the National Institute of Engineering, Mysore, India, in 2010, and the M.Sc. degree in microelectronics and photonics from the University of Arkansas, Fayetteville, AR, USA, in 2014, working on life cycle assessment of solar cells with Dr. Ajay Malshe. She is currently working toward the Ph.D. degree with the University of Arkansas, Fayetteville, AR, USA with Dr. Alan Mantooth on developing design rules to mitigate partial discharge in power modules for an Army Research Lab project.

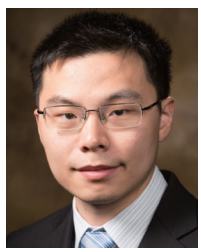
She was with the Tata Consultancy Services Ltd., an information technology firm, in Pune, India from 2010 to 2012. She completed an internship at ANSYS, Inc., in 2017. She has served as a student leader in National Science Foundation's multiuniversity multidisciplinary Engineering Research Center for Power Optimization of Electro-Thermal Systems for many years.

Ms. Shilpi was awarded the Doctoral Academy Fellowship by the University of Arkansas, in 2015.

H. Alan Mantooth (Fellow, IEEE) received the B.S. and M.S. degrees in electrical engineering from the University of Arkansas, Fayetteville, AR, USA, in 1985 and 1986, respectively, and the Ph.D. degree from the Georgia Institute of Technology, Atlanta, GA, USA, in 1990.

He then joined Analogy, a startup company in Oregon, where he focused on semiconductor device modeling and the research and development of modeling tools and techniques. In 1998, he joined the faculty of the Department of Electrical Engineering, University of Arkansas, where he currently holds the rank of Distinguished Professor. His research interests include analog and mixed-signal IC design and CAD, semiconductor device modeling, power electronics, and power electronic packaging. He helped establish the National Center for Reliable Electric Power Transmission (NCREPT) at the UA in 2005. He serves as the Executive Director for NCREPT as well as two of its centers of excellence: the NSF Industry/University Cooperative Research Center on GRid-connected Advanced Power Electronic Systems (GRAPES) and the Cybersecurity Center on Secure, Evolvable Energy Delivery Systems (SEEDS) funded by the U.S. Department of Energy. In 2015, he also helped to establish the University of Arkansas' first NSF Engineering Research Center entitled Power Optimization for Electro-Thermal Systems (POETS) that will focus on high power density systems for transportation applications.

Dr. Mantooth holds the 21st Century Research Leadership Chair in Engineering. He is currently serving the profession as Immediate Past-President of the IEEE Power Electronics Society until 2021. He is a member of Tau Beta Pi and Eta Kappa Nu, and registered Professional Engineer in Arkansas.



Yarui Peng (Member, IEEE) received the B.S. degree from Tsinghua University, Beijing, China, in 2012, the M.S. and Ph.D. degrees from the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA, in 2014 and 2016, respectively.

He is currently an Assistant Professor in Computer Science and Computer Engineering Department with the University of Arkansas, Fayetteville, AR, USA. He studies design methodologies and optimization algorithms for parasitic extraction, signal integrity, power integrity, and thermal reliability. He also develops design automation tools for power electronics to improve performance, reliability, and productivity. His research interests include computer-aided design, analysis, and optimization for emerging technologies and multichip packages, such as 2.5-D/3-D ICs and wide band-gap power electronics.

Dr. Peng was the recipient of the Best Paper Awards in SRC TECHCON 14, ICPT 16, and EDAPS 17.