Hierarchical Layout Synthesis and Design Automation for 2.5D Heterogeneous Multi-Chip Power Modules

Imam Al Razi [a], Quang Le [b], H. Alan Mantooth [b], Yarui Peng [a]
[a] Computer Science and Computer Engineering Department,[b] Electrical Engineering Department
University of Arkansas, Fayetteville, AR, US
ialrazi@email.uark.edu, qmle@email.uark.edu, mantooth@uark.edu, yrpeng@uark.edu

***Abstract*—Multi-chip power module (MCPM) layout design automation has been identified as one of the primary research interests in the power electronics community with the advent of wide bandgap circuits. MCPM physical design requires a time-consuming iterative procedure that is so far explored manually based on the experience of the designers. Though the number of components and routing layers is limited in power electronics, careful physical design is required because of thermal and reliability issues. In this paper, the benefits of a hierarchical design methodology are demonstrated over the state-of-the-art approaches. We propose a generic, scalable, and efficient algorithm to automate not only 2D but also 2.5D (multiple substrates in a planar package) and 3D (multiple device layers stacked on the same substrate) heterogeneous MCPM layouts considering hierarchy. A complete optimization approach for a full-bridge 2.5D power module is demonstrated using hardware-validated electrical and thermal models.**

***Keywords*—PowerSynth, corner stitch, constraint graph, Multi-Chip Power Module, algorithms, hierarchy, 2.5D power module, layout optimization**

I. Introduction

Nowadays design automation and optimization is attracting attention in the field of power electronics [1–6]. Power module physical design is one of the most critical steps to achieve the maximum performance for wide bandgap technologies (e.g., GaN and SiC). Power module, a key part of power electronic systems like power converters, has become one of the most interesting areas for design automation. Currently in the industry, power modules are designed manually through an iterative process that requires a lot of expertise and tedious work [7, 8]. Due to the limitation of human efforts, this process does not always guarantee an optimized solution. Therefore, several studies in power module design automation have been performed to extend VLSI placement-and-routing (P&R) concepts into power module layout synthesis [1, 3, 9]. However, there are some fundamental distinctions in VLSI and power module layout design considerations [10]. For example, the optimum VLSI layout is always the most compact one as the power supply is constant throughout the layout. However, in the case of power module layouts, the optimum design requires a trade-off among several targets (i.e., loop inductance, temperature, mutual coupling effects, etc.). Moreover, due to the high voltage, reliability, partial discharge, and thermal issues in power module layouts, a compact layout is not always a good design. Therefore, physical design algorithms for the power modules should have both compaction and expansion capabilities without violating design constraints. Inspired by VLSI CAD, generalized optimization algorithms are proposed in this work for hierarchical MCPM design automation.

In [3], the sequence pair method is used for representing and optimizing placement of components in a power module layout. For routing optimization, a 1D binary string is used for randomization that requires iterative electrical connectivity checking. Power modules are represented in a simplified manner to reduce computational effort. This simplification leads to reduced flexibility and limited optimization results. Also, this approach is for planar modules only and no straight forward path is mentioned for optimizing complex geometry and 2.5D/3D power modules.

PowerSynth [1] is a Multi-Chip Power Module (MCPM) layout synthesis tool that exploits multi-objective optimizations and generates optimized layout solutions. In this tool, reduced order electrical and thermal models [1] are used to evaluate each layout solution with higher speed with acceptable accuracy. Power modules are represented in a symbolic layout form consisting of lines and points. In physical layout, traces are 2D tiles whereas in symbolic layout, they are represented by 1D lines. Moreover, the devices are annotated as points, which is not appropriate as they also have rectangular shapes. Therefore, symbolic layout representation is not suitable for complex geometry and even some simple geometries, which are not possible to represent with symbolic layouts. The matrix-based methodology for layout solution generation has some significant limitations [9] like design constraint violation, limited solution space, incapability of varying space between components, etc.

Although the above-mentioned approaches handle most planar 2D power modules, only limited types of components have been investigated. To overcome the limitations with

This material is based on work supported by The National Science Foundation under Grant No. EEC-1449548 and Army Research Lab Contract No. 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 and Army Research Lab.

the studies, a scalable, generic, and efficient constraint-aware methodology based on corner stitching data structure [11] with a constraint graph [12] has been proposed in [9]. However, due to the use of a planar constraint graph, the components are correlated with each other and results in in-feasible solutions. All components in the layout always maintain a global relative location to each other. Therefore, there is less variation among the solutions and the solution space reduces. Also, during randomization, the dimensions of all components are randomized. So, in the solutions, all components sizes become variable although some components (i.e., MOSFETs, diodes, leads) should have fixed sizes.

Because of the planar approach [9], some obvious advantages of symmetrical power module layout optimization cannot be leveraged. Symmetrical power module design is crucial for current and thermal balancing, which affect the reliability and performance of the power module design. It has been shown that the asymmetric internal layout of an IGBT module can lead to unbalanced thermal loading given a mission profile, which in turn leads to mismatched lifetime among the IGBT devices [13]. Symmetrical power module layouts have improved the current sharing and reduced heat dissipation mismatch among parallel Si-IGBT devices [14]. To handle symmetrical layouts, the planar approach requires the complete layout as the initial input and evaluation of the layout is computationally expensive. However, large layouts can be divided into symmetrical parts to reduce computational efforts.

In this paper, we propose a hierarchical layout synthesis and optimization approach to prove the benefits of the hierarchical consideration over the existing planar consideration in the physical design automation process. We develop a design constraint-aware physical design algorithm to optimize 2.5D MCPM layouts with fixed dimension components using a hierarchical approach. In this work, power modules are optimized to minimize loop inductance and device temperature. Hardware validated electrical and thermal models have been used for performance evaluation of each solution. The corner stitching data structure and the constraint graph evaluation methodology has been extended to consider hierarchy. The proposed approach allows modeling and optimizing individual blocks that result in better optimization of the entire system efficiently. Hierarchical consideration generates more feasible layouts and reduces computational effort for the symmetric 2.5D power module layout optimization.

II. Physical Design Methodology

The hierarchical physical design automation of a 2.5D power module has four steps. First, the full-bridge symmetric layout needs to be broken into two parts and only one part needs to be considered. Second, a corner stitched tree structure and hierarchical constraint graphs are generated. Then, these constraint graphs are manipulated to generate layout solutions. Finally, these solutions are evaluated using a cost function to optimize the objectives. The trade-offs among multiple-objectives can be represented as a Pareto-front to the user.

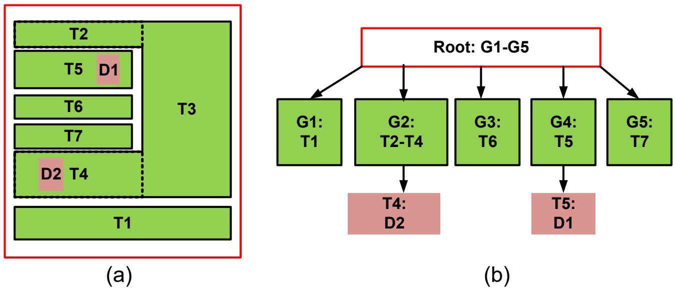

Fig. 1. (a) Initial layout of a power module and (b) its hierarchy structure

A. Corner Stitching with Hierarchy

The corner stitching data structure was originally used in VLSI [15] for layout representation. As this data structure is a planar one, the basic version of the data structure allows only non-overlapping solid tiles. However, in the case of power module layout representation, overlapping of components (i.e., trace and die, trace and lead) is required. So, necessary modifications are performed on the basic algorithms to consider overlapping of tiles. To maintain the hierarchy of components in the layout, a tree structure is developed with each node representing a corner-stitched plane. The root is the initial empty tile (substrate of the power module), where all traces are inserted. The entire layout is considered as a combination of some groups that are defined in the input script. When a component is placed into a group, the component is treated as the background and the hosting group foreground. The background tile is modified with corner stitch tile insertion operations (splitting, merging, etc.). The read-only outline of the background can be found from the parent node. To illustrate this, an initial power module layout and the corresponding tree structure is shown in Fig. 1(a) and (b), respectively. In Fig. 1(a), T and D represent traces and devices, respectively. In Fig. 1(b), all traces are in the root node forming different groups based on the input. Since T2, T3, and T4 are connected, they form a single group. As D2 is placed on top, it becomes the parent of D2. Similarly, D1 is the child of T5. Because D2 has T4 as the background, modified T4 and D2 are in the same node. The same process applies to T5 and D1 as well. In this data structure, for the same layout, two trees are stored: one for the horizontal corner stitched planes and another for the vertical corner stitched planes.

B. Hierarchical Constraint Graph Creation

Constraint graphs are widely used in VLSI floorplan compaction problems and those are planar constraint graphs. In this work, we propose a hierarchical constraint graph creation and evaluation algorithm. From each corner-stitched plane, the constraint graph is created using design constraints. For each node in the tree, a constraint graph is derived by a one-to-one mapping of the design constraints from the corner-stitched plane of that node. To maintain the horizontal and vertical correlations among components, the horizontal constraint graph (HCG) and vertical constraint graph (VCG) are created respectively. Illustration of the hierarchical constraint

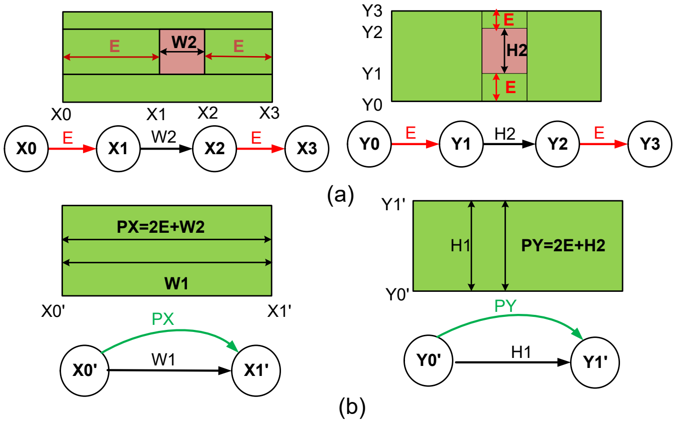

Fig. 2. (a) Horizontal corner stitch and corresponding HCG (left), vertical corner stitch and corresponding VCG (right) of G4 child node, (b) HCG and VCG of G4 considering the propagated constraint from (a) as PX and PY. Here, E, W2, H2, W1, and H1 stand for the minimum enclosure, the minimum width, the minimum length of a die, and the minimum width, the minimum length of a trace respectively.

graph creation is shown in Fig. 2(a) and (b). In the tree shown in Fig. 1(b), G4 is in level 2 and has a child node in level 3. So, G4 is constrained by its child, and the minimum spacing should be propagated from level 3 to level 2. In Fig. 2, HCG and VCG of (a) child node and (b) parent node are shown. The edges with weights of PX and PY are propagated from level 3 to level 2 to ensure the minimum spacing for the child in level 3.

C. Handling Fixed Dimension

In the layout, there are some components like dies and leads, which should have a fixed dimension throughout all solutions. To preserve the fixed dimensions of such components, we have considered two types of vertices in the constraint graph: independent vertex and dependent vertex. Each component width and height are mapped as a fixed edge (an edge with a constant weight) in the constraint graph. Independent vertex locations are randomized while evaluating the graph, whereas the dependent vertex locations are calculated from the corresponding independent vertex. A vertex is called dependent if all incoming or outgoing edges are fixed. All dependent vertices are removable from the constraint graph to reduce graph size as well as the time complexity. The algorithm maintains fixed spacing for a dependent vertex from the corresponding independent one given all constraints are satisfied. All components with fixed dimensions are handled in the same way. For example, in the layout shown in Fig. 3(a), the width (W2) should be fixed as it corresponds to a die. So, in the constraint graph (shown in Fig. 3(b)) the vertex X2 should always maintain W2 distance from the vertex X1. Therefore, X2 becomes dependent on X1. To maintain this dependency, the HCG is modified (shown in Fig. 3(c)). In order to preserve the constraint value, the vertex X2 is bypassed by the blue edge with a weight of W2+E and the edge from X2 is removed that makes X2 removable. A similar approach is applied for the vertical constraint graph to have a fixed height of the component. Thus, while evaluating the constraint graph, any dependent vertex is removed from the longest path, and the dependent vertex location is calculated at the same time with the corresponding independent vertex.

D. Layout Solution Generation

To generate layout solutions, the constraint-aware algorithms [9] are used. Due to the hierarchical consideration, layout generation from a constraint graph has two basic steps: evaluation and propagation. Constraint graph of each node in the tree is evaluated using constraint values, and then the evaluated result is propagated based on the position of the node in the tree structure.

Evaluation: The graphs are evaluated using the longest path algorithm [12] that is widely used in physical design algorithms containing distance constraints. In this algorithm, the vertices of a weighted DAG (directed acyclic graph) are sorted in topological order, and the incremental distance from the source is calculated for each vertex. The maximum distance from the source to each vertex is set as the minimum location of that vertex. This algorithm has a time complexity of O(E), where E is the number of edges in the graph. Like the planar approach, based on the purpose of layout generation, four modes of evaluation have been considered in the hierarchical approach:

- Mode 0: Minimum-sized layouts. In constraint graph evaluation, this mode uses minimum constraint values only.
- Mode 1: Variable-sized layouts. In this mode, the floorplan size of layout solutions is varied arbitrarily.
- Mode 2: Fixed-sized layouts. This mode generates all solutions with a user-defined fixed floorplan size.
- Mode 3: Fixed-sized layouts with fixed component locations. In this mode, not only floorplan size is fixed but also any component in the layout can be located at user-defined valid locations throughout all solutions. This mode is helpful for packaging with pre-defined pin locations.

Propagation: Two types of propagation are required to generate a solution. The first one is propagating the minimum constraint values bottom-up through the tree. In this propagation, the child node CG is evaluated using the longest path algorithm, and the minimum space is then propagated to the parent node CG. The algorithm for bottom-to-top constraint propagation is shown in Algorithm 1. When minimum constraint propagation is done, the root node has all the propagated constraint information and is evaluated for minimum-sized layout generation. Then, the locations are propagated from parent node to child node in a top-down manner. If in Mode 1 with variable-sized floorplan generation, the edge weights are varied in the root node, and then the top-down propagation happens. For the fixed floorplan layout generation mode, the root node CG evaluation is performed using the fixed floorplan algorithms presented in [9], and then the propagation takes place. Therefore, the room for each child node is determined by the corresponding parent node, and the room is propagated until locations of all vertices of each CG are determined. The top-down location propagation algorithm is shown in Algorithm 2.

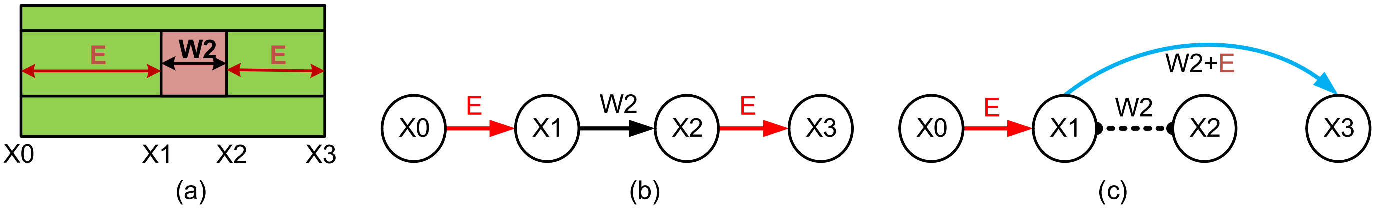

Fig. 3. (a) Horizontal corner-stitched layout, (b) corresponding HCG, (c) modified HCG

Algorithm 1: Bottom-up Constraint Propagation

```
Input: Parent node's constraint graph(G1), Child
  node's evaluated graph (G2)
Output: Updated G1 with propagated constraints from
  evaluated G2
Shared_vertices=list of vertices V, where
        $V \in (G1, G2)$
Location=Dictionary of evaluated G2, where
  key=vertex and value=incremental location of key
for $i = 0$ to $length(Shared\_vertices) - 1$ do
    source= Shared_vertices[i]
    destination = Shared_vertices[i+1]
    V=Location[destination]-Location[source]
    G1.add_edge(source,destination,V)
Return G1
```

Algorithm 2: Top-down Location Propagation

```
Input: Parent node's evaluated constraint graph(G1),
  Child node's evaluated constraint graph(G2)
Output:Updated Location of each vertex in G2
Shared_vertices=list of vertices V, where
        $V \in (G1, G2)$
Parent_Location=Dictionary of evaluated G1, where
  key=vertex and value=incremental location of key
Predecessors=Dictionary of evaluated G2, where
  key=vertex and value=predecessor of that vertex
Location={}
A = Adjacency Matrix(G2)
for $i = 0$ to $length(A) - 1$ do
    if $i$ in Shared_vertices then
        Location[i]=Parent_Location[i]
    else
        P=Predecessor[i]
        Location[i]=Location[P]+A[P][i]
Return Location
```

E. Layout Optimization

If any layout has multiple identical parts, using the hierarchical approach, only one needs to be optimized in a modular design. The optimized result can be replicated for the rest of the parts for faster computation. For example, to optimize a three-phase inverter, a single leg can be considered for optimization, and the results can be reused for the other two phases. This is a major benefit of considering hierarchy in layout design automation. In this paper, two full-bridge power modules consisting of two identical half-bridge modules are demonstrated to illustrate the benefits of a hierarchical approach. By using the constraint graph evaluation methodology, a series of layout solutions is generated by randomizing edge weights of a single half-bridge layout (one symmetrical part). Electrical and thermal models are applied to find the optimum trade-off between the loop inductance and maximum temperature for the module, and a Pareto-front of optimal layouts is then generated from these solutions.

Electrical Model: In the previous work, to quickly evaluate loop parasitics during layout optimization, the Laplacian matrix algorithm has been used. This model takes traces, devices, and leads from the symbolic layout structure as input to form a graph where each edge represents a trace or bonding wire, and each vertex represents a device or lead connection. Combining with the response surface model in [16], the corresponding self-parasitic values can be evaluated for each edge in the graph. Finally, the Laplacian matrix algorithm can be used to evaluate the effective parasitic results between any two vertices. This model has shown a fast and accurate approximation of parasitic results and been hardware validated in [1] for most 2D power module cases. However, this model lacks current density and mutual inductance considerations, which reduces the extraction accuracy for layouts with multiple large traces. Also, for 2.5D and 3D power module cases the model is not accurate enough as it is not considering mutual couplings between components.

To ensure accurate parasitic extraction for the hierarchical layouts in PowerSynth, a new electrical model based on Partial Elements Equivalent Circuit (PEEC) has been developed in [17]. This updated model uses a hierarchical approach and considers both mutual coupling effect and current density while sacrificing some run time. Most importantly, this model can support 2D, 2.5D, and 3D power module layouts. Combined with the hierarchical layout generation algorithms, the new electrical model can optimize more complex geometry as well as 2.5D/3D MCPM layouts.

Thermal Model: Beside electrical consideration, thermal performance is known to be one of the most important considerations while optimizing the MCPM layouts. The thermal model described in [18] and validated in [1], is used to accurately estimate the steady-state thermal performance of a power module. The model takes the layer stack material properties and dimensions as well as steady state power dissipation of each device as input. Finite element analysis (FEA) simulations using Gmsh [19] and Elmer [20] is then run for each device. Then, the temperature, as well as heat flux information, can be stored into multiple rectangular contours. This information can be used later to approximate the thermal coupling effect among devices. The thermal resistance network can be then extracted to quickly evaluate the steady-state thermal performance of each solution. This model is able to predict the result correctly with less than 10% error compared to the FEA simulations and approximately 10,000 times faster.

III. Experimental Results

Hierarchy consideration reduces coordinate correlation among components and produces more feasible layouts compared to the planar approach in [9]. For the sample H-bridge module shown in Fig. 4(a), in the planar approach, all components are in the same hierarchy level (shown in Fig. 4(b)). That results in maximum coordinate correlation among components. Therefore, in the minimum-sized layout solution, traces like 14, 16, 18, 20, 22, 26 are not always in their minimum possible locations (shown in Fig. 5(a)). In addition, the gaps between devices are not equal as well. In Fig. 4(c), the hierarchical tree structure is illustrated, where components in different sub-trees have no correlation among them. For example, the leaf nodes in the tree consisting of devices and their modified parent traces are not correlated with each other. This approach results in a symmetric minimum-sized layout (shown in Fig. 5(b)). So, while generating solutions by randomizing the edge weights, the hierarchical approach gives more variations than the planar approach. A qualitative comparison between two approaches is shown in Table I.

To demonstrate the computational effort reduction in hierarchical MCPM layout optimization, two cases are considered. The first case is a 2.5D full-bridge power module shown in Fig. 8(a), consisting of two symmetrical half-bridge modules. For this module, the optimization is performed to minimize loop inductance and maximum temperature. The module is divided into two parts (left and right) while generating layout solutions. As the layout is symmetrical, only the left half is randomized to generate DRC-clean solutions. These solutions are evaluated for optimizing both objectives. For thermal evaluation, the whole full-bridge base plate is characterized. For electrical evaluation, to measure the loop inductance, the full-loop should be from DC$_1$+ to DC$_2$- assuming a load is connected between Out$_1$ and Out$_2$. Due to the hierarchical approach, only the left symmetrical part is used for electrical evaluation. Therefore, the path is considered from DC$_1$+ to DC$_1$- with the only difference of the load inductance, because of the symmetric lower path. After calculating the half-bridge loop inductance, the load inductance is added to get the full-bridge loop inductance value. So, the above-mentioned methodology reduces the computational effort by half.

TABLE I
Qualitative comparison between hierarchical and non-hierarchical approach

Characteristics	Hierarchical Approach	Planar Approach
Benefits of symmetric geometry	Yes	No
Coordinate correlation	Less	Higher
Reusability of optimization results	Yes	No
Computational complexity	Less	Higher
Solution space	Smaller	Larger

For optimization, about 1000 layout solutions are considered for a set of varying floorplan sizes from 110×66 mm^2 to 160×85 mm^2 with a load inductance of 8 nH. The result is shown in Fig. 8(b). Here, from the Pareto-front, Layout A (Fig. 9(a)) has the highest temperature with the minimum loop inductance, Layout C (Fig. 9(c)) has the highest loop inductance with the lowest temperature rise. However, Layout B (Fig. 9(b)) has the optimum value among all designs. From the Pareto-front, the optimum solution can be selected, exported to 3D modeling software such as ANSYS Q3D and SolidWorks for further analysis and fabrication.

To illustrate the run time improvement with the hierarchical approach over the non-hierarchical one, the second case is chosen (shown in Fig. 6). The minimum-sized solution is shown in Fig. 7. In this solution, a dashed line is drawn to reflect symmetricity of the layout. Also, each group is outlined with black-dotted lines on the left half of the layout to differentiate among correlated groups in the layout. Though the linear approximation model provides a good approximation for loop parasitic of the first layout case in Fig. 8(a), it is not suitable for this case. For the full-bridge and half-bridge layouts in Fig. 6, where the traces are much bigger, the previous method would produce some overestimated result without current density consideration. Therefore, to have a better loop parasitic evaluation during optimization, the model from [17] has been used. However, this model has an asymptotic growth of O(N^2), which results in a much longer evaluation time for the full-bridge versus half-bridge layout. To evaluate the power loop inductance from DC$_1$+ to DC$_1$- hierarchy consideration improves computational efficiency significantly. For this example case, power loop inductance evaluation time is approximately 0.43s and 5.05s using hierarchical and planar approach, respectively. From the result, dividing the full bridge into symmetrical half-bridge parts, around 12 times speedup can be achieved.

IV. Conclusions and Future Work

Our hierarchical layout algorithm provides a structural design automation method towards optimizing complicated large-scale power electronics layouts. It reduces computational complexity and conflicts among layout components over the traditional planar approach. This methodology is scalable to handle an arbitrary number and different types of components

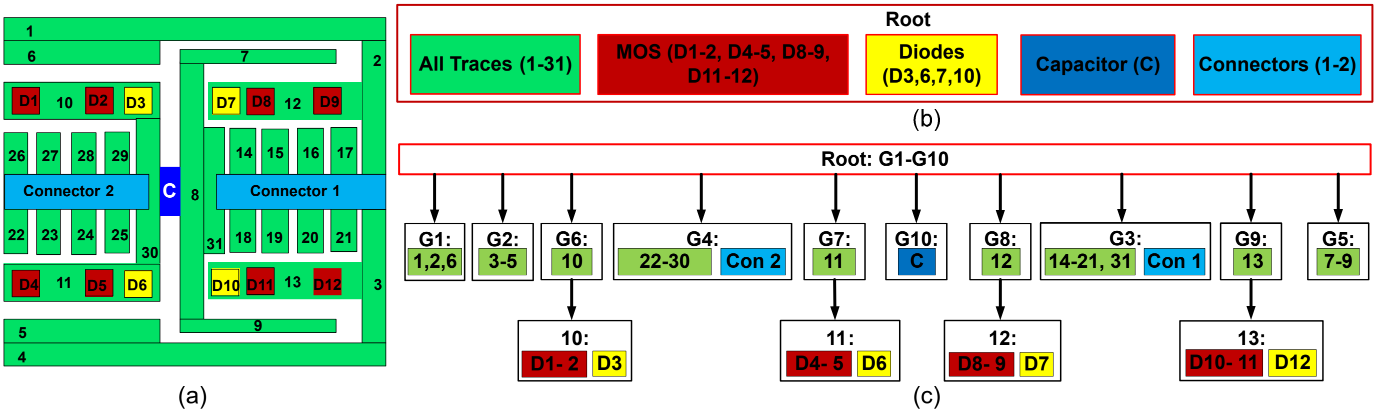

Fig. 4. (a) Layout of an H-bridge power module, representation with a (b) planar structure, (c) tree structure.

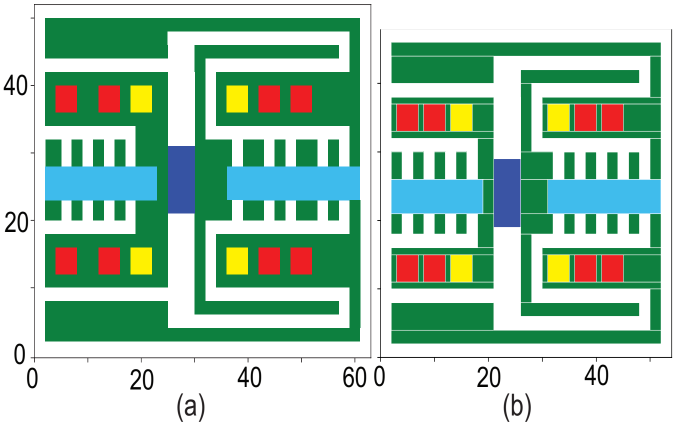

Fig. 5. Minimum-sized layout: (a) without hierarchy (63 × 52), (b) with hierarchy (54 × 48).

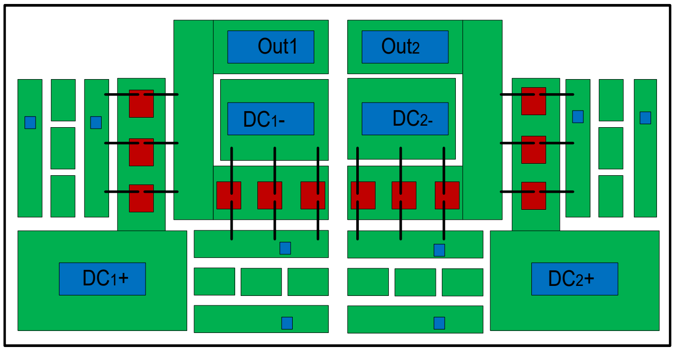

Fig. 6. Layout of a full-bridge power module

in the layout. The next step is to extend the algorithm with generic layout connections so that it can optimize rigid bonding wires and vertical vias in advanced power packaging. Then we will update PowerSynth to optimize 3D power modules layout.

V. Acknowledgment

The authors are grateful to Si Huang, Tristan Evans, and Shilpi Mukherjee for their help throughout the research.

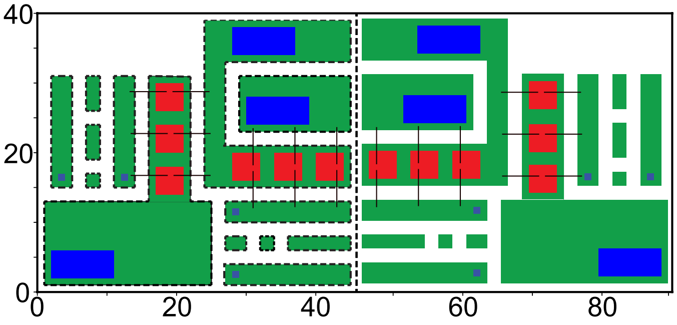

Fig. 7. Minimum-sized solution (94 × 40)

References

[1] T. Evans *et al.*, "PowerSynth: A Power Module Layout Generation Tool," *IEEE Transactions on Power Electronics*, 2018.

[2] T. Wu *et al.*, "Automated Heatsink Optimization for Air-Cooled Power Semiconductor Modules," *IEEE Transactions on Power Electronics*, vol. 34, no. 6, pp. 5027–5031, June 2019.

[3] P. Ning *et al.*, "An Improved Planar Module Automatic Layout Method for Large Number of Dies," *CES Transactions on Electrical Machines and Systems*, vol. 1, no. 4, pp. 411–417, December 2017.

[4] I. Laird *et al.*, "A design optimization tool for maximizing the power density of 3-phase dc–ac converters using silicon carbide (sic) devices," *IEEE Transactions on Power Electronics*, vol. 33, no. 4, pp. 2913–2932, April 2018.

[5] L. Li *et al.*, "Automatic layout design for gate driver of SiC MOSFET," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, Oct 2017, pp. 7941–7946.

[6] D. Zhang *et al.*, "An automatic EMI filter design methodology for electric vehicle application," in *2017 IEEE Energy Conversion Congress and Exposition (ECCE)*, Oct 2017, pp. 4497–4503.

[7] Zhenxian Liang *et al.*, "Integrated CoolMOS FET/SiC-Diode Module for High Performance Power Switching," *IEEE Transactions on Power Electronics*, vol. 20, no. 3, pp. 679–686, May 2005.

[8] C. Martin *et al.*, "Gate Circuit Layout Optimization of Power Module Regarding Transient Current Imbalance," *IEEE Transactions on Power Electronics*, vol. 21, no. 5, pp. 1176–1184, Sept 2006.

[9] I. Al Razi *et al.*, "Constraint-Aware Algorithms for Heterogeneous Power Module Layout Synthesis and Optimization in PowerSynth," in *IEEE Workshop on Wide Bandgap Power Devices and Applications*, October 2018, pp. 323–330.

[10] P. Ning *et al.*, "Automatic Layout Design for Power Module," *IEEE Transactions on Power Electronics*, vol. 28, no. 1, pp. 481–487, Jan 2013.

[11] J. K. Ousterhout, "Corner Stitching: A Data-Structuring Technique for VLSI Layout Tools," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 3, no. 1, pp. 87–100, January 1984.

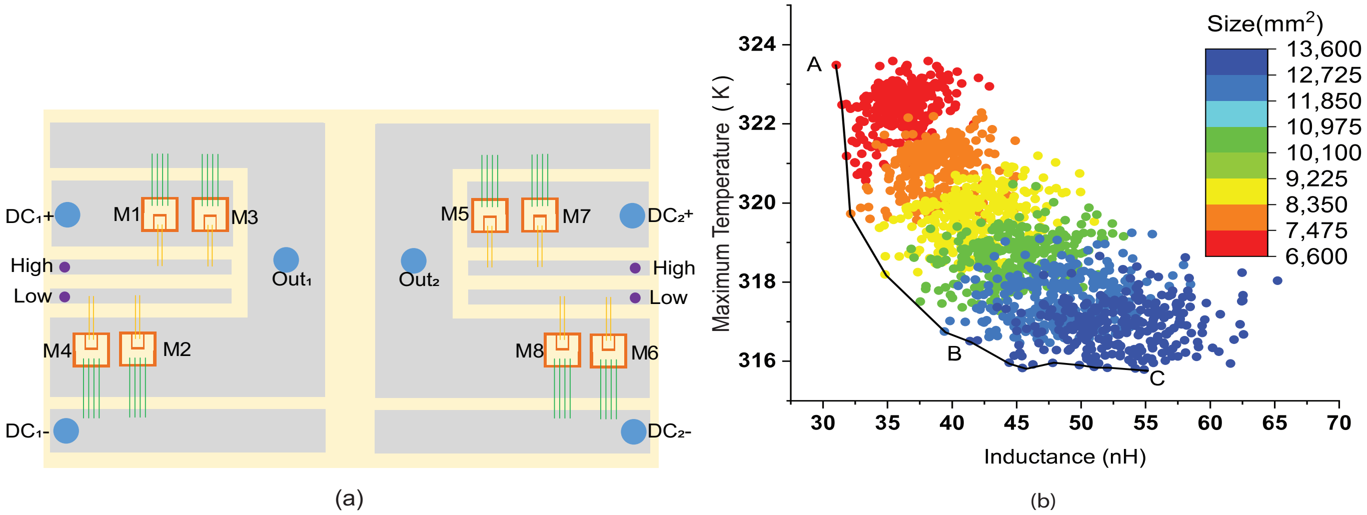

Fig. 8. (a) Layout of a full-bridge power module, b) Pareto-front of optimum layouts with three selected points A, B, and C.

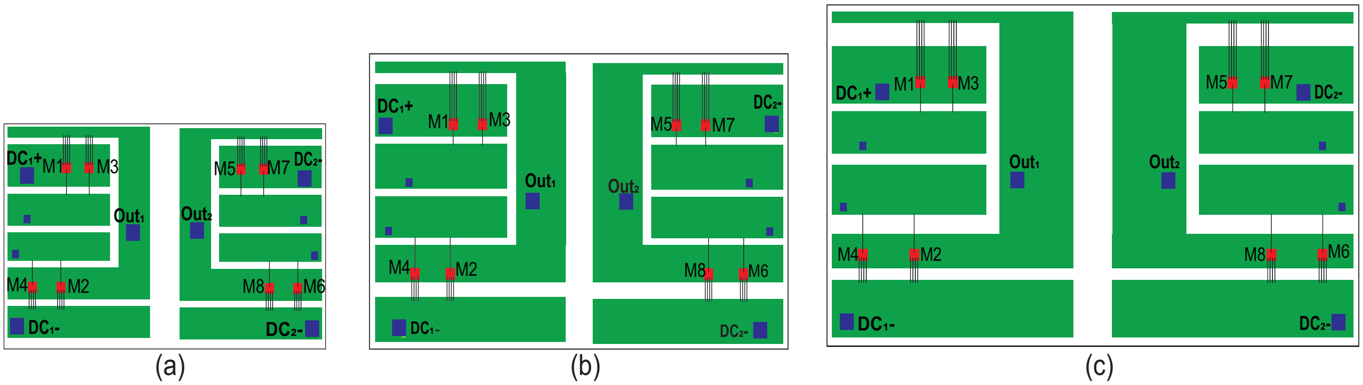

Fig. 9. Three layouts from the Pareto-front in Fig. 8(b) (a) A (110 × 66), (b) B (140 × 75), (c) C (160 × 85).

[12] Y. Z. Liao and C. K. Wong, "An Algorithm to Compact a VLSI Symbolic Layout with Mixed Constraints," in *ACM Design Automation Conference*, June 1983, pp. 107–112.

[13] U. Choi *et al.*, "Effect of asymmetric layout of igbt modules on reliability of motor drive inverters," *IEEE Transactions on Power Electronics*, vol. 34, no. 2, pp. 1765–1772, Feb 2019.

[14] R. Azar *et al.*, "The current sharing optimization of paralleled igbts in a power module tile using a pspice frequency dependent impedance model," *IEEE Transactions on Power Electronics*, vol. 23, no. 1, pp. 206–217, Jan 2008.

[15] J. K. Ousterhout *et al.*, "The Magic VLSI Layout System," *IEEE Design and Test of Computers*, vol. 2, no. 1, pp. 19–30, Feb 1985.

[16] Q. Le *et al.*, "Response Surface Modeling for Parasitic Extraction for Multi-Objective Optimization of Multi-Chip Power Modules (MCPMs)," in *IEEE Workshop on Wide Bandgap Power Devices and Applications*, Oct 2017, pp. 327–334.

[17] Q. Le *et al.*, "PEEC Method and Hierarchical Approach Towards 3D Multichip Power Module (MCPM) Layout Optimization," in *IEEE International Workshop on Integrated Power Packaging*, 2019.

[18] B. W. Shook *et al.*, "Multi-objective layout optimization for Multi-Chip Power Modules considering electrical parasitics and thermal performance," in *IEEE Workshop on Control and Modeling for Power Electronics*, June 2013, pp. 1–4.

[19] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-D Finite Element Mesh Generator with Built-in Pre- and Post-Processing Facilities," *International Journal for Numerical Methods in Engineering*, pp. 1309–1331, September 2009.

[20] M. Malinen and P. Råback, *Elmer finite element solver for multiphysics and multiscale problems*, 09 2013, vol. 19, pp. 101–113.