Fast and Accurate Inductance Extraction for Power Module Layout Optimization Using Loop-Based Method

Quang Le[a], Imam Al Razi[b], Yarui Peng[b], H. Alan Mantooth[a]

[a] Electrical Engineering Department, [b] Computer Science and Computer Engineering Department

University of Arkansas, Fayetteville, AR, USA

qmle@uark.edu, mantooth@uark.edu

***Abstract*—Electrical parasitics, especially parasitic inductance, play an essential role in enhancing power module design performance through reducing voltage overshoot and switching power losses. In this paper, a new electrical parasitic extraction model is developed, inspired by the loop-based extraction method from VLSI. This model shows only within 10% error compared to the FastHenry method while being orders of magnitude more efficient in run time and memory. For the same design, this method takes less than 0.5 seconds on the same machine to achieve similar accuracy. This method also significantly reduces the number of elements in the extracted netlist, which reduces the complexity for loop evaluation a few thousand times. Utilizing the divide-and-conquer strategy, this model demonstrates many advantages over previous work in layout optimization and post-layout simulation. The model is also attractive for use with optimization routines, and therefore has been used in the latest PowerSynth layout optimization tool.**

Keywords—Design Automation, Parasitic Extraction, MCPMs

I. Introduction

In recent years, the maturation of wide-bandgap (WBG) semiconductor researches has allowed more and more commercially available devices in the market [1][2]. This achievement, in turn, drives the development of power module packaging research [3]. These efforts allow power module designs with a much higher power density and switching speed at higher operating voltage. However, high voltage overshoot issues and excessive switching losses arise due to parasitic inductance at higher switching speeds. Thus, the designer needs to pay more attention and mitigate these issues during the layout design stage. The traditional layout design process for these power modules usually requires time-consuming electrothermal simulations and a manual and iterative design process that can take days to weeks. To address these challenges, the focus has been shifted towards design automation tools for power modules in recent years [4-6]. These tools make the design process more efficient and effortless. Among these tools, PowerSynth [4] has been shown to be the most mature and developed due to its efficient and generic layout engine along with its fast and accurate electrical and thermal models for optimization [5]. Recent development efforts from the PowerSynth v2 architecture (Fig. 1) have allowed automated layout design for 2.5D (multiple substrates in a planar package) and 3D (multiple device layers stacked on the same substrate) modules. Within these efforts, a new layout generation algorithm has been demonstrated in [7]. While the layout generation algorithms are efficient and generic, 2.5D and 3D solutions impose higher electrothermal evaluation complexity during the optimization routine. Therefore, more efficient models are needed to ensure a high performance and robust optimization process while maintaining similar evaluation accuracy. In a collaborative effort with the Army Research Lab (ARL), ParaPower, a tool developed by ARL, has been utilized for its rapid and accurate thermal and mechanical stress and strain analyses within PowerSynth's layout optimization process. A software application programming interface (API) between PowerSynth and ParaPower has been developed [8] for this purpose. On the electrical parasitic modeling side, the Partial Element Equivalent Circuit (PEEC) based model presented in [9] is quite efficient and accurate for most 2D layout parasitic extraction for previous versions of PowerSynth. However, as the PEEC matrices are dense and contain many elements, the computation complexity also increases quadratically with the number of

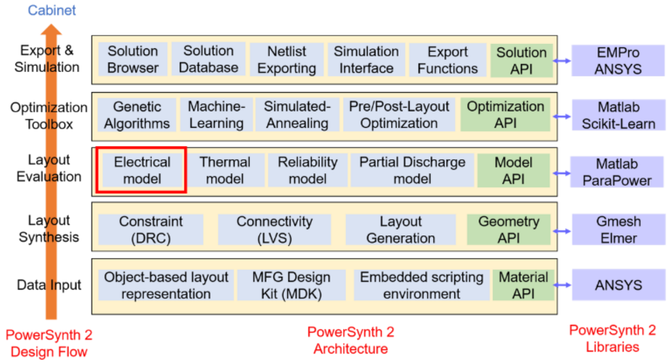

Fig. 1. PowerSynth v2 architecture.

This material is based on work supported by The National Science Foundation under Grant No. EEC-1449548. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science

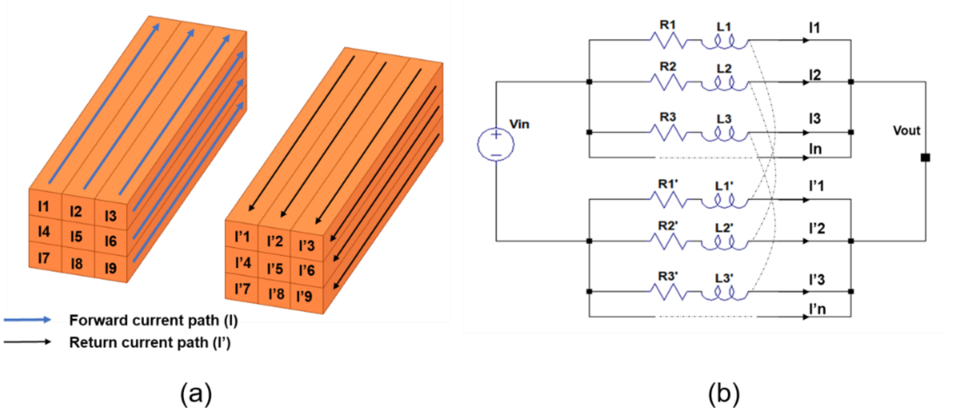

Fig. 2. (a) 3D mesh with forward and return currents (b) circuit representation of loop-based method.

elements. This significant number of elements also poses issues for post-layout circuit simulation analyses, such as long runtime or convergence failure. Therefore, a new model needs to be developed to address these issues.

Many different parasitic extraction and estimation methods have been recently presented by design automation researchers. In [6], a model based on the method of moments technique was presented. In this study, the layout was meshed into many elements to find the current loop and calculate the total loop inductance. This model, however, did not consider the mutual inductance effect among multiple loops. The studies in [10] and [11] apply the response surface modeling technique to form a fitted equation for the power module trace inductance. These models are very efficient for the linear approximation of the loop inductance value in power module layouts. However, these models again did not consider the mutual inductance impact and only work for a simple layout with long and thin traces. The Current-Bunch tool presented in [12] operates based on the current-bunch concept. This study proposes fast and memory-efficient loop parasitic extraction while considering mutual inductance among different nets. However, this model assumes a uniform current distribution inside the conductor, which might lead to inaccuracy for large conductor traces in MCPMs. Moreover, non-uniform meshing in the tool poses an unstable solution as the number of mesh elements increases. More recently, the work in [13], applied a boundary meshing technique to reduce the number of partial elements. This meshing technique reduces the computation effort significantly. However, similar to the PEEC approach, the number of mesh elements also increases significantly in a 3D layout problem. These drawbacks further necessitate a new modeling approach. This work presents a new model based on the loop-based technique [14-18] in Very-Large-Scale-Integration (VLSI). This model has been proven to be efficient and accurate for 3D muti-conductors and on-chip layout interconects extraction in VLSI. This model addresses most of the issues from the above-mentioned methods while maintaining similar performance. Additionally, utilizing the divide-and-conquer strategy, this model can take advantage of parallel processing techniques, which further improves its computational efficiency. Most importantly, the reduced extracted netlist from this model is more efficient for post-layout circuit simulation.

This paper is organized as followed: Section II details the methodology of the loop-based approach. Then, section III shows some validation results versus FastHenry [19] and this method for 2D and 3D layout cases. Section IV shows optimization results using PowerSynth and this model. Finally, Section V concludes the paper.

II. Methodology

A. *Application of Loop-Based Method in MCPMs*

In a VLSI application, the loop impedance approach is usually applied for signal (forward) and their corresponding ground (return) paths as in Fig. 2 (a). These wires together can form circuit loops as shown in Fig. 2 (b), where their impedance can be evaluated using the current values through each loop. In VLSI, the designer usually knows the signal and ground wires in most applications. Similarly, in the case of printed circuit board based applications, there is a return path on the printed circuit board (PCB) ground plane. Thus, the current loops in these applications are typically easily identifiable. In power module applications, however, the current loop is not defined by a ground plane. The power module's backplane is usually capacitively and mutually coupled with the signal and power traces on the top side. In the case of MCPM layout, the current loops are defined by the locations of positive and negative power/signal terminals. These loacations are also varied in a layout optimization problem. Because of this, methods such as PEEC and Finite Element Analysis (FEA) are frequently used. The power module top layer is usually discretized into a significant number of small elements where these methods can automatically define the current paths. Hence, in this work, the process shown in Fig. 3 is applied prior to the impedance calculation step. In this paper, to demonstrate and validate the loop-based model capability in MCPM, three different layouts have been chosen. A simple 2D layout in Fig. 5 is used to demonstrate the flow of this method (Fig. 3). The more complicated 2D layout in Fig. 6 is used for our optimization study. Finally, a 3D layout as shown in Fig. 7 is used to validate our model's 3D extraction capability.

B. *Layout Engine API*

First, the PowerSynth layout engine feeds each layout solution through a software Application Programming Interface (API). This API converts the layout information into

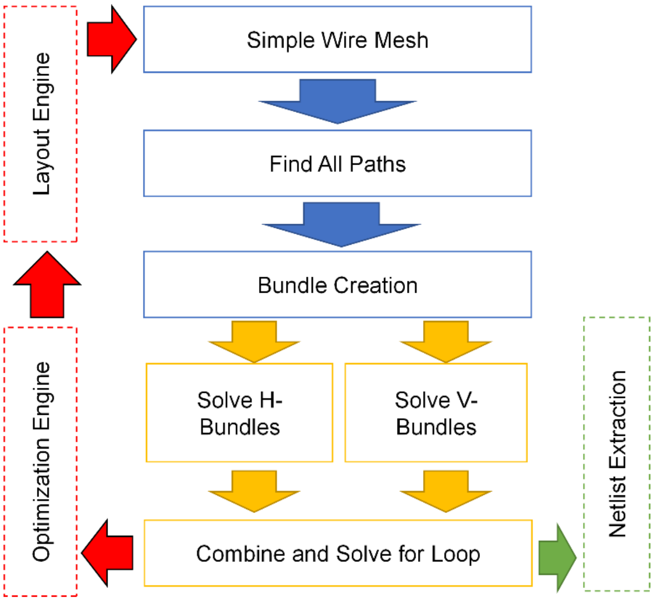

Fig. 3. Loop-based method design flow.

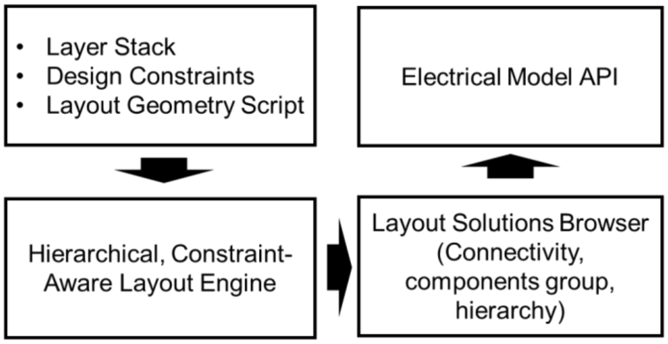
Layer Stack
Design Constraints
Layout Geometry Script
Electrical Model API
Hierarchical, Constraint-Aware Layout Engine
Layout Solutions Browser (Connectivity, components group, hierarchy)

Fig. 4. Layout engine to electrical API.

geometrical information used in the parasitic extraction model. The high-level flow is shown in Fig. 4. The layer stack of the module contains information about the dimensions of the substrate, material information, etc. A set of design constraints is required to ensure the solutions are design rule check (DRC) clean. Additionally, a layout script is required to gather the geometrical and connectivity information. The hierarchical geometry script (shown in [5]) takes the initial layout geometry information from the user. Components are inserted in the layout in a group-wise fashion. Physically connected components are inserted in the same group. The hierarchical layout information is stored in a tree structure, where each group is a node of the tree. This hierarchy and connectivity are preserved throughout the solutions. The layout engine considers all design constraints while generating the solutions. Thus, it ensures a 100% DRC-clean solution space. The solution layouts are fed into the electrical model through the API, which converts the layout geometrical information into equivalent netlist nodes. The layout solution data is arranged into a hash table, where each row of the table contains coordinates and dimensions of each group. Depending on the structure of each layout group, three pre-defined orientations: vertical (V), horizontal (H), and planar (P) of the traces are considered. The meshing algorithm can efficiently generate the mesh for a single direction or multiple directions of current flow using these pre-defined orientations.

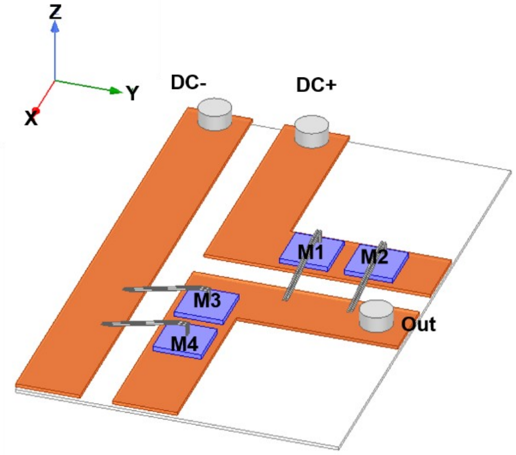

Fig. 5. Layout case 1: simple half-bridge module.

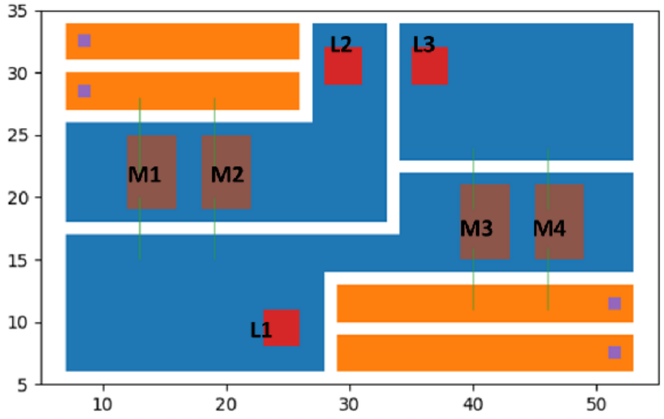

Fig. 6. Layout case 2: half-bridge module for optimization study.

C. Loop-based Evaluation for Each Bundle

Through the layout engine API, the layout geometry is transformed into a simple wire mesh using trace edge location, device center location, and bondwire landing location. These locations are represented as nodes where edges are formed if they share the same layout traces or bondwires. A path-finding based on depth-first search algorithm (shown in Algorithm 1) then finds all existing paths from the source (node 1) and sink (node 12) and transforms the wire mesh into a directed graph, as shown in Fig. 8 (a). Next, the layout is further discretized into disconnected loops in both horizontal and vertical orientation from this directed graph. These circuit loops are defined as bundles where each bundle contains forward and return currents that share the same start and endpoints. As shown in Fig. 8 (b) and (c), multiple bundles specified by different colors are defined. Here, two additional nodes 13 and 14 are added to form the return paths for horizontal bundles

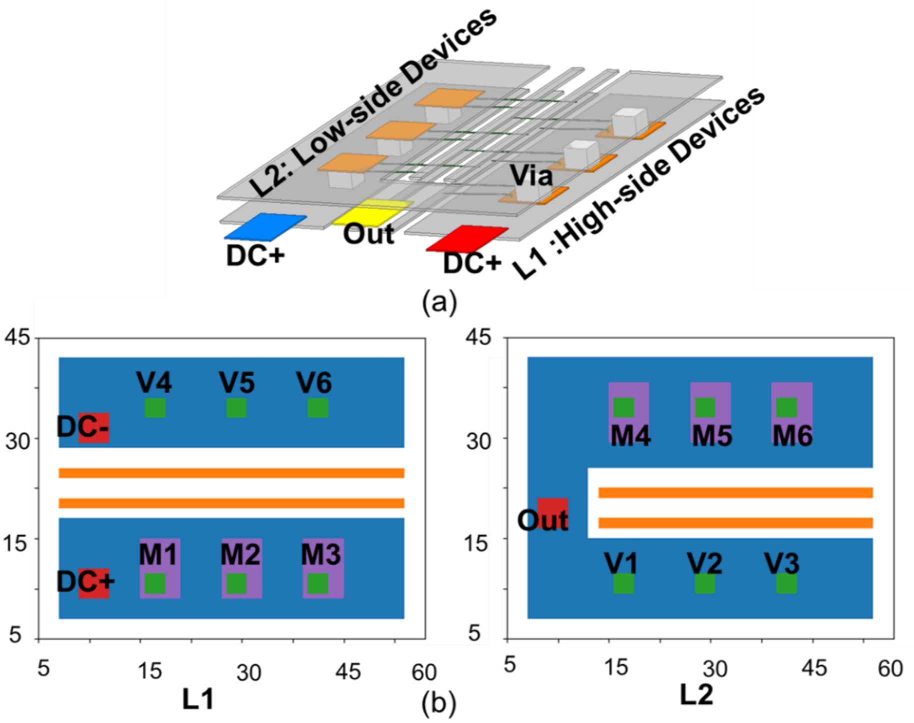

Fig. 7. Layout case 3: A 3D Half-bridge module (a) 3D view of MCPM layout example (b) Layouts of bottom (L1) and top (L2) layers.

(Fig. 8 (b)). The volumetric mesh similar to Fig. 2 (b) can be applied to each bundle where the loop-based approach is used to compute the impedance. There are several advantages of using this volumetric mesh. Firstly, by increasing the number of mesh elements, the skin depth effect can be captured. Secondly, compared to the non-uniform meshing technique in [12], the number of mutual inductance calculations can be optimized using this volumetric mesh. For example, the mutual inductance between elements 1 and 5 is the same as the mutual inductance between 3 and 5 in Fig. 2 (b). A hash-table can be formed to store the calculated mutual inductance value of each pair of conductors, where the hash-key is the geometric distance. The mutual inductance value can be reused many times for elements sharing the same geometric distance. Hence, the mutual inductance evaluation effort can be reduced significantly. From this point, the impedance of each bundle can be calculated using the loop-based method which is described in the next section.

First, a set of analytical equations can be used to compute the resistance, inductance, mutual inductance of rectangular bars [20]. All impedance values are stored in an n-by-n matrix **P** where n is the total number of elements, **P**(i,i) is the partial self-impedance, and **P**(i,j) is the partial mutual impedance between element i and j.

$$\mathbf{P} = \begin{pmatrix} Z_{1,1} & Z_{1,2} & \cdots & Z_{1,n} \\ Z_{2,1} & Z_{2,2} & \cdots & Z_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{n,1} & Z_{n,2} & \cdots & Z_{n,n} \end{pmatrix} \qquad (1)$$

Where:

$$\boldsymbol{Z}_{i,j} = R + j\omega L \qquad (2)$$

```
Algorithm 1: Digraph formation
  Input  : Initial Wire Mesh Graph (G)
  Output: Digraph from source to sink (D)
  if $exist\_path(G,source,sink)$ then
      paths = depth-first-search(source,sink)
  else
      raise error: no path exist
      return
  for $i \leftarrow 0$ to $len\_of(paths)$ do
      /*Get the array of nodes on paths*/
      p = paths[i]
      /*Loop through all node to form digraph*/
      for $j \leftarrow 0$ to $len\_of(p)-1$ do
          if $not(exist\_path(D, p[j], p[j+1]))$ then
              add edge(D, p[j], p[j+1])
```

An n-by-k mesh matrix M is formed where k is the number of forward current elements. In this matrix **M**, if an element i is in the forward current path, **M**(i, j) is 1, and 0 otherwise. A column vector **u** is also formed with n rows of one. Two equations are then formed and evaluated as shown below:

$$\boldsymbol{Pa} = \boldsymbol{u} \qquad (3)$$

and

$$\boldsymbol{PB} = \boldsymbol{M} \qquad (4)$$

Vector **a** is used to compute the current distribution in each element, which is unified to have a norm sum of 1. The result from matrix **B** along with this unified vector is then used to

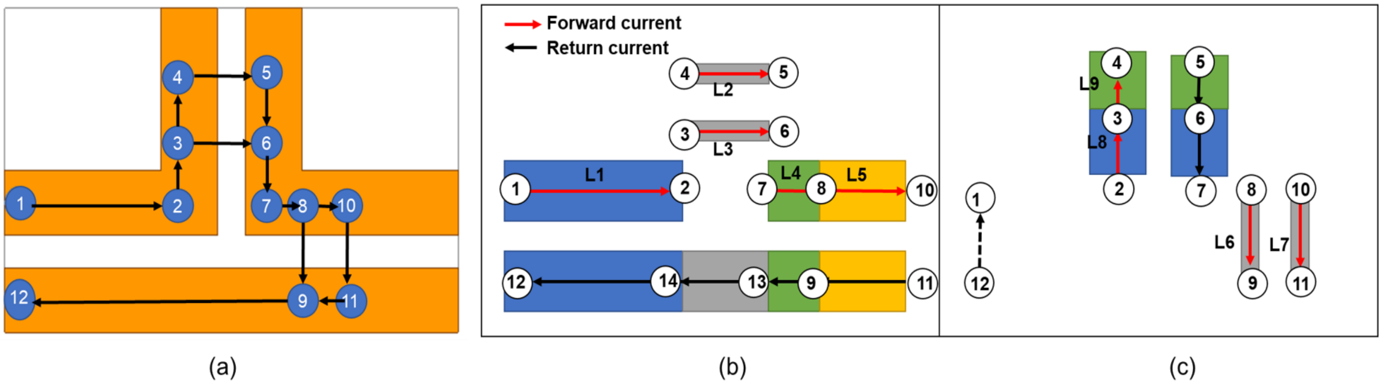

Fig. 8. (a) Digraph formation for half-bridge (b) Horizontal bundles (c) Vertical bundles.

obtain the current matrix **I**, whose the *j*-th column vector **I**(*j*) can be calculated using:

$$\boldsymbol{I}(j) = \boldsymbol{B}(j) - V_{out} \times \boldsymbol{a}, where\ j = 1,2,\dots,n \tag{5}$$

In case there exists at least one return path in the bundle where the return current equals the total sum of forward currents:

$$V_{out} = \frac{\Sigma \boldsymbol{B}(i,j)}{\Sigma \mathbf{a}} \tag{6}$$

However, there are cases where this is not true in the power module, and the bundle only contains open loops or the return path cannot be found due to multiple current branches. An example for this can be seen in Fig. 8 (c), where loops 6 and 7 do not have a corresponding return path. In such a case, V_{out} is simply set to be 0. Once the current matrix **I** is calculated, with *k* being the total number of loops found in the path finding process, the *k*-by-*k* loop impedance matrix **Z** can be found by:

$$\boldsymbol{Z} = (\boldsymbol{M}^{\boldsymbol{T}} \times \boldsymbol{I})^{-\boldsymbol{1}} \tag{7}$$

By dividing the layout into multiple bundles, the total complexity for this algorithm is O(N× M^3) where N is the number of bundles and M is the average number of elements in each bundle. M is usually very small in the case of a power module. Once the impedance of each bundle is calculated, they are stored in matrix **A**.

$$\mathbf{A} = \begin{pmatrix} Z_{loop_{1,1}} & Z_{loop_{1,2}} & \cdots & Z_{loop_{1,k}} \\ Z_{loop_{2,1}} & Z_{loop_{2,2}} & \cdots & Z_{loop_{2,k}} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{loop_{k,1}} & Z_{loop_{k,2}} & \cdots & Z_{loop_{k,k}} \end{pmatrix} \tag{8}$$

A simple matrix evaluation can be formed to evaluate the total loop impedance of the layout using:

$$\boldsymbol{A}\boldsymbol{I}_i = \boldsymbol{V}_i \tag{7}$$

where **A** is a matrix of size *k*-by-*k*, $\mathbf{I}_{\boldsymbol{i}}$ is the current vector for the total current through each loop and $\mathbf{V}_{\boldsymbol{i}}$ is the input voltage vector. In this example, A has a size of 9×9 for five horizontal and four vertical loops found from the path finding algorithm.

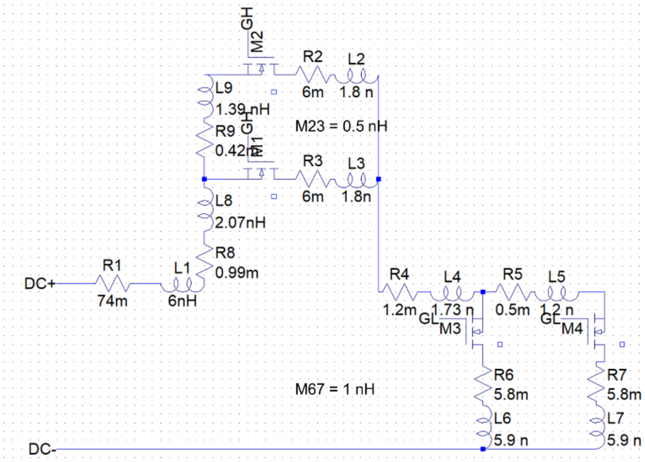

Fig. 9. Extracted netlist for layout 1.

III. Model Validation Results

A. 2D MCPM Cases Validation

Two different 2D MCPM half-bridge layouts are used to show the accuracy of this method. The first example is the simple layout shown in Fig. 5, and a more complicated layout is shown in Fig. 6. For the layout in Fig. 6, L2 is DC+, L1 is Output, and L3 is DC-. The results for the parasitic extraction among different methods for the layouts in Fig. 5 and Fig. 6 are shown in Table I. For both designs, the extraction results show less than 7% error compared to FastHenry extraction using both PEEC and loop-based methods. The total formulation time is the time required for the tool to compute all self and mutual values. For the first case, the formulation time is similar to the PEEC method. However, in the second example, the formulation time is 5 × faster than PEEC. This is because the loop method only calculates the mutual inductance inside each bundle and ignores the bundle-to-bundle mutual inductance. While this might affect the extraction accuracy, the loop-based method considers most mutual inductance among the main current conducting paths. A full bundle-to-bundle mutual consideration can always be provided once the designer selects an optimized layout. An

Table I: Extraction Results and Performance Comparison Among Different Methods

Layout Case	Methods	L_{Loop} (nH)	Run Time (s)			Netlist size	Speed up	Error
			Formulation	Evaluation	Total Time			
Layout 1	FastHenry	17.3	---	---	8	1	---	---
	PEEC [9]	16.1	0.5	0.345	0.8345	1820	9.6×	6.9%
	This work	16.5	0.42	1.4m	0.434	9	18.6×	4.6%
Layout 2	FastHenry	15.3	---	---	22.9	1	---	---
	PEEC [9]	14.5	1.3	2	3.3	32310	11.5×	5%
	This work	15.6	0.26	2.5m	0.285	18	82×	1.9%
Layout 3	FastHenry	7.93	---	---	25	1	---	---
	This work	8.54	0.8	2.43m	0.824	50	30×	7.1%

example in the optimization section will show that this approach can accurately direct the optimization engine. As for the loop evaluation, the loop-based method shows up to 800× speed up for matrix evaluation in the second layout case, thanks to the much smaller matrix size. This small sized matrix is more beneficial as the layout design gets more complicated. Between the two 2D MCPM layouts, the loop-based method has shown a similar performance in evaluation time, while the PEEC method is almost 2× slower. Moreover, the small netlist size from the loop-based method would converge more rapidly during post-layout circuit simulation analysis. The netlist for layout case 1 is illustrated in Fig. 9. There are 18 R, L elements representing the impedances for nine forward loops in Fig. 8.

Additionally, since each bundle can be computed independently, multi-threading and multi-processing can be easily implemented to speed up the formulation and evaluation time. This cannot be done with the dense matrix from the PEEC model.

B. 3D MCPM Case Validation

A hybrid 3D MCPM structure is considered as shown in Fig. 7 (a) to demonstrate the 3D extraction capability. Planar views of each layer are shown in Fig. 7 (b). In the bottom layer (L1), the high-side devices' drains, DC+ and DC- terminals, and in the top layer (L2), the OUT terminal with the low-side devices drains are placed. Via-type metallic post connections are used to connect the source side of devices in between two layers. Wire bonding has been used for gate loop connections. This 3D layout has been evaluated using the proposed electrical model and compared against the FastHenry electrical model.

In this design, since the x-coordinates of most high-side and low-side devices are the same, the number of x-bundles created is ensured to be minimum. This ensures the most accurate extraction using the loop-based method. Here, since the current implementation of the algorithm only creates bundles for the x and y direction, the parasitics from the vias are not considered. Hence, they are shorted in both the loop-based model and FastHenry for a fair comparison. Unlike the 2D design in Fig. 4, there are multiple current branches in this case that do not ensure the same total current to DC-. Because of this, for each bundle, all traces are treated as open loops to ensure an accurate self and mutual calculation. The results for the loop extraction are shown in Table I. This has shown about 30 × speed up versus FastHenry while maintaining good accuracy (7.1%). While the

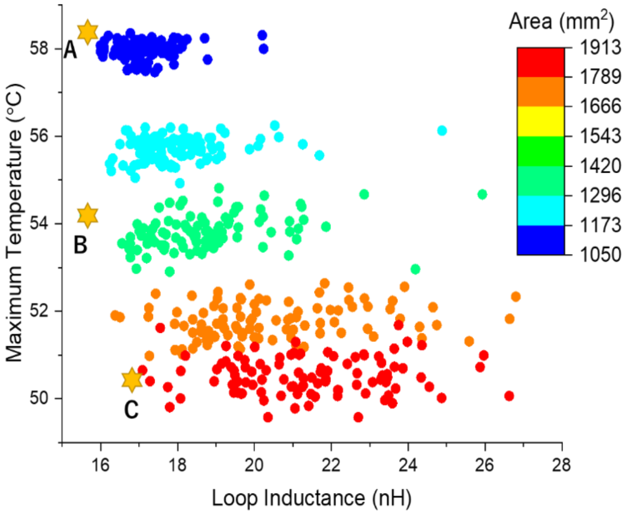

Fig. 10. Solution space for the electrothermal optimization.

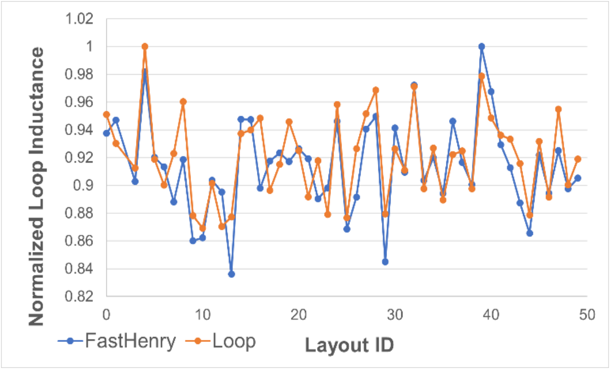

Fig. 11. Optimization trend comparison.

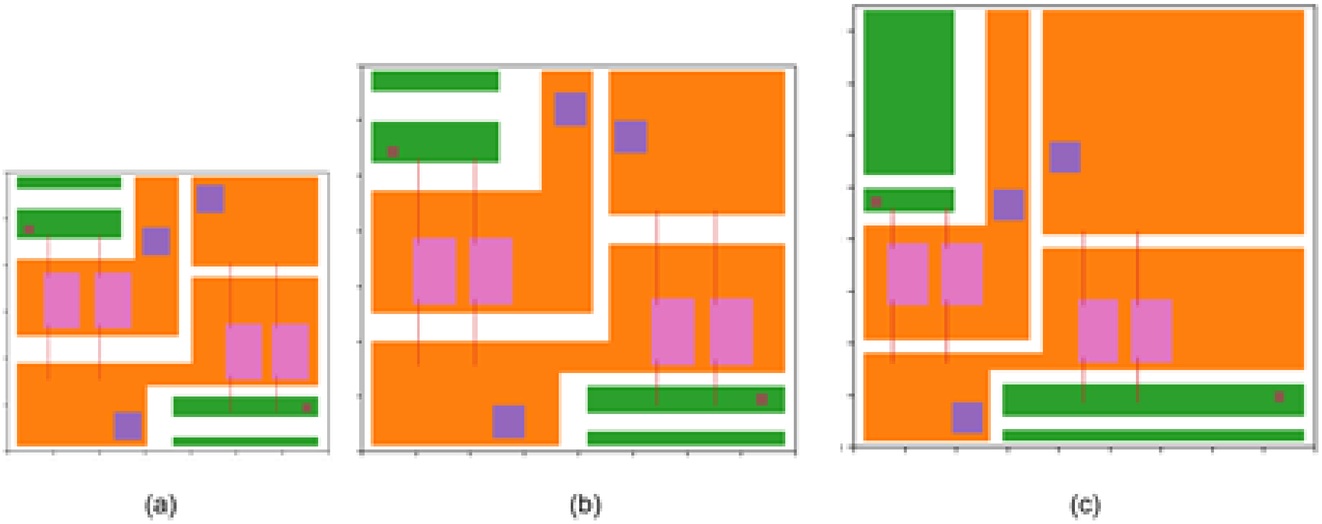

Fig. 12. Optimized solution for (a) 35×30 (mm^2) (b) 40×35 (mm^2) (c) 45×42.5 (mm^2).

3D extraction is accurate in this example, the current implementation is not yet stable for 3D layout optimization. This is because the horizontal and vertical bundles are created based on the coordinates of devices, leads, bondwire, and locations on a flat level. While these locations are varied using the layout optimization algorithm in [7], many bundles are created on the flat level having a small size. This small size might affect the accuracy of the inductance equations, which require a long wire length. Hence, the bundle creation algorithm needs to be updated in the future to mitigate this problem.

IV. Layout Optimization Results

A. Layout Optimization Performance

The second MCPM layout as shown in Fig. 6 has been chosen for the optimization study. For this layout, L2 is DC+, L1 is Output, and L3 is DC-. This optimization study aims to minimize loop inductance from DC+ to DC-, and devices' maximum temperature. Using the layout generation and optimization engine in [7], 500 layout solutions are generated with various footprint sizes from 1050 (mm^2) to 1920 (mm^2). An electrothermal optimization is performed using the thermal model in [8] and this loop-based model. Fig. 10 shows loop inductance values ranging between 15 nH to 27 nH and maximum device temperatures from 50$°C$ to 59$°C$. The total time and mesh size for electrical evaluation of 500 layouts are 143.5 s using this model, 1000 s using PEEC [9], and 11,500 s using FastHenry. These results have shown the advantage of using the loop-based method, which is much faster than the previous PEEC model [9] and FastHenry. More importantly, the loop-based model only needs 825 mesh elements to achieve similar accuracy versus 32,310 elements using PEEC and 2,757 elements using FastHenry. This shows the loop-based method to be more memory efficient than PEEC and FastHenry. Because of this, PowerSynth can quickly explore the solution space using this method.

Table II: Selected Optimized Layouts

	L_{loop} (nH)	Max Temp. $(°C)$	Area (mm^2)
A	15.7	58.4	30 × 35
B	15.5	54.2	40 × 35
C	16.8	50.4	45 × 42.5

B. Validation of the Optimization Capability

While this model is much faster than FastHenry, it is essential to understand how the model performs during the optimization routine. Hence, PowerSynth-FastHenry API has been used to verify the model capability for layout optimization. The first 50 layouts with the same design footprint of 30×35 (mm^2) in the solution space are evaluated using FastHenry and loop-based models. The maximum inductance value for each method is used to normalize for all 50 data points, as seen in Fig. 11. The loop-based method shows the same trend with the FastHenry extraction results while having a 2% average error. This error is from the fact that the loop-based model does not consider bundle-to-bundle mutual inductance. However, as shown in Fig. 11, the optimization engine still correctly finds the optimized solution without the bundle-to-bundle consideration. Once the optimization process is completed, these bundle-to-bundle mutual inductances can be recomputed for any selected solution to improve the accuracy of the extracted netlist. The optimization capability can also be shown through the electrically optimized layouts in the solution space (Fig. 10). The optimization algorithm tries to reduce the loop area from DC+ to DC- for all footprints, resulting in similar optimized inductance results among all footprints. Table II shows the thermal and electrical results for three selected layouts from the solution space (Fig. 12). As seen in this table, the loop inductance is about 16 nH for all layouts, while the layout with the largest footprint has the smallest temperature result.

V. Conclusion and future work

In this paper, a new loop-based technique for inductance extraction in power module layout has been demonstrated. The model has been verified against FastHenry and shown to have similar accuracy with orders of magnitude faster runtime. Hence, the model is attractive for use with optimization routines thanks to its fast computation time and accuracy versus the state-of-the-art method. Additionally, the stability of the model has also been verified against FastHenry. The stability analysis has proven that the model is suitable for the layout optimization routine.

Although the model has shown good accuracy and performance for 3D parasitic extraction, more work needs to be done to improve the bundle creation algorithms. The model also needs to consider vertical electrical parasitics elements from vias, solder ball arrays, etc. Moreover, with the current implementation, this model does not consider large planar traces, which is quite common in MCPM design. Future works will consider a hybrid technique that combines PEEC and loop-based methods to handle planar traces. Moreover, since this model applies the divide-and-conquer technique to calculate the parasitic inductance of disconnected bundles, it can be further sped up using parallel programming techniques. This model will be integrated into PowerSynth 2 to enhance parasitic extraction performance for complicated 2.5D-3D layouts.

References

[1] H. A. Mantooth, T. Evans, C. Farnell, Q. Le, and R. Murphree, “Emerging Trends in Silicon Carbide Power Electronics Design,” *CPSS Transactions on Power Electronics and Applications*, vol. 2, no. 3, pp. 161–169, 2017.

[2] E. A. Jones, F. F. Wang, and D. Costinett, “Review of Commercial GaN Power Devices and GaN-Based Converter Design Challenges,” *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 4, no. 3, pp. 707–719, 2016.

[3] C. Chen, F. Luo, and Y. Kang, “A Review of SiC Power Module Packaging: Layout, Material System and Integration,” *CPSS Transactions on Power Electronics and Applications*, vol. 2, no. 3, pp. 170–186, 2017.

[4] T. M. Evans, Q. Le, S. Mukherjee, I. Al Razi, T. Vrotsos, Y. Peng, and H. A. Mantooth, “PowerSynth: A Power Module Layout Generation Tool,” *IEEE Transactions on Power Electronics*, vol. 34, no. 6, pp. 5063–5078, 2019.

[5] I. Al Razi, Q. Le, T. M. Evans, S. Mukherjee, H. A. Mantooth, and Y. Peng, “PowerSynth Design Automation Flow for Hierarchical and Heterogeneous 2.5-D Multichip Power Modules,” *IEEE Transactions on Power Electronics*, vol. 36, no. 8, pp. 8919–8933, 2021.

[6] P. Ning, F. Wang, and K. D. Ngo, “Automatic layout design for power module,” *IEEE Transactions on Power Electronics*, vol. 28, no. 1, pp. 481–487, 2013.

[7] I. A. Razi, Q. Le, H. A. Mantooth, and Y. Peng, “Physical Design Automation for High-Density 3D Power Module Layout Synthesis and Optimization,” in *IEEE Energy Conversion Congress and Exposition*, 2020, pp. 1984–1991.

[8] T. M. Evans, S. Mukherjee, Y. Peng, and H. A. Mantooth, “Electronic Design Automation (EDA) Tools and Considerations for Electro-Thermo-Mechanical Co-Design of High Voltage Power Modules,” in *IEEE Energy Conversion Congress and Exposition*, 2020, pp. 5046–5052.

[9] Q. Le, T. Evans, Y. Peng, and H. A. Mantooth, “PEEC Method and Hierarchical Approach Towards 3D Multichip Power Module (MCPM) Layout Optimization,” in *IEEE International Workshop on Integrated Power Packaging*, 2019, pp. 131–136.

[10] C. Zhao, L. Wang, F. Zhang, and F. Yang, “A Method to Balance Dynamic Current of Paralleled SiC MOSFETs With Kelvin Connection Based on Response Surface Model and Nonlinear Optimization,” *IEEE Transactions on Power Electronics*, vol. 36, no. 2, pp. 2068–2079, 2021.

[11] Q. Le, T. Evans, S. Mukherjee, Y. Peng, T. Vrotsos, and H. A. Mantooth, “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*, 2017, pp. 327–334.

[12] L. Wang, Z. Zeng, Y. Yu, K. Ou, and J. Wang, “Current-Bunch: A Fast and Accurate Tool to Extract and Optimize Parasitics of Power Packaging,” in *IEEE Energy Conversion Congress and Exposition*, 2020, pp. 3171–3177.

[13] L. Horowitz, N. Pallo, S. Coday, and R. Pilawa-Podgurski, “A Method of Partial Inductances to Evaluate and Optimize Switching Cells,” in *IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2021.

[14] R. Escovar, S. Ortiz, and R. Suaya, “An Improved Long Distance Treatment for Mutual Inductance,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 5, pp. 783–793, 2005.

[15] Y. Peng, D. Petranovic and S. K. Lim, "Chip/package Co-analysis and Inductance Extraction for Fan-out Wafer-level-packaging," *2017 IEEE 26th Conference on Electrical Performance of Electronic Packaging and Systems*, pp. 1-3, 2017.

[16] X. Huang, P. Restle, T. Bucelot, Y. Cao, T.-J. King, and C. Hu, “Loop-based Interconnect Modeling and Optimization Approach for Multigigahertz Clock Network Design,” *IEEE Journal of Solid-State Circuits*.

[17] S. Yu, D. M. Petranovic, S. Krishnan, Kwyro Lee and C. Y. Yang, "Loop-based Inductance Extraction and Modeling for Multiconductor on-chip Interconnects," in *IEEE Transactions on Electron Devices*, vol. 53, no. 1, pp. 135-145, Jan. 2006

[18] R. Kumar, S. C. Rustagi, Kai Kang, K. Mouthaan and T. K. S. Wong, "Characterization and Modeling of CMOS On-chip Coupled Interconnects," *ESSDERC - 37th European Solid State Device Research Conference*, 2007.

[19] M. Kamon, M. Tsuk, and J. White, “Fasthenry: A Multipole-accelerated3-d Inductance Extraction Program,”IEEE Transactions on MicrowaveTheory and Techniques, vol. 42, no. 9, pp. 1750–1758, 1994.

[20] C. Hoer and C. Love, “Exact Inductance Equations for Rectangular Conductors with Applications to More Complicated Geometries,” *Journal of Research of the National Bureau of Standards, Section C: Engineering and Instrumentation*, p. 127, 1965.