12-2022

Electrical Modeling for Dynamic Performance Prediction and Optimization of MCPMs Layout

Quang Minh Le
University of Arkansas, Fayetteville

Electrical Modeling for Dynamic Performance Prediction and Optimization of MCPMs Layout

A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Engineering with a Concentration in Electrical
Engineering

by

Quang Minh Le
University of Arkansas
Bachelor of Science in Electrical Engineering, 2015

December 2022
University of Arkansas

This thesis is approved for recommendation to the Graduate Council

H. Alan Mantooth, Ph. D.
Chair

Yarui Peng, Ph. D.
Committee Member

Yue Zhao, Ph. D.
Committee Member

Wing Ning Li, Ph. D.
Committee Member

Abstract

In recent years, the fast development of Multichip Power Modules (MCPM) packaging and Wide Bandgap (WBG) technology has enabled higher voltage and current ratings, better thermal performance, lower parasitic parameters, and higher mechanical reliability. However, the design of the MCPM layout is a multidisciplinary problem leading to many time-consuming analyses and tedious design processes. Because of these challenges, the design automation tool for MCPM layout has become an emerging research area and gained much attention from the power electronics community. The two critical objectives of a design automation tool for MCPM layout are fast and accurate models for design insights and a feasible layout generation algorithm for quick transition to hardware fabrication. On this end, PowerSynth, the MCPM design automation tool, has met these two key factors and become one of the most mature tools in the design automation community. Along with a generic and constraint-aware layout generation algorithm, the PowerSynth model library allows fast design insights on thermal performance, electrical parasitic extraction, mechanical reliability, and design rules for partial discharge. This diversity in modeling and layout algorithms has opened many challenges to enhance the tool capabilities further.

The focus of this work is modeling and extraction of electrical parasitic parameters within the MCPM design automation scope. This topic has been previously approached by adopting microstrip models combined with Laplacian Matrix evaluation for fast electrical parasitic evaluation. However, the two critical limitations of this approach are inaccurate parasitic models for some Direct Bonded Copper (DBC) substrates and lack the mutual coupling consideration among different conducting paths. In the first part of this work, response surface modeling has been used to replace the inaccurate microstrip equations, while methodologies such as Partial

Element Equivalent Circuit (PEEC) and Loop-based methods are used to consider mutual coupling. This combination ensures fast and accurate design insights during the MCPM layout optimization process. In the previous studies, PowerSynth commonly produced electrically optimized layouts by reducing their power loop inductance. While this approach effectively reduces the overshoot voltage, it lacks crucial information about other dynamic performance criteria. Hence, in the second part of this work, a methodology has been developed to allow the optimization tool to assess a more electrically optimized layout in terms of overall dynamic performance.

Acknowledgement

If you are reading these lines, it means that I have finally made it. The Ph. D. journey has been the most challenging part of my life from which I have learned and grown a lot. First, I would like to give my deepest gratitude to my research advisor, Prof. H. Alan Mantooth, for providing invaluable guidance, financial support, and encouragement during this degree. Thanks for the motivations when I lost my direction in the research.

I also thank Dr. Yarui Peng for his guidance throughout the last four years. His critical thinking and experiences in the VLSI world greatly helped me in my research. I also want to thank my committee members Dr. Wing Ning Li, and Dr. Yue Zhao, for their support and knowledge. I also want to thank Tom Vrotsos, who led the PowerSynth team when I first joined. He is an optimistic person who always encourages the students during their research. Also, thanks to Dr. Jia Di, for introducing me to this excellent research opportunity.

I could not complete this research without the tremendous support of the PowerSynth team members: Imam Al Razi, Tristan Evans, Shilpi Mukherjee, and David Agogo. Special thanks to my colleagues and good friends: Md Maksudul Hossain and Robert Murphree, for motivating me throughout the Ph. D. I also want to thank all of my Vietnamese friends for their encouragement and motivation. To all other colleagues: Huong Tran, Hui Wang, Pengyu Lai, Sudharsan Chinnaiyan, Marvin Suggs, Pablo Alvarez, Sayan Seal, Amol Deshpande, Balaji Narayanasamy, and Arman Ur Rashid thanks for assisting me on this journey. You are the true friends, brothers, and sisters. I am lucky to meet and know you all!

I also want to acknowledge the inputs from scholars at other universities who visited the University of Arkansas during my Ph. D. Many thanks to Danny Lohan, Lorenzo Ceccarelli, and Andrea Stratta.

I am very appreciative for the encouragement and support from my supervisor at Cadence Design System, Patrick O' Halloran. I have learnt a lot through the 4 months internship at Cadence Design System Analog Mixed Signal group.

To Mom, Dad, and Sister, thanks for always being there, listening, and sharing your thoughts throughout my life. I cannot finish this journey without all your continuous motivation, support, and encouragement. No words can express my deep gratitude to my family. I love you all!

Finally, to my lovely girlfriend, Thuy. Thanks for walking into my life and caring for me during the dissertation writing period. I love you with all my heart!

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1 Background and Motivation

In recent years, advanced power electronics' progression with high efficiency, high reliability, and novel functionalities has enabled modern electrified applications such as electric vehicles, data centers, home appliances, and aerospace [1]–[3] . Power conversion and electrification play an essential role in developing a more sustainable and low-carbon energy future [4]. To achieve this goal, the researcher continues to develop new devices using Wide Bandgaps (WBG) materials such as SiC and GaN. These WBG devices have much higher breakdown voltages, faster-operating switching frequency, and higher operating temperature than their silicon counterparts [5]. Furthermore, the low on-resistance characteristic of the WBG device eliminates most of the power losses at the system level. Hence, the power electronic circuit designs using WBG devices are

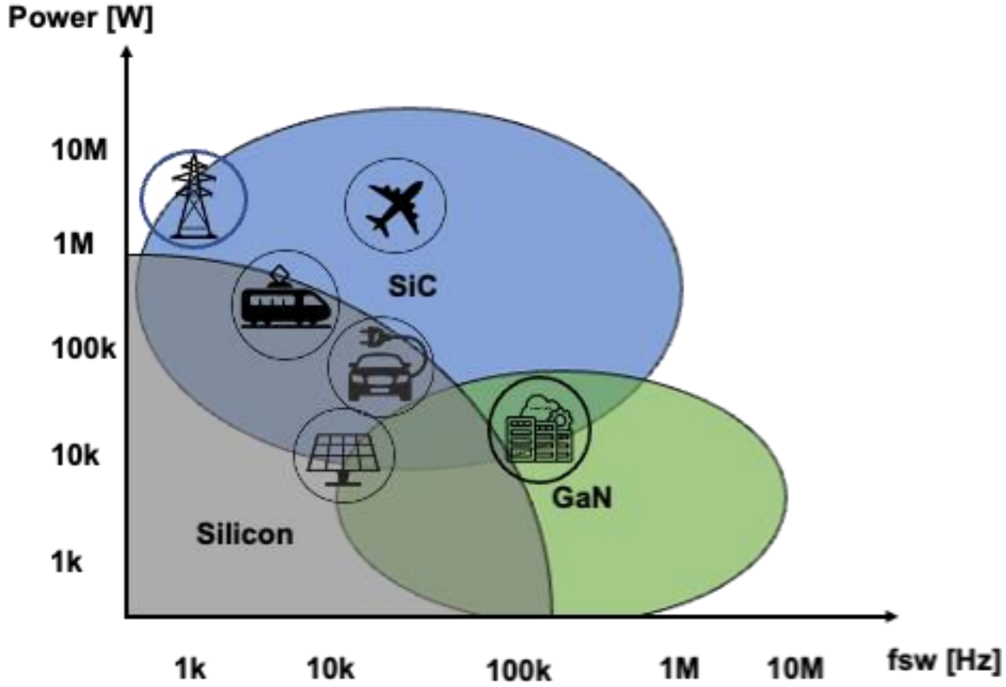


Fig. 1.1 Operating frequency versus power output for WBG devices and their application

more compact, efficient, and reliable than silicon-based devices. These characteristics have pushed the boundaries for power electronic applications to a much higher power at a faster switching speed (Fig. 1.1).

Packaging engineers often integrate multiple WBG devices into a single planar substrate such as Direct Bonded Copper (DBC) to form a Multichip Power Module (MCPM) and improve power density. Thanks to the higher operating temperature characteristic of WBG devices, this greatly reduces the thermal management effort, thereby reducing the size and weight of the system. At the same time, the faster-switching characteristic of WBG devices significantly reduces the sizes of passive components, which make it possible to integrate these passive devices inside the module and further increase the power density [6]. Therefore, along with the continued maturation of WBG technology, there are many research and development efforts on MCPM packaging strategies to

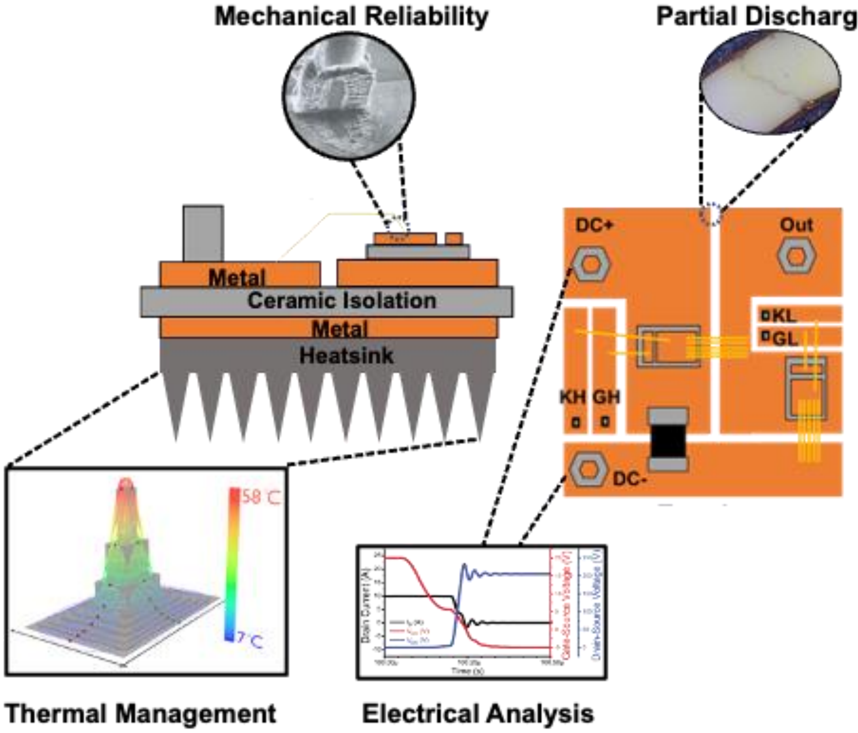


Fig. 1.2 Design consideration for a conventional MCPM layout

improve its reliability and compactness [7]. Besides the selection of materials and components during the design process, packaging engineers need to consider many design objectives to ensure the functionality and reliability of MCPM. Fig. 1.2 summarizes most of the design concerns for a MCPM which include electrical parasitic [8]–[10], thermal management [11], [12], mechanical reliability [13], and partial discharge [14].

Due to the multidisciplinary nature of MCPM design, packaging engineers usually need to perform many analyses during the design process, as illustrated in Fig. 1.3. Starting from the circuit specification (e.g., voltage, current ratings), the designer needs to perform a circuit simulation to develop the best control strategy for the design. Similarly, from the initial design specification (e.g., WBG device selection, layout footprint), an initial layout can be designed using 3D layout design and analysis tools. To begin with, the Finite Element Analysis (FEA) tool, such as Ansys Q3D, is run to extract electrical parasitic parameters from this initial layout. These parasitic

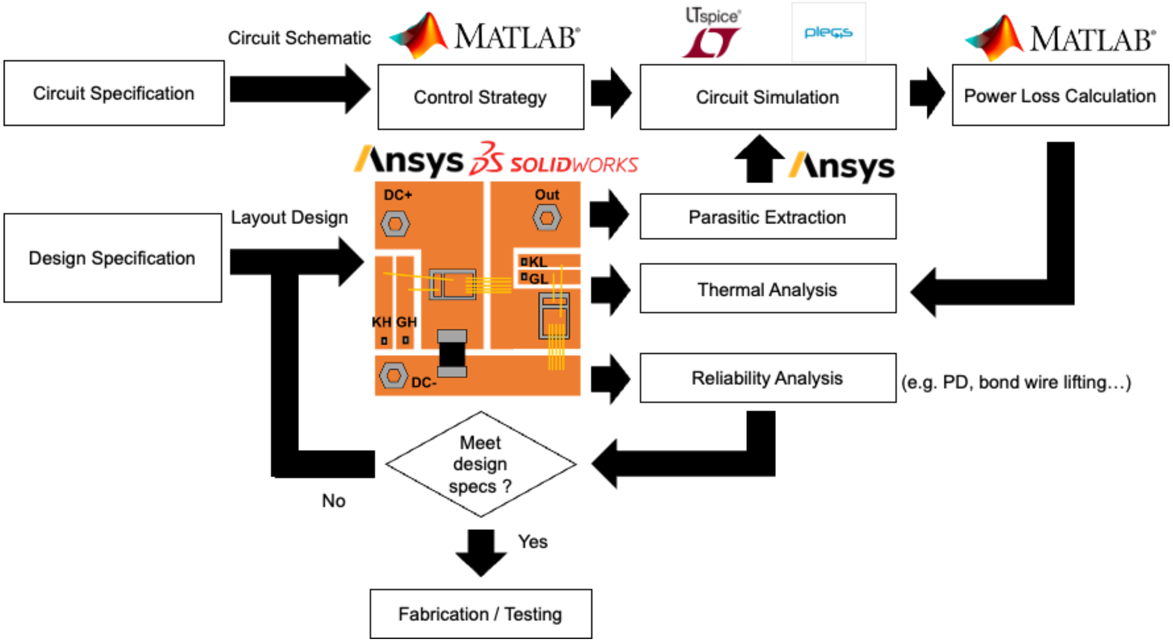


Fig. 1.3 The manual and traditional design process for MCPM

elements are inputs for the circuit simulation tools such as LtSpice for PLECS, where transient simulations are performed to predict the circuit performance. The transient current and voltage waveforms from this circuit simulation can be further analyzed to evaluate switching and conduction losses for each device. Finally, these power losses are fed into the FEA thermal simulation to evaluate the junction temperature of each device. Additionally, reliability analyses such as partial discharge or mechanical reliability are performed depending on the design specification. If any of the design specifications are not met, the designer must manually redraw the layout and run all the analyses again. This takes a lot of modifications and iterations until the design specifications are met. Hence, the total design time can take days or weeks. Finally, the designer can successfully fabricate and experimentally verify the physical design.

The MCPM packaging engineers need an automated design tool to close the gaps between design specification, packaging knowledge, and physical design fabrication. To this end, PowerCAD, a project started in 2010 [15] for an MCPM layout optimization tool, has been initiated. Since then, PowerCAD has transformed into a tool named PowerSynth which has been continuously developed [16]–[18]. A constraint-aware algorithm for the PowerSynth layout engine has been developed to ensure accurate Design Rule Check (DRC) for all layout solutions which

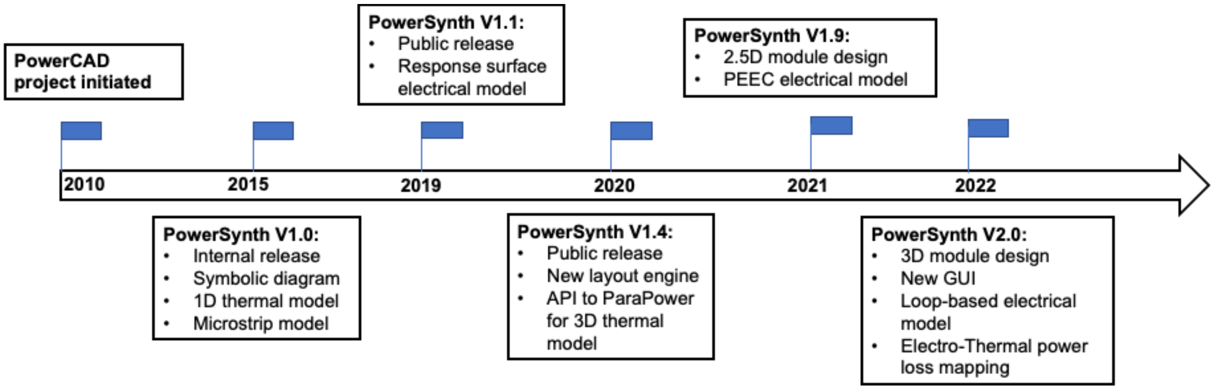


Fig. 1.4 PowerSynth releases and features.

has been demonstrated in [18], [19]. Along with this layout generation algorithm, a modeling library for reduced-order electrical, thermal, and mechanical models has been continuously developed [16], [20], [21] The combination of the layout engine algorithm and reduced-order model library makes it possible to perform 2D-2.5D-3D layout optimization inside the tool which generates hundreds to thousands of layout solutions in just a few hours. These models have been continuously updated and hardware validated against experimental measurements during this continuous maturation of PowerSynth (Fig. 1.4). To this date, this tool is one of the most mature MCPM layout design automation tool in the power electronic society. The newest version of the tool, PowerSynth v.2.0 has a complete software architecture combining the advantages from Linux operating system, Python programing language, layout engine algorithm, modeling library, Application Programing Interface (API) and so on (Fig. 1.5).

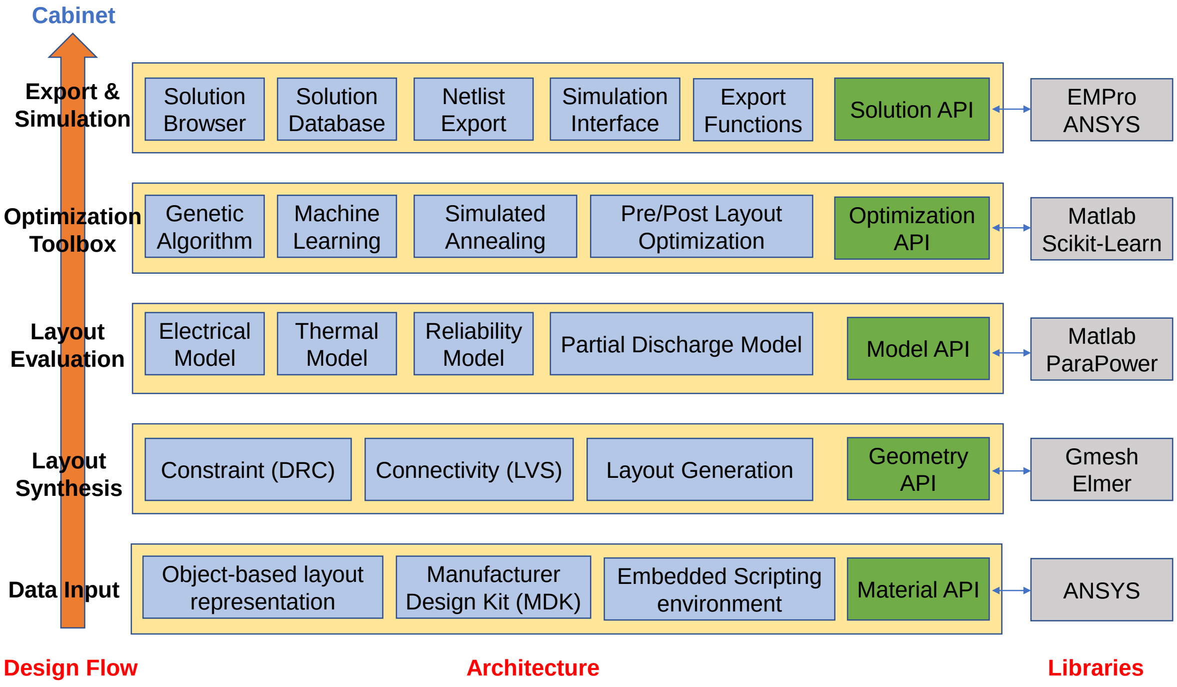


Fig. 1.5 Software Architecture for PowerSynth v2

The combination of layout algorithm, optimization algorithms, and modeling libraries makes the research on MCPM layout design automation a complex and challenging task. Each aspect plays a crucial role in pushing the boundaries of power density, reliability, and efficiency of the power electronic system. Among these aspects, electrical design is the most crucial target, and it is a multidisciplinary problem in and of itself. This electrical design aspect requires the designer to thoroughly understand electromagnetic behavior, device physics, circuit operation, and their interaction. Therefore, this work highlights the development and modeling effort for parasitic extraction in the MCPM layout optimization scope. Different modeling techniques have been investigated throughout the development of the PowerSynth tool. A comparison among the advantages and disadvantages in terms of accuracy, extraction time, and extracted netlist usability for these modeling techniques are also presented.

The extracted parasitic parameters are used for further studies on the dynamic performance of the layout. An investigation on each parasitic parameter on the overall dynamic performance of the circuit through simulation studies and literature reviews is presented in this dissertation. This work also develops and incorporates a WBG physic-based device, power loss estimation, and circuit stability models in the tool. Furthermore, this work links the outputs from PowerSynth electrical and thermal models to accurately assess the circuit performance, stability, and Safe Operation Area (SOA). The model developed from this work can achieve better design insights, which results in an overall electro-thermal optimized layout for given circuit design specifications.

1.2 Literature Review

A. Previous Modeling Methodologies for Parasitic Extraction in PowerSynth

The first step in understanding the Multichip Power Module’s (MCPM) parasitic model's electromagnetic behavior is performing Finite Element Analysis (FEA) on the structure. However, running FEA analyses inside a layout optimization routine could be very computationally expensive. As described in the previous Power-CAD project, placing FEA simulations in the optimization loop can take weeks to finish [15]. Hence, FEA simulations are only suitable for studying the MCPM structure and developing an analytical model for fast and accurate parasitic extraction. This study has been done extensively in previous work, such as [22], searching for the best modeling strategies for PowerSynth's parasitic extraction. In this previous attempt, the author has found that the microstrip self-impedance trace model is very similar to a trace model on the Double Bonded Copper (DBC) substrate. The author also ignores the effect of mutual inductance and claims that it is insignificant in some MCPM layouts. This is not entirely true and later discussed in Chapters 2 and 3 of this dissertation.

The work in [23] implements a Laplacian-Matrix model [24] to efficiently evaluate the equivalent impedance network in PowerSynth v1.0. To begin with, this work uses the symbolic layout concept which is widely used in Very Large-Scale Integration (VLSI) to represent a MCPM layout. This symbolic layout representation is later converted into a undirected graph where each node represents a device or lead connection, and each edge stores the trace parasitic resistance and inductance calculated using the model developed in [22]. However, there are some issues with this modeling approach. First, due to the geometrical constraints in most microstrip models, the

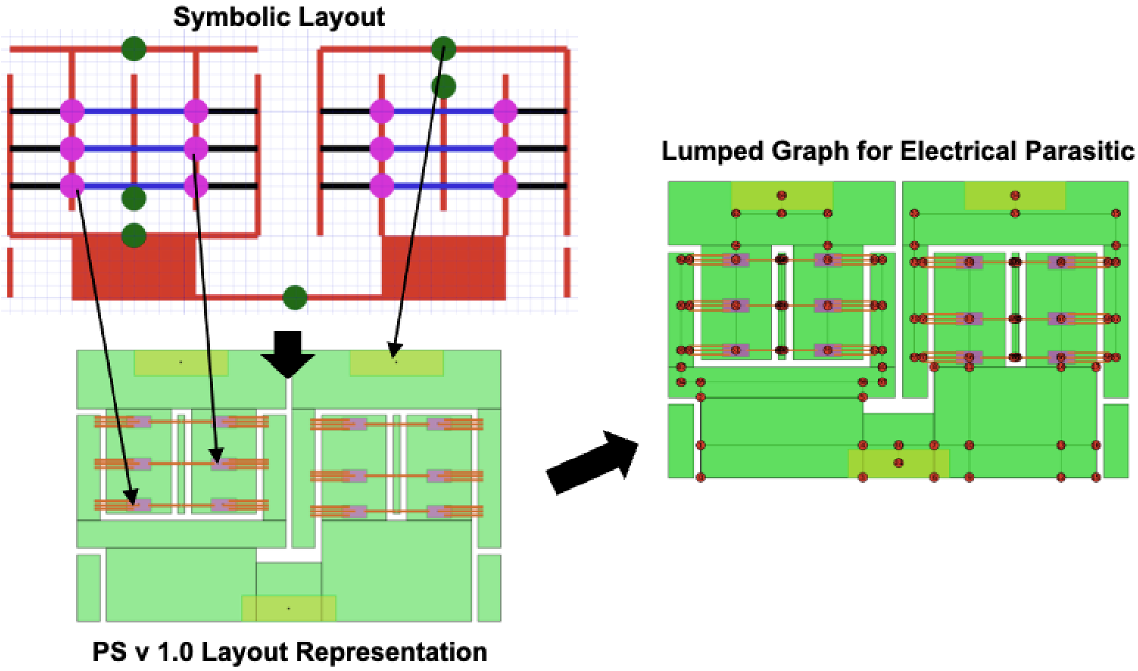

Symbolic Layout
Lumped Graph for Electrical Parasitic
PS v 1.0 Layout Representation

Fig. 1.6 PowerSynth v1 Symbolic layout representation

parasitic inductance result is inaccurate for many cases where the trace width is large which is common for MCPM layout. Second, the microstrip models are designed for antenna and high-speed Printed Circuit Board (PCB) applications which represents parasitic results at GHz range instead of few hundreds kHz to MHz range in MCPM. Finally, the Laplacian-Matrix model treats the parasitic inductance network as a resistance network. This violates the loop concept of inductance calculation since the mutual elements are ignored. However, in some layout cases, this shows quite good parasitic loop approximation. This is because the mutual inductance impact is small in these layouts since the trace-to-trace distance is quite far. One such example is demonstrated in [25] where a generated layout by PowerSynth v.1.0 has been fabricated and hardware validated (Fig. 1.7). Here, the response surface model in [16] is combined with the Laplacian Matrix method to overcome the inaccuracy from the microstrip approach. While this

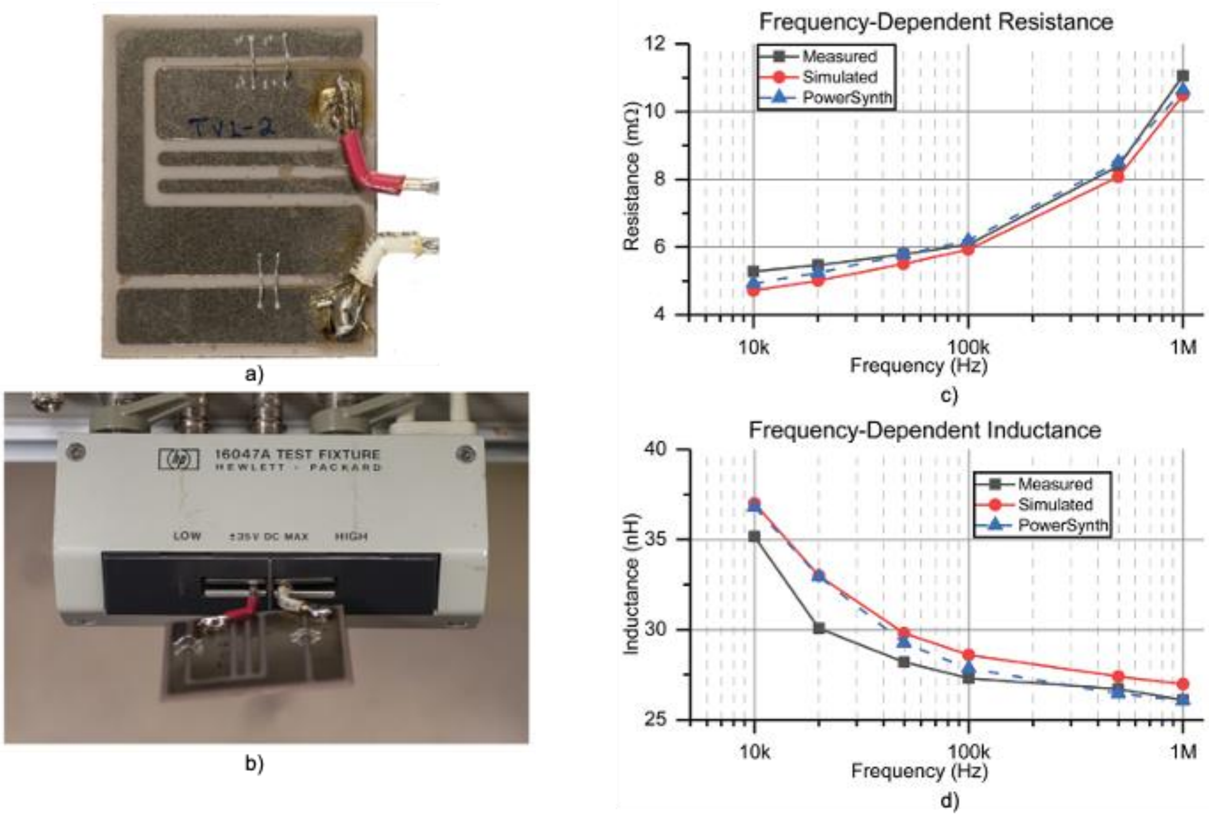
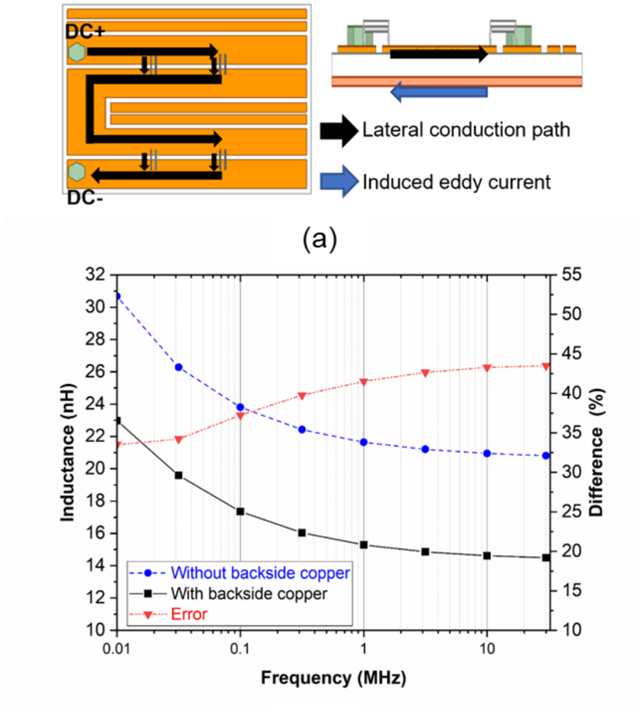


Fig. 1.7 (a) Fabricated layout (b) Impedance analyzer (c) Resistance (d) Inductance

combination is not quite correct for all layout cases, it has demonstrated that reduced-order methodology can be applied to speed up the design and analysis process.

B. Methodologies for Parasitic Extraction in MCPMs Analysis and Design Automation



(a)

(b)

Fig. 1.8 (a) lateral versus eddy current paths (b) eddy current impact on MCPM layout

In the first part of this dissertation, a new impedance model is first developed to overcome the issues from the previous methodologies used in PowerSynth. The backside copper impact of the Direct Bonded Copper (DBC) substrate must be modeled to improve the accuracy of the impedance model. As seen in Fig. 1.8 (a), for a MCPM built on a DBC substrate, there are two main current paths: the lateral and eddy current paths. To have an accurate extraction for loop-impedance in MCPM layout, modeling approaches for both current paths must be considered. The lateral current path requires an accurate method to consider mutual inductance and proximity effect among the conductor traces. As for the eddy current path, as seen in Fig. 1.8 (b), where a parasitic extraction simulation has been performed in ANSYS Q3D, by simply remove the backside copper, there is up to 44% difference versus the results with the backside. On this end, there have been many studies on impedance models which consider eddy current effects in other electrical applications. One such example, which has been used in PowerSynth v1.0 is the closed-form analytical equations for the parasitic inductances of the microstrip structures. These microstrip models are commonly used for designing Printed Circuit Board (PCB) for antenna application [26]–[28]. The issues with these equations are that they are designed for much smaller trace width, conductor thickness, and very high frequency application (GHz) range. At this high-frequency range these equations also assume zero thickness conductor due to the sub-micron thickness of the skin-depth value for antenna application. Moreover, some microstrip equations also require complex integral or numerical computation, which is not computationally attractive for design automation application [27]. To overcome the dimensional constraints of small trace widths, the work in [29] use a 2-D Partial Element Equivalent Circuit (PEEC) technique with microstrip equations for extraction of CMOS monolithic inductors and transformers design. Accurate extraction can be achieved using this method for a high-frequency range (100 MHz – 14 GHz). However, the method also requires numerical integration for self and mutual impedances which is quite computationally expensive.

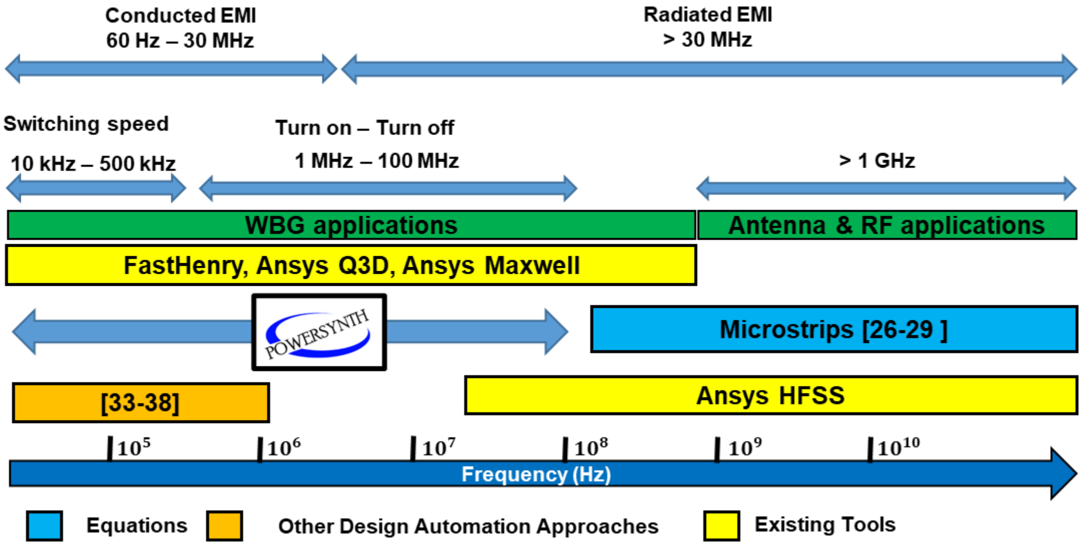


Fig. 1.9 Different frequency range for parasitic extraction tools

Because MCPM layouts are mostly designed for WBG devices, these devices' turn-on and turn-off rates dictate the concerned frequency range for MCPM parasitic extraction. As shown in the literature [5], [7], [12], [30], these turn-on and turn-off rates are about ~27-50 ns for SiC and ~5-10 ns for GaN devices, which limit the frequency range for inductance extraction in an MCPM layout to about 10 kHz to 30 MHz range (Fig. 1.9). In this frequency range, it is also possible to perform more analyses on the Electromagnetic Interference (EMI) issue [31]. There are several well-known quasistatic approximation techniques for parasitic extraction in this frequency range. These include Method of Moments (MoM) and PEEC [32], which have been used in commercial state-of-the-art (SOTA) analysis tools such as Ansys, COMSOL, or the high fidelity open-source analysis tool such as FastHenry [33]. These methods are quite accurate and applicable for broadband frequency extraction; however, they are usually computationally expensive, making them unfavorable for the optimization routine. In addition, it is hard to extract a lumped netlist suitable for circuit simulation using the existing tools.

Within the MCPM design automation scope, the abovementioned well-known methodologies are often modified and simplified to accelerate the extraction time. Additionally, some assumptions have been made to maintain acceptable extraction accuracy [34]–[37]. However, these methodologies usually ignore the induced eddy current effect, which can result in up to 50% of inaccuracy. For instance, the work [38] in applied MoM method as a fitness function in a genetic algorithm optimization routine for MCPM layout. This method first discretizes the layout into many smaller uniform square elements and then solves for the overall DC current path. Then the loop inductance has been solved using the DC current information. While this work claims to have a good agreement with state-of-the-art methodologies there are two key draw backs. First, to have a correct extraction for the lateral loops, the edge size of each mesh element must be close to the skin-depth value, which in turn increases the overall number of mesh elements. Second, this method does not consider the eddy-current impact; hence gives overestimated parasitic loop results. Other methodologies for parasitic extraction in design automation such as [35]–[37] use partial element approach to reduce the required total mesh elements. This can be done by discretizing the layout into long and thin segments. The self-impedance value of each segment is first evaluated through existing analytical equations. Then circuit technique such as Modified Nodal Analysis (MNA) can be used to evaluate the overall loop parasitic value. One such example of this technique is the work in [35] where a tool named Current-Bunch has been developed. In this tool, each segment width is set to be close to the skin-depth value at the extraction frequency. This allows the tool to accurately capture the proximity effect in the lateral conduction path. However, similar to the work in [39], the extracted loop parasitic result might be overestimated because the eddy-current effect is not considered. Table 1.1 below summarizes several development efforts for parasitic extraction within the design automation scope.

Table 1.1 Comparison among different extraction methods for MCPM layout optimization

Reference	***Eddy current***	***Extraction time***	***Accuracy***	***Flexibility***	***Distributed netlist***	***Broadband***
PEEC [36]	Yes	Not Mentioned	Very High	High	Hard to extract	Yes
Ansys Q3D	Yes	Slow	Very High	Very High	Hard to extract	Yes
FastHenry [33]	Yes	Average	Very High	Very High	Hard to extract	Yes
Horowitz [37]	Yes	Not Mentioned	Acceptable	High	N/A	Yes
Ning [38]	No	Fast	N/A	High	N/A	No
Current Bunch [35]	No	Fast	High	High	Yes (R, L and M)	No
PowerSynth-1.1 [25]	Yes	Very Fast	Acceptable	Low	Yes (only R, L)	Yes
PowerSynth-1.9 [17]	Yes	Average	High	High	Yes (R, L, and M)	Yes
Loop-based [20]	Yes	Fast	High	High	Yes (R, L, and M)	Yes

C. Parasitic Inductance Impacts in MCPM Layout Design

Most of the abovementioned works focus on reducing the overall DC+ to DC- parasitic loop inductance value. This is because high DC+ to DC- parasitic inductance can result in high voltage overshoot between the drain and source of the device which in turn causes reliability issues and reduce the overall system performance [39], [40]. However, as shown in the literature there are many other parasitic issues that affect the reliability of the circuit. These issues are summarized in Table 1.2 below. For example, the work in [8], [41] have shown that imbalance source side parasitic inductance can lead to unbalance dynamic current among the parallel MOSFET devices which result in unbalance switching losses among the devices. This is because a higher source inductance value results in a higher induced voltage at the source net of the device which in turn reduce the gate-to-source value V_{gs}. During the turn-on transient of the module, the gate-source voltage of the device with a smaller source side inductance among the parallel devices on the same switching leg will hit the threshold voltage (Vth) first. Similarly, during the turn-off transient, the device with a largest source side inductance will turn off last. This results in an imbalance current distribution among the devices during the dynamic turn-on and turn-off transitions. Furthermore, due to this dynamic imbalance there is up to 50% difference in switching losses among the parallel devices. This in turn results in imbalance total power loss leading to different maximum steady state temperature and different dynamic temperature changes among the parallel devices [42], [43]. Another crucial loop that must be considered during the layout design is the gate loop parasitic inductance value. The voltage feedback from the power loop during device turn-off state couples with the input parasitic capacitance of the MOSFET resulting in an oscillating gate current. A voltage drop between the gate resistor and parasitic inductance causes a gate-to-source voltage oscillation in the gate loop during the turn-off state. This gate-to-source voltage oscillation can cause false-turn on [44], [45] and self-sustained oscillation [46] issues. This might result in short-circuit condition which affects the reliability of the circuit and the overall system. The dynamic problem from electrical parasitic is a very broad research topic. In this dissertation, only the power loss aspect is considered. This aspect has been selected because it creates a link between the thermal and electrical modeling efforts in PowerSynth.

Table 1.2 Summary of parasitic impacts on circuit reliability and performance.

Reference	Parasitic Components	Definition	Issues
[39], [40]	L_{loop}	DC+ to DC- loop inductance	Voltage oscillation, Reduce switching performance
[8], [40]	L_{kelvin}, L_{source}	Kelvin and Source inductance	Imbalance source current, imbalance switching losses
[42], [43]	L_{kelvin}, L_{source}	Kelvin and Source inductance	Imbalance switching losses leading to imbalance temperature among parallel devices
[44]-[46]	R_g, L_g, C_{gs}	Gate Resistance, Gate loop inductance, device's gate-source capacitance	False turn on, Self-sustained oscillation

1.3 Research Objectives and Problem Definition

This dissertation aims to find the most suitable solution for parasitic extraction and electrical modeling with application in the Multichip Power Module (MCPM) layout optimization. While the layout algorithm can support a generic set of geometries, the main goal of PowerSynth is to evaluate and minimize the loop inductance of switching cells or half-bridge circuits, which are the smallest circuit components in many power electronics systems. Here, one of the most important objectives is the incorporation of the mutual inductance effect, which was not previously considered in the first version of PowerSynth.

The second objective is to balance the accuracy and speed of the parasitic extraction model to be fast enough in an optimization routine while maintaining acceptable accuracy. The third objective is to combine results from the electrical and thermal models with the device physics to have more insights into the dynamic performance of the circuit. The optimization algorithm can find a better trade-off between electrical and thermal aspects through this combination.

1.4 Key Contributions

Since the PowerSynth project requires both development and research skills, the technical and software development contributions are listed here:

A. Technical contributions

The main technical contribution of this work is developing a fast and accurate electrical parasitic extraction engine. This new electrical model improves the extraction accuracy by considering both self and mutual inductance as opposed to the self-inductance-only model in the previous work. To consider the mutual inductance impact, the Partial Element Equivalent Circuit (PEEC) method and Loop-based method from VLSI have been investigated, implemented, and modified for the MCPM layout optimization. Furthermore, the eddy-current effect has been considered through regression-based model which can be generated through both batch simulation and analytical approach. This eddy-current effect has up to 40% impact on the accuracy of the extraction in MCPM layout, but has not been considered in other MCPM layout optimization work. The hierarchical meshing approach developed in the PowerSynth-PEEC model improves the computational efficiency while maintaining similar extraction accuracy. It is worth mentioning that this implementation has allowed the layout engine and optimization algorithm to search for a more accurate solution space, which is further described in the results section. Additionally, the netlist extraction method developed in this work allows an efficient post-layout circuit simulation. This has been experimentally verified through fabricated MCPM layout and Double Pulse Test (DPT).

Investigation and development of the loop-based method has shown that this model can be used for both simple 2D-2.5D structures possibly 3D structures in future work. The method has been experimentally verified through impedance measurement of a fabricated 2D MCPM layout. Ongoing research and verification are required to ensure the model is robust and efficient for 3D MCPMs.

To incorporate the electrical parasitic impact on the device power loss, the physics-based model from the MSCAD group has been reimplemented in Python. Using the thermal-dependent equations and physical parameters from this model, a more correct power loss estimation can be achieved. During this study, a collaboration between the software development and modeling group has developed a device optimization tool for the MSCAD device model. Using the multiprocessing and multi-objective optimization strategy, this tool improves the C-V parameter extraction time from an hour to less than a minute. While the I-V parameter extraction is currently investigated, it does help the modeling engineer to have good initial fitting results. This collaboration and development link the parasitic and temperature parameters to perform a true electro-thermal optimization in PowerSynth.

This dissertation also demonstrates an electrothermal co-simulation strategy to consider the impacts of the parasitic parameters on switching losses. Here, a successive approximation method has been demonstrated. This method allows a quick final steady-state assessment of the layout for a given power loss – temperature dependent. Using this strategy, a more meaningful trade-off among electrical parasitic, maximum temperature and circuit operating conditions can be considered.

B. Software Development Contributions

Since the start of this project, a lot of the contributions have been made to improve the functionality, usability, stability, user friendliness, and most importantly the popularity of the PowerSynth tool. One of the biggest milestones has been the realization of the Power Electronics community in the MCPM layout optimization problem. The tool has won first place in the "Software Demo Contest" in ECCE 2021 conference which is one of the most popular conferences in the Power Electronic society. To achieve this goal, the contribution on the PowerSynth development are as follows:

- Maintaining, bugs fixing, and software releases work from version 1.1 to 2.0 of the tool.
- Trained REU students to develop and formalize the software packaging process.
- Design and redesign of the Graphical User Interface (GUI) for PowerSynth v 1.1-1.9.
- Initial development for both electrical and thermal application programming interface (Corner_Stitch_API). This links the layout engine and optimization to their cost-functions.
- Investigation, development and testing of different optimization algorithms, where the Matlab-API has also been developed.
- Export features to Ansys Q3D, FastHenry, and Solidworks for verification.
- Develop a multiprocessing batch simulation procedure with FastHenry API for verification purpose and 3D extraction.
- Initial ideas, development and maintaining of the Command Line Interface (CLI) flow. This CLI makes it much easier for development and debugging purposes. This CLI later translates into the GUI version of PowerSynth 2.0 using the same flow.
- Initial implementation, development, and train REU students to develop the material library and the layer stack data structure

- Redesign the technology files for components, leads, bondwires, and vias.

C. Software Contributions from Other Students

Other students in the PowerSynth group have also shared a huge load of software development work. These include:

Imam Al Razi has helped with most development work on and off hours. He initially designed and invented the layout script for the layout engine and contributed a lot to the tool's GUI and CLI development. He continuously helps to maintain, debug, and package the tools throughout PowerSynth v 1.3 – 2.0. Imam also contributes a lot in the thermal modeling aspect such as ParaPower and electrical modeling aspects such as FastHenry and ANSYS EM APIs. Most importantly, thanks to his guidance for Joshua Mitchener, an REU student in our group, the initial GUI for PowerSynth 2.0 has been made and demonstrated.

Tristan Evans has initially helped to fabricate, and hardware validated the module in PowerSynth V1. He also helped with the EMI modeling aspect and the initial development of the ParaPower API. He initially developed the PowerSynth-EMPro API, which allows studies on layout impacts on electromagnetic interference (EMI). He initially created the PowerSynth-ParaPower API, which allows thermal management studies of more complex MCPM layouts. His initial ideas of the PowerSynth-Solution data structure made an excellent contribution to other APIs such as the ANSYS EM or the Electrical API itself. His implementation has been demonstrated in PowerSynth release version v1.4.

Shilpi Mukherjee has initially helped with debugging, and software packaging of PowerSynth V1.1. While most of her works focus on the experimental aspects of the partial discharge modeling, she always helps to ensure the software package is stable for each PowerSynth release.

REU Students: Johnathan Main’s honor thesis work has introduced the Manufacturer’s Design Kit to PowerSynth. Yugo Isogai has helped with the Material Library. Joshua Mitchener has led the GUI development of PowerSynth 2.0.

Previous Students: Brett Shook was the lead developer for PowerSynth V1.1. Brett Shook and Zihao Gong contributed on most of the development and modeling of PowerSynth V1.1 and older. Andalib Nizam contributes to PowerSynth V1.1 technology library GUI design.

1.5 Overview of the Dissertation

The main contents of this dissertation are as followed:

- Chapter 2: Development of Trace Parasitic Model considering Eddy Current Losses in MCPM
- Chapter 3: Layout Engine to Electrical Evaluation Programming Application Interface (API)
- Chapter 4: Methodologies for Parasitic Extraction in MCPM Layout Optimization
- Chapter 5: Experimental and Optimization Results for Parasitic Extraction
- Chapter 6: Dynamic Performance Optimization for Co-Electrical-Thermal Management Design
- Chapter 7: Conclusion and Future Work

Chapter 2. Partial Self and Mutual Equations for MCPM Trace

This chapter first overviews the basic concept of loop versus self-inductance through Maxwell equations. From here, the chapter discusses the impact of backside copper in Direct Bonded Copper (DBC) substrate, which is usually used for Multichip Power Module (MCPM) fabrication. The chapter also describes the modeling approach using regression methods and response surface modeling to characterize accurate models for MCPM traces. While different simulation tools can be used for this model characterization process, the chapter also provides a partial element approach that is more robust and easier to implement. The contents of the sections are as follows:

Section 2.1 reviews the fundamental background of inductance calculations through Maxwell equations. This section also describes the self-inductance concept and partial elements approach for quasistatic approximation.

Section 2.2 reviews some of the general equations for inductance calculations. These equations are crucial for the formulation of the models in the next sections. Here, multithreading and multiprocessing approaches are investigated to ensure the most efficient inductance calculation. An analysis using different programming techniques is performed to find the best evaluation strategy.

Section 2.3 shows the importance of backside copper consideration in MCPM and its impacts on extraction accuracy. Ansys Q3D simulation is run where eddy current density is measured. Analytical solution for eddy current density is compared versus the Finite Element Analysis (FEA).

Section 2.4 describes the step-by-step process to formulate the regression-based models for the trace inductances. The section also introduces the layer-stack concept for MCPM and how to create a trace model library from the vendor's datasheet.

2.1 The Basic Concept of Self and Mutual Inductance from Maxwell Equations

In Fig. 2.1, a time varying current is circulating inside a closed loop of a U-shaped conductor surrounded by dielectric material. From the Maxwell equation [47], this current generates an electric field **E** inside the conductor and a magnetic field **H** inside the dielectric. The time domain Maxwell equations describing this concept are as follows:

$$\nabla \times \boldsymbol{E} = \frac{\delta \boldsymbol{B}}{\delta t} \tag{2.1}$$

$$\nabla \times \boldsymbol{H} = \boldsymbol{J} + \frac{\delta \boldsymbol{D}}{\delta t} \tag{2.2}$$

$$\nabla . \boldsymbol{D} = \rho \tag{2.3}$$

$$\nabla . \boldsymbol{B} = 0 \tag{2.4}$$

where **E** is the electric field (V/m), **B** is the magnetic flux (T), **H** is the magnetic field (A/m), **D** is the electric displacement (C/m^2), **J** is the electric current density (A/m^2), ρ is the electric density (C/m^3).

The first two equations (2.1), Faraday's law, and (2.2), Ampere's law, describe the relationship among time varying electric field **E**, magnetic field **H**, magnetic flux, and current density **J**. The equations (2.3) and (2.4), also known as Gauss's law describe the relationship between magnetic

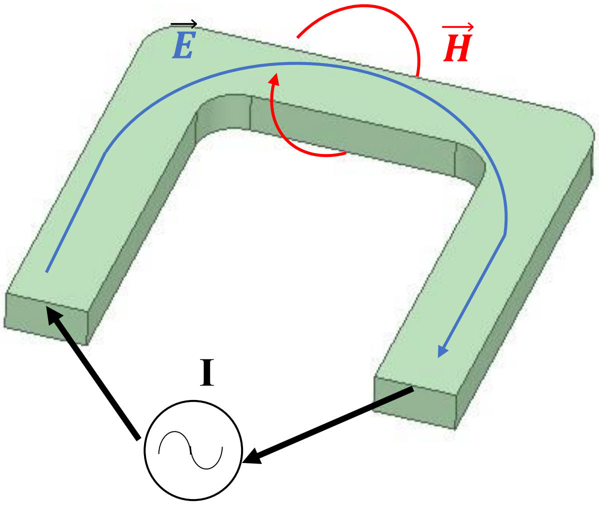


Fig. 2.1 A time varying current in a loop.

fluxes **B** and electric density. For the impedance calculation application, in the quasistatic domain, equations (2.1) and (2.2) are most commonly used to further derive and evaluate the loop value.

Deriving from the Maxwell's equations, Faraday's law [47], equation (2.1) is most crucial for the notion of loop inductance. With the given example in Fig. 2.1, the total magnetic flux penetrating the conductor loop can be evaluated by:

$$\boldsymbol{\phi} = \int_s \boldsymbol{B}.ds \tag{2.5}$$

where s is the surface area the current loop surrounds. From Faraday's law of induction, the total induced electromotive force around a loop contour can be calculated by:

$$emf = \oint_C \boldsymbol{E}.dl = -\frac{d\phi}{dt} \tag{2.6}$$

where c is the contour of the loop surrounding surface s. In the quasistatic domain where the loop is considered electrically small, this emf value can be represented as a voltage source where the voltage value is:

$$V = \frac{d\phi}{dt} \tag{2.7}$$

The loop inductance can be defined as the ratio between the magnetic flux and the current:

$$L = \frac{\phi}{I} \tag{2.8}$$

This inductance value is also known as self-inductance of the loop. equation (2.8) is only true if the surrounding medium is linear, homogenous and isotropic which is the general assumption for all inductance calculations in this work.

Fig. 2.2 illustrates the interaction between two different conductor loops. Given a time varying current in the first loop, the mutual inductance between two loops can be calculated by:

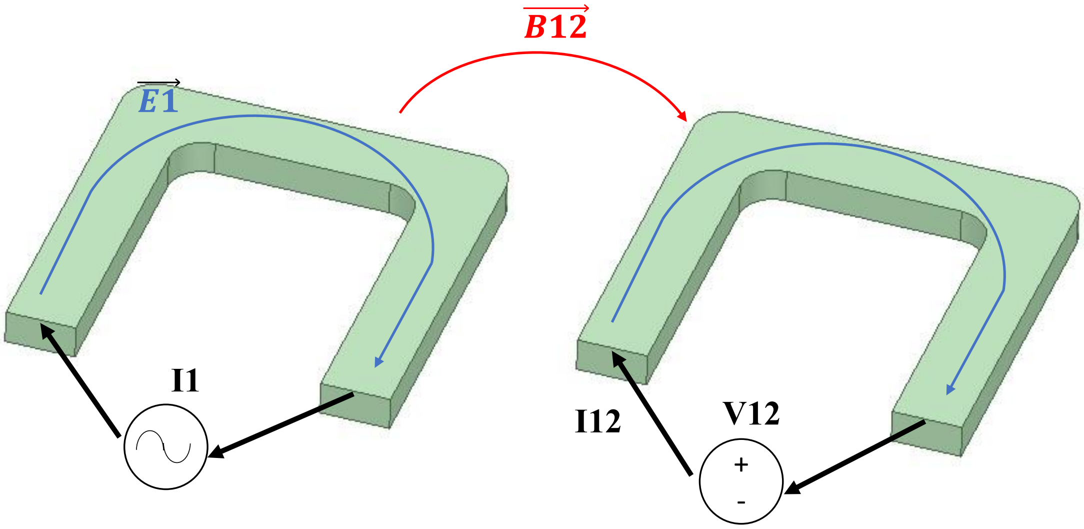


Fig. 2.2Mutual inductance between two loops

$$M_{12} = \frac{\phi 12}{I1} \tag{2.9}$$

where:

$$\phi_2 = \int_{s2} \boldsymbol{B_{12}}.\boldsymbol{ds} \tag{2.10}$$

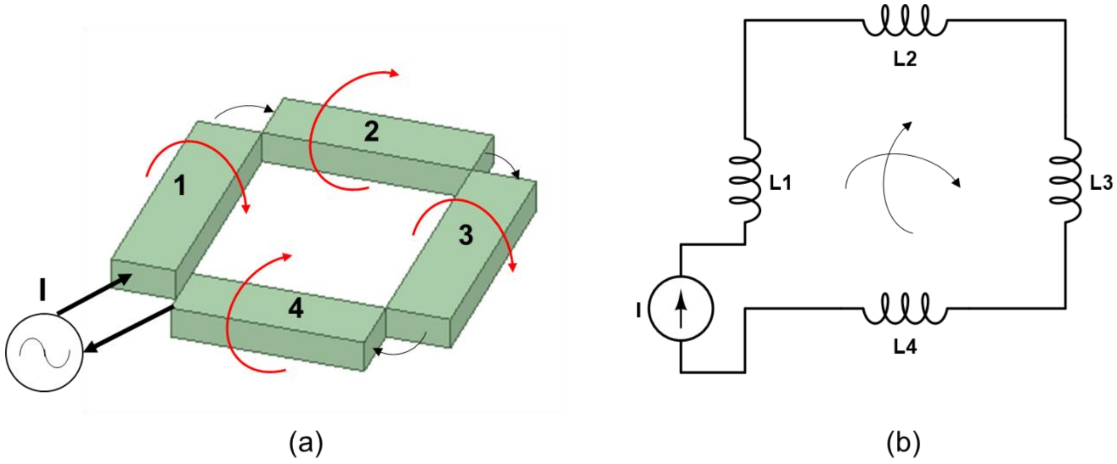
In many practical applications, a conductor loop can be cut into multiple smaller connected loops Fig. 2.3 (a). The circuit representation for this is demonstrated in Fig. 2.3 (b), where the total loop inductance for this specific example is calculated by equation (2.11).

$$L_{total} \ = \ L_1 + L_2 + L_3 + L_4 - \ M_{13} - M_{24} \tag{2.11}$$

The methodology to evaluate the overall loop using the segmented portions of the loop has been described for the first time in [32]. The L_i terms are defined as partial self-inductance and the M_{ij} terms are defined as partial mutual inductance. It is important to notice that these partial terms alone do not have any physical meanings but instead a convenient mathematical approximation to discretize the structure. More partial elements are usually needed in practical applications for a more accurate loop evaluation. These partial elements are fundamental to the Partial Element Equivalent Circuit (PEEC) method, which is discussed in more detail in Chapter 4.

2.2 Analytical Equations for Self and Mutual Inductance Calculation

Accurate models for the partial elements are needed to achieve an accurate loop solution for impedance extraction. These models can be used to further extract loop solution for a more



(a)

(b)

Fig. 2.3 (a) Multiple connected loops (b) Circuit representation

complicated structure. In this section, some of the general equations for inductance calculations are first discussed. Different methods for efficient inductance calculation are also analyzed.

The exact mathematical equations for partial self and partial mutual inductances of the rectangular bars structure in 3D space (Fig. 2.4) have been derived in [48] as follows:

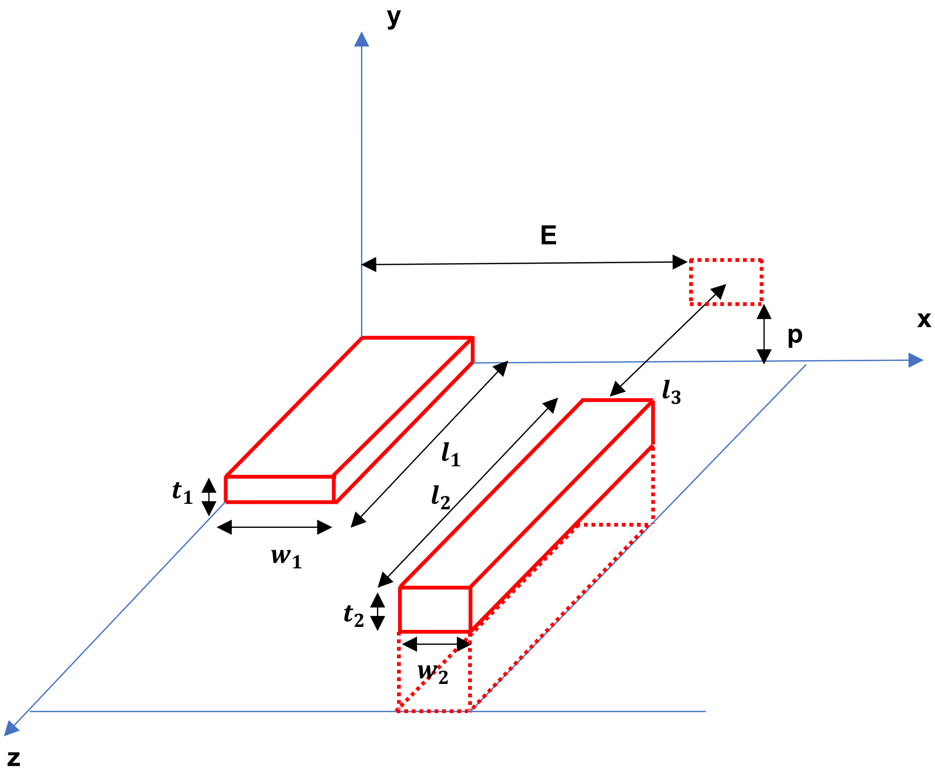


Fig. 2.4 Two parallel rectangular bars in 3D

$$M=\frac{1e^{-3}}{abcd}\Bigg[\Bigg[\Bigg[\left(\frac{y^2z^2}{4}-\frac{y^4}{24}-\frac{z^4}{24}\right)xln\left(\frac{x+\sqrt{x^2+y^2+z^2}}{\sqrt{y^2+z^2}}\right)+\left(\frac{x^2z^2}{4}-\frac{x^4}{24}-\frac{z^4}{24}\right)yln\left(\frac{y+\sqrt{x^2+y^2+z^2}}{\sqrt{z^2+z^2}}\right)+$$
$$\left(\frac{y^2x^2}{4}-\frac{x^4}{24}-\frac{y^4}{24}\right)zln\left(\frac{z+\sqrt{x^2+y^2+z^2}}{\sqrt{y^2+x^2}}\right)+\frac{1}{60}(x^4+y^4+z^4-3x^2y^2-3y^2z^2-$$
$$3x^2z^2)\sqrt{x^2+y^2+z^2}-\frac{xyz^3}{6}\tan^{-1}\frac{xy}{z\sqrt{x^2+y^2+z^2}}-\frac{xzy^3}{6}\tan^{-1}\frac{xz}{y\sqrt{x^2+y^2+z^2}}-$$
$$-\frac{yzx^3}{6}\tan^{-1}\frac{yz}{x\sqrt{x^2+y^2+z^2}}\Bigg]\begin{matrix}E-w_1,E+d\\(x)\\E+d-w_1,E\end{matrix}\Bigg]\begin{matrix}P-t_1,E+w_2\\(y)\\P+t_2-t_1,P\end{matrix}\Bigg]\begin{matrix}l_3-l_1,l_3+l_2\\(z)\\l_3+l_2-l_1,l_3\end{matrix} \tag{2.12}$$

Where $$\Bigg[\Bigg[\Bigg[f(x,y,z)\quad\Bigg]\begin{matrix}q_1,q_3\\(x)\\q_2,q_4\end{matrix}\Bigg]\begin{matrix}r_1,r_3\\(y)\\r_2,r_4\end{matrix}\Bigg]\begin{matrix}s_1,s_2\\(z)\\s_2,s_4\end{matrix}\equiv\sum_{i=1}^{4}\sum_{j=1}^{4}\sum_{k=1}^{4}(-1)^{i+j+k+1}f(q_i,r_i,s_i) \tag{2.13}$$

According to [48] the self-inductance is the special case for the mutual inductance of rectangular bars where $w_1 = w_2, t_1 = t_2, and\ E = p = l_3 = 0$. The self-inductance equations can be reduced to:

$$L = \frac{8e^{-3}}{a^2b^2}\left[\left[\left[f(x,y,z) \quad \right]\begin{matrix} w \\ (x) \\ 0 \end{matrix}\right]\begin{matrix} t \\ (y) \\ 0 \end{matrix}\right]\begin{matrix} l \\ (z) \\ 0 \end{matrix} \qquad (2.14)$$

$$\text{Where } \left[\left[\left[f(x,y,z) \quad \right]\begin{matrix} q_1 \\ (x) \\ q_2 \end{matrix}\right]\begin{matrix} r_1, \\ (y) \\ r_2 \end{matrix}\right]\begin{matrix} s_1 \\ (z) \\ s_2 \end{matrix} \equiv \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2}(-1)^{i+j+k+1} f(q_i, r_i, s_i) \qquad (2.15)$$

In the case where the rectangular bar has small width and thickness values, the partial self-inductance can be approximate using equation (2.16) according to [49].These equations are less computationally expensive than equations (2.13) – (2.15) since it does not require the summation process with the total of 64 calculations.

$$\frac{L_{wire}}{l} = f_{ind}(k) = 2^{-7} * (\ln\left(\sqrt{k^2+1}+k\right) - \sqrt{\frac{1}{k^2}+1} + \frac{0.9054}{k} + 0.25 \qquad (2.16)$$

where k=l/r, l is length of the wire in m, and r is the wire radius in m.

In the case of equations (2.13) and (2.15) it is computationally expensive to compute thousands of summation terms for the final result. The Python code is not efficient for this type of computation. Thus, some alternative programing approaches are needed. This analysis is shows in the benchmark in Appendix A. It worth mentioning that, unlike the simple wire equation in (2.16), (2.13) is the approximation for a complex 3-fold integration. Moreover, in the 3D space, the relative locations of the rectangular bars matter and need to be handle very carefully.

2.3 The Eddy Current Effect in MCPM and Its Impacts on Self Inductance

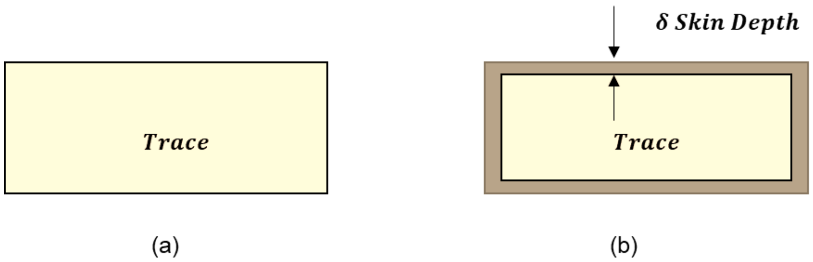


Fig. 2.5 Cross section of a conductor trace (a) at DC range (b) at high frequency

In the case of an MCPM layout, as the AC current flows on the topside of the DBC substrate, there are multiple effects contributing to the changes in the electrical parasitic values. One such effect is known as the “skin effect” where the AC current tends to flow on the outer surface of the conductor through a region called the skin depth (Fig. 2.5). As frequency increases, the current moves toward the surface of the current and result in smaller skin depth (equation (2.19)). The reduction of skin depth in turn increases the effective resistance of the trace. The second effect is known as the eddy-current loss effect. Here, the AC current in the topside DBC creates an alternating magnetic field in the plane perpendicular to its conduction path (Fig. 2.6). According to equations (2.1) and (2.2), this alternating magnetic field generates an induced current in the backside copper which flows in the opposite direction to the AC current. Oppose to the case with no backside in Fig. 2.6 (a), the eddy current in Fig. 2.6 (b) forms a loop with the AC current on the topside copper. This effectively reduce the inductance value as frequency increases. Moreover, this eddy current effect also results in the reduction of mutual inductance among traces sharing the same backside copper.

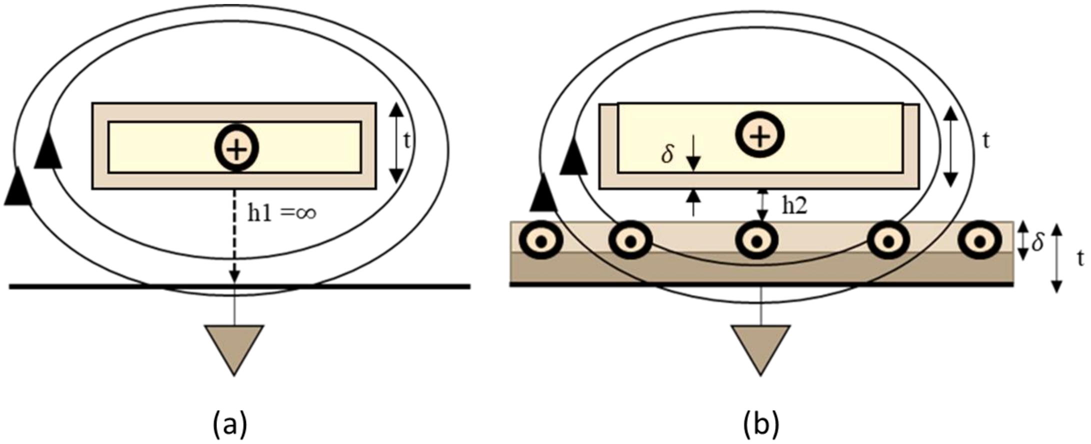


Fig. 2.6 Cross section of the conductor trace for (a) no backside, (b) with backside, where h_i is the distance to backside, δ is the skin-depth value, and t is the copper thickness

$$\delta = \sqrt{\frac{2}{\omega u \sigma}} \tag{2.20}$$

where ω is the angular frequency (rad/s), and σ is the conductivity of the material.

To further understand the eddy current effect and its impacts on the MCPM trace inductance, Ansys Maxwell FEA simulation is performed. A simple trace structure (Fig. 2.7) is created on top of an Alumina DBC substrate (0.2 mm - 0.64 mm - 0.2 mm). Here, the trace width is set to 1mm while the trace length is fixed to 20 mm. A line object is draw on the surface of the backside copper to measure the current density value on the backside. Fig. 2.8 shows the eddy current density for various frequencies: 10kHz, 100 kHz and 1MHz. The results show that the current is more crowded near the trace center for increasing frequency. This also explains why the self-inductance tends to be smaller at higher frequency (Fig. 2.9). As shown in Fig. 2.9, the relative difference between low-frequency value and high-frequency value of the trace self-inductance is about 25%. Fig. 2.10 illustrates the normalized current density of the eddy current measured on the backside. The result shows that the current density of the eddy current maximizes at the center of the trace and reduces further from the center. The eddy current density becomes insignificant at a fixed distance x_0 from the center.

While it is possible to capture this eddy current impact from the FEA simulations, they put a quite high workload on the CPU. Instead, a closed form equation derived from the Faraday's equation from [27] for microstrip trace for PCB can be used:

$$J_{gr}(x) = \frac{I_i}{w\pi}[\tan^{-1}\left(\frac{w-2x}{2h}\right) + \tan^{-1}\left(\frac{w+2x}{2h}\right)] \tag{2.21}$$

where I_i is the total current through the trace, *w* is the trace width, *x* is the distance from the trace center, and *h* is the distance from the trace to backside.

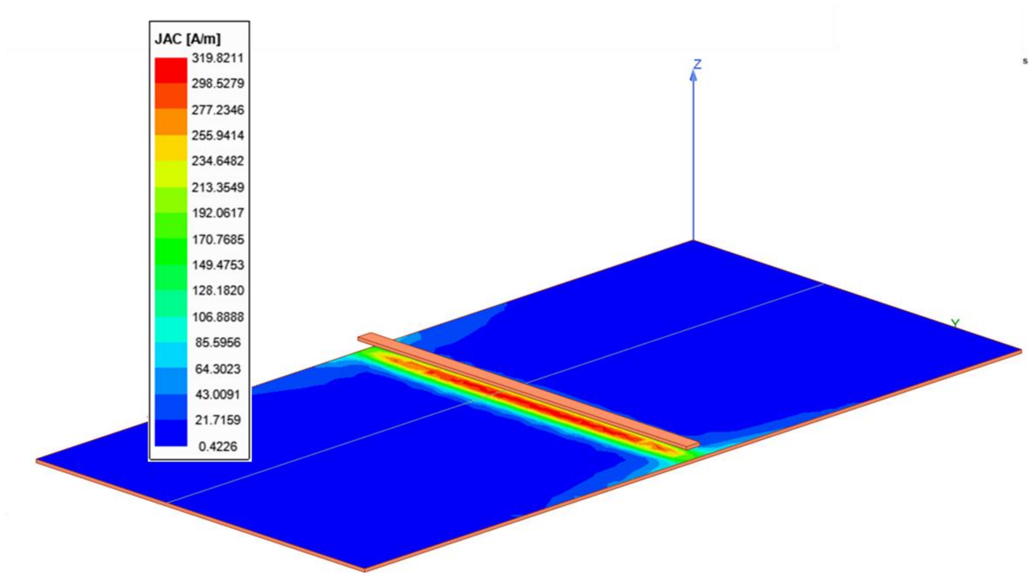


Fig. 2.7 MCPM trace with current density plotted on the backside copper

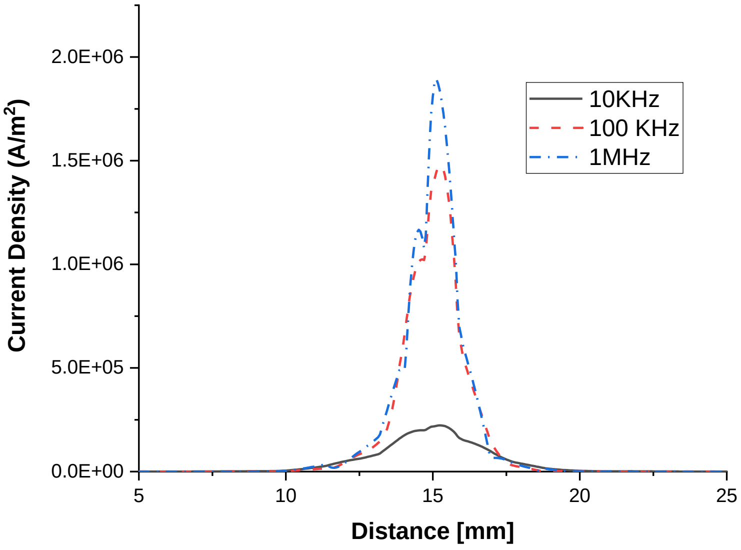


Fig. 2.8 Eddy current density with increasing frequencies

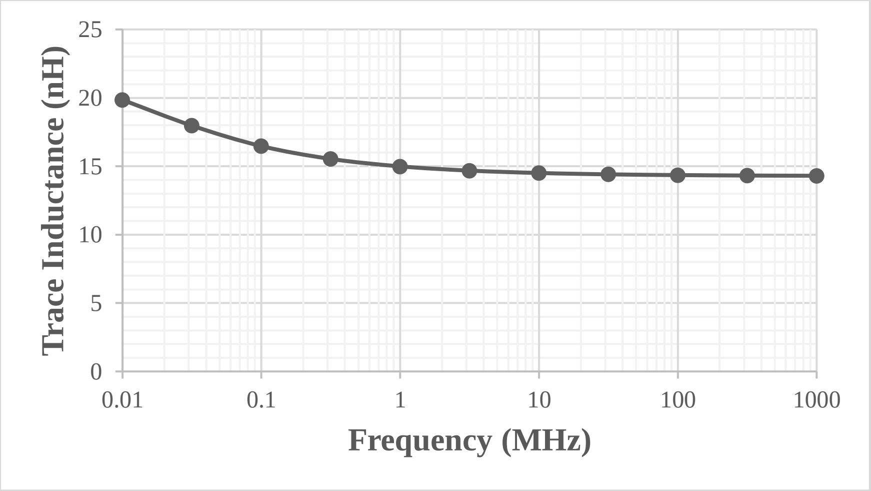


Fig. 2.9 Inductance reduction versus frequency for an MCPM trace

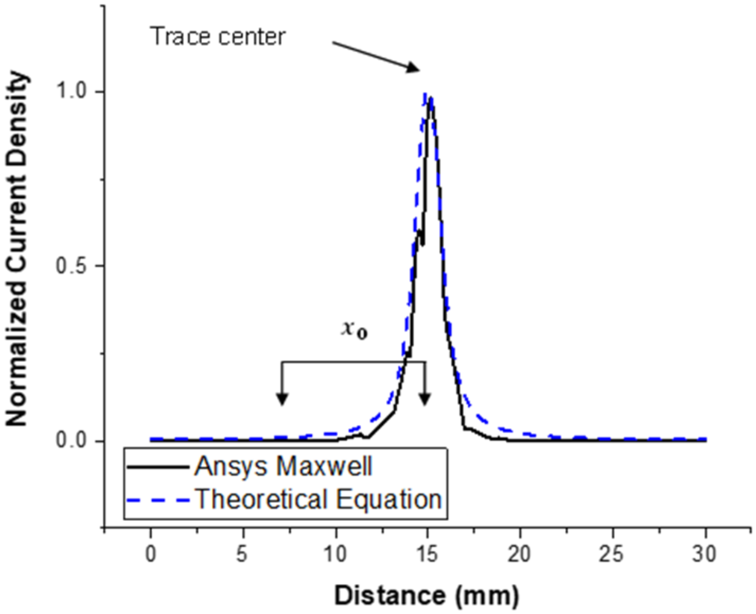


Fig. 2.10 Normalized current density versus analytical equation

From equation (2.21), the eddy current density value is independent from trace length and thickness. The value is dependent with trace width w and trace distance to the copper backside h. As seen in the results from Fig. 2.10, the numerical simulation shows the same distribution (normalized) versus the theoretical case using equation (2.21). Therefore, this equation can be used to evaluate x_0 value where eddy current zeroed out.

2.4 Model Formulation Process in PowerSynth

In the previous development of PowerSynth [22], equations for resistance, inductance, and capacitance have been incorporated from the microstrip equations and applied for MCPM case. In this work, some parametric studies on MCPM geometric parameters such as trace widths, lengths, copper thickness and substrate thickness have been done. This study shows that the analytical parasitic capacitance equation is most accurate compared to FEA simulation. The analytical

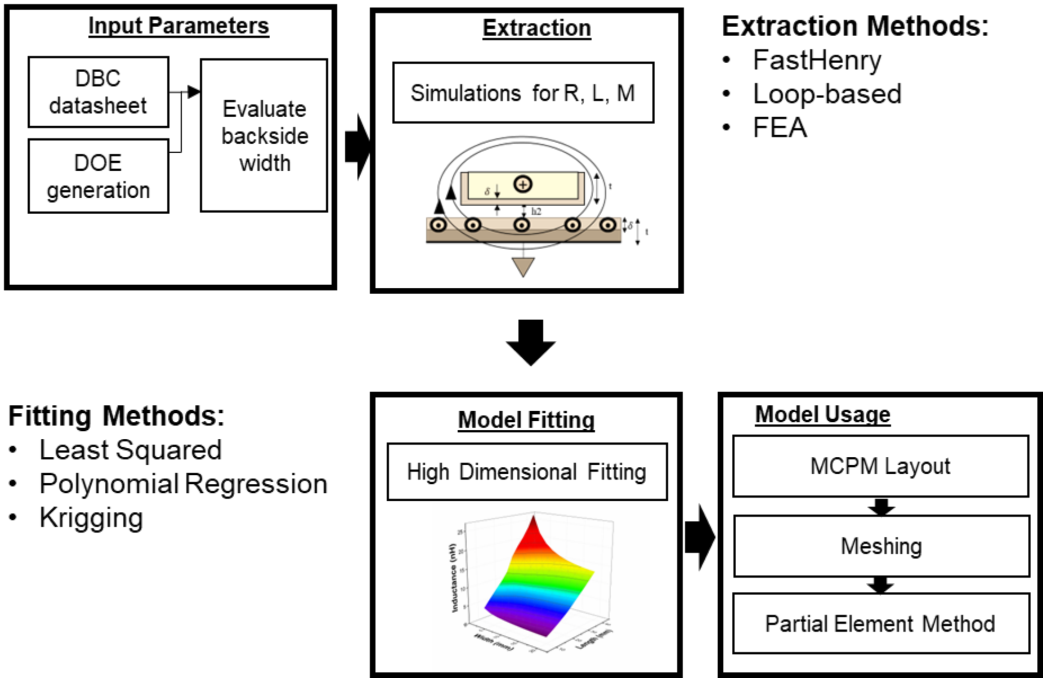


Fig. 2.11 Extraction Process for MCPM trace model

equation for resistance is relatively accurate and also frequency dependent. However, the analytical inductance equation has some limitations and high inaccuracy due to the zero-thickness assumption of the microstrip model. The author in [23] also claims that mutual inductance is insignificant for MCPM layout, which is not entirely true. Some examples in Chapter 4 and 5 will show that the consideration of mutual inductance is very crucial. The consideration for mutual inductance in this work allows the layout optimization algorithm to find a better solution space.

To overcome the limitations from the microstrip model, in this thesis, the self and mutual inductance models are built from numerical simulations. The extracted results from these numerical simulations can be collected to form regression models for L_{trace} and M_{trace} based on the study in [50]. These models can be used in any partial element approach to quickly evaluate the layout's loop impedance. The model formulation process is illustrated in Fig. 2.11. This approach allows a much faster extraction in the scope of design automation and optimization.

A. Input Parameters

A set of input parameters need to be first generated for the parametric simulation through a Design of Experiment (DoE) process. There are two different variable sets in the case of the MCPM. The first set comprises fixed parameters from the vendor DBC substrate datasheet. These include conductor thickness, isolation thickness, and material conductivity. Since there are limited

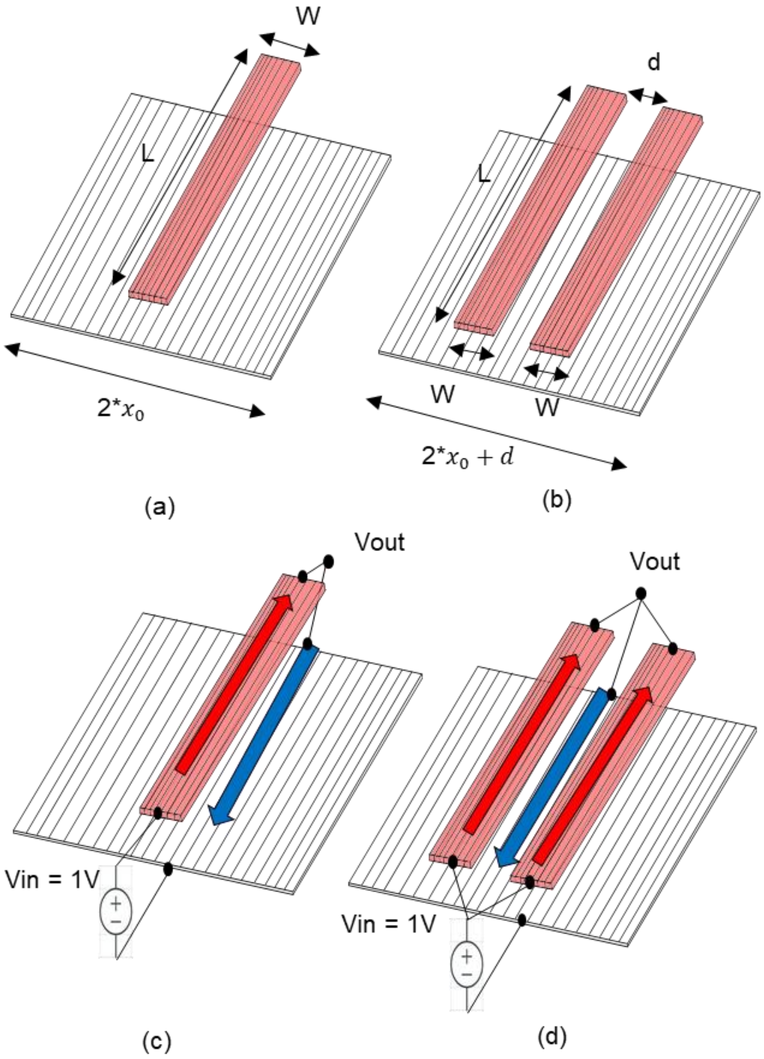


Fig. 2.12 Input parameters for (a) a single trace (b) mutual inductance between traces, Simulation setup for (c) single trace impedance (d) mutual inductance.

DBC types in the market, a library (Fig. 2.11) can be built based on the datasheet information. This library can be further extended if needed for new DBC substrate structures. For each different datasheet, a second set of independent variables are generated. In the case of self-inductance and resistance, the parameters are width W and length L of the trace. For the mutual inductance model the input parameters are the trace widths W, trace length L, and the trace-trace distance d. Appropriate widths and lengths values are generated based on the skin-depth δ_{100MHz} value at 100 MHz. The range for each parameter is shown in Table 2.1 below. These input parameters cover some common dimensions for the mesh size used in the mesh algorithm which is discussed in more detail in Chapter 4.

Table 2.1 Input Parameters Range for Model Formulation

Parameter Name	Definition	Min Value (mm)	Max Value (mm)
W	Trace Width	δ_{100MHz}	1
L	Trace Length	$5 \times \delta_{100MHz}$	25
d	Trace-Trace distance	0	10

B. Impedance Extraction Using Loop-based Extraction Method

The extraction step can be performed using Ansys Q3D FEA simulations or FastHenry [33] as described in [16] through running parametric simulations. In this work, a loop-based approach can be used to perform the extraction while similar extraction accuracy can be maintained. The loop-based approach is a method from VLSI [49] which has been used to improve the computational efficiency of the inductance extraction. This has been used in [50] for MCPM inductance extraction purposes. To begin with, the trace geometry is setup as seen in Fig. 2.12 (a)-(b). The geometry is discretized in multiple parallel wires, where each wire width is set close to the skin-depth of the maximum extracted frequency. For inductance calculation, the analytical equations in Section 2.2 are used to evaluate the self-inductance and mutual inductance among the wires. The self-resistance of the wire is evaluated using the equation below:

$$R_{wire} = \rho . \frac{l}{A} \tag{2.22}$$

where ρ is the resistivity (Ω/m), l is the wire length (m) and A is the wire cross section area (m^2)

These evaluated self and mutual impedance values are stored in an n-by-n matrix **P** where n is the total number of elements. The partial self-impedance values are stored on the diagonal of **P** matrix, and the partial mutual inductance values are stored in the upper and lower triangles of **P**. Next, an n-by-k mesh matrix **M** is formed where k is the number of mesh elements of the trace. If the mesh element belongs to the trace, **M**(i, j) is marked with 1. Otherwise **M**(i, j) is marked with zero if an element belongs to the backside copper. A column vector **u** contains n rows each filled with one. Two different equations are first evaluated to find the current distribution and current value through each wire:

$$\boldsymbol{Pa} = \boldsymbol{u} \tag{2.23}$$

$$\boldsymbol{PB} = \boldsymbol{M} \tag{2.24}$$

Vector $\mathbf{a}$ 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 obtain the current matrix **I**, whose *j-th* column vector **I**(*j*) can be calculated using:

$$\boldsymbol{I}(j) = \boldsymbol{B}(j) - V_{out} \times \boldsymbol{a}, where\ j = 1,2, \dots, n \tag{2.25}$$

where:

$$V_{out} = \frac{\Sigma \boldsymbol{B}(i,j)}{\Sigma \mathbf{a}} \tag{2.26}$$

Let k be the total number of loops forming from the top-trace and backside. The total current through each loop is computed by:

$$\boldsymbol{I_{trace}} = \boldsymbol{M^T} \times \boldsymbol{I} \tag{2.27}$$

then:

$$\boldsymbol{Z_{trace}} = (\boldsymbol{M^T} \times \boldsymbol{I_{trace}})^{-\mathbf{1}} \tag{2.28}$$

These evaluations are performed for all variation of parameters and the extracted frequency (f). The results for R_{trace}, L_{trace}, and M_{trace} are stored in table form for the model fitting process. Table 2.2 shows the comparisons among the extraction using different models and for some selected input parameters. The result shows error within 6% between the loop-based method versus Ansys Q3D for both self and mutual inductance extraction. The open-loop equation shows 30-40% error for both self and mutual inductance cases. Table 2.2 shows the memory and time comparisons using the loop-based method versus FEA simulation using Ansys Q3D. On the same computer, for this simple structure, the model is 7× faster and 62× more memory efficient than the FEA approach.

Table 2.2 Trace Impedance Extraction Performance Loop-based vs FEA

	Ansys Q3D		**Loop-based**	
Runs	**Time**	**Memory**	**Time**	**Memory**
1	70 s	310 MB	9s	5 MB
100	1h 40 m		15 m	

C. Regression Methods for Self and Mutual Impedances

To evaluate the self and mutual impedances in a layout extraction process, all extraction results are collected and stored in a table form. Here, a response surface model is formulated. Response surface model is a mathematical method to find the relationship between the input vectors $\boldsymbol{X}$ to a single for multiple output $\mathbf{Y}$. For this response surface model formulation, the Python Scikit-Learn library [51] has been used. This package offers vary response surface modeling techniques such as linear regression, polynomial regression, support-vector regression, kernel-ridge regression, and machine learning regression such as neural network. To find the best model for self and mutual inductance fitting, an accuracy analysis has been performed for the same set of simulated data. The result has shown that polynomial regression is most suitable for fitting the impedance equation. This is because this model allows a nonlinear fitting while having a simple equation form as shown below:

$$f_{RL}(W,L) = \sum_{n}^{N} a_n W^n + \sum_{n}^{N} a_n L^n + \sum_{m}^{M} a_m W^m L^m \tag{2.29}$$

$$f_M(W,L,d) = \sum_{n}^{N} a_n W^n + \sum_{n}^{N} a_n L^n + \sum_{n}^{N} a_n d^n + \sum_{m}^{M} a_m W^m L^m d^m \tag{2.30}$$

where W is width in mm, L is length in mm, d is distance in mm, a_i are coefficients to be fitted, N is the number of degrees, and M is the number of interaction coefficients.

Here, a third order polynomial ensures a fitting error of 10% for R and 1% for L. As for the mutual inductance model, a fourth order polynomial equation form ensures a fitting error of less than 1%. Once the model is characterized, it can be reused many times for the same DBC structure. More importantly, the characterized model is much faster and memory efficient than using FEA or Loop-based simulation.

For power module applications, most of the time, the extracted inductance results at high frequency are accurate for most circuit simulation. However, the low frequency inductance can couple with other circuit elements such as decoupling capacitors resulting in low-frequency noises in the output [52]. Thus, the consideration of a broadband inductance is necessary for these analyses. A ladder circuit model (Fig. 2.13) from [52] can be used to effectively estimate the inductance result for a broad frequency range. This ladder circuit is accurate for maximum frequency of 6GHz as mentioned in [52] which is well beyond the frequency range for MCPM applications. The elements in the ladder circuit can be calculated using the equations from [52]:

$$R_0 = \frac{R_{dc} L_{dc}}{L_{hf}} \tag{2.31}$$

$$R_1 = \frac{R_{dc} L_{dc}}{L_{dc} - L_{hf}} \tag{2.32}$$

$$L_0 = L_{hf} \tag{2.33}$$

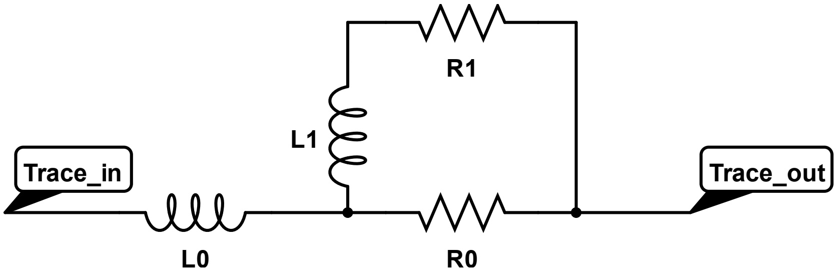


Fig. 2.13 Ladder Circuit for Broadband Frequency Extraction

$$L_1 = \frac{L_{dc}^2}{L_{dc} - L_{hf}} \tag{2.34}$$

where R_{hf} and L_{hf} are high frequency resistance and inductance, R_{dc} and L_{dc} are low frequency resistance and inductance.

Chapter 3. Layout Engine to Electrical Evaluation Application Programming Interface (API)

With a more accurate model, the partial element method can be applied to evaluate the parasitic result of MCPM layouts. To this end, the 2D layout information must first be processed and converted into the 3D geometries for the electrical evaluation. Along with the fast updates in layout engine and the PowerSynth tool development, the Application Programing Interface (API) needs to be updated to handle different types of MCPM structure. During this development, the electrical model needs more information from the layout engine and the tool to ensure accurate connectivity among components, leads, bond wires, and vias.

This chapter describes the process of converting the 2D layout information from the layout engine to the electrical model data structure. An API (Fig. 3.1) is implemented. The conversion from layout to electrical engine for each component is described. Each component, such as traces, devices, bond wires, and via, is handled differently in the API to ensure the correct network connection. Once the layout objects are converted into electrical engine objects, a meshing algorithm discretizes the layout into multiple smaller elements. The Partial Element Equivalent Circuit (PEEC) or loop-based method is then applied to solve the parasitic loop solution.

Section 3.1 details some of the basic elements of a Manufacturer Design Kit (MDK). These include layer-stack information and Design Rule Check (DRC) constraints. These elements also play an important role in the result of the extraction.

Section 3.2 visits some of the previous treatments for devices and components in the previous version of PowerSynth. Due to the incorrect data structure used in the previous PowerSynth, some of the loop evaluation is not accurate. Here, some improvements in both the data structure and evaluation have been made to ensure the most accurate extraction result.

Section 3.3 details the transition between the 2D layout information from the layout engine to the 3D geometries used in the API. This section also discusses how the electrical data structure are designed to handle most of the current supported layout from 2D to 3D in PowerSynth.

Section 3.4 describes the process of circuit hierarchy formulation. This step has been done to ensure correct connections among the components. Here, a simple verification can be performed to ensure a correct Layout Versus a Schematic (LVS) check between the input netlist and the layout.

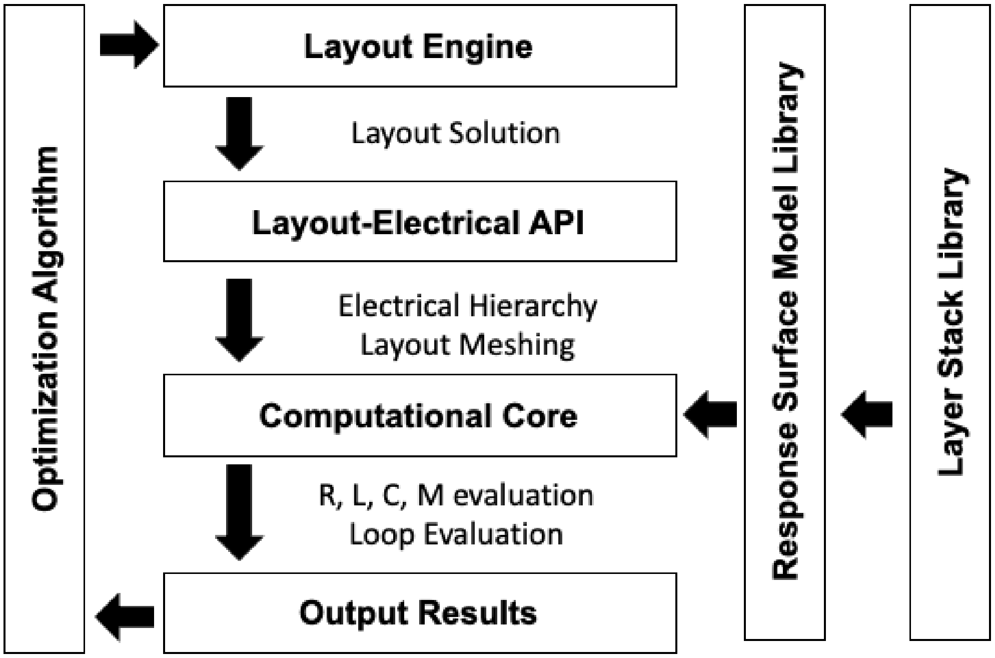


Fig. 3.1 Layout engine to electrical evaluation API

3.1 The Manufacturer Design Kit (MDK)

In VLSI, a Process Design Kit (PDK) [53], [54] is required for every design to ensure a reliable and manufacturable layout. This kit dictates the constraints on the chip layout, such as minimum spacing and minimum widths. Thus, this has a direct impact on the circuit parasitic parameters such as inductance and capacitance. Similarly, PowerSynth adopts a similar concept through a Manufacturing Design Kit (MDK) for MCPM design [55]. This MDK includes material information, layout constraints, layer stack setup, and thermal/electrical information for each MCPM design.

The layout engine takes input constraints information to ensure Design Rule Check (DRC) clean MCPM layout generation by using a constraint-aware layout engine [17]. These constraints are contained in a CSV file provided by the user. The constraints include minimum size trace-

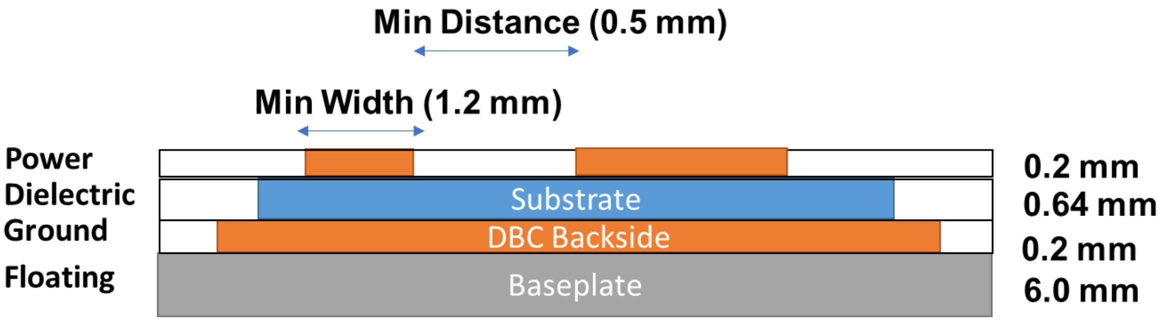


Fig. 3.2 A layerstack representation in PowerSynth

trace, trace-component, component-component, etc. As shown in the study in [21], different design rules and constraints greatly impact the parasitic extraction results. This is because these design rules vary the loop area of the AC current leading to higher inductance results with higher trace gap and vice versa.

Another factor that would affect the parasitic extraction is the layer-stack input. From Chapter 2, it has been shown that as the isolation thickness (h) or conductor thickness (t) changes, the eddy-current density would change, leading to a different parasitic extraction result. Fig. 3.1 illustrates an example of a layer stack setup for a 2D MCPM layout. This layer-stack information is provided through a text file as shown in Table 3.1, which includes a layer-by-layer material type, thickness, area, etc. The first column of the layer stack input shows the layer index for each layer, and this information is mapped to each of the elements in the electrical API, which is later used to correctly assign the elevation of each conductor, device pins, and wire bonds or via. As shown in Table 3.1, layers 2 to 5 contain information for electrical evaluation. Layer 5 is used to find the correct connection between device components to the trace conductor to form a correct electrical hierarchy. Layers 2 to 4 are used to apply correct regression model from the library for an accurate evaluation of electrical partial impedances.

Table 3.1 A layer-by-layer layerstack input from the user

ID	Name	Origin	Width	Length	Thickness	Material	Electrical
1	Baseplate	0,0	50	60	5	copper	F
2	Bottom Metal	0,0	40	50	0.2	copper	G
3	Ceramic1	0,0	40	50	0.64	Al_N	D
4	I1	0,0	40	50	0.2	copper	S
5	C1		40	50	0.18	SiC	C

3.2 Components, Bondwires and Vias

While the layout engine has taken care of the DRC for each different device types, bondwires, and vias, they are simply seen as 2D rectangle or a simple line respectively. Each of these elements need to be considered differently in the electrical evaluation to ensure the most appropriate matrix setup for the Modified Nodal Analysis (MNA) solver.

A. Multi-pins Components Representation

In the previous version PowerSynth V1, a single node has been used to represent any component types. In this representation, the circle object only allows 4 connections to a single net. Hence, this representation limits the number of component types and connections. Moreover, as shown in Fig. 3.3 this component object assumes every terminal is connected to a single node which is not correct. This implementation results in mixed current path between the power and signal, leading to wrong extraction. For example, in this case, the gate loop and the power loop are connected, resulting in two parallel signal and power loops branches.

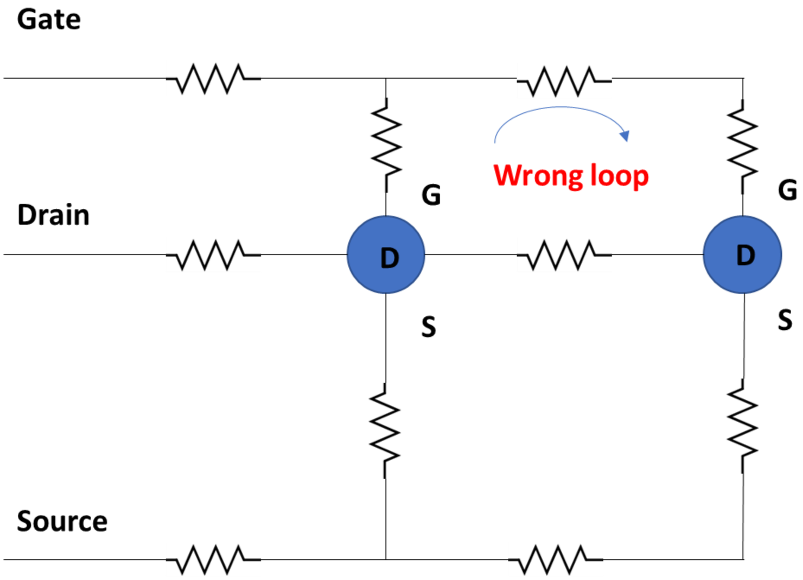
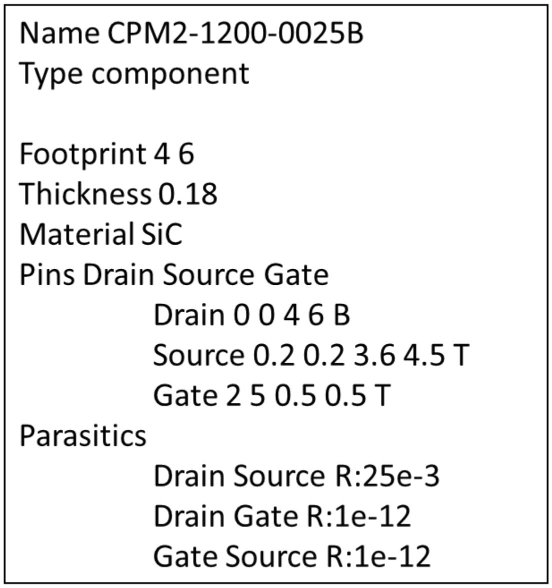


Fig. 3.3 PowerSynth V1 device component representation



```
Name CPM2-1200-0025B
Type component

Footprint 4 6
Thickness 0.18
Material SiC
Pins Drain Source Gate
        Drain 0 0 4 6 B
        Source 0.2 0.2 3.6 4.5 T
        Gate 2 5 0.5 0.5 T
Parasitics
        Drain Source R:25e-3
        Drain Gate R:1e-12
        Gate Source R:1e-12
```

Fig. 3.4 Example device component technology file for a SiC MOSFET

Therefore, a new component object has been updated to ensure that a correct electrical analysis can be performed. Fig. 3.4 shows the text input for the new component object, this new type of input allows the user to set any number of desired pins, relative locations of the pins to the device, and internal parasitic values between each pair of the pins. Fig. 3.5 illustrates the updated net representation of the device pins. For each component, the order of the pin-pin connection is defined below the “Parasitics” line where the user can define device internal pin-pin parasitic values such as R and L. One example use for this pin-pin parasitic value is the incorporation of the “Rds_{On}” parameter of the Mosfet. This value can vary between each physical device due to the imperfection during chip fabrication. Due to this reason, the steady-state current of each device can vary for different “Rds_{On}” values. With this implementation, in PowerSynth, the user can provide different technologies file if needed to represent the “Rds_{On}” variation. It is possible to probe the current through each device as will be discussed in Chapter 4.

Here the pins are initially disconnected, and the user needs to provide an array of 0 and 1 following the same order defined in the component definition. For instance, Fig. 3.5 illustrates the setup with both devices state of [1,0,0] which means there is a connection between Drain-Source pins and no connection between Drain-Gate or Gate-Source. This is a common setup in PowerSynth to find the total power loop parasitic inductance value. Using this configuration, the user can also define the gate-kelvin loop as seen in Fig. 3.6.

The red dotted line in both Fig. 3.5 and Fig. 3.6 are replaced with a zero-voltage independent source in series with a parasitic parameter (e.g Rds_{On}) inside the tool with a distinguished name for each device e.g VD_x, where x denotes the device index. As will be discussed in Chapter 4, by doing so, the two nets are effectively shorted, while the current through each element can be probed during the Modified Nodal Analysis (MNA) evaluation. This allows the tool to correctly calculate the steady-state current through each branch. Such information is important for the lumped netlist extraction step in Chapter 4.

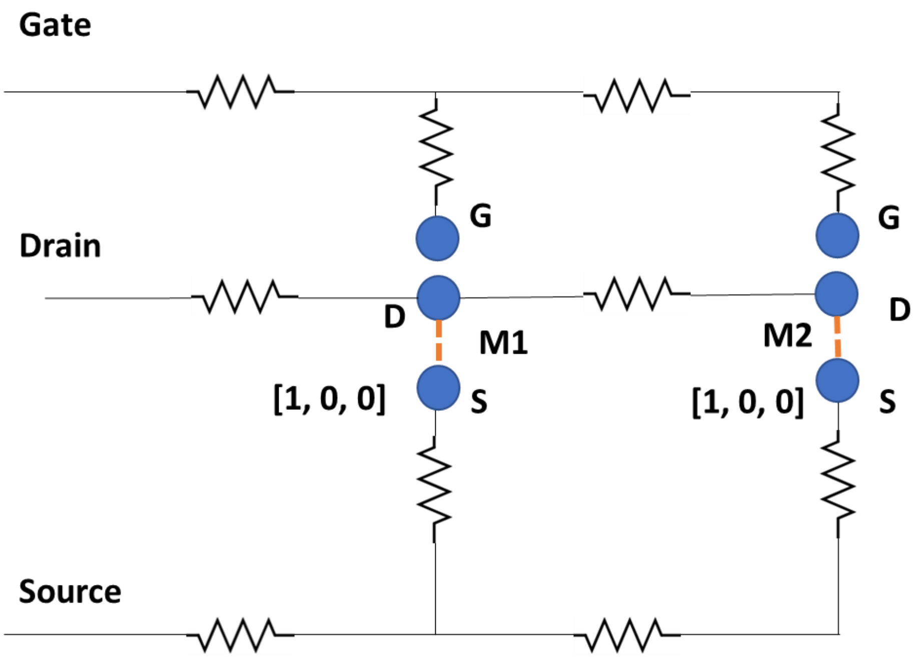


Fig. 3.5 Power loop setup for device state with drain to source connection [1,0,0] for each device

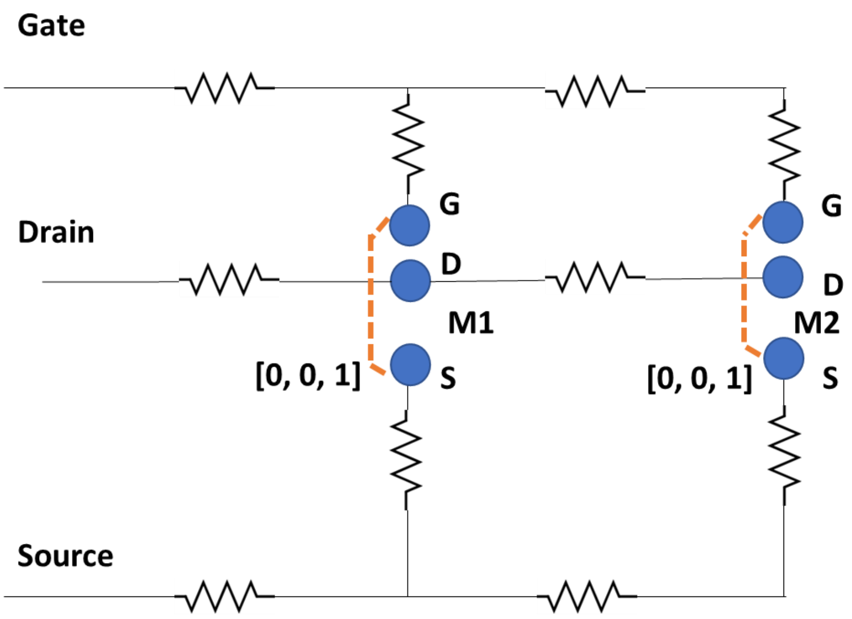


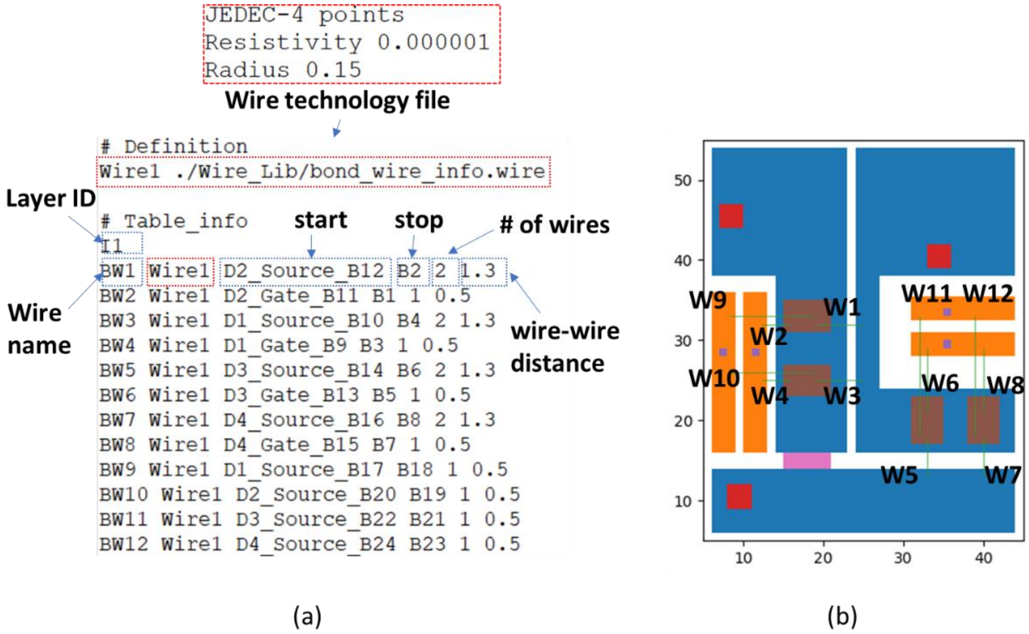
Fig. 3.6 Gate-Kelvin loop setup for device state with gate to source connection [0,0,1]

B. Bond wires and Via groups

Once the device state is provided, the connectivity information for bondwires and vias need to be provided to ensure correct wire and via connections in the electrical model. The connection table for bondwires (or vias) is shown in Fig. 3.7 (a) for the example layout in Fig. 3.7 (b). From the first few releases of the tool, this connection table has been provided as a separate file where the user has to define the bondwire (or via) connections among the bondwire pads. Recently, this connectivity table is also incorporated in the layout script as well to ensure a more user-friendly interaction. To describe the bondwire table, the old version in Fig. 3.7 has been used.

As seen in Fig. 3.7 (a), the user needs to provide the technology file for the bondwire table under the #Definition tag. This technology defines the bondwire technology, material resistivity, and most importantly the wire radius. This technology information is passed into the electrical evaluation engine to ensure the correct evaluation for the R, L, and M of the wires. In the bondwire connectivity table, the user also needs to define which layer the bondwires are on (in this case I1), each line below the layer ID represents different bondwire groups. For each line, the user needs to input the start name, stop name of the pads, number of parallel wires in the group, and wire-wire distance.

This bondwire (Via) table is important for both layout engine algorithm and the electrical parasitic extraction. In the layout engine, these wires and vias dictate some trace intralayer constraints or layer-layer constraints respectively. In the electrical parasitic extraction engine, using this connection table, a virtual edge is connected between the two ends of the bondwire where the edge value is evaluated using the wire resistance, inductance, and mutual inductance as described in [22]. This ensures the most accurate wire group impedance within the network. Similar solution is applied for vias in the 3D layout case. It is worth mentioning that this implementation allows more accurate consideration of current density among the wires and vias (solder balls). This information can be exported from the electrical evaluation engine to support further reliability research. This information can be served as inputs for electromigration model such as [56] for a more accurate reliability optimization.



```
JEDEC-4 points
Resistivity 0.000001
Radius 0.15
```

```
# Definition
Wire1 ./Wire_Lib/bond_wire_info.wire

# Table_info
I1
BW1 Wire1 D2_Source_B12 B2 2 1.3
BW2 Wire1 D2_Gate_B11 B1 1 0.5
BW3 Wire1 D1_Source_B10 B4 2 1.3
BW4 Wire1 D1_Gate_B9 B3 1 0.5
BW5 Wire1 D3_Source_B14 B6 2 1.3
BW6 Wire1 D3_Gate_B13 B5 1 0.5
BW7 Wire1 D4_Source_B16 B8 2 1.3
BW8 Wire1 D4_Gate_B15 B7 1 0.5
BW9 Wire1 D1_Source_B17 B18 1 0.5
BW10 Wire1 D2_Source_B20 B19 1 0.5
BW11 Wire1 D3_Source_B22 B21 1 0.5
BW12 Wire1 D4_Source_B24 B23 1 0.5
```

Fig. 3.7 Bondwire (Via) connectivity input file

3.3 Data Structure for 3D Geometry in Electrical Evaluation Engine

The electrical evaluation engine has two primary data structures for the geometrical representation. The first type is called “E_plate,” a rectangle box with width, height, thickness, and elevation parameter (z). These dimensions use the same base unit (1um) as the layout engine. This “E_plate” object is used in the tool to convert the 2D geometrical information from the layout engine into conductor trace with thickness. Likewise, the second object is “E_sheet”, a rectangle sheet with zero thickness and an elevation parameter (z). This sheet object is used to represent the net connectivity between the trace object and the contact surfaces of each component, such as leads connectors (L), devices pads (D), and bondwire pads (B). Here, a suffix is added automatically for each sheet object depending on the component type defined by the user to handle the pins connections of the components described in section 3.2. For example, in the case of a SiC power Mosfet defined in Fig. 3.4, the three additional pins are “Dx_Drain”, “Dx_Source”, and “Dx_Gate”, where x denotes the index reference of the device in the layout.

Furthermore, in most 2D and 2.5D MCPM layouts, most of the sheet are assumed to face upward. This is because most of the bondwire connections are on the topside of the DBC and most SiC power MOSFETs are vertical devices. However, with the fast development of 3D power module packaging, more directions of the sheets are required. For instance, in a multilayer 3D structure, there are some cases where the sheets are facing towards each other (double-sided cooling) or facing in opposite directions. Hence, a vector parameter “n” is included to represent the direction of the sheet. Fig. 3.8 (a) illustrates the faceup direction for a sheet object for a 2D layout case where Fig. 3.8 (b) represents the possible facedown direction for 3D layout case. This implementation also supports other pad directions. In the future, it is possible to handle folded structure such as [57].

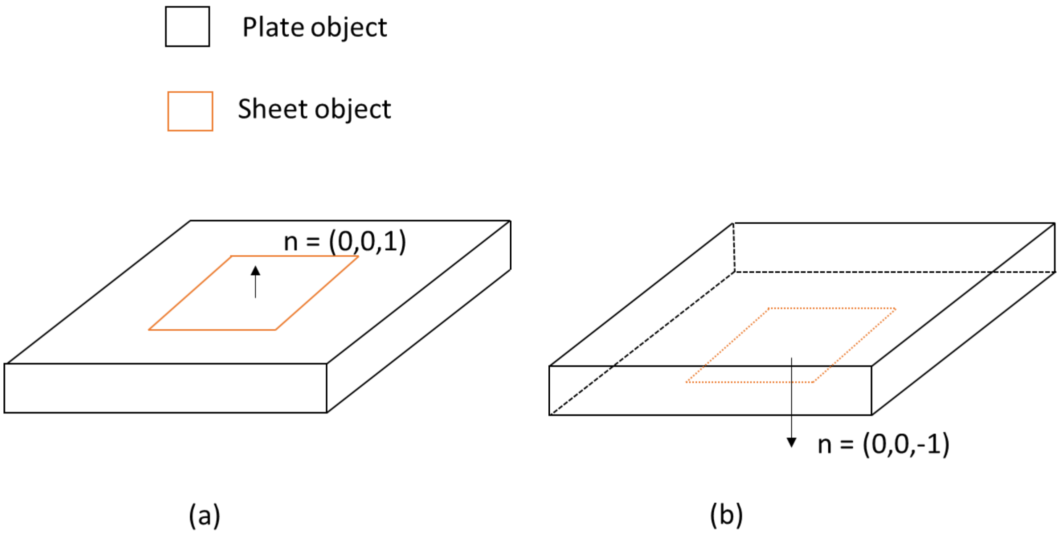


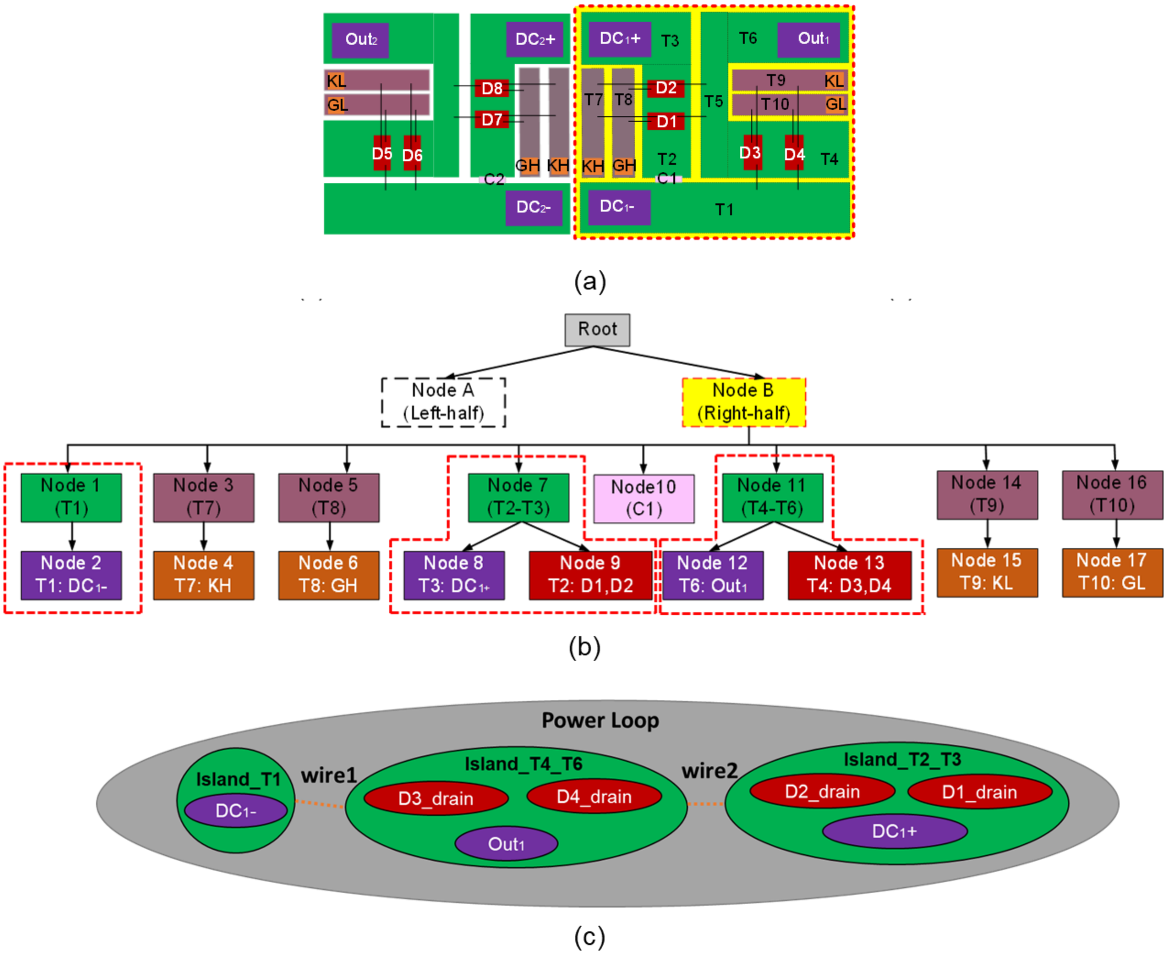
Fig. 3.8 face-up (a) and face-down (b) positions for a sheet object

By using the sheet and plate locations along with the elevation level of each object from the layer stack, the electrical evaluation engine can correctly find connections among the electrical components in the layout. This is important for the electrical hierarchy formation in the next section.

3.4 Layout to Electrical Hierarchy Conversion

The layout engine algorithm [18] hierarchically manages each layout element to ensure correct design constraints. From an MCPM layout input Fig. 3.9 (a), a tree data structure Fig. 3.9 (b) has been used to ensure correct layout constraints among the components. This hierarchical information can also be used in the electrical model to ensure correct connectivity among the nets. In the electrical evaluation engine, an algorithm iteratively searches through each layer, where the electrical hierarchy is formed using the trace island information. Each trace island contains geometry information of the copper traces and the components on top of the island.

Instead of using a tree data structure, a hyper-graph has been used to ensure correct net connectivity on each layer. Fig. 3.9 (c). Once this step is done, if the user provides an input netlist, this hyper-graph can be compared versus the input netlist to perform an LVS verification. This step makes sure the circuit design and the layout are similar. This verification step could be important once the layout synthesis algorithm is ready [58]. Another application for this hyper-graph is to ensure that all floating nets (gate nets) are removed or grounded adequately during the single loop evaluation. For example, the gate net must be grounded or removed during the DC-DC parasitic extraction. If this is not handled, the left-hand-side matrix of the MNA evaluation will be singular due to missing connections in the Adjacency matrix.



(a)

(b)

(c)

Fig. 3.9 (a) Layout Input (b) Layout engine hierarchical tree structure (c) Power loop net in a hypergraph.

Chapter 4. Methodologies for Parasitic Extraction in MCPM Layout Optimization

This chapter first describes the limitation of the previous parasitic extraction approach in PowerSynth v.1.0. The chapter also reviews some of the most common partial element evaluation methodologies for parasitic extraction. The advantages and disadvantages of each approach are summarized at the end of the chapter. The contents for each section of the chapter are as follows:

Section 4.1 overviews the Laplacian Matrix method and symbolic layout representation which have been previously used in PowerSynth. V1. This method has some limitations as it does not consider the mutual inductance effect among conduction paths in the Multichip Power Module (MCPM) layout. The section shows some layout examples where the Laplacian Matrix failed to accurately extract the inductance value.

Section 4.2 describes the Modified Nodal Analysis (MNA) approach [59]. This approach is used in many other methodologies such as Partial Element Equivalent Circuit (PEEC) [60] and Loop-based [49] to find the unknown currents and voltages.

Section 4.3 describes the fundamental of PEEC method and its application for MCPM. This method is later verified experimentally in Chapter 5.

Section 4.4 describes the Loop-based method and its application for MCPM. For a certain type of MCPM layout where most of the traces are oriented in horizontal and vertical directions, the current forward and return paths can be easily predicted through path-finding algorithm such as Depth First Search (DFS). The loop calculation can be simplified to further reduce the total number of circuit elements in the distributed netlist.

Section 4.5 overviews the multiport extraction procedure, which is crucial for some of MCPM layout optimization target such as current balancing and gate loop balancing. The method has been modified in PowerSynth to ensure the same power loop parasitic from DC+ to DC- while having a suitable lumped circuit model for the circuit simulation study. Some of the simulation results have been compared versus the Double Pulse Test (DPT) measurement in chapter 5.

4.1 The Limitation of Laplacian Matrix Method and Symbolic Layout

As described in [23], [25], a Laplacian Matrix method has been used in PowerSynth v1.0 to quickly estimate the loop impedance. In this version of PowerSynth, a Symbolic Layout object has been used to represent the MCPM layout, which has several limitations. To begin with, the matrix-based layout representation using Symbolic Layout does not ensure all Design Rule Checks (DRCs) are met. Here, a post-layout DRC process is run for each layout. If the DRC process fails for a layout during optimization, a very high fitness score is applied for this layout to remove it from the final pareto-frontier solution set. While this layout generation method works for simple layout cases, it is very inefficient for more complicated layout cases. As analyzed in [18], for one of the most complicated layout cases, the matrix-based layout generation method only achieve 40% DRC-free solution. The second limitation from Symbolic Layout representation is its limitation in supporting 2D current distribution in the layout. Here, the layout is discretized into multiple traces with either horizontal or vertical direction where an undirected graph has been used

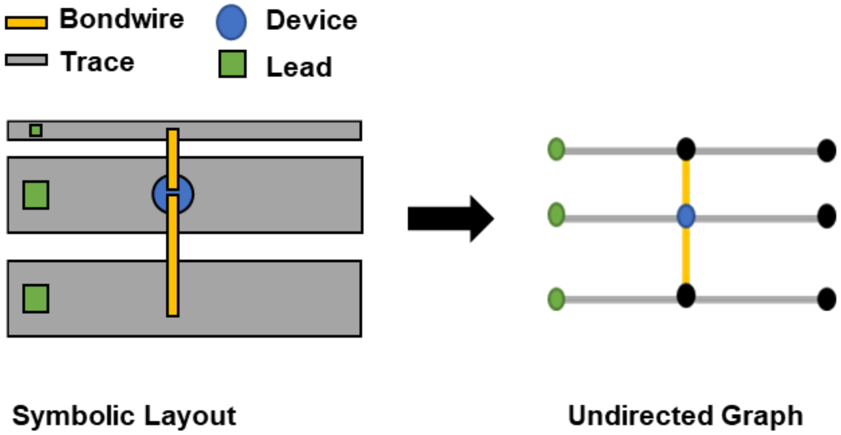


Fig. 4.1 Symbolic layout and undirected graph representation

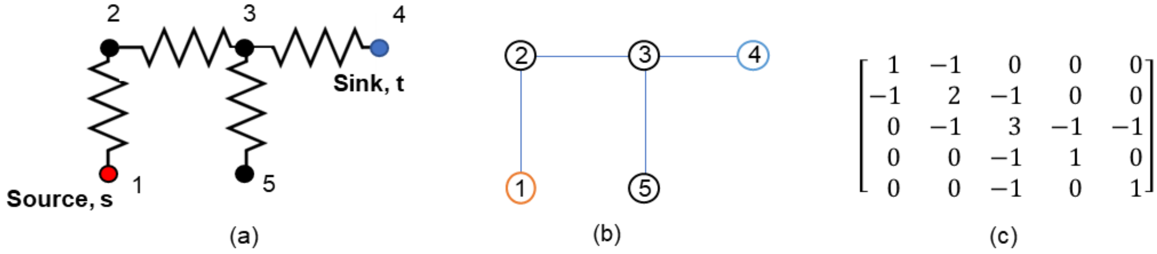
to store the layout information (Fig. 4.1). Each of the edge in the undirected graph stores the width and length information of the trace, while each vertex represents the trace-trace or component-trace connections.

Once the undirected graph is generated, the weights of its edges are populated using the microstrip-based equations from [22]. Each edge has two different weights denoted as R and L for resistance and inductance respectively. The undirected graph is then converted into a Laplacian matrix which is very similar to the admittance matrix representation in Spice [61][62]. The effective resistance and inductance are then calculated using the equation:

$$L_{eff}^{s,t} \ or \ R_{eff}^{s,t} = \boldsymbol{x}_{s,t}^{T}\boldsymbol{L}(G)^{+}\boldsymbol{x}_{s,t} \tag{4.1}$$

where $\boldsymbol{x}_{s,t}$ is a vector represents the current flow from source to sink, $\boldsymbol{L}(G)^{+}$ is the Moore-Penrose inverse of the Laplacian matrix of graph G.

Here, $\boldsymbol{x}_{s,t}$(i) equals 1 if the net i_{th} is a source net or sink net and 0 otherwise. The Laplacian matrix is illustrated in Fig. 4.2. To form this matrix, a value of +1 is added to the matrix diagonal term for every edge connected to net i_{th}. On the other hand, a value -1 is assigned to an off-diagonal term if there is an edge between node i_{th} and j_{th}.



$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

Fig. 4.2 (a) Parasitic network, (b) Graph representation, (c) Laplacian Matrix

The effective resistance or inductance in [63] is calculated by applying a unitary current stimulation from source node to sink node (Fig. 4.2). Then equation (4.1) can be used to quickly calculate the effective resistance or inductance value. This method only gives the best approximation using the parallel and series configuration of both inductance and resistance in a circuit. However, it does not consider mutual inductance which is crucial in capturing the proximity effect. To show the limitation of this method, 3 different simple U-shape configurations are designed in Fig. 4.3. First, Ansys Q3D has been used to extract the self-resistance and inductance of 3 different conductor traces with length of 10mm and width of 1mm, 2mm, and 3mm respectively. The self-resistance and inductance results of each trace piece is shown in Table 4.1 below. The conductor trace pieces are rearranged as shown in Fig. 4.3 to form 3 different U-shape configurations. Then, the inductance of the U-shape is extracted using both Laplacian technique and Ansys Q3D.

Table 4.1 Q3D extraction results for conductor trace of length 10mm and different widths @ 100 MHz

Trace width (mm)	Self-Resistance (mOhm)	Self-Inductance (nH)
1	13.131	6.283
2	8.210	5.023
3	5.798	4.567

Table 4.2 Extraction comparison between Ansys Q3D and Laplacian Matrix @ 100MHz

$\boldsymbol{Case}$	$\boldsymbol{R_{Lapacian}}$ **(mOhm)**	$\boldsymbol{R_{Q3D}}$ **(mOhm)**	$\boldsymbol{L_{Laplacian}}$ **(nH)**	$\boldsymbol{L_{Q3D}}$ **(nH)**
A	39.393	33.6	18.849	15.4
B	34.471	29.87	17.589	14.8
C	32.051	28.6	17.133	14.5

As shown in Fig. 4.3 all of the U-Shape structures have a similar internal loop with the size of 10mm × 10mm. Only a single trace piece with width of 1mm (Fig. 4.3 (a)) is replaced by other trace pieces with width values 2mm and 3mm (Fig. 4.3 (b) (c)). If the Laplacian Matrix method is used and the traces are in a series configuration, the loop inductance values are calculated using equations (4.2) - (4.4):

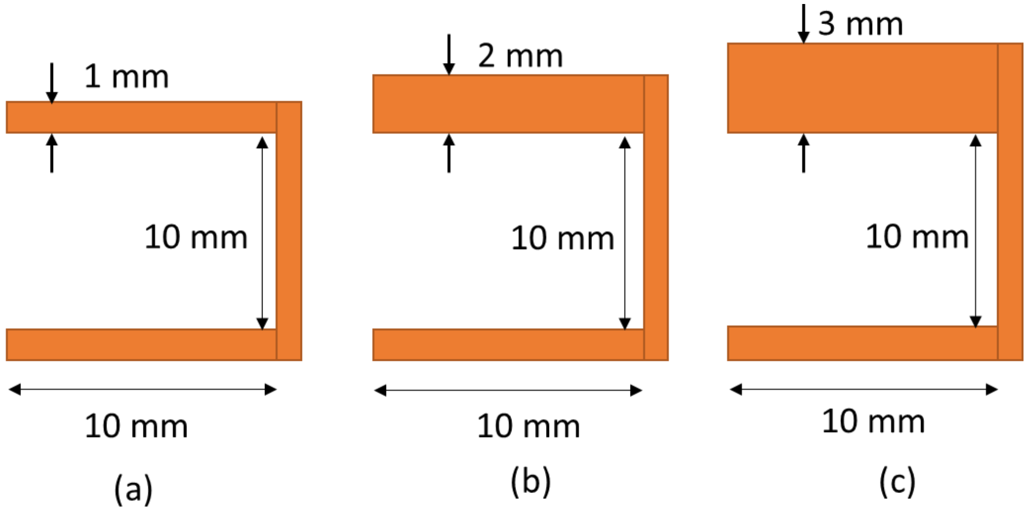


Fig. 4.3 Different U-shape configurations

$$L_{eff}(a) = 3 \times L_{w=1mm} \tag{4.2}$$

$$L_{eff}(b) = 2 \times L_{w=1mm} + L_{w=2mm} \tag{4.3}$$

$$L_{eff}(c) = 2 \times L_{w=1mm} + L_{w=3mm} \tag{4.4}$$

Table 4-2 illustrates the comparison for these 3 cases using both the Ansys Q3D and the Laplacian Matrix method. The results have shown that there is up to 22% error versus Ansys Q3D using the Laplacian Matrix to approximate the loop impedance result. There are two main sources of errors. The first one is that the Laplacian Matrix does not consider mutual inductance among conductors. This mutual inductance will reduce the overall loop inductance result if there are currents flowing in opposite directions. The second source of error is due to the very coarse discretization of the Symbolic Layout and Laplacian Matrix combination. Due to this, the proximity effect among currents flowing in opposite directions is also ignored. The inductance result in Table 4-2 illustrates the importance of the proximity effect. In the case of Ansys Q3D, there are very small changes in the inductance value among the three cases (0.9 nH). This is because at high frequency, due to the proximity effect, most of the currents are crowded at the internal of the U-Shape leading to similar loop-inductance result. In the case of Laplacian Matrix method, the current is assumed to be uniformly distributed within each trace. This in turn creates a bigger loop-area leading to more variation in the loop-inductance among the three cases (1.8 nH).

The Laplacian Matrix method in [23], [25] gives acceptable results for loop-inductance estimation (within 10-20%) of the layout. However, due to the lack of mutual-inductance and proximity effect considerations, the method might overestimate the parasitic inductance results among different layout solutions. For example, the layout optimization algorithm gives the same electrical finesses for the three layout cases above if Ansys Q3D is used as an evaluator. However, in the case of Laplacian Matrix method, case (a) has a worsen electrical fitness. This in turn, would impact the distribution of the final solution space. This is discussed further in Chapter 5.

4.2 Modified Nodal Analysis (MNA) Solver

Section 4.1 shows that the consideration of mutual inductance has a 10-20 % impacts on the impedance extraction and gives a better insight during the layout optimization process. Thus, mutual inductance must be considered in the extraction to have the most accurate extraction and electrical insights of the layout. To consider the mutual inductance, an MNA solver has been implemented in PowerSynth. This solver is the core for both PEEC method and the Loop-based methods which are described in Section 4.3 and 4.4. The full MNA solver for both time and frequency domain has been described in details in the literature [61], [64]. However, for the parasitic extraction purpose, only a frequency domain solver is needed in PowerSynth. To make it easier for the readers to understand the parasitic extraction procedure, the implementation of the MNA solver is briefly described and explained in this thesis. The fundamentals of MNA are based on the concept of nodal and loops. The two main equations are:

$$\sum_{i}^{k} V_i = 0 \tag{4.5}$$

$$\sum_{i}^{k_{in}} I_{in} + \sum_{i}^{k_{out}} I_{out} = 0 \tag{4.6}$$

Equations (4.5) and (4.6) are known as the Kirchhoff's Voltage Law (KVL) and Kirchhoff's Current Law (KCL). The KVL states that the total sum of all voltages around any closed loop must be zero. The KCL states that the sum of all input currents and output currents at a circuit node must be zero. in. KCL and KVL are the two main equations that govern the MNA method. The

KCL and KVL equations can be written for each circuit component where the MNA matrices are formed to solve the voltage and current unknowns. The MNA formulation for both the left-hand side (LHS) and right-hand side (RHS) is described in equation (4.7) below:

$$\begin{bmatrix} \boldsymbol{G} & \boldsymbol{A} \\ \boldsymbol{A}^T & \boldsymbol{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{X_V} \\ \boldsymbol{X_I} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I_{in}} \\ \boldsymbol{V_{in}} \end{bmatrix} \tag{4.7}$$

The description for each of the matrices are as follows:

- Adjacency matrix (**A**) describes the connectivity of each current unknown
- Conductance matrix (**G**) stores the conductance values, mostly for the R and C elements.
- Impedance matrix (**D**) stores the impedance values, mostly for the L and M elements.
- The column vector **X** includes the unknown voltages vector $\boldsymbol{X_V}$ and unknown currents $\boldsymbol{X_I}$
- The RHS column vector stores the input values for current $\boldsymbol{I_i}$ and voltage $\boldsymbol{V_i}$ sources

While MNA can be used for both transient and frequency domain (Laplace domain), for the impedance extraction purpose, a frequency-domain formulation has been implemented in PowerSynth. Some selected circuit elements are included in this solver, these include resistance, inductance, capacitance, independent voltage source, and independent current source. The LHS and right-hand side (RHS) circuit stamps for each different element are described in Appendix B of the dissertation.

4.3 Partial Element Equivalent Circuit (PEEC) for MCPM Optimization

The new layout engine in PowerSynth has allowed the tool to handle most of the 2D, 2.5D, and 3D MCPM layouts. These include layouts with planar traces where the currents flow in multiple directions or traces in multiple layers. This requires the tool to capture the proximity effect among the complex current flows within the layout. For this reason, the Laplacian Matrix solver cannot handle these layouts set accurately. Therefore, a PEEC solver has been implemented to handle most 2D and 2.5D cases in PowerSynth.

PEEC method has been first introduced by Dr. Albert Ruehli from the IBM research center [32], [60]. This method applies the circuit techniques to solve for electromagnetic problem. A full-PEEC implementation can solve Differential Equations (DE) in both frequency domain and time domain and approximate for the electromagnetic solution. The most crucial advantage of PEEC is the application for most complex geometries for both Manhattan and Non-Manhattan styles [65]. The Manhattan structure can be handled by using the equations and regression model described in Chapter 2. It is possible to handle Non-Manhattan style layout with PEEC as described in [65]. In this scenario, since there are no existing equations for the non-orthogonal pieces, numerical integration needs to be done for the self and partial elements evaluation. This would be more computationally expensive to implement in comparison with the Manhattan style implementation. At the moment, since the PoweSynth layout engine only handles Manhattan style layouts, only Manhattan geometries are considered in the PowerSynth-PEEC implementation.

There are three main steps in the PEEC method: layout discretization, impedance calculation, and circuit solving. In the first step, the geometry is discretized into multiple nodes and edges, where a capacitive cell (Fig. 4.4 (a)) is formed at every node and a resistive/inductive cell is connected between every two nodes (Fig. 4.4 (b)). In this meshing cell configuration, according to

[60] the R and L elements are usually combined together in a single branch. This combination reduces the number of total unknown voltage nodes in the MNA formulation and improves the computation efficiency.

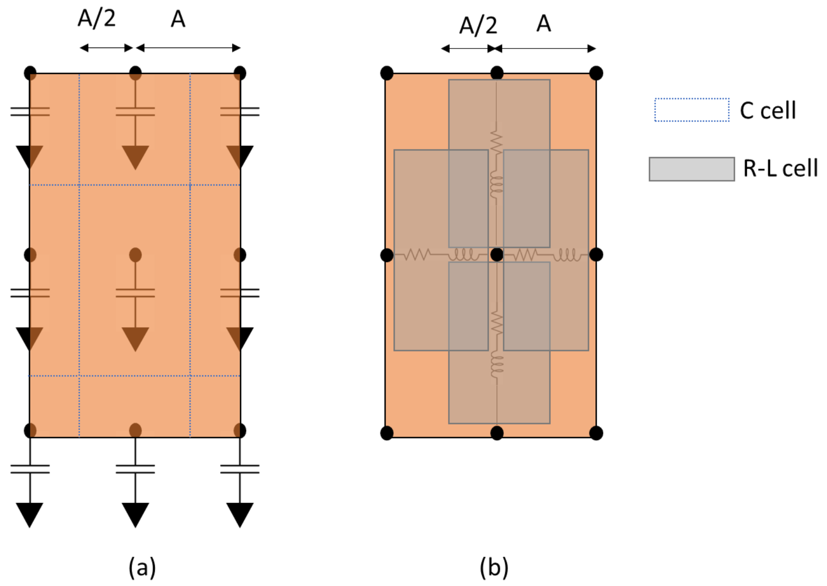


Fig. 4.4 Different meshing cell types in PEEC (a) Capacitor (b) Resistor and Inductor

The meshing process is a very crucial step in PEEC to achieve the closest solution to FEA. In PowerSynth, this process is performed for each trace island on the same layer. The meshing process is described step-by-step in Fig. 4.5 as follows:

a) The mesh algorithm takes the 2D copper trace geometry from layout engine.

b) The trace copper is placed inside a grid; each piece is marked with a matrix location. The pieces are distinguished by “blank” (grey) and “cell” (orange) types.

c) Based on the number of “blank” and “cell” neighbors, each trace corner can be defined as “convex” (blue) or “concave” (red).

d) Using the “concave” and “convex” corners as anchor points, additional mesh nodes are added to model the skin depth effect near the edges of the copper. A parameter can be used to define the number of additional mesh nodes.

e) All of the nodes on the trace level are connected. A new grid is formed with additional “cell” type object

f) The device center is used to find the parent cell for the trace. The parent cell is further meshed to connect the device to the trace-level grid.

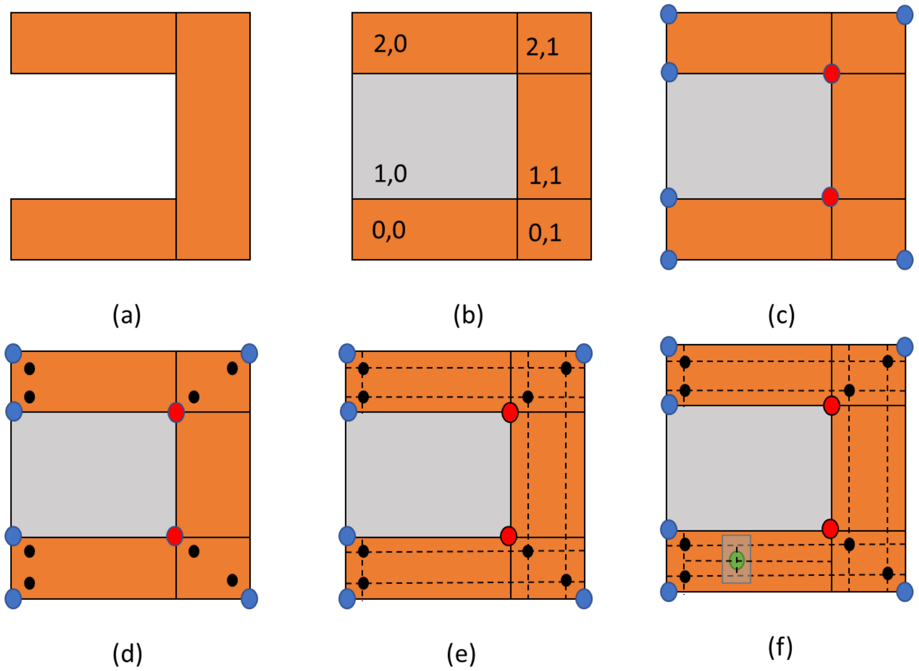


Fig. 4.5 Meshing process of MCPM (a) input layout (b) matrix-based grid (c) detect corner types (d) adding near corner mesh points (e) trace level meshing (f) device placement

Here, the step from (e) to (f) is why PowerSynth approximates the solution faster than other PEEC implementation for optimization such as [36]. Opposed to the grid based mesh in [36] this method only add a few hierarchical mesh edges locally based on the hierarchical approach developed in this work which has been discussed in [66]. However, the meshing algorithm in [66] is based on a uniform mesh and has not captured the frequency dependent effect. The optimization study in Chapter 5 (Section 5.1) shows the improvement using this meshing algorithm on the solution space.

Once a geometry is appropriately meshed into smaller elements, the self and mutual among the traces can be calculated using the equations mentioned in Chapter 2. Finally, MNA solver is applied to solve for all unknown voltages and currents in the mesh. Some of the extraction results are verified versus experiment in Chapter 5 where the layout optimization is performed.

The PEEC approach has many advantages. This method is general and can be applied to most geometries as long as there are available partial element self and mutual equations for the MNA evaluation. However, there are some limitations with this method. In some scenarios, when the number of partial elements is high the computation time for the self and mutual evaluation increases. Moreover, the MNA matrices sizes also expand. This reduces the overall efficiency of the computation.

4.4 Loop-based Extraction Method for MCPM Optimization

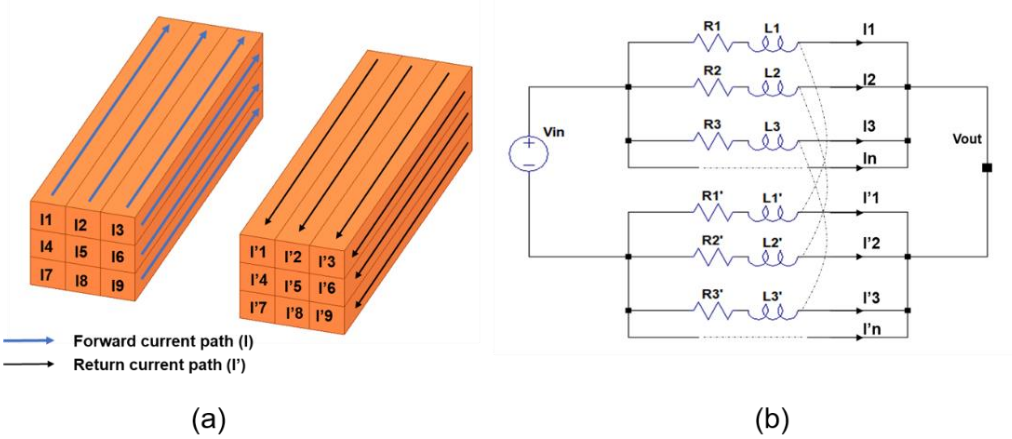


Fig. 4.6 (a) forward and return currents for parallel traces (b) Equivalent circuit representation

While PEEC is general for most geometries, as discussed in the last section, the computational efficiency can reduce significantly when the number of partial elements increases. In the scenario where the layout geometry mainly consists of parallel and straight elements, the current return path for each loop can be easily defined. In this case, the Loop-based technique [67]–[71] in Very-Large-Scale-Integration (VLSI) can be applied to the MCPM layout. This approach has been proven to be efficient and accurate for 3D muti-conductors, and on-chip layout interconnects extraction in VLSI. This method allows a more efficient meshing approach for the conductor traces (Fig. 4.6), where the self and mutual elements can be evaluated more efficiently.

Furthermore, this method significantly reduces the number of partial elements in the MNA solving step by dividing the layout into multiple loops. Here, the divide-and-conquer strategy can be utilized, and the loop-evaluations can be performed using multiple processors. This multiprocessing approach allows further improvement in computational efficiency.

Most importantly, the reduced extracted netlist from this model is more efficient for post-layout circuit simulation. Fig. 4.7 shows the layout engine to loop generalization and the bundle generation algorithms. The main difference between the PEEC implementation and the loop-based model is that the PEEC method can determine any arbitrary current directions in a geometry. Conversely, for the loop-based method, the user needs to define the current direction in the loop. This predefined current direction is also the advantage of the loop-based method because it takes less computational effort to solve for each current direction as opposed to PEEC. In PowerSynth, to use the loop-based approach, an extra input file is needed to define the copper piece as horizontal "H" or vertical "V". Here, "H" and "V" are the general current direction of each conductor defined by the user.

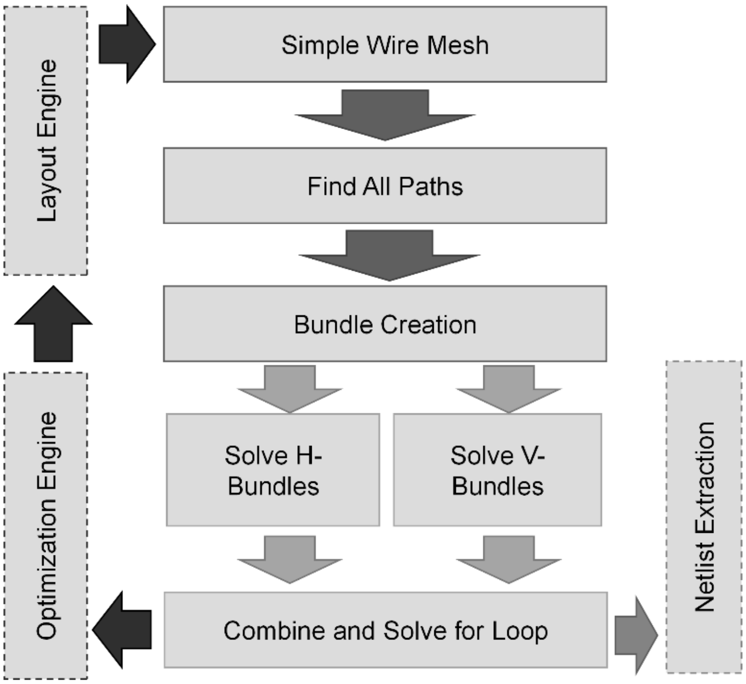


Fig. 4.7 Layout engine to loop-based model conversion

```
Algorithm 1: Digraph formation
Input: Initial Wire Mesh Graph (G)
Output: Digraph from source to sink (D)

paths = depth_first_search(source,sink)

for i from 0 to length_of(paths) do:
    # Get the array of nodes and paths
    p = paths[i]
    # Loops through all node from digraph
    for j from 0 to length_of(p) - 1 do:
        if not (exists_path(D, p[j], p[j+1]) then:
            add_edge(D, p[j], p[j+1])
```

The layout to electrical API converts the "H," and "V" traces into a wire mesh using trace-edge, device, and bondwire landing locations. These locations are represented as nodes where an edge is formed for every trace-trace or trace-bondwire intersection. Next, the depth-first search algorithm [72] available in the Python library has been used to find all existing paths from the source to sink terminals. Using the path information, a directed graph can be formed (Algorithm 1). This algorithm first transforms the wire mesh into a directed graph, as shown in Fig. 4.8. Then, in the bundle creation step, the directed graph is separated into two groups, namely horizontal and vertical bundles. Depending on the current direction of the piece in the directed graph, the forward and return paths can be defined for each bundle.

Each bundle can be further meshed into uniform parallel elements as shown in Fig. 4.6 (a). The advantage of this uniform meshing is that it would reduce the computational effort for both self and mutual impedances. For instance, as seen in Fig. 4.6 (a) the mutual inductance between elements 1 and 5 is the same as the mutual inductance between 3 and 5. Here, a programming technique known as dynamic programming can be applied to solve the problem much faster. A simple hash-table can be used to store the mutual inductance solutions for pairs of elements sharing the same relative location in space. The same evaluation process using equations (2.23)-(2.28) can be applied to find the loop-impedance of each bundle. Once the impedance of each bundle is calculated, they are stored in a matrix **A** below:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{Z_{loop1}} & \cdots & \boldsymbol{Z_{loop1,k}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{Z_{loopk,1}} & \cdots & \boldsymbol{Z_{loopk}} \end{bmatrix} \tag{4.8}$$

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

$$\boldsymbol{AI_i} = \boldsymbol{V_i} \tag{4.9}$$

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.

By dividing the layout into multiple bundles, the total complexity for this algorithm is O(N× M^2) 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.

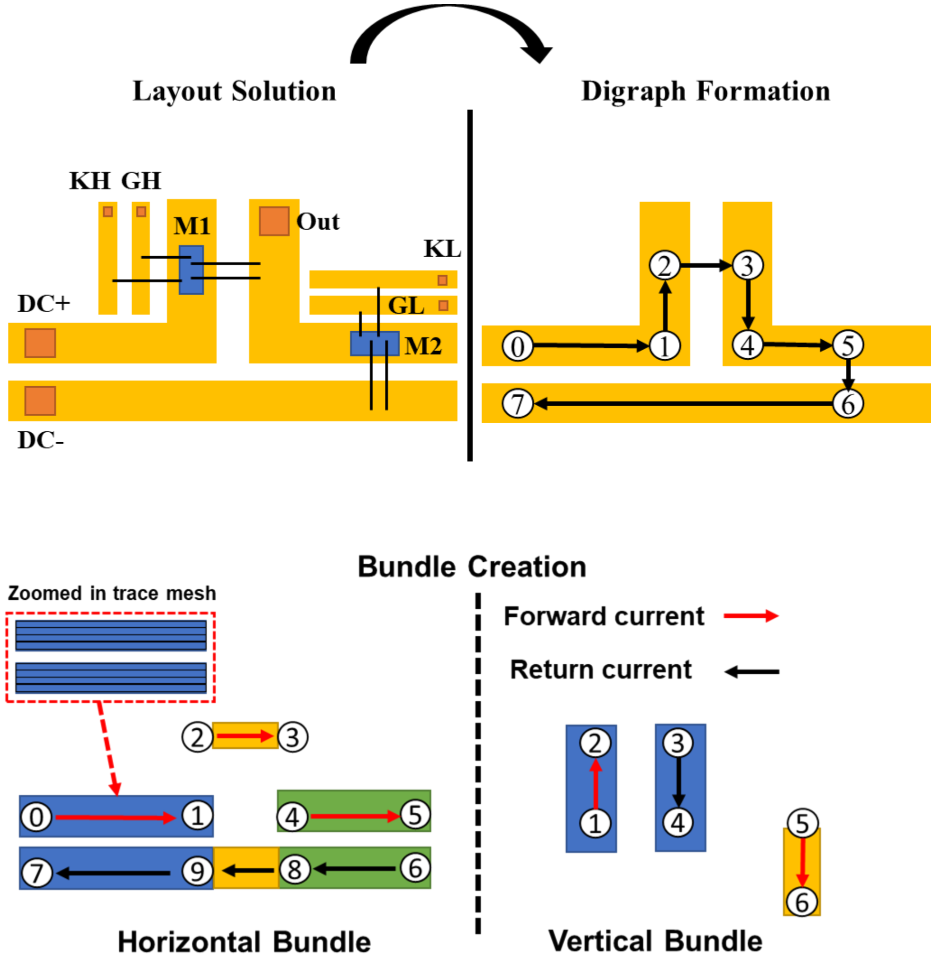


Fig. 4.8 Layout to bundles transformation process

The main limitation of this loop-based method is that it only works with layouts where most of the current directions are horizontal or vertical. This is quite true in many layout cases where there are multiple devices in parallel. Also, this method might be applicable for some 3D MCPM layout cases because they usually have fewer planar-type traces. However, for 2D and 2.5D layouts with many planar traces, the current can travel in unpredictable directions in a plane. The PEEC implementation in Section 4.3 is a better choice in these scenarios. Some of the validations and optimization results of this method are discussed in Chapter 5.

4.5 Netlist Extraction for the Power Loop.

Even with the State of the Art (SOA) FEA tools such as Ansys Q3D, the netlist extraction procedure for an MCPM layout is challenging. This is because the Ansys Q3D tool is built for general geometries cases and does not focus on the MCPM layout problem itself. In the literature, many works have used the Ansys Q3D tool to perform parasitic extraction of the MCPM layout [73]–[75]. There are generally two ways to obtain a lumped netlist for an MCPM. In the literature's first and most common approach (Fig. 4.9), the circuit designer needs to connect all components in the same current conducting loop (DC-DC) and extract the total loop impedance. Here the device is usually modeled as a conductor to form the connection. From Ansys Q3D, the designer obtains a single loop impedance value. Finally, to perform the circuit simulation, the designer needs to manually divide the loop value into multiple smaller elements to distribute them along the conduction path (DC-DC). This method ensures that the DC-DC loop value is the same between the circuit simulation and the FEA extraction result. However, the designer needs to guess the values for the lumped elements. This task can be possible with a single device per switching position, as seen in Fig. 4.9. However, this task is very challenging for a more complicated layout with multiple devices.

The second netlist extraction approach is based on the multiport extraction technique taught in many fundamental circuit courses. First of all, this task is very tedious and time-consuming. The designer has to draw the 3D objects for each port connection manually. From here, the user needs to define which surface on the object is the current injection for a port, where the user must define each port as “source” or “sink” types to predefine the current loop directions. Tools such as Ansys Q3D only allows multi-sources and single-sink problem. Because of this, it is challenging to define the correct source-sink terminals for each net. However, once these source-sink terminals are defined, the designer obtains an N×N matrix of lumped elements, where N is the total of branches. If all of these branches are defined correctly, the netlist obtained from this multiport technique contains both the self and mutual impedances. However, as discussed in [73], even with this feature, the sum of the branch elements in the same loop is not equal to the DC-DC value. Moreover, the multiport extraction algorithm requires the tool to solve the problem in a sequential manner. This task is time consuming and not preferred for the layout optimization process. The work [36] has used this multiport technique for multibranch extraction in the layout optimization process. However, the performance of the model in [36] was not clearly mentioned.

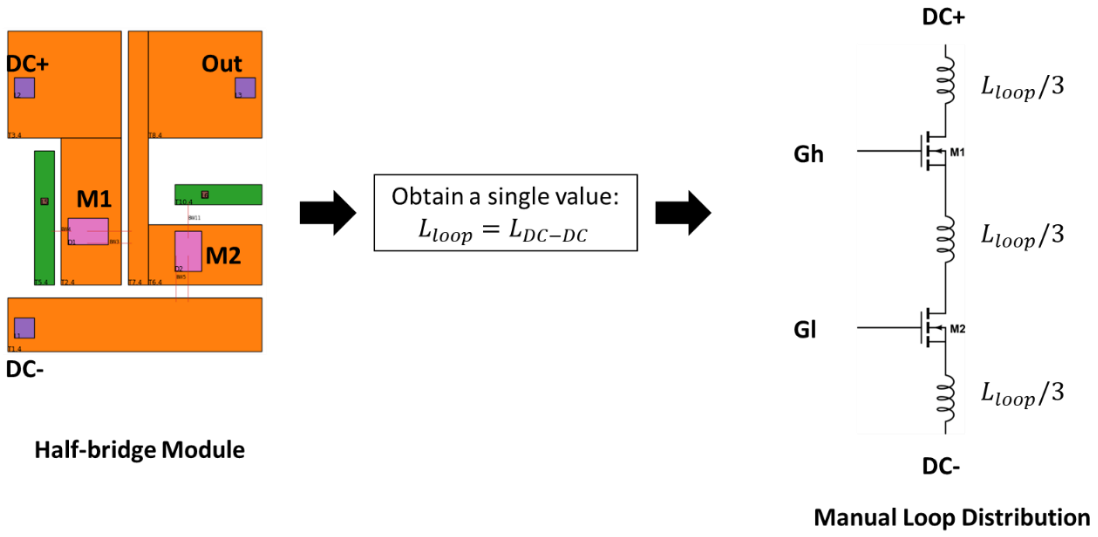


Fig. 4.9 The manual netlist extraction method

Combining the ideas from both of the previously mentioned netlist extraction methods, a different approach has been developed in PowerSynth. This is possible thanks to the unique setup of the device states as mentioned in Chapter 3. Using the virtual connection between the drain and source of each device and the MNA solver implemented in PowerSynth, the current through each device during the ‘ON’ state can be probed during the DC-DC loop extraction. For a half-bridge circuit, the lumped element of each device in the high and low switching positions can be obtained using the equations below:

For high side:

$$L_{DHS,i} = \frac{Im\left(\frac{\left(V(DC+) - V\left(M_{i_{Drain}}\right)\right)}{I_{DS}(M_i)}\right)}{2\pi f} \tag{4.10}$$

$$L_{SHS,i} = \frac{Im\left(\frac{\left(V\left(M_{i_{Source}}\right) - V(Out)\right)}{I_{DS}(M_i)}\right)}{2\pi f} \tag{4.11}$$

For low side:

$$L_{DLS,i} = \frac{Im\left(\frac{\left(V(Out) - V\left(M_{i_{Drain}}\right)\right)}{I_{DS}(M_i)}\right)}{2\pi f} \tag{4.12}$$

$$L_{SLS,i} = \frac{Im\left(\frac{\left(V\left(M_{i_{Source}}\right) - V(DC-)\right)}{I_{DS}(M_i)}\right)}{2\pi f} \tag{4.13}$$

All of the net voltages and the drain-source current for the equations (4.10)-(4.13) can be obtained by the MNA evaluation implemented in PowerSynth. This approach ensures that the obtained equivalent circuit has exact same loop-value with the extraction. Simply because of the KCL relationship in a half-bridge circuit:

$$I_{loop} = \sum I_{DS}(M_{HSi}) = \sum I_{DS}(M_{LSi}) \tag{4.14}$$

Equation 4.14 shows the KCL relationship among the total loop current, drain-source currents of the high-side MOSFET, and drain-source currents of the low-side MOSFET. This can be simply explained by: the sum of the drain-source currents of all parallel devices of a switching position has to be equal to the total loop current. Table 4.3 below shows the comparison of the manual, multiport, and this netlist extraction approaches. The complexity comparison is based on PEEC for M is the number partial elements in the matrix and N is the number of devices.

One main advantage of this this method is that the source-side parasitic inductance of each device can be correctly obtained. These source-side values are important for dynamic circuit optimization such as current-balancing [36]. The limitation of this approach is the lack of gate inductance extraction. Here, a hybrid combination between the multiport and this approach can be performed to extract the missing gate loop values. Hence, in PowerSynth, both the multiport and total loop current methods have been implemented. This allows some flexibility in the case the user wants to dictate the elements for the output netlist.

Table 4.3 Comparison among different netlist extraction approaches

Methods	Number of elements	Element type	Complexity
Manual	1	L_{loop}	O(M^2)
Multiport	3×N	L_D, L_S, L_G	3N×O(M^2)
This method	2×N	L_D, L_S	O(M^2)

Chapter 5. Experimental and Optimization Results for Parasitic Extraction

In this chapter, the Partial Element Equivalent Circuit (PEEC) and Loop-based approaches in combined with the regression model considering eddy current impact in Chapter 2 are experimentally verified in both the frequency and transient domain. The extraction results using Loop-based and PEEC are also compared versus the experiment for a fabricated 3D MCPM layout. For all of the validation in this chapter, the selected circuit type is a half-bridge circuit which is the smallest unit cell for many power electronics applications.

In section 5.1, two half-bridge modules have been designed and fabricated to verified against the extraction results from PowerSynth. The impedance measurement using Vector Network Analyzer (VNA) shows that the model is accurate for a broad range of frequency. The Double Pulse Test (DPT) experiment has also been performed where the time domain circuit simulation is used to verify the test results.

In section 5.2 the layout optimization has been performed. Two different solution spaces using both PEEC and Laplacian Matrix are compared to show the impact of mutual inductance consideration in layout optimization.

In section 5.3 the loop-based model is experimentally verified, and an impedance analyzer has been used to measure and verify the loop-inductance results

Section 5.4 shows the impact of backside copper consideration on the layout optimization result and solution space.

Section 5.5 shows extraction some results using the combination of the PEEC and Loop-based method and regression model for a 3D MCPM layout.

5.1 Experimental Validation for PEEC method with Regression model

A. Validation of the PEEC method on PCB and pervious hardware validation DUT

In the very first attempt on PowerSynth v1[25], the Laplacian Matrix method has been combined with the regression-based model. For the hardware validated layout from this work, the 90-degree-corner-correction regression model [16] has been applied along with the self-impedance model to achieve accurate result versus the impedance measurement. However, this 90-degree-corner-correction is only applicable for this selected layout case to overcome the inaccuracy of Laplacian Matrix method. In fact, most of the inaccuracy from the Laplacian Matrix approach are coming from the lack of mutual inductance consideration and uniform trace mesh as mentioned in Chapter 4 (Section 4.1). In [66], the first time this approach was implemented in PowerSynth, the measurement data for the DUT in [25] has been re-validate to verify the accuracy of the model. In this work, three different validation has been performed to compare the current density extraction versus Ansys HFSS and the extraction versus measurement in [25].

In Fig. 5.1 to verify the PEEC implementation, a 10mm × 10 mm U-Shaped structure has been designed in PowerSynth and Ansys HFSS. Here, the current density of the U-shaped structure has

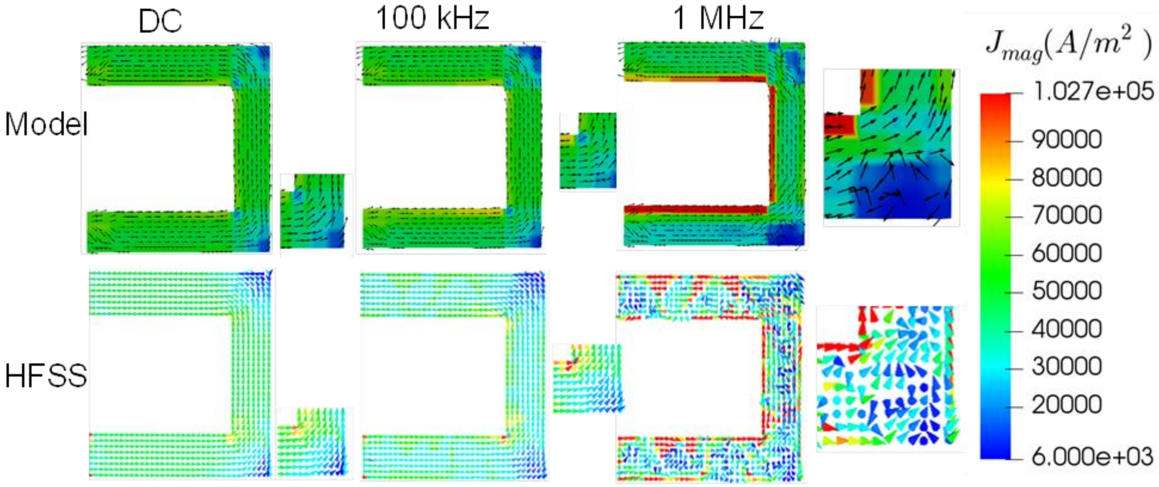


Fig. 5.1 Current Density verification for increasing frequency

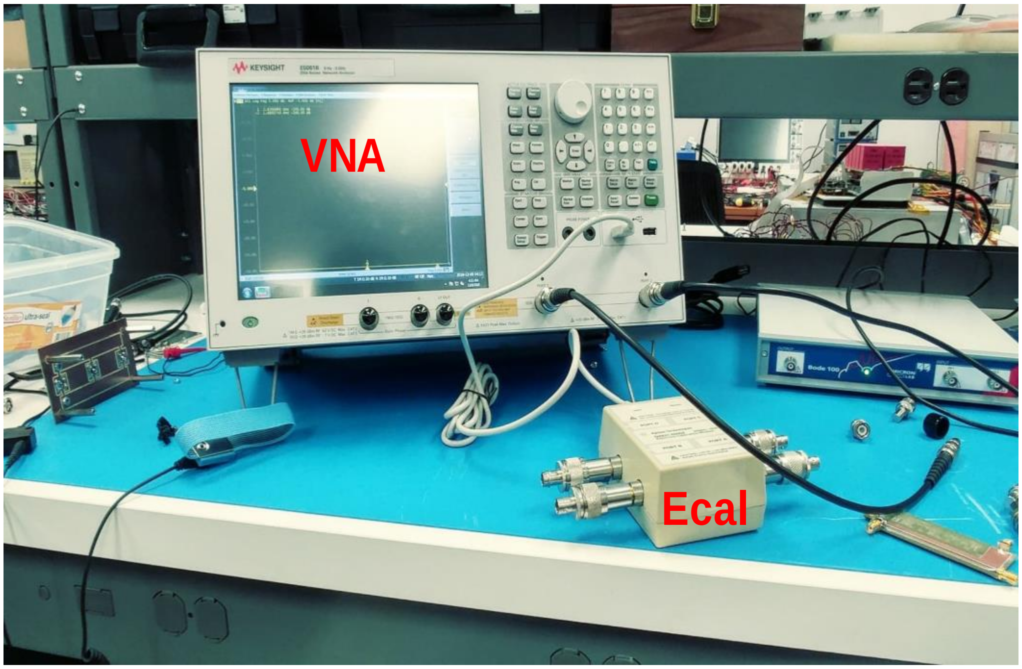


Fig. 5.2 Vector Network Analyzer for S-Parameter measuring

been plotted to verify the PEEC for different frequencies from 1Hz (DC) to 1MHz. The PEEC current density is first extracted to a matrix form and plotted using the ParaView tool. These results have shown a very close approximation between the PEEC implementation versus Ansys HFSS. Most importantly, these results have shown that the PEEC model can capture the proximity effect near the 90-degree-corner of the shape accurately. At DC, most of the current are uniformly distributed. However, as frequency increased, the current trying to find the closest path to close the loop. As such, the current density increases at the 90-degree-corner.

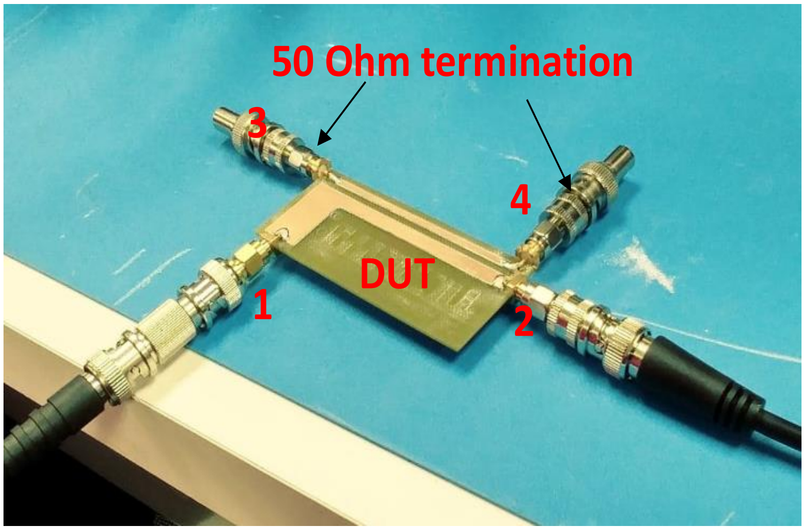


Fig. 5.3 PCB structure for 4 -port measurement

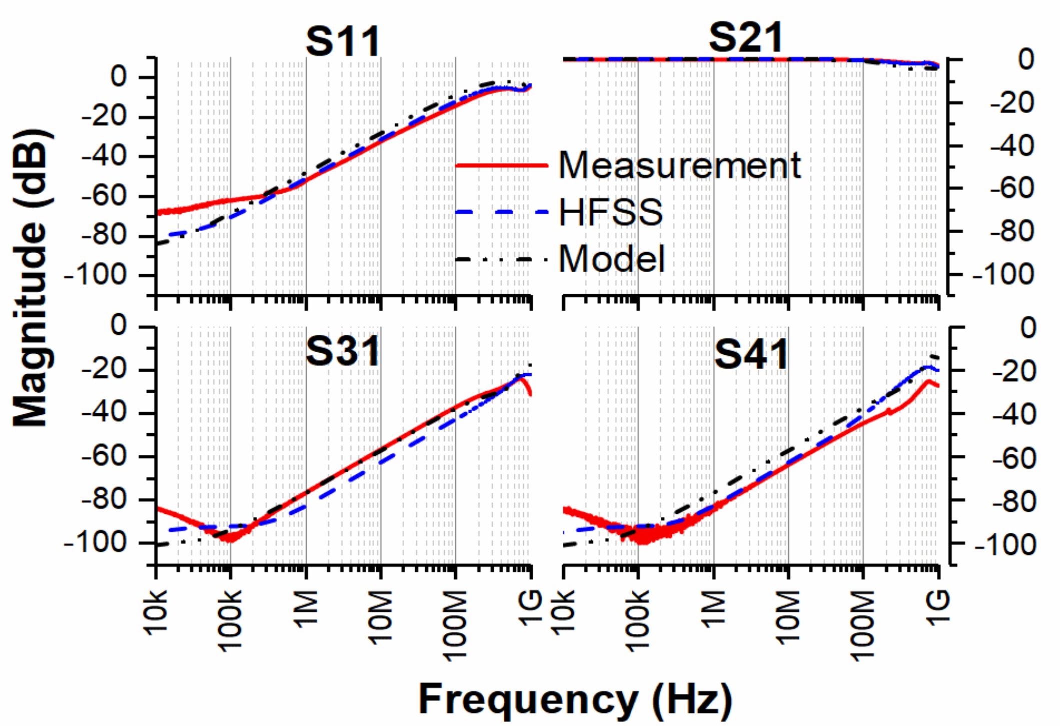


Fig. 5.4 S-parameter comparison of the 4-port DUT

The model is then further experimentally verified versus S-Parameters measurements using the Vector Network Analyzer (VNA) setup in Fig. 5.2. Here, a simple Printed Circuit Board (PCB) has been made with two separate current loops. SMA connectors are soldered to the four locations as shown in Fig. 5.3. The sequentially 2-port measurement has been performed to measure the port-to-port signal where the other ports are terminated using the 50-Ohm termination. The full PEEC matrix with few thousands of elements is dumped into a netlist where Synopsys HSPICE has been used to perform the S-parameter extraction. The same structure is simulated in Ansys HFSS. The S-Parameters results for the PEEC-HSPICE simulation, Ansys HFSS, and VNA measurements are shown in Fig. 5.4. The measurement results from this experiment have shown a good agreement between the model, HFSS simulation, and measurement. As can be seen from these results, there are some noises in the frequency range less than 100 kHz. This error is coming from the solder joint connection between SMA and the board. Otherwise, the model has shown good agreement with the measurement results and HFSS for frequency greater than 100kHz to 1MHz. Finally, the same PEEC model is verified versus the extraction results in [25]. In this first version of the model, the mesh algorithm is not optimized, and a uniform mesh structure has been used as seen in Fig. 5.5 below. Even so, the model shows a quite good agreement with the measurement in the previous validation attempt of PowerSynth.

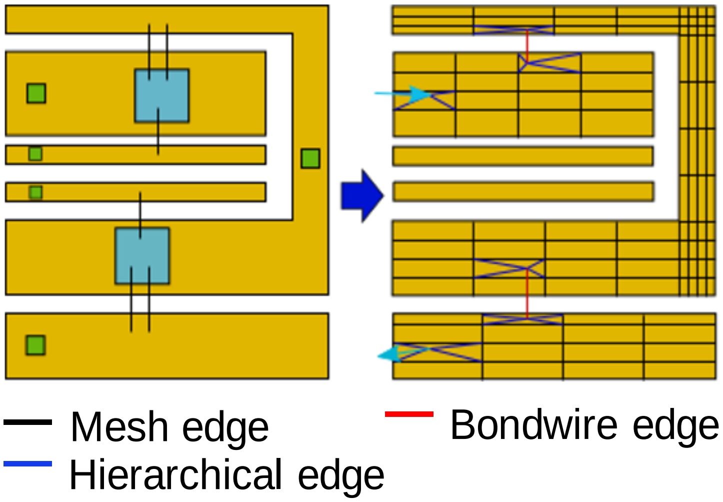
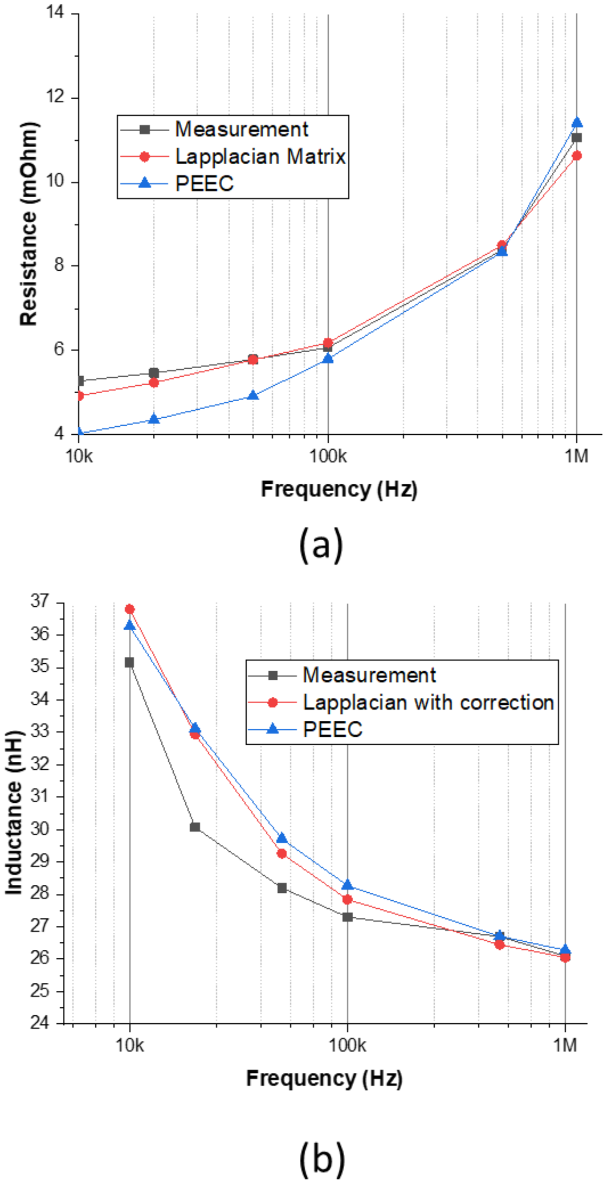


Fig. 5.5 The uniform mesh for the structure from the previous PowerSynth hardware validation DUT



(a)

(b)

Fig. 5.6 Validation of the PEEC method versus the previous PowerSynth hardware validation result. (a) Resistance (b) Inductance

B. Fabrication of the MCPM Design Under Test (DUTs)

In this section, more MCPM designs are fabricated to verify the accuracy of the PEEC model. Here, two test vehicles are first designed in PowerSynth, as shown in Fig. 5.7 and Fig. 5.8. These designs are then fabricated at the High-Density Electronics Center (HiDEC) of the University of Arkansas. From the 2D layout information provided by PowerSynth, the layout CAD files are automatically generated. In the next step, these designs are printed on a thin film for a chemical etching process. Since the purpose of the first design is for the S-Parameter measurement, no bare dies are attached on the DBC substrates. Hence, after the layout is chemically etched on the top side, the 12-mil aluminum wires are bonded to form two different current loops. Then, to form the interface between the design and the VNA (Fig. 5.2), four SMA connectors are soldered at the 1-4 locations denoted in Fig. 5.7.

In the second design (Fig. 5.8), once the layout is chemically etched, the 1.2 kV SiC devices (labeled as M1 to M4) from CREE (CPM2-1200-0040B) are attached. Because the solder attach material of the devices have the highest melting temperature, a fixture is first design to ensure stable locations for each device. Then the DUT with the fixture is placed inside an oven at @C. Once the devices are successfully attached, the 12-mil aluminum bond wires are bonded at the

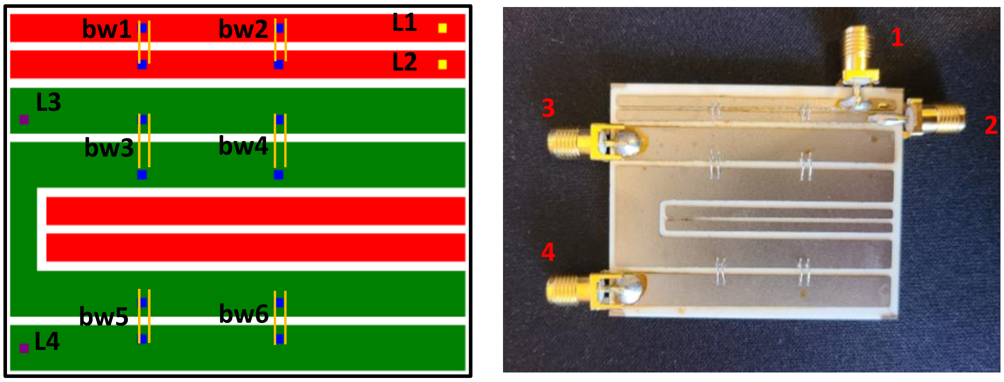


Fig. 5.7 PowerSynth layout (left) and DUT (right) for S-Parameter measurement

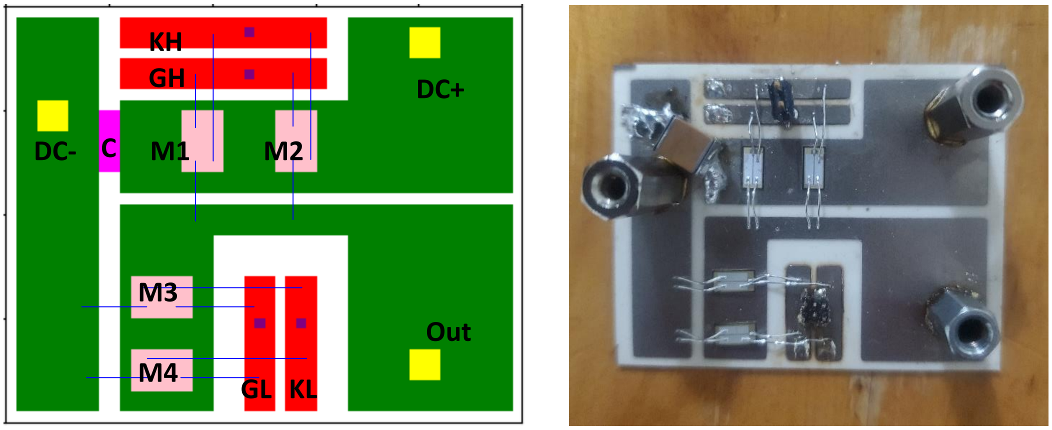


Fig. 5.8 Generated layout from PowerSynth (left) DUT for DPT measurement (right)

designated locations for both source and gate pins. Finally, the hex terminals (labeled as DC+, DC- and OUT) and the decoupling capacitor (labeled as C) are attached.

C. S-Parameters Extraction

For the layout in Fig. 5.7, an element-by-element distributed netlist with all R, L, C, and M is generated from the PEEC matrix. The PEEC subcircuit is then imported into the Keysight ADS circuit simulator where the leads and device pins are exposed as input terminals for the netlist. Using netlist to S-parameter conversion feature from Keysight ADS, the S-parameter simulation was run to extract the S-parameter results for the frequency range between 100 kHz to 100 MHz. The S-parameter results have been extracted by connecting a 50-ohm port to each terminal.

To verify the extracted results versus Finite Element Method (FEM), a 3D structure for the DUT has been designed in Ansys HFSS. Here, S-parameter can also be extracted to verify the accuracy of the extracted netlist from the PEEC model. Fig. 5.2 shows the measurement setup for S-parameter measurement using a Keysight E5061B VNA. To ensure the most accurate measurement, a Keysight ECAL N4431B is first used to perform two-port calibration for each port-port pair (1-2, 1-3, 1-4, 2-3, 2-3, and 3-4). During the measurement, each port is excited with a signal while the other ports terminated with individual 50 Ω loads. Six measurements for the port-port pair are carried out to collect all S-parameter data from 100 kHz to 100 MHz.

Fig. 5.9 illustrates some of the S-parameter for the comparison among the HFSS model, extracted PEEC netlist, and measurement results. Overall, the results show a good agreement among three sets of data in the frequency range from 100 kHz to 100 MHz. It is worth noticing that, while it is possible to simulate for a broader frequency band, the results for the lower frequency range (less than 100 kHz) are quite noisy due to the extra resistance from the solder joints. For the frequency range greater than 100 MHz, the skin-depth value gets much smaller which in-turn requires a larger number of mesh elements. Aside from the required large number of mesh elements in this high frequency region, radiated and coupling capacitance have a more significant effect on the S-parameter results. For the purpose of mainly validate R and L in the conducted EMI region, the selected frequency range is quite sufficient. From these results, it can be confirmed that the PEEC model developed in PowerSynth is stable and applicable for large frequency range.

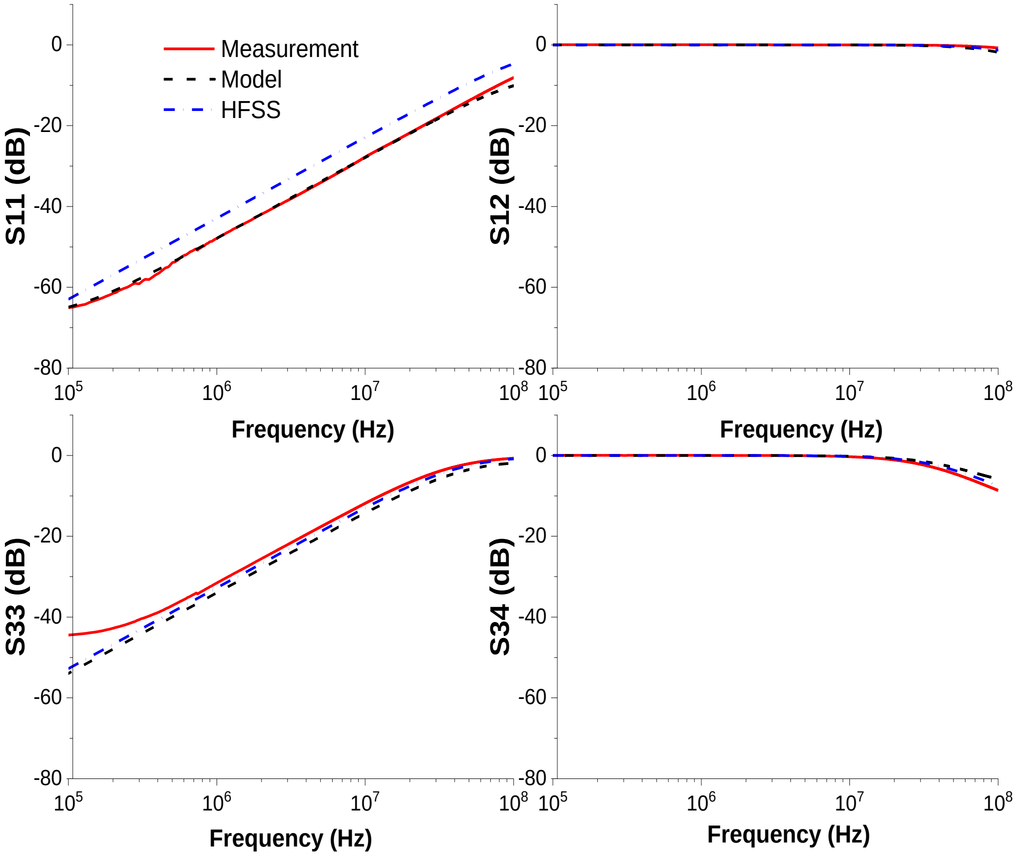


Fig. 5.9 S-parameter measurement versus model extraction and HFSS

D. Double Pulse Test Results Versus Simulation

Fig. 5.10 shows the experimental setup for the DPT test. A simple PCB board fabricated on an FR4 substrate that has been designed to interface between the test vehicle power terminals and the laminated busbar. Since the fabricated design does not have encapsulating gel, the test vehicle is tested at the rating of 400 V/15 A for safety considerations. The load inductance is designed to have a value of 131 μH. Due to the parasitic inductance introduced by the test setup, a decoupling capacitor *C* with the value of 1.27 μF is added between DC+ and DC- The high-side devices are turned off while the low-side devices are switched with the voltage pulse between -5 V and 20 V. Jumper wires are used to connect between the DSP board and the gate pins on the test vehicle. Due to the high parasitic inductance introduced by these wires, the gate signal is a bit noisy. Finally,

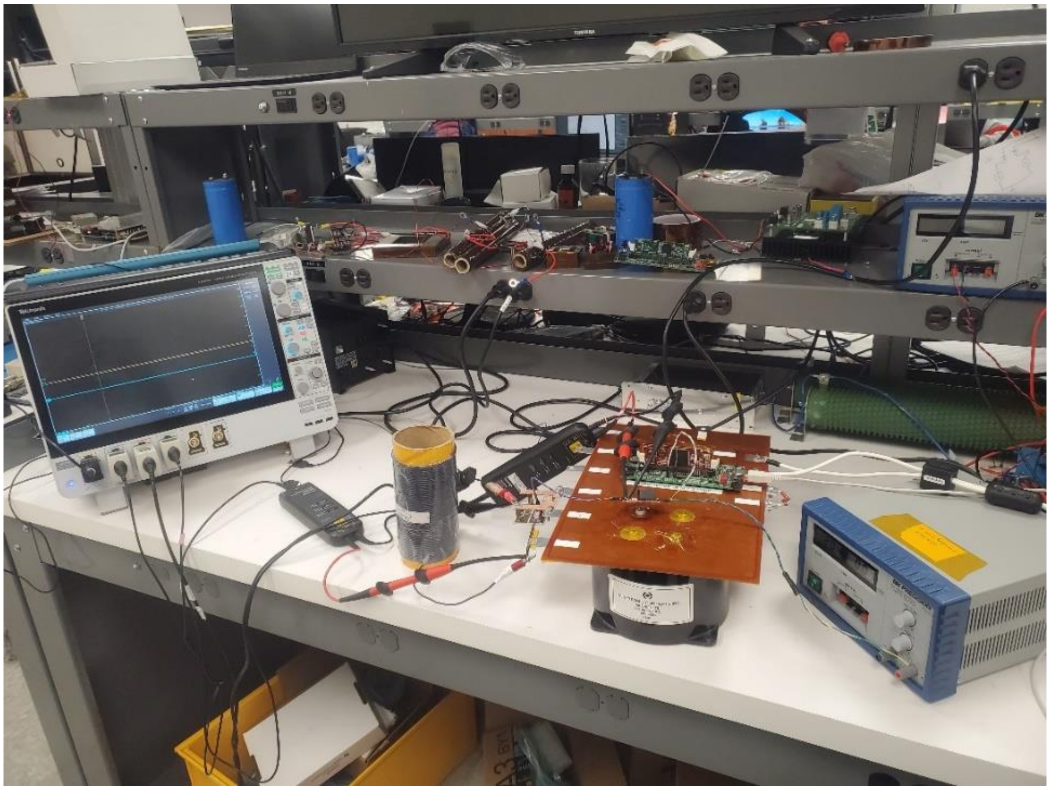


Fig. 5.10 Experimental setup for the DPT measurement

the V_{ds} and load current values of the low-side devices are measured to compare with simulation results.

E. Circuit Simulation Setup and Validation Results.

A lumped circuit netlist is extracted for the layout shown in Fig. 5.8. A double pulse test simulation is run in LTSPICE using this lumped sub-circuit extracted from the layout. The physics based MOSFET model from [76] has been used to obtain the best switching performance compared with measurement results. Fig. 5.11 shows the comparison among different simulation results versus the experimental for the $\mathrm{V_{ds}}$ data. As shown in the zoomed in figure, the simulated data using the distributed netlist is the most accurate in comparison with the measurement. Otherwise, the single lumped (non-distributed) and no-parasitic (baseline) shows inaccurate results versus the measurement.

However, there is a peak-peak mismatch between the simulation using the distributed netlist and the measurement. The main reason for this mismatched peak-to-peak value is the use of jumper wires to interface between the DSP board and the gate pins for the devices. These jumper wires have high parasitic inductance that led to a noisy gate signal during the turn on period of the device which resulted in the shift in peak overshoot voltage. Even so, the turn-off ringing frequency can be calculated from the measurement result to validate the loop inductance. This value is 89.43 MHz in the simulation, while the simulation result shows 86 MHz. Therefore, the extracted loop inductance from PowerSynth is only 3.84% different from that of the measurement.

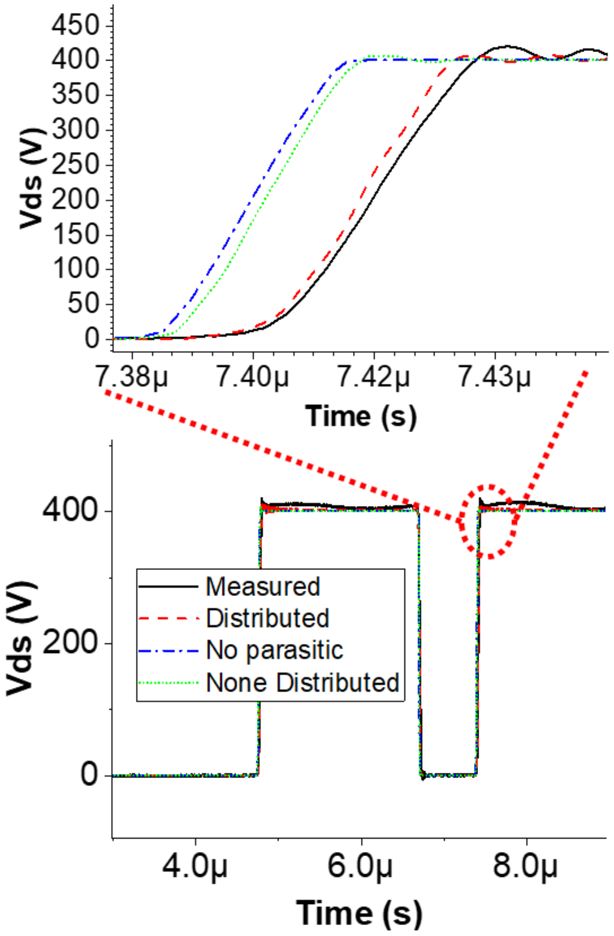


Fig. 5.11 Waveform comparison among different simulation approaches.

5.2 PEEC model efficiency and 2D layout optimization

In this section, extraction accuracy and speed are first compared with state-of-the-art finite element tools to show the model capability as a layout optimization evaluated function. The layout optimization results using this model are also compared with the previous work to show the key improvement in broadening the solution space.

A. Extraction and Accuracy Comparison versus Ansys Q3D

As a layout optimization routine is an iterative process, the efficiency of the evaluated function is one of the key elements for an efficient layout optimization. While a finite element technique such as Ansys Q3D provides an accurate solution for layout parasitic, it is not practical to use such a technique in an optimization routine due to the computationally expensive evaluation time. To illustrate the model capability for layout optimization, a runtime and accuracy comparison between the model and Q3D is shown in Table 5.1 below. The layout in Fig. 5.8 above is used as an example for this comparison where the loop inductance between L3 and L4 is evaluated using two different methods.

As shown in Table 5.1, the extracted loop inductance from the model is only 6.5% different from that of Ansys Q3D. The evaluation time using the model for this layout shows a speed improvement of 40 times on the same machine. Hence, this model allows the optimization algorithm to efficiently search for the solution space in a few minutes to hours instead of a few days.

B. Layout Optimization Example

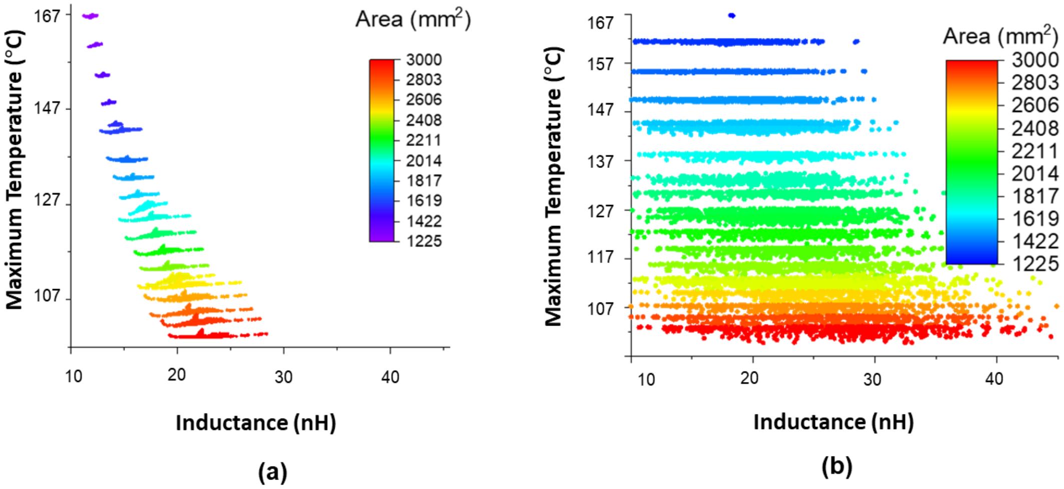


Fig. 5.12 Solution spaces using (a) uniform meshing approach (b) optimized meshing approach

Two layout optimization examples using the algorithm from [18] are run to show the enhancement of using this model versus the previous work in [16], [66]. The same layout, as shown in Fig. 5.8 without the decoupling capacitor has been used in this study. The optimization targets are the module steady-state thermal performance and parasitic loop inductance value from DC+ to DC-. To evaluate steady-state thermal performance, the fast and accurate thermal model which has been hardware validated in [25] has been used. This thermal model is used for both optimization runs to ensure a fair comparison between different electrical models. For the parasitic loop evaluation in Fig. 5.12 (a), the loop inductance is evaluated using the model in [66]. For the parasitic loop evaluation in Fig. 5.12 (b) the model has been upgraded using the frequency-dependent mesh algorithm in Chapter 4.

About 10,000 layout solutions are generated in each optimization run. The layout footprint area is swept between 1225 mm^2 to 3000 mm^2 to cover a broader solution space. Each dot in Fig. 5.12 shows a different possible layout solution. The two axes show the thermal and electrical performances in terms of maximum temperature (in K) and parasitic loop inductance (in nH). The two solution spaces show the same range in maximum temperature performance, which is reasonable since the thermal performance is proportional to the layout footprint area. The solution space in Fig. 5.12 (b) using this model shows some improvements, which are:

- A broader solution space in terms of parasitic inductance performance.
- A similar minimum parasitic inductance performance for all footprint areas (about 10 nH).

For different layout sizes, the optimized loop inductance is the same because the smallest loop can always be found in the bigger footprint area. This new solution space has shown that the new meshing algorithm allows the layout optimization algorithm to find a much larger solution space, giving the designer more choices between different electrothermal design aspects.

Table 5.1 PEEC with regression model versus ANSYS Q3D

	Loop Inductance (nH)	**Evaluation Time (s)**
Ansys Q3D	15.67	73
PEEC	16.7	1.8
Difference	6.5 %	40 ×

5.3 Experimental Validation for Loop-based Method with Regression Model

A. Experimental Measurement Setup and Validation of the PEEC Model

The same DUT for a 2D half-bridge layout is fabricated in Fig. 5.7 has been repurposed for the loop-based model validation. This layout has been designed for a half-bridge circuit with two devices for each switching position on a 39.5 × 52 mm^2 footprint. However, as previously mentioned, bond-wires connections are used to form shorts at the locations of the devices to form the loop between DC+ and DC-. To verify the loop-inductance result, the power loop between L3 and L4 in Fig. 5.7. The existing SMA connectors representing the DC+ and DC- locations on the DUT to interface with the impedance analyzer Fig. 5.14 (d).

The Keysight E4990A impedance analyzer has been used to measure many power electronics systems in the literature [77], [78] and has also been used in this work to verify the power loop inductance against the extraction results. From the datasheet, E4990A is capable of impedance measurement between 20 Hz and 30 MHz, however there is a limit in the impedance resolution making the impedance lower than 100 kHz inaccurate. Hence, the measured data from 20Hz to 100kHz is very noisy and has been omitted. illustrates the limitation in term of frequency range and impedance values from E4990A datasheet [79]. It is worth noticing that there are almost no changes in inductance for the frequency range greater than 10 MHz Therefore, the frequency range has been selected between 100 kHz to 10 MHz for this measurement.

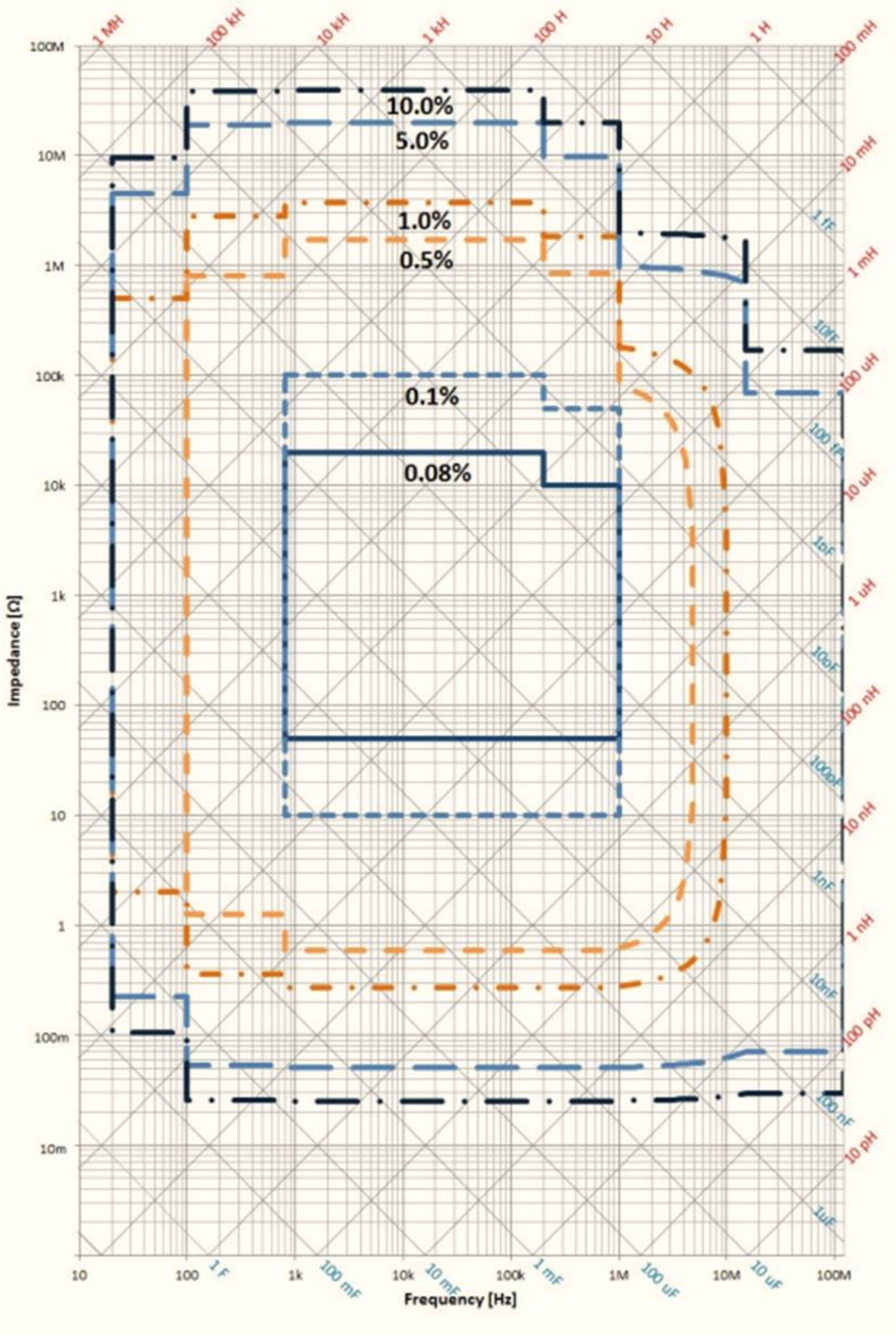


Fig. 5.13 Accuracy range versus frequency and impedance value for E4990A

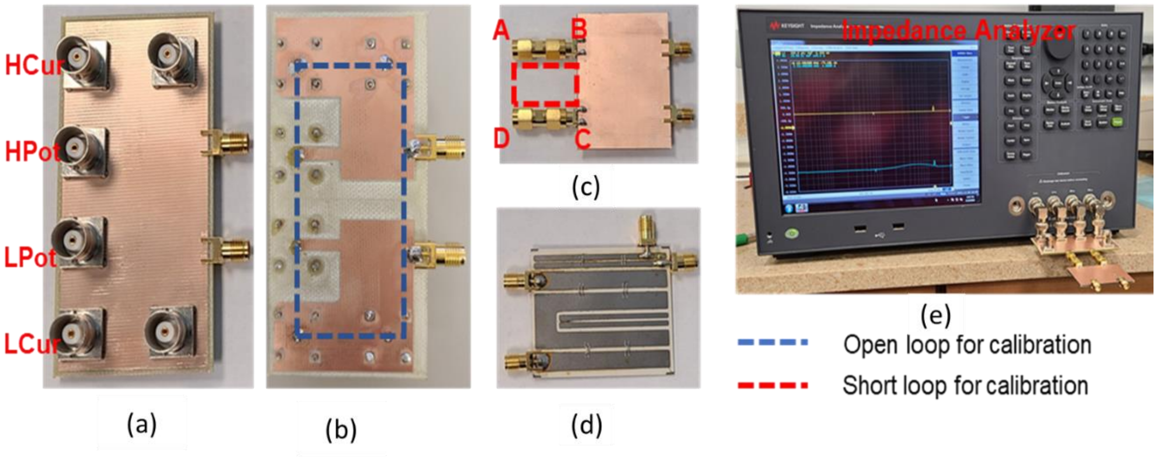
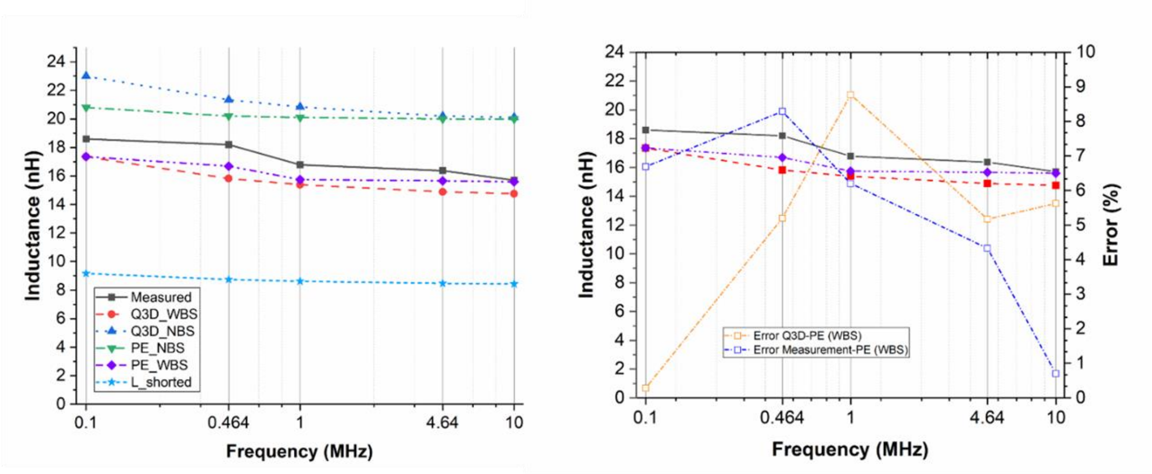


Fig. 5.14 (a) topside of custom-made PCB fixture (b) bottom side of PCB fixture (c) board for short calibration (d) design under test layout (e) impedance analyzer.

This impedance analyzer requires 4 BNC connections for high-low currents and high-low potential ports (HCur, LCur, HPot, and LPot in Fig. 5.14 (a) to interface between the DUT and the tool. Because of this, in this measurement, a custom-made PCB fixture has been designed to interface between the impedance analyzer BNC ports and the layout's SMA ports. To do this, two female-female SMA connectors have been used to interface between the DUT and the PCB board.

While the impedance analyzer provides a calibration kit to perform short and open calibration, the distance between the two terminal of this interface is too far, making inaccurate calibration for small DUT. For this reason, a small PCB piece with the exact same SMA connector's locations Fig. 5.14 (c) has been designed for short calibration. However, since MCPM layouts usually have inductance in the nH range, the impedance of the shorted fixture is quite significant to the MCPM loop inductance result. As seen in Fig. 5.14 (c), the ABCD loop (red dashed line) form a shorted path for the current. In this case, the simulated results show a loop inductance with the value between 9.1 to 8.4 nH for the selected frequency range. These values are quite significant and must be considered during the short-calibration process. Therefore, one needs to consider both measurement and simulation for the most accurate comparison and this additional inductance is extracted using Ansys Q3D for all selected frequency points (Fig. 5.6 a). This shorted inductance value can be defined during the short calibration state in the measurement.

The characterized models in the previous section are used along with a partial element methodology to validate against the measurement results. Here, the loop-based method from [20] has been used to extract the total loop-inductance for the lateral conduction paths in comparison to ANSYS Q3D results. Fig 5.6 a) shows the comparison among the partial elements extraction with and without backside consideration as well as the measurement results from the impedance analyzer. From these results, it can be concluded that extraction results from Partial Element with backside (PE-WBS) are in good agreement with Q3D with back side (Q3D-WBS) and measurement results. The maximum relative error is only 8.5% between measurement versus PE-WBS and 9% between Q3D-WBS versus PE-WBS while the minimum error is only 1% (Fig. 5.6 b).



(a) (b)

Fig. 5.15 (a) Comparison among different methods and measurement (b) Error comparison among measurement and models with backside consideration.

On the other hand, the model without backside (NBS) copper shows an overestimation of about 4-6 nH which is 30-40 % different in comparison to the measurement. These results have confirmed the importance of backside consideration in the extraction of trace inductance for MCPM layouts and shown that the method presented can model this effect with more than 90% accuracy.

B. Time and Memory Performance Comparison

By incorporating the backside impact in the regression model, the number of mesh elements for the MCPM structure can be significantly reduced. Hence, this regression model increases the performance of the partial element method and so the model is efficient for an optimization process. It has been demonstrated in [20] that the loop-based approach is more efficient than the State-of-the-art (SOTA) method in terms of speed and memory. However, the previous work in [20] did not include the backside impact.

Therefore, to illustrate the benefits of the method in this work, the extraction is performed on two more examples in Fig. 5.16 (b) and (c). Here, a comparison has been made between this method and Ansys Q3D (Table 5.2). This experiment has been run on an Ubuntu machine using 2.2 GHz Intel Xeon Silver 4210 CPU. The results have shown a maximum speedup of 34.6× in run time and 16.7× more memory efficient using this method.

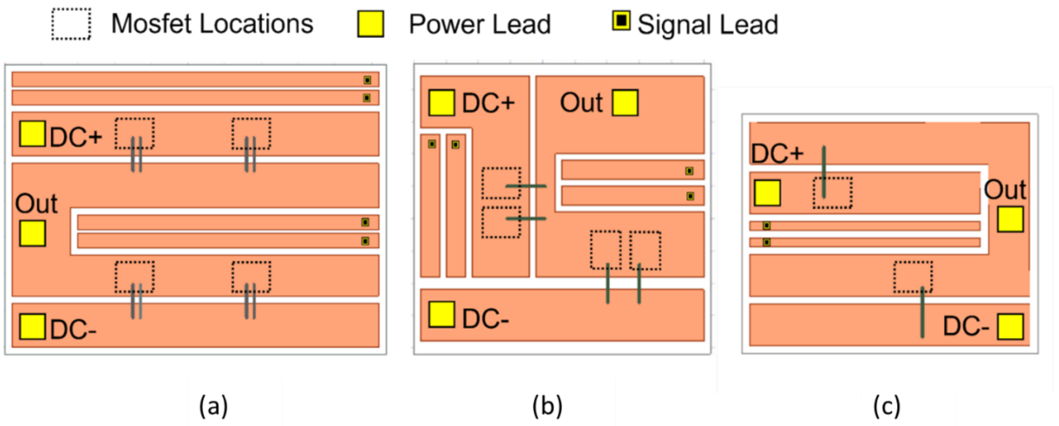


Fig. 5.16 Selected Layout for Comparison versus FEA

Table 5.2 Extraction Time and Memory Comparison for Different Layout Cases

	This model			**Ansys Q3D**			**Comparison**		
ID	**Ls (nH)**	**Runtime (s)**	**Memory (MB)**	**Ls (nH)**	**Runtime (s)**	**Memory (MB)**	**Error**	**Speedup**	**Memory**
1	15.6	1.85	9.4	14.8	64	157	5.5%	× 34.6	1:16.7
2	18.4	1.5	4.7	17.4	40	74.5	5.7%	× 26.7	1:15.8
3	21.4	0.6	6.2	20.6	15	73.7	3.9%	× 25	1:11.8

5.4 Loop Based Model Efficiency for Optimization and Circuit Simulation

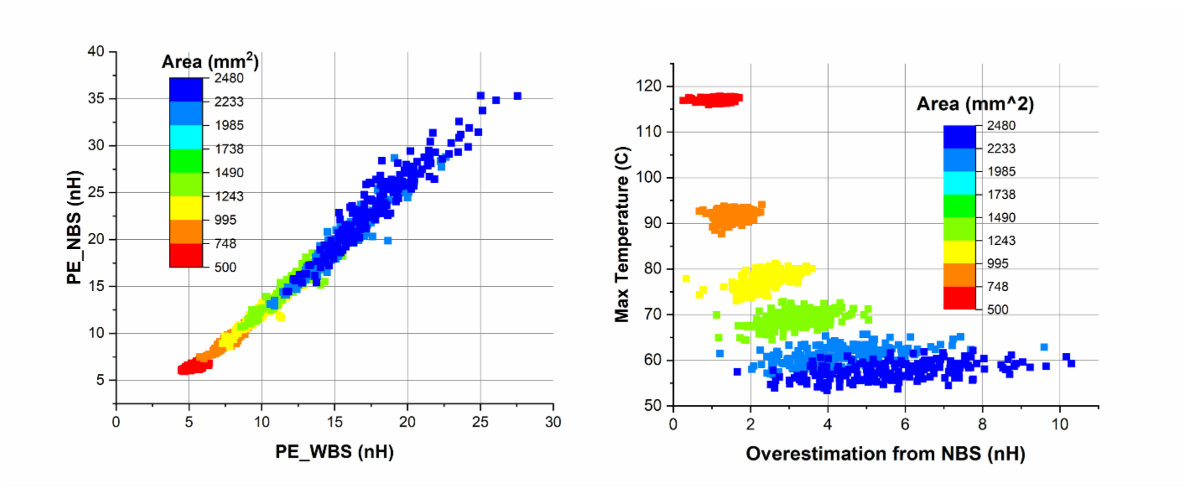
A. The Impact of Eddy Current on Layout Optimization

For the study in [50], an optimization study is performed using both the PE-NBS and PE-WBS approaches to show the impact of eddy current consideration in a layout optimization problem. The models are implemented in the latest PowerSynth tool where the layout generation algorithm in [18] allows a parametric study of multiple different footprint sizes, which has been used to generate the layout solution. The thermal model from [80] has been used through the application interface in [21] to demonstrate the impact of the eddy current on the optimized electrothermal solution space. This API allows a quick and accurate evaluation of the steady state temperature for each component in the MCPM layout geometry. In this study, each of the device heat dissipation is set to 10 W where the heat transfer coefficient is set to 150 W/m^2.K and the ambient temperature is set to 300K.

Using PowerSynth, about 1200 layouts are generated with six different floorplan sizes ranging from 500 mm^2 to 2475 mm^2 . In this experiment, for each floorplan size, the maximum available area for randomization is calculated based on the difference between the minimum floorplan size and the given floorplan size. Then, the room is distributed following a weighted distribution to vary the trace dimensions and other components (i.e., devices, leads) locations. The time required for the 1200 layouts inductance evaluation is about 2200s using the PE-WBS or PE_NBS models.

The regression model for the backside consideration and the analytical equations without backside consideration have almost same computation efficiency. Conversely, if Ansys Q3D has been used for the optimization, it would take about 20 hours. In average, this is a 35× speedups for each layout extraction. A comparison between the PE_NBS and PE_WBS models are shown in Fig. 5.17 (a). Fig. 5.17 (b) illustrates the absolute difference between the two models versus maximum device temperature. From these results, it can be concluded that, the maximum absolute difference between the two models increases with a larger floorplan, indicating that the PE_NBS model is more inaccurate with a larger floorplan size. This is mostly from the higher impact of the eddy current with a larger loop.

It is worth mentioning that a larger floorplan size is usually required for a more complicated MCPM layout (e.g., more devices per switching position). Therefore, the PE_NBS model is not recommended for circuit simulation of more complicated MCPM layout. Additionally, due to this inaccuracy, there is a shift in the optimized pareto-front using PE_NBS and PE_WBS models as shown in Fig. 5.18. From this pareto frontier results, it can be concluded that the incorporation of the backside effect would increase the accuracy of the extraction, however it does not affect the trend of the optimization results.



a)

b)

Fig. 5.17 (a) Direct comparison between 2 approaches (b) Absolute difference between 2 approaches versus maximum temperature.

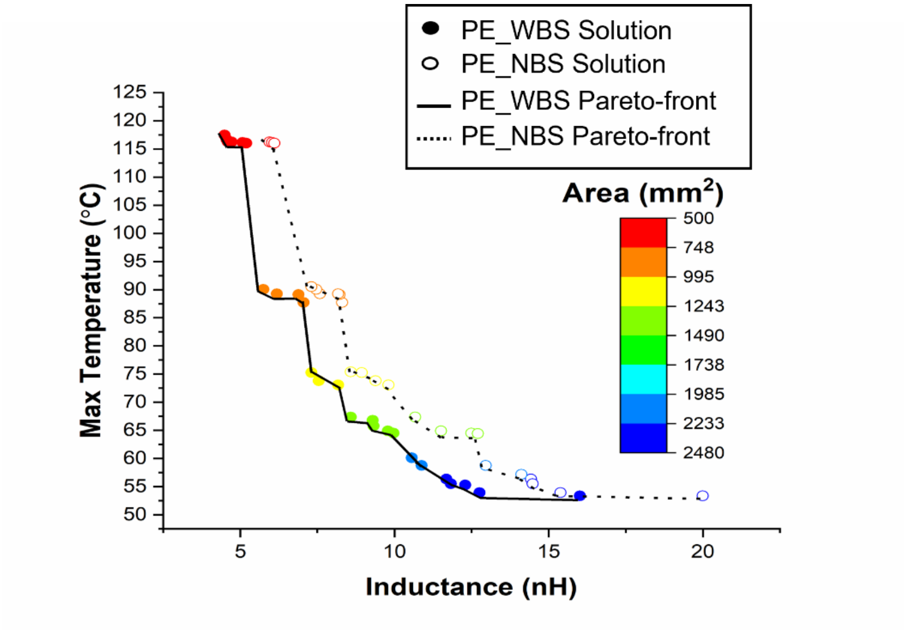


Fig. 5.18 Pareto frontier comparison for with and without backside consideration

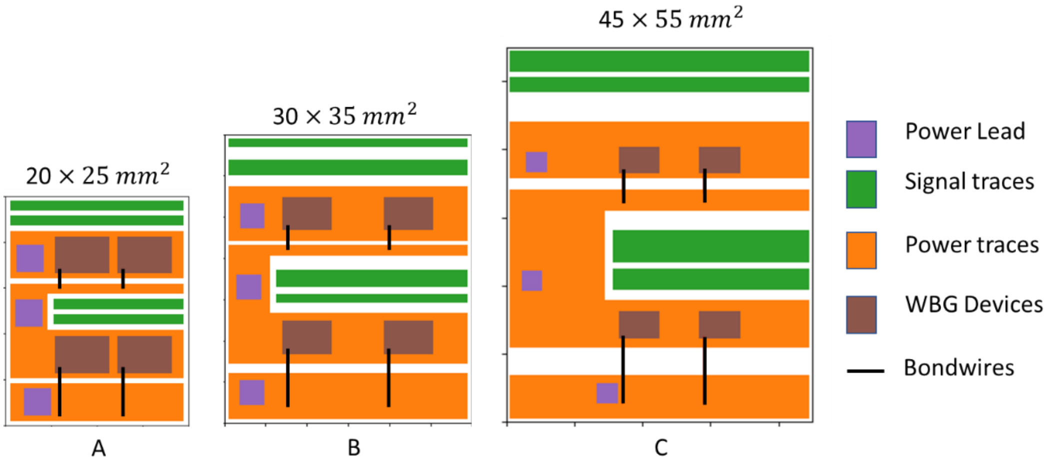


Fig. 5.19 Selected optimized solutions from the Pareto Frontier

Table 5.3 shows the error comparison of some optimized layout solutions from the pareto frontier in Fig. 5.18. The error margin expands with increased floorplan sizes. Despite the large error due to ignorance of eddy current, the PE_NBS model still shows the same trend within the same floorplan size Fig. 5.17 (a). Therefore, one can still apply the PE_NBS to minimize the loop inductance for a fixed floorplan size problem. However, the extracted parasitic result is inaccurate for the further circuit simulation study.

In this case, the characterization steps described in Chapter 2 must be applied to improve the accuracy of the extracted parasitic parameters. In addition, as shown in Chapter 2, this process only takes about 15 minutes to complete for a selected DBC substrate. Therefore, this characterization process can be applied to any partial element models (e.g., [35]–[37]) to improve the extraction accuracy.

Table 5.3 Selected Optimized Layouts from PE_WBS Pareto-front

Layout ID	PE_WBS (nH)	PE_NBS (nH)	ΔL (nH)	Max Temp ($°C$)	Size (mm^2)
A	5.19	6.11	0.92	116	500
B	8.18	9.8	1.62	73.1	1050
C	11.83	14.87	3.04	55.3	2475

B. Circuit Simulation Study Using the Loop-based Model and the Comparison between Distributed vs Non-distributed Netlist

The advantage of the loop-based approach in this work is the extraction of the distributed netlist, enabling a post-layout circuit simulation study and verification of a selected solution. While this process can be done using PEEC or FEA methods, it requires more efforts to set up and run simulations, as discussed in [35].

In this circuit simulation study, a double pulse test (DPT) circuit simulation is performed in LtSpice to demonstrate the importance of the distributed netlist. Here, for the extracted netlist from the half-bridge layout on Fig. 5.19 (a), four CPM2-1200-0040B from Wolfspeed are switched with DC-DC voltage of 400V, the two pulses are shown in Fig. 5.20 (a) where the rise and false time of the gate-signal is 50ns. The total load current is 40A through the load inductor of 50 uH. The LtSpice device models [76] have been used to ensure most accurate simulation results. Two simulations are run to compare the distributed netlist extracted from the model using the process in Chapter 4 and a single lumped element for the loop inductance of 15.5 nH at 10 MHz. As shown in Fig. 5.20 (b), with the same loop-inductance, the ringing oscillation in both cases show the same resonance frequency of about 70 MHz. There is a 31 V difference in the peak overshoot voltage using the lumped versus the distributed model, which is about 75% difference if a single lumped element is used. This is because in the single lumped element simulation does not capture the switching behavior of each device correctly. On the other hand, the distributed netlist allows a more accurate analysis on the interaction between the layout parasitic parameters and device parameters (e.g., Ciss, Coss, Crss …). More importantly, a more detailed analysis of the current sharing among devices can be done using the distributed netlist as seen in Fig. 5.20 (c). In the future, this would help the optimization tool decide on a more optimized layout in terms of current balancing.

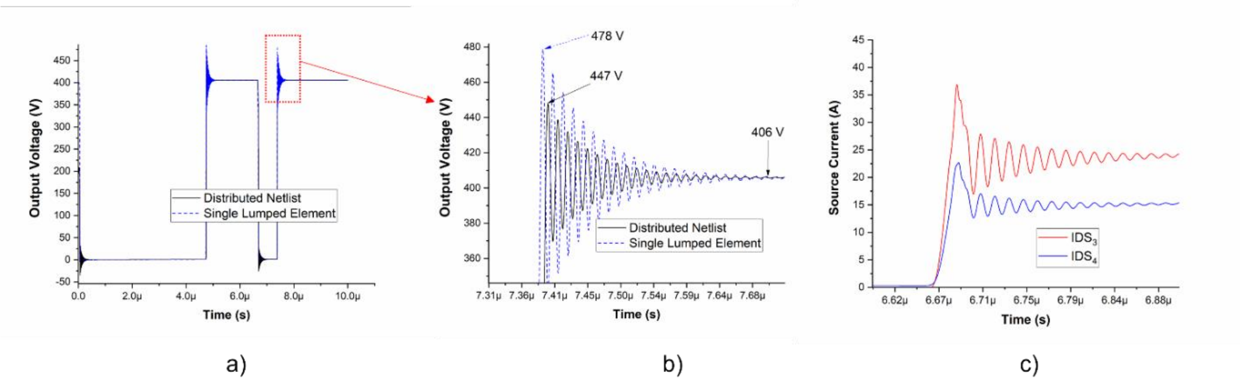


Fig. 5.20 a) DPT VDS waveform b) Zoomed in of VDS c) Source current comparison on lower side devices.

5.5 3D Extraction Results Using PEEC and Loop-based Models

As the layout engine is now able to support different 3D MCPM geometries, the electrical model also needs to be improved. Both the PEEC-based and Loop-based models can be applied for 3D MCPM layout structures. However, due to the limitation of the layout engine to electrical API, a FastHenry API has been used in [81] to obtain the 3D layout optimization solutions. With some current updates in the meshing algorithms and the layout engine to electrical API, the 3D model can be validated against the lasted validation effort of a fabricated 3D module in Fig. 5.22 using the same impedance analyzer E4990A in Fig. 5.14. For this validation, a 3D DUT from PowerSynth has been designed on a gold-plated DBC substrate (0.1 mm Cu – 0.5 mm AlN – 0.1 mm Cu). This 3D design is chemically etched on both sides where the bondwires are used to connect the drain-source, gate signal, and kelvin signal. The layer-layer the cross-section of this design is shown in Fig. 5.21 below.

Table 5.4 shows the extracted results of the PEEC model, Loop-based model, FastHenry, and measurement results at 10MHz. It is worth noting that, in this experiment, the accuracy of the impedance analyzer is not guaranteed as the inductance of the structure is way below 10nH leading to potential measurement error greater than 10% according to the datasheet (Fig. 5.13). Hence, a different measurement method needs to be used in the future to revalidate the accuracy of these models.

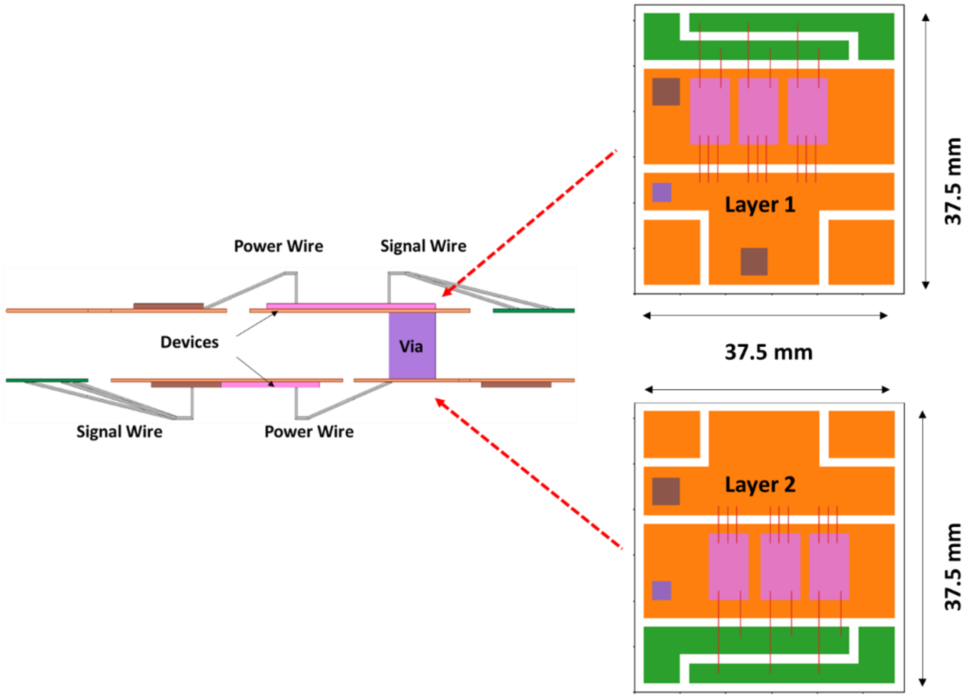


Fig. 5.21 Side and layer-layer views of the fabricated 3D layout.

PEEC and Loop-based results are quite close to the FastHenry results and within 20% of versus the measurement results. The main source of inaccuracy is coming from the different modeling approaches for the bondwires in PEEC, Loop-based, and FastHenry. For instance, the bondwires in Loop-based and FastHenry models are lumped into a single ribbon wire during the loop-evaluation process. This ribbon-wire structure reduces the mutual inductance impact among the wires, leading to smaller overall inductance of the wire-group. Conversely, both the self-inductance and mutual inductance of the wires are considered in PEEC model. These bondwires models need to be improved in the future if they are used in a 3D structure. This is because for the wire-bonded 3D module, the parasitic inductance of the bondwires have the highest contribution to the overall loop value.

For this 3D structure, the main advantage of the loop-based method versus PEEC and FastHenry is the computational efficiency. This makes the loop-based model most suitable for 3D

MCPM layout optimization. Table 5.5 shows the extraction time among these 3 methods. The loop-based model has shown the most speed-up in comparison to FastHenry. There are two main reasons for this speedup. First of all, the uniform meshes of the loop objects allow more efficient self and mutual inductance evaluation which reduces the total formulation time. Secondly, the loop-based model breaks the layout into multiple bundles and solve them separately. This bundle creation step reduces the matrix size to just 9×9 in comparison to the 1574×1574 of the PEEC model, leading to a more efficient matrix evaluation.

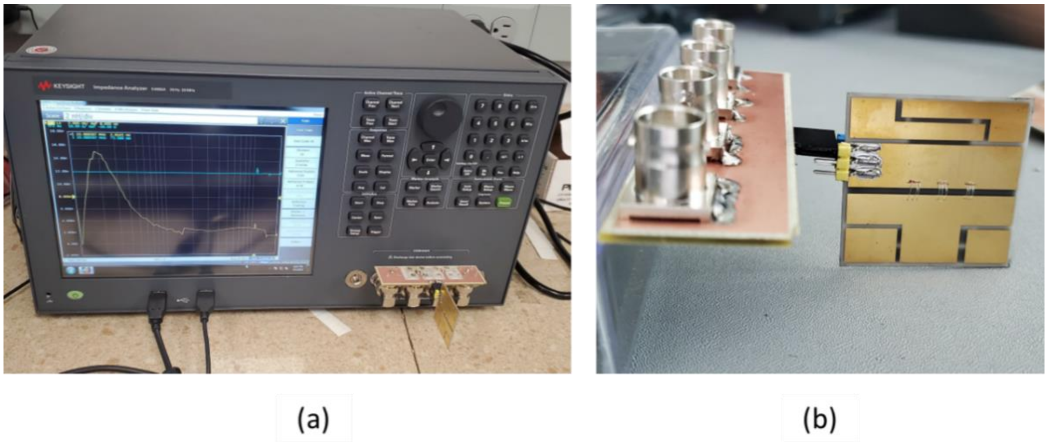


Fig. 5.22 (a) E4990A impedance analyzer (b) Fabricated 3D DUT.

Table 5.4 Comparison versus a 3D DUT case @ 10MHz

Methods	**PEEC**	**FastHenry**	**Loop-based**	**Measurement**
Inductance (nH)	3.54	2.98	3.93	3.43
Error vs Measurement	3.2%	13%	14.5 %	N/A

Table 5.5 Efficiency Comparison among Extraction Methods

Methods	**PEEC**	**Loop-Based**	**FastHenry**
Formulation Time	2.3 s	1.3 s	N/A
Matrix size	1574 × 1574	9 × 9	N/A
Evaluation time	1.9 s	2.3 ms	63.3 s
Speed up	×15	×43	× 1

Chapter 6. Dynamic Performance Optimization for Co-Electrical-Thermal Management Design

PowerSynth has shown the complete design flow for MCPMs, which offers a multi-objective layout optimization algorithm and reduced-order models for electrical parasitic extraction and thermal evaluation. While these models are accurate, there is no connection between the electrical parasitic and device temperature during the layout optimization process. Hence, the multi-objective optimization algorithm optimizes these objectives separately without insights into their impacts on the reliability and performance of the WBG device. In the worst scenario, the device will eventually run into thermal run-away issues due to the exponential increase of WBG device power loss versus temperature. However, in reality, this scenario rarely occurs. Before this event, the solder-attach material and the aluminum metallization melted, leading to an unfunctional circuit. Therefore, this chapter incorporates the physics-based device model into the power loss calculation for an accurate temperature-dependent prediction. To increase the evaluation speed and the robustness of the optimization routine, a Feed Forward Neural Network (FFNN) has been implemented. This method has shown up to 300000 speedups while maintaining under 5% error.

Section 6.1 details the implementation of the MSCAD SiC physics-based model and the device optimization tool. This tool helps the modeling engineer to automate the tedious parameter extraction process.

Section 6.2 details the methodology of the electrothermal co-simulation. Here, the successive approximation method is introduced. This method allows the user to predict the final steady-state value of the MCPM layout through multiple iterative steady-state evaluations.

Section 6.3 describes the formulation of the FFNN model. This method is fast, accurate, and it allows parametric study during the layout optimization process. While circuit simulation is available, multiple circuit simulations within an optimization loop is not preferred because of the amount of memory required for both tasks. Moreover, running parameterized circuit simulation can lead to non-convergence issues, decreasing the robustness of the optimization process.

Section 6.4 shows some of the layout optimization results using the method. The Pareto-frontier is also compared against the static-optimization results. This new layout optimization approaches allow the user to find the operational limit of the MCPM layout with a given circuit condition or thermal management strategy.

6.1 The UA MSCAD Physics-based Model and Device Optimization Tool

A. Implementation of the UA MSCAD SiC model in Python

The physics-based SiC device from the UA MSCAD lab is one of the first datasheet-driven compact models available in the power electronics society [76], [82], [83]. Throughout this continuous research and development, the model has achieved great accuracy for static and transient analyses. The work in [83] has demonstrated this achievement by reaching within 6%-16% error for the device power loss prediction between the circuit simulation and measurement. More importantly, the thermal dependent feature of these models [76], [82], [83] allows a more accurate circuit simulation leading to a more accurate dynamic switching losses evaluation.

To evaluate a more accurate thermal dependence of the switching and conduction losses, the model in [76] has been implemented in Python. The main equations of these models are briefly explained in this work. Fig. 6.1 illustrates the device structure and equivalent circuit of a SiC power MOSFET. The drain to source voltage from this equivalent circuit can be divided into three parts:

- The voltage drop across the source/substrate resistance R_s
- The voltage resulting from the Ohmic drift resistance in the n- epitaxial layer
- The voltage drop across the inversion channel (V_{dnrsnr}) which is a function of gate-source voltage and the core of device operation.

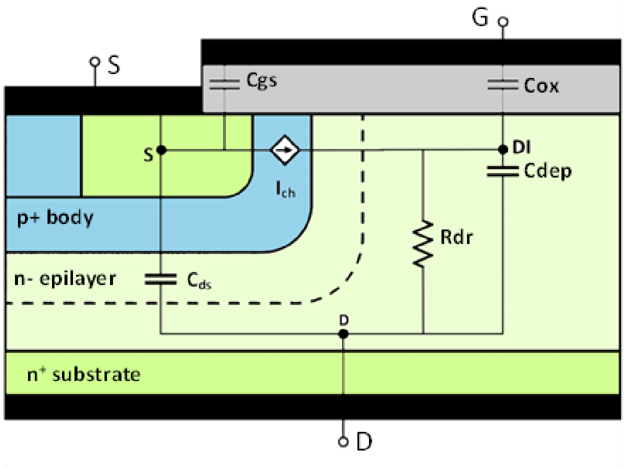


Fig. 6.1 SiC Power MOSFET device structure with corresponding parasitic elements

The channel resistance (drift resistance) is non-linear which is a function of both drain-source, V$_{dnrsnr}$ and gate-source voltage, V$_{gsnr}$. Here, the drain-source current can be divided into 2 regions namely linear and saturation region depending on the drain and gate bias. The linear region (V$_{gs}$ >V$_{th}$ and 0< V$_{ds}$ < V$_{dssat}$) current is expressed by equation (6.1) below:

$$I_{mos_x} = \boldsymbol{kf}_x \cdot \boldsymbol{kp}_x \cdot \left(v_{gsi} - \boldsymbol{vt}_x\right) v_{disi} - \boldsymbol{pvf}^{yx-1} \cdot \frac{v_{disi}^{yx} v_{gsi}^{\frac{2-yx}{yx}}}{1 + \boldsymbol{theta}_x (v_{gsi} - \boldsymbol{vt}_x)} \tag{6.1}$$

here:

$$yx = \frac{kf_x}{kf_x - \frac{pvf}{2}} \tag{6.2}$$

In equation (6.1), x denotes the low and high region (x = low, high) which arises from the gradual de-trapping of the trap states near the conduction band. ***theta*** accounts for the vertical filed dependent mobility reduction while the pinch-off parameter ***pvf*** represents the gradual transition between linear and saturation region, a characteristic of SiC power MSOFETs. ***kf*** and ***kp*** control the transconductance of the saturation and linear region respectively. It should be noted that the identical equation is used for low and high current components in order for decoupling the parameter influence on each other. The distinguished threshold parameters, ***vt*** are to represent the different effective MOS capacitance.

The saturation region current ($V_{ds} > V_{dssat}$) is expressed by equation (6.3):

$$I_{mossat_x} = \frac{\frac{1}{2} kp_x \left(v_{gsi} - \boldsymbol{vt_x}\right)^2}{1 + \boldsymbol{theta_x}(v_{gsi} - \boldsymbol{vt_x})} \tag{6.3}$$

The model includes temperature scalable equations for the temperature dependent parameters such as transconductance, threshold voltage, pinch-off voltage parameter, vertical field mobility reduction parameter etc. based on [76].

The dynamic behavior has been captured with the intrinsic inter-electrode capacitance formulations. Three major dynamic components of the power MOSFETs are gate-drain, gate-source, and drain-source capacitances. They are related to the conventional datasheet provided characteristics in the following manner:

$$C_{iss} = C_{gs} + C_{gd} \tag{6.4}$$

$$C_{oss} = C_{gd} + C_{ds} \tag{6.5}$$

$$C_{rss} = C_{gd} \tag{6.6}$$

The gate-source capacitance is mostly independent of the drain-source voltage since the source side charge is unaffected by the applied drain biases. Gate-drain capacitance, commonly known as Miller-capacitance consists of the oxide capacitance, C_{ox} and the non-linear depletion capacitance, C_{dep}. When the applied bias is below a pinch-off voltage, V_{td}, the capacitance equates to C_{ox} while C_{dep} is given by equation (6.7):

$$C_{dep} = \frac{A\epsilon}{0.5\sqrt{\frac{2\epsilon\left|V_{td} + V_{dg}\right|}{qN_D}}}: \tag{6.7}$$

here, ϵ represents the SiC permittivity, V_{td} denotes the pinch-off voltage and N_D is the drift region doping. The final component, C_{ds} is expressed with the conventional junction capacitance formula as given by:

$$C_{ds} = \boldsymbol{CDS}\left(\frac{\boldsymbol{V_{bi}}}{\boldsymbol{V_{bi}} + V_{ds}}\right)^{M} \tag{6.8}$$

here, M is the junction capacitance grading exponent, $\mathbf{V_{bi}}$ is the contact potential and **CDS** is the zero-bias junction capacitance.

The C-V equations are pretty simple to implement in Python since they do not have dependent variables among different terms. Conversely, the three different voltage drops from the drain to the source pins of the device require an iterative solver. This can be done by setting up a Kirchhoff Voltage Law (KVL) problem in equation (6.9), where the Python Scipy package has been used to iteratively solve the internal V_{dnrsnr} value for every different gate to source bias. Here, the drain to source voltage is usually fixed to 20 V, which is the standard for most device datasheets.

Solve (v_{dnrsnr}) where: (6.9)

$$\left(r_{drift}\left(v_{dnrsnr}, v_{gs}\right) + r_s\right) \times I_{mos}\left(v_{gs}\right) + v_{dnrsnr} = V_{ds}$$

To verify the functionality and accuracy of this evaluation setup, a set of parameters are first fitted to the measurement for a C2M0025120D MOSFET from CREE. This set of parameters are used in the Python implementation where the fitting results are shown in Fig. 6.2 below.

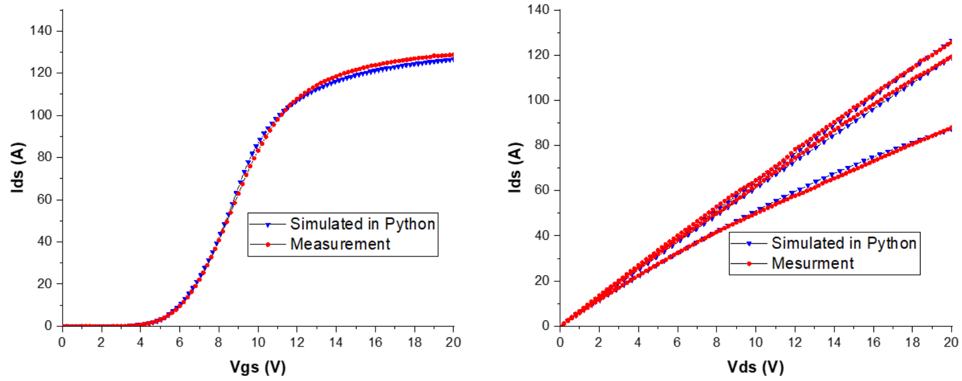


Fig. 6.2 Measurement vs fitting results of SiC MOSFET at room temperature

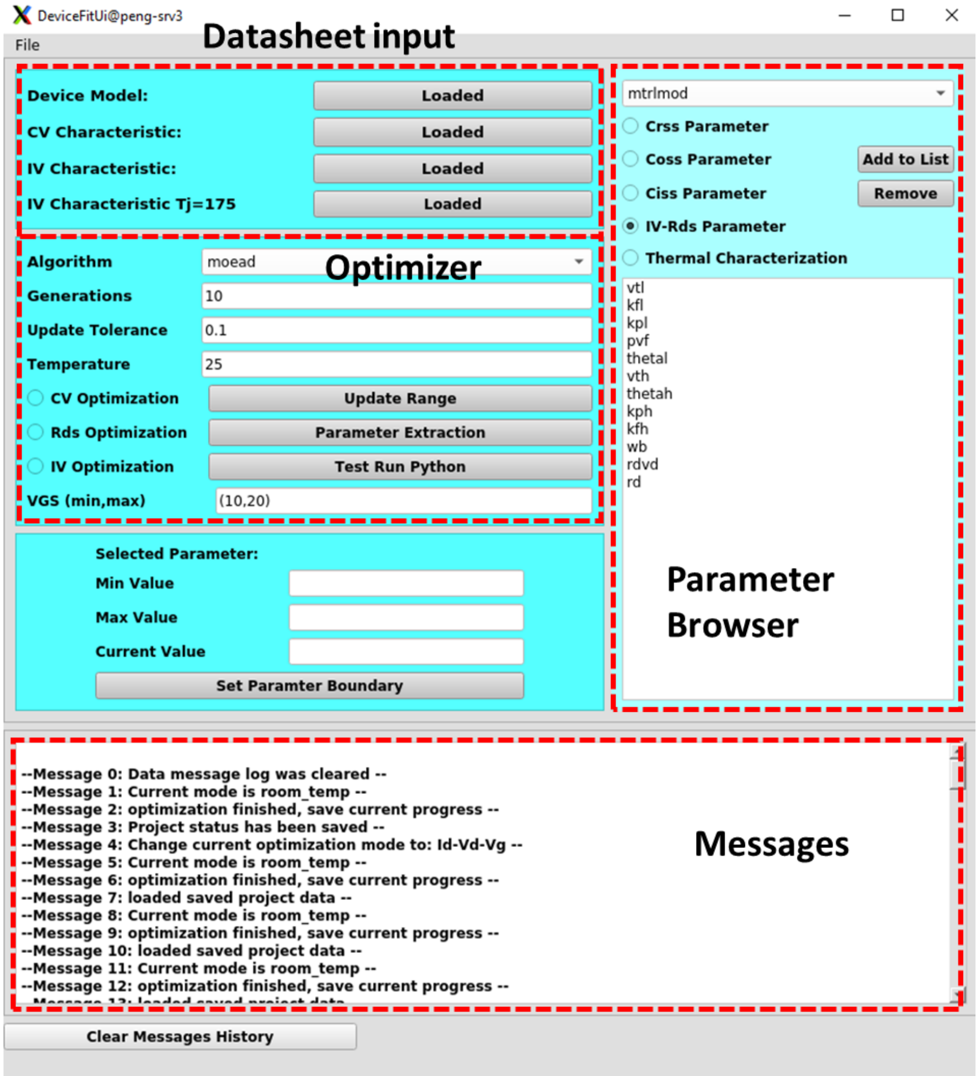
It is worth noting that, this implementation also works for Si, GaN or other MOSFETs. This is because although these devices have different physical properties, they have exactly same internal circuit. Furthermore, this implementation removes the need for an API to the circuit simulator which can have some benefits for software integration and optimization.

B. The Device Parameters Optimization Tool

Along with implementing the SiC MOSFET model in Python, it is possible to implement an automated parameters fitting tool. This has been made possible thanks to the collaboration between the modeling and the PowerSynth group. Device parameter fitting is a very tedious task where the modeling engineers need to manually change each parameter and rerun the circuit simulation to compare the results. Device fitting tool such as Keysight IC-CAP is available and could help to automate this task. However, this tool is not available cross-platforms, while most of the Power Electronics engineers these days prefer to choose an open-source simulator such as LtSpice as the core simulator. The latest LtSpice MOSFET model has made it possible for these demands [76].

For this purpose, a Python automate device parameters optimization tool (Fig. 6.3) has been developed. This tool is cross-platform for Linux and Windows operating systems (OS). It is worth mentioning that this tool also allows the muti-processing multi-objective optimization, which significantly speeds up the parameter extraction process. The tool is still under development and testing state. However, it already shows promising results. As for the C-V curve fitting of the SiC MOSFET model, it would take less than 1 minute to achieve a perfect fit between the model and the datasheet/measurement data. The I-V curve fitting functionality is still under investigation, and the algorithm has not yet been able to achieve a perfect fit versus the datasheet. However, it still gives excellent initial parameters results that the modeling engineer can further optimize manually.

The tool currently uses the Python multi-objective optimization package (pymoo). Here, at least ten optimization algorithms have been tested, including heuristic and gradient based. It has been found that the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [84] is best for the parameter extraction task. This algorithm uses the problem decomposition strategy to combine good solutions from neighboring problems. Because of this, it has a nice convergence property. This algorithm also allows multi-processing features, which further saves the parameter extraction time. Besides the automation features, the tool allows the user to change the parameters and check for the fitted curves manually. Thanks to the cross-platform compatibility of both the device models and the device optimization tool, the tool can be used by any power electronics engineer in its mature state. This would further improve the popularity of the models to the power electronic society. The future plan will also incorporate the automated parameter extraction for the MSCAD GaN MOSFET model [85].



Device Model: Loaded
CV Characteristic: Loaded
IV Characteristic: Loaded
IV Characteristic Tj=175 Loaded

Algorithm: moead
Generations: 10
Update Tolerance: 0.1
Temperature: 25
○ CV Optimization
○ Rds Optimization
○ IV Optimization
VGS (min,max): (10,20)

Selected Parameter:
Min Value
Max Value
Current Value

mtrlmod
○ Crss Parameter
○ Coss Parameter
○ Ciss Parameter
● IV-Rds Parameter
○ Thermal Characterization

vtl
kfl
kpl
pvf
thetal
vth
thetah
kph
kfh
wb
rdvd
rd

```
--Message 0: Data message log was cleared --
--Message 1: Current mode is room_temp --
--Message 2: optimization finished, save current progress --
--Message 3: Project status has been saved --
--Message 4: Change current optimization mode to: Id-Vd-Vg --
--Message 5: Current mode is room_temp --
--Message 6: optimization finished, save current progress --
--Message 7: loaded saved project data --
--Message 8: Current mode is room_temp --
--Message 9: optimization finished, save current progress --
--Message 10: loaded saved project data --
--Message 11: Current mode is room_temp --
--Message 12: optimization finished, save current progress --
--Message 13: loaded saved project data
```

Fig. 6.3 GUI of the automate device parameters extraction tool

6.2 Electrothermal Co-Simulation Idea for the True Trade-offs between Electrical and Thermal Domains

A. PowerSynth Current Limitation and the Needs for Electrothermal Co-Simulation

The reduced modeling efforts in PowerSynth has allowed fast and accurate evaluations for both electrical parasitic parameters and maximum device temperature. The main limitation of the tool is its lack of consideration for the dynamic operating parameters of the circuit such as switching frequency, voltage and current etc. These parameters couple with the layout parameters such as parasitic inductance and capacitance, leading to different switching losses of the device. The variation of the switching losses eventually changes the maximum device temperature, leading to unreliable and unexpected thermal performance. Furthermore, both switching and conduction losses of a MOSFET are temperature dependent. Thus, it is impossible to predict or evaluate these losses without a good understanding of the device physics. Therefore, some model-based engineering [86] (MBE) needs to be incorporated in the tool to consider the dependency among the thermal, electrical and device modeling. Thanks to the implementation of the physics-based model in Python some of these dependencies can be addressed.

In several studies [87][88], although WBG devices such as SiC can handle much higher junction temperatures than their Si counterpart, the thermal run-away problem is still an issue if the thermal dissipation system is designed poorly. As shown in Fig. 6.4 , the power loss (dashed line) of a SiC device is temperature dependent. A steady state can be reached with a good cooling system when the device's power loss equals the cooling capability. Normally, this steady state is defined by the user by fixing the upper limit for device junction temperature from which a good thermal management system is designed to meet the heat dissipation requirements. In the worst scenario, the device will eventually run into thermal run-away issues due to the exponential increase of WBG device power loss versus temperature. Before this event, the solder attach material and the aluminum metallization melts and the circuit fails. In either case, improper cooling system quickly leads to the failure of the whole system. Furthermore, according to [89], this power loss value also depends on the electrical parameters such as parasitic inductance, device's internal parasitic, gate resistance, etc. Therefore, an accurate estimation on device power loss and its correlation with electrical-thermal parameters during optimization is crucial for the performance and reliability of the MCPM design.

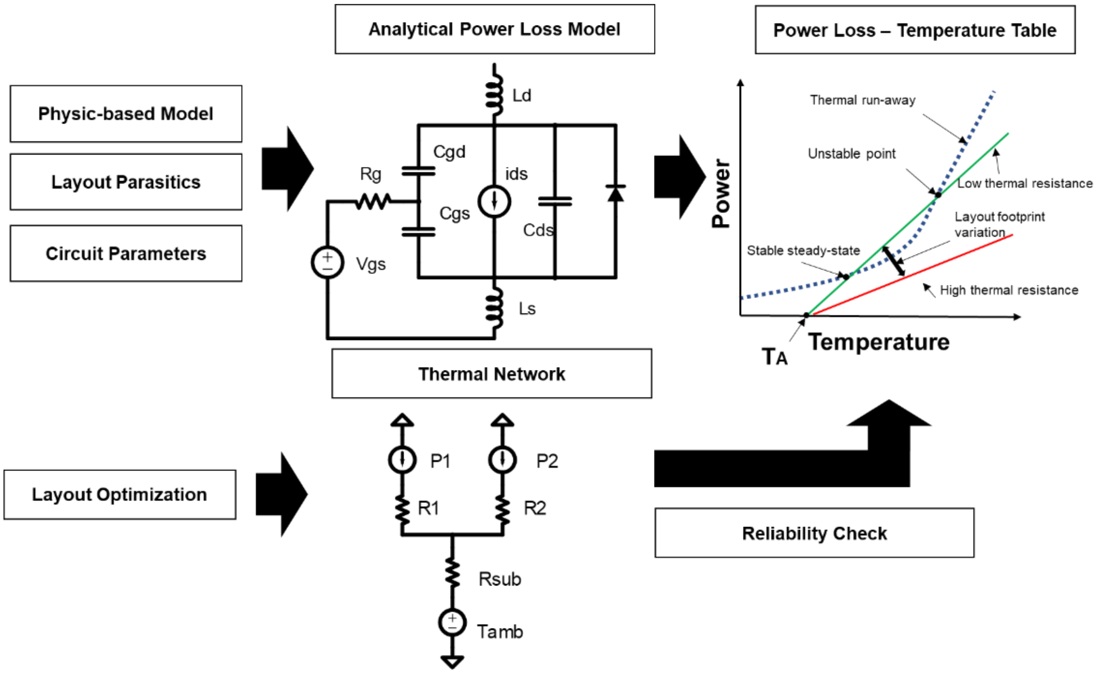


Fig. 6.4 Electrothermal simulation for dynamic performance

B. The Successive Approximation Method

Fig. 6.4 illustrates a very simple relationship between the thermal resistance (as a slope) and the power loss – temperature curve. This representation is only correct for very simple systems that can be represented by a single thermal resistance value. However, this is not the case for MCPM. Due to the interaction among different devices, the thermal coupling resistance among the devices need to be considered. To extract the thermal resistance network, the PowerSynth-ParaPower API [21] has been used. This API allows PowerSynth to compute the steady state temperatures of different MCPM layout elements such as device, trace, substate material, and so on. To extract the thermal network, a 1W heat loss is applied sequentially to top surface of the devices in the setup to find the thermal resistance of the devices in the layout. A heat convection coefficient is also applied to the backside of the baseplate in the ParaPower simulation. The thermal resistance of each device considering the coupling resistance from the device on the opposite switching position can be extracted by calculating the temperature difference between the device top surface and the ambient temperature (T_{amb}) using the equation below.

$$R_{TH} = \frac{T_{device} - T_{amb}}{P_{loss} = 1W} \tag{6.10}$$

Using equation (8), an N×N thermal resistance matrix can be extracted from any layout. To demonstrate this thermal resistance network, an MCPM structure with 2 devices is shown in Fig. 6.5. In this thermal network, R_{11} and R_{22} are the self-thermal resistance of each device. R_{12} is the coupling resistance calculated using the equation (6.10) while the power is applied to one device and measured at the other.

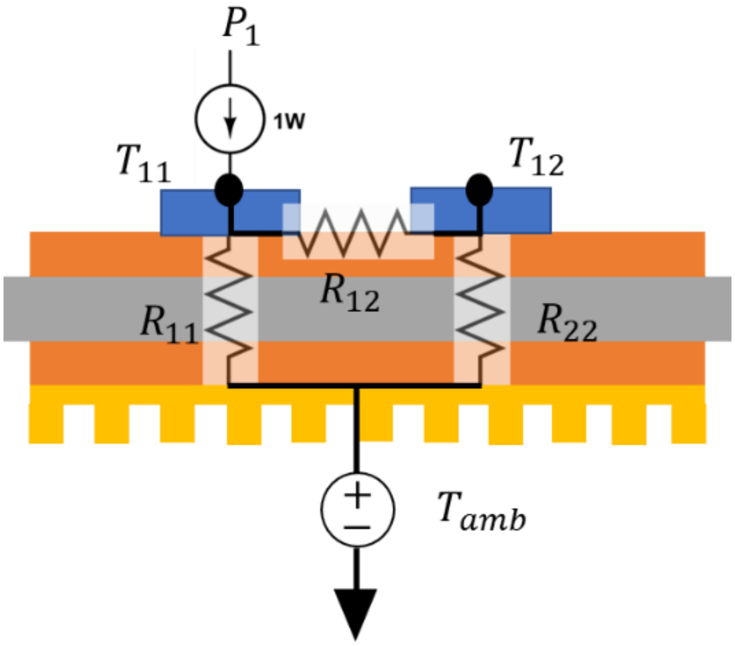


Fig. 6.5 Example thermal network extracted from an MCPM structure with 2 devices

Once the thermal resistance network has been extracted, a successive approximation technique can be applied to find the steady state temperature of the MCPM layout. Since the power loss of a MOSFET is a temperature-dependent value. It is very challenging to accurately evaluate the steady state value of the circuit without performing circuit simulations. The successive approximation method Fig. 6.6 iteratively updates the temperature and power loss value where the device's temperature difference of the consecutive iterations is calculated. When the temperature difference is smaller than a tolerance value (e.g 0.5 °C) the iterative process is reached, and the final steady state temperature is reported. In this work, the steady state junction temperature is set below 220 C for each device. This is because at this temperature, even when the thermal runaway event does not occur, the solder attach of the device has been melted. This method has been used in [90], [91]. In [90], a very simple model for power loss evaluation has been used to optimize the runtime. However, this model does not take into account the parasitic parameters. Because of this, the energy loss in [90] is a fixed value for every parasitic and temperature combination. The work in

[91] first performed the Finite Element Analysis (FEA) to extract the thermal network. From here, iterative circuit simulations have been done to find the final steady state temperature of the circuit. The method has been experimentally validated in [91] by performing continuous switching experiment and temperature measurements of the DUT. The only drawback of this method are the time-consuming circuit simulations.

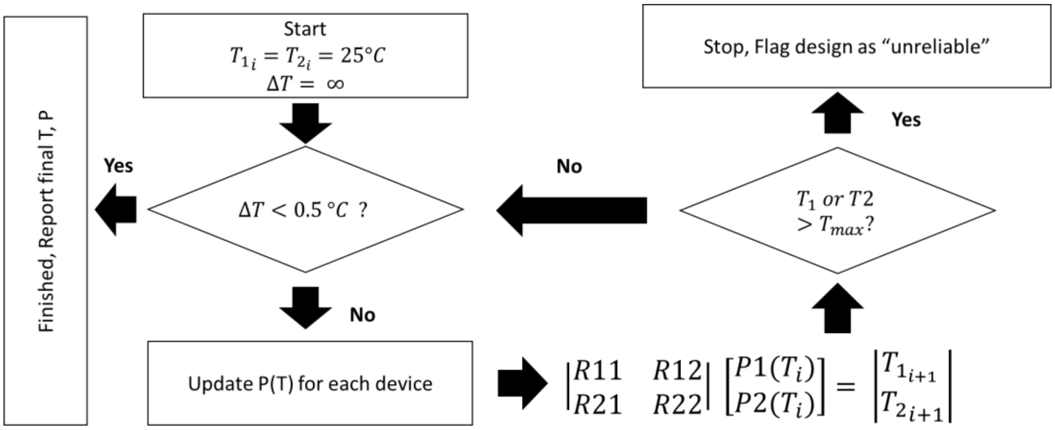

Start
$T_{1_i} = T_{2_i} = 25°C$
$\Delta T = \infty$
Stop, Flag design as "unreliable"
Yes
Finished, Report final T, P
Yes
$\Delta T < 0.5\,°C$?
No
T_1 or $T2$ $> T_{max}$?
No
Update P(T) for each device
$\begin{vmatrix} R11 & R12 \\ R21 & R22 \end{vmatrix} \begin{bmatrix} P1(T_i) \\ P2(T_i) \end{bmatrix} = \begin{vmatrix} T_{1_{i+1}} \\ T_{2_{i+1}} \end{vmatrix}$

Fig. 6.6 The successive approximation method

Using both ideas from [90], [91], a electrothermal co-simulation process in Fig. 6.6 can be performed during the layout optimization. This process takes input both thermal and electrical parasitic networks, which can be quickly extracted from the PowerSynth tool. However, the power loss computation considering the parasitic parameters is still quite computationally expensive since it requires a Differential Equation (DE) solver. This issue can be addressed using the Neural Network (NN) based method in the next section.

6.3 A Feed Forward Neural Network Regression Model for Layout Optimization

While the model in [89] is quite fast, performing a thermal sweep for each different layout parasitic configuration while calculating power loss is quite computationally expensive. Furthermore, since the implementation of the analytical power loss requires a differential equation solver to solve for turn-on and turn-off periods. Due to the implementation of the model in Python, which is an interpreted programming language. There is some delay in the interpretation step from the Python code to machine code. Thus, directly placing the analytical power loss calculation into the layout optimization loop is not preferred. Here, a feed-forward Artificial Neural Network (ANN) regression model using the Scikit-learn machine learning package has been used to remove this interpretation overheads.

The Python SALib library is first used to generate a set of 800 input parasitic parameter combinations for the power loss calculation. This set of parameters includes parasitic inductance values of the gate, drain, and source for each device (Lg, Ld, and Ls). The parameters are randomized in the range of 1-10nH, 1-20nH, and 1-20nH for Lg, Ld, and Ls, respectively. For each variation of Lg, Ld, and Ls a temperature sweep with 50 data points is performed between 25C and 200C to evaluate the switching losses versus temperature dependent.

In this calculation, the load current is set to 60 A and the DC-DC voltage is set to 600V. In the future, the current value can be considered as an input for the model. However, the relationship between current and switching loss is quite linear. Also, the circuit parameters are usually defined prior to the layout optimization. Hence, the circuit parameters such as voltage and current are set to be constants now. There are 4000 data points to train the neural network, and 400 data points are randomly taken to test the accuracy of the trained model. The results show less than 5% of error between the simulated and FFNN implementation. It is worth sharing that, during the training process of the model, 40 Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz are run parallelly on a Linux server. The number of epochs for the FFNN is set to 500 to train the most accurate FFNN model. The total time for data collection, model training, and model validation using the multiprocessing evaluation is 160 seconds. It would take 6400 seconds or about 1.8 hours on a single CPU computer. The total time for 4400 evaluations using the FFNN model is 17ms, with less than 5% error. The FFNN model shows about 376470× speedups in comparison to the analytical model. Hence this model is very suitable for the optimization process. As shown in Fig. 6.7, for some randomly selected parasitic parameters of Lg, Ld, and Ls, the FFNN model shows very good fit versus the analytical results. Here, the total switching loss is calculated for a switching frequency at 10kHz.

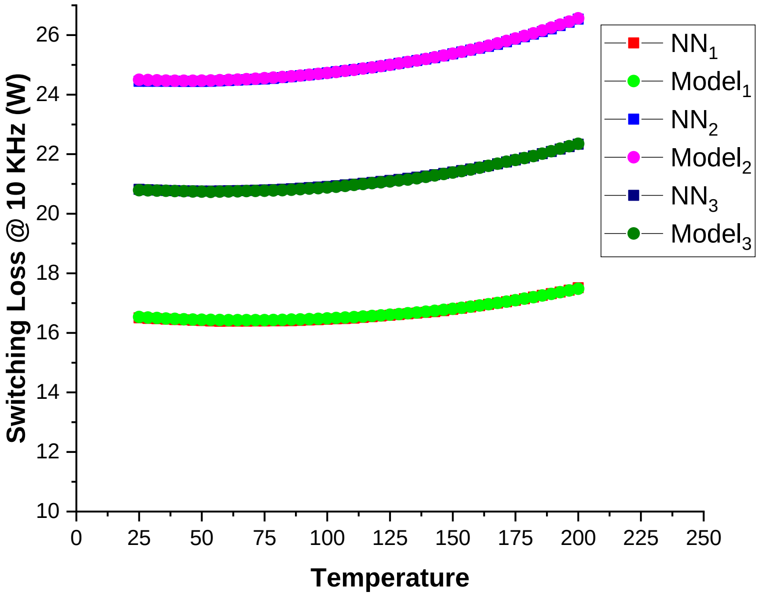


Fig. 6.7 Fitting accuracy between the FFNN and Analytical Model

6.4 Electrothermal Co-optimization Results

To find the parasitic impacts on an MCPM layout during the layout optimization, a layout optimization for a half-bridge layout with the footprint size of (40x50 mm^2) and one device per switching position as shown in Fig. 6.8 has been performed using PowerSynth. In the first optimization study, a power loss of 65W has been applied to each device where the convection coefficient value is set to be 500 (W/m^2 .K). This 65W value has been chosen since it is closed to the conduction loss value at 25°C of the MOSFET for the load current of 60A and DC-DC voltage of 600V. The maximum temperature and DC-DC loop inductance are used for the optimization target. Fig. 6.9 illustrates the solution space for 500 solutions, in this case, the loop inductance results are ranging from 10nH to 30 nH for this layout. The temperature results vary from 146 °C to 152 °C. Since the tool has no information about the power loss and temperature dependency, all of these layout solutions are marked as valid (green).

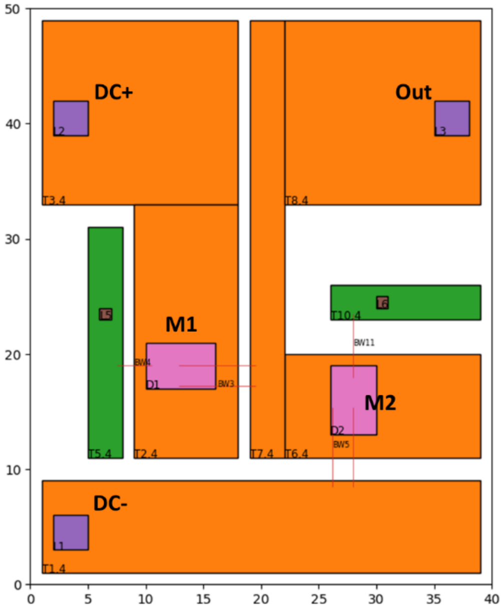


Fig. 6.8 Half-bridge MCPM layout for the study

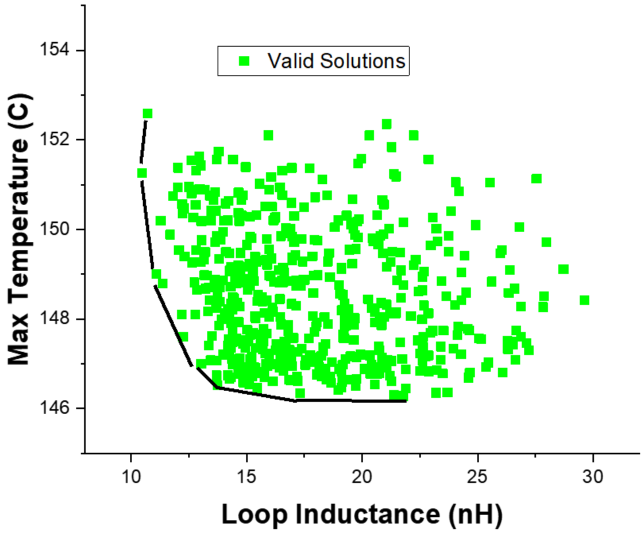


Fig. 6.9 A common solution space obtained from PowerSynth with all valid solutions

For each of the layout solution, both thermal and electrical netlists are extracted for each device switching position. These netlists serve as input for the FFNN power loss model to quickly find the power loss and temperature-dependent curve. The successive approximation method is applied to each layout solution where the new maximum temperature results are updated. If the process in Fig. 6.6 does not converge, or the temperature of a device is higher than the maximum temperature set at 220 °C, the process stops. The layout is then flagged as an invalid layout. Because the FFNN-based power loss model is extremely fast and accurate, the average time it takes to run the successive approximation for 500 layouts is about 5 ms on a single core of Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz. Therefore, performing a frequency sweep simulation and seeing its impacts on the layout solutions is possible. This frequency sweep can give the user a better layout selection with a given circuit operating condition

Fig. 6.10 illustrates the updated Pareto-frontier for various switching frequencies from 10kHz to 20kHz. There are two main points can be noted from these results:

- The maximum temperature range has been extended (80° C in Fig. 6.10) versus (6° C in Fig. 6.9).
- The maximum temperature increases with increasing loop-inductance due to the increased of switching losses.
- The solution space is shrinking for higher operating frequency value leaving only layout solutions with smaller loop-inductance valid.

It is worth noting that many solutions have been invalidated during the process described in Fig. 6.6. This results in the shrinking solution space as seen in Fig. 6.10. From these solutions, two layouts are selected with similar loop inductance values (Fig. 6.11). The first layout has the loop inductance of 10.5nH and the second one has the loop inductance of 10.8 nH. Fig. 6.12 illustrates the temperature increment of a layout solution with the loop inductance value of two chosen layouts at different switching frequencies. The results have shown that although layout 2 (10.8 nH) has a similar loop-inductance value, it has failed to maintain the maximum device temperature of less than 220 °C for the switching frequency greater than 18kHz. The reason for this is although layout 2 has similar loop inductance, the distribution of the inductance inside the loop is different than layout 1. Also, the devices are closer in layout 2, leading to higher thermal coupling among the devices.

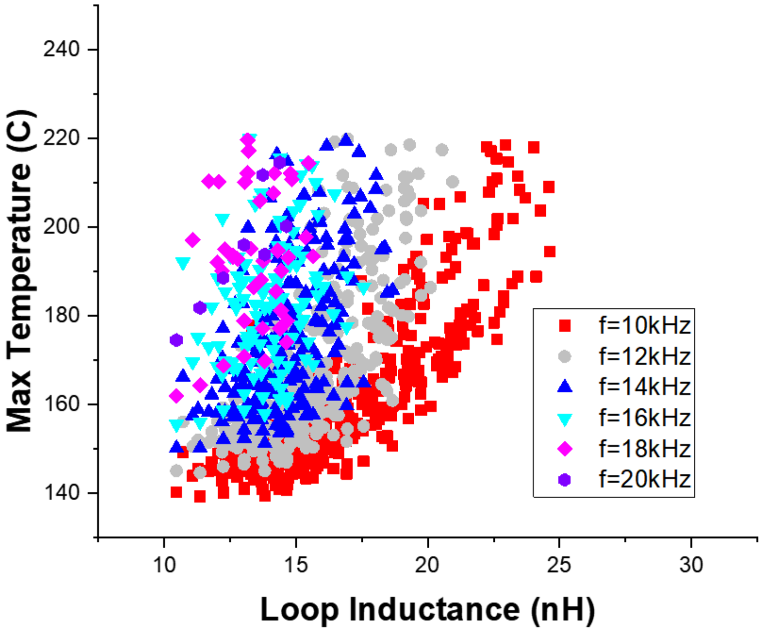


Fig. 6.10 Shrinking solution space for different operating frequencies

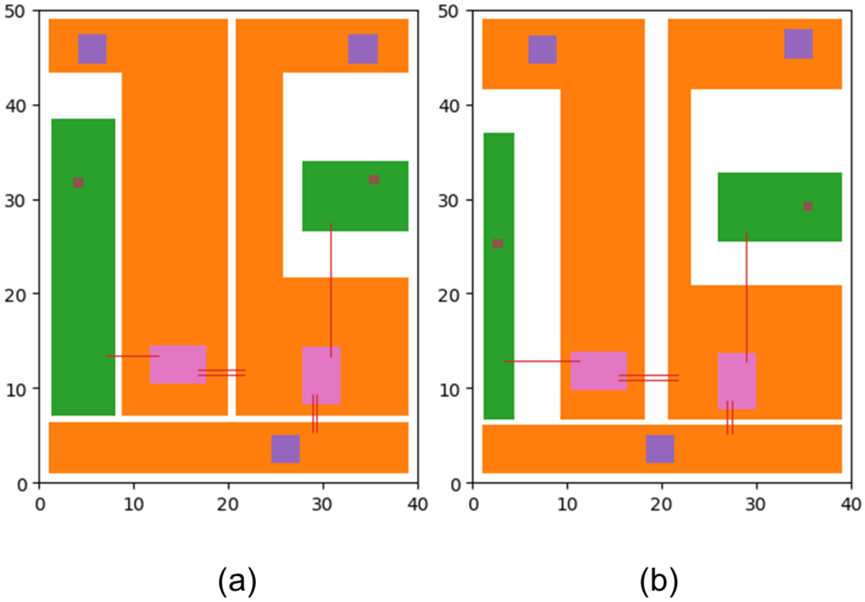


Fig. 6.11 Two selected layouts with similar loop inductance (a) Layout 1: 10.5 nH (b) Layout 2: 10.8 nH

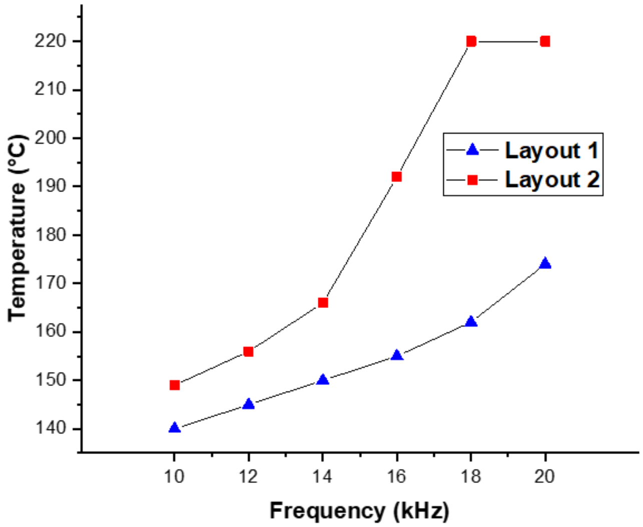


Fig. 6.12 Comparison between 2 layout solutions with similar loop inductance values

Table 6.1 Number of invalid solutions for different switching frequencies

Frequency (kHz)	# Invalid Solutions / 500
10	72
12	186
14	287
16	388
18	460
20	492

Table 6.1 shows the number of invalid solutions for each frequency value. By counting the number of iterations, it takes to convergence, the total number of iterations has been collected during the iterative evaluation of the successive approximation method. The total number of iterations for this experiment is 18489. Even though the analytical model is fast, this would take

up to 7.5 hours to complete. The total time taken for the FFNN model inside the successive approximation method is only 12.5s on the same machine. More importantly, the same process can be done for a totally different layout structure as long as they are sharing the same circuit topology. Therefore, this method is potentially beneficial for future layout synthesis algorithms where more layout structures need to be considered [58]. Moreover, the imbalanced switching losses due to different source-side parasitic parameters can be simulated using the circuit simulator and the physic-based model in [76]. Using the same strategy to build the FFNN power loss model, the same study can be done for MCPM layout with multiple devices in parallel on the same switching position.

Chapter 7. Conclusions and Future Work

7.1 Conclusions

In this dissertation, the modeling strategies for dynamic performance prediction for MCPM layout optimization have been presented. The eddy-current regression-based model has been presented in this work to capture the eddy-current impact of the two-sided DBC. This eddy-current effect can have a significant impact on the extraction accuracy and has not been considered in other parasitic extraction studies with MCPM layout optimization.

The PEEC-based and Loop-based methods show accurate results for 2D and 2.5D MCPM layout structures. These models have been hardware validated through measurements of fabricated modules generated from PowerSynth using the Vector Network Analyzer (VNA), impedance analyzer, and Double-Pulse Tests (DPTs). Some initial results of the 3D MCPM structure have also been validated. However, a better measurement setup is needed in the future to guarantee the accuracy of the models.

Both PEEC-based and Loop-based methods have been used in layout optimization. The layout optimization results with various layout footprints also show the stability of these methods. While the eddy-current impact shifts the Pareto-frontier, it shows a similar layout optimization trend when the eddy-current is not considered. However, for more accurate extraction results of the parasitic parameters, the eddy-current impact must be taken into account.

The dissertation also incorporates some model-based engineering (MBE) techniques, where the MSCAD physics-based device model has been reimplemented in Python. The thermal-dependent parameters from this device model enable a temperature-dependent switching loss study. Here, a successive approximation strategy was used to find the final steady-state temperature of the MCPM layout during the dynamic operation of the circuit. Optimization studies on some dynamic parameters such as switching frequencies have shown a more insightful optimization result. The user can select a more valid MCPM layout for the target application based on these results.

Other contributions noted in this work are the implementation, maintenance, and packaging efforts of the PowerSynth tool, which the power electronics field has recognized. Finally, a device parameter extraction tool has been developed thanks to the range of expertise from the MSCAD group. This tool will help modeling and power electronics engineers fit the model for their circuit simulation target.

7.2 Future Work

In the future, the eddy current library must be extended to consider the eddy-current impact of the more complex structures in 3D MCPM layouts. This task is quite challenging now as, unlike 2D MCPM layouts, the 3D MCPM technologies are not quite standardized yet. Furthermore, the PEEC-based method from this work also allows current density extraction of bondwires and vias during loop-extraction. This information is beneficial to electromigration and mechanical reliability studies. A hybrid model between PEEC and the loop-based can be further researched to consider planar-type traces while maintaining the same speed and accuracy achieved by the loop-based approach.

The power loss studies can be extended to consider different circuit topologies. Hence, the same method can be applied to build a library that can compute the various circuit design objectives from the designer. Moreover, considering the physics-based device model in PowerSynth, a library with different devices can be built into the tool. The users can have more design insights on the device ratings, the highest switching frequency, or the heatsink selection. This design consideration would also reshape the solution space, where more reliable layout solutions can be selected. Furthermore, if the user provides the measurement data of each device, an optimized device placement strategy can be performed to ensure a balanced turn-on and turn-off loss time among the devices. This would be beneficial for the layout synthesis algorithm, such as [58].

References

[1] B. Rahrovi and M. Ehsani, "A Review of the More Electric Aircraft Power Electronics," in *2019 IEEE Texas Power and Energy Conference (TPEC)*, 2019, pp. 1–6.

[2] M. Rabiul Islam *et al.*, "State-of-the-Art of the Medium-Voltage Power Converter Technologies for Grid Integration of Solar Photovoltaic Power Plants," *IEEE Trans. Energy Convers.*, vol. 34, no. 1, pp. 372–384, Mar. 2019.

[3] A. Q. Huang, "Wide bandgap (WBG) power devices and their impacts on power delivery systems," in *2016 IEEE International Electron Devices Meeting (IEDM)*, 2016, pp. 20.1.1-20.1.4.

[4] A. Q. Huang, "Power Semiconductor Devices for Smart Grid and Renewable Energy Systems," *Proc. IEEE*, vol. 105, no. 11, pp. 2019–2047, 2017.

[5] H. A. Mantooth *et al.*, "Emerging trends in silicon carbide power electronics design," *CPSS Trans. Power Electron. Appl.*, vol. 2, no. 3, pp. 161–169, 2017.

[6] S. Seal and H. Mantooth, "High Performance Silicon Carbide Power Packaging—Past Trends, Present Practices, and Future Directions," *Energies*, vol. 10, no. 3, p. 341, 2017.

[7] 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, 2017.

[8] Z. Zeng, X. Zhang, and X. Li, "Layout-Dominated Dynamic Current Imbalance in Multichip Power Module: Mechanism Modeling and Comparative Evaluation," *IEEE Trans. Power Electron.*, vol. 34, no. 11, pp. 11199–11214, 2019.

[9] F. Yang *et al.*, "Design of a low parasitic inductance SiC power module with double-sided cooling," in *2017 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2017, pp. 3057–3062.

[10] D. P. Sadik *et al.*, "Analysis of Parasitic Elements of SiC Power Modules with Special Emphasis on Reliability Issues," *IEEE J. Emerg. Sel. Top. Power Electron.*, vol. 4, no. 3, pp. 988–995, 2016.

[11] A. B. Jørgensen, S. Munk-Nielsen, and C. Uhrenfeldt, "Overview of Digital Design and Finite-Element Analysis in Modern Power Electronic Packaging," *IEEE Trans. Power Electron.*, vol. 35, no. 10, pp. 10892–10905, 2020.

[12] H. Lee, V. Smet, and R. Tummala, "A Review of SiC Power Module Packaging Technologies: Challenges, Advances, and Emerging Issues," *IEEE J. Emerg. Sel. Top. Power Electron.*, vol. 8, no. 1, pp. 239–255, 2020.

[13] G. Bower *et al.*, "SiC power electronics packaging prognostics," *IEEE Aerosp. Conf. Proc.*, 2008.

[14] U. Khayam, "Partial discharge detection system in power module of electric vehicle," in *2017 4th International Conference on Electric Vehicular Technology (ICEVT)*, 2017, pp. 154–158.

[15] N. Hingora *et al.*, “Power-CAD: A novel methodology for design, analysis and optimization of Power Electronic Module layouts,” in *Energy Conversion Congress and Exposition (ECCE), 2010 IEEE*, 2010, pp. 2692–2699.

[16] Q. Le *et al.*, “Response surface modeling for parasitic extraction for multi-objective optimization of multi-chip power modules (MCPMs),” in *2017 IEEE 5th Workshop on Wide Bandgap Power Devices and Applications (WiPDA)*, 2017, pp. 327–334.

[17] I. Al Razi *et al.*, “Constraint-Aware Algorithms for Heterogeneous Power Module Layout Synthesis and Reliability Optimization,” *2018 IEEE 6th Work. Wide Bandgap Power Devices Appl. WiPDA 2018*, pp. 323–330, 2018.

[18] I. Al Razi *et al.*, “PowerSynth Design Automation Flow for Hierarchical and Heterogeneous 2.5-D Multichip Power Modules,” *IEEE Trans. Power Electron.*, vol. 36, no. 8, pp. 8919–8933, 2021.

[19] Q. Le *et al.*, “PowerSynth Integrated CAD flow for High Density Power Modules,” in *Design Methodologies for Power Electronics (DMC)*, 2021.

[20] Q. Le *et al.*, “Fast and Accurate Inductance Extraction for Power Module Layout Optimization Using Loop-Based Method,” in *2021 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2021, pp. 1358–1365.

[21] T. M. Evans *et al.*, “Electronic Design Automation (EDA) Tools and Considerations for Electro-Thermo-Mechanical Co-Design of High Voltage Power Modules,” in *2020 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2020, pp. 5046–5052.

[22] Z. Gong, “Thermal and electrical parasitic modeling for multi-chip power module layout Synthesis,” University of Arkansas, 2012.

[23] B. W. Shook *et al.*, “Multi-objective layout optimization for multi-chip power modules considering electrical parasitics and thermal performance,” in *2013 IEEE 14th Workshop on Control and Modeling for Power Electronics, COMPEL*, 2013.

[24] W. Ellens *et al.*, “Effective graph resistance,” *Linear Algebra Appl.*, vol. 435, no. 10, pp. 2491–2506, 2011.

[25] T. M. Evans *et al.*, “PowerSynth: A Power Module Layout Generation Tool,” *IEEE Trans. Power Electron.*, vol. 34, no. 6, pp. 5063–5078, 2019.

[26] A. R. Djordjevic and T. K. Sarkar, “Closed-form formulas for frequency-dependent resistance and inductance per unit length of microstrip and strip transmission lines,” *IEEE Trans. Microw. Theory Tech.*, vol. 42, no. 2, pp. 241–248, 1994.

[27] C. L. Holloway and E. F. Kuester, “Net and partial inductance of a microstrip ground plane,” *IEEE Trans. Electromagn. Compat.*, vol. 40, no. 1, pp. 33–46, 1998.

[28] H. A. Wheeler, “Transmission-Line Properties of a Strip on a Dielectric Sheet on a Plane,” *IEEE Trans. Microw. Theory Tech.*, vol. 25, no. 8, pp. 631–647, 1977.

[29] A. M. Niknejad and R. G. Meyer, “Analysis of eddy-current losses over conductive substrates with applications to monolithic inductors and transformers,” *IEEE Trans.*

Microw. Theory Tech., vol. 49, no. 1, pp. 166–176, 2001.

[30] E. A. Jones, F. F. Wang, and D. Costinett, “Review of Commercial GaN Power Devices and GaN-Based Converter Design Challenges,” *IEEE J. Emerg. Sel. Top. Power Electron.*, vol. 4, no. 3, pp. 707–719, 2016.

[31] A. Dutta and S. S. Ang, “Electromagnetic interference simulations for wide-bandgap power electronic modules,” *IEEE J. Emerg. Sel. Top. Power Electron.*, vol. 4, no. 3, pp. 757–766, 2016.

[32] A. E. Ruehli, “Inductance Calculations in a Complex Integrated Circuit Environment,” *IBM J. Res. Dev.*, vol. 16, no. 5, pp. 470–481, 1972.

[33] M. Kamon, M. J. Tsuk, and J. K. White, “FASTHENRY: a multipole-accelerated 3-D inductance extraction program,” *IEEE Trans. Microw. Theory Tech.*, vol. 42, no. 9, pp. 1750–1758, 1994.

[34] P. Ning *et al.*, “A fast universal power module layout method,” in *2015 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2015, pp. 4132–4137.

[35] L. Wang *et al.*, “Current-Bunch Concept for Parasitic-Oriented Extraction and Optimization of Multichip SiC Power Module,” *IEEE Trans. Power Electron.*, vol. 36, no. 8, pp. 8593–8599, 2021.

[36] Y. Zhou *et al.*, “Graph-Model-Based Generative Layout Optimization for Heterogeneous SiC Multichip Power Modules With Reduced and Balanced Parasitic Inductance,” *IEEE Trans. Power Electron.*, vol. 37, no. 8, pp. 9298–9313, 2022.

[37] L. Horowitz *et al.*, “A Method of Partial Inductances to Evaluate and Optimize Switching Cells,” in *2021 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2021, pp. 1549–1554.

[38] P. Ning *et al.*, “Automatic Layout Design for Power Module,” *IEEE Trans. Power Electron.*, vol. 28, no. 1, pp. 481–487, 2013.

[39] J. Wang, H. S. Chung, and R. T. Li, “Characterization and Experimental Assessment of the Effects of Parasitic Elements on the MOSFET Switching Performance,” *IEEE Trans. Power Electron.*, vol. 28, no. 1, pp. 573–590, 2013.

[40] P. Nayak *et al.*, “Study of the effects of parasitic inductances and device capacitances on 1200 V, 35 A SiC MOSFET based voltage source inverter design,” in *2014 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, 2014, pp. 1–6.

[41] H. Li *et al.*, “A novel DBC layout for current imbalance mitigation in SiC MOSFET multichip power modules,” in *2016 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2016, pp. 704–708.

[42] J. Wang *et al.*, “A Transient 3-D Thermal Modeling Method for IGBT Modules Considering Uneven Power Losses and Cooling Conditions,” *IEEE J. Emerg. Sel. Top. Power Electron.*, vol. 9, no. 4, pp. 3959–3970, 2021.

[43] J. Wang *et al.*, “A Novel Approach to Model and Analyze Uneven Temperature Distribution Among Multichip High-Power Modules and Corresponding Method to Respecify Device SOA,” *IEEE Trans. Power Electron.*, vol. 37, no. 4, pp. 4626–4640, 2022.

[44] R. Khanna *et al.*, “An Analytical Model for Evaluating the Influence of Device Parasitics on Cdv / dt Induced False Turn-on in SiC MOSFETs,” *2013 Twenty-Eighth Annu. IEEE Appl. Power Electron. Conf. Expo.*, pp. 518–525, 2013.

[45] M. R. Ahmed, R. Todd, and A. J. Forsyth, “Predicting SiC MOSFET Behavior under Hard-Switching, Soft-Switching, and False Turn-On Conditions,” *IEEE Trans. Ind. Electron.*, vol. 64, no. 11, pp. 9001–9011, 2017.

[46] A. Lemmon *et al.*, “Instability in Half-Bridge Circuits Switched With Wide Band-Gap Transistors,” *IEEE Trans. Power Electron.*, vol. 29, no. 5, pp. 2380–2392, 2014.

[47] J. D. Jackson, *Classical Electrodynamics*, 3rd ed. New York, NY: Wiley, 1999.

[48] C. Hoer and C. Love, “Exact Inductance Equations for Rectangular Conductors with Applications to more Complicated Geometries,” *J. Res. Natl. Bur. Stand. Sect. C Eng. Instrum.*, vol. 69C, p. 127, 1965.

[49] R. Escovar, “TOOLS FOR IMPEDANCE EXTRACTION IN INTEGRATED CIRCUITS (IC),” 2006.

[50] Q. Le *et al.*, “Fast and Accurate Parasitic Extraction in Multichip Power Module Design Automation Considering Eddy-Current Losses,” *IEEE J. Emerg. Sel. Top. Power Electron.*, p. 1, 2022.

[51] F. Pedregosa *et al.*, “Scikit-Learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, no. null, pp. 2825–2830, Nov. 2011.

[52] B. Krauter and S. Mehrotra, “Layout based frequency dependent inductance and resistance extraction for on-chip interconnect timing analysis,” in *Proceedings 1998 Design and Automation Conference. 35th DAC. (Cat. No.98CH36175)*, 1998, pp. 303–308.

[53] J. E. Stine *et al.*, “FreePDK v2.0: Transitioning VLSI education towards nanometer variation-aware designs,” in *2009 IEEE International Conference on Microelectronic Systems Education*, 2009, pp. 100–103.

[54] J. E. Stine *et al.*, “FreePDK: An Open-Source Variation-Aware Design Kit,” in *2007 IEEE International Conference on Microelectronic Systems Education (MSE’07)*, 2007, pp. 173–174.

[55] J. Main, “A Manufacturer Design Kit for Multi-Chip Power Module Layout Synthesis,” Univerisy of Arkansas, 2017.

[56] I. Al Razi *et al.*, “Electromigration-Aware Reliability Optimization of MCPM Layouts Using PowerSynth,” in *IEEE Energy Conversion Congress and Exposition (ECCE)*, 2022.

[57] 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, 2019.

[58] T. M. Evans, Y. Peng, and H. A. Mantooth, “Placement and Routing for Power Module Layout,” in *2021 IEEE Design Methodologies Conference (DMC)*, 2021, pp. 1–6.

[59] C.-W. Ho, A. Ruehli, and P. Brennan, “The modified nodal approach to network analysis,” *IEEE Trans. Circuits Syst.*, vol. 22, no. 6, pp. 504–509, 1975.

[60] L. J. Albert E. Ruehli, Giulio Antonini, “Circuit Analysis for PEEC Methods,” in *Circuit Oriented Electromagnetic Modeling Using the Peec Techniques*, John Wiley & Sons, Ltd, 2017, pp. 9–46.

[61] J. Vlach and K. Singhal, *Computer methods for circuit analysis and design*. New York, NY: VAN NOSTRAND REINHOLD COMPANY, 1983.

[62] L. W. Nagel, *SPICE2 : a computer program to simulate semiconductor circuits*. Berkeley, CA: Electronics Research Laboratory, College of Engineering, University of California, 1975.

[63] 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.

[64] L. W. Nagel, “SPICE2: A computer program to simulate semiconductor circuits,” *Ph. D. Diss. Univ. Calif. Berkeley*, 1975.

[65] A. Musing, J. Ekman, and J. W. Kolar, “Efficient Calculation of Non-Orthogonal Partial Elements for the PEEC Method,” *IEEE Trans. Magn.*, vol. 45, no. 3, pp. 1140–1143, 2009.

[66] Q. Le *et al.*, “PEEC Method and Hierarchical Approach Towards 3D Multichip Power Module (MCPM) Layout Optimization,” in *2019 IEEE International Workshop on Integrated Power Packaging (IWIPP)*, 2019, pp. 131–136.

[67] R. Escovar, S. Ortiz, and R. Suaya, “An improved long distance treatment for mutual inductance,” *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 24, no. 5, pp. 783–793, 2005.

[68] Y. Peng, D. Petranovic, and S. K. Lim, “Chip/package co-analysis and inductance extraction for fan-out wafer-level-packaging,” in *2017 IEEE 26th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, 2017, pp. 1–3.

[69] S. Yu *et al.*, “Loop-based inductance extraction and modeling for multiconductor on-chip interconnects,” *IEEE Trans. Electron Devices*, vol. 53, no. 1, pp. 135–145, 2006.

[70] X. Huang *et al.*, “Loop-based interconnect modeling and optimization approach for multigigahertz clock network design,” *IEEE J. Solid-State Circuits*, vol. 38, no. 3, pp. 457–463, 2003.

[71] R. Kumar *et al.*, “Characterization and modeling of CMOS on-chip coupled interconnects,” in *ESSDERC 2007 - 37th European Solid State Device Research Conference*, 2007, pp. 159–162.

[72] T. H. Cormen *et al.*, *Introduction to Algorithms, Third Edition*, 3rd ed. The MIT Press, 2009.

[73] I. Kovacevic-Badstuebner *et al.*, “Parasitic Extraction Procedures for SiC Power Modules,”

in *CIPS 2018; 10th International Conference on Integrated Power Electronics Systems*, 2018, pp. 1–6.

[74] H. Daou *et al.*, “Dynamic electric model for IGBT power module based on Q3D® and Simplorer®: 3D Layout design, stray inductance estimation, experimental verifications,” in *2016 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles & International Transportation Electrification Conference (ESARS-ITEC)*, 2016, pp. 1–6.

[75] B. Aberg *et al.*, “Estimation and minimization of power loop inductance in 135 kW SiC traction inverter,” in *2018 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2018, pp. 1772–1777.

[76] M. M. Hossain *et al.*, “An Improved Physics-based LTSpice Compact Electro-Thermal Model for a SiC Power MOSFET with Experimental Validation,” in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018, pp. 1011–1016.

[77] Z. Wang *et al.*, “Fixtures Design Considerations for Impedance Measurement,” in *2021 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2021, pp. 1555–1559.

[78] B. Nelson *et al.*, “Measurement-Based Modeling of Power Module Parasitics with Increased Accuracy,” in *2020 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2020, pp. 1430–1437.

[79] “E4990A Impedance Analyzer.” [Online]. Available: https://www.keysight.com/us/en/assets/7018-04256/data-sheets/5991-3890.pdf.

[80] “ARL ParaPower,” 2019. [Online]. Available: https://github.com/USArmyResearchLab/ParaPower.

[81] I. Al Razi *et al.*, “Hierarchical Layout Synthesis and Optimization Framework for High-Density Power Module Design Automation,” in *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, 2021, pp. 1–8.

[82] M. Mudholkar *et al.*, “Datasheet Driven Silicon Carbide Power MOSFET Model,” *IEEE Trans. Power Electron.*, vol. 29, no. 5, pp. 2220–2228, 2014.

[83] A. U. Rashid *et al.*, “Datasheet-Driven Compact Model of Silicon Carbide Power MOSFET Including Third-Quadrant Behavior,” *IEEE Trans. Power Electron.*, vol. 36, no. 10, pp. 11748–11762, 2021.

[84] A. Zhou, Q. Zhang, and G. Zhang, “A multiobjective evolutionary algorithm based on decomposition and probability model,” in *2012 IEEE Congress on Evolutionary Computation*, 2012, pp. 1–8.

[85] R. M. Kotecha *et al.*, “Compact Modeling of High-Voltage Gallium Nitride Power Semiconductor Devices for Advanced Power Electronics Design,” *IEEE Open J. Power Electron.*, vol. 2, pp. 75–87, 2021.

[86] P. Wilson and H. A. Mantooth, *Model-Based Engineering for Complex Electronic Systems*. Elsevier, 2013.

[87] K. Sheng, “Maximum Junction Temperatures of SiC Power Devices,” *IEEE Trans. Electron Devices*, vol. 56, no. 2, pp. 337–342, 2009.

[88] W. Zhou, X. Zhong, and K. Sheng, “High temperature stability evaluation of SiC MOSFETs,” in *2014 IEEE 26th International Symposium on Power Semiconductor Devices & IC’s (ISPSD)*, 2014, pp. 305–308.

[89] D. Christen and J. Biela, “Analytical Switching Loss Modeling Based on Datasheet Parameters for mosfets in a Half-Bridge,” *IEEE Trans. Power Electron.*, vol. 34, no. 4, pp. 3700–3710, 2019.

[90] Q. Le *et al.*, “Fast transient thermal and power dissipation modeling for multi-chip power modules: A preliminary assessment of different electro-thermal evaluation methods,” in *Control and Modeling for Power Electronics (COMPEL), 2016 IEEE 17th Workshop on*, 2016, pp. 1–6.

[91] Y. Nakamura *et al.*, “Electrothermal Cosimulation for Predicting the Power Loss and Temperature of SiC MOSFET Dies Assembled in a Power Module,” *IEEE Trans. Power Electron.*, vol. 35, no. 3, pp. 2950–2958, 2020.

Appendix A: Benchmark for different programming approaches for mutual inductance calculation

Python is a high-level language that has been built using C++. Each and every objects in the Python language is based on PythonObject object written in C++. At the runtime, the Python code is interpreted line-by-line and translated to machine code rather than being compiled like C, C++, or Java. Because of line-by-line translation for extensive number of computations of partial elements such as inductance, mutual inductance, and resistance, it is very inefficient. To avoid this inefficiency, there are two alternative approaches which has been applied in the PowerSynth source code for parasitic computation. Two of such approaches are using Cython or Numba packages in Python. An efficiency analysis has been performed for different implementation of Eq. (2.15). From these results, it would be helpful for future PowerSynth students to select the most appropriate programing approach.

The Cython package requires C code compilation into machine code prior to the evaluation, while the Numba package allows a just in time (JIT) compilation. Here the Python code is compiled into machine code at runtime. Table A-1 shows the speed comparison for the mutual inductance evaluation using Equation (2.15) among pure Python, Cython and Numba using a single core Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz. As shown in the results, the pure Python code has the worst performance of 250× slower than both Numba and Cython. The JIT compiler from Numba shows better performance than Cython for 100-10000 evaluations and a bit slower for >100000 evaluation. Additionally, the JIT compiler is much easier to implement and maintain in the source code compared to Cython. For example, recompilation is needed if there is the change in the operating system. Thus, Numba JIT has been selected to compute the self-inductance and mutual inductance matrices to characterize the regression models. One limitation of Numba approach is that it is quite complicated to use multiple processes for the evaluation speedup. This needs to be investigated in the future to improve the evaluation efficiency.

Table A-1 Evaluation Time for Different Programming Approaches for Equation (2.15)

	Evaluation Time (s)		
Evaluation	**Numba-JIT**	**Cython**	**Python**
1e2	7.50E-04	0.0554	0.198
1e3	0.0074	0.054	2.045
1e4	0.077	0.093	**19.68**
1e5	0.75	0.44	**194**
1e6	7.83	2.66	**1979**

Appendix B: Circuit Stamping for some selected elements

In this section, some of the circuit stamping for some basic elements in the Modified Nodal Analysis is presented. This information can be fine in many circuit books. However, some selected elements used in PowerSynth is shown here to make it easier for the reader to understand the method.

Table B-1 Some MNA elements used in PowerSynth

<table>
<tr><th>Element</th><th>Netlist Representation</th><th>LHS and RHS in MNA</th></tr>
<tr><td>Resistor</td><td>**R** ni nk {R}</td><td>
<table>
<tr><th>Row/Column</th><th>V_i</th><th>V_j</th><th>RHS</th></tr>
<tr><td>**i**</td><td>1/R</td><td>-1/R</td><td>0</td></tr>
<tr><td>**j**</td><td>-1/R</td><td>1/R</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>Inductor (self and mutual)</td><td>**L1** ni nk {L1}
L2 nm nn {L2}
M12 L1 L2 {M12l}</td><td>
<table>
<tr><th>R/C</th><th>V_i</th><th>V_k</th><th>V_m</th><th>V_n</th><th>I_{L1}</th><th>I_{L2}</th><th>RHS</th></tr>
<tr><td>**i**</td><td></td><td></td><td></td><td></td><td>1</td><td></td><td>0</td></tr>
<tr><td>**k**</td><td></td><td></td><td></td><td></td><td>-1</td><td></td><td>0</td></tr>
<tr><td>**m**</td><td></td><td></td><td></td><td></td><td></td><td>1</td><td>0</td></tr>
<tr><td>**n**</td><td></td><td></td><td></td><td></td><td></td><td>-1</td><td>0</td></tr>
<tr><td>**Aux**</td><td>1</td><td>-1</td><td></td><td></td><td>-sL1</td><td>-sM12</td><td>0</td></tr>
<tr><td>**Aux**</td><td></td><td></td><td>1</td><td>-1</td><td>-sM12</td><td>-sL2</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>Capacitor</td><td>**C** ni nk {C}</td><td>
<table>
<tr><th>Row/Column</th><th>V_i</th><th>V_j</th><th>RHS</th></tr>
<tr><td>**i**</td><td>1/Cs</td><td>-1/Cs</td><td>0</td></tr>
<tr><td>**j**</td><td>-1/Cs</td><td>1/Cs</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>Independent Current Source</td><td>**Is** ni nj {Is}</td><td>
<table>
<tr><th>Row/Column</th><th>V_i</th><th>V_j</th><th>RHS</th></tr>
<tr><td>**i**</td><td></td><td></td><td>-Is</td></tr>
<tr><td>**j**</td><td></td><td></td><td>Is</td></tr>
</table>
</td></tr>
</table>

Independent Voltage Source	**Vs** ni nj {Vs}	**Row/Column**	**V_i**	**V_j**	**I**	**RHS**
		i			1	0
		j			-1	0
		Aux	1	-1	0	V_s