Automated PCB Parasitics Extraction from EDA Tools for Power Electronics Design Support

Sven Fießer[1], Ulf Schwalbe[2]

[1] Technische Universität Ilmenau, Germany
[2] Hochschule Fulda, Germany

Corresponding author: Sven Fießer, sven.fiesser@tu-ilmenau.de

Abstract

This paper illustrates the automated extraction of parasitic components on printed circuit boards (PCB) from common EDA tools. An algorithm reads the EDA files and feeds them to the open source software FastHenry2 and FasterCap. Those two programs are calculating the frequency-dependent resistances, self and mutual inductances of the PCB tracks as well as the capacitances between those. This information is displayed during the design process of the PCB and the effects of altering PCB tracks will be shown directly. The main aim of this algorithm is to give PCB designers support for an optimized layout for high and low voltage applications in especially high frequency switching domains. The accuracy of the calculated parasitics is verified by experimental results.

1 Introduction

Designing a proper PCB layout is a crucial task in most electronic applications. Having low parasitic components on the PCB is especially of utmost importance in fast switching power electronics devices using GaN or SiC semiconductors. While using such high-performance semiconductors the PCB designer's goal is often to improve the overall power density by increasing the switching frequency and so the influences of parasitic components are getting more pronounced. And even if the PCB designer is acquainted with most PCB design guidelines, the development of a PCB undergoes usually several design loops until a sufficiently good layout is created.

In order to speed up the PCB design process, this paper presents an algorithm for the extraction of parasitic components of a PCB directly from the designer's EDA tool, e.g. Eagle from Autodesk. The extracted parasitics will be shown to the PCB designer in a pop-up window on his PC while he is routing the PCB. Changes made to the layout and so to the parasitic components will be shown on the fly, so those changes can be evaluated directly.

2 Method for parasitic extraction

For the calculation of the frequency-dependent resistances, self and mutual inductances FastHenry2 and for the capacitances FasterCap are used. FastHenry2 is a Windows porting of FastHenry, a three-dimensional inductance extraction program which uses magnetoquasistatic approximations for the solution of Maxwell equations [1] [2]. FasterCap on the other hand is a Windows porting of FastCap, a solver for Maxwell equations for the extraction of capacitances, but also includes automatic mesh refinement for easier use in contrast to FastCap [3].

In order to calculate the inductances and resistances, FastHenry2 needs an input file that specifies every conductor of the PCB as a sequence of straight segments, which are themselves interconnected through nodes. Those nodes contain the information about their location in a three-dimensional space. To calculate frequency-dependent effects like the skin effect, a discretization of each segment needs to be done. For accurately computing the inductances and resistances at the highest frequency to be calculated, the smallest filament should be equal to the skin depth at the given frequency. Polygons

ISBN 978-3-8007-5515-8

or ground planes can be modeled in FastHenry2 with the included plane function, which the program itself converts to segments, including the chosen discretization.

In contrast to FastHenry2, FasterCap needs another kind of input file for the extraction of capacitances. Each copper and substrate section of the PCB must be modeled as three-dimensional squares or triangles. The dielectric properties for each structure needs to be specified.

Unfortunately, most common EDA tools for PCB design do not offer a FastHenry2 or FasterCap export function, so a converter was programmed, which handles the conversion of the board file to the corresponding input files. Chapters 2.1 and 2.2 present the workflow of the two converters.

2.1 Converter for FastHenry2

This and the following chapter describe the conversion processes for the EDA tool EAGLE from Autodesk. This tool was chosen due to its widespread use in the electronics industry and for the fact, that the program is freely available for academic personnel and students. The conversion and parasitic extraction process is easily translatable to other EDA tools, as long as the board files can be read using suitable parsing software packages.

The conversion to the FastHenry2 input file is fairly straightforward due to the structure of the EAGLE board file. The whole workflow is depicted in Fig. 1 and shows the two main sub-processes ”extraction” and ”conversion” of the programmed converter, as well as the needed input information and the output file, which itself is the input file for FastHenry2.

The .brd-file is by its nature a XML-file and can be parsed using common software packages. Because each element (i.e., active and passive components on the PCB), wire, via, and polygon is listed in this file separately, the extraction of these information can be done easily. After the extraction of the relevant information from the PCB file, it needs to be converted to the FastHenry2 input file (file extension ”.inp”). This is done firstly by converting the coordinates of the elements and wires to the so called ”nodes” in FastHenry2. Subsequently, each wire of the PCB file will be converted to an individual

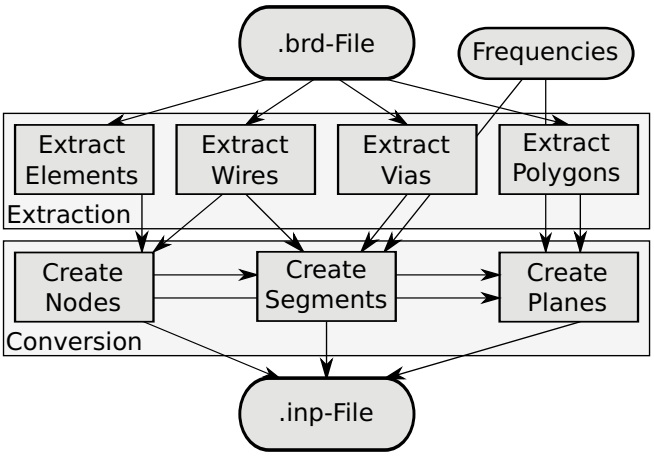

Fig. 1: Algorithm for the FastHenry2 converter.

segment by matching the wire information to the already created nodes. The same will be done with vias, which will be handled like copper wires through the PCB.

In FastHenry2 only planes can’t be converted as easily as wires and vias due to the limitations of the program. Planes can only be modeled as squares and in order to represent a polygon or a ground plane where wires cross it, holes need to be ”cut” from it at the appropriate locations. To model the planes, the dimensions of each polygon in EAGLE will be extracted and then matched to each segment and node in order to find overlapping wires. Holes for the planes will be created accordingly. Even if FastHenry2 doesn’t offer an inbuilt polygon function, a sufficiently good polygon with cut-outs can be represented in it by choosing a suitable high discretization of the planes and their holes.

Lastly, the frequency-dependent effects need to be taken into account for each segment and plane. This will be done by discretizing each segment in their height-direction (nhinc) and width-direction (nwinc) into filaments as shown in Fig. 2. In order to correctly represent the skin effect, the width and height of the smallest filament needs to be smaller than the skin depth [2].

For calculating nhinc and nwinc the two equations Eq. (1) and Eq. (2) need to be solved. Equation (1) calculates the skin depth at f, the highest to be simulated frequency. σ and μ are the conductivity and the permeability of the material respectively. Equation (2) shows the calculation for nwinc for the width W. Substituting W with the height

ISBN 978-3-8007-5515-8

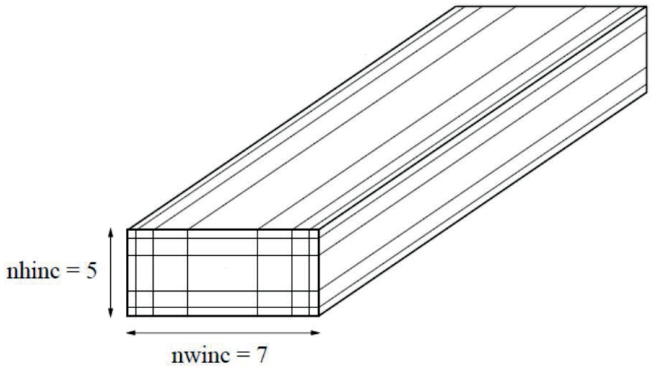

Fig. 2: Example of the discretization of a segment in FastHenry2 (source: [4]).

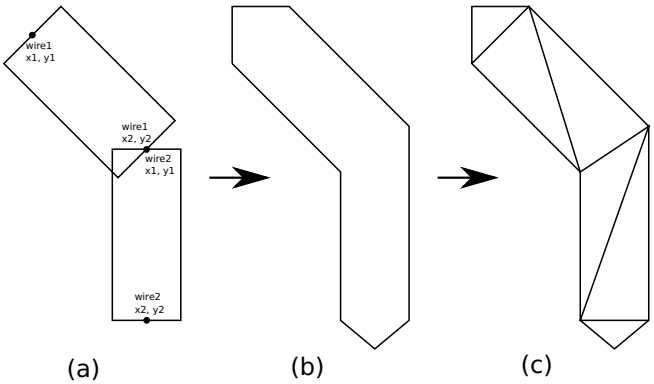

Fig. 3: Meshing algorithm of the FasterCap converter. (a) Wire information from the board file, (b) creation of uniform polygon, (c) meshing.

results in nhinc. Due to the exponential distribution of the current in a conductor, r needs to be set at approximately e. The equation needs to be calculated several times, starting with nwinc/nhinc=1 and incrementing the value by one until the result of the formula is smaller than the skin depth at the highest frequency.

$$\delta = \sqrt{\frac{1}{\pi f \sigma \mu}} \tag{1}$$

$$\delta > \begin{cases} \dfrac{W}{2\sum_{i=0}^{nwinc/2-1} r^i} & \text{if } nwinc = \text{even}, \\ \dfrac{W}{r^{\frac{nwinc-1}{2}} + 2\sum_{i=0}^{(nwinc-1)/2-1} r^i} & \text{if } nwinc = \text{odd}. \end{cases} \tag{2}$$

After compiling all the necessary data from the board file, the converter creates the input file for FastHenry2, which can then be executed by hand or automatically as explained in chapter 3.

2.2 Converter for FasterCap

As mentioned in chapter 2, the PCB needs to be modeled from three-dimensional planar squares or triangles in FasterCap, including the dielectric properties of each material. In case of a PCB, copper traces need to be modeled as ”conductors” and the PCB substrate and the surrounding air as ”dielectrics”. After extracting the locations and dimensions of each wire and polygon from the EAGLE board file, the wires will be modeled by firstly creating a uniform polygon of the conductor and then meshing the three-dimensional

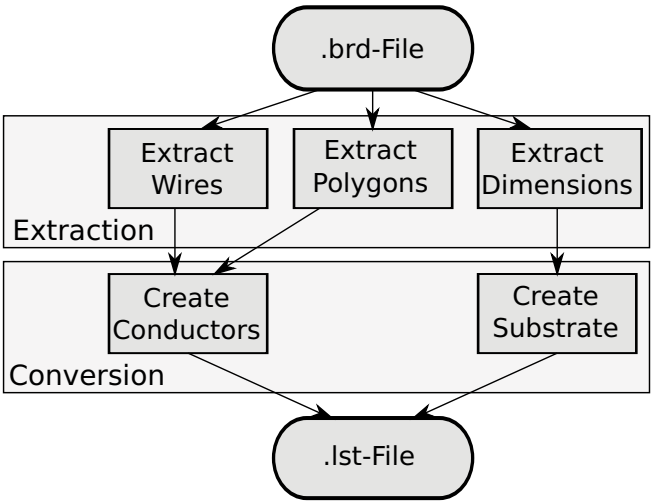

Fig. 4: Algorithm for the FasterCap converter.

information to triangles using the open source 3D mesh generator Gmsh [5]. Figure 3 shows the meshing process for an exemplary conductor with two wires. For creating a three-dimensional representation of a wire or polygon, this meshed plane will be used twice in FasterCap as the top and bottom of the conductor, set apart by the copper thickness. A side wall around the conductor closes the figure.

This procedure prevents the possibility that any of the triangles overlap and interfere with the solver of FasterCap. Polygons in EAGLE, as well as the substrate, will be handled in the same manner as conductors. It is only mandatory to cut out all crossing wires from the polygon. Figure 4 depicts the whole extraction and conversion process.

After the meshing of each PCB file element, the converter creates the input file for FasterCap.

ISBN 978-3-8007-5515-8

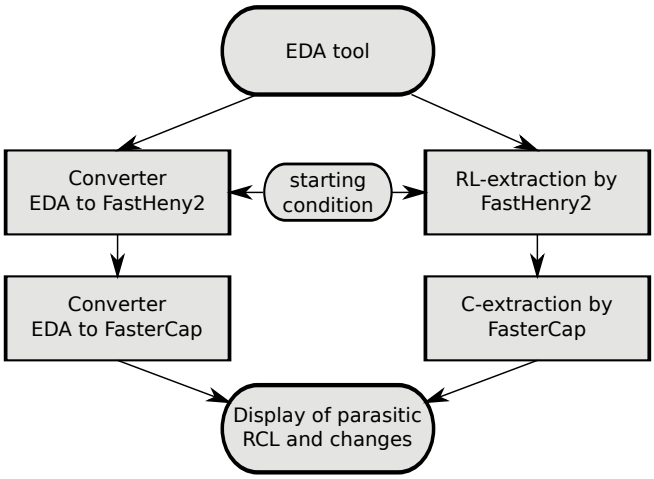

Fig. 5: Workflow of the automation process.

3 Automated RLC-extraction

The main benefit of the in this paper proposed algorithm is the automated extraction of the parasitic components while the engineer is routing the PCB and displaying the calculated parameters. In order to automate the extraction process, another program was written, which starts the converters for FastHenry2 and FasterCap automatically, either at specified time intervals or when the board file is saved. After this, the program starts the two simulation programs and waits until they are finished. The results of the simulation processes will be shown to the PCB designer using an easy to understand graphical user interface.

Especially useful is the automation interface implemented in both FastHenry2 and FasterCap, which allows to remotely launch the programs, perform the simulation, and retrieve the results.

The complete workflow of the automation algorithm including the two converter programs is depicted in Fig. 5.

4 Verification

To verify the simulation results of FastHenry2 and FasterCap a PCB for a boost converter was routed in EAGLE and its parasitic components with the here described algorithms were extracted. The capacitances calculated by FasterCap were added into a SPICE simulation as frequency-independent capacitors. To simulate the frequency-dependent inductances and resistances in SPICE, the inbuilt

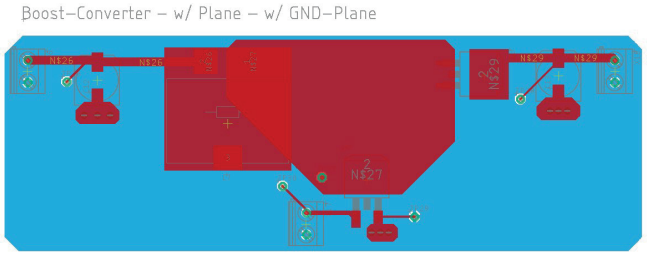

Fig. 6: In Autodesk EAGLE routed boost converter used for the verification.

FastHenry2 function for the generation of a reduced order model was used. This function creates a SPICE subcircuit, that is valid for the whole simulated frequency range [4] [6].

Figure 6 shows the routed boost converter in EAGLE and Fig. 7 depicts the LTSpice simulation used for the verification. It is noted, that in the simulation file only the parasitic components in the commutation path were added to simplify the simulation. The values for the capacitances, the inductor, and the MOSFET driver were taken from the corresponding data sheets. The simulation models for the MOSFET and the diode were provided by the manufacturers.

Comparing the SPICE results with measurements of the actual boost converter resulted in a high accuracy of the simulation. Both the measured overshoot of the MOSFET drain source voltage, as well as the frequency of the ringing are almost identical to the simulation. Solely the attenuation of the ringing could not be predicted with complete confidence, which might be due to further contact resistances or the simulation models provided by the manufacturers. Figure 8 shows the simulated and measured drain-source-voltage of the MOSFET.

Apart from the good prediction of the ringing on the MOSFET, the inclusion of the parasitic components in the SPICE simulation increased the accuracy of the total loss calculation incurred in the boost converter. Simulating the boost converter using only the data sheet data and the manufacturers SPICE model resulted in a relative error of 5.00% compared with the actual losses of the converter. By adding the PCB parasitic components the relative error decreases to 3.80%. The measured data is shown in table 1.

ISBN 978-3-8007-5515-8

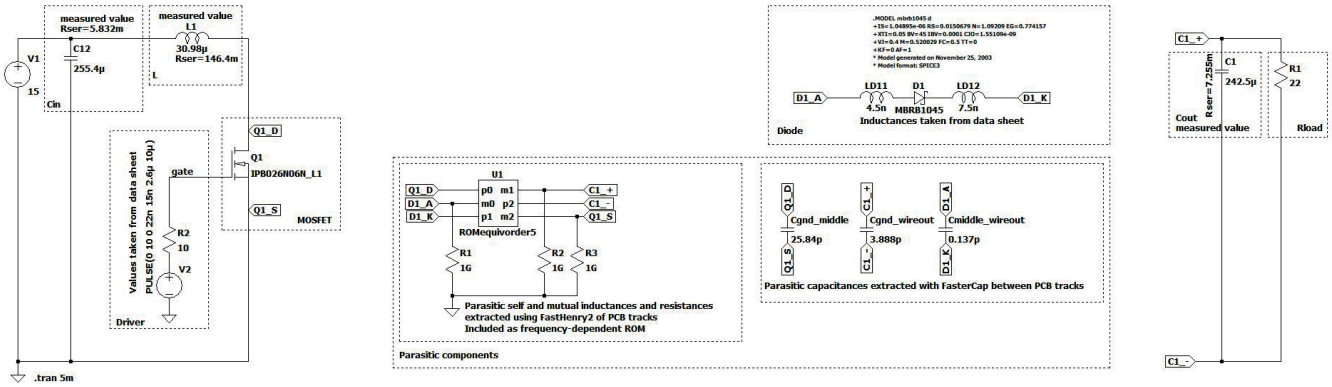

Fig. 7: LTSpice simulation used for the verification.

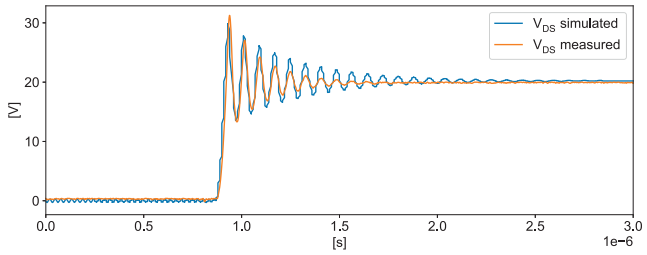

Fig. 8: Simulated and measured turn-off drain-source-voltage of the MOSFET of the boost converter. 100 kHz switching frequency at duty cycle of 0.26.

	Verification measure-ment	Simulation with PCB parasitics	Simulation without PCB parasitics
U_{in}	15.021	15.000	15.000
I_{in}	1.204	1.233	1.234
P_{in}	18.081	18.501	18.507
U_{out}	20.014	19.785	19.793
I_{out}	0.867	0.899	0.900
P_{out}	17.346	17.793	17.807
Pv	0.735	0.708	0.700
rel. error		3.80%	5.00%

Tab. 1: Comparison of measurement and simulation data of a hard-switching boost converter at 100 kHz switching frequency at a duty cycle of 0.26.

5 Conclusion

This paper outlines a procedure which extracts the parasitic components of a PCB design easily and accurately. A converter reads the design files of the EDA tool and translates them to FastHenry2 and FasterCap input files automatically. Those programs then calculate the parasitic components of the PCB and the results are shown to the designer while routing the PCB. The verification with a boost converter showed that those simulated parasitic components are matching the measured results very accurately.

While the use of wide-bandgap semiconductors in power electronics applications is getting increasingly prevalent, a proper PCB layout is all the more important to consider. With the here presented approach, the design process of such a PCB can be reduced significantly by pointing out the effects of rerouting a PCB track or adding a ground plane for example. This is supposed to reduce the number of design iterations and therefore time and money can be saved.

Based on this approach, a further improvement might be to detect and mark the commutation path in a power electronics converter automatically and extract only its parasitic components. This can reduce the simulation time and make the algorithm more user-friendly for inexperienced PCB designers.

References

[1] FastFieldSolvers S.R.L., “FastHenry2,” 2019. [Online]. Available: https://www.fastfieldsolvers.com/fasthenry2.htm.

[2] P. Jacqmaer, J. Zwysen, R. Galagaev, and J. Driesen, “Accurately modelling of parasitics in power electronics circuits using an easy RLC

extraction method," in *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, 2012. DOI: 10.1109/I2MTC.2012.6229327.

[3] FastFieldSolvers S.R.L., "FasterCap," 2019. [Online]. Available: https://www.fastfieldsolvers.com/fastercap.htm.

[4] M. Kamon, L. Silveira, C. Smithhisler, and J. White, *Fasthenry user's guide*, 3.0. Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, Research Laboratory of Electronics, 1996.

[5] C. Geuzaine and J. Remacle, *Gmsh Reference Manual*. 2020.

[6] L. M. Silveira, M. Kamon, and J. White, "Efficient Reduced-Order Modeling of Frequency-Dependent Coupling Inductances Associated with 3-D Interconnect Structures," *IEEE Transactions on Components, Packaging, And Manufacturing Technology: Part B*, vol. 19, no. 2, pp. 283–288, 1996. DOI: 10.1109/96.496030.

ISBN 978-3-8007-5515-8