CAD TOOLS FOR HYBRID INTEGRATION

Radhakrishnan Balakrishnan
Shijith Kunneth Kesavan

This thesis is presented as a part of Degree of Masters of Science in Electrical Engineering with emphasis on Radio communications

KTH Supervisor: Dr. Fredrik Jonsson
BTH Supervisor: Dr. Wlodek J. Kulesza

December 2014
School of Engineering
Blekinge Institute of technology
Karlskrona, Sweden
ABSTRACT

In this thesis, we present a graphical computer-aided design (CAD) environment for the design, analysis and layout of printed electronic batteries in the first phase and the parasitic extraction of the connecting wires in the second phase. The primary motivation of our work is that this prototyping software tool so far does not exist. Our tool has been integrated within the existing CAD tool which allows quick prototyping and simplifies the interface between the system designer and the device manufacturer.

This tool supports the schematic and layout entry, rule checking and netlist generation. The first phase of the device synthesis modelling is based on Enfucell printed batteries, by which using the CAD tool, the shape of the battery is optimized and designed to fit the product and is able to simulate the performance during the optimization, whereas the second phase is the parasitic extraction using an extracting tool named fasthenry, which is integrated to our CAD tool to extract unwanted resistance and inductance within the shared wires between the battery and other devices. We believe that the availability of this tool is useful to the CAD community for novel ideas in the circuit design for flexible hybrid electronics.
This master thesis would not have been possible without the support and guidance of the following people.

First and foremost, we would like to express our sincere gratitude to our supervisors Dr. Fredrik Jonsson (KTH) for the constant guidance, motivation and for providing this wonderful opportunity to complete this master thesis. We would also like to convey our sincere thanks to Dr. Wlodek J. Kulesza (BTH) for being our advisor and supporting us.

Finally, we would also like to thank our families and friends for their enormous support and encouragement for us in completing this project.
# Table of Contents

Abstract .................................................................................................................. 3  
Acknowledgement .................................................................................................. 5  
List of abbreviations ............................................................................................... 8  
Chapter 1 ................................................................................................................. 9  
1 Introduction ......................................................................................................... 9  
1.1 Problem statement and challenges ............................................................... 9  
1.2 Thesis objective ............................................................................................. 10  
1.3 Main contributions ......................................................................................... 10  
1.4 Thesis organization ....................................................................................... 11  
Chapter 2 ................................................................................................................. 13  
2.1 Printed electronics ......................................................................................... 13  
2.2 ECAD tools and design flow for printed electronics .................................... 13  
   2.2.1 Drawing tool ......................................................................................... 14  
   2.2.2 Rule checker ......................................................................................... 14  
   2.2.3 Layout to printer .................................................................................. 15  
DEVICE SYNTHESIS ..................................................................................... 16  
Chapter 3 ................................................................................................................. 17  
3.1 Introduction .................................................................................................... 17  
3.2 Background and related work ...................................................................... 17  
   3.2.1 Elements of battery ............................................................................. 17  
   3.2.2 Printed battery and its detailed design ................................................. 18  
   3.2.3 Methods and models .......................................................................... 19  
Chapter 4 ................................................................................................................. 21  
4.1 Electrical battery modelling ........................................................................ 21  
4.2 Technical background and modelling ......................................................... 22  
   4.2.1 Schematic design and behavioural modelling .................................... 22  
Chapter 5 ................................................................................................................. 26  
5.1 Archipelago ................................................................................................... 26  
5.2 Design interpretation .................................................................................... 26  
   5.2.1 Interpretation with an Ecad tool .......................................................... 28  
5.3 Device models .............................................................................................. 28  
   5.3.1 Standard model .................................................................................. 28  
   5.3.2 Custom model ................................................................................... 29  
5.4 Simulation ...................................................................................................... 30  
EXTRACTION OF ELECTRICAL PROPERTIES ........................................ 33  
Chapter 6 ................................................................................................................. 34  
6.1 Interconnects ................................................................................................ 34  
6.2 Parasitic Extraction ....................................................................................... 35  
   6.2.1 Parasitic effects ................................................................................... 35
List of Abbreviations:

ECAD: Electronic Computer Aided Design
EDA: Electronic Design Automation
PE: Printed Electronics
OE: Organic Electronics
SOC: State of charge
C Rate: Cell Rate
mAH: Milliamp Hour
Ah: Ampere hour
VCVS: Voltage Controlled Voltage Source
GUI: Graphical User Interface
Chapter 1

1. INTRODUCTION

Printed electronics is an emerging technology in electronics manufacturing. The major difference between printed and traditional electronics is the way how the components are handled. With printed electronics the manufacturing of the components is in one complete flow whereas in traditional electronics the components are manufactured by different vendors and assembled on a circuit board. The process of combining printed electronics with the traditional electronics and to exploit the advantages of both the technologies are known as hybrid integration.

1.1 Problem Statement & Challenges

Many of the current researches regarding printed electronics are about the fabrication of electronic devices and the improvement of the printing techniques for higher resolution. The required attention is not given to the definition of design flows or tools which aid the designer. The developer must be able to design a tuneable circuit with better performance without much knowledge of the underlying manufacturing process. There should be a clear abstraction between the designer and the technologist where the former could design without complications and the later makes the required technology changes.

In this thesis we demonstrate the work on flexible printed batteries and the parasitic component extraction from the wiring connections, considering all the challenges as mentioned above we put up a two research questions and formed suitable hypothesis.

- How can a battery model be optimized in terms of shapes and performance using the CAD tool?

The most suitable way to optimize the performance of the battery model for different shapes, are by creating a circuit schematic electrical model and a lookup table based on the physically available measurements and integrate it with the CAD tool.

- How do we extract the unwanted parasitic components from the modelled circuit for creating a better blueprint?
The parasitic element could be extracted by integrating the Fasthenry tool for parasitic extraction and by considering the possible scenarios for the occurrence of the parasitic elements in the wires and implementing them in the CAD tool.

The thesis challenges also rely on developing an increasingly complex circuit design involving various electronic devices and high performance components which are made to fit into smaller or larger areas.

Our thesis is aimed to overcome the challenges such as the unwanted parasitics and for the optimization of the circuit in terms of performance for different battery shapes, with the help of the software tools for hybrid integration.

1.2 Thesis Objective

The objective of this project is to develop a software tool that enables the designer to design hybrid circuits without having specific knowledge of underlying manufacturing steps and also where the circuit designer is able to tune the circuit according to their own circuit geometries and electrical properties to optimize the system performance. The properties of the printed electronic devices are likely to affect the performance of the circuit, so electrical properties of the circuit are extracted from the layout and incorporated to the circuit simulation flow. The tools developed for this project are integrated into the design flow of existing Archipelago ECAD tool.

The work will also involve electronic printed battery modelling and tuning based on the information from Enfucell printed batteries [14], which will be performed at the ES department at KTH in collaboration with the Company Archipelago Design Automation AB.

1.3 Main Contributions

The thesis contributions deal with the device synthesis and extraction of electrical properties.

Shijith performed the literature study on the available dynamic nonlinear models of battery. He developed a new model for the printed flexible battery. The tool for modelling includes the design, layout and analysis of the printed battery. The tool was designed in such a way that can be used for both standard and custom model battery types which supports the schematic and layout entry, netlist generation and rule checking for the battery. The discharge
characteristics of the battery are analysed and the spice simulation results of the developed battery model are compared with the actual Enfucell discharge curves. The developed tool is then incorporated with the existing CAD tool.

Radhakrishnan performed the literature study about the extraction of the parasitic elements from the printed electronic circuits. He worked on the parasitic extraction from the layout modifications using Fasthenry. Fasthenry extracts an RL equivalent circuit from the layout, and that can be imported into spice. The electrical characteristics which are extracted from the layout are then simulated to show the performance. He then incorporated the simulation results from Fasthenry into the Testbench to get the optimized results. The tool for extracting parasitics was developed and incorporated into the existing CAD tool.

1.4. Thesis Organization

This thesis report is organised as the following

Chapter 1 explains the general introduction, scope of the project and the motivation.

Chapter 2 introduces the printed electronics, its entire design flow and the design flow of an ECAD tool

Chapter 3 demonstrates the device synthesis part of the work where the detailed design of the printed battery is briefly explained and also about the methods and the models incorporated to determine the battery discharge.

Chapter 4 explains why the electrical modelling is considered from the other models and also gives the detailed schematic model design technical explanation and the lookup table usage based on the printed Enfucell battery.

Chapter 5 states the design interpretation using the CAD tool Archipelago and also explains about other tools integration within the Archipelago. This chapter also explains about the custom and standard type battery prototype models and their simulated graphical outputs.

Chapter 6 deals with the extraction of electrical properties and also explains the need for the Fasthenry for parasitic extraction. Effects of parasitics and the working of Fasthenry in extracting the parasitics from the layout are also explained.
Chapter 7 shows a tool methodology of parasitic extraction using the CAD tool Archipelago and also explain the integration of Fasthenry with the CAD tool. We also give a brief explanation about the method of extraction using the wire model. The modelling and validation of simulation result using the specific testbench are explained.

Chapter 8 gives the conclusions and future work.
Chapter 2

2.1. Printed electronics

This technology has been appeared under several names such as printed electronics (PE) or organic electronics (OE) [3], the growing field of printed electronics combines functional materials for fabrication and electronic systems to create electronic circuits. This process adapts to existing printing techniques and provides functionality similar to their traditional silicon-based counterparts. There are several printing methods used in printing electronic circuits such as screen printing, roll to roll, ink jet etc. The basic flow for printed electronics includes processes such as:

- Designing the circuit
  - Modelling
  - Simulation
  - Debugging
- Materials and Machinery for printing
  - Inks
  - Electrical properties
- Substrates for printing.

2.2. ECAD tools and design flow for printed electronics

Electronic Computer Aided Design (ECAD) environments provide the tools for generating a physical representation of the circuit from a high-level description. The tools aid the designer for the design flows. The process of designing a CAD tool is shown in Figure 2.1.

The basic tool operations include:

1) Drawing tool
2) Design rule checker
3) Layout to Printer tool
2.2.1. Drawing tool

Here we discuss about the CAD drawing tool, the first step in the physical design is for the designer to create circuit schematic using a schematic editor. Then it is necessary to generate a netlist which describes the interconnections between the circuit components in order to simulate and validate the schematic. After the schematic entry is done the user creates the layout of the circuit. A layout design refers to the character of the elements which corresponds to the patterns of metal, oxide or semiconductor layers that make up the components and the interconnections of a circuit.

2.2.2. Rule checker

The layout design rules are the link between the circuit design and the technology constraints. The rules performs a check to verify whether the schematic and layout are equivalent and also to ensure that the circuit topology, sizes match exactly [1].
A wire usually has some resistance, capacitance and inductance, therefore wiring forms a complex geometry that introduces unwanted resistance, inductance and capacitance parasitic effects. This phase of the CAD tool is also equipped to deal with the unwanted parasitic effects that a circuit could produce during the operation.

### 2.2.3. Layout to printer

The final phase involves the bitmap file generation which is then used by the printer for printing the circuit on a substrate, which is also interpreted by the technology which mentions the type of dye used and the distance between the drops while printing.
DEVICE SYNTHESIS
Chapter 3

Printed batteries

3.1. Introduction

The device synthesis focuses on printed battery. Printed batteries are required for vivid potential applications such as RFID tags, batteries for power supply, multicolour displays, wireless sensor networks and portable consumer electronics. This chapter covers the battery types, composition and the preferred battery model for printed battery technology.

3.2. Background and related work

Battery is a device that converts chemical energy into electrical energy and vice versa, not all batteries are created equal even if they are of the same chemistry. The batteries can either be of high power or high energy but not both. The manufacturing of the batteries are based on these categories.

A primary battery is one that cannot be recharged whereas secondary battery can be recharged. A cell is the smallest packaged form of a battery, a module consists of several cells connected in series or in parallel and a pack of battery is by assembling the modules again in series or in parallel [6].

3.2.1. Elements of battery

The cell system can be described using three main components:

- Anode: releases (oxidation) electrons that flow to the cathode through external circuit.
- Cathode: accepts (reduction) the incoming electrons.
- Electrolyte: completes the circuit by transporting the ions

The anodic and cathode electrodes are separated by a separator which is the electrolyte that prevents the internal short circuit. Commonly available batteries are with the chemical composition of lead acid, Nickel batteries (NIMH & NICD), Lithium- ion and manganese dioxide (Zn-MnO2) batteries.
3.2.2. Printed battery and its detailed design

The primary non-rechargeable printed batteries made of zinc and manganese di oxide (Zn/MnO2) chemistry has already made it to the market and the works on secondary batteries are in process. Our thesis based on Enfucell battery is of Zn/MnO2 composition. One of the important aspect in making the printed battery is by using the chemical components which does not pose any risks to the user, especially in wearable devices. In printed batteries, in order to increase the potential to a sufficient level, the cells are arranged in co-planar or side-by-side manner than by stacking it, which could prevent the increase in thickness of the battery.

Figure 3.1: Stack layout of a printed battery [4]

Before producing the physical battery there are certain models and methods to follow which determine the working of the battery. Electro-chemical, electrical and analytical models are the types of models which could help in calculating the discharge rate, life time of the battery and other important necessary parameters.

Figure 3.2: Series connection of 4 single battery cells [4]

The Figures 3.1 and 3.2 are an example which represents number of four cells connected in a side by side manner by which the battery potential increases whereas the capacity remains unaffected. The parallel connection retains the battery potential whereas
there is an increase in cell capacity and a decrease in total resistance. The capacity of the battery also depends upon the active materials used in the electrode.

### 3.2.3. Methods and models

There are several models developed to predict the discharge rate and life time of the battery under the given load. These different models will be discussed in the following:

- **Electro chemical model** is based on the chemical process that takes place within the battery, to develop this model one needs to know full detailed knowledge about the battery which includes the salt concentration of the electrolyte to the nonlinear equations required to calculate the varying relationship between lifetime of the battery and the rate of discharge. This makes a very accurate model, however it is considered to be one of the most complex model of the other available models.

- **Analytical model** is the easiest model than any of the other available models. This is considered as the simplest model for predicting the non-linearity of the battery, the relationship between discharge rate and the battery life time could be calculated using one of the simple equation under this model that is by using Peukert’s law:

  \[
  L = \frac{\alpha}{I^b}
  \]  

  where \( L \) is the battery life time and \( I \) would be the discharge current, \( \alpha \) and \( b \) are constants obtained from the experiment, however \( \alpha \) would be the value close to the battery capacity and \( b \) is usually the value between 1.2 and 1.7 based on the battery chemical composition. The result obtained from this analytical model for predicting the life time of the battery is reasonably good but it does not deal well with variable load.

- **Electrical circuit model** would be considered the simplest model compared with the electro chemical model and so therefore computationally less expensive, it still takes some effort to configure the electrical circuit model, the challenge for the configuration mainly lies in the look up table. The look up table requires a lot of experimental data regarding the behavior of the battery.
An electrical circuit battery model usually consists of the following core:

- A capacitor which represents the capacity of the battery.
- A resistor to represent the battery resistance.
- A voltage vs. state of charge lookup table.
- A discharge rate normalize to determine the lost capacity during high discharge currents.
- A circuit to discharge the capacity of the battery.

LTSpice and NgSpice simulation program can be used to simulate the circuits, the changes in the battery model for alkaline, nickel cadmium and various other models can be made by altering the parameters in the core.
Chapter 4

Battery modelling

4.1. Electrical battery modelling

Electrochemical and mathematical models are not suitable for real-time applications therefore the equivalent circuit electrical models are the choice in our project to determine the important characteristics of the battery. The model of interest here would be a dynamic battery model which would predict the run time of the battery with the variation of the parameters within, such as the variation in the temperature or the SOC.

The electrical battery modelling in our work is for the Enfucell printed flexible batteries [14]. This electrical model is based on the data from the available Enfucell battery models and could be used to electrically model and optimize the performance for various batteries which requires better discharge curve for different battery shapes. Our model is designed to calculate the potential difference, capacity and the internal resistance by changing the shape of the battery which also considers the other dynamic characteristics such as SOC and cell temperature.

The specifications for a regular 1.5 V Enfucell battery are as follows.

Figure 4.1: Enfucell 1.5 V battery
- Maximum thickness – 0.7 mm
- Outer dimension – 60 mm x 72 mm
- Nominal voltage – 1.5 V
- Initial capacity – 90 mAh
- Initial internal resistance – 25 Ω
- Chemistry Zinc – Manganese dioxide - Zinc chloride
- Operation temperature – 35°C to 50°C
- Other available voltage range – 1.5 V mini, 3 V, 4.5 V & 6 V

4.2. Technical background and modelling

The spice simulator used in the project for electronic circuit modelling and for waveform viewing is LTSpice. The simulator helps to observe the variations of the experimental results with the simulated results and also helps with the required changes to be made on the circuit design or to the look up table if there is a mismatch in results.

4.2.1. Schematic design and behavioural modelling

The circuit design here is based on the physical available data from the Enfucell battery.

Figure 4.2: Schematic model of the printed battery
For modeling a cell, several other discharge curve is measured on a computerized level from a physically available model and make it available for other new models based on the requirements. The schematic model of the printed battery produced using LT Spice is shown in Figure 4.2. The Figure 4.2 could be divided into sections and the behaviour will be explained as follows:

- Capacitor representing capacity Ah (ampere-hour): C_Cellcapacity – The SOC of the cell is modelled using a simple capacitor. The capacitor triggers the charge of the cell, which acts as a charge storing element. The capacitor is at full charge which is 100%, indicating 1 Volt and 0.5 Volt at 50% of the cell capacity. In this model the capacitor is introduced between node 50 and the ground (C_Cellcapacity 50 0 {3600*CAPACITY*Fudge Factor}). The capacity here is the ampere-hour capacity of the cell, if the cell is 90 mAh, the capacity equals .09 Ah x 3600 or 324 ampere hours.

  Fudge Factor adjusts for the difference in the manufacturer’s listed Amp-hour capacity (i.e., the cutoff voltage with some capacity remaining at the cutoff) [7]. A Fudge Factor value ranges between 1.01 to 1.15 [7].

- A circuit to discharge the ampere-hour capacity of the cell: The G_discharge discharges the capacity at the cell rate.

- Discharge rate normalizer for determining the lost capacity: The actual capacity of the cell depends on the rate at which the cell is being discharged [7]. Usually the discharge rate for the cell is listed for greater than 20 hours for an efficient battery. A VCVS (Voltage Controlled Voltage Source) is introduced in series with the C_Cellcapacity, for any inefficiency within the model. The inefficiency could be introduced due to a discharge at a rapid rate which may also introduce non-linearity, this non-linearity could be overcome by characterizing a look up table at many discharge rates.

  The capacity from the cell during the discharge is subtracted by the VCVS and the subtracted amount depends on the rate at which the cell is being discharged. The next step is the normalization where the discharge rate in ampere is converted to the C rate (cell rate). The C rate is the Ah capacity of the cell which is discharged completely in an hour. The normalization makes it easy to determine the inefficiencies within the cell at different discharge rates as the ampere discharge is converted into C units of the battery capacity for an hour. The conversion is done in the model by the VCVS, E_Rate, as follows, value = {I(V_Sense) / CAPACITY}. E_Rate senses the discharged current in ampere and
divides it by the Ah capacity of the cell, for example for a regular 1.5 V Enfucell battery the amp-hour capacity is 90 mAh (CAPACITY).

This rate information is then fed directly to the VCVS, E_Lost_Rate for determining the available actual capacity. An RC lowpass filter is modelled in the normalizer part of the circuit for any delayed rate. E_Lost_Rate is between the node 50 and the SOC node, the table entries in the E_Lost_Rate indicates the unavailable capacity from the cell at the discharge rates, table (0.0, 0.0) (1.5, 0.5) states that at the discharge rate of 0, the cell loses 0% of the capacity and if it is 1.5 ie., (1.5 C rated capacity) it loses 50% of the cell capacity. The SOC node is where the subtraction of C_Cellcapacity voltage and the E_Lost_Rate occurs, which in turn represents the capacity within the cell for a given discharge rate.

- Cell voltage versus state-of-charge lookup table: The lookup table is modelled from the physically measured available data of the Enfucell, where a single curve is made by averaging several curves picked from the available data. The data is then converted to lookup table. In this model a VCVS models the cells output voltage versus the state of charge at different discharge rates. E_Cellx is a VCVS fed with the lookup table for the cell, the table pairs are the output to +OUT and -OUT.

The discharge current sense and the cell resistance are modelled by introducing a zero valued voltage source in series with the output voltage –OUT.

<table>
<thead>
<tr>
<th>State of charge</th>
<th>Voltage[V]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>1.70</td>
</tr>
<tr>
<td>0.03</td>
<td>1.60</td>
</tr>
<tr>
<td>0.07</td>
<td>1.50</td>
</tr>
<tr>
<td>0.10</td>
<td>1.45</td>
</tr>
<tr>
<td>0.20</td>
<td>1.40</td>
</tr>
<tr>
<td>0.36</td>
<td>1.37</td>
</tr>
<tr>
<td>0.55</td>
<td>1.34</td>
</tr>
<tr>
<td>0.60</td>
<td>1.31</td>
</tr>
<tr>
<td>0.65</td>
<td>1.29</td>
</tr>
<tr>
<td>0.70</td>
<td>1.25</td>
</tr>
<tr>
<td>0.74</td>
<td>1.20</td>
</tr>
<tr>
<td>0.75</td>
<td>1.15</td>
</tr>
<tr>
<td>0.78</td>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 4.1: Battery lookup table
The voltage [V] vs. state of charge shown in Table 4.1 used in our proposed model is based on 1.5 V regular size Enfucell battery.

- Cell resistance: The internal resistance of the battery depends upon various factors such as the cell size, chemical properties, age, temperature and discharge current. This prototype model of Enfucell is specified with an internal resistance of 50 Ω.
Chapter 5

Design and simulation using Archipelago

5.1. Archipelago

Archipelago is a collection of tools for Electronic Design Automation (EDA). Archipelago is a prototype software with which the components could be fine-tuned for different sizes, substrates etc. The layout automatically updates to allow optimization of device, the device library contains the required devices, which could be batteries, sensors and other devices which are under development. After confirming the layout the blueprint could then be sent to the factory for manufacturing.

The performance of device can be simulated by modifying the parameter of the device. Using the extraction tools the layout dependent parameters are extracted. As this project is focused upon flexible battery, the device optimization, tuning and the simulation will be explained in brief in this chapter.

5.2. Design Interpretation

The GUI (Graphical User Interface) acts as the front end interface. The library that operates are programmed using C++ and the GUI using Qt framework.

Here the design entry point is made by creating a test bench using the symbol of the cell as in Figure 5.2, in which the symbol captures the whole schematic as in Figure 4.1 of the entire cell model that could be optimized as per the requirement of the user. The netlist of the cell schematic is integrated with the symbol of the battery from the test bench in the layout editor. When the user changes the shape of the printed cell the netlist parameters such as the capacity, voltage, number of cell and the internal resistance are automatically changed and so does the discharge curve output. The netlist and the coding are presented in Appendix A and B. After the optimization and verification of the cell performance, in order to complete the blueprint for printing the flexible circuit, the wiring and their unwanted parasitic extraction under various scenarios are performed which would be briefly explained from the next chapter.
Figure 5.1: Design process

The flowchart in Figure 5.1 represents the entire design process of a printed circuit.

Figure 5.2: Test bench for the cell
The Figure 5.2 shows the test bench of the cell for different width and height where the cell is connected to an external load resistor of 10 kΩ, tstep and tstop (time in seconds) denotes the initial time step and end of simulation respectively. In our model based on Enfucell the discharge rate is shown for 600 hrs.

5.2.1. Integration with an ECAD tool

Here the tool used for simulation is NgSpice and the tool for extracting unwanted resistance and inductance or called parasitic extraction uses Fasthenry, these two are separate programs that are integrated with the tool which makes it quite feasible and ease the design process within the ECAD tool.

5.3. Device models

Based on the physical measured data of an Enfucell battery and for the optimization for new shapes and better performance the design phase provides two models in this project, the standard and the physical models.

5.3.1. Standard model

The standard models are the available Enfucell battery models. The Figure 5.3 represents the model for an available regular size 1.5 V battery with the width of about 36 mm and height 46 mm. In our project we have included all the available company models for 1.5 V, 3 V, 4.5 V and 6 V. The layout pattern changes based on the width and height of the model.
5.3.2. Custom model

The custom model is where the users could design their own battery shape and verify the performance. In this model the active area within the cell is fixed for a size of about 30 mm x 30 mm, i.e., for example a layout width and height which is about 36 mm x 40 mm it could accommodate a single cell with an active area of 30 mm x 30 mm. When the user needs a high potential battery as in Figure 5.4, they manually feed the size of the width and height to double or more, so that it could accommodate multiple active areas within and which also eventually changes the internal resistance and the value of the total battery capacity for each active area addition and later checks the discharge curve performance for producing an optimized battery blueprint. The code for the custom model is presented in Appendix B.
Figure 5.4: Custom battery model

As the user feeds the values for the width and height of the cell, the layout changes accordingly so does the associated internal parameters, from Figure 5.4 the mentioned width and height could accommodate 3 cells and provides a potential of about 4.5 V, considering the fact that a single cell has an active area of about 30 mm x 30 mm size and potential of 1.5 V, the capacity obtained here is 14 mAh and the internal resistance of about 400 Ω.

5.4. Simulation

The Figure 5.5 represents the voltage vs. time discharge rate performance, the graph in Figure 5.5a denotes the already available physically measured result from the Enfucell 1.5 V battery [14], the Figure 5.5b shows the spice simulated discharge curve using LTSpice, which represents 1 kΩ, 2 kΩ and 10 kΩ resistance respectively and the Ah capacity with the help of lookup table and is made to match the physically available measured result from Enfucell, the lookup table is tuned in such a way that the discharge rate is above 20 hours for an efficient battery. The graph in Figure 5.5c is the comparison between the already physically measured and available 1.5 V Enfucell graph with our software modelled battery produced graph.
Figure 5.5a: Discharge curve of an Enfucell 1.5 V battery for different loads [14]

Figure 5.5b: Simulated discharge curve for the prototyped model using LTSpice
The Figure 5.5c shows the discharge characteristics for 1 kΩ, 2 kΩ and 10 kΩ loads, the nominal operating voltage for the flexible battery is at 1.5 V and then the voltage remains relatively constant for higher load resistance whereas gradual decreases takes place in loads of 1 kΩ and 2 kΩ resistance, until it drops off and reaches the cut off voltage point at 0.9 V and then it is completely depleted.
EXTRACTION
OF ELECTRICAL PROPERTIES
Chapter 6

EXTRACTION OF ELECTRICAL PROPERTIES

This chapter explains the second hypothesis of our thesis. In printed electronics, throughout integrated circuits, on chip interconnect wires were needed to be considered while performing high-precision analysis. The properties of printed circuits will affect the performance of the circuit. Therefore the electrical properties must be extracted to know how far the parasitic resistance and inductance affects the device performance. The tool which we developed is able to extract resistance and inductance from the layouts and incorporated into the circuit simulation flow. The extraction of electrical properties is discussed in this chapter.

6.1. Interconnects

An interconnect is a conductive path of thin-film wire that electrically connects two or more components in an integrated circuit. Interconnects can introduce parasitic components such as unwanted capacitance, resistance and inductance. Electrical interconnects such as wire and packages will generate parasitic components. Figure 6.1 represents the bond wires and packages create parasitic inductance and resistance that will affect the circuit performance.

Figure 6.1: Bond wires and packages
6.2. Parasitic Extraction

In high performance circuit designs, on-chip inductance has become more significant due to faster rise times, lower resistance and lower capacitance. These technology advancements make the inductance effect increasingly important than ever. Therefore it is desirable to extract interconnect inductance and impedance accurately for high-performance circuits [18]. As devices are scaled down in size and the number of metal wiring layer increases, the impact of interconnect parasitics increases. The parasitics effects introduced by the wires display a scaling behaviour that differs from active devices such as transistors, and tends to gain in importance as device dimensions are reduced and circuit timing is increased. An interconnect design is starting to be considered at even earlier stages of the design flow such as gate-level netlist generation. Since the interconnect design affects every stage of the design flow, fast and accurate full-chip level parasitic extraction are becoming increasingly important [21].

6.2.1. Parasitic Effects

Parasitic effects are due to interconnect and are not an intended part of the circuit function. These parasitics detrimentally affect the performance propagation delay, power consumption and reliability. Parasitic also has impact on delay, noise and energy consumption. Parasitic effects typically affect the timing behaviour of the circuit and causes voltage drop. The circuit should perform correctly if the frequency is slow enough. Therefore the extraction is needed to support simulation of the effect due to parasitic elements.

6.3. Inductance modelling

The wires that interconnect the devices were treated as short circuits and when the wire cross section dimensions become smaller due to technology scaling, the inductance becomes significant. The wire is modelled as reduced RLC transmission line. Because of the higher operational frequencies and lower resistivity copper interconnects, inductive impedance becomes more important. Therefore resistance ($R$) and inductance ($L$) can no longer be neglected in interconnect design. With the faster circuit speed and relatively lower resistance from the new technologies, the wire inductance becomes important. Figure 6.3a shows the resistance ($R$) and inductive part of the line impedance ($\omega L$) as a function of frequency for a given line structure. $\omega L$ increases drastically as frequency increases [21].
From the Figure 6.3a it depicts that as the frequency increases, the inductive part of the line impedance becomes comparable to the resistive part.

In this section, we review the governing equations for the magnetic inductance and the reduced order model to evaluate the effects of interconnects.

A wire carrying an electric current or a time varying electric field creates magnetic fields. This relationship is represented by Ampere’s law [23] as the following:

\[ \nabla \times \mathbf{B} = \mu \mathbf{J} + \mu \varepsilon \frac{\partial \mathbf{E}}{\partial t} \]  \hspace{1cm} (6.1)

where \( \mu \) is the magnetic permeability and \( \varepsilon \) is the electric permittivity of the material. The first term on the right-hand side represents a magnetic field generated from a wire carrying electric current. The second term corresponds to the magnetic field generated from the displacement current, which represents the ac current flowing between two conductors due to their capacitive coupling. In integrated circuits, the second term is usually neglected because the current flowing in the conductor is much larger than the displacement current [22]. This assumption is called the magneto-quasistatic approximation. [21].

As the inductance becomes significant, the wires are reduced to RLC line model as shown in Figure. 6.3b. To evaluate the effect of interconnects on design performance the wire
has to be modelled [19]. The wire model has segments with uniform current of RLC lines and
the segment is modelled to a reduced order model.

![Image of RLC network](image)

**Figure 6.3b: From electromagnetic analysis to reduce order circuit simulation**

### 6.4. Overview of Fasthenry

Fasthenry is one of the inductance extraction methods. It is a three dimensional LR
extraction program. Fasthenry computes the frequency dependant self and mutual
inductances and resistances between conductors of complex shape.

### 6.4.1. Need for Fasthenry

The most accurate way of extracting inductance is to divide the structure into smaller
regions and numerically solve the magnetic field within each region to find the magnetic flux.
The calculation of inductance remains important for both digital and analogue
superconductive electronic circuit design. Many numerical inductance calculation tools have
been developed for integrated circuit elements [23]. The HFSS (High Frequency Structure
Simulator) from Ansoft is a full wave 3-D electromagnetic solver based on finite element
method optimized for high frequency applications. Although this is an accurate method to extract inductance, solvers require a large memory and very long simulation time [21].

Most of the techniques or tools are limited to basic geometrics and are associated to analytical model and curve fitting and are not integrated to software tools [29]. Whereas Fasthenry extracts $L$ and $R$ together and also considering frequency. Fasthenry has Magneto-quasistatic assumption with terminal pairs of known direction. The wire is partitioned into filaments and current is distributed evenly. Fasthenry has the wide applications on chip wires and packages. Fasthenry can be integrated into software tools.

6.4.2. FASTHENRY

Fasthenry is a fast 3-D inductance and resistance extraction tool developed at M.I.T on Unix platform for the solution of Maxwell equations and extraction of inductance and resistances. It is developed originally for analysing general packaging structures [19]. For analysis, a structure is subdivided into discrete connected segments which can be divided further into filaments (Figure 6.4a). Fasthenry then uses analytic solutions for the filaments and performs a mesh analysis on the structure to calculate the complex impedance between defined ports. The typical inductance extraction problem requires more than a few thousand filaments, which renders Gaussian elimination for the solution of the complex linear system impractical. Therefore an iterative generalized minimal residual (GMRES) algorithm is implemented in Fasthenry [23].

![Figure 6.4a: A single element used by Fasthenry with electrical nodes and filaments](image)

6.4.3. Working of Fasthenry

Fasthenry is used for computing the frequency dependant self and mutual inductances and resistances of a generic tridimensional conductive structure in the magneto quasistatic approximation [18].
The input data (technology file) describing the geometry and the frequencies of interest, must be provided in a file (.inp file). This file specifies every conductor as a sequence of rectilinear segments connected between nodes. Every segment has a finite conductivity and the shape of a parallelepiped, whose height and width can be assigned. A node is a point in the 3D space. The section of a segment can be divided, if required, into an arbitrary number of parallel filaments (that is, parallelepipeds with smaller cross section than the original one), the whole of which constitutes the segment itself, it is then assumed that every filament carries a uniform current (Figure 6.4b) [25].

![Segment discretized into filaments](image)

**Figure 6.4b: Segment discretized into filaments**

In this way is possible to model the high-frequency effects on the segments. In fact, when the frequency increases, the current is no longer uniformly distributed along the cross section of a conductor [23]. However in limited regions of the section, the current can be reasonably approximated as uniform. Therefore being able to specify an arbitrary discretization of the volume of the conductors, the accuracy of the results is affected accordingly and in general is better as the discretization is refined (Figure 6.4c) [19].

![Figures 6.4b and 6.4c](images)

**Figure 6.4c: One conductor is discretized into filaments.**
For computing frequency dependant inductance and resistance matrix associated with the terminal behaviour of collection of conductors involves first approximating each conductor with a set of piecewise conducting sections. The volume of the each straight section is then discretized into a collection of parallel thin filaments through which current is assumed to flow uniformly. The resistance and inductance matrix are deduced by assuming the applied currents and voltages are sinusoidal and that the system is in sinusoidal steady state.

The branch current phasors can be related to branch voltage phasors by

\[ Z I_b = V_b \] (6.2)

where \( V_b, I_b \in C^b \), \( b \) is the number of branches (number of current filaments), and \( Z \in C^{b \times b} \) is the complex impedance matrix given by

\[ Z = R + j\omega L \] (6.3)

where \( \omega \) is the excitation frequency. Considering a geometry consisting of two input-output terminal pairs as shown Figure 6.4d:

\[ Z_r(\omega) = R_r(\omega) + j\omega L_r(\omega) = \begin{bmatrix} R_{11}(\omega) + j\omega L_{11}(\omega) & R_{12}(\omega) + j\omega L_{12}(\omega) \\ R_{21}(\omega) + j\omega L_{21}(\omega) & R_{22}(\omega) + j\omega L_{22}(\omega) \end{bmatrix} \] (6.4)

where \( R_r \) is referred to as the resistance matrix and \( L_r \) the inductance matrix. Also, \( L_{11} \) and \( L_{22} \) are the self-inductances of the conductors and \( L_{12}, L_{21} \) the mutual inductance.

The results are provided in a form of Maxwell impedance matrix \( Z = R + jL \) where the bold letters represent matrices. Then, the results can be converted in equivalent, SPICE-like, lumped elements with a utility provided with Fasthenry program, MakeLcircuit. Thus the spice lumped elements have inductance and resistance values obtained using Fasthenry for
the given frequency and conductivity of the material used. Alternatively, is possible to generate directly with Fasthenry software [25], a spice-like circuit capable to model the frequency-dependant inductances and resistances. This latter opportunity is very powerful because allows to see how signals in the time domain are degraded by the different response of the conductors at the various frequencies.

6.4.4 Advantages:

The advantage of Fasthenry is achieved by the fact that it is possible to find rational approximation of these functions, at least in a fixed range of frequencies, and operating only in the time domain. Following this approach, Fasthenry is capable to generate Reduced Order Models for the system, which is valid according to the selected expansion order upto a defined maximum frequency.
Chapter 7

Tool Methodologies

7.1. Tool flow

Tools that are able to extract the layout and simulate performance are created and integrated with Archipelago CAD tool.

Figure 7.1: Tool flow for parasitic extraction

The flowchart in Figure 7.1 shows the tool flow of parasitic extraction. In the Layout, the designer is able to build layout structure according to their own needs. From which the input file for Fasthenry gets generated with the height and width and for one single frequency and conductivity of the material used. The result of the inductance and resistance is saved as Zc.mat file (see Appendix E). After processing Zc.mat file, the spice netlist of the given layout is produced. Now the generated spice netlist of the layout with inductance and
Resistance was incorporated to the testbench. In testbench the generated netlist is fed as a wired symbol and then the simulation of the testbench with the produced netlist are carried out. These are discussed below with the diagram.

### 7.2. Integrating with archipelago tool

#### 7.2.1. Layout for Fasthenry

The layout of the electrical interconnects designed for printed electronics are made here with proper dimensions in mm. The Figure 7.2 is the screenshot of the tool which we created. Where the User can add their own layout according to their requirements with height and width. This generates the Fasthenry input file for the given layout structure. From the layout the input file is exported to Fasthenry. The input file of Fasthenry contains the description of the conductor geometries.

![Layout interface for the parasites](image)

Resistances and inductances \((RL)\) extraction requires the decomposition of the layout into set of straight wires with constant cross section, usually rectangular. Every time the user changes the size or direction of motion, a node in three dimensional space is generated. Wires are generated as a straight line connecting nodes, consistently with the geometric representation adopted by fasthenry. The cross section of a wire connecting two nodes is given by the user and its size can be modified according to user requirements. The wireframe model extracted can be directly fed into a magnetostatic solver fasthenry.
7.2.2. Generating input file from layout

The physical properties of the layers are given in the technology file. The fasthenry input file gets generated from the layout. The Fasthenry input file has (.inp) extension and this technology file includes the description of the conductor geometries in which the nodes get defined with co-ordinates value in the layout. The Fasthenry input file generated from the above layout is given below:

```plaintext
** Input file created by Archipelago
.Units MM
.Default z=0 sigma=5.8e4
N1 x=-12.48 y=13.44
N2 x=-12.48 y=25.92
N3 x=7.36 y=25.92
N4 x=7.36 y=13.12
N5 x=1.28 y=13.12
E1 N1 N2 w=2.56 h=2.56
E2 N2 N3 w=2.56 h=2.56
E3 N3 N4 w=2.56 h=2.56
E4 N4 N5 w=2.56 h=2.56
.extern N1 N5
.freq fmin=1e4 fmax=1e8 ndec=1
.end
```

**Units MM** defines all coordinates and lengths to be in millimetres. All lines with an N in the first column define nodes, and all lines starting with E define segments.

Here Z=0 denotes that the default z coordinate and the copper conductivity is made as default conductivity. It can be changed according to user requirements. The conductivity is in units 1/(mm*ohms) since the default units are millimetres. The default line is given as .Default Z=0 sigma= 5.8e4

In particular, the line E1 N1 N2 w= 2.56 h= 2.56 defines a segment E1 from node N1 to N2 and the segment E1 has a width and height of 2.56 mm.

.extern N1 N5 defines N1 and N5 as one port of the network.

Fasthenry calculates impedance matrix $Z(\omega)$ at the discrete frequencies described by the line .freq fmin=1e4 fmax=1e8 ndec=1 where $f_{min}$ and $f_{max}$ are the minimum and maximum frequencies of interest, and $n_{dec}$ is the number of desired frequency points per decade. In this case, $Z(\omega)$ will be calculated at $10^4$ Hz, $10^5$ Hz, $10^6$ Hz, $10^7$ Hz and $10^8$ Hz. All input files must end with .end.
7.2.3. Processing the output file:

The information contained in the input file gets processed. The input file gets processed to produce the result in Zc.mat file. The impedance matrices for the frequencies specified in the input file will be placed in the text file Zc.mat. This file also lists the correspondence between columns in the impedance matrix and the ports specified in the input file. For each of the matrices, it divides the imaginary part by the frequency to give the matrix \( R+jL \) and then dumps the results as standard output. (see Appendix D).

7.3. Test bench using wire model

7.3.1. Modelling and validation

The result of the extracted inductance and resistance value for the given layout is fed to the wire model. The wire model contains the extracted parasitic resistance and inductance values. The extracted result is given in (.cir) spice file containing the values of inductance and resistance (see Appendix C). Now the testbench is modelled in order to validate the extracted parasitic result using Fasthenry from the layout. The spice file is made as a .SUBCKT file and then combined along with the below given testbench model and then simulated to generate whole netlist containing fasthenry parasitic results and testbench model. The total netlist is then simulated using Ngspice. The Ngspice simulator is incorporated into the ECAD tool for simulating the netlist.

The testbench is performed with the voltage source and capacitance connected in series with the wire model (Figure 7.3a). The wire model has the resistance and inductance values. Thus the testbench acts as the RLC resonant circuit. Once the testbench gets simulated it generates the spice netlist containing the wire model characteristics with voltage and capacitance. Then the simulated frequency response graph of the RLC circuit is obtained. From the graph, \( Q \) (quality factor) of a circuit can be determined using the calculated center frequency, lower and upper cutoff frequencies. \( Q \) is related to the sharpness of the resonance peak obtained in the simulated frequency response curve.
Figure 7.3a: Testbench model

In our testbench the voltage source has the DC offset of 1 V, amplitude 1 V and frequency is 750 kHz. For the small signal AC analysis AC amplitude is given as 1. The voltage source is connected in series with the wire model and also with the capacitor of 1 pF. Our testbench model is configured with AC log sweep with the starting frequency of 100 kHz and stop frequency of 10 GHz. The number of points per decade is given as 50. These values could be configured according to the designer requirements.

Figure 7.3b: Layout used for extraction along with testbench
The different wire layout (Figure 7.3b) for Fasthenry is also made with the testbench and the simulated results are obtained and it is discussed in the next section.

Figure 7.3c: Different Layout of wire used for extraction along with testbench.

7.4. SIMULATION USING TESTBENCH

The simulation of the testbench is made within our tool using Ngspice simulator. The netlist gets generated from the given layout and the result obtained using Fasthenry is integrated to the testbench model. Then the final simulation of the testbench along with the extracted parasitics is done using Ngspice. The simulation results are shown in Figure 7.4a and Figure 7.4b.

7.4.1. RESULTS

The simulation result obtained using the testbench for the given layout (Figure 7.3 b) is shown in Figure 7.4a. The simulation for the magnitude in dB vs frequency response is made with our tool. From the figure the simulation of the small signal frequency for the given testbench shows the increase in the frequency response at 1 GHz. The center frequency of 0.9 GHz is attained at 37.8 dB magnitude. Then the lower and upper cutoff frequencies are calculated. The simulated graph is analysed and compared using LTspice.
Figure 7.4a: Simulation of the extracted results using testbench.

Figure 7.4b shows the simulation report of extracted wire layout parasitics of the layout using Figure 7.3c. The parasitics $R$ and $L$ obtained from the layout using Fasthenry gives the resistance of 0.13 mΩ and inductance of 20 nH (see Appendix C). These two values are added to our testbench. Then the test bench is simulated along with the parasitic results to produce the simulation small signal frequency response graph.

Figure 7.4b: Simulation of the extracted results using testbench for the layout
The simulation of the extracted wire layout parasitics are validated using LT spice where the testbench model is made and the result is obtained as (Figure 7.4c)

Figure 7.4c: Result of simulated testbench with the extracted resistance and inductance using LT spice.

From the graph, the cutoff frequency is obtained using the magnitude in dB vs. Frequency by performing an AC analysis with 50 points per decade and frequencies ranging from 100 kHz to 10 GHz. The plot of the magnitude of voltage at node 3 vs frequency was made. The inductance value is 20 nH and resistance value is 0.13 mΩ.
The graph in Figure 7.4d denotes the frequency response curve for the extracted parasitics. Extracted RL used in circuit simulation shows impact of inductance and resistance.

From Figure 7.4d magnitude in dB vs frequency ac analysis, the maximum magnitude voltage obtained is 27.78 dB and the center frequency is $1.148 \times 10^9$ Hz. For different layout as per designer requirements the value of inductance and resistances are obtained. Those values can be analysed with the use of the testbench. Also the designers can modify the testbench parameter according to their requirements and the simulation graph can be analysed.
CHAPTER 8
CONCLUSIONS AND FUTURE WORK

8.1. CONCLUSIONS

Device Synthesis:

In this thesis we have implemented the tool for developing which includes the circuit and its layout design and analysis of the printed battery. Initially the best suited electrical model for the designed battery is chosen. The electrical battery model seems to be well suited for the real time applications, even when compared to other battery models such as an analytical and an electro chemical battery models.

The electrical model designed in this work is a dynamic model, which helps in predicting the run time of the battery based on the internal parameter variations such as the internal resistance variation due to the fluctuating temperature or a change in the SOC. The important aspect of this electrical model includes the incorporation of the lookup table, the lookup table data are calculated from the available physical battery, where a single curve is made by averaging several curve from the available data. The data are then converted into lookup table and fed into the VCVS of the electrical circuit model. The netlist from the simulation is then incorporated into the CAD tool.

We have developed the custom and standard model. The custom model is designed with a minimum 20 mm x 20 mm sized active area which has a voltage of 1.5 V, the active areas could be added in series to increase the voltage as per the user requirement and that is done in the CAD tool by just changing the width and height, based on the width and height changes the internal parameters such as the internal resistance and the capacity of the battery changes. The standard model consists of all the available Enfucell battery models..

This CAD tool that we developed for the custom and standard battery type modelling supports the schematic and layout entry, netlist generation and rule checking for the battery. The discharge characteristics of the battery are simulated for both custom and standard battery. The spice simulation results are compared with the actual Enfucell discharge curves. This CAD tool is used to help the developer to create the circuit blue print in a faster way just by feeding in the width and height and then by generating the layout and finally verifying the performance.


**Extraction of Electrical properties:**

For the extraction of electrical properties of the printed circuits, the tools that are able to extract the inductance and resistance in line with the layout modifications are implemented. The electrical characteristics are extracted from the layout using Fasthenry. Fasthenry extracts an RL equivalent circuit from the layout that can be imported into NgSpice to simulate the performance of the circuit. A performance simulation of the layout is analysed using suitable testbench for an AC analysis with 50 points per decade and frequencies ranging from 10 kHz to 10 GHz to verify the performance of the resonance circuit, comprising the extracted inductance and resistance values. The simulation of the extracted electrical properties is then incorporated into the CAD tool. One can conclude from the simulation results that the CAD tool created for the extraction of electrical properties optimizes the design to ensure in design stage of a flexible circuit without the parasitic effects.

### 8.2. FUTURE WORK

This work for fitting the battery into different shapes has certain limitation, which is mostly square and rectangular layout representation, but as a future work the batteries could be designed for regular and irregular polygonal cross sections with many number of sides and shapes as per the users’ requirement and could be included in the ECAD tools.

The temperature effect for different cell types could be added to the model.

In this thesis the modelling of printed battery is made and analysed for the better performance but as a future work the other devices in printed electronics could be modelled and integrated with the proposed design tool to analyse their performance.

The parasitic RL extraction is made, but an extraction of capacitance value also could be added by integrating FASTCAP. The parasitic extraction of connecting wires for the bended and folded types could be modelled in the tool.
REFERENCES:


APPENDIX A:

Netlist for battery modelling:

** Modelcard Printed Battery

*Enfucell model

.subckt printed_batt b a cval=0.1
+ PARAMS: CAPACITY=.09 RESISTANCE=50 CELLS=1

* * DISCHARGE RATE CALCULATION * *
E_Rate RATE 0 VALUE = { I(V_Sense)/CAPACITY }
R2 RATE 60 10 ; R2-C1 -> 10 Second time constant
C1 60 0 1

* * DISCHARGE AND STATE OF CHARGE * *
G_DischARGE SOC 0 VALUE = { I(V_Sense) } ; Discharge Current

* * LOST CAPACITY DURING FAST DISCHARGE DELAYED BY R2-C1 * *
E_Lost_Rate 50 SOC TABLE { V(60) } =
+ (0.05,0.0) (0.089,0.11) (0.16,0.20) (0.62,0.39) (0.8,0.47) (1.6,0.55)

* * AMP-HOUR CAPACITY OF BATTERY * *
C_CellCapacity 50 0 { 3600 * CAPACITY * 1.15 }
R1 50 0 1G

* * CELL RESISTANCE * *
E_Resistance 20 10 VALUE = { I(V_Sense) * 2.0 * V(Cell_Res) }

* * CELL RESISTANCE Vs. REMAINING CHARGE MULTIPLIER FACTOR * *
E_Cell_R Cell_Res 0 TABLE { V(50) } = (0,4) (0.2,2) (1,1)
R3 Cell_Res 0 1G
*E_Battery b 10 Value = { V(Cell_V) * CELLS }

* * CELL OUTPUT CURRENT SENSE * *
V_Sense b 20 0

* * CELL OUTPUT VOLTAGE VS STATE OF CHARGE * *
E_Invert Invert 0 TABLE { V(SOC) } = (0,1) (1,0)
R4 Invert 0 1G
* E_Cell a 10 TABLE { V(Invert) } = (0,2.3) (1,0)
E_Cell a 10 TABLE { V(Invert) } =
+ (0,1.7)
+ (0.03,1.6)
+ (0.07,1.5)
+ (0.1,1.45)
+ (0.2,1.4)
+ (0.36,1.37)
+ (0.55,1.34)
+ (0.6,1.31)
+ (0.65,1.29)
+ (0.7,1.25)
+ (0.745,1.2)
+ (0.755,1.15)
+ (0.788,1.0)
+ (1.0,0.001)
.ends printed_batt
.ic v(x_i1.50)=1 v(x_i1.60)=0
APPENDIX B:

Program code for custom and standard model:

```c
void init()
{
  property W, H, type;
  # These variables are displayed
  W = 60m;
  H = 72m;
  type = 1;
}
void list_properties()
{
  property W, H, type, voltage, capacity, resistance, area_size, no_cells, tW, tH;
  pform.clear();
  pform.text("Flexible battery");
  pform.hbar();
  e = pform.variable(type);
  e.enum();
  e.add_option("Custom", 0);
  e.add_option("Model 1", 1);
  e.add_option("Model 2", 2);
  e.add_option("Model 3", 3);
  e.add_option("Model 4", 4);
  e.add_option("Model 5", 5);
  e = pform.variable(W);
  e = pform.variable(H);
  e = pform.variable(voltage);
  e = pform.variable(capacity);
  e = pform.variable(resistance);
  e = pform.variable(area_size);
  e = pform.variable(no_cells);
  e = pform.variable(tW);
  e = pform.variable(tH);
  pform.show();
}
void on_property_change()
{
  property W, H, type, resistance, capacity, area_size, no_cells, voltage, Wa, Wb, tW, tH;
  if(type == 1){
    W = 60m;
    H = 72m;
    area_size = 2250;
    resistance = 25;
    no_cells = 1;
  }
  if(type == 2){
    W = 36m;
    H = 46m;
    area_size = 450;
    resistance = 75;
    no_cells = 1;
  }
  if(type == 3){
    W = 60m;
  }
}
```
H = 42m;
area_size = 250;
resistance = 150;
no_cells = 2;
}
if(type == 4){
W = 140m;
H = 38m;
area_size = 350;
resistance = 400;
no_cells = 3;
}
if(type == 5){
W = 87m;
H = 65m;
area_size = 300;
resistance = 500;
no_cells = 4;
}
if(type == 0){
Wa = 30m;
Wb = 30m;
while((W+3) > Wa) {
Wa = Wa + 30;
}
if(Wa == 30) {
tW = 30;
} else {
tW = Wa - 30;
}
while((H+3) > Wb) {
Wb = Wb + 30;
}
if(Wb == 30) {
tH = 30;
} else {
tH = Wb - 30;
}
no_cells = (tW / 30) + (tH / 30);
resistance = 75 * no_cells;
}
voltage = 1.5 * no_cells;
if(type == 0) {
if(voltage == 1.5) {
area_size = 450;
}
if(voltage == 3) {
area_size = 250;
}
if(voltage == 4.5) {
area_size = 350;
}
if(voltage == 6) {
area_size = 300;
}
}
capacity = (area_size * 4*1e-5);
APPENDIX C:

Running Fasthenry:

Example RUN: ** Input file created by Archipelago fast_henry_test.inp

.Units MM
.Default z=0 sigma=5.8e4
N1 x=-12.48 y=13.44
N2 x=-12.48 y=25.92
N3 x=7.36 y=25.92
N4 x=7.36 y=13.12
N5 x=1.28 y=13.12
E1 N1 N2 w=2.56 h=2.56
E2 N2 N3 w=2.56 h=2.56
E3 N3 N4 w=2.56 h=2.56
E4 N4 N5 w=2.56 h=2.56
.external N1 N5
.freq fmin=1e4 fmax=1e8 ndec=1
.end

Netlist generated for wire model with the testbench:

Netlist of wire model generated using Fasthenry:

LZ_0 1 int_node0_2 2.02143e-008                          (Inductance 20nH)
RZ_0_0 int_node0_2 int_node0_3 0.000134698      (Resistance 0.13 milli-ohms)

Netlist of testbench along with wire model:

** Netlist created by Archipelago
.include "C:/proj/ext_modules/fasthenry_interface/test_proj/printed_devices/wire_mod el.spi"

C_C1 n2 0 1p
X_I1 n2 n1 wire_model
V_I2 n1 0 1 AC 1 SIN (1 1 750k)
.ac dec 50 100000 1e+010
.end
APPENDIX D:

Processing output:

Running FastHenry 3.0
Solution technique: ITERATIVE
Matrix vector product method: MULTIPOLE
  Order of expansion: 2
Preconditioner: ON
Error tolerance: 0.001
Reading from file: C:\Users\Radha\Desktop\fast_henry_test.inp
Title:
** Input file created by Archipelago

all lengths multiplied by 0.001 to convert to meters
Total number of filaments before multipole refine: 4
Total number of filaments after multipole refine: 22

Multipole Summary
  Expansion order: 2
  Number of partitioning levels: 3
  Total number of filaments: 22
No expansions at level 3 (lowest)
No expansions at level 2
Percentage of multiplies done by multipole: 100%
Scanning graph to find fundamental circuits...
Number of Groundplanes : 0
Number of filaments: 22
Number of segments: 22
Number of nodes: 23
Number of meshes: 1
  ----from tree: 1
  ----from planes: (before holes) 0
Number of conductors: 1 (rows of matrix in Zc.mat)
Number of columns: 1 (columns of matrix in Zc.mat)
Number of real nodes: 23
filling M...
filling R and L...
Total Memory allocated: 49 kilobytes
Frequency = 10000
Forming sparse matrix preconditioner..
conductor 0 from node n1
Calling gmres...
1
Frequency = 100000
Forming sparse matrix preconditioner..
conductor 0 from node n1
Calling gmres...
1
Frequency = 1e+006
Forming sparse matrix preconditioner..
conductor 0 from node n1
Calling gmres...
1
Frequency = 1e+007
Forming sparse matrix preconditioner..
conductor 0 from node n1
Calling gmres...
Frequency = 1e+008
Forming sparse matrix preconditioner..
conductor 0 from node n1
Calling gmres...

Computed matrices (R+jL)
Row 0: n1 to n5
  Freq = 10000
    Row 0: 0.000134698+2.02143e-008j
  Freq = 100000
    Row 0: 0.000134698+2.02143e-008j
  Freq = 1e+006
    Row 0: 0.000134698+2.02143e-008j
  Freq = 1e+007
    Row 0: 0.000134698+2.02143e-008j
  Freq = 1e+008
    Row 0: 0.000134698+2.02143e-008j

All impedance matrices dumped to file Zc.mat

Times:  Read geometry   0.405
        Multipole setup 0.306
        Scanning graph 0.0330002
        Form A M and Z 0.0699999
        form M'ZM 0
        Form precond 0.162
        GMRES time 0.676
        Total: 1.652
**APPENDIX E:**

**Zc.mat file:**

```
prompt % ReadOutput Zc.mat

Not part of any matrix: Row 1: n1 to n5

Reading Frequency 10000
Reading Frequency 100000
Reading Frequency 1e+006
Reading Frequency 1e+007
Reading Frequency 1e+008

freq = 1e+008
Row 0: 0.000134698+2.02143e-008j
freq = 1e+007
Row 0: 0.000134698+2.02143e-008j
freq = 1e+006
Row 0: 0.000134698+2.02143e-008j
freq = 100000
Row 0: 0.000134698+2.02143e-008j
freq = 10000
Row 0: 0.000134698+2.02143e-008j
```