FELLOWSHIP REPORT

ON

Improve Circuit Simulations and incorporate Subcircuits in Esim-On-Cloud

Submitted by

Debanjan Bhattacharjee
B.Tech in Computer Science
NIT, Durgapur

Under the guidance of

Prof. Kannan Moudgalya
Chemical Engineering Department
IIT Bombay

May-December 2023

Acknowledgements

I, Debanjan Bhattacharjee, a semester long intern at **FOSSEE - ESIM on CLOUD** project, am thankful to my team and our mentors who believed in me and helped me contribute something of value in this journey.

I will always be thankful to **Mr. Nagesh Karmali, Ms. Firuza Aibara and Mr. Sumanto Kar** for being such warm and wonderful hosts, and their invaluable guidance. Throughout the fellowship they have been the pillar of support. I have learnt a lot through them and will forever remain indebted to them for such a wonderful experience at IIT Bombay.

Contents

1 INTRODUCTION

1.1 Esim on Cloud — Overview

Esim on Cloud is a web-based simulator for electronic circuits. It is a free and open-source plat- form for electronic simulations. This system allows users to design and simulate analog and digital circuits. The users can use a wide variety of components provided by the system.

It behaves like a drag and drop editor and provides basic functionality like undo/redo, rotate components, resizing to name a few. With respect to circuit simulation, it provides basic electrical check to warn about errors before simulation, 4 circuit simulation modes namely – DC Solver, DC Sweep, Transient Analysis and AC Analysis. The source code for the project is available here.

1.2 Technologies Utilised

1. Django
2. PostgreSQl
3. Celery
4. Redis
5. ReactJS
6. mxGraph
7. ngSpice
8. Docker and Docker Compose
9. Nginx

1.3 Objective

I along with my colleagues have worked to extend the functionality of the system as well as im- prove current features during the period of our fellowship. My work consisted of a combination of research, system design and full stack development. I have tried to primarily explore the avenues through which to incorporate modularity of **Subcircuits**, to make the system more useful in the practical e-CAD industry.

The following is a brief list of the features that I have worked on:

1. Fixed a bug and optimized the DFS for the electrical check, thereby solving grounding simulation issues
2. Helped to set up the dev environment on Windows Subsystem for Linux
3. Analysis and Review of the system architecture
4. Designed optimized ways to incorporate Subcircuits

2 FEATURES ADDRESSED

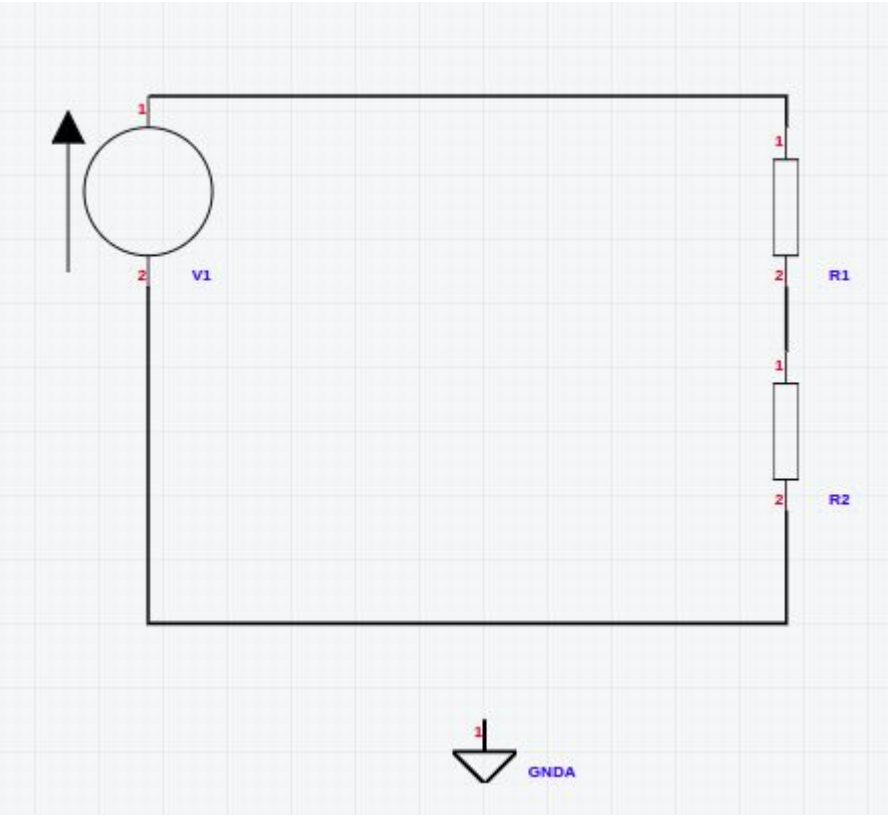
2.1 Fixing Grounding-Connection Bug

2.1.1 Problem Statement

In the frontend, while generating netlist and performing ERC check, the issue observed involved the deletion of a wire connecting the GND to another wire, which persisted after saving the circuit. This problem led to unexpected errors in the simulation process, negatively impacting user experience. Specifically:

- The wire from the GND to another connected wire gets deleted after ERC check/saving the circuit, resulting in circuit malfunction.

The goal was to investigate and fix this behaviour to ensure circuit integrity upon saving.



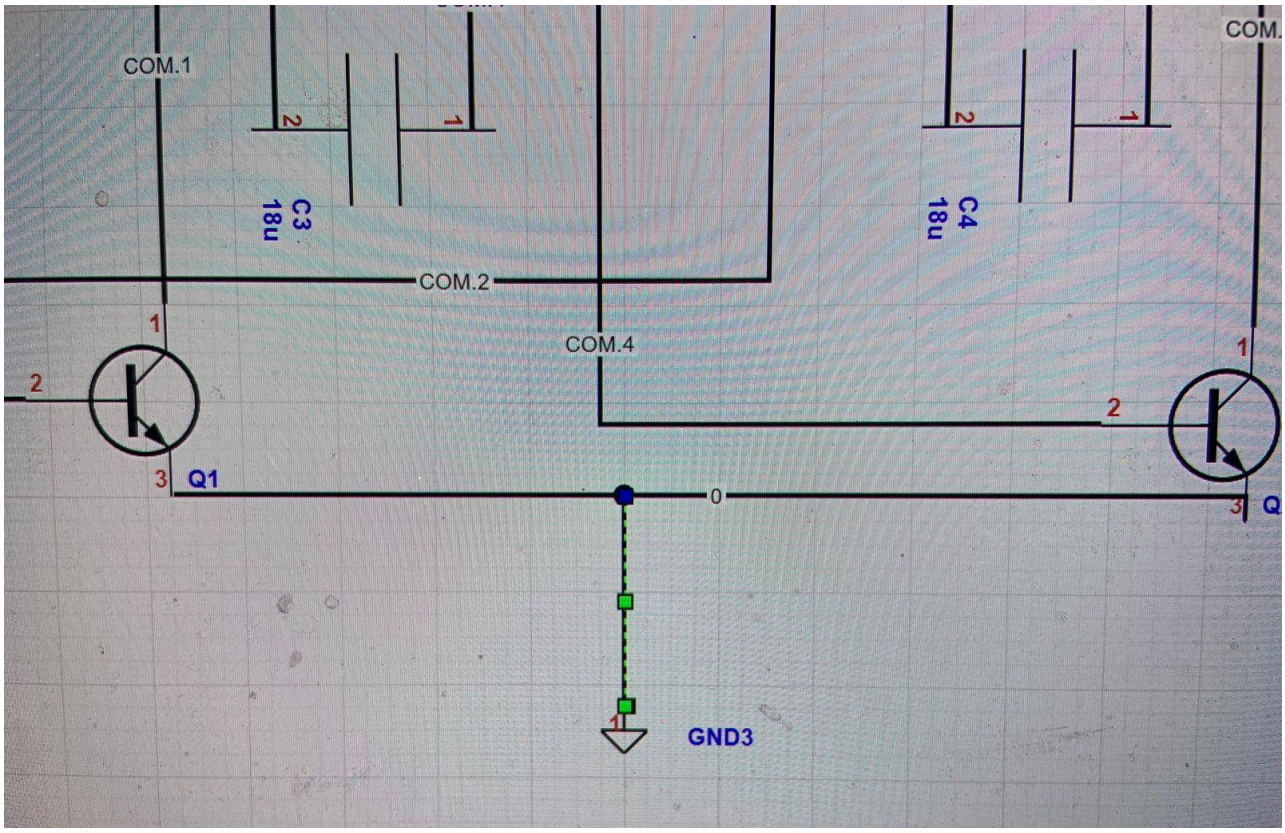
2.1.2 Outcome

PRs [#509]

- The missing condition check in the **generateNetlist** method was fixed, with the DFS code optimized.

- The cells in mxGraph were being incorrectly traversed and stored, where the nodeset with voltage**: 0 volts** , was not being converted to xml as an edge case while saving the circuit.

  Fixing these bugs made the netlists generated properly account for grounding, which was necessary for every circuit.



2.2 Installation Guide for developers on WSL

2.2.1 Problem Statement

The existing installation steps, initially designed for Ubuntu, required substantial modifications to enable efficient development from Windows systems using Docker for Windows Subsystem for Linux (WSL). The previous approach, relying on Oracle VM VirtualBox, resulted in significant latency, making it unsuitable for development purposes.

2.2.2 Outcome

- Developed comprehensive instructions for installing WSL (Ubuntu) and Docker on Windows systems.
- Outlined the steps required for building Docker containers specific to the project environment.
- Provided detailed guidance on manually running migrations within Docker containers to resolve issues related to the pre-loaded circuit gallery not displaying correctly.

2.3 Designed optimized ways to incorporate Subcircuits

2.3.1 Problem Statement

By implementing subcircuits in our EDA tool, we empower users with a powerful mechanism to encapsulate reusable circuit fragments into modular blocks. Subcircuits, also known as hierarchical circuits, enable designers to create complex circuits by assembling and interconnecting smaller subcircuits, reducing redundancy and improving productivity.

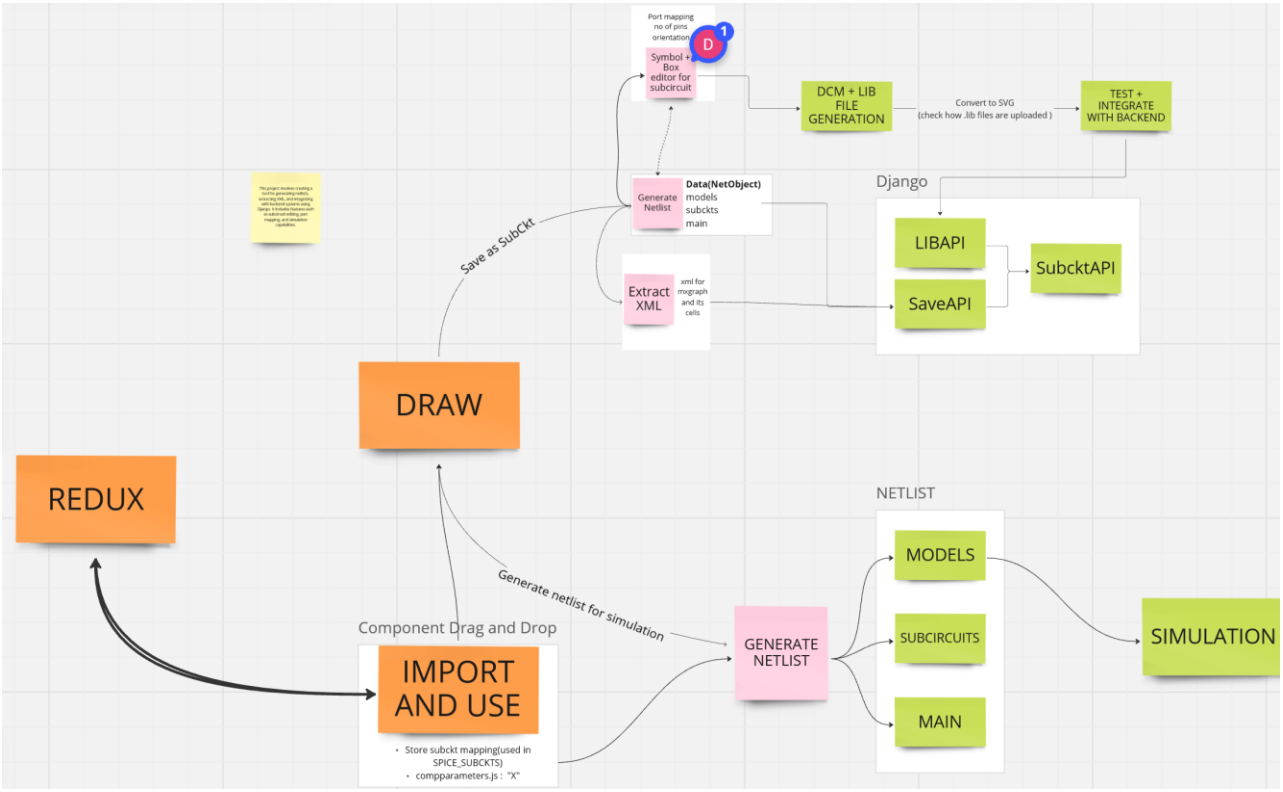
The goal was to understand the existing system architecture and devise ways to implement it cleverly.

2.3.2 Outcome

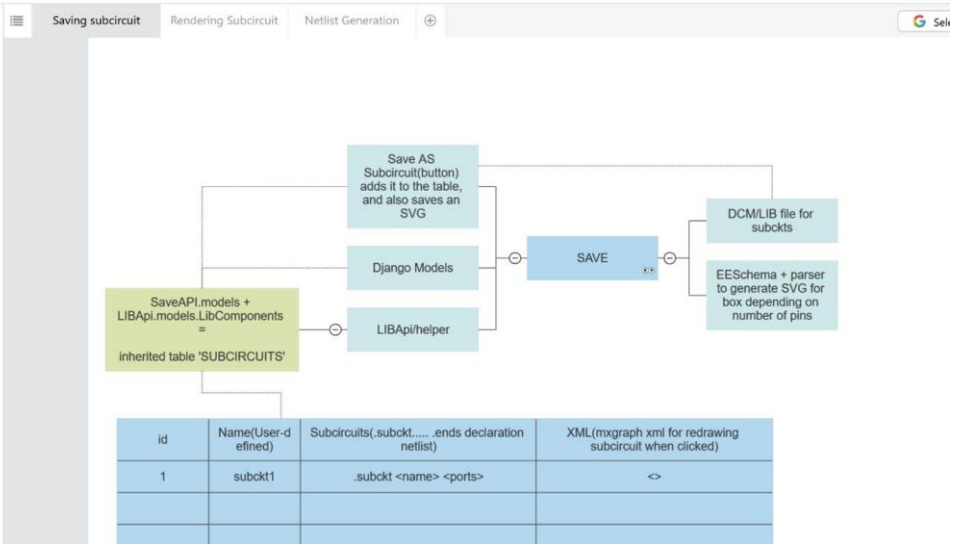
[Link to the Google Doc]

[Link to the Notion-based Enhancement Proposal]

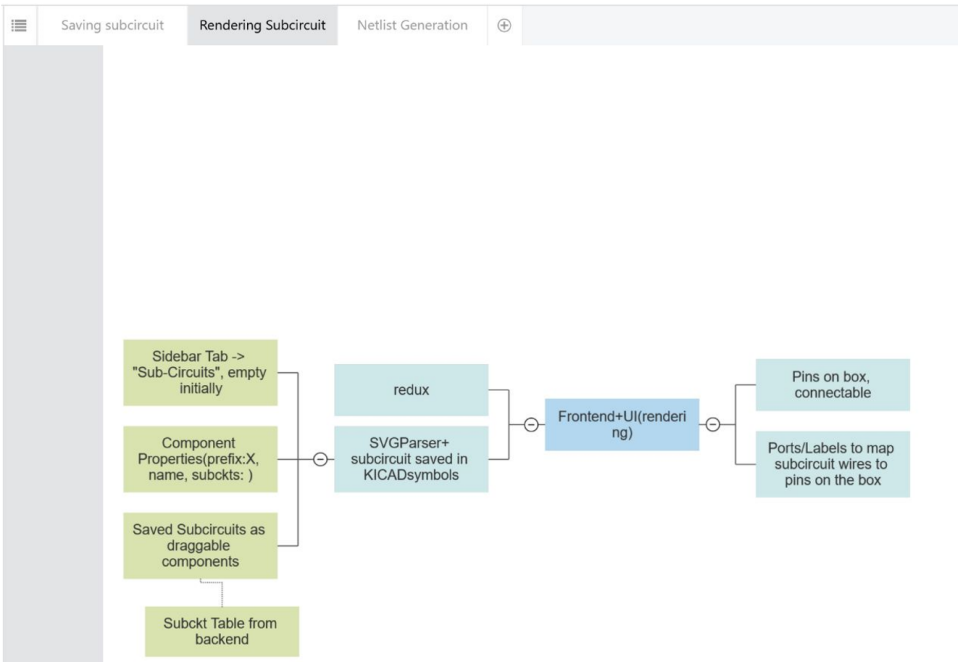
- Devised a schema and workflow to implement the subcircuits API



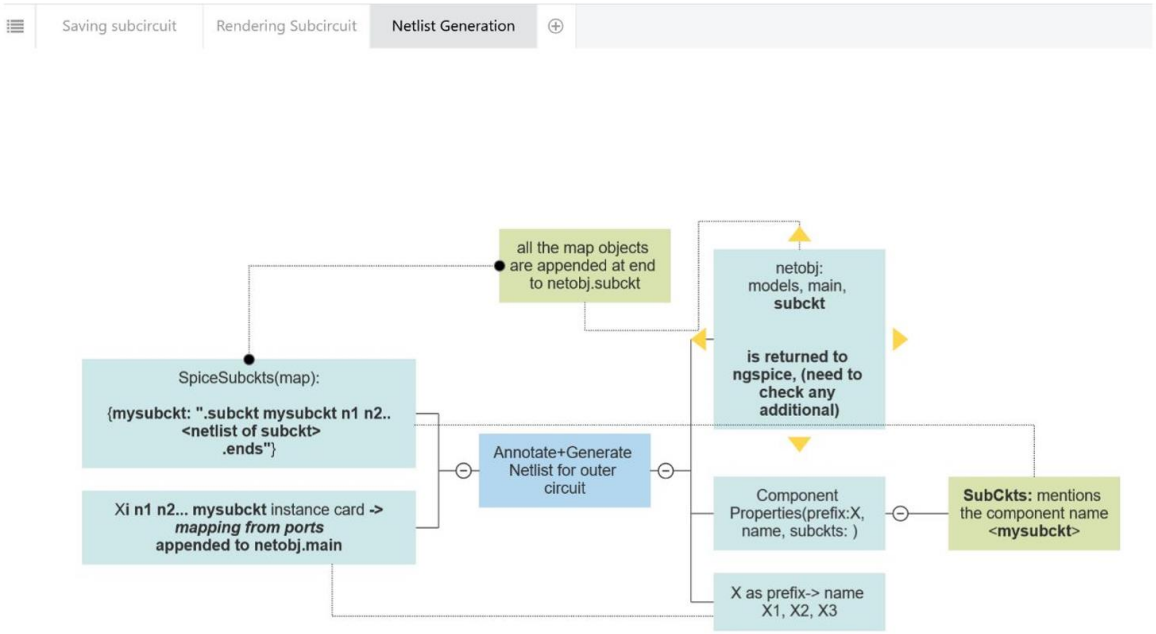
SAVING SUBCIRCUITS



UI+RENDERING



GENERATING + ANNOTATING NETLIST



- Schema: **subcircuitComponent**
  - **libcomponent:** Foreign key to the primary key of libcomponent schema
  - **savedData:** Foreign key to StateSave schema, containing the saved circuit to be used as subcircuit
  - **data_dump:** Contains the schema mxmodel data dump
  - **subcircuitDeclaration:** the NGSpice declaration of the subcircuit component, which shall be generated using the generateNetlist method on the subcircuit
  - **subcircuitPorts:** Mapping each port (label) to each wire that will be connected to the external pin. Will be according to NGSpice subcircuit conventions
- Let add or remove subcircuits like individual circuits, while making them accessible in the left pane as usable library comsponents

3 Bibliography

References

[1] https://esim-cloud.readthedocs.io/en/latest/eSim_on_Cloud/eSimGallery.html

[2] Ngspice Manual
http://ngspice.sourceforge.net/docs/ngspice-manual.pdf