SUMMER FELLOWSHIP REPORT

ON

ESIM ON CLOUD
Improving UI/UX, Management features and adding KiCad Compatibility

Submitted by

Nikhil Kumar
B.Tech in Computer Science and Engineering
NIT, Durgapur

Under the guidance of

Prof. Kannan Moudgalya
Chemical Engineering Department
IIT Bombay

August 2021

Acknowledgements

I, Nikhil Kumar a summer intern at **FOSSEE - ESIM on CLOUD** project, am extremely grateful to acknowledge all those who have been involved with the project and have assigned tasks believing in me and have helped me transform my ideas into something really unique.

I whole-heartedly thank **Mr. Nagesh Karmali** and **Ms. Firuza Aibara** for providing me with the oppurtunity to work on this project. I am really thankful to the mentors for their valuable suggestions. They were always there to show the right path and to help whenever I would hit a roadblock. With the help of their brilliant guidance and management I was able to complete all the tasks assigned to me properly and up to the mark.

Last but not the least I would like to thank all my colleagues who did an excellent job of maintaining cohesion and communicating changes all throught the fellowship period. Together we have been able to achieve difficult tasks and that too in a timely manner.

Contents

1 INTRODUCTION **4**
- 1.1 Esim on Cloud — Overview . . . 4
- 1.2 Technolgoies Utilised . . . 4
- 1.3 Objective . . . 4

2 NEW FEATURES ADDED **5**
- 2.1 Component Library Management . . . 5
  - 2.1.1 Problem Statement . . . 5
  - 2.1.2 Outcome . . . 5
- 2.2 Improving the login system . . . 7
  - 2.2.1 Problem Statement . . . 7
  - 2.2.2 Outcome . . . 7
- 2.3 Fixing the undo/redo actions . . . 9
  - 2.3.1 Problem Statement . . . 9
  - 2.3.2 Outcome . . . 9
- 2.4 New key-bindings(keyboard shorcuts) for EDA editor page . . . 9
  - 2.4.1 Problem Statement . . . 9
  - 2.4.2 Outcome . . . 9
- 2.5 Opening KiCad eeschema .sch files directly into the eSim EDA editor . . . 10
  - 2.5.1 Problem Statement . . . 10
  - 2.5.2 Outcome . . . 10
- 2.6 Improvements in Netlist generation . . . 10
  - 2.6.1 Problem Statement . . . 10
  - 2.6.2 Outcome . . . 10
- 2.7 Adding a new simulation type - Noise Analysis . . . 11
  - 2.7.1 Problem Statement . . . 11
  - 2.7.2 Outcome . . . 11
- 2.8 Bugfix: Component Duplication . . . 12
  - 2.8.1 Problem Statement . . . 12
  - 2.8.2 Outcome . . . 12

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 platform for electronic simulations. This system allows users to design and suimulate 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 Technolgoies 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 improve current features during the period of our fellowship. I have been working on user interactions and have added some management features as well. I have tried to the best of my ability and technical experience to improve the mechanisms at play here. The editor page is one place where most of my contributions can be seen at work.
The following is a brief list of the features that I have worked on and added to our eSim system.

1. Added feature to upload of new library and documentation files and integration into the eda system from the admin panel as well as the frontend.
2. Improvements in the login system.
3. Improvements in the the undo/redo actions
4. New key-bindings for EDA editor page
5. Added feature to import schematic files (.sch) files from KiCad (eeschema editor) directly into esim eda
6. Improved netlist generation feature
7. Added a new simulation type - Noise Analysis

2 NEW FEATURES ADDED

2.1 Component Library Management

2.1.1 Problem Statement

Component libraries are added into the system only while building the containers and as such the only way to add new libraries is to add them into a specified location and then build the containers once again. This might cause some downtime due to the delay in rebuilding the system and also one would need access to the server to do so. So I was instructed to work on such a system in which admins as well as normal users can upload libraries to our system from the admin panel as well as the frontend without causing any interruption in the service.

2.1.2 Outcome

PRs #247, #259

- A new model called **LibrarySet** was created to store collections of libraries uploaded by users and group them together for easy management. These LibrarySet model objects are related to one user object, they have a name given by the admin for sets added through the admin panel and system generated if its uploaded by a user from the frontend. These sets can be set as default so that they are visible to all users. Libraries in LibrarySets uploaded by admins always remain visible to all users irrespective of default flag. The difference that the default flag makes here is that default marked sets will be activated in the workspace by default and the other ones show up as additional libraries that can be added into the workspace at will.
- Libraries can be added to existing LibrarySets through the admin panel.
- Users are alloted one LibrarySet for their account which stores all their uploaded libraries so that the user's actions do not affect other users' libraries.
- In the frontend the libraries can be managed through a pop up modal in the editor's toolbar. Libraries are divided into 3 categories viz. default, additional and uploaded. Default and additional are managed by the admins and uploaded section houses libraries uploaded by the user and has an option to upload more libraries.
  Default and additional libraries have a single toggle to add or remove it from the workspace while uploaded ones have an additional option to delete the library.
  Users can look what components are provided by a library using the library button in the toolbar without including it in the workspace.
- Libraries can be moved from default to additional LibrarySet and vice versa through the click of a button in the admin panel. A button is added on every library in a LibrarySet clicking which the library is removed from that LibrarySet and moved to a default or additional LibrarySet accordingly and if such a LibrarySet doesn't exist a new one is created and then the move is done.

- Added management command that can be used to add libraries from any location on the system using just the terminal. An example:

```
python manage.py load_default_libs --username=admin\
 --location=kicad-symbols/default/ --default
```

  Where:
  `--username`: The username of the user to associate the LibrarySet with
  `--location`: The absolute path of the directory which holds the libraries
  `--default`: Optional flag to set a LibrarySet as default or not

- Default and additional libraries are added into the system at build time through the above mentioned `load_default_libs` command.
  Libraries in the default set:

  - eSim_Sources
  - power
  - pspice
  - Transistor_BJT

  Libraries in the additional set:

  - 4xxx
  - Analog
  - Device
  - Diode
  - eSim_Hybrid
  - LED
  - Motor
  - Oscillator
  - Transistor_FET
  - Transistor_IGBT
  - Triac_Thyristor

- When a circuit is saved the editor stores which libraries were active at that time in the workspace when the save is opened later.

- The saved circuits in the system which used the old library system would become invalid, so a python script was written to rescue those saves. The script as well the steps are in this GitHub gist.

2.2 Improving the login system

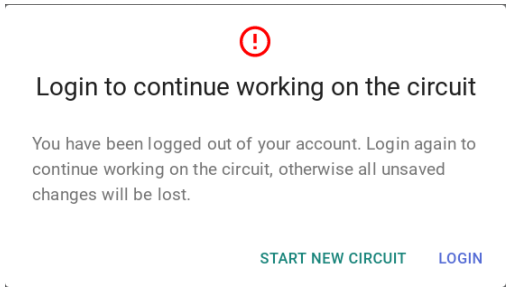
2.2.1 Problem Statement

The login system in place did work but lacked certain quality of life features like enter key to login while entering username or password. While at it work on improving the all around system with concerns regarding usage of the website on multiple tabs and protect user data and preserve the work so that it is not lost when some action is performed.

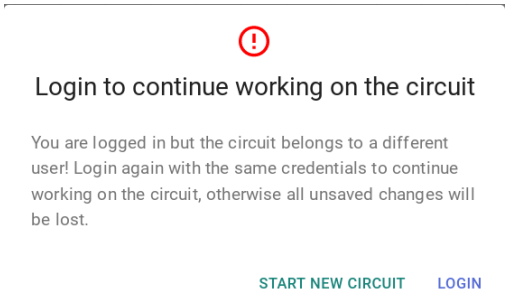
2.2.2 Outcome

PRs #256, #266

- Users can login by pressing enter key when either username or password fields are focussed.
- Login and Logout actions are now reflected across all browser tabs. It is done by checking the browser's localStorage for changes which indicate such an event happened and then appropriate action is taken. This is extremely needed because logout deletes the token at the back-end so other tabs will be making requests with an expired token and some error might pop up potentially causing data loss.
- When an editor tab is open and another window performs the logout process the editor on that page is locked until the user logs in again with the same account. They are also given an alternate option to start a new circuit. If they login using a different account the editor page remains locked and prompts the user to login again to the correct account.



(a) The user logs out from another tab and now the editor is locked



(b) The user logs in from a different account

Figure 1: Login Prompts on Editor

- Clicking logout in the editor tabs now raises a confirm dialog box giving options to cancel or go through with the logout.
- Added tooltip for the show password button in login form.
- Login pages can be opened in a new tab which close upon the completion of login thus preserving work in the initiating tab. If close parameter is passed in the URL of the login page the page closes after the login is done.

- A management command has been added to create a superuser from the terminal without any user input contrary to django's default createsuperuser command.

  ```
  python manage.py createsuperuser_noinput --username=admin --password=admin
  ```

  Where:
  - `--username`: The username of the superuser to be created
  - `--password`: The password for the superuser

2.3 Fixing the undo/redo actions

2.3.1 Problem Statement

The undo actions on components removes first the numbers one by one and then the actual component and vice versa for redo. Every action follows the same order. Expected behaviour is that all numbers on the component should move wlaong with the component as one unit.

2.3.2 Outcome

PR #282

- Each action produces a series of mxEvents which can be used to predict what action was performed just by looking at the order and number of arguments in those mxEvents.

2.4 New key-bindings(keyboard shorcuts) for EDA editor page

2.4.1 Problem Statement

The keybindings that are currently in the editor are not standard ones used across multiple applications and as such I had to confirm they would be remapped to standard ones and add other useful shortcuts.

2.4.2 Outcome

PR #293

- The following keybindings were modified and/or added:
  - Save: Ctrl + S
  - Print: Ctrl + P
  - Open Dialog: Ctrl + O
  - Export as JSON: Ctrl + E
  - Export as Image: Ctrl + Shift + E
  - Rotate Component: Ctrl + R
  - Delete Component: Del
  - Clear All: Shift + Del
  - Undo: Ctrl + Z
  - Redo: Ctrl + Shift + Z
  - Zoom In: Ctrl + +
  - Zoom Out: Ctrl + -
  - Default Zoom: Ctrl + Y
- Added the ability to rotate components in clockwise and counter clockwise directions.
- The component terminal numbers rotates along with the component.

2.5 Opening KiCad eeschema .sch files directly into the eSim EDA editor

2.5.1 Problem Statement

The eSim EDA system is in many ways inspired by the KiCad Eeschema editor, so it is natural that someone might want to migrate their workflow from the desktop application to our cloud based system. For that our system should be able to open and properly simulate .sch files generated by eeschema.

2.5.2 Outcome

PR #345

- Circuits designed using the default KiCad libraries can be easily imported into the our system. There is an option to upload kicad sch files in the open dialog box in the editor's toolbar.
- All the wire connections are proper and the circuit can be simulated, although circuits with multiple levels of wire—to—wire connections or excessive branching the system fails to detect the proper connections. This due to how the wire paths are stored in the files generated by KiCad.
- Most of the things had to be reverse engineered by observing the files generated and are based on my understanding of the KiCad system.

2.6 Improvements in Netlist generation

2.6.1 Problem Statement

The netlist generator didn't used to work for multiple components or wires connected at the same terminal and would name them differently while all the nodes at the same potential should have the same name.

2.6.2 Outcome

PR #336
Worked with Kaustuv K Chattopadhyay on this issue.

- Depth First Search algorithm was used to find all the nodes that are connected and thus would be at the same potential. Such nodes are then given the same names — "0" for ground and "COM.X" for all other edges, where X is a number.
- Added another check to the Electrical Rules Check wherein shortcircuits are checked by checking if multiple terminals of component are at the same potential. The user is then warned of it by a popup at the bottom left of the screen.

2.7 Adding a new simulation type - Noise Analysis

2.7.1 Problem Statement

Noise Analysis was needed to be added to the simulation types available. Noise Analysis calculates the output electrical noise and the input electrical noise in a circuit. The output is generated for a particular node at which the output noise is desired.

2.7.2 Outcome

PR #352
Worked with Kaustuv K Chattopadhyay on this issue.

- In the simulation tab in the editor the user can select noise analysis and the required netlist will be generated and noise analysis will be run.
- The .noise command is used to run noise analysis on the ngspice terminal and looks like this

  ```
  .noise v(output <,ref>) src ( dec | lin | oct ) pts fstart fstop
  ```

  The user has to select the following parameters for analysis:

  - `output` and `ref` — Voltage between Nodes — nodes between which the noise wil be calculated. "ref" is optional and the default is ground if nothing is specified.
  - `src` — Input Voltage SRC — The independent source to which input noise is referred.
  - graph variation — Type — options are `Decade` – `Linear` – `Octal`
  - `pts` — Points/Scale — Determines the precision of the graph
  - `fstart` — Start Frequency — The starting frequency of the range of the analysis
  - `fstop` — Stop Frequency — The stopping frequency of the range of the analysis

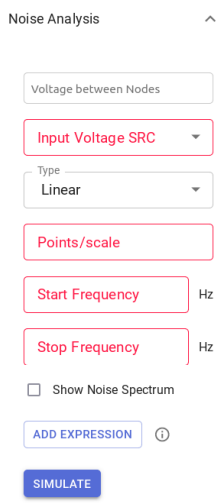


Figure 2: Noise Analysis Window

2.8 Bugfix: Component Duplication

2.8.1 Problem Statement

Clicking on a component in the sidebar and then dragging it into the editor causes it to create copies (2 for each click).

2.8.2 Outcome

PR #281 The bug occured because a `useEffect` was being triggered wrongly because of multiple dependencies. Emptying the dependency array for that `useEffect` fixed the issue.

Conclusion

FOSSEE stands true to its name in promoting open-source software in the educational field where it is most required. This project is one of the many endeavours that have been undertaken by FOSSEE and is quite a first in this field. This project involves a lot of desktop applications and tries to port it over to the cloud which is very commendable. I think it is important to develop such software for educational institutions across the country to use to complement the lack of laboratories in remote areas or to simulate the physical labs impossible currently due to the covid pandemic.

I am very impressed and thrilled with the vision that out mentor **Mr. Nagesh Karmali** possess for this project. The fact that there exists no such comprehensive software for the web platform made working here feel even more innovative and fresh. I was able to get more insight on the process of PCB design and manufacturing through our long discussions during my time here as a fellow.

Working here has made me put forward multiple solutions and critically think and find faults with my own ideas to problems ranging from user interactions to inherent behaviour of the frameworks and languages that go unnoticed otherwise.

Future Involvement

There are things that I would want to work on beyond the fellowship. One of them is sub-circuitry support. Subcircuitry support is already present in ngSpice, just plugging in the subcircuit model (See Chapter 2.4 of [10] for more details) to the netlist generation, a system to create a circuit and a symbol for it are needed.
The KiCad compatibility is also one area where we can add new features, like generating save files supported by KiCad for those who need it. It would also solve an issue of our current method of saving a circuit, changes to mxGraph part of the frontend sometimes renders the saves in a vulnerable state. A way to generate circuits programmatically would be great instead of just saving the state of the circuit diagram.

References

[1] Django Documentation.

[2] django-inline-actions documentation.

[3] react Material UI documentation.

[4] React Redux documentation.

[4] The last guide to the useEffect Hook you'll ever need — Blog post by logrocket.

[5] StackOverflow answer on adding event listeners for localStorage in ReactJS.

[6] Using React hooks to add event listeners to the DOM.

[7] mxGraph Documentation.

[8] KiCad Legacy file format documentation.

[9] KiCad Forum post on Eeschema Orientation Matrix Explanation.

[10] ngSpice manual.

[11] The Half-Wave Rectifier Circuit, Netlist for the half-wave rectifier for noise analysis — From KiCad forums.