SUMMER FELLOWSHIP REPORT

ON

Extension of Functionality and Study of Circuit Simulations in Esim Cloud

Submitted by

Kaustuv Kanchan Chattopadhyay

B.Tech in Electrical Engineering

NIT, Durgapur

Under the guidance of

Prof. Kannan Moudgalya

Chemical Engineering Department

IIT Bombay

August 2021

Acknowledgements

I, Kaustuv Kanchan Chatopadhyay a summer 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** and **Ms. Firuza Aibara** for being such a warm and wonderful host and their kind guidance. Throughout the fellowship they have been the pillar of support, extremely understanding and given constant motivation to successfully complete my tasks. I have learnt a lot through them and will forever remain indebted to them for providing such a wonderful experience.

Last but not the least I would like to thank all my colleagues who were an amazing bunch of people never failed to extend an arm in any roadblock that I faced.

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 platform 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 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. My work consisted of a combination of research and full stack development.I have tried to increase the functionality of the both from the admin as well as from the architectural point of view.

The following is a brief list of the features that I have worked on and added to our eSim system.

1. Added feature of view the Simulation Status Notifications
2. Analysis Improvements
3. Support For Dynamic Update in Gallery
4. Built Several circuits and Recorded Their Responses
5. Redesigned the Netlist generator
6. Added a new simulation type - Noise Analysis

2 NEW FEATURES ADDED

2.1 Simulation Status Notification

2.1.1 Problem Statement

The simulations are run in the backend using ngspice. Now the result for the simulation isn't received immediately but after a delay ranging from few milliseconds to a couple of seconds. From the point of view of the user, in case of a unsuccessful simulation, the message vaguely said that something went wrong.

- Find a way to propagate the error message from ngspice to the user.
- Give admin the privilege to restrict time limit for a simulation

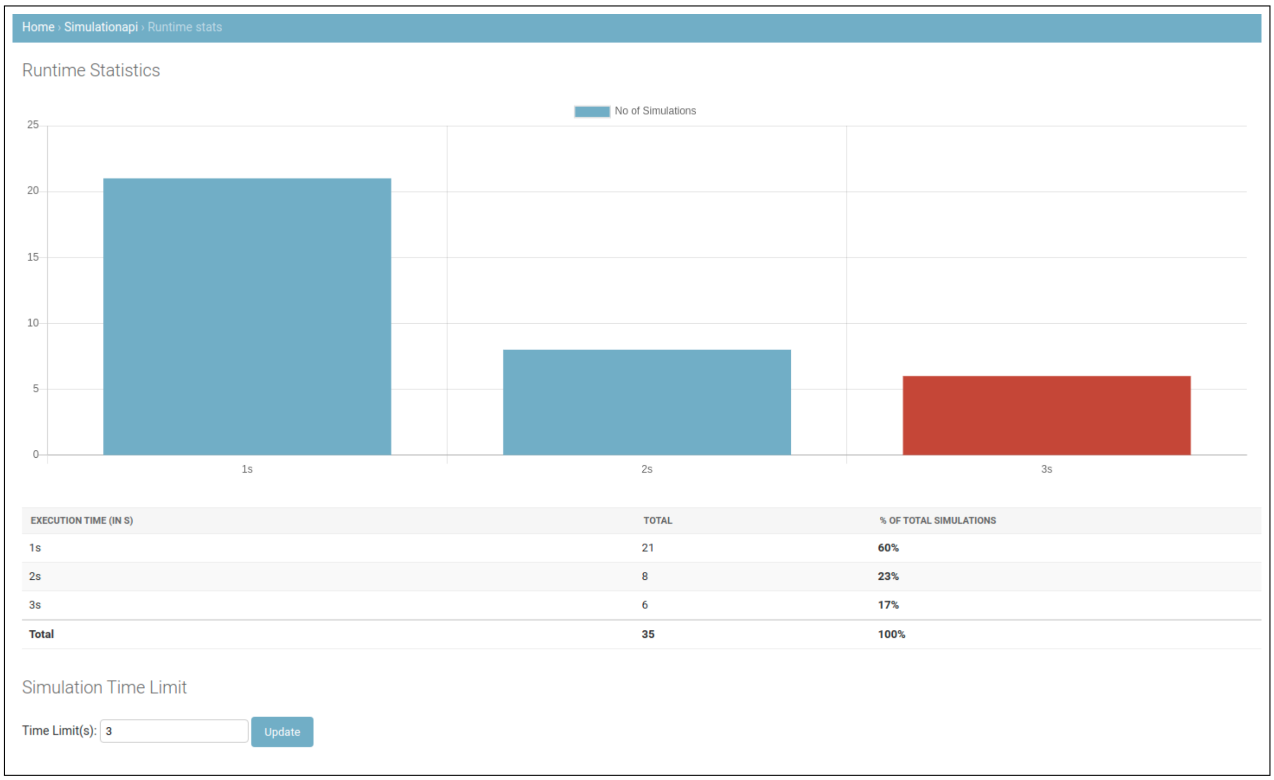
2.1.2 Outcome

`PRs` #277

- The error message which was generated by ngspice was recorded and sent with the results at the corresponding API endpoint. A notification component was added in the frontend code base.
- Time limit was set in the Celery backend. A new model was created named **runtimeStat** which would store the statistics of the different simulations which have been run till this point of time in the form of seconds taken and no of such simulations. The limit of simulations are stored in the first row of model **Limit**
- Schema: **runtimeStat**
  - **exec_time:** The amount of time taken for a simulation in seconds
  - **qty:** No of such simulations which took exec_time amount of seconds
- Schema: **Limit**
  - **timeLimit:** The time limit of the simulations in the system in seconds.
- Three different status reports are achievable for a simulation in both schematic editor as well as in Simulator.
  - Successful (Green): This indicates the simulation requested was successfully ran and the results are available.
  - Running (Yellow):: This indicates the simulation is being run at the backend and the results will be updated as soon as either the verdict arrive or time limit expires
  - Unsuccessful (Red): This indicated the simulation requested ran into an error in the backend and has been further categorized into two different parts:

* Time Limit Exceeded (TLE): This happens when the simulation exceeds the time limit set by the backend, which is set to infinite time by default. This verdict doesn't guarantee that there wasn't any other error which might be mutually exclusive with TLE.
* Error: This indicates that there was an error while trying to run a simulation and it has been aborted due to some error. The detailed error message is shown as a popup to the user

– Runtime Statistics Example



2.2 Analysis Improvements

2.2.1 Problem Statement

Improve the overall experience of Schematic Editor in terms of functionality by adding required utilities and options.

2.2.2 Outcome

`PRs` #248

- Multiple selection of nodes in the analysis methods
- Use Initial Conditions (UIC) option in Transient Analysis
- Auto scaling in Non Graphical Output
- Engineering/scientific notation support in non graphical output
- Added parameters for various circuit components
- Replaced print all > data.txt in netlist preview with plot

2.3 Support For Dynamic Update in Gallery

2.3.1 Problem Statement

Presently, the gallery is being fetched from a json file and can't be updated at run time. Fetch the gallery circuits from the backend and add support for adding to gallery at run time.

2.3.2 Outcome

PR #328

- A new model named **Gallery** was added in the backend. Conscious decision to not add the gallery schematics to the **StateSave** model was made to segregate the gallery circuits from the user saved circuits. This would allow to move the gallery dump data easily.
- Schema: **Gallery**
  - **id:** Auto index field
  - **save_id:** Unique id for each schematic
  - **data_dump:** Contains the schema mxmodel data dump
  - **name:** Name of the circuit
  - **description:** Short description of the circuit
  - **media:** Circuit png which will be shown in gallery
  - **shared:** Permission on whether the schematic can be shared or not
  - **save_time:** Auto time save field
  - **is_arduino:** Arduino or eSim Schematic
  - **esim_libraries:** The libraries which are required to run the schematic. This isn't transferable from one instance to another currently due to conflicting library ids
- A new custom group named Staff has been added with appropriate permissions to let add or remove schematics from gallery
- Now new gallery circuits can be added by custom group Staff members

2.4 Built Several Circuits and Recorded Their Responses

2.4.1 Problem Statement

Build and generate as many circuits as possible to try and simulate circuits which are regularly used in electrical and electronics engineering and gather the associated data.

2.4.2 Outcome

`PR` `Link To Spreadsheet`

Circuits which were added to gallery include:

- Astable Multivibrator
- Half Wave Rectifier Circuit(Noise Analysis)
- Resistive Divider with AC input
- Wheatstone Bridge

2.5 Redesigned the Netlist Generator

2.5.1 Problem Statement

Currently, the netlist generator didn't consider the fact that the potential at all connected nodes should be same and hence should be annotated with the same name, except for ground. Find a way to make the netlist accurate.

2.5.2 Outcome

`PR` #336
Worked with Nikhil Kumar 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 shortcuts 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.

2.6 Adding a New Simulation Type - Noise Analysis

2.6.1 Problem Statement

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

2.6.2 Outcome

`PR` #352 Worked with Nikhil Kumar on this issue.

- In the simulation tab in the editor the user can select noise analysis and the required netlist wil 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` — nodes between which the noise wil be calculated. "ref" is optional and the default is ground if nothing is specified.
  - `src` — The independent source to which input noise is referred.
  - graph type — options are `Decade` `Linear` `Octal`
  - Points/Scale
  - `fstart` — The Start Frequency of the analysis
  - `fstop` — The Stop Frequency of the analysis

3 Bibliography

References

[1] Fundamentals of Electric Circuits 7th Edition
`by Charles Alexander and Matthew Sadiku`

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

[3] CMOS: Circuit Design, Layout and Simulation
`by R. Jacob Baker`