RISC-V Simulator
Web Application

Sai Srirama Saran Konala - AI23BTECH11023
Lakshdeep Singh - CS23BTECH11031

Simulator Configuration:

instruction set: RISC-V
registers : 64 bit x 32
memory: 1024 bytes

Coding Approach - Workflow

Frontend: REACT JS

Home Page:

- There are two primary components - Editor and Registers (apart from the Navbar).
- The code is stored in the local memory of the browser and when the user hits the run button. The component sends a POST request to the backend with the code, and in return gets the updated states of registers, memory and log.
- The component sends a POST request to the backend with the code, and in return gets the updated states of registers, memory and log.
- The obtained data is then made available to other components by creating a context of it and updating them accordingly.

Memory Page:

- The memory is 1024 bytes long and it is updated from the data context.

Backend: NODE JS

- Used spawn from child_process module to run the simulator.cpp files from the server file.
- Fetched the data from the stdout in the terminal.
- The states of registers and memory are sent to the frontend as a JSON response.

Limitations

- It does not support comments.
- It does not have a debugger.
- It does not support syntax highlighting.