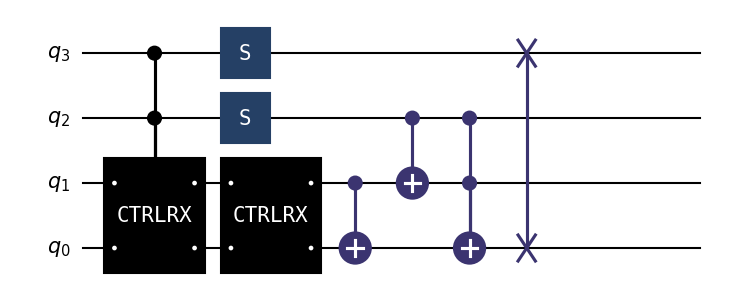
Week#5

Rushiraj Gadhvi

2024-06-27

## What did I do this week?

1. **Bugs fix in MatRenderer**  
   Resolved some of the bugs present in MatRenderer, specifically the connection indicators where not getting rendered properly when bulging of gates was set to zero.



1. **Exploration for TextRenderer**  
   Another significant area of my work involved the exploration of TextRenderer. This phase included experimenting with various approaches to establish an efficient workflow. A critical part of this process was the manual selection and finalization of a specific set of ASCII characters. These characters are integral for accurately representing circuit elements in text form. Drawing inspiration from methodologies used in libraries like Qiskit, I gained valuable insights into optimizing text-based rendering for circuit diagrams.

For reference, here are some of the ASCII codes and corresponding symbols selected for rendering purposes:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | **ASCII Code** | **Symbol** | | --- | --- | | 179 | │ | | 180 | ┤ | | 191 | ┐ | | 192 | └ | | 193 | ┬ | | 194 | ┴ | | 195 | ├ | | 196 | ─ | |  | | **ASCII Code** | **Symbol** | | --- | --- | | 217 | ┘ | | 218 | ┌ | | 186 | ║ | | 187 | ╗ | | 188 | ╝ | | 200 | ╚ | | 201 | ╔ | | 202 | ╦ | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **ASCII Code** | **Symbol** | | --- | --- | | 203 | ╩ | | 204 | ╠ | | 205 | ═ | | 206 | ╣ | | 185 | ╛ | | 45 | - | | 88 | X | | 254 | █ | |

**Example of Rendering Style** To visualize the rendering style, below are examples of circuit diagrams rendered using our current approach:

|  |
| --- |
|  |

1. **Finalised on WorkFlow for TextRenderer**  
   As for the finalized workflow, for easy maintainance of code in future. A method to somehow utilize MatRenderer’s funciton for the TextRenderer would be the optimum solution. This can be achieved by trying to replicate the layer by layer rendering process that was used in MatRenderer, but this would need further verififcation (plenty of edge cases are a possibility here). The main issue with this is MatRenderer uses matplotlib’s feature to renderer objects onto different layers which is not possible in the case of ordered string rendering.

## Points Discussed in Meet

* Finalization of the API structure to streamline functionality.
* Workflow for Text Renderer, including the idea to create a seperate Base Class for MatRenderer and TextRenderer.
* We can improve the extend\_lines() funtion in MatRenderer

## Plan for Next Week

Looking ahead, the following tasks are prioritized for the upcoming week:

* Clean the code, and move latex plotting out of circuit.py
* Transferring LaTeX plotting functionalities out of circuit.py for improved code organization.
* Continuing the implementation and refinement of the text rendering capabilities.