Formalizing Neuromorphic Architecture

Nikita Sharma

September 29, 2022

1 Introduction

For my final project, I want to formalize a neuromorphic architecture, write programs that it should be able to run, and prove some properties of these programs.

2 Neuromorphic Hardware

As we observe the coming end to Moore's law, new forms of computation are being explored. Neuromorphic architecture [3] is one of these such proposals. Neuromorphic hardware takes inspiration from the brain for its structure. The architecture is composed of ”neurons” and ”synapses” that are all connected. These components communicate with one another using ”spikes”. This lends itself to an event driven structure where large components of the hardware are left idle, leaving us with very small power utilization.

In terms of hardware, several neuromorphic architecture implementations use memristors [1] because of their capabilities to remember and store value even without power. This allows us to move further away from CMOS and traditional von Neumann designs.

Neuromorphic architectures are generally being used to implement spiking nerual networks (SNNs). However, it has been shown that this architecture can be used to compute general graph algorithms [5] and it is even turing-complete [2].

From my understanding thus far, these architectures are programmed by specifying a ”neuron”/”synapse” architecture, spiking a neuron, and allowing the spike to propogate across the ”neuron”/”synapse” network, and then reading back values [4].

3 Goals and Timeline

1. [**Oct. 7th**] Get a formal understanding of neuromorphic architectures and how to program them

2. [**Nov. 11th**] Represent the architecture in Coq

3. **[Nov. 25th]** Write some programs and prove that they work as intended
4. **[Dec. 1st]** Finish project report
5. Work with Sindhu and Aditya to see if we can combine our projects
6. Prove properties about this architecture

References

[1] L. Chua. "Memristor-The missing circuit element". In: *IEEE Transactions on Circuit Theory* 18.5 (1971), pp. 507–519. DOI: 10.1109/TCT.1971.1083337.

[2] Prasanna Date et al. "Neuromorphic Computing is Turing-Complete". In: *Proceedings of the International Conference on Neuromorphic Systems 2022*. ICONS '22. Knoxville, TN, USA: Association for Computing Machinery, 2022. ISBN: 9781450397896. DOI: 10.1145/3546790.3546806. URL: https://doi.org/10.1145/3546790.3546806.

[3] Robert A. Nawrocki, Richard M. Voyles, and Sean E. Shaheen. "A Mini Review of Neuromorphic Architectures and Implementations". In: *IEEE Transactions on Electron Devices* 63.10 (2016), pp. 3819–3829. DOI: 10.1109/TED.2016.2598413.

[4] Catherine D. Schuman et al. "A Programming Framework for Neuromorphic Systems with Emerging Technologies". In: *Proceedings of the 4th ACM International Conference on Nanoscale Computing and Communication*. NanoCom '17. Washington, D.C.: Association for Computing Machinery, 2017. ISBN: 9781450349314. DOI: 10.1145/3109453.3123958. URL: https://doi.org/10.1145/3109453.3123958.

[5] Catherine D. Schuman et al. "Shortest Path and Neighborhood Subgraph Extraction on a Spiking Memristive Neuromorphic Implementation". In: *Proceedings of the 7th Annual Neuro-Inspired Computational Elements Workshop*. NICE '19. Albany, NY, USA: Association for Computing Machinery, 2019. ISBN: 9781450361231. DOI: 10.1145/3320288.3320290. URL: https://doi.org/10.1145/3320288.3320290.