Astrocyte-Integrated Dynamic Function Exchange in Spiking Neural Networks

Murat Isik[1][0000−0002−0907−7253] and Kayode Inadagbo[2][0009−0009−9512−3321]

[1] Stanford University, Stanford, CA, USA
mrtisik@stanford.edu
[2] Prairie View A&M University, Prairie View, TX, USA
kayodeinadagbo@gmail.com

Abstract. This paper presents an innovative methodology for improving the robustness and computational efficiency of Spiking Neural Networks (SNNs), a critical component in neuromorphic computing. The proposed approach integrates astrocytes, a type of glial cell prevalent in the human brain, into SNNs, creating astrocyte-augmented networks. To achieve this, we designed and implemented an astrocyte model in two distinct platforms: CPU/GPU and FPGA. Our FPGA implementation notably utilizes Dynamic Function Exchange (DFX) technology, enabling real-time hardware reconfiguration and adaptive model creation based on current operating conditions. The novel approach of leveraging astrocytes significantly improves the fault tolerance of SNNs, thereby enhancing their robustness. Notably, our astrocyte-augmented SNN displays near-zero latency and theoretically infinite throughput, implying exceptional computational efficiency. Through comprehensive comparative analysis with prior works, it's established that our model surpasses others in terms of neuron and synapse count while maintaining an efficient power consumption profile. These results underscore the potential of our methodology in shaping the future of neuromorphic computing, by providing robust and energy-efficient systems.

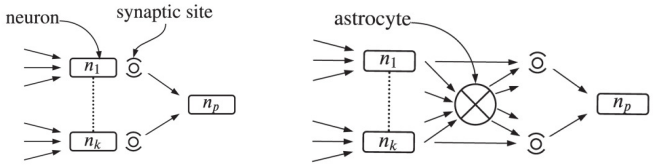
Keywords: Astrocytes, Spiking Neural Networks, FPGA Implementation, Dynamic Function Exchange, Fault Tolerance

1 Introduction

Fault tolerance has become a critical feature of today's increasingly sophisticated computational systems, which require not just high performance, but also continuous and reliable operation. This is especially true for neural networks that mimic the structure of the brain, pushing the limits of existing computing paradigms. Spiking Neural Networks (SNNs), a type of artificial neural network patterned after the brain's neuronal dynamics, are energy efficient, use time-dependent data processing, and have bio-plausible algorithms for learning. In spite of this, SNNs are susceptible to faults and failures, which could disrupt their functionality and reduce their efficiency. Therefore, fault-tolerant mechanisms within SNNs need to be explored. Recent research has demonstrated that astrocytes play a crucial

role in regulating neuronal activity and synaptic transmission in the brain. It has long been believed that neurons contributed significantly to the resilience and adaptability of biological neural networks, but astrocytes have now been found to play a much more important role which is shown in Fig. 1. Dynamically modulating neuronal activity based on state, they effectively support fault tolerance at the molecular level. The hypothesis of integrating astrocytic mechanisms into SNNs is an exciting prospect, potentially leading to dynamic adjustment for fault tolerance in these systems [5] [2] [6].

Field Programmable Gate Arrays (FPGAs) are reprogrammable silicon chips that can be customized to perform complex computations in parallel, making them ideally suited for implementing SNNs. FPGAs have been increasingly used for emulating SNNs due to their high degree of parallelism, energy efficiency, and low latency. Further, their inherent re-programmability makes them a prime candidate for implementing adaptive mechanisms, such as those inspired by astrocytes, to handle faults dynamically. This could potentially enable SNNs implemented on FPGAs to autonomously adapt in the face of faults, mimicking the resilience observed in biological neural networks [4] [3].



(a) Original neural network structure.

(b) Astrocyte-modulated neural network structure.

Fig. 1. Inserting an astrocyte in a neural network. [5]

In this paper, we explore how FPGA-implemented SNNs could benefit from astrocyte-powered dynamic adjustments to enhance fault tolerance. The purpose of this study is to investigate whether introducing astrocyte-inspired mechanisms could enhance network performance and reliability by reducing faults and failures. The rest of the paper is organized as follows: **Section II** discusses astrocytes' significance in SNNs and reviews related works. **Section III** describes SNN architecture and the integration of astrocytes. **Section IV** details our astrocyte-augmented SNN model, emphasizing hardware implementations. **Section V** evaluates the model's fault tolerance and efficiency, comparing it with other models and introducing the Dynamic Function eXchange technology. **Section VI** concludes with our key findings and suggests future research avenues.

2 Background

The principles of biological brains are reflected in SNNs, which are artificial neural networks. A key difference between them is the emulation of time-dependent

spikes or 'action potentials', which are the primary means of communication between neurons in the brain. The SNN is a powerful computational model capable of handling complex tasks such as pattern recognition, sensory processing, and motor control in a highly energy-efficient, low-latency manner. Recent advances in neuromorphic engineering have propelled research in this field, which aims to create hardware and software solutions that mimic neuronal spike dynamics [11]. Fault-tolerance techniques are essential for ensuring robustness and reliability of complex systems like SNNs, particularly when uninterrupted functionality is critical. Several methods have been proposed and implemented, ranging from redundancy and error correction codes to adaptive mechanisms that enable dynamic fault recovery [18]. The disadvantages of these traditional techniques are often increased resource consumption and decreased performance. Therefore, innovative solutions are needed that minimize these trade-offs while ensuring robust fault tolerance. Astrocytes once considered mere supporting cells in the brain, are now recognized as key players in regulating neuronal activity. The ability of biological neural networks to detect and modulate neural activity contributes to their adaptability and resilience [13]. The idea of integrating these astrocytic mechanisms into artificial neural networks to enhance their resilience and adaptability is a novel and promising area of research. Previous works have explored the implementation of SNNs on FPGAs for their advantages in parallelism, energy efficiency, and re-programmability [15]. However, the integration of astrocyte-inspired fault-tolerance mechanisms in such systems has not been adequately explored. This research seeks to fill this gap, extending our understanding of fault tolerance in SNNs and paving the way for more robust and adaptive neural network architectures. By examining how astrocyte-powered dynamic adjustments could enhance fault-tolerance in FPGA-implemented SNNs, this study could provide a valuable contribution to the fields of computational neuroscience and neuromorphic engineering.

3 Astrocyte and Spiking Neural Networks

Astrocytes constitute about 20-40% of the total glial population in the human brain. Studies have revealed that these molecules play an active role in neuronal signaling and information processing. The astrocyte extends its processes near neurons, where it senses and modulates neuronal activity through gliotransmission [16]. This remarkable capability motivates the integration of astrocyte mechanisms into SNNs, providing an intriguing avenue to enhance their fault tolerance and adaptability. An SNN is an artificial neural network that mimics time-dependent and event-driven communication between biological neurons through spikes or 'action potentials'. High temporal resolution, high power efficiency, and bioplausible mechanisms have made them a subject of keen interest [14]. It is possible to mimic the fault tolerance and dynamic adjustment of biological neural networks by incorporating astrocyte mechanisms into SNNs. A bidirectional communication system connects astrocytes to neurons. Neurotransmitters released by neurons can be detected and responded to by them, and the gliotransmitters released can

modulate neuronal activity. Among the main mechanisms of astrocyte-neuron interaction is the tripartite synapse model, in which astrocytes actively contribute to neuronal synaptic transmission [1]. Among the diverse effects of this interaction are the modification of synaptic strength, the regulation of local blood flow, and metabolic support for neurons, thus enhancing network resilience and adaptability. SNNs based on astrocyte functionality can incorporate these aspects to enhance their resilience. Synaptic weights can be modulated by astrocytes to balance neuron firing rates across a network, thereby preventing neurons from 'dying out' or 'overfiring' as a result of neural network models. Moreover, astrocytes are able to sense and respond to changes in neuronal activity, enabling them to design fault-tolerance mechanisms that dynamically adjust to faults in networks [8] [9]. The incorporation of astrocyte-neuron interactions into SNNs, especially those based on FPGAs, has yet to be explored in various computational neuroscience studies.

4 Method

4.1 Dataset

Our project is based on the DAVIS 240C Dataset, a unique collection of event-based data ideal for pose estimation, visual odometry, and SLAM. This dataset, generated using DAVIS 240C cameras by iniLabs, offers event-based images, IMU measurements, and motion-captured ground truth. Some datasets that utilized a motorized linear slider lack motion-capture or IMU data; however, their ground truth derives from the slider's position. The "calibration" dataset provides alternative camera models, with all gray datasets sharing identical intrinsic calibration. This dataset proves invaluable for image data analysis, particularly in SNNs and related domains [10]. For this project, we employ a subset of the DAVIS 240C dataset. Figure 2 showcases the DAVIS 240C event camera which was utilized to produce this dataset.

Fig. 2. DAVIS 240 DVS Event Camera

4.2 Training Details

In our implementation, the SNN is architected to emulate astrocyte functions using a subset of the DAVIS 240C Dataset that records astrocyte activity in response to neuronal behavior. The architecture is composed of:

- **Input Layer:** Simulates neuron-astrocyte interactions, customizable for specific neurological scenarios.
- **Astrocyte Layer:** Represents spiking astrocytes, processing inputs and relaying spike trains to the subsequent layer.
- **Output Layer:** Decodes the spike trains, producing responses analogous to biological outcomes from astrocyte activities.

During compilation, the aim is to synchronize the Output Layer's reactions with the anticipated responses in the training set. We employ the 'Adam' optimizer, recognized for efficiently addressing complex problems. Performance evaluation utilizes the 'accuracy' metric, with the 'EarlyStopping' callback integrated during training to mitigate overfitting. Following training, outcomes are juxtaposed with validation data, assessing accuracy, precision, and recall. This implementation paves the way for deeper explorations into astrocytic roles in SNNs. Subsequent iterations may further refine the model and incorporate additional cellular dynamics, with a recommendation to consider advanced SNN metrics such as spike timing and spiking rate accuracy.

4.3 Hardware Implementation

Hardware implementation is vital for real-world applications, particularly in computationally-intensive tasks. This section presents our methodology for physically implementing the astrocyte model using two different approaches: CPU/GPU and FPGA.

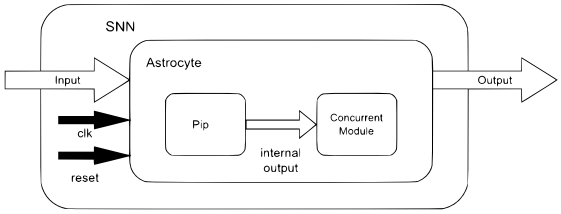


Fig. 3. Block Diagram of Implementation

CPU/GPU Implementations
We utilized Python to execute implementations on the CPU and GPU. The study leveraged the computational prowess of NVIDIA's GeForce RTX 3060 GPU and Intel's Core i9 12900H CPU, both of which are optimized for different tasks, ensuring an efficient execution of our implementations.

FPGA Implementation
Our FPGA implementation was executed on the XCVC1902 FPGA chip, equipped with 400 AI Chips, utilizing the 2021.1 software version of Vivado. Our central module, "Astrocyte", processes a 42-bit input and produces a 42-bit output. The internal operations of the PiP (Place-in-Place) module, which is a

crucial component of this design, are depicted in Fig. 3. The efficiency of our astrocyte-augmented SNN, as presented through metrics, was evident in its low latency and theoretically infinite throughput, emphasizing its computational prowess. The presented metrics stem from an experiment involving an astrocyte-augmented SNN. Our aim was to evaluate how the astrocyte implementation impacts the network's robustness and computational efficiency. Initially, our SNN displayed a fault tolerance of 72.08% without astrocytes, signifying that a single artificially silenced neuron caused the network's output to diverge by this proportion from the original, fault-free state. Such a measure provided an estimate of the SNN's resilience to localized neuronal failures. When astrocytes were incorporated into the SNN, a remarkable reduction in latency was observed; the time required for an entire round of astrocytic updates was essentially zero as per the system clock. This extremely low latency indicated an impressive efficiency in the computational implementation. Moreover, this near-zero latency facilitated theoretically infinite throughput, implying instantaneous processing of all neurons in the network, which further emphasized the exceptional computational efficiency of our astrocyte-augmented SNN. The observed new fault tolerance was quantified as 8.96%, highlighting the degree of enhancement in SNN's fault tolerance as a direct result of astrocyte integration. Post astrocyte integration, the SNN demonstrated an improved fault tolerance of 63.11%. The fault tolerance FT of a SNN is conceptually defined as the proportionate deviation of the SNN's output from the original, fault-free state when subject to a fault condition.

1. FT_{initial}: Initial fault tolerance without astrocytes.
2. FT_{astro}: Fault tolerance after integrating astrocytes.
3. ΔFT: Improvement in fault tolerance due to astrocyte integration, given by $\Delta FT = FT_{\text{initial}} - FT_{\text{astro}}$.

The fault tolerance of the SNN, considering the given description, is represented as:

$$FT = \frac{O_{\text{fault}} - O_{\text{original}}}{O_{\text{original}}} \times 100\% \tag{1}$$

Where:

- O_{original} is the output in the original, fault-free state.
- O_{fault} is the output when a fault (like a silenced neuron) is induced.

From our results:

$$FT_{\text{initial}} = 72.08\%$$
$$FT_{\text{astro}} = 8.96\%$$
$$\Delta FT = 63.11\%$$

This confirms the mathematical relationship:

$$\Delta FT = FT_{\text{initial}} - FT_{\text{astro}} \tag{2}$$

The reduced FT_{astro} implies that the network's output deviates less from the fault-free state when a fault condition is induced, indicating enhanced resilience of the SNN upon integrating astrocytes.

4.4 Adaptive Model Creation with Dynamic Function eXchange Technology

We utilize Dynamic Function Exchange (DFX) technology for an adaptable model construction. Central to this approach is on-the-fly hardware reconfiguration, allowing computational functions to map onto hardware according to emerging demands. Initially, the model engages in "Training & Predicting" using historical data, recognizing patterns for adaptation. It then proceeds to "Adjusting Hyperparameters" for performance refinement. Ultimately, in the "Execute DFX" phase, as illustrated in Fig. 4, DFX's real-time hardware reprogramming facilitates model functionality adjustments according to network state changes, optimizing computational resource allocation. This not only enhances adaptability to SNN variability but also fosters energy efficiency, pivotal for high-demand machine learning tasks. In essence, our DFX-integrated model offers enhanced performance and adaptability in astrocyte-based neuronal network implementations.

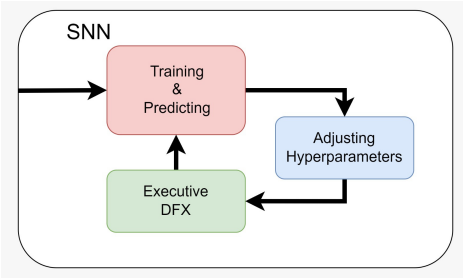


Fig. 4. DFX Diagram

4.5 Quantitative Analysis of the Hardware Accelerator

For computational tasks, especially in real-time scenarios, metrics like throughput and latency are vital. Throughput gauges the system's capability to handle data processing, whereas latency measures the delay before a transfer of data begins. These metrics play a pivotal role in understanding and optimizing the performance of our system.

$$\text{Throughput} = \frac{\text{No. of MACs}}{\text{Operational Latency}} \tag{3}$$

The above equation delineates the throughput as a function of the number of Multiply-Accumulate (MAC) operations over the operational latency. The count

of MAC operations is derived from specialized neural network libraries [12]. On the other hand, the operational latency, which is synonymous with simulation time in this context, predominantly emerges from the inherent characteristics and constraints of the underlying hardware architecture. This is mathematically captured by:

$$\text{Operational Latency} = \frac{\text{Time for Inference}}{\text{Dataset Loader Iteration}} \tag{4}$$

This equation emphasizes the interdependence between the time taken for model inference and the iterations dictated by the dataset loader.

Table 1. Resource utilization summary

VC1902 Versal			
Resource	**Utilization**	**Available**	**% Utilization**
LUT	900	899,840	0.10%
FF	100	75,000	0.13%
BRAM	0	1,000	0%
IO	86	770	11.17%
AI Engine	0	400	0%
DSP	0	1,968	0%

Our FPGA implementation's efficiency can be further understood through the resource utilization summary provided in Table 1. The low percentages in the utilization column indicate efficient use of resources. However, there remains an opportunity to further leverage these resources for complex tasks or to enhance performance.

5 Results

Table 2. Comparison between CPU, GPU, and FPGA

	i9 12900H	**RTX 3060**	**VCK190**
Vendor	Intel	NVIDIA	AMD-Xilinx
Tech (nm)	10	8	7
Freq (MHz)	5200	1320	100
MACs (G)	0.269	0.269	0.269
Latency (ms)	84	11.6	4.6
Power (W)	27	68	2
Throughput (GOP/s)	3.2	24.5	58.5
Efficiency (GOP/s/W)	0.11	0.36	29.2

Table 2 covers metrics such as manufacturing technology, operating frequency, and power consumption. Notably, the parallel execution of GPUs and FPGAs often surpasses CPUs in efficiency, even at lower frequencies. This is demonstrated by

the Xilinx FPGA's 4.6 ms latency and its mere 2 Watts consumption. Emphasizing energy efficiency, the table indicates FPGA's throughput of 58.5 GOP/s and an energy efficiency of 29.2 GOP/s/W, accentuating FPGAs' proficiency for energy-sensitive applications. This underscores the unique attributes and potential applications of each technology. In Table 3, our FPGA-based astrocyte modeling on the advanced Xilinx VCK-190 chip is compared with prior works such as [7], [6], and [5]. Operating at a standard 100 MHz, our model encompasses 680 neurons and 69,888 synapses, outstripping other models in complexity. Correspondingly, our model demonstrates robustness with a fault tolerance rate of 9.96%, on par with [5], and a resilience improvement of 63.11%. Despite a 2W power demand, higher than certain FPGA models, our implementation's extensive neuronal and synaptic counts justify this. This consumption reflects our model's commendable energy efficiency amidst heightened complexity.

Table 3. Comparisons with previous implementations.

	[17]	[7]	[6]	[5]	**Our**
Platform	CPU	FPGA Virtex-5	FPGA Artix-7	FPGA VCU-128	**FPGA VCK-190**
Clock	3.1 GHz	100 MHz	100 MHz	100 MHz	**100 MHz**
Neurons	2	14	-	336	**680**
Synapses	1	100	-	17,408	**69,888**
Fault Tolerance Rate	30%	30%	-	39%	**8.96%**
Resilience Improvement	12.5%	70%	80%	51.6%	**63.11%**
Power	-	1.37 W	0.33 W	0.538 W	**2 W**

6 Conclusions

This work has presented a novel astrocyte-augmented spiking neural network model implemented on CPU/GPU and FPGA platforms. The inclusion of astrocytes has shown significant improvements in the network's fault tolerance, demonstrating the potential benefits of astrocyte integration in artificial neural networks. Additionally, the use of FPGA hardware for this model leverages the advantages of parallel computation and on-the-fly hardware reconfiguration offered by DFX technology. The comparison with different computational architectures and previous works highlighted the strengths of our approach in terms of computational efficiency and network robustness. Future research in this direction could yield more sophisticated and efficient neuromorphic systems, thus paving the way for advanced applications in diverse areas such as robotics, bioinformatics, and cognitive computing.

References

1. Covelo, A., Araque, A.: Lateral regulation of synaptic transmission by astrocytes. Neuroscience **323**, 62–66 (2016)

2. Haghiri, S., Ahmadi, A.: Digital fpga implementation of spontaneous astrocyte signalling. international journal of Circuit Theory and Applications **48**(5), 709–723 (2020)
3. Inadagbo, K., Arig, B., Alici, N., Isik, M.: Exploiting fpga capabilities for accelerated biomedical computing. arXiv preprint arXiv:2307.07914 (2023)
4. Isik, M., Oldland, M., Zhou, L.: An energy-efficient reconfigurable autoencoder implementation on fpga. arXiv preprint arXiv:2301.07050 (2023)
5. Isik, M., Paul, A., Varshika, M.L., Das, A.: A design methodology for fault-tolerant computing using astrocyte neural networks. In: Proceedings of the 19th ACM International Conference on Computing Frontiers. pp. 169–172 (2022)
6. Johnson, A.P., Halliday, D.M., Millard, A.G., Tyrrell, A.M., Timmis, J., Liu, J., Harkin, J., McDaid, L., Karim, S.: An fpga-based hardware-efficient fault-tolerant astrocyte-neuron network. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1–8. IEEE (2016)
7. Johnson, A.P., Liu, J., Millard, A.G., Karim, S., Tyrrell, A.M., Harkin, J., Timmis, J., McDaid, L., Halliday, D.M.: Homeostatic fault tolerance in spiking neural networks utilizing dynamic partial reconfiguration of fpgas. In: 2017 International Conference on Field Programmable Technology (ICFPT). pp. 195–198. IEEE (2017)
8. Karim, S., Harkin, J., McDaid, L., Gardiner, B., Liu, J., Halliday, D.M., Tyrrell, A.M., Timmis, J., Millard, A., Johnson, A.: Assessing self-repair on fpgas with biologically realistic astrocyte-neuron networks. In: 2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI). pp. 421–426. IEEE (2017)
9. Kumar, S.R., Singhal, S.: Implementation of neuron astrocyte interaction dynamics. In: 2023 IEEE 8th International Conference for Convergence in Technology (I2CT). pp. 1–6. IEEE (2023)
10. Mueggler, E., Rebecq, H., Gallego, G., Delbruck, T., Scaramuzza, D.: The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. The International Journal of Robotics Research **36**(2), 142–149 (2017)
11. Pfeiffer, M., Pfeil, T.: Deep learning with spiking neurons: Opportunities and challenges. Frontiers in neuroscience **12**, 774 (2018)
12. PyTorch-OpCounter. `https://pypi.org/project/thop/`, accessed: 2023-08-26
13. Santello, M., Toni, N., Volterra, A.: Astrocyte function from information processing to cognition and cognitive impairment. Nature neuroscience **22**(2), 154–166 (2019)
14. Tavanaei, A., Ghodrati, M., Kheradpisheh, S.R., Masquelier, T., Maida, A.: Deep learning in spiking neural networks. Neural networks **111**, 47–63 (2019)
15. Venkataramani, S., Srinivasan, V., Wang, W., Sen, S., Zhang, J., Agrawal, A., Kar, M., Jain, S., Mannari, A., Tran, H., et al.: Rapid: Ai accelerator for ultra-low precision training and inference. In: 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA). pp. 153–166. IEEE (2021)
16. Volterra, A., Meldolesi, J.: Astrocytes, from brain glue to communication elements: the revolution continues. Nature Reviews Neuroscience **6**(8), 626–640 (2005)
17. Wei, X., Li, C., Lu, M., Yi, G., Wang, J.: A novel astrocyte-mediated self-repairing cpg neural network. In: 2019 Chinese Control Conference (CCC). pp. 4872–4877. IEEE (2019)
18. Zhang, S., Ji, W., Li, X., Huang, K., Yin, R.: Precise failure location and protection mechanism in long-reach passive optical network. Journal of Lightwave Technology **34**(22), 5175–5182 (2016)