Implementation of a Low Power IC for Neuromorphic Computing

ESE 440 Senior Design

Spencer Wu (EE)
Arhaam Hossain (EE)
Huabin Wu (CE)
Ryan Lin (CE)

Faculty Advisor: Milutin Stanacevic

College of Engineering and Applied Sciences
Electrical & Computer Engineering Department
Stony Brook University

I. Project Objective

The objective of this project is to design and implement a low-power integrated circuit (IC) for neuromorphic computing to detect spoken words in noisy environments. Resistive random-access memory (ReRAM) will be used as memory cells and a leaky integrate-and-fire (LIF) spiking neural network will be applied to process audio signals into spikes. The final deliverables will be a fully designed IC including a physical layout which will be ready for fabrication and deployment in real-world applications.

II. Background Research

A. Introduction to Neuromorphic Computing

von Neumann architectures are defined as architectures consisting of a CPU, memory, and input and output devices. These architectures are commonly found in everyday devices such as computers, smartphones, and servers. However, despite its usage, von Neumann architectures face limitations due to the 'memory wall' [1]. This restriction, known as the von Neumann bottleneck, is defined as the physical separation between the CPU and memory bus. This separation means that data must be constantly transferred between these components, leading to high energy consumption and delay. Due to this, neuromorphic computing, a computing approach modeled after the human brain, has been discussed as a promising alternative [1, 2].

The fundamental difference between neuromorphic systems and traditional ones is the usage of artificial neurons and synapses, much like the human brain, to process information with high efficiency. The approach allows asynchronous and event-drive computation, making it more energy efficient than systems that are synchronous and clock-driven [1]. Spiking Neural Networks (SNNs), often considered the third generation of artificial neural networks, play as a core component of neuromorphic computing by offering a more biologically plausible model through its representation and transmission of information as 'spikes', mimicking the action potentials observed in biological neurons [1, 2]. As such, neurons only 'fire' when necessary, resulting in sparse activity and lower power consumption compared to traditional artificial neural networks (ANNs) [2]. This behavior makes it excel at handling complex, dynamic, and noisy data seen in audio processing and speech recognition [2, 3].

B. Hardware Architectures and Implementations

The potential neuromorphic computing has revolves around the development of specific hardware architectures that can perform 'brain-like' computations efficiently. The focal point in these architectures is the joined location of memory and processing units, which is not seen in von Neumann architectures and effectively nullifies the 'memory wall' bottleneck [1, 4]. This approach alongside the event-driven and asynchronous operation allows it to achieve low-power consumption characteristics.

Many companies who are interested in the capability of neuromorphic computing are currently working on state-of-the-art chips. IBM's chip, TrueNorth, for example, is a digital CMOS chip containing one million spiking neurons and 256 million synapses [1, 4]. It employs a lot of the design principles of neuromorphic computing such as a simplified neuron model and binary synapses to achieve energy efficiency. It's power density is approximately 20 mW/cm^2, which is significantly lower than typical CPUs that are around 50-100 W/cm^2) due to the sparse activity, event-driven computing, and asynchronous communication that comes with the properties of neuromorphic computing [1, 4]. However, it should be noted that its synapses are non-plastic during runtime, which limits its on-chip learning capabilities. On the other hand, Intel's chip, Loihi, was developed using 14-nm process technology, integrating 130,000 neurons and 130 million synapses which supports in-hardware adaptation for variables based on local information [1, 4]. It's asynchronous network-on-chip allows for complex communication and advanced learning rules [1,4]. It is critical that event-driven, asynchronous computation, and sparse activity are considered to develop low-power neuromorphic ICs.

Besides fully digital implementations, mixed-signal and analog approaches also contribute to low-power designs. Analog circuits operating in subthreshold regions can provide superior energy efficiency and lower noise energy, though they may suffer from lack of uniformity due to device mismatch [4]. Mixed-signals combine the power efficiency of analog neuron circuits with the reliability of digital synaptic weight storage. The BrainScaleS System, for instance, is a mixed-signal platform that achieves significant speedup factors for spiking network emulations due to its combination of analog neuron circuits and digital communication [4]. Additionally, the NeuroGrid/Braindrop project also utilizes mixed-signal circuits to model continuous-time neural processing elements, leveraging the variability of analog circuits for computation [4]. These hardware strategies try to balance biological realism, computational efficiency, and power consumption for successful neuromorphic ICs. The event-driven nature of these spikes inherently leads to efficient architectures with joined memory and processing units, which increases parallelism and reduces energy usage [4].

C. Spiking Neural Network Fundamentals

The behavior of individual neurons within SNNs is typically modeled using simplified mathematical frameworks. The Leaky Integrate-and-Fire (LIF) model is commonly adopted due to its balance of biological characteristics and computational efficiency [1, 2, 4]. In the model, a neuron integrates incoming synaptic currents, and its membrane potential rises until is crosses a predefined threshold, at which point, it will emit a spike and reset its potential [1, 4]. More complex models such as the Izhikevich model offer higher ranges of neuronal behaviors while maintaining computational traceability, making them suitable for simulating large-scale neural networks [4]. The choice of neuron model is often dependent on the type of application, with simpler models being preferred for hardware implementation to optimize power consumption and area [4].

Fundamentally, SNNs require effective mechanisms to encode analog input signals into spike trains. The two main methods are rate coding and temporal coding [2, 5]. Rate coding represents information by the firing frequency of neurons, where more intense signals results in more frequent spikes. While easy to implement, rate coding is often inefficient for energy and latency, especially for larger values [1, 2]. Tem-

poral coding, on the other hand, encodes information in the precise timing of individual spikes or the time difference between spikes such as Time-to-First-Spike (TTFS) or Temporal Switch Coding [1, 2, 5]. Temporal coding methods can convey more information with fewer spikes, leading to higher energy efficiency with a lower communication workload, making it more favorable for low-power ICs [1, 2, 5]. The tradeoff of it is that temporal coding methods are often harder to implement due to its reliance on precise spike timings. Event-based sensors, which naturally produce spike trains as a result of changes in the stimuli, are inherently compatible with SNNs and can eliminate the need for complex spike encoding stages, further reducing power consumption [2]. A comparative study highlighted TTFS coding as optimal for computational performance with low hardware overhead [4]. Additionally, it also demonstrated phase coding's resilience to input noise as well as burst coding's high network compression and robustness to hardware non-idealities [4].

D. Learning Mechanisms in SNNs

Learning in SNNs can be categorized into two approaches: unsupervised and supervised. Spike-Timing-Dependent Plasticity (STDP) is a biologically plausible unsupervised learning rule that modifies synaptic weights based on the relative timing of pre-synaptic and post-synaptic spikes [1, 2, 4, 5]. STDP is effective for shallow networks and low-level feature extraction, and its local nature makes it hardware-friendly for low-power learning circuit components [2, 4, 5]. For more complex tasks, supervised learning methods are employed, although it is important to note that they present challenges due to the discontinuous and non-differentiable nature of spiking neurons [2, 3]. Approaches such as ANN-to-SNN conversion, where a pretrained ANN is converted into an SNN for inference, can achieve similar results but often at the cost of higher inference latency [2]. Spike-based backpropagation is also considered, using surrogate gradients to enable end-to-end training. While having lower inference latency, they are more demanding of computational and memory resources [2]. There have been hybrid strategies combining ANN-to-SNN with spike-based backpropagation to balance the latency and training costs [2]. Biologically plausible local learning rules, such as the three-factor learning, updates weights based on local information. This process reduces memory overhead compared to global backpropagation and aligns well with event-driven neuromorphic hardware [2, 4].

E. Neuromorphic Audio Processing

The human auditory system is a marvel of biological engineering, capable of processing complex sounds, localizing sources, and discerning speech even in highly noisy environments [5]. This capability has inspired the development of neuromorphic approaches to audio processing and word detection, aiming to replicate the efficiency and robustness of biological hearing in artificial systems [3, 5]. Traditional audio classification methods often rely on signal processing algorithms and manually crafted features, which can struggle to capture the nuances of sound patterns and perform poor in real-world scenarios, especially ones dynamic or noisy data [3]. Neuromorphic computing offers a promising alternative by providing processing capabilities that are well-suited for handling such complexities [3].

Neuromorphic systems for audio processing begin with converting analog audio signals into spike trains, a process known as input encoding. Temporal coding, particularly Time-to-First-Spike (TTFS) and Temporal Switch Coding, is often favored for its energy efficiency and ability to convey significant information with fewer spikes, which is necessary for low-power IC implementations [1, 2]. Event-based sensors, such as neuromorphic cochleae, are well-suited for this task due to their nature of producing spike trains in response to changes in acoustic stimuli, making it have high temporal resolution, wide dynamic range, and low-power consumption [2, 5]. These sensors can incorporate adaptive signal preprocessing, including frequency decomposition and nonlinear amplification, mimicking the cochlea's function to improve signal detection in noisy conditions [5].

Spiking Neural Networks (SNNs) are the most common in neuromorphic audio processing, offering a biologically plausible framework for sound recognition. Research has shown that SNNs can achieve superior energy efficiency and processing speed for voice signal processing and time-sensitive sound recognition tasks compared to conventional deep learning methods [2]. The Tianjic chip, for instance, utilizes SNNs for voice recognition in experimental setups [1]. Various SNN architectures have been explored for tasks such as keyword spotting, speaker identification, and speech analysis [5]. Earlier SNN models focused on simulating spiking activities, with recent advancements having highlighted their learning capabilities and real-time data processing abilities, making them suitable for mobile and wearable technologies [2]. The integration of temporal dynamics within SNNs makes them well-suited for handling time-series data like sound, where the precise timing of events carries significant information [2]. The development of adaptive microelectromechanical system (MEMS)-based cochleae with integrated feedback further pushes the potential of neuromorphic acoustic sensing. These systems can dynamically tune their sensing and processing properties in real-time based on acoustic signal characteristics, with dynamic switching between linear and nonlinear characteristics improving signal detection in noisy conditions [5].

F. Noise Robustness in Neuromorphic Systems

One of the most compelling advantages of neuromorphic computing for applications like word detection is its inherent robustness to noise, a characteristic directly inspired by the resilience of biological neural systems [1, 3]. Traditional speech processing systems often struggle in low signal-to-noise ratio (SNR) conditions, where interfering noise and reverberation can degrade performance significantly [5]. Neuromorphic systems, by mimicking the brain's ability to process sensory information in real-time and adapt to changing conditions, allow for more efficient and robust sound classification algorithms [3].

The event-driven nature of SNNs contributes significantly to their noise robustness. By processing information only when a spike occurs, SNNs inherently filter out irrelevant signals that do not cross a neuronal firing threshold, effec-

tively reducing the impact of background noise [2, 4]. This behavior of neuromorphic designs comes from the key inspiration of the brain's ability to process sensory information in real-time and adapt to dynamic or noisy data [3]. Digital implementations of neuromorphic computing generally offer better noise resilience compared to their analog counterparts, although analog circuits can be designed to exploit variability for certain computational strategies [1, 4]. Recurrent neural networks (RNNs), a common architecture in both biological and artificial neural systems, are also implemented in neuromorphic designs to stabilize signals and suppress noise, further enhancing robustness [1].

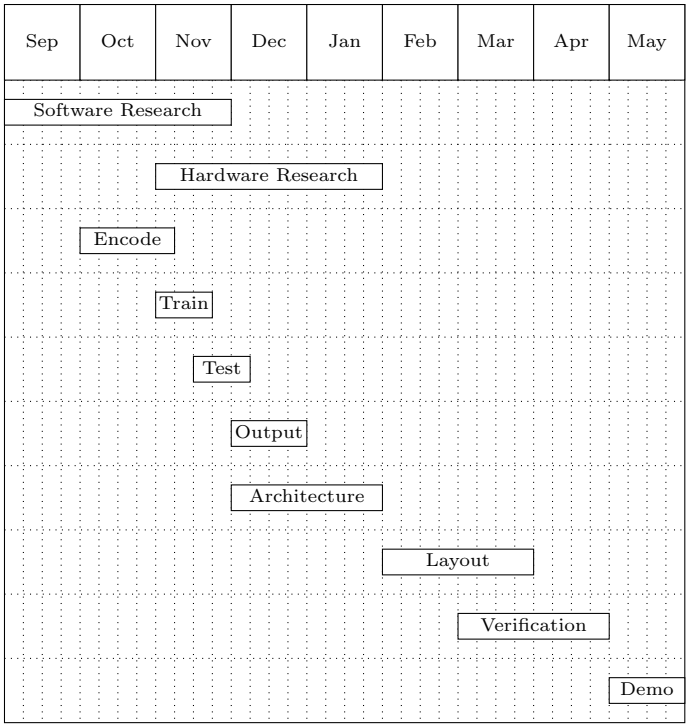
This adaptive capability, coupled with the inherent event-driven processing of SNNs, allows neuromorphic systems to maintain high performance even in challenging acoustic environments, making them particularly suitable for reliable word detection in noisy settings.

III. Proposed Solution

This project will develop a neuromorphic architecture focused on minimizing energy usage and latency while maintaining high accuracy in noisy conditions. The solution centers on a ReRAM-based computing system, where both computation and weight storage occur within the memory array, greatly reducing data transfer bottlenecks common in von Neumann architectures. A SNN is chosen to process audio spike patterns efficiently, emphasizing robustness to background noise. Additionally, the system will incorporate mechanisms for continual learning, enabling adaptation to new users or environments without the requirement of external retraining or cloud-based resources. This integrated approach ensures that the IC is optimized for real-time speech recognition while maintaining its low-power consumption. The Microsoft Scalable Noisy Speech Dataset (MS-SNSD) will be utilized for testing, training, and validation. Infinite memory will be assumed for the input of the neural network. It will then be scaled down as much as possible to the memory usage of the neural network. Testing will be done by comparing the input and the expected output. A singular ReRAM cell is first verified before scaling it for all cells. The current budget for this project is $920, but no spending is expected until the design reaches the fabrication stage.

IV. Deliverables

The primary deliverable of this project is a fully designed neuromorphic IC capable of detecting spoken words in noisy environments with low-power consumption. This includes a detailed circuit design and physical layout ready for fabrication. Additional deliverables will include simulation results that demonstrate speech recognition accuracy and noise robustness, a verification report showing performance metrics meeting the design specifications, and documentation of the design methodology to support future iterations or scaling to other applications. A Gantt chart illustrating the timeline for this project and its goals over a 9 month period from September to May, as well as a list of the primary responsibilities of each member is shown below.



Spencer Wu (Software & Hardware)
- Coordinate software research and support hardware research from initial stages.
- Oversee design parameters and architecture implementation to ensure software-hardware compatibility.
- Assist in verification and demo of software and hardware components.

Arhaam Hossain (Hardware)
- Conduct hardware research on ReRAM-based memory arrays and low-power circuit design.
- Manage IC architecture and layout including schematic capture and physical design.
- Verify hardware performance and support demo preparation.

Huabin Wu (Software)
- Develop and optimize spike encoding and SNN training pipelines.
- Handle testing and output analysis to validate software models.
- Support demo preparation by deploying trained SNN models.

Ryan Lin (Software)
- Collaborate on encoding, training, and preprocessing of input data.
- Implement testing frameworks and analyze output for performance targets.
- Provide ongoing support for system integration, troubleshooting, and performance.

V. Potential Obstacles and Mitigation

Several challenges could arise during this project. The variability and limited precision of ReRAM devices may affect weight storage and learning accuracy, which is expected to be mitigated through error-tolerant circuit designs, redundancy, and calibration techniques. The implementation of continual learning risks new learning degrading previously stored

patterns, which will be addressed using controlled learning rates, weight normalization, or adaptive update rules. Meeting strict power, area, and latency constraints while maintaining recognition performance will require careful circuit-level optimization, low-power neuron designs, and hierarchical memory architectures. Lastly, ensuring robust operation under noisy conditions can be managed through extensive simulation, noise-aware training of the SNN, and hardware-in-the-loop testing before fabrication.

VI. Impact Considerations

The core innovation of developing robust, low-power, on-device speech processing is the focus of several societal challenges. Its potential is most evident in advancing equity and accessibility, as it can empower hearing-impaired individuals and workers in high-noise environments through reliable assistive technology. This push towards autonomous development is inherently sustainable, offering a pathway to reduce the energy footprint of pervasive computing by eliminating the constant data transmission required by cloud-dependent models. Furthermore, this architectural shift fundamentally reconfigures data sovereignty and privacy, ensuring sensitive voice data is processed locally. This convergence of capabilities is foundational for enabling transformative and ethically grounded applications in domains like healthcare, smart cities, and next-generation assistive devices, positioning the technology as a key enabler for a more inclusive, sustainable, and secure future.

VII. Project Team

The Neuricell team is composed of four senior electrical and computer engineering students, each bringing unique expertise to the project. Our areas of expertise are described below.

Spencer Wu: Specializes in PCB design and power electronics, with additional expertise in AI/ML applications for intelligent system modeling and optimization.
Arhaam Hossain: Specializes in VLSI and circuit design, with a background in power electronics and system-level hardware integration, focusing on developing efficient hardware solutions.
Huabin Wu: Concentrates on embedded and digital system design, with skills in hardware–software integration and digital logic implementation for high performance computing applications.
Ryan Lin: Specializes in the development of embedded systems, web automation tools, and IoT device design, with experience in the prototyping of connected systems and optimizing automation workflows.

VIII. References

[1] A. Shrestha, H. Fang, Z. Mei, D. P. Rider, Q. Wu, and Q. Qiu, "A Survey on Neuromorphic Computing: Models and Hardware," *IEEE Circuits and Systems Magazine*, vol. 22, no. 2, pp. 6-35, 2022.
[2] B. Rajendran, A. Sebastian, M. Schmuker, N. Srinivasa, and E. Eleftheriou, "Low-Power Neuromorphic Hardware for Signal Processing Applications," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 97-110, Nov. 2019.
[3] N. Rathi, I. Chakraborty, A. Kosta, A. Sengupta, A. Ankit, P. Panda, and K. Roy, "Exploring Neuromorphic Computing Based on Spiking Neural Networks: Algorithms to Hardware," *ACM Computing Surveys*, vol. 55, no. 12, Art. 3571155, Dec. 2023.
[4] W. Guo, M. E. Fouda, A. M. Eltawil, and K. N. Salama, "Neural Coding in Spiking Neural Networks: A Comparative Study for Robust Neuromorphic Systems," Frontiers in Neuroscience, vol. 15, Art. 638474, Apr. 2021.
[5] B. Meftah, O. Le´zoray, S. Chaturvedi, A. A. Khurshid, and A. Benyettou, "Image Processing with Spiking Neuron Networks," in Artificial Intelligence, Evolutionary Computation and Metaheuristics. Berlin, Heidelberg: Springer, 2013, pp. 525–544.

IX. Appendix A

Faculty Advisor Meetings: Wednesdays 10:30 - 11:00 AM
Team Meetings: Fridays 3:00 - 7:00 PM

X. Appendix B

The following is a list of resumes from each team member.