Technical Report

# Design and Implementation of a Low-Power Mixed-Signal Spiking Neuron

**Linshen Liu**
**Andreas Gerstlauer**

**UT-CERC-25-02**

**May 1, 2025**

**Computer Engineering Research Center**
**Chandra Family Department of Electrical and Computer Engineering**
**The University of Texas at Austin**

The University of Texas at Austin
Chandra Department of Electrical and Computer Engineering
*Cockrell School of Engineering*

# Design and Implementation of a Low-Power Mixed-Signal Spiking Neuron

Linshen Liu, Andreas Gerstlauer

Electrical and Computer Engineering
The University of Texas at Austin

**Abstract**

This work presents a low-power neuromorphic neuron circuit designed as a compact and tunable Leaky Integrate-and-Fire module, compatible with any mixed-signal architecture that incorporates a digital backend. The proposed design comprises three key functional blocks: a spiking input module that converts voltage pulses into charge currents, a voltage leakage module that implements tunable exponential decay, and a digital spike generation module that detects threshold crossings and emits output spikes. This design is based on a TSMC 65 nm CMOS process and evaluated under a 10 MHz input spike frequency. The neuron module exhibits reliable spiking behavior, strong noise robustness, and programmable dynamics. The measured energy consumption per spike is 10.12 pJ, with a static power consumption of 26.12 uW. In contrast to prior analog implementations, this work uniquely supports both adaptive dynamics and on-chip learning within a compact architecture. Unlike existing designs that often sacrifice learning capability or area efficiency, this design achieves all three: adaptation, plasticity, and area-optimized integration.

These results highlight the potential of circuit-level neuromorphic design for constructing dense, scalable, and energy-efficient spiking neural network (SNN) accelerators tailored for edge computing and real-time signal processing. The modularity of the proposed neuron block further supports its integration into large-scale mixed-signal neuromorphic systems.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1: Introduction

The structure of traditional digital processors fundamentally differs from the organization of the human brain. While conventional processors are typically based on the von Neumann architecture, which separates logic and memory and is driven by a global clock, the brain performs computation through massively parallel and asynchronous interactions between neurons and synapses [1]. This organization allows the brain to achieve remarkable energy efficiency, operating at approximately 20 W—which far surpasses the energy efficiency of current general-purpose processors [2]. This striking contrast has motivated extensive research into brain-inspired computing systems that seek to replicate this efficiency and architectural elegance in silicon-based systems.

Motivated by this contrast, neuromorphic systems are designed to move away from clock-driven, centralized computing paradigms and instead adopt event-driven, locally connected architectures that support the co-location of memory and computation. Rather than executing instruction pipelines at uniform, clock-driven intervals, neuromorphic systems operate by propagating discrete spike events only in response to meaningful input activity, allowing inactive regions to remain idle and consume minimal dynamic power. This leads to significantly reduced energy consumption and enables scalable, real-time computation for processing sparse, spatiotemporally structured data. A representative example of such a system is BrainScaleS [3]. It implements analog neuron dynamics at the circuit level, coupled with an event-driven digital communication backend that ensures scalability—allowing additional analog cores to be integrated seamlessly into the architecture.

As the fundamental computational unit of neuromorphic systems, neuromorphic circuits integrate memory and computation at the circuit level, eliminating the need for energy-intensive data movement between physically separated logic and

memory units. These architectures are often implemented using asynchronous, analog, or mixed-signal techniques that emulate the dynamics of biological neurons and synapses. For example, membrane potentials are typically represented using capacitor voltages, synaptic weights by current mirrors or programmable conductances, and spike events by brief switching transients. This circuit-centric approach offers substantial benefits in latency, energy efficiency, and scalability. Event-driven processing enables the circuit to remain idle until relevant stimuli occur, significantly reducing power consumption. Furthermore, the local storage of neuronal state variables allows for distributed computation and near-memory processing, both of which are essential for scalable neuromorphic integration. These advantages make neuromorphic circuits particularly well-suited for deployment in resource-constrained environments, including mobile edge devices, autonomous robots, biomedical implants, and low-latency signal processing applications—contexts where traditional digital architectures often fall short in terms of responsiveness and energy efficiency.

Among existing neuromorphic platforms, only a few offer both analog neuron implementations and flexible system-level programmability. While many systems focus on digital emulation or fixed-function analog designs, the BrainScaleS neuromorphic computing architecture [3] that tightly integrates tunable analog neurons with general-purpose digital infrastructure remains rare. This architecture offers a flexible hardware–software co-design platform. Built on the Power Instruction Set Architecture (PowerISA) [4] and supported by a Linux-based BrainScaleS operating system [5], it enables tight coupling of analog neuron circuits with digital control infrastructure, facilitating real-time, large-scale spiking neural network emulation. While BrainScaleS offers a unique combination of tunable analog neurons and a digital backend, this flexibility comes at a cost: the large number of configurable parameters results in substantial circuit overhead, reducing area efficiency and limiting the scalability of analog core arrays.

To address these limitations, we propose a low-power neuromorphic neuron grid accelerator circuit designed to serve as a modular, area-efficient building block

fully compatible with the BrainScaleS ecosystem. The proposed design exhibits high tunability, robust parameter mapping, and strong resilience to noise, while maintaining a compact physical footprint. By exploiting the time-domain coding characteristics of the Adaptive Exponential neurons model (AdEx) [6], a specialized and more sensitive form of the classical LIF model, capable of capturing rich spiking dynamics through exponential membrane potential evolution and adaptive feedback, the circuit addresses common analog scaling challenges, including bit-depth constraints and susceptibility to noise. The resulting neuron unit is compact, energy-efficient, and reliable, rendering it well-suited for the construction of dense, cost-effective, analog-domain SNN accelerators.

The remainder of this report is organized as follows. Section 2 introduces the computational model and the rationale for adopting a circuit-level AdEx implementation. Section 3 presents the circuit architecture and simulation results. Section 4 concludes the work and discusses potential future directions.

# Chapter 2: Related Work

**Analog Neuromorphic Circuits.** Recent advances in neuromorphic computing have explored a spectrum of hardware implementations, ranging from fully digital logic to hybrid and fully analog designs. Among these, analog neuromorphic circuits stand out for their ability to emulate the continuous-time, event-driven nature of biological neurons with high energy efficiency and compact circuit design. Operating in subthreshold or near-threshold regimes, these systems exploit the intrinsic physics of transistors and capacitors to enable biologically inspired computation. Key building blocks include charge integration units (e.g., capacitors $C_{\text{mem}}$) that accumulate synaptic currents, thresholding circuits (e.g., Schmitt triggers) that generate spikes when membrane voltage crosses a threshold, and leak elements that implement temporal decay. This approach offers distinct advantages in low power consumption, high temporal resolution, and asynchronous operation—making it well-suited for constructing dense and scalable neural substrates.

For example, the BrainScaleS system [3] uses analog circuits to physically integrate membrane potentials, enabling sub-microsecond timescale acceleration of neuronal dynamics. DYNAP-SE [7] employs current-mode analog design to realize adaptive thresholding and tunable synaptic weights using floating-gate transistors. Neurogrid [8], developed at Stanford, uses subthreshold analog CMOS circuits to emulate dendritic integration and conductance-based synaptic dynamics across a massively parallel array of silicon neurons.

The ROLLS neuromorphic processor [9] implements 256 adaptive LIF neurons and 16k plastic synapses using mixed-signal analog-digital design, featuring online STDP learning and ultra-low power operation in the picojoule-per-spike range. The HICANN-DLSchip [10] is implemented in BrainScaleS-2 [4], which employs a highly accelerated analog core with embedded digital calibration support to mitigate device mismatch, enabling real-time closed-loop learning in physical neuron arrays.

**Comparison:** Compared to digital and device-level neuromorphic implementations, analog neuron circuits offer a compelling balance between biological fidelity and hardware efficiency. While digital neuromorphic systems provide programmability, they incur high energy costs due to frequent memory access and clocked control. Device-level implementations, such as those based on memristors, promise high integration density but often suffer from variability and limited temporal precision. In contrast, analog circuits support real-time, continuous-time dynamics with low energy per spike, making them well-suited for scalable, low-power neuromorphic hardware.

**Design Gap and Motivation.** Analog neuromorphic circuits have demonstrated significant advantages in terms of energy efficiency, circuit compactness, and biophysically inspired spiking behavior. Prior designs such as BrainScaleS [3], DYNAP-SE [7], and ROLLS [9] have explored various neuron models, including AdEx [6], adaptive LIF [11], and STDP [12], achieving either adaptive dynamics, on-chip learning, or area efficiency, as detail shows in Table 2.1.

However, few of these implementations simultaneously achieve all three of the following: (1) biophysically meaningful adaptive spiking dynamics (e.g., spike-frequency adaptation, exponential onset), (2) compact and modular circuit design, and (3) compatibility with online learning and large-scale array integration. For instance, while DYNAP-SE supports adaptation and learning, it incurs overhead in area efficiency; conversely, ROLLS is area-efficient but lacks adaptive dynamics. These trade-offs indicate a persistent design gap in current analog neuron circuits.

To address this, our work proposes a compact Adaptive Exponential neuron circuit that integrates adaptation, learning compatibility, and area-efficient design into a single architecture. As shown in Table 2.1, our design is the only one among these analog implementations to achieve all three criteria, providing a promising building block for future large-scale neuromorphic systems.

Table 2.1: Comparison of analog neuromorphic neuron implementations. **Adapt.**: Adaptive Spiking, **Flexibility.**: Neurons Tunable Flexibility, **Area Eff.**: Area Efficiency.

| **System** | **Model** | **Adapt.** | **Learn.** | **Area Eff.** |
|---|---|---|---|---|
| BrainScaleS [3] | AdEx | ✓ | ✗ | ✗ |
| DYNAP-SE [7] | Adapt. LIF | ✓ | ✓ | ✓ |
| Neurogrid [8] | Cond.-based | ✗ | ✗ | ✓ |
| ROLLS [9] | LIF + STDP | ✗ | ✓ | ✓ |
| HICANN-DLS [10] | AdEx (cal.) | ✓ | ✓ | ✗ |
| **This work** | **C-AdEx** | ✓ | ✓ | ✓ |

# Chapter 3: Background

In the field of machine learning and neuromorphic computing, spiking neural networks (SNNs) represent a major milestone. Originally proposed by Maass in 1997 [13], SNNs were the first class of neural networks to model neural computation from an algorithmic perspective, systematically incorporating temporal dynamics and event-driven signaling. This framework significantly improves the energy efficiency of neural processing, bringing it closer to practical hardware-level deployment.

Multiple neuron models and synaptic learning rules have been developed to realize SNNs, among which the most prominent are the leaky integrate-and-fire (LIF) model and spike-timing-dependent plasticity (STDP) [12]. The LIF model, in particular, stands out as the most hardware-friendly and widely adopted due to its structural simplicity and strong biological plausibility. In the following sections, we first outline the motivation behind neuromorphic computing and briefly review the historical development of biologically inspired spiking models. We then detail the Adaptive Exponential (AdEx) neuron model, a biophysically enhanced variant of the Leaky Integrate-and-Fire (LIF) neuron [6], describe how networks of such AdEx-derived units form complete SNNs, and present a comparative analysis between AdEx-based SNNs and traditional artificial neural networks (ANNs) from both architectural and algorithmic perspectives.

## 3.1 Development of Neuromorphic Computation

Since the 1950s, researchers from multiple disciplines have explored concepts that would later contribute to neuromorphic computing. Early theoretical models such as the McCulloch–Pitts neuron [14], the Perceptron [15], and the Hopfield network [16] laid the algorithmic and computational foundations for neural information processing. However, these models were primarily developed in the context of sym-

bolic computation or software simulation, rather than as physical implementations.

A significant shift occurred in 1989, when Carver Mead formally introduced the term "neuromorphic computing" [17] to describe a class of hardware systems that directly emulate the structure and dynamics of biological neural circuits using analog VLSI. Mead's work marked a turning point by bridging biological modeling with silicon-based realizations. In the 21st century, research attention further shifted toward spiking neural networks (SNNs) [18], which use discrete spike events to encode and transmit information, closely mimicking real neural communication. This renewed focus on biologically realistic dynamics led to the emergence of several representative neuromorphic hardware platforms.

Today, neuromorphic computing is propelled by three key application demands. First, the post-Moore era necessitates fundamentally new architectures that enable energy-efficient and cost-effective computation. Second, brain and cognitive sciences require biologically realistic hardware capable of simulating neural dynamics with high temporal and spatial resolution. Third, the proliferation of edge computing and the Internet of Things (IoT) calls for lightweight, high-performance hardware that can operate under stringent power constraints.

In response to these needs, neuromorphic computing has advanced through decades of interdisciplinary progress in neuroscience, learning algorithms, circuit design, and systems integration. Biologically inspired models—such as the LIF neuron [11] and STDP learning rules [12]—have provided a principled foundation for temporal computing and event-based adaptation. SNNs now offer a compelling model for low-power, low-latency computation.

On the algorithmic front, innovations in training and adaptation methods have significantly expanded the applicability of SNNs to a range of real-world tasks such as classification, control, and sensory processing. Notably, surrogate gradient techniques [19, 20] have enabled gradient-based optimization in non-differentiable spiking models, bridging the gap between biological plausibility and machine learning scala-

bility. In addition, local unsupervised learning rules such as Spike-Timing-Dependent Plasticity (STDP) [12] and Hebbian-inspired dynamics have demonstrated success in pattern recognition and temporal feature extraction. Online learning and homeostatic adaptation mechanisms [21] further contribute to the robustness and lifelong learning capability of SNNs, making them promising for edge intelligence and event-driven environments.

On the hardware side, various neuromorphic platforms—from fully digital processors like IBM TrueNorth [22] to analog and mixed-signal implementations—have demonstrated the feasibility of brain-inspired computing architectures. Recent advances in compact neuron and synapse circuits have achieved high fidelity and ultra-low power consumption. In parallel, the software ecosystem has matured significantly. Tool-chains such as PyNN [23], Lava [24], and SNNTorch [25] have lowered development barriers and enabled efficient mapping from high-level SNN models to hardware platforms.

Beyond algorithmic and circuit-level advances, system-level support for neuromorphic computing has also grown more favorable. Modern system-on-chip (SoC) platforms—particularly those based on the ARM architecture—now support highly customizable compute pipelines [26]. Their modular and energy-efficient nature makes them suited to hosting neuromorphic accelerators alongside general-purpose processors. Moreover, the widespread adoption of the Linux kernel and its extensible device driver framework enables seamless integration of non-standard analog and mixed-signal components. This tight hardware-software coupling—once a critical bottleneck—has become an enabler, allowing neuromorphic circuits to operate within heterogeneous compute stacks while maintaining compatibility with user-space interfaces and system-level resource management [27].

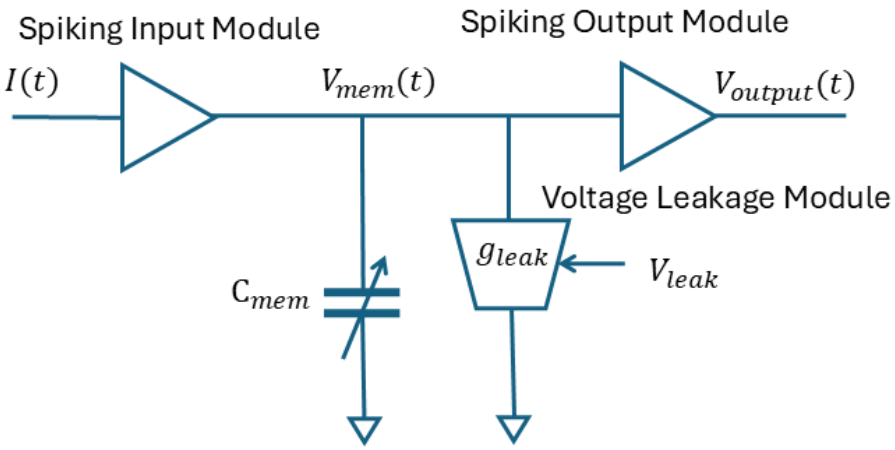



Figure 3.1: AdEx neuron model structure.

## 3.2 Adaptive Exponential Neuron Model

**Adaptive Exponential Integrate-and-Fire Neuron Circuit:** The *Adaptive Exponential* (AdEx) neuron [6] is an advanced extension of the classical *Leaky Integrate-and-Fire* (LIF) model. It incorporates biologically inspired features such as an exponential term for sharp spike initiation and a spike-triggered adaptation current to emulate neuronal fatigue. These additions allow the AdEx model to exhibit a broader repertoire of spiking dynamics compared to standard LIF models, including regular spiking, bursting, and spike-frequency adaptation. Similar to LIF neurons, the AdEx model benefits from time-domain signal encoding, which provides inherent robustness to analog noise and enables energy-efficient processing. These properties make AdEx particularly suitable for analog circuit implementations. Based on this model, we design an AdEx circuit module that emulates both the exponential membrane voltage dynamics and the adaptive current feedback. Figure 3.1 shows the overview of AdEX neuron model. Table 3.1 presents the conceptual correspondence between components in the AdEx circuit module and their biological neuron counterparts. On neuromorphic hardware acceleration platforms, the AdEx circuit module integrates synaptic weights transfer, spike-based encoding, and state memory along

Table 3.1: Physiological analogy between AdEx model parameters and biological neuron properties.

| **AdEx Parameter** | **Biological Interpretation** |
|---|---|
| Membrane potential $V_{\text{mem}}(t)$ | Voltage difference across neuronal membrane |
| Input current $I(t)$ | External stimuli or synaptic input |
| Capacitance $C_{\text{mem}}$ | Membrane capacitance of the neuron |
| Leak conductance $g_{\text{leak}}$ | Ion channel conductivity of the membrane |
| Leak potential $V_{\text{leak}}$ | Resting membrane potential |

the temporal dimension, forming a complete and efficient computational flow.

Analogous to biological neurons, the AdEx circuit model also operates in two primary dynamic states: the subthreshold state (before spike firing) and the spiking state (upon firing). During the subthreshold state, the membrane capacitor $C_{\text{mem}}$ accumulates charge from input synaptic currents $I(t)$ while simultaneously leaking a portion of the charge according to the leaky mechanism, which is controlled by $g_{\text{leak}}$. The dynamics of the membrane potential $V_{\text{mem}}(t)$ under the subthreshold condition can be mathematically described as follows:

$$C_{\text{mem}} \cdot \frac{dV_{\text{mem}}(t)}{dt} = -g_{\text{leak}} \cdot (V_{\text{mem}}(t) - V_{\text{leak}}) + I(t) \tag{3.1}$$

The subthreshold state's dynamics of the AdEx neuron can also be reformulated as:

$$\frac{dV_{\text{mem}}(t)}{dt} = -\frac{1}{\tau_{\text{mem}}} (V_{\text{mem}}(t) - V_{\text{leak}}) + \frac{I(t)}{C_{\text{mem}}} \tag{3.2}$$

where the membrane time constant $\tau_{\text{mem}}$ is defined as:

$$\tau_{\text{mem}} = \frac{C_{\text{mem}}}{g_{\text{leak}}} \tag{3.3}$$

This time constant represents the absolute rate at which the membrane potential decays toward the resting potential in the absence of input, effectively determining the "memory length" of the neuron's voltage response.

**Computation of AdEx Neurons in SNNs.**When drawing an analogy between SNNs and traditional artificial neural networks (ANNs), the AdEx unit can be regarded as a combination of a temporal activation function and a state memory function. The AdEx neuron integrates input spikes over time and emits a spike once the membrane potential surpasses a predefined threshold [28].

In conventional ANNs, the fundamental computation consists of a linear weighted summation function and a following nonlinear activation function, which enables the ANN to extract and process complex, multimodal information. Likewise, in SNNs, computation also comprises both linear and nonlinear components. Specifically, the linear computation is performed through the weighted summation function of input currents in front of the AdEx module, while the nonlinear part is realized via the integration of the temporal activation function and the state memory function realized by the AdEx module. Table 3.2 summarizes the analogy between ANN and SNN computational processes.

In the AdEx model, the weighted summation function takes as input the synaptic signals $S_i(t)$ from the preceding layer and the corresponding synaptic weights $w_{ij}$, which determine the contribution of each input. The resulting output is the synaptic input current $I_j(t)$, which can be formally described as:

$$I_j(t) = \sum_i w_{ij} \cdot S_i(t) \tag{3.4}$$

The value of $S_i(t)$ is either 0 or 1, indicating whether a spike was emitted by the $i$-th presynaptic neuron at time $t$. The membrane potential of the AdEx circuit model $j$ is $V_{\mathrm{j}}(t)$ is formed through a temporal integration process, which accumulates the input currents $I_{\mathrm{j}}(t)$ over time from $t_{\text{last spike generation}}$ to $t$:

$$\tau_m \cdot \frac{dV_j(t)}{dt} = -V_j(t) + R_m \cdot I_j(t) \tag{3.5}$$

In discrete time steps, the membrane potential update of the AdEx neuron $j$ can be expressed as:

$$V_j[t+1] = \alpha \cdot V_j[t] + (1-\alpha) \cdot I_j[t] \tag{3.6}$$

By drawing an analogy with artificial neural networks (ANNs), the computational behavior of spiking neurons can be expressed as:

$$a_j = f(z_j) \tag{3.7}$$

where:

- $z_j$: Linear weighted sum of inputs;
- $f$: Nonlinear activation function (e.g., ReLU, Sigmoid);
- $a_j$: Output activation value.

Table 3.2 summarizes the component-wise mapping between SNN and their ANN counterparts.

Table 3.2: Mapping between core SNN components and their analogous ANN counterparts.

| **SNN Component** | **Analogous ANN Component** |
|---|---|
| $\sum w_{ij} \cdot S_i(t)$ | $\sum w_{ij} \cdot x_i$ (weighted sum) |
| Membrane potential $V_j(t)$ | Pre-activation value $z_j$ |
| Spike generation: $S_j(t) = \begin{cases} 1, & \text{if } V_j(t) \geq V_{\text{th}} \\ 0, & \text{otherwise} \end{cases}$ | Activation function $f(z_j)$ |
| Temporal integration + state memory | Stateless (single-step activation) |

# Chapter 4: Neuron Circuit Implementation

We propose an AdEx neuromorphic circuit that is carefully designed to balance high computational precision, low power consumption, and cost efficiency. In the first stage, a digital-to-analog converter (DAC) module transforms discrete digital inputs into an analog voltage signal and generates corresponding pulse signals $V_{\text{in}}$, as shown by the Spiking Input Voltage Signal in Figure 4.1. As shown in Figure 4.1, these voltage pulses are then converted into input currents $I_{\text{charge}}$ through the Spiking Input Module, which charges the membrane capacitor $V_{\text{mem}}$, thereby resulting in a rising membrane potential $V_{\text{mem}}$. Once $V_{\text{mem}}$ exceeds the leakage threshold $V_{\text{leak}}$, the leakage control module activates, gradually discharging $V_{\text{mem}}$. When consecutive input spikes arrive, $V_{\text{mem}}$ continues to accumulate and eventually surpasses the threshold voltage $V_{\text{threshold}}$, triggering the spike generation module. This module emits an output pulse to downstream neurons and resets $C_{\text{mem}}$ by releasing its stored charge. After a brief refractory period—analogous to the silent phase of a biological neuron—the module is ready to receive new incoming spikes.

To implement the AdEx algorithm described in Chapter 3, our core circuit consists of three primary components: the Spiking Input Module, the Voltage Leakage Module, and the Spike Generation Module. The Spiking Input Module and the Voltage Leakage Module are designed with $V_{\text{mem}}$ charging and leakage interfaces. The Spiking Input Module is designed with a high gain, enabling rapid charge injection into the neuron model. In contrast, the Voltage Leakage Module is designed with a limited gain, allowing for a gradual and controlled discharge of the membrane potential toward the resting state. This asymmetric design allows the AdEx unit to rapidly accumulate charge in response to input spikes and gradually release it over time, faithfully emulating the AdEx behavior. In the following sections, we present a detailed description of each module.

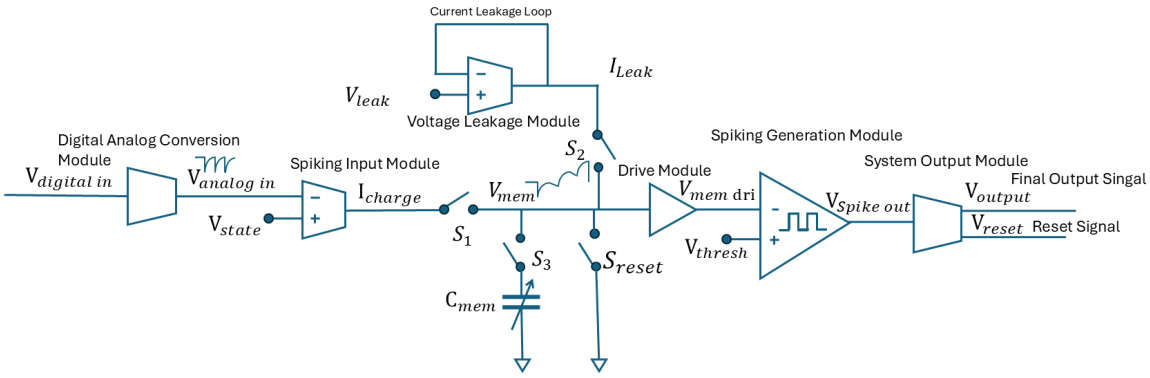



Figure 4.1: Top view of the neuromorphic circuit layout.

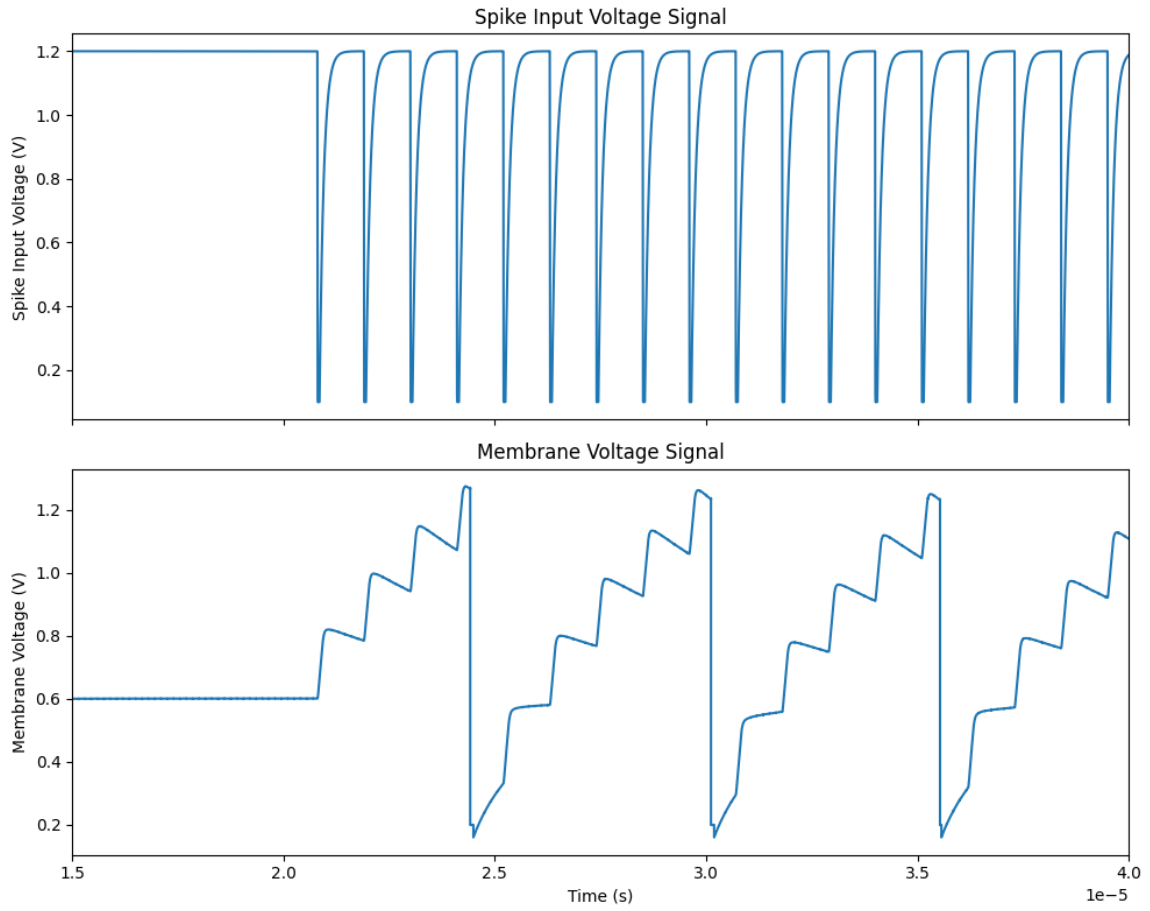



Figure 4.2: The neural membrane voltage value during stimulus response.

Table 4.1: Transistor sizing and operating region for the Spike Input Module and the Voltage Leakage Module.

| **Name** | **W/L** | **Region** |
|---|---|---|
| M1, M2 | $1\,\mu$m / $0.28\,\mu$m | 2 |
| M3, M4, M8, M10 | $5\,\mu$m / $0.6\,\mu$m | 2 |
| M5, M6, M7, M9 | $5\,\mu$m / $0.4\,\mu$m | 2 |
| M11, M12 | $1\,\mu$m / $0.8\,\mu$m | 2 |
| M13 | $0.5\,\mu$m / $0.5\,\mu$m | 1 |
| M14 | $20\,\mu$m / $0.4\,\mu$m | 1 |
| M15 | $3\,\mu$m / $0.28\,\mu$m | 1 |
| M16, M17 | $0.5\,\mu$m / $0.5\,\mu$m | 2 |
| M18 | $0.5\,\mu$m / $0.5\,\mu$m | 3 |
| M19 | $0.5\,\mu$m / $0.45\,\mu$m | 2 |
| M20 | $3\,\mu$m / $0.28\,\mu$m | 2 |

## 4.1 Spiking Input Block

The Spiking Input Block receives voltage pulses from a DAC module and outputs corresponding current pulses. Functionally, this module behaves as a transconductance amplifier, with its transconductance denoted as $G_{\mathrm{m}}$, defined as:

$$G_m = \left( \frac{V_t}{V_{\mathrm{in}}} \cdot \frac{I_{\mathrm{out}}}{V_t} \right) \tag{4.1}$$

Where $V_{\mathrm{t}}$ represents the bias voltage applied at the current mirror node. A schematic of the detailed design is shown in Figure 4.3. As with other voltage-input, current-output amplifiers, the Spiking Input Block is designed to exhibit both high input and high output impedance. As shown in Figure 4.3, high input impedance is achieved by choosing a large $W/L$ ratio for the input transistors $M_1$ and $M_2$. For achieving high output impedance, cascode structures are employed together with small $W/L$ ratios for $M_{10}$ and $M_{18}$. These smaller dimensions not only reduce the common-mode gain but, in conjunction with the cascode configuration, also provide higher bandwidth while maintaining the desired output impedance. The sizing parameters of each MOSFET are listed in Table 4.1.

To reduce the overall chip area, the membrane capacitor $C_{\mathrm{mem}}$ is constrained to

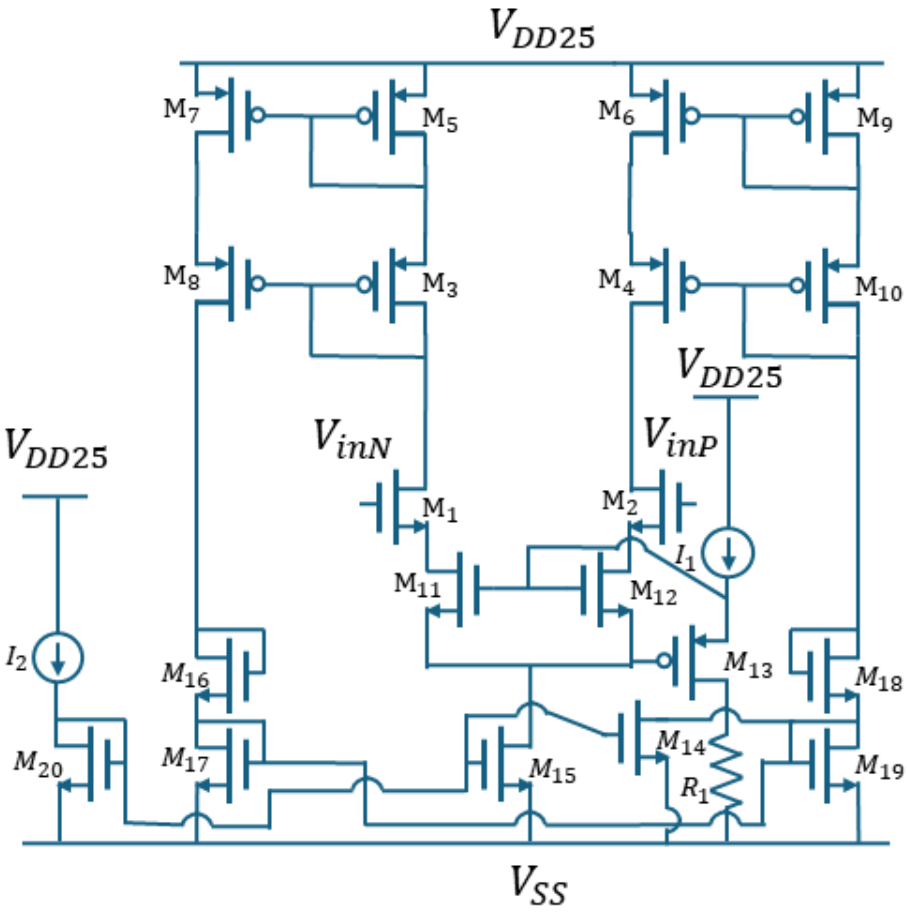



Figure 4.3: The core circuit of the Spiking Input Module and the Voltage Leakage Module.

the picofarad range (0.1–2 pF). This necessitates lower output current and a smaller $G_{\mathrm{m}}$. To achieve this, a differential pair of transistors $M_{11}$ and $M_{12}$ is introduced at the input stage to act as source degeneration resistors, thereby reducing the effective transconductance. To allow finer control over the reduction in transconductance, a clamping MOSFET $M_{13}$ is integrated to adjust the effective transconductance $g_{\mathrm{mr}}$ of transistors $M_{11}$ and $M_{12}$. As a result, the effective transconductance of the input block becomes:

$$G_m = \frac{g_m}{1 + \left(\frac{g_m}{g_{mr}}\right)} \tag{4.2}$$

This expression temporarily omits the effects of channel length modulation for simplicity. Incorporating transistors $M_{11}$ and $M_{12}$ effectively extends the linear operating range of the spiking generator. Additionally, the high $W/L$ ratio of the input differential pair $M_{11}$ and $M_{12}$ further improves the overall linearity of the module. By integrating these design strategies, the Spiking Input Module exhibits high linearity, a wide linearity range, low differential gain $G_{\mathrm{m}}$, low common-mode gain, and high bandwidth. These properties make it well-suited for spiking neural circuit applications. Figure 4.4 shows that the Spiking Input Module exhibits ultra-low common-mode gain. The DC output voltage is consistently maintained at 608.5 mV.

## 4.2 Voltage Leakage Module

The Voltage Leakage Module also adopts a transconductance-based current amplifier as its core, with the detailed implementation shown in Figure 4.3. It shares a similar core circuit topology with the Spiking Input Module. The two modules operate under different static bias conditions. By configuring distinct bias currents $I_1$ in each module, the resulting transconductances differ, thereby creating a controlled disparity between the charging and discharging rates—an essential feature for achieving the desired temporal dynamics.

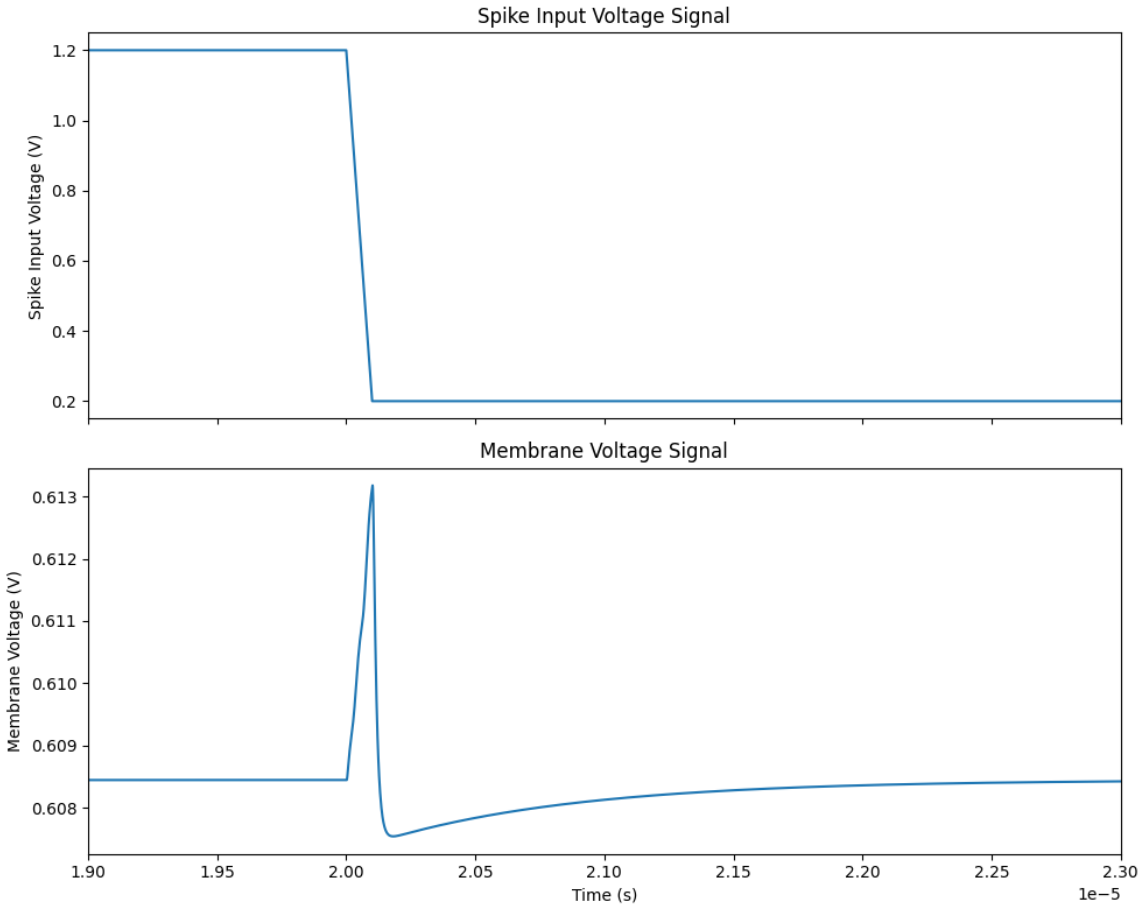



Figure 4.4: Common-mode rejection performance based on membrane voltage response with input common-mode voltage from $0.2\,\mathrm{V} - 1.2\,\mathrm{V}$.

Table 4.2: The behavior of the Voltage Leakage Module under varying input voltages $V_{\text{leak}}$.

| **Vmem** | **Leakage** | **Time** |
|---|---|---|
| 559 | 572 (6 plus) | 6.5 $\mu$s |
| 609 | 587 (5.5 plus) | 5.64 $\mu$s |
| 659 | 602 (5 plus) | 5.5 $\mu$s |

The effective output resistance $1/g_{m_r}$, which shows in Equation 4.2, determined by the transconductance of transistors $M_{11}$ and $M_{12}$ in Figure 4.3, is designed to be approximately an order of magnitude larger in the Voltage Leakage Module than in the Spiking Input Module. This design choice not only supports asymmetrical temporal behavior but also improves linearity, resulting in a smoother $V_{\text{mem}}$ discharge profile.

The time constant of the voltage leakage module is defined as

$$\tau_{\text{mem}} = \frac{C_{\text{mem}}}{G_{\text{mem}}}. \tag{4.3}$$

It can typically be adjusted through three parameters: (1) the transconductance of the core leakage submodule, which consists of $M_1, M_2, M_{11}, M_{12}$ and is controlled by the bias current $I_1$ (see Figure 4.3); (2) the total leakage current of the module, directly related to $M_9, M_{10}, M_{18}, M_{14}, M_{19}$; and (3) the capacitance value of the $C_{\text{mem}}$ array.

Among these, the tuning granularity follows a coarse-to-fine hierarchy: first, switching the $C_{\text{mem}}$ value (0.1–2 pF); second, adjusting the overall static current through $M_9, M_{10}, M_{18}, M_{14}, M_{19}$; and third, fine-tuning the core transconductance using the bias current of $M_{13}$.

The total transconductance of the leakage path, denoted $G_m$, is jointly determined by both the core transconductance and the total leakage current, and varies in the range of 0.106–0.164 $\mu$S. Consequently, the time constant $\tau_{\text{mem}}$ can be tuned over a wide range: 0.609–18.83 $\mu$s.

To ensure high linearity, transistors $M_{11}$ and $M_{12}$ are consistently biased to operate in the linear region (also referred to as region 1, where MOSFETs function

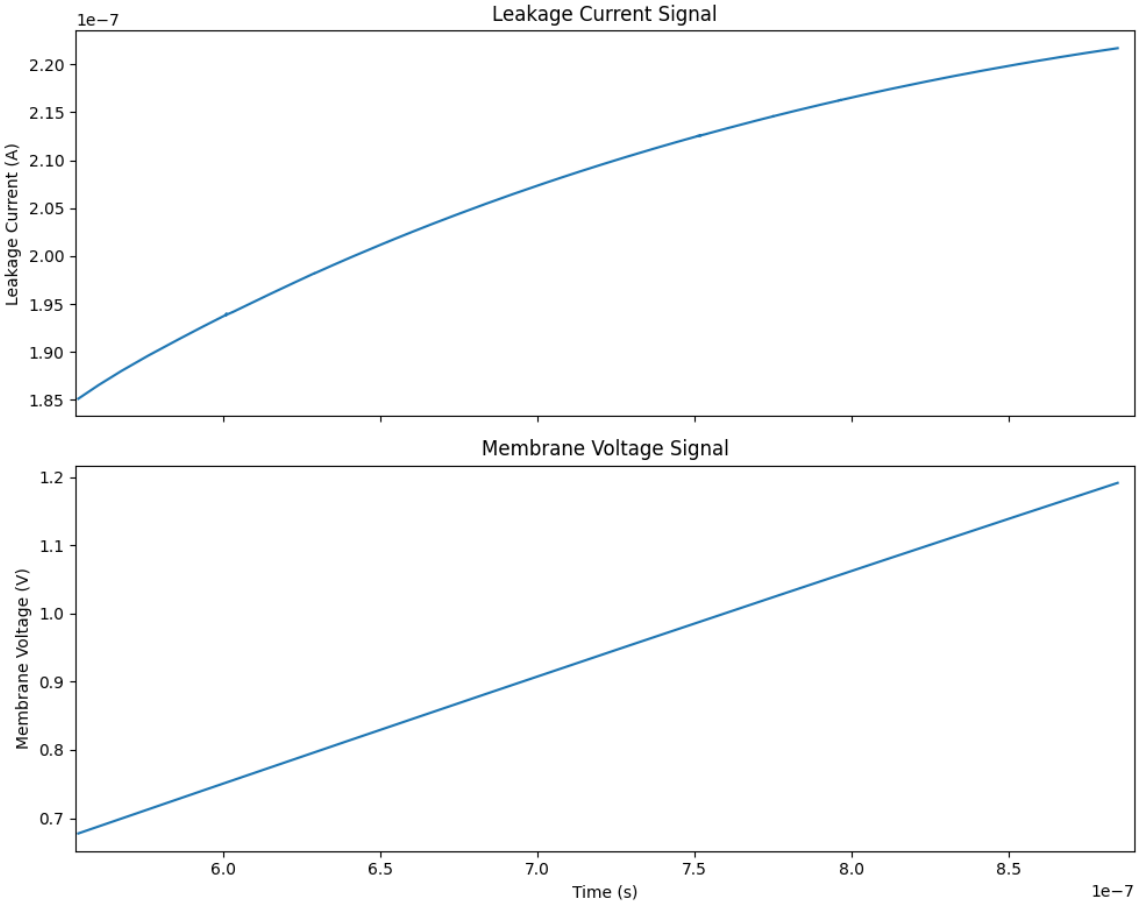
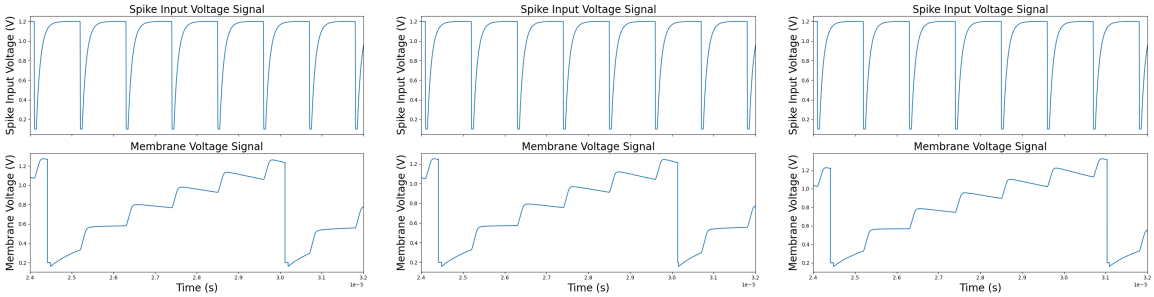



Figure 4.5: The linearity of the Voltage Leakage Module.

as resistive elements). This design avoids the need for extremely low bias currents to tune the time constant. To meet the demands of linear computation in neuromorphic accelerators, we introduce a programmable $V_{\text{leak}}$, which is one of the inputs of the Voltage Leakage Module (Figure 4.1). It is connected to the $V_{\text{inN}}$ node (Figure 4.3) to control the leakage rate, thereby enabling dynamic modulation of the total leakage conductance. As shown in Figure 4.5, the leakage current varies with changes in $V_{\text{mem}}$. When $V_{\text{mem}}$ lies within the range of 747–1083 mV, the transconductance exhibits less than 10% variation, ensuring high linearity.

Figure 4.5, Figure 4.6, and Table 4.2 demonstrate the calibration range of the leakage behavior. By adjusting $V_{\text{leak}}$, the leakage rate can be effectively modulated,




(a) $V_{\text{leak}}$= 559 mV (b) $V_{\text{leak}}$= 609 mV (c) $V_{\text{leak}}$= 659 mV

Figure 4.6: The linearity of the Voltage Leakage Module under different leakage control voltages $V_{\text{leak}}$.

improving the calibration flexibility of the system.

Similar to the Spiking Input Module, the $M_{14}$ MOSFET in the Voltage Leakage Module is used to set the static value of $V_{\text{mem}}$. Neither the Voltage Leakage Module nor the Spiking Input Module includes a dedicated startup circuit. As a result, when $V_{\text{mem}}$ lies between 0.5 and 1.4 V, the output-stage MOSFETs $M_{18}$ and $M_{10}$ operate in the saturation region. To improve the robustness of the startup behavior, we add transistor $M_{14}$ to provide a small additional current, pulling $V_{\text{mem}}$ toward its resting potential. To maintain high output impedance and minimize power consumption, $M_{27}$ is designed with a small $W/L$ ratio and operates in the subthreshold region.

## 4.3 Spiking Generation Module

The Spiking Generation Module consists primarily of a digital comparator and its associated driver circuit. The comparator is designed to operate at 10 MHz with the system and reliably handle input voltages from 0.5 V to 1.2 V, in accordance with system requirements. High sensitivity is essential to ensure timely and accurate operation, allowing the comparator to respond to differential inputs as small as 1%. To meet this specification, we designed a high-speed, high-precision comparator comprising two main components: a pre-amplifier stage and a regenerative latch core, as illustrated in Figure 4.7, Figure 4.8, Table 4.3, and Table 4.4. This digital design

integrates the functionality of an analog comparator and a sampling circuit, enabling direct discretization of the output signal. As a result, it significantly reduces both power consumption and area overhead for the neuromorphic system.

In the pre-amplifier, a differential pair composed of MOSFETs $M_1$ and $M_2$ is used. A tail current source transistor $M_3$ connects their shared source node to ground. When $V_{\text{inN}} > V_{\text{inP}}$, a larger portion of the current flows through $M_1$. The tail current source enhances this imbalance by increasing the current in the branch with the higher gate-source voltage differential. This differential amplification increases the voltage difference between $V_{\text{outN}}$ and $V_{\text{outP}}$, thereby improving the sensitivity of the comparator. Furthermore, the tail transistor suppresses common-mode voltage variations, enhancing the circuit's common-mode rejection ratio (CMRR). Because the tail current source is clock-controlled, it enables dynamic switching behavior in the comparator. The small-signal sensitivity prior to the addition of the tail transistor is approximately:

$$S_0 = A_{v0} = g_m \cdot r_p \tag{4.4}$$

When the regenerative latch is activated, the comparator exhibits an exponential increase in gain due to positive feedback. The comparator sensitivity during clock-triggered evaluation can be modeled as:

$$S_0 = A_{v0} \cdot e^{t/\tau} \cdot \delta(t), \quad \tau = \frac{C}{g_m} \tag{4.5}$$

Here, $\delta(t)$ represents the clock-induced trigger, and $e^{t/\tau}$ characterizes the gain amplification provided by the regenerative loop.

As shown in Figure 4.8, the regenerative comparator core comprises two sub-blocks: a cross-coupled latch and a differential input amplifier. The cross-coupled latch consists of $M_3$, $M_4$, $M_{2\text{b}}$, and $M_{3\text{b}}$—four MOSFETs, with each output node fed back to the gate of its complementary transistor to form a strong positive feedback

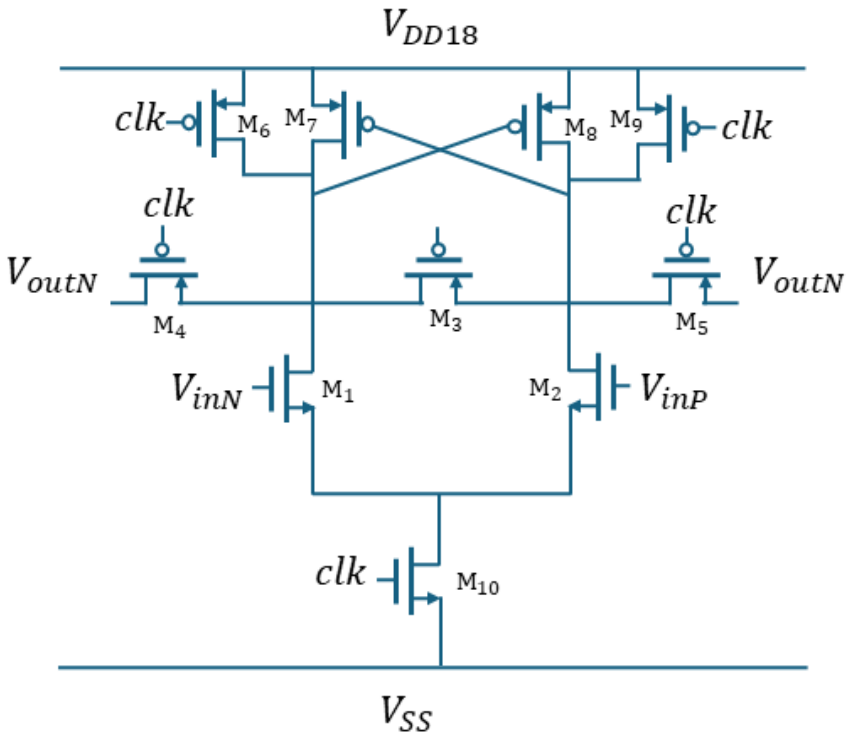



Figure 4.7: Detailed schematic of the pre-amplifier circuit within the Spiking Generation Module.

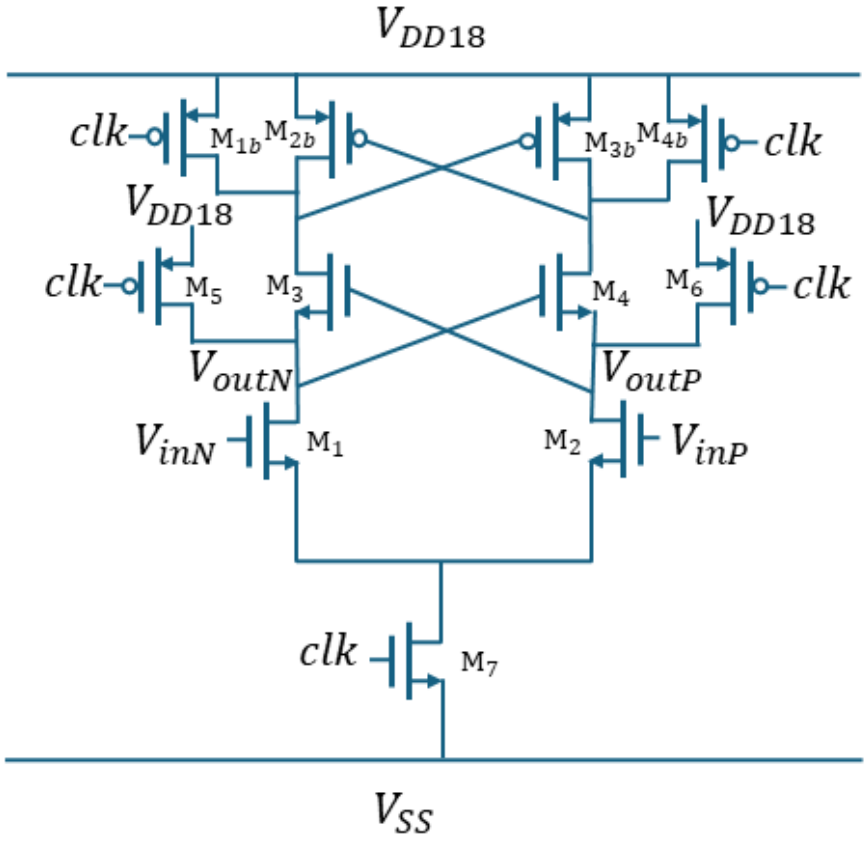



Figure 4.8: Detailed schematic of the comparator circuit within the Spiking Generation Module.

Table 4.3: Transistor sizes of the pre-amplifier in the Spiking Generation Module.

| Name | W/L |
|---|---|
| M1 | 30 $\mu$m / 0.28 $\mu$m |
| M2 | 32.1 $\mu$m / 0.28 $\mu$m |
| M3 | 20 $\mu$m / 0.28 $\mu$m |
| M4/M5 | 20 $\mu$m / 0.28 $\mu$m |
| M6/M9 | 40 $\mu$m / 2 $\mu$m |
| M7/M8 | 0.4 $\mu$m / 10 $\mu$m |
| M10 | 40 $\mu$m / 0.28 $\mu$m |

Table 4.4: Transistor size of the comparator in the Spiking Generation Module.

| Name | W/L |
|---|---|
| M1/M2 | 1.7 $\mu$m / 0.28 $\mu$m |
| M3/M4 | 2 $\mu$m / 0.28 $\mu$m |
| M5/M6 | 40 $\mu$m / 0.28 $\mu$m |
| M7 | 60 $\mu$m / 0.28 $\mu$m |
| M1b, M4b | 80 $\mu$m / 2 $\mu$m |
| M2b, M3b | 0.41 $\mu$m / 10 $\mu$m |

loop. The differential amplifier consists of $M_1$, $M_2$, $M_5$, $M_6$, $M_{1b}$, and $M_{2b}$. Among these, $M_1$ and $M_2$ receive the $V_{outN}$ and $V_{outP}$ signals from the pre-amplifier. MOSFETs $M_5$, $M_6$, $M_{1b}$, and $M_{2b}$ serve as clock-controlled pull-down switches responsible for resetting the regenerative comparator.

The Spiking Generation Module operates in two distinct phases:

**Phase 1: Pre-charge Phase (clk = 0):** The output nodes are reset to $V_{DD}$ through PMOS $M_5$, $M_6$, $M_{1b}$, and $M_{2b}$, as shown in Figure 4.8. During this period, the circuit remains in a static, non-comparing state.

**Phase 2: Evaluation Phase (clk = 1):** As shown in Figure 4.7, $M_{1b}$ turns on, enabling current flow through the differential pair. If $V_{inP} > V_{inN}$, the left branch conducts more, causing $V_{outN}$ to drop. This initiates positive feedback via the cross-coupled structure, further lowering the gate voltage of the right transistor and accelerating the rise of $V_{outP}$. This regenerative process can be described by the following expression:

$$v_{\text{gc-out}}(t) = v_{\text{gc0}} \cdot e^{t/\tau_{\text{gc}}}, \quad \tau_{\text{gc}} = \frac{C_{\text{gc}}}{g_{\text{gc}}} \tag{4.6}$$

where $v_{\text{gc0}}$ is the initial voltage difference at the output, $g_{\text{gc}}$ is the effective transconductance of the cross-coupled pair, and $C_{\text{gc}}$ is the effective capacitance at

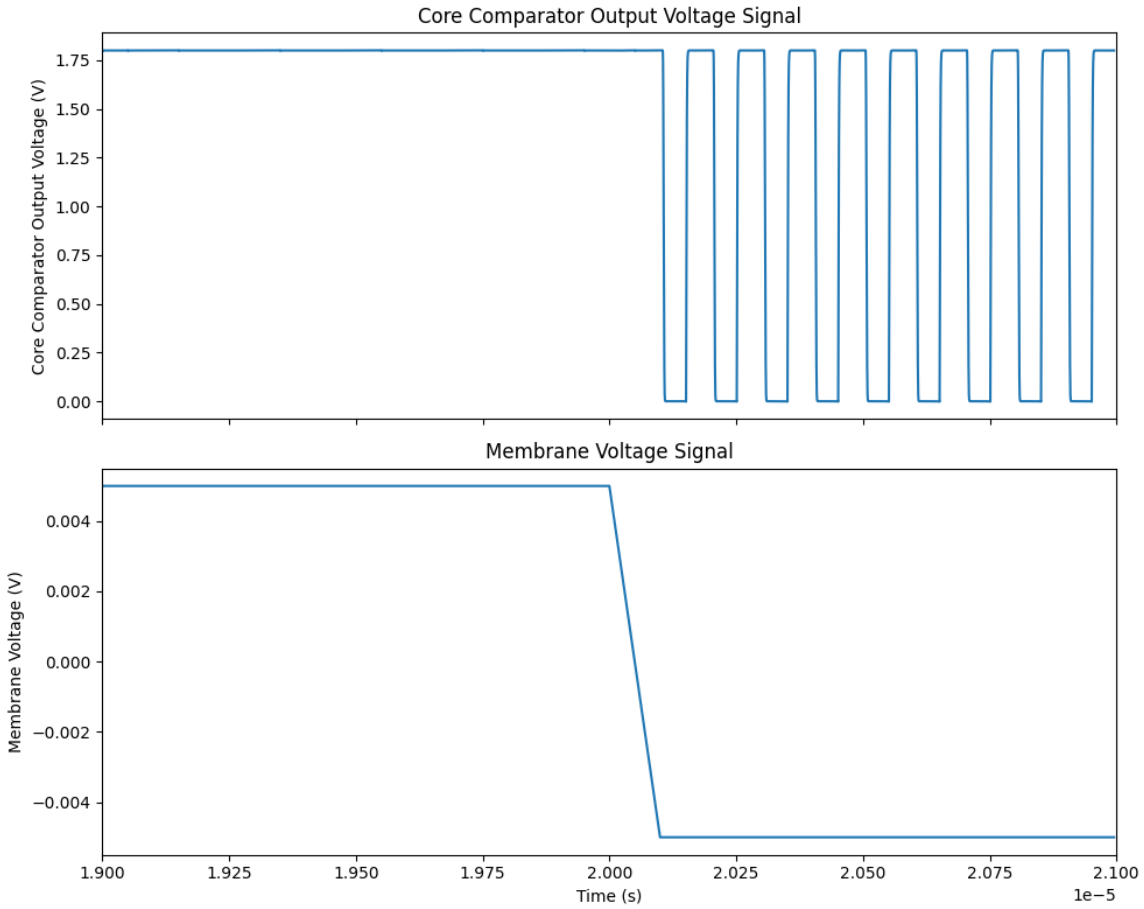



Figure 4.9: Linearity performance of the Voltage Leakage Module, derived from the temporal response of $V_{\text{mem}}$ shown in Figure 4.1.

the output node. To strengthen the load-driving capability of the differential stage, the comparator output is connected to a multi-stage inverter buffer. This buffer reduces the effective capacitance $C_{\text{gc}}$, increases the comparator's speed, and ensures robust signal delivery to downstream digital logic.

As shown in Figure 4.9, the shortest half-cycle of the clock-related signal is 45.325 ns. The comparator output circuit meets the timing requirement of occupying at least 90% of the half-cycle duration—i.e., with a 50 ns half-cycle, this corresponds to 45 ns.

## 4.4 System Reset Module

The system reset block consists of a driver circuit and a set of switches. Once the membrane voltage $V_{\text{mem}}$ reaches $V_{\text{thresh}}$, the comparator output transitions to a logic high level $V_{\text{OH}}$. After a defined delay $t_{\text{delay}}$, which is implemented by the system reset module, the comparator output is pulled down to a logic low level $V_{\text{OL}}$, effectively generating a voltage pulse. This pulse can serve as the input to a subsequent neuromorphic module. To regulate the delay $t_{\text{delay}}$ and rapidly discharge the membrane capacitor during the reset phase, the buffer adjusts the delay timing, while the switches discharge $C_{\text{mem}}$ and reset the spiking input, leakage, and generation modules.

If no downstream neuromorphic module is connected, the generated spike signal can be directly converted into a digital value by recording its timing. A lookup table is then used to map the spike timing to the corresponding digital output value, serving as the final computational result.

## 4.5 Transaction Performance and Power Consumption

Figure 4.2 illustrates the transient behavior of the entire neuromorphic model. The neuromorphic module can operate reliably under an input spiking frequency of 10 MHz, thanks to the seamless coordination among the submodules described in this chapter.

Another critical performance metric is the module's power consumption. It consists of two components: **dynamic power**—the energy consumed for each spike event—and **static power**—the baseline power consumption of all submodules during idle periods. Figure 4.10 presents the current waveform at the supply voltage ($V_{\text{DD}}$) during a spike event. A sharp current peak is observed when the membrane capacitor ($C_{\text{mem}}$) discharges. The energy consumed solely by the capacitor discharge is estimated to be approximately 1.44 pJ. When considering contributions from all active submodules during the spike duration, the total energy per spike amounts to 10.12 pJ. In addition, the measured static power consumption of the module is 26.12 uW. As

shown in Table 4.5, these values demonstrate that the proposed design achieves competitive energy efficiency among state-of-the-art neuromorphic computing systems.

Table 4.5: Comparison of energy per spike across neuromorphic platforms.

| **Platform** | **Energy per Spike** |
|---|---|
| HICANN-DLS [10] | 790 pJ |
| IBM TrueNorth [22] | 26 pJ |
| Neurogrid [8] | 941 pJ |
| BrainScaleS [3] | 10 pJ -14 pJ |
| DYNAP-SE [9] | 50 pJ |
| Intel Loihi [29] | 23.6 pJ |
| **This work** | 10.12 pJ |

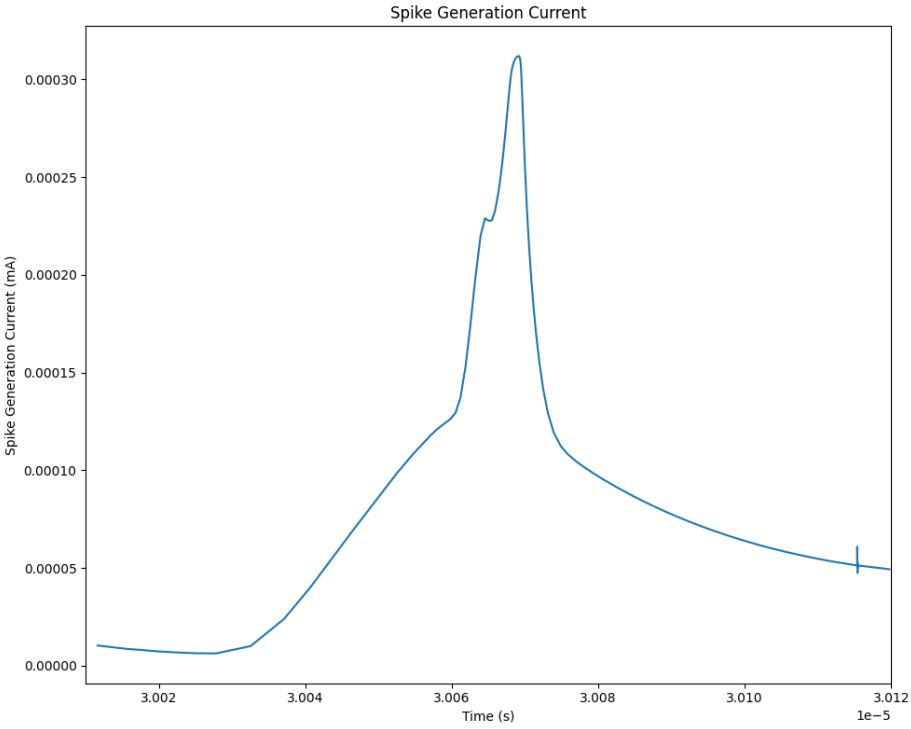



Figure 4.10: Power consumption per spike.

# Chapter 5: Summary and Conclusion

This work presents a low-power, area-efficient neuromorphic neuron accelerator circuit tailored for compatibility with the BrainScaleS architecture. Inspired by the efficiency of biological computation, our design integrates core principles of neuromorphic engineering—including local memory-compute co-location, event-driven processing, and temporal coding—to deliver a compact hardware solution for spiking neural network (SNN) acceleration.

This paper demonstrated that the proposed circuit achieves reliable performance under high-frequency spike input (10 MHz) and exhibits consistent membrane voltage behavior across a wide range of input conditions. The power consumption analysis confirms that the design operates with high energy efficiency, consuming only 10.12 pJ per spike and maintaining a static power of 26.12 uW. These results position the design as a promising candidate for integration into larger neuromorphic systems such as BrainScaleS.

In conclusion, this work contributes a robust and energy-efficient building block for future neuromorphic processors. By adhering to biologically inspired design principles while leveraging circuit-level optimization, this approach advances the development of scalable analog-domain SNN accelerators. Future work may focus on extending this architecture to support synaptic plasticity mechanisms such as STDP, integrating multi-neuron arrays, or adapting the design for fabrication in advanced CMOS or emerging memristive technologies.

# References


[1] G. Indiveri and S.-C. Liu, "Memory and information processing in neuromorphic systems," *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1379–1397, 2015.

[2] D. V. Christensen, R. Dittmann, B. Linares-Barranco, A. Sebastian, M. Le Gallo, A. Redaelli, S. Slesazeck, T. Mikolajick, S. Spiga, S. Menzel *et al.*, "2022 roadmap on neuromorphic computing and engineering," *Neuromorphic Computing and Engineering*, vol. 2, no. 2, p. 022501, 2022.

[3] S. Schmitt, J. Klähn, G. Bellec, A. Grübl, M. Guettler, A. Hartel, S. Hartmann, D. Husmann, K. Husmann, S. Jeltsch *et al.*, "Neuromorphic hardware in the loop: Training a deep spiking network on the brainscales wafer-scale system," in *International Joint Conference on Neural Networks*, 2017, pp. 2227–2234.

[4] C. Pehle, S. Billaudelle, B. Cramer, J. Kaiser, K. Schreiber, Y. Stradmann, J. Weis, A. Leibfried, E. Müller, and J. Schemmel, "The brainscales-2 accelerated neuromorphic system with hybrid plasticity," *Frontiers in Neuroscience*, vol. 16, p. 795876, 2022.

[5] E. Müller, S. Schmitt, C. Mauch, S. Billaudelle, A. Grübl, M. Güttler, D. Husmann, J. Ilmberger, S. Jeltsch, J. Kaiser *et al.*, "The operating system of the neuromorphic brainscales-1 system," *Neurocomputing*, vol. 501, pp. 790–810, 2022.

[6] R. Brette and W. Gerstner, "Adaptive exponential integrate-and-fire model as an effective description of neuronal activity," *Journal of Neurophysiology*, vol. 94, no. 5, pp. 3637–3642, 2005.

[7] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asyn-

chronous processors (dynaps)," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 12, no. 1, pp. 106–122, 2017.

[8] B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J.-M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. A. Merolla, and K. Boahen, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 699–716, 2014.

[9] N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, and G. Indiveri, "A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses," *Frontiers in Neuroscience*, vol. 9, p. 141, 2015.

[10] S. A. Aamir, Y. Stradmann, P. Müller, C. Pehle, A. Hartel, A. Grübl, J. Schemmel, and K. Meier, "An accelerated lif neuronal network array for a large-scale mixed-signal neuromorphic architecture," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 12, pp. 4299–4312, 2018.

[11] M. A. Nahmias, B. J. Shastri, A. N. Tait, and P. R. Prucnal, "A leaky integrate-and-fire laser neuron for ultrafast cognitive computing," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 19, no. 5, pp. 1–12, 2013.

[12] N. Caporale and Y. Dan, "Spike timing–dependent plasticity: A hebbian learning rule," *Annu. Rev. Neurosci.*, vol. 31, no. 1, pp. 25–46, 2008.

[13] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[14] S. Hayman, "The mcculloch-pitts model," in *International Joint Conference on Neural Networks. Proceedings* , vol. 6, 1999, pp. 4438–4439.

[15] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain." *Psychological Review*, vol. 65, no. 6, p. 386, 1958.

[16] J. J. Hopfield, "Hopfield network," *Scholarpedia*, vol. 2, no. 5, p. 1977, 2007.

[17] C. Mead and M. Ismail, *Analog VLSI Implementation of Neural Systems*. Springer Science & Business Media, 2012, vol. 80.

[18] S. Ghosh-Dastidar and H. Adeli, "Spiking neural networks," *International Journal of Neural Systems*, vol. 19, no. 04, pp. 295–308, 2009.

[19] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 51–63, 2019.

[20] F. Zenke and T. P. Vogels, "The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks," *Neural Computation*, vol. 33, no. 4, pp. 899–925, 2021.

[21] G. Bellec, F. Scherr, A. Subramoney, E. Hajek, D. Salaj, R. Legenstein, and W. Maass, "A solution to the learning dilemma for recurrent networks of spiking neurons," *Nature Communications*, vol. 11, no. 1, p. 3625, 2020.

[22] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, G.-J. Nam *et al.*, "Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip," *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, vol. 34, no. 10, pp. 1537–1557, 2015.

[23] A. P. Davison, D. Brüderle, J. M. Eppler, J. Kremkow, E. Muller, D. Pecevski, L. Perrinet, and P. Yger, "Pynn: A common interface for neuronal network simulators," *Frontiers in Neuroinformatics*, vol. 2, p. 388, 2009.

[24] Lava Neuromorphic Computing Project, "Lava-dl: A deep learning library for neuromorphic computing," https://github.com/lava-nc/lava-dl, 2021, accessed: April 14, 2025.

[25] J. K. Eshraghian, M. Ward, E. O. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu, "Training spiking neural networks using lessons from deep learning," *Proceedings of the IEEE*, vol. 111, no. 9, pp. 1016–1054, 2023.

[26] Arm Education Media, *Modern System-on-chip Design on Arm.* ARM LTD., 2021. [Online]. Available: https://www.arm.com/resources/education/books/modern-soc

[27] Analog Devices, "Analog devices expands linux distribution with over 1000 device drivers to support the development of high-performance solutions," https://www.businesswire.com/news/home/20211130005011/en/Analog-Devices-Expands-Linux-Distribu 2021, accessed: April 17, 2025.

[28] D. Tal and E. L. Schwartz, "Computing with the leaky integrate-and-fire neuron: Logarithmic computation and multiplication," *Neural Computation*, vol. 9, no. 2, pp. 305–318, 1997.

[29] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.