Unsupervised SFQ-Based Spiking Neural Network

Mustafa Altay Karamuftuoglu, Beyza Zeynep Ucpinar, Sasan Razmkhah, Mehdi Kamal, *Senior Member, IEEE*, and Massoud Pedram, *Fellow, IEEE*

(Invited Paper)

***Abstract*—Single Flux Quantum (SFQ) technology represents a groundbreaking advancement in computational efficiency and ultra-high-speed neuromorphic processing. The key features of SFQ technology, particularly data representation, transmission, and processing through SFQ pulses, closely mirror fundamental aspects of biological neural structures. Consequently, SFQ-based circuits emerge as an ideal candidate for realizing Spiking Neural Networks (SNNs). This study presents a proof-of-concept demonstration of an SFQ-based SNN architecture, showcasing its capacity for ultra-fast switching at remarkably low energy consumption per output activity. Notably, our work introduces innovative approaches: (i) We introduce a novel spike-timing-dependent plasticity mechanism to update synapses and to trace spike-activity by incorporating a leaky non-destructive readout circuit. (ii) We propose a novel method to dynamically regulate the threshold behavior of leaky integrate and fire superconductor neurons, enhancing the adaptability of our SNN architecture. (iii) Our research incorporates a novel winner-take-all mechanism, aligning with practical strategies for SNN development and enabling effective decision-making processes. The effectiveness of these proposed structural enhancements is evaluated by integrating high-level models into the BindsNET framework. By leveraging BindsNET, we model the online training of an SNN, integrating the novel structures into the learning process. To ensure the robustness and functionality of our circuits, we employ JoSIM for circuit parameter extraction and functional verification through simulation.**

***Index Terms*—Single flux quantum, superconductor electronics, spiking neural network, synapse, STDP.**

I. Introduction

GROWING demand for neural networks has led to innovative solutions that combine fundamental biological principles with hardware implementations. These solutions and the developments in computational neuroscience have a notable influence on the paradigm shift from artificial neural networks (ANNs) to the domain of spiking neural networks (SNNs) due to their distinctive properties of energy efficiency and inference capabilities [1]. SFQ circuits with spike-based behavior show great promise in efficient and fast SNN implementation.

Neural data represented with spikes intrinsically resembles the data on superconductor devices [2], [3]. Furthermore, the shift from the conventional floating point representation to a binary paradigm of 0 s and 1 s results in notable simplifications and reduced memory requirements. The inherent sparsity in SNNs, where neurons spend most of their time resting, aligns perfectly with the concept of event-driven processing with asynchronous superconductor circuits. This ultimately leads to substantial power savings by eliminating the need for most of the computational operations. Thus, the utilization of superconductor devices on SNN holds great promise for the performance of neuromorphic computing systems [4].

Superconductor-based SNN designs necessitate the integration of superconductor circuits that accurately replicate the intricate dynamics observed in biological neurons, specifically translating states and actions into neural spikes. Within this paradigm, leveraging the unsupervised learning mechanisms inherent to SNNs, in conjunction with the capabilities of superconductor technology, empowers us to establish a biologically plausible framework for simulating neural networks.

Schneider et al. [5] showcased character recognition using an SNN model, explicitly emphasizing the letters 'z,' 'v,' and 'n.' In their study, the authors employed a 3×3 input pixel array and implemented a two-layer inference SNN that incorporated Integrate-and-Fire (IF) neurons and Magnetic Josephson Junctions (MJJs). Bozbey et al. [6] extended the SNN research by utilizing superconductor Leaky Integrate-and-Fire (LIF) neurons with CMOS-superconductor synapses for the inference SNN. The training process was executed using genetic algorithms applied to the iris dataset. Furthermore, Zhang et al. [7] contributed to the field by exploring SNNs featuring IF neurons with Quantum Phase-Slip Junctions (QPSJ). They analyzed superconductor SNN training, using the digit '0' from the MNIST dataset as their experimental basis. Of particular interest, Segall et al. [8] introduced a 1-bit resolution Spike-Timing-Dependent Plasticity (STDP) structure, advancing the prospects of unsupervised learning with superconductor devices.

Collectively, these papers share a common focus on alternative inference neural networks. However, it is crucial to acknowledge a significant limitation–the immaturity of superconductor fabrication technologies for MJJs and QPSJs. Consequently, our contributions primarily center around online training with

Manuscript received 26 September 2023; revised 11 January 2024; accepted 24 January 2024. Date of publication 20 February 2024; date of current version 1 March 2024. This work was supported by National Science Foundation through Expedition DISCoVER Project under Grant 2124453. *(Corresponding author: Mustafa Altay Karamuftuoglu.)*

The authors are with the Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA 90007 USA (e-mail: karamuft@usc.edu; ucpinar@usc.edu; razmkhah@usc.edu; mehdi.kamal@usc.edu; pedram@usc.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASC.2024.3367618.

Digital Object Identifier 10.1109/TASC.2024.3367618

conventional superconductor elements that can be readily fabricated using available foundry processes.

This work focuses on training SNNs utilizing Spike-Timing-Dependent Plasticity (STDP) while providing justifications for integrating superconductor components. Within the scope of our research, we have carefully designed an STDP mechanism tailored for a synaptic finite state machine specifically optimized for Single Flux Quantum (SFQ)–based SNNs. Additionally, we introduce leaky integrate-and-fire (LIF) neurons *with dynamic thresholds* achieved through self-inhibition. This unique feature empowers LIF neurons to adapt dynamically to input patterns, leveraging the temporal diversity among neurons to enhance overall network performance. We conducted simulations using JoSIM [9] to ensure the functionality and accuracy of our designs.

For network analysis, we leveraged the BindsNET framework's capabilities [10] and applied them to an architecture representing an asynchronous SNN with two layers [11]. Throughout our analysis, we maintained an evaluation range aligned with the capabilities of superconductor hardware, yielding high levels of accuracy in our observations and assessments.

The key contributions of this paper are as follows. (i) Quantized STDP Mechanism: We introduce a novel STDP mechanism that is quantized and utilizes a leaky non-destructive readout (NDRO); (ii) Dynamic Threshold Behavior: We demonstrate an innovative self-inhibition technique that temporarily modulates LIF neurons' membrane potential, enabling dynamic threshold behavior; (iii) Winner-Take-All Superconductor Structure: We present a novel superconductor structure designed to implement the winner-take-all principle within the context of neural networks; and (iv) Computational Framework: We employ plausible mechanisms within a computational framework to systematically verify and observe the computational behavior of SNNs.

II. Methodology

The development of spiking neurons and their computational models are dedicated to faithfully mirroring the behavior of biological neurons. These methodologies focus on capturing spike-based activity, which enables precise temporal information encoding. By integrating these neurons with synaptic plasticity mechanisms, neural networks evolve into powerful tools for facilitating unsupervised learning. To achieve this, Spike-Timing-Dependent Plasticity (STDP) plays a key role, enabling networks with adaptive capabilities by modulating the strengths of synaptic connections based on the precise timing of spikes. To evaluate the high-level performance of an SNN, we leverage the BindsNET framework considering the superconductor electronics constraints.

BindsNET is an open-source library offering a user-friendly solution for training and evaluating SNNs on CPU and GPU platforms. Built on the PyTorch library [12], it incorporates ML tools and robust data structures and can be implemented on different platforms. BindsNET framework supports various backends, including TensorFlow [13] and SpiNNaker [14]. By providing an interface to the OpenAI Gym library, training and evaluation of spiking networks can be facilitated in reinforcement learning environments. This comprehensive approach addresses the critical need for effective integration between SNNs and real-world tasks. In the following subsections, we delve into the network architecture and the crucial role of synaptic adaptability as a foundational feature, effectively emulating the intricate dynamics of biological neural networks.

A. Spiking Neural Network Architecture

The network architecture that we follow fundamentally consists of two layers: an input layer and a processing layer as an output layer [11]. For the input, neural encoding techniques are applied to transform input pixels into spikes, such as rate coding, temporal coding, and sparse coding. In particular, we focus on the rate coding scheme that converts a pixel value into a rate of spikes using Poisson distribution [15]. In this approach, the source of the input spikes can be any asynchronous input, such as a sensor. The incoming spikes are then propagated to the processing layer after being weighted by synapses.

The processing layer consists of excitatory and inhibitory neurons. The overall decision-making is performed by excitatory neurons with the help of inhibitory neurons. After being weighted by synapses, the input spikes are initially provided to the excitatory neurons. The synaptic connections from the input layer to excitatory neurons are established in a fully connected fashion. Inhibitory neurons are incorporated into the structure to introduce competitive dynamics among the excitatory neurons.

The connections from excitatory to inhibitory neurons are one-to-one. In contrast, the links from inhibitory to excitatory neurons are fully connected. Here, the excitatory neuron that provides the initial spikes to the inhibitory neuron is excluded. In this approach, if an excitatory neuron generates output, these spikes trigger the corresponding inhibitory neuron. Once the inhibitory neuron generates an output spike, it will prevent the rest of the excitatory neurons from firing. This paradigm is defined as the winner-take-all (WTA) principle [16]. This network configuration resembles recurrent neural networks due to lateral inhibition.

In our approach, we utilize a slightly modified version, shown in Fig. 1, of the previously described architecture. For our work, we assigned a single-spike threshold to inhibitory neurons. As a result, these neurons perform just the propagation of spikes with a high fanout back to the excitatory neurons for the operation of inhibition. Therefore, we exclude the inhibitory neurons from the architecture and create a WTA feedback mechanism among the excitatory neurons to establish the same functionality as inhibitory neurons.

B. Spike-Timing-Dependent Plasticity (STDP)

STDP is a phenomenon in which the timing of spikes in neural networks influences both the direction (sign) and magnitude of changes in synaptic strength. It is considered one of the primary learning rules governing synaptic plasticity and is a biologically plausible mechanism for unsupervised learning, as discussed in reference [17]. Conceptually, STDP is often interpreted as a form

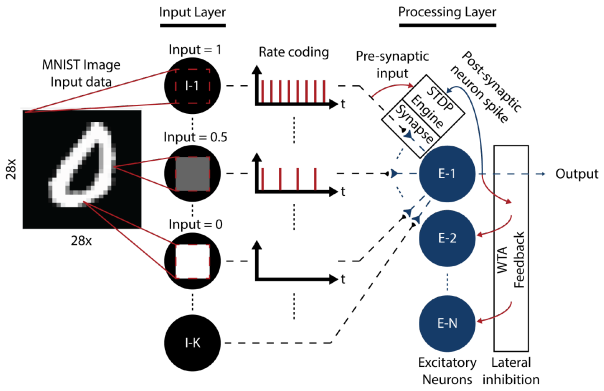


Fig. 1. Visualization of rate coding on the MNIST input image pixels and our network architecture. In the processing layer, a single excitatory neuron performs lateral inhibition over the other excitatory neurons, preventing them from firing due to the WTA feedback mechanism. The labels *I* and *E* represent input and excitatory neuron vertices, respectively. The variable *K* corresponds to the number of input pixels, whereas *N* shows the number of neurons in the processing layer.

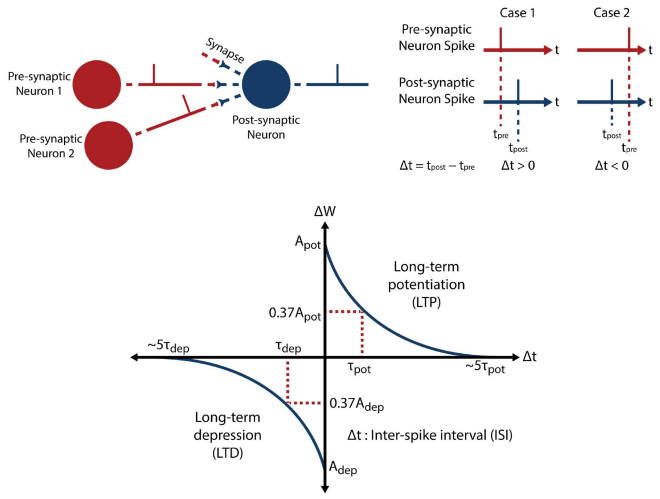


Fig. 2. Example neuron and synapse scheme with pre and post-synaptic neuron spike cases for the learning curves of STDP.

of Hebbian learning, which posits that synapses are strengthened when neurons fire together.

In STDP, the precise timing of pre-synaptic and post-synaptic neuron spikes within a narrow time window plays a critical role in determining the direction of synaptic changes. When a pre-synaptic neuron spike precedes a post-synaptic neuron spike within this window, it leads to a phenomenon known as long-term potentiation (LTP), which strengthens the synaptic connection. Conversely, if the order is reversed, with the post-synaptic neuron spike preceding the pre-synaptic one, it results in long-term depression (LTD), which weakens the synaptic connection.

In experimental settings, researchers often repeatedly evoke pairs of pre-synaptic and post-synaptic spikes with a fixed time interval, denoted as Δt. These pairs of spikes are typically repeated at a low frequency, and the resulting changes in synaptic response size are measured. By conducting this experiment for various values of Δt, the timing-dependence of plasticity is mapped, creating what is referred to as an *STDP curve*. This curve is a valuable tool for predicting the plasticity outcomes when Δt varies, such as in response to arbitrary sequences of pre-synaptic and post-synaptic neuron spikes under less controlled conditions [18].

The visual representation of neurons and synapses undergoing different weight update scenarios in the context of STDP is depicted in Fig. 2. Let's consider two specific cases. **Case 1:** In this scenario, the input from pre-synaptic neuron 1 triggers the post-synaptic neuron to generate output spikes. This causal relationship increases the synapse's strength that connects the pre and post-synaptic neurons. **Case 2:** In contrast, pre-synaptic neuron 2 does not contribute to the output generation, and its spike arrives later than the output of the post-synaptic neuron. As a result, the strength of the synapse connecting pre-synaptic neuron 2 and the post-synaptic neuron is decreased.

The mathematical expression for the weight changes in these scenarios is provided by (1). This equation quantifies how the synaptic weights are updated based on the timing and causal relationship between pre-synaptic and post-synaptic spikes, reflecting the principles of STDP.

$$\Delta W = \begin{cases} A_{pot} e^{-\Delta t/\tau_{pot}} & \text{if } \Delta t > 0 \\ A_{dep} e^{+\Delta t/\tau_{dep}} & \text{if } \Delta t < 0 \end{cases} \tag{1}$$

The weight modulation in (1) expresses the principles of the asymmetric learning rule with two regions: LTP for weight increment and LTD for weight reduction. ΔW represents the amount of change in synaptic strength. In order to realize this functionality in a circuit, a trace-based method can be implemented [19]. The exponential curve on the LTP region corresponds to the post-synaptic neuron spike-trace, whereas the LTD region curve represents the pre-synaptic neuron spike-trace. These traces capture the spiking activity of the pre-synaptic and post-synaptic neurons. The two variables A_{pot} and A_{dep} denote the maximum increment and decrement of synaptic strength on the *STDP curve*, respectively. The direction of ΔW depends on the sign value of Δt as determined by the arrival order of pre and post-synaptic neuron spikes corresponding to $\Delta t = t_{post} - t_{pre}$. Due to the resource constraints on the superconductor hardware, we employed quantized STDP update levels on ΔW and synaptic weights that are discussed in the following section.

III. Proposed SFQ-Based Online Training

This section introduces essential superconductor-based mechanisms for the implementation of SNNs. Firstly, we present an STDP engine that effectively enforces the asymmetric rate STDP learning rule, underlining the motivation for synaptic implementations with a degree of biological plausibility [17]. Our design employs a *modified NDRO circuit* to monitor pre- and post-synaptic neuron spikes precisely. Furthermore, our framework seamlessly integrates dynamic threshold behavior within LIF neurons, achieved through self-inhibition based on

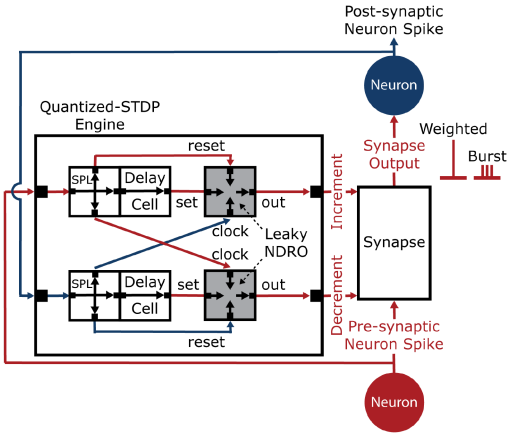


Fig. 3. Cell view of the quantized STDP design following the scheme given in Fig. 2. Trace-based spiking activity of pre and post-synaptic neurons is recorded in each leaky NDRO.

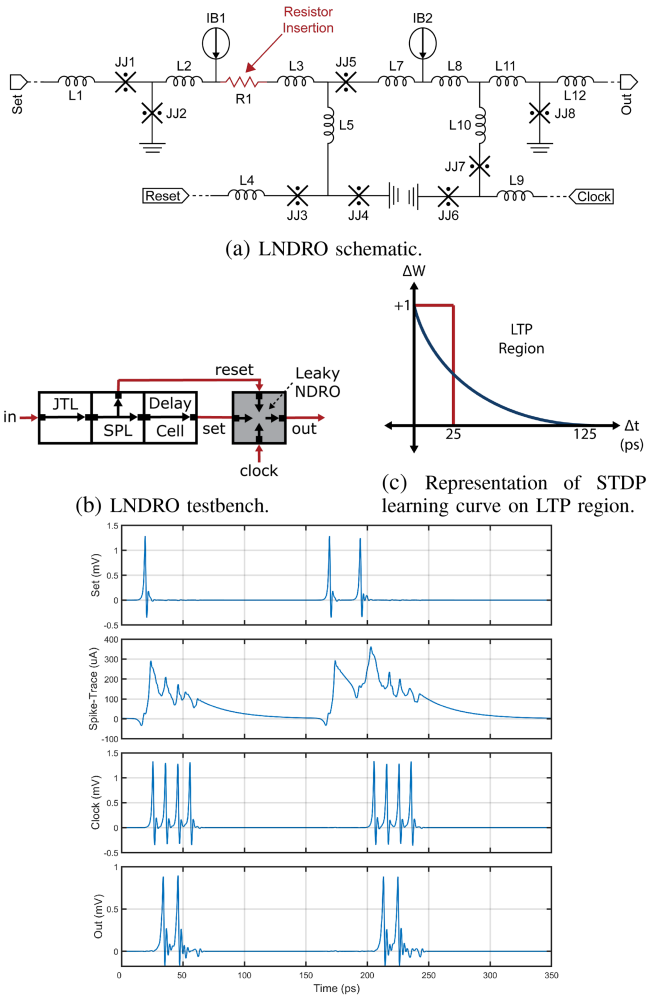


Fig. 4. LNDRO functional verification and its representation for the quantized STDP on the learning curve.

neuron output spikes [20]. Lastly, we propose a novel implementation of feedback interactions between neurons, facilitating our framework's realization of WTA characteristics. These combined mechanisms advance the field of SNNs and provide valuable tools for neuromorphic computing applications.

A. SFQ-Based STDP Engine

The proposed design of the STDP mechanism is tailored to perform both increment and decrement functions, thereby adjusting a finite state machine associated with the synapse structure. To enable STDP using superconductor-based components, we discretized the learning curve. This design, featuring a 1-bit resolution, incorporates two splitters (SPLs) with a fanout of 3, in addition to two leaky Non-Destructive Readout (LNDRO) circuits. The SPLs play a crucial role in internally increasing the fanout of inputs derived from pre- and post-synaptic neuron spikes, which are then assigned to the LNDRO pins. Furthermore, the STDP engine includes two output pins, labeled as *increment* and *decrement*, as illustrated in Fig. 3.

To generate a decrement output spike, one must establish the spike-trace relationship between pre- and post-synaptic neurons. In this operation, incorporating Splitters (SPLs) and a single LNDRO cell proves sufficient to generate decrement behavior with a 1-bit resolution. A similar setup can be set up to produce spikes on the increment output pin for generating an increment behavior. If a higher bit resolution is desired, the overall design necessitates the incorporation of SPLs with increased fanout, additional LNDROs with varying decay rates, and the inclusion of two merger components for LNDRO outputs. The introduction of leaky behavior in the standard NDRO cell is achieved by inserting a resistor into the SFQ storage loop, as illustrated in Fig. 4(a). In this design, an NDRO with multi-flux storing characteristics is suitable to create the synapse behavior since the amount of stored flux corresponds to the state of a synapse [21]. For such module, it is feasible to incorporate a parallel implementation of multi-flux NDROs for scaling up the structure, implementing the functionality of an up-down counter. In this approach, creating an output pin for each NDRO enables delivering multiple pulses concurrently to a neuron in order to meet the time window of a neuron's decay rate for the activation.

The LNDRO is simulated using JoSIM, and the waveforms are given in Fig. 4(d). When a spike arrives at the input *in*, the spike is split for the *reset* and *set* pins. To recreate the correct spike-trace functionality, the spike from *reset* initially erases the current in the leaky storing loop (JJ2-L2-R1-L3-L5-JJ4). By applying the spike arriving late due to the *Delay cell*, the spike-trace is updated, and the current gradually decays with a constant time of τ due to the inserted resistor. Concurrently, the state of the LNDRO can be read by a spike from the *clock* pin. Due to the quantization of the STDP curve shown in Fig. 4(c), a spike on *out* can only be observed until a quantization point, set as 25 ps in the simulation.

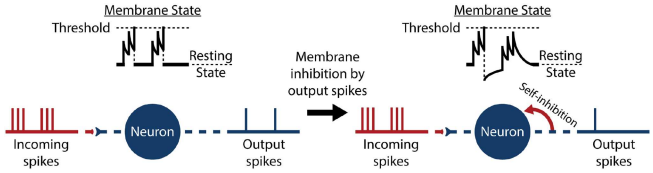


Fig. 5. LIF neuron with dynamic threshold using self-inhibition. The membrane state of a superconductor LIF neuron corresponds to the amount of current stored in its leaky loop. The neuron resting state is the baseline condition where no spike is received or generated.

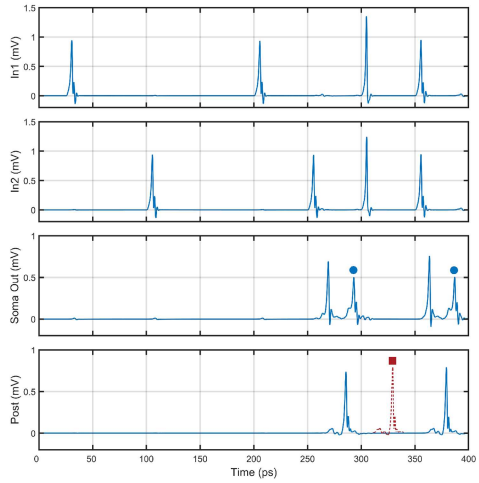
B. LIF Superconductor Neurons With Dynamic Threshold

The threshold value of a neuron determines the firing rate and shapes the computational behavior of a neural network. In this context, relying on a neuron with a fixed threshold can impose challenges in neural processing, such as high sensitivity to input fluctuations and excessive spike firing, resulting in high dynamic power consumption. The adaptability of a neuron with a dynamic threshold contributes to the network stability and contextual responsiveness [22]. For instance, the digits in the MNIST handwritten dataset may share the same pixels. In this case, the multiple neurons in the output layer may have high-valued synaptic weights on these pixels and generate an output spike due to the neurons having the same threshold for the classification, hindering the overall performance. Hence, the implementation of an adaptive threshold behavior becomes indispensable to address this issue effectively. Adjusting the threshold of a superconductor-based neuron is typically achieved by dynamically changing the bias current and critical current of the JJs. Such modifications introduce additional hardware design complexity. Therefore, we utilize the generated output spike as a feedback input, creating self-inhibition behavior as shown in Fig. 5.

The membrane state of a neuron plays a pivotal role in determining whether an output spike is generated at a specific moment in time. When the membrane state surpasses a certain threshold, a single spike is triggered at the neuron's output. The resting state of a neuron corresponds to its default state, characterized by the absence of spike activity. Within a neural network, certain neurons may exhibit a high firing rate, exerting a disproportionate influence on decision-making and potentially disrupting the network's balance. Consequently, there is a need for methods to mitigate such behavior and maintain network equilibrium.

The proposed self-inhibition technique serves as a mechanism for temporarily preventing a neuron from firing multiple spikes in quick succession. It achieves this by reducing the membrane state below the resting state, as illustrated in Fig. 5. In the context of superconductor circuits, the membrane state corresponds to the amplitude of stored current, and the threshold is determined by the amount of current required to trigger the decision-making Josephson junction (JJ). By employing the self-inhibition mechanism, a neuron can momentarily dip below its resting state, thus demanding more input current to reach the threshold level. An example design utilizing α-Soma [23] is depicted in Fig. 6(a).

(a) Cell view for LIF neuron behavior with α-Soma using the same cell parameters (left without self-inhibition, right with self-inhibition).



(b) Simulation with digital self-inhibition. The self-inhibition spikes are marked with a circle. In the case of no self-inhibition, a spike with a square mark would also be expected.

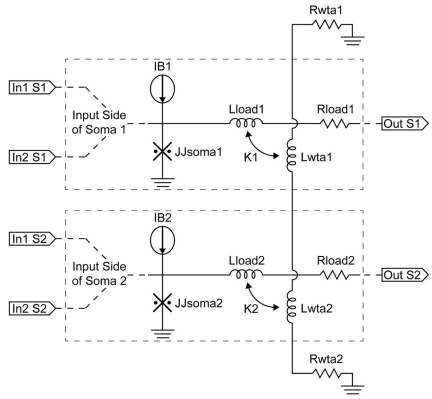
Fig. 6. Design of LIF neuron with self-inhibition and JoSIM results for α-Soma. All parameter values of the cells are kept the same, and the self-inhibition is prevented by disconnecting the node between the input of α-cell and output of SPL.

A fundamental component influencing the computation and decision-making in neural networks is the soma circuit. The soma generates an output pulse upon reaching action potential [24]. This behavior corresponds to a comparator circuit, the primary activation function for the behavior of a LIF neuron. Moreover, the self-inhibitory effect on the circuit is accomplished through the introduction of an α-cell, which performs bi-directional pulse propagation on the same datapath. In addition to the given design with electrical connections, a similar effect can be achieved for neurons receiving inputs from inductances with mutual couplings [25] by directing the neuron's output spike back into the same neuron's input through an inductance with negative coupling. This generates a current in the opposite direction, counteracting the threshold mechanism.

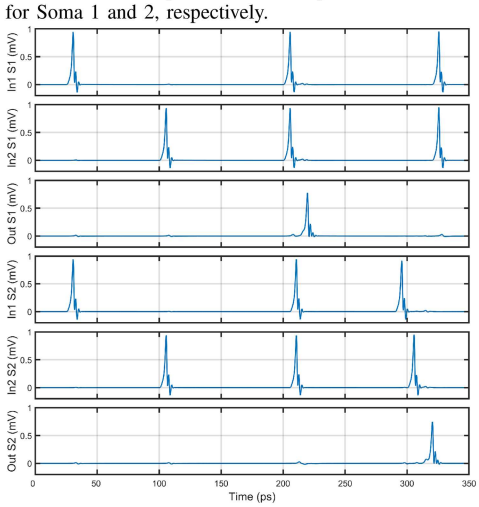
The simplified test case utilizes a soma circuit that acts as an asynchronous threshold gate with a threshold of two spikes within $\approx$50 ps i.e., an output spike is generated whenever two input spikes arrive within the designated time frame. Within this configuration employing α-Soma, the output spike is provided to the following SPL cell. One of the SPL outputs is connected to the input of α-cell, enabling the spike to propagate back into the soma cell. This operation negatively impacts the loop current due to the opposite direction of the spike. The example demonstration of digital inhibition with α-Soma preventing the generation of continuous output spikes is shown in Fig. 6.

C. WTA Mechanism With Superconductor Devices

One of the fundamental computational models in spiking neural networks is a winner-take-all (WTA) principle, establishing



(a) WTA mechanism with superconductor components. S1 and S2 represent the labels for Soma 1 and 2, respectively.



(b) Simulation of WTA mechanism. (Lload = 2 pH, Rload = 0.6 Ω, Lwta = 3 pH, Rwta = 0.3 Ω, K = 0.7)

Fig. 7. WTA mechanism and JoSIM results for soma circuits.

a form of competition for activation among the neurons within the same layer [16]. The neuron with a higher activation affects the activity of interconnected neurons by inhibition. As a result, the winner neuron becomes the sole source of output spikes. This selective mechanism enables noise filtering and input focus on the critical data.

We present a way that enables interaction among the excitatory neurons to prevent each other from firing, fundamentally implementing the WTA principle. The firing information is obtained from an inductance at the output *Lload*, next to the JJ, that determines the threshold operation. Note that the choice of inductance for coupling can be placed between the threshold junction *JJsoma1* and the ground node; however, this option requires a balance adjustment between the decay rate of the input side and lateral inhibition since such inductance is a part of the leaky storage loop of the soma circuit [23].

The interaction among neurons is established by inductive coupling K between an output inductance (Lload) and inductance in a feedback loop (Rwta1-Lwta1-Lwta2-Rwta2) shown in Fig. 7(a). The feedback loop consists of Lwta inductances for each neuron and resistors (Rwta) on each end. Therefore, any activation on a threshold junction will result in a change of current within this loop. Due to the coupling, the current on the feedback loop will affect the other neurons via coupled inductances. Each input spike creates a current determined by a resistor within the leaky storage loop of superconductor LIF neurons. Unlike this input, the inhibition current from a neuron to other neurons mainly depends on the value of the coupling K. Therefore, high inductive coupling results in a higher lateral suppression by a neuron. The JoSIM results of the proposed method with an example of soma circuits are shown in Fig. 7.

In our testbench, two soma circuits are designed to have a threshold of 2 spikes and $\approx$ 50 ps decay time. Initially, these circuits receive spikes with 75 ps time difference. Due to the current decay within the spike-storing loop on the input side of the soma, no output is generated. Next, two spikes with a short time interval are applied to the soma circuits. The first soma receives its inputs before the inputs of the second soma. As a result, the first soma fires earlier than the second one. The late-firing soma is expected to generate an output when no interaction exists among the somas. However, if the WTA mechanism is employed, the first soma inhibits the second soma. Note that there will be no lateral inhibition if the spike frames between the soma circuits do not overlap.

IV. Results

In our work, we utilized the BindsNET framework for high-level network modeling. We primarily focused on the intrinsic network properties and performance. We targeted the digits 0 and 1 of the MNIST dataset to evaluate the on-chip training network. The MNIST images are separated into two groups: the training set and the testing set. In our implementation, the image count was 633 for the training and 105 for the testing images of 0 s and 1 s, corresponding to 5% of the overall digit images of 0 s and 1 s within the dataset. Images were randomly selected, and each image was down-scaled from 28 by 28 pixels to 14 by 14 pixels. The training epoch count was assigned as 1, and each image was shown for 100 ps to the network during both the training and inference phases. For the inference phase, the network preserves the most up-to-date weight values while introducing the network to randomly selected test images that have not been shown before.

The neuron resting state was 0, and their neuron threshold was set to eight spikes. While threshold decay to the resting state was active, the dynamic threshold increment from the post-synaptic neuron spike was set to 32 inhibitory spikes. In our network structure, this value prevented the firing neurons from entering burst mode and gave other neurons a chance to fire. The membrane decay time of neurons was set to 25 ps.

For the synaptic characteristics, the increment and decrement in weight adjustments from the STDP mechanism were assigned as two spikes and one spike, respectively. Therefore, the spike generation on the increment side of the proposed STDP engine required a modification to have twice as much impact as the decrement side. For the spike activity of pre- and post-synaptic neurons, we assigned 10 ps to the spike-trace decay rate τ, keeping it within a reasonable time frame. Two network architectures

were considered, including architectures with 4 and 9 excitatory neurons with a weight resolution of 4 bits, corresponding to 16 different synaptic stages.

To eliminate the need for additional peripheral circuitry, a separate bias line can be implemented in the quantized STDP engine. While the network training involves applying this dedicated bias of the engine, the absence of bias during inference phase prevents the generation of both increment and decrement pulses for the synapse adjustment, establishing the distinction between the training and inference stages. Consequently, the synapse values, with multi-flux storing characteristics in the NDROs [21], can be maintained by triggering arbitrary pulses only from the input images. This approach not only saves static power during the inference phase but also ensures uniformity in component usage throughout the entire computational process within the architecture given in Fig. 1.

Unlike the conventional implementations, we did not use any weight normalization technique in the on-chip training setting. Accessing all weights to perform such an operation is not hardware-friendly, even though weight normalization gives all neurons a chance to fire and offers a better weight convergence. The dynamic threshold adaptation with decaying fashion temporarily compensates for the weight normalization. Some neurons can rapidly go into a burst mode without this feature and become dominant. However, when the decay time is relatively short compared to the duration of multiple input images, the neuron tends to forget the learned pattern, leading to an increase in the majority of its weights. On the other hand, we kept a weight clipping operation to limit the weight between 0 and 1 (scaled from the range of 0–15). Implementing Multi-flux NDROs in parallel, with a critical margin of $\pm 10\%$ as reported in [21], enables to achieve the desired weight range. With consistent and reliable fabrication across the chip for MJJs, it is possible to achieve a resolution higher than the currently employed value.

The total resource count of the circuit is determined for the downscaled images. Although the size of the input and dataset affects accuracy, the fundamental design approach of the hardware in the neural network implementation remains the same. When considering the entire MNIST dataset, accommodating the complete set requires adjustment in the hardware resource, impacting the number of neurons while the overall network model remains consistent for capturing additional digits. Each PTL is counted to establish the connections from input to synapse, from synapse to soma, and from the SPL tree of soma (LIF neuron) back to the STDP engine. Each neuron requires synaptic connections established by Multi-fluxon NDROs and the corresponding STDP engine. The overall resources required per neuron would approximately be 468 PTL Drivers and Receivers, 312 JTLs, 624 SPLs with a fanout of 2, 121 SPLs with a fanout of 3, 312 NDROs, and 624 Multi-flux NDROs. Consequently, it corresponds to 23.4 k JJs per neuron for the network computation, including the peripherals and interconnects.

In the case of training first architecture (4 excitatory neurons), we acquired an accuracy of 90.32% for training and 81.9% for testing. During the convergence, we observed that the neurons representing digit 1 still have traces of digit 0 and vice versa due to the bit resolution and no weight normalization. We also observed cases where some neurons go into a burst mode due to the random weight initialization. Therefore, a mechanism to thoroughly address this issue in lower-bit resolutions must be investigated. The 2D weight values are illustrated in Fig. 8 for each neuron of the considered architecture. The displayed weight values, arranged from left to right, represent snapshots taken at every one-third interval of the training process, spanning from iteration 0 to 633.

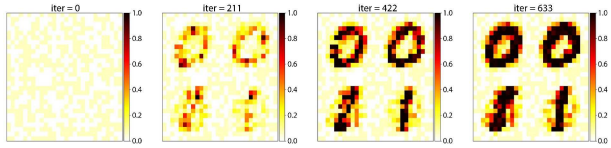


Fig. 8. Network training result using four excitatory neurons.

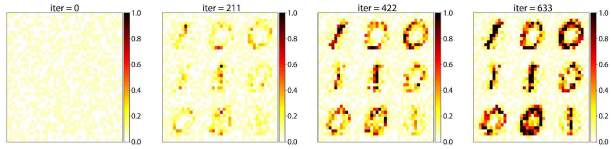


Fig. 9. Network training result using nine excitatory neurons.

The training settings are kept the same for the second network architecture (with nine neurons) to observe the impact of neuron count on the performance. The results showed an accuracy of 96.77% for training and 97.1% for testing, indicating a performance improvement over the case with four neurons. Therefore, increasing the number of neurons in the network positively influences the overall accuracy with a trade-off of hardware resources, supporting the motivation for large-scale implementations. The 2D weight values of this architecture during the training are illustrated in Fig. 9

V. Conclusion

This paper explored the capabilities of an on-chip training mechanism on superconductor spiking neural networks. We designed a leaky NDRO circuit and simulated its behavior with JoSIM. The leaky NDROs record spike traces to achieve a quantized STDP mechanism. Furthermore, we demonstrate a self-inhibition method for superconductor-based structures to establish the dynamic threshold behavior in LIF neurons. We also implement a superconductor winner-take-all mechanism to support the correct network behavior. The on-chip training capabilities are shown with a computational BindsNET framework, and we achieved $\approx$97% accuracy with 9 neurons for the classification of digits 0 and 1. These findings collectively highlight the promise of on-chip training in superconductor-based spiking neural networks.

References

[1] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, Nov. 2019.

[2] K. K. Likharev and V. K. Semenov, "RSFQ logic/memory family: A new josephson-junction technology for sub-terahertz-clock-frequency digital systems," *IEEE Trans. Appl. Supercond.*, vol. 1, no. 1, pp. 3–28, Mar. 1991.
[3] S. Razmkhah and P. Febvre, "Superconducting quantum electronics," in *Beyond-CMOS*. Berlin, Germany: Springer, 2023, pp. 295–391.
[4] M. Schneider, E. Toomey, G. Rowlands, J. Shainline, P. Tschirhart, and K. Segall, "SuperMind: A survey of the potential of superconducting electronics for neuromorphic computing," *Supercond. Sci. Technol.*, vol. 35, no. 5, Mar. 2022, Art. no. 053001.
[5] M. L. Schneider et al., "Energy-efficient single-flux-quantum based neuromorphic computing," in *Proc. IEEE Int. Conf. Rebooting Comput.*, 2017, pp. 1–4.
[6] A. Bozbey et al., "Single flux quantum based ultrahigh speed spiking neuromorphic processor architecture," 2020, *arXiv:1812.10354*.
[7] H. Zhang, C. Gang, C. Xu, G. Gong, and H. Lu, "Brain-inspired spiking neural network using superconducting devices," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 1, pp. 271–277, Feb. 2023.
[8] K. Segall, C. Purmessur and A. D'Addario, and D. Schult, "A superconducting synapse exhibiting spike-timing dependent plasticity," *Appl. Phys. Lett.*, vol. 122, no. 24, 2023, Art. no. 242601.
[9] J. A Delport, K. Jackman, P. I. Roux, and C. J. Fourie, "JoSIM–Superconductor SPICE Simulator," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, Aug. 2019, Art. no. 1300905.
[10] H. Hazan et al., "BindsNET: A. machine learning-oriented spiking neural networks library in python," *Front. Neuroinform.*, vol. 12, 2018, Art. no. 89.
[11] P. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Front. Comput. Neurosci.*, vol. 9, 2015, Art. no. 99.
[12] A. Paszke et al., "Automatic differentiation in PyTorch," in *Proc. NIPS Workshop Autodiff*, Long Beach, CA, USA, 2017. [Online]. Available: https://openreview.net/forum?id=BJJsrmfCZ
[13] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: tensorflow.org
[14] L. A. Plana et al., "SpiNNaker: Design and implementation of a GALS multicore system-on-chip," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 7, no. 4, pp. 1–18, 2011.
[15] W. Guo, M. E. Fouda, A. M. Eltawil, and K. Salama Nabil, "Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems," *Front. Neurosci.*, vol. 15, 2021, Art. no. 638474.
[16] M. Oster, R. Douglas, and S.-C. Liu, "Computation with spikes in a winner-take-all network," *Neural Computation*, vol. 21, no. 9, pp. 2437–2465, 2009.
[17] L. Abbott and S. Song, "Temporally asymmetric Hebbian learning, spike timing and neural response variability," in *Proc. Adv. Neural Inf. Process. Syst.*, 1998, vol. 11, pp. 69–75.
[18] H. Shouval, S. Wang, and G. Wittenberg, "Spike timing dependent plasticity: A consequence of more fundamental learning rules," *Front. Comput. Neurosci.*, vol. 4, 2010, Art. no. 19.
[19] , M. Diesmann, and W. Gerstner, "Phenomenological models of synaptic plasticity based on spike timing," *Biol. Cybern.*, vol. 98, no. 6, pp. 459–478, May 2008.
[20] Y. Liu and Xiao Jing Wang, "Spike-frequency adaptation of a generalized leaky integrate-and-fire model neuron," *J. Comput. Neurosci.*, vol. 10, no. 1, pp. 25–45, Jan. 2001.
[21] B. Z. Ucpinar, Y. Kopur, M. A. Karamuftuoglu, S. Razmkhah, and M. Pedram, "Design of a superconducting multiflux non-destructive readout memory unit," 2023, *arXiv:2309.14613*.
[22] A. Shaban, S. B. Sukruth, and M. Suri, "An adaptive threshold neuron for recurrent spiking neural networks with nanodevice hardware implementation," *Nature Commun.*, vol. 12, no. 1, Jul. 2021, Art. no. 4234.
[23] M. A. Karamuftuoglu and M. Pedram, "α-Soma: Single flux quantum threshold cell for spiking neural network implementations," *IEEE Trans. Appl. Supercond.*, vol. 33, no. 5, Aug. 2023, Art. no. 1801005.
[24] K. Sidiropoulou, E. Pissadaki, and P. Poirazi, "Inside the brain of a neuron," *EMBO Rep.*, vol. 7, pp. 886–892, 2006.
[25] M. L. Schneider and K. Segall, "Fan-out and fan-in properties of superconducting neuromorphic circuits," *J. Appl. Phys.*, vol. 128, no. 21, 2020, Art. no. 214903.