SUSHI: Ultra-High-Speed and Ultra-Low-Power Neuromorphic Chip Using Superconducting Single-Flux-Quantum Circuits

Zeshi Liu
State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Science
School of Computer Science and Technology, University of Chinese Academy of Sciences
Beijing, China
liuzeshi@ict.ac.cn

Shuo Chen
State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Science
School of Computer Science and Technology, University of Chinese Academy of Sciences
Beijing, China
chenshuo20s@ict.ac.cn

Peiyao Qu
State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Science
School of Computer Science and Technology, University of Chinese Academy of Sciences
Beijing, China
qupeiyao@ict.ac.cn

Huanli Liu
Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Science
Shanghai, China
liuhuanli@mail.sim.ac.cn

Minghui Niu
Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Science
Shanghai, China
niumh@mail.sim.ac.cn

Liliang Ying
Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Science
Shanghai, China
llying@mail.sim.ac.cn

Jie Ren
Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Science
Shanghai, China
jieren@mail.sim.ac.cn

Guangming Tang*
State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Science
Beijing, China
tangguangming@ict.ac.cn

Haihang You*
State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Science
Zhongguancun Laboratory
Beijing, China
youhaihang@ict.ac.cn

ABSTRACT

The rapid single-flux-quantum (RSFQ) superconducting technology is highly promising due to its ultra-high-speed computation with ultra-low-power consumption, making it an ideal solution for the post-Moore era. In superconducting technology, information is encoded and processed based on pulses that resemble the neuronal pulses present in biological neural systems. This has led to a growing research focus on implementing neuromorphic processing using superconducting technology. However, current research on superconducting neuromorphic processing does not fully leverage the advantages of superconducting circuits due to incomplete neuromorphic design and approach. Although they have demonstrated the benefits of using superconducting technology for neuromorphic hardware, their designs are mostly incomplete, with only a few components validated, or based solely on simulation. This paper presents SUSHI (**S**uperconducting ne**U**romorphic proce**S**sing c**HI**p) to fully leverage the potential of superconducting neuromorphic processing. Based on three guiding principles and our architectural and methodological designs, we address existing challenges and enables the design of verifiable and fabricable superconducting neuromorphic chips. We fabricate and verify a chip of SUSHI using superconducting circuit technology. Successfully obtaining the correct inference results of a complete neural network on the chip, this is the first instance of neural networks being completely executed on a superconducting chip to the best of our knowledge. Our evaluation shows that using approximately 10^5 Josephson junctions, SUSHI achieves a peak neuromorphic processing performance of 1,355 giga-synaptic operations per second (GSOPS) and a power efficiency of 32,366 GSOPS per Watt (GSOPS/W). This power efficiency outperforms the state-of-the-art neuromorphic chips TrueNorth and Tianjic by 81 and 50 times, respectively.

CCS CONCEPTS

• **Hardware → Superconducting circuits**.

KEYWORDS

Superconducting, Single-Flux-Quantum, Neuromorphic, Spiking Neural Networks

ACM Reference Format:
Zeshi Liu, Shuo Chen, Peiyao Qu, Huanli Liu, Minghui Niu, Liliang Ying, Jie Ren, Guangming Tang, and Haihang You. 2023. SUSHI: Ultra-High-Speed

*Corresponding author.

MICRO '23, October 28–November 01, 2023, Toronto, ON, Canada
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0329-4/23/10.
https://doi.org/10.1145/3613424.3623787

and Ultra-Low-Power Neuromorphic Chip Using Superconducting Single-Flux-Quantum Circuits. In *56th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '23), October 28–November 01, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3613424.3623787

1 INTRODUCTION

In the post-Moore era, improving the performance of computer systems while meeting energy budget requirements has become an increasingly challenging task. Fortunately, emerging device technologies such as photonic, quantum, biological, and neuromorphic computing show great potential in terms of operational speed and energy efficiency. As such, exploring these technologies and enhancing their feasibility is necessary as alternative candidate technologies for the traditional semiconductor technology.

Among these technologies, the rapid single-flux quantum (RSFQ) [21] based superconductor circuit technology has emerged as one of the most attractive alternatives due to its ultra-low latency of ~$10^{-12}s$ and energy consumption of ~$10^{-19}J$ to complete a state flipping [9, 15]. Numerous research efforts have been devoted towards promoting the RSFQ-based superconductor circuit technology across various aspects [17, 24, 38, 44]. As a result, RSFQ technology has gained significant attention as an interesting and promising post-Moore solution.

Neuromorphic circuits based on superconducting technology represent an application that can effectively exploit the properties of RSFQ technology. This is possible since the information in RSFQ logic is encoded and processed via the single-flux quantum (SFQ) pulse, which closely resembles the neuronal pulse present in biological neural systems [39]. Indeed, the superconducting chip is considered a highly promising hardware platform for the fabrication of naturally neuromorphic hardware, and researchers have devoted increasing efforts towards implementing neuromorphic computing on superconducting circuits [13, 42].

However, current research on neuromorphic processing using superconducting circuits faces several challenges. These issues hinder researchers from fully leveraging the benefits of superconducting circuits in implementing neuromorphic processing and, in some cases even exceed the current technological level of superconducting circuits, making them difficult to be effectively applied.

Firstly, the use of pulse-driven synchronous timing and storage renders their design not truly neuromorphic in processing. Furthermore, given the ultra-low flip time of superconducting cells, synchronous timing in superconducting circuits frequently necessitates aligning pulses by extending the length of transmission lines. This consequently incurs a significant wiring overhead. Additionally, deficiency in high-frequency memory poses challenges in designing storage on superconducting circuits.

Secondly, most existing work does not meet the constraints of superconducting circuits, especially in terms of integration. The superconducting circuit technology is still in its infancy, integrating such elements on a single chip remains significantly challenging. It is indeed difficult to fabricate and verify circuit designs that exceed current integration constraints.

Thirdly, the existing designs for neuromorphic processing are mostly incomplete, with validation limited to only certain components or reliance on simulation results. So far, no superconducting neuromorphic chips capable of performing complete neural network inference have been successfully validated.

In this paper, we present the SUSHI (**S**uperconducting ne**U**romorphic proce**S**sing c**HI**p) to explore the potential of neuromorphic processing on superconducting circuits. To overcome existing challenges and design verifiable and fabricable superconducting neuromorphic chips that meet the constraints of superconducting circuits, we present SUSHI based on three guiding principles: 1) Focusing exclusively on neuromorphic processing features, such as asynchronous processing and pulse-driven mechanisms; 2) Conforming to superconducting circuit constraints, which include limiting memory, integration, and timing constraints; and 3) Implementing a complete on-chip network structure with the ability to perform spiking neural network (SNN) processing.

To achieve these objectives and fully explore the potential of superconducting neuromorphic processing, we propose several design solutions for SUSHI, including:

First, we propose the design of a superconducting neuromorphic element (NPE) that is built on the principle of neuromorphic processing. Our NPE is pulse-driven and can process pulse-encoded information asynchronously. This asynchronous feature provides significant savings on wiring overhead, which accounts for up to 80% of the design resource footprint in synchronous timing designs. Furthermore, our NPE leverages the state flipping of superconducting cells to accomplish the storage and switching of neuron states, which essentially eliminates most of the memory requirements. This low memory requirement considerably reduces the impact of the memory wall on the NPE, enabling our design to fully implement neuromorphic processing.

Second, we introduce superconducting spiking neural network (SSNN) that aligns with the constraints of our NPE design on superconducting circuits. We simplify the weight processing to be directly based on the pulse and proposed a stateless neuron model that eliminates the challenges in meeting the superconducting circuit constraint for membrane potential processing. Moreover, we design an asynchronous neuron timing that corresponds with the proposed model. Our pulse-based asynchronous timing enables the neuron model to process the input and obtain the output asynchronously, ensuring compatibility with the superconducting circuit.

Third, we design the complete on-chip network structure, enabling processing of arbitrary topologies among NPEs, making on-chip neural network processing possible. We then introduce a bit-slice SSNN method to perform complete neural network inference on our on-chip network. Unlike traditional multiplexing, our bit-slice SSNN method leverages the state-preserving ability of superconducting cells to decompose the SNN processing process into batches, without the need for additional registers. Utilizing our bit-slice SSNN method, our superconducting neuromorphic circuit fits better with the restrictions of current superconducting technology.

With our design, we completely implement SUSHI on the superconducting circuit. We have fabricated and verified a superconducting neuromorphic chip of SUSHI and successfully obtained correct inference results from a complete spiking neural network on the chip. To the best of our knowledge, this is the first time that a neural network is completely performed on a superconducting chip.

Our evaluation results demonstrate that, using roughly 10^5 Josephson junctions (the fundamental integrated unit in superconducting circuits), SUSHI achieves a peak performance of 1,355 giga-synaptic operations per second (GSOPS), which is 23 times that of TrueNorth [22]. Moreover, it achieves a power efficiency of 32,366 GSOPS per watt (GSOPS/W), which surpasses the power efficiency of TrueNorth [22] and Tianjic [26] (two state-of-the-art neuromorphic chips) by 81 and 50 times, respectively.

In summary, we present SUSHI to fully exploit the potential of superconducting neuromorphic processing, and our work makes the following contributions:

- **Architect the first fabricable superconducting neuromorphic chip:** Our architectural designs of SUSHI include the neuromorphic processing element and the on-chip network structure.
- **Neuromorphic methodologies for superconducting circuits:** Our methodologies enable neuromorphic processing of SUSHI within the constraints of superconducting circuits.
- **Fabrication and verification of SUSHI:** To the best of our knowledge, it is the first time that a neural network is completely performed on a superconducting chip.
- **Superior performance and power efficiency:** Our evaluation demonstrates that SUSHI can provide ultra-high-speed and ultra-low-power superconducting neuromorphic processing.

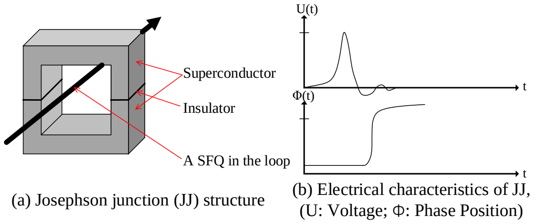
2 BACKGROUND AND PRELIMINARIES

2.1 RSFQ Superconducting Circuits

2.1.1 Principle and Logic Representation. Rapid single-flux quantum (RSFQ) technology is one of the leading superconducting computing technologies, renowned for its low power consumption and ultra-fast flipping characteristics. The fundamental component of the circuit is a superconducting loop (Fig. 1(a)), which comprises at least two Josephson junctions (JJs). A JJ consists of a superconducting metallic niobium (Nb) ring with an AlOx thickness ranging from 2-3 nm in the middle. When the magnitude of the induced current exceeds the JJ’s threshold current, Cooper pairs cross the AlOx, creating a loop current in the superconducting loop that produces a voltage of approximately 1 mV and an SFQ pulse that lasts around 1 ps (Fig. 1(b)). Unlike COMS circuits, the basic passive device of the RSFQ circuit is an inductor and not a capacitor.

In RSFQ circuits, information is carried and stored as a single flux quantum (SFQ) and transmitted in the form of SFQ pulses. In asynchronous design, the presence of an SFQ pulse on the transmission line denotes a logic value of "1", while the absence of an SFQ pulse denotes a logic value of "0", as shown in Fig. 2(a). In synchronous design, the presence of an SFQ pulse between two successive clock pulses denotes a logic value of "1", while its absence denotes a logic value of "0", as shown in Fig. 2(b). Typically, designers employ synchronous timing methods when designing RSFQ digital circuits.

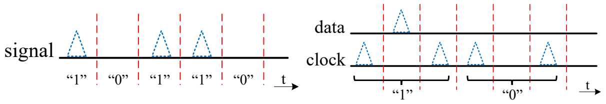
2.1.2 Common Cells. Cells in RSFQ circuits differ from those in CMOS circuits. Some of them are exclusive to the RSFQ circuit, while others possess comparable functions but slightly different implementation methods [4]. The following sections outline some of the typical cells that we employ in our design. Table 1 lists the RSFQ cell constraints we use, and we employ larger interval constraints to ensure the correct operation of the cells.



(a) Josephson junction (JJ) structure

(b) Electrical characteristics of JJ, (U: Voltage; Φ: Phase Position)

Figure 1: The superconducting loop.



(a) asynchronous circuit

(b) synchronous circuit

Figure 2: Logic representation in RSFQ circuit.

CB		SPL		NDRO	
dinA/B-dinA/B	19.9	din-din	19.9	din/rst-rst/din	39.9
dinA/B-dinB/A	5.7	TFF		clk-clk	39.9
DFF		din-din	19.9	din-clk	14.81
din-clk	8.53	JTL		rst-clk	16.61
clk-clk	19.9	din-din	19.9		

Table 1: Constraint for RSFQ cells. "A-B" is the time (ps) that the B channel input must lag behind the A channel input.

A. DFF

A D flip-flop (DFF) is the most fundamental storage cell in RSFQ circuits, and is commonly utilized for synchronizing data in synchronous RSFQ digital circuit design. Fig. 3(a)(e) gives the symbol and timing diagram of DFF. The *din* port, the *clk* port, and the *dout* port respectively represent the data input, the clock input, and the data output of DFF. The clock input is used for releasing any data stored in the DFF. An SFQ pulse appears at the *dout* port when and only when two pulses are input to both *din* and *clk* ports. However, the sequence in which the pulses reach the *din* and *clk* ports varies depending on the chosen timing method.

B. NDRO

Another commonly used storage cell is the non-destructive readout (NDRO) cell, which differs from the DFF in that the latter is a destructive read, and NDRO is non-destructive. Fig. 3(b)(f) gives the symbol and timing diagram of NDRO. The *rst* port represents the reset signal that clears the stored data in NDRO to reset it. Resetting the NDRO is necessary before new data is input, to ensure that it functions correctly.

C. SPL

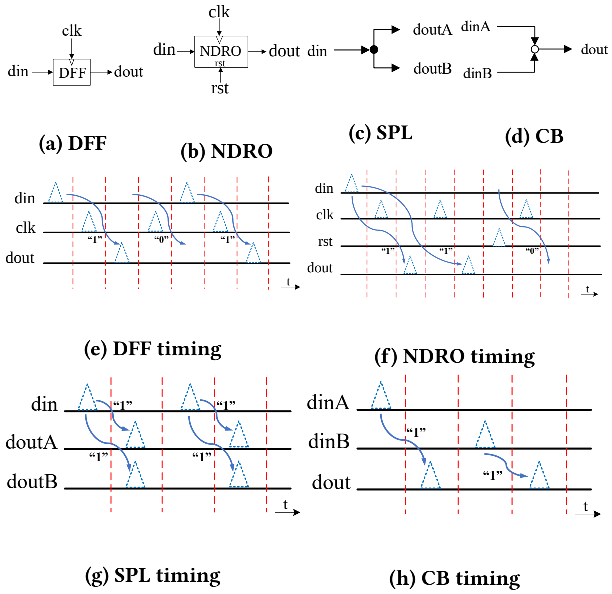
Compared to CMOS circuits, the splitter (SPL) is a distinctive cell in RSFQ circuits, primarily because the fan-out drive of RSFQ cells is restricted to a value of 1. When the fan-out demand exceeds 1, an SPL is necessary to divide the output of the cell into two or three outputs. Fig. 3(c)(g) shows the symbol and timing of an SPL2, which denotes 1-to-2 output division.

D. CB

The confluence buffer (CB) is another unique cell that is exclusive to RSFQ circuits. The CB merges two or three inputs into one output. In RSFQ synchronous circuits, only one pulse can be input into the CB during a clock cycle to ensure its proper functioning. Contrarily, an OR gate can operate normally even if there are two inputs during a clock cycle. In RSFQ asynchronous circuits, a minimum interval is needed between two input pulses, which varies depending on the fabrication technology. Fig. 3(d)(h) shows the symbol and timing of a CB2 that performs 2-to-1.

E. TFFR and TFFL

The toggle flip-flop (TFF) used in RSFQ circuits performs a function that is similar to the TFF in CMOS circuits. Both types of TFFs output a pulse for every two inputs received. However, in the RSFQ circuit, TFFL and TFFR are used to differentiate between TFFs that output pulses when state 0 is flipped to state 1 and TFFs that output pulses when state 1 is flipped to state 0.



(a) DFF (b) NDRO (c) SPL (d) CB

(e) DFF timing (f) NDRO timing

(g) SPL timing (h) CB timing

Figure 3: Common cells in RSFQ circuit.

2.2 Spiking Neural Networks (SNNs)

In recent years, SNNs have garnered increasing attention as the third generation of neural networks, as mentioned in several research works [29, 40, 45]. Their event-driven architecture and adaptability to run on neuromorphic chips [1, 8, 27] give SNNs an advantage over traditional Artificial Neural Networks (ANNs), resulting in significantly lower power consumption. As a result, SNNs have emerged as one of the most promising paradigms in the field of artificial intelligence.

The structure of SNNs includes two fundamental components: the neuron model and the network topology. The neuron model is the fundamental unit of SNNs and comprises three primary components: dendrites, soma, and axons. Dendrites collect signals from other neurons or inputs, while the soma maintains the membrane potential that is modulated by these signals. Once the membrane potential reaches a specific threshold, the neuron generates an output spike that propagates through the axons to neighboring neurons. Several neuron models have been proposed to model the membrane potential process, offering varying levels of biological plausibility and computational complexity, including the Hodgkin-Huxley (H-H) model[14] and the Integrate-and-Fire (I&F) model[23]. In practical applications, discrete versions of these neuron models are often employed for processing. The behavior of an individual neuron in SNNs can be represented by three discrete equations: charging, firing, and resetting.

$$H[t] = f(V[t-1], X[t]) \tag{1}$$

$$S[t] = \Theta(H[t] - V_{threshold}) \tag{2}$$

$$V[t] = H[t] \cdot (1 = S[t]) + V_{reset} \cdot S[t] \tag{3}$$

Equations (1), (2) and (3) describe the charging, firing, and resetting processes of a neuron. The variable $V_{threshold}$ represents the threshold required to trigger a spiking event, while $\Theta()$ is the Heaviside step function. The input and output spike sequences of the neuron are represented by $X[t]$ and $S[t]$, respectively, and can take binary values of 0 and 1. The neuron receives external inputs, and it evolves according to (1). When the membrane potential reaches the threshold, a spiking event occurs, and the membrane potential returns to its resting value, as represented by (3).

Various topological structures can be developed in SNNs by combining neurons and their connections. These structures include linear mapping layers, convolutional layers, and recurrent layers, which are similar to those used in ANNs. By combining these layers with pooling and regularization operations, several effective network topologies like AlexNet, VGG, and ResNet have been successfully implemented on common datasets such as Mnist, Cifar-10, and ImageNet[45].

3 MOTIVATION AND CHALLENGES

We are motivated to develop superconducting neuromorphic processing and our SUSHI, based on the challenges encountered in traditional superconducting circuit design, which include:

A. Difficulties in timing

As mentioned earlier, synchronous RSFQ designs require an extra clock distribution network, which consumes a significant portion of the wiring hardware resources in RSFQ circuit designs. Our design experience has shown that the wiring overhead for synchronous timing-based superconducting structures (e.g., ALU [2, 36], multiplier [19, 37], etc.) typically accounts for about 80% of the total design, as each synchronous cell requires a separate clocking line. This overhead can be primarily attributed to the use of Josephson transmission lines (JTL) in RSFQ circuits, which, being composed of JJs, lead to similar magnitudes of delay and area overhead for both wiring and functional cells. This is in contrast to CMOS circuit design, where this issue is not prevalent. Therefore, to design a superconducting neuromorphic unit that can be applied and verified on a real superconducting chip using current techniques, we need to minimize the hardware resource overhead for timing.

B. Taller Memory wall

The "memory wall" is a consistent and urgent issue in mature CMOS circuit design, and it is even more pronounced in RSFQ circuit design. In RSFQ circuits, shift registers made up of multiple DFFs in series are the most commonly used on-chip memory, leveraging the gate-level pipeline characteristics of DFF cells. However, shift registers are only suitable for sequential access, and achieving efficient random access is challenging. The use of shift registers as on-chip memory in SuperNPU[16] resulted in only 16% of its peak inference throughput during the inference process.

Several cryogenic memory technologies, such as Vortex Transition Memory (VTM) [31], Magnetic Memory (MRAM) [25], Superconducting Nanowire Memory (SNM) [5], and Josephson-CMOS SRAM [34], have been proposed to meet the low-temperature operation requirements and characteristics of RSFQ circuits. However, these memory technologies do not currently offer practical solutions for mitigating the "memory wall" issue in RSFQ circuit design. Hence, mitigating the performance degradation caused by the "memory wall" remains a significant challenge in designing RSFQ circuits.

C. Low integration

Current RSFQ fabrication technology makes it challenging to implement bit-parallel processing neural networks. Despite previous work demonstrating the advantages of implementing neuromorphic processing using superconducting circuits, they have only verified components of the design or relied on simulation results due to integration limits [3]. A significant gap still exists between superconducting circuit technology and traditional semiconductor technology, making conventional design methods potentially unsuitable for current superconducting circuits and difficult to verify practically. To design a fabricable superconducting neuromorphic chip, our approach must consider the limits of the integration. Additionally, our designs must be flexible in both architecture and methodology to accommodate different levels of superconducting circuit integration.

The challenges described above pose significant obstacles to applying traditional designs to current superconducting circuits, as traditional designs typically have higher computational and memory requirements compared to superconducting technology. Consequently, the main **goal of SUSHI** is to exploit the potential of neuromorphic processing on emerging superconducting technology. Our proposed superconducting neuromorphic chip is based on our superconducting SNN, allowing us to leverage the unique features of neuromorphic processing, such as asynchronous processing and pulse-drive. The chip is designed to conform to superconducting circuit constraints using our proposed methodologies, enabling us to perform neural networks on current superconducting circuits.

4 ARCHITECTURAL DESIGN

Our proposed design for SUSHI is broken down into two phases: the design of the neuromorphic processing element (NPE) and the network of NPEs, which are then combined to create an executable neuromorphic architecture.

4.1 Neuromorphic Processing Element (NPE)

As the fundamental element, the NPE must minimize the hardware resource requirement as much as possible, such as reducing the timing signal and memory requirement. Hence, the NPE design aims to leverage the characteristics of neuromorphic processing, such as asynchronous processing, and utilize the pulse-driven state flipping of the neuron to process information.

4.1.1 State controller. To prevent the additional overhead caused by synchronous timing design, we designed the asynchronous state controller as the minimal component of the NPE. The state controller leverages state holding and asynchronous state flipping instead of storage and computation when processing information based on the neuron model.

Fig. 4 provides an overview of the main functional structure of our state controller, which utilizes non-destructive readouts (NDROs) and T-flip-flops (TFFs) for the design. In the superconducting circuit, these two cells can operate asynchronously (with the *din* port of the NDRO serving as the *setup* port, the NDRO can be treated as a configurable switch). By taking advantage of the TFF's two distinct states (TFFL and TFFR), the state controller can asynchronously generate the pulses involved in state flipping. Additionally, through the use of the setup/rst control of the NDROs, we can maintain states and output a pulse when necessary.

Fig. 5 presents the state diagram of our state controller. Initially, the state controller is in state 0, and it flips to state 1 when given an in pulse. At that point, the flip pulse is generated based on the NDRO that is set. If NDRO0, which corresponds to the TFFL in the state controller, is set, a flip pulse will be output. Similarly, if the state controller is in state 1, the in pulse will flip it to state 0. At the same time, depending on whether NDRO1 is set or not, the flip pulse will be output; this time, the flip pulse is generated from TFFR.

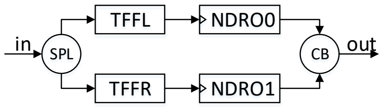


Figure 4: The main functional structure of the state controller (SC).

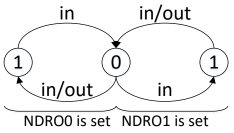


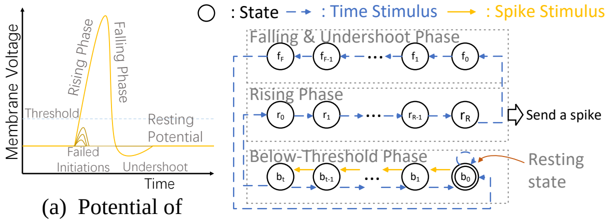
Figure 5: State diagram of SC.

4.1.2 Multi-State Processing Element. Our state controller (SC) makes it straightforward to construct arbitrary multi-state superconducting neuromorphic processing element. We can adaptively apply different numbers of SCs based on the complexity of the neural network, thereby constituting a sufficient number of neuron states and reducing resource overhead. Fig. 9 illustrates a multi-state processing element that comprises 10 SCs.

Using the multi-state neuromorphic processing unit, we can represent the states of the neuron model. Fig. 6(a) exhibits a biological neuron model that consists of four phases: resting, rising, falling, and undershoot. The state diagram depicting this neuron model is shown in Fig. 6(b). The corresponding state transition functions are given in Fig. 7. We employ the state series that are triggered by the time stimulus to represent the different phases of the neuron model.

By describing neuron models through state flipping, we can avoid using synchronous logic-based digital processing and memory in the design of neuromorphic processing elements. Consequently, this reduces the circuit resource overhead in designing our neuromorphic processing elements.

On the other hand, we conducted a quantitative analysis of SNNs and discovered that using at least ~500 states is adequate to model a neuron that can be utilized directly for SNNs inference. Moreover, the overhead of this scale falls within the acceptable limits of current superconducting circuit technology.

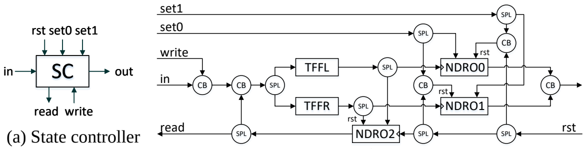


(a) Potential of biological neuron (b) Neuron model based on state transition

Figure 6: Overview of the neuron model.

$$
\begin{array}{ll}
\delta(b_0, spike) = b_1 & \delta(r_0, time) = \mathrm{r}_1 \\
\delta(b_1, spike) = b_2 & \delta(r_1, time) = r_2 \\
\ldots & \ldots \\
\delta(b_{threshold-1}, spike) = b_{threshold} & \delta(r_{R-2}, time) = r_{R-1} \\
\delta(b_0, time) = b_0 & \delta(r_{R-1}, time) = r_R, send\ a\ spike \\
\delta(b_1, time) = b_0 & \delta(r_R, time) = f_0 \\
\delta(b_2, time) = b_1 & \delta(f_0, time) = f_1 \\
\delta(b_3, time) = b_2 & \delta(f_1, time) = f_2 \\
\ldots & \ldots \\
\delta(b_{threshold-1}, time) = b_{threshold-2} & \delta(f_{F-1}, time) = f_F \\
\delta(b_{threshold}, time) = r_0 & \delta(f_F, time) = b_0
\end{array}
$$

Figure 7: State transition function of the neuron model.



(a) State controller symbol (b) Logic design of the state controller

Figure 8: Complete design of the state controller.

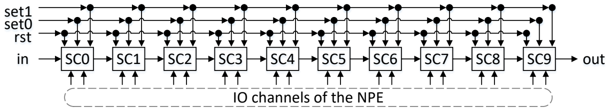


Figure 9: Complete design of the NPE with 10 SCs.

4.1.3 Overall Design of NPE. To better manage the state controller, we add additional channels for setting its behavior and state. Fig. 8(a) shows the symbol of the state controller. To avoid conflict between the outputs of NDRO0 and NDRO1, the use of *set0* and *set1* is not compatible, and any one will disable the other. We introduce an additional NDRO to monitor the state of the SC, which enables asynchronous reset, read, and write functionality for the SC. Overall, the resource overhead of implementing a single SC is minimal. Fig. 8(b) shows the logic design of the state controller in the superconducting circuit design. Fig. 9 shows a comprehensive NPE structure based on 10 state controllers. The state controllers are linked together via serial links, and their *rst*, *set0*, and *set1* can be arbitrarily bound together for ease of use, while *read* and *write* must be set up individually.

4.2 System Architecture

The NPEs eliminate the need for extra timing logic, which enables us to fully utilize the on-chip area for neuromorphic processing. To create the SUSHI architecture, we also need to design the network of NPEs, which should reflect the impact of the weights on the NPEs. We also consider the resources and area cost to design the corresponding structure. Finally, we provide an overview of the system architecture of SUSHI.

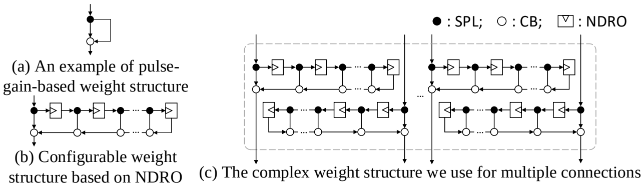
4.2.1 Weight Structure. In conventional neural networks, weights are typically stored and processed using numerical values, but implementing this in superconducting circuits requires designing suitable processing units and storage structures, which can introduce significant design overhead. Consequently, we propose using the number of pulses to encode the weights instead.

To alter the number of pulses in a superconducting circuit, the use of SPL and CB cells is required. A simple pulse gain structure is given in Fig. 10(a). When a pulse passes through the pulse-gain-based weight structure, it splits into two pulses via the SPL cell. One pulse continues along the original line, while the other pulse converges on the same line through a section of JTL transmission lines with an added delay, resulting in two pulses in the original line.

In the weight structure, we leverage the configurability of the NDRO cell to adjust the strength of the weight influence to match the quantified weight value and to enhance the network complexity, as illustrated in Fig. 10(b). The weight structures operate asynchronously, and once configured through the din/rst channels, they can directly impact the state of the target neuron with varying strengths. To ensure proper processing of the weights, we regulate the pulse interval during input creation based on the cell constraints (Table 1).

In our design, we use a more intricate pulse-gain-based weight structure for weight processing. As shown in Fig. 10(c), the structure comprises multiple gain loops that can expand a single pulse into several pulses, which can have varying effects on the state of the target neuron.

In traditional neural networks, the weights are frequently stored as float or int types. Therefore, they must be quantified for use in our weight structure. In SUSHI, the weight structure only influences the strength of the pulse impact. The polarity of the weights is only distinguished when the weights reach the neuron, through the *set* channels in Fig. 8. Additionally, the quantization method assists us in controlling the total number of pulses. As described in Section 5.1.



(a) An example of pulse-gain-based weight structure

(b) Configurable weight structure based on NDRO

(c) The complex weight structure we use for multiple connections

Figure 10: Overview of weight structures.

4.2.2 *On-chip Network of NPEs.* Designing an on-chip network for NPE interconnects presents two challenges. First, the transmission line in the superconducting circuit has a significant area footprint, and the transmission line crossing overhead is high (twice the width of the original transmission line), resulting in a considerable overhead of NPE interconnects. The second challenge is that when operating in a high-frequency environment, the delay of long-distance pulse transmission makes it difficult to provide timely feedback to the input. Thus designing a complex feedback structure for superconducting circuits can be challenging.

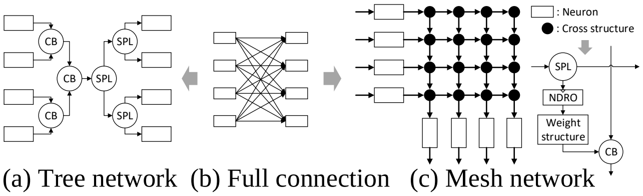
To design an NPE network with low resource costs, the connection between NPEs needs to take advantage of their on-chip position. We consider two structures for implementing on-chip NPE networks: a tree structure and a mesh structure (Fig. 11). Both structures aim to minimize transmission line crossings while keeping the pulse transmission distance as short as possible.

The main difference between these two structures is that the tree network maximizes the utilization of SPL and CB cells in superconducting circuits, reduces the length and crossings of transmission lines, and saves design area by allowing flexible placement of NPEs. However, the tree network can only make simple distinctions of normalized weights and cannot be applied to build arbitrary connections. In contrast, the mesh network is capable of distinguishing the weights between any pair of NPEs. Moreover, the NDRO cell can be used to design a configurable structure in the mesh network, enabling the implementation of arbitrary connections between NPEs. However, the mesh network requires a cross structure between each pair of NPEs, resulting in more transmission line crossings. Therefore, its resource footprint increases rapidly as the design scale becomes larger.

Based on SUSHI’s design, the weight structures need to be reloaded with the predetermined values from the model before inputting the data to neurons. However, this can lead to decreased overall performance due to repeated reloading when different batches of input data have varying weights. In SUSHI, weight reloading is efficiently managed through the use of NDROs as switches. This allows weight reloading to be conducted in parallel per synapses (as illustrated in Fig. 12(e)(g)). At each synapse, the weights are reloaded independently, and this reloading process does not impact the critical path of neural network inference. Consequently, the overhead associated with weight reloading is solely determined by the time it takes to reach the NDRO and is separate from the inference process. As a result, as the network scales, there are no additional costs incurred for weight reloading. Furthermore, by reducing the frequency of weight reloading, we can effectively reduce its performance impact.

To minimize the frequency of weight reloading, we introduce a strategy that involves the reordering and bucketing of synapses (described in Section 5.1). For each time step, when multiple pulses are sent to a neuron, we use bucketing to ensure that pulses within the same batch have the same polarity. Then, through reordering, we enable inputs from adjacent batches that pass through the same cross structure to share the same weight strength. With this combination, weight reloading only occurs between buckets with different attributes and on the remaining small fraction of weights that cannot be adjusted. This approach has little impact on inference results, and the frequency of weight reloading can be significantly reduced.

We analyze the inference processes following optimization and find that the utilization of synapse reordering and bucketing has a negligible impact of less than 1% on accuracy. Besides, we evaluate the delays encountered by weight control pulses in reaching NDRO per synapse at various scales. We also examine the frequency of weight reloading required after optimization. Our results indicate that the optimized weight reloading accounts for 20% of the total inference time on average. In practice, the most suitable network structure to be applied should be evaluated based on the specific scenario, the complexity of the network, and the amount of resources available.



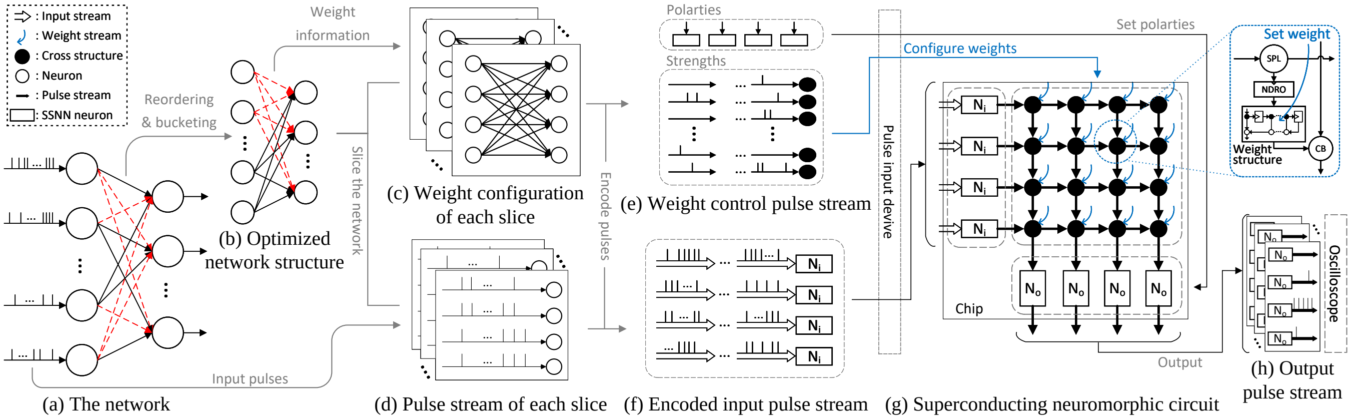
(a) Tree network (b) Full connection (c) Mesh network

Figure 11: Two structures for implementing NPE network on superconducting circuits.

4.2.3 *Overall System Architecture.* Fig. 12 illustrates the overall structure and workflow of our design. A complete SNN inference on SUSHI includes two phases. The first phase only executes once off-chip for the trained network. Based on the constraints (Table 1) and the optimized synaptic order (Section 5.1), we encode the channels and input times of weight and input pulses. In the second phase, the pulse streams are fed into the chip.

Table 2 provides information on the circuit resource overhead related to the mesh network in Fig. 12(g) (based on the standard cell library of SIMIT-Nb03 [12]). The design of a 4 × 4 mesh network requires a total of 45,542 JJs and occupies 44.73 mm^2 of circuit area. Of this, 31,026 JJs (68% of the total) account for the wiring overhead, and 14,516 JJs (32% of the total) account for the logic overhead. The wiring overhead in our design is significantly reduced compared to typical superconducting designs, which often exceed 80%.

We apply neural networks to the circuit using the bit-slice SSNN method. As introduced in Section 5.3, bit-slice method controls the channel and timing of the inputs by encoding the synapses after the network is trained. This process is incorporated into the encoding phase, and enabling us to feed the neural network in blocks without relying on memory or feedback structures. By leveraging the state-preserving capability of the superconducting cells, we



(a) The network
(b) Optimized network structure
(c) Weight configuration of each slice
(d) Pulse stream of each slice
(e) Weight control pulse stream
(f) Encoded input pulse stream
(g) Superconducting neuromorphic circuit
(h) Output pulse stream

Figure 12: The overall workflow of neuromorphic processing on SUSHI.

can retain processed information and move on to the next block of computation.

The weight configuration (Fig. 12(c)(e)) consists of strength (for weight structures) and polarity (for neurons), which appear zero or once before each input block. We optimize the synaptic order to minimize the need for reloading (Section 4.2.2). After configuration, pulse streams encoded in block order (Fig. 12(d)(f)) can be fed into the chip to directly generate the corresponding neural network output (Fig. 12(g)(h)). Notably, the architectural design in Fig. 12 is scalable, with the circuit scale further compressible or expandable based on the level of superconducting circuit technology.

total JJs	45,542	**wiring JJs**	31,026 (68.13%)
total area	44.73 mm^2	**logic JJs**	14,516 (31.87%)

Table 2: Resource overhead of a 4×4 mesh network of NPEs.

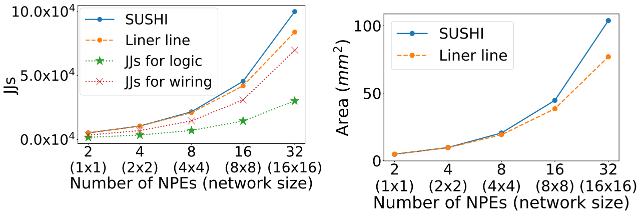
4.3 Resource Overhead Analysis

To demonstrate that SUSHI is fabricable and verifiable, it is necessary to analyze its resource overhead on superconducting circuit technology. In this context, fabricability is evaluated in terms of the number of JJs and the chip area required. As discussed earlier, the number of JJs and the chip area of SUSHI are determined by the number of NPEs and the size of the corresponding network. Thus, the relationship between these parameters is an important metric for evaluating the feasibility of fabricating a superconducting chip.

To analyze the feasibility of fabricating SUSHI, we examine the relationship between the number of JJs and circuit area when different scales of NPEs and corresponding networks are employed, as illustrated in Fig. 13. Fig. 13(a) shows the growth of the number of JJs with the number of NPEs, while Fig. 13(b) shows the corresponding growth of circuit area. The dotted line in the figure represents the linear growth for reference.

In superconducting circuit design, larger designs often exhibit higher wiring overhead due to delays and increased transmission line area. This can significantly hinder the fabricability of large designs with excessive wiring overhead. Fig. 13 shows that the resource overhead of SUSHI design only slightly exceeds the linear reference line, and the increase in wiring overhead remains within acceptable limits with increasing NPEs. This favorable resource scaling is due to our asynchronous design, which permits direct connections between different circuit elements without requiring strict timing alignment. As a result, wiring overhead is significantly reduced.

Our result shows that the resource overhead of SUSHI remains within acceptable limits, indicating that it is capable of being fabricated and validated. Furthermore, the growth trend of resource overhead for our design tracks the linear reference line, allowing the design to flexibly adapt to different levels of superconducting circuit integration constraints by adjusting the number of NPEs and corresponding network size on-chip.



(a) JJs, JJs for logic, and JJs for wiring of SUSHI.

(b) Area of SUSHI.

Figure 13: Resource overhead of SUSHI with different number (network size) of NPEs.

5 METHODOLOGY

Our methodologies for maximizing the potential of neuromorphic processing on SUSHI include several key approaches. Firstly, we introduce superconducting SNN (SSNN) which improves the neuron model and weights processing for SUSHI. Secondly, we utilize an asynchronous neuron timing method specifically tailored for superconducting neuromorphic chips. Thirdly, we propose the bit-slice SSNN method for further optimization of SSNN on the superconducting neuromorphic chip. Currently, our research is focused on the inference phase of SNNs, with the aim of demonstrating the potential of superconducting neuromorphic circuit.

5.1 Superconducting SNN (SSNN)

In order to address challenges associated with processing on NPEs due to process restrictions and structural characteristics, we propose superconducting SNN (SSNN), which includes the following aspects: (1) The design of the NPE only allows for 1-bit pulse-driven processing. Thus, in practice, it is necessary to map SNNs into binary networks. (2) Using on-chip storage as traditional structures is expensive in superconducting circuits. Therefore, neuron model of SSNN is based on pulse streams for information processing. (3) The neuron models in our SSNN need to adapt to pulse-based asynchronous excitation. (4) To minimize the number of states required by the neuron, we recode the pulse stream to match its possible upper and lower bounds.

To address the first challenge, SSNN employs the XNOR-Net algorithm [28] to binarize the network. This involves mapping the original model onto a binary network with {-1, 1} values. In practice, SSNN replaces floating-point arithmetic operations with single-bit arithmetic operations multiplied by scaling parameters. To eliminate the overhead of weight processing, we normalize the weights to scaling parameters and process them during thresholding while training the network. Consequently, in SSNN, the weight multiplication is divided into two phases: scaling the 1-bit input pulse based on the weight structure, and applying the effect of weight polarity to the neuron through the neuron setting.

To address the second challenge, we design a stateless neuron model that eliminates the storage overhead associated with the temporary storage of neuron membrane potentials. Our forward process is fully stream-based. The charge equation of the stateless neuron is based on the accumulation of input pulses and the corresponding synaptic weight. We also simplify the reset procedure by resetting the membrane potential to zero at the end of each time step. After implementing our improvements, the neuron model is superconducting-circuit-friendly, and in our simulation of the superconducting circuit, the performance of SSNN was significantly improved while maintaining correct results.

To mitigate accuracy drops caused by challenges 3 and 4, we propose a synapse bucketing and reordering algorithm. To prevent premature or incorrect firing caused by the membrane potential exceeding the threshold, we traverse all inhibitory synapse connections (synaptic weight of -1) first to obtain the minimum membrane potential value. We then traverse all excitatory connections (synaptic weight of 1) to ensure that possible firing spikes appear last and find the maximum required range of states to use. This approach could lead to an overflow of the lower number of states, resulting in an inference error. Therefore, we introduce the bucketing idea by encoding a certain number of inhibitory and excitatory synapses in a bucket and traversing them according to the bucket when accumulating the product of weight and spike. This method controls the range of states of the neuron and only necessitates one-time action on the trained synapse. In our simulation of superconducting circuits, this algorithm has alleviated the problem of erroneous excitation while also significantly improving the accuracy of the model.

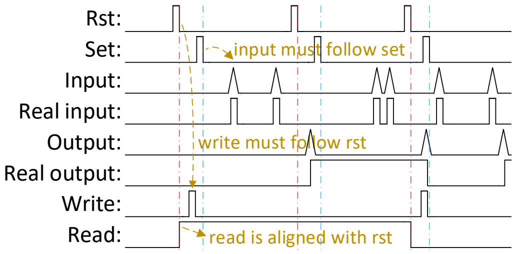


Figure 14: Example of asynchronous neuron timing.

5.2 Asynchronous Neuron Timing

As previously discussed, SUSHI comprises exclusively of asynchronous elements, such as the state controller, NPE, weight structure, and network structure. These elements do not have any additional clock lines, which means they cannot use the synchronous timing logic typically used in standard superconducting circuit designs to represent data. As a result, this presents a challenge to our timing design.

To overcome this challenge, we propose a pulse-based asynchronous neuron timing strategy for SUSHI. Our timing strategy evolves from the conventional pulse-level conversion used in superconducting circuits. Fig. 14 illustrates an example of pulse-level conversion, a short-time high-level signal is used to represent the input pulses to the superconducting circuit, e.g., "input" and the corresponding "real input". In this example, 6 pulses are input. When we sample the output pulses from the superconducting circuit, each output pulse will invert the sampled level, e.g., "output" and the corresponding "real output". In the figure, 3 pulses are sampled at the output channel, so the level at the real output channel is inverted by 3 times.

We then implement a heterogeneous timing approach for the IO and control pulses in SUSHI. Since processing of pulses by the NPEs is fully asynchronous, the input or output pulses of any timing can be correctly processed. Consequently, input pulse streams to the neural network can be arbitrarily fed without constraints, as shown in Fig. 14 under the "input" label.

For control pulses given to the SUSHI, we apply a simple asynchronous constraint to ensure correct operation, such as "rst", "set", "read", and "write" in Fig. 14. The following constraints are provided for these control channels: 1) the "write" pulse must follow the "rst" pulse as input; 2) the "input" pulse must follow the "set" pulse as input; and 3) the "read" pulse output is triggered by the "rst" pulse and aligned with it.

Our implementation of the asynchronous neuron timing has minimal impact on the asynchronous operation of SUSHI because control signals typically appear at the beginning or end of each batch of inputs. The inputs to the circuit primarily comprise input pulses that can be completely asynchronous without constraint, meaning the overall timing remains asynchronous.

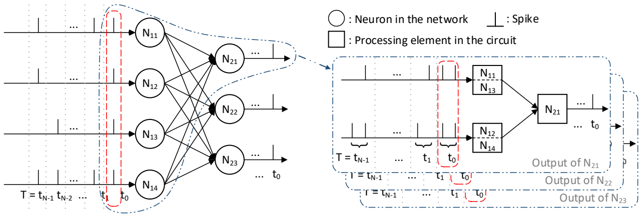
5.3 Bit-Slice SSNN Method

Our superconducting neuromorphic chip design has a small circuit resource overhead. However, applying large-scale on-chip neural network structures is still not feasible, mainly due to the current limitations of superconducting circuit integration. For instance, the Nb03 process can support up to 10^4 JJs in a $5mm \times 5mm$ chip [43]. The resource overhead of large-scale on-chip network structures still exceeds the integration limit, which presents a significant challenge to the practical application of neuromorphic processing on superconducting circuits.

To address the limitations of superconducting circuit integration, we propose the bit-slice SSNN method. Unlike conventional time-division multiplexing, the novel bit-slice processing method in superconducting circuits was first proposed by G. Tang [35]. The bit-slice method is based on the state-preserving capability of superconducting cells, which allows for the decomposition of an operation into batches without introducing additional storage units during processing. Typically, the bit-slice method is used for designing processing elements such as ALUs [36]. However, in this paper, we propose to apply the bit-slice method to the processing of SNNs for the first time.

Fig. 15 illustrates an example of our bit-slice SSNN method to superconducting neuromorphic circuits. The left side of the figure represents a neural network while the right side shows the on-chip structure. The bit-slice SSNN method is derived from the conventional superconducting bit-slice method, where the neurons in SNNs are treated as bits, sliced per layer to suit the on-chip width, and assigned different time periods. This approach solely focuses on regulating the channel and time of each synapse activity, so we incorporate it into the encoding phase. By leveraging the state-preserving capability of the superconducting cells, we eliminate the need for introducing additional control between the recoded slices. Once a time slice is completed, the output neuron produces the same output, and the subsequent time slice can be processed smoothly.

Our bit-slice SSNN method permits the design of superconducting neuromorphic chips of varying scales, compatible with different levels of superconducting circuit technology. This ensures that our SSNN can be utilized within the constraints of superconducting circuit integration, thereby enabling the design of fabricable neuromorphic chips with current technology.



(a) The network to be processed (b) On-chip Processing

Figure 15: Example of bit-slice SSNN method.

6 EXPERIMENTS AND RESULTS

Our evaluation consists of three phases. Firstly, we compare the output of our microarchitecture-level design with the SNN software to confirm that the behavior of SUSHI is accurate. In the second phase, we validate SUSHI by comparing the sampling output of a fabricated chip with the simulation results. Finally, we evaluate the performance and power efficiency of SUSHI, and compare it with state-of-the-art asynchronous neuromorphic chips.

The basic SSNN architecture used in our experiments is a fully-connected spiking neural network consisting of INPUT28*28-Flatten-FC800-IF-FC10-IF. We employ the IF neuron model with a threshold voltage of 1.0, and a simulation period of 5 time steps. The input data is generated using the Poisson encoder and encoded according to the constraints in Table 1. We use adam [18] as the optimizer, with a learning rate of 1e-3. For our experiments, we utilize the MNIST dataset for handwritten digit recognition [20], as well as the more complex Fashion MNIST dataset for clothing image recognition [41]. These datasets comprise 60,000 examples and a test set of 10,000 examples. Our SNN design and experiments are based on SpikingJelly [10], which is an open-source deep learning framework for SNN, widely renowned for its implementation of deep SNN [11].

Our RSFQ-based superconducting chip design is based on the standard cell library of SIMIT-Nb03 [12], which has succeeded in fabricating real chips many times. Since the current superconducting fabrication technique is more stable for chips with low JJs density, we only place the necessary number of NPEs without weight structure to meet evaluation demands during the fabrication process, which is 2. We utilize liquid helium to create a low-temperature experimental environment of 4.2K (Fig. 18). We use the mesh network in our design, as detailed in section 4.2.2. For designs requiring larger scales, our simulation environment for the design is Synopsys VCS [32] and Verdi [33] tool flow. We execute the complete SNN on the chip using our bit-slice SSNN method.

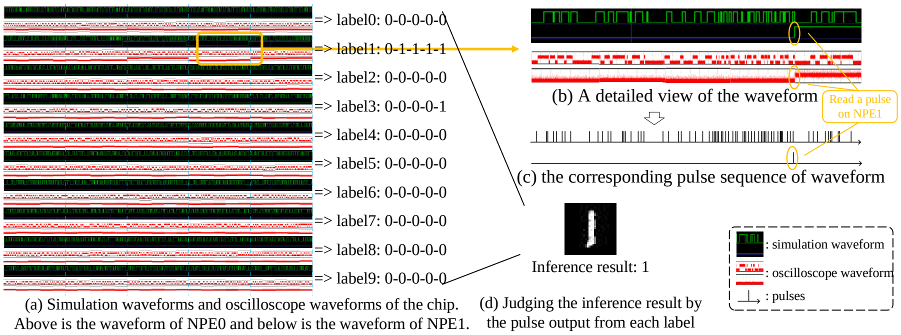
6.1 Simulation and Verification

We first validate our SUSHI design. Table 3 presents the discrepancy in the inference results between SpikingJelly and SUSHI for MNIST and Fashion-MNIST datasets. Both methods utilize the same network structure and neurons. However, we carry out optimization to meet the width and memory limitations of SUSHI. This optimization entails the elimination of potential residuals in the neuron model and the 1-bit quantization (SpikingJelly uses floating point) for the simulation. As a result of these optimizations, some disparities exist between the chip and the SpikingJelly results.

Table 3 provides accuracy and consistency metrics obtained using both methods. These metrics show the number of correct labels hit by each method, as well as the number of hits of the same label (which may not be correct) by both methods. The experimental results demonstrate that there is little differentiation between the inference results obtained by SUSHI and SpikingJelly. Despite a 1.82% and 11.29% difference in the inference results of MNIST and Fashion-MNIST datasets, respectively, the accuracy differed by only 0.81% and 2.67%. Thus, our optimization of the inference process does not impact the neuromorphic processing, and our SUSHI design can provide reliable inference results.

6.2 Fabrication and Measurement

Fig. 17 illustrates the microstructure of our chip, while Fig. 16 demonstrates our validation of a SUSHI chip and the overall workflow for obtaining a neural network inference result from the chip. The verification of the chip is performed by sampling its output



(a) Simulation waveforms and oscilloscope waveforms of the chip. Above is the waveform of NPE0 and below is the waveform of NPE1.

(b) A detailed view of the waveform

(c) the corresponding pulse sequence of waveform

(d) Judging the inference result by the pulse output from each label

Figure 16: Comparison of simulation waveforms and oscilloscope waveforms of SUSHI and the overall workflow for obtaining inference results from the chip.

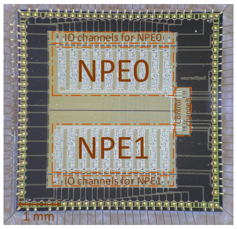


Figure 17: Microphotograph of SUSHI

Figure 18: Experimental environment of 4.2K

Dataset	MNIST		Fashion-MNIST	
Platform	SpikingJelly	SUSHI	SpikingJelly	SUSHI
Accuracy	98.65%	97.84%	88.90%	86.23%
Consistency	100.0%	98.18%	100.0%	88.71%

Table 3: Differences in SNN inference results between SpikingJelly and SUSHI.

Platform	TrueNorth	Tianjic	SUSHI
Model	SNN	Hybrid	SSNN
Memory	SRAM	SRAM	-
Technology	CMOS, 28 nm	CMOS, 28 nm	RSFQ, 2 μm
Clock (MHz)	Async	300	Async
Area (mm^2)	430	14.44	103.75
Power (mW)	63-300	950	41.87
GSOPS	58	-	1,355
GSOPS/W	400	649	32,366

Table 4: Comparison of SUSHI with the state-of-the-art neuromorphic chips.

waveform, which should be consistent with the simulation waveform. Initially, we evaluate the functionality of the NPE implemented on the chip, such as the flip, fire, and reset mechanisms mentioned in Section 4.1, to ensure that the design is accurate and operational.

Subsequently, we confirm the correctness of the chip using the exact same model as Table 3. Fig. 16(a) shows a comparison between the simulation waveforms and the oscilloscope waveform of the chip output. Fig. 16(b) shows a detailed view of the waveform. Finally, we validate the chip output based on our asynchronous neuron timing, as described in Section 5.2. Fig. 16(c) exhibits the output sequence of the chip that corresponds to the waveform in Fig. 16(b). A thorough inspection of Fig 16 indicates that the output of the chip is consistent with the results of our simulation of the superconducting neuromorphic chip design.

We execute the complete input sequence and obtained the output inference sequence, as shown in Fig. 16(a-d). To the best of our knowledge, this is the first instance that a complete SNN has been executed on a superconducting neuromorphic chip yielding accurate inference results. Previously published research has only run incomplete SNNs or resorted to simulation outcomes.

6.3 Performance Evaluation

In this section, we evaluate the performance of our design. Unlike general-purpose computing that are optimized for high-precision operations (e.g., measured by FLOPS), neuromorphic computing focuses on fundamental operations to perform neuromorphic processing, i.e., synaptic operations. Synaptic operations per second (SOPS) is frequently utilized to benchmark the neuromorphic chips. It is calculated by $avg.firing.rate \times avg.active.synapses$, without taking into consideration the details required to implement each synaptic operation [7]. SOPS indicates the average magnitude of the pulses passing through connected synapses.

SOPS does not consider the implementation details of synaptic operations, so it enables a fair comparison between different neuromorphic chips. We use the state-of-the-art neuromorphic chip designs, Truenorth [22] and Tianjic [26], for comparison. We conduct our performance evaluation by employing the same network structure used during verification, along with high spike rates. Our peak performance is obtained using a design comprising 99,982 JJs and an area of 103.75mm^2 (32 NPEs in Fig. 13).

Table 4 provides an evaluation and comparison of the results of our design. During evaluation, the peak performance of SUSHI is 1,355 GSOPS, which is 23 times greater than that of TrueNorth. Additionally, the maximum power efficiency of our design reaches 32,366 GSOPS/W, which is 81 and 50 times higher than TrueNorth and Tianjic, respectively. The remarkable performance and power efficiency of SUSHI are primarily attributes to the ultra-low flip latency and ultra-low-power characteristics of the superconducting single-flux-quantum cells. On the other side, although it is difficult to directly compare with synchronous designs, we evaluate SUSHI's performance in terms of frames per second (FPS). In the case of MNIST and Fashion-MNIST, SUSHI achieves up to 2.61×10^5 FPS.

A. Performance

Fig. 19 illustrates the performance of the design of SUSHI with an increase in the number of NPEs. As the figure shows, performance of SUSHI can be increased by augmenting the number of NPEs and the size of the NPE network, reaching a maximum of 1,355 GSOPS, which is 23 times higher than TrueNorth's peak performance. It is important to note that SOPS measures the average magnitude of pulses within synapses, and thus it is related to the size of the network. E.g., a 4 × 4 network with 8 neurons has 16 synapses, so its scale is 4× than a 2 × 2 network (4 synapses).

However, this performance enhancement is limited by wiring overhead in superconducting circuits and may not maintain its growth trend. We found that transmission line delay, which cannot be ignored in large-scale superconducting circuit designs, has a substantial impact on the performance of the chip. Through analysis, we found that when processing a single pulse, the transmission delay accounts for about 53% of the total in the 16 × 16 design, while only about 6% in the 1 × 1 design. This is one of the major challenges facing current superconducting chip design.

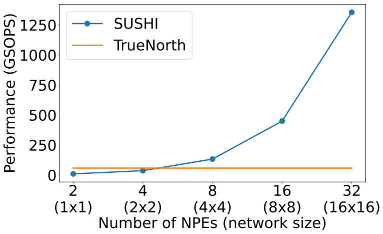


Figure 19: Performance of SUSHI with different number (network size) of NPEs.

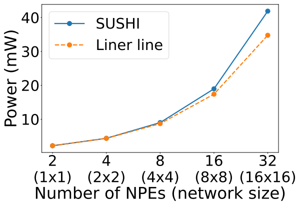


Figure 20: Power of SUSHI with different number (network size) of NPEs.

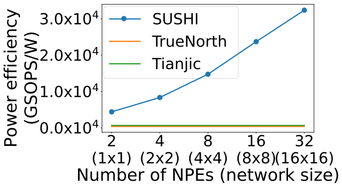


Figure 21: Power efficiency of SUSHI with different number (network size) of NPEs.

B. Power efficiency

Fig. 20 presents the power consumption of SUSHI as the NPE number increases, and Fig. 21 shows its power efficiency. We evaluate the power of SUSHI without considering the cooling costs. Additionally, for comparison, the power efficiency of TrueNorth and Tianjic are provided in the figure. As shown, power efficiency of SUSHI notably surpasses that of the state-of-the-art design. This is also primarily attributed to ultra-low flipping latency and energy consumption of the superconducting circuit cells. Similar to the performance result, the power efficiency gains that attained by enlarging the SUSHI scale will be slightly impacted due to the rapid growth of the accumulated energy consumption of transmission lines in larger designs. Our evaluation results demonstrate the strong potential of SUSHI in neuromorphic processing.

7 RELATED WORK

A. Neuromorphic processing

Neuromorphic processing is a technique that customizes SNN algorithms and hardware based on artificial intelligence algorithms and human brain processes. In recent years, the spatio-temporal backpropagation (STBP) learning rule [40] and customized threshold-based regularization methods [45] have enabled SNNs to achieve comparable performance to that of ANNs in classical deep structures like VGG and ResNet. Neuromorphic chips use the "neuron core" as basic elements and replace the unified external memory with the local memory of neuron cores, receiving widespread interest. When neuromorphic chips are applied to SNN processing, the event-driven nature of SNNs ensures that the neuron cores perform computations only when spike events arrive, leading to lower dynamic power consumption compared to traditional hardware. TrueNorth [1] employs this neuromorphic architecture, containing 4096 neuron cores on a single chip, routed and connected through a two-dimensional grid, and supporting multi-chip expansion. Tianjic [27] accommodates the computational modes of both SNN and ANN on this architecture, while Loihi [8] has achieved online learning and supports various SNN learning algorithms.

B. Superconducting neuromorphic units

As the implementation of Moore's Law approaches its physical limits, superconducting devices are becoming increasingly crucial in emerging neuromorphic processing. Some neural network accelerators implemented with superconducting circuits, such as SuperNPU [16] and SMART [46], DNN accelerators implemented with SFQ circuits, and a DNN accelerator based on stochastic computing implemented with superconducting AQFP circuits [6], are currently available. In the field of superconducting neuromorphic processing, most works utilize JJ to implement neurons. A neuron based on the IFN model was proposed, but only at the simulation level [13]. Researchers have verified and demonstrated the relationship between the frequency of the output pulse of the proposed JJ neuron and the phase of the input pulse [30]. A superconductive pseudo sigmoid function generator based on SFQ technology has also been implemented and tested [42], while an SNN architecture was proposed [3], but only the soma part of the neuron was fabricated and demonstrated. In this paper, for the first time, a complete SNN is applied to a real superconducting chip.

8 CONCLUSION

Rapid Single-Flux-Quantum superconducting technology is a highly promising solution for neuromorphic processing, yet no existing work has managed to fabricate and validate a chip capable of running a complete neural network. In response to this challenge, we present SUSHI, a superconducting neuromorphic processing chip that fully leverages the potential of superconducting neuromorphic processing. We design the architecture of SUSHI and propose methodologies that enable neuromorphic processing of SUSHI within constraints of superconducting circuits, including SSNN, asynchronous neuron timing, and a bit-slice SSNN method. We successfully fabricate and verify SUSHI, representing the first time that neural networks have been completely executed on a superconducting chip. Our evaluation results demonstrate that SUSHI achieves a peak performance of 1,355 GSOPS, which is 23 times greater than TrueNorth, and a power efficiency of 32,366 GSOPS/W, outperforming the state-of-the-art TrueNorth and Tianjic by 81 and 50 times, respectively. We believe our work will be informative for the field of superconducting neuromorphic processing. Our future research will focus on developing more functional superconducting neuromorphic processing units.

ACKNOWLEDGMENTS

This work was supported in part by the Strategic Priority Research Program of Chinese Academy of Sciences, under Grant XDA18000000.

REFERENCES

[1] Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, et al. 2015. 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* 34, 10 (2015), 1537–1557.

[2] Yuki Ando, Ryo Sato, Masamitsu Tanaka, Kazuyoshi Takagi, and Naofumi Takagi. 2015. 80-GHz Operation of an 8-Bit RSFQ Arithmetic Logic Unit. In *2015 15th International Superconductive Electronics Conference (ISEC)*. 1–3. https://doi.org/10.1109/ISEC.2015.7383427

[3] Ali Bozbey, Mustafa Altay Karamuftuoglu, Sasan Razmkhah, and Murat Ozbayoglu. 2020. Single Flux Quantum Based Ultrahigh Speed Spiking Neuromorphic Processor Architecture. arXiv:1812.10354 [cs.ET]

[4] Darren K Brock. 2001. RSFQ technology: Circuits and systems. *International journal of high speed electronics and systems* 11, 01 (2001), 307–362.

[5] Brenden A Butters, Reza Baghdadi, Murat Onen, Emily A Toomey, Owen Medeiros, and Karl K Berggren. 2021. A scalable superconducting nanowire memory cell and preliminary array test. *Superconductor Science and Technology* 34, 3 (2021), 035003.

[6] Ruizhe Cai, Ao Ren, Olivia Chen, Ning Liu, Caiwen Ding, Xuehai Qian, Jie Han, Wenhuo Luo, Nobuyuki Yoshikawa, and Yanzhi Wang. 2019. A Stochastic-Computing based Deep Learning Framework using Adiabatic Quantum-Flux-Parametron Superconducting Technology. In *2019 ACM/IEEE 46th Annual International Symposium on Computer Architecture (ISCA)*. 567–578.

[7] Andrew S. Cassidy, Rodrigo Alvarez-Icaza, Filipp Akopyan, Jun Sawada, John V. Arthur, Paul A. Merolla, Pallab Datta, Marc Gonzalez Tallada, Brian Taba, Alexander Andreopoulos, Arnon Amir, Steven K. Esser, Jeff Kusnitz, Rathinakumar Appuswamy, Chuck Haymes, Bernard Brezzo, Roger Moussalli, Ralph Bellofatto, Christian Baks, Michael Mastro, Kai Schleupen, Charles E. Cox, Ken Inoue, Steve Millman, Nabil Imam, Emmett Mcquinn, Yutaka Y. Nakamura, Ivan Vo, Chen Guok, Don Nguyen, Scott Lekuch, Sameh Asaad, Daniel Friedman, Bryan L. Jackson, Myron D. Flickner, William P. Risk, Rajit Manohar, and Dharmendra S. Modha. 2014. Real-Time Scalable Cortical Computing at 46 Giga-Synaptic OPS/Watt with 100× Speedup in Time-to-Solution and 100,000× Reduction in Energy-to-Solution. In *SC '14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 27–38. https://doi.org/10.1109/SC.2014.8

[8] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. 2018. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro* 38, 1 (2018), 82–99.

[9] Ted Van Duzer and Charles W. Turner. 1998. *Principles of Superconductive Devices and Circuits, (Second Ed.)*. Prentice Hall PTR, USA.

[10] Wei Fang, Yanqi Chen, Jianhao Ding, Ding Chen, Zhaofei Yu, Huihui Zhou, Timothée Masquelier, Yonghong Tian, and other contributors. 2020. SpikingJelly. https://github.com/fangwei123456/spikingjelly. Accessed: 2022-12-20.

[11] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. 2021. Deep Residual Learning in Spiking Neural Networks. In *Advances in Neural Information Processing Systems*, Vol. 34. 21056–21069. https://proceedings.neurips.cc/paper/2021/file/afe434653a898da20044041262b3ac74-Paper.pdf

[12] Xiaoping Gao, Qi Qiao, Mingliang Wang, Minghui Niu, Huanli Liu, Masaaki Maezawa, Jie Ren, and Zhen Wang. 2021. Design and verification of SFQ cell library for superconducting LSI digital circuits. *IEEE Transactions on Applied Superconductivity* 31, 5 (2021), 1–5.

[13] Tetsuya Hirose, Ken Ueno, Tetsuya Asai, and Yoshihito Amemiya. 2006. Single-flux-quantum circuits for spiking neuron devices. *International Congress Series* 1291 (2006), 221–224. https://doi.org/10.1016/j.ics.2006.02.005 Brain-Inspired IT II: Decision and Behavioral Choice Organized by Natural and Artificial Brains. Invited and selected papers of the 2nd International Conference on Brain-inspired Information Technology held in Hibikino, Kitakyushu, Japan between 7 and 9 October 2005.

[14] Alan L Hodgkin and Andrew F Huxley. 1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology* 117, 4 (1952), 500.

[15] D. Scott Holmes, Andrew L. Ripple, and Marc A. Manheimer. 2013. Energy-Efficient Superconducting Computing—Power Budgets and Requirements. *IEEE Transactions on Applied Superconductivity* 23, 3 (2013), 1701610–1701610. https://doi.org/10.1109/TASC.2013.2244634

[16] Koki Ishida, Ilkwon Byun, Ikki Nagaoka, Kosuke Fukumitsu, Masamitsu Tanaka, Satoshi Kawakami, Teruo Tanimoto, Takatsugu Ono, Jangwoo Kim, and Koji Inoue. 2020. SuperNPU: An extremely fast neural processing unit using superconducting logic devices. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 58–72.

[17] Tahereh Jabbari, Gleb Krylov, Stephen Whiteley, Eric Mlinar, Jamil Kawa, and Eby G Friedman. 2019. Interconnect routing for large-scale RSFQ circuits. *IEEE Transactions on Applied Superconductivity* 29, 5 (2019), 1–5.

[18] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG]

[19] Nobutaka Kito and Kazuyoshi Takagi. 2021. An RSFQ flexible-precision multiplier utilizing bit-level processing. *Journal of Physics: Conference Series* 1975, 1 (jul 2021), 012025. https://doi.org/10.1088/1742-6596/1975/1/012025

[20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[21] K.K. Likharev and V.K. Semenov. 1991. RSFQ logic/memory family: a new Josephson-junction technology for sub-terahertz-clock-frequency digital systems. *IEEE Transactions on Applied Superconductivity* 1, 1 (1991), 3–28. https://doi.org/10.1109/77.80745

[22] Paul A. Merolla, John V. Arthur, Rodrigo Alvarez-Icaza, Andrew S. Cassidy, Jun Sawada, Filipp Akopyan, Bryan L. Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, Bernard Brezzo, Ivan Vo, Steven K. Esser, Rathinakumar Appuswamy, Brian Taba, Arnon Amir, Myron D. Flickner, William P. Risk, Rajit Manohar, and Dharmendra S. Modha. 2014. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 6197 (2014), 668–673. https://doi.org/10.1126/science.1254642 arXiv:https://www.science.org/doi/pdf/10.1126/science.1254642

[23] Ammar Mohemmed, Stefan Schliebs, Satoshi Matsuda, and Nikola Kasabov. 2012. Span: Spike pattern association neuron for learning spatio-temporal spike patterns. *International journal of neural systems* 22, 04 (2012), 1250012.

[24] Oleg A Mukhanov, Paul D Bradley, Steven B Kaplan, Sergey V Rylov, and AF Kirichenko. 1995. Design and operation of RSFQ circuits for digital signal processing. In *Proc. 5th Int. Supercond. Electron. Conf.* 27–30.

[25] Minh-Hai Nguyen, Guilhem J Ribeill, Martin V Gustafsson, Shengjie Shi, Sriharsha V Aradhya, Andrew P Wagner, Leonardo M Ranzani, Lijun Zhu, Reza Baghdadi, Brenden Butters, et al. 2020. Cryogenic memory architecture integrating spin Hall effect based magnetic memory and superconductive cryotron devices. *Scientific reports* 10, 1 (2020), 248.

[26] J. Pei, L. Deng, S. Song, M. Zhao, and L. Shi. 2019. Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature* 572, 7767 (2019), 106.

[27] Jing Pei, Lei Deng, Sen Song, Mingguo Zhao, Youhui Zhang, Shuang Wu, Guanrui Wang, Zhe Zou, Zhenzhi Wu, Wei He, et al. 2019. Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature* 572, 7767 (2019), 106–111.

[28] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. 2016. XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. In *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Springer International Publishing, Cham, 525–542.

[29] Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. 2019. Towards spike-based machine intelligence with neuromorphic computing. *Nature* 575, 7784 (2019), 607–617.

[30] Ken Segall, Matthew LeGro, Steven Kaplan, Oleksiy Svitelskiy, Shreeya Khadka, Patrick Crotty, and Daniel Schult. 2017. Synchronization dynamics on the picosecond time scale in coupled Josephson junction neurons. *Physical Review E* 95, 3 (2017), 032220.

[31] Vasili K Semenov, Yuri A Polyakov, and Sergey K Tolpygo. 2019. Very large scale integration of Josephson-junction-based superconductor random access memories. *IEEE Transactions on Applied Superconductivity* 29, 5 (2019), 1–9.

[32] Synopsys. 2023. *VCS*. https://www.synopsys.com/verification/simulation/vcs.html

[33] Synopsys. 2023. *Verdi*. https://www.synopsys.com/verification/debug/verdi.html

[34] Masamitsu Tanaka, Masato Suzuki, Gen Konno, Yuki Ito, Akira Fujimaki, and Nobuyuki Yoshikawa. 2016. Josephson-CMOS hybrid memory with nanocryotrons. *IEEE Transactions on Applied Superconductivity* 27, 4 (2016), 1–4.

[35] Guang-Ming Tang. 2016. Studies on Datapath Circuits for Superconductor Bit-Slice Microprocessors.

[36] Guang-Ming Tang, Pei-Yao Qu, Xiao-Chun Ye, and Dong-Rui Fan. 2018. Logic design of a 16-bit bit-slice arithmetic logic unit for 32-/64-bit RSFQ microprocessors.

IEEE Transactions on Applied Superconductivity 28, 4 (2018), 1–5.

[37] Guang-Ming Tang, Pei-Yao Qu, Xiao-Chun Ye, Dong-Rui Fan, and Ning-Hui Sun. 2018. 32-bit 4× 4 bit-slice RSFQ matrix multiplier. *IEEE Transactions on Applied Superconductivity* 28, 7 (2018), 1–5.

[38] Sergey K Tolpygo, Vladimir Bolkhovsky, Terence J Weir, Alex Wynn, Daniel E Oates, Leonard M Johnson, and Mark A Gouker. 2016. Advanced fabrication processes for superconducting very large-scale integrated circuits. *IEEE Transactions on Applied Superconductivity* 26, 3 (2016), 1–10.

[39] T. Van Duzer, C. W. Turner, D. G. McDonald, and Alan F. Clark. 1982. Principles of Superconductive Devices and Circuits. *Physics Today* 35, 2 (02 1982), 80–81. https://doi.org/10.1063/1.2914944 arXiv:https://pubs.aip.org/physicstoday/article-pdf/35/2/80/8290405/80_2_online.pdf

[40] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. 2018. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience* 12 (2018), 331.

[41] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv:1708.07747 [cs.LG]

[42] Yuki Yamanashi, Kazumasa Umeda, and Nobuyuki Yoshikawa. 2012. Pseudo sigmoid function generator for a superconductive neural network. *IEEE transactions on applied superconductivity* 23, 3 (2012), 1701004–1701004.

[43] Liliang Ying, Xue Zhang, Minghui Niu, Jie Ren, Wei Peng, Masaaki Maezawa, and Zhen Wang. 2021. Development of multi-layer fabrication process for SFQ large scale integrated digital circuits. *IEEE Transactions on Applied Superconductivity* 31, 5 (2021), 1–4.

[44] N Yoshikawa, J Koshiyama, K Motoori, F Matsuzaki, and K Yoda. 2001. Cell-based top-down design methodology for RSFQ digital circuits. *Physica C: Superconductivity* 357 (2001), 1529–1539.

[45] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. 2021. Going Deeper With Directly-Trained Larger Spiking Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 12 (May 2021), 11062–11070. https://doi.org/10.1609/aaai.v35i12.17320

[46] Farzaneh Zokaee and Lei Jiang. 2021. SMART: A Heterogeneous Scratchpad Memory Architecture for Superconductor SFQ-Based Systolic CNN Accelerators. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture* (Virtual Event, Greece) *(MICRO '21)*. Association for Computing Machinery, New York, NY, USA, 912–924. https://doi.org/10.1145/3466752.3480041