Energy-Efficient Single-Flux-QuantumBased Neuromorphic Computing

Michael L. Schneider, Christine, A. Donnelly, Stephen E. Russek, Burm Baek, Matthew R. Pufall, Peter F. Hopkins, William H. Rippard
National Institute of Standards and Technology, Boulder, CO 80305, USA

***Abstract*—Recent experimental work has demonstrated nano-textured magnetic Josephson junctions (MJJs) that exhibit tunable spiking behavior with ultra-low training energies in the attojoule range. MJJ devices integrated with standard single-flux-quantum neural systems form a new class of neuromorphic technologies that have spiking energies between 10^{-18} J and 10^{-21} J, operation frequencies up to 100 GHz, and nanoscale plasticity. Here, we present the design of neural cells utilizing MJJs that form the basic elements in multilayer perception and convolutional networks. We present SPICE models, using experimentally derived Verilog A models for MJJs, to assess the performance of these cells in simple neural network structures. Modeling results indicate that the tunable Josephson critical current I_C can function as a weight in a neural network. Using SPICE we model a fully connected two layer network with 9 inputs and 3 outputs.**

Index Terms—Neuromorphic computing, single flux quantum, magnetic Josephson junctions.

I. Introduction

Many neuromorphic hardware technologies are being explored for their potential to increase the efficiency of computing certain problems and thus facilitate machine learning with greater energy efficiency or with more complexity. Among the technologies being developed, single-flux-quantum-based Josephson junctions are a promising choice for their extremely low energy consumption and intrinsic spiking behavior.

The basic element in single-flux-quantum (SFQ) circuits is the Josephson junction (JJ)[1], which emits a voltage spike with an integrated amplitude equal to the flux quantum ($\varphi_0 = 2.07 \times 10^{-15}\,\mathrm{V \cdot s}$) when the superconducting order parameter undergoes a 2π phase slip[2]. These very low energy pulses have been demonstrated in circuits at frequencies well above 100 GHz[3]. Recently, it has been demonstrated that incorporating a nano-textured magnetic barrier into a Josephson junction results in a JJ whose properties can be dynamically changed via currents and magnetic fields[4]. Here we model neuromorphic circuits based on a combination of traditional SFQ JJs and tunable MJJs using physically realistic parameters.

SFQ circuits are being developed on a large scale for digital supercomputing because of their high speed and extremely low operating energies[5]. This effort can be leveraged in neuromorphic SFQ circuits[6-9]. In this case, we model the neuromorphic circuits with WRSPICE[10].* a version of SPICE that has integrated support for the physics of SFQ circuits. In addition, we used Verilog A to define our own physical model of the nano-textured magnetic JJ. This compact model was compiled and integrated with WRSPICE.

II. Dynamically Tunable Josephson Junction Model

The resistively and capacitively shunted junction (RCSJ) model provides a dynamic equation for the phase evolution of a current-biased Josephson junction[11]. A maximum current of I_C may flow as a supercurrent without any voltage developing across the junction. The relationship between superconducting phase and current is given by the Josephson relation: $I = I_C \sin(\varphi)$, where φ is the phase difference across the junction. In the voltage state, a quasiparticle current contributes to the total current according to $I = G_N(V) \times V$, where G_N is the junction conductivity and V is the junction voltage. The junction voltage V can be related to the phase by the Josephson voltage-phase relationship $d\varphi/dt = 2eV/\hbar$, where e is the electron charge and $\hbar$ is the reduced Planck's constant. . $G_N(V)$ can be strongly voltage-dependent for tunnel junctions or voltage independent for a superconducting-normal-superconducting junction. Here, $G_N(V)$ is assumed to remain constant with voltage for the low resistance junctions in this model. Finally, the displacement current is given by $I = C\,dV/dt$, where C is the junction capacitance, which can again be related to junction phase by the voltage-phase relationship.

The dynamics in the voltage state depend on the Stewart-McCumber parameter[2],

(1) $\beta_C = \frac{2e}{\hbar} I_C R_N^2 C$,

where R_N is the normal state resistance. This parameter is equal to the squared quality factor, Q^2, of a parallel LRC circuit (here, "L" is the small-signal inductance found by linearizing the supercurrent phase-current relationship about the zero bias phase difference across the junction). The model described in this work uses junctions that are close to critical damping $\beta_C \approx 1$ or overdamped $\beta_C < 1$. For these junctions, a phase evolution in the voltage state followed by a return to the zero-bias phase will cause the junction phase to evolve through one or multiple quantized 2π phase slips. Each of these 2π phase slips can be shown by the Josephson phase-voltage relationship to correspond to a voltage pulse of integrated area $\int V(t)dt = h/2e = \Phi_0$. These phase slips define "single flux quantum pulses"[11].

We modified the RCSJ model that was outlined above to include a dependence of the superconducting critical current on the magnetic order parameter for our MJJ devices[12]. Physically, in the case of the nano-textured magnetic JJs, the

* *This does not imply endorsement by NIST or that the product is the best available for the purpose.*

978-1-5386-1553-9/17/$31.00 ©2017 IEEE

critical current of the junction can be tuned by changing the order of the magnetic clusters[4]. We model this by defining the magnetic order parameter as

$$(2)\quad m = \frac{1}{M}\sum_i \vec{m}_i$$

where $\vec{m_i}$ are the moments of each of the magnetic clusters, and M is the total magnetic moment of all clusters in the junction. We define the effect of magnetic order on the junction critical current as:

$$(3)\quad I_C(m,T) = \left((1-m)I_{CV} + I_{CM}\right)\left(1-\left(\frac{T}{T_c}\right)^2\right),$$

where I_{CV} is the variable portion of the critical current, which is influenced by the order of the magnetic particles, I_{CM} is the minimum critical current defined when the barrier has a maximum magnetic order, and T_c is the superconducting transition temperature. The magnetic order parameter varies with the integrated junction voltage according to $dm \propto V(t)dt \mid E_{pulse} > E_m$ between $m = 0$ and $m = 1$, where E_{pulse} is the pulse energy and E_m is the minimum energy required to change the magnetic order. In other words, a positive voltage pulse across the junction, with enough energy, causes the order parameter to increase and the critical current to decrease up to a saturation point. In the circuits presented here, we set the minimum pulse energy equal to 3 aJ, so that once the weights are set, they are not affected by the operational SFQ pulses.

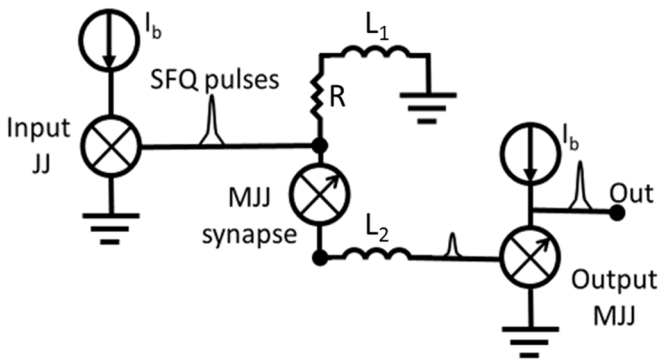

Fig. 1. Circuit diagram of the basic elements connected as 1 input through 1 weight to 1 output. $L_1 = L_2 = 10$ picohenry and $R = 10$ mΩ. The voltage pulse schematics indicate the measured nodes for the input, middle and output neurons.

III. Neuromorphic Unit Cell

Figure 1 shows the schematic of the basic neuromorphic unit cell. The JJ on the left side is performing the function of an input "neuron." If the current bias is on, then the junction will spike. The spiking rate of this junction has been tuned to be roughly 0.35 GHz. The JJ in the center is a "synaptic" MJJ which will weight the transmitted pulse from the input "neuron." We adjust the critical current of these junctions between 1 μA and 100 μA to achieve the desired "synaptic" weighting. This adjustment would be accomplished physically by adjusting the magnetic order in the MJJ, something that can be done dynamically in the system. The JJ on the right side of the circuit acts as the output threshold "neuron". This JJ is also an MJJ but with a higher range of critical currents that we can adjust from 100 μA to 500 μA and is thus analogous to the bias of a neural network. Physically, one can adjust the range of the critical currents that the MJJ has by changing the size of the junction. For the case of 1 – 100 μA currents, devices would be approximately 1 μm in diameter, while MJJs with a critical current range of 5-500 μm can be achieved in MJJs with a diameter of a 3 μm [4].

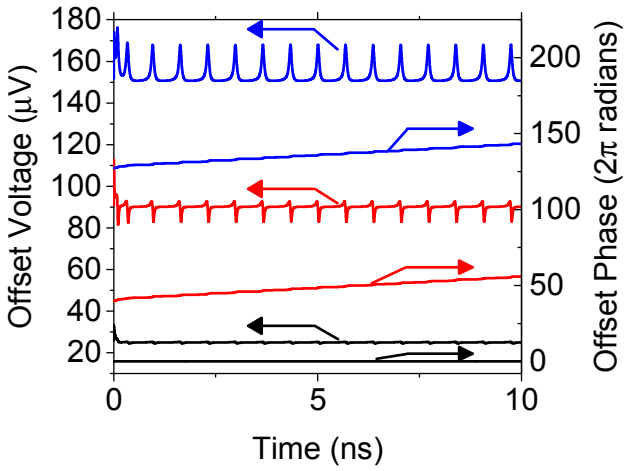

Fig. 2. Behavior of the basic MJJ neural network. The input neuron is on top in blue, the middle MJJ is in red and the output MJJ is in black. All voltages (left axis) are displayed on top of their respective phase evolution (right axis). In this example, the middle MJJ has a critical current of 10 μA corresponding to a low weight, and leading to no spiking on the output.

The circuit operates when the input DC bias is turned on. At this point the JJ on the left will start firing with a frequency of roughly 0.35 GHz. These pulses are then seen as current pulses by the middle synaptic MJJ. This MJJ will fire with a pulse energy ($I_C\varphi_0/2\pi$) that depends on the critical current of this MJJ. The resistor and inductor above the middle MJJ act to stabilize the circuit by introducing a recovery time between spikes, this acts to desensitize the overall circuit from small reflected pulses. The output JJ has a small constant DC bias that is below the spiking threshold. This JJ will undergo a 2π phase slip if the pulse transmitted through the middle JJ is large enough. It should be noted that these bias lines can be shared within a layer as the only energy that should be dissipated is the SFQ energy when the input or output spikes. This has previously been shown to allow for extremely low energy operation in energy efficient rapid single flux quantum circuits[13].

Figures 2 and 3 show the spiking output and phase evolution of the three JJs in the circuit. In Fig. 2 the critical current of the middle MJJ is 10 μA and the output MJJ has an $I_C = 300$ μA. As can be seen in the data, the input and middle JJs are spiking, but the output JJ in this configuration is not spiking. In this example, as the middle junction enters the voltage state, it becomes resistive and more of the input current spike is shunted to the ground above the synaptic JJ. In Fig. 3, the middle MJJ has an $I_C = 80$ μA and the output MJJ has an $I_C = 300$ μA. As can be seen in both the voltage and phase traces, the input and output JJs are firing in this configuration. The weighting JJ in this case

is not undergoing 2π phase slips, but rather is remaining in the superconducting state, and thus passing the initial input pulse through to the output JJ.

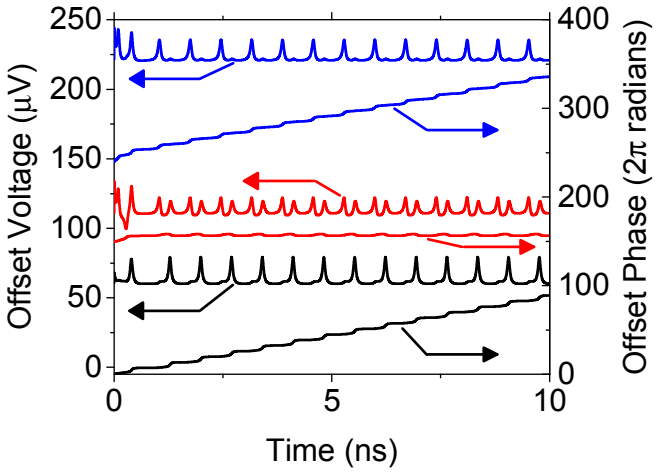

Fig. 3. Behavior of the basic MJJ neural network. The input neuron is on top in blue, the middle MJJ is in red and the output MJJ is in black. All voltages (left axis) are displayed on top of their respective phase evolution (right axis). In this example, the middle MJJ has a critical current of 80 μA corresponding to a high weight, and leading to spiking on the output.

IV. Nine Input Pixel Example

These elements (input JJ, synaptic MJJ, output threshold MJJ) can be used to implement a neural network in hardware. We use an example that was developed as a test for physical memristor circuits[14]. With a 3×3 input pixel array one can define 3 unambiguous letters (z, v, n) and train the neural network to identify these letters and report the answer on one of three outputs. Further, one can add 1 pixel of noise to each of the letters without duplicating patterns that conflate any of the letters. In our example, we use a standard neural network script written in Python to find the weights of a fully connected 9×3 neural network that will be used to solve this example problem[15]. Once the weights are found with the script, we use linear scaling to map these weights to critical current values in the middle layer of the network. In this example, there are only 30 unique input sequences and therefore we use all of the available data for training. As one might expect, we are able to train a standard neural network to reach 100 % identification for this example. In addition, by mapping the weights to critical current values, we are also able to get 100 % correct identification in the SPICE simulation of our MJJ network.

Figure 4 shows the circuit diagram for the JJ neural network. There are 9 input JJs each of which has a DC input bias that is either off (0 μA) or on (500 μA). These inputs are connected to 27 synaptic MJJs which have their critical currents adjusted to form the weights of the middle layer. These are then connected to 3 threshold MJJs forming the output layer.

A SPICE simulation of this circuit was run for 89 ns. In this simulation, the input biases were changed every 2.83 ns to a new test case. After 89 ns, the circuit has gone through all 30 test cases. The circuit correctly identifies the input 100 % of the time. These results are a promising indication that the circuit could be run in real time for image recognition with input images changing as quickly as every 3 ns in this example.

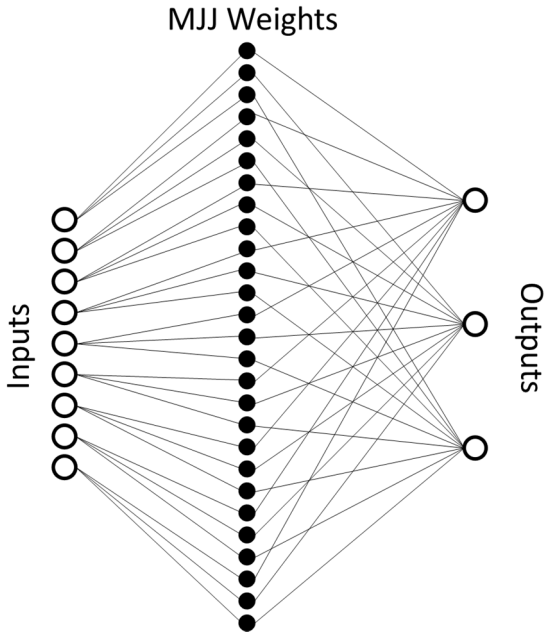

Fig. 4. Schematic of the 9 x 3 MJJ neural network connections.

Figure 5a shows the results of the 89 ns full input image test. The output pulse state for each of the 3 output MJJ neurons is plotted as a function of time. The pattern input is changing every 2.83 ns to the next image input. The output is shown on the graph. The blue output shows spiking on the "z" output neuron, which correctly spikes with the first 10 input cases of "z" or 1 pixel of noise away from "z". The red output shows the spiking on the "v" output neuron, which correctly spikes with the middle 10 input cases of "v" or one pixel away from "v". The black output shows the spiking on the "n" output neuron, which correctly spikes with the last 10 input cases of "n" or 1 pixel of noise away from "n".

V. Discussion

We consider the energy used in this model of spiking JJs by measuring the voltage and current output of each JJ. The value we measure agrees, as expected, with the SFQ pulse energy of $I_C\varphi_0/2\pi$ for each of the JJs. This energy is about 70 zJ for the input JJs, 35 zJ for each of the synaptic JJs and approximately 150 zJ for each of the output JJs. Thus, if we consider only the pulse energies in our network of 39 JJs, this yields an energy dissipation of 2 aJ per classification.

It should be noted however, that this is not the total energy of the system. In a large-scale implementation, there would be an overhead of approximately 1 kW of cooling power consumed per watt dissipated at 4 K. Such a large scale implementation would be capable of processing~ 10^{18} spikes per second in the

ideal case where all energy is dissipated only by spike energy. In our current demonstration there are two other energy dissipation sources that should be considered. The first is that each of the "neuron" layers is DC biased. Since the bias level is not being changed between JJs, this bias current should be able to be shared with a common line on each layer (for example). The other source of power dissipation that should be considered is the resistor in the RL filter that is used to stabilize the circuit. In the current configuration, this resistor is somewhat problematic because it is a source of current dissipation. When the MJJ in the middle layer is firing, this resistor dissipates power that is similar to an additional SFQ pulse energy, which is not problematic, as this would represent a factor of two loss in spike processing. However, there is a small quiescent current that also flows through the resistor in the stabilization element in the present model. The current is a result of the *on* condition being represented using a constant current. By averaging the current and voltage across this resistor for the entire simulation, we observe a dissipation on this resistor of about 5 pW. This amount of dissipation in a stabilizing element will need to be changed if the networks are going to be implemented on a large scale. One potential solution would be to use a switched input where only one pulse is transmitted for the input *on* condition. In this case, the resistor should dissipate at most an additional SFQ pulse energy, which would not be problematic for energy scaling since this energy is already accounted for as the input spike energy.

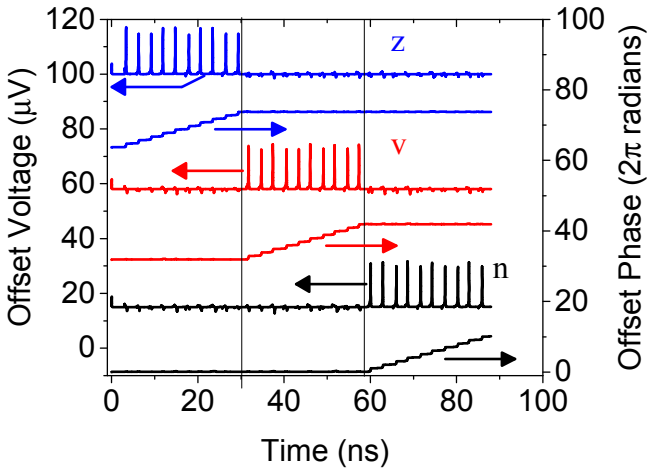

Fig. 5. Voltage spike and phase slip results of the trained 9 x 3 MJJ neural network. 10 spikes in each of the top middle and bottom output MJJs (in that order) corresponding to 100 % classification after training.

VI. Summary

We have demonstrated a 9 input pixel simulation of a neural network based on a combination of traditional JJs and tunable MJJs. We have shown that, with physically realistic parameters, we can solve a simple image recognition problem including recognition in the presence of 1 input pixel of noise. We used standard neural network training techniques to find the weights and biases which we mapped to the critical current in the MJJ elements. The total spiking energy for the computation of one input is roughly 2 aJ (≈ 2fJ with refrigeration), which is comparable to the energy of a single human synaptic event ≈ 10 fJ. To realize this potentially ultralow power computational scheme, we need to significantly reduce or eliminate the need for the stabilization resistor, which currently dominates the power consumption of the circuit. Given, the scalability of neural nets, we expect that large images can be processed in ns time scales using this technology.

Acknowledgment

The authors thank Manuel Castellanos Beltran for helpful circuit design discussions and the IARPA C3 program for partial support.

References

[1] Josephson, B.D.: "Possible new effects in superconductive tunnelling", Physics Letters, 1962, 1, (7), pp. 251-253

[2] Tinkham, M.: 'Introduction to superconductivity' (Dover, 1996. 1996)

[3] Chen, W., Rylyakov, A.V., Patel, V., Lukens, J.E., and Likharev, K.K.: 'Rapid Single Flux Quantum T-flip flop operating up to 770 GHz', Ieee Transactions on Applied Superconductivity, 1999, 9, (2), pp. 3212-3215

[4] Schneider, M.L., Donnelly, C.A., Russek, S.E., Baek, B., Pufall, M.R., Hopkins, P.F., Dresselhaus, P.D., Benz, S.P., and Rippard, W.H.: 'Ultra-low power artificial synapses using nano-textured magnetic Josephson junctions', Submitted, 2017

[5] Holmes, S., Ripple, A.L., and Manheimer, M.A.: 'Energy-Efficient Superconducting Computing-Power Budgets and Requirements', Ieee Transactions on Applied Superconductivity, 2013, 23, (3), pp. 10

[6] Segall, K., Guo, S.Y., Crotty, P., Schult, D., and Miller, M.: 'Phase-flip bifurcation in a coupled Josephson junction neuron system', Physica B, 2014, 455, pp. 71-75

[7] Hirose, T., Asai, T., and Amemiya, Y.: 'Pulsed neural networks consisting of single-flux-quantum spiking neurons', Physica C, 2007, 463, pp. 1072-1075

[8] Onomi, T., Kondo, T., and Nakajima, K.: 'High-speed single flux-quantum up/down counter for neural computation using stochastic logic', in Hoste, S., and Ausloos, M. (Eds.): '8th European Conference on Applied Superconductivity' (Iop Publishing Ltd, 2008)

[9] Yamanashi, Y., Umeda, K., and Yoshikawa, N.: 'Pseudo Sigmoid Function Generator for a Superconductive Neural Network', IEEE Transactions on Applied Superconductivity, 2013, 23, (3), pp. 1701004-1701004

[10] 'Whiteley Research Inc., WRSPICE, http://www.wrcad.com/wrspice.html'

[11] Van Duzer, T., and Turner, C.W.: 'Superconductive Devices and Circuits' (Prentice Hall, 1999. 1999)

[12] Russek, S.E., Donnelly, C.A., Schneider, M.L., Baek, B., Pufall, M.R., Rippard, W.H., Hopkins, P.F., Dresselhaus, P.D., and Benz, S.P.: 'Stochastic single flux quantum neuromorphic computing using magnetically tunable Josephson junctions', Proc. IEEE ICRC, 2016, pp. 1-5

[13] Kirichenko, D.E., Sarwana, S., and Kirichenko, A.F.: 'Zero Static Power Dissipation Biasing of RSFQ Circuits', IEEE Transactions on Applied Superconductivity, 2011, 21, (3), pp. 776-779

[14] Prezioso, M., Merrikh-Bayat, F., Hoskins, B.D., Adam, G.C., Likharev, K.K., and Strukov, D.B.: 'Training andoperation of an integrated neuromorphic network based on metal-oxide memristors', Nature, 2015, 521, (7550), pp. 61-64

[15] Nielsen, M.A.: 'Neural Networks and Deep Learning' (Determination Press, 2015. 2