Magnetic Tunnel Junction Enabled All-Spin Stochastic Spiking Neural Network

Gopalakrishnan Srinivasan, Abhronil Sengupta and Kaushik Roy
School of Electrical and Computer Engineering, Purdue University
Email: {srinivg, asengup, kaushik}@purdue.edu

***Abstract*—Biologically-inspired spiking neural networks (SNNs) have attracted significant research interest due to their inherent computational efficiency in performing classification and recognition tasks. The conventional CMOS-based implementations of large-scale SNNs are power intensive. This is a consequence of the fundamental mismatch between the technology used to realize the neurons and synapses, and the neuroscience mechanisms governing their operation, leading to area-expensive circuit designs. In this work, we present a three-terminal spintronic device, namely, the magnetic tunnel junction (MTJ)-heavy metal (HM) heterostructure that is inherently capable of emulating the neuronal and synaptic dynamics. We exploit the stochastic switching behavior of the MTJ in the presence of thermal noise to mimic the probabilistic spiking of cortical neurons, and the conditional change in the state of a binary synapse based on the pre- and post-synaptic spiking activity required for plasticity. We demonstrate the efficacy of a crossbar organization of our MTJ-HM based stochastic SNN in digit recognition using a comprehensive device-circuit-system simulation framework. The energy efficiency of the proposed system stems from the ultra-low switching energy of the MTJ-HM device, and the in-memory computation rendered possible by the localized arrangement of the computational units (neurons) and non-volatile synaptic memory in such crossbar architectures.**

***Keywords*—Spiking Neural Networks, Probabilistic Spiking Neuron, Stochastic STDP, Magnetic Tunnel Junction.**

I. Introduction

Neuromorphic computing paradigms have been widely adopted for a range of classification and recognition applications due to their superior performance over traditional machine learning algorithms. The state-of-the-art deep neural networks [1] trained on large supercomputing clusters are compute-intensive, thereby resulting in higher energy consumption. This has led to the exploration of alternative computing models that are inspired by the efficiency with which the human brain performs similar tasks. Spiking neural networks (SNNs) are one such cognitive computing paradigm, which process and encode information temporally in the form of action-potentials or spikes. The intrinsic sparse event-driven processing capability of SNNs results in computational efficiency, which can be exploited to achieve energy-efficient hardware implementations.

SNNs typically consist of a layer of input (pre) neurons connected by weighted synapses to the output (post) neurons as shown in Fig. 1. An input pre-neuronal (pre-synaptic) voltage spike is modulated by the synaptic strength to generate a resultant current, which influences the spiking behavior of the connected post-neuron. The spiking neurons and synapses have traditionally been emulated using deterministic models [2] requiring multibit-precision to store the parameters governing their dynamics. Custom CMOS implementations of deterministic neuronal and synaptic models are power- and area-intensive due to their mismatch with the underlying neuroscience processes. Emerging nanoelectronic devices including multilevel Ag-Si memristors [3] and spintronic devices [4] have been proposed to inherently mimic the deterministic dynamics. However, as the technology is scaled, these devices suffer from limited bit-precision while being significantly impacted by noise. Interestingly, neuroscience studies have shown that stochasticity plays a prominent role in the characteristics of biological neurons and synapses [5], [6]. Recent theoretical research efforts have demonstrated the applicability of stochastic SNNs in complex applications including Bayesian inference [7], [8].

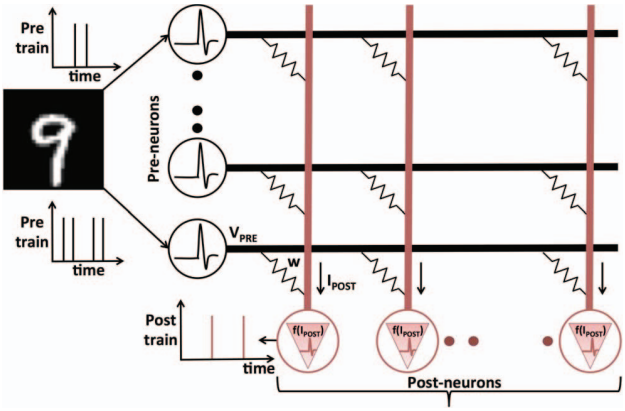

Fig. 1. Spiking neuromorphic architecture consisting of pre-neurons and post-neurons interconnected by synapses. The neurons encode information temporally as a series of voltage pulses over a period of time. An input pre-neuronal voltage spike (V_{PRE}) is modulated by the synaptic conductance (w) to generate a resultant current (I_{POST}), which impacts the spiking behavior of the connected post-neuron ($f(I_{POST})$).

We further note that the neurons and synapses implemented with emerging non-volatile device technologies are commonly interfaced together using a crossbar organization (explained in Section V), which is capable of performing *in-memory computations* as illustrated in Fig. 1. The crossbar architecture eliminates the data-transfer overhead imposed by conventional von-Neumann computing machines with decoupled memory and processing units, resulting in energy-efficient operation.

978-3-9815370-8-6/17/$31.00 ©2017 IEEE

In this work, we investigate a stochastic computing model for the spiking neurons and synapses, which exploits the noisy device characteristics. We explore a *probabilistic neuron* that conditionally fires an output (post-synaptic) spike depending on the input current, and a *stochastic binary synapse* that achieves plasticity using the probabilistic algorithm presented in [9]. According to the learning algorithm, the synaptic switching probability is dependent on the difference in the timing of the corresponding pre- and post-synaptic spikes.

We propose a three-terminal Magnetic Tunnel Junction (MTJ)-Heavy Metal (HM) multilayer device to mimic the stochastic neuronal and synaptic dynamics. We note that phase-change devices [10] have been demonstrated to realize similar stochastic computing models. However, they necessitate higher read and/or programming voltages (order of few volts) while the proposed MTJ-HM device can be operated at ultra-low voltages (order of millivolts). An MTJ possesses two stable resistance states, which can be switched by passing a charge current through the HM layer underneath the MTJ stack. As a *probabilistic neuron*, an input current through the HM conditionally switches the MTJ to produce an output spike. As a *stochastic synapse*, the MTJ is switched based on the timing of the pre- and post-synaptic spikes, by injecting an appropriate write current into the HM. We demonstrate the efficacy of a crossbar organization of our All-Spin SNN in digit recognition, using a device to system-level simulation framework. The energy efficiency of the proposed system arises from the ultra-low switching energy of the spintronic device, the non-volatility of ferromagnets that provides negligible leakage energy consumption, and the computational efficiency of the crossbar architecture.

Finally, we present a forced-learning methodology with neuronal spike-count based homeostasis (explained in Section IV) to realize efficient synaptic learning in stochastic SNNs. The proposed scheme achieves the following two-fold objectives. First, it enables individual neurons in the SNN to learn unique classes of input patterns. Second, it equalizes the spiking rate of all the neurons in the network, which effectively prevents just a few neurons from entirely dominating the learning process. The key contributions of our work are:

1) We propose a single spintronic device to emulate both the stochastic neuronal and synaptic dynamics.
2) We present a forced-learning scheme with neuronal spike-count based homeostasis to achieve efficient learning in stochastic SNNs.
3) We demonstrate the effectiveness of a crossbar arrangement of the proposed All-Spin SNN in digit recognition using a device-circuit-system simulation framework.

The rest of the paper is organized as follows. Section II provides the relevant background on stochastic SNNs. Section III describes the proposed spintronic neuron and synapse. Sections IV details the synaptic learning methodology. Section V illustrates the SNN crossbar architecture. Section VI outlines the simulation framework. Section VII presents the results and Section VIII concludes the paper.

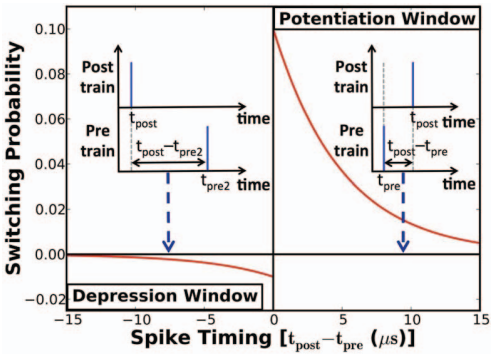

Fig. 2. Stochastic-STDP learning algorithm for achieving plasticity in a binary synapse. If a pre-synaptic spike (t_{pre}) causes the post-neuron to fire (t_{post}), the synapse should increase in strength (potentiate). The probability of potentiation is exponentially related to the spike timing difference ($t_{post}-t_{pre}$). However, if a pre-synaptic spike (t_{pre2}) occurs subsequent to the post-neuronal spike (t_{post}), the synapse must decrease in strength (depress).

II. Stochastic Spiking Neural Network

The stochastic SNN consists of a layer of input neurons connected via binary synapses to the probabilistic post-neurons. The input voltage spikes are modulated by the synaptic weights to produce a resultant current into the respective post-neurons. Each neuron spikes conditionally depending on the magnitude of the input current, and is subsequently reset back to its earlier state in the event of an output spike. The interconnecting binary synapses require a probabilistic algorithm to be able to acquire memory. We use the bio-inspired stochastic-spike timing dependent plasticity (STDP) learning mechanism presented in [9] to effectuate plasticity in binary synapses. According to the stochastic-STDP algorithm, the synaptic switching probability depends exponentially on the degree of temporal correlation between a pair of pre- and post-synaptic spikes as illustrated in Fig. 2. The spike timing information is effectively embedded in the switching probability of binary synapses to achieve plasticity. Stochastic synaptic potentiation (depression) is realized by conditionally switching to the high (low) conductance state when a pre-spike precedes (follows) the post-spike (inset of Fig. 2).

III. Spintronic (MTJ-HM) Neuron and Synapse

A. MTJ-HM Device Characteristics

An MTJ consists of a tunneling barrier (MgO) sandwiched between two ferromagnetic layers as shown in Fig. 3(a). The magnetization of one of the layers is fixed, and is referred to as the *pinned layer* (PL) while that of the *free layer* (FL) can be changed by passing a charge current through the MTJ. It exhibits two stable resistance states, and is said to be in the low (high) resistance state if the magnetization of the FL is parallel (anti-parallel) with respect to the PL. The magnetization dynamics of the FL at zero temperature is modeled using the *Landau-Lifshitz-Gilbert-Slonczewski* (LLGS) equation [11]. At non-zero temperatures, the FL dynamics are impacted by thermal noise that is incorporated into the LLGS framework by an additional thermal field [12], which gives rise to stochasticity in the MTJ switching dynamics. Recent experiments have shown that the FL with in-plane magnetic

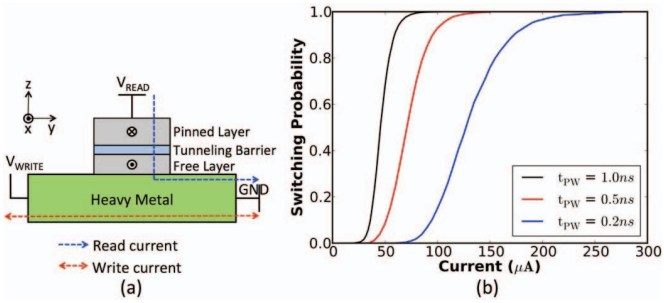

Fig. 3. (a) MTJ-HM device structure consisting of a tunneling barrier sandwiched between two ferromagnetic layers, namely, the pinned layer (PL) that is magnetostatically fixed and the free layer (FL) with in-plane anisotropy whose magnetization can be switched by passing a write current through the heavy metal (HM). The read current flowing between the terminals, V_{READ} and GND, is determined by the MTJ conductance. (b) Stochastic switching dynamics of the FL magnetization due to a charge current flowing through the HM for different programming pulse widths (t_{PW}).

TABLE I. MTJ-HM Device Simulation Parameters.

Parameters	Value
Free layer area	$\frac{\pi}{4}$ x 100 x 40 nm^2
Free layer thickness	1.2 nm
Heavy metal thickness, t_{HM}	2 nm
Saturation Magnetization, M_s	1000 KA/m [13]
spin-Hall Angle, θ_{SH}	0.3 [13]
Gilbert damping factor, α	0.0122 [13]
Energy Barrier, E_B	20 K_BT
Resistivity of HM, ρ_{HM}	200 $\mu\Omega.cm$ [13]
Write pulse width, t_{PW}	0.2 ns, 0.5 ns, 1.0 ns
Temperature, T	300 K

anisotropy can be switched by a charge current through a heavy metal (HM) layer underneath the MTJ stack as a result of spin-Hall effect [13], [14]. The MTJ-HM device can be readily interfaced with a resistive crossbar array due to the low resistance and switching current requirement of the HM. Device simulations were carried out using the parameters in Table 1 to obtain the MTJ-HM switching characteristics. Fig. 3(b) shows that the switching probability increases with the write current magnitude, while the dispersion in the switching characteristics is controlled by the corresponding pulse width.

B. Probabilistic MTJ-HM Neuron

Our proposed probabilistic spiking neuron [15] (shown in Fig. 4(a)) consists of a *reference-MTJ* in series with a *neuron-MTJ* that has an additional HM underlayer. It operates in a sequence of write, read, and reset cycles as indicated in the timing diagram in Fig. 4(b). During the write cycle spanning a duration of $0.5ns$, an input charge current that is generated by the pre-neuronal voltage spikes based on the corresponding synaptic conductances, is passed through the HM by enabling the write-access transistor (M_W). This conditionally switches the orientation of the FL magnetization of the *neuron-MTJ* as illustrated in Fig. 5. The resulting change in the conductance of the *neuron-MTJ* is sensed by passing a read current through the *reference-MTJ*. The voltage at the input of the inverter drops due to the voltage-divider action resulting in an output spike. Note that the write current path is gated off during the read mode of operation. A subsequent reset operation is carried out by injecting a sufficient current through the HM in the opposite direction.

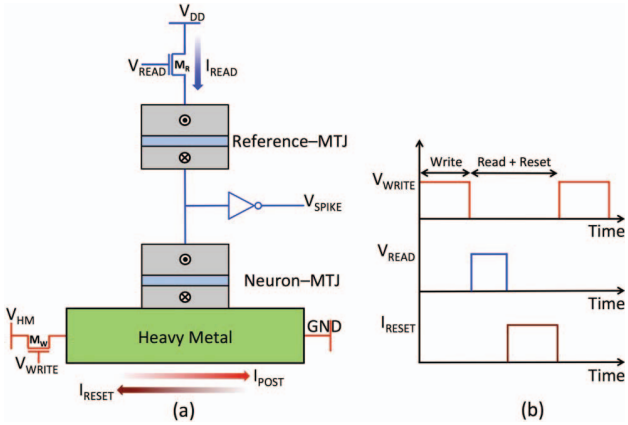

Fig. 4. (a) Schematic of the spintronic neuron consisting of a *reference-MTJ* and a *neuron-MTJ*, both of which are initialized to the high-resistance state. During the write cycle, the input current (I_{POST}) flowing through the HM conditionally switches the resistance state of the *neuron-MTJ*. This is sensed by passing a read current (I_{READ}) through the *reference-MTJ*, which discharges the input node of the inverter to produce a rising transition at its output (V_{SPIKE}). The *neuron-MTJ* is subsequently reset by passing a current (I_{RESET}) through the HM in the opposite direction. (b) Timing diagram illustrating the various modes of operation of the MTJ-HM neuron.

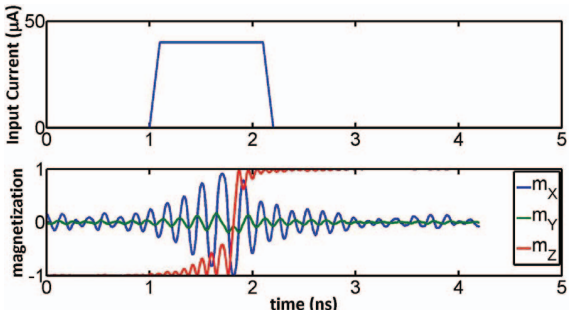

Fig. 5. Variation in the FL magnetization components upon the application of a $40\mu A$ current pulse. The component m_Z switches from $-z$ to $+z$ direction.

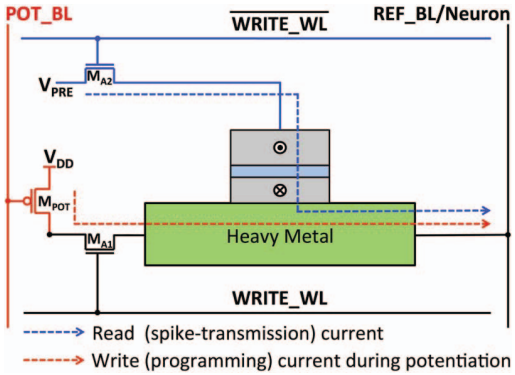

Fig. 6. Schematic of the MTJ-HM synaptic bitcell. During the read mode, an input voltage spike (V_{PRE}) is modulated by the MTJ conductance to produce a resultant current into the post-neuron. The write wordline (WRITE_WL) is asserted high following a post-neuron spike, which activates the write current path. The reference bitline (REF_BL) is connected to ground while the potentiation bitline (POT_BL) is driven by the programming voltage, which is sampled by M_{POT} to generate the required write current.

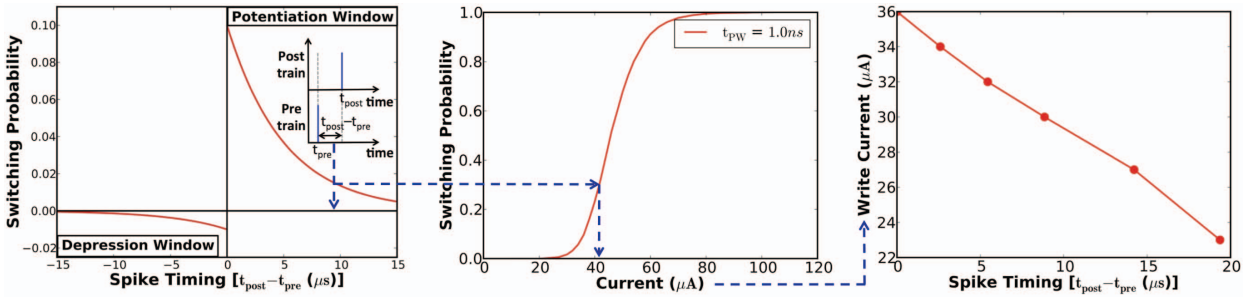

Fig. 7. Illustration of the plasticity mechanism for an MTJ-HM synapse. The stochastic STDP algorithm specifies the synaptic switching probability based on spike timing. The MTJ-HM device simulations indicate the write current necessary to switch the MTJ with the corresponding probability. This is used to obtain a correlation between spike timing and the write current required to achieve stochastic learning in the proposed spintronic synapse.

C. Stochastic MTJ-HM Synapse

Our proposed stochastic synapse [9] (shown in Fig. 6) has two primary modes of operation, namely, the read and the programming modes. During the read mode, an input pre-synaptic voltage spike gets modulated by the MTJ conductance to generate a current into the spintronic neuron. During the programming mode, a suitable charge current is passed through the HM to conditionally switch the MTJ in accordance with the stochastic-STDP algorithm, which is illustrated in Fig. 7 for synaptic potentiation. The stochastic learning algorithm specifies the switching probability of a synapse based on the relative difference in the timing of the corresponding pre- and post-synaptic spikes. The MTJ-HM device simulations stipulate the write current required to switch the MTJ with the determined probability. This provides a correlation between spike timing and the proportionate write current necessary for stochastic synaptic learning.

A CMOS implementation of the synaptic programming circuit necessary to generate the appropriate write current was presented and validated in [9], and is illustrated here for synaptic potentiation. The potentiation bitline (POT_BL) is driven to a reset potential at the instant of a pre-synaptic spike (t_{pre}). It subsequently charges linearly with time. The write wordline (WRITE_WL) is enabled when the connected post-neuron fires an output spike (t_{post}). The transistor M_{POT} then samples the voltage on POT_BL (proportional to $t_{post} - t_{pre}$), and generates the required write current through the HM. We note that synaptic depression can similarly be achieved by passing an appropriate write current through the HM in the opposite direction.

IV. Learning Methodology

We propose a stochastic-STDP based forced-learning methodology, with neuronal spike-count based homeostasis, to achieve plasticity in our All-Spin SNN. First, we describe the network topology used for pattern recognition and identify the key processes required for efficient learning. We subsequently illustrate the efficacy of the proposed methodology in realizing the vital plasticity mechanisms.

We use the hierarchical SNN topology presented in [16] for pattern recognition, which consists of the input, excitatory, and inhibitory layers as shown in Fig. 8. Every pixel in the

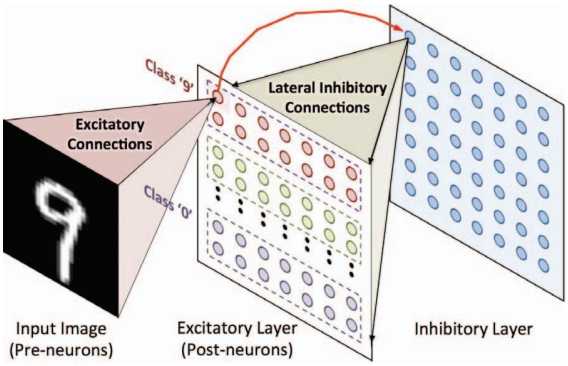

Fig. 8. SNN topology for pattern recognition consisting of the input, excitatory, and inhibitory layers. The input neurons are fully connected to the post-neurons in the excitatory layer. These neurons are further segmented into different clusters, each of which is trained on a specific class of input patterns. The excitatory neurons are connected to the corresponding inhibitory neurons in a one-to-one manner. Each of these neurons inhibits all the excitatory neurons except the forward connected one.

image pattern to be recognized constitutes an input neuron, whose spiking rate is proportional to the corresponding pixel intensity. The input neurons are fully connected to the neurons in the excitatory layer. It is important to note that plasticity is effected solely on the excitatory synaptic connections. The excitatory neurons are connected to the corresponding inhibitory neurons in a one-to-one manner, each of which inhibits all the excitatory neurons except the one from which it received a forward connection. Lateral inhibition plays an important role in differentiating the receptive field of every neuron, which causes the connected synapses to encode a unique representative input pattern in the corresponding weights.

In an effort to further the efficiency of synaptic learning, we explore a forced-learning methodology, where the excitatory neurons are segmented into different clusters. The neurons in each cluster (for instance, cluster '9' in Fig. 8) are trained on input patterns belonging to a specific class ('9') fixed *a priori*. For a given input training pattern, the stochastic-STDP driven synaptic updates are effected only on the neuronal cluster that is pre-assigned to learn the specific input class. The proposed scheme effectively *forces* the neurons in individual clusters (inter-cluster neurons) to learn distinct classes of input patterns. At the same time, lateral inhibition causes the neurons within a cluster (intra-cluster neurons) to learn varying

representations of a specific class of input patterns. We note that the forced-learning scheme can be implemented without any hardware overhead as will be explained in Section V.

In addition to lateral inhibition, homeostasis [17] is another key mechanism that is critical to achieving efficient synaptic learning in SNNs. During the training phase, as the input patterns are presented to a neuronal cluster, few of the excitatory neurons begin to fire. As the corresponding synapses are strengthened, these neurons spike dominantly and continue to inhibit the rest of the excitatory neurons. Homeostasis attempts to equalize the spiking rate of all the neurons, thereby suppressing the dominance of few individual neurons. We present a neuronal spike-count based homeostasis implementation to achieve uniform spiking rate for all the excitatory neurons. According to this scheme, a probabilistic (spintronic) neuron is disconnected from the learning process, once it fires a definite number of spikes during the training period. This provides an opportunity for the previously inactive neurons to achieve plasticity at a comparable spiking rate.

V. Spintronic SNN Implementation

We present a crossbar arrangement of the spintronic neurons and synapses as shown in Fig. 9. The input spikes fired by the pre-neurons drive the read-current path in the corresponding row of MTJ-HM synapses. At any given instant of time, an input pre-neuronal voltage spike (V_i) is modulated by the synaptic (MTJ) conductance (G_{ij}) to produce a resultant current (I_j) into the heavy metal of the spintronic neuron.

$$I_j = \sum_i \frac{G_{ij} V_i}{1 + \frac{\sum_i G_{ij}}{G_{HM}}} \quad (1)$$

It can be seen that the total current is lowered due to the finite HM resistance ($\frac{1}{G_{HM}}$). The neurons spike conditionally depending on the magnitude of the input current. Every lateral inhibitory line is driven to a negative voltage for a specific time period following the corresponding post-neuronal spike, so that the inhibitory currents are subtracted. Homeostasis is realized by recording the output spikes, and isolating a neuron from the crossbar once the spike-count exceeds a definite threshold. This is accomplished by suitably controlling the gate voltage (V_{WRITE}) of the write-access transistor (M_W), which connects a post-neuron to the crossbar. In order to implement the forced-learning algorithm, we propose designing similar crossbar arrays for each neuronal cluster that needs to be trained on a specific class of input patterns.

VI. Simulation Framework

The crossbar implementation of our All-Spin SNN is functionally modeled in a Python-based simulator termed BRIAN [18], to recognize handwritten digits from the MNIST [19] dataset. The 28×28 pixels in an MNIST digit pattern constitute the neurons in the input layer. The information contained in each pixel is converted to a Poisson-distributed spike train depending on the corresponding pixel intensity. At every simulation timestep, the resultant currents generated by the input spikes are used to effect the probabilistic spiking of

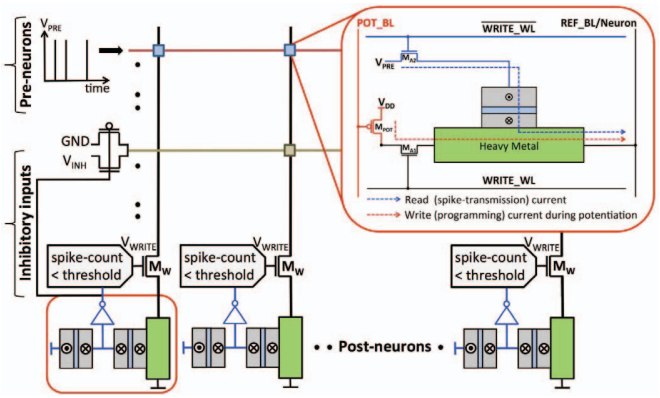

Fig. 9. Crossbar arrangement of the MTJ-HM neurons and synapses. Each input pre-neuronal voltage spike (V_{PRE}) gets modulated by the MTJ conductance in the corresponding row of synapses to produce a resultant current into the post-neurons, causing them to conditionally fire a spike. A post-neuronal spike drives a negative voltage (V_{INH}) on the corresponding inhibitory line that is otherwise pulled to ground. Homeostasis is implemented by counting the output spikes, and disconnecting a neuron from the crossbar as soon as the spike-count exceeds a threshold.

the spintronic neurons, based on the MTJ-HM switching characteristics (Fig. 3(b)) obtained from device simulations. Plasticity is achieved in a spintronic synapse by recording the pre- and post-synaptic spike times, and switching it conditionally as stipulated by the stochastic-STDP learning algorithm (Fig. 2). The simulation framework is further enhanced to implement the stochastic-STDP based forced-learning scheme and the neuronal spike-count based homeostasis mechanism. The stochastic SNN is trained using examples from the MNIST training dataset. The energy expended by the neuron (write, read, and reset circuit paths in Fig. 4(a)) and the synapse (synaptic bitcell in Fig. 6) during the training phase is determined in BRIAN using the results from device and circuit (SPICE) simulations.

At the end of the training period, each neuron learns to spike for a pre-assigned class of input patterns. The classification accuracy of the stochastic SNN is subsequently estimated using the entire MNIST testing dataset. Each test pattern is predicted to belong to the class associated with the neuronal cluster with the highest average spike-count during the simulation period. The classification is accurate, if the actual class of the test pattern matches with that predicted by the stochastic SNN. The classification accuracy together with the neuronal and synaptic energy consumption are used to evaluate the effectiveness of our All-Spin SNN.

VII. Results

We trained our proposed All-Spin SNN consisting of 200 excitatory neurons using 800 examples from the MNIST training dataset. Fig. 10(a) demonstrates the efficacy of the forced-learning methodology in differentiating the receptive field of each individual neuron in the SNN. The network of 200 neurons provided a classification accuracy of 71% on the MNIST testing dataset. Fig. 10(b) shows that the classification accuracy of the All-Spin SNN can be enhanced by increasing the number of excitatory neurons.

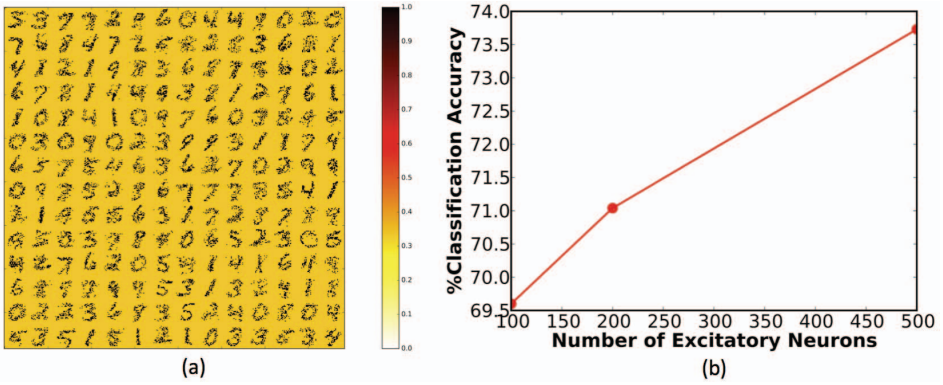

Fig. 10. (a) Final conductance states of the 28×28 stochastic synapses connecting the input to each of the 200 excitatory neurons (arranged in a 14×14 grid) after 2000 training iterations. The ratio of the minimum to maximum synaptic conductance used in the system-level simulations is 1:3. (b) Classification accuracy of the SNN versus the number of excitatory neurons.

Next, we present the average energy consumption of an All-Spin SNN of 200 excitatory neurons during the training period. The spintronic neuron expended an average of $1.16fJ$ per time-step for the write, read, and reset operations. The spintronic synapse, on the other hand, consumed a maximum of $36fJ$ per spiking event for realizing the stochastic-STDP learning algorithm. This demonstrates the energy efficiency of the proposed spintronic neuron and synapse compared to CMOS implementations that are reported to consume energy in the order of pJ [20], [21].

VIII. Conclusion

Emerging spintronic devices, with their non-volatility and ultra-low switching energy, offer immense potential for achieving energy-efficient realizations of spiking neuromorphic computing paradigms. However, such devices exhibit noisy behavior due to the impact of thermal noise, which aggravates as the device dimensions are scaled. In this work, we explored a brain-inspired stochastic computing model for the neurons and synapses constituting a spiking neural network (SNN), which exploits the noisy device characteristics. We showed that a single spintronic device, namely the MTJ-HM heterostructure, which exhibits probabilistic switching behavior naturally mimics the stochastic neuronal and synaptic characteristics. We proposed a forced-learning methodology to achieve efficient synaptic plasticity in such stochastic SNNs. Our simulations of a crossbar arrangement of the All-Spin SNN trained for digit recognition indicate that the spintronic neurons and synapses expend a few fJ of energy, which is three orders of magnitude lower compared to CMOS-based implementations.

Acknowledgments

The work was supported in part by, Center for Spintronic Materials, Interfaces, and Novel Architectures (C-SPIN), a MARCO and DARPA sponsored StarNet center, by the Semiconductor Research Corporation, the National Science Foundation, Intel Corporation, and the DoD Vannevar Bush Fellowship.

References

[1] A. Krizhevsky et al. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process Syst.* 1097–1105 (2012).
[2] Ghosh-Dastidar, S. & Adeli, H. Spiking neural networks. *Int. J. Neural Syst.* 19, 295–308 (2009).
[3] S. H. Jo et al. Nanoscale Memristor Device as Synapse in Neuromorphic Systems. *Nano Lett.* 10, 1297–1301 (2010).
[4] A. Sengupta et al. Spin-orbit torque induced spike-timing dependent plasticity. *Appl. Phys. Lett.* 106, 093704 (2015).
[5] B. B. Averbeck et al. Neural correlations, population coding and computation. *Nat. Rev. Neurosci.* 7, 358–366 (2006).
[6] T. Branco et al. The probability of neurotransmitter release: variability and feedback control at single synapses. *Nat. Rev. Neurosci.* 10, 373–383 (2009).
[7] B. Nessler et al. Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. *PLoS Comput. Biol.* 9, e1003037 (2013).
[8] E. Wallace et al. Emergent oscillations in networks of stochastic spiking neurons. *Plos one* 6,e14804 (2011).
[9] G. Srinivasan et al. Magnetic Tunnel Junction Based Long-Term Short-Term Stochastic Synapse for a Spiking Neural Network with On-Chip STDP Learning. *Sci. Rep.* 6, 29545 (2016).
[10] T. Tuma et al. Stochastic phase-change neurons. *Nat. Nanotechnol.* 11, 693–699 (2016).
[11] J. C. Slonczewski. Conductance and exchange coupling of two ferromagnets separated by a tunneling barrier. *Phys. Rev. B* 39, 6995 (1989).
[12] W. Scholz et al. Micromagnetic simulation of thermally activated switching in fine particles. *J. Magn. Magn. Mater.* 233, 296–304 (2001).
[13] C. F. Pai et al. Spin transfer torque devices utilizing the giant spin Hall effect of tungsten. *Appl. Phys. Lett.* 101, 122404 (2012).
[14] I. M. Miron et al. Perpendicular switching of a single ferromagnetic layer induced by in-plane current injection. *Nature* 476, 189–193 (2011).
[15] A. Sengupta et al. Magnetic Tunnel Junction Mimics Stochastic Cortical Spiking Neurons. *Sci. Rep.* 6, 30039 (2016).
[16] P. U. Diehl et al. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* 9, 1–9 (2015).
[17] G. G. Turrigiano et al. Hebb and homeostasis in neuronal plasticity. *Curr. Opin. Neurobiol.* 10, 358–364 (2000).
[18] D. Goodman et al. Brian: a simulator for spiking neural networks in Python. *Front. Neuroinform.* 2, 1–10 (2008).
[19] Y. Lecun et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 2278–2324 (1998).
[20] P. Livi et al. A current-mode conductance-based silicon neuron for address-event neuromorphic systems. *IEEE international symposium on circuits and systems (ISCAS),* Taipei, Taiwan (2009, May 24).
[21] J. s. Seo et al. A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons. *IEEE Custom Integrated Circuits Conference (CICC),* San Jose, CA (2011, Sep 19).