Leveraging Probabilistic Switching in Superparamagnets for Temporal Information Encoding in Neuromorphic Systems

Kezhou Yang, Dhuruva Priyan G M, and Abhronil Sengupta, Member, IEEE

Abstract—Brain-inspired computing—leveraging neuroscientific principles underpinning the unparalleled efficiency of the brain in solving cognitive tasks—is emerging to be a promising pathway to solve several algorithmic and computational challenges faced by deep learning today. Nonetheless, current research in neuromorphic computing is driven by our well-developed notions of running deep learning algorithms on computing platforms that perform deterministic operations. In this article, we argue that taking a different route of performing temporal information encoding in probabilistic neuromorphic systems may help solve some of the current challenges in the field. The article considers superparamagnetic tunnel junctions as a potential pathway to enable a new generation of brain-inspired computing that combines the facets and associated advantages of two complementary insights from computational neuroscience: 1) how information is encoded and 2) how computing occurs in the brain. The hardware-algorithm co-design analysis demonstrates 97.41% accuracy of a state-compressed 3-layer spintronics-enabled stochastic spiking network on the MNIST dataset with high spiking sparsity due to temporal information encoding.

Index Terms—Magnetic tunnel junction (MTJ), neuromorphic computing, stochasticity.

I. INTRODUCTION

Deep learning has undergone unprecedented growth in the past decade and has witnessed success in a plethora of applications. However, with the scaling complexity of the problem space and with the ever-growing dimensions of data, computational expenses to train and implement such artificial intelligence (AI) systems have also grown beyond limits. Driven by this motivation, “neuromorphic computing” attempts to decode the operation of the biological brain by mimicking the core functionalities in the underlying algorithms and hardware substrate. In particular, we focus on the more bi-plausible “spiking” neural/synaptic computing models in this text due to its promise of enabling low-power, asynchronous “compute only when needed” neuromorphic hardware. We will refer to such a computing model as spiking neural networks (SNNs) for the remainder of this text. While SNNs have shown initial promise as a low-power, event-driven alternative computing paradigm, significant challenges remain from both the algorithms and hardware perspective to ensure scalability in terms of key performance metrics, such as recognition accuracy, hardware power, energy, and area efficiency. Most prior studies have used smaller subproblems or have converted nonspiking deep neural networks (DNNs) to SNNs [1]—a nonoptimal approach in demonstrating the abilities of SNNs. Currently, SNNs remain very similar to nonspiking networks with the analog neural computation in DNNs distributed as binary information over time in the case of a spiking neuron—with the temporal aspect remaining largely unexploited. This has significantly limited SNN efficiency in large-scale problems [2].

In order to address these limitations, we formulate our solution against two complementary backdrops.

1) Information Encoding (Goal—Enhanced Sparsity and Reduced Latency): The vast majority of SNN algorithm formulations have been based on rate coding [3], [4] where the neuron output is encoded in the spike rate, i.e., the total number of spikes generated in a sufficiently long time duration. However, in temporal encoding, the precise time duration required to spike is believed to encode the neuron output information. The principal advantages of using temporal encoding [5] for modeling spiking behavior are multiple. Since information is now transmitted in precise spike timings instead of the signal rate, such neural codes can be sparse and much faster to avoid the temporal-averaging effect.

2) Computing Paradigm (Goal—State-Compressed Hardware): The computing perspective is motivated by a bottom-up hardware viewpoint that emerging technologies like spintronics exhibit stochastic switching behavior (due to thermal noise) at room temperature, specially at aggressively scaled dimensions [6], [7]. The potential benefits of such a computing framework from the hardware implementation perspective is that they allow multilevel neural/synaptic state compression to a single bit (in turn, leading to scaled device implementations) due to the additional probabilistic encoding of information. However, such stochastic SNNs have been mostly utilized in the rate encoding framework.

In order to leverage the benefits of increased information capacity in SNNs for enhanced power, latency, and energy metrics and simultaneously to utilize the advantages of state-compressed hardware enabled by these nanomagnetic devices, the article explores a device-algorithm co-design approach—where we explore the implementation of spintronics enabled stochastic SNNs bearing temporal domain encoding of information. Section II discusses basic device preliminaries of magnetic tunnel junction (MTJ) devices. Section III outlines the novel device physics enabling the dynamic temporal control of the stochastic magnetization dynamics that are leveraged in Section IV to formulate algorithms for stochastic SNNs with temporally encoded spikes. Recognition accuracy and spiking sparsity advantages for fully connected network architectures on the MNIST dataset are reported in Section IV. Section V concludes this article with potential future research directions.

II. MAGNETIC TUNNEL JUNCTION AS STOCHASTIC COMPUTING ELEMENT

MTJ is a fundamental device building block of spintronic hardware systems. A typical MTJ consists of two ferromagnetic layers and a sandwiched oxide layer. One of the ferromagnetic layers is called
Fig. 1. Device preliminaries: magnetization components for a magnet with anisotropy along the z-direction is shown during a switching process. For a superparamagnetic device, the switching is spontaneous as shown by the noisy switching characteristics. However, the device lifetimes can be modulated by the external current stimulus, $I$, resulting in a sigmoid device switching rate, $R$, variation with external current magnitude. The green transients represent the plot without time averaging.

pinned layer (PL) because its magnetization direction is pinned and does not change during operation. The other ferromagnetic layer is called free layer (FL) since its magnetization can be switched freely by an external stimulus, such as spin current or magnetic field. The state of the device is determined by the relative orientation of the two ferromagnetic layers. The device is in anti-parallel (AP)/parallel (P) state if the two ferromagnetic layers have opposite/same magnetization direction. The device possesses a higher resistance in the AP state than in the P state. Energy barrier height determined by device volume and anisotropy stabilizes the two states.

The Landau–Lifshitz–Gilbert (LLG) equation with a spin torque term is used to characterize the probabilistic switching of an MTJ device [8]

$$\frac{d\hat{m}}{dt} = -\gamma (\hat{m} \times \mathbf{H}_{\text{eff}}) + \alpha (\hat{m} \times \frac{d\hat{m}}{dt}) + \frac{1}{qN_s} (\hat{m} \times \mathbf{I}_s \times \hat{m}) \tag{1}$$

in which $\hat{m}$ is the FL magnetization unit vector, $\gamma = (2\mu_B\rho_0/\hbar)$ is the gyromagnetic ratio, $\alpha$ is Gilbert’s damping ratio, $\mathbf{H}_{\text{eff}}$ is the effective magnetic field, $N_s = (M_s V/\mu_B)$ is the number of spins in FL of volume $V$ (where $M_s$ is saturation magnetization and $\mu_B$ is Bohr magnetron), $q$ is the charge of a single electron, and $\mathbf{I}_s$ is the input spin current. Thermal noise is included by adding an additional thermal field, $\mathbf{H}_\text{thermal} = \sqrt{\langle I_s^2 \rangle} + \alpha^2 (2k_B T_K/\gamma \mu_0 M_s V_0) G_{0,1}$, where $G_{0,1}$ is the Gaussian distribution with zero mean and standard deviation, $k_B$ is the Boltzmann constant, $T_K$ is the absolute temperature, and $\delta_t$ is the simulation time step.

Recently there has been a lot of interest in superparamagnetic devices for unconventional computing. In essence, these are aggressively scaled nanomagnetic MTJs in the sub-10$\text{K}_\theta T_K$ barrier height regime where the magnet loses its nonvolatility and does not need to be triggered by a pulse for state transitions (see Fig. 1). The thermal noise becomes significant and is large enough to overcome the barrier height, resulting in spontaneous stochastic switching behavior. However, the metastable state transitions can be modulated by an external current and the time-averaged response of the device, $R = (\tau_{\text{AP}}/\tau_P + \tau_{\text{AP}})$ (where $\tau_P$ and $\tau_{\text{AP}}$ are device lifetimes in the P and AP states, respectively) has a nonlinear sigmoid response that can be utilized for stochastic spiking neuron functionalities [9]. The main advantage of transitioning to a superparamagnetic system would lie in the faster operating speeds and asynchronous operation [10]. However, careful peripheral circuitry design, sensitivity to noise, and variations remain open challenges [10]. In addition to neuro-morphic applications [7], [10], [11], [12], [13], [14], stochasticity inherent in magnetic devices (superparamagnets or higher barrier height magnets) have been leveraged to implement true random number generators [15], and even for other unconventional computing platforms, such as Ising computing, quantum-inspired algorithms, combinatorial optimization problems, on-chip temperature sensors, among others [6], [16], [17], [18].

While the intrinsic temporal dynamics of superparamagnets have been utilized in certain applications like Ising computing, the vast majority of neuromorphic SNN applications have primarily leveraged the superparamagnetic device characteristics in the rate encoding regime, i.e., the continuous-time dynamic behavior of superparamagnets have been ignored and the time-averaged behavior has been used from the computing perspective. This leads us to the question—Can the unique probabilistic switching behavior of superparamagnetic devices be utilized for temporal information encoding in stochastic SNNs?

III. LEVERAGING THE DYNAMIC TEMPORAL BEHAVIOR OF MTJs

In order to design a magnetic device where the intrinsic physics is able to support temporal information encoding, one needs to precisely control the device lifetimes $\tau_P$ and $\tau_{\text{AP}}$. This is difficult in a superparamagnet under sole external current stimulation. As shown in Fig. 1, the external current magnitude and direction controls the time-averaged firing rate of the device and both the device lifetimes get modulated together with a change in the external current magnitude.

However, as explained in (1), the magnetization dynamics is a function of both external current and external magnetic field which opens up the possibility of tuning the two device lifetimes by two separate independent control knobs. When an external “write” voltage is applied to the MTJ (resulting in spin torque) along with an external magnetic field, the lower MTJ resistance in the P state results in much larger modulation of $\tau_P$ than $\tau_{\text{AP}}$ due to an external voltage. Consequently, the external spin current can be used to control $\tau_P$. On the other hand, the magnetic field can be used to tune $\tau_{\text{AP}}$ by manipulating the energy profile. In this manner, under certain conditions [19], independent control of $\tau_P$ and $\tau_{\text{AP}}$ can be realized by adjusting the externally applied magnetic field and current. Recent experiments [20] and theoretical modelling [19] have shown that such a controlling scheme can be realized in a CoFeB MTJ stack within a range of applied field and current.

However, on-chip external magnetic field control of nanoscale devices is not promising from the perspective of scalability and power consumption [21]. A potential alternative path can be to design novel device structures exploiting emerging emerging physics like the magnetoelectric effect [22]. Recent work [21] explored a three-terminal magneto-electric (ME) MTJ device concept where voltage applied across an ME layer ($V_{\text{ME}}$) lying underneath the MTJ was used to mimic the effect of an effective magnetic field while voltage across the MTJ stack ($V_I$) was used to induce an external spin current, as shown in Fig. 2(a). ME effect was modeled by considering the effect of an external magnetic field acting on the magnet, whose magnitude is directly proportional to the applied voltage [23], [24], with the proportionality factor ($\alpha_{\text{ME}}$) being a material property. The device modeled at room temperature (300 K) has an elliptic ferromagnetic layer, the size of which is 17 nm in width, 42.5 nm in length, and 0.8 nm in thickness. Tunnel magnetoresistance (TMR) ratio of the device is 200%. The saturation magnetization is 750 KA/m. The Gilbert damping ratio is chosen to be 0.0122. The ME layer has a thickness of 5 nm and ME constant of $5 \times 10^{-9}$ s/m [10], [24]. The device state can be detected by a circuit shown in Fig. 2(b). The transistor working in saturation region provides a constant current, $I_{\text{Total}}$. $V_I$ is the input voltage applied to the MTJ. The MTJ resistance modulates the current flowing through the MTJ, $I_{\text{MTJ}}$, leading to the control of current flowing through the load resistance $R_L$. As a result, the output current, $I_{\text{Output}} = I_{\text{Total}} - I_{\text{MTJ}}$, will be an indicator of the...
state equals the duration of a "timestep" in the system. Note that the duration of "timestep" will be determined by circuit and architecture level constraints for simulating the SNN. If we interpret the device AP state as the "spike" of the neuron, then the average time to fire for that neuron will be given by $\tau_P$, which can be controlled by the external neuron input $V_2$. For an SNN inferring data based on temporal encoding, this time to fire will dictate the winning neuron. The neuron which fires earliest will be interpreted as the winning class and is based on time-to-first-spike encoding. Note that the SNN can be turned off after the first spike, thereby resulting in significant sparsity and latency benefits. Such a fine-grained control of time to fire is not possible in the case of stochastic magnetic devices driven by only a single external input signal since both the device lifetimes will be modulated together. It is also worth mentioning here that while our proposal is based on the ME-MTJ device, the formulation can be easily extended to experimentally demonstrated stochastic devices operating under the influence of external spin current and magnetic field [19], [20]. In order to train the network, let us assume that we set the winning class neuron to fire at timestep $t_1$ while the other neurons target a firing time $t_2$. In order to infer with sufficient confidence margin, $\Delta t = t_2 - t_1$ should be reasonably high. Note that $\Delta t$, $t_1$, and $t_2$ are hyperparameters for our algorithm and user specified. In this work, we used a value of $t_1 = 1$ ns and $t_2 = 300$ ns.

IV. ALGORITHM FORMULATION

Fully connected neural network architectures with stochastic temporal encoding were trained on the MNIST dataset [25] based on algorithmic formulations described next. Since the real-time device lifetimes follow an exponential distribution in the low current regime [26], we utilize the Kullback–Leibler (KL) divergence to model the loss function. Assuming the target average device lifetime in the P state to be $\lambda$ and the expected device lifetime due to the external input to be $z$, the KL divergence between the expected and target spike probability distributions are given by

$$ L = \sum_{a \in A} \frac{1}{\lambda} z^{-\frac{1}{\lambda}} \log \left( \frac{z^{\frac{1}{\lambda}}}{\lambda} \right) $$

where $\lambda$ is the probability space. From a network perspective, each neuron receives the weighted summation of synaptic inputs ($\sum_i w_i l_i$) as the input voltage $V_2$ [see Fig. 3(a)]. Note that the output current in the spike detection circuit [see Fig. 2(b)] can be used to charge a capacitor till the input neuron device spikes, thereby converting the timing information to an analog voltage input for the next layer. Assuming the intrinsic device function mapping from the synaptic dot product to the average P state device lifetime to be $g(.)$ [which can be formulated by the exponential variation shown in Fig. 3(b)]

$$ z = g \left( \sum_i w_i l_i \right) = g(V_2). $$

It is worth mentioning here that the output $z$ represents the average value of P-state device lifetime under the influence of $V_2$, although the real-time characteristics follow an exponential distribution [26]. The operating voltage range of the device is also chosen properly [Fig. 3(b)] such that the change in $\tau_P$ is much larger than $\tau_AP$ (assumed constant equal to spike duration in the algorithm formulation) within this working range.

Using gradient descent, the weights of the network can be learned through the following relations:

$$ w = w - \alpha \left( \frac{\partial L}{\partial w} \right) \frac{\partial L}{\partial w} = \frac{\partial L}{\partial z} \frac{\partial z}{\partial w} $$

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
There is no dependency of spiking activity on network scaling. While it depends on the average firing rate but rather on the time of firing, since information transmission from one layer to another does not involve improvement over rate encoding methods and substantiates the consideration for both the hidden and output layers. This is a significance achieved near-maximum values with only 2–3 spikes being observed to have a test accuracy of 97%. Simulated accuracy of the hardware MTJ network approaches the baseline software accuracy with time-to-2nd/3rd spike of the winning neuron.

As demonstrated in Fig. 3(c), the accuracy of the hardware network implementations [3], [4]. Similar observations were achieved when the network was scaled to a 3-layer architecture with 784 × 400 × 10 neurons. The network had a test accuracy of 97.41%, at par with iso-architecture standard deterministic networks (a conventional nonspiking network with such architectures along with providing accuracies at par with other implementations [3], [4].

V. DISCUSSION AND OUTLOOK

The article presents a unique perspective of designing efficient stochastic neuromorphic systems with temporal information encoding driven by an interdisciplinary perspective from devices to brain-inspired algorithm development. The work provides algorithmic formulations to leverage the stochastic temporal device characteristics of superparamagnetic devices and provides proof-of-concept demonstrations through extensive simulations. Such an end-to-end co-design effort to leverage unique properties of neuromorphic computing is an ideal fit for application drivers characterized by temporal information (for instance, sparse data collected by event-driven sensors [28], [29], among others).

REFERENCES

[1] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, “Going deeper in spiking neural networks: VGG and residual architectures,” Front. Neurosci., vol. 13, p. 95, Mar. 2019.

[2] M. Davies et al., “Advancing neuromorphic computing with Loihi: A survey of results and outlook,” Proc. IEEE, vol. 109, no. 5, pp. 911–934, May 2021.

[3] W. Severa, C. M. Vineyard, R. Dellana, S. J. Verzi, and J. B. Aimone, “Training deep neural networks for binary communication with the whetstone method,” Nat. Mach. Intell., vol. 1, no. 2, pp. 86–94, 2019.

[4] S. B. Shrestha and G. Orchard, “SLAYER: Spike layer error reassignment in time,” in Proc. Int. Conf. Adv. Neural Inf. Process. Syst., vol. 31, 2018, pp. 1419–1429.

[5] W. Guo, M. E. Fouda, A. M. Eltawil, and K. N. Salama, “Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems,” Front. Neurosci., vol. 15, p. 212, Mar. 2021.

[6] K. Y. Camsari, R. Faria, B. M. Sutton, and S. Datta, “Stochastic p-bits for invertible logic,” Phys. Rev. X, vol. 7, no. 3, 2017, Art. no. 31014.

[7] A. Sengupta, P. Panda, P. Wijesinghe, Y. Kim, and K. Roy, “Magnetic tunnel junction mimics stochastic cortical spiking neurons,” Sci. Rep., vol. 6, Jul. 2016, Art. no. 30039.
[8] J. C. Slonczewski, “Conductance and exchange coupling of two ferromagnets separated by a tunneling barrier,” Phys. Rev. B, Condens. Matter, vol. 39, no. 10, p. 6995, 1989.
[9] A. Sengupta, M. Parsa, B. Han, and K. Roy, “Probabilistic deep spiking neural systems enabled by magnetic tunnel junction,” IEEE Trans. Electron Devices, vol. 63, no. 7, pp. 2963–2970, Jul. 2016.
[10] C. M. Liyanagedera, A. Sengupta, A. Jaiswal, and K. Roy, “Stochastic spiking neural networks enabled by magnetic tunnel junctions: From nontelegraphic to telegraphic switching regimes,” Phys. Rev. Appl., vol. 8, no. 6, 2017, Art. no. 64017.
[11] A. Sengupta, G. Srinivasan, D. Roy, and K. Roy, “Stochastic inference and learning enabled by magnetic tunnel junctions,” in Proc. IEEE Int. Electron Devices Meeting (IEDM), 2009, pp. 1–4.
[12] G. Srinivasan, A. Sengupta, and K. Roy, “Magnetic tunnel junction based long-term short-term stochastic synapse for a spiking neural network with on-chip STDP learning,” Sci. Rep., vol. 6, Jul. 2016, Art. no. 29545.
[13] K. Roy, A. Sengupta, and Y. Shim, “Perspective: Stochastic magnetic devices for cognitive computing,” J. Appl. Phys., vol. 123, no. 21, 2018, Art. no. 210901.
[14] B. Behin-Aein, V. Diep, and S. Datta, “A building block for hardware belief networks,” Sci. Rep., vol. 6, Jul. 2016, Art. no. 29893.
[15] D. Vodenicarevic et al., “Low-energy truly random number generation with superparamagnetic tunnel junctions for unconventional computing,” Phys. Rev. Appl., vol. 8, no. 5, 2017, Art. no. 54045.
[16] A. Sengupta, C. M. Liyanagedera, B. Jung, and K. Roy, “Magnetic tunnel junction as an on-chip temperature sensor,” Sci. Rep., vol. 7, no. 1, pp. 1–8, 2017.
[17] Y. Shim, A. Sengupta, and K. Roy, “Biased random walk using stochastic switching of nanomagnets: Application to sat solver,” IEEE Trans. Electron Devices, vol. 65, no. 4, pp. 1617–1624, Apr. 2018.
[18] K. Y. Camsari, B. M. Sutton, and S. Datta, “P-bits for probabilistic spin logic,” Appl. Phys. Rev., vol. 6, no. 1, 2019, Art. no. 11035.
[19] B. R. Zink, Y. Lv, and J.-P. Wang, “Independent control of antiparallel- and parallel-state thermal stability factors in magnetic tunnel junctions for telegraphic signals with two degrees of tunability,” IEEE Trans. Electron Devices, vol. 66, no. 12, pp. 5353–5359, Dec. 2019.
[20] B. R. Zink, Y. Lv, and J.-P. Wang, “Telegraphic switching signals by magnet tunnel junctions for neural spiking signals with high information capacity,” J. Appl. Phys., vol. 124, no. 15, 2018, Art. no. 152121.
[21] K. Yang and A. Sengupta, “Stochastic magnetoelectric neuron for temporal information encoding,” Appl. Phys. Lett., vol. 116, no. 4, 2020, Art. no. 43701.
[22] Y. Cheng, B. Peng, Z. Hu, Z. Zhou, and M. Liu, “Recent development and status of magnetoelectric materials and devices,” Phys. Lett. A, vol. 382, no. 41, pp. 3018–3025, 2018.
[23] D. E. Nikonov and I. A. Young, “Benchmarking spintronic logic devices based on magnetoelectric oxides,” J. Mater. Res., vol. 29, no. 18, pp. 2109–2115, 2014.
[24] I. Chakraborty, A. Agrawal, and K. Roy, “Design of a low-voltage analog-to-digital converter using voltage-controlled stochastic switching of low barrier nanomagnets,” IEEE Magn. Lett., vol. 9, May 2018, Art. no. 3103905.
[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[26] A. F. Vincent, N. Locatelli, J.-O. Klein, W. S. Zhao, S. Galdin-Retailleau, and D. Querlioz, “Analytical macrospin modeling of the stochastic switching time of spin-transfer torque devices,” IEEE Trans. Electron Devices, vol. 62, no. 1, pp. 164–170, Jan. 2015.
[27] A. Sengupta and K. Roy, “Encoding neural and synaptic functionalities in electron spin: A pathway to efficient neuromorphic computing,” Appl. Phys. Rev., vol. 4, no. 4, 2017, Art. no. 41105.
[28] S. Singh, A. Sarma, S. Lu, A. Sengupta, V. Narayanan, and C. R. Das, “Gesture-SNN: Co-optimizing accuracy, latency and energy of snn for neuromorphic vision sensors,” in Proc. IEEE/ACM Int. Symp. Low Power Electron. Des. (ISLPED), 2021, pp. 1–6.
[29] K. Mahapatra, S. Lu, A. Sengupta, and N. R. Chaudhuri, “Power system disturbance classification with online event-driven neuromorphic computing,” IEEE Trans. Smart Grid, vol. 12, no. 3, pp. 2343–2354, May 2021.