Exploring the Opportunity of Implementing Neuromorphic Computing Systems with Spintronic Devices

(Invited Paper)

Bonan Yan*, Fan Chen*, Yaojun Zhang†, Chang Song*, Hai Li*, Yiran Chen*
*Duke University, †University of Pittsburgh
*{bonan.yan, fan.chen, chang.song, hai.li, yiran.chen}@duke.edu, †yaz24@pitt.edu

***Abstract*—Many cognitive algorithms such as neural networks cannot be efficiently executed by von Neumann architectures, the performance of which is constrained by the memory wall between microprocessor and memory hierarchy. Hence, researchers started to investigate new computing paradigms such as neuromorphic computing that can adapt their structure to the topology of the algorithms and accelerate their executions. New computing units have been also invented to support this effort by leveraging emerging nano-devices. In this work, we will discuss the opportunity of implementing neuromorphic computing systems with spintronic devices. We will also provide insights on how spintronic devices fit into different part of neuromorphic computing systems. Approaches to optimize the circuits are also discussed.**

1. Introduction

Machine learning techniques, such as deep neural networks (DNNs), have achieved remarkable success in many research areas and applications [1]–[3]. Some of these techniques also obtained the ability to achieve close to or ever better than human-level perception. Such a success, to a great extent, is enabled by the advances in hardware designs of neuromorphic computing systems (NCS) [4]–[8], [10]. NCS utilize new devices and circuit components to implement the behavior of neural networks with complex structures or multiple nonlinear transformations to exact a high-level abstraction of data.

In many applications, extending the depth of neural networks for better accuracy becomes a popular approach, exacerbating the requirement for computation resources and data storage of hardware platforms. However, some researchers believe that conventional CMOS-based computing paradigms may not be suitable to implement large-scale NCS due to their poor scalability, low energy efficiency and rapidly increased hardware cost. One solution to solve the discrepancy between the fast growth of the neural network model size and the slow performance improvement of conventional CMOS-based paradigms is to revolutionize the implementation of NCS with emerging technologies.

Figure 1 shows a conceptual diagram of nonvolatile memory (NVM) based NCS. Based on different functions, it can be logical divided into three parts: synapses to compute matrix multiplications, neuron circuits to process the calculation results and then transmit the results to proceeding layers and peripheral circuits to fulfill miscellaneous functions including decoding, timing control and post-neuron

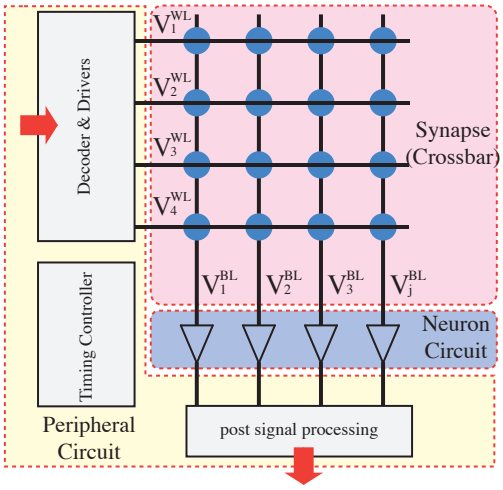

Figure 1. System diagram for NVM based NCS.

signal processing. In this paper, we target circuit-level and architecture-level NCS designs using emerging nonvolatile memory (eNVM) technologies to achieve high computing parallelism and integration density. We choose spintronic devices, i.e., *Spin-Transfer Torque RAM* (STT-RAM) and *Domain Wall Memory* (DWM, also know as *Racetrack Memory*), to implement NCS because of their proven CMOS compatibility and manufacturing matureness in commercial semiconductor foundries (e.g., TSMC) and stable resistance states. Moreover, spintronic NVM device possesses the potential to realize not only high density and high performance memory storage but also in-memory computing systems.

In this work, we will discuss the opportunity of implementing neuromorphic computing systems with spintronic devices. Prior research on implementing NCS on STT-RAM [11] and Domain Wall Memory [8], [10] are also reviewed. The corresponding computation form and the structure of the computing paradigms are discussed in details.

2. Fundamental of Spintronic Devices

2.1. Basics of domain wall memory

The structure of domain-wall nanowire device is illustrated in Figure 2. As the figure shows, multiple magnetic domains are integrated on a memory track separated by ultra narrow domain walls (the red part). Multiple access ports are uniformly distributed along each track. The binary value of a magnetic domain is represented by its relative magnetization

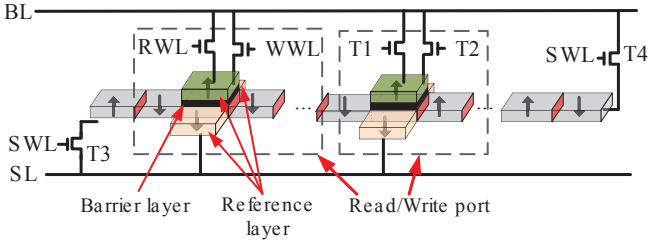

Figure 2. The diagram of domain-wall nanowire device.

directions to the reference layer (the green part) at the access port. A selected device together with a magnetic tunneling junction (MTJ) sensor build an access port. The wordline is split into read worline (RWL), write wordline (WWL) and shift wordline (SWL), which support the read, write, and shift operations.

By detecting the resistance of MTJ, the stored bit can be read out. Moreover, write operation is realized by a highly-efficient shift-based write method introduced in [12]. As indicated in Figure 2, two fixed domains (the pink part) with different magnetizations sandwich the memory track. During write operations, the write transistor T2 is turned on; different voltages pairs are then applied to the bitline (BL) and the source line (SL) to shift one of the two fixed domains into the memory track. Note that a shift operation is required to align the memory cell to an access port during read/write operations.

In particular, a recent work [13] investigated the latest *Skyrmions Racetrack Memory* (SKM) for cache designs. *Spin Hall Effect* (SHE) based motion in SKM further improves the energy efficiency in existing DWM, demonstrating great potentials in ultra-low power neuromorphic computing.

2.2. Giant spin hall effect

Giant Spin Hall Effect (GSHE) MTJ is a three-terminal device, as depicted in Figure 3 [14]. A GSHE MTJ is comprised of a stack of various layers, i.e., a free layer, an dielectric oxide layer, and a reference layer, a GSHE electrode strip, and a antiferromagnetic layer. Similar to conventional MTJs, the relative angle of the magnetization directions in the reference layer and the free layer determines the device resistance. Parallel (low resistance) and anti-parallel (high resistance) orientations represent the binary data '0' and '1', respectively. The antiferromagnetic layer on the top is used to ensure that the magnetic orientation of the reference layer is always fixed. Moreover, the electrode strip is coupled with the free layer, which filters the charge current flowing in it into spin-polarized current. It will exert a spin torque on the free layer and change the free layer magnetization between parallel and anti-parallel states.

Such a scheme isolates the programming and sensing to avoid read disturbance. When the applied programming current on the strip (from terminal A to B as shown in Figure 3) is larger than a switching threshold, the device will be written to '0', and it will switch to '1' if a current larger than the threshold is applied on the opposite direction. If the current from either direction is lower than the threshold

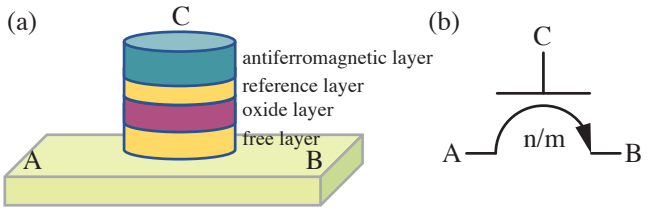

Figure 3. (a) Diagram of GSHE MTJ; (b) circuit symbol of GSHE MTJ.

current, the device will not be written and remain its current data state.

Figure 3 (b) shows the symbol for the GSHE MTJ. The 'n' and 'm' denote the selections of input nodes and switching threshold, respectively [14]. The threshold can be adjusted through device film engineering.

3. Spintronic Device as Synapse

Emerging nonvolatile memories, especially the ones with cross-point or crossbar array structure, can be used as the analogy of matrix multiplication to implement NCS. The resistance values of emerging nonvolatile memory devices represent the weights in the neural networks. At algorithm-level, the trained weights generally follow lognormal distribution, shown as the green curve in Figure 4. However, for the nonvolatile memories, the programming process is limited by the resolution that CMOS circuitry can offer. The limited programming resolution requires a quantization process that maps each analog weight to one of the values that are represented by the discrete resistance states of NVMs. Blue curve in Figure 4 depicts an example of direct quantization after training, where the well-trained weights are quantized to 6 (peaks) levels.

At device-level, binary MTJ is the most common due to the ease of fabrication to control the magnetic anisotropy. Multi-bit storage cells can be realized by either producing more stable states [15] or the stack of MTJs [16]. However, these approaches still cannot satisfy the algorithm-level requirement of 8-bit weight for general applications [17]. Except for device engineering optimization, algorithm-level technique is also helpful and efficient to relax this strong requirement of the resistance state resolution.

By changing the regularization in training, the weight distribution can be tuned to fit the resistance values provided by the MTJs [18]. In the study of machine learning, regularization is referred to as the process of introducing additional information to prevent overfitting [18]. Without

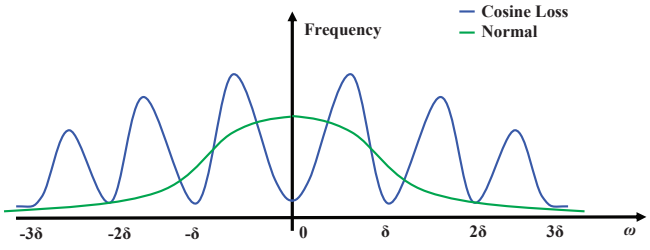

Figure 4. Weight distribution before (green) and after (blue) quantization [18].

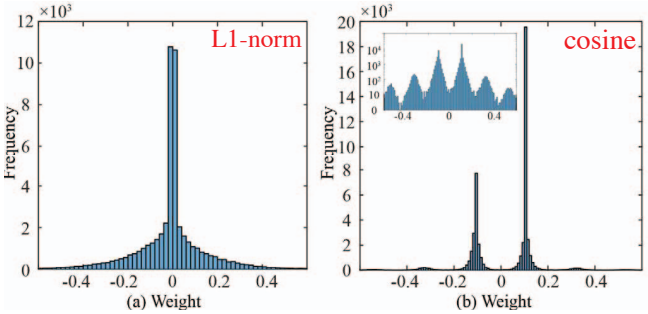

Figure 5. Weight distribution before (green) and after (blue) quantization using the periodic regularization method in training [18].

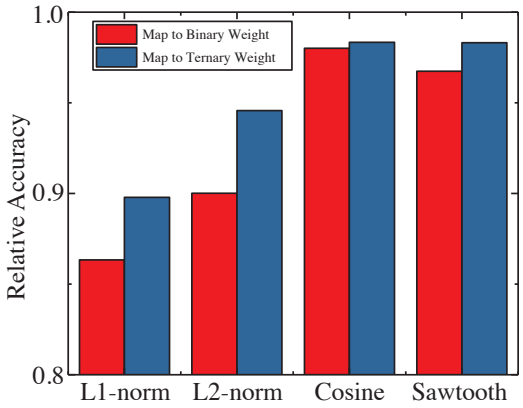

Figure 6. Relative accuracy after quantization using different regularization function in training [18]. The neural network is for the recognition for LeNet-5 database.

loss of generality, the regularization term can be expressed by a complexity penalty added to the cost function of the learning process. Traditional regularization function is realized by a monotonically (in the most input region) rising, which is not efficient to tune the trained weight. If a periodic function, such as cosine and sawtooth, is used as the regularization function, the trained weights will shift to the target digitalized levels in training, leading to a best balance of computation accuracy and weight digitalization. Figure 5 shows the weight distribution with a revised regularization in training. Compared with Figure 4, the regularization method is capable of tuning the trained weight distribution according to the available resistance states. Figure 6 shows the relative accuracy using this method after quantization. The accuracies are normalized to the trained baseline without using quantization. The use of cosine or sawtooth regularization saves the quantization loss compared with conventional L1 or L2 norm regularization. In this way, algorithm-level weights are tailored for accuracy enhancement for NVM based NCS implementations.

4. Domain Wall Memory As Neuron Circuit

This section reviews some previous works about implementing neuromorphic computing systems with domain wall memory.

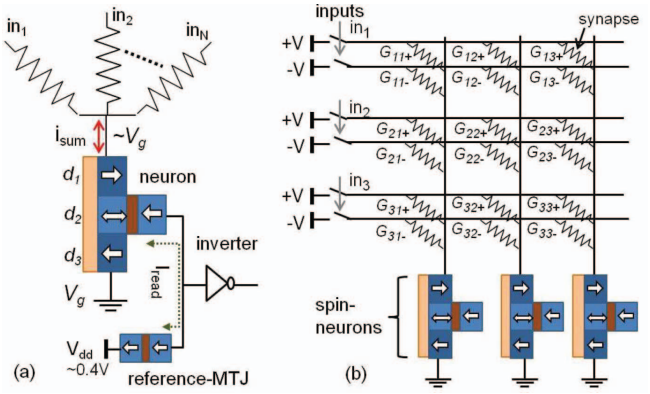

Figure 7. (a) Spin neurons connected with synapses, (b) Nerual network circuit using spin neurons. [8]

4.1. Spin neurons

Spin torque neuron designs based on DWM are proposed in [8] for low power neural network computation. As shown in Figure 7(a), a spin neuron has three terminals which can be used for current-mode thresholding. Figure 7(b) illustrates a 3×3 neural network circuit using spin neurons. Experimental results demonstrate that spin neurons can achieve more than two orders of magnitude lower energy and beyond three orders of magnitude reduction in energy-delay product compared to CMOS-based analog circuit-model of neurons.

4.2. Domain wall memory based convolutional neural networks (CNNs)

[10] proposed to implement CNN convolutional layers, which are the core building blocks of CNNs, by adopting the partial dot product implementation using DWM-based cell strings. Multiple partial dot products are then merged in DWM-based sub-array. In the proposed DWM-based architecture, each cell array is composed of a DWM-based sub-array and an Analog to Digital Converter (ADC) sub-array. Due to page limit, we skip the details but refer the readers who are interested in this architecture to [10].

Simulation results using 65 nm CMOS process show that the proposed design archives 34% energy savings with some extendible for high resolution classifications, compared to the conventional implementation using memristor-based crossbar.

5. GSHE Logic As Peripheral Circuit

Although the adoption of nonvolatile memory dramatically improves computation and storage efficiency in these designs, the peripheral circuits, including the decoders and the controllers etc., greatly hinder the performance metrics of these NCS, such as area, power, recognition accuracy, and training speed [19]. For example, the voltage-based sensing scheme in [20] requires *analog-digital and digital-analog converters* (ADC/DAC), which result in large signal distortion and power consumption. The spiking based designs commonly adopt *integrate-and-fire circuit* (IFC) to generate

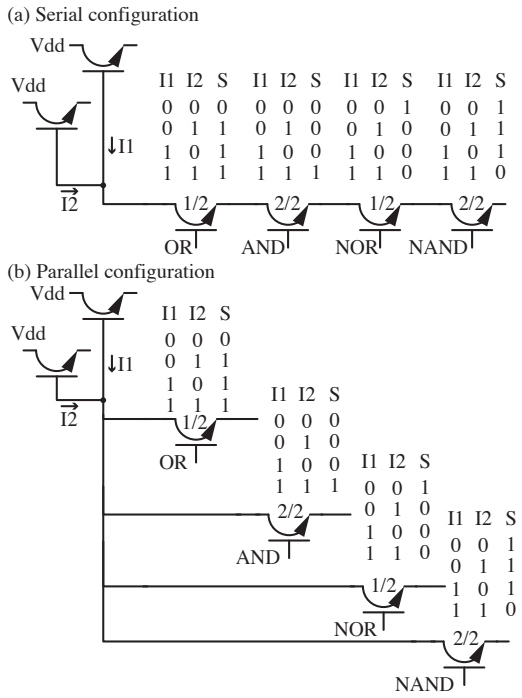

Figure 8. Fundamental logic gates are implemented by GSHE MTJs.

output spikes. Although the power efficiency dramatically improves, an IFC component still requires a number of transistors and a capacitor for charge accumulation [19].

Further reduction of power and area requires the improvements of peripheral circuits. Spintronic devices based logic serves as a promising design alternative to replace the CMOS based implementations.

For the implementations of complex types of neural networks, e.g., Brain-State-in-a-Box (BSB), high density is critical to efficient on-chip implementations. [14] proposed to use GSHE devices to implement digital logic, which is a design alternative to the CMOS-based implementation of peripheral circuits that usually need a large number of gates. The basic structures of OR, AND, NOR and NAND gates are shown in Figure 8. The logic is implemented by threshold logic design.

Due to the small size of GSHE devices, digital logic can be implemented on a very small area. For a two-input AND gate, GSHE logic uses only one GSHE MTJ, which can be scaled to the range from 10nm to 30nm [21]. However, in comparison, conventional CMOS logic consumes several hundreds of feature size square or even larger to meet the requirements of performance or drive capability [22].

6. Conclusion

In this paper, we review several research papers on implementations of NCS using spintronic devices. The development of spintronics leads to various devices that demonstrate appropriate features to construct neural networks. We give our insights on how STT, DWM and GSHE devices fit into different parts of NCS according to their characteristics. By exploring the promising approaches, we provide different approaches to optimize the circuits to obtain satisfactory performance.

Acknowledgment

This work was supported in part by NSF-1725456, 1744111 and DE-SC0018064. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of grant agencies or their contractors.

References

[1] P. Sermanet and et al., "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks," *ArXiv e-prints*, 2013.
[2] C. Farabet and et al., "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.
[3] A. Krizhevsky and et al., "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.
[4] S. K. Esser and et al., "Cognitive computing systems: Algorithms and applications for networks of neurosynaptic cores," in *IJCNN*. IEEE, 2013, pp. 1–10.
[5] M. Sharad and et al., "Ultra-low power neuromorphic computing with spin-torque devices," in *2013 Third Berkeley Symposium on Energy Efficient Electronic Systems (E3S)*, 2013, pp. 1–2.
[6] L. Song and et al., "Pipelayer: A pipelined reram-based accelerator for deep learning," in *HPCA*, 2017, pp. 541–552.
[7] F. Chen and et al., "Regan: A pipelined reram-based accelerator for generative adversarial networks," in *ASP-DAC*, 2018.
[8] M. Sharad and et al., "Spin-neurons: A possible path to energy-efficient neuromorphic computers," *Journal of Applied Physics*, vol. 114, no. 23, 2013.
[9] H. Yu and et al., "Energy efficient in-memory machine learning for data intensive image-processing by non-volatile domain-wall memory," in *ASP-DAC*, 2014, pp. 191–196.
[10] J. Chung and et al., "Domain wall memory based convolutional neural networks for bit-width extendability and energy-efficiency," in *ISLPED*, 2016, pp. 332–337.
[11] M. Sharad and et al., "Spin-based neuron model with domain-wall magnets as synapse," *IEEE Transactions on Nanotechnology*, vol. 11, no. 4, pp. 843–853, 2012.
[12] R. Venkatesan and et al., "Dwm-tapestri - an energy efficient all-spin cache using domain wall shift based writes," in *DATE*, 2013, pp. 1825–1830.
[13] F. Chen and et al., "Process variation aware data management for magnetic skyrmions racetrack memory," in *ASP-DAC*, 2018.
[14] Y. Zhang and et al., "Giant spin hall effect (gshe) logic design for low power application," in *DATE*. IEEE, 2015, pp. 1000–1005.
[15] Q. Stainer and et al., "Self-referenced multi-bit thermally assisted magnetic random access memories," *Applied Physics Letters*, vol. 105, no. 3, p. 032405, 2014.
[16] Y. Zhang and et al., "Multi-level cell stt-ram: Is it realistic or just a dream?" in *ICCAD*. IEEE, 2012, pp. 526–532.
[17] T. Pfeil and et al., "Is a 4-bit synaptic weight resolution enough?–constraints on enabling spike-timing dependent plasticity in neuromorphic hardware," *Frontiers in neuroscience*, vol. 6, 2012.
[18] C. Song and et al., "A quantization-aware regularized learning method in multilevel memristor-based neuromorphic computing system," in *NVMSA*. IEEE, 2017, pp. 1–6.
[19] C. Liu and et al., "A spiking neuromorphic design with resistive crossbar," in *DAC*. IEEE, 2015, pp. 1–6.
[20] M. Hu and et al., "Hardware realization of bsb recall function using memristor crossbar arrays," in *DAC*. ACM, 2012, pp. 498–503.
[21] S. Manipatruni and et al., "Energy-delay performance of giant spin hall effect switching for dense magnetic memory," *Applied Physics Express*, vol. 7, no. 10, p. 103001, 2014.
[22] N. H. Weste and D. Harris, *CMOS VLSI design: a circuits and systems perspective*. Pearson Education India, 2015.