IEEE TRANSACTIONS ON
EMERGING TOPICS
IN COMPUTING

Received 22 April 2022; revised 12 August 2022; accepted 25 September 2022.
Date of publication 11 October 2022; date of current version 7 June 2023.

Digital Object Identifier 10.1109/TETC.2022.3212341

Scalable Reasoning and Sensing Using Processing-In-Memory With Hybrid Spin/CMOS-Based Analog/Digital Blocks

MOUSAM HOSSAIN, (Student Member, IEEE), ADRIAN TATULIAN, (Student Member, IEEE), SHADI SHEIKHFAAL, (Student Member, IEEE), HARSHAVARDHANA R. THUMMALA, (Student Member, IEEE), AND RONALD F. DEMARA, (Senior Member, IEEE)

The authors are with the University of Central Florida, Orlando, FL 32814 USA

CORRESPONDING AUTHOR: MOUSAM HOSSAIN (EMAIL: mousam.hossain@knights.ucf.edu).

This work was supported by the National Science Foundation (NSF) through CCSS 1810256.

ABSTRACT In this article, we leverage in-memory computation for data-intensive applications to surmount the bandwidth restrictions inherent in the Von-Neumann computing paradigm, while addressing transistor technology scaling challenges facing Moore's Law. We introduce *the Spintronically Configurable Analog Processing in-memory Environment (SCAPE)* which incorporates top-down architectural approaches along with bottom-up intrinsic device switching behaviors of spin-based post-CMOS devices. SCAPE embeds analog arithmetic capabilities providing a selectable thresholding functionality to realize generalized neuron activation functions that are integrated within the 2-dimensional memory array. Within each module, circuit-switched connections allow in-field configuration of the partly reconfigurable neuron activation function to suit the target application, and the intrinsic computation is performed using spin-based devices. This hybrid-technology design advances in-memory computation beyond previous approaches by integrating analog arithmetic, runtime reconfigurability, and non-volatile devices within a selectable 2-dimensional topology. Simulation results of error rates, power consumption, power-error-product metric, are examined for real-world applications including edge-of-network based Compressive Sensing and Machine Learning use cases, along with process variation analysis. Results show up to 7% improvement in error rate using proposed implementation of enhanced activation function versus baseline conventional sigmoidal activation, whereas realization of AMP signal processing algorithm shows ∼95% reduction in energy consumption at comparable accuracy.

INDEX TERMS Beyond-cmos devices, neuron activation function, non-von neumann architectures, spin-hall effect mtj

I. INTRODUCTION

Recent advances in technology continue significant shifts towards data intensive applications such as image processing utilizing machine learning techniques [1]–[3]. Simultaneously, these are sought to operate under energy constraints imposed by edge-of-network based embedded components. In particular, Artificial Neural Network (ANN) architectures and edge-of-network applications make significant use of *Vector-Matrix Multiplication (VMM)* operations which impose significant memory transfer demands [4]. VMM operations are pervasive within ANN processing, as well as Compressive Sensing tasks targeted for emerging real-world applications at the edge of the computing network.

Despite VMM operations becoming widely rehosted from a general-purpose computing paradigm to GPUs, TPUs, and FPGAs, they continue to face challenges including high energy consumption and memory-wall obstacles. Due to the high bandwidth demands of data transfer inherent in such VMM-intensive applications, the *Von-Neumann architectural model* of data transfer between discrete memory and processing units, is being reconsidered as it suffers from large latency and energy costs. In order to overcome the memory bottleneck,

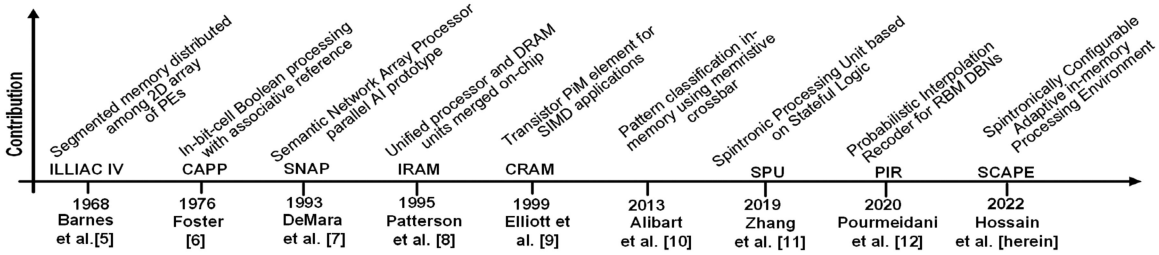


FIGURE 1. Timeline of foundational works towards hybrid spin/CMOS-based application-specific processing-in-memory.

devices and architectures that go beyond Von Neumann architectural principles are increasingly sought to offer processing capability closer to where the data resides. This has given rise to the study of memory-centric approaches to attain improved throughput and energy efficiency, both with and without modification to the underlying storage/switching devices utilized in the design of the Processing-in-Memory (PiM) component itself. Most recently with the fabrication, demonstration, and preliminary commercialization of post-CMOS devices such as Spin Transfer Torque Magnetic Tunnel Junctions (STT-MTJs), and Spin Hall Effect Magnetic Tunnel Junctions (SHE-MTJs), such devices are investigated here towards advancing PIM paradigm to enable emerging opportunities for future edge-of-network computing platforms.

A. PROCESSING-IN-MEMORY ARCHITECTURAL MILESTONES

Architectural advancements in pursuit of PiM computational paradigms have targeted various gainful attributes for special-purpose computing over the last five decades. Although a comprehensive summary would be too extensive, Figure 1 delineates the progression of the noteworthy research milestones that have laid the foundation for the research herein. Specifically, application-specific PiM approaches have continued to evolve from distributed memory modules in conventional array processors up through hybrid spin/CMOS-based memory/processing cells capable of intrinsic execution of selected computations. Starting with segmented memory distributed physically across an ensemble of Processing Elements (PEs), Slotnick et al. fielded the *Illinois Automatic Computer (ILLIAC)* by researching the concept of distributed memory closely-coupled with localized parallel processing operations via the association of segmented memory among identical PEs [5]. Next, by drilling down to the bit-cell level while focusing on the referencing capability of data when resident inside the memory component, Foster advocated the benefits and capabilities of a *Content Addressable Parallel Processor (CAPP)* [6]. The CAPP provided an umbrella term for hardware implementation of Boolean logic gates elements replicated within each SRAM bit cell, which tagged contents as responders for further processing without involving off-chip processor/memory transactions. Leveraging the concept of content addressability for PiM, DeMara developed the *Semantic Network Array Processor parallel AI prototype* which used in-place computation initiated with SIMD broadcast mode [7]. The responder PEs storing the semantic network then launched an MIMD model of spreading-activation to conduct reasoning tasks without bus transactions using a multi-ported memory approach. Later on, when microprocessors became ubiquitous in the computing landscape, including the MIPS chip he designed and helped to commercialized, Patterson advocated the case for *Intelligent RAM (IRAM)* to unify logic elements within a DRAM memory module, thereby bridging the memory-wall between the processor and memory [8]. Next, while furthering the IRAM-style PIM paradigm, Elliot et al. researched tightly-coupled integrations of more complex logic networks to capture data parallelism via SIMD architectural implementations of PiM. Elliot evaluated transistor count and area costs versus throughput benefits of embedding PiM of various granularities up through rudimentary ALUs consisting of a few hundred transistors [9].

During the last decade, the aforementioned works promoted considerable research interest to extend the PiM paradigm beyond the use of transistors alone. These utilize emerging logic devices, such as memristors and spintronic devices as alternatives to CMOS-based memory designs. For instance, Strukov et al. in [10] showed emerging memristive devices could be used in a 2D-crossbar layout to conduct pattern recognition tasks leveraging the intrinsic switching behaviors of titanium-dioxide-based memristive devices within a *Computational RAM (CRAM)* component. Zhang et al. in [11] present a PiM platform called Spintronic Processing Unit (SPU), configurable at the individual cell level for performing different logic functions using memory-like read and write operations. Different logic functions are computed by altering the final state of the memory cell based on different input operands. The final state of an STT-MRAM bit-cell is given by $B_{i+1} = AC + A'B_i$; where, A and C are the inputs to the WL and BL, respectively, and B_i and B_{i+1} are the initial data and final result stored in the MTJ device, respectively. Different Boolean functions are achieved by altering the input variables A, C, and B_i. This work also shows how the ISA can be modified with additional instructional support such as MOV and LOG, for moving data to the target bit-cell and carrying out the logic operation based on value of input operands, respectively. Although intrinsic switching functionalities of memristors in this context were shown to offer a viable new approach to PiM, the limited endurance of their write cycles and substantial drift of ON/

OFF resistances presented new challenges. Thus, Pourmeidani et al [12] advanced a crossbar of non-volatile tunable stochastic elements based on MTJs by developing *Probabilistic Interpolation Recoder (PIR)* for Deep Belief Networks (DBNs). The MTJ devices were used to realize near-zero energy barrier switching supporting an unlimited endurance approach to PiM, whereas PIR provided a stochastic based energy and area efficient alternative to conventional interpolation technique of using resistor-capacitance (RC) tanks and analog-to-digital (ADC) convertors. The use of MTJ-based Non-Volatile Memories (NVMs) like commercialized Magnetic Random-Access Memories (MRAMs) allows feasibility for performing arithmetic and logic operations inside memory word lines. This memory word line approach to PiM led to energy-efficient hardware implementation of a *Restricted Boltzmann Machine (RBM) based Deep Belief Network (DBN)* using a conventional sigmoidal activation function. Furthermore, it was found that MTJs can be employed to realize area-efficient and wire-count efficient realization of neurons and synapses, elevating them an emerging device technology useful for accelerating neural networks [13], [14]. Their properties of near-zero standby power, compatibility with CMOS Back End of Line (BEOL) fabrication process offering high integration density enables the implementation of efficient hybrid MRAM/CMOS circuits to combine the benefits of both technologies.

Taking inspiration from various technical attributes of these milestones in PIM approaches spanning the last five decades, herein we consider new roles and approaches to PiM for CS and ML applications. Specifically, we further the efforts in Edge-of-Network PiM with hardware implementation of a Generalized Activation Function in a *Spintronically Configurable Analog Processing-in-Memory Environment (SCAPE)* architecture for selected applications.

B. SENSING AND REASONING OPERATIONS AMENABLE TO PROCESSING-IN-MEMORY

Advancing beyond the foundational works on PiM, the last several years have witnessed interest in pursuing beyond Von Neumann approaches for efficient processing of data in edge-of-network applications such as compressive sensing and automated reasoning. Research has spanned multiple layers of the system stack, ranging from execution model and architectural topology down to algorithmic formulation, as well as the data representation and fundamental signal encoding methods. At the signal encoding stage, emerging spintronic devices enable new tradeoffs beyond the use of digital computation exclusively. In addition to providing computation ability to storage bit-cells in the memory, spintronic devices, due to their vertical-integration capability on MOS transistors, also offer potential area benefits at the cost of incurring additional fabrication complexity. A single bit-cell size comparison of different memory technologies found in [15] shows that STT-MRAM technology has lower cell size than SRAM, but may be comparable to cell size of DRAM technologies. On the other hand, benefits of analog-based computations include reduced wire counts and device counts when compared to digital implementation of non-linear operations such as multiplication and exponentiation, spanning computer vision, signal processing, and machine learning applications. For instance, a traditional digital implementation of multiplication and exponentiation functions can incur significant area and delay overheads in the digital domain, requiring 12 or more clock cycles to execute and hundreds of Boolean logic gates [13]. Analog computation can be especially compatible in edge-of-network application domain owing to the tolerance for approximate computation. Analog circuits trade off computational accuracy for reduction in overheads such as power and area; this is an attractive tradeoff for error-tolerant applications where power and area are constrained, e.g., Internet of Things (IoT) devices. The benefits offered by analog computation are amplified when used with vector-valued data, since the output data can be transferred to a memristive crossbar array for further processing without the need for digital-to-analog conversion. Multiplication and exponentiation operations are critical for a variety of applications, including computer vision, signal processing, and machine learning. Such applications rely extensively upon VMM, wherein its fundamental operation of multiplication requires execution that is efficient and co-located near the data being operated upon. Square and square root, for example, are commonly used for normalizing vectors in signal processing applications, and square root may serve as an activation function for neural networks [13]. One example of a representative use case entailing VMM is Compressive Sensing (CS) involving compression and transmission of a spectrally-sparse signal, and then reconstruction of the signal at the receiving end. Machine learning via neural networks is another example. Herein we propose a device to architecture level compound PiM implementation based on hybrid spin/CMOS, analog as well as digital computational blocks, re-distributed within the memory fabric, inter-communicating via simple control logic modifications to the peripheral circuitry. The major contributions of the paper include:

1) a novel crossbar topology for PiM which provides in-field configurability of Hybrid Spin/CMOS-based Analog/Digital Blocks. Various synapse and neuron designs are evaluated including use of SHE-MTJs for memristive-based computation and activation function calculation.
2) a generalized activation function is developed to mitigate the gradient decay problem while increasing recognition rate. Analog computation of the generalized activation function demonstrates acceptable accuracy, reduced area, and decreased energy consumption, as evaluated on MNIST dataset.
3) the concept of Power Error Product is introduced as a transportable performance metric and is evaluated for various activation functions.
4) quantification of Process Variation (PV) effects when using SHE-MTJ devices. Approach and results for PV

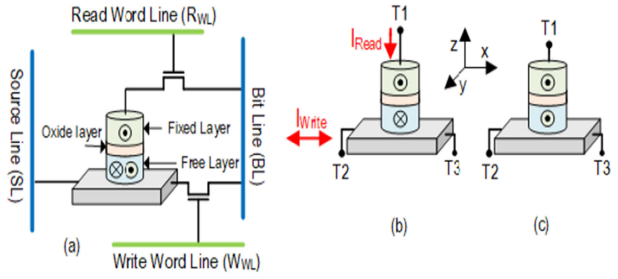


FIGURE 2. (a) SHE-MTJ, (b) anti-parallel (AP) and (c) parallel (P) configurations.

versus neuron activation function deviation are provided using Monte-Carlo method. Standard deviations of 5% for MTJ Parameters such as length, width, thickness are considered.

The manuscript is organized as follows: Section II provides a background on key concepts of spintronics, emerging devices used for memristive PiM and introduces the proposed SCAPE approach. Section III presents hybrid spin/CMOS based synapse design used in deep learning networks, Section IV describes the proposed SCAPE topology with spin-based analog/digital *GAAF* module. The application of the proposed SCAPE components in compressive sensing techniques is demonstrated in Section V. In Section VI, we focus on qualitative and quantitative results with comparison of our approach with similar works in literature in machine learning and compressive sensing applications, along with the effects of process variation on the output accuracy of the GAAF module. Finally, Section VII concludes this paper with a discussion of future challenges.

II. EMERGING DEVICES FOR CROSSBAR-BASED PIM

A. SPINTRONIC MAGNETIC TUNNEL JUNCTIONS

Integrating memory devices into a PiM array should address various important metrics of both storage and computation. In this manuscript, we focus on the use of spintronic devices for PiM, as opposed to other alternatives such as titanium dioxide based memristors, due to their virtually unlimited write endurance documented as 10^{16} write cycles. Magnetic Tunnel Junctions (MTJs) are a class of spin-based emerging logic device which have been recently researched due to numerous advantages, including non-volatility, near-zero standby power dissipation, high endurance [16] and vertical integration capabilities resulting in high density [17], thereby maximizing area efficiency and simultaneously minimizing data transfer overheads [18]. As the building block of MRAMs, MTJs have been proposed as a non-volatile alternative to SRAM in cache memory. Further applications benefiting from a hybrid CMOS/MRAM approach include full adders and analog-to-digital converters. An emerging research thrust is to consider the use of various MTJs in both storage and computation roles with a PiM array [19].

An MTJ consists of two ferromagnetic layers called pinned layer and free layer separated by a thin oxide layer, such as MgO. There are two stable states for the magnetization orientations of the two ferromagnetic layers, parallel (P) and antiparallel (AP). Thus, the MTJ can exhibit two different resistance states due to the Tunneling MagnetoResistance (TMR) effect quantified by the resistivity of the low resistance state (R_P) and a high resistance state (R_{AP}). Specifically, the device resistance is given by $R_P = R_{MTJ}$ and $R_{AP} = R_{MTJ}\ (1 + TMR)$ whereby

$$R_{MTJ} = \frac{t_{ox}}{FactorXArea\sqrt{\varphi}}\ \exp(1.025 t_{ox}\sqrt{\varphi}) \tag{1}$$

$$TMR\ = \frac{TMR_0}{1 + \left(\frac{V_b}{V_h}\right)^2} \tag{2}$$

$$E_B = \frac{1}{2}\ H_K M_S(\pi(d/2)^2 t_f) \tag{3}$$

in which t_{ox} is the oxide layer thickness, *Factor* a material-dependent parameter which depends on the resistance-area product of the device, *Area* the surface area of the device, φ the oxide layer energy barrier height, V_b bias voltage, and V_h the bias voltage at which TMR drops to half of its initial value. MTJs have been fabricated at varying resistance levels ranging from the kilo-Ohm to mega-Ohm range [20]. E_B, is the energy barrier of the MTJ, required to switch from P (AP) to AP (P) states, H_K is the magnetic anisotropy field, M_S saturation magnetization, where d and t_f are the diameter and thickness of the MTJ's free layer which may be tuned based on fabrication dimensions.

Within this paper we focus on spintronic devices using SHE-MTJ shown in Figure 2a. SHE-MTJ is a three-terminal device, with isolated write and read paths with lower switching energy compared with Spin Torque Transfer Magnetic Tunnel Junctions (STT-MTJs). It consists of heavy metal (HM) nanowire beneath an MTJ with two ferromagnetic layers, called the pinned and free layers, separated by a thin oxide barrier. In order to write into the MTJ, the spin Hall effect is leveraged where an unpolarized current through the heavy metal layer along +/- *x* axis results in a change in magnetization along +/-*y* axis and generation of spin-polarized current along +/- *z* axis direction perpendicular to that of the unpolarized current. The spin current so produced transfers its angular momentum to the free layer resulting in switching behavior as shown in Figure 2b.

B. SPINTRONICALLY CONFIGURABLE ADAPTIVE IN-MEMORY PROCESSING ENVIRONMENT (SCAPE) ARCHITECTURE

Recently SHE-MTJs have been explored as means to realize in-memory computing architectures. Herein, we develop the *Spintronically Configurable Adaptive in-memory Processing Environment (SCAPE)* architecture which incorporates top-down architectural approaches along with bottom-up intrinsic device switching behaviors of SHE-MTJs. Key technical objectives of SCAPE are to provide explicit hardware support collocated with large amounts of data that the edge of network devices must encounter, to process and send only

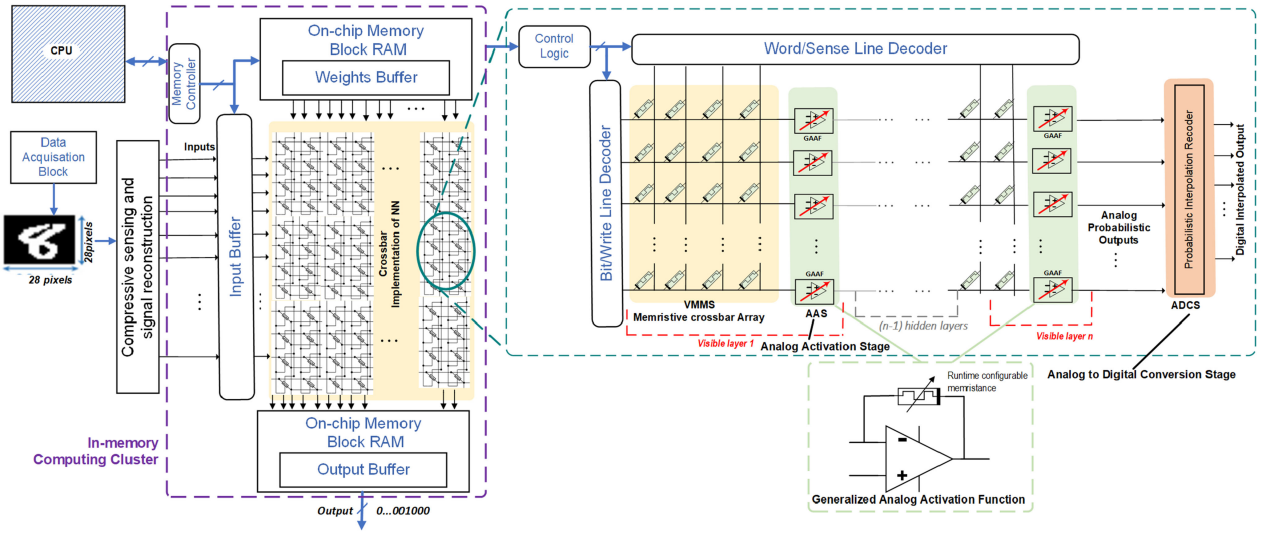


FIGURE 3. Proposed spintronically configurable adaptive in-memory processing environment (SCAPE) architecture.

higher-level information up the network to the cloud. Such applications have high data requirements, whereas they are typically streaming data as well as large templates of matrices, which can stress the memory bottleneck. Therefore, PiM is desirable. Both applications also manipulate data elements via dot product and rely on a large number of VMM operations at various precisions. In the context of machine learning domain, both the synapse and neuron have mathematical operations to perform. The synapse conducts a multiplication operation, while the neuron must perform activation based on thresholding using some type of activation function such as a sigmoid limiter. As mentioned in the previous section, a memristive crossbar conducts the synapse operation as an analog multiplication using current based representation of the values to be multiplied. For these operations, beyond-CMOS devices can add capability to calculate them as intrinsic behaviors of the switching device itself without having complex and area-consuming floating-point hardware units distributed throughout the memory.

An innovation in this paper has been to provide a PiM element that can perform generalized analog multiplication and a Generalizable Analog Activation Function (GAAF). Figure 3 shows the high-level topology of the proposed SCAPE architecture. The memory component is laid out as a 2D crossbar array implementation to realize memristance at crossbar nodes. The SCAPE topology can embed an ANN within the memory as visible layers at the input/output interface of the memory component, and internal cascaded hidden layers, connected as per the machine learning network specification. Each of these layers can be abstracted into three distinct phases/stages: (1) a Vector Matrix Multiplication Stage (VMMS) depicting the synaptic connections between the multiple nodes in each layer and computing the weighted dot-product of the input signals via the crossbar implementation, (2) an Analog Activation Stage (AAS) consisting of proposed GAAF blocks, composed of hybrid spin-analog components realizing various activations of the neuron in response to inputs, and (3) an Analog to Digital Conversion Stage, consisting of a spin-based Probabilistic Interpolation Recoder (PIR) [12] which converts the analog outputs of the AAS stage to digital at a low energy and area footprint.

For illustration, we show the process flow for an edge-of-the-network system, where an image from a benchmark dataset such as MNIST may be acquired from an input image acquisition block, and then via the on-board sensing and signal reconstruction stored into the input buffer of the memory unit. In the case of compressive sensing dot-product also needs to be performed which can be conducted intrinsically by the SHE-MTJ, as elaborated in Section V. The training weights of the dataset are stored on the on-chip block RAM for efficient and quick access. The input buffer data and weights are then fed into the crossbar implementation of the ANN to produce dot products via analog computation. The weighted sums of inputs then propagate through the hidden layers of the neural network, and the corresponding activation layers comprised of the proposed GAAF blocks. A GAAF block consists of an analog hybrid-spin based three stage op-amp, with runtime configurable resistance providing the user with an in-field selectable range of more expressive activation functions, which can be configured at runtime to achieve high accuracy as per the data set to be inferred, as elaborated in Section IV D. Finally, the outputs of the last visible layer are fed to the PIR [12] to achieve the digital outputs to be interfaced with other embedded digital system for further processing. Within this paper we describe the design and tradeoffs using various approaches to embed these processing steps within the memory element. We also evaluate its performance for real world applications of handwritten digit recognition for the MNIST dataset.

III. HYBRID SPIN-CMOS SYNAPSE DESIGN

One way to realize machine learning at the edge of the network is to apply a Short Term Memory-Long Term memory

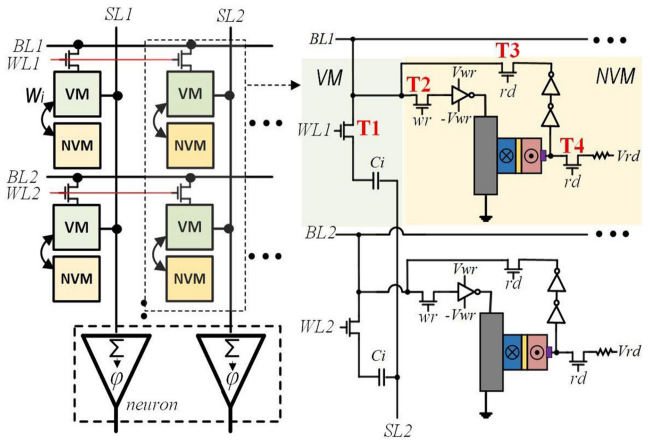


FIGURE 4. **STM-LTM architecture comprising of hybrid SHE-MTJ/CMOS devices [21].**

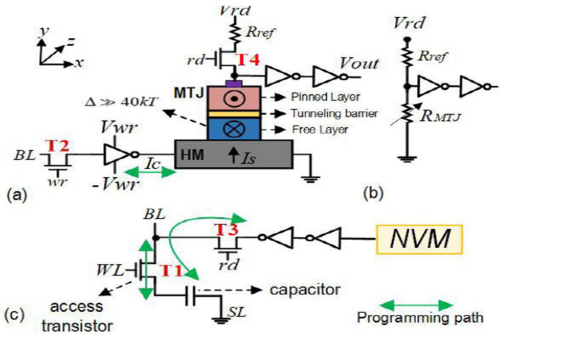


FIGURE 5. **(a) Structure of an SHE-MTJ as NVM. (b) Resistive equivalent read circuit of SHE-MTJ. (c) VM structure programming path [21].**

(STM-LTM) approach. A crossbar-based synapse interconnect can be efficient, as delimited in [21]. There are a variety of hybrid arrangements of device technologies that can exploit alternative mechanisms, such as capacitive synapses used in place of resistive coupling, which feature an ultra-small static power dissipation [22]. In [23], a capacitive neural network has been proposed that utilizes a charge-based capacitor crossbar to perform VMM operation. Such designs realize the weighted summation of inputs through capacitive coupling and voltage division to generate the output in a read-like operation performed by memory devices.

A. BIOLOGICALLY-INSPIRED STM-LTM ARCHITECTURE

A biologically-inspired binary STM-LTM memory architecture, as shown in Figure 4, consists of a 2-D array of memory components leveraging a pair of volatile memory (VM) and NVM as the memory bit cell to realize STM and LTM, respectively. The VM utilizes a capacitor, controlled by an access transistor, in a fashion analogous to a DRAM structure. The NVM is designed with a SHE-MTJ. Each memory bit cell is connected to a bit-line (BL), word line (WL), and source line (SL) managed by the control unit's voltage driver, commensurate with conventional memory array designs. The BL and WL are shared amongst the cells within the same row, and the SL is shared between cells within the same column, as shown in Figure 4, to allow the architecture to operate in three distinct modes of computing, LTM-to-STM data transfer, and STM-to-LTM data transfer [21].

B. MEMORY UNIT DESIGN

1) CAPACITOR AS STM

Recently, several works have explored the potentials of such capacitor-based memories in neural network applications [22], [23]. Training neural networks to high degrees of accuracy requires consecutive, small changes in weights, for which NVMs are not ideal due to limited speed and endurance. Thus, DRAM offers a suitable mechanism for online (in situ) training due to its relatively high speed and symmetrical read/write with infinite endurance, which is a critical aspect for networks that necessitate constant training in an extended period, such as the IoT edge devices [24]. In digital capacitor-based accelerators [25], every memory BL can perform bitwise digital Boolean logic operations, where each capacitor stores a binary synaptic weight, and so, a low-bit-width and parallel computation has been realized, without requirement of ADC/DAC peripherals as in Re-RAM based accelerators [43].

In [21], Shiekhfaal et al. aim to implement a capacitive crossbar enhanced with an NVM in a new fashion based on the STM-LTM features inspired by biology. Each memory bit-cell's capacitor represents a binary synaptic weight ("1" or "0") stored as the "charged" or "discharged" capacitor states. The STM's access transistor [T1 in Figure 5c] is controlled by WL enabling selective write/read operation on the cells located within one row. Storing the network weights in the STM (through a write operation) and strengthening the memory (through STM-to-LTM transfer) are two crucial tasks that need to be carried out. For both operations, the capacitor is initially in the pre-charged state (P.S.), i.e., the BL voltage is preset to (VDD/2) by the voltage driver. To save weight on a capacitor the memory decoder first activates the corresponding WL, and the BL is set to high (VDD) or low voltage (GND). This will provide enough bias voltage to change the capacitor data in a DRAM fashion. The synaptic weight representing STM will be then used to perform the computation or STM-to-LTM transfer.

2) SHE-MTJ AS LTM

The NVM element in the STM-LTM memory architecture is a spintronic device named SHE-MTJ that uses a stable nanomagnet ($E_B >> 40$kT), with two CMOS inverters to amplify the output, as shown in Figure 5a. In order to store the data in the SHE-MTJ, the free-layer magnetization is manipulated by injecting a charge current (I_c) to HM in the +x(/-x)-direction, as shown in Figure 5a. Figure 5b shows an equivalent read circuit of an SHE-MTJ. To read out data from the SHE-MTJ, a read voltage is applied to sense the resistance of the device through realizing a resistive voltage divider. We have considered three access transistors to control the SHE-MTJ with respect to our volatile element, as shown in Figure 4.

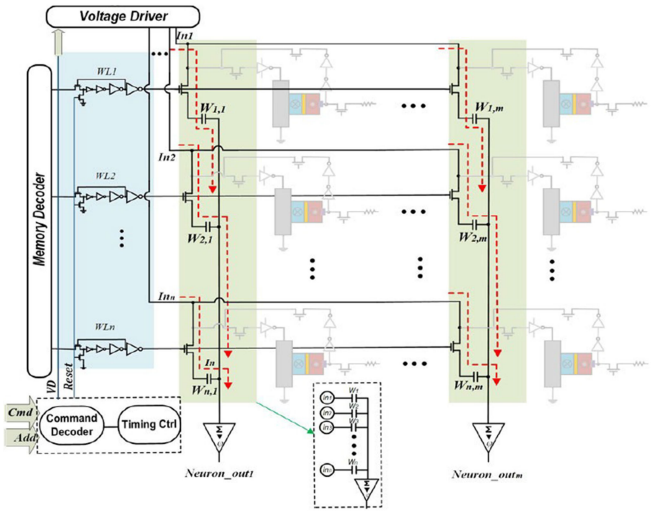


FIGURE 6. Realization of the capacitive network [24] within the proposed LTM-STM memory architecture [21].

The T3 and T4 transistors are devised to activate the read path, and T2 is devised to control NVM and VM data transfer.

C. COMPUTING MODE OPERATION VIA NVM CROSSBAR

In this mode, by activating multiple WLs simultaneously (T1 is ON in Figure 4) and applying input voltages on BLs, VMs can modulate the input and realize the weighted summation of inputs using a capacitive voltage divider circuit and send it to the output neuron via SL, while NVM is deactivated (T2–T4 are OFF in Figure 4). The realization of an n x m capacitive network inspired by [23] and [24] is shown in Figure 6. The memory decoder outputs are enhanced by the inverter chain (blue shaded area) to activate multiple WLs simultaneously. The controller governs the timing of the signal going through the crossbar by controlling the memory address and assigning suitable input voltages through the voltage driver. The input signals are encoded as voltage pulse and simultaneously charge the array in each capacitive node. In order to perform VMM operation, by applying the V_{in} as an input signal to each row, the charges in capacitors will be redistributed and averaged by a reference capacitance, and finally, the output voltage can be written as $V_{out} = ((\sum_{i=1}^{input} C_{i,j}V_{in,i}/C_{ref}))$through voltage division between the cells located in the same column [26]. Table 1 compares the STM-LTM platform in [21] with existing designs in terms of technology, applicability, and potentials of a single synapse unit. The MTJ-based and memristor-based synaptic designs presented in [27], [29] imitates long-term potentiation based on the magnitude, frequency, and duration of input stimulus, with the STM state acting as a transition to LTM state, without any other practical functionality. Also, no circuit implementation to support the utilization of STM during computation was presented. [28] presents a fully functional binary synapse utilizing two SHE-MTJs operated by relatively distinct read voltages to enhance synaptic learning efficiency. To the best of our knowledge, [28] is the only SHE-MTJ based design that offers a practical STM achieving the least synapse programming energy consumption (23.7 pJ) among all designs. The design proposed in [21] enhances the synapse programming energy consumption by 29.6% and 41% compared with memristor and MTJ designs. It should be noted that the STM state in [21] still incurs the capacitive network refresh power. From the STM-to-LTM transition delay standpoint, the design in [21] requires 30 ns, while memristor and MTJ designs require 80 and 30 ns, respectively, on constant stimulation.

TABLE 1. Performance analysis of STM-LTM architectures [21].

	[27]	[28]	[29]	[21]
Synapse	STT-MTJ	SHE-MTJ	Memristor	SHE-MTJ
Memory Implementation	No	Yes	No	Yes
Separate LTM/STM	No	Yes	No	Yes
STM computation	No	Yes	No	Yes
Refresh Needed	No	No	No	Yes
Synapse programming	110 pJ	23.7 pJ	92.4 pJ	65 pJ
Delay	30 ns	N/A	80 ns	30ns

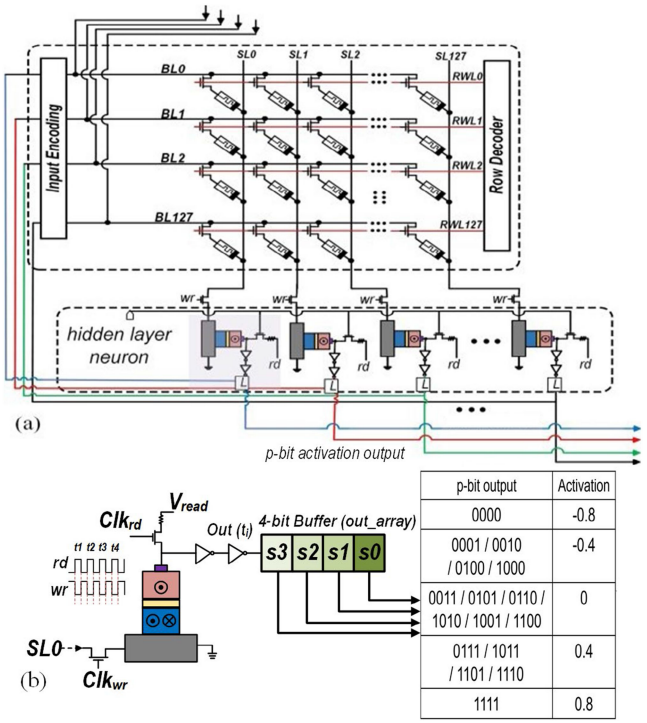
IV. HYBRID SPIN-ANALOG NEURON DESIGN

A. PREVIOUS NEURON DESIGNS

CMOS-based neuron implementations in prior works on ReRAM crossbar based PiM have shown to require large built-in truth tables with extra clock cycles leading to higher area and energy [30], [31]. Recently efficient hardware implementations of brain inspired neurons utilizing emerging NVM devices is being widely explored, to implement VMM operations via the intrinsic weighted summation capability of crossbar designs based on PiM architecture. The SHE-MTJ device shown in Figure 5a is considered to be low-barrier under the condition energy barrier $E_B << 40$kT, in which case thermal fluctuations at room temperature are sufficient to change the state of the device.

B. BINARY AND NON-BINARY NEURONS

The Long Short Term Memory (LSTM) networks requires sigmoid and tanh-based neurons for multiple gating purposes. Figure 5a shows circuit implementation of a sigmoidal behavior achieved by connecting an inverter to VDD and GND, provided the SHE-MTJ used in the circuit has $E_B <<$ 40kT. The time-averaged output of the device can provide both sigmoid and tanh function behaviors via slightly different circuit designs [14]. These output voltages are stored and mapped to a low- overhead Look-Up Table (LUT) which contains the voltage values. The hardware implementation of p-bit based stochastic neuron has been improved as delineated in [14] by adding two components, as shown in Figure 7b, along with a NN implementation shown in Figure 7a. To latch the output, a 4-bit buffer is inserted first corresponding to the four times of applying the crossbar output. Second, the neuron output is formed using a LUT. As shown in Figure 7, two complementary signals for *wr* and *rd* are considered. The *wr* signal goes high for each sample and based on the crossbar output current the p-bit device is



p-bit output	Activation
0000	-0.8
0001 / 0010 / 0100 / 1000	-0.4
0011 / 0101 / 0110 / 1010 / 1001 / 1100	0
0111 / 1011 / 1101 / 1110	0.4
1111	0.8

FIGURE 7. The proposed spin-based LSTM network with non-binary neurons [14].

programmed. To read out the device resistance and produce the output bit, the *wr* signal goes low and the *rd* signal goes high. The 4-bit buffered data is then given to the converter LUT which is prestored with the sampled floating-point activation values corresponding to output combinations in the buffer. For example, if the buffer content is 001, the LUT selects -0.4 as the output. This value can be triggered by any of 0001/0010/0100/1000 output bitstreams. Such non-binary neuron design is applicable in a variety of ANN applications needing non-linear and deterministic tanh and sigmoid activation functions.

C. CONFIGURABLE ANALOG MULTIPLIER FOR GENERALIZABLE ACTIVATION FUNCTION

The reconfigurable analog multiplier in [13] is based on the op-amp design presented in Figure 8a. The op-amp consists of two cascaded stages: an input stage consisting of a differential amplifier, followed by a gain stage. A simple op-amp design consisting of only 10 CMOS transistors as show in Figure 8b is chosen to optimize for power consumption as well as area and simulated using models from the PTM 14nm LSTP library, at VDD = 0.8V. The translinear principle is applied to attain exponentiation of the input signal [32]. As shown in Figure 8b, we introduce a three-stage design whose output is a power function of the input. The design accepts a single input for performing exponentiation operations; the design can also be reconfigured to accept two inputs for performing analog multiplication. The first stage, outlined in red in Figure 8b, is a logarithmic amplifier:

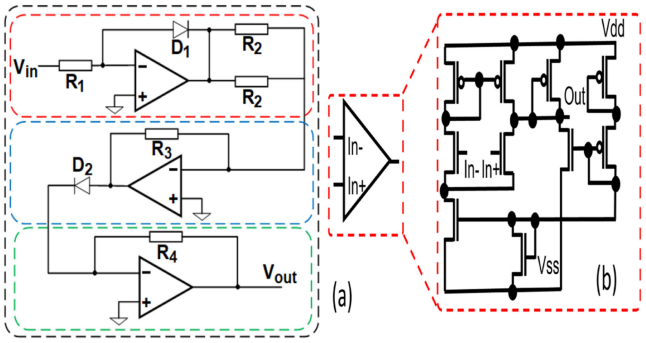


FIGURE 8. (a) Three stage Analog Multiplier, (b) Op-amp implementation comprised of 10 MOSFETs [13].

$$V_1 = -A_{OL}V_0 \tag{4}$$

$$-\frac{V_0 - V_{in}}{R_1} = I_{S1}\left[\exp\left(\frac{V_0 - V_{in}}{V_T}\right) - 1\right] \tag{5}$$

where A_{OL} is open loop gain and I_{S1} is the saturation current of diode D_1. Eq. (4) is from general op-amp theory and Eq. (5) follows from KCL. Thus, solving Eqs. (4) and (5) simultaneously yields:

$$V_1\left(1 + \frac{1}{A_{OL}}\right) = -V_T ln\left(\frac{V_{in} + \frac{V_1}{A_{OL}}}{R_1 I_{S1}} + 1\right) \tag{6}$$

In the limit of infinite open loop gain and sufficiently high input voltage, Eq. (6) is approximated as:

$$V_1 = -V_T \ln\left(\frac{V_{in}}{R_1 I_{S1}}\right) \tag{7}$$

The second stage is an analog adder, whereby a similar analysis yields $V_2 = \frac{2V_1 R_3}{R_2}$ Finally, the third stage is an anti-log amplifier with output approximately given by:

$$V_{out} = -R_4 I_{S2} e^{\frac{V_2}{V_T}} \tag{8}$$

where I_{S1} represents the saturation current of diode D_2. Overall, the output of this circuit is given by:

$$V_{out} = -e^{\frac{V_2}{V_T}} \frac{R_4 I_{S2}}{(R_1 I_{S1})^a} (V_{in})^a \tag{9}$$

where $a = 2R_3/R_2$,realizing any positive power function of the input as shown in Figure 8b. In addition, a dual-input stage consisting of two logarithmic amplifiers can be inserted to attain an analog multiplier. Finally, an inverting amplifier can also be inserted between the second and third stages to realize inverse power functions [13].

D. PROPOSED SELECTIVELY-RECONFIGURABLE ACTIVATION FUNCTION NEURON FUNCTIONALITY AND DESIGN

As mentioned, the sigmoid and tanh activation functions are the most commonly employed activation functions for inferencing tasks on neural networks. Herein, we go beyond

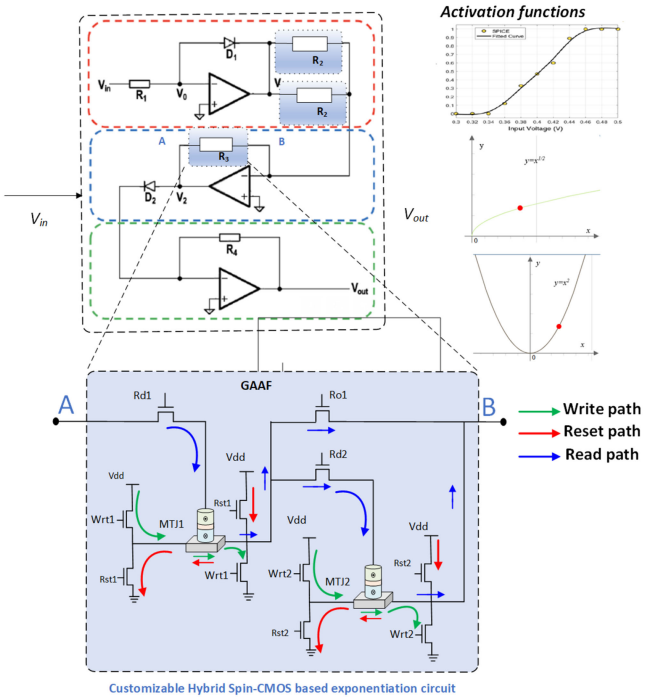


FIGURE 9. Proposed Generalizable Analog Activation Function based on customizable Hybrid Spin-CMOS devices.

previous work by realizing hardware for more expressive activation functions which can be runtime configured within the memory to achieve different variations in activation functions as per the target dataset/application. The hardware implementation of the proposed GAAF is demonstrated in Figure 9, based on the op-amp design presented in Figure 8. As delimited in [13], it can be seen that various exponential functions may result from this analog multiplier by varying $a = 2R_3/R_2$. Hence by varying resistances either R_2, or R_3 or both, more expressive activation functions can be generated. SHE-MTJs offer runtime configurable variable resistances based on parallel (P) and anti-parallel (AP) magnetization states (R_A, R_{AP}) determined by intrinsic device parameters. Herein we have only replaced R_3 in the feedback path of the op-amp with the hardware implementation depicted in Figure 9 and providing a control mechanism to demonstrate the various activation functions generated. Error rates, performance metrics and effects of process variation of GAAF are evaluated on network sizes 784*200*10 and 784*500*10 for MNIST dataset in Section VI. The input V_{in} to the GAAF is a sigmoid function output of the device shown in Figure 7b.

E. APPLICATION MAPPING AND EXECUTION MODEL

Herein we have implemented a series connection of two SHE-MTJs in the feedback path of the op-amp to analyze the feasibility of our design approach and the different activations achievable. Identical SHE-MTJs having design parameters listed in Table 2 are employed. The P and AP resistance values as obtained via SPICE simulations show the R_P and R_{AP} resistances of 2.8 KΩ and 5.6 KΩ respectively. Table 3 lists the

TABLE 2. SHE-MTJ simulation parameters.

Symbol	*Parameter*	*Value*
R_P	Parallel MTJ Resistance	2.8 KΩ
R_{AP}	Anti-Parallel MTJ Resistance	5.6 KΩ
TMR	Tunnel Magnetic Ratio	100%
α	Damping Coefficient	0.007
T	Temperature	300K
P	Polarization	0.52
V_{th_pmos}	Threshold Voltage (PMOS)	460mV
W_{pmos}	Width (PMOS)	44nm
V_{th_nmos}	Threshold Voltage (NMOS)	500mV
W_{nmos}	Width (NMOS)	22nm
MTJ Area	MTJ Length × MTJ Width × $\pi/4$	60nm×30nm×$\pi/4$
HM Volume	L × W × T	100 nm×60 nm×3 nm

KΩ = kilo-ohm, K = Kelvin, mV = milli-volt, nm = nanometer.

control signals required for configuration of the two SHE-MTJs during write phase, i.e., MTJ1 and MTJ2 in Figure 9 in P-OFF, AP-OFF, P-P, P-AP, AP-AP states respectively, where P is the parallel, AP anti-parallel, and OFF is the turned off state of MTJs, $V_{DD} = 0.8$ V. Since, in this phase the MTJs are being written their resistances, hence all the read signals are set to low (GND). Table 4 lists the corresponding resultant resistance values and activation functions generated from the GAAF unit upon reading the MTJs with a read voltage of 0.8V, and all the write and reset signals are set to low in this phase. Initially, MTJ1 is configured in parallel magnetization state and MTJ2 cutoff from the circuit by *Ro1* signal set to VDD via the pass transistor. In this case, the equivalent MTJ resistance evaluates to 2.8 KΩ and the output of GAAF evaluates the sigmoidal square root activation function. To switch the device to AP state, *Wrt1* is set to VDD = 0.8 V, and read signal *Rd1*, reset signal *Rs1* are kept low, such that write current passes along the heavy metal layer and the free layer magnetization switches to AP state. In this stage, with MTJ1 in AP state and MTJ2 OFF, the resultant equivalent MTJ series resistance evaluates to 5.8 KΩ, and inverted sigmoidal activation is evaluated by the GAAF. In a similar fashion, sigmoidal power of 3/2 activation function can also be produced by the GAAF unit, by suitably setting the control signals to their corresponding values in Table 3. A control unit takes care of the timing and setting of different control signals to appropriate voltages. Figure 10 shows the corresponding timing diagram of the various control signals and corresponding switching behavior of the two SHE-MTJs, evaluated on SPICE.

For software applications to utilize the SCAPE architecture, the execution mechanism needs additional software support. This is done congruent with the concept of Gather/Scatter techniques as illustrated in [33]. Although the premise of [33] and our work is distinct, the concept is expanded to support our architecture in the scenario of activation and access/write to multiple target cells located in a crossbar memory layout. This requires additional circuitry including modification to the control logic and memory decoder structure. Communication between the CPU and SCAPE is established via a 64-bit data bus and an address bus serving each crossbar layer. Our approach to utilize SCAPE capabilities is

TABLE 3. GAAF configuration phase control logic.

Switching transitions		*Control Signals*						
MTJ1	MTJ2	*Rd1*	*Rd2*	*Ro1*	*Wrt1*	*Rst1*	*Wrt2*	*Rst2*
P → AP	OFF	0	0	0	V_{DD}	0	0	0
AP → P	OFF	0	0	0	0	V_{DD}	0	0
P → AP	P → AP	0	0	0	V_{DD}	0	V_{DD}	0
AP → P	P → AP	0	0	0	0	V_{DD}	V_{DD}	0
P → AP	AP → P	0	0	0	V_{DD}	0	0	V_{DD}
AP → P	AP → P	0	0	0	0	V_{DD}	0	V_{DD}

via particular additions to the ISA including a `SET` operation for writing data to the array, `SCATTER` for activating word lines, and `GATHER` for reading output data. Besides the dynamic activation of multiple word lines for synaptic weight calculation as exhibited in Figure 4 of our manuscript, SCAPE provides infield configurability of Hybrid Spin/CMOS-based Analog/Digital Blocks to enable hardware for more expressive neural network activation functions. Thus, the GAAF units can be runtime configured within the memory array to achieve various activation functions as per the target dataset/application. A generalized activation function is developed in the manuscript, which is shown to achieve better recognition rate for MNIST dataset. The activation of target GAAF neurons is achievable by introducing two new instructions into the ISA, i.e., `ACTIVATE` and `EVALUATE`. The following is an overview of the ISA modifications required for functionality of SCAPE:

1: `SET (REGID, addr)` which is used to write the data from CPU register specified by REGID to a specific SCAPE memory cell specified by `addr`. In this context, `addr` can be broken down to {`layerID`, `rowID`, `columnID`} to identify a specific crossbar memory cell. SET is used to load matrix data, and input data, into SCAPE; a `columnID` of 0 is used to denote input vector data.

2: `SCATTER (REGID, layerID, WL1, WL2)` which is used to set all of the word lines between `WL1` and `WL2` in a specified layer of SCAPE, using the configuration data initially stored in `REGID`. This is achieved at the hardware level through a latch/reset mechanism similar to that described in [34].

3: `GATHER (REGID, layerID, BL1, BL2)` which is used to load output data from a range of bit lines in a specific layer of SCAPE into the CPU register labeled `REGID`.

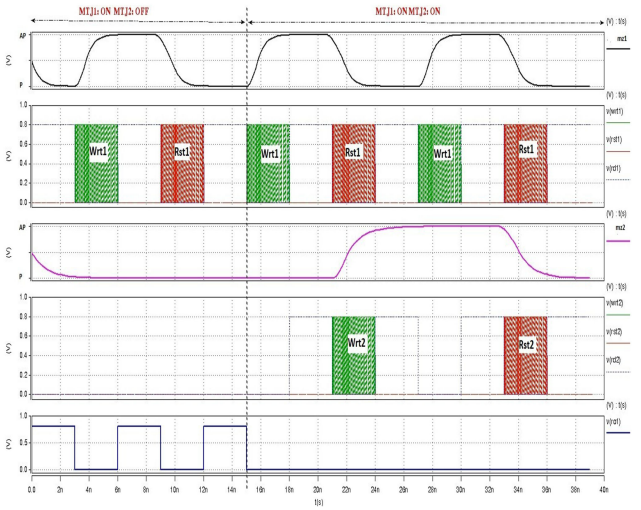


FIGURE 10. Control signal mapping for GAAF configuration and evaluation stages.

4: `ACTIVATE (layerID, configID)` that configures the GAAF units by setting internal MTJ values to their required 'P' or 'AP' or 'OFF' (disconnected from circuit) orientations based on the desired neuron activation functions. The parameter `layerID` identifies the particular GAAF enhanced neuron layer in the SCAPE to be activated. The `configID` in SCAPE is a 3-bit identifier corresponding to each of the six unique combinations of MTJ1 and MTJ2 resistance states in the GAAF neuron, as listed in Table 4, which achieves a specific activation function, by generating the corresponding control signals through the control logic circuitry. For instance, in order to generate an inverted sigmoidal activation function, MTJ1 and MTJ2 are configured be in 'AP' and 'OFF' states in the circuit, respectively. As such, a `configID` of '000' generates the required control signal values as listed in row one of Table 3, to set the MTJs to their required states.

5: `EVALUATE (layerID, funcID)` that generates the desired neuron activation function at the GAAF output. The `funcID` denotes the type of activation function that we want the GAAF neuron to output. The `funcID` is encoded as a 2-bit identifier generating the control signals corresponding to evaluating one of the four unique functions: inverted sigmoid, $(\text{sigmoid})^2$, $(\text{sigmoid})^{3/2}$, and $(\text{sigmoid})^{1/2}$ at the GAAF output, as listed in Table 4.

TABLE 4. GAAF evaluation/read phase operation and control logic.

Resistance State		*Total Series Resistance*	*Control Signals*							*Activation function*
MTJ1	MTJ2		*Rd1*	*Rd2*	*Ro1*	*Wrt1*	*Rst1*	*Wrt2*	*Rst2*	
P	OFF	$R_{P1} = 2.8K\Omega$	V_{DD}	0	V_{DD}	0	0	0	0	*Sig. Sq. root* $\sqrt{V_{in}}$
AP	OFF	$R_{AP1} = 5.6K\Omega$	V_{DD}	0	V_{DD}	0	0	0	0	*Inv.Sig. -* V_{in}
P	P	$R_{P1} + R_{P2} = 5.6K\Omega$	V_{DD}	V_{DD}	0	0	0	0	0	*Inv.Sig. -* V_{in}
AP	P	$R_{AP1} + R_{P2} = 8.4K\Omega$	V_{DD}	V_{DD}	0	0	0	0	0	*Sig. Pow(3 / 2)* $V_{in}^{\wedge(3/2)}$
P	AP	$R_{P1} + R_{AP2} = 8.4K\Omega$	V_{DD}	V_{DD}	0	0	0	0	0	*Sig. Pow(3 / 2)* $V_{in}^{\wedge(3/2)}$
AP	AP	$R_{AP1} + R_{AP2} = 11.2k\Omega$	V_{DD}	V_{DD}	0	0	0	0	0	*Sig. Sq.* V_{in}^2

Algorithm 1. Approximate Message Passing

Inputs: Measurement matrix Φ, Measurement vector $\mathbf{y}$, # of measurements m

Output: Approximate signal vector, $\hat{\mathrm{x}}$

Procedure: 1) Initialize residual $\mathrm{r}_0 = y$, Signal approximation $\hat{\mathrm{x}}_0 = 0$, counter $i = 1$

while $i < k$ **do**

2) $\theta = \|\mathrm{r}_{i-1}\|/\sqrt{m}$

3) $\mathrm{a} = \hat{\mathrm{x}}_{\mathrm{i}-1} + \Phi^{\mathrm{T}}\mathrm{r}_{i-1}$

4) $\hat{\mathrm{x}}_{\mathrm{i}} = \mathrm{sign}(\mathrm{a})\max(|\mathrm{a}| - \theta,\ 0)$

5) $b_i = \|\hat{\mathrm{x}}\|0/m$

6) $\mathrm{r}_i = \mathrm{y} - \Phi\hat{\mathrm{x}}_i + b_i\mathrm{r}_{i-1}$

end while

V. INTRINSIC VMM ON SCAPE: CS APPLICATIONS

Compressive Sensing (CS) allows for reduction in transmission and storage overheads of spectrally-sparse and wideband data by sampling at the information rate rather than the Nyquist rate [13]. As such, by limiting the number of samples taken per frame, CS provides a solution to unprecedented challenges associated with 5G communication, including complexity and power consumption associated with increased bandwidths. Implementing CS sampling and reconstruction in hardware presents unique challenges. Sampling requires the use of a random number generator, which is traditionally implemented using a Linear Feedback Shift Register (LFSR) that can present significant power and area overheads. Qian et al. [35] introduced memristive crossbar arrays for VMM operations during CS sampling, observing that signal reconstruction using $\ell 1$-minimization yields similar Signal to Noise Ratios (SNRs) to that of using a Gaussian matrix. CS entails sampling a signal of length n using m measurements, with $m \ll n$. Sampling is achieved through the linear transformation $\boldsymbol{y} = \boldsymbol{Ax}$, where $\boldsymbol{x} \in \mathbb{R}^n$ is the signal vector , $\boldsymbol{y} \in \mathbb{R}^m$ is the measurement vector, and $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is the measurement matrix. Reconstruction of the original signal can be accomplished at the receiving end by solving the basis pursuit problem:

$$\hat{\boldsymbol{x}} = \ \mathrm{argmin}\|\boldsymbol{x}\|_1 \ \mathrm{s.t.}\ \boldsymbol{y} \ = \ \boldsymbol{A}\hat{\boldsymbol{x}} \tag{13}$$

where $\|\boldsymbol{x}\|_1$ represents the ℓ_1 norm of $\boldsymbol{x}$. It can be shown that the signal vector can be reconstructed if the signal is sufficiently sparse, and $\boldsymbol{A}$ satisfies the Restricted Isometry Property, i.e., if for any k-sparse vector $\boldsymbol{x}$,

$$\|\boldsymbol{x}\|_2^2(1-\delta) \le \|\boldsymbol{\Phi x}\|_2^2 \le \|\boldsymbol{x}\|_2^2(1+\delta),\ 0 < \delta < 1 \tag{14}$$

A variety of algorithms have been developed as alternatives to basis pursuit for the purpose of CS reconstruction. For instance, *Approximate Message Passing (AMP)* serves as a soft thresholding algorithm optimized for fast convergence [36]. The design is shown as Algorithm 1. In Line 1, the AMP algorithm initializes the residual vector, $\mathbf{r_0}$, to the measurement vector $\mathbf{y}$, as well as initializing the estimate of the signal vector $\hat{\boldsymbol{x}}$ to zero. Line 2 computes the threshold, $\boldsymbol{\theta}$, as the root mean square error of the residual. Next, Lines 3 – 4 follow the Iterative Soft Thresholding technique [37] to generate an estimate of the reconstructed signal vector. The notation in Line 4 refers to elementwise operations on the components of vector $\mathbf{a}$, with the function sign(x) defined as -1 when $x < 0$ and as 1 when $x > 0$. Finally, Lines 5 – 6 update the residual, $\|\hat{\boldsymbol{x}}_{\boldsymbol{i}}\|_0$, based on the current estimate of the signal as well as the residual of the previous iteration, $\mathbf{r_{i-1}}$.

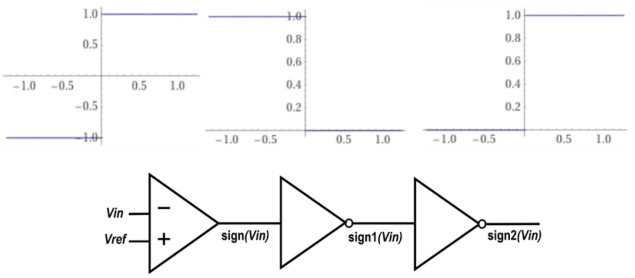


FIGURE 11. **An analog design for thresholding operations.**

The AMP algorithm is implemented using the SCAPE hardware architecture presented in Figure 3. AMP requires vector-matrix multiplication operations, which are executed using the VMMS. Furthermore, a three-stage analog circuit based on the design shown in Figure 8 is used for basic arithmetic operations, including multiplication (by use of a dual first-input stage), addition (using the second computational stage) and exponentiation operations such as square, square root and inverse square root. Besides the operations listed above, AMP requires thresholding operations which are also achievable with the AAS using the simple analog design shown in Figure 11. In this design, an analog comparator circuit computes the function $y = \mathrm{sign}(x)$ when $\mathrm{V_{ref}} = 0$. A three-stage design based on a chain of inverters is used for the computation of two-additional functions: $y = \mathrm{sign1}(x,\mathrm{ref})$, defined as 1 when $x<$ref and as 0 when $x>$ref, and $y = \mathrm{sign2}$ (x,ref), defined as 1 when $x>$ref and as 0 when $x<$ref. Based on this hardware, the remaining three functions necessary for AMP may be computed. First, $y = |\mathrm{x}|$ is rewritten as $y = x\mathrm{sign}(x)$. Next, $y = \max(x,0)$ is equivalent to $y = x\mathrm{sign2}$ $(x,0)$. Finally, $y = \|\boldsymbol{x}\|_0$ is roughly equivalent to $y = \sum(\mathrm{sign1}(x, 0.05) + \mathrm{sign2}(x, -0.05))$, assuming any input with an absolute value greater than 0.05 is considered as "non-zero." Figure 12 demonstrates a hardware implementation of one loop of the AMP algorithm, based on the architecture presented herein. Reconstruction based on a signal size $n = 256$ and $m = 64$ requires a 256×64 VMMS array to execute the VMM operations in Line 4 and Line 6, and 256 AAS functional units for scalar operations. Performance analysis of the AMP algorithm using the proposed SCAPE topology is demonstrated in Section VI B.

VI. RESULTS AND ANALYSIS

A. BENCHMARK VALIDATION ON MNIST DATASET FOR ML

For evaluating our SCAPE topology, MNIST data set containing 70000 images has been utilized, out of which 3000

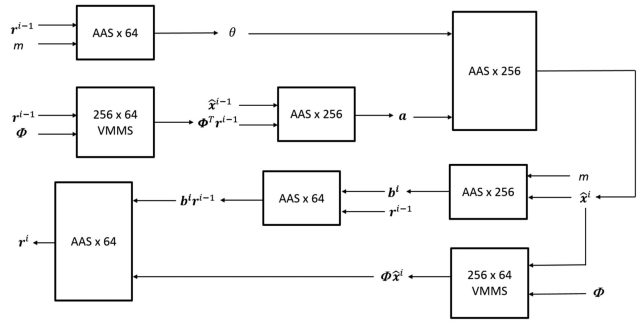


FIGURE 12. **Hardware implementation of AMP algorithm.**

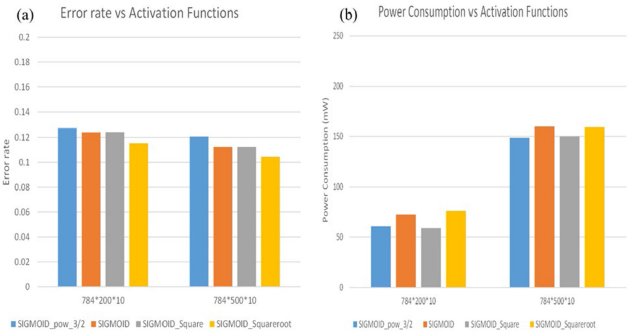


FIGURE 13. **(a) Error rates, and (b) Overall Power (milli-Watts) consumption of four GAAF activations for 2 ANNs (784x200x10; 784x500x10).**

images are employed for training the ANN. The trained weights and biases obtained for the network are accordingly assigned to the crossbar array, and testing is done for the hardware network using the 100 test images and PIN-Sim framework [12]. Figure 13 shows the error rate obtained at the final layer of the SCAPE topology in Figure 3, and the overall power consumption of the ANN for the four activation functions namely, $(\text{sigmoid})^2$, sigmoid, $(\text{sigmoid})^{3/2}$ and $(\text{sigmoid})^{1/2}$. Figure 13a shows that accuracy achieved by the sigmoidal square-root activation function is best with lowest error rate, sigmoidal power (3/2) performs worst, whereas baseline sigmoidal and square achieve similar error rates for all the topologies for MNIST dataset evaluated using PIN-Sim [12]. Figure 13b shows that the overall power consumption for the sigmoid square root activation is comparable to the power consumption of plain sigmoidal activation function. Switching from one activation function to other is achieved by GAAF configuration as mentioned previously. Appropriate control signals are given to the block so that the MTJ's switch between P and AP states to get the desired activation function. Table 5 represents the comparison of GAAF performance for different activation functions with other digital/analog activation function generators. It can be observed that the number of components used in GAAF block is less with comparable power consumption and delay, as with other circuits in literature. Table 6 lists the error rate, average DBN power consumption, and power-error-product of proposed SCAPE topology for various sized ANNs and activation functions evaluated on MNIST dataset. The Power Error Product (PEP) metric is also calculated as a product of power consumption and error rate to better establish the error efficiency of the SCAPE topology compared to plain sigmoidal activation function. PEP for sigmoidal square root activation function for 784x200x10 topology was observed to be the lowest i.e., most efficient. For datasets larger than MNIST, SCAPE limits accuracy loss and accumulated current associated with larger arrays by matrix partitioning using a similar method described in [41].

B. COMPARATIVE ANALYSIS OF CS AMP ALGORITHM

To determine the total energy cost of AMP, SPICE simulations are performed to determine the per-cell energy cost of the VMMS, as well as the energy cost per operation of the scalar functions performed by the AAS; the results are aggregated to determine the total computational energy cost of running one cycle of AMP. The VMMS consumes a total of 3.15nJ while total energy consumption by the AAS is 2.02nJ, for a total computational energy consumption of 5.17nJ. For 50 iterations, this gives an energy overhead equal to 258nJ for running AMP. Analysis of signal reconstruction error associated with approximations in the AAS units was performed for a signal of size $n = 1000$, and sparsity $k = 100$, where n is the total number of elements in each frame of the signal, and k is the total number of elements per frame that are non-zero. The average accuracy degradation resulting from computational error was found to be 1.1dB, which is negligible. Table 7 lists the breakdown of energy per computation in execution of a single AMP cycle using the proposed design. A total energy cost of 5.17nJ per cycle yields a total energy consumption of 1.0nJ per sample, assuming 50 iteration cycles and a reconstructed signal consisting of 256 samples. Table 8 displays an energy comparison to two recent ASIC implementations for AMP; hardware running

TABLE 5. **Performance comparison of GAAF.**

	[32]	[38]	[39]	[40]	Herein	Herein
Mode	Analog	Digital	Digital	Analog	Analog	Analog
Operation	Square	Multiplier	Square root	Square	Square root	Square
Tech node	180nm	28nm	45nm	500nm	14nm	14nm
V_{DD}	1.3V	1V	1V	1.5V	0.8V	0.8V
#Components	100	~1000	>1000	12	55+2 SHE-MTJs	55+2 SHE-MTJs
Power	149mW	126mW	21.02mW	600mW	121mW	126mW
Delay	N/A	0.8ns	3.61ns	N/A	6.4ns	3.5ns

TABLE 6. Power error product of sigmoid activation vs. SCAPE topology for various network sizes.

Attributes	Activation function			
	Sigmoid		GAAF enhanced Sigmoid + square root	
ANN	784×200 ×10	784×500 ×10	784×200 ×10	784×500 ×10
Error rate	0.1239	0.1124	0.1152	0.1046
Power(mW)	72.4	160.1	76.1	159.5
PEP	8.97	18	8.77	16.68

TABLE 7. Breakdown of AMP circuit energy consumption.

Operation	Hardware Units	Energy Cost
$\|\|r^{i-1}\|\|.$	AAS	47.6 pJ
$\theta = \|\|r^{i-1}\|\|/\sqrt{m}$	AAS	1.1 pJ
$a = \hat{x}^{i-1} + \Phi^T r^{i-1}$	VMMS + AAS	1.654 nJ
$\hat{x}^i = \text{sign}(a)\max(\text{abs}(a) - \vartheta, 0)$	AAS	1.24 nJ
$b^i = \|\|\hat{x}^i\|\|_0/m$	AAS	0.58 nJ
$r^i = y - \Phi\hat{x}^i + b^i r^{i-1}$	VMMS + AAS	1.65 nJ
Total		5.17 nJ

the Enhanced AMP algorithm (EAMP) [42] over 50 iterations under the same CS parameters of $(n,m) = (256,64)$ consumes 315mW of power and executes in 8900 clock cycles on a 400MHz system. Thus, the energy consumption is roughly 7μJ, and roughly 27nJ per sample. EAMP is roughly in line with the standard AMP algorithm in terms of mean square error, up to 100 iterations. Thus, the full-analog approach to AMP presented herein provides significant benefits in energy while having a minimal impact on reconstruction accuracy.

C. PROCESS VARIATION (PV) ANALYSIS

Two justified concerns facing analog computation are sensitivity to noise, and the ability to deliver sufficient accuracy in the computation. Approaches to mitigating variation and adapting operational tolerances span design margin, redundancy, and reconfiguration [44], [45]. Device parameters such as Anisotropy field (H_k), Diameter (d) and Thickness (t) for the MTJ's may vary due to the process variation (PV) in MTJ fabrication, resulting in changes in R_P and R_{AP} resistance values. Inconsistencies in R_P and R_{AP} result in variations in activation function, thereby affecting the inference accuracy of the NN hardware. Figure 14 depicts the deviation in square and square root activation functions due to PV in the GAAF MTJs, using 100-trial Monte-Carlo (MC) simulation runs in SPICE with standard deviation (SD) of 5% for MTJ length, width, thickness, V_{in} represents the input to the GAAF and V_{out} represents the output obtained by using (9), where I_{s1}, I_{s2} are the diode saturation currents. R_1, R_2, R_4 are the resistance values in the multiplier circuit. R_3 (2.8 KΩ/5.6 KΩ/8.4 KΩ/ 11.2 KΩ) is decided by the state of MTJ's, thereby determining the neuron activation function in the network. A deviation of 5% in R_3 resistance value of 2.8 KΩ of the GAAF with a sigmoidal square root activation function was found to result

TABLE 8. Comparison of AMP energy consumption.

	Herein	Herein	[37]	[42]
Tech. node	14nm	14nm	65nm	65nm
V_{DD}	0.8V	0.8V	1.2V	N/A
Array size	256x64	1024x512	1024x512	256x64
Array precision	8 bits	8 bits	26 bits	1 bit
#Iterations	50	20	20	50
Energy/sample	1.0nJ	2.1nJ	61nJ	27nJ

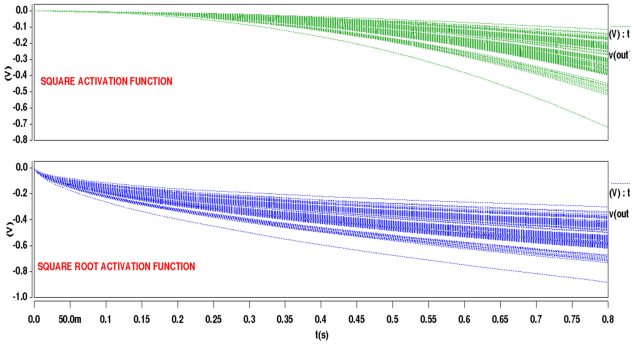


FIGURE 14. Effect of PV on two activation functions applying 5% SD on MTJ Length, width, and thickness.

in a maximum 5% increase in ANN inference error rates using PIN-Sim framework [12] on the MNIST dataset.

VII CONCLUSION

In this article, we explored a 2D array-based approach to PiM by developing the SCAPE topology targeting efficient analog activation. Namely, an innovative GAAF based on spin-configurable activation function computes more expressive activation functions intrinsically in analog. Realization of AMP signal processing algorithm show ∼95% reduction in energy consumption at comparable accuracy. Simulation results of power consumption and error rate for MNIST dataset using sigmoidal square root activation of GAAF shows up to 7% accuracy improvement versus baseline conventional sigmoidal activation. Future work includes adding enhanced functionality to GAAF and evaluating effects on more varied datasets for additional real-world applications.

REFERENCES

[1] W. Wang and M. Zhang, "Tensor deep learning model for heterogeneous data fusion in Internet of Things," *IEEE Trans. Emer. Topics Comp. Intell.*, vol. 4, no. 1, pp. 32–41, Feb. 2018.

[2] S. Wen, H. Wei, Z. Zeng, and T. Huang, "Memristive fully convolutional network: An accurate hardware image-segmentor in deep learning," *IEEE Trans. Emer. Topics Comp. Intell.*, vol. 2, no. 5, pp. 324–334, Oct. 2018.

[3] Y. Kunpeng et al., "Reinforcement learning-based mobile edge computing and transmission scheduling for video surveillance," *IEEE Trans. Emerg. Topics Comp.*, vol. 10, no. 2, pp. 1142–1156, Second Quarter 2022, doi: 10.1109/TETC.2021.3073744.

[4] M. Taghavi and M. Shoaran, "Hardware complexity analysis of deep neural networks and decision tree ensembles for real-time neural data classification," in *Proc. IEEE Int. Conf. Neural Eng.*, 2019, pp. 407–410.

[5] G. H. Barnes, R. M. Brown, M. Kato, D. J. Kuck, D. L. Slotnick, and R. A. Stokes, "The ILLIAC IV computer," *IEEE Trans. Comput.*, vol. 100, no. 8, pp. 746–757, Aug. 1968.

[6] C. C. Foster, *Content Addressable Parallel Processors*, Hoboken, NJ, USA: Wiley, 1976.
[7] R. F. DeMara and D. I. Moldovan, "The SNAP-1 parallel AI prototype," *IEEE Trans. Parallel Distrib. Syst.*, vol. 4, no. 8, pp. 841–854, Aug. 1993.
[8] D. Patterson et al., "A case for intelligent RAM," *IEEE Micro*, vol. 17, no. 2, pp. 34–44, Mar./Apr. 1997.
[9] D. G. Elliott, M. Stumm, W. M. Snelgrove, C. Cojocaru, and R. McKenzie, "Computational RAM: Implementing processors in memory," *IEEE Des. Test Comput.*, vol. 16, no. 1, pp. 32–41, Second Quater 1999.
[10] F. Alibart, E. Zamanidoost, and D. Strukov, "Pattern classification by memristive crossbar circuits using Ex situ and in situ training," *Nature Commun.,* vol. 4, 2013, Art. no. 2073, doi: 10.1038/ncomms3072.
[11] H. Zhang, W. Kang, K. Cao, B. Wu, Y. Zhang, and W. Zhao, "Spintronic processing unit in spin transfer torque magnetic random access memory," *IEEE Trans. Electron Devices*, vol. 66, no. 4, pp. 2017–2022, Apr. 2019.
[12] H. Pourmeidani, S. Sheikhfaal, R. Zand, and R. F. DeMara, "Probabilistic interpolation recoder for energy-error-product efficient DBNs with P-Bit devices," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 4, pp. 2146–2157, Fourth Quarter 2021.
[13] A. Tatulian and R. F. DeMara, "Generalized exponentiation using STT magnetic tunnel junctions: Circuit design, performance, and application to neural network gradient decay," *SN Comput. Sci.*, vol. 3, no. 2, 2022, Art. no. 148.
[14] S. Sheikhfaal, M. R. Vangala, A. Adepegba, and R. F. DeMara, "Long short-term memory with spin-based binary and non-binary neurons," in *Proc. IEEE Int. Midwest Symp. Circuits Syst.*, 2021, pp. 317–320, doi: 10.1109/MWSCAS47672.2021.9531773.
[15] P. Chi, S. Li, Y. Cheng, Y. Lu, S. H. Kang, and Y. Xie, "Architecture design with STT-RAM: Opportunities and challenges," in *Proc. Asia South Pacific Des. Automat. Conf.*, 2016, pp. 109–114, doi: 10.1109/ASPDAC.2016.7427997.
[16] S. Miura et al., "Scalability of quad interface p-MTJ for 1X nm STT-MRAM With 10-ns low power write operation, 10 years retention and endurance> 10^{11}," *IEEE Trans. Electron Devices*, vol. 67, no. 12, pp. 5368–5373, Dec. 2020.
[17] S. Verma and B. K. Kaushik, "Low-power high-density STT MRAMs on a 3-D vertical silicon nanowire platform," *IEEE Trans. Very Large-Scale Integration (VLSI) Syst.*, vol. 24, no. 4, pp. 1371–1376, Apr. 2016, doi: 10.1109/TVLSI.2015.245 4859.
[18] V. K. Joshi, P. Barla, S. Bhat, and B. K. Kaushik, "From MTJ device to hybrid CMOS/MTJ circuits: A review," *IEEE Access*, vol. 8, pp. 194105–194146, 2020.
[19] S. Jain, A. Ranjan, K. Roy, and A. Raghunathan, "Computing in memory with spin-transfer torque magnetic RAM," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 3, pp. 470–483, Mar. 2018, doi: 10.1109/TVLSI.2017.2776954.
[20] S. S. P. Parkin, R. E. Fontana, and A. C. Marley, "Low-field magnetoresistance in magnetic tunnel junctions prepared by contact masks and lithography: 25% magnetoresistance at 295 K in mega-ohm micron-sized junctions," *J. Appl. Phys.*, vol. 81, no. 8, 1997, Art. no. 5521, doi: /10.1063/1.364588.
[21] S. Sheikhfaal and R. F. Demara, "Short-term long-term compute-in-memory architecture: A hybrid spin/CMOS approach supporting intrinsic consolidation," *IEEE J. Explor. Solid-State Comput. Devices Circuits*, vol. 6, no. 1, pp. 62–70, Jun. 2020, doi: 10.1109/JXCDC.2020.2983450.
[22] Cilingiroglu, "A purely capacitive synaptic matrix for fixed-weight neural networks," *IEEE Trans. Circuits Syst.*, vol. 38, no. 2, pp. 210–217, Feb. 1991.
[23] D. Kwon and I. Y. Chung, "Capacitive neural network using charge-stored memory cells for pattern recognition applications," *IEEE Electron Device Lett.,* vol. 41, no. 3, pp. 493–496, Mar. 2020.
[24] Z. Wang et al., "Capacitive neural network with neuro-transistors," *Nature Commun.*, vol. 9, no. 1, pp. 1–10, Dec. 2018.
[25] S. Angizi and D. Fan, "ReDRAM: A reconfigurable processing-in-DRAM platform for accelerating bulk bit-wise operations," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Des.*, 2019, pp. 1–8.
[26] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, "NVSim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 31, no. 7, pp. 994–1007, Jul. 2012.
[27] A. Sengupta and K. Roy, "Short-term plasticity and long-term potentiation in magnetic tunnel junctions: Towards volatile synapses," *Phys. Rev. A Gen. Phys. Appl.*, vol. 5, no. 2, Feb. 2016, Art. no. 024012.
[28] 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, no. 1, Sep. 2016, Art. no. 29545.
[29] T. Chang, S. H. Jo, and W. Lu, "Short-term memory to long-term memory transition in a nanoscale memristor," *ACS Nano*, vol. 5, no. 9, pp. 7669–7676, Sep. 2011.
[30] Y. Long, E. M. Jung, J. Kung, and S. Mukhopadhyay, "ReRAM crossbar based recurrent neural network for human activity detection," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2016, pp. 939–946.
[31] Y. Long, T. Na, and S. Mukhopadhyay, "ReRAM-based processing-in-memory architecture for recurrent neural network acceleration," *IEEE Trans. Very Large Scale Int.(VLSI) Syst.*, vol. 26, no. 12, pp. 2781–2794, Dec. 2018.
[32] R. J. D'Angelo and S. R. Sonkusale, "A time-mode translinear principle for nonlinear analog computation," *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 62, no. 9, pp. 2187–2195, Sep. 2015, doi: 10.1109/TCSI.2015.2451912.
[33] V. Seshadri et al., "Gather-scatter DRAM: In-DRAM address translation to improve the spatial locality of non-unit strided accesses," in *Proc. Int. Symp. Microarchitecture*, 2015, pp. 267–280, doi: 10.1145/2830772.2830820.
[34] S. Li, C. Xu, Q. Zou, J. Zhao, Y. Lu, and Y. Xie, "Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories," in *Proc. Des. Automat. Conf.*, 2016, Art. no. 173, doi: 10.1145/2897937.2898064.
[35] F. Qian, Y. Gong, G. Huang, M. Anwar, and L. Wang, "Exploiting memristors for compressive sampling of sensory signals," *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, vol. 26, no. 12, pp. 2737–2748, Dec. 2018.
[36] L. Bai, P. Maechler, M. Muehlberghuber, and H. Kaeslin, "High-speed compressed sensing reconstruction on FPGA using OMP and AMP," in *Proc. IEEE Int. Conf. Electron. Circuits Syst.*, 2012, pp. 53–56.
[37] P. Maechler et al., "VLSI design of approximate message passing for signal restoration and compressive sensing," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 2, no. 2, pp. 569–590, Sep. 2012.
[38] H. Jiang, C. Liu, F. Lombardi, and J. Han, "Low-power approximate unsigned multipliers with configurable error recovery," *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 66, no. 1, pp. 189–202, Jan. 2018, doi: 10.1109/TCSI.2018.2856245.
[39] N. Arya, T. Soni, M. Pattanaik, and G. K. Sharma, "Area and energy efficient approximate square rooters for error resilient applications," in *Proc. IEEE 33rd Int. Conf. VLSI Des., 19th Int. Conf. Embedded Syst.*, 2020, pp. 90–95, doi: 10.1109/VLSID49098.2020.00033.
[40] M. T. Abuelma'Atti and A. M. Abuelmaatti, "A new current-mode CMOS analog programmable arbitrary nonlinear function synthesizer," *Microelectronics J.*, vol. 43, no. 11, pp. 802–808, 2012, doi: 10.1016/j.mejo.2012.07.003.
[41] B. R. Fernando, Y. Qi, C. Yakopcic, and T. M. Taha, "3D memristor crossbar architecture for a multicore neuromorphic system," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2020, pp. 1–8, doi: 10.1109/IJCNN48605.2020.9206929.
[42] K. N. S. Batta and I. Chakrabarti, "VLSI architecture for enhanced approximate message passing algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 9, pp. 3253–3267, Sep. 2020.
[43] P. Chi et al., "Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory," *ACM SIG ARCH Comput. Archit. News*, vol. 44, no. 3, pp. 27–39, 2016.
[44] N. D. P. Avirneni and A. Somani, "Low overhead soft error mitigation techniques for high-performance and aggressive designs," *IEEE Tran. Comput.*, vol. 61, no. 4, pp. 488–501, Apr. 2012.
[45] W. Zhao et al., "A radiation hardened hybrid spintronic/CMOS nonvolatile unit using magnetic tunnel junctions," *J. Phys. D: Appl. Phys.*, vol. 47, no. 40, Art. no. 405003.

MOUSAM HOSSAIN (Student Member, IEEE) received the MS degree in computer engineering from the Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND, in 2019, on Formal Verification of Asynchronous designs. She is currently working toward the doctoral degree in computer engineering with the Computer Architecture Laboratory (CAL), University of Central Florida (UCF). Her research interests include computer architecture, post-CMOS devices, non-volatile memories, asynchronous designs. She is a member of IEEE-HKN.

ADRIAN TATULIAN (Student Member, IEE) received the BSc degree in physics from the University of Central Florida, Orlando, FL, in 2013. He is currently working toward the PhD degree in computer engineering with the University of Central Florida, Orlando, FL. His research interests include analog arithmetic, reconfigurable computing, and spin-based hardware for machine learning and compressive sensing applications.

SHADI SHEIKHFAAL (Student Member, IEEE) received the BSc degree in computer engineering from Azad University, Ardebil, Iran, in 2012, the MSc degree in computer engineering and computer systems architecture from the Science and Research Branch, Azad University, Tehran, Iran, in 2014, and the PhD degree in computer engineering from the University of Central Florida, Orlando, FL. Her current research interests include biologically inspired computing, neuromorphic computing, and spin-based computing.

HARSHAVARDHAN R. THUMMALA (Student Member, IEEE) received the BS degree in electronics and communication engineering from Jawaharlal Nehru Technological University, Telangana, in 2019. He is currently working toward the MS degree in electrical engineering with the University of Central Florida, His current research interests include reconfigurable computer architecture, field programmable gate arrays, neuromorphic computing, and Spin-based computing.

RONALD F. DEMARA (Senior Member, IEEE) is currently pegasus professor with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL, where he has been a full-time faculty member, since 1993. His research interests include computer architecture, post-CMOS devices, and reconfigurable fabrics with applications to intelligent and neuromorphic systems, on which he has published more than 320 articles and holds one patent. He received the Joseph M. Biedenbach Outstanding Engineering Educator Award from the IEEE. He has served seven terms as a Topical Editor and/or associate editor for the *IEEE Transactions on Computers*, the *IEEE Transactions on Emerging Topics in Computing*, the *IEEE Transactions on Very Large Scale Integration (VLSI)*, the *IEEE Spectrum*, and Technical Program Committees of various IEEE conferences. He has been a Keynote Speaker of the IEEE Reconfigurable Architectures Workshop, IEEE ReConFig, and IEEE IEMtronics conferences. He has been a guest editor of the *IEEE Transactions on Computers* 2017 Special Section on Innovation in Reconfigurable Fabrics and 2019 Special Section on Non-Volatile Memories.