

SSRL: Single Skyrmion Reconfigurable Logic Utilizing 2-D Magnus Force on Magnetic Racetracks

MOHAMMAD NAZMUS SAKIB¹ (Member, IEEE), HAMED VAKILI^{2,3},
SAMIRAN GANGULY^{1,4}, AVIK W. GHOSH^{1,2} (Senior Member, IEEE),
and MIRCEA STAN¹ (Fellow, IEEE)

¹Charles L. Brown Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22903 USA

²Department of Physics, University of Virginia, Charlottesville, VA 22903 USA

³Department of Physics and Astronomy and Nebraska Center for Materials and Nanoscience,
University of Nebraska, Lincoln, NE 68588 USA

⁴Department of Electrical and Computer Engineering, Virginia Commonwealth University, Richmond, VA 23284 USA

CORRESPONDING AUTHOR: M. N. SAKIB (ms3xy@virginia.edu)

This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) Topological Excitations in Electronics (TEE) Program under Grant D18AP00009. (Mohammad Nazmus Sakib and Hamed Vakili contributed equally to this work.)

ABSTRACT Magnetic racetrack memory has frequently been complicated by the pinning of domain wall bits on the one hand and the need to engineer precise synchronization and inter-track repulsion between skyrmionic bits on the other. Such proposals, however, do not capitalize on the complex 2-D motion of skyrmions, such as transverse Magnus force that tends to deviate the skyrmion trajectory from rectilinear motion along the current drive. The transverse deviation associated with such a skyrmion Hall effect is normally considered a liability for skyrmions, and efforts have focused on eliminating rather than utilizing it for proposed device applications. We propose a simple single skyrmion-based circuit macro with elementary and higher-order logic gates that utilize Magnus force and propose reconfigurable logic built on these gates. We demonstrate the reliability of the proposed approach with micromagnetics simulation. The energy consumption in this circuit lies mainly in the overhead, with the racetrack consuming a small fraction. The energy–delay product (EDP) is correspondingly low and can be improved by boosting the skyrmion speed.

INDEX TERMS Domain walls, magnetic tunnel junction (MTJ), racetrack, reconfigurable logic, skyrmions, voltage-controlled magnetic anisotropy (VCMA).

I. INTRODUCTION

RECONFIGURABLE hardware tends to offer increased performance per watt than general-purpose processors [1]. Though application-specific integrated circuits (ASICs) are more efficient than modern reconfigurable hardware, such as field programmable gate arrays (FPGAs), FPGAs enable significantly faster time to market, as reconfigurability offers workload acceleration with nonrecursing engineering costs, as it maintains hardware homogeneity [2]. Modern FPGAs use volatile static random access memory (SRAM) cells to store a configuration. Logic blocks resembling lookup tables (LUTs) use that SRAM for storing logic function data. However, SRAM cells used in reprogrammable FPGAs result in a larger area and data volatility. Emerging nonvolatile memories (NVMs) can be an excellent candidate to resolve the data volatility and density issues in FPGAs. NVMs in FPGAs can act as embedded memory blocks and play a vital role in programming bits, ensuring zero boot-up delays, energy efficiency, lower footprint, and secure real-

time reconfigurability [3]. Accordingly, there is an ongoing effort in the research community to use different NVM technologies to design FPGAs [4], [5], [6], [7]. Moreover, there has been a growing interest to deploy FPGAs on edge devices [8]. However, a limited energy budget on edge devices with energy harvesting poses a major challenge to realizing a reliable operational period for IoT edge platforms. One possible solution is to enable intermittent operation [9], where the data are temporarily stored in memory during the transition from one subsystem to another and are later processed depending upon the availability of energy. Thus, nonvolatility is required to facilitate intermittent operation to maintain data integrity when the system is powered off or in a deep sleep mode, thereby making a strong case for NVM-based FPGAs for IoT at the edge.

The magnetic NVM can be integrated onto the back-end of silicon CMOS, as has been demonstrated with magnetic tunnel junctions (MTJs) and spin-transfer torque-based random access memory (STTRAM) devices. However,

ultrasmall magnets tend to lose their magnetization to thermal fluctuations, arguing in favor of solitonic excitations in magnetic films, such as domain walls or skyrmions. Of these excitations, isolated skyrmions show ballistic and tunable transport at modest current drives, which could bear advantages over magnetic domain wall devices [10].

Several concepts have been proposed using single or multiple skyrmions with precision-tuned parameters [11], [12], [13]. In this article, we propose a reconfigurable logic device based on reshuffling a single stable skyrmion. Such a conservative logic gate, with a fixed number of zeros and ones, bypasses the need for multiple energy-intensive nucleation and annihilation processes. The proposed design is also suitable for nonconservative approaches where skyrmions must be annihilated and renucleated before performing a new logic operation.

Skyrmions on racetracks typically suffer from occasional scattering events at edges, point and grain defects, or other skyrmions. These stochastic events make it especially hard to realize logic that relies on precise skyrmion synchronization or inter-skyrmion repulsion. It is possible to synchronize skyrmions with notches, but there still is thermal jitter due to diffusion and an energy cost for hold and release [14]. In contrast, our proposed design makes use of a single skyrmion on a race track, alleviating the need for positional and timing correlations involving multiple skyrmions on adjacent tracks. The contributions of this article are as follows.

- 1) We propose a novel device-to-circuit design that can be used for conventional logic-in-memory devices as well as reconfigurable logic. The novelty arises in using a single skyrmion and taking advantage of its 2-D dynamics, specifically the skyrmion Hall effect arising from a transverse Magnus force, along with magnetic anisotropy engineering along a racetrack.
- 2) We conduct a detailed deconstruction of the energetics of the entire skyrmion-based computing unit, comprising the magnetic racetracks, the clock, control signals, as well as the transistors that drive these racetracks.
- 3) We conduct a case study on the reliability of a skyrmionic logic device by varying the temperature from 280 to 350 K and show that the proposed device is stable while operating at high temperatures.
- 4) We discuss how the proposed design can be both conservative (reusing skyrmion rather than annihilation) and nonconservative in skyrmion count. We provide a detailed analysis of energy consumption for both approaches. We also identify the sources of energy consumption of a skyrmion-based circuit for different operations and show that the proposed design consumes minimum energy in most logic operations.

II. MODELING OF SKYRMIONS IN 2-D RACETRACKS

1) SKYRMION NUCLEATION

There are various methods of nucleating skyrmions; the most popular that is compatible with digital circuits is by applying current pulses through a nanocontact or through a heavy metal underlayer. The important property of the latter is the existence of point defects that act as nucleation centers in the racetrack, which do not affect the circuit operation thereafter.

TABLE 1. Material parameters for simulation (CoFeB/Pt) [15].

Parameter	Value
Gilbert damping constant (α)	0.2
Saturation magnetization (M_s)	6×10^5 A/m
Magnetic anisotropy (K)	600 kJ/m ³
Dzyaloshinskii-Moriya interaction strength (D)	2.5 mJ/m ²
Exchange stiffness (A_{ex})	15×10^{-12} J/m
Domain wall width ($\Delta = \pi D/4K$)	7.0 nm
MTJ diameter	30 nm
Bare TMR	≈ 400 %
FM width (W), length (L), thickness (t_{FM})	$(200 \times 200 \times 1)$ nm
HM thickness	1 nm
ρ_{HM} ($\Omega \cdot m$)	2.5×10^{-7}
Spin hall angle θ_{sh}	0.15
Skyrmion Hall angle	64°
Spin polarization P	0.8
Skyrmion Radius and wall width	16.5 nm [16], [17]
VCMA Coefficient	$100\text{-}200$ fJ/Vm
Nucleation MTJ resistance	$0.5\text{-}2$ k Ω
Nucleation and Annihilation time	$0.1\text{-}0.2$ ns
Minimum Nucleation current densict	8×10^{11} A/m ²
Minimum VCMA voltage	0.5 V
Driving current density	$8\text{-}15 \times 10^{10}$ A/m ²
Minimum Annihilation current density	2×10^{11} A/m ²
Speed	$80\text{-}150$ m/s

2) SKYRMION MOTION—DRIVE AND STEERING

The skyrmion is driven by the applied spin current. The motion of skyrmion is also affected by the repulsion from the edges, the engineered high anisotropy region, and thermal effects. The high anisotropy region is controlled by voltage-controlled magnetic anisotropy (VCMA) as shown and can be used for mediating skyrmion dynamics [18], [19]. In our circuit simulation, the speed of the skyrmion is taken directly from micromagnetic LLG simulations, which take into account all the aforementioned contributions to skyrmion motion. However, a simplified Thiele equation for skyrmion speed that treats it as a rigid object with minimal transverse spin precession is still useful to get an understanding of the operation time.

The main physics the Thiele approach misses is at high current densities, where the skyrmion distorts to resemble a domain wall, and its velocity saturates with increasing current density [20]. However, for modest current densities, the quasi-ballistic equation above works quite well [10]. When the VCMA is on, the high anisotropy region provides the repulsive force needed to cancel out the vertical component of skyrmion motion caused by the Magnus force, so the skyrmion will stay in its designated horizontal lane.

3) SKYRMION DETECTION

Along with heterogeneous material integration, the efficiency of electrical read is one of the key challenges dominating much of spintronics. To electrically detect skyrmions, there are two popular methods. One would be to use an MTJ stack consisting of a tunnel barrier, such as MgO, fixed between a hard magnetic layer placed by exchange bias with an antiferromagnet and a freely switching layer. Such an MTJ stack will have different tunnel magnetoresistances (TMRs) for parallel and antiparallel orientations of the two sandwiching magnets. In this case, the skyrmion sits in the free layer of the MTJ stack, with its shape factor and small

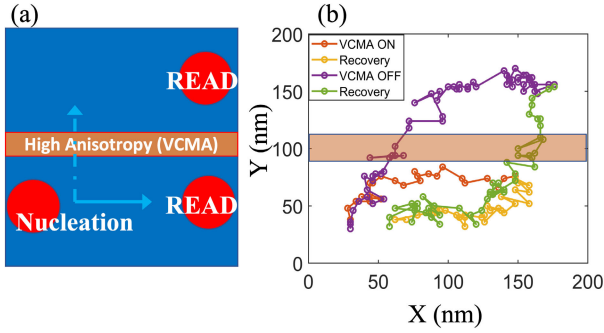


FIGURE 1. (a) Device structure from a top view. Red circles are MTJs responsible for nucleation and read. (b) Micromagnetic simulations for VCMA ON/OFF and their corresponding recovery process at room temperature. The skyrmion Hall angle during changing lanes is 64° ; on average, the propagation time for when the skyrmion changes the lane is around 30% longer than cases when the skyrmion does not change lanes.

size reducing the overall TMR. We simulate an MTJ in our circuit by considering it as a variable resistance. Another method for detecting skyrmions is using the anomalous Hall effect (AHE), where a voltage difference is generated across the magnetic layer edges perpendicular to the driving current. This voltage difference will depend on the position of the skyrmion along the racetrack and can accordingly be used for detection. In our simulation, AHE needs no extra resistance, with only the voltage swing from AHE included as an input for circuit simulations.

In our analyses, we assume that the bare TMR without the skyrmion shape factor is large (400%), which is a typical target in the spintronics community. We also assume an ultrascaled MTJ with a diameter of 30 nm to reduce the TMR dilution due to shape. However, with an effective TMR of 50%, the device would still work as demonstrated in [17]. To get this effective TMR, a minimum bare TMR of around 100% is needed.

4) SKYRMION ANNIHILATION

A high driving current can be used to annihilate skyrmions either at the boundaries or designated nonidealities in the racetrack. The minimum required current and time can be determined from either simulations or experiments. These numbers for current and time will then be inputs of the circuit. The geometric parameters we use for our simulations are listed in Table 1, along with material parameters for CoFeB/Pt taken from [15].

III. PROPOSED RECONFIGURABLE LOGIC CIRCUIT

In this section, we explain the device (see Fig. 1) and circuit (see Fig. 2) operation of the proposed reconfigurable logic element. We exploit the 2-D dynamics of skyrmions together with the CMOS control circuitry to perform basic and derived logic operations with minimal change in the biasing conditions, whereas the racetrack and MTJ stacks of the design remain unchanged. Fig. 2 presents the circuit schematics for all the logic operations and programmable logic blocks.

Though we are performing logic operations, the computed logic is also saved due to the nonvolatility of a skyrmion in

a racetrack. For any skyrmion-based devices to have proper functionality, skyrmion nucleation, propagation, detection, and annihilation must be performed successfully. We use CMOS control circuitry to function as a control element (as an ideal switch), with the skyrmion operating as a data storage element. Once a nucleated skyrmion moves to the right end of the racetrack (see Fig. 2), the logic computation is finished, and the output of the operation can be known by the skyrmion detection operation. Thus, for a specific input condition in a logic operation, the skyrmion only needs to move from one end to the other only once, and the computed logic output can be known by just detecting the skyrmion.

To nucleate the skyrmion, the supply voltage, V_{DD} , is connected to the nucleation MTJ via the transistor, T_2 , which is controlled by V_{NUC} (0.7 V). We size the transistor, T_2 , properly to allow sufficient current through the MTJ to ensure successful skyrmion nucleation. However, the nucleation process does not play a critical role in conservative logic, where we shuttle skyrmions without annihilating, so the cost of initial nucleation is amortized over the compute cycles.

The VCMA gate placed at the middle of the racetrack (Fig. 1) divides the racetrack into a top and a bottom lane through a gate tunable anisotropy barrier. During the logic operations, the racetrack is connected to the supply voltage, V_{DD} , via the transistor T_1 . The supply voltage, V_{DD} , provides the necessary currents for skyrmion propagation as long as the V_{PROP} is high. The skyrmion either moves to the top lane or stays confined to the bottom lane, depending on the VCMA gate ON–OFF state that controls the barrier and serves as a lane selection process. However, the skyrmion requires 1.33 ns to reach the top lane MTJ, whereas the skyrmion reaches the bottom lane MTJ in 1.11 ns. Since the skyrmion lane selection process is input-dependent, as explained later in this section, V_{PROP} needs to be high for the longest skyrmion propagation time, which is the delay for logic operation for the proposed design. For the case of bottom lane selection, the skyrmion will stay under the read MTJ as long as the annihilation operation (or recovery in case of conservative operation) is not applied. Meaning in the time range of 1.1–1.33 ns, the skyrmion will be under the bottom MTJ when the bottom lane is selected during a logic operation. The skyrmion annihilation process is similar to propagation, except that a higher current is applied to annihilate the skyrmion. We use the supply voltage, V_{DD} , as 1 V for skyrmion nucleation and annihilation, while 0.7 V for moving skyrmion.

The skyrmion logic is set by the competition between the transverse Magnus force that naturally moves a skyrmion from the lower to upper lane in the forward (and reverse in the backward recovery) step versus the VCMA gate that raises an inter-lane barrier and prevents that crossover. There are also thermal fluctuations that randomize the motion under ambient conditions and will need to be circumvented by ensuring the drive current and the VCMA strength are significantly larger. A typical micromagnetic simulation trace [Fig. 1(b)] shows that both the switching process (ON–OFF) during the forward drive and recovery process along the reverse drive happen reliably in the face of thermal fluctuations, using the parameters in Table 1. To increase the reliability at higher currents that will lower the operating

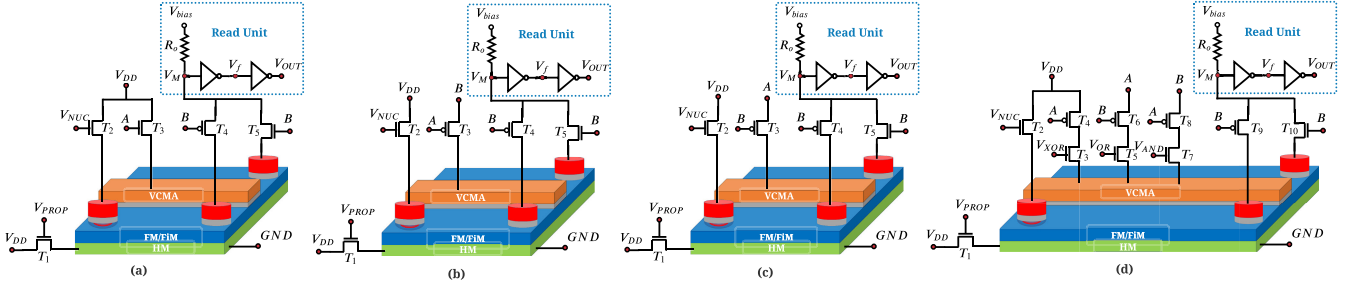


FIGURE 2. (a) XOR logic gate schematic, (b) AND logic gate schematic, (c) OR logic gate schematic, and (d) programmable logic gate schematic.

time, we need to avoid edge annihilation by raising the edges or increasing the edge anisotropy [21]. However, this would add to fabrication difficulties, which would be a trade-off that should be taken into account.

For skyrmion detection, we place two detection MTJ stacks at the ends of the top and bottom lanes of the racetrack. Either one of the top and the bottom lane MTJs is connected to the read unit during a logic operation, depending on the applied input combinations. We call this the MTJ selection process. The read unit provides the output based on the presence/absence of the skyrmion under the selected MTJ. If there is a match between the lane selection and the MTJ selection processes, the output is high; else, it is low. We size the read unit access transistors to get sufficient voltage swing at the buffer input node M , V_M . The buffer amplifies node voltage, V_M , to either high or low. We show the truth table for the basic and derived logic gates in Table 2.

We can move the skyrmion from the left-hand side to the right-hand side of the racetrack or vice versa by properly biasing the circuit. It is worth mentioning that the proposed design is suitable for both conservative and nonconservative logic with skyrmions. The skyrmion nucleation unit shown in Fig. 2 can be disregarded for the conservative approach, as we can assume a prenucleated skyrmion at the beginning of the racetrack. For the conservative approach, we move the skyrmion back to its initial position before applying different input conditions. We discuss both the conservative and nonconservative approaches in detail in Section IV.

A. XOR OPERATION

We first explain the XOR logic operation. In the case of XOR operation, one of the inputs (A) is connected directly to the gate of the transistor T_3 that controls the ON/OFF condition of the VCMA gate. The other input (B) is connected to the gate of the transistors, T_4 and T_5 , respectively. The VCMA gate will be in the ON (OFF) state when input A is 1 (0). On the other hand, the bottom (top) lane MTJ will be connected to the read unit when input B is 0 (1). When input A is high, the skyrmion will move along the bottom lane until it reaches under the read MTJ stack. If input B is low at that time, the bottom lane MTJ will be connected to the read unit, and the output will be high as the skyrmion will be under the bottom lane MTJ stack. On the other hand, if input B is high, the top lane MTJ will be connected to the read unit, and the resistance of the MTJ will be in the parallel state due to the absence of skyrmion under the MTJ. The voltage at node M , V_M will be low, ensuring 0 V at the output node, V_{OUT} .

Table 2 shows the resulting truth table. For the logic gate described and all subsequent configurations, whenever the

lane selection and MTJ selection columns agree, we get $V_{OUT} = 1$.

B. AND OPERATION

The lane selection process for AND operation differs from XOR operation, while the MTJ selection process remains unchanged. We use a pMOS transistor instead of an nMOS transistor to control the state of the VCMA gate. We connect inputs A and B to the gate and source, respectively, of the pMOS transistor T_3 . The VCMA gate will be in the ON state if the transistor, T_3 , is ON ($A = 0$), and the voltage at the source of the transistor is high ($B = 1$). Thus, the skyrmion will be in the bottom lane only for the input combination when input A is low, and input B is high. In all the other three cases, the skyrmion will be in the top lane. However, the top MTJ is connected to the read unit only when input B is high. Thus, the skyrmion will be under the selected MTJ only when both inputs A and B are high, ensuring a successful AND logic computation.

C. OR OPERATION

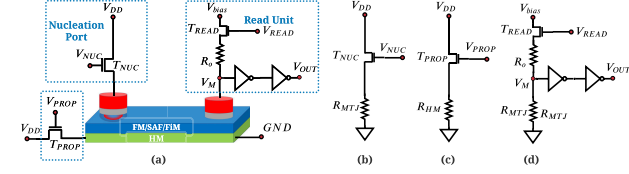
The racetrack and the control circuitry to perform OR operation are similar to the AND operation. The only difference lies in how we control the VCMA gate state. We connect inputs A and B to the source and gate of the transistor T_3 , exactly opposite to the input connections for the AND logic operation. The VCMA gate will be in the ON state if and only if input A is high, and the transistor T_3 is ON (i.e., B is 0). Thus, the skyrmion will only be confined in the bottom lane when inputs A and B are high and low, respectively, and will be in the top lane in all other conditions.

First, we consider the conditions where input B is 0. Since we keep the MTJ selection process unchanged, the bottom lane MTJ is selected whenever input B is 0. Thus, the matching (mismatching) between the lane selection and the MTJ selection process will depend on the input condition of A . If A is high (low), there will be a match (mismatch) between the lane selection and the MTJ selection process and will return a 1 (0) at the output node, V_{OUT} .

On the other hand, when input B is 1, the VCMA gate is in the OFF state irrespective of the value of A , which ensures the position of the skyrmion is in the top lane. Also, the top lane MTJ is selected when B is 1 and results in a 1 at the output node, V_{OUT} , due to the matching between the lane selection and the MTJ selection process. We can, thus, perform the logical OR operation by simply changing the input connection of the CMOS-skyrmion hybrid circuit used in the logical AND operation.

TABLE 2. Truth table of the proposed reconfigurable logic. The MTJ selection is set by input *B* for all the logic operations. The delay times used here are the maximum delay time we got from the micromagnetic simulations to ensure the device works appropriately.

A	B	VCMA			Lane Selection			MTJ Selection	V_{OUT}			Moving Energy (fJ)			Moving Delay
		AND	OR	XOR	AND	OR	XOR		AND	OR	XOR	AND	OR	XOR	
0	0	OFF	OFF	OFF	Top	Top	Top	Bottom	0	0	0	98.19	98.19	98.19	1.33 ns
0	1	ON	OFF	OFF	Bottom	Top	Top	Top	0	1	1	119.72	92.12	92.12	1.33 ns
1	0	OFF	ON	ON	Top	Bottom	Bottom	Bottom	0	1	1	98.19	125.79	125.79	1.33 ns
1	1	OFF	OFF	ON	Top	Top	Bottom	Top	1	1	0	98.19	92.12	119.72	1.33 ns

**FIGURE 3.** (a) Generic skyrmionic racetrack structure with skyrmion nucleation, propagation, and detection port, and the equivalent circuit for (b) nucleation, (c) propagation and annihilation, and (d) detection.

D. XNOR, NAND, AND NOR OPERATIONS

We can perform XNOR, NAND, and NOR operations by using the XOR, AND, and OR circuits in two different methods. One method involves initializing the MTJ fixed layer state as an antiparallel state while using the same circuit. Thus, if there is a mismatch between the MTJ selection and lane selection process, the output will be high and low otherwise.

The second method involves a slight change in the read unit circuit. If we use one inverter instead of a buffer in the read unit, the same circuit used in XOR, AND, and OR operations will perform XNOR, NAND, and NOR operations. However, the inverter needs to be sized accordingly to ensure that the node voltage, V_M , is amplified correctly.

E. PROGRAMMABLE LOGIC BLOCK

We design a programmable logic block by cascading the control circuitry for the VCMA gate state. We use the same racetrack structure, MTJ selection process, and shared read unit. We add an extra nMOS transistor in series with each of the VCMA gate control units for each logic operation. The transistors, T_3 , T_5 , and T_7 , will be ON while performing the logical XOR, AND, and OR operations [see Fig. 2(d)]. For example, in the case of XOR operation, only the transistor, T_3 , will be ON, and the other two extra transistors, T_5 and T_7 , will be OFF. Now, the equivalent circuit of reconfigurable XOR will be similar to the XOR circuit, as shown in Fig. 2(a), with an added switch. Reconfigurable OR and AND operations will also follow the same trend. We can also perform programmable XNOR, NAND, and NOR operations with the same circuit by employing any one of the two methods explained above in Section III-D. The proposed programmable logic block can be used as a two-input LUT.

IV. ENERGY CONSUMPTION IN SKYRMION-BASED DEVICES

We show a generic skyrmionic structure with skyrmion nucleation, propagation, detection, and annihilation mechanism and their respective equivalent circuits in Fig. 3(a)–(d). This generic structure can be used as a basis of skyrmion-based memory and processing-in-memory devices. The racetrack structure can be altered to accommodate notches, VCMA

gates, and so on, to design large-scale custom skyrmionic chips. However, the nucleation, propagation, detection, and annihilation mechanisms will be similar for any skyrmion-based electronic circuits. The energy consumption of each operation for a skyrmionic device can be obtained from the equivalent circuits shown in Fig. 3(b)–(d)

$$E_{\text{Nucleation}} = E_{\text{MTJ}} + E_{\text{Switching}} \quad (1)$$

$$E_{\text{Propagation/Annihilation}} = E_{\text{Racetrack}} + E_{\text{Switching}} \quad (2)$$

$$E_{\text{Detection}} = E_{\text{MTJ}} + E_{\text{Switching}} + E_{\text{Ref}} + E_{\text{Buffer}} \quad (3)$$

Using the same parameters mentioned in Table 1, we calculated the energy consumption for each operation, and the results are shown in Fig. 4. We used the nucleation MTJ resistance as 1 k Ω and V_{DD} as 1 V, the bias voltage, V_{bias} , as 0.7 V, and the reference resistance as 15 k Ω .

A. CONSERVATIVE VERSUS NONCONSERVATIVE APPROACHES

Energy and delay of the generic skyrmionic logic for conservative and nonconservative approaches are as follows:

$$E_{\text{Conservative}} = E_{\text{Propagation}} + E_{\text{Detection}} + E_{\text{Reset}} \quad (4)$$

$$D_{\text{Conservative}} = T_{\text{Propagation}} + T_{\text{Detection}} + T_{\text{Reset}} \quad (5)$$

$$E_{\text{Non-conservative}} = E_{\text{Nucleation}} + E_{\text{Propagation}} + E_{\text{Detection}} + E_{\text{Annihilation}} \quad (6)$$

$$D_{\text{Non-conservative}} = T_{\text{Nucleation}} + T_{\text{Propagation}} + T_{\text{Detection}} + T_{\text{Annihilation}} \quad (7)$$

Depending on the design, the skyrmion annihilation can be done in a single clock cycle for all the circuit elements in a cascaded circuit. As explained earlier, skyrmionic devices need to perform four different operations, and generally, the energy associated with these four different operations is different. For example, the nucleation energy is high, as a high current is needed to nucleate a skyrmion, whereas the skyrmion propagation and detection energies largely depend on the skyrmion speed, which is a function of the applied current density. In a longer racetrack, the skyrmion propagation time is high and results in higher propagation and detection energy, irrespective of the design. In the case of the conservative approach, the energy consumption of the reset operation is the same as the propagation energy, as the skyrmion is shuttled back to its initial position during the reset operation. The propagation energy is high for a lower-skyrmion speed and dominates the overall energy for both the conservative and nonconservative approaches. In fact, the higher propagation energy makes the conservative approach more energy-hungry than the nonconservative approach for low-speed skyrmions. However, high-speed skyrmions

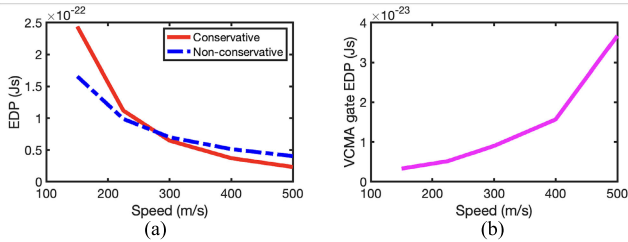


FIGURE 4. (a) EDP of a generic skyrmionic device for conservative and nonconservative approaches with one skyrmion nucleation site and one detection unit. (b) EDP of the VCMA gate for the proposed design shown in Fig. 2 (including the CMOS transistor) for different propagation speeds of skyrmions. The VCMA gate EDP will be added to the EDP of a generic skyrmionic device for the proposed design when the VCMA gate is in the ON state. We used our in-house SPICE model [22] to simulate the circuits.

consume lesser energy for propagation, thereby reducing the propagation and reset energy for the conservative approach. Conversely, for the nonconservative approach, the nucleation energy dominates the overall energy for high-speed skyrmions. Thus, the nonconservative approach consumes more energy than the conservative approach for high-speed skyrmions for generic skyrmionic devices.

The total delay in performing a logic operation is, however, always high for the conservative approach, as the skyrmion needs to be back in its initial position before a new logic operation can be performed ($2\times$ propagation). As skyrmion nucleation and annihilation time are far less than the propagation time irrespective of the skyrmion propagation speed, the energy–delay product (EDP) of the conservative logic approach is higher than that of the nonconservative approach for low-speed skyrmions. High-speed skyrmions will result in lower propagation delay, and the EDP of the conservative approach will be better if the speed of skyrmion is >280 m/s [as shown in Fig. 4(a)] for the generic racetrack structure (200×200 nm). It is worth noting that we considered one skyrmion nucleation site. For devices where multiple skyrmions need to be nucleated and, multiple detection units are active for a single logic operation, the conservative approach will have better EDP for low-speed skyrmions as well (once again, details are design-dependent).

V. RELIABILITY OF SKYRMION-BASED LOGIC DEVICE: A CASE STUDY

We show temperature-dependent simulations for the proposed device and alternate devices in Figs. 5 and 6, respectively. In alternate devices that rely on skyrmion–skyrmion repulsion, schematically shown in Fig. 6, it is necessary for skyrmions to be confined in a relatively small region. Since thermalized random scattering is nontopological, this will affect skyrmion movement, especially near the edges, as the skyrmion needs to keep its robustness to repel from them. This would make the skyrmion susceptible to annihilation at the edges or not function as intended at room or higher temperatures. As seen from Fig. 6, in a generic skyrmion–skyrmion-based device at room temperature, skyrmions can be annihilated at the edges or notches (the corresponding room temperature paths shown in

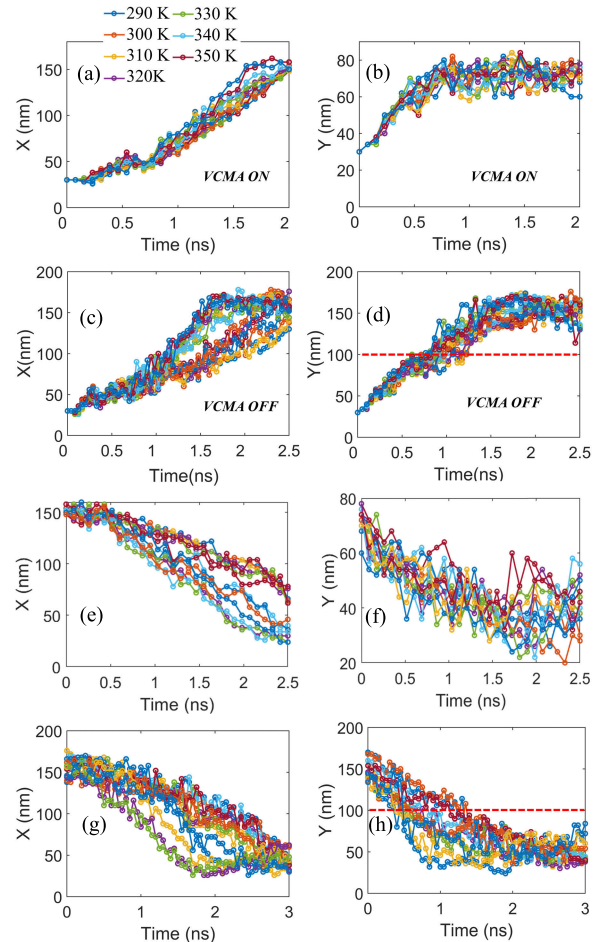


FIGURE 5. X, Y position versus time for temperatures from 290 to 350 K. Here, (e)–(h) are recovery operations for (a)–(d), respectively. (a) and (b) VCMA is ON, and as we can see, the skyrmion does not pass the VCMA (high anisotropy region). (c) and (d) VCMA is OFF, and skyrmion moves to the top lane. The red dashed horizontal lines show the location of the VCMA gate, which separates the two lanes.

red, blue, and green occasionally end prematurely before they reach their desired targets). This can be alleviated to some degree by engineering the edges to have higher anisotropy or thickness. But, as with all the other “billiard-ball” logic, the arrival times of the balls (in this case, skyrmions) must be exact to be reliable; otherwise, it will lead to significant error.

The material parameters chosen here are based on experimental results of CoFeB/Pt and MTJ heterostructures [15]. The secondary parameters (nucleation, annihilation time, and current) are all based on room-temperature micro-magnetic simulations. Note that these are not necessarily the ideal theoretical parameters. To make a one-to-one comparison, we will show at the end a comparison with the existing devices proposed in the literature using the same parameters.

In order to use a magnetic racetrack as part of a general digital circuit, it has to be compatible with fabrication processes. The racetrack geometry we propose here does not need any fine-tuned notch or barrier. The VCMA gate

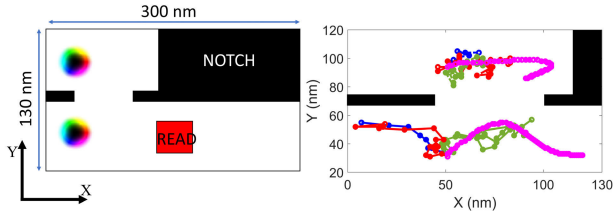


FIGURE 6. Room versus zero temperature simulations for a two-skyrmion-based logic device. The top figure shows the starting position of skyrmions and the geometry. The simulation is for 1,1 cases. The purple lines are for 0 temperature, while the red, blue, and green colored lines are for room temperature simulations. We can see that only at zero temperature, the bottom skyrmion reliably reaches the designated read location; for room temperature simulations, the bottom skyrmion does not reach the read location.

has already been experimentally demonstrated [23]. The geometry of the VCMA gate is also assumed to be simple, with no need for fine-tuning.

We ran temperature-dependent simulations for the proposed device in the temperature ranges of 280–350 K. As seen from Fig. 5, the proposed device works consistently even above room temperatures. As the skyrmion, in our case, is not confined to a small area, it has a lower chance of getting pinned or annihilated by the edges or boundaries. The reason for fluctuations between Fig. 5(b) and (f) is the thermal effects make the skyrmion lose some of its rigidity and its circular shape, which leads to more random movements, and the deformation makes it slower in the recovery operation. For Fig. 5(b) and (f), the skyrmion is limited to one lane, and the edge effects make it less rigid compared with the other cases. In each of the 20 temperature-dependent simulation runs, the skyrmion position at the end of every operation was precise as intended, indicating a very small error rate. In comparison, as seen in Fig. 6, a device based on skyrmion–skyrmion repulsion shows a significant error, and for about 50% of the temperature simulations, the device did not work as intended.

To quantify the success rate of the devices and make a meaningful comparison, we define a successful operation as when the skyrmion reaches the desired location to have the correct read. As seen in Fig. 7, the nonconservative approach in our design has higher reliability, followed closely by the conservative approach. The reason for a decrease in the reliability of the conservative methods is the longer back-and-forth travel time of the skyrmion, which increases the probability of getting annihilated at the edges. For the two-skyrmion logic simulations, we used the same structure as Fig. 6. Due to the confinement of skyrmions and the requirement of fine-tuned skyrmion–skyrmion repulsion, the success rate of the two-skyrmion approach is much lower.

VI. DISCUSSIONS

The proposed circuit is designed in such a way that changing the bias connections of the VCMA gate control switch allows us to perform all the basic and derived logic operations. Moreover, it is reconfigurable, with only three extra switches needed to ensure reconfigurability without altering the racetrack geometry. Similar to a conventional CMOS logic, the circuit can be cascaded to implement any function

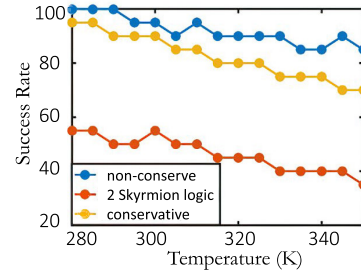


FIGURE 7. Success rate for three different skyrmion-based logic devices. The two skyrmion logic is based on the geometry shown in Fig. 6. The success rate is defined as a successful read of skyrmion based on the desired operation. The success rate was below 20% for a conservative logic design with two skyrmions.

with reconfigurable capability. Our design supports both conservative and nonconservative logic designs. In the case of a conservative approach, skyrmions do not need to nucleate before performing a new logic operation, although we may need to replenish the device or renucleate the skyrmion after several cycles depending on the expected skyrmion lifetime. The cost of occasional renucleation, however, is amortized over several compute cycles.

The minimum energy consumption of a skyrmionic device for both conservative and nonconservative approaches can be obtained from (4) and (6), respectively. The details will depend on the design. For example, having multiple nucleations and detection ports, and VCMA gates will result in extra switching energy. The minimum energy consumption of a skyrmionic circuit will also depend on the material parameters and the technology node of the switching transistor. However, this latter variation in energy is independent of design. Further scaling down of the energy consumption will require better material engineering and the advent of new technology nodes for the switching transistor. Next, we argue that our proposed design consumes minimum energy based on the assumption mentioned in Section IV.

In the case of the proposed design, only one read port is active among the two read ports placed on two lanes during a logic operation. We also use an additional switch to control the VCMA gate. However, the switch is only ON for two [(1,0) and (1,1)] of the four input conditions in the case of XOR operation and one of the four input conditions for AND (0,1) and OR (1,0) operations, respectively. Apart from these four conditions, the proposed circuit will consume minimum energy for the remaining input conditions. The reconfigurable logic operations will also follow the same trend with an additional switching transistor energy. We show the EDP of the VCMA gate branch at different skyrmion speeds in Fig. 4(b). Moreover, a smaller racetrack would consume less energy due to the shorter propagation time compared with a longer racetrack for a fixed applied current. The limit on racetrack dimensions is set by the skyrmion size, which, in turn, is determined by the material parameters. As we use a small-sized racetrack, we argue that the proposed design consumes minimum energy in most cases for all the basic and derived logic gates.

As explained in Section V, our design is stable at different temperatures, which is really important in modern

TABLE 3. Comparison of transistor count and MTJ count among conventional LUT, MRAM-based LUTs, and proposed LUT considering two-input LUTs.

LUT Circuit	# of CMOS transistors	# of MTJs
Proposed LUT	14	3
Conventional LUT	32	0
MLUT in [25]	23	8
MLUT in [26]	19	8
MLUT in [27]	47	8
MLUT in [28]	40	8

high-performance computing systems, as the temperature is likely to get high while running computationally heavy workloads. To the best of our knowledge, this is the first work to suggest thermally stable operability of the skyrmionic logic circuits. Thus, we believe that a simpler racetrack structure that provides flexibility in skyrmion speeds will be more reliable. We argue that our design is robust, as the output only depends on the position of the skyrmion after a certain propagation delay and is not sensitive to any fine-tunings in the racetrack geometry. In comparison, other proposed reconfigurable skyrmionic devices that rely on skyrmion–skyrmion repulsion and engineering notches or barriers to manipulate skyrmion movement [11], [24] suffer significantly from thermal effects and (unavoidable) nonidealities of the racetrack, as the logic operation in such devices depends on the exact arrival times of skyrmions at exact positions. Moreover, the proposed design is energetically efficient, as it consumes minimum energy in most cases, as explained earlier. We achieve $30\times$ EDP improvement over the reconfigurable design proposed in [11]. It should be noted that we only compared the racetrack energy consumption and delay in this comparison, as peripheral energy is not reported in [11].

We also compare the proposed design that can be used as a two-input LUT with conventional LUT, and MRAM-based LUTs (MLUTs) [25], [26], [27], [28] list the transistor and MTJ counts in Table 3. We chose to compare LUTs, as LUTs are one of the fundamental building blocks of FPGA. An n -input conventional LUT that can implement n input Boolean function requires 2^n SRAM cells and an $n:1$ mux. Thus, for a two-input LUT, there would be four SRAM cells (16 transistors) and a 4:1 mux (eight transistors considering pass transistor-based mux).

The proposed 2-input LUT does not require extra multiplexers, such as conventional two-input LUT, and only requires 14 transistors (including the read unit) and a magnetic racetrack. The reason behind this is that the inputs are directly applied to the programmable logic block, and the output is a direct function of the applied inputs. However, it should be noted that the read unit can be shared among the programmable logic blocks similar to the design proposed in [29] and [30]. The area efficiency of the proposed LUT compared to the conventional LUT will largely depend on the CMOS technology node, as the proposed LUT area is dominated by the transistors.

Apart from the area efficiency, one major benefit for the proposed LUT is the nonvolatility. Since the proposed LUT is nonvolatile, the skyrmion will propagate only during the configuration time. The saved computed data can be

read during the FPGA runtime once the LUT is programmed. Unless the inputs are changed, only the read unit needs to be connected to the LUT to read the output of the LUT during the FPGA runtime. The read time is only 30 ps and consumes only 1.015 and 0.515 fJ for reading 0 and 1, respectively, each time. Furthermore, the data can be read as many times as needed with zero standby power. However, this is not the case for SRAM-based conventional LUT due to its volatility. If the target application needs to run for 10 min, there will be standby power consumption during this entire time. The exact energy efficiency of the proposed design over the conventional LUT will depend on the runtime and how many times the data need to be read from the SRAM cells. Moreover, conventional LUTs use 6T SRAM cells, which suffer from higher leakage, thereby consuming higher static power.

From Table 3, we can clearly see that the proposed LUT requires fewer transistors compared with MLUTs proposed in [25], [26], [27], and [28] and also requires eight MTJs. These MTJs will also incur a large area overhead, as a separation of around 90 nm between MTJ units is needed [31] to have a reliable working MTJ array in STT-MRAM. Thus, the use of eight MTJs will incur additional area overhead for MLUTs [25], [26], [27], [28]. Though the proposed design uses three MTJs, these MTJs are placed on top of the magnetic racetrack and do not add additional area overhead other than the overhead of the magnetic racetrack.

Since MLUTs are also nonvolatile, the runtime energy consumption for proposed LUT and MLUTs will depend on the data read energy. The read time for the proposed LUT is approximately $7.66\times$ faster compared with the lowest MLUT read time reported in [25], [26], [27], and [28] and yields an order of magnitude improvement in read energy consumption over MLUTs proposed in [25], [26], [27], and [28]. The reason behind this faster read operation is the design of a simplistic read circuit adopted in the proposed design [17].

VII. CONCLUSION

The proposed reconfigurable logic architecture is flexible, as we can make use of the same circuit by tuning the control circuitry. We argue that the proposed architecture is more reliable and comparatively easy to fabricate, as we are not changing the racetrack structure. Our proposed design not only takes advantage of the 2-D dynamics of magnetic skyrmion but also takes advantage of the peripheral control circuitry. Apart from the 2-D movement of magnetic skyrmion, our approach does not rely on the phenomena, such as synchronization of the skyrmion, skyrmion–skyrmion repulsion, and so on, which makes the device less reliable. We believe the proposed design can be adopted for applications, such as binary neural networks and other logic-in-memory applications, as well as it can be a building block of nonvolatile reconfigurable hardware fabric for future computing systems.

ACKNOWLEDGMENT

The authors would like to thank Md Golam Morshed for his fruitful discussions.

REFERENCES

- [1] E. Nurvitadhi, D. Sheffield, J. Sim, A. Mishra, G. Venkatesh, and D. Marr, "Accelerating binarized neural networks: Comparison of FPGA, CPU, GPU, and ASIC," in *Proc. Int. Conf. Field-Program. Technol. (FPT)*, Dec. 2016, pp. 77–84.
- [2] H. Parvez and H. Mehrez, *Application-Specific Mesh-Based Heterogeneous FPGA Architectures*. Springer, 2010.
- [3] Y. Chen, J. Zhao, and Y. Xie, "3D-NonFAR: Three-dimensional non-volatile FPGA architecture using phase change memory," in *Proc. 16th ACM/IEEE Int. Symp. Low Power Electron. Design*, Aug. 2010, pp. 55–60.
- [4] S. Paul, S. Mukhopadhyay, and S. Bhunia, "Hybrid CMOS-STTRAM non-volatile FPGA: Design challenges and optimization approaches," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2008, pp. 589–592.
- [5] K. Liu, M. Zhao, L. Ju, Z. Jia, J. Hu, and C. J. Xue, "Applying multiple level cell to non-volatile FPGAs," *ACM Trans. Embedded Comput. Syst.*, vol. 19, no. 4, pp. 1–22, Jul. 2020.
- [6] L. Bagheriye, S. Toofan, R. Saeidi, B. Zeinali, and F. Moradi, "A reduced store/restore energy MRAM-based SRAM cell for a non-volatile dynamically reconfigurable FPGA," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 11, pp. 1708–1712, Nov. 2017.
- [7] S. M. Williams and M. Lin, "Architecture and circuit design of an all-spintronic FPGA," in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays*, Feb. 2018, pp. 41–50.
- [8] S. Biookaghazadeh, M. Zhao, and F. Ren, "Are FPGAs suitable for edge computing?" in *Proc. USENIX Workshop Hot Topics Edge Comput. (HotEdge)*, 2018, pp. 1–6.
- [9] B. Chatterjee, N. Cao, A. Raychowdhury, and S. Sen, "Context-aware intelligence in resource-constrained IoT nodes: Opportunities and challenges," *IEEE Des. Test*, vol. 36, no. 2, pp. 7–40, Apr. 2019.
- [10] H. Vakili et al., "Skyrmionics—Computing and memory technologies based on topological excitations in magnets," *J. Appl. Phys.*, vol. 130, no. 7, Aug. 2021, Art. no. 070908.
- [11] S. Luo et al., "Reconfigurable skyrmion logic gates," *Nano Lett.*, vol. 18, no. 2, pp. 1180–1184, Feb. 2018.
- [12] M. N. Sakib, H. Vakili, S. Ganguly, S. Mosanu, A. W. Ghosh, and M. Stan, "Magnetic skyrmion-based programmable hardware," in *Proc. SPIE*, vol. 11470, 2020, pp. 129–140.
- [13] M. Chauwin et al., "Skyrmion logic system for large-scale reversible computation," *Phys. Rev. A, Gen. Phys.*, vol. 12, no. 6, Dec. 2019, Art. no. 064053.
- [14] M. G. Morshed, H. Vakili, and A. W. Ghosh, "Positional stability of skyrmions in a racetrack memory with notched geometry," *Phys. Rev. A, Gen. Phys.*, vol. 17, no. 6, Jun. 2022, Art. no. 064019.
- [15] M. Xu et al., "Voltage-controlled antiferromagnetism in magnetic tunnel junctions," *Phys. Rev. Lett.*, vol. 124, no. 18, May 2020, Art. no. 187701.
- [16] X. S. Wang, H. Y. Yuan, and X. R. Wang, "A theory on skyrmion size," *Commun. Phys.*, vol. 1, no. 1, p. 31, Jul. 2018.
- [17] H. Vakili et al., "Temporal memory with magnetic racetracks," *IEEE J. Explor. Solid-State Comput. Devices Circuits*, vol. 6, pp. 107–115, 2020.
- [18] Z. Zhang et al., "Ultra-low-power reservoir computing based on synthetic antiferromagnetic skyrmion pairs," *IEEE Electron Device Lett.*, vol. 43, no. 9, pp. 1567–1570, Sep. 2022.
- [19] X. Wang, W. Gan, J. C. Martinez, F. N. Tan, M. B. A. Jalil, and W. S. Lew, "Efficient skyrmion transport mediated by a voltage controlled magnetic anisotropy gradient," *Nanoscale*, vol. 10, no. 2, pp. 733–740, 2018.
- [20] K. Litzius et al., "The role of temperature and drive current in skyrmion dynamics," *Nature Electron.*, vol. 3, no. 1, pp. 30–36, Jan. 2020.
- [21] H. T. Fook, W. L. Gan, I. Purnama, and W. S. Lew, "Mitigation of Magnus force in current-induced skyrmion dynamics," *IEEE Trans. Magn.*, vol. 51, no. 11, pp. 1–4, Nov. 2015.
- [22] M. N. Sakib, H. Vakili, S. Ganguly, M. Stan, A. W. Ghosh, and C. L. Brown, "Compact SPICE model of topological textures on magnetic racetracks for design space exploration," in *Proc. Int. Conf. Simul. Semiconductor Processes Devices (SISPAD)*, Sep. 2021, pp. 276–279.
- [23] B. Rana and Y. Otani, "Towards magnonic devices based on voltage-controlled magnetic anisotropy," *Commun. Phys.*, vol. 2, no. 1, pp. 1–12, Aug. 2019.
- [24] Z. Z. Zhang et al., "Skyrmion-based ultra-low power electric-field-controlled reconfigurable (SUPER) logic gate," *IEEE Electron Device Lett.*, vol. 40, no. 12, pp. 1984–1987, Dec. 2019.
- [25] R. Rajaei, "Radiation-hardened design of nonvolatile MRAM-based FPGA," *IEEE Trans. Magn.*, vol. 52, no. 10, pp. 1–10, Oct. 2016.
- [26] D. Suzuki, M. Natsui, T. Endoh, H. Ohno, and T. Hanyu, "Six-input lookup table circuit with 62% fewer transistors using nonvolatile logic-in-memory architecture with series/parallel-connected magnetic tunnel junctions," *J. Appl. Phys.*, vol. 111, no. 7, Apr. 2012, Art. no. 07E318.
- [27] Y. Lakys, W. S. Zhao, J.-O. Klein, and C. Chappert, "Hardening techniques for MRAM-based nonvolatile latches and logic," *IEEE Trans. Nucl. Sci.*, vol. 59, no. 4, pp. 1136–1141, Aug. 2012.
- [28] Y. Guilleminet, L. Torres, and G. Sassatelli, "Non-volatile run-time field-programmable gate arrays structures using thermally assisted switching magnetic random access memories," *IET Comput. Dig. Techn.*, vol. 4, no. 3, pp. 211–226, May 2010.
- [29] M. N. Sakib, R. Sreekumar, X. Zhu, T. Tracy, and M. R. Stan, "Processing-in-memory with temporal encoding," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Jul. 2022, pp. 56–61.
- [30] M. N. Sakib, R. Sreekumar, V. Verma, T. Tracy, and M. R. Stan, "ATCPiM: Analog to time coded processing in memory for IoT at the edge," in *Proc. IEEE 7th World Forum Internet Things (WF-IoT)*, Jun. 2021, pp. 704–709.
- [31] K. Rho et al., "A 4Gb LPDDR2 STT-MRAM with compact 9F2 1T1MTJ cell and hierarchical bitline architecture," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 396–397.

• • •