A Comparison Study of Spin-Transfer Torque- and Spin-Orbit Torque-Based Stochastic Computing Using Computational Random Access Memory (SC-CRAM)

Brandon R. Zink, Marc D. Riedel, Ulya R. Karpuzcu, and Jian-Ping Wang, *Fellow, IEEE*

Department of Electrical and Computer Engineering, University of Minnesota Twin Cities, Minneapolis, MN 55455 USA

Stochastic computing (SC) is a computing method that benefits from high noise resiliency and can perform complex arithmetic functions with a small number of logic gates, thus making it a promising solution for next generation neuromorphic systems. However, generating random bit-streams using CMOS-based technologies requires a large number of transistors, thus drastically increasing the total computation delay and energy consumption. In our previous work, we presented a solution where stochastic computation and stochastic bit generation were embedded within the same memory cell using a spintronic-based in-memory computing architecture called computational random access memory (CRAM), which we called SC-CRAM. In this work, we evaluate and compare the performance of SC-CRAM between spin-transfer torque (STT) and spin-orbit torque (SOT) switching mechanisms using key parameters of magnetic tunnel junctions (MTJs) from the research laboratories, the current industry standards, and the project performance metrics. Our calculations showed that, based on current trends, the performance of SC-CRAM can be further optimized by utilizing the SOT switching mechanism when the tunneling magnetoresistance (TMR) ratio of the MTJ pillars increases and the resistance-area (RA) product of the pillars is minimized. SC-CRAM benefits from high noise resilience and small array sizes, and our results demonstrate that its performance metrics can be enhanced.

***Index Terms*— In-memory computing, magnetic tunnel junctions (MTJs), spin-orbit torque (SOT), spin-transfer torque (STT), stochastic computing (SC).**

I. Introduction

HARDWARE realization of biologically plausible neuromorphic computing algorithms is very difficult, and in some cases impossible, on modern computing platforms. This is due to the large number of transistors required to perform neuronal and synaptic tasks as well as the large latencies and energy dissipation caused by the Von Neumann bottleneck [1]. There are certain neuromorphic tasks, such as recognition, classification, and prediction, that do not require high mathematical precision, but rather, require extracting key information and approximations from large data sets. Therefore, recent studies have examined beyond-CMOS technologies and information encoding schemes as alternatives to the current CMOS-based computing platforms for various neuromorphic applications [2], [3], [4]. In particular, various probabilistic computing schemes have been attractive solutions for these types of tasks since these methods are highly resilient to noise and small variations in the input data [5]. Spintronics is a beyond-CMOS technology [6], [7] that are promising solutions for probabilistic computing schemes and have been proposed as random number generators [8], [9], [10], [11].

One particular probabilistic computing scheme that has shown to be promising for neuromorphic computing applications is stochastic computing (SC) [12]. In SC, each real-valued number x ($0 \leq x \leq 1$) is represented by a sequence of random bits, each of which has probability x of being one and probability $1 - x$ of being zero. These bits can either be serial streaming on a single wire or in parallel on a bundle of wires. When serially streaming, the signals are probabilistic in time; when in parallel, they are probabilistic in space. Consider the operation of multiplication implemented in SC. It consists of but a single AND gate. The inputs are two independent input stochastic bit-streams A and B. The number represented by the output stochastic bit-stream C is as follows:

$$\begin{aligned} c &= P(C = 1) = P(A = 1 \text{ and } B = 1) \\ &= P(A = 1)P(B = 1) = a \cdot b. \end{aligned} \tag{1}$$

The probability of getting a one at the output, $P(C = 1)$, is equal to the probability of simultaneously getting ones at the inputs, namely, $P(A = 1)$ times $P(B = 1)$. So the AND gate multiplies the two values represented by the stochastic bit-streams. Consider the operation of addition implemented in SC. It is not feasible to add two probability values directly; this could result in a value greater than one, which cannot be represented as a probability value. However, we can perform scaled addition with a multiplexer (MUX), a digital construct that selects one of its two input values to be the output value, based on a third "select" input value. Call the select input S and the two data inputs A and B. When $S = 1$, the output $C = A$. Otherwise, when $S = 0$, the output $C = B$. With the assumption that the three input stochastic bit-streams A, B, and S are independent, the number represented by the output stochastic bit-stream C is as follows:

$$c = P(C = 1)$$

Manuscript received 7 August 2023; revised 5 September 2023; accepted 3 October 2023. Date of publication 19 October 2023; date of current version 25 April 2024. Corresponding authors: B. R. Zink and J. P. Wang (e-mail: zinkx030@umn.edu; jpwang@umn.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMAG.2023.3326076.

Digital Object Identifier 10.1109/TMAG.2023.3326076

$$\begin{aligned} &= P(S = 1 \text{ and } A = 1) + P(S = 1 \text{ and } B = 1) \\ &= P(S = 1)P(A = 1) + P(S = 0)P(B = 1) \\ &= s \cdot a + (1 - s) \cdot b. \end{aligned} \tag{2}$$

SC can not only perform arithmetic functions with a small number of gates, but it is also very noise resilient [17], [18], [19]. Some studies have shown that it can perform image filtering tasks even under conditions with 30% bit flips [17]. One of the key disadvantages of SC using CMOS-based random number generators is the large number of transistors required to generate high-quality random numbers. In some cases, random number generation can account for 80% of the total circuit area and total energy consumption [17]. One possible solution is to replace CMOS-based random number generators with spintronic-based ones. Previous studies have exploited the intrinsic stochasticity of magnetic tunnel junctions (MTJs) in order to generate stochastic bit-streams using a single nanodevice [20], [21] rather than using a large number of transistors required for CMOS-based random number generators. While these solutions significantly reduce the area cost of generating stochastic bits, external circuitry and additional computation steps are still required.

To avoid this short-coming, SC was implemented using the computational random access memory (CRAM) architecture, which is a spintronics-based memory platform capable of performing logic operation directly within the memory cell [22], [23]. CRAM was first proposed in [22] and [23] and demonstrated based on the STT-RAM array recently [24]. Performing SC operations via CRAM using a process called SC-CRAM has two key benefits over SC using CMOS-based technologies. One is that the MTJ, which is a key component of the CRAM cell, has intrinsic stochasticity, which is an advantage for any MTJ-based stochastic bit-generator. The second is that stochastic computation and random bit generation can be done within the same memory cell in CRAM, thus eliminating the additional area cost and computation delay needed for CMOS-based SC. In our previous work, we demonstrated that SC-CRAM outperformed conventional computing in CRAM in local image thresholding, Bayesian inference, Bayesian belief network (BBN), and kernel density estimation (KDE) in terms of computation delay, total circuit area, and noise margin [25].

While our previous results in SC-CRAM were promising, the overall performance could be improved. In particular, the energy consumption of SC-CRAM was higher than in conventional CRAM for all four neuromorphic applications examined. In this study, we investigate how the intrinsic properties of the MTJs affect the total energy consumption in SC-CRAM for various arithmetic functions and neuromorphic applications. Additionally, we compare the performance of SC-CRAM between the spin-transfer torque (STT) and spin-orbit torque (SOT) switching mechanisms. STT-RAM [26] has been developed for decades and now commercially available from industry [27], [28], [29], [30]. SOT-RAM has been proposed and developed in past ten years [31]. Promising aspects of SOT-RAM over STT-RAM have been presented by many research groups, for example, speed, endurance, reliability, and potential ultralow writing energy [32], [33], [34], [35]. Our calculations use key performance metrics from the experimental results from research laboratories, the current industry standards, and metrics for projected future MTJ devices for both STT and SOT switching. While this work focuses on the usage of MTJs for SC-CRAM with two different switching mechanisms, it should be noted that MTJs can be also tuned into the building blocks for probabilistic computing to address optimization problems [36], [37], [38], [39]. This article is organized as follows. The background information on the CRAM architecture, the working principles of SC-CRAM, and the four neuromorphic computing applications tested are explained in Section II. The calculations and benchmarking methods are explained in Section III. The results of our calculations are presented in Section IV and an analysis of the results obtained is discussed in Section V. The article is then summarized in Section VI.

II. Background

A. SOT Versus STT-CRAM

Diagram for an STT- and SOT-CRAM cell is shown in Fig. 1(a) and (c), respectively. The configuration of these two cells have several similarities. Both use a two-transistor/one MTJ structure, which allows for separate logic and memory paths. This enables true in-memory computing capability in CRAM, where memory read and write operations are performed using the memory word line (WL) and logic operations are performed within the CRAM cell by enabling the logic bitline (LBL). The difference between these two cells is that the write operations are done via the STT mechanism in Fig. 1(a) whereas the write operations are done via the SOT mechanism in Fig. 1(c).

During the logic operation, LBL is high, which allows for CRAM cells in the same row to be connected electrically through the logic line (LL). The examples illustrated in Fig. 1(b) and (d) show an array of three cells in STT-CRAM and SOT-CRAM, respectively. In both cases, two cells operate as input cells and one operates as the output cell. Voltage pulses (V_{BSL}) are applied to the bit select lines (BSLs) of the input cells, which generates a current through the output MTJ (I_{OUT}), which is dependent on the resistance states of the input MTJs. The output MTJ will switch states if the voltage at the output MTJ exceeds the critical switching voltage (V_C) of the output MTJ. For STT-CRAM, the voltage of the output MTJ is $I_{\mathrm{OUT}} * R_{\mathrm{MTJ}}$, where R_{MTJ} is the resistance of the output MTJ and switching is done via STT. For SOT-CRAM, the I_{OUT} is applied to the spin Hall channel of the output cell, and the output voltage is $I_{\mathrm{OUT}} * R_{\mathrm{SHE}}$, where R_{SHE} is the resistance of the spin Hall channel. The criteria for V_{BSL} and the initial state of the output MTJ (output preset state) for various logic operations are shown in Table I.

B. Working Principles of SC-CRAM

The process for implementing SC in CRAM is called SC-CRAM. This process is unique to other SC schemes as it allows for stochastic bit generation and computation to be performed directly within the same CRAM cells, therefore,

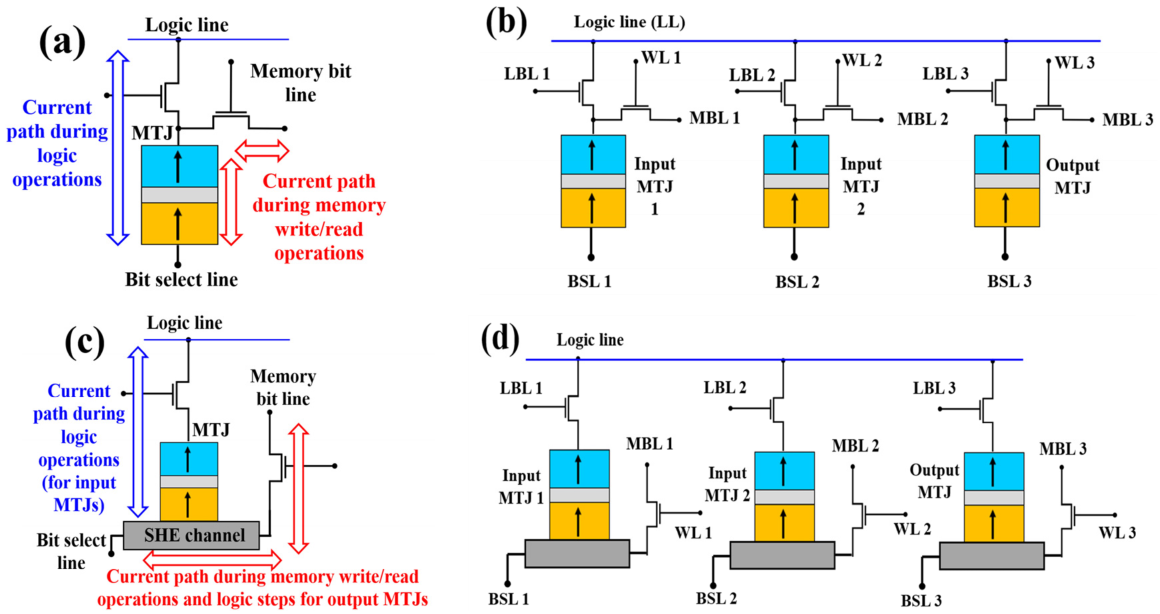

Fig. 1. Diagram for (a) STT-CRAM cell, (b) STT-CRAM array, (c) SOT-CRAM cell, and (d) SOT-CRAM array.

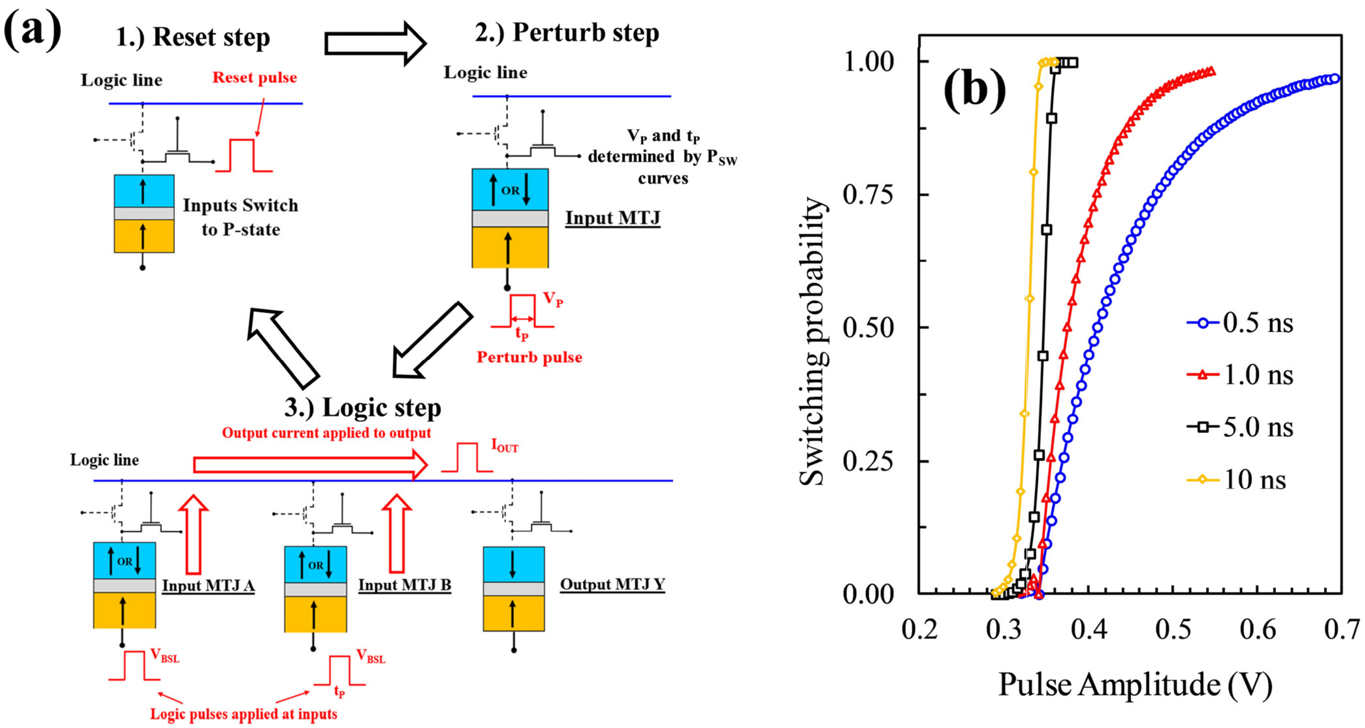

Fig. 2. (a) Illustration of task scheduling for the SC-CRAM process (read step not shown). (b) Example of switching probability distribution curves for projected STT MTJs at various pulse widths. Note that the switching probability distribution data is used to determine the proper pulse amplitude for the perturb step.

external circuitry is not required for random number generation. The full process of SC-CRAM consists of synchronized reset, perturb, logic, and read cycles.

To illustrate how multiplication of two bit-streams, *A* and *B*, is done using AND logic in SC-CRAM, consider the example shown in Fig. 2(a). This figure outlines all the four cycles of SC-CRAM. During the reset step, voltage pulses, V_{RES}, are applied along the memory bit-lines of each cell, which initializes the input MTJs to the P-state and the output MTJ, Y, to the AP-state. Note that the initial state of the MTJ depends on the function of the cell. Any cells that undergo the perturb step will be initialized to the P-state whereas any cells that serve as output cells during logic operations will be set to the proper preset state for the desired logic function (recall Table I). Since the example in Fig. 2(a) is performing AND logic, the output MTJ is set to the AP-state.

TABLE I

CRAM Logic Criteria

Logic operation	Preset output state	STT voltage criteria (V_B)	SOT voltage criteria (V_B)
BUFFER	AP	$V_C^{ST}\left[\frac{R_O+R_P}{R_O}\right] < V_B < V_C^{ST}\left[\frac{R_O+R_{AP}}{R_O}\right]$	$V_C^{SH}\left[\frac{R_{SHE}+R_P}{R_{SHE}}\right] < V_B < V_C^{ST}\left[\frac{R_{SHE}+R_{AP}}{R_{SHE}}\right]$
NOT	P		
AND	AP	$V_C^{ST}\left[\frac{R_O(R_{AP}+R_P)}{R_{AP}R_P+R_O(R_{AP}+R_P)}\right] < V_B < V_C^{ST}\left[\frac{2R_OR_{AP}}{R_{AP}^2+2R_OR_{AP}}\right]$	$V_C^{SH}\left[\frac{R_{SHE}(R_{AP}+R_P)}{R_{AP}R_P+R_{SHE}(R_{AP}+R_P)}\right] < V_B < V_C^{SH}\left[\frac{2R_{SHE}R_{AP}}{R_{AP}^2+2R_{SHE}R_{AP}}\right]$
NAND	P		
OR	AP	$V_C^{ST}\left[\frac{2R_OR_P}{R_P^2+2R_OR_P}\right] < V_B < V_C^{ST}\left[\frac{R_O(R_{AP}+R_P)}{R_{AP}R_P+R_O(R_{AP}+R_P)}\right]$	$V_C^{SH}\left[\frac{2R_{SHE}R_P}{R_P^2+2R_{SHE}R_P}\right] < V_B < V_C^{SH}\left[\frac{R_{SHE}(R_{AP}+R_P)}{R_{AP}R_P+R_{SHE}(R_{AP}+R_P)}\right]$
NOR	P		

$R_{AP(P)}$ = MTJ resistance in the AP(P) state. R_O = MTJ resistance of preset state of output MTJ. R_{SHE} = resistance of the spin Hall channel. V_C^{ST} = STT critical switching voltage. V_C^{SH} = SOT critical switching voltage.

During the perturb step, voltage pulses $V_{P(A)}$ and $V_{P(B)}$ are applied along the memory bit lines of input cells A and B. Unlike the reset and logic steps, the voltage pulses during the perturb step cause the MTJs to switch probabilistically rather than deterministically. The amplitude and duration of $V_{P(A)}$ and $V_{P(B)}$ determine the switching probability of A and B, which can be obtained through switching probability distribution data from the MTJs, an example of which is shown in Fig. 2(b). There are three ways of determining the desired switching probability of A and B. One is through machine learning algorithms, however, this is only applicable in neural networks where the input cell being used sets a synaptic weight. The second is when performing functions involving a fixed constant, in which case, the switching probability needs to be equal to the desired value of the constant. The third method of determining the desired switching probabilities is when these values are dependent on the input data. For example, in image processing applications, the switching probability of A would be dependent on the intensity of the corresponding pixel.

The logic step in SC-CRAM follows the same criteria that is listed in Table I. Since the example shown in Fig. 2(a) is illustrating AND logic, V_B should be set so that Y switches to the P-state when A, B, or both are in the P-state, but Y should remain in the AP-state if both A and B are in the AP-state. Note that the resistance state of A and B are probabilistic, however, switching during the logic step is still deterministic.

During the read step, the final state of Y is measured using a voltage pulse V_R, which should be small enough so that it does not affect the resistance state of Y. It should be noted that some functions require multiple logic steps. For these functions, the read step is replaced by the proceeding logic operation. In this case, the output cell for the first logic function serves as the input cell for the second logic function. The read step for each function only needs to be performed at the output of the final logic operation.

C. Arithmetic Functions in SC-CRAM

Fig. 2(a) outlines the process for computing the multiplication of bit-streams A and B using AND logic. However, SC-CRAM is capable of performing a wide variety of arithmetic functions. In this study, we will investigate the performance of five additional functions, which are outlined in Table II. These functions are scaled addition, scaled division, absolute valued subtraction, square root, and exponential. Note that these functions were chosen because they are important functions in the four applications being analyzed, which are described in further detail in Section II-C. Also note that the SC-CRAM process for these functions are also described in our previous work [25].

Scaled addition can be accomplished using MUX, which can be implemented using two AND gates, one NOT gate, and one OR gate. Note that in SC-CRAM, it is more feasible to perform NOT logic using a NAND gate, where one of the inputs is set to a constant value of "1." Furthermore, OR logic should be performed using two NOT gates and a NAND gate (see [25]). The total circuit in SC-CRAM consists of three input cells (A, B, and S), five intermediate cells ($\overline{S}$, M_1, M_2, $\overline{M_1}$, and $\overline{M_2}$), and one output cell (Y). First, perturb pulses are applied to A, B, and S, where the switching probabilities of A and B are input dependent and the switching probability of S is 0.5. This is followed by one NOT logic step on S to obtain $\overline{S}$ and two AND logic steps on A, S, and B, $\overline{S}$ to obtain M_1 and M_2, respectively. Two additional NOT logic steps are applied to M_1 and M_2 to obtain $\overline{M_1}$, and $\overline{M_2}$ and finally, NAND logic is applied to obtain Y. The final output bit-stream should produce the function $Y \approx S * A + (1 - S) * B$ or $Y \approx 0.5 * (A + B)$.

Scaled division of stochastic bit-streams A and B can be calculated using the logic for a JK flip-flop. Note that inputs A and B correspond to the J and K terminals, respectively. The JK flip-flop is implemented using one NOT gate, four NAND gates, and one BUFFER. It should be noted that BUFFER logic in SC-CRAM is more feasible using an AND gate, where one of the inputs is set to a constant value of "1." The total circuit in SC-CRAM consists of two input cells (A and B), five intermediate cells (Q, $\overline{Q}$, J, K_1, K_2), and one output MTJ (Y). Perturb pulses are applied to A and B, which are input dependent. This is followed by one NOT step on Q to obtain $\overline{Q}$, one NAND step on $\overline{Q}$ and A to determine J, one NAND step on Q and B to determine K_1, and one NAND step on Q and K_1 to obtain K_2. The final steps are one NAND step on

TABLE II
CRITERIA FOR KEY ARITHMETIC FUNCTIONS IN SC-CRAM

Function	Number of CRAM cells [(a)]	Number of Computation steps [(b)]	Equation	Logic gates used
Multiplication	3	769	$Y \approx A * B$	AND
Scaled Addition	9	1027	$Y \approx S * A + (1 - S) * B$	MUX (1 NOT, 2 AND, 1 OR)
Scaled division	8	1027	$Y \approx \frac{A}{A + B}$	JK Flip Flop (1 NOT, 1 BUFFER, 4 NAND)
Absolute valued subtraction[(c)]	8	1282	$Y \approx \lvert A - B \rvert$	XOR (1 NAND, 1 OR, 1 AND)
Square root	11	1538	$Y \approx \sqrt{X}$	1 AND + 2 OR
Exponential	19	3331	$Y \approx \exp(-4X)$	3 NAND, 2 AND + 4 cascading AND

[(a)] Number of cells indicate the number of CRAM columns used in the sub-array. All functions only require a single row of CRAM cells.
[(b)] Assumes 8 bit resolution (256 length bit streams).
[(c)] Assumes maximum correlation between input bit-streams

K_2 and J to determine Y followed by one BUFFER operation on Y to obtain Q for the next cycle. Note Q is set to "0," or the P-state, for the first cycle. The bit-stream for Y should produce the function $Y \approx A/(A + B)$.

Absolute valued subtraction of stochastic bit-streams A and B can be done using XOR logic. In CRAM, an XOR gate can be implemented using one NAND gate, one AND gate, and one OR gate, which consists of two NOT gates and one NAND gate. The total circuit in SC-CRAM consists of two input cells (A and B), four intermediate cells (M_1, M_2, $\overline{A}$, and $\overline{B}$), and one output cell (Y). Perturb pulses are applied to A and B, which are input dependent, however, unlike the other functions, the bit-streams for A and B need maximum correlation. The process for achieving maximum correlation will be explained further in Section II-C. The perturb step is followed by two NOT steps on A and B to obtain $\overline{A}$ and $\overline{B}$, one NAND step on A and B to obtain M_1, one NAND step on $\overline{A}$ and $\overline{B}$ to obtain M_2, and one AND logic step on M_1 and M_2 to obtain Y. The final output bit-stream should produce the function $Y \approx |A - B|$.

The square root of stochastic bit-stream X can be calculated using AND logic followed by two layers of OR logic. Note that the two OR gates both consist of two NOT gates and one NAND gate. The total SC-CRAM circuit consists of four input cells (X_1, X_2, C_1, and C_2), six intermediate cells ($\overline{X_2}$, M_1, M_2, $\overline{M_1}$, $\overline{M_2}$, and $\overline{C_2}$) and one output MTJ (Y). Perturb pulses are applied to X_1, X_2, C_1, and C_2, where the switching probabilities of X_1 and X_2 are equal and input dependent and the switching probabilities of C_1 and C_2 are fixed at 67% and 18%, respectively. Note that even though the switching probabilities of X_1 and X_2 are equal, the bit-streams should be generated independently, meaning they are decorrelated. This is followed by one AND step on X_1 and C_1 to obtain M_1, two NOT logic steps on M_1 and X_2 to obtain $\overline{M_1}$ and $\overline{X_2}$, one NAND step on $\overline{M_1}$ and $\overline{X_2}$ to obtain M_2, two NOT steps on M_2 and C_2 to obtain $\overline{M_2}$ and $\overline{C_2}$, and one NAND step on $\overline{M_2}$ and $\overline{C_2}$ to obtain Y. The final output bit-stream should produce the function $Y \approx \sqrt{(X_1)}$.

The method of solving an exponential function for our analysis was the same as the one described in [15]. In our case, our desired function was $Y \approx \exp(-4x)$. Note that this function can be rewritten as $Y \approx \exp[(-4/5x)5]$. The first step is to solve $\exp(-4/5x)$ using the third order Maclaurin expansion. This can be implemented in SC-CRAM using three NAND gates and two AND gates. The total SC-CRAM circuit for this first step consists of six input cells (X_1, X_2, X_3, A_1, A_2, and A_3), four intermediate cells (M_1, M_2, M_3, and M_4), and one output cell (B_0). Perturb pulses are applied to all the input cells, where the switching probabilities of X_1, X_2, and X_3 are all equal, decorrelated, and input dependent and the switching probabilities of A_1, A_2, and A_3 are fixed at 80%, 40%, and 26.7%, respectively. NAND logic is used on X_1 and A_3 to obtain M_1, AND logic is used on M_1 and A_2 to obtain M_2, NAND logic is used on M_2 and X_2 to obtain M_3, AND logic is used on M_3 and A_1 to obtain M_4, and NAND logic is used on M_4 and X_3 to obtain B_0. Note that the bit-stream for B_0 is approximately $\exp(-4/5x)$. In order to calculate $\exp(-4x)$ from B_0, four additional cascading AND gates are used on the bit-stream for B_0, where B_0 is buffered at each gate.

D. Neuromorphic Computing Applications

In this section, we describe the four applications that were analyzed in this study. These applications are local image thresholding for character recognition, Bayesian inference for object location, BBN for heart disaster (HD) prediction, and KDE for object recognition. For each application, we compare the energy consumption and noise margin between each category of MTJs. It should be noted that the applications were also analyzed in our previous work [25], where we compared the performance metrics of SC-CRAM to conventional CRAM. Our results showed that SC-CRAM required significantly less CRAM cells than conventional CRAM, which leads to improved noise margin. Additionally, SC-CRAM requires less computation steps to perform these tasks, which can be attributed to the large number of logic gates required for conventional CRAM. Furthermore, the perturb and reset steps can be performed in parallel with the logic steps in separate cells, meaning that the computation speed of SC-CRAM does not increase as drastically with network size as it does with conventional CRAM. In this study, we are comparing the performance metrics of different MTJ technologies.

Image thresholding is a very important step for optical character recognition. Najafi and Salehi [19] applied SC to a local image threshold technique called the Sauvola method. In this method, a window of 9 × 9 pixels is defined within a subsection of a degraded input image. At each subsection, a threshold (T) is calculated using (3a), where $A(x, y)$ is the pixel intensity at point (x, y) and $\overline{A}$ and σA are the mean and standard deviation of all the pixels within the window and σA is calculated using (3b). The circuit for calculating (3a) is shown in Fig. 3(a). Note that XOR logic is used to find $|\overline{A^2} - (\overline{A})^2|$. In this example, maximum correlation between the bit-streams for $\overline{A^2}$ and $(\overline{A})^2$ can be ensured since the bit-streams for A are used to generate bit-streams for A^2.

$$T(x, y) = \overline{A}(x, y) \cdot \left(\frac{\sigma A(x, y) + 1}{2}\right) \tag{3a}$$

$$\sigma A(x, y) = \sqrt{\left|\overline{A^2} - \left(\overline{A}\right)^2\right|}. \tag{3b}$$

In the second application, a Bayesian inference system is used to determine the location of an object using distance (D) and bearing (B) data from three noisy sensors. The Bayesian inference mechanism, which is described in further detail in [36], produces a distribution of object locations by calculating the product series of conditional probabilities. The models for the probabilities for D and B for sensor j at position (x, y) can be described the Gaussian distributions in (4a) and (4b), where μ_{dj} is the Euclidean distance between sensor j and position (x, y), θ_{dj} is equal to $5 + \mu_{dj}/10$, μ_{bj} is the viewing angle of sensor j, and θ_{bj} is set to 14.0626°. This data is used to calculate the object location probability at position (x, y), $p(x, y)$, using (4c). The problem described in [40] and used for our analysis $p(x, y)$ data on a 64 × 64 2-D grid and the sensors are located at positions (0, 0), (0, 32), and (32, 0). The circuit to solve (4c) is shown in Fig. 3(b). Each grid location uses five AND gates to calculate $p(x, y)$. When implemented in SC-CRAM 11 cells are needed at each location (three at the input AND gate and two for the remaining four layers)

$$p\left(D_j \middle| x, y\right) = \mathrm{N}\left(\mu_{dj}, \theta_{dj}\right) \tag{4a}$$

$$p\left(B_j \middle| x, y\right) = \mathrm{N}\left(\mu_{bj}, \theta_{bj}\right) \tag{4b}$$

$$p(x, y) = \prod_j p\left(B_j \middle| x, y\right) * p\left(D_j \middle| x, y\right). \tag{4c}$$

The third application is a BBN for HD prediction, which is a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph. The parent nodes in this network are exercise (E) and diet (D) and the child nodes are high blood pressure (BP) and chest pain (CP). The conditional probability tables for each node are shown in [40]. The probability of an HD, $P(\mathrm{HD})$, is shown in (5a), where $P_{\mathrm{HD}}^{E,D}$ is the HD probability when only considering E and D, P_{BP} and P_{CP} are the HD probabilities for cases of high BP and CP, respectively. The expression for $P_{\mathrm{HD}}^{E,D}$ is shown in (5b), where P_D, P_E, and $P_{E,D}$ represent the HD probabilities for cases of regular exercise, a good diet, and both, respectively. The circuit for calculating (5a) is shown in Fig. 3(c). Note that the final value for $P(\mathrm{HD})$ is calculated using a JK flip-flop, which was implemented in SC-CRAM using the process described in the previous section

$$p(\mathrm{HD}) = \frac{P_{\mathrm{BP}} * P_{\mathrm{CP}} * P_{\mathrm{HD}}^{E,D}}{P_{\mathrm{BP}} * P_{\mathrm{CP}} * P_{\mathrm{HD}}^{E,D} + \overline{P_{\mathrm{BP}}} * \overline{P_{\mathrm{CP}}} * \overline{P_{\mathrm{HD}}^{E,D}}} \tag{5a}$$

$$p_{\mathrm{HD}}^{E,D} = \left[P_{E,D}P_D + P_{E,\overline{D}}P_{\overline{D}}\right]P_E + \left[P_{\overline{E},D}P_D + P_{\overline{E},\overline{D}}P_{\overline{D}}\right]P_{\overline{E}}. \tag{5b}$$

The fourth application examined is KDE, which is an image segmentation algorithm for object recognition, surveillance, and tracking [18]. The KDE algorithm is based on recent information that is continuously updating. Sample values of pixel intensity, X, are captured from recent iterations $(X_t, X_{t-1}, \ldots, X_{t-N})$, which can be used to determine the probability density function (pdf), as described in (6). The circuit for calculating (6) is shown in Fig. 3(d), where X_t represents the pixel intensity at time t and X_{t-i} represents the pixel intensity at the ith previous cycle

$$\mathrm{PDF}(X_t) = \frac{1}{N}\sum_{i=1}^{N} e^{-4|X_t - X_{t-i}|}. \tag{6}$$

III. Calculations and Benchmarking

For our analysis, we considered the performance of SC-CRAM for six different categories of MTJs. These are the top STT and SOT performing MTJs from research groups, STT and SOT from industry, and projected performance of STT and SOT MTJs. For research MTJs, the parameters for STT-CRAM, were obtained from [41], [42], [43], [44] and the parameters for SOT-CRAM were obtained from [34], [48], [49], [50], [51]. For industry MTJs, the parameters for STT-CRAM and SOT-CRAM were obtained from those recently reported by IBM [45] and IMEC [52], [53], respectively. For projected MTJs in STT-CRAM, we used the parameters that were predicted in [46] and [47]. However, since SOT-MRAM is not as well established for industrial purposes as STT-MRAM, accurate projected parameters for SOT MTJs are not yet reported. Therefore, for projected MTJs in SOT-CRAM, we used the best reported parameters from various different studies [35], [54].

For each category, we calculated the total energy consumption for six different arithmetic functions. These are multiplication, scaled addition, scaled division, absolute valued subtraction, square root, and exponential function. Recall that the logic gates used for each of these calculations were described in Section II-B and are shown in Table II. Total energy consumption was also calculated for each of the four neuromorphic computing applications described in Section II-C. Finally, our analysis considered the influence of variations in device dimensions and determined which category of MTJs was the most resilient to these random variations in SC-CRAM. Details of which parameters we considered, how energy consumption was determined, and how the effect of device variations were implemented in our calculations are described in more detail in the following subsections.

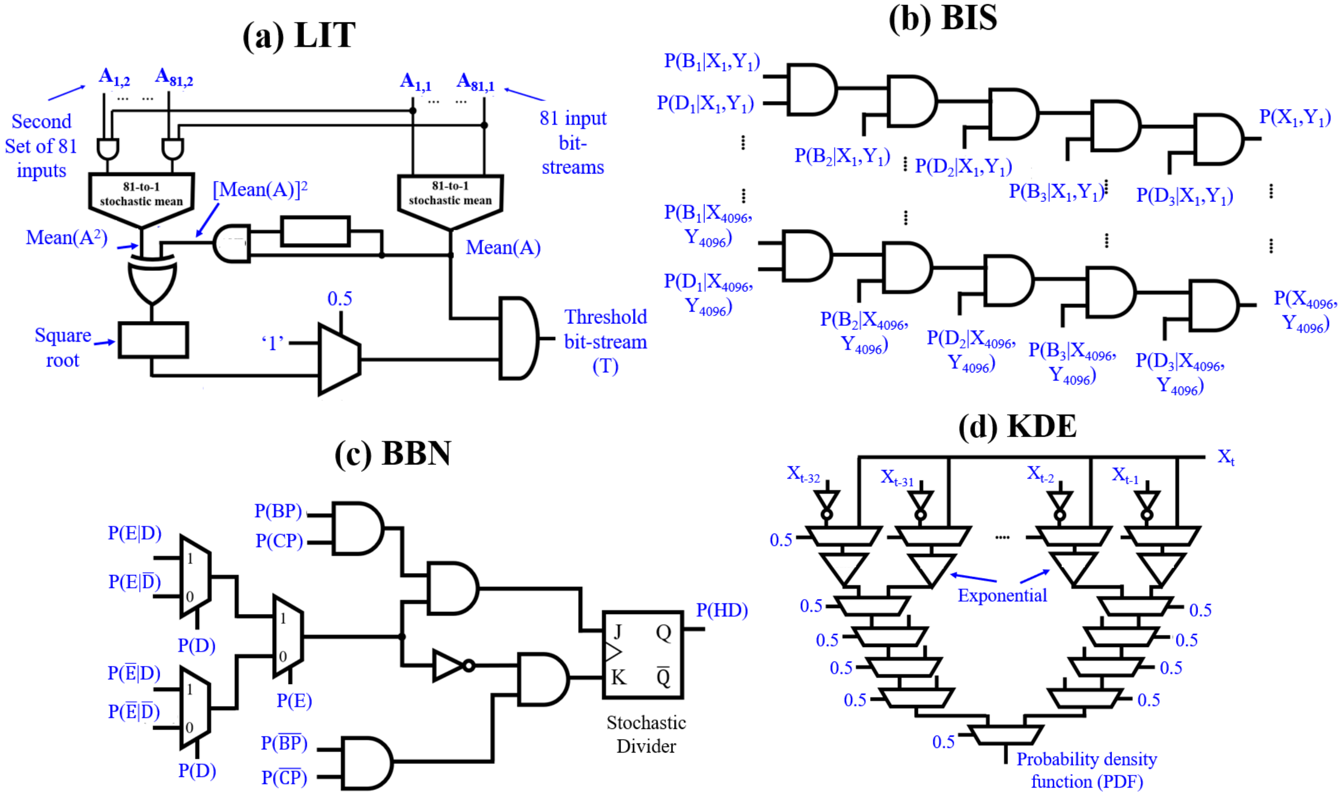

Fig. 3. Circuit diagrams for (a) local image thresholding (LIT), (b) Bayesian inference system (BIS), (c) Bayesian belief network (BBN), and (d) Kernel density estimation (KDE). Images extracted from [25].

A. Parameters for Analysis

There are six key MTJ properties that were considered for our analysis. These properties are the resistance-area (RA) product, the tunneling magnetoresistance (TMR) ratio, the thermal stability factor (Δ), the intrinsic critical switching current density (J_{C0}), the switching time (τ_{SW}), and the precessional switching coefficient (A_V). For all of our calculations, we assumed that the MTJ pillars were patterned into circular nanopillars with a diameter of 20 nm, meaning that the area is approximately 314 nm^2. Therefore, the resistances in the AP and P-state (R_{AP} and R_P, respectively) can be calculated from the RA product and the TMR ratio, where $R_P = \text{RA}/\text{area}$ and $\text{TMR} = 100 * [(R_{\text{AP}} - R_P)/R_P]$.

In our calculations, τ_{SW} is determined to be the perturb pulsewidth where the switching energy is minimized, which will be explained in further detail in the next section. When operating in the precessional switching regime ($\tau_{\text{SW}} < 5$ ns), A_V determines the relation between pulsewidth and pulse amplitude, where $\tau_{\text{SW}}^{-1} = A_V * (V - V_{C0})$. In some studies, the values for A_V were explicitly reported and for other studies, we had to determine A_V based on the data provided. In either case, values provided for A_V assume a switching probability of 50%. This is an accurate assumption when determining τ_{SW} for the perturb step, however, the actual value for A_V may be different for the reset and logic steps since the desired switching probability is close to 100%. Therefore, for the STT MTJs and research SOT MTJs, pulse widths of 5 ns were used for the reset and logic steps since the calculated energy consumption was minimized at this pulsewidth.

There are three additional parameters that we considered for our SOT-CRAM calculations. These are the resistivity of the spin Hall channel (ρ), the spin Hall angle (θ_{SH}), and the thickness of the spin Hall channel (t_{SOT}). The values for ρ were determined based on the material used for the spin Hall channel for each category. These materials were Ta, W, and BiSe for research MTJs, industrial MTJs, and projected MTJs, respectively. BiSe was chosen for projected MTJs since it has the largest θ_{SH} reported [54]. The values for t_{SOT} were chosen based on the spin Hall channel thickness that provided the maximum θ_{SH} for each material. For our calculations in SOT-CRAM, we assumed a spin Hall channel width and length of 40 and 120 nm, respectively. From these values, we determined the resistance of the spin Hall channel (R_{SHE}) using the calculation $R_{\text{SHE}} = (\rho * \text{length})/(t_{\text{SOT}} * \text{width}) = 3\rho/t_{\text{SOT}}$.

B. Voltage and Energy Consumption Calculations

For each category of MTJs, the total energy consumption was calculated for each arithmetic function described in Table II. The energy consumed from a single voltage pulse is calculated from (7), where R_{MTJ} is the MTJ resistance, V is voltage pulse amplitude, and t_P is the pulsewidth. Note that for STT switching, R_{MTJ} is either R_{AP} or R_P, depending on the initial state of the MTJ, and for SOT switching, R_{MTJ} is replaced by R_{SHE}. The total energy consumption (E_{TOT}) is determined by (8), where N_B is the total number of bits and E_{PERT}, E_{LOGIC}, and E_{RES} are the energies during the perturb, logic, and reset step, respectively. For our calculations, we assumed 8-bit resolution, therefore, $N_B = 256$

$$E = \frac{V^2 t_P}{R_{\text{MTJ}}} \tag{7}$$

$$E_{\text{TOT}} = N_B * (E_{\text{PERT}} + E_{\text{LOGIC}} + E_{\text{RES}}). \tag{8}$$

For all six categories of MTJs, three different combinations of V and t_P were determined, one for perturb, another for logic, and a third for reset operations. To calculate these values, we first considered the desired switching probability (P_{SW}), as defined in (9), where τ is the characteristic switching time. For thermal activation switching ($t_P \geq 5$ ns), τ is defined by (10) where τ_0 is the inverse attempt frequency, which we assumed to be 1 ns [55] and V_{C0} is the intrinsic switching voltage, which was directly calculated from J_{C0} using $V_{C0} = (J_{C0}/\text{area}) * R_{\text{MTJ}}$

$$P_{\text{SW}} = 1 - \exp\left(-\frac{t_P}{\tau}\right) \tag{9}$$

$$\tau = \tau_0 \exp\left[\Delta\left(1 - \frac{V}{V_{C0}}\right)\right]. \tag{10}$$

It should be noted that (10) cannot be used for precessional switching ($t_P < 5$ ns). For the perturb step, we assumed $P_{\text{SW}} = 0.5$, which means that $t_P \approx \tau$. This means that V could be calculated directly from t_P using $t_P^{-1} = A_V * (V - V_{C0})$. However, for the reset and logic steps, we assumed $P_{\text{SW}} = 0.99$, so this calculation needs to be modified. From (7), we can determine that $t_P/\tau \approx 4.6$ when $P_{\text{SW}} = 0.99$, meaning that τ can be used to determine V. This means that AV should be modified accordingly for cases when $P_{\text{SW}} = 0.99$. Note that this modification of A_V was only done for the logic and reset steps but was not necessary for perturb steps.

We calculated V from t_P values ranging between 0.25 and 20 ns for perturb, reset, and logic operations and calculated the switching energy at each point using (7). The combination of V and t_P that provided the lowest switching energy was the one that was chosen in our calculations. The example provided in Fig. 4(a) shows a plot of the switching energy versus t_P for the perturb operation in the research STT MTJs. Here we see that the switching energy is minimized when $t_P = 1.25$ ns, which corresponds to $V = 544$ mV, as shown in Fig. 4(b). This calculation was repeated for logic and reset operations as well as for all operations for the remaining five MTJ categories.

C. Device Variations

Our calculations also considered the effect of device variations to see which category of MTJs were the most resilient to noise and imperfections in SC-CRAM operations. In our analysis, device variations were measured in terms of a percent change in MTJ diameter from the nominal 20 nm. For SOT-CRAM, we also had to account for percent change in spin Hall channel width from the nominal 40 nm. Note that a percent change in MTJ diameter directly correlates to an equal percent change in R_P and R_{AP} (σR_P and σR_{AP}, respectively). Furthermore, since Δ and V_{C0} are proportional to the free layer volume, we assumed that $\sigma\Delta = -\sigma R_{\text{MTJ}}$ and $\sigma V_{C0} = 0.1 * \sigma R_{\text{MTJ}}$.

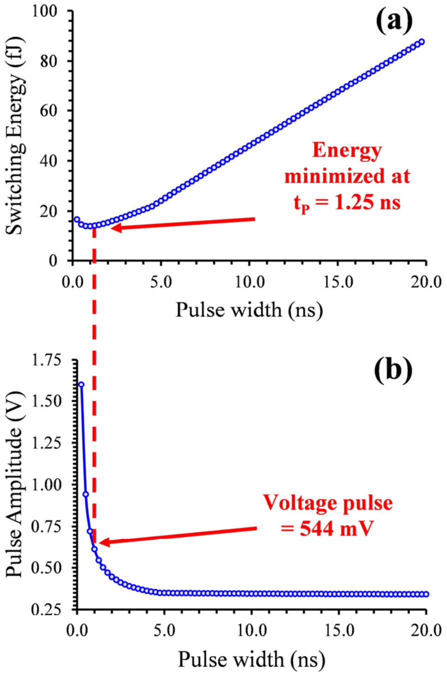

Fig. 4. Plot for (a) switching energy versus pulsewidth and (b) pulse amplitude versus pulsewidth for the perturb operation in research MTJs.

The SC-CRAM process for the six arithmetic functions listed in Table II was simulated in MATLAB for bit-stream probabilities ranging from 10% to 90% for all six MTJ categories. Each simulation was repeated 100 times and the final calculation was averaged among all trials. To quantify the accuracy of the output, the mean square error (MSE) was calculated from these simulations for all six categories of MTJs considered. Expression for MSE is shown in (11), where N_{PTS} is the total number of points plotted, and Y_{CALC} and Y_{EXP} are the calculated and expected outputs, respectively. In our study, we repeated each calculation for $\sigma R_{\text{MTJ}} = 0\%$–$30\%$, assuming a linear distribution. Note that R_{MTJ} was recalculated for each trial for each value of σR_{MTJ} tested. Fig. 5 provides an example for output probability versus input probability for multiplication via AND logic in SC-CRAM using research STT MTJs for 0% noise and 20% noise, which illustrates how MSE was calculated

$$\text{MSE} = \frac{1}{N_{\text{PTS}}} \sum_{i=1}^{N_{\text{PTS}}} \left(Y_{\text{EXP},i} - Y_{\text{CALC},i}\right)^2. \tag{11}$$

IV. Results

A. Influence of Device Variations

Fig. 6(a)–(f) shows the results for MSE calculations for MTJ device variations (σR_{MTJ}) between 0% and 30%. MSE for multiplication via AND logic is shown in Fig. 6(a). These results show that the projected SOT and STT MTJs are the most resilient to MTJ variations, where MSE $< 10^{-4}$ even when $\sigma R_{\text{MTJ}} = 30\%$. Research SOT and STT MTJs will maintain an acceptable MSE $< 10^{-3}$ when $\sigma R_{\text{MTJ}} < 20\%$. However, the industry STT MTJs had the worst performance at both low and high σR_{MTJ}. Recall that the projected SOT

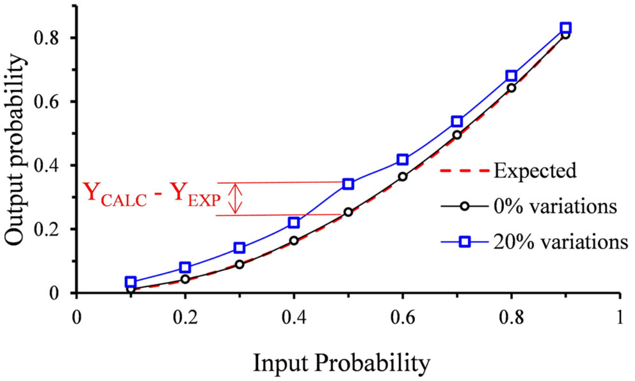

Fig. 5. Output probability versus input probability for 0% and 20% variations in MTJ diameter. This example shows multiplication of two input bit-streams via AND logic in SC-CRAM for research STT MTJs.

and STT MTJs both have a TMR ratio of 200% whereas the industry STT MTJs have a TMR ratio of 84%. These results show that the accuracy and resiliency to device variations for multiplication using the AND logic are highly dependent on the TMR ratio.

The MSE calculations for scaled addition via MUX are shown in Fig. 6(b). These results show that the MSE is low when σR_{MTJ} is low, but increases rapidly as σR_{MTJ} increases. This means that the results for scaled addition are more susceptible to variations than for multiplication. One possible explanation is that the output accuracy is dependent on the accuracy of the bit-stream generated from the selector (S) MTJ. The accuracy of the output will vary if output probability of S deviates from 50%. However, the MSE is still lowest for projected SOT and STT MTJs, where MSE $< 10^{-3}$ for σR_{MTJ} up to 20%.

Fig. 6(c) shows that for scaled division via a JK flip-flop, the MSE $< 10^{-4}$ for σR_{MTJ} up to 10% or 15% for all categories except for industry STT MTJs. Furthermore, the research and projected STT MTJs show the most resiliency for $\sigma R_{\mathrm{MTJ}} < 20\%$. However, the MSE spikes above 10^{-3} for $\sigma R_{\mathrm{MTJ}} = 20\%$. These results illustrate that device variations have a much stronger impact on accuracy as the number of consecutive NAND gates increases.

Fig. 6(d) shows that the behavior of MSE with σR_{MTJ} for absolute valued subtraction via XOR logic is similar to that of scaled division. These results show that industry and research STT MTJs are the most susceptible to device variations. For the other four categories, the MSE values are less than 10^{-3} for $\sigma R_{\mathrm{MTJ}} < 25\%$. Furthermore, the projected STT MTJs were the most resilient to device variations.

The MSE calculations for the square root function are shown in Fig. 6(e). These results show that the MSE values are higher for the square root function than any other functions. This can be attributed to the fact that the square root function relies on two MTJs that generate constant bit-streams, C_1 and C_2, with probabilities of 18% and 67%, respectively. As with the scaled addition function, the accuracy of the output will vary when the output probabilities of C_1 and C_2 deviate from their desired values. At low σR_{MTJ}, the research STT MTJs have the lowest MSE, however, the projected STT and SOT MTJs are the most resilient to MTJ variations where MSE $< 10^{-3}$ for $\sigma R_{\mathrm{MTJ}} = 15\%$.

Fig. 6(f) shows the results of MSE with σR_{MTJ} for the exponential function. For most of the MTJ categories, MSE increased steadily with σR_{MTJ} from 10^{-5} for $\sigma R_{\mathrm{MTJ}} \leq 10\%$ to MSE $\approx 10^{-3}$ for $\sigma R_{\mathrm{MTJ}} = 30\%$. The two exceptions are the STT research MTJs and the projected SOT MTJs. The research STT MTJs showed the highest MSE for all σR_{MTJ} values. Alternatively, the projected SOT MTJs showed the best resiliency to device variations, where MSE $\approx 10^{-4}$ at $\sigma R_{\mathrm{MTJ}} = 30\%$. It should be noted that the exponential function has more perturb steps than the other five functions analyzed. Therefore, these results indicate that calculations in SC-CRAM have the best resiliency to variations in switching probability when using the projected SOT MTJs and the worst resiliency when using the research STT MTJs.

B. Comparison of Energy Consumption

The bar plots in Fig. 7 show the energy consumption for each function in each MTJ category. These plots show that for all categories, multiplication consumes the least amount of energy and the exponential function consumes the most energy at around 10× more than multiplication. Scaled addition, scaled division, and absolute valued subtraction functions all consume more energy than multiplication but less than the exponential function. This trend does not differ significantly between each MTJ category because it reflects the number of CRAM cells used and the number of logic steps required for each function.

The energy consumption for STT research MTJs is about one order of magnitude larger than for STT industrial and projected MTJs for all functions. This can be attributed to both larger J_{C0} and larger RA product for research STT MTJs when compared to industrial and projected STT MTJs. Both of these factors lead to larger voltages required for the perturb, logic, and reset steps. Additionally, the energy consumption for the industrial and projected STT MTJs is reduced further in comparison to the research STT MTJs by the fact that their switching energy is minimized at shorter pulses (recall Table III).

The results in Fig. 7 also shows that the energy consumption for research SOT MTJs is approximately 3× larger than for industrial SOT MTJs. This may seem counterintuitive since J_{C0} is larger for the industrial SOT MTJs. However, there are two factors that cause the energy consumption in research SOT MTJs to increase when compared to industrial SOT MTJs. One is that the industrial SOT MTJs use shorter pulsewidths for switching (recall Table III) and the second is the smaller RA product in the MTJ pillars for the industrial MTJs. The projected SOT MTJs had the lowest energy consumption of the SOT categories, where the energy consumption was over one order of magnitude lower than the industrial SOT MTJs over two orders of magnitude lower than the research SOT MTJs.

One key observation is that the energy consumption in the STT MTJs is significantly lower than the research and industrial SOT MTJs. There are two factors that cause this increase in energy consumption for the research and industrial

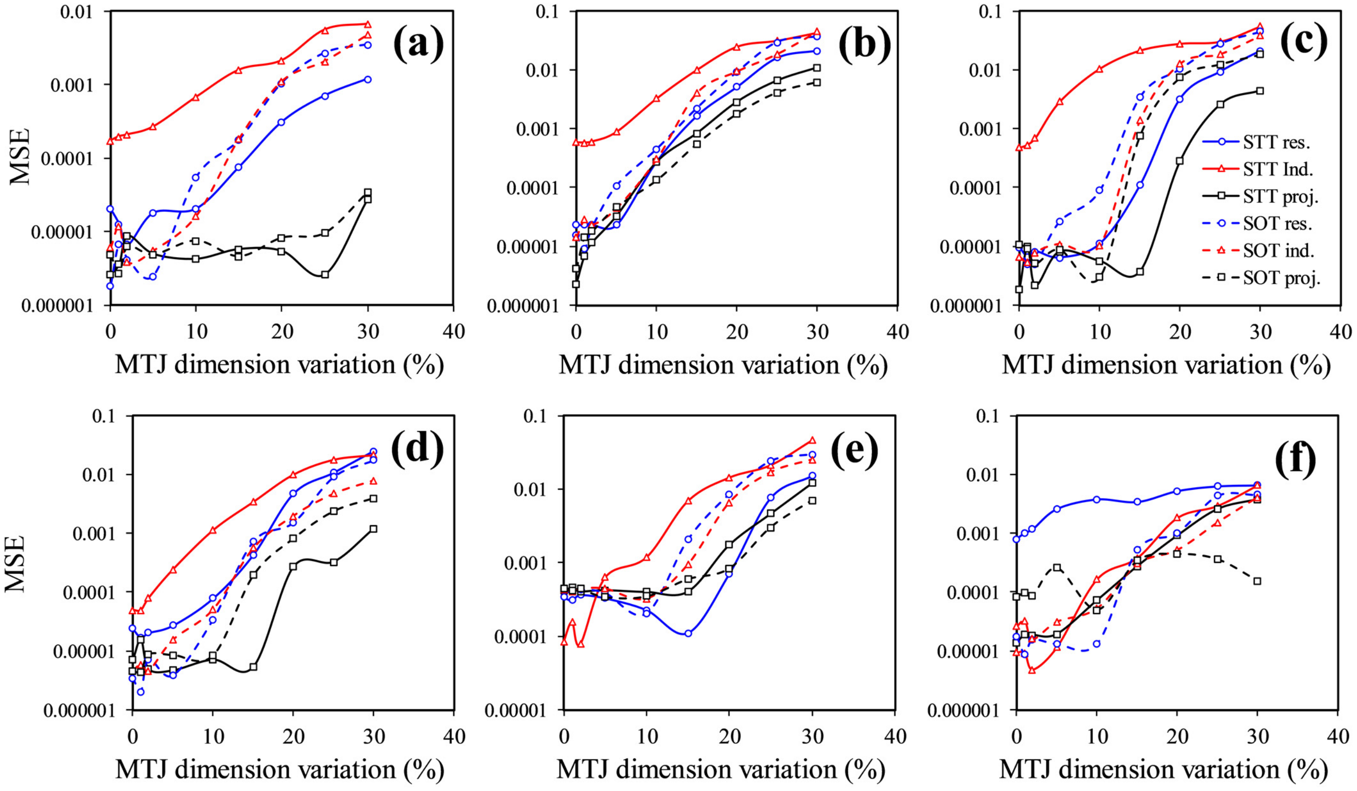

Fig. 6. MSE versus MTJ dimension variation for (a) multiplication, (b) scaled addition, (c) scaled division, (d) absolute valued subtraction, (e) square root, and (f) exponential function in SC-CRAM. The results for the STT and SOT MTJs are shown as solid and dotted lines, respectively. The lines for the research, industrial, and projected MTJs are shown with blue circles, red triangles, and black squares, respectively.

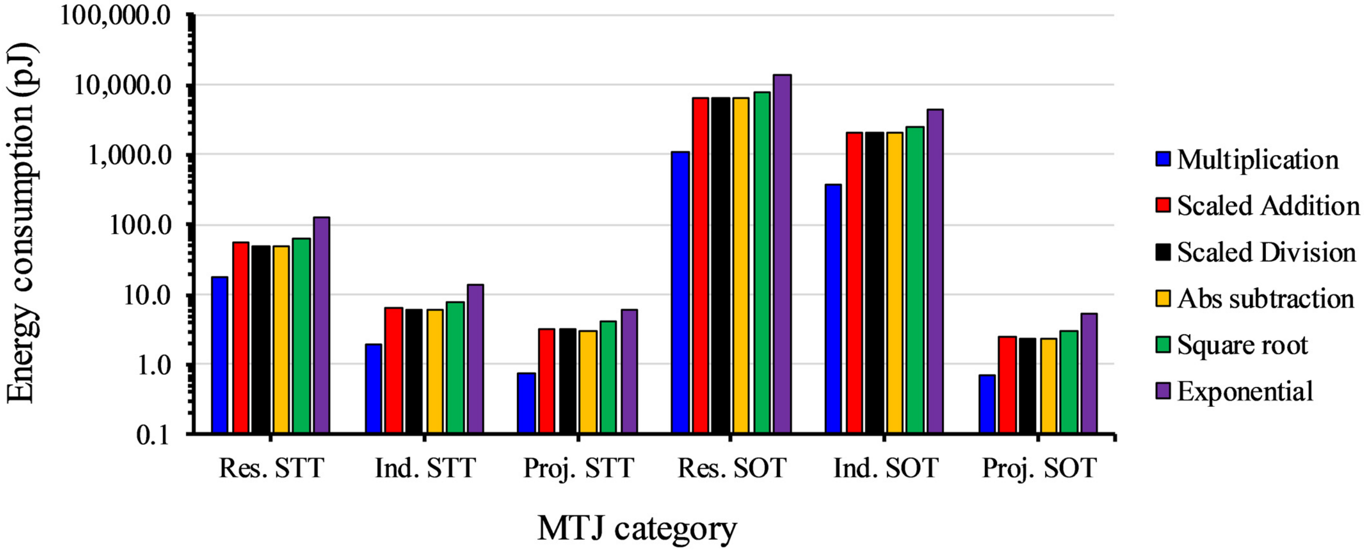

Fig. 7. Energy consumption for the six arithmetic functions analyzed for all MTJ categories.

SOT MTJs. One is the larger J_{C0} values, which leads to larger voltages for the perturb and reset steps. The second, and much more significant factor is the large RA products of the MTJ pillars for the research and industrial SOT MTJs. This increases the voltage required for the logic step since the resistance of the pillars is much larger than the resistance of the spin Hall channel. However, the energy consumption in the projected SOT MTJs is lower than any of the STT MTJ categories. This is because the logic voltages are reduced significantly for projected SOT MTJs because the RA product is much lower and the resistance of the BiSe spin Hall channel is much larger when compared to the research and projected SOT MTJs. For optimal performance of SOT-based SC-CRAM, the resistance of the spin Hall channel should be equivalent to the resistance of the MTJ pillar. Furthermore, low RA product of the MTJ pillars is desired.

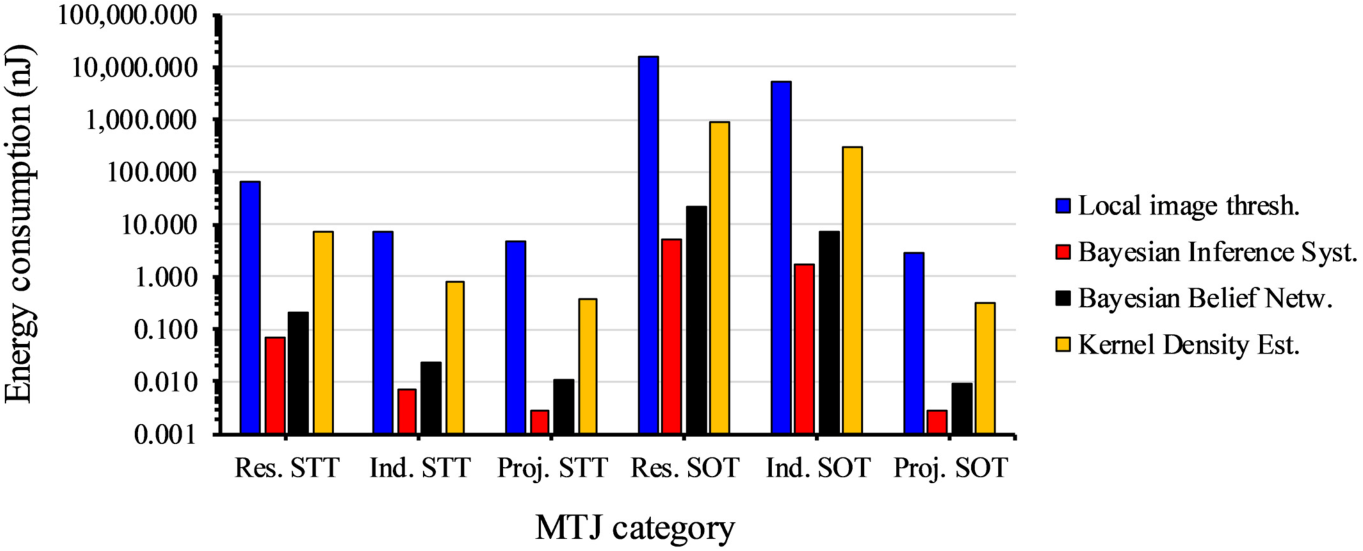

Fig. 8. Energy consumption for the four neuromorphic computing applications analyzed for all MTJ categories.

TABLE III

Parameters Used for Analysis

Parameter	Research	Industry	Projected
STT-CRAM			
RA product ($\Omega\cdot\mu m^2$)	5 [37]	3.68 [41]	1 [42]
TMR ratio	133 [38]	82 [41]	200
Δ	60 [39]	45 [41]	75 [43]
J_{C0} (MA/cm^2)	3.1 [37]	1.25 [41]	1 [37]
τ_{SW}(ns)	1.25	0.75	0.75
A_V ($s^{-1}V^{-1}$)	2.1 x 10^9 [40]	1.5 x 10^{10} [41]	1.5 x 10^{10} [41]
SOT-CRAM			
RA product ($\Omega\cdot\mu m^2$)	12.3 [44]	17.5 [48]	1 [42]
TMR ratio	94 [45]	110 [49]	200
Δ	45 [45]	48 [49]	60 [50]
J_{C0} (MA/cm^2)	75 [46]	100 [48]	1 [35]
τ_{SW} (ns)	2	0.75	0.25
A_V ($s^{-1}V^{-1}$)	4.76 x 10^8 [48]	1.46 x 10^{10} [48]	1.46 x 10^{10} [48]
SOT material/ ρ ($\mu\Omega\cdot$cm)	Ta / 190 [34]	W / 160 [49]	BiSe / 2150 [50]
θ_{SH}	-0.25 [34]	-0.32 [48]	2.88 [50]
t_{SOT} (nm)	5 [47]	3.5 [49]	8 [50]

C. Performance Metrics in Neuromorphic Applications

Fig. 8 shows the total energy consumption for local image thresholding, object location, HD prediction, and KDE for all MTJ categories. It should be noted that the analysis for number of CRAM cells and computation steps in SC-CRAM was done in our previous work [25] and will not change between the MTJ categories. The bar plots in Fig. 8 show that for each MTJ category, local image thresholding has the highest energy consumption followed by KDE, HD prediction, and object location. This is simply a reflection on the number of CRAM cells required for each task. Furthermore, the trends observed in Fig. 8 follow similar trends to those observed in Fig. 7, where the research and industrial SOT MTJs have the largest energy consumption among the MTJ categories and the projected SOT MTJs have the lowest.

One of the key observations from the results in Fig. 8 is that relative energy consumption for local image thresholding and KDE between SOT and STT MTJs is lower than the results in Fig. 7. In Fig. 7, the energy consumption for the projected SOT MTJs is approximately 1.05–1.3× lower than for projected STT MTJs. However, in Fig. 8, the energy consumption for projected SOT MTJs is approximately 1.5–1.6× lower than for projected STT MTJs for local image thresholding and KDE. This is because both of these tasks require a large number of NAND logic operations. For SOT MTJs, the energy consumption for NAND logic is the same for AND logic. However, for STT MTJs, the energy consumption for NAND logic is larger than for AND logic. The energy consumption is similar between STT and SOT MTJs for most arithmetic functions. However, these results show that the energy consumption in projected SOT MTJs decreases relative to the projected STT MTJs for larger scale tasks, particularly those involving NAND logic.

V. Discussion

A. Explanation of Key Results

The two sources of error that contribute to the increase in MSE with σR_{MTJ} shown in Fig. 6(a)–(f) are errors in the logic operation and inaccurate switching probabilities during the perturb step. With no MTJ variations, the MSE should be minimal ($<10^{-5}$) since the only source of error is the small changes in switching probability. Note that this can be improved by increasing the bit-stream length. However,

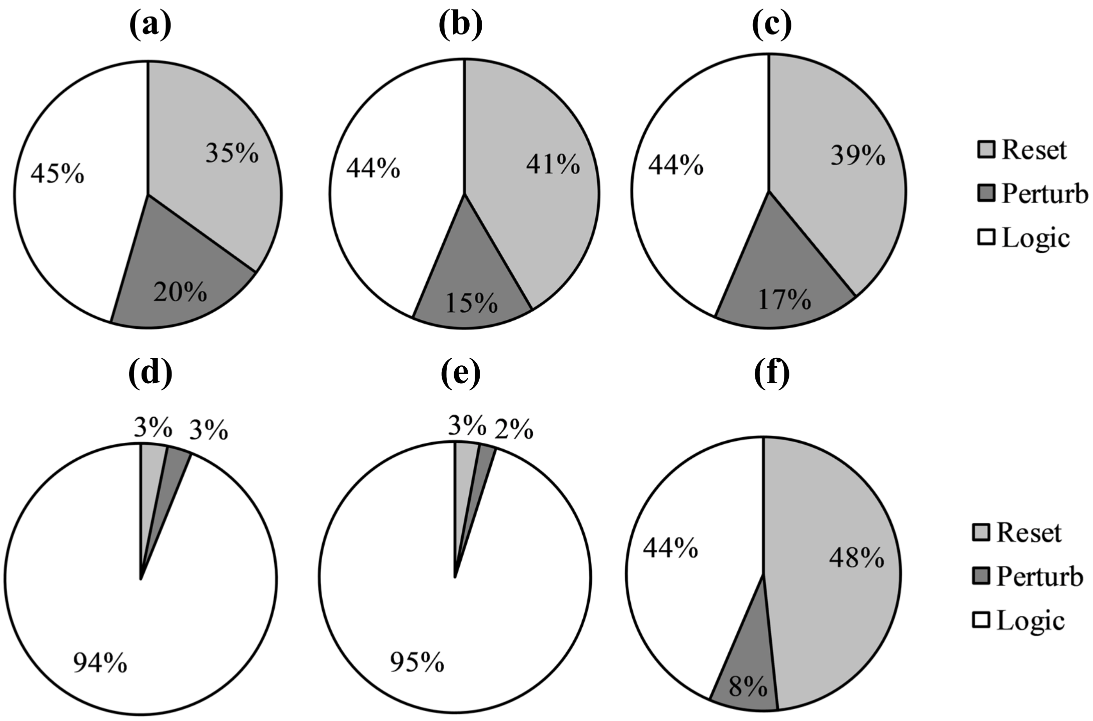

Fig. 9. Breakdown of energy consumption between the reset, perturb, and logic steps for multiplication via AND logic for all MTJ categories. (a) Research STT. (b) Industrial STT. (c) Projected STT. (d) Research SOT. (e) Industrial SOT. (f) Projected SOT.

MTJ variations will change their properties, meaning that V_B may fall outside the range of values outlined in Table I. This will lead to logic errors since V_B will either be too small and fail to switch the output MTJ for the proper input configuration or too large and will switch the MTJ for incorrect input configurations. The range of V_B values for proper logic operations in CRAM increases as the TMR ratio of the MTJs increases. This explains why the projected SOT and STT MTJs were the most resilient to MTJ variations for most of the functions in Fig. 6(a)–(f) and the industry MTJs were the most sensitive to these variations.

Perturb errors are relatively insignificant compared to logic errors, however, there are a few functions where perturb errors have a larger impact. Namely, scaled addition and square root function are more influenced by perturb errors. This is because these two functions rely on bit-streams that are generated separately from the input bit-streams with a constant, predetermined probability. Not only does this add additional sources of perturb errors, but also it changes the function being processed. For example, the square root function uses constant probabilities of 18% and 67% to process the function $Y \sim X^{0.5}$. Changing the constant probabilities will change the function being processed.

Another key result shown in the data in Fig. 6(a)–(f) is that variations had a larger influence on SOT MTJs than STT MTJs. Note that projected SOT MTJs have more resilience to variations than industry MTJs, they do not have more resilience to projected STT MTJs for most functions. In some cases, the research STT MTJs had better resilience to variations than the projected SOT MTJs. This is because, for SOT MTJs, device variations influence the diameter of the MTJ pillar as well as the width of the spin Hall channel whereas for STT MTJs, device variations only influence the diameter of the MTJ pillar. This additional source of variation for SOT MTJs means that V_{C0} changes more with device variations for SOT MTJs than for STT MTJs. This is because when the width of the spin Hall channel changes, R_{SHE} changes accordingly. While J_{C0} remains the same, V_{C0} changes with R_{SHE}, which will drastically affect the SOT switching properties.

There are three components that contribute to the total energy consumption in Figs. 7 and 8. These are the energy for the perturb step, reset step, and logic step. For the perturb and reset steps, the biggest contributing factors for the energy consumption are J_{C0} and switching speed. Having a low J_{C0} along with a low RA product means that V_{C0} is low, which leads to low voltage pulse amplitudes for the perturb and reset steps. However, for the logic step, the resistance of the input MTJs relative to the resistance of the output is another factor that contributes to the energy consumption. If the resistance of the output is low relative to the resistance of the input MTJs, then the amplitude of the voltage pulse needs to increase accordingly.

Fig. 9 shows the breakdown of the total energy consumption for AND multiplication for all six MTJ categories. These results show that the breakdown is very similar between research, industrial, and projected STT MTJs, where the reset and logic steps each consume approximately 35%–45% of the total energy consumption and the perturb step consumes around 15%–20%. This means that for STT MTJs, the reduction in total energy consumption almost entirely depends on reducing J_{C0} and the RA product, which is confirmed by the results in Fig. 7. However, there are a few more factors to consider when analyzing the total energy consumption for SOT MTJs. Fig. 9 shows that the logic step consumes 95% of the total energy consumption for industry and research SOT MTJs. On the other hand, the logic step only consumes 44% of the total energy consumption for projected SOT MTJs. This is because the resistance of the MTJ pillars for the research and industry MTJs is approximately 40 to 100 kΩ, which is very high compared to the resistance of the SOT channel, which is around 1.1 to 1.3 kΩ. Alternatively, for the projected SOT MTJs, the resistance of the MTJ pillars was around 3 kΩ in the P-state and 9 kΩ in the AP-state, which is on the same order of magnitude as the resistance of the SOT channel, which was ∼8 kΩ. The lower J_{C0} value and faster switching speed for projected SOT MTJs relative to research and industry SOT MTJs certainly are factors in reducing the total energy consumption. However, the results in Fig. 9 show that the biggest factor in reducing the total energy consumption in SOT-CRAM is reducing the energy during the logic step. By reducing the RA product of the pillars while maintaining high SOT efficiency in the SOT channel, the total energy consumption can be significantly reduced.

B. Outlook

Our previous results showed that the energy consumption in SC-CRAM is on the same order of magnitude as that of conventional CRAM [25]. This is quite promising, considering the fact that the energy consumption in conventional CRAM is around 40× lower than for modern near memory processors [41]. In this study, we demonstrated that the energy consumption in SC-CRAM can be reduced even further by optimizing the intrinsic properties of the MTJ, including reducing J_{C0} and RA product of the pillar. Furthermore, utilizing the SOT switching mechanism in SC-CRAM may reduce the energy consumption even more, especially for large-scale applications with large number of NAND operations. Another factor that was not focused on in this study but should be considered is that SC-CRAM benefits from the noise resiliency and robustness to variations associated with SC, however, in SC-CRAM, there is minimal sacrifice in computation delay and energy for stochastic bit generation since this process is embedded within the computation process.

One of the potential challenges in achieving CRAM cells that utilize the SOT switching mechanism is to create SOT channels with high θ_{SH} (low J_{C0}) along with pillars with low RA products. This is because the SOT materials with the largest θ_{SH} values are topological insulators, which typically have large resistivities. This creates potential problems when combining these channels with low RA pillars, since the SOT write current may be shunted into the MTJ pillar rather than contributing to SOT switching. It should be noted that despite this challenge, SOT switching is still preferred over STT switching for two reasons. One is that STT switching has limitations on minimizing switching speed and J_{C0}. The second is that STT switching is less susceptible to breakdown from the write pulse the current direction is adjacent to the free layer rather than across the tunnel junction.

Furthermore, there could be a few solutions to the potential for SOT current shunting. One is that a thin oxide layer could be inserted between the SOT channel and the free layer, which would drastically reduce shunting [56]. However, this would increase the RA product of the pillar, thus increasing the energy consumption during the logic step. Another option is to combine the STT and SOT mechanisms for each step. In previous studies, significant reduction in both switching energy and switching speed has been achieved with this strategy [52]. One method to combine STT and SOT is to use two bias terminals on a three-terminal device. However, this is not an attractive method for the purposes of SC-CRAM since an additional transistor will need to be added to every cell, thus increasing the total circuit area. Another method is to use a two-terminal SOT device, which was done in [48]. This strategy allows for the same reduction in switching energy and speed as the three-terminal devices, but without adding any transistors to each CRAM cell.

Finally, another way that the performance of SC-CRAM could be improved is to use new switching mechanisms, for example, utilizing the voltage-controlled magnetic anisotropy (VCMA) [57] and the voltage-controlled exchange coupling (VCEC) mechanism and its integration with SOT [58], [59]. Previous studies have shown that VCEC switching can be achieved at current densities one order of magnitude lower than for STT switching [59]. Furthermore, our previous studies have shown that VCEC switching can be achieved at current densities as low as 10^3 A/cm^2 when combined with an SOT current [60]. This strategy was not investigated in this study since there are currently no experimental demonstrations that test the switching speed of the VCEC mechanism. However, SC-CRAM based on VCMA and VCEC switching should be investigated in future studies.

VI. Conclusion

SC-CRAM has the same advantages as any other SC scheme, which is resilient to noise and requiring low number of logic gates. However, SC-CRAM has additional advantages since stochastic bit-stream generation and computation are performed in parallel. In this study, we analyzed the performance of SC-CRAM for both STT and SOT switching using metrics from devices obtained from academic studies, current industrial standards, and projected metrics. Our calculations found that the accuracy of the output is dependent on the TMR ratio of the MTJ pillar for both STT and SOT switching. Additionally, we determined that SC-CRAM cells based on STT switching consume significantly less energy than cells based on SOT switching for MTJs with the current academic and industrial properties. This is primarily due to the large RA product of the benchmarked MTJ pillars for SOT switching, therefore, large voltages are required to perform logic operations in SOT-based SC-CRAM. Based on the projected performance metrics, the energy consumption for SOT-CRAM cells can be reduced to levels below STT-CRAM cells if the RA product of the MTJ pillar is minimized.

Acknowledgment

This work was supported in part by the Center for Probabilistic Spin Logic for Low-Energy Boolean and Non-Boolean Computing (CAPSL), one of the Nanoelectronic

Computing Research (nCORE) centers as task 2759.001; in part by the Semiconductor Research Corporation (SRC) Program sponsored by the National Natural Science Foundation (NSF) under Grant 1739635; in part by the Applications and Systems-Driven Center for Energy-Efficient Integrated Technologies (ASCENT), which is sponsored by the NSF under Grant 2230963; in part by SMART, one of the seven centers of nCORE, an SRC Program, sponsored by the National Institute of Standards and Technology (NIST); in part by UMN MRSEC Program under Award DMR-1420013; in part by ASCENT, one of six centers of JUMP, an SRC Program that is sponsored by MARCO and DARPA; in part by DARPA under Grant HR001117S0056-FP-042 "Advanced MTJs for computation in and near random access memory;" in part by NIST; and in part by DARPA Non-Volatile Logic Program, NSF SPX, under Grant 1725420.

References

[1] J. Gómez-Luna, I. E. Hajj, I. Fernandez, C. Giannoula, G. F. Oliveira, and O. Mutlu, "Benchmarking a new paradigm: Experimental analysis and characterization of a real processing-in-memory system," *IEEE Access*, vol. 10, pp. 52565–52608, 2022.

[2] J. Wang and F. Zhuge, "Memristive synapses for brain-inspired computing," *Adv. Mater. Technol.*, vol. 4, no. 3, Jan. 2019, Art. no. 1800544.

[3] J. Grollier, D. Querlioz, K. Y. Camsari, K. Everschor-Sitte, S. Fukami, and M. D. Stiles, "Neuromorphic spintronics is the study of devices that mimic the behavior of neurons using spin-based phenomena," *Nature Electron.*, vol. 3, no. 7, pp. 360–370, Mar. 2020.

[4] K. Byun et al., "Recent advances in synaptic nonvolatile memory devices and compensating architectural and algorithmic methods toward fully integrated neuromorphic chips," *Adv. Mater. Technol.*, Oct. 2022, Art. no. 2200884.

[5] Z.-X. Li, X.-Y. Geng, J. Wang, and F. Zhuge, "Emerging artificial neuron devices for probabilistic computing," *Frontiers Neurosci.*, vol. 15, Aug. 2021, Art. no. 717947.

[6] J.-P. Wang et al., "A pathway to enable exponential scaling for the beyond-CMOS era: Invited," in *Proc. Des. Autom. Conf. (DAC)*, vol. 16. New York, NY, USA, Jun. 2017, pp. 1–6.

[7] J.-P. Wang, "Special topic on spintronic devices for energy-efficient computing," *IEEE J. Explor. Solid-State Comput. Devices Circuits*, vol. 8, pp. 2–3, 2022.

[8] W. H. Choi et al., "A magnetic tunnel junction based true random number generator with conditional perturb and real-time output probability tracking," in *IEDM Tech. Dig.*, San Francisco, CA, USA, Dec. 2014, pp. 12.5.1–12.5.4.

[9] A. Fukushima et al., "Spin dice: A scalable truly random number generator based on spintronics," *Appl. Phys. Exp.*, vol. 7, no. 8, Jul. 2014, Art. no. 083001.

[10] D. Vodenicarevic et al., "Low-energy truly random number generation with superparamagnetic tunnel junctions for unconventional computing," *Phys. Rev. Appl.*, vol. 8, no. 5, Nov. 2017, Art. no. 054045.

[11] Z. Fu et al., "An overview of spintronic true random number generator," *Frontiers Phys.*, vol. 9, Apr. 2021, Art. no. 638207.

[12] W. Gross, N. Onizawa, K. Matsumiya, and T. Hanyu, "Application of stochastic computing in brainware," *Nonlinear Theory Appl.*, vol. 9, no. 4, pp. 406–422, 2018.

[13] P. Li, W. Qian, D. J. Lilja, K. Bazargan, and M. D. Riedel, "Case studies of logical computation on stochastic bit streams," in *Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation—PATMOS* (Lecture Notes in Computer Science), vol. 7606. Heidelberg, Germany: Springer, 2013, pp. 235–244, doi: 10.1007/978-3-642-36157-9_24.

[14] P. Li, D. J. Lilja, W. Qian, K. Bazargan, and M. D. Riedel, "Computation on stochastic bit streams digital image processing case studies," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 3, pp. 449–462, Mar. 2014.

[15] Y. Liu and K. K. Parhi, "Computing hyperbolic tangent and sigmoid functions using stochastic logic," in *Proc. 50th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2016, pp. 1580–1585.

[16] A. Alaghi, W. Qian, and J. P. Hayes, "The promise and challenge of stochastic computing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 8, pp. 1515–1531, Aug. 2018.

[17] W. Qian, X. Li, M. D. Riedel, K. Bazargan, and D. J. Lilja, "An architecture for fault-tolerant computation with stochastic logic," *IEEE Trans. Comput.*, vol. 60, no. 1, pp. 93–105, Jan. 2011.

[18] P. Li and D. J. Lilja, "A low power fault-tolerant architecture for the kernel density estimation based image segmentation algorithm," in *Proc. IEEE Int. Conf. Appl.-Specific Syst. Archit. Process.*, Sep. 2011, pp. 161–168.

[19] M. H. Najafi and M. E. Salehi, "A fast fault-tolerant architecture for Sauvola local image thresholding algorithm using stochastic computing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 2, pp. 808–812, Feb. 2016.

[20] M. W. Daniels, A. Madhavan, P. Talatchian, A. Mizrahi, and M. D. Stiles, "Energy-efficient stochastic computing with superparamagnetic tunnel junctions," *Phys. Rev. Appl.*, vol. 13, no. 3, Mar. 2020, Art. no. 034016.

[21] Y. Shao et al., "Implementation of artificial neural networks using magnetoresistive random-access memory-based stochastic computing units," *IEEE Magn. Lett.*, vol. 12, 2021, Art. no. 4501005.

[22] J.-P. Wang and J. D. Harms, "General structure for computational random access memory," U.S. Patent 9 224 447, B2, Dec. 29, 2015.

[23] Z. Chowdhury et al., "Efficient in-memory processing using spintronics," *IEEE Comput. Archit. Lett.*, vol. 17, no. 1, pp. 42–46, Jan./Jun. 2018.

[24] Y. Lv et al., "Experimental demonstration of magnetic tunnel junction-based computational random-access memory," to be published.

[25] B. R. Zink et al., "A stochastic computing scheme of embedding random bit generation and processing in computational random access memory (SC-CRAM)," *IEEE J. Explor. Solid-State Comput. Devices Circuits*, vol. 9, pp. 29–37, 2023.

[26] D. C. Worledge, "Spin-Transfer-Torque MRAM: The next revolution in memory," in *Proc. IEEE Int. Memory Workshop (IMW)*, Dresden, Germany, May 2022, pp. 1–4.

[27] J. M. Slaughter et al., "Technology for reliable spin-torque MRAM products," in *IEDM Tech. Dig.*, San Francisco, CA, USA, Dec. 2016, pp. 21.5.1–21.5.4.

[28] J. G. Alzate et al., "2 MB array-level demonstration of STT-MRAM process and performance towards L4 Cache applications," in *IEDM Tech. Dig.*, San Francisco, CA, USA, Dec. 2019, pp. 2.4.1–2.4.4.

[29] C.-H. Chen et al., "Reliability and magnetic immunity of reflow-capable embedded STT-MRAM in 16 nm FinFET CMOS process," in *Proc. Symp. VLSI Technol.*, Kyoto, Japan, Jun. 2021, pp. 1–2.

[30] T. Y. Lee et al., "World-most energy-efficient MRAM technology for non-volatile RAM applications," in *IEDM Tech. Dig.*, San Francisco, CA, USA, Dec. 2022, pp. 10.7.1–10.7.4.

[31] I. M. Miron et al., "Perpendicular switching of a single ferromagnetic layer induced by in-plane current injection," *Nature*, vol. 476, no. 7359, pp. 189–193, Aug. 2011.

[32] I. Ahmed, Z. Zhao, M. G. Mankalale, S. S. Sapatnekar, J.-P. Wang, and C. H. Kim, "A comparative study between spin-transfer-torque and spin-Hall-effect switching mechanisms in PMTJ using SPICE," *IEEE J. Explor. Solid-State Comput. Devices Circuits*, vol. 3, pp. 74–82, 2017.

[33] C. Bi, N. Sato, and S. X. Wang, "Spin-orbit torque magnetoresistive random-access memory (SOT-MRAM)," in *Advances in Non-Volatile Memory and Storage Technology*. Sawston, U.K.: Woodhead, Jun. 2019, pp. 203–235.

[34] X. Han, X. Wang, C. Wan, G. Yu, and X. Lv, "Spin-orbit torques: Materials, physics, and devices," *Appl. Phys. Lett.*, vol. 118, no. 12, Mar. 2021, Art. no. 120502.

[35] Q. Shao et al., "Roadmap of spin–orbit torques," *IEEE Trans. Magn.*, vol. 57, no. 7, Jul. 2021, Art. no. 800439.

[36] K. Y. Camsari et al., "Stochastic *p*-bits for invertible logic," *Phys. Rev. X*, vol. 7, Jul. 2017, Art. no. 031014.

[37] W. A. Borders, A. Z. Pervaiz, S. Fukami, K. Y. Camsari, H. Ohno, and S. Datta, "Integer factorization using stochastic magnetic tunnel junctions," *Nature*, vol. 573, no. 7774, pp. 390–393, Sep. 2019.

[38] Y. Lv, R. P. Bloom, and J.-P. Wang, "Experimental demonstration of probabilistic spin logic by magnetic tunnel junctions," *IEEE Magn. Lett.*, vol. 10, 2019, Art. no. 4510905.

[39] O. Hassan, S. Datta, and K. Y. Camsari, "Quantitative evaluation of hardware binary stochastic neurons," *Phys. Rev. Appl.*, vol. 15, no. 6, Jun. 2021, Art. no. 064046.

[40] X. Jia, J. Yang, Z. Wang, Y. Chen, H. H. Li, and W. Zhao, "Spintronics based stochastic computing for efficient Bayesian inference system," in *Proc. 23rd Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2018, pp. 580–585.

[41] M. Zabihi, Z. I. Chowdhury, Z. Zhao, U. R. Karpuzcu, J.-P. Wang, and S. S. Sapatnekar, "In-memory processing on the spintronic CRAM: From hardware design to application mapping," *IEEE Trans. Comput.*, vol. 68, no. 8, pp. 1159–1173, Aug. 2019.

[42] G. Jan et al., "Demonstration of fully functional 8 MB perpendicular STT-MRAM chips with sub-5 ns writing for non-volatile embedded memories," in *Proc. Symp. VLSI Technol. VLSI-Technol., Dig. Tech. Papers*, Honolulu, HI, USA, Jun. 2014, pp. 1–2.

[43] J. Z. Sun, "Spin-transfer torque switched magnetic tunnel junction for memory technologies," *J. Magn. Magn. Mater.*, vol. 559, Oct. 2022, Art. no. 169479.

[44] H. Zhao et al., "Low writing energy and sub nanosecond spin torque transfer switching of in-plane magnetic tunnel junction for spin torque transfer random access memory," *J. Appl. Phys.*, vol. 109, no. 7, Apr. 2011, Art. no. 07C720.

[45] C. Safranski et al., "Reliable sub-nanosecond switching in magnetic tunnel junctions for MRAM applications," *IEEE Trans. Electron Devices*, vol. 69, no. 12, pp. 7180–7183, Dec. 2022.

[46] H. Maehara et al., "Tunnel magnetoresistance above 170% and resistance-area product of $1\Omega(\mu\text{m})^2$ attained by in situ annealing of ultra-thin MgO tunnel barrier," *Appl. Phys. Exp.*, vol. 4, no. 3, Mar. 2011, Art. no. 033002.

[47] K. C. Chun, H. Zhao, J. D. Harms, T.-H. Kim, J.-P. Wang, and C. H. Kim, "A scaling roadmap and performance evaluation of in-plane and perpendicular MTJ based STT-MRAMS for high-density cache memory," *IEEE J. Solid-State Circuits*, vol. 48, no. 2, pp. 598–610, Feb. 2013.

[48] N. Sato, F. Xue, R. M. White, C. Bi, and S. X. Wang, "Two-terminal spin-orbit torque magnetoresistive random access memory," *Nature Electron.*, vol. 1, no. 9, pp. 508–511, Sep. 2018.

[49] S. Fukami, T. Anekawa, C. Zhang, and H. Ohno, "A spin–orbit torque switching scheme with collinear magnetic easy axis and current configuration," *Nature Nanotechnol.*, vol. 11, no. 7, pp. 621–625, Mar. 2016.

[50] M. Cubukcu et al., "Ultra-fast perpendicular spin–orbit torque MRAM," *IEEE Trans. Magn.*, vol. 54, no. 4, Apr. 2018, Art. no. 9300204.

[51] Z. Zhao, A. K. Smith, M. Jamali, and J. Wang, "External-field-free spin Hall switching of perpendicular magnetic nanopillar with a dipole-coupled composite structure," *Adv. Electron. Mater.*, vol. 6, no. 5, May 2020, Art. no. 1901368.

[52] E. Grimaldi et al., "Single-shot dynamics of spin–orbit torque and spin transfer torque switching in three-terminal magnetic tunnel junctions," *Nat. Nanotech.*, vol. 15, pp. 111–117, Feb. 2020.

[53] K. Garello et al., "Manufacturable 300 mm platform solution for field-free switching SOT-MRAM," in *Proc. Symp. VLSI Technol.*, Kyoto, Japan, Jun. 2019, pp. T194–T195.

[54] P. Kumar and A. Naeemi, "Benchmarking of spin–orbit torque vs spin-transfer torque devices," *Appl. Phys. Lett.*, vol. 121, no. 11, Sep. 2022, Art. no. 112406.

[55] L. Lopez-Diaz, L. Torres, and E. Moro, "Transition from ferromagnetism to superparamagnetism on the nanosecond time scale," *Phys. Rev. B, Condens. Matter.*, vol. 65, no. 22, May 2002, Art. no. 224406.

[56] H. H. Huy, Z. Ruixian, T. Shirokura, S. Takahashi, Y. Hirayama, and P. N. Hai, "Integration of a BiSb topological insulator and CoFeB/MGO with perpendicular magnetic anisotropy using an oxide interfacial layer for ultralow power SOT-MRAM cache memory," *IEEE Trans. Magn.*, early access, May 11, 2023, doi: 10.1109/TMAG.2023.3275171.

[57] J. Song et al., "Evaluation of operating margin and switching probability of voltage-controlled magnetic anisotropy magnetic tunnel junctions," *IEEE J. Explor. Solid-State Comput. Devices Circuits*, vol. 4, pp. 76–84, 2018.

[58] D. Lyu, D. Zhang, D. B. Gopman, Y. Lv, O. J. Benally, and J.-P. Wang, "Ferromagnetic resonance and magnetization switching characteristics of perpendicular magnetic tunnel junctions with synthetic antiferromagnetic free layers," *Appl. Phys. Lett.*, vol. 120, no. 1, Jan. 2022, Art. no. 012404.

[59] D. Zhang et al., "Bipolar electric-field switching of perpendicular magnetic tunnel junctions through voltage-controlled exchange coupling," *Nano Lett.*, vol. 22, no. 2, pp. 622–629, Jan. 2022.

[60] B. R. Zink et al., "Ultralow current switching of synthetic-antiferromagnetic magnetic tunnel junctions via electric-field assisted by spin-orbit torque," *Adv. Electron. Mater.*, vol. 8, no. 10, Jul. 2022, Art. no. 2200382.