SIMBA: A Skyrmionic In-Memory Binary Neural Network Accelerator

Venkata Pavan Kumar Miriyala, Student Member, IEEE, Kale Rahul Vishwanath, Xuanyao Fong, Member, IEEE
Department of Electrical and Computer Engineering, National University of Singapore, Singapore, 117583.

Magnetic skyrmions are emerging as potential candidates for next generation non-volatile memories. In this paper, we propose an in-memory binary neural network (BNN) accelerator based on the non-volatile skyrmionic memory, which we call as SIMBA. SIMBA consumes 26.7 mJ of energy and 2.7 ms of latency when running an inference on a VGG-like BNN. Furthermore, we demonstrate improvements in the performance of SIMBA by optimizing material parameters such as saturation magnetization, anisotropic energy and damping ratio. Finally, we show that the inference accuracy of BNNs is robust against the possible stochastic behavior of SIMBA (88.5% ±1%).

Index Terms—Binary neural networks, magnetic skyrmions, in-memory computing, spintronics, spin hall effect and spin-transfer torque nano oscillators.

I. INTRODUCTION

Spintronic memories utilizing magnetic skyrmions have recently been the subject of great interest [1]–[6]. This is due to scalability of skyrmions down to 2 nm [7]–[9], the ease of electrical manipulation [1], [10] and their robustness and stability due to topological protection of skyrmions [11], [12]. Existing skyrmionic memory proposals replace the magnetic domain wall devices in racetrack memories [13], [14] with their skyrmionic counterparts. However, micro-magnetic simulations recently show that when a spin-torque nano-oscillator (STNO) [15] is used to generate spin waves (SWs) in a ferromagnetic film, the presence of a skyrmion in the film can influence its steady state dynamics [16]. A skyrmion is topologically protected and hence, does not allow the SWs to pass through it. Moreover, the skyrmion backscatters the SWs as if it is reradiating the SWs. The interference between SWs emitted by the STNO and SWs backscattered by the skyrmion, which are well-described by wave equations, can affect the steady state dynamics of the STNO—sufficiently strong constructive interference between SWs can lead to nucleation of a skyrmion in the STNO [16].

In this paper, we propose a novel way of using the SW mediated interactions between magnetic skyrmions and STNOs to augment the skyrmionic memory with compute capabilities and implement an ultra-low power non-volatile in-memory computing engine (NVIMCE) using the presence and absence of a skyrmion as a state variable. To the best of our knowledge, this is the first skyrmionic in-memory computing proposal. Moreover, unlike other skyrmionic logic devices that have been proposed in the literature [17]–[19], the issues arising from the transverse motion of skyrmions [11] are avoided in our proposed device because logic operations do not rely on skyrmion motion.

On the other hand, the NVIMCE, which saves on energy associated with data storage and movement is a crucial design technique for accelerating neural network algorithms [20]–[22]. The NVIMCEs can be powered down to achieve near-zero standby power consumption when the device is in sleep mode. When the device wakes up, data in the NVIMCE is updated and processed in-situ, which saves on latency and energy consumed to transfer data between memory and processing elements in the conventional von Neumann architecture.

In this paper, we also propose a skyrmionic NVIMCE (abbreviated as SIMC) based hardware accelerator for binary neural networks (BNNs) [23], which we call as SIMBA. A hybrid simulation framework consisting of device, circuit and architecture-level simulation tools is developed to design, analyze and evaluate SIMBA. Our simulation results show that high energy efficiency and fast inference operations are achieved in SIMBA due to reduced data movement. Further, we have also demonstrated improvements in performance of SIMBA by optimizing the material parameters.

The rest of this paper is structured as follows. The SW mediated interactions between skyrmions and STNOs, and the skyrmionic logic gates will be first introduced in Section II. The hardware architectures of our proposed SIMC and SIMBA are then presented in Section III and Section IV, respectively. Results and discussion on proposed devices, circuits and architectures are then presented in Section V. Finally, Section VI concludes this paper.

II. SKYRMIONIC LOGIC COMPUTATION USING STNOs

For completeness sake, the underlying device physics of the non-volatile skyrmionic devices with which SIMBA is implemented is briefly presented in the following subsection. Further details of the device-level simulations are discussed in Section V.

A. Interactions between Skyrmions and STNOs

Let us first consider the device structure shown in Fig. 1(a), which consists of an oxide/ferromagnetic free layer (FL)/heavy metal (HM) trilayer structure in which the magnetization of FL can be manipulated. Two magnetically pinned layers (PL)
are placed on top of the oxide layer to form the magnetic tunnel junctions (MTJs) labeled as R-1 and R-2.

Now, consider the injection of current into the FL through R-2 (by applying a voltage, $V_2$, across the corresponding MTJ) in the absence of any skyrmions in the FL layer. Results of the micromagnetic simulation in MuMax3 [26], shown in Fig. 1(b), demonstrate that current-induced spin-transfer torque (STT) [27] can cause the local magnetization in R-2 to oscillate continuously like an STNO. Due to the interaction between various magnetization energies (exchange, demagnetizing, anisotropies, etc.), the oscillations in R-2 radiate outward through the rest of FL in the form of SWs as Fig. 1(c) shows. Hence, applying a voltage across the MTJ in R-2 in this manner activates the STNO at R-2.

However, if the current-induced STT on the FL is sufficiently strong, the local magnetization of the FL may switch to a skyrmionic magnetization state instead of continuously oscillating, as shown in Fig. 1(b). A skyrmion acts as a scatterer of SWs due to its topological protection. Hence, if a skyrmion is present at R-1 (which may be nucleated by applying a voltage, $V_1$, across the corresponding MTJ) before the STNO at R-2 is activated, the SWs emitted by the STNO at R-2 ($SW_{\text{emitted}}$) will be backscattered by the skyrmion at R-1. The SWs backscattered by the skyrmion in R-1 ($SW_{\text{scattered}}$) interfere with $SW_{\text{emitted}}$ and can affect the dynamics of the STNO at R-2. Results of MuMax3 simulations show that STNO oscillations are amplified (Fig. 1(e)) and attenuated (Fig. 1(f)) when the interference is constructive and destructive, respectively. This phenomenon can be modeled using the following wave equations:

$$SW_{\text{emitted}} \propto A_1 e^{i\frac{\pi}{\lambda}x-\omega t \cos(kx-\omega t)} \quad (1)$$
$$SW_{\text{scattered}} \propto B_1 e^{i\frac{\pi}{\lambda}x-\omega t - \frac{\Phi}{\pi} \cos(kx-\omega t - \Phi)} \quad (2)$$
$$SW_{\text{total}} = SW_{\text{emitted}} + SW_{\text{scattered}} \quad (3)$$

where $A_1$ and $B_1$ are the wave amplitudes, $k = 2\pi/\lambda$ is the wave vector, $\omega$ is the wave frequency, $k_f$ and $\Phi_f$ are fitting constants, $\Phi = 2k(d-R)$ is the phase shift.

---

**TABLE I**

**SIMULATION PARAMETERS**

| Quantity                          | Value             |
|-----------------------------------|-------------------|
| Size of FL                        | $300\times300\times0.4$ nm $^3$ |
| Mesh size                         | $2$ mm$\times2$ nm$\times0.4$ nm     |
| Diameter of $A$, $B$, $O_1$, and $O_2$ | $40$ nm          |
| Saturation Magnetization           | $580\times10^3$ A/m |
| Exchange constant                  | $1.5\times10^{13}$ J/m |
| Anisotropic energy density         | $0.8\times10^6$ J/m$^2$ |
| Interfacial Dzyaloshinskii–Moriya interaction constant | $3.5$ mJ/m$^2$ |
| Damping ratio ($\alpha$)           | $0.03$            |
| Polarization ($P$)                 | $0.56$            |
| Spin torque efficiency ($\Lambda$) | $1$              |
| Field-free torque coefficient ($\epsilon$) | $10^{-7}$  |
| Spin hall angle ($\theta_{\text{sh}}$) | $0.1$           |
| Resistance area product            | $1.63$ $\Omega$ $\mu$ m$^2$ |
| Tunneling Magnetoresistance ratio  | $125\%$          |
| HM resistivity                     | $2$ $\mu$ $\Omega$ m |

---

**Fig. 1.** The cross-sectional view of the device structure used to demonstrate the interactions between skyrmions and STNOs. $V_1$ generates the skyrmion in region R-1 and $V_2$ causes the local magnetization in region R-2 to precess continuously, shown in (b) and (c). The precession leads to the propagation of SWs outward from R-2 as shown in (c). (d) The local magnetization in R-2 when no skyrmion is present in R-1. (e) and (f) show the local magnetization in R-2 when a skyrmion is at $d = 100$ nm and $d = 125$ nm, respectively. The stabilized negative in-plane magnetization in (e) denotes the formation of new skyrmion in R-2.
of skyrmionic logic gates that utilize the effects discussed in this sub-section.

B. Spin-wave-based Skyrmionic Logic

The structure of our proposed skyrmionic logic gate, shown in Fig. 2(a), contains four point-contact regions (A, B, O₁ and O₂). The regions A and B are used for storing inputs, whereas regions O₁ and O₂ are used during computation. The presence (absence) of the skyrmions in these regions denote the bit ‘1’ (‘0’) and can be electrically sensed as the resistance of corresponding MTJs [23]. Fig. 2(b) shows the time evolution of FL magnetization when A and B are programmed to be ‘0’ and ‘1’, respectively, whereas Fig. 2(c) shows the time evolution of FL magnetization when A and B are both programmed to be ‘1’.

Computation on inputs stored at A and B is performed by using O₁ and O₂ as STNOS. O₁ and O₂ are placed at distance, \( d = 125 \text{ nm} \) apart from each other and at a distance, \( d = 150 \text{ nm} \) from both the input bits, A and B. The material parameters assumed in this work lead to \( \lambda \approx 100 \text{ nm} \). In this geometry, if the STNO at O₁ is activated, the presence of a skyrmion at either A or B leads to constructive interference between SWs emitted and SWs scattered in O₁ and amplify the oscillations in the STNO, resulting in the formation of a skyrmion at O₁. Furthermore, if skyrmions are present in both A and B, the amplification in STNO oscillations would be large and a skyrmion forms at O₁ within relatively short time delays. Due to symmetry, the same effect is observed when the STNO in O₂ is activated. However, in the presence of a skyrmion at O₁, SWs emitted by O₂ and SWs scattered by the skyrmion O₁ interfere destructively and oscillations at O₂ become attenuated.

Consider only the STNO at O₁ with no skyrmions in O₂. If O₁ is activated for 2.5 ns, a new skyrmion will only form in O₁ if skyrmions are present in both A and B (see Fig. 2(d) from \( t_{c1} = 0 \text{ ns} \) to 2.5 ns). Otherwise, no skyrmion will form in O₁ (see Fig. 2(e)-(f) from \( t_{c1} = 0 \text{ ns} \) to 2.5 ns). Hence, the output stored at O₁ may be expressed using the Boolean expression \( O₁ = A \cdot B \) (i.e., \( A \ AND \ B \)). If only one of A or B has a skyrmion instead, the pulse width of \( V_{O1} \) needs to be 5 ns in order for a skyrmion to form in O₁. Due to the symmetry between O₁ and O₂, same effect can be observed at O₂ as well. We demonstrate the latter case by activating O₂ for 5 ns (see Fig. 2(e)-(f) from \( t_{c2} = 0 \text{ ns} \) to 2.5 ns). The output stored at O₂ may be expressed using the Boolean expression \( O₂ = A + B \) (i.e., \( A \ OR \ B \)). Hence, A \・ B and A + B operations can be performed by ensuring that O₂ and O₁ do not have any skyrmion before activating the STNO with the required pulse width at the desired location.

The XOR operation between A and B (A \( \oplus \) B) needs to be performed in two phases. First, we note that the Boolean expression for the XOR operation may be written as

\[
A \oplus B = (A \cdot B) + (\overline{A} \cdot \overline{B}) = (A + B) \cdot (\overline{A} + \overline{B})
\]

Hence, in the first phase, A \・ B operation is performed and stored at O₁. In the second phase, the voltage pulse width is given to the STNO in O₂ as if A + B operation is being
Fig. 3. (a) The device structure of a bit-cell in the proposed SIMC with four access transistors (T₁, T₂, T₃ and T₄). Regions, A, B, O₁ and O₂ are connected to select line, CSL₁ at the bottom (not shown in (a) for better illustration). (b) Electrical model of the bit-cell. The regions A, B, O₁ and O₂ are assumed as simple MTJs with resistances, Rₐ, Rₐ, Rₒ₁ and Rₒ₂. Therefore, the proposed bit-cell can be seen as a simple circuit block with four 1-transistor 1-resistances. (c) Layout schematic of a bit-cell obtained using the NCSU FreePDK45 [28]. (d) Array-level design of SIMC with 3×3 bit-cells. Row select lines, RSL₁, RSL₂ are represented as yellow-colored line and dotted-lines, respectively. Similarly, CSL₃, CSL₄ are represented as red-colored line and dotted-lines, respectively. The column select lines, CSL₁, CSL₂ and CSL₃ are represented as green, blue and black-colored lines, respectively. (e) An example layout of this 3×3 bit-cell array. The rectangular box in (e) marks a single bit-cell. Perform the operation of O₂ depends on the output stored in O₁, as discussed earlier. Due to destructive interference between \( SW_{\text{emitted}} \) by O₂ and \( SW_{\text{scattered}} \) by the skyrmion O₁, the result at O₂ at the end of the two-phase operation is \((A + B) \cdot (\overline{A} \cdot \overline{B}) = A \oplus B\). For instance, if no skyrmions are present in both A and B (i.e., \( A = B = '0' \)), magnetization at O₁ during the first phase and magnetization at O₂ during the second phase would only oscillate continuously and no skyrmion will form at O₂, which is in agreement with the Eq. 4. If a skyrmion is present at either A or B (i.e., \( A \oplus B = '1' \)), the \( SW_{\text{emitted}} \) by O₂ and \( SW_{\text{scattered}} \) by the skyrmion interfere constructively at O₂. Consequently, a skyrmion will form at O₂ in agreement with the Eq. 4. Finally, if skyrmions are present at both A and B (i.e., \( A = B = '1' \)), they would also cause the SWs to interfere constructively at O₂. However, the interference between SWs at O₂ is not as strong in the presence of SWs backscattered by the skyrmion that is nucleated at O₁ during the first phase. Consequently, a skyrmion will not form in O₂ when skyrmions are present in both A and B, which satisfies the Eq. 4. The XOR operation between A and B is operated in micromagnetic simulations shown in Fig. 2 (d)-(f). As shown in Fig. 2 (d), when A and B are ‘1’, O₂ is ‘0’. If only one of A or B = ‘1’, O₂ = ‘1’ (Fig. 2 (e)). Finally, when both A and B are ‘0’, O₂ is ‘0’ (Fig. 2 (f)). Note that if any skyrmion in the FL is to be annihilated in any operation, then all skyrmions in the FL would have to be annihilated. Skyrmions are then nucleated at the desired locations after the annihilation (reset) process. The reset process is performed by passing current in the plane of the HM layer. A spin-orbit torque (SOT) induced by the current flow is exerted on the FL magnetization [29]. If this torque is sufficiently strong, skyrmions get pushed off the boundaries of the FL. Simulation results in Fig. 2 (g) show the scenario where skyrmions are being pushed towards the left boundary of the FL and finally getting annihilated.

III. SKYRMIONIC IN-MEMORY COMPUTING ENGINE (SIMC)

The device structure presented in Section II-B is used to implement a multi-level bit-cell (see Fig. 3) in our proposed SIMC. Since our proposed device is non-volatile, it may also be used as a memory device that stores data at A and B. As discussed in Section II-B, O₁ and O₂ may be used for performing logic operations on the stored data.
A. Multi-level bit cell

Each of the four regions of our proposed device is connected to an access transistor, as Fig. 3(a) shows. Regions \( A \) and \( B \) are connected to column select line, \( CSL_1 \) at the bottom, and to column select line, \( CSL_2 \) through transistors, \( T_1 \) and \( T_2 \) (which are controlled by row select lines, \( RSL_1 \) and \( RSL_2 \), respectively). Similarly, regions \( O_1 \) and \( O_2 \) are connected to column select line, \( CSL_3 \) at the bottom, and to column select line, \( CSL_4 \) through transistors, \( T_3 \) and \( T_4 \) (which are controlled by row select lines, \( RSL_3 \) and \( RSL_4 \), respectively). For better illustration, connections to \( CSL_1 \) at the bottom are omitted from Fig. 3(a). However, all the connections are depicted in Fig. 3(b), which represents the circuit model of our proposed bit-cell.

The write operation to the bit-cell occurs as follows. When bit ‘1’ is to be written into \( A \) (i.e., nucleate a skyrmion in \( A \)), \( T_1 \) is turned ON to pass current from \( CSL_2 \) to \( CSL_1 \) through \( A \). To write bit ‘0’ into \( A \) (i.e., do not nucleate a skyrmion), \( T_1 \) is turned OFF and no current passes through \( A \). Similarly, to write bit ‘1’ into \( B \), \( T_2 \) is turned ON and the current is passed from \( CSL_2 \) to \( CSL_1 \) through \( B \). To write bit ‘0’ into \( B \), \( T_2 \) is turned OFF and no current is passed through \( B \).

For performing the logic computations on data stored in \( A \) and \( B \), \( O_1 \) and \( O_2 \) are utilized as STNOs. To activate the STNO in \( O_1 \), \( T_3 \) is turned ON and the current is passed from \( CSL_3 \) to \( CSL_1 \) through \( O_1 \). Otherwise, \( T_3 \) is turned OFF and no current is passed through \( O_1 \). Similarly, to activate the STNO in \( O_2 \), \( T_4 \) is turned ON and the current is passed from \( CSL_4 \) to \( CSL_1 \) through \( O_2 \). Otherwise, \( T_4 \) is turned OFF and no current is passed through \( O_2 \).

Note that, as mentioned earlier in Section 4.3, if any input bit needs to switch from ‘1’ to ‘0’ or if the logic computation needs to be re-evaluated, the reset current must first be passed in-plane through the HM to annihilate the skyrmions, followed by nucleation of skyrmions at the desired locations. This process may require the read operation of the data stored in \( A \) and \( B \). The data stored in \( A \) and \( B \), and the output of computation stored in \( O_1 \) and \( O_2 \), may be sensed as the resistance of the MTJ at the corresponding regions. A current-sensing scheme with conventional sense amplifiers may be used to perform the read operation [20-23]. Note that the voltages across the MTJs during read operations need to be optimized to avoid the read-failure or read-disturb errors.

B. Hardware architecture of SIMC

The proposed multi-level bit-cells are arranged into regular arrays to construct the SIMC. The layout for an individual bit-cell is depicted in Fig. 3(c). The \( CSL_1 \) and \( CSL_2 \) are routed on metal-2 whereas \( CSL_3 \) is routed on metal-3. The bit-cells are arranged into a large array and a 3×3 area of the bit-cell array is shown in Fig. 3(d). The light orange-colored region in Fig. 3(d) depicts the HM layer that is shared among the bit-cells whereas the blue-colored square regions depict the magnetic device structures. The colored-lines and dotted-lines represent the various control lines to the bit-cells. Note that separate row and column control circuitry drive the control lines as shown in Fig. 3(d). Furthermore, a constant current driver supplies the reset current to the HM during reset operations. The layout of a 3×3 area of the bit cell array is shown in Fig. 3(e).

IV. HARDWARE ARCHITECTURE OF SIMBA

The SIMC proposed in Section III can be used to implement the hardware accelerator for binary neural network (BNN) algorithms. We first introduce BNNs and show the operations that are needed to be performed. The mapping of these operations to dedicated hardware are then discussed before showing how they are put together to give the hardware implementation of SIMBA.

A. Binary Neural Networks

BNNs are a class of deep learning algorithms that perform inference operations using binarized inputs and weights [23]. The binarized pixel data of images are first fed to the binary convolution (BinConv) layer in the form of input feature maps (IFs) with size \((W, H)\) (see Fig. 4(a)). In BinConv layer, the data in IFs is convolved with binary weights stored in filters (FIFs) of size \((kw, kh)\). Mathematically,

\[
f(i, j) = \text{popcount}(IF(W, H) \ XNOR \ F(KW(kw, kh)))
\]

where \((i, j)\) are the array indices of output feature map (OF). When given a string of bits, the \text{popcount} operation counts the number of ‘1s’ in the string. As expressed in Eq. 5, convolution on binary data is achieved by performing a sequence of bitwise \text{XNOR}, \text{popcount} and integer \text{comparator} operations.

The dimensions of resultant OFs from the BinConv layer are compressed in the maxpooling (Maxpool) layer. In the Maxpool layer, the maximum pixel value in a selected group of pixel values is estimated and forwarded to the subsequent layers. Due to the binarization of the data, the results of the Maxpool layer can be obtained using bit-wise \text{OR} operations. As shown in Fig. 4(a), the compressed OFs are propagated through several BinConv and Maxpool layers. The final OFs are then given as inputs to the fully-connected (FullyConn) layer to obtain the inference/classification results. FullyConn layer is a feed-forward multi-perceptron network, where the weighted sum of inputs is compared with a threshold in each perceptron. Due to the binarization of the network, computations in this layer can also be executed using bit-wise \text{AND} operations followed by \text{popcount} and \text{comparator}.

B. Hardware Implementation of SIMBA

The proposed hardware architecture of SIMBA is shown in Fig. 4(b). It is assumed that training of the BNN is performed offline and the configuration of the trained BNN then is loaded into the SIMBA hardware, which performs the test/inference operations. The hardware architecture of SIMBA consists of hardware dedicated to the execution of operations for the BinConv, Maxpool, and FullyConn layers. Note from Eq. 5 that the BNN algorithm requires the computation of bit-wise \text{XNOR} followed by \text{popcount}, which counts the number of
Pre-processed filters with binary weights

Input image

Binary Convolution
Max Pooling
Fully Connected

Dog = 0.65
Cat = 0.2
Monkey = 0.1
Rabbit = 0.05

∙∙∙∙

bits that match each other. The same result may be obtained by counting the number of bit mismatches and subtracting the result from the total number of bit comparisons, which is known a priori. Hence, the result of \( f(i,j) \) in Eq. 5 can also be computed using XOR operations (which returns whether the input bits mismatch) followed by a suitably implemented popcount circuit. For executing BinConv layers in SIMBA, SIMC units capable of 1 KB data storage perform the XOR computations, which are read out and passed to the popcount circuitry.

Since our SIMC units perform the XOR operation, \( OF(i,j) \) may be calculated using one of two methods. In the first method, the conventional popcount is used to count the number of ‘1’s returned by the XOR operations. \( OF(i,j) \) is set to ‘1’ if the result exceeds the threshold and ‘0’ otherwise. In the second method, a popcount circuit that counts the number of ‘0’s instead of ‘1’ is given the result of the XOR operations. \( OF(i,j) \) is set to ‘1’ if the result exceeds the threshold and ‘0’ otherwise. In this paper, we implemented the former method to compute \( OF(i,j) \). Fig. 4(c) shows our proposed popcount circuit, which consists of \( n \) spin-orbit torque MTJs (SOT-MTJs), each of which receives as input the output of each XOR operation. All the popcount SOT-MTJs are initially set to low resistance state (i.e., state ‘0’) by providing the negative reset voltage \((-V_P < 0 \text{ V})\) to the HMs of SOT-MTJs. The comparator SOT-MTJ is initialized to the high resistance state (‘1’) by passing current through the HM of comparator SOT-MTJ. When the output of \( i \)-th XOR operation is ‘0’ (‘1’), \( P_i \) is set as 0 V (+\( V_P > 0 \text{ V} \)). Positive voltage on \( P_i \) causes current to flow through the corresponding HM, which switches the resistance state of the corresponding SOT-MTJ from low (i.e., state ‘0’) to high (i.e., state ‘1’).
After all XOR results have been processed, the transistor, \( T_{\text{comp}} \), is turned ON to pass positive current from \( CL_1 \) to the ground. The resistance states of \( \text{popcount} \) MTJs determine the magnitude of \( I_{\text{comp}} \) passing through the comparator SOT-MTJ (see Fig. 2 (c)). If the magnitude of \( I_{\text{comp}} \) exceeds the threshold switching current, the resistance state of the comparator SOT-MTJ changes from high to low (‘1’ to ‘0’). The output result can be read by turning ON the transistor, \( T_{\text{read}} \) (by setting the read voltage \( V_{\text{read}} \)) and sensing the current flowing into the comparator SOT-MTJ through \( CL_2 \).

For executing the Maxpool layer, as depicted in Fig. 4 (b), the same SIMC unit hardware design may be used. However, the SIMC units for Maxpool layers are configured to perform only OR operations. Furthermore, \( \text{popcount} \) and comparator circuitry are not needed. Also shown in Fig. 4 (b), the hardware used for executing the computations in the FullyConn layer is similar to the hardware for executing the BinConv layer. However, the SIMC units for the FullyConn layer are configured to perform AND operations only.

V. RESULTS AND DISCUSSION

We will now focus on the design and evaluation of the devices, circuits and hardware architecture that constitute SIMBA. In order to study the interactions between the different levels of design abstraction, we developed a hybrid mixed-mode devices-to-hardware architecture simulation framework. We will first present the simulation framework used for this work before discussing the simulation results for the devices, circuits and hardware architecture of SIMBA.

A. Modeling and Simulation Framework

Fig. 5 shows the flow-chart of our simulation framework. At the device-level, we used the MuMax3 micromagnetic simulator [26] to study the magnetization dynamics and the interactions between skyrmions and STNOs. The magnetic device and material parameters we assumed in our simulations are extracted from experimental results [11] and listed in Table 1. We have modified the MuMax3 simulator [34] to model the current flowing through an MTJ (in Fig. 1 and Fig. 2) and the current-induced spin-transfer torque when a voltage is applied across it.

The structure and electrical model of our proposed multi-level bit-cell are shown in Fig. 3 (a)-(b). The resistances, \( R_A, R_B, R_{O1} \) and \( R_{O2} \) vary with respect to the magnetization state of the regions \( A, B, O_1 \) and \( O_2 \), respectively and can be written as

\[
R_X = R_p + (R_{AP} - R_p) \left( \frac{1 - m \cdot m_p}{2} \right) \tag{6}
\]

where, \( R_X \in \{R_A, R_B, R_{O1} \text{ and } R_{O2}\} \), \( m \) and \( m_p \) are unit vectors describing the magnetization direction of FL and PL, respectively, in the corresponding regions. \( R_p \ (R_{AP}) \) is the resistance of the region when \( m \) and \( m_p \) are parallel (anti-parallel) to each other.

In order to study the magnetization dynamics at the bit-cell level, \( m \)-dependent values of \( R_A, R_B, R_{O1} \) and \( R_{O2} \) are first calculated using Eq. 6. The electrical model of our proposed bit-cell is simulated using SPICE-like [35], [36] circuit simulation tools to extract the electrical stimuli on the individual resistors and access transistors (Fig. 3 (b)). In this work, we have modeled the access transistors using the NCSU FreePDK45 [28]. The circuit simulation results are used in the MuMax3 micromagnetic simulation to determine the magnetization dynamics. At this point, the characterization data about the individual bit-cell is available and fed to the array/architecture-level simulator. We have used the NVSim simulation tool [30] to estimate the array/architecture-level energy consumption and latency of the proposed SIMC.

Fig. 6. Simulated electrical response at \( A, B, O_1 \) and \( O_2 \) of the bit-cell during write and XOR operations. The output of XOR computation between \( A = 1 \) and \( B = 1 \) is sensed from the current output at \( O_2 \). Low normalized current outputs in \( A, B \) and \( O_1 \) corresponds to bit ‘1’, whereas the high normalized current output in \( O_2 \) corresponds to bit ‘0’.
TABLE II

| Operation | Energy (pJ) | Latency (ns) |
|-----------|-------------|--------------|
| Write (A) | 6.721       | 2.308        |
| Write (B) | 1.739       | 1.15         |
| OR        | 15.793      | 5.308        |
| AND       | 8.233       | 2.808        |
| XOR       | 24          | 8.1160       |
| Reset     | 0.432       | 4            |

**All the values are in units of volts (V)**

B. Electrical Behavior of the Bit-cell

Fig. 6 shows the electrical behavior of the bit-cell during write follow by the XOR operation. The electrical stimuli listed in “Write (A)” row of Table II is applied to the bit-cell when bit ‘1’ is being written into A. Consequently, a skyrmion is nucleated in A and $R_A$ increases, which reduces the magnitude of current flow in A (see Fig. 6 (a)). The total delay for the write operation is about 2.5 ns (inclusive of 0.5 ns relaxation delay). Similarly, when bit ‘1’ is being written into B, the electrical stimuli listed in “Write (B)” row of Table II is applied to the bit-cell. Consequently, the skyrmion gets nucleated in B and the magnitude of current flow in B reduces (see Fig. 6(b)). The total delay of this write operation is also about 2.5 ns (inclusive of 0.5 ns relaxation delay).

The two-cycle XOR operation is next performed by first activating the STNO at $O_1$ to compute $A \cdot B$ and then activating the STNO at $O_2$ in the second phase. For activating the STNO at $O_1$, the electrical stimuli listed in “Compute ($O_1$)” row of Table II is applied to the bit-cell. As shown in Fig. 6 when both the inputs, A and B, are in state 1, a skyrmion is nucleated in $O_1$. As a result, $R_{O1}$ increases and current flow through $O_1$ is reduced (See Fig. 6(c)). The total delay of this operation is about 3.0 ns (inclusive of 0.5 ns relaxation delay). In the final compute cycle, the STNO in $O_2$ is activated for 5 ns using the electrical stimuli listed in “Compute ($O_2$)” row of Table II. Despite the voltage stimulus, no skyrmion gets nucleated in $O_2$ due to the presence of a skyrmion in $O_1$. Hence, there is no change in $R_{O2}$ and the magnitude of current flow through $O_2$ (see Fig. 6(d)). In this case, the output at $O_2$ is ‘0’, which corresponds to the result of the XOR operation between data stored at A and B. After the compute operation has finished, the output at $O_2$ can be read by applying the electrical stimuli listed in “Read” row of Table II to the bit-cell.

The voltages applied to CSL2 and CSL3 during write and compute operations, respectively, are determined from the magnetization phase diagram shown in Fig. 7, which was obtained by varying the voltage magnitude and pulse width. When the input voltage is less than 0.78 V, the magnetic oscillations attenuate within the time period of 4 ns (Window-A). If the input voltage is in between 0.788 V and 0.79 V, magnetization oscillates continuously for 6 ns (Window-B). Finally, if the voltage is greater than 0.8 V, magnetic oscillations are amplified and a skyrmion is nucleated (Window-C). Therefore, we apply 0.81 V to CSL2 during write operation, and 0.79 V to CSL3 during compute operation (Table III). The voltage used for the read operation (i.e., 0.25 V) is selected to avoid read-failure and read-disturb errors.
TABLE IV
VGG-like BNN Model

| Description     | W×H×IFs          | W×H×OFs          |
|-----------------|------------------|------------------|
| 1 BinConv1      | 32×32×3          | 32×32×128        |
| 2 BinConv2      | 32×32×128        | 32×32×128        |
| 3 Maxpool1      | 16×16×128        | 16×16×128        |
| 4 BinConv3      | 16×16×256        | 16×16×256        |
| 5 BinConv4      | 16×16×256        | 16×16×256        |
| 6 Maxpool2      | 16×16×512        | 4×4×512          |
| 7 BinConv5      | 8×8×256          | 8×8×256          |
| 8 BinConv6      | 8×8×512          | 8×8×512          |
| 9 Maxpool3      | 8×8×512          | 4×4×512          |
| 10 FullyConn1   | 1×1×8912         | 1×1×1024         |
| 11 FullyConn2   | 1×1×1024         | 1×1×1024         |
| 12 FullyConn3   | 1×1×1024         | 1×1×10           |

Fig. 8. Estimated layer-wise (a) energy consumption and (b) execution times per inference in SIMBA.

C. Evaluation of SIMBA Hardware Architecture

Based on the device write current requirements and FreePDK45 design rules, the area of each bit-cell in the SIMC is about 900 F² (Fig. 3(a)). We note that the bit-cell area is limited by the transistor size due to the high write current requirements (∼ 490 µA) of the memory device. Using the extracted electrical characteristics of an individual bit-cell, an SIMC with 1 KB data storage capability was simulated in NVSim to estimate the energy consumption and latency for different operations, which are summarized in Table III. The simulation results for the 1 KB SIMC units are used to evaluate the overall SIMBA hardware architecture.

In addition, the electrical stimuli applied to the popcount and comparator circuits proposed in Fig. 4 (c) are listed in Table IV since the switching characteristics of SOT-MTJs are well known in literature [37]-[39], the simulation results of popcount and comparator circuits have been omitted. The energy and latency consumed during these operations are determined to be 7.5 pJ and 10 ns, respectively. A popcount circuit processing 100 bits at a time is used for estimating these energy and latency values.

In this work, we designed SIMBA to accelerate the computations for a VGG-like BNN with 12 layers (see Table IV [23], [40]). The VGG-like BNN model performs the classification task on CIFAR-10 image data with an accuracy of 88.5%. Each BinConv layer in the network performs i) kw × kh × OF × W × H × IF number of XOR operations, ii) W × H × OF number of popcount operations (each on kw × kh × IF number of bits), and iii) W × H × OF number of comparator operations. Each Maxpool layer performs 3 × W × H × OF number of 2-bit OR operations, where the maxpooling filter size is 2×2. Each FullyConn layer performs i) IF × OF number of AND operations, ii) popcount operation on IF number of bits, and iii) OF number of comparator operations. Using the results in Table III and results from circuit and device simulations, the estimated energy consumption and time delay in each BNN layer are shown in Fig. 8 respectively. Over all, SIMBA is estimated to require 26.7 mJ of energy and 2.7 ms time delay to perform one classification task, which translates to throughput of 370.4 (images/sec).

1) Analysis of Device Characteristics on SIMBA Performance

We have also explored the impact of material properties (such as damping ratio, α, anisotropic energy density, KU, and saturation magnetization, MS) on the performance of proposed hardware. We found that a 33.3% of reduction in α improves the energy efficiency by 47% with 1.3× speedup (Fig. 9 (b-c)). Reduction in α decreases the current required to drive magnetization oscillations and improves energy efficiency and execution times. Separately, over 37.5% reduction of Ку improves the energy efficiency by 36.84% with 1.18× speedup (Fig. 9 (d-e)). Similar to the reduction in α, the current required to induce magnetization oscillations or switching reduces with reduced KU. When only MS is increased by 10%, the energy consumption is reduced by 23.68% with speedup of 1.13× (Fig. 9 (f-g)). This is due to increase in the demagnetizing field as MS increases. The demagnetizing field tends to reduce the current needed to induce magnetization oscillations or switching.

Since thermal effects can affect the magnetization dynamics of the spintronic devices and introduce errors in SIMBA, there is a need to study the influence of hardware errors on the overall classification accuracy. We induced errors in 1%-30% of the pixels in each BNN layer and observed the overall classification accuracy. Note that the magnitude of error in each pixel is randomly varied from ±1 to ±30. As shown in Fig. 2(a), the classification accuracy degrades by only 1% despite the presence of 30% errors in each BinConv layer. The worst classification error rate we observed is 12.4% whereas the error rate without thermal effects is around 11.5%.

VI. CONCLUSION

In summary, we propose a binary neural network (BNN) accelerator, referred to as SIMBA, based on a novel non-volatile skyrmionic in-memory processing engine. A hybrid mixed-mode device-to-architecture simulation framework was developed to design and evaluate SIMBA. Simulation results show that SIMBA consumes 26.7 mJ of energy and 2.7 ms of delay per inference task on a VGG-like BNN. The inference error rate is less than 1% even in the presence of 30% bit errors in the skyrmionic devices. Finally, we showed improvements in the performance of SIMBA that can be achieved by tuning.
device material parameters such as damping ratio, anisotropic energy, and saturation magnetization.

**ACKNOWLEDGMENT**

This work was supported in part by Singapore Ministry of Education (MOE) Grant MOE 2017-T2-1-114, in part by
E. Grimaldi, M. Baumgartner, D. Crotti, A. Fummont, P. Gambardella, and G. S. Kar, “Sot-mram 300nm integration for low power and ultrafast embedded memories,” in 2018 IEEE Symposium on VLSI Circuits, June 2018, pp. 81–82.

[38] X. Ma, L. Chang, S. Li, L. Deng, Y. Ding, and Y. Xie, “In-memory multiplication engine with sot-mram based stochastic computing,” arXiv preprint arXiv:1809.08358v1, 2018. [Online]. Available: https://arxiv.org/abs/1809.08358v1

[39] F. Parveen, S. Angizi, Z. He, and D. Fan, “Low power in-memory computing based on dual-mode sot-mram,” in 2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), July 2017, pp. 1–6.

[40] P. Guo, H. Ma, R. Chen, P. Li, S. Xie, and D. Wang, “Fbna: A fully binarized neural network accelerator,” in 2018 28th International Conference on Field Programmable Logic and Applications (FPL), Aug 2018, pp. 51–513.

Venkata Pavan Kumar Miriyala received the B.Tech degree in electronics and communication engineering from SSN Engineering College, India in 2014. He previously held positions at IBM-Research Tokyo, Japan, and Indian Institute of Technology (IIT), Hyderabad. He is currently working towards a Ph.D. at the National University of Singapore.

Kale Rahul Vishwanath received the M.Sc. Degree in Electrical Engineering from National University of Singapore and B.Tech. degree in Electronics Engineering from VJTI, Mumbai, India, in 2016 and 2014 respectively. He is currently working as a Research Engineer at the National University of Singapore.

Xuanyao Fong (SM’06, M’14) received the Ph.D. and B.Sce. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 2014 and 2006, respectively. He previously held positions at Advanced Micro Devices, Inc., Boston Design Center, Boxboro, MA, was a Research Assistant and Postdoctoral Research Assistant in Purdue University, and a Research Scientist in the Institute of Microelectronics, A*STAR, Singapore. Dr. Fong is currently an Assistant Professor in the Department of Electrical and Computer Engineering, National University of Singapore.