STT-BNN: A Novel STT-MRAM In-Memory Computing Macro for Binary Neural Networks

Thi-Nhan Pham[*], *Student Member,* Quang-Kien Trinh[*], *Member, IEEE,*
Ik-Joon Chang, *Member, IEEE,* Massimo Alioto, *Fellow, IEEE*

Abstract– **This paper presents a novel architecture for in-memory computation of binary neural network (BNN) workloads based on STT-MRAM arrays. In the proposed architecture, BNN inputs are fed through bitlines, then, a BNN vector multiplication can be done by single sensing of the merged SL voltage of a row. Our design allows to perform unrestricted accumulation across rows for full utilization of the array and BNN model scalability, and overcomes challenges on the sensing circuit due to the limtitation of low regular tunneling magnetoresistance ratio (TMR) in STT-MRAM.**

Circuit techniques are introduced in the periphery to make the energy-speed-area-robustness tradeoff more favorable. In particular, time-based sensing (TBS) and boosting are introduced to enhance the accuracy of the BNN computations. System simulations show 80.01% (98.42%) accuracy under the CIFAR-10 (MNIST) dataset under the effect of local and global process variations, corresponding to an 8.59% (0.38%) accuracy loss compared to the original BNN software implementation, while achieving an energy efficiency of 311 TOPS/W.

Keywords— **In-memory computing, STT-MRAM, binary neural networks, energy efficiency.**

I. INTRODUCTION

DEEP neural networks (DNNs) have enabled tremendous advances in prominent applications such as image classification, visual intelligence, and natural language processing. However, the computationally intensive nature of DNNs limits their adoption in energy- and area-constrained integrated systems at the edge of the Internet of Things (IoT), wearable, and mobile devices, among others.

At the algorithm level, precision scaling down to binary for both activations and weights has enabled substantial energy, area, and memory usage reductions [1]. This has facilitated on-chip storage and further energy efficiency improvements. In such binary neural networks (BNNs), multiplications are replaced by XNOR and bit-counting operations [2]. At the circuit level, in-memory computation of BNNs offers further energy and area reductions, in view of stricter data locality enforcement and inherent density [2]-[12]. In [2], [4], in-memory BNN accelerators were proposed based on 6T-SRAM and 8T-SRAM, respectively.

As embedded non-volatile memories (NVMs) potentially introduce further advantages in terms of density and leakage power, RRAM-based BNN in-memory accelerator architectures were recently introduced [5]-[10]. In [5], [6], the restriction to 0 or +1 activation values substantially limits the achievable accuracy [4]. Multiply-and-accumulate (MAC) operations with ±1 weights and activations are instead allowed in the memristor-based accelerator in [7] and the XNOR-RRAM in [8]. However, [7] relies on a crossbar array and hence suffers from well-known sneak current path issues. The XNOR-RRAM in [8] overcomes these limitations through a pseudo-crossbar 2T2R array, although it suffers from limited neural network model scalability. Indeed, RRAM bitcell rows must be twice the number of input neurons, substantially dilating the array size for a given in-memory model size. Also, current-domain operation inherently limits energy efficiency. STT-MRAM arrays are also being explored as a promising non-volatile option for in-memory computing applications, leveraging on their inherent advantages over other memory technologies [13]-[16], [17], [18]. For example, spin-orbit torque magnetic random access memory (SOT-MRAM), spin-transfer torque magnetic memory (STT-MRAM) have attracted considerable attention [11]- [18]. SOT-MRAM-based works are reported in [11], [12] perform only bulk bit-wise logic operations among a few operands, prohibiting highly-parallel single-access MAC operations. The PXNOR-BNN architecture in [12] is a write scheme-based solution, its energy per operation is increased by the extensive usage of write accesses, whose energy is well known to be higher than read [19]. However, the low tunneling magnetoresistance ratio (TMR) as known as the major challenge of STT-MRAM that makes it is difficult to apply STT-MRAM for analog computing, as discussed in [20]. As a consequence, most STT-MRAM-based in-memory BNN macros only perform bit-wise operations with two or three operands per memory access, as larger numbers of operands degrade the accuracy of the MAC computation and ultimately the classification accuracy [13], [14], [17], [18]. The macros in [15], [16] only support 0 or +1 values for activations with the limitations as in [5], [6].

This paper introduces a novel STT-MRAM in-memory highly-scalable architecture for BNNs (both the weights and neuron activations that are binarized to -1 or +1) with the single-access many-MAC operation, allowing to perform unrestricted accumulation across rows for full utilization of the

* These co-first authors contributed equally to the paper.

Manuscript received April 21, 2022. Thi-Nhan Pham is with the Department of Electronics and Radio Engineering, Kyung Hee University, Yongin-si, Gyeonggi-do 446-701, Korea (e-mail: *nhanpt@khu.ac.kr*). Quang-Kien Trinh is with Department of Microprocessor Engineering, Faculty of Radio Electronics–

Le Quy Don Technical University, Vietnam (e-mail: *kien.trinh@lqdtu.edu.vn*). Ik-Joon Chang is with the Department of Electronics and Radio Engineering, Kyung Hee University, Yongin-si, Gyeonggi-do 446-701, Korea (e-mail: *ichang@khu.ac.kr*). Massimo Alioto is with ECE – National University of Singapore, (e-mail: *massimo.alioto@nus.edu.sg*).

array and BNN model scalability. In the proposed architecture, inputs are fed through bitlines, and the products between inputs and weights are accumulated row-wise via the source lines, allowing simultaneous activation of multiple rows. Besides, the BNN vector multiplication is performed in the form of accumulated select line voltage, instead of power-hungry bitline current sensing in [5], [8].

Moreover, the reliable operation under process variations is enhanced via a time-based sense amplifier [21] and boosted voltage sensing [22]. The coordinated adoption of such techniques improves the energy efficiency of the periphery, eliminating voltage reference generation and distribution. This work is an extended version of our previous work in [23]. As additional contributions of this manuscript, select line boosting is introduced to further enhance the classification accuracy, and the area/throughput/energy/accuracy tradeoff is explored and quantified from circuit to algorithm. Also, the effect of process and environmental variations on robustness and classification accuracy is studied. Results are validated with a CNN under the CIFAR-10 dataset.

The remainder of the paper is structured as follows. Section II introduces the proposed STT-BNN architecture at the bitcell level. The architecture and the periphery circuitry are discussed in Section III. Section IV covers the effect of process variations on classification accuracy and its tradeoff with performance and energy. Design considerations on energy efficiency are discussed in Section V. Section VI validates the overall BNNs performance of the proposed in-memory macro under practical BNN workloads. Section VII concludes this work.

II. PROPOSED IN-MEMORY ACCELERATOR AT BITCELL LEVEL

Let us consider a generic BNN model with primary inputs $i_0, i_1 \ldots i_{m-1}$, inputs $h_0, h_1 \ldots h_{n-1}$ of the subsequent hidden layers, and outputs $o_0, o_1 \ldots$ of the output neurons. The generic weight in the l-th layer between its i-th input and its j-th output is named $w_{i,j,l}$, as summarized in Fig. 1. In the following, such binary weights are assumed to be stored within the memory array.

Fig. 1 shows how a conventional architecture for in-memory BNN acceleration is mapped onto an STT-MRAM memory array (or multiple of them, depending on the BNN size), based on traditional column-level accumulation. In detail, the binary inputs of a given layer are applied to the wordlines of the array, and the bit-wise products with the respective weights stored in a column are summed up and accumulated in the form of its bitline current. These currents are binarized to generate the outputs of the portion of the layer being computed.

This paper introduces a novel STT-MRAM in-memory computing architecture for binary neural networks, where both weights and neuron activations are binarized to -1 or +1. The binary weights are stored within the bitcells of the memory array, which are assumed to be of 2T-2J type as in Fig. 2. This structure allows performing unrestricted accumulation across rows for full utilization of the array, enabling the execution of substantially larger BNN model at a given memory capacity. The adopted bitcell is based on the popular 2T-2J structure, whose area is 2.2X larger than the densest 1T-1J bitcell
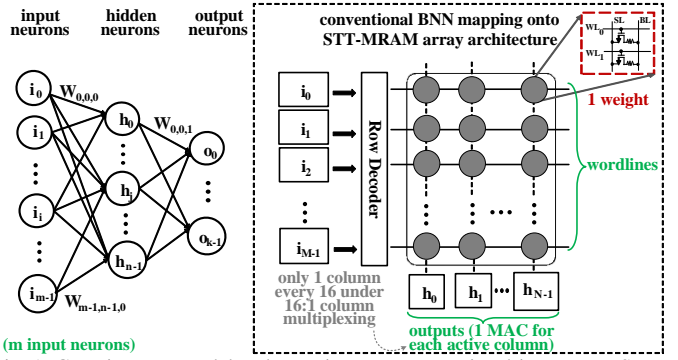

Fig. 1. Generic BNN model and mapping onto conventional in-memory STT-MRAM architecture with column multiplexing (e.g., one column multiplexed to the sense amplifier every 16 adjacent columns).
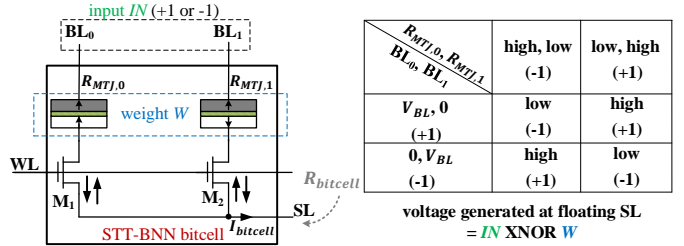

Fig. 2. Proposed usage of 2T2J STT-MRAM bitcell for single-bit XNOR of input feature and weight (modified from [23]).

structure [24].

As shown in Fig. 2, the gate of the access transistors M_1 and M_2 are both connected to the wordline WL, and the source is connected to the select line SL. The magnetic junction MTJ_0 is set either high (anti-parallel magnetization resistance R_{AP}) or low resistance (parallel magnetization resistance R_P) to respectively store -1 and +1 weight W. The other magnetic junction MTJ_1 is set to the complementary state. The source line voltage V_{SL} generated by a single bitcell being activated within a source line can be expressed through the Norton theorem as a function of the short-circuit bitcell current $I_{bitcell}$ (i.e., its current flowing from M_1-M_2 when SL is grounded), and the cumulative resistance seen from the SL terminal.

For a single bitcell under a +1 (-1) input IN, $V_{BL,0} = V_{BL}$ and $V_{BL,1} = 0$ ($V_{BL,0} = 0$ and $V_{BL,1} = V_{BL}$) and the bitcell short-circuit current $I_{bitcell}$ is expressed as

$$I_{bitcell}(IN = +1) = \begin{cases} \dfrac{V_{BL}}{R_P + R_{access,0}} & if\ W = +1 \\ \dfrac{V_{BL}}{R_{AP} + R_{access,0}} & if\ W = -1 \end{cases} \quad (1a)$$

$$I_{bitcell}(IN = -1) = \begin{cases} \dfrac{V_{BL}}{R_{AP} + R_{access,1}} & if\ W = +1 \\ \dfrac{V_{BL}}{R_P + R_{access,1}} & if\ W = -1 \end{cases} \quad (1b)$$

On the other hand, the overall resistance $R_{bitcell}$ seen from the SL terminal of the bitcell is data-independent and equal to $(R_{MTJ,0} + R_{access,0}) || (R_{MTJ,1} + R_{access,1})$. The access transistor resistance $R_{access,0}$ and $R_{access,1}$ inevitably contribute to $I_{bitcell}$ due to their series connection with $R_{MTJ,0}$ and $R_{MTJ,1}$,

and need to be kept small enough to appreciate the state change in the MTJs, and hence achieve an adequate sensing margin as discussed below. By applying the Norton theorem as in Fig. 2 and considering that $R_{access,0} = R_{access,1} = R_{access}$ (see effect of variations in Section IV), from (1a-b) the resulting voltage V_{SL} for an isolated active bitcell at position (i, j) within the memory array is

$$V_{SL} = R_{bitcell} \cdot I_{bitcell,ij} = \frac{V_{BL}}{X_{ij}}, \qquad (2)$$

where X_{ij} is defined as

$$X_{ij} = \begin{cases} \dfrac{R_P + R_{access}}{R_{bitcell}} & if \ \overline{W_{ij} \oplus IN_j} = +1 \\ \dfrac{R_{AP} + R_{access}}{R_{bitcell}} & if \ \overline{W_{ij} \oplus IN_j} = -1 \end{cases} \qquad (3)$$

being W_{ij} the weight stored by the bitcell, and IN_j is the input fed to column j. From (2)-(3), the source line voltage is proportional to the bitline voltage and directly depends on the XNOR of the input and the weight stored in the bitcell, as desired to implement a MAC unit with binary inputs and weights.

The resulting sensing margin $SM_{bitcell}$ of the V_{SL} voltage is the difference between the values associated with an output equal to +1 and -1. From (2), the sensing margin results to

$$SM_{bitcell} = V_{BL}R_{bitcell}\left(\frac{1}{R_P + R_{access}} - \frac{1}{R_{AP} + R_{access}}\right)$$
$$= V_{BL} \cdot TMR \cdot \frac{R_P}{R_P + R_{AP} + 2R_{access}} \qquad (4)$$

where TMR is the MTJ technology-dependent tunneling magnetoresistance ratio $(R_{AP} - R_P)/R_P$ [25], [26]. From (4), the sensing margin can be improved by adopting larger bitline voltages, and hence at the cost of higher read current and energy per operation. The sensing margin is also improved by adopting a stronger access transistor, at the cost of higher area/bitcell when increasing larger transistor aspect ratio, or higher power under lower threshold voltage and higher WL voltage. As a limit to the sensing margin, the read current is upper bounded by the maximum value that avoids unintentional MTJ bitflips [22]. Circuit simulations were performed through the popular Spice-compatible macroscopic model in [27], [28] for MTJs with parameters summarized in Table I, and a commercial 65-nm CMOS design kit. Fig. 3 shows the resulting waveforms when the WL voltage V_{WL} is asserted at 0.7 V and V_{BL}=0.3 V, and the access transistor width is 4X the minimum allowed by the technology (i.e., W=0.54 um, L= 0.06 um). Under this sizing, the read current through the MTJs in Fig. 2 is $I_{MTJ} = 29.9$ μA

TABLE I. MAIN MTJ PARAMETERS

MTJ size (WxL)	60 nm x 60 nm
MTJ thickness (Tm)	1.5 nm
oxide thickness (T$_{MgO}$)	1.15 nm
MTJ resistance variability	6.9%*
nominal R_P (R_{AP})	2 KΩ (5.3 KΩ)
TMR	165%
max MTJ read current (1E-9 bitflip rate) [22]	50 μA

* The MTJ model variability is calibrated using the experimental results in [28]. The MTJ resistance variability is set by the variation of MTJ sizes as well as variations in the T$_{MgO}$ thickness.
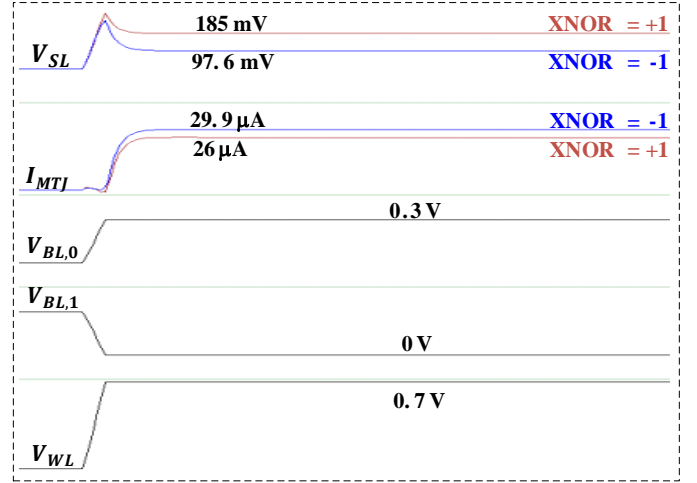

Fig. 3. SL voltages and bitcell currents output at V_{BL}=0.3 V and V_{WL}=0.7 V.
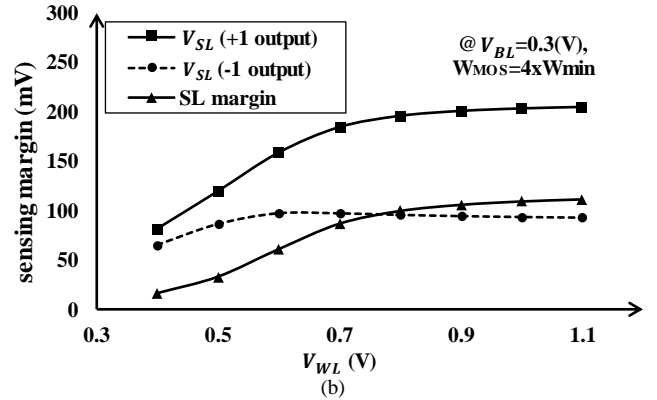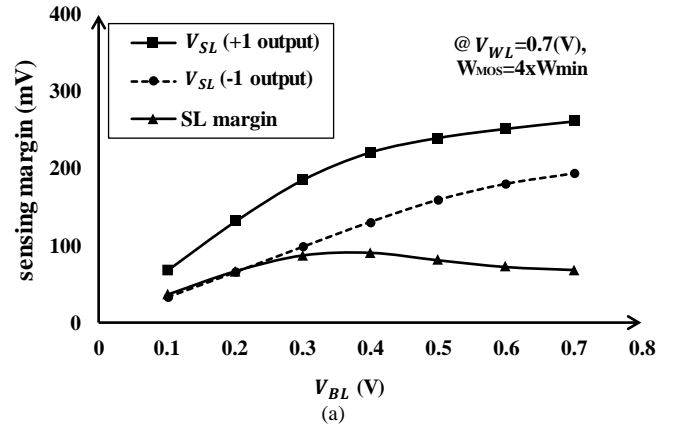

Fig. 4. SL voltage and sensing margin dependence on a) V_{BL}, b) V_{WL}.

(26 μA) when $XNOR = -1$ (+1). These currents are lower than the 50-μA upper limit imposed by unintended bitflips from Table I. From Fig. 3, the source line voltage is 97.6 mV under (+1,-1) or (-1,+1) input-weight pairs, whereas it is 185 mV otherwise. This leads to a sensing margin of 87.4 mV.

The effect of V_{BL} and V_{WL} on the sensing, the margin is shown in Figs. 4(a-b), under the above parameter values. From Fig. 4(a), a V_{BL} increase first leads to a linear increase in the sensing margin, it then reaches a peak at V_{BL}~0.3 V, and finally leads to a moderate monotonic decrease. The linear increase is simply due to the voltage drop increase over the MTJ resistances as in

(4), whereas the decrease is explained by the moderately non-linear behavior of $R_{MTJ,P}$ and $R_{MTJ,AP}$ leading to a reduction in their difference in (4) at relatively large voltages [24]. On the other hand, from Fig. 4(b) an increase in V_{WL} initially improves the sensing margin, and then makes it saturate at $V_{WL} \sim 0.7$ V, as the access transistor resistance starts becoming much smaller than $R_{MTJ,P}$ in (4). Very similar considerations hold in terms of access transistors aspect ratio, whose values greater than 4X the minimum lead to sensing margin saturation. Accordingly, V_{BL} and V_{WL} are respectively set to a minimum value of 0.3 V and 0.7 V to maximize sensing margin and robustness. From Figs. 4(a-b), the generation of V_{BL} and V_{WL} does not need to be particularly accurate, as even large deviations of 50-100 mV have a minor effect on the sensing margin.

These considerations justify the choice $V_{BL} \sim 0.3$ V and $V_{WL} \sim 0.7$ V along with the 4X transistor size in the following, as they maximize the sensing margin while avoiding over-design.

III. STT-BNN Array Architecture And Circuit Design

This section presents the proposed STT-BNN architecture. The MxN array in Fig. 5 consists of M rows and N columns (i.e., $2N$ bitlines), with wordlines being used to enable computation in the respective rows, and the source lines as outputs. The bitline (BL) drivers encode the input features of the neural network being mapped in the form of a signal pair as discussed in the previous section (see NMOS and PMOS selection transistors at the bottom of Fig. 5). Each wordline (WL) is driven by a 2-stage buffer to keep the settling time of the WL signal negligible compared to the overall access time.

A. Array Organization and Computation

As all rows operate independently of each other, let us consider a single row as depicted in blue in Fig. 5. Let the number of XNOR outputs equal to +1 (-1) across the row be N_1 ($N_0 = N - N_1$). Compared to the single bitcell resistance $R_{bitcell}$ in Section II, the resistance seen from the SL terminal in a row is reduced to $R_{bitcell}/N$. From (2)-(3), the select line voltage $V_{SL,i}$ in the considered row i-th is

$$V_{SL,i} = \frac{R_{bitcell}}{N} \sum_{j=0}^{N-1} I_{bitcell,ij} = V_{BL} \frac{R_{bitcell}}{N} \sum_{j=0}^{N-1} \frac{1}{X_{ij}} =$$
$$= V_{BL} \left(\frac{N_0}{N} \cdot \frac{R_{bitcell}}{R_{AP} + R_{access}} + \frac{N_1}{N} \cdot \frac{R_{bitcell}}{R_P + R_{access}} \right) \quad (5)$$

where X_{ij} is the resistance determining the bitcell current of the bitcell (i,j) in (1)-(3). From (5), the row select voltage linearly depends on N_1, ranges from $V_{BL} \cdot \frac{R_{bitcell}}{R_{AP}+R_{access}}$ to $V_{BL} \cdot \frac{R_{bitcell}}{R_P+R_{access}}$, and its mid-point value (i.e., $N_1 = N/2$) is $V_{SL,50\%} = 0.5 \cdot V_{BL}$.

The output binarization is performed by assigning the output OUT_i of the select line in the generic i-th row to +1 if $V_{SL,i} \geq V_{SL,50\%}$ and to -1 if $V_{SL,i} < V_{SL,50\%}$. This simply requires a sense amplifier (see Subsection B) and the generation of the voltage $V_{SL,50\%}$, which is straightforwardly derived from V_{BL}. Hence, the logic output of the senseamp in the i-th row is equal to the sum of the XNOR computations across its bitcells

$$OUT_i = \sum_{j=0}^{N-1} \overline{W_{ij} \oplus IN_j}. \quad (6)$$
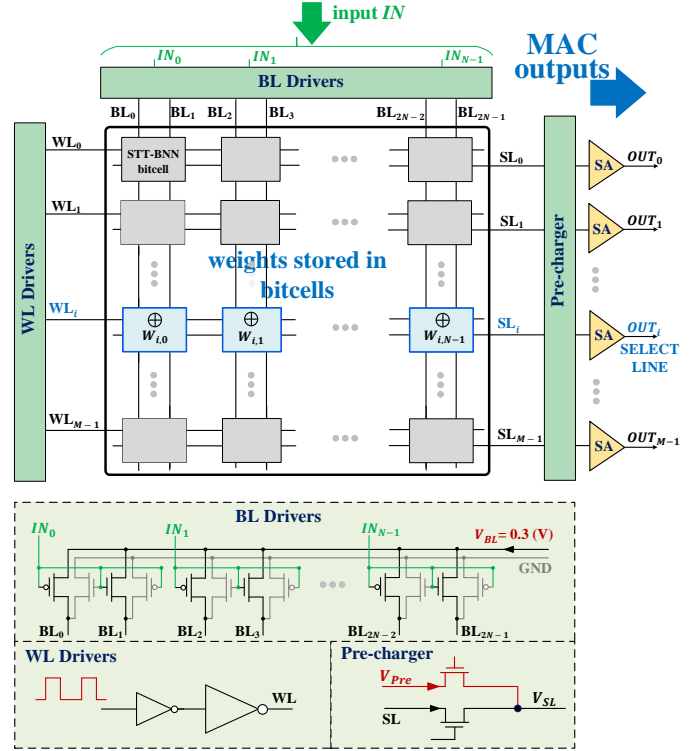

Fig. 5. STT-BNN architecture and MAC operation mapping onto the array.

The above results are immediately generalized to the entire array, as the above scheme applies independently to each row. The independent execution of the accumulations in the M rows achieves full utilization of the array with the maximum degree of accumulation-level parallelism of M.

B. Source Line Read Out via Time-Based Sense Amplifier

The binarization of the BNN accumulators in Fig. 5 requires the comparison of the V_{SL} voltage with the mid-point value $V_{SL,50\%}$. To suppress the need for a standalone and accurate voltage reference, the time-based sensing (TBS) technique in [21] is adopted to move read-out from voltage to time domain.

The senseamp circuit for the binarization of the accumulated value in each row is shown in Fig. 6(a). The TBS senseamp converts the SL voltage through a voltage-controlled current-starved inverter gate. The resulting current $I_{starving}$ is then converted to time through the current-starved gate delay, which is proportional to $1/I_{starved}$. Such delay is then compared with a delay line-based pulse of duration T_{REF} that inherently tracks process/voltage/temperature variations (see delay line in Fig. 6(a)). The comparison simply requires a latch driven by such a pulse, which comes at no cost since the sensing output needs to be stored in a latch anyway [21]. An inverter is added between the LOAD node and the latch input to sharpen the pulse. Time binarization is carried out by digitally configuring the delay line, and generates the accumulated result at the latch output.

The benefits and the limitations of time-based sensing in BNNs are evaluated in Fig. 6(b), which plots the SL voltage and the sensing delay versus the accumulation result for a typical
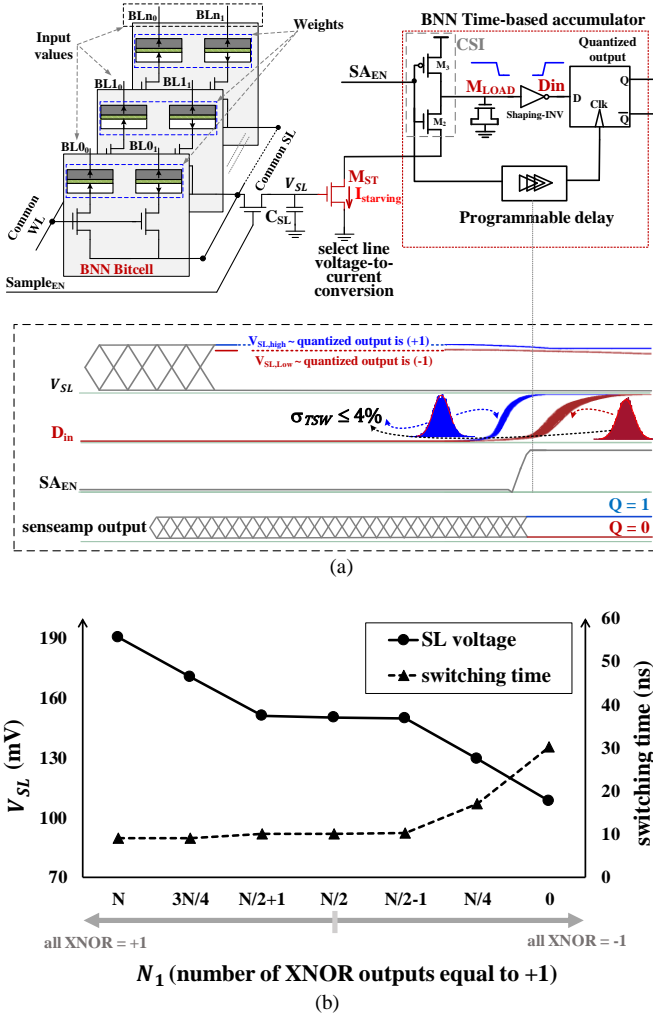
Fig. 6. (a) Time-based sensing and related timing diagram, (b) SL voltage and sensing time vs. N_1 (number of accumulated XNOR outputs equal to +1) from a row with N=128 bitcells [23]. Monte Carlo simulation results are shown for the waveform of the latch input D_{in}.

array with N=128. From (5), V_{SL} is highest (lowest) when $N_1 = N$ ($N_1 = 0$), i.e., when all XNOR outputs within the same row are +1 (-1). The same figure shows that the delay of the complete in-memory computation is data-dependent, as the SL voltage is maximum (minimum) when $N_1 = N$ ($N_1 = 0$) since X_{ij} in (3) is determined by the lower MTJ resistance R_P (higher MTJ resistance R_{AP}).

From a robustness viewpoint, the senseamp discrimination ability is more challenging when the accumulated value is closest to the decision threshold $V_{SL,50\%}$, and hence when having to distinguish V_{SL} from the logic value $N_1 = N/2 + 1$ and $N_1 = N/2 - 1$. In this worst case, the corresponding delay difference at the input of the final latch in Fig. 6(a) was found to be 313 ps, which is much higher than a gate delay[1], and can be hence easily distinguished in any sub-100nm technology. As an example, the pulse duration T_{REF} of a delay line based on an inverter chain can be tuned with a step down to a few tens of ps or lower. The delay line pulse duration T_{REF} inherently mimics

the same circuit as the TBS path delay [21], tracking its PVT variations.

It is worth observing that the computation completion occurs at the mid-point delay, rather than after the longest delay determined by the input voltage state. Indeed, the output Q of the TBS in Fig. 6(a) is latched at the mid-point delay and is hence equal to 1 (0) if the overall delay is lower (higher) than the mid-point delay generated by the delay line. After latching the value, the computation output is stored and a new one can be executed, without waiting for the longest delay. After setting the mid-point delay to be able to discriminate the above values around the decision threshold, the resulting computation time is 10 ns, which is substantially shorter than prior BNN implementations in CMOS logic or memory [6], [8], [13], [14].

The architecture with time-based sensing naturally introduces a sampling capacitor C_{SL} at the point of sensing (see Fig. 6(a)). Such capacitive buffer allows to turn off the select line current as soon as its voltage is stable, thus saving energy. Also, the worst-case sensing margin and hence robustness can be further improved by optimizing the current $I_{starving}$ through the size of transistor M_{ST}, and the current-starved inverter load determined by the size of M_{LOAD} in Fig. 6(a). Transistor size optimization and the exploration of the area-energy-sensing error rate tradeoff at design time are discussed in Section V.

IV. IMPACT OF PROCESS VARIATIONS

Process variations inevitably degrade the sensing margin, potentially inducing errors in the logical value at the output of each row and hence each accumulation. To quantify the impact of variations, Monte Carlo simulations were run under global and local variations in both transistors and MTJs, based on the simulation framework in Section II. Typical arrays with M=128 rows and N=64, 128, and 256 columns will be considered in the examples below. All building blocks from decoding to read-out are included to make the array fully functional and self-consistent. In the adopted simulation framework, the effect of fluctuations in V_{BL} and V_{WL} due to process variations in the bitline and wordline drivers was explicitly included. Their effect is insignificant, as it leads to variations in V_{BL} and V_{WL} in the 0.3% range or lower.

To quantify the impact of process variations on robustness, some important metrics were introduced. First, the cell-level *row error rate* (*RER*) is the probability of an erroneous output caused by variations in an elementary MAC operation [29]:

$$ RER = \Pr[TM < 0] = \frac{1}{2}\left[1 + \text{erf}\left(-\frac{1}{\sqrt{2}} \cdot \frac{1}{\sigma_{TM}/\mu_{TM}}\right)\right] \quad (7), $$

where $TM = T_{SW} - T_{REF}$ is the time margin or the difference between switching time at the TBS output T_{SW} and the reference time T_{REF} generated by the delay line (see Fig. 6(a)). In (7), μ_{TM} and σ_{TM} denote the mean and the standard variation of the time margin TM.

The row-level *conditional row error rate*[2] $CRER(k)$ is the conditional error probability under the condition $N_1 = k$ [2], [8]

---

[1] For example, the fan-out-of-4 delay of an inverter gate in the adopted technology is 15 ps.

[2] $CRER(k)$ is generally defined as the complement of the *pass_rate* [2], [8], i.e., $CRER(k) = 1 - pass\_rate$.

(i.e., the output accumulation has a number of $(+1)$ equal to k, which occurs with probability $C_N^k/2^N$)

$$CRER(k) = RER(k) \cdot \frac{C_N^k}{2^N} \qquad (8),$$

where $C_N^k/2^N$ is the probability that the output accumulation has a number of $(+1)$ equal to k, and C_N^k is the number of combinations of k bits chosen from a total of N bits [29].

Finally, the array-level *average error rate* (AER) is the average error rate for all possible values of N_1 (i.e., averaging $CRER(k)$ across the values of k from 0 to N):

$$AER = \sum_{k=0}^{N} CRER(k) = \sum_{k=0}^{N} RER(k) \cdot \frac{C_N^k}{2^N} \qquad (9),$$

which is the overall error probability that quantifies the impact of process variations on the output.

A. Error Analysis under Time-Based Sensing

As discussed above, the sizes of transistors M_{ST} and M_{LOAD} in Fig. 6(a) are key design knobs in the senseamp to improve robustness. Accordingly, their sizes were swept to explore the underlying tradeoffs and discretized as a multiple of an elementary minimum-sized transistor.

A larger size of M_{ST} directly reduces the variability of $I_{starving}$ and hence the time margin σ_{TM}/μ_{TM} according to a square root law [30]. Accordingly, the RER in (7) is improved (i.e., reduced) by selecting a larger size for transistor M_{ST}. Regarding the effect of M_{LOAD} in Fig. 6(a), a larger size monotonically and linearly increases the mean μ_{TM} of the time margin T_{SW}, due to the linear increase in the current starved inverter delay. Hence, the RER in (7) can be arbitrarily reduced at the cost of slower sensing and hence lower array throughput.

From a design standpoint, the size of the M_{ST} and M_{LOAD} elementary transistor was set with a channel width of 16X (4X) the minimum allowed by the technology, and a channel length of 8X (4X) longer than the minimum value. In this design, the size of M_{LOAD} and M_{ST} is set so that their ratio keeps the reference delay in Fig. 6(a) equal to 10 ns, and their absolute size is set to explore different RER targets, hence AER values. To achieve an AER equal to 0.1 with an array of N=128 at the access transistor width of four times the minimum, the multiplicities of M_{ST} and M_{LOAD} respectively need to be set to 12 and 19. For the reasons discussed in Section II, the access transistor in the bitcell was chosen to be 4X the minimum size, as a tradeoff between array density and robustness. The resulting error rate from 1,000-run Monte Carlo simulations is shown in Fig. 7 versus the number N_1 of XNOR outputs equal to +1 in the same row. The worst-case error rate is expectedly observed around the decision threshold $\frac{N}{2}$, as discussed above. Quantitatively, the row-level $CRER$ at the center points ($N/2 - 1$ and $N/2 + 1$) has the highest value of 2.4E-2.

The $CRER$ rapidly improves when moving away from the decision threshold, as in Fig. 7. In particular, $CRER$ is halved when moving to the next pair of points $N/2 \pm 2$ around the decision threshold. Then, it is reduced by more than 10X at the close pair $N/2 \pm 4$, and rapidly decreases to the very low values of 1.2E-5 and 4.1E-5 at $N/2 \pm 8$. The error rate keeps

decreasing at an even faster rate under values of N_1 that are farther away from N/2.

From the above considerations, the most significant row-level errors are confined to very few values of N_1, whereas most of the others lead to very small error rates. At the array level, this means that the AER in (9) is contributed only by a few RER values near the reference point, whereas the other components give a negligible contribution. From Fig. 8, larger arrays lead to a nearly-linearly increase in AER. This is because a larger number of columns N makes the SL voltages around the decision threshold closer to each other from (5), which in turn makes their discrimination harder and increases the number of significant RER contributions to the overall AER. From Fig. 8, the increase in AER in larger arrays can be mitigated by choosing a bitcell with larger access transistors and hence layout footprint, as this reduces the effect of local variations across bitcells. However, this comes at a significant array area increase, as it impacts all bitcells within it.

To ameliorate the AER in the proposed STT-BNN architecture, SL boosting is introduced in the next subsection to significantly improve the robustness at minor area overhead and at no increase in the bitcell footprint.

B. SL Boosting for Sensing Error Rate Improvement at Low Area

To improve the robustness against process variations, the SL voltage can be preliminarily magnified before being compared with the threshold decision (i.e., before sensing). The boosting concept was originally proposed for bitline readout in STT-MRAM arrays [22], and is here extended to the source line.

In source line boosting, a voltage booster with voltage gain $G = V_{boosted}/V_{SL}$ is inserted between the SL and the time-based senseamp as in Fig. 9. The voltage gain G is easily set to the targeted value by cascading an adequate number n of voltage boosting stages as in Fig. 9 [22]. Accordingly, the senseamp is now receiving a much wider input voltage swing $V_{Boosted}$, thus enlarging the sensing margin at the cost of relatively small area overhead. For example, the area overhead of a second-order booster (n=2) with 200-fF boosting capacitors is only 2.8% in a 128x128 STT-BNN array.

Fig. 10 plots the resulting $CRER$ versus N_1 under second-order SL boosting. Its comparison with Fig. 7 shows an evident reduction in the error rate by 2X for the points $N/2 \pm 2$ around
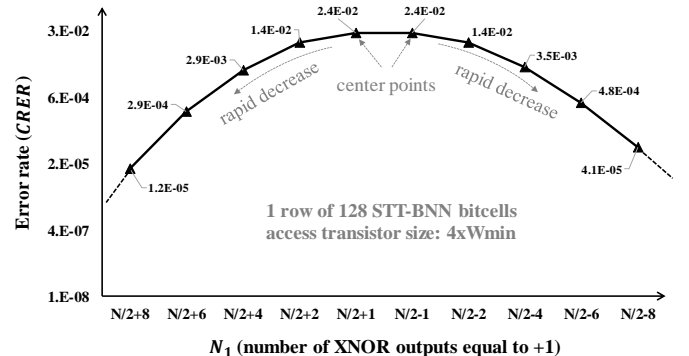

Fig. 7. Row-level conditional row error rate $CRER$ vs. N_1 (number of XNOR outputs equal to +1 in the same row, under N=128 bitcells).
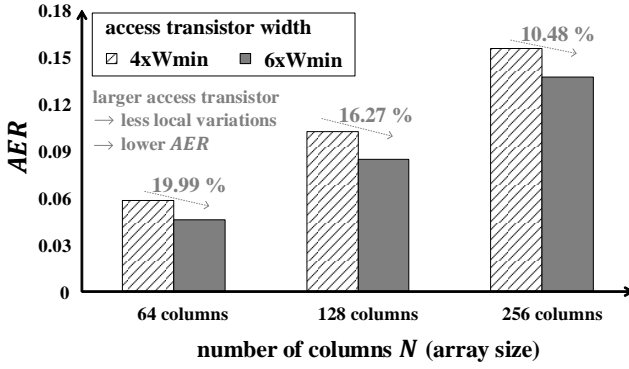
Fig. 8. Array-level average error rate AER vs. STT-BNN sub-array size for different access transistor size.
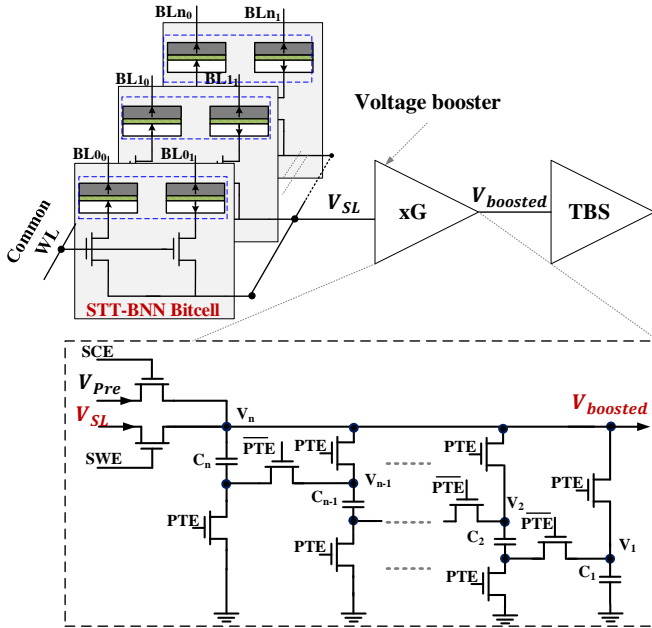

Fig. 9. SL voltage boosting applied to source line sensing (used for bitline sensing in [22]).
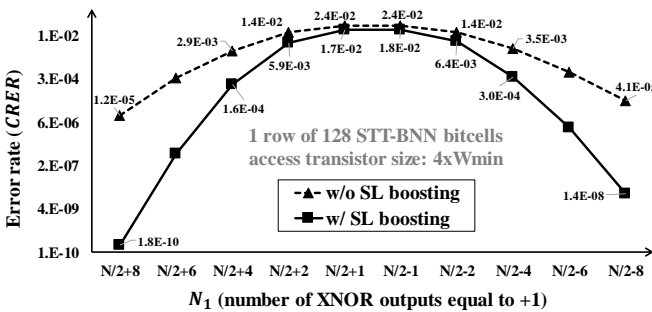

Fig. 10. Row-level conditional row error rate $CRER$ vs. N_1 with and without SL boosting.

the decision threshold, and by three orders of magnitude or more for the points $N/2 \pm 8$. This clearly improves the AER at the array level, as the number of significant RER contributions in (9) is further reduced. In particular, the AER under second-order SL boosting is respectively improved by 54.67%, 49.8% and

---

3 Simultaneous (time-staggered) operation of all sub-arrays occupies a larger (smaller) fraction of the array at shorter (longer) execution time, as a tradeoff that is common to any in-memory compute framework.

41.92% for N equal to 64, 128 and 256, compared to the case without SL boosting under an access transistor width of four times the minimum.

The above improvements are more pronounced than those brought by access transistor oversizing as evidenced in Fig. 8, and also come with much lower area overhead (e.g., 2.8% instead of 16.2% for the array size of 128x128), since they impact only senseamps. Accordingly, SL boosting is the preferred design choice when tighter AER targets need to be met.

V. Evaluation of Energy Efficiency of STT-BNN

In the proposed STT-BNN macro in Fig. 5, the energy per accumulation across the N bitcells in the same row is contributed by the bitcells and the row-line periphery as follows:

$$E_{accumulation} = E_{row} + E_{SA}$$
$$= E_{WL} + E_{BL} + E_{pre} + E_{SA} \quad (10)$$

where E_{row} is the energy consumed to develop the final SL voltage corresponding to the accumulation in a row. This includes the energy of bitline pre-charge (E_{pre}), WL drivers (E_{WL}), and the energy consumed by the N bitcells that are performing the computations (E_{BL}, which is proportional to the number of bitcells involved in the process). E_{SA} is the senseamp energy to binarize the SL voltage for the entire row accumulation.

Fig. 11 shows the energy breakdown for each accumulation across the N=128 bitcells, as evaluated with and without applying the source line boosting technique. From this figure, E_{BL} is the dominant contribution and exceeds 75% of the overall energy/accumulation. Then, E_{SA} is the second largest contribution accounting for 13.26% (11.65%) when SL boosting is (is not) applied, confirming that SL boosting comes at a minor energy cost. The third largest energy contribution is E_{WL}, which accounts for about 10%. E_{pre} is generally insignificant, being less than 1%.

Fig. 12 plots the energy per row-wise accumulation, and the inference accuracies as evaluated from two neural networks built in Torch7 [31], and mapped into the proposed STT-BNN macro similar to [8]. The first and simpler neural network is a multi-layer perceptron (MLP) for handwritten digit classification on the MNIST dataset [32], whose structure is detailed in Table II. The MLP is structured into four layers, and the MAC computations for each layer is partitioned into sub-computations that fit the array size. The table also details the resulting number of STT-MRAM sub-arrays necessary to cover each layer[3]. For instance, Table II indicates that the second layer requires a (2,048x2,048) matrix multiplication and accumulation with 2,048 input features and 2,048 output features.

From the same table, the overall matrix multiplication and accumulation in Table II is mapped onto different sub-array sizes from 256x256 down to 64x64, for which the resulting number of sub-arrays required for the computation it is evaluated.

The second example of the neural network is a convolutional
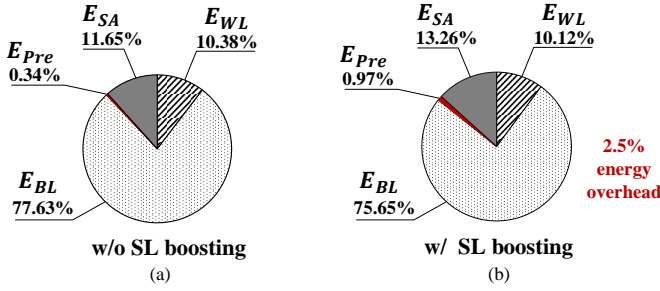
Fig. 11. The energy breakdown per accumulation (a) without and (b) with source line boosting (N=128 bitcells).

TABLE II. STRUCTURE OF MLP FOR HANDWRITTEN DIGIT CLASSIFICATION AND MAPPING OF EACH LAYER ONTO THE STT-BNN ARRAY (MxN)

MLP Layer	matrix size in the neural network (MLP)	STT-BNN sub-array size		
		256x256	128x128	64x64
1	784x2048	-	-	-
2	2048x2048	64	256	1024
3	2048x2048	64	256	1024
4	2048x10	8	16	32

TABLE III. STRUCTURE OF CNN FOR IMAGE CLASSIFICATION AND MAPPING OF EACH LAYER ONTO THE STT-BNN ARRAY (MxN)

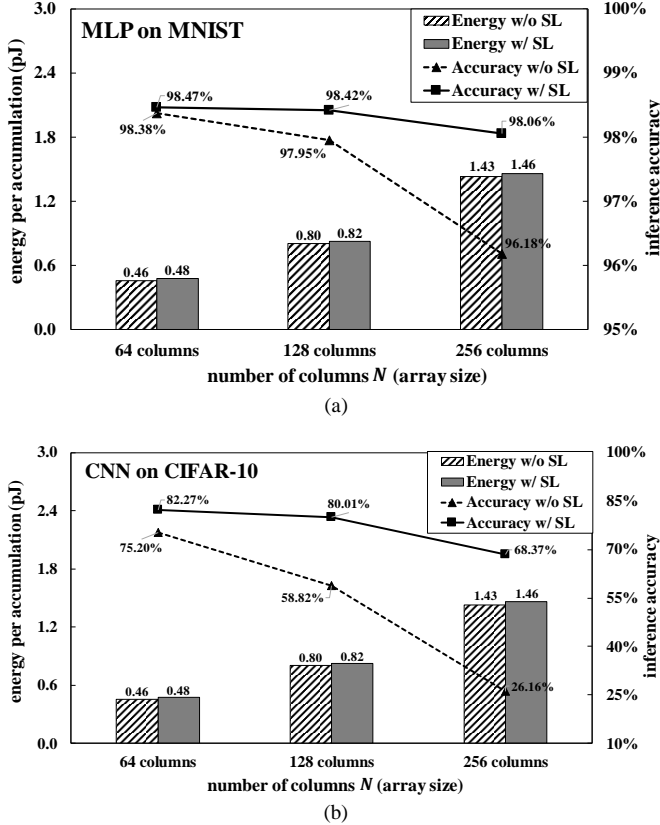| CNN Layer	kernel size in the neural network (CNN)	STT-BNN sub-array size		
		256x256	128x128	64x64
1	(3, 128, 3, 3)	-	-	-
2	(128, 128, 3, 3)	3	9	36
3	(128, 256, 3, 3)	6	18	72
4	(256, 256, 3, 3)	9	36	144
5	(256, 512, 3, 3)	18	72	288
6	(512, 512, 3, 3)	36	144	576
7	8192x1024	128	512	2048
8	1024x1024	16	64	256
9	1024x10	4	8	16

Fig. 12. STT-BNN energy and inference accuracy vs. number of columns N in (a) MLP classifying handwritten digits (MNIST), (b) CNN classifying images (CIFAR-10), both with and without source line boosting (no re-training).

neural network (CNN) for image classification on the CIFAR-10 dataset [33]. The CNN comprises six convolutional layers and three fully-connected layers, as described in Table III. For example, the kernel size (128, 256, 3, 3) in its second layer is associated with 128 input features, 256 output features, and the filter size is 3x3.

As an illustrative example, Fig. 12(b) shows that source line boosting substantially improves the CNN accuracy from 26.16% to 68.37% with array size N as 256, at the cost of a marginal energy increase from 1.43 pJ to 1.46 pJ. The underlying tradeoff is discussed in the following section.

VI. SYSTEM-LEVEL VALIDATION AND COMPARISON WITH PRIOR ART

The impact of process variations and the effectiveness of the proposed architecture were studied by mapping the neural networks in the previous section onto the STT-BNN array. To propagate the effect of errors to the system level, errors were injected in the array according to the AER profile in Section IV.

As discussed above, the size of the STT-BNN sub-array and the size of the bitcell access transistor affect the average error rate, which in turn determines the classification accuracy of the neural network mapped onto the array. Regarding the size of the array, Figs. 12(a)-(b) show that the classification accuracy tends to be degraded in larger sub-arrays with an increased number of columns N. Such accuracy degradation is expected from the error rate increase caused by the larger N in (9), which places a larger number of accumulation values close to the decision threshold of the senseamp. Under source line boosting, the degradation of the MLP classification accuracy is negligible in wider arrays with practical size. For example, an array with N=128 experiences a 0.05% accuracy drop, compared to an array with N=64. The accuracy degradation is somewhat more pronounced yet still rather small (0.41%), when moving to N=256 from Fig. 12(a).

The CNN accuracy degradation at N=128 in Fig. 12(b) is expectedly more pronounced (2.26%) than the MLP, being image classification (CIFAR-10 dataset) a significantly more complex task than handwritten digit classification (MNIST dataset). The CNN accuracy is degraded much more significantly (13.9%) when N is increased to 256. From the same figures, the accuracy without source line boosting is degraded much faster and to an intolerable extent (49.04%) at N=256. In other words, source line boosting is very effective in preserving accuracy in spite of local variations, and is necessary to keep accuracy within reasonable bounds in larger arrays. From Figs. 12(a)-(b), this is achieved at the cost of a minor energy increase (~2%) compared to the case without source line boosting.

From the above considerations, the number of columns N per sub-array should be increased to improve the array density, and to increase the maximum degree of parallelism in the accumulation process, and hence improve the throughput. Source line boosting helps extend the range of sub-array sizes

at which the accuracy degradation is within a given target, and hence makes the area-throughput-accuracy tradeoff more favorable at the expense of energy.

The effect of the access transistor size on the inference accuracy is shown in Figs. 13(a)-(b) for the MLP and the CNN neural network. From these figures, the access transistor 4X size is confirmed to be a good compromise between array density and accuracy. Indeed, a size increase to 6X the minimum allowed leads to a minor accuracy improvement towards the ideal accuracy obtained when no process variations occur. As a representative example, the accuracy at 4X size for the MLP (CNN) computation at N=128 drops by 0.29% (5.7%) compared to the ideal accuracy obtained without variations, from Figs. 13(a-b).

The effect of temperature variations on the inference accuracy is depicted in Fig. 13 for N=128. Compared to the STT-BNN design at room temperature, the classification accuracy of the MLP slightly drops by up to 0.1% at 85 oC, whereas it drops more significantly by 3.41% for the CNN in view of its more complex task. Also, compared to the baseline BNN in [31], the classification accuracy of the MLP drops by up to 0.48% at 85 oC, whereas it drops by 12% for the CNN. Overall, the effect of mismatch is markedly more pronounced than temperature.

Our proposed STT-BNN architecture is able to utilize 100% of a sub-array simultaneously, in view of its row-level accumulation process over source lines, which can simultaneously utilize all columns. Indeed, column multiplexing does not affect accumulation over source lines, allowing to accumulate overall bitlines instead of only those selected by column multiplexers.

Table IV summarizes the comparison with prior art on in-memory computing architectures based on non-volatile arrays, including the proposed STT-BNN architecture as reported for a 128x128 sub-array size. From this table, the maximum achievable classification accuracy of STT-BNN over the MNIST dataset is equivalent to [8], [35] and better than [15] by 3.4%. The work in [8] uses a 3-bit multilevel sense amplifier (i.e., a 3-bit ADC) to push the classification accuracy over the CIFAR-10 dataset at 86.08%, which is 6.07% higher than the proposed STT-BNN. However, the adoption of ADCs in the as readout circuitry comes at a very significant penalty in terms of throughput, which is improved by four orders of magnitude in STT-BNN thanks to the adoption a simple (1-bit) sense amplifier, which is ubiquitously available in memory periphery even for conventional read/write. Similarly, STT-BNN improves the energy efficiency over [8] by 2.2X, and by 680-18,400X compared to [17], [18]. The energy efficiency improvement offered by STT-BNN can be attributed to its unique ability to support full-array MAC operations in a single memory access phase, and the replacement of common current sensing with the more energy-frugal voltage (ADC-less) sensing.

VII. CONCLUSION

In this paper, the STT-BNN architecture for in-memory BNN acceleration in STT-MRAM arrays has been introduced. In the STT-BNN architecture, the products of inputs and weights are accumulated over source lines at the row level. This allows full utilization of the array and hence higher throughput than prior STT-MRAM architectures. Since the input neurons are fed through bitlines instead of wordlines, the MAC computation also achieves an energy efficiency that is superior to prior non-volatile memory architectures.

To gain an insight into the impact of process variations, circuit and analytical models have been derived to quantify the error rate at the bitcell, row, and array level. The resilience against variations has been improved through the adoption of time-based sensing (TBS) and source line boosting, making the area-throughput-accuracy tradeoff more favorable.

In summary, the unique full array utilization, the energy efficiency, and the accuracy in practical neural network computations make the STT-BNN architecture well suited for
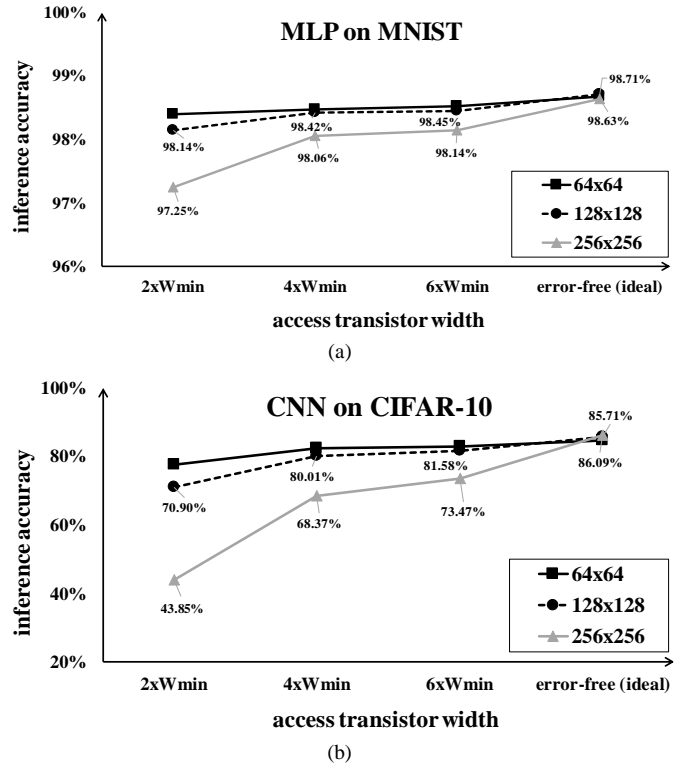

(a)

(b)

Fig. 13. Inference accuracy of proposed STT-BNN on (a) MNIST and (b) CIFAR-10 dataset vs. access transistor size for various sub-array sizes.
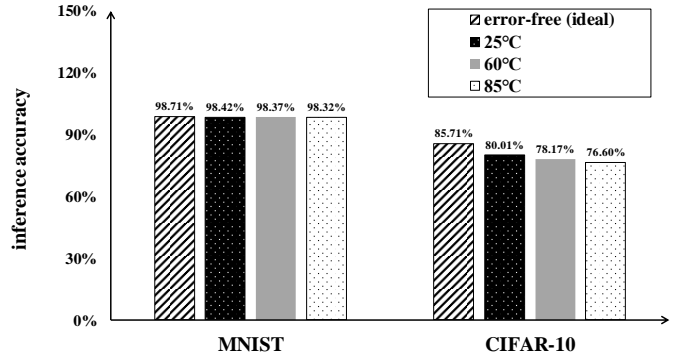

Fig. 14. Inference accuracies of the STT-BNN array of 256 bitlines (N=128) on MNIST and Cifar-10 with different temperatures.

TABLE IV. COMPARISON WITH PRIOR ART

	VLSI'21 [34]	DATE'18 [8]	IEDM'18 [15]	IEEE Trans. Magn.'18 [17]	TCAD'20 [18]	ISCAS'19 [35]	This work
memory type	SRAM	RRAM	STT-MRAM	STT-MRAM	STT-MRAM	STT-MRAM	STT-MRAM
technology	28 nm	65 nm	45 nm	45 nm	45 nm	22 nm	65 nm
single-memory access MAC operations	YES	YES	YES	NO	NO	YES	YES
neural network supported	high-precision DNN	BNN	BNN	XNOR-Net	binary-weight CNNs	high-precision DNN	BNN
activations / weights	5b /1b	1b/1b	1b/1b	1b/1b	1b/1b	4b/5b	1b/1b
sub-array size	1152x256	128x128	128x256	NA	512x256	64x576	128x128
DAC required	YES	YES	NO	NO	NO	YES	NO
max throughput (TOPS)	6.144	NA	NA	NA	NA	NA	3.28
max energy efficiency (TOPS/W)	5796	141	NA	0.0169	0.455	NA	311 [a] (319 [b])
accuracy in MLP-MNIST / CNN-CIFAR-10	NA / 91.1%	98.43% / 86.08%	< 95% / N/A	NA/ NA	NA/ NA	98%/ 91%	98.42% / 80.01%

[a] This work with second-order SL boosting

[b] This work without SL boosting

in-memory compute frameworks that leverage available STT-MRAM arrays via simple enhancement of their periphery.

ACKNOWLEDGEMENT

REFERENCES

[1] M. Courbariaux, Y. Bengio, "Binarized Neural Networks: Training Neural Networks with Weights and Activations Constrained to +1 or -1," *arXiv*: 1602.02830, 2016.

[2] J. Kim, J. Koo, T. Kim, Y. Kim, H. Kim, S. Yoo, J. J. Kim, "Area-Efficient and Variation Tolerant In-Memory BNN Computing Using 6T SRAM Array," 2019 *Symposium on VLSI Circuits*.

[3] S. Jain, L. Lin, M. Alioto, "Broad-Purpose In-Memory Computing for Signal Monitoring and Machine Learning Workloads," *IEEE Solid-State Circuits Letters* (invited), vol. 3, pp. 394-397, Sept. 2020.

[4] R. Liu, et al., "Parallelizing SRAM Arrays with Customized Bit-Cell for Binary Neural Networks," in *Proc. of ACM/IEEE DAC*, 2018.

[5] X. Sun, X. Peng, P. Y. Chen, R. Liu, J. S. Seo, S. Yu, "Fully Parallel RRAM Synaptic Array for Implementing Binary Neural Network with (+1, -1) Weights and (+1, 0) Neurons," in Proc. of *ASPDAC*, 2018, pp. 574-579.

[6] W. H. Chen, K. X. Li, W. Y. Lin, et al., "A 65nm 1Mb Nonvolatile Computing-in-Memory ReRAM Macro with Sub-16ns Multiply-and-Accumulate for Binary DNN AI Edge Processors," in *ISSCC Dig. Tech. Papers*, 2018, pp. 494-496.

[7] K. V. Pham, T. V. Nguyen, S. B. Tran, H. K. Nam, M. J. Lee, B. J. Choi, S. N. Truong, K. S. Min, "Memristor Binarized Neural Networks," *Journal of Semiconductor Techology and Science*, vol.18, no.5, October, 2018.

[8] X. Sun, S. Yin, X. Peng, R. Liu, J. S. Seo, S. Yu, "XNOR-RRAM: A Scalable and Parallel Resistive Synaptic Architecture for Binary Neural Networks," in Proc. of *DATE*, 19-23 March 2018.

[9] S. Yu, *et al.,* "Binary Neural Network with 16 Mb RRAM Macro Chip for Classification and Online Training", *IEEE International Electron Devices Meeting (IEDM)*, 2016.

[10] A. P. Chowdhury, P. Kulkarni, M. N. Bojnordi, "MB-CNN: Memristive Binary Convolutional Neural Networks for Embedded Mobile Devices," *Journal of Low Power Electronics and Applications,* October 2018.

[11] H. Cai, *et al.,* "Interplay Bitwise Operation in Emerging MRAM for Efficient In-memory Computing," *CCF Transactions on High Performance Computing*, Vol. 2, pages 282–296 (2020).

[12] L. Chang, *et al.,* "PXNOR-BNN: In/With Spin-Orbit Torque MRAM Preset-XNOR Operation-Based Binary Neural Networks," *IEEE Trans. on VLSI Systems,* vol. 27, no.11, Nov. 2019.

[13] S. Resch, *et al.,* "PIMBALL: Binary Neural Networks in Spintronic Memory," *ACM Trans. Archit. Code Optim.* 16, 4, Article 41, Oct. 2019.

[14] S. Jain, A. Ranjan, K. Roy, A. Raghunathan, "Computing in Memory with Spin-Transfer Torque Magnetic RAM," *IEEE Trans. on VLSI Systems*, vol. 26, no. 3, pp. 470–483, Mar. 2018.

[15] N. Xu, *et al.,* "STT-MRAM Design Technology Co-Otimization for Hardware Neural Networks," *IEEE International Electron Devices Meeting (IEDM)*, 2018.

[16] S. Gao, *et al.,* "MRAM Acceleration Core for Vector Matrix Multiplication and XNOR-Binarized Neural Network Inference," *International Symposium on VLSI Technology, Systems and Applications (VLSI-TSA)*, 2020.

[17] Y. Pan et al., "A Multilevel Cell STT-MRAM-Based Computing In-Memory Accelerator for Binary Convolutional Neural Network," *IEEE Transactions on Magnetics*, vol. 54, no. 11, pp. 1-5, Nov. 2018.

[18] S. Angizi, Z. He, A. Awad and D. Fan, "MRIMA: An MRAM-Based In-Memory Accelerator," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 5, pp. 1123-1136, May 2020.

[19] Y. Seo, K. W. Kwon, "Area Optimization Techniques for High-Density Spin-Orbit Torque MRAMs", *Electronics* 2021, 10, 792.

[20] H. Cai, *et al.,* "Proposal of Analog In-Memory Computing with Magnified Tunnel Magnetoresistance Ratio and Universal STT-MRAM Cell," *arXiv*:2110.03937, 2021.

[21] Q. K. Trinh, S. Ruocco, M. Alioto, "Time-Based Sensing for Reference-Less and Robust Read in STT-MRAM Memories," *IEEE Trans. on CAS – part I*, vol. 65, no. 10, pp. 3338–3348, 2018.

[22] Q. K. Trinh, S. Ruocco, M. Alioto, "Novel boosted-voltage sensing scheme for variation-resilient STT-MRAM read," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 10, pp. 1652–1660, Oct. 2016.

[23] T. N. Pham, Q. K. Trinh, I. J. Chang, M. Alioto, "STT-MRAM Architecture with Parallel Accumulator for In-Memory Binary Neural Networks", in Proc. of *ISCAS*, 2021, pp. 1-5.

[24] D. Apalkov, *et al.,* "Spin-Transfer Torque Magnetic Random Access Memory (STT-MRAM)," *ACM Journal on Emerging Technologies in Computing Systems*, Vol. 9, No. 2, Article 13, May 2013.

[25] C. Augustine, N. N. Mojumder, X. Fong, S. H. Choday, S. P. Park, K. Roy, "Spin-Transfer Torque MRAMs for Low Power Memories: Perspective and Prospective," *IEEE Sensors Journal*, Vol.12, No.4, April 2012.

[26] A. Raychowdhury, D. Somasekhar, T. Karnik, V. De, "Design Space and Scalability Exploration of 1T-1STT MTJ Memory Arrays in the Presence of Variability and Disturbances," *IEEE International Electron Devices Meeting (IEDM)*, 2009.

[27] X. Fong, S. H. Choday, P. Georgios, C. Augustine, K. Roy. (Aug. 2013). *SPICE Models for Magnetic Tunnel Junctions Based on Monodomain*

Approximation. [Online] - Available: https://nanohub.org/resources/19048.

[28] C. J. Lin et al., "45nm low power CMOS logic compatible embedded STT-MRAM utilizing a reverse-connection 1T/1MTJ cell," *2009 IEEE Int. Electron Devices Meeting*, pp. 1–4, Dec. 2009.

[29] R. E. Walpole, R. H. Myers, S. L. Myers, K. Ye, *Probability & Statistics for Engineers & Scientists*, Prentice Hall, 2006.

[30] M. Alioto, G. Scotti, A. Trifiletti, "A Novel Framework to Estimate the Path Delay Variability on the Back of an Envelope via the Fan-Out-of-4 Metric," *IEEE Trans. on CAS – part I*, vol. 64, no. 8, pp. 2073-2085, Aug. 2017.

[31] Y. Kim, H. Kim, J. Kim, "Neural Network-Hardware Co-design for Scalable RRAM-Based BNN Accelerators," *arXiv*: 1811.02187, 2019.

[32] LeCun, Y. & Cortes, C. (2010). MNIST handwritten digit database. [Online] Available: http://yann.lecun.com/exdb/mnist/.

[33] Alex Krizhevsky master thesis "Learning Multiple Layers of Features from Tiny Images" (CIFAR-10 dataset), 2009.

[34] J. Lee, H. Valavi, Y. Tang and N. Verma, "Fully Row/Column-Parallel In-memory Computing SRAM Macro employing Capacitor-based Mixed-signal Computation with 5-b Inputs," *2021 Symposium on VLSI Circuits*, 2021, pp. 1-2.

[35] A. D. Patil, H. Hua, S. Gonugondla, M. Kang and N. R. Shanbhag, "An MRAM-Based Deep In-Memory Architecture for Deep Neural Networks," in Proc. of *ISCAS*, 2019, pp. 1-5.

Thi-Nhan Pham received the BS degree in Electronics and Telecommunications from Le Quy Don Technical University, Hanoi, Vietnam, in 2017. She is currently pursuing the M.S. and Ph.D. combined degree in the Department of Electronics and Radio Engineering, Kyung Hee University (KHU), Republic of Korea. Her research interests are emerging memory technologies, energy-efficient architecture and in-memory computing.

Quang-Kien Trinh (M' 15) received BS. (2007) and MS(2009) Degrees of Applied Mathematics and Physics respectively from Institute of Physics and Technologies (State University) in Moscow, Russia; he received Ph.D. (2018) in Computer Engineering from the National University of Singapore. Currently, he is a senior researcher and the deputy head of the Department of Microprocessor Engineering, Faculty of Radio-Electronics, Le Quy Don Technical University. He has authored or coauthored more than 35 publications. His research interest includes low-power integrated circuit design, emerging memory technologies, and hardware security.

Ik-Joon Chang (M'12) received the B.S. degree, with summa cum laude, in electrical engineering from Seoul National University, Seoul, Korea. And he acquired his M.S. and Ph.D. degree from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, in 2005 and 2009, respectively. After his graduation, he worked in Samsung flash design team for two years. Now, he is an associate professor of Kyunghee University, Korea.

Massimo Alioto (M'01–SM'07-F'16) is currently with the ECE Department, National University of Singapore, where he leads the Green IC Group, directs the Integrated Circuits and Embedded Systems area and the FD-fAbrICS research center. Previously, he held positions at the University of Siena, Intel Labs, University of Michigan Ann Arbor, University of California, Berkeley, and EPFL.

He has authored or coauthored more than 330 publications and four books, including *Enabling the Internet of Things-From Circuits to Systems* (Springer, 2017) and *Adaptive Digital Circuits for Power-Performance Range Beyond Wide Voltage Scaling* (Springer, 2020). His primary research interests include self-powered wireless integrated systems, widely energy-scalable systems, data-driven integrated systems and hardware security, among the others.

He is the Editor in Chief of the IEEE Transactions on VLSI Systems (2019-2022), and was the Deputy Editor in Chief of the IEEE Journal on Emerging and Selected Topics in Circuits and Systems (2018). In 2020-2021 he was Distinguished Lecturer of the IEEE Solid-State Circuits Society. In 2022-2023 and 2009-2010, he is/was Distinguished Lecturer of the IEEE Circuits and Systems Society, for which he was also member of the Board of Governors (2015-2020), and Chair of the "VLSI Systems and Applications" Technical Committee (2010-2012). He served as Guest Editor of several IEEE journal special issues, and Associate Editor of a number of IEEE and ACM journals. He is/was Technical Program Chair in a number of IEEE conferences (e.g., ISCAS 2023, SOCC, ICECS), and is currently in the IEEE "Digital architectures and systems" ISSCC subcommittee, and the ASSCC TPC. Prof. Alioto is an IEEE Fellow.