IEEE Journal on Exploratory Solid-State Computational Devices and Circuits

Received 1 February 2020; revised 6 March 2020; accepted 22 March 2020.
Date of publication 3 April 2020; date of current version 6 July 2020.

Digital Object Identifier 10.1109/JXCDC.2020.2985314

Analyzing the Effects of Interconnect Parasitics in the STT CRAM In-Memory Computational Platform

MASOUD ZABIHI, ARVIND K. SHARMA, MEGHNA G. MANKALALE, ZAMSHED IQBAL CHOWDHURY (Member, IEEE), ZHENGYANG ZHAO, SALONIK RESCH, ULYA R. KARPUZCU, JIAN-PING WANG (Fellow, IEEE), and SACHIN S. SAPATNEKAR (Fellow, IEEE)

Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA
CORRESPONDING AUTHOR: M. ZABIHI (zabih003@umn.edu)

This work was supported in part by NSF SPXunder Award CCF-1725420.

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

ABSTRACT This article presents a method for analyzing the parasitic effects of interconnects on the performance of the STT-MTJ-based computational random access memory (CRAM) in-memory computation platform. The CRAM is a platform that makes a small reconfiguration to a standard spintronics-based memory array to enable logic operations within the array. The analytical method in this article develops a methodology that quantifies the way in which wire parasitics limit the size and configuration of a CRAM array and studies the impact of cell- and array-level design choices on the CRAM noise margin. Finally, the method determines the maximum allowable CRAM array size under various technology considerations.

INDEX TERMS In-memory computing, spin-transfer torque computational random access memory (STT-CRAM), spintronics.

I. INTRODUCTION

HIGH energy and delay overheads for transferring data between processing and memory units have motivated intense interest in reducing the distance between memory and computation units. While near-memory computing places computational units at the periphery of memory for fast data access, true in-memory computing uses the memory array to perform computations through simple reconfiguration. True in-memory computing systems outperform near-memory and conventional computing systems because they massively reduce data communication energy and can provide high levels of parallelism. This article studies the impact of interconnect parasitics in the spin-transfer torque computational random access memory (STT-CRAM), a true in-memory processing platform [1]–[3]. Only a few prior works [4], [5] have attempted to incorporate the parasitic effects of interconnects in their analysis on in-memory computing, but their models did not consider all contributing factors based on realistic layout considerations.

The CRAM uses a small modification to the high-endurance MTJ-based [6] memory cell to enable true in-memory logic operations. In the CRAM, the segment resistances of wires that carry the current are significantly smaller than MTJ resistances. This can be falsely lead to this assumption that the interconnect parasitic effects are negligible. This article develops an analytical method based on the layout considerations, which is used to study the effects of design parameters on parasitics and performance, in order to build a robust CRAM design. The method considers multiple contributing factors simultaneously, e.g., reducing the access transistor resistance can potentially enhance the performance, but it also increases the area of the array and increases interconnect lengths, which can harm performance. We use this methodology to determine an optimal size for CRAM subarray.

In Section II, we provide an overview of the STT-CRAM. We then motivate the problem in Section III. Next, in Section IV, we develop a layout model for the CRAM in a FinFET technology, considering both the cell level and array level while also specifying metal layer usage. We develop the models for the impact of parasitics in Section V, evaluate the results of our analysis in Section VI, and conclude in Section VII.

II. OVERVIEW OF THE STT-CRAM

The core storage unit in an STT-CRAM is the STT-MTJ, which consists of a fixed layer, with a fixed magnetization orientation, and a free layer whose magnetization can be in one of the two possible states—parallel (P) and antiparallel (AP) [7]. The two states have different electrical resistances: the parallel state resistance $R_P < R_{AP}$ and the AP state resistance. We denote the P and AP states as logic 0 and 1, respectively. The MTJ state can be altered by passing a critical current of magnitude I_c through it in the appropriate direction.

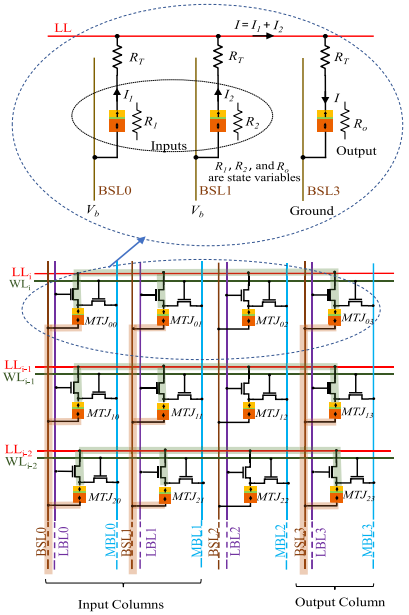


FIGURE 1. Structure of STT-CRAM array, highlighting the current paths during a logic operation with two inputs and one output.

Fig. 1 shows the structure of an STT-CRAM array, which uses a 2T1MTJ bit cell [1], [2], [8]. The configuration of the array is very similar to that of a standard 1T1MTJ STT-MRAM, and as in the case of the STT-MRAM, each bit cell is addressed using a memory wordline (WL). The additional transistor in the STT-CRAM is used for logic operations, and is turned ON by selecting the corresponding logic bitline (LBL).

The array can thus function in the memory or logic mode. In the memory mode, the transistor connected to WL is ON and read from (or write to) an MTJ is realized through memory bitline (MBL). Note that during the memory mode, the second transistor, which is connected to LBL, is OFF. In the logic mode, in the columns of the bit cells that correspond to inputs and the output, this LBL-connected transistor is turned ON. The access transistor connected to WL is OFF for all columns. This configuration allows the MTJs in the selected Bit cells in each row to be connected to a logic line (LL). By applying an appropriate voltage to the bit select lines (BSLs) of the input bit cells and grounding the BSL of the output, a state-dependent current, whose value depends on the resistance of MTJs and transistor resistance (R_T), flows through the output MTJ. If this current exceeds I_c, the output state (the resistance of the output) is altered; otherwise, it remains the same.

Different logic functions can be realized in the STT-CRAM by altering two parameters [1], [2], [8]: 1) the bias voltage (V_b) applied to the BSLs of the input MTJs and 2) the output preset state. In [2], for each gate, a range for V_b is calculated.

Next, we show how [2] derives an allowable range for V_b for a two-input AND gate, ignoring the parasitic effects of lines and transistors. An AND gate in each row can be realized by the configuration shown in the Top of Fig. 1, which highlights the path of current through the MTJs. Current I can be calculated by dividing V_b by the equivalent resistance $((R_1 + R_T)||(R_2 + R_T)) + R_o$, where "||" represents the equivalent resistance of parallel resistors. If $R_A = R_P + R_T$ and $R_B = R_{AP} + R_T$, the current for each input state is

$$I_{00} = V_b/(0.5R_A + R_B) \quad I_{11} = 2V_b/(3R_B)$$
$$I_{01} = I_{10} = V_b/((R_A||R_B) + R_B).$$

Since $R_P < R_{AP}$, $R_A < R_B$, implying that

$$I_{11} < I_{01} = I_{10} < I_{00}. \tag{1}$$

The output MTJ is preset to logic 1. For an AND gate, $I_{01} = I_{10} > I_c$, switching the output state from 1 to 0. From (1)

$$V_b > (R_A||R_B + R_B)I_c. \tag{2}$$

On the other hand, V_b cannot be too large; if it is, the output is switched regardless of the states of inputs, i.e., we must ensure that I_{11} must not be larger than I_c, i.e., from (1)

$$V_b < 3R_BI_c/2. \tag{3}$$

Considering these two constraints, we can present a bias voltage range for the AND gate. The voltage ranges and preset values for other logic functions can be obtained in a similar manner and are summarized in Table 1. The precise range of V_b is technology-dependent. To account for anticipated advances in spintronics [9], this article considers the MTJ specifications in today's technology and an advanced near-future technology [2]. In the rest of this article, we use today's and advanced MTJ parameters listed in Table 7 (see the Supplementary Material) for our calculations and evaluations.

TABLE 1. Bias voltage ranges and output preset values [2].

Gate (Preset)	V_{min} = Minimum V_b	V_{max} = Maximum V_b
BUFFER(1)	$(R_A + R_B)I_c$	$2R_BI_c$
NOT(0)	$2R_AI_c$	$(R_A + R_B)I_c$
AND(1)	$(R_A\|\|R_B + R_B)I_c$	$1.5R_BI_c$
NAND(0)	$(R_A\|\|R_B + R_A)I_c$	$(0.5R_B + R_A)I_c$
OR(1)	$(0.5R_A + R_B)I_c$	$(R_A\|\|R_B) + R_B)I_c$
NOR(0)	$1.5R_AI_c$	$(R_A\|\|R_B + R_A)I_c$
MAJ3(1)	$(0.5R_A\|\|R_B + R_B)I_c$	$(R_A\|\|0.5R_B + R_B)I_c$
$\overline{\text{MAJ3}}$(0)	$(0.5R_A\|\|R_B + R_A)I_c$	$(R_A\|\|0.5R_B + R_A)I_c$
MAJ5(1)	$((1/3)R_A\|\|0.5R_B+R_B)I_c$	$(0.5R_A\|\|(1/3)R_B+R_B)I_c$
$\overline{\text{MAJ5}}$(0)	$((1/3)R_A\|\|0.5R_B+R_A)I_c$	$(0.5R_P\|\|(1/3)R_B+R_A)I_c$

III. IMPACT OF WIRE PARASITICS

To show the impact of parasitics in a CRAM array, we consider a scenario where each row of the CRAM performs a BUFFER operation between Columns 1 and 10. The CRAM array can be built using either today's technology (today's CRAM) or using advanced technology (advanced CRAM). An electrical model of the current path is shown in Fig. 2; the bias voltage is applied between BSLs 1 and 10, and in each row, the current path goes through input and output MTJs, two access transistors, and a segment of LL. The model includes parasitic capacitances associated with each line segment and the transistor resistance. For the motivational example in Section III, built around Fig. 2, the transistors N_{fin} and N_{finger} are specific to the example and lead to the computed values of W_{cell} and L_{cell}. The remaining parameters are used throughout the rest of this article. The parameters used in the following motivational example are listed for both today's and advanced CRAMs in Table 2.

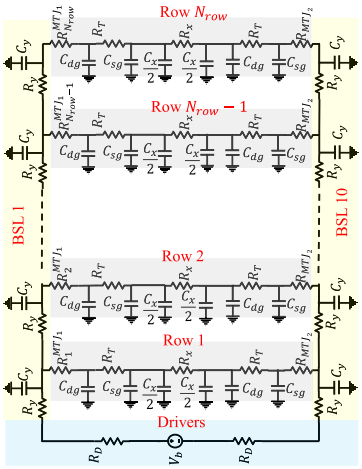


FIGURE 2. Circuit model of the current path for the implementation of BUFFER gates in CRAMs of various sizes.

TABLE 2. Parameters in the motivational example.

Parameter	Description	Today's CRAM	Advanced CRAM
N_{fin}	Number of fins	4	2
N_{finger}	Number of fingers	8	4
W_{cell}	Cell width	189nm	135nm
L_{cell}	Cell length	1323nm	675nm
R_T	Transistor resistance	0.178Ω	0.713Ω
d_{column}	Input-output distance	9	9
R_x	LL segment resistance	33.300Ω	25.100Ω
R_y	BSL segment resistance	0.026Ω	0.032Ω
C_x	LL segment capacitance	11.240fF	4.223fF
C_y	BSL segment capacitance	816aF	341aF
C_{gd}	Transistor g-d capacitance	320aF	320aF
C_{sg}	Transistor g-s capacitance	330aF	330aF
V_b	Applied bias voltage	670mV	96AmV

In the absence of wire parasitics and process variations, the bias voltage range for the implementation of a BUFFER gate can be obtained from Table 1. In [2], we reported the numerical values of bias voltages for different gates. For the BUFFER gate, under today's technology and advanced technology, the voltage ranges are 552–788 and 70–121 mV, respectively [2]. To maximize the noise margin (NM), we would choose the midpoint of the interval, $V_b = 670$ mV for today's CRAM and $V_b = 96$ mV for advanced CRAM, to implement the BUFFER. This voltage is applied through drivers at the edge of the CRAM array, where each driver has a resistance R_D. The CRAM rows in Fig. 2 are numbered from 1 (the nearest row to the driver) to N_{row} (the farthest row). When parasitics are accounted for, it can be seen that the path to row 1 encounters the fewest parasitics and that the path to row N_{row} the most, due to IR drop along the line. Thus, the voltage $V_b = 670$ mV in today's CRAM (and $V_b = 96$ mV in advanced CRAM) may not be significantly changed as it reaches the first row, but the voltage at row N_{row} may be significantly degraded.

Given the fixed voltage range (552–788 mV for today's CRAM and 70–121 mV for advanced CRAM) within which the BUFFER operates correctly, the entire array will operate correctly when the BSL voltage for Row 1 is at the maximum V_b value and the BSL voltage for Row i is at the minimum allowable V_b for the BUFFER. Thus, the maximum allowable voltage drop is the difference between the maximum and minimum V_b, i.e., 226 mV in today's CRAM and 51 mV in advanced CRAM for BUFFER, and a similarly calculated value from Table 1 for any other gate. In practice, the drop must be even smaller to allow for NMs.

TABLE 3. IR drop differential between the BSL voltage for the first row and the last row, and the *RC* delay of the transition.

# rows	Today's CRAM		Advanced CRAM	
	IR drop	RC delay	IR drop	RC delay
64	5.8mV	87fs	0.2mV	52fs
128	22.4mV	346fs	0.6mV	210fs
256	82.3mV	1.3ps	2.3mV	798fs
512	250.0mV	4.1ps	8.5mV	3.1ps
1024	507.9mV	10.3ps	38.0mV	10.6ps
2048	650.3mV	18.4ps	75.6mV	26.1ps

We consider six CRAM array configurations, each with a different number of rows, and use Table 3 to show the degradation of V_b as it reaches the farthest row for each of these configurations. If each row performs an identical operation in the worst case, it should carry an equal current I_{row} and the total voltage drop to the last row is

$$nI_{\text{row}}R_y + (n-1)I_{\text{row}}R_y + \cdots I_{\text{row}}R_y = n(n+1)/2\, I_{\text{row}}R_y$$

i.e., the IR drop increases quadratically with the number of rows. For the 64-row array (in both CRAMs), Table 3 shows that this IR drop is not large, but for arrays with 256 rows and larger in today's CRAM (and for 2048 rows in advanced CRAM), the IR drop is a significant fraction of V_b. The quadratic trend is seen between the first few rows, but the trend becomes subquadratic in the last few rows; this is due to the high voltage drop, the current supplied to that row is significantly less than supplied to the first row (e.g., for $N_{\text{row}} = 2024$, the voltage level at the last row is about 20 mV for today's CRAM (and 30 mV for advanced CRAMA), far less than V_{min} for a buffer). This invalidates the assumption in the abovementioned derivation that an equal current of I_{row} is supplied to each row.

Next, we compute the impact of *RC* parasitics on the CRAM delay. Defining the transition time as the time to 90% of the final value, Table 3 shows that this time is negligible in comparison with the nanosecond-range MTJ switching time. Thus, wire parasitics do not impact the delay but only the IR drop.

IV. LAYOUT MODELING

The key parameters that affect the IR drop are given as follows:

1) number of rows, for reasons illustrated in Table 3;
2) transistor resistance R_T, which is in series with the MTJ resistance; a higher value reduces the NM (the value of R_T can be reduced by increasing the transistor width, which may increase the cell area);
3) cell area, $A_{\text{cell}} = W_{\text{cell}}L_{\text{cell}}$ (where W_{cell} and L_{cell} are the cell width and length, respectively), which impacts the BSL and LL lengths, thus affecting IR drop;
4) cell aspect ratio, $\text{AR}_{\text{cell}} = (W_{\text{cell}}/L_{\text{cell}})$, which determines the BSL and LL lengths (a larger AR_{cell} makes the BSLs longer, causing increasing parasitics on them, while shortening LLs and reducing their parasitics);
5) configuration of BSLs and LLs, whose resistance can be reduced using a multimetal layer structure and whose length and width depend on other parameter choices.

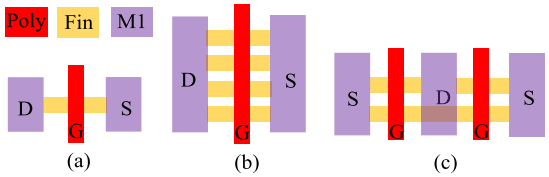


FIGURE 3. Layout of FinFET devices with (a) one fin and one finger, (b) four fins and one finger, and (c) two fins and two fingers.

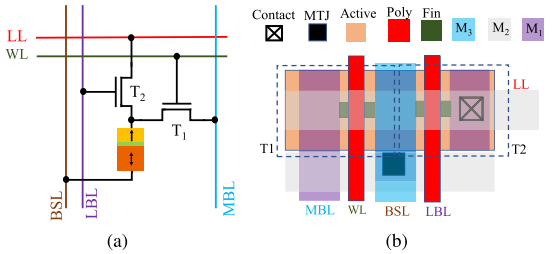


FIGURE 4. CRAM cell. (a) Schematic. (b) Layout.

A. LAYOUT OF A CRAM CELL

Designing an STT-MRAM with a FinFET access transistor can reduce the cell area and improve leakage power and reliability [10]. The design in this article is based on ASAP 7-nm Predictive PDK [11]. Fig. 3(a) shows the layout of a FinFET using a single fin. By applying the proper voltage to the gate (G), the current flows from drain (D) to the source (S) through the fin. By increasing the number of fins, the ON current increases and the drain–source resistance of the of FinFET decreases, at the cost of an increase in FinFET area. Such transistors can be drawn in multiple ways: a 4× FinFET is shown in Fig. 3(b) with four fins, or alternatively, in Fig. 3(c), using two fingers with two fins each. For the same transistor ON resistance, one can change the aspect ratio of the FinFET device by varying the numbers of fins and fingers.

Fig. 4 shows the schematic and layout of the CRAM cell using a one-fin one-finger FinFET. The source of T1 is connected to MBL and M1 is allocated for MBL routing; the poly in T1 is used for WL; the drain of T1 is connected to the MTJ, which is physically placed between M2 and M3. For T2, the drain is connected to LL, the poly is used locally for LBL, and the source is connected to the MTJ. In the layout, a horizontal M2 stripe is used for LL routing, and a vertical M3 stripe is used for BSL. Larger transistor sizes can be achieved by using multiple fins and fingers for each transistor, changing the cell dimension in the vertical and horizontal directions.

B. LAYOUT OF THE CRAM ARRAY

The CRAM cell can be tessellated into an array. Fig. 5 shows the layout of four adjacent CRAM cells (2 × 2), again using a one-fin one-finger FinFET, under ASAP7 design rules [11], [12]. For example, the minimum allowable active width is 27 nm; the poly length and pitch are 20 and 54 nm, respectively; the minimum active-to-active distance in our design can be 54 nm: under these constraints, the size of the smallest one-fin, one-finger CRAM cell is 108 nm × 189 nm. The addition of each fin increases the cell width (vertical dimension) by 27 nm, keeping the cell length (horizontal

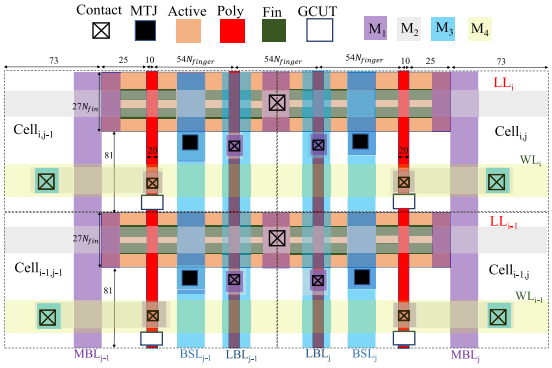


FIGURE 5. Layout of four adjacent CRAM cells in ASAP7.

dimension) fixed, while adding each finger increases the cell length by 108 nm, leaving the width fixed. The width and length for a cell with N_{fin} fins and N_{finger} fingers are

$$W_{\text{cell}} = 108 + 27(N_{\text{fin}} - 1) \tag{4}$$

$$L_{\text{cell}} = 189 + 162(N_{\text{finger}} - 1). \tag{5}$$

C. IMPACT OF LAYOUT CHOICES ON (A_{cell}, AR_{cell}, and R_T)

The choice of W_{cell} and L_{cell} can impact the cell area, $A_{\text{cell}} = W_{\text{cell}} \times L_{\text{cell}}$, and the cell aspect ratio, $\text{AR}_{\text{cell}} = W_{\text{cell}}/L_{\text{cell}}$. From (4) and (5), the following trends can be inferred as the numbers of fins and fingers are changed.

1) By increasing N_{fin} and N_{finger}, both W_{cell} and L_{cell} increase, increasing A_{cell}.
2) For fixed N_{fin}, the largest AR_{cell} has the lowest L_{cell}, i.e., $N_{\text{finger}} = 1$. If we fix N_{finger}, then by increasing N_{fin}, W_{cell} increases; thus, AR_{cell} increases.

Next, we study the impact of the numbers of fins and fingers on the transistor resistance, R_T, for both advanced CRAM and today's CRAM. We apply the nominal voltage of ASAP7 (0.7 V) to the FinFET gate, and for this value of gate-to-source voltage V_{gs}, we use the transistor I–V curve to determine the resistance corresponding to the drain-to-source current $I_{\text{ds}} = I_c$ (Table 7) required to switch an MTJ. Today's MTJ requires larger I_c than the advanced MTJ; hence, for the same N_{fin} and N_{finger}, R_T is larger for today's CRAM. The $N_{\text{fin}} = N_{\text{finger}} = 1$ case can deliver I_c for the advanced MTJ, but not for today's MTJ; larger sizes must be used for the latter. As expected, as N_{fin} and N_{finger} are increased, R_T reduces.

D. METAL LAYER CONFIGURATIONS AND SPECIFICATIONS

As seen in Section III, parasitics in the BSLs and LLs play a large part in limiting the allowable size of the CRAM array. To overcome this, we use a multimetal layer architecture for BSLs and LLs, shown in Fig. 6 for the four adjacent CRAM cells of Fig. 5. Here, metal layers M3, M5, M7, and M9 are allocated to the BSLs, and M2 and M4 are allocated to LLs, with vias connecting each type of line across layers. The interconnect specifications—the metal thickness (t_M), resistivity (ρ_M), minimum spacing (S_{min}), minimum width (W_{min}), and via parameters—are taken from [11] and [12].

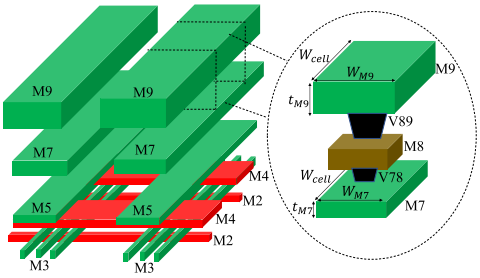


FIGURE 6. Configuration of BSLs and LLs. The green and red lines correspond to BSLs and LLs, respectively.

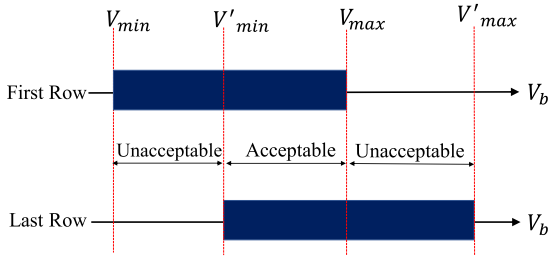


FIGURE 7. Required voltage ranges for the implementations of the same gate in the first row and the last row.

V. THEVENIN MODELING FOR EACH CRAM ROW

From Section III, V_b is degraded by IR drops as it reaches the last row. We use a Thevenin model to model the Thevenin voltage, V_{th} and resistance R_{th} at the last row of the CRAM (prior work [2] that neglects wire parasitics is a special case of our model where $V_{\text{th}} = V_b, R_{\text{th}} = 0$). We denote

$$\alpha_{\text{th}} = V_{\text{th}}/V_b. \tag{6}$$

Clearly, $\alpha_{\text{th}} \leq 1$ because V_{th} is a degraded version of V_b due to the voltage drop across the wire parasitics. We propose recursive expressions (see the Appendix) for R_{th} and α_{th} as functions of array parameters.

Fig. 7 shows the voltage ranges for the implementations of the same gate in the first row and the last row. For the first row, the effect of parasitics is negligible and the allowable voltage range lies within the minimum and maximum values specified in Table 1: we denote these as V_{min} and V_{max}, respectively. However, for an implementation of the same gate in the last row, we must consider R_{th} in series with the equivalent resistance across the MTJ devices in the last row and an applied voltage of V_{th}.

For example, for a BUFFER in the first row, the range of V_b is provided in Table 1. The last row is driven by with V_{th} in series with R_{th}, and the corresponding range is

$$\begin{aligned} &(R_A + R_B + R_{\text{th}})I_c \leq V_{\text{th}} \leq (2R_B + R_{\text{th}})I_c \\ \text{i.e., } &(R_A + R_B + R_{\text{th}})\frac{I_c}{\alpha_{\text{th}}} \leq V_b \leq (2R_B + R_{\text{th}})\frac{I_c}{\alpha_{\text{th}}} \end{aligned} \tag{7}$$

where the latter expression follows from (6). Since $\alpha_{\text{th}} < 1$, this implies that both the lower and upper bounds for V_b are higher in the last row than in the first row.

For each gate type, expressions for V'_{min} and V'_{max} can be modified from the parasitic-free cases mentioned in Table 1 as follows:

$$V'_{\text{min}} = \frac{V_{\text{min}} + R_{\text{th}} \times I_c}{\alpha_{\text{th}}}; \quad V'_{\text{max}} = \frac{V_{\text{max}} + R_{\text{th}} \times I_c}{\alpha_{\text{th}}}. \tag{8}$$

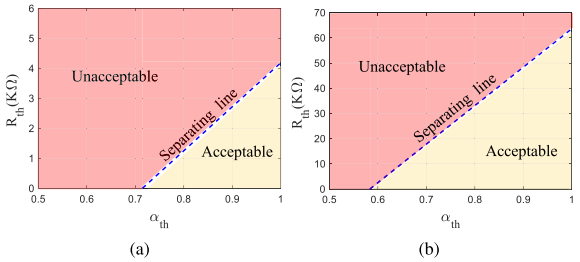


FIGURE 8. Separating lines of implementation for the AND gate in (a) today's CRAM and (b) advanced CRAM.

For the gate to function correctly in all rows, the allowable range of V_b is the intersection of the intervals $[V_{\text{min}}, V_{\text{max}}]$ and $[V'_{\text{min}}, V'_{\text{max}}]$; this is marked as the acceptable region for V_b in Fig. 7. Clearly, for correct functionality, these two intervals must have nonzero intersection, i.e., $V'_{\text{min}} < V_{\text{max}}$

For each gate type, this leads to a boundary ("separating line") between a functional and a nonfunctional implementation. From (8), the separating line constraint is

$$R_{\text{th}} < (V_{\text{max}} \times \alpha_{\text{th}} - V_{\text{min}})/I_c. \tag{9}$$

Fig. 8 shows the separating lines for today's and advanced MTJ technology in a R_{th} versus α_{th} plot, while (8) shows the equation for the separating line for each gate. The separating line demarcates the unacceptable region, where the gate functions incorrectly, from the acceptable region. It can be observed that the acceptable region of advanced CRAM is larger that of today's CRAM (note that the y-axis scale in the plots is different), providing more choices for designing parameters in advanced CRAM.

We define the NM as the range of allowable values for V_b. When all wire parasitics are zero, $\text{NM} = (V_{\text{max}} - V_{\text{min}})/V_{\text{mid}}$, where $V_{\text{mid}} = (V_{\text{max}} + V_{\text{min}})/2$, but in the presence of parasitics, this changes to

$$NM = \left(V_{\text{max}} - V'_{\text{min}}\right)/V'_{\text{mid}} \tag{10}$$

where $V'_{\text{mid}} = (V_{\text{max}} + V'_{\text{min}})/2$. Clearly, we desire $\text{NM} > 0$.

VI. RESULTS AND DISCUSSION

A. IMPACT OF CRAM PARAMETERS ON NM

1) EFFECT OF N_{row}

To examine how NM changes when the number of rows is altered, we fix the transistor configuration by choosing $N_{\text{fin}} = 2$, and $N_{\text{finger}} = 4$. This corresponds to an R_T of 570 Ω for today's CRAM and 597 Ω for the advanced CRAM. We also fix $\text{AR}_{\text{cell}} = 0.26$ and set $d_{\text{column}} = 10$, i.e., we consider the worst case NM when $d_{\text{column}} = 10$.

We analyze eight different cases with different N_{row} values (16, 32, 64, 128, 256, 512, 1024, and 2048) in today's and advanced CRAMs. Each case corresponds to a point in the R_{th}–α_{th} plane. For today's CRAM, the points corresponding to $N_{\text{row}} \leq 128$ are located in the acceptable area, i.e., the maximum allowable N_{row} under this choice of $\{R_T, AR_{\text{cell}}, d_{\text{column}}\}$ is 128. For the advanced CRAM, the acceptable points correspond to $N_{\text{row}} \leq 512$. The NMs are graphically shown in Fig. 9(a).

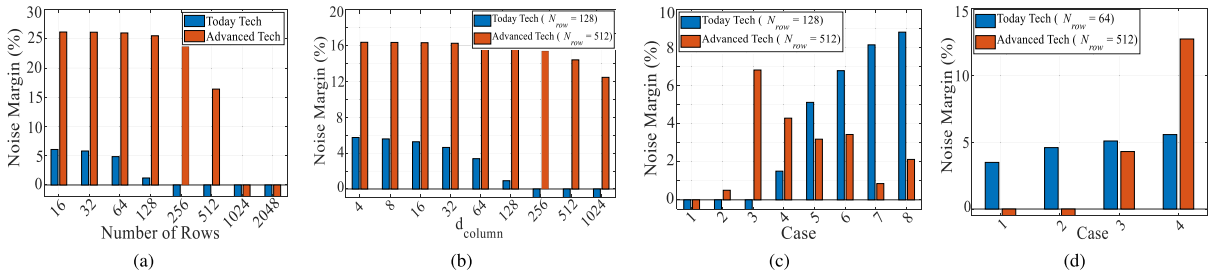


FIGURE 9. NM for an AND gate in today's and advanced CRAM, varying (a) N_{row}, (b) d_{column}, (c) R_T, and (d) AR_{cell}.

2) EFFECT OF d_{column}

By increasing the relative distance between the input columns and the output column (d_{column}), the parasitics associated with the LL in each row (R_x) increase. Fig. 9(b) shows NM for different cases with different d_{column} values in today's and advanced CRAMs. For the advanced CRAM, the value of d_{column} does not affect NM significantly for the shown values because the LL parasitic resistance, $R_x \ll R_{\text{MTJ}}$, the MTJ resistance to which it is connected in series. For today's CRAM, only $d_{\text{column}} \leq 64$ provide a positive NM because R_x is comparable with R_T for today's MTJs. High values of d_{column} create a large drop across the parasitics, causing V_b to be infeasible.

Similar trends are seen for the BUFFER, where the range of a copy operation is limited using today's CRAM. Thus, copy operations over large distances must be performed in multiple steps, adding to the energy and computation time.

3) EFFECT OF R_T

To analyze the effect R_T, we must consider that A_{cell} changes accordingly if we vary R_T. The choice of R_T can affect NM through two mechanisms: 1) directly, since a reduction in R_T increases the noise margin for an array of constant size and 2) indirectly, since a reduction in R_T increases the cell size, and hence the array size, thereby increasing line parasitics R_x and R_y.

In Table 4, we present eight cases where R_T gradually decreases from case 1 to case 8. For both advanced and today's CRAMs, case 1 has the smallest R_T (the largest A_{cell}), and case 8 has the largest R_T (the smallest A_{cell}). We choose W_{cell} and L_{cell} so that the AR_{cell} values are roughly constant (it is not possible to ensure equality since R_T is changed over a discrete space by altering N_{fin} and N_{finger}). We set N_{row} to 128 and 512, respectively, for today's and advanced technologies and set $d_{\text{column}} = 10$ for both cases.

Fig. 9(c) shows the NM for each of these cases. For today's technology, large R_T values (cases 1–3) cause negative NM, as in these cases, R_T values are comparable with today's MTJ resistances, and reducing R_T further improves NM. The direct mechanism is dominant here, and reducing R_T improves NM monotonically. In contrast, for advanced MTJs, there is a nonmonotone relationship as R_T is reduced. At first, NM improves due to the first mechanism, and then, it worsens due to the second mechanism. Part of the nonmonotonicity (e.g., between cases 5 and 6) can be attributed to the fact that AR_{cell} is not strictly constant in Table 4 (in fact, for case 1, for the advanced CRAM, the FinFET has $N_{\text{fin}} = N_{\text{finger}} = 1$, and the corresponding AR = 0.57 is the only option). Over the eight choices, one can choose case 3 as the optimal point that provides the best NM.

TABLE 4. Analyzing the effect of R_T.

	Today's CRAM			Advanced CRAM		
	R_T	A_{cell}	AR_{cell}	R_T	A_{cell}	AR_{cell}
1	5.99KΩ	0.020μm^2	0.4	5.73KΩ	0.020μm^2	0.57
2	1.72KΩ	0.038μm^2	0.31	2.87KΩ	0.029μm^2	0.40
3	1.04KΩ	0.047μm^2	0.38	1.90KΩ	0.038μm^2	0.31
4	0.76KΩ	0.058μm^2	0.31	0.95KΩ	0.047μm^2	0.38
5	0.49KΩ	0.067μm^2	0.37	0.63KΩ	0.057μm^2	0.41
6	0.36KΩ	0.082μm^2	0.43	0.48KΩ	0.067μm^2	0.38
7	0.29KΩ	0.096μm^2	0.36	0.35KΩ	0.081μm^2	0.43
8	0.23KΩ	0.110μm^2	0.42	0.23KΩ	0.110μm^2	0.42

TABLE 5. Analyzing the effect of AR_{cell}.

	Today's CRAM			Advanced CRAM		
	AR_{cell}	A_{cell}	R_T	AR_{cell}	A_{cell}	R_T
1	0.80	0.058μm^2	0.59KΩ	1.14	0.041μm^2	1.14KΩ
2	0.54	0.066μm^2	0.49KΩ	0.60	0.044μm^2	0.95KΩ
3	0.38	0.069μm^2	0.49KΩ	0.38	0.047μm^2	0.95KΩ
4	0.26	0.069μm^2	0.59KΩ	0.21	0.055μm^2	1.14KΩ

4) EFFECT OF AR_{cell}

We now vary AR_{cell} by changing N_{fin} and N_{finger} while keeping R_T and A_{cell} relatively fixed. As stated before, we set N_{row} to 128 and 512, respectively, for today's CRAM and the advanced CRAM; $d_{\text{column}} = 10$, and R_T and A_{cell} are kept roughly constant, to the extent possible in the discrete space of N_{fin} and N_{finger}.

Fig. 9(d) shows the results for the four cases Table 5. Cases with smaller AR_{cell}, which have shorter BSLs with lower parasitic resistances (R_y), have a larger NM value. Thus, an appropriate choice of AR_{cell} can improve the performance of the CRAM without area overhead. For example, in advanced CRAM, the NM for case 1, with the smallest AR_{cell}, is negative, but NM improves as AR_{cell} is increased.

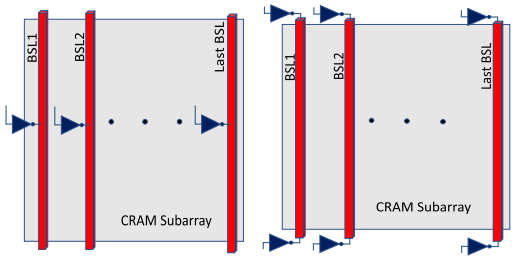
5) OPTIMAL DESIGN FOR EACH GATE

Table 6 evaluates the implementations of three types of arrays using both today's and advanced CRAMs with various degrees of versatility: Array 1 implements a basic set of combinational logic gates (INV, BUFFER, AND/NAND, and OR/NOR), Array 2 adds the MAJ3 and MAJ3 gates to this set, and Array 3 further adds MAJ5 and MAJ5. It is easily seen that for more versatile arrays, the array size is more constrained. The improvement from today's CRAM to the advanced CRAM is also visible: for example, today's CRAM cannot implement Array 3, regardless of array size [2].

To obtain the largest allowable size of N_{row}, we change the locations of the BSL drivers. Compared with the previous

TABLE 6. Optimal design options for arrays with different functionalities.

		N_{fin}, N_{finger}	R_T (KΩ)	A_{cell} (μm^2), AR_{cell}	N_{row}, d_{column}	Subarray Size
1	Advanced	4, 4	0.357	0.127, 0.280	512, 512	32KB
	Today's	5, 7	0.113	0.251, 0.186	128, 64	1024KB
2	Advanced	2, 6	0.476	0.135, 0.134	256, 256	4KB
	Today's	4, 9	0.101	0.281, 0.127	128, 16	256KB
3	Advanced	3, 9	0.171	0.267, 0.098	256, 64	2KB
	Today's	-	-	-	-	-



(a) (b)

FIGURE 10. Increasing N_{row} by inserting (a) 2× drivers in the middle of the array and (b) two 1× drivers at either end.

analysis where a driver was placed at one end of the array, we effectively double N_{row} by using a 2× driver in the middle of the array or by using two 1× drivers at either end of the array (see Fig. 10). The area overheads are modest.

Note that the constraint on N_{row} limits the array size but not the CRAM size; the overall CRAM consists of a tiled set of arrays, each with N_{row} rows, and all controlled by the same set of control signals. The choice of d_{column}, however, does not constrain the tile size but merely the computation distance. If the operands of a computation are at a distance $> d_{\text{column}}$ from each other, then they must be copied to new cells that are within the d_{column} limit. This is often not a problem; all the computations shown in [2] lie within the d_{column} constraint listed in Table 6. For this reason, practically, N_{row} is much more constraining than d_{column}.

VII. CONCLUSION

We have presented a methodology based on actual layout considerations for analyzing the parasitic effects in STT-CRAM. We have demonstrated that interconnect parasitics have a significant effect on CRAM performance and have developed a comprehensive model for analyzing this impact. Using this methodology, we have developed guidelines for the array size, N_{row}, and the maximum distance between the columns for an operation. We show that for both today's and advanced technologies, CRAM cell layouts with smaller aspect ratios are desirable, as this helps control critical BSL parasitics. Reducing access transistor resistance is important for today's technology but is not a significant factor for advanced technologies. For the SHE-CRAM [13], a similar analysis shows that interconnect parasitics are not significant as the current values are much smaller.

APPENDIX

A. THEVENIN MODEL FOR ONE-INPUT GATES

We derive the recursive expressions for R_{th} and α_{th} for each row in the CRAM array. The exposition here derives an expression for V_{th}, and α_{th} is trivially obtained from (6).

Within the footprint area of a CRAM cell, we define R_y, R_x, and R_{Via} as lumped resistances for BSL segment, LL segment, and vias, respectively. Fig. 11 shows the equivalent simplified circuit of the path for the implementation of BUFFER (or NOT) gates on CRAM rows. Row i is separated from its predecessor by resistances R_y at each end and is connected through R_{via} to an input MTJ cell, represented by $R_i^{\text{MTJ}_1}$ and R_T. The input cell is connected to the output cell, d_{column} rows away, through a resistance R_x, and the output cell is represented by a transistor resistance R_T in series with a preset MTJ resistance, R_{MTJ_2}. The resistances in the last row are rearranged to create a two-port structure consisting of the MTJ resistances so that the rest of the network can be modeled using the Thevenin equivalents (R_{th} and V_{th}).

Depending on the states of the inputs in different rows, which are application-dependent, the resistances of the MTJs, and hence the Thevenin parameters, change and typically vary in different rows. To provide a robust design, we consider the worst case in which the combination of the values of inputs in different rows results in the worst voltage drop across the MTJs of the last row. This worst case corresponds to the worst case current, which is drawn when all inputs MTJs in all rows are in the parallel state, creating $N_{\text{row}} - 1$ paths with the lowest possible resistance possible between the input and output BSLs.

As explained in Section IV-D, the BSL and LL lines have a multilayer structure. The metal layer resistances are considered to be in parallel, and R_y and R_x are expressed by

$$R_y^{-1} = R_{M_3}^{-1} + R_{M_5}^{-1} + R_{M_7}^{-1} + R_{M_9}^{-1} \tag{11}$$

$$R_x^{-1} = d_{\text{column}}^{-1}\left(R_{M_2}^{-1} + R_{M_4}^{-1}\right) \tag{12}$$

where d_{column} is the number of wire segments between the input and output columns. The resistance R_{M_k} is given by: $R_{M_k} = (\rho_{M_k} L_{M_k})/(t_{M_k} W_{M_k})$, where ρ_{M_k}, L_{M_k}, t_{M_k}, and W_{M_k} are, respectively, the resistivity, length, thickness, and width in metal layer k. The equivalent resistance of the vias, R_{Via}, depends on the configuration of the CRAM cell; a larger CRAM cell contains more vias in its footprint area, as a consequence of which R_{Via} is smaller. The number of vias between two metal layers in the footprint area of a CRAM cell can be calculated based on the via characteristics in Table 9 (see the Supplementary Material), as the parallel resistance of the available number of vias.

The abovementioned equivalent resistive introduces a minor simplification because the parallel wires do not coincide at a single point, but the vias are a small distance apart. HSPICE simulations show less than 0.5% error.

To calculate R_{th} and V_{th}, we derive recursive expressions. For conciseness, we define the resistance R_{row_i} of row i as

$$R_{\text{row}_i} = 2(R_{\text{Via}} + R_T) + R_x + R_i^{\text{MTJ1}} + R_{\text{MTJ2}}. \tag{13}$$

The input logic value depends on the application, i.e., R_i^{MTJ1} can be either R_P or R_{AP}. For each gate, the output resistance is a known preset value; for a Buffer (NOT) gate implemented across all rows, the preset is 1 (0). Therefore, R_{MTJ2} for the Buffer and NOT gates are R_{AP} and R_P, respectively.

We can obtain R_{th}, using the notations in Fig. 11(b), as

$$R_{\text{th}} = 2(R_y + R_{\text{Via}}) + R_x + R_{N_{\text{row}}-1} \tag{14}$$

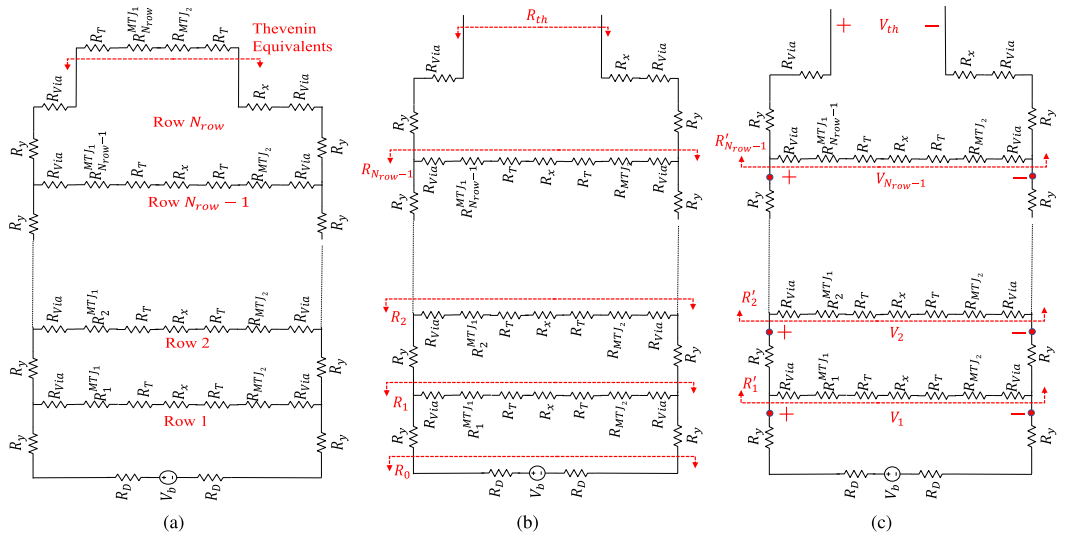


FIGURE 11. (a) Circuit model for one-input gates, showing the observation point for calculating Thevenin equivalent. Notations used in the chain of rows for defining (b) Thevenin resistance (R_{th}) and (c) Thevenin voltage (V_{th}).

where $R_{N_{\text{row}}-1}$ is calculated using the recursive expression

$$R_i = \frac{R_{\text{row}_i}(R_{i-1} + 2R_y)}{R_{\text{row}_i} + R_{i-1} + 2R_y}. \tag{15}$$

The base case corresponds to the driver row that precedes the first row and is $R_0 = 2\,R_D$, as shown in Fig. 11(a).

To compute V_{th}, as shown in Fig. 11(c), we first compute the intermediate variable R'_j, which corresponds to the effective downstream resistance (away from the source) seen from node j. The computation proceeds in a recursive fashion from the last row toward the first as

$$R'_{j-1} = \frac{R_{\text{row}_{j-1}}(R'_j + 2R_y)}{R_{\text{row}_{j-1}} + R'_j + 2R_y} \tag{16}$$

with the base case $R'_{N_{\text{row}}-1} = R_{\text{row}_{N_{\text{row}}-1}}$.

Having computed R'_j, we may now compute $V_{\text{th}} = V_{N_{\text{row}}}$ using a recursive computation on V_i

$$V_j = \frac{R'_j}{2R_y + R'_j} V_{j-1} \tag{17}$$

in which $2 \le j \le N_{\text{row}} - 1$ and the base case is

$$V_1 = \frac{R'_1}{R'_1 + 2R_y + 2R_D} V_b. \tag{18}$$

B. THEVENIN MODEL FOR N-INPUT GATES

The one-input model can easily be extended for the case where the number of inputs $N > 1$. For this case, we have N columns of input MTJs that connect to an output MTJ in each row; the worst case corresponds to the scenario where all inputs are adjacent to each other and d_{column} columns away from the output. As a simplification, we assume that all units are equally far and that the resistance to the output for each is R_x; this is reasonable because the horizontal resistance between the adjacent columns is negligible. In this case, in column i, N parallel structures, each consisting of series connections of R_y, R_{via}, $R_i^{\text{MTJ}_1}$, and R_T, connect through R_x to the output cell, modeled as a series connection of R_T and R_{MTJ_2}. We generalize (13) to

$$R_{\text{Row}_i} = \left(1 + \frac{1}{N}\right)(R_{\text{Via}} + R_T) + R_x + \frac{R_i^{\text{MTJ1}}}{N} + R_{\text{MTJ2}}. \tag{19}$$

Proceeding similar to the one-input case, we generalize (14) to compute R_{th} as

$$R_{\text{th}} = \left(1 + \frac{1}{N}\right)(R_y + R_{\text{Via}}) + R_x + R_{N_{\text{row}}-1} \tag{20}$$

where $R_{N_{\text{row}}-1}$ can be obtained using the recursion

$$R_i = \frac{R_{\text{Row}_i}(R_{i-1} + \left(1 + \frac{1}{N}\right)R_y)}{R_{\text{Row}_i} + R_{i-1} + \left(1 + \frac{1}{N}\right)R_y} \tag{21}$$

where $1 \le i \le N_{\text{row}} - 1$ and the base case is $R_1 = (1 + (1/N))R_D$, corresponding to the fact that each input line is driven by a source V_b with a series resistance R_D to the first via.

Similarly, one can recursively compute V_{th}. Analogously to (16), we first compute R'_j recursively, from the last row to the first, as

$$R'_{j-1} = \frac{R_{\text{Row}_{j-1}}(R'_j + \left(1 + \frac{1}{N}\right)R_y)}{R_{\text{Row}_{j-1}} + R'_j + \left(1 + \frac{1}{N}\right)R_y)} \tag{22}$$

where the base case is $R_{N_{\text{row}-1}} = R_{\text{Row}_{N_{\text{row}}-1}}$.

We can then compute $V_{\text{th}} = V_{N_{\text{row}}}$ recursively using the following recursion for V_j:

$$V_j = \frac{R'_j}{\left(1 + \frac{1}{N}\right)R_y + R'_j} V_{j-1} \tag{23}$$

in which $2 \le j \le N_{\text{row}} - 1$ and the base case is

$$V_1 = \frac{R'_1}{R'_1 + \left(1 + \frac{1}{N}\right)R_y + \left(1 + \frac{1}{N}\right)R_D} V_b. \tag{24}$$

REFERENCES

[1] J.-P. Wang and J. D. Harms, "General structure for computational random access memory (CRAM)," U.S. Patent 9 224 447 B2, Dec. 29, 2015. [Online]. Available: https://www.google.com/patents/US9224447

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

[3] M. Zabihi *et al.*, "True in-memory computing with the CRAM: From technology to applications," in *Proc. Great Lakes Symp. VLSI (GLSVLSI)*, 2019, p. 379.

[4] Y. Jeong, M. A. Zidan, and W. D. Lu, "Parasitic effect analysis in memristor-array-based neuromorphic systems," *IEEE Trans. Nanotechnol.*, vol. 17, no. 1, pp. 184–193, Jan. 2018.

[5] N. M. G., F. Lalchhandama, K. Datta, and I. Sengupta, "Modelling and simulation of non-ideal MAGIC NOR gates on memristor crossbar," in *Proc. 8th Int. Symp. Embedded Comput. Syst. Design (ISED)*, Dec. 2018, pp. 124–128.

[6] J.-P. Wang *et al.*, "A pathway to enable exponential scaling for the beyond-CMOS era," in *Proc. 54th Annu. Design Autom. Conf. (DAC)*, 2017, pp. 1–6.

[7] J. Kim *et al.*, "Spin-based computing: Device concepts, current status, and a case study on a high-performance microprocessor," *Proc. IEEE*, vol. 103, no. 1, pp. 106–130, Jan. 2015.

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

[9] A. Hirohata *et al.*, "Roadmap for emerging materials for spintronic device applications," *IEEE Trans. Magn.*, vol. 51, no. 10, pp. 1–11, Oct. 2015.

[10] A. Shafaei, Y. Wang, and M. Pedram, "Low write-energy STT-MRAMs using FinFET-based access transistors," in *Proc. IEEE 32nd Int. Conf. Comput. Design (ICCD)*, Oct. 2014, pp. 374–379.

[11] L. T. Clark *et al.*, "ASAP7: A 7-nm finFET predictive process design kit," *Microelectron. J.*, vol. 53, pp. 105–115, Jul. 2016.

[12] L. T. Clark, V. Vashishtha, D. M. Harris, S. Dietrich, and Z. Wang, "Design flows and collateral for the ASAP7 7nm FinFET predictive process design kit," in *Proc. IEEE Int. Conf. Microelectron. Syst. Edu. (MSE)*, May 2017, pp. 1–4.

[13] M. Zabihi *et al.*, "Using spin-Hall MTJs to build an energy-efficient in-memory computation platform," in *Proc. 20th Int. Symp. Qual. Electron. Design (ISQED)*, Mar. 2019, pp. 52–57.

[14] G. Jan *et al.*, "Demonstration of fully functional 8Mb perpendicular STT-MRAM chips with sub-5ns writing for non-volatile embedded memories," in *Symp. VLSI Technol. (VLSI-Technol.): Dig. Tech. Papers*, Jun. 2014, pp. 1–2.

[15] H. Maehara *et al.*, "Tunnel Magnetoresistance above 170% and resistance–area product of 1 Ω (μm) 2 attained by in situ annealing of ultra-thin MgO tunnel barrier," *Appl. Phys. Express*, vol. 4, no. 3, Mar. 2011, Art. no. 033002.

[16] H. Noguchi *et al.*, "7.5 A 3.3 ns-access-time 71.2 μW/MHz 1Mb embedded STT-MRAM using physically eliminated read-disturb scheme and normally-off memory architecture," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2015, pp. 1–3.

MASOUD ZABIHI received the B.Sc. degree in electrical engineering from the University of Tabriz, and the M.S. degree in electrical engineering from the Sharif University of Technology, Iran. He is currently pursuing the Ph.D. degree with the University of Minnesota, Minneapolis, MN, USA.

His research interests include in-memory computing, emerging memory technologies, and VLSI design automation.

ARVIND K. SHARMA received the Ph.D. degree from IIT Roorkee, Roorkee, India, in 2018.

He is currently a Post-Doctoral Associate at the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA. His current research interests include device physics, circuit device interaction, layout automation, and variability-aware circuit design.

MEGHNA G. MANKALALE received the B.E. degree from Visvesvaraya Technological University, India, in 2007, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA, in 2020.

She has worked as a Research and Development Engineer at the Electronic Design Automation Group, IBM, India, from 2007 to 2013. She currently works as a Sr. R&D Engineer at Synopsys.

ZAMSHED IQBAL CHOWDHURY (Member, IEEE) is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA.

He is a Faculty Member (on leave) at Jahangirnagar University, Bangladesh. His primary research interests include emerging non-volatile memory technologies, application-specific hardware design, and computer performance analysis.

ZHENGYANG ZHAO received the B.S. degree in electrical engineering from Xi'an Jiaotong University, China. He is currently pursuing the Ph.D. degree in electrical engineering with the University of Minnesota, Minneapolis, MN, USA.

His research focuses on the development of spintronic devices for memory and computing applications.

SALONIK RESCH received the bachelor's degree in computer engineering from the University of Minnesota, Minneapolis, MN, USA, in 2016, where he is currently pursuing the Ph.D. degree.

His research interests include processing-in-memory, intermittent computing, and quantum computing.

ULYA R. KARPUZCU received the M.S. and Ph.D. degrees in computer engineering from the University of Illinois, Urbana-Champaign, IL, USA.

She is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA. Her research interests span the impact of technology on computing, energy-efficient computing, application domain specialized architectures, approximate computing, and computing at ultra-low voltages.

JIAN-PING WANG (Fellow, IEEE) received the Ph.D. degree in magnetism from the Institute of Physics, CAS, in 1995. He finished his post-doctoral training from the National University of Singapore in 1996.

He was the Director of the Center for Spintronic Materials, Interfaces and Novel Architectures (C-SPIN). He is currently the Robert F. Hartmann Chair and a Distinguished McKnight University Professor of Electrical and Computer Engineering at the University of Minnesota. He is also the Director of the Center for Spintronic Materials for Advanced Information Technology (SMART), one of two SRC/NIST nCORE research centers.

Dr. Wang received the INSIC Technical Award in 2006 and the SRC Technical Excellence Award in 2019.

SACHIN S. SAPATNEKAR (Fellow, IEEE) received the B.Tech. degree from IIT Bombay, the M.S. degree from Syracuse University, and the Ph.D. degree from the University of Illinois.

He is on the faculty at the University of Minnesota, Minneapolis, MN, USA, where he holds the Distinguished McKnight University Professorship and the Robert and Marjorie Henle Chair in the Department of Electrical and Computer Engineering.

Dr. Sapatnekar is a fellow of the ACM. He has received eight conference Best Paper Awards, a Best Poster Award, two ICCAD 10-year Retrospective Most Influential Paper Awards, the SRC Technical Excellence Award, and the SIA University Research Award.