IMFlexCom: Energy Efficient In-Memory Flexible Computing Using Dual-Mode SOT-MRAM

FARHANA PARVEEN, SHAAHIN ANGIZI, and DELIANG FAN, University of Central Florida

In this article, we propose an In-Memory Flexible Computing platform (IMFlexCom) using a novel Spin Orbit Torque Magnetic Random Access Memory (SOT-MRAM) array architecture, which could work in dual mode: memory mode and computing mode. Such intrinsic in-memory logic (AND/OR/XOR) could be used to process data within memory to greatly reduce power-hungry and long distance massive data communication in conventional Von Neumann computing systems. A comprehensive reliability analysis is performed, which confirms ~90mV and ~10mV (worst-case) sense margin for memory and in-memory logic operation in variations on resistance-area product and tunnel magnetoresistance. We further show that sense margin for in-memory logic computation can be significantly increased by increasing the oxide thickness. Furthermore, we employ bulk bitwise vector operation and data encryption engine as case studies to investigate the performance of our proposed design. IMFlexCom shows ~35× energy saving and ~18× speedup for bulk bitwise in-memory vector AND/OR operation compared to DRAM-based in-memory logic. Again, IMFlexCom can achieve 77.27% and 85.4% lower energy consumption compared to CMOS-ASIC- and CMOL-based Advanced Encryption Standard (AES) implementations, respectively. It offers almost similar energy consumption as recent DW-AES implementation with 66.7% less area overhead.

CCS Concepts: • **Computer systems organization → Reconfigurable computing**; • **Hardware → Memory and dense storage**; *Spintronics and magnetic technologies*;

Additional Key Words and Phrases: In-memory computing, memory architecture, SOT-MRAM, giant spin Hall effect, magnetic tunnel junction

ACM Reference format:
Farhana Parveen, Shaahin Angizi, and Deliang Fan. 2018. IMFlexCom: Energy Efficient In-Memory Flexible Computing Using Dual-Mode SOT-MRAM. *J. Emerg. Technol. Comput. Syst.* 14, 3, Article 35 (October 2018), 18 pages.
https://doi.org/10.1145/3223047

1 INTRODUCTION

Modern information and computing sectors require efficient hardware platforms to support memory-oriented processing at exascale (10^{18} bytes/s) (Chi et al. 2016). Traditional Von Neumann architectures incorporating separation of memory and computing units suffer from long memory access latency, significant congestion at I/Os, limited memory bandwidth, and huge leakage power consumption due to massive data transfer between processor and memory. These critical bottlenecks of Von Neumann machines degrade the system performance and efficiency due to the

This work is supported in part by the National Science Foundation under Grant No. 1740126 and Semiconductor Research Corporation nCORE.
Authors' addresses: F. Parveen, S. Angizi, and D. Fan, Department of Electrical and Computer Engineering, 4328 Scorpius Street, University of Central Florida, Orlando, FL 32816-2362, USA.

https://doi.org/10.1145/3223047

exasperating challenge entitled "memory wall" or "power wall" (Wulf and McKee 1995), especially for big-data driven applications (Wang et al. 2015). Hence, nowadays near-memory or in-memory computing platforms (Ahn et al. 2015; Angizi et al. 2017a; Chi et al. 2016; Draper et al. 2002; Fan 2016; Farmahini-Farahani et al. 2015; Guo et al. 2013; He et al. 2017; Li et al. 2016; Nair et al. 2015; Oskin et al. 1998; Patterson et al. 1997a; Pawlowski 2011; Riedel et al. 2001; Seshadri et al. 2015; Wang and Harms 2015; Zhang et al. 2014; Zhu et al. 2012) are being investigated to integrate memory and logic, thus offering energy efficient massive data processing within memory itself.

DRAM-based memories (Darper et al. 2002; Kogge 1994; Off 1992; Patterson et al. 1997b; Seshadri et al. 2015) have been proposed for in-memory pre-processing of data. However, DRAM has some inherent limitations such as volatility, need for repeated refresh, destructive read/compute operation, slow charging and discharging latency, and so on, which degrade the system performance. Furthermore, for embedding logic functionality within memory chip, the complexity and cost of DRAM chip increases significantly. In view of these, a more efficient, compact and non-volatile in-memory computing platform that is capable of simultaneously offering density optimized memory and performance optimized computing facility, is one of the most important focus points for modern computing platform design.

Many recent researches can provide promising in-memory computing facilities using emerging non-volatile memories (NVM) (Jain et al. 2017; Kang et al. 2016; Li et al. 2016; Prenat et al. 2016; Zhang et al. 2017). Spin-transfer torque magnetic random access memory (STT-MRAM) is one of the leading candidates for next-generation high-performance on-chip universal non-volatile memories to replace current CMOS-based memory solutions owing to its unique features (Chen et al. 2017; Fong et al. 2016; Guo et al. 2010, 2013; Jain et al. 2017; Kang et al. 2017; Kim et al. 2012; Seo et al. 2016), for example, non-volatility, zero standby leakage, high write/read speed, excellent endurance, compatibility with CMOS fabrication process, scalability and excellent integration density, and so on. However, STT-MRAM based on 2-terminal Magnetic Tunnel Junction (MTJ) poses some challenging limitations. First, it has high energy consumption due to high write current requirements (Seo et al. 2016); and second, read disturbance failure due to the shared read/write current path (Del Bel et al. 2014; Fong et al. 2014; Kang et al. 2015; Kwon et al. 2015). To avoid the above issues, Spin Orbit Torque Magnetic Random Access Memory (SOT-MRAM) has been introduced (Parveen et al. 2017; Prenat et al. 2016), which requires lower write current due to spin-orbit coupling and offers more reliable operation by separating the read and write current paths.

Pinatubo (Li et al. 2016) proposes a generalized in-memory computing platform using non-volatile memories exploiting the parallel resistance between two or more adjacent memory cells. However, it is difficult to differentiate the parallel resistances for two logic states, unless there is a very high difference between ON and OFF resistances of the storage element. Furthermore, they proposed to use capacitors to perform logic XOR operation which takes multiple clock cycles. Jain et al. (2017) also has the same limitations of exploiting parallel resistance, multiple cycles for XOR. Additionally, they have restrictions on the locations of the two cells on which the logic operations are to be conducted, as these need to be on adjacent rows. Parveen et al. (2017) also comes with the shortcoming of such operand locality problem due to the restriction on two operand cells' locations. Besides, it also has some other drawbacks. For example, it could not implement XOR and requires one additional transistor per two cells to activate the computing mode.

In this article, we propose an In-Memory Flexible Computing (IMFlexCom) platform using a novel SOT-MRAM array architecture that can offer an energy efficient computing facility within the non-volatile memory array without sacrificing memory density. Our proposed architecture is capable of performing any two input Boolean logic functions (AND/OR/XOR) between any two cells residing in any odd row and any even row within the memory array, which defines its flexibility. Major contributions of our proposed IMFlexCom are the following:

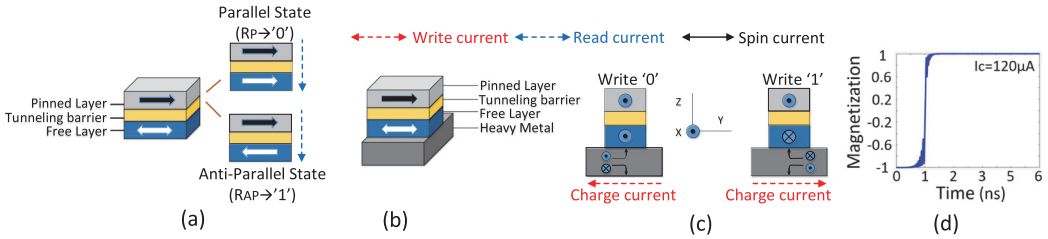


Fig. 1. (a) MTJ structure, (b) SOT MRAM device structure, (c) spin Hall effect (spin current generation due to spin Hall effect), and (d) normalized magnetization dynamics of the free FM layer.

—Novel array circuit design to incorporate an in-memory computing platform using SOT-MRAM, with two modes of operations: memory mode and computing mode.
—Capable of working in either mode (memory or computing) with two transistors per memory cell.
—Every two input Boolean logic function, AND/NAND, OR/NOR, XOR/XNOR, could be implemented within one sensing clock cycle, without prolonged latency.
—Boolean logic between any two cells residing in any odd row and any even row within the memory array could be computed, which solves the *operand locality* issue.

We investigated the performance of our proposed design by implementing a bulk bitwise Boolean vector logic operation and data encryption engine using the Advanced Encryption Standard (AES) algorithm. The rest of the article is organized as follows. Section 2 simply describes the SOT-MRAM device structure. The proposed novel SOT-MRAM array architecture is presented in Section 3. The performance evaluations from device to system are analyzed in Section 4 and Section 5. The case studies are presented along with their performance evaluation in Section 7. Section 8 concludes the article.

2 SOT-MRAM DEVICE STRUCTURE

In this section, we present an SOT-MRAM device structure and operation (Prenat et al. 2015; Seo et al. 2016; Zhang et al. 2015). Figure 1(a) shows an MTJ structure, while Figure 1(b) shows the structure of an SOT-MRAM device (Parveen et al. 2017).

2.1 Magnetic Tunnel Junction

A Magnetic Tunnel Junction (MTJ) is formed with a pinned and a free ferromagnetic (FM) layer (CoFeB (Honjo et al. 2015)) sandwiching a tunneling barrier (MgO (Autès et al. 2010)). MTJ works as a storage element for the memory cell. The resistance states, High and Low, are used to encode logic "1" and "0," respectively. Due to the tunneling magnetoresistance (TMR) effect (Autès et al. 2010), the resistance of MTJ is High, R_{AP} (/Low, R_P) when the magnetization of two ferromagnetic layers are anti-parallel, AP (parallel, P), as shown in Figure 1(a). The free layer magnetization could be manipulated by applying an external magnetic field, or through the current induced spin-transfer torque (STT) (Fong et al. 2016), or through spin-orbit torque (SOT) produced by giant spin Hall effect (Prenat et al. 2015). This resistance could be measured by injecting read current through MTJ as shown in Figure 1(a). In this work, the MTJ is modeled in Verilog-A, using Non-Equilibrium Green's Function (NEGF)–Landau-Lifshitz-Gilbert (LLG) equation solution for spin to charge interface (Fong et al. 2011).

Table 1. Device Parameters Used for Modeling

Symbol	Quantity	Value
α	Damping Coefficient	0.0122
D_x, D_y, D_z	Demagnetization Factors	0.066, 0.911, 0.022
M_S	Saturation Magnetization	0.85×10^6A/m
ρ_{SHM}	Resistivity of SHM	$200\mu\Omega$ cm
θ_{SHM}	Spin Hall Angle	0.3
$\lambda_s f$	Spin Flip Length	1.4×10^{-9}m
γ	Gyromagnetic Ratio	1.76×10^{11}Am^2/Js
t_{MgO}	MgO Thickness	1.5 nm
$(t \times W \times L)_{FM}$	Magnet Dimensions	1.5nm $\times$ 80nm $\times$ 40nm
$(W \times L)_{MTJ}$	MTJ Dimensions	80nm $\times$ 40nm
$(t \times W \times L)_{SHM}$	SHM Dimensions	2.8nm $\times$ 80nm $\times$ 50nm

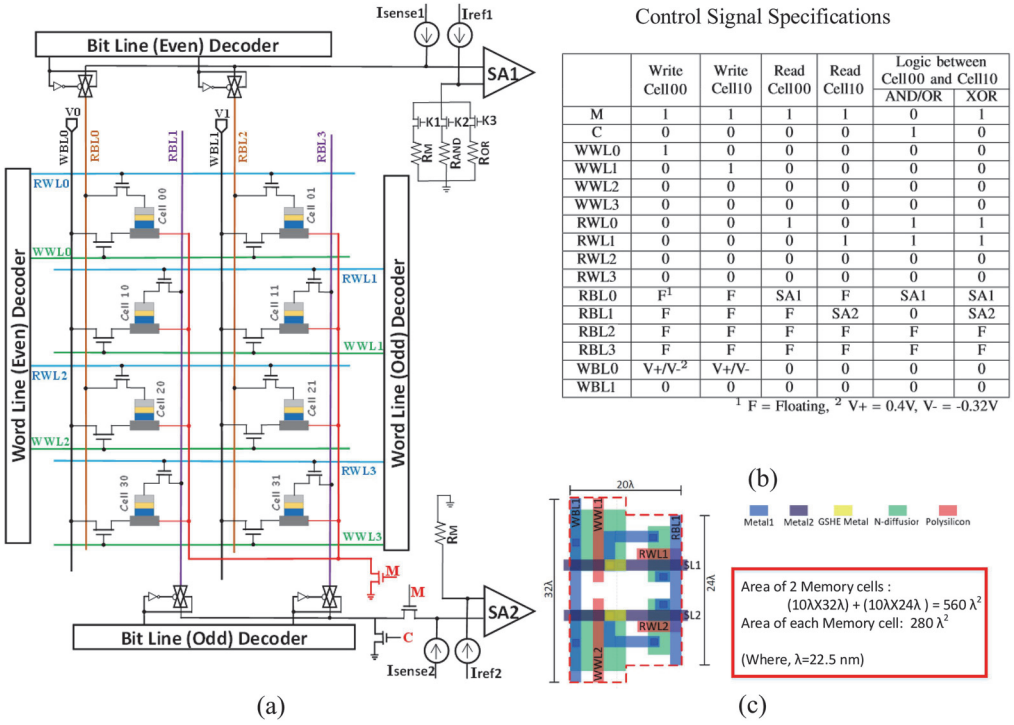
2.2 GSHE Driven SOT-MRAM

Figure 1(b) shows the Giant Spin Hall Effect (GSHE) driven MTJ-based Spin Orbit Torque Magnetic Random Access Memory (SOT-MRAM), with the composite structure of spin Hall metal (SHM) and MTJ. Here the flow of charge current ($\pm y$) through the SHM (Tungsten, W (Seo et al. 2016)) will cause accumulation of opposite directed spin on both surfaces of SHM due to spin Hall effect (Pai et al. 2012). Thus, a spin current in $\pm z$ is generated and further produces SOT on the adjacent free magnetic layer, causing a switch of magnetization (Prenat et al. 2015). For example, as shown in Figure 1(c), spin injection current produced in the $+z$ direction by flowing charge current through the SHM in the $+y$ ($-y$) direction will align the magnetization of the free FM layer in the $-x$ ($+x$) direction, which is anti-parallel (parallel) to that of the pinned FM layer. Hence, this will write logic "1" ("0") encoded as R_{AP} (R_P) in the MTJ. In this work, the magnetization dynamics of the free FM layer is modeled by an LLG equation with an STT term and SHE term (He and Fan 2016).

The simulation result of normalized magnetization dynamics of MTJ's free FM layer is shown in Figure 1(d), with 120μA charge current through the SHM. The device parameters used for modeling the structure are given in Table 1. Note that the ferromagnets in MTJ have In-plane Magnetic Anisotropy (IMA) (Hindmarch et al. 2011). With the given thickness of the tunneling layer, $R_P \approx$ 30.9kΩ and $R_{AP} \approx$ 85.9kΩ lead the tunneling magnetoresistance (TMR) of the MTJ to be $\sim$178%.

3 PROPOSED IN-MEMORY COMPUTING ARCHITECTURE

The proposed dual-mode SOT-MRAM architecture can perform both memory read-write and logic AND/OR/XOR operations. Figure 2(a) shows the architecture of the 4×2 memory array. Here, each memory cell is designed using the SOT-MRAM structure (Seo et al. 2016). Each cell is associated with the Read Word Lines (RWL), Write Word Lines (WWL), Read Bit Lines (RBL), and Write Bit Lines (WBL). The odd/even WLs are controlled by the Word Line (odd/even) Decoders. A significant modification (Parveen et al. 2017; Seo et al. 2016) in our proposed array architecture is that we have used two RBLs for each column of the array, where the even (odd) RBLs are connected to the even (odd) cells of a column. Two Sense Amplifiers (He and Fan 2016), SA1 and SA2, are connected to the even and odd RBLs, respectively, to inject sensing current through the activated current path. Note that all the other terminals (right terminals in Figure 2(a)) of the SHMs are connected together. Two additional control signals, M and C, are used to select the mode of operation. The control signal specifications for different operations are mentioned in Figure 2(b). Please



Control Signal Specifications

	Write Cell00	Write Cell10	Read Cell00	Read Cell10	Logic between Cell00 and Cell10	
					AND/OR	XOR
M	1	1	1	1	0	1
C	0	0	0	0	1	0
WWL0	1	0	0	0	0	0
WWL1	0	1	0	0	0	0
WWL2	0	0	0	0	0	0
WWL3	0	0	0	0	0	0
RWL0	0	0	1	0	1	1
RWL1	0	0	0	1	1	1
RWL2	0	0	0	0	0	0
RWL3	0	0	0	0	0	0
RBL0	F[1]	F	SA1	F	SA1	SA1
RBL1	F	F	F	SA2	0	SA2
RBL2	F	F	F	F	F	F
RBL3	F	F	F	F	F	F
WBL0	V+/V-[2]	V+/V-	0	0	0	0
WBL1	0	0	0	0	0	0

[1] F = Floating, [2] V+ = 0.4V, V- = -0.32V

Fig. 2. (a) Proposed SOT-MRAM array architecture. (b) Control Signal specifications for memory mode and computing mode operations. (c) Layout of two SOT-MRAM cells.

note, our design can perform the dual-mode functionality while implementing all the Boolean logic functions within the memory array without any additional transistor (Parveen et al. 2017). The layout of the two memory cells is shown in Figure 2(c).

3.1 Modes of Operation

Our proposed SOT-MRAM array can operate in two modes: memory mode and computing mode. The operations of the two modes are described in detail in the following sub-sections. Also the memory write, read, and compute operations for an entire word are explained in detail.

3.1.1 Memory Mode. In memory mode, the control signal M is set at High voltage (VDD) and C is set at Low voltage (GND). So the right terminals of all the SHMs are connected to the ground. Now data can be written or read from a memory cell by activating a write or read current path through that cell. Here the "memory write" and "memory read" operations are explained in detail.

Memory Write: To write a bit in any of the cells of the memory array shown in Figure 2(a), write current should be injected through the corresponding SHM. For example, to write into cell00, WWL0 is activated using the word line (even) decoder. All the other Word Lines are kept deactivated. Now to write Data "1" ("0"), the voltage driver, V0, is set to positive (negative) write voltage. All the other write voltage drivers are kept at zero voltage. This allows charge current to flow from V0 to ground (ground to V0) through the SHM. As described in Figure 1(c), this will set the MTJ resistance to be High, R_{AP} (Low, R_P) which is encoded as Data "1" ("0"). The write current path is shown in Figure 3(a).

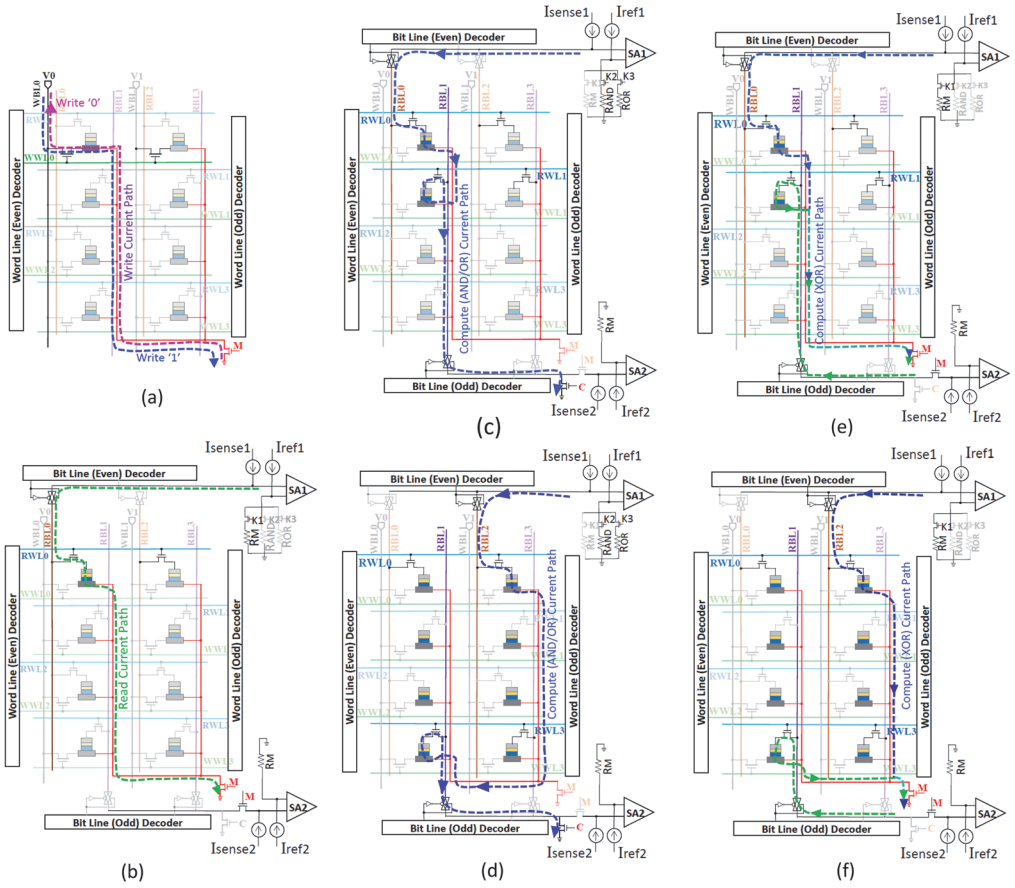


Fig. 3. (a) Write current path; (b) Read current path; (c) and (d) Compute (AND/OR) current path; (e) and (f) Compute (XOR) current path.

Memory Read: To read data stored in a memory cell, read current should be injected through that cell. For example, to read data stored in cell00, RWL0 is activated using the word line (even) decoder and RBL0 is connected to SA1 by activating the read bit line (even) decoder. All the other Word Lines and Bit Lines are kept deactivated. The reference MTJ, R_M of SA1 is selected by setting the key values K1K2K3 = 100. Hence, a read current is injected from SA1 through Cell00 to ground (since M = 1), generating a sense voltage at the input of SA1. SA1 then compares the Cell00 resistance with R_M whose resistance is set in between R_P and R_{AP}. So, if the resistance of Cell00 is higher (lower) than R_M, that is, R_{AP} (R_P), then the output of SA1 gives High (Low) voltage, which indicates that logic "1" ("0") is read from Cell00. The read current path is shown in Figure 3(b).

Note that a complete word can be read in one cycle if each bit line is connected to a separate SA (Bishnoi et al. 2014).

3.1.2 Computing Mode. Here the logic AND/OR and XOR operations are discussed below.

AND/OR: For logic AND/NAND and OR/NOR operations between the data stored in two cells, the series resistance of the two cells is sensed by flowing sense current through those cells in

Table 2. Implementation of Computing Mode (AND/OR)

Data stored in the two MTJs	0 0	0 1	1 0	1 1
Summation of MTJ resistances, R_{TOT}	$R_P + R_P$	$R_P + R_{AP}$	$R_{AP} + R_P$	$R_{AP} + R_{AP}$
Compare R_{TOT} to R^*_{AND}	$<R_{AND}$	$<R_{AND}$	$<R_{AND}$	$>R_{AND}$
SA1 output (AND)	0	0	0	1
Compare R_{TOT} to R^{**}_{OR}	$<R_{OR}$	$>R_{OR}$	$>R_{OR}$	$>R_{OR}$
SA1 output (OR)	0	1	1	1

*R_{AND} = Between $2R_{AP}$ and $R_P + R_{AP}$.
**R_{OR} = Between $2R_P$ and $R_P + R_{AP}$.

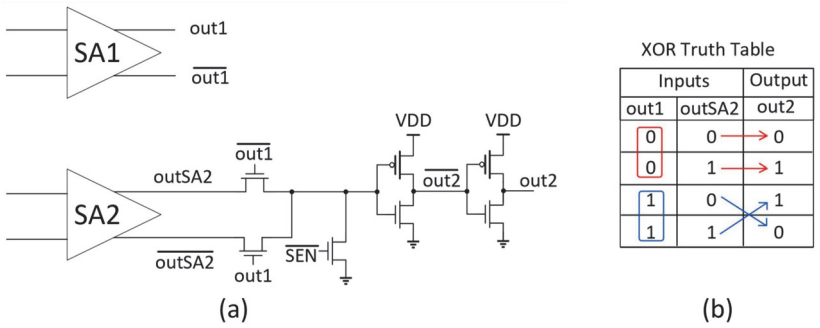


Fig. 4. Peripheral extension of SA2 for implementation of XOR/XNOR logic functions.

series. The control signal C is set at High voltage (VDD) and M is set at Low voltage (GND). So, the right terminals of all the SHMs are now floating, while still being connected together. Now, for example, in order to compute logic AND/OR between cell00 and cell10, RWL0 and RWL1 are activated using even and odd WL decoders, respectively. RBL0 and RBL1 are connected to SA1 and ground, respectively, using even and odd BL decoders. For computing AND (OR), the reference MTJ, R_{AND} (R_{OR}) is selected by setting K1K2K3 = 010 (001). Hence, a compute current (Figure 3(c)) is injected from SA1 through Cell00 and Cell10 to ground, generating a sense voltage at the input of SA1. SA1 then compares the summation of the cells' resistances with R_{AND} (R_{OR}) whose resistance is in between $R_P + R_{AP}$ and $2R_{AP}$ ($2R_P$), generating the output as per Table 2 . So, we get logic AND (OR) operation result from the SA1 output. Note that, as we used differential sense amplifier, the logic NAND (NOR) result can be obtained from the differential output of SA1.

Please note, logic AND/OR can be computed between any cell in any even row and any cell in any odd row in the same way. For example, the compute current path for the AND/OR operation between Cell01 (or any other cell in any other even row) and Cell30 (or any other cell in any other odd row) is shown in Figure 3(d).

XOR: For logic XOR operation, both SA1 and SA2 are used to read data simultaneously from the two corresponding operand cells. To activate both of these read current paths, M is set to 1 and C is set to 0. For example, the compute (XOR) current path for computing XOR between Cell00 and Cell10 is shown in Figure 3(e), where RWL0 and RWL1 are activated and RBL1 and RBL2 are connected to SA1 and SA2, respectively. So, two separate read currents will flow through Cell00 and Cell10 to ground, which are further sensed by SA1 and SA2, respectively. Now, the proposed modification to the SA2, that is, the extended peripheral circuitry at the output of SA2, will give the logic XOR computation result at the out2 terminal (Figure 4(a)) by copying (toggling) data

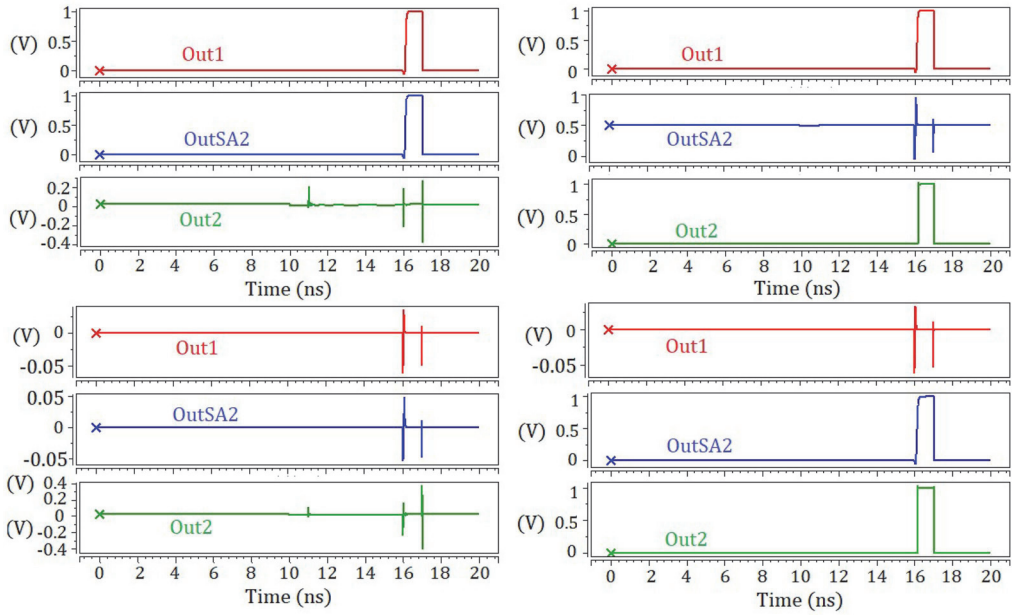


Fig. 5. Waveforms from the peripheral extension of SA2 for XOR operation with K1K2K3=100.

from SA2 output if SA1 output is 0 (1). The reason for this is clarified from the truth table shown in Figure 4(b), which demonstrates that the XOR result is equal to (or the inverse of) SA2 output, if SA1 output is 0 (or 1). Note that the logic XNOR result can be obtained from the $\overline{out2}$ terminal (Figure 4(a)) of the extended SA2 output. The waveforms from the peripheral extension of SA2 for XOR operation are shown in Figure 5. Here, Out2 terminal gives the result of the XOR operation between Out1 and OutSA2.

Please note, logic XOR can be computed between any cell in any even row and any cell in any odd row in the same way. For example, the compute current path for XOR operation between Cell01 (or any other cell in any other even row) and Cell30 (or any other cell in any other odd row) is shown in Figure 3(f).

4 DEVICE AND CIRCUIT LEVEL PERFORMANCE EVALUATION

In this section, we discuss the device and circuit level simulation framework for evaluating the performance of our proposed design. The device level simulation is conducted by numerically solving the LLGS equation (Equation (1)) with a consideration of the spin Hall effect (Jabeur et al. 2014) to simulate the dynamics of the SOT-MRAM cell based on the device parameters mentioned in Table 1.

$$(1+\alpha^2)\frac{\partial \vec{m}}{\partial t} = -|\gamma|(\vec{m}\times\vec{H_{eff}}) - \alpha|\gamma|\vec{m}\times(\vec{m}\times\vec{H_{eff}}) - \vec{m}\times\vec{m}\times\frac{I_s}{qN_{sx}} - \alpha\vec{m}\times\frac{I_s}{qN_{sx}}. \quad (1)$$

Here, α is the Gilbert damping factor, γ is the gyromagnetic ratio, and H_{eff} is the effective field including dipolar coupling field, demagnetization field, thermal noise field, and anisotropy

Table 3. Comparison of Memory Mode Operation of Single Cell

	Standard STT MRAM (Bi et al. 2013)	1R/1W SOT MRAM (Seo et al. 2016)	Proposed Design
Write access transistor width	500nm	180nm	240nm
Read access transistor width		180nm	90nm
Write Voltage (V)	1	0.3/−0.2	0.4/−0.32
Avg. Write Power (μW)	266.44	17.80	44.23
Avg. Read Power (μW)	30.71	10.86	12.8
Write Speed (ns)	>10	10	1
Compute (AND/NAND) Power (μW)	N/A	N/A	11.08
Compute (OR/NOR) Power (μW)	N/A	N/A	12.56
Compute (XOR/XNOR) Power (μW)	N/A	N/A	25.48

field. $N_s = M_s V/\mu_B$ is the number of spins, μ_B is Bohr magneton, and M_s and V are the saturation magnetization and volume of ferromagnet, respectively. I_s is the spin polarized current. According to SHE, the generated spin current can be written as

$$I_s = \theta_{SHM} \frac{A_s}{A_c} I_c \sigma. \tag{2}$$

Here, θ_{SHM} is the spin Hall angle characterizing the strength of SHE in SHM, σ is the polarization of the electron spin, and A_s and A_c are the cross-sectional areas through which spin and charge current flow, respectively.

The current required to be injected through the SHM for switching the MTJ resistance from $P \rightarrow AP$ or $AP \rightarrow P$ within 1ns is found to be $\sim 120\mu$A (Figure 1(d)). This could be obtained by properly sizing the write access transistors associated with WWL. Besides, $\sim 3\mu$A sense current is injected through the IMA (where magnetic domains lie along the wire axis (Fong et al. 2016)) MTJ during read, which is much lower than the write current. Hence, read disturbance is much reduced. Table 3 shows the memory mode performance compared to standard STT-MRAM (Bi et al. 2013) and contemporary 1R/1W SOT-MRAM (Seo et al. 2016) designs. Our proposed design shows significant improvement both in terms of read and write latency and power, compared to the standard STT MRAM. However, it requires 148.5% more average write power compared to 1R/1W SOT MRAM for 90% reduction in write speed, which is achieved by proper tuning of the write voltage and write access transistor width (Table 3).

For circuit level simulation, North Carolina State University (NCSU) Product Development Kit (PDK) with 45nm library (NCS 2011) is used in SPICE to verify the proposed design and evaluate the performance. A Verilog-A model of the GSHE-MTJ device is developed to co-simulate with the interface CMOS circuits in SPICE to validate the functionality and evaluate the performance of the proposed SOT-MRAM sub-array design.

Note that the logic AND/OR output is almost similar to the reading data from memory array through SA1. Hence, the equivalent average logic (AND/OR) power is almost similar to the average read power. Again, logic XOR operation is similar to reading out two memory cells simultaneously using both SA1 and SA2. Hence, the average logic (XOR) power is almost twice the average read power of a single cell.

Our design can perform both memory and computing operations without the need of any add-on circuits. From the layout (Figure 2(c)) of two SOT-MRAM cells used in our proposed array design, the area of each memory cell is found to be $280\lambda^2$, where $\lambda = 22.5$nm.

Table 4. SRAM, DRAM, STT-MRAM and IMFlexCom Memory Model Validation and Comparison for a Sample 4MB Memory

	SRAM		DRAM		STT-MRAM		IMFlexCom	
Metrics	Write	Read	Write	Read	Write	Read	Write	Read
Latency (ns)	1.53		2.7		11.13	2.62	1.97	1.99
Dynamic Energy (pJ)	297.46	312.56	967	1483	1219	791.13	285.35	269.62
Leakage Power (mW)	5258		185.5		830.3		730.97	
Area (mm^2)	10.544		4.702		4.611		7.326	

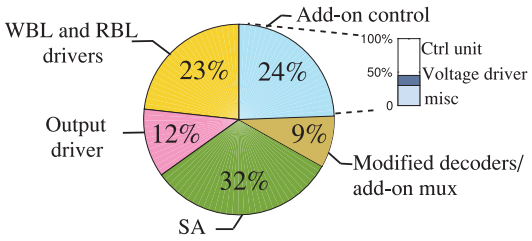


Fig. 6. Area overhead of IMFlexCom within one Bank.

5 SYSTEM LEVEL PERFORMANCE EVALUATION

In this section, the memory mode performance of the proposed SOT-MRAM array architecture is evaluated at system level. To do so, we have configured the memory chip organization by dividing it into multiple Banks (Bank organization: total-4×4, active-1×1) consisting of multiple Mats (Mat organization: total-2×2, active-1×1). Each Mat includes multiple sub-arrays (sub-array size: 1,024×512) organized in an H-tree routing manner. Table 4 tabulates and compares the performance of our design with different memory arrays employed for in-memory processing in SRAM, DRAM, and standard STT-MRAM (R_{ON} = 3kOhm, R_{OFF} = 6kOhm, cell area = $54F^2$, cell aspect ratio = 0.54) for a sample memory capacity of 4MB in 45nm process node.

For the simulation, the modified self-consistent NVSim (Dong et al. 2014) along with an in-house developed C++ code is employed to verify the system level performance and to report the accurate latency, energy, and area. Table 4 tabulates and compares the performance of our design with different memory arrays employed for in-memory processing in previous works (i.e., SRAM, DRAM, and standard STT-MRAM) for a sample memory capacity of 4MB with 256 bit Data width in 45nm process node.

According to Table 4, the proposed IMFlexCom memory model shows the least write dynamic energy due to its low write voltage (~400mV for "1" and ~−320mV for "0") in comparison to other designs. Besides, SOT-MRAM greatly reduces the total leakage power compared to SRAM. Although the proposed design improves the write latency up to 82% compared to standard STT-MRAM, both designs have shown longer write latency compared to SRAM due to the longer write latency of magnetic memory storage devices. Moreover, the area overhead of the proposed IMFlexCom memory model is ~37% more than DRAM and standard STT-MRAM design but still ~30% less than SRAM design. It is noteworthy that the first and foremost benefit of spintronic memories, compared to SRAM and DRAM, is their non-volatility with almost 10 years' retention time (Seo et al. 2016).

Figure 6 shows a breakdown of the area overhead resulted from add-on hardware to memory chip. Our experiments show that, in total, IMFlexCom based on SOT-MRAM imposes 2.8% area

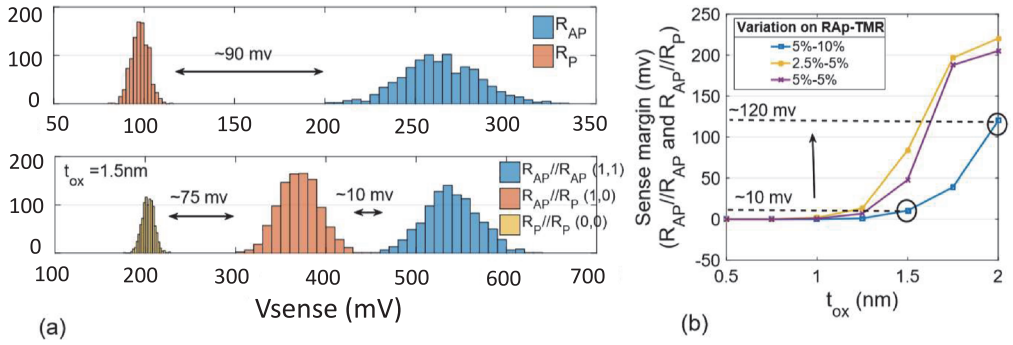


Fig. 7. (a) Sense margin analysis with 5% and 10% stochastic variations in MTJ resistance-area product and TMR, respectively, for (top) read operations, and (bottom) compute operations (sensing current $\sim 3\mu$A and t_{ox}=1.5nm). (b) Sense margin of AND-logic operation vs. thickness of MTJ oxide with different variations on RAp and TMR.

overhead to original memory die, where Pinatubo (Li et al. 2016) incurs 0.9% area overhead. It can be seen that the add-on controller and SAs contribute to more than 50% of this area overhead in a Bank.

6 SENSING RELIABILITY ANALYSIS

In our proposed IMFlexCom, the reliability of memory sensing and logic sensing are investigated by carrying out Monte-Carlo simulation with 100,000 iterations, while considering different stochastic variations on MTJ resistance-area product and TMR (5%-10%, 2.5%-5%, and 5%-5%, respectively) owing to PVT variation w.r.t. t_{ox}. We have adopted the MTJ process variation modeling scheme used by Samsung memory group in Ryu and Kwon (2017). Simulation results for both read operations and compute operations under 5%-10% variations (with t_{ox}=1.5nm) are shown in Figure 7(a). As shown, the sense margin is large ($\sim$75mV) between logic operands as (0,0) and (1,0)/(0,1), which means logic-OR has a large sense margin. However, our design has a relatively lower sense margin ($\sim$10mV) between operands as (1,0)/(0,1) and (1,1), which means logic-AND has a relatively smaller sense margin. In order to provide a larger sense margin for logic-AND operation, we propose to increase t_{ox}. As can be observed in Figure 7(b), increasing t_{ox} from 1.5nm to 2nm leads to $\sim$110mV increase in the sense margin which considerably enhances the reliability of AND-logic operation in IMFlexCom.

7 CASE STUDIES

The proposed SOT-MRAM array could be efficiently leveraged for in-memory data processing due to its inherent computing ability. In this section, we have investigated the performance of our proposed design for bulk bitwise Boolean vector operations and for implementing an in-memory data encryption engine.

7.1 In-Memory Bulk Bit-Wise Boolean Vector Operation

The inherent in-memory computing operations could be leveraged to perform bulk bit-wise Boolean vector logic operations within the data stored in the memory array. This can lead to efficient re-use of the internal memory bandwidth without any add-on logic circuit.

7.1.1 Mapping of Vector Data. To perform bit-wise logic (AND/OR/XOR) operation on two vectors, data from the two vectors are needed to be written into the memory array first. For example,

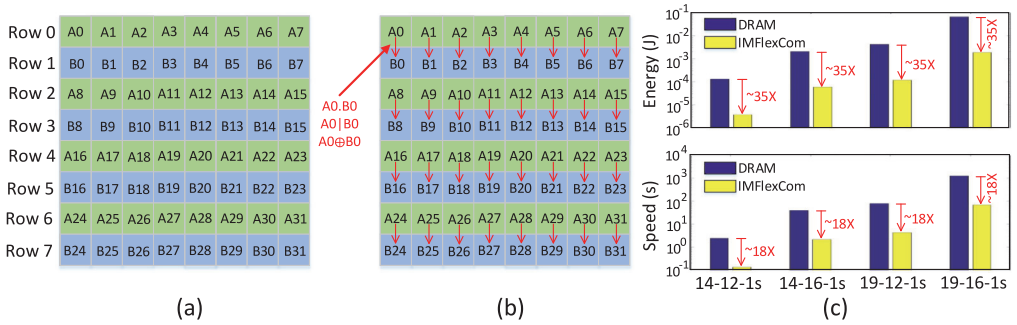


Fig. 8. Bulk bit-wise vector operation. (a) Data organization in an 8×8 array. (b) Computing cycles. (c) Energy saving and speedup for bit-wise AND/OR operation for different vector dataset compared to DRAM.

for computing the bit-wise AND (/OR/XOR) operation between two vectors A and B, each being 32 bits long, vector A could be written in the even rows of the memory array, whereas vector B could be written in the odd rows of the memory array as shown in Figure 8(a). Now, the vector operation could be done by computing the Boolean AND (/OR/XOR) operation between the two cells in two rows, or in other words, in two vectors as shown in Figure 8(b).

7.1.2 Performance Evaluation. The performance of the bulk bit-wise vector operation using our proposed architecture is evaluated using the similar simulation framework as described in Section 4. The performance for bulk bit-wise operations leveraging in-memory computing platform using DRAM (Seshadri et al. 2015) is also evaluated. We developed an in-house simulation platform incorporating $6F^2$ DRAM cell structure with 16fF cell capacitance (Yanagawa et al. 2011) using 45nm technology node. The complete compute operation is completed within three consecutive operation cycles—precharge, access, and sense (Seshadri et al. 2015; Wang 2005)—assuming data is already written in the memory array. The computation of AND/OR logic function is done in DRAM using a three input majority gate concept (Seshadri et al. 2015, 2017), with two data bits and one dummy bit as the three inputs. The energy and latency for the bit-wise AND/OR operations for different vector datasets (Li et al. 2016) are calculated. Please note, 19-16-1s refers to a vector dataset with vector length 2^{19}, number of vectors 2^{16}, and AND/OR operation is done between 2^1 rows, where "s" means sequentially. Figure 8(c) shows the energy saving and speedup of in-memory bulk bit-wise vector operation using our proposed SOT-MRAM array compared to that using DRAM array. Here the in-memory computing platform using our proposed SOT-MRAM array architecture offers ~35× energy saving and ~18× speedup compared to that using the DRAM-based in-memory computing platform (Seshadri et al. 2015). Again, the computation in DRAM is destructive, that is, data stored in three DRAM cells associated with the computation are overwritten with the result of the logic operation. Whereas, the stored data are retained even after computing in our SOT-MRAM–based design.

Please note, IMFlexCom performs bitwise logic computing operation serially in bit-by-bit fashion. So, only one bitwise logic (AND/OR/XOR) computation can be performed between two cells in a sub-array at a given time. In this context, if the data vector is stored in the same sub-array, then it will need multiple cycles (as many computing cycles as the number of bits in the vector) to complete the bitwise logic operation on the entire vector. In case any application requires the bitwise computing operation for the entire data vector to be completed with a faster speed, then the bits of such data vector should be stored in different sub-arrays to enable parallel processing.

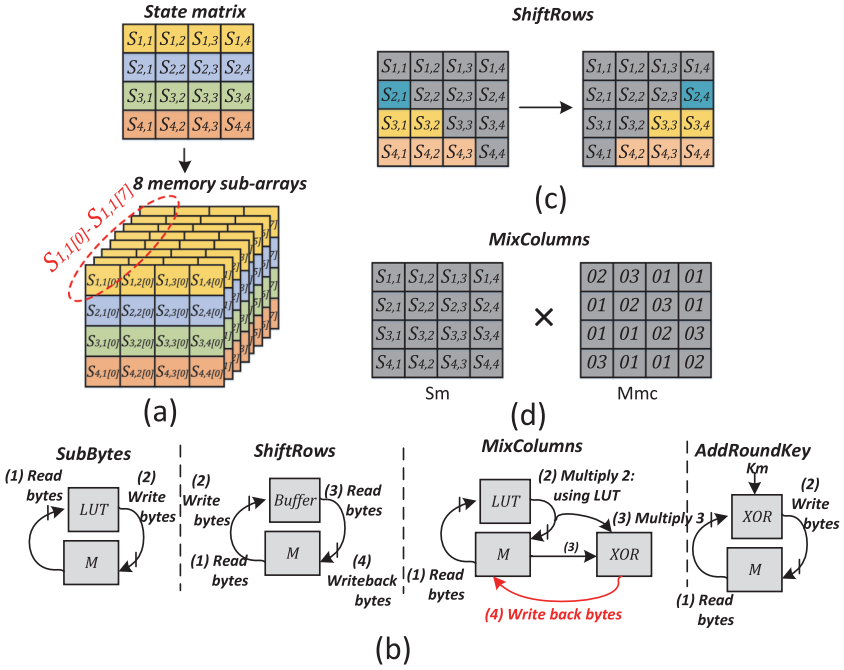


Fig. 9. (a) Data organization. (b) Data mapping of four AES transformations to the proposed in-memory processing platform. (c) Schematic representation of ShiftRows transformation. (d) Schematic representation of MixColumn transformation.

7.2 In-Memory Data Encryption: AES

The proposed SOT-MRAM array structure could be easily employed as an in-memory data encryption engine to implement various algorithms such as AES (Daemen and Rijmen 2002), PRESENT (Bogdanov et al. 2007), SIMON (Beaulieu et al. 2015), and KATAN (Canniere et al. 2009). In this section, we take the AES algorithm, which is widely used in data encryption, as an example to elucidate the mapping of its transformations into the proposed SOT-MRAM–based in-memory computing architecture. It reveals its benefits of energy-efficiency and high throughput for in-memory data encryption applications.

AES works on the standard input length of 16 bytes (128 bits) data organized in a 4×4 matrix (called the state matrix) while using three different key lengths (128, 192, and 256 bits) (Standard 2001). For 128-bit key length, AES encrypts the input data after 10 rounds of consecutive transformations (Wang et al. 2016), enumerated as SubBytes, ShiftRows, MixColumns, and AddRoundKey.

7.2.1 Mapping of AES Transformations. To facilitate working with input data in the AES algorithm (Angizi et al. 2017b; He et al. 2017) as shown in Figure 9(a), each byte in input data is initially distributed into 8-bit. Accordingly, eight memory sub-arrays are filled by 4×4 bit-matrices that can perform simultaneously. Overall, to increase the parallelism for the AES algorithm, we found that the number of employed sub-arrays needs to be increased. After the data is stored by memory sub-arrays, the data will undergo four different transformations in each AES round. In each round, the original data storage structure (eight memory sub-arrays filled by 4×4 bit-matrices) remains unchanged; however, data will be transformed after each stage.

Table 5. The Performance Comparison of Different Implementations for 128-bit AES

Platforms	Energy (nJ)	Cycles	Area (μm^2)
GPP (Malbrain 2009)	460	2309	2.5e+6
ASIC (Mathew et al. 2015)	6.6	336	4400
CMOL (Abid et al. 2009)	10.3	470	320
Baseline DW (Wang et al. 2016)	2.4	1022	78
Pipelined DW (Wang et al. 2016)	2.3	2652	83
Multi-issue DW (Wang et al. 2016)	2.7	1320	155
Proposed	1.5	1620	26

SubBytes: In this stage, each byte of state matrix undergoes a Look-up table (LUT) based transformation using S-box and updated by a nonlinear transformation $f(Si, j \leftarrow f(Si, j))$. The input of S-Box LUT (16×16 memory array) is essentially a Byte which is divided into two 4-bit data patterns. This 8-bit can be readily read out from eight sub-arrays and used as an address index for LUT. So, each pattern yields a row or a column index for the decoders reaching target cell in S-box. Then, the addressed data byte in S-Box is written back to the memory unit and substitutes the original data as shown in Figure 9(b). **ShiftRows:** In this stage, state matrix undergoes a cyclical shift operation by a certain offset (Figure 9(c)). To perform the shift operation, one sub-array can be considered as a buffer to temporarily save the readout data as in Figure 9(b). Thus, after reading the data from the second to fourth row (three rows), they can be easily rewritten to the memory with desired order. **MixColumns:** The state matrix is multiplied by a preset matrix (Mmc) depicted in Figure 9(d). The addition could be efficiently executed using multi-cycle in-memory XOR. To maximize the efficiency of AES performance, we use a LUT-based transformation followed by XOR operations to implement 2× operation, similar to the design in Wang et al. (2016). The 3× operation is defined as 2× result XOR with the original value as shown in Figure 9(d). Here the main challenging operation is XOR that can be readily performed using IMFlexCom as we explain in next stage. **AddRoundKey:** For each round, a key expansion unit produces a subkey derived from the main key using Rijndael's key schedule (Malbrain 2009). The 16-byte round keys are organized in a similar 4×4 array as the state matrix with Ki,j as the matrix entry. Here, each byte of the state matrix is replaced by bit-wise XOR of Si,j and Ki,j. This stage can be easily performed using the three-cycle in-memory XOR rather than nine cycles as in He et al. (2017) as follows: (1) the round key is written in the corresponding state matrix sub-arrays where eight memory sub-arrays are filled by 4×4 key bit-matrices; (2) in the computing mode, eight parallel XOR operations in eight sub-arrays will be performed to calculate one byte; and (3) the Sense Amplifier's outputs will be written back to the sub-array.

7.2.2 Performance Evaluation. The memory mode performance obtained in Section 4 in addition to in-memory computation (i.e., speed and energy) are used to evaluate the performance of in-memory AES, and compare with other software and hardware implementations (Table 5). For an evaluation of the AES performance in a general purpose processor (GPP), we use the similar method in Wang et al. (2016) at 2GHz. At first, AES C code is extracted from Malbrain (2009) and compiled, then cycle-accurate architecture simulator gem5 (Binkert et al. 2011) is employed to take AES binary and accordingly system level processor power evaluating tool McPAT (Li et al. 2009) is used to estimate the power dissipation. For evaluation of AES in CMOS ASIC (1.133GHz), Synopsys Design Compiler tool is used. Here, the performance for all the platforms is listed in Table 5 for 32nm technology. Since the performance evaluation of our design was carried out using 45nm process parameters, for fair comparison, we have done fixed-voltage scaling of the results obtained

from of our work to 32nm by using the appropriate scaling factor, which is $(1/S^2)$ for area and $(1/S)$ for energy (Stillmaker et al. 2011); here $S = L/32\text{nm}$, where $L = 45\text{nm}$. It can be seen that the proposed platform significantly improves the data encryption performance due to having less energy consumption and area in comparison to GPP-, ASIC-, and CMOL-AES implementations. The device to architecture co-simulation results shows that the proposed method can achieve 77.27% and 85.4% lower energy consumption compared to CMOS-ASIC and CMOL-based implementations, respectively. Furthermore, it shows ~66.7% area reduction compared to baseline DW-AES. This significant improvement mainly comes from our proposed in-memory processing energy efficient operations. Note that, Baseline DW-AES (Wang et al. 2016) requires a lower number of cycles due to intrinsic shift operation and multi-bit data storage of DWM racetrack devices. However, racetrack device suffers from data corruption due to non-uniform velocities of multiple DWs and huge Joule heating effects due to large device dimension.

8 CONCLUSION

In this article, we proposed an SOT-MRAM–based dual-mode in-memory flexible computing architecture, IMFlexCom. Extensive device, circuit, and system level simulation results reveal significant performance improvement over standard STT-MRAM and SRAM-based memory architectures. Our proposed design can be exploited for both memory and logic operations with reasonably comparative speed and power consumption with respect to other contemporary SOT-MRAM designs, with no increase in the number of transistors per cell. Furthermore, as case studies, we investigated bulk bitwise Boolean vector logic operation and implemented in-memory data encryption engine using the AES algorithm to elucidate its in-memory logic performance. Our design shows ~35× energy saving and ~18× speedup for in-memory bulk bitwise vector logic operations compared to the DRAM-based in-memory computing platform. Again, the device to architecture co-simulation results show that the AES implementation with the proposed design can achieve 77.27% and 85.4% lower energy consumption compared to CMOS-ASIC- and CMOL-based implementations, respectively. Whereas, it offers almost similar energy consumption as recent DW-AES implementation, but with 66.7% less area overhead. To summarize, by adding several significant features such as non-volatility, in-memory logic operation, zero leakage power, low dynamic power consumption, high packing density, our proposed design can thrive as a new paradigm for future power efficient in-memory computing platforms for big data applications.

REFERENCES

Z. Abid and others. 2009. Efficient CMOL gate designs for cryptography applications. *IEEE Transactions on Nanotechnology* 8 (2009), 315–321.

Junwhan Ahn, Sungjoo Yoo, Onur Mutlu, and Kiyoung Choi. 2015. PIM-enabled instructions: A low-overhead, locality-aware processing-in-memory architecture. In *Proceedings of the 2015 ACM/IEEE 42nd Annual International Symposium on Computer Architecture (ISCA'15)*. IEEE, 336–348.

Shaahin Angizi, Zhezhi He, and Deliang Fan. 2017a. Energy efficient in-memory computing platform based on 4-terminal spin Hall effect-driven domain wall motion devices. In *Proceedings of the on Great Lakes Symposium on VLSI 2017*. ACM, 77–82.

Shaahin Angizi, Zhezhi He, Nader Bagherzadeh, and Deliang Fan. 2017b. Design and evaluation of a spintronic in-memory processing platform for non-volatile data encryption. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2017), 428–430.

G. Autès, J. Mathon, and A. Umerski. 2010. Strong enhancement of the tunneling magnetoresistance by electron filtering in an Fe/MgO/Fe/GaAs (001) junction. *Physical Review Letters* 104, 21 (2010), 217202.

Ray Beaulieu, Stefan Treatman-Clark, Douglas Shors, Bryan Weeks, Jason Smith, and Louis Wingers. 2015. The SIMON and SPECK lightweight block ciphers. In *Proceedings of the 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC'15)*. IEEE, 1–6.

Xiuyuan Bi, Mohamed Anis Weldon, and Hai Li. 2013. STT-RAM designs supporting dual-port accesses. In *Proceedings of the Conference on Design, Automation and Test in Europe*. EDA Consortium, 853–858.

Nathan Binkert and others. 2011. The gem5 simulator. *SIGARCH* 39 (2011), 1–7.

Rajendra Bishnoi, Mojtaba Ebrahimi, Fabian Oboril, and Mehdi B. Tahoori. 2014. Architectural aspects in design and analysis of SOT-based memories. In *Design Automation Conference (ASP-DAC), Proceedings of the 2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC'14)*. IEEE, 700–707.

A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe. 2007. PRESENT: An ultra-lightweight block cipher. In *Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems (CHES'07)*. Springer-Verlag, Berlin, 450–466. DOI: http://dx.doi.org/10.1007/978-3-540-74735-2_31

De Canniere and others. 2009. Katan and Ktantan—A family of small and efficient hardware-oriented block ciphers. In *Proceedings of the International Workshop on CHES*. Springer, 272–288.

Xunchao Chen, Navid Khoshavi, Ronald F. DeMara, Jun Wang, Dan Huang, Wujie Wen, and Yiran Chen. 2017. Energy-aware adaptive restore schemes for MLC STT-RAM cache. *IEEE Transactions on Computers* 66, 5 (2017), 786–798.

Ping Chi, Shuangchen Li, Z. Qi, P. Gu, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y. Xie. 2016. PRIME: A novel processing-in-memory architecture for neural network computation in ReRAM-based main memory. In *Proceedings of ISCA*, Vol. 43.

Joan Daemen and Vincent Rijmen. 2002. *The Design of Rijndael: AES—The Advanced Encryption Standard*. Springer-Verlag. 238 pages.

J. Darper and others. 2002. The architecture of DIVA processing in memory chips. In *Proceedings of the 16th International Conference on Supercomputing (ICS'02)*.

Brandon Del Bel, Jongyeon Kim, Chris H. Kim, and Sachin S. Sapatnekar. 2014. Improving STT-MRAM density through multibit error correction. In *Proceedings of the Conference on Design, Automation & Test in Europe*. European Design and Automation Association, 182.

Xiangyu Dong, Cong Xu, Norm Jouppi, and Yuan Xie. 2014. NVSim: A circuit-level performance, energy, and area model for emerging non-volatile memory. In *Emerging Memory Technologies*. Springer, 15–50.

Jeff Draper, Jacqueline Chame, Mary Hall, Craig Steele, Tim Barrett, Jeff LaCoss, John Granacki, Jaewook Shin, Chun Chen, Chang Woo Kang, and others. 2002. The architecture of the DIVA processing-in-memory chip. In *Proceedings of the 16th International Conference on Supercomputing*. ACM, 14–25.

Deliang Fan. 2016. Low power in-memory computing platform with four terminal magnetic domain wall motion devices. In *Proceedings of the 2016 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH'16)*. IEEE, 153–158.

Amin Farmahini-Farahani, Jung Ho Ahn, Katherine Morrow, and Nam Sung Kim. 2015. NDA: Near-DRAM acceleration architecture leveraging commodity DRAM devices and standard memory modules. In *Proceedings of the 2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA'15)*. IEEE, 283–295.

Xuanyao Fong, Sumeet K. Gupta, Niladri N. Mojumder, Sri Harsha Choday, Charles Augustine, and Kaushik Roy. 2011. KNACK: A hybrid spin-charge mixed-mode simulator for evaluating different genres of spin-transfer torque MRAM bit-cells. In *Proceedings of the 2011 International Conference on Simulation of Semiconductor Processes and Devices*. IEEE, 51–54.

Xuanyao Fong, Yusung Kim, Sri Harsha Choday, and Kaushik Roy. 2014. Failure mitigation techniques for 1T-1MTJ spin-transfer torque MRAM bit-cells. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 22, 2 (2014), 384–395.

Xuanyao Fong, Yusung Kim, Karthik Yogendra, Deliang Fan, Abhronil Sengupta, Anand Raghunathan, and Kaushik Roy. 2016. Spin-transfer torque devices for logic and memory: Prospects and perspectives. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35, 1 (2016), 1–22.

Qing Guo, Xiaochen Guo, Ravi Patel, Engin Ipek, and Eby G. Friedman. 2013. AC-DIMM: Associative computing with STT-MRAM. In *ACM SIGARCH Computer Architecture News*, Vol. 41. ACM, 189–200.

Xiaochen Guo, Engin Ipek, and Tolga Soyata. 2010. Resistive computation: Avoiding the power wall with low-leakage, STT-MRAM based computing. In *ACM SIGARCH Computer Architecture News*, Vol. 38. ACM, 371–382.

Zhezhi He, Shaahin Angizi, Farhana Parveen, and Deliang Fan. 2017. Leveraging dual-mode magnetic crossbar for ultra-low energy in-memory data encryption. In *Proceedings of the on Great Lakes Symposium on VLSI 2017*. ACM, 83–88.

Zhezhi He and Deliang Fan. 2016. A low power current-mode flash ADC with spin Hall effect based multi-threshold comparator. In *Proceedings of the 2016 International Symposium on Low Power Electronics and Design*. ACM, 314–319.

A. T. Hindmarch, A. W. Rushforth, R. P. Campion, C. H. Marrows, and B. L. Gallagher. 2011. Origin of in-plane uniaxial magnetic anisotropy in CoFeB amorphous ferromagnetic thin films. *Physical Review B* 83, 21 (2011), 212404.

H. Honjo, H. Sato, S. Ikeda, S. Sato, T. Watanebe, S. Miura, T. Nasuno, Y. Noguchi, M. Yasuhira, T. Tanigawa, and others. 2015. 10 nmf perpendicular-anisotropy CoFeB-MgO magnetic tunnel junction with over 400°C high thermal tolerance by boron diffusion control. In *Proceedings of the 2015 Symposium on VLSI Technology (VLSI Technology'15)*. IEEE, T160–T161.

Kotb Jabeur, Gregory Di Pendina, Guillaume Prenat, Liliana Daniela Buda-Prejbeanu, and Bernard Dieny. 2014. Compact modeling of a magnetic tunnel junction based on spin orbit torque. *IEEE Transactions on Magnetics* 50, 7 (2014), 1–8.

Shubham Jain, Ashish Ranjan, Kaushik Roy, and Anand Raghunathan. 2017. Computing in memory with spin-transfer torque magnetic RAM. *arXiv:1703.02118*.

Wang Kang, Haotian Wang, Zhaohao Wang, Youguang Zhang, and Weisheng Zhao. 2017. In-memory processing paradigm for bitwise logic operations in STT-MRAM. *IEEE Transactions on Magnetics* 53, 11 (2017).

Wang Kang, Zhaohao Wang, Youguang Zhang, Jacques-Olivier Klein, Weifeng Lv, and Weisheng Zhao. 2016. Spintronic logic design methodology based on spin Hall effect–driven magnetic tunnel junctions. *Journal of Physics D: Applied Physics* 49, 6 (2016), 065008.

Wang Kang, Liuyang Zhang, Weisheng Zhao, Jacques-Olivier Klein, Youguang Zhang, Dafiné Ravelosona, and Claude Chappert. 2015. Yield and reliability improvement techniques for emerging nonvolatile STT-MRAM. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 5, 1 (2015), 28–39.

Yusung Kim, Sumeet Kumar Gupta, Sang Phill Park, Georgios Panagopoulos, and Kaushik Roy. 2012. Write-optimized reliable design of STT MRAM. In *Proceedings of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design*. ACM, 3–8.

Peter M. Kogge. 1994. EXECUBE—A new architecture for scaleable MPPs. In *Proceedings of the International Conference on Parallel Processing (ICPP'94)*, Vol. 1. IEEE, 77–84.

Kon-Woo Kwon, Xuanyao Fong, Parami Wijesinghe, Priyadarshini Panda, and Kaushik Roy. 2015. High-density and robust STT-MRAM array through device/circuit/architecture interactions. *IEEE Transactions on Nanotechnology* 14, 6 (2015), 1024–1034.

Sheng Li and others. 2009. McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In *2009 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'09)*. ACM, 469–480.

Shuangchen Li, Cong Xu, Qiaosha Zou, Jishen Zhao, Yu Lu, and Yuan Xie. 2016. Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories. In *Proceedings of the 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC'16)*. IEEE, 1–6.

K. Malbrain. 2009. Byte-oriented-aes: A public domain byte-oriented implementation of AES in C. (2009).

Sanu Mathew and others. 2015. 340 mV–1.1 V, 289 Gbps/W, 2090-gate nanoAES hardware accelerator with area-optimized encrypt/decrypt GF (2 4) 2 polynomials in 22 nm tri-gate CMOS. *IEEE Journal of Solid-State Circuits* 50, 4 (2015), 1048–1058.

Ravi Nair, Samuel F. Antao, Carlo Bertolli, Pradip Bose, Jose R. Brunheroto, Tong Chen, C.-Y. Cher, Carlos H. A. Costa, Jun Doi, Constantinos Evangelinos, and others. 2015. Active memory cube: A processing-in-memory architecture for exascale systems. *IBM Journal of Research and Development* 59, 2/3 (2015), 17–1.

NCSU EDA FreePDK45. (2011). Retrieved May 2011 from http://www.eda.ncsu.edu/wiki/FreePDK45:Contents.

I. Off. 1992. Computational RAM: A memory-SIMD hybrid and its application to DSP.

Mark Oskin, Frederic T. Chong, and Timothy Sherwood. 1998. *Active Pages: A Computation Model for Intelligent Memory*. Vol. 26. IEEE Computer Society.

Chi-Feng Pai, Luqiao Liu, Y. Li, H. W. Tseng, D. C. Ralph, and R. A. Buhrman. 2012. Spin transfer torque devices utilizing the giant spin Hall effect of tungsten. *Applied Physics Letters* 101, 12 (2012), 122404.

Farhana Parveen, Shaahin Angizi, Zhezhi He, and Deliang Fan. 2017. Low power in-memory computing based on dual-mode SOT-MRAM. In *Proceedings of the 2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED'17)*. IEEE, 1–6.

David Patterson, Thomas Anderson, Neal Cardwell, Richard Fromm, Kimberley Keeton, Christoforos Kozyrakis, Randi Thomas, and Katherine Yelick. 1997a. Intelligent RAM (IRAM): Chips that remember and compute. In *Proceedings of the 1997 43rd IEEE International Solid-State Circuits Conference (ISSCC'97), Digest of Technical Papers*. IEEE, 224–225.

David Patterson, Thomas Anderson, Neal Cardwell, Richard Fromm, Kimberly Keeton, Christoforos Kozyrakis, Randi Thomas, and Katherine Yelick. 1997b. A case for intelligent RAM. *IEEE Micro* 17, 2 (1997), 34–44.

J. Thomas Pawlowski. 2011. Hybrid memory cube (HMC). In *Proceedings of the 2011 IEEE Hot Chips 23 Symposium (HCS'11)*. IEEE, 1–24.

Guillaume Prenat, Kotb Jabeur, Gregory Di Pendina, Olivier Boulle, and Gilles Gaudin. 2015. Beyond STT-MRAM, spin orbit torque RAM SOT-MRAM for high speed and high reliability applications. In *Spintronics-Based Computing*. Springer, 145–157.

Guillaume Prenat, Kotb Jabeur, Pierre Vanhauwaert, Gregory Di Pendina, Fabian Oboril, Rajendra Bishnoi, Mojtaba Ebrahimi, Nathalie Lamard, Olivier Boulle, Kevin Garello, and others. 2016. Ultra-fast and high-reliability SOT-MRAM: From cache replacement to normally-off computing. *IEEE Transactions on Multi-Scale Computing Systems* 2, 1 (2016), 49–60.

Erik Riedel, Christos Faloutsos, Garth A. Gibson, and David Nagle. 2001. Active disks for large-scale data processing. *Computer* 34, 6 (2001), 68–74.

J.-W. Ryu and K.-W. Kwon. 2017. Self-adjusting sensing circuit without speed penalty for reliable STT-MRAM. *Electronics Letters* 53, 4 (2017), 224–226.

Yeongkyo Seo, Kon-Woo Kwon, Xuanyao Fong, and Kaushik Roy. 2016. High performance and energy-efficient on-chip cache using dual port (1R/1W) spin-orbit torque MRAM. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 6, 3 (2016), 293–304.

Vivek Seshadri, Kevin Hsieh, Amirali Boroum, Donghyuk Lee, Michael A. Kozuch, Onur Mutlu, Phillip B. Gibbons, and Todd C. Mowry. 2015. Fast bulk bitwise AND and OR in DRAM. *IEEE Computer Architecture Letters* 14, 2 (2015), 127–131.

Vivek Seshadri, Donghyuk Lee, Thomas Mullins, Hasan Hassan, Amirali Boroumand, Jeremie Kim, Michael A. Kozuch, Onur Mutlu, Phillip B. Gibbons, and Todd C. Mowry. 2017. Ambit: In-memory accelerator for bulk bitwise operations using commodity DRAM technology. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*. ACM, 273–287.

NIST-FIPS Standard. 2001. Announcing the advanced encryption standard (AES). *FIPSP* 197 (2001).

Aaron Stillmaker, Zhibin Xiao, and Bevan Baas. 2011. *Toward More Accurate Scaling Eestimates of CMOS Circuits From 180 nm to 22 nm*. VLSI Computation Lab Technical Report ECE-VCL-2011-4. ECE Department, University of California, Davis.

David Tawei Wang. 2005. *Modern Dram Memory Systems: Performance Analysis and Scheduling Algorithm*. Ph.D. Dissertation. Unversity of Maryland.

Jian-Ping Wang and Jonathan D. Harms. 2015. General structure for computational random access memory (CRAM). (Dec. 29 2015). US Patent No. 9,224,447, Filed April 23, 2014, Issued Dec. 29 2015.

Yuhao Wang, Leibin Ni, Chip-Hong Chang, and Hao Yu. 2016. DW-AES: A domain-wall nanowire-based AES for high throughput and energy-efficient data encryption in non-volatile memory. *IEEE Transactions on Information Forensics and Security* 11, 11 (2016), 2426–2440.

Yuhao Wang, Hao Yu, Leibin Ni, Guang-Bin Huang, Mei Yan, Chuliang Weng, Wei Yang, and Junfeng Zhao. 2015. An energy-efficient nonvolatile in-memory computing architecture for extreme learning machine by domain-wall nanowire devices. *IEEE Transactions on Nanotechnology* 14, 6 (2015), 998–1012.

Wm. A. Wulf and Sally A. McKee. 1995. Hitting the memory wall: Implications of the obvious. *ACM SIGARCH Computer Architecture News* 23, 1 (1995), 20–24.

Y. Yanagawa and others. 2011. In-substrate-bitline sense amplifier with array-noise-gating scheme for low-noise 4F 2 DRAM array operable at 10-fF cell capacitance. In *Proceedings of VLSIC*. IEEE, 230–231.

Dongping Zhang, Nuwan Jayasena, Alexander Lyashevsky, Joseph L. Greathouse, Lifan Xu, and Michael Ignatowski. 2014. TOP-PIM: Throughput-oriented programmable processing in memory. In *Proceedings of the 23rd International Symposium on High-Performance Parallel and Distributed Computing*. ACM, 85–98.

He Zhang, Wang Kang, Lezhi Wang, Kang L. Wang, and Weisheng Zhao. 2017. Stateful reconfigurable logic via a single-voltage-gated spin Hall-effect driven magnetic tunnel junction in a spintronic memory. *IEEE Transactions on Electron Devices* 64, 10 (2017), 4295–4301.

Yaojun Zhang, Bonan Yan, Wenqing Wu, Hai Li, and Yiran Chen. 2015. Giant spin Hall effect (GSHE) logic design for low power application. In *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*. EDA Consortium, 1000–1005.

Qiuling Zhu, Kaushik Vaidyanathan, Ofer Shacham, Mark Horowitz, Larry Pileggi, and Franz Franchetti. 2012. Design automation framework for application-specific logic-in-memory blocks. In *Proceedings of the 2012 IEEE 23rd International Conference on Application-Specific Systems, Architectures and Processors (ASAP'12)*. IEEE, 125–132.

Received July 2017; revised May 2018; accepted May 2018