A Robust Dual Reference Computing-in-Memory Implementation and Design Space Exploration Within STT-MRAM

Liuyang Zhang, Wang Kang, Hao Cai, Peng Ouyang, Lionel Torres, Youguang Zhang, Aida Todri-Sanial, Weisheng Zhao

A Robust Dual Reference Computing-in-Memory Implementation and Design Space Exploration Within STT-MRAM

Liuyang Zhang*, Wang Kang*, Hao Cai†, Peng Ouyang*, Lionel Torres‡, Youguang Zhang*
Aida Todri-Sanial‡ and Weisheng Zhao*
*Fert Beijing Institute and School of Electronic and Information Engineering
Beihang University, Beijing 100191, China
†Telecom Paristech, University of Paris-Saclay, Paris 75013, France
‡LIRMM/University of Montpellier, Montpellier 34095, France
Email: {wang.kang, weisheng.zhao}@buaa.edu.cn

***Abstract*—Due to the "memory wall" in conventional Von-Neumann computer architectures, the limited bandwidth between processors and memories has become one of the most critical bottlenecks to improve system performance. With the emerging of non-volatile memories, the computing-in-memory (CIM) paradigm has regained interest to tackle the issue at the architecture level. CIM can effectively alleviate the stress on the limitted bandwidth by performing logic operations within memories. However, CIMs are not yet studied carefully at the circuit level, and even its reliability and performance. In this paper, we proposed a CIM implementation: dual reference (DualRef) scheme at the circuit level within STT-MRAM (Spin Transfer Torque Magnetic Random Access Memory) array. Simulations were carried out to verify the functionality and assess the reliability and performance of DualRef scheme in terms of operation error rate, sensing margin, operation delay and dynamic energy consumption. Simulation results validate DualRef scheme and reveal that it is reliable to perform bitwise logic opertions within STT-MRAM while the TMR (Tunnel Magnetoresistance Ratio) varying between 100% and 300% and supply voltage V_{dd} varying from 0.9V to 1.2V. This work provides a robust circuitry scheme and design space to effectively implement CIM in STT-MRAM.**

I. Introduction

With the rapid growth of big data and Internet of Things (IoT) applications, a huge amount of data is generated and exchanged in the network consisting of billions of devices [1]. However, the processor frequency and the memory access efficiency are in state of imbalance in the conventional Von-Neumann computer architectures. Data needs to be transferred back and forth between processors and memories in the architectures. The limitted bandwidth between processors and memories makes energy consumption, data transferring and processing highly inefficient, especially within data-intensive applications [2], [3]. About 200 times more energy is comsumed to access DRAM one time compared with a floating-point computation [4]. It makes the limited bandwidth a critical bottleneck to improve the system performance, which is the well-known processor-memory gap or "memory wall" [5].

To bridge the processor-memory gap, many efforts have been spent on integrating processing unit and storage unit into one chip. These efforts can be classified into three types: 1) adding logic modules in memories, 2) embedding limited memory array into processing unit, and 3) moving some specified computations to memories. These methods aim to overcome the data transfer stress on the limited bandwidth between processors and memories when running data-intensive applications [6]. However, these efforts failed to tackle the issue due to some practical considerations. In CMOS based memories, it is complex and cost inefficient to integrate processing unit and storage unit together until 3D integration technology emerges [7]. Moreover, these technologies also suffer from the reliability issues [8], [9].

With the emerging of non-volatile memories, it becomes possible to address this issue [10], [11], for example, STT-MRAM (Spin Transfer Torque Magnetic RAM), RRAM (Resistive RAM), SOT-MRAM (Spin Orbit Torque Magnetic RAM) and other spintronics based devices, logics or memories, etc. [12], [13], [14]. STT-MRAM has been considered as one of the most promising candidates due to its distinctive advantages over other non-volatile memories, such as, non-volatility, good scalability, compatibility with CMOS, ultra fast accessing speed, etc. [15], [16], [17], [18]. The concept of computing-in-memory (CIM) is proposed several years ago, and regained interest recently, which is based on the idea of adding necessary peripheral circuit to memories [19]. CIM makes memories have some kind of computation capability and storage capability at the same time [20], [21]. It is just needed to modify the peripheral circuitry to implement CIM within STT-MRAM, which lets it process and store data simultaneously [21]. Some CIM paradigms at architecture level have been presented and assessed [6], [22], [23], [24]. There are two types of CIMs: one utilizes two or more reference cells to implement CIM, while the other one performs CIM operations depending on its complementary structure[5], [25], [26]. These paradigms introduced in above give us two ways to implement CIM within STT-MRAM. However, these efforts have been done at architecture level to implement CIM, there are few detailed implementations in circuit and few studies

2159-3477/18/$31.00 ©2018 IEEE
DOI 10.1109/ISVLSI.2018.00058

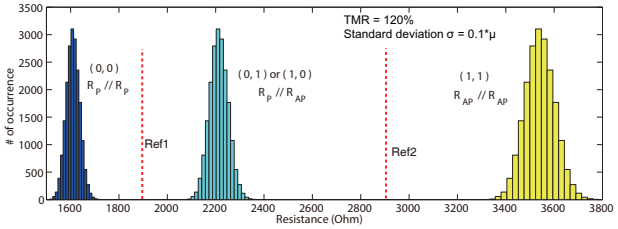

Fig. 1. Resistance distribution of two MTJs aligned in low resistance parallel state (R_P//R_P), antiparallel state (R_P//R_{AP}) and high resistance parallel state (R_{AP}//R_{AP}). The MTJ model used comes from [31].

TABLE I
TRUTH TABLE FOR DUALREF CIM IMPLEMENTATION

Logic	Bit pattern	Ouput
OR	**(00)**(R_P, R_P)	**0**((R_P+R_T)//(R_P+R_T)<R_{LRef})
	(01)(R_P, R_{AP})	**1**((R_P+R_T)//$(R_{AP}+R_T)$>R_{LRef})
	(10)(R_{AP}, R_P)	**1**($(R_{AP}+R_T)$//(R_P+R_T)>R_{LRef})
	(11)(R_{AP}, R_{AP})	**1**($(R_{AP}+R_T)$//$(R_{AP}+R_T)$>R_{LRef})
AND	**(00)**(R_P, R_P)	**0**((R_P+R_T)//(R_P+R_T)<R_{HRef})
	(01)(R_P, R_{AP})	**0**((R_P+R_T)//$(R_{AP}+R_T)$<R_{HRef})
	(10)(R_{AP}, R_P)	**0**($(R_{AP}+R_T)$//(R_P+R_T)<R_{HRef})
	(11)(R_{AP}, R_{AP})	**1**($(R_{AP}+R_T)$//$(R_{AP}+R_T)$>R_{HRef})

carefully on its reliability and performance [27], [28], [29], [30]. In this work, we proposed a CIM implementation at circuit level. Our contribution can be expressed as follows.

- Proposed a dual reference CIM implementation DualRef within STT-MRAM.
- Optimized the parameters of MTJ and CMOS transistors to meet the design requirements, and validated the DualRef CIM implementation.
- Carried out simulations to assess the reliability and performance of the DualRef CIM implementation by calculating the operation error rate, sensing margin, operation delay and dynamic energy consumption.

The rest of this paper is organized as follows. Section II introduces the scheme to implement DualRef in detailed, and then the functionality of the CIM implementation is validated. After that, the reliability and performance assessment for DualRef are included in Section III. At last, Section IV concludes this paper.

II. PROPOSED DUALREF CIM IMPLEMENTATION

In this section, the proposed DualRef CIM implementation is introduced, validated and assessed, which is realized by adding necessary peripheral circuit. DualRef can perform bitwise logic operation, which also can be used to store data at the same time. In the follows, we will show a four-by-four DualRef CIM array and a single DualRef CIM cell circuit, and then present how to excute basic bitwise logic operation in the CIM array.

A. Design of DualRef Scheme

The AND and OR bitwise logic operations can be executed in the DualRef CIM implementation, other bitwise

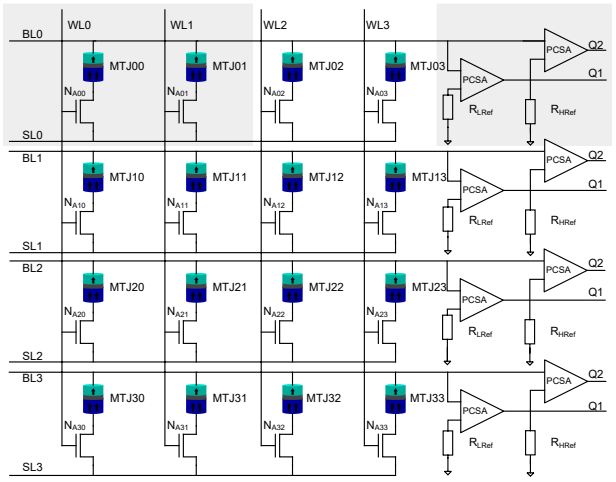

Fig. 2. A four-by-four DualRef CIM array. R_{LRef} and R_{HRef} are designed to distinguish the three resistance states as shown in Fig. 1.

logic operations, for example, XOR, XNOR, can also be implemented by adding essential assistant logic circuit after the sensing amplifiers. As the key storage element of STT-MRAM, MTJ device has two resistance states: low resistance state R_P and high resistance state R_{AP}. The two resistance states can be switched to each other by applying bi-directional currents to BL (Bit Line) and SL (Source Line). R_P usually encodes logic "0", and R_{AP} represents logic "1". There are three resistance states of two MTJs connected in parallel. To distinguish these three resistance states, two different reference resistors Ref1 and Ref2 are required, as shown in Fig. 1. However, the normal STT-MRAM cell has the 1T1MTJ (a MTJ and a transistor connected in series) structure. In order to be consistant with the two bit cells in parallel, the two reference cells are designed with the same structure as the bit cells. Assumed that the resistance of the transistor connected to the MTJ in the bit cells is R_T. Therefore, the resistance of the low reference cell R_{LRef} should be between $(R_P + R_T)//(R_P + R_T)$ and $(R_P + R_T//(R_{AP} + R_T)$, while the resistance of the high reference cell R_{HRef} locates in the middle of $(R_P+R_T)//(R_{AP}+R_T)$ and $(R_{AP}+R_T)//(R_{AP}+R_T)$. OR bitwise logic operations are performed by comparing the resistance of the two paralleled bit cells with that of specified reference cell, the results is "0" or "1" when less or more than R_{LRef}; while AND bitwise logic operation results can be obtained after checking the resistance of the two paralleled bit cells with R_{HRef}, which will be "0" or "1" when less or more than R_{LRef}. The truth table reveals the details from Table I.

With these knowledge, the DualRef CIM scheme is presented in Fig. 2. The figure shows a case of four-by-four array, which is adapted from a normal STT-MRAM array by adding a PCSA (PreCharge Sensing Amplifier), a reference cell for every row. With different configuartions, it can be a DualRef array for CIM or a normal STT-MRAM array, and also can be switched to each other freely. In this scheme, the two PCSAs work together to distinguish one resistance state

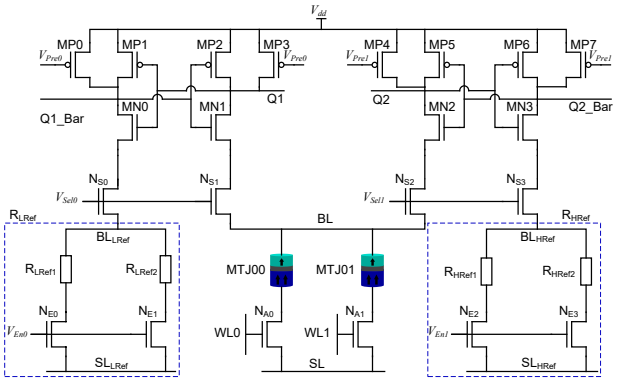

Fig. 3. Schematic of the DualRef CIM cell. It is the part with the gray background shown in Fig. 2.

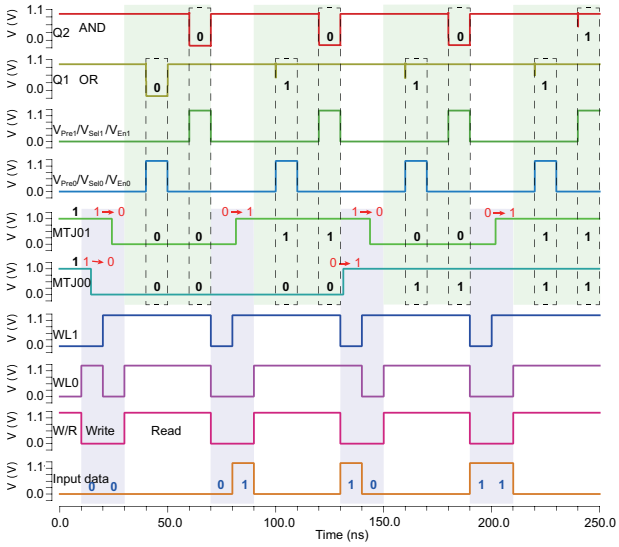

Fig. 4. Transient simulation waveform of the DualRef CIM scheme.

from others [32]. The final OR bitwise logic operation result can be obtained from the Q1, and AND bitwise logic operation result from the Q2 in this scheme.

The DualRef CIM cell is shown in Fig. 3. The schematic has a symmetrical structure, which consists of two PCSAs, two bit cells and two reference cells. Resistor R_{LRef1} and R_{LRef2} connected in parallel and two transistors comprise the low reference cell, while the high reference cell is made of resistor R_{HRef1}, R_{HRef2} and two transistors. Two inverters (MP1 and MN0, MP2 and MN1) connected to each other form the PCSA in the left side, and inverters (MP5 and MN2, MP6 and MN3) comprise the right side PCSA. NMOS N_{S0}, N_{S1}, N_{S2} and N_{S3} work to switch in three states: write operation, read operation and bitwise logic operation.

B. Working Principle of DualRef Scheme

We take the OR bitwise logic operation with bit pattern (00) as an example to show how to perform bitwise logic operation by the DualRef CIM scheme. As shown in Fig. 3, MTJ00 and MTJ01 are assumed in low resistance state (logic "0"). The procedure of execution is as follows: V_{Pre0}, V_{Pre1}, V_{Sel0}, V_{Sel1}, V_{En0} and V_{En1} are first set to "0", and the drain terminal of NMOS MN0 and MN1 are charged to V_{dd}. When WL1 and WL2 are enabled, V_{Pre0}, V_{En0} and V_{Sel0} switch from "0" to "1", the precharged voltage starts to discharge. The discharge speeds are different due to the different resistances of the reference and bit cells. Therefore, the voltage in the low resistance branch will decline faster [33]. In the example, the resistance of the two bit cells in parallel is less than R_{LRef}, so the voltage in the bit cell branch will decline faster. The two inverters (MP1 and MN0, MP2 and MN1) reach a stable state ($Q1$ = "0", $Q1\_Bar$ = "1") when the voltage of the NMOS MN1 drain terminal is lower than the threshold voltage of the inverter (MP1 and MN0). Finally, the result of OR bitwise logic operation can be got form Q1.

Some CIM schemes are conceptual and not presented in detailed circuit. It is difficult to know their feasibility, even less the reliability and performance. Different from these schemes, The DualRef CIM scheme we proposed in this paper are implemented by slightly modifing the sensing amplifier and the peripheral circuit, can be run in SPICE simulator. In the following, the functionality of the DualRef CIM scheme will be checked, and then its reliability and performance will be evaluated by groups of simulations.

C. Functionality Verification

For the purpose of verifying the functionality of the DualRef CIM scheme within STT-MRAM, a hybrid MTJ/CMOS transient simulation is carried out with a 45 nm PTM CMOS model and a 40 nm compact perpendicular magnetic anisotropy MTJ model [34], [35], [36], [37]. The supply voltage is fixed in $V_{dd} = 1.1V$, and the TMR equals to $TMR = 300\%$, other related parameters are their default values in the MTJ model[38]. We use a larger CMOS transistors channel width in the sensing circuit to enlarge the sensing margin, and use its minimum channel width in the write circuit and logic circuit to eliminate influence of the parasitic capacitor as possible.

Fig. 4 shows the transient simulation waveform of the DualRef CIM scheme, OR and AND bitwise logic operations are performed one after another. The initial state of (MTJ00, MTJ01) is (1, 1), the state of MTJ00 switches from "1" to "0" by applying currents when $Write$ and WL1 are enabled, while MTJ01 switches to "0" after WL2 is enabled. After writting data into the two MTJs, OR bitwise logic operation is carried out when V_{Pre0}/V_{Sel0}/V_{En0} and $Read$ are set to "1". AND bitwise operation are performed by enabling V_{Pre1}/V_{Sel1}/V_{En1}. Because the resistance of the two bit cells is less than that of R_{LRef} and R_{HRef}, the two PCSAs in both left and right side output "0". Afterwards, the computing results can be obtained in Q1 and Q2 for OR and AND bitwise logic operation respectively. The results are consistant with the truth table shown in Table I.

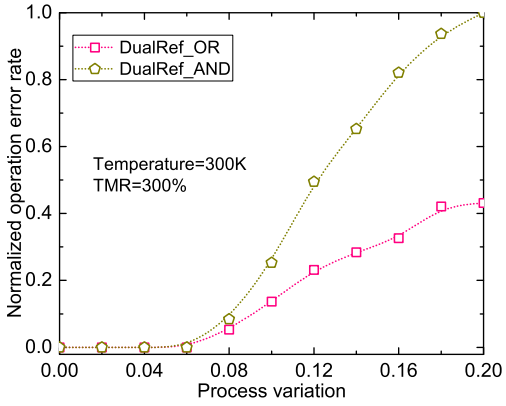

Fig. 5. Normalized operation error rate with respect to the process variation of CMOS transistor.

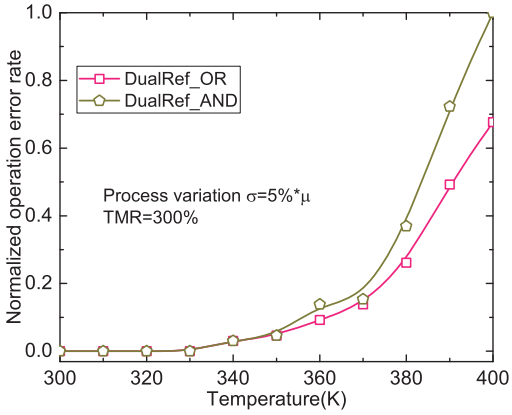

Fig. 6. Normalized operation error rate with respect to the temperature.

With this validated CIM scheme, a series of simulations are carried out in the following section to assess the reliability from the aspects of its operation error rate and sensing margin, and performance in terms of operation delay and dynamic energy consumption with respect to TMR and supply voltage.

III. Design Space Exploration

Besides the feasibility, the reliability and performance are also important for the DualRef CIM implementation when it suffers from PVT (Process, Voltage, Temperature) variations. In this section, we carried out six groups of simulations to assess the CIM scheme. Operation error rate and sensing margin are used to indicate the reliability, while the performance is evaluated by calculating the operation delay and dynamic energy consumption. The value at every point in the opeartion error rate are the arithmetic mean of many times Monte Carlo simulations.

A. Operation Error Rate

Operation error rate is used to indicate the reliability of the DualRef CIM scheme directly. There are four bit patterns: (00), (01), (10) and (11) in OR or AND bitwise logic operation. The operation error rate are the arithmatic mean at each bit pattern. We measured the operation error rate in different process variations of CMOS transistor and different temperatures. The temperature is fixed at 300K, and the TMR is set to 300% when obtained operation error rate with the process variation by varying from 0% to 20%; while evaluating the operation error rate with respect to the temperature, the process variation is set to 5%, and the TMR is fixed at 300%. The measured results are shown in Fig. 5 and 6 respectively. As can be seen from the two figures, operation error rate keep very low at small process variations, and it is the same in low temperatures. However operation error rate raises up rapidly after 6% process variation and 330K. With large process variations or higher temperatures, both the resistance of the bit cell and the reference cell changed, the driving currents of transistors decreased. All of this resulted in the reduction of the sensing margin, which can be explain more and more operation errors occur.

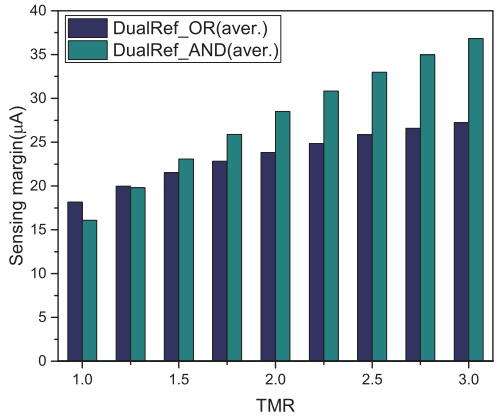

Fig. 7. The sensing margin of bitwise logic operations. It is defined as the currents difference of the two discharge branches in the sensing amplifiers.

B. Sensing Margin

As known from the introduction in Section II, the computing of these bitwise logic operations are executed in sensing amplifiers. The sensing margin indicates how many variations the DualRef CIM scheme can tolerate to ensure computing without errors. Therefore, sensing margin of OR and AND bitwise logic operations are the important indicator of the reliability, which is defined as the current difference of the two discharge branches in the sensing amplifiers. The sensing margin are calculated without considering the process variation of transistor. The resistance difference between the low and high resistace state of MTJ device varies as TMR. Therefore, the sensing margin were checked by increasing TMR from 100% to 300%. The calculated sensing margin are shown in Fig. 7. Every data is the average value of the sensing margin at the four bit patterns:(00), (01), (10) and (11). As can be seen from this figure, the smallest sensing margin is more than 15μA, and the biggest one is about 35μA. Both the sensing margin of OR and AND bitwise logic operation

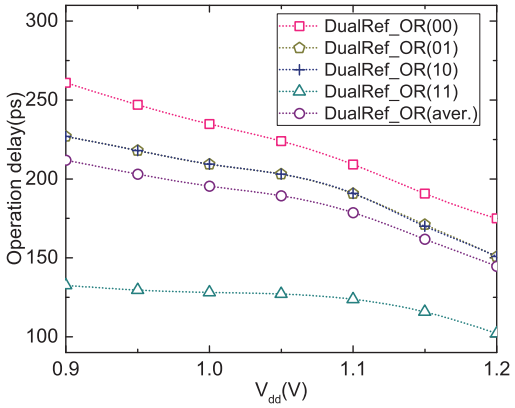

Fig. 8. The operation delay of the OR bitwise logic opeartion. The simulations are carried out with the fixed temperature 300K and no process variation.

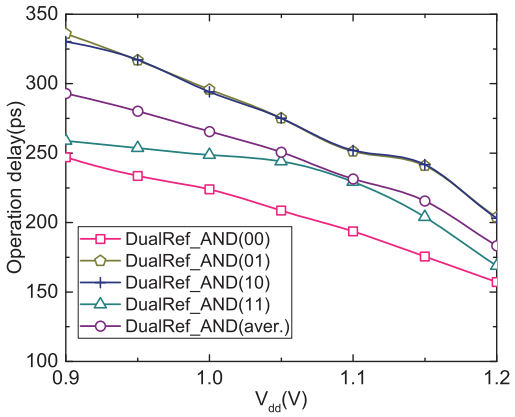

Fig. 9. The operation delay of the AND bitwise logic opeartion. The simulations are carried out with the fixed temperature 300K and no process variation.

arise linearly by increasing TMR. However, the increasing speed and sensing margin of AND bitwise logic operation are more than that of OR bitwise logic operation. With increasing of TMR, only the resistance of MTJs in anti-parallel state increase. So the resistance of the two bit cells in parallel at bit pattern (00) keeps constant with the increased TMR, and increases at bit pattern (01) and (10), and increases faster at bit pattern (11). According to truth Table I, the low reference cell was used to distinguish bit pattern (00) from others, but the resistance of which does not increase as fast as TMR. Therefore, the above explained why the sensing margin of OR bitwise logic operation and its increasing speed are less than that of AND bitwise logic operation.

C. Operation Delay

The operation delay represents the time consumed by one bitwise logic operation. The delay of OR and AND bitwise logic operations for the DualRef CIM scheme are measured from the rising edge of enable signal to the time when output results reach stable state. The operation delay are calculated with respect to the supply voltage V_{dd}, by varying from 0.9V to 1.2V. The results are shown in Fig. 8 and 9. It can be seen in the two figures that both the delay of OR and AND bitwise logic operations decline with increasing supply voltage V_{dd}. Delay time is dominated by the charge and discharge time of capacitor in circuit. Larger supply voltage can reduce the time. The average delay of AND bitwise logic operation decrease from 295ps to 180ps when supply voltage increases, and are more than that of OR bitwise logic operation. That is because the higher resistance reference cell is used, and the current discharge of it when sensing is slower than with the low resistance reference cell. The difference delay within different bit patterns are caused by both the effect of current discharge speed and parasitic capacitors.

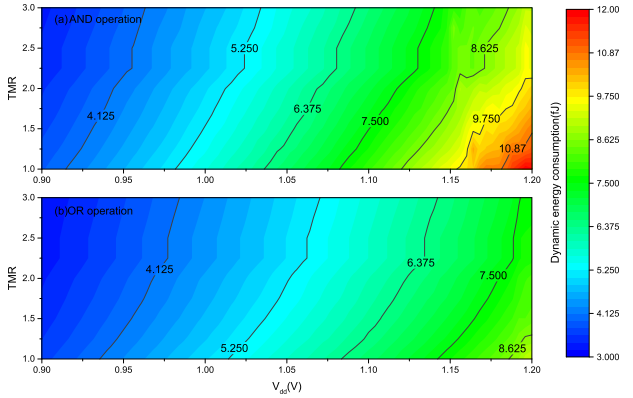

Fig. 10. The dynamic energy consumption of OR and AND bitwise logic operations. It is measured without process variation, and the temperature is fixed at 300K.

D. Dynamic Energy Consumption

The operation dynamic energy consumption of the DualRef CIM scheme is calculated by integrating the product of voltage and current of the two discharge branches in the sensing amplifiers with respect to its operation delay, and shown in Fig. 10. These data are obtained by fixing the temperature at 300K and without process variation. As shown in this figure, single bitwise logic operation consumes several femto Joules, and which increases with the supply voltage V_{dd}, but slowly declines with the TMR. Increasing V_{dd} shortens the operation time, but it will linearly raise up the current. Therefore, the energy consumption surely goes up with larger V_{dd}. The resistance of discharge branches decline as the TMR, which results in the reduction of discharge current, so the dynamic energy consumed by single bitwise logic operation decrease with TMR. It is found that more energy are consumed in AND bitwise logic operation with same TMR and V_{dd} by comparing Fig. 10 (a) and (b), which is caused by its bigger operation delay as shown in Fig. 8 and 9.

In summary, the reliability and performance of the DualRef CIM scheme were quantitatively analyzed. We found that, the reliability of DualRef can be enhanced by enlarging the TMR as possible. However, as the indicator of its performance, operation delay and dynamic energy consumption are irrec-

oncilable. Increasing voltage supply can reduce the operation delay, but result in more energy consumption. These results provide how robust is the DualRef CIM implementation under PVT varations.

IV. Conclusion

A CIM implementation DualRef is proposed here within STT-MRAM. DualRef can be used to perform OR and AND bitwise logic operation, other bitwise logic operation can also be supported by adding peripheral circuit. The functionality, reliability and performance of the CIM implementation are evaluated at 45nm technology node. Simulations reveale that DualRef can work correctly when the temperature is less than 330K and process variation of the CMOS transistor does not supass 6%. Improving TMR of MTJ device can enhance the reliability of this CIM scheme. This work provides a robust circuit scheme to implement CIM at circuit level and explores the design space, but efforts on system, instruction set and software interface are also expected.

References

[1] M. Imani, S. Gupta, and T. Rosing, "Ultra-efficient processing in-memory for data intensive applications," in *Proceedings of the 54th Annual Design Automation Conference 2017*, Jun. 2017, pp. 6:1–6:6.

[2] J. Zhou, X. Yang, J. Wu*, et al.*, "A memristor-based architecture combining memory and image processing," *Sci. China Inform. Sci.*, vol. 57, no. 5, pp. 1–12, May 2014.

[3] N. S. Kim, T. Austin, D. Blaauw*, et al.*, "Leakage current: Moore's law meets static power," *Computer*, vol. 36, no. 12, pp. 68–75, Dec. 2003.

[4] S. W. Keckler, W. J. Dally, B. Khailany*, et al.*, "GPUs and the future of parallel computing," *IEEE Micro*, vol. 31, no. 5, pp. 7–17, Sep. 2011.

[5] S. Li, C. Xu, Q. Zou*, et al.*, "Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories," in *2016 53nd ACM/EDAC/IEEE Design Automation Conference*, Jun. 2016, pp. 1–6.

[6] L. Koskinen, J. Tissari, J. Teittinen*, et al.*, "A performance case-study on memristive computing-in-memory versus Von Neumann architecture," in *2016 Data Compression Conference*, Mar. 2016, pp. 613–613.

[7] J. Ahn, S. Hong, S. Yoo*, et al.*, "A scalable processing-in-memory accelerator for parallel graph processing," in *2015 ACM/IEEE 42nd Annual International Symposium on Computer Architecture*, Jun. 2015, pp. 105–117.

[8] K. Chen, S. Li, N. Muralimanohar*, et al.*, "CACTI-3DD: Architecture-level modeling for 3D die-stacked DRAM main memory," in *2012 Design, Automation Test in Europe Conference Exhibition*, Mar. 2012, pp. 33–38.

[9] X. Dong, X. Wu, G. Sun*, et al.*, "Circuit and microarchitecture evaluation of 3D stacking magnetic RAM (MRAM) as a universal memory replacement," in *2008 45th ACM/IEEE Design Automation Conference*, Jun. 2008, pp. 554–559.

[10] W. Zhao and G. Prenat, *Spintronics-Based Computing*. Switzerland: Springer, 2015.

[11] C. J. Xue, G. Sun, Y. Zhang*, et al.*, "Emerging non-volatile memories: opportunities and challenges," in *2011 Proceedings of the Ninth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, Oct. 2011, pp. 325–334.

[12] F. Parveen, S. Angizi, Z. He*, et al.*, "Low power in-memory computing based on dual-mode SOT-MRAM," in *2017 IEEE/ACM International Symposium on Low Power Electronics and Design*, Jul. 2017, pp. 1–6.

[13] W. Kang, Z. Wang, Y. Zhang*, et al.*, "Spintronic logic design methodology based on spin hall effect-driven magnetic tunnel junctions," *J. Phys. D: Appl. Phys.*, vol. 49, no. 6, p. 065008, Jan. 2016.

[14] H. Zhang, W. Kang, L. Wang*, et al.*, "Stateful reconfigurable logic via a single-voltage-gated spin hall-effect driven magnetic tunnel junction in a spintronic memory," *IEEE Trans. Electron Devices*, vol. 64, no. 10, pp. 4295–4301, Oct. 2017.

[15] W. Kang, Y. Zhang, Z. Wang*, et al.*, "Spintronics: Emerging ultra-low-power circuits and systems beyond MOS technology," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 12, no. 2, pp. 16:1–16:42, Aug. 2015.

[16] S. A. Wolf, D. D. Awschalom, R. A. Buhrman*, et al.*, "Spintronics: A spin-based electronics vision for the future," *Science*, vol. 294, no. 5546, pp. 1488–1495, Nov. 2001.

[17] H. S. P. Wong and S. Salahuddin, "Memory leads the way to better computing," *Nat. Nanotechnol.*, vol. 10, no. 3, pp. 191–194, Mar. 2015.

[18] W. Kang, L. Zhang, J.-O. Klein*, et al.*, "Reconfigurable codesign of STT-MRAM under process variations in deeply scaled technology," *IEEE Trans. Electron Devices*, vol. 62, no. 6, pp. 1769–1777, Jun. 2015.

[19] D. Patterson, T. Anderson, N. Cardwell*, et al.*, "A case for intelligent ram," *IEEE Micro*, vol. 17, no. 2, pp. 34–44, Mar. 1997.

[20] M. Imani, Y. Kim, and T. Rosing, "MPIM: Multi-purpose in-memory processing using configurable resistive memory," in *2017 22nd Asia and South Pacific Design Automation Conference*, Jan. 2017, pp. 757–763.

[21] D. Fan, S. Angizi, and Z. He, "In-memory computing with spintronic devices," in *2017 IEEE Computer Society Annual Symposium on VLSI*, Jul. 2017, pp. 683–688.

[22] J. Yu, R. Nane, A. Haron*, et al.*, "Skeleton-based design and simulation flow for computation-in-memory architectures," in *2016 IEEE/ACM International Symposium on Nanoscale Architectures*, Jul. 2016, pp. 165–170.

[23] A. Haron, J. Yu, R. Nane*, et al.*, "Parallel matrix multiplication on memristor-based computation-in-memory architecture," in *2016 International Conference on High Performance Computing Simulation*, Jul. 2016, pp. 759–766.

[24] S. Hamdioui, M. Taouil, H. A. D. Nguyen*, et al.*, "CIM100x: Computation in-memory architecture based on resistive devices," in *2016 15th International Workshop on Cellular Nanoscale Networks and their Applications*, Aug. 2016, pp. 1–2.

[25] P. Chi, S. Li, C. Xu*, et al.*, "PRIME: A novel processing-in-memory architecture for neural network computation in ReRAM-based main memory," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture*, vol. 44. IEEE, Jun. 2016, pp. 27–39.

[26] W. Kang, H. Wang, Z. Wang*, et al.*, "In-memory processing paradigm for bitwise logic operations in STT-MRAM," *IEEE Trans. Magn.*, vol. 53, no. 11, pp. 1–4, Nov. 2017.

[27] S. Hamdioui, L. Xie, H. A. D. Nguyen*, et al.*, "Memristor based computation-in-memory architecture for data-intensive applications," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, Mar. 2015, pp. 1718–1725.

[28] S. Hamdioui, M. Taouil, H. A. D. Nguyen*, et al.*, "Memristor: the enabler of computation-in-memory architecture for big-data," in *2015 International Conference on Memristive Systems*, Nov. 2015, pp. 1–3.

[29] S. Hamdioui, "Computation in memory for data-intensive applications: Beyond CMOS and beyond Von-Neumann," in *Proceedings of the 18th International Workshop on Software and Compilers for Embedded Systems*, Jun. 2015, pp. 1–1.

[30] J. J. Yang, D. B. Strukov, and D. R. Stewart, "Memristive devices for computing," *Nat. Nanotechnol.*, vol. 8, no. 1, pp. 13–24, Dec. 2013.

[31] L. Zhang, W. Kang, Y. Zhang*, et al.*, "Channel modeling and reliability enhancement design techniques for STT-MRAM," in *2015 IEEE Computer Society Annual Symposium on VLSI*, Jul. 2015, pp. 461–466.

[32] H. Cai, Y. Wang, L. A. D. B. Naviner*, et al.*, "Robust ultra-low power non-volatile logic-in-memory circuits in FD-SOI technology," *IEEE Trans. Circuits Syst. I*, vol. 64, no. 4, pp. 847–857, Apr. 2017.

[33] W. Zhao, C. Chappert, V. Javerliac*, et al.*, "High speed, high stability and low power sensing amplifier for MTJ/CMOS hybrid logic circuits," *IEEE Trans. Magn.*, vol. 45, no. 10, pp. 3784–3787, Oct. 2009.

[34] W. Zhao and Y. Cao, "Predictive technology model for nano-CMOS design exploration," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 3, no. 1, Apr. 2007.

[35] Y. Wang, Y. Zhang, E. Deng*, et al.*, "Compact model of magnetic tunnel junction with stochastic spin transfer torque switching for reliability analyses," *Microelectron. Reliab.*, vol. 54, no. 9, pp. 1774 – 1778, Mar. 2014.

[36] L. Zhang, A. Todri-Sanial, W. Kang*, et al.*, "Quantitative evaluation of reliability and performance for STT-MRAM," in *2016 IEEE International Symposium on Circuits and Systems*, May 2016, pp. 1150–1153.

[37] L. Zhang, Y. Cheng, W. Kang*, et al.*, "Addressing the thermal issues of stt-mram from compact modeling to design techniques," *IEEE Trans. Nanotechnol.*, vol. 17, no. 2, pp. 345–352, Mar. 2018.

[38] M. Wang, W. Cai, K. Cao*, et al.*, "Current-induced magnetization switching in atom-thick tungsten engineered perpendicular magnetic tunnel junctions with large tunnel magnetoresistance," *Nat. Commun.*, vol. 9, no. 1, p. 671, Feb. 2018.