IEICE Electronics Express, Vol.19, No.20, 1–6

LETTER

An in-memory computing multiply-and-accumulate circuit based on ternary STT-MRAMs for convolutional neural networks

Guihua Zhao[1], Xing Jin[1], Huafeng Ye[2], Yating Peng[2], Wei Liu[1], Ningyuan Yin[2], Weichong Chen[2], Jianjun Chen[1], Ximing Li[1], and Zhiyi Yu[2, 3, a)]

Abstract In-memory computing (IMC) quantized neural network (QNN) accelerators are extensively used to improve energy-efficiency. However, ternary neural network (TNN) accelerators with bitwise operations in non-volatile memory are lacked. In addition, specific accelerators are generally used for a single algorithm with limited applications. In this report, a multiply-and-accumulate (MAC) circuit based on ternary spin-torque transfer magnetic random access memory (STT-MRAM) is proposed, which allows writing, reading, and multiplying operations in memory and accumulations near memory. The design is a promising scheme to implement hybrid binary and ternary neural network accelerators.

Keywords: in-memory computing, STT-MRAM, multiply-and-accumulate, ternary neural networks, binary neural networks
Classification: Circuits and modules

1. Introduction

Convolutional neural networks (CNNs) are developing rapidly and applied widely in computer vision [1, 2, 3, 4, 5, 6, 7, 8]. For CNNs, massive real-valued weight parameters and high-precision operations demand enormous memory and computation resources, and result in high power consumption and latency. The training and prediction tasks are limited for real-time or energy-critical applications in embedded systems. As we know, MAC operations cause the most computing overhead. To address this problem, many technologies of algorithms and hardware have been developed.

At the algorithm level, QNNs are presented in the past [9, 10, 11, 12, 13, 14, 15]. Typically, binary weight networks (BWNs) and ternary weight networks (TWNs) constrain the weights to be binary $-\alpha$, α and ternary $-\alpha$, 0, α, respectively. In this way, the MAC operations are converted to only additions, which consume less computing and memory resources. Further, the binary neural networks (BNNs) and ternary neural networks (TNNs) constrain both weights and activations to binary -1, 1 and ternary -1, 0, 1 respectively. As a result, the MAC operations are composed of only bit-wise logic (i.e. XOR and AND) and bit-count operations and further lower the overhead of memory and computing. Besides, ternary-binary networks (TBN) provide an optimal tradeoff between memory and efficiency.

At the hardware level, due to high energy and latency for traditional GPU and FPGA based computing, emerging IMC technology based on SRAMs, ReRAMs, and MRAMs were introduced to improve the energy and latency [16, 17, 18, 19, 20, 21, 22]. STT-MRAM is a non-volatile device with high density and near-zero leakage. STT-MRAM based MAC can perform a range of arithmetic, logic, and vector operations for general purpose or binary/ternary CNNs [23, 24, 25, 26, 27, 28]. The multilevel cell (MLC) STT-MRAM was used for bitwise operations of BNN. However, it increased the process and operation complexity. Nonvolatile logic gates storing weights were used for TNN. However, the input was volatile, which implied extra cost of data read and transfer. The schemes of operating both activations and weights in the memory is a possible way of improving the enery-efficiency. Besides, specific functions make current MAC circuits only suitable for single algorithm. Mutiple functions can make MAC circuits more flexible and suitable for different algorithms. As we know, for aformentioned low-bit width neural networks, the research of hybrid binary and teranry MAC circuits with bitwise operations in memory are lacked, which implies the necessity of exploration.

In this paper, an MAC circuit architectrue based on ternary STT-MRAMs is proposed. The circuit allows novel writing, reading and hybrid binary and ternary mutiplicaitons. Our design offers a uniform and promising MAC circuit which can be simutaneously used for BNN, TBN and TNN.

2. Ternary STT-MRAM-based MAC architecture

An architecure of ternary STT-MRAM-based MAC are proposed in Fig. 1. The orange part is the array composed of ternary memory cells. Rows are addressed by an enhanced decocoder with input of $Addr_i$ and $Addr_j$ and the wordline (WL) select signals. The write, read and multiply operations are controled by a controler. When read or multiply are enabled, the global reference circuit generates reference voltages V_{refs}, and the ternary MTJ MAC array output sensing voltages such as V_i and V_j [29]. Next, the sensing and reference voltages are input to the read/multiply sensing circuit to get the results. For accumulations, the multiply results are input to a specialized ternary accumulators enabled by

[1] School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China
[2] School of Microelectronics Science and Technology, Sun Yat-sen University, Zhuhai 519082, China
[3] Guangdong Provincial Key Laboratory of Optoelectronic Information Processing Chips and Systems, Sun Yat-Sen University, China
[a)] yuzhiyi@mail.sysu.edu.cn

DOI: 10.1587/elex.19.20220399
Received September 3, 2022
Accepted September 6, 2022
Publicized September 20, 2022
Copyedited October 25, 2022

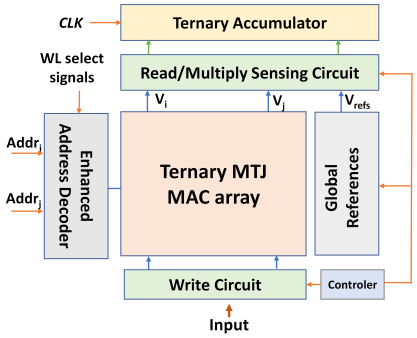

Fig. 1 Architecure of ternary STT-MRAM-based MAC.

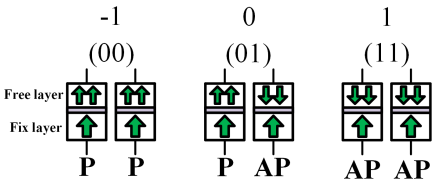

Fig. 2 Ternary encoding with MTJ states

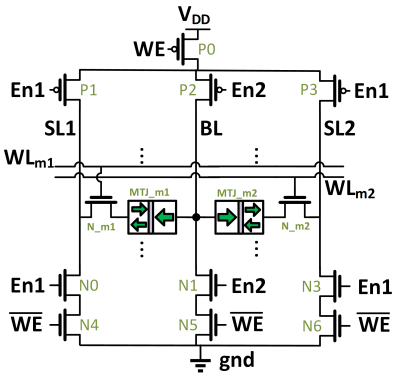

Fig. 3 Ternary STT-MRAM write circuit.

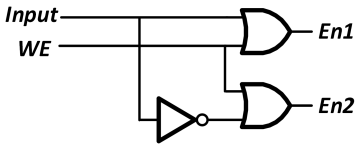

Fig. 4 Peripheral ternary STT-MRAM write circuit.

CLK edge signals near the memory array. Relavent circuit details are introduced in the following paragraphs.

3. STT-MTJ device

An MTJ is mainly composed of two ferromagnetic films and a thin oxide barrier as shown in Fig. 2. For parallel (P) or anti-parallel (AP) magnetizaiton orientation, MTJs have resistances of R_P or R_{AP}, which are correlated by tunnel magnetic resistance ratio (TMR) = $(R_{AP} - R_P)/R_P$. Based on STT effect, current beyond threshold values (I_C) can switch the magnetization of free layer. In this situation, current flowing into or out of the fix layer can switch the MTJ to AP and P states, respectively when the duration is beyond threshold switching time [30].

4. Ternary STT-MRAM cell

Conventionally, for BNNs the activation and weights are binarized as -1 and 1, which are programed as 0 and 1 correponding to high and low resistance states in the memory. Meanwhile, for TNN they are ternarized as -1, 1 and 0, which are programed as 01, 10 and 00 coresponding to high-low, low-high and high-high resistance states. In this work, unprecedentedly, -1, 1 and 0 are programed as 00, 11 and 01 correspoding to low-low (P-P), high-high (AP-AP) and low-high (P-AP) resistance states of dual MTJs, as shown in Fig. 2. This method encodes ternary memory to different resistances, making the read and multiply simple.

In our design, a ternary memory cell consists of dual serial 1T1MTJ sub-cells as shown in Fig. 3. Different with other reported works, the electrodes of free layer instead of fix layer of MTJs are connected to NMOS source/drain, which can strengthen the currents to switch the MTJ from P to AP states. For this scheme, the left and right sub-cells have a symmetric line of BL, which makes -1 and 1 writable within one step when BL, SL1, SL2 have proper voltages. The write, read and multiply circuits are introduced in the following parts.

5. Ternary STT-MRAM write circuit

Due to that a ternary value is encoded to two binary bits, the write of ternary values can be decomposed to operations of binary 0 and 1. The write circuits are presented in Fig. 3 with a peripheral circuit as shown in Fig. 4. There are four selections of writing.

1. No write
   When $WE = 1, \overline{WE} = 0$. whatever $Input$ is, $En1 = 1, En2 = 1$ according to the logic shown in Fig. 4. In this case, the transistors P0, P1, P2, P3, N4, N5 and N6 are turned off according to the circuits in Fig. 3. As a result, the current pathway between sub-cells and power/ground are cut off and the write is forbiddened.
2. Write -1
   When $WE = 0, \overline{WE} = 1$. If $Input = 0$, $En1 = 0$ and $En2 = 1$. In this case, the transistors P0, P1, P3, N1, N4, N5, N6 are turned on and P2, N0, N3 are turned off. Then, both SL1 and SL2 are high and BL is low. Once WL_{m1} and WL_{m2} are set to be high, there will be current flowing from side free layers to middle fix layers, which sets MTJs to be 00.
3. Write 1
   When $WE = 0, \overline{WE} = 1$. If $Input = 1$, $En1 = 1$ and $En2 = 0$. In this case, the transistors P0, P2, N0, N3, N4, N5, N6 are turned on and P1, P3, N1 are turned off. Then, BL is high and SL1, SL2 are both low. Once WL_{m1} and WL_{m2} are set to be high, there will be currents flowing from middle fix layers to side free layers, which sets MTJs to be 11.
4. Write 0
   When $WE = 0, \overline{WE} = 1$. Ternary 0 can be written by two-step writing of binary 0 and 1. (1) When $Input = 0$, similar to the case of writing of -1, both SL1 and SL2 are high and BL is low. Once WL_{m1} and WL_{m2} are set to be high and low respectively, the first MTJ is

set to be binary 0 and the second MTJ keeps its state. (2) When $Input = 1$, similar to the case of writing of 1, BL is high and SL1, SL2 are both low. Once WL_{m1} and WL_{m2} are set to be low and high respectively, the first MTJ keeps binary 0 and the second one is set to be binary 1. As a result, ternary 0 is written successfully with encoding of 01.

6. Ternary STT-MRAM read circuit

The read circuit is presented in Fig. 5. The basic princile is comparing the resistances of cells with the reference ones and converted the relationship into two-bit binary values for ternay encoding. To achieve this goal, we use a current mirror to generate the same magnitude of current to pass through the resistances to ground and comparing the voltage drop of the resistances. The selection of reference resistances is the key step. The total resistances of two serial MTJs are $R_{00} = R_p + R_p$, $R_{11} = R_{Ap} + R_{Ap}$, $R_{01} = R_p + R_{Ap}$ corresponding to -1, 1, 0, respectively. The reference resistances are set as $R_{ref1} = R_{AP} + R_{AP}/2$ and $R_{ref2} = R_P + R_{AP}/2$. If $TMR > 100\%$, $R_{00} < R_{ref2} < R_{01} < R_{ref1} < R_{11}$. Then, the voltage drop $V_{00} < V_{ref2} < V_{01} < V_{ref1} < V_{11}$ and the ternary memory value can be correctly read out as dual bits of A and B as shown in Fig. 5. The read operations are run by setting RE to be 0, WL_{m1} and WL_{m2} to be 1 simutaneously. Then, the read current flows along the path of SL1-BL-SL2-N0-gnd.

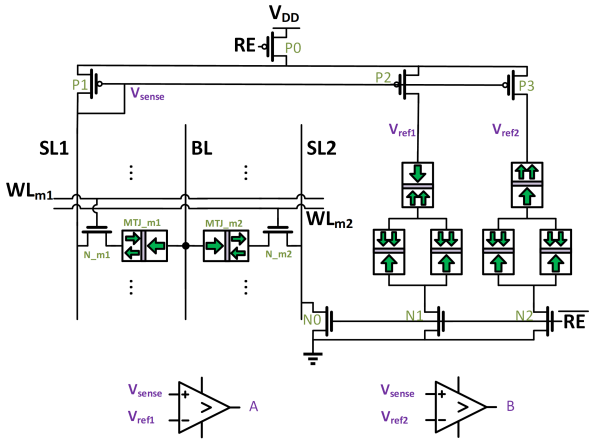

Fig. 5 Ternary STT-MRAM read circuit.

7. Ternary STT-MRAM multiply circuit

The multiply circuits are presented in Fig. 6. Like the read, the multiply operations compare the resistances of cells with the reference ones and converted the relationship into two-bit binary values. The multiply circuits share the same current mirror with the read ones and get the voltage drops from the resistances. The selection of reference resistances is also the key step.

The multiplication between ternary $(-1, 0, 1)$ and $(-1, 0, 1)$ can be classified to six cases, inluding $(-1) \times (-1), (-1) \times 0, (-1) \times 1, 0 \times 0, 0 \times 1, 1 \times 1$. $R_{sense} = (R_{MTJ_{m1}} \parallel R_{MTJ_{n1}}) + (R_{MTJ_{m2}} \parallel R_{MTJ_{n2}})$. Specifically, $R_{-1,-1} = (R_P \parallel R_P) + (R_P \parallel R_P)$, $R_{-1,0} = (R_P \parallel R_P) + (R_P \parallel R_{AP})$, $R_{-1,1} = (R_P \parallel R_{AP}) + (R_P \parallel R_{AP})$, $R_{0,0} = (R_P \parallel R_P) + (R_{AP} \parallel R_{AP})$, $R_{0,1} = (R_P \parallel R_{AP}) + (R_{AP} \parallel R_{AP})$, $R_{1,1} = (R_{AP} \parallel R_{AP}) + (R_{AP} \parallel R_{AP})$. Four reference resistances are set to be $R_{ref3} = (R_P \parallel R_P) + (R_P \parallel R_{AP}/2)$, $R_{ref4} = (R_{AP} \parallel R_{AP}) + (R_{AP} \parallel R_{AP}/2)$, $R_{ref5} = (R_{AP} \parallel R_P) + (R_{AP} \parallel R_{AP}/2)$ and $R_{ref6} = (R_P \parallel R_{AP}) + (R_P \parallel R_{AP}/2)$. If $TMR > 100\%$, the resistances conform to the relationship $R_{-1,-1} < R_{ref3} < R_{-1,0} < R_{ref6} < R_{-1,1} < R_{ref5} < R_{0,0} < R_{0,1} < R_{ref4} < R_{1,1}$, Therefore, $V_{-1,-1} < V_{ref3} < V_{-1,0} < V_{ref6} < V_{-1,1} < V_{ref5} < V_{0,0} < V_{0,1} < V_{ref4} < V_{1,1}$. Using the comparing circuits the intermediate outputs C, D, E, F can be obtained as shown in Fig. 6. Then, the two bits of multiply result are Out1 and Out2, where Out1=$\overline{CD}$ and Out2=$\overline{EF}$ only using NAND operations. The multiply circuit can be used in bitwise operatins for binary-binary, ternary-binary, binary-ternary and ternary-teranry values row by row. The multiply operations are performed by setting RE to be low and $WL_{m1}, WL_{m2}, WL_{n1}, WL_{n2}$ to be high simutateously to turn on the transistors P0-P5 and N0-N4. The multiply current flows along the path of SL1-BL-SL2-N0-gnd, which is similar to the read operations.

8. Enhanced address decoder

The address $Addr_i$ and $Addr_j$ are respectively input to two independent decoders to make the ith and jth output wires high and corresponding outputs are ORed to enable the rows of ternary cell to be operated.Simultaneously, two WL select

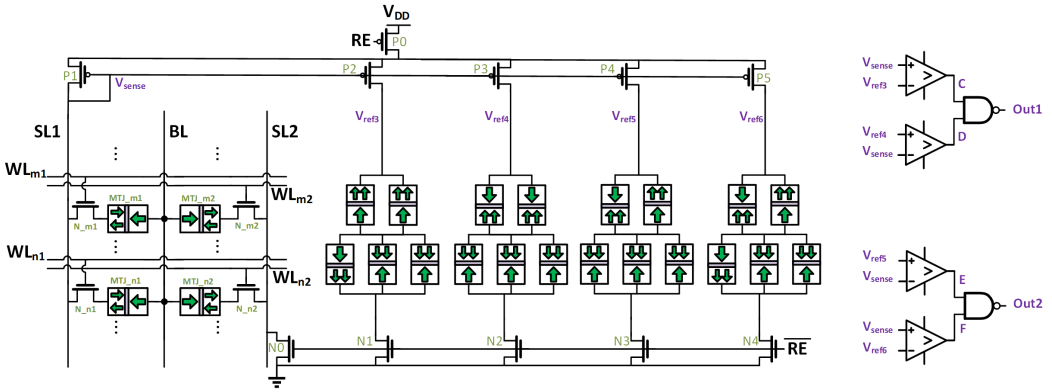

Fig. 6 Ternary STT-MRAM multiply circuit.

signals are used to set the sub-rows of 1T1MTJ to be in one of four states including (Enabled, Unenabled), (Unenabled, Enabled), (Enabled, Enabled) and (Unenabled, Unenabled). The four states can meet the requirements of writing, reading, multiplying and no operation.

9. Ternary accumulator

For a MAC operation, the outputs from row-by-row multiply should be added or counted to get the accumulation result. Mathematically, $-1 = [(-1)+(-1)]/2, 0 = [(-1)+1]/2, 1 = (1+1)/2$. According to the aforementioned encoding, when binary 0 is input, 1 is subtracted. when binary 1 is input, 1 is added. After the counting is finished, one-bit right shift is performed and the accumulation result is obtained.

10. Circuit simulation and evaluation

To verify the function of proposed circuits, transient simulation of ternary write, read, and multiply are perfomred. We use the same MTJ models and similar parameters of 40 nm MTJ reported by Reference [30]. $R_P = 3219\,\Omega$, $I_{C,(P-AP)} = 72\,\mu A$, $I_{C,(AP-P)} = 28\,\mu A$. The only difference is TMR(0) of 150% instead of 120%. Likewise, 40 nm CMOS technology is used for hybrid MTJ/CMOS circuit simulation. For the simulation as shown in Fig. 7, V2, V3 and V4 indicate the times for reading −1, 1 and 0 respectively after writing. The reading results are consistent with stored values of the ternary cell. For the simulation as shown

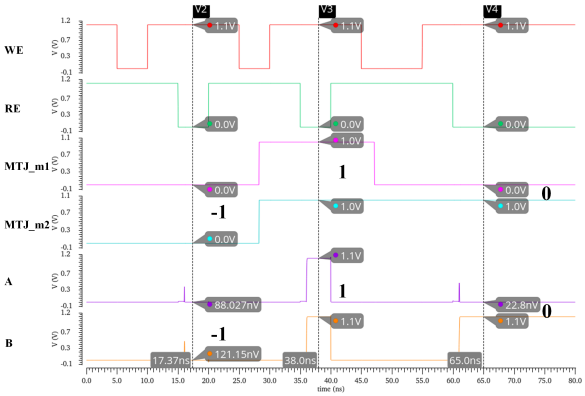

Fig. 7 Write and read simulation of ternary −1, 1 and 0.

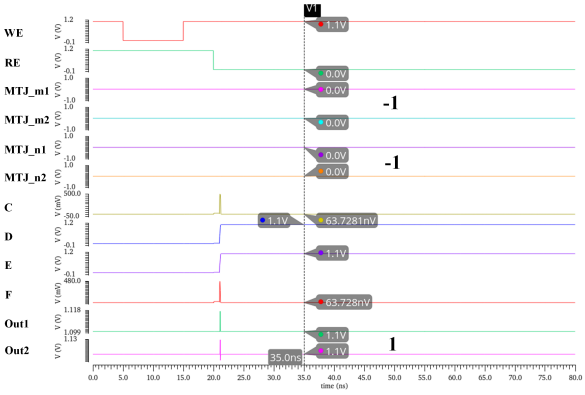

Fig. 8 Write and multiply simulation of ternary −1 and −1.

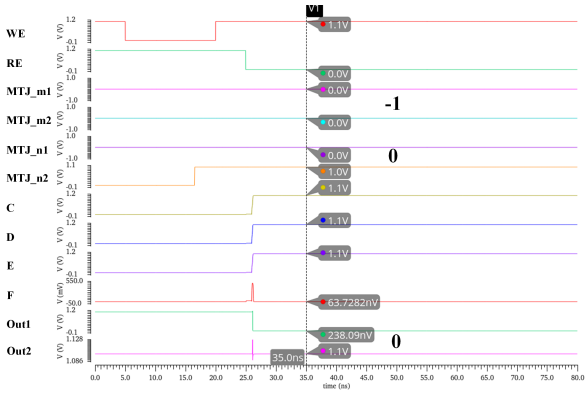

Fig. 9 Write and multiply simulation of ternary −1 and 0.

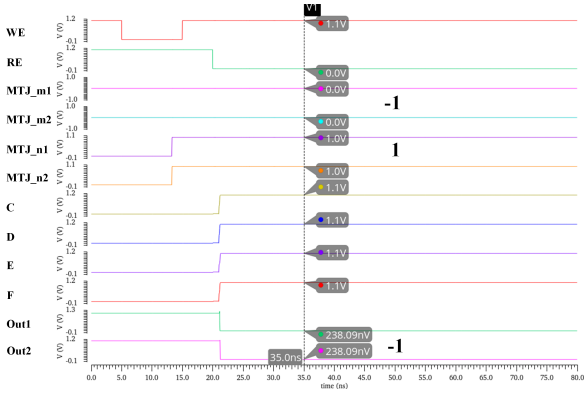

Fig. 10 Write and multiply simulation of ternary −1 and 1.

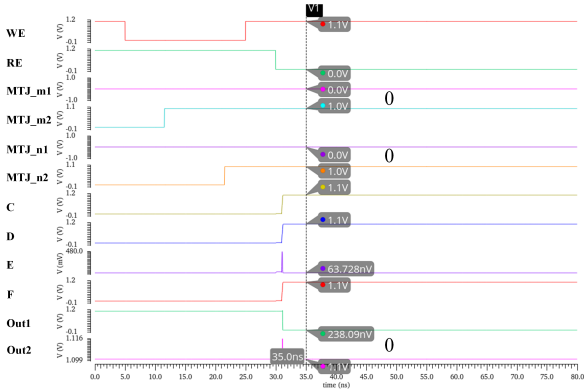

Fig. 11 Write and multiply simulation of ternary 0 and 0.

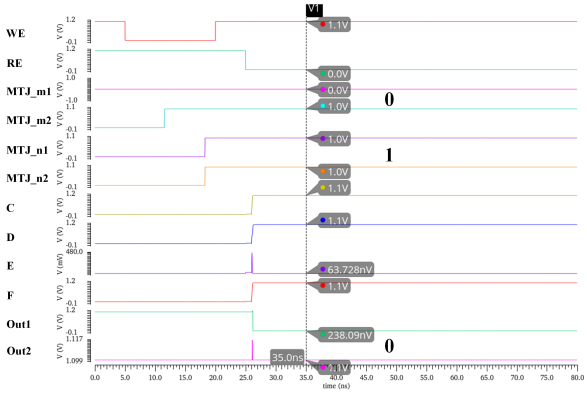

Fig. 12 Write and multiply simulation of ternary 0 and 1

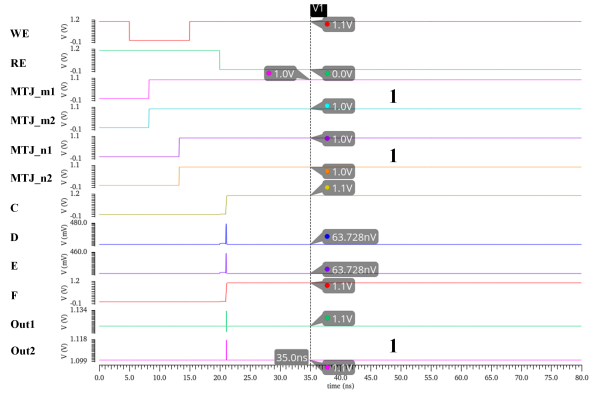

Fig. 13 Write and multiply simulation of ternary 1 and 1.

Table I Performance comparison.

Ternary architecture	Energy[fJ]	Delay[ns]	Area[a.u.]
NV-LIM [24]	20.0	0.217	201.0
NV-Multiply	24.6	0.181	372
Comparators	6.7 (27.2%)	–	68 (18.3%)
References	5.3 (21.5%)	–	172 (46.2%)

in Fig. 8–Fig. 13, V1 indicates the time for multiplications of six cases after writing. The simulated multiply results are equel to the theoritical values of multiplying the two stored values of two ternary cells. Therefore, the results of transient simulation indicate the correct function of proposed write, read and mutiply circuits.

The perfomance of ternary multiply circuits are listed in Table I and are compared with the-state-of-the-art results [24]. The simulated performance is based on the operation of 1×1 to show the performance potential. The latency is 16.6% less than NV-LIM. The energy and area are 23% and 85.1% more than NV-LIM. As shown in the brackets of Table I, refereces and comparators have great contributions to the energy and area, which can be averaged and lowered due to the sharing in MRAM arrays. The performance can be further optimized by justifying the parameters such as R_P, TMR and I_c [30]. Monte Carlo method is used to simulating the write-multiply circiuts to learn the success rate [31]. To be simple, every time we stochastically change the crosss-sectional area of all the MTJs independently with a variation which is defined as the ratio of standard deviation and 40 nm-diameter area. Write-multiply operations are repeatedly performed for 100 times for each variations (0, 1%, 2%, 3%, 4%) and the write-multiply success rates are obtained. The allowed variations for 100% success rate must be less than 1% as shown in Fig. 14. Though sensitive to MTJ variations, the circuit remains promising for QNNs due to fault tolerance of neural networks [32].

11. Conclusion

In this work, a MAC circuit based on ternary STT-MRAMs is proposed. A novel encoding method and ternary memory cells of 2T2MTJ configuraition are presented. The write, read and multiply circuits are proposed and verified by simulation. Besides, the decoder and accumulator cicuits are disscused. The proposed circuits offer promising nonvolatile IMC schemes suitable for BNN, TBN and TNN simultaneously.

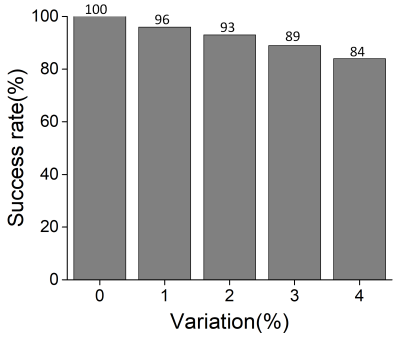

Fig. 14 Write-multiply success rate depending on MTJ area variations.

Acknowledgments

This work was supported in part by grants from National Key R&D Program of China 2017YFA0206200&2018YFB2202601, and national nature science foundation of China (NSFC) under grant No. 61674173, 61834005, and 61902443.

References

[1] J. Dai, *et al.*: "R-FCN: object detection via region-based fully convolutional networks," NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing Systems (2016) 379 (DOI: 10.5555/3157096.3157139).
[2] R. Girshick: "Fast R-CNN," 2015 IEEE International Conference on Computer Vision (ICCV) (2015) 1440 (DOI: 10.1109/ICCV.2015.169).
[3] R. Girshick, *et al.*: "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014 IEEE Conference on Computer Vision and Pattern Recognition (2014) 580 (DOI: 10.1109/CVPR.2014.81).
[4] A. Krizhevsky, *et al.*: "ImageNet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems **25** (2012) (DOI: 10.1145/3065386).
[5] L. Yann, *et al.*: "Deep learning," Nature **521** (2015) 436 (DOI: 10.1038/nature14539).
[6] L. Li, *et al.*: "Discretely coding semantic rank orders for supervised image hashing," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 5140 (DOI: 10.1109/CVPR.2017.546).
[7] P. Pinheiro, *et al.*: "Learning to segment object candidates," Proceedings of Advances in Neural Information Processing Systems (2015) 1990 (DOI: 10.48550/arXiv.1506.06204).
[8] C. Szegedy, *et al.*: "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 1 (DOI: 10.1109/CVPR.2015.7298594).
[9] M. Rastergari, *et al.*: "XNOR-Net: ImageNet classification using binary convolutional neural networks," Computer Vision — ECCV (2016) 525 (DOI: 10.1007/978-3-319-46493-0_32).
[10] H. Qing, *et al.*: "Binary neural networks: a survey," Pattern Recognition **1051** (2020) 107281 (DOI: 10.1016/j.patcog.2020.107281).
[11] F. Li, *et al.*: "Ternary weight networks," Computer Vision and Pattern Recognition (2016) arXiv:1605.04711 (DOI: 10.48550/arXiv.1605.04711).
[12] H. Alemdar, *et al.*: "Ternary neural networks for resource-efficient AI applications," 2017 International Joint Conference on Neural Networks (IJCNN) (2017) 2547 (DOI: 10.1109/IJCNN.2017.7966166).
[13] H. Yonekawa, *et al.*: "A ternary weight binary input convolutional neural network: realization on the embedded processor," 2018 IEEE 48th International Symposium on Multiple-Valued Logic (ISMVL)

(2008) (DOI: 10.1109/ISMVL.2018.00038).
[14] D. Wan, *et al.*: "TBN: convolutional neural network with ternary inputs and binary weights," Computer Vision — ECCV 2018 (2018) (DOI: 10.1007/978-3-030-01216-8_20)
[15] K. Hwang and W. Sung: "Fixed-point feedforward deep neural network design using weights +1, 0, and −1," 2014 IEEE Workshop on Signal Processing Systems (SiPS) (2014) 1 (DOI: 10.1109/SiPS.2014.6986082).
[16] R. Andri, *et al.*: "YodaNN: an architecture for ultralow power binary-weight CNN acceleration," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **37** (2018) 48 (DOI: 10.1109/TCAD.2017.2682138).
[17] Z. Jiang, *et al.*: "XNOR-SRAM: in-memory computing SRAM macro for binary/ternary deep neural networks," 2018 IEEE Symposium on VLSI Technology (2018) (DOI: 10.1109/VLSIT.2018.8510687).
[18] J. Song, *et al.*: "15.2 A 28 nm 64 Kb inference-training two-way transpose multibit 6T SRAM compute-in-memory macro for AI edge chips," IEEE International Solid- State Circuits Conference (ISSCC) (2020) 240 (DOI: 10.1109/ISSCC19947.2020.9062949).
[19] H. Liu, *et al.*: "Binary memristive synapse based vector neural network architecture and its application," IEEE Trans. Circuits Syst. II, Exp. Briefs **68** (2021) 772 (DOI: 10.1109/TCSII.2020.3015337).
[20] Z. Li, *et al.*: "Design of ternary neural network with 3-D vertical RRAM array," IEEE Trans. Electron Devices **64** (2017) 2721 (DOI: 10.1109/TED.2017.2697361).
[21] T. Tang, *et al.*: "Binary convolutional neural network on RRAM," 2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC) (2017) 782 (DOI: 10.1109/ASPDAC.2017.7858419).
[22] X. Sun, *et al.*: "Fully parallel RRAM synaptic array for implementing binary neural network with (+1, −1) weights and (+1, 0) neurons," 2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC) (2018) 574 (DOI: 10.1109/ASPDAC.2018.8297384).
[23] X. Fong, *et al.*: "Spin-transfer torque devices for logic and memory: prospects and perspectives," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **35** (2016) 1 (DOI: 10.1109/TCAD.2015.2481793).
[24] M. Natsui, *et al.*: "Design of MTJ-based nonvolatile logic gates for quantized neural networks," Microelectronics Journal **82** (2018) 13 (DOI: 10.1016/j.mejo.2018.10.005).
[25] P. Yu, *et al.*: "A multilevel cell STT-MRAM-based computing in-memory accelerator for binary convolutional neural network," IEEE Trans. Magn. **54** (2018) 1 (DOI: 10.1109/TMAG.2018.2848625).
[26] S. Jain, *et al.*: "Computing in memory with spin-transfer torque magnetic RAM," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **26** (2018) 470 (DOI: 10.1109/TVLSI.2017.2776954).
[27] S. Angizi, *et al.*: "IMCE: energy-efficient bit-wise in-memory convolution engine for deep neural network," 2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC) (2018) 111 (DOI: 10.1109/ASPDAC.2018.8297291).
[28] S. Jung, *et al.*: "A crossbar array of magnetoresistive memory devices for in-memory computing," Nature **601** (2022) 211 (DOI: 10.1038/s41586-021-04196-6).
[29] L. Zhang, *et al.*: "Design and analysis of the reference cells for STT-MRAM," IEICE Electron. Express **10** (2013) 20130352 (DOI: 10.1587/elex.10.20130352).
[30] Y. Zhang, *et al.*: "Compact modeling of perpendicular-anisotropy CoFeB/MgO magnetic tunnel junctions," IEEE Trans. Electron Devices **59** (2012) 819 (DOI: 10.1109/TED.2011.2178416).
[31] Y. Zhang, *et al.*: "STT-RAM cell optimization considering MTJ and CMOS variations," IEEE Trans. Magn. **47** (2011) 2962 (DOI: 10.1109/TMAG.2011.2158810).
[32] U. Zahid, *et al.*: "FAT: training neural networks for reliable inference under hardware faults," 2020 IEEE International Test Conference (ITC) (2020) 1 (DOI: 10.1109/ITC44778.2020.9325249).