SCIENCE CHINA
Information Sciences

• **RESEARCH PAPER** •

A 3D MCAM Architecture based on Flash Memory Enabling Binary Neural Network Computing for Edge AI

Maoying BAI[1], Shuhao WU[1], Hai WANG[1], Hua WANG[1], Yang FENG[1],
Yueran QI[1], Chengcheng WANG[1], Zheng CHAI[2], Tai MIN[2], Jixuan WU[1],
Xuepeng ZHAN[1*] & Jiezhi CHEN[1*]

[1] *School of Information Science and Engineering, Shandong University, Qingdao* 266200*, China;*
[2] *Center for Spintronic and Quantum Systems, State Key Laboratory for Mechanical Behavior of Materials, and School of Materials Science and Engineering, Xi'an Jiaotong University, Xi'an* 710000*, China*

Abstract The in-memory computing (IMC) architecture implemented by non-volatile memory units shows great possibilities to break the traditional Von-Neumann bottleneck. In this paper, a 3D IMC architecture is proposed whose unit is based on a multi-bit content-addressable memory (MCAM). The MCAM unit is comprised of two 65 nm flash memory and two transistors (2Flash2T), which is reconfigurable and multifunctional for both data write/search and XNOR logic operation. Moreover, the MCAM array can also support the population count (POPCOUNT) operation, which can be beneficial for the training and inference process in binary neural network (BNN) computing. Based on the well-known MNIST dataset, the proposed 3D MCAM architecture shows a 98.63% recognition accuracy and a 300% noise-tolerant performance without significant accuracy deterioration. Our findings can provide the potential for developing highly energy-efficient BNN computing for complex artificial intelligence (AI) tasks based on flash-based MCAM units.

Keywords Reconfigurable, multifunction, MCAM, bitwise operation, Binary Neural Network, Edge AI, Flash Memory, IMC

1 Introduction

In the era of big data, the increasing demand for efficient data processing has triggered an urgent challenge in the traditional von Neumann architecture, whose computing units and storage units are separated leading to a memory wall issue. With the capability to perform a storage function and logic function simultaneously, in-memory computing (IMC) is proposed and regarded as a potential candidate to break the von Neumann bottleneck [1,2]. There is a large amount of effort in developing IMC techniques both on hardware implementation and software algorithms [3-7]. At the hardware unit level, there is a lot of nonvolatile memory including resistive random access memory (RRAM), ferroelectric random access memory (FeRAM), magnetoresistive random access memory (MRAM), phase change memory (PCM), and flash memory [8,9]. Owing to the merits of fast response to write/read operation and good adaptability to large arrays, flash memory has attracted extensive attention in the field of artificial intelligence at the edge (Edge AI) including image recognition & classification, object detection & segmentation, natural language processing and so on. [10-12]. Moreover, content-addressable memory (CAM) is one type of hardware unit that can easily search its entire contents in one clock cycle. With the unique feature of distinguishing the mismatching distance, CAM is capable of performing highly parallel and efficient search operations for data-intensive applications like pattern matching [13-16]. Compared to floating-point GPU implementations, Kazemi *et al.* achieve similar accuracies by using flash-based multi bit-CAM (MCAM) with the ImageNet dataset, which significantly reduces energy consumption and operation latency [17]. At the software algorithm level, inspired by the human nervous system, there are considerable numbers

[*] Corresponding author (email: zhanxuepeng@sdu.edu.cn, chen.jiezhi@sdu.edu.cn)

https://engine.scichina.com/doi/10.1007/s11432-023-4019-4

of reports to construct highly-parallel and energy-efficient artificial neural networks (ANNs) including the deep neural network (DNN), convolutional neural network (CNN), binary neural network (BNN), recurrent neural network (RNN), etc. Among them, BNN is a special one whose weights and activations are binary numbers (+1 or -1). Compared to other ANNs, BNN can support XNOR logic and population count (POPCOUNT) operations to supersede the floating-point operations for convolution and multiply-accumulate (MAC), which can significantly reduce the hardware resource [18-20]. Thus, BNN computing has received wide attention in various AI tasks in particular for those data-centered ones [21-24].

Although extensive research has been carried out on the employment of flashed-based CAM in BNN computing, there are still some challenges in reducing the computing hardware resource and realizing the offline training process. Firstly, at the unit level, several MCAM units and their peripheral circuits are required to realize the XNOR operation [25], which may increase memory space and energy costs. Secondly, there are normally two data lines (DL and ($\overline{DL}$)) at the traditional MCAM array for achieving MAC functions [26-28], which hinders the quick change/iterate the memory states and brings obstacles for off-line training. Lastly, with the limited capability to distinguish the voltage in match lines (ML), the CAM arrays are preferred to certain AI tasks, which severely confined their further applications with complex datasets. Therefore, a compact, multifunctional, and reconfigurable flash-based MCAM unit is highly demanded, which may be of great importance for achieving highly energy-efficient and resource-constrained BNN computing with complex tasks.

In this paper, a reconfigurable MCAM unit (2Flash2T) supporting multi-bit stable states is proposed, which enables bitwise operations without additional peripheral devices. With a separated DL and ($\overline{DL}$) strategy, the proposed unit adopts a pair of input voltage vectors, which may be beneficial for achieving offline training in BNN computing. By mapping the partitioned matrixes to the multiple blocks, the proposed 3D MCAM architecture can perform typical AI tasks like image recognition, which shows an accuracy of $\sim$ 98.63% with a noise-tolerant capability (input noise up to $\sim$ 300%) based on the Mixed National Institute of Standards and Technology database (MNIST). Our findings could provide a novel strategy to design 3D MCAM architecture for high energy-efficient, resource-constrained, and robust Edge AI based on the 65 nm flash memory.

2 MCAM Unit and Array Characteristic

Figure1 shows the schematic of the proposed 3D MCAM architecture for binary-valued matrix multiplications, which includes a digital-to-analog converter (DAC), 3D MCAM blocks, analog-to-digital converter (ADC), and adder subtractor (ADDER). The DAC is adopted to transfer the input matrixes to voltage matrixes, which are further transferred to binary matrixes by using ADC. The MCAM block (grey region) composed of several 3D MCAM arrays is used to realize the bitwise operations. The ADDER modules are utilized for integrating the output voltages from ADC leading to an output result. There are two binary-valued input matrixes (Input $\boldsymbol{A}$ and Input $\boldsymbol{B}$) with the output corresponding to the result of matrix multiplication ($\boldsymbol{A}\times\boldsymbol{B}$). The bitwise matrix operations are mainly performed at the MCAM blocks with the help of peripheral modules.

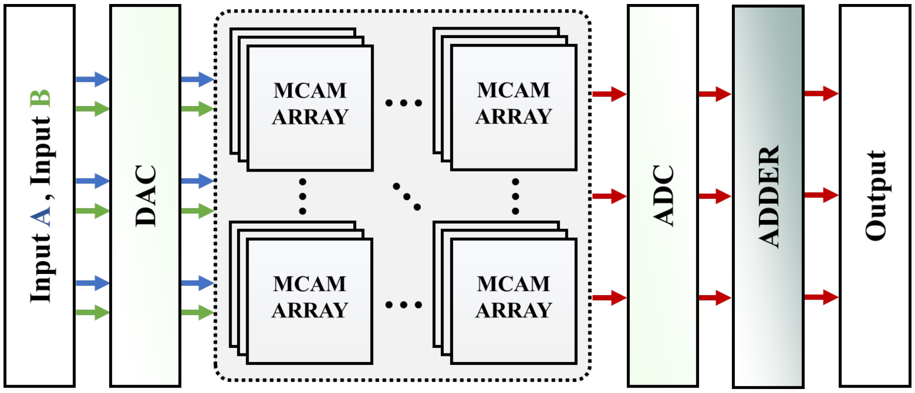

Figure 1 The schematic image of the proposed 3D MCAM architecture with peripheral devices, which supports the binary-valued matrix multiplication.

Table 1 THE SIMULATION SETUP KEY PARAMETERS AND MODELS

	Store -1	Store +1
M_1	$V_{TH-L} < V_{TH-P} < V_{TH-H}$	
M_2		
M_3	V_{TH-L}	V_{TH-H}
M_4	V_{TH-H}	V_{TH-L}
	Search -1	Search +1
DLA	V_{TH-L}	V_{TH-H}
DLB	V_{TH-H}	V_{TH-L}

Figure 2 (a) shows the schematic of the 3D MCAM array, where the blue, green, and red lines stand for two data lines (DLA and DLB) and match line (ML) respectively. The MCAM array contains x×y×z MCAM units (Figure 2(b)), which are all comprised of two 65 nm flash memory and two depletion-type PMOS transistors (2Flash2T). Note that, x is the number of units paralleled in ML, and y is the column number of ML in the 3D array, and z is the layer number. Normally, two complementary DLs (DL and ($\overline{DL}$)) are used for single-datum operations in traditional CAM units. In our proposed MCAM unit, a separated DLs (DLA, DLB) strategy is adopted to represent 2 different data, which is achieved by the PMOS transistors to break the complementary constraint between two data lines. Note that in Figure 2, under the condition that apply a high voltage on the gate of PMOS and flash, the PMOS won't get broken. The reason is that the range of the programming and threshold voltage of flash memory is strictly controlled to fit the working voltage range of the PMOS. The line connecting the point between M_1 and M_3 to the point between M_2 and M_4 is necessary to ensure to CAM unit be able to work especially when searching "-1" of storing "+1" and searching "+1" of storing "-1". Moreover, the setup parameters of the SPICE simulation are shown in Table 1 where $V_{TH-L} = 0.1V$, $V_{TH-P} = 0.5V$, $V_{TH-H} = 0.9V$ and the discharge capacitor connected between ML and the grounded is $0.1pF$.

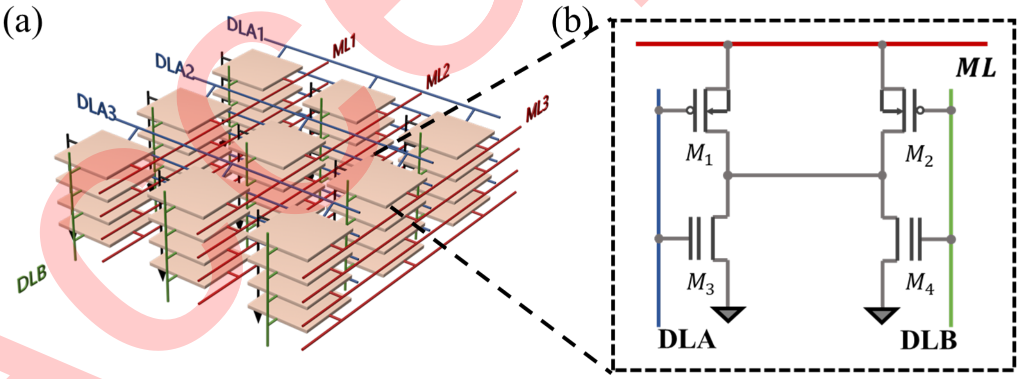

Figure 2 The schematic image of (a) the 3D structure with the flash-based MCAM units; (b) the MCAM unit comprised of two depletion-type PMOS transistors (M_1 and M_2) and two flash memories (M_3 and M_4).

Figure 3(a) displays the threshold voltage of individual flash memory and PMOS transistor simulated in SPICE. The 65nm flash memory shows 16 storage states with threshold voltages (from V_{TH0} to V_{TH15}) ranging from 0.3V to 4.8V. By adjusting the threshold voltages of flash memory to fit the threshold voltages of the PMOS (V_{TH-P}), the proposed unit can behave like typical MCAM and multifunctional MCAM. With V_{TH-P} larger than V_{TH15} (blue region), the proposed unit corresponds to a typical CAM unit with 16 different memory states. When V_{TH-P} is set within V_{TH0} and V_{TH15} (grey region), the proposed unit can also support the data write/search function with a reduced memory state. With V_{TH-P} smaller than V_{TH0} (green region), the proposed unit is always off (disabled). The 16 different transfer curves of the flash memory are shown in Figure 3 (b) corresponding to state-0 ($V_{TH0} = 0.3V$) to state-15 ($V_{TH15} = 4.8V$). When mapping a matrix multiplication, the voltages on DLAs and DLBs represent the row vectors and column vectors in 2D scale respectively. In 3D scale, the different DLAs and DLBs are mapped into different 2D planes. The benefits of the 3D structure lie in that the matrixes only required to be mapped once to DLAs and DLBs. Owing to the separated DLs design, the mapping efficiency is improved, which is suitable for direct voltage iterating.

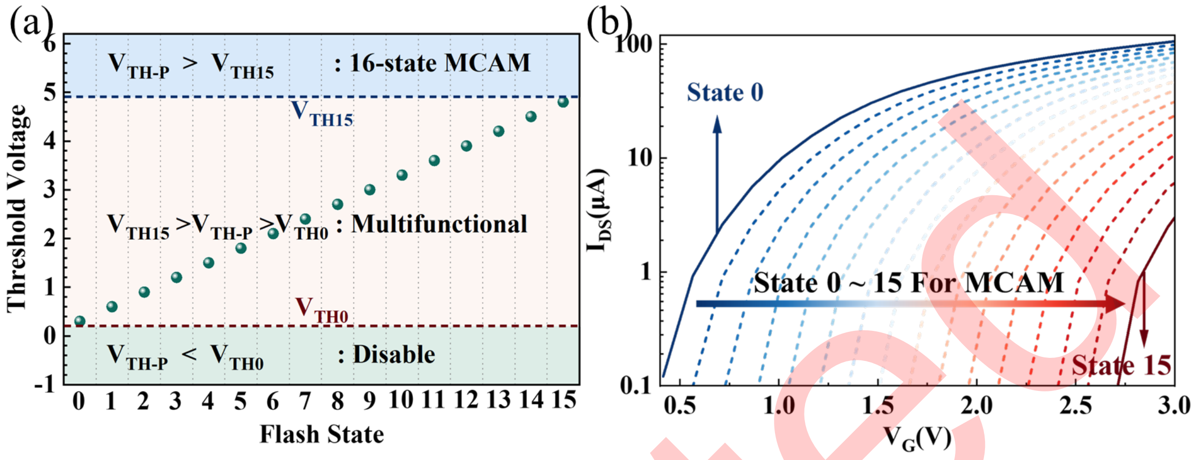

Figure 3 (a) The threshold voltage distributions of a single flash and single PMOS; (b)The transfer curves of the flash memory with 16 different states corresponding to state-0 to state-15.

Table 2 THRESHOLD VOLTAGE ENCODING OF FLASH CORRESPONDING TO DIFFERENT FUNCTIONS OF THE MCAM UNIT

M_1&M_2	$V_{TH-L} < V_{TH-P} < V_{TH-H}$			
Task	Case1	Case2	Case3	Case4
M_3	V_{TH-L}	V_{TH-L}	V_{TH-H}	V_{TH-H}
M_4	V_{TH-L}	V_{TH-H}	V_{TH-L}	V_{TH-H}
Truth Table	(-1,-1)=Match (-1,+1)=Mismatch (+1,-1)=Mismatch (-1,-1)=Match	(-1,-1)=Match (-1,+1)=Match (+1,-1)=Mismatch (-1,-1)=Match	(-1,-1)=Match (-1,+1)=Mismatch (+1,-1)=Match (-1,-1)=Match	(-1,-1)=Match (-1,+1)=Match (+1,-1)=Match (-1,-1)=Match

The PVT (process-voltage-temperature) analysis is conducted to reveal their impacts on the function of CAM unit. The smaller discharge time (the time of V_{ML} reach back to its saturated and stabilized value under matching condition) and larger voltage margin (the difference between the stable voltage of match and mismatch) can be obtained by using FF process corner, higher pre-charge voltage, and lower temperature [37-41], and larger voltage margin ratio can be obtained by lower pre-charge voltage. Compared to the process corner, the temperature and pre-charge voltage are dominated factors that affects the discharging time and voltage margin. Since the fabricating process is important factor on affecting the flash V_t distribution [42-45], a V_t shift ranging from -0.2V to 0.2V is adopted to reveal the impacts on the proposed CAM unit functionality in the SPICE simulation. The voltage margin decreases as the V_t shift increases. To guarantee a reliable function, the process variation induced V_t shift should be suppressed within 0.2V for the proposed unit.

A. The multifunctional MCAM Unit for write/search and XNOR logic functions

For the write/search function, the PMOS transistors always keep ON with $V_{TH-P} > V_{TH15}$, making the proposed MCAM unit behave like a typical MCAM unit with 16 memory states. Thus, the encoding scheme of the flash (M_3 and M_4) threshold voltage is the same as the traditional method.

For the XNOR logic operation, V_{TH-P} is required to be set to a certain range (the grey region in Fig. 3(a)). The two binary data are transferred to input voltages imposed to DLA and DLB via DAC modules, which should be equal to the memory states (V_{TH0} to V_{TH15}). The lower input voltage is denoted as V_{TH-L} and the higher one is denoted as V_{TH-H} corresponding to logic -1 and logic +1, respectively. Note that the V_{TH-P} should be larger than V_{TH-L} and smaller than V_{TH-H} ($V_{TH-L} < V_{TH-P} < V_{TH-H}$).

As summarized in Table 2, different threshold voltages of M_3 and M_4 could lead to four different functions (Cases 1 to 4). Only when the two flash are both programmed to V_{TH-L}, the XNOR function can be achieved. If V_{DLA} equals V_{DLB}, the MCAM unit could give rise to a match condition. Otherwise, a mismatch result is obtained, which corresponds to the XNOR logic function (Case 1). In the other cases (Case 2-4), the truth tables are also displayed, which can be further adopted based on the user's requirement. For example, Case4 can be applied as wildcards to uniform the length when the data length is inconsistent. Moreover, with the separated DLs strategy, the input information of DLA and DLB (V_{DLA} and V_{DLB}) do not need to be saved in the MCAM unit, which is beneficial for rapid change.

B. MCAM array characteristics for POPCOUNT functions

For the POPCOUNT logic operation, the MCAM array contains $x \times y \times z$ units that are adopted to count the number of mismatch conditions ((A, B) corresponds to (-1, +1) and (+1, -1)). On the array level, x is limited by the saturated voltage on ML (V_{ML}) and its distinguish margin. Thus, in this work, 16 MCAM units are adopted to ensure the mismatched conditions can be clearly distinguished. Figure 4 (a) shows the relationship between discharge time and V_{ML}, which falls rapidly at first 0.5μs owing to the disturbance from the PMOS. If all 16 units are matched, the V_{ML} will recover to the recharging voltage, while the V_{ML} increases to various saturated voltages (smaller than the recharging voltage) corresponding to various mismatch numbers. With a discharge time of 6μs, the relationship between mismatching numbers versus the V_{ML} is displayed in Figure 4 (b), which can be fitted as the following equation

$$num = ae^{bv} + c \tag{1}$$

where a, b, and c are all constant.

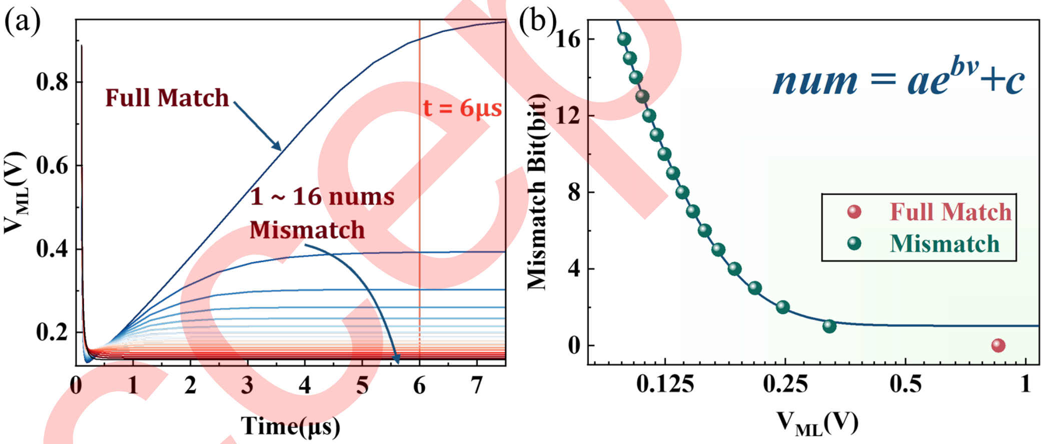

Figure 4 (a) The discharge time-dependent distribution of ML current with various mismatching numbers; (b) The relationship of mismatching numbers versus the voltage of ML (V_{ML}) at the discharge time of 6μs.

C. MCAM architecture for binary matrix multiplication function

To perform binary matrix multiplication, periphery devices are further required in dealing with the POPCOUNT results.

Figure 5 shows the flow chart of the operating process for binary matrixes (A and A) multiplication, where A_1/B_1stands for the first row/column vector. The number of XNOR results -1/+1 is assumed as m/n, which corresponds to the mismatch/match condition on ML. The length of the vector (A_1 and B_1) is denoted as k corresponding to the unit number on a single ML. The summary of mismatch conditions (m) and match conditions (n) equals (k):

$$m + n = k \tag{2}$$

After the XNOR logic processing, the result of $A_1 \times B_1$ corresponds to:

$$(+1) \times (k - m) + (-1) \times m \tag{3}$$

which can be simplified to:

$$k - 2m \tag{4}$$

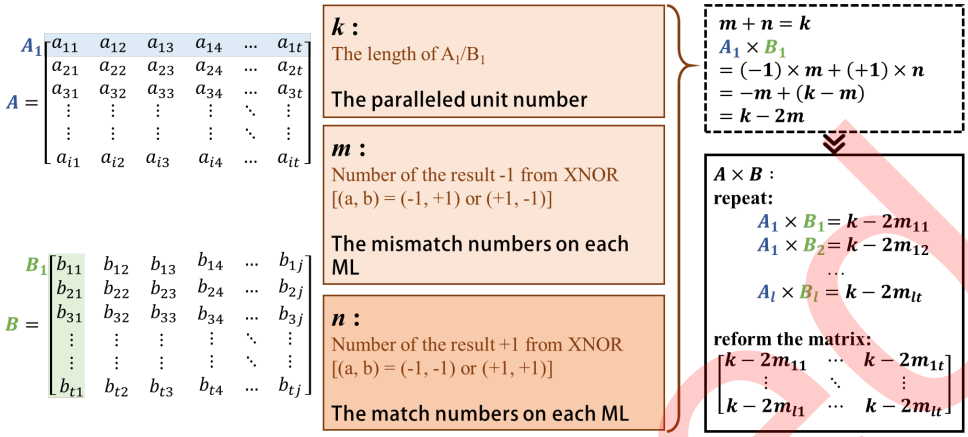
A_1
$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & \cdots & a_{1t} \\ a_{21} & a_{22} & a_{23} & a_{24} & \cdots & a_{2t} \\ a_{31} & a_{32} & a_{33} & a_{34} & \cdots & a_{3t} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & a_{i3} & a_{i4} & \cdots & a_{it} \end{bmatrix}$
B_1
$B = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & \cdots & b_{1j} \\ b_{21} & b_{22} & b_{23} & b_{24} & \cdots & b_{2j} \\ b_{31} & b_{32} & b_{33} & b_{34} & \cdots & b_{3j} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{t1} & b_{t2} & b_{t3} & b_{t4} & \cdots & b_{tj} \end{bmatrix}$
k :
The length of A_1/B_1
The paralleled unit number
m :
Number of the result -1 from XNOR
[(a, b) = (-1, +1) or (+1, -1)]
The mismatch numbers on each ML
n :
Number of the result +1 from XNOR
[(a, b) = (-1, -1) or (+1, +1)]
The match numbers on each ML
$m + n = k$
$A_1 \times B_1$
$= (-1) \times m + (+1) \times n$
$= -m + (k - m)$
$= k - 2m$
$A \times B$:
repeat:
$A_1 \times B_1 = k - 2m_{11}$
$A_1 \times B_2 = k - 2m_{12}$
…
$A_l \times B_l = k - 2m_{lt}$
reform the matrix:
$\begin{bmatrix} k-2m_{11} & \cdots & k-2m_{1t} \\ \vdots & \ddots & \vdots \\ k-2m_{l1} & \cdots & k-2m_{lt} \end{bmatrix}$

Figure 5 The flow chart of the multiplication calculation process of $i \times t$ binary matrix A and $t \times j$ binary matrix B.

With the process repeated several times, the output of $A \times B$ can be obtained as illustrated in Figure 5. In this way, both the inputs for the matrix calculation can be applied to the circuit with the form of voltage, making it possible for inputs rapid iteration.

3 3D MCAM Array Implement BNN Computing

A. The partitioned mapping for matrix multiplications and convolution operations
For typical AI tasks requiring large-scale matrix operations, a matrix partition method is adopted by using the proposed MCAM blocks. Figure 6 (a) shows the schematic image for large-scale matrix multiplications. A and B are binary matrixes, which are segmented into smaller ones (A_{11} to A_{2p} and B_{11} to B_{p2}) that can be directly mapped to the MCAM arrays. The multiplication results of large matrixes can be obtained by successively integrating the partitioned matrixes.
Figure 6 (b) shows the scheme of convolution operations with a 4×4 kernel (Input B). The input matrix (Input B) is unrolled step by step as the kernel slides upon it, leading to a recombinant matrix denoted as Input A'. The convolution kernel is unrolled to a vector denoted as Input B'. Then, the convolution results can be obtained by applying A' and B' into matrix multiplications.

B. The network performances for BNN computing
To simplify the hardware implementation process, the BNN structure employed in this work is constructed by two convolutional layers and two fully connected layers. The first convolutional layer and the last fully-connected layer are performed with float point operations, while the others are binary. By using the well-known MNIST dataset, the network performances of BNN computing based on the proposed 3D MCAM architecture are evaluated. The recognition accuracy is mainly affected by two factors which are hyper-parameter setting and hardware implementation. The hyper-parameter values correspond to different matrix scales in data flowing of BNN computing, which directly determines the array size. The training of the proposed CAM based BNN is achieved by using Bin_LeNet (the binarized LeNet) model. Several hyperparameters (the number of layers, convolutional kernel size, the i/o channel sizes) are adjusted to improve the network efficiency and accuracy. During training, there are some compressions like the convolution and pool operations for feature extraction, while no extra compression operations in the hardware simulation to maximize the accuracy.
Figure 7(a) shows the accuracies with different $x \times y \times z$ arrays (constant x & z of 16 & 64). Note that x is the number of units paralleled in ML, y is the column number of ML in the array, and z is the layer number of the output channel for *cnov2*. The accuracy increases slightly from 94.51% to 98.91% as y increases from 4 to 81. For different hardware implementation conditions, Figure 7(b) shows the accuracies with the mapped layer implemented by *cnov2*, only *fc1*, as well as both *cnov2* and *fc1* at various array sizes. It is clear that the hardware implementation has limited impacts on the accuracy. Further experimental results show that the circuit error has little effect on the training accuracy of large-scale neural networks. For more complex datasets, large array size is normally required, which brings challenges in

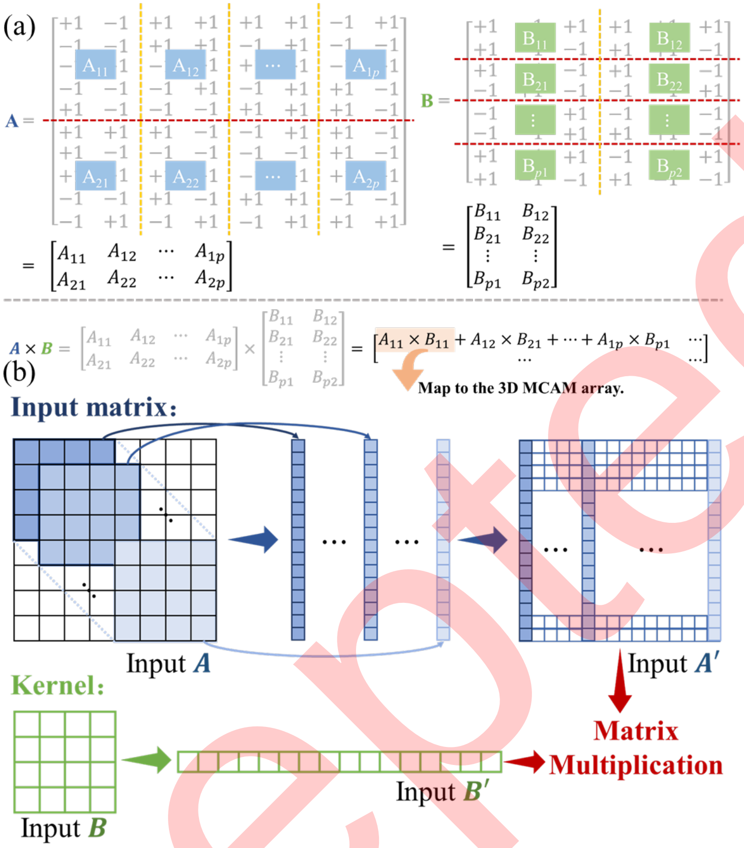
(a)
A =
A11
A12
A1p
A21
A22
A2p
= [A11 A12 ⋯ A1p; A21 A22 ⋯ A2p]
B =
B11
B12
B21
B22
Bp1
Bp2
= [B11 B12; B21 B22; ⋮ ⋮; Bp1 Bp2]
A × B = [A11 A12 ⋯ A1p; A21 A22 ⋯ A2p] × [B11 B12; B21 B22; ⋮ ⋮; Bp1 Bp2] = [A11 × B11 + A12 × B21 + ⋯ + A1p × Bp1 ⋯; ⋯ ⋯]
Map to the 3D MCAM array.
(b)
Input matrix:
Input A
Input A′
Kernel:
Input B
Input B′
Matrix Multiplication

Figure 6 The schematic image of (a) matrix partitioning scheme for large-scale matrix multiplications; (b) converting convolution operations to matrix multiplications.

increasing the number of CAM units paralleled in one ML and of MLs paralleled in one plane. This may be achieved by suppressing the device variation and enlarging the voltage margin. Additionally, a dynamically matrix partition strategy is also important, which can adjust the matrix size based on different data complexities and required accuracies. The proposed architecture can be applied on some more complex dataset. For example, the classification accuracy is $\sim$ 78.09% by using Bin_VGG13 model in CIFAR10 dataset. The network performances can be optimized by using complex neural network model (Bin_VGG16, Bin_VGG19) [29,30], enlarging the array size and improving the CAM unit performance. Moreover, the noise immunity is verified based on the MCAM block implemented BNN computing. Figure 8 shows the accuracy (left axis) and error (right axis) under different noise rate (α), which is a coefficient of the imposed noise matrix. The noise matrix is comprised of random values (0 1) with the same size as the MNIST images which have pixel values ranging from 0 to 1 originally. It is clear that the accuracy decreases slightly until the noise rate α approaches 3.5, which indicates a strong noise-tolerant capability of the proposed MCAM-based BNN computing.

C. Comparison with other CAM units and peripheral circuit
Table 3 shows the comparison of different CAM units' performance with our work. The energy efficiency is defined as the average energy consumption per search for each unit, which is about $0.18fJ/bit/search$ in our work. The latency is defined as search delay between rising edges of the clock and V_{ML}, which corresponds to the time interval of pre-charge stage and match stage [31]. The area is evaluated by the device structures & number and process node, considering the device dimensions are unknown in the SPICE simulation. Compared to other related work, the proposed CAM shows advantages in terms of average energy consumption, reconfigurable characteristics, multi-state storage and bitwise operation. The proposed architecture shows advantages in the multi-functional characteristics, high-integrated array and potentials for off-line training. The architecture can accomplish the typical 16-state MCAM function and bitwise operation of XNOR calculations with energy consumption of $0.18fJ/bit/search$.

https://engine.scichina.com/doi/10.1007/s11432-023-4019-4

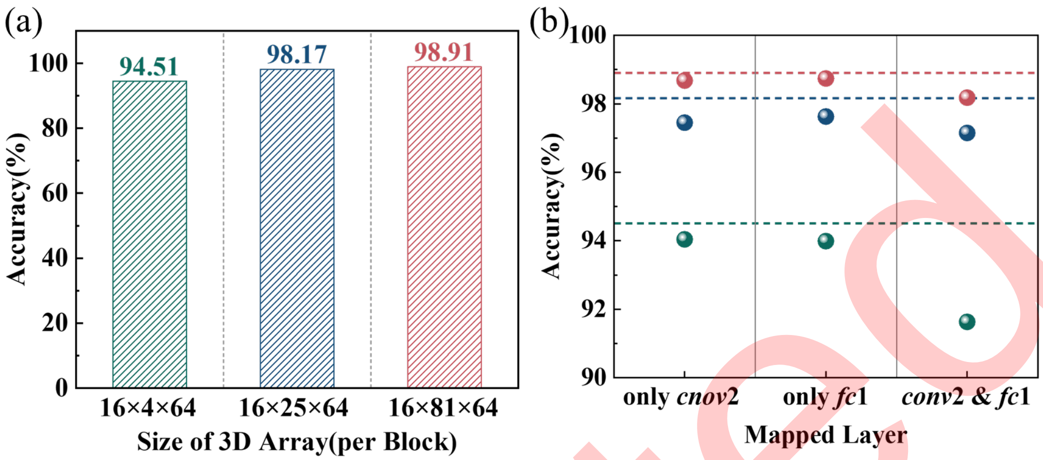

Figure 7 (a) The different accuracies with various 3D array sizes in the form of $x \times y \times z$; (b) The accuracies versus mapped layer implemented by only *cnov2* (first column), only *fc1* (second column), as well as both *cnov2* and *fc1* (third column).

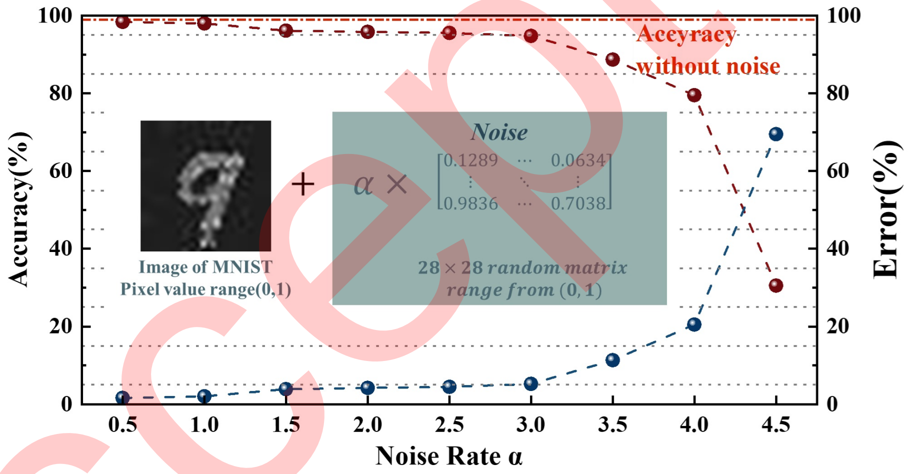

Figure 8 The recognition accuracy (left) and error (right) with noise disturbance, where the noise is 28 × 28 random matrix between 0 ∼ 1 and the rate α is a coefficient.

Benefitting from the reliable characteristics and mature process of flash memory, the proposed CAM unit is easy for large-scale and 3D integration. The design of two different DLs may provide the possibility for accomplishing off-line training.

4 Conclusion

In this work, a reconfigurable MCAM unit (2Flash2T) with 14 stable states is proposed to realize the XNOR and POPCOUNT operations. This allows the input values applied on the reconfigurable unit to realize a 14-state search function as a traditional MCAM unit does. Based on the block matrix multiplication scheme, the proposed 3D MCAM array is capable of disposing of the well-known MNIST dataset and receives a recognition accuracy of up to ∼98.63% and a noise-tolerant capability with ∼300% input noise without apparent accuracy drop. A novel design of 3D MCAM architecture based on the 65 nm flash memory for high energy-efficient, resource-constrained, and robust Edge AI tasks is provided by the findings of our work.

https://engine.scichina.com/doi/10.1007/s11432-023-4019-4

Acknowledgements This work was supported by National Key Research and Development Program of China (2023YFB02500, 2023YFB4402400), National Natural Science Foundation of China (Nos. 62034006, 92264201, U23B2040), Natural Science Foundation of Shandong Province (Nos. ZR2023LZH007, TSQN202306059) and Program of Qilu Young Scholars of Shandong University.

Supporting information Appendix A. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 D. Ielmini and H.-S. P. Wong, “In-memory computing with resistive switching devices,” Nat Electron, vol. 1, no. 6, pp. 333–343, Jun. 2018, doi: 10.1038/s41928-018-0092-2.E. P. Wigner, “Theory of traveling-wave optical laser,” Phys. Rev., vol. 134, pp. A635–A646, Dec. 1965.

2 Y. Li, Z. Wang, R. Midya, Q. Xia, and J. J. Yang, “Review of memristor devices in neuromorphic computing: materials sciences and device challenges,” J. Phys. D: Appl. Phys., vol. 51, no. 50, p. 503002, Dec. 2018, doi: 10.1088/1361-6463/aade3f.

3 M. A. Zidan, J. P. Strachan, and W. D. Lu, “The future of electronics based on memristive systems,” Nat Electron, vol. 1, no. 1, pp. 22–29, Jan. 2018, doi: 10.1038/s41928-017-0006-8.

4 C. Li et al., “Long short-term memory networks in memristor crossbar arrays,” Nat Mach Intell, vol. 1, no. 1, pp. 49–57, Jan. 2019, doi: 10.1038/s42256-018-0001-4.

5 A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, “Memory devices and applications for in-memory computing,” Nat. Nanotechnol., vol. 15, no. 7, pp. 529–544, Jul. 2020, doi: 10.1038/s41565-020-0655-z.

6 Y. Feng et al., “Near-threshold-voltage operation in flash-based high-precision computing-in-memory to implement Poisson image editing,” Sci. China Inf. Sci., vol. 66, no. 12, p. 222402, Dec. 2023, doi: 10.1007/s11432-022-3743-x.

7 X. Zhan, J. Chen, and Z. Ji, “Insights of VG-dependent threshold voltage fluctuations from dual-point random telegraph noise characterization in nanoscale transistors,” Sci. China Inf. Sci., vol. 65, no. 8, p. 189405, Aug. 2022, doi: 10.1007/s11432-021-3330-8.

8 S. Yu, H. Jiang, S. Huang, X. Peng, and A. Lu, “Compute-in-Memory Chips for Deep Learning: Recent Trends and Prospects,” IEEE Circuits Syst. Mag., vol. 21, no. 3, pp. 31–56, 2021, doi: 10.1109/MCAS.2021.3092533.

9 S. Jeloka, N. Bharathwaj Akesh, D. Sylvester, and D. Blaauw, “A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T Bit Cell Enabling Logic-in-Memory,” IEEE Journal of Solid-State Circuits, vol. 51, no. 4, pp. 1009–1021, Apr. 2016, doi: https://doi.org/10.1109/jssc.2016.2515510.

10 H.-T. Lue et al., “Optimal Design Methods to Transform 3D NAND Flash into a High-Density, High-Bandwidth and Low-Power Nonvolatile Computing in Memory (nvCIM) Accelerator for Deep-Learning Neural Networks (DNN),” in 2019 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, USA: IEEE, Dec. 2019, p. 38.1.1-38.1.4. doi: 10.1109/IEDM19573.2019.8993652.

11 Y. Xiang et al., “Efficient and Robust Spike-Driven Deep Convolutional Neural Networks Based on NOR Flash Computing Array,” IEEE Trans. Electron Devices, vol. 67, no. 6, pp. 2329–2335, Jun. 2020, doi: 10.1109/TED.2020.2987439.

12 R. Han et al., “A Novel Convolution Computing Paradigm Based on NOR Flash Array With High Computing Speed and Energy Efficiency,” IEEE Trans. Circuits Syst. I, vol. 66, no. 5, pp. 1692–1703, May 2019, doi: 10.1109/TCSI.2018.2885574.

13 B. Kwak, H. Kim, and D. Kwon, “Ferroelectric-gate tunnel field-effect transistor one-transistor ternary contents addressable memory,” Semicond. Sci. Technol., vol. 38, no. 5, p. 055013, May 2023, doi: 10.1088/1361-6641/ac7d03.

14 Y. Chen, J. Mu, H. Kim, L. Lu, and T. T.-H. Kim, “BP-SCIM: A Reconfigurable 8T SRAM Macro for Bit-Parallel Searching and Computing In-Memory,” IEEE Trans. Circuits Syst. I, vol. 70, no. 5, pp. 2016–2027, May 2023, doi: 10.1109/TCSI.2023.3240303.

15 Z. Jianwei, Y. Yizheng, L. Binda, L. Jinbao, A cascaded charge-sharing technique for an EDP-efficient match-line design in CAMs, J. Semicond., 30, 6, 065009, 6 2009, doi: 10.1088/1674-4926/30/6/065009.

16 X. Yin et al., “Ferroelectric Ternary Content Addressable Memories for Energy-Efficient Associative Search,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 42, no. 4, pp. 1099–1112, Apr. 2023, doi: 10.1109/TCAD.2022.3197694..

17 C. Zhuo et al., “Design of Ultracompact Content Addressable Memory Exploiting 1T-1MTJ Cell,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 42, no. 5, pp. 1450–1462, May 2023, doi: 10.1109/TCAD.2022.3204515.

18 A. Kazemi, S. Sahay, A. Saxena, M. M. Sharifi, M. Niemier, and X. S. Hu, “A Flash-Based Multi-Bit Content-Addressable Memory with Euclidean Squared Distance,” in 2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), Boston, MA, USA: IEEE, Jul. 2021, pp. 1–6. doi: 10.1109/ISLPED52811.2021.9502488.

19 M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, “XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks.” arXiv, Aug. 02, 2016. Accessed: Mar. 14, 2023. [Online]. Available: http://arxiv.org/abs/1603.05279

20 M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, “Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1.” arXiv, Mar. 17, 2016. Accessed: Apr. 15, 2023. [Online]. Available: http://arxiv.org/abs/1602.02830

21 L. Deng, P. Jiao, J. Pei, Z. Wu, and G. Li, “GXNOR-Net: Training deep neural networks with ternary weights and activations without full-precision memory under a unified discretization framework,” Neural Networks, vol. 100, pp. 49–58, Apr. 2018, doi: 10.1016/j.neunet.2018.01.010.

22 X. Si et al., “A Dual-Split 6T SRAM-Based Computing-in-Memory Unit-Macro With Fully Parallel Product-Sum Operation for Binarized DNN Edge Processors,” IEEE Trans. Circuits Syst. I, vol. 66, no. 11, pp. 4172–4185, Nov. 2019, doi: 10.1109/TCSI.2019.2928043.

23 S. Yin, Z. Jiang, J.-S. Seo, and M. Seok, “XNOR-SRAM: In-Memory Computing SRAM Macro for Binary/Ternary Deep Neural Networks,” IEEE J. Solid-State Circuits, pp. 1–11, 2020, doi: 10.1109/JSSC.2019.2963616.

24 H. Qiu et al., “RBNN: Memory-Efficient Reconfigurable Deep Binary Neural Network With IP Protection for Internet of Things,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 42, no. 4, pp. 1185–1198, Apr. 2023, doi: 10.1109/TCAD.2022.3197499.

25 Y. Halawani, B. Mohammad, M. Abu Lebdeh, M. Al-Qutayri, and S. F. Al-Sarawi, “ReRAM-Based In-Memory Computing for Search Engine and Neural Network Applications,” IEEE J. Emerg. Sel. Topics Circuits Syst., vol. 9, no. 2, pp. 388–397, Jun. 2019, doi: 10.1109/JETCAS.2019.2909317.

26 Y. Chen, L. Lu, B. Kim, and T. T.-H. Kim, “Reconfigurable 2T2R ReRAM with Split Word-Lines for TCAM Operation and In-Memory Computing,” in 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain: IEEE, Oct. 2020, pp. 1–5. doi: 10.1109/ISCAS45731.2020.9180665.

27 A. F. Laguna, X. Yin, D. Reis, M. Niemier, and X. S. Hu, “Ferroelectric FET Based In-Memory Computing for Few-Shot Learning,” in Proceedings of the 2019 on Great Lakes Symposium on VLSI, Tysons Corner VA USA: ACM, May 2019, pp. 373–378. doi: 10.1145/3299874.3319450.

28 X. Wang et al., “A 4T2R RRAM Bit Cell for Highly Parallel Ternary Content Addressable Memory,” IEEE Trans. Electron Devices, vol. 68, no. 10, pp. 4933–4937, Oct. 2021, doi: 10.1109/TED.2021.3107497.

29 G. S. Nugraha, M. I. Darmawan, and R. Dwiyansaputra, “Comparison of CNN’s Architecture GoogleNet, AlexNet, VGG-16, Lenet -5, Resnet-50 in Arabic Handwriting Pattern Recognition,” KINETIK, May 2023, doi: 10.22219/kinetik.v8i2.1667.

30 S. Yin, Z. Jiang, J.-S. Seo, and M. Seok, “XNOR-SRAM: In-Memory Computing SRAM Macro for Binary/Ternary Deep Neural Networks,” IEEE J. Solid-State Circuits, pp. 1–11, 2020, doi: 10.1109/JSSC.2019.2963616.

31 K. Pan, A. M. S. Tosson, N. Wang, N. Y. Zhou, and L. Wei, “A Novel Cascadable TCAM Using RRAM and Current Race Scheme for High-Speed Energy-Efficient Applications,” IEEE Trans. Nanotechnology, vol. 22, pp. 214–221, 2023, doi: 10.1109/TNANO.2023.3271308.

32 C.-C. Lin et al., “7.4 A 256b-wordlength ReRAM-based TCAM with 1ns search-time and 14× improvement in wordlength-energyefficiency-density product using 2.5T1R cell,” in 2016 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA: IEEE, Jan. 2016, pp. 136–137. doi: 10.1109/ISSCC.2016.7417944.

33 X. Yin et al., “Ferroelectric Ternary Content Addressable Memories for Energy-Efficient Associative Search,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 42, no. 4, pp. 1099–1112, Apr. 2023, doi: 10.1109/TCAD.2022.3197694.

34 S. Cho, S. Kim, I. Choi, M. Kang, S. Baik, and J. Jeon, “Non-volatile logic-in-memory ternary content addressable memory circuit with floating gate field effect transistor,” AIP Advances, vol. 13, no. 4, p. 045211, Apr. 2023, doi: 10.1063/5.0141131.

35 A. Kazemi, S. Sahay, A. Saxena, M. M. Sharifi, M. Niemier, and X. S. Hu, “A Flash-Based Multi-Bit Content-Addressable Memory with Euclidean Squared Distance,” in 2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), Boston, MA, USA: IEEE, Jul. 2021, pp. 1–6. doi: 10.1109/ISLPED52811.2021.9502488.

36 S. Yin, Z. Jiang, J.-S. Seo, and M. Seok, “XNOR-SRAM: In-Memory Computing SRAM Macro for Binary/Ternary Deep Neural Networks,” IEEE J. Solid-State Circuits, pp. 1–11, 2020, doi: 10.1109/JSSC.2019.2963616.

37 C. J. Sagario, B. Q. Iii, K. G. Jimenez, I. B. Escabal, A. C. Lowaton, and J. A. Hora, “Design of Single Poly Flash Memory Cell with Power Reduction Technique at Program Mode in 65nm CMOS Process,” International Conference on Control, 2018.

38 D. Resnati, A. Goda, G. Nicosia, C. Miccoli, A. S. Spinelli, and C. Monzio Compagnoni, “Temperature Effects in NAND Flash Memories: A Comparison Between 2-D and 3-D Arrays,” IEEE Electron Device Lett., vol. 38, no. 4, pp. 461–464, Apr. 2017, doi: 10.1109/LED.2017.2675160.

39 W. Lee, C. Park, and K. Kim, “Temperature Dependence of Endurance Characteristics in NOR Flash Memory Cells,” in 2006 IEEE International Reliability Physics Symposium Proceedings, San Jose, CA, USA: IEEE, 2006, pp. 701–702. doi: 10.1109/RELPHY.2006.251332.

40 J. Wang et al., “Reconfigurable Bit-Serial Operation Using Toggle SOT-MRAM for High-Performance Computing in Memory Architecture,” IEEE Trans. Circuits Syst. I, vol. 69, no. 11, pp. 4535–4545, Nov. 2022, doi: 10.1109/TCSI.2022.3192165.

41 Ali, Mustafa F., A. Jaiswal, and K. Roy. ”In-Memory Low-Cost Bit-Serial Addition Using Commodity DRAM Technology.” Circuits and Systems I: Regular Papers, IEEE Transactions on PP.99(2019):1-11.

42 H. An, K. Kim, S. Jung, H. Yang, K. Kim, and Y. Song, “The threshold voltage fluctuation of one memory cell for the scaling-down NOR flash,” in 2010 2nd IEEE InternationalConference on Network Infrastructure and Digital Content, Beijing, China: IEEE, Sep. 2010, pp. 433–436. doi: 10.1109/ICNIDC.2010.5657806.

43 H. Li, “Modeling of Threshold Voltage Distribution in NAND Flash Memory: A Monte Carlo Method,” IEEE Trans. Electron Devices, vol. 63, no. 9, pp. 3527–3532, Sep. 2016, doi: 10.1109/TED.2016.2593913.

44 W. Kim, Y. Kim, S. H. Park, J. Y. Seo, D. B. Kim, and B.-G. Park, “Variation of Threshold Voltage and ON-Cell Current Caused by Cell Gate Length Fluctuation in Virtual Source/Drain NAND Flash Memory,” Jpn. J. Appl. Phys., vol. 51, no. 7R, p. 074301, Jul. 2012, doi: 10.1143/JJAP.51.074301.

45 T. Yang, Z. Xia, D. Shi, Y. Ouyang, and Z. Huo, “Analysis and Optimization of Threshold Voltage Variability by Polysilicon Grain Size Simulation in 3D NAND Flash Memory,” IEEE J. Electron Devices Soc., vol. 8, pp. 140–144, 2020, doi: 10.1109/JEDS.2020.2970450.