ARTICLES

Journal of Semiconductors
(2023) 44, 054101
doi: 10.1088/1674-4926/44/5/054101

Flash-based in-memory computing for stochastic computing in image edge detection

Zhaohui Sun[1], Yang Feng[1], Peng Guo[2], Zheng Dong[1], Junyu Zhang[3], Jing Liu[4], Xuepeng Zhan[1], Jixuan Wu[1], and Jiezhi Chen[1, †]

[1]School of Information Science and Engineering (ISE), Shandong University, Qingdao 266000, China

[2]Shandong Sinochip Semiconductors Co. Ltd, Jinan 250101, China

[3]Neumem Co., Ltd, Hefei 230088, China

[4]Key Laboratory of Microelectronic Devices and Integrated Technology, Institute of Microelectronics of Chinese Academy of Sciences, Beijing 100029, China

Abstract: The "memory wall" of traditional von Neumann computing systems severely restricts the efficiency of data-intensive task execution, while in-memory computing (IMC) architecture is a promising approach to breaking the bottleneck. Although variations and instability in ultra-scaled memory cells seriously degrade the calculation accuracy in IMC architectures, stochastic computing (SC) can compensate for these shortcomings due to its low sensitivity to cell disturbances. Furthermore, massive parallel computing can be processed to improve the speed and efficiency of the system. In this paper, by designing logic functions in NOR flash arrays, SC in IMC for the image edge detection is realized, demonstrating ultra-low computational complexity and power consumption (25.5 fJ/pixel at 2-bit sequence length). More impressively, the noise immunity is 6 times higher than that of the traditional binary method, showing good tolerances to cell variation and reliability degradation when implementing massive parallel computation in the array.

Key words: in-memory computing; stochastic computing; NOR flash memory; image edge detection

Citation: Z H Sun, Y Feng, P Guo, Z Dong, J Y Zhang, J Liu, X P Zhan, J X Wu, and J Z Chen, Flash-based in-memory computing for stochastic computing in image edge detection[J]. *J. Semicond.*, 2023, 44(5), 054101. https://doi.org/10.1088/1674-4926/44/5/054101

1. Introduction

In traditional von Neumann architecture, data is frequently transferred between the processing unit and the memory unit, which causes significant power consumption and seriously limits processing efficiency. In-memory computing (IMC), wherein memory units designed with the capability to store data and execute computational tasks simultaneously, has been proposed as a promising approach to solve this issue and break the "memory wall". Recently, many impressive IMCs have been reported by using commercial memories (SRAM, DRAM, flash) and emerging memories (ReRAM, PRAM, etc.)[1–3]. For data-intensive computing tasks, it is necessary to construct a large memory array for large-matrix computation, which means that the current of each memory cell has to be low so that the accumulated currents will not exceed the capability of the sensing amplifier (SA)[3]. At the same time, the leakage currents in the array should be suppressed to reduce power consumption. As a non-volatile memory, flash memory has matured the fabrication process with good reliabilities, abilities to construct large memory arrays, and good compatibility with peri-circuits. All these make flash memory a promising candidate to meet the stringent requirements of data-intensive tasks.

Stochastic computing (SC) is one type of approximate calculation, which is an effective approach for data-intensive tasks. SC is implemented based on probabilistic calculations with inherent tolerance for noise, as shown in Fig. 1(a). Besides, SC shows great potential to simplify the hardware circuits and allows massively parallel computations towards realizing complex calculations in simple logic circuits[4]. It is anticipated that SC could be utilized in applications requiring high-speed processing for single-target detection.

So far, most works are more focused on computing complexity or merely accuracy[5], while the studies on the tradeoff of these two contradictory aspects are still limited[6]. In this work, a novel IMC architecture is designed to perform SC process for image edge detection, which provides an effective solution to address the concerns of power consumption and accuracy simultaneously.

In this work, a flash-based high-efficiency and high-precision SC strategy has been proposed, which can effectively reduce computational complexity and power consumption while retaining accuracy and interference immunity. Massive parallel operations in a large memory array have also been achieved by a novel and simple IMC architecture.

2. Implementing IMC by flash memory

The flow chart of the method is shown in Fig. 1(e). After the pre-processing of an image, the generated stochastic numbers (SNs) input to the memory array, can be applied in the image edge detection.

Correspondence to: J Z Chen, chen.jiezhi@sdu.edu.cn
Received 5 DECEMBER 2022; Revised 28 DECEMBER 2022.

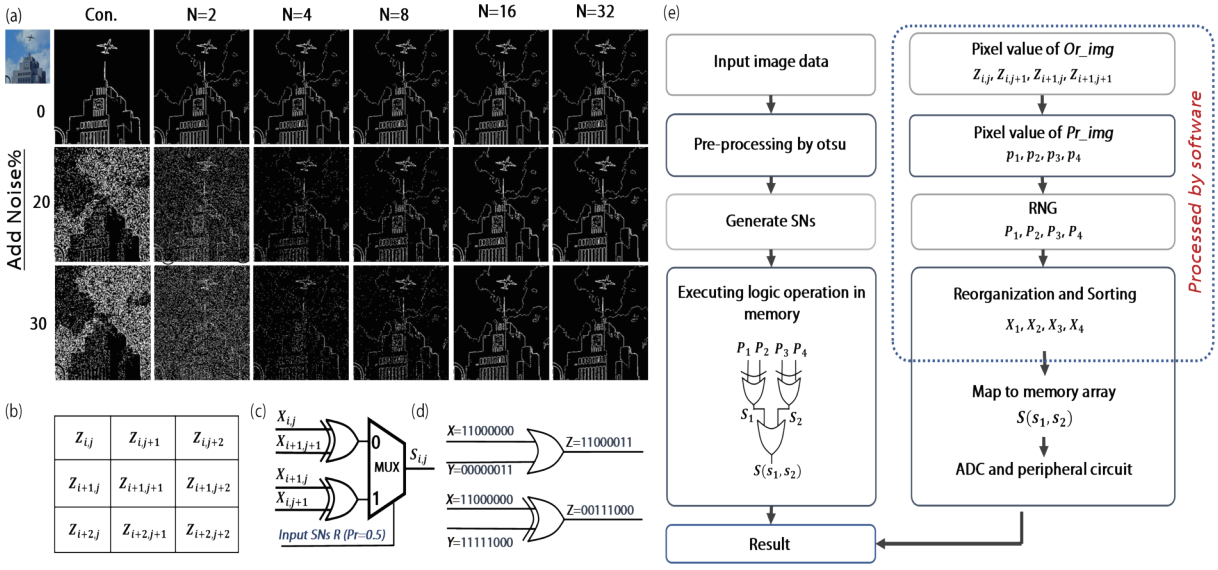


Fig. 1. (Color online) (a) Comparison between conventional and SC methods, SNs length = N bits. (b) Region of the image. (c) The stochastic computational element to realize image detection algorithmic in logic circuits. (d) Scaled addition realized by the OR gate, scaled subtraction, and absolute value calculation realized by the XOR gate. (e) The data processing flow chart in the proposed method.

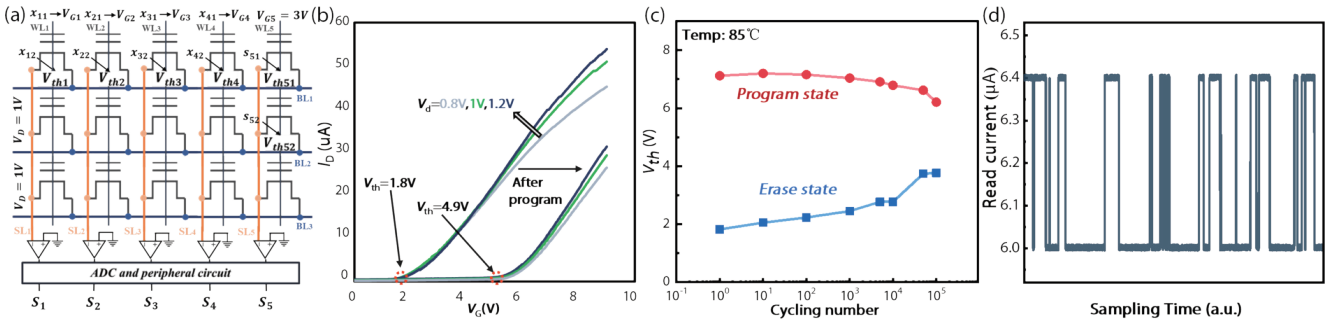


Fig. 2. (Color online) (a) The NOR flash architecture for logic operation. (b) I–V curve for NOR flash array. (c) Memory window degradation by P/E cycling. (d) Read current fluctuations can be observed caused by RTN.

2.1. Pre-processing

SC with longer SNs improves computing accuracy, but power consumption and delay will increase. To address this concern, the image can be pre-processed by the Otsu algorithm before computing[7]. Here we adopted the extended 2D Otsu to segment the pixel value of the image into three levels. The pixel values of these three levels are set to 0, 0.5, and 1. That is, each pixel value of this image is represented by the SNs with the probability value of 0, 0.5, and 1. By this means, only SNs with 50% probability needed to be computed. The overhead of a random number generator (RNG) is greatly reduced, thus decreasing the computational complexity. Besides, SNs with a 50% probability are easier to be generated by the true random number generator (TRNG), eliminating the computational trouble caused by correlation terms among SNs. Although power consumption and computational complexity are reduced, the simulation results show that it can still maintain high computational accuracy. Therefore, the required SNs are generated according to the values after three-valued segmentation processing.

2.2. Logic operations in NOR flash

After image processing and SN generating, the logic in memory operation can be implemented by NOR flash memory. Logic operations in IMC are based on Ohm's law and the Kirchhoff's law, which utilizes the cel0-ls' characteristics and the designed array to perform calculations. Fig. 2(a) shows the basic structure of the flash cells, and Fig. 2(b) shows the basic current curves. In this work, we choose the sub-saturation region as the operation region, and the source-line current (I_{SL}) can be expressed by Eq. (1),

$$I_{SL} = w(V_G - V_{th})V_D, \tag{1}$$

where w is a constant referring to the feature of devices; V_D, V_G, and V_{th} represent the drain bias, the gate bias, and the cell threshold voltages, respectively. V_{th} in each memory cell can be tuned by using erasing and programming. The entire computational process is implemented in a large array, and the number of rows enabled in the array is related to the length of the input sequence. In this way, logic operations can be implemented in the flash memory array. For example, AND operation can be accomplished by mapping in_1 to V_{G1}, and mapping in_2 to V_{th1}, thus I_{SL} can represent the logic results. Only when V_{G1} and V_{th1} are both 1, the result is 1. Other logic operations also can be realized by combing device properties and the design of flash arrays.

Along with the memory scaling, the variations of cells will have a much larger impact on the accuracy of calculations[8]. Firstly, the work regions should be optimized because large V_D (the saturation region) will result in high

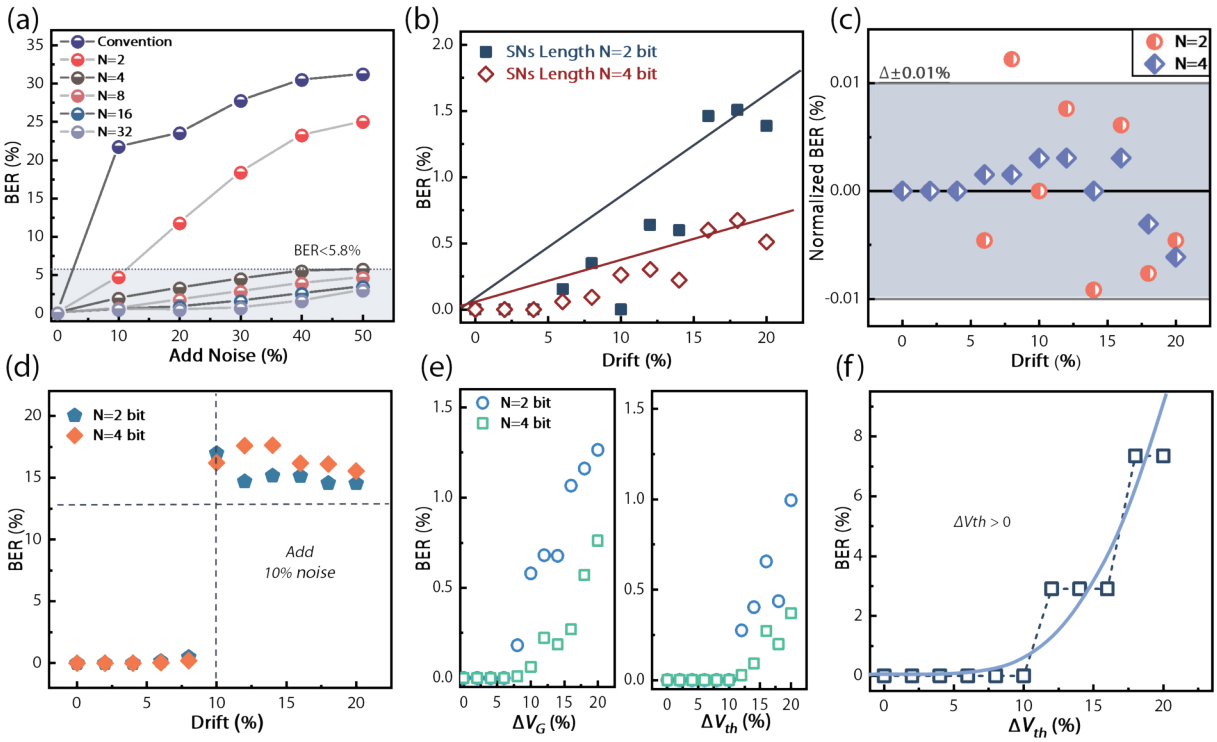


Fig. 3. (Color online) (a) BER comparison between the conventional method and SC method when the signal noise is considered. (b) Effects of simultaneous offset of device parameters $V_G/V_D/V_{th}$ on BER. (c) RTN effects are ignorable on BER. (d) Add 10% noise in (b). (e) Under different V_G/V_D conditions, effects of simultaneous 10% drift of three device parameters on BER($N = 2$). (f) Effects of V_{th} shifts.

power consumption while low V_D (the linear region) will suffer from serious variation effects. Secondly, the repeated cycling will inevitably cause memory window degradation due to V_{th} shifts, as shown in Fig. 2(c). In addition, the read noise will be serious, such as the enhanced random telegraph noise (RTN) that is caused by the traps in the tunneling oxide. For one single trap, there are two current levels at the fixed read conditions, and the current fluctuation is shown in Fig. 2(d). Scaling the cell size will further increase the fluctuations, leading to larger read current instability. All these should be well investigated for identifying the noise tolerance boundary of the designed memory array. In this work, we take the sub-saturation region (V_D = 1 V) to implement the calculations in IMC to suppress the impacts of V_D variations and guarantee low power consumption.

2.3. Image edge detection by SC

The Roberts cross kernel[9] is one of the most common edge detection operators that can be converted into simple logic operations in SC. It is easy to be adopted in IMC to implement low-precision and data-intensive applications. The Roberts cross kernel consists of two 2 × 2 conventional kernels. The magnitude result is described as Eq. (2) when considering the 3 × 3 region in Fig. 1(b).

$$S(i,j) \approx 0.5\left(\left|Z_{i,j} - Z_{i+1,j+1}\right| + \left|Z_{i+1,j} - Z_{i,j+1}\right|\right), \qquad (2)$$

where $S(i,j)$ and $Z_{i,j}$ represent the results of edge detection and the pixel value of the original image at (i,j), respectively. According to Eq. (2), we need to convert the scaled addition, the scaled subtraction, and the absolute value calculation into logic circuits respectively[4]. The scaled addition is implemented by the multiplexer. In addition, it also can be done by the OR gate. The scaled subtraction and absolute value calculation can be implemented by the XOR gate, as shown in Figs. 1(c) and 1(d).

Table 1. XOR truth table (V_D = 1 V).

x_{i1}	x_{i2}	V_G (V)	V_{th} (V)	Device state	Logical value
0	0	3	4	CLOSE	0
0	1	3	2	Sub-saturation	1
1	1	0	2	CLOSE	0

Table 2. OR truth table (V_G = 3 V, V_D = 1 V).

x_{i2}	V_{th} (V)	Device state	Logical value
0	4	CLOSE	0
1	2	Sub-saturation	1

As the key of the proposed method, the SNs are mapped into the NOR flash memory array to implement the Roberts cross kernel in Eq. (1). After XOR operations in $|Z_{i,j} - Z_{i+1,j+1}|$ and $|Z_{i+1,j} - Z_{i,j+1}|$, the OR logical operation is used as the way to execute the scaled addition.

For more detail, we take the example of the sequence length of 2-bit. As shown in Fig. 1(e), after pre-processing, p_1 and p_2 are the results after processing and represent the pixel values $Z_{i,j}$, $Z_{i+1,j+1}$, respectively. P_1 (p_{11}, p_{12}) and P_2 (p_{21}, p_{22}) are the sequences generated by TRNG. First, split and combine P_1 and P_2 to P_1' (p_{11}, p_{21}) and P_2' (p_{12}, p_{22}). Then, sort the sequences in ascending order to the new sequences, X_1 (x_{11}, x_{12}), X_2 (x_{21}, x_{22}). The above two steps are the process of reorganization and sorting, and X_1 and X_2 are the results. Finally, X_1 and X_2 are mapped to the memory array in Fig. 2(a) to complete the logical calculation. For the XOR operation, the mapping rules refer to Table 1. The first element of X_1 (x_{11}) is coded to the value of V_{G1}, and the second element of X_1 (x_{12})

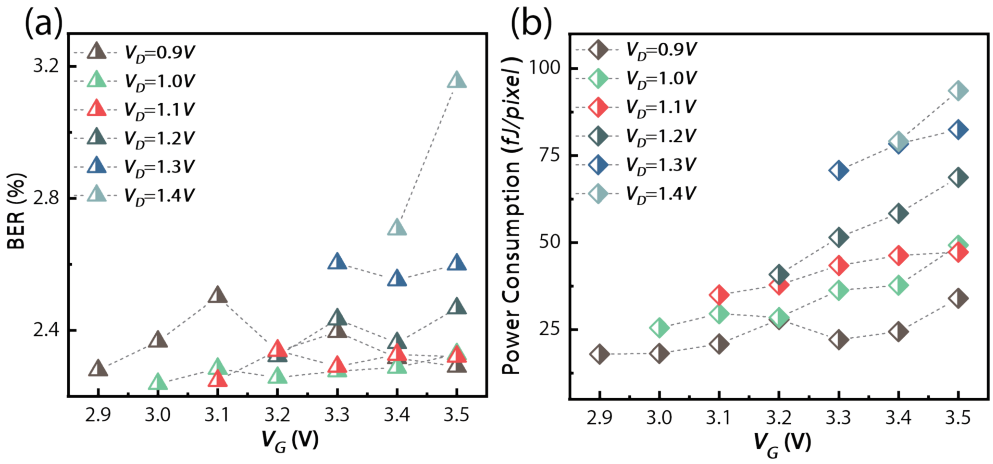


Fig. 4. (Color online) (a) BER and (b) power consumption at various V_G/V_D voltages when $N = 2$.

is coded to the value of V_{th1}. We can get the result $s_1 = p_{11}$ XOR p_{21} according to the read current I_{SL}. Similar to the OR operation, the mapping rules refer to Table 2. X_1 (x_{12}) is represented by the value of V_{th51}. X_2 (x_{22}) is represented by the value of V_{th52}. The value of OR operation is determined by the read current I_{SL}. As mentioned above, $| Z_{i,j} - Z_{i+1,j+1} | = S_1 (s_1,s_2)$, and $| Z_{i+1,j} - Z_{i,j+1} | = S_2 (s_3, s_4)$ can be implement by XOR operation, and $S = S_1 + S_2$ is calculated by OR operation.

3. Results and discussions

In this paper, a grayscale image with 256 × 256 pixels is used for the evaluation. The data is measured from flash memory of 65 nm NOR flash technology. The noise-immunity performance, device variation, and power consumption have been comprehensively investigated.

3.1. Noise-immunity performance

As shown in Fig. 3(a), different levels of interference noise (bit flipping) are added to the SNs generation stage and then compared to the noise-free results. It is observed that noise immunity is positively correlated with the sequence length. With the addition of 50% noise power, the bit error rate (BER) of the SC implementation with SNs length of 4 bits is only 1/6 of that of the conventional implementation, showing the great noise-immunity performance.

3.2. Impact of device variations

The variations of device parameters also influence SC results. Here, variations are simulated by considering the shifts from V_G, V_{th}, and V_D.

As shown in Fig. 3(b), by including 20% simultaneous drifts of three parameters at $V_G = 3$ V, $V_D = 1$ V, BER degradation is only 0.7% in the case of 4-bit SNs length. By adding RTN noise at this condition, it is found that it has a negligible effect on computational results. The change in BER is around 0.01% in Fig. 3(c). Then, we evaluate BER with the effect of 10% bit-flip noise and the simultaneous presence of parameter offsets in Fig. 3(d). The accuracy of the calculation remains stable at drift < 10%. The effects of V_G and V_{th} variation are summarized in Fig. 3(e). Benefiting from the sub-saturation region, V_D drift (up to ~20%) does not affect BER. In addition, considering read disturb, the overall rightward bias of V_{th} occurs during iterations, as shown in Fig. 3(f). In V_{th} right offset to a certain extent, the device will change from the sub-saturation region to the saturation region, and the magnitude of the current will change. To avoid this degradation, we can adjust the V_{th} according to the device characteristics and reduce its effects on BER. As shown in Fig. 4(a), the simulation results indicate that different V_G and V_D also cause different BER. In this work, the operation biases are set as $V_G = 3$ V and $V_D = 1$ V. According to different flash technologies, these parameters should be optimized to minimize BER.

3.3. Power consumption

The implementation of SC in IMC architecture has a very high degree of parallelism compared to conventional algorithms. It brings better power performance with reduced complexity. Power consumption is related to many factors, such as the size of image matrices, SN length, and the bias conditions. In Fig. 4(b), power consumption is positively correlated with the SNs' length N and correlated with the V_G and V_D, showing power consumption is also an important parameter when we choose a suitable working situation for the system. The power consumption of the proposed method is as low as 25.5 and 47.5 fJ/pixel when the sequence length is set to be $N = 2$ and $N = 4$. It should be noted, we just qualified the power consumption brought by read current, and it is necessary to reload the flash array when the new images come in, thereby, the reloading cost exists. For reference, at sub-100-ns pulse width program or erase operation[10], the power dissipation can be controlled under 20 pJ/bit (42 pJ/bit) in the program (erase) operations, respectively. These can be minimized by further optimizations on the string currents. As for the hardware resource usage, it is related to SNs length N and image size. For example, the 256 × 256 pixels point image occupies a memory size of 32 KB ($N = 2$) without any iteration, at least 30 times less than traditional methods (2T-1R)[11].

In the reported work[12], it has multiple steps to do XOR when operating in the memory array, while the work in Ref. [11] needs multiple SLIM bit cells (2T-1R in NOR flash) structures which has a larger hardware area cost. As for comparisons, in this work, our method simplifies the logic computation (only one step for XOR operation) and realizes parallel computations, which is more suitable for the application in IMC architectures. Furthermore, compared with other works by implementing logic calculation in RRAM, flash can realize much larger arrays and it has a mature technology to support large-scale operations[13, 14].

4. Conclusion

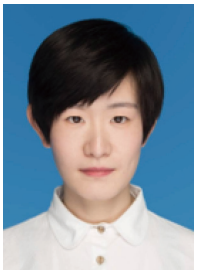
In this paper, SC in IMC for image edge detection is realized by designing logic functions in NOR flash arrays. On the one side, with respect to the standard architecture, our strategy can significantly improve the performances, such as simpler computational complexity, lower power consumption (25.5 fJ/pixel and 32 KB in case of 2-bit sequence length), and reduced occupancy of the hardware resources. On the other side, with respect to the standard IMC or SC solutions, our method has optimized the traditional SC algorithm by combining SC and IMC to lower the computational complexity and improve the parallelism. Simultaneously, it brings excellent anti-interference properties.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 62034006, 91964105, 61874068), the China Key Research and Development Program (No. 2016YFA0201802) and the Natural Science Foundation of Shandong Province (No. ZR2020JQ28), Program of Qilu Young Scholars of Shandong University.

References

[1] Wang Y, Yang Y, Hao Y, et al. Embracing the era of neuromorphic computing. J Semicond, 2021, 42(1), 010301
[2] Hao Y, Wu H, Yang Y, et al. Preface to the special issue on beyond moore: Resistive switching devices for emerging memory and neuromorphic computing. J Semicond, 2021, 42(1), 010101
[3] Lue H T, Hsu P K, Wei M L, 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). 2019 IEEE International Electron Devices Meeting (IEDM), 2019, 38.1.1
[4] Li P, Lilja D J. Using stochastic computing to implement digital image processing algorithms. 2011 IEEE 29th International Conference on Computer Design (ICCD), 2011, 154
[5] Zhang Y, Wang R, Jiang X, et al. Design guidelines of stochastic computing based on FinFET: A technology-circuit perspective. 2017 IEEE International Electron Devices Meeting (IEDM), 2017, 6.6.1
[6] Xiong H, He G. Hardware implementation of an improved stochastic computing based deep neural network using short sequence length. IEEE Trans Circuits Syst II, 2020, 67(11), 2667
[7] Otsu N. A threshold selection method from gray-level histograms. IEEE Trans Syst, Man, Cyber, 1979, 9(1), 62
[8] Mendiratta N, Tripathi S L. A review on performance comparison of advanced MOSFET structures below 45 nm technology node. J Semicond, 2020, 41(6), 061401
[9] Gonzalez, Rafael C, Digital image processing. Pearson education India. 3rd ed. , 2009, 242
[10] Feng Y, Chen B, Liu J, et al. Design-technology co-optimizations (DTCO) for general-purpose computing in-memory based on 55nm NOR flash technology. 2021 IEEE International Electron Devices Meeting (IEDM), 2021, 12.1.1
[11] Kingra S K, Parmar V, Chang C C, et al. SLIM: simultaneous logic-in-memory computing exploiting bilayer analog OxRAM devices. Sci Rep, 2020, 10(1), 1
[12] Lee J, Park B G, Kim Y. Implementation of boolean logic functions in charge trap flash for in-memory computing. IEEE Electron Device Lett, 2019, 40(9), 1358
[13] Milo V, Malavena G, Monzio Compagnoni C, et al. Memristive and CMOS devices for neuromorphic computing. Materials, 2020, 13(1), 166
[14] Yao P, Wu H, Gao B, et al. Fully hardware-implemented memristor convolutional neural network. Nature, 2020, 577(7792), 641



Zhaohui Sun got her B.S. from Shandong University in 2018. Now she is an M.S. student at Shandong University under the supervision of Prof. Jiezhi Chen. Her research focuses on in-memory computing and NOR Flash device reliability.

Jiezhi Chen received the Ph.D. degree from the Department of Informatics and Electronics, the University of Tokyo, in 2009. He is currently a Professor at the School of Information Science and Engineering, Shandong University, China. His research interests include flash memory, emerging non-volatile memories, nanoscale transistors, and computing-in-memory architectures, with the main focus on reliability physics and optimization strategies.