Fast Fourier Transform (FFT) using Flash Arrays for Noise Signal Processing

Dong Zhang, Hai Wang, Yang Feng, Xiaolin Wang, Gan Liu, Kaizhen Han, Xiao Gong, *Member, IEEE*, Jing Liu, *Member, IEEE*, Xuepeng Zhan, Jiezhi Chen, *Senior Member, IEEE.*

***Abstract*— We propose an In-Memory Computing (IMC) scheme to implement Fast Fourier Transform (FFT) for noise signal processing using flash memory. By taking advantage of the algorithms of complex number calculation and butterfly operation in FFT, the analyses of low-frequency noise (LFN) and random telegraphy noise (RTN) signals using flash-based IMC were demonstrated successfully, showing the frequency properties of 1/*f* and 1/*f*², respectively, and matching well with the Matlab numerical calculation. An energy efficiency of 2.7 TOPS/W has been achieved with a dynamic power of 27.7 mW. Our proposed IMC scheme provides an efficient way to enable FFT using a hardware solution and is extendable for other important applications such as image processing and voice recognition.**

Index Terms-NAND flash, In-Memory Computing, Fast Fourier Transform, Random Telegraphy Noise, Low Frequency Noise

I. INTRODUCTION

Fast Fourier Transform (FFT) is one of the most significant algorithms in image and signal processing which performs the transformation between the time domain and the frequency domain. Compared with the traditional Discrete Fourier Transform (DFT) algorithms, FFT reduces the computational complexity from $O(n{\cdot}n)$ to $O(n{\cdot}log_2 n)$ [1]. Despite this advantage, the process of FFT still requires a massive amount of Multiplication and Accumulation (MAC) operations, the number of which increases drastically with the explosive data [2-3]. Implementing the computation-intensive FFT using conventional Von-Neuman architecture suffers from inherent drawbacks due to the "Memory Wall" issue where a tremendous amount of data movement between the processing element and memory hierarchy is unavoidable, with the dilemma of very limited bandwidth between the computing and storage units [4-7]. This necessitates the urgent search for innovative approaches beyond what is capable by the Von-Neuman architecture.

In-Memory Computing (IMC) has emerged as an elegant solution in recent years due to its advantage of high processing parallelism and its capability of minimizing data movement [8-10]. In fact, various emerging non-volatile memories have been reported as promising candidates to perform IMC, including resistive random access memory (RRAM), ferroelectric random access memory (FeRAM), magnetic memory, etc [11-17]. However, serval technological challenges are yet to be addressed before the wide adoption of these technologies. In contrast, NAND flash enjoys the benefits of mature device technology, multiple bits per cell, good conductance linearity, and long retention time, making it particularly attractive to perform FFT [18-20].

In this work, the NAND flash array compatible with nowadays fabrication technology is proposed to enable FFT in an IMC manner. When employing this approach to analyze the noisy signals, excellent agreement with Matlab numerical calculations was obtained. The principle and detailed IMC procedures are explained, including the pre-processing of input and the FFT vector, complex number calculation, and butterfly operation. In addition, the dynamic performances considering peripheral circuits have been evaluated.

II. METHODS

Fig.1 shows some preliminary work for the input vector. The first step is to transform the analog data into digital data. For a clearer view, different data points in digital are represented by different colors. Note that in Fig. 1, the processing of 8 data points is used as an example to illustrate the mechanism, and 4096 data points are used in the following simulation analyses. The next step of bit reversal is crucial: before calculation, the signal data needs to be rearranged for (N-1) times, where N is equal to $log_2 N_d$, and N_d is the number of data points. For each rearrangement, the odd numbers and even numbers are separated and then put together respectively. After that, for pre-processing, the data are transformed into binary, split according to the weights, and then converted to the drain voltage (V_d) pulses. There are three types of weights considered: the magnitude, the sign of value, and the characteristics of a complex number: real (R) or imaginary (I). In the first two stages of calculation, the input vectors are real numbers, while in the rest stages, the input vectors contain complex numbers since the corresponding FFT vectors to be multiplied are complex numbers starting from the second stage.

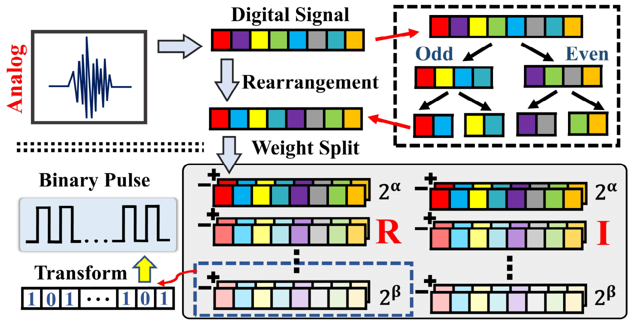

Fig. 1. The schematic of the pre-processing work for the input vector, which is coded after bit reversal as well as weight split and then transformed to binary V_d pulses.

This work was supported in part by National Natural Science Foundation of China (Nos. 62034006, 91964105, 61874068), China Key Research and Development Program (No.2016YFA0201802), Natural Science Foundation of Shandong Province (No. ZR2020JQ28), Singapore Ministry of Education (Tier 2: MOE2018-T2-2-154).

Dong Zhang, Hai Wang, Yang Feng, Xuepeng Zhan, and Jiezhi Chen are with the School of Information Science and Engineering, Shandong University, Qingdao 266237, China (e-mail: chen.jiezhi@sdu.edu.cn).

Dong Zhang, Xiaolin Wang, Gan Liu, Kaizhen Han, and Xiao Gong are with the Department of Electrical and Computer Engineering, National University of Singapore, 117576, Singapore (e-mail: elegong@nus.edu.sg).

Jing Liu is with the Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China (e-mail: liujing@ime.ac.cn).

(Corresponding authors: Jiezhi Chen, Xiao Gong, Jing Liu).

This article has been accepted for publication in IEEE Electron Device Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LED.2022.3183111

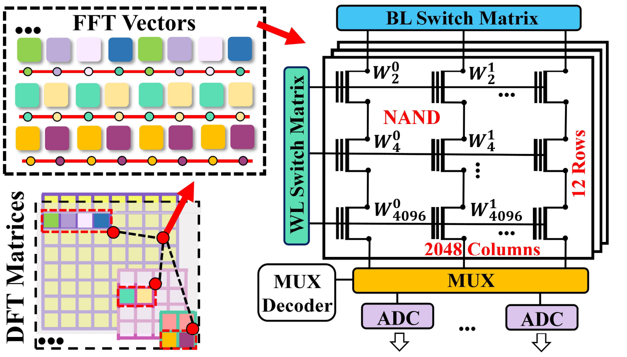

Fig. 2. The illustration of the processing of FFT vectors and the IMC circuit with NAND flash arrays. The values in DFT matrices are extracted and transformed to the conductance states in NAND flash cells.

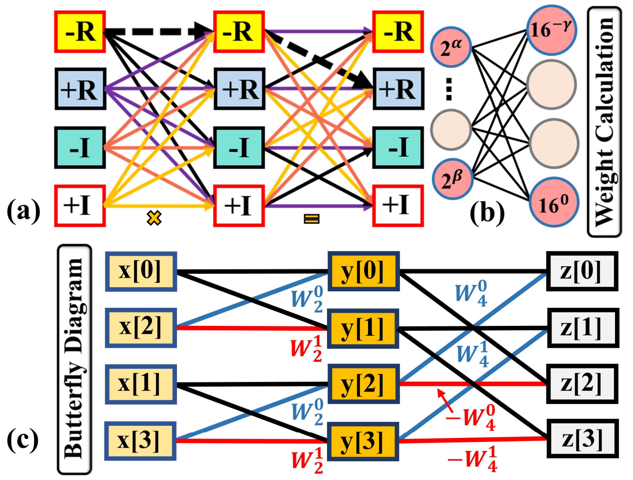

Fig. 3. (a)-(b) The diagram of complex number calculation considering different weights. (c) The schematic of the butterfly algorithm, where two stages are shown as an example.

The FFT vectors extracted from DFT matrices need to be pre-processed as well. The splitting of FFT vectors, which contain the elements of $e^{i\theta}$ (complex numbers), is based on the Euler formula shown in (1). Hex conversion and weight split are illustrated in our previous work in detail [10].

$$e^{i\theta} = \cos(\theta) + i\sin(\theta)\,. \tag{1}$$

The selected FFT vectors are then converted to the conductance and mapped into NAND flash cells repeatedly, as shown in Fig. 2. The number of repetition times is determined by the length of the input data and the FFT vector, where the length of the FFT vector multiplies the repetition times equals half the length of input data points. By applying gate voltage pulses to adjust the threshold voltage of the NAND flash, FFT vector mapping can be done. We adopt a 65 nm NAND flash with 4 bits/cell in this work, whose conductance ranges from 10^{-8} S to 2×10^{-7} S. In memory array, the number of bit lines equals half the number of data points ($N_d/2$), and the number of word lines equals the number of stages (N). Therefore, the number of bit lines should be much larger than the number of word lines when processing big data containing a large number of data points. Such requirements can be fulfilled by state-of-the-art NAND flash structure, indicating the great compatibility of our scheme with the modern industrial platform. The IMC circuit design, including memory array and peripheral circuits, is shown in the right part of Fig. 2. For processing 4096 data points, the corresponding memory array size is 2048×12. Some specific circuit parameters, such as read voltage and pulse duration, can be found in the reference papers [21-22].

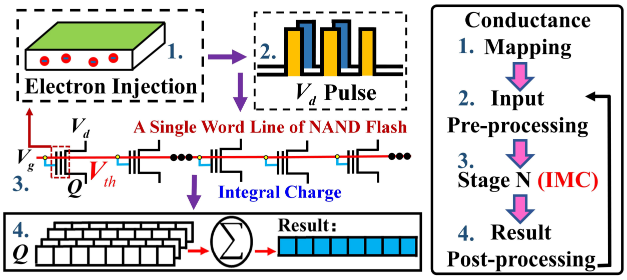

Fig. 4. The illustration of parallel calculation with a single word line of NAND flash and the flow chart of the scheme, including four steps and one iteration process.

The calculation method considering different weights is shown in Figs. 3(a) and (b). After pre-processing, input and FFT vectors are divided into several vectors with just a single weight. For instance, complex numbers are divided into four parts: +R, -R, +I, and -I. The divided data are split further, considering the weight of the magnitude. When calculating, as to the real part, negative times negative yields positive, which is highlighted with a dashed line. In Fig. 3(b), the values of α and β are related to the magnitude of the input vector, and the value of γ is determined by the bit precision of the FFT vector. The parameter along with the lines in the same color, $W_{N_d}^{n}$, which is equal to $e^{i2kn\pi/N_d}$, represents the value to be multiplied. For example, the calculation process of z[2] is shown in (2). Such a multiplication process can be performed under our proposed circuit structure.

$$\mathrm{z}[2]=\mathrm{x}(0)+W_2^0\cdot\mathrm{x}(2)\text{-}W_4^0[\mathrm{x}(1)+W_2^0\cdot\mathrm{x}(3)]\,. \tag{2}$$

The derivation of FFT is shown in (3), where n ∈ [0, N_d -1], m ∈ [0, N_d /2-1], and k ∈ [0, N_d -1], which are integer. In (3), the DFT equation is divided into two parts, corresponding to two types of inputs: odd [$\mathrm{x}(2\mathrm{m}+1)$] and even [$\mathrm{x}(2\mathrm{m})$]. The two divided parts can be divided further into four simpler parts. This process proceeds until the simplest FFT vector is recurved. This is the reason why the input vector should be rearranged for (*N-1*) times before calculations.

$$\begin{aligned}\mathrm{X(k)} &= \sum_{\mathrm{n=0}}^{N_d\text{-}1}\mathrm{x(n)}e^{i\frac{2\mathrm{kn}\pi}{N_d}} \\ &= \sum_{\mathrm{m=0}}^{N_d/2\text{-}1}\mathrm{x(2m)}e^{i\frac{2\mathrm{k(2m)}\pi}{N_d}} + \sum_{\mathrm{m=0}}^{N_d/2\text{-}1}\mathrm{x(2m{+}1)}e^{i\frac{2\mathrm{k(2m+1)}\pi}{N_d}} \\ &= \sum_{\mathrm{m=0}}^{\mathrm{N_d}/2\text{-}1}\mathrm{x(2m)}e^{i\frac{2\mathrm{km}\pi}{N_d/2}} + e^{i\frac{2\mathrm{k}\pi}{N_d}}\cdot\sum_{\mathrm{m=0}}^{N_d/2\text{-}1}\mathrm{x(2m{+}1)}e^{i\frac{2\mathrm{km}\pi}{N_d/2}}\end{aligned} \tag{3}$$

Fig. 4 shows the flow chart of the calculations. The first step is conductance mapping in flash arrays. Then, the processed input vector is calculated in arrays based on Ohm's law. During the calculation, only one row of NAND flash operates in the near-threshold voltage (V_{th}) region, while other rows work in the super V_{th} region, by adjusting gate voltage (V_g). The mechanism is the same as the reading process of NAND flash memory. It is known that the conductance of NAND flash in the super V_{th} region is much larger than that in the near V_{th} region. Hence, the conductance of a NAND flash column is dominated by the cell working in the near V_{th} region. In this way, a single row selecting mode can be achieved in every calculation stage. Once one calculation stage is done, the results (integral charges) are summed considering different weights. After that, the results are pre-processed, converted to input voltages, and prepared for the next stage of calculation.

This article has been accepted for publication in IEEE Electron Device Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LED.2022.3183111

III. Results and Analyses

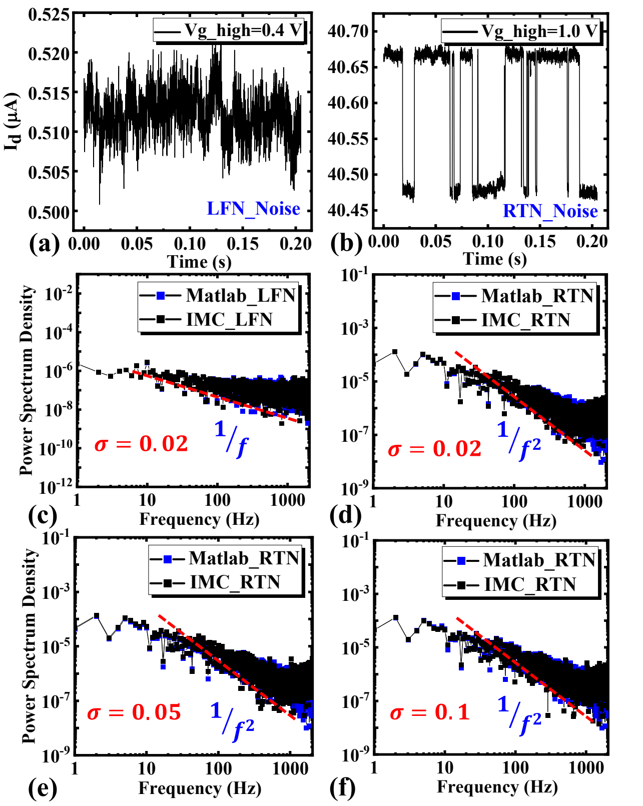

Fig. 5. (a)-(b) The measured noisy signals under different gate voltages. (c)-(d) The comparison between the results from Matlab calculation and IMC scheme. (e)-(f) The comparison of the results under various read noise.

To study our proposed method, we perform FFT calculations using NAND flash-based IMC platform to study noise signals: low-frequency noise (LFN) and random telegraphy noise (RTN). LFN is a serious concern in the nano-scale device because even a single trap can trigger a large RTN and cause obvious current fluctuations [23-26]. Figs. 5(a) and (b) are noisy signals measured in transistors with 90 nm channel length under various V_g pulses, where the sampling rate is 20000 Hz, showing large drain current (I_d) fluctuations.

Figs. 5(c) and (d) show the simulation results of LFN and RTN signals. σ, set as 0.02, is the standard deviation of a Gaussian distributed read noise induced by circuit parameter fluctuations. The numerical results calculated by the software of Matlab are put in the same figure with the simulation results. The results show that for LFN, the Power Spectrum Density (PSD) shows the slope of $1/f$; while for RTN, the slope is $1/f^2$. The results from IMC using our proposed method match well with the numerical ones and other previous reports [27-28]. It can be observed that the accuracy degrades with the decrease of Power Spectrum Density (PSD). The reason is that the selected bit precision of input and FFT vector are compromised to realize better energy efficiency. In addition, read noise also affects accuracy.

However, high accuracy is achieved over 5 orders of PSD, which can satisfy the requirements of most applications. Figs. 5(c) and (d) show an accurate range of less than 5 orders since the DC data cannot be plotted in a log-scaled coordinate. The impact of read noise is also simulated and analyzed. The results with $\sigma = 0.05$ and $\sigma = 0.1$ are presented in Figs. 5(e) and (f), respectively, and it can be found that as σ becomes higher, more data points at large PSD values get affected.

Table 1 Comparison with DFT Processor

	This work	[13]
Algorithm	FFT	DFT
Computational Complexity	$O(n{\cdot}log_2 n)$	$O(n{\cdot}n)$
Memory Cell	NAND Flash	RRAM
Area Efficiency	√√	√
Mapping method	Weight Split	Full Analog
Energy Consumption	√√	√√
Big Data Processing	√√	√

Table 2 Comparison with other FFT Processors

	This work	[2]	[3]
FFT size (N)	4096	1024	1024
Memory Type	NAND (non-volatile)	SRAM	Register Based
Bit/cell	4	1	1
Area (mm^2)	0.078	0.99	3.6
Power (mW)	27.7	123	60.3

Table 1 shows the comparison between this work and the recent study of DFT based on IMC [13]. Taking advantage of the less computational complexity of the FFT algorithm, a smaller array size can be obtained, leading to a better area efficiency. Such an advantage would be more significant as the array size scales up. When processing big data, it is challenging to ensure high precision in each cell in a huge array size. Due to the mature technology, NAND flash is capable of realizing decent precision control for the unit cell at a large array size. The method of weight split is adopted as well to give additional precision control. Although the energy consumption of the weight split is higher than the full analog scheme at the same array size, the total energy consumption can be significantly reduced by employing a small array size, thanks to the superiority of the FFT algorithm.

We also perform some dynamic assessments using NeuroSim to evaluate the proposed FFT scheme, and some key figure-of-merits are benchmarked with the state-of-the-art FFT processors, as shown in Table 2. The evaluation is at the circuit level, considering both the memory array and peripheral circuits shown in Fig. 2. During the FFT calculations under the IMC frame, our design shows excellent energy efficiency of 2.7 TOPS/W with a dynamic power of 27.7 mW. By leveraging the high-density property of NAND flash, the achieved area size is as small as 0.078 mm^2. Moreover, due to the 3D integration compatibility of NAND flash technology, the array structure proposed in this work is promising as a functional module in future 3D ICs to further improve area efficiency.

IV. Conclusion

In this work, we have demonstrated the feasibility of enabling FFT using the hardware-based NAND Flash having mature device technology and multiple bits per cell. The results obtained through analyzing the properties of LFN as well as RTN of the nano-scale transistors and the dynamic analysis suggest the great possibility and promise of using our proposed Flash-based IMC scheme for many other signal processing tasks.

References

[1] B. N. Mohapatra and R. K. Mohapatra, "FFT and sparse FFT techniques and applications," *Fourteenth International Conference on Wireless and Optical Communications Networks (WOCN),* 2017, pp. 1-5, doi: 10.1109/WOCN.2017.8065859.

[2] H. E. Yantir, W. Guo, A. M. Eltawil, F. J. Kurdahi, and K. N. Salama, "An ultra-area-efficient 1024-point in-memory FFT processor," *Micromachines*, vol. 10, no. 8, pp. 509, Jul. 2019. doi: 10.3390/mi10080509.

[3] N. Le Ba and T. T. Kim, "An area efficient 1024-point low power radix-22 FFT processor with feed-forward multiple delay commutators," *IEEE Trans. Circuits Syst. I*, Reg. Papers, vol. 65, no. 10, pp. 3291-3299, Oct. 2018, doi: 10.1109/TCSI.2018.2831007.

[4] W. Zhang, B. Gao, J. Tang, P. Yao, S. Yu, M.-F. Chang, H.-J. Yoo, H.Qian, and H. Wu, "Neuro-inspired computing chips," *Nature Electron.,* vol. 3, no. 7, pp. 371–382, Jul. 2020, doi: 10.1038/s41928-020-0435-7.

[5] R. Han, P. Huang, Y. Xiang, C. Liu, Z. Dong, Z. Su, Y. Liu, X. Liu, and J. Kang, "A novel convolution computing paradigm based on NOR flash array with high computing speed and energy efficiency," *IEEE Trans. Circuits Syst. I*, Reg. Papers, vol. 66, no. 5, pp. 1692-1703, May 2019, doi: 10.1109/TCSI.2018.2885574.

[6] Q. Xia and J. J. Yang, "Memristive crossbar arrays for brain-inspired computing," *Nature Mater*., vol. 18, no. 4, pp. 309–323, Apr. 2019, doi: 10.1038/s41563-019-0291-x.

[7] M. Le Gallo, A. Sebastian, R. Mathis, M. Manica, H. Giefers, T. Tuma, C. Bekas, A. Curioni, and E. Eleftheriou, "Mixed-precision in-memory computing," *Nature Electron*., vol. 1, no. 4, pp. 246-253, Apr. 2018, doi: 10.1038/s41928-018-0054-8.

[8] Y. Xiang, P. Huang, H. Yang, K. Wang, R. Han, W. Shen, Y. Feng, C. Liu, X. Liu, and J. Kang, "Storage reliability of multi-bit flash oriented to deep neural network," in *IEDM Tech. Dig.,* Dec. 2019, pp. 38.2.1-38.2.4, doi: 10.1109/IEDM19573.2019.8993508.

[9] Y. Feng, X. Zhan and J. Chen, "Flash memory based computing-in-memory to solve time-dependent partial differential equations," in *Proc. Silicon Nanoelectron. Workshop (SNW),* June. 2020, pp. 27-28, doi: 10.1109/SNW50361.2020.9131425.

[10] D. Zhang, H. Wang, Y. Feng, X. Zhan, J. Chen, J. Liu, and M. Liu, "Implementation of image compression by using high-precision in-memory computing scheme based on NOR flash memory," *IEEE Electron Device Lett.,* vol. 42, no. 11, pp. 1603-1606, Nov. 2021, doi: 10.1109/LED.2021.3114407.

[11] R. Han, P. Huang, Y. Zhao, X. Cui, X. Liu, and J. Kang, "Efficient evaluation model including interconnect resistance effect for large scale RRAM crossbar array matrix computing," *Sci. China Inf. Sci.,* vol. 62, no. 2, pp. 1–11, Feb. 2019, doi: 10.1007/s11432-018-9555-8.

[12] Z. Zhou, L. Jiao, J. Zhou, Q. Kong, S. Luo, C. Sun, Z. Zheng, X. Wang, D. Zhang, G. Liu, G. Liang, and X. Gong, "Time-dependent Landau-Ginzburg equation-based ferroelectric tunnel junction modeling with dynamic response and multi-domain characteristics," *IEEE Electron Device Lett.*, doi: 10.1109/LED.2021.3128998.

[13] H. Zhao, Z. Liu, J. Tang, B. Gao, Y. Zhou, P. Yao, Y. Xi, H. Qian, and H. Wu, "Implementation of discrete fourier transform using RRAM arrays with quasi-analog mapping for high-fidelity medical image reconstruction," in *IEDM Tech. Dig.* Dec. 2021, pp. 12.4.1-12.4.4, doi: 10.1109/IEDM19574.2021.9720547.

[14] P. Yao, H. Wu, B. Gao, J. Tang, Q. Zhang, W. Zhang, J. J. Yang, and H. Qian, "Fully hardware-implemented memristor convolutional neural network," *Nature*, vol. 577, no. 7792, pp. 641–646, Jan. 2020, doi: 10.1038/s41586-020-1942-4.

[15] Y. Xiang, P. Huang, Z. Zhou, Y. Jiang, Q. Shu, Z. Su, Y. Liu, X. Liu, and J. Kang, "Analog deep neural network based on NOR flash computing array for high speed/energy efficiency computation," *in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS),* May 2019, pp. 7–10, doi: 10.1109/ISCAS.2019.8702401.

[16] J. Zhou, Z. Zhou, X. Wang, H. Wang, C. Sun, K. Han, Y. Kang, and X. Gong, "Temperature dependence of ferroelectricity in Al-doped HfO_2 featuring a high Pr of 23.7 μC/cm^2," *IEEE Trans. Electron Devices,* vol. 67, no. 12, pp. 5633-5638, Dec. 2020, doi: 10.1109/TED.2020.3032350.

[17] C. Sun, K. Han, S. Samanta, Q. Kong, J. Zhang, H. Xu, X. Wang, A. Kumar, C. Wang, Z. Zheng, X. Yin, K. Ni, and X. Gong, "First demonstration of BEOL-compatible ferroelectric TCAM featuring a-IGZO Fe-TFTs with large memory window of 2.9 V, scaled channel length of 40 nm, and high endurance of 108 cycles," *in Proc. IEEE Symp. VLSI Technol.*, 2021, pp. 1-2.

[18] K. Ishimaru, "Future of non-volatile memory-from storage to com puting-," *in IEDM Tech. Dig.,* Dec. 2019, pp. 1.3.1-1.3.6, doi: 10.1109/IEDM19573.2019.8993609.

[19] Y. Feng, B. Chen, J. Liu, Z. Sun, H. Hu, J. Zhang, X. Zhan, and J. Chen, "Design-technology co-optimizations (DTCO) for general-purpose computing in-memory based on 55nm NOR flash technology," in *IEDM Tech. Dig.* Dec. 2021, pp. 12.1.1-12.1.4, doi: 10.1109/IEDM19574.2021.9720625.

[20] H. Lue, P. Hsu, M. Wei, T. Yeh, P. Du, W. Chen, K. Wang, and C. Lu, "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 *IEDM Tech. Dig.,* Dec. 2019, pp. 38.1.1-38.1.4, doi: 10.1109/IEDM19573.2019.8993652.

[21] P. Chen, X. Peng and S. Yu, "NeuroSim+: An integrated device-to-algorithm framework for benchmarking synaptic devices and array architectures," in *IEDM Tech. Dig.* Dec. 2017, pp. 6.1.1-6.1.4, doi: 10.1109/IEDM.2017.8268337.

[22] P. Chen, X. Peng and S. Yu, "NeuroSim: A circuit-level macro model for benchmarking neuro-inspired architectures in online learning," *in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 12, pp. 3067-3080, Dec. 2018, doi: 10.1109/TCAD.2018.2789723.

[23] T. Gong, Q. Luo, X. Xu, J. Yu, D. Dong, H. Lv, P. Yuan, C. Chen, J. Yin, L. Tai, X. Zhu, Q. Liu, S. Long, and M. Liu, "Classification of three-level random telegraph noise and its application in accurate extraction of trap profiles in oxide-based resistive switching memory," *IEEE Electron Device Lett.,* vol. 39, no. 9, pp. 1302-1305, Sept. 2018, doi: 10.1109/LED.2018.2858245.

[24] M. Mehedi, K. H. Tok, J. F. Zhang, Z. Ji, Z. Ye, W. Zhang, and J. S. Marsland,"An assessment of the statistical distribution of random telegraph noise time constants," *IEEE Access,* vol. 8, pp. 182273–182282, 2020, doi: 10.1109/ACCESS.2020.3028747.

[25] M. Luo, R. Wang, S. Guo, J. Wang, J. Zou, and R. Huang, "Impacts of random telegraph noise (RTN) on digital circuits," *IEEE Trans. Electron Devices,* vol. 62, no. 6, pp. 1725-1732, June 2015, doi: 10.1109/TED.2014.2368191.

[26] X. Zhan, C. Shen, Z. Ji, J. Chen, H. Fang, F. Guo, and J. Zhang, "A dual-point technique for the entire ID–VG characterization into subthreshold region under random telegraph noise condition," *IEEE Electron Device Lett.,* vol. 40, no. 5, pp. 674-677, May 2019, doi: 10.1109/LED.2019.2903516.

[27] A. Sarkar, S. De, A. Dey, and C.K. Sarkar, "1/f noise and analogue performance study of short-channel cylindrical surrounding gate MOSFET using a new subthreshold analytical pseudo-two-dimensional model," *IET Circuits Dev. Syst,* vol. 6, no. 1, pp. 28-34, Jan. 2012, doi:10.1049/iet-cds.2011.0093.

[28] W. Wu, H. Wu, M. Si, N. Conrad, Y. Zhao, and P. D. Ye, "RTN and low frequency noise on ultra-scaled near-ballistic Ge nanowire nMOSFETs," *in Proc. IEEE Symp. VLSI Technol.,* Jun. 2016, pp. 1–2, doi: 10.1109/VLSIT.2016.7573421.