Proceedings of CONF-SEML 2026 Symposium: Multimodal Data Acquisition: Applications in Physiological and Behavioral Research
DOI: 10.54254/2755-2721/2026.AD33666

HZO FeFET-Based Computing-in-Memory Array for Matrix Multiplication

Zhihao Lu

School of Integrated Circuits, Southeast University, Nanjing, China
213232525@seu.edu.cn

Abstract. The fast development of artificial intelligence toward big models, many parallel calculations, and edge-cloud cooperation has further shown the limits in performance and energy use of traditional von Neumann structures. In these structures, memory and processing units are physically separated, which makes data transfer happen very often. This leads to too much delay and extra energy waste linked with the memory wall. Computing-in-memory is a good new choice because it does arithmetic tasks directly inside memory arrays, so it cuts down data moving as much as possible. Among new nonvolatile memory devices, ferroelectric field-effect transistors are very good for CIM. They have many advantages, such as nonvolatility, high ability to read data correctly, low energy use for writing data, and good compatibility with CMOS integration. Especially, FeFETs made of $Hf_{1-x}Zr_xO_2$ show steady ferroelectric properties even in very small nanoscale sizes. This makes it possible to make CIM hardware that can be scaled up and uses energy efficiently. In this work, we use a crossbar array based on HZO FeFETs to do matrix multiplication in a CIM system. We put forward a calculation way where weight values are stored in FeFET polarization states, input activation signals are sent through word lines, and multiply–accumulate operations are done by adding currents in bit lines. We design a 3×3 HZO FeFET array to test the matrix-multiplication method in real experiments. Test results show that the array does in-memory MAC operations correctly, and the output results are the same as theoretical values. We also study its scalability, and it shows that this design can be extended from a 3×3 array to any common n×n array without changing the basic calculation principle. These findings prove that HZO FeFET-based CIM can work well for efficient matrix operations. It is a practical hardware way for future AI computing systems that use low power and have high parallel working ability.

Keywords: FeFET, HZO, computing-in-memory, matrix multiplication.

1. Introduction

With the fast growth of artificial intelligence (AI), especially in large-scale models, many parallel calculations and edge-cloud collaborative systems, the limits of traditional von Neumann architecture have become more and more obvious. It has turned into a main block for further improvements in computing performance and energy efficiency [1]. In the von Neumann architecture, processing units and memory units are separated physically. Data needs to be

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.AD33666

transferred often between processors and memories through communication buses. This repeated data movement brings large delay and energy use, which is usually called the memory wall [2]. Such a block is especially serious for AI work. AI work needs a lot of data and supports massive parallel calculations. In both Transformer-based model training and convolutional neural network inference, the main computing work is large-scale multiply-and-accumulate (MAC) operations. These operations usually take up most of the total computing cost [3]. As model sizes keep growing, the difference between processor throughput and memory bandwidth becomes more clear. Computing resources are not fully used because data supply is not enough [4]. So, solving the memory block at the architecture level has become a key need for next-generation AI hardware.

Computing-in-memory (CIM) is widely regarded as an effective way to ease the von Neumann block. It integrates computing work directly into the memory array [5]. In a common CIM architecture, weights are stored in memory cells as physical device states. Input signals are sent to array lines. Then multiplication and accumulation can be done directly in the array through device-level electrical responses [6]. This in-array computing way improves data locality and computing parallelism a lot. So it has clear advantages in working speed and energy efficiency. Present CIM uses can be mainly divided into volatile-memory-based solutions and nonvolatile-memory-based solutions [7]. SRAM- and DRAM-based CIM architectures have mature making technologies and fast working speed. But they have relatively high static power use and cannot keep data without power. On the contrary, CIM architectures based on nonvolatile memory (NVM) are more suitable for energy-limited uses. They can keep data without continuous power supply and use low power when waiting [8].

Among NVM choices, resistive random-access memory (RRAM) and phase-change memory (PCM) have been studied widely for crossbar-based computing. Weights are shown by conductance levels. Matrix operations are done following Ohm's law and Kirchhoff's current law [9]. But these two-terminal resistive devices often have some problems. They have random filament formation, big differences between different devices and complex writing mechanisms. These problems may lower computing accuracy and add system cost in large arrays [10]. In this case, ferroelectric field-effect transistors (FeFETs) have become a good choice for CIM uses [11]. FeFETs add a ferroelectric layer into the gate stack. They store information in the form of polarization-modulated threshold-voltage states. So they can work without continuous power. Compared with two-terminal resistive devices, FeFETs have better read selectivity and can well stop sneak-path currents. What's more, their voltage-controlled programming way helps lower writing energy and improve working speed. Their three-terminal structure also makes it easy to work with CMOS peripheral circuits, which is good for making real systems [12].

The performance of FeFETs is closely connected with the choice of ferroelectric material. Traditional perovskite ferroelectrics such as $Pb(Zr,Ti)O_3$ have good polarization characteristics. But they are not suitable for advanced CMOS processes and have limits in being scaled down [13]. In recent years, hafnium-oxide-based ferroelectrics, especially $Hf_{1-x}Zr_xO_2$ (HZO), have helped FeFET technology make great progress [14]. HZO keeps stable ferroelectricity at nanometer thicknesses because it has non-centrosymmetric orthorhombic phase. So it matches well with scaled transistor technologies. Besides, HZO-based FeFETs have many key advantages. They are compatible with CMOS, easy to scale down, use low polarization switching voltage and can support multi-level-cell operation. Better retention and endurance also make them more suitable for large-scale integration [15]. So HZO FeFETs are a good device platform for low-power and high-density CIM systems.

Although FeFETs have these advantages, most present studies only focus on material optimization or single-device features. Array-level functional verification of FeFET-based CIM is

DOI: 10.54254/2755-2721/2026.AD33666

still not enough. Especially, practical hardware mapping ways for matrix computation have not been shown clearly and in a scalable way. Also, some reported CIM schemes need multilevel analog conductance modulation. This has strict requirements for device uniformity and peripheral-circuit precision. So it makes implementation and scaling more difficult. Driven by these challenges, this work studies a simple FeFET-array-based CIM way and checks if it can work for matrix multiplication. Specifically, an HZO FeFET crossbar array is built for in-memory matrix computation. A computing way is put forward first. Weight values are mapped to FeFET polarization states, input activations are sent through word lines and MAC operations are realized by bit-line current accumulation. Then a 3 × 3 HZO FeFET array is designed to test the matrix multiplication process in experiments. Finally, the scalability of the proposed architecture to a general n × n array is discussed. The proposed way provides a feasible hardware path for AI-oriented CIM systems based on HZO FeFET technology.

2. Operating principle of HZO FeFET devices

2.1. Device structure and basic operating mechanism

An HZO FeFET is a three-terminal device that can store data without continuous power. It is made by adding an HZO ferroelectric layer to the gate dielectric stack of a common MOSFET. Its basic parts include the gate, source, drain and the ferroelectric gate dielectric layer. A regular MOSFET controls channel conduction mainly by changing electric charges. But an FeFET uses the polarization state of the ferroelectric layer to change the number of carriers in the channel. This lets its electrical features be set and adjusted freely.

When an external voltage is added to the gate, ferroelectric domains in the HZO layer change direction under the electric field. When the electric field is removed, a steady remanent polarization still remains. This leftover polarization makes carriers gather or reduce at the channel interface. It causes the threshold voltage Vth to shift back and forth. So different polarization states match different conduction states. For this reason, the device can keep its electrical state without constant power supply. It can realize nonvolatile data storage. The structure and working principle of this device are shown in Fig. 1.

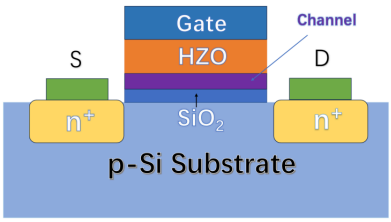

Figure 1. Schematic structure of the HZO FeFET

2.2. Polarization-modulated data storage mechanism

The data storage function of an HZO FeFET comes from the ferroelectric polarization state that changes channel conductance. When the ferroelectric layer is polarized to different directions, the device shows clearly different threshold voltages and conduction features, so it can form stable storage states.

DOI: 10.54254/2755-2721/2026.AD33666

As shown in Fig. 2, when a positive gate voltage makes ferroelectric polarization point to the channel, electrons gather near the interface. Channel conductance becomes higher, and the device stays at a state with lower threshold voltage. On the contrary, when a reverse gate voltage moves polarization away from the channel, carriers in the channel become fewer. Then the device turns to a state with higher threshold voltage. These two stable polarization states can match binary logic states well, and they act as the physical basis for nonvolatile storage in FeFETs.

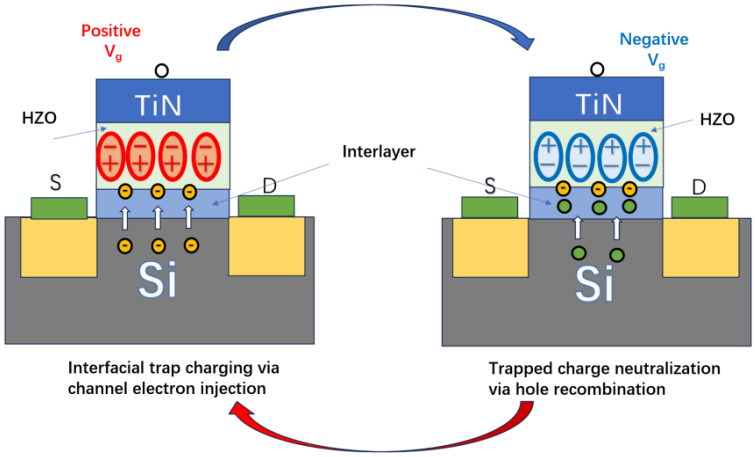

Figure 2. Schematic illustration of ferroelectric domain polarization after gate-voltage application

Besides binary data storage, partial polarization switching can take place if we control the strength and length of the programming pulse in a right way. This makes middle polarization states appear between the two full polarization states, and it brings about several different levels of channel conductance. This kind of device performance can be used for multi-level-cell storage and multi-bit weight representation. It is very helpful for raising storage density and computing accuracy in CIM systems.

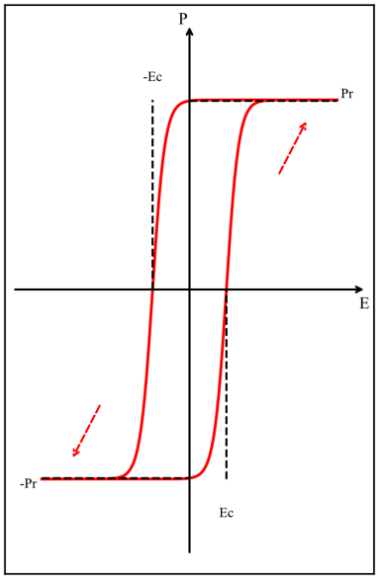

Figure 3. Polarization-electric-field (P-E) hysteresis curve of the ferroelectric material

DOI: 10.54254/2755-2721/2026.AD33666

The polarization–electric-field hysteresis characteristic of the ferroelectric layer, as shown in Fig. 3, underlies this nonvolatile behavior. Since the stored state is associated with a stable lattice polarization rather than trapped charge, the FeFET can retain information for a long duration without power supply. This property is particularly attractive for CIM architectures, where weight values must remain stored in the array during repeated computations.

3. FeFET-array-based computing-in-memory matrix multiplication

3.1. FeFET crossbar architecture and computation mechanism

At the array level, HZO FeFETs can be arranged in a crossbar structure to realize CIM functionality. The crossbar consists of orthogonally arranged word lines (WLs) and bit lines (BLs), with one FeFET located at each cross-point to store a weight value. The word lines are used to apply input activations row-wise, while the bit lines collect output currents column-wise. The conductance state of each FeFET is determined by its ferroelectric polarization and serves as the stored weight during computation. Owing to its regular row–column topology, the crossbar architecture naturally supports the hardware mapping of matrix operations.

During computation, each FeFET is biased in the linear region, such that its channel current can be approximated as the product of the device conductance and the input voltage. In this way, each cell performs a multiplication between the stored weight and the applied input. Meanwhile, the currents of all cells in the same column are naturally summed along the corresponding bit line because of the parallel electrical connection. Therefore, multiplication is realized at the device level and accumulation is realized at the column level, enabling large-scale MAC operations within the array. By performing the computation directly in the memory array, the proposed architecture effectively reduces data movement and improves energy efficiency.

3.2. Hardware mapping of matrix operations

Matrix multiplication can be directly implemented in the FeFET crossbar through a mapping between mathematical variables and physical electrical quantities. Specifically, the elements of the weight matrix are encoded into the conductance states of the FeFET cells, while the input vector or matrix is applied to the word lines in the form of voltages. Under this mapping, the current generated by each FeFET cell represents the product of the corresponding input and weight.

Specifically, in the matrix multiplication $\mathrm{Y} = \mathrm{X} \cdot \mathrm{W}$, each element of the weight matrix W is mapped to the conductance value $\mathrm{G_{i,j}}$ of the corresponding FeFET device in the crossbar array, while the input matrix or input vector X is applied to the word lines in the form of voltages. For an input vector $\mathrm{X} = [\mathrm{V_1}, \mathrm{V_2}, \ldots, \mathrm{V_M}]$, each element is applied to the corresponding row of the array through its associated word line. Under this mapping scheme, the FeFET cell located at the i-th row and j-th column receives the input voltage $\mathrm{V_i}$ and generates an output current $\mathrm{I_{i,j}}$ according to its stored conductance $\mathrm{G_{i,j}}$. When the device operates in the linear region, the output current can be approximately expressed as the product of the input voltage and the conductance, i.e.,

$$I_{i,j} = G_{i,j} \cdot V_i \tag{1}$$

which realizes a single multiplication operation at the device level. Furthermore, all FeFET cells in the same column are connected in parallel along the bit line, so their output currents are naturally summed in the circuit, yielding the total output current of the j-th column:

DOI: 10.54254/2755-2721/2026.AD33666

$$I_{out,j} = \sum_{i=1}^{M} G_{i,j} \cdot V_i \tag{2}$$

At the physical level, this expression directly corresponds to the dot-product operation between the input vector and the j-th column of the weight matrix in matrix multiplication, namely the j-th component of the output vector Y. Therefore, for an $M \times N$ FeFET array, N output results can be obtained simultaneously in a single parallel operation, thereby enabling parallel matrix-vector multiplication.

3.3. Experimental demonstration of 3 × 3 matrix multiplication

To verify the feasibility of the proposed HZO FeFET-based CIM scheme, a 3 × 3 array was constructed and used to demonstrate matrix multiplication at the array level. To validate the proposed scheme, we consider the following matrix multiplication:

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \tag{3}$$

In this demonstration, the preceding matrix is treated as the activation matrix and the following matrix is treated as the weight matrix.

3.3.1. Array structure and matrix mapping

As shown in Fig. 4, this array has three word lines (WL_1–WL_3) and three bit lines (BL_1–BL_3). Nine FeFET cells are formed at the crossing points of these lines. The gate of each FeFET connects to the matching word line to receive input signals. The drain connects to the matching bit line to collect output current. The source is linked to ground and serves as a common reference point. This array structure creates a one-to-one match between physical FeFET cells and the elements in a 3×3 weight matrix. It allows matrix data to be mapped directly to the device array in hardware.

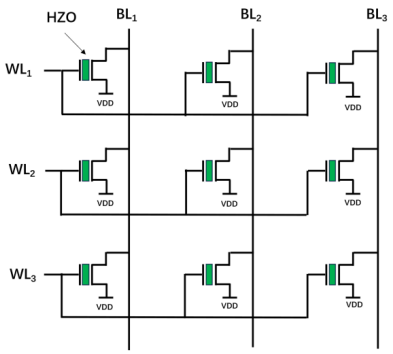

Figure 4. Schematic of the 3 × 3 FeFET computing array

3.3.2. Weight programming through polarization control

Before we start computing, we have to program the weight matrix into the array. In the method we put forward, we can finish this step by controlling the polarization state of each FeFET. As shown in

DOI: 10.54254/2755-2721/2026.AD33666

Fig. 5, we apply positive or negative programming pulses to the gate of every device based on the weight value we need. When a weight is set as logic "1," the corresponding FeFET is programmed to a state with high conductivity and a fairly low threshold voltage. When a weight is set as logic "0," the device is programmed to a state with weak conductivity or an OFF state and a fairly high threshold voltage. In this way, binary weights can be stored in the array without continuous power supply. Since the programmed state can be kept when there is no power, we do not need to load the weights over and over again in later computing work. This is a key advantage that helps reduce the extra cost caused by data movement.

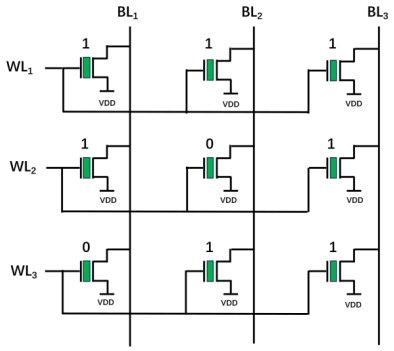

Figure 5. Polarization programming results of the FeFET

3.3.3. Row-wise input application

After programming the weights, we send the activation matrix to the array row by row through the word lines. As shown in Fig. 6, we apply the first row of the activation matrix [1, 1, 0] by setting the voltages of WL_1, WL_2 and WL_3 to 1 V, 1 V and 0 V separately. Under this input condition, each FeFET produces an output current, and the current is decided by both its stored conductance and the word-line voltage we use. We then reset the word lines to ground to clear the leftover voltage effects from the last round, and apply the second row [0, 1, 1] in the same way. At last, we apply the third row [0, 1, 0] following the same steps. We load each input row one after another and keep the stored weight states all the time, so the array can finish the whole matrix multiplication work.

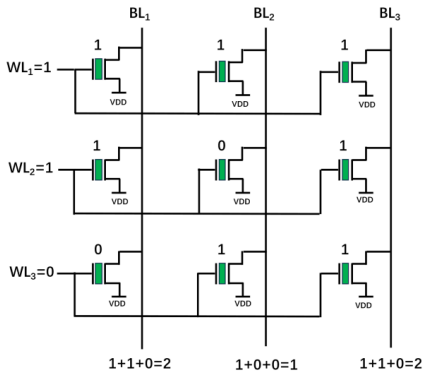

Figure 6. MAC operation results of the FeFET

DOI: 10.54254/2755-2721/2026.AD33666

3.3.4. In-array MAC operation

During the input use stage, all FeFET cells in the array take part in the calculation at the same time. Each cell does the multiplication work between the input voltage and the stored conductance state. At the same time, the bit-line currents are added up automatically by the array structure. So each bit-line current shows the total result from one part of the matrix output. By mixing device-level multiplication and column-level addition, the array can finish full MAC functions. It does not need extra independent calculation units.

3.3.5. Output result and correctness verification

After we process the three input rows one by one, the bit lines produce three groups of output current signals. These signals match the three rows of the output matrix. We can change these current signals into voltage or logic outputs with proper reading circuits. For example, we can use current-to-voltage conversion or threshold detection methods. Test results show that the output matrix we get is the same as the theoretical result of matrix multiplication. This proves that the HZO FeFET-based CIM method we put forward is correct. Besides, the output of the array can be sent directly to the next processing steps. This shows this structure can be used for continuous calculation. Because weight values are always stored in the array during the whole calculation, this method also has natural advantages in data placement and energy use efficiency.

4. Scalability analysis

The FeFET-array-based matrix calculation method we put forward has good scalability. We can expand this structure from a 3×3 array to an n×n array, and we only need to increase the number of word lines, bit lines and FeFET cells in the same proportion. The basic calculation rule will not change at all. In the bigger array, weights are still stored in FeFET cells without continuous power supply. Input signals are still sent through word lines, and output results are still obtained by adding up bit-line currents. So the same working principle can be kept for arrays of different sizes.

All cells in the array can work at the same time, so the total computing ability becomes stronger as the array gets larger. What's more, weights are stored in a nonvolatile way, so they can be used repeatedly in many calculation rounds. We do not need to spend extra resources on loading data again. These good features make this structure very suitable for large-scale matrix calculations, such as the calculations used in neural-network inference. So the HZO FeFET crossbar provides a scalable hardware solution for CIM systems with high parallel working ability and low power consumption.

5. Conclusion

This work introduced an HZO FeFET-based computing-in-memory structure for matrix multiplication, and it proved the feasibility of this design through array-level implementation and testing. At the device level, we analyzed the structural features and polarization-based storage mechanism of HZO FeFETs, which showed their ability to adjust conductance states in a nonvolatile way. At the architecture level, we built a FeFET crossbar-based CIM method. In this method, weight values were stored as device polarization states, input activation signals were sent through word lines, and MAC operations were completed by bit-line current summation.

DOI: 10.54254/2755-2721/2026.AD33666

Based on this framework, we designed and used a 3×3 HZO FeFET array to show matrix multiplication. Through polarization programming and row-wise input sending, we successfully finished in-array matrix computation. The experimental results were in line with the expected theoretical outputs, so they proved the correctness and feasibility of the proposed method for basic linear algebra operations. Besides, scalability analysis showed that the same computation principle can be extended to larger n×n arrays without changing the core working mechanism.

Overall, the proposed HZO FeFET-based CIM array provides a practical hardware way for highly parallel and energy-saving matrix computation. The results of this work indicate that HZO FeFET technology has great potential for AI-oriented CIM hardware. Future work may further add multibit weight storage, improve device uniformity, and design peripheral circuits together. These steps can help enhance computation precision and large-scale integration ability.

References

[1] M. Horowitz, "Computing's energy problem (and what we can do about it), " in IEEE Int. Solid-State Circuits Conf. (ISSCC), 2014.
[2] A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory devices and applications for in-memory computing, " Nature Nanotechnology, vol. 15, pp. 529–544, 2020.
[3] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey, " Proc. IEEE, vol. 105, no. 12, pp. 2295–2329, 2017.
[4] N. P. Jouppi et al., "In-datacenter performance analysis of a tensor processing unit, " in Proc. ISCA, 2017.
[5] Maurer, J. T., Ahmed, A. M. M., Khorrami, P., Moon, S. H., & Reis, D. A. (2025). A Survey on Computing-in-Memory (CiM) and Emerging Nonvolatile Memory (NVM) Simulators. Chips, 4(2), 19.
[6] S. Yu, "Neuro-inspired computing with emerging nonvolatile memories, " Proc. IEEE, vol. 106, no. 2, pp. 260–285, 2018.
[7] Y. Zhou et al., "TReCiM: Lower Power and Temperature-Resilient Multibit 2FeFET-1T Compute-in-Memory Design, " in IEEE/ACM ICCAD, 2024.
[8] P. Chi et al., "PRIME: A novel processing-in-memory architecture for neural network computation in ReRAM-based main memory, " in Proc. ISCA, 2016.
[9] M. Prezioso et al., "Training and operation of an integrated neuromorphic network based on metal-oxide memristors, " Nature, vol. 521, pp. 61–64, 2015.
[10] C. Li et al., "Analogue signal and image processing with large memristor crossbars, " Nature Electronics, vol. 1, pp. 52–59, 2018.
[11] H. Park, J.-G. Lee, and C. S. Hwang, "Review of ferroelectric field-effect transistors for three-dimensional storage applications, " Nano Select, vol. 2, no. 6, pp. 1230–1245, 2021.
[12] S. Dünkel et al., "A FeFET based super-low-power ultra-fast embedded NVM technology for 22 nm FDSOI, " in IEDM, 2017.
[13] S. Mueller et al., "Ferroelectricity in simple binary ZrO_2 and HfO_2, " Nano Letters, vol. 12, no. 8, 2012.
[14] T. Mikolajick, S. Slesazeck, and U. Schroeder, "Ferroelectric hafnium oxide for ferroelectric random-access memories and beyond, " MRS Bulletin, vol. 43, no. 5, pp. 340–346, 2018.
[15] G. Yuan, C. Wang, W. Tang, R. Zhang, and X. Lu, "Structure, performance regulation, and typical device applications of HfO_2-based ferroelectric thin films, " Acta Physica Sinica, vol. 72, no. 9, p. 097703, 2023.