FeFET Based In-Memory Hyperdimensional Encoding Design

Qingrong Huang, *Student Member, IEEE*, Zeyu Yang, *Student Member, IEEE*, Kai Ni, *Member, IEEE*, Mohsen Imani, *Member, IEEE*, Cheng Zhuo, *Senior Member, IEEE*, and Xunzhao Yin, *Member, IEEE*

Abstract*—The data explosion of Internet-of-Things (IoTs) and machine learning tasks raises a great demand on highly efficient computing hardware and paradigms. Brain-inspired hyperdimensional computing (HDC) is becoming a promising computing paradigm, which encodes data as hypervectors with homogeneous elements instead of numbers, and can perform learning/classification tasks through simple logical or arithmetic operations on the encoded hypervectors. Therefore HDC has much lower computational complexity than conventional computational models such as neural networks. However, due to its high-dimensional data representation, processing and encoding hypervectors in conventional Von-Neumann architectures (e.g., CPU and GPU) requires a large amount of energy- and time-consuming data transfer, thus weakening its efficiency benefiting from low complexity. In this paper, we proposed an ultra-low power and fast computing-in-Memory (CiM) design based on non-volatile (NV) Ferroelectric FET (FeFET) for HDC encoding. The proposed design mainly support hyperdimensional bit-wise XOR and parallel majority vote (MAJ) operations for HDC encoding, which are implemented by FeFET based memories together with CMOS peripheral circuits. The 1FeFET1T based memory cell effectively mitigates the impact of transistor variations on the operation. A highly parallel and pipelined computing workflow of the proposed design further boosts the energy efficiency and performance with negligible extra area overhead. Experimental results demonstrate that our proposed design achieves 5.04× energy efficiency improvement over other CiM designs for HDC encoding.

I. Introduction

Hyperdimensional computing (HDC) is an emerging cognitive computational framework based on the imitation of the behavior of human brains where key aspects of memory, perception and cognition can be explained by hyperdimensional space pattern activities [1]. Instead of processing numbers, HDC alogrithm exploits hypervector (HV) with thousands of homogeneous, independent and identically distributed (i.i.d.) dimensions (e.g., 10k) as the basic data unit. Since the dimensionality is at thousands, the ultra-long HVs are quasi-orthogonal, thus introducing algorithmic redundancy and robustness, and enabling high parallelism for HDC systems. The HDC computing model typically consists of training and inference phases. The training process firstly encodes the original training data into HVs, then combines such HVs into several class HVs using a set of well-defined operators. Unlike a conventional DNN that requires many iterations involving intensive forward and back propagation to converge the training, an unique advantage of HDC is that the training usually works in only one- or few- pass, and the dataflow is unidirectional. The inference is realized by checking the similarities between queries and class HVs. Comparable accuracy to DNN models has been demonstrated on multiple applications such as robotics, computer vision, speech detection, etc., [2]–[7].

As HDC mainly performs mostly bit-wise operations defined in hyperdimensional space such as binding (bit-wise XOR), bundling (bit-wise majority vote) and permutation over the HVs stored in memory [1], and is intrinsically robust to a few failures in HV dimensions due to the quasi-orthogonal property of HVs, recently, non-volatile (NV) device based in-memory computing (IMC) becomes a suitable and promising candidate for efficient HDC implementation. By specifically designing computational memory arrays (e.g., Associative memories (AM) [8]–[12]) or computing units located near memories, IMC enables highly parallel computations in-memory to alleviate the massive and costly data transfer between processor and memory, thus breaks the"memory wall" bottleneck [13]. Moreover, emerging NV memories (NVMs) such as resistive random access memory (ReRAM) further improve the memory density and the computational energy efficiency, while the computational errors induced by the limited storage precision and device variations of NVMs can be well tolerant by HDC framework, thus maintaining the IMC implementation accuracy. Many prior works have proposed memristor based IMC designs implementing the main operations of HDC, such as bit-wise XOR [14], majority vote (MAJ) [3], [15], addition [16], and pattern searching [17], [18]. Memristor based IMC HDC systems have also been fabricated [14], [19]. However, the two-terminal structure, low ON/OFF resistance ratio and current-driven write/read scheme of memristor devices lead to significant energy and area overheads associated with write and computation, which is hardly elucidated in prior works.

This work was supported in part by NSFC (92164203, 62104213), National Key Research and Development Program of China (2022YFB4400300), Zhejiang Provincial NSF (LQ21F040006, LD21F040003), SGC Cooperation Project (M-0612), Zhejiang Lab (2021MD0AB02).

Q. Huang and Z. Yang are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China.

K. Ni is with the Department of Electrical & Microelectronic Engineering, Rochester Institute of Technology, Rochester, USA.

M. Imani is with the Department of Computer Science at University of California Irvine, Irvine, USA.

C. Zhuo is with School of Micro-Nano Electronics, Zhejiang University, ZJU-Hangzhou Global Scientific and Technological Innovation Center and Key Laboratory of Collaborative Sensing and Autonomous Unmanned Systems of Zhejiang Province, Hangzhou, China. E-mail: czhuo@zju.edu.cn.

X. Yin is with the College of Information Science and Electronic Engineering, Zhejiang University, Key Laboratory of Collaborative Sensing and Autonomous Unmanned Systems of Zhejiang Province, and Zhejiang Lab, Hangzhou, China. E-mail: xzyin1@zju.edu.cn.

As a promising technology, ferroelectric field effect transistor (FeFET) has been fabricated and studied for IMC circuit designs [20]–[23]. Compared with the aforementioned IMC designs, FeFET based designs have exhibited higher energy efficiency and better write/read performance due to the three terminal structure, relatively high ON/OFF current ratio and voltage-driven write scheme, which make the FeFET a natural candidate for efficient HDC hardware. In this paper, we hereby propose a FeFET based IMC architecture which implements the key operations of HDC encoding based on FeFET memory arrays. We leverage a FeFET memory array to perform the HV binding and permutation, and another FeFET memory array to perform HV bundling which was hardly implemented using the NVM array in prior works. A pipeline interface between the two memory arrays is proposed, thus achieving the entire HDC encoding process. The major contributions of our paper are as follows:

- **A Complete In-Memory HDC Encoder**: We propose IMC designs that support binding, bundling and permutation, which are the major operations used for HDC encoding phase in both inference and training. The IMC designs consists of two FeFET based arrays that implement bit-wise XOR/permutation and MAJ, respectively, achieving higher energy efficiency than prior IMC based implementations.
- **Pipeline Interface for the FeFET based HDC Encoder**: Prior works mainly focus on proposing IMC designs for the HDC binding and bundling operations, while the interface between these designs is hardly studied. We hereby propose a novel charge based and pipeline write scheme. The proposed scheme hides the write latency of FeFETs by pipelining the intermediate HV result transfers between the proposed IMC arrays, thus realizing the interface between the proposed IMC arrays and achieving a complete HDC encoder.
- **Approximate MAJ operation for large scale HDC encoding**: Given the limited size of IMC array for MAJ operation, we demonstrate that iterative mini-batch MAJ operations implemented by multiple IMC arrays still resemble the original bundling operation of HDC encoding, with negligible hardware overhead and HDC algorithm accuracy degradation.

The rest of this paper is organized as follows, Sec. II provides background of this work, including HDC preliminaries, FeFET basics and related works. Sec. III introduces the FeFET based IMC HDC design. Sec. IV introduces the algorithm-level approximate MAJ operation method. Evaluations at both circuit level and system level are provided in Sec. V. Sec. VI concludes.

II. Background

A. FeFET Basics

The structure of FeFET is similar to a MOSFET, except that a ferroelectric (FE) layer is replacing the high-κ dielectric in the gate stack, as shown in Fig. 1(b)(d). The compatibility with CMOS VLSI technology brings great advantages of HfO_2 FeFET and makes it a promising candidate for novel memory

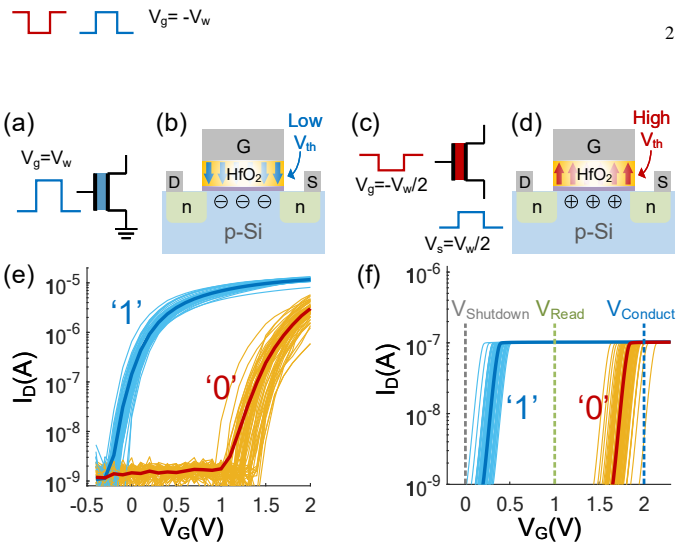


Fig. 1. (a)/(c) Write pulse for and (b)/(d) physical structure of FeFET in low/high threshold state, (e) $I_d - V_g$ characteristics of FeFET, (f) $I_d - V_g$ curves and the operating voltages of FeFET in serials with a resistor [24].

technologies and IMC designs. As the ferroelectricity in thin Si doped HfO_2 layer was demonstrated [25], the feasibility of integrating FeFET on advanced technology platforms has been further validated, such as the 28 nm bulk [26] and 22 nm FDSOI technology [27], accelerating the FeFET based IMC design.

The hysteresis of FE makes its threshold voltage (V_{th}) adjustable, in general, the binary encoded FeFET can be set to low-V_{th} state by simply applying write pulse with relatively a large amplitude $+V_W$ on its gate, as shown in Fig. 1(a)(b), and similarly reset to high-V_{th} state by a negative pulse. However, this direct write method may cause write disturbances when writing an FeFET array [28], as the FeFETs in a same row/col always have their gates/sources connected together. [29] proposed the disturb inhibition scheme "$\frac{1}{2}V_{GS/B}$" which applies opposite write pulses with half amplitude to both gate and source of the target FeFET as shown in Fig. 1(c), while for other FeFETs in the array, their gate-source voltages are not sufficient to change the polarization states due to the coercive effect of the HfO_2 FE layer [30], thus inhibiting write disturb to unselected cells. In this work, we combine the aforementioned write schemes and propose a novel charge based write method for the FeFET memory array, which is described in Sec. III-B.

Fig. 1(e) shows the experimental measured $I_D - V_{GS}$ curves of high-V_{th} and low-V_{th} states for 60 FeFET devices [24], read voltage V_{read} can be selected between the two V_{th} states, thus sensing the I_{DS} as read output, i.e., I_{ON} for low-V_{th} and I_{OFF} for high-V_{th}, respectively. However, the device process variation causes significant I_{DS} variation, which severely affects the read or computing results. In this work, we exploit a 1FeFET1R cell design proposed in [24], [31] and experimental demonstrated in [32] to alleviate the variability of I_{ON} by integrating a series resistor. The simulation characteristics of the 1FeFET1R cell using the experimental calibrated FeFET compact model [33] are shown in Fig. 1(f).

B. Hyperdimensional Computing

HDC is a light-weight and robust learning system [1], motivated by the understanding that the human brain operates

on *high-dimensional* representations of data originated from the large size of brain circuits [34]. It thereby models the human memory using points of a high-dimensional space, that is, with *hypervectors*. The hyperspace typically refers to tens of thousand dimensional vectors (e.g., $D = 10k$) with i.i.d. components. This indicates that the computation over different dimensions can be parallelized. Recently, HDC has been vastly applied to a wide range of learning problems [35], [36] HDC is well suited to perform learning tasks on emerging memory technologies as: (i) HDC has a memroy-centric architecture as well as simple arithmetic operations, thus it is computationally efficient to train and amenable to memory-centric hardware optimization [2], [3], and (ii) HDC provides strong robustness to noise, defect and hardware variations due to the fault tolerance of quasi-orthogonal hyper-dimensions, which is a critical advantage for NVM technologies.

The first step of HDC model is encoding input data into high-dimensional space. The two most commonly used encoding methods are record-based encoding and N-gram encoding [4]. Assume a feature vector $F = \{f_1, f_2, \cdots, f_m\}$ into is encoded high-dimensional space. The goal of HDC encoding is to keep the information of feature values along with their position in the feature vector. The record-based encoding generates two set of binary HVs: (i) position HVs, i.e., $\{\vec{P}_1, \vec{P}_2, \cdots, \vec{P}_m\}$, where $\vec{P}_i \in \{0,1\}^D$, (ii) feature HVs that can be computed by quantizing the feature values to q linear or non-linear levels, i.e., $\{\vec{L}_1, \vec{L}_2, \cdots, \vec{L}_q\}$, where $\vec{L}_i \in \{0,1\}^D$. For example, the pixel values of a black and white image (no gray color) can be quantized by two HVs ($q = 2$), and L_1 and L_2 represent the positions of black and white pixels, respectively. To this end, record-based encoding performs the encoding by associating each feature HV with the corresponding position HV, and then summing the results [37]:

$$\vec{H} = \vec{P}_1 \oplus \vec{L}'_1 + \vec{P}_2 \oplus \vec{L}'_2 + \cdots + \vec{P}_m \oplus \vec{L}'_m \qquad (1)$$

where $\oplus$ is an XOR operation and $\vec{L}'$s are the level HVs corresponding to the feature values. The N-gram-based encoding uses a unique permutation instead of position HVs to associate position information into encoding, the positions of features are reflected by the permutation number of the corresponding feature HVs. For example, a feature HV corresponding to the N^{th} feature in the feature vector can be permuted for $N-1$ times. The N-gram HV is thus formulated as below:

$$\vec{G} = \vec{L}'_1 \oplus \rho\vec{L}'_2 \oplus \rho^2\vec{L}'_3 \oplus \cdots \oplus \rho^{N-1}\vec{L}'_N \qquad (2)$$

where ρ represents permutation. The permutation function is usually implemented by (circular) shift hardware [4]. N is a preset parameter, whose values are typically 3, 4, 5 (i.e., Tri-gram, Quad-gram, Penta-gram). Generally, the length of feature vector m is always much larger than N (e.g., a text string in language recognition contains thousands of characters). The N-gram-based encoding firstly encodes all sub-vectors containing N adjacent elements of the original feature vector separately using Eq. 2, then the obtained N-gram HVs are further combined by addition to generate the HV representing the original feature vector. To keep the HV binary, the encoded HV is binarized by comparing each HV element value with $m/2$, which is defined as thresholding. Intuitively, the N-gram-based encoding pays attention to the patterns composing of adjacent elements, rather than the absolute positions of all elements.

In above encoding methods, XOR and Addition are the key functions. while XOR operation can be parallelized over different feature dimensions, the addition operation is a slow process. Prior work [16] implemented the encoding by sequentially adding the XOR outputs or using a tree based addition structure to obtain an non-binary output. Then the sum result is performed with thresholding operation to generate the output. Instead, we propose an in-memory MAJ logic to efficiently realize the addition and thresholding. The MAJ output is '1' when the number of '1's in MAJ inputs is more than the number of '0's, otherwise MAJ outputs '0'. Compared with the time-consuming addition and thresholding implementation in prior work, our proposed in-memory MAJ design can process parallel MAJ over all HV dimensions in one clock cycle.

C. Related Work

Various NVM based IMC designs for HDC encoding acceleration have been proposed. For example, [19] designed an in-memory bit-wise multiplication-addition-permutation (MAP) kernel based on a configurable 3-D vertical ReRAM pillar, achieving more than 2 orders of magnitude area efficiency and $\sim$2$\times$ energy efficiency improvement over digital counterparts due to the compact 3-D structure and in-memory logic design. Another ReRAM based work [16] fully exploits the connectivity of memristive crossbars to efficiently implement parallel in-memory logic as well as the dataflow. However, both works need to rewrite the target output device to store the output value during the in-memory computation, while the rewrite takes hundreds of nanosecond and consume up to picojoule levels of energy due to its current-driven write scheme [38]. [14] presents a phase change memory (PCM) based IMC design implementing the bit-wise XOR for N-gram-based HDC encoding, while the costly addition and thresholding operations are still implemented by digital peripherals. SearcHD [3] exploited multiple memristor based IMC arrays to implement the XOR and MAJ operations, while the time- and energy- consuming programming from the XOR IMC arrays into MAJ IMC array significantly limits the utilization of the IMC arrays, as well as the energy and area efficiency. FeFET based counterpart MIMHD [39] leveraged multi-bit storage FeFET to improve the density, and introduced analog-digital converters (ADCs) and shift registers to avoid the intermediate HVs write back, thus outperforms SearcHD in energy by $\sim$ 13$\times$. RelHD [40] extends the capability of HDC algorithm and hardware accelerator to graph computing. While the circuit design is similar to MIMHD, it has been enhanced with architectural techniques to process large data sets, including the memory allocation and calculation scheduling. However, both of them suffer from degraded parallelism and performance due to the costly ADC overheads. [15] presents FeFET based IMC designs that support single- and multi-input logic, including XOR similar to [20] and multi-input MAJ. However, these two IMC designs are only discussed at circuit level, the possibility of using these designs for HDC encoding, and the further reliability issues have rarely included.

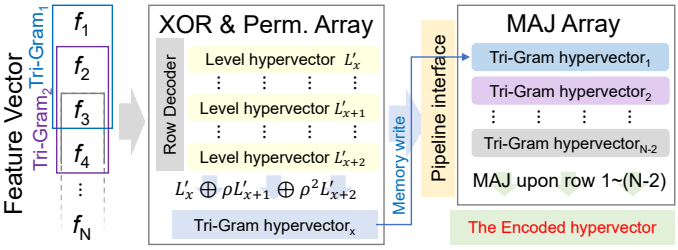
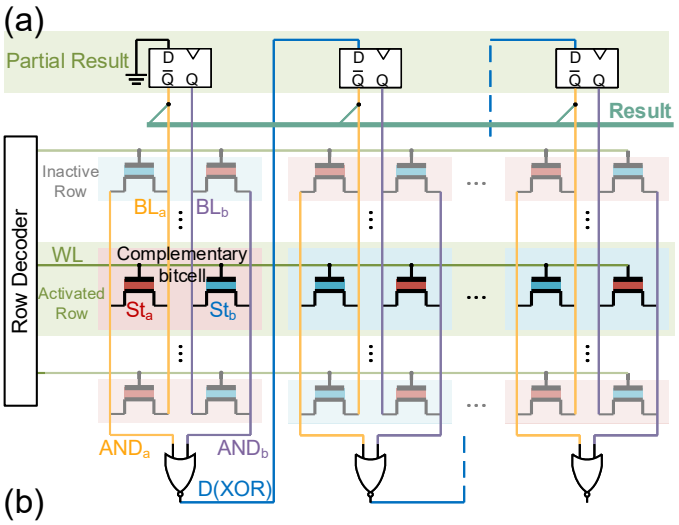


Fig. 2. Overview of the proposed IMC for HDC encoding design, tri-gram encoding are illustrated here as an example.

(a)



(b)

Stored Logic	St_a	St_b	Input Logic	BL_a (Q')	BL_b (Q)	AND_a	AND_b	D (XOR)
'0'	'0' high-vth	'1' low-vth	'0'	'1'	'0'	'0'	'0'	'0'
			'1'	'0'	'1'	'1'	'0'	'1'
'1'	'1' low-vth	'0' high-vth	'0'	'1'	'0'	'1'	'0'	'1'
			'1'	'0'	'1'	'0'	'0'	'0'

Fig. 3. (a) Schematic and (b) truth table of the proposed FeFET based in-memory XOR array.

[20], [41] both support in-memory XOR logic function. These designs select two rows as the input, and the XOR output is generated by comparing the bitline current of each column with a reference current, and then written back to the memory array for the HDC encoding. Since both designs support various logics and need intermediate result write back, complex sense amplifiers (SAs) are employed, and two-step operation schemes are required, thus leading to large delay and energy consumption. On the contrary, our proposed in-memory design performs XOR operation on a selected row with the external input word, and XOR logic gates and shift registers are exploited to generate the output and pass the intermediate results, which are more suitable for the sequential XOR operations in HDC encoding.

III. IMC Design for Hyperdimensional Encoding

In this section we present our proposed FeFET based IMC design for HDC encoding. Fig. 2 shows the overview of the design. It consists of 2 IMC arrays: the in-memory XOR array and the in-memory MAJ array. Besides, we propose a pipeline interface for the 2 arrays.

A. In-Memory XOR

The schematic of our proposed in-memory XOR array is depicted in Fig. 3(a), the basis HVs in Eq. 2 is stored row-wise in the FeFET memory array. In every clock cycle, the array computes the bit-wise XOR between the HV stored in the register and the HV in the activated array row, whose WordLine (WL) is set to high while the WLs of other rows are set to low. 2 FeFETs storing complementary values are exploited in our cell design (i.e., a bit value is mapped to the V_{th} of one FeFET in a cell, and its complementary value is mapped to the other). According to Fig. 1, when applying a read voltage on the device gate (e.g., $V_g = 1V$), an FeFET can perform bit-wise AND operation upon its stored state (St) (i.e., high/low V_{th} representing '0'/'1') and the voltage applied to the drain, generating an output current flowing through the drain-source path. To compute the XOR results, complementary voltages are applied to the drains of 2 FeFETs in a cell (i.e., BL_a and BL_b), therefore the minterms of XOR (i.e., $a \oplus b = \bar{a}b + a\bar{b}$) are generated at the device sources (i.e., AND_a and AND_b). An NOR gate is used to combine the AND terms. Fig. 3(b) shows the truth table of the FeFET based IMC XOR array. In our proposed IMC XOR array, all columns execute the bit-wise XOR operation within 1 clock cycle in parallel. The XOR output result bits are then latched in the register as the partial results of the N-gram HV formulation in Eq. 2.

Eq. 2 is implemented by repeatedly performing the above bit-wise XOR operation for N times. Upon the begin of N-gram encoding, the registers of the array are initiated to '0', and the HV bits of the activated row are read out to the registers as the first XOR operation. The rest (N-1) XOR operations are then performed between a HV stored in an activated array row and the partial results in the registers. Note that the output of the NOR gate in the i-th column is connected to the input of the $(i+1)$-th D-flip flop (DFF), such that the shift (permutation) operation in Eq. 2 can be naturally realized in the IMC XOR array. Compared with the IMC XOR design in [14], our proposed design significantly reduces the number of devices per cell to 2, therefore achieving nearly 50% area overhead saving. Moreover, our design computes accurate XOR operations over the HVs in the memory array, while the design in [14] implements an approximation for the minterms of the XOR operation.

B. In-Memory MAJ

Fig. 4 shows our proposed IMC MAJ array for HDC encoding. Different from the cell we employed in the IMC XOR array, the memory cell utilized in MAJ array (Fig. 4(a)) only contains 1 FeFET for 1-bit storage and a PMOS as an access transistor. A series resistor is integrated with the FeFET to alleviate the variability of FeFET ON current as described in Sec. II-A. The control signals including GroundLineEnable (GLE), WordLineFeFET (WLF), WordLineMOS (WLM), BitLine (BL) are summarized in Table. I. Such array writes the output HVs of IMC XOR array to the 1FeFET-1T memory array and performs bit-wise MAJ operations across the array once the write process completes. During the MAJ

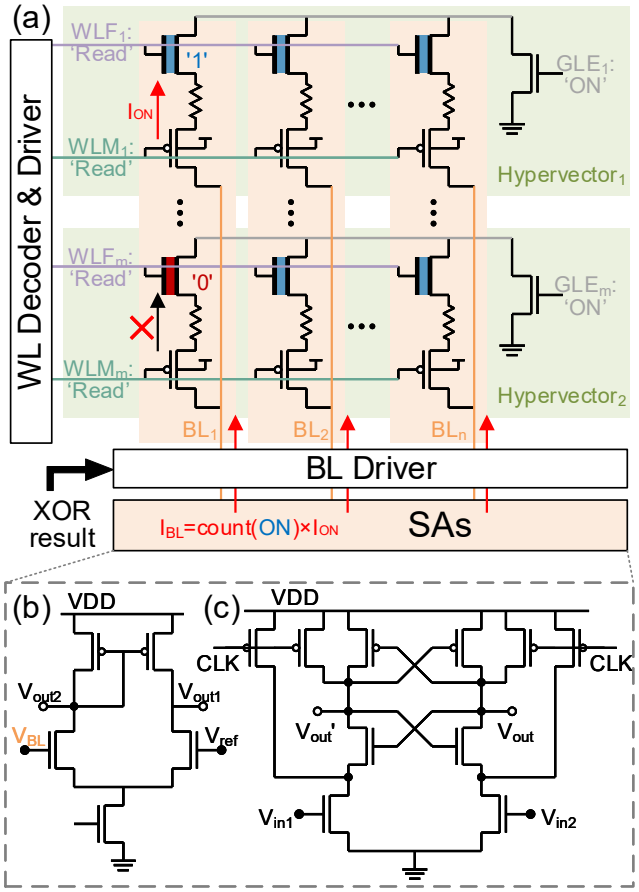


Fig. 4. (a) Overview of the proposed IMC MAJ array; (b)(c) the 2-stage sense amplifier, including a differential amplifier as buffer stage and a StrongARM latch [42] as a comparator.

TABLE I
SUMMARY OF THE MAIN CONTROL SIGNALS

Control Signals	Value	Definition
WLF	V_W/$\frac{V_W}{2}$	FeFET write Voltage
	$V_{conduct}$	Voltage that turns the FeFET ON, slightly larger than high-Vth, as shown in Fig. 1(f)
	$V_{shutdown}$	Voltage that turns the FeFET OFF, slightly lower than low-Vth, as shown in Fig. 1(f)
	V_{read}	FeFET read voltage
WLM	'ON'	Gate voltage that turns PMOS ON
	'OFF'	Gate voltage that turns PMOS OFF
BL	V_W/2	Half Write Voltage according to [29]
	GND	
	'Compute'	Bitline voltage for MAJ execution.
GLE	'ON'	Gate voltage that turns NMOS ON
	'OFF'	Gate voltage that turns NMOS OFF

computation, BLs are set to 'Compute', WLFs, WLMs and GLEs corresponding to unselected rows are 'OFF'. For the selected rows, WLMs are set to turn PMOS on, passing the voltage at BL to the FeFET source, WLFs are set to V_{read} to read the corresponding FeFET cells, GLEs are also set to 'ON'. The FeFET cells storing '1' value contribute I_{ON} to I_{BL} while the other cells only conduct leakage currents.

The column current I_{BL} is then converted to voltage V_{BL} by a load resistor, and sensed by the sense amplifier (SA). The 2-stage SA contains a differential amplifier as shown in Fig. 4(b) and a StrongARM Latch based comparator as shown in Fig. 4(c) [42]. Besides V_{BL} as one input of the differential

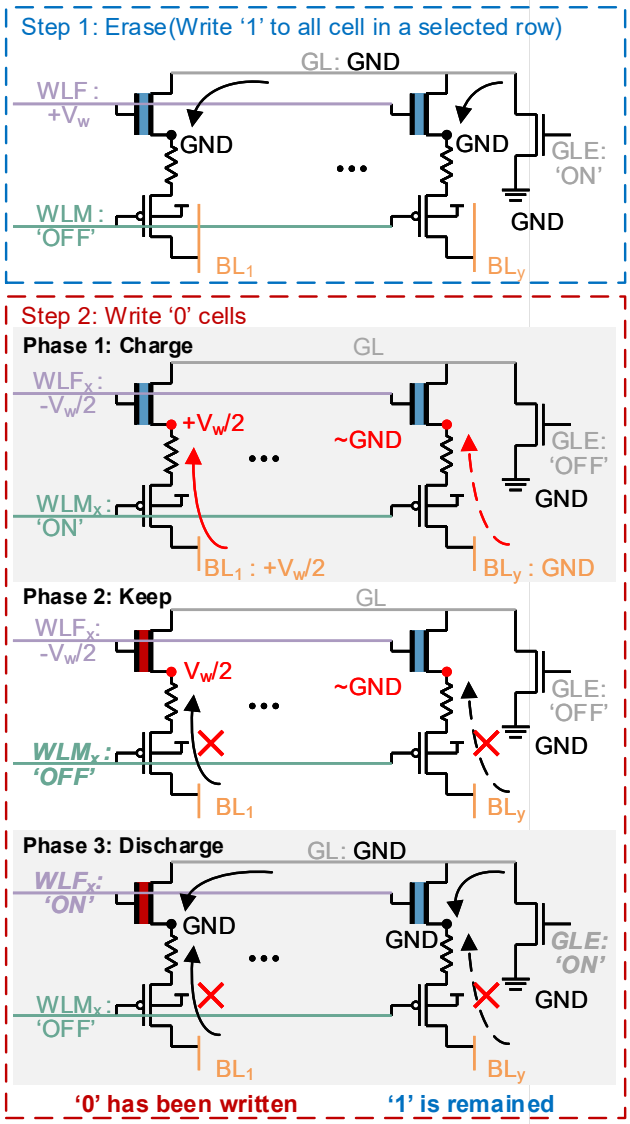


Fig. 5. The proposed charge based and pipeline write scheme.

amplifier, the other input is the reference voltage V_{ref}, which is generated by a reference column, storing the same number of '0's and '1's. V_{ref} is shared by all SAs, however, the dynamic CLK in StrongARM Latch based comparator may introduces significant kickback noise [42] to V_{BL} and V_{ref} due to parasitic capacitance, thus severely distorting V_{BL} and V_{ref} and inducing comparison errors. To tackle this problem, a differential amplifier as shown in Fig. 4(b) is added as a buffer stage. The differential amplifier can suppress the kickback noise by its open loop gain, and isolates the second stage strongARM Latch based comparator as shown in Fig. 4(c), thus ensuring the SA function and the parallelism of the MAJ array. The SA compares V_{BL} and V_{ref}, and the outputs V_{out} and V'_{out} indicate the MAJ result of corresponding column, i.e., V_{out}/V'_{out}='1'/'0' means MAJ output is '1', and vice versa.

C. Pipeline Interface for IMC arrays

With our proposed IMC XOR and MAJ arrays, writing the output HVs of XOR array is still time-consuming, as

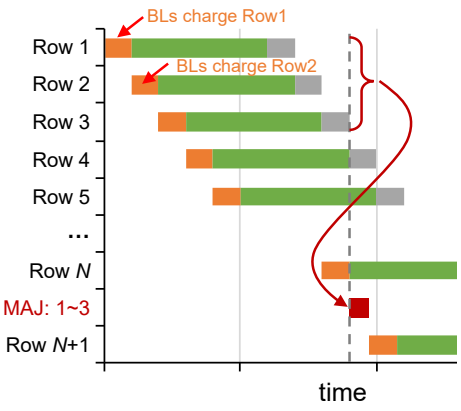


Fig. 6. Time diagram of the proposed charge based

programming the FeFET is much slower th
tion, causing speed mismatch between the
MAJ array, and thus the performance bottl
encoder. [3] directly implement parallel wr
the number of write drivers to shorten the
however, such method brings significant ex
overhead.

To address above problem, we propose
for the XOR array and the MAJ array, in
2-step charge based write scheme for writi
memory, as shown in Fig. 5. The first step
the selected row x to '1' (low-V_{th}) by si... write voltage $+V_W$ to WLF$_x$ and setting GLE$_x$ to 'ON' to ground the GroudLine(GL) and the FeFET sources, the WLM remains 'OFF' to keep PMOS off. Step 2 includes 3 phases: Charge, Keep, and Discharge. In Charge phase, WLM$_x$ of the selected row x is set to 'ON', and the BL voltages are passed to the sources of FeFETs, WLF$_x$ is set to $-\frac{1}{2}V_W$. For the cells to be written with '0', the corresponding BLs are set to $+\frac{1}{2}V_W$ such that the corresponding FeFET $V_{GS} = -V_W$. The BLs of other cells are connected to GND so that the voltage on FeFET sources can not exceed the PMOS threshold voltage $V_{th(PMOS)}$, and these cells remain '1'. In Keep phase, WLM$_x$ switches to 'OFF' to turn PMOS off, and the voltages on FeFET sources are retained till the write is complete. In phase 3, the sources of all FeFETs are discharged to GND by switching WLF$_x$ and GLE$_x$ to 'ON'.

Note that the BL voltages are passed to FeFET sources in phase 1, while in phase 2 and 3 WLM$_x$ turns the access PMOS off. Therefore the BLs can be then applied to other rows, indicating a pipeline of the 3 phases. Fig. 6 shows a workflow of our proposed pipeline interface, while the FeFET write time is hidden by the phase pipeline, and the equivalent write latency of a row is reduced to the duration of phase 1, which is much less than the FeFET write time. In this way, our charge based pipeline write method can match the operation speed of XOR array and significantly reducing the write time of XOR array, thus bringing great improvement in overall system performance.

IV. Approximate Mini-Batch MAJ

For most HDC learning tasks, the amount of learning data for HDC encoding can be more than millions, which is far more than the practical size of our proposed IMC arrays. Moreover, storing all the input data for encoding and performing a MAJ operation across the entire array is physically infeasible. In this section, we introduces Mini-Batch MAJ, which accommodates the large amount data associated with HDC encoding given the limited size of our proposed IMC designs.

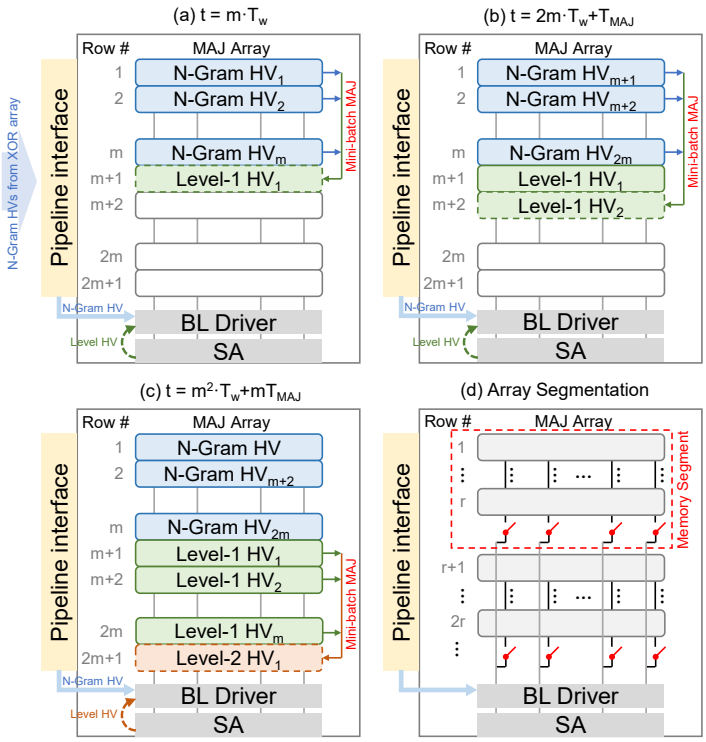


Fig. 7. (a-c)The workflow of the proposed mini-batch MAJ; (d) array segmentation for reducing BL charging energy.

To tackle above problems, we propose an iterative mini-batch MAJ method, which performs multiple and multi-level MAJ with a small number of inputs, to approximate the one shot MAJ operation over the entire learning data. Fig. 7(a-c) explain the workflow of our proposed method, where mini-batch MAJ with pre-selected input number m (i.e., batch size) is used for encoding a large amount of N-Gram HVs. MAJ$_m$ is used to refer to mini-batch MAJ operations with batch size m. (1) write m N-Gram HVs to the first m rows of the MAJ array one by one, then perform mini-batch MAJ across them to generate the level-1 HV$_1$ and write it back to the MAJ array (Fig. 7(a)); (2) write another m N-Gram HVs to the first m rows of the MAJ array (i.e., overwrite the m N-Gram HVs stored here which are written in the previous step) and perform mini-batch MAJ (Fig. 7(b)), repeat this step by $m-1$ times thus level-1 HV$_2$ $\sim$ HV$_m$ are obtained; (3) perform mini-batch MAJ across the m level-1 HVs to generate a level-2 (Fig. 7(c)); (4) repeat step (1)(2)(3), until all N-Gram HVs are processed, and once the number of level-l HVs reaches m, we perform a mini-batch MAJ to generate a level-$(l+1)$ HVs. The highest level HV is the approximation of encoding result.

Given the total number of N-Gram HVs Y, the number of levels is $log_m(Y)$, and each level occupies m rows of the memory array, thus the total memory size required by mini-batch MAJ method is only $mlog_m(Y)$. For example,

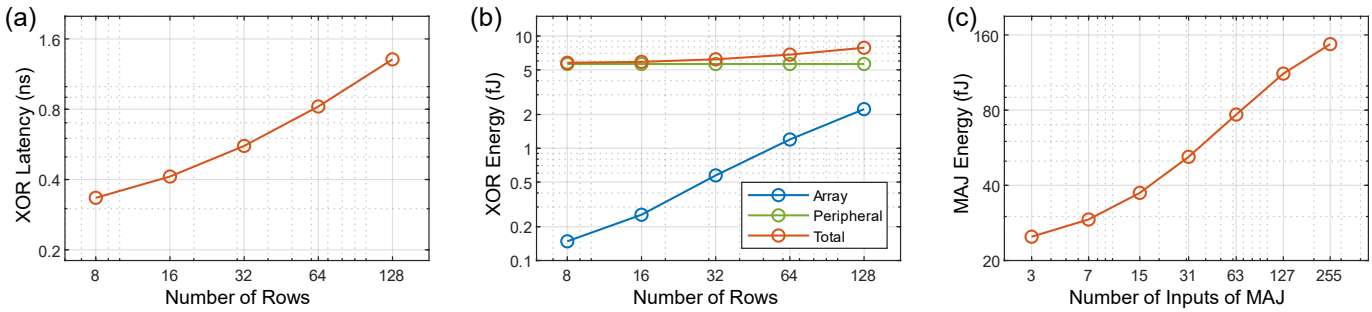


Fig. 8. (a) Latency and (b) energy of the proposed in-memory XOR array; (c) energy of the proposed In-memory MAJ array.

encoding 10000 N-Gram HVs using the conventional one shot MAJ needs a 10000-row array, while the proposed mini-batch MAJ method with a batch size m=100 only needs 200 rows. therefore the proposed method can significantly alleviate the memory overhead.

However, writing a HV to a row needs to charge BLs and the overall charging energy for writing an array is proportional to the number of rows. Storing multiple level HVs may still consume a large array, thus causing high write energy. To further improve the energy efficiency, we partition the MAJ array into several segments, as shown in Fig. 7(d), where every segment has local BLs, and is connected to the global BLs through switches. Only one memory segment rather than the entire array is charged in a HV write, thus significantly reducing the write energy.

It should be noted that generally the iterative mini-batch MAJ is only needed in the training phase, where a large amount of training objects are encoded and combined together. During the inference, the number of N-Grams in a query is relatively small (e.g., 152 in average in language identification task), thus all N-Gram HVs from one inference query can be directly processed by one MAJ operation using our IMC MAJ array.

V. Evaluation

In this section, we evaluate our proposed designs at both circuit level and system level, and compare the results with the state-of-the-art in-memory HDC accelerators [14], [16], [19].

A. Experimental Setup

We perform SPICE simulations on the proposed IMC XOR and MAJ arrays using Cadence Spectre Simulator. Both arrays are built using a compact multi-domain FeFET model [33] and the 40nm MOSFET based on the Predictive Technology Model (PTM). The FeFET write pulse V_W is 4V and the FeFET threshold voltage V_{th} variation is extracted from data in Fig 1(e), the standard deviation is 0.059V/0.14V for low-/high-V_{th}, and the resistor value refers to [24]. Furthermore, we build a cycle-accurate and variation-aware simulator, which can perform Monte Carlo simulations with the circuit level variation model and support system level energy calculation, for evaluating the robustness and performance of our design. A typical application of HDC—European language identification task is exploited for algorithm/system-level simulation.

B. Circuit-Level Evaluation

We first evaluate the IMC XOR array. The latency of the array is measured under the worst case, where a 3-sigma deviation on FeFET V_{th} is considered, thus the FeFET with the smallest conductance corresponds to the latency. The results in Fig. 8(a) show that the computation latency increases linearly with the number of rows of the array. This is because that the capacitance of BLs include the source capacitance of all FeFETs in the same column, thus the charge time t_{charge} is depicted as below:

$$t_{charge} \propto R_{ON} \times C_{SL} = R_{ON} \times (m \times C_{source}) \quad (3)$$

where R_{ON} is the equivalent resistance of FeFET with 'ON' state and m is the number of rows of array. The energy results of a single XOR operation are measured and shown in Fig. 8(b). The total energy increases linearly with the number of rows. This is because that the array energy, which is dominated by the BLs charging, is proportional to the number of rows, while the peripheral energy remains almost constant.

Next we evaluate the latency and energy consumption of MAJ operations based on a 255-row IMC MAJ array with varying numbers of inputs. The latency of MAJ is almost constant, as the latency is dominated by charging the parasitic capacitance of cells. The energy consumption increases with the number of activated MAJ inputs as shown in Fig. 8, which is due to the increase of bitline current I_{BL}.

We then analyze the robustness of our proposed IMC MAJ design considering the following hardware variations: (1) Memory variation including the FeFET variation and the resistor variation. Due to the large resistor value, the effect of FeFET variation can be negligible [24]. The resistor variation deviates the bitline current I_{BL} from its expected value, and may cause the MAJ error. We hereby verify the robustness to resistor variation of the MAJ design by initiating Monte Carlo simulation. The worst input patterns are considered, where the number of '1's in the MAJ inputs is closest to half of the total number of MAJ inputs (e.g., 127 '1's and 128 '1's are the worst input patterns for MAJ255=0 and MAJ255=1, respectively). These worst input patterns differ by I_{ON}, while the deviation of I_{BL} caused by device variation increases as the MAJ input size grows. Therefore, as can be shown in Fig. 9(a-t), MAJ with larger input size has low tolerance to device variation. The scalability of our design is also studied in Fig. 9(a-t). It can be seen that our proposed MAJ array can accurately execute MAJ operation with input size up to 255 under 1% resistor variation (Fig. 9(q)). When variation

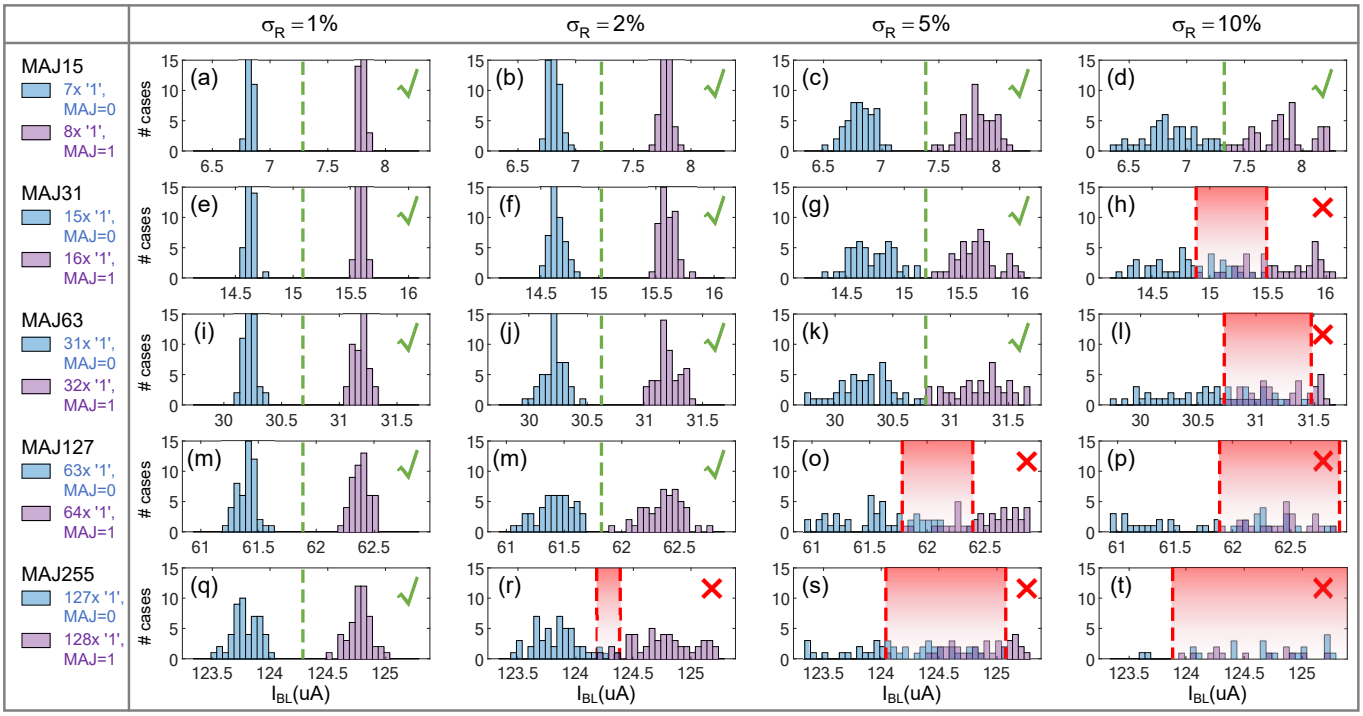


Fig. 9. I_{BL} distributions from 50 Monte Carlo runs for the worst input pattern of MAJ, with different input sizes and σ_R.

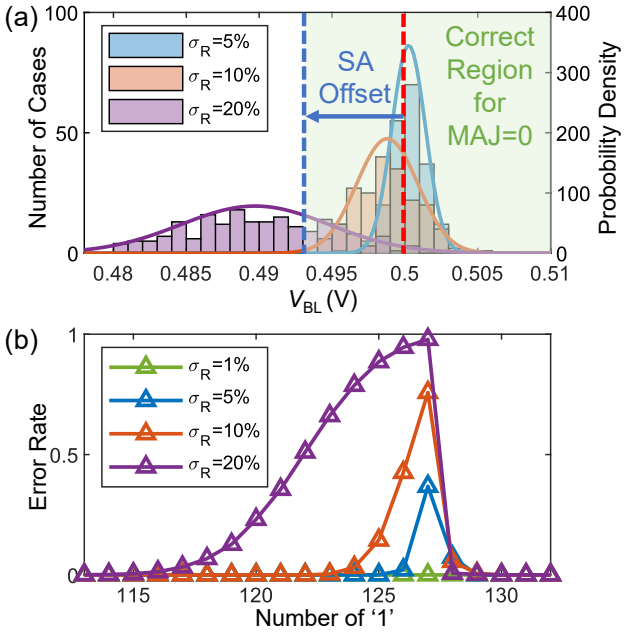


Fig. 10. (a) MAJ array bitline voltage distribution under different resistor variation levels; (b) error rate of MAJ255 with different input pattern.

is larger (e.g., 10%), our design can still at least support MAJ15 (Fig. 9(d)). Fig. 10(a) shows the distributions of bitline voltage V_{BL} under different resistor variation, and shows how variation causes the MAJ error. The error rates of MAJ255 in Fig.10(b) indicate that the variation mainly affects the MAJ operations whose input patterns is closer to the worst case.

(2) The SA offset which can be considered as the overall effects of the imperfections associated with SA such as PVT variations. It can be equivalent to an offset of reference voltage V_{ref} (shown by the dotted blue line in Fig. 10(b)), and it causes MAJ error especially when V_{BL} is close to V_{ref}.

Though the hardware variations may cause the MAJ operation error, however, in the following subsection we will show that partially incorrect MAJ operations due to the variations are still tolerable for HDC algorithms.

TABLE II
PERFORMANCE COMPARISON OF THE PROPOSED IMC ARRAYS WITH STATE-OF-THE-ART IMC DESIGN.

Metrics	XOR		MAJ3	
	Latency(ns)	Energy(fJ)	Latency(ns)	Energy(fJ)
The proposed	0.557	6.21	1.70	24.9
FELIX [16]	∼3	34.97	∼3	65.65
Improvement	∼5.39×	5.63×	∼1.76×	2.64×
PCM work [14]	2.8	9.8	/	/
Improvement	5.03×	1.58×	/	/

Table II concludes the performance metrics of our proposed IMC arrays, with comparison with state-of-the-art IMC designs for HDC [14], [16]. For XOR operation, our design achieves 4.85× to 5.39× and 1.58× to 5.63× improvements in latency and energy efficiency, respectively. For MAJ3 operation, our design with a 255-row MAJ array can also outperform FELIX [16] by 1.76 × in latency and 2.64× in energy, respectively. Moreover, our design can support MAJ operations with varying input numbers (e.g., MAJ16 ∼ MAJ255), obtaining more improvements.

C. System-Level Evaluation

We first measure the impact of our approximate mini-batch MAJ on inference accuracy, the results are listed in table III. It can be seen that the impact of our proposed mini-batch MAJ method on the inference accuracy is negligible. Note that when the MAJ batch size is 255, the corresponding mini-batch MAJ method only contains 256 MAJ operations and achieves

TABLE III
ACCURACY OF HDC USING MINI-MAJ METHOD UNDER DIFFERENT MAJ INPUTS SIZE.

Batch size		Baseline[1]	3	7	15	31	63	127	255
Number of MAJ levels		/	12	7	5	4	3	2	2
Number of MAJ operations		/	266k	137k	54.2k	30.7k	4.03k	128	256
The amount of N-Grams		1M	531k	824k	759k	923k	250k	16.1k	65.0k
Accuracy	Tri-Gram, D=2000	94.45%	94.06%	94.37%	94.33%	94.62%	94.46%	93.40%	94.17%
	Quad-Gram, D=10000	97.16%	97.05%	97.09%	97.25%	97.21%	97.14%	96.29%	96.83%

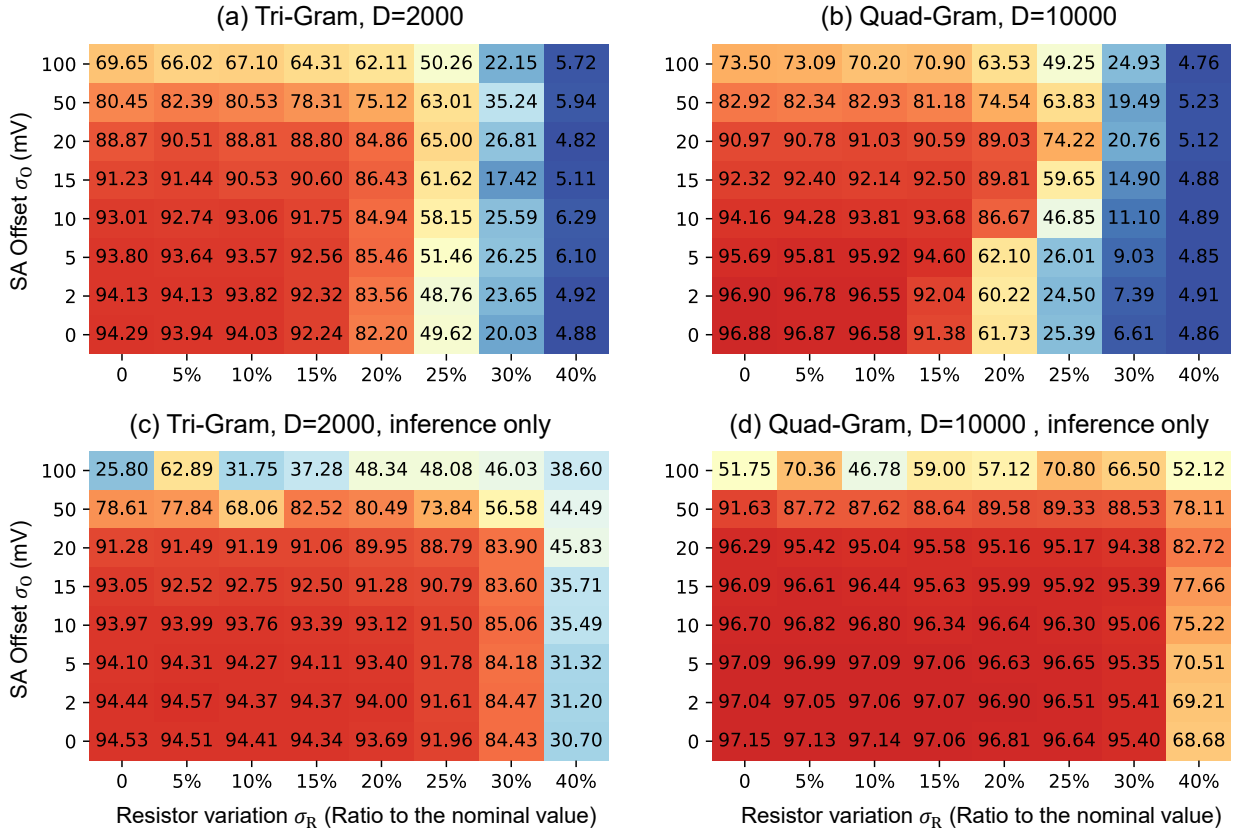


Fig. 11. The accuracy of language identification task using our proposed design for (a)(b) both training and inference, and (c)(d) for inference only, considering various resistor variations levels and SA offsets.

comparable accuracy to the baseline, while the number of N-grams and accumulation operations needed for encoding reaches 1 million. We hereby use this batch size 255 in the evaluation.

With the aforementioned hardware variation at circuit level, we then evaluate the robustness of our proposed IMC encoder design at system/algorithm level in the context of HDC language identification tasks. MAJ255 is employed, and 2-level approximate mini-batch MAJ method is applied to encode the training data (i.e., 65025 HVs). The encoder design is employed in both training and inference, and the identification task accuracy results are shown in Fig.11(a)(b), which represent 2 typical N-Gram based HDC methods: Tri-Gram with $D = 2000$ and Quad-Gram with $D = 10000$, respectively [2], [3]. The results shows that our proposed IMC encoder can achieve less than 1% accuracy loss with resistor variation up to 10% and SA offset standard deviation up to 5mV. More variation and offset can be tolerated if further relaxing the accuracy (e.g., $\sim$5% accuracy loss with resistor variation up to 15% and SA offset up to 15mV standard deviation). Moreover, when our encoder is employed for HDC inference only (i.e., the accurate trained model is used), the accuracy can be improved, resulting in less than 1% loss with up to $15\% \sim 20\%$ resistor variation and $15 \sim 20$mV SA offset, as shown in Fig. 11(c)(d).

TABLE IV
COMPARISON OF ENERGY EFFICIENCY OF THE PROPOSED DESIGN WITH STATE-OF-THE-ART IMC HDC ACCELERATOR.

Design	NV devices	Energy per Query(nJ)	Normalized Energy
Proposed (Quad-Gram, D=10000)	FeFET	85.39	1×
[14]	PCM	430.3	5.04×
[19]	RRAM	318000	3724×

We further evaluate the energy efficiency of HDC query encoding implemented by our IMC design, and compare it with state-of-the-art IMC accelerators for HDC [14], [19]. The results are listed in Table IV. It can be seen that by fulling utilizing the 3-terminal structure and energy efficient

voltage driven read/write scheme of FeFET, our proposed IMC HDC encoder, which consists of a compact XOR design and a MAJ design that supports single cycle efficient MAJ operation, achieves 5.04× and 3724× energy efficiency improvements compared with PCM based [14] and RRAM based [19] counterpart, respectively.

VI. Conclusion

In this paper, we proposed an efficient IMC design for fast and energy-efficient HDC encoding. Exploiting the unique characteristics of FeFET, 2 IMC arrays are designed for accelerating the binding, permutation and bundling operations of HDC. The 2-FeFET storage cell based XOR array improves the data density, and the in-memory MAJ array supports costly accumulation and thresholding by performing a single-cycle MAJ operation. A highly pipelined write scheme maximizes the utilization of the proposed designs thus boosts the performance. Evaluations of our designs as well as application benchmarking demonstrate that our design outperforms other IMC based counterparts at least 1 order of magnitude in energy efficiency, with excellent robustness to hardware variation and negligible accuracy loss.

References

[1] P. Kanerva, "Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors," *Cognitive Computation*, vol. 1, no. 2, pp. 139–159, 2009.

[2] A. Rahimi, P. Kanerva, and J. M. Rabaey, "A Robust and Energy-Efficient Classifier Using Brain-Inspired Hyperdimensional Computing," in *Proceedings of the 2016 International Symposium on Low Power Electronics and Design*. ACM, 2016, pp. 64–69.

[3] M. Imani, X. Yin, J. Messerly, S. Gupta, M. Niemier, X. S. Hu, and T. Rosing, "SearcHD: A Memory-Centric Hyperdimensional Computing With Stochastic Training," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2422–2433, 2020.

[4] L. Ge and K. K. Parhi, "Classification using Hyperdimensional Computing: A Review," *IEEE Circuits and Systems Magazine*, vol. 20, no. 2, pp. 30–47, 2020.

[5] Y. Ni, M. Issa, D. Abraham, M. Imani, X. Yin, and M. Imani, "Hdpg: Hyperdimensional policy-based reinforcement learning for continuous control," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, 2022, pp. 1141–1146.

[6] P. Poduval, Z. Zou, X. Yin, E. Sadredini, and M. Imani, "Cognitive correlative encoding for genome sequence matching in hyperdimensional system," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 781–786.

[7] A. Hernández-Cano, C. Zhuo, X. Yin, and M. Imani, "Reghd: Robust and efficient regression in hyper-dimensional learning system," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 7–12.

[8] X. Yin, Y. Qian, M. Imani, K. Ni, C. Li, G. L. Zhang, B. Li, U. Schlichtmann, and C. Zhuo, "Ferroelectric ternary content addressable memories for energy efficient associative search," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022.

[9] J. Cai, M. Imani, K. Ni, G. L. Zhang, B. Li, U. Schlichtmann, C. Zhuo, and X. Yin, "Energy efficient data search design and optimization based on a compact ferroelectric fet content addressable memory," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, 2022, pp. 751–756.

[10] R. Karam, R. Puri, S. Ghosh, and S. Bhunia, "Emerging Trends in Design and Applications of Memory-Based Computing and Content-Addressable Memories," *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1311–1330, 2015.

[11] A. Kazemi, M. M. Sharifi, A. F. Laguna, F. Müller, X. Yin, T. Kämpfe, M. Niemier, and X. S. Hu, "Fefet multi-bit content-addressable memories for in-memory nearest neighbor search," *IEEE Transactions on Computers*, vol. 71, no. 10, pp. 2565–2576, 2021.

[12] X. S. Hu, M. Niemier, A. Kazemi, A. F. Laguna, K. Ni, R. Rajaei, M. M. Sharifi, and X. Yin, "In-memory computing with associative memories: a cross-layer perspective," in *2021 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2021, pp. 25–2.

[13] W. A. Wulf and S. A. McKee, "Hitting the memory wall: Implications of the obvious," *ACM SIGARCH computer architecture news*, vol. 23, no. 1, pp. 20–24, 1995.

[14] G. Karunaratne, M. Le Gallo, G. Cherubini, L. Benini, A. Rahimi, and A. Sebastian, "In-memory hyperdimensional computing," *Nature Electronics*, vol. 3, no. 6, pp. 327–337, 2020.

[15] Q. Huang, D. Reis, C. Li, D. Gao, M. Niemier, X. S. Hu, M. Imani, X. Yin, and C. Zhuo, "Computing-In-Memory Using Ferroelectrics: From Single- to Multi-Input Logic," *IEEE Design Test*, vol. 39, no. 2, pp. 56–64, 2022.

[16] S. Gupta, M. Imani, and T. Rosing, "FELIX: Fast and energy-efficient logic in memory," in *Proceedings of the International Conference on Computer-Aided Design*. IEEE, 2018, pp. 1–7.

[17] K. Ni, X. Yin, A. F. Laguna, S. Joshi, S. Dünkel, M. Trentzsch, J. Müller, S. Beyer, M. Niemier, X. S. Hu, and S. Datta, "Ferroelectric ternary content-addressable memory for one-shot learning," *Nature Electronics*, vol. 2, no. 11, pp. 521–529, 2019.

[18] X. Yin, C. Li, Q. Huang, L. Zhang, M. Niemier, X. S. Hu, C. Zhuo, and K. Ni, "FeCAM: A Universal Compact Digital and Analog Content Addressable Memory Using Ferroelectric," *IEEE Transactions on Electron Devices*, vol. 67, no. 7, pp. 2785–2792, 2020.

[19] H. Li, T. F. Wu, A. Rahimi, K.-S. Li, M. Rusch, C.-H. Lin, J.-L. Hsu, M. M. Sabry, S. B. Eryilmaz, J. Sohn, W.-C. Chiu, M.-C. Chen, T.-T. Wu, J.-M. Shieh, W.-K. Yeh, J. M. Rabaey, S. Mitra, and H.-S. P. Wong, "Hyperdimensional computing with 3D VRRAM in-memory kernels: Device-architecture co-design for energy-efficient, error-resilient language recognition," in *2016 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2016, pp. 16.1.1–16.1.4.

[20] D. Reis, M. Niemier, and X. S. Hu, "Computing in memory with fefets," in *Proceedings of the International Symposium on Low Power Electronics and Design*. ACM, 2018, pp. 1–6.

[21] X. Chen, X. Yin, M. Niemier, and X. S. Hu, "Design and optimization of fefet-based crossbars for binary convolution neural networks," in *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*. IEEE, 2018, pp. 1205–1210.

[22] M. Lee, W. Tang, B. Xue, J. Wu, M. Ma, Y. Wang, Y. Liu, D. Fan, V. Narayanan, H. Yang, and X. Li, "FeFET-based low-power bitwise logic-in-memory with direct write-back and data-adaptive dynamic sensing interface," in *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*. ACM, 2020, pp. 127–132.

[23] C.-K. Liu, H. Chen, M. Imani, K. Ni, A. Kazemi, A. F. Laguna, M. Niemier, X. S. Hu, L. Zhao, C. Zhuo, and X. Yin, "Cosime: Fefet based associative memory for in-memory cosine similarity search," in *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, 2022, pp. 1–9.

[24] T. Soliman, F. Müller, T. Kirchner, T. Hoffmann, H. Ganem, E. Karimov, T. Ali, M. Lederer, C. Sudarshan, T. Kämpfe, A. Guntoro, and N. Wehn, "Ultra-low power flexible precision fefet based analog in-memory computing," in *2020 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2020, pp. 29.2.1–29.2.4.

[25] T. Böscke, J. Müller, D. Braeuhaus, U. Schroeder, and U. Bottger, "Ferroelectricity in Hafnium Oxide Thin Films," *Applied Physics Letters*, vol. 99, no. 10, pp. 102 903.1–102 903.3, 2011.

[26] M. Trentzsch, S. Flachowsky, R. Richter, J. Paul, B. Reimer, D. Utess, S. Jansen, H. Mulaosmanovic, S. Müller, S. Slesazeck *et al.*, "A 28nm hkmg super low power embedded nvm technology based on ferroelectric fets," in *2016 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2016, pp. 11–5.

[27] S. Dünkel, M. Trentzsch, R. Richter, P. Moll, C. Fuchs, O. Gehring, M. Majer, S. Wittek, B. Müller, T. Melde *et al.*, "A fefet based super-low-power ultra-fast embedded nvm technology for 22nm fdsoi and beyond," in *2017 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2017, pp. 19–7.

[28] M. Ullmann, H. Goebel, H. Hoenigschmid, and T. Haneder, "Disturb free programming scheme for single transistor ferroelectric memory arrays," *Integrated Ferroelectrics*, vol. 34, no. 1-4, pp. 155–164, 2001.

[29] D. Reis, K. Ni, W. Chakraborty, X. Yin, M. Trentzsch, S. D. Dünkel, T. Melde, J. Müller, S. Beyer, S. Datta, M. T. Niemier, and X. S. Hu, "Design and analysis of an ultra-dense, low-leakage, and fast fefet-based random access memory array," *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 5, no. 2, pp. 103–112, 2019.

[30] K. Ni, P. Sharma, J. Zhang, M. Jerry, J. Smith, K. Tapily, R. Clark, S. Mahapatra, and S. Datta, "Critical role of interlayer in hf0.5zr0.5o2

ferroelectric fet nonvolatile memory performance," *IEEE Transactions on Electron Devices*, vol. 65, no. 6, pp. 2461–2469, 2018.

[31] X. Yin, F. Müller, Q. Huang, C. Li, M. Imani, Z. Yang, J. Cai, M. Lederer, R. Olivo, N. Laleni *et al.*, "An ultra-compact single fefet binary and multi-bit associative search engine," *arXiv preprint arXiv:2203.07948*, 2022.

[32] D. Saito, T. Kobayashi, H. Koga, N. Ronchi, K. Banerjee, Y. Shuto, J. Okuno, K. Konishi, L. Di Piazza, A. Mallik, J. Van Houdt, M. Tsukamoto, K. Ohkuri, T. Umebayashi, and T. Ezaki, "Analog in-memory computing in fefet-based 1t1r array for edge ai applications," in *2021 Symposium on VLSI Circuits*. IEEE, 2021, pp. 1–2.

[33] K. Ni, M. Jerry, J. A. Smith, and S. Datta, "A circuit compatible accurate compact model for ferroelectric-fets," in *2018 IEEE Symposium on VLSI Technology*. IEEE, 2018, pp. 131–132.

[34] B. Babadi and H. Sompolinsky, "Sparseness and expansion in sensory representations," *Neuron*, vol. 83, no. 5, pp. 1213–1226, 2014.

[35] A. Mitrokhin, P. Sutor, C. Fermüller, and Y. Aloimonos, "Learning sensorimotor control with neuromorphic sensors: Toward hyperdimensional active perception," *Science Robotics*, vol. 4, no. 30, p. eaaw6736, 2019.

[36] M. Imani, Y. Kim, S. Riazi, J. Messerly, P. Liu, F. Koushanfar, and T. Rosing, "A framework for collaborative learning in secure high-dimensional space," in *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*. IEEE, 2019, pp. 435–446.

[37] M. Imani, C. Huang, D. Kong, and T. Rosing, "Hierarchical hyperdimensional computing for energy efficient classification," in *Proceedings of the 55th Annual Design Automation Conference*. ACM, 2018, pp. 1–6.

[38] H. Jeong and L. Shi, "Memristor devices for neural networks," *Journal of Physics D: Applied Physics*, vol. 52, no. 2, p. 023003, 2019.

[39] A. Kazemi, M. M. Sharifi, Z. Zou, M. Niemier, X. S. Hu, and M. Imani, "MIMHD: Accurate and Efficient Hyperdimensional Inference Using Multi-Bit In-Memory Computing," in *2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. IEEE, 2021, pp. 1–6.

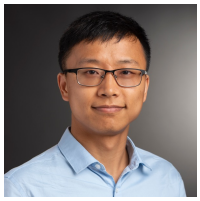
[40] J. Kang, M. Zhou, A. Bhansali, W. Xu, A. Thomas, and T. Rosing, "RelHD: A Graph-based Learning on FeFET with Hyperdimensional Computing," in *2022 IEEE 40th International Conference on Computer Design (ICCD)*, Oct. 2022, pp. 553–560.

[41] M. Lee, W. Tang, B. Xue, J. Wu, M. Ma, Y. Wang, Y. Liu, D. Fan, V. Narayanan, H. Yang, and X. Li, "Fefet-based low-power bitwise logic-in-memory with direct write-back and data-adaptive dynamic sensing interface," in *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, 2020, p. 127–132.

[42] B. Razavi, "The strongarm latch [a circuit for all seasons]," *IEEE Solid-State Circuits Magazine*, vol. 7, no. 2, pp. 12–17, 2015.

Qingrong Huang received his B.S. degree in electronic science and technology from Zhejiang University in 2020. He is currently pursuing a Ph.D. degree with the College of Information Science and Electronic Engineering, Zhejiang University. His current research interests include in-memory computing circuit and architecture designs and brain-inspired computing frameworks.

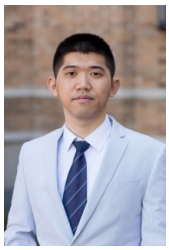
Zeyu Yang received the B.S. degree in electronic science and technology from Zhejiang University in 2021. He is currently pursuing a Ph.D. degree with the College of Information Science and Electronic Engineering, Zhejiang University. His current research interests include design and optimization of circuits & architectures for in-memory computing.



Kai Ni received the B.S. degree in Electrical Engineering from University of Science and Technology of China, Hefei, China in 2011, and Ph.D. degree of Electrical Engineering from Vanderbilt University, Nashville, TN, USA in 2016 by working on characterization, modeling, and reliability of III-V MOSFETs. Since then, he became a postdoctoral associate at University of Notre Dame, working on ferroelectric devices for nonvolatile memory and novel computing paradigms. He is now an assistant professor in Electrical & Microelectronic Engineering at Rochester Institute of Technology. He has more than 100 publications in top journals and conference. His current interests lie in nanoelectronic devices empowering unconventional computing, domain-specific accelerator, and memory technology.

Mohsen Imani received the B.Sc. and M.S. degrees from the School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran, in 2011 and 2014, respectively, and the Ph.D. degree with the Department of Computer Science and Engineering, University of California at San Diego, La Jolla, CA, USA, in 2020. He is now an Assistant Professor with the University of California, Irvine. His current research interests include brain-inspired computing, approximation computing, and processing in-memory.

Cheng Zhuo received the B.S. and M.S. degrees in electronic engineering from Zhejiang University, Hangzhou, China, in 2005 and 2007, respectively. He received the Ph.D. degree in computer science and engineering from the University of Michigan, Ann Arbor, MI, USA, in 2010. He is currently a Professor with the college of Information Science and Electronic Engineering, Zhejiang University. His current research interests include computing in memory, deep learning, hardware accelerator, and general VLSI EDA areas. Prof. Zhuo has published over 100 technical papers and received 4 Best Paper Nominations in DAC'16, CSTIC'18, ICCAD'20, and VTS'21. He also received 2012 ACM/SIGDA technical leadership award, and 2017 JSPS Invitation Fellowship. He has served on the technical program and organization committees of many international conferences and as Associate Editor for IEEE TCAD, ACM TODAES, and Elsevier Integration. He is a senior member of IEEE and a Fellow of IET.



Xunzhao Yin is an assistant professor of the College of Information Science and Electronic Engineering at Zhejiang University. He received his Ph.D. degree in Computer Science and Engineering (CSE) from the University of Notre Dame in 2019 and B.S. degree in Electronic Engineering from Tsinghua University in 2013, respectively. His research interests include emerging circuit/architecture designs and novel computing paradigms with both CMOS and emerging technologies. He has published top journals and conference papers, including Nature Electronics, IEEE TCAD, TC, DAC, ICCAD and IEDM, etc. He has received the best paper award of ASPDAC 2023, ICITES 2022, and the best paper award nomination of ICCAD 2020, DATE 2022.