TAP-CAM: A Tunable Approximate Matching Engine based on Ferroelectric Content Addressable Memory

Chenyu Ni[1], Sijie Chen[1], Che-Kai Liu[2], Liu Liu[3], Mohsen Imani[4], Thomas Kämpfe[5], Kai Ni[3],
Michael Niemier[3], Xiaobo Sharon Hu[3], Cheng Zhuo[1,6,*], Xunzhao Yin[1,6,*]

[1]Zhejiang University, Hangzhou, China; [2]Georgia Institute of Technology, GA, USA
[3]University of Notre Dame, IN, USA; [4]University of California Irvine, CA, USA
[5]Fraunhofer IPMS, Dresden, Germany
[6]Key Laboratory of Collaborative Sensing and Autonomous Unmanned Systems of Zhejiang Province, China
*Corresponding authors, email: {czhuo, xzyin1}@zju.edu.cn

Abstract—**Pattern search is crucial in numerous analytic applications for retrieving data entries akin to the query. Content Addressable Memories (CAMs), an in-memory computing fabric, directly compare input queries with stored entries through embedded comparison logic, facilitating fast parallel pattern search in memory. While conventional CAM designs offer exact match functionality, they are inadequate for meeting the approximate search needs of emerging data-intensive applications. Some recent CAM designs propose approximate matching functions, but they face limitations such as excessively large cell area or the inability to precisely control the degree of approximation. In this paper, we propose TAP-CAM, a novel ferroelectric field effect transistor (FeFET) based ternary CAM (TCAM) capable of both exact and tunable approximate matching. TAP-CAM employs a compact 2FeFET-2R cell structure as the entry storage unit, and similarities in Hamming distances between input queries and stored entries are measured using an evaluation transistor associated with the matchline of CAM array. The operation, robustness and performance of the proposed design at array level have been discussed and evaluated, respectively. We conduct a case study of K-nearest neighbor (KNN) search to benchmark the proposed TAP-CAM at application level. Results demonstrate that compared to 16T CMOS CAM with exact match functionality, TAP-CAM achieves a $16.95\times$ energy improvement, along with a 3.06% accuracy enhancement. Compared to 2FeFET TCAM with approximate match functionality, TAP-CAM achieves a $6.78\times$ energy improvement.**

I. INTRODUCTION

In the era of advancing artificial intelligence, the computational demands on AI models are rapidly increasing. Training data volumes across various domains like computer vision (CV) [1], natural language processing (NLP) [2], and speech recognition [3] have surged, posing significant challenges to computing hardware and architectures, both at the edge and in data centers. The traditional von Neumann architecture, with its constant data movement between memory and processing units, exacerbates energy consumption and latency issues, intensifying the "Memory Wall" problem. To tackle this challenge, emerging computing paradigms, notably In-Memory Computing (IMC), have gained attention. IMC directly employs parallel data operations within the memory, enhancing core performance and efficiency while alleviating the "Memory Wall" problem [4]–[9].

Content Addressable Memory (CAM) emerges as a hardware solution of IMC, enabling parallel and efficient searching and similarity measurement within the memory. CAMs compare input data with all stored data simultaneously, and output the stored entry that matches with input or has the highest similarity to the input. Therefore, CAMs are viewed as a potential solution for accelerating various data-centric workloads like bioinformatics [10], [11], machine learning [12]–[14], and neural language processing [2]. Specifically, CAMs significantly speed up Hyperdimensional Computing (HDC), making this brain-inspired computing paradigm efficient for tasks like image classification and speech recognition [15]–[17]. This effectiveness arises from CAMs' ability to transform sequential pattern matching into highly parallelizable computational tasks and simplify the complex distance measurements into Hamming distance [18]. The rapid search and matching capability of CAMs make them essential components in applications requiring efficient data access and retrieval.

Conventional CMOS based CAM design consists of 10-16 transistors per cell, which results in large area overhead and high energy consumption [19]. To tackle the area and energy challenges, researchers have proposed utilizing emerging non-volatile memory (NVM) devices to construct more compact and efficient CAM designs, as these CAMs merge the storage and logic within the NVM devices, thus offering significant area and energy saving. CAMs based on 2-terminal NVMs like resistive RAM (RRAM) [20], [21], magnetic tunneling junction (MTJ) [22], [23], phase change memory (PCM) [24], and 3-terminal ferroelectric field effect transistor (FeFET) [25]–[33] have been explored. Among these devices, FeFETs stand out in constructing the compact and efficient CAM designs due to their unique hysteresis I-V characteristics, high current ON/OFF ratio, high off-state resistance, low write energy, and compatibility with CMOS technology [34]. While non-volatile storage can achieve high area efficiency and mitigate the high energy consumption caused by CMOS technology, these CAMs still encounter limitations for data-

intensive applications due to their exact search functionality. In the era of big data, as the amount of data for processing bursts and the chances of exact matching drop down, these CAMs with limited array size fail to maintain the hardware utilization efficiency while consuming extra area and energy overheads. Many applications require approximate pattern search functions where entries with a similarity within a certain threshold distance to the search query are desired. To address the challenge of limited CAM utilization efficiency, various CAM designs implementing approximate pattern search have been proposed. These approximate CAMs improve the utilization and overall energy efficiency by compensating the search accuracy within an acceptable range. For instance, HD-CAM [35] introduced a 10T CMOS-based approximate CAM with a matchline (*ML*) charge redistribution technique, but it suffers from a large cell area and lacks the support for wildcard (*don't care*) bits. Moreover, the design is unable to precisely control the degree of approximation, bit-by-bit. MHCAM [36] presented an approximate CAM design based on FeFET with programmable thresholds, but it's tailored to applications requiring multi-state Hamming distance. [37] implemented threshold matching by leveraging voltage scaling and controlling the precharge period, but its high energy consumption and inability to precisely control the threshold limit its applications. [12] introduced approximate matching capabilities using 2FeFET TCAM. It computes the Hamming distance between search and stored vectors in a highly parallelized manner by monitoring *ML* discharge rate. Despite achieving notable energy efficiency and density in TCAM, it lacks fine-grained control over approximate search precision.

To address aforementioned challenges of existing approximate CAMs, in this work, we propose TAP-CAM, a general approximate matching engine featuring a bit-by-bit tunable threshold match function. We consider FeFET as a representative NVM device, and propose to utilize a novel 2FeFET-2R ternary CAM (TCAM) cell structure to store ternary value. An evaluation transistor is employed between the parallel connected TCAM cells and the CAM array sense amplifier to control the *ML* discharge rate, and the tunable threshold of the approximate matching functionality is set by the bias voltage of the evaluation transistor. We validate the bit-wise XNOR logic and the tunable threshold matching functionality of TAP-CAM design at cell and array levels, respectively, and conduct extensive Monte Carlo simulations to examine the robustness against device-to-device variations. We use the K-nearest neighbor search (KNN) as a representative application to investigate the benefits of TAP-CAM at application level. Evaluation results demonstrate that TAP-CAM achieves a 16.95× energy improvement and 3.06% accuracy improvement compared to 16T CMOS CAM with exact match function. Compared to 2FeFET TCAM with approximate match functionality, TAP-CAM achieves a 6.78× energy improvement.

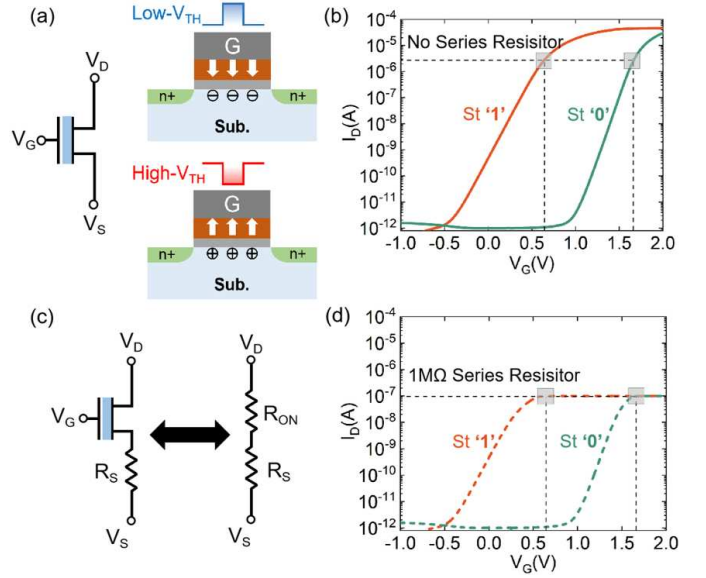The rest of paper is organized as follows: Sec. II reviews the FeFET device characteristics and existing CAM designs.

Fig. 1. **(a)** FeFET polarization directions and channel conditions after memory write operations; **(b)** The FeFET I_D-V_G characteristics after positive/negative gate write; **(c)** 1FeFET-1R structure and equivalent circuit; **(d)** The 1FeFET-1R I_D-V_G characteristics after positive/negative gate write.

Sec. III introduces the proposed TAP-CAM. Sec. IV presents the evaluation results and the KNN case study. Finally, Sec. V summarizes the paper.

II. BACKGROUND

In this section, we discuss the structure and operational principles of FeFETs, and review existing CAM design works.

A. FeFET Basics

Recent advancements in ferroelectric material, particularly hafnium oxide (HfO_2), have spurred research interest in ferroelectric transistors and the development of non-volatile circuit designs compatible with CMOS technology [32]. FeFETs incorporate a ferroelectric (FE) layer within the gate stack. These devices exhibit unique electrical hysteresis characteristics, exhibiting reversible polarization states upon an applied voltage-driven electric field. The FE layer induces a shift in the threshold voltage of the FeFET depending on the orientation of FE polarization [38], enabling non-volatile (NV) storage capabilities. By applying gate voltage pulses, such as -4V/+4V, to a FeFET device, as depicted in Figure 1(a), it can be programmed to store low and high V_{TH} states corresponding to logic '0' and '1', respectively. The associated hysteresis I_D-V_G transfer characteristics are shown in Figure 1(b) [39]. FeFETs, being voltage-driven for read and write operations, exhibit superior energy efficiency compared to two-terminal current-driven NVMs.

When the FeFET operates as a current source, its ON current gradually increases with the rise in gate voltage, as depicted in Figure 1(b). Consequently, there's a certain variability in the conduction current regarding the gate read
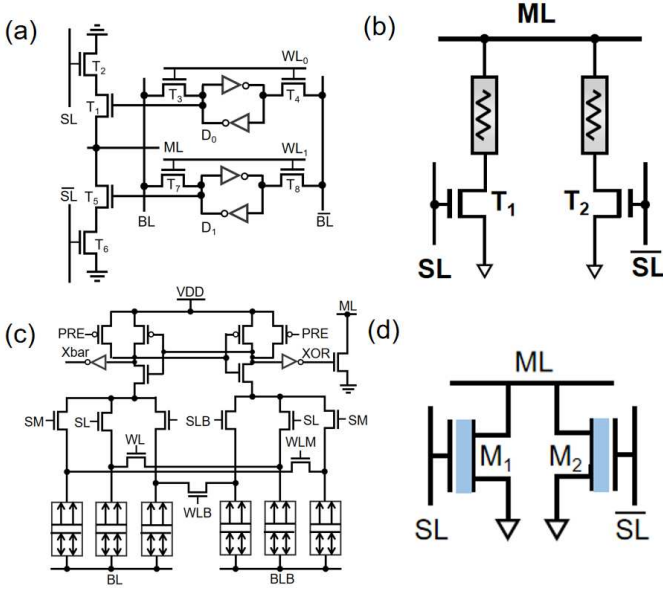
Fig. 2. Schematics of **(a)** 16T CMOS TCAM cell; **(b)** 2T-2ReRAM TCAM cell; **(c)** 20T-6MTJ TCAM cell; **(d)** 2FeFET TCAM cell.

Fig. 3. **(a) Exact match:** The stored entry that matches exactly with the query; **(b) Best match:** The stored entry that has the smallest distance to the query; **(c) Threshold match:** The stored entry whose distance to the query is below specified thresholds.

voltage. To ensure stable ON current during operation and enhance the design robustness, a current limiter is connected to the source of the FeFET, as shown in the equivalent circuit of Figure 1(c). Prior studies [27], [29] have shown that a series resistor on the drain/source of a FeFET can regulate the ON current, with 1FeFET-1R integration experimentally demonstrated [40]. Such integration suppresses the ON current variability, making it independent of the V_{TH} state and gate voltage when the series resistor is sufficiently large. The transfer characteristic curve of the 1FeFET-1R structure is depicted in Figure 1(d). We adopt the 1FeFET-1R structure using a series resistor as a current limiter in this work. This approach mitigates the impact of ON current variability on *ML* discharging in a CAM array achieving low power consumption and robust tunable approximate matching functionality.

B. Existing CAM Designs

Various CAM designs have been proposed based on CMOS technology and NVM devices. A conventional 16T CMOS TCAM cell is shown in Figure 2(a). CAMs leveraging NVM typically demonstrate enhanced performance over CMOS-based counterparts. For example, a 2T-2R TCAM design based on ReRAM was proposed in [24] for its compact structure, as shown in Figure 2(b). While it consumes less area compared with conventional CMOS-based CAM designs, the low HRS/LRS ratio, low variable resistance and current-driven write-in mechanism associated with large access transistors make the write and search energy significant concerns. [41] proposed a 20T-6MTJ TCAM design as illustrated in Figure 2(c), greatly enhancing the search speed and search performance. However, the reduced sense margin caused by the limited TMR ratio of STT-MRAM necessitates numerous
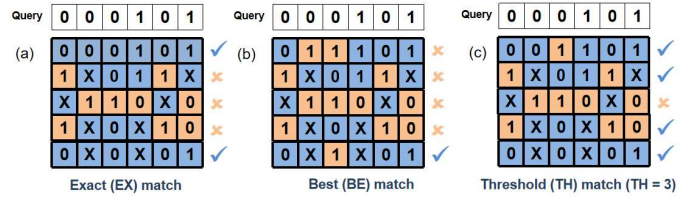
transistors to address this issue, thus severely impacting area and power consumption.

Among NVM based CAM designs, utilizing FeFET stands out due to its high ON/OFF current ratio, efficient voltage-driven write mechanisms, low energy consumption, and cost-effectiveness, enabling significant performance improvements compared to conventional CMOS designs and other NVM-based designs. Building upon advanced FeFET models, researchers have proposed various FeFET CAM designs, particularly designs of TCAM. The 2FeFET TCAM design as depicted in Figure 2(d) offers a compact alternative than CMOS counterparts [25]. 2FeFET TCAM features a smaller cell area, reduced write and search energy consumption, and search delay. However, it faces limitations such as the lack of support for approximate matching functionality.

C. Threshold Matching Concepts and Related Works

Most CMOS and NVM based CAM designs discussed earlier prioritize exact matching, as depicted in Figure 3(a), limiting their adaptability for data-intensive applications. In contrast, approximate matching gains favor due to its potential to enhance hardware utilization while maintaining acceptable accuracy. As a means to achieve approximate matching, best match CAMs, as illustrated in Figure 3(b), aim to output the stored entry with the highest similarity to the search query. For example, A-HAM [42] evaluates similarities across stored entries and identifies the closest Hamming distance to the input query. 4T-2MTJ utilizing STT-MRAM [43] measures similarity between input query and stored entries in terms of *ML* current and outputs the entry with the highest similarity. [44] introduced a CAM design for minimum Hamming distance search using digital circuits for bit comparison. A Winner-Take-All (WTA) circuit at the output selects the entry with the highest degree of matching to the search query. However, CAMs designed for best matching may fail in applications requiring the output of multiple entries with specific similarities. Therefore, threshold matching CAMs were devised.

Threshold matching CAMs, as illustrated in Figure 3(c), aim to provide multiple stored entries with similarity within a pre-defined Hamming distance (HD) threshold. For instance, the HD-CAM proposed in [35] utilizes a 10T CMOS-based design incorporating *ML* charge redistribution, enabling threshold matching with large HD tolerance, notably used in virus DNA
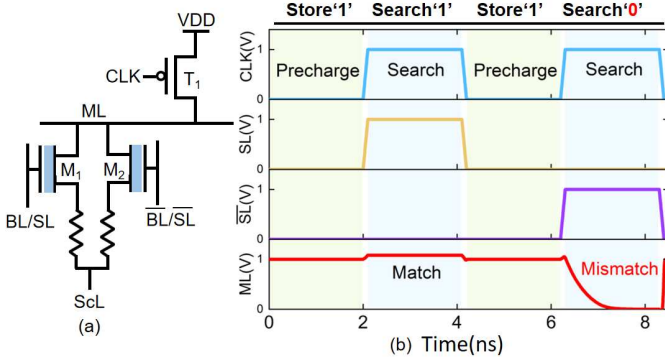
Fig. 4. **(a)** Structure of the proposed 2FeFET-2R TCAM cell; **(b)** Transient voltage waveforms of 2FeFET-2R CAM cell storing '1'.

classification. However, the SRAM based HD-CAM cell incurs substantial area and energy overheads. Furthermore, its effectiveness is limited in discerning patterns with substantial HDs due to the intricate tuning of *ML* discharge current, making bit-by-bit tuning of HD thresholds impractical. [36] introduced MHCAM, a multi-state CAM design encoding multiple CAM cells into distinct multi-states per dimension to perform both dimension-wise exact matching and reconfigurable threshold matching. However, additional transistors introduce fixed bit precisions (1-bit/2-bit/4-bit/8-bit per dimension), restricting fine-grained tunability in threshold matching and adaptability to applications demanding multi-state HD. The ReRAM-based CAM proposed in [37] implements threshold matching by leveraging voltage scaling and controlling the precharge period. However, the current-driven mechanisms of ReRAMs result in high power consumption during operation and limited HD thresholds can be achieved due to the large *ML* discharge current and non-trivial threshold-associated period sampling. [12] implements approximate matching functionality based on 2FeFET TCAM. It calculates the HD between search and stored vectors in a parallel manner by sensing the discharge rate of *ML*. While achieving high energy efficiency and density in TCAM, it lacks precise control over the degree of approximate searching.

These threshold search CAMs all face a common issue, that they cannot precisely control the degree of approximate matching. Therefore, our design will focus on implementing bit-by-bit tuning of threshold to control the degree of approximate matching.

III. PROPOSED TAP-CAM DESIGN

In this section, we present the TAP-CAM design with bit-by-bit tunable HD threshold match functionality, exploiting the 2FeFET-2R structure and incorporating a threshold-defined evaluation transistor. We first discuss the structure and operation principles of the cell, and then elucidate the threshold approximate match implementation at the array level.

V_{write} = 4V	V_{search} = 1V	*BL/SL*	$\overline{BL/SL}$	*ScL*	M_1	M_2
Write'1'	Step1	V_{write}	0	0	'1'	hold
	Step2	V_{write}	0	V_{write}	hold	'0'
Write'0'	Step1	0	V_{write}	V_{write}	'0'	hold
	Step2	0	V_{write}	0	hold	'1'
Write *don't care*		V_{write}	V_{write}	0	'1'	'1'

A. 2FeFET-2R TCAM Cell

Figure 4(a) shows the structure of the proposed 2FeFET-2R TCAM Cell. It comprises a pair of parallel 1FeFET-1R structures, with the FeFET drain connected to the matchline (*ML*), and the other end of the structure connected to the sourceline (*ScL*), driven by either V_{write} or *GND*. The FeFET gate connects to the bitline and searchline (*BL/SL* and $\overline{BL/SL}$). By adjusting the write gate input, the FeFET threshold aligns with different storage values. The 2FeFET-2R structure can store logic '1', '0', and *don't care* wildcard state. Table I outlines the write operations of the 2FeFET-2R cell. Data bits are written in two steps, storing complementary logic states in each FeFET. To write logic '1', V_{write} is applied to *BL/SL*, while '0' to *ScL* and $\overline{BL/SL}$. This sets V_{GS} of M_1 to 4V, writing logic '1' to M_1. In the second step, V_{write} is applied to *ScL*, while gate voltage remains the same, writing logic '0' to M_2. Thus, the complementary stored values represents logic '1'. Similarly, to write logic '0' into the cell, '0' is written to M_1 and '1' to M_2, respectively. To write *don't care* state, logic '1' is written to both M_1 and M_2. This sets both FeFETs to high-V_{TH} state, matching regardless of the search value, aligning with the masking function of '*don't care*' bits. During writes, *ML* is grounded to eliminate static current. Figure 1(b) displays I_D-V_G curves for FeFETs under different write pulses.

During search, *ML* voltage is precharged to high via a precharge transistor, and the search voltages are applied to searchlines (*SL*/$\overline{SL}$) according to the query data. For logic '1', *SL* set to 1V, and 0 for logic '0', the *ML* voltage indicates the matching result. Figure 4(b) validates the function of the 2FeFET-2R cell. *ML* is first precharged by controlling T_1's gate voltage *CLK*, and then left floating upon search phase. When searching '1', *ML* voltage stays high with *SL* = 1V, indicating a match. Conversely, searching '0' rapidly drops *ML* voltage to 0, indicating a mismatch.

B. 2FeFET-2R TCAM Array

Figure 5 demonstrates the schematic of the proposed 2FeFET-2R TAP-CAM array storing a 64-bit word with corresponding peripheral circuits. PMOS T_1 precharges *ML* before the search operation, while an evaluation transistor T_2 is connected between *ML* and V_o to enable tunable threshold matching function. Adjusting the gate voltage of the evaluation transistor controls the discharge rate of *ML*, allowing varying mismatch bits to be sensed by the sense amplifier (SA) as a match case.

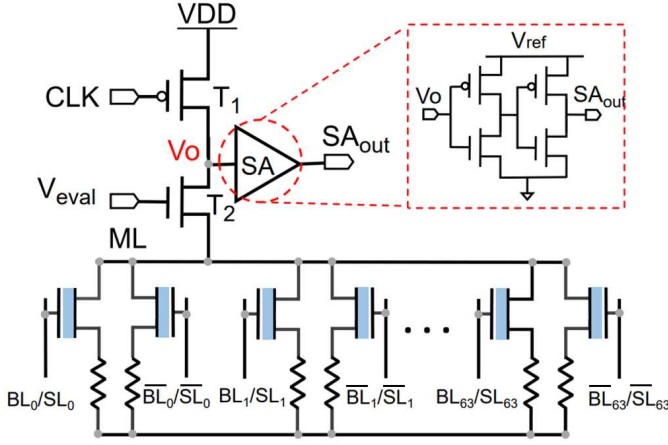During the precharge, *CLK* is set to low, turning T_1 and T_2 ON, and precharging *ML* to *VDD*. During the search phase,

Fig. 5. Structure of a 2FeFET-2R TCAM array with wordlength 64.

Mismatch Threshold(bit)	0	1	2	3	4	5
V_{eval}(V)	1	0.75	0.63	0.52	0.43	0.37

setting the *CLK* signal high turns T_1 OFF and cutting the charging path. Pre-defined bias voltages are applied to the gate of evaluation transistor V_{eval} based on required mismatch thresholds. A mismatch between the stored entry and the search query forms a conduction path from V_o to *GND*, discharging V_o and decreasing the voltage. The rate of voltage decrease depends on the number of mismatched cells and T_2's gate voltage V_{eval}. This rate affect the output of SA SA_{out} which indicates the time for SA_{out} to transition from high to low. With constant V_{eval}, more mismatched bits increase the discharge current from V_o to *GND*, accelerating SA_{out} voltage drop. Similarly, with constant mismatched bits, higher V_{eval} boosts the conduction of T_2, hastening SA_{out} voltage drop. Hence, given the fixed SA sense time, decreasing the V_{eval} allows for increasing the mismatch threshold.

Without loss of generality, for the TAP-CAM with n bits mismatch threshold (Th-n), i.e., $\leq$n mismatch bits are sensed as a match case, and $\geq$(n+1) bits mismatch indicates a mismatch, the sense margin between the n bits mismatch and (n+1) bits mismatch is determined by the equivalent resistance and associated *ML* capacitance of the array C_M. The equivalent resistance for the two mismatch cases can be expressed as follows:

$$R_n = \frac{1}{n} \cdot (R_{ON} + R_S) \tag{1}$$

$$R_{n+1} = \frac{1}{n+1} \cdot (R_{ON} + R_S) \tag{2}$$

where R_n represents the approximate equivalent resistance of array with n bits mismatch, and R_{n+1} represents the approximate equivalent resistance of array with (n+1) bits mismatch. R_{ON} represents the equivalent resistance of an ON FeFET, and R_S represents the series resistance. From charging and discharging formula of RC circuit, we can approximately formulate the *ML* voltage U:

$$U = U_0 \cdot e^{-\frac{t}{RC_M}} \tag{3}$$

$$\frac{dU}{dt} = U_0 \cdot (-\frac{1}{RC_M})e^{-\frac{t}{RC_M}} \tag{4}$$

where U_0 represents the initial voltage of *ML*. From Equation 4 we can conclude that the rate of *ML* voltage drop will be faster as the equivalent resistance decreases. From Equation 1 and Equation 2, R_n is larger than R_{n+1}. Therefore, the voltage of *ML* corresponding to (n+1) bits mismatch drops faster than that of n bits mismatch. Upon the sensing, the sense margin of Th-n ΔU can be expressed as follows:

$$\Delta U = U_n - U_{n+1} = U_0 \cdot (e^{-\frac{t}{R_n C_M}} - e^{-\frac{t}{R_{n+1} C_M}}) \tag{5}$$

where U_n represents the *ML* voltage corresponding to n bits mismatch, and U_{n+1} represents the *ML* voltage corresponding to (n+1) bits mismatch. From Equation 5, we observe that R_S affects the magnitude of ΔU over time t, thus influencing the sense margin. Simultaneously, a larger R_S value introduces larger search delay. Therefore, selecting an appropriate R_S value is necessary to ensure that both sense margin and search delay remain within reasonable limits. We here select $R_S =$ 0.3M.

Another factor that affects the sense margin and the search time is the bias voltage at evaluation transistor gate. To implement the functionality of bit-by-bit tunable threshold approximate matching, we determine appropriate evaluation voltages V_{eval} to distinguish different mismatch thresholds, taking the threshold ranging 0-6 bits as an example. This involves adjusting the gate voltage of the evaluation transistor to differentiate between 0-bit and 1-bit mismatch (Th-0), 1-bit and 2-bit mismatch (Th-1), and so forth. Increasing the number of mismatch bits and evaluation transistor gate voltage V_{eval} lead to faster SA_{out} voltage decrease. Hence, with increasing mismatch threshold, we decrease V_{eval} to maintain consistent sense time window across different mismatch thresholds. The evaluation voltages are therefore experimentally examined and configured as summarized in Table II to ensure that the sense time for distinguishing different mismatch thresholds falls within the same time window. Different evaluation voltages correspond to different mismatch thresholds. This evaluation voltage configuration lays the foundation for subsequent performance and latency analysis.

The *ML* transient waveforms corresponding to different mismatch thresholds in Figure 6 validate the bit-by-bit tunable threshold matching function. Solid lines show the *ML* voltage waveforms when the number of mismatched bits equals to the pre-defined mismatch threshold, while dashed lines show the *ML* voltages when the number of mismatched bits exceeds the pre-defined threshold. The sense margin of mismatch thresholds decreases as the threshold increases. According
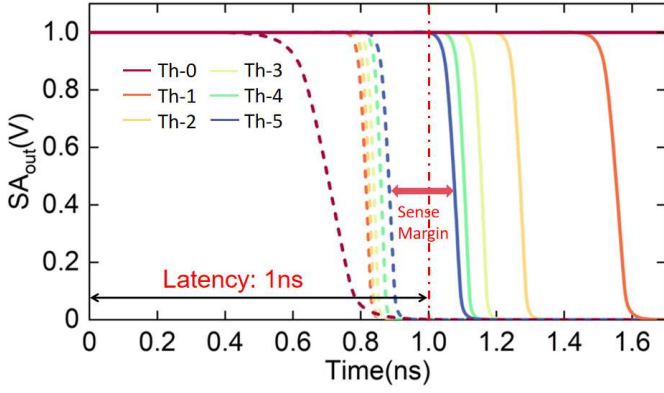
Fig. 6. Transient waveforms of *ML* under different mismatch thresholds. Solid and Dashed lines represent the match and mismatch cases corresponding to a certain mismatch threshold, respectively.
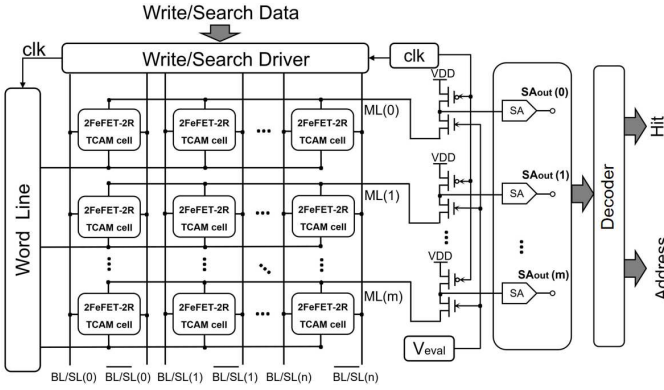

Fig. 7. Schematic of m×n TAP-CAM array.

Fig. 8. 100 Monte Carlo simulations considering device-to-device variations: **(a)** The output waveforms under *VDD* = 0.6V; **(b)** The output waveforms under *VDD* = 1V.

Fig. 9. Energy and latency of the proposed 2FeFET-2R TAP-CAM array with varying **(a)** *VDD*; **(b)** mismatch thresholds; **(c)** number of rows and **(d)** number of bits per row.

to Figure 6, the search latency for distinguishing adjacent mismatch threshold ranging from Th-0 to Th-5 is 1 ns.

IV. EVALUATION

In this section, we first evaluate the energy and performance of the proposed TAP-CAM design. We then benchmark the proposed TAP-CAM array in the context of K-nearest neighbor search tasks as tunable approximate matching engine.

A. Evaluation Setup

For the energy and performance evaluations, we conduct our experiments on a TAP-CAM array with m rows and n columns, as shown in Figure 7. The cells within the same row share the *ML* and *ScL*, and the cells within the same column share *SLs*, enabling parallel search operations. Write/Search buffer drive stored/search vectors into *SLs* for search operations, consistent with Table I. During the search, all rows compare the same input query with stored entries. If a mismatch occurs, *ML* discharges. If *ML* voltage drops below the sense amplifier threshold within the pre-defined sense time window, the corresponding SA output transitions to 0, recognized by the decoder as mismatch. Conversely, if a match occurs, the address of the stored entry matching the search query is output.
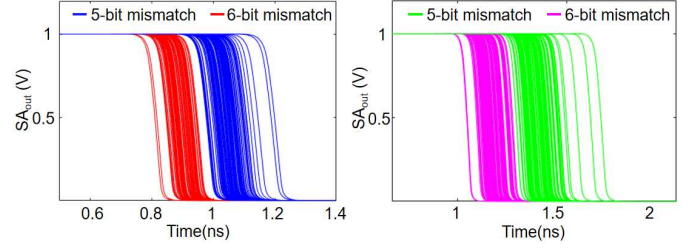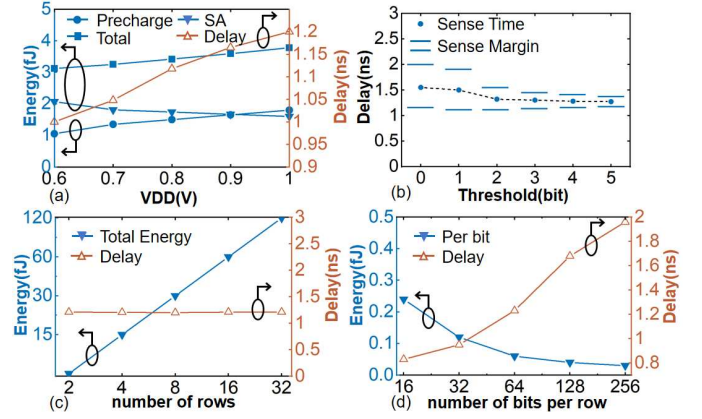
The proposed 2FeFET-2R TAP-CAM array is evaluated using SPECTRE. The FeFETs are simulated based on the Preisach FeFET model [39]. All MOSFETs are modeled using the 45nm PTM model and the 27°C TT process corner [45]. The wordlength is set to 64 cells.

B. Robustness Validation

The robustness of the proposed TAP-CAM design under varying operating conditions is examined, specifically with *VDD* = 0.6V and *VDD* = 1V, respectively. The FeFETs are assumed to feature the stored low/high V_{TH} threshold voltage states with a deviation σ = 54mV, and 8% series resistor variability is considered [40]. 100 Monte Carlo simulations have been conducted to distinguish between 5-bits and 6-bits mismatches when the mismatch threshold is set to 5 bits (Th-5). Figure 9 consistently reveals that the time windows across the 100 runs can be identified. This observation suggests that the proposed design effectively distinguishes between the adjacent numbers of mismatched bits by employing the evaluation transistor. Based on these results, it can be inferred that the proposed TAP-CAM design demonstrates the robustness, as it reliably achieves approximate threshold matching functionality given the variations in operating voltage and device variations.

TABLE III
METRIC COMPARISON SUMMARY OF CAM DESIGNS

Reference	[19], [12]	[35]	[37]	[12]	Our Work
Technology	CMOS	CMOS	ReRAM	FeFET	FeFET
Node(nm)	45	65	45	45	45
Transistors/cell	16T	10T	2T-2R	2FeFET	2FeFET-2R
Match Style	Exact	Threshold	Threshold	Threshold	Threshold
Cell size(μm^2)	1.2	5.45	0.41	0.15	0.15*
Search delay(ps)	582	1000	1450	355	1200
Energy (fJ/bit/search)	1.00 16.95×	0.76 12.88×	0.56 9.49×	0.4 6.78×	0.059 1×

*: Back-end-of-line resistor incurs no additional area overhead as reported in [40].

C. CAM Array Evaluation

The search energy consumption of the proposed array mainly originates from precharging the *ML* and SA energy consumption. Precharging the *ML*, primarily done by T_1, depends heavily on *VDD* and the associated *ML* parasitic capacitance. Figure 9(a) demonstrates the impact of scaling *VDD* on the search energy consumption and latency. As *VDD* scales up, the precharging energy increases, leading to overall higher search energy consumption. At the same time, the amplitude of *ML* dropping from high to low level when mismatch occurs increases, thereby increasing the search delay. Figure 9(b) shows the sense time and sense margin for different mismatch thresholds at *VDD* = 1V. The sense margin is the narrowest at the 5-bit mismatch threshold (Th-5), thus is selected as the sense margin for the SA sense time. Figure 9(c) demonstrates how search energy and latency change with varying row numbers. Increased rows allow parallel search operations, linearly increasing the energy consumption with negligible latency change. Finally, Figure 9(d) examines the wordlength's effect on the search latency and energy consumption per bit. Longer wordlengths associate more parasitic capacitance on the *ML*, slowing down the discharge speed and thus increasing the search latency. The increase in capacitance leads to a rise in precharge energy per word. But increasing wordlength has minimal impact on the energy consumption of SA, so the search energy per bit decreases. The increasing latency and decreasing energy consumption per bit show trade-offs in the CAM array design optimization.

Table III provides a comprehensive comparison of the proposed 2FeFET-2R TAP-CAM with other CAM designs, in terms of device type, technology node, device count per cell, cell size, performance and normalized search energy. Cell size estimation is based on a 2×2 layout of the 2FeFET-2R TAP-CAM array. Compared to the conventional CMOS CAM designs, our proposed 2FeFET-2R TAP-CAM design offers a much smaller cell size. The comparisons highlight the significant advantages of the proposed 2FeFET-2R TAP-CAM design over other CAM designs in terms of energy consumption per bit per search. The energy efficiency of 2FeFET-2R TAP-CAM is notably superior, being 16.95×, 12.88×, 9.49×, and 6.78× more efficient compared to 16T TCAM, 10T CAM, 2T-2R TCAM, and 2FeFET TCAM, respectively. While some existing designs achieve approximate

search functionality, their energy consumption remains substantially higher than that of 2FeFET-2R structure. Although our design incurs relatively high search delay, considering the search latency and energy trade-offs and the substantial energy advantages of our proposed design, increased delay is deemed acceptable.

These findings validate the remarkable energy efficiency of 2FeFET-2R TAP-CAM array, emphasizing its immense potential for data-intensive search applications. This suggests that 2FeFET-2R TAP-CAM architecture is well-positioned to address the evolving needs of modern computing environments, particularly those requiring efficient and high-performance solutions for processing large volumes of data in search-intensive applications.

D. Case Study: K-Nearest Neighbor Search

To demonstrate the efficiency of the proposed design, we benchmark the proposed 2FeFET-2R TAP-CAM array in the context of K-nearest neighbor (KNN) search framework. KNN, a fundamental algorithm in machine learning, embodies a non-parametric supervised model, particularly effective when *K* = 1, representing the nearest neighbor (NN) classification. This algorithm finds widespread use across various fields, including HDC [46], [47], reinforcement learning [48], and bioinformatics [11], etc.

At the core of the KNN approach lies the calculation of distances between the query instance, denoted as x, and the stored vectors, denoted as y_i, within the CAM array. This process utilizes a distance function, typically denoted as $d(x, y_i)$, which quantifies the dissimilarity or similarity between the data points. When *K* = 1, i.e. NN classification, the class label attributed to the query instance x corresponds to the category of the nearest stored vector y_i, identified by the smallest distance metric. This intuitive method allows for straightforward classification based on proximity, making it particularly suitable for scenarios with intricate decision boundaries or complex dataset patterns. Conversely, when *K* exceeds 1 instead of relying on the nearest neighbor, the algorithm considers the k closest neighbors of the query instance x. The class label assigned to x is determined by a majority voting mechanism, where the most frequent class label among the k nearest neighbors prevails. This adaptive approach enables KNN to capture more nuanced relationships within the dataset, thereby enhancing its predictive capability and robustness in various applications.

In benchmarking our proposed 2FeFET-2R TAP-CAM, for a given a function $d(x, y_i)$, which measures the distance between the query x and the i-th stored vector y_i in the CAM array, NN assigns the class label with the smallest distance value to x. Similarly, in KNN, given a query x, it assigns the most common class label of x's k nearest neighbors to x [49], as illustrated in Equation 6.

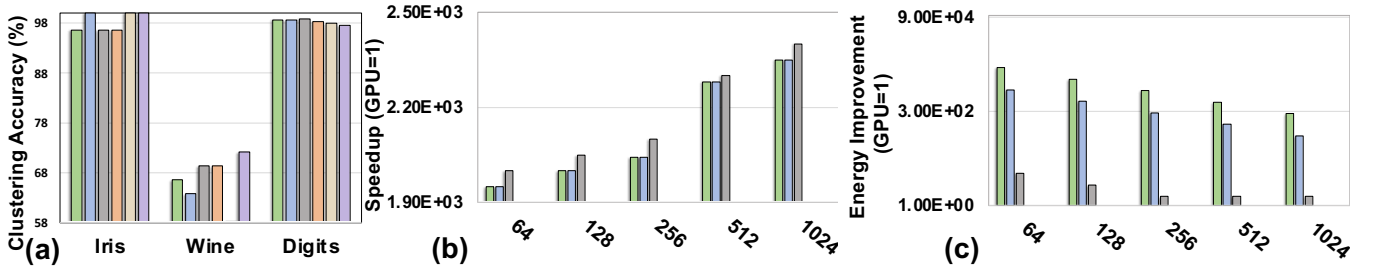$$c(x) = argmax \sum_{i=1}^{k} \delta(c, c(y_i)) \qquad (6)$$

Fig. 10. **(a)** KNN clustering accuracy under different TAP-CAM thresholds, ranging from Th-1 to Th-6 (left to right); **(b)** Computational speedup and **(c)** energy efficiency improvement of TAP-CAM with varying wordlengths compared to a GPU implementation. Datasets from left to right are Iris, Wine and Digits.

TABLE IV
DATASETS (n: TOTAL INSTANCES, f: FEATURES, K: NUMBER OF CLASSES)

Dataset	n	f	K	Description
Iris	150	4	3	Species of Iris [50]
Wine	178	13	3	Chemical analysis of wines [50]
Digits	5620	64	10	Hand-written digits [50]

where $c(x)$ represents the class label of the query x, while $c(y_i)$ represents that of y_i. y_i with i ranges from 1 to k represent the k nearest neighbors. We have $\delta(c, c(y_i)) = 1$ when the query's label c equals the label of y_i, otherwise $\delta(c, c(y_i)) = 0$.

To comprehensively evaluate the effectiveness and performance of the proposed TAP-CAM architecture, KNN clustering analysis is conducted under the three most frequently referenced datasets in the UCI Machine Learning Repository, as shown in Table IV. The datasets include Iris, Wine, and Digits, representing a wide range of data types and complexities. In order to achieve a robust evaluation, we have partitioned these datasets into training sets and test sets at an 8:2 ratio to ensure accurate testing and comparison of TAP-CAM model's performance.

Figure 10(a) illustrates the effectiveness of the proposed TAP-CAM architecture across different datasets. Among Iris, Wine, and Digits, the *Wine* dataset exhibits the highest susceptibility to hardware device-level variations. This observation emphasizes the importance of robustness in hardware designs, particularly in applications where environmental factors introduce variability. Additionally, we have examined the accuracy performance of KNN search under different TAP-CAM thresholds. Interestingly, the results indicate that identifying the nearest neighbor may not always yield the optimal solution. For instance, the Iris, Wine, and Digits datasets achieve their respective maximum clustering accuracies at $K = 2$, $K = 6$, and $K = 3$, respectively. With the proposed tunable approximate matching scheme, an average 3.06 % accuracy improvement is observed compared to existing exact-match CAM methods.

Power consumption is obtained via the *Nvidia-smi* toolkit,

with the study conducted on *Nvidia 2080ti GPU*, and the TAP-CAM operations are analyzed via the *Pytorch profiler*. Assuming 256 TAP-CAM rows, feasible in current manufacturing technology, the KNN clustering benchmark considers different TAP-CAM wordlengths at the algorithmic level. Idling power is excluded from the results. Figure 10(b) illustrates that TAP-CAM exhibits at least 1.95×10^3 speedup compared to GPU implementation. In addition, the energy consumption in TAP-CAM grows linearly with the number of cells per row, whereas GPU implementations show little increase with dimensionality increment. Consequently, as dimensionality increases, energy efficiency improvement decreases as demonstrated in Figure 10(c). For the *Digits* dataset, TAP-CAM energy increases with the large number of instances and features, resulting in an average improvement of $3.15\times$ compared to GPU implementations.

These results illustrate the effectiveness of the proposed TAP-CAM architecture across multiple datasets and scenarios, confirming its feasibility and superiority in practical applications. Through evaluation and comparison with existing methodologies, we highlight the potential of our design to advance CAM technology and contribute to machine learning research and development.

V. CONCLUSION

In this paper, we introduce TAP-CAM, a compact and energy-efficient TCAM design capable of threshold approximate matching. We propose a novel 2FeFET-2R TCAM design which employs an evaluation transistor to adjust the ML discharge rate and measure the Hamming distance between the input query and the stored entries. Through gate bias voltage configuration, TAP-CAM achieves bit-by-bit tunable HD threshold matching functionality that is a crucial operation in many data-intensive applications. Evaluation results and application benchmarking suggest that our proposed 2FeFET-2R TAP-CAM array surpasses other advanced CAM technology in both energy efficiency and performance.

ACKNOWLEDGEMENTS

REFERENCES

[1] A. Krizhevsky *et al.*, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[2] D.-T. Nguyen *et al.*, "Deepcam: A fully cam-based inference accelerator with variable hash lengths for energy-efficient deep neural networks," in *DATE*. IEEE, 2023, pp. 1–6.

[3] J. Oruh *et al.*, "Long short-term memory recurrent neural network for automatic speech recognition," *IEEE Access*, vol. 10, pp. 30 069–30 079, 2022.

[4] N. Verma *et al.*, "In-memory computing: Advances and prospects," *IEEE SSC-M*, vol. 11, no. 3, pp. 43–55, 2019.

[5] A. Sebastian *et al.*, "Memory devices and applications for in-memory computing," *Nature nanotechnology*, vol. 15, no. 7, pp. 529–544, 2020.

[6] X. Yin *et al.*, "Deep random forest with ferroelectric analog content addressable memory," *Science Advances*, vol. 10, no. 23, p. eadk8471, 2024.

[7] X. Yin *et al.*, "Ferroelectric compute-in-memory annealer for combinatorial optimization problems," *Nature Communications*, vol. 15, no. 1, p. 2419, 2024.

[8] Z. Yang *et al.*, "Energy efficient dual designs of fefet-based analog in-memory computing with inherent shift-add capability," in *ACM/IEEE DAC*, 2024.

[9] C. Li *et al.*, "Febim: Efficient and compact bayesian inference engine empowered with ferroelectric in-memory computing," in *ACM/IEEE DAC*, 2024.

[10] H. Zhong *et al.*, "Asmcap: An approximate string matching accelerator for genome sequence analysis based on capacitive content addressable memory," in *ACM/IEEE DAC*. IEEE, 2023, pp. 1–6.

[11] A. F. Laguna *et al.*, "Seed-and-vote based in-memory accelerator for dna read mapping," in *ICCAD*, 2020, pp. 1–9.

[12] K. Ni *et al.*, "Ferroelectric ternary content-addressable memory for one-shot learning," *Nature Electronics*, vol. 2, no. 11, pp. 521–529, 2019.

[13] Z. Xu *et al.*, "Ferex: A reconfigurable design of multi-bit ferroelectric compute-in-memory for nearest neighbor search," in *DATE*. IEEE, 2024, pp. 1–6.

[14] X. S. Hu *et al.*, "In-memory computing with associative memories: A cross-layer perspective," in *IEDM*. IEEE, 2021, pp. 25–2.

[15] T. F. Wu *et al.*, "Hyperdimensional computing exploiting carbon nanotube fets, resistive ram, and their monolithic 3d integration," *IEEE JSSC*, vol. 53, no. 11, pp. 3183–3196, 2018.

[16] M. Imani *et al.*, "Searchd: A memory-centric hyperdimensional computing with stochastic training," *IEEE TCAD*, vol. 39, no. 10, pp. 2422–2433, 2019.

[17] L. Ge *et al.*, "Classification using hyperdimensional computing: A review," *IEEE Circuits and Systems Magazine*, vol. 20, no. 2, pp. 30–47, 2020.

[18] Y. Kim *et al.*, "Geniehd: Efficient dna pattern matching accelerator using hyperdimensional computing," in *DATE*. IEEE, 2020, pp. 115–120.

[19] K. Pagiamtzis *et al.*, "Content-addressable memory (cam) circuits and architectures: a tutorial and survey," *IEEE JSSC*, vol. 41, no. 3, pp. 712–727, 2006.

[20] H. Li *et al.*, "Sapiens: A 64-kb rram-based non-volatile associative memory for one-shot learning and inference at the edge," *IEEE T-ED*, vol. 68, no. 12, pp. 6637–6643, 2021.

[21] M.-F. Chang *et al.*, "17.5 a 3t1r nonvolatile tcam using mlc reram with sub-1ns search time," in *ISSCC*, 2015, pp. 1–3.

[22] S. Matsunaga *et al.*, "A 3.14 µm² 4t-2mtj-cell fully parallel tcam based on nonvolatile logic-in-memory architecture," in *VLSIC*, 2012, pp. 44–45.

[23] C. Zhuo *et al.*, "Design of ultracompact content addressable memory exploiting 1t-1mtj cell," *IEEE TCAD*, vol. 42, no. 5, pp. 1450–1462, 2022.

[24] J. Li *et al.*, "1 mb 0.41 µm² 2t-2r cell nonvolatile tcam with two-bit encoding and clocked self-referenced sensing," *IEEE JSSC*, vol. 49, no. 4, pp. 896–907, 2014.

[25] X. Yin *et al.*, "An ultra-dense 2fefet tcam design based on a multi-domain fefet model," *IEEE TCAS-II*, vol. 66, no. 9, pp. 1577–1581, 2019.

[26] X. Yin *et al.*, "Design and benchmarking of ferroelectric fet based tcam," in *DATE*. IEEE, 2017, pp. 1444–1449.

[27] T. Soliman *et al.*, "Ultra-low power flexible precision fefet based analog in-memory computing," in *IEDM*, 2020, pp. 29.2.1–29.2.4.

[28] X. Yin *et al.*, "Ferroelectric ternary content addressable memories for energy-efficient associative search," *IEEE TCAD*, vol. 42, no. 4, pp. 1099–1112, 2022.

[29] X. Yin *et al.*, "An ultracompact single-ferroelectric field-effect transistor binary and multibit associative search engine," *Advanced Intelligent Systems*, vol. 5, no. 7, p. 2200428, 2023.

[30] H. Xu *et al.*, "On the challenges and design mitigations of single transistor ferroelectric content addressable memory," *IEEE EDL*, 2023.

[31] Q. Huang *et al.*, "A fefet-based time-domain associative memory for multi-bit similarity computation," in *DATE*, 2024, pp. 1–6.

[32] X. Yin *et al.*, "Fecam: A universal compact digital and analog content addressable memory using ferroelectric," *IEEE T-ED*, vol. 67, no. 7, pp. 2785–2792, 2020.

[33] C. Li *et al.*, "A scalable design of multi-bit ferroelectric content addressable memory for data-centric computing," in *IEDM*. IEEE, 2020, pp. 29–3.

[34] L. Liu *et al.*, "Eva-cam: A circuit/architecture-level evaluation tool for general content addressable memories," in *DATE*, 2022, pp. 1173–1176.

[35] E. Garzón *et al.*, "Hamming distance tolerant content-addressable memory (hd-cam) for dna classification," *IEEE Access*, vol. 10, pp. 28 080–28 093, 2022.

[36] L. Liu *et al.*, "A reconfigurable fefet content addressable memory for multi-state hamming distance," *IEEE TCAS-I*, 2023.

[37] M. Imani *et al.*, "Masc: Ultra-low energy multiple-access single-charge tcam for approximate computing," in *DATE*, 2016, pp. 373–378.

[38] D. Reis *et al.*, "Design and analysis of an ultra-dense, low-leakage, and fast fefet-based random access memory array," *IEEE JXCDC*, vol. 5, no. 2, pp. 103–112, 2019.

[39] K. Ni *et al.*, "A circuit compatible accurate compact model for ferroelectric-fets," in *2018 IEEE Symposium on VLSI Technology*, 2018, pp. 131–132.

[40] D. Saito *et al.*, "Analog in-memory computing in fefet-based 1t1r array for edge ai applications," in *2021 Symposium on VLSI Circuits*, 2021, pp. 1–2.

[41] C. Wang *et al.*, "Design of magnetic non-volatile tcam with priority-decision in memory technology for high speed, low power, and high reliability," *IEEE TCAS-I*, vol. 67, no. 2, pp. 464–474, 2020.

[42] M. Imani *et al.*, "Exploring hyperdimensional associative memory," in *HPCA*, 2017, pp. 445–456.

[43] Y. Ma *et al.*, "A spin transfer torque magnetoresistance random access memory-based high-density and ultralow-power associative memory for fully data-adaptive nearest neighbor search with current-mode similarity evaluation and time-domain minimum searching," *Japanese journal of applied physics*, vol. 56, no. 4S, p. 04CF08, 2017.

[44] H. Mattausch *et al.*, "Compact associative-memory architecture with fully parallel search capability for the minimum hamming distance," *IEEE JSSC*, vol. 37, no. 2, pp. 218–227, 2002.

[45] R. Vattikonda *et al.*, "Modeling and minimization of pmos nbti effect for robust nanometer design," in *ACM/IEEE DAC*, 2006, pp. 1047–1052.

[46] C.-K. Liu *et al.*, "Cosime: Fefet based associative memory for in-memory cosine similarity search," in *IEEE/ACM ICCAD*, 2022, pp. 1–9.

[47] S. Shou *et al.*, "See-mcam: Scalable multi-bit fefet content addressable memories for energy efficient associative search," in *IEEE/ACM ICCAD*. IEEE, 2023, pp. 1–9.

[48] M. Li *et al.*, "Associative memory based experience replay for deep reinforcement learning," in *IEEE/ACM ICCAD*, 2022, pp. 1–9.

[49] L. Jiang *et al.*, "Survey of improving k-nearest-neighbor for classification," in *FSKD*, vol. 1. IEEE, 2007, pp. 679–683.

[50] "Uci machine learning repository," https://archive.ics.uci.edu/ml/datasets.