

Dynamic Priority and Multi-Key -TCAM Architecture for Scalable Ultra-Low-Latency Network Packet Transmission

Varanganti Sravanthi¹, Maddhi Raju^{1*}

¹Department of Electronics & Communication Engineering, Vaagdevi Engineering College,
Warangal, 506005, Telangana, India.

*Correspondence: Maddhi Raju (Laxmirajumaddhi@gmail.com)

Abstract

The rapid growth of cloud computing, artificial intelligence, and high-speed communication networks has significantly increased the demand for efficient packet lookup mechanisms in modern networking infrastructures. Recent industry reports indicate that the TCAM memory market is projected to grow from approximately USD 2.8 billion in 2025 to over USD 6.4 billion by 2034, while global data center traffic continues to increase annually due to the expansion of cloud services and hyperscale computing environments. However, existing Content Addressable Memory (CAM) and Ternary CAM (TCAM) based lookup architectures suffer from several limitations, including fixed-width memory structures, inefficient memory utilization, increased lookup latency, higher LUT utilization, elevated power consumption, limited scalability, and inadequate priority management under heavy traffic conditions. These challenges negatively impact network throughput and overall hardware efficiency, particularly in large-scale communication environments. To address these issues, this work proposes a Dynamic Priority with Multi-Key TCAM (DPMK-TCAM) Architecture for Scalable Ultra-Low-Latency Network Packet Transmission. The proposed architecture integrates a Multikey Search Engine for simultaneous packet attribute matching, a Programmable Dynamic Priority based Multiplexer (PDP Multiplexer) for adaptive priority assignment and conflict resolution, and a Variable Width TCAM (VW-TCAM) for efficient memory utilization and flexible storage management. The architecture further incorporates matching percentage analysis,

dynamic comparison, maximum matching identification, adaptive memory allocation, and feedback-driven optimization mechanisms to enhance lookup accuracy and processing efficiency. By combining intelligent priority control with adaptive TCAM organization, the proposed system aims to reduce lookup delay, minimize hardware resource consumption, improve throughput, lower power consumption, and provide scalable performance for next-generation networking and VLSI-based communication systems.

Keywords: Content Addressable Memory (CAM), Ternary Content Addressable Memory (TCAM), Dynamic Priority, Multi-Key Search, Variable Width TCAM (VW-TCAM), Packet Lookup.

1. Introduction

The rapid growth of digital communication, cloud computing, artificial intelligence, Internet of Things (IoT), and hyperscale data centers has significantly increased the demand for high-speed packet processing and memory lookup systems in modern networking infrastructures. Recent industry reports indicate that the global TCAM memory market reached nearly USD 2.8 billion in 2025 and is expected to grow to approximately USD 6.4 billion by 2034 with a CAGR of 9.8%, driven by the increasing deployment of routers, switches, security appliances, and data center networking equipment. Similarly, the global data center networking market is projected to grow from USD 55.64 billion in 2025 to USD 139.08 billion by 2031 due to the continuous expansion of cloud services, AI workloads, and large-scale enterprise digital transformation. Modern networking systems process billions of packets every day, requiring ultra-fast search operations

with deterministic latency and high throughput. The semiconductor and networking industries have experienced remarkable growth over the last decade due to increasing internet penetration, cloud infrastructure deployment, edge computing, and AI-driven applications. Billions of connected devices continuously exchange information through complex networking systems, generating enormous amounts of packet traffic that must be processed in real time. Modern routers and switches require high-speed lookup engines capable of handling large routing tables, flow classification databases, firewall rules, and quality-of-service policies. TCAM-based architectures have emerged as one of the most important hardware solutions because they enable parallel search operations and deterministic lookup latency, making them highly suitable for high-performance communication systems. Industry studies indicate that TCAM technology remains a critical component in networking equipment, data centers, security devices, and telecommunication infrastructures.

2. Literature Survey

Kishore et al. [1] proposed a comprehensive review of circuit-level leakage reduction techniques for CMOS-based SRAM and TCAM sense amplifiers. The study analysed various leakage mitigation approaches including power gating, transistor stacking, body biasing, and adaptive voltage scaling. The authors compared different sense amplifier structures in terms of leakage power, sensing accuracy, and operational reliability. The review also investigated the impact of technology scaling on leakage characteristics and memory performance. Various design trade-offs between speed, power consumption, and area overhead were discussed to identify suitable low-power memory architectures. The work mainly provides a survey of existing techniques and does not propose a novel hardware architecture for improving TCAM performance. Vicuña et al. [2] proposed a

precharge-free non-volatile TCAM architecture using Voltage-Controlled Magnetic Anisotropy Magnetic Tunnel Junctions (VCMA-MTJs). The design incorporated operational amplifier-based sensing circuits to improve match detection accuracy while eliminating the conventional precharge phase. The architecture also embedded a Physical Unclonable Function (PUF) capability to enhance hardware security. The proposed TCAM utilized non-volatile storage elements to reduce standby power consumption and improve data retention. The use of complex sensing circuitry may increase implementation complexity and fabrication cost.

Mallaiah et al. [3] proposed an error-resilient TCAM architecture designed to improve reliability during memory operations. The architecture introduced specialized error detection and correction mechanisms to identify faults occurring within TCAM cells. The design incorporated redundant structures and fault-tolerant search logic to maintain lookup accuracy. Various fault scenarios were analyzed to evaluate robustness under unreliable operating conditions. Additional fault-tolerance circuitry introduces area overhead and may increase power consumption. Subramanyam et al. [4] proposed a PVT-resilient GNRfet-based TCAM architecture capable of tolerating process, voltage, and temperature variations. The design employed Graphene Nanoribbon Field Effect Transistors to improve electrical characteristics and reduce sensitivity to manufacturing variations. Adaptive design techniques were incorporated to maintain stable operation under different environmental conditions. Simulation studies were performed to evaluate search performance and variation tolerance. The GNRfet technology is still emerging and faces practical manufacturing challenges for large-scale deployment.

Yuan et al. [5] developed a monolithic 3D integrated architecture combining RRAM-based computing-in-memory and SOT-

MRAM-based TCAM for one-shot learning applications. The framework utilized vertical integration to improve computational density and reduce data movement. RRAM arrays were used for learning and inference tasks, while TCAM structures enabled efficient associative search operations. The architecture supported rapid pattern recognition and memory retrieval within a unified platform. The monolithic 3D integration process increases fabrication complexity and manufacturing cost. Nagata et al. [6] proposed a configurable TCAM architecture implemented in 3 nm technology for automotive applications. The design introduced macro-wise pipelined search methods to improve search throughput while minimizing energy consumption. Configurable memory structures were employed to support diverse lookup requirements. The architecture optimized memory density and search latency through advanced pipelining techniques. The highly advanced 3 nm implementation may not be economically feasible for general-purpose applications.

Wang et al. [7] proposed a Gate-Induced Drain Leakage effect-driven 1T FeFET TCAM architecture for large-scale integration. The design exploited leakage characteristics of ferroelectric FET devices to perform ternary matching operations. The proposed single-transistor structure reduced cell area while supporting TCAM functionality. Device-level simulations were conducted to evaluate matching accuracy and integration scalability. Device reliability and long-term stability remain challenging in FeFET-based memory technologies. Huang et al. [8] proposed a reconfigurable TCAM-SRAM fusion array capable of supporting both storage and associative search operations. The architecture integrated SRAM and TCAM functionalities within a unified computing-in-memory platform. Stability enhancement techniques were introduced to improve tolerance against disturbances. The design enabled dynamic reconfiguration between memory and search

modes depending on application requirements. The fusion architecture increases control complexity and may require additional hardware resources.

Kim et al. [9] proposed a 1T-TCAM architecture utilizing binary storage with bias-controlled wildcard generation. The design employed feedback field-effect transistors to achieve ternary matching functionality using a compact memory cell structure. Dynamic bias control mechanisms were used to configure wildcard states efficiently. Device simulations evaluated matching behavior and memory density improvements. The architecture achieved area reduction while maintaining TCAM capabilities. Precise bias control requirements may increase circuit sensitivity and design complexity. Yang et al. [10] proposed dual-channel WSe₂ reconfigurable field-effect transistors integrated with charge-trapping layers for 2T TCAM and reconfigurable logic applications. The design combined memory and logic functionalities within a compact transistor structure. Charge trapping mechanisms enabled programmable device behavior and flexible logic operations. Lee et al. [11] proposed a Ternary Content-Addressable Memory architecture utilizing one capacitor and one nanoelectromechanical memory switch for data-intensive applications. The design employed a compact memory cell structure to reduce transistor count and improve storage density. Nanoelectromechanical switches were integrated to perform ternary matching operations with low leakage characteristics. The architecture supported efficient associative search functionality while minimizing standby power consumption. The mechanical switching elements may introduce reliability concerns and slower switching speeds compared to fully electronic memory technologies. Ren et al. [12] proposed MT-PUF, a reconfigurable intrinsic Physical Unclonable Function integrated with MRAM-based TCAM architecture. The design leveraged the intrinsic randomness of magnetic

tunnel junction devices to generate unique security signatures. Reconfigurable mechanisms were incorporated to improve security flexibility and enhance resistance against cloning attacks. The architecture was evaluated under temperature and voltage variations to verify reliability and stability. Additional security circuitry increases hardware complexity and may introduce area overhead.

Singh et al. [13] proposed a compact 3T1F FeRAM compute cell for in-memory multiply-accumulate operations using partial polarization switching. The architecture utilized ferroelectric memory devices to perform storage and computation within the same cell. Partial polarization switching was employed to achieve efficient analog computation while minimizing energy consumption. The design supported in-memory processing by reducing data movement between memory and processing units. Analog computation accuracy may be affected by device variability and polarization degradation. Yun et al. [14] proposed a Processing-in-Memory architecture for in-storage information retrieval using 3-D NAND flash solid-state drives. The framework integrated computational capabilities directly within storage devices to minimize data transfer overhead. Specialized retrieval algorithms were implemented to accelerate search operations within flash memory arrays. The architecture exploited memory-level parallelism to improve query processing performance. The architecture is application-specific and may not be suitable for general-purpose computing tasks.

Morris et al. [15] proposed NP-CAM, a scalable DNA classification framework based on a Network-on-Chip partitioned CAM architecture. The design divided CAM resources into multiple partitions connected through a NoC infrastructure to improve scalability. DNA sequence matching operations were distributed across partitions for parallel

processing. Efficient communication mechanisms were incorporated to manage data exchange between CAM modules. The NoC communication overhead may increase latency and resource consumption in large deployments. Merlin et al. [16] proposed HardCAM, a radiation-hardened CAM architecture implemented in 65 nm technology for space applications. The design incorporated fault-tolerant techniques to withstand radiation-induced errors and harsh environmental conditions. Specialized circuit-level hardening mechanisms were employed to improve reliability and operational stability. The architecture was evaluated under various radiation scenarios to assess robustness. Radiation-hardening techniques increase circuit area and power consumption.

3. Proposed System

The proposed DPMK-TCAM Architecture as shown in figure 1 for Scalable Ultra-Low-Latency Network Packet Transmission is designed to accelerate packet lookup and forwarding operations in high-speed networking environments. The architecture integrates a Multikey Search Engine, a PDP Multiplexer, and a VW-TCAM to efficiently process incoming packet information. By combining multi-key searching with adaptive priority management and flexible TCAM storage, the architecture reduces lookup latency, improves search accuracy, and enhances throughput. The inclusion of read and write control mechanisms enables dynamic memory updates, while the variable-width TCAM supports different packet header formats and network protocols. This integrated design ensures scalable, low-latency packet transmission suitable for next-generation communication networks, data centers, and software-defined networking infrastructures.

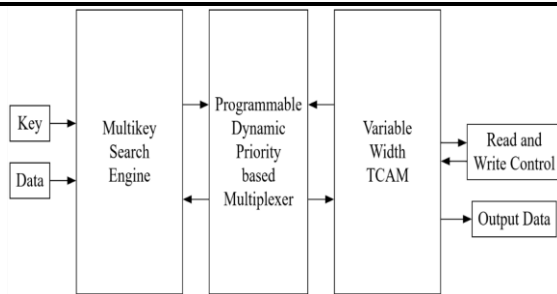


Figure. 1: Proposed system architecture.

Key and Data Input Processing: The operation begins when the key and corresponding packet data are supplied to the Multikey Search Engine. The key represents the search criteria derived from packet headers, routing information, flow identifiers, or classification parameters. Simultaneously, the associated packet data are forwarded into the search engine. The search engine receives these inputs and prepares them for parallel matching operations. Since multiple search parameters can be examined simultaneously, the architecture significantly reduces the time required for packet classification and lookup compared with conventional single-key search mechanisms.

Multikey Search Engine Operation: The Multikey Search Engine performs concurrent matching of multiple packet attributes against stored lookup patterns. Instead of examining a single field, the engine evaluates several search keys simultaneously, enabling efficient packet identification and classification. The search results generated by this module are forwarded to the PDP Multiplexer. The bidirectional communication between these two blocks allows continuous exchange of search information, control signals, and feedback required for dynamic priority assignment and adaptive routing decisions.

PDP Multiplexer: The PDP Multiplexer acts as the intelligent decision-making unit of the architecture. It receives search outcomes from the Multikey Search Engine and dynamically determines the priority of each matching entry. When multiple matches occur simultaneously, the multiplexer resolves conflicts by assigning

priorities according to programmable policies. These policies can be modified based on traffic conditions, quality-of-service requirements, or network management objectives. The dynamic priority mechanism ensures that high-priority packets receive faster access to memory resources, thereby minimizing transmission delays and improving overall network responsiveness.

Interaction Between Multiplexer and VW-TCAM: After determining the appropriate priority levels, the multiplexer communicates with the VW-TCAM through a bidirectional interface. The multiplexer forwards selected search requests and receives matching information from the TCAM. This continuous interaction enables efficient storage access while maintaining flexible priority control. The bidirectional data exchange also allows the multiplexer to update priority decisions based on the matching results returned by the TCAM, ensuring optimal packet handling under varying traffic conditions.

VW-TCAM Processing: The VW-TCAM serves as the primary lookup storage component of the architecture. Unlike fixed-width TCAM structures, this memory supports variable entry widths, allowing efficient storage of different packet header formats and protocol fields. The TCAM performs high-speed parallel comparisons between incoming search patterns and stored entries, generating match results within a single lookup cycle. This flexibility reduces memory wastage, improves storage utilization, and enables the architecture to accommodate diverse networking applications without requiring extensive hardware modifications.

Read and Write Control Operation: The Read and Write Control block manages all memory access operations associated with the VW-TCAM. Through bidirectional communication, it facilitates the insertion, deletion, modification, and retrieval of TCAM entries. Network administrators or control-plane processes can dynamically update routing

tables, forwarding rules, and classification policies without interrupting packet transmission. This capability ensures adaptability to changing network conditions while maintaining continuous operation and high lookup efficiency.

Output Data Generation: Once the appropriate TCAM match is identified and verified, the resulting lookup information is forwarded to the Output Data block. This output may contain forwarding decisions, routing paths, flow classifications, access control actions, or other packet processing instructions. The generated output data are then supplied to subsequent network processing stages, enabling rapid packet forwarding with minimal delay. By combining multikey searching, dynamic priority management, and variable-width TCAM storage, the architecture achieves scalable ultra-low-latency packet transmission while maintaining high throughput and efficient resource utilization.

3.1 Multikey Search Engine

The Multikey Search Engine as shown in Figure 2 is the primary packet lookup component of the proposed architecture, responsible for performing rapid and simultaneous searches using multiple packet attributes. Unlike conventional search engines that process a single key at a time, the Multikey Search Engine evaluates several packet fields concurrently to improve lookup efficiency and reduce processing latency. By supporting parallel matching operations, the module accelerates packet classification, routing table searches, and forwarding decisions. This capability enables the architecture to handle large volumes of network traffic while maintaining high throughput, low latency, and efficient utilization of memory resources. The search engine serves as the first intelligent processing stage, ensuring that relevant packet information is quickly identified before being forwarded to the priority management and TCAM modules.

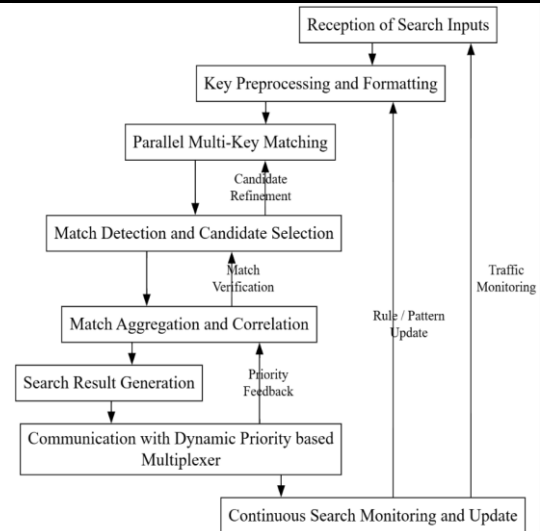


Figure. 2: Multikey search engine flowchart.

Reception of Search Inputs: The operation begins when packet-related information arrives at the search engine. The incoming data contain multiple searchable fields such as source address, destination address, protocol identifier, port number, flow identifier, or application-specific attributes. These fields are extracted and organized into multiple search keys that will be processed simultaneously. The search engine prepares these keys for parallel evaluation to minimize lookup delay.

Key Preprocessing and Formatting: After receiving the search inputs, the search engine performs preprocessing operations to convert the incoming packet attributes into a standardized format suitable for matching. This stage may include key alignment, normalization, field extraction, and pattern preparation. The formatted keys are then distributed across the internal search units to enable efficient parallel processing.

Parallel Multi-Key Matching: Once preprocessing is completed, the search engine activates multiple comparison units to perform concurrent matching operations. Each comparison unit evaluates a specific key against the stored lookup patterns. Since all keys are processed simultaneously, the search engine significantly reduces the total search time compared to sequential search approaches. This parallel matching mechanism allows the

architecture to support high-speed packet processing even under heavy network traffic conditions.

Match Detection and Candidate Selection:

During the comparison process, the search engine identifies all entries that satisfy the matching conditions. Multiple candidate matches may be generated when different search keys correspond to valid entries. The engine records the locations and matching scores of all candidate entries and prepares them for further evaluation. This stage ensures that no valid match is overlooked during the lookup process.

Match Aggregation and Correlation: The detected candidate matches are aggregated and correlated to determine their relationship with the incoming packet. The search engine combines information obtained from different search keys and constructs a comprehensive matching result. This aggregation process improves search accuracy by considering multiple packet characteristics simultaneously rather than relying on a single lookup criterion.

Search Result Generation: After aggregating the matching information, the search engine generates a consolidated search result containing the identified entries, associated metadata, and lookup status information. The generated result represents the most relevant matches corresponding to the incoming packet and is prepared for transmission to the next processing stage.

Communication with the Dynamic Priority Multiplexer: The consolidated search results are transferred to the PDP Multiplexer through the bidirectional communication interface. The multiplexer receives the matching information and evaluates the priority of each candidate entry. Feedback signals may also be returned to the search engine to support adaptive search optimization and priority-aware processing.

Continuous Search Monitoring and Update: The search engine continuously monitors incoming packet streams and repeats the search process for each packet entering the system. It

dynamically adapts to updated lookup rules and modified search patterns supplied through the TCAM control mechanisms. This continuous operation enables real-time packet classification, rapid lookup execution, and scalable performance suitable for ultra-low-latency network packet transmission environments.

3.2 PDP Multiplexer

The PDP Multiplexer as shown in Figure 3 is the intelligent decision-making component of the proposed architecture that manages packet lookup requests and controls access to the VW-TCAM. Its primary function is to dynamically assign priorities to multiple matching entries generated by the Multikey Search Engine and select the most appropriate request for further processing. Unlike conventional multiplexers that operate with fixed selection criteria, this programmable multiplexer adapts its priority policies according to network conditions, traffic demands, quality-of-service requirements, and system objectives. By performing dynamic conflict resolution and adaptive path selection, the module minimizes packet processing delays, improves resource utilization, and ensures efficient handling of high-priority traffic. The bidirectional communication with both the Multikey Search Engine and VW-TCAM enables real-time decision making and continuous optimization of packet forwarding operations.

Reception of Search Results: The operation begins when the PDP Multiplexer receives multiple candidate matches from the Multikey Search Engine. These matches represent valid entries corresponding to the incoming packet's search keys. Since several matches may be generated simultaneously, the multiplexer collects all candidate requests and prepares them for priority evaluation. This stage ensures that all potential lookup results are considered before a final selection is made.

Request Buffering and Organization: After receiving the candidate matches, the multiplexer temporarily stores and organizes

them within internal buffering structures. The incoming requests are categorized based on their associated attributes, lookup identifiers, traffic class information, or forwarding requirements. This organized representation enables efficient priority computation and simplifies the subsequent selection process.

Dynamic Priority Evaluation: The buffered requests are evaluated using programmable priority rules. The multiplexer analyses various parameters such as packet importance, service class, latency requirements, flow characteristics, traffic congestion levels, and administrative policies. Based on these criteria, a dynamic priority value is assigned to each request. Unlike fixed-priority architectures, the assigned priorities can change according to current network conditions, providing greater flexibility and adaptability.

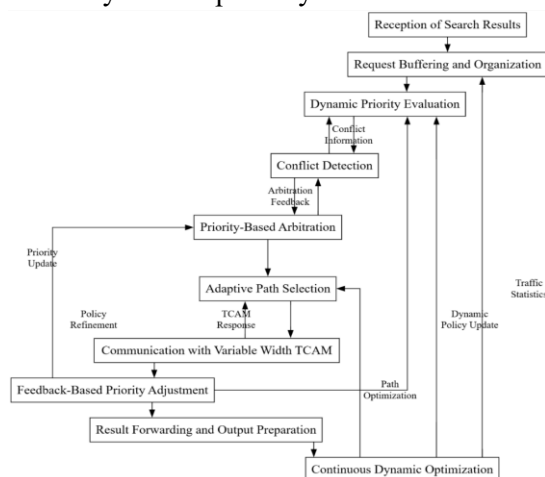


Figure. 3: PDP Multiplexer flowchart.

Conflict Detection: During priority evaluation, multiple requests may compete for the same TCAM resources. The multiplexer detects such conflicts by identifying requests that simultaneously require access to identical memory locations or lookup resources. This conflict detection mechanism prevents resource contention and ensures orderly access to the VW-TCAM.

Priority-Based Arbitration: Once conflicts are identified, the multiplexer performs arbitration to determine which request should receive immediate service. Requests with higher dynamic priority values are selected,

while lower-priority requests are temporarily deferred. This arbitration process guarantees that critical network traffic experiences minimal delay and receives preferential treatment during periods of heavy system activity.

Adaptive Path Selection: Following arbitration, the multiplexer establishes the most efficient processing path for the selected request. The path selection mechanism dynamically determines how the request should be routed toward the VW-TCAM. Depending on system conditions, different access paths may be chosen to reduce bottlenecks and maximize lookup throughput. This adaptive routing capability contributes significantly to achieving ultra-low-latency packet processing.

Communication with VW-TCAM: The selected request is transmitted to the VW-TCAM through the bidirectional communication interface. The multiplexer forwards the lookup information while simultaneously monitoring responses returned by the TCAM. This continuous interaction allows the multiplexer to verify successful memory access and collect matching results generated during the lookup process.

Feedback-Based Priority Adjustment: After receiving responses from the TCAM, the multiplexer analyzes the lookup outcomes and system performance metrics. Feedback information such as lookup latency, resource utilization, queue status, and match success rates is used to refine future priority assignments. This adaptive feedback mechanism enables the system to continuously optimize its decision-making process and respond effectively to changing traffic conditions.

Result Forwarding and Output Preparation: Once the lookup operation is completed, the multiplexer forwards the processed results toward the subsequent stages of packet transmission. The selected TCAM match information, routing decisions, classification outcomes, or forwarding instructions are

prepared for final output generation. This stage ensures that the packet processing pipeline continues without interruption while maintaining efficient data flow throughout the architecture.

Continuous Dynamic Optimization: The multiplexer continuously repeats the priority evaluation, arbitration, and path selection processes for every incoming packet request. Through ongoing monitoring and programmable policy updates, it adapts to varying network workloads and traffic patterns. This continuous optimization capability enables the proposed architecture to maintain scalable performance, high throughput, efficient resource utilization, and ultra-low-latency network packet transmission.

3.3 VW-TCAM Processing

The VW-TCAM Processing module as shown in Figure 4 is the core memory lookup and adaptive storage management component of the proposed architecture. It is responsible for performing flexible-width pattern matching, identifying the best matching entries, managing memory allocation, and maintaining efficient packet classification records. Unlike conventional TCAM architectures that rely on fixed-width entries, the proposed VW-TCAM dynamically accommodates different key lengths and packet formats, thereby improving memory utilization and lookup efficiency. The architecture incorporates matching percentage analysis, dynamic comparison, maximum match identification, and map match line generation to ensure accurate search operations. Furthermore, the inclusion of intelligent memory reservation and feedback mechanisms enables continuous adaptation to changing network conditions, thereby supporting scalable ultra-low-latency packet transmission with high throughput and optimized memory utilization.

Initialization and Input Acquisition: The operation begins with the reception of Reset, Data/Key, and Read/Write (RD/WR) signals. The Reset signal initializes the TCAM

processing unit and clears previous operational states whenever required. The Data/Key input provides the packet header fields, search patterns, or lookup information that must be matched against the TCAM entries. Simultaneously, the RD/WR control signal determines whether the current operation corresponds to a search request, memory update, or storage operation.

Control Logic Processing: After receiving the inputs, the Control Logic block coordinates all subsequent operations within the TCAM architecture. It interprets the incoming control signals, activates the required processing units, and manages the flow of information between different modules. The control logic ensures proper synchronization between lookup, comparison, matching, and storage operations, thereby maintaining stable and efficient system functionality.

Matching Percentage Analysis: The incoming Data/Key values are forwarded to the Matching Percentage Analyzer. This module evaluates the degree of similarity between the search key and the stored TCAM entries. Instead of performing only exact matching, the analyser determines the percentage of matching bits or fields associated with each candidate entry. This matching percentage provides a quantitative measure of similarity that can be used for further prioritization and selection.

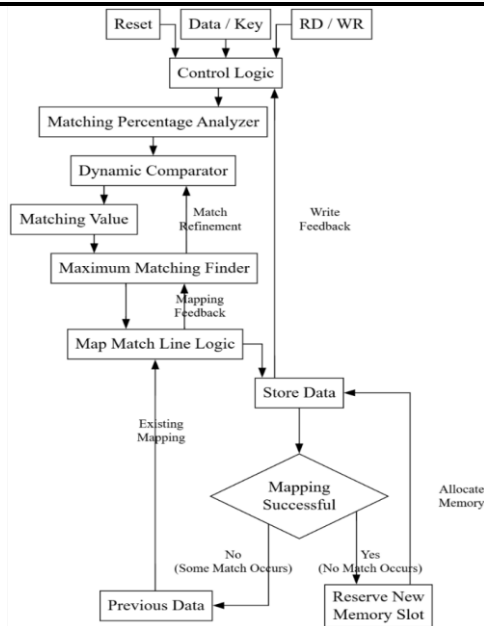


Figure. 4: Proposed VW-TCAM module.

Dynamic Comparison Process: The matching percentage information is transferred to the Dynamic Comparator. This module performs adaptive comparison operations by examining multiple candidate entries and evaluating their relative matching strengths. The comparator dynamically adjusts its comparison criteria according to the lookup requirements and current system conditions. Through this process, the architecture identifies the entries that most closely correspond to the incoming search key.

Matching Value Generation: Following comparison, the Dynamic Comparator generates a Matching Value for each candidate entry. These values represent the quality and relevance of the detected matches. Higher matching values indicate stronger correspondence between the search key and stored TCAM entries. The generated matching values serve as the primary inputs for the subsequent maximum matching selection stage.

Maximum Matching Identification: The Maximum Matching Finder receives all generated matching values and determines the entry that exhibits the highest degree of correspondence with the incoming search key. By selecting the maximum matching value, the architecture ensures that the most relevant entry

is chosen for packet classification and forwarding operations. This stage significantly improves lookup accuracy and minimizes the probability of selecting suboptimal matches.

Match Line Mapping: The selected maximum matching result is forwarded to the Map Match Line Logic module. This block converts the identified match into the corresponding TCAM match line representation. The generated match line establishes the relationship between the search key and the physical memory location containing the matched entry. This mapping process enables efficient retrieval and storage management within the TCAM architecture.

Mapping Feedback Optimization: During the mapping process, feedback information is continuously exchanged between the Map Match Line Logic and Maximum Matching Finder. The Mapping Feedback mechanism verifies the correctness of the selected mapping and refines the match selection whenever necessary. This iterative interaction improves mapping accuracy and ensures reliable lookup performance.

Data Storage Operation: Once the mapping process is completed, the corresponding information is transferred to the Store Data module. This module performs memory write operations and updates the TCAM contents according to the generated mapping results. The stored information may include packet classification records, routing information, forwarding rules, or application-specific lookup entries.

Write Feedback Processing: After data storage is completed, the Store Data module sends Write Feedback signals back to the Control Logic block. This feedback confirms successful memory operations and allows the control unit to update its internal state. The feedback mechanism ensures consistency between lookup operations and memory management activities.

Mapping Verification: Following data storage, the Mapping Successful decision block evaluates whether the generated mapping

operation has been successfully completed. This verification stage determines whether an existing memory location already satisfies the matching requirements or whether additional memory resources must be allocated.

Existing Match Handling: If the decision block determines that some matching entry already exists, the process follows the "No (Some Match Occurs)" path. The corresponding entry is retrieved from the Previous Data module. The existing mapping information is then forwarded back to the Map Match Line Logic block through the Existing Mapping feedback path. This approach avoids unnecessary memory allocation and improves storage efficiency by reusing previously stored entries.

New Memory Allocation: If no suitable match is found, the process follows the "Yes (No Match Occurs)" path. The Reserve New Memory Slot module allocates a new memory location within the VW-TCAM. This newly reserved slot provides storage space for the incoming data and enables the architecture to dynamically expand its lookup database when required.

Memory Reservation Feedback: After reserving a new memory slot, Allocate Memory information is sent back to the Store Data module. The new entry is then stored in the allocated location, ensuring that future lookup operations can access the newly inserted information. This feedback loop enables efficient memory growth while maintaining consistent storage management.

Continuous Adaptive TCAM Operation: The entire process continuously repeats for every incoming packet, search key, or memory update request. Through the combined operation of matching percentage analysis, dynamic comparison, maximum match identification, mapping verification, adaptive memory allocation, and multiple feedback mechanisms, the proposed VW-TCAM achieves highly efficient lookup performance, optimized memory utilization, scalable storage

management, and ultra-low-latency packet processing suitable for next-generation networking applications.

4. Results and Discussions

Figure 5 presents the simulation waveform of the proposed Dynamic Priority and Multi-Key TCAM architecture. The waveform illustrates the behavior of key signals including the clock (clk), address input address [7:0], write enable (we), output enable (oe), input data data[16:1], error output error[23:1], stored data stored_data[23:1], and output data out[16:1]. During simulation, the address values transition through 12, 14, 18, and subsequently repeat, while the write enables signal remains active, allowing memory updates. The input data initially changes from 10 to 20 and later to 30, demonstrating successful storage operations. Correspondingly, the stored_data signal updates from 3735682 to 5636953 and finally to 7274821, confirming correct memory write functionality. The output signal initially remains undefined (X) before stabilizing to valid outputs of 10 and 20 after successful memory retrieval operations. The waveform verifies that the proposed architecture correctly performs memory access, data storage, and retrieval operations while maintaining stable timing behavior throughout the simulation period.



Figure 5: Proposed simulation outcome.

Figure 6 illustrates the FPGA resource utilization summary of the proposed architecture. The implementation requires only 23 LUTs out of the available 134,600 LUTs, corresponding to an extremely low utilization of 0.02%, indicating a highly area-efficient design. The architecture utilizes 0.50 BRAM blocks from the available 365 BRAM resources, resulting in 0.14% utilization,

identical to the existing implementation. The design uses 89 IO pins out of 500 available pins, corresponding to 17.80% utilization, which represents the largest hardware resource consumption in the architecture. Additionally, a single BUFG resource is employed from the available 32 global clock buffers, resulting in 3.13% utilization. The results demonstrate that the proposed architecture significantly reduces LUT utilization from 59 to 23 while maintaining similar memory and clock resource requirements, thereby achieving a more compact hardware implementation.

Resource	Estimation	Available	Utilization %
LUT	23	134600	0.02
BRAM	0.50	365	0.14
IO	89	500	17.80
BUFG	1	32	3.13

Figure 6: Proposed area outcome

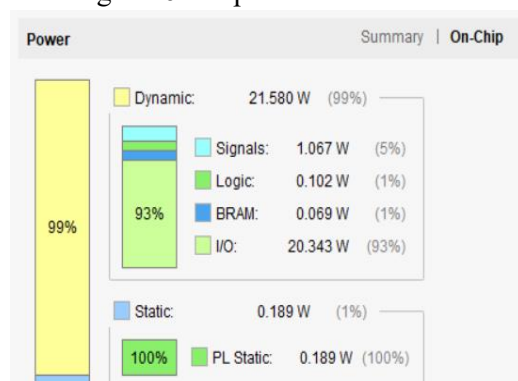
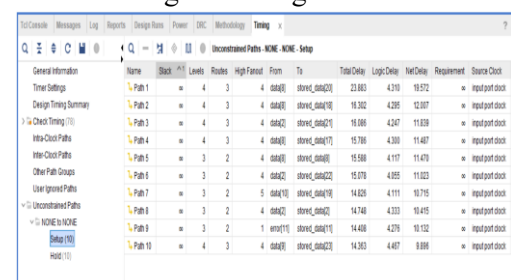


Figure 7: Proposed Power Summary

Figure 7 presents the on-chip power consumption characteristics of the proposed architecture. The dynamic power consumption is reported as 21.580 W, accounting for 99% of the total power, while the static power consumption is only 0.189 W, contributing the remaining 1%. Within the dynamic power component, IO resources consume 20.343 W, representing the dominant share of 93%. Signal routing consumes 1.067 W, accounting for 5% of the dynamic power. Logic resources consume only 0.102 W, while BRAM contributes 0.069 W, with both representing approximately 1% each of the total dynamic power consumption. The PL static power is measured at 0.189 W. Compared to the existing

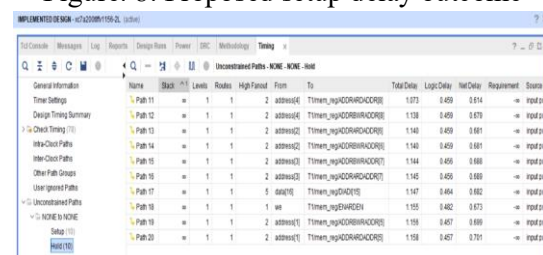
architecture, the proposed design demonstrates reductions in dynamic power, logic power, signal power, and static power, confirming that the optimized architecture achieves improved energy efficiency while maintaining functional performance.

Figure 8 shows the setup timing analysis of the proposed architecture under unconstrained path conditions. The most critical setup path, Path 1, exhibits a total delay of 23.883 ns, consisting of 4.310 ns logic delay and 19.572 ns net delay between data [8] and stored_data [20]. Path 2 records a total delay of 16.302 ns with 4.295 ns logic delay and 12.007 ns net delay. Similarly, Paths 3 to 10 exhibit total delays of 16.086 ns, 15.786 ns, 15.588 ns, 15.078 ns, 14.826 ns, 14.748 ns, 14.408 ns, and 14.363 ns, respectively. The corresponding net delays range from 9.896 ns to 11.839 ns, while logic delays remain approximately between 4.055 ns and 4.467 ns. Most timing paths consist of 3–4 logic levels and 2–4 routing segments, with fanout values ranging from 1 to 5. The analysis indicates that interconnect propagation remains the dominant contributor to setup delay; however, the majority of timing paths exhibit lower delays than the existing design, demonstrating improved timing optimization and efficient signal routing.



Path	Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
Path 1	data[8]	-	4	3	4	data[8]	stored_data[20]	23.883	4.310	19.572	19.572	input_pclk
Path 2	data[8]	-	4	3	4	data[8]	stored_data[20]	16.302	4.295	12.007	12.007	input_pclk
Path 3	data[8]	-	4	3	4	data[8]	stored_data[20]	16.086	4.247	11.839	11.839	input_pclk
Path 4	data[8]	-	4	3	4	data[8]	stored_data[20]	15.786	4.200	11.586	11.586	input_pclk
Path 5	data[8]	-	3	2	4	data[8]	stored_data[20]	15.588	4.157	11.431	11.431	input_pclk
Path 6	data[8]	-	3	2	4	data[8]	stored_data[20]	15.078	4.055	11.023	11.023	input_pclk
Path 7	data[8]	-	3	2	5	data[8]	stored_data[20]	14.826	4.111	10.715	10.715	input_pclk
Path 8	data[8]	-	3	2	4	data[8]	stored_data[20]	14.748	4.033	10.715	10.715	input_pclk
Path 9	data[8]	-	3	2	1	data[8]	stored_data[20]	14.408	4.079	10.329	10.329	input_pclk
Path 10	data[8]	-	4	3	4	data[8]	stored_data[20]	14.363	4.467	9.896	9.896	input_pclk

Figure 8: Proposed setup delay outcome



Path	Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
Path 11	data[8]	-	1	1	2	data[8]	stored_data[20]	1.073	0.459	0.614	0.614	input_pclk
Path 12	data[8]	-	1	1	2	data[8]	stored_data[20]	1.138	0.459	0.679	0.679	input_pclk
Path 13	data[8]	-	1	1	2	data[8]	stored_data[20]	1.140	0.459	0.681	0.681	input_pclk
Path 14	data[8]	-	1	1	2	data[8]	stored_data[20]	1.140	0.459	0.681	0.681	input_pclk
Path 15	data[8]	-	1	1	2	data[8]	stored_data[20]	1.140	0.459	0.681	0.681	input_pclk
Path 16	data[8]	-	1	1	2	data[8]	stored_data[20]	1.140	0.459	0.681	0.681	input_pclk
Path 17	data[8]	-	1	1	5	data[8]	stored_data[20]	1.147	0.459	0.688	0.688	input_pclk
Path 18	data[8]	-	1	1	1	data[8]	stored_data[20]	1.155	0.462	0.693	0.693	input_pclk
Path 19	data[8]	-	1	1	2	data[8]	stored_data[20]	1.158	0.457	0.691	0.691	input_pclk
Path 20	data[8]	-	1	1	2	data[8]	stored_data[20]	1.158	0.457	0.701	0.701	input_pclk

Figure 9: Proposed Hold Delay Outcome.

Figure 9 presents the hold timing analysis of the proposed architecture. The timing report lists ten critical hold paths connecting address signals and memory register destinations. Path 11 exhibits a total delay of 1.073 ns, including 0.459 ns logic delay and 0.614 ns net delay. Path 12 records a total delay of 1.138 ns with 0.459 ns logic delay and 0.679 ns net delay. Paths 13 through 20 exhibit total delays of 1.140 ns, 1.140 ns, 1.144 ns, 1.145 ns, 1.147 ns, 1.155 ns, 1.156 ns, and 1.158 ns, respectively. The logic delays vary only slightly between 0.456 ns and 0.482 ns, while net delays range from 0.614 ns to 0.701 ns. All hold paths contain a single logic level and one routing segment, with fanout values between 1 and 5. The close distribution of hold delays around 1.1 ns demonstrate highly balanced short-path timing characteristics and stable synchronization behaviour. The results confirm that the proposed architecture satisfies hold timing requirements while maintaining reliable memory addressing and data transfer operations.

4.1 Comparative Analysis

Table 1 compares the FPGA area utilization of the existing and proposed architectures. The results show that the proposed architecture requires only 23 LUTs compared to 59 LUTs utilized by the existing architecture, achieving a significant improvement of 61.02%. This substantial reduction demonstrates that the proposed Dynamic Priority and Multi-Key TCAM architecture employs a more compact and efficient hardware design, thereby lowering logic resource requirements and reducing implementation complexity. Similarly, the LUT utilization percentage decreases from 0.04% in the existing architecture to 0.02% in the proposed architecture, representing a 50.00% reduction in FPGA resource occupancy. These results indicate that the proposed design achieves superior area efficiency while maintaining the required functionality, making it more suitable for scalable FPGA and ASIC implementations

where hardware resources are limited and optimization of chip area is a critical design objective.

Table 1. Area comparison between existing and proposed architectures

Resource Metric	Existing Architecture	Proposed Architecture	Improvement (%)
LUT Utilization	59	23	61.02
LUT Utilization (%)	0.04	0.02	50.00

Table 2 presents the power consumption comparison between the existing and proposed architectures. The proposed architecture reduces dynamic power consumption from 23.108 to 21.580, achieving an improvement of 6.61%, while static power decreases from 0.199 to 0.189, resulting in a 5.03% improvement. Significant reductions are observed in signal power and logic power, where the proposed architecture lowers signal power from 2.763 to 1.067 and logic power from 0.733 to 0.102, corresponding to improvements of 61.38% and 86.08%, respectively. BRAM power consumption is also slightly reduced from 0.071 to 0.069, yielding a 2.82% improvement. Consequently, the total power consumption decreases from 23.307 to 21.769, providing an overall improvement of 6.60%. These results demonstrate that the proposed architecture effectively minimizes switching activity and logic overhead, thereby enhancing energy efficiency and making the design more suitable for high-performance networking systems requiring reduced power dissipation.

Table 2. Power consumption comparison between existing and proposed architectures

Power Metric (uW)	Existing Architecture	Proposed Architecture	Improvement (%)

Dynamic Power	23.108	21.580	6.61
Static Power	0.199	0.189	5.03
Signal Power	2.763	1.067	61.38
Logic Power	0.733	0.102	86.08
BRAM Power	0.071	0.069	2.82
Total Power	23.307	21.769	6.60

Table 3 evaluates the setup timing performance of the existing and proposed architectures. The maximum setup delay increases from 19.274 ns to 23.883 ns, resulting in a negative improvement of -23.91%, indicating the presence of one critical timing path with higher delay in the proposed design. However, the minimum setup delay improves significantly from 17.128 ns to 14.363 ns, achieving a 16.14% reduction. More importantly, the average setup delay decreases from 17.749 ns in the existing architecture to 16.507 ns in the proposed architecture, providing a 7.00% improvement in overall timing performance. The average net delay is reduced from 13.590 ns to 12.223 ns, corresponding to a 10.06% improvement, demonstrating more efficient signal routing and reduced interconnect overhead. Although the average logic delay increases slightly from 4.159 ns to 4.284 ns, the overall reduction in average setup delay confirms that the proposed architecture provides faster data propagation and improved timing characteristics for most operational paths.

Table 3. Setup Delay Comparison Between Existing and Proposed Architectures

Delay Metric (ns)	Existing Architecture	Proposed Architecture	Improvement (%)
Dynamic Power	23.108	21.580	6.61
Static Power	0.199	0.189	5.03
Signal Power	2.763	1.067	61.38
Logic Power	0.733	0.102	86.08
BRAM Power	0.071	0.069	2.82
Total Power	23.307	21.769	6.60

Minimum Setup Delay	17.128	14.363	16.14
Average Setup Delay	17.749	16.507	7.00
Average Net Delay	13.590	12.223	10.06

Table 6.4 compares the hold delay characteristics of the two architectures. The minimum hold delay decreases slightly from 1.086 ns to 1.073 ns, resulting in an improvement of 1.20%, indicating marginally faster short-path signal propagation in the proposed architecture. Furthermore, the average logic delay remains almost unchanged, decreasing from 0.461 ns to 0.460 ns and achieving an improvement of 0.22%. The negligible variation in hold timing parameters demonstrates that the proposed architecture maintains stable short-path timing behavior while preserving reliable synchronization between memory and logic components.

Table 4. Hold delay comparison between existing and proposed architectures

Delay Metric (ns)	Existing Architecture	Proposed Architecture	Improvement (%)
Minimum Hold Delay	1.086	1.073	1.20
Average Logic Delay	0.461	0.460	0.22

5. Conclusion

The proposed DPMK-TCAM Architecture for Scalable Ultra-Low-Latency Network Packet Transmission successfully improves hardware efficiency, power consumption, and timing performance compared to the existing CAM-based architecture. Experimental FPGA implementation results demonstrate that the

proposed design reduces LUT utilization from 59 to 23, achieving a significant 61.02% improvement, while LUT utilization percentage decreases from 0.04% to 0.02%. In terms of power consumption, the proposed architecture lowers dynamic power from 23.108 to 21.580, static power from 0.199 to 0.189, signal power from 2.763 to 1.067, logic power from 0.733 to 0.102, and total power from 23.307 to 21.769, providing overall power savings of 6.60%. The timing analysis further shows that the average setup delay decreases from 17.749 ns to 16.507 ns, achieving a 7.00% improvement, while the average net delay is reduced from 13.590 ns to 12.223 ns, resulting in a 10.06% improvement. Additionally, the minimum hold delay improves from 1.086 ns to 1.073 ns and the average logic delay decreases from 0.461 ns to 0.460 ns. These results confirm that the integration of the Multikey Search Engine, Programmable Dynamic Priority based Multiplexer, and Variable Width TCAM effectively enhances resource utilization, reduces power dissipation, improves lookup efficiency, and provides a scalable low-latency solution for next-generation networking and VLSI-based communication systems.

References

- [1]. Kishore, Devesh, Saswata Chakraborty, Arindam Biswas, and Ankur Dumka. "Circuit-Level leakage reduction techniques in CMOS sense amplifiers for SRAM and TCAM: A review." *Analog Integrated Circuits and Signal Processing* 126, no. 2 (2026): 37.
- [2]. Vicuña, Kevin, Marco Villegas, and Lionel Trojman. "Precharge-Free Non-Volatile TCAM Using VCMA-MTJs with Op-Amp-Based Sensing and Embedded PUF Functionality." In *2026 International VLSI Symposium on Technology, Systems and Applications (VLSI TSA)*, pp. 1-4. IEEE, 2026.
- [3]. Mallaiah, Sirisha, and M. Vinodhini. "An Efficient Error Resilient Ternary Content Addressable Memory Architecture." *Journal of VLSI Circuits and Systems* 8, no. 1 (2026): 1-8.
- [4]. Subramanyam, M. V., S. Rambabu, E. Subbarayudu, K. Ajay Kumar, C. Bhanu Teja, and K. Nagesh. "A PVT-Resilient Architecture for an Effective, Variation-Tolerant GNR-FET-Based Ternary Content Addressable Memory." In *2026 6th International Conference on Expert Clouds and Applications (ICOECA)*, pp. 438-445. IEEE, 2026.
- [5]. Yuan, Jian, Jingkai Xu, Aitian Chen, Yuyan Wang, Wensi Yue, Shixuan Liang, Ran An et al. "First M3D Integration of RRAM-based CIM and SOT-MRAM-based TCAM for One-Shot Learning." In *2026 10th IEEE Electron Devices Technology & Manufacturing Conference (EDTM)*, pp. 1-3. IEEE, 2026.
- [6]. Nagata, Shunya, Yohei Sawada, Kenichiro Takiguchi, Masao Morimoto, Shinji Tanaka, Naoya Fujita, Tomohiro Miura, Daisuke Nakamura, and Takashi Ito. "A 3nm 0.167 fJ/b 5.27 Mb/mm² Configurable TCAM with Macro-Wise Pipelined Search Methods for Automotive Applications." In *2026 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 69, pp. 262-264. IEEE, 2026.
- [7]. Wang, Xinghui, Jiuren Zhou, Siying Zheng, Haiwen Xu, Wenxin Sun, Zhou Wang, Jie Liang et al. "Gate-Induced Drain Leakage Effect-Driven 1T FeFET TCAM for Large-Scale Integration." *IEEE Transactions on Electron Devices* (2026).
- [8]. Huang, P. H., K. H. Chang, and E. Ray Hsieh. "Highly-stable and Disturbing-

- tolerant Reconfigurable TCAM-SRAM fusion array for Bit-storage and TCAM Multiple-functionality in Computing-in-memory." In 2026 10th IEEE Electron Devices Technology & Manufacturing Conference (EDTM), pp. 1-3. IEEE, 2026.
- [9]. Kim, Sujin, Junyoung Park, Doohyeok Lim, and Sola Woo. "1T-TCAM with binary storage enabling ternary matching via bias-controlled wildcards using a feedback field-effect transistor." *Nanotechnology* 37, no. 13 (2026): 135202.
- [10]. Yang, Eunyeong, Jiwon Ma, Changwook Lee, and Jiwon Chang. "Large-Scale Dual-Channel WSe₂ Reconfigurable Field-Effect Transistors with Charge-Trapping Layer for 2T TCAM and Reconfigurable Logic." *Nano Letters* (2026).
- [11]. Lee, Jin Wook, Changha Kim, Geun Tae Park, Seung Hun Baek, Myeong Su Shin, and Woo Young Choi. "Ternary Content-Addressable Memory Using One Capacitor and One Nanoelectromechanical Memory Switch for Data-Intensive Applications." *Advanced Intelligent Systems* 8, no. 3 (2026): 2500586.
- [12]. Ren, Shimao, Pengjun Wang, Xiangyu Li, Bo Chen, Xinrong Yang, and Yongbin Cai. "MT-PUF: A reconfigurable intrinsic PUF for MRAM TCAM achieving 0.07%/10°C and 0.44%/0.1 V sensitivity." *Integration* (2026): 102726.
- [13]. Singh, Vikrant, Anand Bulusu, and Girish Pahwa. "A Compact 3T1F FeRAM Compute Cell for In-Memory MAC Using Partial Polarization Switching." In 2026 International VLSI Symposium on Technology, Systems and Applications (VLSI TSA), pp. 1-2. IEEE, 2026.
- [14]. Yun, Huiwon, Inho Jeong, Kyoyun Lee, Jaeyong Lee, Myoungjun Chun, Jihong Kim, and Dongsuk Jeon. "Processing-in-Memory Architecture for In-Storage Information Retrieval on 3-D NAND Flash Solid-State Drives." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* (2026).
- [15]. Morris, Benjamin F., Tergel Molom-Ochir, Changchun Zhou, Yiran Chen, Alex K. Jones, and Hai Li. "NP-CAM: Efficient and Scalable DNA Classification using a NoC-Partitioned CAM Architecture." In 2026 IEEE International Symposium on High Performance Computer Architecture (HPCA), pp. 1-14. IEEE, 2026.
- [16]. Merlin, Itay, Yuval Harary, Daniel Amar, Eyal Rechef, Avner Haran, Eran Mazal Tov, Alexander Fish, and Leonid Yavits. "HardCAM: A 65 nm radhard CAM design and evaluation for space applications." *Array* (2026): 100769.