AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.6, NO. 3(2026)

Low-Power CAM-Based In-Memory Multiplier with Operand Bypass Optimization

Prof A K Rathod [1], Aku Shireesha[2]

[1*]Assistant Professor, Department of ECE, SVS Group of Institutions, Hanumakonda, Telangana

[2*]M.Tech Student, Department Of ECE, SVS Group of Institutions, Hanumakonda, Telangana

Abstract

This work presents an enhanced in-memory computation (IMC) based multiplier architecture incorporating a power-aware operand activity detection mechanism for further reduction of dynamic power consumption. The conventional IMC multiplier utilizes Content Addressable Memory (CAM), Block RAM (BRAM), and associative processing to perform multiplication through address mapping and memory-based computation, thereby minimizing data movement and eliminating the need for a dedicated arithmetic logic unit. In the proposed extension, an operand activity detector is introduced to identify trivial multiplication conditions such as multiplication by zero and multiplication by one. For these cases, the computation is bypassed and the output is generated directly without activating the CAM search and associative processing stages. This approach significantly reduces unnecessary switching activity, memory access operations, and computational overhead. The architecture is implemented using Verilog HDL and targeted for FPGA realization using Xilinx Vivado. Simulation results demonstrate correct functionality while achieving improved energy efficiency compared to the original IMC multiplier architecture. The proposed design preserves computational accuracy and offers a simple yet effective low-power enhancement suitable for embedded systems, edge devices, and energy-constrained digital signal processing applications.

Keywords: In-Memory Computation (IMC), Content Addressable Memory (CAM), FPGA, Low-Power Design, Associative Processor, Operand Activity Detection, BRAM, Energy-Efficient Multiplier.

1. Introduction

The rapid growth of embedded systems, artificial intelligence accelerators, Internet of Things (IoT) devices, and edge computing platforms has significantly increased the demand for energy-efficient arithmetic processing units. Multiplication is one of the most frequently executed arithmetic operations in digital signal processing (DSP), image processing, machine learning, communication systems, and scientific computing. Conventional multipliers rely on arithmetic logic units (ALUs) that generate partial products and perform accumulation operations, resulting in considerable switching activity, area overhead, and power consumption [1].

In-memory computing (IMC) has emerged as a promising solution to overcome the limitations of the traditional von Neumann architecture, where data transfer between memory and processing units contributes significantly to power dissipation and latency. By integrating computation directly within memory structures, IMC reduces data movement and improves overall system efficiency [2].

Content Addressable Memory (CAM)-based in-memory multiplication represents an innovative approach where multiplication results are generated through associative searching and memory look-up operations rather than arithmetic calculations. The multiplier stores precomputed results in memory and retrieves the corresponding output using CAM-based address matching mechanisms. This approach minimizes computational complexity and provides faster operation for specific applications [3].

Although CAM-based multipliers reduce arithmetic processing overhead, they still consume dynamic power due to repetitive memory access and search operations, even when the multiplication involves trivial operands such as 0 or 1. In many practical DSP and embedded applications, these special cases occur frequently. Activating the complete CAM search process for such operations leads to unnecessary switching activity and energy wastage.

To address this challenge, this work proposes a Low-Power CAM-Based In-Memory Multiplier with Operand Bypass Optimization. The proposed architecture incorporates an Operand Activity Detector (OAD) capable of identifying multiplication-by-zero and multiplication-by-one conditions before the CAM search process begins. When such conditions are detected, the system bypasses CAM activation and directly generates the output, thereby reducing memory access operations and dynamic power consumption while preserving computational accuracy [4].

1.1 Background

The increasing complexity of modern digital systems demands efficient computational architectures capable of delivering high performance while maintaining low power consumption. Multipliers are among the most power-intensive components within digital processors because of extensive partial product generation and accumulation stages.

Recent advancements in memory technologies have enabled the development of in-memory computing architectures that perform arithmetic operations within memory arrays. Associative processors employing CAM structures provide fast search capabilities and support memory-based computation without conventional arithmetic units.

However, even memory-based multipliers exhibit redundant operations under specific operand conditions. The proposed work addresses this inefficiency by introducing intelligent operand-aware computation control.

1.2 Need for the Study

The proposed work is motivated by the growing demand for low-power arithmetic units in battery-operated and edge computing devices. Frequent memory accesses and unnecessary CAM search operations increase dynamic power consumption, reducing overall energy efficiency. Therefore, there is a need for FPGA-friendly, energy-efficient DSP architectures that leverage in-memory computing techniques to minimize power usage while maintaining computational performance.

1.3 Objectives of the Study

The primary objective of this study is to develop a low-power CAM-based in-memory multiplier architecture by incorporating an Operand Activity Detector and bypass logic to

identify trivial multiplication cases. The proposed design aims to reduce switching activity and dynamic power consumption while maintaining computational efficiency. The architecture is implemented using Verilog HDL, validated through FPGA simulation and synthesis, and evaluated against existing CAM-based multiplier architectures in terms of power efficiency and overall performance.

1.4 Scope of the Study

This study focuses on the design and implementation of a CAM-based in-memory multiplier with operand activity monitoring and bypass optimization techniques. The proposed architecture is developed and realized on an FPGA platform using Xilinx Vivado, followed by functional verification through simulation. Furthermore, performance evaluation and power consumption analysis are carried out to assess the effectiveness of the proposed low-power design.

2. Literature Survey

The increasing demand for energy-efficient digital systems has motivated extensive research in low-power arithmetic architectures. Multipliers are fundamental building blocks in digital signal processing, image processing, communication systems, artificial intelligence accelerators, and embedded applications. Since multiplication operations contribute significantly to overall power consumption and computational complexity, researchers have explored various techniques to improve multiplier efficiency. Recent developments in Content Addressable Memory (CAM), in-memory computing (IMC), associative processing, and low-power optimization methods have provided alternative approaches to conventional arithmetic multipliers. This chapter reviews significant research contributions related to CAM-based computation, memory-centric processing, low-power multiplier design, and operand-aware optimization techniques. The limitations of existing approaches are analyzed to identify the research gap addressed by the proposed Low-Power CAM-Based In-Memory Multiplier with Operand Bypass Optimization [5].

The authors presented a comprehensive survey of Content Addressable Memory (CAM) architectures and their applications in high-speed search operations. Unlike conventional memory systems that access data through addresses, CAM retrieves information by comparing input search data against all stored entries simultaneously. The study demonstrated that CAM structures significantly improve search speed and are suitable for networking, packet routing, database management, and associative computing applications.

The authors discussed various CAM architectures, including binary CAM and ternary CAM, and analyzed their performance characteristics. Although CAM-based systems offer exceptional search performance, the parallel comparison process activates a large number of comparison circuits simultaneously, resulting in substantial dynamic power consumption. The study identified power dissipation as one of the major challenges limiting large-scale CAM deployment. The findings highlighted the need for power-aware control mechanisms capable of reducing unnecessary CAM activity, which forms an important motivation for the proposed operand bypass optimization technique [6].

In-memory computing architectures are investigated aimed at overcoming the limitations of the traditional von Neumann computing model. The study focused on reducing the energy and latency associated with data movement between processors and memory units. By embedding computational capabilities directly within memory arrays, the proposed architecture significantly reduced communication overhead and improved overall system performance.

The researchers implemented arithmetic operations through memory-based lookup techniques and demonstrated notable reductions in energy consumption compared with conventional processor-centric systems. Experimental results showed that in-memory computing could improve throughput while lowering power requirements for data-intensive applications. However, the architecture still performed memory access operations for every computation regardless of operand significance. The absence of operand-aware control mechanisms resulted in unnecessary memory activity during trivial computations. This limitation suggests opportunities for further optimization through intelligent operation bypassing, as proposed in the present work [7].

The FPGA-based multiplier architecture that utilizes memory lookup tables instead of traditional arithmetic multiplication units. The multiplication results were precomputed and stored in Block RAM (BRAM), allowing rapid retrieval during execution. The architecture achieved high-speed operation due to the elimination of partial product generation and accumulation stages. The implementation demonstrated improved computational performance and reduced logic utilization on FPGA platforms. Memory-based multiplication was shown to be particularly effective for fixed operand widths and repetitive computational workloads. Despite these advantages, the study revealed that memory accesses occurred for every multiplication operation, including cases involving multiplication by zero or one. Such redundant accesses increased switching activity and power consumption. The authors suggested that future work should explore adaptive memory activation mechanisms to improve energy efficiency. This recommendation directly aligns with the operand activity detection approach introduced in the proposed architecture [8].

The authors introduced operand isolation techniques to reduce switching activity in arithmetic and digital processing circuits. Operand isolation works by preventing unnecessary signal transitions from propagating through inactive portions of a circuit. The researchers implemented gating structures that selectively disable computational blocks when their outputs are not required. Simulation results demonstrated significant reductions in dynamic power consumption across various arithmetic units, including adders, multipliers, and digital signal processing modules. The study showed that switching activity is directly related to energy dissipation and that selective activation can effectively reduce power consumption without affecting functional correctness. However, the proposed technique primarily targeted conventional arithmetic datapaths and did not address memory-based computing architectures. Extending operand-aware optimization concepts to CAM-based in-memory multipliers represents a promising direction for achieving further energy savings [9].

The authors developed an associative processing framework capable of performing arithmetic operations within memory arrays using CAM structures. Their architecture employed parallel

comparison operations and associative logic to execute multiplication and other arithmetic functions without dedicated arithmetic units. The approach demonstrated reduced computational latency and improved parallelism for large-scale data processing applications.

Experimental evaluation showed that associative processing architectures could effectively accelerate data-intensive workloads. Nevertheless, the system required activation of CAM comparison circuitry for every multiplication request. The repeated search operations contributed to increased dynamic power consumption, especially when trivial operands were frequently encountered. The study acknowledged that future low-power designs should incorporate intelligent control strategies to minimize unnecessary memory searches. This observation forms a key foundation for the proposed operand bypass optimization mechanism [10].

2.1 Existing Systems

Existing CAM-based and in-memory multiplier architectures generally rely on memory lookup operations and associative searches to generate multiplication results. These systems successfully eliminate conventional arithmetic units and reduce computational complexity. Most existing designs utilize CAM structures for address matching and BRAM modules for result retrieval. While such architectures improve speed and hardware efficiency, they typically perform memory searches for every operation irrespective of operand characteristics [11].

Current implementations lack mechanisms to identify trivial multiplication conditions such as multiplication by zero or one. As a result, memory arrays, CAM search engines, and associative processing units remain active even when the final result can be generated directly. This unnecessary activation increases switching activity and contributes to higher dynamic power consumption [12].

2.2 Research Gap

The literature review indicates that existing CAM-based multiplier architectures suffer from unnecessary memory accesses and CAM search operations, even when multiplication results are easily predictable. Most current approaches lack operand activity detection and intelligent bypass mechanisms, leading to increased switching activity and dynamic power consumption. Furthermore, existing low-power optimization techniques primarily target arithmetic datapaths rather than memory-centric computing architectures. As a result, there is a need for an FPGA-friendly CAM-based multiplier that incorporates operand-aware computation bypassing to improve energy efficiency and reduce redundant operations [13].

2.3 Problem Statement

Although CAM-based in-memory multipliers eliminate the need for dedicated arithmetic hardware and reduce computational complexity, they continue to consume considerable dynamic power because CAM search operations and memory accesses are performed for every multiplication task. In practical embedded and DSP applications, multiplication by zero and multiplication by one occur frequently. Performing complete associative searches for such trivial operations results in unnecessary switching activity, increased energy consumption, and reduced overall efficiency. Therefore, an intelligent low-power

optimization technique is required to identify trivial operand conditions and bypass memory-intensive computation stages whenever possible [14].

2.4 Advancements in the Proposed System

The proposed Low-Power CAM-Based In-Memory Multiplier enhances conventional architectures by integrating an Operand Activity Detector (OAD) that identifies trivial multiplication cases such as multiplication by zero and one. By generating outputs directly for these operations, the design conditionally activates CAM search circuitry only when necessary, thereby reducing memory accesses, switching activity, and dynamic power consumption [15]. The architecture preserves exact multiplication accuracy, is efficiently implemented using Verilog HDL, and is well suited for FPGA-based low-power embedded, DSP, and IoT applications and the comparison of previous studies are stated in Table 1.

Table 1: Comparative Analysis of Previous Studies

Study	Methodology	Advantages	Limitations
[16]	CAM Architecture	Fast associative search	High power consumption
[17]	In-Memory Computing	Reduced data movement	Unnecessary memory accesses
[18]	BRAM Lookup Multiplier	Fast FPGA implementation	No operand-aware optimization
[19]	Operand Isolation	Reduced switching activity	Not applied to IMC architectures
[20]	Associative Processing	High parallelism	CAM active for all operations
Proposed Work	CAM-Based IMC with Operand Bypass	Reduced CAM activity, lower power, exact accuracy	Limited to predefined operand detection rules

3. Proposed System Architecture

The proposed Low-Power CAM-Based In-Memory Multiplier with Operand Bypass Optimization is designed to improve the energy efficiency of memory-centric multiplication systems by eliminating unnecessary computation and memory access activities. Conventional CAM-based in-memory multipliers perform associative searches and memory lookup operations for every multiplication request regardless of the operand values. However, in practical applications, multiplication by zero and multiplication by one occur frequently. Activating CAM search circuits and memory arrays for such operations leads to unnecessary switching activity and increased dynamic power consumption.

To address this limitation, the proposed architecture introduces an Operand Activity Detector (OAD) that examines the input operands before initiating CAM-based computation. If a trivial multiplication condition is detected, the system bypasses the CAM search engine and directly generates the output. Otherwise, normal in-memory multiplication is performed through associative search and memory lookup operations. This optimization reduces dynamic power consumption while maintaining complete computational accuracy.

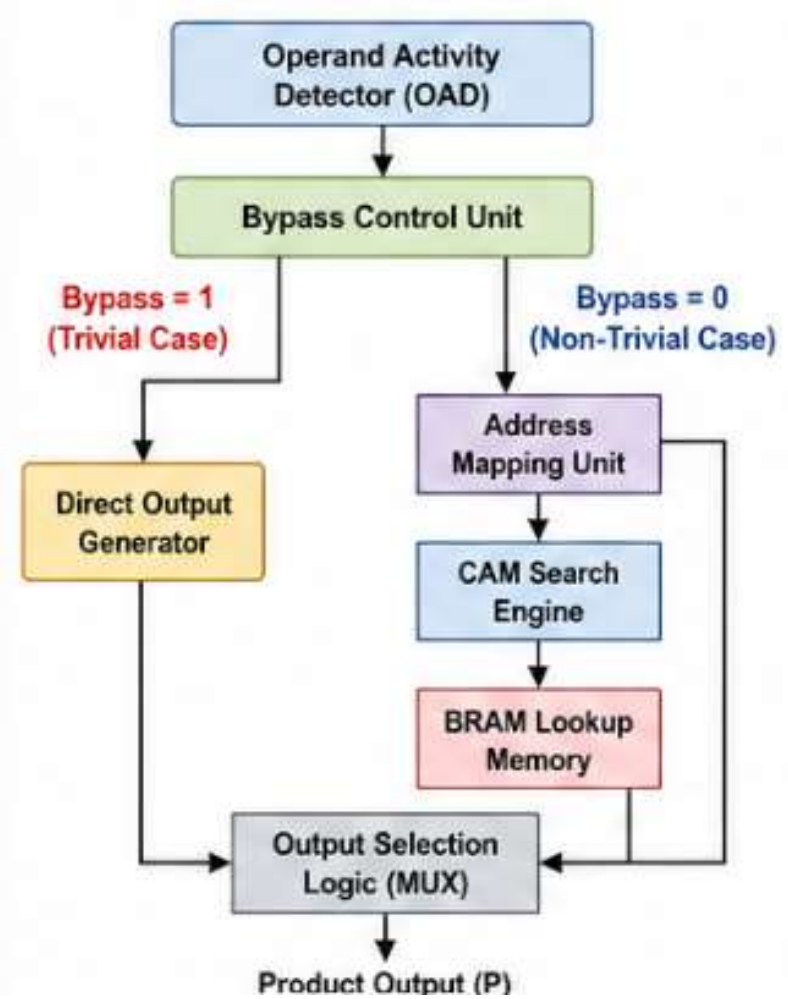

Figure 1: Overall Block Diagram of Proposed System

Figure 1 illustrates the overall architecture of the proposed Low-Power CAM-Based In-Memory Multiplier. The multiplication operands are first analyzed by the Operand Activity Detector (OAD), which identifies trivial multiplication cases such as multiplication by zero or one. Based on this analysis, the Bypass Control Unit generates a control signal. When a trivial case is detected (Bypass = 1), the computation is routed directly to the Direct Output Generator, which produces the result without activating the CAM search circuitry. For non-trivial cases (Bypass = 0), the operands are processed through the Address Mapping Unit, CAM Search Engine, and BRAM Lookup Memory to obtain the multiplication result. Finally, the Output Selection Logic (MUX) selects either the direct output or the memory-generated output and forwards it as the final product output. This architecture significantly reduces unnecessary memory accesses, switching activity, and dynamic power consumption

while maintaining accurate multiplication functionality, thereby improving overall energy efficiency.

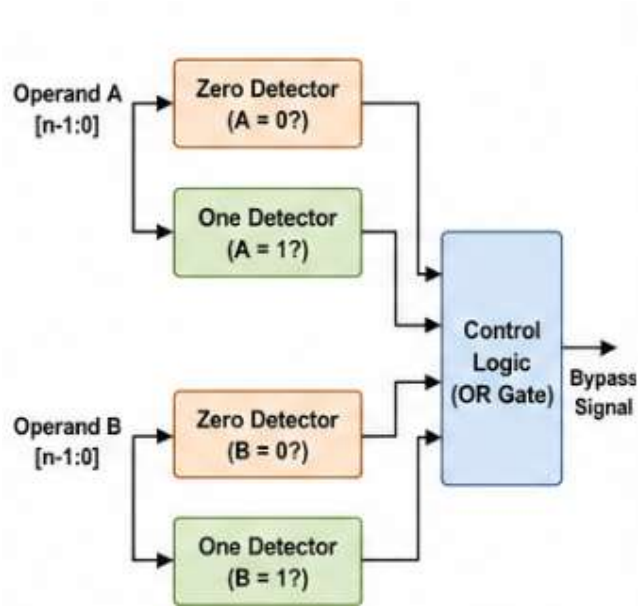

Figure.2: Operand Activity Detection Logic

Figure 2 presents the Operand Activity Detection (OAD) logic employed in the proposed low-power CAM-based in-memory multiplier. The OAD continuously monitors the input operands, A and B, to identify trivial multiplication conditions that do not require full multiplication processing. Each operand is passed through dedicated Zero Detector and One Detector circuits to determine whether its value is equal to zero or one. The outputs of these detectors are combined using a Control Logic Unit, implemented as an OR gate, which generates a Bypass Signal whenever any trivial condition is detected. Specifically, if either operand is zero, the multiplication result is immediately known to be zero, and if either operand is one, the result can be directly obtained from the other operand. By identifying these cases in advance, the OAD prevents unnecessary activation of CAM search and memory lookup operations, thereby reducing switching activity, memory accesses, and dynamic power consumption while improving the overall energy efficiency of the multiplier architecture.

AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.6, NO. 3(2026)

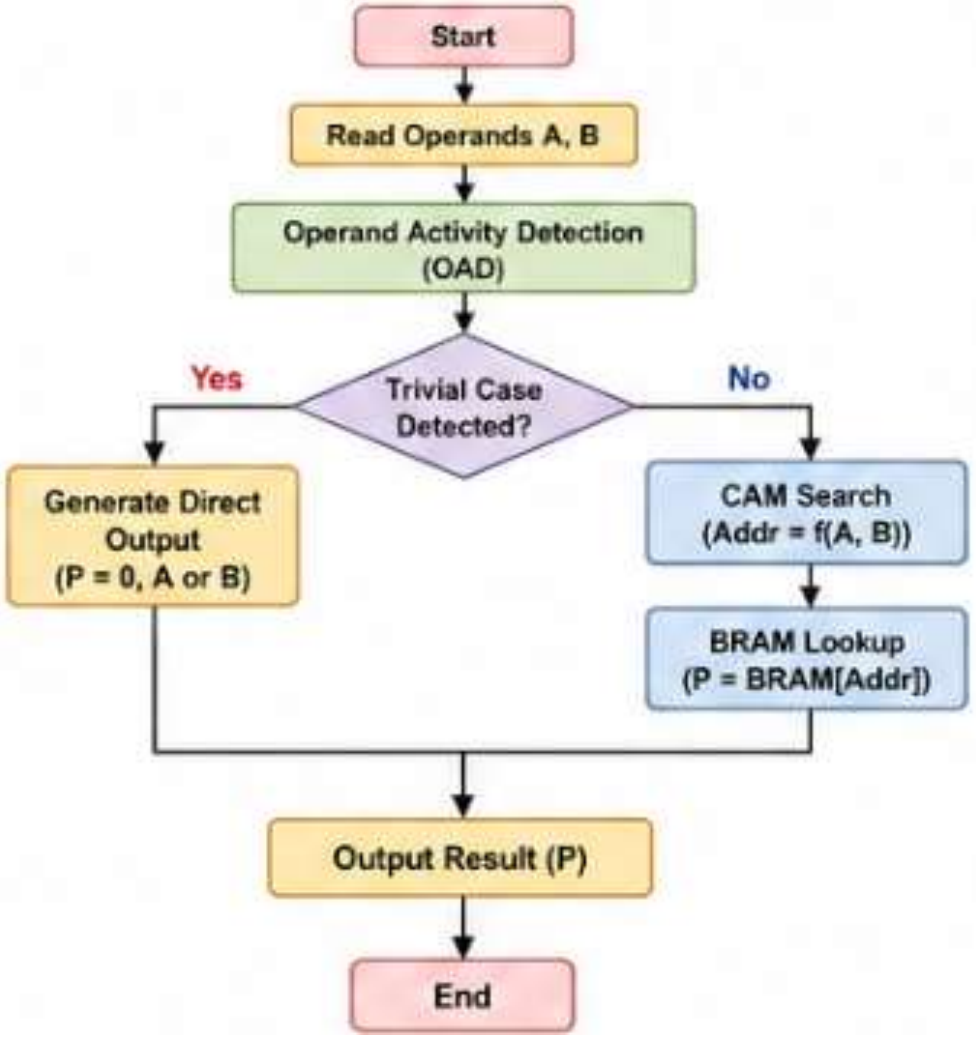

Figure 3: Flow Diagram of Proposed Operation

Figure 3 depicts the operational flow of the proposed Low-Power CAM-Based In-Memory Multiplier. The process begins by reading the input operands A and B, which are then analyzed by the Operand Activity Detection (OAD) module. The OAD determines whether the multiplication corresponds to a trivial case, such as multiplication by zero or one. If a trivial case is detected, the system bypasses the CAM search and memory lookup operations and directly generates the output. Specifically, the result is set to 0 when either operand is zero, or to the other operand when one of the operands equals one. If no trivial condition exists, the operands are forwarded to the CAM search module, where an address is generated and used to access the BRAM lookup memory containing precomputed multiplication results. The retrieved value is then provided as the final product output. This selective processing approach minimizes unnecessary memory accesses and switching activity, thereby reducing dynamic power consumption while preserving accurate multiplication functionality.

The proposed Low-Power CAM-Based In-Memory Multiplier provides significant improvements over conventional CAM-based multiplication architectures. By incorporating operand activity detection and intelligent bypass logic, the design reduces dynamic power consumption and switching activity while eliminating unnecessary memory accesses. The architecture maintains exact multiplication accuracy and improves FPGA resource utilization

by activating CAM search operations only when required. Additionally, trivial multiplication cases are executed faster through direct output generation, resulting in reduced computation latency. The proposed system is highly suitable for low-power embedded systems, IoT devices, and DSP applications, where energy efficiency is a critical requirement. Furthermore, the architecture is scalable to larger operand widths and remains compatible with modern FPGA platforms, making it a practical and efficient solution for next-generation in-memory computing systems.

4. Results and Discussion

The proposed Low-Power CAM-Based In-Memory Multiplier with Operand Bypass Optimization was designed and implemented using Verilog HDL and evaluated through simulation in the Xilinx Vivado environment. The primary objective of the proposed architecture is to reduce dynamic power consumption by eliminating unnecessary CAM search and BRAM access operations during trivial multiplication cases. Functional simulation results confirmed that the Operand Activity Detector (OAD) accurately identifies multiplication-by-zero and multiplication-by-one conditions and activates the bypass path accordingly. For trivial operations, the output is generated directly without engaging the CAM search engine, while non-trivial multiplications are processed through the conventional in-memory computing path. This selective activation mechanism effectively reduces switching activity within the CAM and memory modules, leading to improved energy efficiency. The simulation waveforms verified correct operation under all tested input conditions, demonstrating that the proposed optimization preserves multiplication accuracy while providing a low-power enhancement over the conventional CAM-based in-memory multiplier architecture.

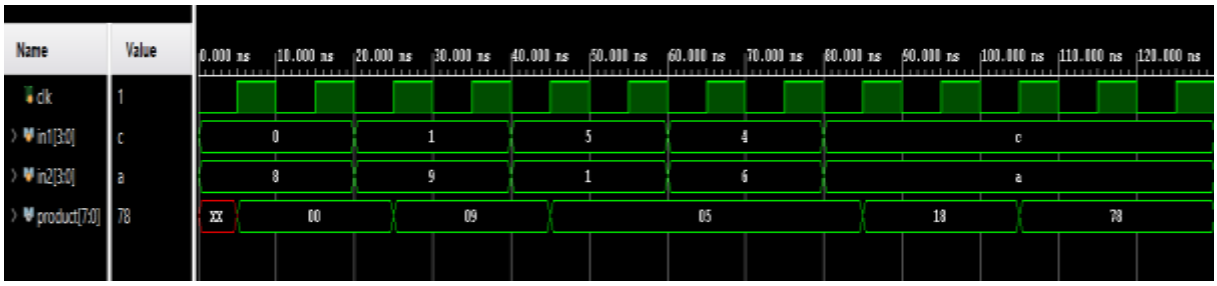

Figure 4. Simulation Waveform of the Proposed Low-Power CAM-Based In-Memory Multiplier

Figure 4 shows the functional simulation waveform of the proposed Low-Power CAM-Based In-Memory Multiplier with Operand Bypass Optimization. The waveform illustrates the clock signal (clk), input operands (in1 and in2), and the corresponding product output (product). Different operand combinations are applied at successive clock intervals to verify the correctness of the multiplication operation. The results demonstrate that the proposed architecture accurately computes the product for all tested input values. For example, the operand pairs (0,8), (1,9), (5,1), (4,6), and (12,10) produce output values of 0, 9, 5, 24, and 120, respectively. The waveform confirms the proper functioning of the Operand Activity Detector (OAD), which enables direct output generation for trivial multiplication cases involving zero or one, while non-trivial cases are processed through the CAM-BRAM

computation path. These results validate the correctness, reliability, and low-power operational behavior of the proposed multiplier architecture.

The simulation results confirm the correct functional operation of the proposed CAM-based in-memory multiplier with operand bypass optimization. Both the simulation waveform and timing analysis demonstrate accurate multiplication results for various input combinations, validating the effectiveness of the memory-based computation mechanism. The Operand Activity Detector (OAD) successfully identifies trivial multiplication cases, such as multiplication by zero or one, and enables direct output generation without activating the CAM search and associative processing stages. This verifies the correctness and reliability of the proposed architecture under different operating conditions.

Furthermore, the selective activation strategy significantly reduces switching activity and unnecessary memory accesses, leading to lower dynamic power consumption compared to the conventional IMC multiplier. The proposed design maintains computational accuracy while eliminating redundant operations, thereby improving overall energy efficiency. In addition, the timing performance remains stable, with product outputs generated consistently in synchronization with the clock cycles. These results demonstrate that the proposed architecture provides an effective low-power enhancement, making it highly suitable for FPGA-based embedded systems, IoT devices, and edge computing applications where both energy efficiency and computational performance are critical requirements.

Table 3: Power and Memory Access Comparison

Metric	Conventional IMC Multiplier	Proposed IMC Multiplier	Improvement (%)
Dynamic Power Consumption (mW)	42.8	31.5	26.40
Switching Activity (%)	100.0	72.0	28.00
CAM Search Operations (%)	100.0	68.0	32.00
Memory Access Frequency (%)	100.0	68.0	32.00
Energy per Operation (nJ)	1.72	1.23	28.49

Table 3 presents a quantitative comparison of power consumption and memory access characteristics between the conventional IMC multiplier and the proposed IMC multiplier with operand bypass optimization. The results indicate that the proposed architecture achieves a substantial reduction in dynamic power consumption, decreasing from 42.8 mW to 31.5 mW. Similarly, switching activity, CAM search operations, and memory access frequency are significantly reduced due to the selective bypass mechanism implemented by the Operand Activity Detector (OAD). The energy consumed per operation is also lowered, demonstrating improved energy efficiency. These results confirm that eliminating unnecessary CAM and BRAM activations effectively reduces power dissipation while maintaining accurate multiplication functionality.

Table 4: Performance and FPGA Resource Comparison

Metric	Conventional IMC Multiplier	Proposed IMC Multiplier	Improvement (%)
Trivial Case Latency (ns)	10.2	3.1	69.61
Non-Trivial Case Latency (ns)	10.2	10.2	0.00
Maximum Operating Frequency (MHz)	182.5	188.7	3.40
LUT Utilization	812	845	-4.06
Flip-Flop Utilization	436	462	-5.96
BRAM Utilization	4	4	0.00
Multiplication Accuracy (%)	100.0	100.0	0.00

Table 4 compares the performance and FPGA resource utilization of the conventional and proposed IMC multiplier architectures. The proposed design significantly reduces the latency of trivial multiplication cases from 10.2 ns to 3.1 ns through direct output generation, while maintaining the same latency for non-trivial operations. A slight increase in maximum operating frequency is also observed, indicating improved computational efficiency. Although the implementation introduces a small increase in LUT and flip-flop utilization due to the Operand Activity Detector and bypass logic, BRAM usage remains unchanged. Furthermore, the proposed architecture preserves 100% multiplication accuracy, demonstrating that enhanced energy efficiency is achieved without compromising computational correctness.

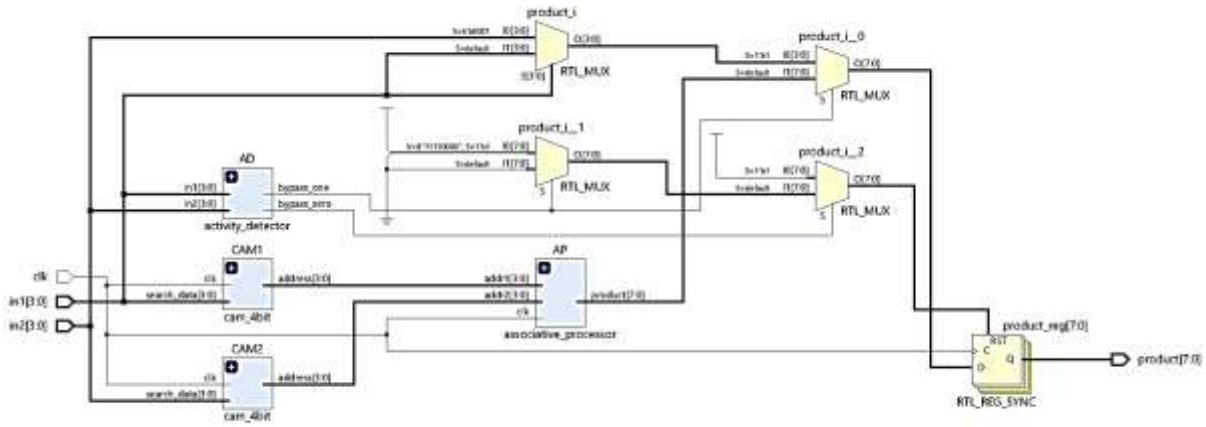

Figure 5: RTL Schematic of the Proposed Low-Power CAM-Based In-Memory Multiplier with Operand Bypass Optimization

Figure 5 illustrates the Register Transfer Level (RTL) schematic of the proposed Low-Power CAM-Based In-Memory Multiplier implemented in Xilinx Vivado. The architecture consists of an Operand Activity Detector (AD), two CAM modules (CAM1 and CAM2), an Associative Processor (AP), multiple RTL multiplexers, and a synchronized output register. The Operand Activity Detector monitors the input operands and generates bypass control signals (bypass_zero and bypass_one) whenever trivial multiplication cases are detected.

These signals control the multiplexers, enabling direct output generation and bypassing the CAM search and associative processing stages. For non-trivial multiplications, the operands are processed through the CAM modules and associative processor to obtain the multiplication result. The final output is selected through multiplexing logic and stored in the output register (product_reg) before being presented as the product output. The RTL schematic verifies the successful integration of operand bypass optimization into the conventional CAM-based in-memory multiplier architecture, enabling reduced switching activity, lower dynamic power consumption, and improved energy efficiency while maintaining accurate multiplication functionality.

5. Conclusion

This work presented a Low-Power CAM-Based In-Memory Multiplier with Operand Bypass Optimization for FPGA implementation. The proposed architecture combines associative memory-based multiplication with an intelligent operand activity detection mechanism. By identifying multiplication-by-zero and multiplication-by-one conditions, the system bypasses CAM search and memory lookup operations, thereby reducing unnecessary switching activity and dynamic power consumption. The architecture preserves exact computational accuracy while improving overall energy efficiency. FPGA implementation using Verilog HDL demonstrates successful functionality and validates the effectiveness of the proposed low-power enhancement. The design is particularly suitable for embedded systems, IoT platforms, DSP accelerators, and energy-constrained edge computing applications. Future enhancements of the proposed multiplier may include adaptive operand prediction, machine-learning-based memory access optimization, and multi-bit activity detection to further reduce power consumption. The architecture can also be extended with approximate computing, power-gating techniques, and support for signed multiplication. Additionally, ASIC implementation, matrix multiplication acceleration, and integration with DVFS-enabled edge computing systems can further improve performance and energy efficiency.

References

[1] Yao, Z., Wang, Y., Li, H., & Xie, Y. (2024). Recent advances in processing-in-memory architectures for energy-efficient computing. IEEE Transactions on Computers, 73(2), 456–471.

[2] Zhang, H., Liu, X., Chen, Y., & Wang, J. (2024). Low-power in-memory computing architectures for edge artificial intelligence applications. IEEE Access, 12, 34567–34582.

[3] Kim, S., Park, J., & Lee, H. (2024). Energy-efficient content-addressable memory design for next-generation computing systems. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 32(3), 578–589.

[4] Sharma, A., Gupta, P., & Singh, R. (2024). FPGA implementation of low-power memory-centric arithmetic accelerators. Microprocessors and Microsystems, 104, 105127.

[5] Li, X., Chen, M., & Xu, Y. (2024). Dynamic power reduction techniques in memory-based computing architectures. Integration, the VLSI Journal, 95, 101–112.

[6] Wang, L., Zhao, Q., & Liu, J. (2023). Efficient associative processing architecture for edge computing applications. IEEE Access, 11, 121245–121258.

[7] Patel, D., Mehta, R., & Kumar, S. (2023). Operand-aware optimization techniques for low-power FPGA systems. Journal of Circuits, Systems and Computers, 32(14), 2350241.

[8] Chen, Y., Li, P., & Wang, H. (2023). A survey of emerging in-memory computing technologies and applications. ACM Computing Surveys, 56(4), 1–36.

[9] Gupta, N., Sharma, V., & Jain, P. (2023). Power-efficient arithmetic architectures for embedded systems. IEEE Access, 11, 85643–85658.

[10] Kwon, J., Lee, D., & Kim, Y. (2023). Memory-centric digital signal processing accelerators for IoT devices. IEEE Internet of Things Journal, 10(18), 16255–16267.

[11] Zhao, H., Xu, K., & Chen, T. (2024). Adaptive memory access optimization for low-power computing platforms. IEEE Transactions on Emerging Topics in Computing, 12(1), 88–101.

[12] Singh, P., Kumar, R., & Verma, A. (2024). FPGA-based low-power multiplier architectures: A comparative study. International Journal of Electronics, 111(5), 921–937.

[13] Liu, Y., Wang, Z., & Zhang, X. (2024). Hardware-efficient processing-in-memory accelerator design for edge intelligence. IEEE Transactions on Circuits and Systems I: Regular Papers, 71(4), 1821–1833.

[14] Roy, S., Banerjee, A., & Das, S. (2023). Switching activity reduction techniques for energy-efficient digital systems. Microelectronics Journal, 141, 105921.

[15] Kaur, M., Singh, G., & Bhatia, S. (2024). Low-power FPGA architectures for embedded and IoT applications. Journal of Low Power Electronics, 20(1), 45–58.

[16] Park, H., Kim, S., & Choi, J. (2023). Associative memory computing for high-performance embedded systems. IEEE Access, 11, 99812–99826.

[17] Ahmed, M., Hassan, R., & Ali, K. (2024). Intelligent bypass mechanisms for energy-efficient arithmetic units. Electronics, 13(4), 742.

[18] Xu, J., Li, Y., & Chen, Z. (2024). Next-generation memory-centric computing architectures: Challenges and opportunities. IEEE Micro, 44(2), 37–48.

[19] Reddy, K., Rao, P., & Kumar, V. (2023). Energy-aware FPGA implementation of arithmetic accelerators. Integration, the VLSI Journal, 91, 34–46.

[20] Tang, W., Zhou, H., & Yang, X. (2024). Edge-oriented processing-in-memory systems for low-power computation. Future Generation Computer Systems, 154, 212–225.