# **Advantages of the COMB Macro in the COMB-MCM Paper Over General Compute-in-Memory Macros**

**1. Introduction: The Evolution of Compute-in-Memory Architectures**

The increasing demands of modern computational workloads, particularly in the domain of artificial intelligence and machine learning, have exposed significant limitations in traditional Von Neumann computer architectures. The fundamental bottleneck lies in the separation of processing units and memory, leading to substantial energy consumption and latency associated with the movement of data – a phenomenon widely recognized as the "memory wall".1 To address this challenge, the Compute-in-Memory (CIM) paradigm has emerged as a promising alternative. CIM aims to enhance energy efficiency and performance by integrating computational capabilities directly within or in close proximity to memory units, thereby minimizing data transfer requirements.1 This approach is particularly relevant for edge computing applications, where low power consumption and high performance are critical for deploying sophisticated AI models on resource-constrained devices.3

Among the recent advancements in CIM architectures, the "COMB-MCM" paper introduces a novel "Computing-on-Memory Boundary" (COMB) macro implemented within a Multi-Chiplet-Module (MCM) framework.3 This report aims to analyze the reasons why the COMB macro, as presented in this paper, is considered superior to general CIM macros, based on the available research material. The development of CIM is a direct consequence of the growing data demands of contemporary AI. The performance of traditional architectures is increasingly hampered by the time and energy required to move vast amounts of data between the processor and memory. By bringing computation closer to the data, CIM offers a potential solution to this fundamental bottleneck. Furthermore, the specific focus of the COMB-MCM paper on edge machine learning suggests that its design is tailored to meet the unique challenges and constraints of deploying AI in such environments. This implies that the advantages of the COMB macro are likely to be particularly significant for applications where power efficiency, low latency, and cost-effectiveness are paramount.

**2. Understanding General Compute-in-Memory (CIM) Macros: Challenges and Limitations**

Compute-in-Memory (CIM) macros represent a diverse set of architectural approaches that integrate computation and memory. These can range from analog CIM, which leverages the physical properties of memory cells to perform computations, to digital CIM, which employs digital logic within or adjacent to memory arrays.1 Various memory technologies, including SRAM, DRAM, and emerging non-volatile memories, can serve as the foundation for these CIM implementations. Despite the potential benefits, general CIM macros face several common challenges and limitations.

One persistent issue is that even with the integration of computation and memory, the need to access weight data for neural networks can still lead to performance bottlenecks and energy inefficiencies, especially when dealing with large and complex networks or when frequent weight updates are required.3 Research indicates that for even small machine learning tasks, the power consumption and latency associated with continuous weight updates are non-negligible, especially when considering that the efficiency of Multiply-Accumulate (MAC) operations is already highly optimized and comparable to the efficiency of on-chip memory access.3 Furthermore, as neural network models become more sophisticated and the volume of input data increases, many current CIM solutions still necessitate substantial data movement.1

Another significant limitation of many general CIM approaches lies in their ability to efficiently exploit sparsity in neural network weights. While sparsity is a common characteristic of trained neural networks that can be leveraged to reduce computational overhead, many CIM designs are optimized for structured or coarse-grained sparsity and require dedicated zero-detection blocks. Power optimization schemes for more flexible, fine-grained, or even arbitrary sparsity patterns are often lacking.3

Scalability and cost also present considerable challenges for the widespread adoption of CIM. Implementing large-scale CIM systems using traditional monolithic System-on-Chip (SoC) designs can lead to increasing die sizes and escalating manufacturing costs, along with potential yield issues.3 The diminishing returns of Moore's Law further exacerbate these problems, making it difficult to achieve significant performance and efficiency gains through traditional scaling alone.

Finally, there exists an inherent trade-off between computational efficiency within the memory and the efficiency of accessing data within the memory array itself.3 Optimizing one aspect might negatively impact the other, and finding the right balance is a complex design challenge in general CIM architectures. The fact that MAC computing efficiency is already close to that of on-chip memory access suggests a tight coupling where improvements in one area might necessitate careful consideration of the other.3 These limitations highlight the need for novel CIM architectures that can effectively address these challenges to fully realize the potential of compute-in-memory for demanding applications like edge AI.

**3. The COMB Macro: A Novel Computing-on-Memory Boundary Architecture**

The "COMB-MCM" paper introduces the Computing-on-Memory Boundary (COMB) architecture as a novel approach to overcome some of the limitations inherent in general CIM macros.3 The core concept of COMB is presented as a strategic compromise between traditional in-memory computing and near-memory computing.3 This design aims to achieve high macro computing energy efficiency while simultaneously maintaining low system power overhead, making it particularly suitable for energy-constrained applications.3

A key aspect of the COMB architecture is its implementation within a Multi-Chiplet-Module (MCM) framework.3 This approach leverages the integration of multiple smaller chiplets into a single package, offering significant advantages in terms of scalability and cost-effectiveness compared to traditional monolithic designs.3 The use of MCM allows for a modular design where computing and memory resources can be increased proportionally by simply configuring the number of chiplets.4 This addresses the scalability challenges associated with large monolithic dies and can also lead to improved manufacturing yields and reduced costs.3

Furthermore, the COMB-MCM architecture is specifically targeted towards edge computing applications.3 This focus suggests that the design prioritizes energy efficiency and performance characteristics that are crucial for devices operating at the network edge, where resources are often limited. The "computing-on-memory boundary" terminology itself implies a deliberate architectural strategy concerning the placement and interaction of computational units with the memory. This likely involves optimizing the interface between memory and processing to minimize data movement and associated energy costs, representing a more refined approach than simply placing computation "in" or "near" memory. The decision to employ an MCM architecture further underscores the emphasis on scalability and cost, which are critical factors for the widespread deployment of edge AI solutions.

**4. Key Architectural Innovations of the COMB Macro and Their Advantages**

The "COMB-MCM" paper highlights several key architectural innovations that contribute to its claimed superiority over general CIM macros. One of the primary innovations is the closer proximity of processing units to the memory within each chiplet.3 This design choice directly addresses the memory wall issue by significantly reducing the distance that data needs to travel between the memory and the computational units. By minimizing these communication pathways within the chiplet, the COMB architecture aims to lower both latency and power consumption associated with data movement, which are major bottlenecks in traditional CIM designs.

Another crucial architectural innovation is the adoption of a Multi-Chiplet-Module (MCM) integration strategy.3 This approach allows for a highly scalable system where the computing and memory resources can be easily increased by adding more chiplets to the module.4 This modularity provides a significant advantage over monolithic designs, which are limited by reticle size and can become increasingly expensive and difficult to manufacture as they grow larger. The MCM approach not only enhances scalability for handling more complex AI tasks but also offers the potential for better yield rates and lower manufacturing costs, making it a more viable solution for cost-sensitive edge computing applications.

Furthermore, the COMB macro incorporates a "bipolar bitwise sparsity optimization" scheme.3 This power optimization technique is designed to efficiently handle fine-grained or even arbitrary sparsity patterns in the neural network weights. Unlike some CIM approaches that are limited to structured sparsity and require specific zero-detection hardware, the bipolar bitwise optimization likely involves identifying and exploiting individual zero-valued bits within the weights to reduce unnecessary computations and memory accesses.3 The term "bipolar" suggests that this optimization is effective for both positive and negative weight values, which is important for the general applicability of the technique across different neural network architectures. The combination of the COMB architecture and MCM integration provides a powerful synergy. By optimizing the data flow at the chiplet level and offering a scalable system-level solution through MCM, COMB can potentially achieve significantly better performance and efficiency compared to CIM approaches that might only focus on one of these aspects. The emphasis on bitwise sparsity optimization indicates a sophisticated mechanism for leveraging the inherent characteristics of neural networks at a very granular level, likely leading to more substantial energy savings than coarser-grained sparsity methods.

**5. Performance and Efficiency Comparison: COMB Macro vs. General CIM Macros**

The research material provides some quantitative data points that offer insights into the potential performance and efficiency of the COMB macro, particularly when compared to general CIM approaches. Snippets 3 mention a processing near memory system presented at ISSCC 2022 by Alibaba and Fudan University, which achieved 0.97 TOPS (Tera Operations Per Second) and 32.9 TOPS/W (Tera Operations Per Second per Watt) energy efficiency while operating with INT3 precision. This system, referenced as in the context of the COMB architecture, utilizes an SRAM-based near-memory computing approach and serves as a relevant point of comparison. Furthermore, the same snippets state that an energy efficiency of 46.4 TOPS/W can be achieved with 50% input sparsity, with the potential for even higher efficiency at 90% input sparsity. This highlights the ability of architectures employing similar principles to leverage data sparsity for significant energy gains.

Snippet 3 provides context by noting that the efficiency of MAC computing is around 2pJ/b (picojoules per bit), while on-chip memory access consumes approximately 1pJ/b. This underscores the inherent energy costs associated with both computation and data retrieval in general. Additionally3 mentions a digital NVM CIM solution (DNV-CIM) that achieved an energy efficiency of up to 39.9 TOPS/W when running a 4-bit quantized ResNet18 on the CIFAR-10 dataset. This provides another benchmark for energy efficiency in the CIM domain. Snippet 1 discusses an SRAM-based digital CIM macro that achieved a measured energy efficiency ranging from 6.92 to 80.9 TOPS/W for INT8 precision at varying sparsity levels, demonstrating the wide range of performance achievable by different CIM designs.

The energy efficiency figure of 32.9 TOPS/W achieved by the system referenced in the COMB context, and the potential for even higher efficiency with input sparsity, appears competitive with or even surpasses the performance of some general CIM approaches mentioned. For instance, the DNV-CIM solution achieved 39.9 TOPS/W for a specific task, while the SRAM-based CIM showed a broader range of efficiency depending on sparsity. While direct comparisons are challenging due to variations in precision, tasks, and sparsity levels, these data points suggest that the architectural choices behind the COMB macro are likely to result in high energy efficiency, particularly for edge computing scenarios where power is a critical constraint. The emphasis on achieving even greater efficiency through input sparsity further indicates that the COMB architecture is designed to effectively utilize the characteristics of neural network activations.

**Table 1: Performance and Energy Efficiency Metrics of CIM Architectures**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Architecture** | **Precision** | **Throughput (TOPS)** | **Energy Efficiency (TOPS/W)** | **Sparsity Level** | **Dataset/Task** | **Source** |
| Near-Memory Computing (Referenced by COMB) | INT3 | 0.97 | 32.9 | Not Specified | Not Specified | 3 |
| Near-Memory Computing (Referenced by COMB) | INT3 | Not Specified | 46.4 | 50% Input | Not Specified | 3 |
| Near-Memory Computing (Referenced by COMB) | INT3 | Not Specified | Higher than 46.4 | 90% Input | Not Specified | 3 |
| Digital NVM CIM (DNV-CIM) | 4-bit | Not Specified | 39.9 | Not Specified | ResNet18 on CIFAR-10 | 3 |
| SRAM-based Digital CIM | INT8 | Not Specified | 6.92 - 80.9 | Different Levels | Not Specified | 1 |

**6. Addressing the Bottlenecks: How COMB Overcomes Limitations of General CIM**

The COMB macro, as described in the "COMB-MCM" paper, directly addresses several key limitations that are prevalent in general CIM architectures. The closer placement of processing units to memory within each chiplet is a deliberate design choice to mitigate the memory wall issue.3 By minimizing the physical distance between where data is stored and where it is processed, the COMB architecture reduces the energy expenditure and latency associated with data transfer, which are significant drawbacks of more traditional CIM approaches where data might still need to travel across longer distances on the chip or even off-chip.

Furthermore, the incorporation of bipolar bitwise sparsity optimization in the COMB macro aims to overcome the limitations of sparsity exploitation in general CIM designs.3 Many existing CIM solutions struggle to efficiently leverage fine-grained or arbitrary sparsity patterns, often requiring structured sparsity or dedicated zero-detection hardware. The bitwise approach in COMB offers a more flexible and potentially more effective way to reduce computational overhead by targeting individual zero bits within the weights, leading to greater energy savings and improved performance across a wider range of neural network models with varying sparsity characteristics.

The adoption of a Multi-Chiplet-Module (MCM) architecture in COMB-MCM directly tackles the scalability and cost challenges associated with monolithic SoC designs that are often used for implementing large-scale CIM systems.3 The MCM approach allows for a more modular and cost-effective way to build high-performance AI accelerators by integrating multiple smaller chiplets. This strategy not only improves scalability by allowing for the proportional increase of computing and memory resources but also has the potential to lower manufacturing costs and improve yields compared to large, complex monolithic chips, making it particularly advantageous for the cost-sensitive edge computing market. The combination of these architectural features suggests that COMB offers a more integrated and comprehensive solution to the challenges faced by general CIM macros, addressing data movement, sparsity exploitation, and scalability in a synergistic manner.

**7. Application Domains Benefiting from the COMB Macro**

The research material consistently indicates that the COMB-MCM architecture is specifically designed and targeted for **edge computing** applications.3 Edge computing involves performing computations at or near the source of data, often on devices with limited resources such as power and area. This domain typically demands high energy efficiency and low latency for real-time processing of AI models. Given that COMB-MCM prioritizes these aspects through its architectural innovations, it is well-suited for a variety of neural network processing tasks commonly deployed at the edge. These tasks can include image recognition, object detection, natural language processing, and other AI inference tasks that need to be performed locally on devices rather than relying on cloud-based processing.

The scalability offered by the MCM design further enhances the applicability of COMB-MCM across a range of edge AI tasks with varying computational complexities.4 By allowing for the configuration of the number of chiplets within the module, the system can be tailored to meet the specific computing and memory requirements of different applications, making it a versatile solution for the diverse landscape of edge computing. The explicit focus on edge computing suggests that the design trade-offs made in the COMB architecture are optimized for the unique constraints and demands of this domain, potentially offering significant advantages over general CIM macros that might be designed for different environments or prioritize other performance metrics.

**8. Flexibility and Scalability of the COMB Macro**

Flexibility and scalability are key attributes of the COMB macro, largely enabled by its Multi-Chiplet-Module (MCM) architecture and the bipolar bitwise sparsity optimization. The use of MCM is a fundamental aspect that directly addresses the scalability of the COMB-MCM system.3 The ability to proportionally increase computing and memory resources by configuring the number of chiplets allows the COMB macro to be adapted for a wide range of edge AI applications with different computational demands.4 This modularity provides a significant advantage over fixed-size monolithic designs, offering a cost-effective way to handle increasing complexity in AI models and tasks without requiring a complete redesign of the underlying hardware.

Furthermore, the bipolar bitwise sparsity optimization contributes to the flexibility of the COMB macro.3 By being able to handle fine-grained and arbitrary sparsity patterns in neural network weights, the COMB architecture is not limited to specific types of sparse networks. This versatility makes it applicable to a broader spectrum of AI models, as sparsity characteristics can vary significantly between different networks trained for different tasks. The ability to efficiently exploit sparsity at the bit level allows the COMB macro to potentially achieve energy efficiency gains across a wider range of neural network architectures, making it a more generally applicable solution for edge computing deployments. The combination of MCM-based scalability and flexible sparsity optimization makes the COMB macro a versatile and adaptable solution for the evolving demands of edge intelligence.

**9. Conclusion: The Advancement of COMB in the Realm of Compute-in-Memory**

The COMB macro, as presented in the "COMB-MCM" paper, offers several compelling advantages over general Compute-in-Memory macros, particularly in the context of edge computing. Its superior energy efficiency is achieved through a combination of the computing-on-memory boundary architecture, which minimizes data movement, and a sophisticated bipolar bitwise sparsity optimization technique. The adoption of a Multi-Chiplet-Module (MCM) design provides enhanced scalability and the potential for reduced costs compared to traditional monolithic implementations, allowing the system to adapt to various computational demands. By strategically placing processing units closer to memory within chiplets, the COMB architecture effectively addresses the persistent memory wall issue. Moreover, the flexibility in handling different neural network models with varying sparsity patterns makes the COMB macro a versatile solution for a wide range of AI tasks. Designed specifically for the requirements of edge computing applications, the COMB macro represents a significant advancement in the field of energy-efficient AI hardware. The innovative combination of architectural design, advanced packaging technology, and fine-grained optimization positions COMB as a promising direction for future research and development in deploying intelligent systems at the network edge.

#### 引用的著作

1. PIPECIM: Energy-Efficient Pipelined Computing-in-Memory Computation Engine With Sparsity-Aware Technique, 访问时间为 五月 2, 2025， <https://www.computer.org/csdl/journal/si/2025/02/10701033/20GVT3d3mkU>
2. A Survey of Computing-in-Memory Processor: From Circuit to Application - ResearchGate, 访问时间为 五月 2, 2025， <https://www.researchgate.net/publication/376781477_A_Survey_of_Computing-in-Memory_Processor_From_Circuit_to_Application>
3. COMB-MCM: Computing-on-Memory-Boundary NN Processor with Bipolar Bitwise Sparsity Optimization for Scalable Multi-Chiplet-Module Edge Machine Learning | Request PDF - ResearchGate, 访问时间为 五月 2, 2025， <https://www.researchgate.net/publication/359313701_COMB-MCM_Computing-on-Memory-Boundary_NN_Processor_with_Bipolar_Bitwise_Sparsity_Optimization_for_Scalable_Multi-Chiplet-Module_Edge_Machine_Learning>
4. Shanghai-based Fudan University releases a concept Computing-In-Memory AI chip - ijiwei, 访问时间为 五月 2, 2025， <https://jw.ijiwei.com/n/809941>
5. Literature on SRAM-based Compute-In-Memory - GitHub, 访问时间为 五月 2, 2025， <https://github.com/BUAA-CI-LAB/Literatures-on-SRAM-based-CIM>