

MONET: A Mixture-of-Experts Accelerator with a Multicast-Optimized Two-Tier Network-on-Chip

Siqin Liu, Maya Roediger and Avinash Karanth
School of Electrical Engineering and Computer Science
Ohio University
 Athens, OH, USA
 E-mail: ls847719, mr030021, karanth@ohio.edu

Abstract—The growing complexity of Mixture-of-Experts (MoE) models in machine learning applications demands innovative hardware solutions to address their unique computational and data movement challenges. Some of the critical challenges facing MoE models include sparse activation, dynamic token routing and irregular computation patterns that lead to low utilization and higher communication latency. In this paper, we introduce MONET, a novel two-tier Network-on-Chip (NoC) architecture designed to efficiently execute MoE workloads by co-optimizing compute, memory, and interconnect subsystems. The first tier consists of a reconfigurable systolic processing element (PE) island, executing both gating and expert computations, with runtime-configurable support for sparse/dense operations, expert reordering, and activation functions. The second tier incorporates a dual mesh network connecting a grid of PE islands; one network manages input token delivery with a broadcast scheme optimized for the gating phase of MoE, while the other is tailored for efficient inter-expert communication necessary for result aggregation. Evaluated on MoE benchmarks, MONET demonstrates up to $8.5\times$ lower latency and over $6\times$ better energy efficiency compared to state-of-the-art MoE accelerators.

I. INTRODUCTION

The explosive growth of foundation models, such as large-scale vision transformers and language models, has underscored the need for computational efficiency at scale. Among the most promising strategies to reduce inference and training costs is the Mixture-of-Experts (MoE) framework [1]–[4], which conditionally activates a subset of specialized sub-networks (experts) for each input. By exploiting sparsity and task-adaptivity, MoE models drastically reduces computation while preserving, or even improving, performance in multitask learning (MTL), multilingual translation, and multi-modal reasoning [5]–[7].

Recent deployments of MoE-based models such as Mixtral-8x7B [5], DeepSeek-V2 [8], Arctic [9], and Grok-1 [10] demonstrate the scalability and practical viability of sparse expert routing in real-world systems. However, these gains are severely limited by the hardware inefficiencies when mapped to traditional accelerators such as GPUs or even generalized systolic arrays. Specifically, the combination of sparse activation, dynamic token routing, and irregular computation pattern leads to low utilization, memory thrashing, and high communication latency [7], [11].

At the core of these challenges lies the complex interplay between the *gating mechanism* which dynamically selects the top- k experts for a given input and the *expert computation*, which executes high-dimensional matrix multiplications across

sparse, irregular expert subsets. While the MoE paradigm enables significant computational savings at the algorithmic level, MoE models imposes communication patterns that traditional Network-on-Chip (NoC) designs are ill-suited to handle. In the gating phase, broadcast and multicast operations are essential to distribute input tokens and gating weights to all experts or gating units. Conventional XY-mesh or ring-based NoCs typically serialize this distribution, introducing additional latency and cause contention. During the expert phase, selected tokens and expert parameters must be routed in a fine-grained, dynamically changing pattern, exacerbating congestion and leading to inefficient routing and under-utilization of compute units.

These limitations have been partially addressed in recent MoE acceleration frameworks such as FastMoE, Tutel, and OpenMoE [6], [12], which focus on optimizing runtime scheduling and token-to-expert mapping at the software level. However, software solutions are typically deployed on conventional NoC fabrics (generic CPU/GPU interconnects or mesh topologies) without architectural support for MoE-specific requirements such as sparse expert activations, adaptive routing, or dynamic token multicast. Existing NoC designs often lack efficient multicast mechanisms tailored for high fan-out communication (across PE array) or capable of aggregating results from multiple experts. This leads to injection bottlenecks, idle compute islands, and substantial backpressure during expert fusion phases. Critically, they also fail to exploit reuse opportunities in gating weights or shared tokens, which is a key knob for bandwidth reduction in MoE execution. Thus, a dedicated NoC architecture that natively supports *multicast-aware token delivery*, *expert-parallel routing*, and *low-latency aggregation* are imperative for enhancing the efficiency of MoE accelerators.

In this paper, we present Mixture-of-Experts Accelerator with Optimized Network-on-Chip (**MONET**), a hardware accelerator architecture specifically designed to meet the dynamic sparsity, irregular communication, and compute heterogeneity of modern MoE workloads. MONET introduces a **two-tier NoC** system that partitions (a) token multicast, (b) expert weight routing, and (c) output aggregation into dedicated communication planes. The first tier consists of a 2D grid of processing elements (PEs) using systolic-style arrays to maximize weight reuse and throughput for expert computation. The second tier implements dual functional mesh planes: one

dedicated to token and gating weight multicast, and the other for expert output aggregation. To support this dataflow, we introduce two specialized router designs—*Mel*, a multicast-enabled link-reversal router, and *Bel*, a bypass-optimized low-latency router, each customized to different phases of MoE execution. The architecture integrates double-buffered memory, expert reordering logic, and token reuse mechanisms, enabling high throughput even under top- k sparsity and dynamic token-to-expert mappings. We evaluate MONET in representative MoE models, including M3ViT [7], DeepSpeed-MoE [2], and Switch-base-8 [13], demonstrating significant performance and energy efficiency gains over three baselines: a simulated 2D systolic array, EdgeMoE [14] on FPGA, and Space-Mate [15], an ASIC-based sparse MoE processor. Using RTL synthesis, NoC-level simulation, and real-world MoE workloads, MONET achieves up to $8.5\times$ latency reduction and more than $6\times$ energy savings over existing baselines.

II. BACKGROUND AND MOTIVATION

Mixture of Experts: MoE models are a scalable architecture that conditionally routes input data through a small, specialized subset of multiple parallel expert networks. Originally introduced by Jacobs et al. [16] and later extended with hierarchical gating [17], MoE architectures consist of a set of expert functions $\{E_i\}_{i=1}^N$ and a gating network $G(x) \in \mathbb{R}^N$ that assigns importance weights to each expert based on input x . The output y of the MoE model is a weighted combination of expert outputs:

$$y = \sum_{i \in \mathcal{S}(x)} \frac{\exp(G_i(x))}{\sum_{j \in \mathcal{S}(x)} \exp(G_j(x))} \cdot E_i(x) \quad (1)$$

Here, $\mathcal{S}(x) \subseteq \{1, \dots, N\}$ denotes the top- k experts selected by the gating function. This sparse activation mechanism ensures that only a small number of experts are active per input, significantly reducing computational cost while preserving model expressiveness. MoE models have been integrated into large-scale systems such as GShard [2], Switch Transformer [13], and Mixtral [5], demonstrating improved efficiency and scalability across NLP and vision tasks.

Challenges in Accelerating MoE Models: While MoE models offer substantial gains in compute efficiency by activating only a sparse subset of expert networks per input, they also introduce significant challenges at the system level. First, the gating networks in MoE models vary substantially in size and structure—for example, 256×8 in Sparsely-Gated and 384×64 in M3ViT—requiring additional memory, logic, and runtime resources for expert selection. Since gating decisions are based on dynamic top- k token-to-expert mappings, this process generates irregular and fine-grained communication patterns, placing high demands on NoC routing flexibility and broadcast capabilities. Second, expert configurations across models are highly heterogeneous. Some employ deep-narrow layers (e.g., 3072×128), while others use large, square-shaped matrices (e.g., 384×384), which complicates static compute scheduling and results in uneven workload distribution across processing units. Third, MoE workloads impose considerable pressure on

memory bandwidth. Despite sparse activation, the movement of input tokens and expert weights results in high-volume memory access, reaching 10^{10} bytes per batch in DeepSpeed-MoE and Switch-base-8. Without bandwidth-aware dataflows or on-chip reuse strategies, these volumes quickly become a performance bottleneck. Finally, computation sparsity poses a dual-edged challenge. While models such as DeepSpeed-MoE and Switch-base-8 utilize different expert counts, ($k = 1$) and ($k = 4$) respectively, they still demand comparable numbers of MACs, underscoring the importance of efficient token routing and expert utilization.

Network-on-Chips: NoCs is a scalable interconnect architecture that replaces shared buses with structured communication networks composed of routers and links [18], [19]. Due to their criticality to communication, several multicore architectures in the past have focused on multicast and broadcast capabilities for coherence traffic [20], [21]. As AI workloads grow in scale and irregularity, NoC designs face mounting pressure to support high-throughput, low-latency communication. Architectures like Eyeriss [22] and MAERI [23] focus on dense CNN-style compute, offering spatial reuse and reconfigurable routing, but are not optimized for sparse activation patterns. More recent systems, including EIE [24], Ruche networks [25] and SCALE-Sim [26], address aspects of memory bandwidth and contention-aware scheduling, yet they still assume fixed dataflow or lack multicast capabilities. Our work builds on these limitations by proposing a two-tier NoC specifically optimized for MoE workloads, with router-level multicast and aggregation support to accommodate top- k expert selection and sparse computation.

Collectively, these observations underscore a fundamental mismatch between MoE execution patterns and conventional accelerator architectures, particularly in the interconnect subsystem. This motivates the need for a custom hardware design that incorporates multicast-aware routing, token reuse, expert-adaptive scheduling, and load-balanced data movement at the Network-on-Chip level.

III. PROPOSED ARCHITECTURE: MONET

The proposed accelerator, MONET seamlessly integrates into a transformer-based pipeline that includes word embedding, self-attention, and MoE blocks as shown in Figure 1. The MoE computation is offloaded to the proposed hardware, where tokens and weights are transmitted via a two-tier Network-on-Chip (NoC) to a 2D array of Processing Element (PE) islands. These islands form a reconfigurable grid capable of executing both gating and expert computations. An on-chip global buffer enables low-latency data movement between external memory and PE islands.

Within each PE island, a unified systolic array architecture consists of dedicated input, weight, and expert buffers. The microarchitecture includes runtime configurable logic for selecting activation functions such as GELU or ReLU, and for toggling between dense and sparse matrix execution depending on the expert configuration. It also supports expert-level reordering and grouping, allowing flexible runtime scheduling based on workload balance and expert availability. Additionally, the

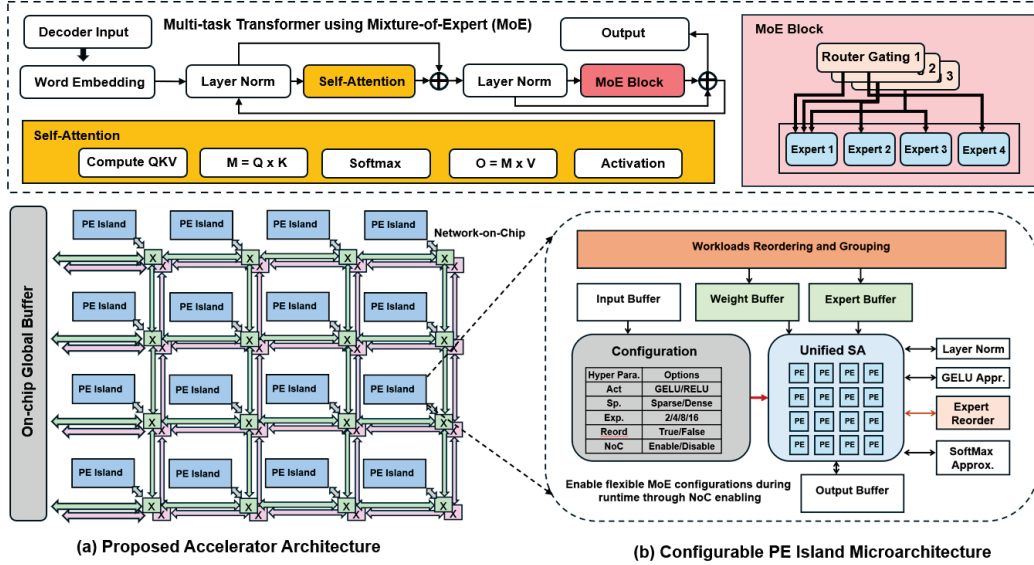


Fig. 1. Overview of MONET accelerator architecture. (a) The top-level architecture integrates MoE blocks within a multi-task transformer model, accelerated by a NoC-connected array of PE islands. (b) Each PE island features configurable logic supporting sparse/dense execution, expert-level reordering, activation functions, on-chip buffers, and a unified systolic-style PE array .

architecture provides fine-grained control over NoC routing behavior, enabling or disabling specific data movement paths depending on the execution phase.

A. Two-Tier Communication and PE Island Array

MONET integrates a 2D array of PE islands interconnected by a two-tier NoC, designed to support the different communication patterns of MoE execution. These islands execute either gating functions or expert layers depending on the computation phase. The NoC architecture consists of three logically decoupled routing planes, each aligned with a specific stage of the MoE processing pipeline: gating, expert weight delivery, and token-expert execution with output aggregation. These stages are illustrated in Figure 2.

In the first phase, shown in Figure 2(a), input tokens are broadcast into the array from the entry point at the top left and routed horizontally across multiple gating units. Tokens and gating weights are distributed using multicast-capable routers that support multiple simultaneous outputs per flit. This allows efficient parallel gating evaluations with minimal injection overhead. Notably, these routers also support bidirectional links and dynamic paths, which is particularly useful when token-expert mappings are sparse or spatially skewed.

The second phase, depicted in Figure 2(b), handles expert weight distribution. After top- k expert selection, weights for the activated experts are routed to their designated PE islands. Each island may host one or more experts depending on the configuration. This routing layer supports concurrent, targeted delivery of expert weights and avoids contention with token and output traffic by operating on an isolated communication plane.

The third phase, shown in Figure 2(c), is responsible for dispatching tokens vertically to selected expert islands and

collecting the resulting expert outputs. Since tokens may be shared across multiple experts, this plane enables token reuse through vertical multicast paths. Once expert computations are complete, their outputs are routed to aggregation units. Here, low-latency, bypass-enabled routers are employed to accelerate output fusion when contention is low. These routers allow flits to skip standard routing pipeline stages when downstream paths are idle.

B. Reconfigurable PE Island Microarchitecture

To support the diverse computational demands of gating functions and expert layers in MoE models, each PE island is designed as a reconfigurable compute unit capable of executing a range of activation, sparsity, and expert configurations. As shown in Figure 1(b), the PE island includes a unified systolic array (SA) core, programmable buffers, and a lightweight configuration controller that collectively enables dynamic specialization at runtime.

Each island is equipped with three distinct buffers for inputs, weights, and expert parameters connected to a configuration unit that manages mode selection. This unit controls execution modes through a set of programmable hyperparameters, including activation function (GELU or ReLU), execution sparsity (sparse or dense), and expert parallelism (e.g., 2, 4, 8, or 16 active experts per island). Additional flags allow for runtime toggling of expert reordering and NoC logic, enabling fine-grained workload tailoring and dynamic communication behavior. The unified SA processes token-expert matrix multiplications in a weight-stationary dataflow. It supports both sparse and dense computation pipelines, depending on the activation pattern and the current expert workload. Post-processing units surrounding the SA include support for layer normalization, softmax approximation for gating, and GELU approximation

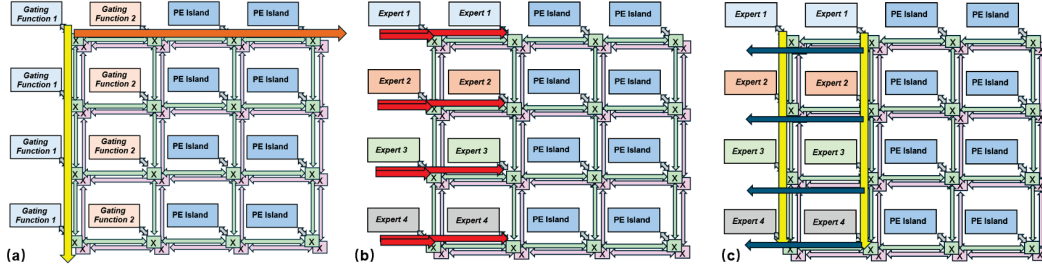


Fig. 2. Illustration of the two-tier NoC-based architecture. (a) Multicast input tokens and gating weights, (b) expert weights routed to selected PE islands, and (c) input tokens dispatched to experts with output accumulation.

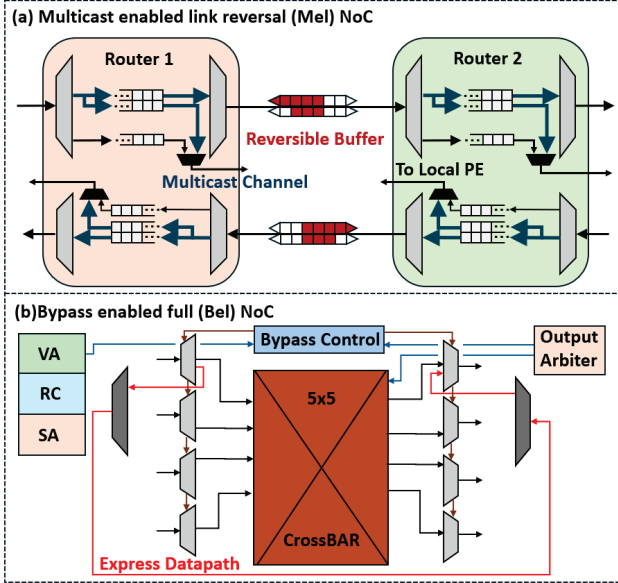


Fig. 3. Specialized router microarchitectures: (a) Multicast-enabled link reversal (*Mel*) NoC; (b) Bypass-enabled full (*Bel*) NoC.

for expert activations. An expert reordering module facilitates load balancing by enabling tokens to be reassigned or reordered across available compute units when required. This level of reconfigurability allows each PE island to adapt dynamically across gating and expert phases, maximizing resource utilization and maintaining execution efficiency in the presence of variable token sparsity, batch sizes, or model configurations.

C. Specialized NoC Router Microarchitectures

MONET relies on the customized NoC infrastructure tailored to the highly irregular and sparse traffic patterns of MoE workloads. To efficiently handle both multicast traffic during the gating phase and high-throughput expert communication during aggregation, we design two specialized router microarchitectures, illustrated in Figure 3.

Multicast-enabled Link Reversal *Mel* Router: The *Mel* router is designed to enable efficient data distribution during the gating and expert weight broadcast phases, which require a single token or parameter to be simultaneously delivered to multiple destinations. As illustrated in Figure 3(a), it features dual-directional data paths, internal multicast-capable buffers,

and dynamic link reversal. Unlike conventional routers that forward data based on fixed directional priorities, *Mel* routers support reverse-path routing where output ports can be temporarily reassigned as inputs, enabling back-propagation of flits under certain scheduling configurations. This allows tokens to traverse irregular topologies efficiently, especially when expert selections are non-contiguous or spatially distributed.

Internally, multicast support is implemented using a combination of packet duplication at the buffer level and arbitration logic that can drive multiple output ports per cycle. We adjust the design principle established in R-NoC [27] and extend the multicast capability to mesh topologies, allowing distributed multicast without centralized replication units. This approach significantly reduces injection latency and alleviates network congestion by enabling direct, one-hop replication within the router. A key architectural enhancement in *Mel* is its support for reversible links. In scenarios where token-to-expert mappings vary dynamically across inputs, traditional unidirectional routing leads to inefficient path utilization and increased hop counts. By enabling reverse-path injection, *Mel* routers can flexibly reroute flits toward underutilized or spatially distant expert tiles, mitigating serialization bottlenecks and improving overall bandwidth utilization.

Bypass-enabled Full *Bel* Router: To optimize expert result aggregation and minimize per-hop latency, we design the *Bel* router, shown in Figure 3(b). The *Bel* router architecture features a 5×5 crossbar switch with integrated bypass paths. When there is no contention on the output port, incoming flits can bypass standard routing computation (RC), virtual channel allocation (VA), and switch allocation (SA) stages, allowing single-cycle forwarding. The Bypass Control unit dynamically detects conflict-free conditions and redirects eligible flits along the fast path. This significantly reduces traversal latency for sparsely activated experts and boosts overall throughput under low-contention scenarios.

D. Multicast Scheduling and Reuse-Driven Dataflow

The execution flow of MONET accelerator is driven by a tightly integrated hardware-software co-design that combines multicast-aware token scheduling with local buffer reuse to sustain high throughput under sparse activation. The hardware supports three major stages: gating, expert execution, and output aggregation, each managed via a lightweight software scheduler and executed across the two-tier NoC and reconfig-

urable PE islands. The scheduler begins by analyzing the gating network output to assign each input token to its top- k selected experts. These assignments are compiled into routing metadata that drives multicast token injection across the NoC. During the gating phase, tokens and corresponding gating weights are simultaneously multicast to multiple gating units in parallel, reducing redundant data movement and enabling concurrent top- k expert selection. Once expert selections are finalized, input tokens are reused and multicast to multiple PE islands hosting the selected experts. Each PE island supports double-buffered input and weight SRAMs, enabling overlapping of compute and communication. Expert weights are fetched from the global buffer and broadcast in parallel to target PE islands using decoupled routing lanes, avoiding contention with token movement. Each PE island retains token data locally, allowing the same token to be reused across multiple expert evaluations without reinjection. The results from active experts are then routed to output aggregation units, where weighted outputs are combined to complete the MoE inference.

IV. EVALUATION

A. Experimental Methodology

We evaluate the MONET accelerator using a combination of system-level simulation, RTL-based synthesis, and NoC cycle modeling. Our simulation framework models the full accelerator stack, including the two-tier NoC, reconfigurable PE islands, multicast-aware scheduling, and expert aggregation. We implemented the accelerator architecture in Verilog and synthesize them using Synopsys Design Compiler with a commercial 22nm standard-cell library. SRAM modules were modeled using CACTI 7.0 for access latency and dynamic power estimation. Router delays extracted from RTL synthesis were integrated into the cycle-accurate simulator to calibrate NoC behavior. We synthesized MONET with the configuration of a 4×4 grid of PE islands (8×8 PEs per island) interconnected via the two-tier NoC, operating at 300 MHz and 0.8V supply. Three representative MoE workloads were benchmarked: M3ViT (vision-oriented with dense experts), DeepSpeed-MoE (sparse and irregular expert routing), and Switch-base-8 (large-scale, high top- k expert usage).

As a general dense compute baseline, we simulate a 2D systolic array using SCALE-Sim [26], calibrated to the same MAC count and bandwidth as MONET without MoE-specific optimizations. For a reconfigurable design comparison, we include EdgeMoE [14], a task-sparse, multi-task transformer architecture implemented on FPGA. Performance and resource usage for EdgeMoE are extracted from its published reports and normalized to match MONET’s hardware resources. Finally, we compare against Space-Mate [15], an ASIC accelerator optimized for sparse MoE inference.

B. RTL Synthesis Results

The synthesized RTL architecture of MONET comprises five primary components: a global on-chip buffer, a 4×4 array of reconfigurable PE islands, multicast-enabled *Mel* routers, low-latency *Bel* routers, and a centralized control unit. Each PE

TABLE I
RTL AREA AND POWER BREAKDOWN BY MODULE IN MONET

| Module | Area (mm ²) | Power (mW) |
|--------------------------------------|-------------------------|-------------|
| Global Buffer | 14.7 | 420 |
| PE Islands (MAC + Buffer) | 39.5 | 2180 |
| <i>Mel</i> Routers (Multicast NoC) | 6.2 | 125 |
| <i>Bel</i> Routers (Aggregation NoC) | 8.1 | 138 |
| Control Unit | 5.6 | 133 |
| Total | 74.1 | 2996 |

TABLE II
LATENCY COMPARISON BY PLATFORM (BATCH SIZE = 16 TOKENS)

| Platform | M3ViT (μ s) | DeepSpeed (μ s) | Switch-base (μ s) |
|----------------|------------------|----------------------|------------------------|
| Systolic Array | 8824 | 496 | 140918 |
| EdgeMoE | 1295 | 184 | 12465 |
| Space-Mate | 873 | 120 | 5483 |
| MONET | 550 | 62 | 3195 |

island integrates a systolic MAC array, double-buffered SRAM, and a local workload controller. The NoC routers implement phase-specific communication logic to optimize multicast distribution and output aggregation under sparse MoE workloads. Table I summarizes the post-synthesis area and dynamic power consumption for each module. The PE islands dominate both area and power due to dense MAC arrays and buffer structures designed to support token reuse and sparse activation. The *Mel* and *Bel* routers, while compact in area, provide critical routing functionality for data delivery. The centralized control unit, which orchestrates token scheduling and expert mapping, remains lightweight in both footprint and energy.

C. Performance Evaluation Results

1) *Latency Comparison*: Table II presents the average per-batch inference latency (batch size = 16 tokens) for three representative MoE workloads across four hardware platforms. MONET consistently achieves the lowest latency across all workloads, demonstrating the effectiveness of its dataflow-aware architecture and sparse workload specialization.

The systolic array baseline, while area-efficient and commonly used in dense compute tasks, suffers from excessive latency on sparse MoE models due to the lack of conditional execution, dynamic expert routing, or reuse mechanisms. In contrast, EdgeMoE leverages task-level sparsity on an FPGA and achieves improved performance for lightweight models like DeepSpeed-MoE, but its latency increases significantly for larger-scale workloads due to memory bandwidth constraints and limited on-chip reuse. Space-Mate, a fixed-function ASIC optimized for NeRF-SLAM with static expert clustering, outperforms EdgeMoE by applying hardware token reuse and expert grouping. However, it lacks fine-grained token scheduling and dynamic routing capabilities. By leveraging a two-tier NoC with multicast-enabled token delivery, bypassed expert aggregation, and double-buffered local memory, MONET achieves up to $8.5 \times$ lower latency than EdgeMoE and $2 \times$ improvement over Space-Mate on large-scale tasks like Switch-base-8.

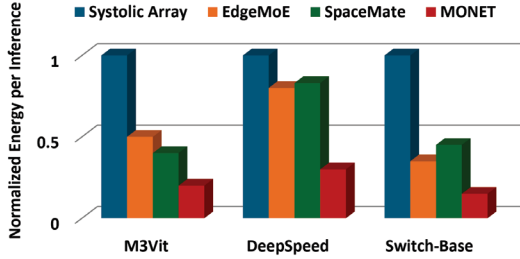


Fig. 4. Normalized energy consumption per inference across MoE workloads.

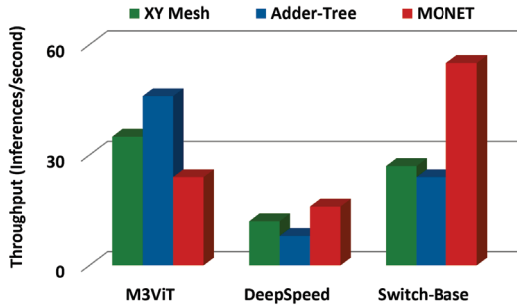


Fig. 5. Comparison of throughput for XY mesh, adder-tree and MONET for MoE workloads.

2) *Energy Efficiency Comparison*: Figure 4 illustrates the normalized energy per inference across three representative MoE workloads. Energy consumption is normalized to the Systolic Array baseline for each workload. MONET achieves the lowest energy usage, demonstrating the effectiveness of MoE-specific optimizations in dataflow, reuse, and interconnect. For M3ViT, MONET consumes less than one-third the energy of EdgeMoE and significantly outperforms both SpaceMate and the dense systolic baseline. The gap becomes more pronounced in sparse workloads such as Switch-base-8, where MONET achieves over $4\times$ lower energy compared to SpaceMate and more than $6\times$ reduction relative to the Systolic Array. EdgeMoE performs reasonably well on small-to-moderate workloads, leveraging task sparsity, but suffers from off-chip memory bandwidth limitations and control overhead inherent to FPGA implementations. SpaceMate achieves better efficiency than EdgeMoE for larger models, due to hardware token reuse and clustering, but lacks runtime reconfigurability and token scheduling flexibility. MONET combines multicast-aware routing, reconfigurable PE islands, and scheduling-guided data reuse to minimize both memory movement and active compute cycles.

D. Impact of NoC Design in MONET on MoE Performance

To evaluate the architectural benefits of MONET’s multicast-aware two-tier NoC, we compare it against two baseline topologies: a conventional 2D XY mesh and a source-routed tree-based multicast (Adder-Tree). All configurations share the same PE array, memory buffers, and workload scheduler as adopted in MONET to ensure a fair evaluation of NoC-level performance. Figure 5 shows throughput (inferences per second) across three MoE workloads. The XY mesh baseline suffers

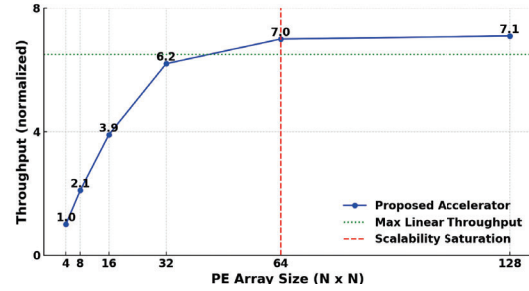


Fig. 6. Throughput scaling of MONET as the PE array size increases from 4×4 to 128×128 .

from high communication overhead due to serialized routing, lack of multicast support, and inefficient token reuse, which collectively limit parallelism and PE utilization. The Adder-Tree design offers moderate gains through token replication, but its hierarchical structure introduces path serialization and branching bottlenecks. The two-tier NoC of MONET achieves higher throughput across all workloads—up to $2.1\times$ over the XY mesh and $1.6\times$ over the tree design. This improvement is driven by the *Mel* routers’ in-network multicast replication and the *Bel* routers’ single-cycle expert aggregation paths. Combined with the scheduling-aware buffer design, the NoC sustains higher token delivery rates and compute overlap, leading to improved throughput and resource utilization for sparse and dynamic MoE execution patterns.

We examine how MONET scales as the PE array grows from 4×4 to 128×128 . As shown in Figure 6, throughput increases nearly linearly up to a 64×64 configuration, highlighting the effectiveness of MONET’s multicast-aware NoC and parallel token dispatch mechanisms. At 64 PEs, the design reaches its optimal throughput operating point. Beyond this, saturation is observed due to routing congestion, shared buffer contention, and the inherent sparsity of MoE workloads.

V. CONCLUSIONS

This paper presents MONET, a domain-specialized hardware accelerator designed to meet the unique demands of MoE inference. The architecture is built around a two-tier NoC that separates token distribution and expert aggregation across distinct routing planes. The proposed *Mel* and *Bel* router microarchitectures enable in-network multicast and low-latency expert fusion, respectively, while reconfigurable PE islands support sparse and dense compute modes with double-buffered memory for overlapping data movement and execution. MONET is co-designed with a lightweight runtime scheduler that dynamically maps tokens to experts and balances workloads across the compute fabric. Across MoE benchmarks, MONET achieved up to $8.5\times$ lower latency and over $6\times$ improvement in energy efficiency, demonstrating consistent gains across both compact and large-scale sparse inference scenarios.

VI. ACKNOWLEDGMENT

This research was partially supported by NSF grants CCF-1936794, CCF-2324645, and CCF-2311544.

REFERENCES

- [1] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv:1701.06538*, 2017.
- [2] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "Gshard: Scaling giant models with conditional computation and automatic sharding," *arXiv preprint arXiv:2006.16668*, 2020.
- [3] R. Aljundi, P. Chakraborty, and T. Tuytelaars, "Expert gate: Lifelong learning with a network of experts," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3366–3375.
- [4] Z. Chen, Y. Deng, Y. Wu, Q. Gu, and Y. Li, "Towards understanding the mixture-of-experts layer in deep learning," *Advances in neural information processing systems*, vol. 35, pp. 23 049–23 062, 2022.
- [5] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand *et al.*, "Mixtral of experts," *arXiv preprint arXiv:2401.04088*, 2024.
- [6] N. Du, Y. Huang, A. M. Dai, S. Tong, D. Lepikhin, Y. Xu, M. Krikun, Y. Zhou, A. W. Yu, O. Firat *et al.*, "Glam: Efficient scaling of language models with mixture-of-experts," in *International conference on machine learning*. PMLR, 2022, pp. 5547–5569.
- [7] Z. Fan, R. Sarkar, Z. Jiang, T. Chen, K. Zou, Y. Cheng, C. Hao, Z. Wang *et al.*, "M³vit: Mixture-of-experts vision transformer for efficient multi-task learning with model-accelerator co-design," *Advances in Neural Information Processing Systems*, vol. 35, pp. 28 441–28 457, 2022.
- [8] A. Liu, B. Feng, B. Wang, B. Wang, B. Liu, C. Zhao, C. Deng, C. Ruan, D. Dai, D. Guo *et al.*, "Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model," *arXiv preprint arXiv:2405.04434*, 2024.
- [9] Snowflake, "Arctic: An open, efficient foundation for language models on snowflake," <https://www.snowflake.com/en/blog/arctic-open-efficient-foundation-language-models-snowflake/>, 2023.
- [10] XAI Organization, "Grok-1," <https://github.com/xai-org/grok-1>, 2023.
- [11] B. Zoph, I. Bello, S. Kumar, N. Du, Y. Huang, J. Dean, N. Shazeer, and W. Fedus, "St-moe: Designing stable and transferable sparse expert models," *arXiv preprint arXiv:2202.08906*, 2022.
- [12] F. Xue, Z. Zheng, Y. Fu, J. Ni, Z. Zheng, W. Zhou, and Y. You, "Openmoe: An early effort on open mixture-of-experts language models, 2024," *URL* <https://arxiv.org/abs/2402.01739>.
- [13] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *Journal of Machine Learning Research*, vol. 23, no. 120, pp. 1–39, 2022.
- [14] R. Sarkar, H. Liang, Z. Fan, Z. Wang, and C. Hao, "Edge-moe: Memory-efficient multi-task vision transformer architecture with task-level sparsity via mixture-of-experts," *arXiv preprint arXiv:2306.01831*, 2023.
- [15] G. Park, S. Song, H. Sang, D. Im, D. Han, S. Kim, H. Lee, and H.-J. Yoo, "20.8 space-mate: A 303.5mw real-time sparse mixture-of-experts-based nerf-slam processor for mobile spatial computing," in *Proceedings of the 2024 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 67. IEEE, 2024, pp. 374–376.
- [16] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.
- [17] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the em algorithm," *Neural computation*, vol. 6, no. 2, pp. 181–214, 1994.
- [18] W. J. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," *Proceedings of the 38th Design Automation Conference (DAC)*, pp. 684–689, 2001.
- [19] L. Benini and G. De Micheli, "Networks on chips: A new soc paradigm," *Computer*, vol. 35, no. 1, pp. 70–78, 2002.
- [20] N. E. Jerger, L.-S. Peh, and M. Lipasti, "Virtual circuit tree multicasting: A case for on-chip hardware multicast support," in *2008 International Symposium on Computer Architecture*, 2008, pp. 229–240.
- [21] W.-C. Kwon, T. Krishna, and L.-S. Peh, "Locality-oblivious cache organization leveraging single-cycle multi-hop nocs," in *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 715–728. [Online]. Available: <https://doi.org/10.1145/2541940.2541976>
- [22] Y.-H. e. a. Chen, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2016.
- [23] M. e. a. Kwon, "Maeri: Enabling flexible dataflow mapping over dnn accelerators via reconfigurable interconnects," in *ASPLOS*, 2018.
- [24] S. e. a. Han, "Eie: Efficient inference engine on compressed deep neural network," in *ISCA*, 2016.
- [25] D. C. Jung and M. Taylor, "Evaluating ruche networks: Physically scalable, cost-effective, bandwidth-flexible nocs," in *Proceedings of the 52nd Annual International Symposium on Computer Architecture*, ser. ISCA '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 1035–1048. [Online]. Available: <https://doi.org/10.1145/3695053.3731010>
- [26] D. Narayanan, X. Phan, A. Gholami, A. Azad, K. Keutzer, A. Nikolaev, N. Ardalani, and D. Milojkic, "Scale-sim: Systolic cnn accelerator simulator," in *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, 2019, pp. 46–55.
- [27] M. Abdel-Majeed and V. Betz, "Scalencoc: A scalable evaluation framework for on-chip networks of manycore accelerators," in *IEEE Transactions on VLSI Systems*, 2020.