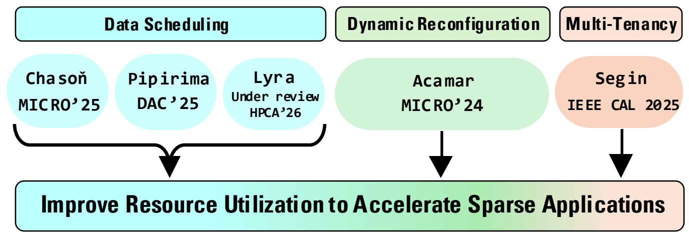
Architecting Efficiency - Data Scheduling, Dynamic Reconfiguration, and Multi-Tenancy for Sparse Acceleration

Ubaid Bakhtiar
Advisor: Bahar Asgari
University of Maryland, College Park
Graduation Date: May 2027

Research Statement

As data volumes continue to grow exponentially across domains from scientific computing and graph analytics to machine learning, computation has become increasingly dependent on the exploitation of sparsity. Sparsity, referring to the prevalence of zeros in data, has evolved into a deliberate design goal, playing a critical role in minimizing data movement, reducing memory footprint, and accelerating computation. However, the existence of sparsity also presents a significant architectural challenge: poor resource utilization, which leads to wasted silicon area and limited system scalability. The issue is exacerbated even more in the twilight of Moore's law when the emphasis shifts from simply adding more transistors to extracting more performance from existing hardware. *General-purpose processors and specialized accelerators often fail to achieve the theoretical gains offered by sparsity due to underutilized hardware resources. To close the gap between peak and achievable performance, techniques such as data scheduling, multi-tenancy, and dynamic reconfiguration must be integrated into sparse applications hardware to mitigate the high resource underutilization and consequently, fully exploit the benefits of sparsity.* This research addresses the challenges caused by sparsity by demonstrating that data scheduling, dynamic reconfiguration, and multi-tenancy are complementary techniques that significantly improve resource utilization in sparse applications, each targeting distinct inefficiencies. By applying these techniques across different layers of the hardware stack, this research demonstrates substantial improvements in utilization and throughput, building toward a future where sparse acceleration is not only performant but scalable and adaptable. Figure 1 provides an overview of this research, along with the venues where the work has been peer-reviewed and published.

Figure 1. Publications of this research in different venues



Data Scheduling. This research proposes low-cost, novel data scheduling and placement strategies to enable the streamlined execution of sparse kernels by improving the resource utilization of compute cores.

1. One such strategy is employed in **Chasoň**, a sparse accelerator designed to enhance processing element (PE) efficiency via a novel data scheduling scheme called Cross-HBM Channel Out-of-Order Scheduling (CrHCS)**.** CrHCS extends intra-channel non-zero scheduling to the inter-channel domain**,** enabling migration of non-zero values across HBM channels. This migration helps mitigate PE stalls by ensuring a more balanced and continuous supply of useful data (non-zeros) to idle compute units. The architecture of Chasoň is implemented on AMD Xilinx Alveo U55C and achieves 301MHz clock frequency. The architecture demonstrates strong performance gains: up to $8\times$ over Serpens (state-of-the-art SpMV accelerator), $20.33\times$ over Nvidia RTX 4090 (cuSparse), $11.65\times$ over Nvidia RTX A6000 (cuSparse) and $2.67\times$ over Intel Core i9 (Intel MKL) respectively.

1. Given the widespread adoption of large language models, this research proposes **Lyra,** an efficient architecture for unstructured sparse LLMs. Unlike fixed matrix partitioning, this research advocates for dynamic matrix partitioning such that the number of non-zeros within a tile remains fixed but the tile size becomes variable. Fixing the number of non-zeros per tile ensures high resource utilization, translating directly into improved performance and energy efficiency. The architecture of Lyra is implemented on AMD Xilinx Alveo U55C and achieves 299MHz clock frequency. Lyra achieves a peak throughput of 99.8 $GFLOP/s$ and delivers up to 4.6 $GFLOP/s$ per $watt$ across Llama2-13B, Llama2-7B, OPT-125M and OPT-2.7B models. Lyra exhibits $2\times$ better resource utilization than DFX (state–of–the–art LLM accelerator) and reduces the latency by $5\times$. Lyra offers up to $9\times$ better performance–per–watt than Nvidia RTX Ada 6000 and Nvidia RTX 4080.

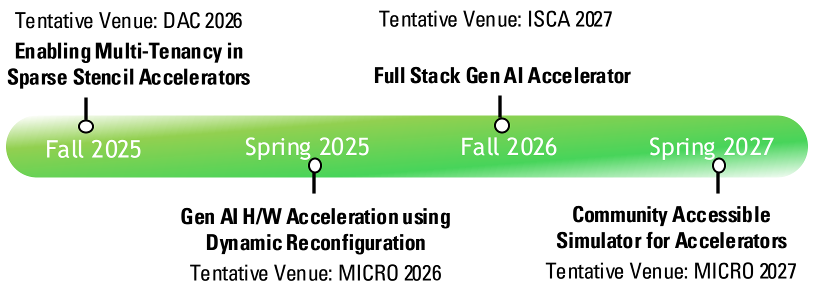
3. This research also introduces **Pipirima**, a prediction-driven on-chip memory scheduling architecture. This approach employs a lightweight, rapid sparsity pattern predictor based on counter-based prediction (CBP), similar to branch predictors. The prediction method predicts the structural properties of the sparse matrix and schedules the data on runtime on the on-chip memory blocks. This allows for fine-grained parallelism among the compute while minimizing resource underutilization. Pipirima demonstrated a significant speed up compared to baseline sparse accelerators; ExTensor and Tensaurus at the expense of very small memory and computational overhead of prediction related components.

Dynamic Reconfiguration. This research enables dynamic reconfiguration based on runtime profiling of the structural characteristics of the underlying sparse workloads and the quality of intermediate outputs to improve resource utilization of the sparse workload. This research proposes **Acamar**, a dynamically reconfigurable FPGA-based design that has the ability to not only reconfigure the FPGA fabric to different solvers but also reconfigure the sparse computational unit based on the structural properties of coefficient matrix A. To achieve this, Acamar can seamlessly switch between three widely used iterative solvers. Moreover, Acamar can reconfigure the sparse computational unit, to ensure enhanced resource utilization. Acamar also offers a multi-level iterative decision chain to minimize the reconfiguration rate to incur less reconfiguration overhead. Experiments show improvement in resource utilization and performance over a static design and Nvidia GTX 1650 Super GPU.

Multi-Tenancy. This research focuses on extend the single workload execution model of sparse accelerators, which pose challenges in multi-tasking scenarios common in real-world deployments. When multiple workloads need to be executed sequentially, the host must repeatedly reconfigure the accelerator—reloading data into off-chip memory such as HBM for each job. This constant setup and teardown process introduces substantial latency and bandwidth overhead, severely impacting throughput and amplifying already prevalent resource underutilization. This research proposes **Segin**, a fine-grained multi-tenant approach that leverages idle resources to run multiple sparse workloads simultaneously, synergistically reducing inefficiencies and improving throughput. Segin leverages a lightweight bitmap-based search mechanism to quickly and efficiently identify overlapping indices using a hierarchical bitmap approach.

Future Work. These advancements pave the way for future work that primarily focuses on building hardware solutions and accelerators for sparse applications that supports efficient data scheduling, dynamic reconfiguration and multi-tenancy achieving maximum performance given limited resources. The future trajectory of this research is structured into four cohesive projects. The first project aims to extend multi-tenancy into state-of-the-art sparse stencil accelerators, enabling concurrent execution of multiple sparse workloads and addressing existing limitations in throughput and resource utilization. The second project focuses on employing the concepts of dynamic reconfiguration to accelerate generative AI models like LLMs and Vision transformers. Building upon these efforts, the third project seeks to unify data scheduling, dynamic reconfiguration, and multi-tenancy into a holistic, full-stack accelerator architecture for generative AI—designed for high performance, energy efficiency, and real-world deployment in environments like data centers. Finally, the fourth project will culminate in the development of a community-accessible cycle-accurate simulator, built upon the architectural insights gathered throughout this research. This simulator will allow users to model the performance and behavior of sparse accelerators across varying configurations and workloads. A high-level roadmap and tentative venues for these projects are outlined in Figure 2.

Figure 2. Roadmap of future works and tentative venues


Tentative Venue: DAC 2026
Enabling Multi-Tenancy in Sparse Stencil Accelerators
Tentative Venue: ISCA 2027
Full Stack Gen AI Accelerator
Fall 2025
Spring 2025
Fall 2026
Spring 2027
Gen AI H/W Acceleration using Dynamic Reconfiguration
Tentative Venue: MICRO 2026
Community Accessible Simulator for Accelerators
Tentative Venue: MICRO 2027

Conclusion. Collectively, this research demonstrates that addressing sparsity requires novel approaches across architecture, memory systems, and runtime adaptability. By developing lightweight data schedulers, reconfigurable compute units, and fine-grained multi-tenancy strategies, this research have taken concrete steps toward making sparse hardware acceleration more efficient, scalable, and deployable. Looking ahead, my long-term goal is to design practical accelerator stacks that bridge the gap between hardware potential and real-world application demands—particularly in domains like scientific computing and generative AI. I am excited to continue exploring research at the intersection of efficiency, adaptability, and usability, contributing both systems and tools.