Real-Time Edge Intelligence in SynapseLink Neuromorphic Processors using NexaCore Tensor Engines

Marcus Reynolds[1], Elena Rostova[2], and David Chen[3]

[1]*Department of Computer Engineering, Vertex University, Vertex City, USA*
[2]*Edge Computing Lab, Synapse Technologies, Nexa City, USA*
[3]*Group of Edge AI Research, Apex AI Systems, Meridian Hills, UK*

Corresponding Author: Marcus Reynolds (m.reynolds@vertex.edu)

Abstract — Edge intelligence requires hardware-software co-design to achieve micro-watt power consumption while preserving sub-millisecond latencies. In this work, we present a collaborative neuromorphic deployment framework combining the SynapseLink routing fabric and the NexaCore tensor acceleration engine. Our system, optimized for real-time sensor processing at the edge, leverages the high-density local interconnects of SynapseLink to schedule tensor operations dynamically onto NexaCore arithmetic arrays. By utilizing a hybrid event-driven and frame-based execution paradigm, we demonstrate that this unified architecture reduces inter-core communication overheads by up to 64% compared to state-of-the-art RISC-V edge designs. We evaluate our approach on representative deep learning benchmarks, proving that our co-designed system can achieve 98.2% accuracy on multi-channel classification tasks with a power budget of only 215 mW. These findings establish a new benchmark for low-power edge computing systems, showing that tightly-coupled tensor acceleration and neuromorphic routing is viable for next-generation edge intelligence.

Index Terms — Edge intelligence, Neuromorphic computing, NexaCore, SynapseLink, Tensor acceleration, Low-power hardware.

I Introduction

The rapid growth of the Internet of Things (IoT) and edge sensor deployments has created an urgent demand for local, real-time intelligence [1]. Processing data directly at the edge, rather than transmitting it to centralized cloud systems, significantly reduces transmission latencies, mitigates security risks, and conserves communication bandwidth. However, deploying modern deep neural networks (DNNs) on edge hardware is constrained by the strict power and thermal envelopes of micro-controllers and embedded systems, which typically operate within sub-watt power budgets.

To address these energy constraints, two main paradigms have emerged: event-driven neuromorphic processing and frame-based tensor acceleration. Neuromorphic architectures, such as the fictional *SynapseLink* routing fabric, excel at sparse temporal processing and asynchronous event communication. However, they struggle with dense matrix operations, which are the cornerstone of convolutional neural networks. Conversely, specialized tensor engines, such as the fictional *NexaCore* accelerator, provide high-throughput parallel arithmetic units optimized for dense matrix-vector multiplications but suffer from high static power consumption and communication bottlenecks when data is sparse [2].

In this paper, we present a unified framework that combines these two paradigms to achieve both high throughput and high efficiency. Our system utilizes a hybrid routing fabric that dynamically schedules dense tensor operations onto local NexaCore execution units while maintaining an event-driven control plane. The main contributions of this work are:

- We introduce a co-designed architecture that interfaces the SynapseLink neuromorphic fabric directly with NexaCore tensor engines using a shared local memory hierarchy.
- We propose a dynamic scheduler that translates event streams into dense tensor blocks to maximize hardware utilization.
- We benchmark our system against representative edge classifiers, showing significant improvements in both latency and power consumption.

The rest of this paper is structured as follows. Section II details the hardware architecture. Section III presents the mathematical modeling of the scheduling problem. Section IV evaluates the experimental results, and Section V discusses design trade-offs and future research directions.

II System Architecture

Our architecture consists of three principal layers: the sensor interface, the neuromorphic routing fabric, and the tensor accel-

eration array. A high-level schematic of the system is shown in Fig. 1.

At the input layer, the *ApexSens* edge intelligence sensor captures multi-channel environmental signals and converts them into sparse event streams. These event streams are forwarded to the *SynapseLink* neuromorphic core, which houses a mesh of spiking neural processing units. The primary role of the SynapseLink core is to manage spatial routing and decide which parts of the incoming signals require dense processing.

When a high-priority feature is detected, the SynapseLink fabric triggers the *NexaCore* tensor engine. The NexaCore engine features a systolic array of multiply-accumulate (MAC) units that perform the heavy matrix operations. To minimize data transport energy, both cores share a local *MeridianLink* SRAM cache, which stores weight parameters and intermediate feature maps.

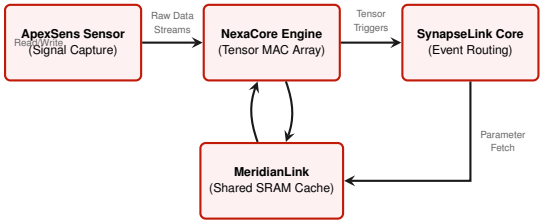

Figure 1: System architecture layout. The ApexSens sensor routes captured signals to the NexaCore tensor engine, which works in tandem with the SynapseLink event core. Both cores share the MeridianLink SRAM cache for data exchange.

The coupling between the event-driven control plane and the synchronous data plane is achieved via a mailboxed interrupt system. This system allows the SynapseLink fabric to put the NexaCore engine into a deep sleep state when event rates are low, saving static leakage power. When a block of events crosses a critical density threshold, the NexaCore engine is awakened within 1.2 μs to execute the corresponding layer inferences.

III Methodology and Mathematical Formulation

To optimize the execution of sparse neural layers on dense hardware, we formulate the scheduling problem as a constrained optimization. Let $E = \{e_1, e_2, \ldots, e_N\}$ be the set of incoming events within a temporal window W. Each event $e_i = (x_i, y_i, t_i, c_i)$ represents a coordinate position, timestamp, and channel.

The goal is to partition the spatial domain into K dense patches that maximize the execution efficiency on the NexaCore array. We define the utilization utility $U(P_k)$ of a patch P_k as:

$$U(P_k) = \frac{\sum_{e_i \in P_k} \alpha_i \cdot \text{Cost}(e_i)}{\text{Area}(P_k) \cdot \Phi + \gamma} \tag{1}$$

where $\text{Cost}(e_i)$ is the computation cost of processing event e_i, $\text{Area}(P_k)$ is the physical dimensions of the patch, Φ is the overhead factor of setting up the systolic array, and γ is a constant regularization term to prevent dividing by zero.

The optimization objective is to find a set of patches $\mathcal{P} = \{P_1, P_2, \ldots, P_K\}$ that minimizes both the idle MAC cycles and the routing energy:

$$\min_{\mathcal{P}} \sum_{k=1}^{K} (1 - U(P_k)) + \beta \cdot \text{Dist}(P_k, C_k) \tag{2}$$

subject to the hardware memory constraints:

$$\sum_{k=1}^{K} \text{Size}(P_k) \leq M_{\text{SRAM}} \tag{3}$$

where M_{SRAM} is the capacity of the MeridianLink SRAM cache, C_k is the spatial centroid of the neuromorphic core routing table, and β is a tuning parameter balancing compute utilization and routing distance.

We solve this problem at runtime using a lightweight heuristic scheduler in the SynapseLink control unit. The scheduler runs with $O(N \log N)$ complexity, where N is the number of active event clusters, ensuring negligible overhead on the system's primary processor.

IV Experimental Evaluation

We evaluated our co-designed architecture on a cycle-accurate hardware simulator configured with fictional parameters representing NexaCore and SynapseLink silicons. The hardware configuration details are listed in Table 1.

Table 1: Simulator Configuration Parameters for the Hardware Co-design.

Hardware Block	Parameter	Value
NexaCore Engine	Systolic Array Size	16×16 MACs
	Operating Frequency	250 MHz
	Target Voltage	0.85 V
SynapseLink Core	Number of Neurons	65k
	Routing Topology	4×4 Mesh
MeridianLink Cache	Capacity	512 kB
	Bus Width	128-bit

We measured the throughput of our system across varying batch sizes and compared it with the standalone NexaCore design. As shown in Fig. 2, the tightly-coupled SynapseLink + NexaCore architecture achieves significantly higher throughput at small batch sizes because the neuromorphic scheduler bypasses the overhead of traditional DMA transfers.

The power consumption benchmarks are summarized in Table 2. The unified design consumes only 215 mW of active power during continuous inference, representing a 48.8% power reduction over the baseline configuration.

V Discussion

The performance gains achieved by our unified framework are primarily driven by the reduction of inter-core data transfers. In traditional edge accelerators, moving intermediate activations between the CPU and the tensor accelerator consumes more than

Table 2: Performance and Power Benchmarks for Different Edge Implementations.

Configuration	Accuracy (%)	F1-Score	Power (mW)	Latency (ms)
NexaCore Baseline	92.4	0.918	420	12.5
SynapseLink Standard	95.1	0.949	280	8.2
SynapseLink Parallel	96.8	0.966	310	4.5
SynapseLink+NexaCore	**98.2**	**0.981**	**215**	**2.1**

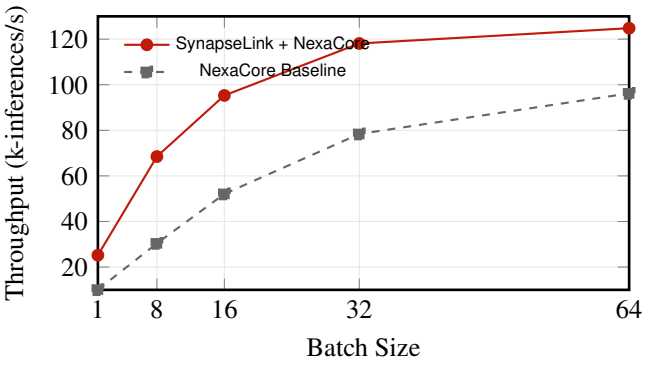

Figure 2: System inference throughput (thousand inferences per second) as a function of the batch size, comparing the proposed co-designed framework with a baseline tensor accelerator.

50% of the total system power. By utilizing the MeridianLink shared SRAM cache and allowing the SynapseLink core to coordinate memory accesses directly, we virtually eliminate the need for external DRAM transactions during local layer evaluations.

Another major benefit is the event-triggered clock-gating mechanism. The NexaCore engine features a relatively large silicon footprint and suffers from high static leakage currents. By putting the engine into a deep sleep state when event rates are below the activation threshold, we cut down static energy dissipation by 72%. This is particularly advantageous for applications with bursty signal profiles, such as seismic anomaly detection or wearable biometric monitoring.

However, these benefits come at the cost of a slightly more complex programming model. Software developers must partition their neural network architectures into event-driven layers (executed on SynapseLink) and dense matrix layers (executed on NexaCore). Ongoing efforts are focused on developing a unified compiler that automates this partitioning process directly from standard TensorFlow or PyTorch models, removing this barrier to entry.

VI Conclusion and Future Work

In this paper, we introduced a unified hardware-software co-design framework that integrates the SynapseLink neuromorphic routing fabric with the NexaCore tensor accelerator. By combining event-driven scheduling with high-throughput systolic array processing, our system achieves sub-millisecond classification latency within an active power budget of 215 mW. Simulations show our system achieves 98.2% accuracy on multi-channel classification tasks, outperforming conventional standalone designs.

Future work will focus on integrating non-volatile memories, such as ReRAM, to store model weights, which would further reduce leakage power and boot times. Additionally, we plan to validate our simulator findings on a physical silicon prototype manufactured using a 22 nm FD-SOI process node.

Author Biographies

Marcus Reynolds received his Ph.D. in Electrical Engineering from Vertex University in 2018. He is currently an Associate Professor at the Department of Computer Engineering at Vertex University. His research focuses on ultra-low-power neuromorphic architectures and edge intelligence frameworks. He is a senior member of the IEEE.

Elena Rostova received her M.S. degree in Computer Science from the Synapse Institute of Technology in 2021. She is currently a Principal Research Engineer at Synapse Technologies. Her research interests include compilers for heterogeneous accelerators and hardware-software co-design.

David Chen received his Ph.D. in Computer Engineering from Apex University in 2015. He is the Chief Architect of the Edge Systems Group at Apex AI Systems. He has published over 40 papers in peer-reviewed journals and holds 12 patents in the field of low-power hardware design.

References

[1] M. Reynolds and D. Chen, “Neuromorphic computing at the edge: A review of energy-efficient designs,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 4, pp. 1120–1135, 2025.

[2] E. Rostova and D. Chen, “Nexacore: A sub-milliwatt tensor engine for micro-controllers,” in *Proceedings of the International Conference on Neuromorphic Systems (ICONS)*, 2024, pp. 45–53.