Mapping a BERT-Style Encoder to the AMD Versal Network-on-Chip

Saad Syed, Ozayr Raazi, Chaitanya Sharma, Mohamed Bekdach
Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Canada
{sn3syed, oraazi, c34sharm, mtbekdac}@uwaterloo.ca

***Abstract*—Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based neural network model whose performance is driven by large matrix multiplications and high-bandwidth tensor movement between memory and compute. This paper investigates mapping a BERT-style encoder datapath onto the AMD Versal hardened Network-on-Chip (NoC). Our prototype streams activations and weights from external DDR through NoC-connected endpoints into programmable-logic (PL) compute blocks via AXI and DMA, and writes results back to DDR.**

The implemented system functionally verifies a complete multi-head self-attention datapath under multiple parameterizations. We first validate end-to-end correctness using a reduced bring-up configuration (4 tokens, 64-dimensional embeddings, 4 heads), and then demonstrate successful operation at larger parameters representative of meaningful scaling (32 tokens, 256-dimensional embeddings, 8 heads). Downstream encoder sub-layers, including self-output and feed-forward blocks, remain partially integrated and expose practical challenges related to AXI backpressure, buffering, and system-level scaling. Because DDR simulation models provide limited observability, functional correctness is validated using SystemVerilog backdoor inspection at key streaming boundaries prior to full closure with the Vivado NoC compiler flow. Overall, this work demonstrates the feasibility of NoC-centric integration for transformer workloads and highlights concrete scaling and verification challenges encountered in practice.

***Index Terms*—BERT, Transformer, FPGA, Network-on-Chip, Versal, AXI, DMA**

I. Introduction

Transformer-based models such as Bidirectional Encoder Representations from Transformers (BERT) [1] have become foundational to modern natural language processing workloads. A BERT model consists of multiple stacked encoder layers [2], each combining multi-head self-attention with a position-wise feed-forward network (FFN), residual connections, and layer normalization.

Although matrix multiplication dominates arithmetic complexity, practical accelerator performance is frequently limited by data movement rather than compute throughput. Large activation and weight tensors must be repeatedly transported between memory and compute units, making memory bandwidth, interconnect scalability, and flow control central design concerns.

AMD Versal devices integrate a hardened Network-on-Chip (NoC) that provides scalable, high-bandwidth connectivity between DDR memory controllers and programmable logic (PL) [3], [4]. Rather than relying on soft interconnect fabrics, the NoC enables structured routing, quality-of-service (QoS) guarantees, and higher sustainable bandwidth. This project explores how a realistic BERT-style encoder datapath maps onto this NoC-centric execution model, emphasizing system integration (NoC endpoints, DMA streaming, orchestration, and verification) as a first-class engineering task.

A. Contributions

The main contributions of this work are:

- A NoC-centric streaming architecture for a BERT-style encoder datapath using DDR→NoC→DMA→AXI-Stream transport.
- A functionally verified multi-head self-attention implementation operating on quantized data at both reduced and larger tensor dimensions, including an 8-head scaled configuration.
- A practical integration and verification methodology that exposes real-world scaling challenges related to AXI flow control, buffering, and simulation observability, including parameter-compatibility constraints imposed by the compute tiling structure.

II. Background and Motivation

A. BERT Encoder and Multi-Head Attention

Within a BERT encoder layer, an input activation matrix $X \in \mathbb{R}^{T\times D}$ is projected into query, key, and value matrices:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V. \tag{1}$$

Attention scores are computed and normalized as:

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{2}$$

In multi-head attention, the embedding dimension D is partitioned across H heads, enabling parallel attention over multiple subspaces. Our design supports multiple configurations; in this work we verify correct operation for both a reduced configuration used for deterministic bring-up and a larger configuration that stresses bandwidth, buffering, and tiling more realistically.

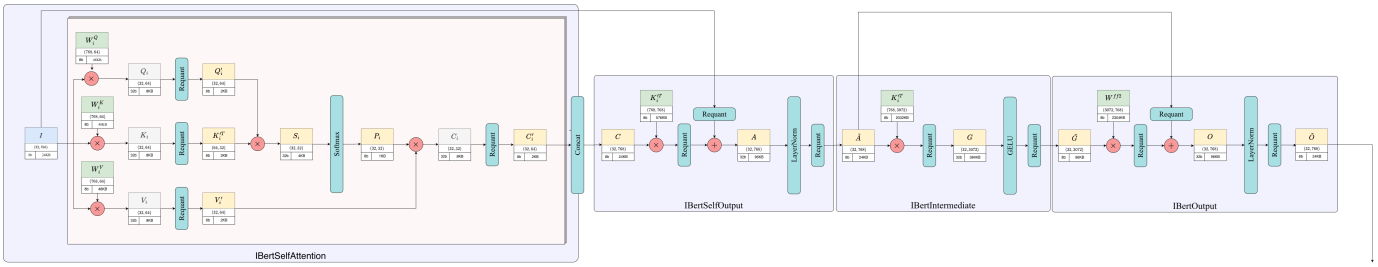

Fig. 1. Target BERT-style encoder datapath mapped onto the Versal NoC. Activations and weights are streamed from DDR through NoC-connected endpoints and DMA into AXI-Stream compute blocks implementing self-attention and downstream stages.

B. Motivation for a NoC-Centric Approach

As transformer models scale in width (embedding size, number of heads) and sequence length, traditional point-to-point or fabric-based interconnect schemes become increasingly difficult to manage. The Versal hardened NoC provides an alternative by offering high-bandwidth, structured connectivity between memory and compute with reduced routing complexity in the programmable fabric [3], [4]. This motivates an architecture where the NoC, rather than custom wiring, becomes the backbone for data movement, and where DMA streaming boundaries provide modularity and testability.

III. System Overview

A. Verified Configurations and Parameters

We validate functional correctness using two representative operating points: (i) a small configuration for fast iteration and deterministic debug, and (ii) a larger configuration corresponding to the active scaled self-attention bring-up. Table I summarizes these verified configurations.

TABLE I
VERIFIED SELF-ATTENTION CONFIGURATIONS

Parameter	Bring-up	Scaled
Tokens (T)	4	32
Embedding Dim. (D)	64	256
Attention Heads (H)	4	8
Head Dim. (D/H)	16	32
Input/Weight Width	8-bit	8-bit
Accumulator Width	32-bit	32-bit
Systolic Array ($N_1 \times N_2$)	2×2	2×2

In addition to these two verified points, our parameterization supports a small set of "pre-defined" shapes that satisfy the core tiling constraints (Section III-B) and are convenient for simulation-driven iteration (e.g., medium-size and base-like configurations with longer runtimes).

B. Design Constraints and Valid Parameter Choices

Beyond algorithmic correctness, our implementation imposes additional constraints on admissible (T, D, H) choices due to hardware partitioning, systolic tiling, and practical memory alignment. These constraints directly influenced our bring-up configurations and also shape scaling behavior.

(1) Integer head dimension. Multi-head attention requires an integer per-head dimension:

$$\texttt{HEAD\_DIM} = \frac{D}{H}, \quad \text{thus } D \bmod H = 0. \tag{3}$$

(2) Systolic array compatibility. The matrix multiplication core is tiled across a fixed $N_1 \times N_2$ systolic structure. To avoid fractional tiles and simplify control, we require the token dimension and head dimension to align with the systolic geometry:

$$T \bmod N_1 = 0, \quad \texttt{HEAD\_DIM} \bmod N_2 = 0. \tag{4}$$

In our setup, $N_1 = N_2 = 2$, so T must be divisible by 2 and HEAD_DIM must be divisible by 2.

(3) Memory-alignment friendly dimensions. For DDR burst efficiency and simpler packing of quantized tensors, we prefer embedding/head sizes that are powers of two or multiples of 16. This reduces edge-case handling in address generation, avoids irregular tail fragments in DMA reads, and improves practical alignment of streamed payloads.

Together, these constraints mean that scaling "tokens/embedding/heads" is not only a bandwidth question: invalid combinations can introduce partial tiles, inefficient bursts, or additional buffering and control complexity. Both verified configurations in Table I satisfy all constraints.

C. End-to-End Encoder Datapath (as in Fig. 1)

Figure 1 depicts the *full* BERT-style encoder datapath targeted by our system integration. Conceptually, the layer is organized into four consecutive regions:

(A) Self-Attention. The layer begins by reading the input activation tensor X (tokens $\times$ embedding) and projection weights from DDR. Streamed matrix multiplications form the per-head projections Q, K, and V. Attention scores are produced by multiplying Q with K^T on a per-head basis, normalized by softmax, and multiplied by V to produce context vectors. An output projection maps the concatenated context back to the embedding dimension.

(B) Self-Output (post-attention). The attention output is combined with the original input via a residual addition, then normalized. This stage is flow-control sensitive because it couples two operand streams (main path and residual path) that must remain token-aligned under backpressure.

(C) Intermediate Feed-Forward. The normalized tensor feeds a first FFN linear transformation followed by a non-linear activation (e.g., GELU in standard BERT). In a streaming system, this region can become bandwidth- and buffering-sensitive due to expanded intermediate tensors.

(D) Output Feed-Forward. A second FFN linear transformation returns to the embedding dimension, followed by residual addition and normalization to produce the final encoder-layer output.

In a resource-constrained prototype, intermediate tensors may either be forwarded directly as streams (preferred) or materialized in DDR between major boundaries (sometimes useful for buffering/debug). Our current validated milestone is region (A): a complete multi-head attention datapath verified at both reduced and scaled parameters. Regions (B)–(D) are partially integrated and are the current focus of bring-up due to tighter backpressure coupling and higher sensitivity to scaling.

D. NoC-Centric Streaming Architecture

The system instantiates:

- DDR-backed buffers mapped into a global address space.
- Versal NoC endpoints (NMUs) issuing memory-mapped read and write transactions.
- Read and write DMA engines converting memory-mapped AXI transactions into AXI-Stream packets.
- AXI-Stream compute blocks implementing matrix multiplication, softmax, requantization, and auxiliary operations across the encoder datapath.

This structure decouples transport from computation and allows each compute block to be developed and validated independently while relying on the NoC for scalable data movement.

E. Compute RTL Provenance

The RTL implementations of the core compute blocks (matrix multiplication engines, softmax, LayerNorm, GELU, and requantization units) were provided via an instructor-maintained repository used in the course infrastructure. Our work primarily focuses on system-level integration and validation: NoC endpoint connectivity, DMA configuration, address mapping, control/orchestration, and simulation-driven verification.

IV. Implementation

A. Self-Attention Datapath

The self-attention datapath streams input activations and weights from DDR through the NoC and DMAs to compute Q, K, and V projections. Attention scores are formed, normalized via softmax, and multiplied by V to produce context vectors, followed by an output projection. This pipeline is functionally verified end-to-end under both configurations in Table I, demonstrating correct operation as tokens, embedding dimension, and head count increase.

B. Numeric Representation and Requantization

Accumulator outputs are produced at 32-bit precision and requantized to 8-bit prior to write-back:

$$\text{Output} = \text{clip}\left((Input \times M) \gg E\right), \tag{5}$$

where M and E are fixed-point scaling parameters. This reduces DDR bandwidth requirements and aligns with the quantized streaming format used throughout the design.

C. AXI, DMA, and Flow Control

Read DMAs fetch tiles from DDR and emit AXI-Stream packets, while write DMAs commit AXI-Stream outputs back to memory. As tensor sizes increase, correct handling of AXI backpressure and FIFO sizing becomes critical. While bring-up configurations are robust and fast to iterate, larger configurations amplify sensitivity to stalls, buffering limits, and stream coupling, motivating a staged validation approach and adherence to the parameter constraints in Section III-B.

V. Verification Methodology

A. Functional Correctness Strategy

Verification focuses on:

1) Correct transport of data through DDR→NoC→DMA→AXI-Stream (ordering, completeness, and tiling).
2) Numerical correctness of multi-head self-attention outputs against software references (reduced-size deterministic tests first, then scaled tests).
3) Integrity of write-back streams at DMA boundaries when DDR model visibility is limited.

B. Simulation Constraints

Vivado xsim with Versal NoC and CIPS VIP is used for simulation. Due to limited DDR observability, SystemVerilog backdoor access is employed to inspect internal streams and boundary signals during bring-up. In practice, simulation time increases significantly as parameters scale; therefore, reduced configurations are used for rapid debugging, while larger configurations are used to validate scaling behavior and integration stability.

VI. Results and Current Status

At the time of writing:

- NoC-connected DDR→DMA→AXI-Stream transport is integrated in simulation.
- A complete multi-head self-attention datapath (Fig. 1, region A) is functionally verified end-to-end at both:
  - a reduced bring-up configuration ($T = 4, D = 64, H = 4$), and
  - a larger scaled configuration ($T = 32, D = 256, H = 8$).
- Downstream encoder sub-layers (self-output and feed-forward/output blocks; Fig. 1, regions B–D) are partially integrated and expose scaling and flow-control challenges, particularly around coupled residual paths and AXI backpressure.

Quantitative throughput and latency evaluation are left as future work once robust scaling is achieved across the full encoder datapath and the system is closed under the Vivado NoC compiler flow.

VII. Discussion

Our experience indicates that scaling transformer workloads on NoC-based FPGA platforms is primarily constrained by data movement and flow control rather than raw compute capability. Even with a working attention core, increasing tokens, embedding size, or number of heads places significant pressure on DDR bandwidth, DMA buffering, and AXI backpressure handling. Additionally, valid scaling must respect the dimension constraints introduced by multi-head partitioning, systolic tiling, and memory alignment (Section III-B). These effects motivate careful system-level design, disciplined parameter selection, and staged validation when targeting larger models.

VIII. Conclusion

This paper presented a NoC-centric mapping of a BERT-style encoder datapath onto AMD Versal. By integrating DDR, NoC, DMA, and AXI-Stream compute blocks, we demonstrated a functionally verified multi-head self-attention implementation under realistic simulation constraints, including successful operation at larger scaled parameters. The results highlight both the feasibility of NoC-based transformer integration and the practical challenges that arise when scaling beyond reduced configurations, including parameter compatibility constraints driven by tiling and alignment and flow-control sensitivity in residual/normalization and feed-forward stages.

References

[1] J. Devlin *et al.*, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019.

[2] A. Vaswani *et al.*, "Attention is all you need," in *Proc. NeurIPS*, 2017.

[3] AMD, "Programmable Network on Chip and Integrated Memory Controller (PG313)."

[4] AMD, "Versal ACAP Network on Chip and Integrated Memory Controller (UG1387)."

[5] ARM Ltd., "AMBA AXI and ACE Protocol Specification," 2013.

[6] H. T. Kung and C. E. Leiserson, "Systolic arrays (for VLSI)," 1978.