LLMCompass Simulation Report: Benchmarking LLaMA-2-7B on Xilinx Alveo U280 FPGA

Executive Summary

This report summarizes the work conducted to date on simulating LLaMA-2-7B inference using the LLMCompass framework on a Xilinx Alveo U280 FPGA. The baseline (unoptimized) simulation reveals key performance metrics, including a total latency of 3.67 seconds for a single forward pass, with 91% of operations being memory-bound. These insights highlight opportunities for hardware optimizations, such as tiling and pipelining, to improve throughput. Future steps include implementing these optimizations and validating physical hardware.

1. Introduction

1.1 Background

Large Language Models (LLMs) like LLaMA-2-7B require significant compute and memory resources for inference. LLaMA-2-7B is a 7-billion-parameter decoder-only transformer model with 32 layers, a hidden size of 4096, 32 attention heads, and a context length of 4096 tokens. It uses Grouped Query Attention (GQA) for efficiency and was trained on 2 trillion tokens.

The LLMCompass framework, developed at Princeton University, allows architects to evaluate custom hardware designs for LLM inference without RTL development. It features a performance model, mapper for optimal scheduling, and area/cost estimates. Validated on GPUs like NVIDIA A100 (4.1% error for LLM inference), it supports FPGA simulations via customizable hardware descriptions.

1.2 Objectives

- Simulate LLaMA-2-7B inference on Xilinx Alveo U280 FPGA.
- Identify latencies, bottlenecks, and utilization in an unoptimized setup.
- Provide a baseline for optimizations (e.g., tiling, buffer adjustments).

1.3 Hardware Overview: Xilinx Alveo U280

- Resources: 1,304K LUTs, 2,607K registers, 9,024 DSP slices, 75.9 Mb BRAM, 960 Mb URAM
- Memory: 8 GB HBM2 (460 GB/s), 32 GB DDR4 (38 GB/s)
- Interconnect: PCIe Gen3 x16 / Gen4 x8, QSFP28 100 Gbps
- Power: 200W typical
- Clock: 300 MHz simulated

2. Methodology

2.1 Setup

- Model: LLaMA-2-7B, sequence length 512, batch size 1
- Graph: Python-generated (Matmul, Softmax, LayerNorm, All-Reduce)
- Hardware: Custom U280 description with systolic arrays
- Simulation time: Minutes on commodity CPU

2.2 Execution
- Used LLMCompass simulator & mapper
- Outputs: per-op latency, FLOPs, JSON summary

3. Results

3.1 Performance Baseline
- Latency: 3.67s
- FLOPs: 17.57B
- Throughput: 0.005 TFLOPS
- Utilization: ~0.02 (memory-bound)

3.2 Bound Analysis
- Memory-bound ops: 91%
- Compute-bound ops: 9%

3.3 Operator Latencies
- Softmax: 0.112s per layer
- Matmul: 0.00007–0.001s

4. Analysis

Memory dominance suggests improving data locality (tiling) or buffers. Compute bottleneck in Softmax requires better vector units. FPGA-specific optimizations: pipeline and exploit HBM bandwidth.

5. Conclusion & Next Steps

- Memory-bound (91%), Softmax is compute hotspot.
- Next: mapper optimizations, tiling, reruns, and validation on U280.
Target: throughput >0.005 TFLOPS

Operations	Latency (s)	Compute Time (s)	Memory Time (s)	Bound	FLOPs
token_embedding	3.65e-05	0.0	3.65e-05	Memory	2K
ln_attn	3.65e-05	6e-07	3.65e-05	Memory	8M
qkv_proj	7.29e-05	1.55e-06	7.29e-05	Memory	16M
softmax_attn	0.112	0.112	0.00058	Compute	134M
attn_v	0.0012	2.48e-05	0.0012	Memory	268M