The Anatomy of an Efficient Blackwell GEMM

Antonio Moral Villarín

Table of contents

1 Abstract

2 Acknowledgements

3 List of Figures and Tables

To be auto-generated by Quarto.

4 Chapter 1 – Introduction

4.1 1.1 Motivation and Context: The Need for Hardware–Software Co-Design

4.2 1.2 Challenges in Efficient Compute for AI and Edge Applications

4.3 1.3 Objectives and Scope of the Thesis

4.4 1.4 Methodology Overview

4.5 1.5 Structure of the Thesis

5 Chapter 2 – Background and Related Work

5.1 2.1 Evolution of GPU Architectures: From Volta to Blackwell

5.2 2.2 Hardware–Software Co-Design: Principles and Applications

5.3 2.3 General Matrix-Matrix Multiplication (GEMM) in AI Workloads

5.4 2.4 Domain-Specific Languages (DSLs) for GPU Programming

5.5 2.5 Relevant Publications and Tools (NVIDIA Research, Citadel, JAX Scaling Book, etc.)

6 Chapter 3 – Architecture Comparison: Hopper vs Blackwell

6.1 3.1 Overview of Hopper Architecture

6.2 3.2 Overview of Blackwell Architecture

6.3 3.3 Key Innovations in Blackwell

6.3.1 3.3.1 Ultra Tensor Cores and New Precision Formats (FP8, FP4)

6.3.2 3.3.2 Transformer Engine and FP4 Micro Scaling

6.3.3 3.3.3 Multi-Die Chip Design and Interconnect (NVLink, NVSwitch)

6.3.4 3.3.4 Memory System: HBM3e, L2 Cache, and Shared Memory

6.4 3.4 Performance/Watt and Area Efficiency Considerations

6.5 3.5 Summary of Architectural Differences

7 Chapter 4 – Metrics for GPU Efficiency

7.1 4.1 Performance per Watt

7.2 4.2 Compute Throughput by Data Type