Special Project Report

侯奕安 b11202014

June 13, 2025

1 Introduction

This special project aims to implement the hardware inference accelerator for DNN proposed in [2]. This accelerator cuts activation and weight matrix in tiles and using a 16-lane mac datapath, thus has the advantage of parallel multiplication. Furthermore, this circuit design supports various quantization methods, including int8, int4, and int4-vsq. The per-vector quantization is based on concepts in [1], which serve the purpose of lowering quantization error by using two scale factors instead of one: coarse-grain scale for whole matrix and fine-grain scale for each vector.

2 Design Features

1. A and B buffer: tiling A and B matrix into shape $VS \times VL$ and $AD \times VS$ respectively.

2. MAC and accumulation collector: with 16 vector lanes to do parallel calucation. By **reusing A vector in 16 cycles and broadcasting B input to 16 lanes**, A input stationary and B input reuse is achieved. Also, **adder tree is used in MAC** to lower the area.

3. PPU: allow post-processing of the multilpication result. The scaling module supports **fp8 format** (E4M3, in particular) of coarse-grain scales for the two matrices. Also, expensive GeLU function is replaced by ReLU to increase space efficiency. Finally, approximate softmax module **uses 2 as base instead of e in order to avoid costly power calculation**, and can apply to one column at a time.

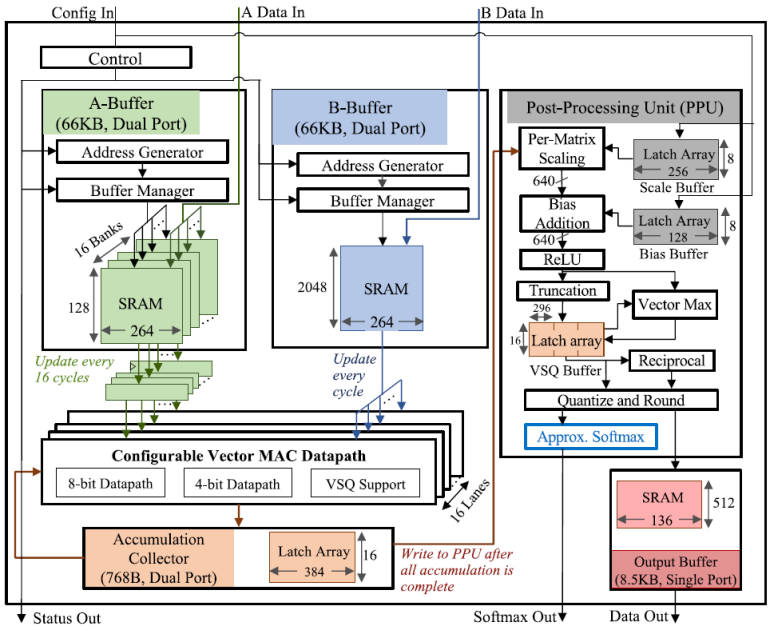

Figure 1: The architecture proposed in [2], which is mostly covered in this project with a little modification.

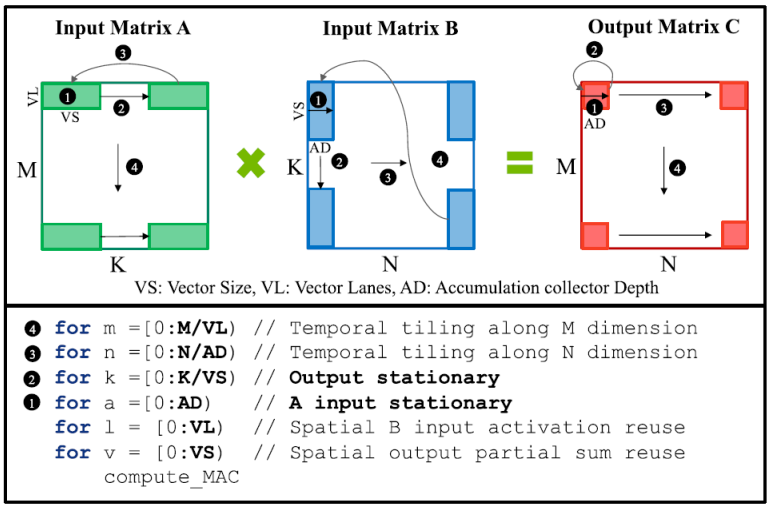

Figure 2: The tiling method to map matrix multiplication to hardware in [2]

3 Future Progress

Though not able to implement more features this time, there are a list of them I would really like to do to improve my design:

1. Further lowering area: I used a rather large divider in approximate softmax module. I want to replace it with a reciprocal input and multiplication instead.

2. Better pipeline: My current design wastes a lot of time to completely finsih one part of calculation to enter the next one. I want to design a better pipeline so that the stall time can be reduced.

References

[1] S. Dai et al. "VS-Quant: Per-vector Scaled Quantization for Accurate Low-Precision Neural Network Inference". In: *Proceedings of Machine Learning and Systems (MLSys)*. San Jose, CA, USA, Feb. 2021. URL: https://arxiv.org/abs/2102.04503.

[2] B. Keller et al. "A 95.6-TOPS/W Deep Learning Inference Accelerator With Per-Vector Scaled 4-bit Quantization in 5 nm". In: (2023). Member, IEEE or Senior/Fellow where applicable.