technische universität dortmund

computer science 12

Bachelor/Master Thesis

Efficient Data Format Conversion for Low-Bitwidth Vector Processing

Yun-Chih Chen
Prof. Dr. Jian-Jia Chen

Otto-Hahn Str. 16
Technische Universität Dortmund
Email: yunchih.chen@tu-dortmund.de
July 15, 2024

Mainstream highly-parallel accelerators, such as Nvidia's T4 and Google's TPU, are usually optimized for low bit-width inputs. Each computing unit require less circuitry and generate less heat compared to larger bit width. For instance, 8-bit multipliers save up to 91% of power and is 2.17 times faster than a 32-bit multiplier [2]. While TPUs are designed for matrix multiplication, we foresee the rise of vector processors with numerous low-bit, RISC-style ALUs that can support a broader range of parallel operations.

Traditionally found as DSPs in embedded systems specialized for signal processing, we envision these vector processors becoming essential co-processors in mobile SoCs or being integrated into CPUs like a built-in graphics unit. They could also be embedded in memory controllers for near-data computing, accelerating various tasks such as vector searches, exhaustive scans, and reduction operations like summation and finding minimum values[1].

However, as such a low-bitwidth co-processor collaborates with other wide-bitwidth compute units (e.g., CPU), data format conversion must be performed frequently to realize cache-efficient computation. This could require creating many copies of the same data in various formats. In addition, conversion is required in each iteration of mixed-precision (8-bit and 32-bit) linear algebra algorithms [3], which may be a bottleneck.

The objective of this thesis is to investigate several conversion procedures that trade off complexity and accuracy. You will study current rounding techniques in embedded DSP tailored for signal processing [5], floating-point arithmetic emulation schemes built for TPU [3], and low-bitwidth aggregation operations used in approximation query processing in databases [4].

You will write C++ program with x86 SIMD intrinsics to create prototypes for basic operations such as vector addition, matching, and reduction. You will use various rounding strategies, compare their complexity, adjust the bitwidth and vector length, and assess their algorithmic accuracy. Your research will set the groundwork for a general hardware-based rounding approach that is embedded in memory and provides efficient data layout for a future scalar-vector collaborative processing system.

[1]UPMEM exemplifies this approach, featuring native 8-bit hardware multipliers but relying on software-based shift-and-add multiplication for higher bit widths and emulating floating-point arithmetic [1]

Required Skills:

- Experiences in basic data structures
- Experiences in C++ programming

Acquired Skills after the thesis:

- Hands-on experience with x86 SIMD programming.
- Experiences in extending and generalizing existing codebase [6].
- Experiences on mixed-precision algorithms and rounding techniques.

References:

[1] Friesel, Birte, Marcel Lütke Dreimann and Olaf Spinczyk. "A Full-System Perspective on UPMEM Performance." Proceedings of the 1st Workshop on Disruptive Memory Systems (2023).

[2] Oliveira, Geraldo F., Juan Gómez-Luna, Saugata Ghose, Amirali Boroumand and Onur Mutlu. "Accelerating Neural Network Inference With Processing-in-DRAM: From the Edge to the Cloud." IEEE Micro 42 (2022): 25-38.

[3] Lin, Zejia, Aoyuan Sun, Xianwei Zhang and Yutong Lu. "MixPert: Optimizing Mixed-Precision Floating-Point Emulation on GPU Integer Tensor Cores.", Proceedings of the 25th ACM SIGPLAN/SIGBED International Conference on Languages, Compilers, and Tools for Embedded Systems (2024)

[4] Pirk, Holger, Stefan Manegold and Martin L. Kersten. "Waste not... Efficient co-processing of relational data." 2014 IEEE 30th International Conference on Data Engineering (2014): 508-519.

[5] `https://docs.amd.com/r/en-US/am004-versal-dsp-engine/Rounding`

[6] `https://github.com/megagonlabs/vecscan`