Accelerating applications with RISC-V Systolic Array Coprocessors

Piljić I., Mrković L.,
Kovač M., Kovač M.
University of Zagreb
Zagreb, Croatia
{igor.piljic, mate.kovac, luka.mrkovic,
mario.kovac}@fer.hr

Şarkışla A., Kurt C.,
Öztürk H., Seferbey S.
TÜBİTAK BİLGEM
Kocaeli, Turkey
{alp.sarkisla, can.kurt, hikmet.ozturk,
said.seferbey}@tubitak.gov.tr

Fell A., Ferrer R., Cervero T., Davis J.
Barcelona Supercomputing Center
Barcelona, Spain
{roger.ferrer, alexander.fell,
teresa.cervero, john.davis}@bsc.es

I. Introduction

Supercomputers targeting ExaFLOP performance require a highly efficient computational infrastructure. The MareNostrum Exascale Experimental Platform (MEEP) [1] is a digital laboratory for creating new hardware platforms and the associated software ecosystem. An example of a new accelerator demonstrated in MEEP is a self-hosted accelerator depicted in Fig. 1, the Accelerated Compute and Memory Engine (ACME). The memory engine of ACME includes a set of Memory Tiles (MT) responsible for optimizing memory operations to and from main memory (HBM). The compute engine of the ACME is formed by a Vector and Systolic Arrays in the VAS Tiles, in which the computational core is composed of a RISC-V scalar core (SC), one 16-laned vector processing unit (VPU), and two different systolic array (SA) accelerators. Each SA in a VAS Tile is designed for a specific application - either Bolt65 or Neutral Network (NN).

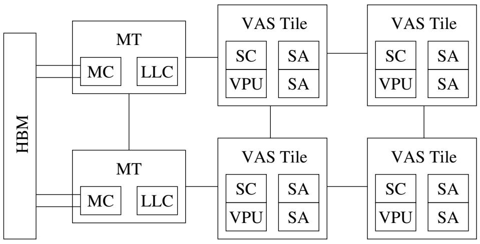

Figure 1. A block diagram about the key components of ACME

II. Bolt65 application and SA-HEVC

Bolt65 [2] is a performance-optimized HEVC hardware/software suite for Just-in-Time video processing developed by UniZG. It consists of an encoder, decoder, and transcoder based on the HEVC standard. Special focus in Bolt65 is set on the performance efficiency achieved by low-level optimizations and hardware-software co-design. Therefore, the MEEP platform, with ACME and custom SA accelerators offers an ideal opportunity for porting applications such as Bolt65. To fully exploit the benefits of the MEEP platform, we analyzed the application workflow and identified key kernels suitable for exporting to the SA accelerator, due to their computational complexity and interconnectedness: Discrete Cosine Transform (DCT), Quantization, Dequantization, and Inverse DCT (IDCT). Systolic array SA-HEVC implements these four kernels.

III. Neural Network and SA-NN

SA-NN, designed by TÜBİTAK, is a general-purpose NN accelerator with systolic array configuration to accelerate any pre-trained NN model. Fused multiply and accumulate (MAC) units, activation functions, and registers for input, weight, bias, and output are the main components of the SA-NN. The inputs in the scratchpad memory are fetched to SA-NN and processed in the fused MAC units to calculate the results. Then, the results are written back to the scratchpad memory. Bfloat16 format is chosen to increase the accelerator efficiency and minimize memory requirements without sacrificing the dynamic range. The SA-NN, which has 256 MAC units as standard configuration, is designed with a modular structure to be reconfigured with fewer or additional MAC units per accelerator depending on the performance, power, and area constraints.

IV. Compiler

To enable exploitation of custom SA accelerators from the application side, we introduced the MEEP Systolic Array Extension (MEEP SA). This is an extension of the RISC-V ISA intended to offer an instruction level interface to SA accelerator operation as envisioned in the context of the MEEP project. To that effect, MEEP SA contains instructions enabling setting of SA accelerator's operational parameters, loading and storing data into and from the scratchpad memory, and finally triggering the execution of SA accelerators.

V. Systolic Array Shell (SA-Shell)

The SA-Shell, developed by the UniZG, is a unified assemblage of auxiliary modules aimed at streamlining the development of custom SA accelerators and their integration into the VAS Tile.

To an SA accelerator, it offers a multi-level buffer for instructions received from the RISC-V core through the OVI [3], predefined and user-defined operational parameter registers, and an automated scratchpad access using an adjustable number of MicroEngines. We intentionally omitted an overseeing controller orchestrating SA-Shell operations to allow for greater flexibility in how the SA developers choose to utilize provided resources.

VI. Conclusion

This work demonstrates a full-stack implementation of a high-performance application on the MEEP platform, from the application design, compiler support to custom SA accelerator.

References

[1] A. Fell *et al.*, "The MareNostrum Experimental Exascale Platform (MEEP)," *Supercomput. Front. Innov*, vol. 8, no. 1, pp. 62–81, 2021.

[2] I. Piljić *et al.*, "Bolt65 – performance-optimized HEVC HW/SW suite for Just-in-Time video processing," in *MIPRO*, 2019, pp. 966–970.

[3] R. Espasa *et al.*, "AVISPADO - VPU Interface." [Online]. Available: https://github.com/semidynamics/OpenVectorInterface

The MEEP project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 946002. The JU receives support from the European Union's Horizon 2020 research and innovation program and Spain, Croatia, and Turkey.