Accelerating applications with RISC-V Systolic Array Coprocessors

MEEP | MareNostrum Experimental Exascale Platform

Piljić I., Mrković L., Kovač M., Kovač M.
Faculty of Electrical Engineering and Computing, University of Zagreb | Zagreb, Croatia
igor.piljic@fer.hr, luka.mrkovic@fer.hr, mate.kovac@fer.hr, mario.kovac@fer.hr

Sarkısla A., Kurt C., Öztürk H., Seferbey S.
TUBITAK BILGEM | Kocaeli, Turkey
alp.sarkisla@tubitak.gov.tr, can.kurt@tubitak.gov.tr, hikmet.ozturk@tubitak.gov.tr, said.seferbey@tubitak.gov.tr

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

INTRODUCTION

Supercomputers targeting ExaFLOP performance require a highly efficient computational infrastructure. MareNostrum Exascale Experimental Platform (MEEP) is a digital laboratory intended 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 responsible for performing memory operations by analysing memory access patterns which are then optimised by being aggregated and rearranged into current and future requests to the main memory (HBM). The compute engine of ACME consists of Vector and Systolic Arrays (VAS) Tiles, in which the computational core is composed of a RISC-V scalar core including its subsystems such as caches, and several accelerators: 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 Neural Network (NN).

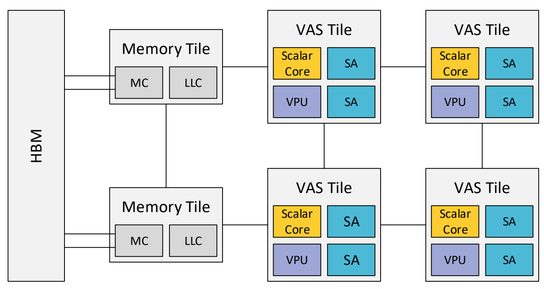

Figure 1: A block diagram of key components of ACME

SA-SHELL

Systolic Array Shell (SA-Shell) is a unified assemblage of auxiliary modules developed by UniZG aimed at streamlining the development of Systolic Array (SA) accelerators and integration into the VAS Tile. The design's primary focus is on providing an easy-to-use environment to SA developers producing accelerators for the MEEP project.

SA-Shell consists of:

- an OVI Controller which combines an instruction decoder with instructions and writeback queues
- Control/Status Registers (CSR) and Systolic Specific Registers (SSR)
- an adjustable number of instantiated MicroEngines processing memory operations.

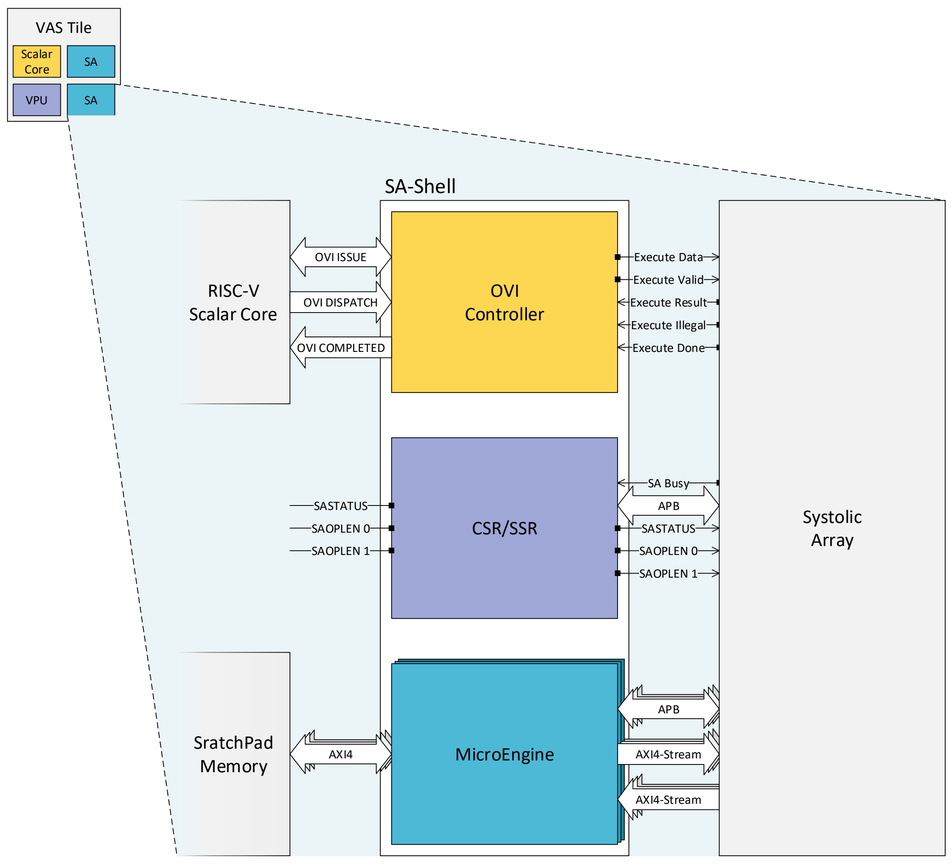

Figure 2: A block diagram of the SA-Shell in context of VAS Tile

We intentionally omitted a controller supervising SA-Shell's operation to allow for greater flexibility in the way SA developers choose to utilize provided resources. We suggest offloading all related tasks and logic from the SA accelerator's core by implementing a separate controller regulating the SA-Shell.

COMPILER

To enable the 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.

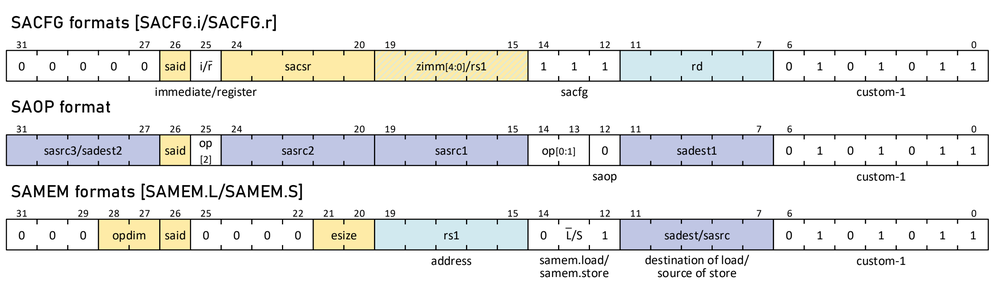

Figure 3: MEEP SA instruction formats

Bolt65 & SA-HEVC

Bolt65 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.

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, several key steps were taken:

- Application workflow analysis
- Identification of key kernels suitable for exporting to the SA accelerator. Due to their computational complexity and interconnectedness, four kernels were selected: Discrete Cosine Transform (DCT), Quantization, Dequantization, and Inverse DCT (IDCT).
- Implementation of application-specific SA accelerator SA-HEVC, consisting of selected kernels

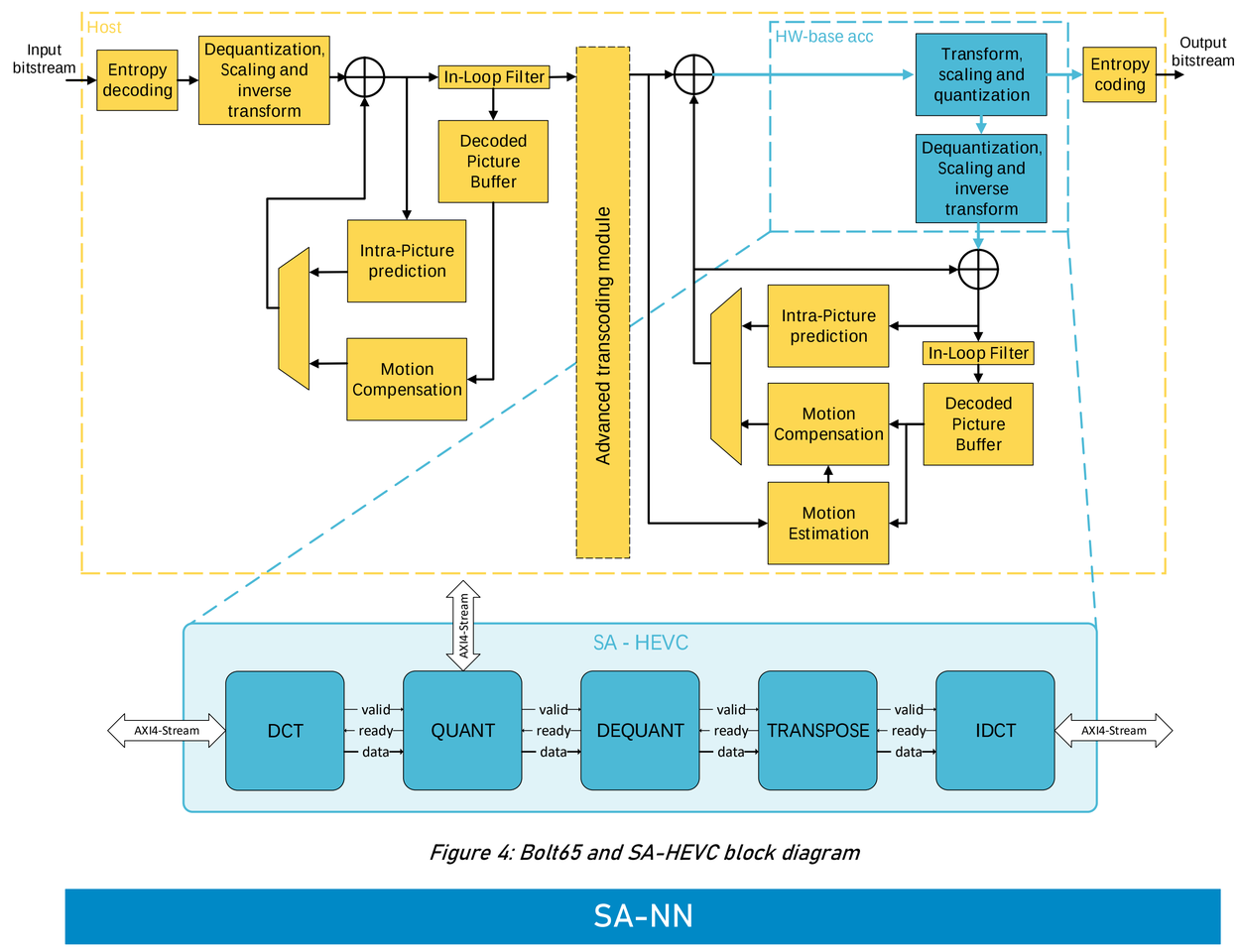

Figure 4: Bolt65 and SA-HEVC block diagram

SA-NN

SA-NN is a general-purpose reconfigurable neural network inference accelerator developed by TÜBİTAK. It utilizes systolic array architecture, and it can execute any pre-trained neural network. SA-NN consists of many primary elements (PEs) that each has its own multiply and accumulate (MAC) units. The number and structure of PEs can be reconfigured. As an example, the 8x8 configuration of the SA-NN is shown in Figure 5. The inputs in the scratchpad memory are fetched to SA-NN and processed in the MAC units to calculate the results. Small NN layers can be mapped as a whole, and large NNs can be mapped by reusing the same block multiple times per layer.

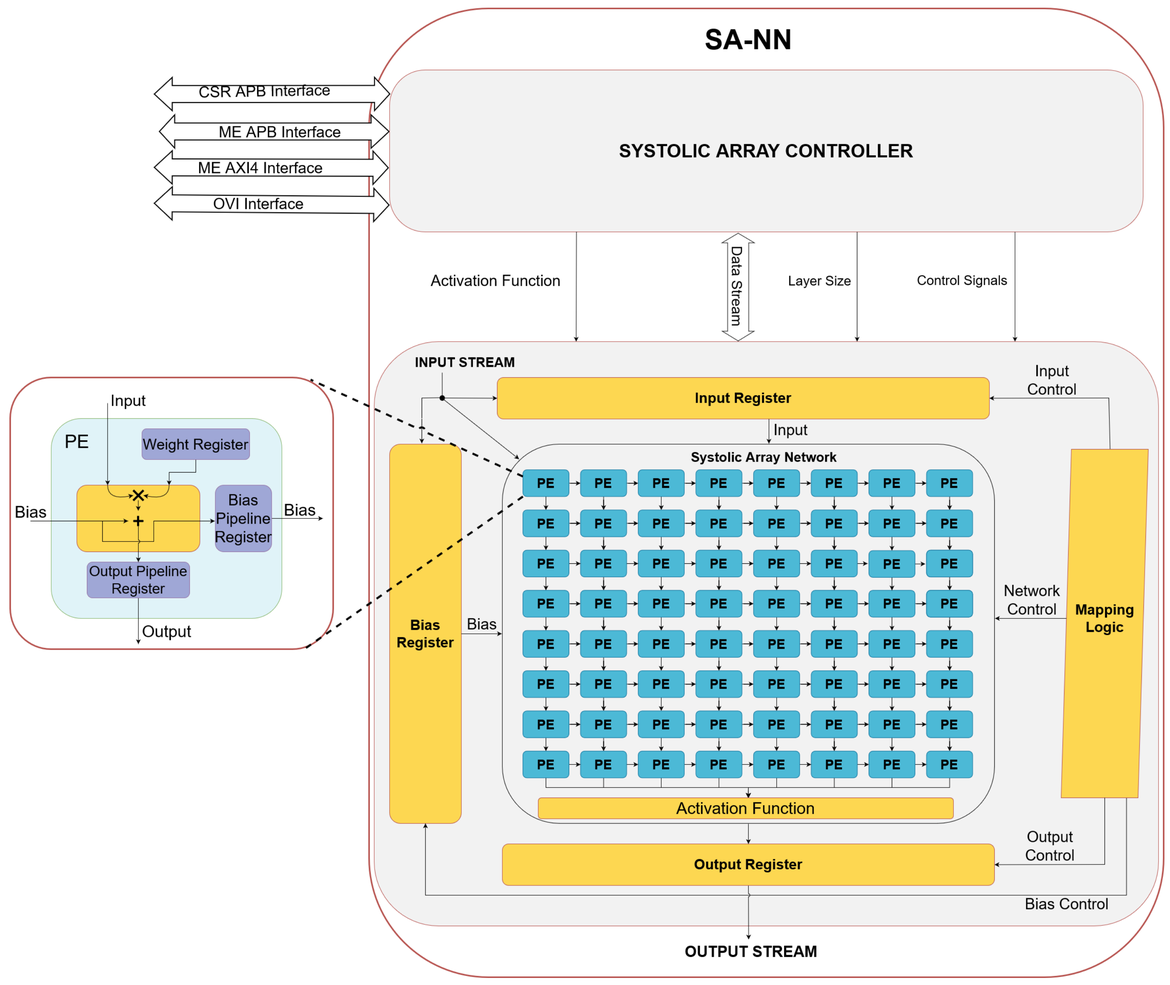

Figure 5: SA-NN block diagram

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 programme and Spain, Croatia, Turkey.