

Study and Implementation of CGRA architectures for edge applications

Candidate: Alessio Naclerio

The increasing deployment of data-intensive applications on edge devices requires hardware architectures capable of delivering high performance and energy efficiency under strict area and power constraints, while still retaining enough flexibility to adapt to evolving workloads. In this context, **Coarse-Grain Reconfigurable Arrays** (*CGRAs*) represent a promising design point between general-purpose processors and application-specific accelerators, since they combine spatial parallelism with post-fabrication reconfigurability. However, although recent CGRA research has increasingly focused on broadening code coverage toward dynamic-control and irregular workloads, the analysis carried out in this thesis shows that the vast majority of workloads effectively accelerated by state-of-the-art CGRAs still belongs to **Static Control Programs** (*SCoPs*), namely workloads characterized by compile-time predictable control flow and affine memory accesses.

SCoP kernels are particularly well suited to the intrinsic execution model of CGRAs, where an array of **Processing Elements** (*PEs*) can be exploited to expose the spatial parallelism available in such structured workloads. Focusing on this kernel class removes the need for CGRAs to support complex dynamic control flow inside the compute fabric, while also making them especially suitable for the **Decoupled Access-Execute** (*DAE*) paradigm, in which memory access and address generation are separated from data computation. Starting from this insight, this thesis investigates the design and evaluation of two complementary DAE CGRA architectures explicitly optimized for SCoP workloads, namely **Mage** and **Rogue**, in which address generation is offloaded to dedicated units, allowing the PEs to focus exclusively on arithmetic operations. This combination, together with *fully static kernel mapping*, enables the proposed CGRAs to achieve high performance on the considered SCoP workloads while also delivering strong area and energy efficiency, with competitive or superior results compared to state-of-the-art standard and DAE CGRAs. In addition, both Mage and Rogue have been integrated within the open-source X-HEEP RISC-V microcontroller platform exploiting two different interfaces.

As regards **Mage**, it is a DAE CGRA tailored for *General Matrix Multiplication* (*GeMM*) and GeMM-related SCoP workloads. Mage combines a dedicated **Address Generation Unit** (*AGU*) capable of generating multidimensional affine access patterns, with an internal multi-bank **Scratchpad Memory** (*SpM*), whose data layout must comply with the *Generalized Memory Partitioning* scheme. This organization enables conflict-free memory accesses and high-bandwidth data delivery to the **Processing Element Array** (*PEA*),

which exclusively performs arithmetic operations in either scalar or vector mode. Mage is implemented as a template-based architecture, enabling the exploration of different PEA, AGU, and SpM sizes, crossbar complexities, and PEA compositions, thus allowing several implementations with different area/performance trade-offs to be generated. Experimental results show that Mage achieves high throughput and area/energy efficiency on several SCoP kernels, reaching up to 7.8 GOPS, 13.8 GOPS/mm², and 345.6 MOPS/mW on `int8` matrix multiplication. Moreover, on an `int8` piecewise dilated 1D convolution, a key kernel in *Temporal Convolutional Networks*, Mage provides up to 50× speedup and 18× energy savings over execution on the CV32E40P RISC-V core with XCVLPULP extensions exploited through PULP `builtin` macros. These gains also translate to speedups of up to 12.9× and 9.7× on `int16` and `int8` 11-layer TCN pipelines, respectively.

The second CGRA is **Rogue**, which explores a lighter streaming-oriented DAE organization. Instead of relying on an internal memory subsystem, Rogue exploits a FIFO-based interface to directly connect to system DMAs and operate as a **DMA-coupled streaming CGRA**. In addition, Rogue implements a dedicated ISA that captures recurring computational patterns across different integer approximations of non-linear activation functions, while also supporting GeMM-style computation. Similarly to Mage, Rogue adopts a template-based design that allows the specification of its PEA size and composition, as well as the number of DMA channels it connects to and the complexity of this interconnection. The design-space exploration carried out for Rogue shows that, on integer activation functions, it achieves up to 98× speedup on Sigmoid and about 48× energy savings on GELU over software execution on the CV32E40P exploiting PULP extensions through compiler-generated code. In addition, on `int32` matrix multiplication, Rogue reaches up to 932 MOPS, 67.8 MOPS/mW, and 6522.6 MOPS/mm², while introducing only limited system-level area overhead in an X-HEEP-based SoC, up to 12.2%. Rogue has also been integrated into the **polHEEPo** SoC, an X-HEEP-based chip taped out in TSMC 65 nm technology at 100 MHz, where it will be exploited to address edge AI workload acceleration.

Overall, Mage and Rogue demonstrate two complementary CGRA specializations of the DAE paradigm for edge applications. Mage targets maximum throughput on GEMM-like kernels through conflict-free memory accesses and an internal SpM, whereas Rogue favors a compact streaming organization with low integration overhead. Together, they show that focusing CGRA design on SCoP workloads can unlock substantial performance, area-efficiency, and energy-efficiency gains while preserving a practical degree of flexibility for edge workloads.

Keywords: Coarse-Grain Reconfigurable Arrays, Edge AI, Hardware Accelerators, Decoupled Access-Execute, RISC-V, X-HEEP