



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

EXECUTIVE SUMMARY OF THE THESIS

qGamma: An Exploration Framework for the Mapping of Mixed-Precision Quantized DNN Models on Hardware Accelerators

LAUREA MAGISTRALE IN COMPUTER SCIENCE AND ENGINEERING - INGEGNERIA INFORMATICA

Author: LUIGI ALTAMURA

Advisor: PROF. CRISTINA SILVANO

Co-advisor: DR. LEANDRO FIORIN

Academic year: 2023-2024

1. Introduction

Nowadays, there is a huge interest in custom spatial accelerators for on-the-edge Artificial Intelligence applications. In particular, on-the-edge Deep Neural Network accelerators are typically based on spatial architectures composed of multiple processing elements interacting with the memory hierarchy through a network-on-chip. The energy-performance efficiency of these accelerators is given by an optimized mapping of the dataflow to the hardware resources and strategies to optimize data movement and reuse. In addition, mixed-precision quantization models can help reduce latency, energy, and memory consumption.

The goal of this thesis is to propose an exploration framework for mapping Deep Learning (DL) models to a mixed-precision quantized target architecture as a DNN accelerator.

To achieve this main goal, we developed qGamma, a flexible framework enabling the exploration and optimal mapping of mixed-precision DNNs on general on-the-edge accelerators supporting multiple compute fixed- and floating-point precisions. qGamma uses a redesigned domain-specific genetic algorithm-based method and qMaestro, an analytical per-

formance and energy model based on hardware synthesis results and CACTI-D, to explore the huge design space to find the mapping that minimizes latency, total energy, or energy-delay product. We evaluated the exploration results on various DNNs inference workloads showing the impact of using mixed-precision models when compared to fixed-precision implementations.

2. Background on DNN Accelerators

In this thesis, we consider a spatial architecture (SA) [1] accelerator. SAs are a class of accelerators that exploit direct communication between PEs to achieve high computational parallelism. The PEs typically consist of a scratchpad memory (**L1**) and arithmetic logic units (**ALUs**) responsible for performing MAC operations. To reduce the power and time required to access dynamic random access memory (**DRAM**), many DNN accelerators include a scratchpad buffer (**L2**) with sufficient capacity to stage data for all PEs. The shared buffer and the PEs are connected via a network-on-chip (**NoC**) [3]. The described architecture is shown in Figure 1.

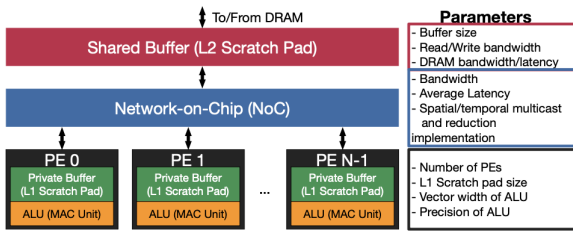


Figure 1: Abstract DNN accelerator architecture model [3].

Data reuse plays an important role in enhancing both latency and energy efficiency of DNN accelerators by minimizing the number of remote buffer accesses. Organizing data usage provides opportunities for data reuse either at successive time intervals on the same PE, known as **temporal reuse**, or across multiple PEs, known as **spatial reuse**. The choice of data handling strategies within the accelerator is also called dataflow [3]. The dataflow of an accelerator consists of two aspects: the scheduling of DNN computations over time, leveraging a wide spectrum of reuse, and the mapping of the DNN computation across PEs for parallelism.

Dataflows can be characterized by four key components, as outlined by [4]:

- **Spatial Map(size, offset)** α defines how the dimension α of a data structure is distributed across PEs. The *size* parameter specifies the number of indices in the dimension α that are assigned to each PE, while the *offset* parameter specifies the shift of the starting indices of α across successive PEs.
- **Temporal Map(size, offset)** α specifies how the dimension α is distributed across time steps within a PE. The *size* parameter denotes the number of indices in dimension α allocated to each PE, and the *offset* parameter describes the shift of the starting indices of α across successive time steps within a PE. The chunk of dimension indices mapped remains consistent across PEs within a given time step.
- **Data Movement order** determines the order in which spatial and temporal maps are arranged in the dataflow specification. This sequence dictates how data mappings to PEs change over time.
- **Cluster(size)** enables exploration of the

spatial distribution of more than one data dimension. This directive groups PEs of the *size* parameter or creates nested subclusters if there are more than one cluster directives.

3. Related Work

This section describes the open-source frameworks GAMMA and MAESTRO.

GAMMA [2] is a domain-specific genetic algorithm (GA) designed to efficiently explore the huge space of possible mappings. This framework constructs a flexible and comprehensive map space. To implement the GA, an encoding and decoding scheme was designed to translate between the genome and the mapping. For a one-level mapper, this scheme encodes the mapping into a genome with seven pairs of genes (7,2). Each pair contains a DNN layer dimension notation and its corresponding tile size. The first pair of genes specifies the parallelization dimension. For a two-level mapper, the encoding is similar, with the number of parallel L1 mappers being limited by the number of available PEs. The L1 mapper represents the inner loop, while the L2 mapper represents the outer loop, containing multiple instances of the L1 mapper.

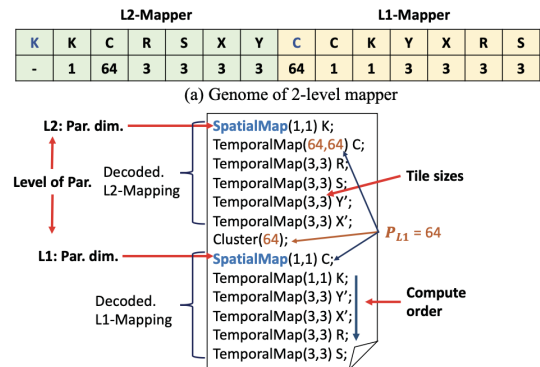


Figure 2: Decoded 2-level genome in MAESTRO's description [2].

Fig. 2 describes the decoding of a 2-level genome into a mapping, translating the genetic coding into a description for the cost model, MAESTRO, described hereafter. The order and tiling size of dimensions in the genome are mapped to the computation order and tiling size in the cost model. The parallelized dimension is associated to SpatialMap directive, and other dimensions are associated to TemporalMap directive. A mapper level in GAMMA is translated as a

cluster in MAESTRO, fully describing the tiling strategy, computation order, and parallelism dimensions. Multiple levels of parallelism are formulated by concatenating these clusters, with L1 and L2 mappers decoded into the lower and upper clusters, respectively.

The GA starts with a random population, performs crossover, mutation, and gene reordering to modify mappings, and selects the best candidates. The goal is to minimize hardware performance metrics such as latency, energy, or energy-delay product. The interactive environment uses MAESTRO to evaluate mappings, collect performance data, and compute fitness scores for optimization.

MAESTRO [4] is a comprehensive framework for cost-benefit analysis based on a systematic analysis of data reuse. MAESTRO takes into account DNN layer characteristics, hardware specifications, and mapping strategies as input, generating over 20 estimated statistics for efficiency evaluation. The framework supports different layer sizes and operations of state-of-the-art DNN models, ensuring versatility and applicability in different scenarios. MAESTRO proposes a data-centric mapping representation, whose directives are described in section 2, to facilitate concise and compiler-friendly descriptions of potential mappings, estimate the complex interaction between hardware components, mappings, and DNN layers, and evaluate data reuse within the scratchpad memory hierarchy.

4. Target Architecture

Our target architecture is a spatial architecture consisting of an array of PEs, each equipped with an L1 scratchpad memory and a shared L2 scratchpad global buffer. The architecture considers the possibility of having different levels of PEs clusterization. An example of the architecture described in shown in Fig. 3.

Fig. 4 shows a simplified block diagram of PE considered in this work. The model considers two decoupled pipelines for the floating-point and fixed-point compute engines. When both are available, this configuration allows optimizing each pipeline differently, increasing the system’s energy efficiency at the cost of some area overhead.

Our experiments consider the PE’s input and output ports of 32 bits. For the floating-point

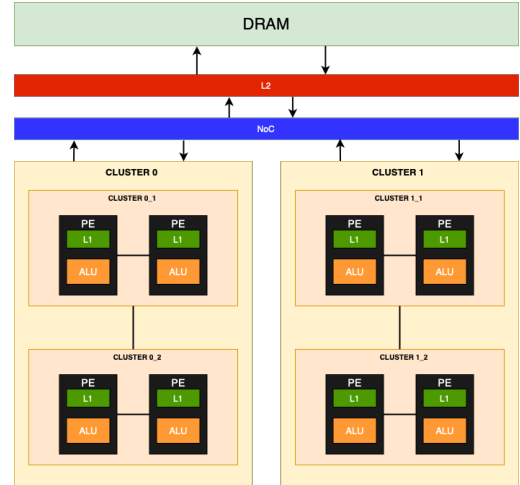


Figure 3: Target architecture.

pipeline, we model the possibility of implementing instructions using FP32, FP16, FP8, and FP4. For the fixed-point pipeline, we provide models for the INT32, INT16, INT8, INT4, INT2 data types. In the FP32 (INT32) mode, the engine performs a fused multiply-accumulate instruction (MAC). When using a smaller quantization Q , the 32-bit lane is partitioned in $P = 32/Q$ operands, and the engine performs a $\sum_P A_i \cdot B_i + C$ instruction.

The template components are automatically instantiated depending on the precision requirements of the use-cases targeted during the design-space exploration, together with the proper accumulator data width and output results packing.

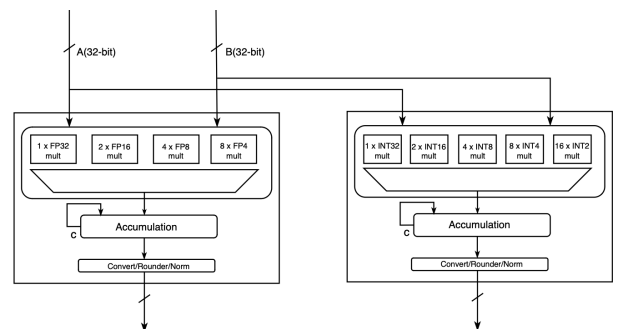


Figure 4: Simplified block diagram of the configurable PE.

5. Proposed Methodology

This section describes the re-design of the open-source frameworks GAMMA and MAESTRO to include mixed-precision quantization. We

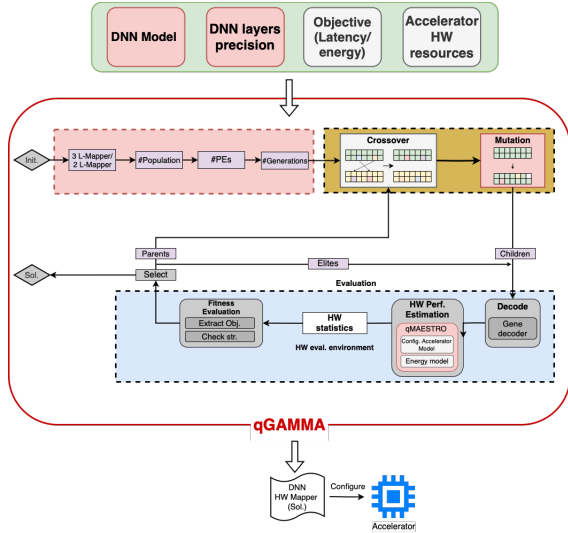


Figure 5: qGamma and qMaestro overview.

named the new versions of the tools qGamma (quantized GAMMA) and qMaestro (quantized MAESTRO).

5.1. High-level overview

Fig. 5 shows the new design of qGamma and qMaestro. The boxes in pink are the ones completely redesigned in this thesis. The new features of qGamma are as follows:

- Increase the genome size to enable the support of the reconfigurable PE based on the quantization.
- Create a new dataflow representation from output-centric to input-centric dataflow and allow different strides to be supported.
- Add heuristics to the GA to reduce the search space. Increasing the genome size significantly increases the search space, and introducing heuristics limits the search space by excluding inefficient dataflows.

The new features of qMaestro are:

- Integrate the configurable accelerator model into the framework, providing a configurable PE based on the precision described in the target architecture. As part of the integration of the new accelerator model, there was an expansion of the dataflows accepted by the model.
- Propose a new activity-based energy model with memory energy access values calculated at runtime based on the sizes found by the framework. The correct MAC energy value is selected based on the quantization

and the NoC is computed considering the quantization of the activations and filters.

5.2. qGamma

This section describes the methodology developed and the re-design made to GAMMA. It describes the work done on the genome to extend it to the new architecture model, the new input model in qGamma, and the introduction of the heuristics used to reduce the search space of the exploration.

5.2.1 Genome Extension

To enable mixed-precision quantization, we model a configurable PE with different FUs based on filters and activations precision, as described in Subsection 4. To enable dataflows suitable for our target architecture, we add another level of parallelism, passing from a 2-level mapper to a 3-level (L3) mapper. In the L3 mapper, the genome consists of 3 blocks representing dataflow for clusters, PEs within clusters, and FUs. The inner cluster size is fixed by quantization and represents the number of FUs inside a PE, while the outer cluster can be explored. An example of 3-level genome is shown in Fig. 6. The L3 mapper is needed for any quantization other than the standard one, which covers the FP32 and INT32 cases. In these cases, the algorithm uses an L2 mapper because the target architecture is the same as the one already implemented in MAESTRO. Increasing the dimension of the genome means even increasing the search space for exploration. For this reason, when the L3 mapper is instantiated, the population is doubled concerning the L2 mapper. An early stop mechanism is implemented to reduce execution time if there is no improvement over several generations.

5.2.2 Dataflow Description

Extending the genome and implementing an L3 mapper isn't enough to represent the target architecture. As described, the innermost level of the genome represents the dataflow between FUs, where parallelization in the K dimension must be excluded, as merging partial sums of different output feature maps would yield incorrect results based on the hardware PE model described in section 4. In addition,

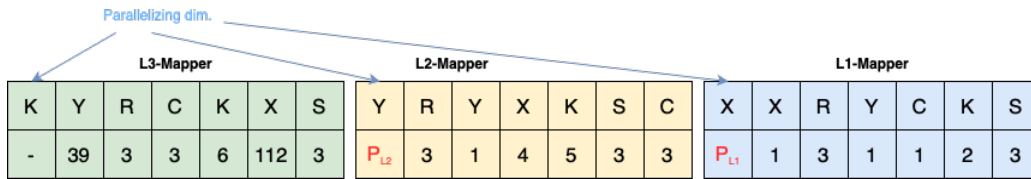


Figure 6: Example of 3-level genome.

we switched from an output-centric to an input-centric dataflow representation to better reflect the new architecture and extended support for different strides in the input model file.

5.2.3 GA Heuristics

Increasing the genome size significantly expands the search space, so we introduce heuristics to exclude inefficient dataflows. Since accelerators typically have a power of two PEs, we limited the number of PEs per cluster to powers of two to ensure full utilization.

The L3 mapper parallelizes across three dimensions, with larger dimensions providing better distribution across PEs. For the first level, the three largest dimensions are selected for parallelization; for the second level, the two largest are chosen based on the dataflow of the previous level. At the third level, no heuristic is applied because the space is already reduced by excluding the K dimension to follow the architectural model.

In addition, during genome generation, the K and C dimensions are mapped to powers of two, as they often follow this structure in most layers. This ensures an even distribution of these dimensions between clusters and PEs.

When comparing the exploration time of the baseline without heuristics with our implementation with heuristics in the case of ResNetV1, we observed an exploration time reduction of 19%. Excluding the early termination condition, the reduction is about 36%.

5.3. qMaestro

This section describes the methodology developed and the re-design made to MAESTRO. In particular, it describes the new architecture supported by qMaestro and the new energy model used.

To implement mixed-precision quantization we reflected the changes made in qGamma to qMae-

stro. We introduced precision for each layer, defining the quantization levels for activations and filters. Have the possibility to change the precision in the same neural network means change the accelerator configuration at runtime. Therefore for each layer the accelerator is configured based on the target architecture. Regarding the dataflow, MAESTRO already implements three-level clustering scheme. In this scheme, the dataflow is divided into three parts: the first part represents the dataflow for clusters, the second part represents the dataflow between PEs within clusters, and the last part represents the dataflow of the FUs of the configurable PE model. As part of the integration of the new accelerator model, there was an expansion of the dataflows accepted by the model, in particular for the input-centric representation now supported in qGamma, and we added support for different strides within the architecture.

qMaestro also implements a new activity-based energy model that takes into account memory access energy, MAC energy, and NoC energy.

The memory access energy for L1 and L2 memory is calculated based on the memory size provided by the framework at runtime. Energy for the several memory elements were generated with CACTI-D [5], scaling the obtained results from a 32-nm to a 22-nm technology.

The MACs energy and are also calculated taking into account the precision for every layer.

The NoC energy is also calculate at runtime based on the quantization of the activations and filters.

6. Experimental results

Due to space limitations, this section presents the experimental results for ResNet18 evaluated with INT8 and mixed-precision configurations. More experiments are reported in the thesis on several CNNs.

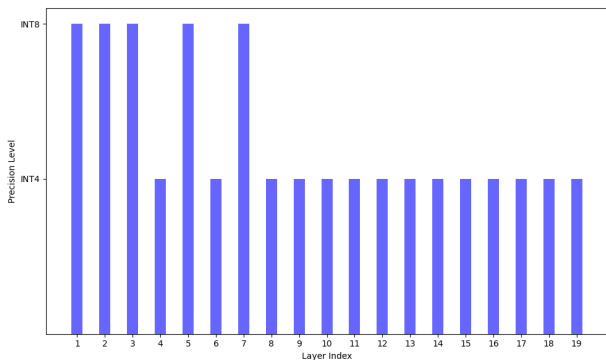


Figure 7: Quantization per layer in ResNet18 mixed-precision configuration.

6.1. Experimental Setup

For running the experiments it is used a MacBook Pro 2021 with 16GB of RAM and M1Pro Chip. In every experiment, we used qGamma and qMaestro to find and evaluate the best mapping for the CNNs. To simulate an on-the-edge DNN accelerator, we opted to run the simulation on an accelerator with 16 PEs, 512B of L1 memory maximum capacity, and 512KB of L2 memory maximum capacity. We used qGamma for two different optimizations. The first is a per-layer optimization, in which the clusterization is explored for each layer and can vary based on the layer characteristics. In the second, we consider a clusterization that fits all layers since the clusterization is a hardware organization. For every quantization, we ran qGamma for 1000 generations with a population of 150 individuals for the INT32 and FP32 quantizations that implement an L2 mapper, and with a population of 300 individuals for the other quantizations that implement an L3 mapper, since this significantly increases the search space exploration.

6.2. Results on ResNet18 optimized by latency

This experiment analyzes a ResNet18 in a mixed-precision configuration of INT8 and INT4 layers taken from [6] and compares it to an INT8 single-quantization configuration with similar accuracy. Both configurations are optimized for latency. In this experiment, we consider the clustering fixed for all layers. Fig. 7 shows in detail the precision of each layer in the case of

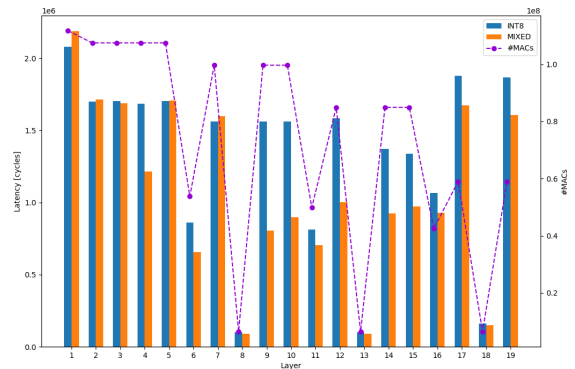


Figure 8: Latency results for ResNet18 INT8 version compared to mixed-precision version. Number of MACs per-layer also reported.

mixed-precision quantization. Fig. 8 shows the latency per layer. The Y-axis on the left shows the latency expressed in cycles, while the Y-axis on the right shows the number of MACs for each layer. The Figure shows how latency is affected by MACs and quantizations. In the layers where the quantization is INT4, latency is reduced than the corresponding INT8 quantization. The slight difference in latency results for INT8 between runs on the same layer is due to the nature of the genetic algorithm, which can for some layers converge on different suboptimal solutions, resulting in performance differences. The improvement ratio from INT8 single-quantization to mixed-precision quantization is of 1.20x.

7. Conclusions

This thesis presents two new frameworks, qGamma and qMaestro, that find and evaluate optimal mappings for mixed-precision quantization DNNs on hardware accelerators. The effectiveness of these frameworks has been proven by the experimental results presented in this thesis, with ResNet18 showing a 1.20x improvement in latency using mixed-precision quantization with respect to the higher single-quantization configuration. Future work may include extending the available architecture to digital in-memory computing accelerators, including these frameworks in network architecture searchers, and extending qGamma to a framework that jointly optimizes mixed-precision quantization mapping and hardware accelerators.

References

- [1] Yu-Hsin Chen, Joel Emer, and Vivienne Sze. Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. In *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, pages 367–379, 2016.
- [2] Sheng-Chun Kao and Tushar Krishna. Gamma: Automating the hw mapping of dnn models on accelerators via genetic algorithm. In *2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pages 1–9, 2020.
- [3] Hyoukjun Kwon, Prasanth Chatarasi, Michael Pellauer, Angshuman Parashar, Vivek Sarkar, and Tushar Krishna. Understanding reuse, performance, and hardware cost of dnn dataflow: A data-centric approach. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture, MICRO '52*, page 754–768, New York, NY, USA, 2019. Association for Computing Machinery.
- [4] Hyoukjun Kwon, Prasanth Chatarasi, Vivek Sarkar, Tushar Krishna, Michael Pellauer, and Angshuman Parashar. Maestro: A data-centric approach to understand reuse, performance, and hardware cost of dnn mappings. *IEEE Micro*, 40(3):20–29, 2020.
- [5] Shyamkumar Thoziyoor, Jung Ho Ahn, Matteo Monchiero, Jay B. Brockman, and Norman P. Jouppi. A Comprehensive Memory Modeling Tool and Its Application to the Design and Analysis of Future Memory Hierarchies. In *2008 International Symposium on Computer Architecture*, pages 51–62, 2008.
- [6] Zhewei Yao, Zhen Dong, Zhangcheng Zheng, Amir Gholami, Jiali Yu, Eric Tan, Leyuan Wang, Qijing Huang, Yida Wang, Michael W. Mahoney, and Kurt Keutzer. Hawqv3: Dyadic neural network quantization, 2021.