Post-doc and Research Engineer positions in TIMA Laboratory SLS Team

October 13, 2023

Introduction

The TIMA Laboratory (website) is an academic research entity situated in Grenoble city, in France. The research topics of TIMA cover the specification, design, verification, test, CAD tools and design methods for integrated systems, from analog and digital components on one end of the spectrum, to multiprocessor Systems-on-Chip together with their basic operating system on the other end.

The SLS team (System Level Synthesis - website) focuses on highly efficient architectures for general purpose computing or AI-dedicated algorithms, and system-level modeling and design methodology : specification, simulation and verification of hardware/software systems on chip; design exploration and synthesis of hardware.

This document presents Post-doc and/or Research Engineer positions that are available in the SLS team, linked to acceleration of AI (Artificial Intelligence) on FPGA and to more generic hardware acceleration on FPGA through PCI-Express.

About TIMA: TIMA is located within historical building of the Grenoble Polytechnic Institute, just near the city train station. It is fully accessible for people with disabilities.

TIMA is a multinational team, with members and interns from all over the world. The Laboratory and its parent institutions ensure that everyone is treated equally, respectfully, regardless of gender, religion, disabilities, etc.

Requirements: New applicants should send a CV and an overview of their work experience and full course-work and grades, by email and preferably in PDF format, to the contacts below. Please present clear, exhaustive and verifiable curriculum, as there is a standard inquiry before taking position (especially for non-European applicants).

Contacts:

Frédéric Pétrot, frederic.petrot@univ-grenoble-alpes.fr
Adrien Prost-Boucle, adrien.prost-boucle@univ-grenoble-alpes.fr

1 Positions : Hardware Acceleration of IA

1.1 Framework for benchmarking quantized neural networks on FPGA

Context: Nowadays, research in neural network efficiency involves heterogeneous quantization, custom number representation, sparse data access, etc. In order to demonstrate efficiency improvements in efficient hardware implementation of neural networks, fully operational FPGA prototyping is necessary to exhibit convincing results. Demonstration fully functional hardware prototypes is a heavy challenge that requires a lot of development.

We have research-only experiment tools to model quantized neural networks for acceleration on FPGA. This framework was originally built to construct highly efficient implementations for ternary quantization, then support for other types of quantization has been added (binary, logarithmic, generic integers). It is composed of a software tool (C/C++), and of a set of manually-written RTL components for execution of layers of neural networks.

The software tool is in charge of the following tasks:

- modelling the networks with all necessary low-level hardware implementation details,
- generating the necessary RTL files and configuration that implements the corresponding hardware pipelines,
- and configuring and controlling the hardware accelerator to execute accelerated inference tasks.

The hardware layers include the following:

- convolution layers (for neuron layers and pooling layers),
- activation layers for ternary numbers and usual integer ReLU,
- fork and concat layers to execute series of layers in parallel,
- etc.

The framework is not (yet) open-source. The goal is to release a version that is solid enough for public experiments.

Objectives: The capabilities of the framework will be extended to modelling all low-level hardware implementation details. The goal is to handle complex networks, to represent the equally complex operations (common arithmetic and custom operations), to enforce a global coherence and optimization of all hardware constraints throughout the execution pipelines, and to quickly report estimates about resource usage, latency, throughput and power consumption.

- extend support to bare-metal execution of inference tasks (Zynq target),
- extend support to remote accelerators connected through Ethernet,
- support more exotic or custom number representations, custom operations and activation functions,
- support user-provided custom layer implementations,
- extend ternary weight compression to potentially lossy compression schemes,
- maturation of handling of hard-coded network parameters,
- help other researchers and PhD students with proof-of-concept support for heterogeneous quantizations,
- extend implementations to support near-FPGA HBM memories,
- support some form of time multiplexing of layers or series of layers,
- etc.

Another potential direction is to study if and how the N2D2 toolkit can be used as front-end to model and train the networks.

Duration : 18 to 24 months

Requirements:

- Experience with AI, neural network architectures and tools
- Experience with Python scripting