Semester Thesis (2P) or Master Thesis

Approximate Matrix Multiplication based Hardware Accelerator to achieve the next 10x in Energy Efficiency: Full System Integration

Advisors: Jannis Schönleber, janniss@ethz.ch
Lukas Cavigelli, lukas.cavigelli@huawei.com
Renzo Andri, renzo.andri@huawei.com

Supervisor: Prof. Luca Benini, benini@iis.ee.ethz.ch

1. Introduction

The continued growth in DNN model parameter count, application domains and general adoption led to an explosion of the needed computing power and energy. Especially the energy needs have become large enough to be economically unviable or extremely difficult to cool down. That led to a push for more energy-efficient solutions. Energy efficient accelerator solutions have a long tradition in IIS, with a multitude of proven accelerators published in the past. Standard accelerator architectures try to increase throughput via higher memory bandwidth, improved memory hierarchy or reduced precision (FP16, INT8, INT4). The approach of the accelerator used in the project is a different one. It uses an approximate matrix multiplication (AMM) algorithm called MADDness, which replaces the matrix multiplication with a lookup into a look-up-table (LUT) and an addition. That can significantly reduce the overall computing and energy needs.

2. Project Details

The MADDness algorithm is split into two parts. We have an encoding part, which translates the input matrix A into the addresses of the LUT. After the translation follows a decoding part which adds the corresponding LUT values together to calculate the approximate output of the matrix multiplication. MADDness is then integrated into deep neural networks. The most common seen layers in DNNs are convolutional layers and linear layers, both can be replaced by MADDness. The current RTL implementation is fully post-simulation tested and includes both the encoder and decoder unit. Additionally, a post-layout simulation-based energy estimation has been done. The accelerator is not yet integrated into a full system.

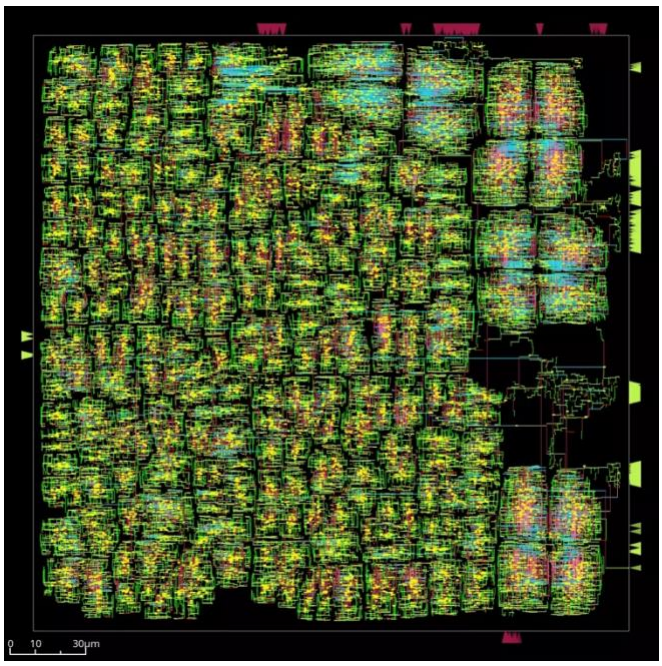

Figure 1: Clock layout of the MADDness accelerator using ASAP7 technology

Energy estimates with the current implementation using GF 22nm FDX technology suggest an energy efficiency of up to 32 TMACs/W compared to a state-of-the-art datacenter NVIDIA A100 (TSMC 7nm FinFET) at around 0.7 TMACs/W (FP16).

In this project, we would like to integrate the accelerator into a full system. The aim at the end would be to be able to have a tape-out ready full system which includes the MADDness accelerator. A full system includes a suitable memory hierarchy to support the bandwidth needs. We are envisioning an integration into one of the existing PULP systems (for example: PULP clusters or ARA). The evaluation of which system suiting the accelerator the best and defining the final architecture is part of the thesis.

More information can be found here:

- Code: https://github.com/joennlae/halutmatmul
- Reference Paper: https://arxiv.org/abs/2106.10860
- HN discussion: https://news.ycombinator.com/item?id=28375096
- and please do not hesitate to reach out to me: janniss@ethz.ch

3. Project Plan

1. **Acquire background knowledge & familiarize with the project**
    a. Read up on the MADDness algorithm and product quantization methods
    b. Familiarize yourself with the current state of the project
    c. Familiarize with the IIS compute environment
2. **Setup the project & rerun RTL simulations**
    a. Setup the project with all its dependencies
    b. Try to rerun the current RTL simulations
3. **Evaluate suitable systems to integrate and refine an architecture**
    a. Define bandwidth needs and brainstorm suitable memory hierarchies
    b. Spreadsheet based evaluation of the different target systems like PULP clusters, ARA etc. this includes exploring different configurations and estimated size of the chip
    c. Decide on a final architecture that we will pursue for the remainder of the project
4. **Integrate the accelerator into the defined architecture**
    a. Implementing the integration in SystemVerilog & add testbenches
    b. Replace the currently used standard cell memories with compiled memories
5. **Setup the design flow**
    a. Setup and integrate the (most likely tsmc65) design flow into the project
6. **Synthesize + Place-and-Route & make design tape-out ready**
    a. Synthesize + Place-and-Route the design
    b. Get a working post-layout simulation

c. Place macros & power routing, IR drop checks

d. The goal is to have everything ready for a design review

   i. http://eda.ee.ethz.ch/index.php?title=Design_review (ETH domain)

7. **Project finalization**

   a. Prepare final report

   b. Prepare project presentation

   c. Clean up code

4. Project Organization

4.1. Weekly Meeting & Report

The student and advisor meet weekly to discuss the project status – what has been done, whether there have been or we expect any problems or blockers, what are the current results, and how to proceed. It is advisable for the student to reflect on this and put together 2-3 slides for this every week ahead of the meeting.

After the meetings, the student shares a short summary of the status update and planned next steps by email with everyone involved in the project.

4.2. Project Plan

This project plan serves as a guideline to be reasonably specific on the intended steps and the expectations on the progress. As new insights and results are gathered during project execution, the plan can and should be adjusted flexibly to follow the most promising path.

4.3. Project Presentation & Hand-In

The student is responsible to ensure the requirements of their degree program are met. For ETH Zurich's EE department, the requirements are to hand in a report before the due date (by email to the supervisor and advisor in PDF form). Additionally, any developed code should be submitted either as a container file or within an accessible code repository.

Besides the report and code, there will be a project end presentation (e.g., 20 min. talk and 5 min. discussion) – either within the supervisor's research group or department and adjusted to their requirements. The exact date will be determined towards the end of the work within a few weeks before or after the deadline.

4.4. Intellectual Property

The product of the work of the student remains their property. The student permits and provides access to all relevant data for the supervision and evaluation of the work to the supervisor and advisor's institutions. Code must be shared under a permissive license. The student is required to sign a license agreement at Frank's office.