

## Master Thesis

# Approximate Matrix Multiplication based Hardware Accelerator to achieve the next 10x in Energy Efficiency: Training Strategy & Algorithmic optimizations

**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. Fully tested drop-in PyTorch layers have already been developed and used. Currently, only a single layer replacement analysis has been done rigorously. So far the layers have only been replaced with the MADDness algorithm and the network has not been

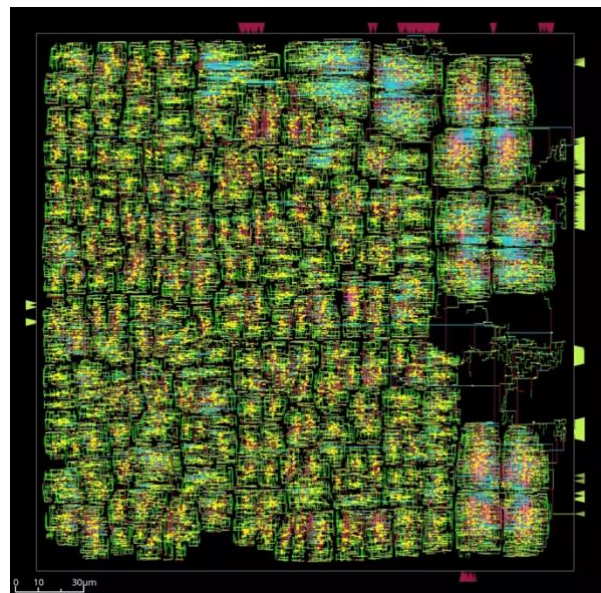


Figure 1: Clock layout of the MADDness accelerator using ASAP7 technology

retrained with the corresponding new outputs of the layers.

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 investigate if we can improve our accelerator's accuracy by implementing a retraining strategy and framework. The goal would be to be able to replace multiple layers of a DNN without a significant drop in accuracy. A new realm of possible inter-layer optimization can then be analyzed afterwards. For example: Only calculating the needed dimensions for the next MADDness layer or including the activation layer into the MADDness algorithm.

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](mailto:janniss@ethz.ch)

### 3. Project Plan

- 1. Acquire background knowledge & familiarize with the project** **3 weeks**
  - 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 single layer analysis** **2 weeks**
  - a. Setup the project and rerun a single layer analysis (for example for ResNet-50)
  - b. Update the single layer analysis with a larger than previously used LeViT model
- 3. Set up and evaluate a first retraining pipeline** **8 weeks**
  - a. Using the simple method of replacing one layer with MADDness and then retrain that layer. Afterwards freezing it the previous layer for the training of the following layers.
  - b. Evaluate and optimize the pipeline including a detailed analysis of the accuracy development for the ResNet-50, LeViT and DS-CNN networks
  - c. Include the developed framework into the already developed learning framework
- 4. Extend the MADDness algorithm with intra-layer optimizations** **10 weeks**
  - a. Include the activation function into the MADDness algorithm
  - b. Can we optimize memory bandwidth and/or compute by only computing the dimensions needed for the following MADDness layer.
  - c. Is the encoding function that we are using the most accurate? Can we improve it?

## **5. Project finalization**

**3 weeks**

- 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.