# DAC Young Fellows

DESIGN AUTOMATION CONFERENCE 58
FROM CHIPS TO SYSTEMS — LEARN TODAY, CREATE TOMORROW

# Accelerating a Visual Wake Word Application on a CGRA

Melina Soysal, Tulika Mitra, Technical University of Munich / National University of Singapore

## 1 Introduction and Motivation

Traditional Machine Learning (ML)

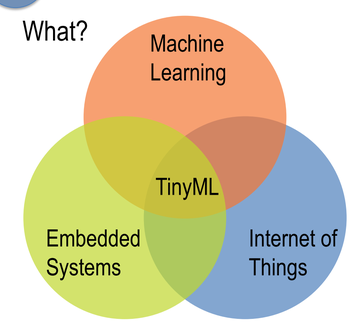
- computationally expensive
- requires a lot of memory
- high energy consumption
- For widely applicable ML, it is necessary to make it lightweight and cheap

Benefits of Microcontrollers (µCs)

- cheap
- prevalent
- low-power
- Combining the benefits of a µC with the power of ML will enable intelligent edge devices and eliminate the necessity for raw data transmission
- Further accelerate this process through task-specific hardware

## 2 Tiny Machine Learning (TinyML)

What?

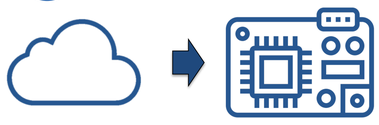



Why?

- reduce energy consumption
- reduce latency
- increase security and privacy

How?

- paradigm shift for ML inference



compute centric in the cloud → data centric on task-specific µCs

Enabled through

- Task-specific HW-acceleration
- Lightweight ML (reducing redundancy & sparsity)

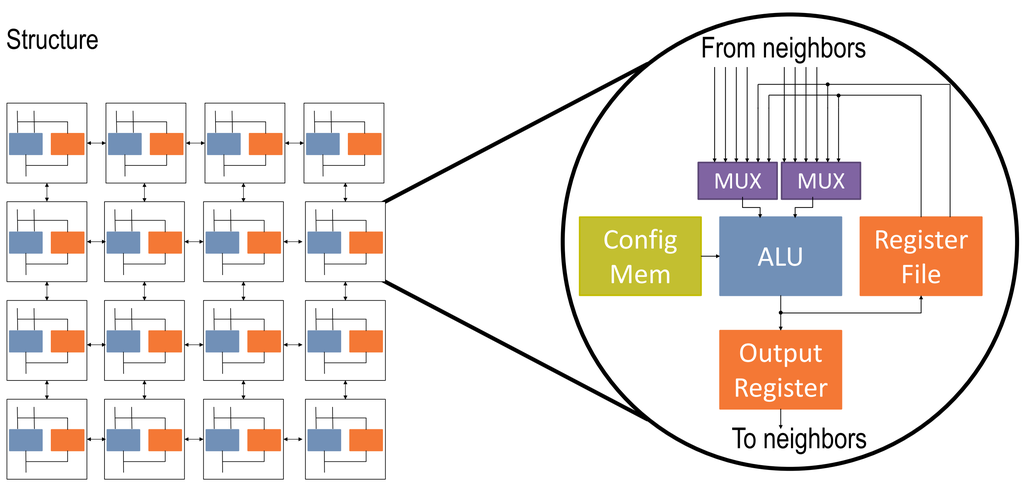
TinyML could enable the broad and cost-effective application of machine learning

## 3 Coarse-Grained Reconfigurable Array (CGRA)

- reconfigurable system on word level
- trade-off between performance & flexibility
- well suited for acceleration of loop structures

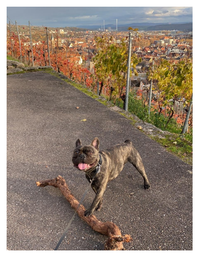
interesting for ML acceleration as computational load often comes from convolution layers which can be represented as loops

Structure




## 4 Visual Wake Word (VWW) application

Idea:



NO person detected

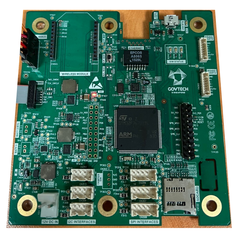
person detected

Why Wake Word?

- based on the idea of Wake Words (e.g.: “Hey Siri”, “Hey Google”, etc.)
- small device detects wake word
- larger, more energy consuming device gets “waken” up

## 5 Experimental Setup

Current Setup:



**DECADA Embedded**

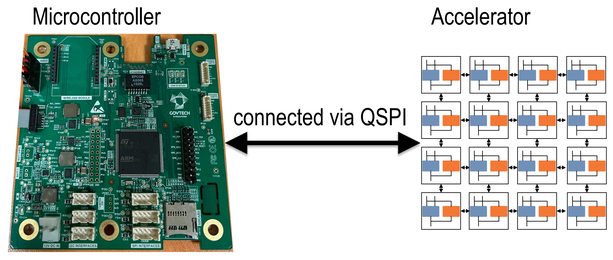
MCU: STM32F767ZI based on ARM Cortex M7 CPU

Flash: 2 MB

SRAM: 512 kB

complete application runs on DECADA Embedded

Future Setup:

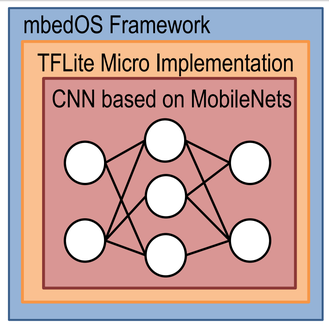



Convolution layers get accelerated on the Hycube[1] CGRA

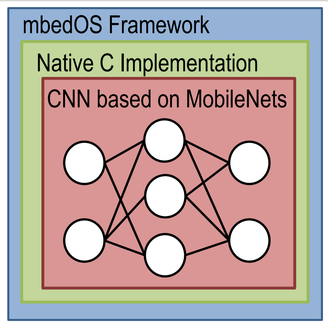
Currently the ML Algorithm is only implemented on the µC. In the future the the convolution loops will be accelerated on a CGRA.

## 6 ML Implementation

- Person detection is performed by a CNN based on MobileNets[2]
- Depth-wise separable convolutions, GEMM and quantization enable a more lightweight model
- Two implementations:
  1. using the TFLite Micro Framework
  2. using only a Native C implementation

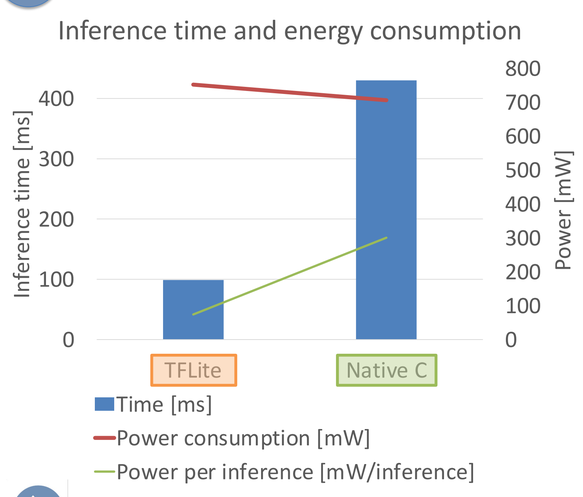


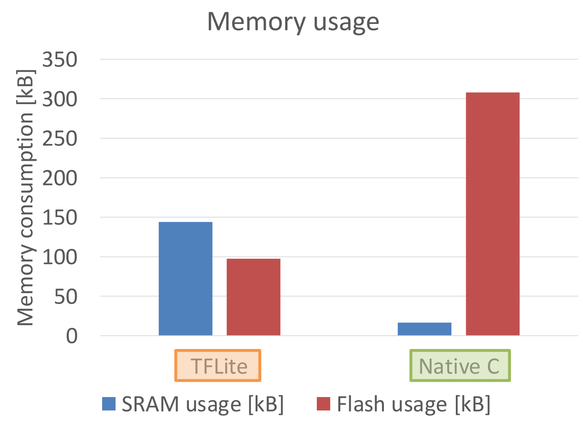

Implementation 1




Implementation 2

## 7 Comparison and analysis of both implementations




The TFLite implementation is faster, more energy efficient per inference and consumes less flash memory.

The Native C implementation will be accelerated with the CGRA, TFLite will serve as baseline.

## 8 Future Work and Conclusion

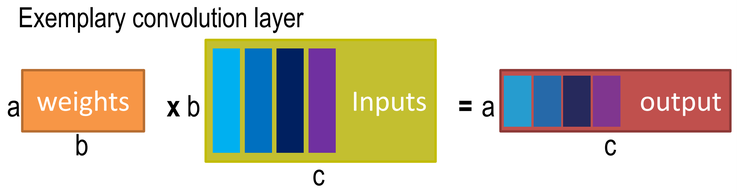
**Acceleration with HyCube[1] CGRA**

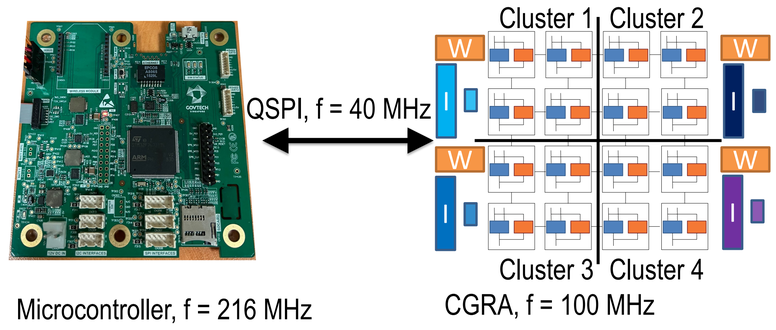
**weights** get communicated all at once, **input** gets communicated in sections due to memory restrictions

for the VWW application some weights are too large to be stored on the CGRA as a whole

- exploding communication overhead (= bottleneck)
- CGRA needs more on-chip memory

Exemplary convolution layer




## 9 References

1. M. Karunaratne, A. K. Mohite, T. Mitra, and L.-S. Peh. Hycube: A CGRA with Reconfigurable Single-cycle Multi-hop Interconnect, 2017, In Proceedings of the 54th Annual Design Automation Conference 2017
2. A.G. Howard et. al., 2017, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

sdw Stiftung der Deutschen Wirtschaft

NUS National University of Singapore | Computing