AI inference acceleration on FPGA

AIRA Seminar, 23.06.2022

Hardware Acceleration
Lab

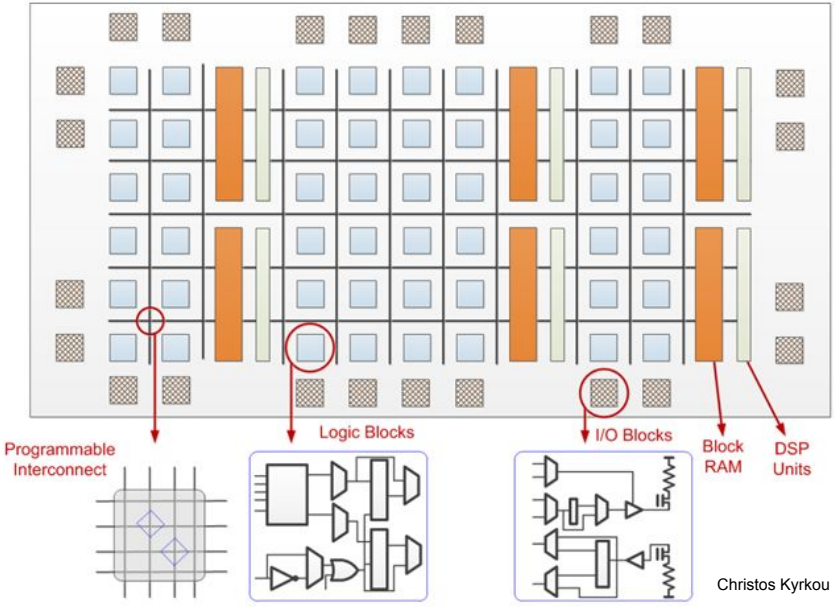
Bartosz Soból

Outline

- What is an FPGA?
- Differences between CPU, GPU and FPGA
- Inference software with case studies
- What about training?
- FPGA@FAIS

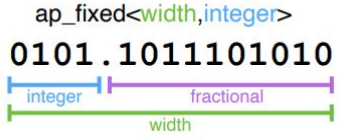
What is an FPGA?

- **Field Programmable Gate Array**
  - array of **universal** logic blocks
  - that can be **reprogrammed**
  - at **any time**
- Algorithms are mapped directly into hardware
- Supporting components:
  I/O, memory, DSPs, …

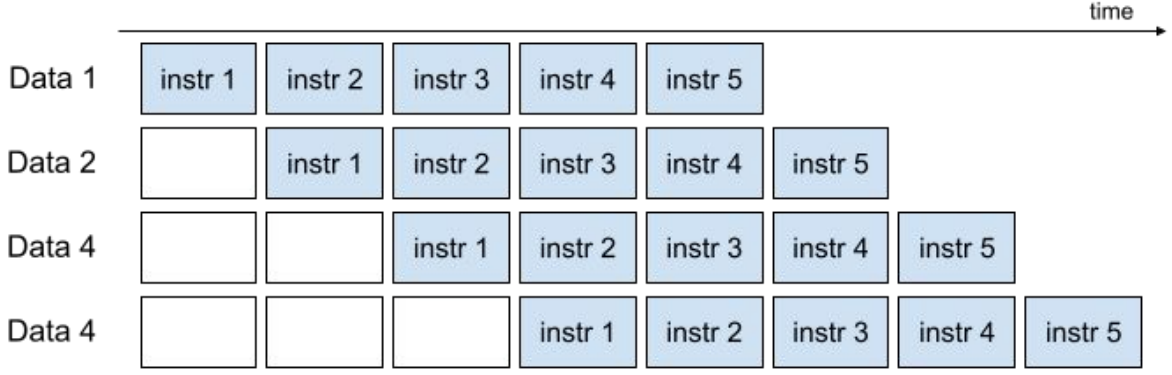


FPGA characteristics

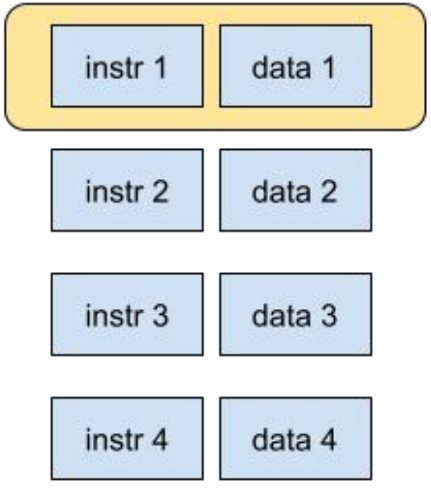
- Amount of chip resources used by algorithm is known
- Upper bound on processing time - latency - in clock cycles is strictly defined
- Arbitrary data types can be implemented efficiently - in hardware
  - eg. n-bit integer, fixed point types

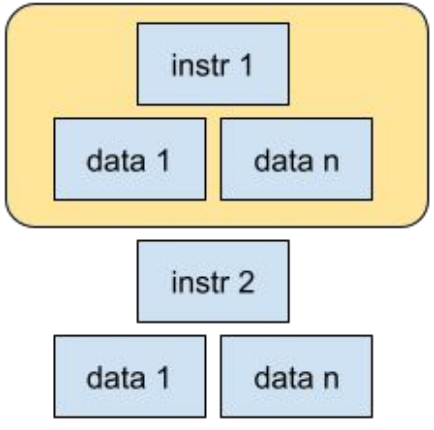


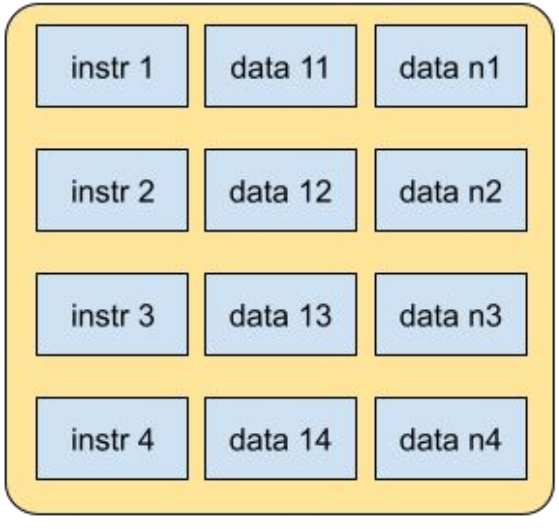
- Pipeline processing of data streams



CPU vs . GPU vs. FPGA







- Single Instruction Single Data
- Fixed architecture and instruction set
- High clock frequency: 2 - 5 GHz
- Executes programs
- Sequential/parallel computing
- Operating System

- Single Instruction Multiple Data
- Fixed architecture and instruction set
- Medium clock frequency: ~1 GHz
- Executes parts od programs (kernels)
- Parallel computing
- Accelerator platform

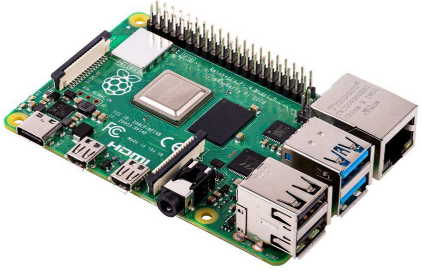
- Multiple Instruction Multiple Data
- Flexible architecture and instruction set
- Low clock frequency: < 500 MHz
- Executes hardware design (can be kernel)
- Parallel/pipeline computing
- Standalone or accelerator platform

CPU vs . GPU vs. FPGA

AMD EPYC

NVIDIA A100

AMD Xilinx ZCU102



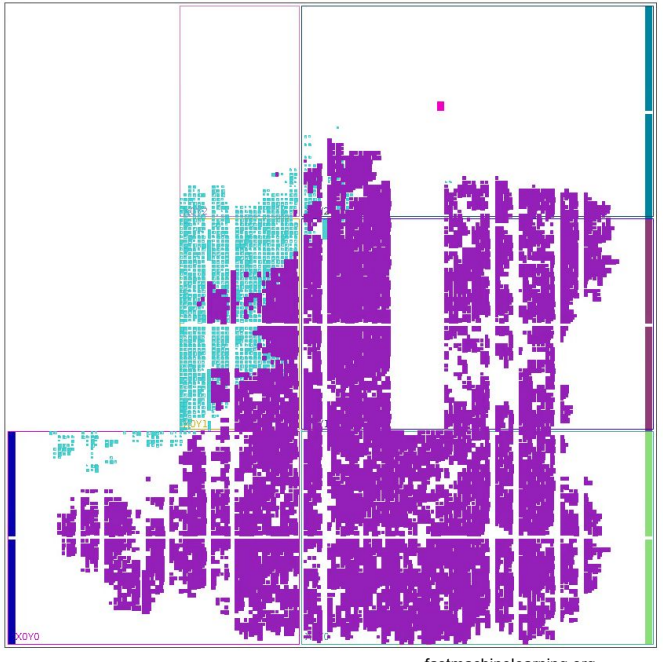
Raspberry Pi 4

NVIDIA Jetson Nano

AMD Xilinx Alveo U250

hls4ml

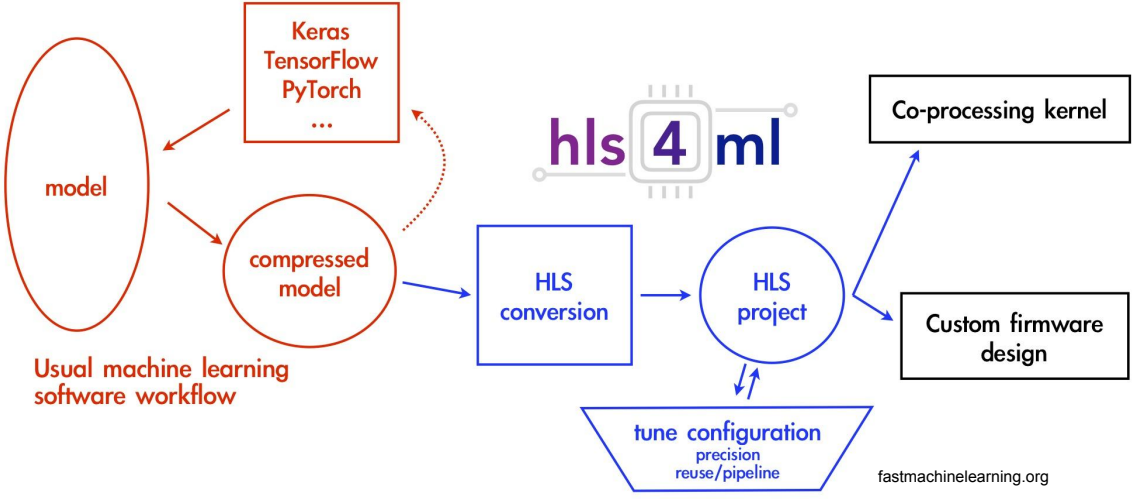
- Python package, developed as part of Fast Machine Learning Lab project
- Aims to translate pretrained models (PyTorch, TensorFlow, Keras, QKeras)
  - Into HLS (C++) code that can be compiled into FPGA firmware
  - Support for MLP, CNN, experimental RNN/LSTM/GRU
  - Support for all kinds of Xilinx devices (with enough resources) and also Intel
- Resulting design can be optimised for latency / throughput / resource usage



fastmachinelearning.org

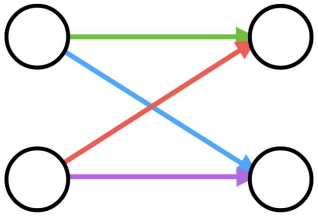
Hls4ml - optimisations

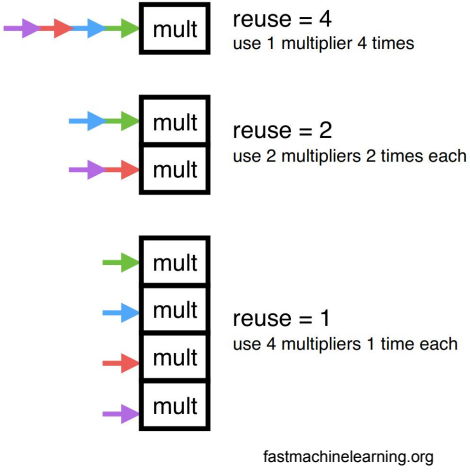
- **Precision/quantization** - customizable fixed-point arithmetic
  - Post-training in code/config - simple casting to lower precision
  - Training-aware - imported from QKeras models
- Utilities to test accuracy drop after post-training quantization



Hls4ml - optimisations

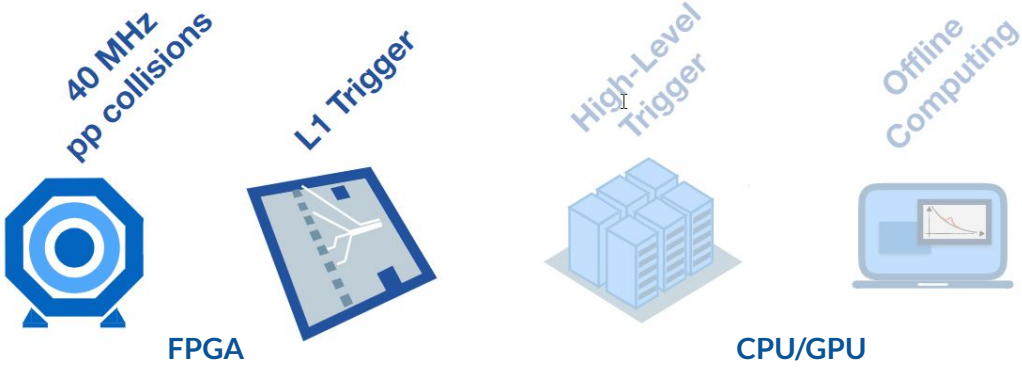
- **Reuse factor** - how many times multiplying unit is used in layer calculation
  - Higher reuse factor ->
    Lower latency, higher throughput
  - Lower reuse factor ->
    Lower resource usage





hls4ml - use case - **CMS L1 trigger**

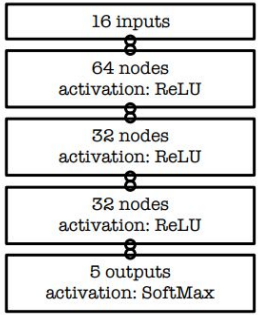
- **CMS** - second largest experiment at Large Hadron Collider (CERN)
- **L1 trigger** - first level data filtering, reducing data rate for further processing

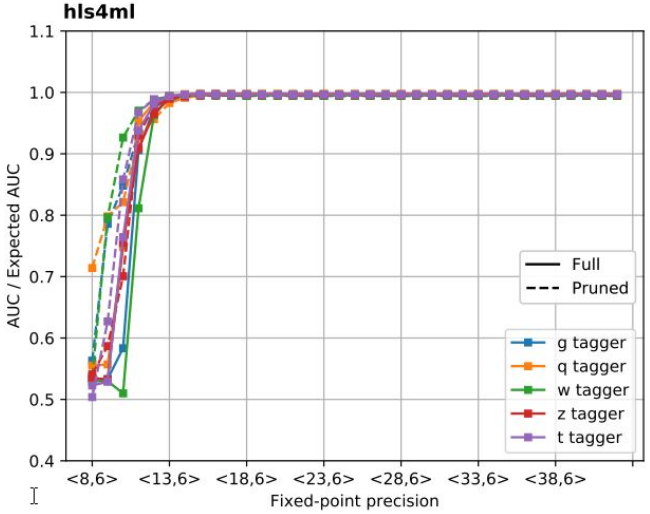


- Events (datasets) coming in with frequency 40MHz, designed for >100TB/s with future upgrades
- Latency constraint (~1 microsecond)

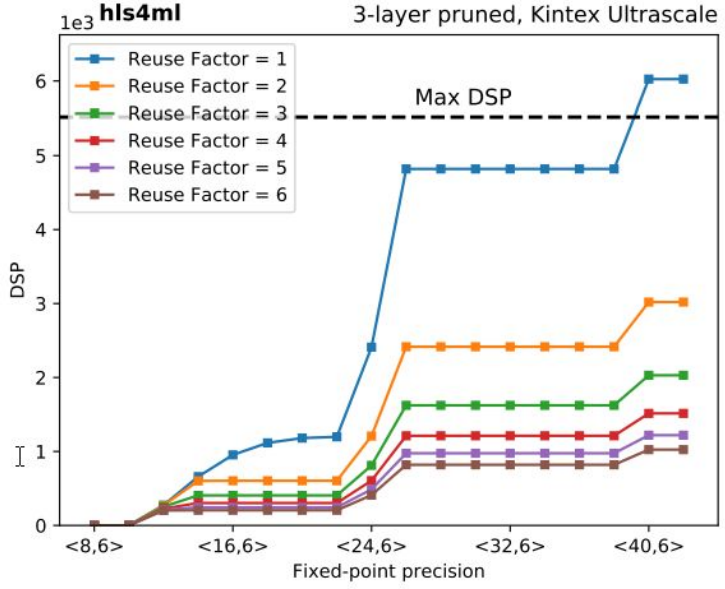
hls4ml - use case - **CMS L1 trigger** - ML solution

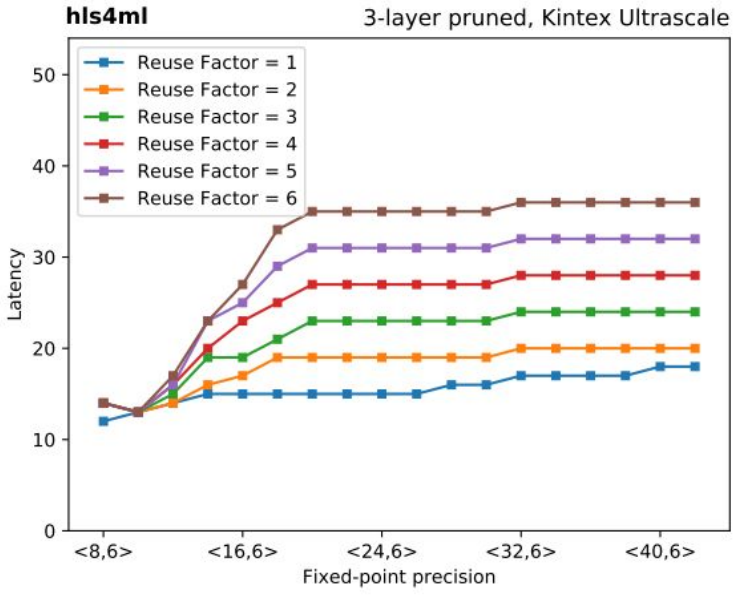
- ML techniques perform very well for this case
  - But it's hard to meet latency/throughput expectations on CPU/GPU
  - Traditional FPGA development: expensive and time-consuming
  - Usually simple models, a few layers, dense
- **Solution** - use existing models and automatically port to FPGA -
- **Studied model:**
  Jet tagging with 3-hidden layer MLP, 4.5k parameters





hls4ml - use case - **CMS L1 trigger** - Results



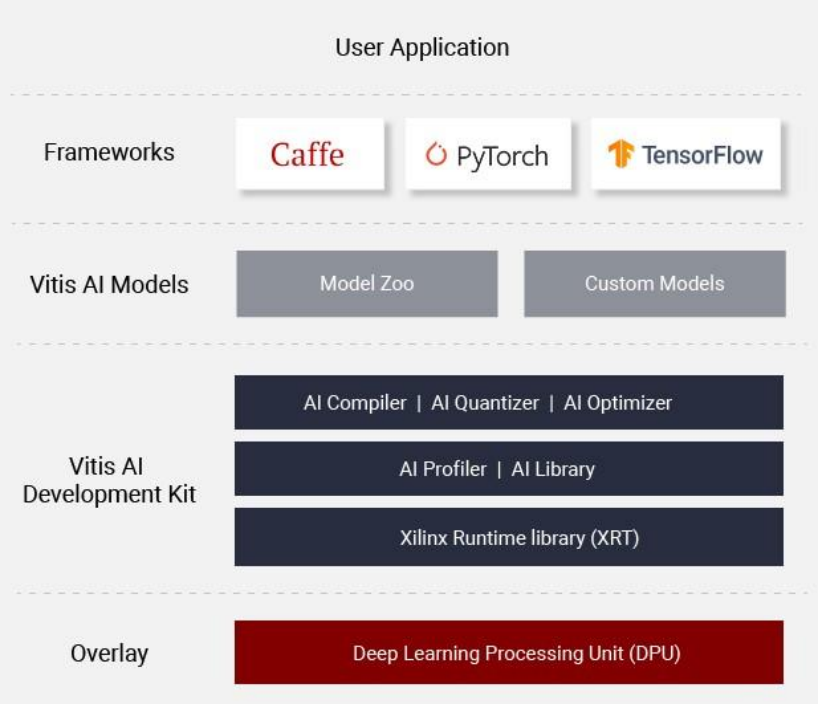


Vitis AI

- Official Xilinx stack for AI inference
  - Relatively new (2020)
- Basic concept similar to hls4ml
  - Take pretrained model and translate for FPGA
  - Different, more general implementation
  - More complete feature set
  - Requires host CPU
    - Supports wide range of Xilinx MPSoC devices with ARM cores and Alveo accelerators

Vitis AI

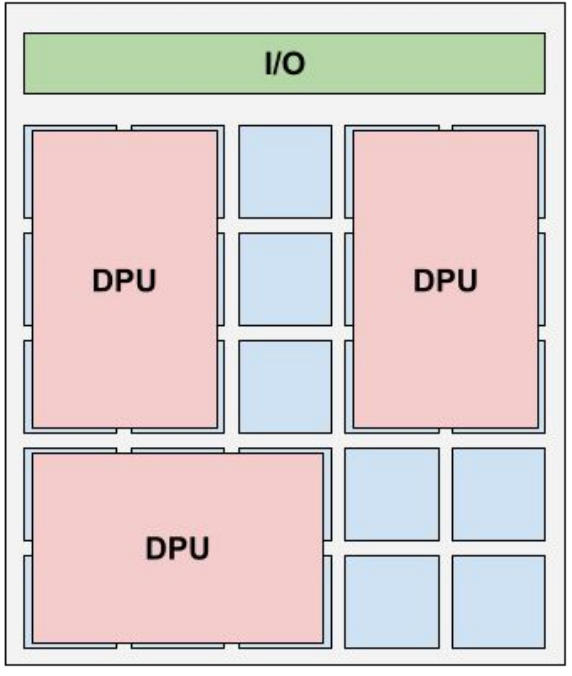
- Model development layer
  - Framework support,
  - Pre-prepared models
- Optimisations layer
  - Training-aware pruner and quantizer,
  - Profiler
- Hardware layer - DPU
  - Universal logic unit
  - Domain Specific Architecture
  - Instruction set for NN acceleration
  - Precompiled



AMD Xilinx

Vitis AI - DPU-based architecture

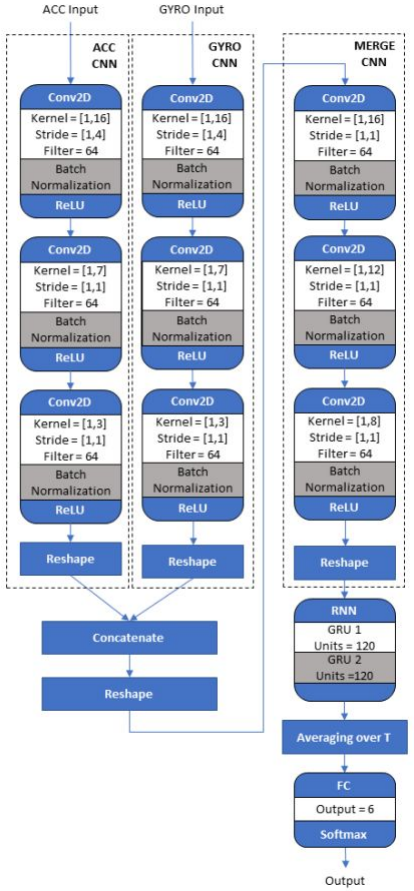
- DPU can act as specialized coprocessors,
  - are, in fact, programmed blocks on FPGA
- Multiple DPUs can run on single FPGA
- DPUs can accelerate MLP, CNN, RNN models
- Multiple variants for
  - Quantization method and bitwidth
  - Platform-specific features (HBM memory)
- Models aren't compiled to FPGA firmware, but to DPU instruction set
  - faster compilation and deployment



Vitis AI - use case - **low-power DeepSense**

- **DeepSense** - multimodal ML framework for
  - activity recognition based on data from gyroscope and accelerometer
- **Goal** - achieve low-power and low-latency inference in DeepSense
- **Approach** - Vitis AI applied on small Xilinx MPSoCs: Ultra96 and ZCU102
  - Baseline comparison with LG Nexus 5 smartphone and Raspberry Pi 3

Resource	Ultra96-V2	ZCU102
LUT (K)	154	600
FF (K)	141	548
BRAM (Mb)	7.6	32.1
DSP Slices	360	2,520

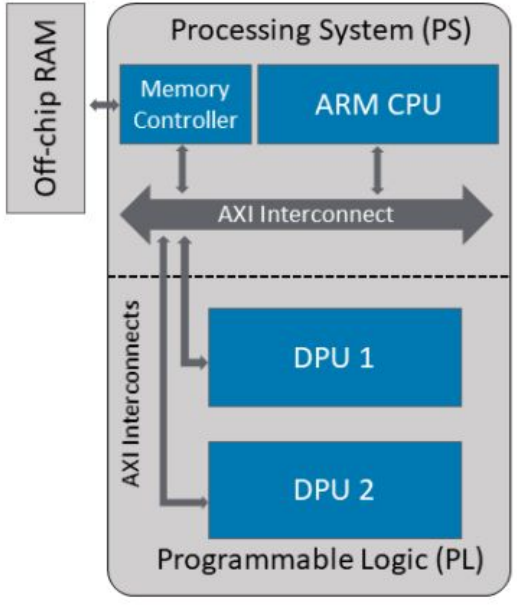


Vitis AI - use case - **low-power DeepSense**

- Hardware design based on 2 DPUs
- Different configurations tested

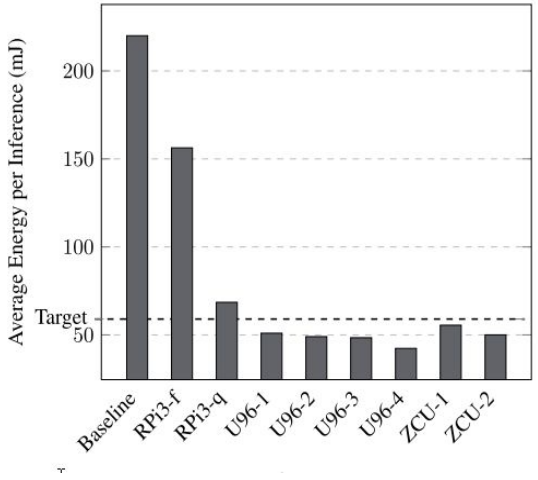
Config.	Board	PE Clock Frequency (MHz)	RAM usage	DSP usage	Low-Power Mode
U96-1	Ultra96	400	High	Low	Off
U96-2	Ultra96	400	Low	Low	Off
U96-3	Ultra96	400	Low	High	Off
U96-4	Ultra96	400	Low	High	On
ZCU-1	ZCU102	600	Low	High	Off
ZCU-2	ZCU102	600	Low	High	On

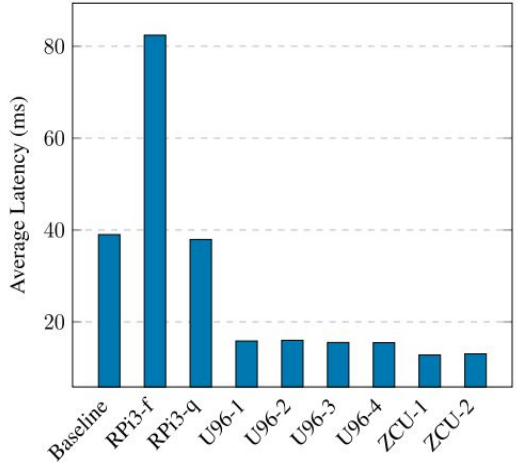
- Compared with:
  - Standard float32 model on Nexus and RPi
  - Quantized int8 model on RPi

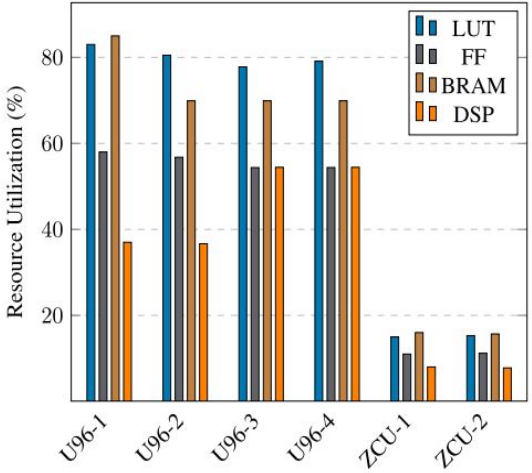


Vitis AI - use case - **low-power DeepSense**

	Baseline	RPi3-q	U96/ZCU
Accuracy	95.31%	94.69%	94.19%







Licensing

- **hls4ml** - open-source, free, Apache License
- **Vitis AI** - open-source, free, Apache License
    - Except AI Optimizer (pruner) which is proprietary and paid (researchers can apply for free temporary license)

Training on FPGA?

- Training require a lot of resource
  - Large FPGA-based PCIe accelerators similar to GPUs are a relatively new topic (Xilinx Alveo)
- Not much effort put into this topic, GPUs are good enough
- A few research projects exploring possibilities, mostly on CNNs (e.g. DarkFPGA, Barista)
  - Whether modern FPGA architectures can perform competitively with GPU is an open question
  - Usually not raw performance (speed) but energy efficiency is compared

FPGA AI activities at FAIS

- MSc student, Grzegorz Kozioł, performing in-house evaluation of Vitis AI stack
  - Custom CNN networks and ResNet models
  - Pruning, quantization
  - Computing performance, accuracy
  - Profiling tools
- Gathering know-how
- Potential future use: J-PET project, PANDA experiment

FPGA resources at FAIS

- AMD Xilinx Alveo accelerator cards: 1xU280, and 1xU50
- Two AMD EPYC servers with 2xU280 each to arrive in July
- Many development boards with smaller chips
- We also have 2xRTX2080 GPU
- Contact us if you need access
  - grzegorz.korcyl@uj.edu.pl
  - bartosz.sobol@doctoral.uj.edu.pl

Summary

- There exist usable tools for deploying NN inference on FPGAs
  - They perform best on power or latency constrained applications
  - They improve rapidly, aggressive development (mostly from Xilinx)
- Training NN on FPGA is one open topic
- We have solid FPGA resources available at FAIS