Adaptive CNN execution on edge FPGAs

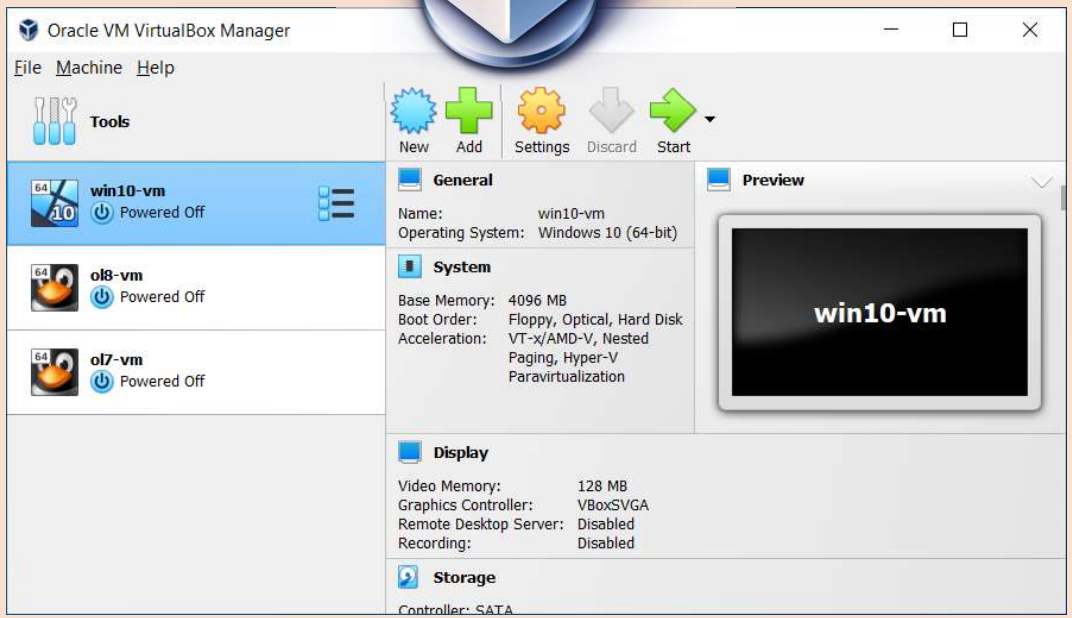
Francesco Ratto[1], Federico Manca[1] and Claudio Rubattu[2]

{francesco.ratto, federico.manca}@unica.it

{crubattu}@uniss.it

Getting Ready

Option 1:



Option 2:

GitLab

Follow the README at:

https://gitlab.com/fedemanca/budva2024/-/tree/CPS_Alghero_2024

Outline

1. **Introduction**
   - Who we are
   - MYRTUS approach
   - Activities in MYRTUS
2. **Model inference on reconfigurable edge devices**
   - Reconfigurable architectures
   - High-level synthesis
   - Architectures for CNN inference on FPGAs
3. **Model training for reconfigurable edge devices**
   - Convolutional Neural Networks (CNN)
   - Reducing CNN Complexity
   - Exporting a CNN
4. **MYRTUS Toolchain for CNN Inference on FPGAs**
   - From a CNN to Streaming Architecture
   - From QONNX to a Streaming Specification
   - Streaming Architecture Synthesis
   - Results
   - Towards Adaptivity

1. Introduction

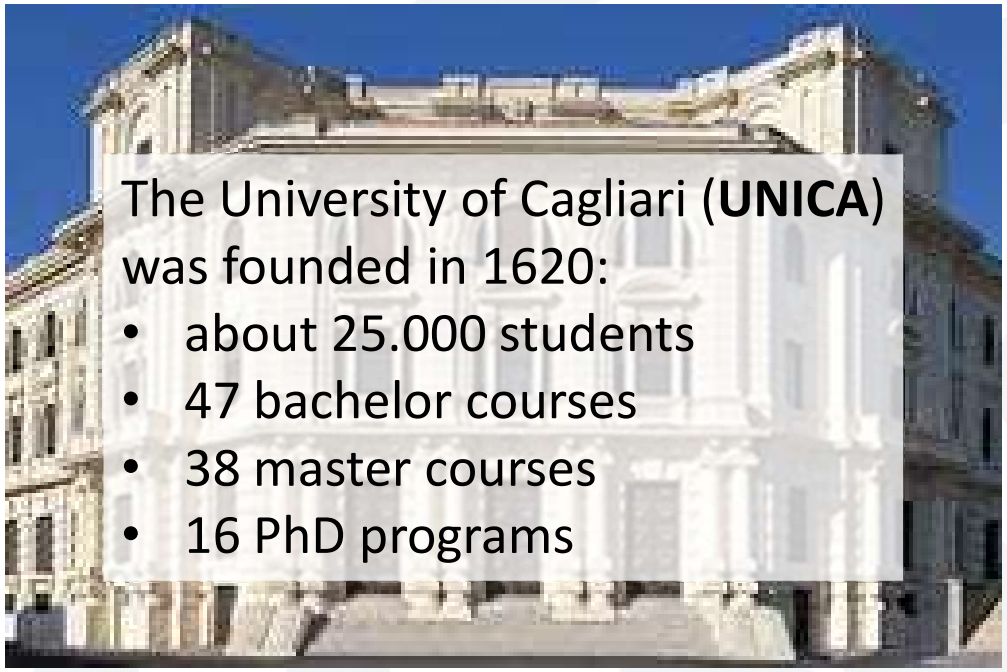
Who we are

UNISS
UNIVERSITÀ
DEGLI STUDI
DI SASSARI

The University of Sassari (**UNISS**) was founded in 1558:

- over 10.000 students
- 34 bachelor courses
- 28 master courses
- 10 PhD programs

UNIVERSITÀ DEGLI STUDI
DI CAGLIARI



The University of Cagliari (**UNICA**) was founded in 1620:

- about 25.000 students
- 47 bachelor courses
- 38 master courses
- 16 PhD programs

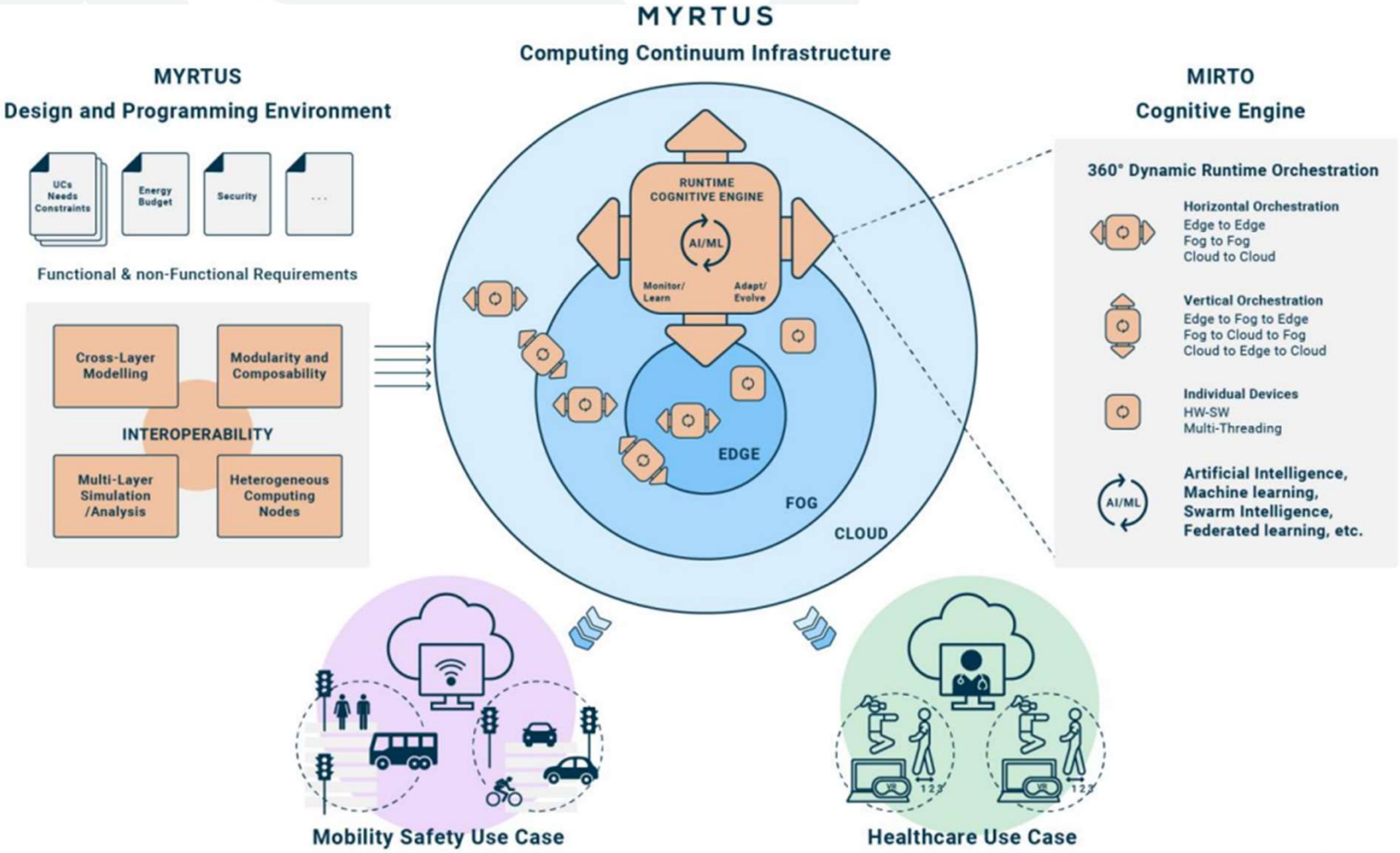
Who we are

Francesco Ratto is an Assistant Professor at UNICA.

Federico Manca is a PhD Student at UNICA.

Claudio Rubattu is an Assistant Professor at UNISS.

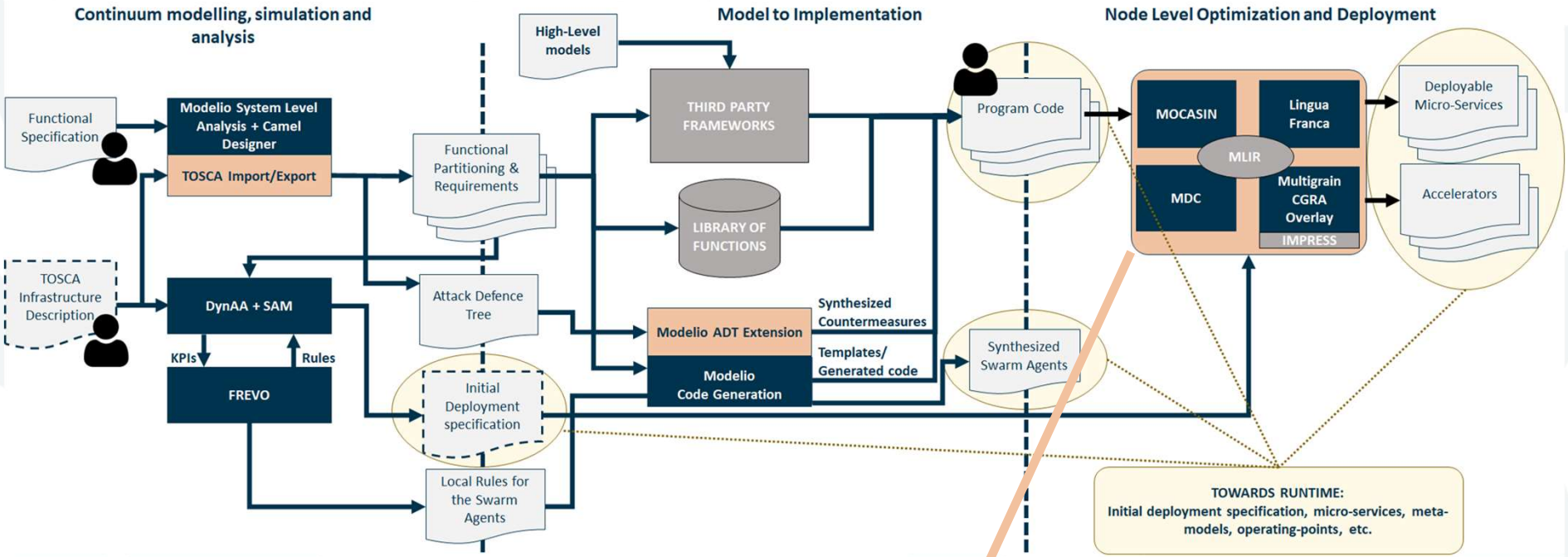
MYRTUS approach



The **MYRTUS consortium** comprises 8 countries and **14 partners**, and brings together different types of competencies from low-level architectural details definition to software management strategies.

MYRTUS has received funding from the European Commission for ~5,6M €.

Activities in MYRTUS

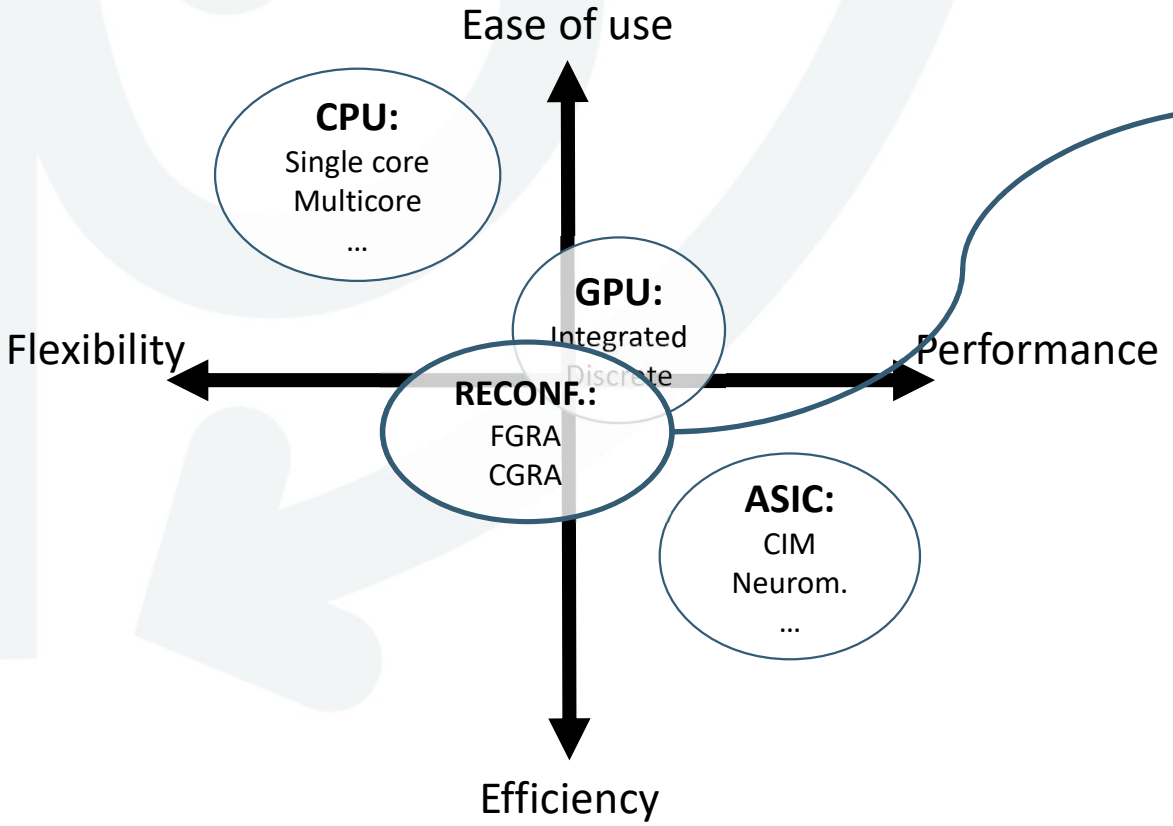


Multi-dataflow Composer tool extension:
support for approximate computing and CNN deployment from ONNX

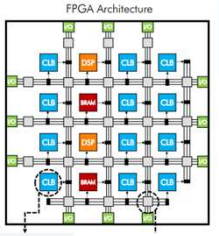
2. Model inference on reconfigurable edge devices

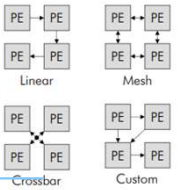
Reconfigurable architectures

Heterogenous Multi-processor SoCs



- **Fine Grained (FG):**
  - FPGA only
  - Bitstream load
- **Coarse Grained (CG):**
  - both in ASIC and FPGA
  - Register configuration

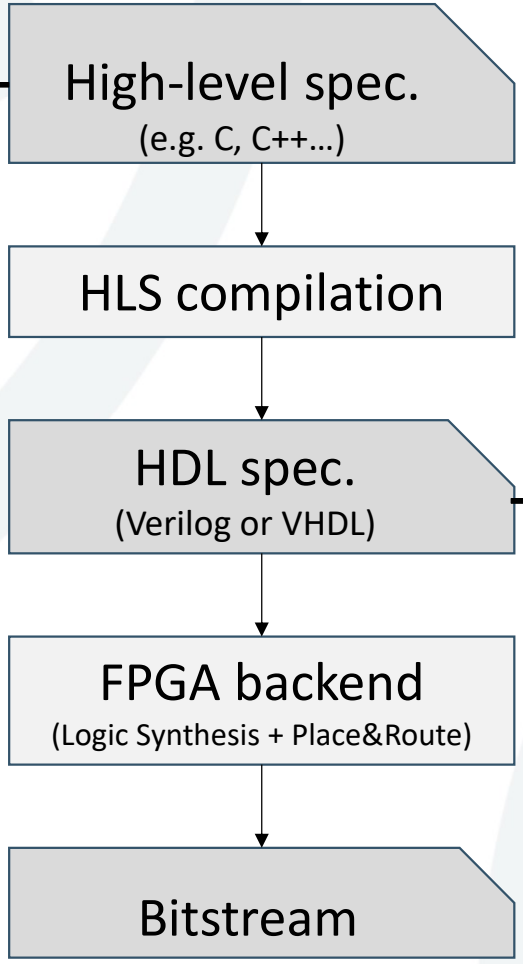




	FG	CG
Granular.	**bit-level**	**word-level**
Flexibility	☺	😐
Speed	😐	☺
Memory	☹	☺

High-level Synthesis

```
int add(int a, int b){
        return a+b;
}
```



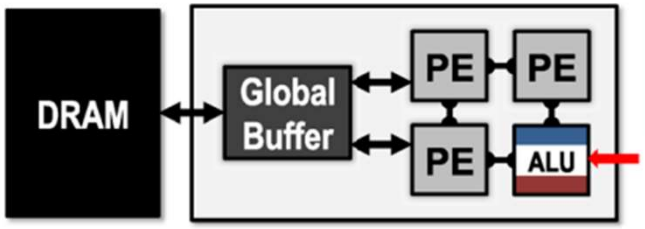

```
module add (
        input  ap_start;
        output  ap_done;
        output  ap_idle;
        output  ap_ready;
        input  [31:0] a;
        input  [31:0] b;
        output  [31:0]
ap_return;
);
assign ap_done = ap_start;
assign ap_idle = 1'b1;
assign ap_ready = ap_start;
assign ap_return = (b + a);
endmodule //add
```

Architectures for CNN inference on FPGAs

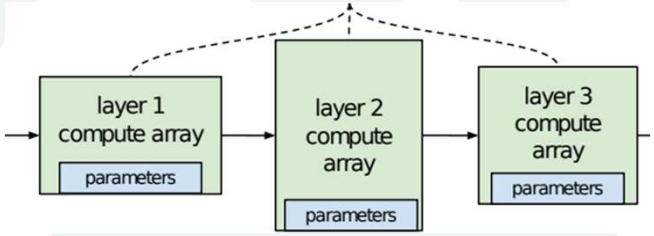
- a **vector processor** with instructions specific to accelerating the primitives' operations of convolution [Cococcioni, Garofalo].

RISC-V®
RISC-V: The Free and Open RISC Instruction Set Architecture

- A **single processing engine**, usually in the form of a systolic array [Cnp, FPDNN, NEURAGHE, AMD-DPU].



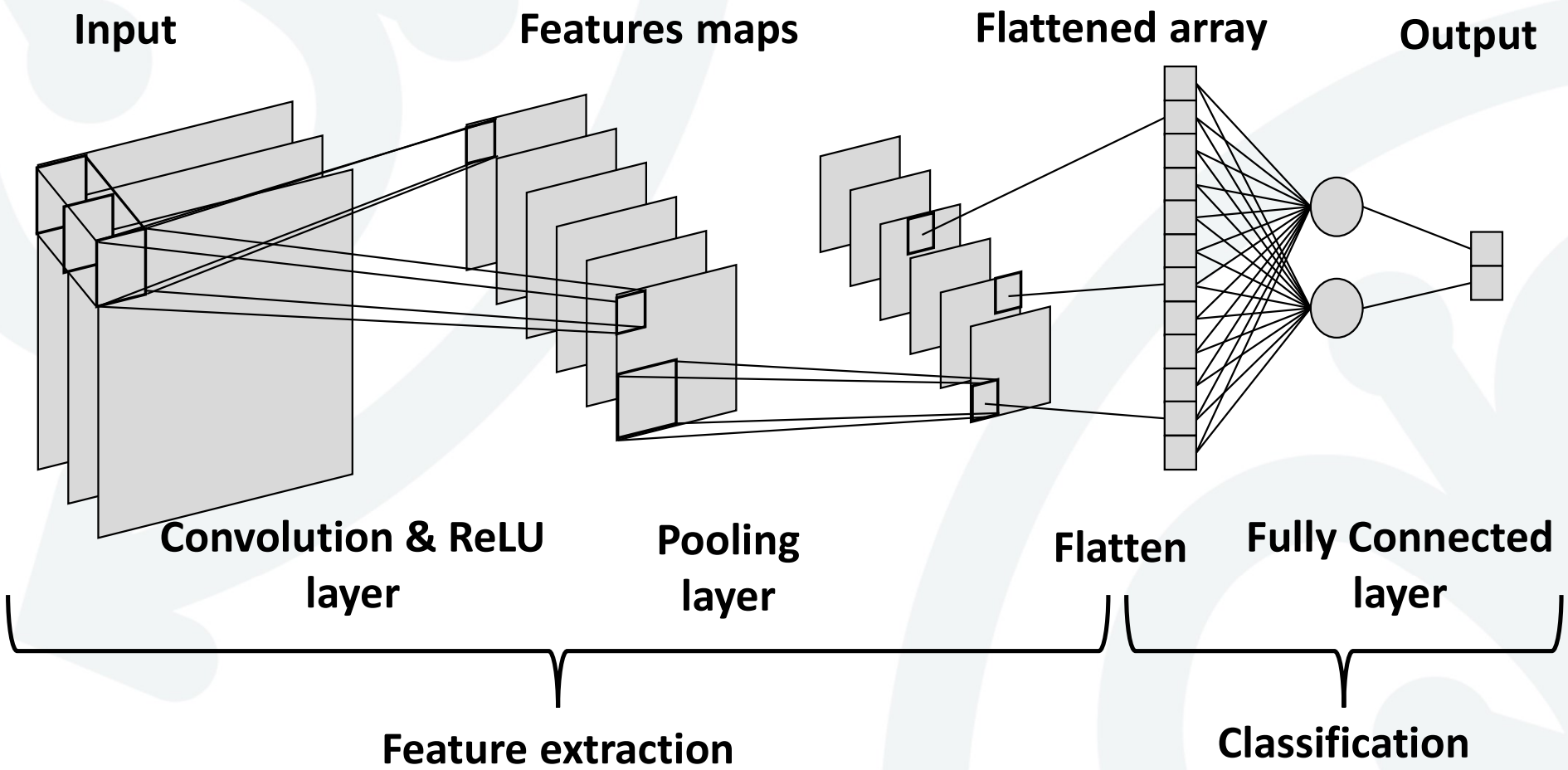
- a **streaming/dataflow architecture**, consisting of one processing engine per network layer [FINN, HLS4ml, Ratto].



Programmability ⟷ Specialization

3. Model training for reconfigurable edge devices

Convolutional Neural Networks (CNN)



Reducing CNN Complexity: Approximation

Approximate computing trades off computation quality with effort expended [Mittal]

Floating point

Decimal Representation

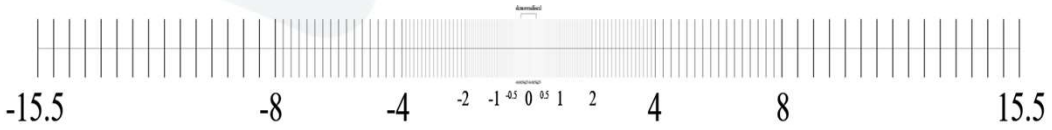
Mantissa

Exponent

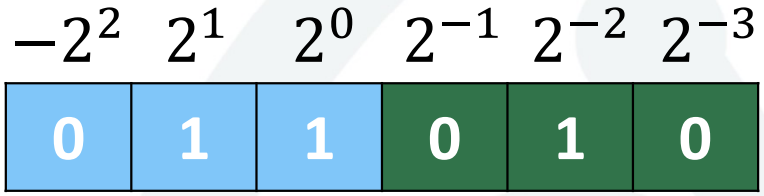
$$12345 = 1.2345 \times 10^4$$

IEEE 754 float

1 bit 8 bits 23 bits



Fixed point



$= 3.25_{10}$

Integer bits Fractional bits

Reducing CNN Complexity: Quantization

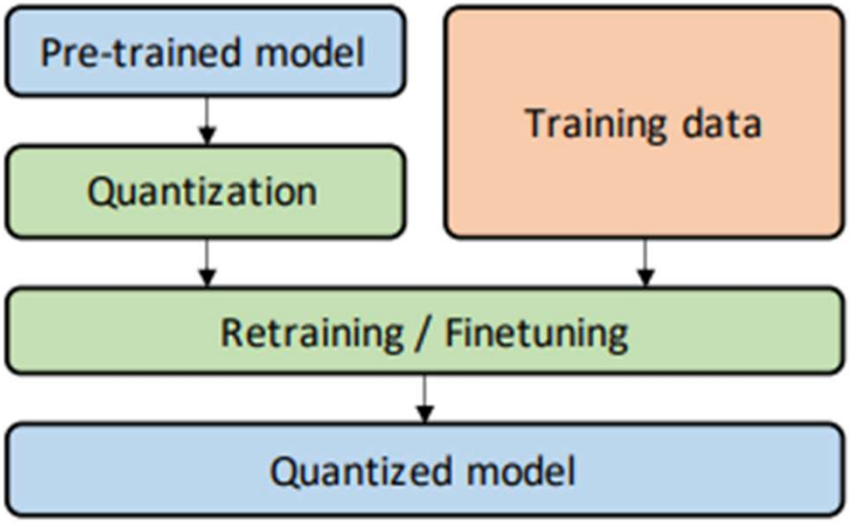
Quantization is the process of mapping continuous infinite values to a smaller set of discrete finite values.

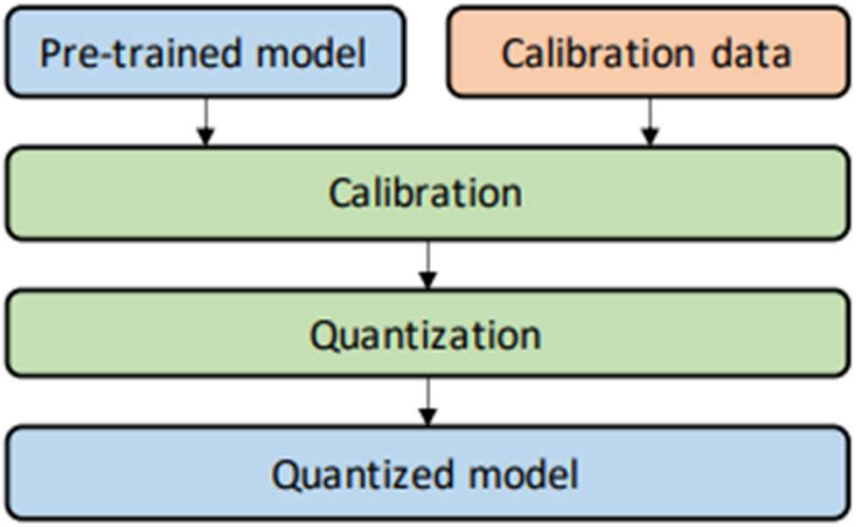
The former are represented by floating-point values, the latter by **fixed-point** values:

$$Q(x) = \Delta \times \left\lfloor \frac{x}{\Delta} \right\rfloor$$

Quantization is particularly relevant in applications like NNs that have demonstrated remarkable **resilience to errors** [Hubara].

Reducing CNN Complexity: Training





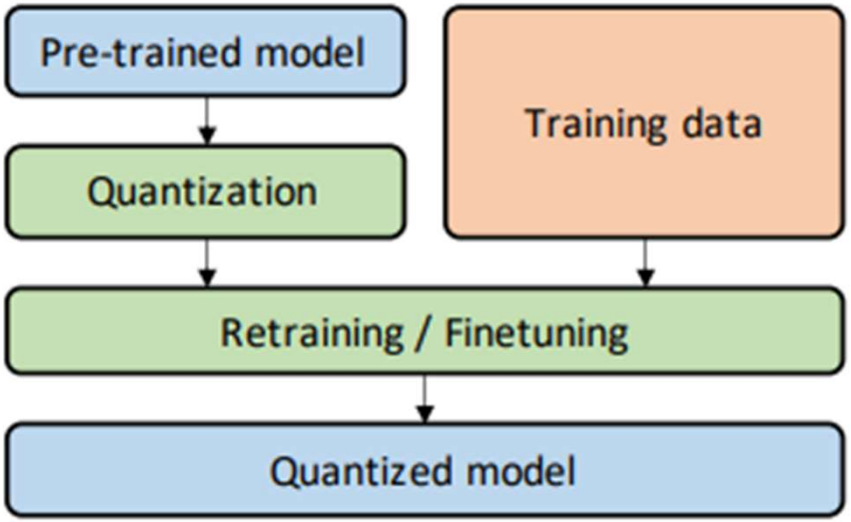
Quantization-Aware Training (QAT):

- achieves higher accuracy;
- It uses quantized data in the forward pass and float in the backward pass [Gholami];

Post-Training Quantization (PTQ):

- It is faster and simpler than QAT;

Reducing CNN Complexity: QAT libraries

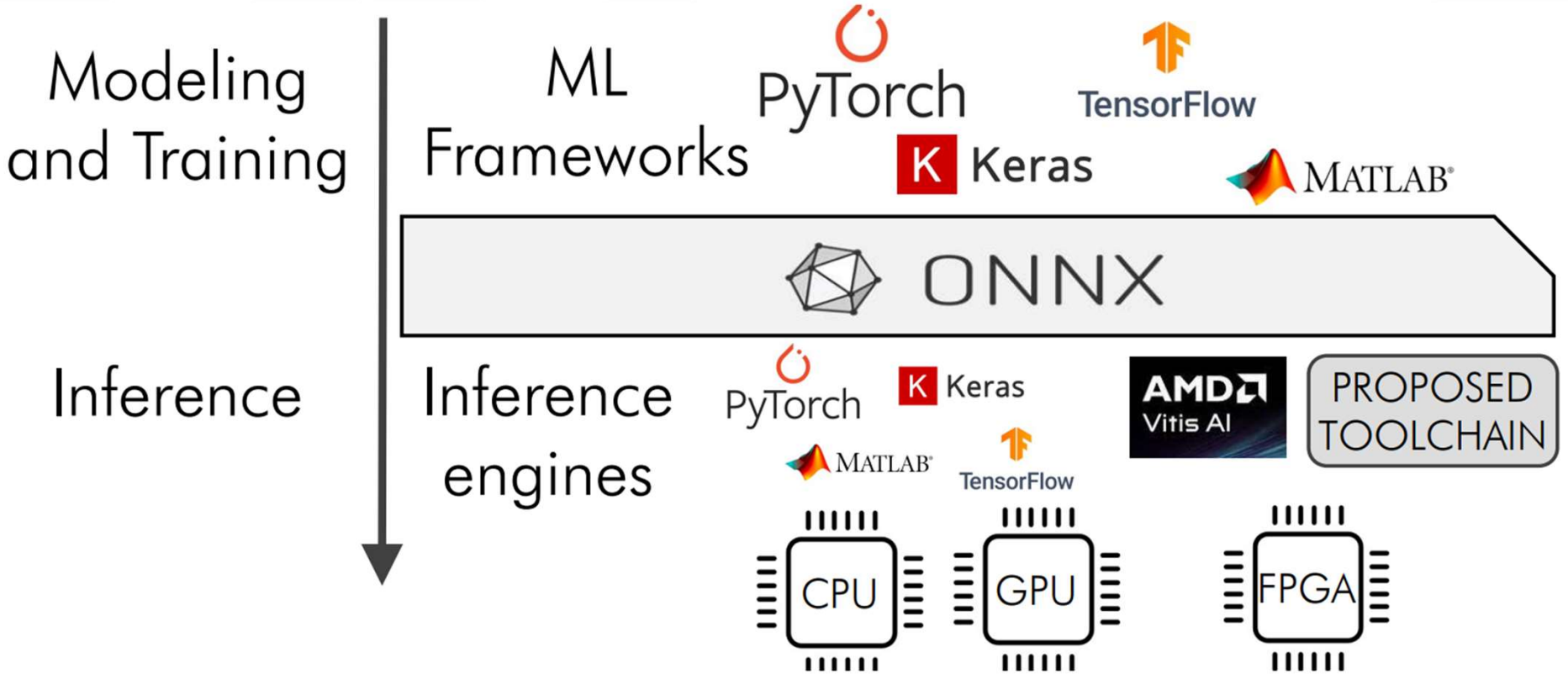


Quantization-Aware Training (QAT):

- achieves higher accuracy
- It uses quantized data in the forward pass and float in the backward pass

Library	From	API
Brevitas	Xilinx	Pytorch
Larq	Larq	Keras - TF
QKeras	Google	Keras - TF

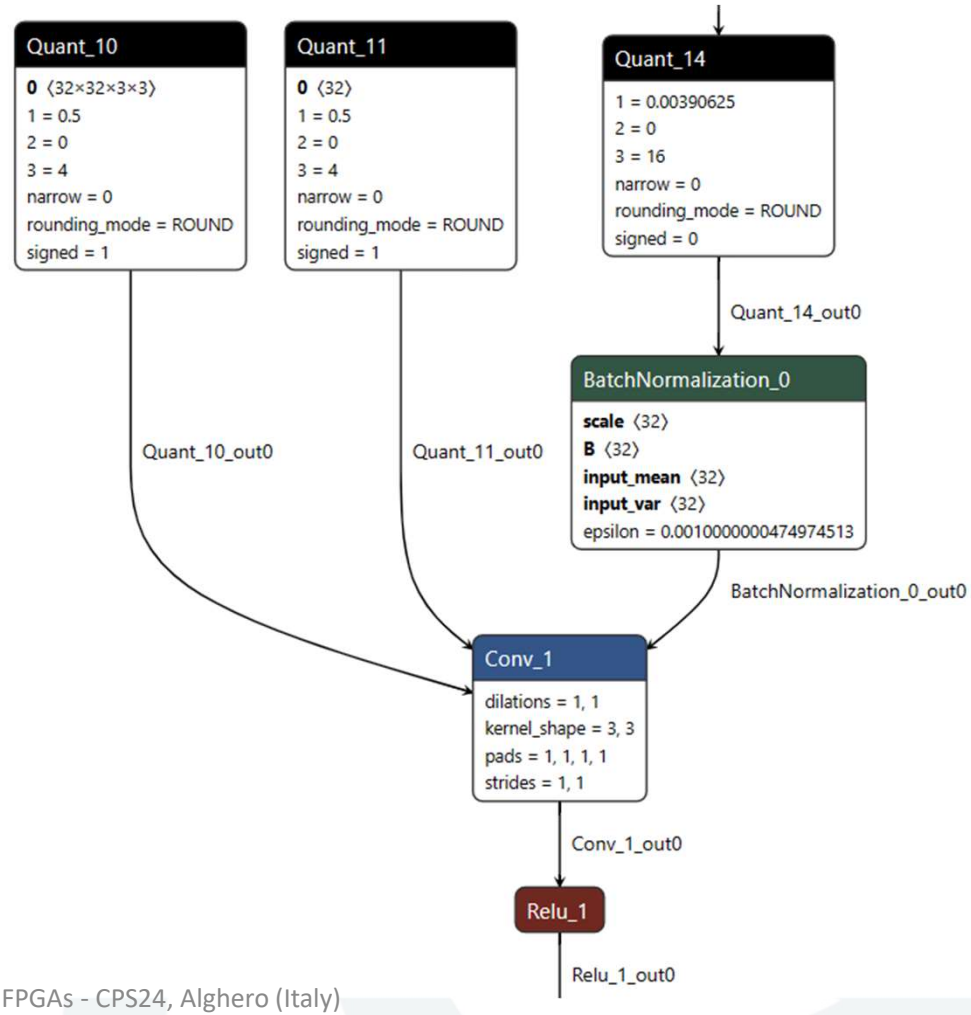
Exporting a CNN: the ONNX format



Exporting a CNN: the QONNX format

QONNX (Quantized ONNX) introduces new custom operators for quantization to represent arbitrary-precision uniform quantization in ONNX [Qonnx]:

- Quant
- BipolarQuant
- Trunc



How to open the notebook

 # Open the terminal and browse to the budva2024 folder:

```
cd Tutorial_Myrtus/ #(VM Only)
cd budva_2024
```

 # Activate the virtual environment:

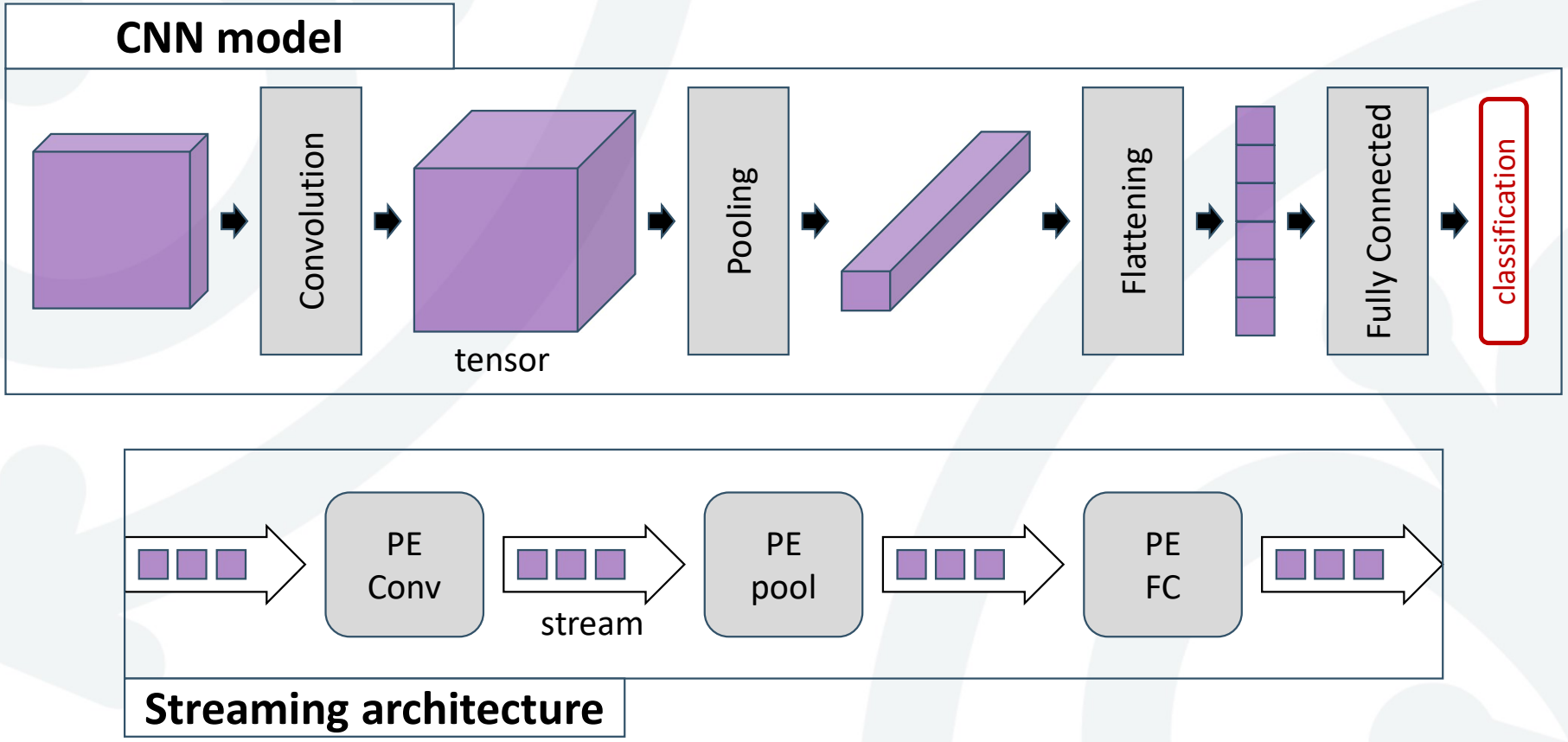
```
conda activate myenv
```

 # Launch the interactive notebook:

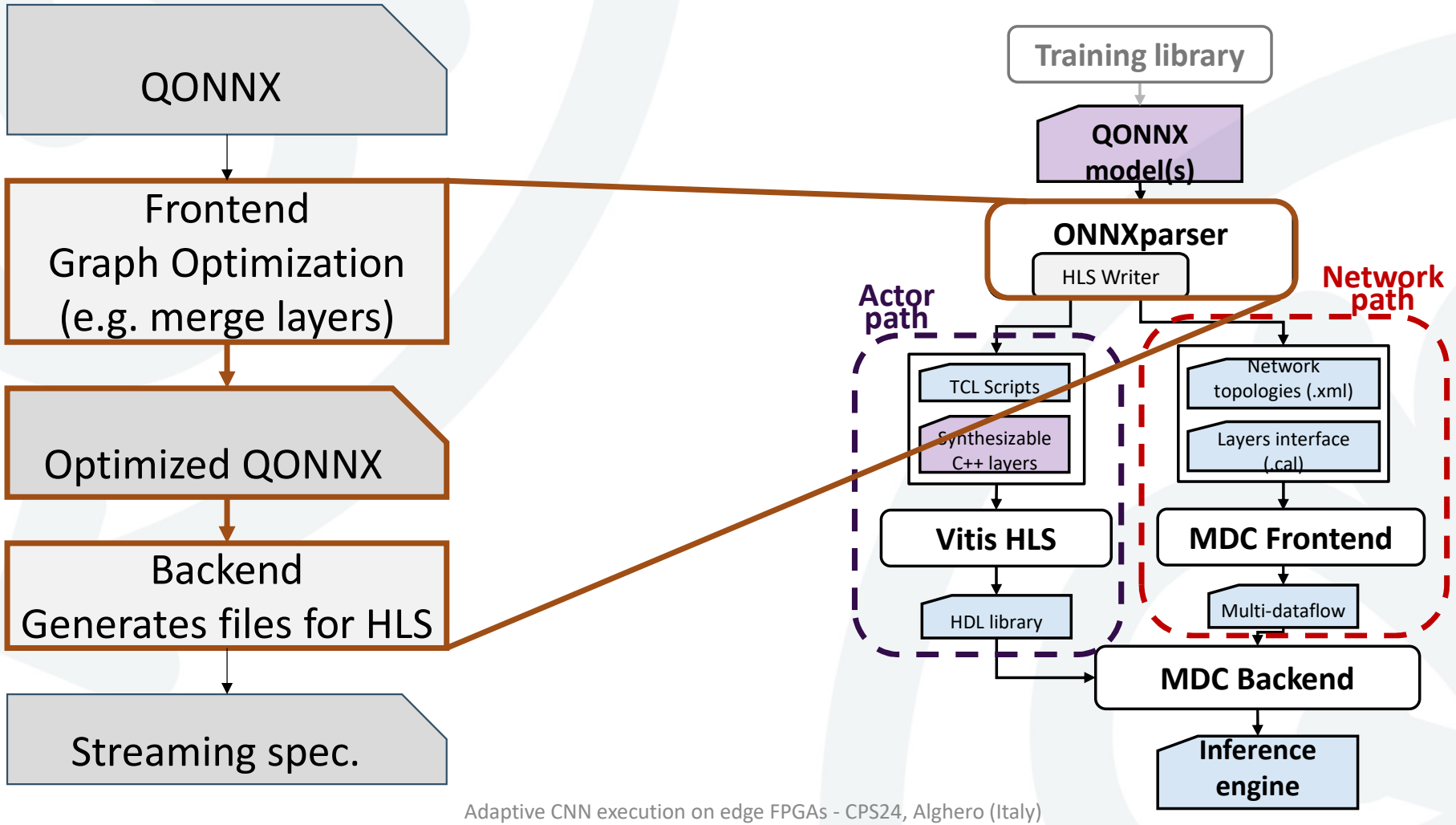
```
jupyter notebook
```

4. MYRTUS Toolchain for CNN Inference on FPGAs

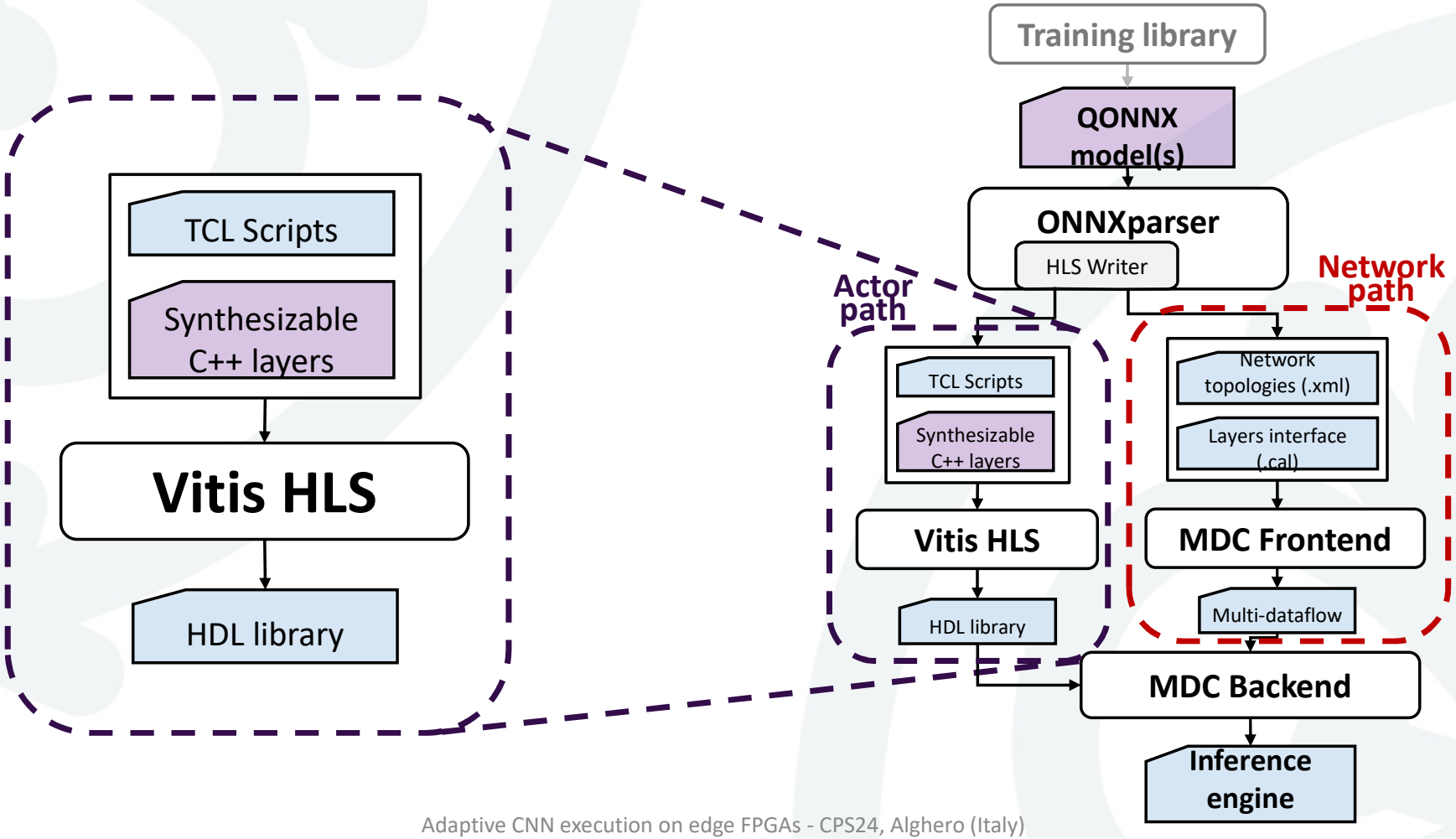
From a CNN to a Streaming Architecture



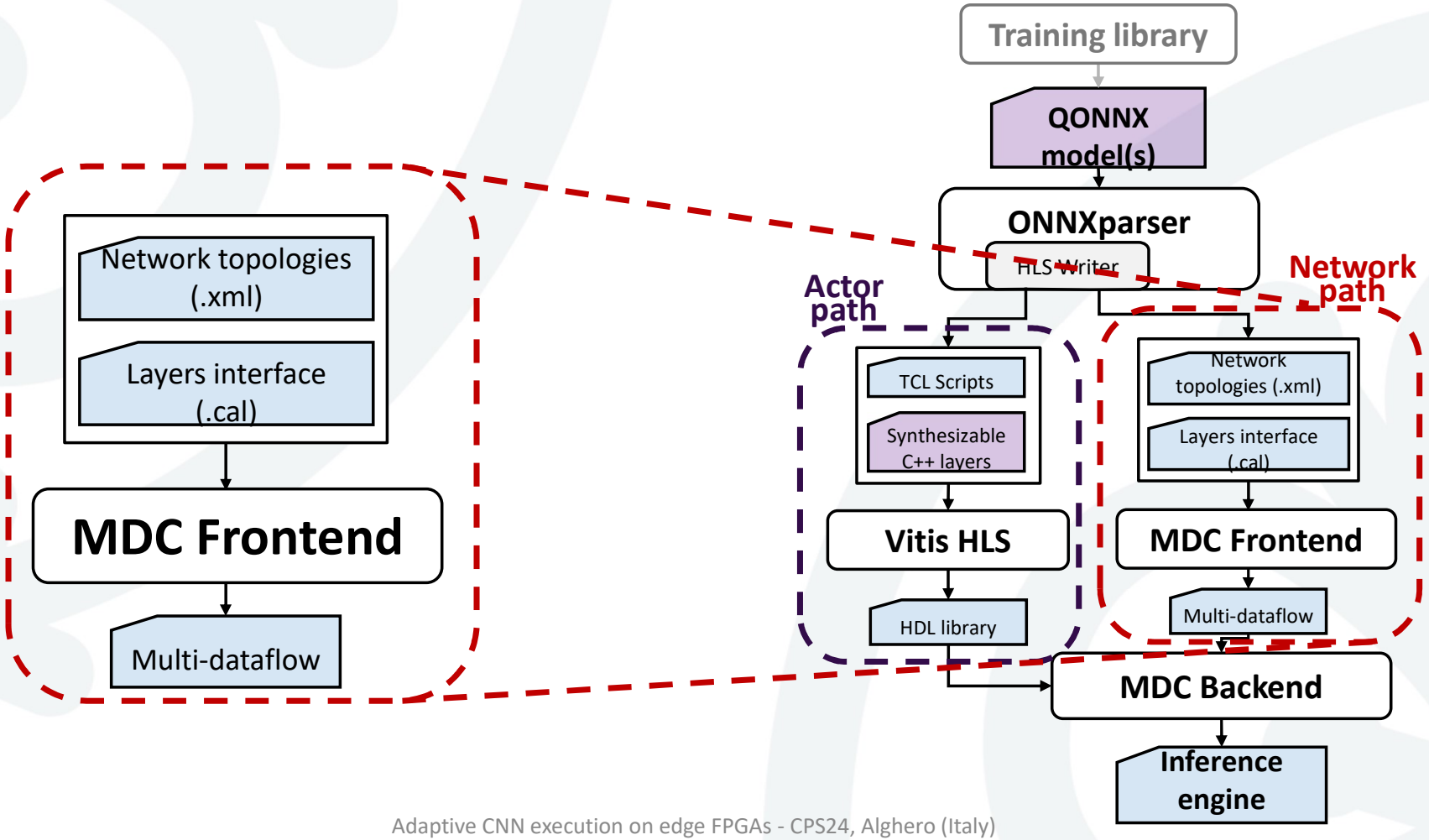
From QONNX to a Streaming Specification



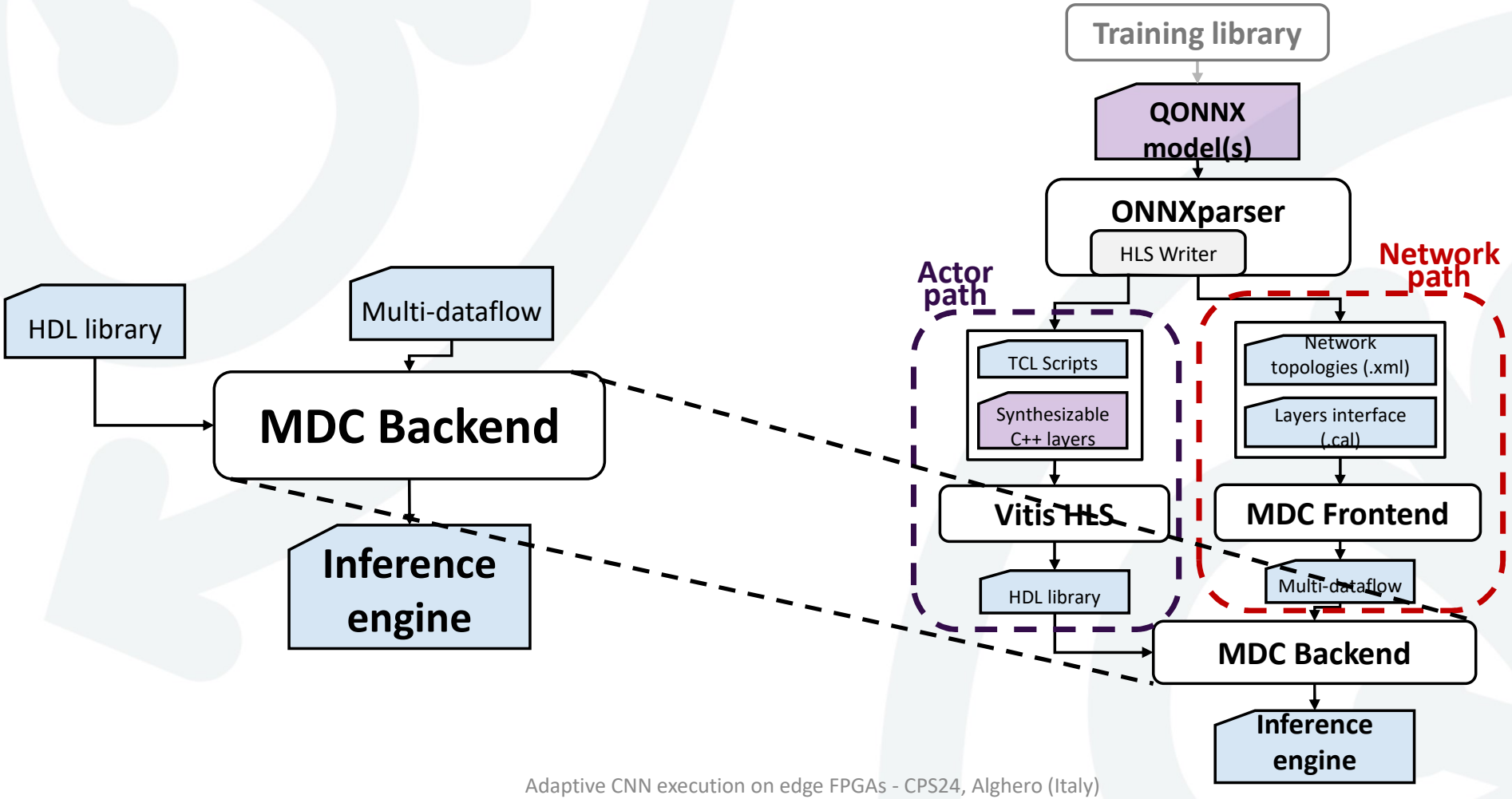
Streaming Architecture Synthesis: Actors



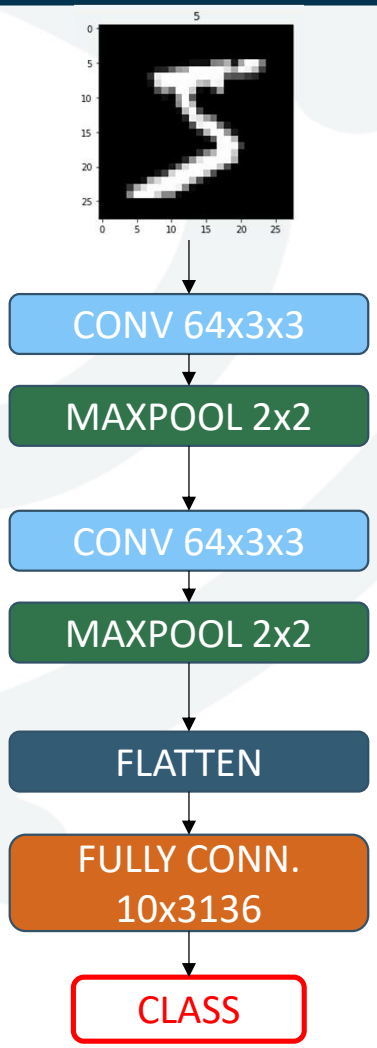
Streaming Architecture Synthesis: Network



Streaming Architecture Synthesis



Results: tiny CNN for MNIST classification

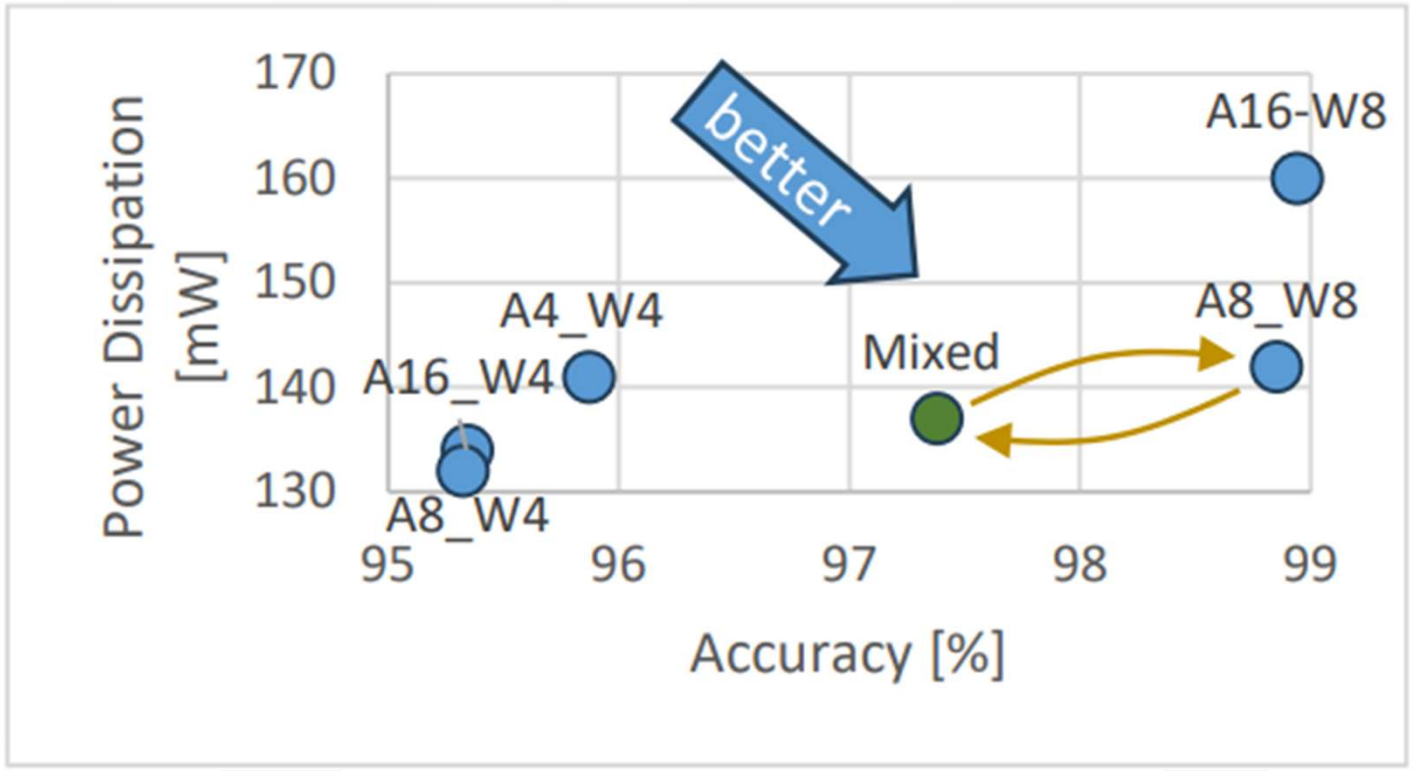


Application: tiny CNN for MNIST classification [Manca]

Board: AMD KRIA SoM

Demo

Results: Execution trade-offs

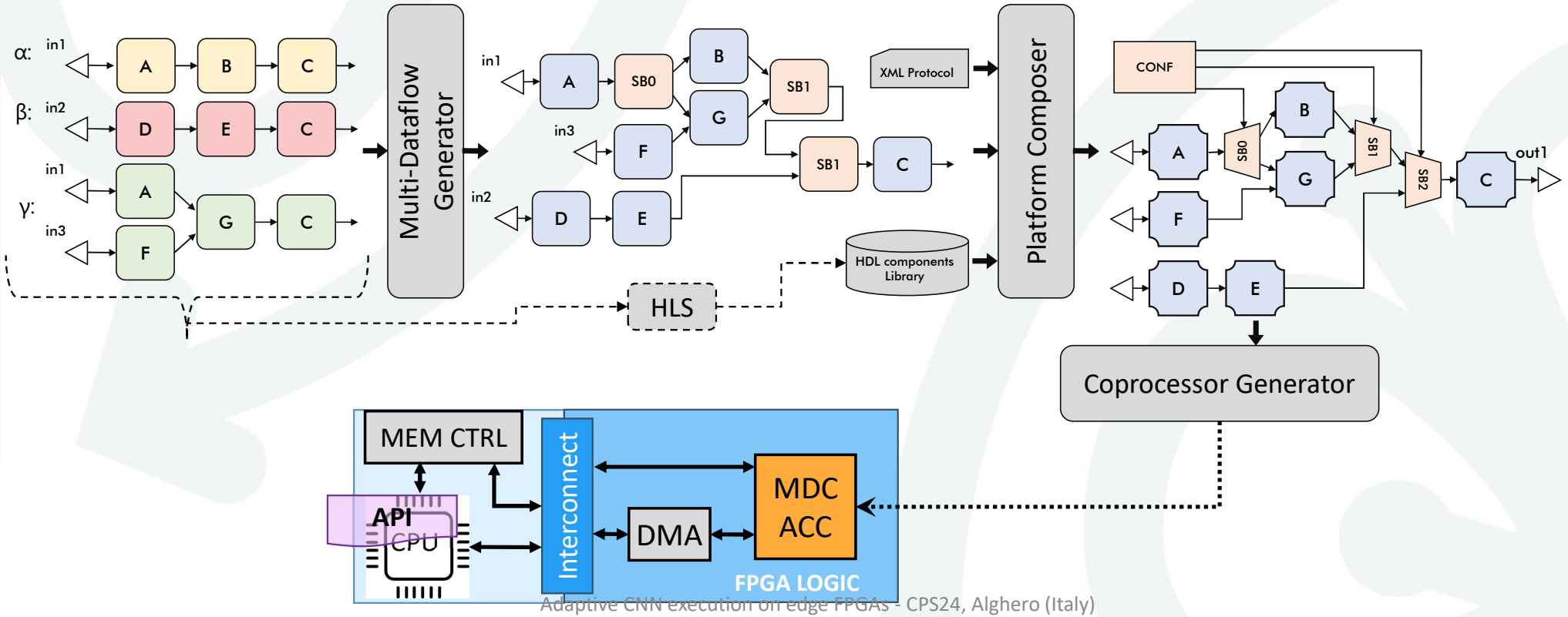


Ax_Wy:

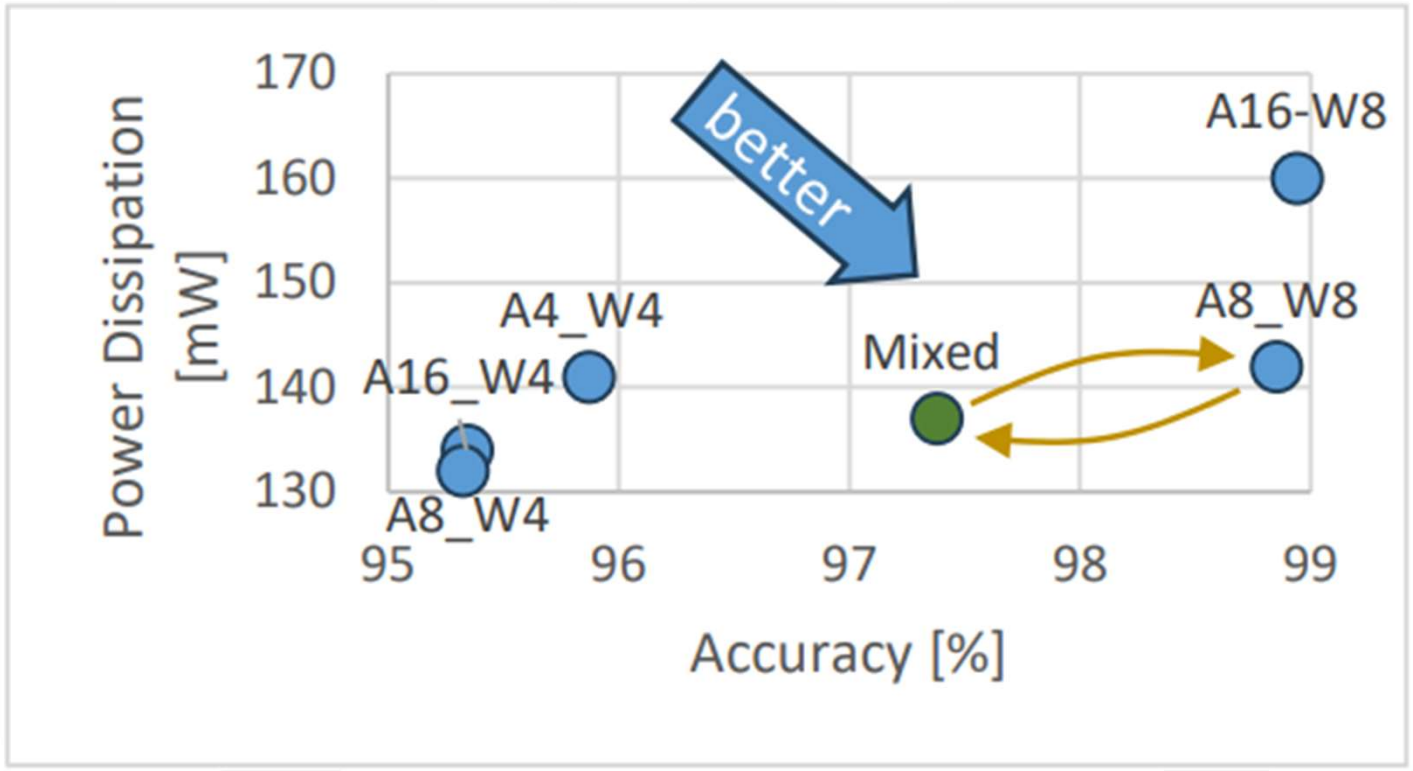
- **x** is the number of bits used for representing **activations**;
- **y** is the number of bits used for representing **weights**;

Towards Adaptivity: Multi-Dataflow Composer

- MDC is an open-source tool for designing and deploying CG reconfigurable accelerators [Sau].



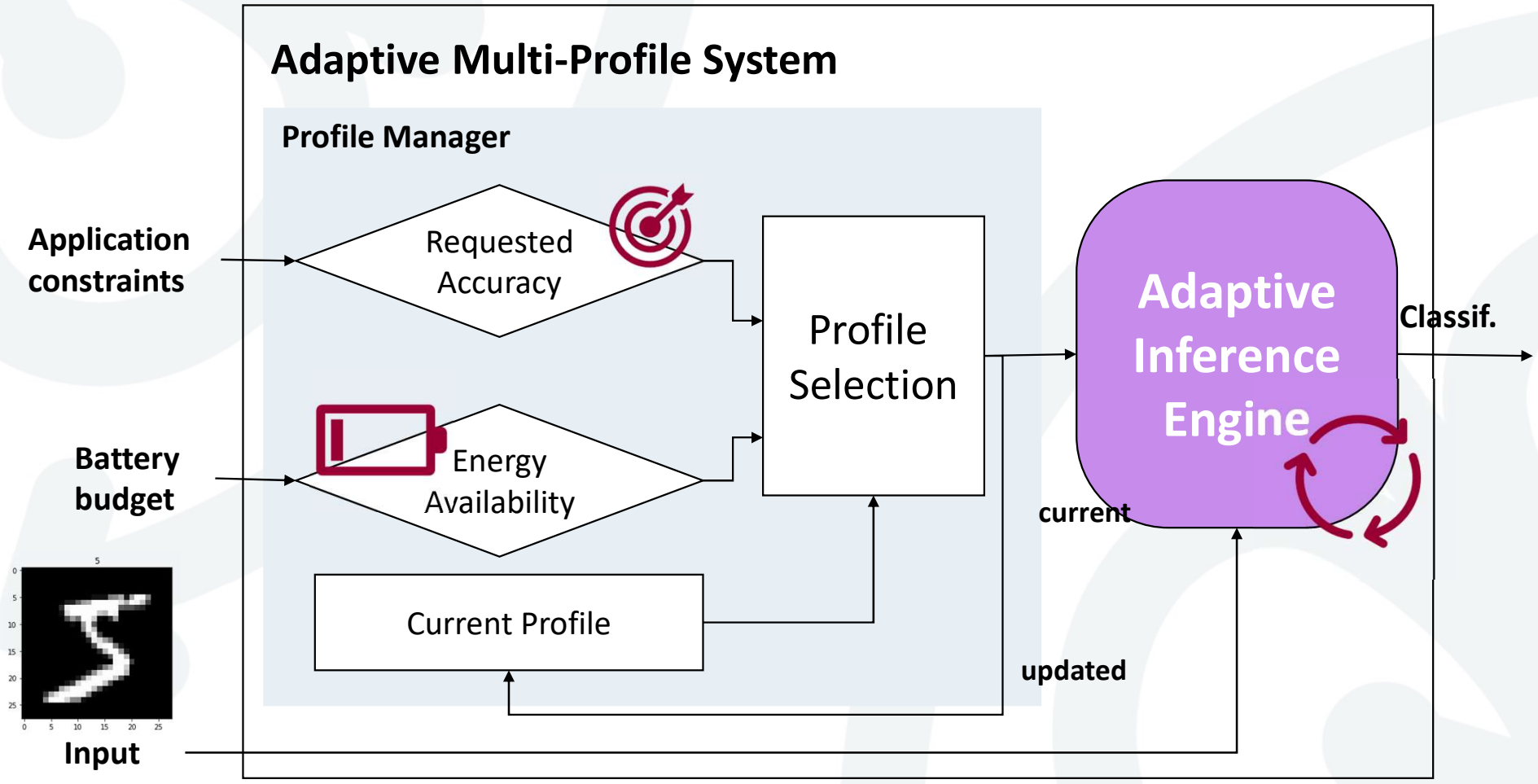
Results: Execution trade-offs



Ax_Wy:

- **x** is the number of bits used for representing **activations**;
- **y** is the number of bits used for representing **weights**;

Towards Adaptivity: MYRTUS Approach



Bibliography

1. Vivienne **Sze**, Yu-Hsin Chen, Tien-Ju Yang, and Joel S Emer. “Efficient processing of deep neural networks: A tutorial and survey.” In: Proceedings o
2. Marco **Cococcioni**, Federico Rossi, Emanuele Ruffaldi, and Sergio Saponara. “A lightweight posit processing unit for RISCV processors in deep neural network applications.” In: IEEE Transactions on Emerging Topics in Computing 10.4 (2021), pp. 1898– 1908. f the IEEE 105.12 (2017), pp. 2295– 2329.
3. Angelo **Garofalo**, Manuele Rusci, Francesco Conti, Davide Rossi, and Luca Benini. “PULP-NN: accelerating quantized neural networks on parallel ultra-low-power RISC-V processors.” In: Philosophical Transactions of the Royal Society A 378.2164 (2020), p. 20190155.
4. Clément Farabet, Cyril Poulet, Jefferson Y Han, and Yann LeCun. “**Cnp**: An fpga-based processor for convolutional networks.” In: 2009 International Conference on Field Programmable Logic and Applications. IEEE. 2009, pp. 32–37.
5. Yijin Guan, Hao Liang, Ningyi Xu, Wenqiang Wang, Shaoshuai Shi, Xi Chen, Guangyu Sun, Wei Zhang, and Jason Cong. “**FPDNN**: An automated framework for mapping deep neural networks onto FPGAs with RTL-HLS hybrid templates.”
6. Paolo Meloni, Daniela Loi, Gianfranco Deriu, Marco Carreras, Francesco Conti, Alessandro Capotondi, and Davide Rossi. “Exploring **NEURAGHE**: A Customizable Template for APSoCbased CNN Inference at the Edge.” In: IEEE Embedded Systems Letters PP (Oct. 2019), pp. 1–1. doi: 10.1109/LES.2019. 2947312.
7. Thea Aarrestad, Vladimir Loncar, Nicolò Ghielmetti, Maurizio Pierini, Sioni Summers, Jennifer Ngadiuba, Christoffer Petersson, Hampus Linander, Yutaro Iiyama, Giuseppe Di Guglielmo, et al. “Fast convolutional neural networks on FPGAs with **hls4ml**.” In: Machine Learning: Science and Technology 2.4 (2021), p. 045015.
8. Nicholas J Fraser, Yaman Umuroglu, Giulio Gambardella, Michaela Blott, Philip Leong, Magnus Jahre, and Kees Vissers. **FINN** “Scaling binarized neural networks on reconfigurable logic.” In: Proceedings of the 8th Workshop and 6th Workshop on Parallel Programming and Run-Time Management Techniques for Many-core Architectures and Design Tools and Architectures for Multicore Embedded Computing Platforms. 2017, pp. 25–3
9. Deep learning Processor Unit (**DPU**) designed for the Zynq® UltraScale+™ MPSoC, DPUCZDX8G for Zynq UltraScale+ MPSoCs Product Guide (PG338). https://docs.amd.com/r/en-US/pg338-dpu?tocId=3xsG16y_QFTWvAJKHbisEw
10. **Mittal**, Sparsh. "A survey of techniques for approximate computing." ACM CSUR 48.4: 1-33 (2016).
11. **Hubara**, I., Courbariaux, M., Soudry, D., El-Yaniv, R., & Bengio, Y. (2016). Binarized neural networks. *Advances in neural information processing systems*, *29*.
12. **Gholami**, Amir, et al. "A survey of quantization methods for efficient neural network inference." *arXiv preprint arXiv:2103.13630* (2021).
13. Pappalardo, A., Umuroglu, Y., Blott, M., Mitrevski, J., Hawks, B., Tran, N., ... & Duarte, J. (2022). **Qonnx**: Representing arbitrary-precision quantized neural networks. *arXiv preprint arXiv:2206.07527*.
14. **Sau**, Carlo, et al. "The Multi-Dataflow Composer tool: An open-source tool suite for optimized coarse-grain reconfigurable hardware accelerators and platform design." *Microprocessors and Microsystems* 80 (2021): 103326
15. Palumbo, Francesca, et al. “**MYRTUS**: Multi-layer 360° dYnamic orchestration and interopeRable design environmenT for compute-continuum”, To appear, https://doi.org/10.1145/3637543.3654618
16. **Ratto**, F., Máinez, Á.P., Sau, C. et al. An Automated Design Flow for Adaptive Neural Network Hardware Accelerators. J Sign Process Syst 95, 1091–1113 (2023). https://doi.org/10.1007/s11265-023-01855-x
17. **Manca**, F., Ratto, F., Palumbo, F. ONNX-to-Harware Design Flow for Adaptive Neural-Network Inference on FPGAs, Proceedings of the XXIV SAMOS International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation, June 29 - July 4, 2024 (To appear)

MYRTUS has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101135183.