AI for streaming readout: an architectural perspective

Roberto Ammendola

INFN Roma Tor Vergata

ro@infn.it

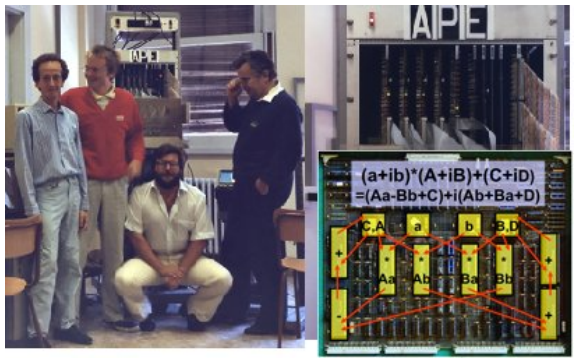
AI4EIC - Williamsburg - 13/10/22

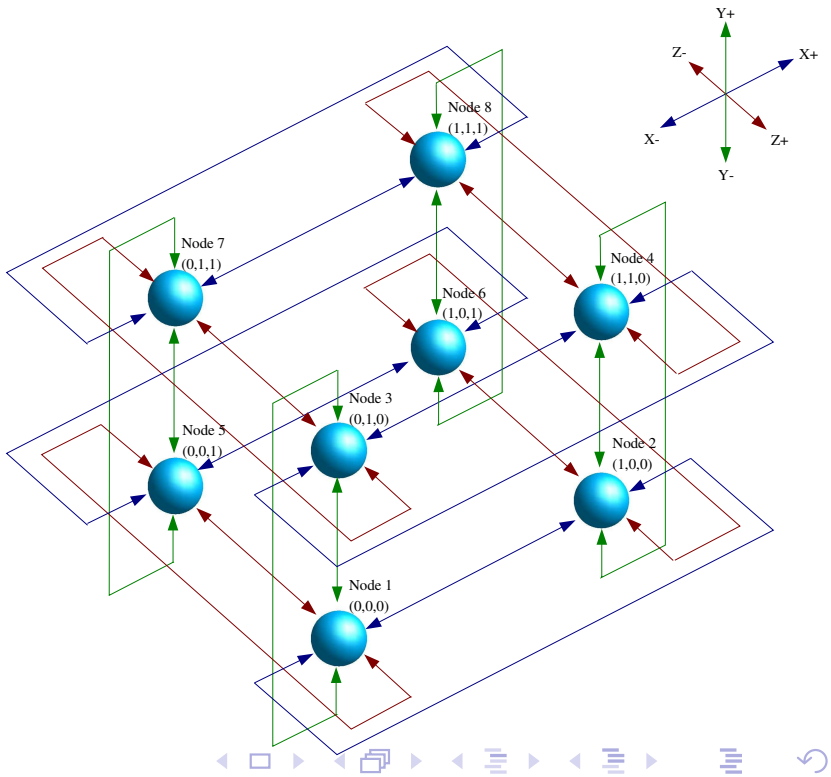
A few words about my research group APE (INFN)

Array Processor Experiment is a 30+ years old supercomputing project based on two main pillars:

Arithmetic engine tailored for physics application (LQCD, . . .)

Custom network to exploit communication pattern peculiarities





Motivations

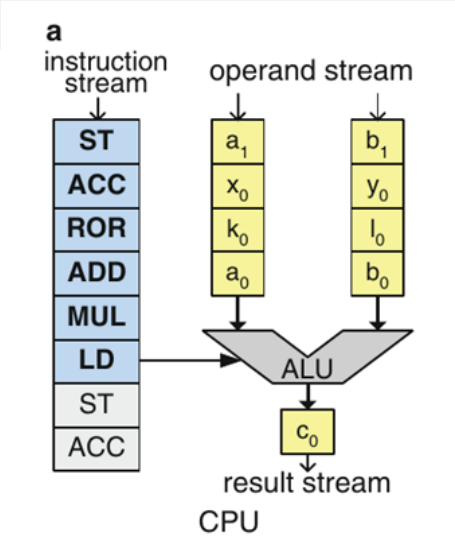
- an architectural/hardware point of view: computing engines and network are still key technologies when it comes to "new" algorithms (such AI/ML/DL) and high data throughput.
- we'll see what main technology vendors are proposing.
- some of the new ideas within the HEP community that can have an architectural impact on streaming readout systems
- briefly showing APEIRON, a framework proposal we are working on for building heterogeneous real-time systems

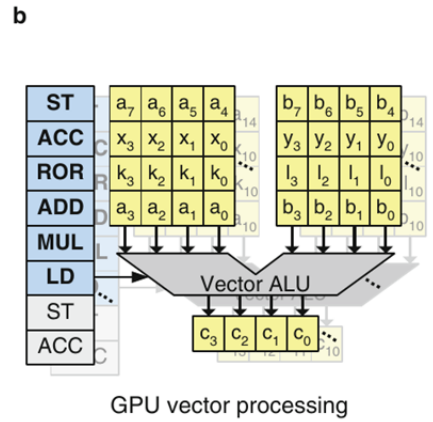
Introducing FPGAs and Streaming readout

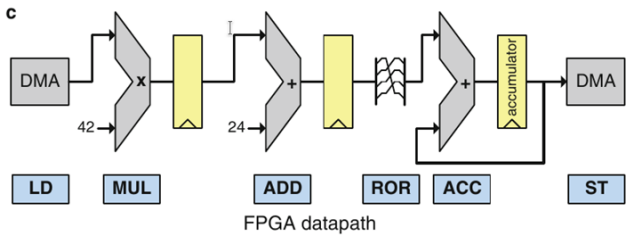
- Field programmable gate arrays (FPGAs) are chip made up of a finite number of predefined resources with programmable interconnects to implement reconfigurable digital circuits and I/O blocks.
- FPGAs features specialized blocks for arithmetics (DSP), memory and high speed transceivers.
- We have learned to use programmable devices in the years in triggered DAQ systems.
- We mainly needed two features: reduce data on selected subsystems in order to generate trigger signals, and store as much data as possible on the detector side waiting for triggers.
- In triggered DAQ most of the computational part is done at arrival.
- Thanks to the increasing bandwidth of interconnection technology and the computing power available on programmable devices, we are now able to design streaming DAQ systems.

Why FPGA are good in streaming readout

- Computing models have evolved in time with the increasing complexity of the underlying hardware.
- In the scalar model, an instruction stream is prepared for a single general purpose processing unit.
- In the SIMD/MIMD model multiple processing unit are able to work in parallel with the same/different instruction stream.
- Programmable devices exploit parallelism in a different way: the instruction stream is segmented in pipeline stages, where each stage performs a specialized operation.
- This model is described with different metrics than usual computing models: latency and throughput.
- Moreover, people working on AI algorithms such as deep learning neural networks have started representing those networks already in a pipelined model.

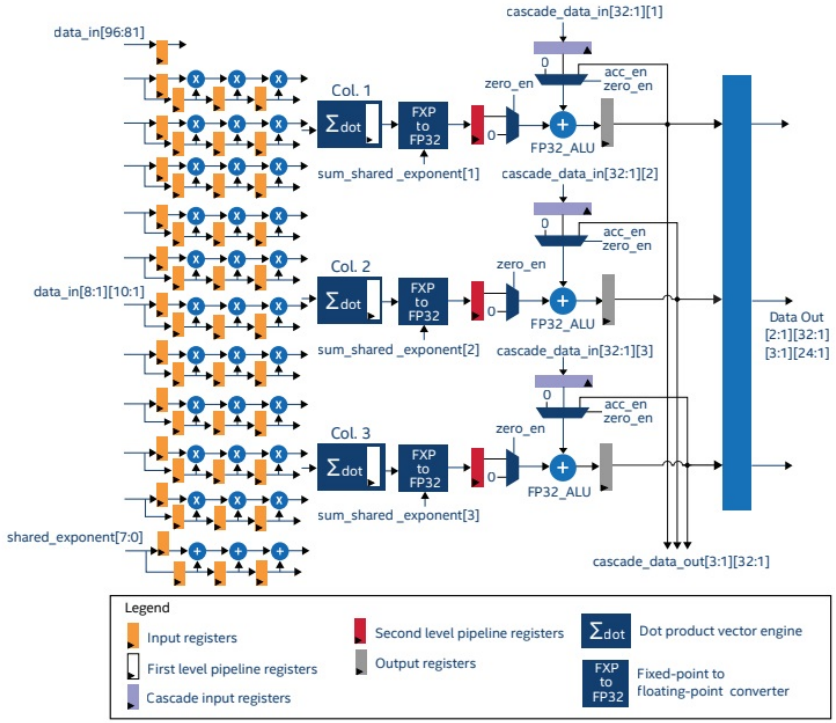






AI on FPGA, Intel trends (I)

- Intel announced Stratix 10 NX series and Agilex D series with a specific AI Tensor Blocks.
- Upgrade of DSP block for matrix operations with support for mixed precision.
- efficiency and performance numbers using 3,960 Tensor blocks for various precision targets

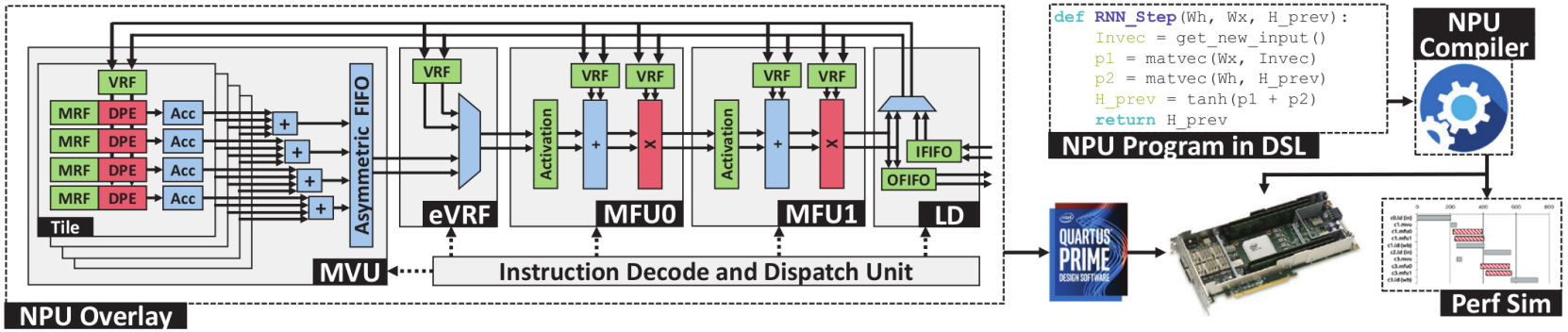


PRECISION	PERFORMANCE	EFFICIENCY
INT4	286 TOPS	2 TOPS/W
INT8	143 TOPS	1 TOPS/W
Block FP12	286 TFLOPS	2 TFLOPS/W
Block FP 16	143 TFLOPS	1 TFLOPS/W
@600 MHz Max Frequency		

A. Boutros et al., "Beyond Peak Performance: Comparing the Real Performance of AI-Optimized FPGAs and GPUs," 2020 International Conference on Field-Programmable Technology (ICFPT), 2020, pp. 10-19, doi: 10.1109/ICFPT51103.2020.00011.

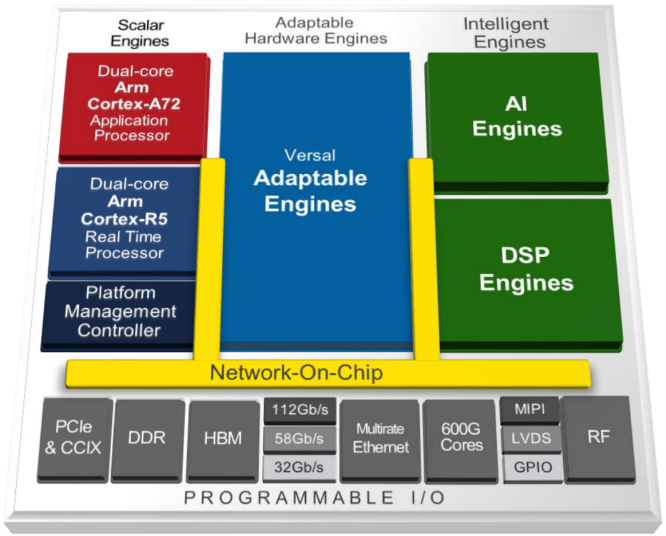
AI on FPGA, Intel trends (II)

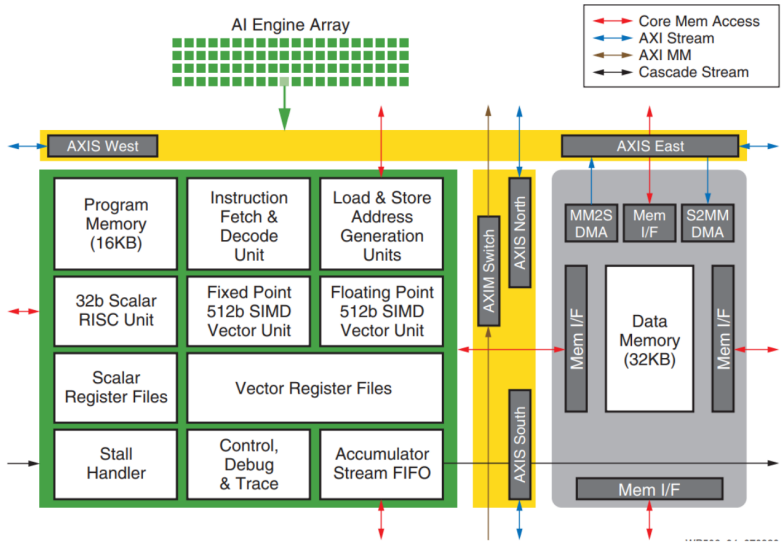
- a real performance issue
- the workloads are deployed using an implementation of a soft AI processor overlay called the Neural Processing Unit (NPU)
- a software toolchain allows you to program the FPGA without invoking any FPGA-specific Electronic Design Automation (EDA) tools.



AI on FPGA, Xilinx trends (I)

- new Versal architecture with AI engines, to replace soft DPU block
- featuring up to 400 AI engines, each with 32b RISC scalar processor
- 512b fixed-point and 512b floating-point vector processor with associated vector registers
- For each clock cycle, two scalar instructions, two vector reads, a single vector write, and a single vector instruction executed (6-way VLIW).
- Dedicated connectivity paired with DMA engines for scheduled data movement using connectivity between AI Engine tiles.

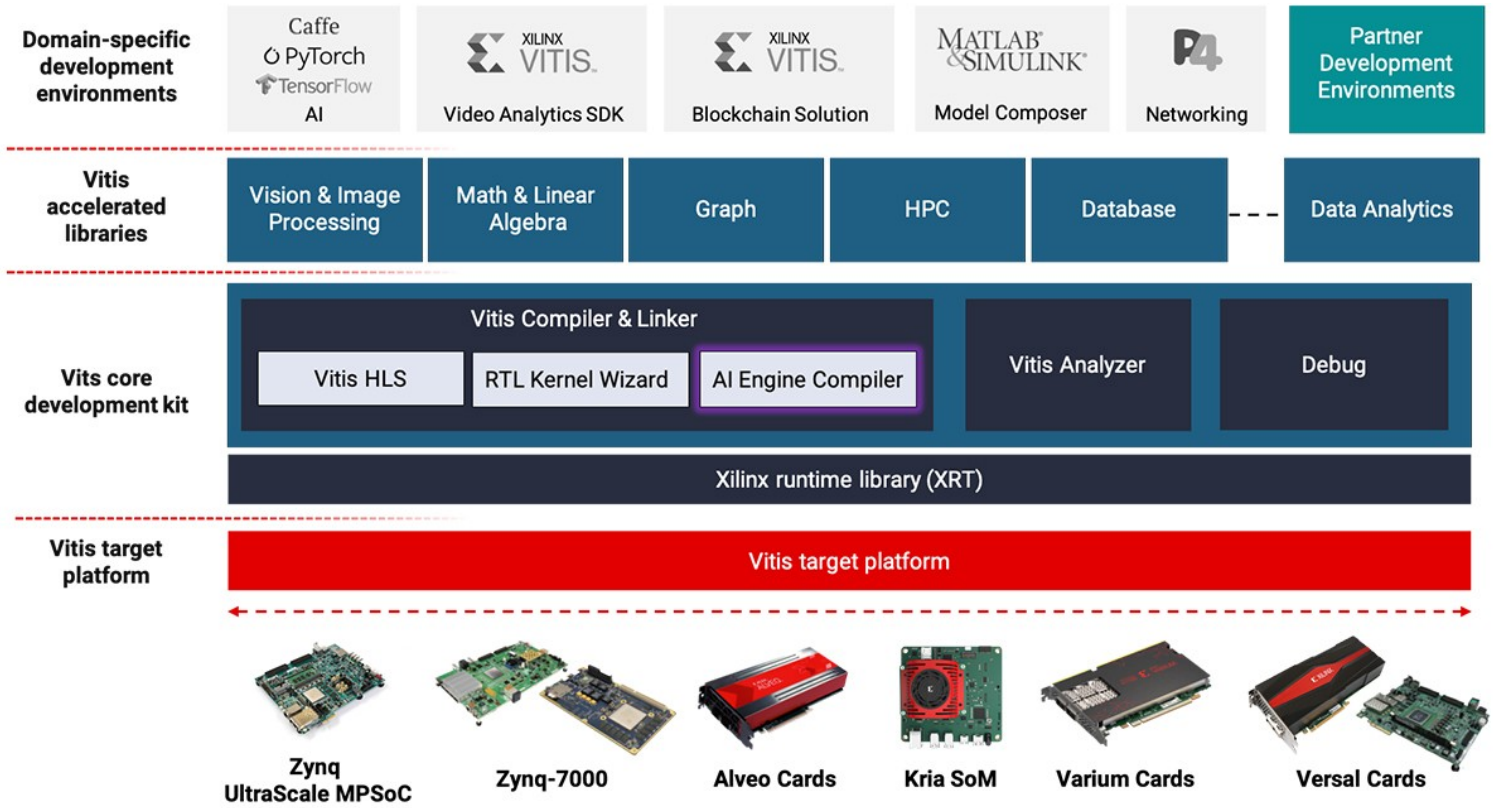




- B. Gaide, D. Gaitonde, C. Ravishankar, and T. Bauer, "Xilinx Adaptive Compute Acceleration Platform: Versal Architecture," in proceedings of the 2019 ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA'19). ACM
- Xilinx AI Engines and Their Applications (WP506)

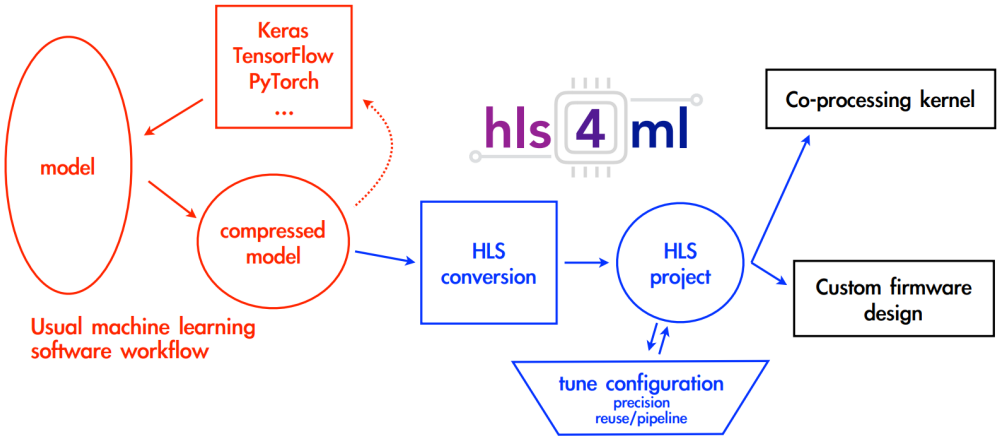
AI on FPGA, Xilinx trends (II)

- Promised a tremendous peak performance, but need to rely on Vendor Tool to use it.
- Are AI Engines good for real time applications?



Metodology improvements with HLS tools

- Traditionally design tools foresee digital hardware deployment through hardware description languages.
- Now FPGA are accessible with high level synthesis tools, enabling C/C++ code to be directly targeted into devices
- The HEP community is developing an independent tool (HLS4ML) to map ML algorithms (described in high level languages) on FPGA (or ASIC).
- Very active and growing community.



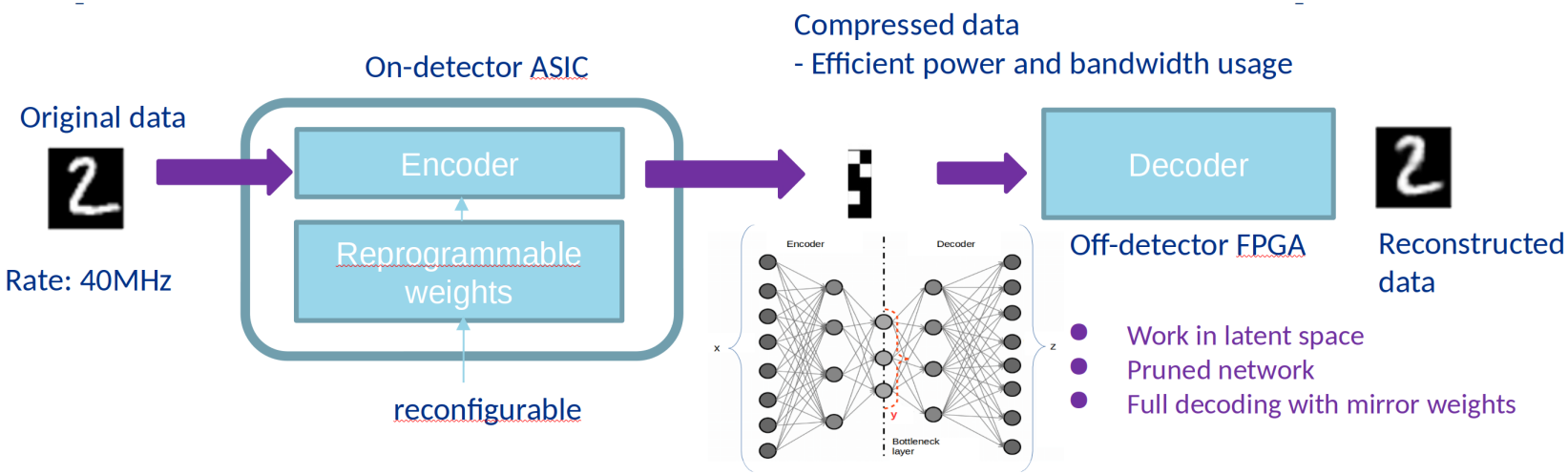
Architectures/Toolkits	Keras/TensorFlow/QKeras	PyTorch	ONNX
MLP	supported	supported	supported
Conv1D/Conv2D	supported	in development	in development
RNN/LSTM	in development	in development	in development

Duarte et al., "Fast inference of deep neural networks in FPGAs for particle physics", JINST 13 P07027 (2018), arXiv:1804.06913.

State-of-art architectural trends (I)

ML technique are getting closer to the detectors electronics:

- Data compression in order to reduce bandwidth demand on detector side
- Extract and use high level feature as interpretable observables

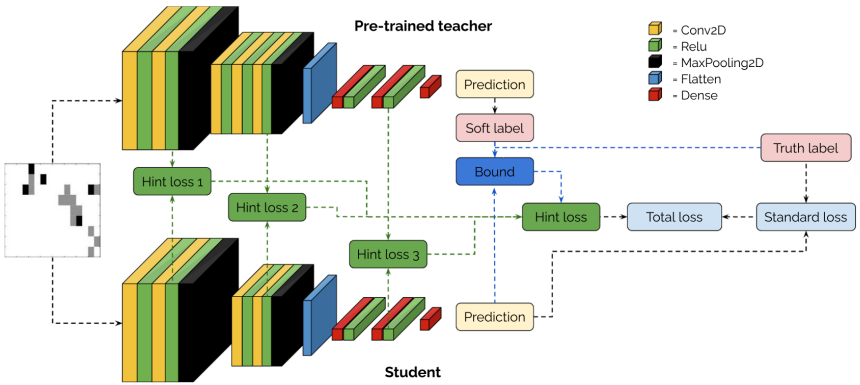


Di Guglielmo, Giuseppe, et al. "A reconfigurable neural network ASIC for detector front-end data compression at the HL-LHC." IEEE Transactions on Nuclear Science 68.8 (2021): 2179-2186.

State-of-art architectural trends (II)

Algorithms should adapt to hardware microarchitecture: higher efficiency when using pre-defined AI blocks.

- Knowledge Distillation is a recent technique to deploy large DL models on lighter devices
- Transfer knowledge learned by a larger neural network pre-trained for the same task to a smaller and quantised model
- Used in single muon trigger application in ATLAS
- Obtained a reduction on size of the model of a factor 100 with only a limited reduction in performance



FPGA resource occupation

Table 3 Percentage occupancy relative to the total FPGA available resources (model xcvu13p-fhga2104-2L-e [14])

Model (9 × 16)	BRAM	DSPs	FF	LUT
Teacher (%)	20.9	258.0	69.4	15.3
Student 32 bit (%)	3.2	31.0	8.4	2.7
QStudent 4 bit (%)	0.2	0.05	0.4	1.7

Inference time per event on FPGA
Xilinx Ultrascale+ XCV13P

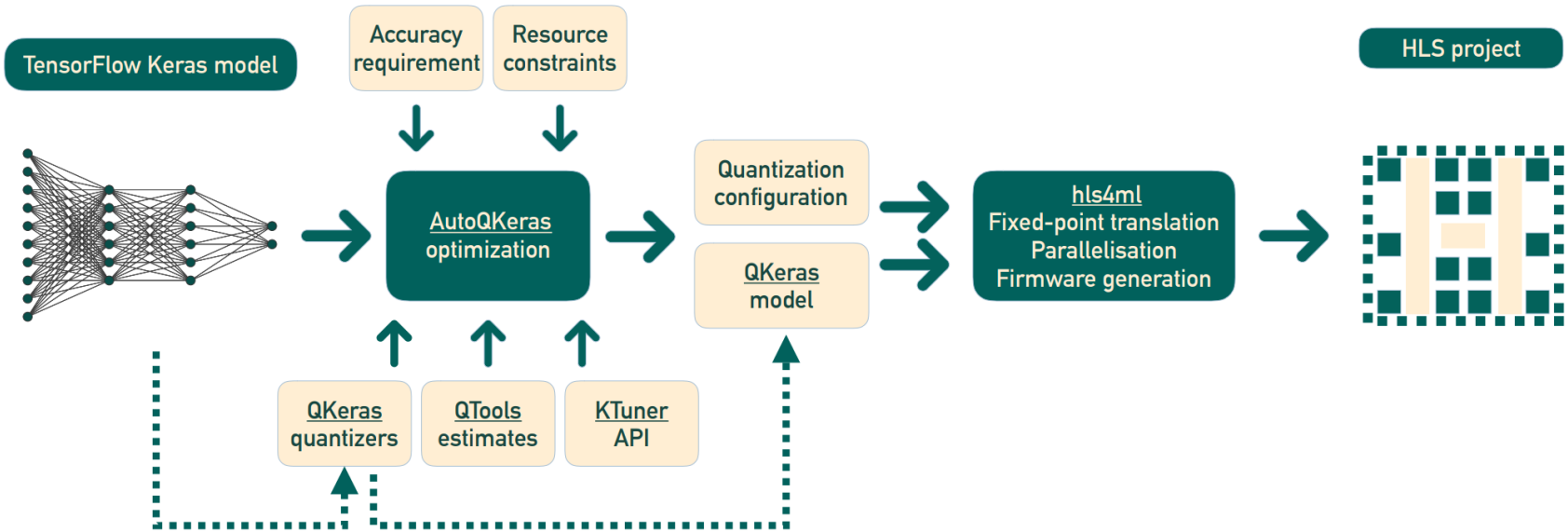
- Teacher fp32: 5 ms (Tesla V100 GPU)
- Student 4 bit: 438 ns (hls4ml)
- Student 4 bit: 84 ns (our VHDL implementation)

Francescato, S., Giagu, S., Riti, F. et al. Model compression and simplification pipelines for fast deep neural network inference in FPGAs in HEP. Eur. Phys. J. C 81, 969 (2021).

State-of-art architectural trends (III)

Compression by quantization: Google Qkeras

- Quantization-Aware Training approach means that the quantization is applied before the training of the model.
- full integration of QKeras layers in the hls4ml library
- in two simple steps, Keras models can be translated into ultra-compressed, highly parallel FPGA firmware.



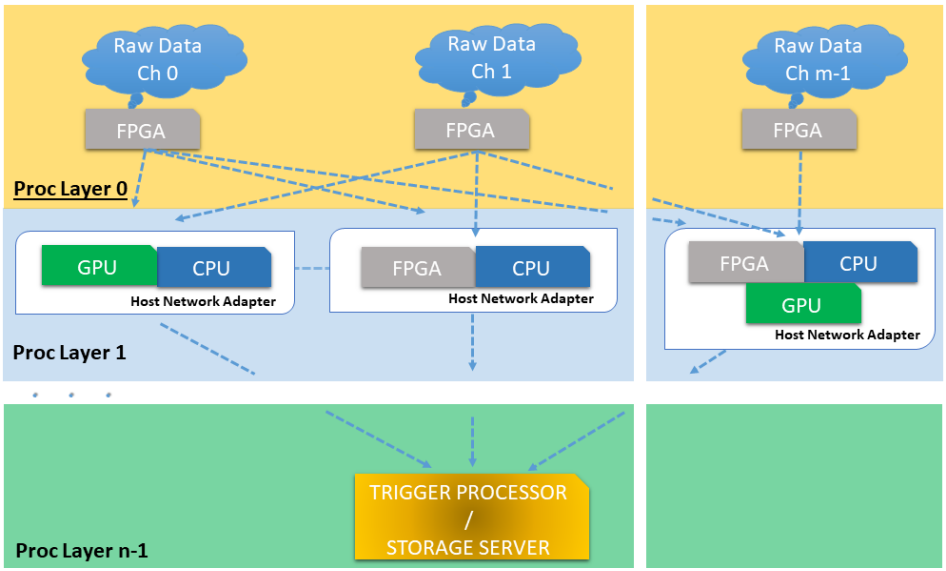
"Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors", Nature Machine Intelligence (2021), https://www.nature.com/articles/s42256-021-00356-5

INFN APEIRON Project

- stands for: Abstract Processing Environment for Intelligent Read Out systems based on Neural networks
- It is an INFN 5th commission 2019-2022 project.
- Started from the experience of NaNet project at NA62 CERN experiment.
- NaNet was a first use of data processing on a stream before L0 trigger, leveraging the GPU Direct capability we developed earlier.
- A readout system of an experiment is composed by many sub-detectors with many data sources.
- A low-latency (sub-microsecond), modular and scalable network infrastructure is needed.
- Configurable in terms of channels, topology and size.

- Ammendola, R., et al. "NaNet-10: a 10GbE network interface card for the GPU-based low-level trigger of the NA62 RICH detector." Journal of Instrumentation 11.03 (2016): C03030.
- Ammendola, R., et al. "Progress report on the online processing upgrade at the NA62 experiment." Journal of Instrumentation 17.04 (2022): C04002.

APEIRON Architecture



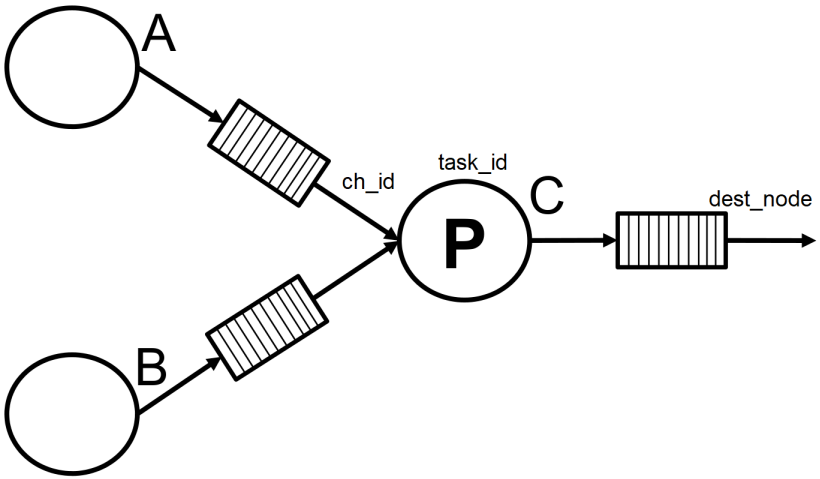
- Distributed online processing on heterogeneous computing devices in n subsequent layers.
- Exploit the specialization of modern computing devices.
- Keep the definition of processing and communication the more abstract and device independent as possible.
- Features extraction will occur in the first NN layers on FPGAs: reduced precision and/or DNN compression techniques are to be studied and implemented.

Dataflow programming model

Programming Model based on Kahn Process Networks (KPNs):

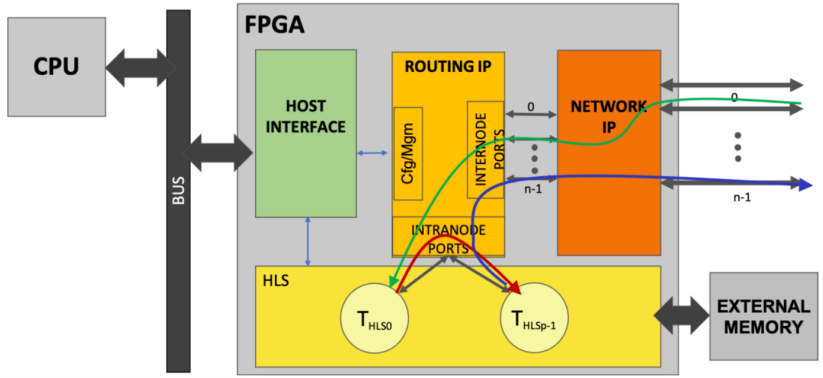
- Determinism: for the same input history the network produces exactly the same output
- Monotonicity: partial information of the input stream to produce partial information of the output stream
- Processes can run concurrently and synchronize through blocking read on input channels

Expressing tasks in this model allows to map them on heterogenous computing nodes and adapt underlying network among them.
The building block of such a distributed/parallel system is an FPGA card with I/O and computing capability.



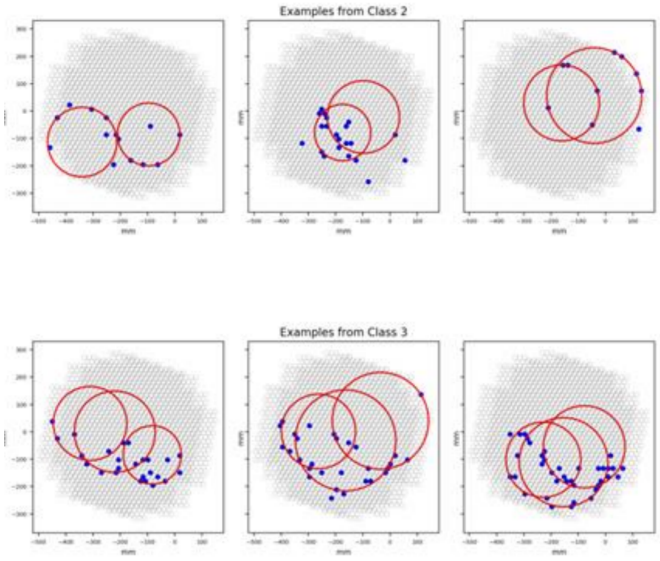
APEIRON C++ HLS API

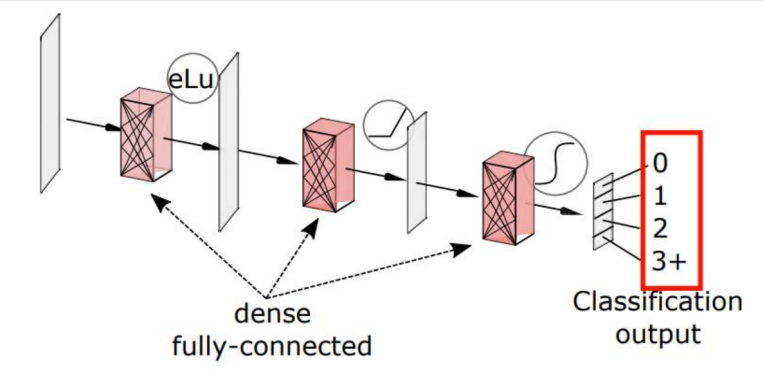
- **send**(msg, size, dest_node, task_id, ch_id)
- **receive**(ch_id)



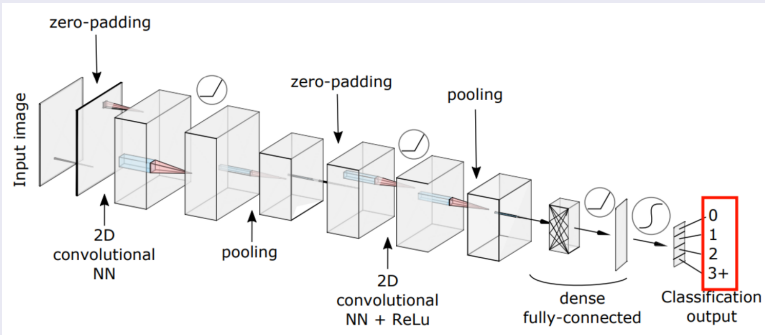
APEIRON Use case: PPID at CERN NA62 RICH

- The number of Cherenkov rings is a good candidate to improve L0 decisions: can we achieve required throughput with a good accuracy?
- Nominal event rate at L0: 10MHz
- Work in progress on two different models





1) fully connected network: less accurate, low footprint, low latency, high throughput

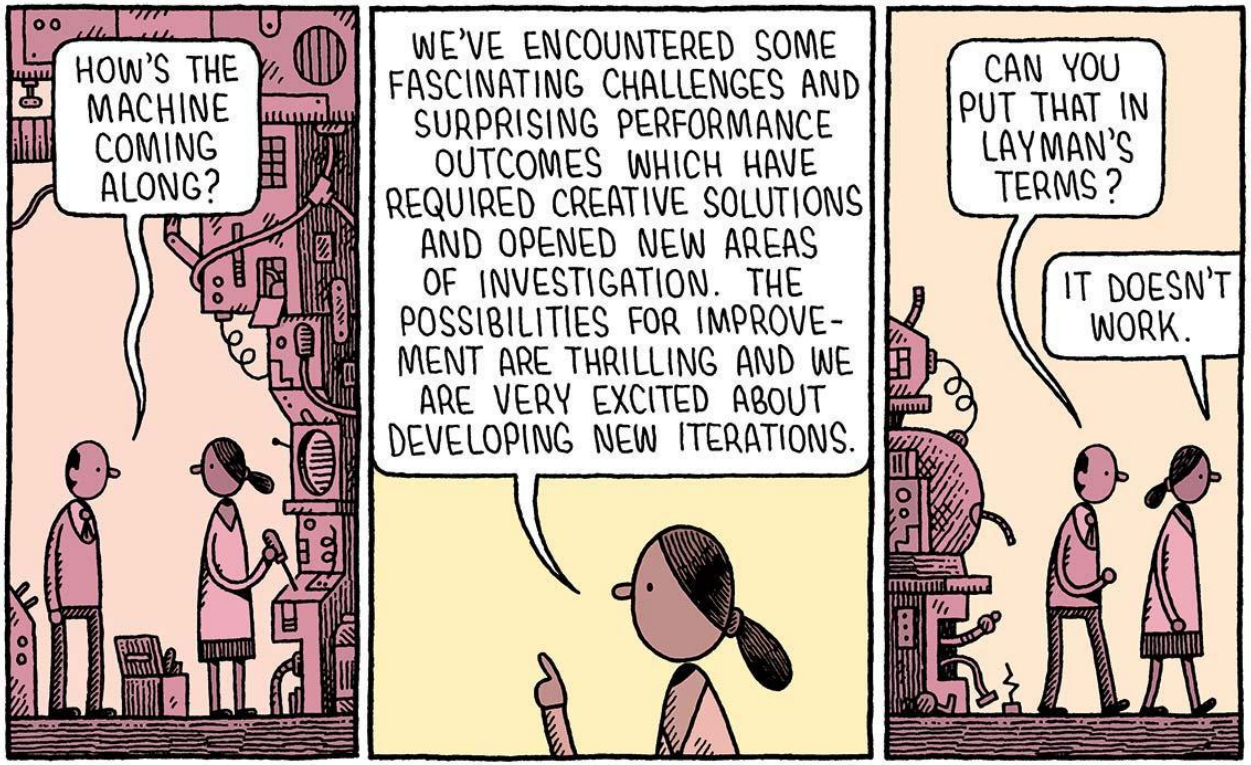


2) convolutional network: more accurate, low footprint, higher latency, low throughput

Final words

- FPGA are getting paradoxically easier to use thanks to HLS programming
- ... but more difficult to exploit them well (efficiently)
- sometimes the good old methods work best, but how long will it be supported?
- need to work also on algorithm side (i.e. knowledge distillation) to adapt/exploit hardware
- when scaling to many heterogeneous computing elements a more abstract framework is needed (i.e. APEIRON)
- more details on APEIRON at ACAT 2022 by A. Lonardo and proceedings will be available

Thanks for your attention.



AI for streaming readout: an architectural perspective

Summary of talk

- Streaming readout architecture in general and DL algorithm in particular fit well on programmable devices such as FPGAs
- Technologies (HW) and methodologies (SW) to use them are getting complex and complex (HLS4ML, AI Engines, ...)
- Some ideas can help riding technology: data compression on the edge to reduce bandwidth, knowledge distillation to obtain simpler networks, quantization aware training to exploit more efficient data types.
- A readout system could be composed of many processing nodes: we need a framework to distribute computing tasks and orchestrate them for real time processing.
- APEIRON project has been briefly presented: a system of multiple interconnected FPGAs with applications developed according to a dataflow programming model.
- We are able to map the directed graph of tasks and network channels with direct coupling among them.