Innovative Workflows Astro & Particle Physics (IWAPP)

APEIRON

***A**bstract **P**rocessing **E**nvironment for **I**ntelligent **R**ead-**O**ut systems based on **N**eural networks*

Luca Pontisso – APE Group (INFN Roma)

2021 March 9th

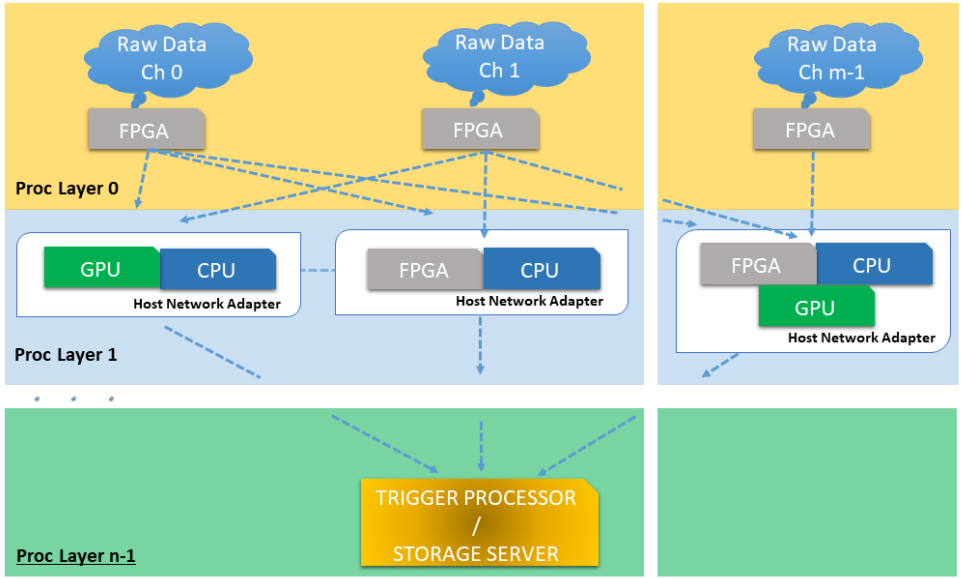
- Introduction to the APEIRON project
- FPGA a brief overview
- Physics use case: rings identification in NA62 RICH detector
- Example of heterogeneous computing node: GPU-RICH
- Overview of dataflow

- Neural Network on FPGA with High Level Synthesis details, deployment, workflow

Matteo's talk!

APEIRON

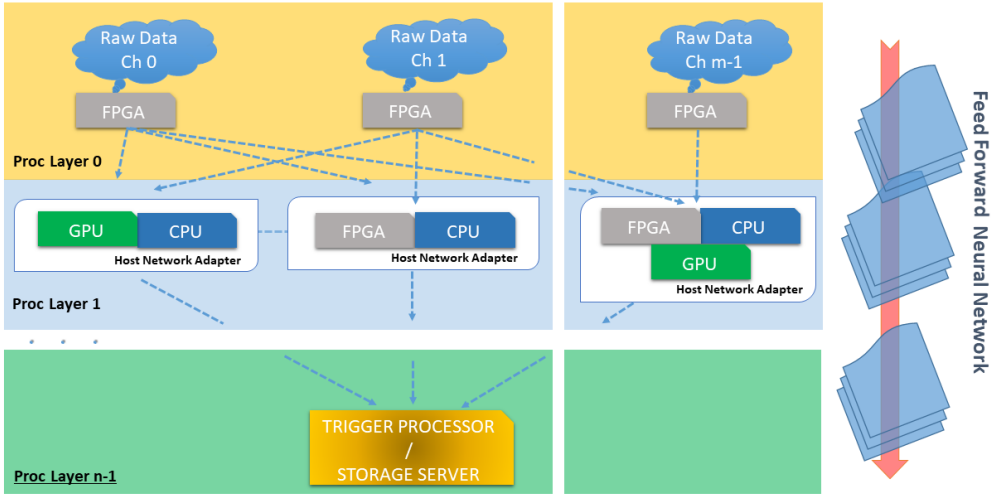
Abstract Processing Environment for Intelligent Read-Out systems based on Neural networks



- Input data from several different channels (data sources, detectors/sub-detectors).
- **Data streams** from different channels recombined through the processing layers using **a low-latency, modular and scalable network infrastructure** (configurable in number of channels, topology and size).
- 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.

APEIRON

Abstract Processing Environment for Intelligent Read-Out systems based on Neural networks



- Features extraction will occur in the first NN layers on FPGAs: reduced precision and/or DNN compression techniques are to be studied and implemented.
- More resource-demanding NN layers (e.g. CNN) implemented in subsequent processing layers.
- Classification produced by the NN in last processing layer (e.g. pid) will be input for the **trigger processor/storage online data reduction stage for triggerless systems**

FPGA a brief overview

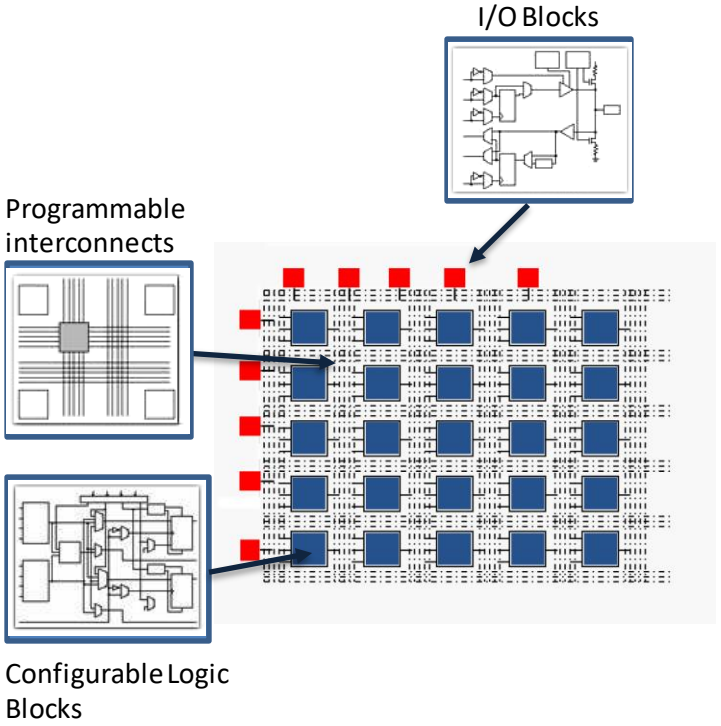
Field-programmable gate arrays (FPGAs)
chip made up of a finite number of predefined resources with programmable interconnects to implement a reconfigurable digital circuit and I/O blocks

FPGA adoption is driven by their flexibility, hardware-timed speed and reliability, and parallelism

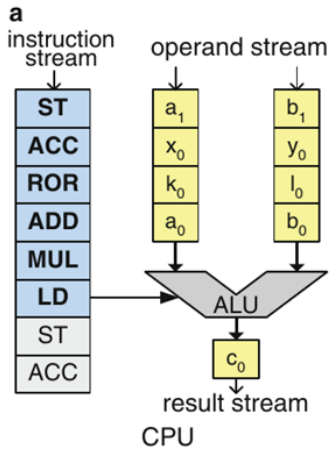
FPGA resources: CLB (flip-flops, LUTs), DSP and block RAM

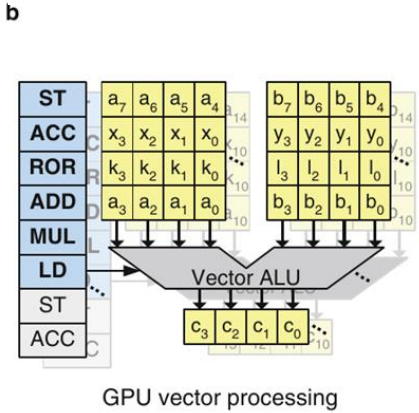
Development from hardware description languages (HDLs) such as VHDL and Verilog into a bitstream requires a different approach than usual software development

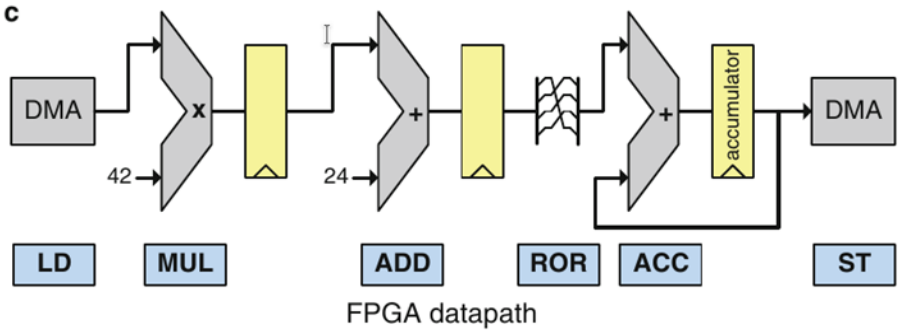
High-level synthesis (HLS) tools permit designers to work at a higher level of abstraction through



Dataflow



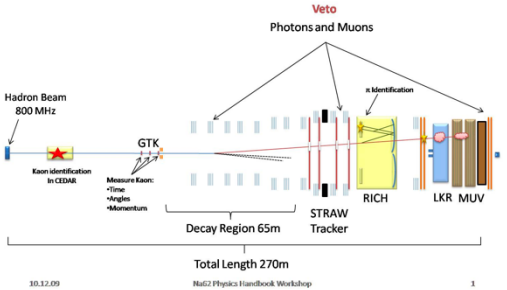




- Pipelined design produce one output for each clock
- The greater the number of pipeline stages, the greater the latency
- *Initiation interval:* Number of clock cycles before the function can accept new input data. The lower, the higher the throughput
- Once latency is overcome a pipelined design yields one output per clock cycle irrespective of the number of pipeline stages Increasing throughput through parallelism

NA62 Experiment

- Located at the CERN SPS
- Measure rare kaon decay:
  - $k \to \pi\nu\nu$ with $BR(k \to \pi\nu\nu) = (8.4 \pm 1.0) \times 10^{-11}$
- In 2017-18 data taking at ~60% of nominal intensity
- In 2021 plans to scale at 100% of nominal intensity

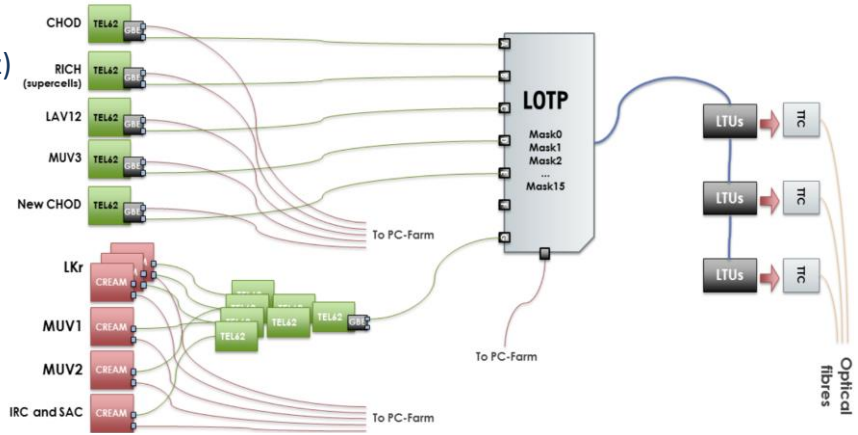


Multi-level trigger

Level-0 **(L0TP,** Level 0 Trigger Processor) HW trigger on FPGA (**10MHz**->1MHz)
Level-1 software trigger running in DAQ farm (1MHz –> 100kHz)
Level-2

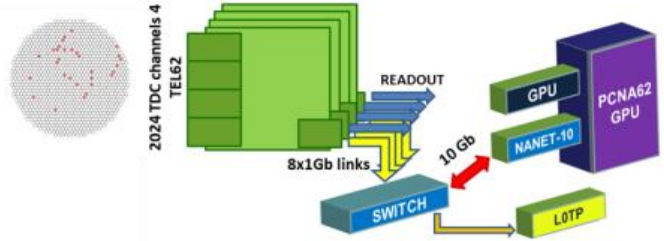
DAQ

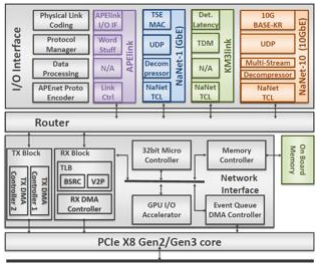
Data bursts are ~6s long
Some detectors primitives, generated from TEL62 readout boards, are sent to L0TP
L0TP generates trigger with **max latency of 1ms**
L2 trigger run over the complete event information.

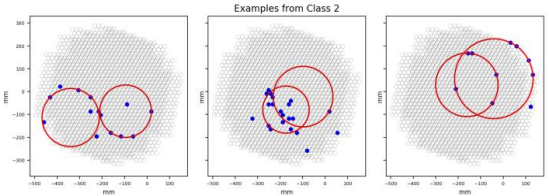


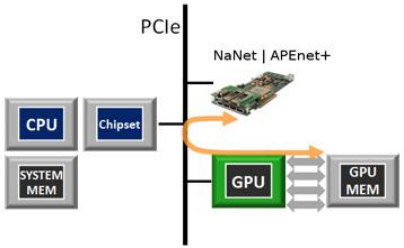
GPU-RICH Heterogeneous Processing node

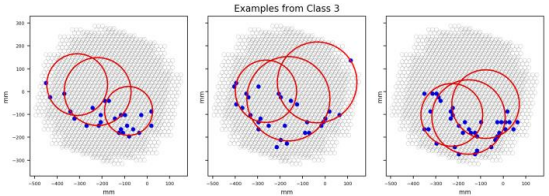
NaNet: *Heterogenous Computing Node* for HEP low level trigger











- UDP streams from 4 ReadOut boards (2048 PMT channels) with RICH primitives to the switch using 8x1GbE links.
- NaNet-10 FPGA-based NIC receives UDP RICH primitives streams over a 10GbE link (from the switch).
- NaNet-10 FPGA performs preprocessing on data streams:
  - decompression;
  - merging of events split on different streams;
  - change data alignment for GPU memory access on the 4kB staging buffer.
- NaNet-10 DMA writes over PCIe merged/re-aligned event data to a receive buffer in GPU memory.
- When the receive buffer is full (or the gathering timeout has expired), the CPU gets notified and launches a sequence of CUDA kernels.
- On-line RICH reconstruction algorithm based on histograms of hits distances
- Trigger primitives (n of rings and their type) are sent directly from GPU memory to L0TP

APEIRON use case

Partial Particle Id with NN on FPGA

Exploring different workflows ...

TensorFlow

model, weights

Custom Library C/C++ implementation

hls4ml

https://fastmachinelearning.org/hls4ml/

High-level *synthesis tools*

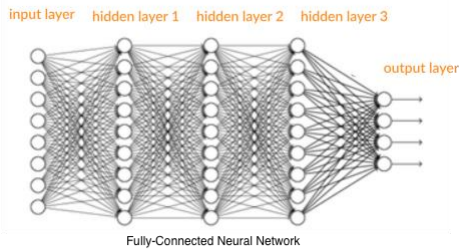
(Xilinx Vitis HLS)

Partial Particle Id with NN on FPGA

Starting with implementation on a single device as a first step to exploring distributed solutions

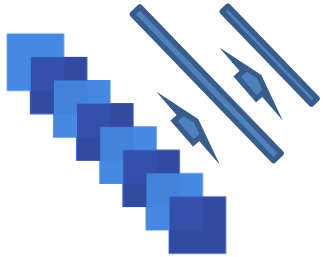
Exploring two different models ...

Multiple *Fully Connected Layers*



Convolutional and Fully Connected Layers

(work in progress)



Activity in collaboration with A. Ciardiello (Sapienza University and INFN Rome)

To keep in mind …

… when writing the model... some technological constraints

- Not all the features (layers, activation functions) available in machine learning package models are supported by translation tools
- High-level synthesis tools cannot transform all the C/C++ code into a register transfer level (RTL) implementation (e.g. malloc(), hard limit on unrolled loop iteration 16k)
- Resources on devices are not unlimited (in the thousands neurons ball park not millions even on a high end device)
- Choosing the right data representation also according to the FPGA distinctive features
- Fixed point representation mandatory but with loss of range and precision

Even experimenting with implementations of small NN models…

RiNNgs dense model

Fully Connected

Dense, 64, activation ELU
Dense, 16, activation RELU
Batch Normalization
Dense, 4

- Input: 64 hits per event
- Arch: 3 fully connected layers
- Output: 4 classes (0,1,2,3+ rings per event)
- Training: 80000 examples/class

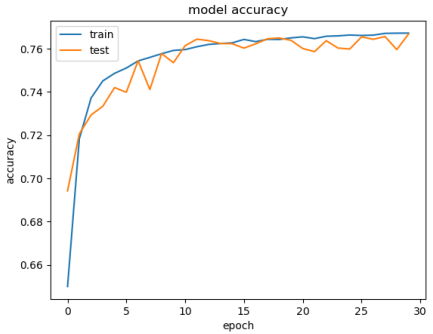
Which data representation for input…

Positional encoding
PMT 2048 bit [0001000….]

→ NO…. Facing HLS tools (VITIS) hard limit in unrolling loops

List of channels

→ OK



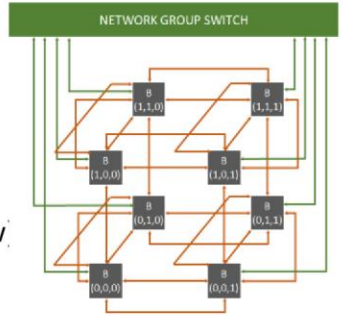
BACKUP

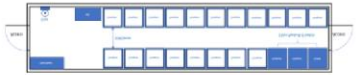
Low-latency, modular and scalable network

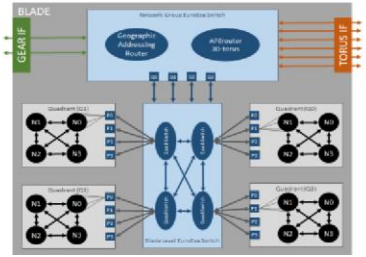
ExaNet is the EuroEXA approach for large-scale, multi-tiered interconnect.

ExaNet at a glance:

- Hybrid Torus Topology for inter-node communication
  - Quadrant Level → all to all
  - Blade Level → Dragonfly/Full Crossbar
  - Network Group → 3D Torus
  - Rack Level → Fat Tree
- Light-Weight Custom Communication Protocol (low latency/high bw)
- Reliability and QoS
  - traffic congestion control
  - end-to-end reliability guaranteed by retransmission
  - fault-awareness at system level
- RDMA offload
- Optimized communication libraries
  - API: user-space software stack
  - MPIpoint-to-point and collective







NetworkGroup (NG)
- 8 blades
- 3.2Tbps
- Torus topology

Rack
- 4 NG
- 6.4Tbps
- Fat Tree
- 110kW

System
- 2MW in a modular facility
- PUE 1.0x
- Heat Reuse Capable
- Low Cost Facilities

RiNNgs convolutional model

Convolutional

Conv2D, 8 filters (7,7), stride 5, activation ELU
Dense, 8, activation RELU
Batch Normalization
Dense, 4

- Input: PMT channels into image 49x49 pixels
- Arch: 1 CNN 8 filters, stride 5, filter kernel 7x7
- Output: 4 classes (0,1,2,3+ rings per event)
- Training: 20000 examples/class