RICH pattern recognition with FPGA: from NA62 to dRICH

Alessandro Lonardo

(INFN Roma1, APE Lab)

for the EIC_NET Roma1&ToV team

Giornate Nazionali EIC_NET 2024
27-28 Giugno 2024
Bologna

The NA62 Experiment at CERN SPS

- Measurement of the K⁺ decay:

  $BR(K^+ \rightarrow \pi^+ \nu \bar{\nu})$

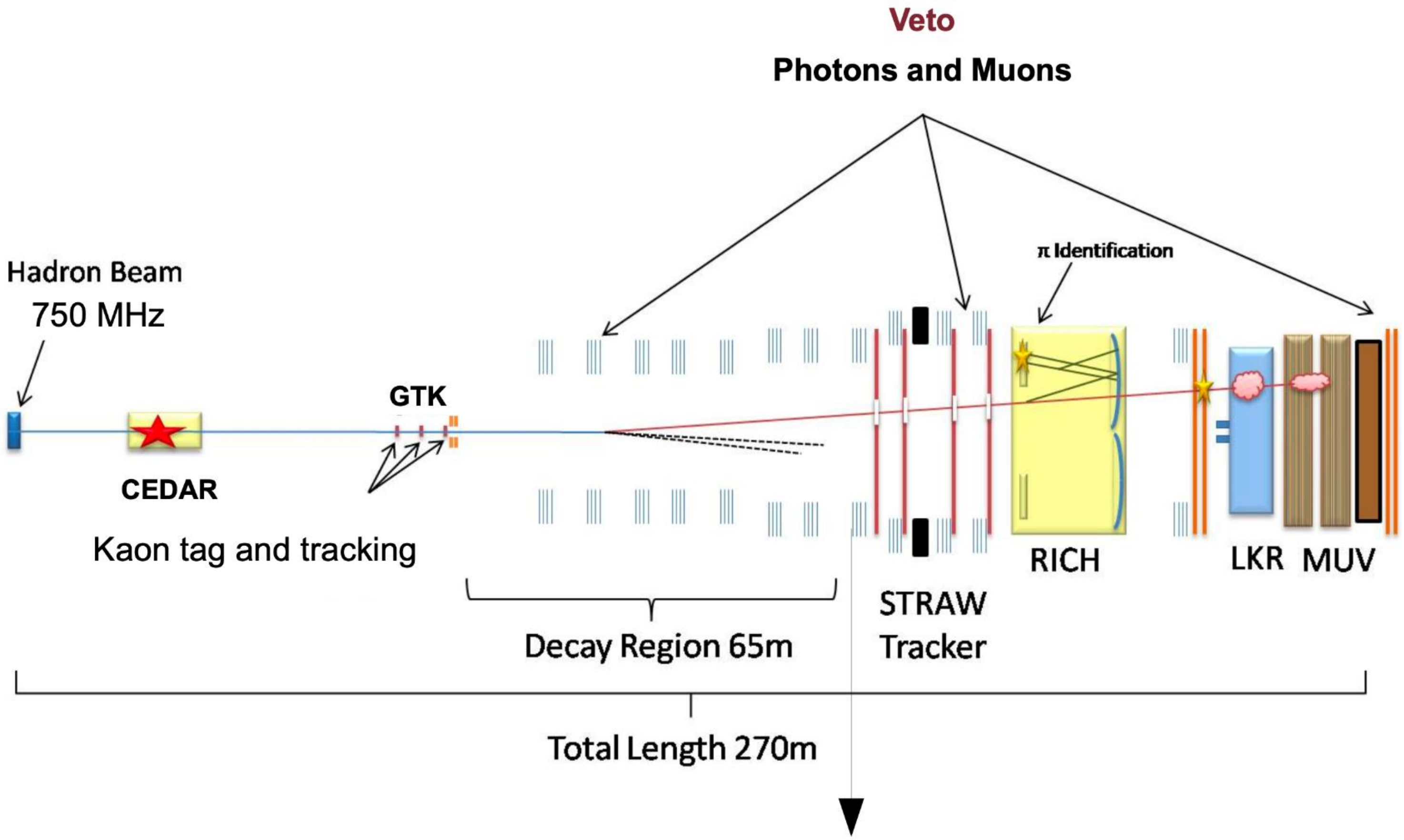
- **Ultra-rare** channel, **SM prediction**:

  $BR_{SM} = (8.60 \pm 0.42) \times 10^{-11}$

  Run I NA62 measurement:

  $BR_{NA62} = (10.6^{+4.0}_{-3.4}|_{stat} \pm 0.9_{syst}) \times 10^{-11}$

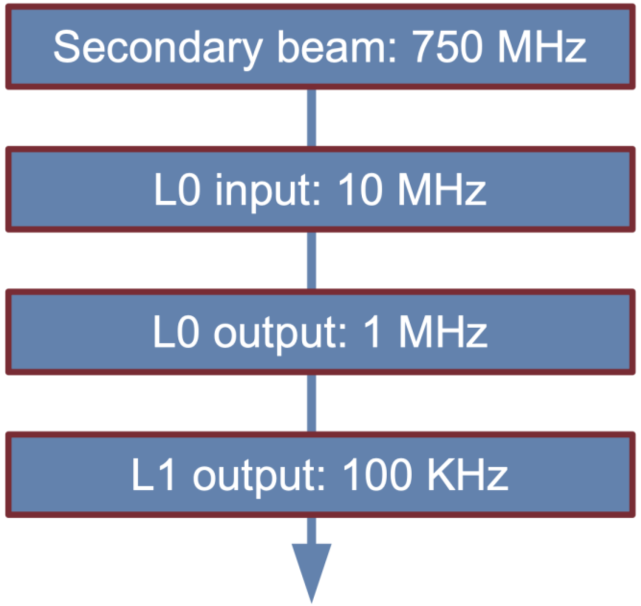
- **Fixed Target experiment**:
  75 GeV secondary hadron beam (6% kaons).

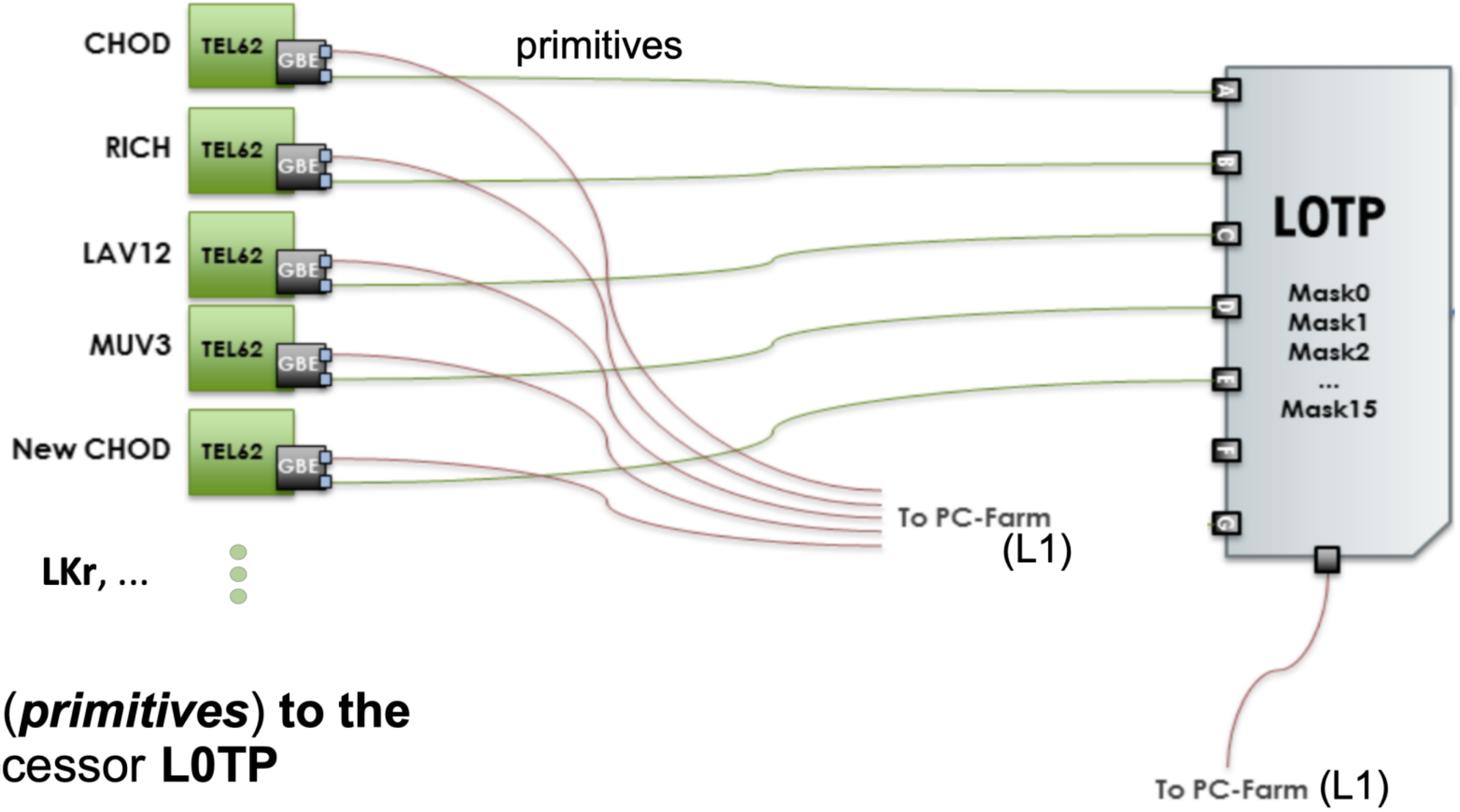


10 MHz event rate

Need highly selective filtering sytem

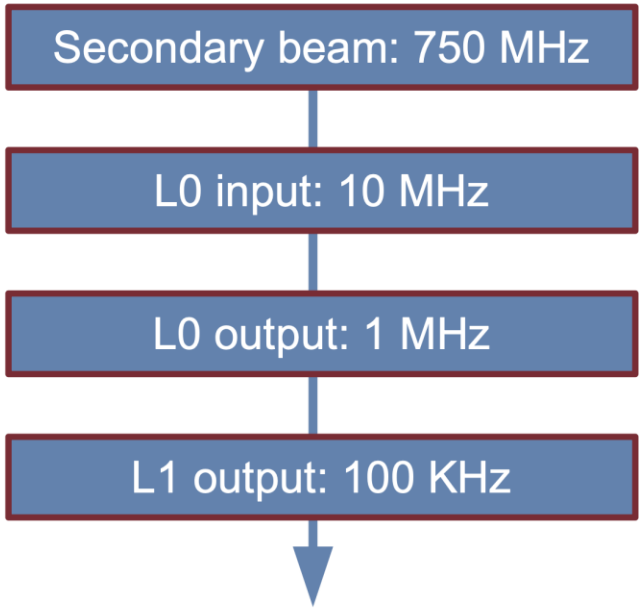
The NA62 Data Acquisition and Low Level Trigger

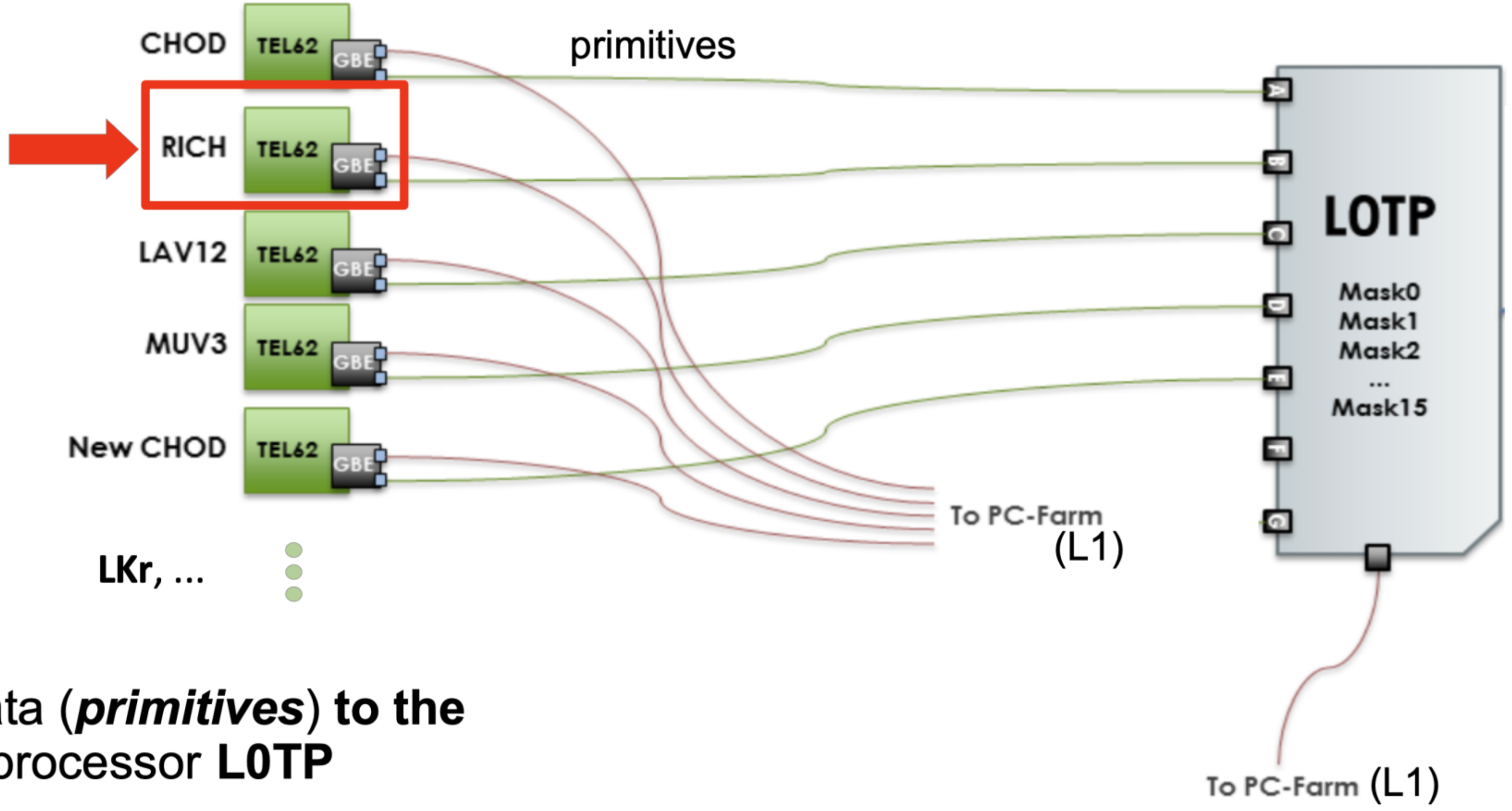




- Some detectors send raw data (***primitives***) **to the** FPGA-based level-0 trigger processor **L0TP**
- Primitives are generated from **TEL62 read out boards**
- L0TP **checks conditions** (**Masks**) to determine if an event should be selected and sent to L1
- Masks rely on the **physics information inside the primitives** (Energy, hit multiplicity, position, ...)

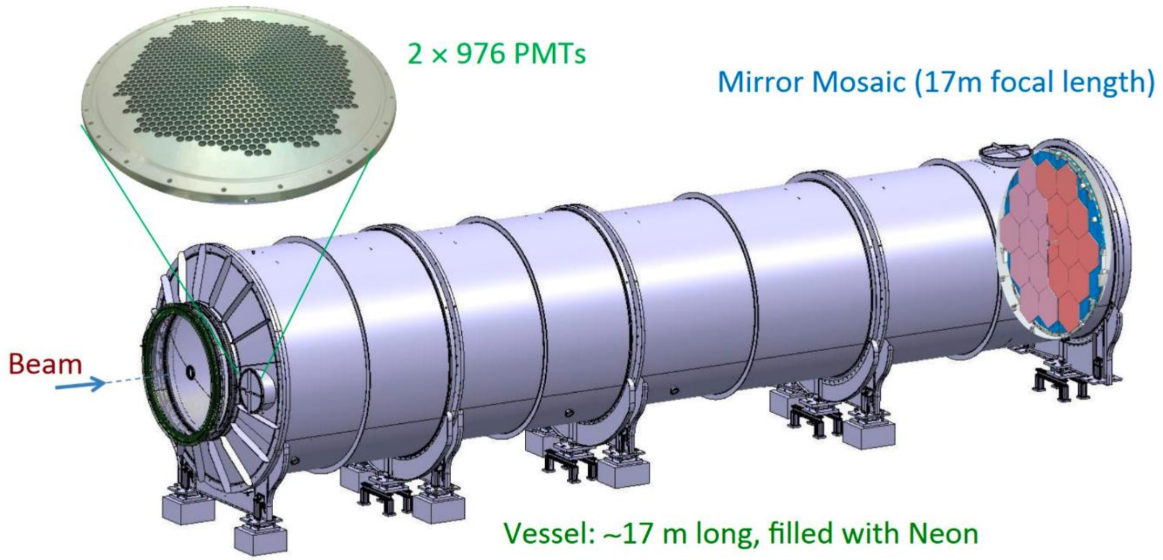
The NA62 Data Acquisition and Low Level Trigger

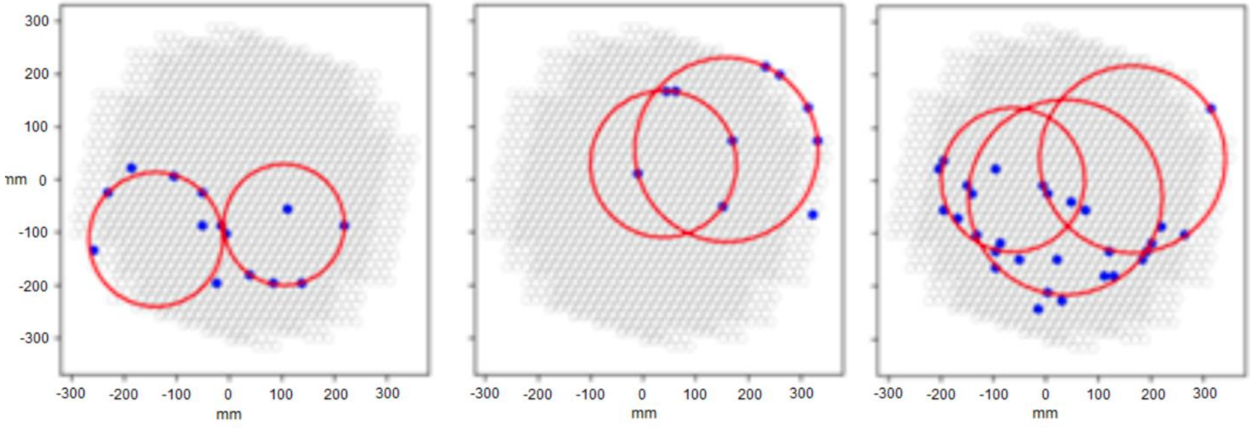




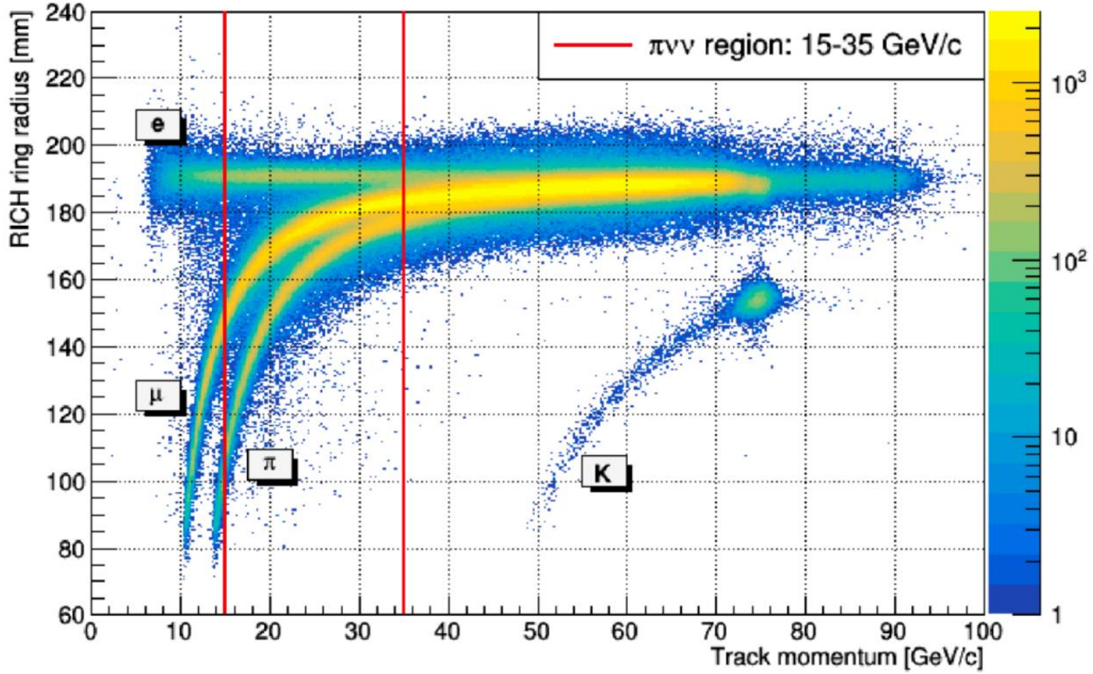
- Some detectors send raw data (***primitives***) **to the** FPGA-based level-0 trigger processor **L0TP**
- Primitives are generated from **TEL62 read out boards**
- L0TP **checks conditions** (**Masks**) to determine if an event should be selected and sent to L1
- Masks rely on the **physics information inside the primitives** (Energy, hit multiplicity, position, ...)

The NA62 Ring Imaging Cherenkov detector (RICH)

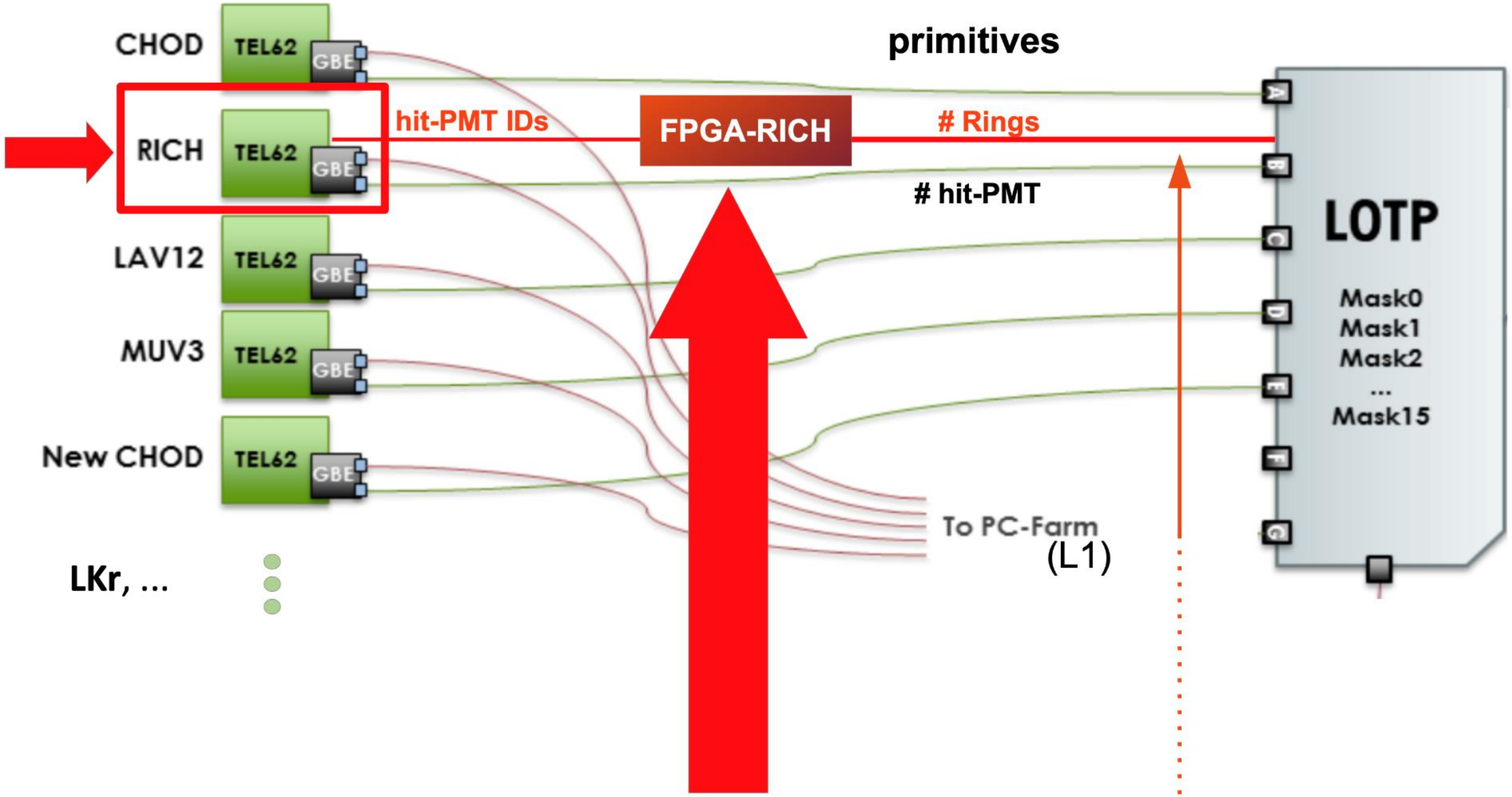

2 × 976 PMTs
Mirror Mosaic (17m focal length)
Beam
Vessel: ~17 m long, filled with Neon

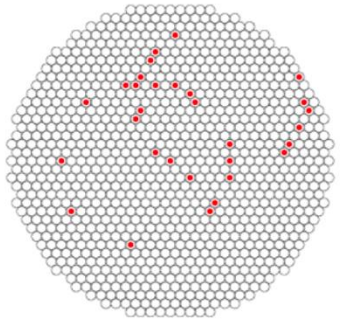


- During offline data analysis, it provides PID to distinguish between pions and muons from 15 to 35 GeV
- Uses the Cherenkov rings radius and track momentum
- **L0 primitives contain only number of HIT PMTs**

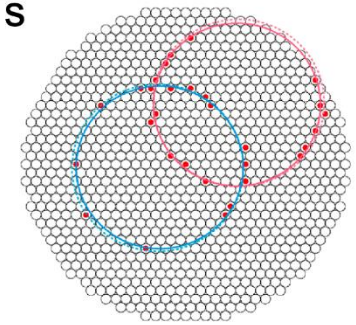


Smart Primitives: FPGA-RICH





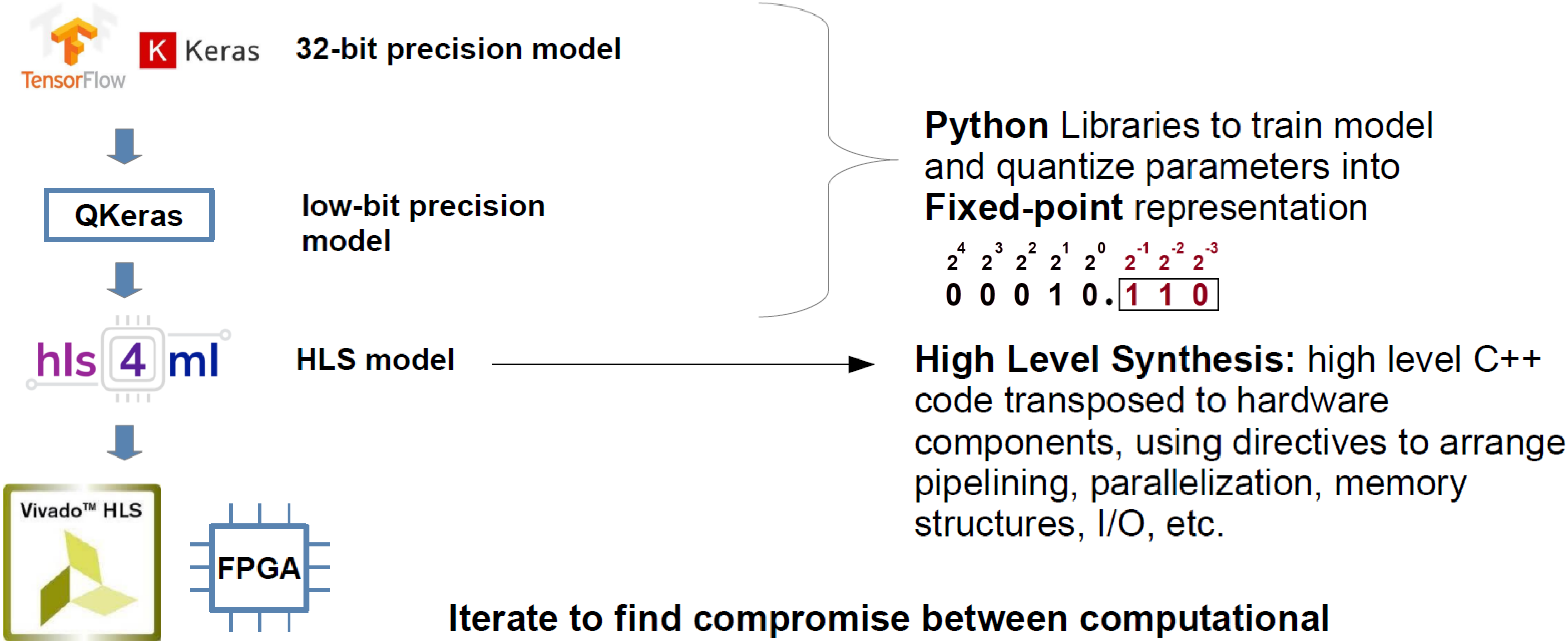
RICH primtives: # of hit-PMTs



FPGA-RICH primtives:
# of rings (0, 1, 2, 3+)
(more in the future, e.g. # of e^-)

FPGA-RICH: **reconstruct the rings geometry online** using an AI algorithm on FPGA, to generate a refined primitive stream for L0TP selection masks

Workflow for Neural Networks deployment on FPGA



32-bit precision model

low-bit precision model

HLS model

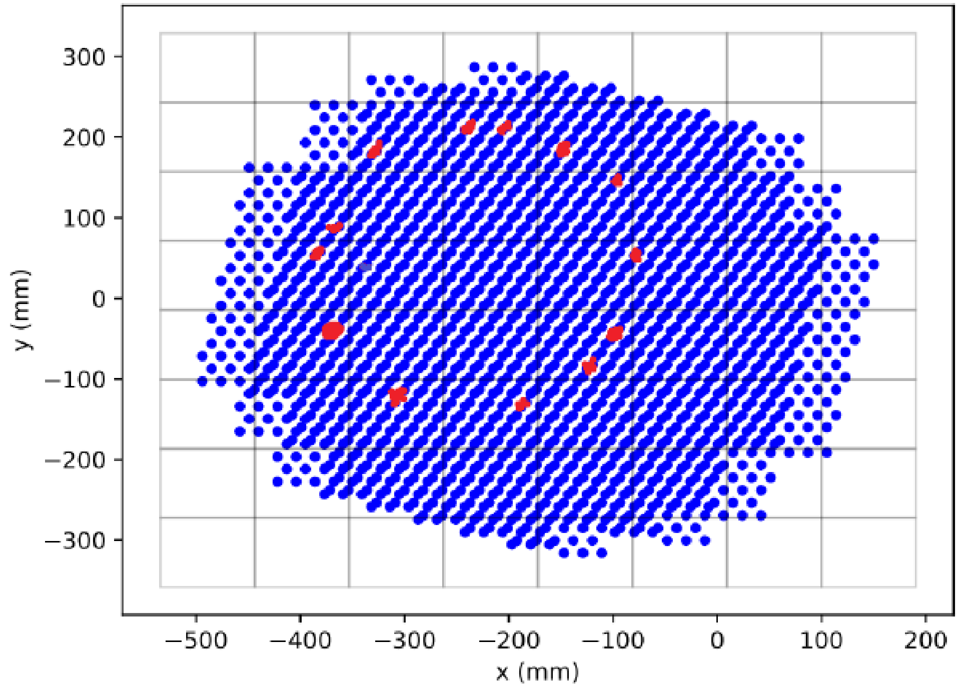
Python Libraries to train model and quantize parameters into **Fixed-point** representation

2^4	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}
0	0	0	1	0	.	1	1	0

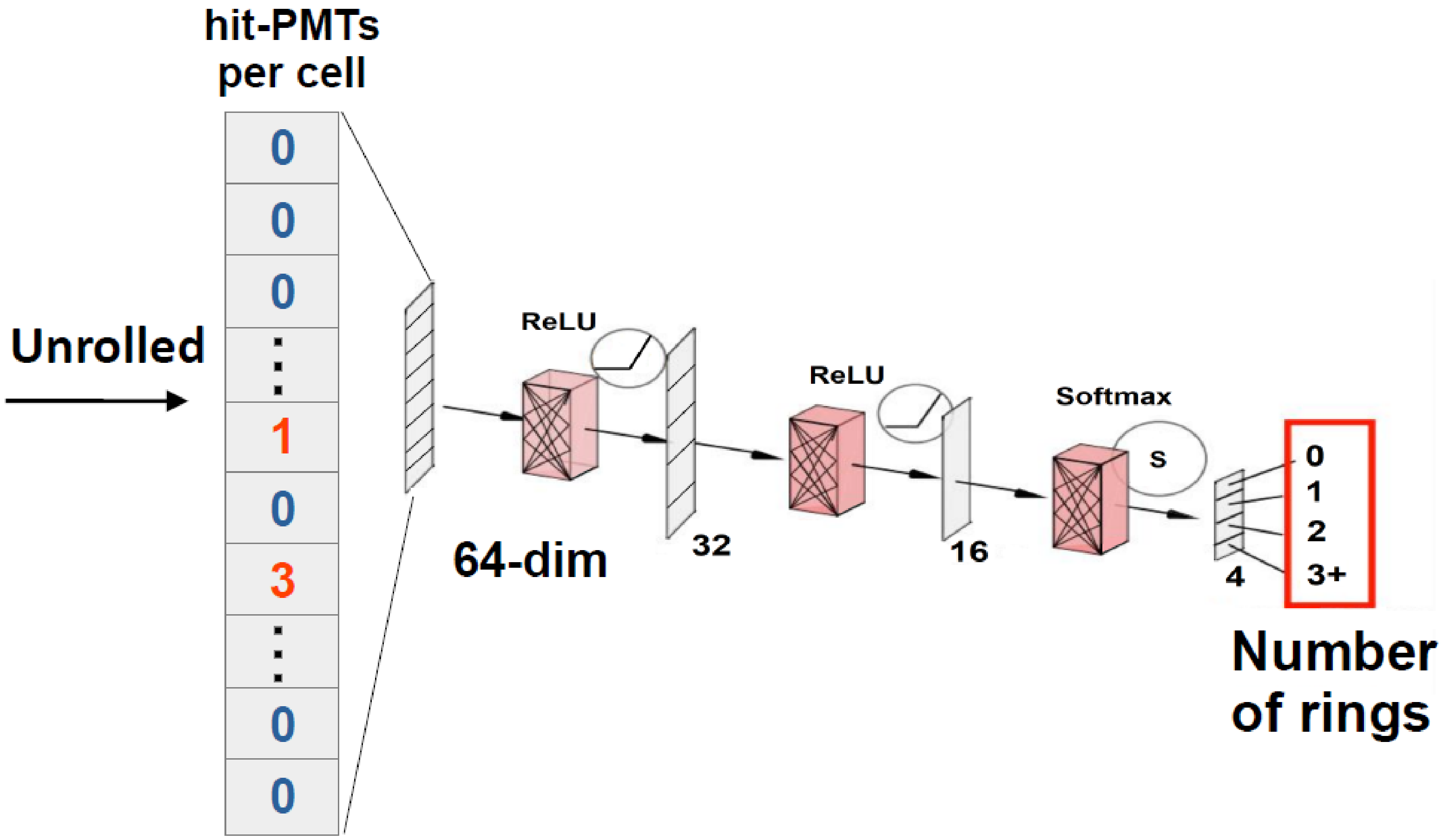
High Level Synthesis: high level C++ code transposed to hardware components, using directives to arrange pipelining, parallelization, memory structures, I/O, etc.

Iterate to find compromise between computational resources, throughput, and NN accuracy

Neural Network Model (actually one of them…)



8 x 8 cells

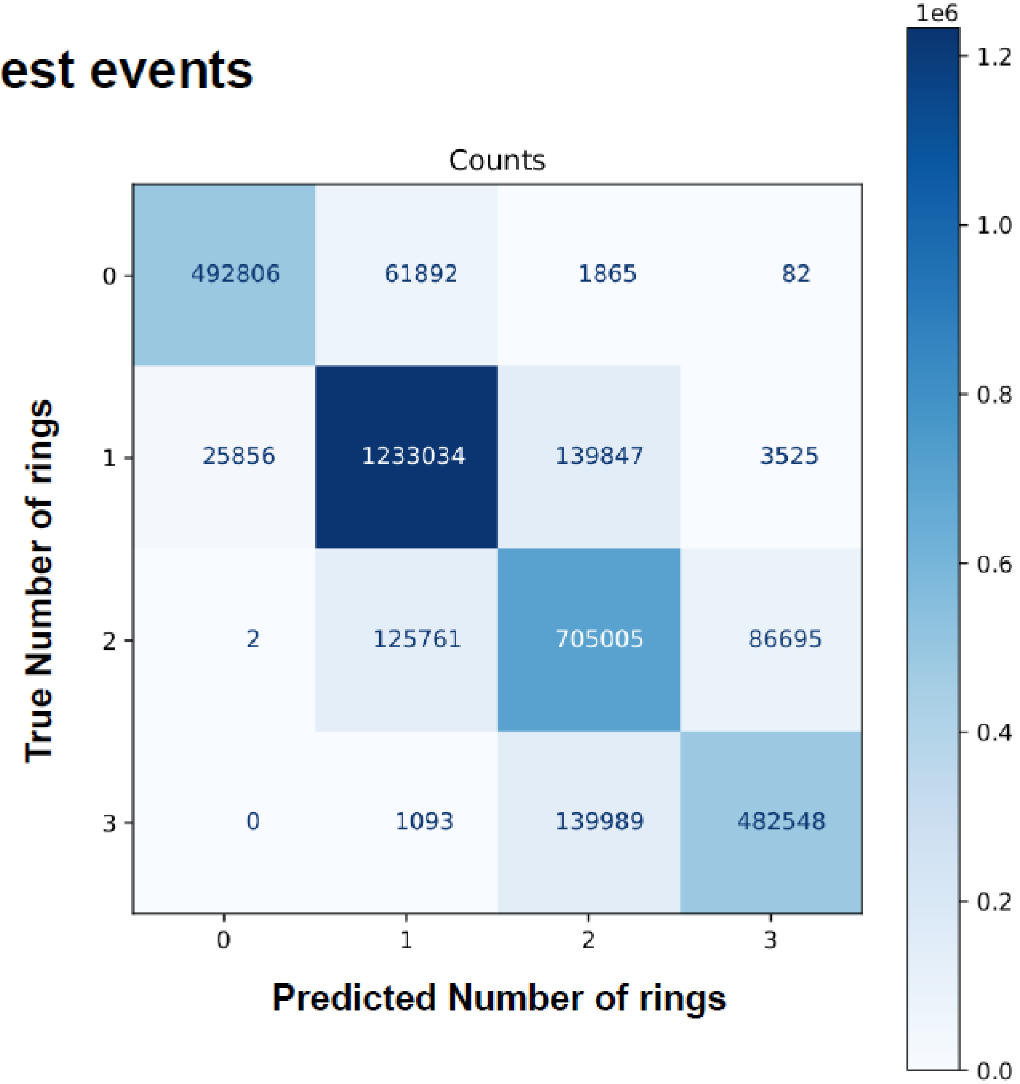


- **Encoding of the PTMs geometrical positions in the input layer.**

LUT = 14%
Flip−Flop = 6%
DSP = 7%
BRAM = 3%
on Versal VCK190

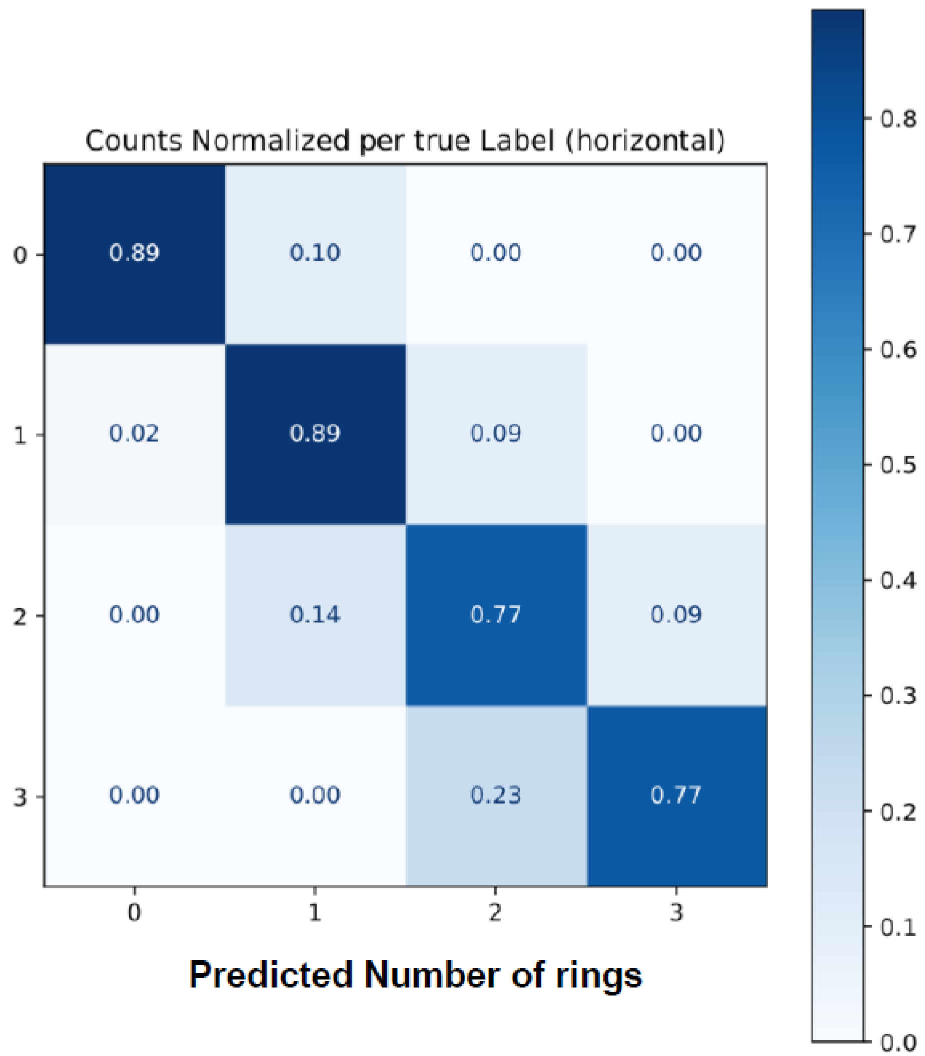
Neural Network Sensitivity

3.5 M test events



Counts

True Number of rings \ Predicted Number of rings	0	1	2	3
0	492806	61892	1865	82
1	25856	1233034	139847	3525
2	2	125761	705005	86695
3	0	1093	139989	482548

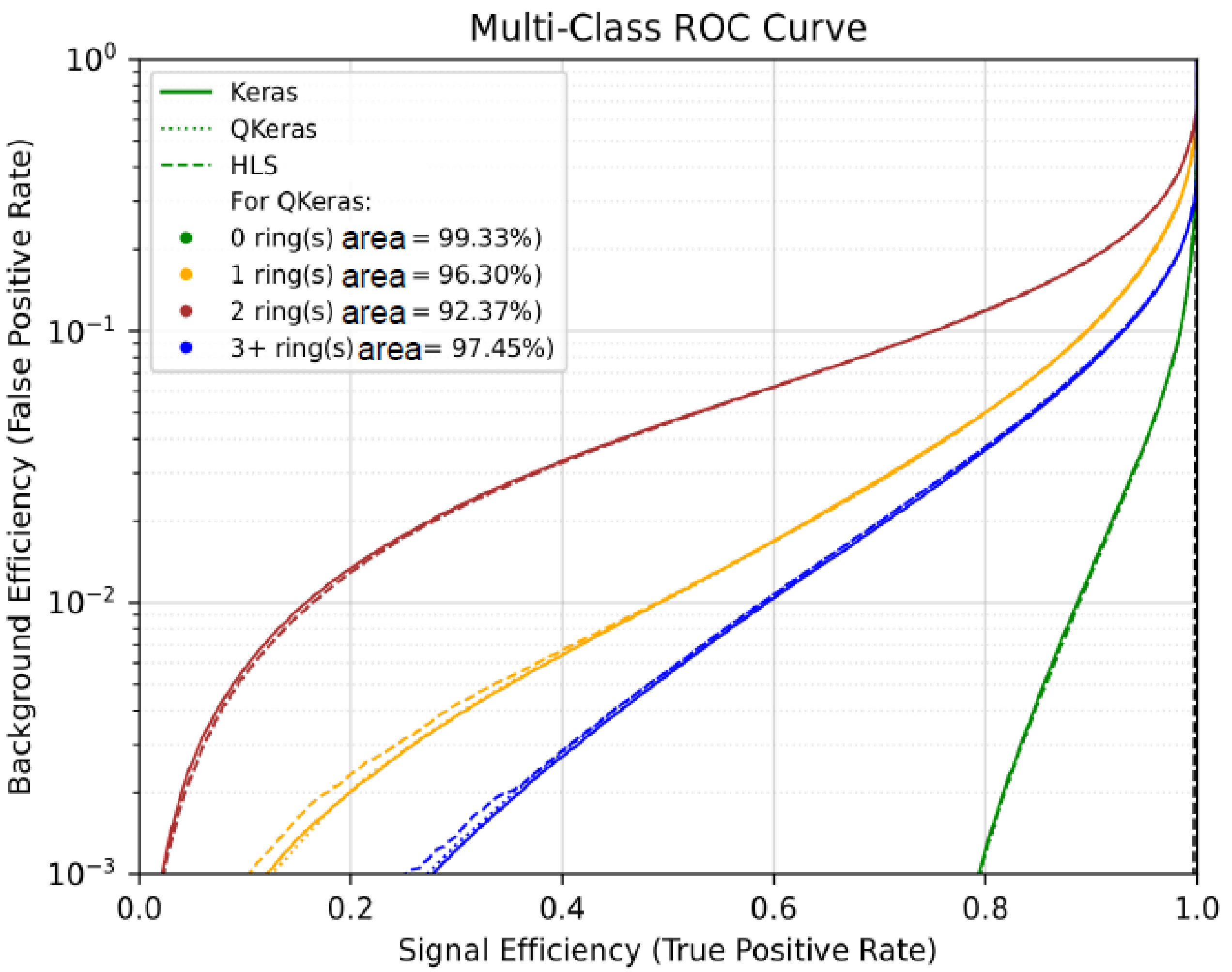


Counts Normalized per true Label (horizontal)

True Number of rings \ Predicted Number of rings	0	1	2	3
0	0.89	0.10	0.00	0.00
1	0.02	0.89	0.09	0.00
2	0.00	0.14	0.77	0.09
3	0.00	0.00	0.23	0.77

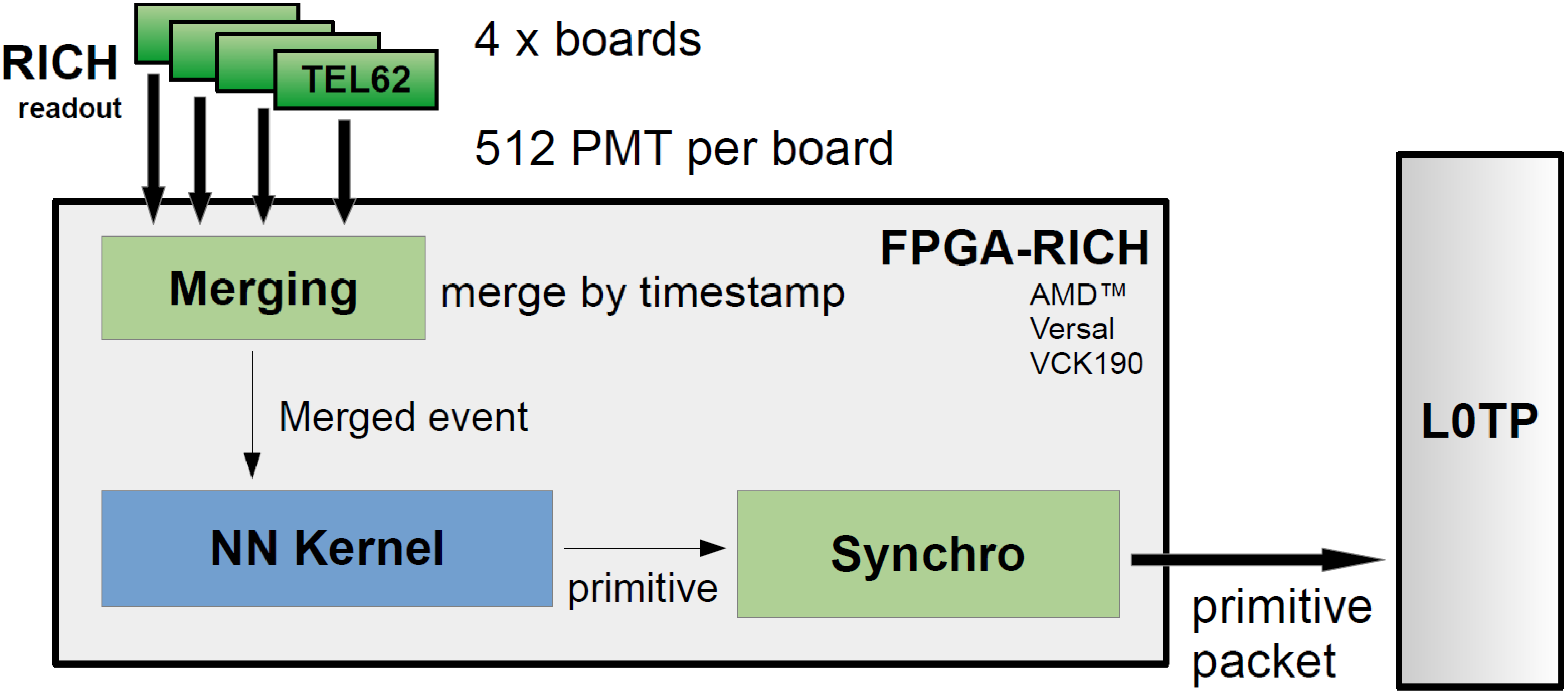
NN git: https://baltig.infn.it/ape-lab/fpgarich

ROC Curve, Throughput & Latency



NN: **avg Throughput ≈ 21 MHz** Latency = 160 ns at 300 MHz clock

Integration of the FPGA-RICH Pipeline



RICH readout

TEL62

4 x boards

512 PMT per board

FPGA-RICH

AMD™ Versal VCK190

Merging

merge by timestamp

Merged event

NN Kernel

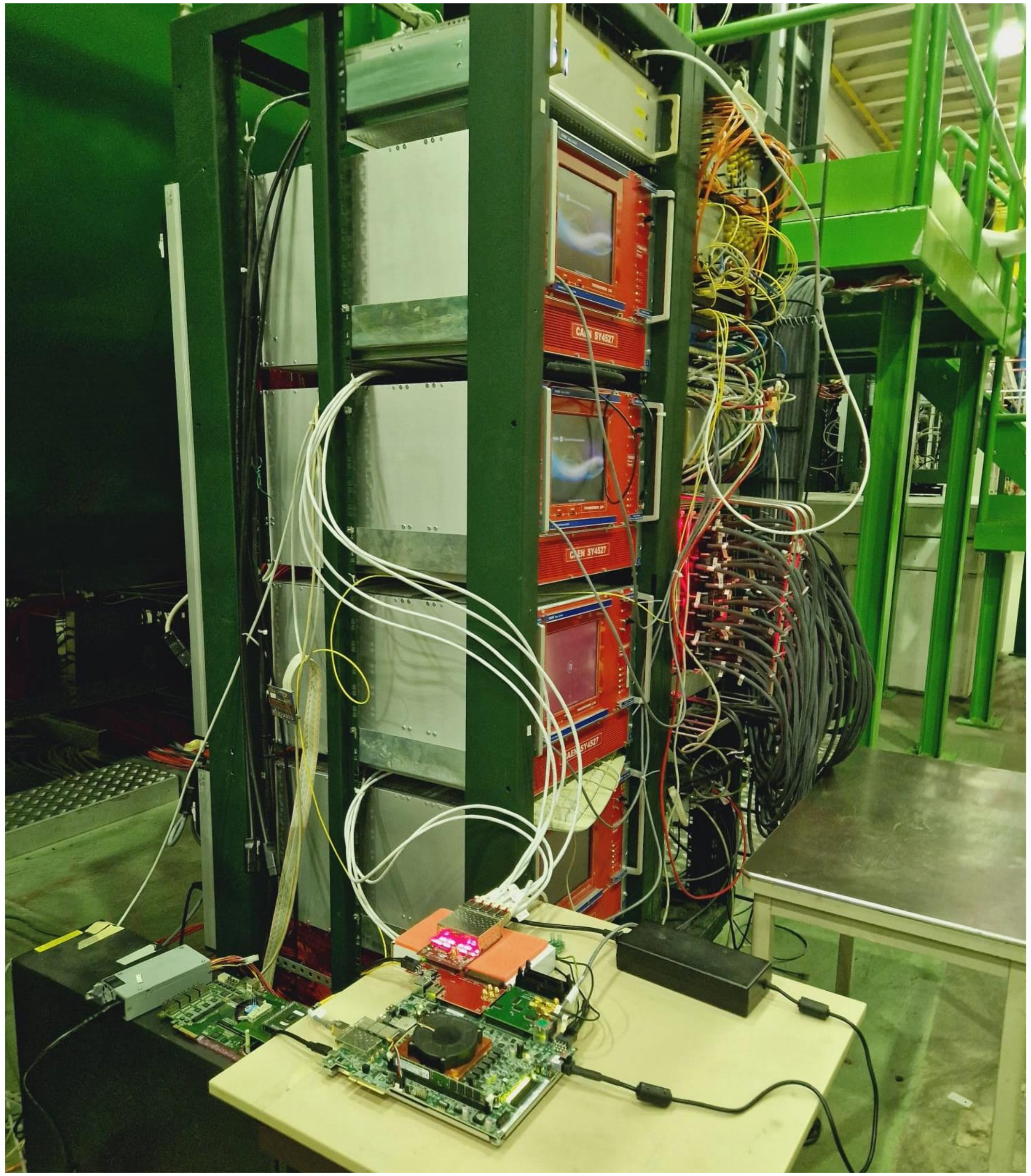
primitive

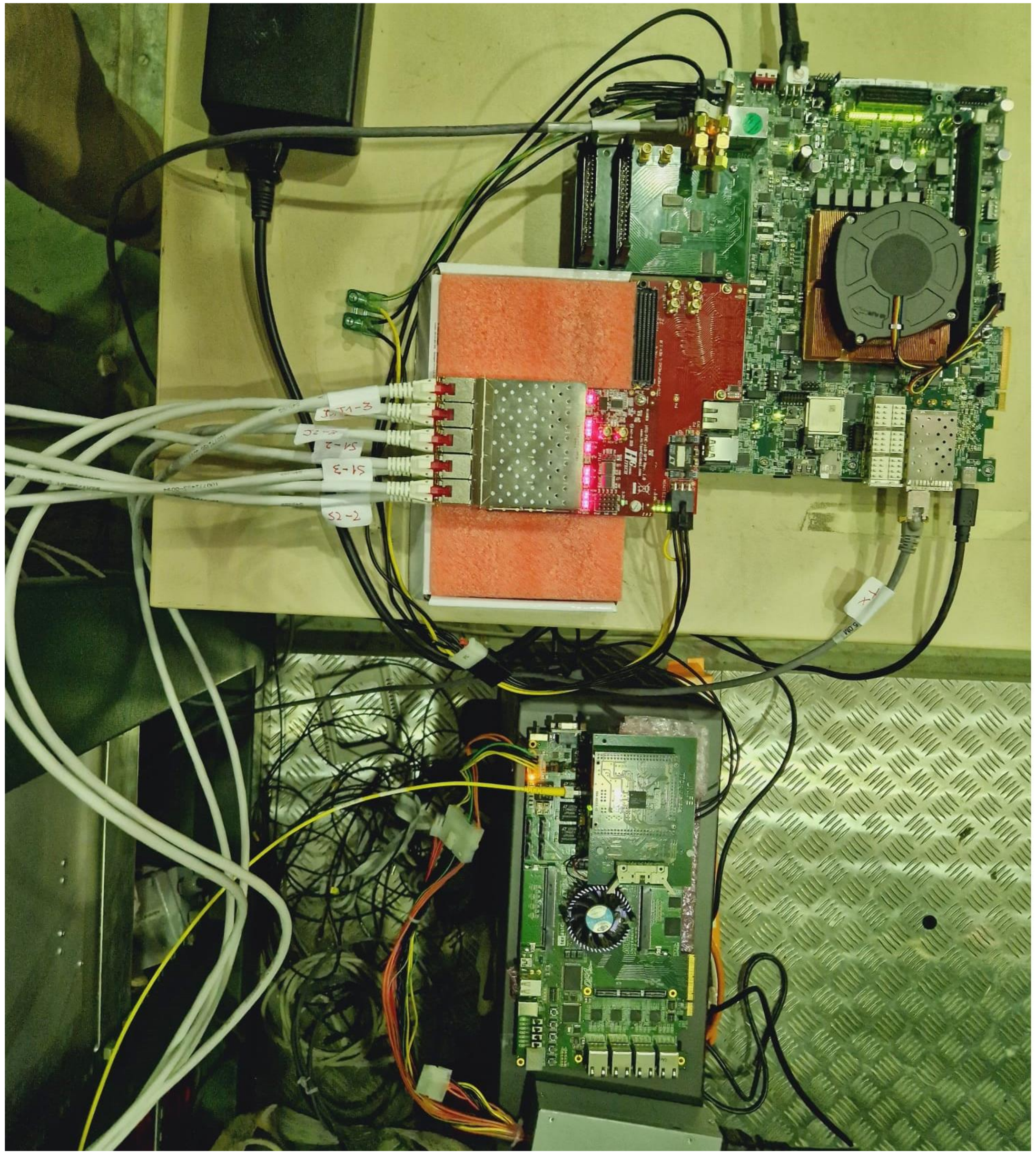
Synchro

primitive packet

L0TP

Integration of the FPGA-RICH Pipeline



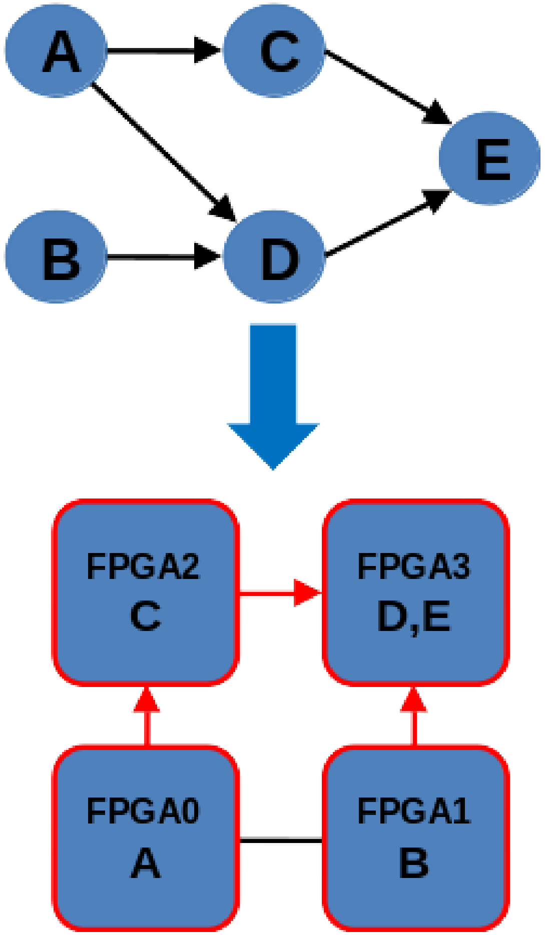


APEIRON: an overview

- **Goal:** develop a framework offering hardware and software support for the execution of real-time dataflow applications on a system composed by interconnected FPGAs .

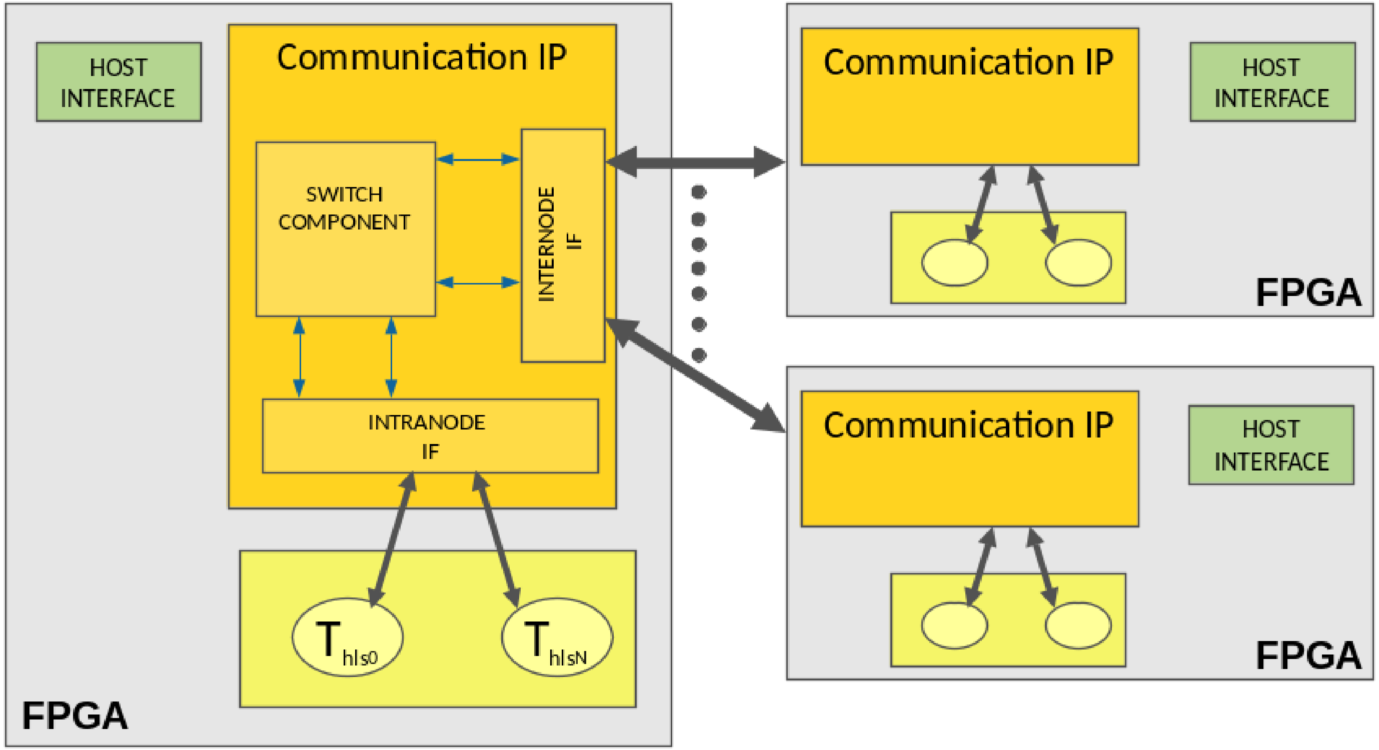
  - Map the dataflow graph of the application on the distributed FPGA system and offers runtime support for the execution.
  - Allow users with no (or little) experience in hardware design tools, to develop their applications on such distributed FPGA-based platforms
    - Tasks are implemented in C++ using High Level Synthesis tools (Xilinx Vitis).
    - Lightweight C++ communication API
      - Non-blocking *send()*
      - Blocking *receive()*

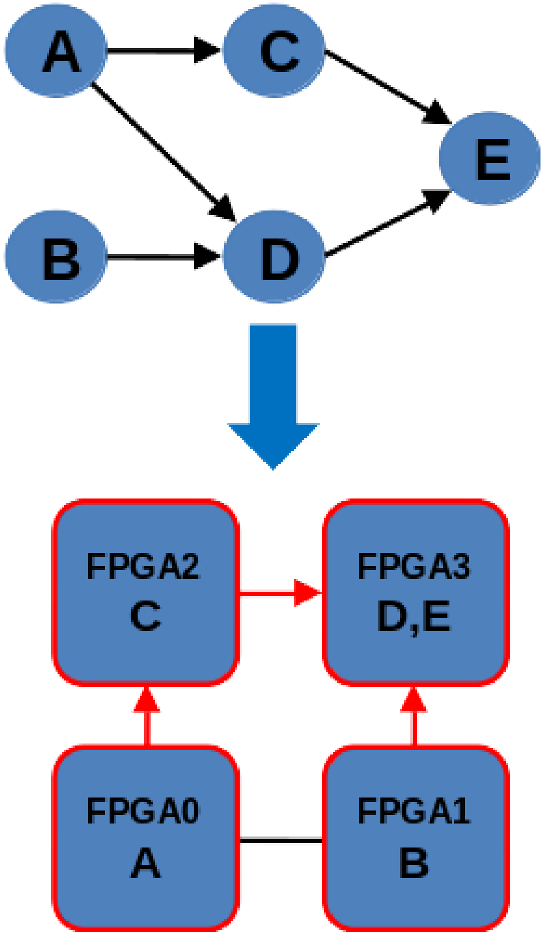
  - **APEIRON is based on Xilinx Vitis High Level Synthesis framework and on INFN Communication IP (APE Router)**



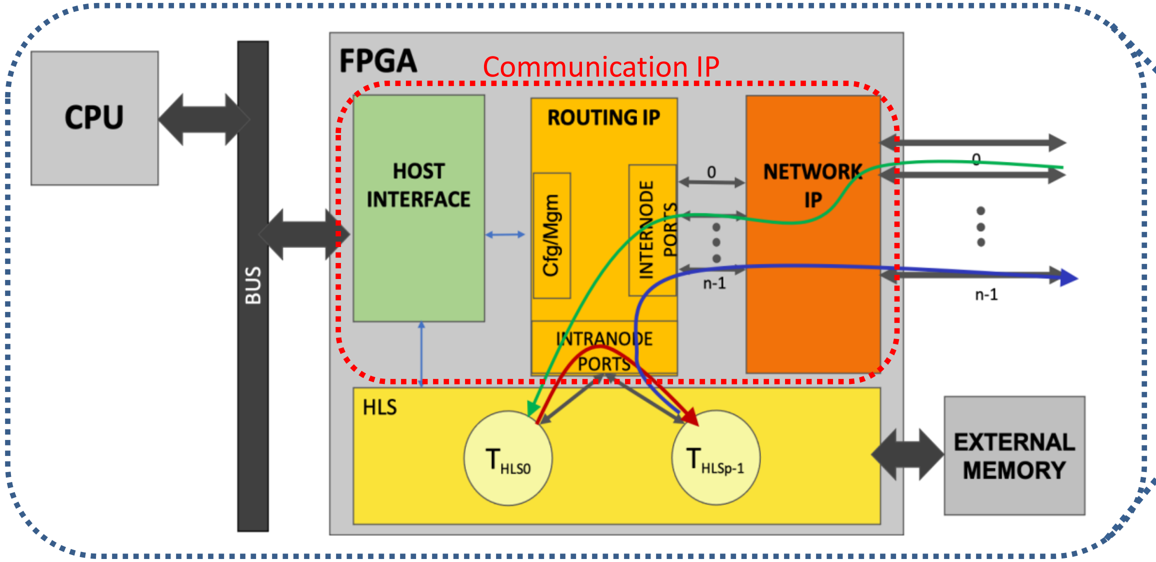
APEIRON: INFN Communication IP

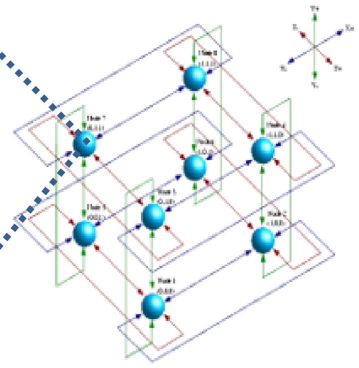
- INFN is developing the IPs implementing a **direct network** that allows **low-latency** data transfer between processing tasks deployed on the same FPGA (intra-node communication) and on different FPGAs (inter-node communication).





APEIRON: the Node

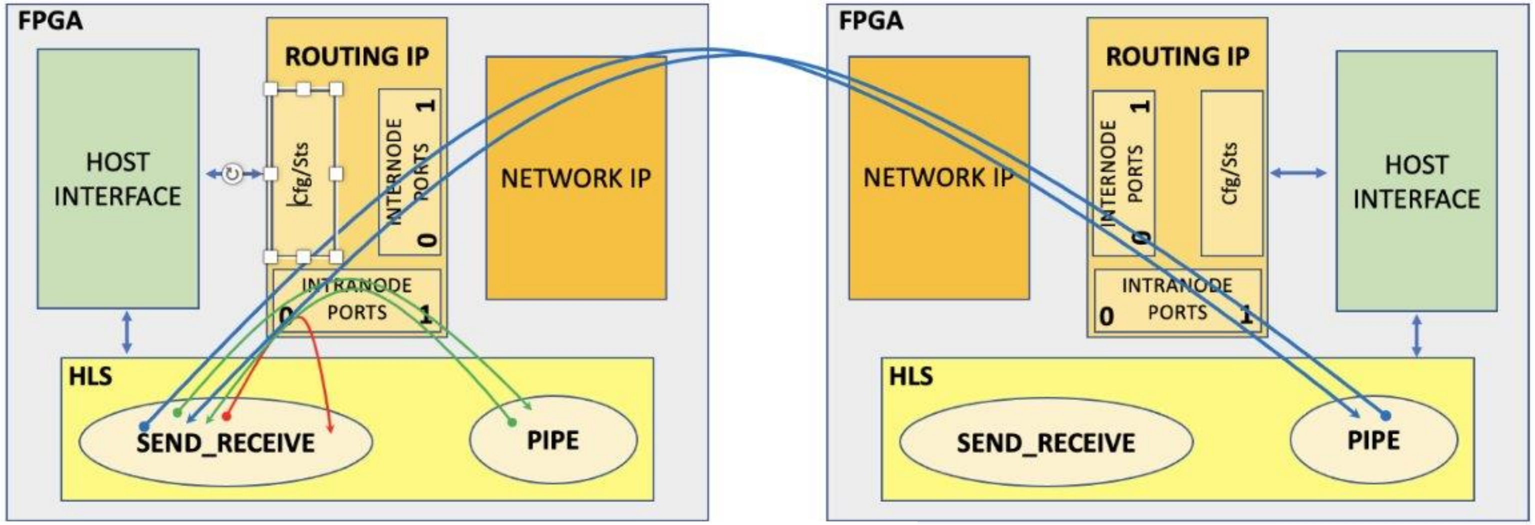


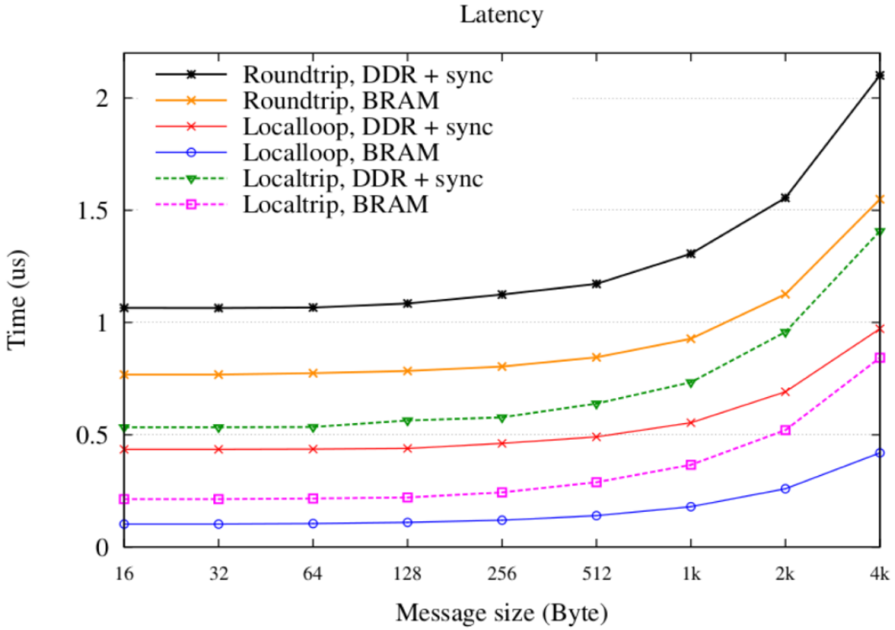


APEIRON node in a 3D Torus network topology

- **Host Interface IP**: Interface the FPGA logic with the host through the system bus.
  - Xilinx XDMA PCIe Gen3
- **Routing IP**: Routing of intra-node and inter-node messages between processing tasks on FPGA.
- **Network IP**: Network channels and Application-dependent I/O
  - APElink 40 Gbps
  - UDP/IP over 10 GbE
- **Processing Tasks**: user defined processing tasks (Xilinx Vitis HLS Kernels)

APEIRON: Communication Latency





Test modes

- Local-loop (red arrow)
- Local-trip (green arrows)
- Round-trip (blue arrows)

Test Configuration

- IP logic clock @ 200 MHz
- 4 intranode ports
- 2 internode ports
- 256-bit datapath width
- 4 lanes inter-node channels

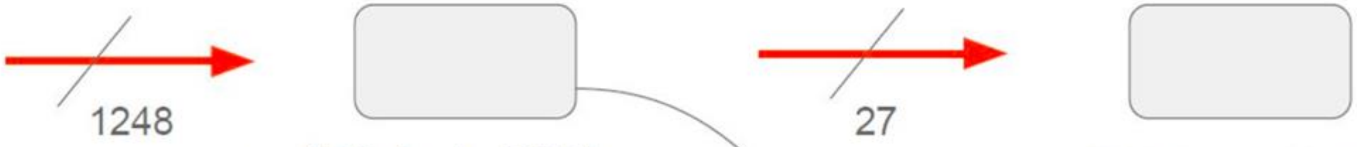
Inter-node LATENCY (orange line) < 1us for packet sizes up to 1kB (source and destination buffers in BRAM)

RDO and ePIC DAQ

P. Antonioli
https://indico.bnl.gov/event/20457/contributions/80658/attachments/49752/85138/20230914-DAQ.pdf

1248

- PDU: 1248
- RDO: 1248
- FEB: 4992



I-level DAM (27)

FELIX

- 47 links to PDU
- 1 link to II-level DAM

27

in exp. hall, rack mounted

PC with 4 FELIX each (??)

II-level DAM (1)

FELIX

- 27 links to I-level DAM
- link from central ePIC [clock/trigger]

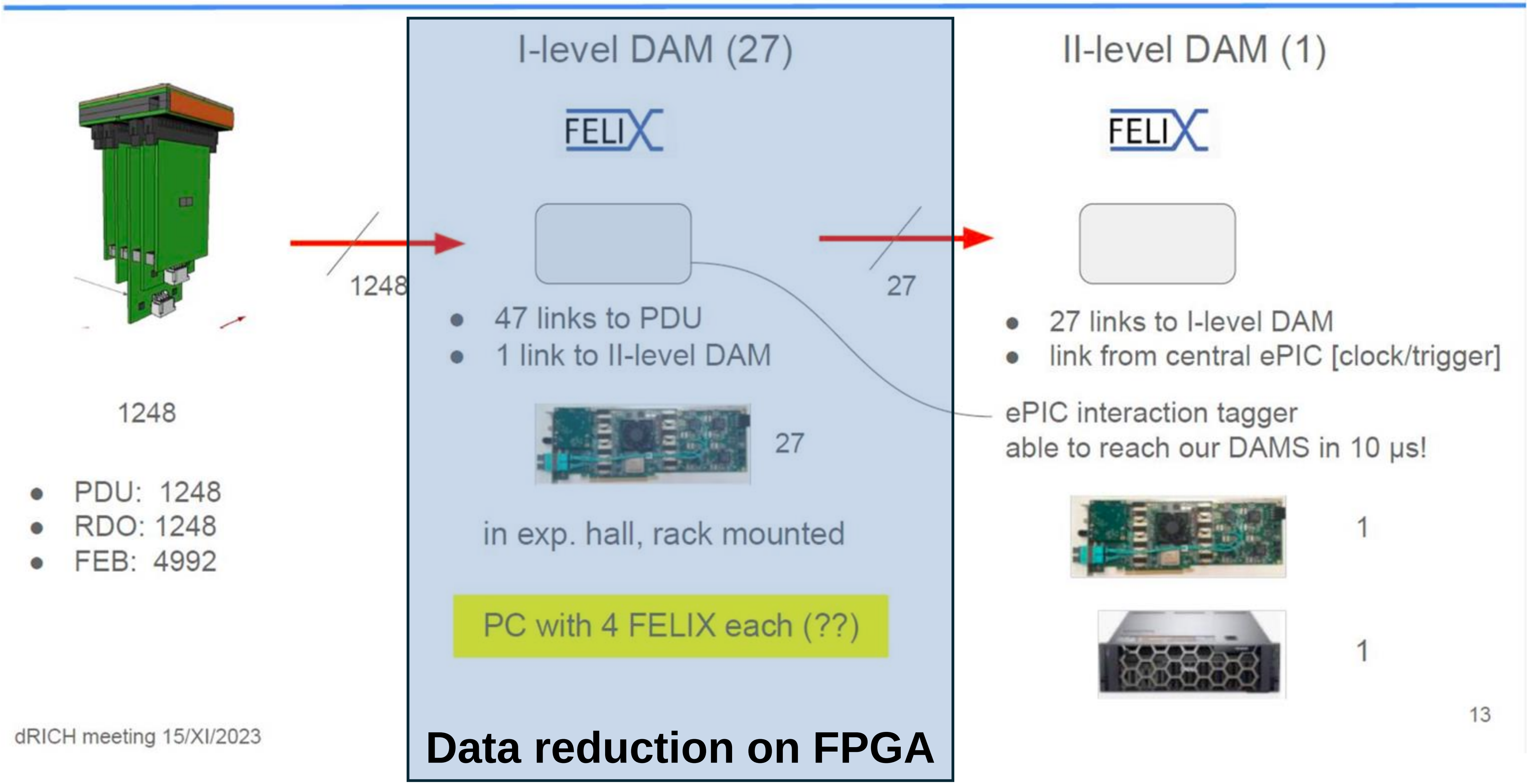
ePIC interaction tagger
able to reach our DAMS in 10 μs!

1

1

RDO and ePIC DAQ

P. Antonioli
https://indico.bnl.gov/event/20457/contributions/80658/attachments/49752/85138/20230914-DAQ.pdf



1248

- PDU: 1248
- RDO: 1248
- FEB: 4992

I-level DAM (27)

FELIX

- 47 links to PDU
- 1 link to II-level DAM

in exp. hall, rack mounted

PC with 4 FELIX each (??)

II-level DAM (1)

FELIX

- 27 links to I-level DAM
- link from central ePIC [clock/trigger]

ePIC interaction tagger
able to reach our DAMS in 10 μs!

dRICH meeting 15/XI/2023

13

Data reduction on FPGA

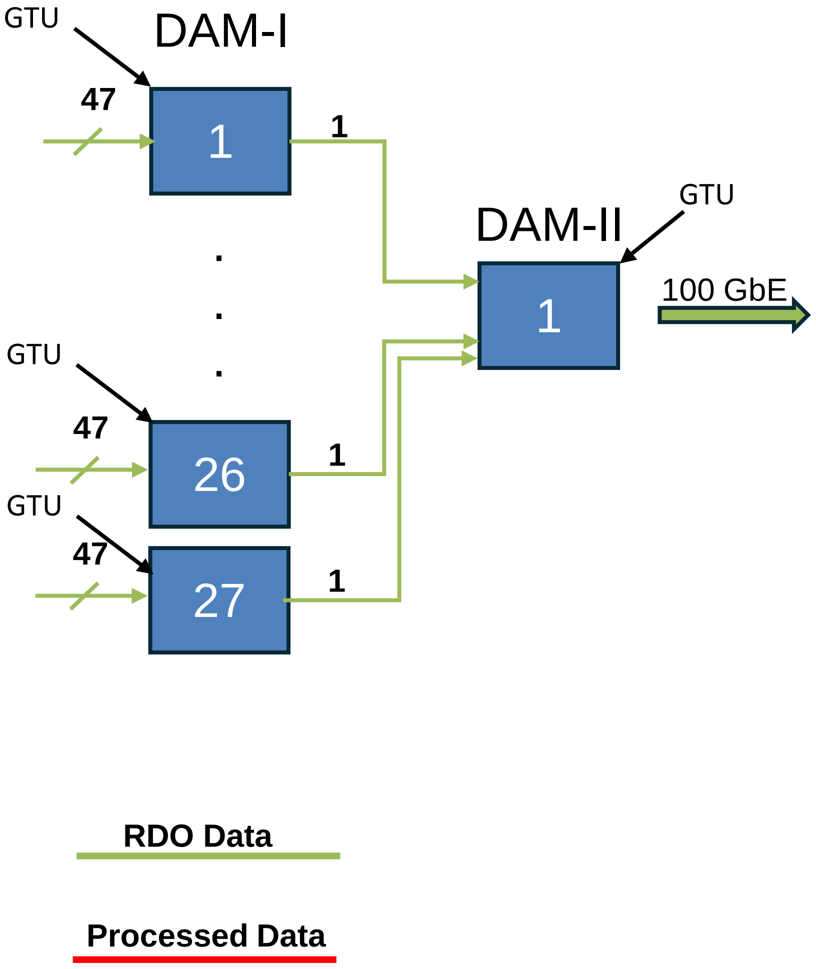
dRICH Data Reduction Stage on FPGA

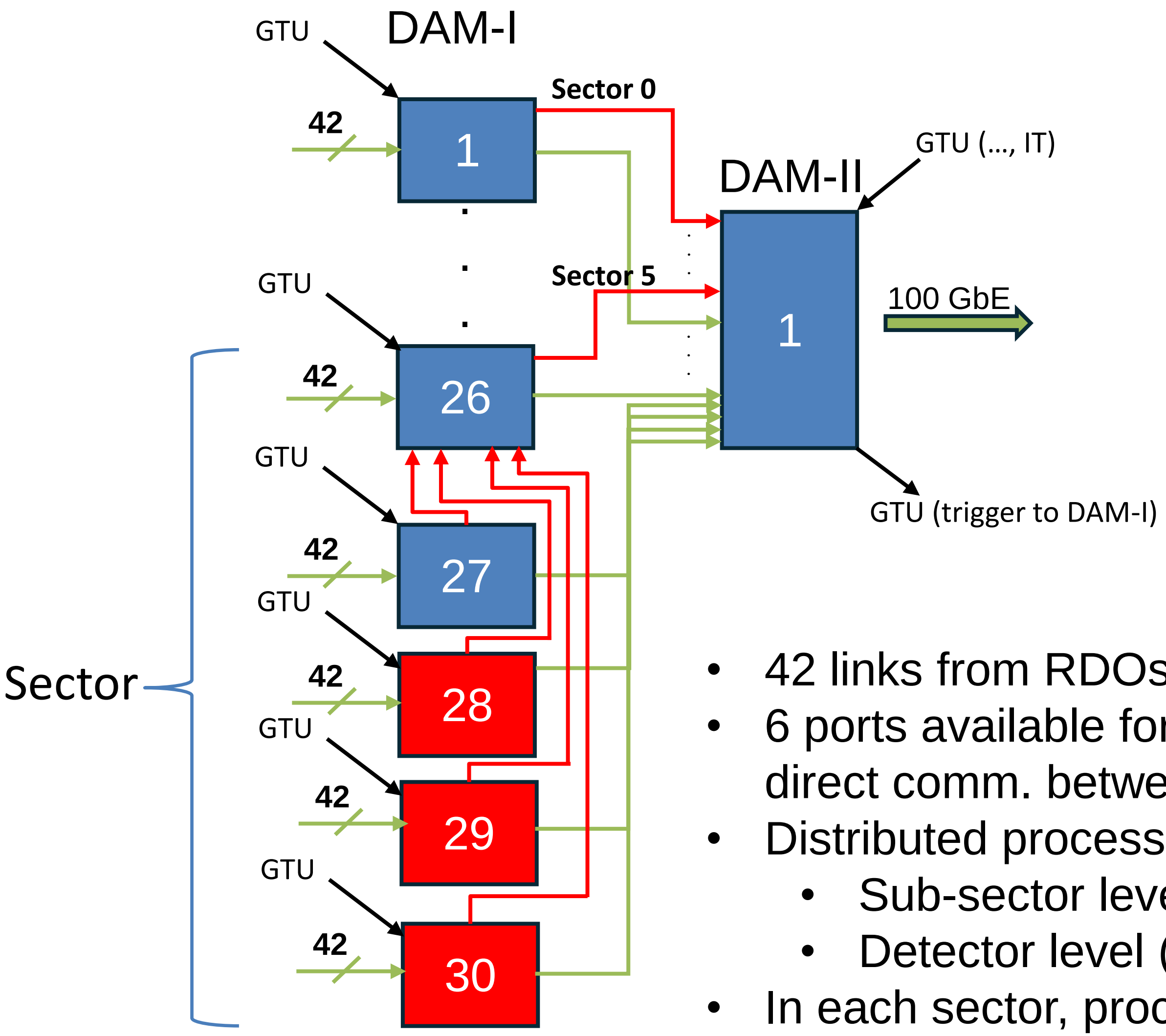
- Objective: design of a data reduction stage for the dRICH with a ~100 data bandwidth reduction in DAM-I level output to DAM-II level input.
- Make exclusive use of DAQ components (Felix DAMs)
    - Add few DAM units wrt the bare minimum needed to readout the 1248 RDO links to implement a distributed processing scheme.
    - Integration with the Interaction Tagger to boost performance and enable other features.
- Online Signal/Background discrimination using ML
    - Collecting datasets using data available from simulation campaigns
        - Background:
            - e/p with beam pipe gas
            - Synchrotron radiation (MC only?)
        - Merged: signal + e/p with beam pipe gas background (full), few events
        - SiPM Noise
            - DCR modelled in the reconstruction stage
                - Spatial and time dependency of the rate?

dRICH Data Reduction Stage on FPGA

- Online Signal/Background discrimination using ML (continued)
    - Study of Inference Models
        - Restricting our study to inference models that can be deployed on FPGA with reasonable effort (using a High-Level Synthesis workflow)
            - MLP, CNN, GNN NN Models (HLS4ML)
            - BDT (Conifer)
        - Inference throughput (98.5 MHz) is the main concern.
        - HDL optimized implementation is an option.
    - Deployment on multiple Felix DAMs directly interconnected with the APE communication IPs

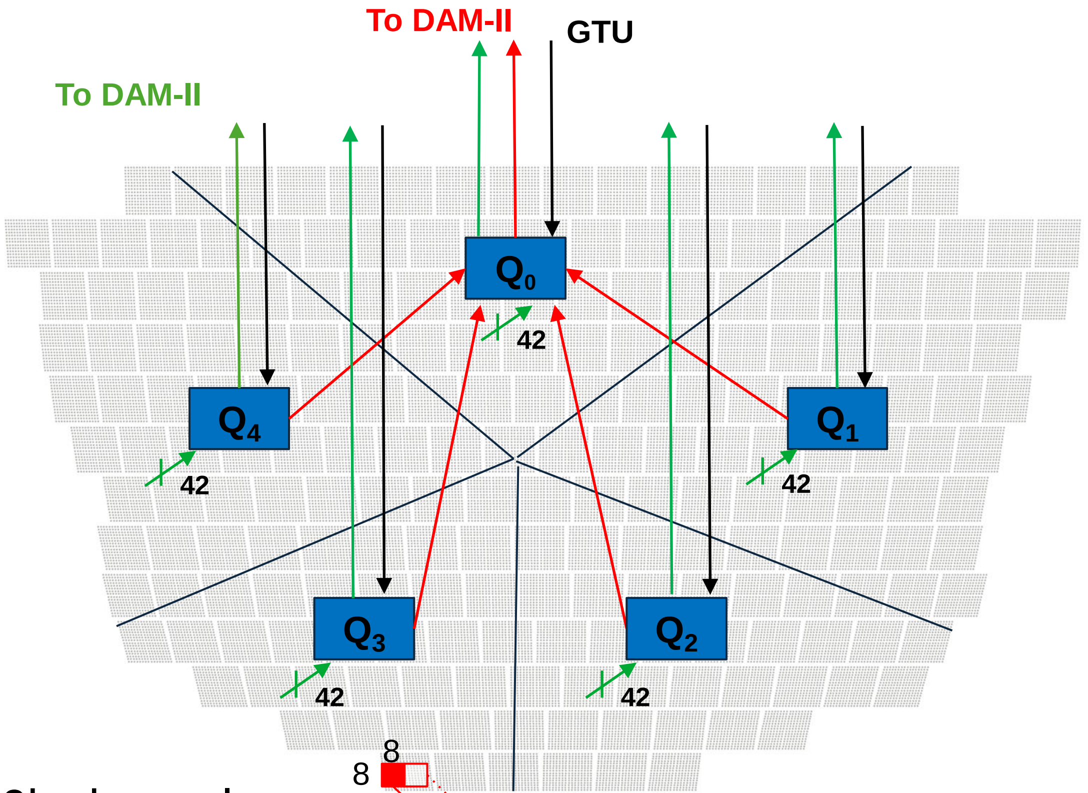
dRICH Data Reduction Stage on FPGA: example deployment





- 42 links from RDOs
- 6 ports available for direct comm. between DAMs
- Distributed processing
  - Sub-sector level (DAM-I)
  - Detector level (DAM-II)
- In each sector, processed data routed by one DAM-I to DAM-II

dRICH Data Reduction Stage on FPGA: example deployment

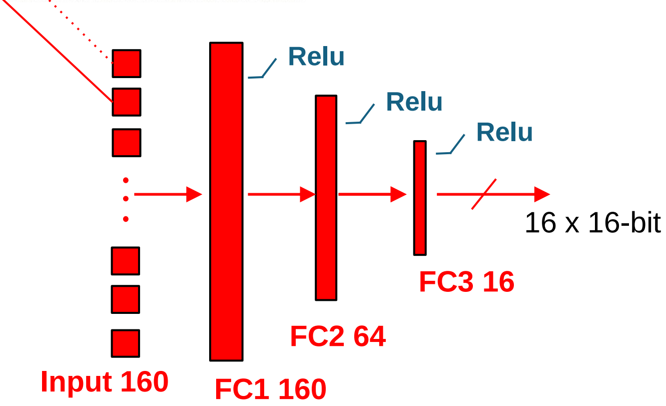


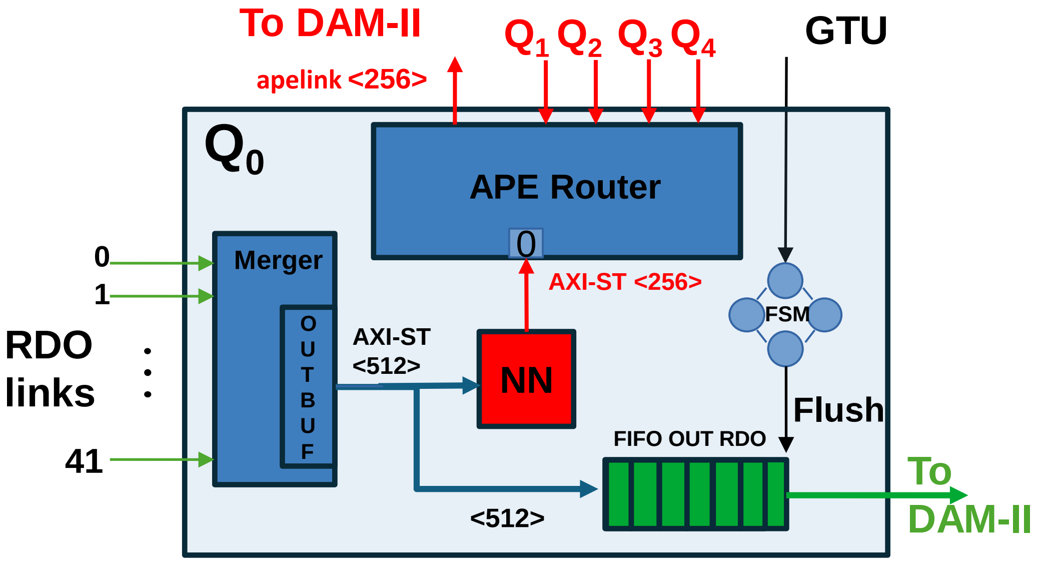
50k channels per sector

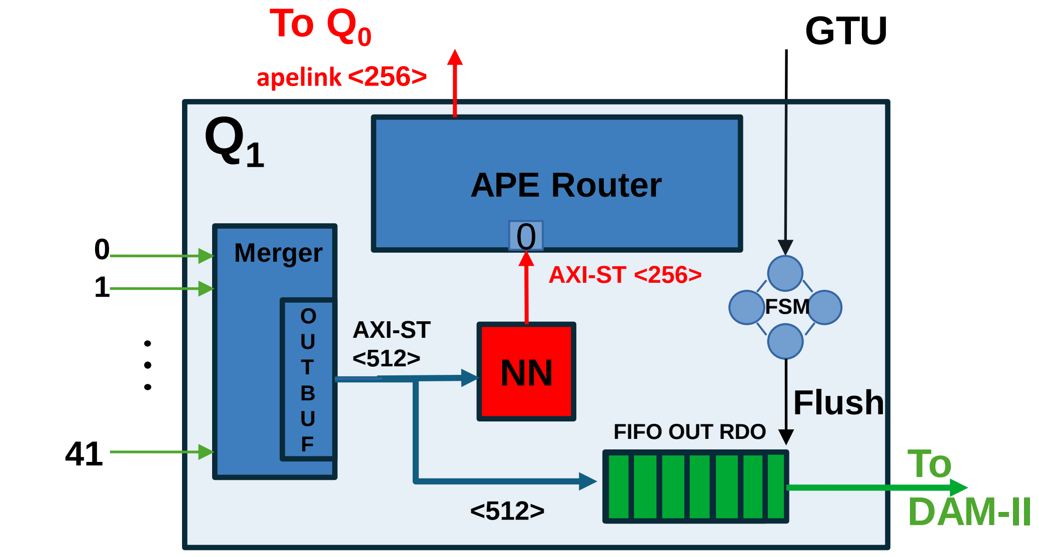
→ RDO flow

→ PROC flow

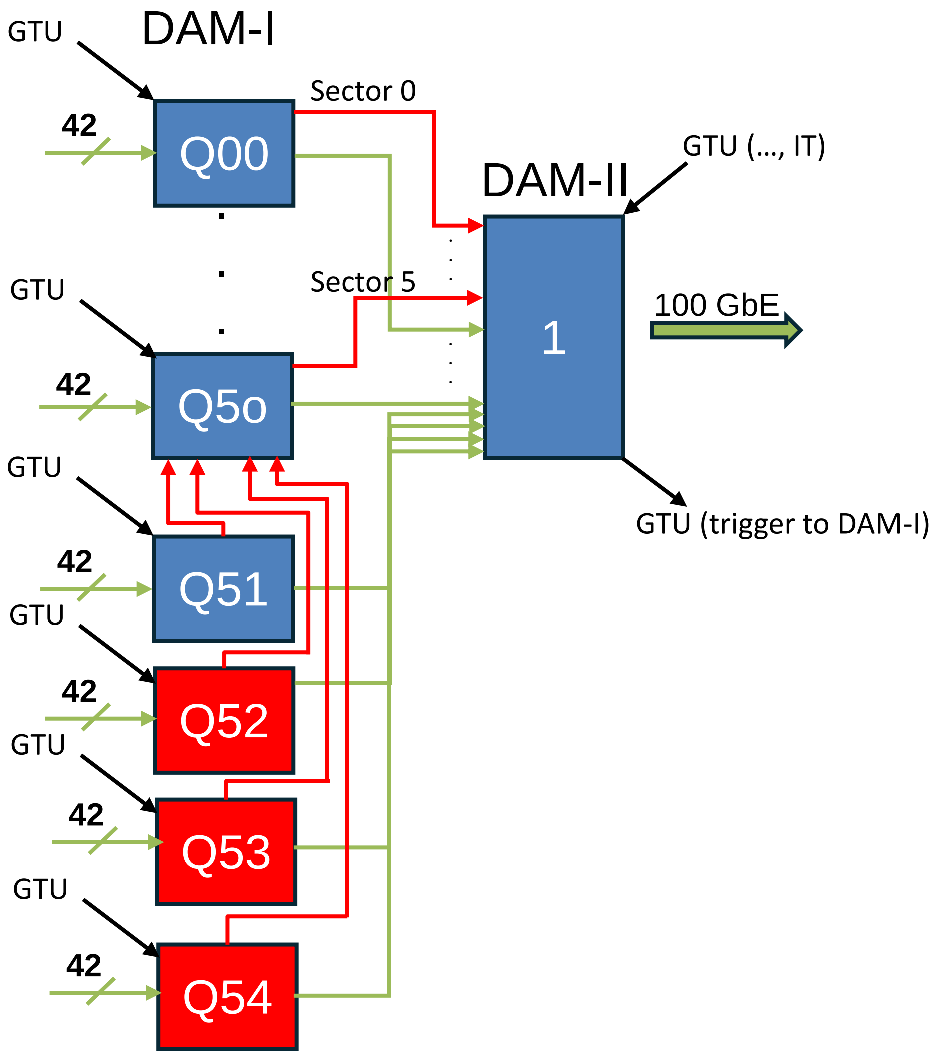
→ GTU I/O

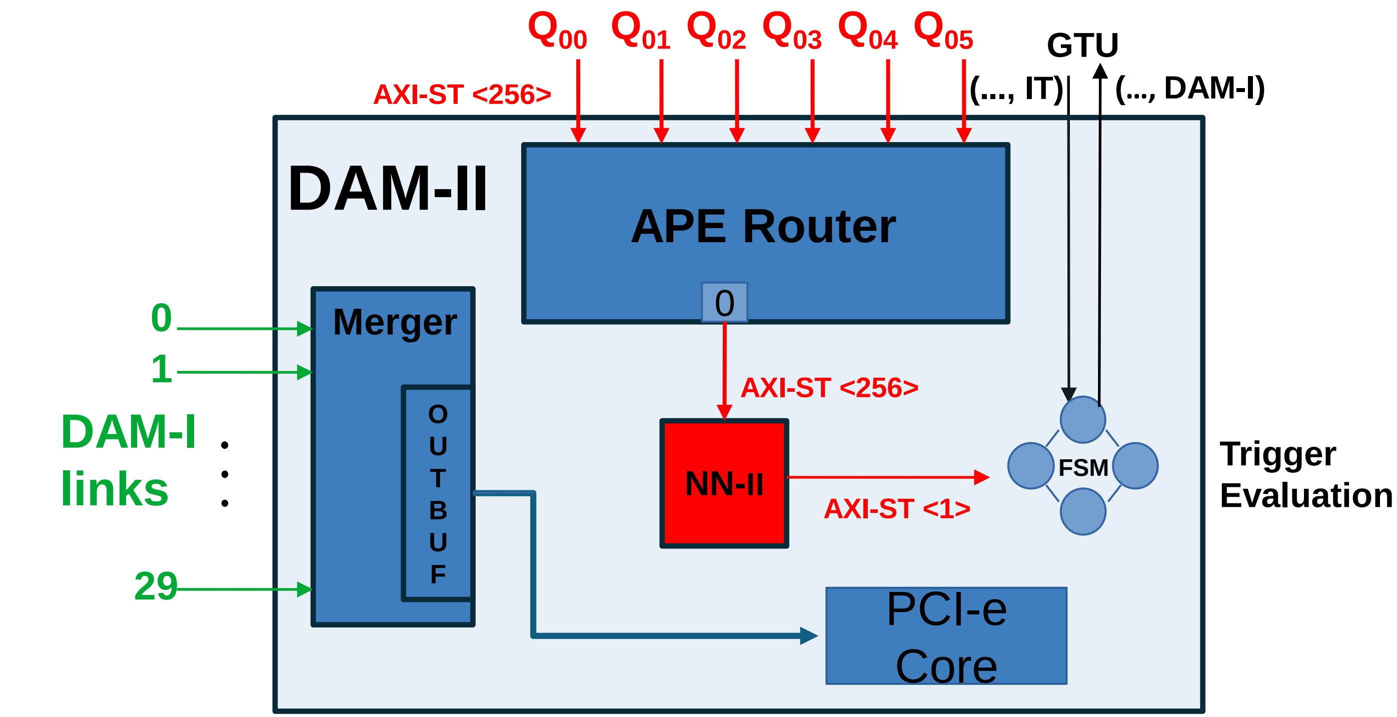


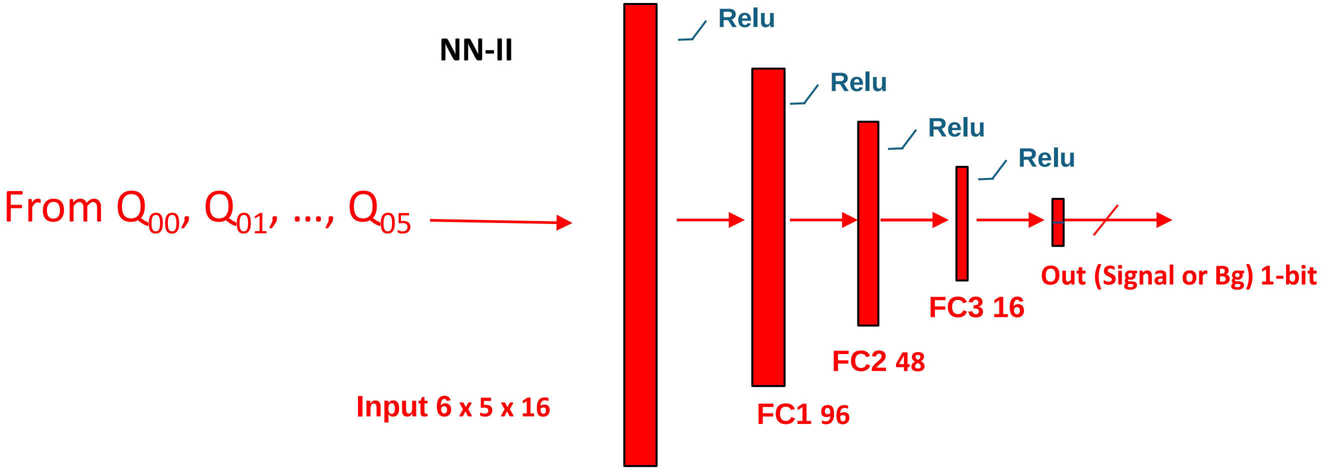




dRICH Data Reduction Stage on FPGA: example deployment





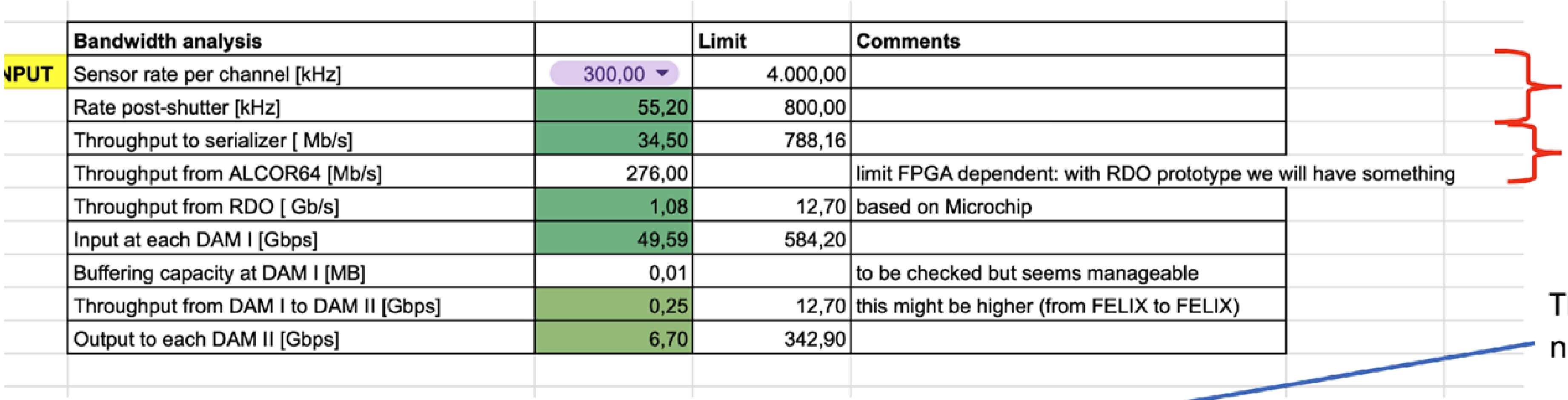


Current status and outlook

- We have started collecting datasets and experimenting with inference models.
- Details of the final deployment will be affected by several factors
    - Final selection on the inference model(s): BDT, MLP, CNN, GNN, …
    - Net amount of FPGA resources available (discounting the “standard” DAQ firmware) in DAMs.
    - Actual additional DAQ resources (DAMs, …) dedicated to the data reduction system.
- Possible additional features
    - Provide services (statistics) for the online monitoring.
    - Having track seeds information from the Interaction Tagger could enable more sophisticated features
        - Particle counting
        - Particle identification
            - We have devised a method to tag reconstructed events with PIDs.

Backup Slides

Throughput modeling (results)

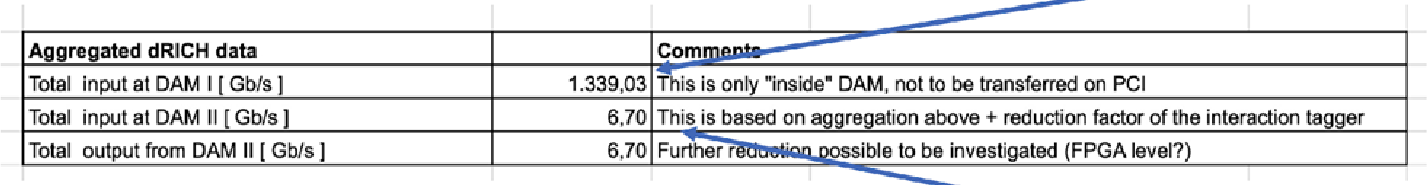


	Bandwidth analysis		Limit	Comments
NPUT	Sensor rate per channel [kHz]	300,00 ▾	4.000,00	
	Rate post-shutter [kHz]	55,20	800,00	
	Throughput to serializer [Mb/s]	34,50	788,16	
	Throughput from ALCOR64 [Mb/s]	276,00		limit FPGA dependent: with RDO prototype we will have something
	Throughput from RDO [Gb/s]	1,08	12,70	based on Microchip
	Input at each DAM I [Gbps]	49,59	584,20	
	Buffering capacity at DAM I [MB]	0,01		to be checked but seems manageable
	Throughput from DAM I to DAM II [Gbps]	0,25	12,70	this might be higher (from FELIX to FELIX)
	Output to each DAM II [Gbps]	6,70	342,90	

ALCOR

RDO

This is the aggregated number we could sell



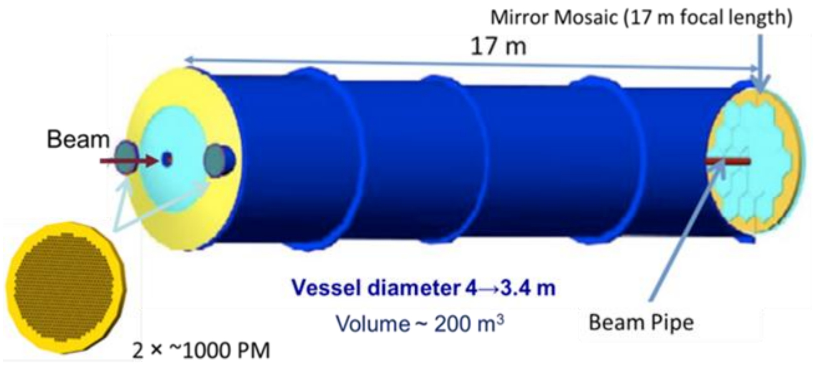
Aggregated dRICH data		Comments
Total input at DAM I [Gb/s]	1.339,03	This is only "inside" DAM, not to be transferred on PCI
Total input at DAM II [Gb/s]	6,70	This is based on aggregation above + reduction factor of the interaction tagger
Total output from DAM II [Gb/s]	6,70	Further reduction possible to be investigated (FPGA level?)

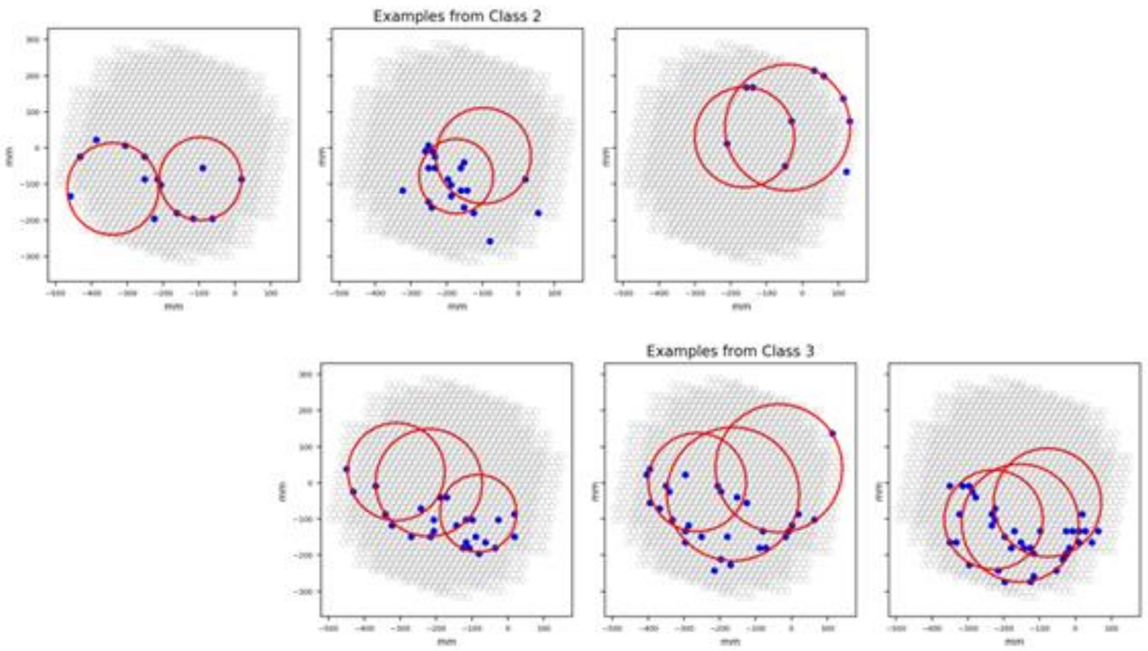
Note: first hard limit (RDO-DAM link) hit only at 3 MHz input..

But we should think how to present things... (see next slide)

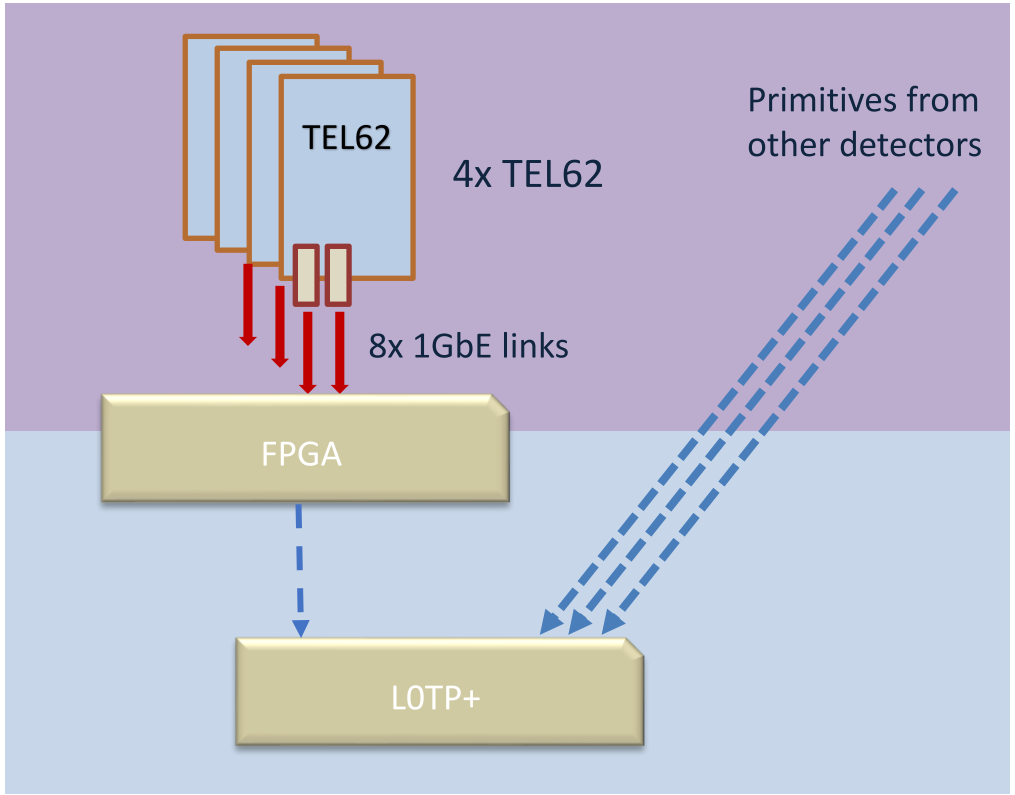
PID in NA62 with the RICH using NN on FPGA at L0 Trigger

- Goal: for any event detected by the RICH provide an estimate for the **number charged particles** and the **number of electrons**
- Streaming readout processing on FPGA using Neural Networks for classification (10 MHz).
- Produce a new primitives stream for L0TP+
- **The main challenge is the proc. throughput**

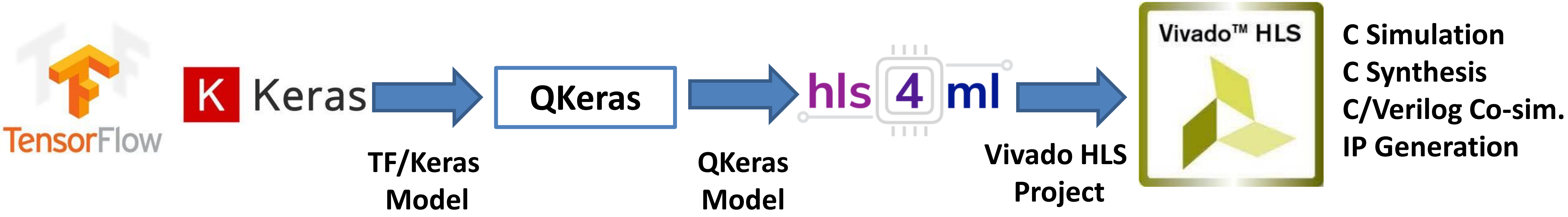



Examples from Class 2

Examples from Class 3



Design and Implementation Workflow

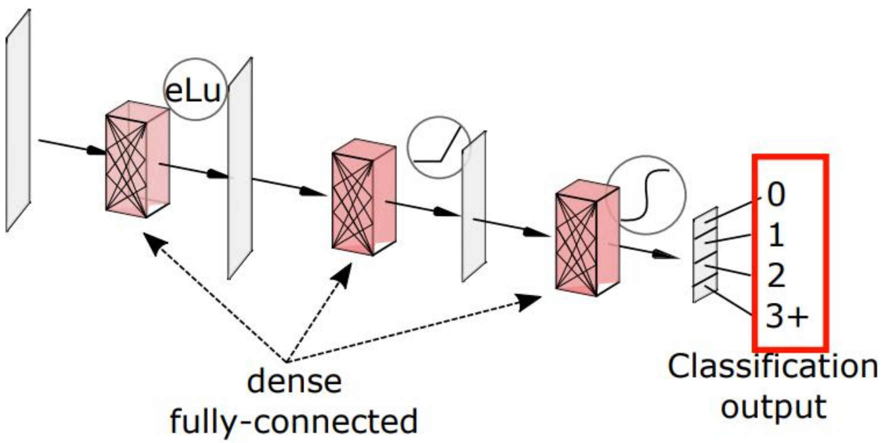


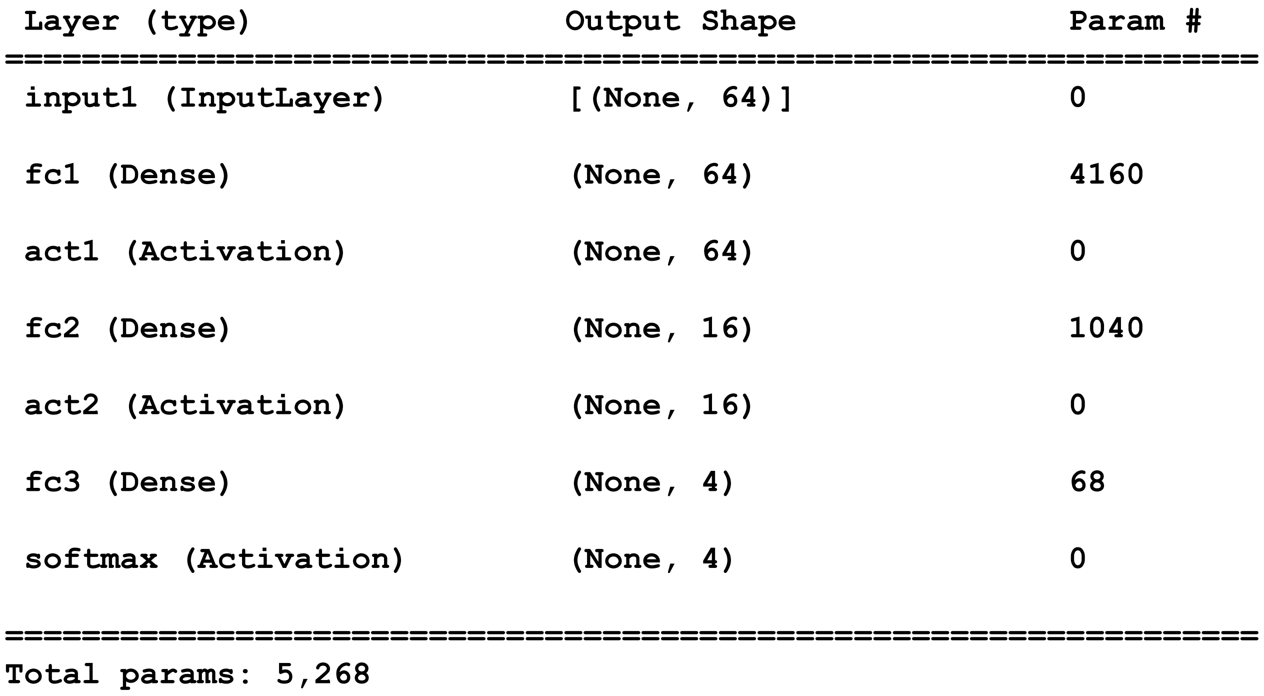
Design targets (efficiency, purity, throughput, latency) and constraints (mainly FPGA resource usage) must be taken into account and verified at any stage:

- Generation strategy of training and validation data sets.
- **TF/KERAS** NN architecture (number and kind of layers) and **representation of the input**
  - Training strategy (class balancing, batch sizes, optimizer choice, learning rate,...).
- **QKeras** Serach iteratively the minimal representation size in bits of weights, biases and activations, possibly by layer.
- **hls4ml** Tuning of REUSE FACTOR config param (low values -> low latency, high throughput, high resource usage), clock frequency.
- **Vivado HLS** co-simulation for verification of performance (experimented very good agreement with QKeras Model)

NN Architectures: Dense Model

- Input representation: normalized hitlist (max 64 hits per event)
- Output: 4 classes (0, 1, 2, 3+ rings)
- Quantization (fixed point)
  - Weights and biases: 8 bits <8, 1>
  - Activations: 16 bits <16, 6>
- FPGA resource usage (VCU118)
  LUT 14%, DSP 2%, BRAM 0%
- Latency: 22 cycles @ 150MHz
- Initiation Interval (II): 8 cycles
- Throughput: 18.75 MHz

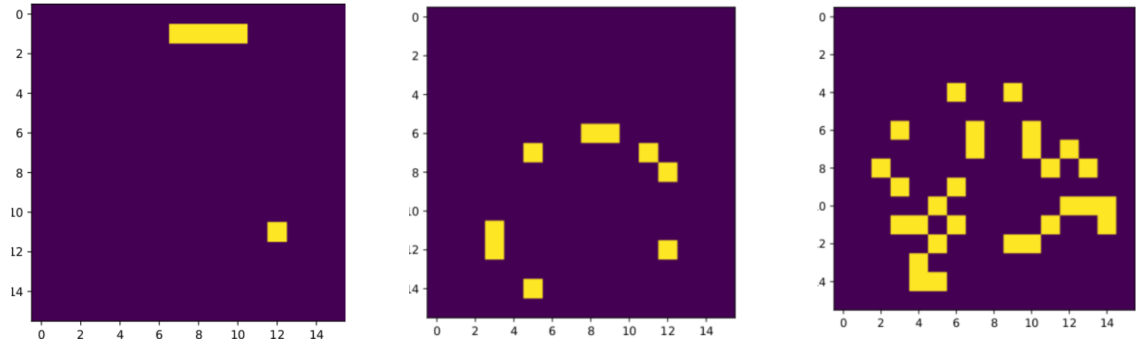


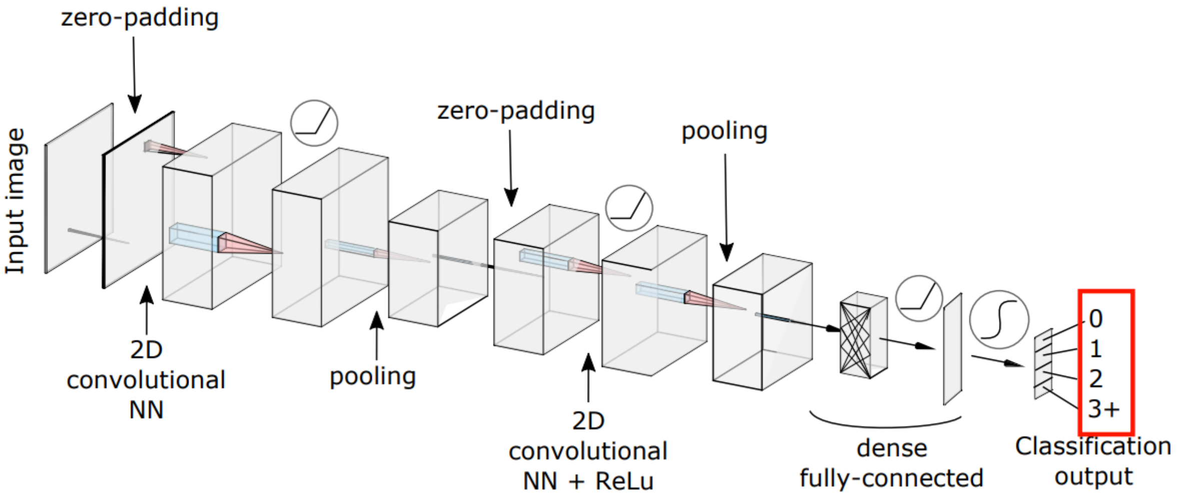


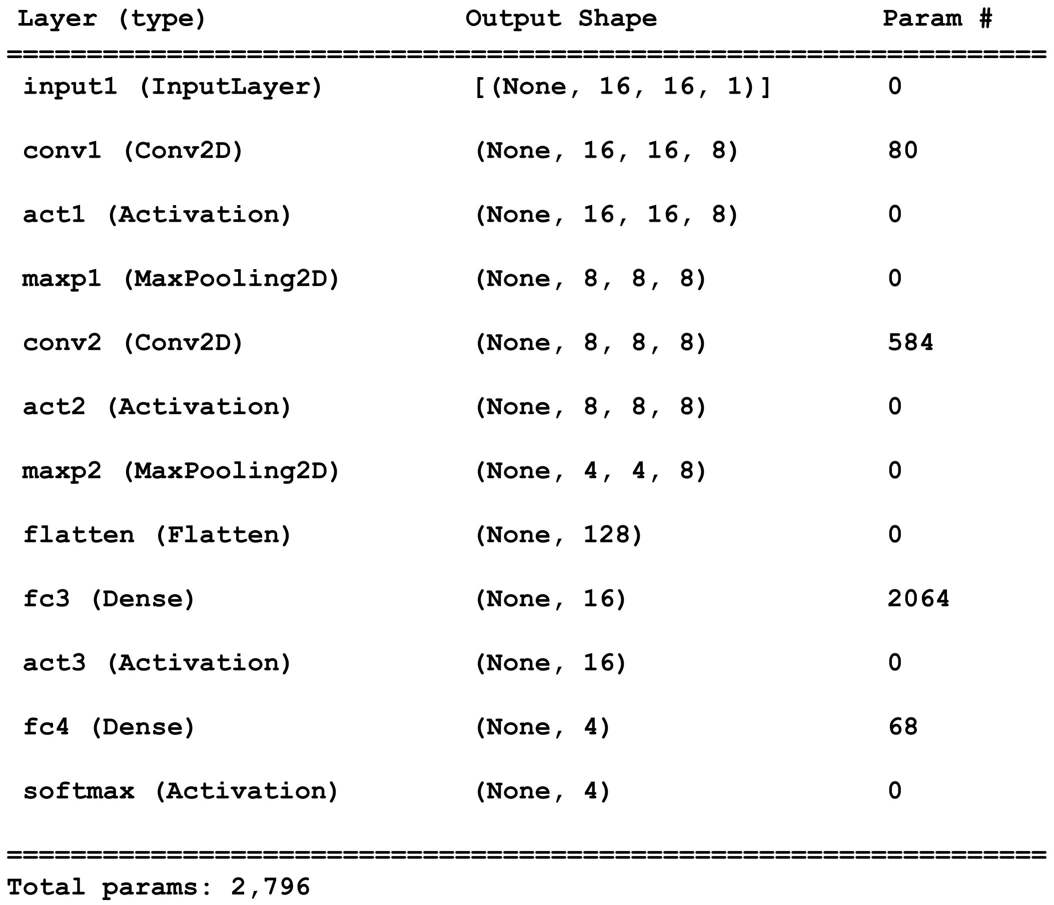
```
Layer (type)                 Output Shape              Param #
=================================================================
input1 (InputLayer)          [(None, 64)]              0

fc1 (Dense)                  (None, 64)                4160

act1 (Activation)            (None, 64)                0

fc2 (Dense)                  (None, 16)                1040

act2 (Activation)            (None, 16)                0

fc3 (Dense)                  (None, 4)                 68

softmax (Activation)         (None, 4)                 0

=================================================================
Total params: 5,268
```

NN Architectures: Convolutional Model

- **Input representation: 16x16 images**
- **Output: 4 classes (0, 1, 2, 3+ rings)**
- **Quantization (fixed point):**
  - Weights and biases: 8 bits <8, 1>
  - Activations:16 bits <16, 6>
- **FPGA resource usage (Alveo U200)**
  LUT 5.2%, FF 1.5%, DSP 4.8%,
  BRAM 0.05%
- **Latency: 388 cycles @ 220MHz**
- **Initiation Interval (II): 369 cycles**
- **Throughput: 0.6 MHz**



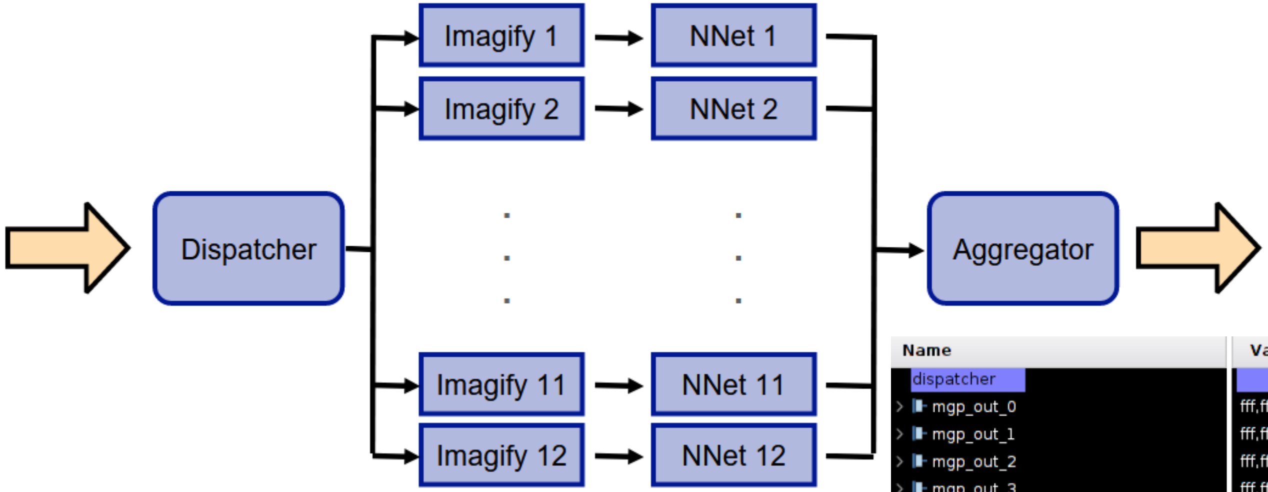




```
Layer (type)                    Output Shape              Param #
=================================================================
input1 (InputLayer)             [(None, 16, 16, 1)]       0

conv1 (Conv2D)                  (None, 16, 16, 8)         80

act1 (Activation)               (None, 16, 16, 8)         0

maxp1 (MaxPooling2D)            (None, 8, 8, 8)           0

conv2 (Conv2D)                  (None, 8, 8, 8)           584

act2 (Activation)               (None, 8, 8, 8)           0

maxp2 (MaxPooling2D)            (None, 4, 4, 8)           0

flatten (Flatten)               (None, 128)               0

fc3 (Dense)                     (None, 16)                2064

act3 (Activation)               (None, 16)                0

fc4 (Dense)                     (None, 4)                 68

softmax (Activation)            (None, 4)                 0

=================================================================
Total params: 2,796
```

Convolutional model – Kernel replication

Throughput is not enough to sustain L0 rate, but we can replicate the network multiple times, also on multiple devices if necessary.

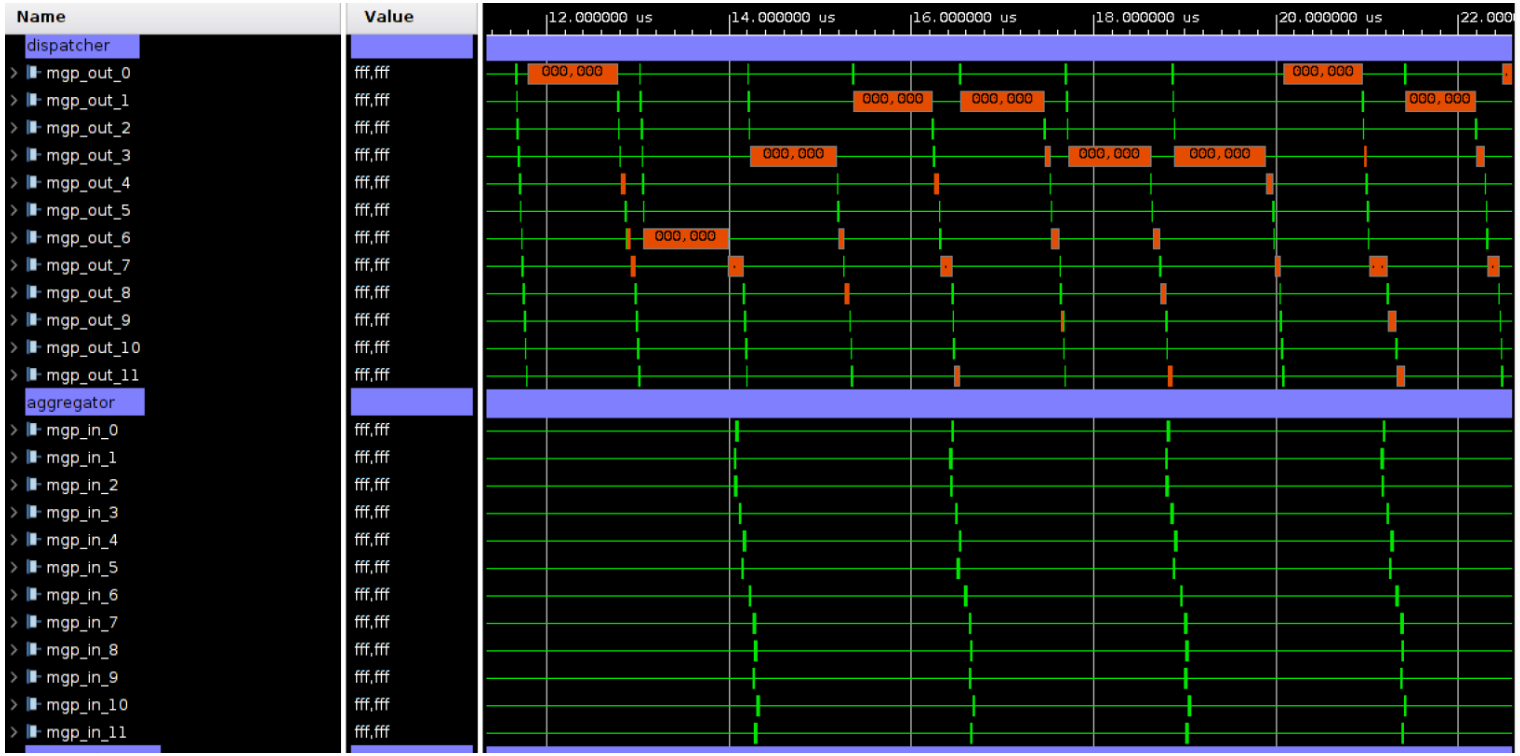


Resources usage for 12 replicas:

- LUT 74%
- FF 17%
- DSP 61%
- BRAM 1.4%

Processing time @220MHz: 137 ns per event

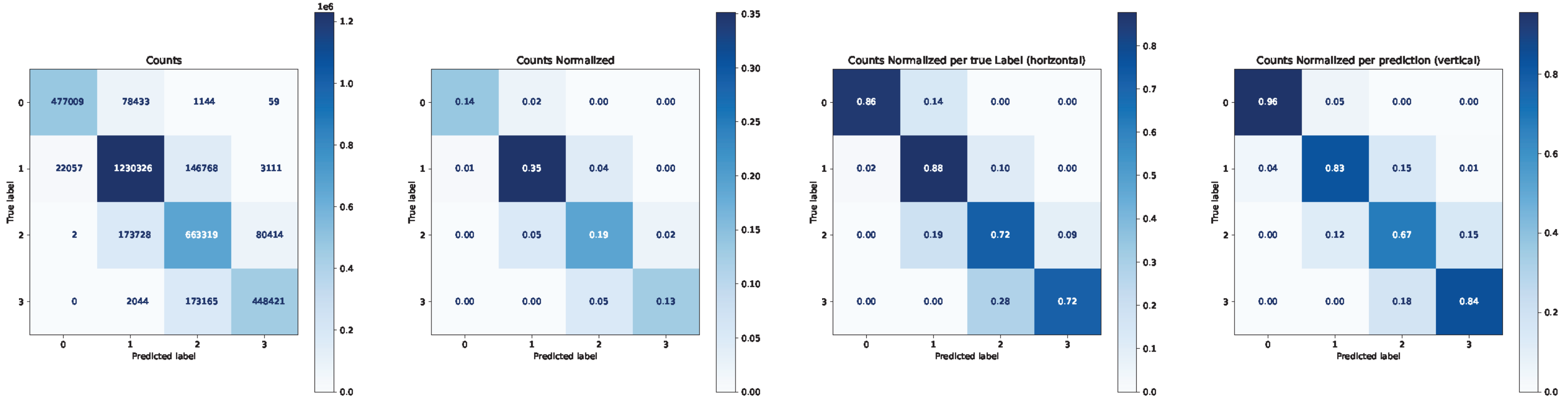
Processing throughput: 7.2 MHz



Name	Value
dispatcher	
mgp_out_0	fff,fff
mgp_out_1	fff,fff
mgp_out_2	fff,fff
mgp_out_3	fff,fff
mgp_out_4	fff,fff
mgp_out_5	fff,fff
mgp_out_6	fff,fff
mgp_out_7	fff,fff
mgp_out_8	fff,fff
mgp_out_9	fff,fff
mgp_out_10	fff,fff
mgp_out_11	fff,fff
aggregator	
mgp_in_0	fff,fff
mgp_in_1	fff,fff
mgp_in_2	fff,fff
mgp_in_3	fff,fff
mgp_in_4	fff,fff
mgp_in_5	fff,fff
mgp_in_6	fff,fff
mgp_in_7	fff,fff
mgp_in_8	fff,fff
mgp_in_9	fff,fff
mgp_in_10	fff,fff
mgp_in_11	fff,fff

Dense Model: results for classification of number of rings

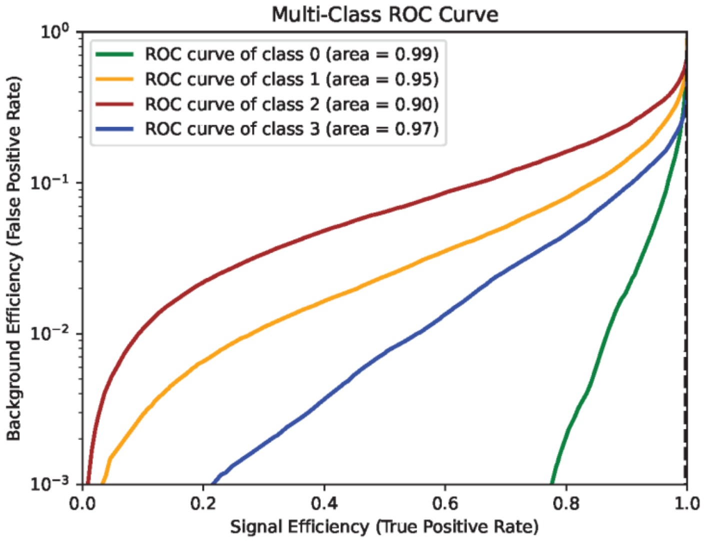
- Trained on 3 Mevents from run 8011, Validated on 3.5 Mevents from run 8893, ground truth label 1



Class 0 (0 rings) Efficiency 85.7 Purity 95.6
Class 1 (1 rings) Efficiency 87.7 Purity 82.9
Class 2 (2 rings) Efficiency 72.3 Purity 67.4
Class 3 (3+ rings) Efficiency 71.9 Purity 84.3

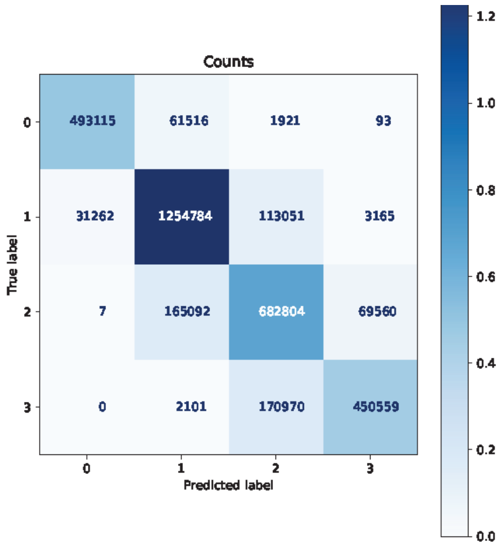
Efficiency = TP / (TP + FN)

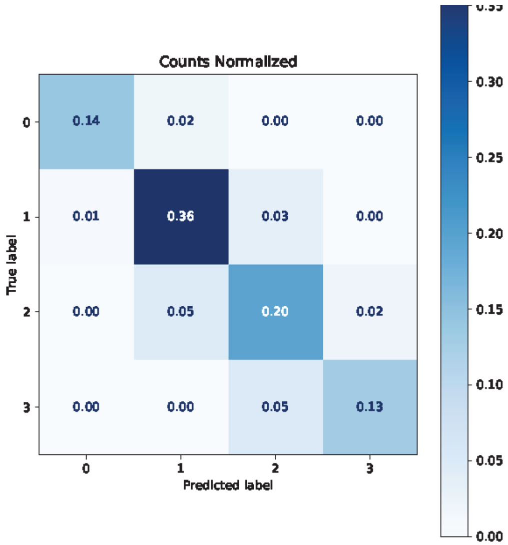
Purity = TP / (TP + FP)

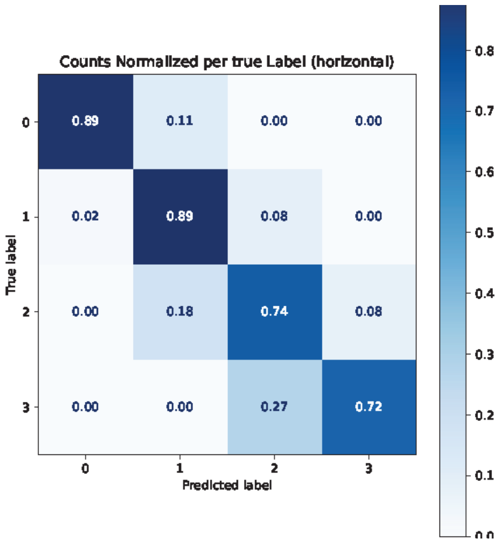


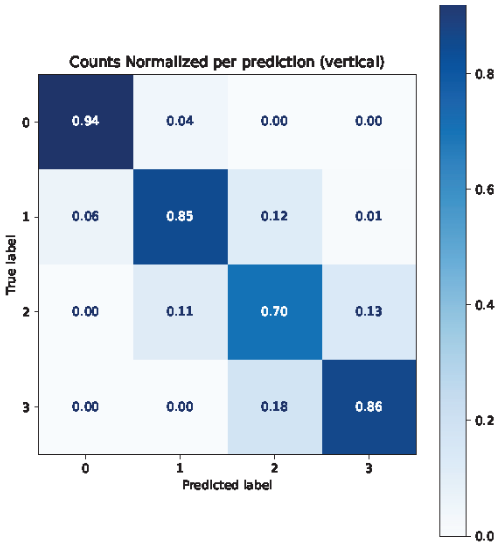
Convolutional Model: results for classification of number of rings

- Trained on 3 Mevents from run 8011, Validated on 3.5 Mevents from run 8893, ground truth label 1

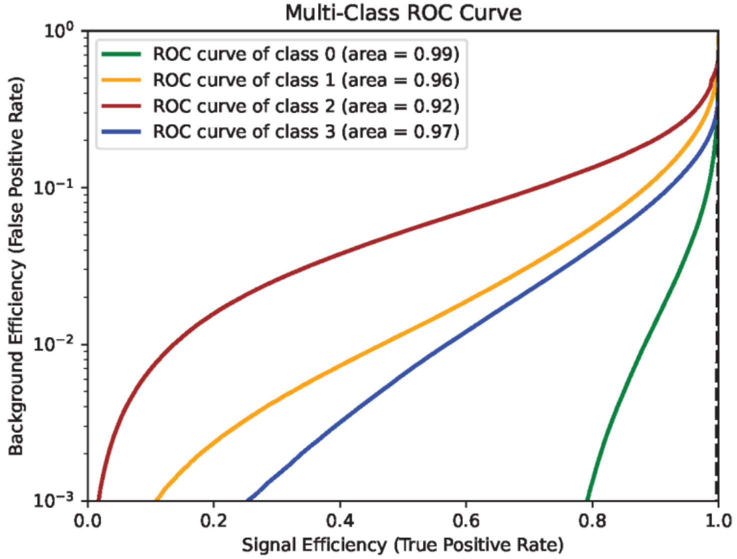








Class 0 (0 rings) Efficiency 88.6 Purity 94.0
Class 1 (1 rings) Efficiency 89.5 Purity 84.6
Class 2 (2 rings) Efficiency 74.4 Purity 70.5
Class 3 (3+ rings) Efficiency 72.2 Purity 86.1



Efficiency = TP / (TP + FN)

Purity = TP / (TP + FP)