Last results on Online Data Reduction System for the ePIC dRICH Detector

Cristian Rossi

INFN Roma, APE Lab

for the ePIC Roma1/2 team

dRICH DAQ Meeting
Sep 5th 2025

dRICH: Data Reduction

Online Signal/ Noise discrimination using ML

- **Signal (i.e. Merged Phys Signal + Bkg)**:
  - **Physics Signal:**
    - e.g DIS
  - **Physics Background:**
    - e/p with beam pipe
    - Synchrotron radiation (currently not including it)

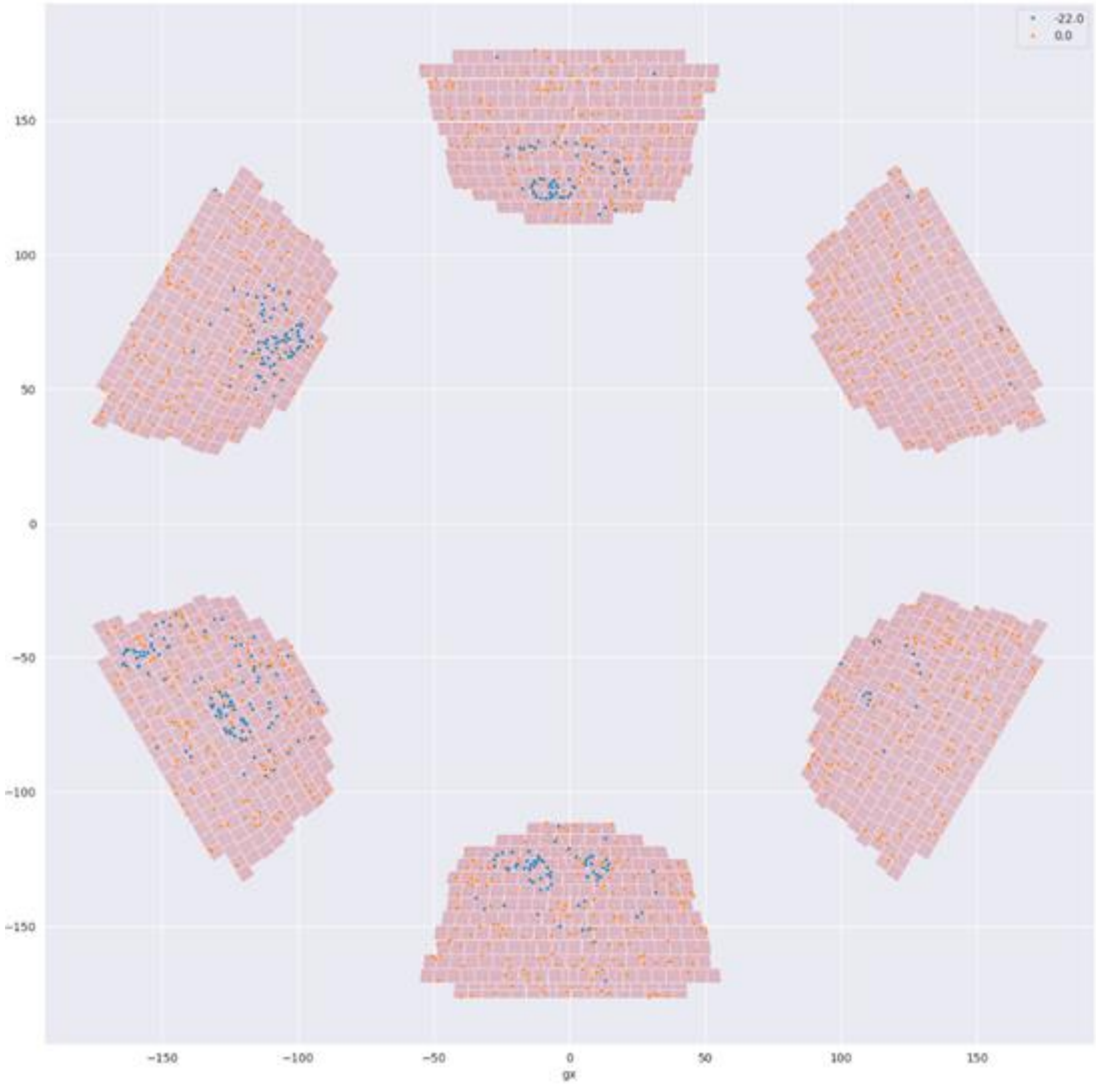
- **SiPM Noise:**
  - Dark current rate (DCR) modelled in the reconstruction stage (*recon.rb* eic-shell method)

ML task:

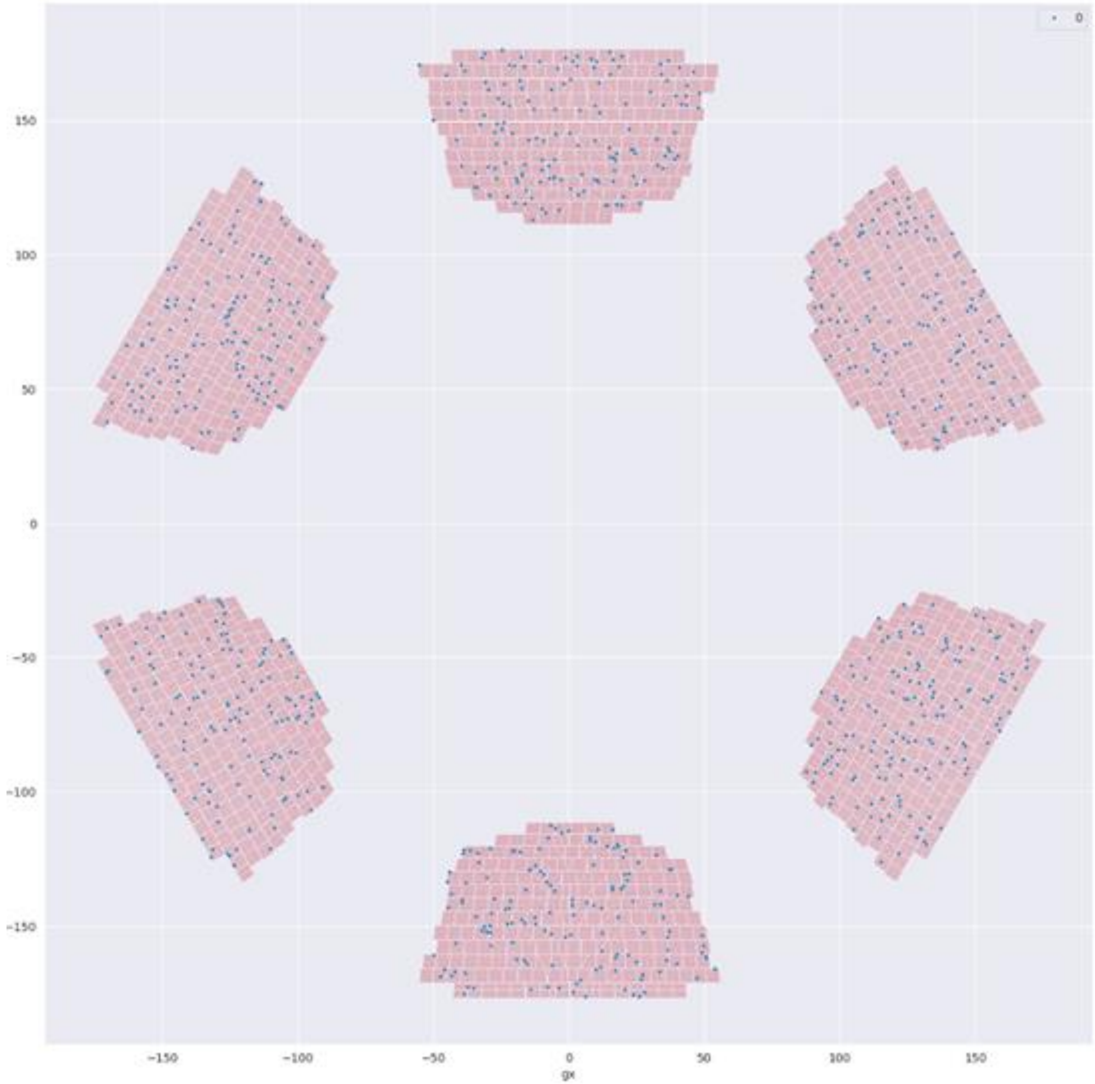
Discriminate between **Noise Only** and **Signal + Noise** events

dRICH: Dataset for training, classes

Phys Signal+Phys Background+Noise

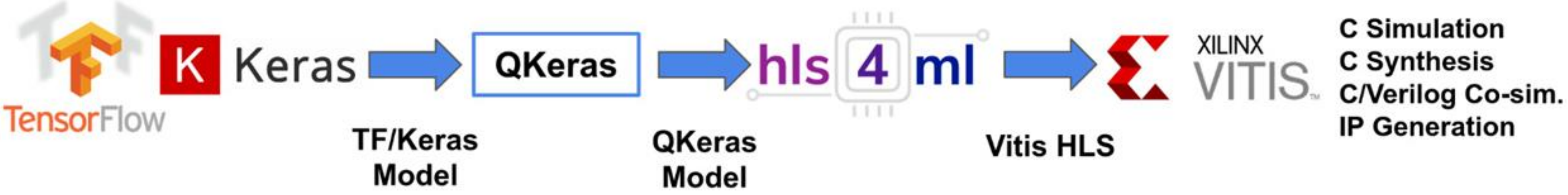


Noise Only



dRICH Data Reduction Stage on FPGA

- **Online** «Noise only» classifier using ML:
  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):
  ⇒ Multi Layer Perceptron (HLS4ML)

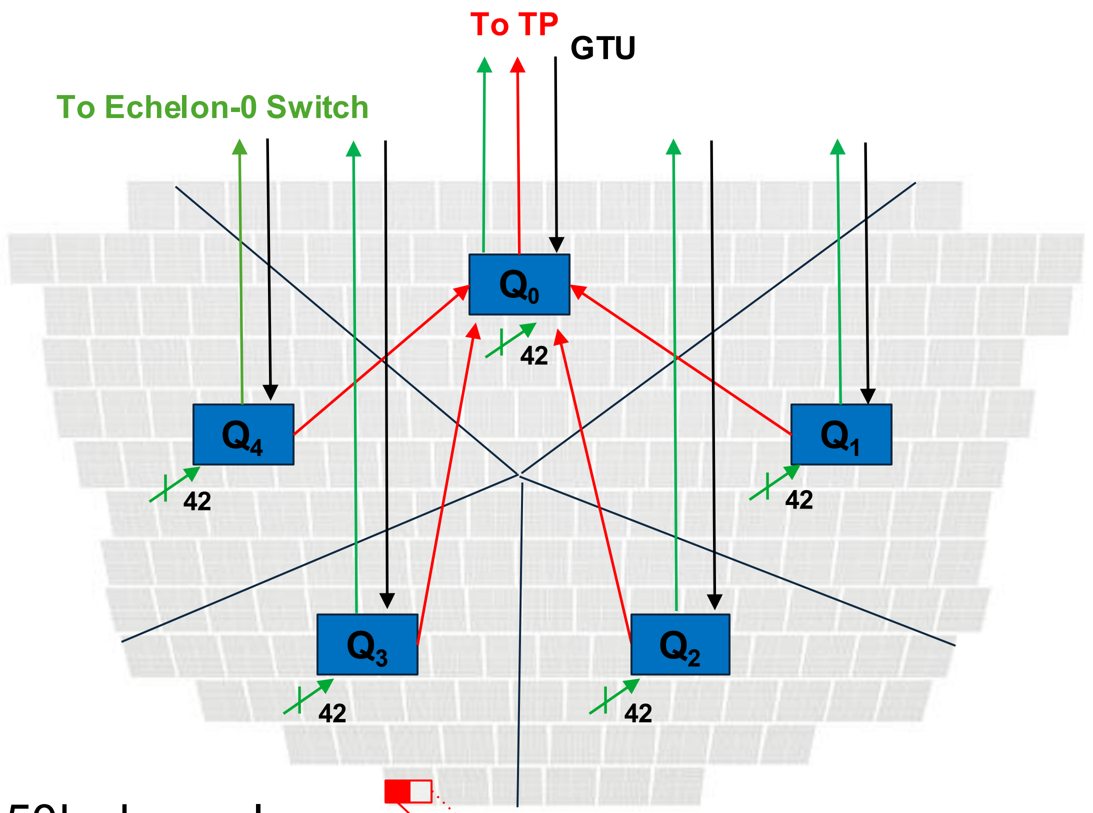


- **Inference** throughput (98.5 MHz) is the main challenge

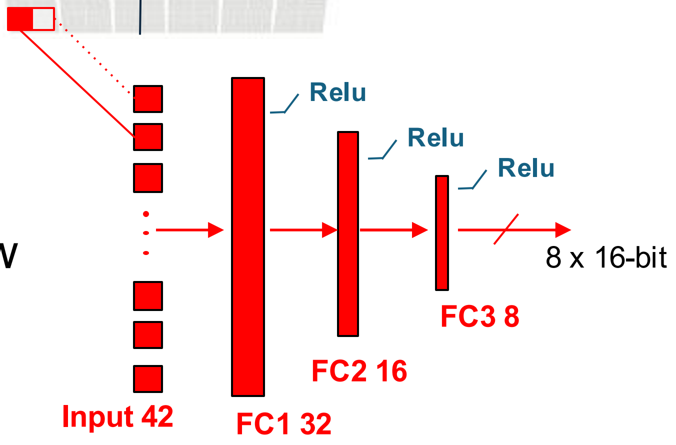
- Deployment on **multiple Felix DAMs** and on an additional FPGA (**TP – Trigger Processor**) directly interconnected

- Possibly integrate with the dRICH Interaction Tagger to boost performance

dRICH Data Reduction on FPGA - Deployment



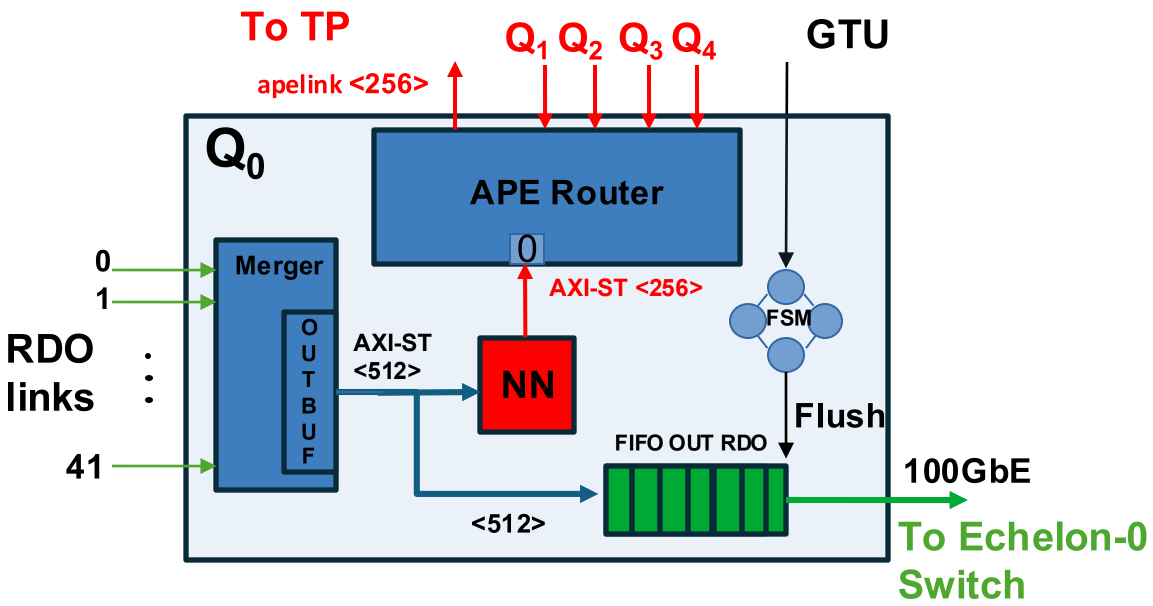
~50k channels per sector

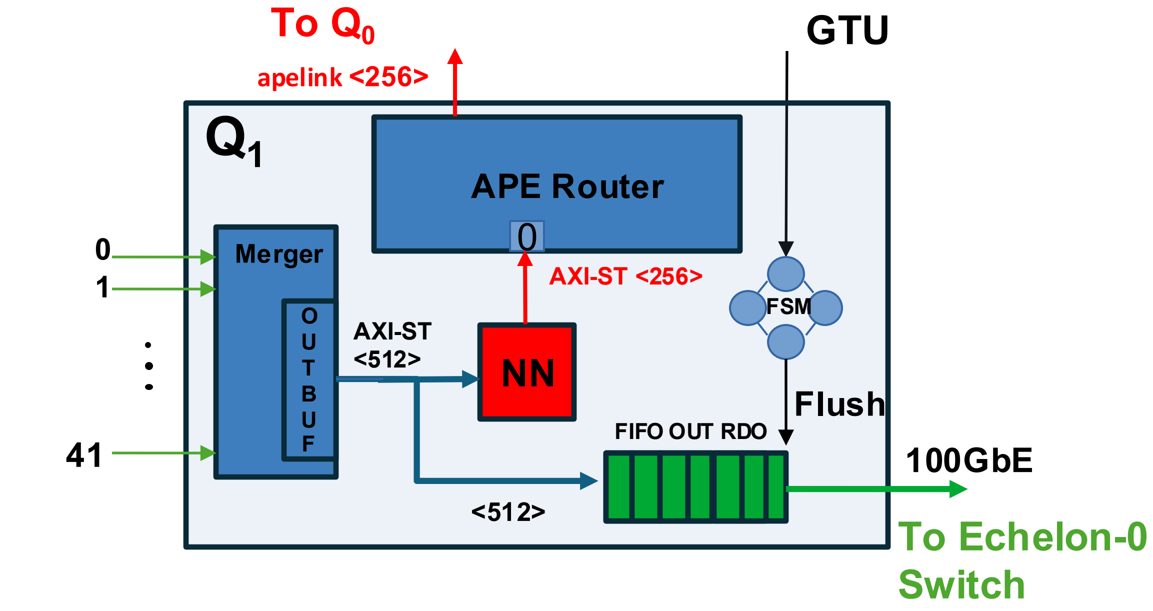


→ RDO flow

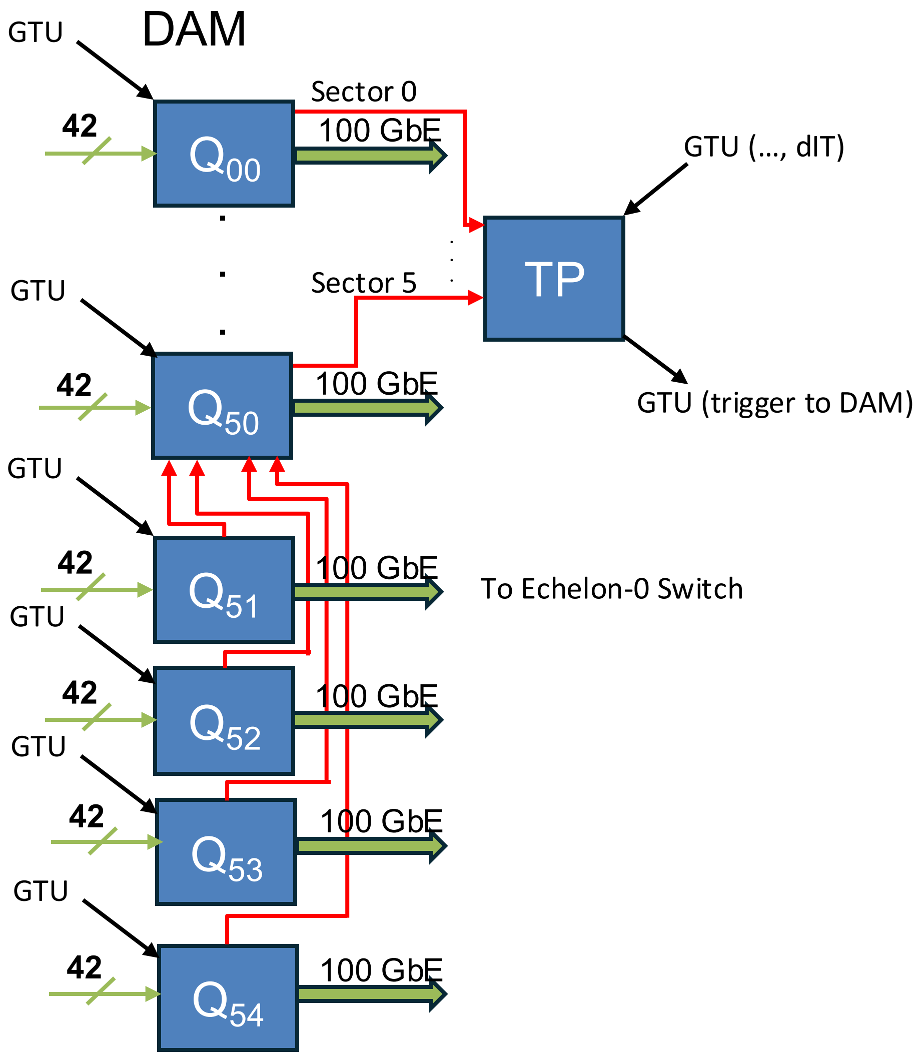
→ PROC flow

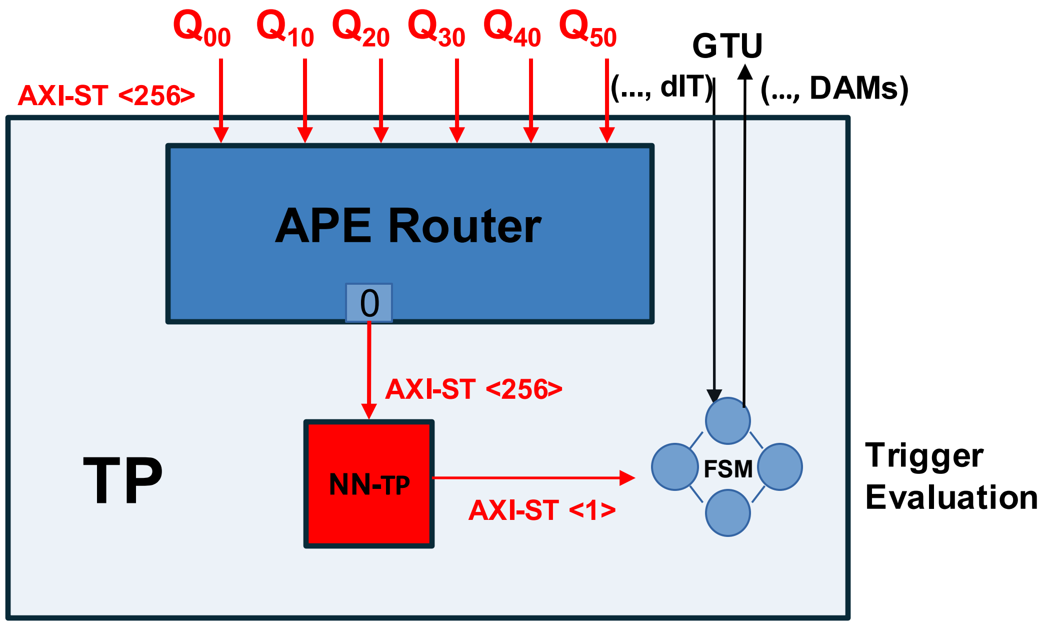
→ GTU

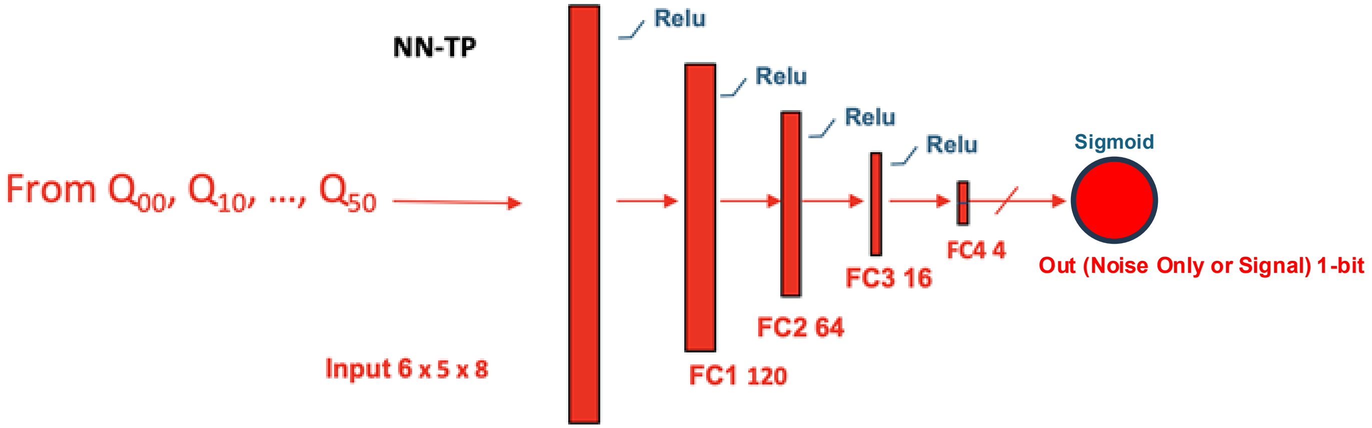




dRICH Data Reduction on FPGA - Deployment

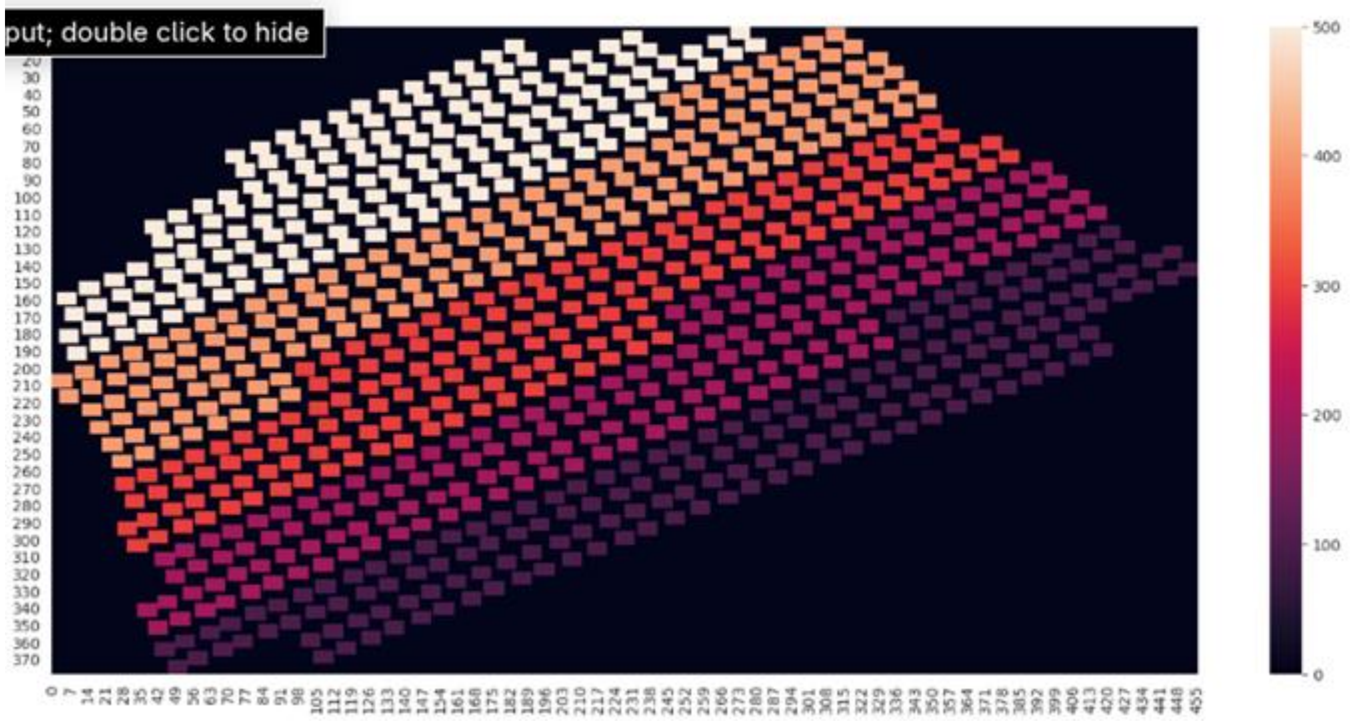


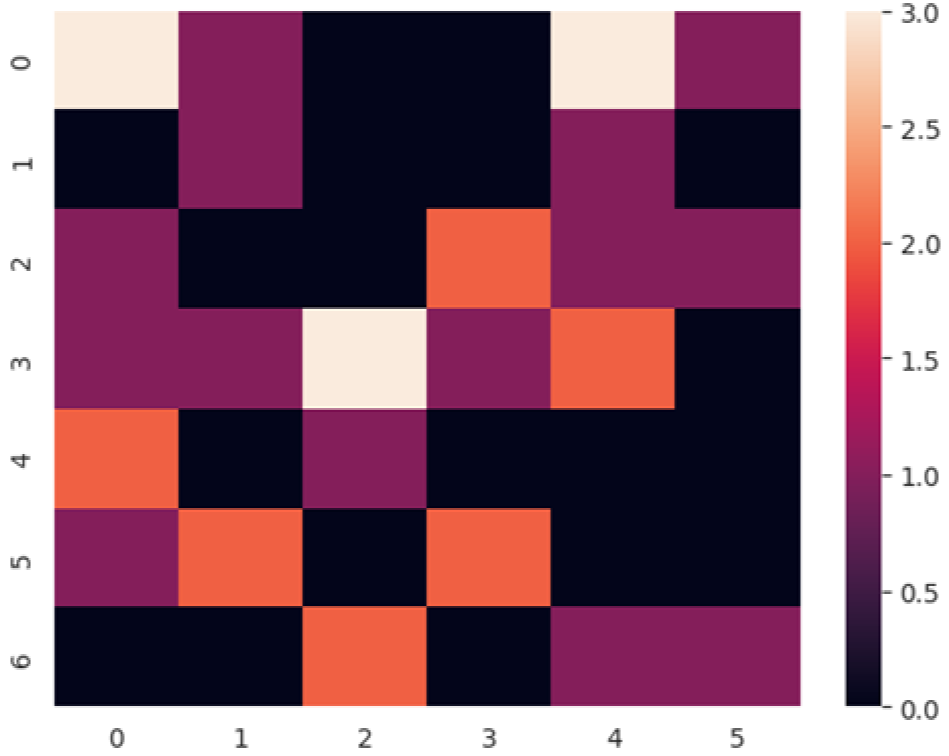



NN-TP
Relu
Relu
Relu
Relu
Sigmoid
From Q_{00}, Q_{10}, ..., Q_{50}
Input 6 x 5 x 8
FC1 120
FC2 64
FC3 16
FC4 4
Out (Noise Only or Signal) 1-bit

dRICH: Data reduction ⇒ Subsectors

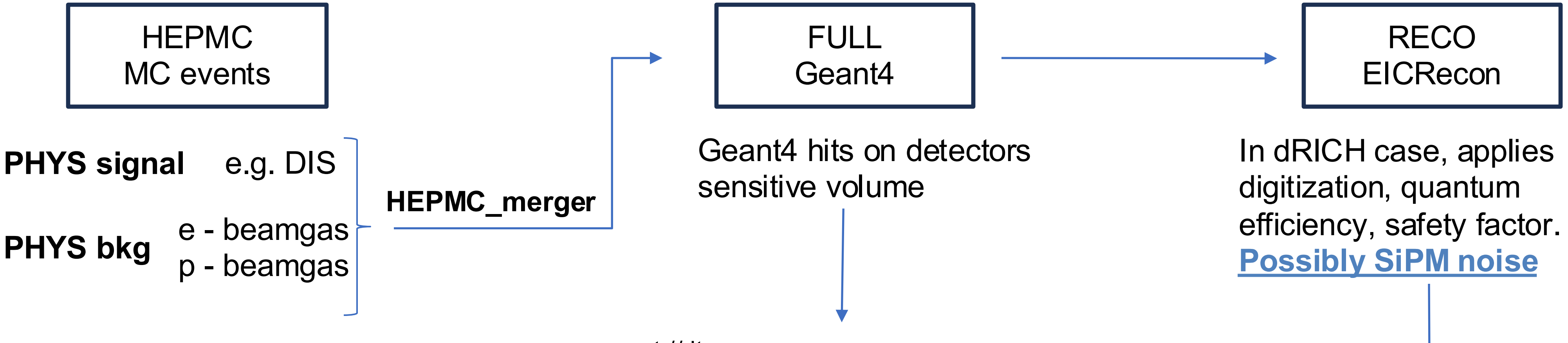
- From our design proposal, we indicate **42 input links for each DAM** occurring into the streaming readout data reduction computation.
  ⇒ This number **(42)** is coherent with the number of expected PDUs per subsector (~210/5 = 42).
  (“Answer to the Ultimate Question of Life,
  the Universe, and Everything”)
- Thus, to cope with the realistic composition of the dRICH hardware readout, we decided to take the **information of each PDU as input** for the respective subsector MLP NN model





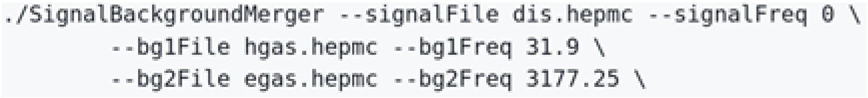
dRICH: Data reduction Dataset

ePIC simulation pipeline:



Geant4 hits on detectors sensitive volume

In dRICH case, applies digitization, quantum efficiency, safety factor. **Possibly SiPM noise**



```
./SignalBackgroundMerger --signalFile dis.hepmc --signalFreq 0 \
        --bg1File hgas.hepmc --bg1Freq 31.9 \
        --bg2File egas.hepmc --bg2Freq 3177.25 \
```

root://dtn-eic.jlab.org//work/eic2/EPIC/FULL/24.05.0/epic_craterlake/BACKGROUNDS/MERGED/HEPMC_merger-1.0.2/10x100/RealisticSignalPerFrame/

By default SiPM noise disabled and not present on root://dtn-eic.jlab.org

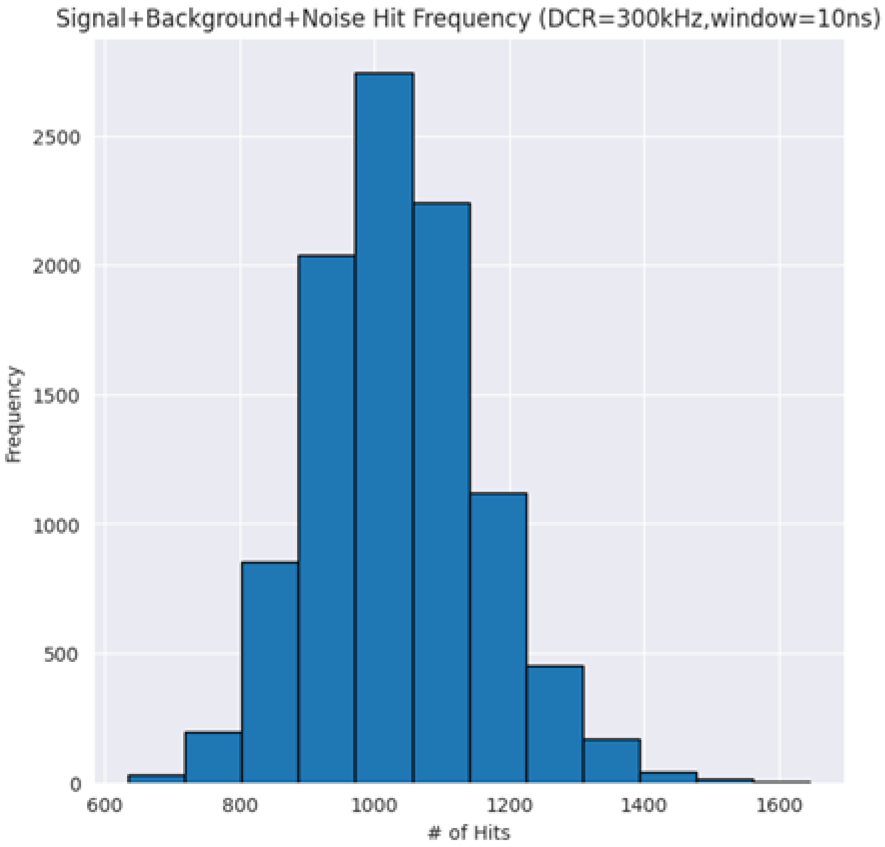
Options:

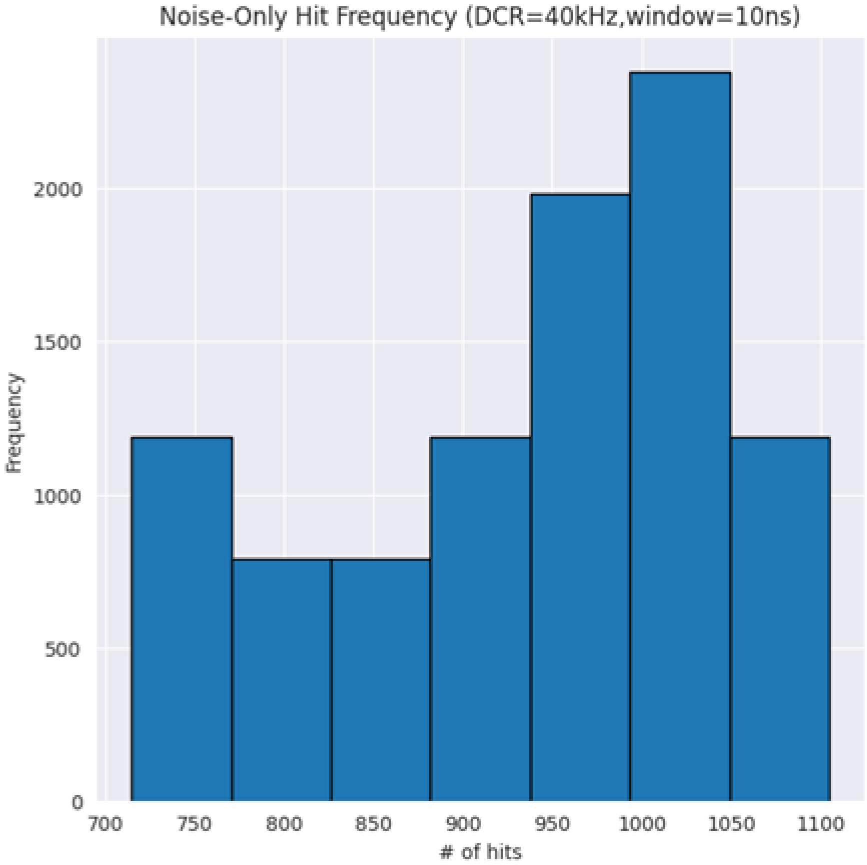
- Start from Merged FULL root files available on server and enable noise at RECO stage using *drich-dev/recon.rb* with configs (but only ~ 7k events present on dtn-eic)
- **Run the entire simulation pipeline ourselves, starting from HEPMC files.**
  - Up to now we have produced 800k events to train and test our ML models

⇒ Various noise rates and noise models for each generated dataset

dRICH Data reduction: Noise hits distribution

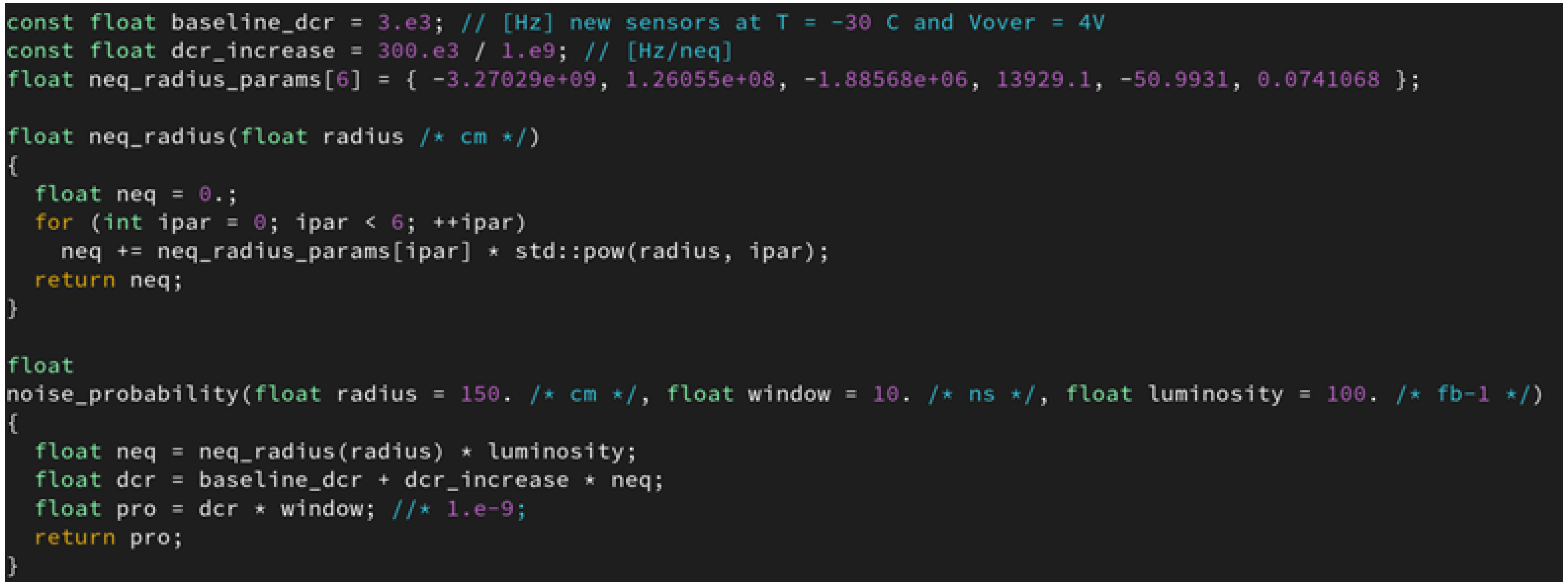
- **Gaussian** dark current SiPM **noise hits distribution,** obtained by modifying EICRecon source:
  - avg = noiseRate*noiseTimeWindow*NumberOfSiPMsDRICH
  - sigma = 0.1*avg
  - noiseTimeWindow = 10 ns (no shutter)





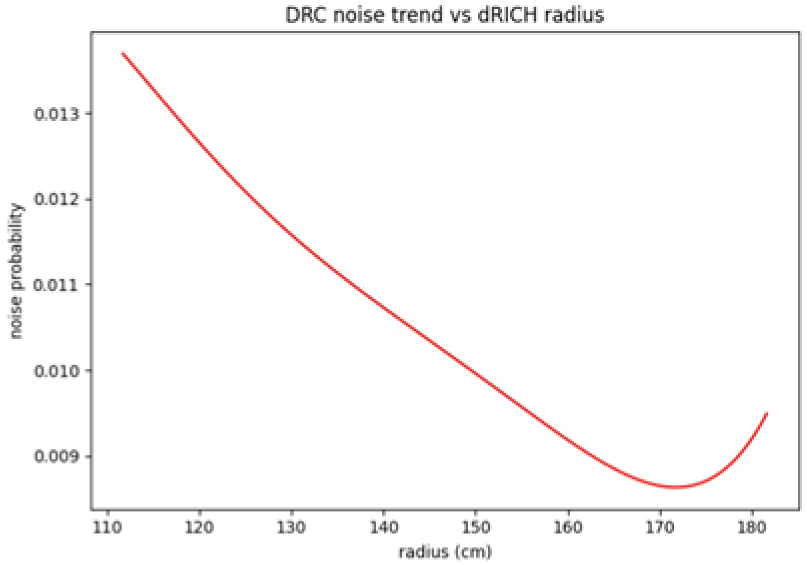
dRICH Data reduction: Noise hits distribution

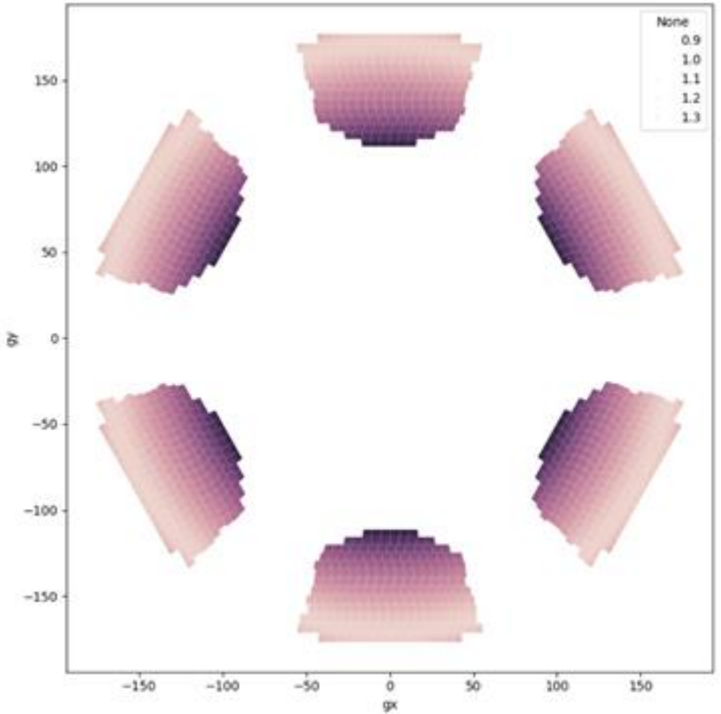
- Dark current SiPM **noise hits distribution,** obtained by introducing Dark Count probability of single dRICH SiPM with a dependence on its **radial distance from the detector z-axis** and on the **integrated luminosity**
  ⇒ Implemented in EICRecon digitization step (new flag to enable new model noise)



```
const float baseline_dcr = 3.e3; // [Hz] new sensors at T = -30 C and Vover = 4V
const float dcr_increase = 300.e3 / 1.e9; // [Hz/neq]
float neq_radius_params[6] = { -3.27029e+09, 1.26055e+08, -1.88568e+06, 13929.1, -50.9931, 0.0741068 };

float neq_radius(float radius /* cm */)
{
  float neq = 0.;
  for (int ipar = 0; ipar < 6; ++ipar)
    neq += neq_radius_params[ipar] * std::pow(radius, ipar);
  return neq;
}

float
noise_probability(float radius = 150. /* cm */, float window = 10. /* ns */, float luminosity = 100. /* fb-1 */)
{
  float neq = neq_radius(radius) * luminosity;
  float dcr = baseline_dcr + dcr_increase * neq;
  float pro = dcr * window; //* 1.e-9;
  return pro;
}
```

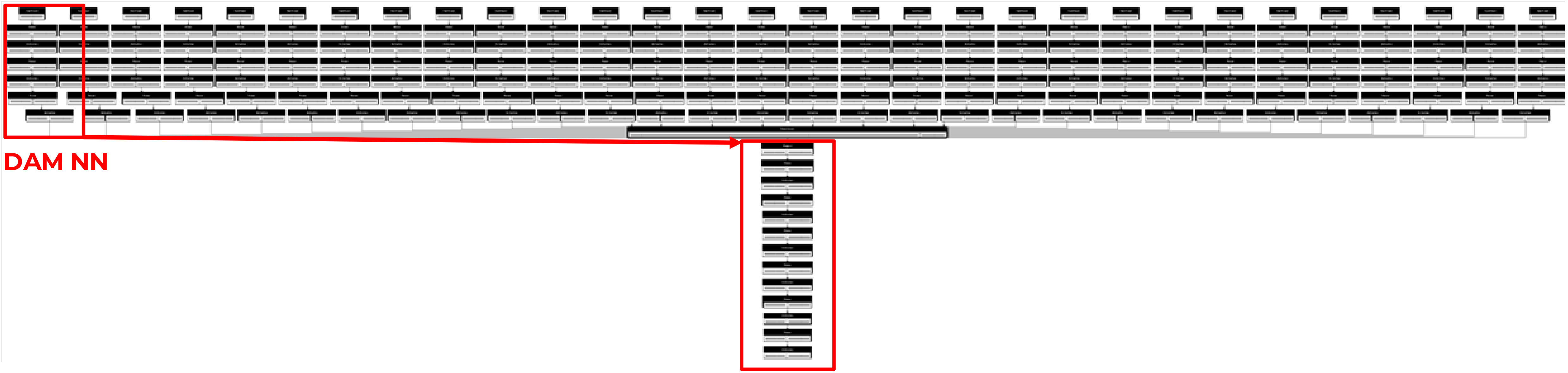
(R. Preghenella’s contribution)





dRICH Data reduction: Tensorflow-Keras Model definition

- To be coherent with the hardware design composition of the proposed system, we trained **30** (# of subsectors x #number of sectors) **concatenated MLP networks** into a single MLP model to be deployed on 30 DAM FPGAs + 1 TP FGPA

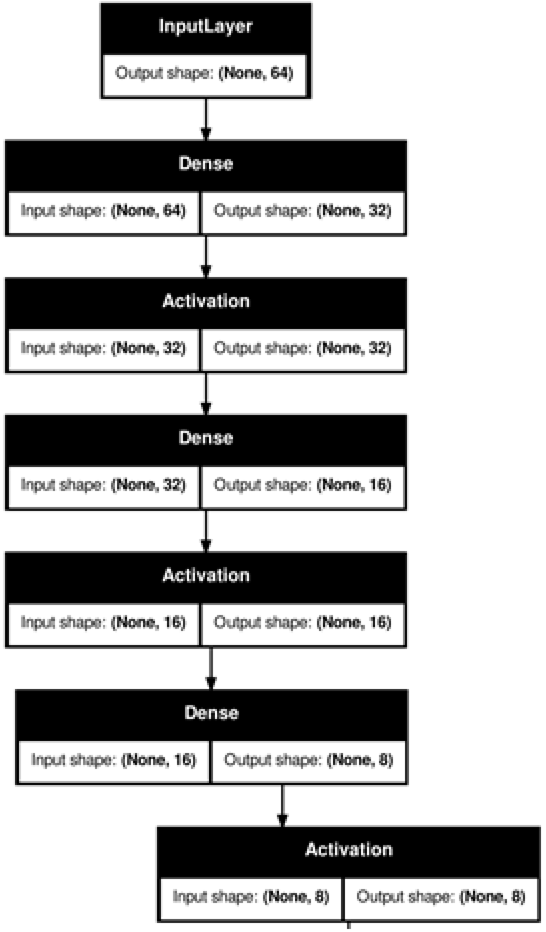


DAM NN

Trigger Processor NN

«Distributed MLP Model»

Distributed MLP Tensorflow Model



InputLayer
Output shape: (None, 64)

Dense
Input shape: (None, 64) | Output shape: (None, 32)

Activation
Input shape: (None, 32) | Output shape: (None, 32)

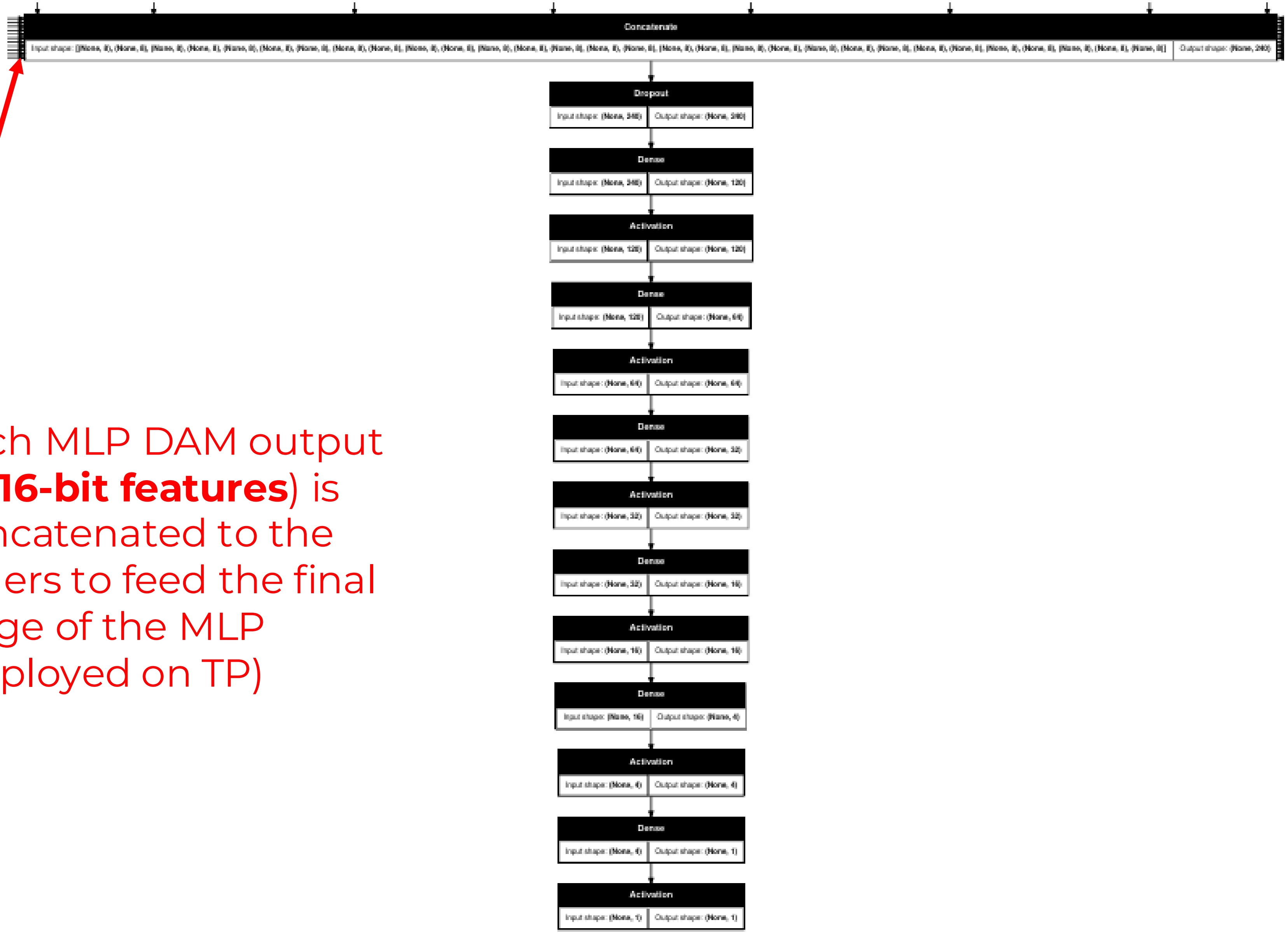
Dense
Input shape: (None, 32) | Output shape: (None, 16)

Activation
Input shape: (None, 16) | Output shape: (None, 16)

Dense
Input shape: (None, 16) | Output shape: (None, 8)

Activation
Input shape: (None, 8) | Output shape: (None, 8)

Each MLP DAM output (**8x16-bit features**) is concatenated to the others to feed the final stage of the MLP (deployed on TP)



Concatenate
Input shape: [(None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8), (None, 8)] | Output shape: (None, 240)

Dropout
Input shape: (None, 240) | Output shape: (None, 240)

Dense
Input shape: (None, 240) | Output shape: (None, 120)

Activation
Input shape: (None, 120) | Output shape: (None, 120)

Dense
Input shape: (None, 120) | Output shape: (None, 64)

Activation
Input shape: (None, 64) | Output shape: (None, 64)

Dense
Input shape: (None, 64) | Output shape: (None, 32)

Activation
Input shape: (None, 32) | Output shape: (None, 32)

Dense
Input shape: (None, 32) | Output shape: (None, 16)

Activation
Input shape: (None, 16) | Output shape: (None, 16)

Dense
Input shape: (None, 16) | Output shape: (None, 4)

Activation
Input shape: (None, 4) | Output shape: (None, 4)

Dense
Input shape: (None, 4) | Output shape: (None, 1)

Activation
Input shape: (None, 1) | Output shape: (None, 1)

dRICH Data reduction: model training & validation

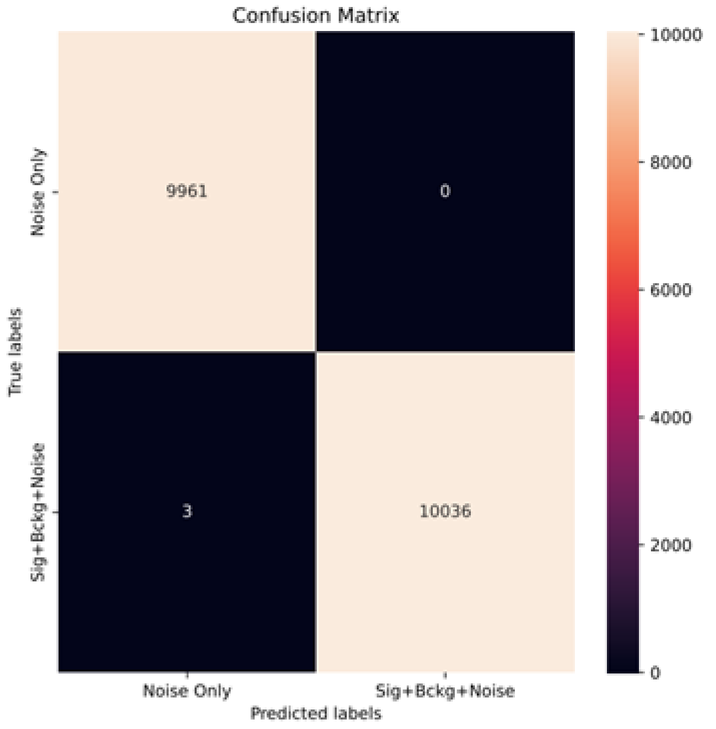
- We trained the 30 MLP DAM models concatenated to the single MLP TP model by using 100k Signal+Background+Noise and 100k Noise Only event

- **200k balanced dataset** (90% training set, 8% testing set, 2% validation set) for any of the considered noise hits distribution models, varying their typical parameters:
  - **Gaussian model:**
    - noiseRate = **40 kHz**, timeWindow = 10ns;
    - noiseRate = **100 kHz**, timeWindow = 10ns;
    - noiseRate = **200 kHz**, timeWindow = 10ns;
    - noiseRate = **300 kHz**, timeWindow = 10ns;
  - **Radial-dependent model:**
    - luminosity = **25 fb-1**, timeWindow = 10ns;
    - luminosity = **50 fb-1**, timeWindow = 10ns;
    - luminosity = **100 fb-1**, timeWindow = 10ns;

G Model performance @ noiseRate = 300 KHz

Keras model

window = 10ns



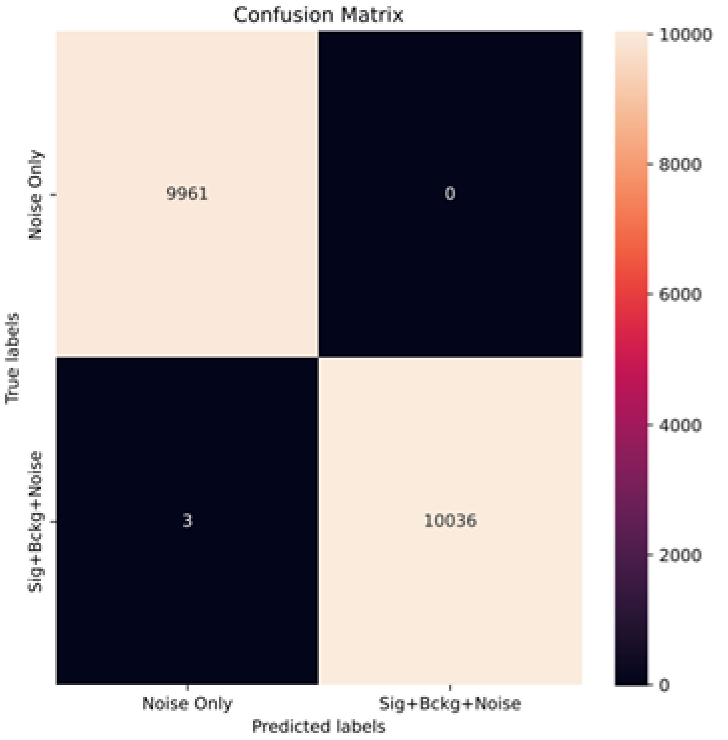
❑ Accuracy = (TP+TN)/(TP+TN+FP+FN)= 0.999

❑ Purity = TP/(TP+FP) = 0.999

❑ Recall = TP/(TP+FN) = 1.000

Model Quantization

- Inputs, Activations: fixed point<16,6>
- Weights, Biases: fixed point<8,1>

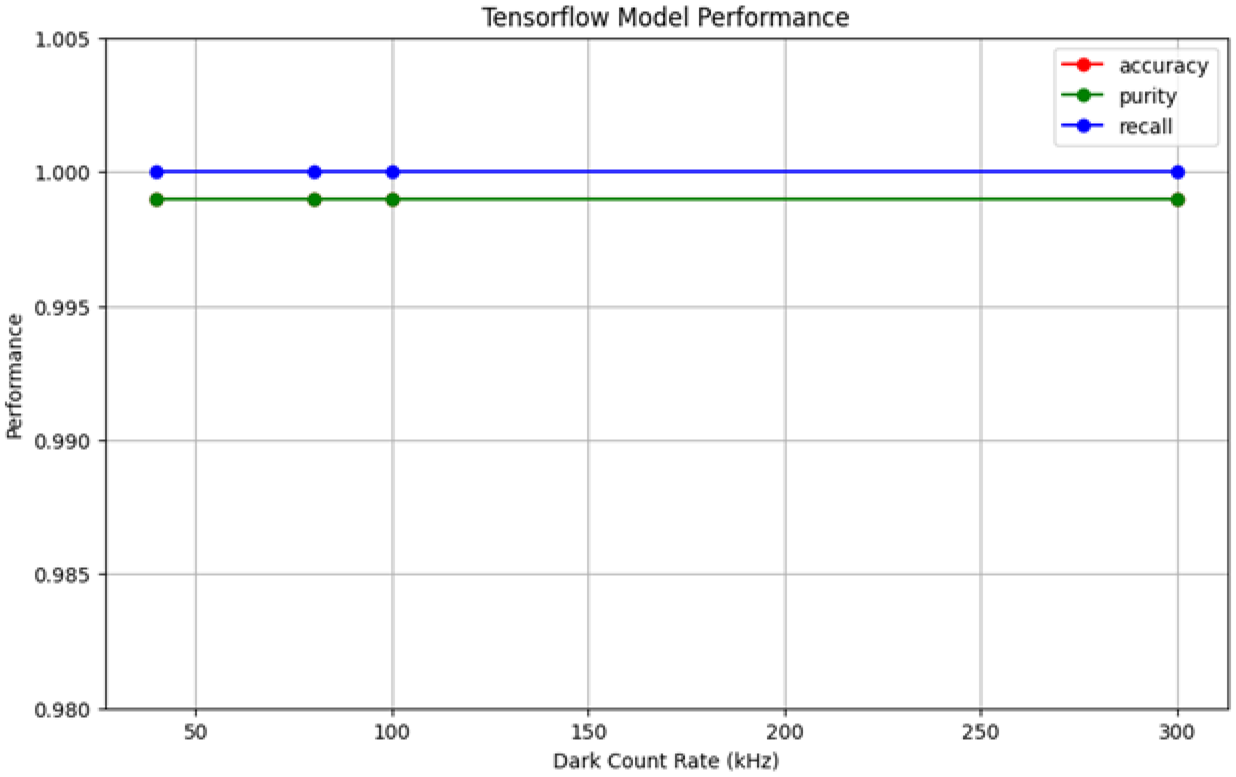


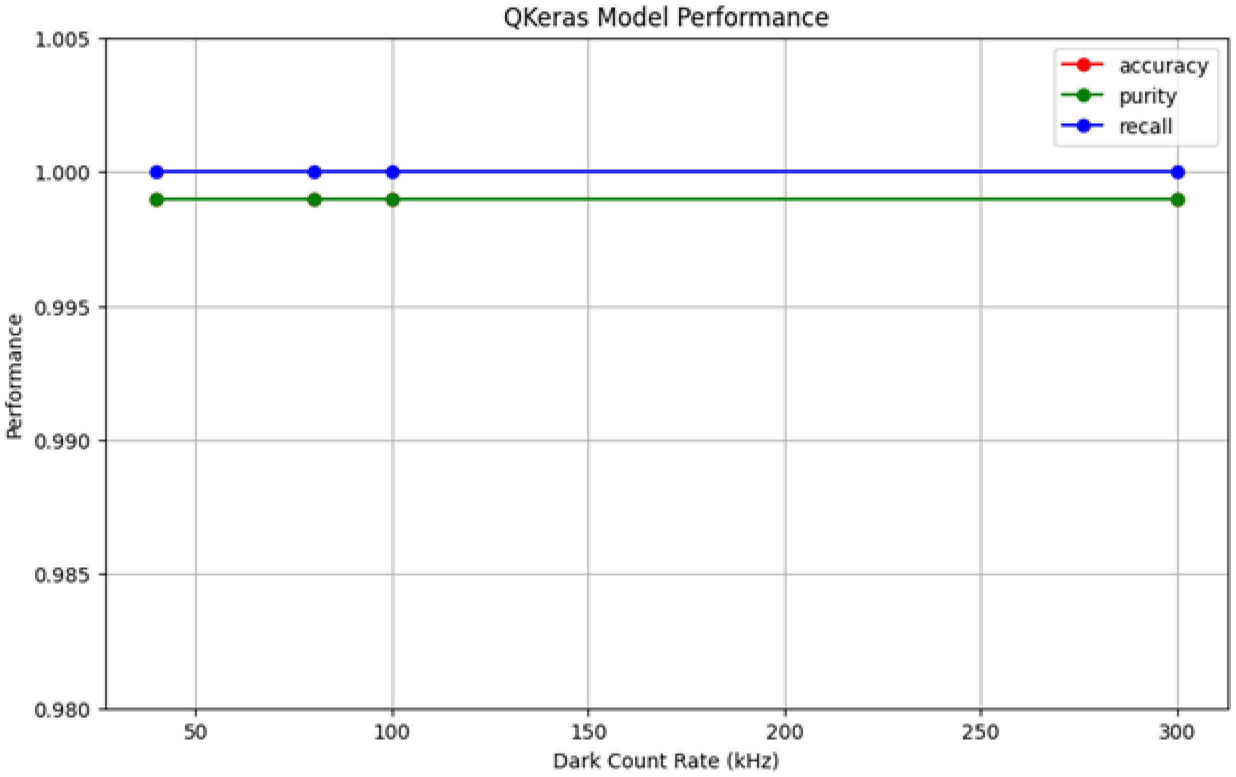
❑ Accuracy = (TP+TN)/(TP+TN+FP+FN) = 0.999

❑ Purity = TP/(TP+FP) = 0.999

❑ Recall = TP/(TP+FN) = 1.000

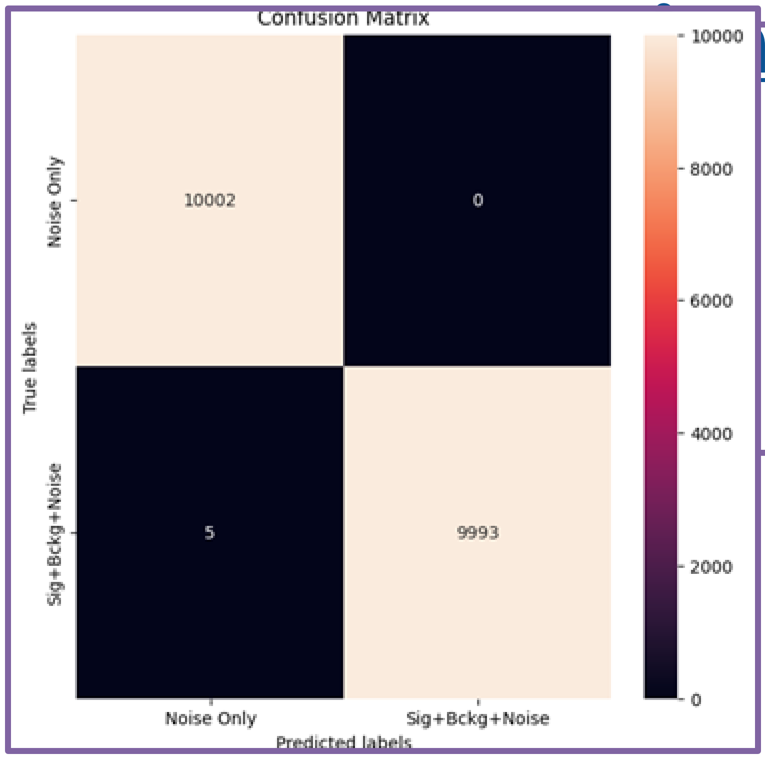
Gaussian Model performance: summary





R Model performance @ luminosity = 100fb-1

Keras model

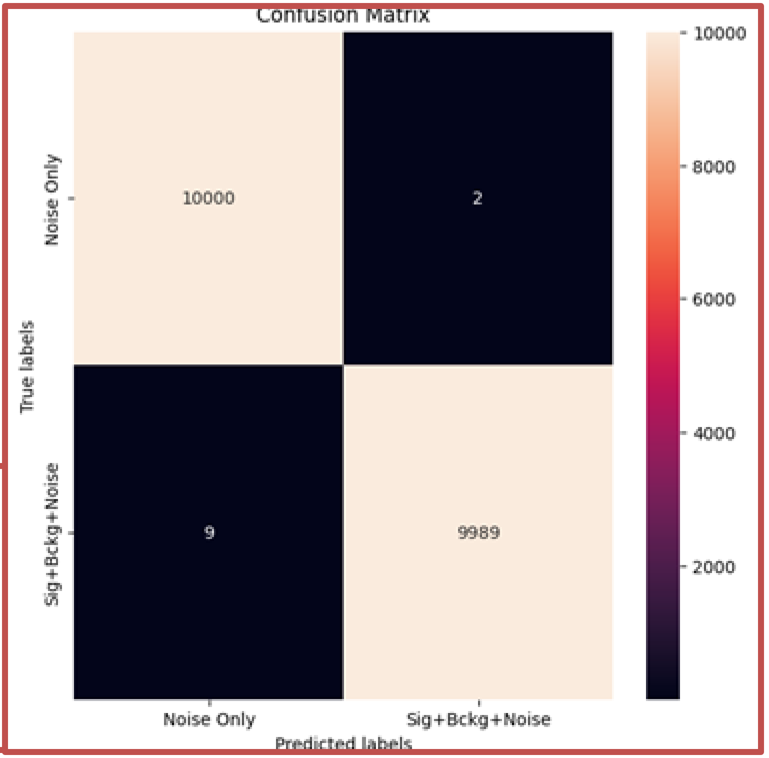


ewindow = 10ns

- ❑ Accuracy = (TP+TN)/(TP+TN+FP+FN)= 0.999
- ❑ Purity = TP/(TP+FP) = 0.999
- ❑ Recall = TP/(TP+FN) = 1.000

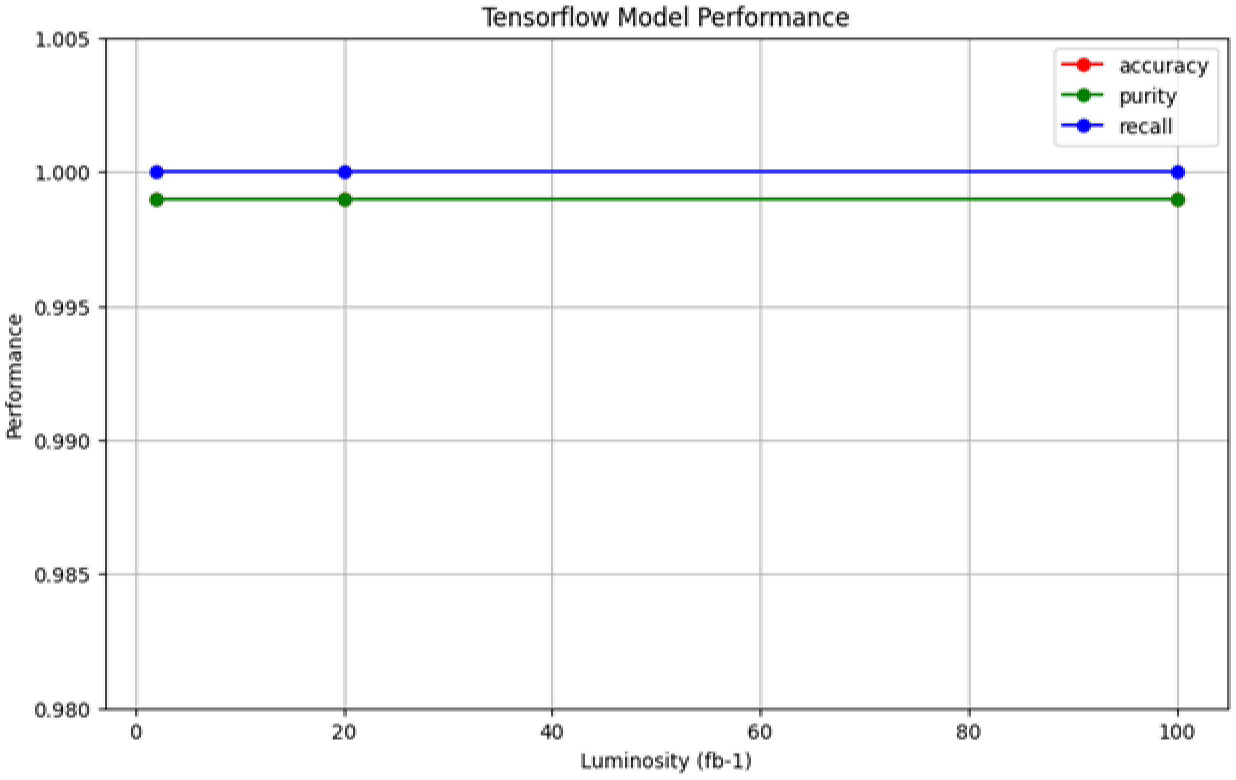
Model Quantization

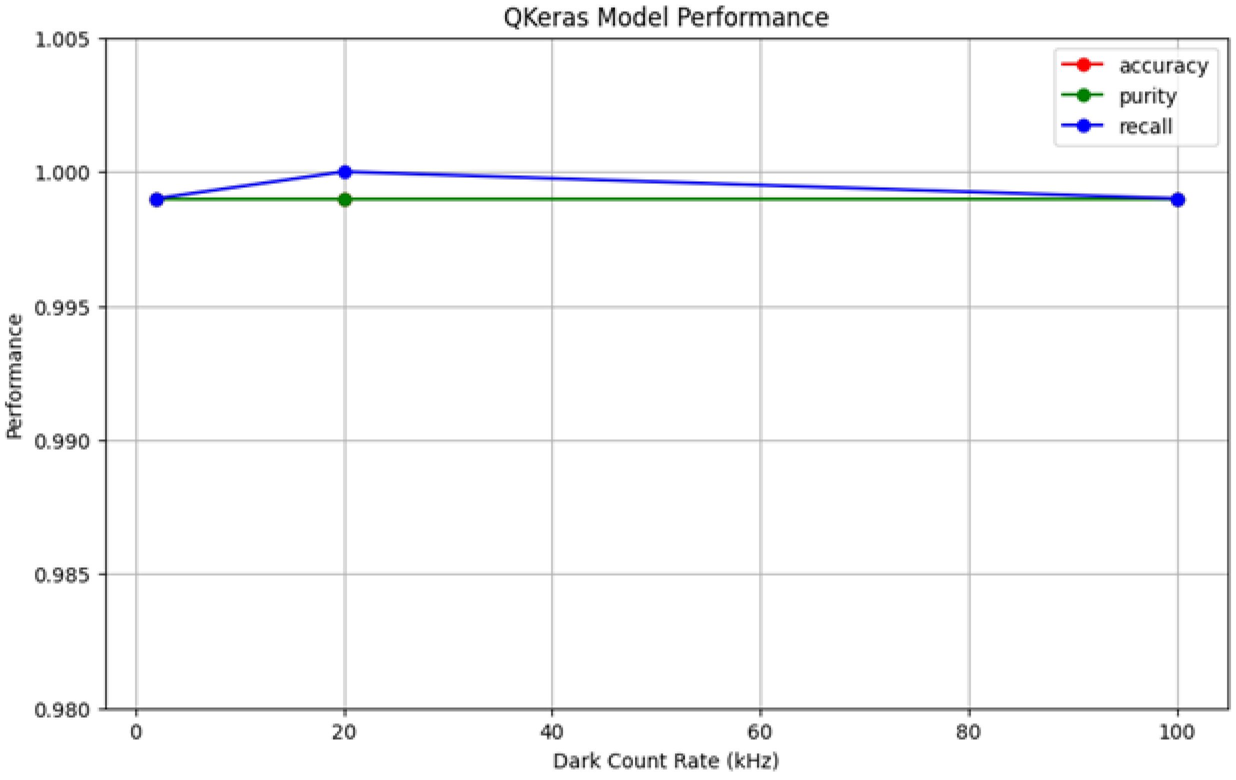
- Inputs, Activations: fixed point<16,6>
- Weights, Biases: fixed point<8,1>



- ❑ Accuracy = (TP+TN)/(TP+TN+FP+FN) = 0.999
- ❑ Purity = TP/(TP+FP) = 0.999
- ❑ Recall = TP/(TP+FN) = 0.999

Radial-dependent Model performance: summary



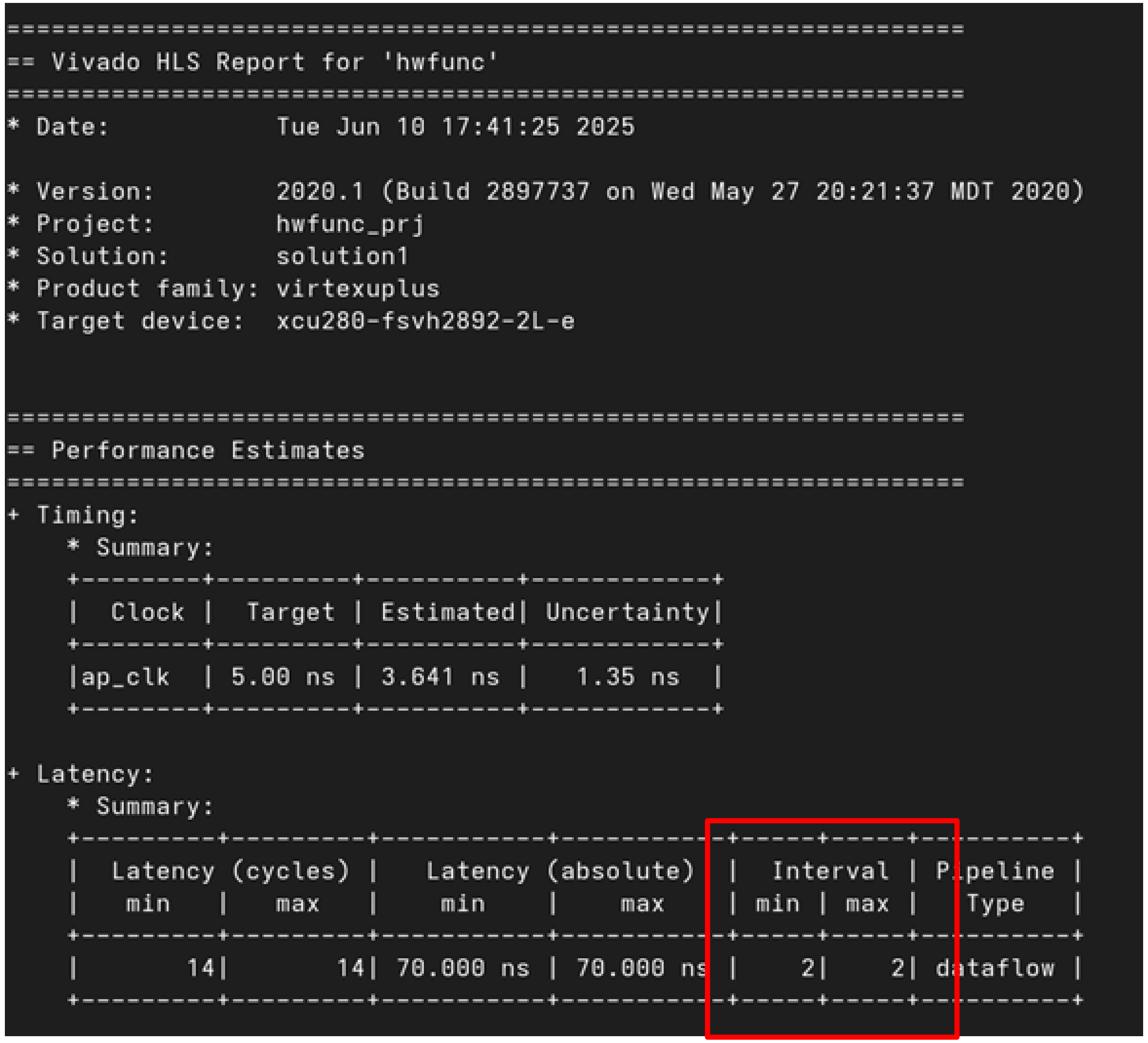


dRICH Data Reduction: HLS4ML ⇒ HW Synthesis for DAM MLP NN

⇒ To correctly synthetize the model at 200 MHz of operational clock, we used a **REUSE FACTOR = 1**, obtaining an instantiation interval **II = 2 clock cycles**

⇒ **Throughput = 100 MHz**

(this results are obtained after synthesis of the HLS4ML code on Xilinx Alveo U280, used in our lab as a starting testbed in order to validate hw implementation of a simple DAM+TP setup)

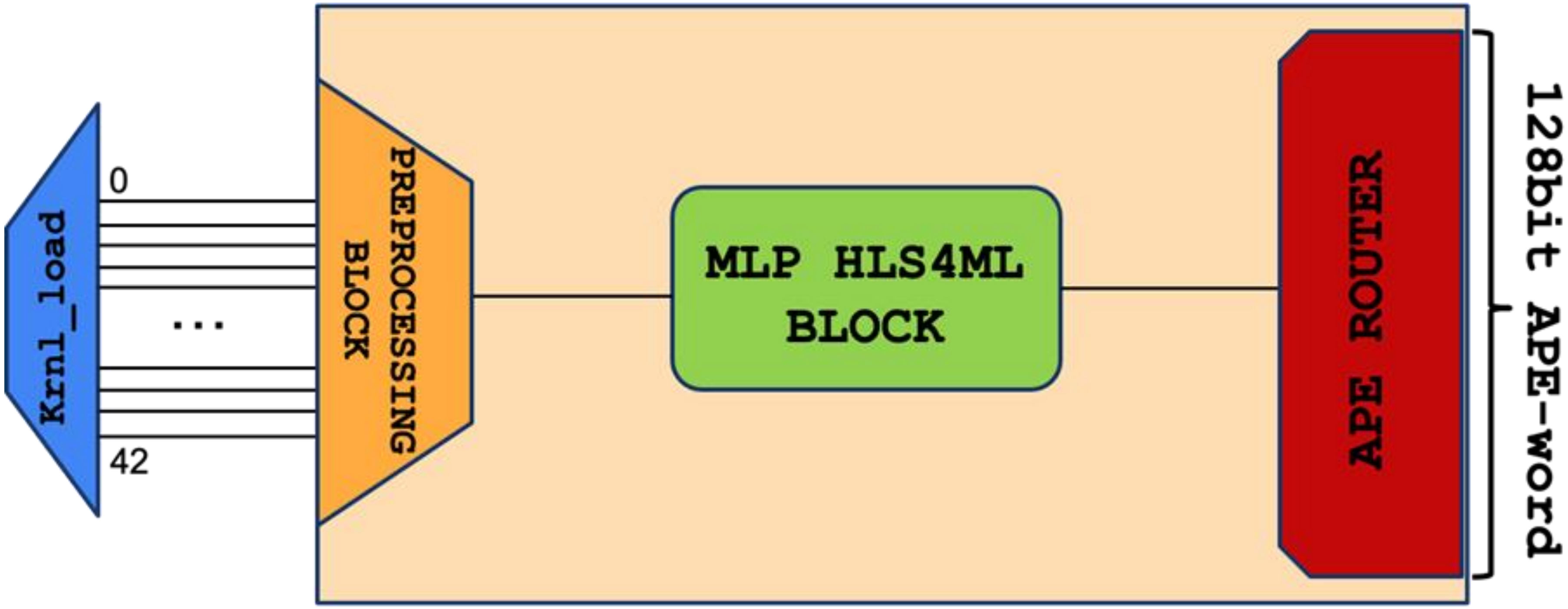


```
================================================================
== Vivado HLS Report for 'hwfunc'
================================================================
* Date:           Tue Jun 10 17:41:25 2025

* Version:        2020.1 (Build 2897737 on Wed May 27 20:21:37 MDT 2020)
* Project:        hwfunc_prj
* Solution:       solution1
* Product family: virtexuplus
* Target device:  xcu280-fsvh2892-2L-e


================================================================
== Performance Estimates
================================================================
+ Timing:
    * Summary:
    +--------+---------+----------+------------+
    |  Clock |  Target | Estimated| Uncertainty|
    +--------+---------+----------+------------+
    |ap_clk  | 5.00 ns | 3.641 ns |   1.35 ns  |
    +--------+---------+----------+------------+

+ Latency:
    * Summary:
    +---------+---------+-----------+-----------+-----+-----+----------+
    |  Latency (cycles) |  Latency (absolute)   |  Interval | Pipeline |
    |   min   |   max   |    min    |    max    | min | max |   Type   |
    +---------+---------+-----------+-----------+-----+-----+----------+
    |       14|       14| 70.000 ns | 70.000 ns |    2|    2| dataflow |
    +---------+---------+-----------+-----------+-----+-----+----------+
```

dRICH Data Reduction: HW Implementation (U280)

- ➔ To validate the correct interaction between the **MLP HLS4ML block**(NN computation on FPGA) and the **INFN Communication IP** (in which the APE Router is responsible for the inter-FPGA communication), we decide to design an HW toy-model to prove the correct behaviour of the firmware on our Xilinx Alveo U280.

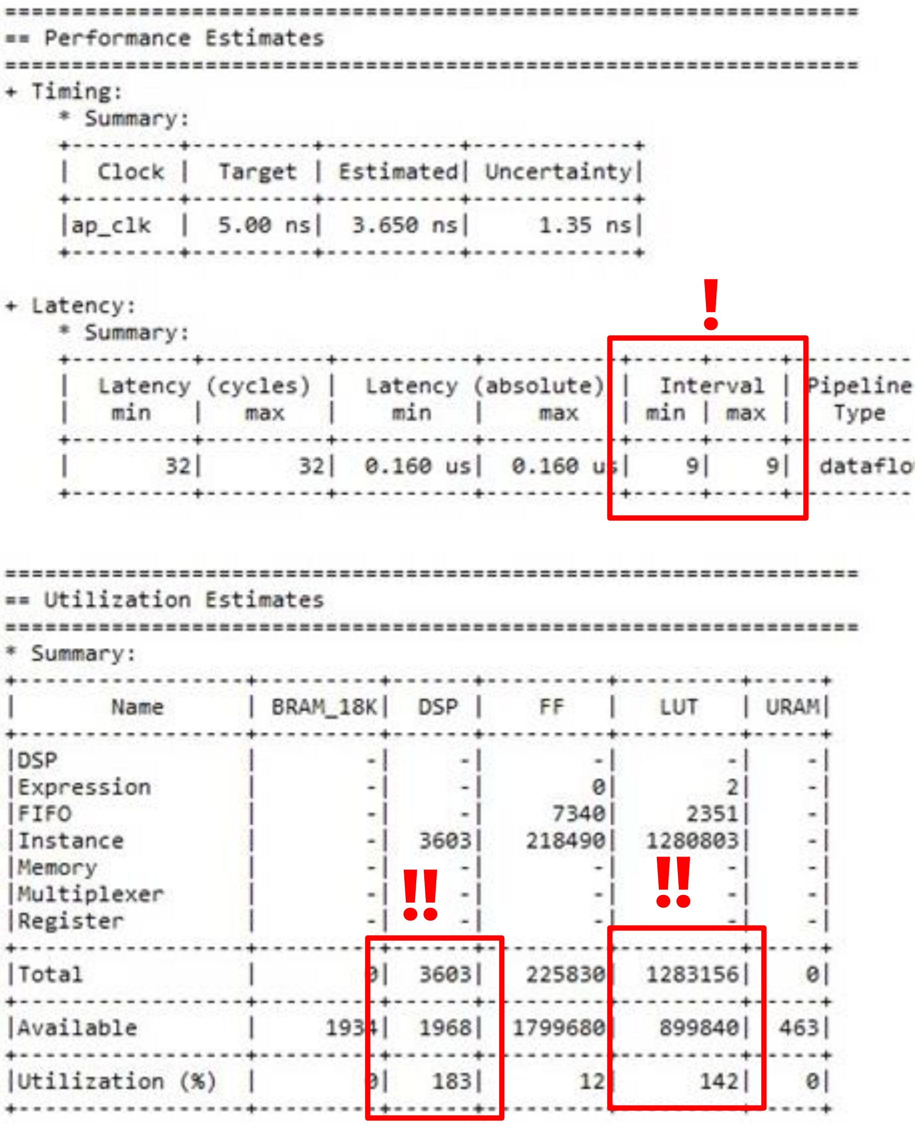
- ➔ data are loaded from the Host via the **krnl_load** HLS block, and streamed via 42 links (*hls::stream<ap_axis<16,0,0,0>>*) to the **preprocessing HLS** block, which prepares the input to feed the NN



dRICH Data Reduction: HLS4ML ⇒ HW Synthesis for TP MLP NN

⇒ We tried to synthetize the model of the TP MLP model at 200 MHz of operational clock on the AMD Versal Prime (the FPGA equipped on FLX-182 in APE-Lab)
⇒ it results to be impossible to reach an II to cope with the required throughput

⇒ out of resources!! 183% of DSP and 142% required!!
(even compiling the HLS4ML with “resource” flag to optimize occupation!!)

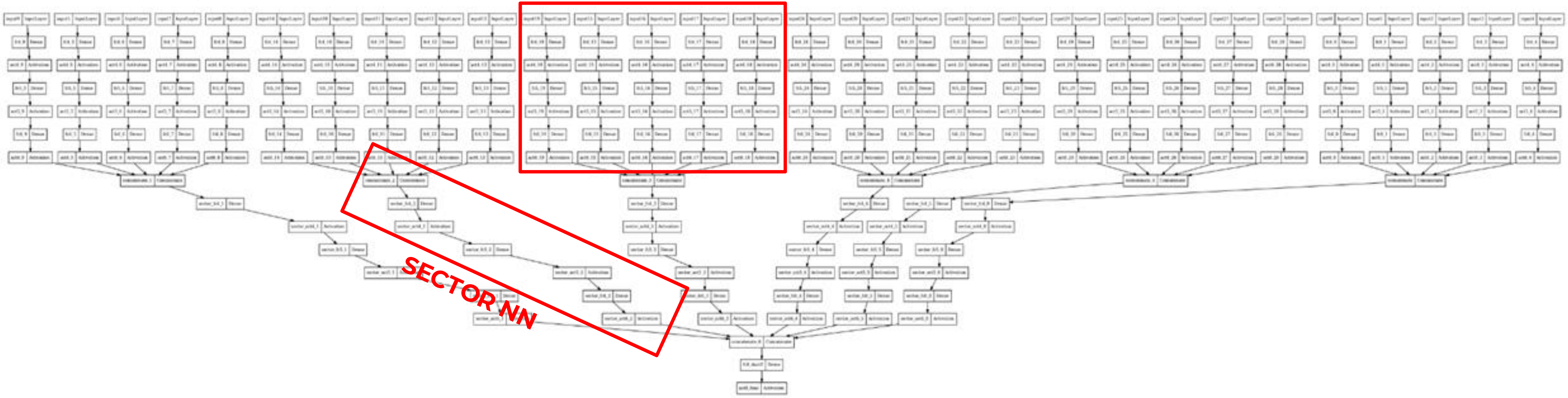


```
================================================================
== Performance Estimates
================================================================
+ Timing:
    * Summary:
    +--------+---------+----------+------------+
    |  Clock |  Target | Estimated| Uncertainty|
    +--------+---------+----------+------------+
    |ap_clk  |  5.00 ns|  3.650 ns|     1.35 ns|
    +--------+---------+----------+------------+

+ Latency:
    * Summary:
    +---------+---------+----------+----------+-----+-----+---------
    |  Latency (cycles) |  Latency (absolute) |  Interval | Pipeline
    |   min   |   max   |    min   |    max   | min | max |   Type
    +---------+---------+----------+----------+-----+-----+---------
    |       32|       32| 0.160 us| 0.160 us|    9|    9| dataflo
    +---------+---------+----------+----------+-----+-----+---------


================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+------+--------+--------+-----+
|       Name      | BRAM_18K|  DSP |   FF   |   LUT  | URAM|
+-----------------+---------+------+--------+--------+-----+
|DSP              |        -|     -|       -|       -|    -|
|Expression       |        -|     -|       0|       2|    -|
|FIFO             |        -|     -|    7340|    2351|    -|
|Instance         |        -|  3603|  218490| 1280803|    -|
|Memory           |        -|     -|       -|       -|    -|
|Multiplexer      |        -|     -|       -|       -|    -|
|Register         |        -|     -|       -|       -|    -|
+-----------------+---------+------+--------+--------+-----+
|Total            |        0|  3603|  225830| 1283156|    0|
+-----------------+---------+------+--------+--------+-----+
|Available        |     1934|  1968| 1799680|  899840|  463|
+-----------------+---------+------+--------+--------+-----+
|Utilization (%)  |        0|   183|      12|     142|    0|
+-----------------+---------+------+--------+--------+-----+
```

→ **HOW TO SOLVE NOW!??**

dRICH Data reduction:
New NN design ⇒ Sector MLP introduction

- The occupation problem occurred with the TP MLP NN is connected to the huge amount of computation required for its first layers (240x120). Thus, we tried to re-design our global distributed model by introducing 6 lighter intermediate model (called **Sector MLP NN**) each working on the aggregated information of a single sector.
- The 6 outputs are then aggregated and processed in a **lightweight TP NN** (single MLP layer, 5 neurons)



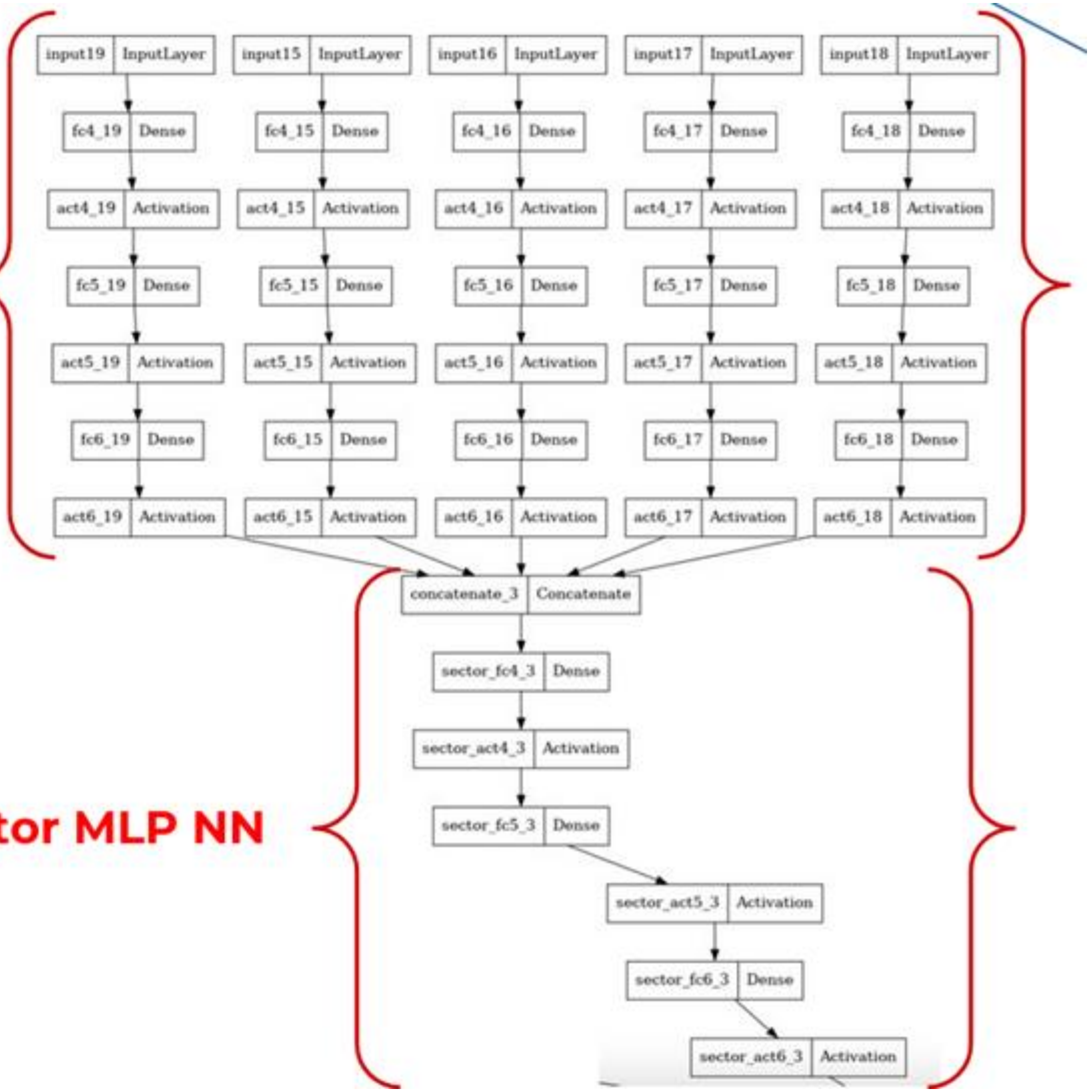
⇒ WHAT ABOUT PERFORMANCE?

dRICH Data reduction:
New NN design ⇒ Sector MLP introduction

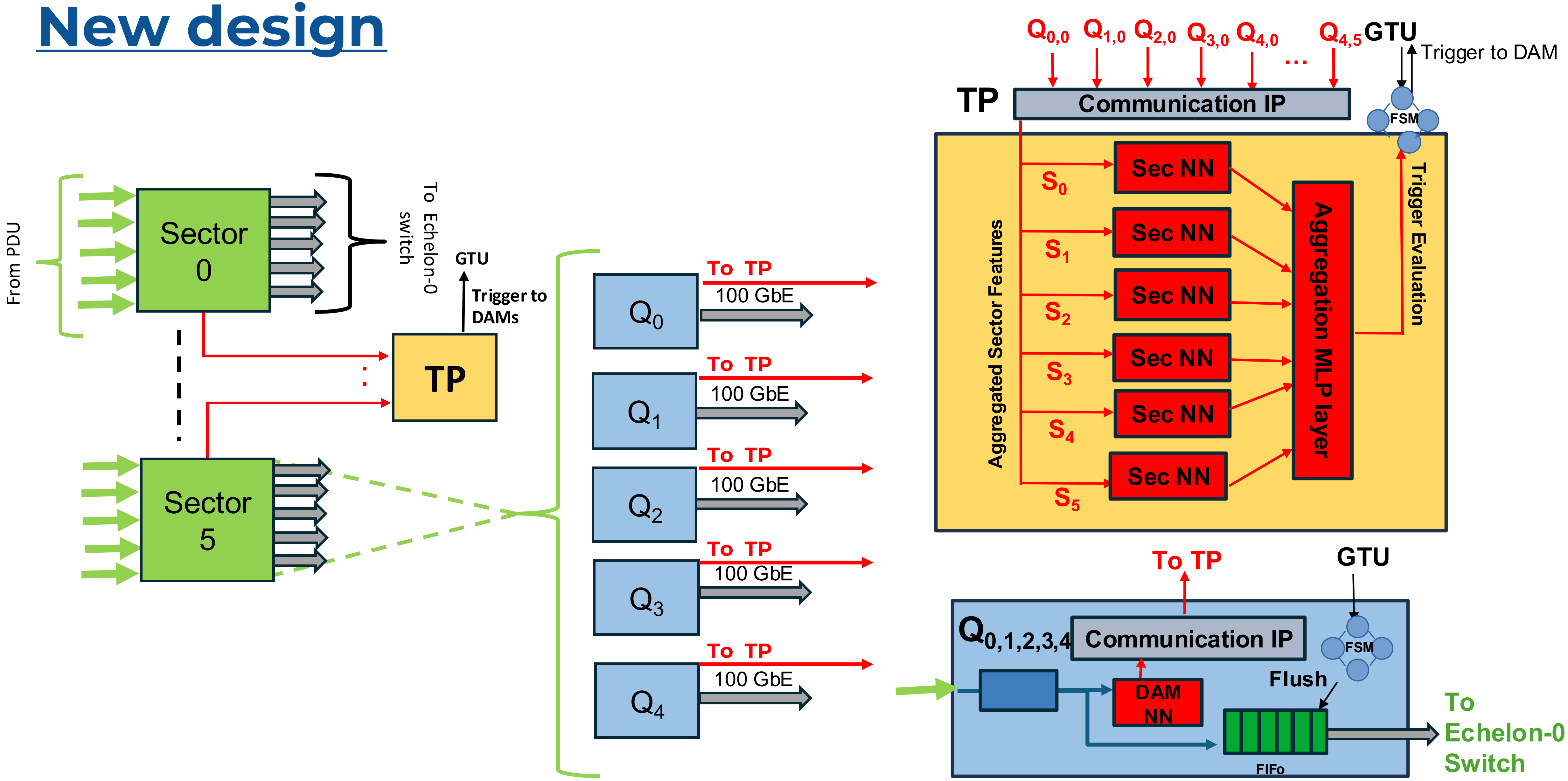
5 MLP DAM NNs (same sector)

For each sector, 5 MLP DAM output (**embedding**) are concatenated and then used to feed the Sector MLP model ⇒ **sector local information** extracted from the incoming data to perform the final prediction

Sector MLP NN

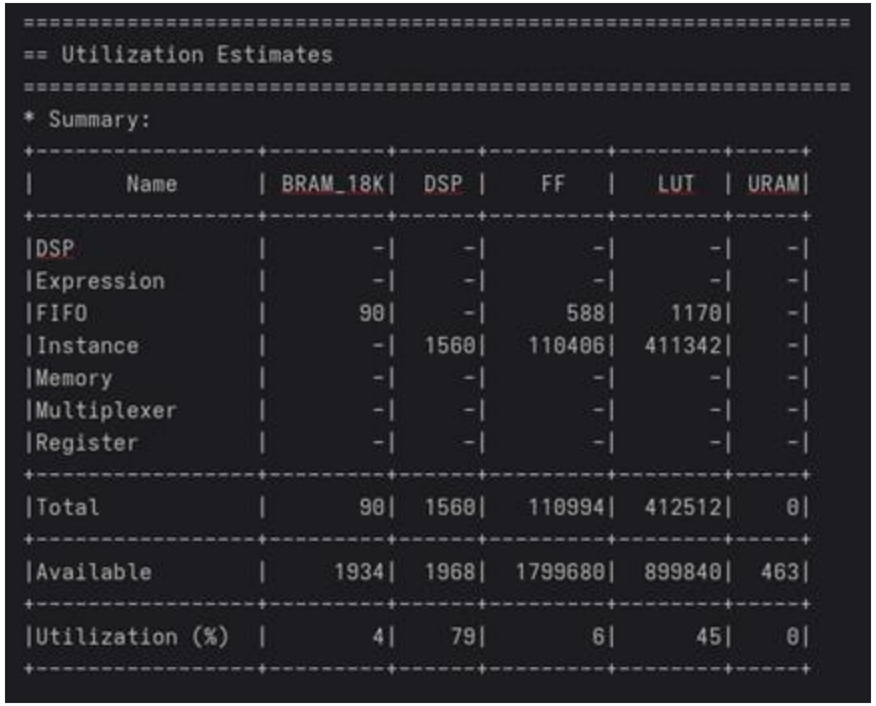


dRICH DAQ and Data Reduction: New design

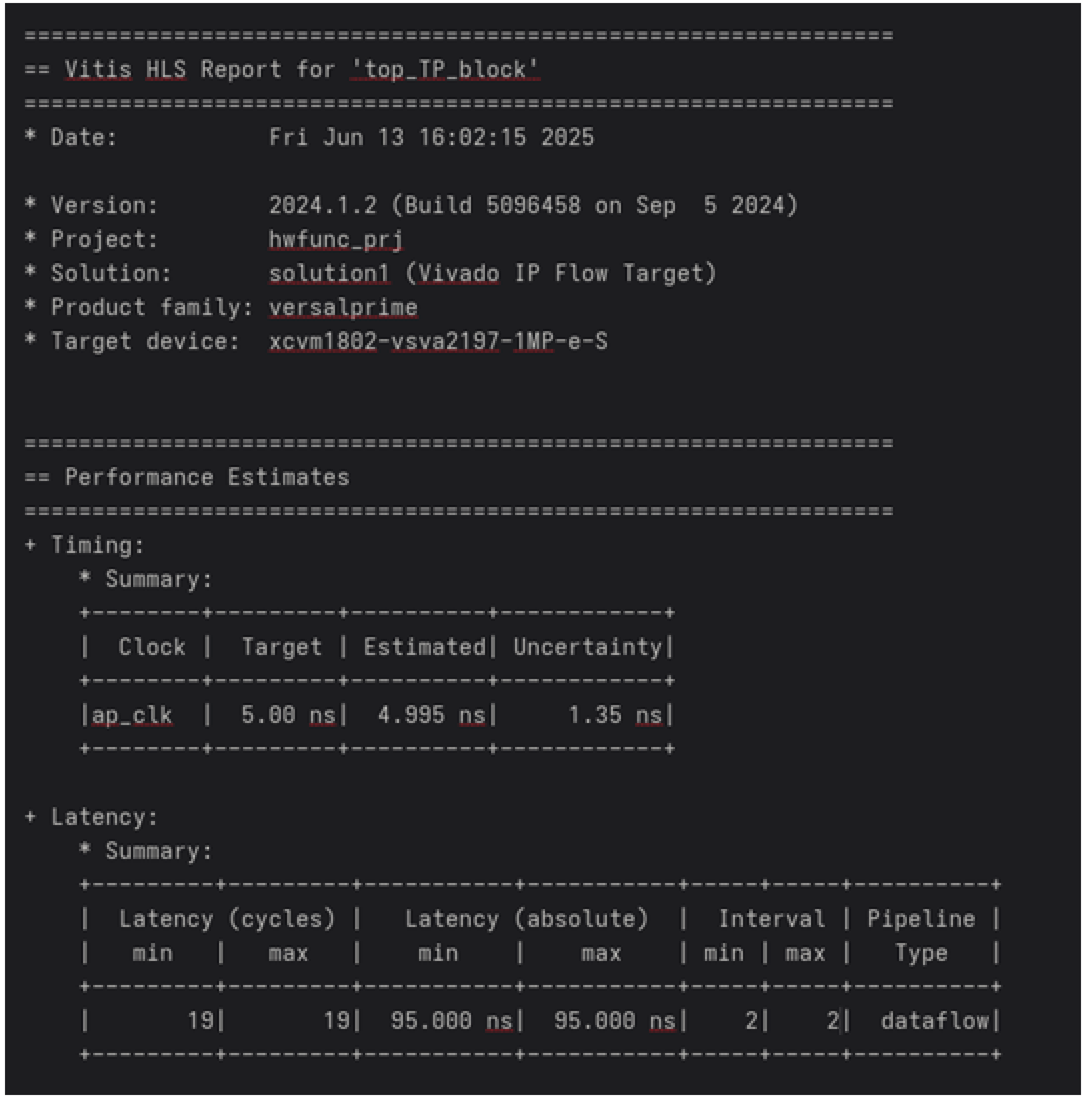


dRICH Data Reduction: HLS4ML □ HW Synthesis for TP NN

- The new **TP firmware** (composed by 5 Sector NN Hw blocks) have been correctly synthetized
  ⇒ **enough resources and II=2**



```
================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+------+---------+--------+-----+
|       Name      | BRAM_18K|  DSP |    FF   |   LUT  | URAM|
+-----------------+---------+------+---------+--------+-----+
|DSP              |        -|     -|        -|       -|    -|
|Expression       |        -|     -|        -|       -|    -|
|FIFO             |       90|     -|      588|    1170|    -|
|Instance         |        -|  1560|   110406|  411342|    -|
|Memory           |        -|     -|        -|       -|    -|
|Multiplexer      |        -|     -|        -|       -|    -|
|Register         |        -|     -|        -|       -|    -|
+-----------------+---------+------+---------+--------+-----+
|Total            |       90|  1560|   110994|  412512|    0|
+-----------------+---------+------+---------+--------+-----+
|Available        |     1934|  1968|  1799680|  899840|  463|
+-----------------+---------+------+---------+--------+-----+
|Utilization (%)  |        4|    79|        6|      45|    0|
+-----------------+---------+------+---------+--------+-----+
```

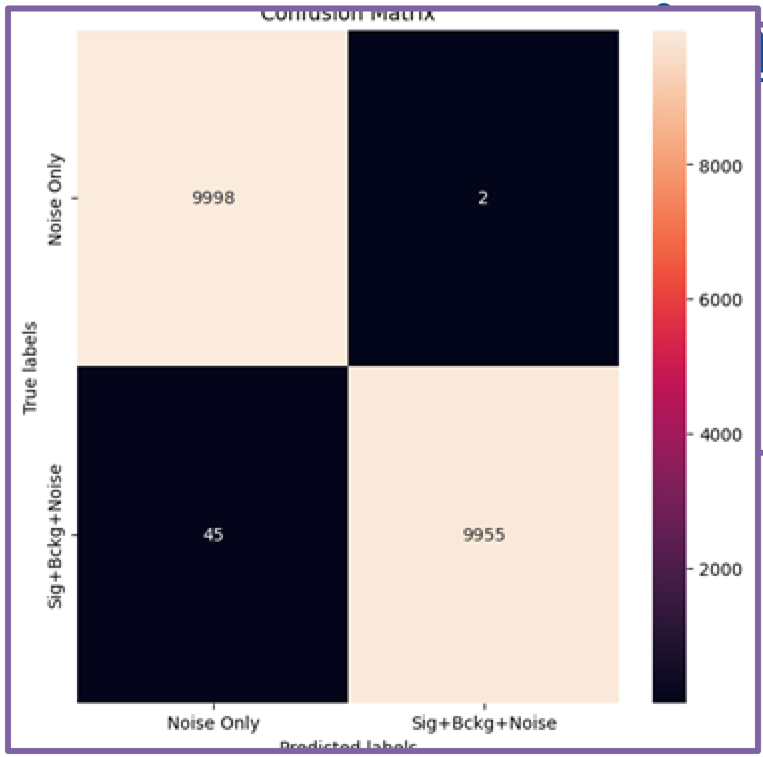


```
================================================================
== Vitis HLS Report for 'top_TP_block'
================================================================
* Date:           Fri Jun 13 16:02:15 2025

* Version:        2024.1.2 (Build 5096458 on Sep  5 2024)
* Project:        hwfunc_prj
* Solution:       solution1 (Vivado IP Flow Target)
* Product family: versalprime
* Target device:  xcvm1802-vsva2197-1MP-e-S


================================================================
== Performance Estimates
================================================================
+ Timing:
    * Summary:
    +--------+---------+----------+------------+
    |  Clock |  Target | Estimated| Uncertainty|
    +--------+---------+----------+------------+
    |ap_clk  |  5.00 ns|  4.995 ns|     1.35 ns|
    +--------+---------+----------+------------+

+ Latency:
    * Summary:
    +---------+---------+-----------+-----------+-----+-----+---------+
    |  Latency (cycles) |  Latency (absolute)   |  Interval | Pipeline |
    |   min   |   max   |    min    |    max    | min | max |   Type  |
    +---------+---------+-----------+-----------+-----+-----+---------+
    |       19|       19|  95.000 ns|  95.000 ns|    2|    2| dataflow|
    +---------+---------+-----------+-----------+-----+-----+---------+
```

Sec-MLP performance @ luminosity = 100fb-1

Keras model

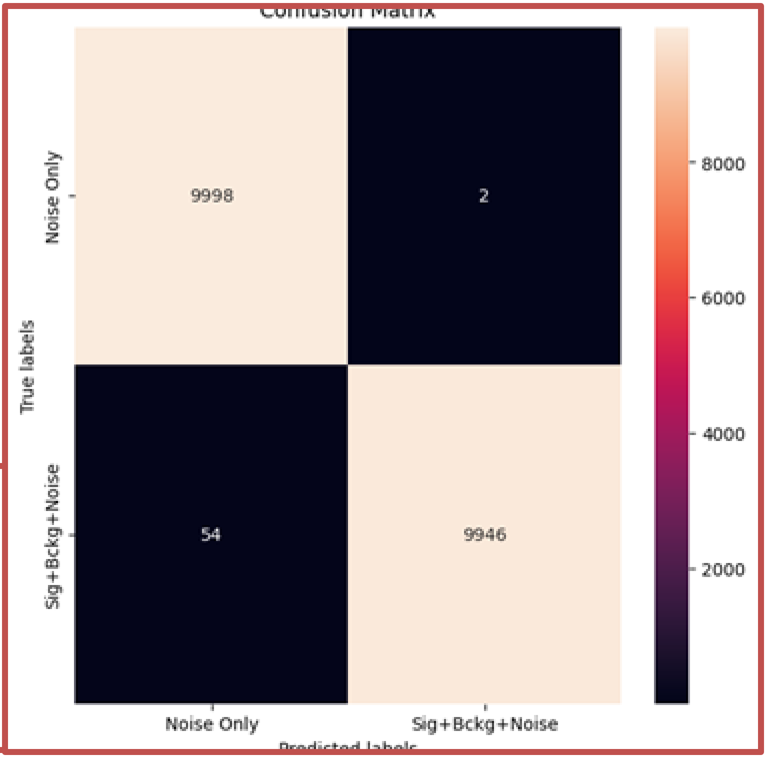
ewindow = 10ns



- ❑ Accuracy = (TP+TN)/(TP+TN+FP+FN)= 0.997
- ❑ Purity = TP/(TP+FP) = 0.995
- ❑ Recall = TP/(TP+FN) = 0.999

Model Quantization

- ● Inputs, Activations: fixed point<16,6>
- ● Weights, Biases: fixed point<8,1>

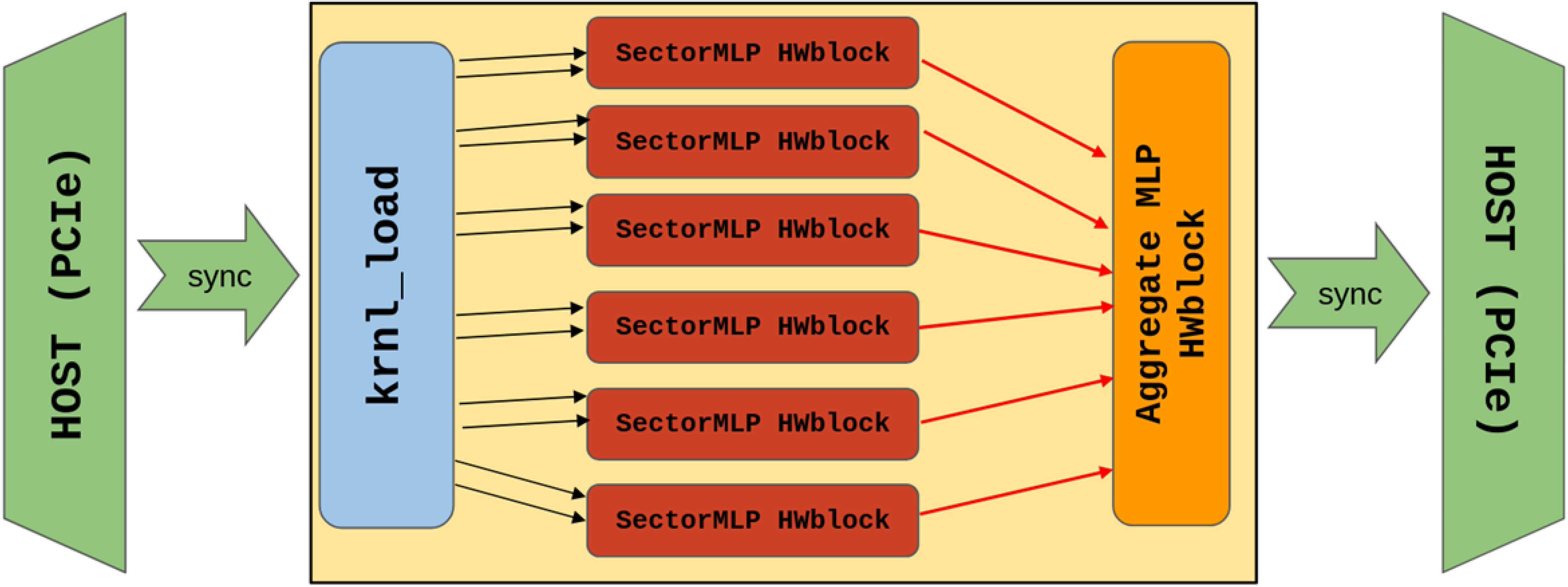


- ❑ Accuracy = (TP+TN)/(TP+TN+FP+FN) = 0.997
- ❑ Purity = TP/(TP+FP) = 0.994
- ❑ Recall = TP/(TP+FN) = 0.999

HLS4ML FPGA performance @ noiseRate= 200kHz

timewindow = 10ns

→ To validate the correct implementation of the **TP Sector MLP HLS4ML blocks** and evaluate system's performance, we decide to design an HW toy-model to prove the correct behaviour of the firmware on our Xilinx Alveo U280.



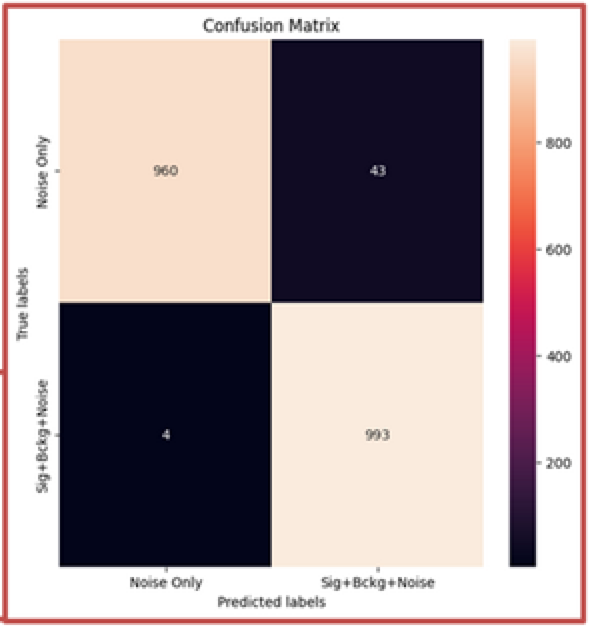
HLS4ML FPGA performance @ noiseRate= 200kHz

timewindow = 10ns

- □ Throughput (DDR) = 2.065 MHz
  ➔ instantiation interval II~97 cycles (@200 MHz)
- □ Throughput (BRAM) = 10.867 MHz
  ➔ instantiation interval II~19 cycles (@200 MHz)

Model Quantization

- ● Inputs, Activations: fixed point<16,6>
- ● Weights, Biases: fixed point<8,1>



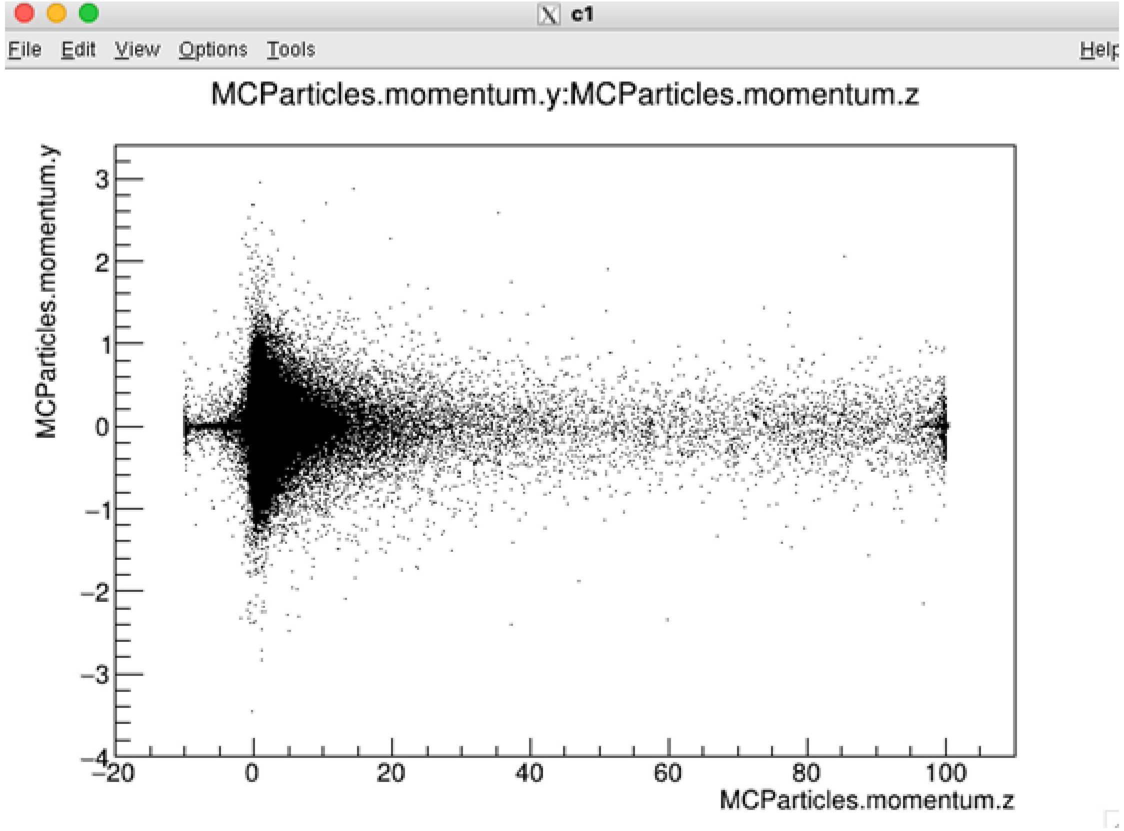
- □ Accuracy = (TP+TN)/(TP+TN+FP+FN) = 0.977
- □ Purity = TP/(TP+FP) = 0.996
- □ Recall = TP/(TP+FN) = 0.957

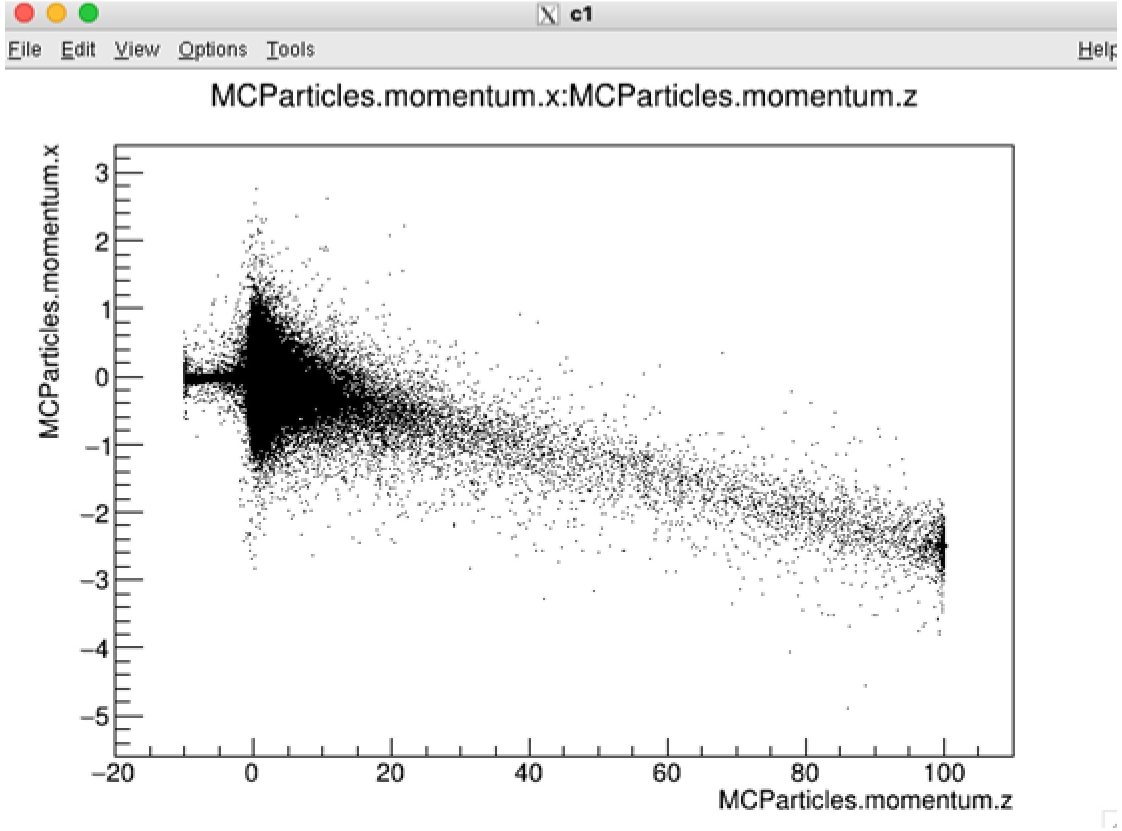
Conclusions

- Optimization of the performance in terms of **accuracy/purity/recall (ML parameters)** and **resources/throughput (HW implementation)** has been performed.
- The distributed MLP model has been tested on **different noise hits distribution** models (gaussian and radial-dependent) that have been included in the reconstruction pipeline.
- ⇒ **results are nearly optimal on simulated/reconstructed data**
- New design for the Sector TP NN model still under complete validation (new datasets training/testing, but first predictions seems accurate), but definitively convenient in terms of resources
- Deployment of the TP NN model on our testbed is ongoing
  ⇒ test for the interconnection with the DAM NN **(throughput issue to be solved)**

Backup Slides

Distribution of Events Particles Momenta





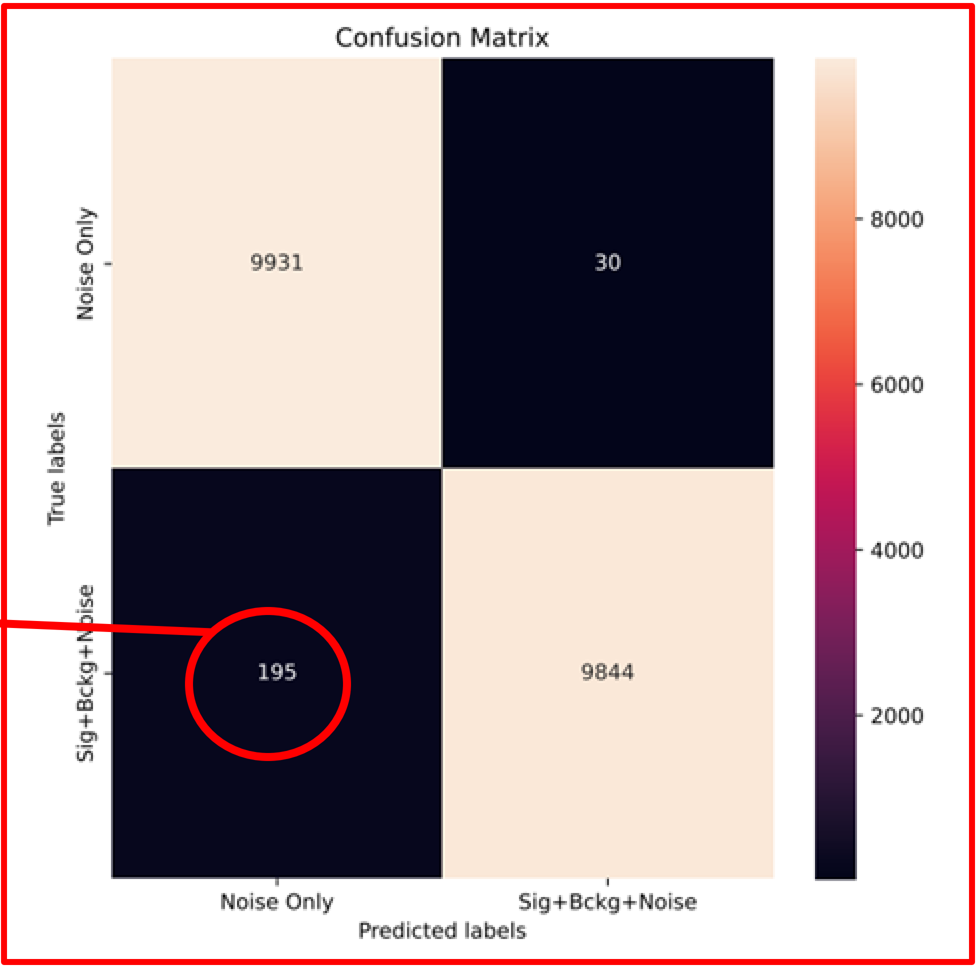
Close-up □ False Positive Events

Training and validating with datasets of 100 kHz dark count rates, we obtain a **99% accurate model**.

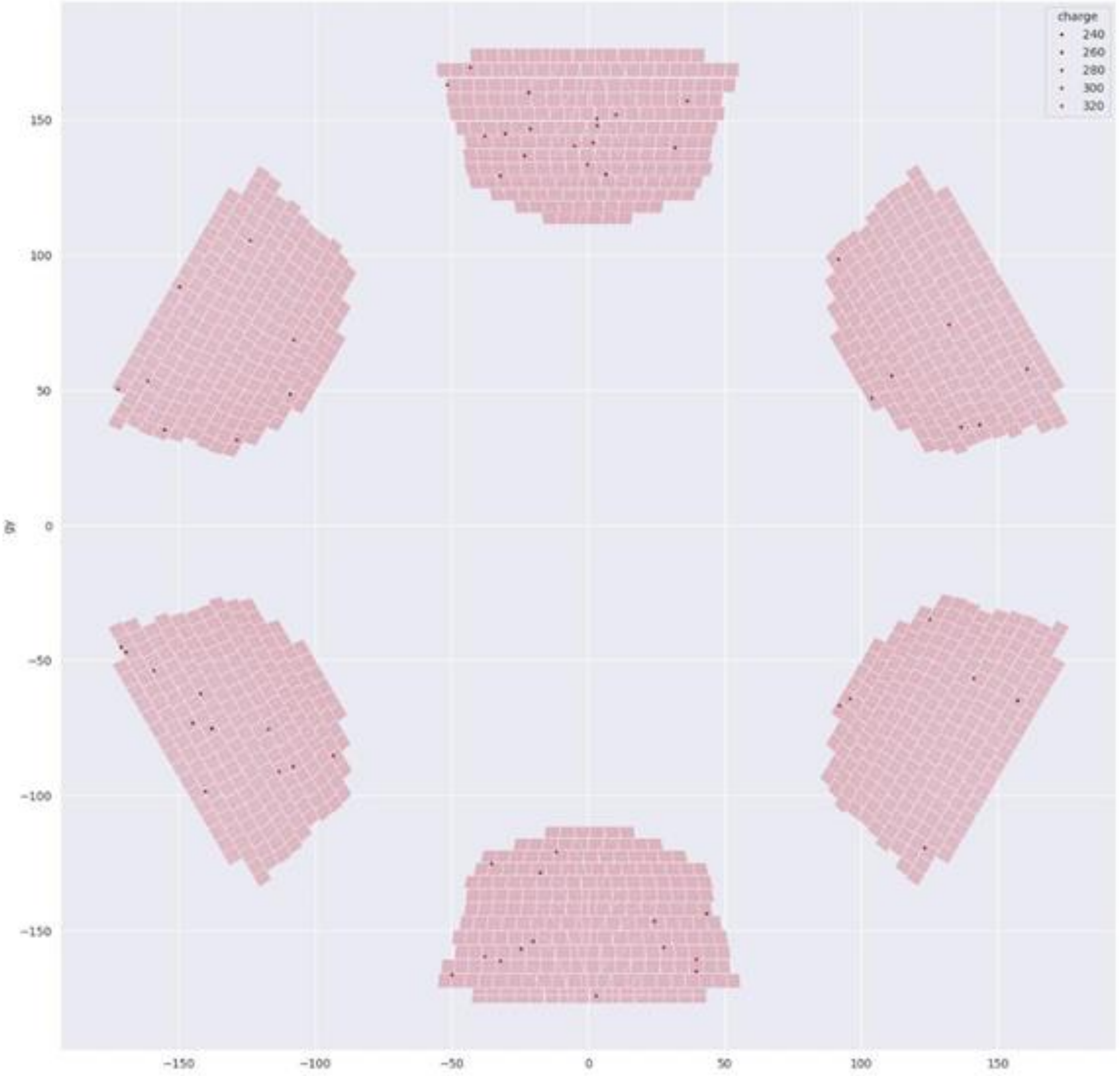
BUT WHAT ABOUT THESE FALSE POSITIVE EVENTS?

WHAT THEY LOOK LIKE?

ARE THEY **TRULY** SHOWING SIGNAL+BACKGROUND FEATURES?



Close-up □ False Positive Events



Example of a **False Positive event** (signal+background+noise, but **classified as noise**):

- Low number of dRICH hits
- **No Cherenkov rings detected**
- No evident dRICH hits clusters
- Homogenous dRICH hits distribution □ **comparable with a noise hits distribution**

Close-up □ False Positive ROOT TTree

```
MCParticles.PDG = 22, 11, 2212, 9900330, 2212, -311, 313, 2212, 11, 130, 311,
111, 310, 22, 22, 111, 111, 22, 22, 22
MCParticles.generatorStatus = 21, 21, 21, 21, 21, 2, 2, 1, 1, 1, 2, 2, 2, 1, 1,
2, 2, 1, 1, 1
[…]
MCParticles.time = 212.173264, 212.173264, 212.173264, 212.173264, 212.173264,
212.173264, 212.173264, 212.173264, 212.173264, 212.173264, 212.173264,
212.173264, 212.173264, 212.173264, 212.173264, 212.184769, 212.184769,
212.184769, 212.184769, 212.184769
[…]
MCParticles.momentum.x = 0.000092, -0.000105, -2.521645, 0.352699, -2.874251,
0.030792, 0.321907, -2.874251, -0.000105, 0.030793, 0.075927, 0.245985,
0.075927, 0.121866, 0.124117, 0.146451, -0.070528, 0.060760, 0.085690, -
0.060012
MCParticles.momentum.y = -0.000563, 0.000807, -0.012031, 0.239004, -0.251596, -
0.168178, 0.407180, -0.251596, 0.000807, -0.168186, 0.058748, 0.348438,
0.058748, 0.155661, 0.192775, 0.125229, -0.066484, 0.037693, 0.087535, 0.007046
MCParticles.momentum.z = -1.228703, -8.770502, 99.992050, -0.499420, 99.262772,
0.339107, -0.838527, 99.262772, -8.770502, 0.339123, -0.206120, -0.632420, -
0.206120, -0.410238, -0.222180, -0.279635, 0.073525, -0.013916, -0.265717,
0.087298
```

Close-up □ False Positive ROOT TTree

```
                   γ
MCParticles.PDG = 22, 11, 2212, 9900330, 2212, -311, 313, 2212, 11, 130, 311,
111, 310, 22, 22, 111, 111, 22, 22, 22
MCParticles.generatorStatus = 21, 21, 21, 21, 21, 2, 2, 1, 1, 1, 2, 2, 2, 1, 1,
2, 2, 1, 1, 1
[…]
MCParticles.time = 212.173264, 212.173264, 212.173264, 212.173264, 212.173264,
212.173264, 212.173264, 212.173264, 212.173264, 212.173264, 212.173264,
212.173264, 212.173264, 212.173264, 212.173264, 212.184769, 212.184769,
212.184769, 212.184769, 212.184769
[…]
MCParticles.momentum.x = 0.000092, -0.000105, -2.521645, 0.352699, -2.874251,
0.030792, 0.321907, -2.874251, -0.000105, 0.030793, 0.075927, 0.245985,
0.075927, 0.121866, 0.124117, 0.146451, -0.070528, 0.060760, 0.085690, -
0.060012
MCParticles.momentum.y = -0.000563, 0.000807, -0.012031, 0.239004, -0.251596, -
0.168178, 0.407180, -0.251596, 0.000807, -0.168186, 0.058748, 0.348438,
0.058748, 0.155661, 0.192775, 0.125229, -0.066484, 0.037693, 0.087535, 0.007046
MCParticles.momentum.z = -1.228703, -8.770502, 99.992050, -0.499420, 99.262772,
0.339107, -0.838527, 99.262772, -8.770502, 0.339123, -0.206120, -0.632420, -
0.206120, -0.410238, -0.222180, -0.279635, 0.073525, -0.013916, -0.265717,
0.087298
```

Close-up False Positive ROOT TTree

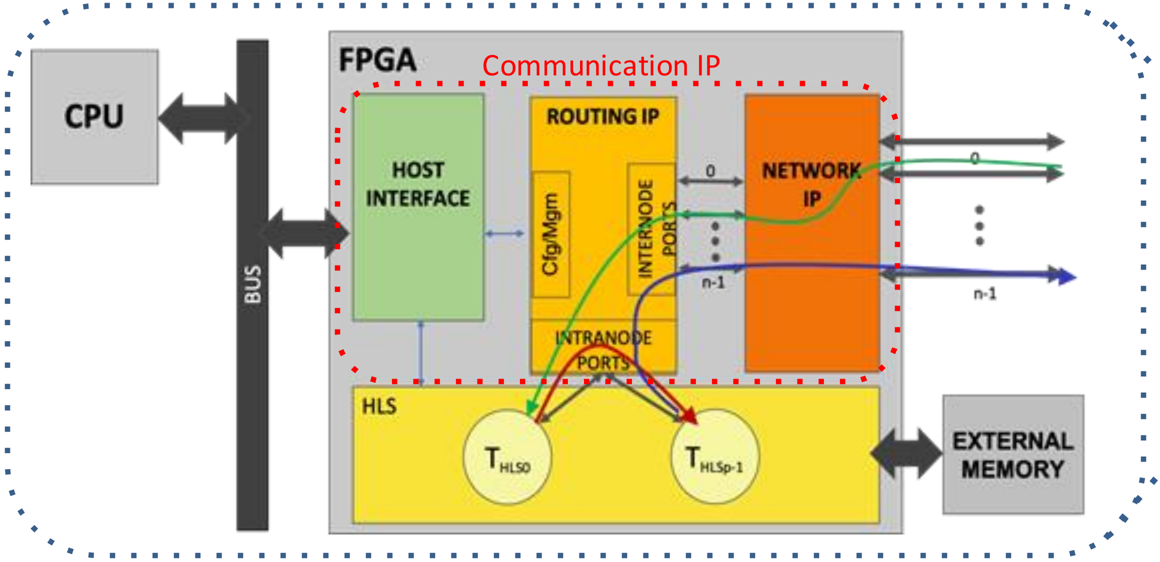
e^{-}

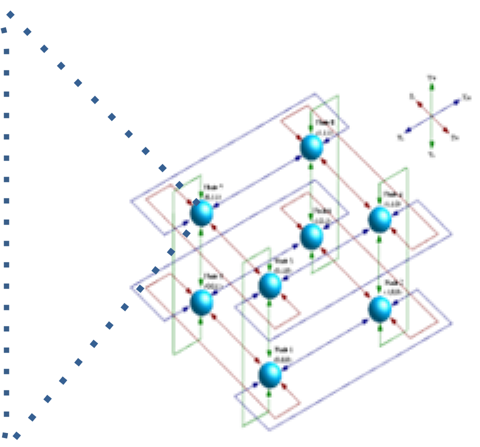
```
MCParticles.PDG = 22, 11, 2212, 9900330, 2212, -311, 313, 2212, 11, 130, 311,
111, 310, 22, 22, 111, 111, 22, 22, 22
MCParticles.generatorStatus = 21, 21, 21, 21, 21, 2, 2, 1, 1, 1, 2, 2, 2, 1, 1,
2, 2, 1, 1, 1
[…]
MCParticles.time = 212.173264, 212.173264, 212.173264, 212.173264, 212.173264,
212.173264, 212.173264, 212.173264, 212.173264, 212.173264, 212.173264,
212.173264, 212.173264, 212.173264, 212.173264, 212.184769, 212.184769,
212.184769, 212.184769, 212.184769
[…]
MCParticles.momentum.x = 0.000092, -0.000105, -2.521645, 0.352699, -2.874251,
0.030792, 0.321907, -2.874251, -0.000105, 0.030793, 0.075927, 0.245985,
0.075927, 0.121866, 0.124117, 0.146451, -0.070528, 0.060760, 0.085690, -
0.060012
MCParticles.momentum.y = -0.000563, 0.000807, -0.012031, 0.239004, -0.251596, -
0.168178, 0.407180, -0.251596, 0.000807, -0.168186, 0.058748, 0.348438,
0.058748, 0.155661, 0.192775, 0.125229, -0.066484, 0.037693, 0.087535, 0.007046
MCParticles.momentum.z = -1.228703, -8.770502, 99.992050, -0.499420, 99.262772,
0.339107, -0.838527, 99.262772, -8.770502, 0.339123, -0.206120, -0.632420, -
0.206120, -0.410238, -0.222180, -0.279635, 0.073525, -0.013916, -0.265717,
0.087298
```

Close-up □ False Positive ROOT TTree

```
                          p
MCParticles.PDG = 22, 11, 2212, 9900330, 2212, -311, 313, 2212, 11, 130, 311,
111, 310, 22, 22, 111, 111, 22, 22, 22
MCParticles.generatorStatus = 21, 21, 21, 21, 21, 2, 2, 1, 1, 1, 2, 2, 2, 1, 1,
2, 2, 1, 1, 1
[…]
MCParticles.time = 212.173264, 212.173264, 212.173264, 212.173264, 212.173264,
212.173264, 212.173264, 212.173264, 212.173264, 212.173264, 212.173264,
212.173264, 212.173264, 212.173264, 212.173264, 212.184769, 212.184769,
212.184769, 212.184769, 212.184769
[…]
MCParticles.momentum.x = 0.000092, -0.000105, -2.521645, 0.352699, -2.874251,
0.030792, 0.321907, -2.874251, -0.000105, 0.030793, 0.075927, 0.245985,
0.075927, 0.121866, 0.124117, 0.146451, -0.070528, 0.060760, 0.085690, -
0.060012
MCParticles.momentum.y = -0.000563, 0.000807, -0.012031, 0.239004, -0.251596, -
0.168178, 0.407180, -0.251596, 0.000807, -0.168186, 0.058748, 0.348438,
0.058748, 0.155661, 0.192775, 0.125229, -0.066484, 0.037693, 0.087535, 0.007046
MCParticles.momentum.z = -1.228703, -8.770502, 99.992050, -0.499420, 99.262772,
0.339107, -0.838527, 99.262772, -8.770502, 0.339123, -0.206120, -0.632420, -
0.206120, -0.410238, -0.222180, -0.279635, 0.073525, -0.013916, -0.265717,
0.087298
```

APEIRON: the Node

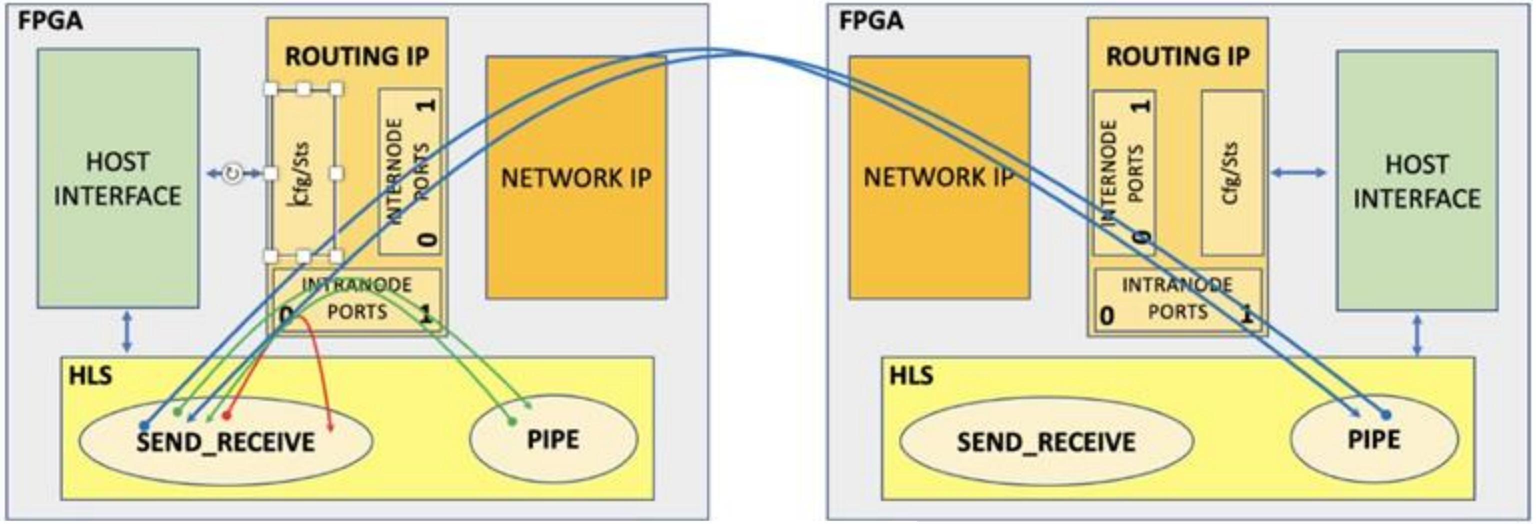




APEIRON node in a 3D Torus network topology

- **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

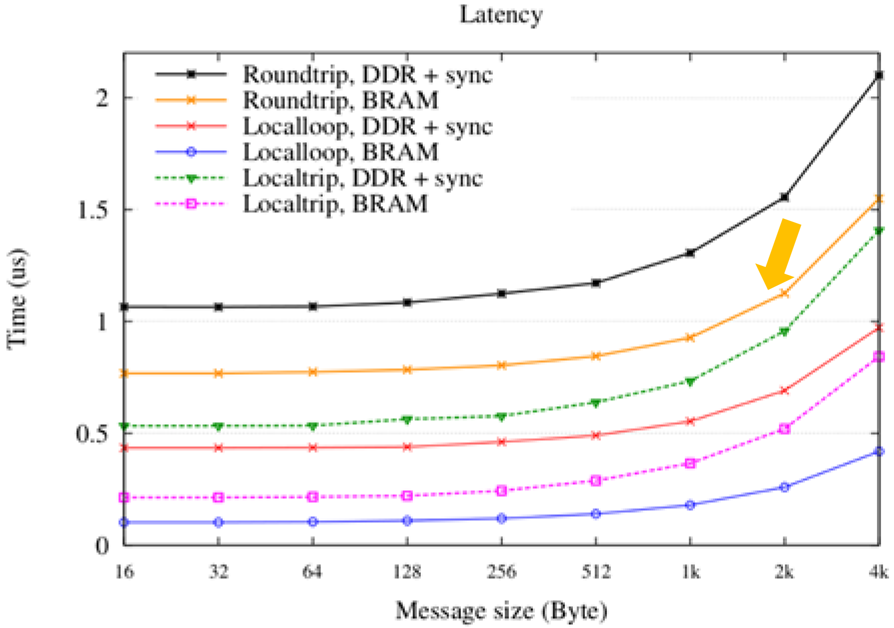


Test modes

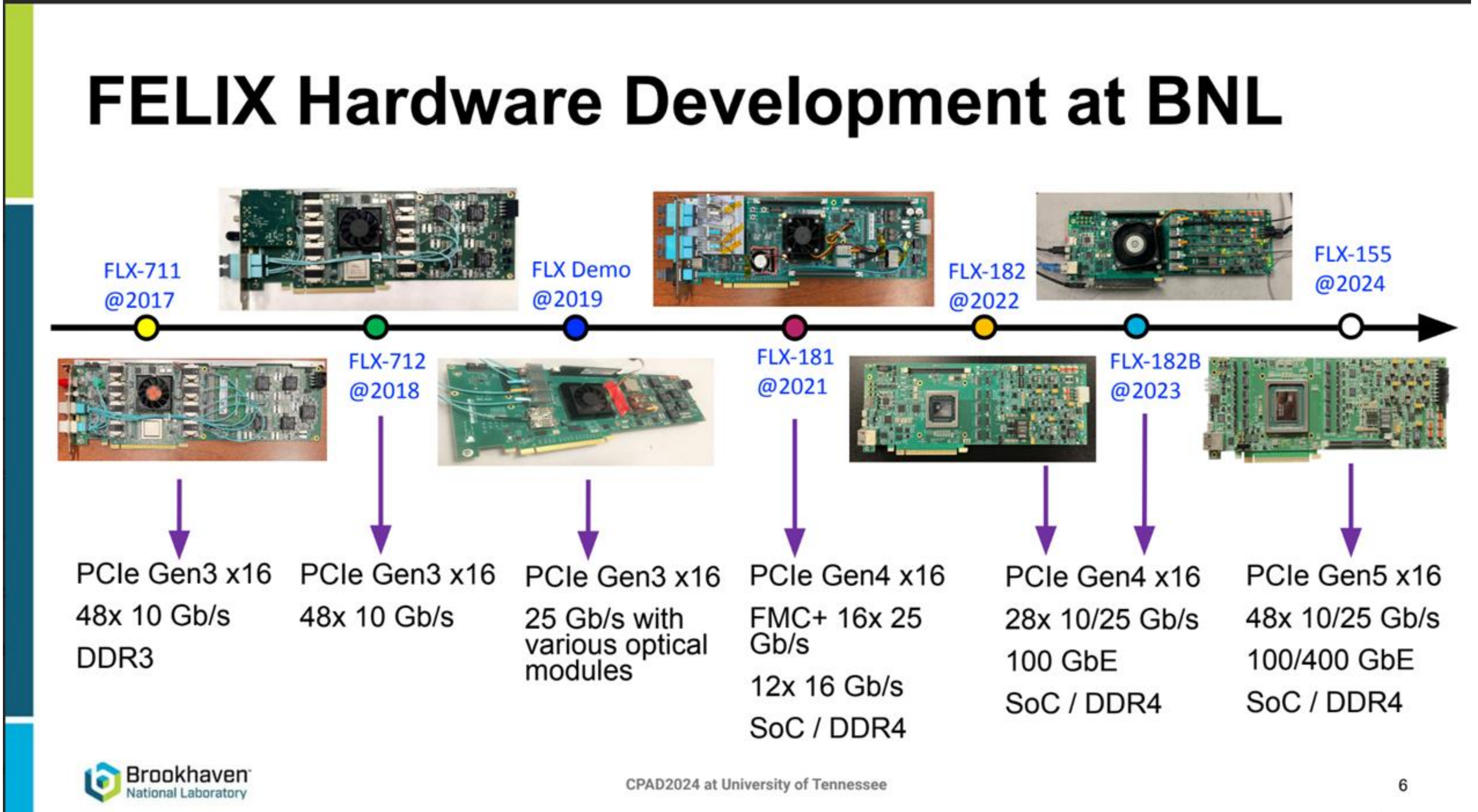
- Local-loop (red arrow)
- Local-trip (green arrows)
- Round-trip (blue arrows)

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)



FELIX Hardware Development at BNL

FLX-711 @2017

PCIe Gen3 x16

48x 10 Gb/s

DDR3

FLX-712 @2018

PCIe Gen3 x16

48x 10 Gb/s

FLX Demo @2019

PCIe Gen3 x16

25 Gb/s with various optical modules

FLX-181 @2021

PCIe Gen4 x16

FMC+ 16x 25 Gb/s

12x 16 Gb/s

SoC / DDR4

FLX-182 @2022

FLX-182B @2023

PCIe Gen4 x16

28x 10/25 Gb/s

100 GbE

SoC / DDR4

FLX-155 @2024

PCIe Gen5 x16

48x 10/25 Gb/s

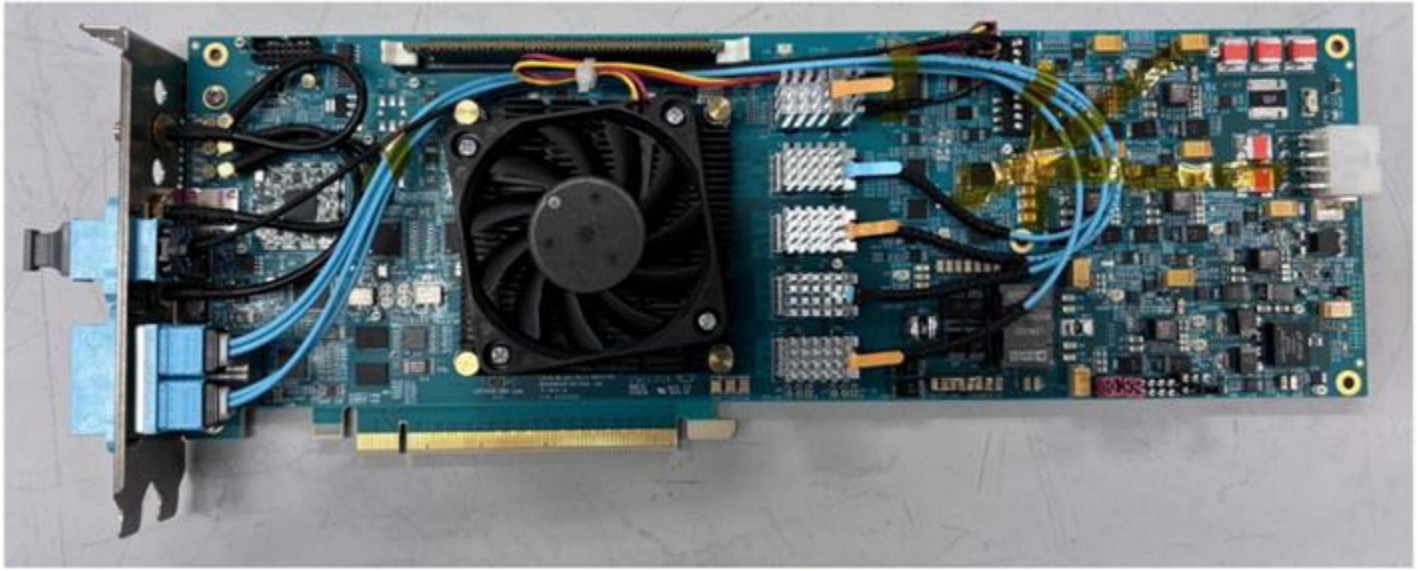
100/400 GbE

SoC / DDR4

Brookhaven National Laboratory

CPAD2024 at University of Tennessee

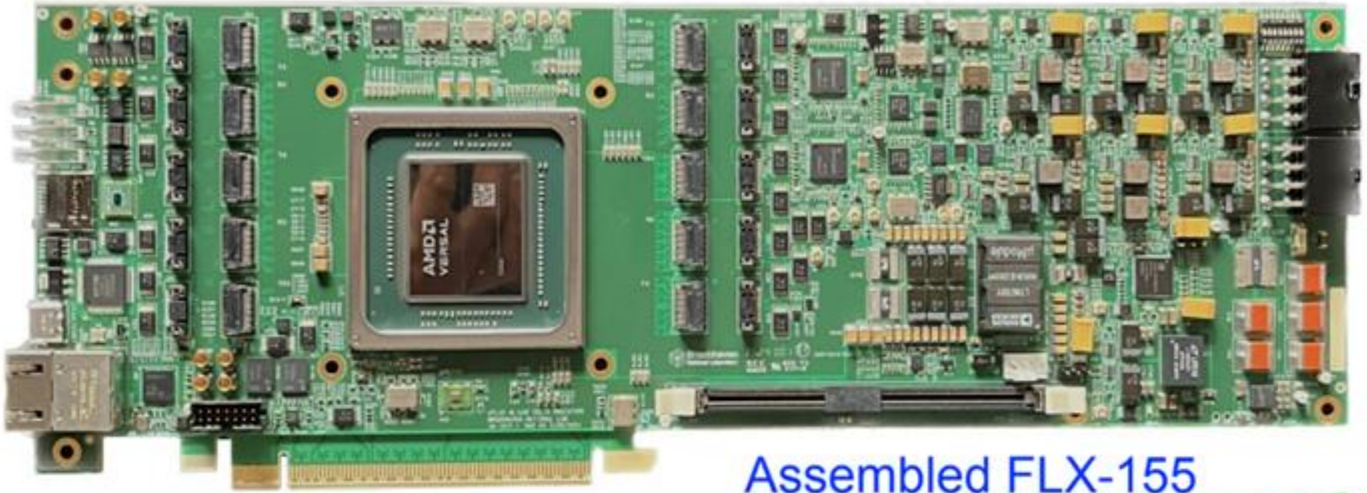
FLX-182B Hardware



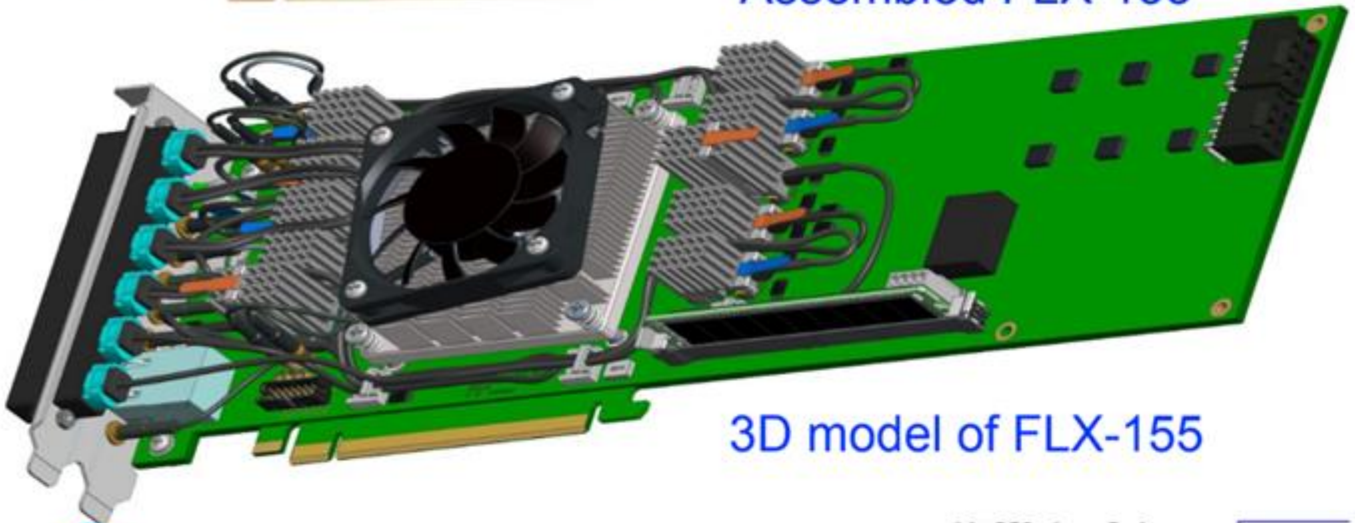
Assembled FLX-182B

- FPGA: Xilinx Versal Prime XCVM1802
- PCIe Gen4 x16, 256 GT/s
- 24 FireFly links with 3 possible configurations
  - 24 links up to 25 Gb/s
  - 24 links up to 10 Gb/s (CERN-B FireFly)
  - 12 links up to 25 Gb/s + 12 links up to 10 Gb/s
- 4 FireFly links with 2 possible configurations with 14 or 25 Gb/s FireFly TRx
  - LTI interface
  - 100 GbE
- Built-in self test, online configuration and monitoring
- White Rabbit
- DDR4 Mini-UDIMM
- GbE/SD3.0/PetaLinux

FLX-155 Hardware

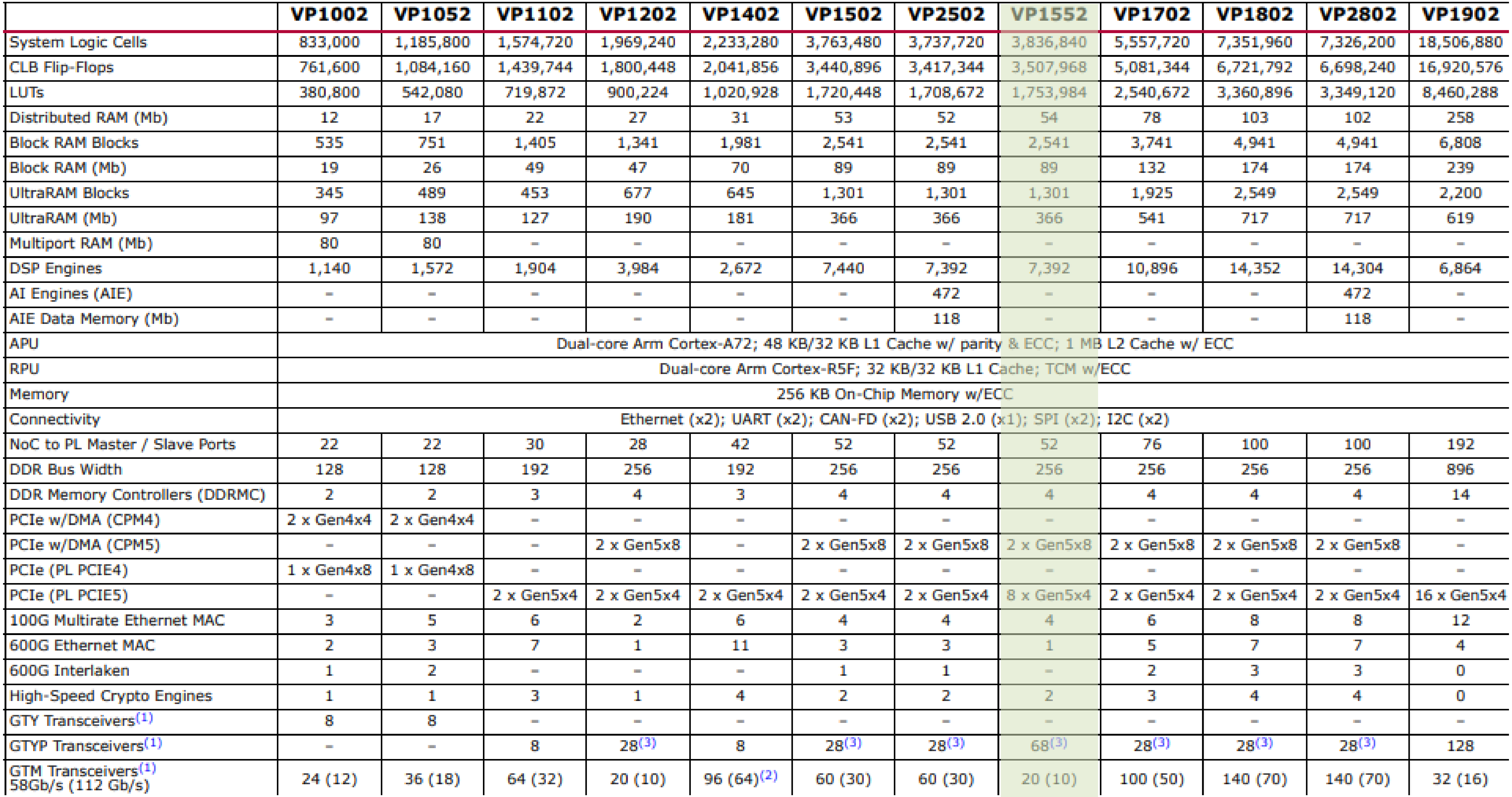


Assembled FLX-155



3D model of FLX-155

- AMD/Xilinx Versal Premium FPGA: XCVP1552-2MSEVSVA3340
- 2 x PCIe Gen5 x8 512 GT/s
- 56 FireFly optical links
  - Compatible with various options
  - Default configuration for ATLAS
    - 48 data links up to 25 Gb/s
    - 4 links for LTI
  - Optional 4 links for 100 GbE
- Electrical IOs
- Built-in self test, online configuration and monitoring
- 1 16GB DDR4 Mini-UDIMM
- USB-JTAG/USB-UART
- GbE/SD3.0/PetaLinux
- Optional White Rabbit



	VP1002	VP1052	VP1102	VP1202	VP1402	VP1502	VP2502	VP1552	VP1702	VP1802	VP2802	VP1902
System Logic Cells	833,000	1,185,800	1,574,720	1,969,240	2,233,280	3,763,480	3,737,720	3,836,840	5,557,720	7,351,960	7,326,200	18,506,880
CLB Flip-Flops	761,600	1,084,160	1,439,744	1,800,448	2,041,856	3,440,896	3,417,344	3,507,968	5,081,344	6,721,792	6,698,240	16,920,576
LUTs	380,800	542,080	719,872	900,224	1,020,928	1,720,448	1,708,672	1,753,984	2,540,672	3,360,896	3,349,120	8,460,288
Distributed RAM (Mb)	12	17	22	27	31	53	52	54	78	103	102	258
Block RAM Blocks	535	751	1,405	1,341	1,981	2,541	2,541	2,541	3,741	4,941	4,941	6,808
Block RAM (Mb)	19	26	49	47	70	89	89	89	132	174	174	239
UltraRAM Blocks	345	489	453	677	645	1,301	1,301	1,301	1,925	2,549	2,549	2,200
UltraRAM (Mb)	97	138	127	190	181	366	366	366	541	717	717	619
Multiport RAM (Mb)	80	80	-	-	-	-	-	-	-	-	-	-
DSP Engines	1,140	1,572	1,904	3,984	2,672	7,440	7,392	7,392	10,896	14,352	14,304	6,864
AI Engines (AIE)	-	-	-	-	-	-	472	-	-	-	472	-
AIE Data Memory (Mb)	-	-	-	-	-	-	118	-	-	-	118	-
APU	Dual-core Arm Cortex-A72; 48 KB/32 KB L1 Cache w/ parity & ECC; 1 MB L2 Cache w/ ECC											
RPU	Dual-core Arm Cortex-R5F; 32 KB/32 KB L1 Cache; TCM w/ECC											
Memory	256 KB On-Chip Memory w/ECC											
Connectivity	Ethernet (x2); UART (x2); CAN-FD (x2); USB 2.0 (x1); SPI (x2); I2C (x2)											
NoC to PL Master / Slave Ports	22	22	30	28	42	52	52	52	76	100	100	192
DDR Bus Width	128	128	192	256	192	256	256	256	256	256	256	896
DDR Memory Controllers (DDRMC)	2	2	3	4	3	4	4	4	4	4	4	14
PCIe w/DMA (CPM4)	2 x Gen4x4	2 x Gen4x4	-	-	-	-	-	-	-	-	-	-
PCIe w/DMA (CPM5)	-	-	-	2 x Gen5x8	-	2 x Gen5x8	2 x Gen5x8	2 x Gen5x8	2 x Gen5x8	2 x Gen5x8	2 x Gen5x8	-
PCIe (PL PCIE4)	1 x Gen4x8	1 x Gen4x8	-	-	-	-	-	-	-	-	-	-
PCIe (PL PCIE5)	-	-	2 x Gen5x4	2 x Gen5x4	2 x Gen5x4	2 x Gen5x4	2 x Gen5x4	8 x Gen5x4	2 x Gen5x4	2 x Gen5x4	2 x Gen5x4	16 x Gen5x4
100G Multirate Ethernet MAC	3	5	6	2	6	4	4	4	6	8	8	12
600G Ethernet MAC	2	3	7	1	11	3	3	1	5	7	7	4
600G Interlaken	1	2	-	-	-	1	1	-	2	3	3	0
High-Speed Crypto Engines	1	1	3	1	4	2	2	2	3	4	4	0
GTY Transceivers[1]	8	8	-	-	-	-	-	-	-	-	-	-
GTYP Transceivers[1]	-	-	8	28[3]	8	28[3]	28[3]	68[3]	28[3]	28[3]	28[3]	128
GTM Transceivers[1] 58Gb/s (112 Gb/s)	24 (12)	36 (18)	64 (32)	20 (10)	96 (64)[2]	60 (30)	60 (30)	20 (10)	100 (50)	140 (70)	140 (70)	32 (16)