



1

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

Cristian Rossi INFN Roma, APE Lab for the ePIC Roma1/2 team

> Giornate Nazionali ePIC Italia Padova, June 17<sup>th</sup> 2025

## dRICH: Data Reduction

### **Online Signal/ Noise discrimination using ML**



#### SiPM Noise:

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

#### ML task:

•

2

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

## **dRICH: Dataset for training, classes**

#### Phys Signal+Phys Background+Noise

3





## **dRICH Data Reduction Stage on FPGA**

4

 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 <u>dRICH Interaction Tagger</u> to boost performance

## **dRICH Data Reduction on FPGA - Deployment**



5



## **dRICH Data Reduction on FPGA - Deployment**



6





## **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**



#### 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 <u>noise rates</u> and <u>noise models</u> 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
  - o sigma = 0.1\*avg

9

noiseTimeWindow = 10 ns (no shutter)





## <sup>10</sup> 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)



### <sup>11</sup> 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



«Distributed MLP Model»

#### 12 **Distributed MLP Tensorflow Model**



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



## dRICH Data reduction: model training & validation

- → We trained the 30 MLP DAM models concatenated to the single MLP TP model by using <u>100k Signal+Background+Noise</u> and <u>100k Noise Only</u> <u>event</u>
- → 200k balanced dataset (90% training set, 10% validation set) for any of the considered noise hits distribution models, varying their typical parameters:
  - Gaussian model:
    - noiseRate = **40 kHz**, timeWindow = 10ns;
    - noiseRate = **80 kHz**, timeWindow = 10ns;
    - noiseRate = 100 kHz, timeWindow = 10ns;
    - noiseRate = **300 kHz**, timeWindow = 10ns;

### Radial-dependent model:

- luminosity = **2 fb-1**, timeWindow = 10ns;
- luminosity = **20 fb-1**, timeWindow = 10ns;
- luminosity = **100 fb-1**, timeWindow = 10ns;

# <sup>14</sup> <u>G Model performance @ noiseRate = 40 KHz</u>

Keras model

## <u>@timewindow = 10ns</u>

Noise Only

Predicted labels

Sia+Bcka+Noise

#### Confusion Matrix - 10000 Accuracy = (TP+TN)/(TP+TN+FP+FN) = **Model Quantization** Noise Only 8000 0.999 **Inputs, Activations:** 9961 0 **Purity = TP/(TP+FP) = 0.999** fixed point<16,6> Recall = TP/(TP+FN) = 1.000 6000 Weights, Biases: Irue labels fixed point<8,1> 4000 **Confusion Matrix** Sig+Bckg+Noise - 10000 10037 - 2000 loise Only 8000 9961 Noise Only Sia+Bcka+Noise 6000 Predicted labels True labels 4000 Accuracy = (TP+TN)/(TP+TN+FP+FN) = 10033 0.999 2000 **Purity = TP/(TP+FP) = 0.999**

Recall = TP/(TP+FN) = 1.000

# <sup>15</sup> <u>G Model performance @ noiseRate = 80 KHz</u>

Keras model

## <u>@timewindow = 10ns</u>



Recall = TP/(TP+FN) = 1.000



### 16 <u>**R Model performance @ luminosity = 100kHz</u></u></u>**

### <u>@timewindow = 10ns</u>

2000

Sig+Bckg+Noise

Noise Only

Predicted labels

#### **Confusion Matrix** - 10000 Accuracy = **Model Quantization** (TP+TN)/(TP+TN+FP+FN) = **Inputs, Activations:** Noise Only - 8000 0.999 0 9961 fixed point<16,6> **Purity = TP/(TP+FP) = 0.999** Weights, Biases: 6000 Recall = TP/(TP+FN) = 1.000 True labels fixed point<8,1> **Confusion Matrix** 4000 - 10000 Sig+Bckg+Noise 10037 2000 Noise Only 8000 9961 6000 Sig+Bckg+Noise Noise Only True labels Predicted labels 4000 Sig+Bckg+Noise Accuracy = (TP+TN)/(TP+TN+FP+FN) = 5 10034

0.999

Keras model

- **Purity = TP/(TP+FP) = 0.999**
- Recall = TP/(TP+FN) = 1.000

# <sup>17</sup> <u>G Model performance @ noiseRate = 300 KHz</u>

Keras model

### <u>@timewindow = 10ns</u>

Noise Only

Predicted labels

Sig+Bckg+Noise

#### **Confusion Matrix** - 10000 Accuracy = **Model Quantization** (TP+TN)/(TP+TN+FP+FN) = **Inputs, Activations:** Noise Only - 8000 0.999 fixed point<16,6> 9961 0 **Purity = TP/(TP+FP) = 0.999** Weights, Biases: **Recall = TP/(TP+FN) = 1.000** 6000 fixed point<8,1> **Irue** labels Confusion Matrix 4000 - 10000 Sig+Bckg+Noise 10036 2000 Noise Only 8000 9961 0 6000 Noise Only Sig+Bckg+Noise True labels Predicted labels 4000 Sig+Bckg+Noise Accuracy = (TP+TN)/(TP+TN+FP+FN) = 10036 0.999 2000 **Purity = TP/(TP+FP) = 0.999**

Recall = TP/(TP+FN) = 1.000

# <sup>18</sup> Gaussian Model performance: summary



## **R Model performance @ luminosity = 2fb-1**

### <u>@timewindow = 10ns</u>

Sig+Bckg+Noise

Predicted labels

Noise Only

#### **Confusion Matrix** - 10000 Accuracy = **Model Quantization** (TP+TN)/(TP+TN+FP+FN) = **Inputs, Activations:** - 8000 Noise Only 0.999 fixed point<16,6> 9994 **Purity = TP/(TP+FP) = 0.999** Weights, Biases: Recall = TP/(TP+FN) = 1.000 6000 fixed point<8,1> True labels Confusion Matrix 4000 10000 Sig+Bckg+Noise 10003 Voise Only 8000 2000 9993 6000 Sig+Bckg+Noise Noise Only **True labels** Predicted labels 4000 sig+Bckg+Noise Accuracy = (TP+TN)/(TP+TN+FP+FN) = 10002 0.999 2000 **Purity = TP/(TP+FP) = 0.999**

Recall = TP/(TP+FN) = 0.999

19

Keras model

# <sup>20</sup> <u>R Model performance @ luminosity = 20fb-1</u>

#### **Keras model**

## <u>@timewindow = 10ns</u>

Sig+Bckg+Noise

Noise Only

Predicted labels



Recall = TP/(TP+FN) = 1.000

#### <sup>21</sup> <u>R Model performance @ luminosity = 100fb-1</u>

Keras model

## timewindow = 10ns

Predicted labels



#### <sup>22</sup> <u>Radial-dependent Model performance: summary</u>



### <sup>23</sup> <u>dRICH Data Reduction:</u> <u>HLS4ML ⇒ HW Synthesis for DAM MLP NN</u>

⇒ 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 <u>after synthesis</u> 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 2020.1 (Build 2897737 on Wed May 27 20:21:37 MDT 2020) Version: Project: hwfunc\_prj Solution: solution1 Product family: virtexuplus Target device: xcu280-fsvh2892-2L-e Performance Estimates Timing: \* Summarv: Clock | Target | Estimated | Uncertainty |ap\_clk | 5.00 ns | 3.641 ns | 1.35 ns Latency: \* Summarv: Latency (cycles) Latency (absolute) Interval | Pipeline min max max mín 14 70.000 ns | 70.000 ns 21 21 dataflow

#### <sup>24</sup> 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 <u>APE Router</u> is responsible for the inter-FPGA communication), we decide to design an <u>HW toy-model</u> 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



#### <sup>25</sup> dRICH Data Reduction: HW Implementation (U280)

[[rossi@apequad02 apeiron_framework]\$ ./host -b test_u280_200MHz.xclbin	-n 100000
Device name: xilinx_u280_gen3x16_xdma_base_1	
Device bdf: 0000:17:00.1	
Device max freq: 450	
Device m2m: 0	
Device nodma: 0	
Device kdma: 0	
Bitstream loading	
Bitstream loaded	
Switch kernel	
Resetting switch	
CHANNEL UP: 0	
Running receiver kernel:	
Starting sr kernel	
Thread started ==> Press a key to print registers	
FINISH	
Time elapsed: 6012614.000000 n Throughput: 16.631701 MHz	
	II=12 ??
X0000000000000000000000000000000000000	
0000	

- → a throughput issue occurred while executing the firmware on the U280 testbed
- → despite the information present in the Vitis\_HLS report (where a II=2 is indicated), we found an erroneous II=12 studying the hw\_emu waveforms
- → still under investigation

### <sup>26</sup> <u>dRICH Data Reduction:</u> <u>HLS4ML ⇒ HW Synthesis for TP MLP NN</u>

⇒ 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!!)

### → HOW TO SOLVE NOW!??

- Timing:					
* Summary:		+		+	
Clock   T	arget   Esti	mated  L	Incertainty	/	
ap_clk   5	.00 ns  3.6	50 ns	1.35 ns	5   • +	
Latency: ★ Summary:			-	!	
Latency (c   min	ycles)   La max	tency (a min	max	Interva   min   ma	1   Pipeli x   Type
32	32 0.	160 us	0.160 us	9	9 data
= Utilization Es	timates				
== Utilization Es	timates			·····	+ +
== Utilization Es * Summary: Name	timates 	DSP	FF	LUT	+ URAM
== Utilization Es Summary: Name DSP	timates BRAM_18K	DSP   -	FF   -	LUT   -1	URAM  
= Utilization Es Summary: Name DSP Expression	timates   BRAM_18K    -    -	DSP   -  -	FF   	LUT   -  2  2251	URAM  
== Utilization Es Summary: Name DSP Expression FIFO Instance	timates   BRAM_18K    -    -    -	DSP   -  -  -  3603	FF   -  0 7340 218490	LUT   -  2  2351  1280803	URAM  + -  -  -  -  -
= Utilization Es Summary: Name DSP Expression FIFO Instance Memory	timates BRAM_18K BRAM_18K 	DSP   -  -  -  3603  -	FF   -  0  7340  218490  -	LUT   -  2  2351  1280803  -	URAM  + -  -  -  -  -  -
== Utilization Es Summary: Name DSP Expression FIFO Instance Memory Multiplexer	timates BRAM_18K  	DSP   -  -  -  3603  -  -	FF   -  0 7340 218490 -  -	LUT   -  21 2351 1280803 -  -	URAM  -  -  -  -  -  -  -
= Utilization Es Summary: Name DSP Expression FIFO Instance Memory Multiplexer Register	timates   BRAM_18K    -    -    -    -    -	DSP   -  -  3603 -  -  -	FF   -  0  7340  218490  -  -  -	LUT   -  2  2351 1280803 -  -  -	URAM  -  -  -  -  -  -  -  -
= Utilization Es Summary: Name DSP Expression FIFO Instance Memory Multiplexer Register Total	timates BRAM_18K   -    -    -    -    -    -	DSP   -  -  -  3603 -  -  -  3603	FF   0 7340 218490 - - - 225830	LUT   2 2351 1280803 - - - - 1283156	URAM - I - I - I - I - I - I - I - I - I - I
== Utilization Es Summary: Name DSP Expression FIFO Instance Memory Multiplexer Register Total Available	timates   BRAM_18K    -  	DSP   -  -  -  3603  -  -  -  3603  1968	FF   7340 218490 - - 225830 1799680	LUT -  2351 1280803 -  - 1283156 899840	URAM -  -  -  -  -  -  -  -  463
= Utilization Es Summary: Name DSP Expression FIFO Instance Memory Multiplexer Register Total Available Utilization (%)	timates   BRAM_18K    -  	DSP   -	FF   0 7340 218490 - - 225830 1799680 12	LUT -  2 2351 1280803 -  -  1283156 899840 899840	URAM -  -  -  -  -  -  -  463

# <sup>27</sup> dRICH Data reduction:

## <u>New NN design ⇒ Sector MLP introduction</u>

- The occupation problem occurred with the TP MLP NN is connected to the <u>huge amount of</u> <u>computation required</u> 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?

# <sup>28</sup> dRICH DAQ and Data Reduction:

From PDU



### <sup>29</sup> <u>dRICH Data Reduction:</u> <u>HLS4ML || HW Synthesis for TP NN</u>

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

ates 	DSP   	FF   	LUT	=====+ + URAM  +
======= BRAM_18K   -  -	+- DSP   +-	FF   	LUT	====== + URAM  +
BRAM_18K  	DSP   	FF   	LUT I	+ URAM  +
BRAM_18K  	DSP   	FF   	LUT I	URAM  +
BRAM_18K   +- -   -	DSP   +- -	FF   	LUT	URAM  +
  - <u> -</u>	-+   	+- -1	+	+
-  <u>- </u>	- !	-		
-			-1	-1
	-1	-	-1	-1
90	-1	588	1170	-1
-	1560	110406	411342	-1
-1	-	-1	-	-
-1	-1	-	-1	-1
-	-1	-1	-1	-1
+-	+-	+-	+	+
90	1560	110994	412512	0
1934	1968	1799680	899840	463
4	+- 79	6	45	+ 0
		-  1560  -  -  -  -  -  -  90  1560  	-  1560  110406  -  -  -  -  -  -  -  -  -  90  1560  110994  	-  1560  110406  411342  -  -  -  -  -  -  -  -  -  -  -  -  90  1560  110994  412512  1934  1968  1799680  899840  4  79  6  45



### <sup>30</sup> Sec-MLP performance @ luminosity = 100fb-1

#### Keras model

## <u>timewindow = 10ns</u>

Noise Only

Sig+Bckg+Noise



- Purity = TP/(TP+FP) = 0.994
- Recall = TP/(TP+FN) = 0.999

# <sup>31</sup> Conclusions

- Optimization of the performance in terms of accuracy/purity/recall (ML parameters) and resources/throughput (HW implementation) has been performed.
- o 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.
- $\circ \Rightarrow$  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 <u>convenient in terms of resources</u>
- o 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**

# <sup>33</sup> Distribution of Events Particles Momenta



## **Close-up Give 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

### **<u>Close-up</u> Give 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, 21, 2, 2, 1, 1, 1, 2, 2, 2, 1, 1, 2, 2, 1, 1, 1 [...] **MCParticles.time** = 212.173264, 212.173266, 212.1732664, 212.173266, 212.173266, 212.173266, 212.1732 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

### **<u>Close-up</u> False Positive ROOT Tree**

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, 212.184769, 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

### **<u>Close-up</u> False Positive ROOT Tree**

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, 212.184769,

#### [...]

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 F**

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, 212.184769, 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**



- 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-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**



- AMD/Xilinx Versal Premium FPGA: XCVP1552-2MSEVSVA3340
- 2 x PCle Gen5 x8 512 GT/s
- 56 FireFly optical links
  - Compatible with various options 0
  - Default configuration for ATLAS 0
    - 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**



CPAD2024 at University of Tennessee

Δ	5
T	J

2)	1004000	1004050	1004400	1004 202	1004 400	VELEDO	VERTON	VDAEED	1004 700	100000	VIDAGOA	1004000
	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)	1.4	-	-	-	-	-	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					2	56 KB On-Chip	Memory w/E	C				
Connectivity		80	10	Ethernet	(x2); UART (x	2); 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		-	-	-	1	-	-	-	-	-
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		-	-	-	1.2	-	-	-	-	-
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	(C-2)	2	3	3	0
High-Speed Crypto Engines	1	1	3	1	4	2	2	2	3	4	4	0
GTY Transceivers <sup>(1)</sup>	8	8	-	-	-	-		-	-	-	-	-
GTYP Transceivers <sup>(1)</sup>	-	-	8	28(3)	8	28(3)	28(3)	68(3)	28(3)	28(3)	28(3)	128
GTM Transceivers <sup>(1)</sup> 58Gb/s (112 Gb/s)	24 (12)	36 (18)	64 (32)	20 (10)	96 (64) <sup>(2)</sup>	60 (30)	60 (30)	20 (10)	100 (50)	140 (70)	140 (70)	32 (16)