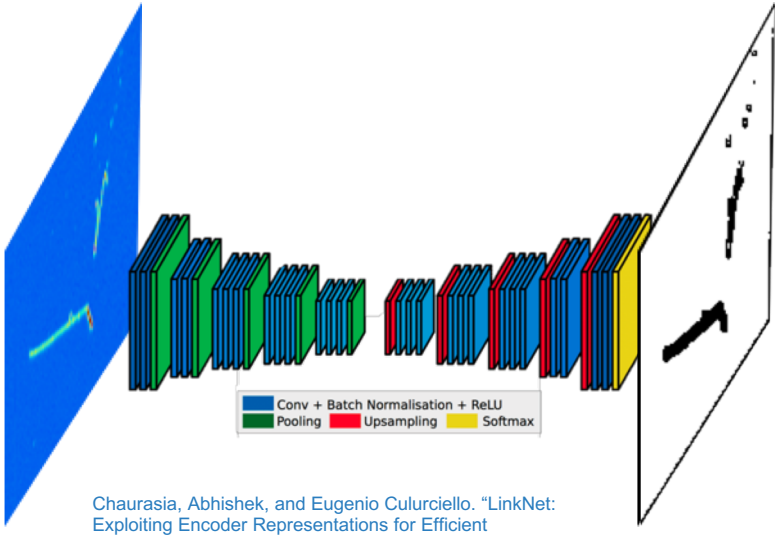
Micron FPGA test on protoDUNE-SP

Manuel J. Rodriguez, Saul Alonso

OUR PLANS

Data selection and trigger generation

- Focus on identifying areas of interest where there is activity on the detector.
- Fully Convolutional Networks to do image segmentation (**UNets**).
- **Input**: raw signals.
- **Goal**: checking the raw signals to get information from the waveforms.
    - **Locate where there are hits!**

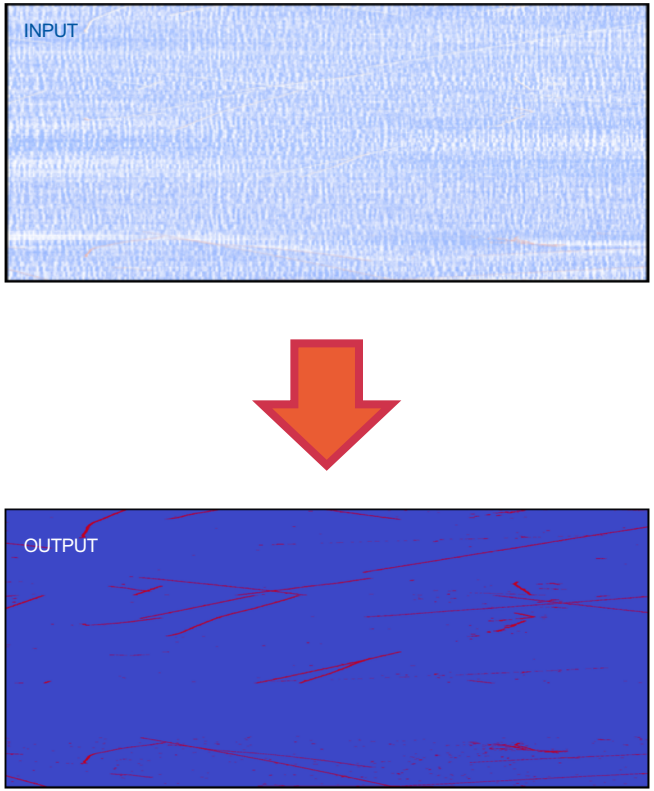


Chaurasia, Abhishek, and Eugenio Culurciello. "LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation." 2017 IEEE Visual Communications and Image Processing (VCIP) (2017): n. pag. Crossref. Web. arXiv:1707.03718

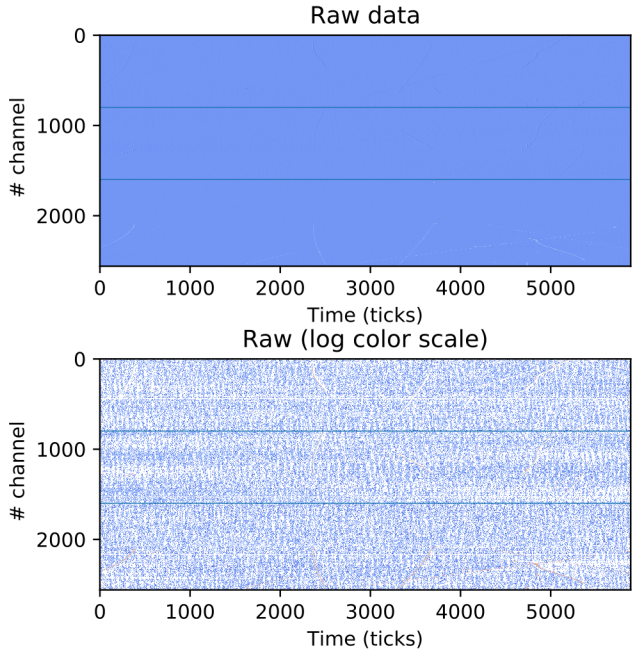
OUR PLANS

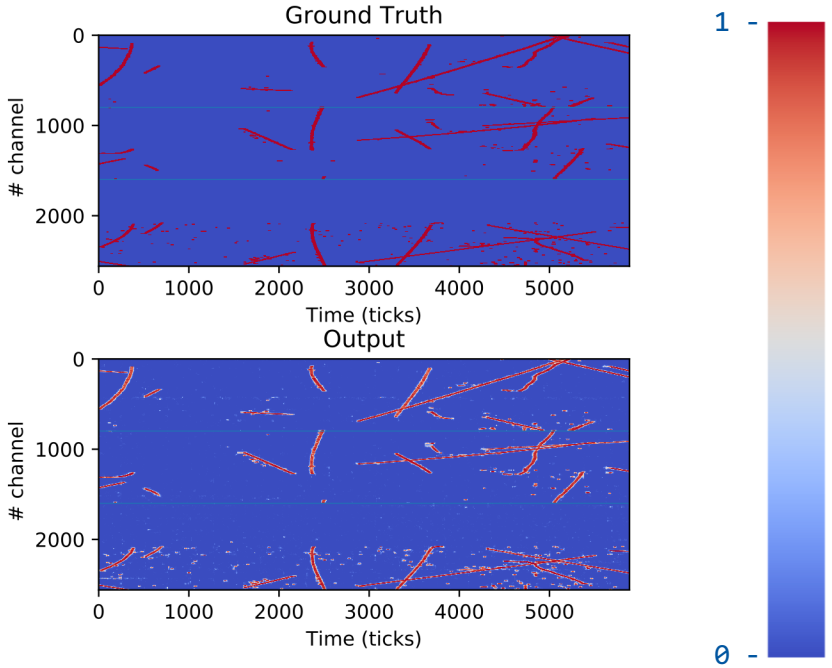
Data selection and trigger generation

- Focus on identifying areas of interest where there is activity on the detector.
- Fully Convolutional Networks to do image segmentation (**UNets**).
- **Input**: raw signals.
- **Goal**: checking the raw signals to get information from the waveforms.
    - **Locate where there are hits!**

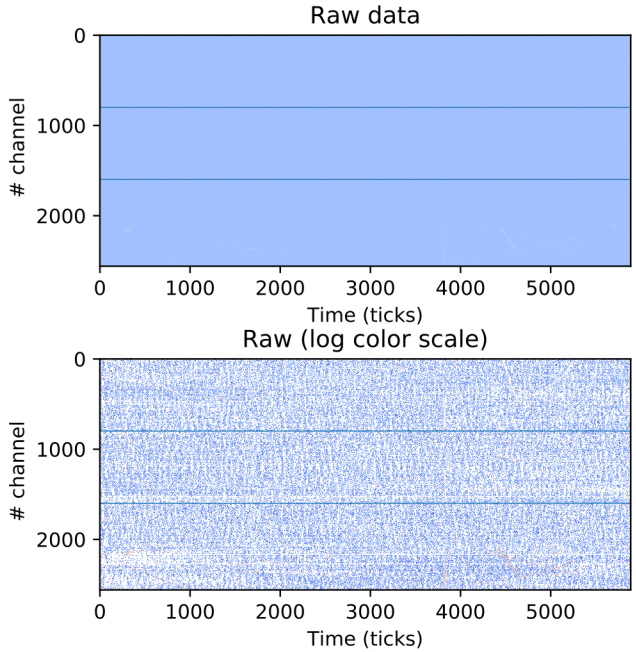


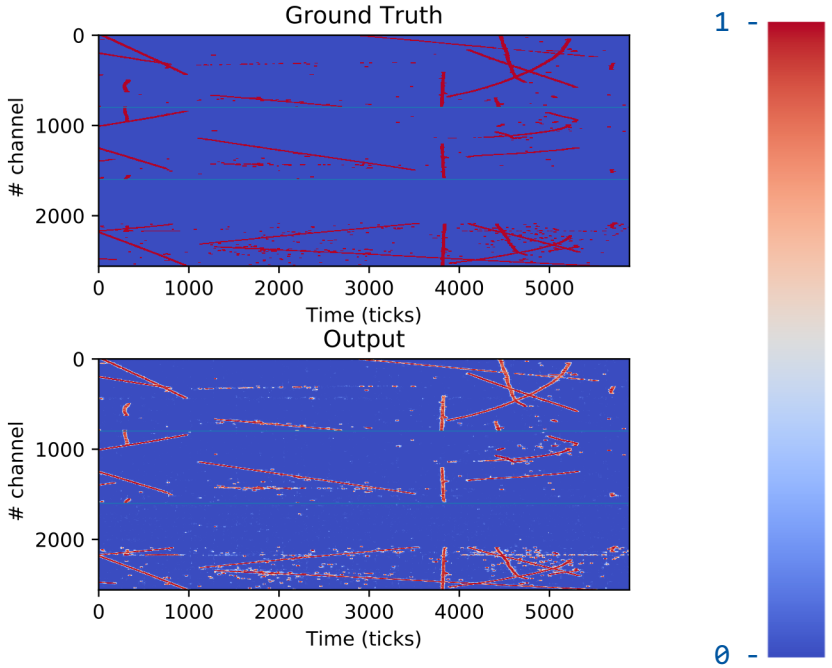
OFFLINE RESULTS



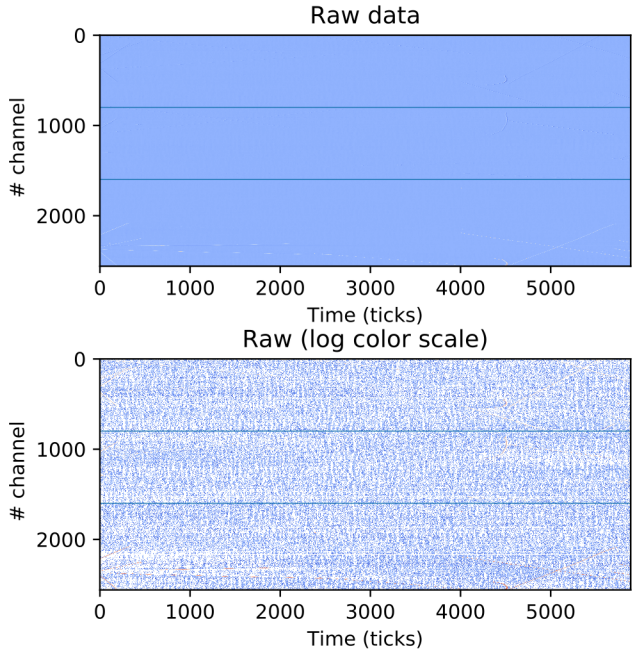


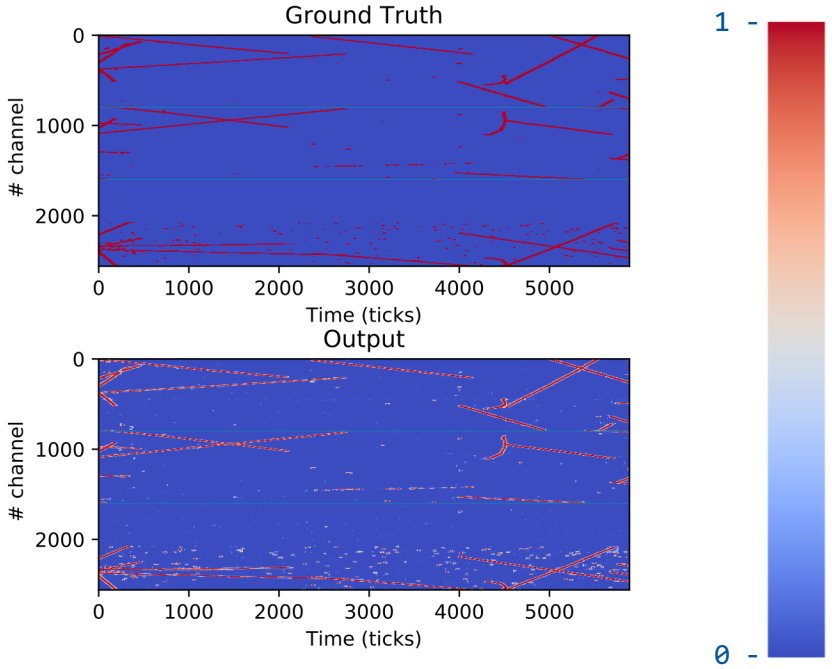
OFFLINE RESULTS





OFFILINE RESULTS





ONLINE SOLUTIONS

- We aim to find these Regions of Interest (RoI) on real time.
- To analyze a whole trigger window of 3 ms we need to run the inference over 15’360’000 pixels (2560 channels times 6000ms clock ticks)
- After some research and reducing the network to its minimum, this cannot be done with all the incoming data. We need triggered data.
- We our goal is to run it at 12.5 Hz ,meaning that we have 80 ms to run the inference, per trigger window.

MICRON DLA

Direct deployment of neural networks on the inference engine

Micron Deep Learning Accelerator[1]:

- No HDL programming.
- Natively supported neural networks.
- Most of the common layers are supported.
- Any framework that supports export to ONNX.
- Inference engine as an accelerator.

“Machine learning powers your world”

[1] https://fwdnxt.com/

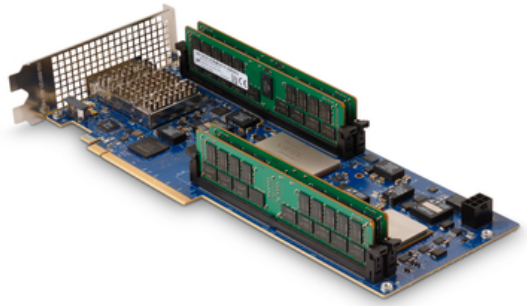
INFERENCE ENGINE

An FPGA ready for machine learning!

Micron Advanced Computing Solutions (ACS)

SB-852[1]:

- Xilinx Virtex Ultrascale+ UV9P.
- 64GB DDR4 SODIMM (up to 512GB).
- 2GB Hybrid Memory Cube.
- 2 QSFP transceiver connectors.
- PCIe x16 Gen3 to the host.



- With the 2 Clusters version, the inference will take 700ms. X

[1] https://www.micron.com/products/advanced-solutions/advanced-computing-solutions/hpc-single-board-accelerators/sb-852

INFERENCE ENGINE

An FPGA ready for machine learning!

Micron Advanced Computing Solutions (ACS)

AC-511 (x3)[1]:

- Xilinx Virtex Ultrascale+ UV7P.
- 16GB DDR4 SODIMM
- 2GB Hybrid Memory Cube.
- PCIe x8 Gen3 to the host.
- SDAccel (OpenCL) support

- With the 4 Clusters version,
- the inference will take 100ms.

➡ Almost nominal!

[1] https://www.micron.com/products/advanced-solutions/advanced-computing-solutions/ac-series-hpc-modules/ac-511

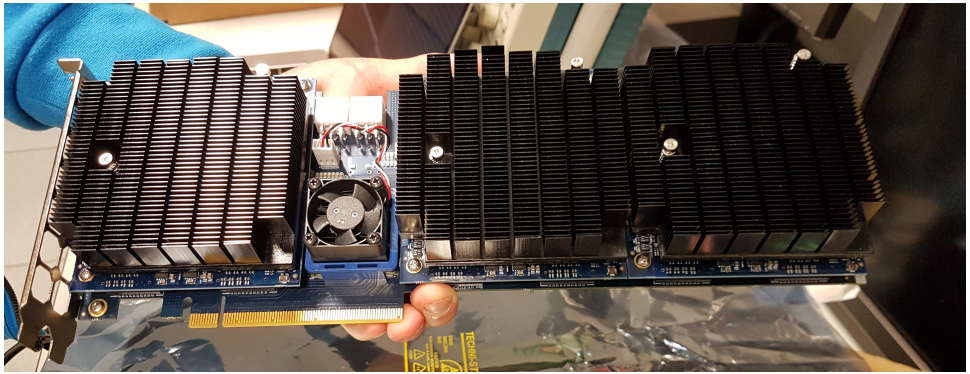
INFERENCE ENGINE

An FPGA ready for machine learning!

Micron Advanced Computing Solutions (ACS)

AC-511 (x3)[1]:

- Xilinx Virtex Ultrascale+ UV7P.
- 16GB DDR4 SODIMM
- 2GB Hybrid Memory Cube.
- PCIe x8 Gen3 to the host.
- SDAccel (OpenCL) support

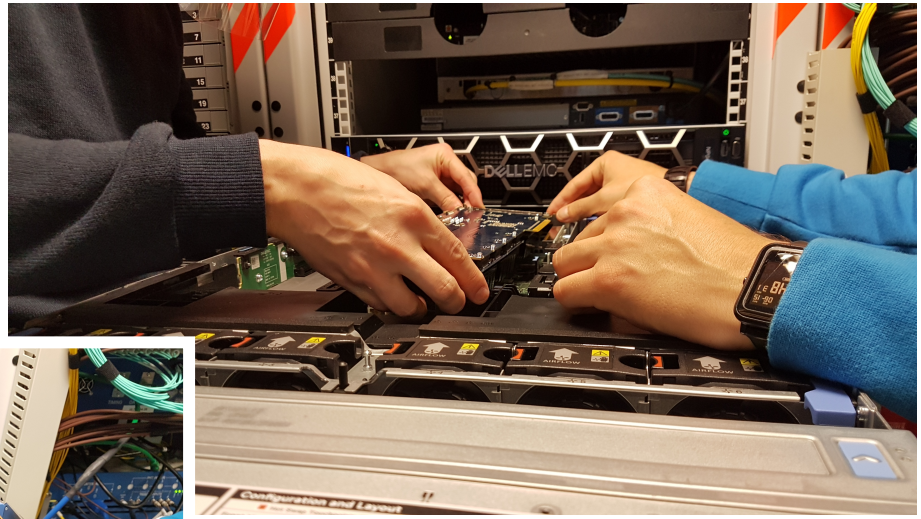


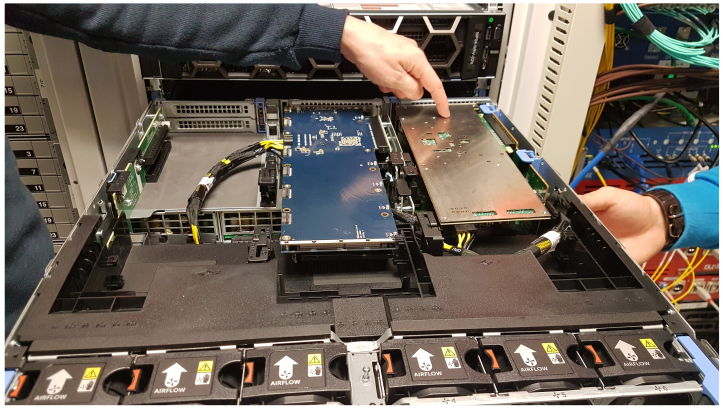
- With the 4 Clusters version,
- the inference will take 100ms.

➡ Almost nominal!

[1] https://www.micron.com/products/advanced-solutions/advanced-computing-solutions/ac-series-hpc-modules/ac-511

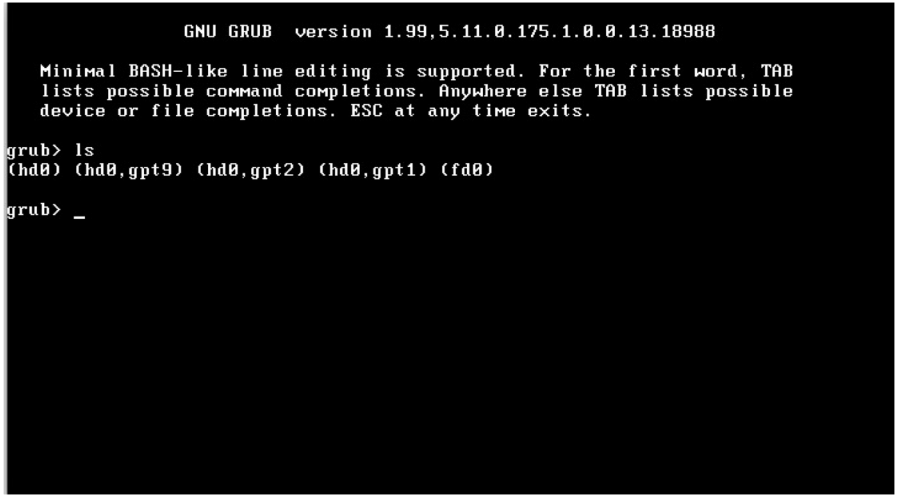
PROTODUNE-SP TEST





PROTODUNE-SP TEST

- We installed the driver for the Micron board and…
- We lost np04-srv-028

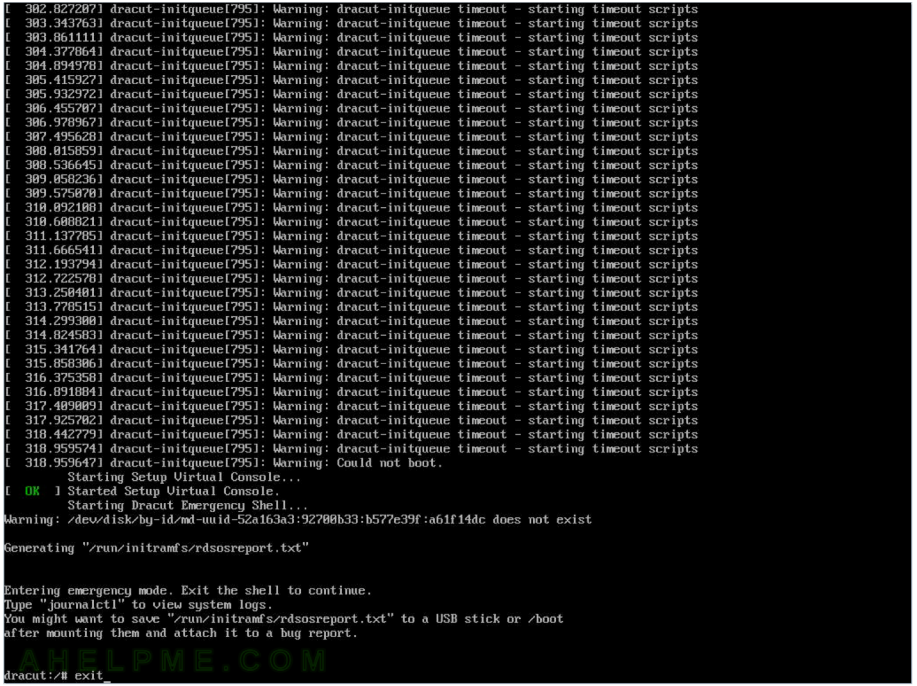


PROTODUNE-SP TEST

- We managed to fix the Grub, but Dracut wasn't happy either...

- The only solution:

  -> To call to our great System Administrators



PROTODUNE-SP TEST

- The diagnosis was that the driver module (which is compiled on the host to ensure compatibility) was corrupted. Therefore the system failed to load the module and all its dependencies.
- We reinstalled it and it worked.

TESTING THE BOARD ON srv-028

- We tested it over and over and over again.
- However, every time we tried to run the FPGA it was throwing a “bad fpga seq”
- At this point Micron joined the test.
- We tried together to debug it without any success. Even with a simple demo firmware on the FPGA it was failing.
- They thought that it could be a hardware failure.

```
[mjrodrig@np04-srv-028 ProtoDUNE-scripts]$ ./threadedbatchdemo -i test/ -s
tinylinknet_20200528.bin -r 1024x2560x1 -f 3 -C 4 -B
============================================================
ie_init: Initialize Micron DLA system
DLA binary to be read is tinylinknet_20200528.bin
------------------------------------------------------------
Using FPGA 0x511 Device 0511
^C
```

```
[Jun16 16:31] pico: bad fpga seq for fpga 1 stream 254! expected 0x190,
              got 0x180. last_host_seq: 0x180 (desc seq: 0x190)
[  +0.000003] pico:pico_newfw(): pico_newfw_internal() return error: -
              10011

[  +0.000002] pico: couldn't send 'read' command to system PicoBus: -10011
[  +9.303503] pico: interrupted while waiting for dma
```

TESTING THE BOARD ON srv-028

- We tried replacing one of the three FPGA, that seems faulty, but we still were having the same issue.
- Micron is still investigating this issue.
- Solution: Try the old SB-852

Not ideal at all.

```
[mjrodrig@np04-srv-028 ProtoDUNE-scripts]$ ./threadedbatchdemo -i test/ -s
tinylinknet_20200528.bin -r 1024x2560x1 -f 3 -C 4 -B
============================================================
ie_init: Initialize Micron DLA system
DLA binary to be read is tinylinknet_20200528.bin
------------------------------------------------------------
Using FPGA 0x511 Device 0511
^C
```

```
[Jun16 16:31] pico: bad fpga seq for fpga 1 stream 254! expected 0x190,
              got 0x180. last_host_seq: 0x180 (desc seq: 0x190)
[  +0.000003] pico:pico_newfw(): pico_newfw_internal() return error: -
              10011

[  +0.000002] pico: couldn't send 'read' command to system PicoBus: -10011
[  +9.303503] pico: interrupted while waiting for dma
```

TESTING THE BOARD ON srv-028

- We didn't manage to use the 4 Cluster version on the SB-852 (actually this firmware was experimental)

- With the 2 Cluster it takes 700 ms per trigger window

⬇

Trigger rate at 1.4 Hz 😔

```
[mjrodrig@np04-srv-028 ProtoDUNE-scripts]$ ./threadedbatchdemo -i test/ -s
tinylinknet_20200528.bin -r 1024x2560x1 -f 3 -C 4 -B
============================================================
ie_init: Initialize Micron DLA system
DLA binary to be read is tinylinknet_20200528.bin
------------------------------------------------------------
Using FPGA 0x511 Device 0511
^C
```

```
[Jun16 16:31] pico: bad fpga seq for fpga 1 stream 254! expected 0x190,
              got 0x180. last_host_seq: 0x180 (desc seq: 0x190)
[  +0.000003] pico:pico_newfw(): pico_newfw_internal() return error: -
              10011

[  +0.000002] pico: couldn't send 'read' command to system PicoBus: -10011
[  +9.303503] pico: interrupted while waiting for dma
```

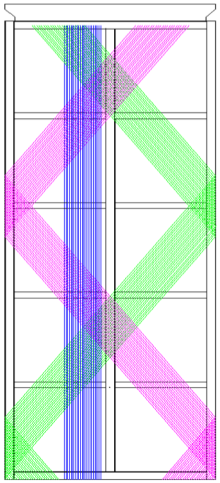
THE REORDERING ISSUE

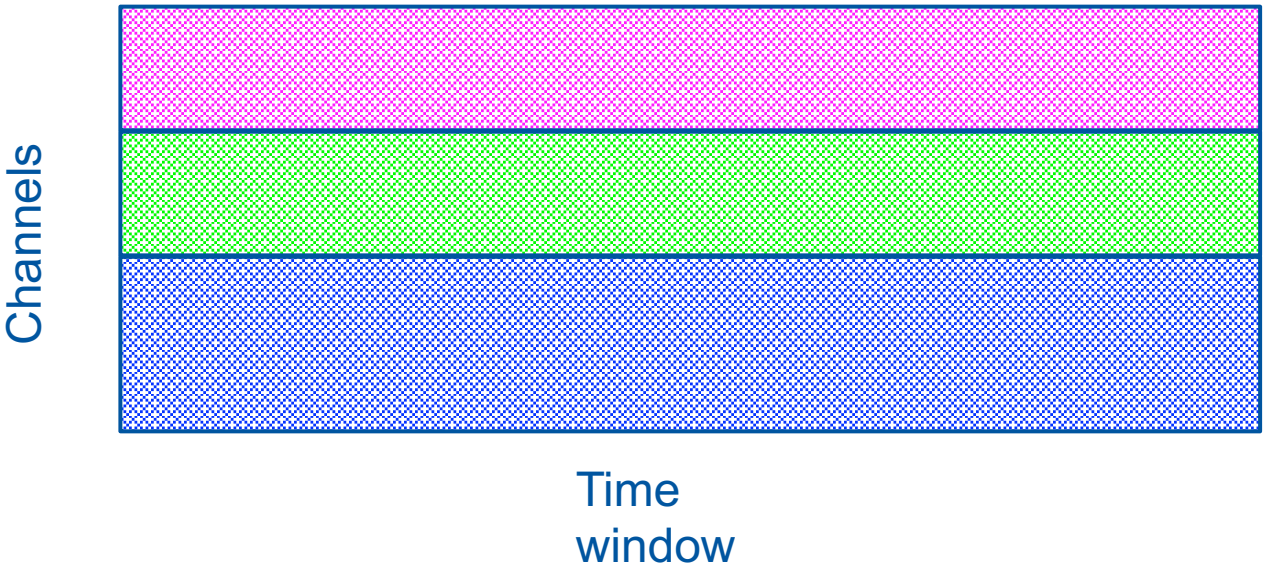
- After all the issues, we managed to send data to the FPGA, in a one shot approach.

- However, we found a totally different issue.

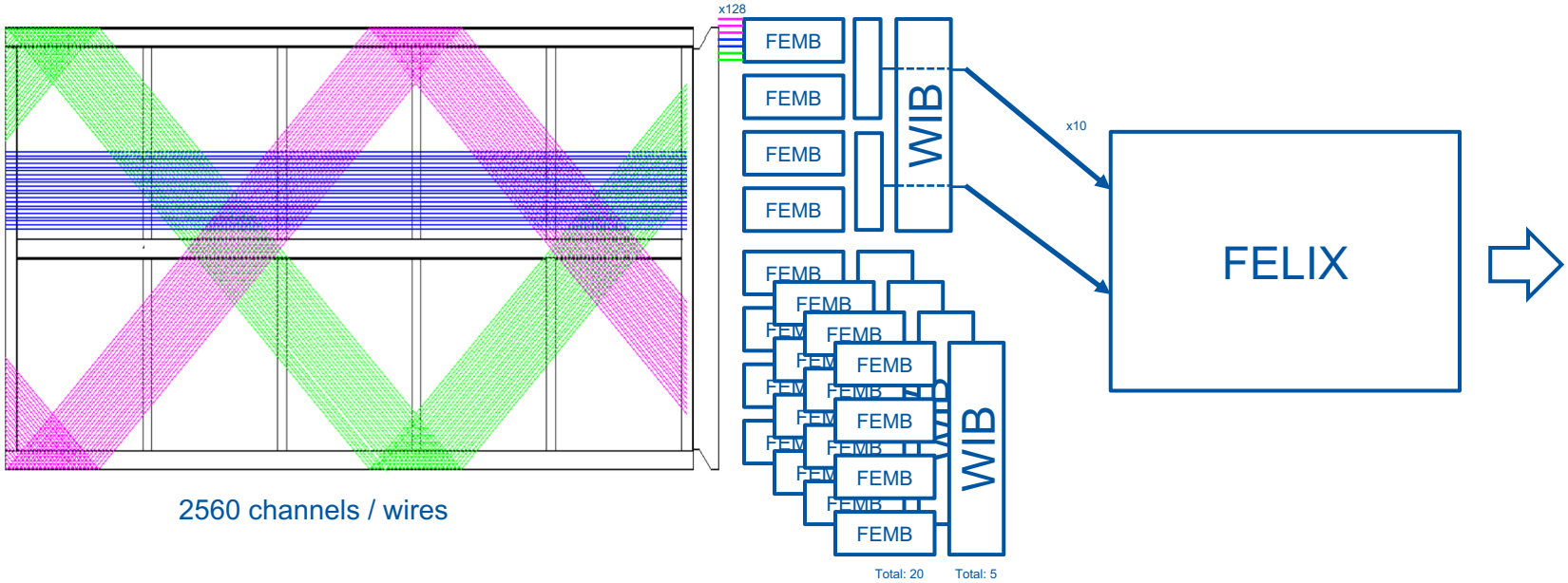
THE REORDERING ISSUE

- Our images in our dataset are like this:



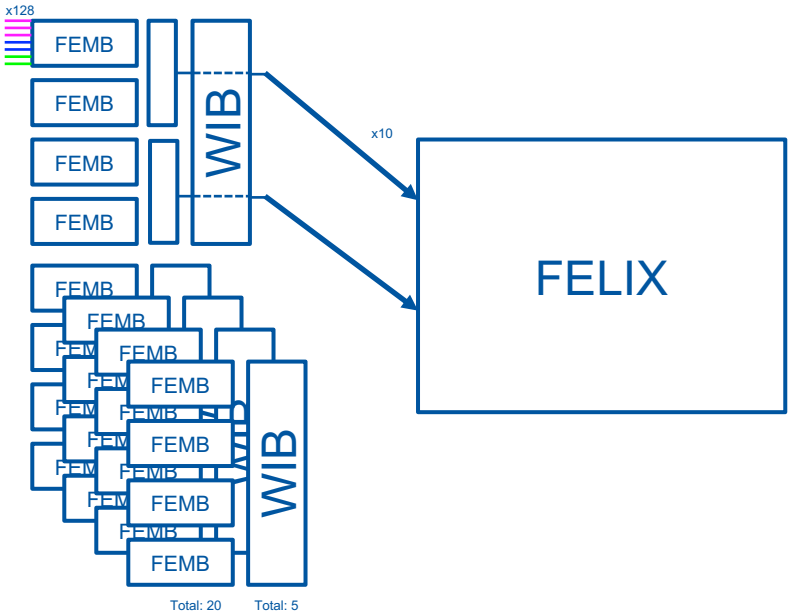


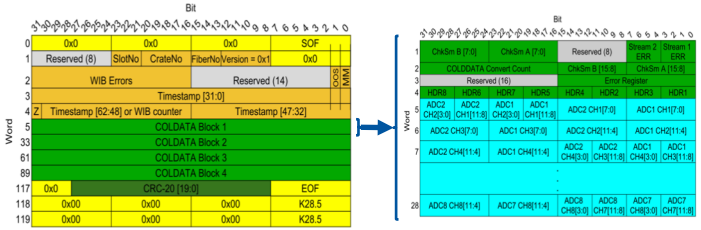
THE REORDERING ISSUE

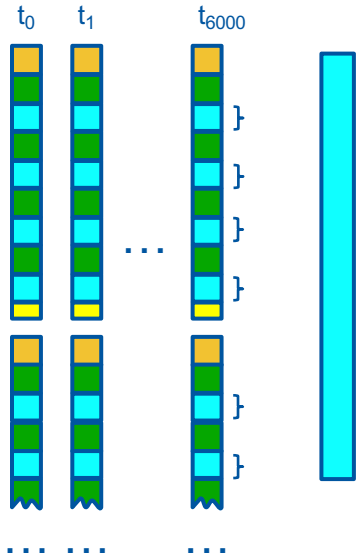


2560 channels / wires

THE REORDERING ISSUE

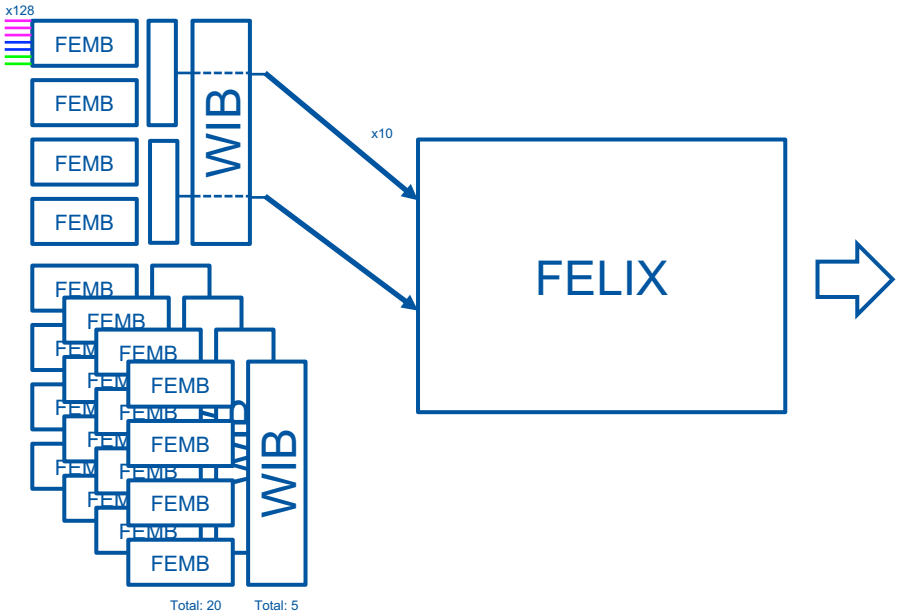


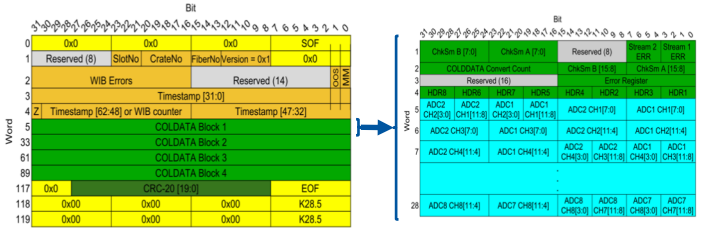


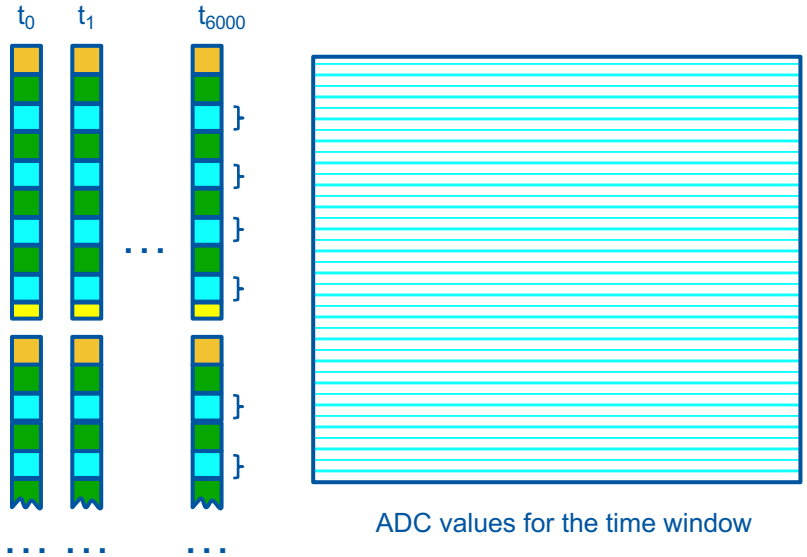


ADC values for t_0

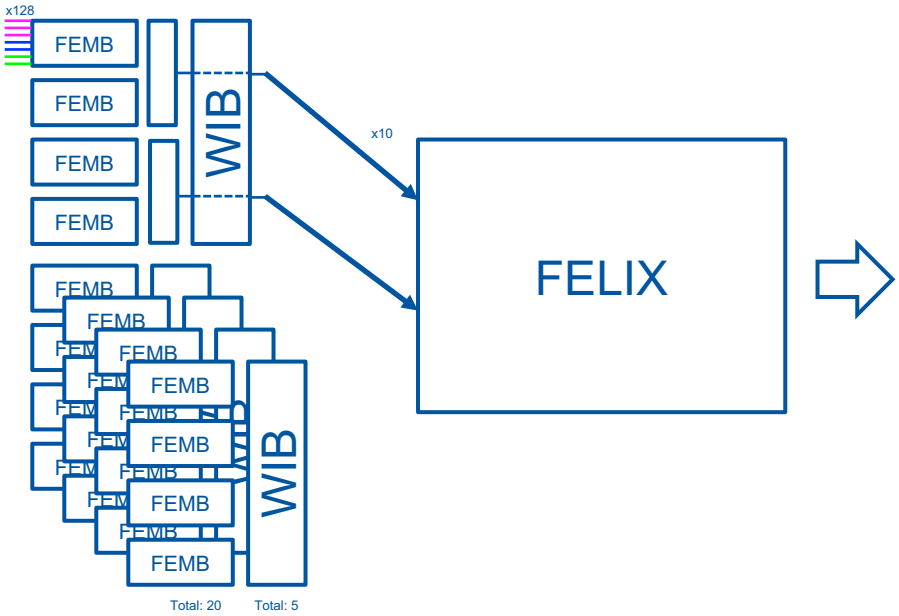
THE REORDERING ISSUE

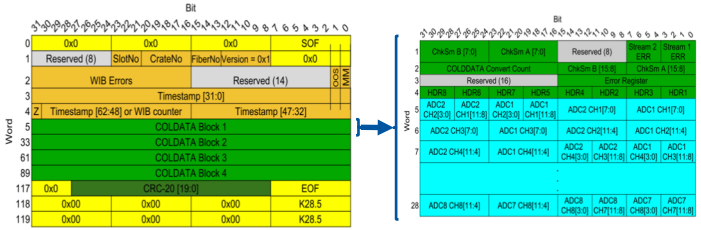


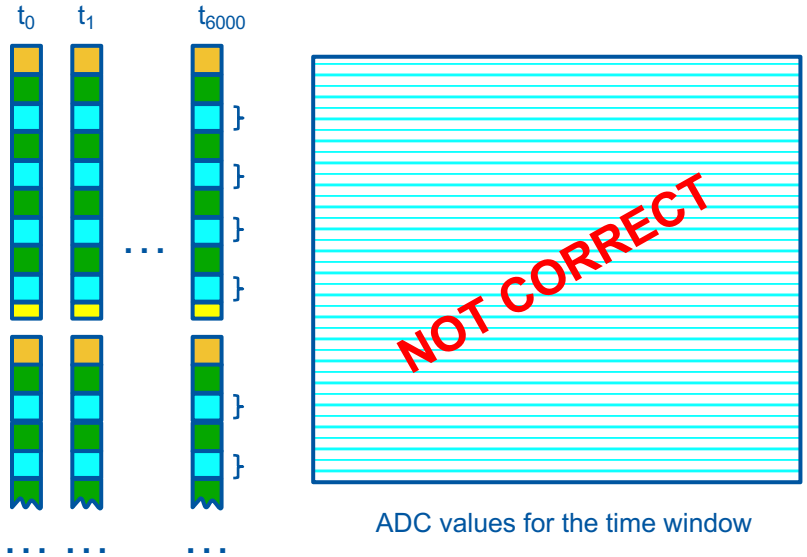




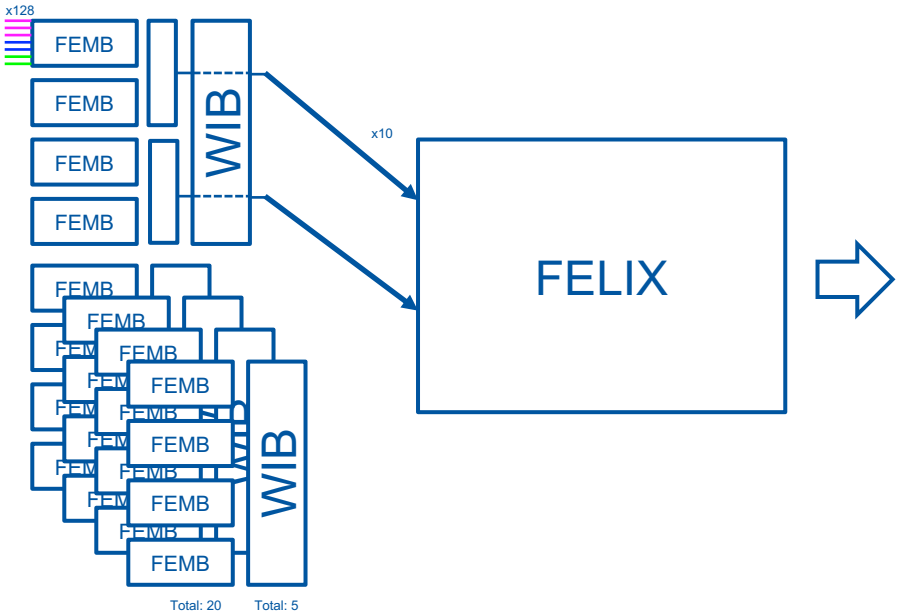
THE REORDERING ISSUE

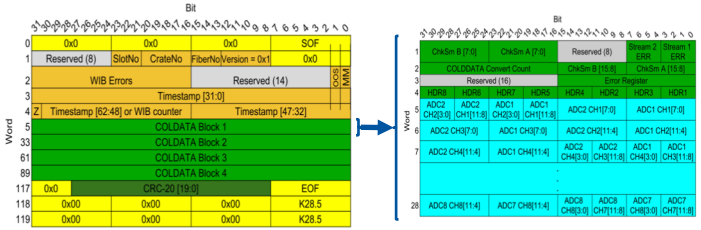


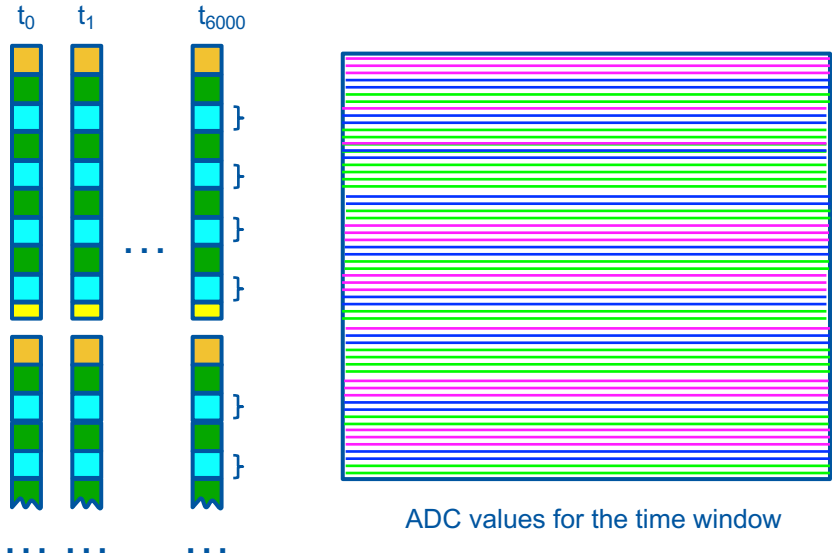




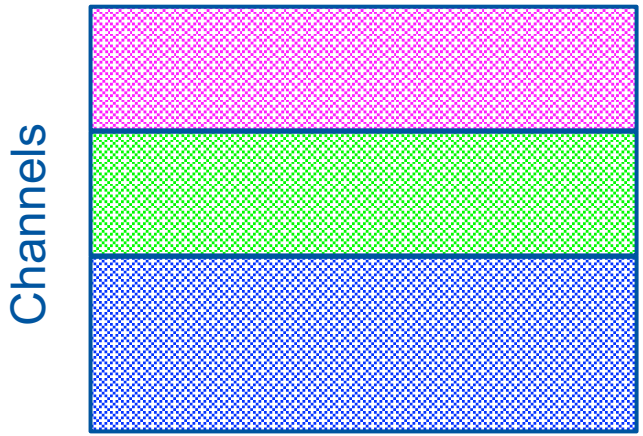
THE REORDERING ISSUE



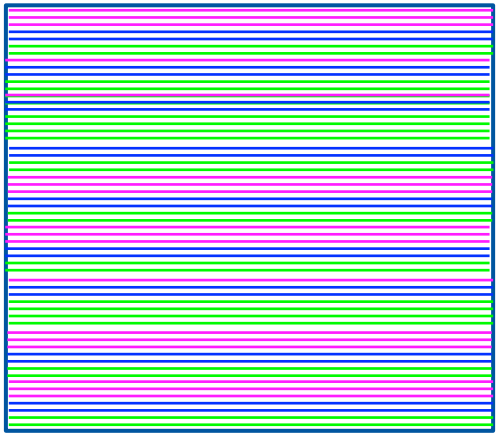




THE REORDERING ISSUE



≠



ADC values for the time window

THE REORDERING ISSUE

- Removing the headers is fine
- Reordering the data (2560 channels time 6000 ticks) is not.
- Possible solutions:
  - Retrain the network using the online channel number
  - Do the reorder on FPGA (FELIX or Inference Engine)

CONCLUSIONS

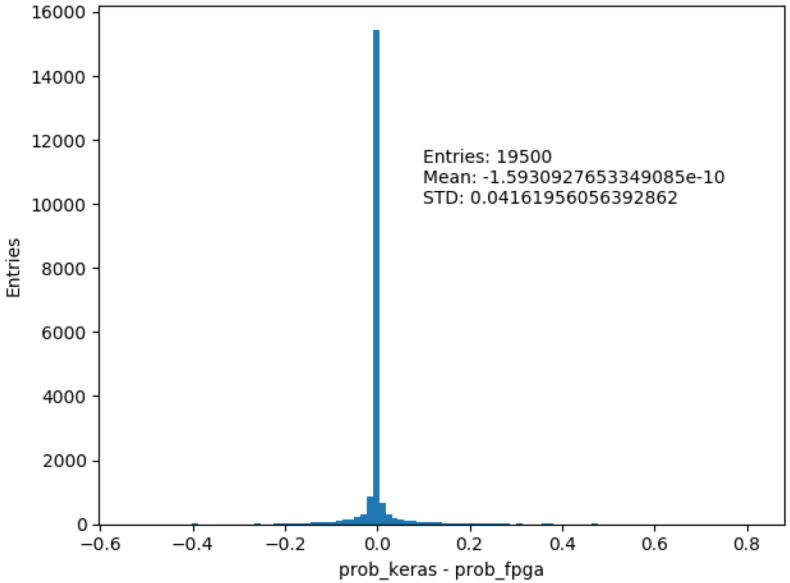
- We wanted to test the integration of the Micron DLA on the protoDUNE DAQ chain.
- The hardware we used was an unreleased version made for this test with some issues unseen before. Thanks to the test, Micron can study it and debug it to make the system more robust.
- We faced as well a different issue not taken into account. In my opinion, it was great that we worked with online raw data. This gave us a much better understanding on how that data is coming from the detector.
- It's a pity that we don't have more time to test. However, thanks to the binary data recorded we can continue evolving the system.

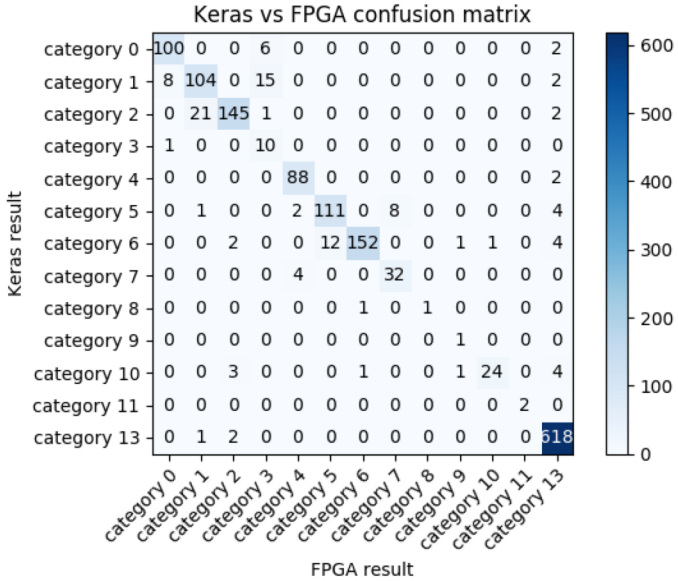
THANK YOU

BACKUP

GPU-FPGA RESULTS COMPARISON

How good our FPGA behaves





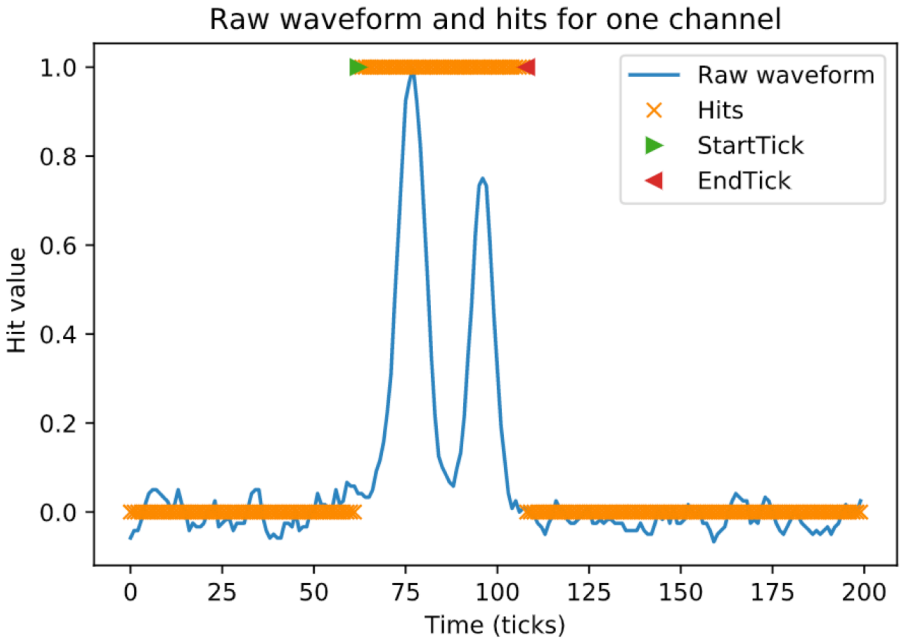
Our dataset

- On the hits file we have:

```
(int)hit.Channel(), hit.StartTick(),
hit.EndTick(), (int)hit.SummedADC(),
(int)hit.RMS()
```

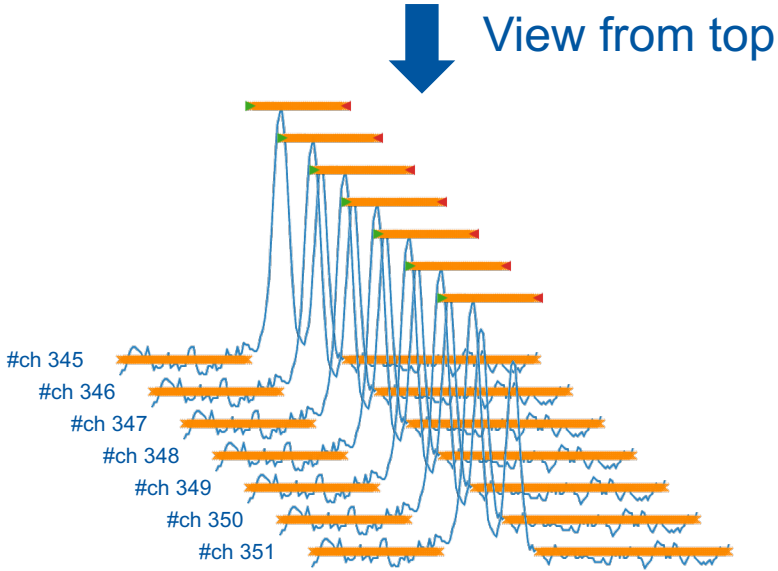
- We take the StartTick and EndTick and we mark the whole range as

```
hit(channel,[startTick,endTick]) = TRUE
```



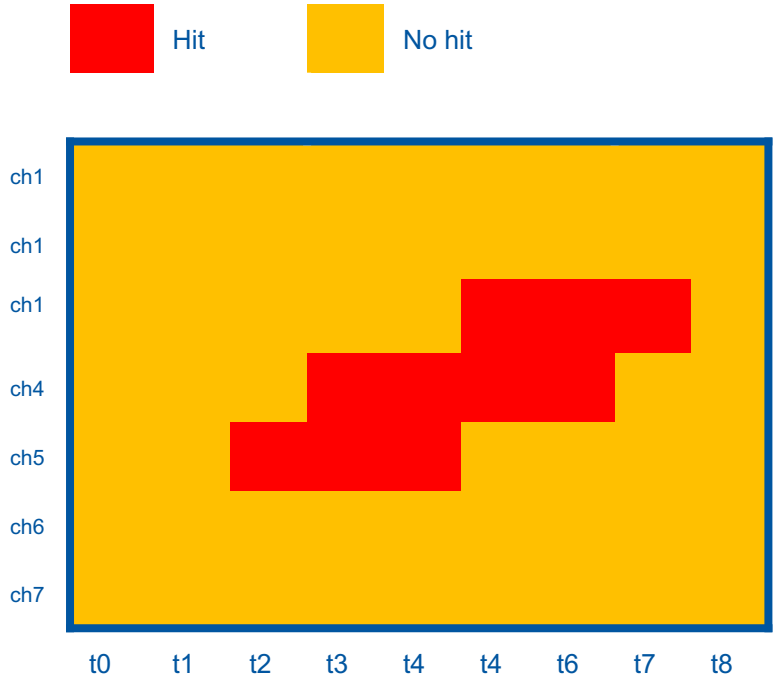
Region of interest

- Once we have the hits for all the channels.
- We artificially augment the hits area in time and channels to get our region of interest.
- $\forall\ i, j$: If hit(i,j) == 1
  - hit(i +1, j) = 1
  - hit(i -1, j) = 1
  - hit(i, j+1) = 1
  - hit(i, j-1) = 1
  - hit(i +1, j+1) = 1
  - […]



Region of interest

- Once we have the hits for all the channels
- We artificially augment the hits area in time and channels to get our region of interest
- $\forall\ i, j$: If hit(i,j) == 1
  - hit(i +1, j) = 1
  - hit(i -1, j) = 1
  - hit(i, j+1) = 1
  - hit(i, j-1) = 1
  - hit(i +1, j+1) = 1
  - […]



Region of interest

- Once we have the hits for all the channels
- We artificially augment the hits area in time and channels to get our region of interest
- $\forall\ i, j$: If hit(i,j) == 1
  - hit(i +1, j) = 1
  - hit(i -1, j) = 1
  - hit(i, j+1) = 1
  - hit(i, j-1) = 1
  - hit(i +1, j+1) = 1
  - […]
- We use the augmented area as our ground truth for the neural network