This space left intentionally blank
Please do not adjust your projector.

INDRA Lab projects

JLEIC meeting

June 14th 2019

Streaming ReadOut Workshop IV – Camogli, Italy

- Two day workshop covered:
  - Verification/Validation techniques for Streaming Readout
    - Contributions from PANDA, ALICE, CLAS12, BDX
  - Electronics – Front End, Timing and Synchronization
    - ASIC and Micro-Electronic developments, Ultra Fast Silicon Detectors for Timing/Tracking, DAQ Electronics for Large Detectors[Industry Partners – CAEN], FADC250 in Streaming mode with Ethernet, and Streaming Readout Timing/Synch and TDC
  - Streaming Readout Software
    - Real-Time Analysis at LHC, Software Consortium report, TriDAS for EIC, Prototype [Protocol] results
  - Streaming Readout for EIC Detectors
    - TOPSIDE, JLEIC, eRHIC/FELIX-DAq, eRD23 proposal discussion

https://agenda.infn.it/event/18179/overview

Streaming ReadOut Workshop IV – Camogli, Italy

- Broadly and generally, the presentations were very good and in particular it was encouraging to see that other groups have implemented SRO techniques. PANDA and ALICE are two examples where SRO hardware and software have been implemented.

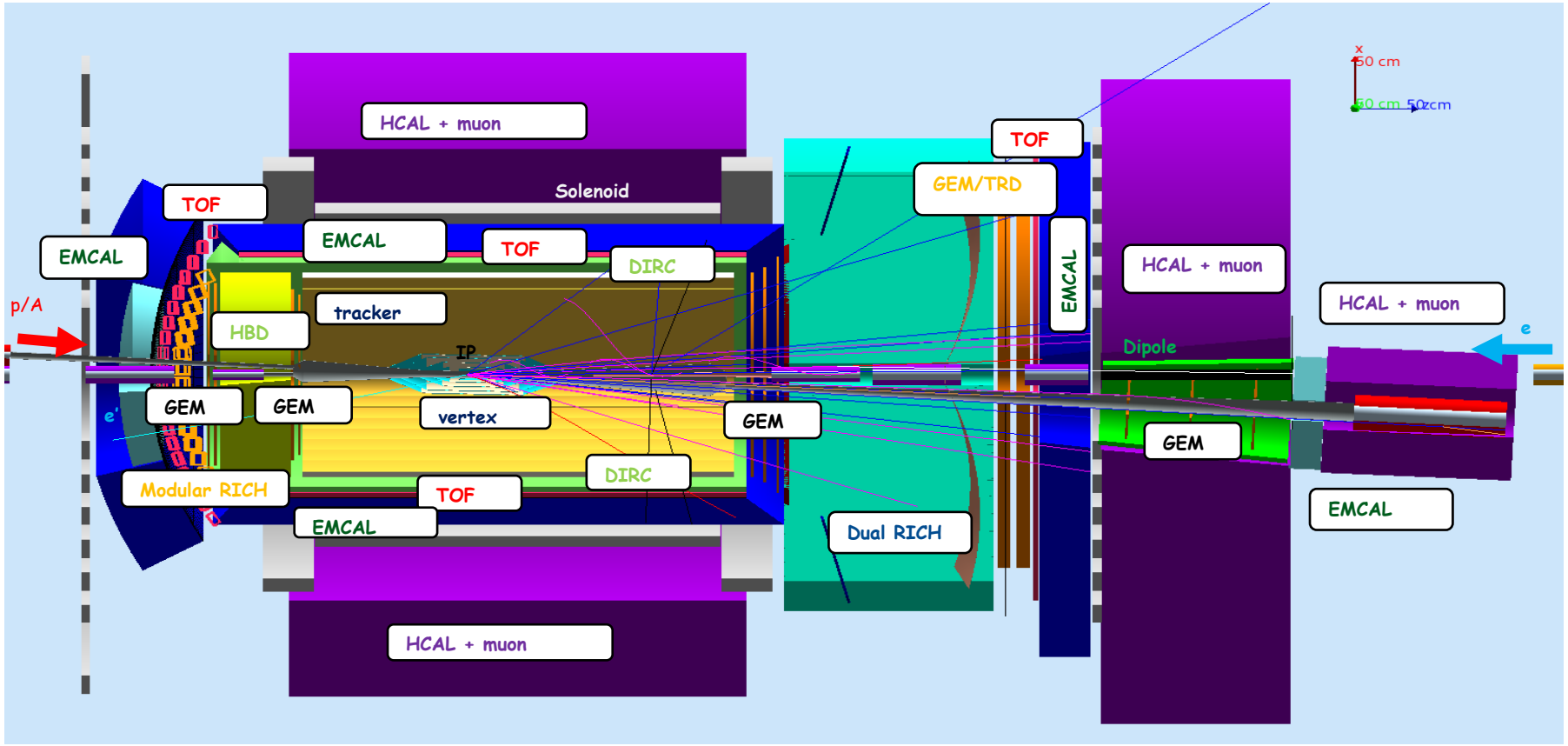
- The software work and report from the consortium were very informative plus we heard about very new micro-electronics [ASIC] and developments with Ultra-Fast Silicon detectors for timing/tracking during the hardware session.

- All of the EIC session included presentations that were similar in terms of channel count, data rates and methods for DAQ. The eRHIC/FELIX solution is advanced because the hardware will be used for sPHENIX.

- Definitions for the SRO data streaming protocol were presented and discussed. This protocol including other specific ‘standards’ for hardware and software will have to be documented and circulated soon. Good progress since 2018-December’s workshop.

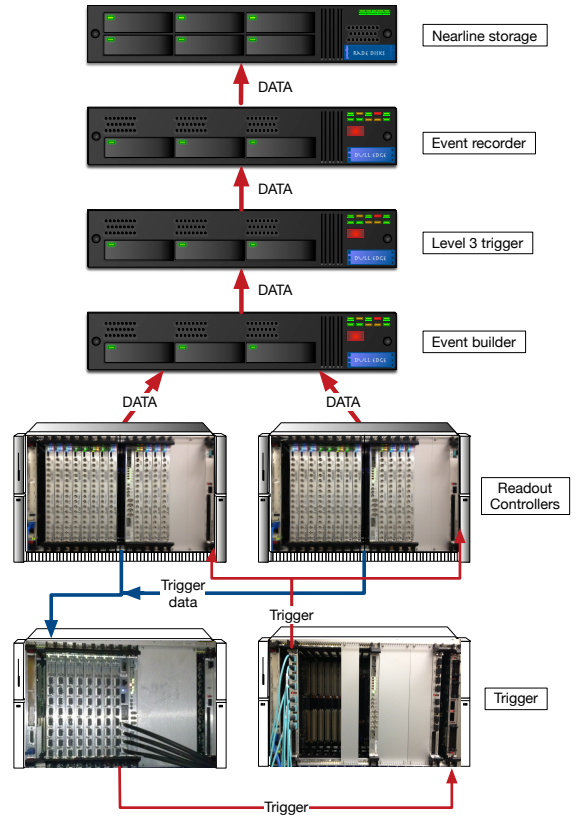
Example EIC detector design

- Just counting labels on the diagram there are ~25 detector packages.
  - Wide range of response times for the detector types.
- The largest single channel count is the Vertex Detector.



What happens if we use traditional DAQ - crates

- Back of the envelope calculation ignoring the Vertex detector (which could be 20-50 M channels).
- The rest of the detector is ~1M channels.
  - CLAS12 : ~90k channels read by 100 ROCs
  - GLUEX : ~40k channels read by 50 ROCs
  - Average ~1 ROC per 1000 channels, seems like a lot of channels per ROC but is dominated by high channel count detectors.
  - EIC detector would be ~1,000 ROCs.
    - We need to distribute triggers to 1000 devices.
    - We could have up to 1000 devices contributing signals to the trigger.
- Assume average 1% occupancy.
  - Vertex detector rate ~240 GB/s. (yes bytes)
  - Rest of the detector ~5 GB/s total.
    - Fair agreement with CLAS12 and GLUEX if we were to scale them up to 100 kHz and 1% of 1M channels.



The Vertex detector

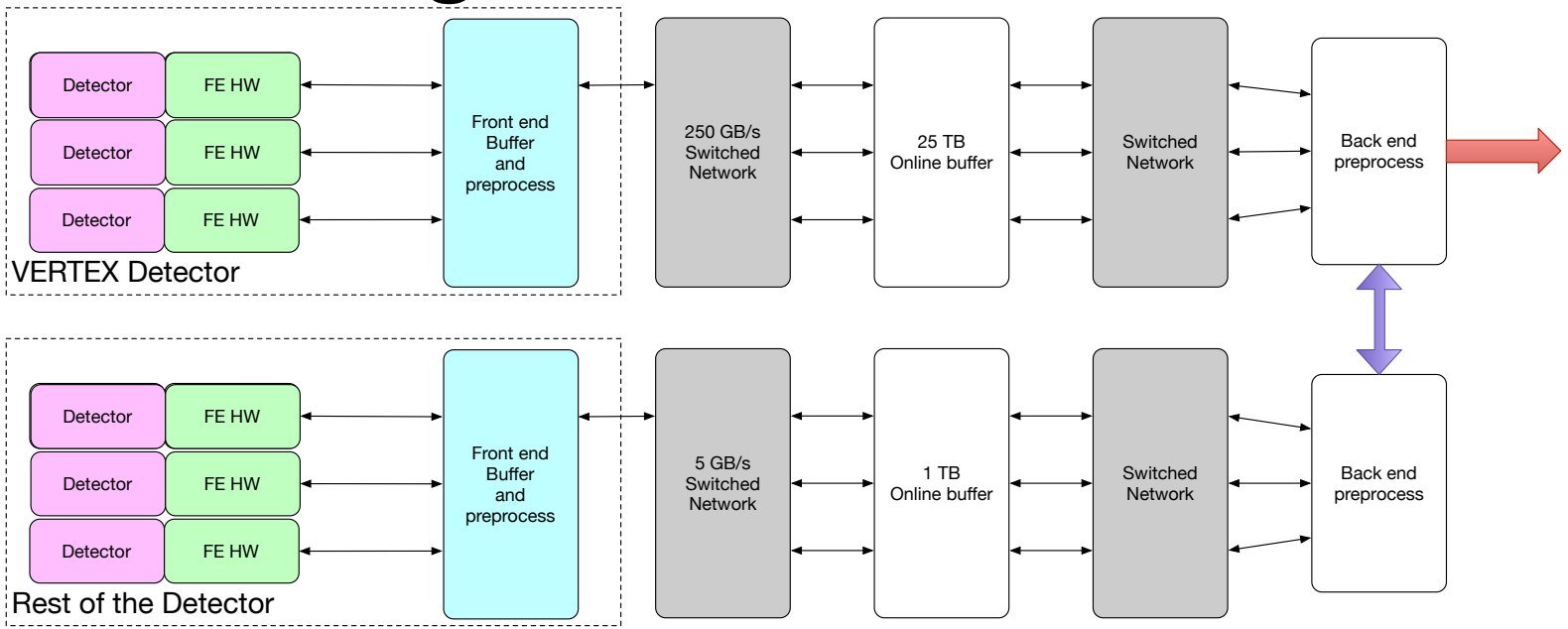
- How would you read out a detector that generates 240 GB/s ?

- The only way that even remotely makes sense is massive parallelism.
  - Split the detector into small regions and read those out in parallel.
  - No sensible way to sync the different regions in real time without spending a lot on electronics.

- Aim to reduce the rate to storage by using data from the rest of the detector to define regions of interest.
  - Have to hold on to the Vertex Tracker data until R.O.I. can be identified.

- Dealing with the Vertex Tracker dominates the design of the DAQ.

Dealing with the Vertex detector

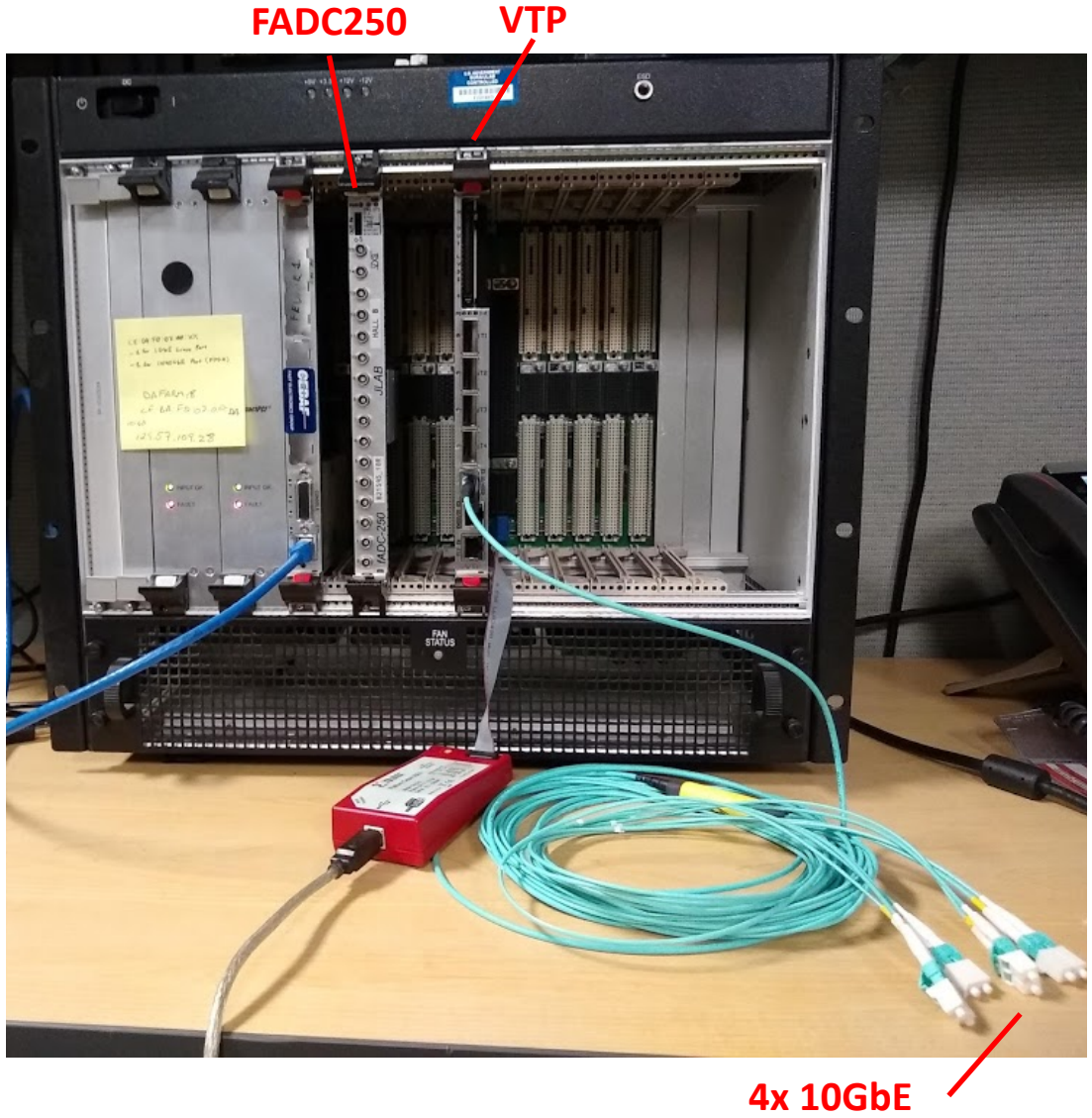


- Vertex Detector is read in parallel streams into online buffers.
  - Say 25 front end buffers at 10 GByte/s (Using today's 100 Gbit/s HW).
  - Main Online buffer is 25 nodes with 1TB of memory each ~100s buffer time.
- Rest of detector streams to a smaller online buffer, 5 GB/s total, single 1TB buffer ~200s buffer time.
  - Actually almost identical to 1/25th of the Vertex system – so 26 main streams in total.
- Identify regions of interest in Vertex Detector : 4D regions = 3D volume in detector + time range.
- Vertex Detector back end processors pull ROI data from online buffer. Unwanted data is discarded

Ben's talk - Read FADC250 via VTP - Current Test Setup

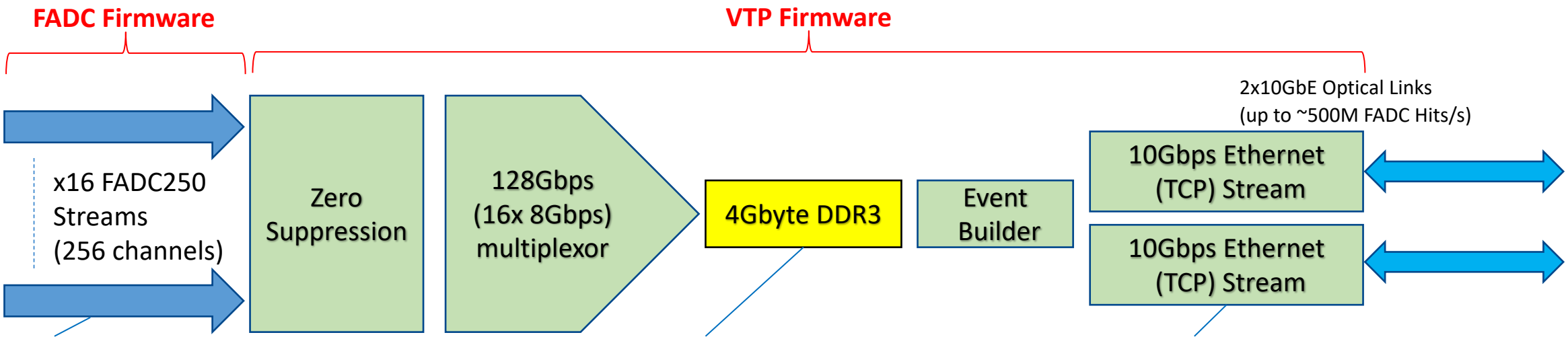
- Current test setup sits in Ben's office:
  - Small VXS Crate
  - 1 FADC250
  - 1 VTP
  - Old PC w/10GbE (Mellanox ConnectX-3)

- Will move to INDRA-ASTRA lab soon
  - Expanding to 16 FADC250 modules
  - High performance servers



Ben - Firmware Development

- FADC250 – no firmware development needed
  - Reusing trigger path, which discriminates and provides pulse time and charge
- VTP – nearly all firmware completed



16x FADC 8Gbps streams, each:
- 16 channels
- 32ns double pulse resolution
- 4ns leading edge timing
- Pulse integral

Large buffer:
- 200Gbps bandwidth
- Allows high burst rates
- (doesn't need to be this big, but it's what the hardware has, so might as well use it)

TCP Stack:
- Hardware accelerated
- Up to 4 links can be used
- Could be a single 40GbE (if we could afford the TCP stack IP – not worth it for R&D)

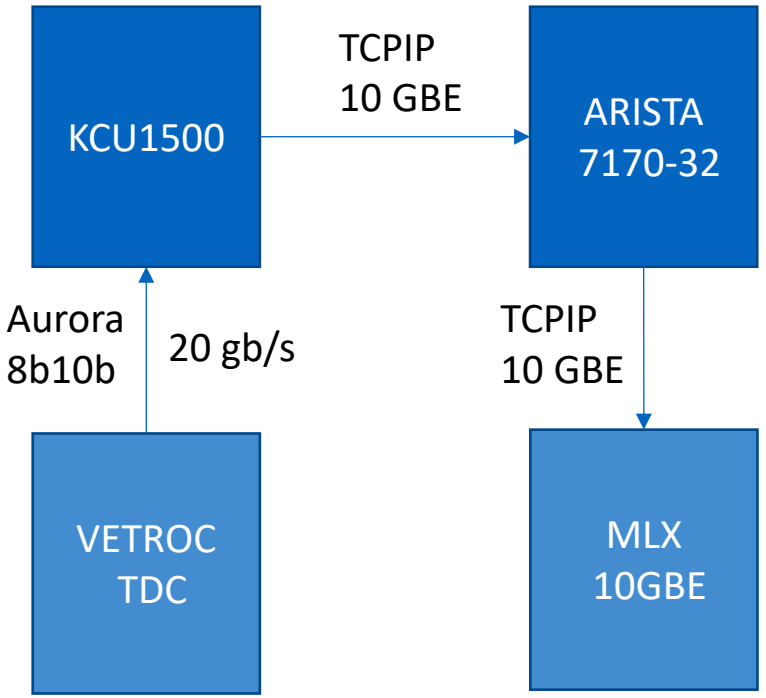
Ben's talk continued

- TCP hardware accelerated stack is probably one of the trickier parts that luckily we have a vendor providing. We have found a number of issues with the IP, but the vendor has been working with us to resolve them – shouldn't prevent us from reach test goals.

- Delays due to CLAS12 & HPS experiment preparations, but these will be complete in the next few weeks so Ben can actually spend good time to wrap up this project!

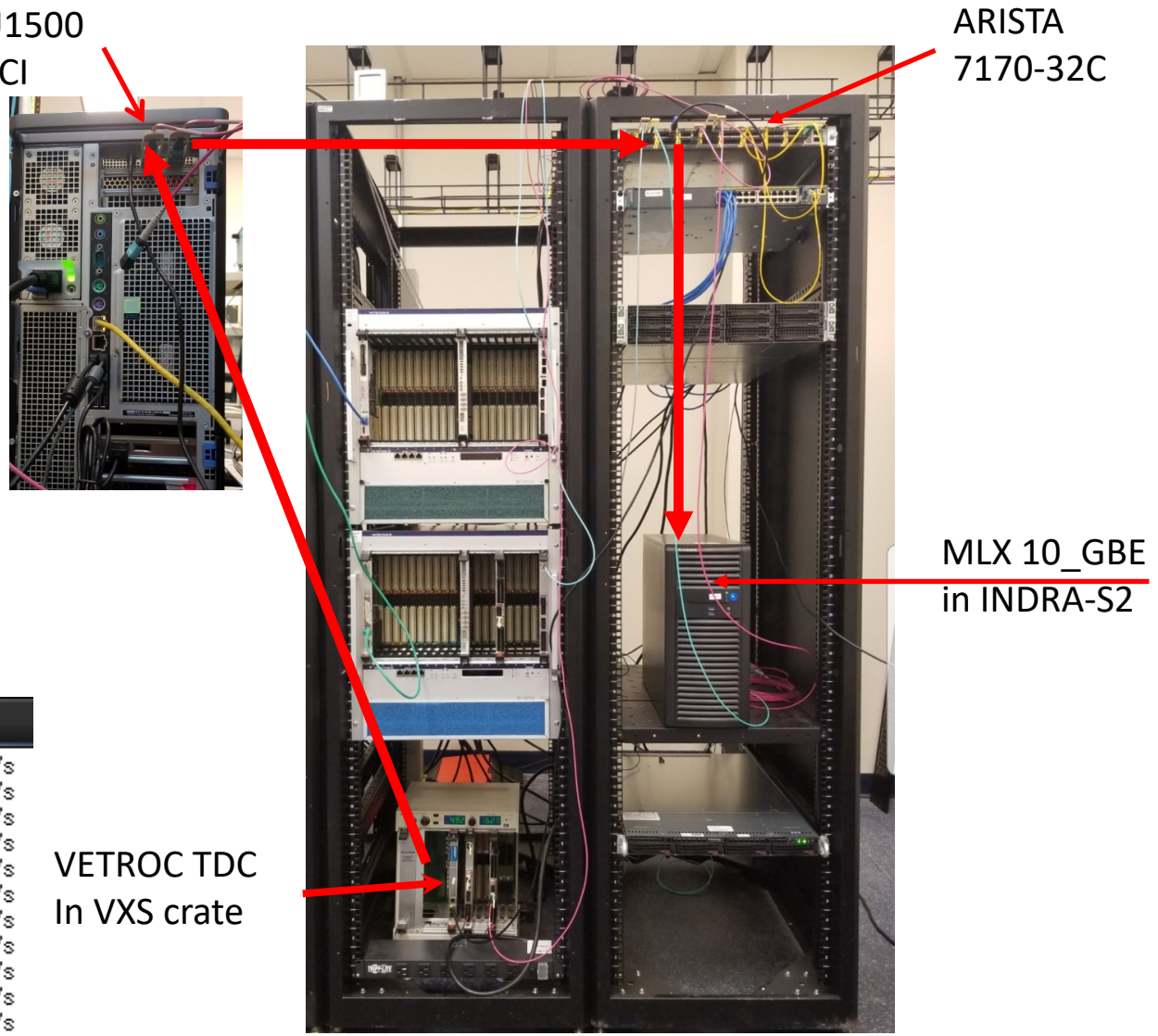
- Making progress towards FADC250 crate streaming over Ethernet using TCP.
  - expected to have a functional demonstration this summer!

William - Streaming Readout of a VETROC 128 chan streaming TDC



Graham's
Stream_Router
Software
~400 Mbyte/s

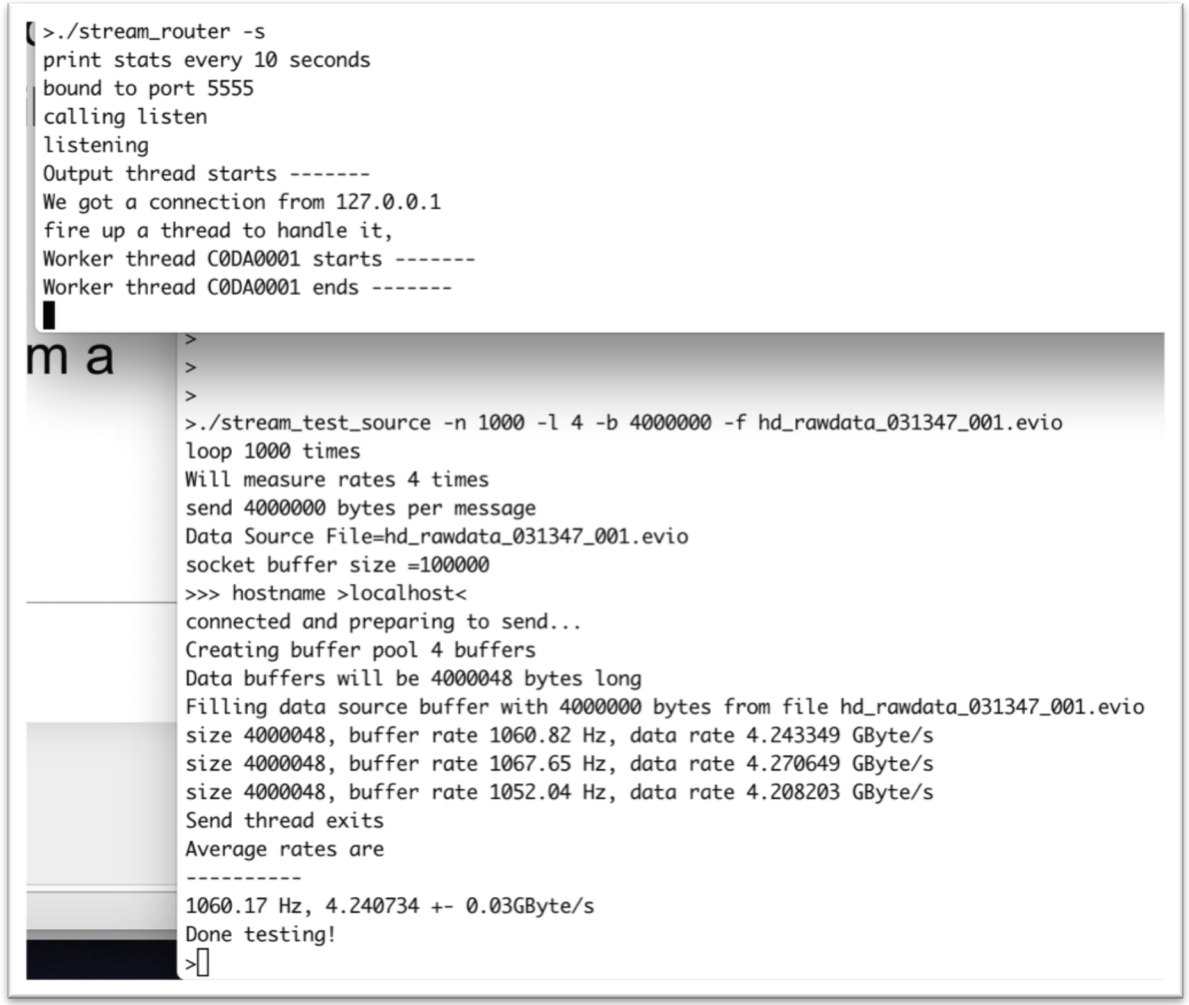
```
xterm (on indra-s2)
ID CODA0048 - buffer rate 379895.25 Hz, data rate 0.407248 GByte/s
ID CODA0048 - buffer rate 391878.30 Hz, data rate 0.420094 GByte/s
ID CODA0048 - buffer rate 381900.90 Hz, data rate 0.409398 GByte/s
ID CODA0048 - buffer rate 389568.43 Hz, data rate 0.417617 GByte/s
ID CODA0048 - buffer rate 382177.78 Hz, data rate 0.409695 GByte/s
ID CODA0048 - buffer rate 386047.66 Hz, data rate 0.413843 GByte/s
ID CODA0048 - buffer rate 369711.18 Hz, data rate 0.396330 GByte/s
ID CODA0048 - buffer rate 357362.87 Hz, data rate 0.383093 GByte/s
ID CODA0048 - buffer rate 376956.18 Hz, data rate 0.404097 GByte/s
ID CODA0048 - buffer rate 366132.41 Hz, data rate 0.392494 GByte/s
ID CODA0048 - buffer rate 398968.96 Hz, data rate 0.427695 GByte/s
ID CODA0048 - buffer rate 390347.00 Hz, data rate 0.418452 GByte/s
ID CODA0048 - buffer rate 393121.97 Hz, data rate 0.421427 GByte/s
ID CODA0048 - buffer rate 359522.01 Hz, data rate 0.385408 GByte/s
```



Streaming test code (not shown at workshop)

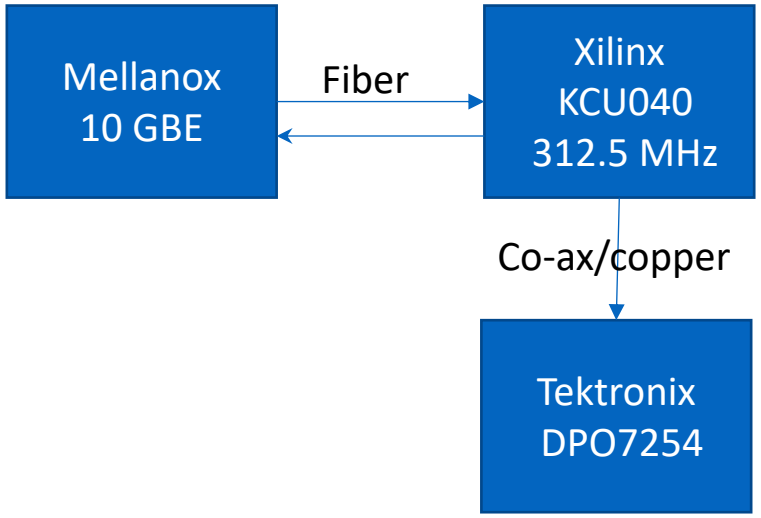
Three pieces of test code:

- stream_test_client
  - Simulated data source.
  - Sends fixed length data blocks.
  - Data from file or random numbers.
  - Sends multiple blocks to allow rate tests.
- stream_router
  - Receives TCP packets from data source.
  - Optionally route to ZeroMQ subscriber.
  - Measures throughput.
- stream_test_subscriber
  - Example of how to receive data from a stream_router.



William's talk – Clock Recovery study

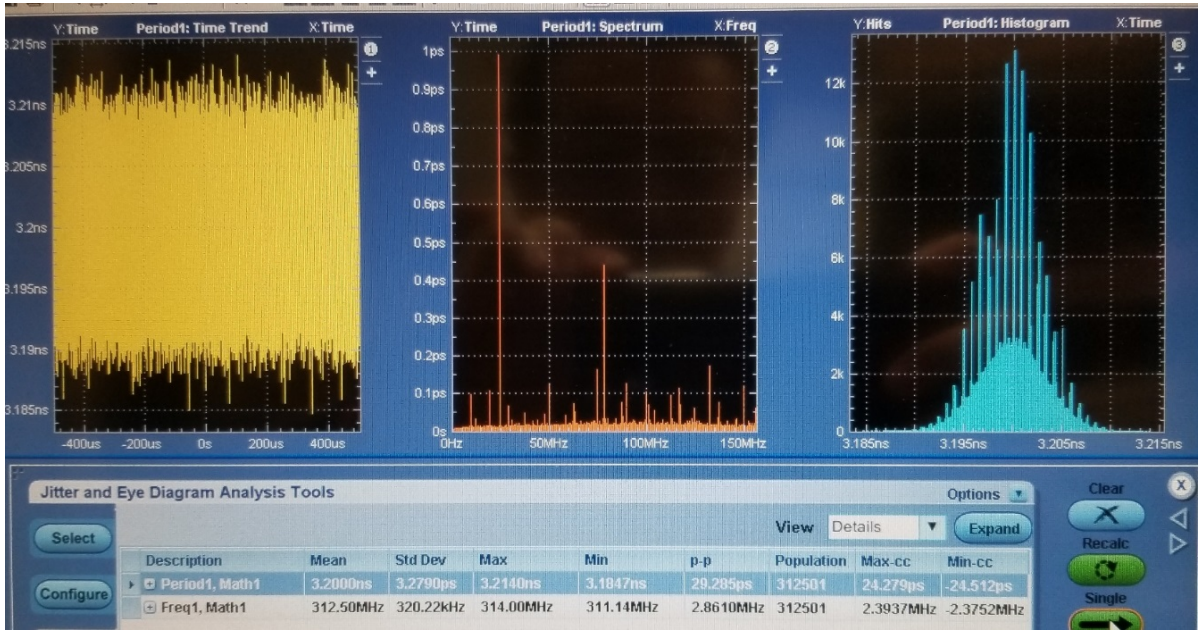
How to do the timing?
Can a low jitter clock can be recovered from the fiber data links?



Mellanox
(10 GBE)

Xilinx KCU105
(UltraScale Kintex)

Recovered clock to Tektronix DPO7254



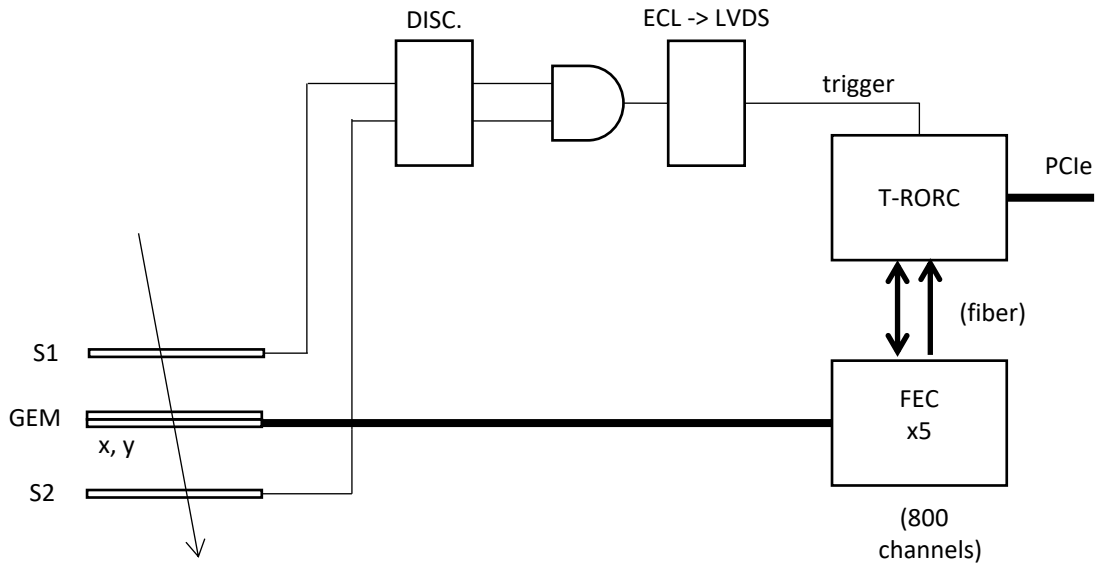
Recovered clock period: 3.2000ns
Std Dev: 3.28 ps
Enabling the Mellanox data on same link, Std Dev → 3.30 ps
Enabling the KCU040 data on same link, Std Dev → 6.42 ps

Scope single pulse period measurement: Std Dev: 6~7 ps
DPX mode (200 Gsample/s): 700~900 fs

Answer : yes!

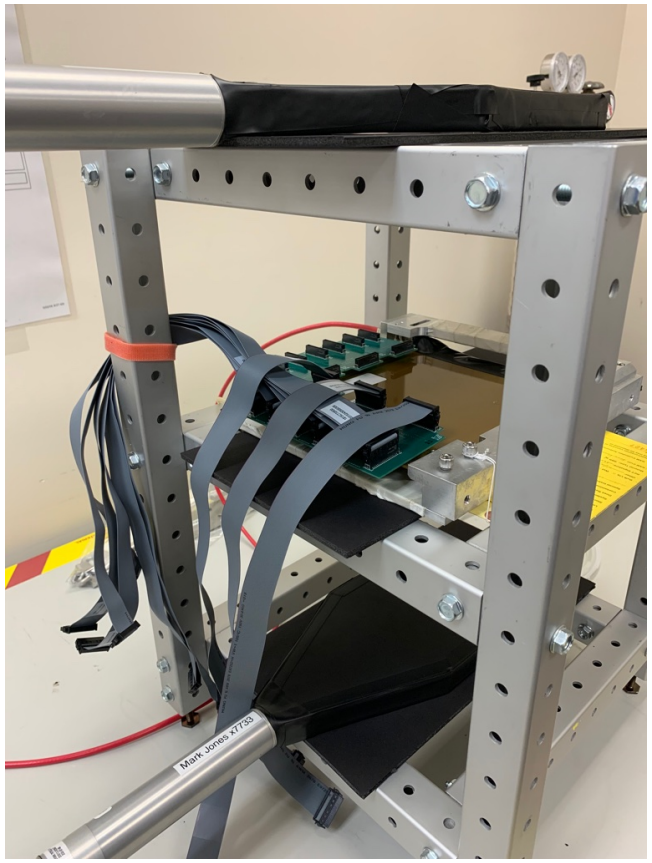
Streaming GEM RO cosmic Ray Test Setup – Ed & Eric

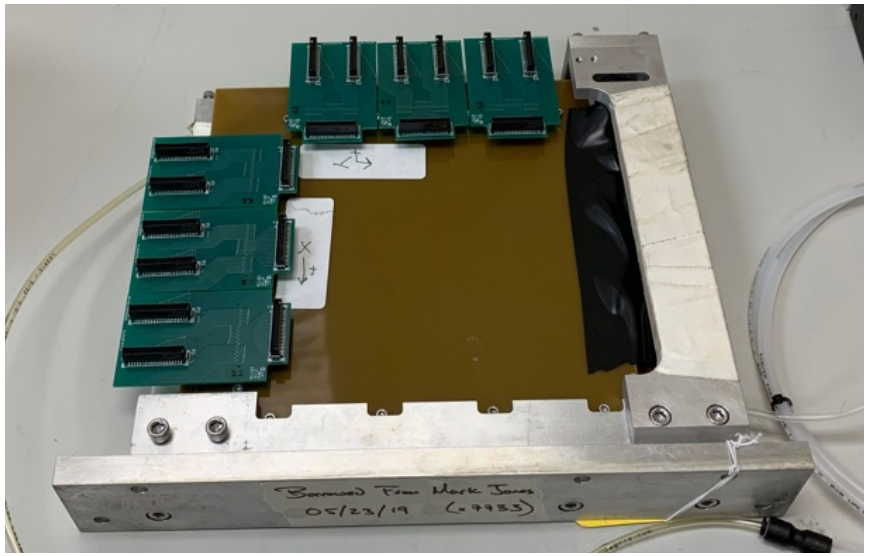
- Data is continuously streaming from the FECs to the T-RORC.
- Most of the 9 Gb/s data sent from each FEC to the T-RORC (45 Gb/s total) consists of sync packets that serve to keep the serial links from the SAMPAs active when there is no hit data to send.
- With the current firmware the T-RORC transmits all of this data directly to PC memory.
- Trigger zero suppresses - T-RORC captures a time window after the trigger.
- Plan to modify the firmware of the T-RORC to remove need for this trigger.
  - Then we can acquire data in a truly continuous fashion.

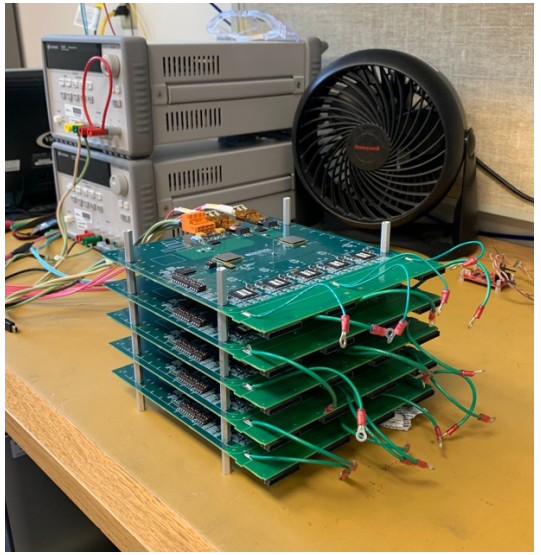


S1, S2	– plastic scintillators
GEM	– 1x, 1y plane 384 ch. Each
T-ROC	– ALICE /ATLAS Readout receiver
FEC	– ALICE Front End Card (JLAB version), 5 SAMPA chips = 160 channels

Test stand pictures







Top row left to right :

- Test stand on bench in INDRA lab
- Test stand closeup.
- Closeup of GEM before mount

Bottom right :

- Front End Card stack.

So where are we now?

- In the EIC streaming DAQ design key elements are:
  - A data source outputting on fiber.
  - A front end buffer with FPGA.
  - A high speed low latency network.
  - An online compute resource to buffer and process data.
  - A timing system.
- We are testing streaming readout of 250 MHz fADCs (Ben).
- We are testing a GEM detector readout in the INDRA lab. (Ed and Eric).
- We are testing clock distribution over network. (William)
- In the INDRA lab at JLab we have a test stand using
  - a VETROC TDC to provide a Front End data source
  - A Xilinx FPGA KCU1500 PCI board as a front end buffer and preprocessing device.
  - A Linux PC, with 100 Gbit/s network link as the online buffer/back end processing node. (William)