DAQ Developments for FPGA-Based Hardware Systems at JLab

Streaming Readout VIII – Apr 28-30, 2020

Virtual Meeting

David Abbott

Ben Raydo

FEDAQ Group

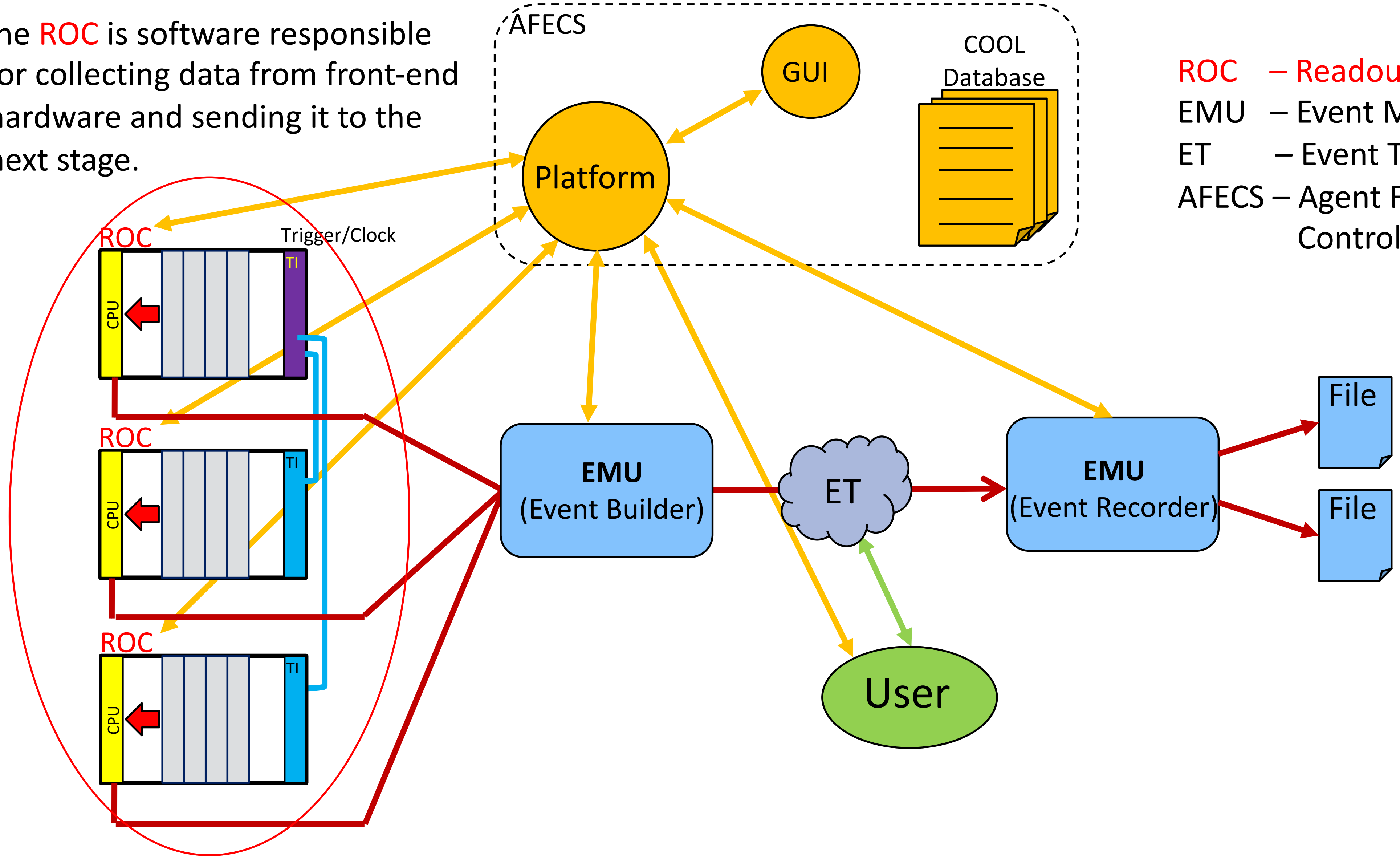
Jefferson Lab – Physics Division

Data Acquisition at JLab

- At Jlab we have 4 Experimental Halls, all running with different detectors - and physics priorities.
  - Future approved experiments are prepping for their turn on the floor.
  - Of course most have increased demands on the DAQ.
- Experiments are increasingly reliant custom electronics to interface detectors and digitize signals.
  - ASICs and FPGAs are becoming the norm (and the future) for the front-end.
- Interest in the Streaming Model for DAQ is growing quickly.
  - Proof of principle tests have been successful in several Halls.
- Our goal is to support both the traditional Triggered model along with the Streaming model within one integrated DAQ framework.
  - Leverage existing hardware to implement streaming
  - Add support for new electronics
  - Try to make it as seamless and user friendly as possible

The CODA Data Acquisition Toolkit

The ROC is software responsible for collecting data from front-end hardware and sending it to the next stage.



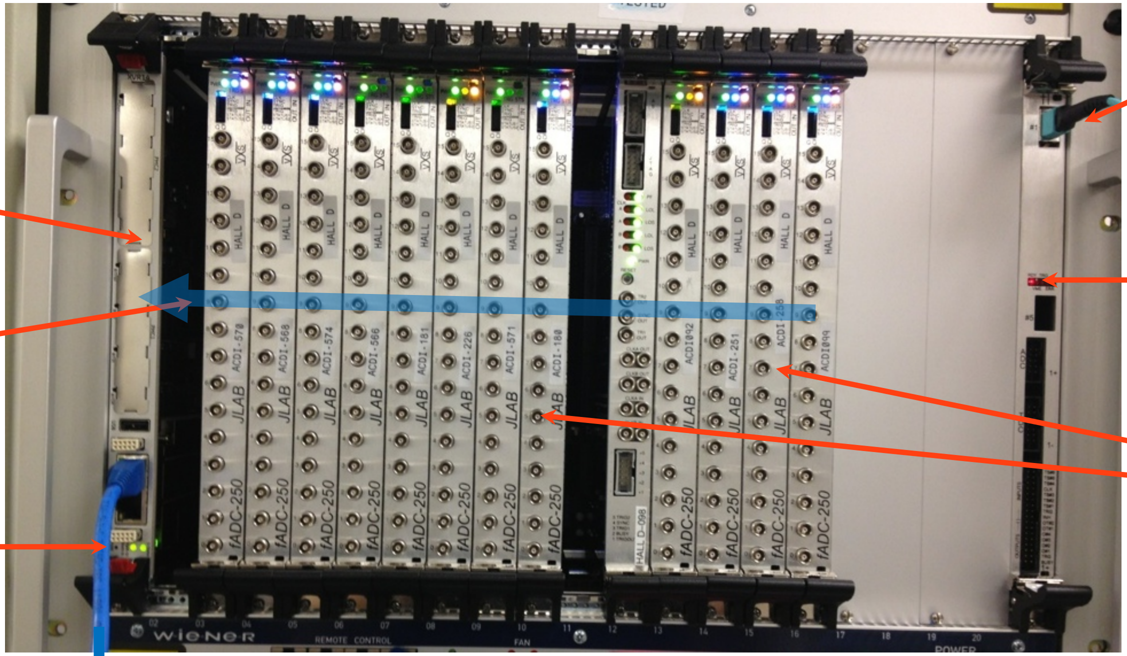
ROC – Readout Controller

EMU – Event Management Unit

ET – Event Transport (Shared memory)

AFECS – Agent Framework Experiment Control System

So long to the "Good Old Days"



Intel CPU ReadOut Controller or **ROC**

Readout via VME at up to 200MB/s

Output via network link (1 Gbps or ~110 MB/s)

MTP Fiber coming from the Trigger/Clock Distribution System

Trigger Interface board (TI)

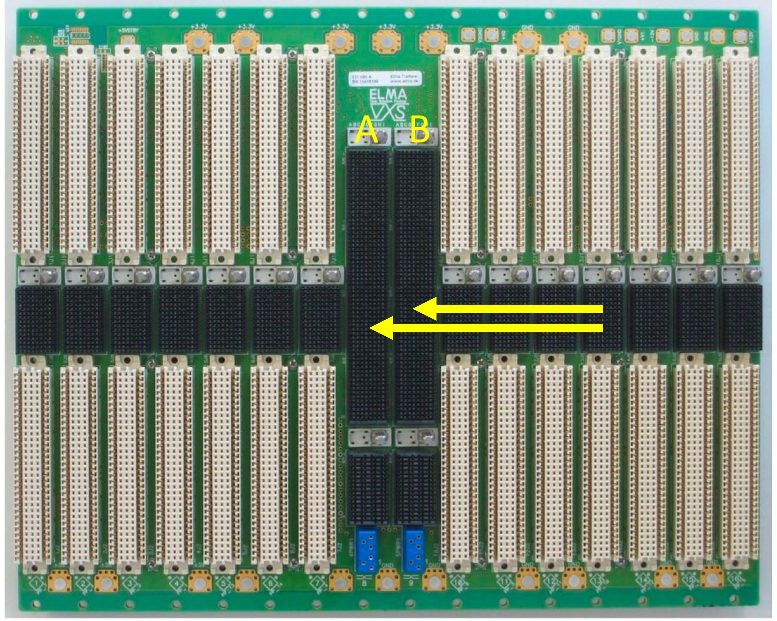
Digitizing payload boards (FADC)

One 250MHz FADC board can generate - 48Gb/s for all 16 channels
That's 768 Gb/s for a full crate (16 payload boards).

Nobody wants to deal with ALL that data. But for many experiments cutting things down to less than 1Gbps off the crate is just not not going to work any more.

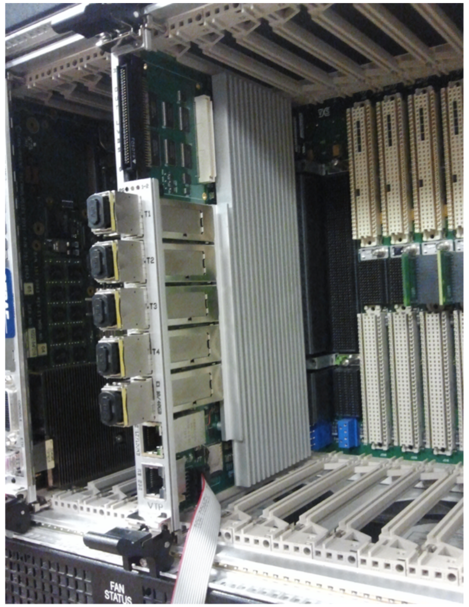
Not for Triggered readout and definitely not for Streaming

VXS to the Rescue



- JLab standardized on this technology for the 12GeV Upgrade
- Originally used for the L1 trigger data path
- Dual Star – switched serial backplane (along with VME)
- Up to 20Gb (4 lanes) from each Payload to the 2 Switch slots (A, B)
- Easy distribution of Trigger or low jitter clock to all modules in the crate.

Just need something in one of the Switch slots to coordinate it all…



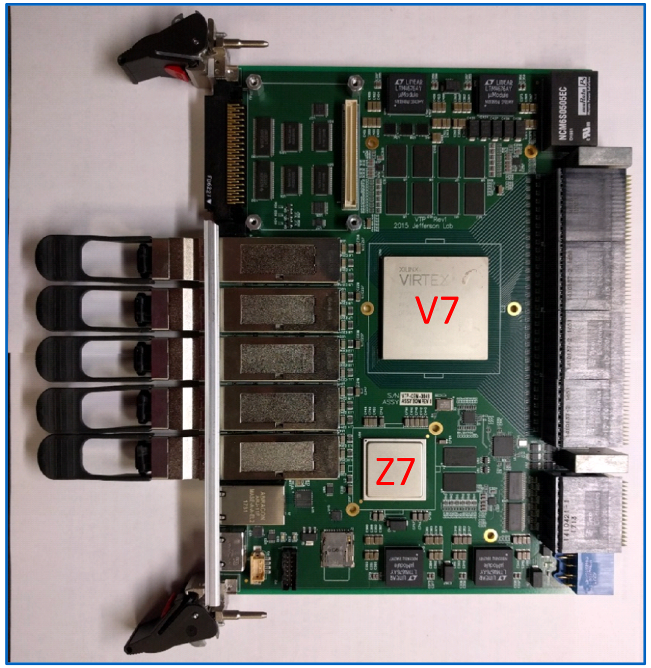
VXS Trigger Processor (VTP)

- Relieve the ROC of all of the “heavy lifting” tasks and implement them in the FPGAs.
- Triggered or Streaming readout from All payload modules in parallel
- This requires the payload modules to have some intelligence/programmability and serial link capability (e.g. FPGA-based).
- The Software ROC now is primarily responsible only for Configure, Control and Monitoring the VTP-Based DAQ

JLAB – VTP Board

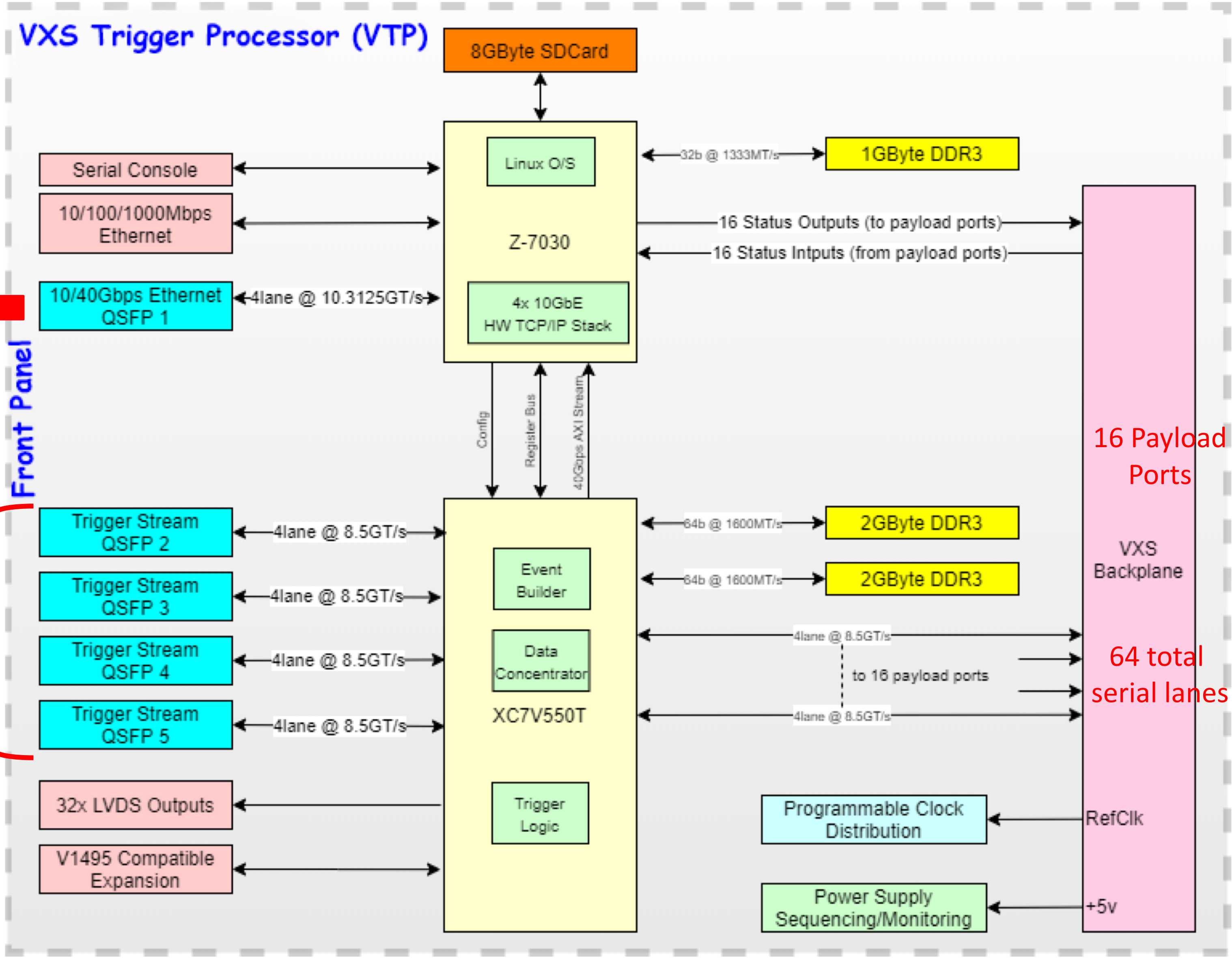
Linux OS on the Zync-7030 SoC
(2-core ARM 7L , 1GB DDR3)
10/40Gbps Ethernet option

Xilinx Virtex 7 FPGA
Serial Lanes from both the VXS backplane and the Front panel
4GB DDR3 RAM



Four 10Gbps Ethernet

16 total lanes for external serial links



VXS Trigger Processor (VTP)

Front Panel

8GByte SDCard

Serial Console

10/100/1000Mbps Ethernet

10/40Gbps Ethernet QSFP 1

4lane @ 10.3125GT/s

Z-7030

Linux O/S

4x 10GbE HW TCP/IP Stack

32b @ 1333MT/s

1GByte DDR3

16 Status Outputs (to payload ports)

16 Status Intputs (from payload ports)

Config

Register Bus

40Gbps AXI Stream

Trigger Stream QSFP 2

Trigger Stream QSFP 3

Trigger Stream QSFP 4

Trigger Stream QSFP 5

4lane @ 8.5GT/s

4lane @ 8.5GT/s

4lane @ 8.5GT/s

4lane @ 8.5GT/s

XC7V550T

Event Builder

Data Concentrator

Trigger Logic

64b @ 1600MT/s

2GByte DDR3

64b @ 1600MT/s

2GByte DDR3

4lane @ 8.5GT/s

to 16 payload ports

4lane @ 8.5GT/s

32x LVDS Outputs

V1495 Compatible Expansion

Programmable Clock Distribution

Power Supply Sequencing/Monitoring

VXS Backplane

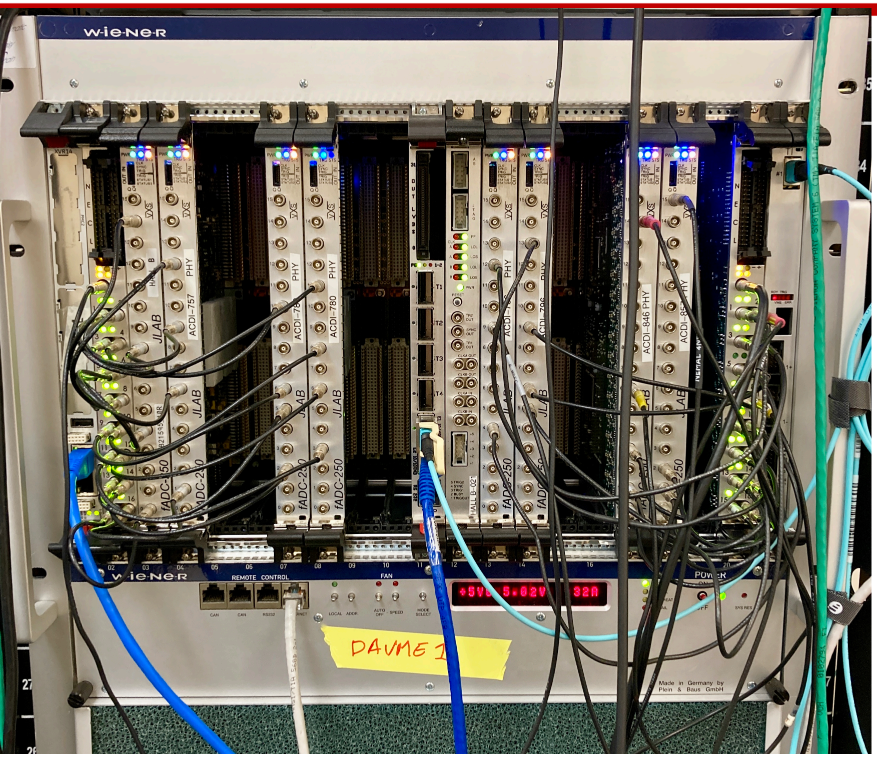
RefClk

+5v

16 Payload Ports

64 total serial lanes

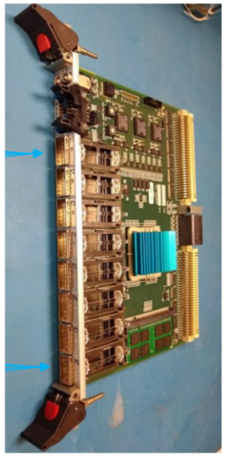
Flexible Platform



250 MHz FADC
16 Channels
12 bits

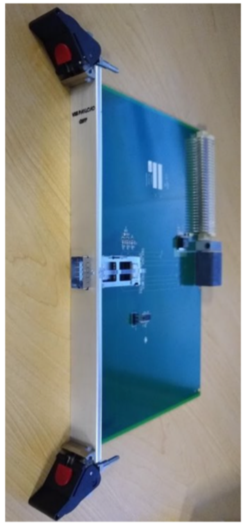
JLab workhorse in all current experiments

Used in Streaming testbed (above)



Subsystem Processor (SSP)

FPGA board
8 QSFP Inputs

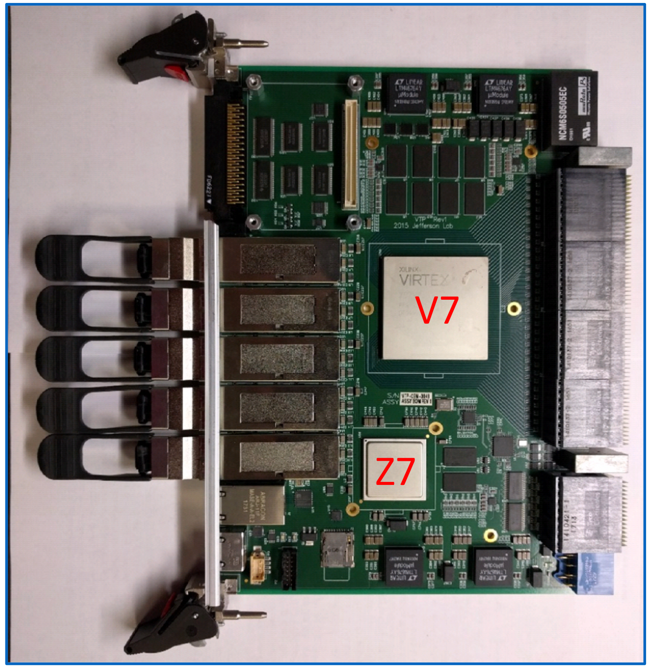


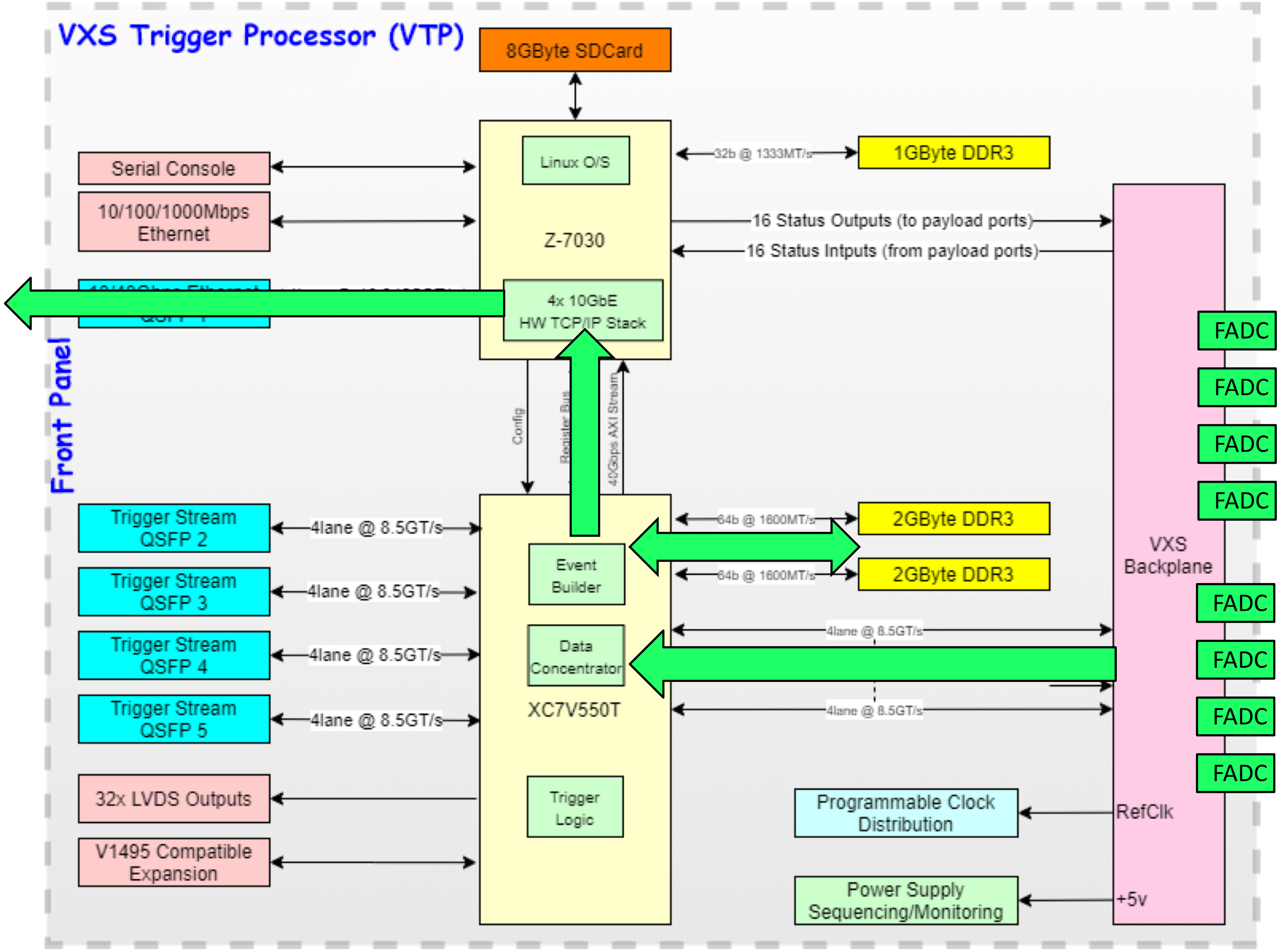
QSFP->VXS adapter card
Simple direct serial link access to the VTP for External custom electronics
(e.g. MPD module)

- In addition to supporting all the older VME electronics, we have developed a number of custom boards that take advantage of the VXS backplane.
- Streaming Model tests grew out of the original purpose of VXS for the trigger data path.
- More immediate needs of experiments, however, are for Triggered data readout via the VTP as well – rather than over VME.

JLAB – VTP Board

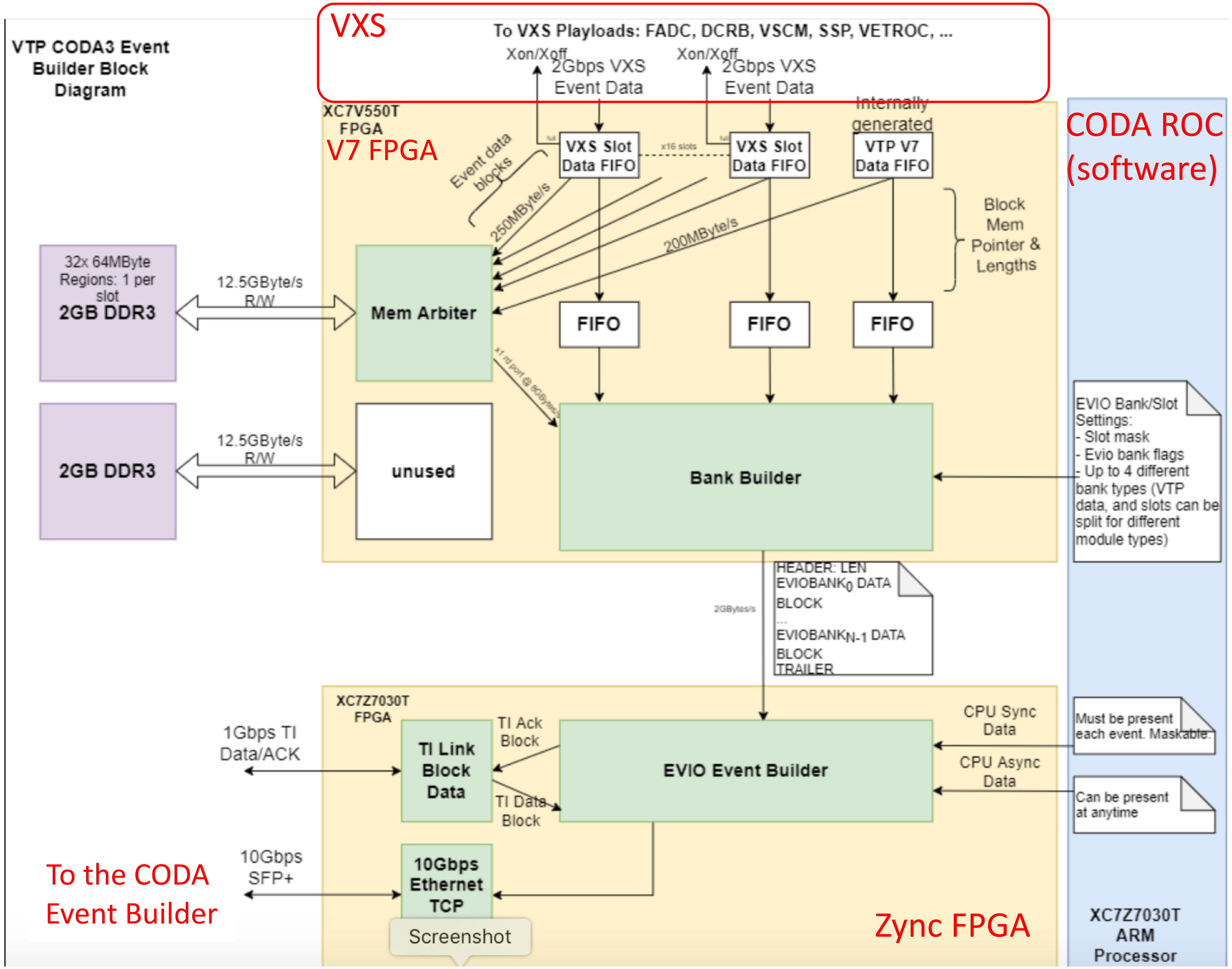
- Standard CODA ROC can run on the Zync ARM processor
- Implement firmware & driver libraries that allow ROC to control the FPGA event building and data flow.
- Both Streaming and Triggered models are very similar.





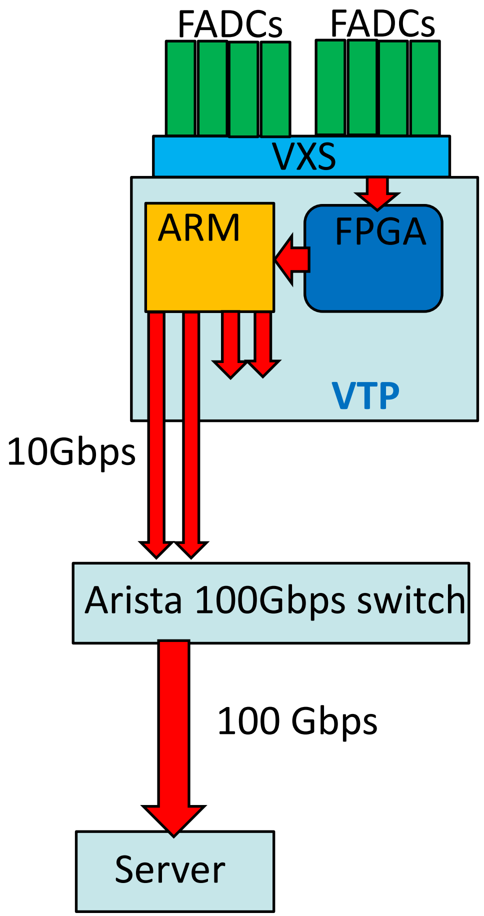
Jefferson Lab

The FPGA-Based ROC for the VTP



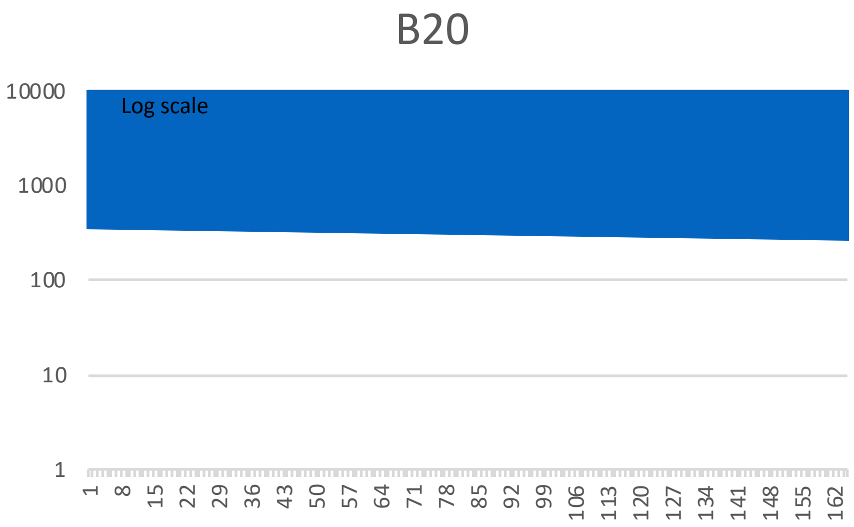
- This schematic shows the current FPGA ROC for Triggered readout, but much of the structure is the same for the Streaming firmware.
- The original Software ROC still allows the User full access to configure the readout data banks as needed before starting a run.
- Trigger and timestamp data are received via VXS from the TI module and acknowledges sent.
- Asynchronous data can be inserted into the data stream by the Software ROC during state transitions or on periodic “Sync” events
- As of this past week this all works. Testing and development continues…

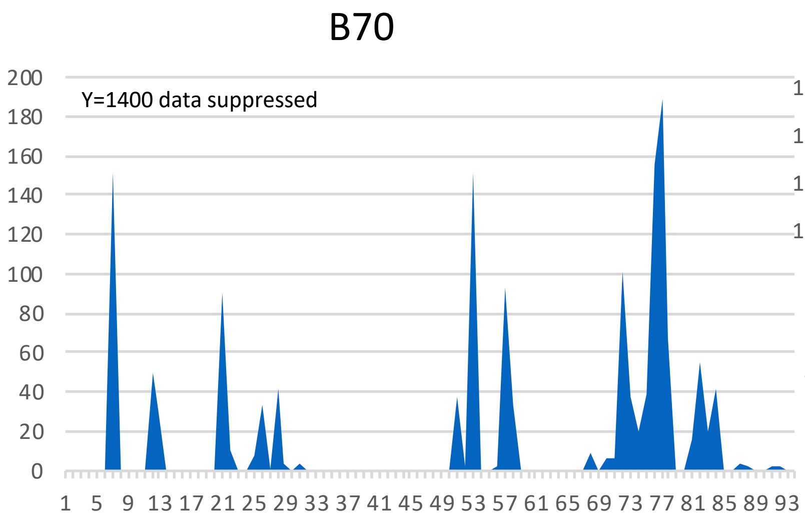
Performance of the VTP TCP Stack

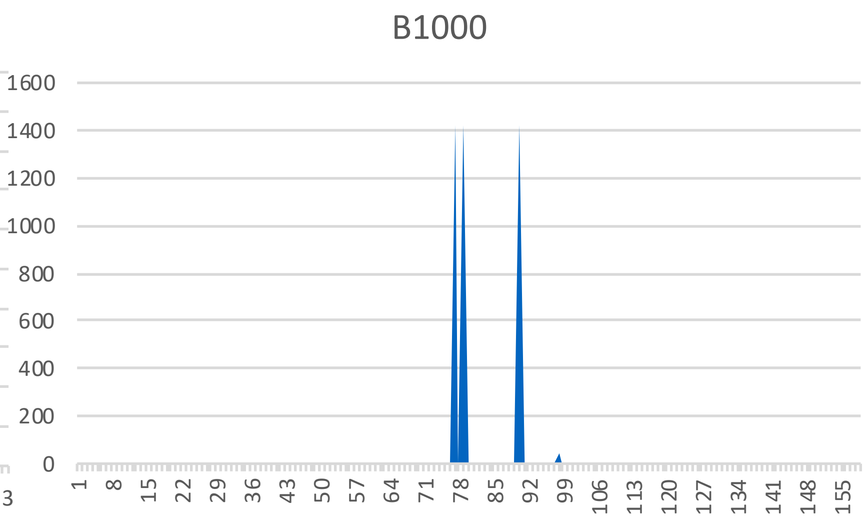


- For our Streaming tests we were able to make some performance measurements of the TCP stack used on the Zync SoC.
  - Up to 4 10Gbit ethernet links supported
  - Both Client and Server modes of operation
  - 64 kB TCP Send buffers per socket.
  - Buffer Frames in DDR3 memory to allow for variable network latency or retransmits. (between 20-1000 frames)
- Run two independent streams from the VXS crate to the same Server.
  - 4 FADC boards for each stream.
  - 1 MHz pulser data – corresponds to about 620MB/s per stream (around 50% bandwidth)
  - 15 kHz Data Frame rate (~65μs time slices, ~42kB/Frame)
  - Server connected with a single 100Gbps ethernet link.

Frame Drops





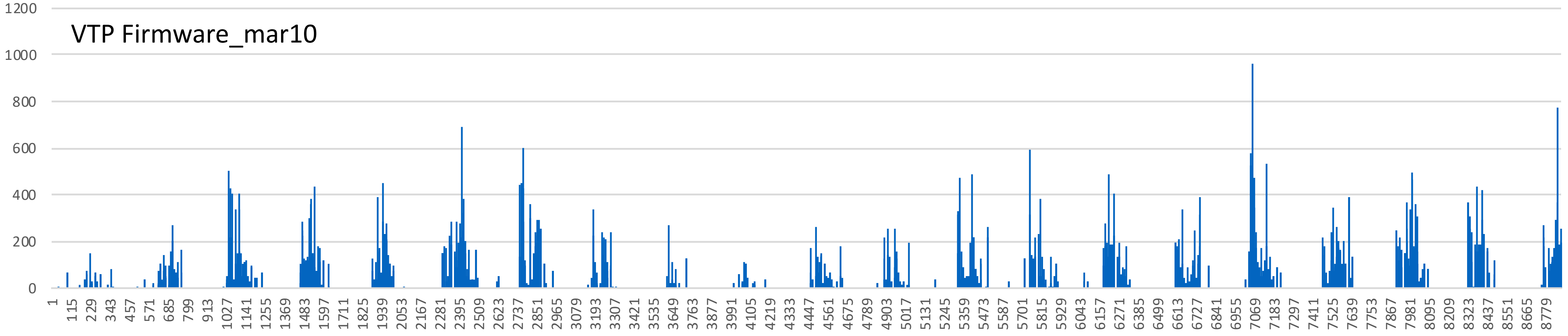


- For a buffer level of 20 we see regular frame drops in the VTP
  - Still only at < 0.5%
- At buffer level 70 - frame drops become more intermittent
- At a buffer level of 1000 only some very few large (but very short in time) “hiccups” still remain

To see the longer term trends we let streams run for several hours

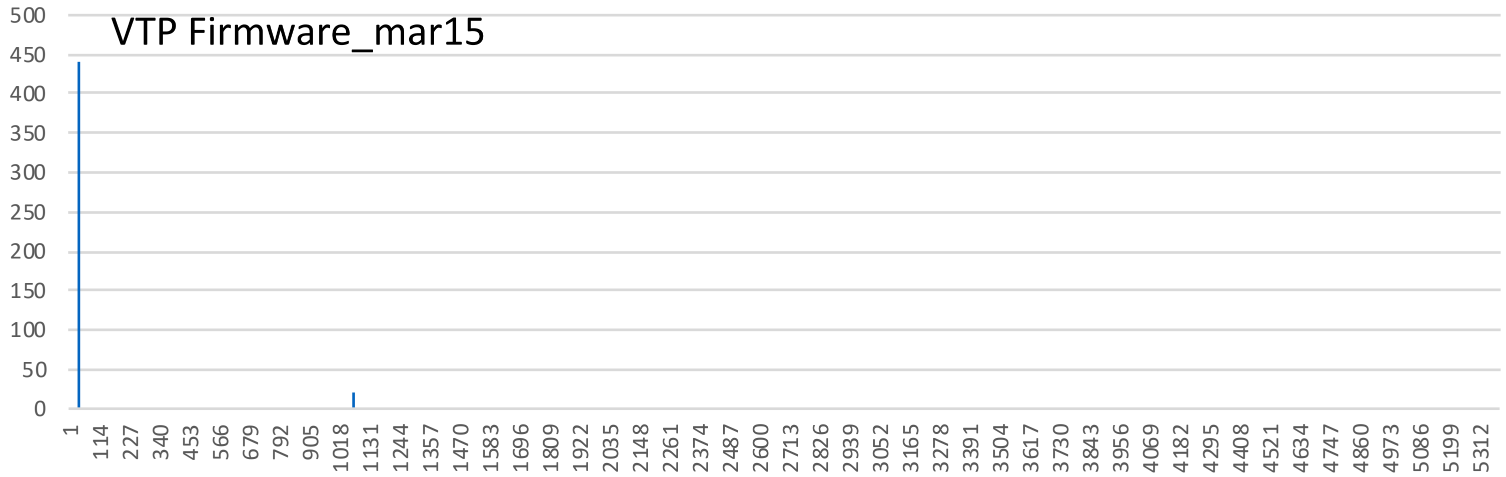
TCP Performance Tests from VTP

Lost frame rates in Hz. Stream1



VTP Firmware_mar10

Lost frame rates in Hz (TCP Tx window changed to 128kB)



VTP Firmware_mar15

Streaming – Further testing

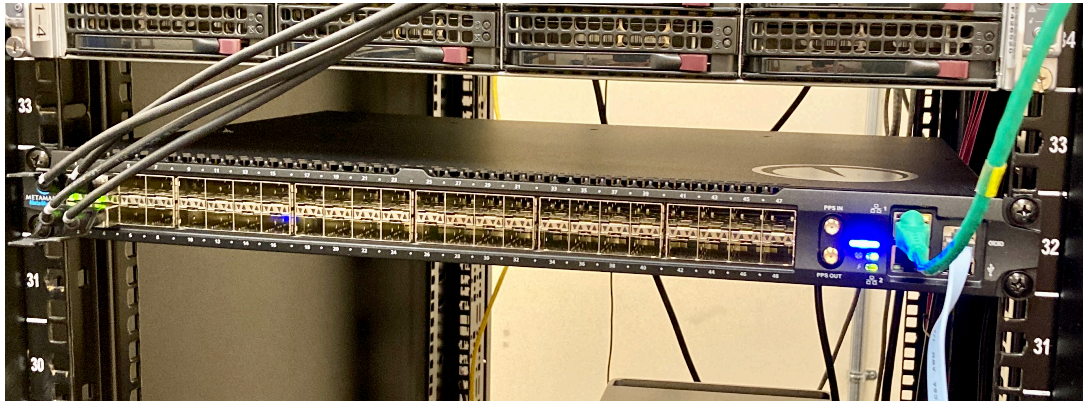
- Testing with the current TCP Stack on the VTP is maxing out at between 7-8 Gbps. We can probably do better than this but it may cost more in resources (i.e. buffering).

- From an FPGA perspective the most efficient way to transport Streams over ethernet would be using UDP (and Jumbo frames). This is something that we will investigate.
  - Not practical however for the Triggered Model

- The VTP by it's very nature is a Stream Aggregation point. We still need to develop some standard methods for throwing data away or inhibiting all streams synchronously at their source.
  - We currently use a distributed "Sync" signal from the Trigger/Clock system to start all streams at the same timestamp at the beginning of a run.

- Current streaming tests have guaranteed sufficient bandwidth between FADCs and the VTP
  - Only sending pulse sums and times.
  - How do we deal with FADC full waveforms or other payload modules that may generate too much data?
  - Drop entire time slices? Truncate certain data streams?

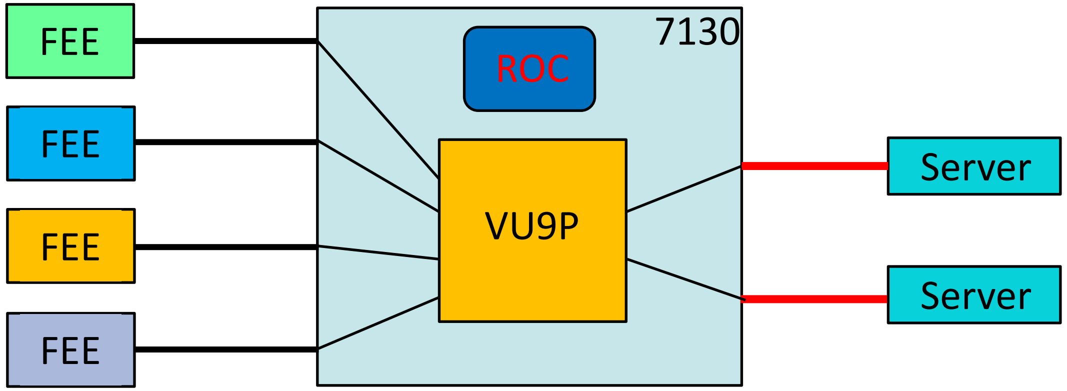
Arista 7130 – FPGA-Based "switch"



- Virtex Ultrascale+ VU9P-3 FPGA
- 48 SFP+ Ports can be mapped to 60 application ports directly on the FPGA
- 32 GB (4x8GB) DDR4-2200 RAM
- JTAG and Gen 2 PCIe x8 access to FPGA by on board Intel x86 CPU running Linux.
- Available Vendor application support including - port aggregation and high resolution timestamps.
- Development kits for full access to FPGA resources and custom user applications.
- All ports can support standard 10Gb ethernet or custom serial link protocols.

- Next commercial hardware option we will be working with. It is kind of a VTP on steroids.

- Potentially useful for aggregating serial links from different front-end detector electronics and presenting them as standard ethernet streams for back-end processing.



Summary

- The VXS platform provides a reasonably long term solution to support the next generation of experiments needing higher performance front-end triggered readout as well as streaming support.

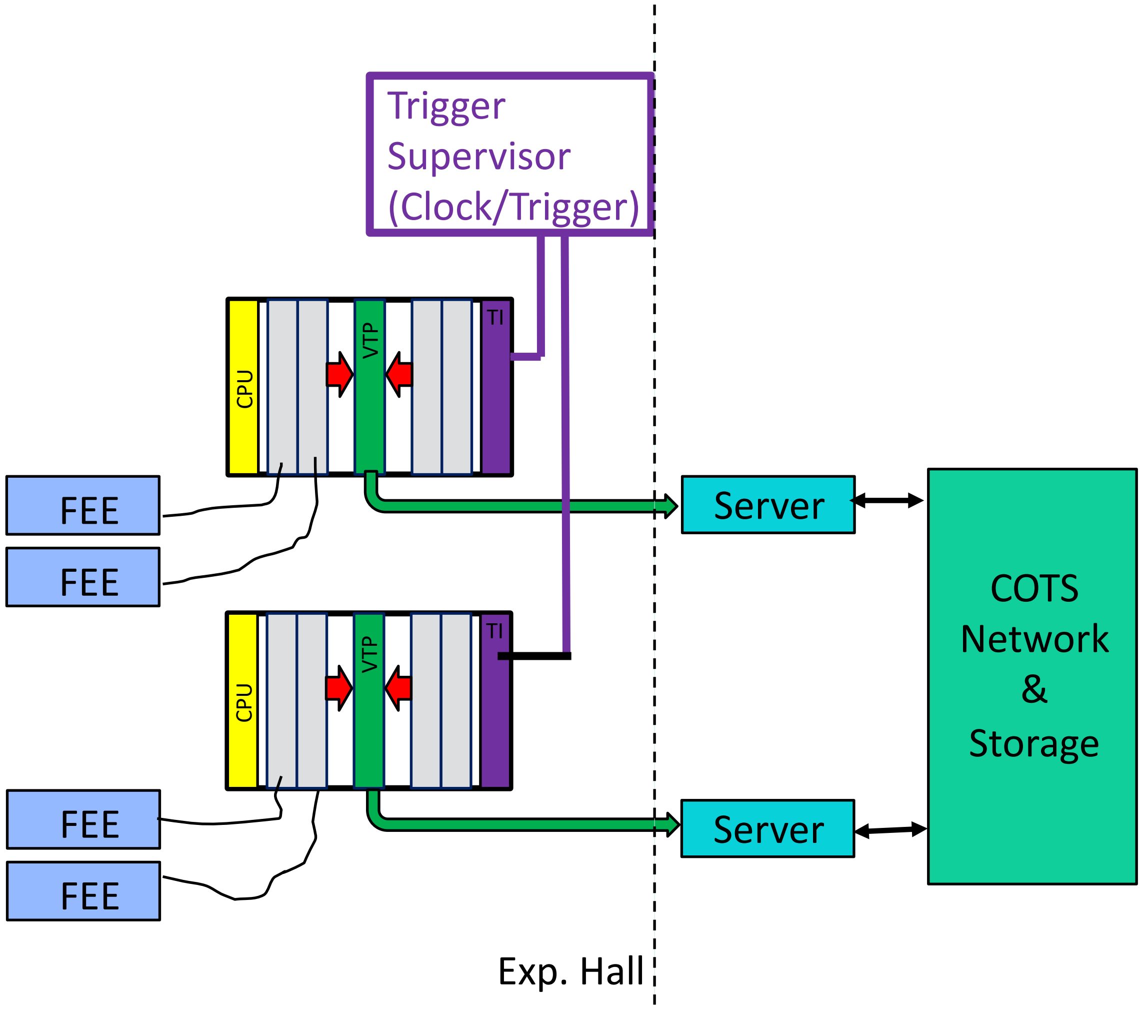
- Transition from the CODA DAQ system’s traditional software-based Readout Controller (ROC) to a “hybrid” hardware accelerated application is currently underway. Much of the work ahead will involve making it robust against whatever the front-end electronics may throw at it.

- The nature of the varied types of experiments and detectors here at JLab motivates our small electronics and computing support groups to look for both commercial solutions as well as standardized software and firmware to help manage all the data acquisition challenges.

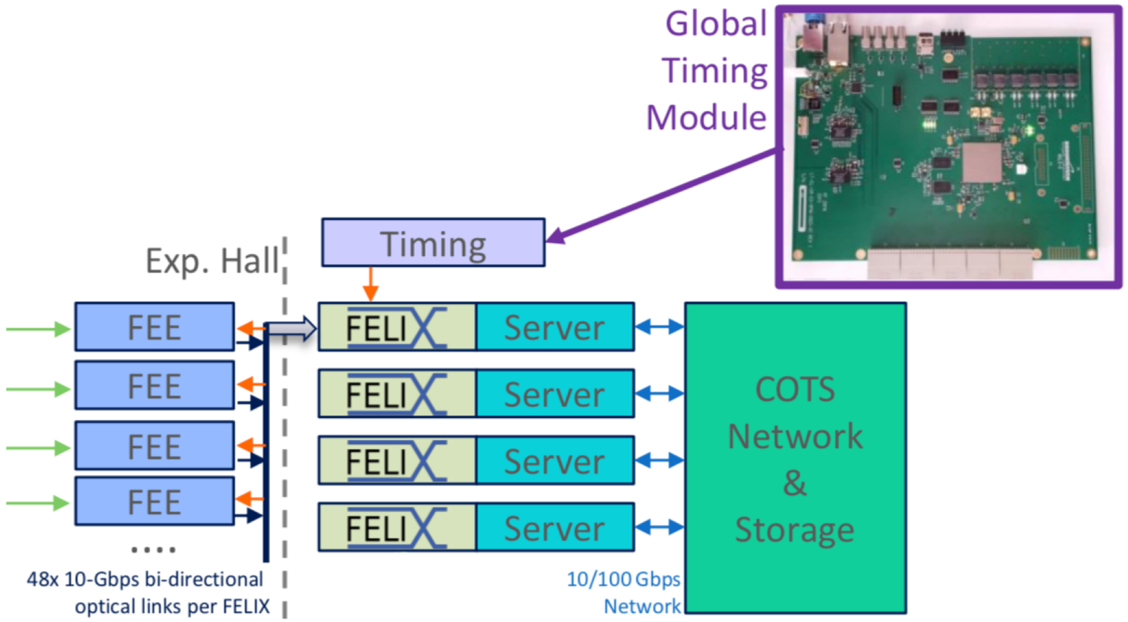
Backup Slides

We are not so different, you and I…

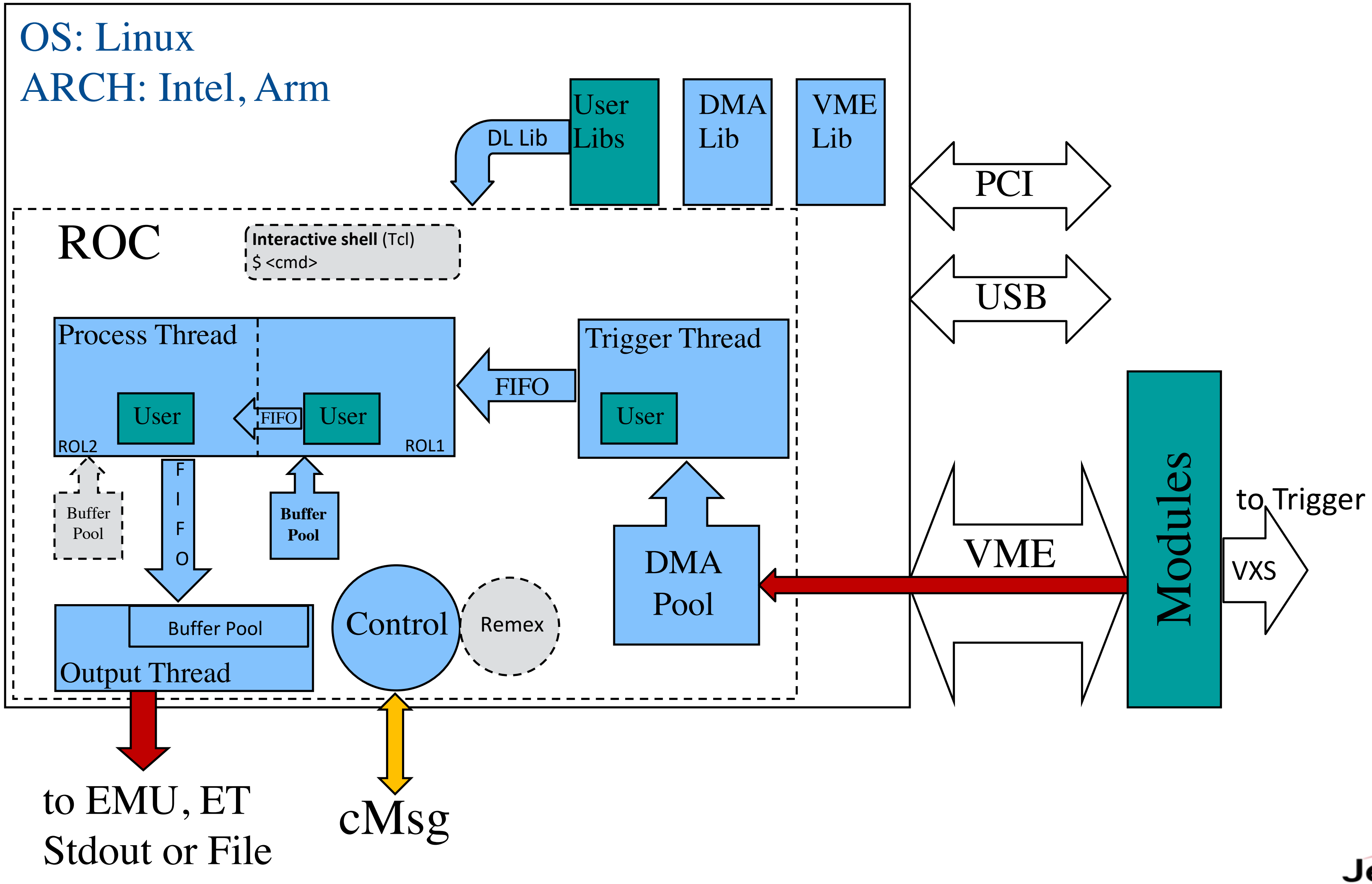
JLAB



sPhenix

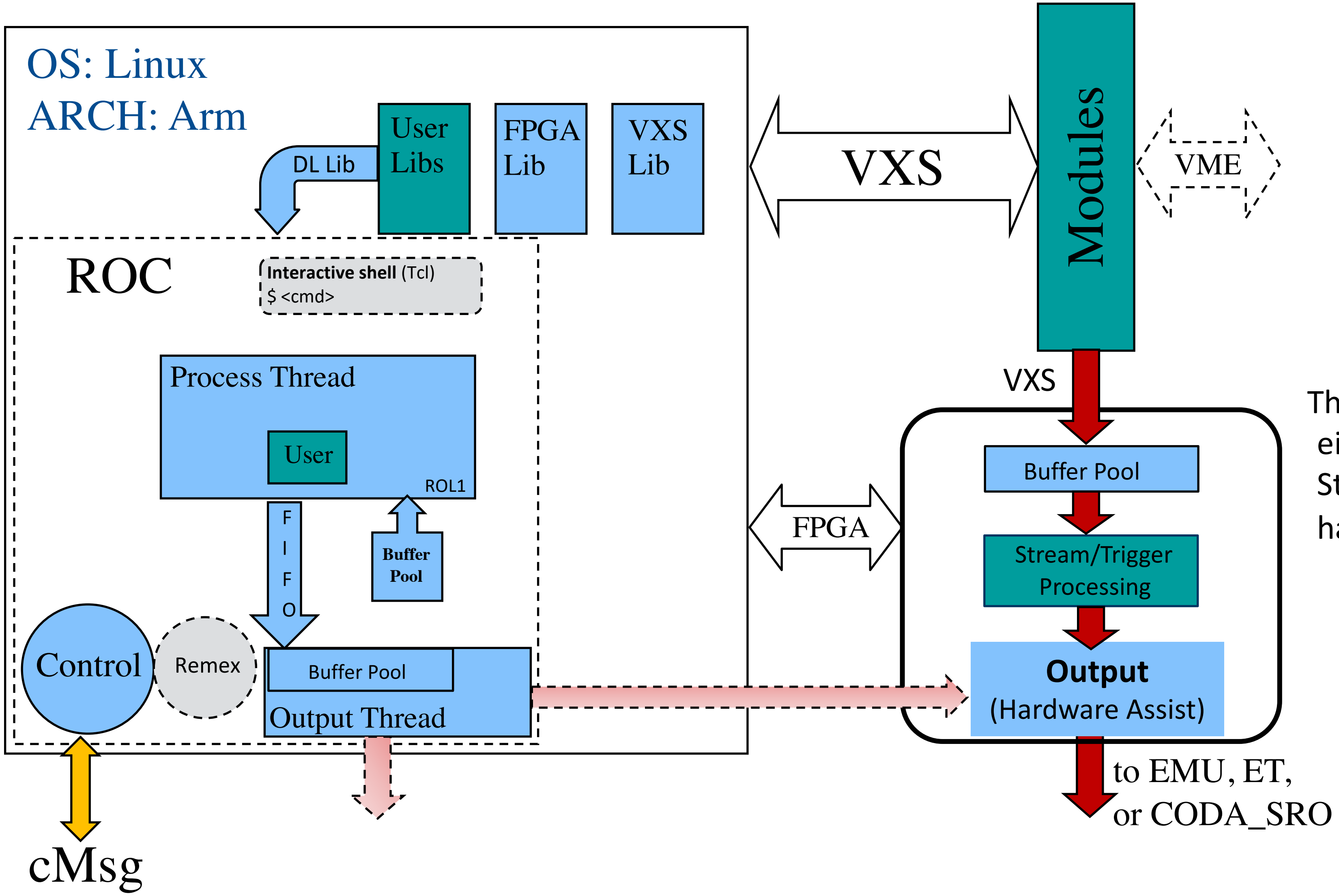


CODA 3 – Current Front End

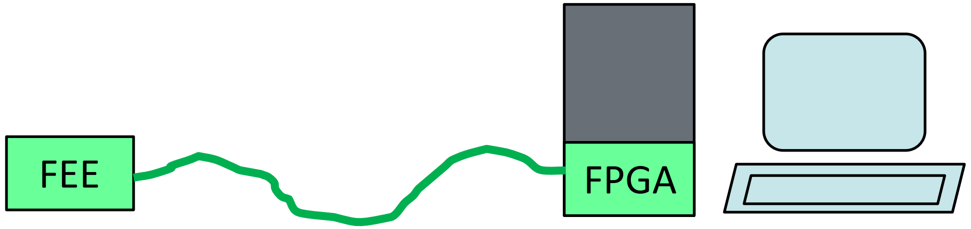


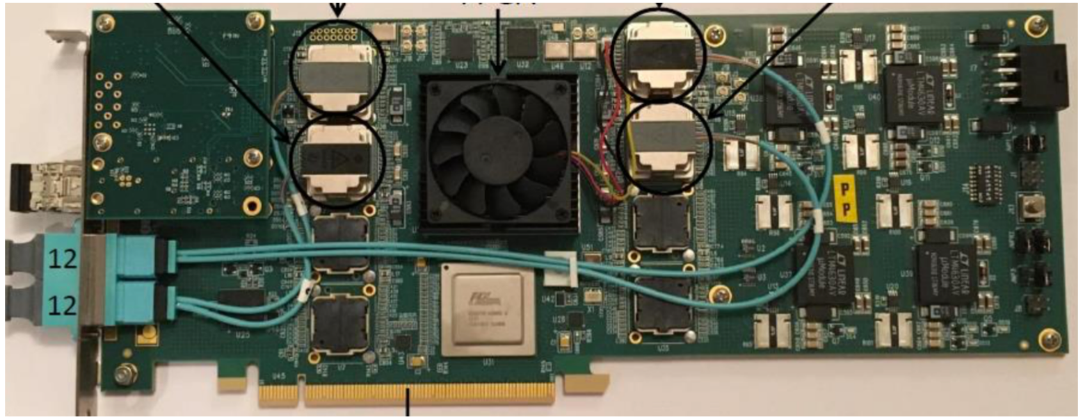
Jefferson Lab

CODA – SRO Front End (on VTP only)



The primary Data either Triggered or Streamed stays in hardware


FEE
FPGA


12
12