Streaming Readout Development for CODA @ JLab

23rd IEEE NPSS Real Time Conference

August 1-5, 2022

David Abbott

Ben Raydo

FEDAQ Group

Jefferson Lab – Physics Division

Jefferson Lab

Data Acquisition at Jefferson Lab

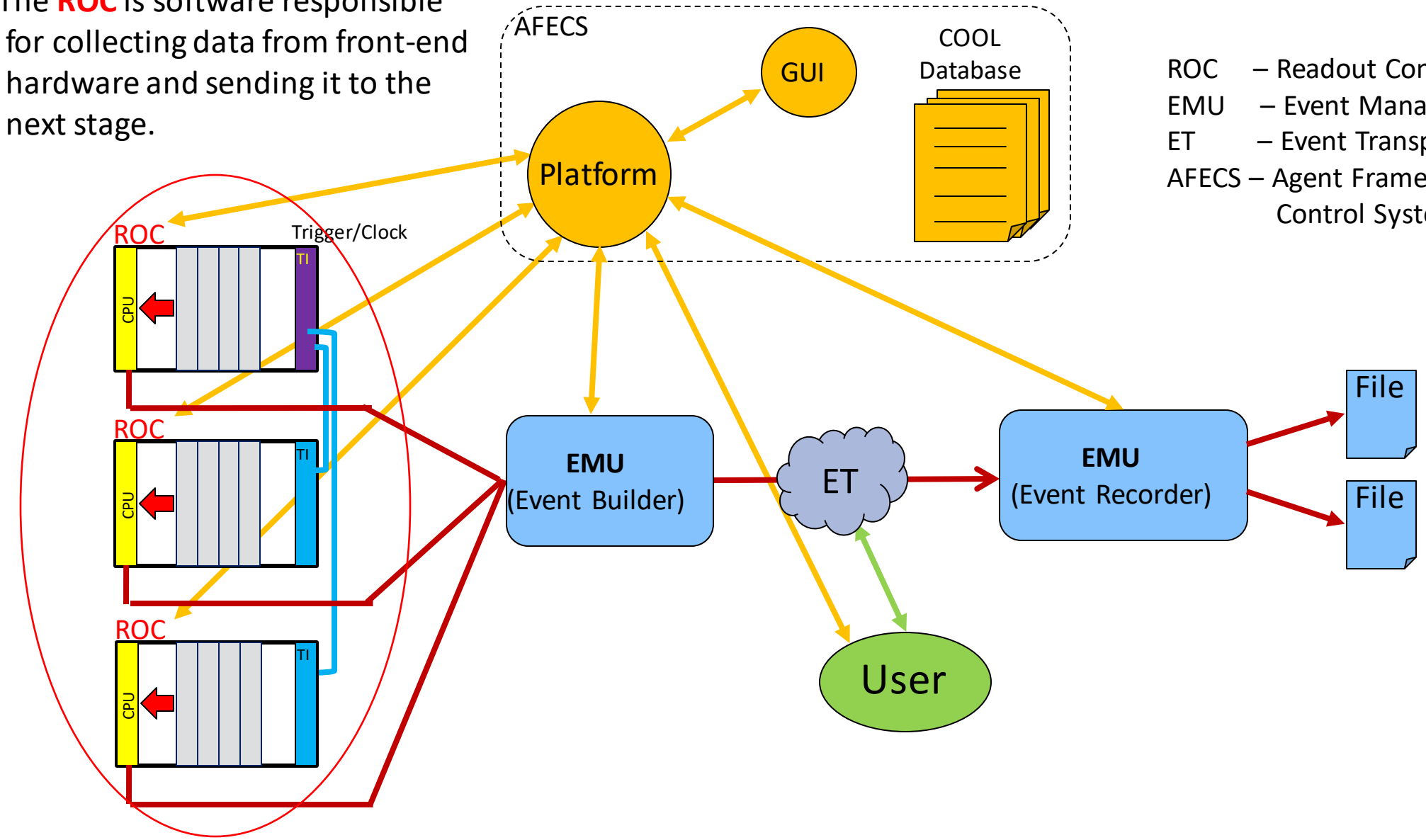
- At JLab we have 4 Experimental Halls, all running with different detectors - and physics priorities.
  - Of course, all are having increased demands for the DAQ.

- 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.
  - But older hardware is still relevant and useful (particularly for starving budgets)

- 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

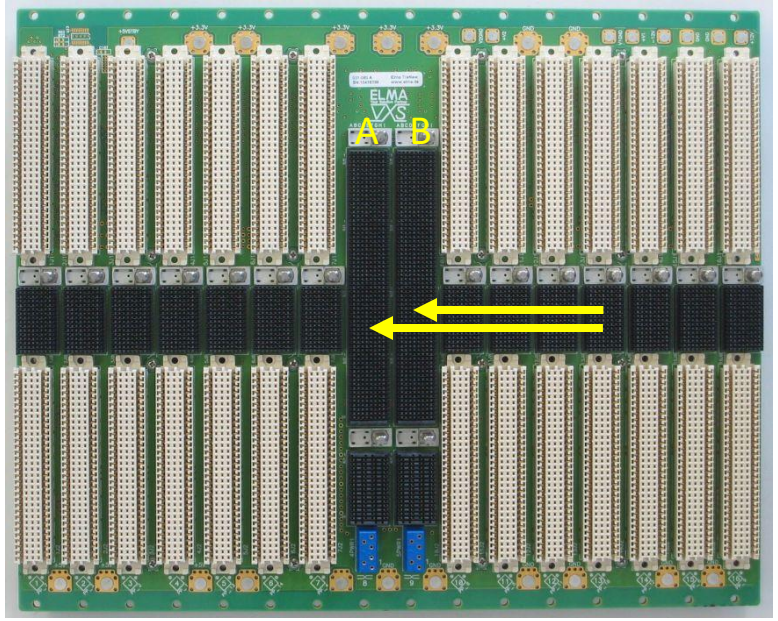
The **ROC** is software responsible for collecting data from front-end hardware and sending it to the next stage.



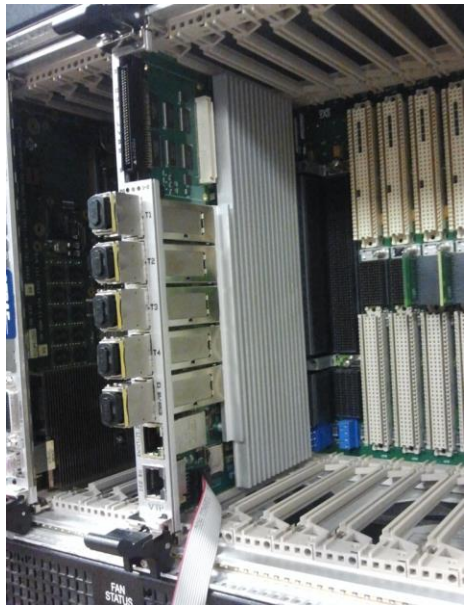
ROC – Readout Controller
EMU – Event Management Unit
ET – Event Transport (Shared memory)
AFECS – Agent Framework Experiment Control System

Jefferson Lab

VXS Standard (VITA 41)



- JLab standardized on this technology for the 12GeV Upgrade
  - Originally used for the L1 trigger data path
- Dual Star – switched serial backplane (along with original VME)
- Up to 20Gb (4 lanes) from each Payload to the 2 Switch slots (A, B)
- Up to 18 Payload slots are available
- Easy distribution of Trigger, Sync and low jitter clock to all modules in the crate.



VXS Trigger Processor (VTP)

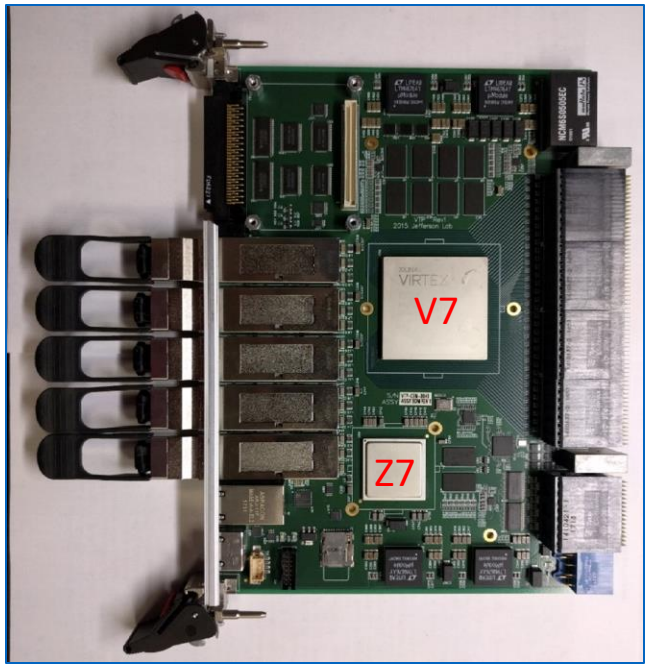
- Relieve the ROC of all the "Readout" tasks and implement them in the FPGAs.
- Triggered or Streaming readout from ALL payload modules in parallel
- In genral, the payload modules should have some intelligence/programmability and serial link capability (e.g. FPGA-based).
- The Software ROC now is primarily responsible only for Configure, Control and Monitoring the **VTP**-Based DAQ.

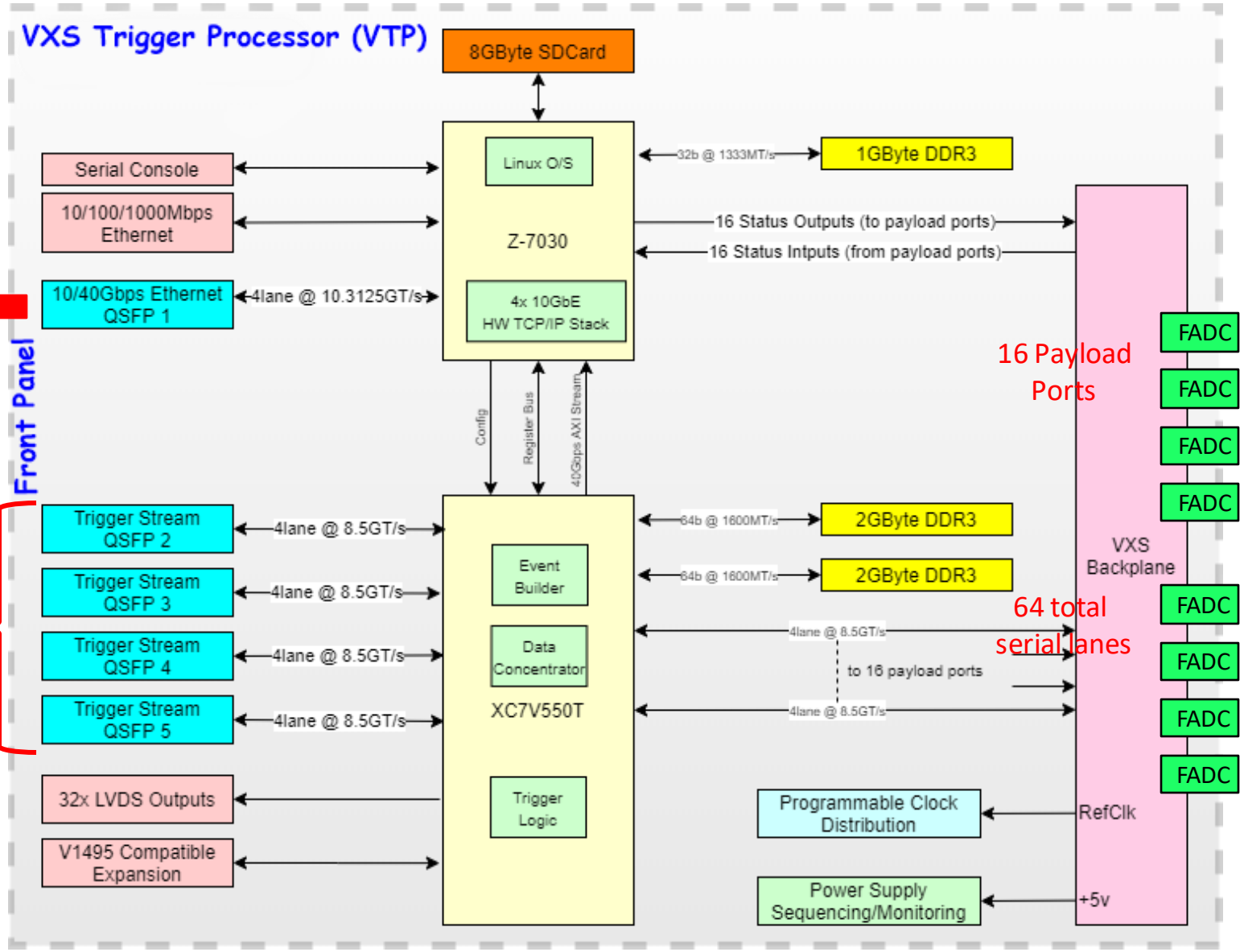
Jefferson Lab

JLAB – VTP Board

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

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

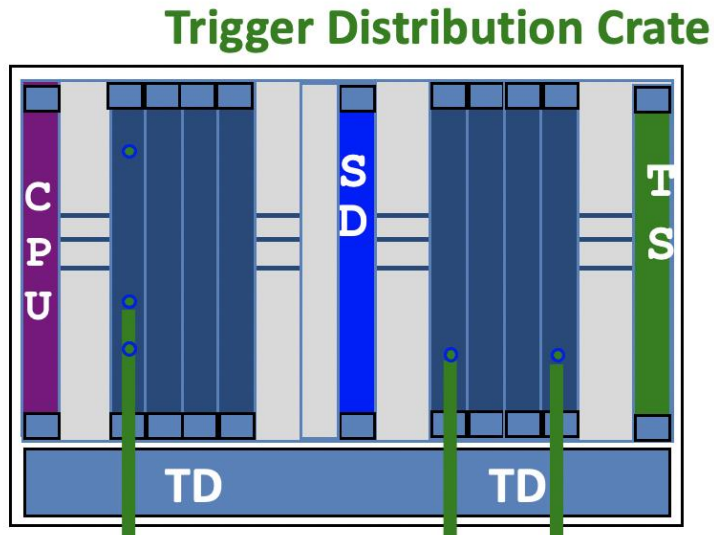




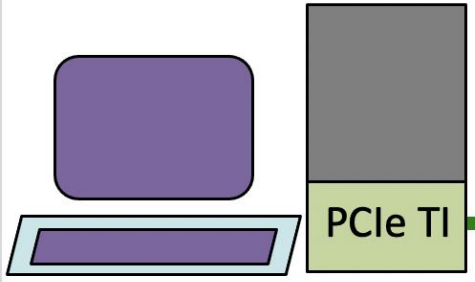
Jefferson Lab

JLAB Clock and Trigger Distribution System

TS – Trigger Supervisor (VME/VXS)
SD – Signal Distribution Board (VXS)
TD – Trigger Distribution (VME/VXS)
TI – Trigger Interface (comes in several flavors)



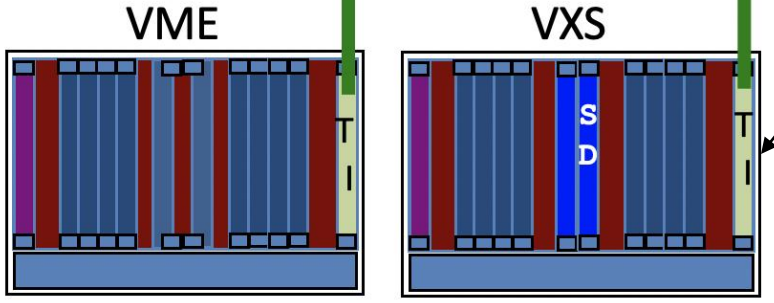
For large DAQ systems with many front-ends (ROCs)



CTS

(TRIGGER,CLOCK, SYNC) (BUSY)

Front-End ROCs:
(up to 127)



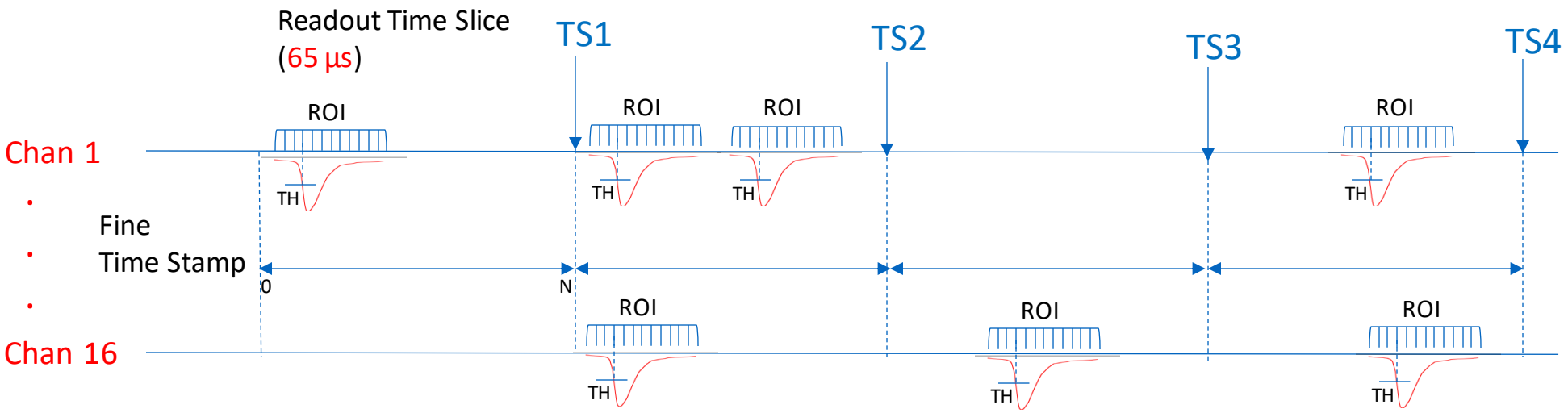
The TI board can be used as a TS for small systems
(up to 9 front-ends)

Jefferson Lab

JLAB FADC – Streaming mode

A 250 MHz FADC generates a 12 bit sample every 4ns. That's 3 Gb/s for one channel. 16 channels is 48 Gb/s.
Currently, we identify a threshold crossing (hit) and integrate charge over a ROI and send only a sum and timestamp for each hit.

Available bandwidth will allow for 1 hit every 32ns from all channels.
A data frame (Time Slice) for all available hits is generated in the VTP every 65µs



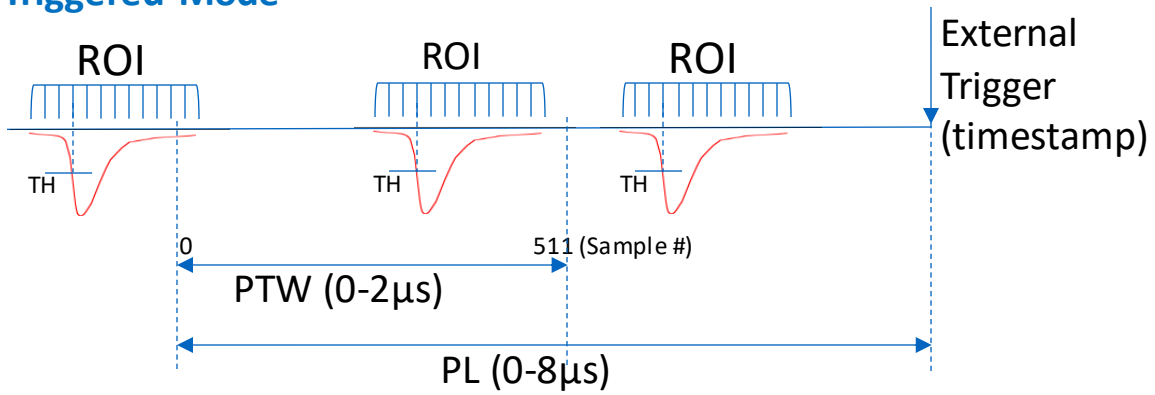
The next revision to the firmware will have an option for full ROI wave forms to be streamed, but this will allow possible dropped hits due to bandwidth limitations

The FADC can still simultaneously operate in triggered mode with an 8µs pipeline and 2µs readout window.

Jefferson Lab

FADCs - Triggered vs Streaming

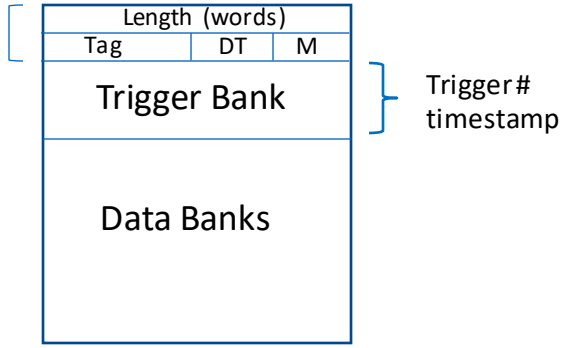
Triggered Mode



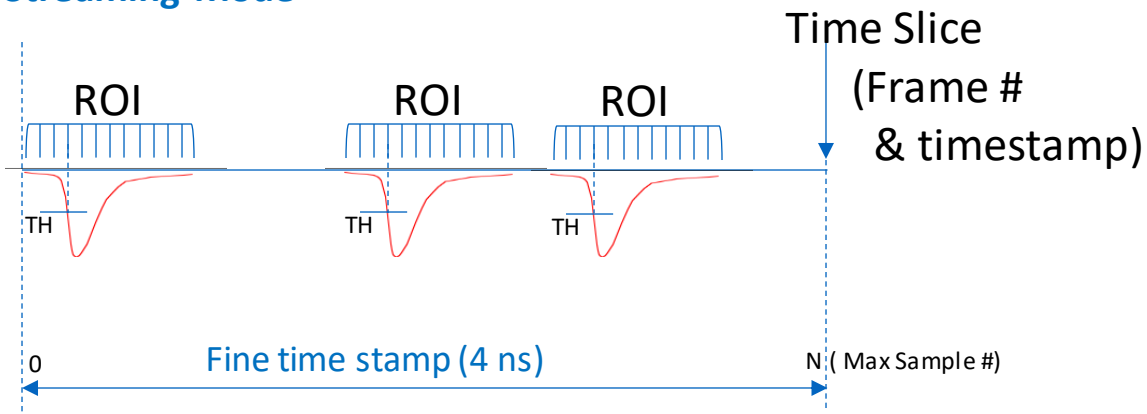
PL: Programmed Lookback
PTW: Time window

Data we get on a trigger:
- FADC waveform values for the ROI
- Threshold Sample # (hit time)
- Trigger absolute time stamp

ROC Data Format



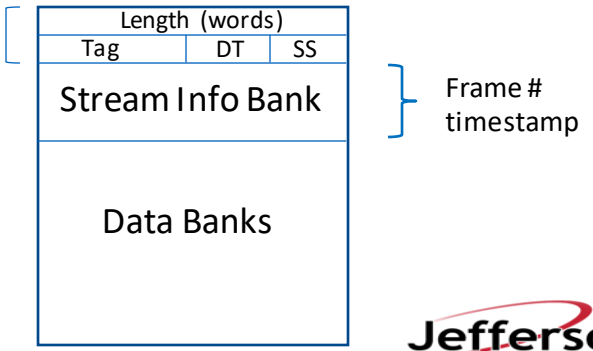
Streaming Mode



1 Frame = N Clocks (up to 16bits, currently 65536 ns)

Data we get for a Frame:
- Pedestal subtracted sums over an ROI for every hit over threshold
- Threshold sample # fine time stamp for each hit
- Frame # and absolute time stamp for the frame

ROC Data Format



Jefferson Lab

Firmware Compiler

Primary tools:

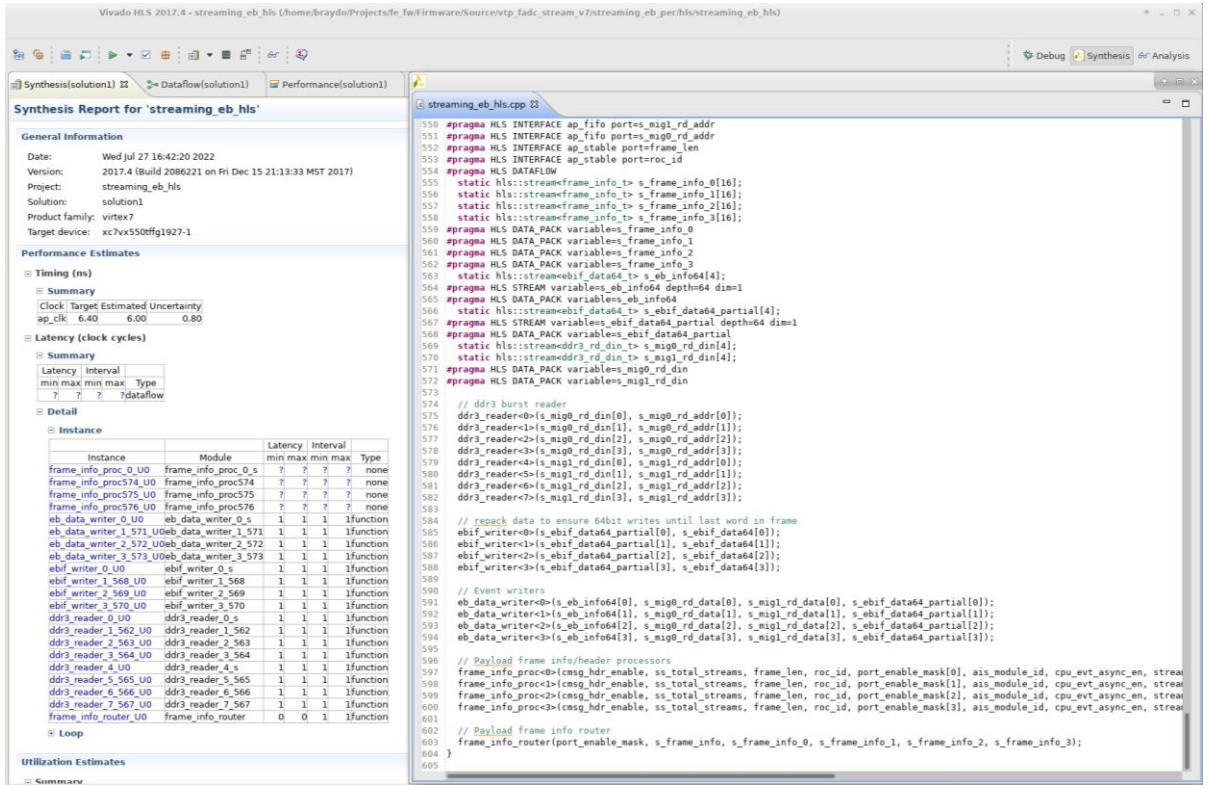
Vivado (VHDL & SystemVerilog)

- Needed for fast and/or minimized logic, specialized bus interfaces, specialized IP, multi-clock domain handling

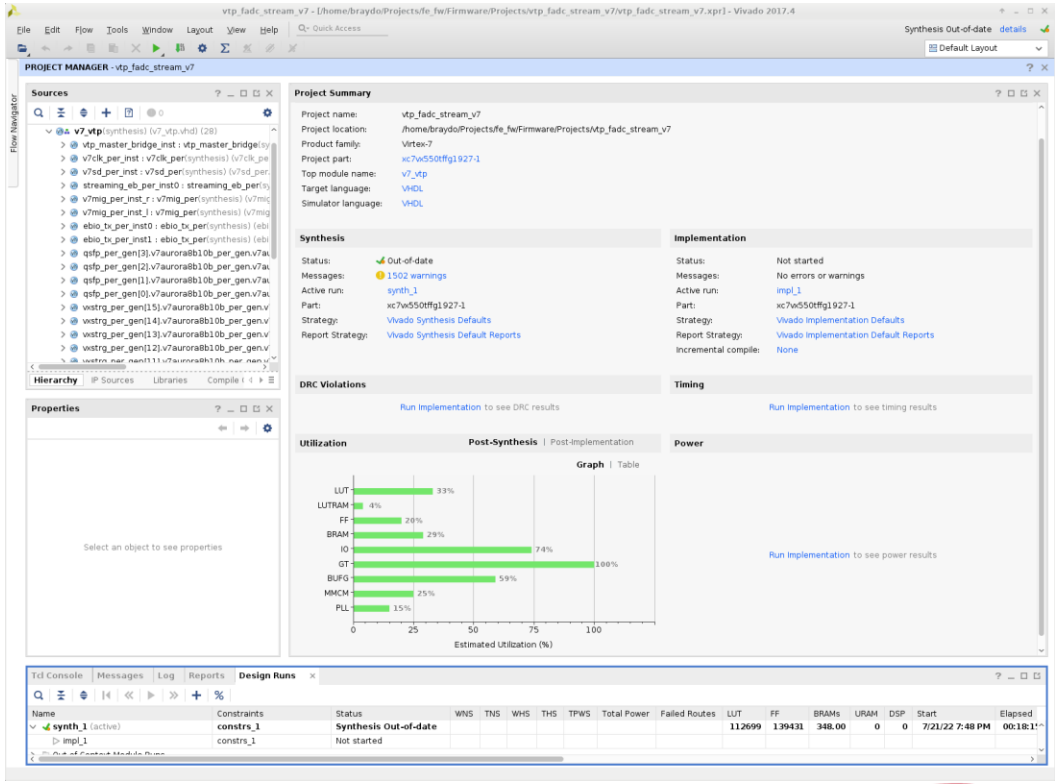
Vivado HLS (C/C++)

- Used for data processing, formatting, and aggregation
- No problem saturating 10G Ethernet ports

Vivado HLS Project - C++ w/special HLS directives to instruct parallelism:



Vivado Project - usual mix of VHDL and Verilog custom and Vendor IP:



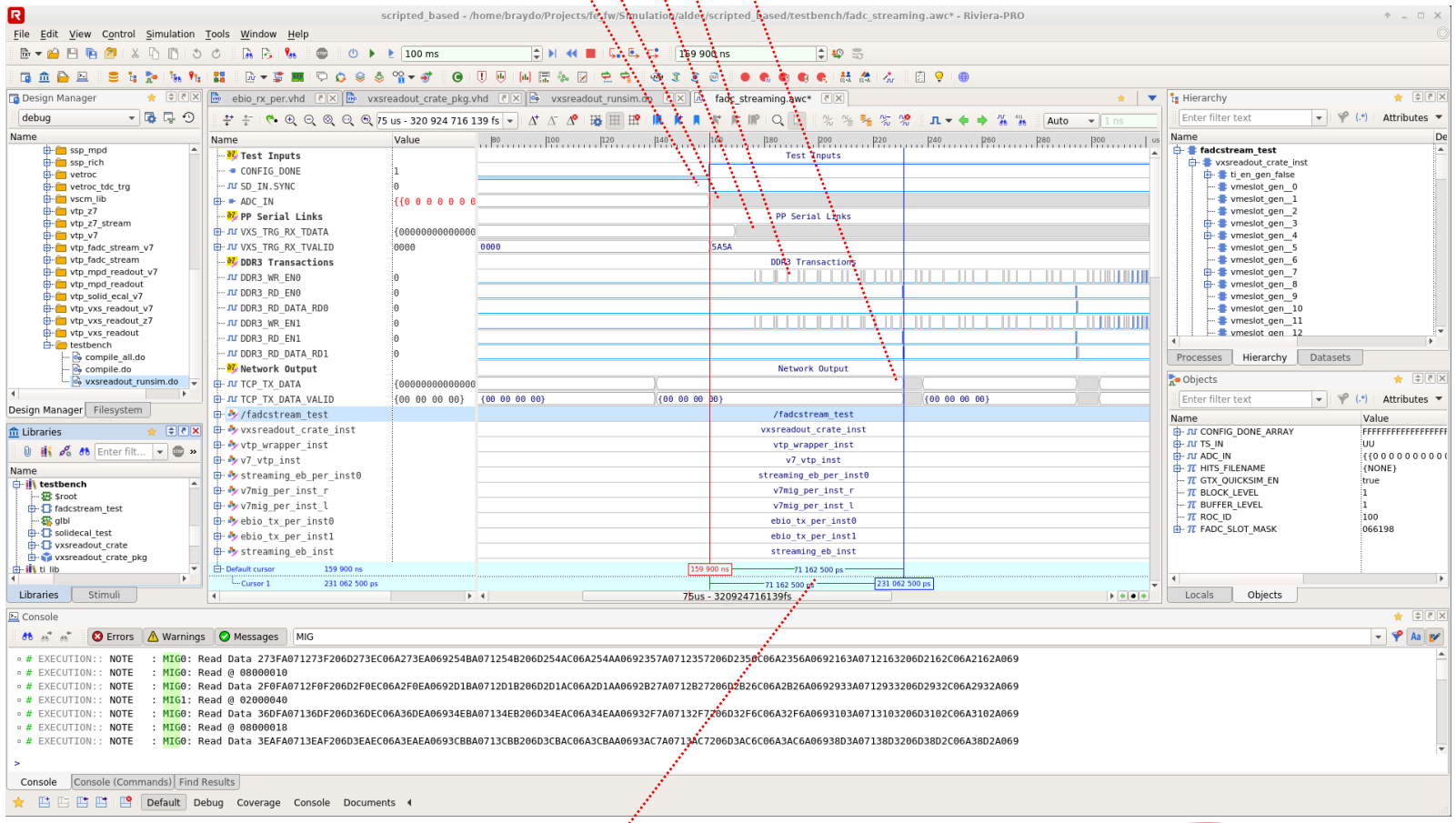
Jefferson Lab

Front-end Crate Simulation

Aldec Riviera used for VHDL, SystemVerilog, C/C++

- Read & write files & network
- Library access (ROOT, EVIO, etc)
- Detailed waveform viewing for low-level debugging
- Accurate latency information
- Have been able to reproduce many failure cases from real system tests
- Simulation isn't fast: few minutes to simulate a 65µs for a single crate of 256 FADC channels, but often only takes a few frames to get needed information

1. SYNC released (starts data flow)
2. Analog pulses to FADC250
3. Serialized pulses to VTP
4. Writes/Reads to/from DDR buffer
5. Frame builder writes TCP stream



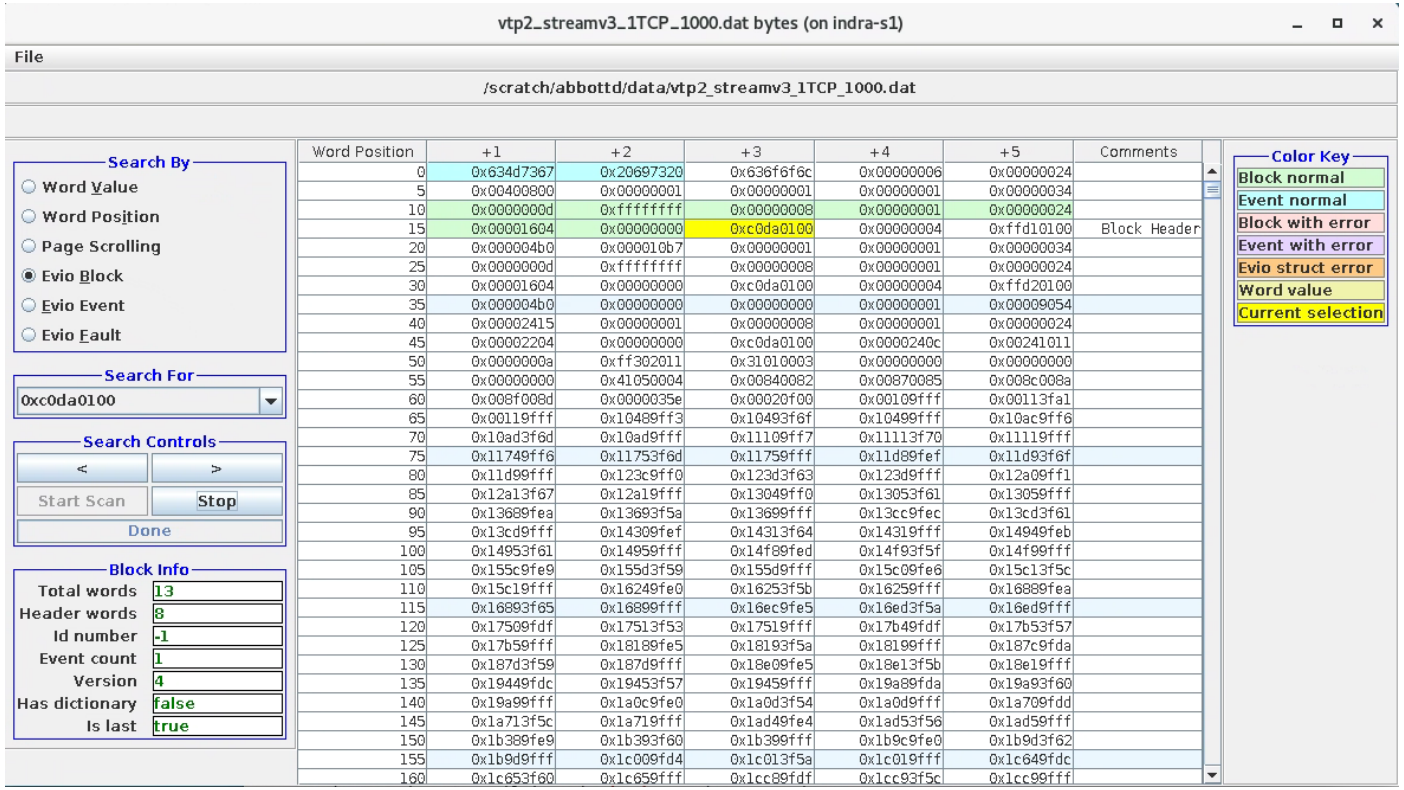
71µs latency: analog pulse to network interface

Jefferson Lab

Event Viewer/Processing Tools

EVIO compliant files come from real DAQ and simulation output. Allows standard Jlab DAQ tools and event builders to use these sources from the new streaming data system.

E.g. jeviodmp is checking 'event block' format for correctness:



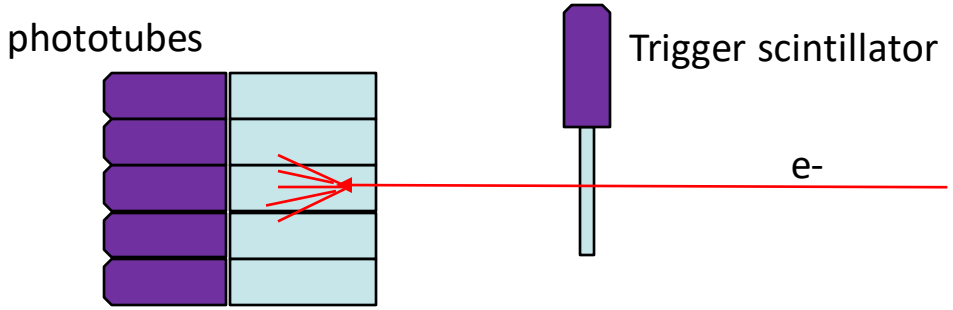
Jefferson Lab

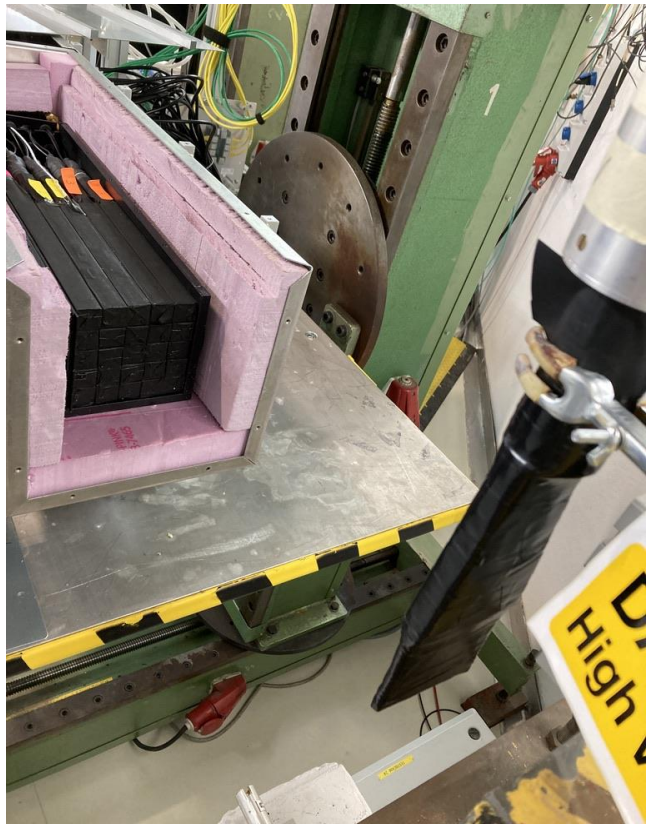
Beam Tests

- Recent beam tests with a calorimeter prototype at DESY (*Thanks to Doug Hasell for coordinating this opportunity*)

- 5x5 PbWO4 Crystal Array (2 cm^2 face) with 2-5GeV electron test beam

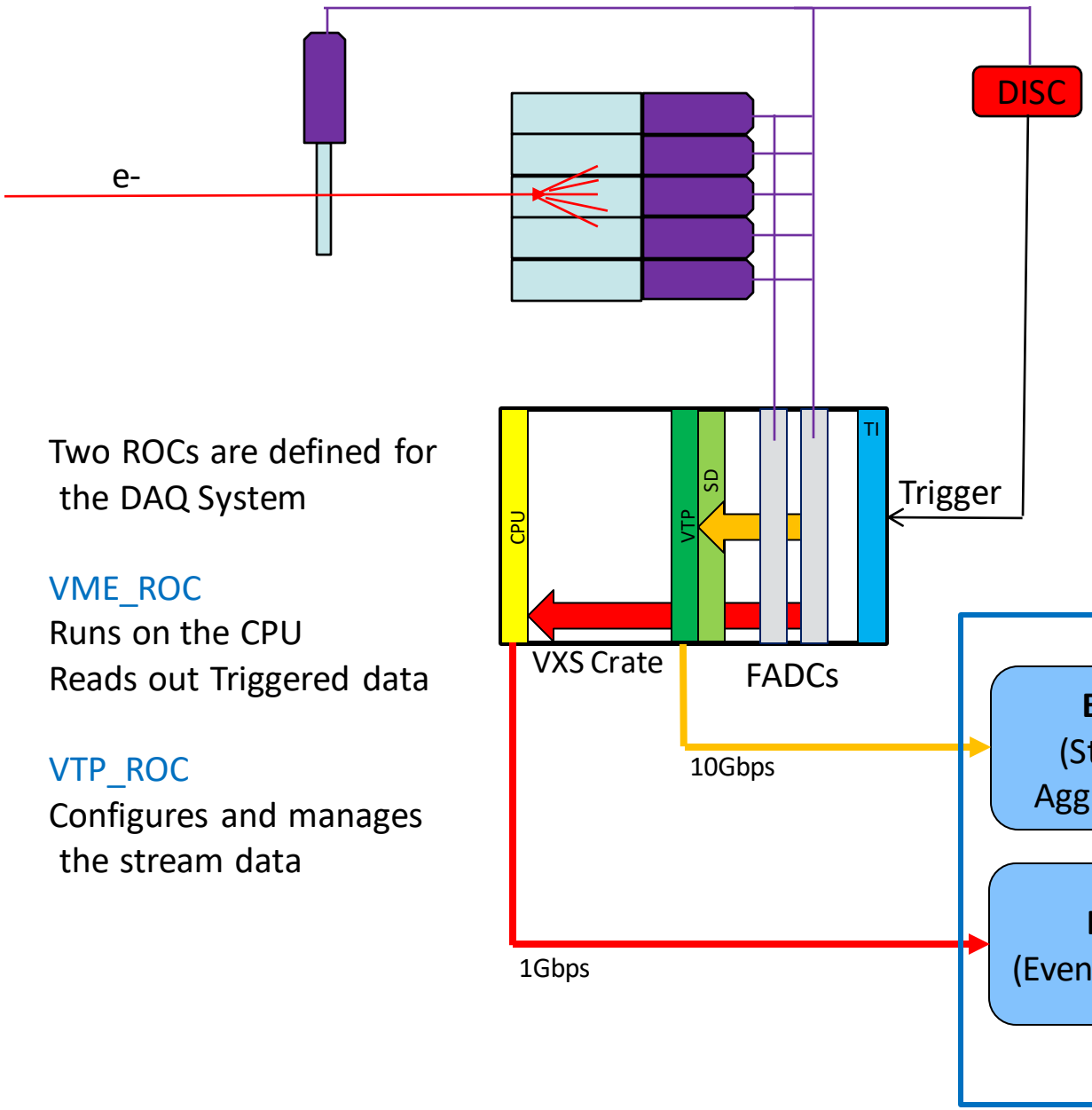
- Jlab 250Mhz FADC boards
  - Triggered data are waveforms read out over VME bus.
  - Stream data are integrated sums and times of all hits over a threshold in the calorimeter regardless of the trigger status.





Jefferson Lab

Simple Hybrid CODA System



Two ROCs are defined for the DAQ System

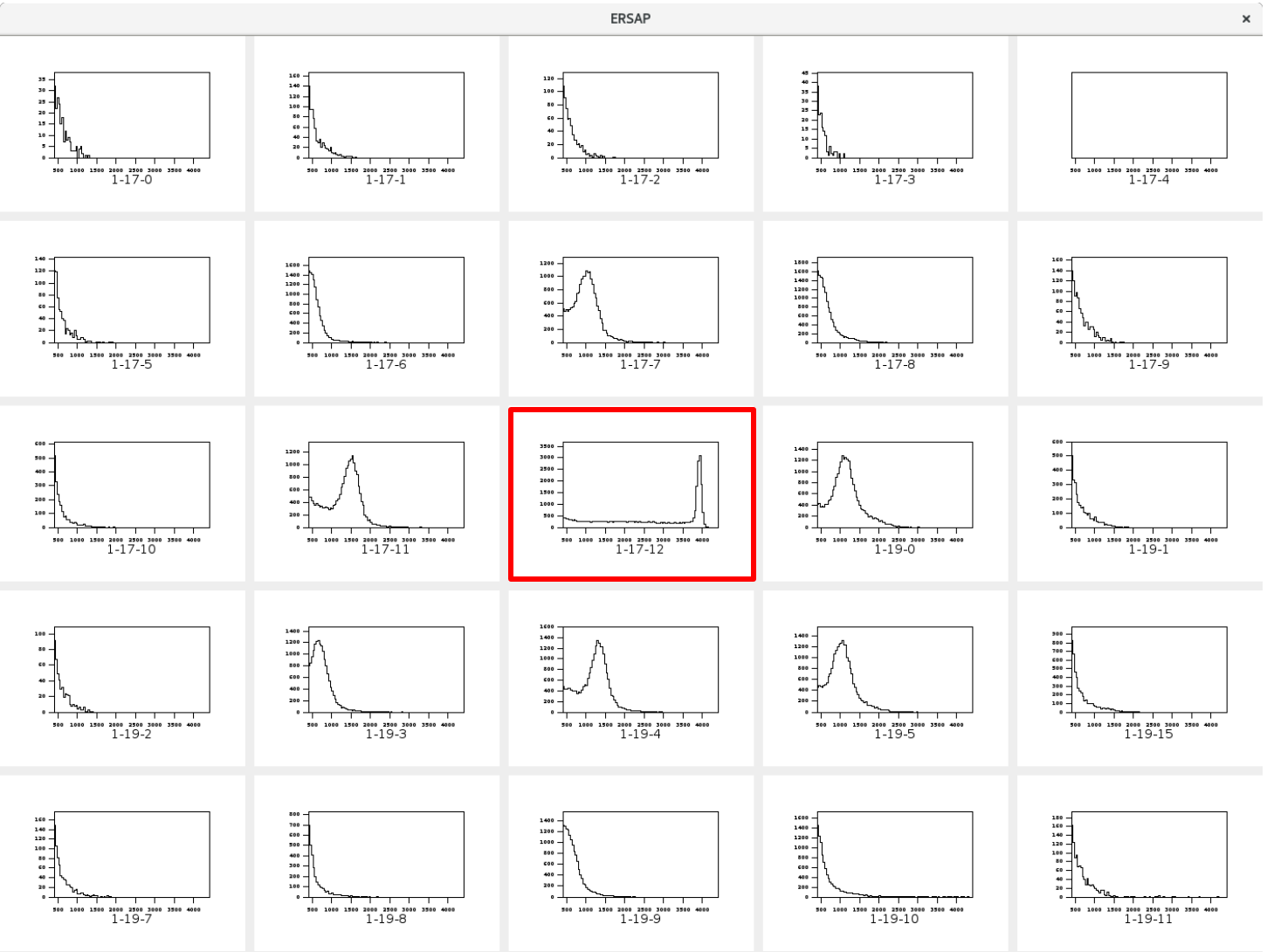
VME_ROC
Runs on the CPU
Reads out Triggered data

VTP_ROC
Configures and manages the stream data

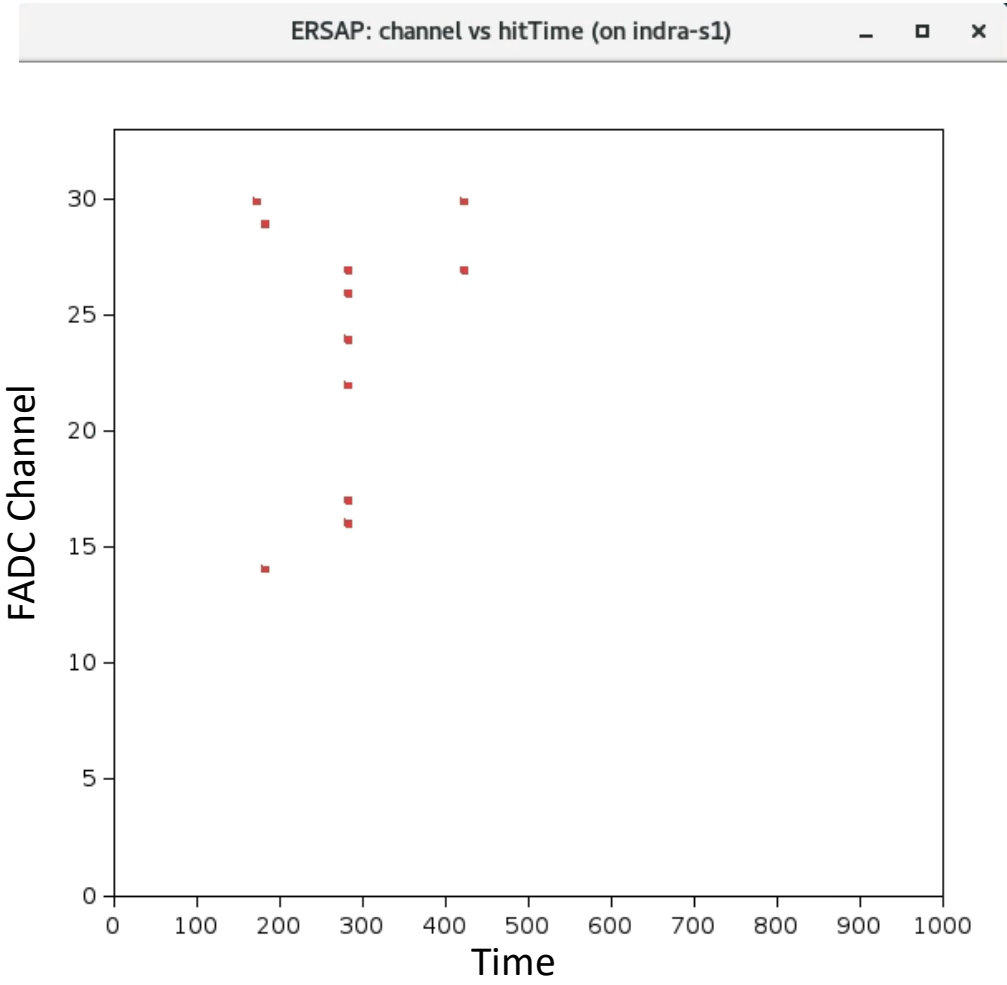
Beam Tests

Calorimeter spectra – Streaming Data

(Electron beam centered on the central crystal)



Time Slice Frame containing a Trigger

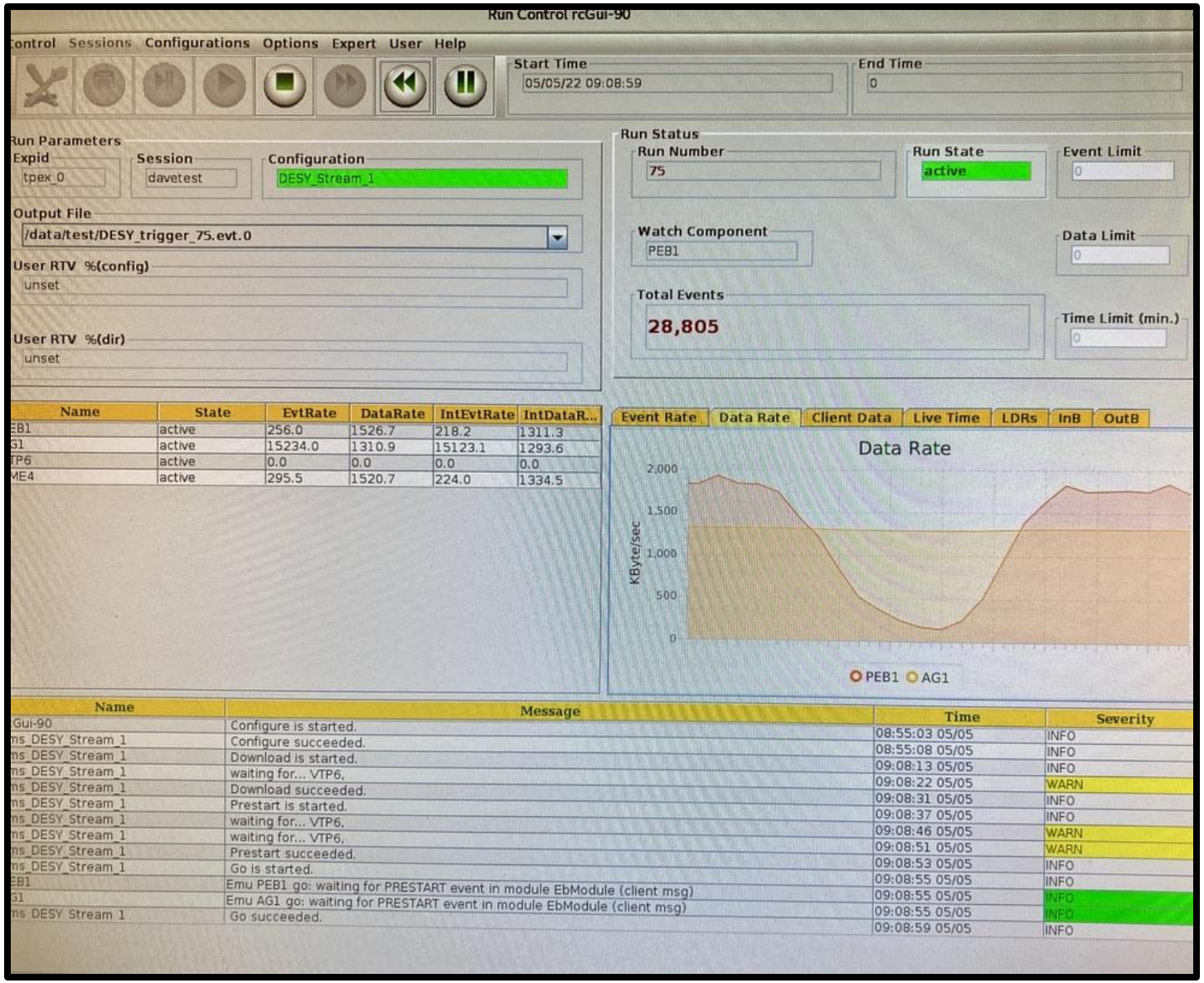


Jefferson Lab

Beam Tests cont…

- Rates were relatively low for these tests
- The electron beam varied in current depending on if they were filling the PETRA synchrotron.

- Note the respective data rates on Run Control.
  - Streaming frame and data rates are relatively constant (15.2kHz, ~1.3MB/s)
  - Triggered rates rise and fall as the electron beam comes and goes (reading out wave forms even in triggered mode generates a lot of data)

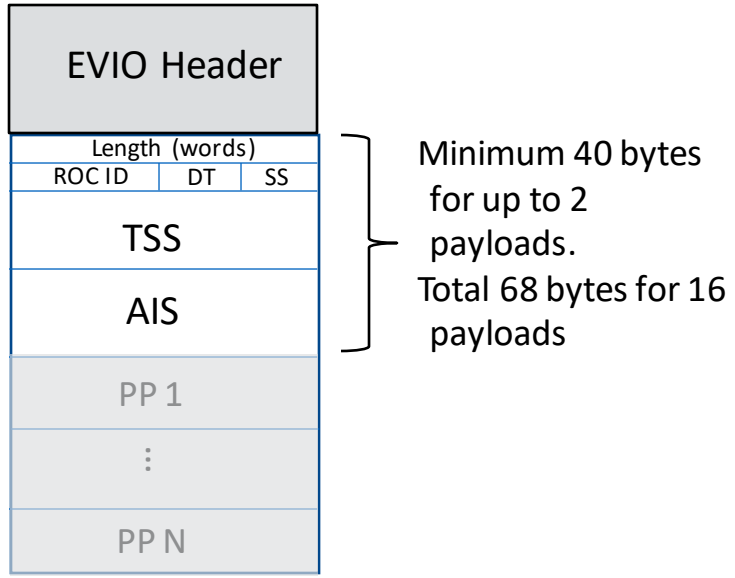


“Zero” Suppression

- In the streaming environment we have to manage these the two extremes
  - Empty time frames
  - Too much data for available bandwidth
- For the current Streaming ROC format there is a minimum 72 bytes/frame sent.
  - For 65µs frames that comes to ~1.1MB/s ”empty” data rate.
- Making time frames longer reduces the overhead. Allow for an adjustable time frame (planning to expand support fo frame size 65µs to >1ms).
- EVIO Header (32 bytes) is just for transport to the Aggregator – then stripped.

Empty Frames for the Beam Tests

TET (threshold over pedestal)	Empty Frames	Frames with Triggers
10	68.9%	0.084%
50	97.0%	0.64%



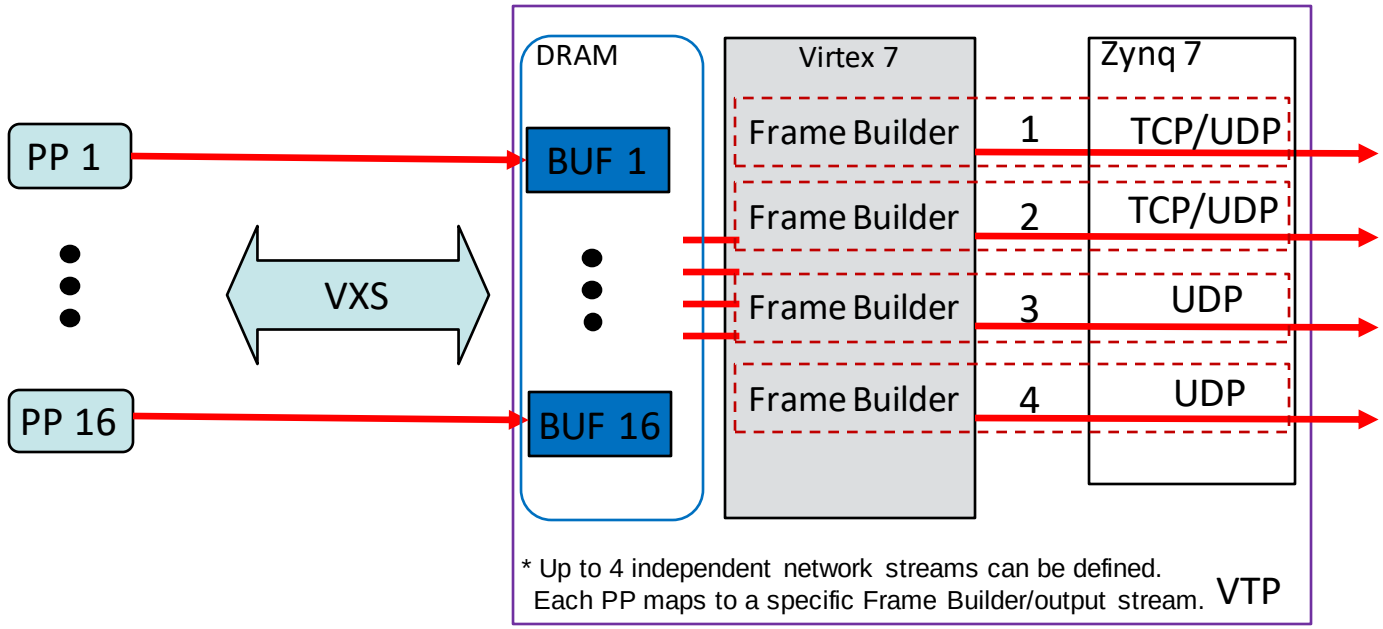
Jefferson Lab

Congestion & Latency Management

- Too much data can be handled both locally and globally
- Globally "Sync" is used to start and stop all the streams at their source (FADCs)
  - An optional "Busy" generated by any ROC can feed back and inhibit all streams for all ROCs

- Locally, PPs stream hits to a VTP DRAM buffer. The Frame Builders have a frame "fifo"
  - DRAM can hold programmable number of frames (anywhere from 1 to many thousands).
  - When frame buffer is full, the front-end payload card will drop the frame (frame timestamps allow tracking loss).
  - This frame buffer depth constrains the latency of the frame -> network (when UDP is used):

    Our typical settings: max latency = 131kByte $_{(FrameSize\_max)}$ * 256 $_{(FrameBuffer\_max)}$ * 4 $_{(PP\_num)}$ / 9.5Gbps $_{(LinkSpeed)}$ = **113ms maximum**
    80µs minimum is typical since we expect to operate at low link occupancy



Note:

TCP Performance
~7-8 Gbps per link without frame drops

UDP performance (8000 MTU)
>9.5Gbps per link without frame drops
~50% CPU utilization for a single stream

(These tests were done with both VTP and a Server connected through a single switch)

Jefferson Lab

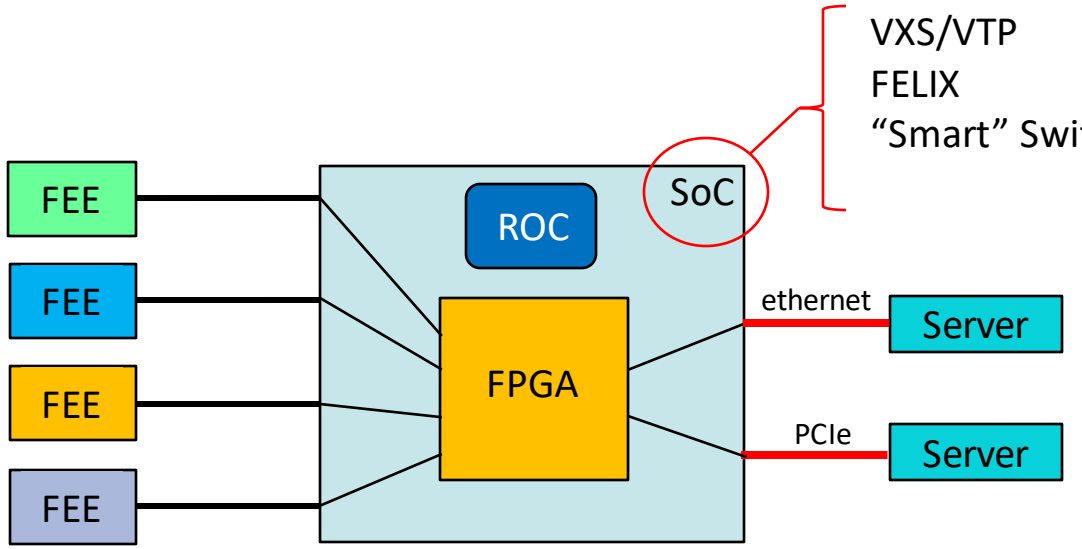
Some general observations…

- The design of our Clock/Trigger/Sync/Busy distribution system is critical to the flexibility and functionality of the CODA Hybrid DAQ.

- The VXS platform works for JLAB (as we have a large inventory of this hardware), but is an impractical and/or financial overhead for small university groups. The same could be said for other solutions like the FELIX/DAM architecture for EIC.

- It seems there is a need for an affordable COTs or alternative ”lightweight” solution which supports Streaming that can be made available to the community for development and test systems. Front-end modules with optical outputs ideally would have Ethernet support for easy compatibility with standard PCs.

The critical component to just about any system is a System on a Chip with enough resources to support at least a few Front End electronics serial link protocols and perform 1st stage hardware stream aggregation. And present the data to the next stage in a standardized way.



Summary

- We have successfully started integration of Streaming support within the CODA software framework and supported hardware.
- The Hybrid DAQ system give us a lot of flexibility to support older hardware within a Triggered system as well as newer hardware that can conform to the the Streaming requirements.
- UDP data transport from the VTP is proving to be reliable and the most efficient method for getting data to backend processing.
- Upgrades to JLAB FADC firmware this Summer will allow for more streaming options including waveforms.
- The ability to take both Triggered and Streaming data simultaneously should provide useful data for Online processing to better define efficient event identification algorithms as part of a high level trigger.
- Integration of other ASIC-based front-end electronics within the CODA streaming environment still needs to be developed.
- CLAS12 DAQ is working to replace a few front-end electronics that don't support streaming. They are making an effort to secure the option to switch to a streaming DAQ. Would be a great test bed that may not require a huge investment.

Timing System Components