Streaming with CODA at JLab

Streaming readout development and CODA support for Jefferson Lab experimental programs

Streaming Readout Workshop – SRO XII

Tokyo, Japan, Hybrid Meeting

Dec 2-4, 2024

David Abbott

FEDAQ Group

Jefferson Lab – Physics Division

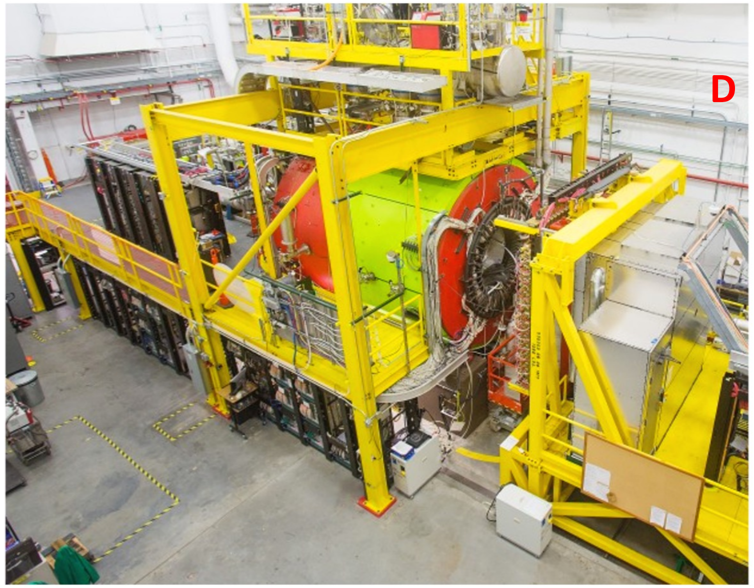
Jefferson Lab

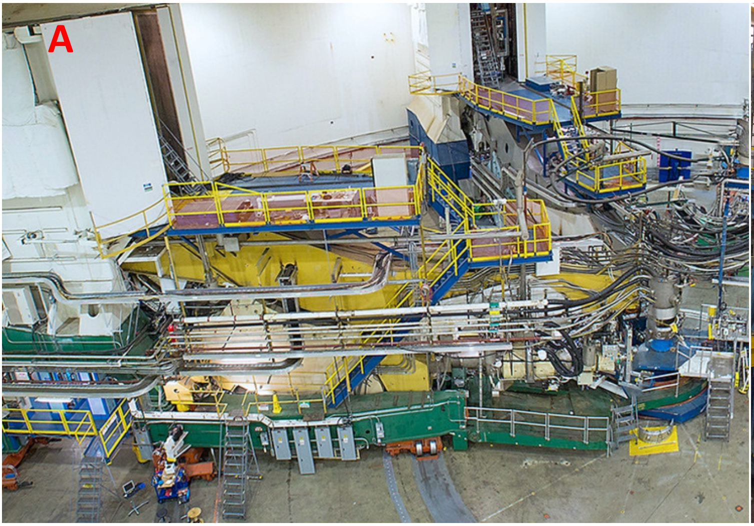
Data Acquisition at Jefferson Lab

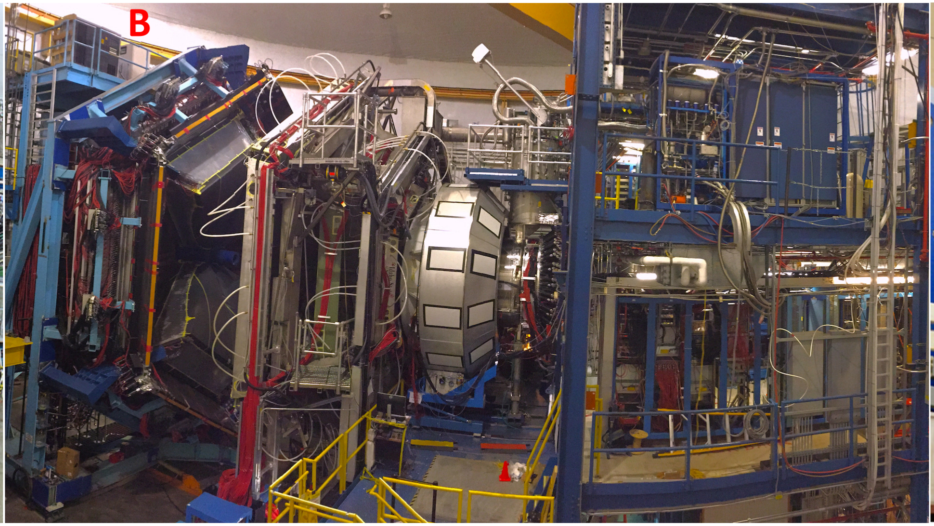
- At JLab we have 4 Experimental Halls, all running with different detectors, and physics focus.

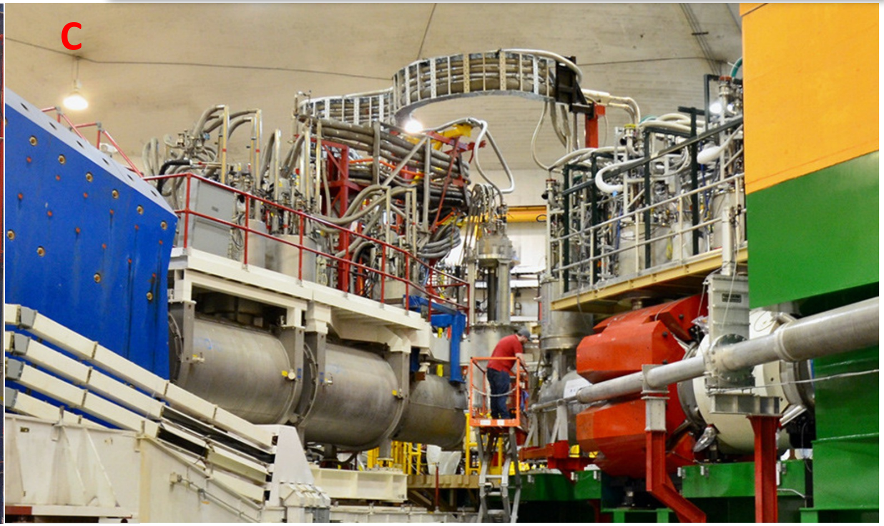
- Experiments are increasingly reliant custom electronics to interface detectors and digitize signals.
  - But older hardware is still relevant and useful (particularly for starving budgets)

- Our goal is to support both the existing **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 keep it as consistant and user friendly as possible









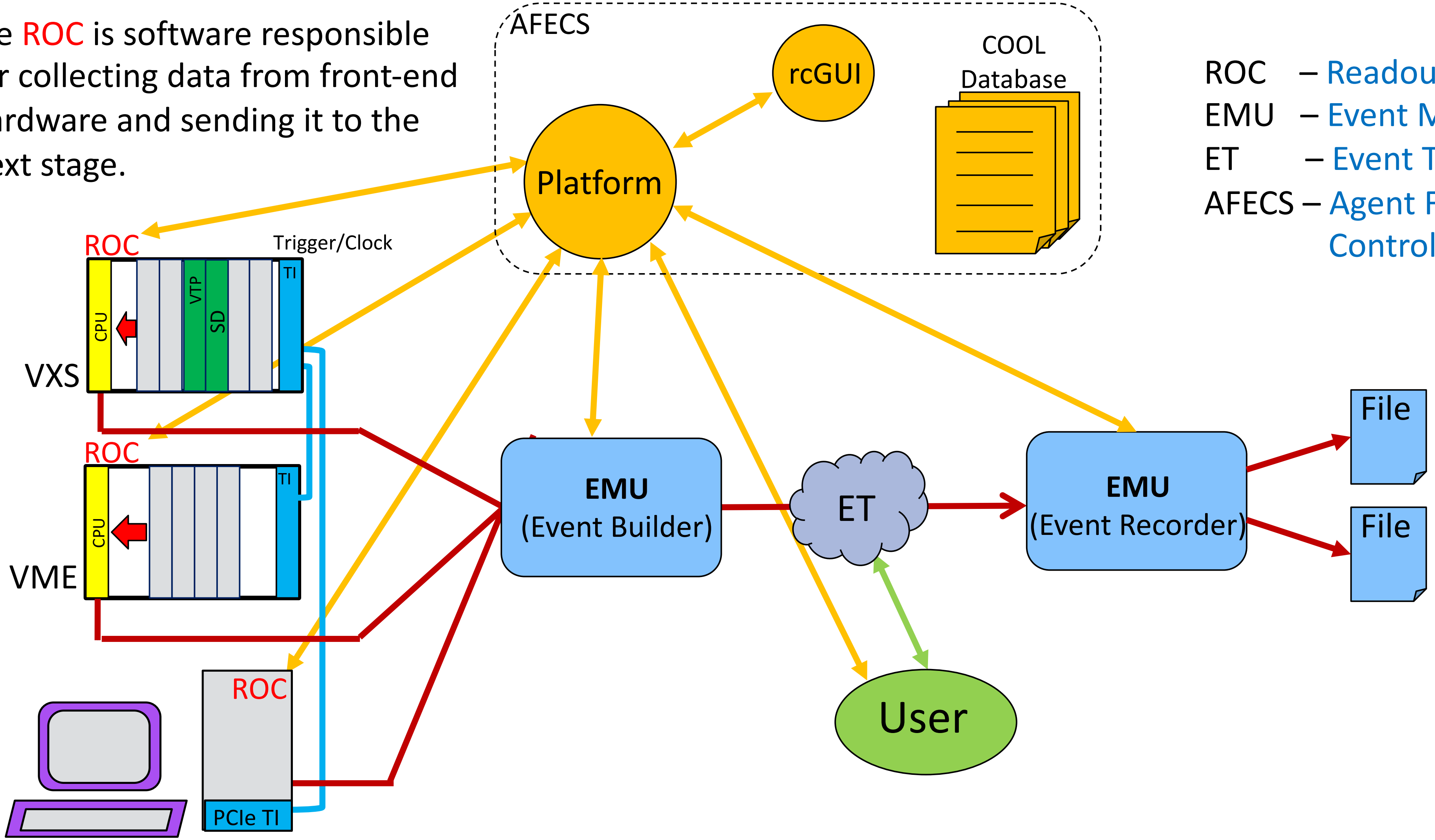
CODA – CEBAF Online Data Acquisition

- What is CODA? (also see https://coda.jlab.org)
  - Software toolkit for implementing data acquisition systems.
  - Hardware/Electronics
    - Custom boards (e.g. Trigger/Clock, TDCs and ADCs)
    - Support for commercial hardware.
  - Software includes :
    - Interface with electronics (libraries/drivers).
    - Readout Front End and format data (ROC)
    - Inter-process communication - Control and Data (cMsg)
    - Merge data streams (Event Building, Stream Aggregation)
    - Give users access to data for analysis and monitoring (ET System)
    - Write data to files (EVIO, Event recording)
    - Manage and control the data acquisition system (AFECS)

- CODA is modular
  - Build a single crate DAQ test bed or a full experimental hall system

The CODA Data Acquisition (software) Toolkit

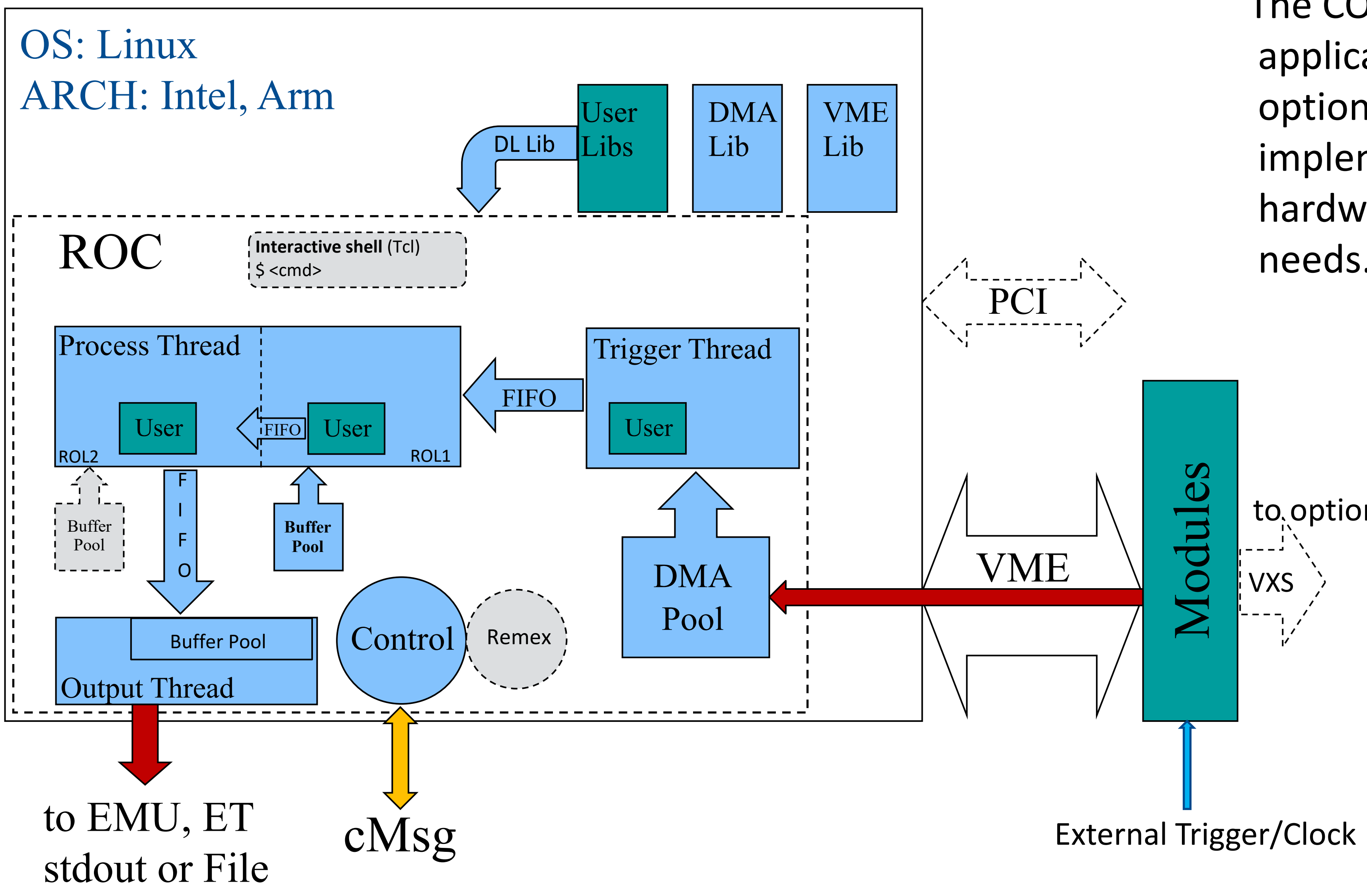
The ROC is software responsible for collecting data from front-end hardware and sending it to the next stage.



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

Data transport and all files are all written in: CODA EVIO Data Format

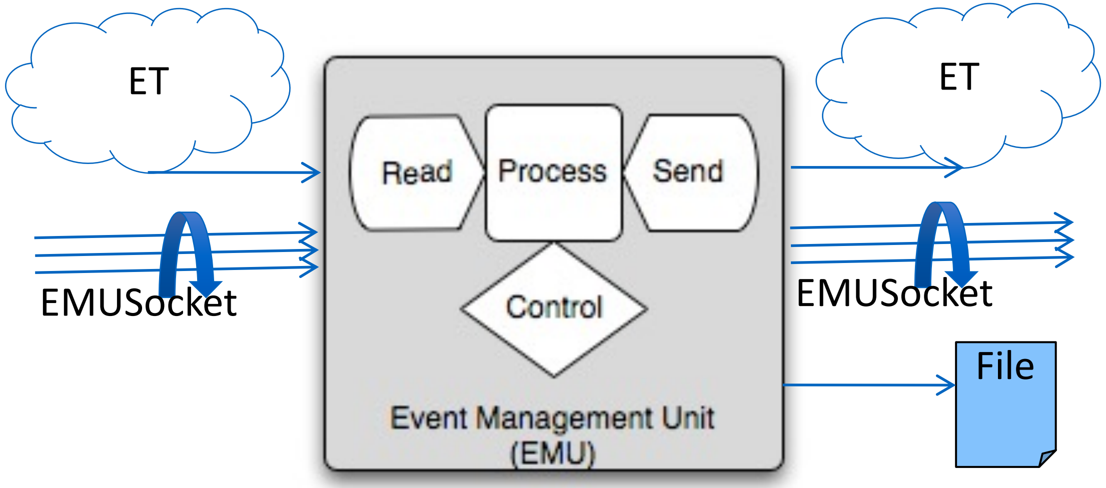
Readout Controller (ROC)



The CODA ROC is a software application that has multiple options for the **User** to implement custom code or hardware for their specific needs.

The EMU Component

- EMU – Event Management Unit – is a JAVA-based general processing application for DAQ. It comes in many flavors:
  - DC – Data Concentrator
  - PEB – Primary Event Builder
  - SEB – Secondary Event Builder
  - ER – Event Recorder
  - PAG – Primary Aggregator
  - SAG – Stream Aggregator



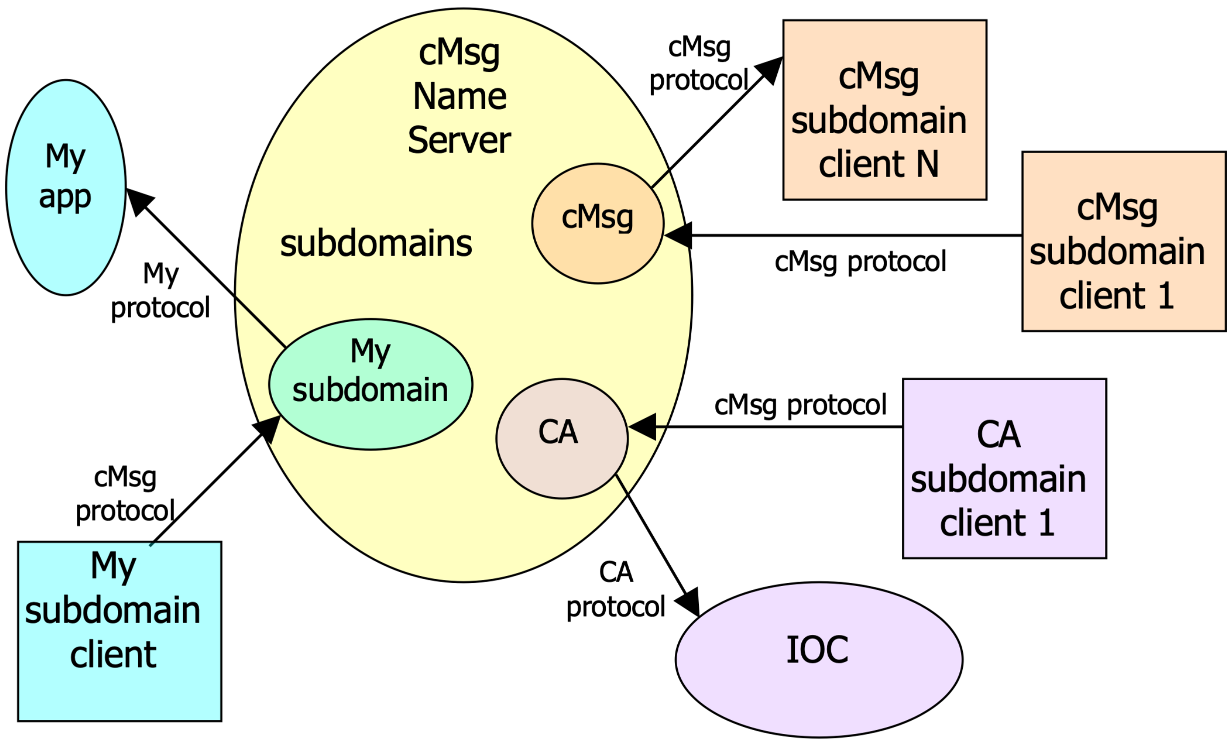
Input/Output Connections:

- Event Transport (ET) system
- EMUSocket protocol - part of the CODA cMsg library (it allows for multiple connections or “fat” pipes on high bandwidth networks)
- EVIO data file

cMsg – CODA Messaging

- **cMsg is a publish-subscribe, inter-process messaging system.**
- At the most basic level it is an API for sending and receiving messages. This API is used to wrap a variety of communication protocols.
- Supported in C, C++ and Java on Linux and MacOS.
- All online CODA components use cMsg to communicate both control information and create high-speed data links to each other.

- Available sub-domains include:
  - cMsg – General publish-subscribe
  - rc - Run Control
  - rcs - Run Control server
  - rcm - Run Control multicast
  - emu - EMU data links
  - CA - EPICS channel access



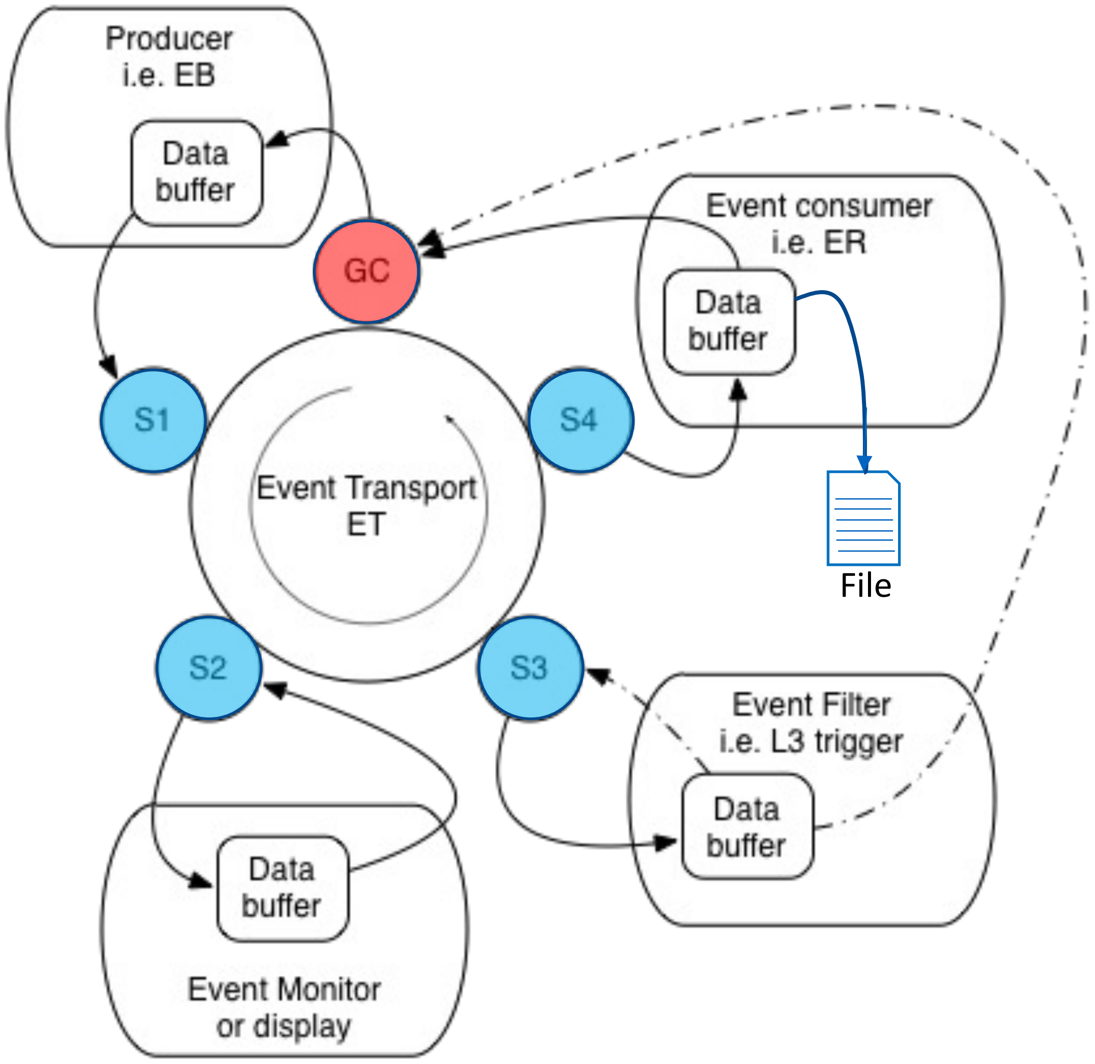
Jefferson Lab

Event Transport (ET System)

- The ET system gives programs access to data via pre-allocated **shared memory** buffers.
- The system uses a railroad metaphor. Free data buffers are queued at Grand Central. They are filled by data producers and tagged to describe the content.
- The buffers "move" around a circular track and at each **Station** the tag is checked to see if the buffer should queue at the station.

Examples:

- An event monitor could set up station, **S2**, to take 1% of the events.
- An event filter could set up **S3** to take all events. Discarded events are sent back to GC good ones move on.
- An event recorder (**S4**) takes all events and, after the data is written to a file sends the buffer back to GC.

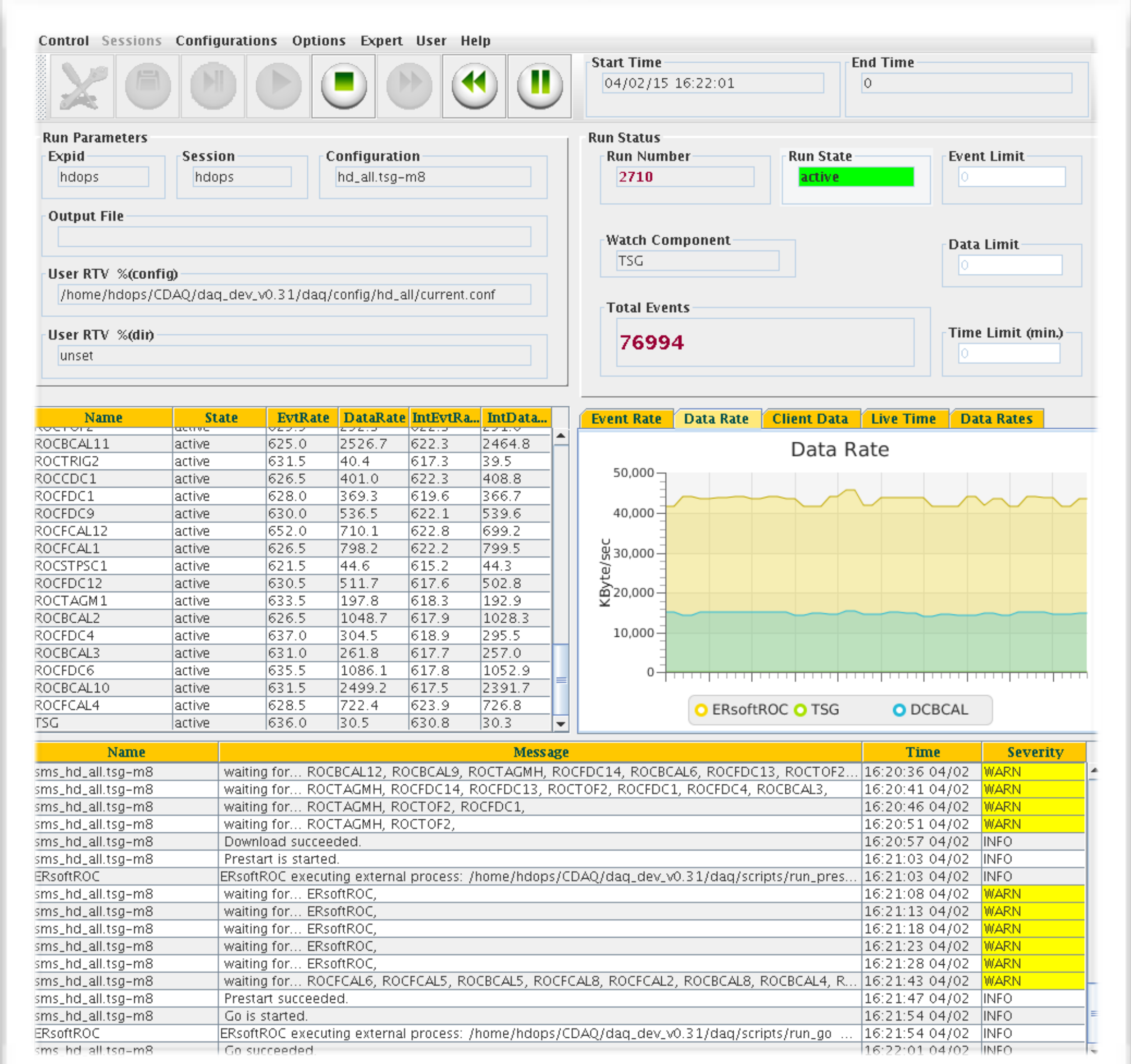


Run Control - AFECS

The **Platform** is a JAVA-based application running multiple “agents” that monitor and control external CODA client components (ROC, PEB, ER etc.) or internal processes (scripts).

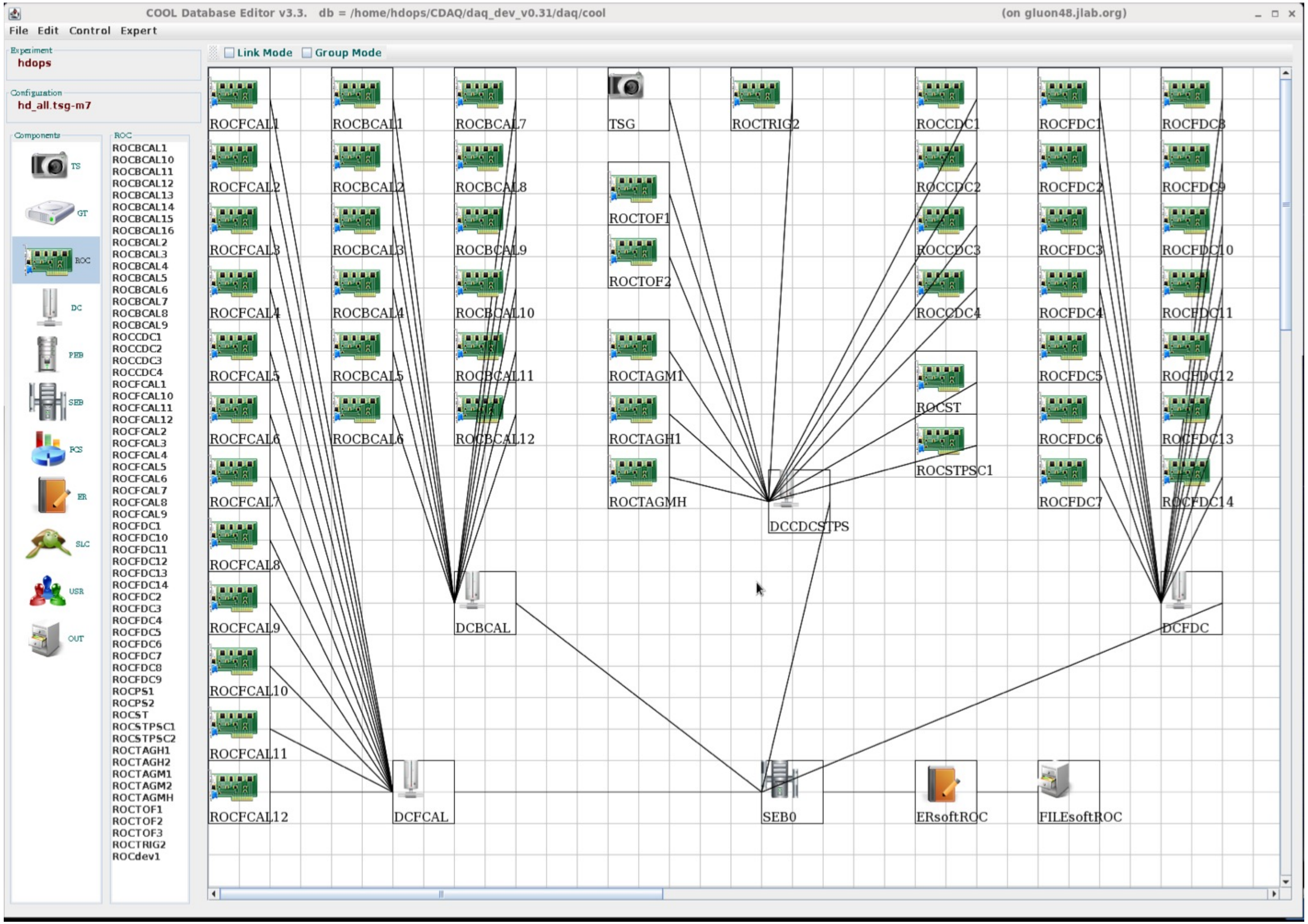
Multiple run configurations can also be operated simultaneously.

Many **rcGUI** processes can communicate with a single Platform that is defined by a COOL Database with a Name = env(EXPID).



External commands can be used in User scripts to communicate directly with the platform.

COOL Database Configuration Editor – "jcedit"



The Java program **jcedit** allows the User to graphically create different DAQ configurations, defining the components, data links, data files and other details.

Example: Hall D GlueX

1 Trigger Supervisor
50 Readout Controllers
4 Data Concentrators
1 Secondary Event Builder
1 Event Recorder
1 File output

CODA EVIO File Display - jeviodmp

Built Physics Triggered Event Block

Built Trigger Bank (+5 ROC trigger banks)

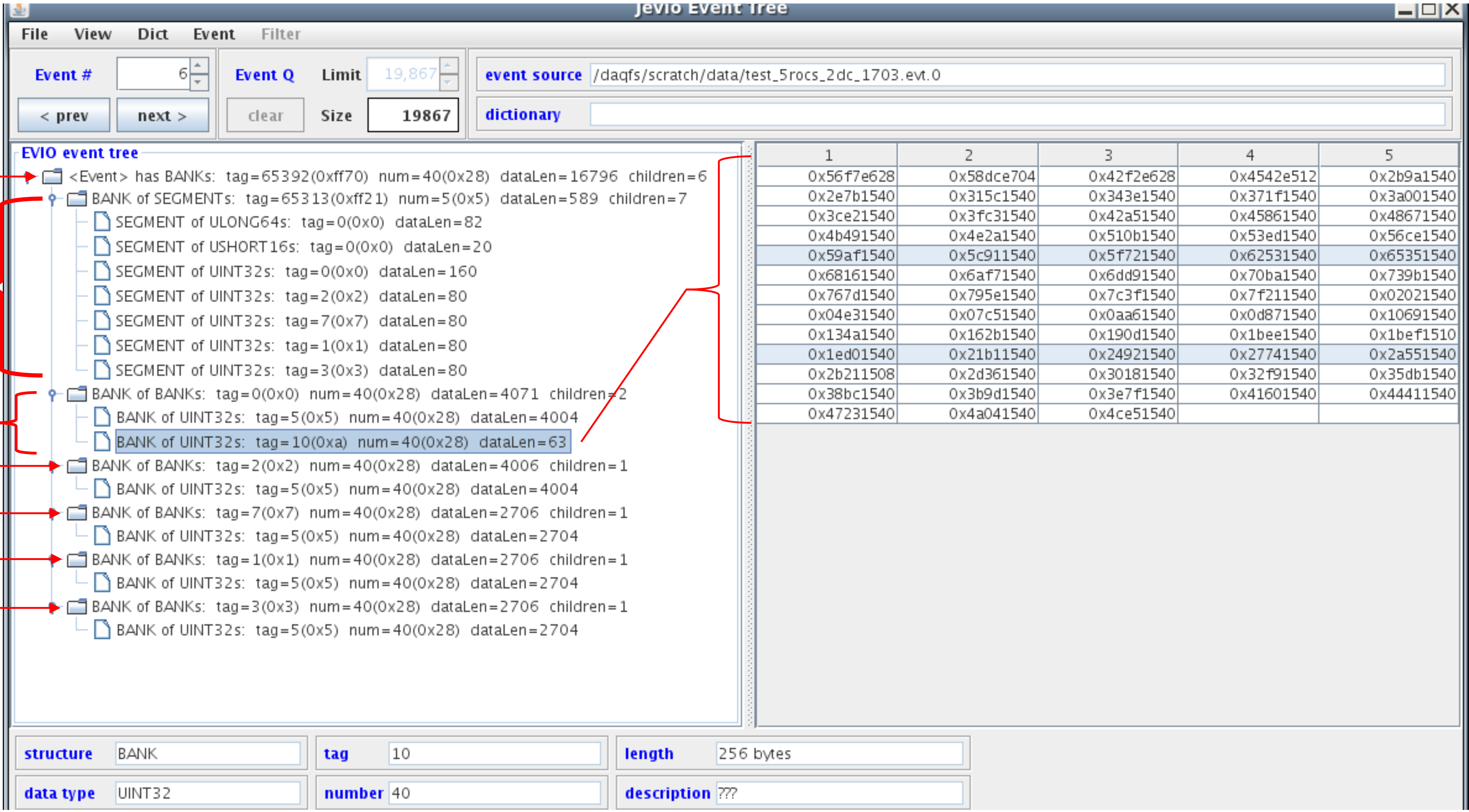
ROC 0 Data Banks (2)

ROC 2 Data Bank

ROC 7 Data Bank

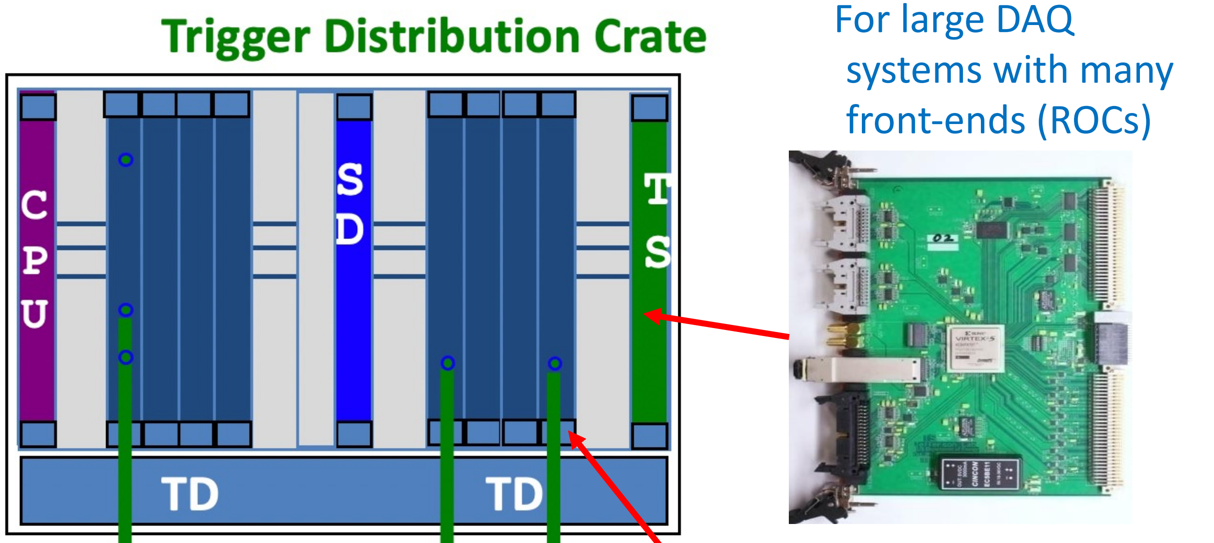
ROC 5 Data Bank

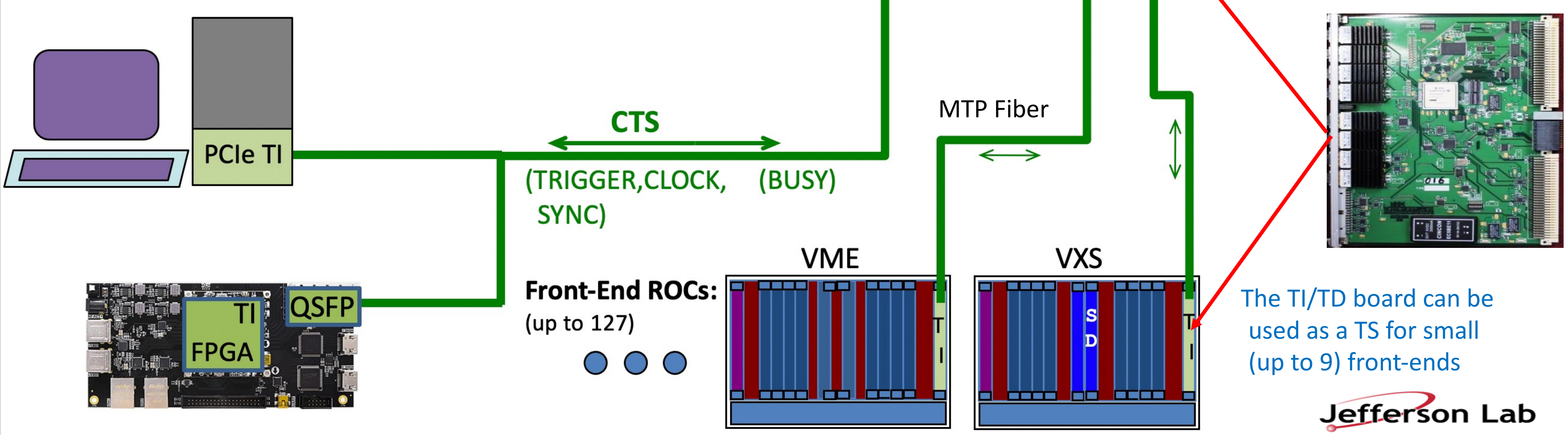
ROC 3 Data Bank



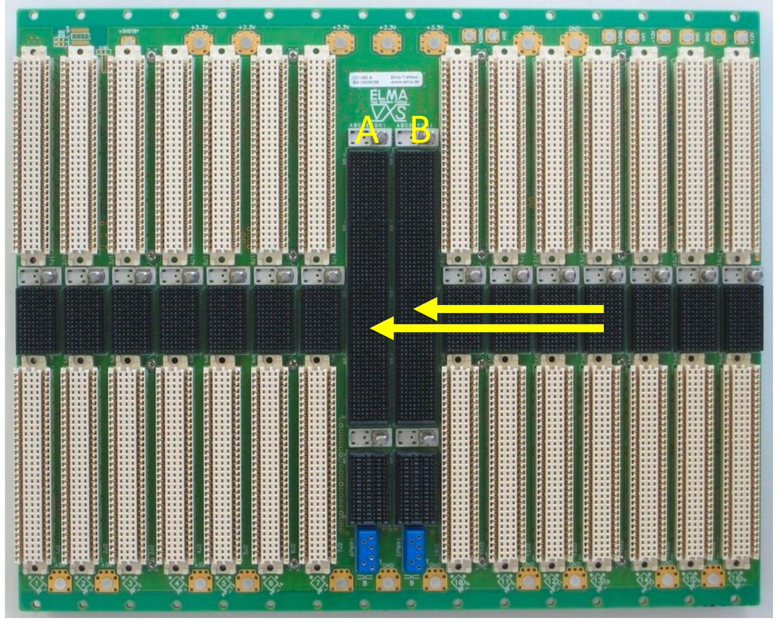
JLAB Clock, Trigger, Sync (CTS) Distribution System

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

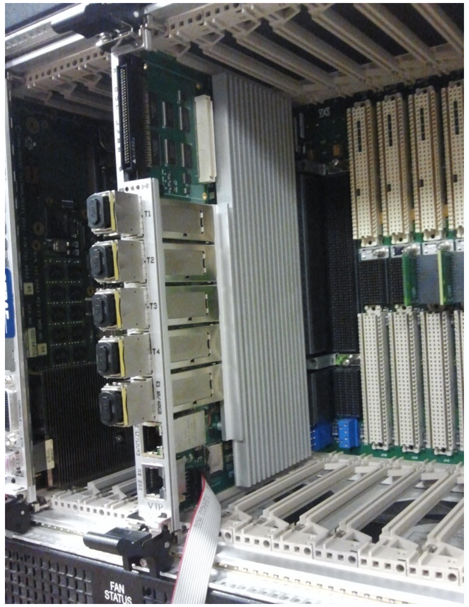




VXS Standard (VITA 41)



- 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.



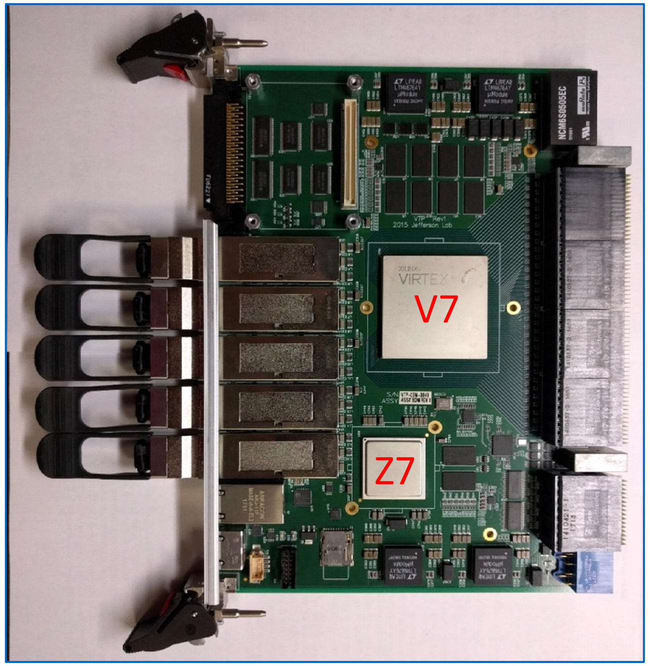
VXS Trigger Processor (VTP)

- Relieve the ROC of all of the "Readout" 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 CODA ROC now is primarily responsible only for Configure, Control and Monitoring the VTP-Based streaming DAQ.

JLAB – VTP Board

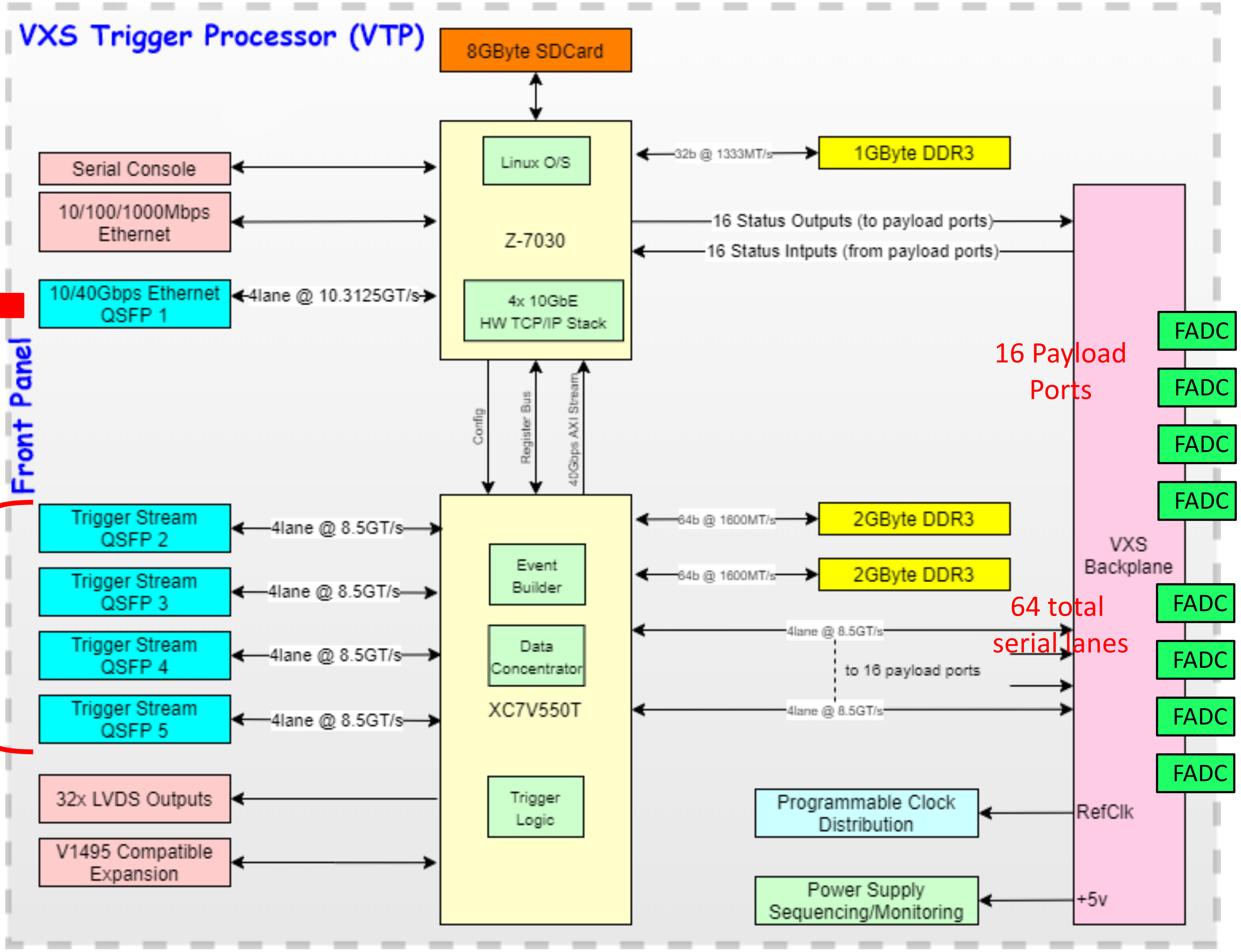
Linux OS on the Zync-7030 SoC
(2-core ARM 7L , 1GB DDR3)
10/40Gbps Ethernet option
(runs the CODA ROC)

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



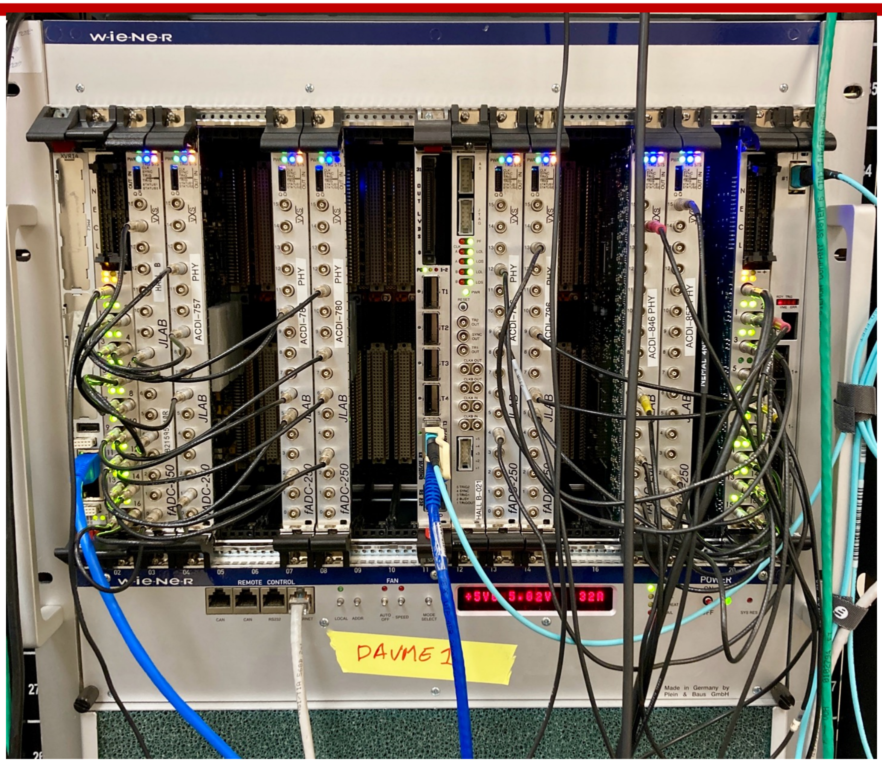
Four 10Gbps Ethernet

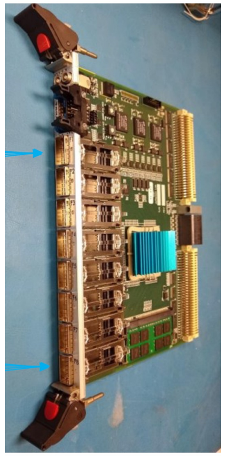
16 total lanes for external serial links



Jefferson Lab

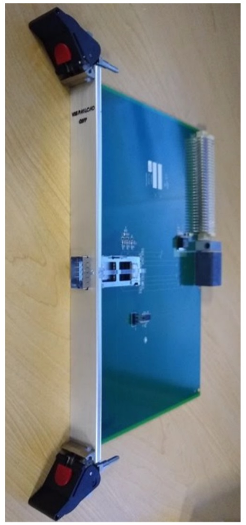
VXS Crates/Modules - Flexible Streaming Platform





Subsystem Processor (SSP)

FPGA board
8 QSFP Inputs (32 links)



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.

250 MHz FADC
16 Chan / 12 bits

JLab workhorse in all current experiments

Used in Streaming testbed (above)

DCRB (TDC)
Drift Chamber Readout Board

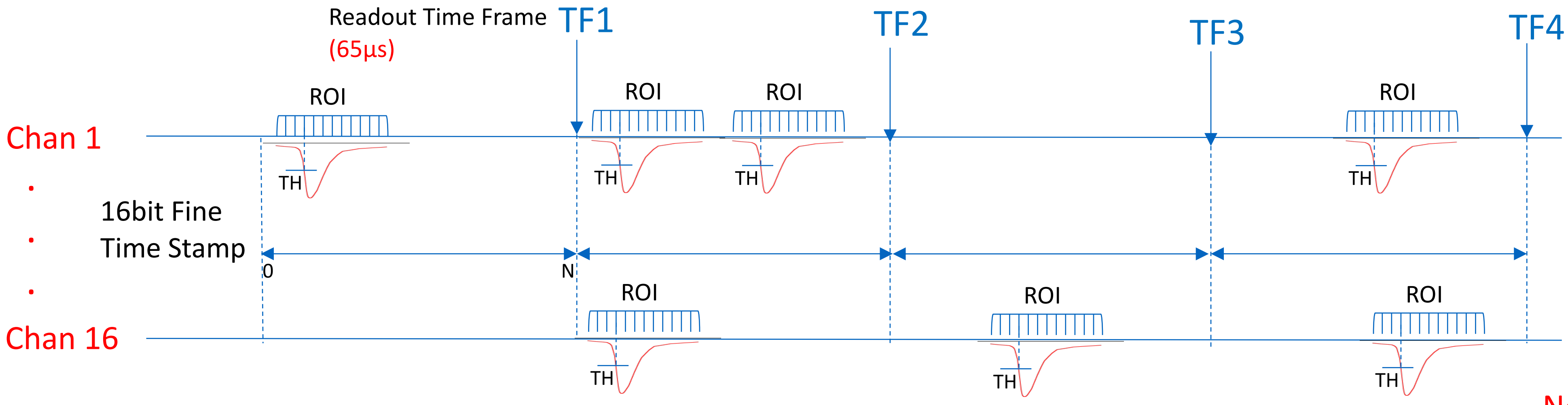
CLAS 12
DC Readout
Triggered or Streaming

Jefferson Lab

JLAB FADC – Streaming and Triggered…

Streaming data can be thought of as Triggered mode where the trigger is a fixed pulser and you keep all the data for a single channel generated between pulses.

A 250 MHz FADC generates a 12 bit sample every 4ns. That's 3 Gb/s for one channel. A 16 channel module is 48 Gb/s. That is over twice the available VXS bandwidth. But we don't need ALL the samples.



Within the FPGA we keep only the data around a Region of Interest (ROI) from each channel, along with a fine time stamp (4ns ticks) in each time frame window (65µs).

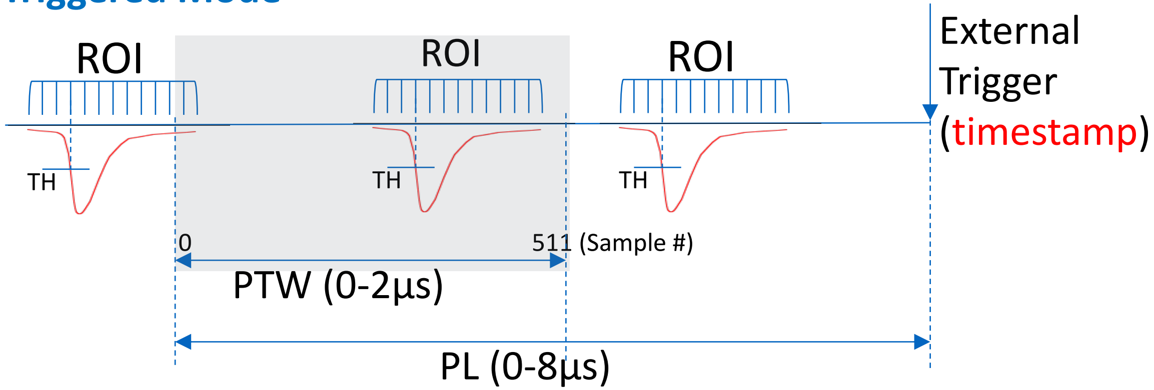
Current FADC<->VTP bandwidth allows for 1 hit/32ns if we compute a sum. Keeping all ROI samples could generate congestion issues that must be handled in firmware. This is currently being developed.

Note: The JLAB FADC can simultaneously operate in "triggered" mode with an 8µs pipeline and 2µs window.

Jefferson Lab

FADCs - Triggered vs Streaming

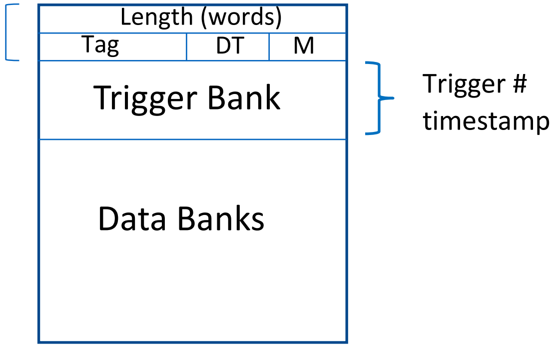
Triggered Mode



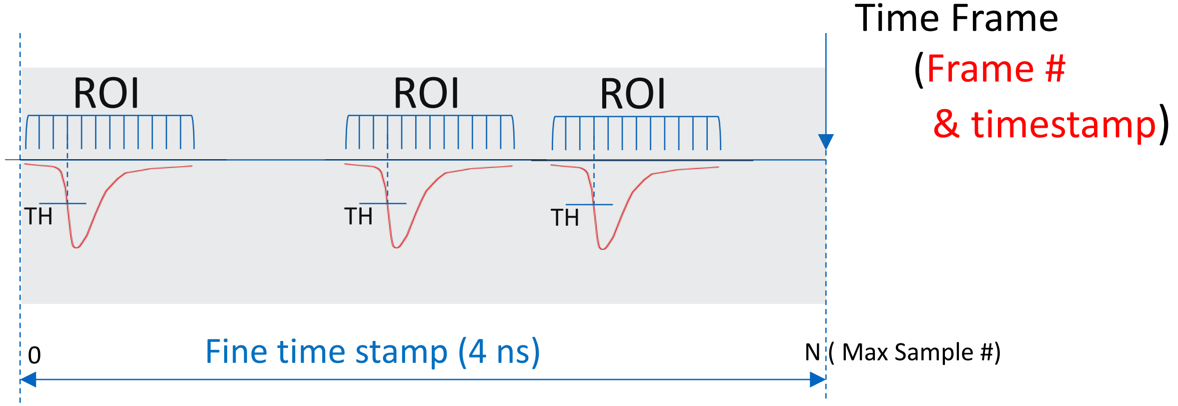
PL: Programmed Lookback
PTW: Time window

Data we get on a trigger:
- FADC waveform values for the ROI
- Threshold Sample # (hit time)
- Trigger absolute timestamp (48 bits)

ROC Data Format



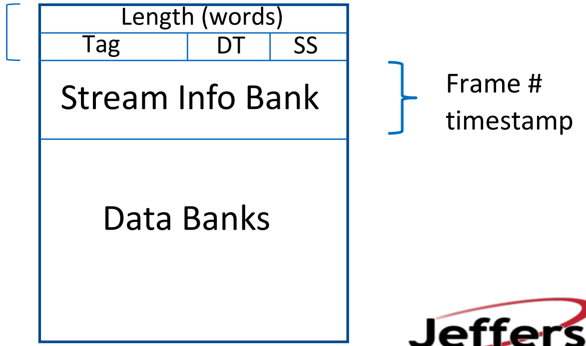
Streaming Mode



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

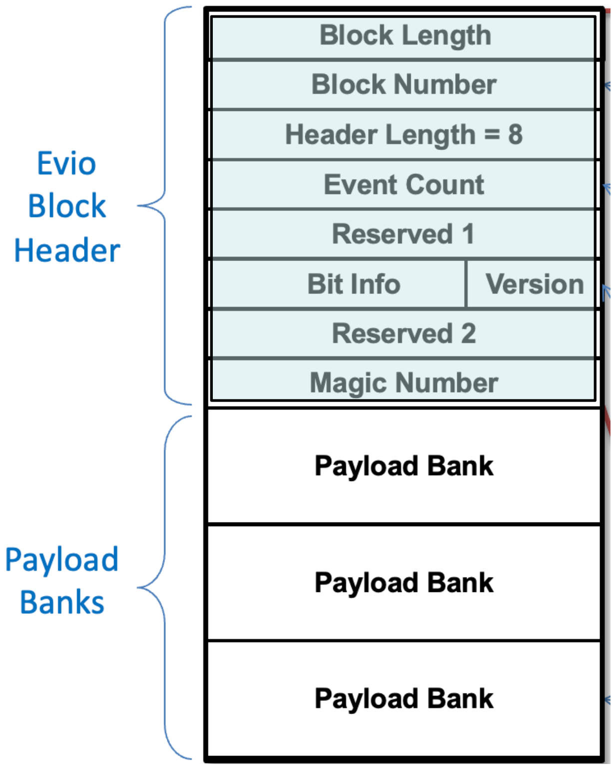
ROC Data Format

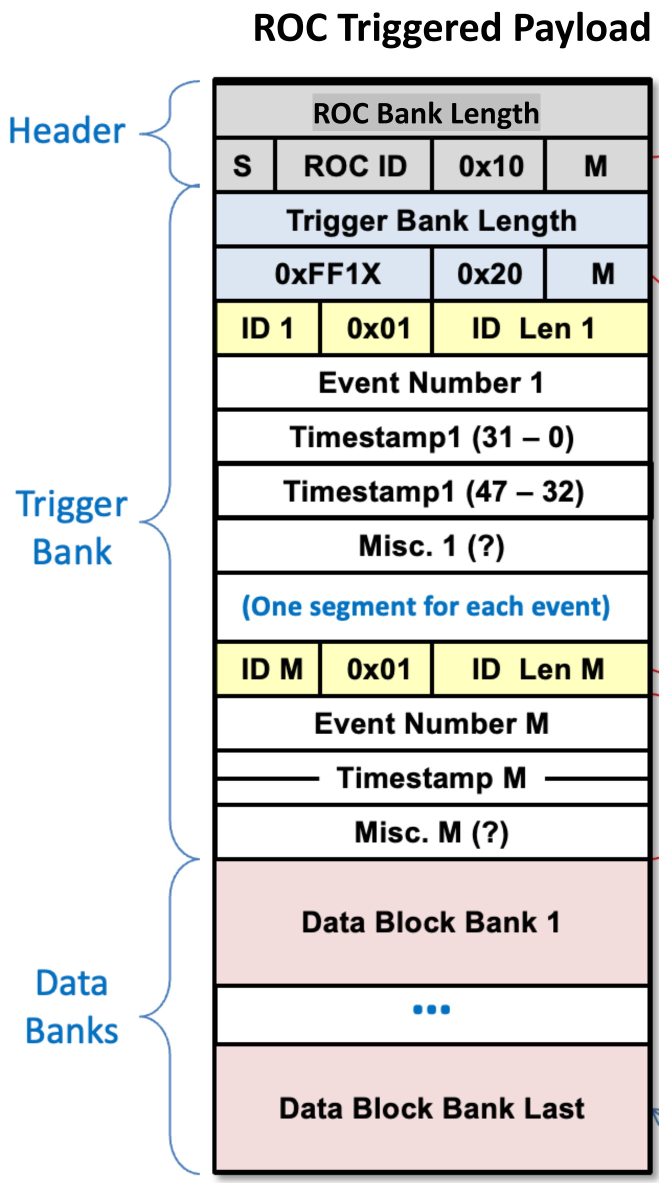


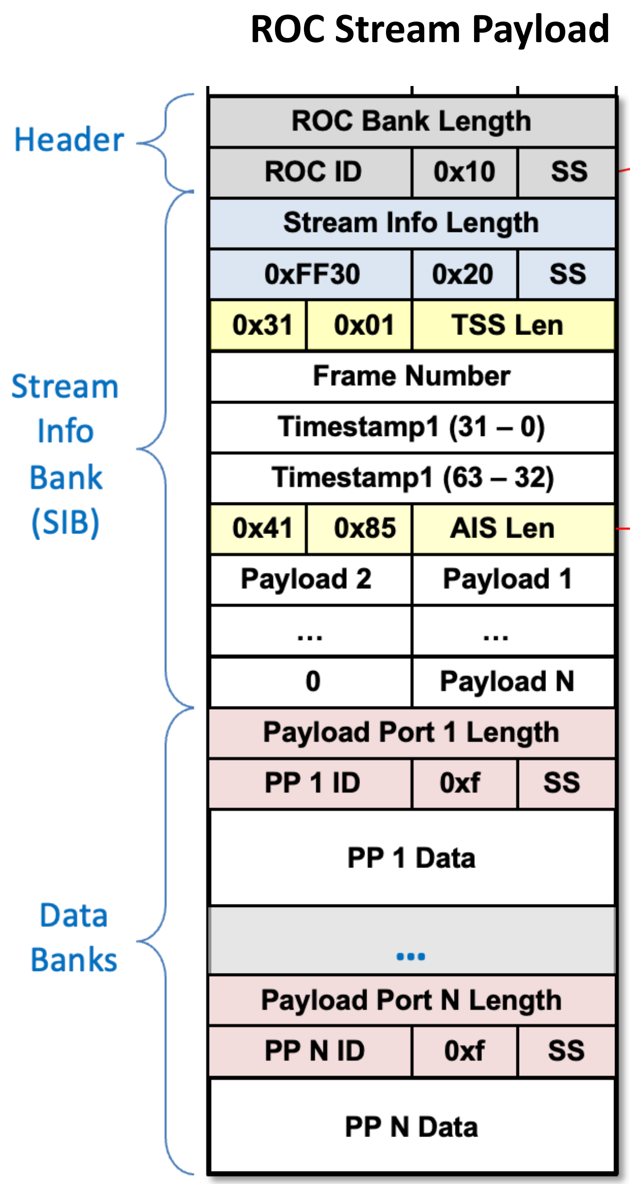
Jefferson Lab

EVIO Data Formats

- EVIO data are stored as a series of four-byte integers.
- EVIO data files are written in "Blocks" and can be appended.
- Each Block is defined by a Header followed by "*Event Count*" payload banks.





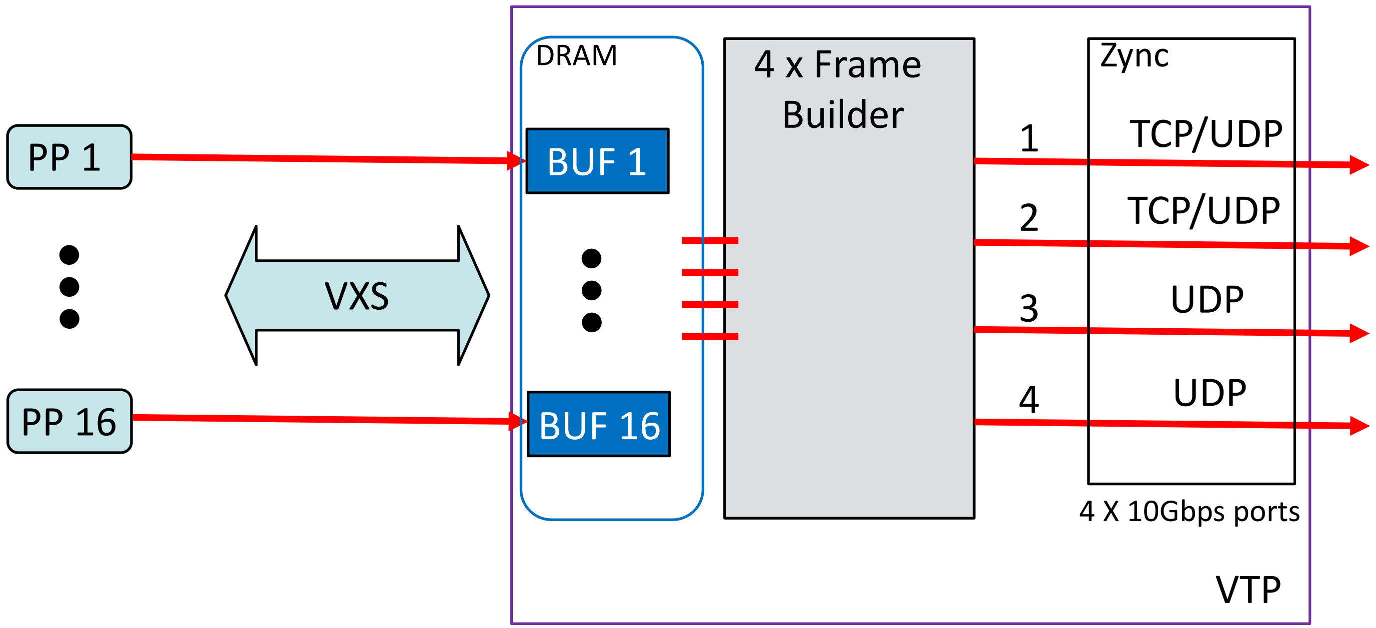


Jefferson Lab

Hardware Accelerated ROC Data transport

- The CODA VTP ROC uses a firmware based "Frame-Builder" for stream data management
- Globally at JLAB CTS "Sync" is used to start and stop all the streams at their source (FADC boards)
- Payload Port FADC boards are currently configured via the VME Based CODA ROC

- Locally, PPs stream hits to a VTP DRAM buffer. The Frame Builders have a frame "fifo"
- A Frame builder will only aggregate hits into a ROC frame if the fifo is not full. Otherwise the frame is dropped. (**This is only relevant for TCP streams**.)
- Built ROC frames at the Zync will always get sent (eventually).
- Up to 4 independent network streams can be defined. Each PP maps to a specific output stream.
- Limited Zync resources only provide 2 high performance TCP connections (or 4 UDP streams)



Note:

TCP Performance

~7.5 Gbps per link without frame drops

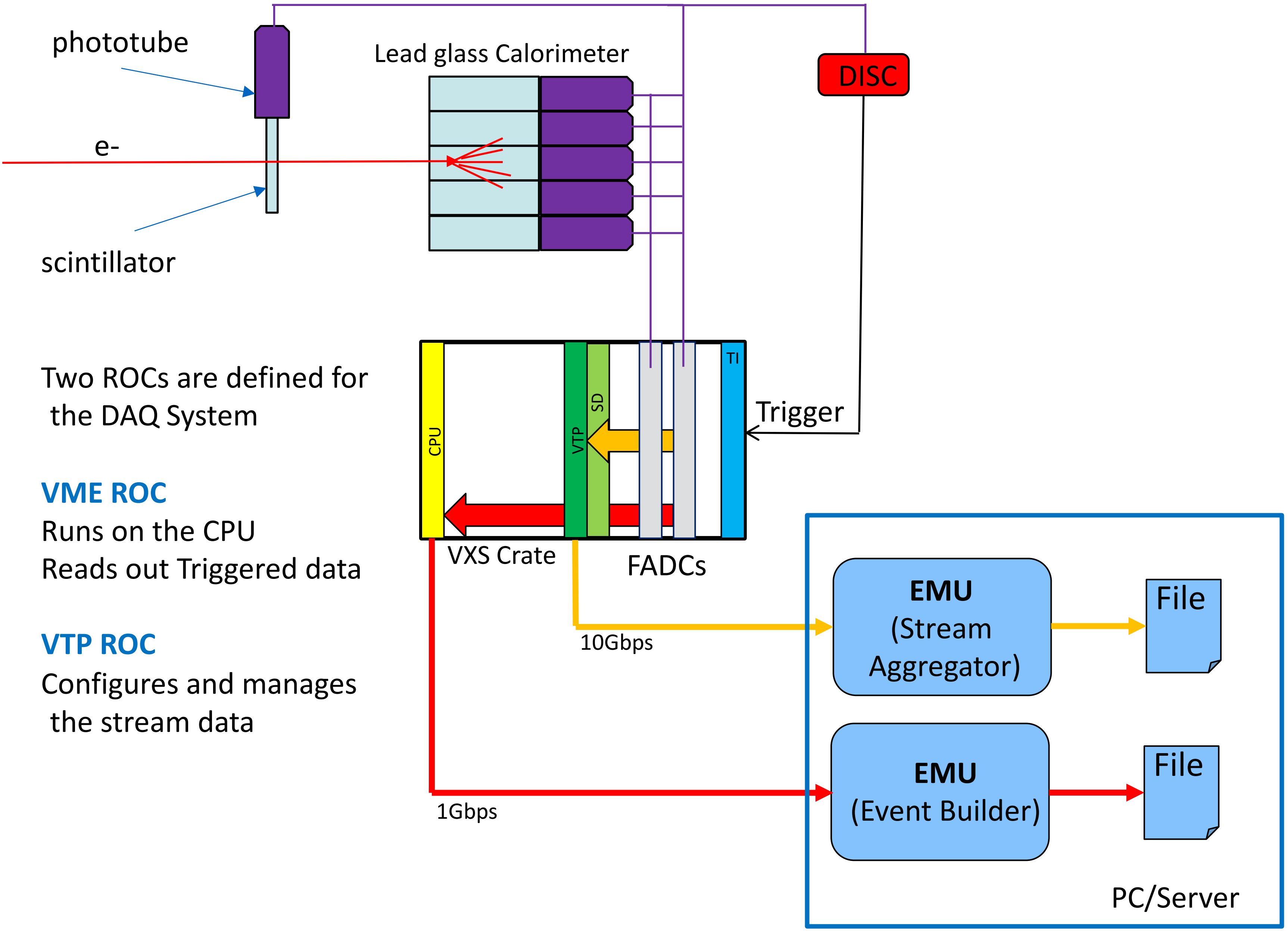
UDP performance (8000 MTU)

>9.5Gbps per link

~50% CPU utilization to read a single stream

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

Simple Hybrid DAQ System



Two ROCs are defined for the DAQ System

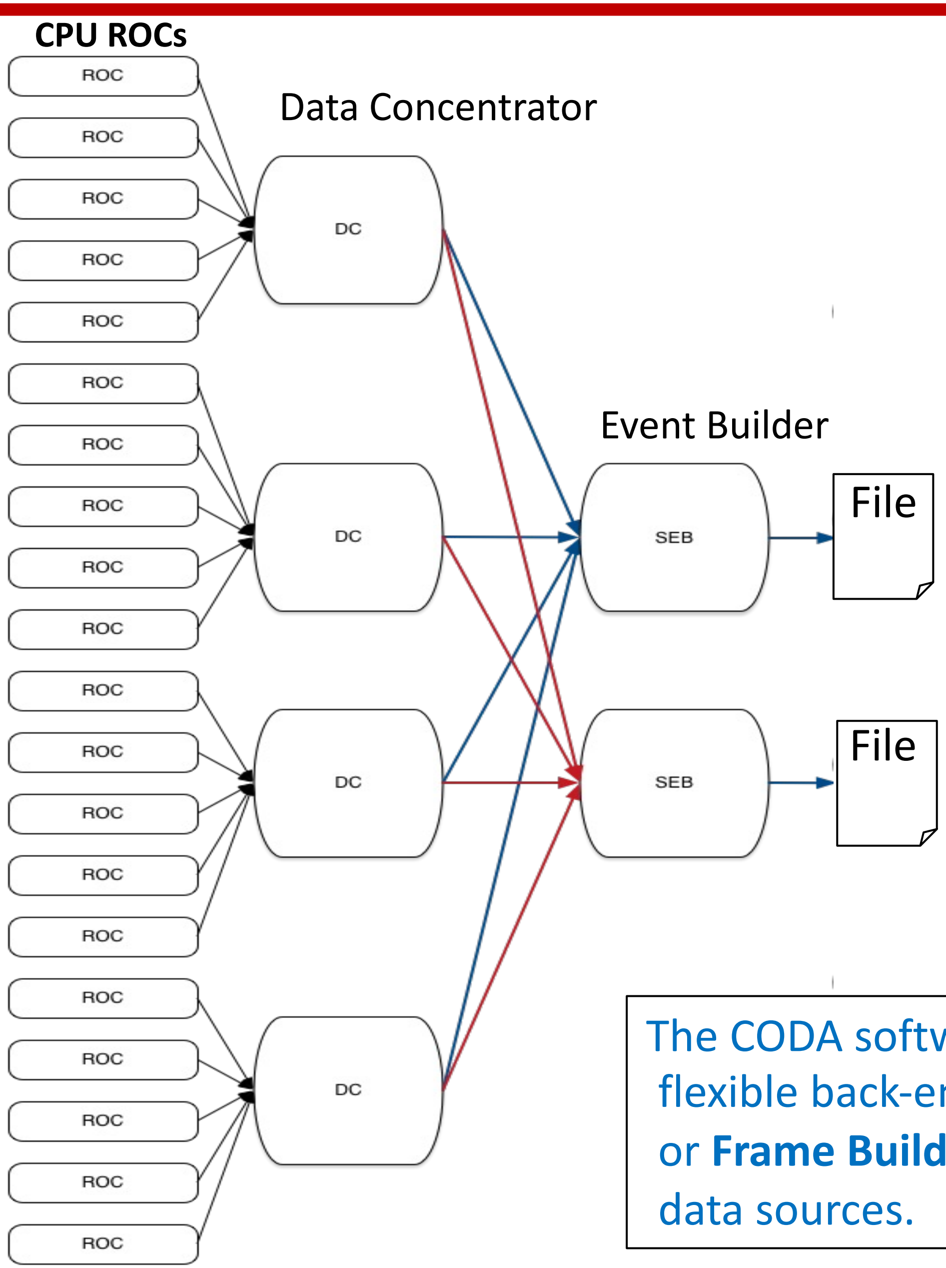
VME ROC
Runs on the CPU
Reads out Triggered data

VTP ROC
Configures and manages the stream data

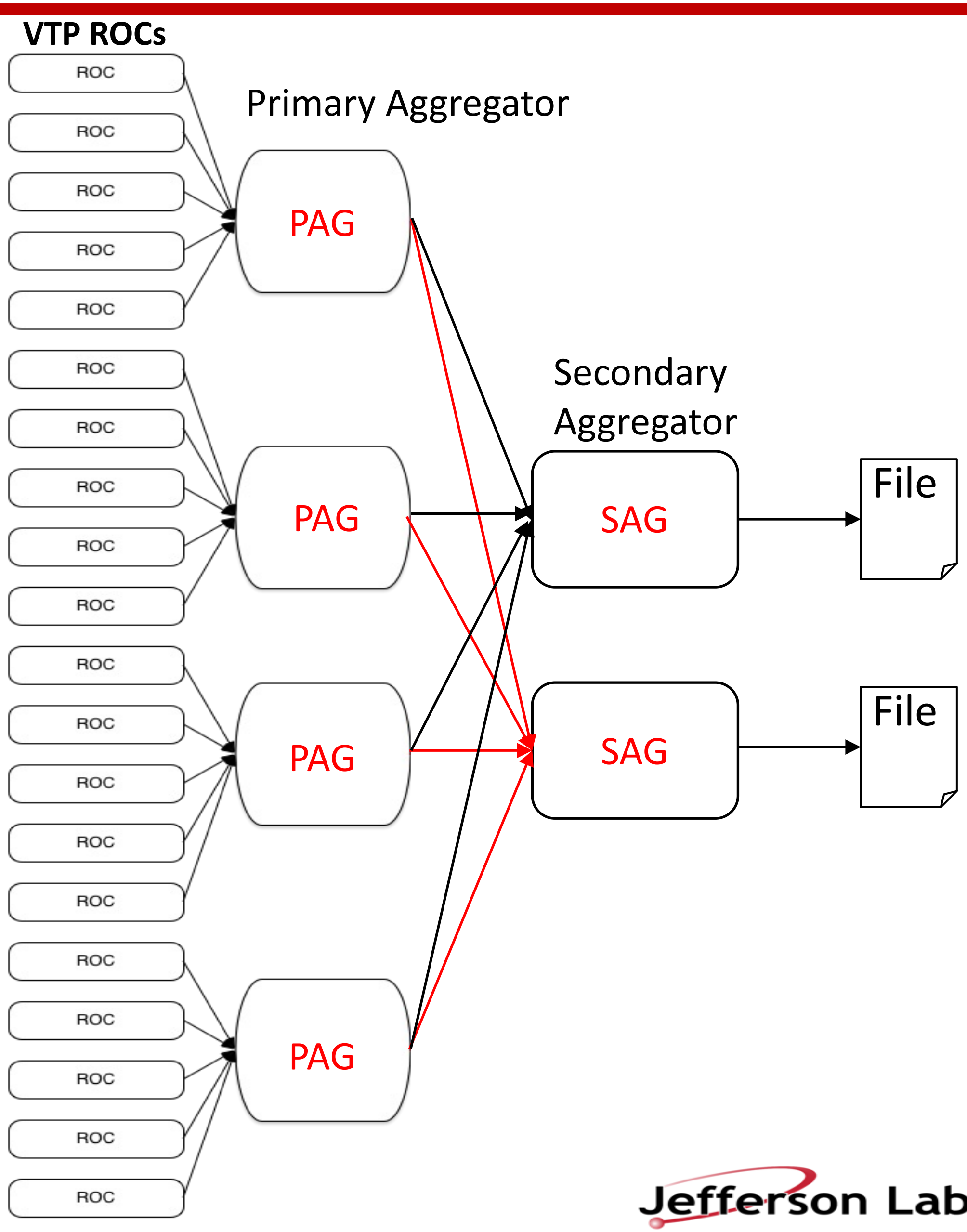
Jefferson Lab

Event/Frame Building

Triggered DAQ



Streaming DAQ



The CODA software toolkit allows flexible back-end **Event Building** or **Frame Building** from many data sources.

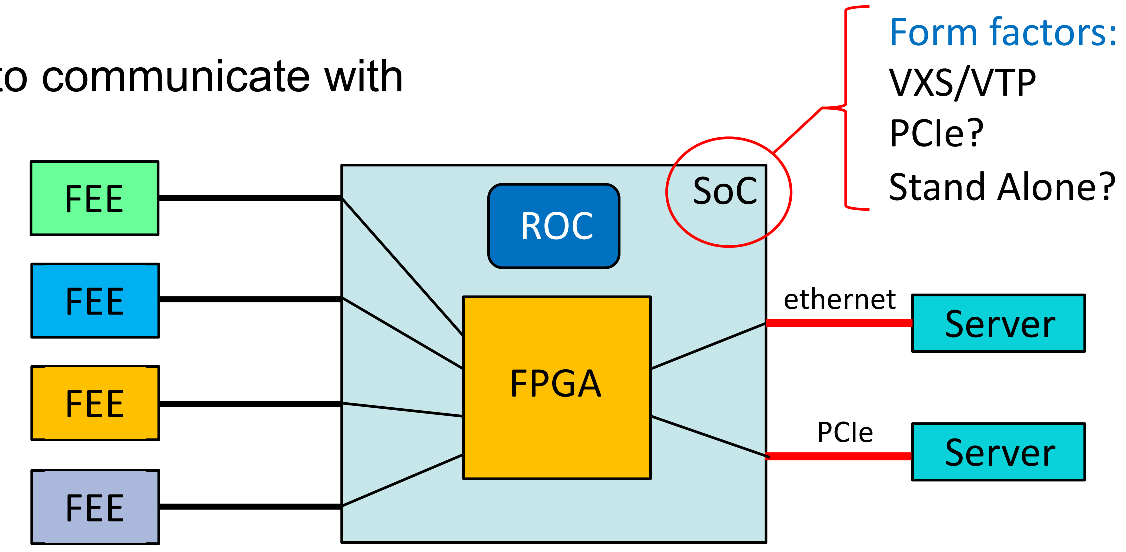
Jefferson Lab

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 for the near term, but how do these solutions evolve for future experiments, in particular the Streaming model.

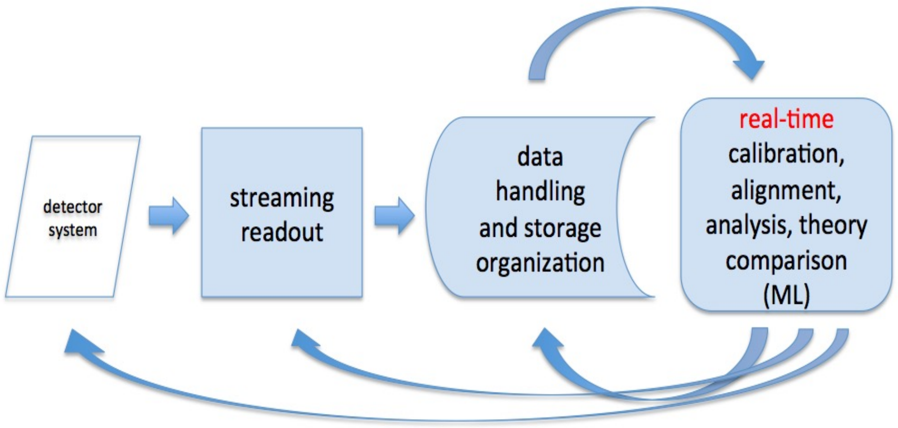
- Managing data “streams” is ideally done in a deterministic way – e.g. FPGAs
- Proprietary communication with the front-end electronics.
  - One Bi-directional link (fiber)
  - Send: clock, commands, control, config
  - Receive: high speed data streams
- System on Chip (SoC) facilitates DAQ/User applications to communicate with the Front-end.
  - Readout, Configuration and Slow Control

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 Back-end processing in a standardized way.



"Intelligent" Routing - EJFAT

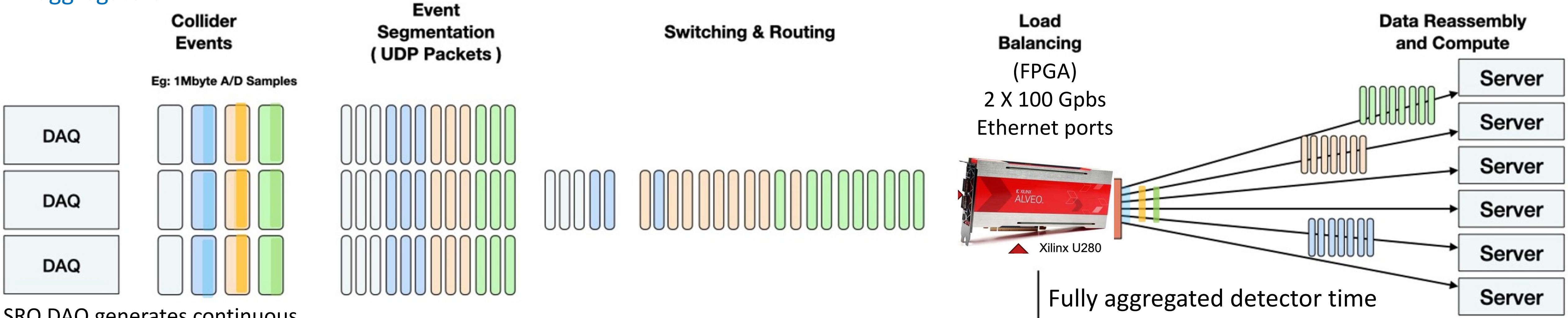
Future experiments are looking to implement an SRO model enabling full offline analysis chains to be ported into real-time…



EJFAT

ESnet-Jefferson Lab FPGA Accelerated Transport

With CODA we are considering implementing the **EJFAT Load Balancer** component as a very efficient frame-builder in place of the software stream aggregators



SRO DAQ generates continuous timestamped data frames from all detector systems

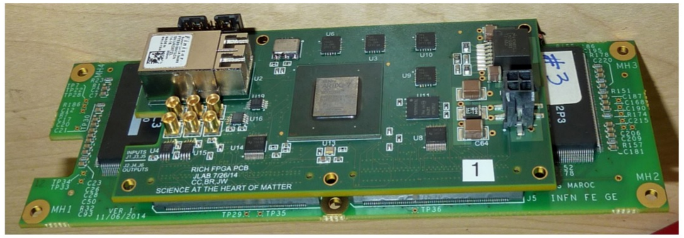
Fully aggregated detector time frames get distributed to available processing

Jefferson Lab

New Detectors - New Front End Electronics

ASIC board produced for the RICH detector for CLAS12 with MAROC chips installed

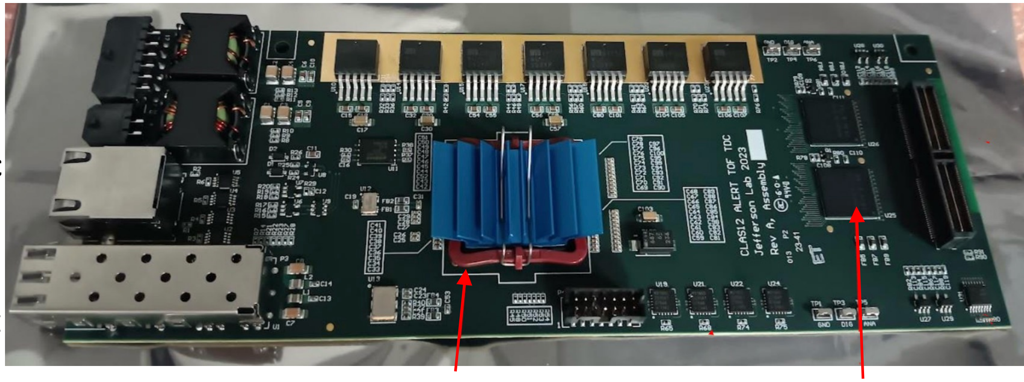
MAROC3 ASICs
64 channel
Fast ”trigger” bits
MAPMT/SiPM readout
Optional 8/10/12bit ADC



FPGA b Screenshot for the RICH detector for CLAS12 with ASIC board attached

Artix FPGA Daughter card
with 1Gb Ethernet port

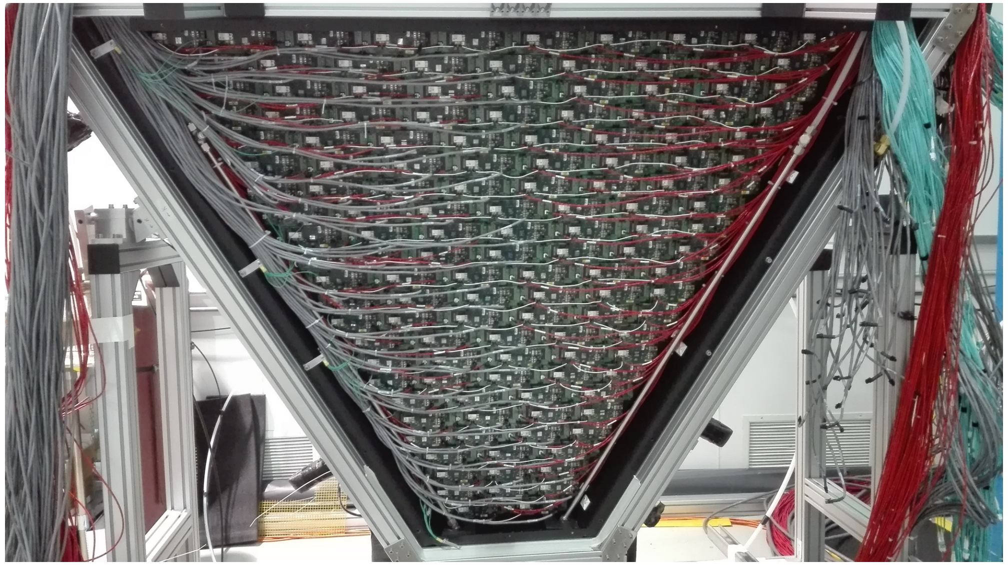
- Used with the new RICH Detector in Hall B and and the DIRC detector in Hall D
- MAROC capable of streaming “Fast” bits but current readout is via a trigger.
- Clock and trigger are provided by copper cable fanouts



- Planned use with upcoming ALERT experiment in Hall B (TOF detector).
- Additional applications in PET readout and image processing with the JLab Detector Group
- The Petiroc ASIC supported both triggered and streaming readout.
- ALERT will use it in triggered mode. Clock, Sync and Trigger will be provided via RJ45 copper fanout.
- PET development is investigating a streaming option.

Jefferson Lab

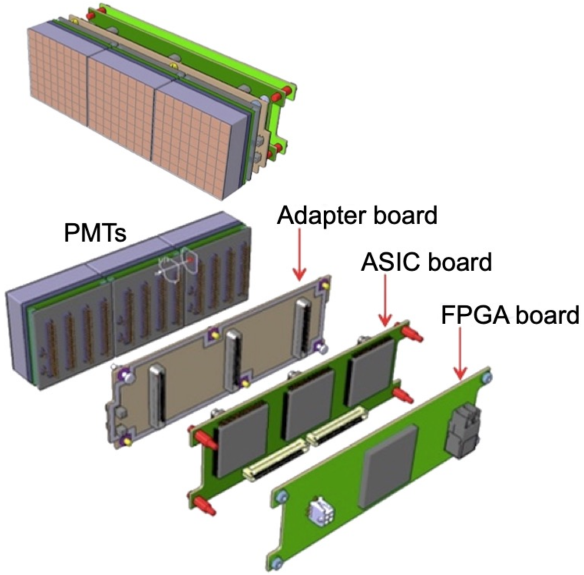
HALL B RICH Detector - Readout

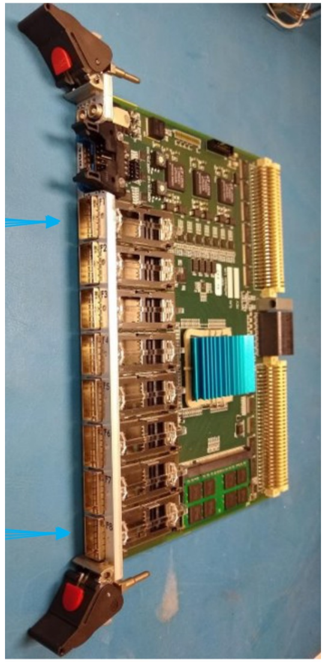


138 Readout Boards (Tiles)
(Blue fiber cables)

ASIC board produced for the RICH detector for CLAS12 with MAROC chips installed

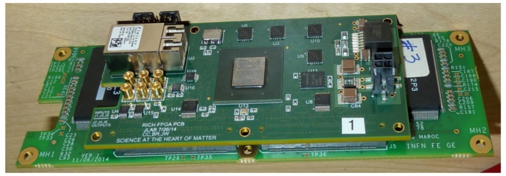
MAPMT Sensors (25024 Channels)





5 SSP FPGA boards aggregate the fibers in a single VXS crate

ROC Readout over VME Bus



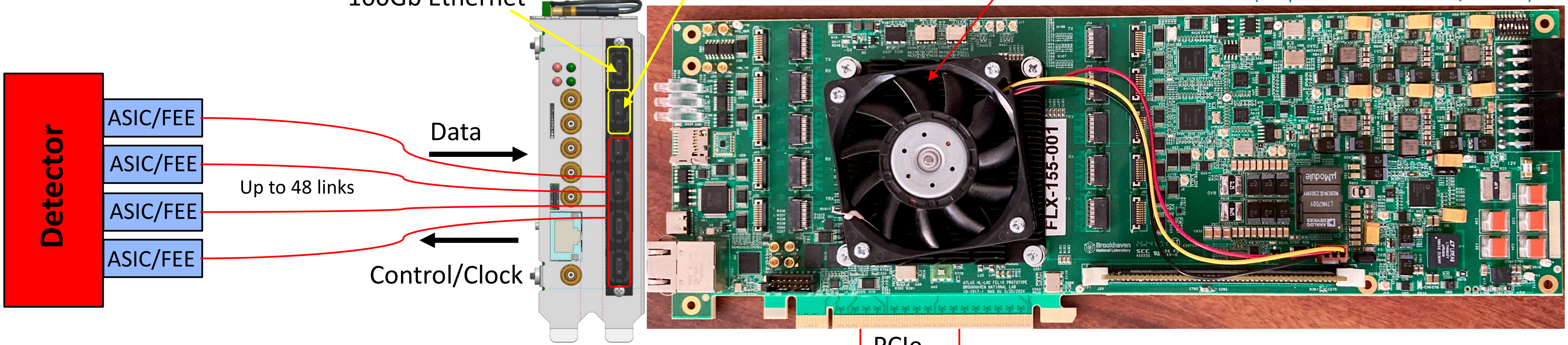
FPGA b Screenshot for the RICH detector for CLAS12 with ASIC board attached

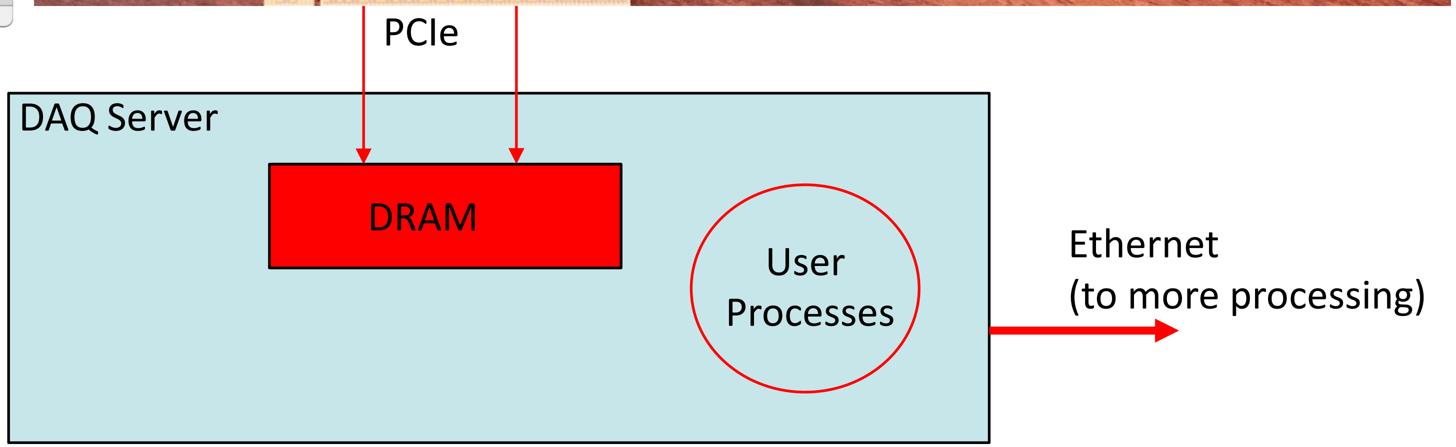
Streaming is possible via the VTP in the future, using the SSP as an intermediate aggregation point.

FELIX Hardware Integration

FELIX: Front-End Link eXchange

Model 155





Currently FLX155 Engineering article is being tested at BNL. (See Hao Xu talk at this workshop)

It is also going to be used for the ePIC Detector DAQ for the new Electron Ion Collider (See Jeff Landgraf talk)

Future plans are to integrate the use of the FELIX board with CODA streaming firmware and software to support more detector resident electronics at JLab.

Summary

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

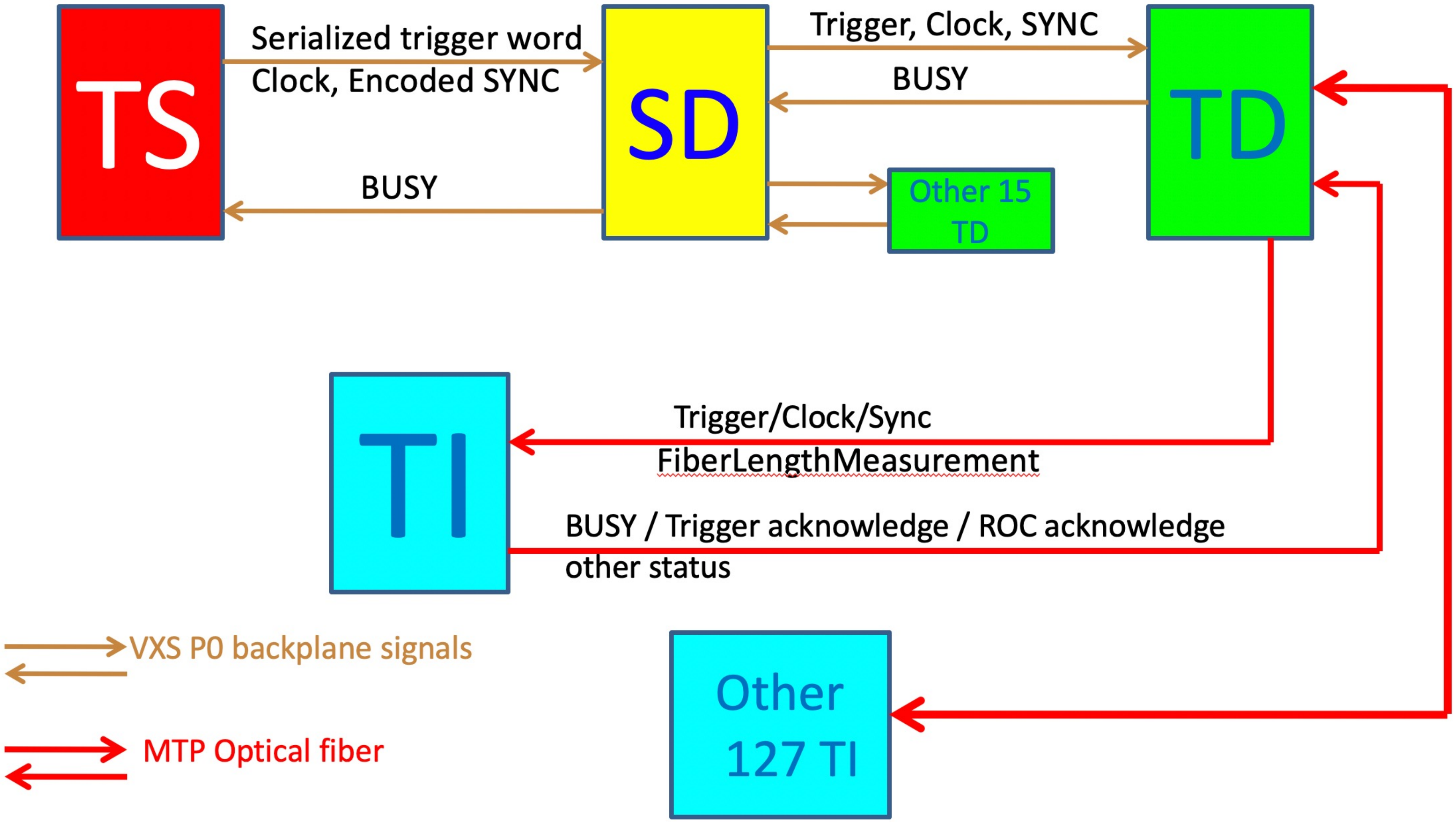
- Transition from the CODA DAQ system's traditional software-based Readout Controller (ROC) to a "hybrid" hardware accelerated application has been successfully implemented.

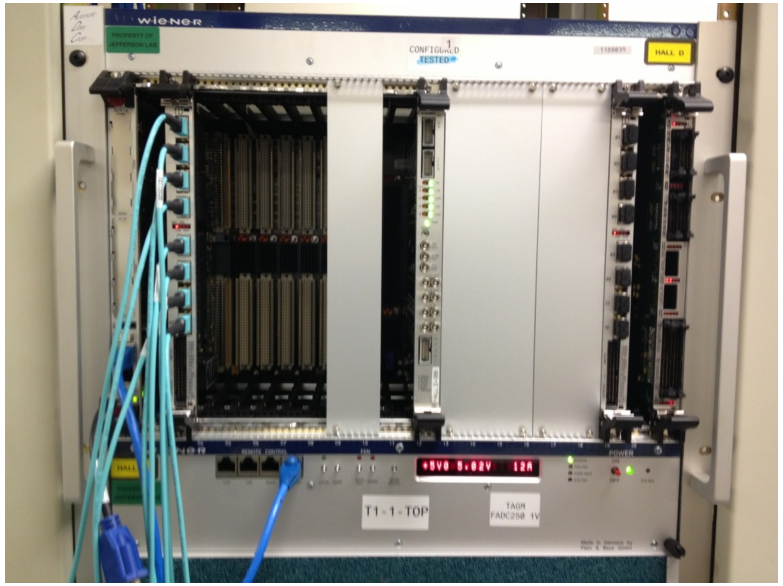
- There is still work ahead will involve making CODA robust against whatever the new front-end electronics may require

- 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

JLab Timing System Components (GTU)





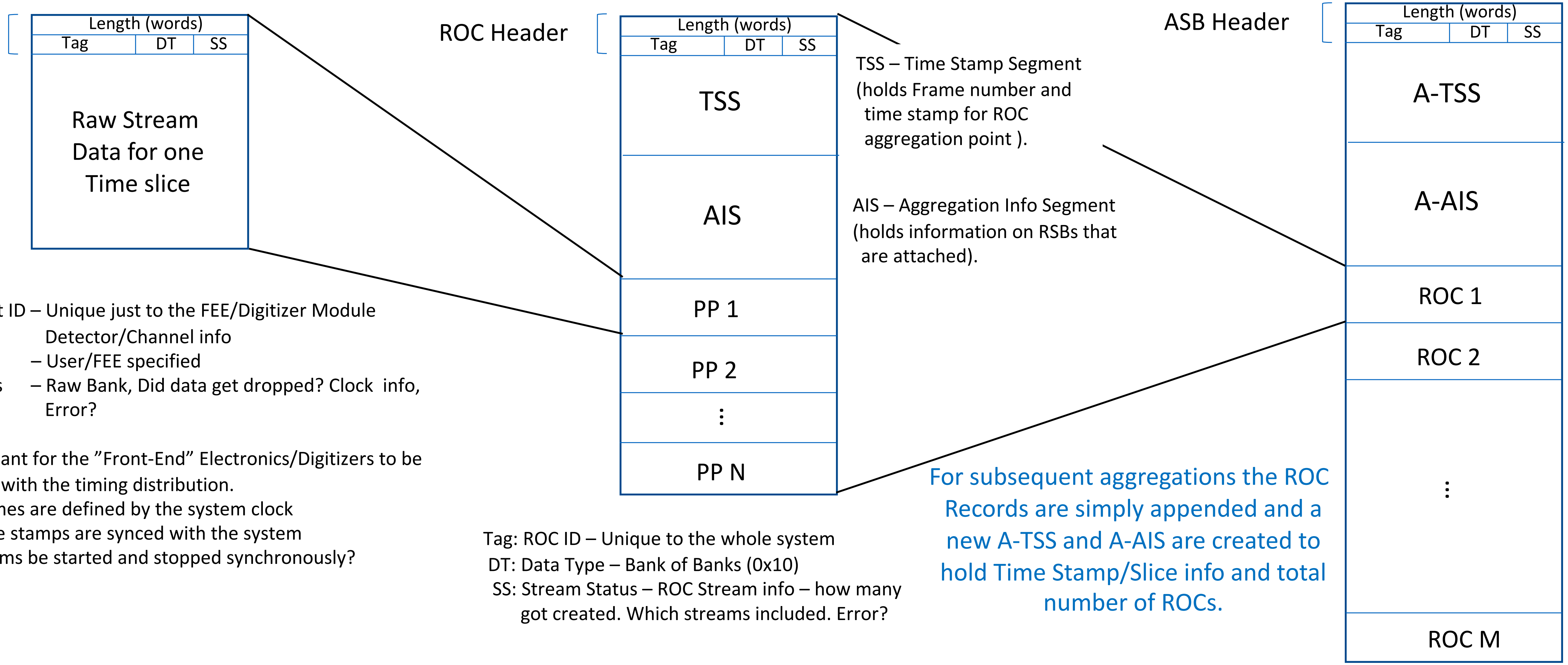
Stream Aggregation – Data formats

Raw Stream Bank (RSB)
(Payload Port – e.g. FADC)

Aggregate Stream Records

1st Aggregation (ROC)

Subsequent Aggregations



TSS – Time Stamp Segment (holds Frame number and time stamp for ROC aggregation point).

AIS – Aggregation Info Segment (holds information on RSBs that are attached).

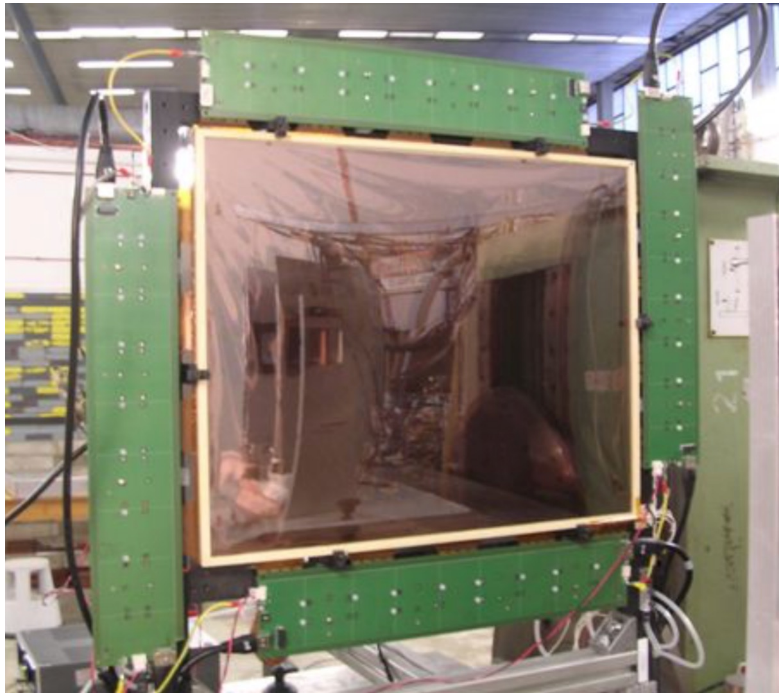
Tag: Payload port ID – Unique just to the FEE/Digitizer Module
Detector/Channel info
DT: Data Type – User/FEE specified
SS: Stream Status – Raw Bank, Did data get dropped? Clock info, Error?

Note: It is important for the ”Front-End” Electronics/Digitizers to be integrated with the timing distribution.
- Data Frames are defined by the system clock
- Local time stamps are synced with the system
- Can streams be started and stopped synchronously?

Tag: ROC ID – Unique to the whole system
DT: Data Type – Bank of Banks (0x10)
SS: Stream Status – ROC Stream info – how many got created. Which streams included. Error?

For subsequent aggregations the ROC Records are simply appended and a new A-TSS and A-AIS are created to hold Time Stamp/Slice info and total number of ROCs.

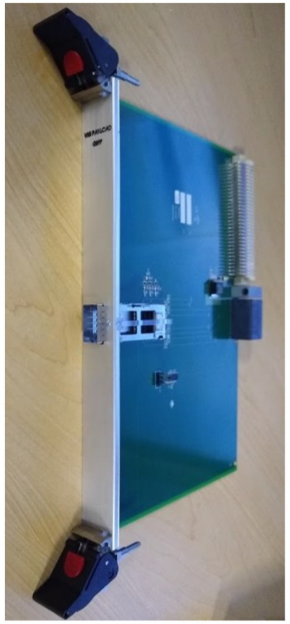
SBS GEM Trackers

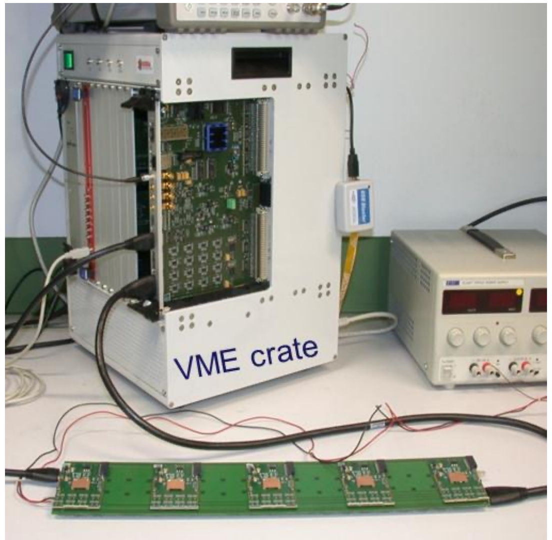


MPD: Multi-Purpose Digitizer Board

Used to manage and digitize signals from the 128 channel APV25 ASIC Front-End Electronics boards.

MPD supports both VME readout as well as a front-panel serial link.





- Not practical to keep MPD boards in a VME/VXS crate and readout over VME
  - MPDs needed to be closer to the GEM detector
  - Total data rates generated by GEMs could not be handled by VME bandwidth
- Use a remote serial link interface with the VXS Crate and a VTP (4 MPDs per slot)
  - MPD configuration and control as well as high speed data
  - Clock and trigger still sent via copper cable fanouts

Serial Protocol (VTP– MPD)

- Typically 1.25Gbps to 10Gbps per lane, bi-directional
- Fiber connected front-ends typically 1 lane (up to 10Gbps per remote front-end module)
- VXS backplane connected front-ends typically 4 lanes (up to 20Gbps per local front-end module)
- Custom protocols implemented:
  - Completely custom when fixed latency low-jitter links needed
  - Supports remote front-end register read/write access, event data transport (and optionally trigger data and fixed latency trigger/sync distribution)
  - When fixed latency isn't required we typically use Xilinx Aurora framing procotol (with custom frame formats). Aurora is a light-weight protocol to establish links (bond and align data) and support streaming and framing data.
- e.g. VTP <-> MPD protocol:
  - 1.25Gbps multi-mode optical, 8b10b encoded. After the deserialization on either end (VTP or MPD) it looks like a 16bit wide bus @ 62.5MHz (1Gbps) for transmit and received.