Streaming Readout Development for CODA @ JLab

Streaming Readout X – May 17-19, 2022

Hybrid Meeting

David Abbott

Ben Raydo

FEDAQ Group

Jefferson Lab – Physics Division

Data Acquisition at JLab

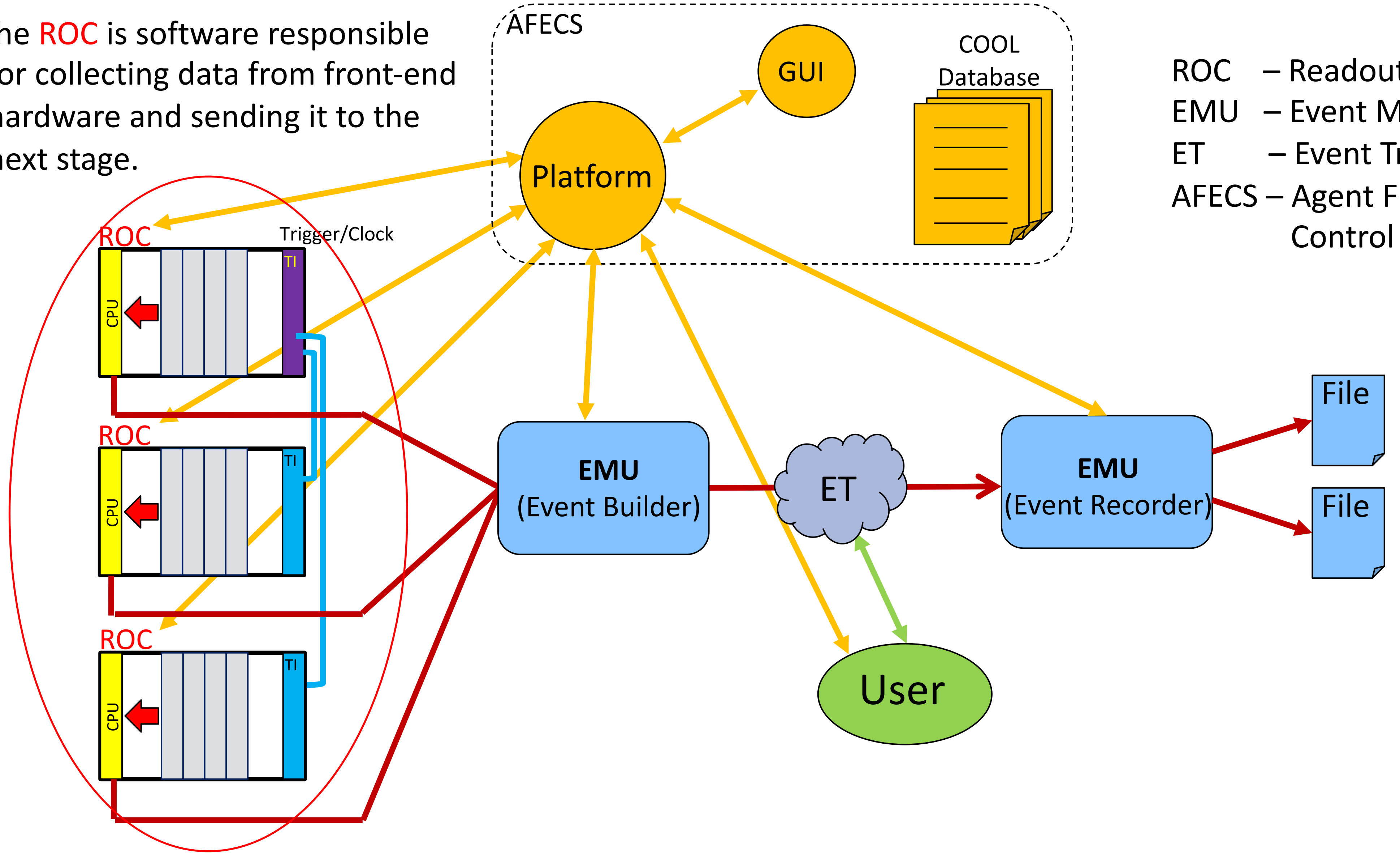
- At JLab we have 4 Experimental Halls, all running with different detectors - and physics priorities.
  - Of course all are having increased demands for the DAQ.

- 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.



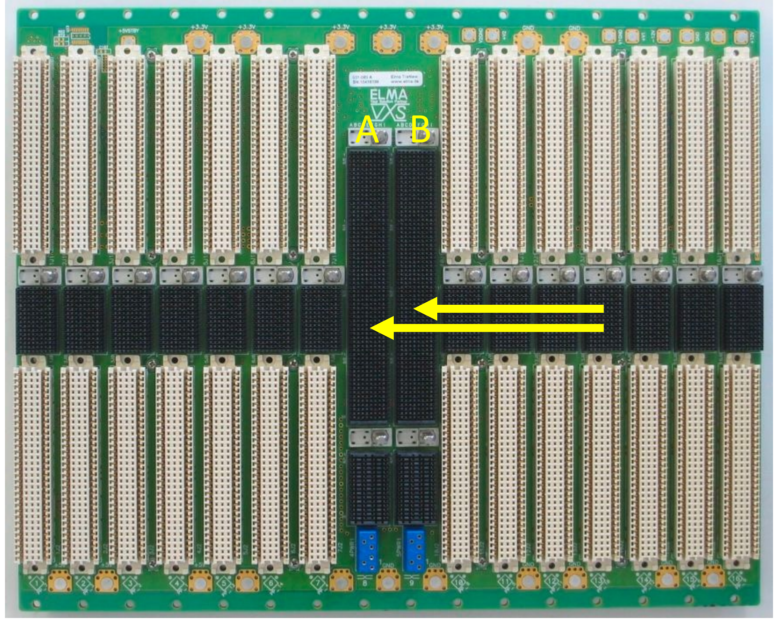
ROC – Readout Controller

EMU – Event Management Unit

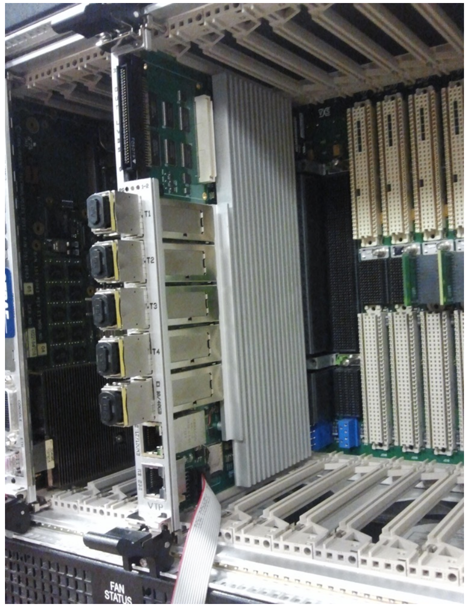
ET – Event Transport (Shared memory)

AFECS – Agent Framework Experiment Control System

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.



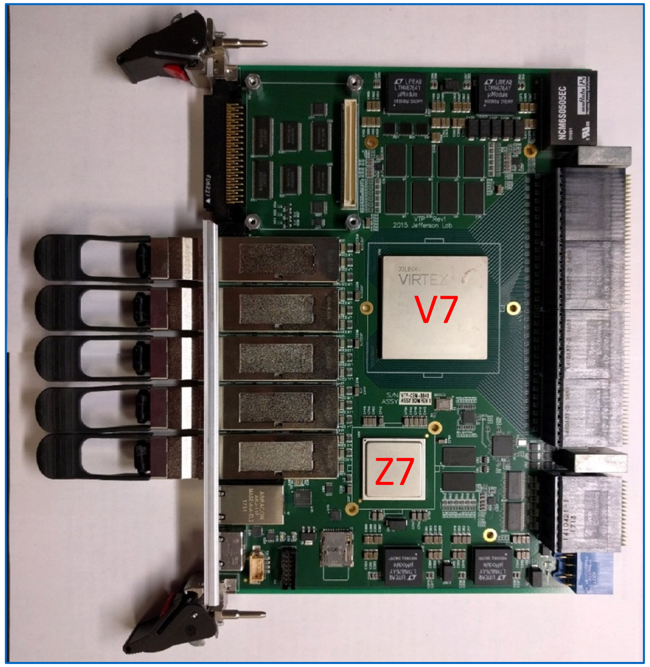
VXS Trigger Processor (VTP)

- 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 ROC now is primarily responsible only for Configure, Control and Monitoring the VTP-Based DAQ.

JLAB – VTP Board

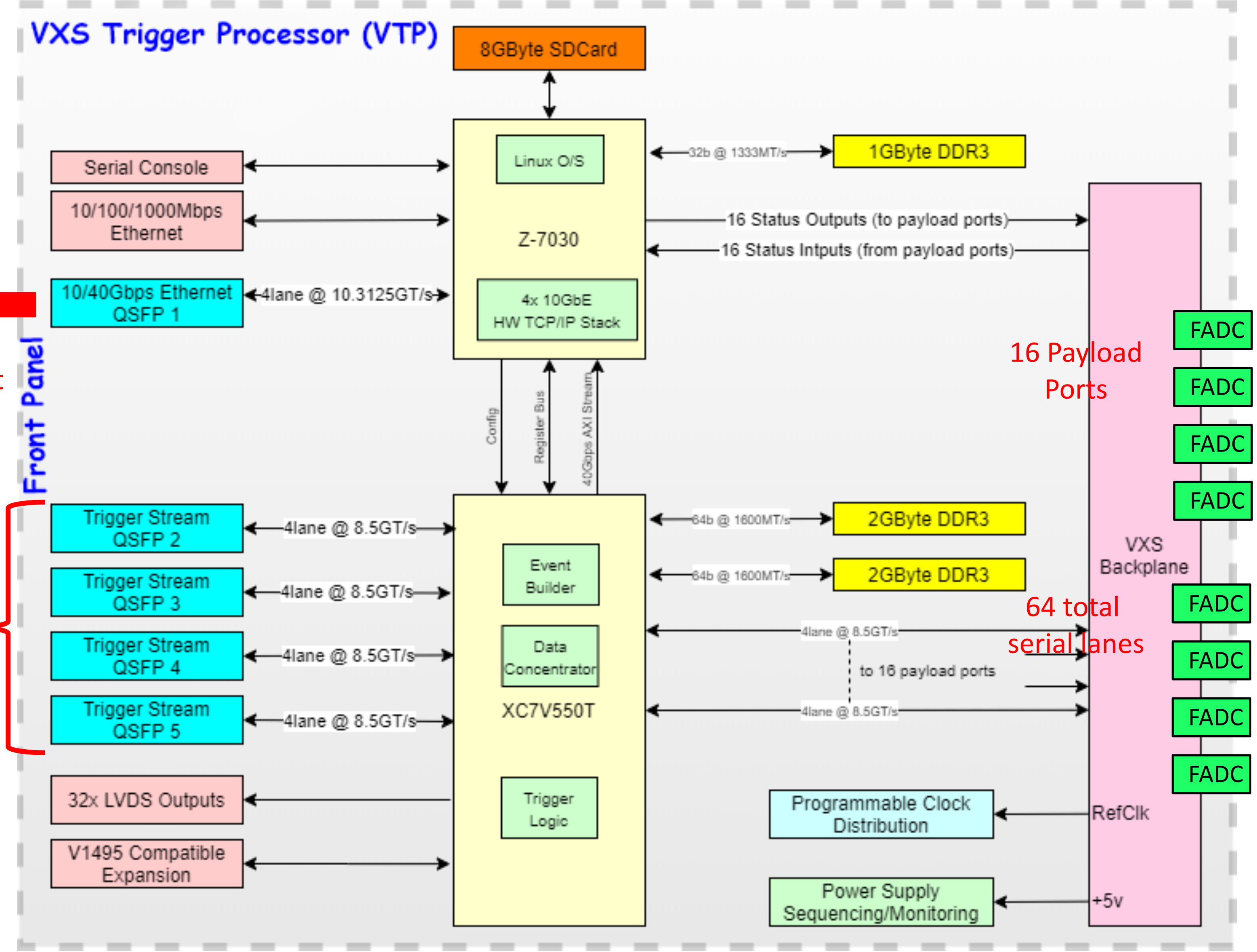
Linux OS on the Zync-7030 SoC
(2-core ARM 7L , 1GB DDR3)
10/40Gbps Ethernet option
(runs the CODA ROC)

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



VXS Trigger Processor (VTP)

Front Panel

8GByte SDCard

Serial Console

10/100/1000Mbps Ethernet

10/40Gbps Ethernet QSFP 1 ← 4lane @ 10.3125GT/s →

Linux O/S

Z-7030

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 ← 4lane @ 8.5GT/s →

Trigger Stream QSFP 3 ← 4lane @ 8.5GT/s →

Trigger Stream QSFP 4 ← 4lane @ 8.5GT/s →

Trigger Stream QSFP 5 ← 4lane @ 8.5GT/s →

32x LVDS Outputs

V1495 Compatible Expansion

Event Builder

Data Concentrator

XC7V550T

Trigger Logic

← 64b @ 1600MT/s → 2GByte DDR3

← 64b @ 1600MT/s → 2GByte DDR3

← 4lane @ 8.5GT/s →

to 16 payload ports

← 4lane @ 8.5GT/s →

Programmable Clock Distribution ← RefClk

Power Supply Sequencing/Monitoring ← +5v

VXS Backplane

FADC
FADC
FADC
FADC
FADC
FADC
FADC
FADC

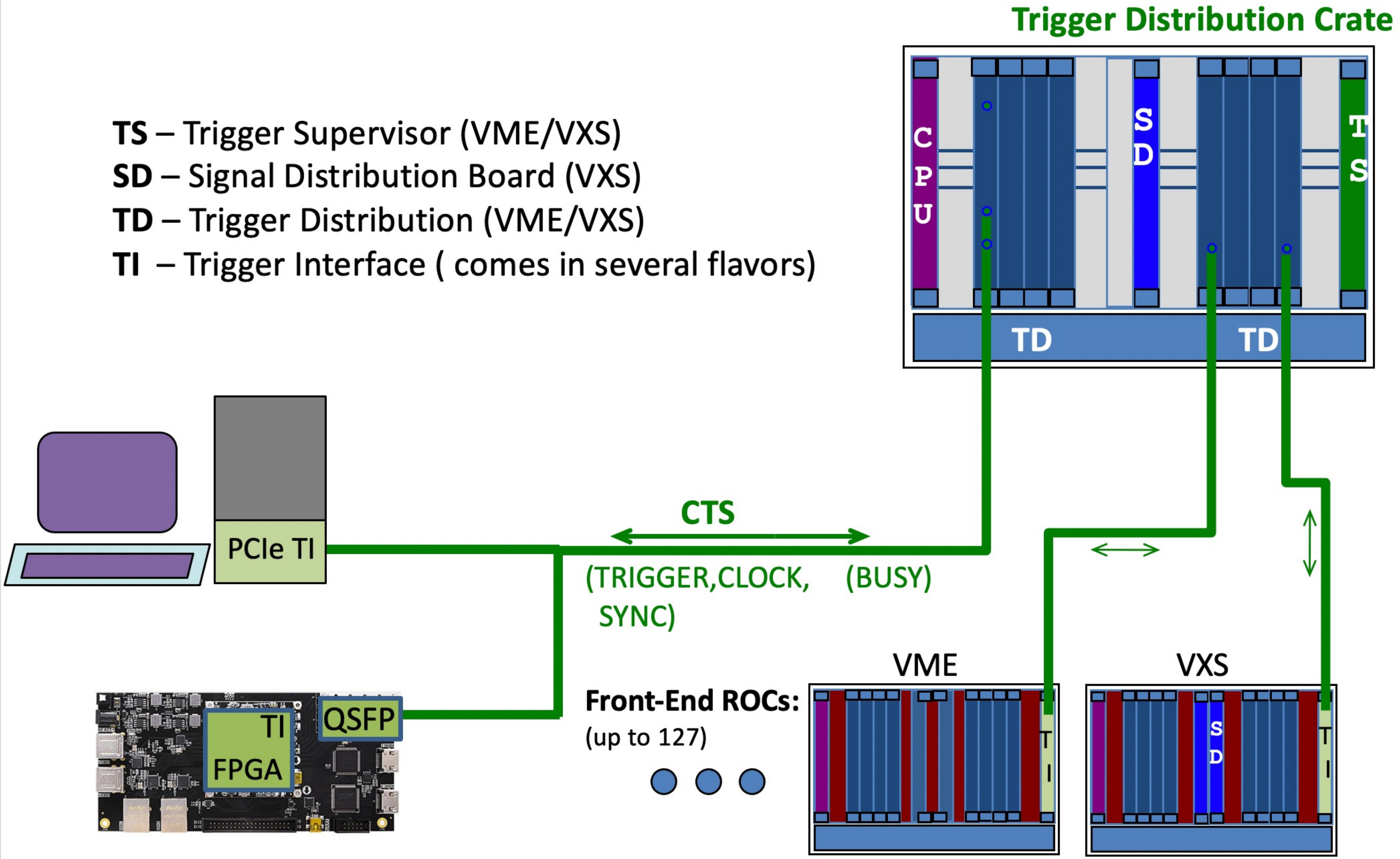
16 Payload Ports

64 total serial lanes

Jefferson Lab

JLAB Clock and Trigger Distribution System

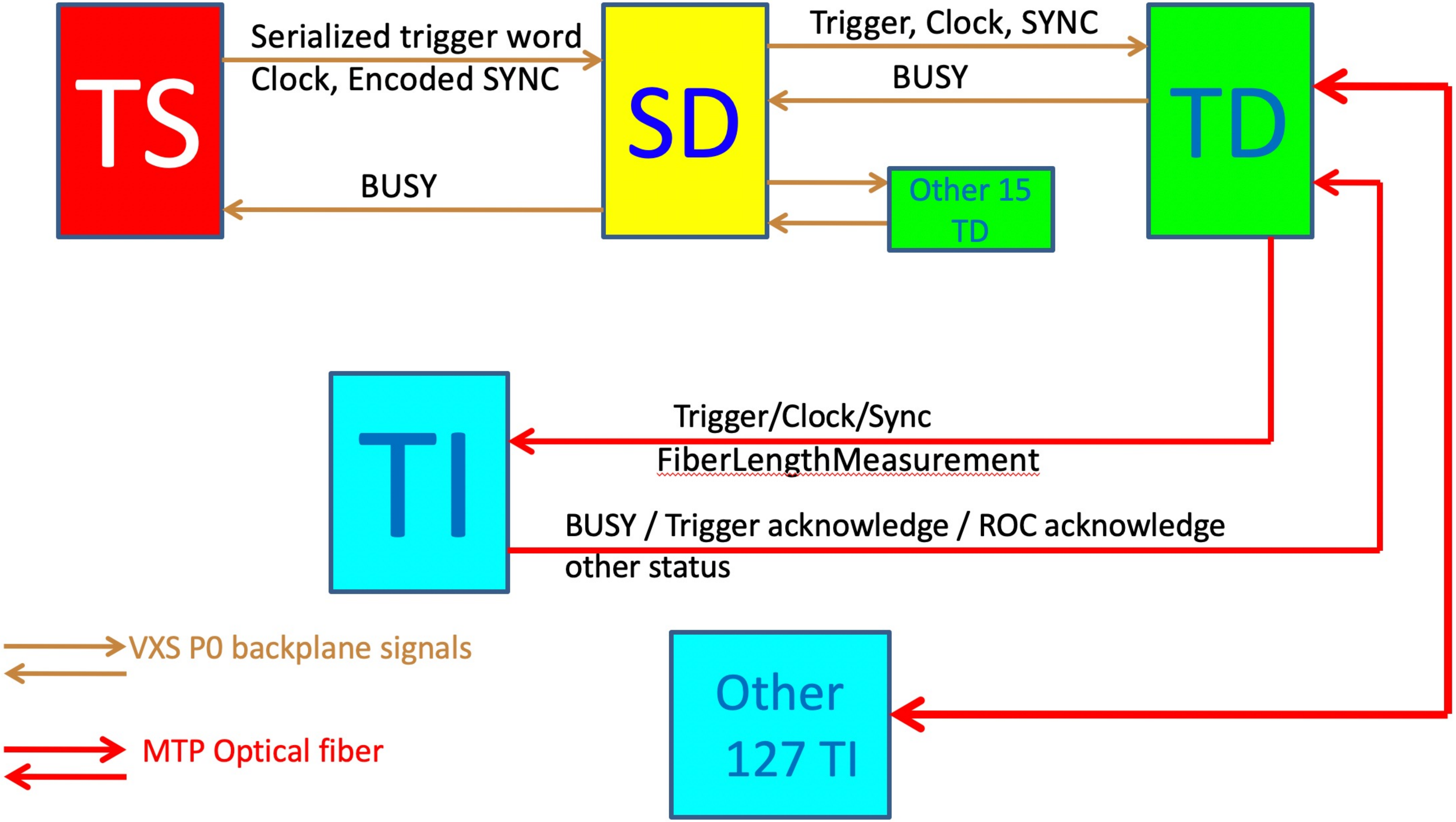
TS – Trigger Supervisor (VME/VXS)
SD – Signal Distribution Board (VXS)
TD – Trigger Distribution (VME/VXS)
TI – Trigger Interface (comes in several flavors)



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

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

Timing System Components

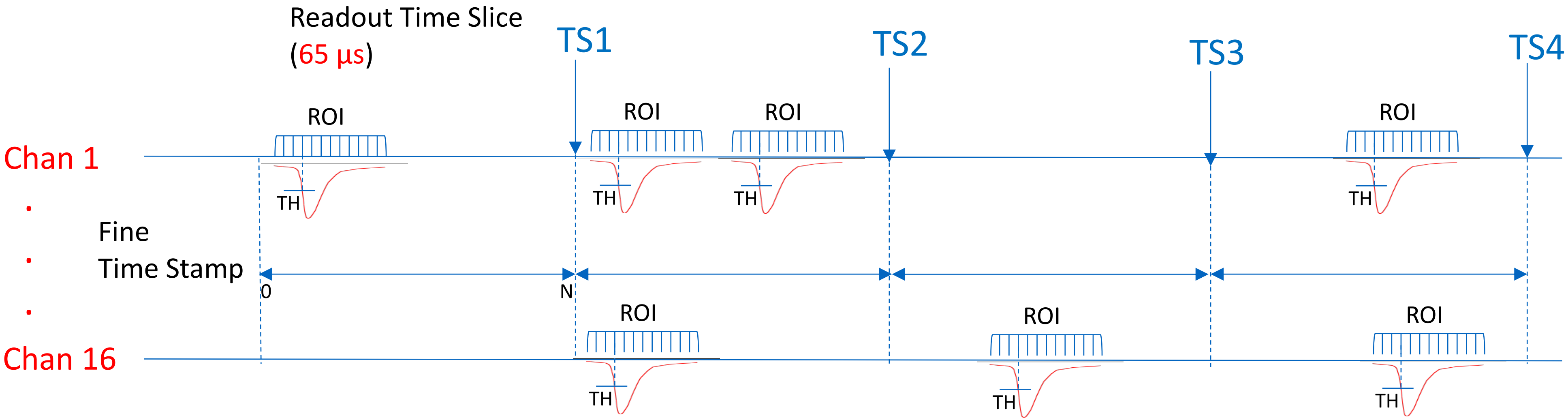


Jefferson Lab

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

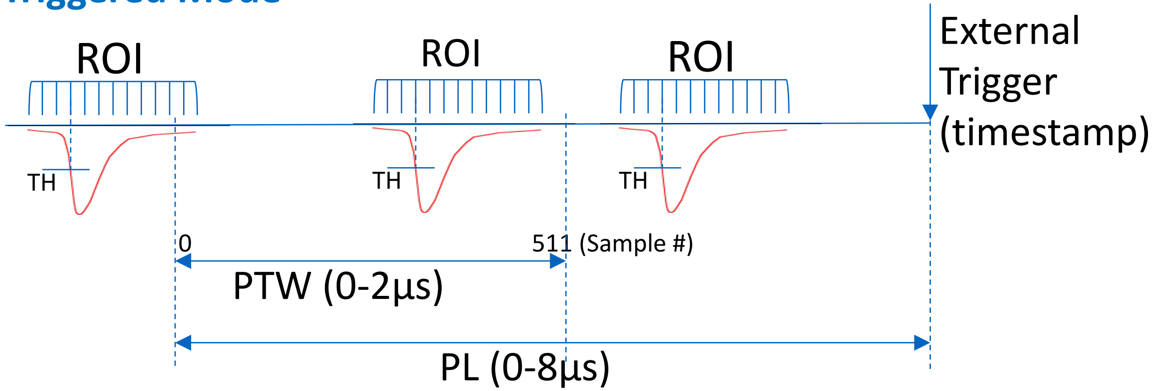


The next revision to the firmware will have an option for full ROI wave forms to be streamed, but this will allow possible dropped hits due to bandwidth limitations

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

FADCs - Triggered vs Streaming

Triggered Mode



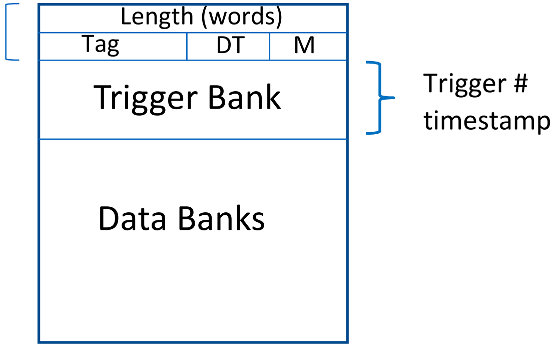
PL: Programmed Lookback

PTW: Time window

Data we get on a trigger:

- FADC waveform values for the ROI
- Threshold Sample # (hit time)
- Trigger absolute time stamp

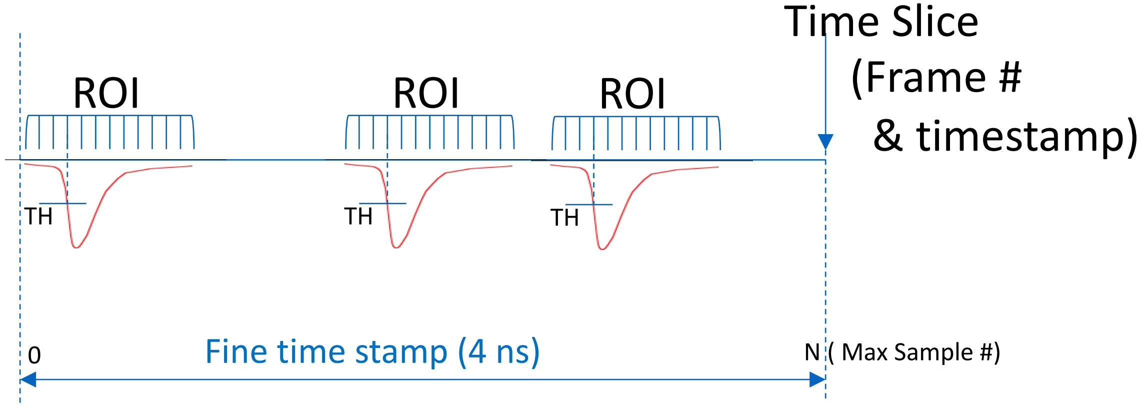
ROC Data Format



Length (words)		
Tag	DT	M
Trigger Bank		
Data Banks		

Trigger #
timestamp

Streaming Mode

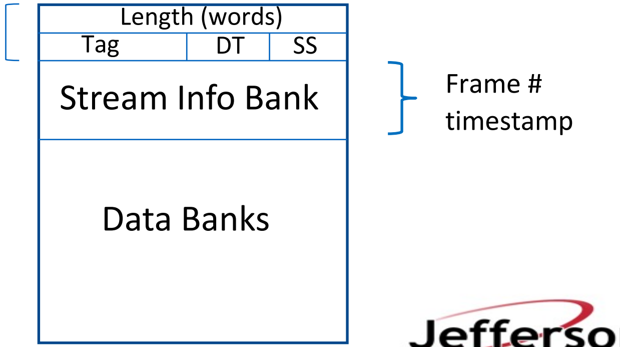


1 Frame = N Clocks (up to 16bits, currently 65536 ns)

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



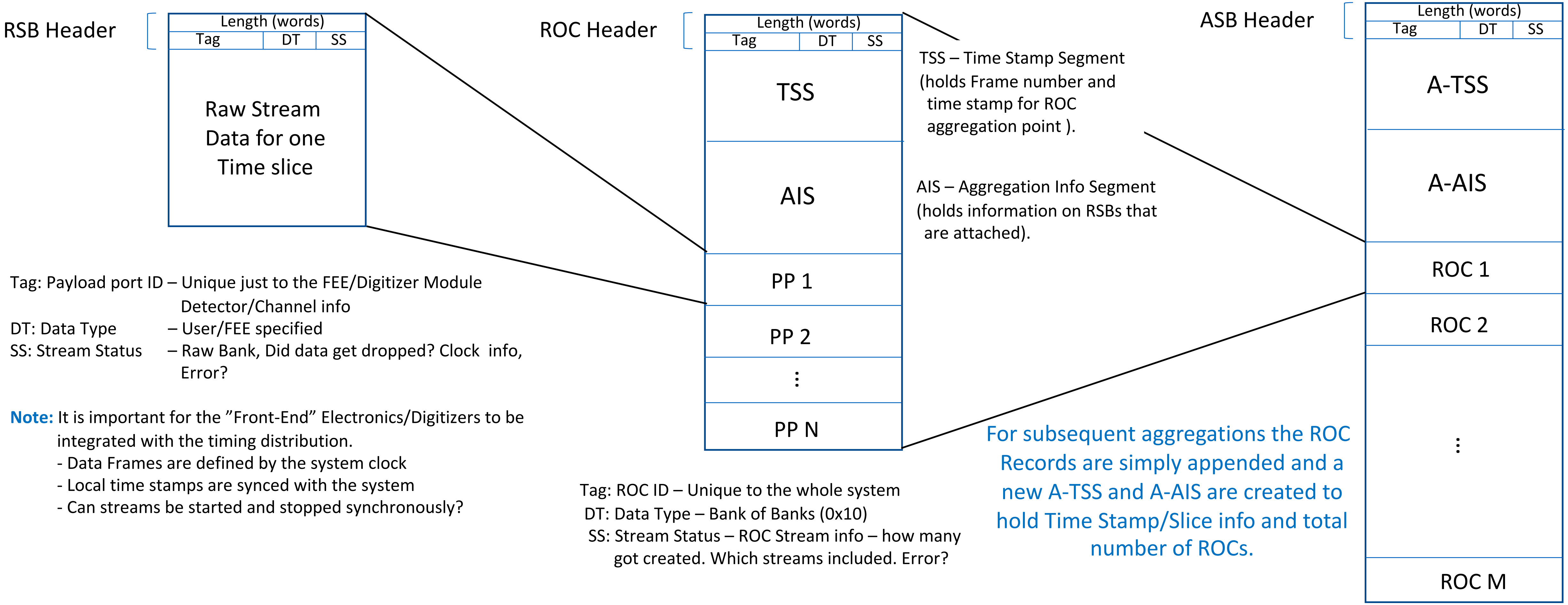
Length (words)		
Tag	DT	SS
Stream Info Bank		
Data Banks		

Frame #
timestamp

Stream Aggregation – Data formats

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

Aggregate Stream Records



1st Aggregation (ROC)

Subsequent Aggregations

RSB Header

Length (words)		
Tag	DT	SS

Raw Stream Data for one Time slice

ROC Header

Length (words)		
Tag	DT	SS

TSS

AIS

PP 1

PP 2

⋮

PP N

ASB Header

Length (words)		
Tag	DT	SS

A-TSS

A-AIS

ROC 1

ROC 2

⋮

ROC M

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?

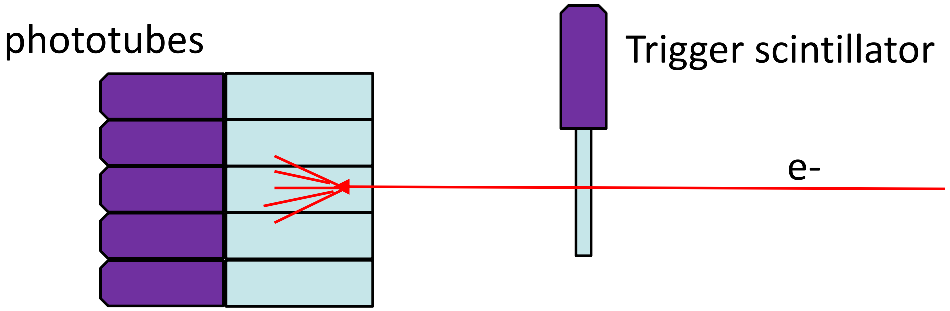
For subsequent aggregations the ROC Records are simply appended and a new A-TSS and A-AIS are created to hold Time Stamp/Slice info and total number of ROCs.

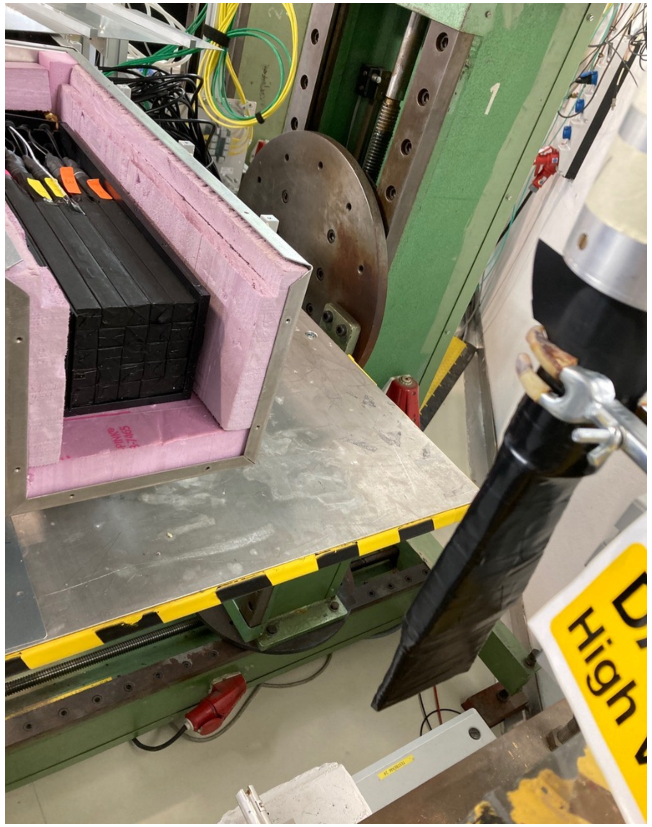
Beam Tests

- 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.





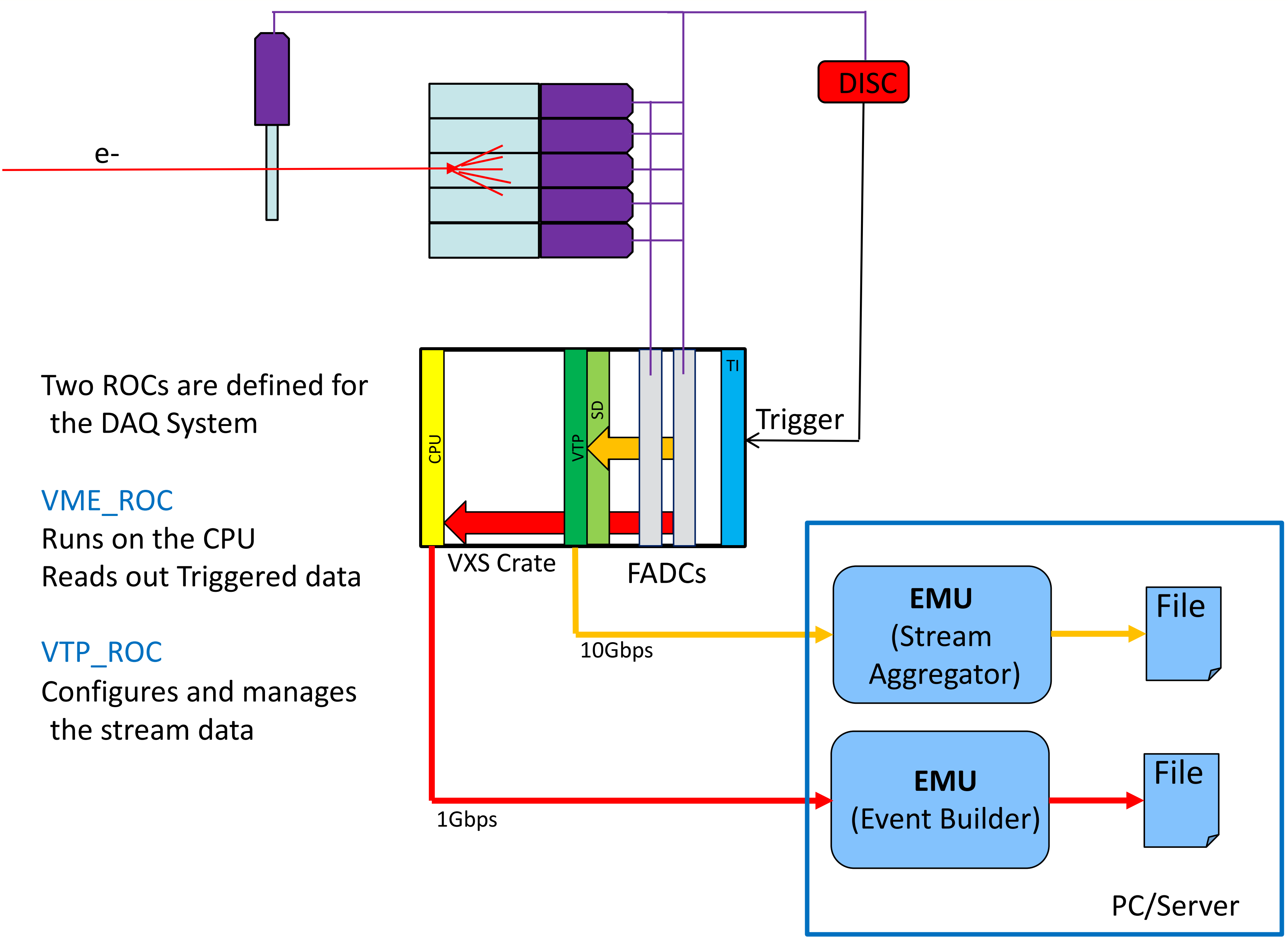
Jefferson Lab

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

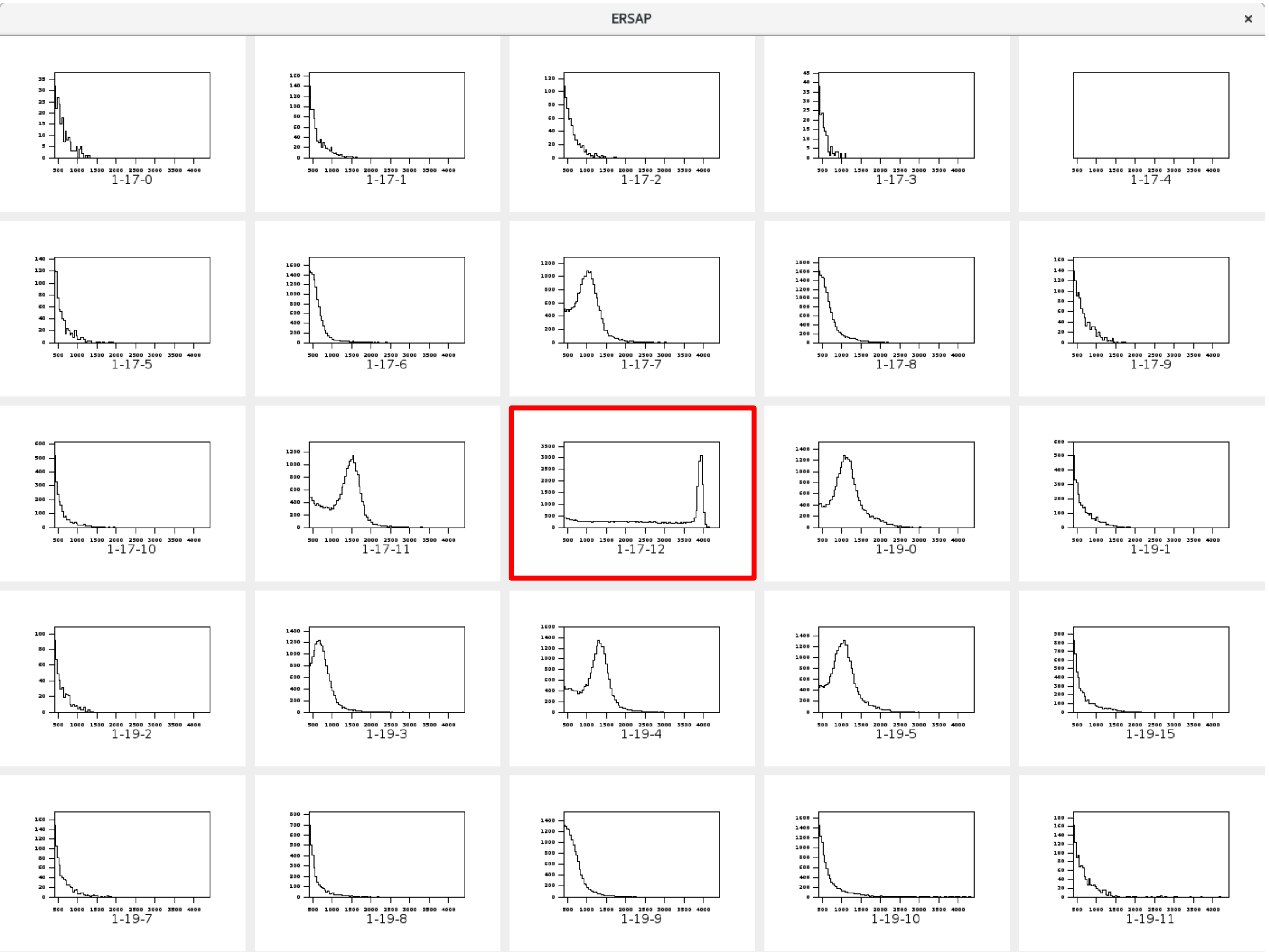


Jefferson Lab

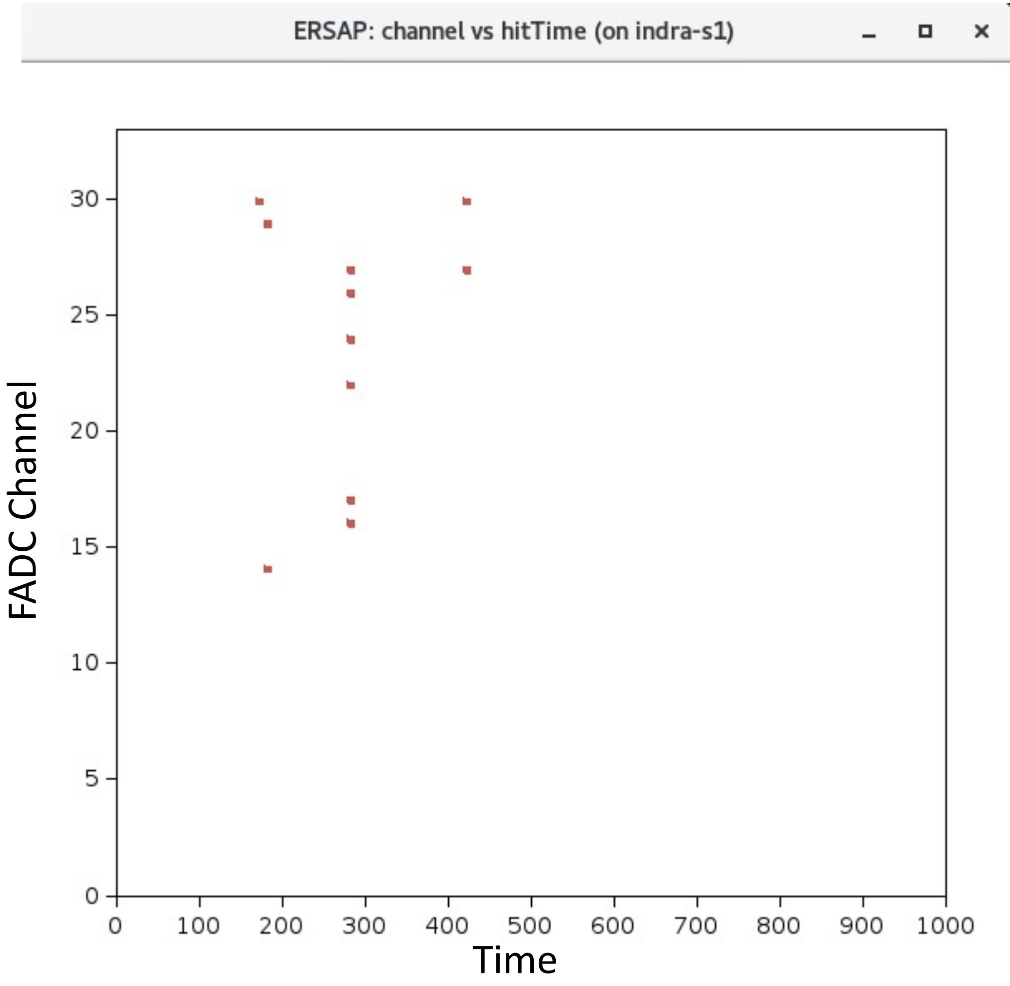
Beam Tests

Calorimeter spectra – Streaming Data

(Electron beam centered on the central crystal)



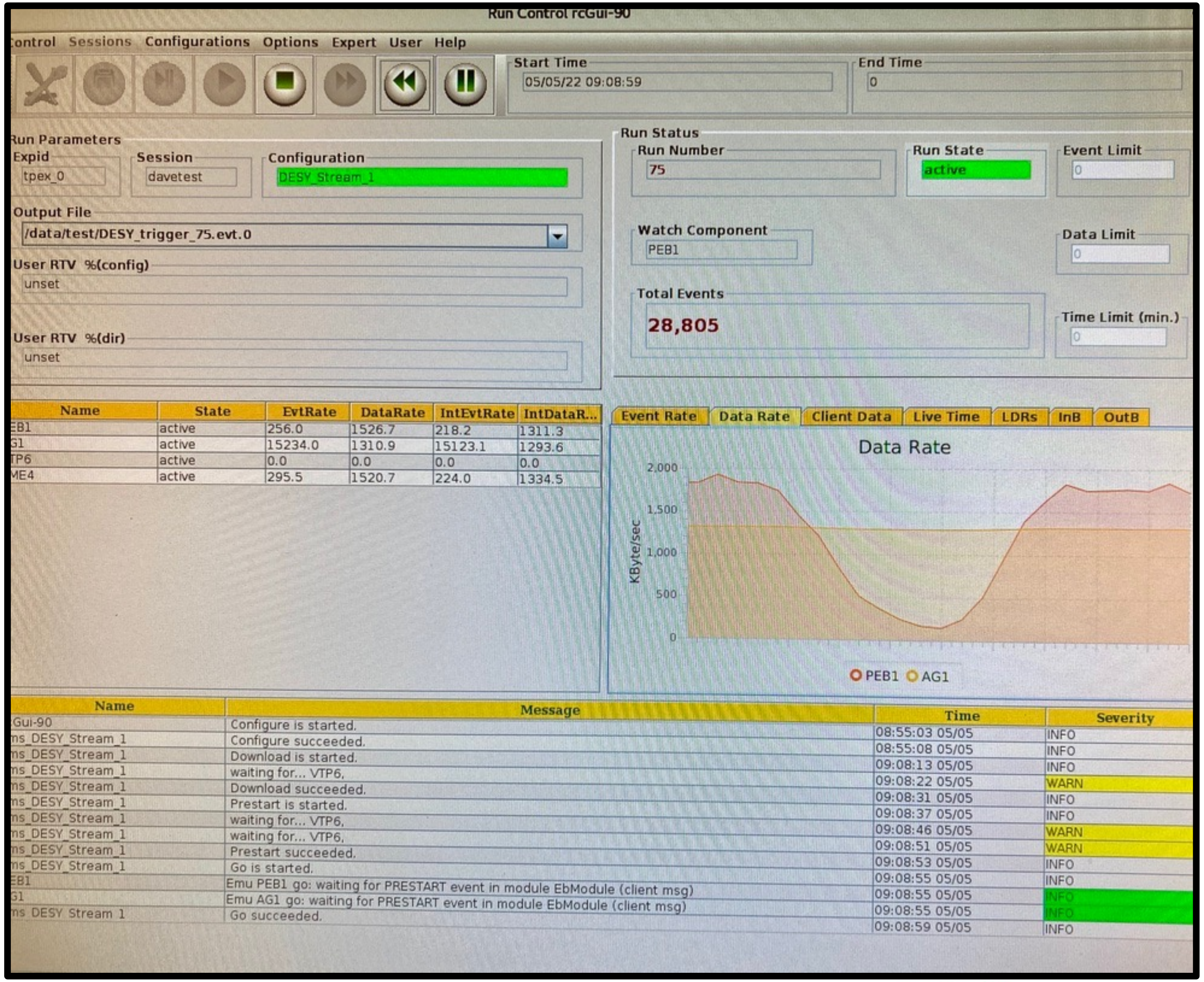
Time Slice Frame containing a Trigger



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)



Start Time: 05/05/22 09:08:59

End Time: 0

Run Parameters

Expid: tpex_0

Session: davetest

Configuration: DESY_Stream_1

Output File: /data/test/DESY_trigger_75.evt.0

User RTV %(config): unset

User RTV %(dir): unset

Run Status

Run Number: 75

Run State: active

Event Limit: 0

Watch Component: PEB1

Data Limit: 0

Total Events: 28,805

Time Limit (min.): 0

Name	State	EvtRate	DataRate	IntEvtRate	IntDataR...
EB1	active	256.0	1526.7	218.2	1311.3
G1	active	15234.0	1310.9	15123.1	1293.6
TP6	active	0.0	0.0	0.0	0.0
ME4	active	295.5	1520.7	224.0	1334.5

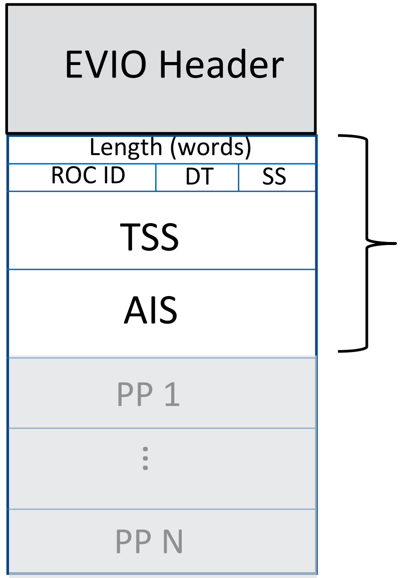
Name	Message	Time	Severity
Gui-90	Configure is started.	08:55:03 05/05	INFO
ns_DESY_Stream_1	Configure succeeded.	08:55:08 05/05	INFO
ns_DESY_Stream_1	Download is started.	09:08:13 05/05	INFO
ns_DESY_Stream_1	waiting for... VTP6,	09:08:22 05/05	WARN
ns_DESY_Stream_1	Download succeeded.	09:08:31 05/05	INFO
ns_DESY_Stream_1	Prestart is started.	09:08:37 05/05	INFO
ns_DESY_Stream_1	waiting for... VTP6,	09:08:46 05/05	WARN
ns_DESY_Stream_1	waiting for... VTP6,	09:08:51 05/05	WARN
ns_DESY_Stream_1	Prestart succeeded.	09:08:53 05/05	INFO
ns_DESY_Stream_1	Go is started.	09:08:55 05/05	INFO
EB1	Emu PEB1 go: waiting for PRESTART event in module EbModule (client msg)	09:08:55 05/05	INFO
G1	Emu AG1 go: waiting for PRESTART event in module EbModule (client msg)	09:08:55 05/05	INFO
ns_DESY_Stream_1	Go succeeded.	09:08:59 05/05	INFO

“Zero” Suppression

- In the streaming environment we have to mange 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. So for 65μs frames that comes to ~1.1MB/s ”empty” data rate.
- If there is no Payload data then they are not present. AIS entries establish they were empty.
- 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%

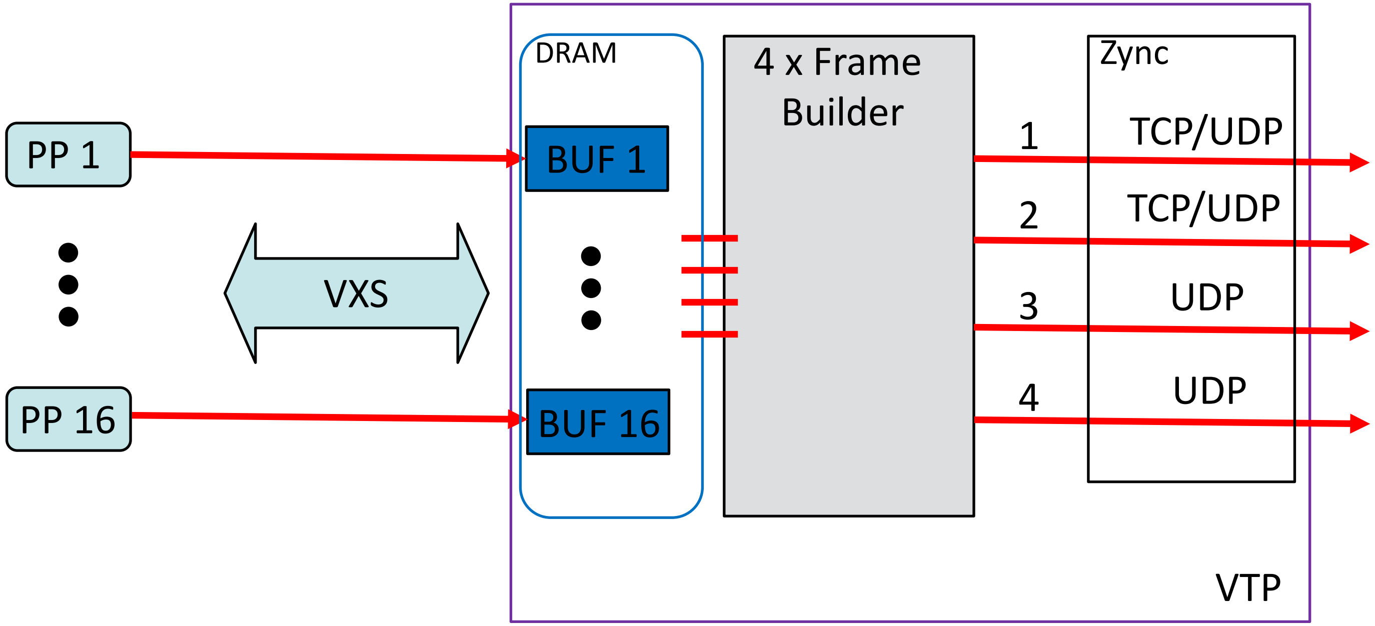


Minimum 40 bytes for up to 2 payloads.
Total 68 bytes for 16 payloads

Congestion Management

- Too much data is handled both locally an globally
- Globally "Sync" is used to start and stop all the streams at their source (FADCs)
  - A "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"
- A Frame builder will only aggregate hits into a ROC frame if the fifo is not full. Otherwise the frame is dropped.
- 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
~7-8 Gbps per link without frame drops

UDP performance (9000 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)

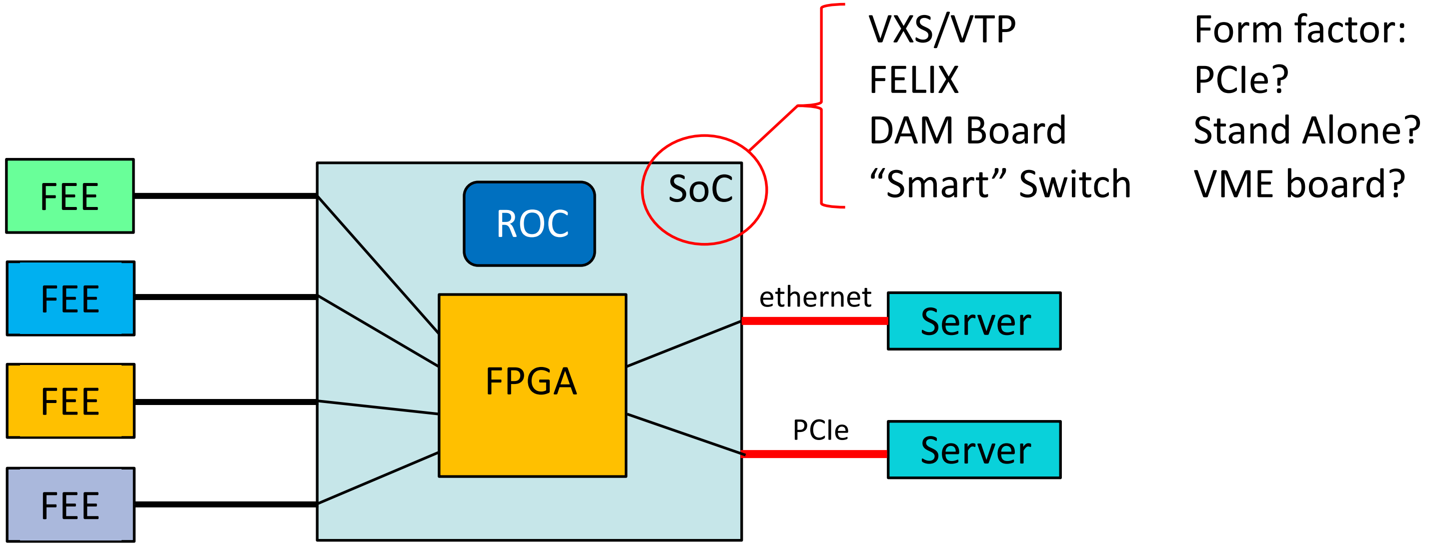
Some general observations…

- 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, 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.

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.