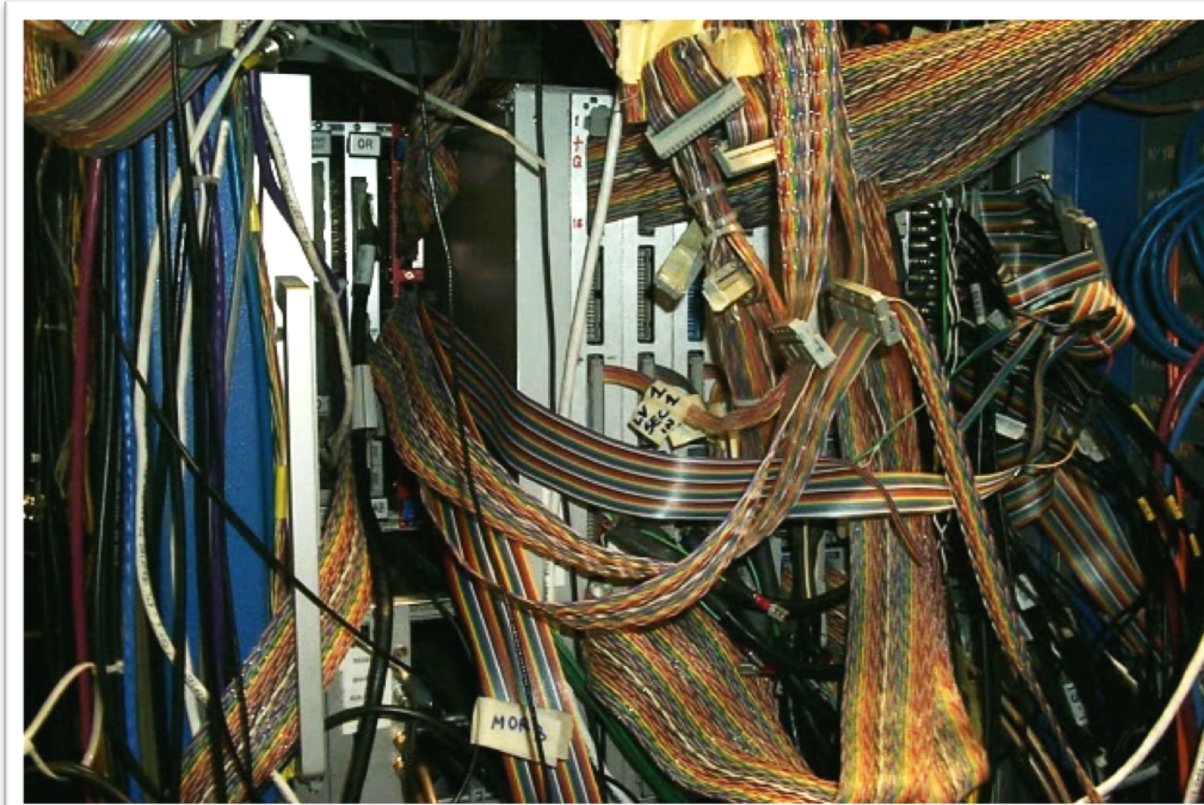


Data Acquisition at JLab



Graham Heyes

Data Acquisition Support

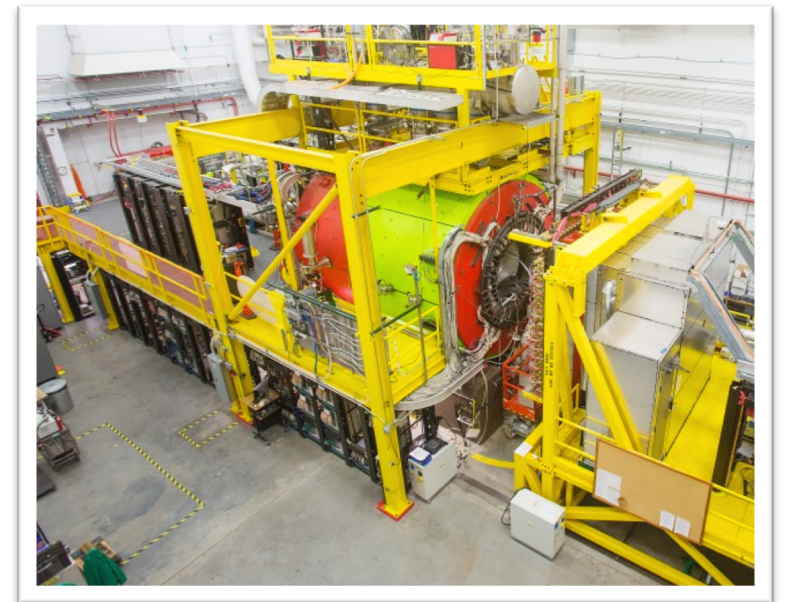
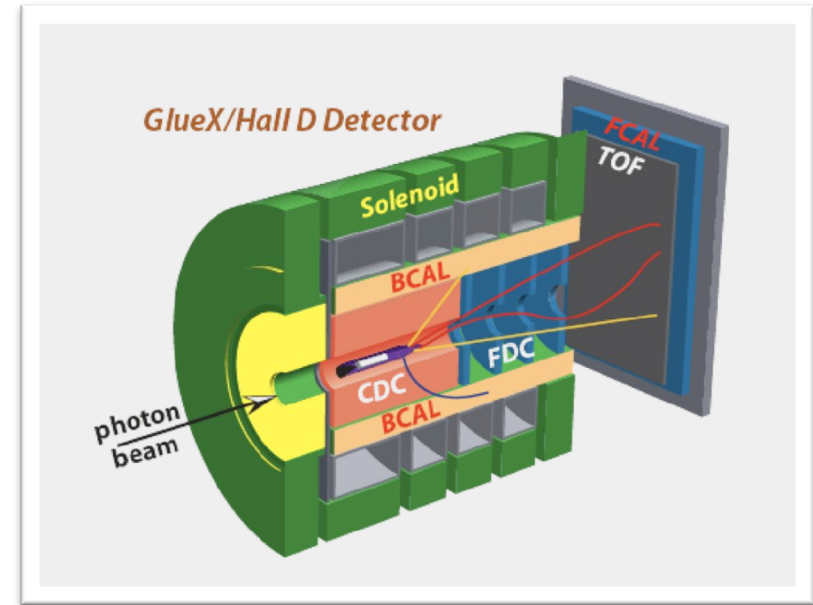
Experimental Nuclear Physics

Data Acquisition

- Data Acquisition
 - Data acquisition is defined as the process of collecting and organizing information.
- The data acquisition support group helps experimenters to take the data from their detectors and store it for future analysis.
- Hopefully this talk will :
 - Show why such a group of people is needed.
 - Demonstrate some of the challenges and techniques.
 - Be interesting!

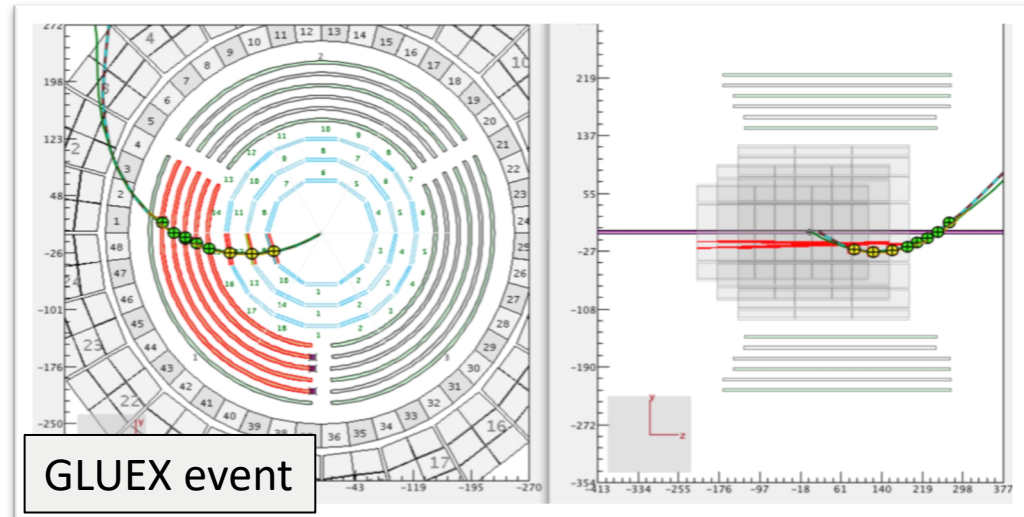
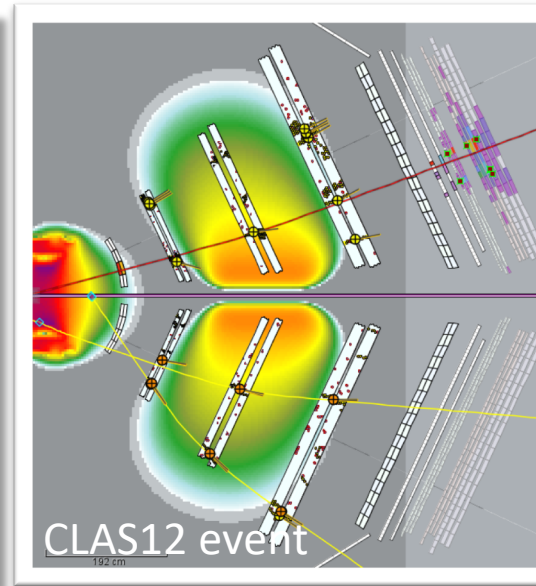
Detection and data acquisition

- Electrons from the CEBAF beam, or photons generated using the electron beam, interact with a target.
- Particles are produced or scattered in the interaction.
- Properties of these particles are measured by detectors.
- The detector can have many parts or sub-detectors.
- The output of sub-detectors is electrical signals.
- Electronics and software convert signals into digital data that can be stored and later analyzed. Three basic types of measurements are.
 - Charges or voltages – ADC (Analog to Digital Converter)
 - Times – TDC (Time to Digital Converter)
 - Counts of things - Scaler
- **All of the data generated from one interaction is grouped together and called an event.**



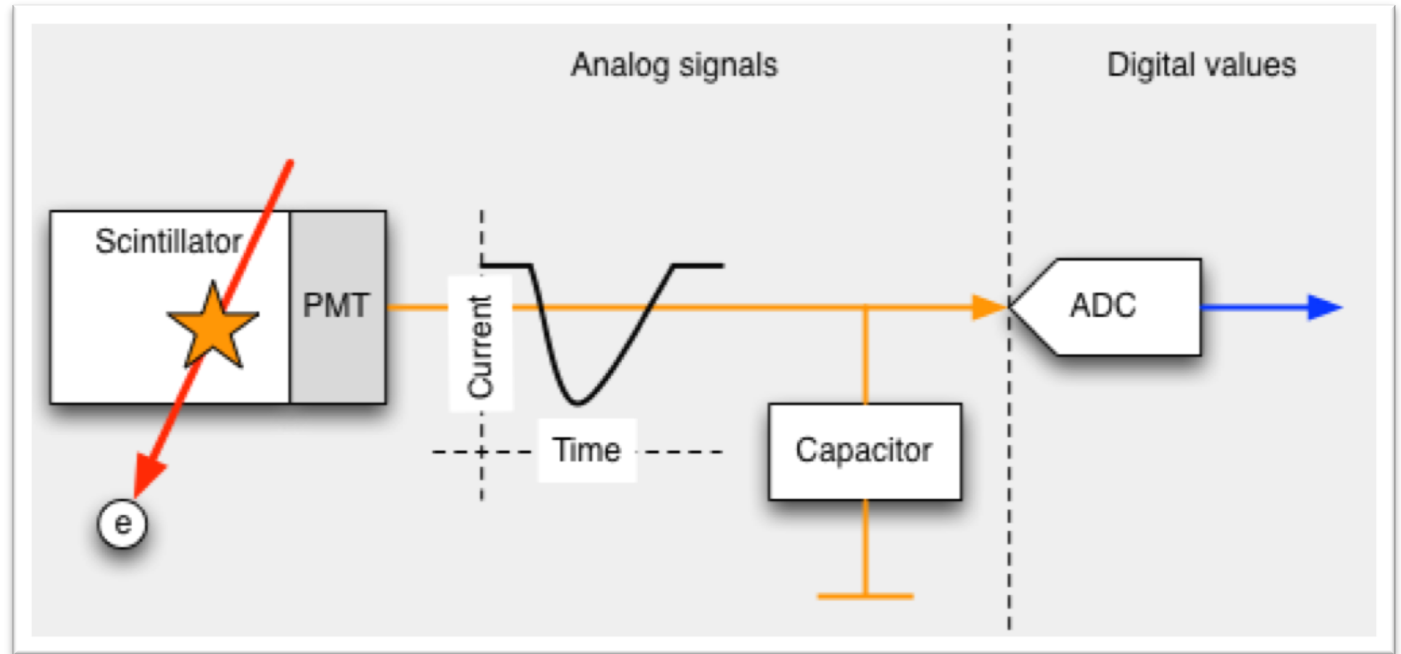
The event as a unit of data

- Data from one event has no history. It doesn't depend upon events that went before and doesn't influence later events.
- Events occur with random timing.
 - Hardware may not be ready for new data.
 - Dead time when data is lost.
 - Events may overlap in time, event pileup.
 - Peak event rate can be more than the average.
- Event size depends upon the physics.
 - Accidental hits unconnected with event.
 - Electronic noise.
 - Distribution of event sizes.
 - Some very large events.



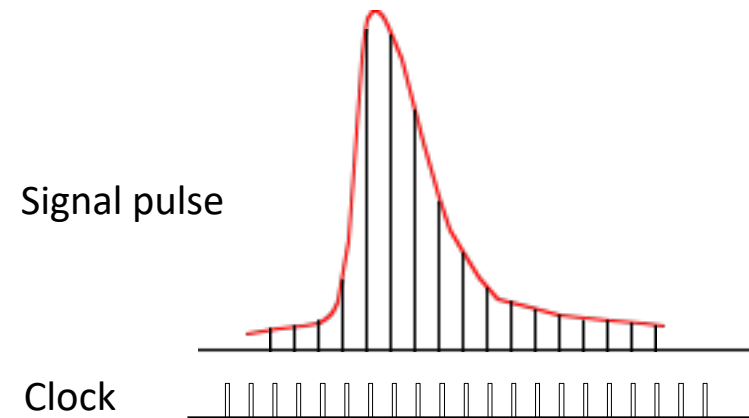
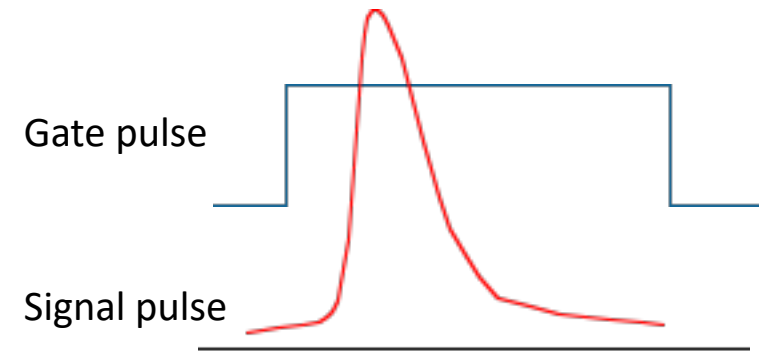
Detector example, a scintillator

- A particle deposits energy in a scintillating material and causes pulse of light.
- A Photo Multiplier Tube (PMT) converts the light pulse into a pulse of electricity.
- The pulse of electrical charge is captured to generate a voltage.
- An Analog to Digital Converter (ADC) measures the voltage as a digital value.



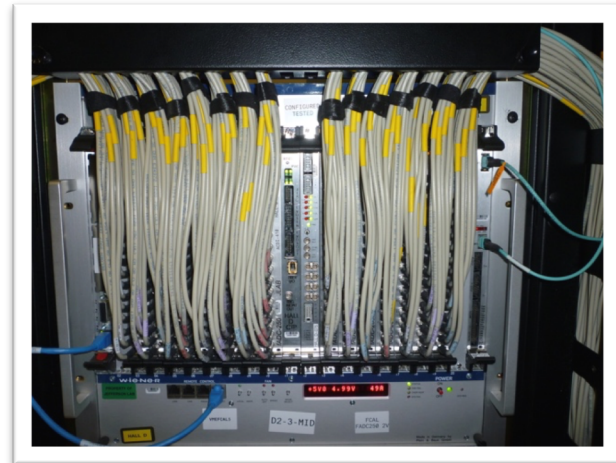
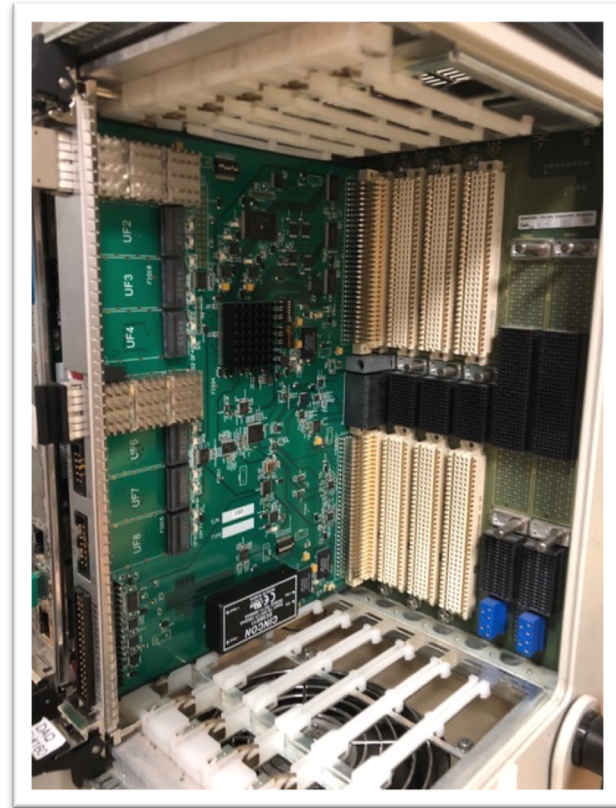
Sampling vs Integration

- A traditional “integrating” ADC takes 6 to 10 μsec to digitize a pulse. A gate pulse, generated by other detectors marks the region of interest to electronics electronics that integrates the charge from the signal pulse.
- This type of ADC generates a single measurement representing the charge sum during the gate.
- A Flash ADC samples continuously at a fixed rate driven by a clock pulse train.
- A sequence of measurements describe the pulse shape as well as measuring charge by summing the measurements.
- For example, a 250 MHz ADC samples every 4 nsec and generates ~ 10 -15 measurements during the gate.



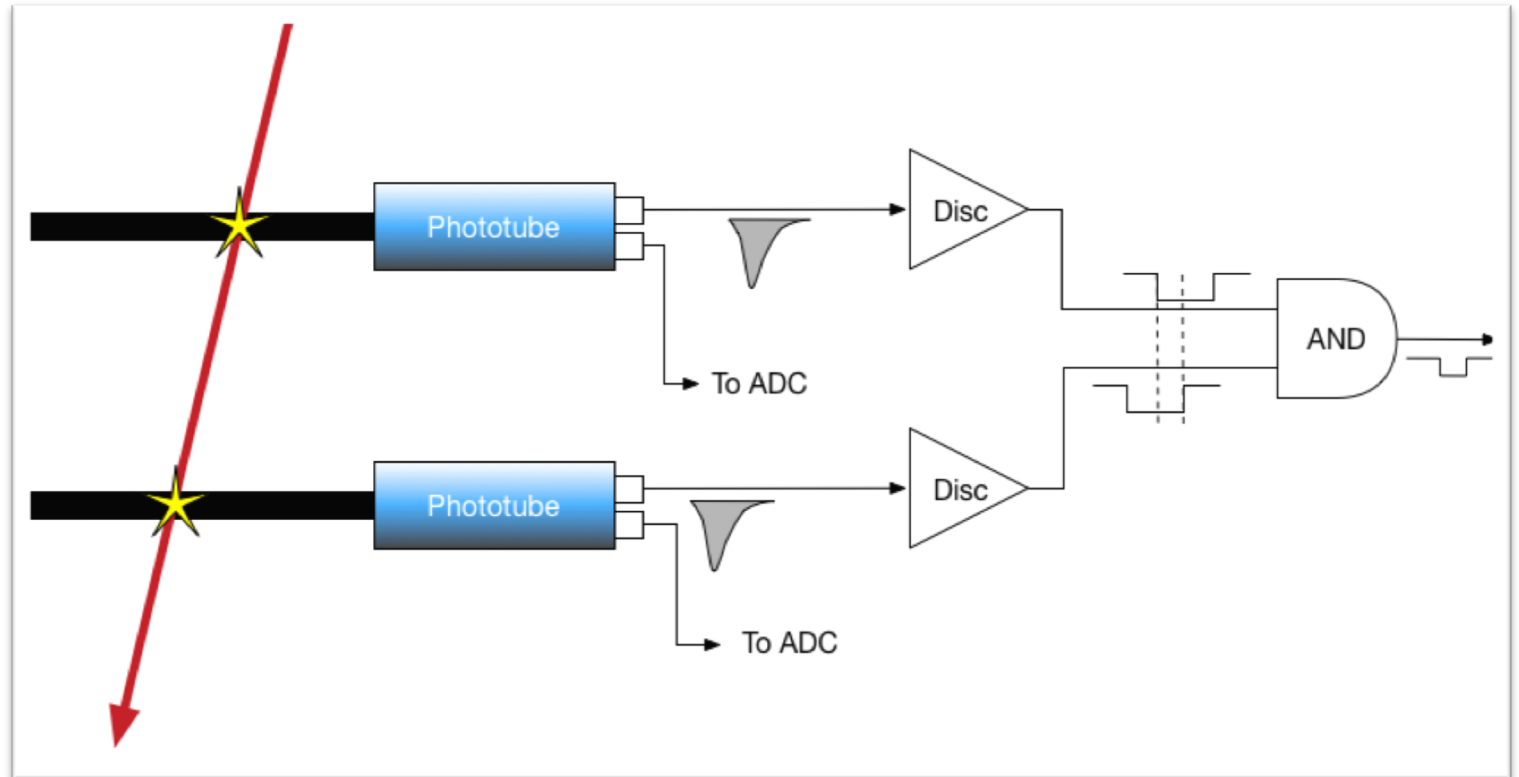
Putting together a system

- In reality there are many data sources in a detector like GLUEX so we need many ADCs, TDCs and other electronics.
- We have ADC boards with 16 ADCs on one board.
- Boards slide into slots in a “crate” and plug into one or more backplanes that provide power and interconnect the boards.
 - At JLab we commonly use VXS, which is a variant of the VME standard for interconnecting electronics.
 - 16 ADCs per VXS crate = 265 channels of readout.
- Usually there is a single board computer in the left most slot to configure and read out the boards.
- VXS has a third “backplane” that provides high speed point-to-point serial data links between boards and two switch slots in the center of the crate.



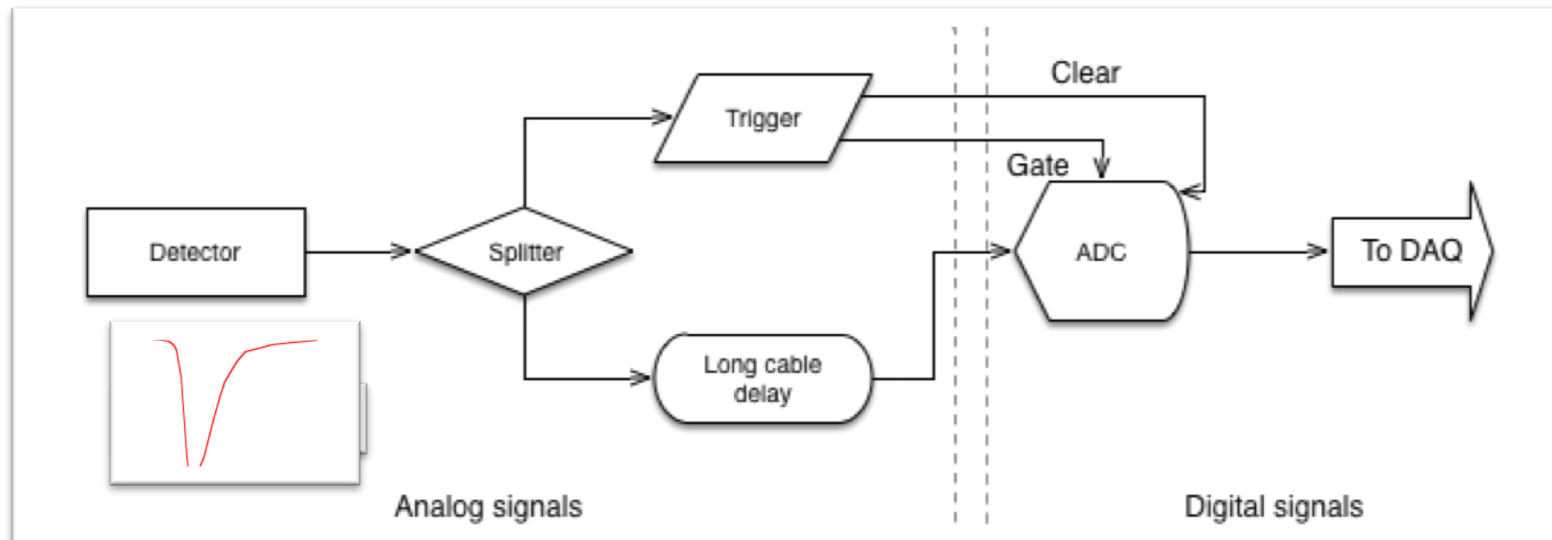
A Simple Trigger

- How do we know the signal on a particular channel came from an event?
 - Fortunately we have more than one detector.
 - Combine data from different detectors to characterize events.
 - Determine which events are interesting.
- Example - coincidence trigger – two detectors have data within a time window.



An analog trigger is simple but not great

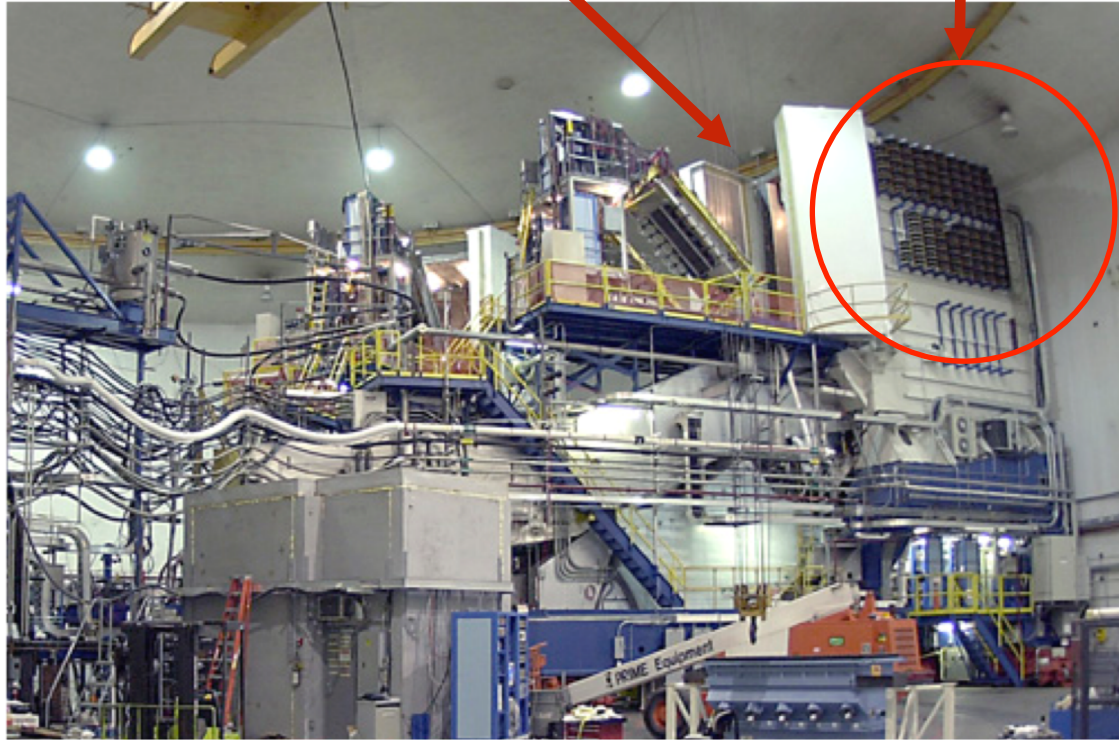
- It takes some time for the trigger logic to decide if a signal should be digitized.
- The analog signal must be delayed so that the gates and signals arrive at the correct time. Typical coax cables ~ 1 ns/ft so you could simply delay the signals using long cables.
 - Matching cable lengths is very important.
 - The ADC cannot process a new signal until it is read or cleared.
 - This limits the trigger rate.



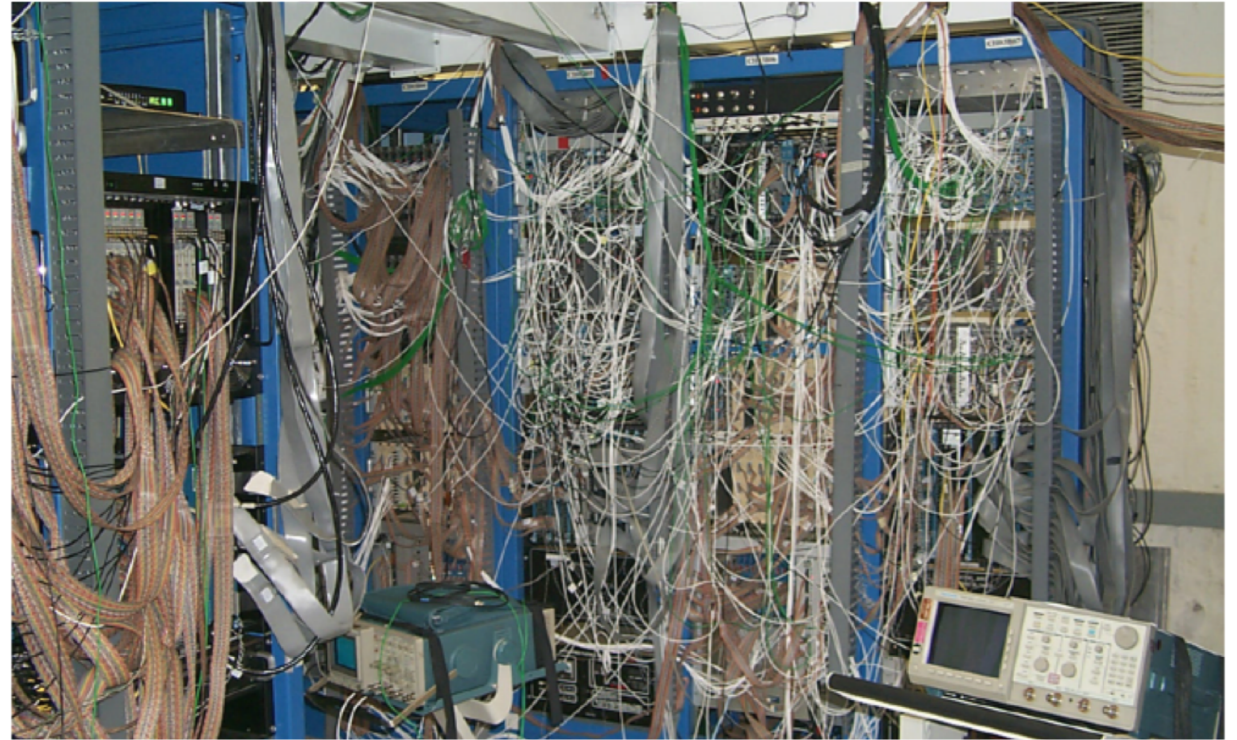
The problem with analog signal cables.

Detector hut

Delay cable

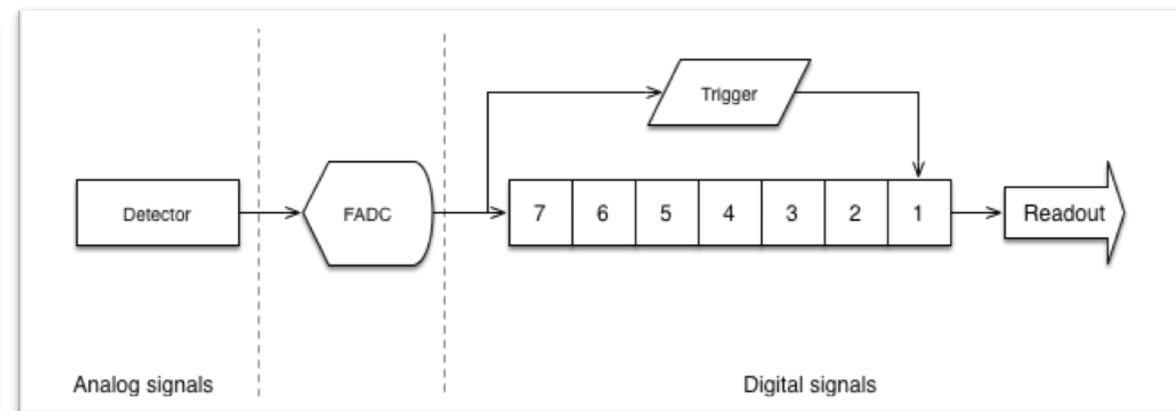


Using cable lengths to adjust timing
can lead to a big mess!



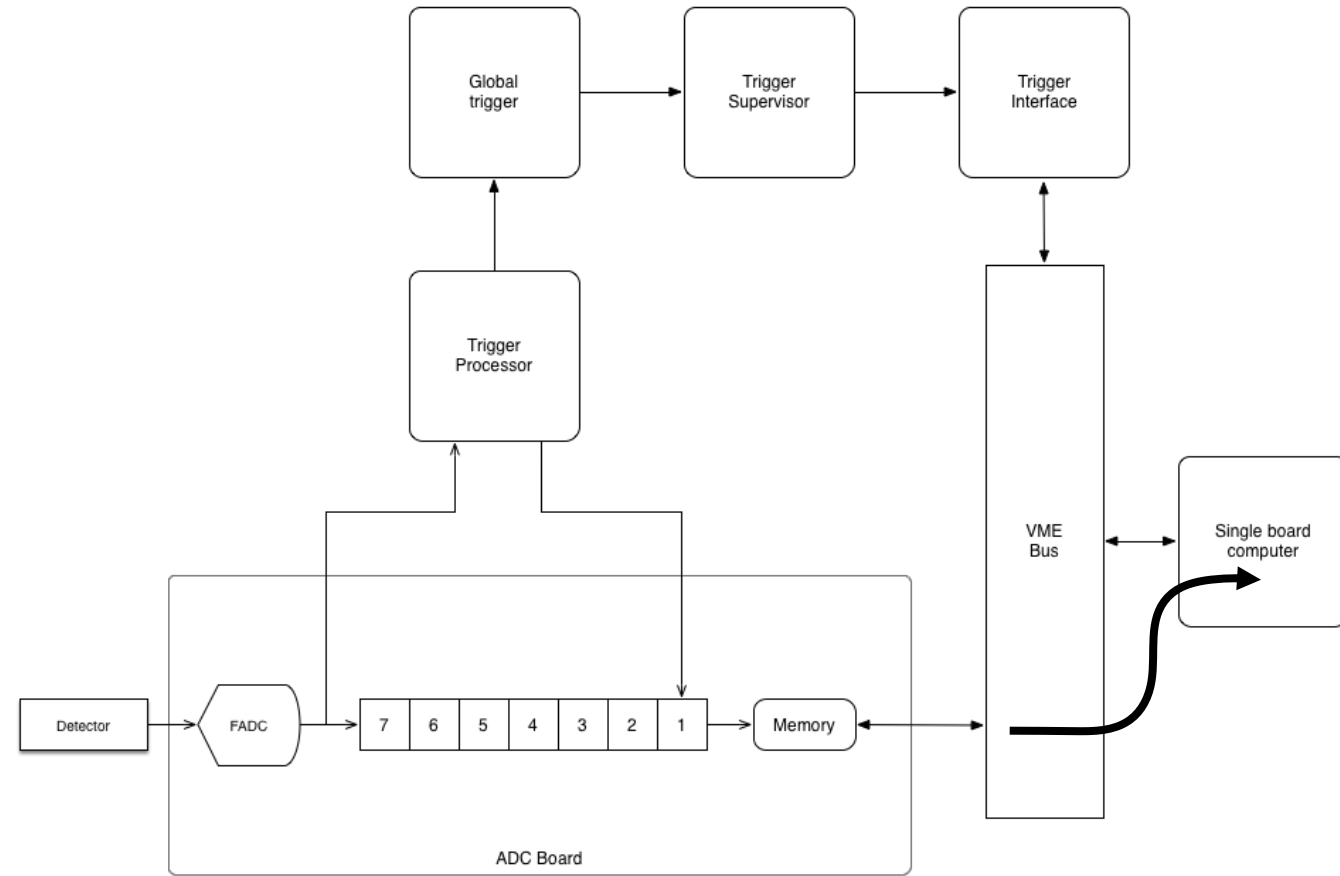
All digital pipeline triggers

- Digital electronics lets us replace all that cable with digital memory.
- In a pipelined system a Flash ADC digitizes at a constant rate and stores the values in a memory.
- Values are clocked into memory at the same rate as the Flash ADC sampling clock which, in the case of GLUEX, is 250MHz (one clock every 4 nS).
- For example, if the trigger logic takes 28 nS we know that trigger corresponds to measurements $28/4 = 7$ cells down the memory pipeline.
- The readout software can selectively read only the useful samples.
 - With out ADCs we can chose to read all the samples or integrate in firmware and read only the charge + time of arrival.



A global trigger system

- Modern experiments require very complex trigger decisions involving several detectors.
- Use a “pipeline global trigger distribution system”.
 - Each ADC sends signals to a crate level trigger processor over VXS serial bus.
 - Crate level processor makes local decisions and passes the result
 - To a global trigger processor over fiber optic cable.
 - Global trigger makes decisions based on many detectors.
 - Global trigger is passed to a trigger supervisor (TS) which generates the trigger and data describing the trigger decision.
 - TS passes this information to a Trigger Interface (TI) board in each crate.
 - The TI signals the single board computer to read out ADCs and TI.



GlueX trigger, starts at ADC crate

16 channels per board

15 boards = 240 channels per crate

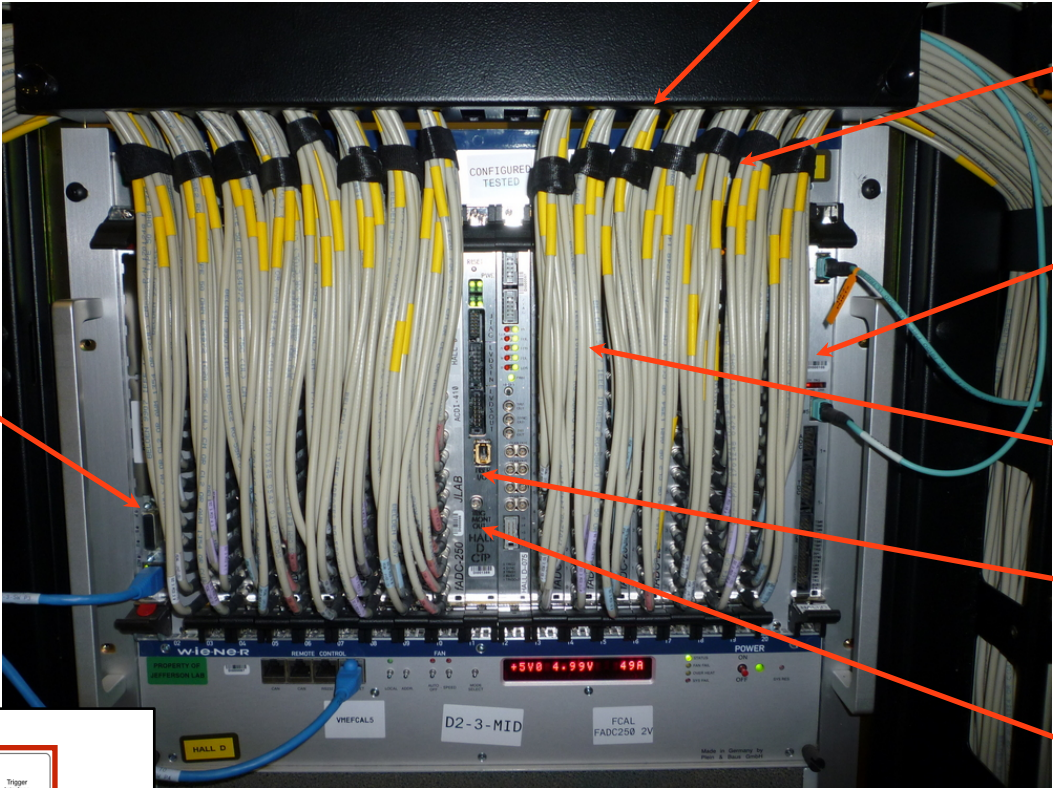
Trigger interface

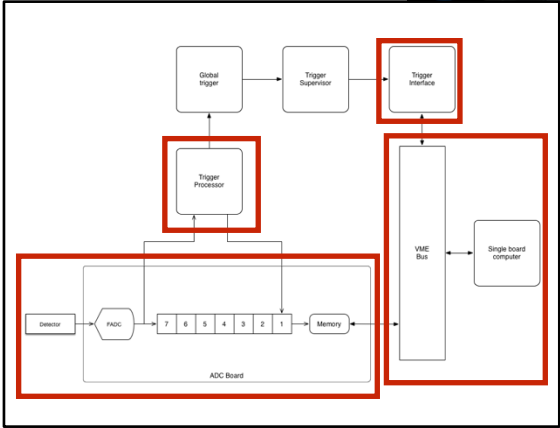
Boards are connected to CPU via VME bus.

ADC trigger data over VXS serial bus

CTP Board sending signals to trigger over fiber

Intel CPU Read Out Controller (ROC) running Linux





```
graph TD
    GlobalTrigger[Global Trigger] --> TriggerSupervisor[Trigger Supervisor]
    TriggerSupervisor --> TriggerInterface[Trigger Interface]
    TriggerInterface --> TriggerProcessor[Trigger Processor]
    TriggerProcessor --> ADCBoard[ADC Board]
    ADCBoard --> VMEBus[VME Bus]
    VMEBus --> SingleBoardComputer[Single board computer]
    SingleBoardComputer --> TriggerInterface
```

The diagram illustrates the GlueX trigger system architecture. It shows a flow from a Global Trigger to a Trigger Supervisor, then to a Trigger Interface, a Trigger Processor, and finally to an ADC Board. The ADC Board is connected to a VME Bus, which in turn connects to a Single board computer. The Single board computer is also connected to the Trigger Interface. The ADC Board contains a Detector, FADC, and Memory components. The Trigger Processor and Trigger Interface are highlighted with red boxes, as are the ADC Board and VME Bus components.

All crates connect to Global Trigger Crate

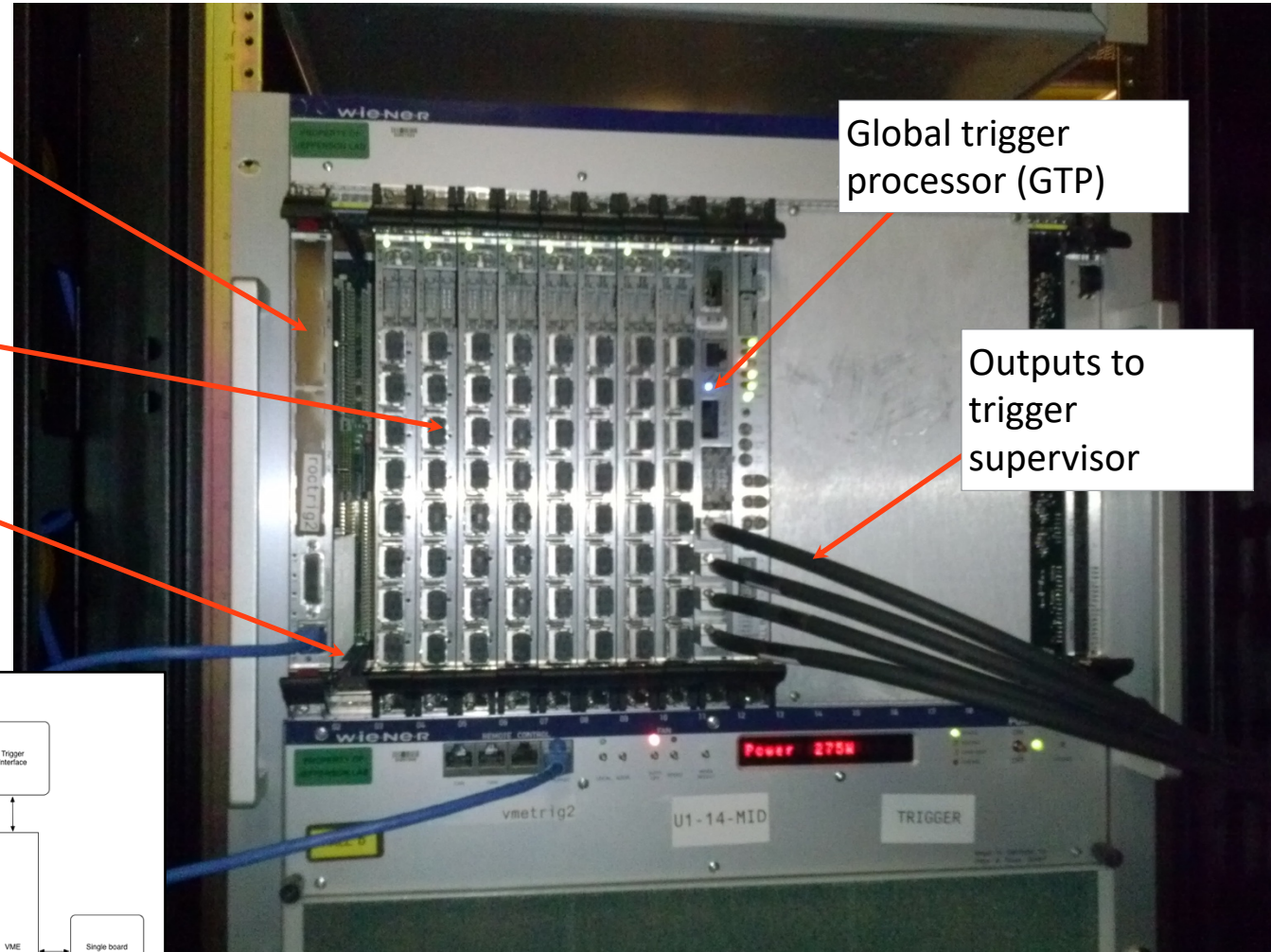
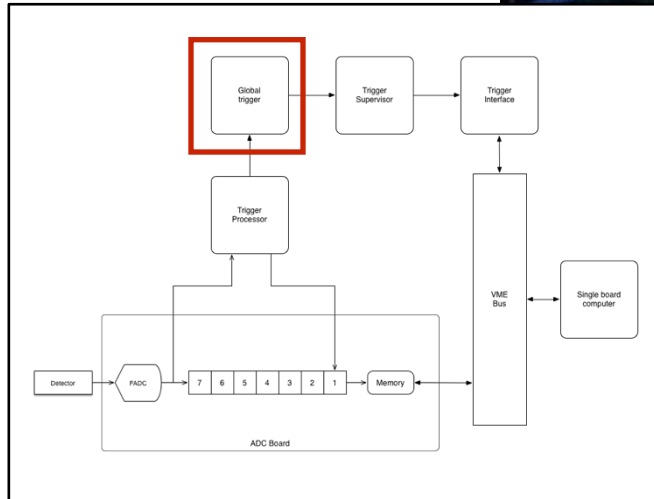
Intel CPU controller

Sub-system processor board (SSP)

Eight boards with eight connectors each so up to 64 crates.

Global trigger processor (GTP)

Outputs to trigger supervisor



Final trigger goes to Trigger Supervisor crate

Trigger Distribution board (TD).

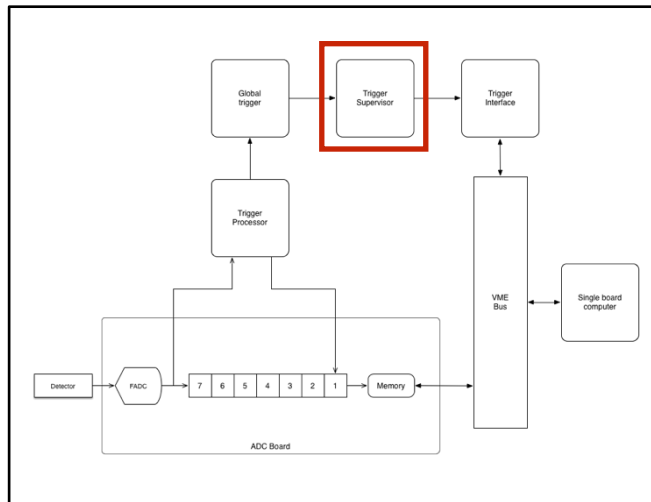
Intel CPU for control and configuration.

Optical trigger link back to crates.

VXS serial backplane

Signal Distribution board (SD)

Trigger Supervisor (TS)



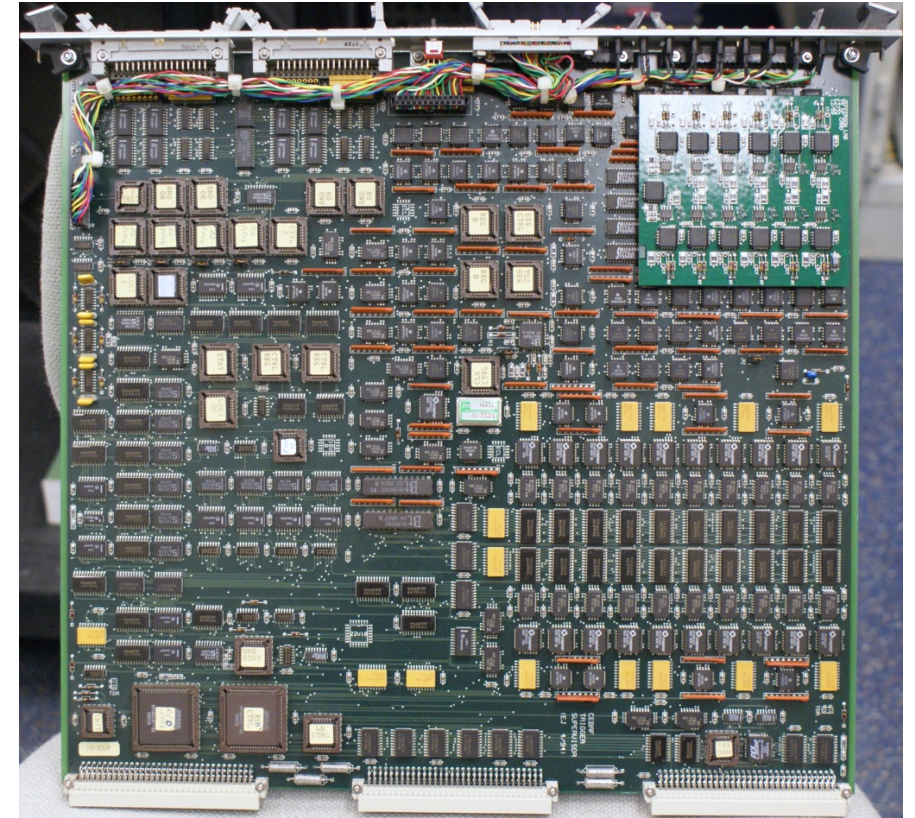
Then back to the start to trigger VME readout



- The TI board gets from the trigger supervisor:
 - Signal to CPU to read the memory of the ADC boards.
 - TI is read to get data about which events the ADC data belong to.
- The CPU:
 - Copies ADC and TI data into memory via VME backplane,
 - CPU encodes data in EVIO data format.
 - Sends the data over the network to the rest of the data acquisition system.

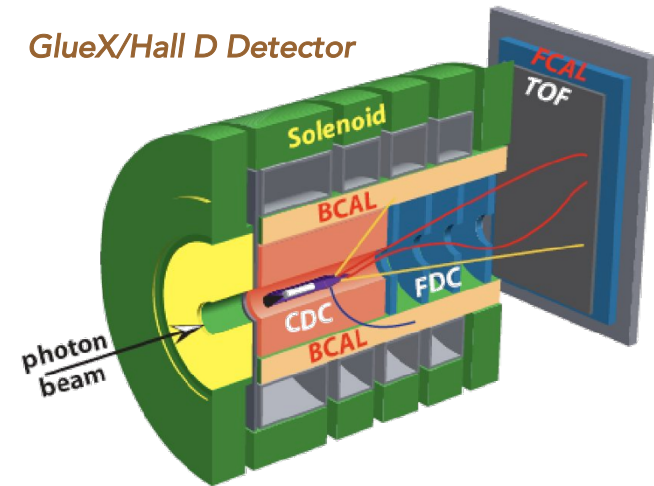
Complex electronics

- We can now buy programmable logic arrays that allow us to implement complex algorithms in the firmware on a single chip.
- The two pictures are of boards with similar functionality.
- The one on the left designed in the early 1990's.
- The one below designed in 2011.
- The scale is the same for both photos.

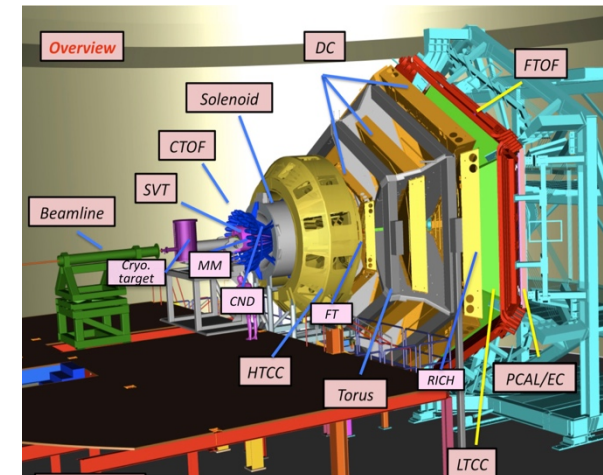


Data acquisition for big experiments

- Particle detecting hardware is physically distributed within the detector so we need to:
 - Record where in the detector the data came from.
 - Record which event the data belongs to.
 - Transport the data to computer systems
 - Gather all of the data from one event together.
 - Record the data in some sort of long term storage.
 - All of this requires a well known data format.
- Experiments run for months or years so need :
 - Software and hardware stability.
 - A control system to start and stop data taking.
 - To way to monitor and record the conditions under which data was taken.

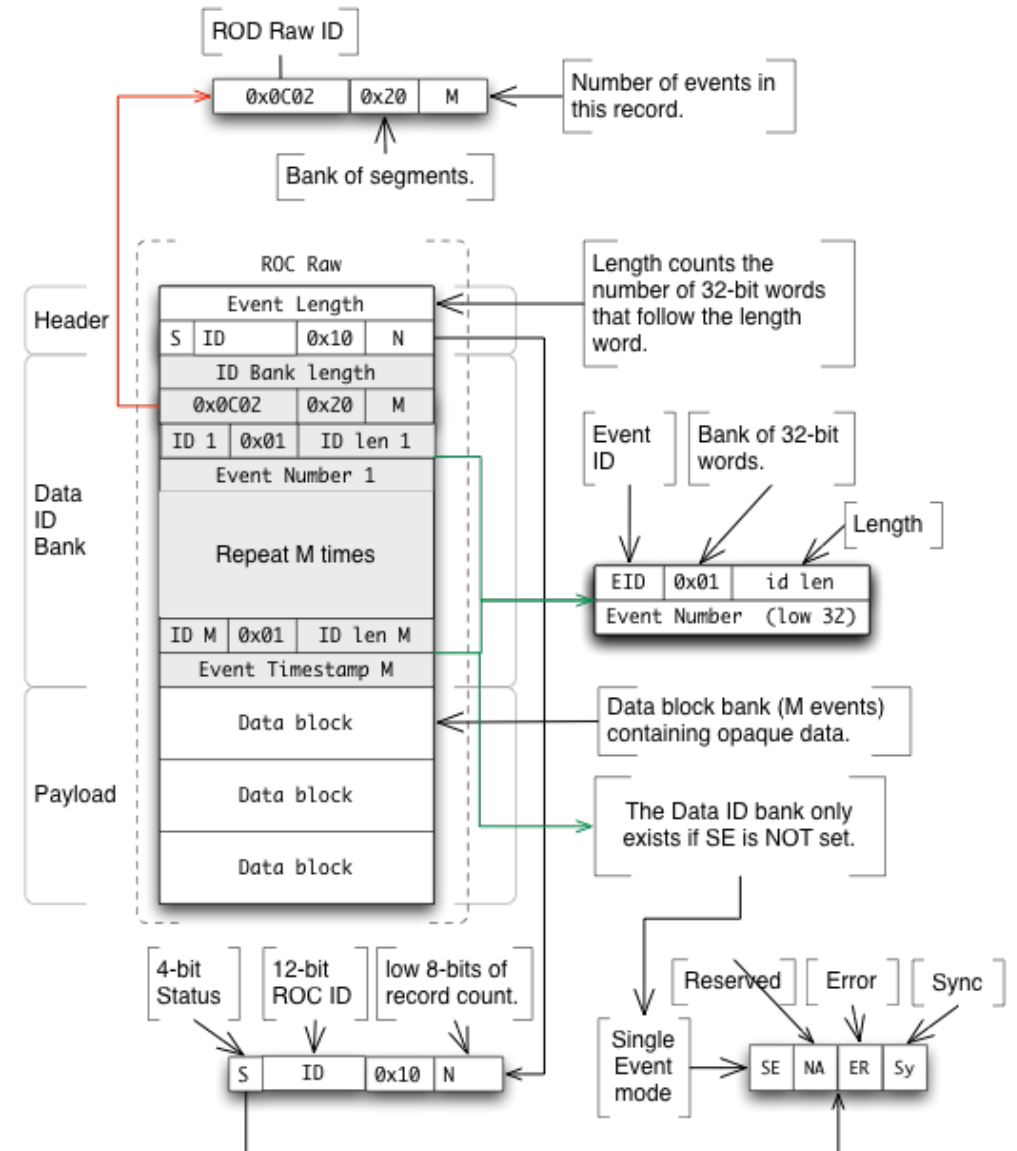


Clas12/Hall B Detector



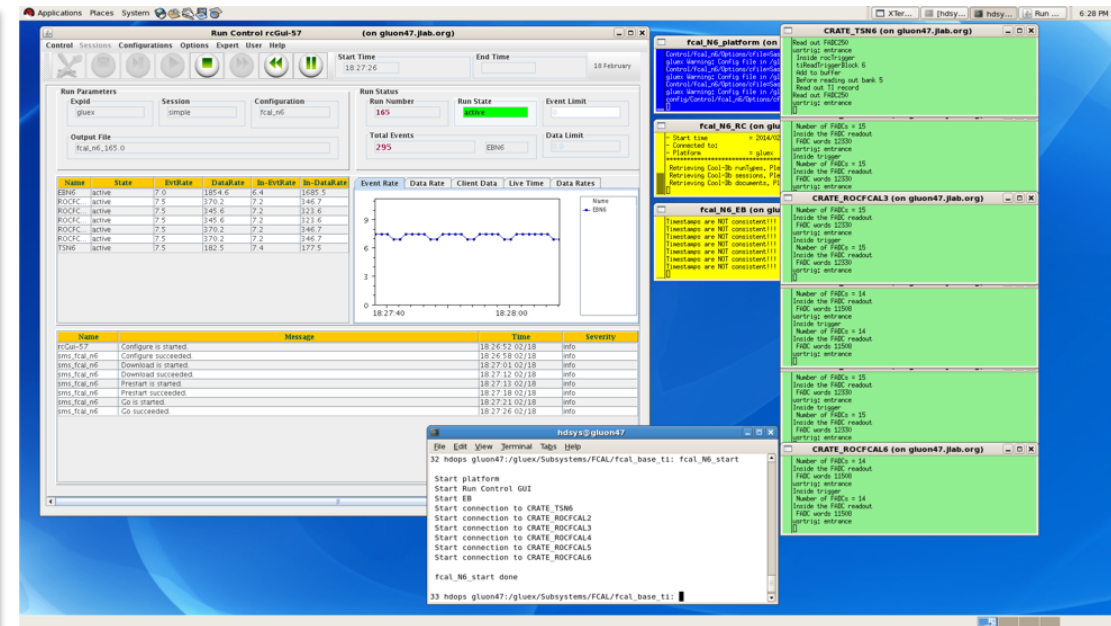
EVIO Data format, a real example

- EVIO is a self describing hierarchical format consisting of nested banks.
 - Bank - container for other data.
- Each bank starts with a header.
 - Length.
 - Description of content.
- The code 0x10 in the outer header tells us this bank contains other banks that are 32-bits wide.
- The first bank is a list of trigger information for all the events in the block. To save space this info is 16-bits.
 - The code 0x20 tells us that this is a bank of 16-bit wide segments (mini banks).
- The following “payload banks” contain blocks of raw data read from ADCs or TDCs.



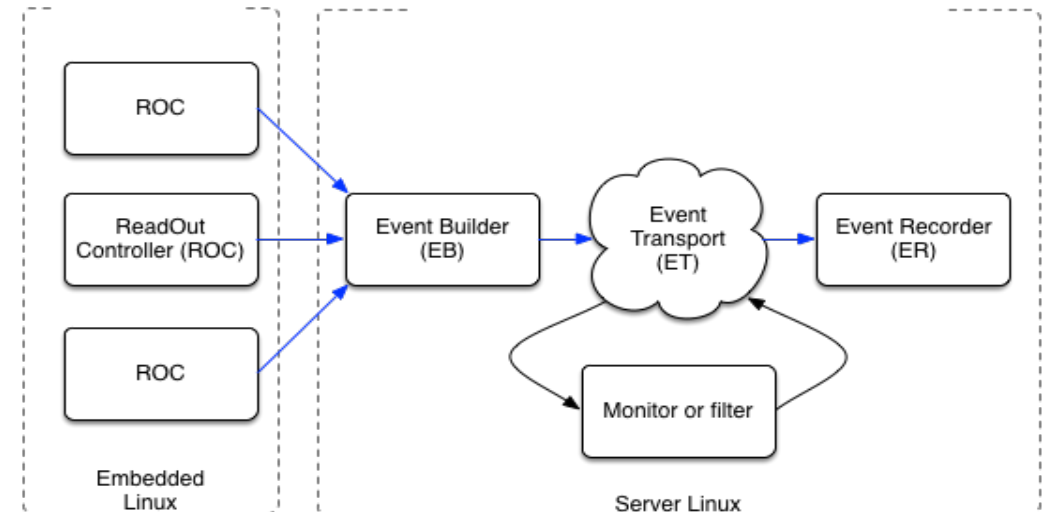
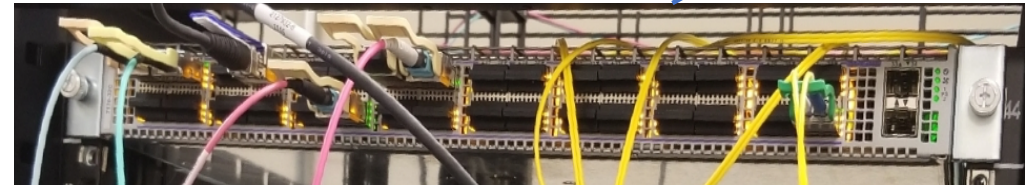
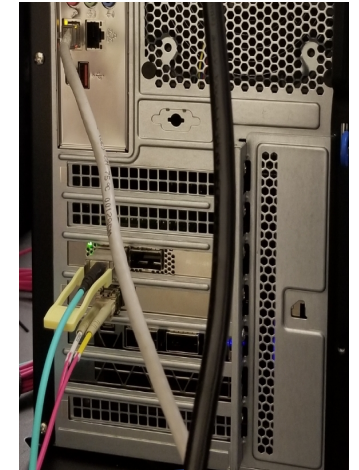
CODA

- CEBAF Online Data Acquisition (see coda.jlab.org)
 - Kit for implementing data acquisition systems.
 - Electronics
 - Custom boards like trigger, TDCs and ADCs.
 - Support for commercial hardware.
 - Software to :
 - Interface with electronics.
 - Readout boards and format data.
 - Move data.
 - Merge data streams.
 - Give users access to data for monitoring.
 - Write data to files.
 - Control the data acquisition system
- CODA is modular, solves big problems by splitting them into smaller ones.



Network based data acquisition

- The DAQ components are spread out around the detector.
 - Need to bring the data from and event together
 - VME CPU has ethernet built in so ROCs send data via a network.
 - Use a network switch to route data.
 - PCs and servers can match network bandwidth to DAQ rates.
- Bits and pieces of events arrive at different times from different places but are tagged with an event number.
 - Assemble pieces with the same event number into an event.
 - This is done by the Event Builder (EB) software.
- Need to monitor data taking online.
 - This is done using Event Transport (ET) event sorting software.
- Need to store the data.
 - This is done by the Event Recorder (ER) software.
- A lot of pieces need to work together.
 - All managed by Run Control software.

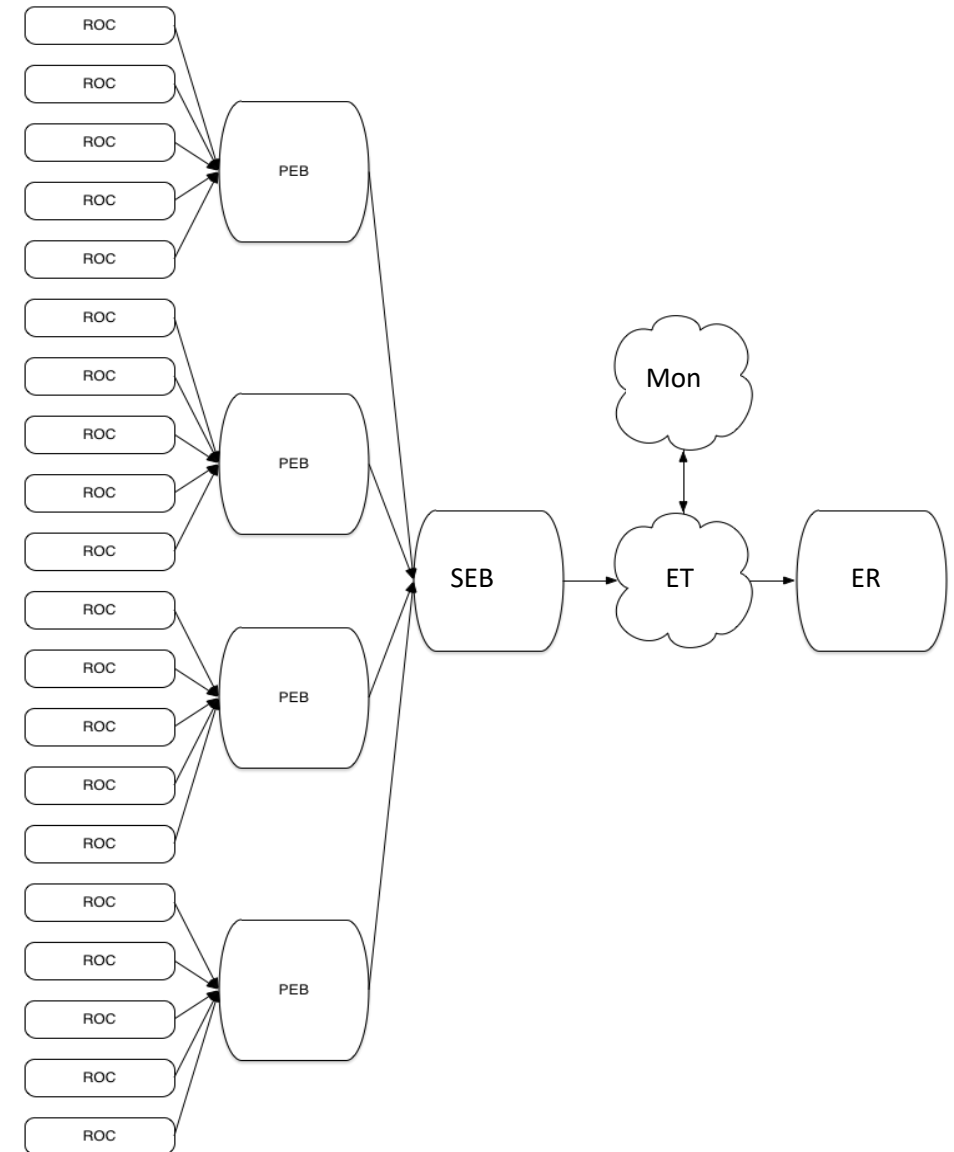


Event Building challenges

- In 2018 GlueX event rate was 90 kHz and there are about 50 crates in the system.
 - The Event Builder has 11 μ S to:
 - Find all 50 parts of an event.
 - Decode the incoming data headers.
 - Check for errors.
 - Generate new headers for the complete event.
 - Generate a trigger bank per event.
 - Copy all of the data into place.
- The GLUEX current goal data rate is 1.5 GByte/s.
 - ROCs work in parallel with 30 MByte/s average per network link over 50 incoming links.
 - Full 1.5 GByte/s through the EB, ET and ER.
 - Since data is copied several times during the building process the data rate inside the machine running the EB is several times 1.5 GByte/s.
- That could be a lot of work for one machine to handle.
 - Solution: multi stage parallel event builder.

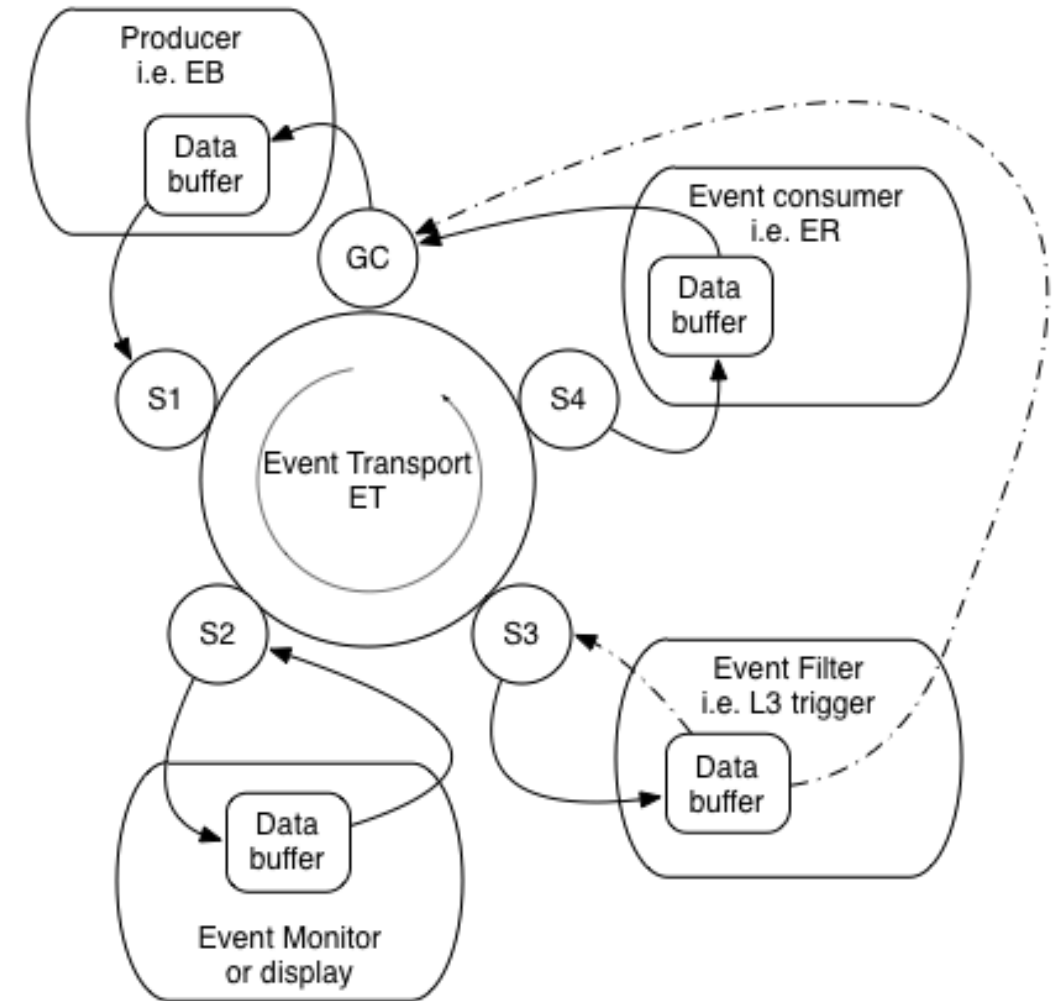
Staged Event Builder, 20 ROC example

- Correctly building events involves a
- Four Primary Event Builders (PEB) are connected to five ROCs each by 1 Gbit/s Ethernet links.
- For each PEB this divides the formatting/checking work load and the data throughput for by four.
- The four PEBs are connected to a single Secondary Event Builder (SEB) via 40 Gbit/s Infiniband links
- The SEB has to handle the full rate but only has four incoming streams to handle.
- If this is too much we can use two SEBs in parallel.
- The SEB outputs to a system called ET which distributes the events to online data monitors.
- A final program called the Event Recorder (ER) writes data files to disk.



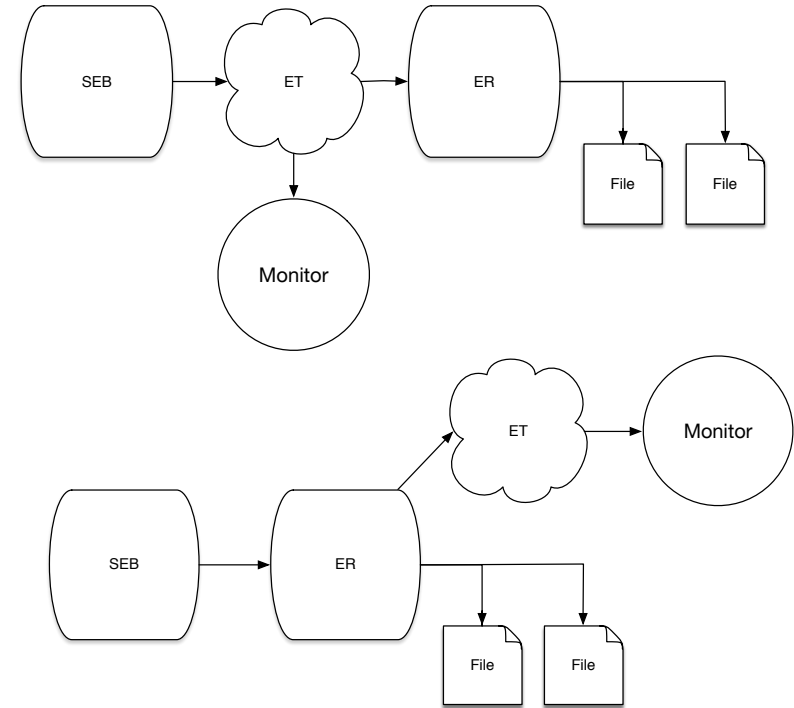
Event Transport, ET

- Allocating and freeing buffers is time consuming.
- The ET system gives programs access to data via preallocated shared buffers.
- The system uses a railroad metaphor.
 - Empty data buffers originate at Grand Central.
 - They are filled by data producers and tagged to describe the content.
- The buffers are passed from station to station and the tag is checked to see if the buffer should stop at the station.
- Example :
 - An event monitor on station S2 samples 1% of the events.
 - An event filter on station S3 looks at all events.
 - Discarded events are sent back to GC good ones move on.
 - An event recorder takes all events.
 - After the data is stored to a file it sends the buffer back to GC.



Real World Performance

- Even with a fast server and fast disk there is a limit to how quickly one program can write to a single file.
- Solution – ER writes in parallel.
- The ET system was designed for monitoring – it was never intended to run a single ET over 1 GB/s.
- Quick solution: move the ET out of the data path and give it a sample of events.
- Slow solution: rewrite the internal ET code to present the same API but move data using more efficient modern techniques.



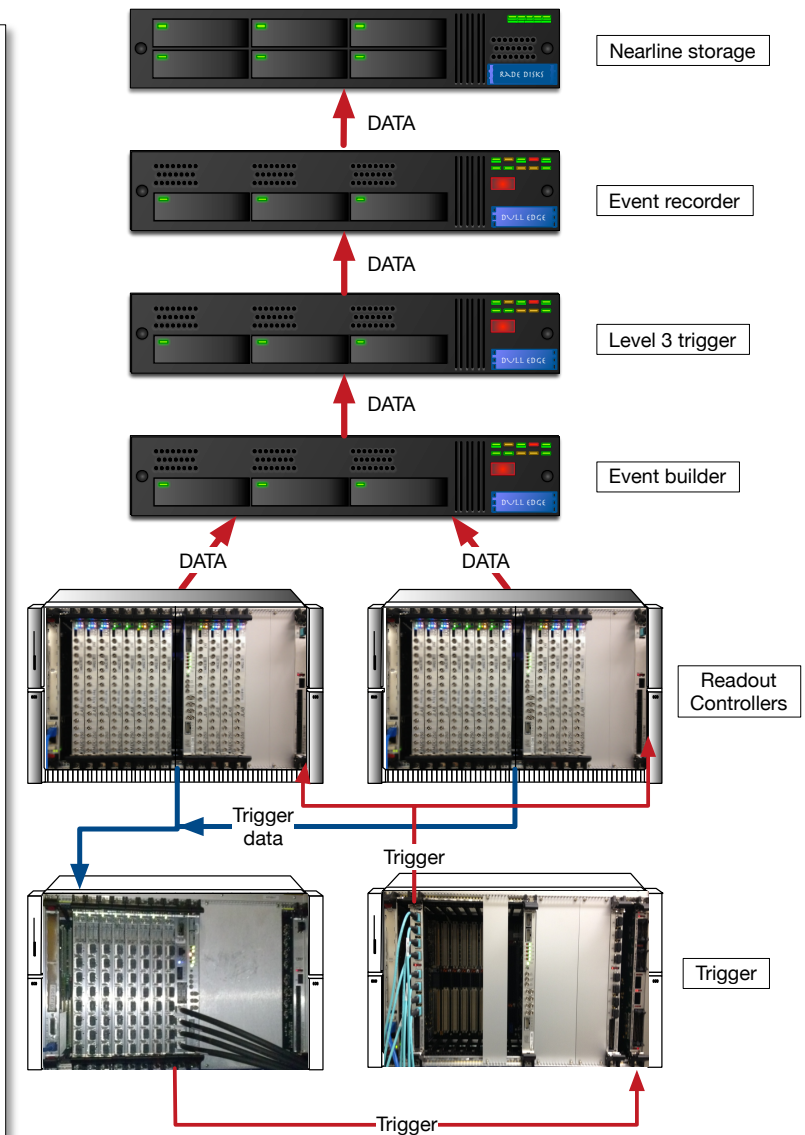
Data storage

- 6 GeV experiments ran at tens of MB/s.
- 12 GeV experiments, hundreds of MB/s.
- Generate tens of petabytes per year.
- Tape is cheap but disk is faster.
- We write data to disk then copy from disk to tape later.
 - Tape speed only needs to handle average rate over a 24 hour period.
 - Tape drives and library robots are expensive and fragile. Writing to disk allows data taking to continue if the tape system breaks.
 - We typically have enough disk to hold three days of raw data.



Summary of SoA at JLab – pipeline mode

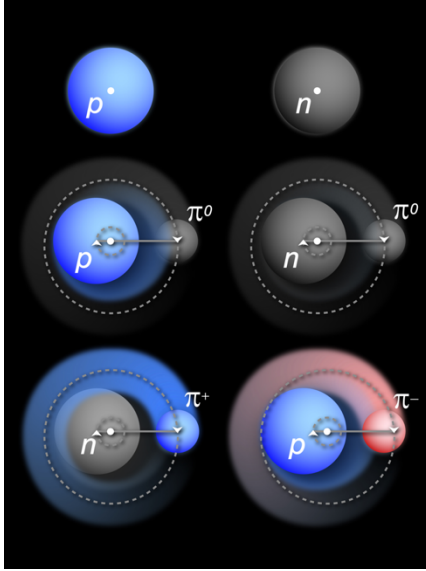
- Data is split into trigger and DAQ paths.
- Trigger data read over VXS serial backplane by Crate Trigger Processor and passed to global trigger. Trigger formed in custom electronics
- Blocks of data queued in “pipeline” while trigger is made and read out by CPU over VME.
- Data sent over network to Event Builder
- Pros:
 - Data stream is a stream of events containing data from detectors.
 - Trigger filters “unwanted data”.
 - Well understood way of doing things.
- Cons:
 - Have to delay prompt data until slowest data appears.
 - Relies on good understanding of trigger.
 - All parts of DAQ have to work.
 - One failure stops the pipeline.
 - Doesn't work well when events overlap in time.
 - **Obvious bottlenecks!**



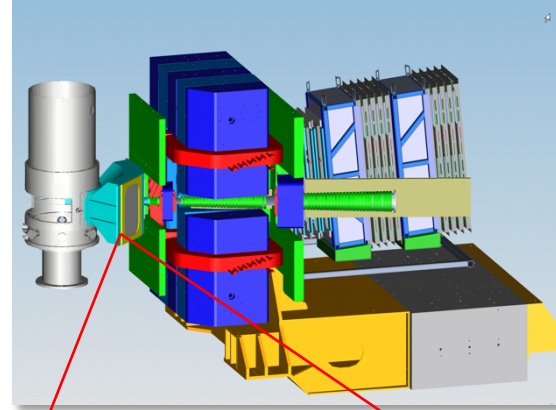
Looking forward

- New experiments are being proposed
 - Detectors that do not play well together due to timing.
 - Traditional trigger and event builder strategies are not ideal.
 - Detectors with peculiar topologies.
 - Detectors split or segmented in a way that makes forming a trigger hard.
 - High event and/or data rates.
 - Particles from more than one event in a detector at the same time – need to disentangle.
- The data acquisition from these experiments does not fit well with current techniques.

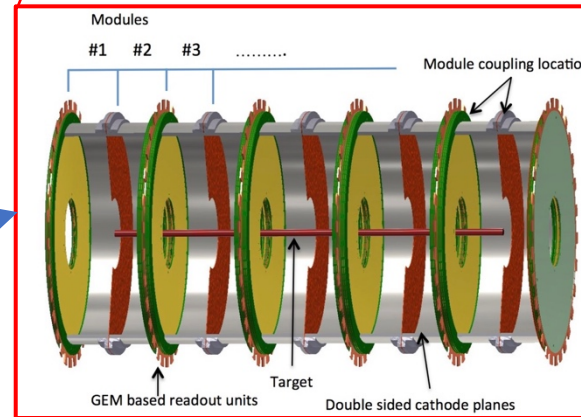
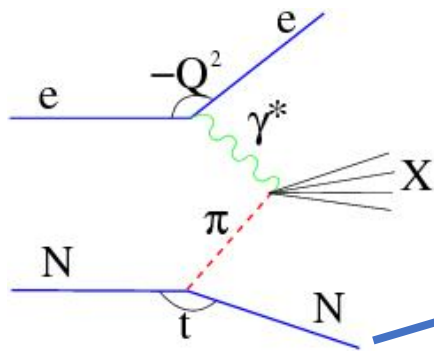
Tagged Deep Inelastic Scattering (TDIS)



TDIS Scientific Goal: Access Elusive Partonic Structure of Mesons by Using Mesons in Nucleons as “Target”



Scattered electrons detected in planned Hall-A Super Bigbite Spectrometer.



“Spectator” protons detected by radial Time Projection Chamber (rTPC).

TDIS - Challenges

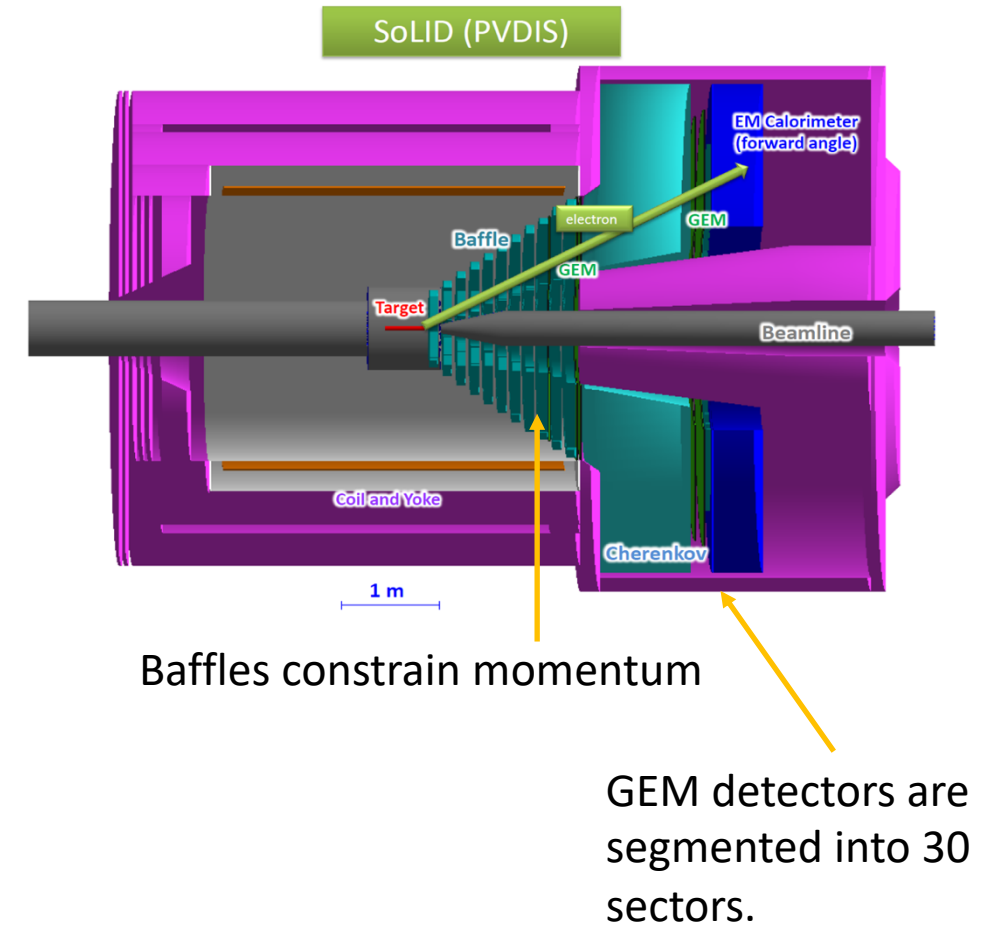
- It takes up to a microsecond for the proton track to trace itself out on the wall of the TPC.
- Super Bigbite electron detectors are much faster – large timing mismatch.
- Large flux of background electrons not associated with a proton.
- Multiple protons in the TPC at the same time.

- TPC has 25,000 pads, hit rate per pad ~800 kHz.
- Data rate from TPC up to 4 Gbyte/s total.

- How to read this out and match up the electrons with the protons?
 - Event building online at these rates is a bottleneck.

SoLID

- SoLID is another experiment proposed for installation hall-A at JLab.
- In the PVDIS configuration electrons are scattered of a fixed target at high luminosity.
- Spiral baffles cut background.
- The detector is split into radially 30 sectors.
 - Single track event topology allows 30 DAQ systems to be run in parallel.
 - Each at rates of up to 1 Gbyte/s each 30 Gbyte/s total.
- Challenges :
 - How to handle 30 Gbyte/s affordably?
 - Hits at sector edges span two sectors?
 - How to integrate GEM with other detectors?

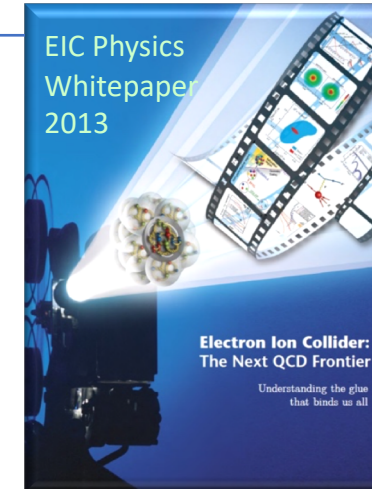
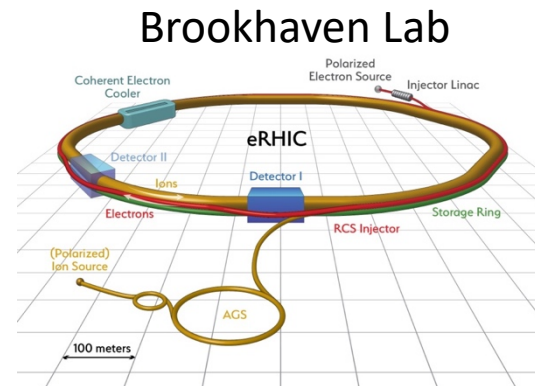
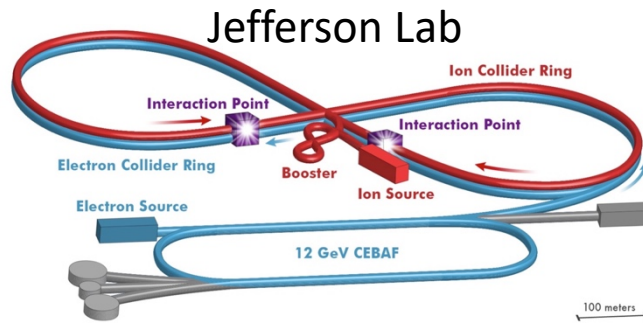


Electron-Ion Collider: The Next QCD Frontier

Electron-Ion Collider (EIC) parameters

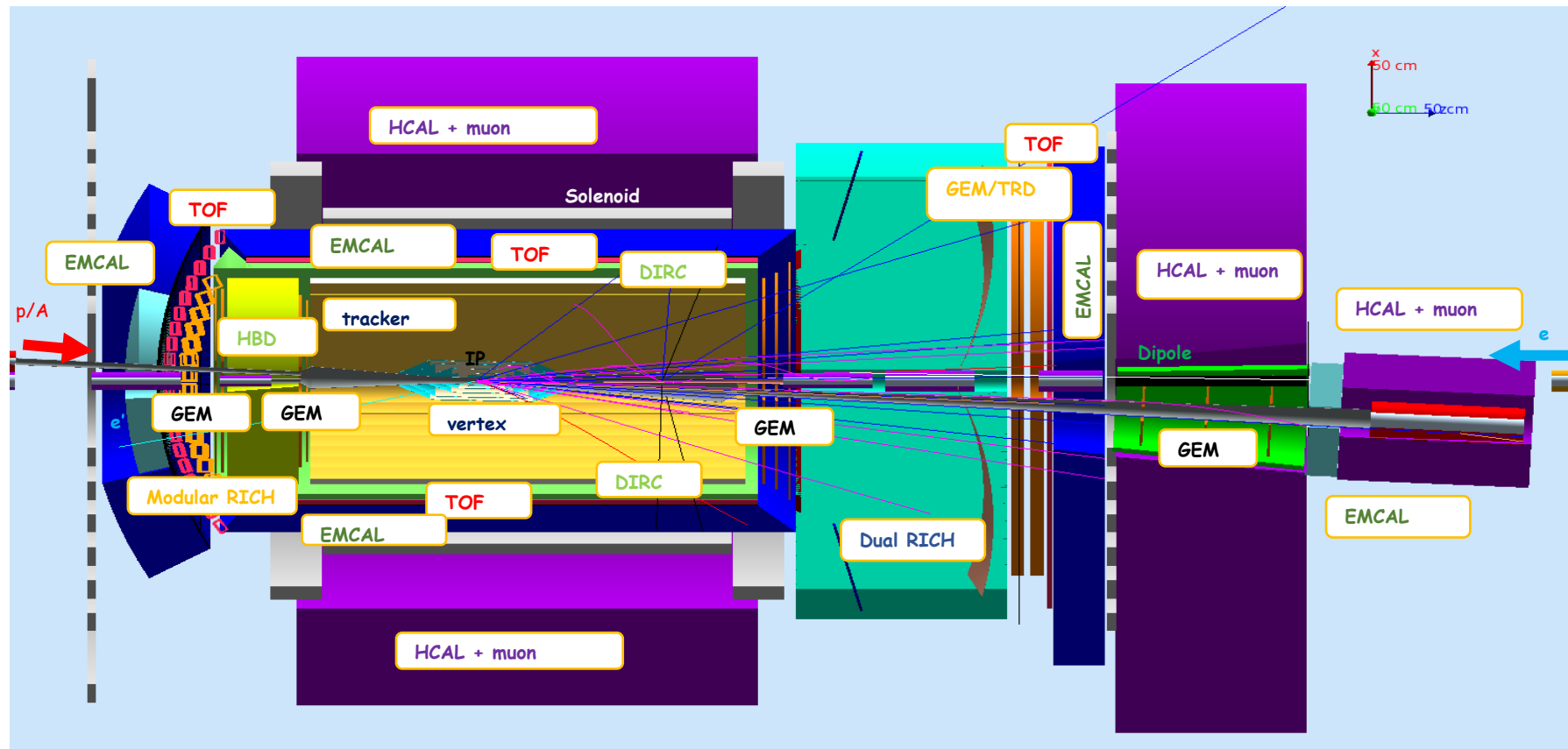
- Highly polarized ($\sim 70\%$) electron and nucleon beams
- Ion beams from deuteron to the heaviest nuclei (uranium or lead)
- Variable center of mass energies from $\sim 20 - \sim 100$ GeV, upgradable to ~ 150 GeV
- High collision luminosity $\sim 10^{33-34} \text{ cm}^{-2}\text{s}^{-1}$
- Possibilities of having more than one interaction region

Realization Proposals



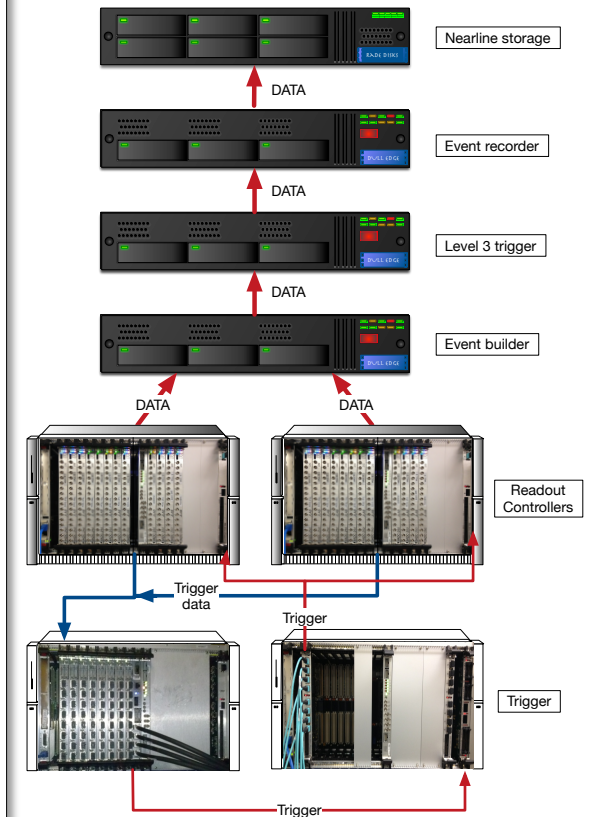
Example EIC detector design

- 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.



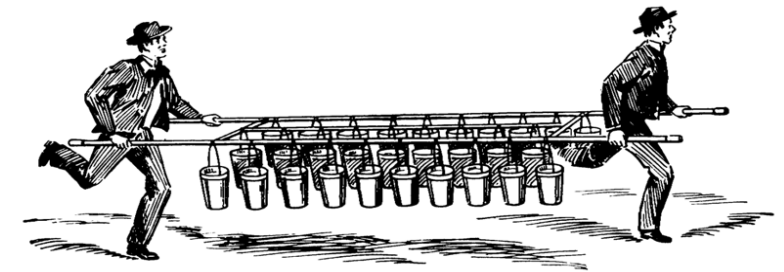
EIC DAQ guesses

- The Vertex detector could be 20-50 M channels.
 - Assume average 1% occupancy rate **~240 GB/s. (yes bytes)**
- 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 rate ~5 GB/s.
 - Fair agreement with CLAS12 and GLUEX if we were to scale them up to 100 kHz and 1% of 1M channels.
- How would you read out a detector that generates 240 GB/s ?
 - **Dealing with the Vertex Tracker dominates the design of the DAQ.**



Streaming mode

- In traditional triggered readout:
 - Data is digitized into buffers and a trigger, per event, starts readout.
 - Parts of events are transported through the DAQ to an event builder where they are assembled into events.
 - At each stage the flow of data is controlled by “back pressure”.
 - Data is organized sequentially by event.
- Traditional pipeline readout increases bandwidth by moving blocks of events.
- In a Streaming readout:
 - Data is read continuously from all channels.
 - Validation checks at source reject noise and suppress empty channels.
 - The data then flows unimpeded in parallel channels to storage or a local compute resource.
 - Data flow is controlled at source.
 - Data is organized in multiple dimensions by channel and time.



Streaming advantages

- The lack of a trigger means that:
 - Potentially useful physics is not discarded.
 - Run groups of experiments in parallel.
 - The system is simplified.
 - Readout speed is independent of detector response time.
 - Flow control at data source not via backpressure.
- Parallel timestamped streams mean:
 - System is robust against minor hardware or firmware glitches.
 - Can use different analysis techniques such as looking for hit patterns rather than reconstructing events.
- Requires robust and accurate time stamp generation and distribution.
 - Is still a simpler task than an online trigger.

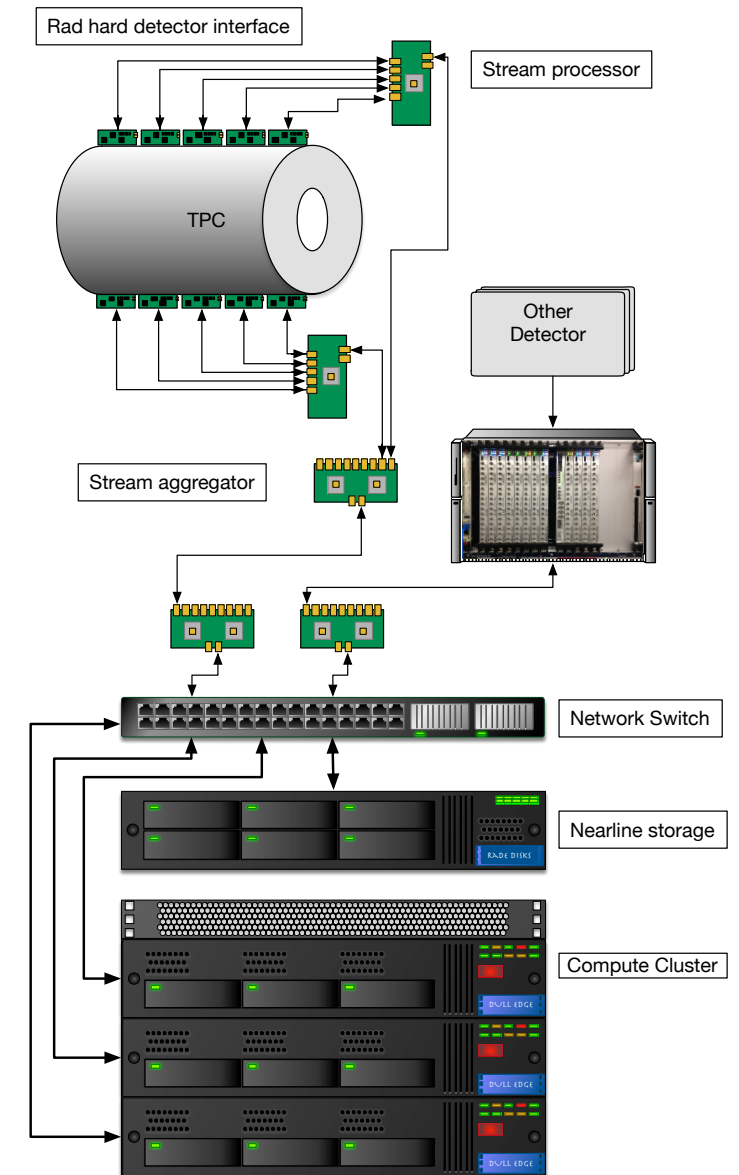


What does streaming readout solve?

- TDIS
 - High data rate from TPC handled as parallel streams – no one stream handles the full rate.
 - Electron data is it's own stream and is matched up with proton candidates offline – removes high rate event building issue.
- SoLID
 - High rates from GEMs handled in parallel – no real time event building.
 - Sector edge effects and correlation with other detectors is handled in software offline.
- EIC
 - All of the above.
 - New data analysis modes – for example looking statistically at hit distributions instead of event-by-event analysis.
 - Analogy – crystallography looks at diffraction patterns rather than ray tracing every x-ray through a crystal.

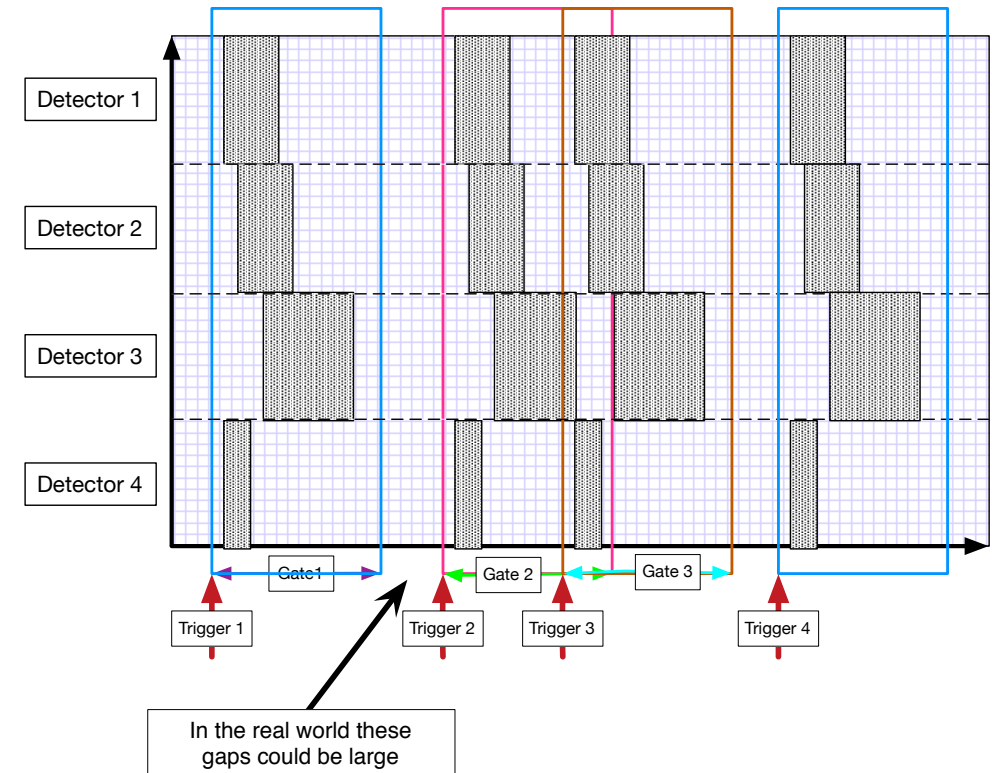
Generic stream based readout

- Raw data from the detector is continuously streamed
 - Rad hard detector interfaces convert detector signals into serial stream.
 - Instead of selecting ADC samples using a gate we take all samples and do the gate in software/firmware.
- Rest of the system uses generic hardware that manipulates streams
- We can reject noise and “zero suppress” using a firmware to reject time periods unassociated with events.
- Data packets are packed into an efficient standard format and sent over a standard hardware interface.
 - If all components use the same hardware interface and format we have a “plug and play” system!!
- Individual components should be cheap!
 - One of our VXS crates with support boards such as CPU and trigger cards costs ~\$90k!
- Stream data via a commercial standard like Ethernet.



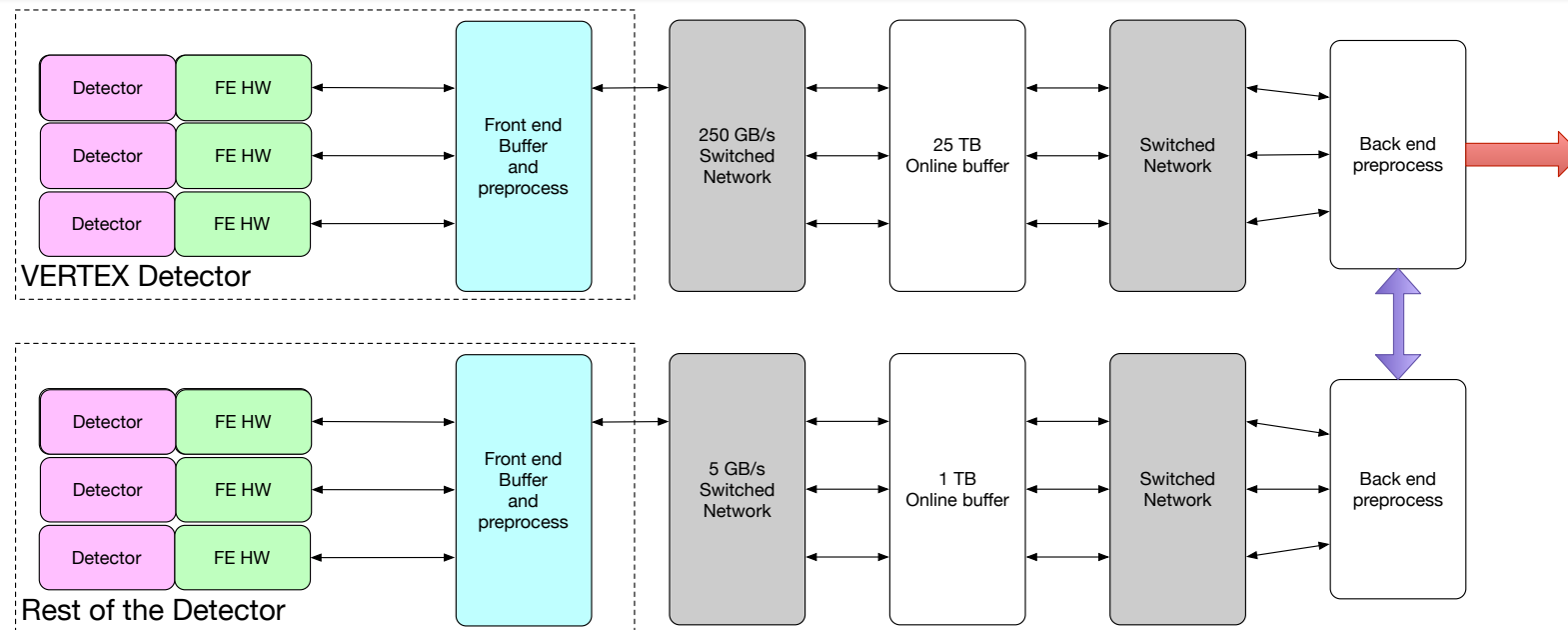
Concept – analysis software

- Packets containing data are stored indexed by time and channel.
- How to store and process data in this form? We are used to 1D files of events, this is a 2D structure indexed by detector and time.
- Reconstruction software requests data for a particular channel based on a time window where the data associated with an interaction is likely to be found.
- In this example detectors 1 and 4 are the used in software to generate a virtual trigger.
 - The data belonging to the rest of the even falls within a time window after the trigger, similar to the gate many slides ago.
- In this example triggers 2 and 3 are close together.
 - The request for data from trigger 3 gets part of trigger 2 data from detector 3.
 - In this case it is resolved because that data appears too early in the region of interest to be from the trigger 3 event.
- Note that this does not represent a layout in storage. Areas where there is no data would be compressed.



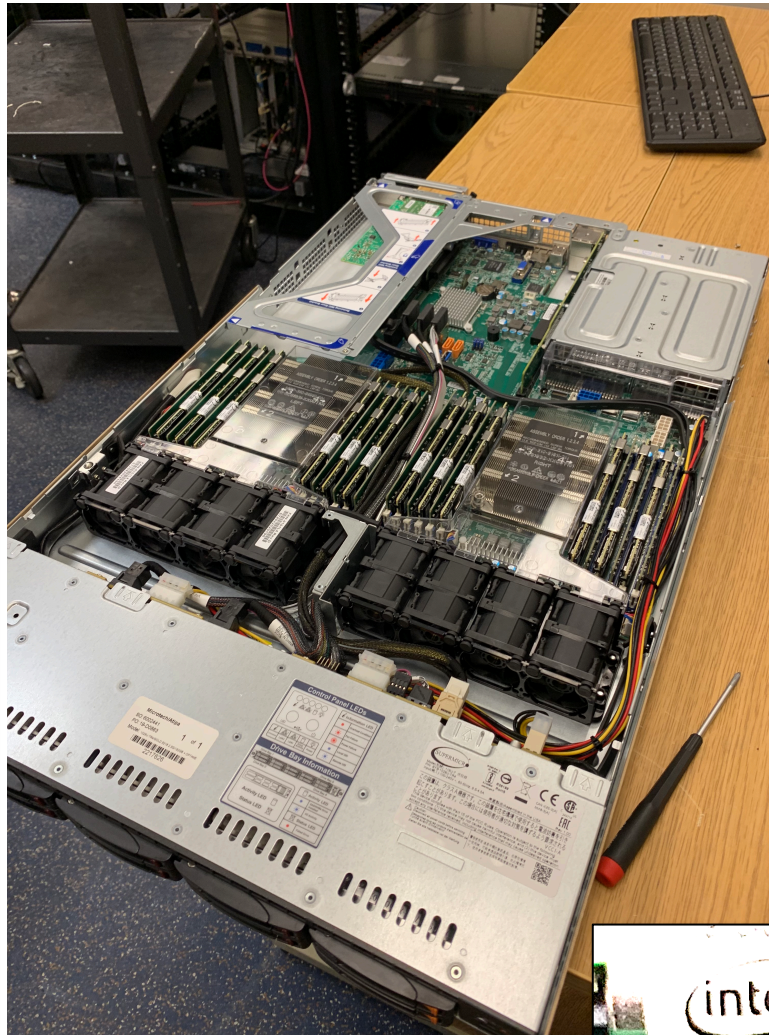
EIC detector readout with streaming DAQ.

- Read Vertex Detector in parallel streams into fast buffers.
 - Say 25 front end buffers at 10 GByte/s (Using today's 100 Gbit/s HW).
 - Main Online buffer is 25 computers 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 (ROI) 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



It's all real

- This machine has:
 - Dual 16 core 2.3 GHz processors
 - 1.5 TB of memory.
 - 1 TB solid state storage.
 - 100 Gbit/s network interface.
- Bought as R&D machine for streaming DAQ.

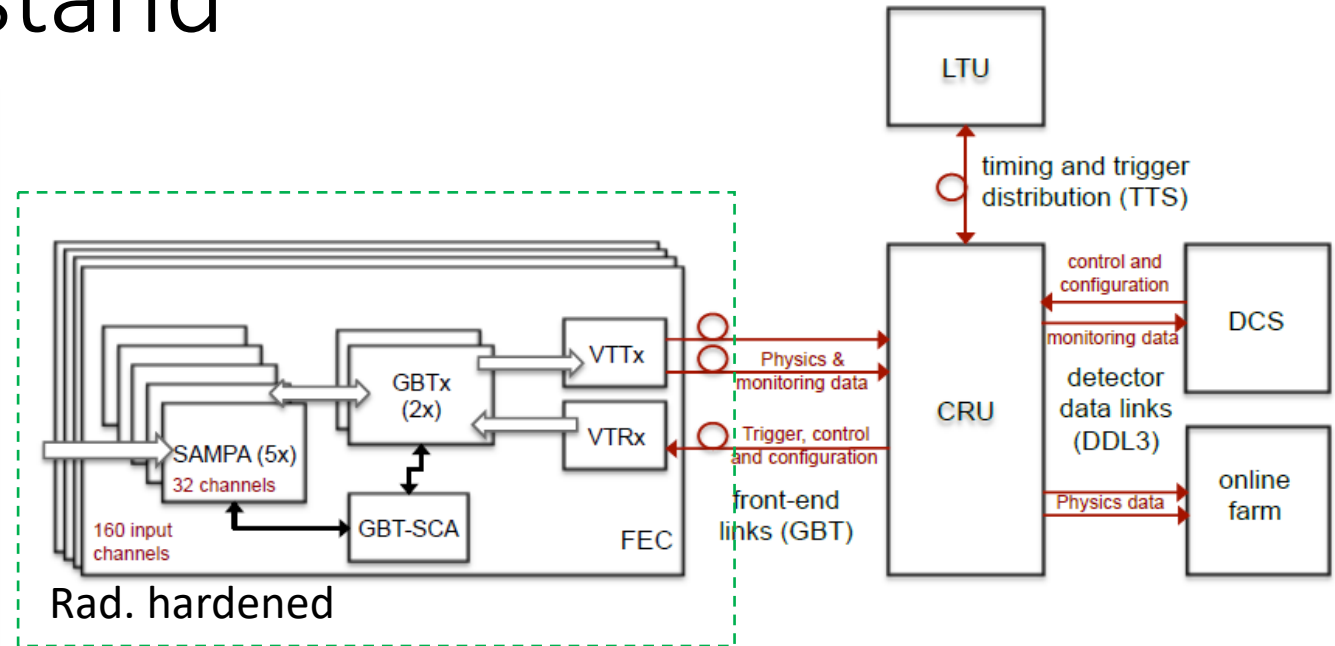


Where do we go from here

- JLab has several proposals, TDIS and SoLID are examples, that would benefit from streaming mode readout.
- The technology, both hardware and software is available and affordable to allow us to move to this mode.
 - Proven by LHC readout and adopted at other labs.
 - Simpler and more robust than the current readout mode.
 - Other advantages over pipeline readout already mentioned.
- We will develop a test stand to investigate techniques and technologies.
- We will develop a way to integrate legacy hardware – CLAS12?
- The goal is to use this mode for TDIS.
- If pilot system is successful:
 - Investigate moving CLAS12 and GlueX to this mode.
 - Use for SoLID and future JLab experiments.
 - I hope we may see new or modified proposals based on this.
 - Use for EIC.

JLab TDIS TPC test stand

- Fast Track test stand based on ALICE TPC readout hardware.
 - Use as many components of the ALICE TPC readout/control chain as possible.
 - Design uses custom SAMPA 32 channel readout chip.
 - Validate use of SAMPA for TDIS.
 - Experience reading a detector in streaming mode.
 - Guide future R&D effort in this area.
 - Replace CRU with JLab design.



FEC – Front End Card

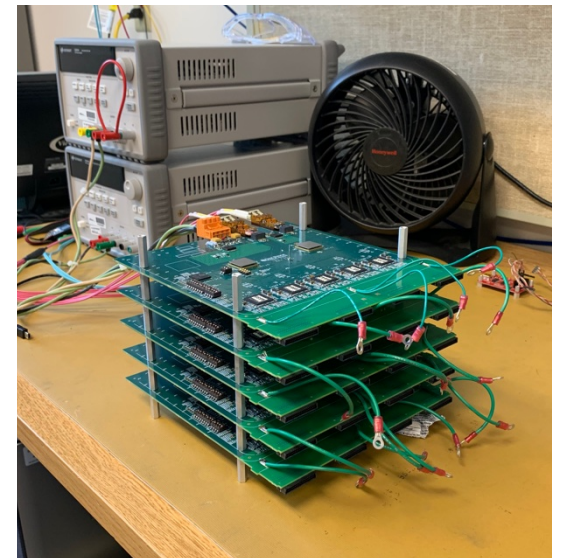
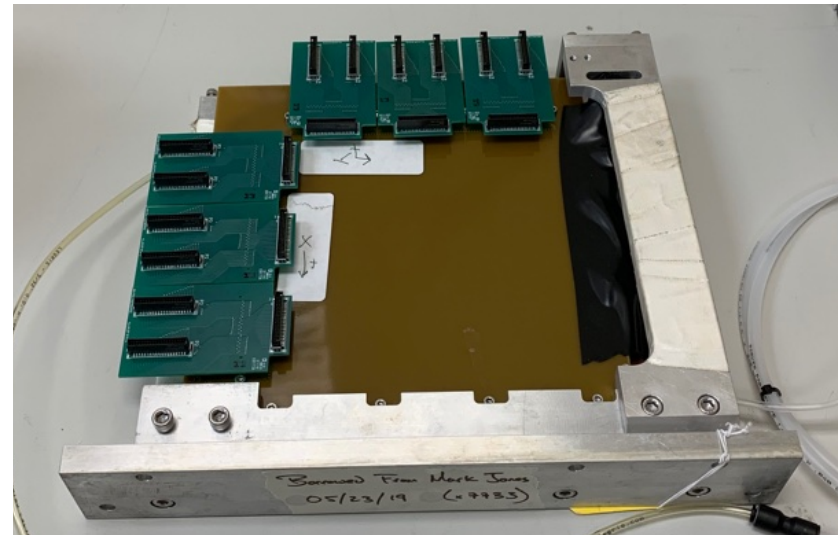
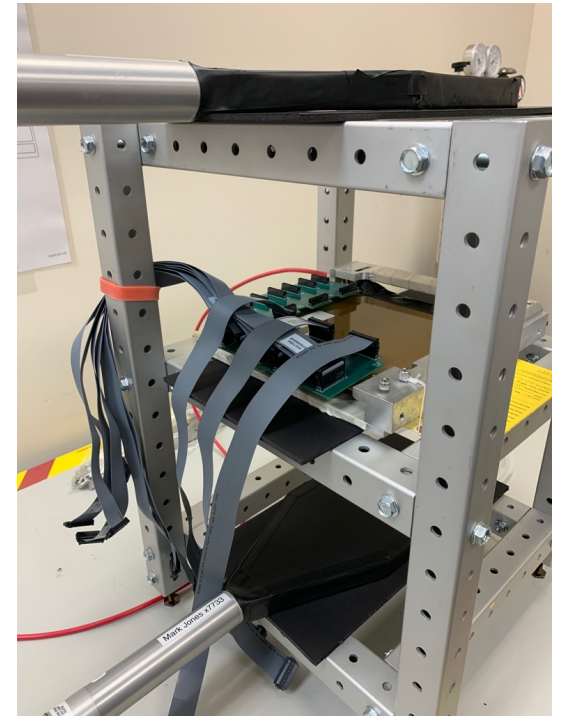
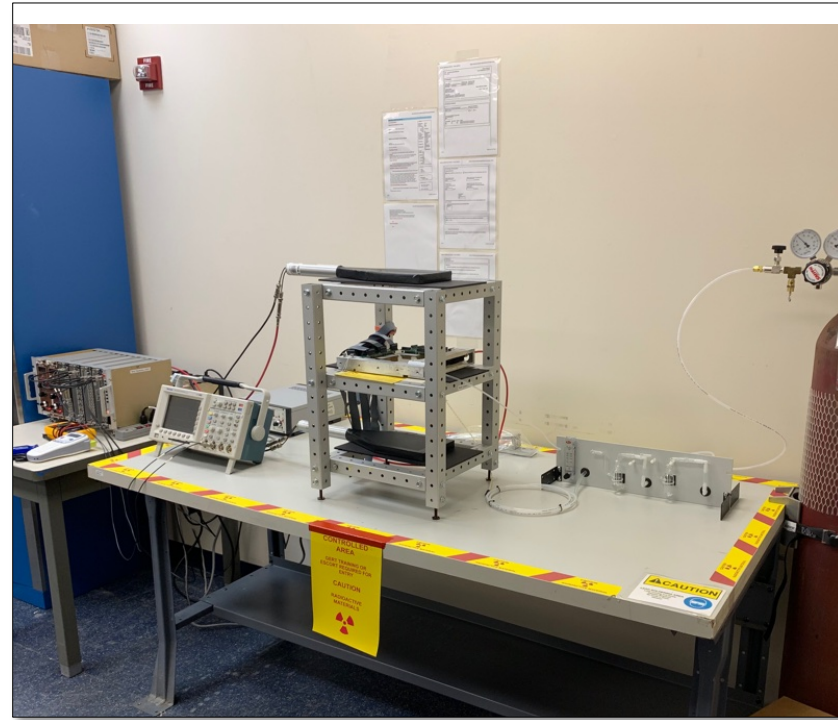
CRU – Common Readout Unit

DCS – Detector Control System

LTU – Local Trigger Unit

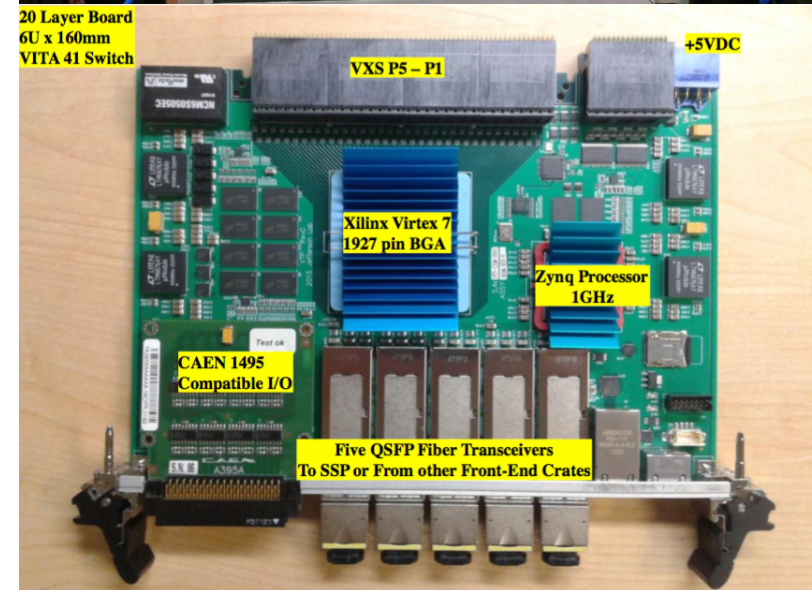
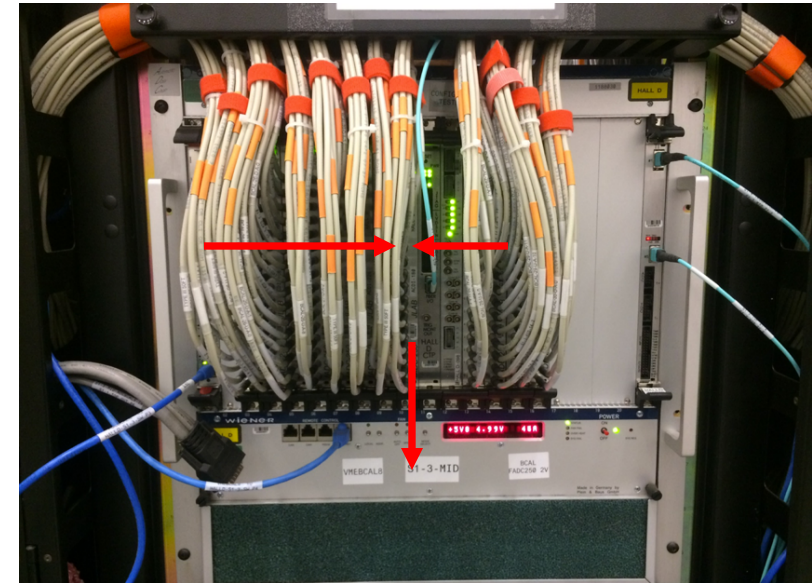
GEM test stand

- Top row left to right :
 - Test stand on bench in INDRA lab
 - Test stand closeup.
- Bottom right :
 - Closeup of GEM before mount
 - Front End Card stack.



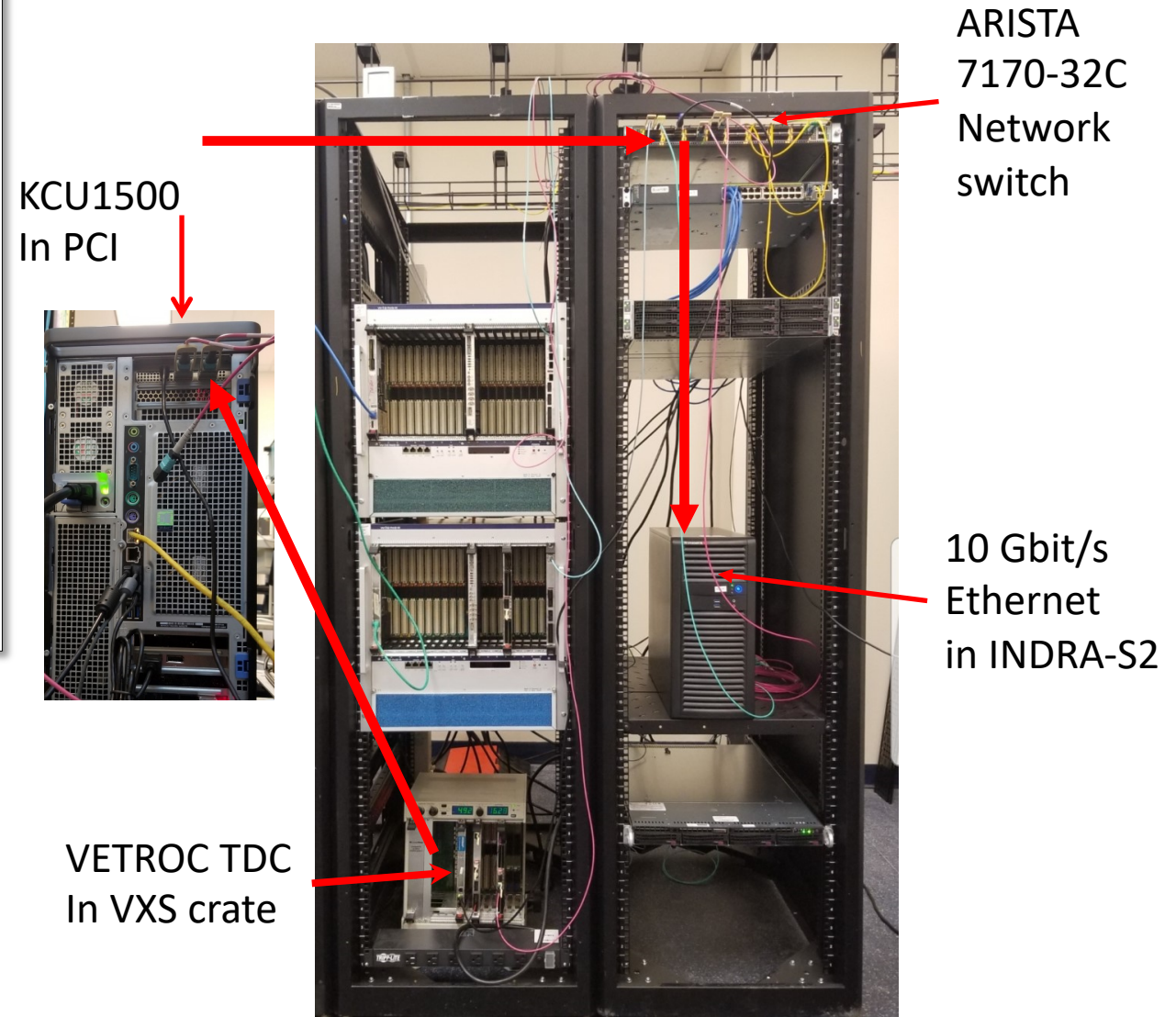
JLab – DAQ and Electronics R&D

- The VTP board is able to read 250 MHz fADC via the VXS serial backplane and stream the data out over the front panel transceivers.
- Read a crate in this mode.
 - Zero suppression logic
 - Flow control
 - What comes out on the front panel fiber?
 - How to interface with DAQ?
- Read out some detectors in streaming mode!
- Integrate with TDIS test-stand.
 - Can we replace the CRU etc?
- Can we repackage to remove need for expensive VXS crates etc.
 - Cheap system for university groups!



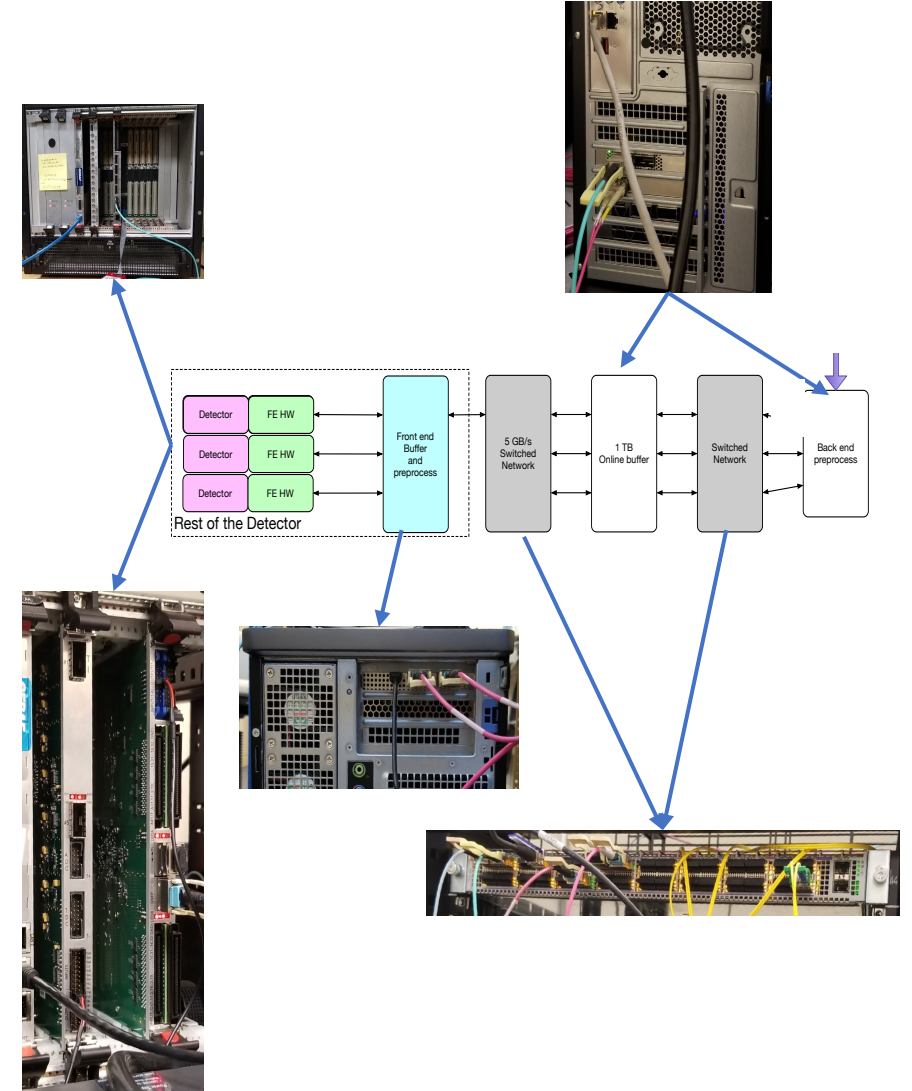
Readout of a VETROC 128 chan. streaming TDC

- Streaming TDC outputs on fiber.
- KCU1500 is a FPGA development board in PCI with fiber in and out.
- KCU1300 streams data via ARISA switch to PC.
- PC processes data.
- System tested a couple of months ago.
Working on software and firmware bugs.



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.
- We are testing a GEM detector readout
- We are testing clock distribution over network.
- 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)



Summary

- Data acquisition is constantly challenging.
 - Technology changes all the time.
 - Physicists think up experiments with tougher requirements.
 - The boundary between hardware and software is fluid and depends on what is available when a system is implemented.
 - There is always some R&D time to discover new algorithms and techniques.
 - To be honest, we do it because it's fun and we get to play with all of the cool toys!

Expect the unexpected

- Engineer: I told the Captain I'd have this analysis done in an hour.
- Scotty: How long will it really take?
- Engineer: An hour!
- Scotty: Och, you didn't tell him how long it would **really** take, did ya?
- Engineer: Well, of course I did.
- Scotty: Och, laddie. You've got a lot to learn if you want people to think of you as a miracle worker.

