Computing Trends in Nuclear Physics

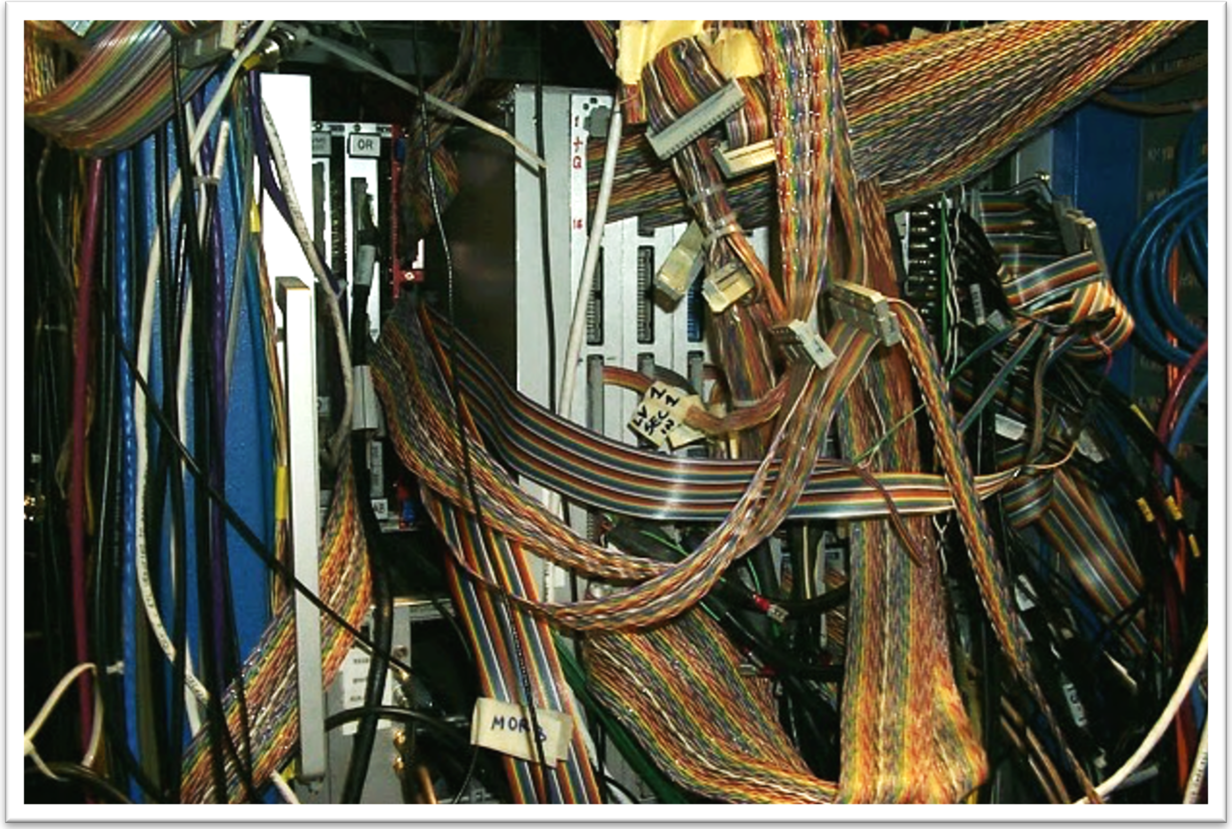
DAQ, Streaming, calibration and triggering

The Online World

Graham Heyes

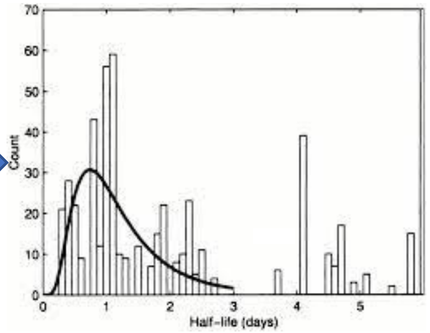
Head of Scientific Computing

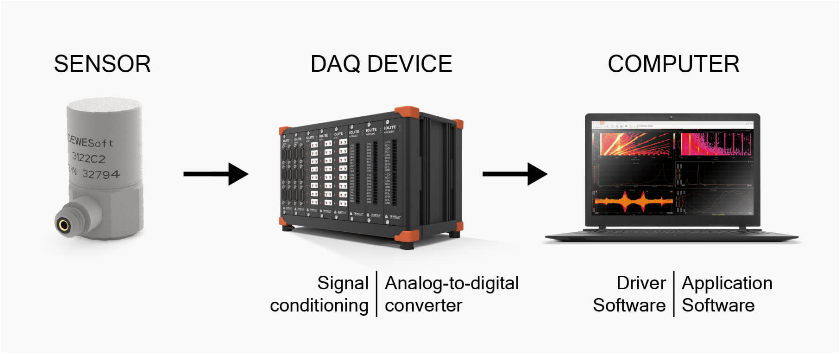
Computational Science and Technology Division



Part 1 -DAQ - Data Acquisition

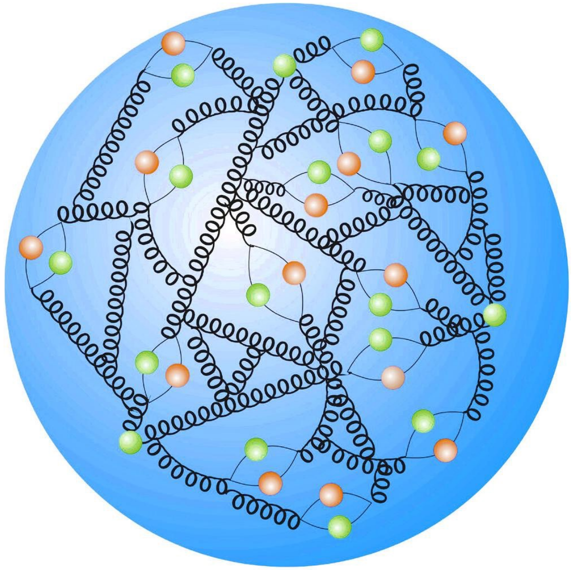
- Data Acquisition
  - Data acquisition is defined as the process of collecting and organizing information.
  - 100+ years ago research into radioactivity literally involved counting the number of flashes of light per time interval.
    - Experiment control rooms are still called “The Counting House”.
  - Nowadays electronic sensors measure physical quantities that are processed on computers
- In the context of Nuclear Physics experiments there are two types of data acquisition:
  - Fast data acquisition – high-rate data from large and complex detectors.
  - Slow controls – low-rate monitoring of conditions in the detector.
- This talk will focus on <u>fast data acquisition</u>.

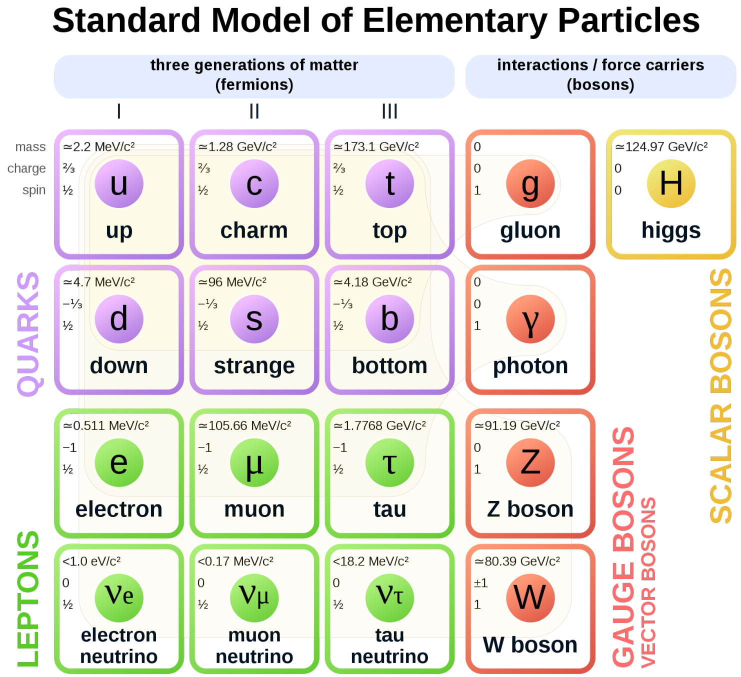




What are we measuring?

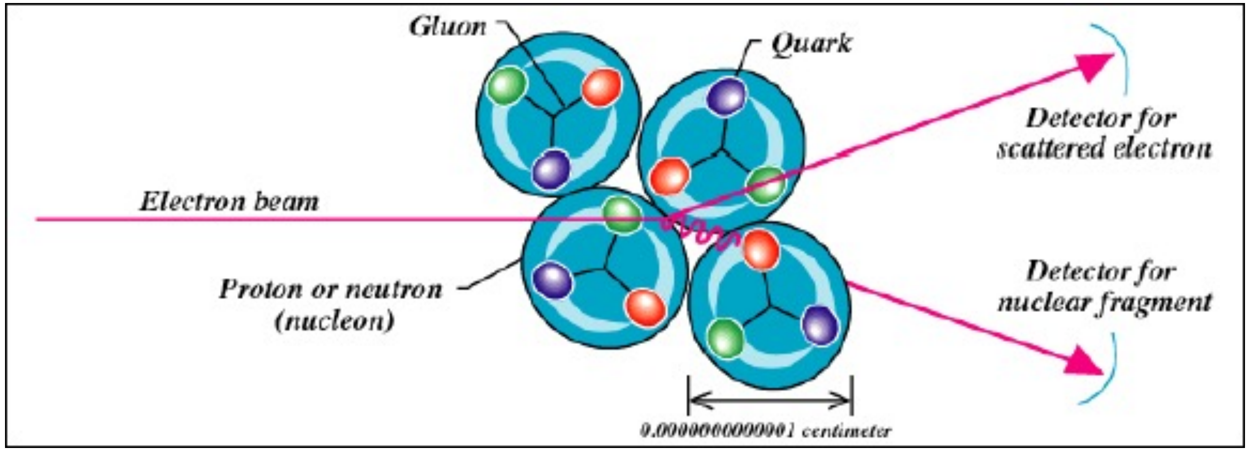
- At JLAB we investigate how the quarks and gluons that make up protons and neutrons behave when they are part of a nucleus.
  - We often see diagrams of a nucleus where the protons and neutrons are neatly stacked. In reality, it's a mess in there!
- At HEP labs, like CERN, there is more of a focus on fundamental particles, of which quarks and gluons are examples, and how the forces that shape their behavior work.
- 19th century analogy:
  - HEP = study of atoms.
  - NP = study of molecules.





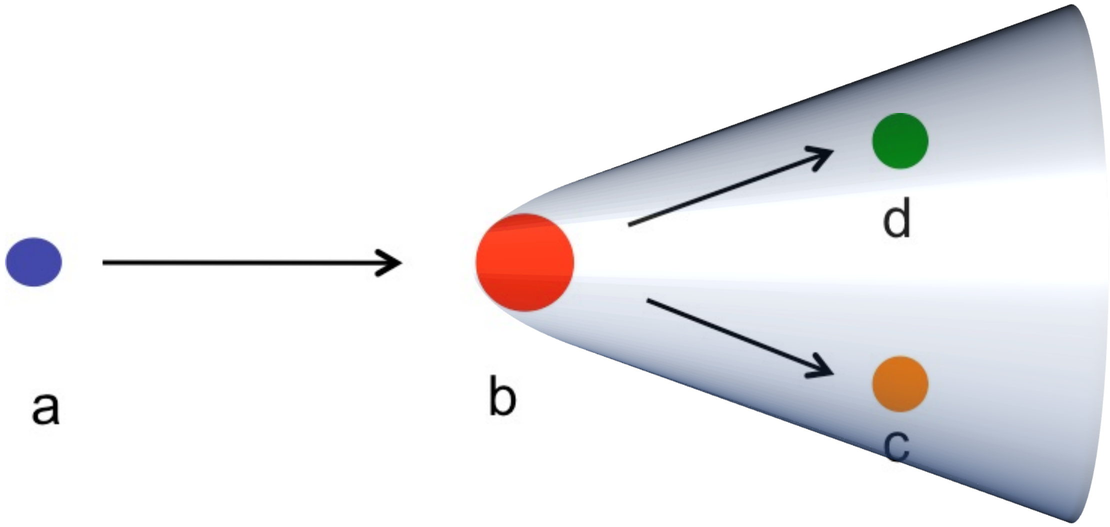
Nuclear Physics at JLAB

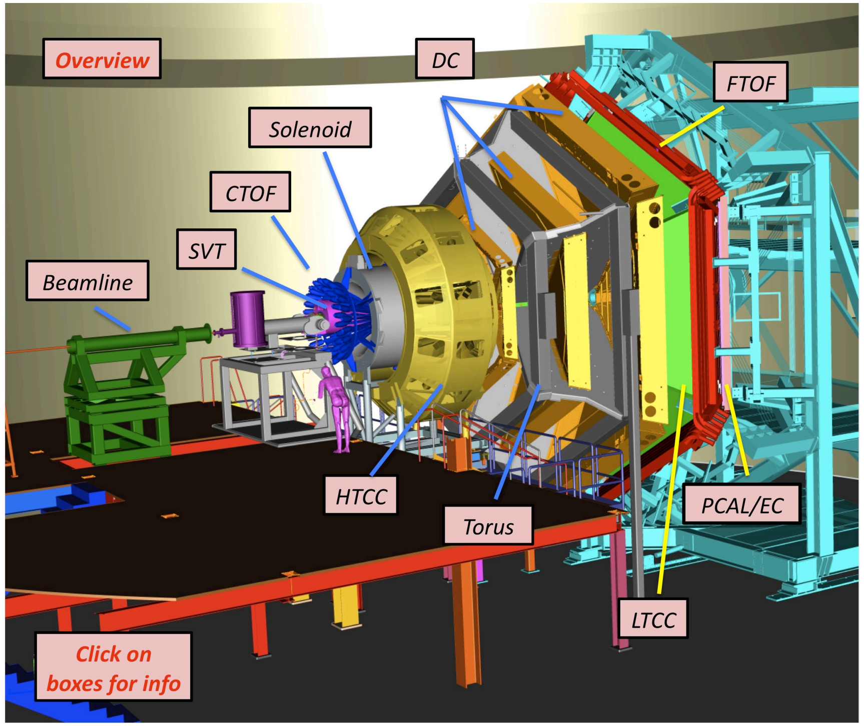
- The CEBAF accelerator generates a beam of electrons at up to 12 GeV.
- All the experiments at JLAB are fixed target.
- Electrons behave as a particles and waves.
- If the energy of the electron is high enough then its wavelength is short enough to resolve details of the nucleus at the quark-gluon level. It seems simple -
  - Hit a target with electrons.
  - Measure the energy, momentum, and type of particles produced by the interaction.
  - Statistically analyze many interactions.



Detectors for fixed targets

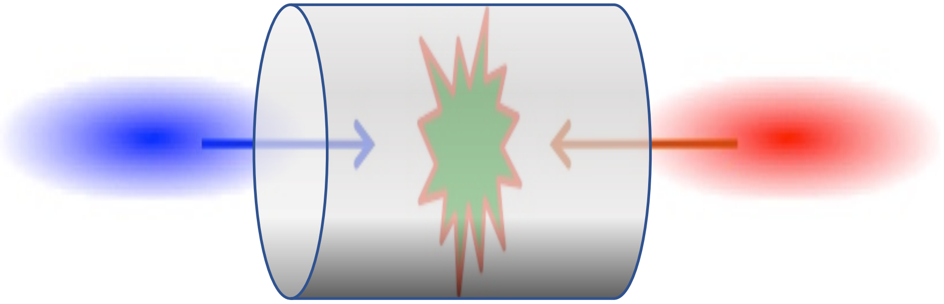
- In a fixed target experiment the particles from the interaction mostly travel in the beam direction. Detector designs take advantage of this by putting most of the detecting hardware downstream of the target.
- Interaction are simulated in software before the detector is designed so the optimum solid angle and types of detectors can be chosen.
- In a large experiment the detector contains many different types of sub-detector to measure different properties of particles as they pass through.
- Apart from close to the target most of the detectors are planes stacked in the beam direction.

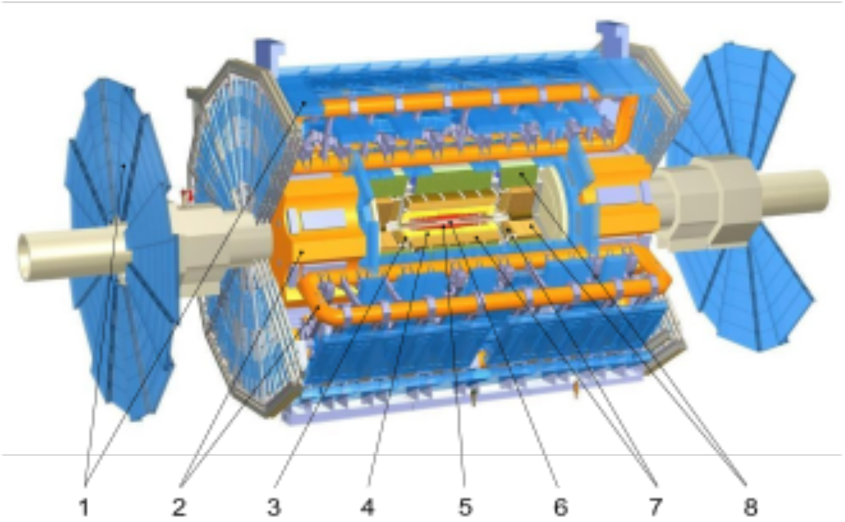


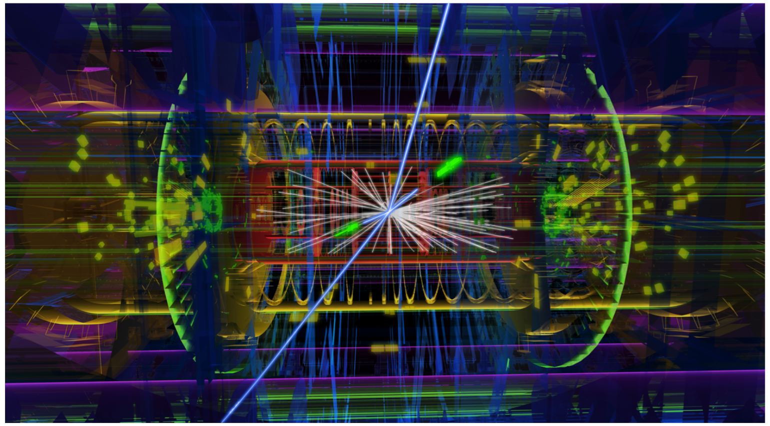


Detectors for colliders

- In a collider interactions produce particles with momentum distributions in a 3D volume around the target.
- Building a spherical detector is hard so most detectors are cylindrical with stacks of disk-shaped planar detectors at the end (referred to as "end caps").
- As with the previous example the detector has many different types of sub-detector hardware.
- Most detectors take advantage of the way that moving charged particles follow curved tracks in magnetic fields.
  - This is used to measure particle type and momentum.
  - In the case of colliders there is a complication for acquiring data because most of the particle detecting hardware is buried inside the magnet.

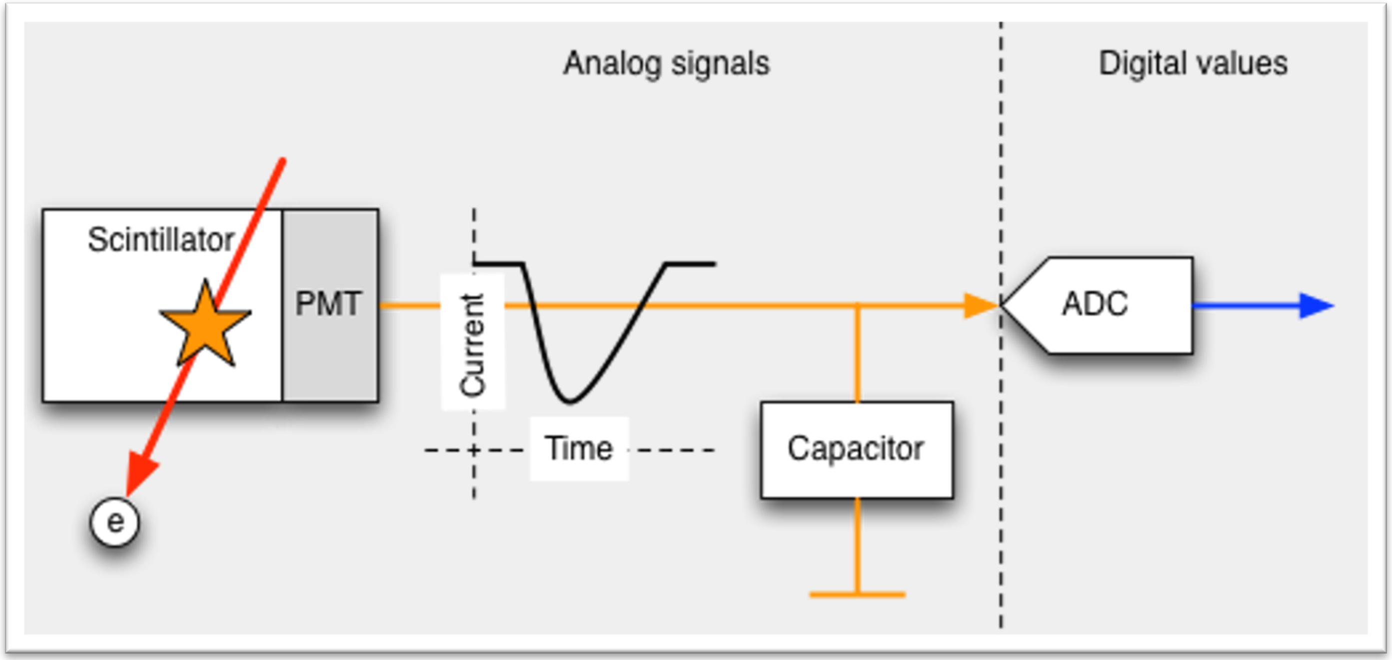






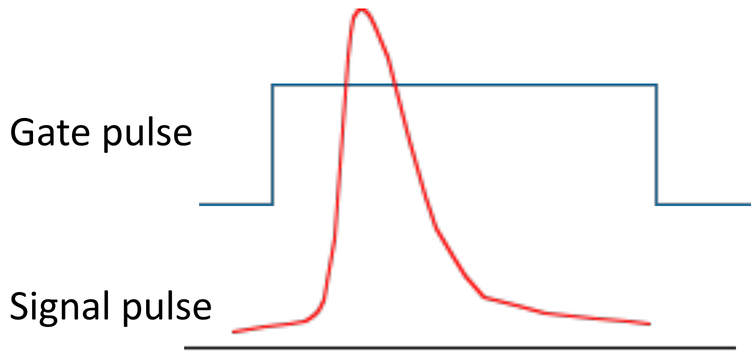
Detector example, a scintillator

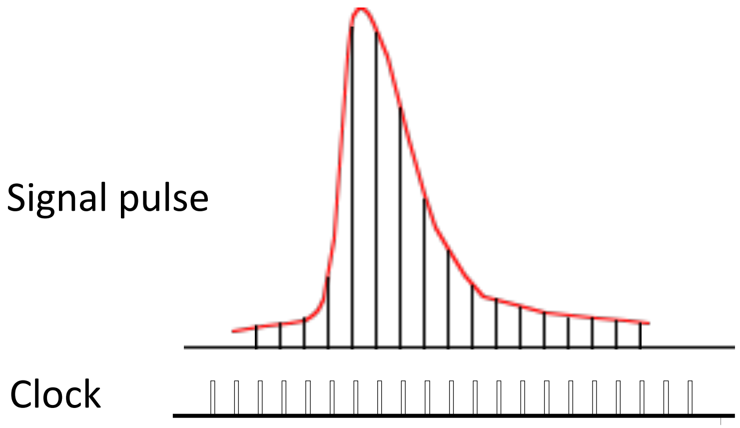
- A particle deposits energy in a scintillating material and causes pulse of light.
- A Photo Multiplier (PMT) converts the light pulse into a pulse of electricity.
- The pulse of electrical charge is captured to generate a voltage.
  - proportional to the area of the pulse and so the amount of energy.
- An Analog to Digital Converter (ADC) converts the voltage to a digital value.
- Three basic types of measurements are.
  - Charges or voltages – ADC (Analog to Digital Converter)
  - Times – TDC (Time to Digital Converter)
  - Counts of things - Scaler



Sampling vs Integration

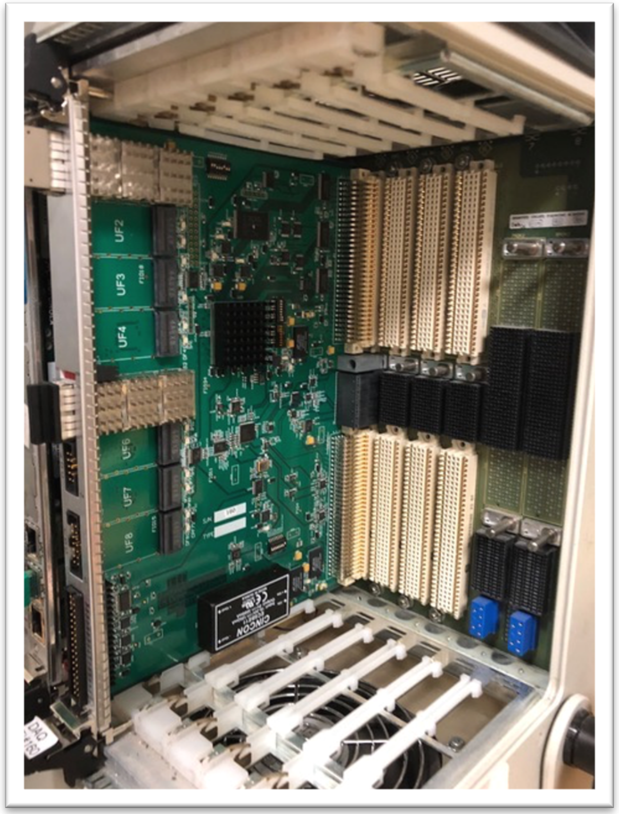
- A traditional "integrating" ADC takes 6 to 10 μsec to digitize a pulse. It relies on a gate pulse, generated by other detectors to mark the region of interest over which the signal is integrated.
- This type of ADC generates a single measurement representing the charge sum during the gate.
- A Flash 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.

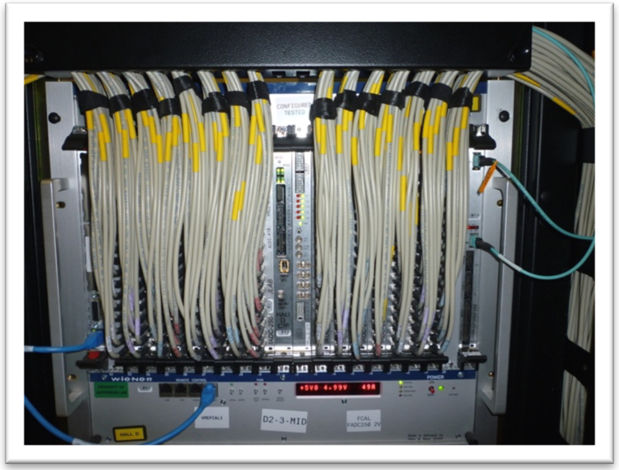




Putting together a system

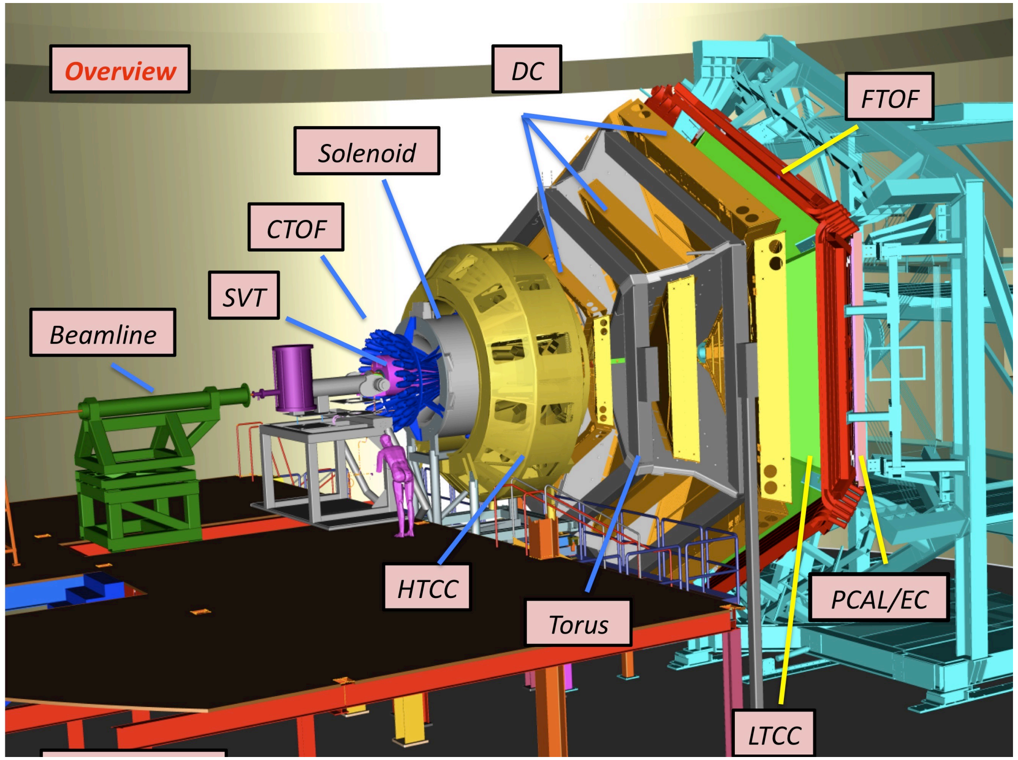
- There are many data thousands of detector channels in a NP detector.
- We need thousands of channels of ADCs and TDCs as well as other electronics.
- The current standard is to build electronics with several channels per board. Several boards then plug into some sort of backplane to connect them to a processor that reads the data off the electronics
- At JLAB we have ADC boards with 16 ADCs on one board. We commonly use VXS, which is a variant of the VME standard for interconnecting electronics.
- Up to 16 ADCs per VXS crate = 256 channels of readout.
- The large JLAB experiments need tens of thousands of channels and so have 50-100 crates of electronics.
  - Data acquisition systems can be physically big things.

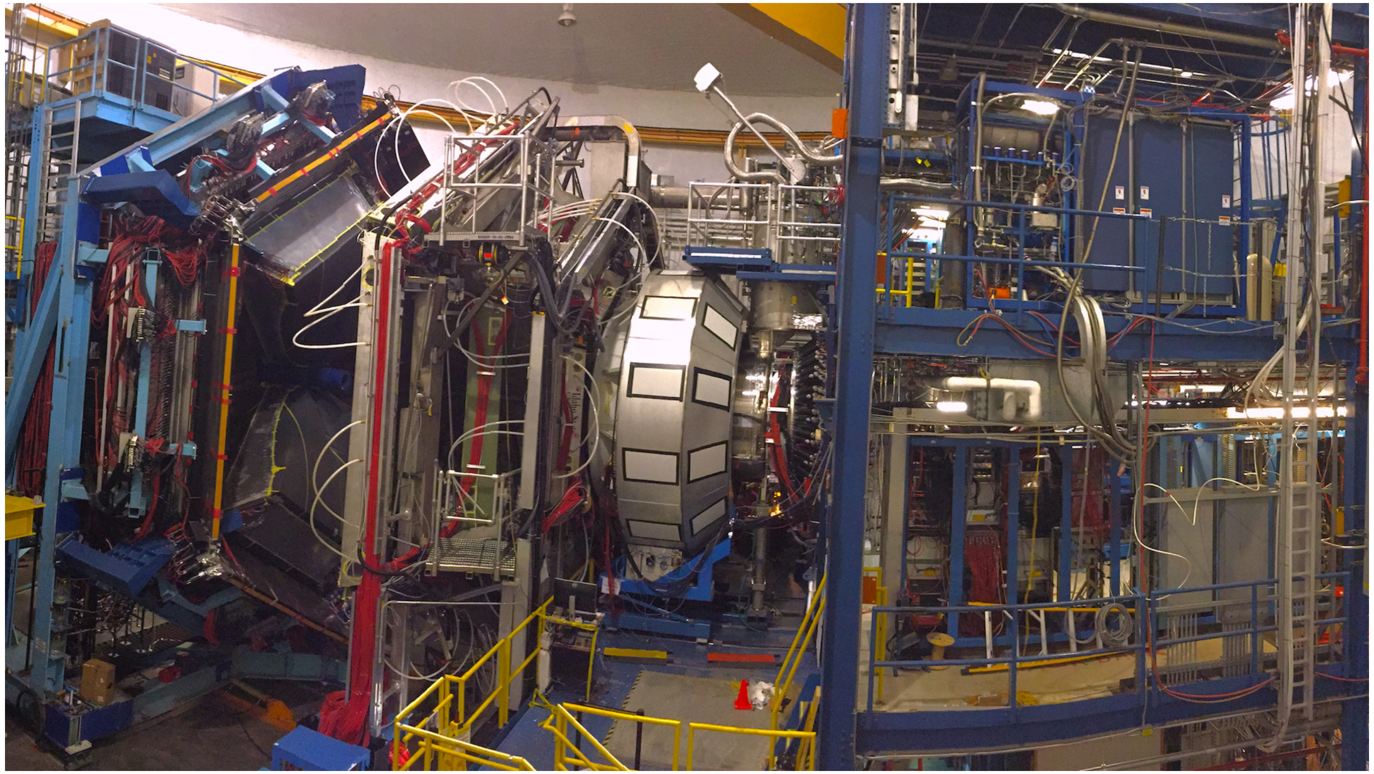




CLAS12

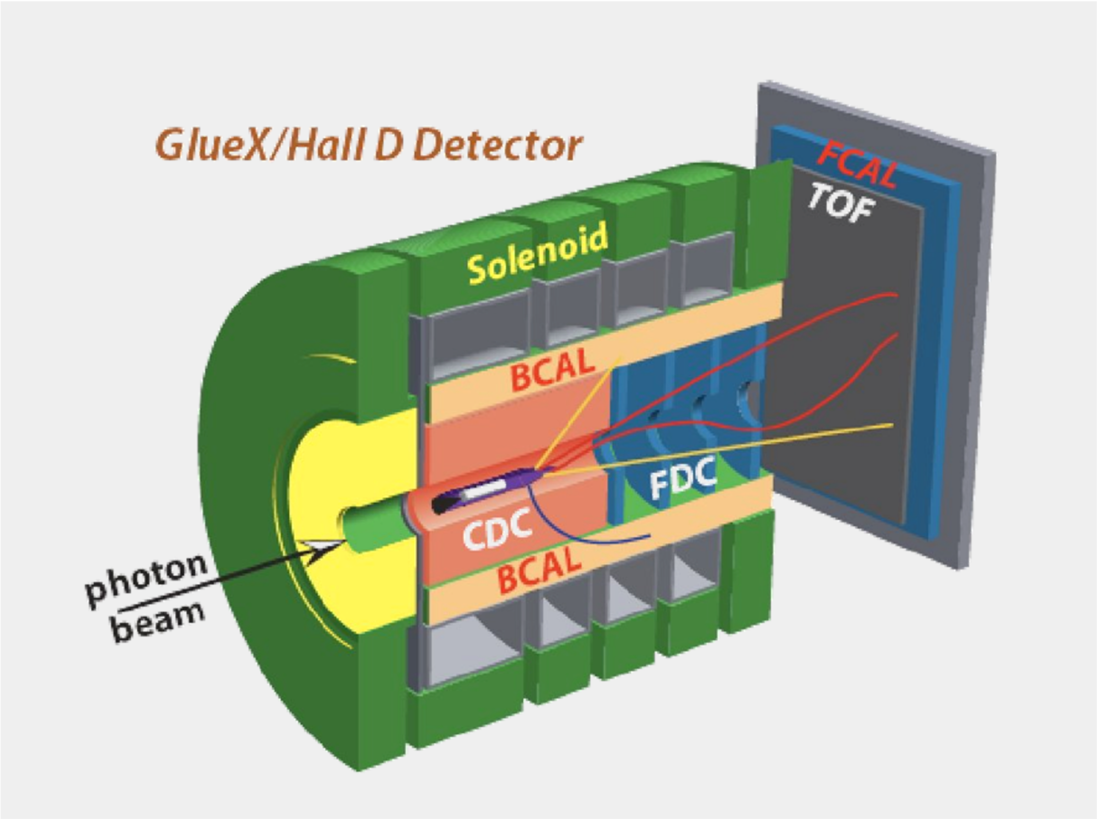
- The CLAS12 detector is a cone shaped detector.
  - The photo on the right shows the real detector. The diagram on the left shows the planes of detectors.
  - The blue frame on the right of the photo holds most of the DAQ system electronics.

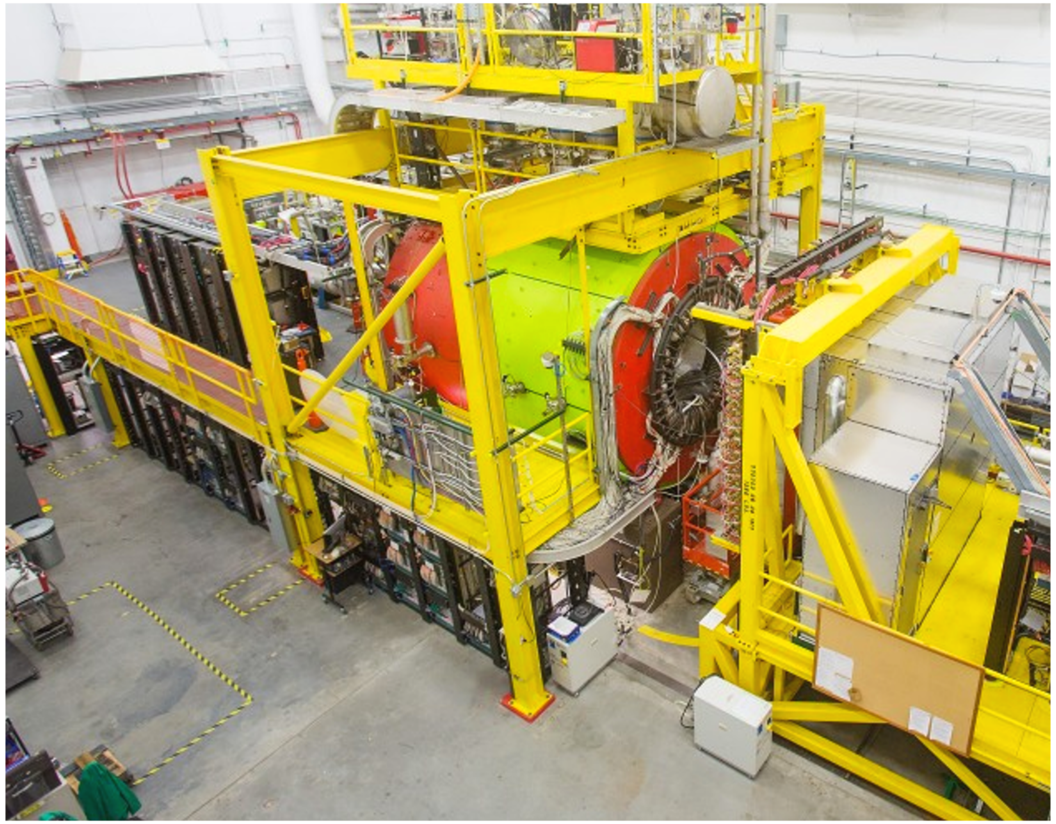




GLUEX

- The GLUEX detector is cylindrical with the target (white cylinder) offset from the center. Most of the detectors are forward of the beam direction.
  - The picture on the right shows the real detector with a schematic on the left.
  - Most of the DAQ system is in the racks under and on top of the yellow frame.



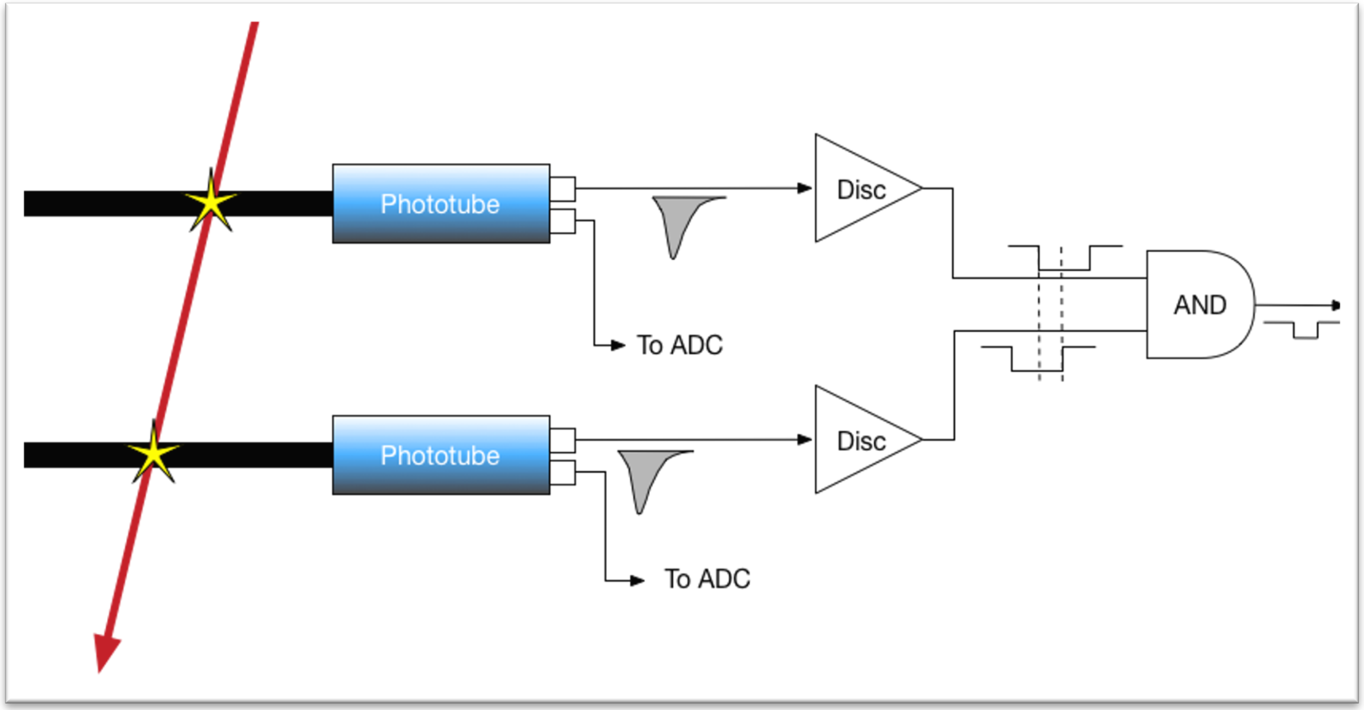


Part 2 - Triggers

- The story so far:
  - We want to study how particles in a beam interact with a fixed target or another beam.
  - Various types of particles are arranged in a 3D volume to capture the products of the interaction.
  - The physical properties of these particles are measured by detectors and encoded in electrical signals as amplitudes, times, and counts.
  - Electronics measure these and produce digitized values that are recorded by a computer.
- There, data acquired, job done, we can all go home!
- Not so fast:
  - How do we know that any of this data coincides with a particle interaction?
  - Also, if you do the math, a single 250 MHz ADC channel generates 4 bytes every 4nS = 1 Gbyte/s so a 30,000 channel detector generates 30 Tbytes/s of data if everything is free running.
  - Maybe we need to narrow it down a bit – that is what triggers are for.

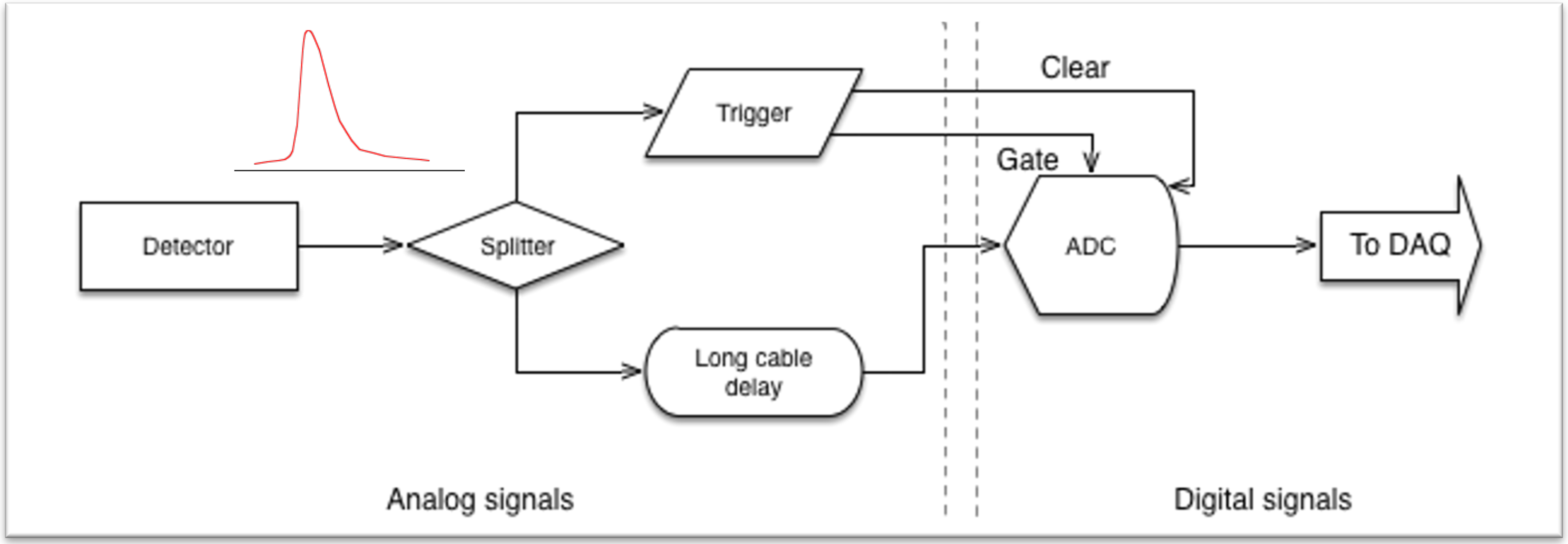
A Simple Trigger

- How do we know the signal on a particular channel came from an interaction?
  - Fortunately, we have more than one detector.
  - Combine data from different detectors to signal when an interaction occurred.
  - Algorithmically determine which are interesting.
- Example: a coincidence trigger
  - Two detectors generate a signal. Each signal is sent to an ADC and trigger logic.
  - The discriminator converts the analog pulse into a digital one with a width that corresponds to how long it would take a particle to pass from one detector to the other.
  - If two signals overlap within this window there is a good chance both signals came from the same particle.
  - The trigger electronics generates a pulse that enables the ADCs.
  - So, the ADCs only generate data if something interesting is happening.



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 are ~1 ns/ft so you can delay signals by precise amounts using cables of various lengths.
  - Matching cable lengths is very important.
  - The ADC processes one trigger at a time. It cannot process a new signal until the last is read or cleared.
  - This limits the trigger rate.
  - When combining data from fast and slow detectors the fast signals can need very long delay cables.

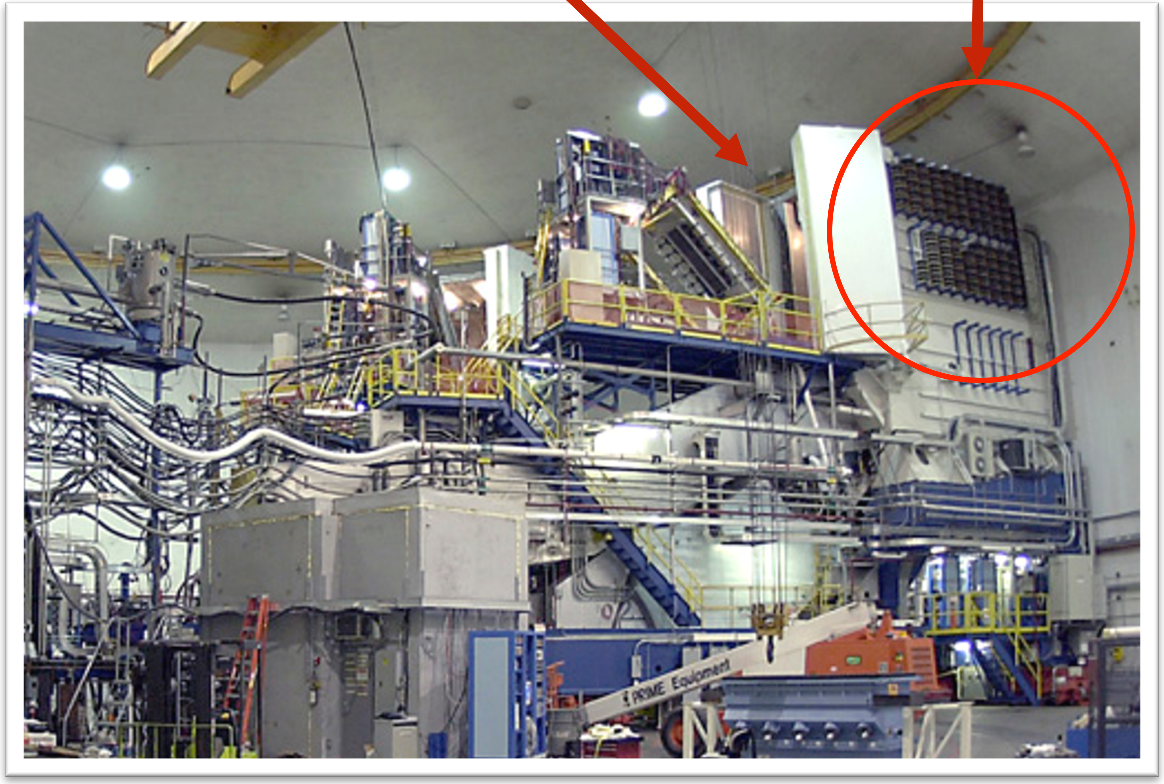


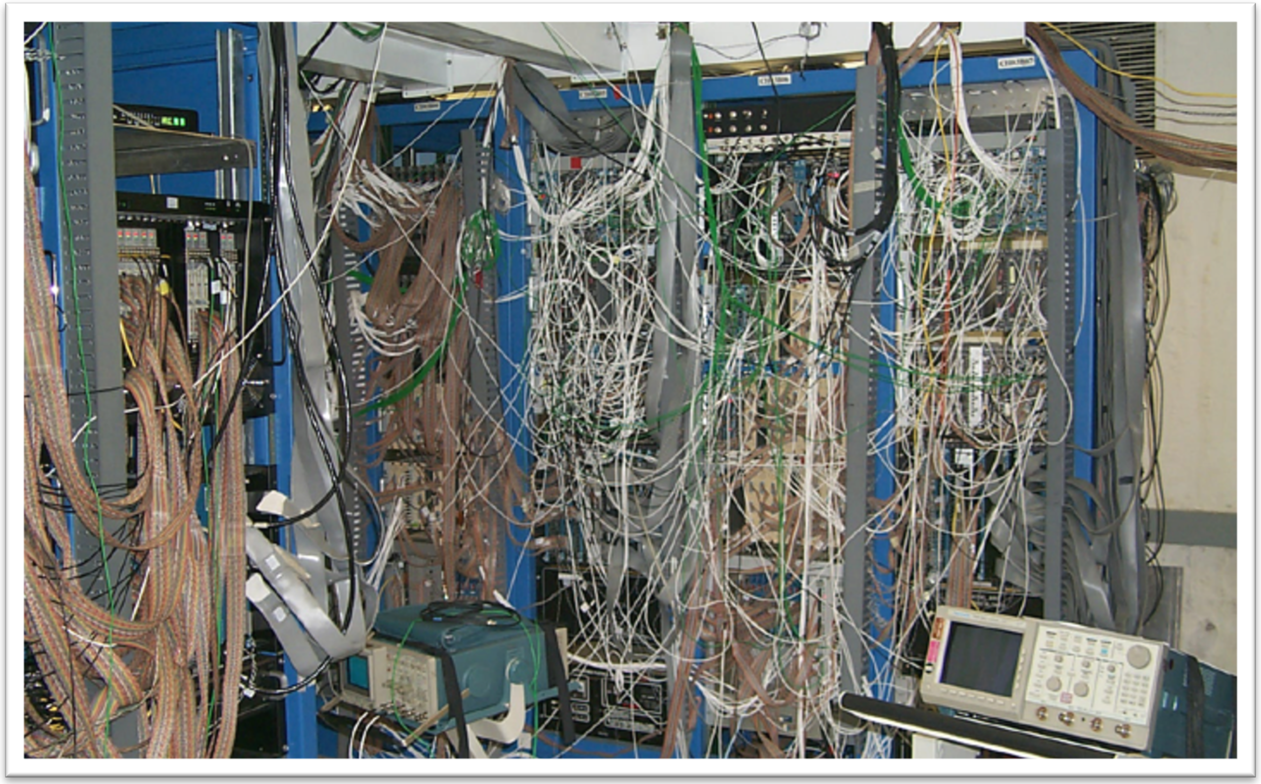
The practical problem with analog signal cables.

Detector hut

Delay cable

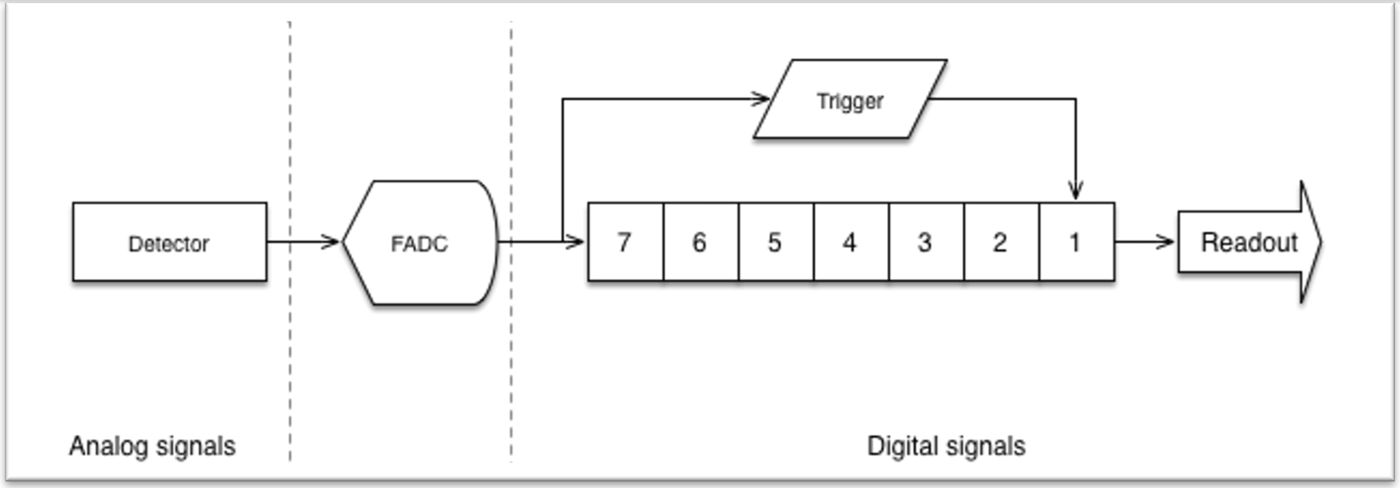
Using cable lengths to adjust timing
can lead to a big mess!





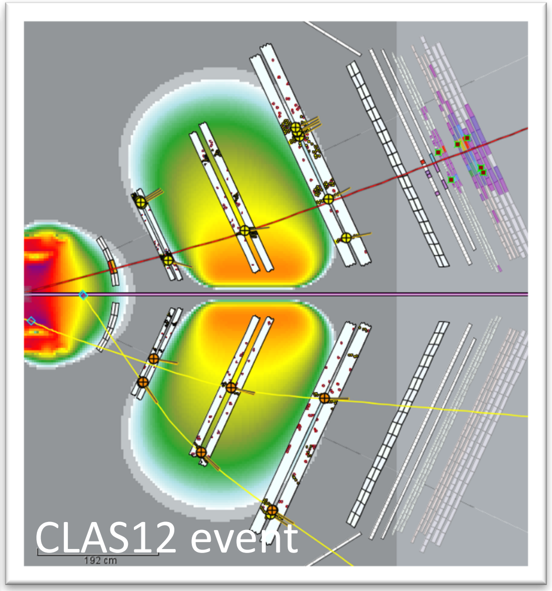
A solution - all digital pipeline triggers

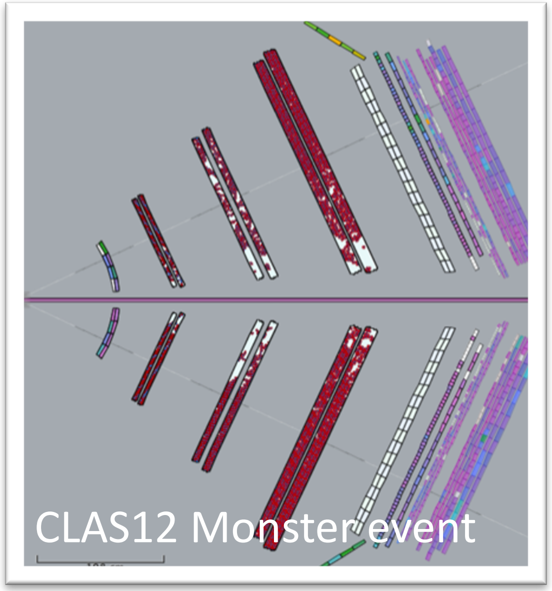
- Digital electronics lets us replace all that cable with digital memory.
- A Flash ADC digitizes at a constant rate - the Flash ADC sampling clock.
  - In the case of JLAB fADCs, 250MHz or one clock tick every 4 nS.
- In a digital pipeline we store a value in memory and increment the address every clock tick.
- For example, if the trigger logic takes 28 nS we know that trigger corresponds to measurements 28/4 = 7 cells earlier down the pipeline.
- The readout software can selectively read the memory to keep only the useful samples.
- The ADC can be digitizing at one end while the readout is happening at the other.
  - A big plus is that the ADC is almost always "live". It doesn't have to wait for the trigger logic.

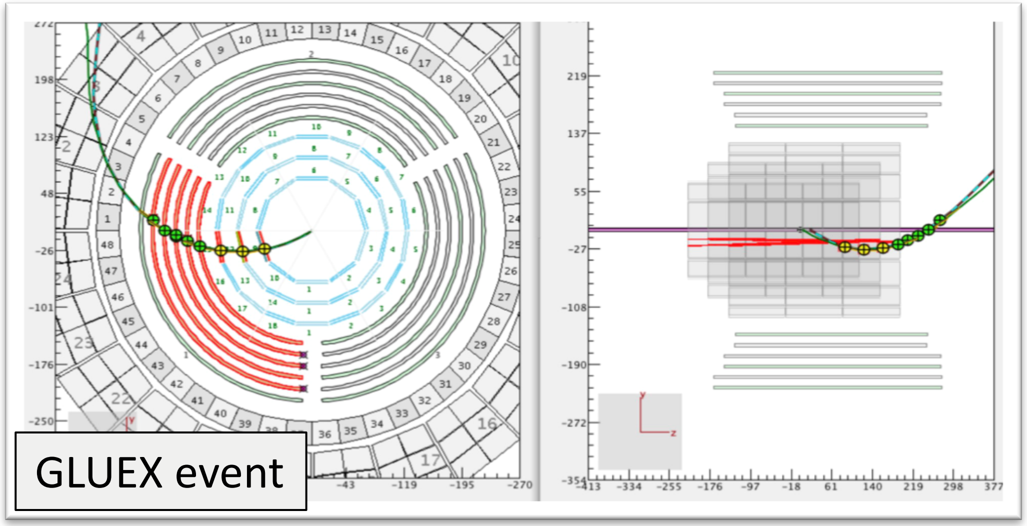


The event as a unit of data

- The trigger signals that there is enough useful data to indicate that an interaction has occurred.
- All the data from a single interaction is called “an event”. Important characteristics of events are:
  - Data from one event has no history. It doesn’t depend upon events that went before and doesn’t influence later events.
  - Triggers 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 :
    - Accidental signals unconnected with the event.
    - Electronic noise.
    - Distribution of physics event sizes.
  - There can sometimes be very large events.

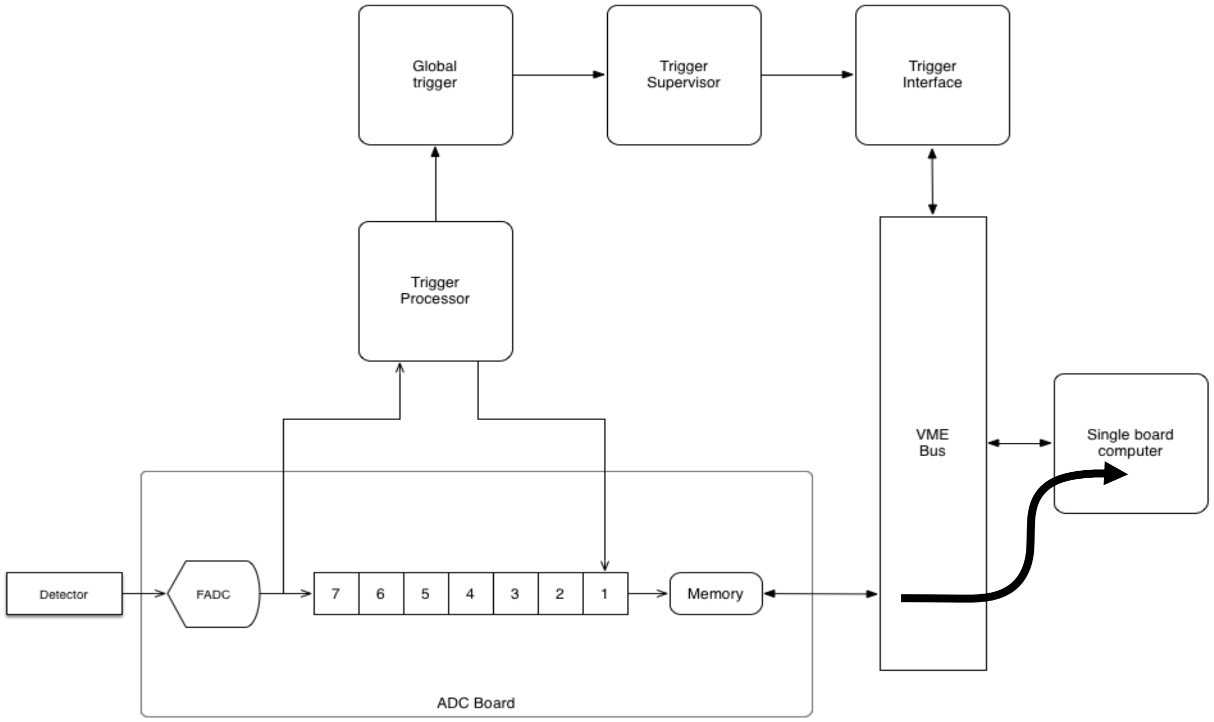






A global trigger system

- Even though the crate level trigger may indicate that an event is stored in the pipeline it may still not be an event that is useful for the science being studied.
- Most experiments use complex trigger algorithms involving several detectors.
- At JLAB we use a “pipeline global trigger system”.
  - Each ADC sends signals to a crate level trigger processor that runs algorithms using data from the single crate and passes the result to a global trigger processor.
  - The global trigger processor uses algorithms requiring data from several detectors.
  - The global trigger is distributed back to all the readout crates.
  - Depending on this trigger the event is read out or discarded.

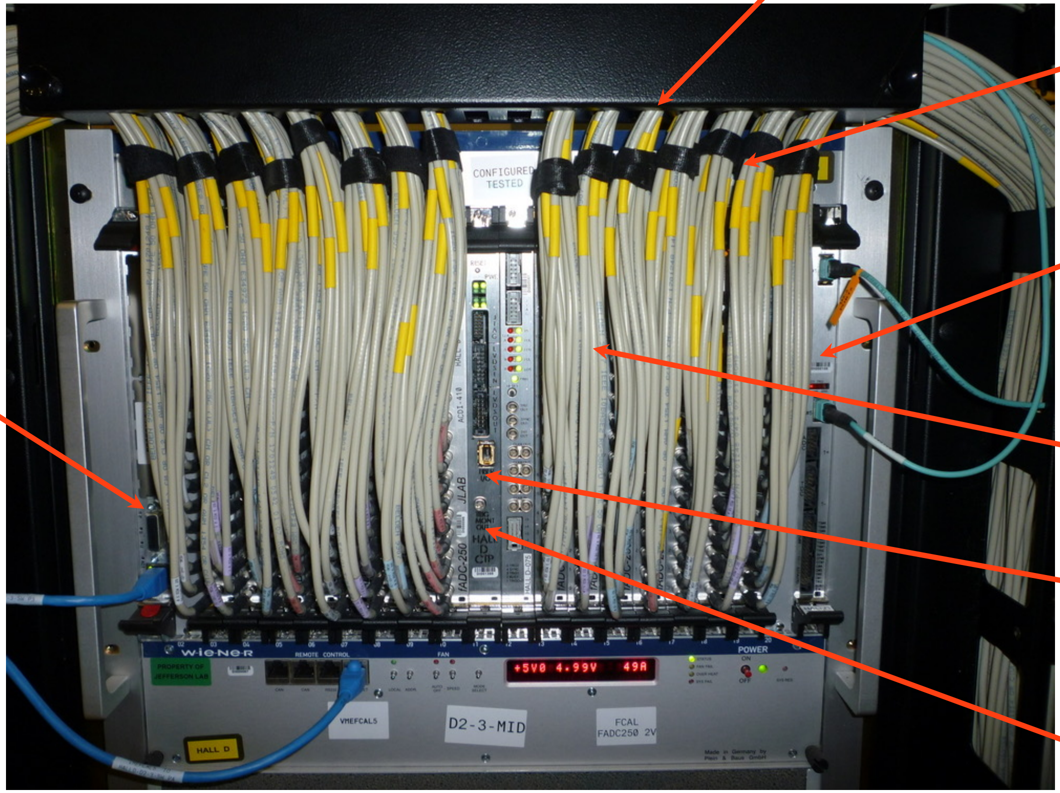


What it looks like – an ADC crate

16 channels per board

15 boards = 240 channels per crate

Intel CPU Read Out Controller (ROC) running Linux



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

What it looks like – Global Trigger Crate

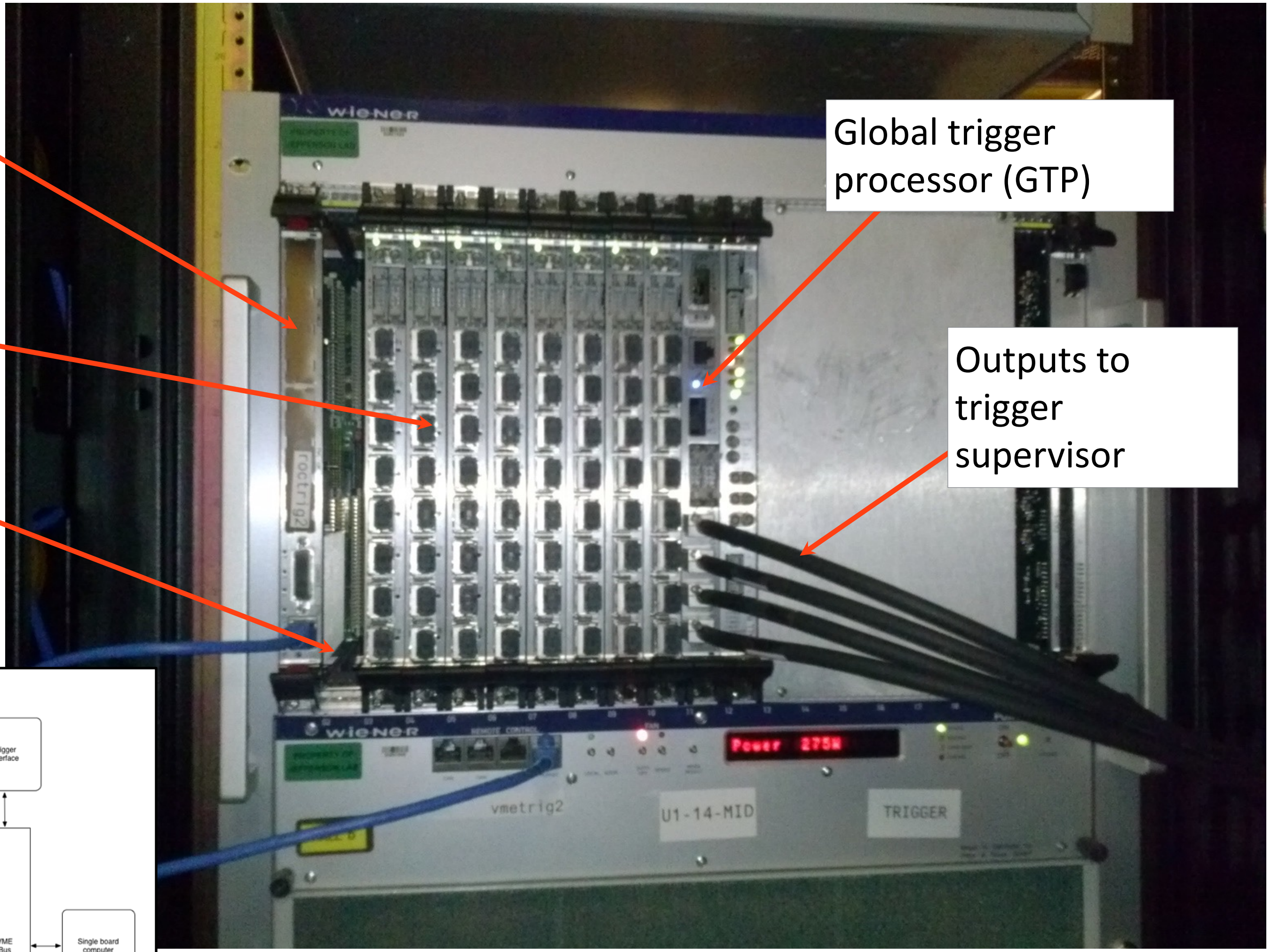
Intel CPU controller

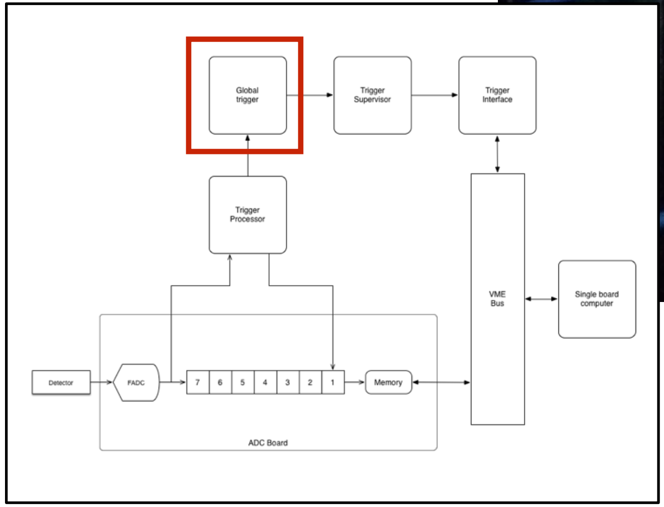
Sub-system processor board (SSP)

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

Global trigger processor (GTP)

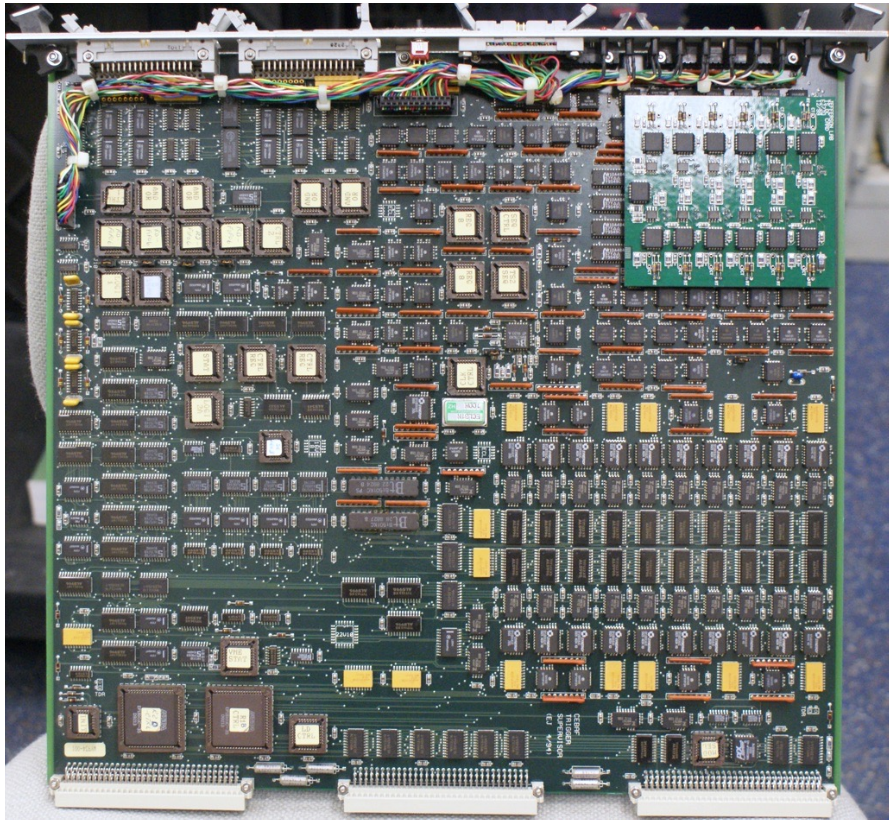
Outputs to trigger supervisor

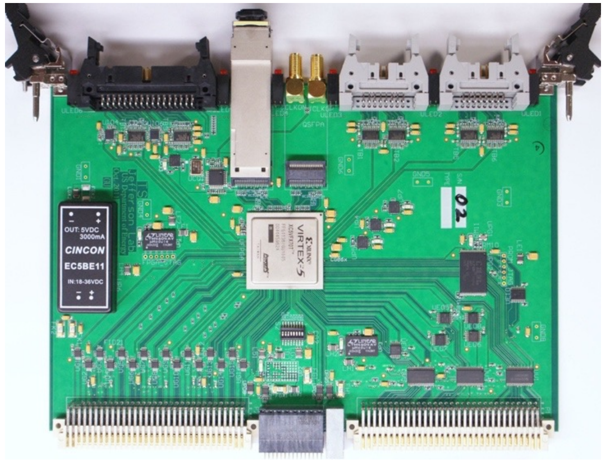




Complex electronics

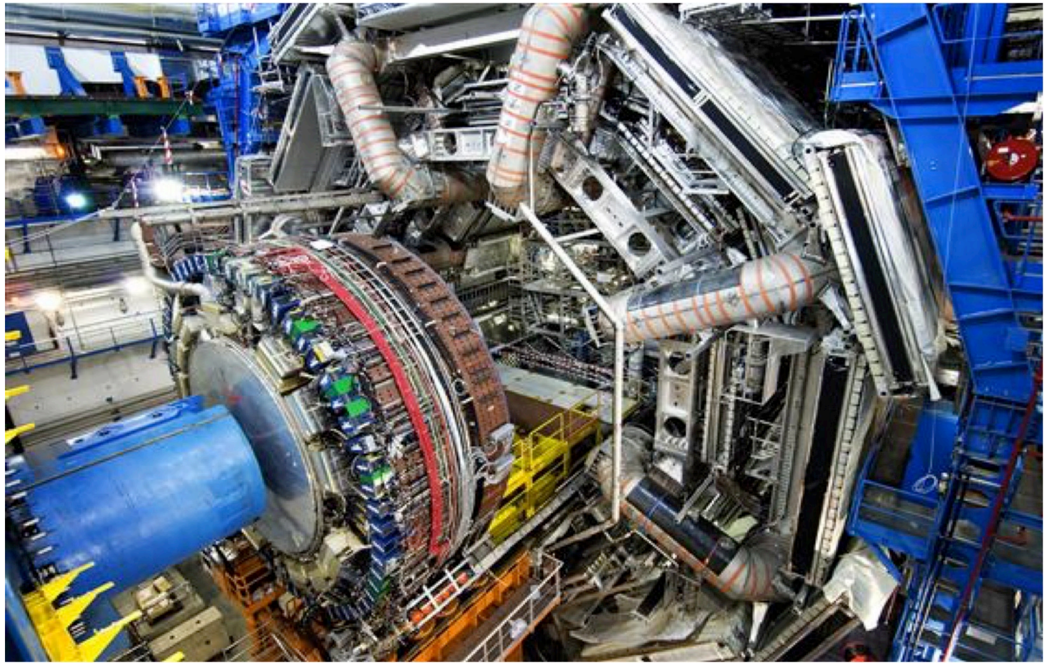
- The two pictures are of boards with similar functionality that were designed 20 years apart.
  - Upper – 1991
  - Lower - 2011
- The scale is the same for both photos.
- We can now buy programmable logic arrays that allow us to implement complex trigger algorithms in the firmware on a single chip.
- The other thing to notice is that the older board has connectors for cables on the front panel (top of pic).
- The lower board has some of the same connectors so that it can be used with older systems. It also has a fiberoptic transceiver for use with modern systems.





Part 3 - Data acquisition for big experiments

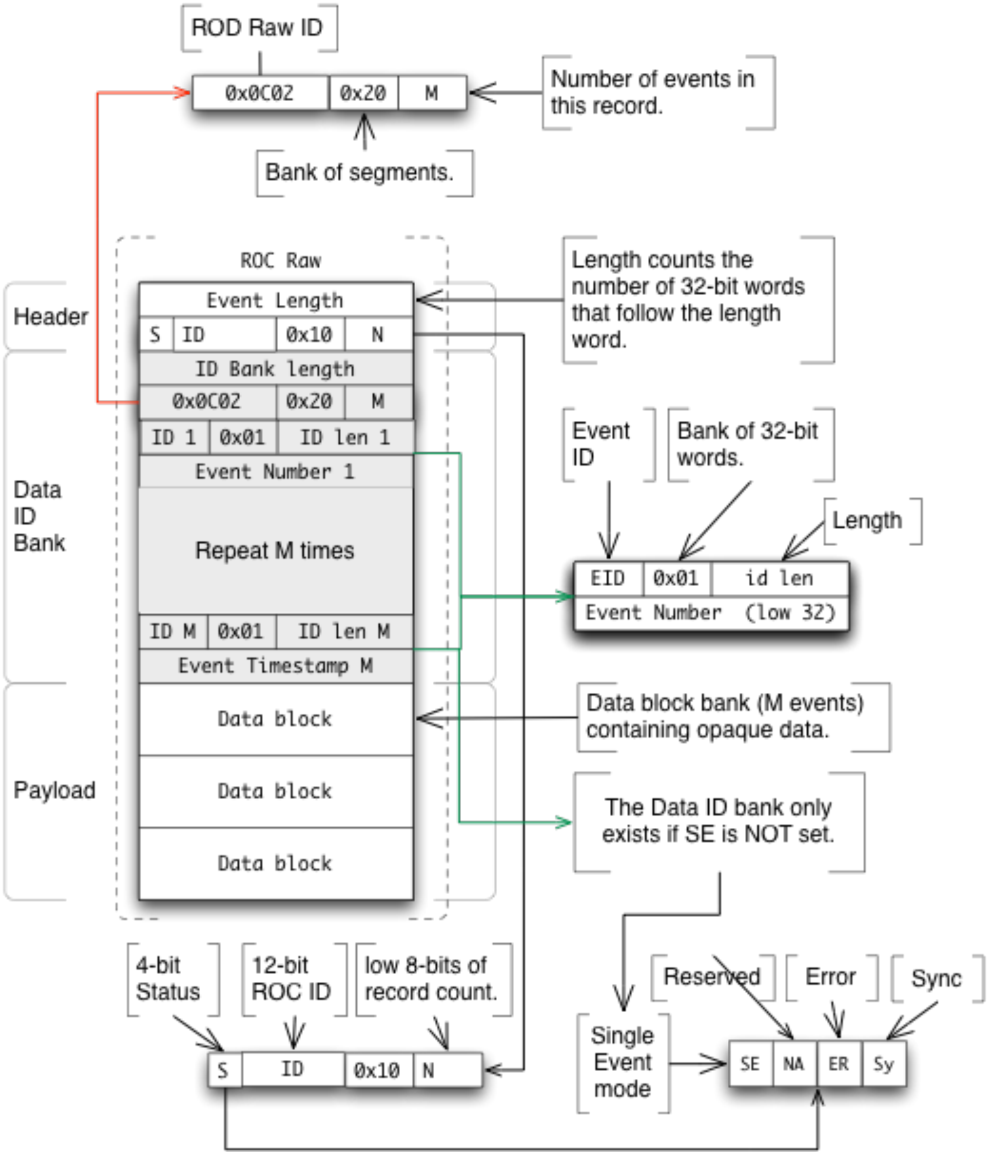
- Particle detecting hardware is physically distributed within the detector, so we need to:
  - Add metadata like :
    - where in the detector the data came from.
    - which trigger the data belongs to.
  - Transport the data to computer systems
  - Gather all the data from one event together.
  -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



ATLAS detector at the LHC is 7000 t and 46 m long. Much of the electronics is buried inside the detector.

EVIO Data format, a real example

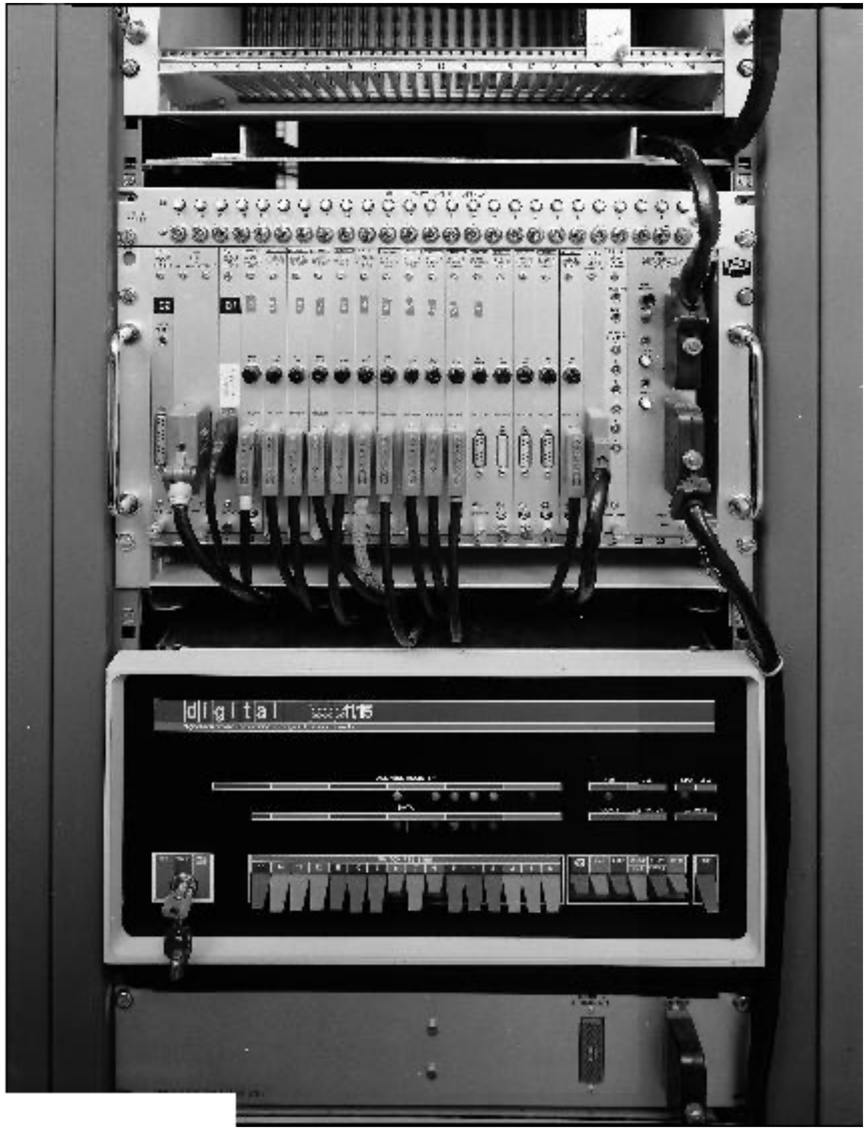
- EVIO is a self describing hierarchical format consisting of nested banks.
  - Bank - container for other data.
- Each bank starts with a header.
  - Length.
  - Description of content.
- The code 0x10 in the outer header tells us this bank contains other banks that are 32-bits wide.
- The first bank 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.



Data Transport methods:

YE OLDE 1980'S DAQ SYSTEM

- The DAQ components are physically spread out around the detector.
- Need to transport the data so that all the data for an event is available for processing.
- In the past DAQ systems were custom-built for each detector using specialized hardware specifically designed for NP or HEP experiments.
- Interconnect busses for electronics conformed to community wide standards not used outside of physics. Like:
  - CAMAC (Computer Automated Monitoring And Control)
  - FASTBUS – a Fast bus.
- The big drawback was lack of commercial options.
- Computers were expensive so a single Mainframe or Minicomputer system processed the data and connected to the DAQ using custom interfaces.
  - For example, CAMAC and Fastbus defined parallel inter-crate branch busses that ran over cables.



Network based DAQ (1980s to mid-90s)

- During the late 1980’s Ethernet network hardware was becoming cheap enough and fast enough that it could compete with the custom data transport methods for small experiments.
- Microprocessors replaced minicomputers and mainframes for readout.
  - Still used CAMAC but VME (VERSA bus Europa) was appearing.
- Television drove down the price of tape storage.
  - We stored data on VHS tapes! (but it was unreliable!)
- Pros
  - Widely available commercial hardware.
  - Standardization.
- Cons
  - Still slow for large experiments.

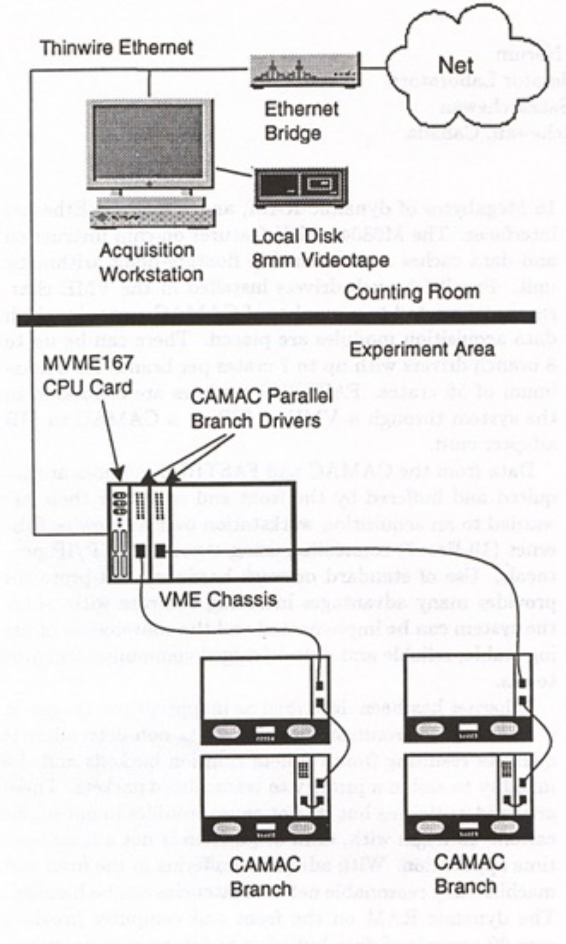
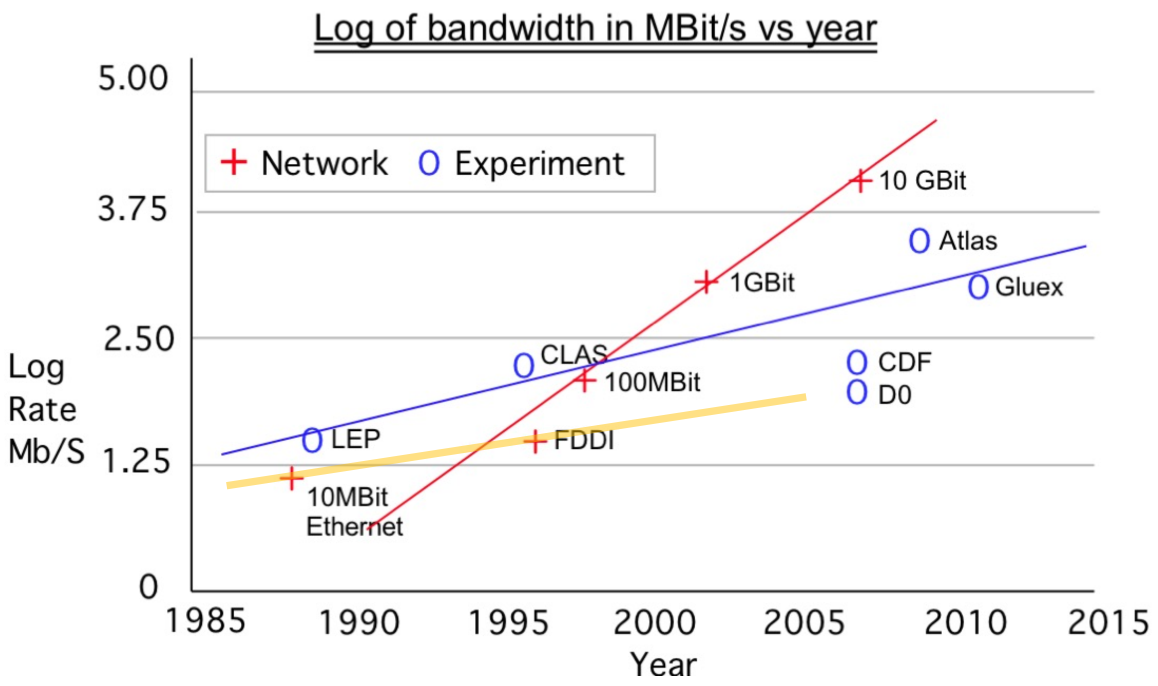


Figure 1: Hardware Configuration

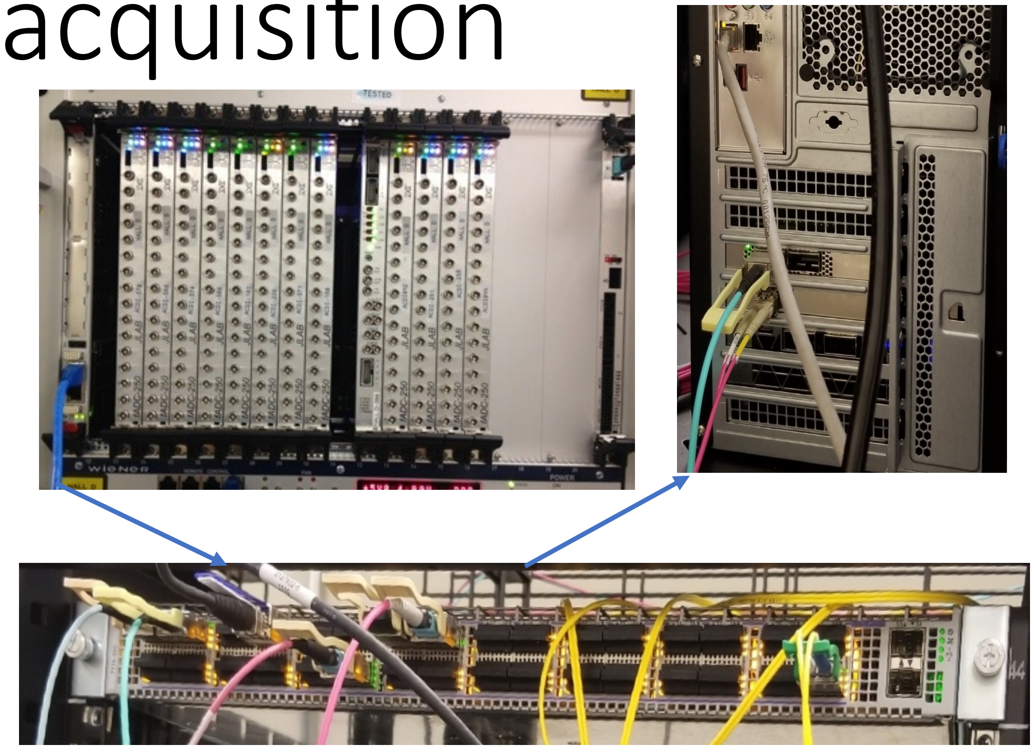
Network based DAQ – the internet era

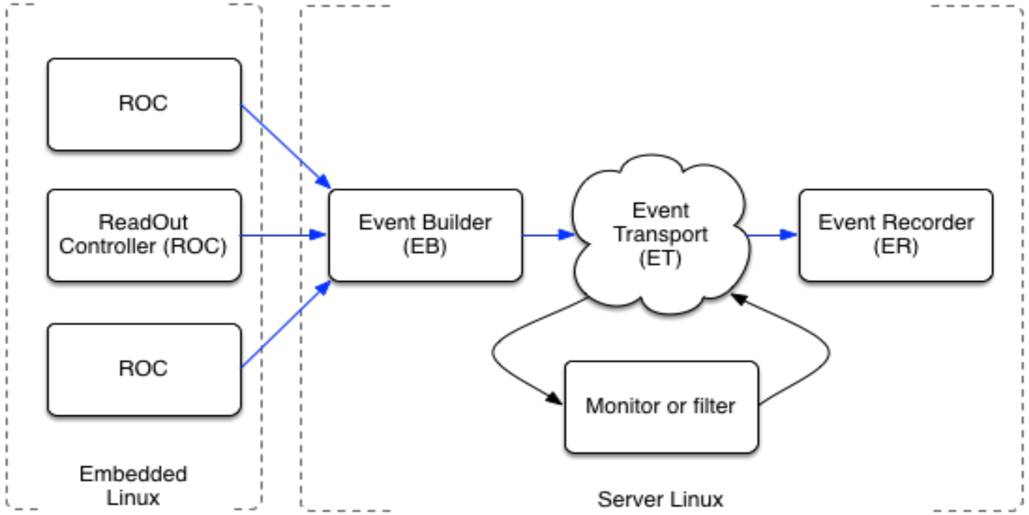
- Between the mid 80's and 90's the trends for growth in network and data acquisition bandwidth meant that network-based readout would not be practical for large experiments.
- The commercial internet has driven bandwidth up and prices down to the point where, for most experiments, designing custom data transport hardware is a thing of the past.
- All the data at JLab is now transported using commercial network hardware.



JLAB - Network based data acquisition

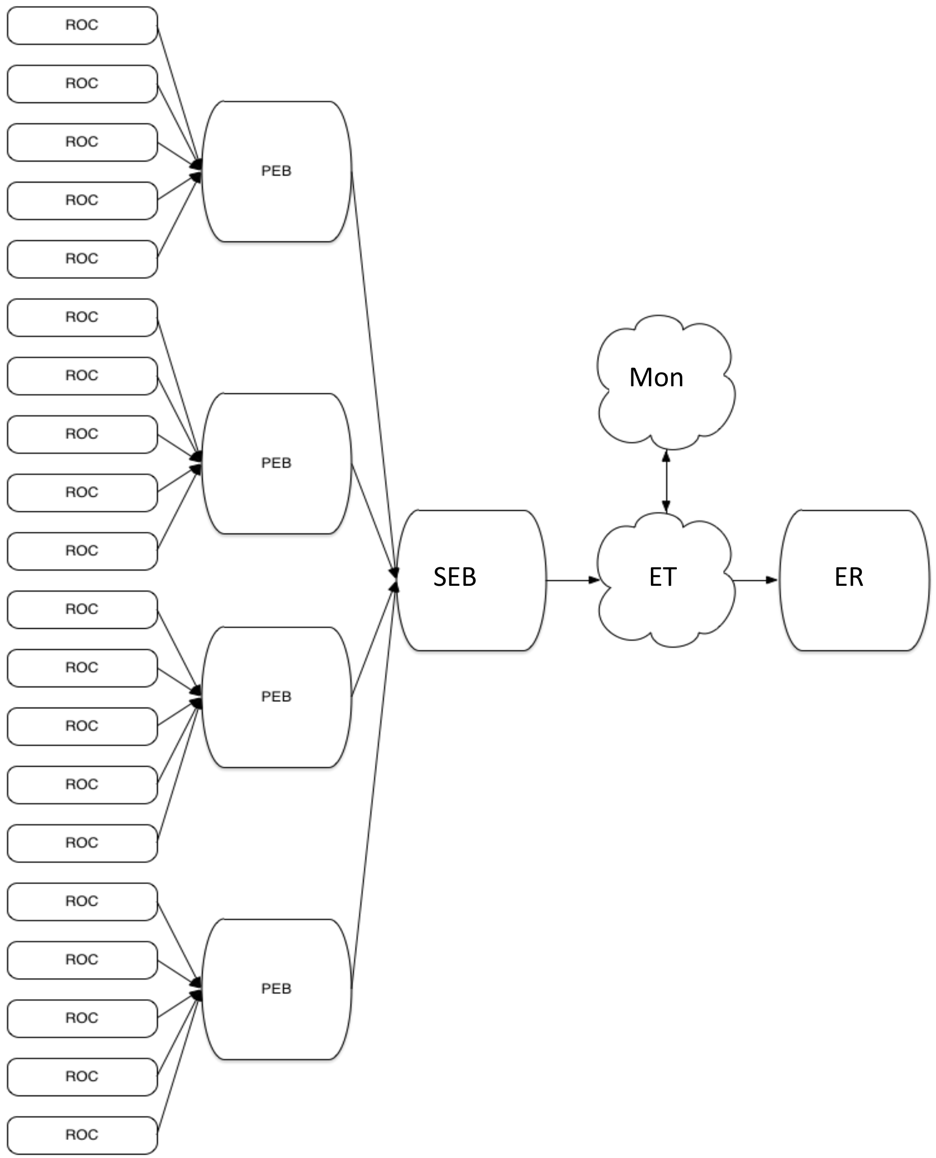
- VME CPU has ethernet built in, so readout controllers (ROCs) send data via a network.
  - Use a network switch to route data.
  - PCs and servers can match network bandwidth to DAQ rates.
- Event Builder software reads data off the network and assembles pieces with the same event number into an event.
  - This needs a piece from every sub-detector.
- We need to monitor data quality online.
  - This is done using Event Transport (ET) event transport software.
- We 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.





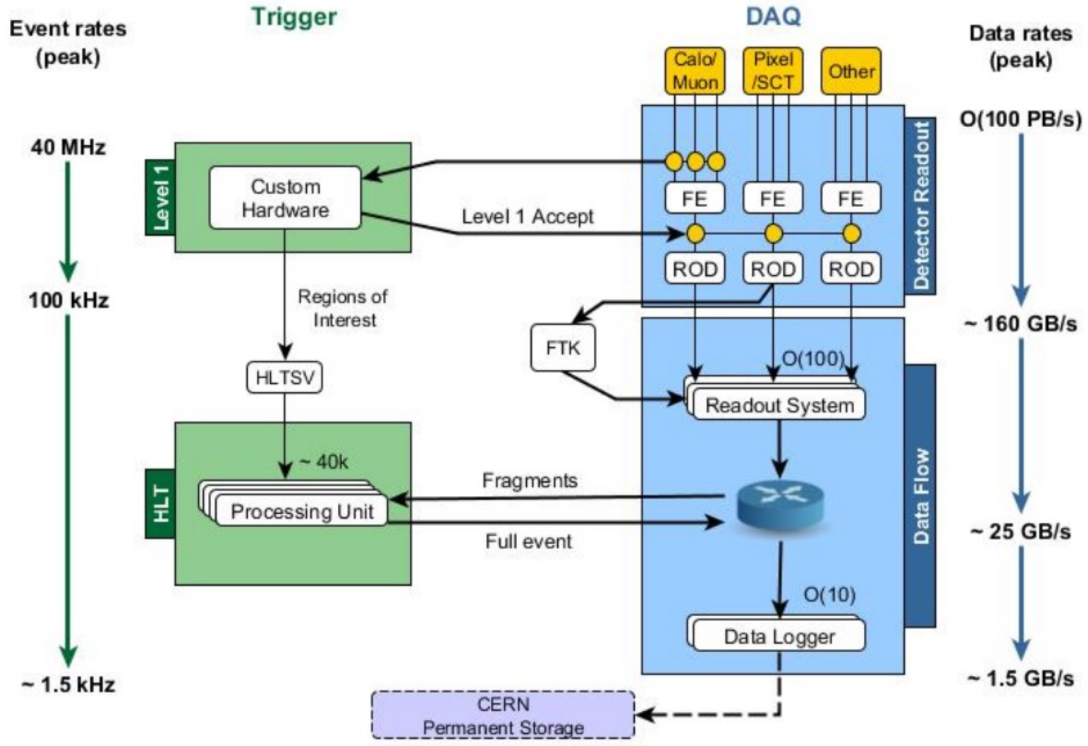
Event building

- In a large system the data and event rates can be too large for a single server to handle. Solution is to split the task into two levels.
- In this 20 ROC example:
  - Groups of five ROCS send data to one of four Primary Event Builders (PEBs).
  - This divides the formatting/checking workload, and the data throughput, for each PEB by four.
  - The four PEBs are send data to a single Secondary Event Builder (SEB). It must handle the full data rate but only has four incoming streams to merge.
- If this is too much, we can use two SEBs in parallel.
- Scalability via parallelism is an important technique.

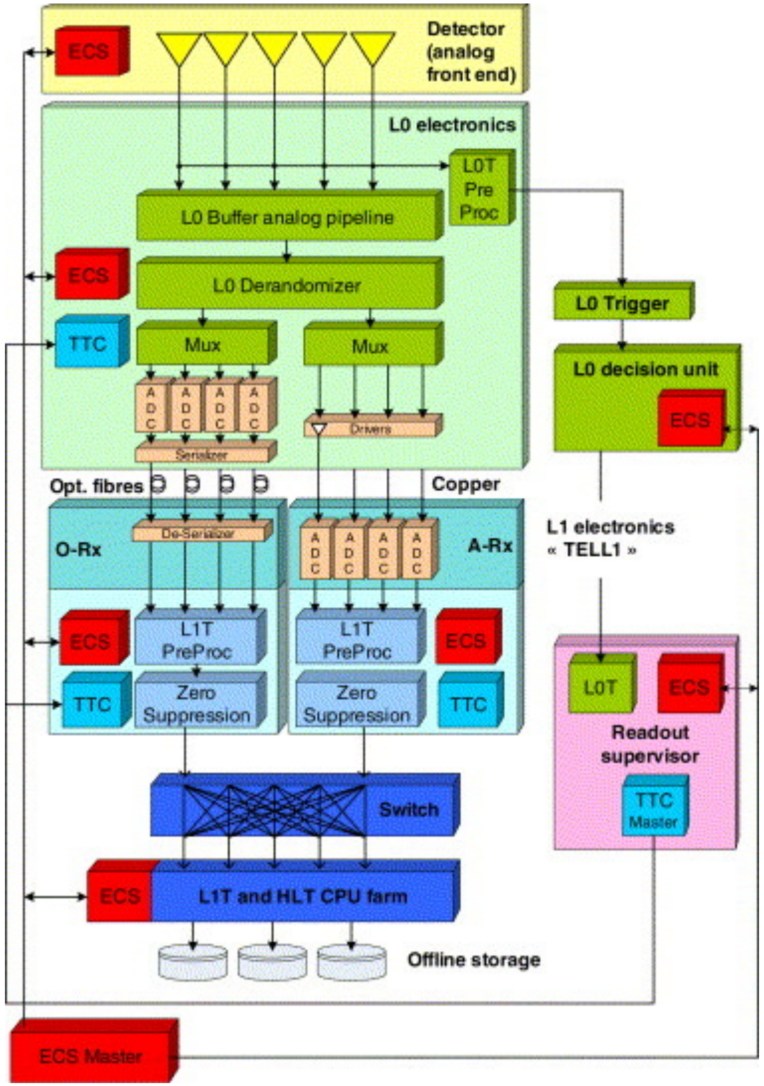


Non-JLAB examples

- Here are two examples from the LHC at CERN.
- In both cases the same steps outlined in the previous slides take place.



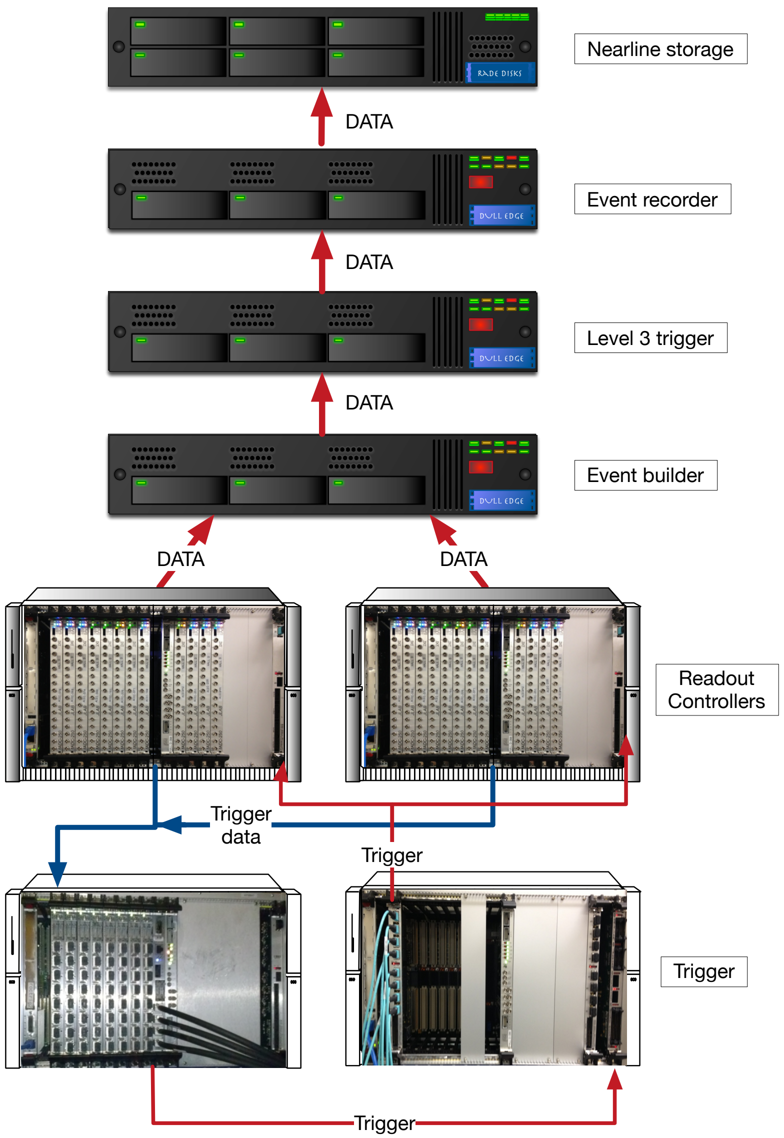
ATLAS DAQ



LHCb DAQ

Summary of SoA at JLab – pipeline mode

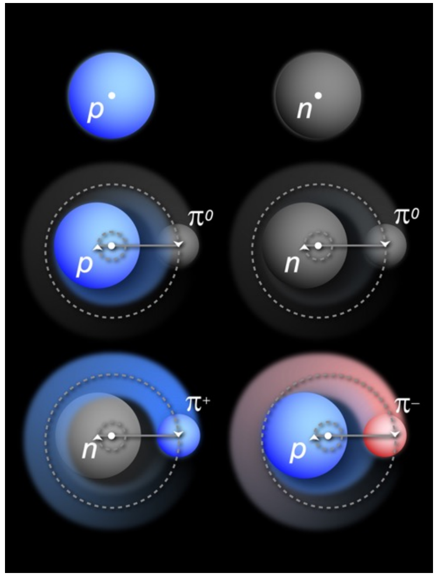
- Data is split into trigger and DAQ paths.
- Trigger data read by a Crate Trigger Processor and passed to a global trigger.
  - Trigger formed in custom electronics
- Blocks of data are queued in "pipeline" while trigger is made and read out by CPU over VME.
- Data sent over network to an Event Builder
- Pros:
  - Raw data is generated as a stream of events.
  - Trigger has already filtered most "unwanted data".
  - Well understood way of doing things – 30+ years of experience.
- Cons
  - Pipelines must be long enough to delay prompt data until slowest data appears.
  - Filtering is done before acquisition which relies on good understanding of trigger algorithms and any bias they may introduce.
  - All parts of DAQ must work otherwise events can't be built.
    - One failure stops the pipeline.
  - Doesn't work well when events overlap in time.
  - **Obvious bottlenecks!**



Isn’t this good enough though?

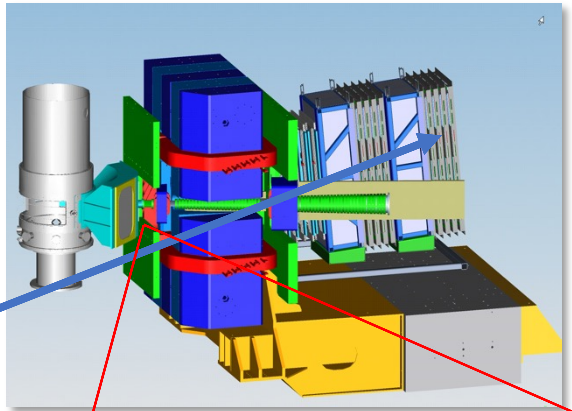
- No, not really.
- New experiments are being proposed. They present challenges because they have:
  - Detectors that do not play well together due to timing.
    - Traditional trigger and event builder strategies are not ideal when some of the data arrives long after the rest of the event.
  - Detectors 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)



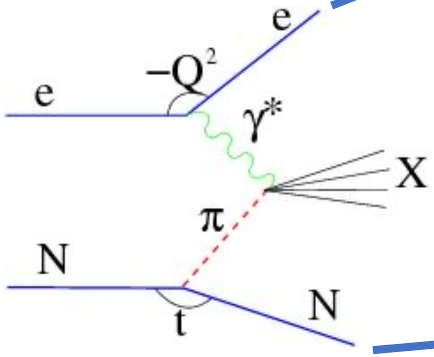
<u>TDIS Scientific Goal:</u> Access Elusive Partonic Structure of Mesons by Using Mesons in Nucleons as "Target"

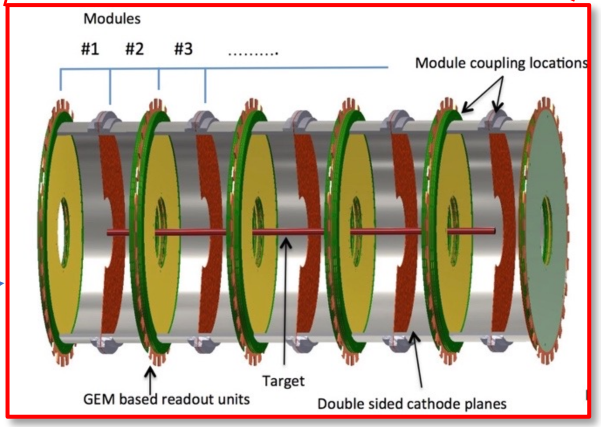
- In simpler words: As the soup of quarks and gluons in the nucleus jiggles about sometimes a nucleon can convert into a nucleon plus a π particle. The trick of the experiment is to use the electron to kick the pion out giving a way to study this strange process.
- OK, that wasn't much simpler but that's physics for you…



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

The electron arrives in nanoseconds after the interaction.





Protons are detected by radial Time Projection Chamber (rTPC).

This uses drift of ions in a chamber to trace out the path of a proton onto the inside wall which is covered in sensors.

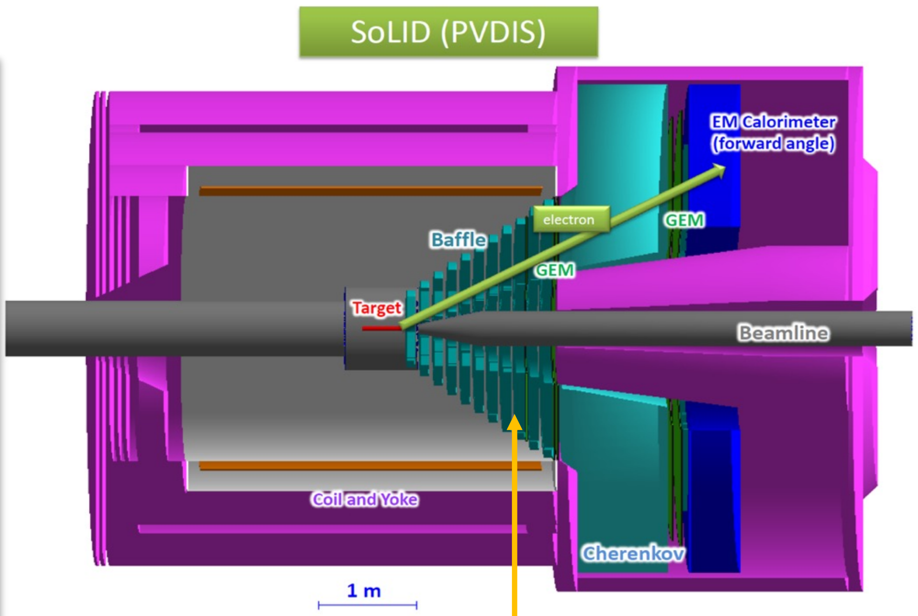
The ions drift slowly so it can take microseconds.

TDIS - Challenges

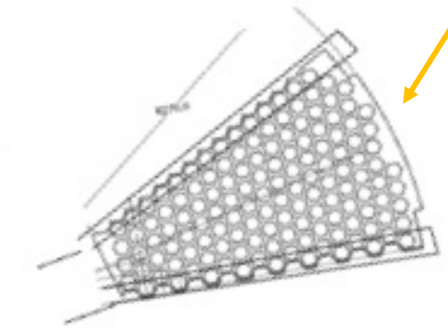
- It takes much longer for the proton track to trace itself out on the wall of the TPC than for the Super Bigbite detector to pick up the electron. There is a large timing mismatch.
- It's an electron beam so a large background of scattered electrons not associated with a proton.
- Even with a small rTPC there are multiple proton tracks 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 off a fixed target at high luminosity.
- Spiral baffles cut background.
- Electron detected by a GEM detector split radially into 30 sectors.
  - Each sector data rate of up to 1 Gbyte/s, 30 Gbyte/s total.
- Challenges :
  - How to handle 30 Gbyte/s affordably?
  - How to handle hits at sector edges span two sectors?
  - How to integrate non-GEM detectors?



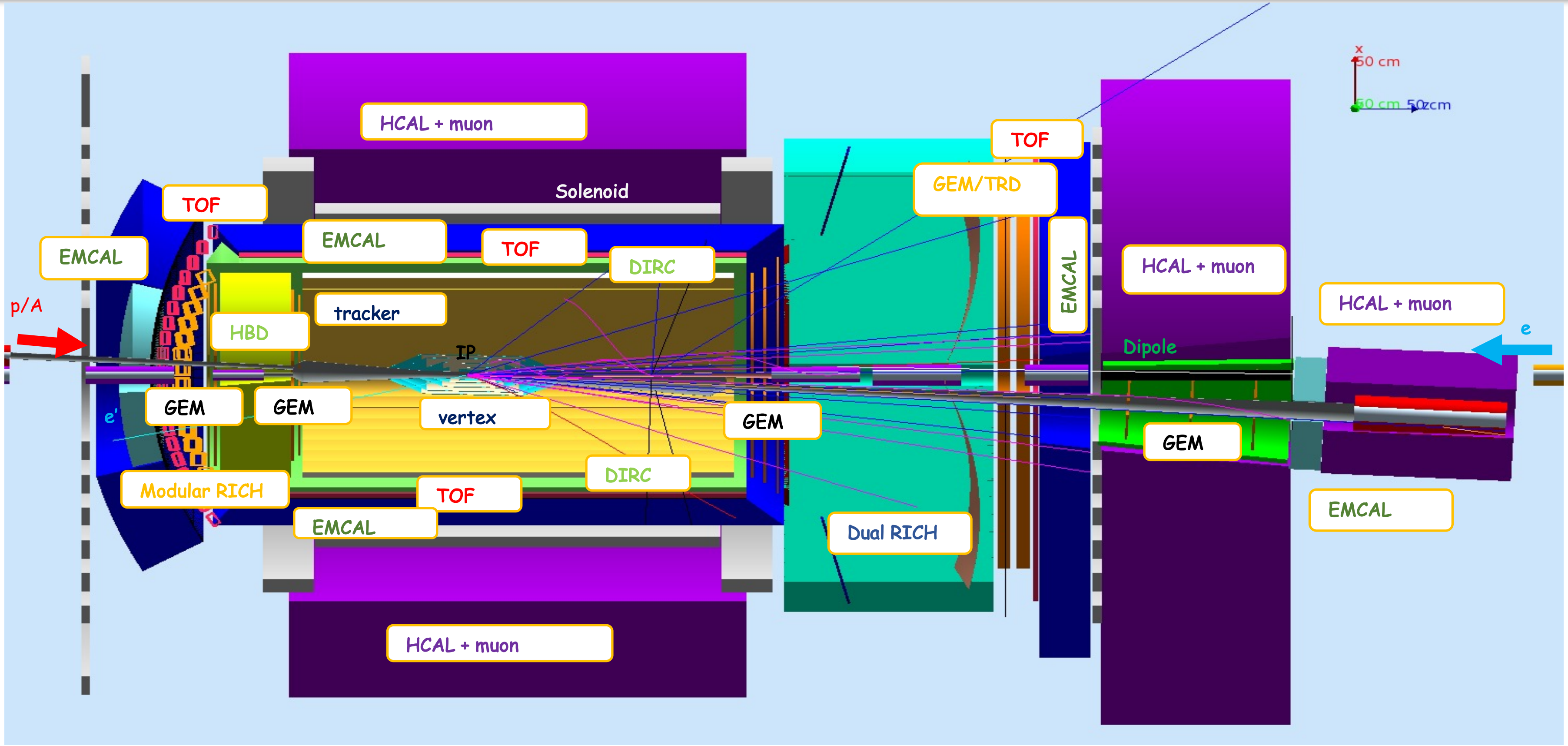
Baffles constrain momentum



GEM detectors are segmented into 30 sectors.

Electron Ion Collider, example detector design

- Collider experiments colliding electrons with ions. Momentum difference between the beams leads to asymmetric 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.



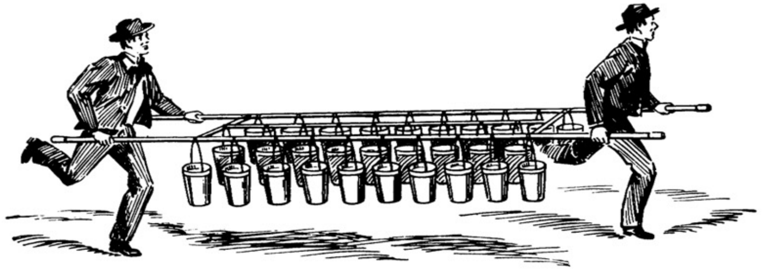
EIC DAQ guesses

- The Vertex detector could be 20-50 M channels.
  - Assume average 1% occupancy rate ~240 GB/s. (yes bytes)
- Ignore the elephant, the rest of the detector is ~1M channels.
  - Assume we use the same technology we currently use.
  - EIC would need ~1,000 crates for non-vertex DAQ.
  - Assume average 1% occupancy rate ~5 GB/s.
- How to read out a detector that generates 240 GB/s ?
  - Dealing with the Vertex Tracker dominates the design of the DAQ.

Part 4 - Streaming mode

- In 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.
- Pipeline readout increases bandwidth by moving blocks of events.
- Why is it done this way? For most of the history of data acquisition three things were assumed:

1. It is impossible to acquire all the data from every channel in a detector without a trigger.
2. Even if it could be acquired it would be impossible to store.
3. Even if it could be stored it would be an impractically large dataset to ever be able to process.

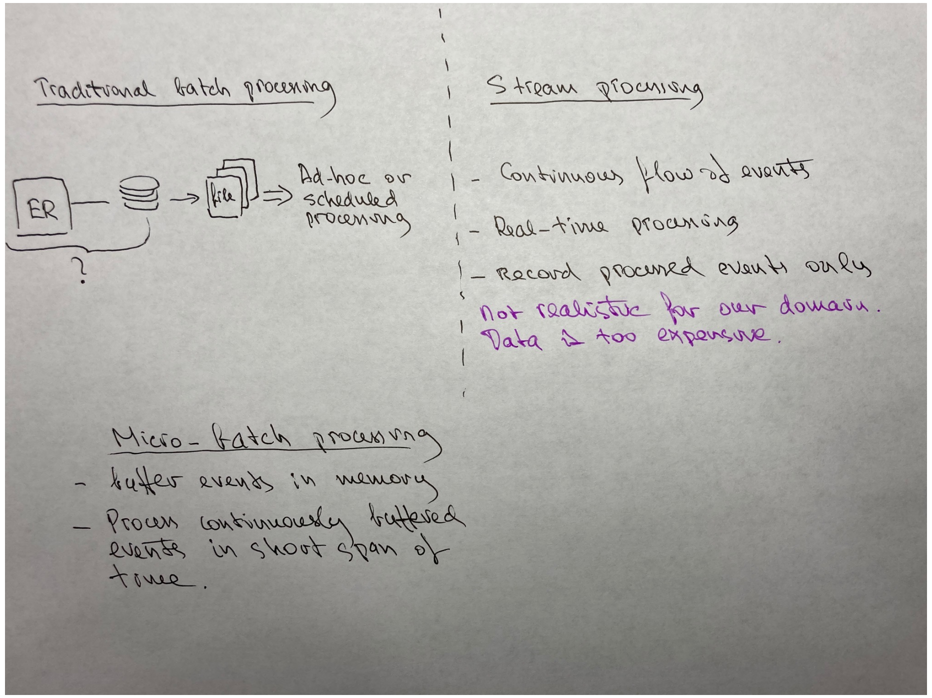


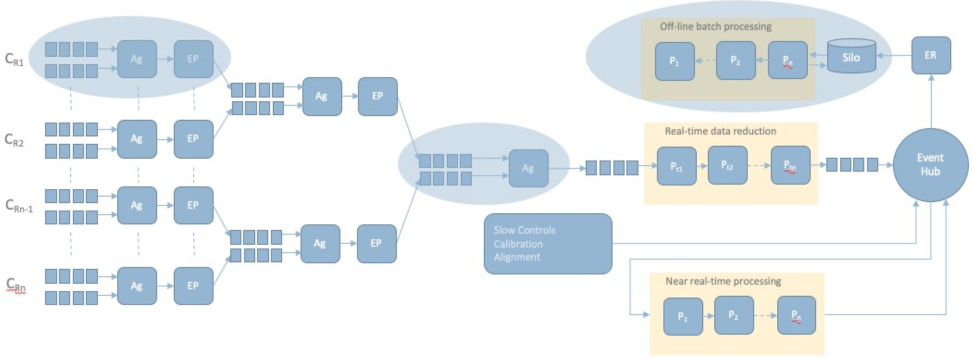
Streaming mode

- Advances in technology over the last ten years mean that none of those assumptions are true.
- In a Streaming readout:
  - Data is read continuously from all channels.
  - Validation checks at source reject noise and suppress empty channels.
  - Rather than a trigger or event number the data is organized by time.
  - The data then flows unimpeded in parallel streams to storage or a local compute resource.
  - Data flow is controlled at source.
  - Data is processed as it is taken to reduce data volume.

Streaming advantages

- The lack of a trigger means that:
  - Potential unintended bias due to the trigger is eliminated.
  - We could run groups of experiments in parallel.
    - One person's trash is another person's treasure.
  - The system is simplified – no expensive custom electronics.
  - Readout speed is independent of detector response time.
- Requires robust and accurate time stamp generation and distribution.
  - Is still a simpler task than an online trigger.
- Parallel timestamped streams mean:
  - System is robust against minor hardware or firmware glitches.
  - Can use novel analysis techniques such as AI/ML looking for patterns rather than numerically processing events.



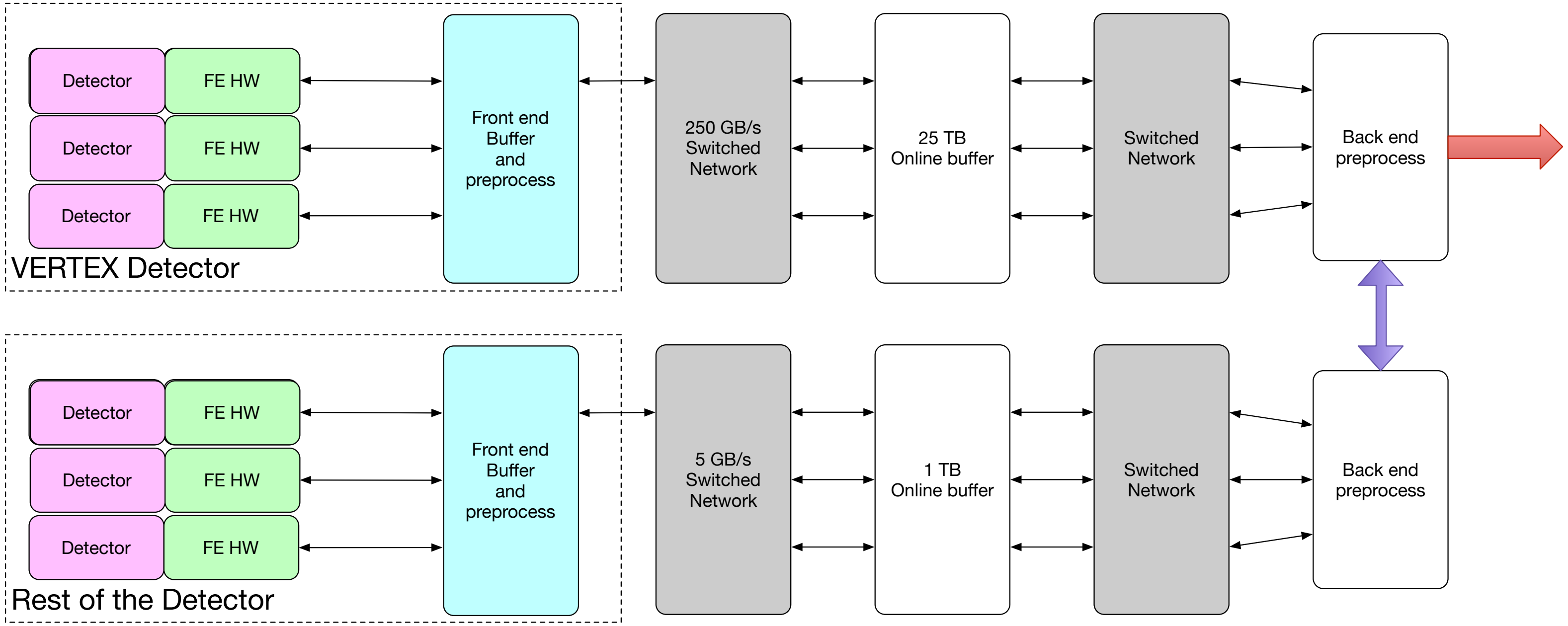


What does streaming readout solve?

- TDIS
  - High data rate from TPC handled as parallel streams – no single stream handles the full rate.
  - Electron data is its own stream and is matched up with proton candidates offline – removes high - rate event building issue – store it and work it out later.
- SoLID
  - High rates from segmented detectors are handled in parallel – no real time event building.
  - Sector edge effects and correlation with other detectors is handled in software offline.
- EIC
  - We handle the vertex detector by reading sections out in parallel and holding the data temporarily in memory while data from the rest of the detector is used to define regions of interest.

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

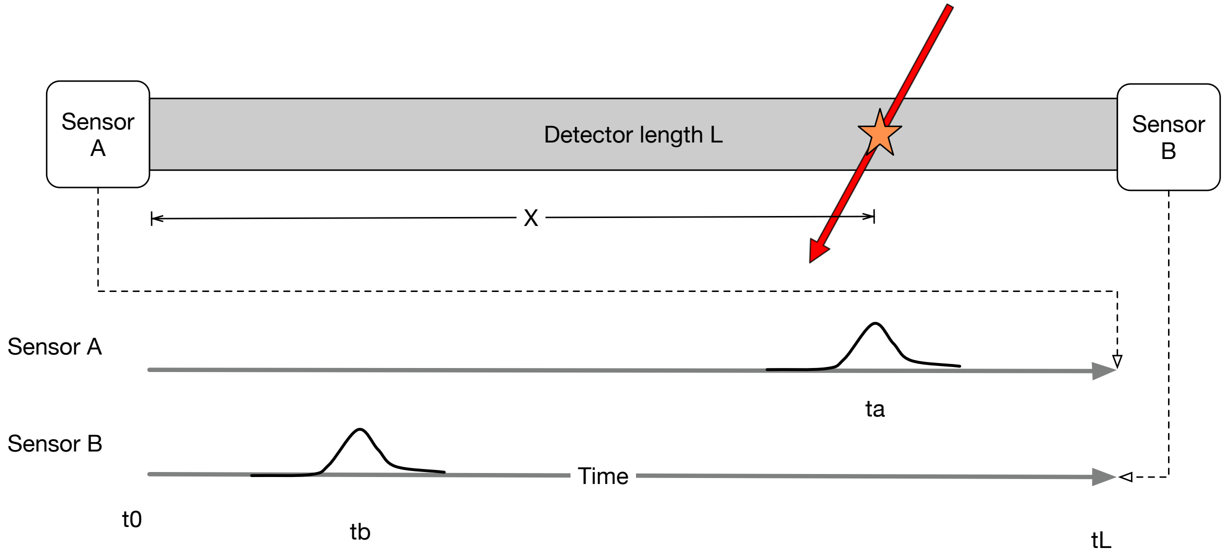


Part 5 - Calibration

- So, we’ve acquired the data, either via triggering and event building or by streaming the data to temporary storage and working it all out in software. We now have events.
  - We’re done and we can all go home, right?
- Not so fast!
  - What we have, as events, are a sequence of blocks of numbers that represent amplitudes and times in whatever units the digitizing electronics spits out.
  - To do physics analysis we need particle identities, the energy and momentum of those particles.
    - i.e., physical quantities in the correct units.
  - This is achieved in a data processing stage known as “reconstruction”.
    - Literally reconstructing the physics from the raw numbers.
  - The raw data is converted into physical values such as position, energy, and time, using a detector calibration. So where do we get one of those from?

Example of measuring position

- In this example we have a simple linear detector in the form of a bar L units long.
- A particle crosses the bar X units from the left end and deposits some energy at a time t_0, which is detected by sensors at each end.
- Sensor A detects a pulse at a time t_a. Sensor B detects one at a time t_b.
- We take two measurements because we don't know what t_0 is. If we subtract t_b from t_a then t_0 is eliminated and we have an equation that can be solved for X.
- Once we know X we can calculate t_0.
- So, we measured three things at once:
  - The energy deposited.
  - The position that the particle "hit" the detector.
  - The time when the particle "hit".
- Many complex detectors are arrangements of linear components in sheets or cylinders.



$X = ½ ((t_a - t_b)/V_p + L)$

$t_0 = t_a – X/V_p$

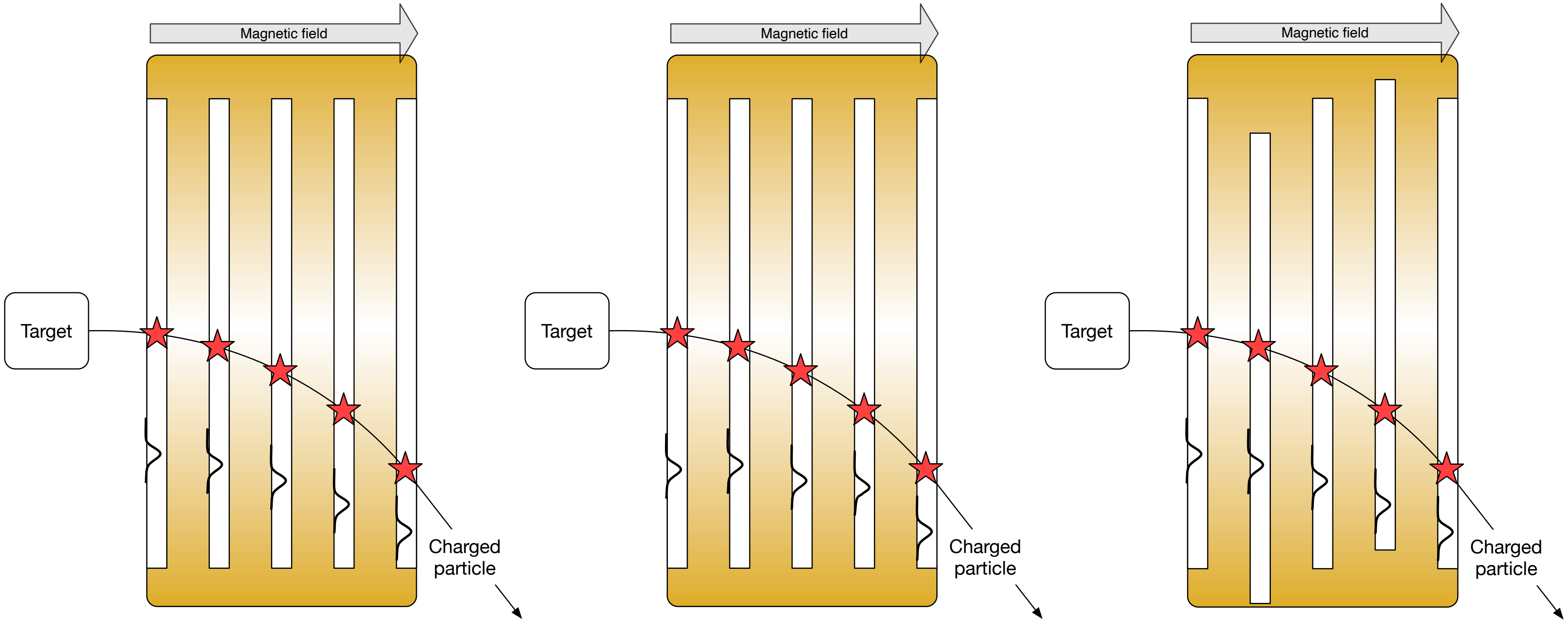
Where V_p is the pulse velocity in the detector.

The two extreme cases are when the particle is close to either sensor.

Neither t_a or t_b can be larger than L/V_p so the pulse velocity can be determined from the cutoff of a histogram of t_a or t_b.

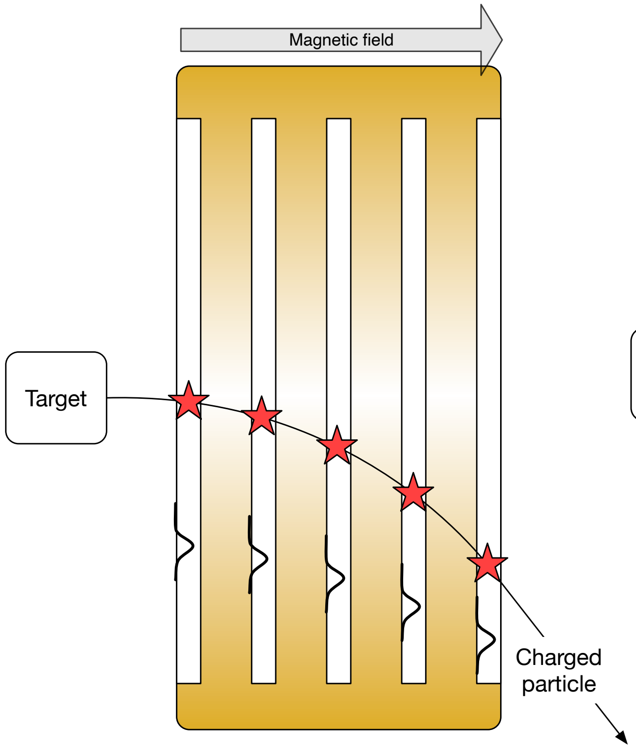
Example calibration

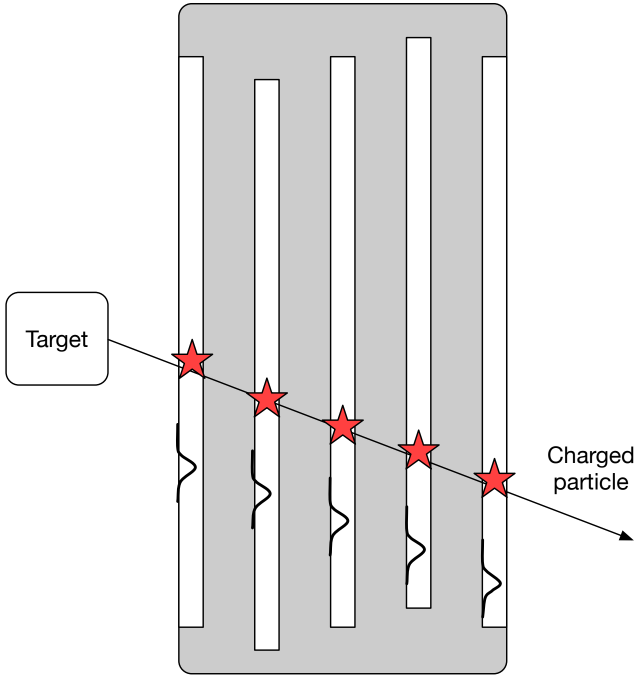
- On the left a charged particle from an interaction curves through a magnetic field and is detected in five detector planes, 2D arrays of the sort 1D detectors from the previous slide.
- In the center the pulse velocity is subtly different for each channel of the detector.
  - So, we must measure the velocity for each of the channels.
- On the right not only are the pulse velocities different but also the channels are shifted relative to each other. How to sort that mess out?



Example calibration

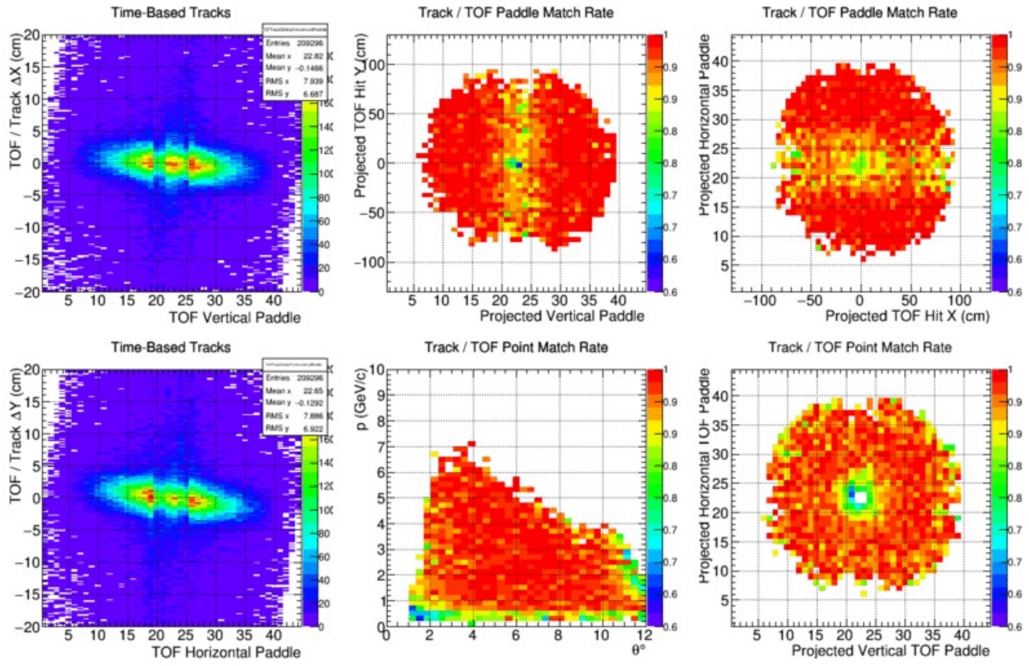
- It turns out to be a lot easier to turn off the magnetic field, so the tracks are straight.
- Take a few million events with the field off.
  - Calculate the pulse velocities for each channel.
  - Calculate the position of each hit.
  - The hits must fall on a line so the offsets can be calculated.

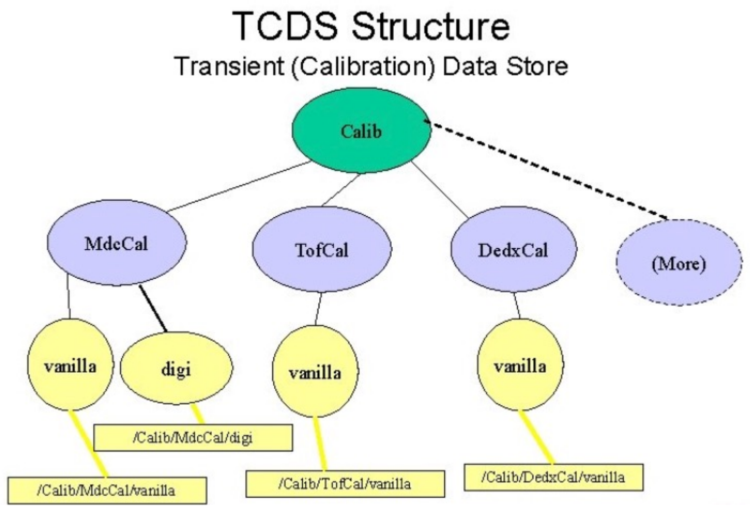




Calibration is complicated

- These were very simple examples. There is much more to calibration than there is time for here.
  - We run simulations to produce simulated data then compare that data with experiment.
    - Does the detector generate data that looks right?
  - We look at interactions that are not necessarily the ones the experiment is interested in but that are very well studied.
    - Make sure our detector measures the correct masses for various particles.
- Usually, the output of the calibration process is a database of all the parameters needed to convert measured values into physics variable.
  - Calibration changes with time…
- One thing about calibration in the context of streaming readout is that, since there is no trigger, there is no need for special data taking under “calibration trigger conditions”. We use a sample of the data to perform as much of the calibration online as is possible.





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.
    - This is not necessarily what was available when it was designed.
  - There is always some R&D time to discover new algorithms and techniques.
- The key principles DAQ systems rely on are:
  - Modularity – break large problems into small ones
  - Standardization – data formats and transport protocols allow future tech to be incorporated.
  - Parallelism – streaming readout is the ultimate in parallelism, do as much as possible in parallel.