Summary from ISOTDAQ 2023

(Part I: DAQ & Trigger)

School sponsored by:

June 13-22, 2023
https://indico.cern.ch/event/1182415/


ISOTDAQ
International School of Trigger and Data Acquisition

CERTIFICATE
OF PARTICIPATION

This is to certify that
Angira Rastogi
has participated to the 13th International School of Trigger and Data Acquisition (ISOTDAQ) that took place at the Department of Physics, Istanbul University, Istanbul, Turkey from Tuesday 13 to Thursday 22 June 2023.

Aytül Adıgüzel
for the Organising Committee

Angira Rastogi, LBL

LBL Weekly Instrumentation Meeting BERKELEY LAB **June 30th, 2023**

About ISOTDAQ 2023

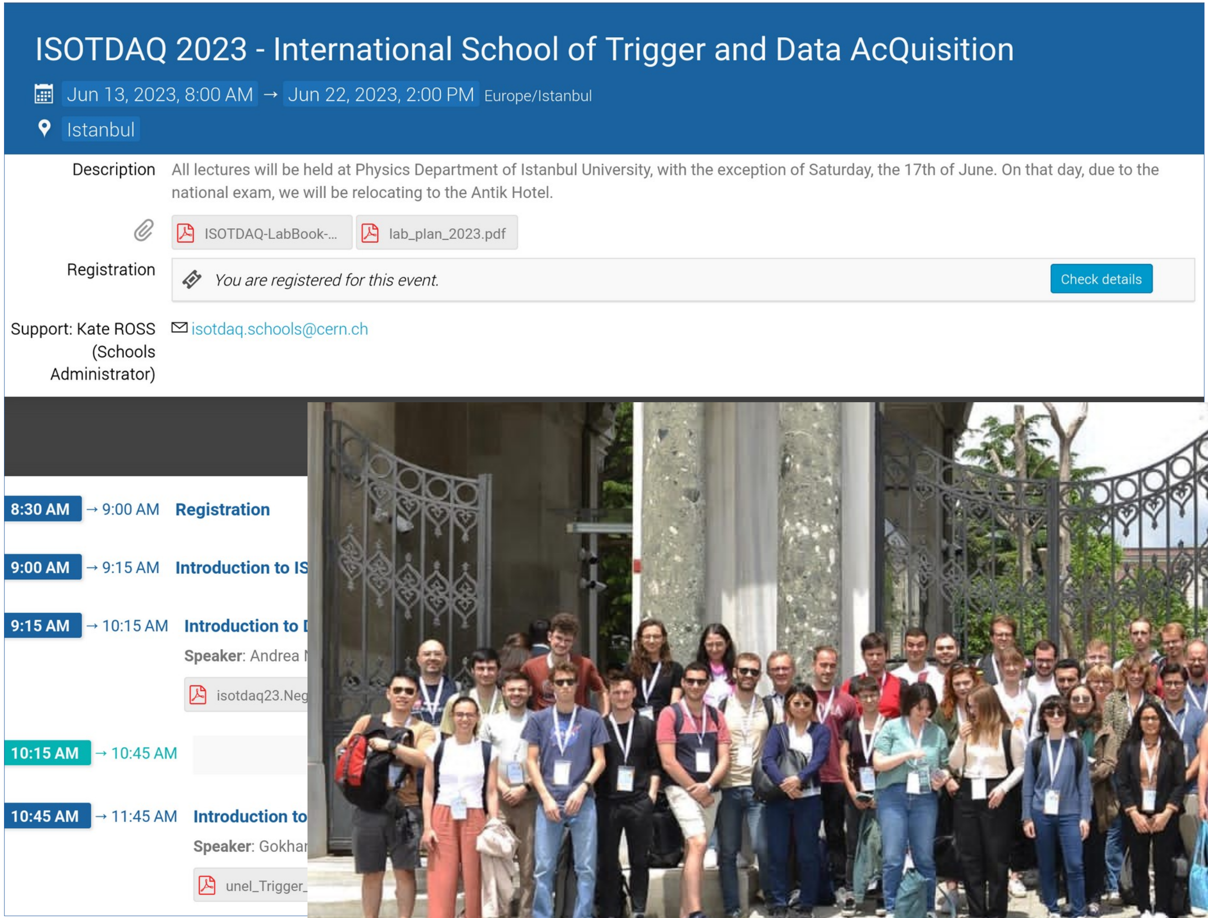
- **13th** of a series of International Schools dedicated to introduce "arts and crafts" of **triggering** and **acquiring data** for physics experiments.
- To provide an overview of the basic **instruments** and **methodologies** used in high energy physics, spanning from **small experiments** in the lab to the very **large LHC experiments.**
- Emphasizing the main **building blocks** as well as the different **choices** and **architectures** at different levels of complexity.
- Split 50-50 between a series of **lectures** and **hands-on** laboratory exercises.

ISOTDAQ
International School of Trigger and Data Acquisition

ISOTDAQ 2023 – List of topics covered

- ***DAQ (6 lectures + 2 labs)***
- ***Trigger (4 lectures + 2 labs)***
- ***Detector readout and controls (6 lectures + 2 labs)***
- ***FPGA (2 lectures + 3 labs)***
- ***Data transfer/storage (4 lectures + 3 labs)***
- ***Programming, Machine learning (2 lectures)***
- ***Microcontrollers (1 lecture + 1 lab)***
- ***LabView (1 lecture + 1 lab)***
- ***TDAQ in Space, Medical imaging applications (1 lecture each)***

ISOTDAQ 2023 – Highlights



- *10 days, ~70 participants*

ISOTDAQ 2023 – Highlights

ISOTDAQ 2023 - International School of Trigger and Data AcQuisition

Jun 13, 2023, 8:00 AM → Jun 22, 2023, 2:00 PM Europe/Istanbul

Istanbul

Description: All lectures will be held at Physics Department of Istanbul University, with the exception of Saturday, the 17th of June. On that day, due to the national exam, we will be relocating to the Antik Hotel.

ISOTDAQ-LabBook-... lab_plan_2023.pdf

Registration: *You are registered for this event.* Check details

Support: Kate ROSS (Schools Administrator) isotdaq.schools@cern.ch

Tuesday, June 13

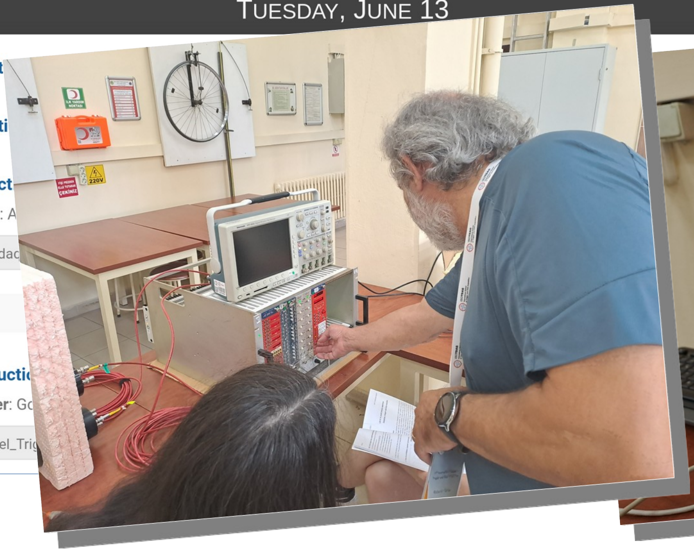
8:30 AM → 9:00 AM Registrat

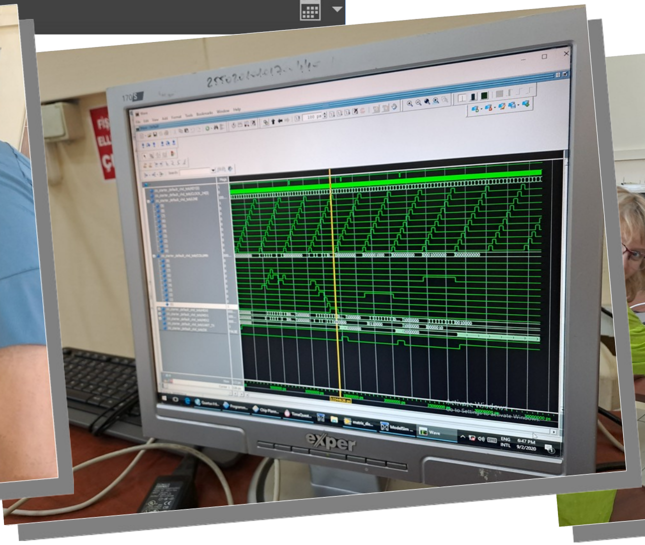
9:00 AM → 9:15 AM Introducti

9:15 AM → 10:15 AM Introduct
Speaker: A
isotdac

10:15 AM → 10:45 AM

10:45 AM → 11:45 AM Introducti
Speaker: Gc
unel_Trig





- *10 days, ~70 participants*
- *30 lectures (~30 hours)*
- *14 labs (~28 hours)*

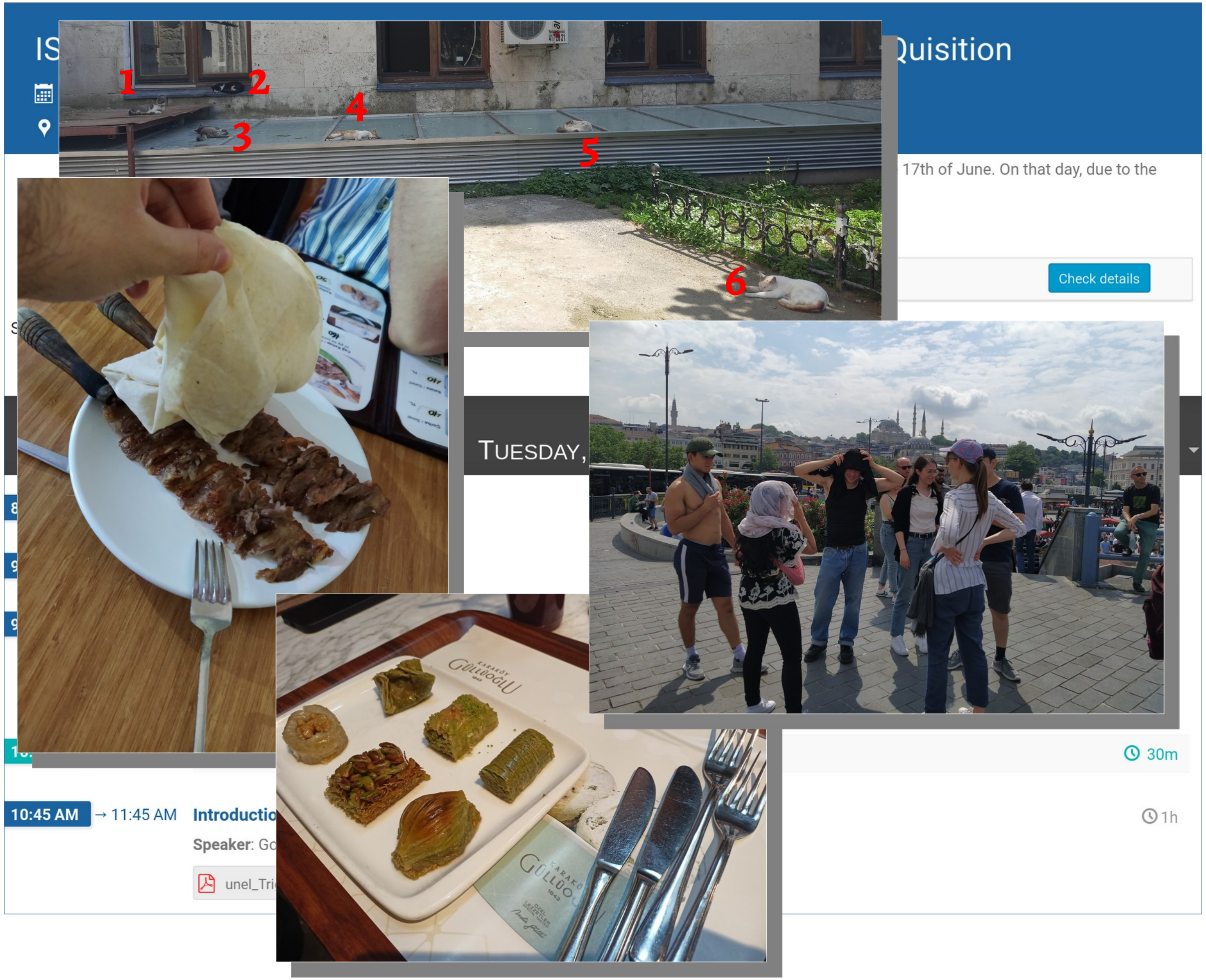
ISOTDAQ 2023 – Highlights

- *10 days, ~70 participants*

- *30 lectures (~30 hours)*
- *14 labs (~28 hours)*

- *Welcome drinks at Cistern!*
- *Social dinner*

ISOTDAQ 2023 – Highlights



- *10 days, ~70 participants*
- *30 lectures (~30 hours)*
- *14 labs (~28 hours)*
- *Welcome drinks at Cistern!*
- *Social dinner*
- *Lots of cats, kebaps, sweets and heat…*

ISOTDAQ 2023 – List of topics covered

- ***DAQ (6 lectures + 2 labs)***
- ***Trigger (4 lectures + 2 labs)***
- ***Detector readout and controls (6 lectures + 2 labs)***
- ***FPGA (2 lectures + 3 labs)***
- ***Data transfer/storage (4 lectures + 3 labs)***
- ***Programming, Machine learning (2 lectures)***
- ***Microcontrollers (1 lecture + 1 lab)***
- ***LabView (1 lecture + 1 lab)***
- ***TDAQ in Space, Medical imaging applications (1 lecture each)***

In this talk (roughly)

Executive Summary

- ***Introduction to DAQ***
  - *DAQ hardware*
  - *DAQ software*
  - *Networking for DAQ systems + Lab exercise*
  - *Timing for DAQ*
- ***Introduction to Trigger***
  - *Trigger hardware*
  - *Intelligent triggering with pattern recognition*
- ***GPU in trigger for HEP***
  - *Lab exercise*
  - *Towards Run3*
- ***TDAQ design: from test beam to medium size experiment***
  - *Detector readout, modular electronics*
  - *Lab exercise*
- ***Detector Control Systems***
  - *Control, configuration & monitoring*
  - *Lab exercise*
- ***Machine learning for TDAQ***

In this talk (roughly)

Introduction to DAQ

What is DAQ?

- Data AcQuisition (DAQ) is

  – the process of **sampling signals**

  – that **measure** real world physical conditions

  – and **converting** the resulting samples **into digital** numeric values that can be manipulated by a PC.

- Components:

  DAQ hardware

  – **Sensors:** convert physical quantities to electrical signals

  – **Analog-to-digital converters:** convert conditioned sensor signals to digital values

  Networking

  – **Processing** and **storage** element

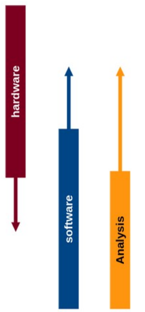
  DAQ software

Involves:

- Physics
- Electronics
- Computer science
- Networking
- Hacking
- Experience

End goal: always Physics…

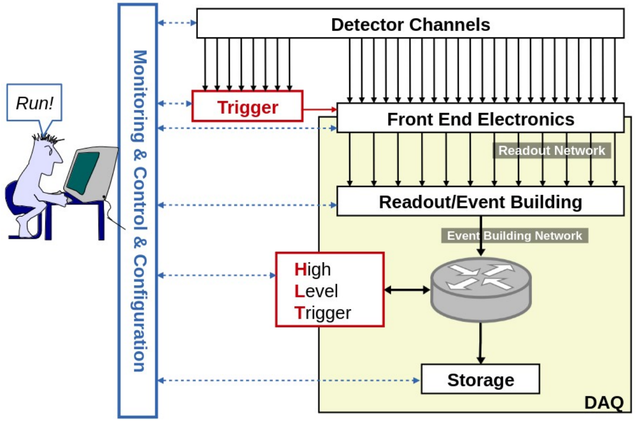
- **Experiment/detector –** produces physics (electrical signals)
- **Data AcQuisition –** Extracts physics from detector (digitized data)
- **Analysis (offline) –** Extracts physics from data (searches, measurements)



DAQ responsibilities

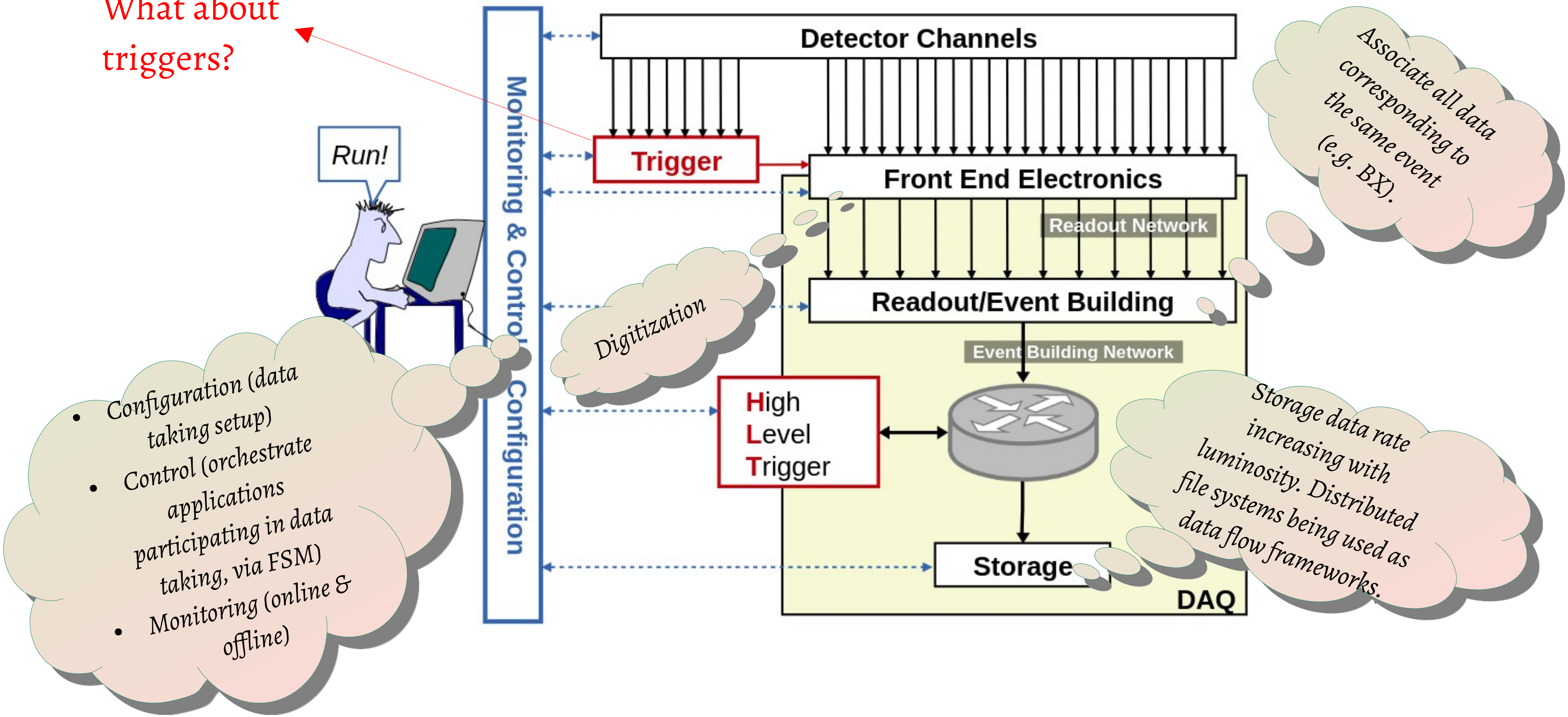
- Gather data produced by detectors
  - **Readout, digital**
- Form complete events
  - **Data Collection** and **Event Building**
- Possibly feed other trigger levels
  - **High Level Trigger**
- Store event data
  - **Data Logging**
- Manage the operations
  - Run **Control, Configuration, Monitoring**

Data flow



The DAQ umbrella

What about triggers?



Triggers

Should have the following features:

– **High efficiency:** selective for “signal” and resistant to “background”.

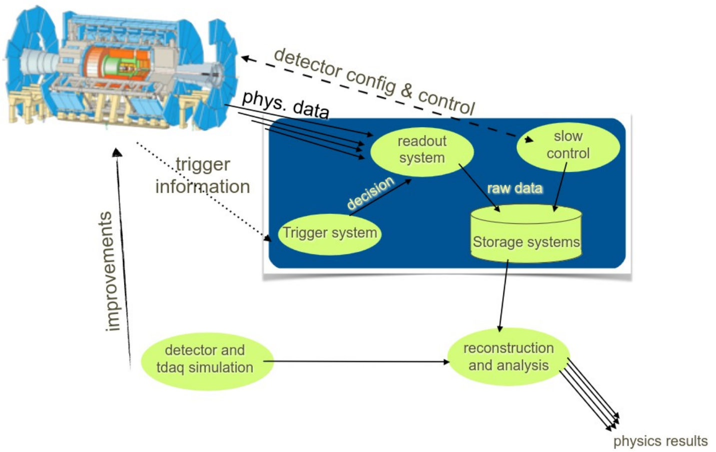
– **Low latency:** quick reaction & arriving at a decision for the event in question.

– **Low deadtime:** quick execution of tasks following to the occurrence, to distribute the signal to front end electronics.

– **Flexible:** Allow for quick modification, if necessary, while running.

– **Good synchronization:** to coordinate data arriving from multiple subdetetctors.

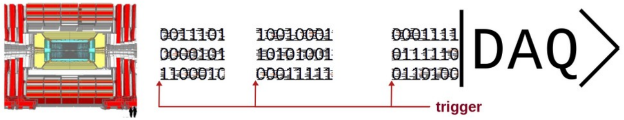
– **Affordable.**

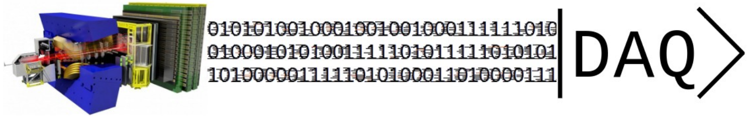


$$Trigger + DAQ = TDAQ$$

- **Triggered readout**: Data is readout only when a trigger signal is raised.

- **Trigger-less readout**: Detector pushes data at its speed and the downstream DAQ must keep the pace.





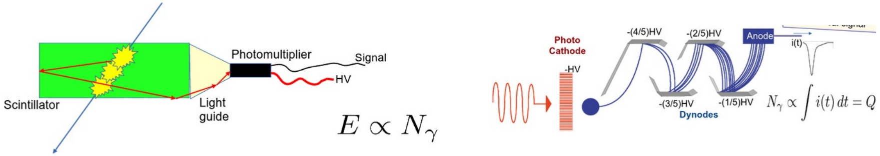
Some examples of DAQ hardware

- Measuring energy deposition, e.g.

  Detector: Scintillator

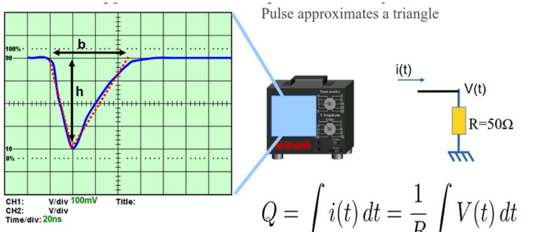
  Electronics: Photomultiplier

  Readout: ADC, QDC on oscilloscope



$E \propto N_\gamma$

$N_\gamma \propto \int i(t)\,dt = Q$



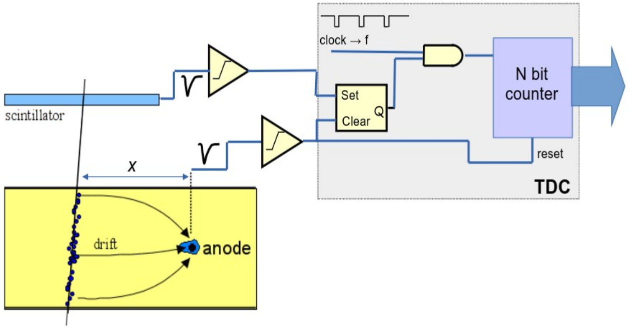
$$Q = \int i(t)\,dt = \frac{1}{R}\int V(t)\,dt$$

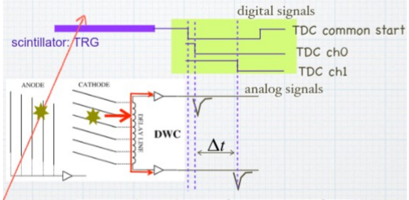
$$Q \approx \frac{1}{R}\frac{bh}{2} = \frac{1}{50\Omega}\frac{(3.5\cdot(20\text{ns}))(4\cdot(100\text{mV}))}{2} = 280\text{pC}$$

- Measuring position, e.g.

  Detector: Wire chambers, DWC.

  Readout: TDC





$y = \alpha \cdot \Delta t + \beta = \alpha \cdot (t_{top} - t_{bottom}) + \beta$

Modular electronics – what & why?

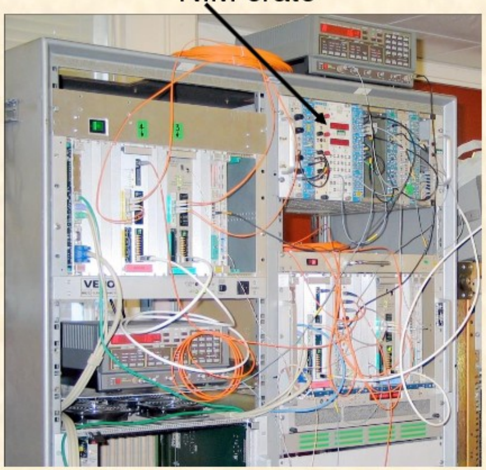
- Modularizing DAQ electronics helps in these respects:
  - Allows for the **re-use of generic modules** in different applications.
  - Limiting the complexity of individual modules increases their **reliability and maintainability.**
  - You can profit from **3rd party support** for common modules.
  - Makes it easier to achieve **scaleable** designs.
  - **Upgrades** (for performance or functionality) are less difficult.
  - Competition gives you **better prices and alternative suppliers.**
  - Standards make it easier to define **interfaces between sub-systems.**

- **NIM modules** (usually): Need no software, are not connected to a computer, and are used to implement trigger logic.
- These functions (and many others) are available: **Discriminators, Coincidences, Amplifiers, Timers, Logic gates (and / or), Level converters, HV power supplies.**

Initially (1964): **NIM = Nuclear Instrument Modules**

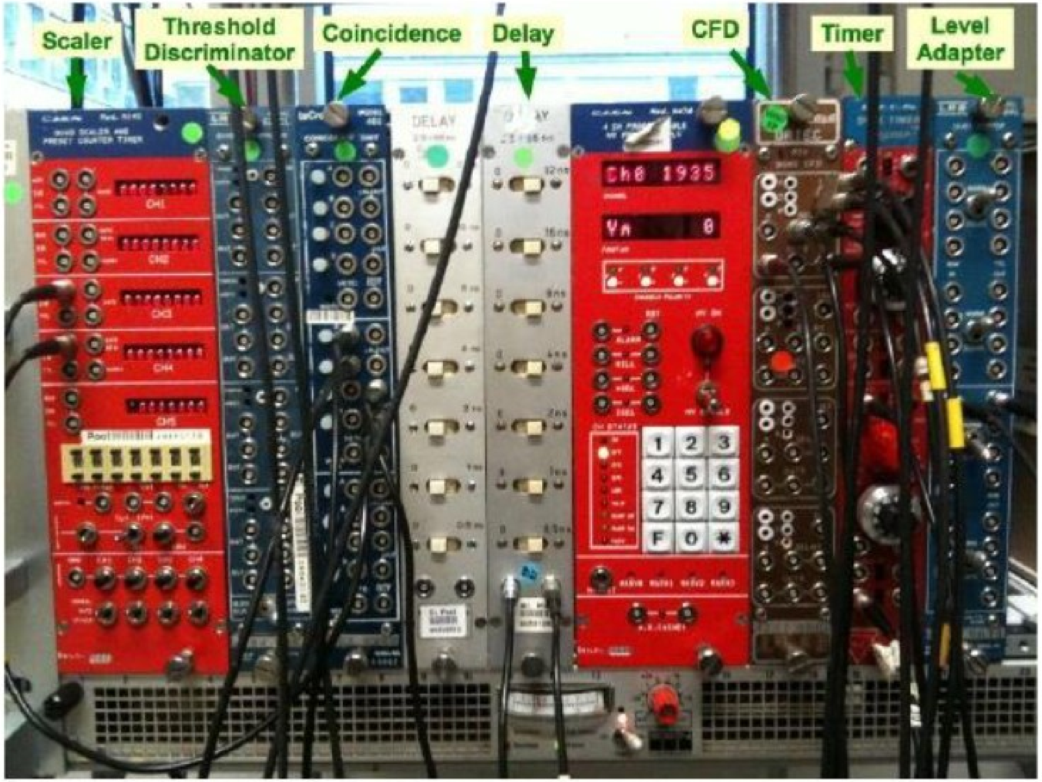
- But it was used outside of "nuclear science"
  - Therefore: **NIM = National Instrument Modules**
    - But is was used outside of the USA
      - Therefore: **NIM stands for NIM**

NIM crate



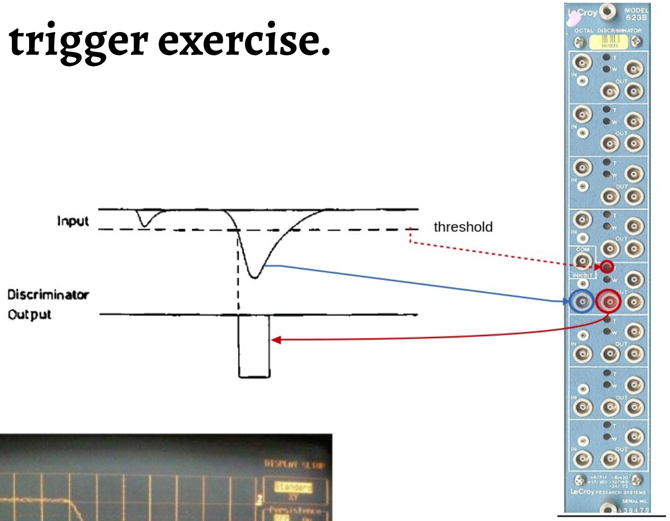
A small DAQ system

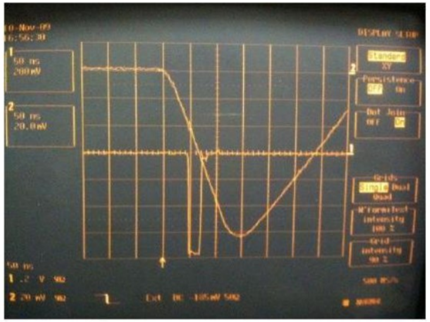
Lab exercise – I



Trigger logic unit (NIM modules)

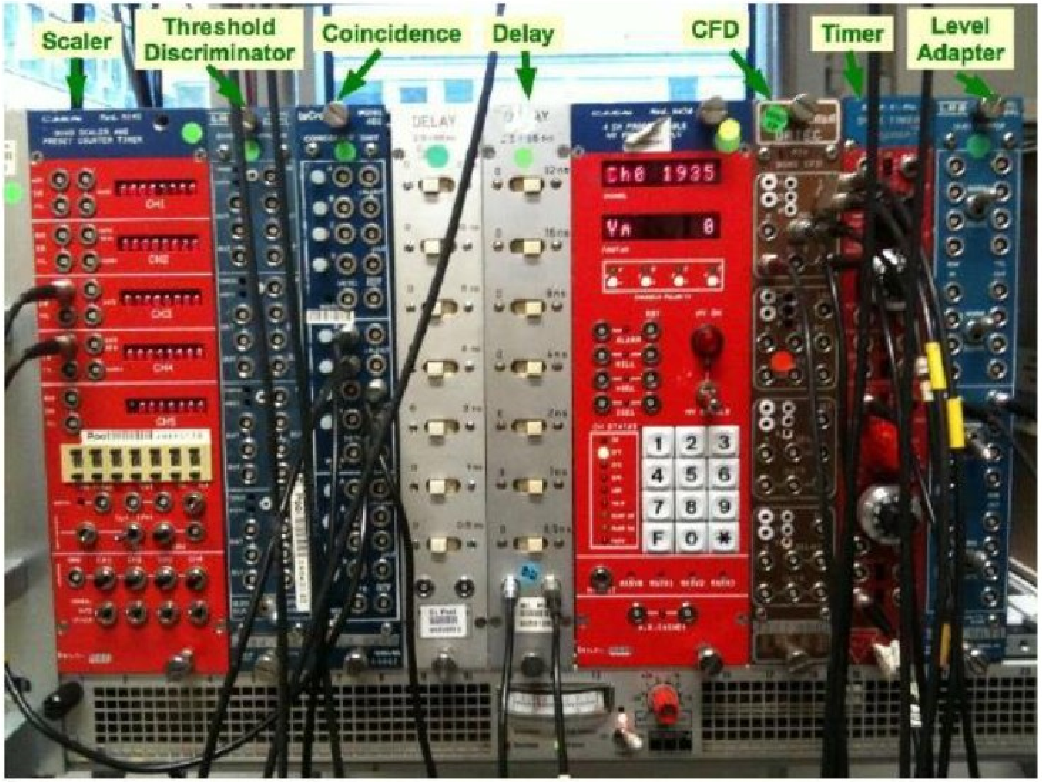
A simple trigger exercise.





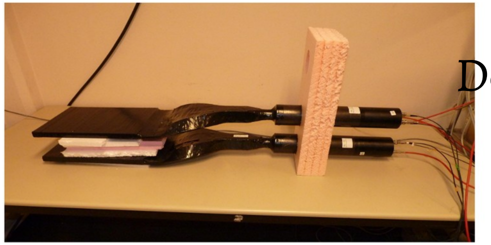
- Looking at output from Threshold discriminator.
- Output from Constant fraction discriminator (CFD).
- Comparing the delays on the discriminated signal wrt reference.

Lab exercise – II

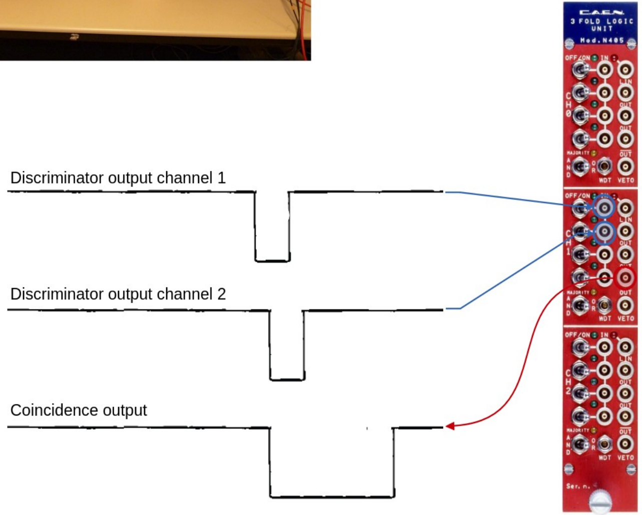


Trigger logic unit (NIM modules)

Detector and trigger exercise.

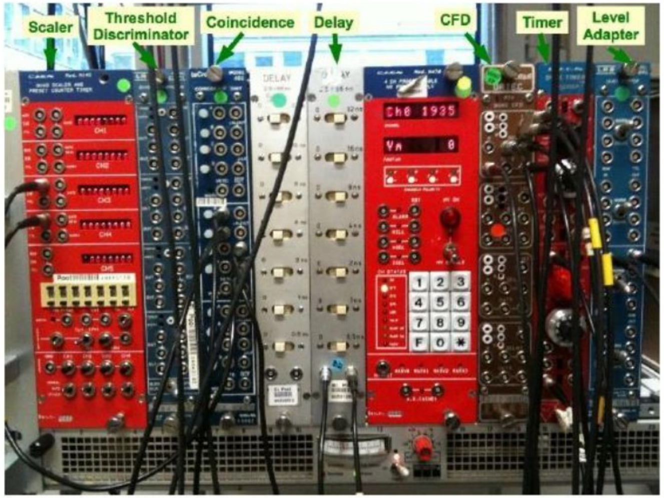


Detector (Scintillators)



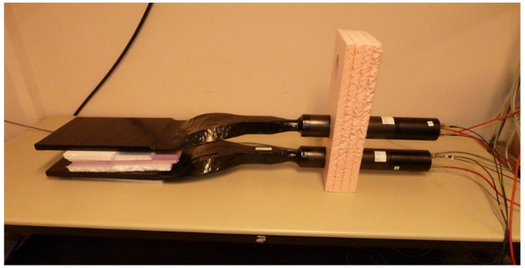
- To measure the rate of cosmic muons.
- Calculate the total charge produced from PMTs.

Lab exercise – III

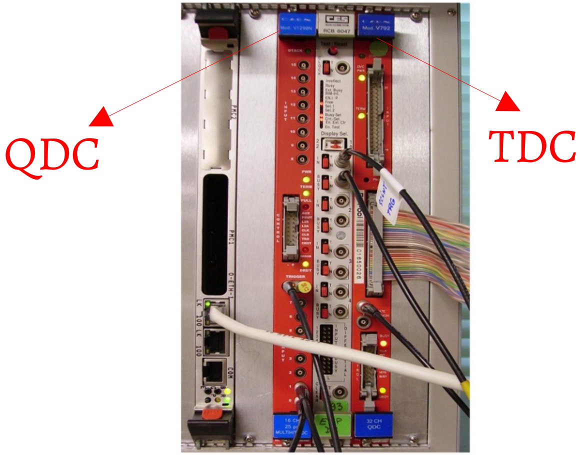


Trigger logic unit (NIM modules)

A small physics experiment – Detector, trigger and data acquisition.



Detector (Scintillators)



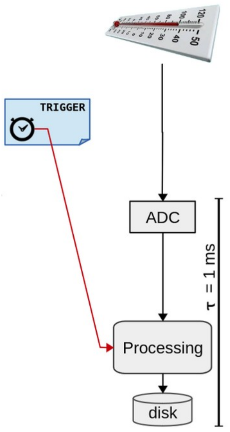
DAQ system (VMEbus)

- The "beam" is provided by cosmic rays (muons) and the detector consists of a pair of scintillation counters.
- The trigger logic, built from NIM electronics, forms a coincidence between the signals from the scintillation counters which indicates that a muon has traversed the detector.
- A data acquisition system based on VMEbus is used to record the pulse heights from the scintillation counters and measure the time of flight of the muon. The overall run control and monitoring is provided via software running on a (Linux) single board computer (SBC).
- An event monitoring program produces histograms of the QDC and TDC channel data which allow to compute the charges of the input signals to the QDC and the speed of the cosmic muons.

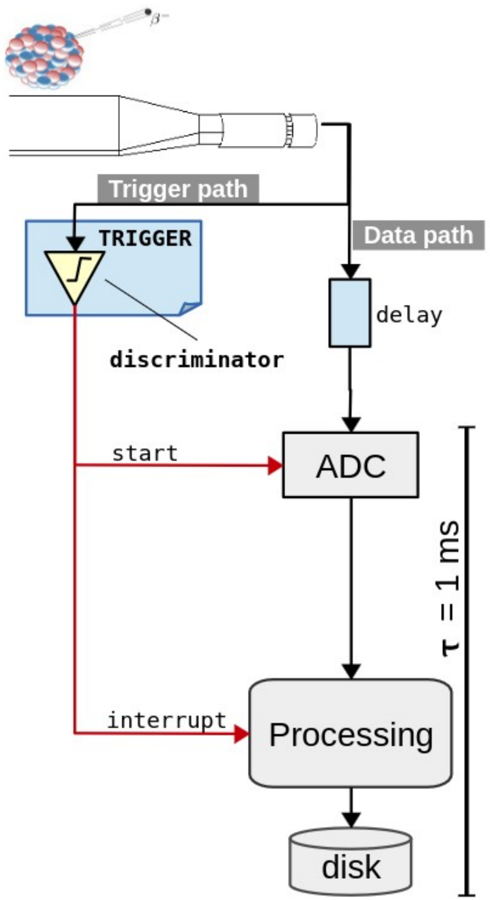
DAQ basics: BUSY logic

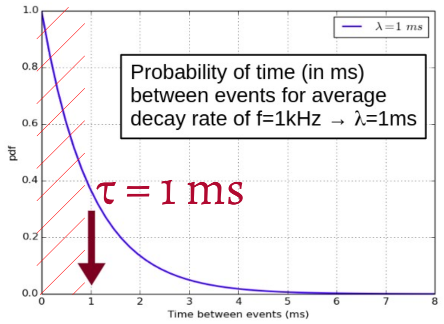
For a uniform measurement with a **periodic** trigger, system limited by time 'τ' to process an event.

Hence, DAQ rate is R = 1/τ i.e. 1 KHz.



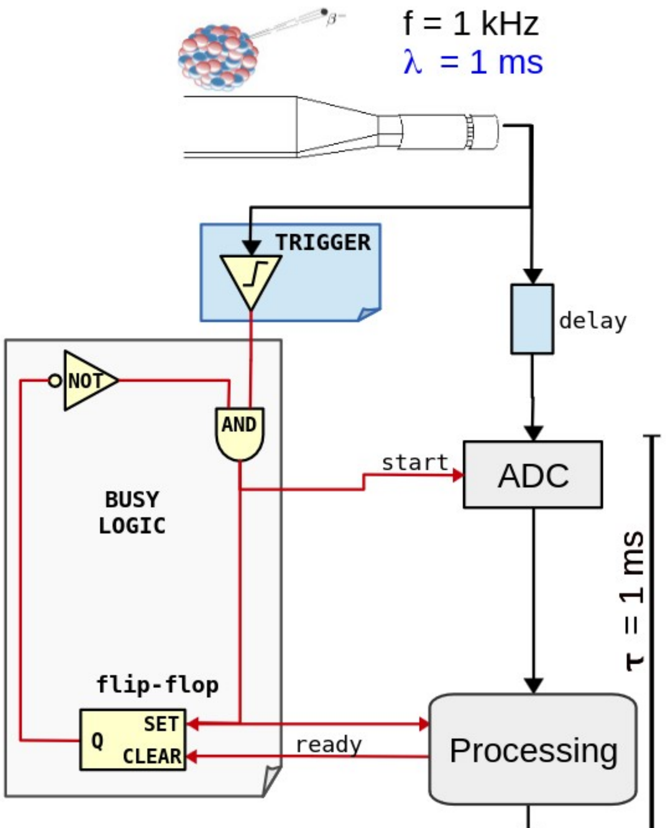
- But in reality, events are **asynchronous and unpredictable**, e.g. beta decay. Hence, we need a physics trigger (Discriminator).
- A delay is introduced to compensate for the trigger latency.

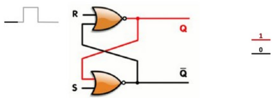




- What if the trigger is created when the system is busy?
- Need a feedback mechanism to know if data processing pipeline is free to process a new event – **busy logic**.

DAQ basics: De-randomization





- probabilities: P[busy] = $\nu\ \tau$; P[free] = 1 - $\nu\ \tau$

Therefore:

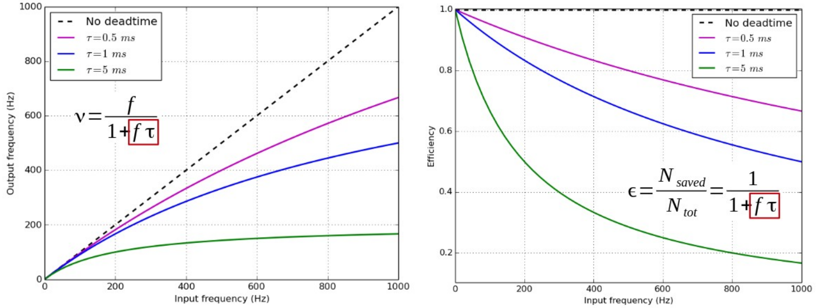
$$\nu = f\,P[free] \;\Rightarrow\; \nu = f(1-\nu\tau) \;\Rightarrow\; \nu = \frac{f}{1+f\,\tau}$$

Avg. rate of DAQ output

- Due to stochastic fluctuations
  - DAQ rate always < physics rate $\nu = \frac{f}{1+f\,\tau} < f$
  - Efficiency always < 100% $\epsilon = \frac{N_{saved}}{N_{tot}} = \frac{1}{1+f\,\tau} < 100\%$

- So, in our specific example
  - Physics rate 1 kHz
  - Deadtime 1 ms

$$\begin{array}{l} f = 1\,kHz \\ \tau = 1\,ms \end{array} \rightarrow \begin{array}{l} \nu = 500\,Hz \\ \epsilon = 50\% \end{array}$$

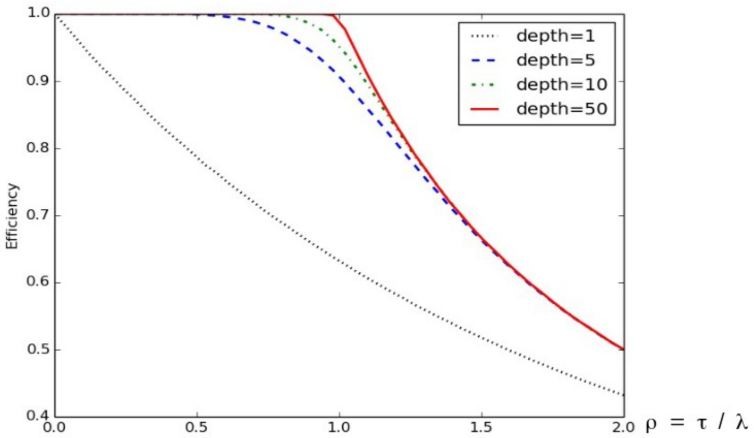


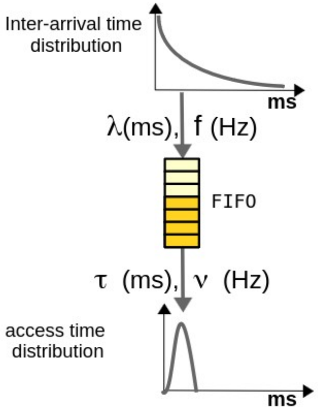
- In order to obtain ε~100% (i.e.: ν~f) → fτ << 1 → τ << λ
  - E.g.: ε~99% for f = 1 kHz → τ < 0.01 ms → 1/τ > 100 kHz

Over-design DAQ system by a factor of 100!!

Solution: De-randomize the events.

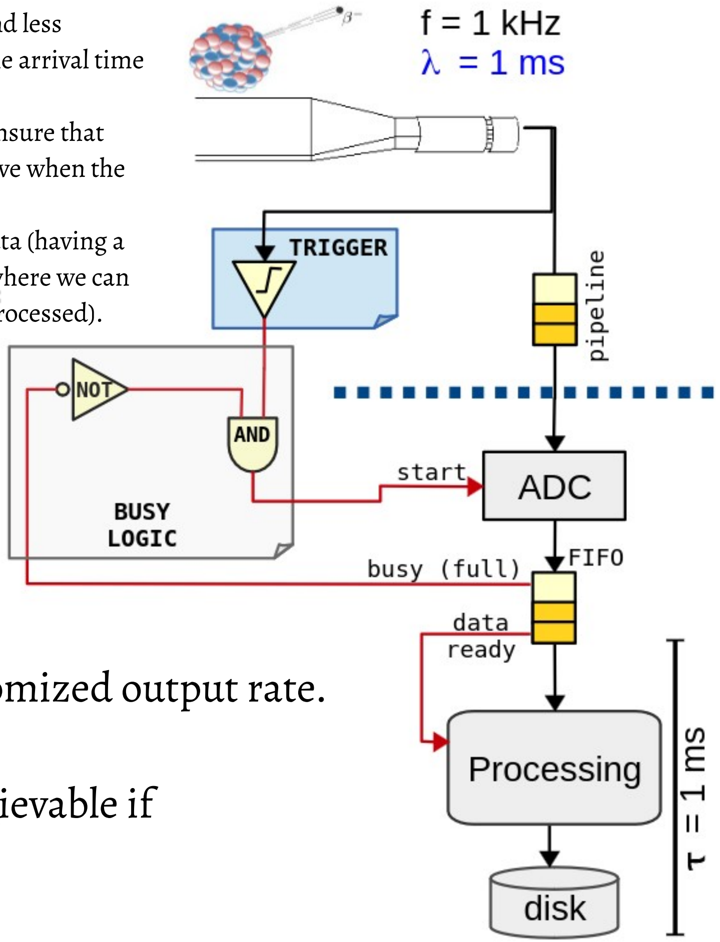
DAQ basics: Buffers





- ρ > 1: the system is overloaded (τ > λ)
- ρ << 1: the output is over-designed (τ << λ)
- ρ ~ 1: using a queue, high efficiency obtained even w/ moderate depth

- Make the system more deterministic and less dependent on the arrival time of our signals.
- Then we could ensure that events don't arrive when the system is busy
- Buffering the data (having a holding queue where we can slot it up to be processed).

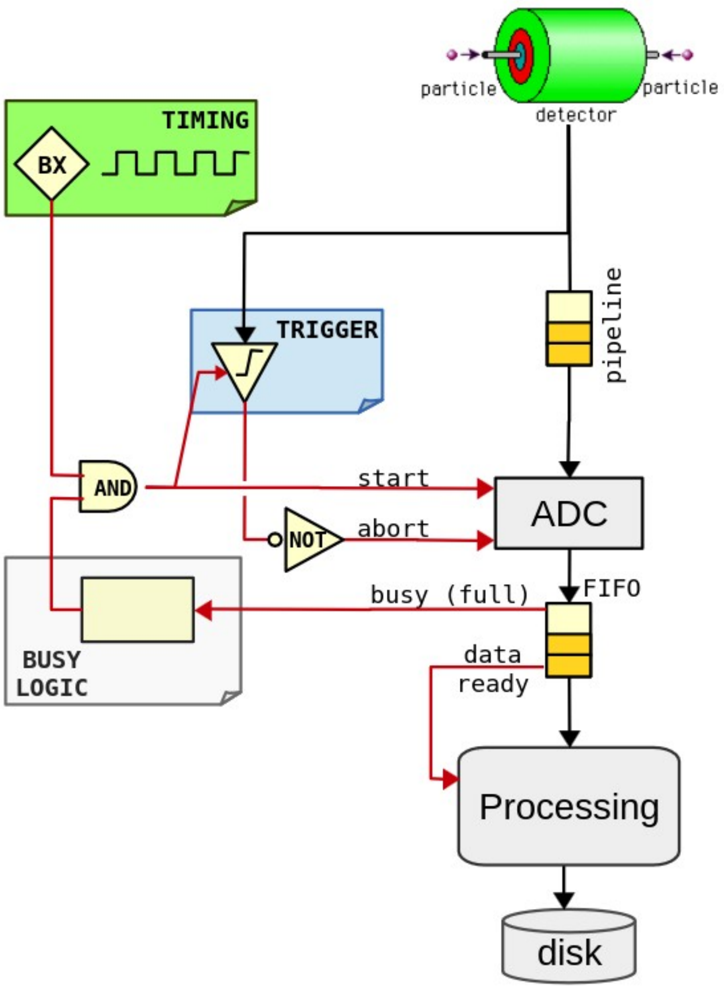


- FIFO (buffers) provides a steady & de-randomized output rate.
- 'Busy' is now defined by buffer occupancy.
- 100% efficiency with minimal deadtime achievable if
  - ADC rate >> f
  - Data processing and storage rate > f.

Q- Do we need randomization in collider setups?

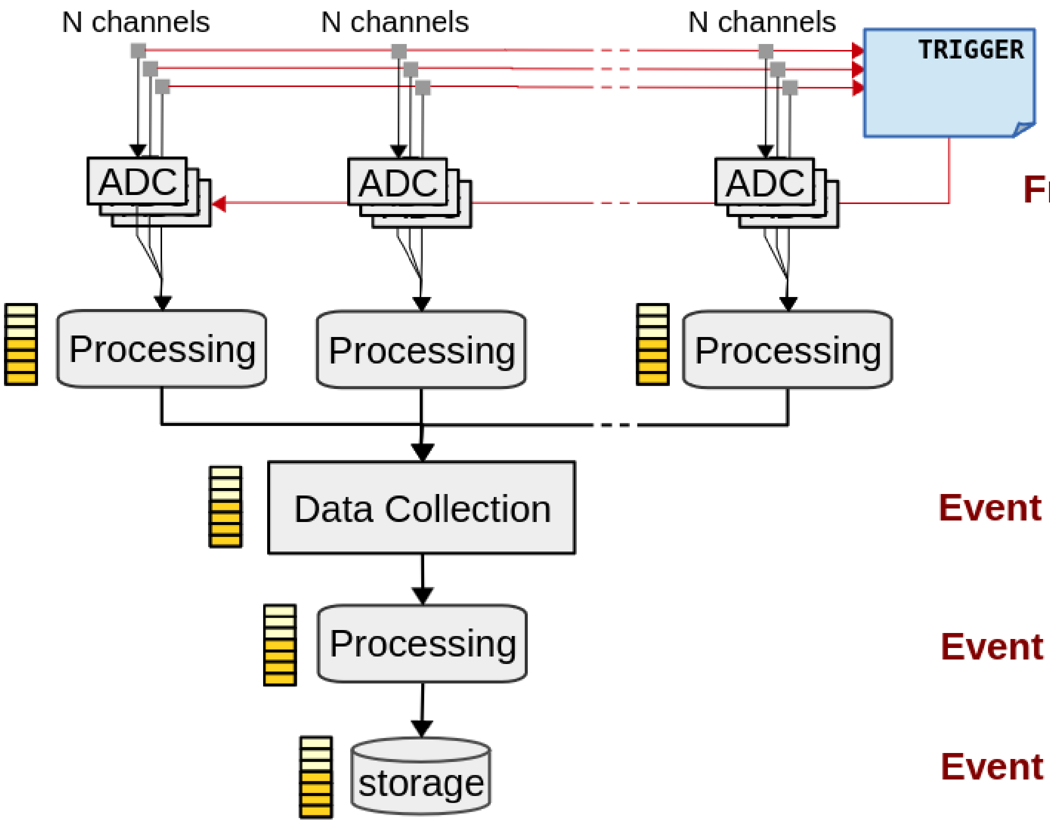
Buffer in collider setups

- Particle collisions are synchronous.
- But time distribution of triggers is random, since good events are unpredictable.
- De-randomization is still needed.
- More complex busy logic to protect buffers and detectors.



Adding more channels – realistic detector

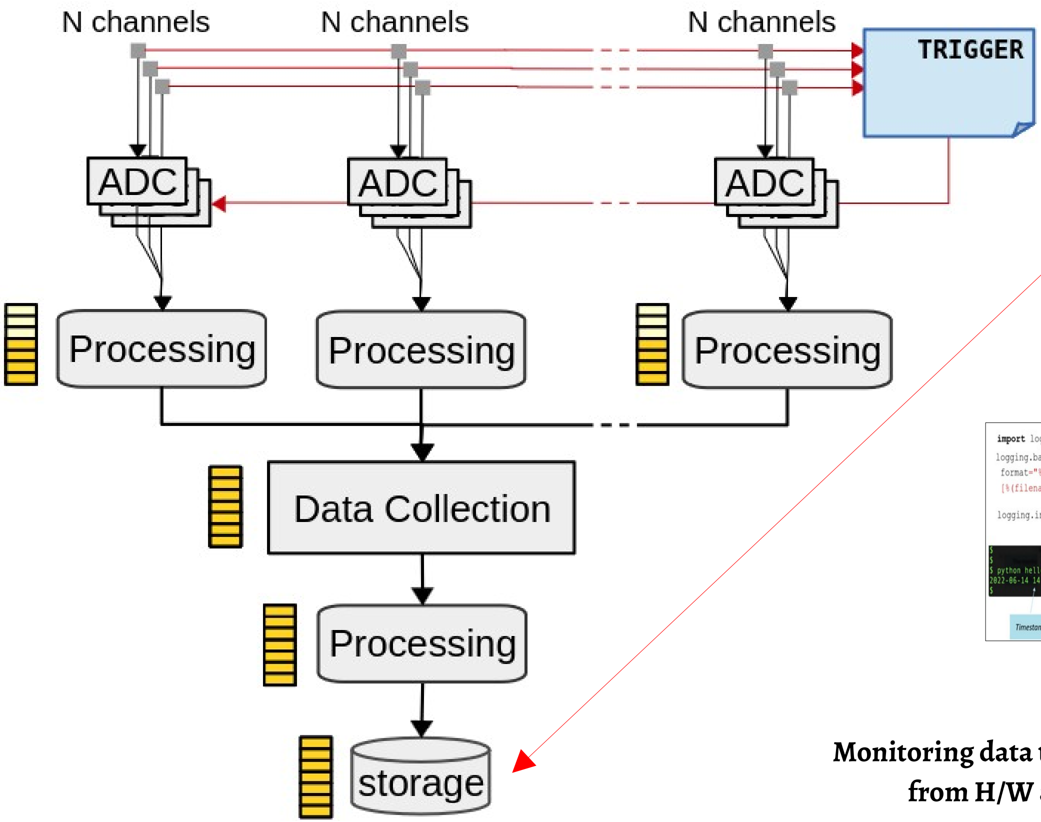
- DAQ is essentially multi-level buffering system.



Front-End	data digitization data **buffering**
Readout	data extraction data formatting data **buffering**
Event Building	event assembly event **buffering**
Event Filtering	event rejection event **buffering**
Event Logging	file storage file **buffering**

Adding more channels – realistic detector

- DAQ is essentially multi-level buffering system.



Creates **"back-pressure"**

When a system/buffer gets saturated, the pressure is propagated upstream.

- Buffer occupancy can be monitored by the **DAQ online monitoring system.**

The basic architecture of monitoring software

Prometheus (back-end)

Monitoring & alerting platform, collects and stores metrics as time-series data.

+ Grafana (front-end)

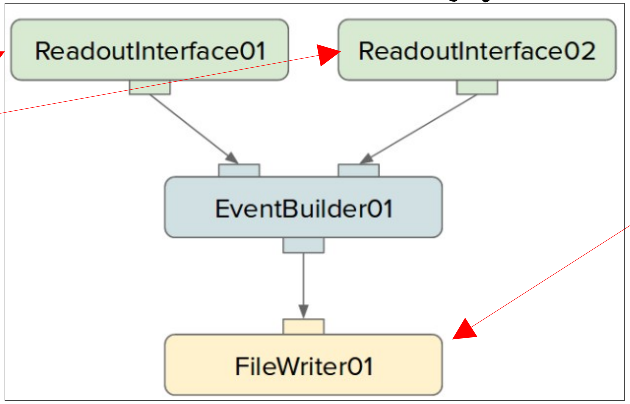
Analytics & interactive visualization web app. To ingest data sources, query this data & display on customized charts for easy analysis.

Monitoring data type: Time-series data from H/W and S/W components.

DAQ online software – lab exercise

Readout modules generating random data and writing them to buffer, waiting to be processed by the event builder application.

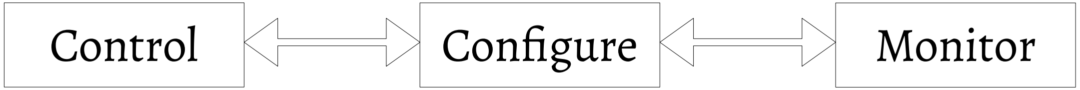
DAQ system



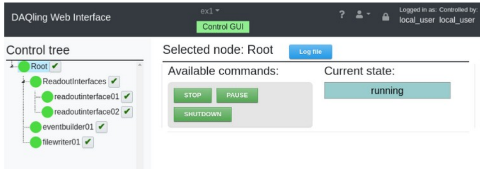
Once the events have been built, FileWriter application writes them to permanent storage for offline analysis.

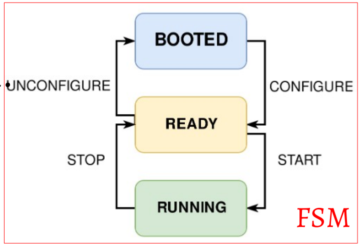
DAQling framework

DAQ systems of small and medium-sized experiments such as NA61/FASER.

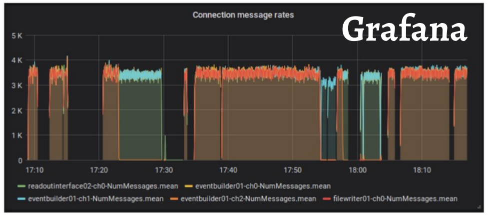


Web-based entity i.e. "run control", handles modules via Final State Machine (FSM) paradigm





Through JSON configuration files, setting up modules, hosts, sender & receiver ports, distributed system.



DAQ system – from FE to processing/storage

- **Very Front End**
  ADC, low-level compression, zero suppression, loss-less compression, rad-tolerant, low power.

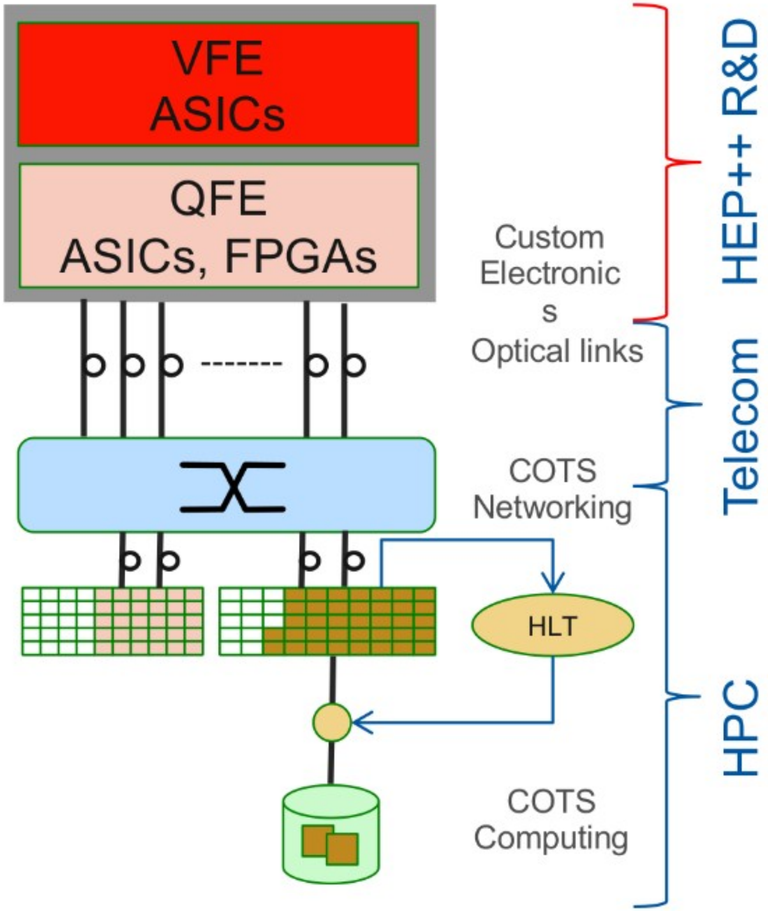
- **Quasi Front End**
  Medium scale aggregation, local reconstruction, transition to standard protocol on optical links.

- **Commodity Of The Shelf (COTS) switched networks**
  Further aggregation for event building

  Gigabit to infiniband many network solutions are available. Disk servers, tape robots, custom solutions… many choices…

- **COTS computing**
  With co-processors e.g. GPU, FPGA. Final selection before storage.

- **Networking & Storage**

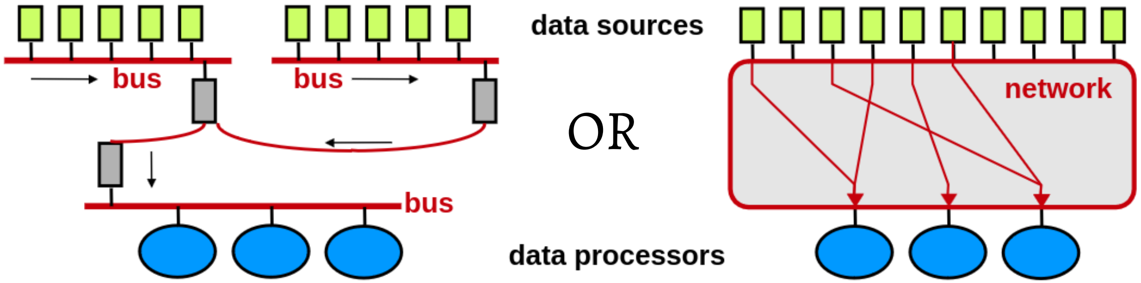


Networking in DAQ

- Data Acquisition Systems
  – Receive content (event fragments) from detector back-end over multiple links O(1000), build events from fragments, filter events (online processing), and store it on disks in a single place e.g. CERN tier-0 for further distribution and offline processing
- Worldwide LHC Computing Grid (WLCG)
  – Content is distributed and stored in 170 sites in 42 countries for processing in WLCG
  – Example of WAN (Wide-area network)
- Several data fragments – from different parts of detector (sources).
- Data flow from readout system to event filter to data storage.
  - To be assembled together in the event builder, to be stored on self-consistent files.

Readout topology →

How to connect data sources & data destinations?

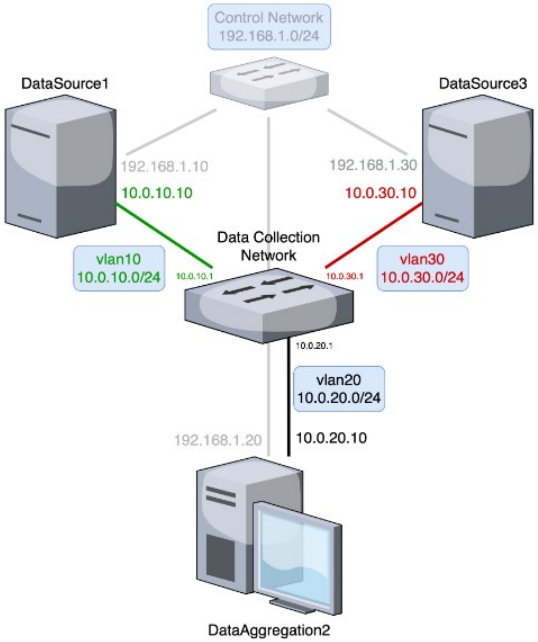


- Simple, but scalability issues.
- Shared bandwidth, limited number of devices depending on bus length.

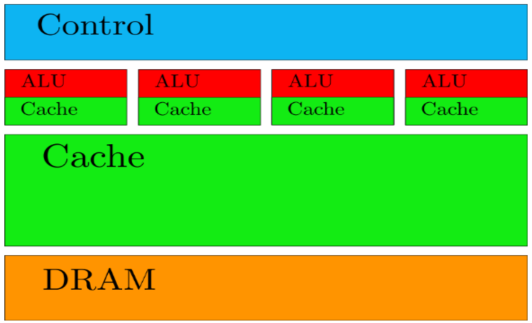
- All devices are equal, communicate directly with each other via messages. Highly scalable.
- No arbitration. Congestion handled via buffering.

Lab exercise

- Through the use of a simplified network setup, we learned about:
  - configuration: MAC and IP addresses, switch management, VLAN, routing, DNS
  - monitoring: SNMP, traffic analysis
  - testing: performance benchmarking, TCP/IP
  - troubleshooting: tcpdump
  - optimizations: DHCP, DNS
- The lab setup consisted of:
  i. 2 headless desktop computers (pc-isotdaq-1 and pc-isotdaq-3) mimicking readout system servers that read data out of the detector, process them, and transfer them: they are the data sources.
  ii. 1 desktop computer (pc-isotdaq-2) mimicking a data aggregator server receiving and formatting data from different data sources: it is the data destination.
  iii. 1 enterprise-grade network switch mimicking a data collection network.
  iv. 1 control network used to connect to the different computers, collect monitoring data, and allow us to configure and use the data collection network. The control network is running on the switch in a separate VLAN to avoid interference with the data collection network.

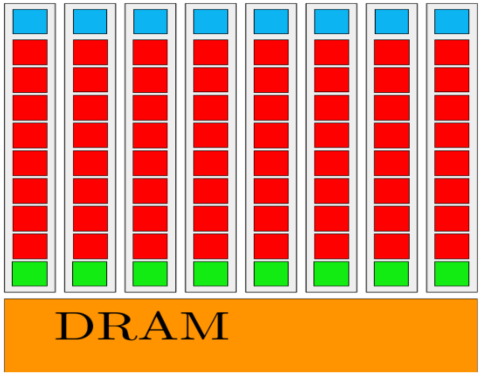


GPUs in HEP



CPU: latency oriented design

vs



GPU: throughput oriented design

Winner: Heterogeneous computing

- CPUs for sequential parts (can be 10X faster than GPU for sequential code)
- GPUs for parallel part where throughput wins (can be 100X faster than CPU for parallel code)

Lab exercise: Matrix multiplication with GPU!

Where to use GPU in the triggers?

- In High Level Trigger

It is the "natural" place. If your problem can be parallelized (either for events or for algorithm) you can gain factor on speed-up → smaller number of PC in Online Farm.

- In Low Level Trigger

Bring power and flexibility of processors close to the data source → more physics.

GPU in Run3

CMS

- Pixel tracking for HLT on GPU (Patatrack).
- Clustering by Energy (CLUE) on GPU – both electromagnetic & hadronic.
    - CLUE on CPU is 30x faster.
    - CLUE on GPU is additional 6x faster.
- New parallelizable algorithm designed for HGCAL to be run on GPU.
- HLT vertex reconstruction can also be offloaded to GPU.

ATLAS

- Demonstrators in Run-I
    - Inner detector, tracking based on Cellular Automata (CA).
    - Calorimeter, jet finding and clusterization based on CA.
    - Muon, tracking based on hough transforms.
- As a result, tracking seeding algorithm faster by 28x times. Conclusion: not to use GPU for marginal gains.
- Mainly because of Athena software lacking concurrency and multithreading.
    - New: Athena MT framework
- Search for hardware trigger tracking also motivating GPU development.

ISOTDAQ 2023 – List of topics covered

- ***DAQ (6 lectures + 2 labs)***
- ***Trigger (4 lectures + 2 labs)***
- ***Detector readout and controls (6 lectures + 2 labs)***

In this talk (roughly)

- ***FPGA (2 lectures + 3 labs)***
- ***Data transfer/storage (4 lectures + 3 labs)***
- ***Programming, Machine learning (2 lectures)***
- ***Microcontrollers (1 lecture + 1 lab)***
- ***LabView (1 lecture + 1 lab)***
- ***TDAQ in Space, Medical imaging applications (1 lecture each)***

Part 2: Luc Le Pottier