Electronics, Trigger and Data Acquisition

Summer Student Programme 2012, CERN

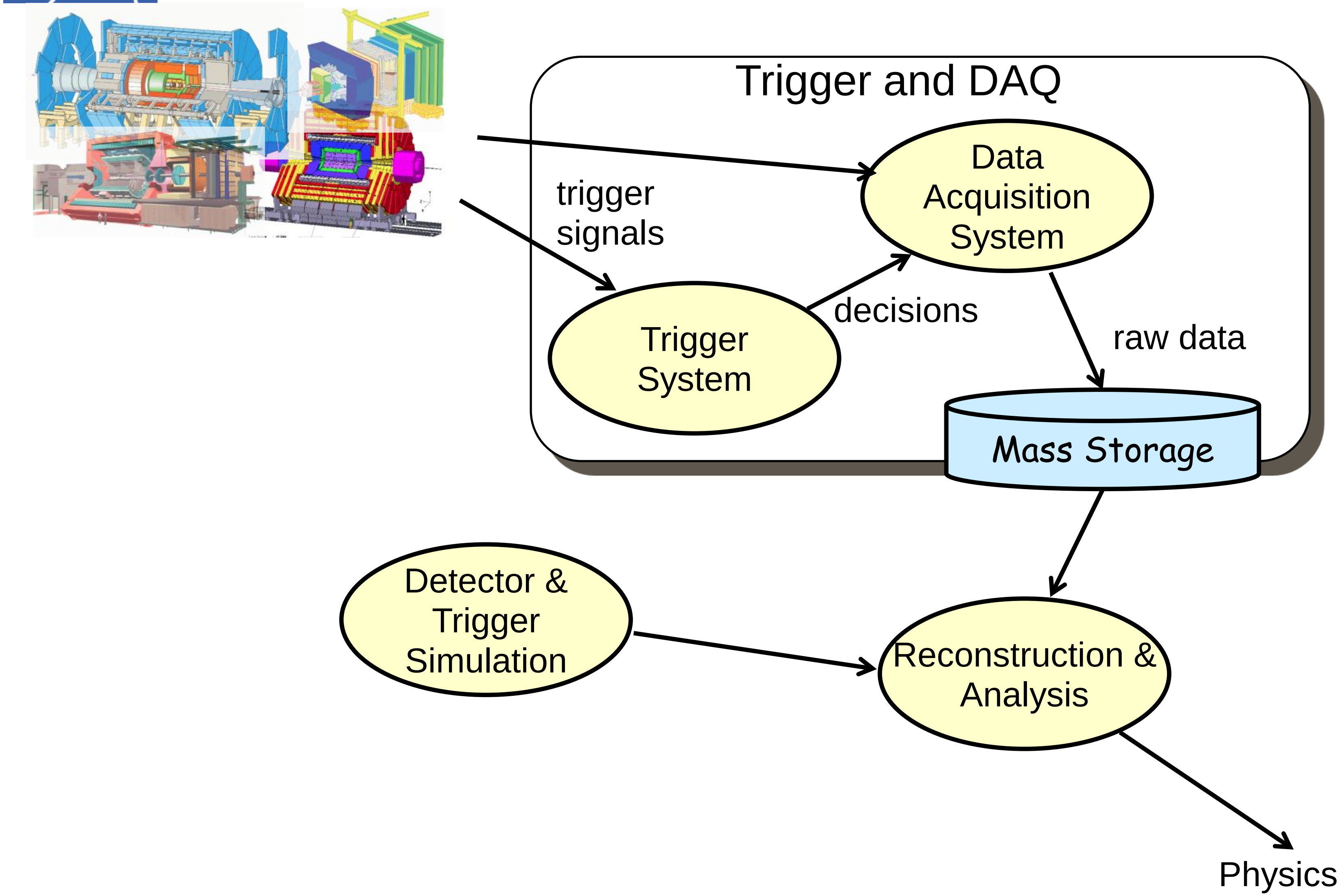
W.Vandelli CERN/PH-ATD

Introduction

- ➔Data acquisition is **not an exact science**. It is an alchemy of electronics, computer science, networking, physics
  - …, money and manpower matter as well, …
- ➔I will mostly refer to DAQ in High-Energy Physics
- ➔Topics are pretty much correlated → You will experience this through the lecture non-linearities

Material and ideas from my predecessors (N.Neufeld and C.Gaspar), the “Physics data acquisition and analysis” lessons given by R.Ferrari at the University of Parma, Italy, “Analog and Digital Electronics for Detectors” of H. Spieler and all lectures of ISOTDAQ schools, in particular M.Joos and C.Schwick

General Overview



➔Overall the main role of Electronics Trigger & DAQ is to process the signals generated in a detector, storing the interesting information on a permanent storage

Electronics

What is needed for?

➔ Collect electrical signals from the detector. Typically a short current pulse

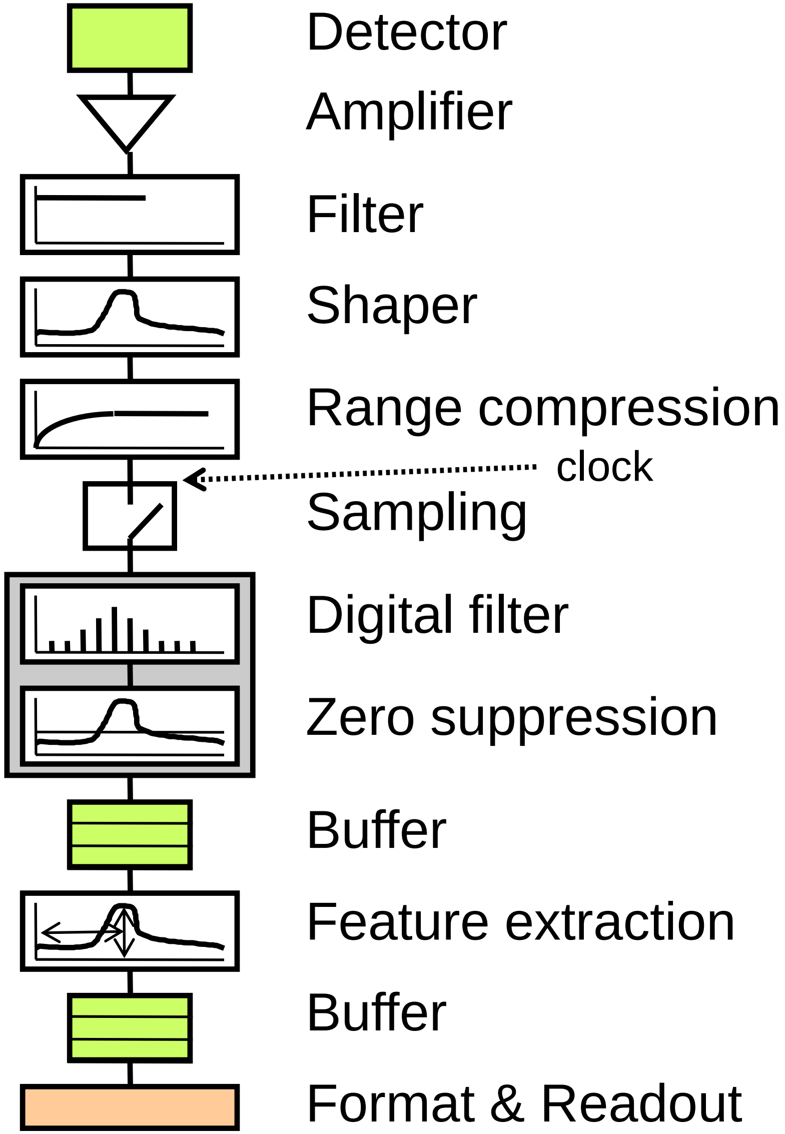
➔ Adapt the signal to optimize different, **incompatible**, characteristics → Compromise

- minimum detectable signal
- energy measurement
- signal rate
- timing
- insensitivity to pulse shape

➔ Digitize the signal

- allow for subsequent processing, transmission, storage using digital electronics → Computers, Networks, …

Read-out chain



➔ Front-end electronics very specialized

- custom build to match detector characteristics

➔ Cannot discuss all design and architecture details

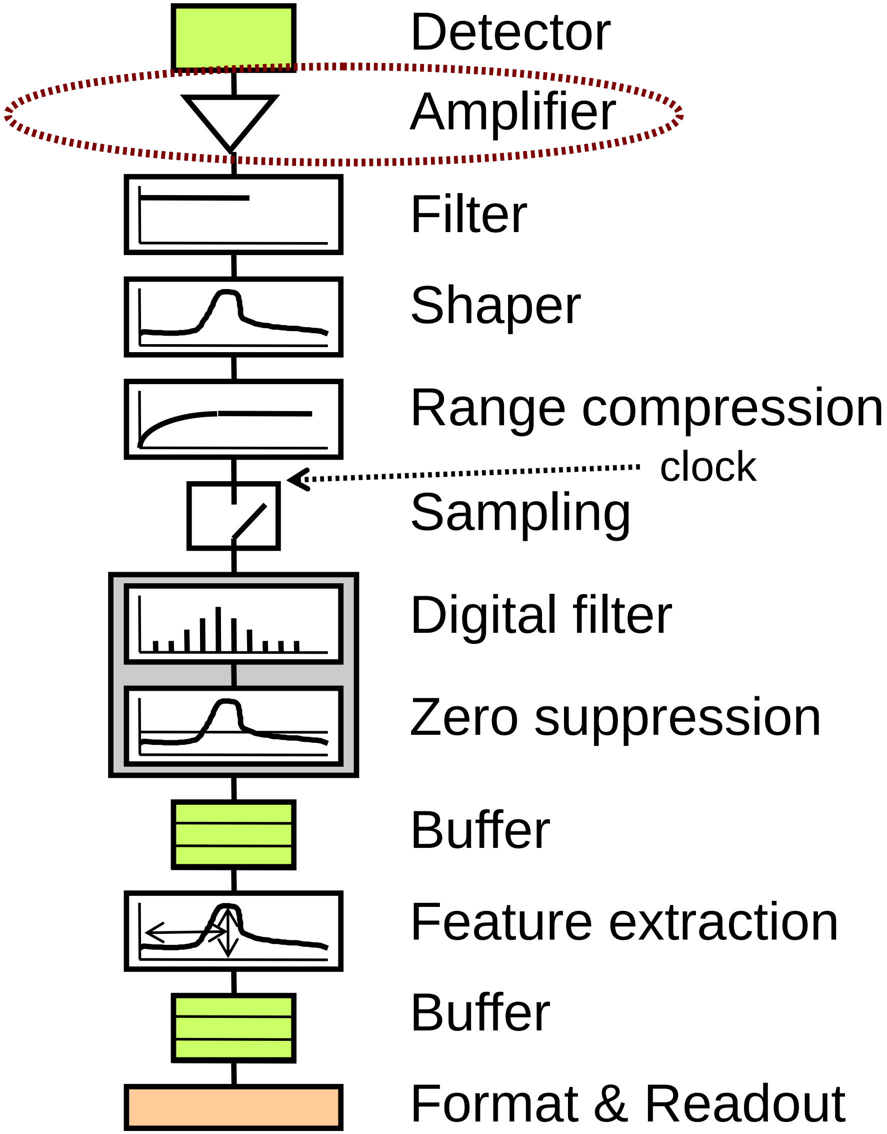
- if you are into electronic design you already know more than me

➔ Find yourself dealing or choosing commercial electronics

- provide you with base guidelines

➔ Selected functions and principles

Read-out chain



What is a signal?

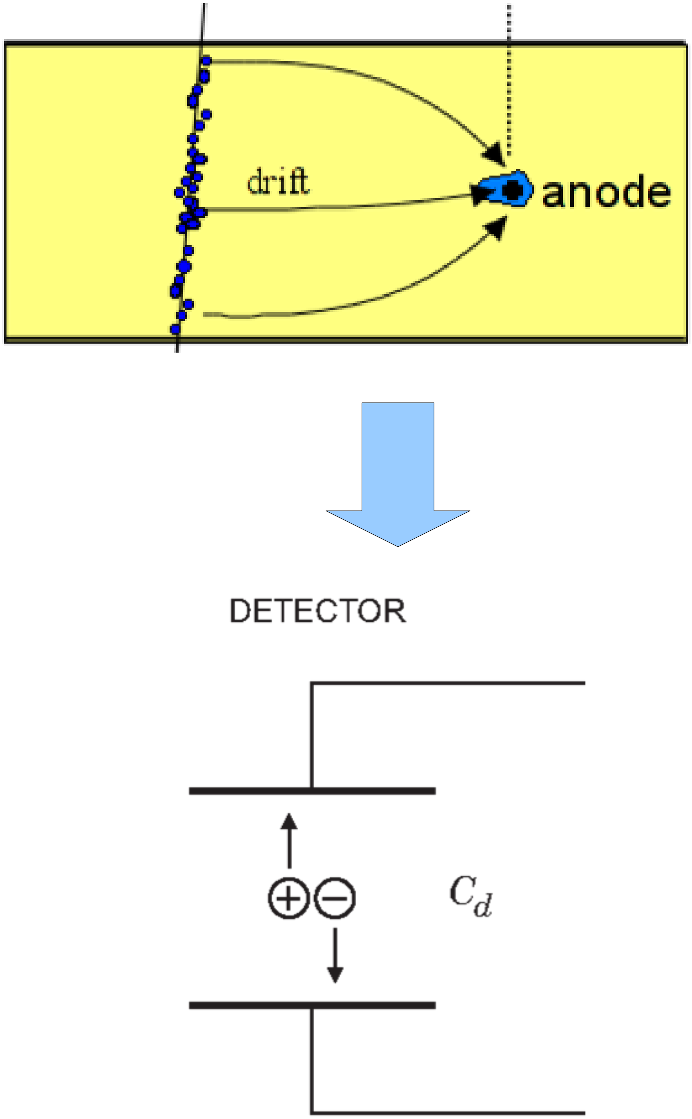
➔ Detector electrically represented by a capacitor Cd

- more realistic scheme will include other contributions

➔ The interaction with a passing particle generate a small current pulse i_s due to the release of energy E

$$E \propto Q_s = \int i_s(t)dt$$

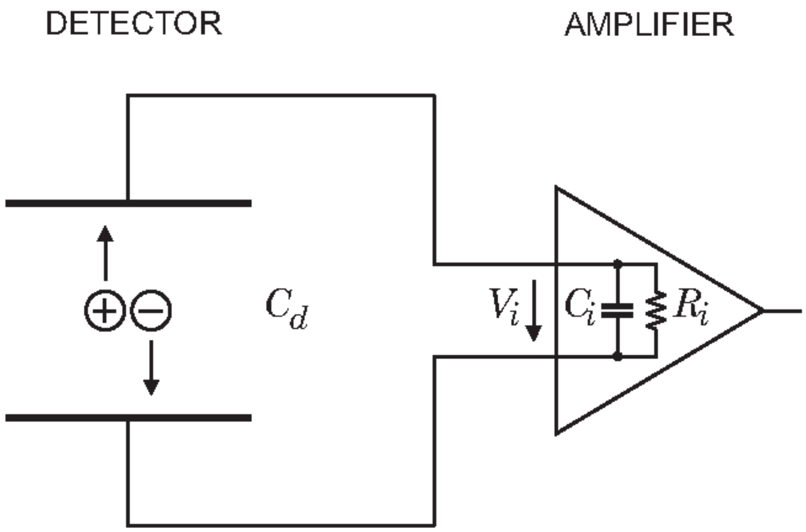
➔ Current pulse duration can range from 100 ps to O(10) □s



Amplification

➔Signals are possibly very small

- improve signal resolution, adapt it to next stages
- improve signal-to-noise ratio …



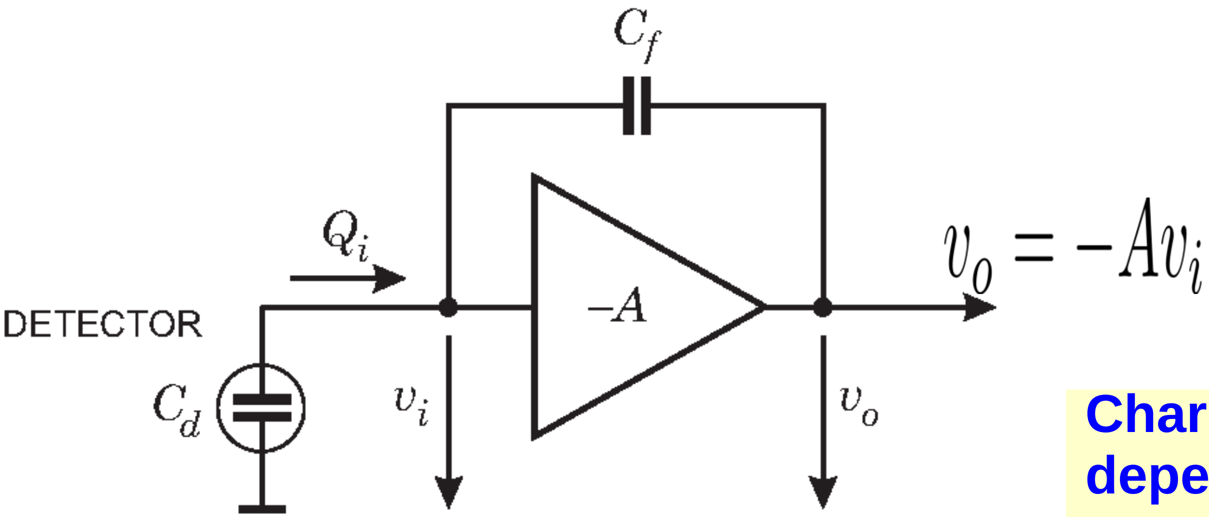
Using a simple voltage amplifier, the sensed **input voltage V_i depends on the detector capacitance.**

Detector capacitance could be a function of the operation point (e.g. high voltage) and/or detector dimension.

Additional **calibration** efforts

$$V_i = \frac{Q_S}{C_d + C_i}$$

Charge-Sensitive Amplification



Charge amplification only depends on a well-controlled component

$$A_Q = \frac{v_o}{Q_i} = \frac{Av_i}{C_f(A+1)v_i} = \frac{A}{A+1}\frac{1}{C_f} \approx \frac{1}{C_f} \quad (A \gg 1)$$

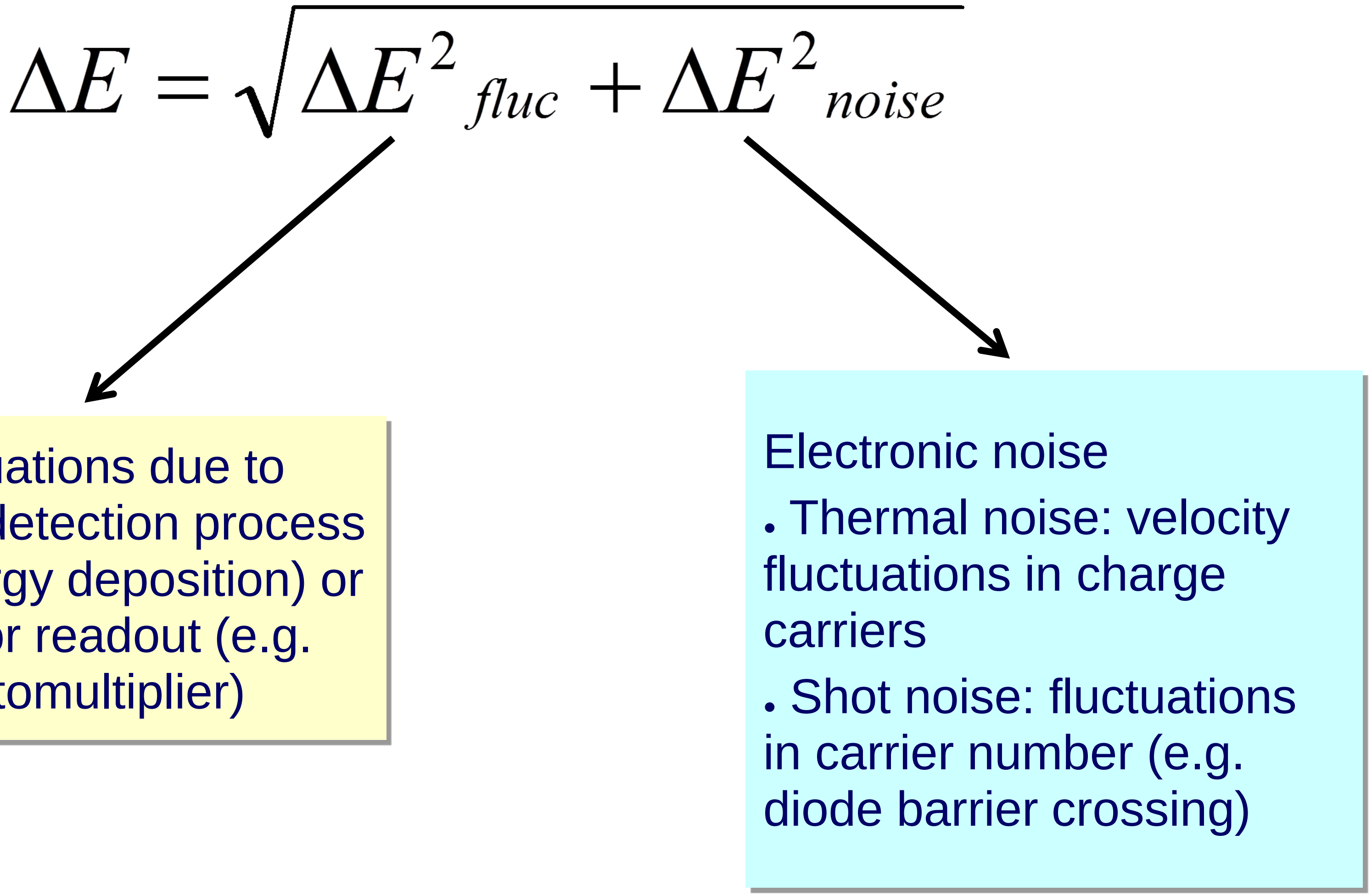
$$\frac{Q_i}{Q_s} = \frac{C_i}{1 + \frac{C_d}{C_i}}$$

Large input capacitance to improve charge sharing

Signal-to-Noise ratio

➔Improving signal-to-noise ratio improves the minimum detectable signal

➔Electronic noise does not necessarily dominate in every measurements



$$\Delta E = \sqrt{\Delta E^2{}_{fluc} + \Delta E^2{}_{noise}}$$

Fluctuations due to physical detection process (e.g. energy deposition) or detector readout (e.g. photomultiplier)

Electronic noise

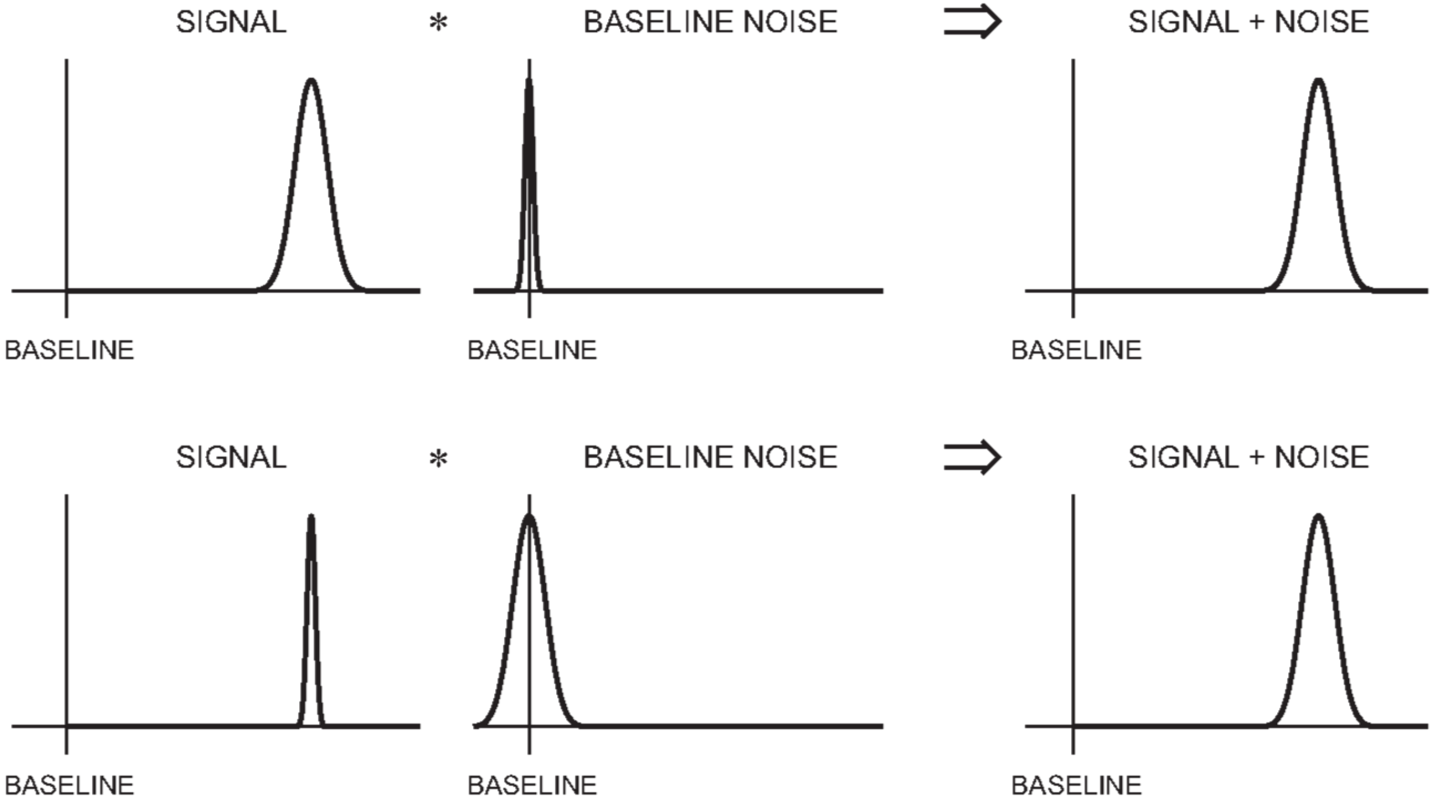
- Thermal noise: velocity fluctuations in charge carriers
- Shot noise: fluctuations in carrier number (e.g. diode barrier crossing)

Signal-to-Noise ratio

➔Improving signal-to-noise ratio improves the minimum detectable signal

➔Electronic noise does not necessarily dominate in every measurements

$$\Delta E = \sqrt{\Delta E^2{}_{fluc} + \Delta E^2{}_{noise}}$$

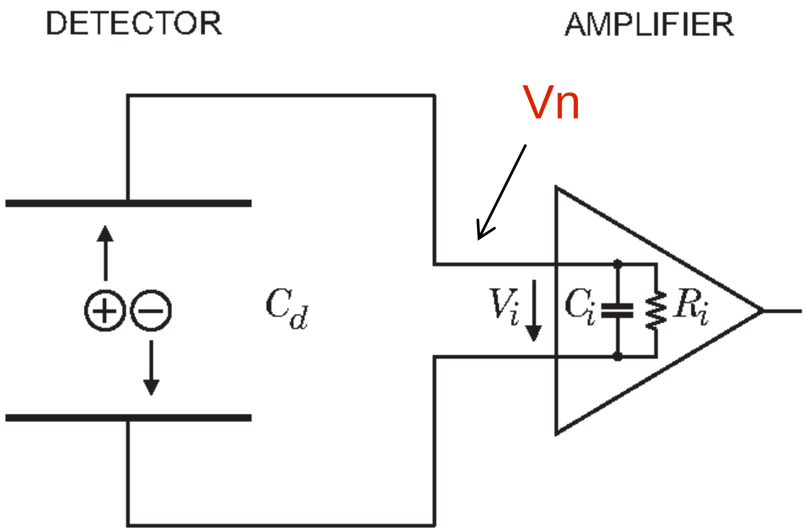


S/N needs optimization

SNR and capacitance

➔Given an signal charge Qs

$$V_s = \frac{Qs}{C_d + C_i}$$



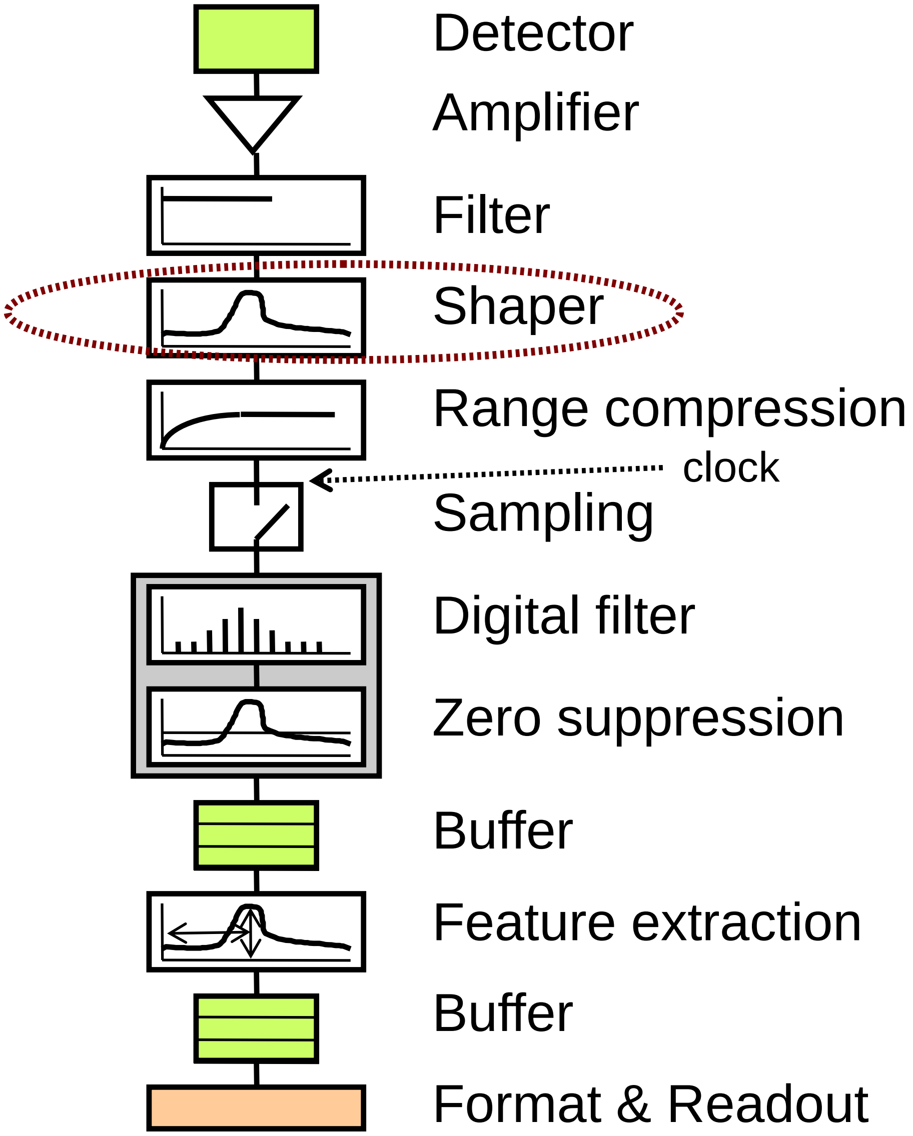
➔Assuming a input noise Vn

$$\frac{V_s}{V_n} = \frac{Q_s}{V_n \underbrace{(C_d + C_s)}}$$

SNR inversely proportional to input capacitance

Thick detectors normally provide larger signals **and noise**

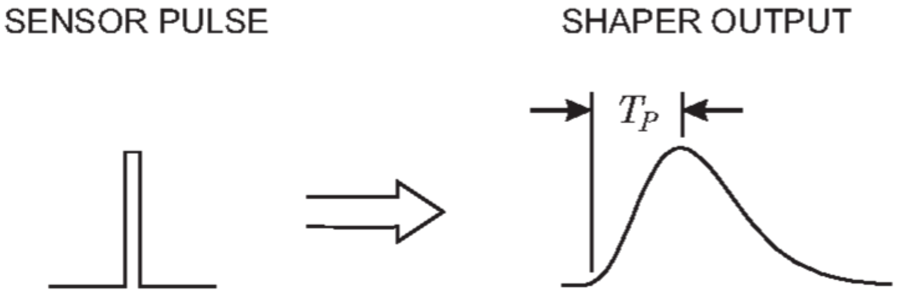
Read-out chain



Pulse shaping

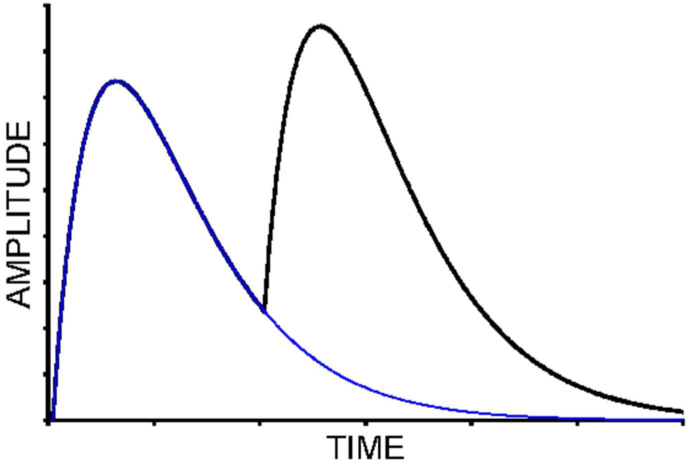
➔ Reduce signal bandwidth → improve SNR

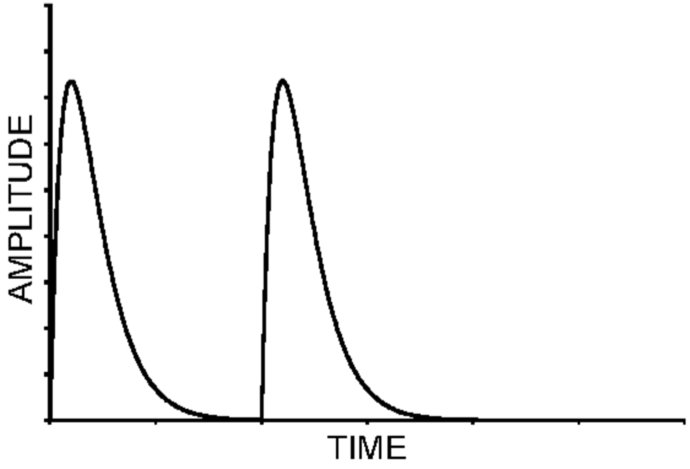
- fast rising signals have large bandwidth
- shaper broadens signals



➔ Limit pulse width → avoid overlap of successive pulses

- increase maximum signal rate at the cost of more noise

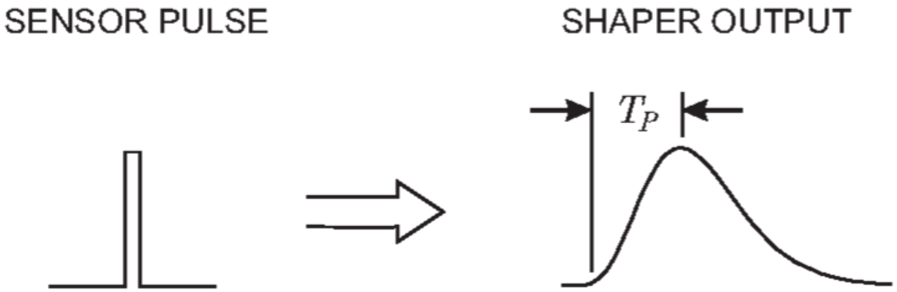




Pulse shaping

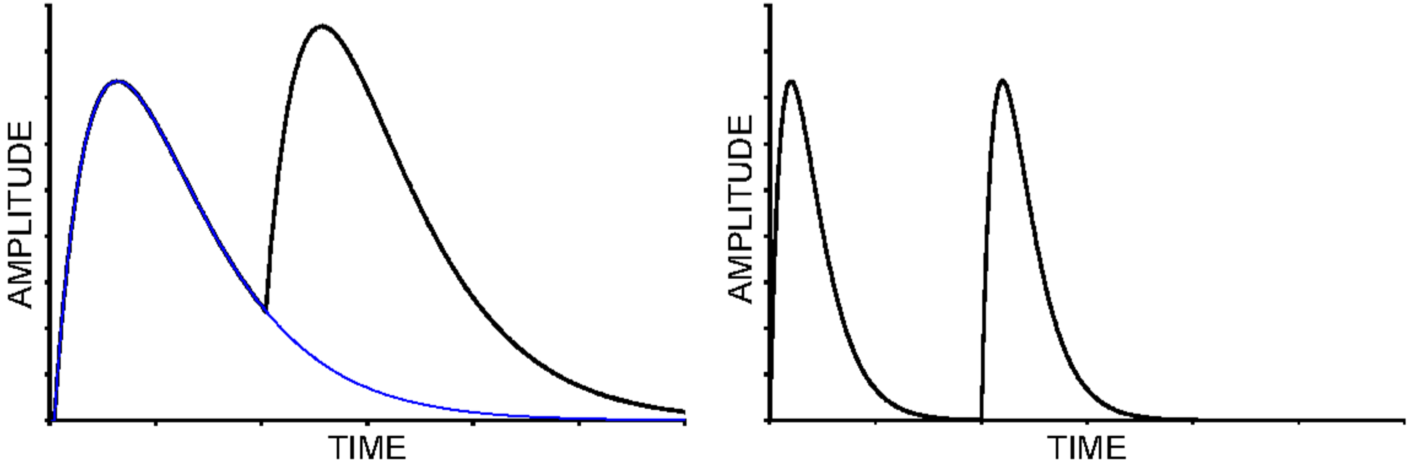
➔Reduce signal bandwidth → improve SNR

- fast rising signals have large bandwidth
- shaper broadens signals



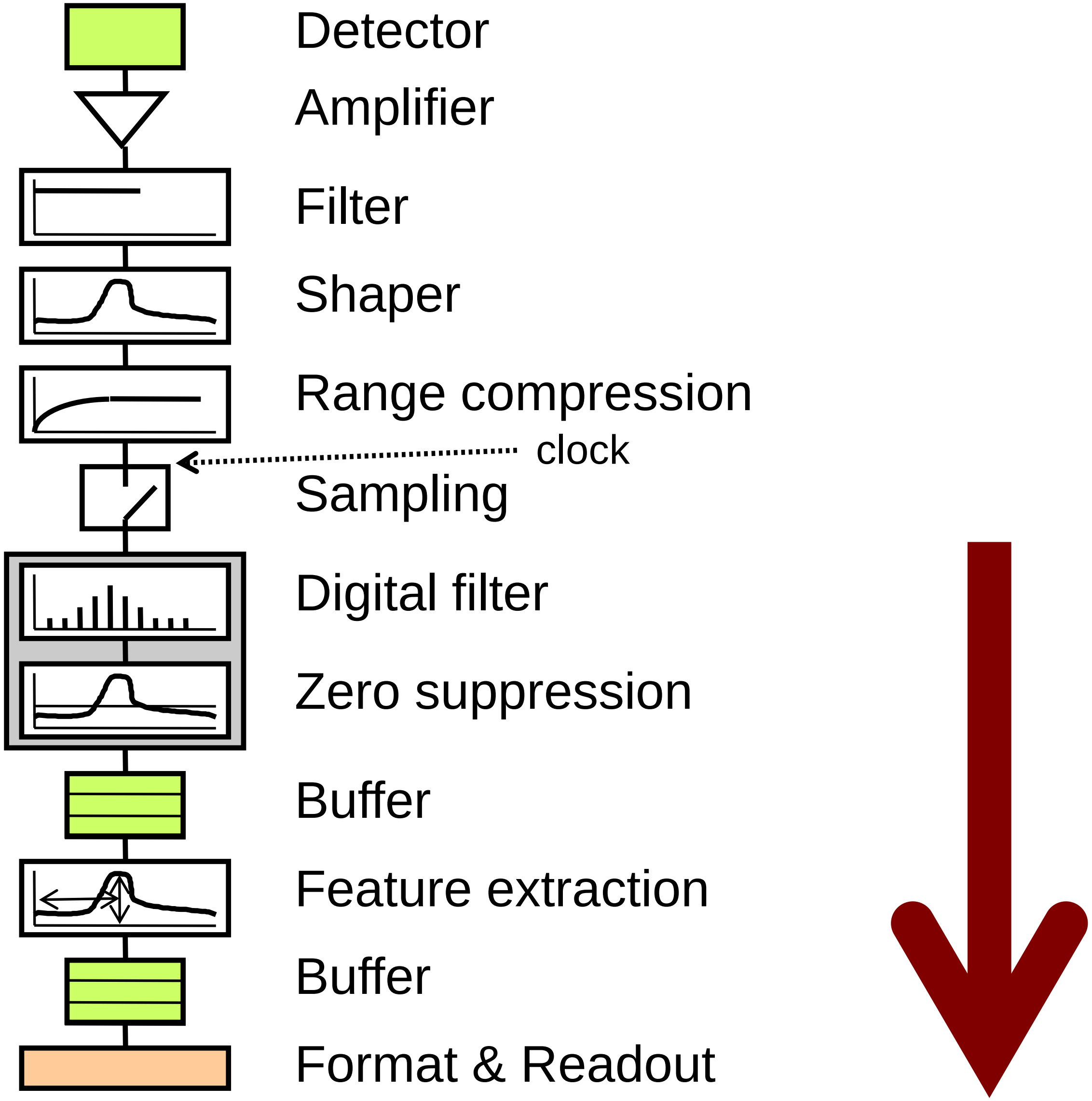
➔Limit pulse width → avoid overlap of successive pulses

- increase maximum signal rate at the cost of more noise



Shaping is often a compromise

Read-out chain



Detector

Amplifier

Filter

Shaper

Range compression

clock

Sampling

Digital filter

Zero suppression

Buffer

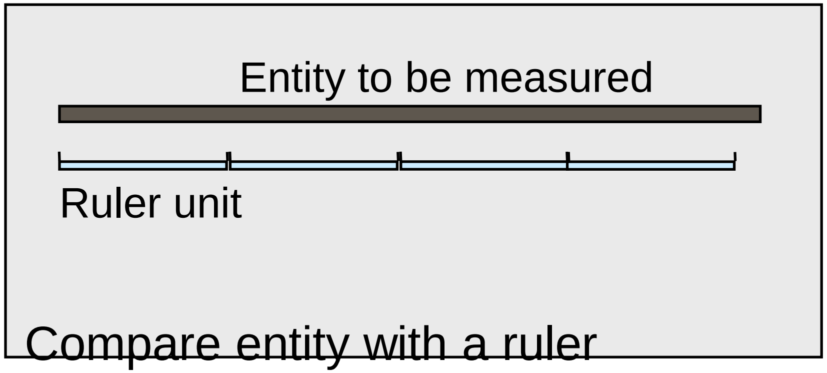
Feature extraction

Buffer

Format & Readout

Analog to digital conversion: introduction

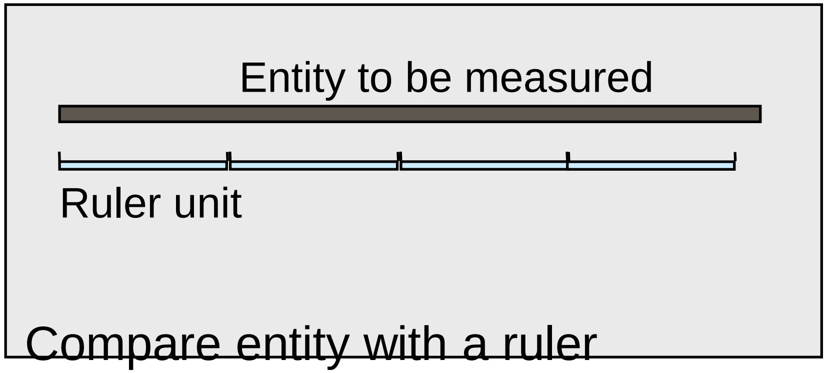
➔Digitization → Encode a analog value into a binary representation

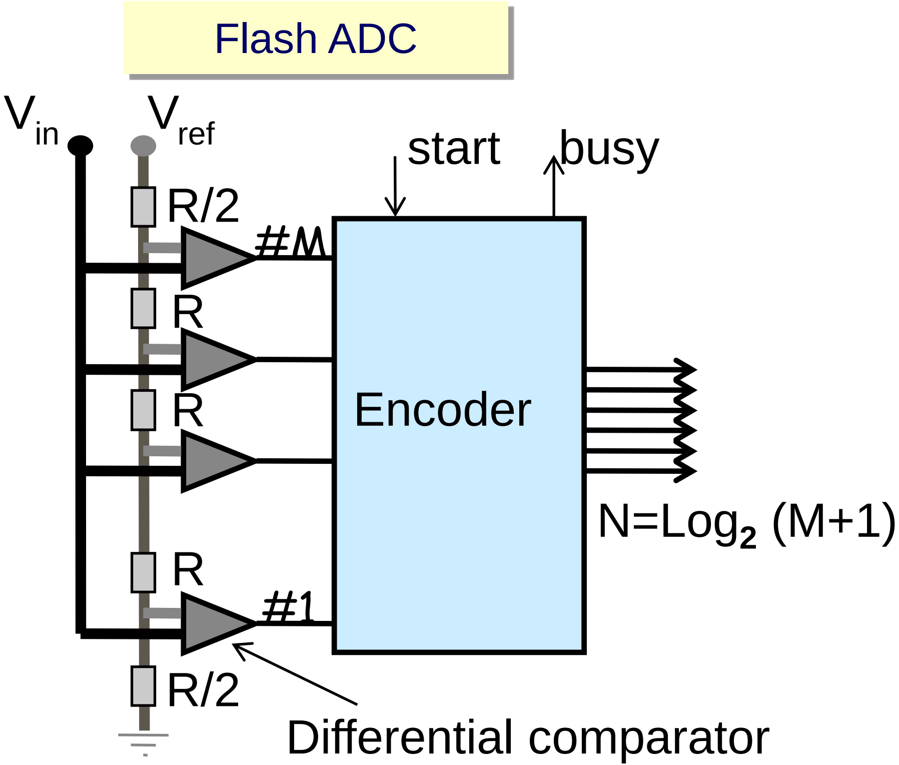


Allow further processing and storage in digital electronics and eventually computers

Analog to digital conversion: Flash ADC

➔ Digitization → Encode a analog value into a binary representation





➔ Flash ADC simplest and fastest implementation

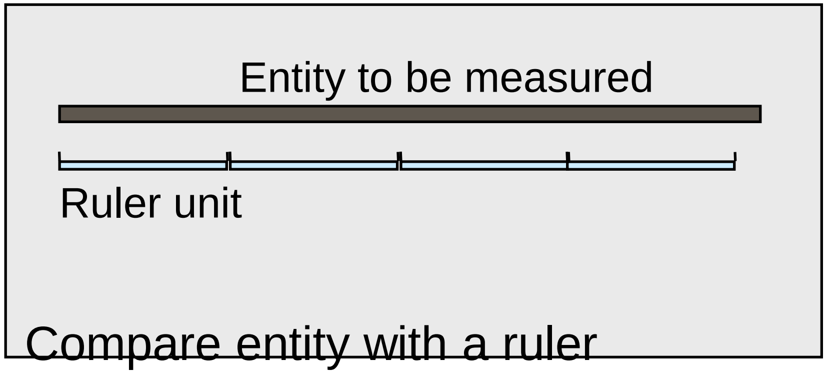
➔ Performs M comparisons in parallel

– input voltage is compared with M fractions of a reference voltage: $V_{ref/}/(2M)$ → $(M-1/2)V_{ref}/M$

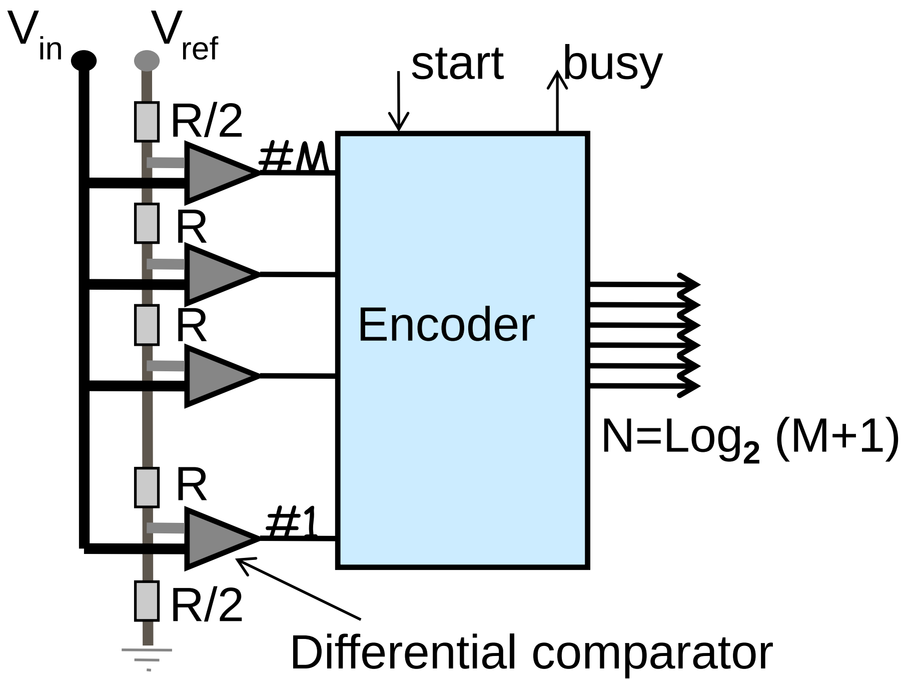
➔ Result is encoded into a compact binary form of N bits

Analog to digital conversion: Flash ADC

➔Digitization → Encode a analog value into a binary representation



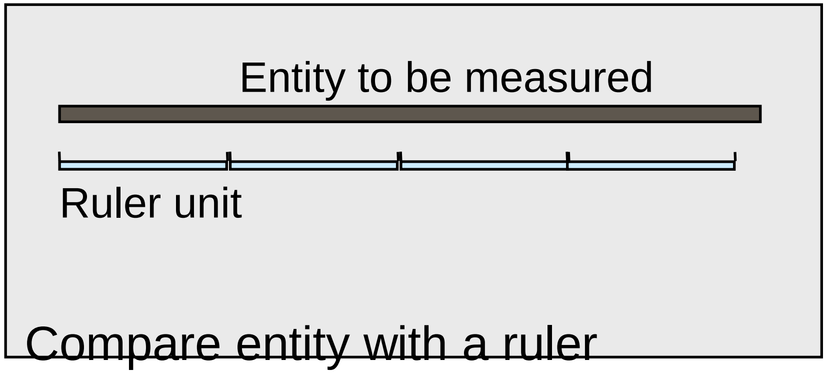
Flash ADC



➔Example M=3 → N=2

ADC Characteristics

- ➔Digitization → Encode a analog value into a binary representation

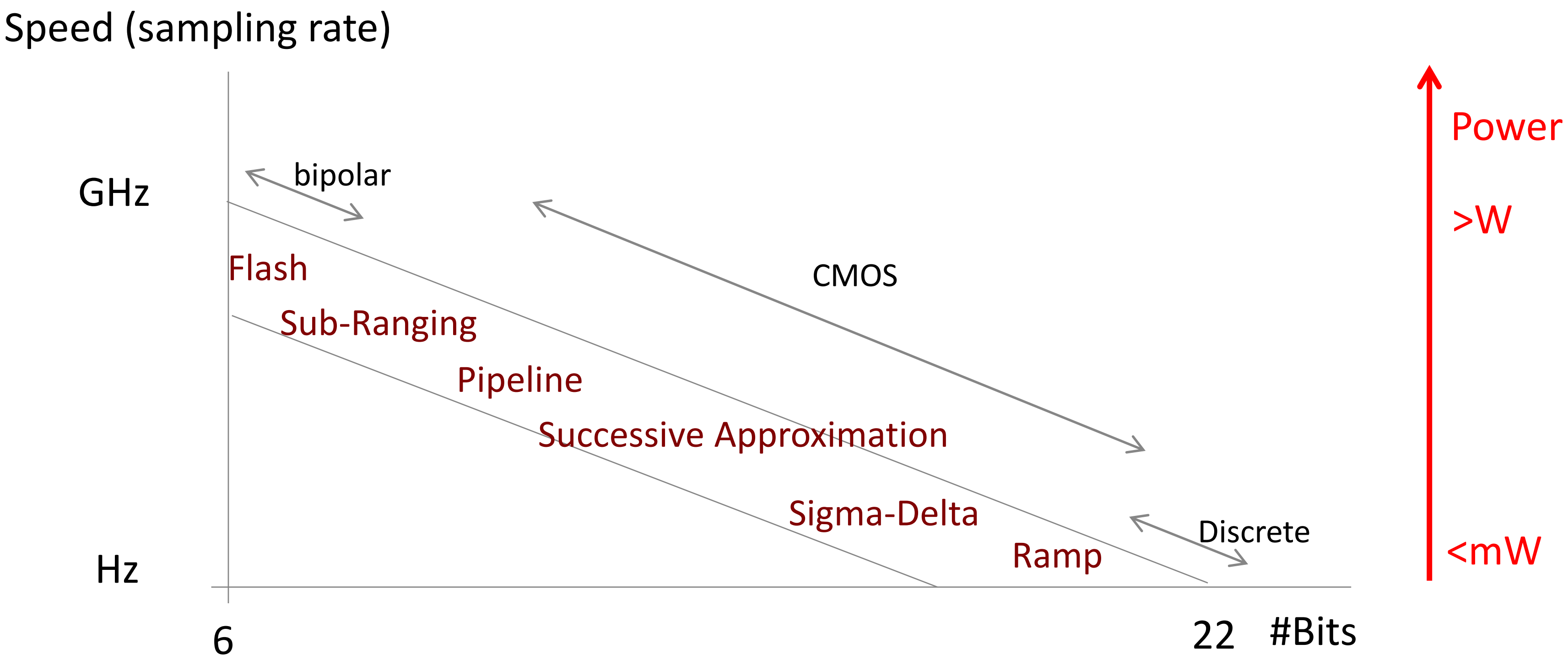


Compare entity with a ruler

- ➔Resolution (LSB), the ruler unit: $V_{max}/2^N$
  - 8bit, 1V → LSB=3.9mV
- ➔Quantization error, because of finite size of the ruler unit: □LSB/2
- ➔Dynamic range: V_{max}/LSB
  - N for linear ADC
  - >N for non-linear ADC
    - constant relative resolution on the valid input range

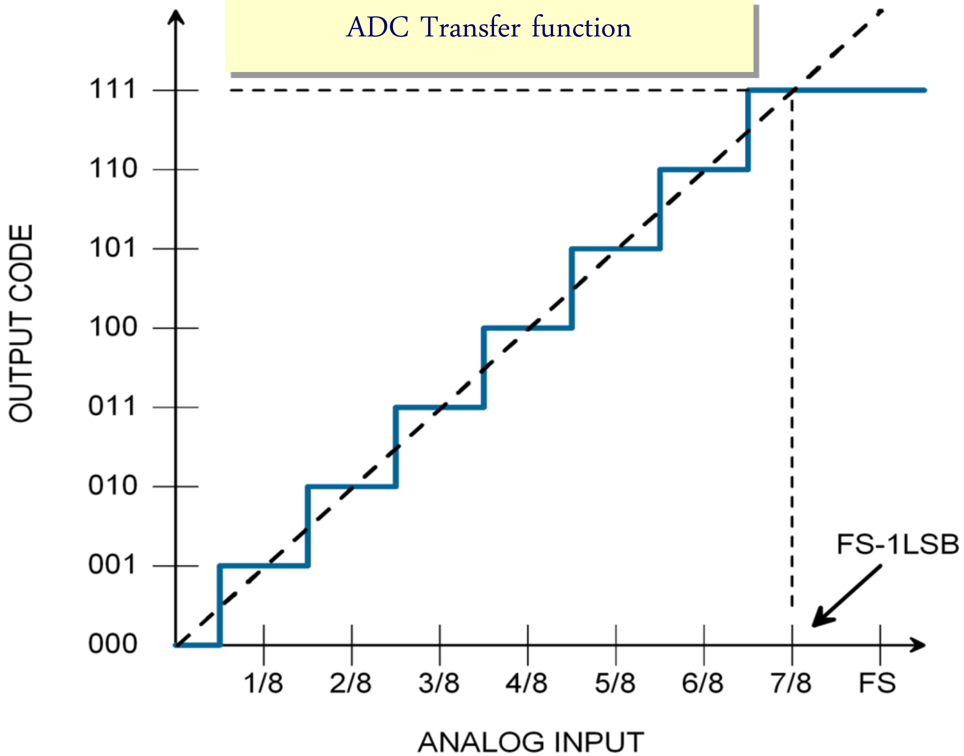
ADC phase-space

Many different ADC technique exists, mostly because of the trade-off between speed, resolution and power consumption (and cost)



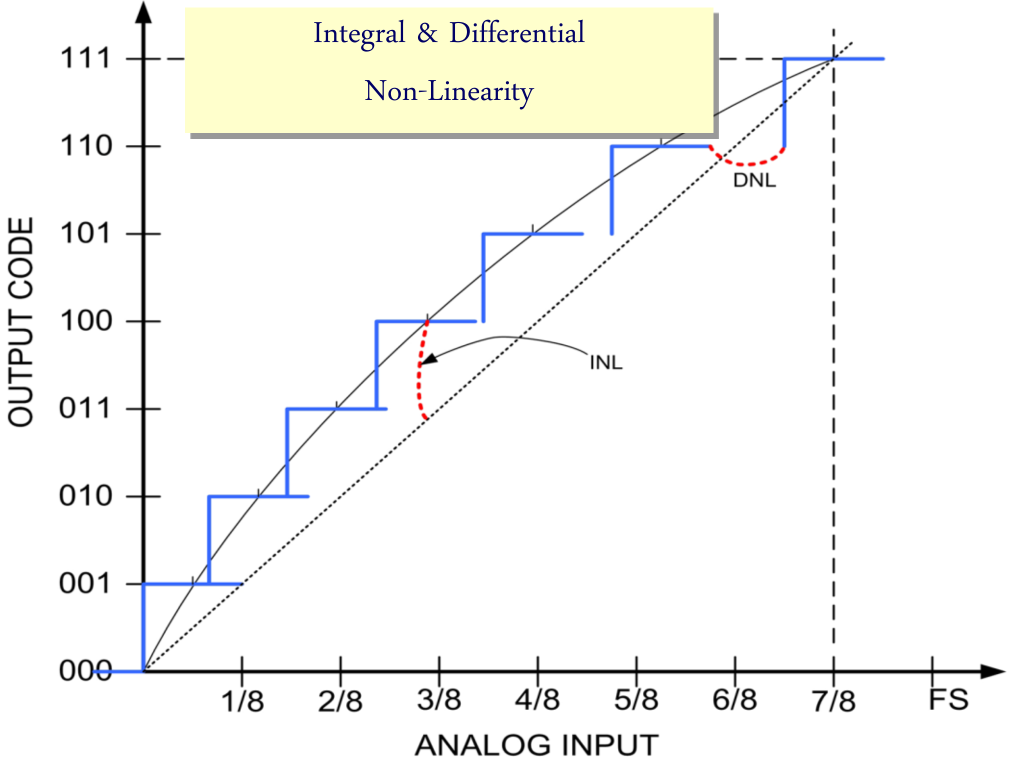
ADC (In)Accuracies

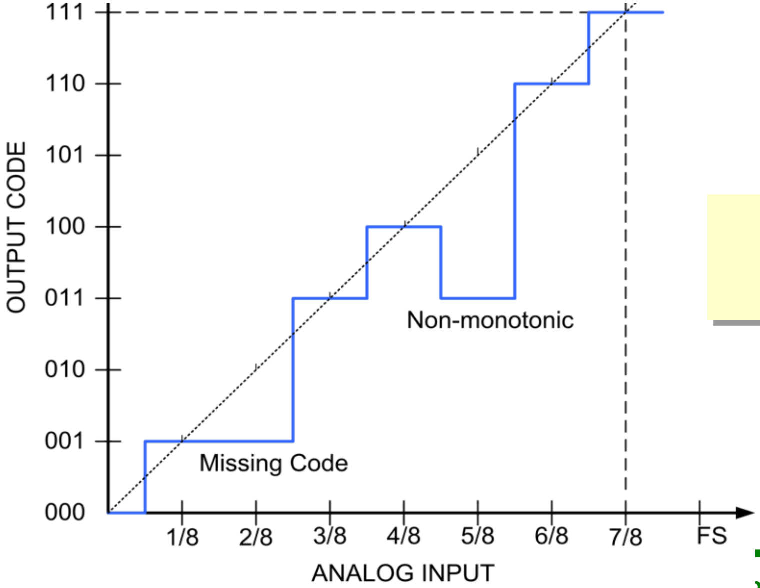
ADC Transfer function



Integral & Differential

Non-Linearity



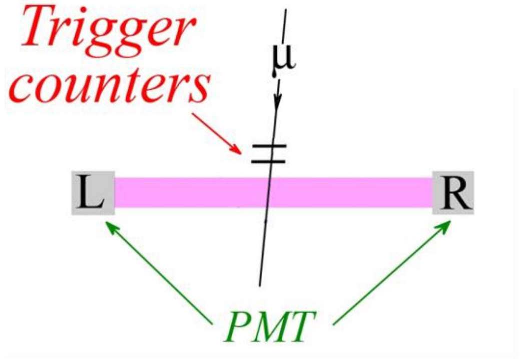


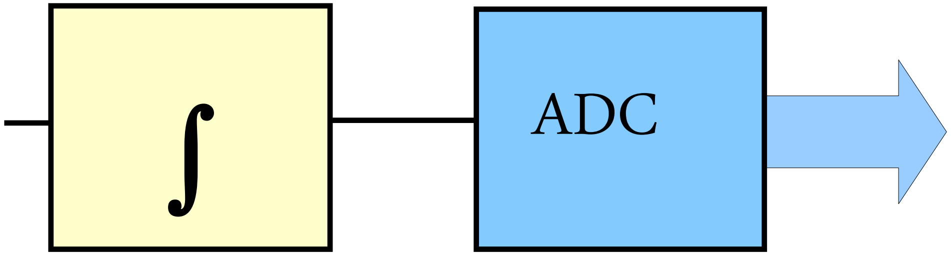
Extreme Differential

Non-Linearity

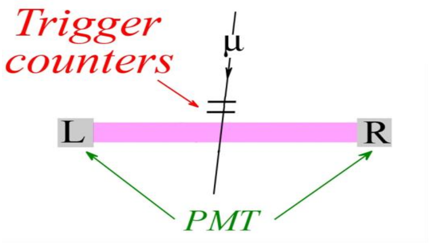
Real ADC at work

- ➔ Real data from a beam test @CERN
- ➔ $PbWO_4$ (scintillating) crystal equipped with two PMTs and exposed to e,□ and □ beams
- ➔ QDC ⟶ charge integrator followed by ADC

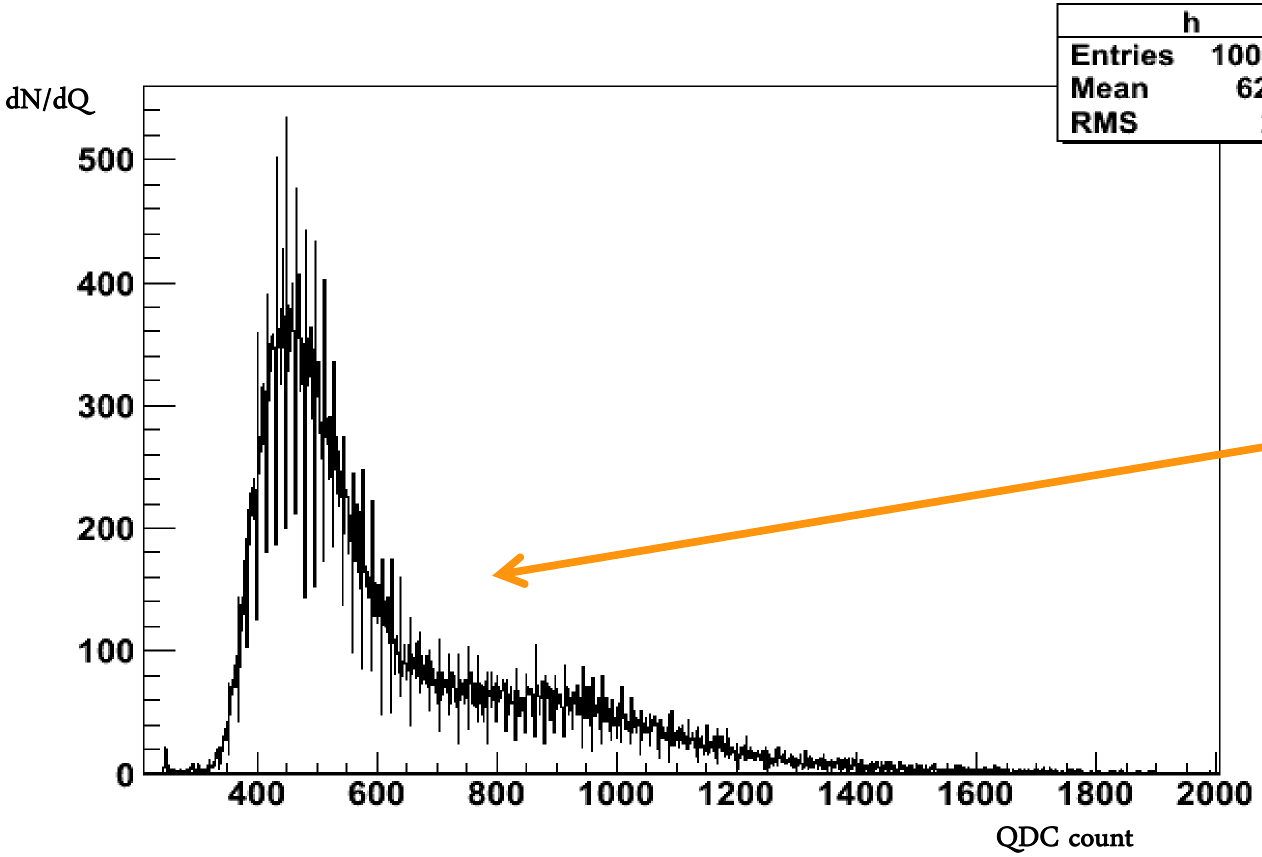




Real QDCs at work

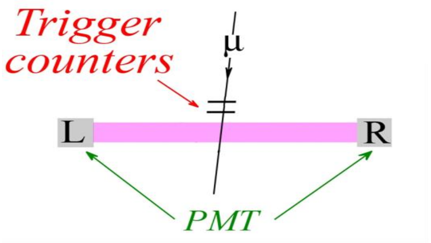


- Real data from a beam test @CERN
- $PbWO_4$ (scintillating) crystal equipped with two PMTs and exposed to e,□ and □ beams

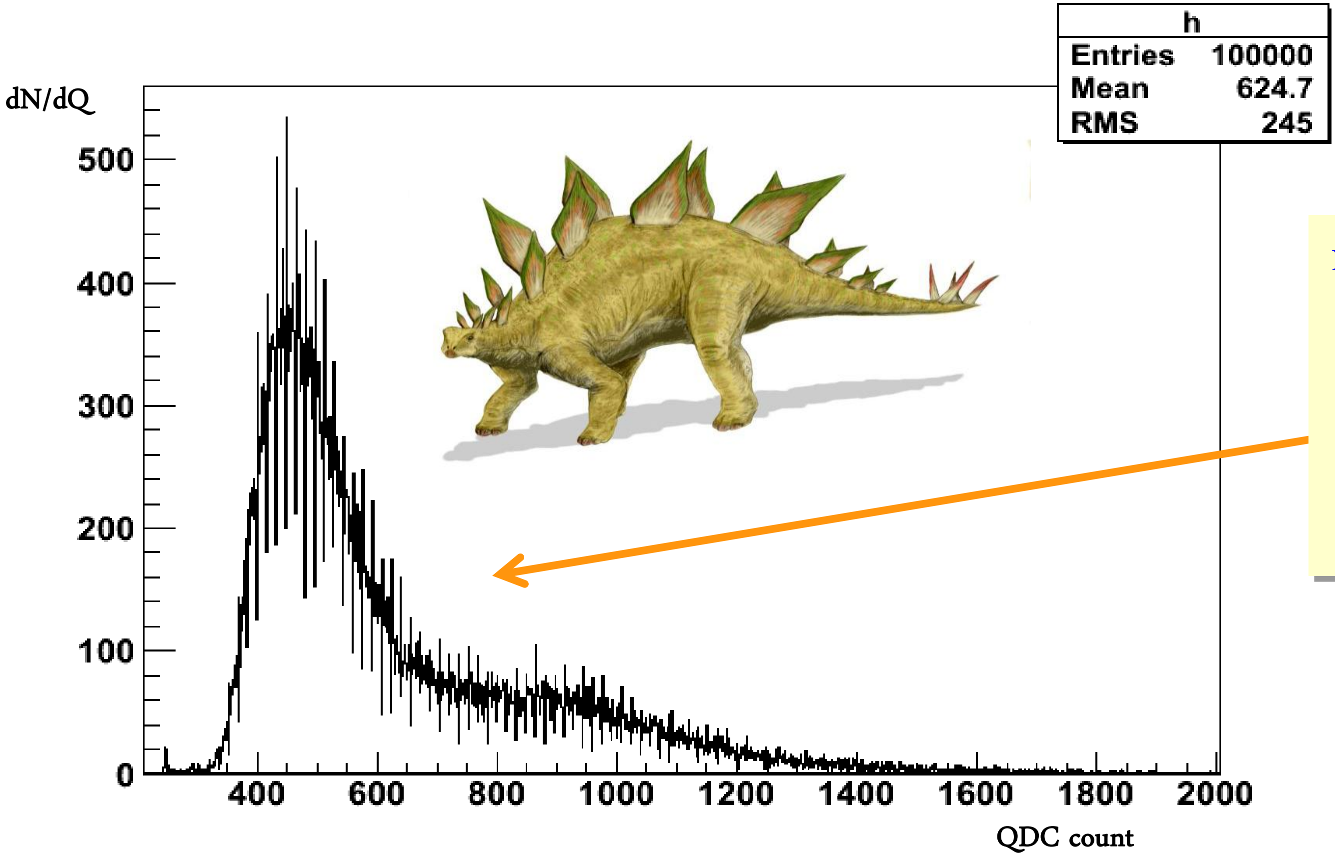


Nice pion-beam charge-distribution for one PMT

Real QDCs at work



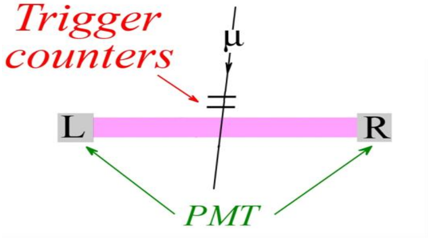
- Real data from a beam test @CERN
- $PbWO_4$ (scintillating) crystal equipped with two PMTs and exposed to e,□ and □ beams

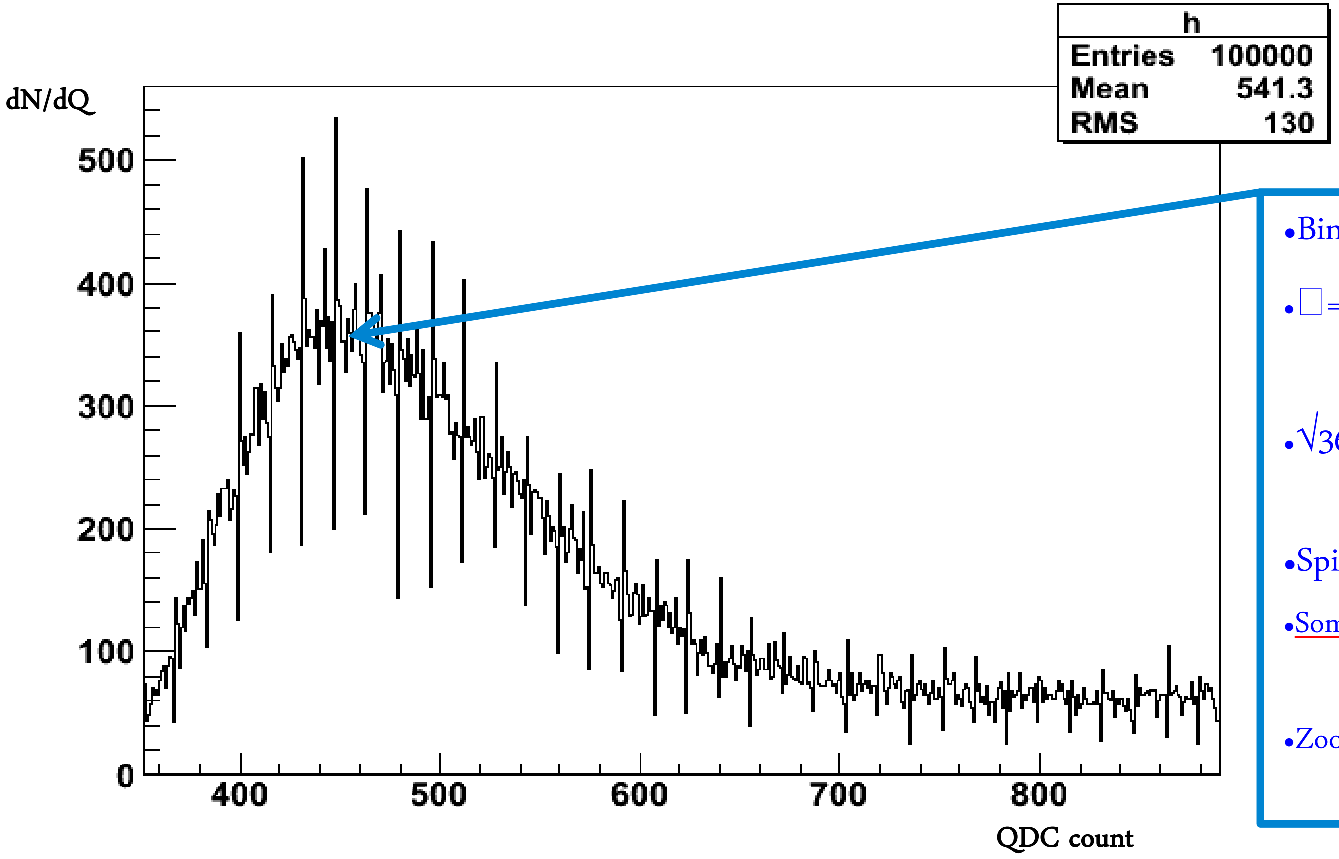


Nice pion-beam charge-distribution for one PMT

But, what are all those little peaks? Just statistical fluctuations? Let's zoom in!

Real QDCs at work

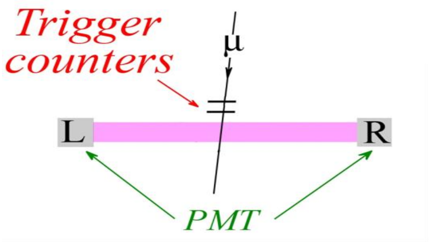


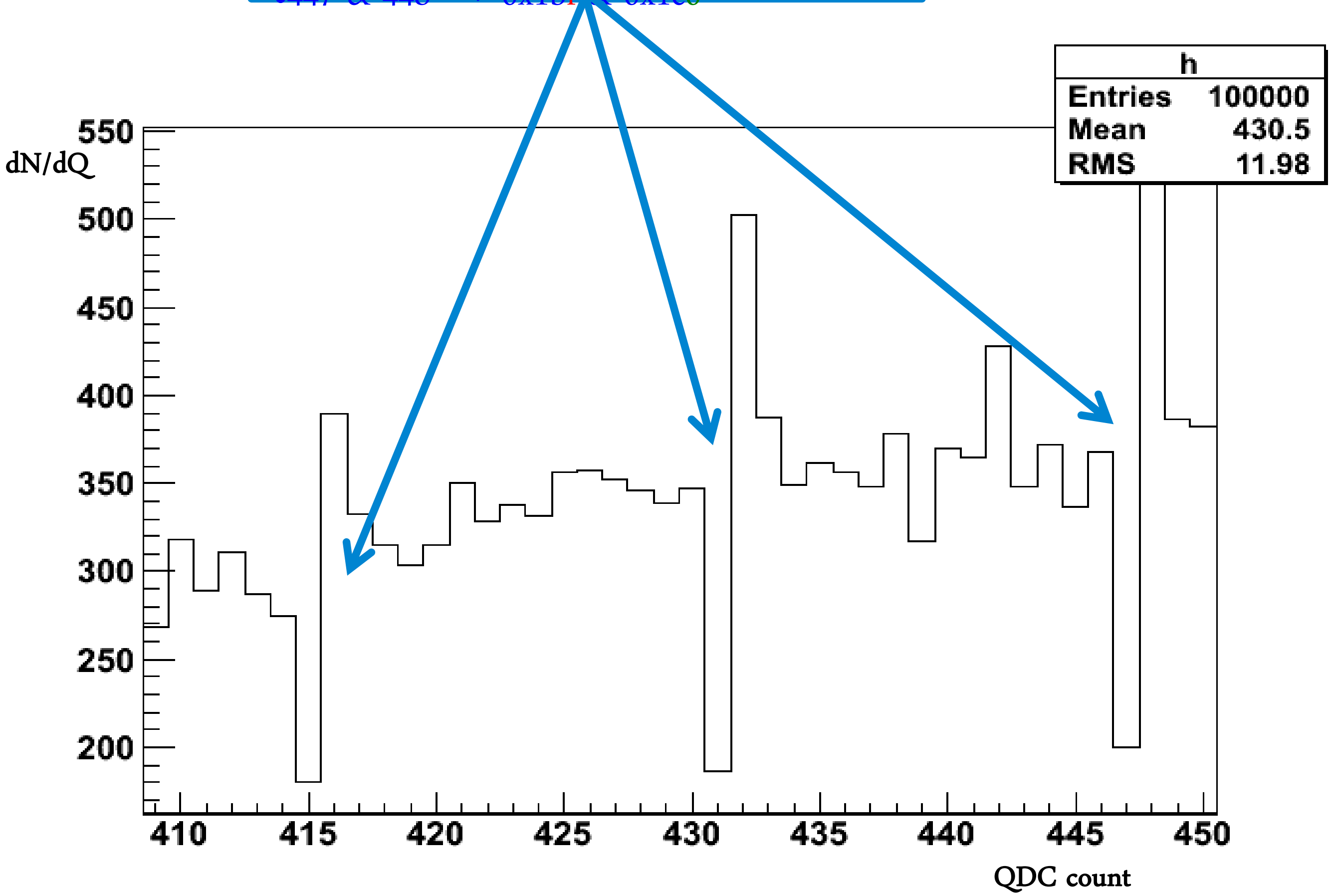


- Bin with N entries shall fluctuate with
- □=$\sqrt{N}$
- $\sqrt{360}$~19 ⟶ ()~10□
- Spikes are regularly distributed
- Some systematic effect must be taking place
- Zoom in a bit more!

Real QDCs at work

- 415 & 416 ⟶ 0x19f & 0x1a0
- 431 & 432 ⟶ 0x1af & 0x1b0
- 447 & 448 ⟶ 0x1bf & 0x1c0





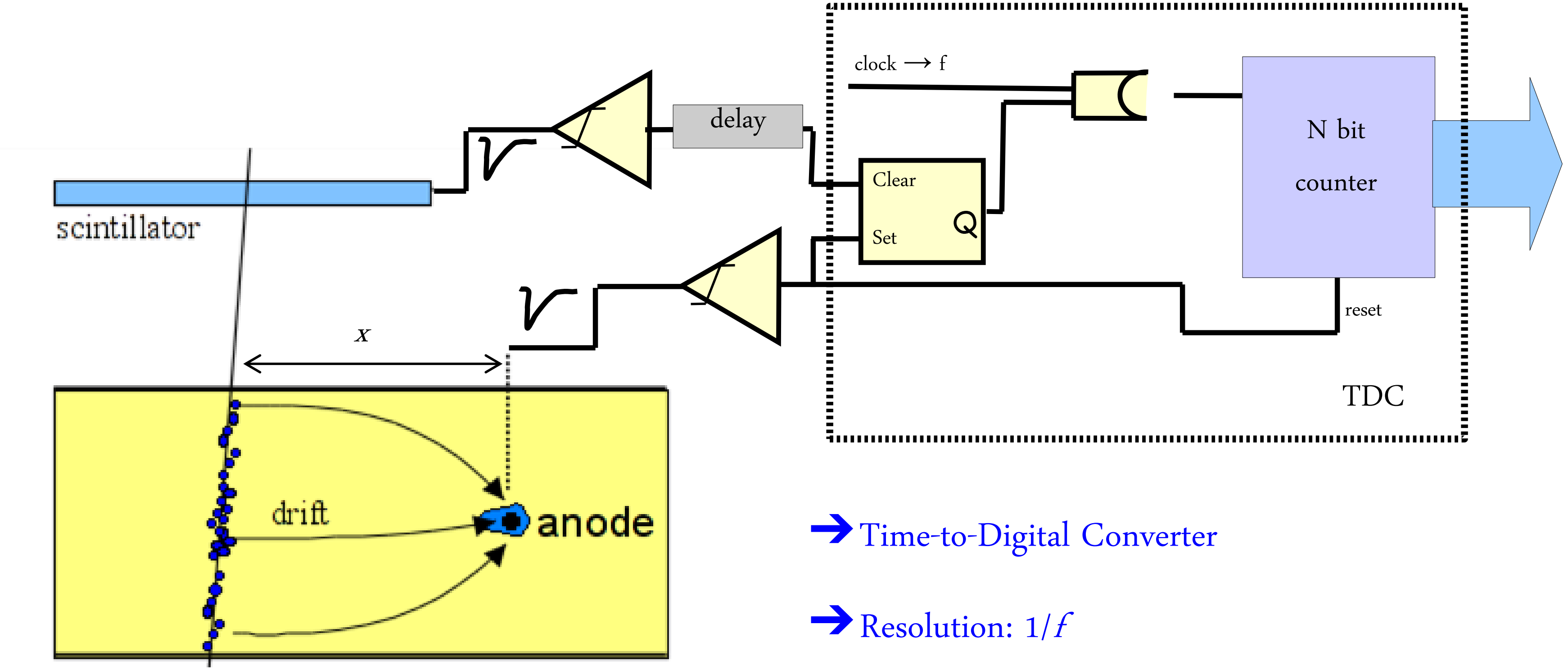
Can you see the effect?

The QDC prefers output configurations of type *xxx*0 in respect of those like *xxxf*

Typical differential non-linearity of successive approximation ADCs

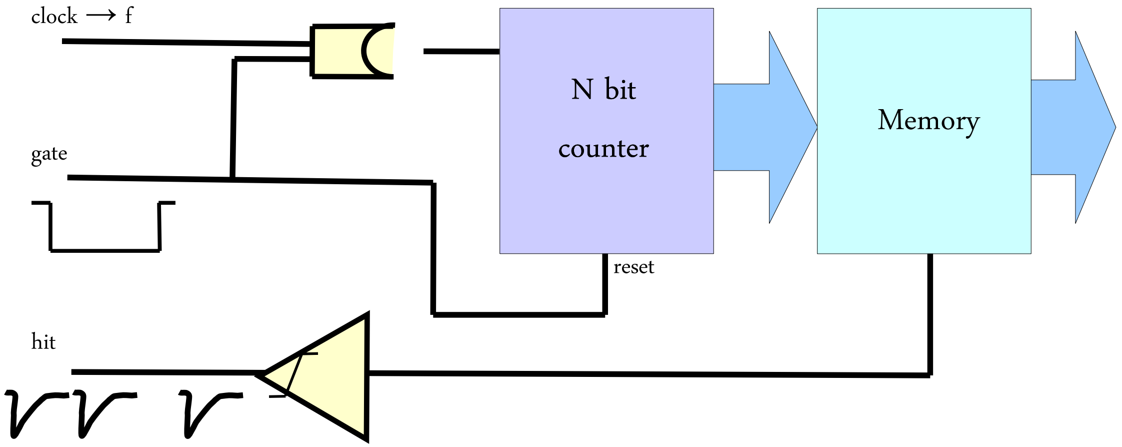
Homework: which is the simplest way to fix this problem in the data? At which cost?

Time measurement → TDC



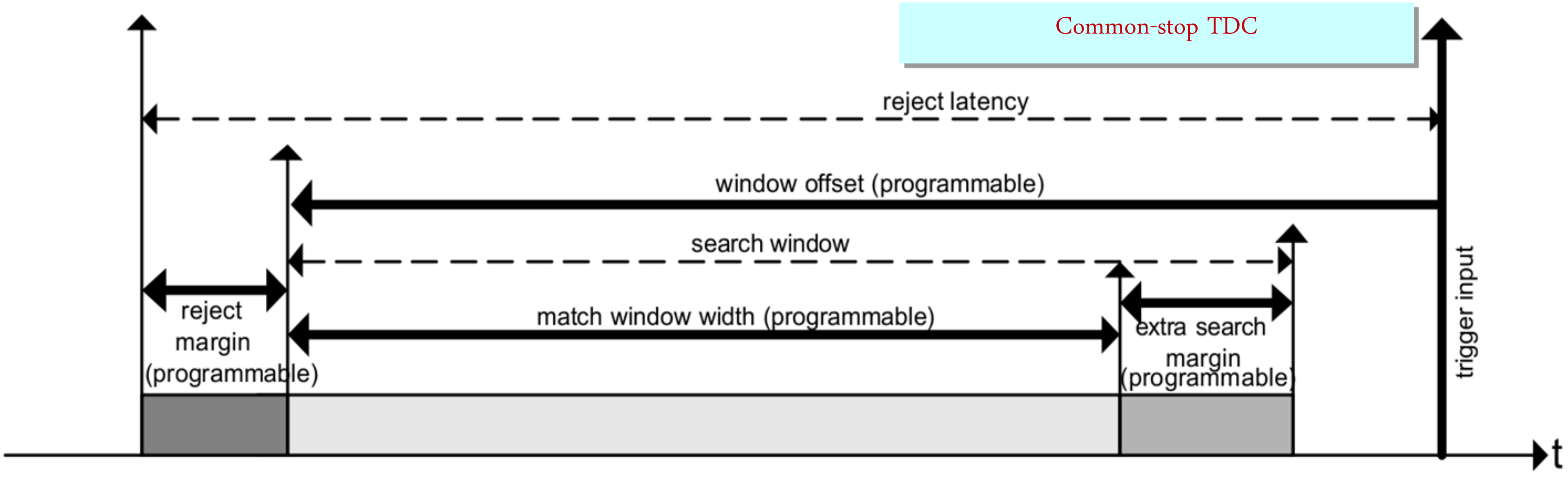
- Time-to-Digital Converter
- Resolution: $1/f$
- Dynamic range: N
- Single hit TDC
  - e.g. a noise spikes comes just before the signal → measure is lost

Multi-hit TDC



- Gate resets and starts the counter. It also provides the measurement period. It must be smaller than $2^N/f$
- Each “hit” (i.e. signal) forces a memory (FIFO) to load the current value of the counter, that is the delay after the gate start
  - in order to distinguish between hits belonging to different gates, some additional logic is need to tag the data
- Common-start configuration

Real TDCs


Common-stop TDC
reject latency
window offset (programmable)
search window
reject margin (programmable)
match window width (programmable)
extra search margin (programmable)
trigger input
t

→ Real TDCs provide advanced functionalities for fine-tuning the hit-trigger matching

- internal programmable delays
- internal generation of programmable gates
- programmable rejection frames

Calibration

Calibration

➔ Often our experiments provide relative measurements. The values obtained via our system are in some (known) relation with the interesting quantity

- due to physical detection mechanism
- due to signal processing

$$E \propto Q_s = \int i_s(t)dt$$

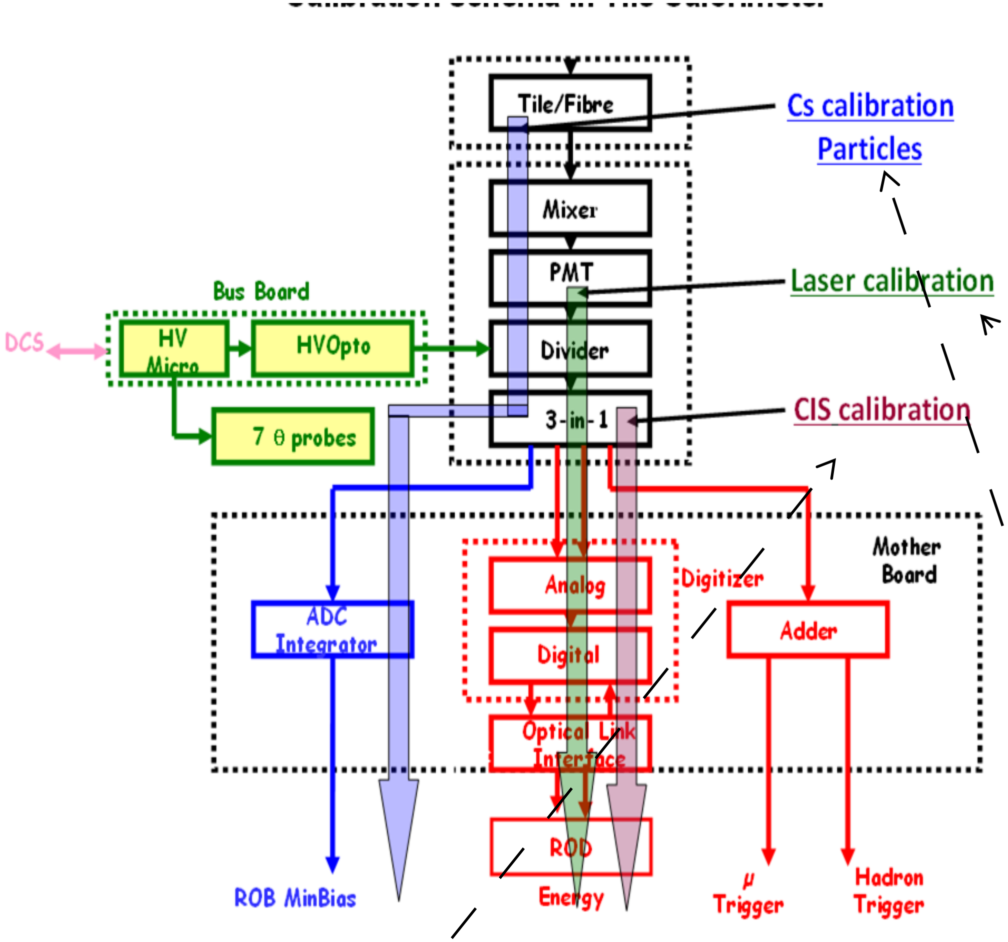
➔ Detectors need to be calibrated in order to give us the answer we are looking for

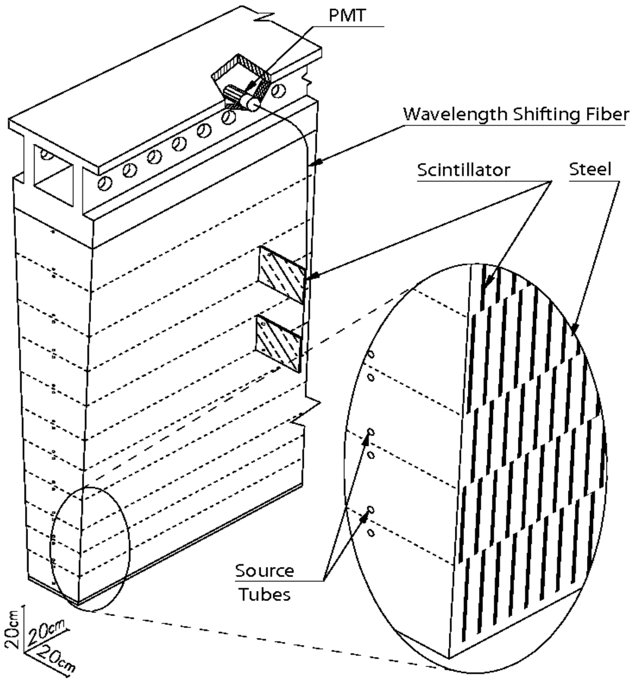
- determine the parameters that transform the raw data into a physics quantities
- normally depend on the experimental setup (e.g. cable length, delay settings, HV settings, ...)
- parameters might change with aging (radiation) and beam conditions

➔ The design of our detector and DAQ have to foreseen calibration mechanisms/procedures

- injection of known signals
- dedicated calibration *triggers* and data streams

ATLAS Tile Calorimeter Calibration





PMT non-linearities

Laser System

ADC count to charge

Charge Injection System

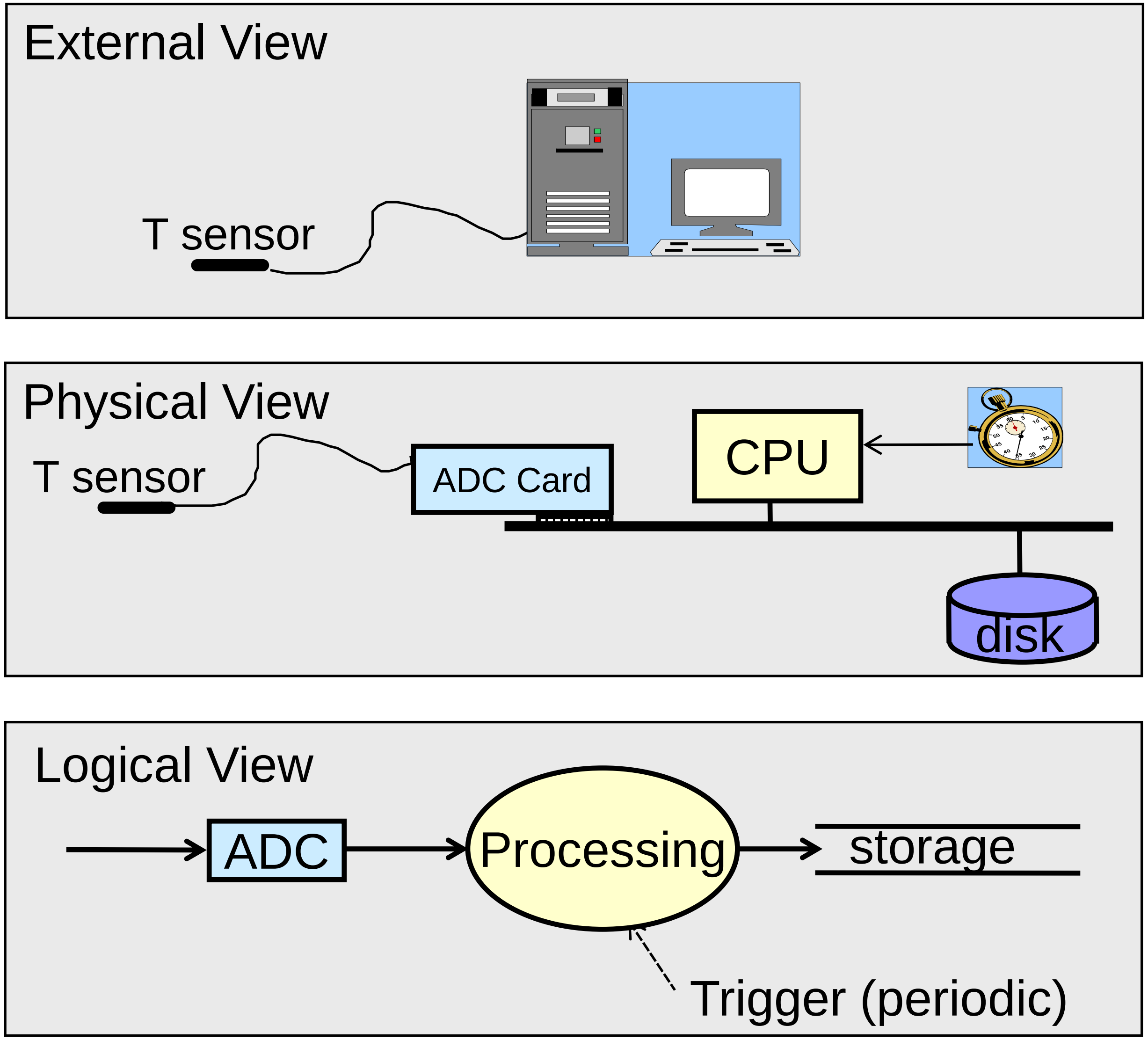
Detector non-uniformities

Cesium System

$$E_{channel} = A \cdot C_{ADC\rightarrow pC} \cdot C_{pC\rightarrow GeV} \cdot C_{Cs} \cdot C_{laser}$$

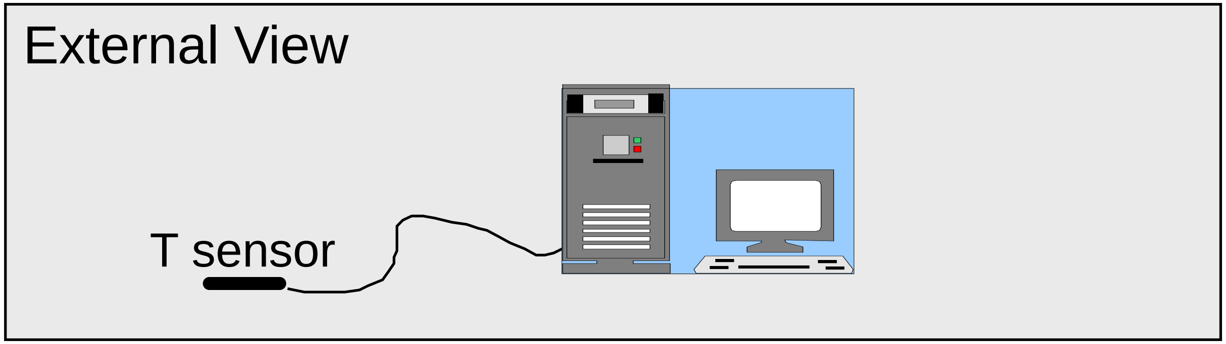
DAQ & Trigger

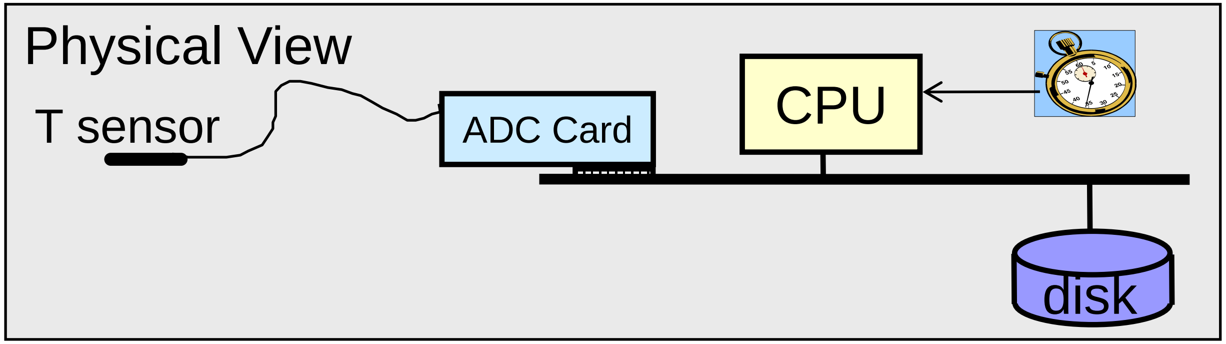
Basic DAQ: periodic trigger

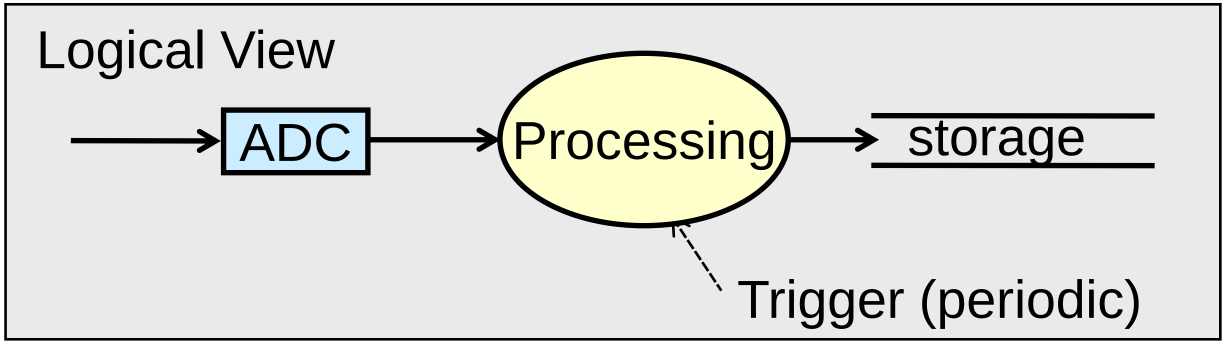


- ➔ Measure temperature at a fixed frequency
- ➔ ADC performs analog to digital conversion
  - our front-end electronics
- ➔ CPU does readout and processing

Basic DAQ: periodic trigger







- Measure temperature at a fixed frequency
- The system is clearly limited by the time to process an “event”
- Example □=1ms to
  - ADC conversion +CPU processing +Storage
- Sustain ~1/1ms=1kHz ***periodic trigger*** rate

What is a “trigger”?

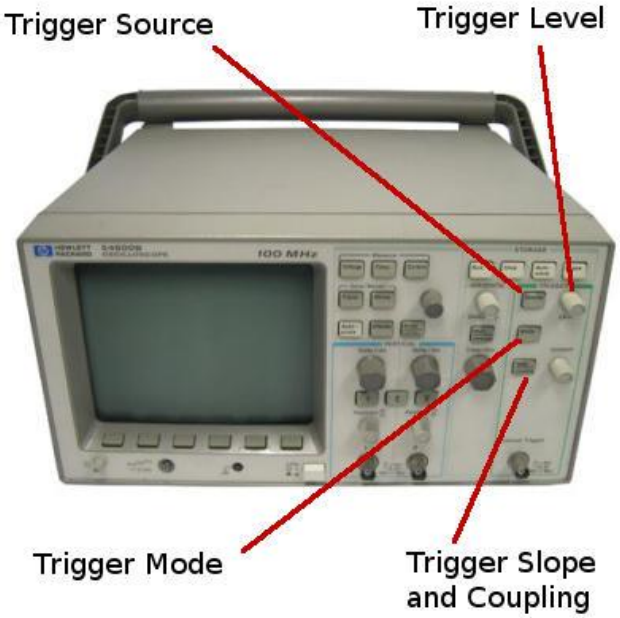
→ A “trigger” is a system that rapidly decides, based on simple criteria, if an interesting event took place, initiating the data-acquisition process

→ Simple, rapid, selective are the trigger keywords

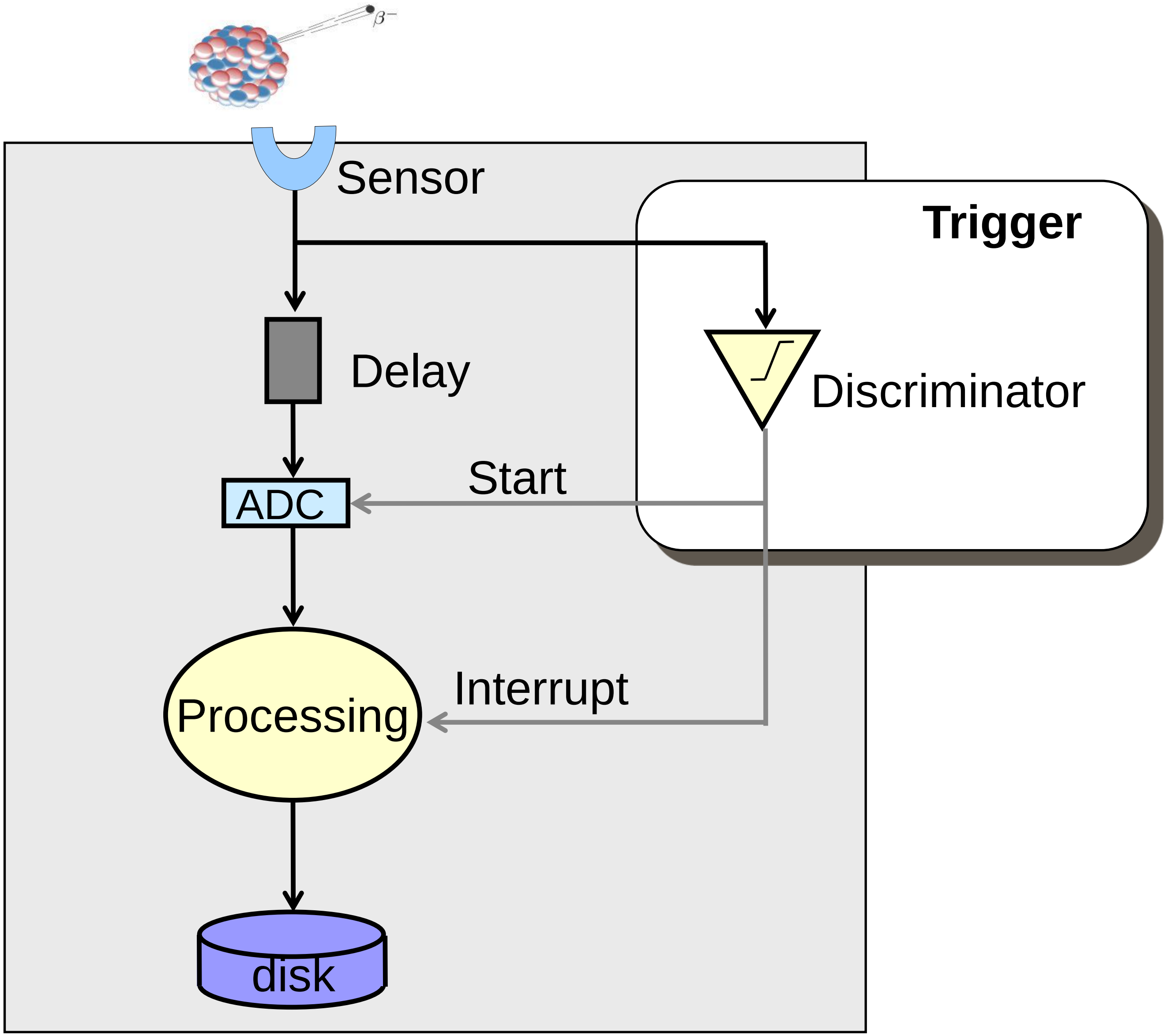
→ Relative parameters that depend on the operating conditions

- in a multi-level trigger system the last level is normally way slower and more complex than the first one

The oscilloscope trigger does exactly this. Informs the instrument to initiate the internal signal acquisition and visualization

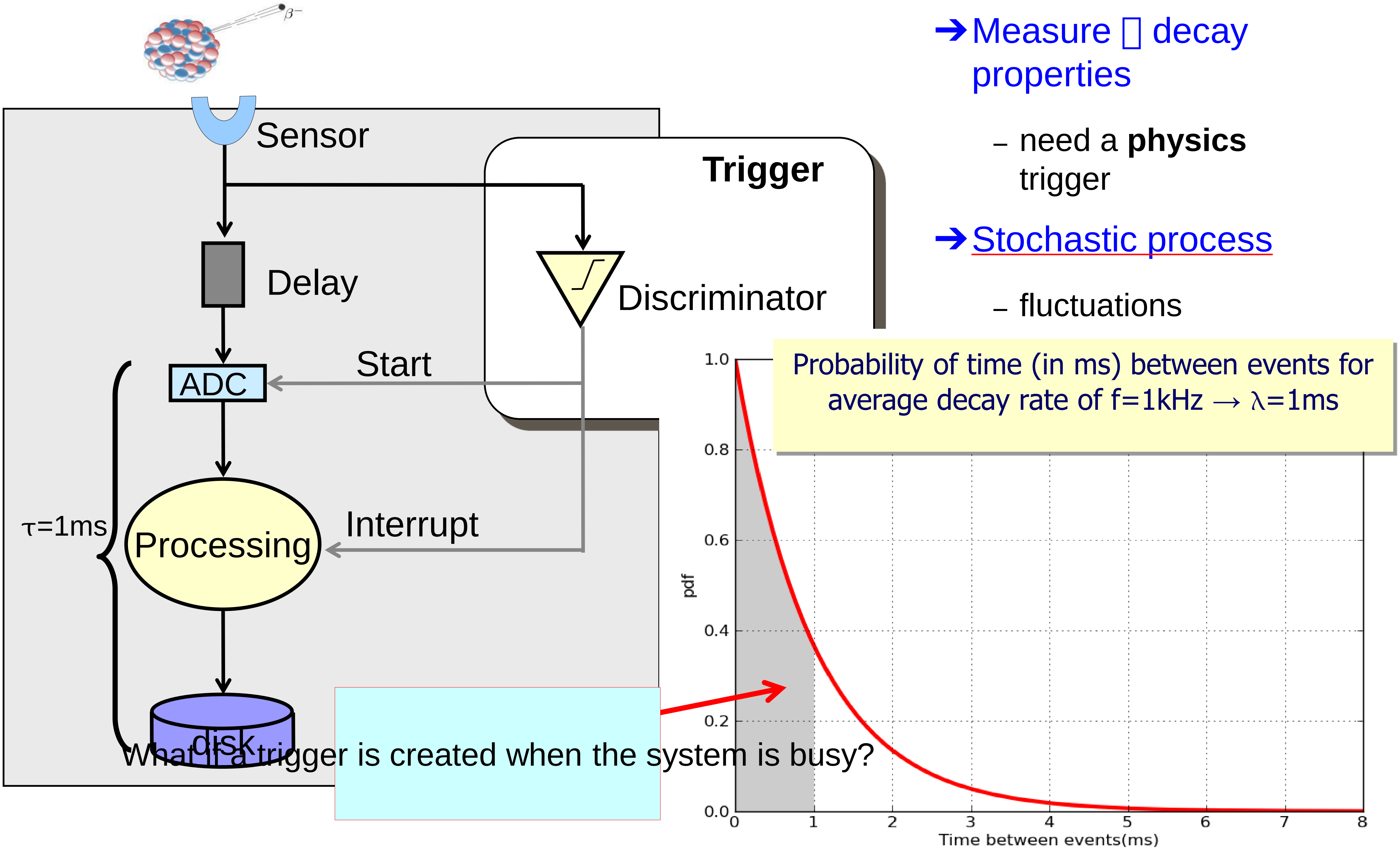


Basic DAQ: real trigger



- Measure □ decay properties
- Events are asynchronous and unpredictable
  - need a **physics** trigger
- Delay compensates for the trigger latency
  - time needed to reach a decision

Basic DAQ: real trigger

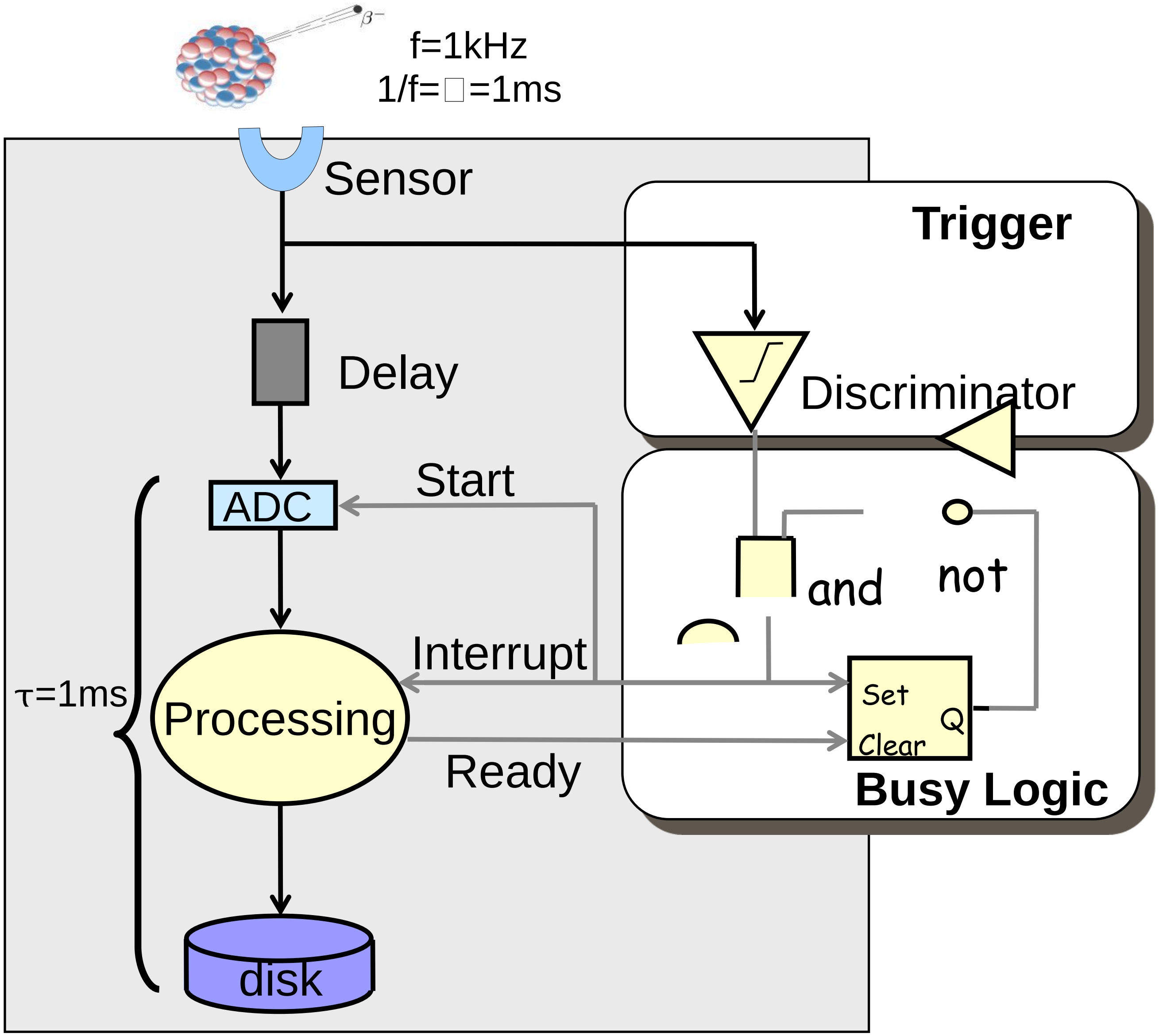


- ➔ Measure □ decay properties
  - need a **physics** trigger
- ➔ Stochastic process
  - fluctuations

Probability of time (in ms) between events for average decay rate of f=1kHz → λ=1ms

What if a trigger is created when the system is busy?

Basic DAQ: real trigger & busy logic



- ➔ Busy logic avoids triggers while processing
- ➔ Which (average) DAQ rate can we achieve now?
  - reminder: □=1ms was sufficient to run at 1kHz with a clock trigger

DAQ Deadtime & Efficiency (1)

Define ν as average DAQ frequency

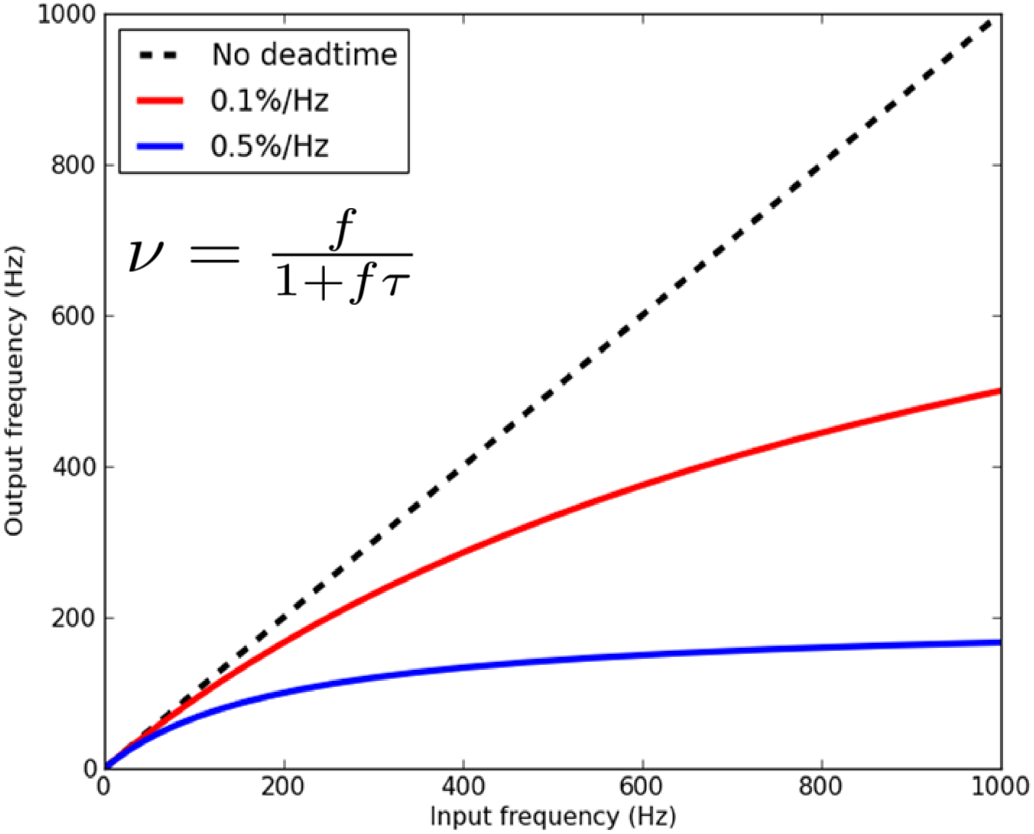
$\nu\tau \rightarrow$ DAQ system is busy - $(1-\nu\tau) \rightarrow$ DAQ system is free

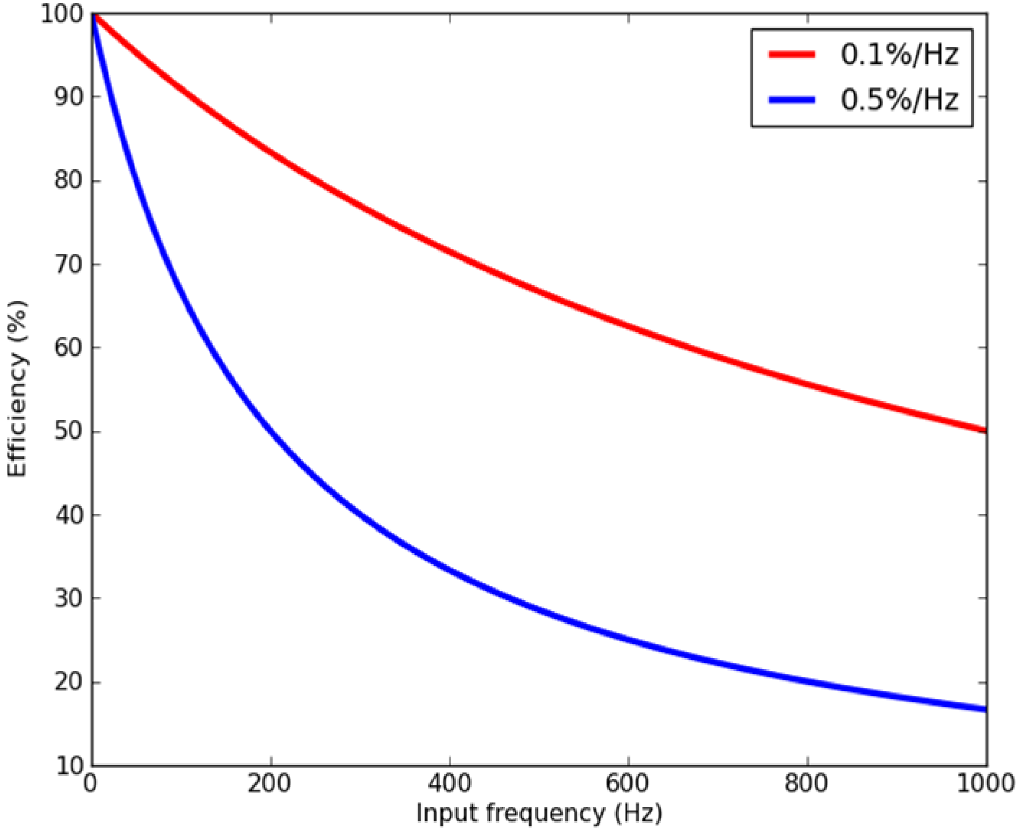
$$f(1-\nu\tau) = \nu \rightarrow \nu = \frac{f}{1+f\tau} < f$$

$$\epsilon = \frac{N_{saved}}{N_{tot}} = \frac{1}{1+f\tau} < 100\%$$

- ➔Define DAQ deadtime (d) as the time the system requires to process an event, without being able to handle other triggers. In our example d=0.1%/Hz
- ➔Due to the fluctuations introduced by the stochastic process the efficiency will always be less 100%
  - – in our specific example, d=0.1%/Hz, f=1kHz → □=500Hz, □=50%

DAQ Deadtime & Efficiency (2)



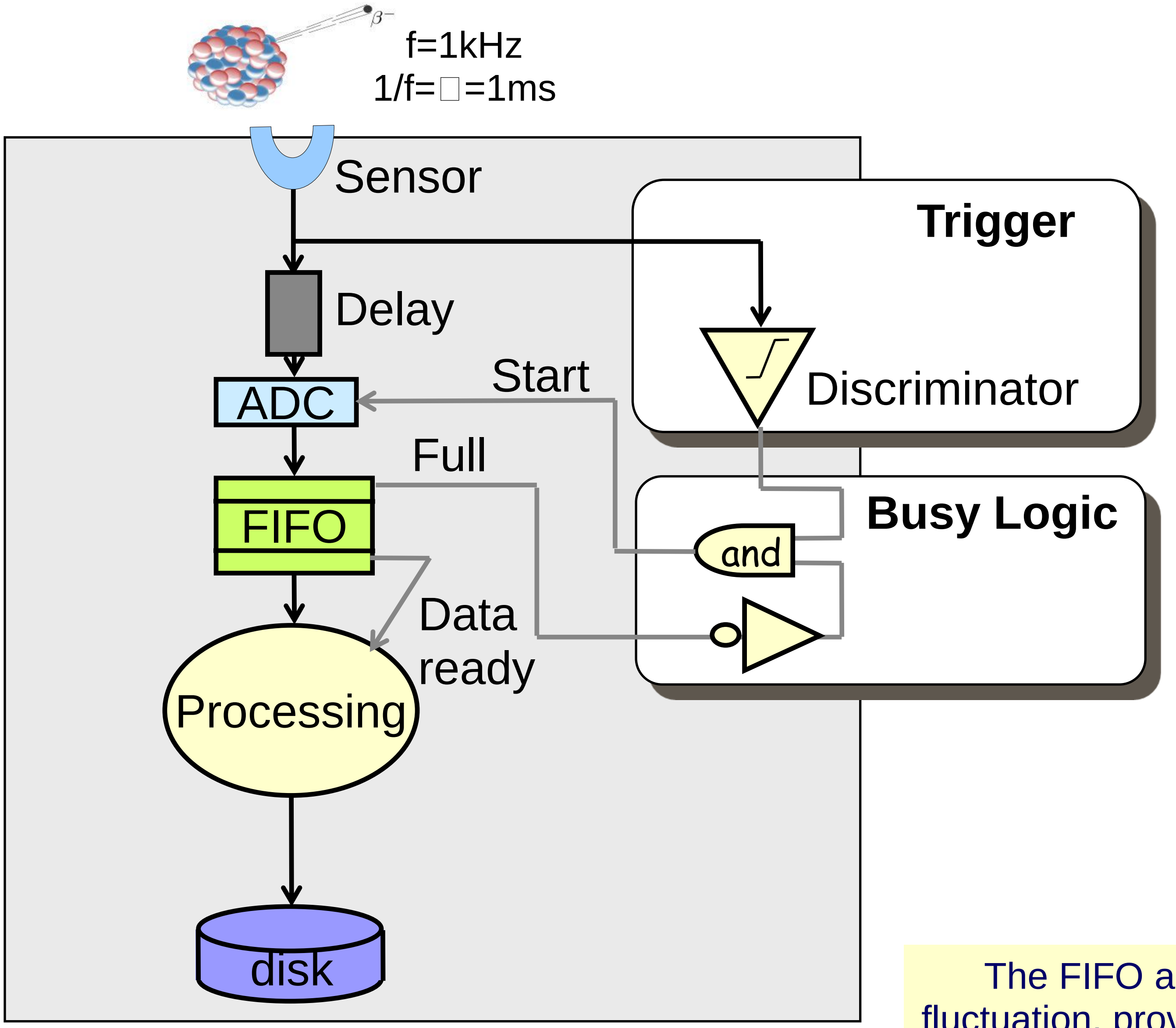


➔If we want to obtain □~f (□~100%) → f□<<1 → □<<□

– f=1kHz, □=99% → □<0.1ms → 1/□>10kHz

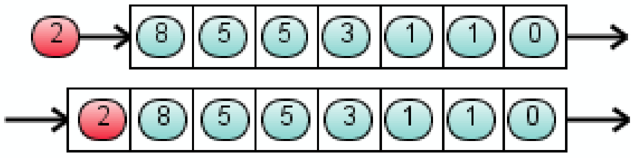
➔In order to cope with the input signal fluctuations, we have to over-design our DAQ system by a factor 10. This is very inconvenient! Can we mitigate this effect?

Basic DAQ: De-randomization



➔First-In First-Out

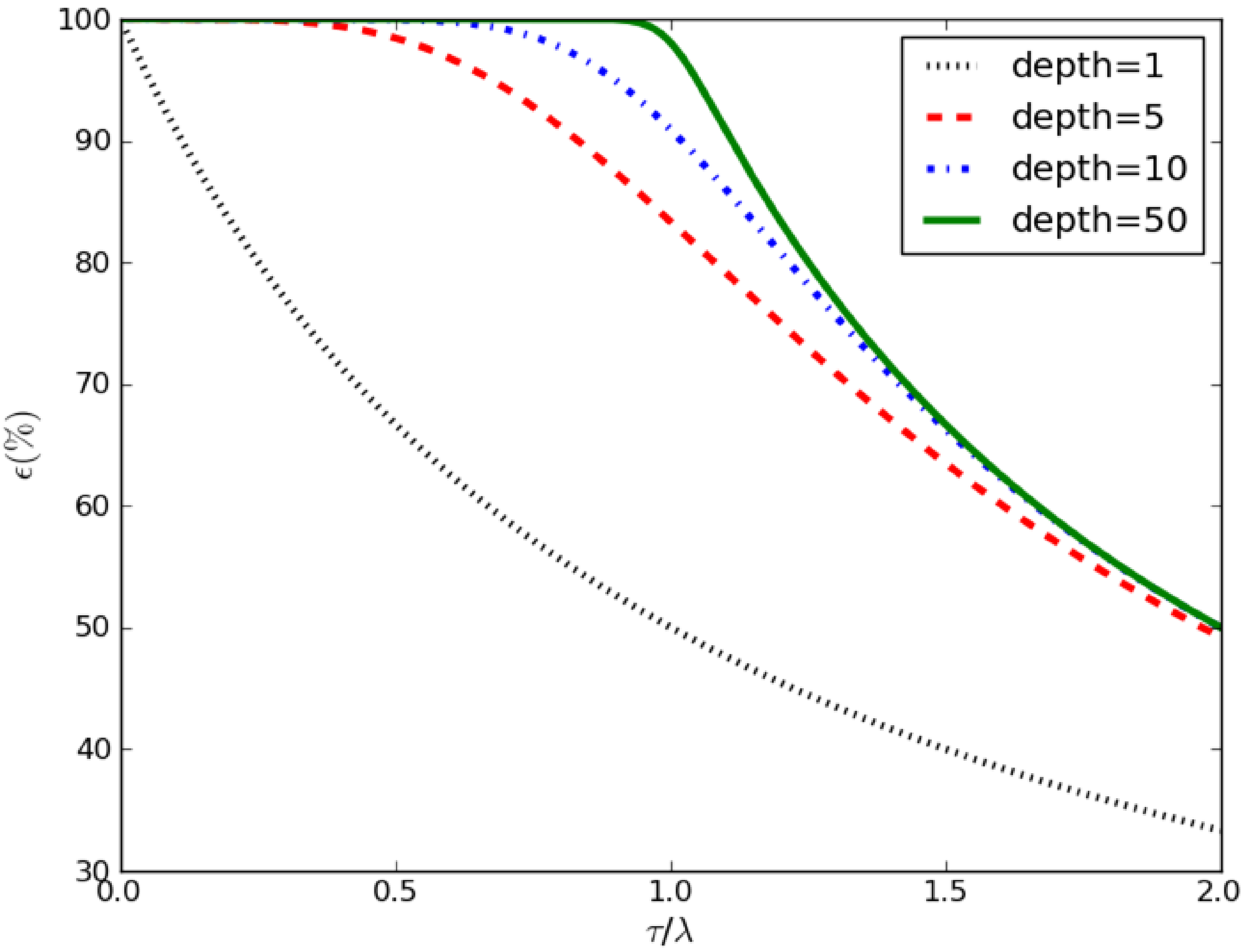
- buffer area organized as a queue
- depth: number of cells
- implemented in HW and SW

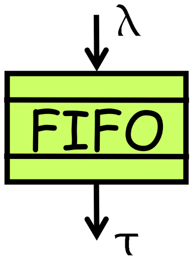


➔FIFO introduces an additional latency on the data path

The FIFO absorbs and smooths the input fluctuation, providing a ~steady (De-randomized) output rate

De-randomization: queuing theory

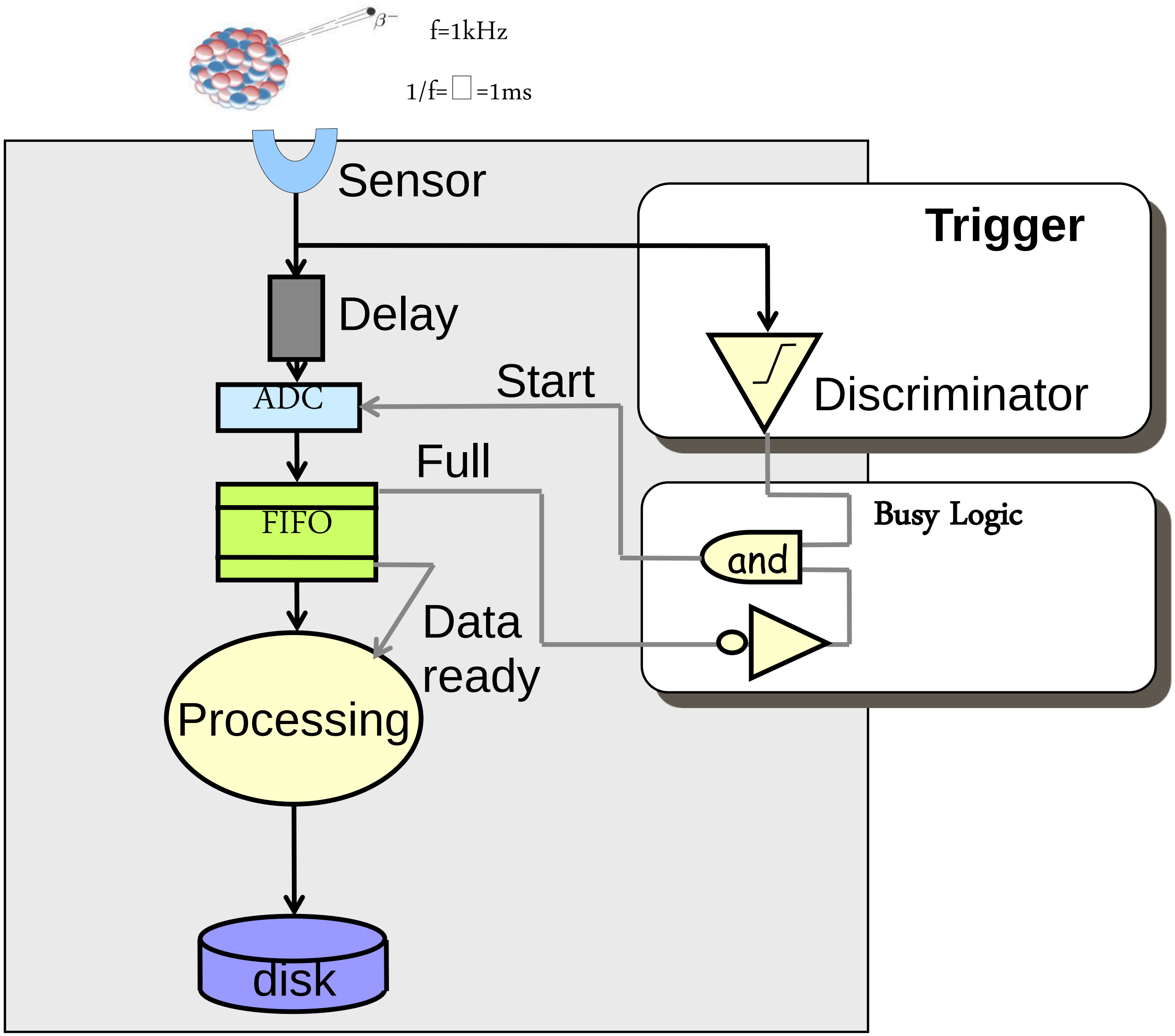




- ➔We can now attain a FIFO efficiency ~100% with □~□
  - moderate buffer size

Analytic calculation possible for very simple systems only. Otherwise simulations must be used.

De-randomization: summary



→ Almost 100% efficiency and minimal deadtime are achieved if

- ADC is able to operate at rate >>f
- data processing and storing operates at ~f

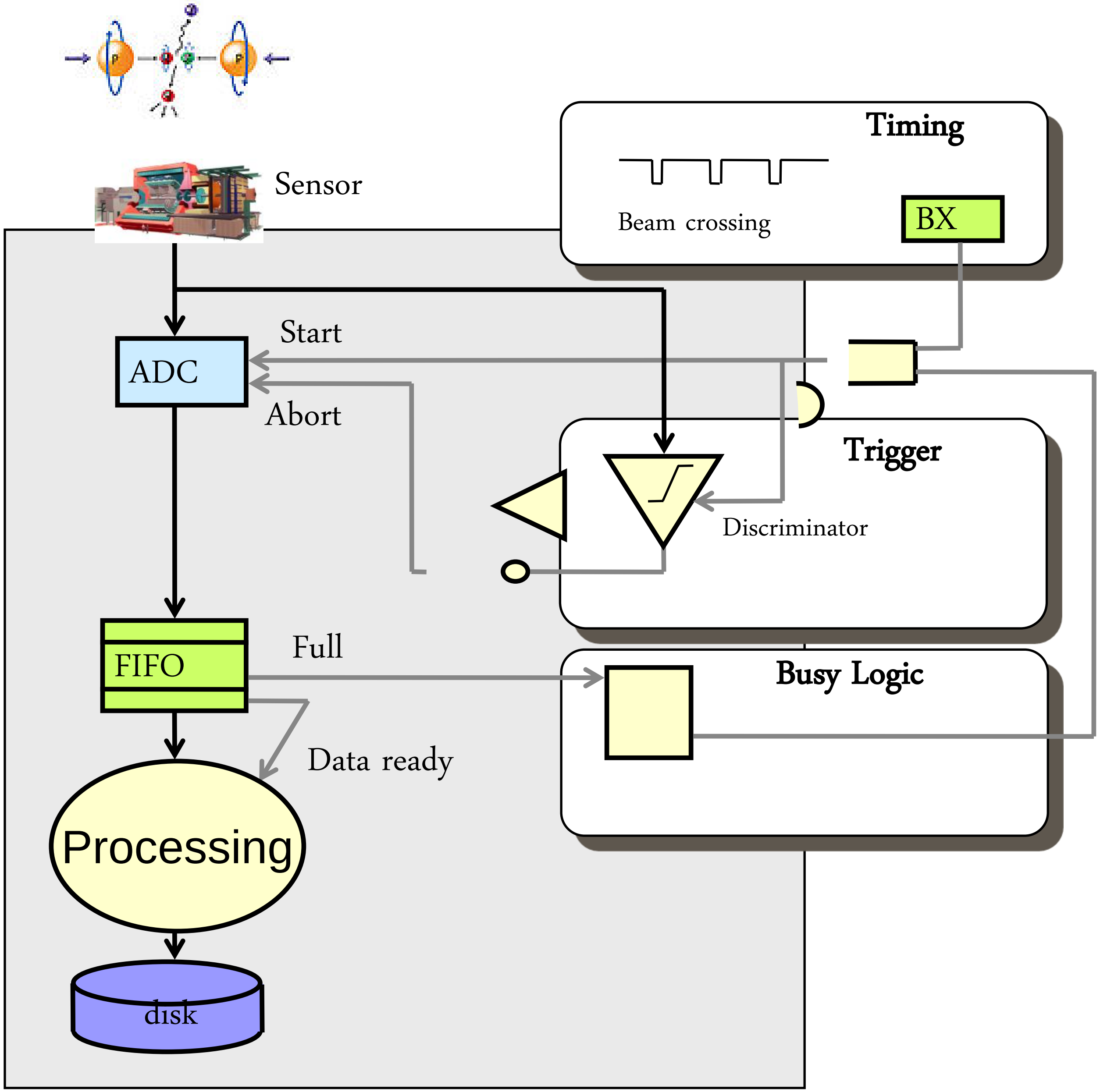
→ The FIFO decouples the low latency front-end from the data processing

- minimize the amount of "unnecessary" fast components

→ Could the delay be replaced with a "FIFO"?

- analog pipelines ⟶ Heavily used in LHC DAQs

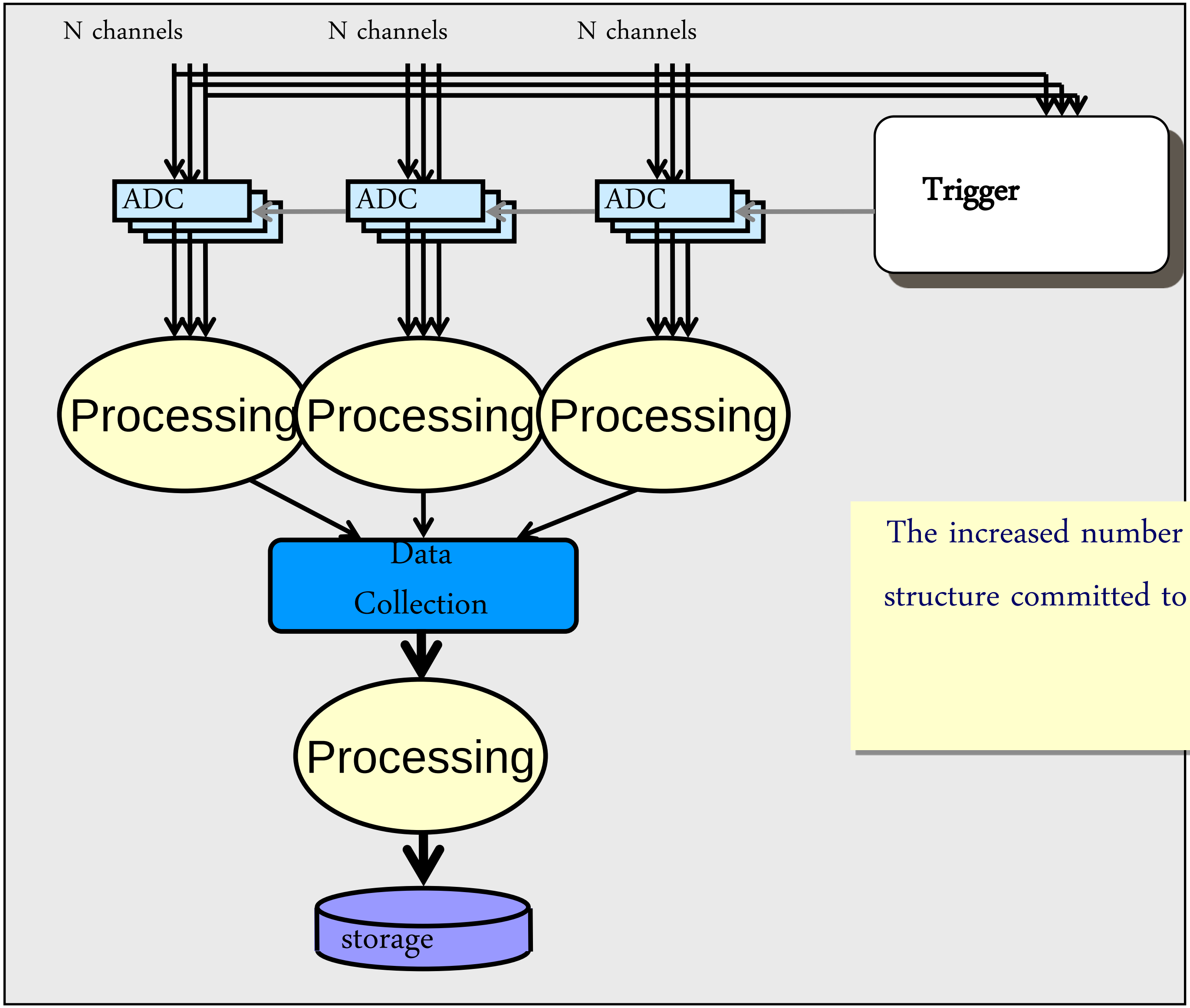
Basic DAQ: collider mode



- Particle collisions are synchronous
- Trigger rejects uninteresting events
- Even if collisions are synchronous, the triggers (i.e. good events) are unpredictable
- De-randomization is still needed

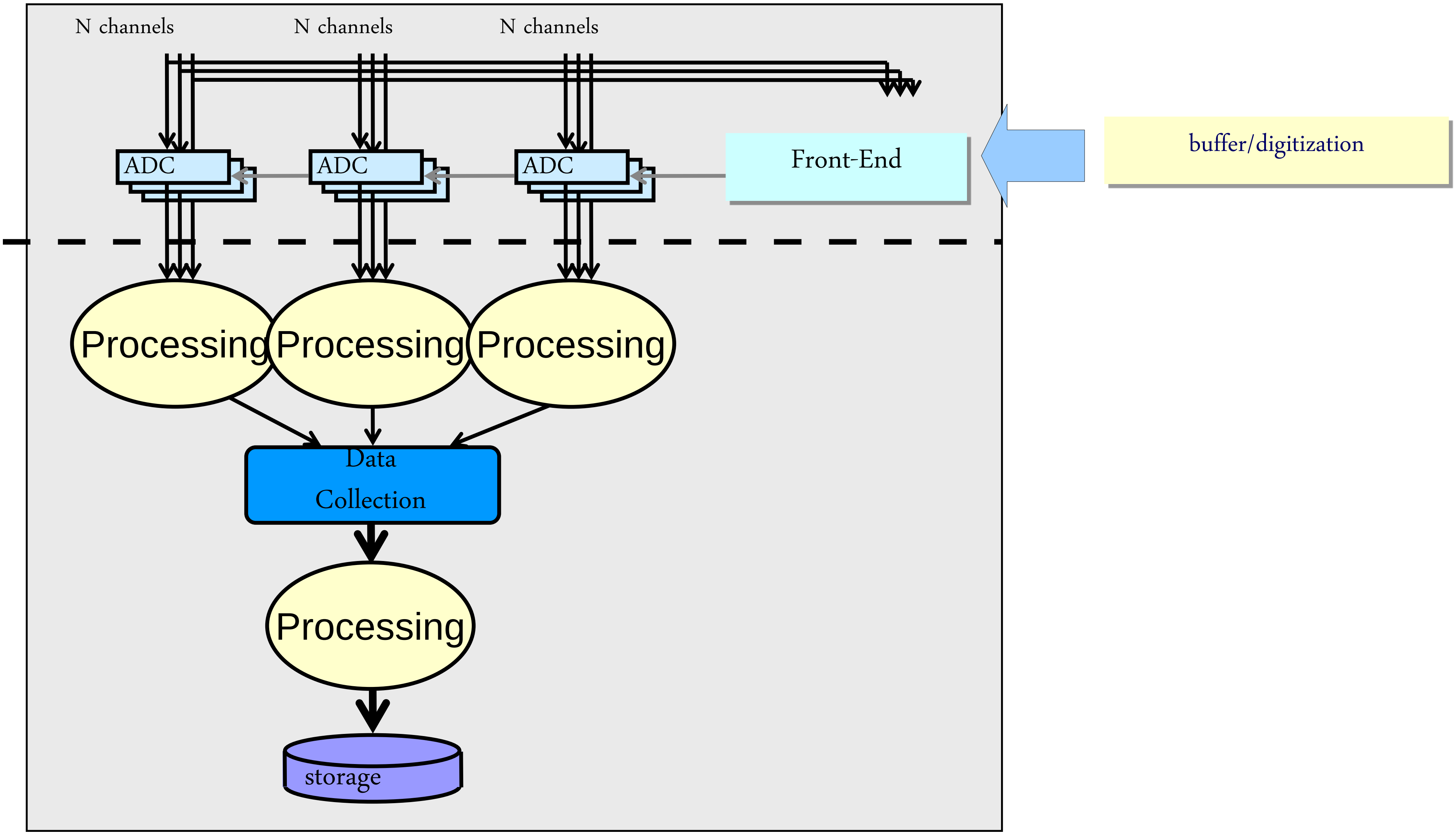
Scaling up: Network & Bus

Basic DAQ: more channels

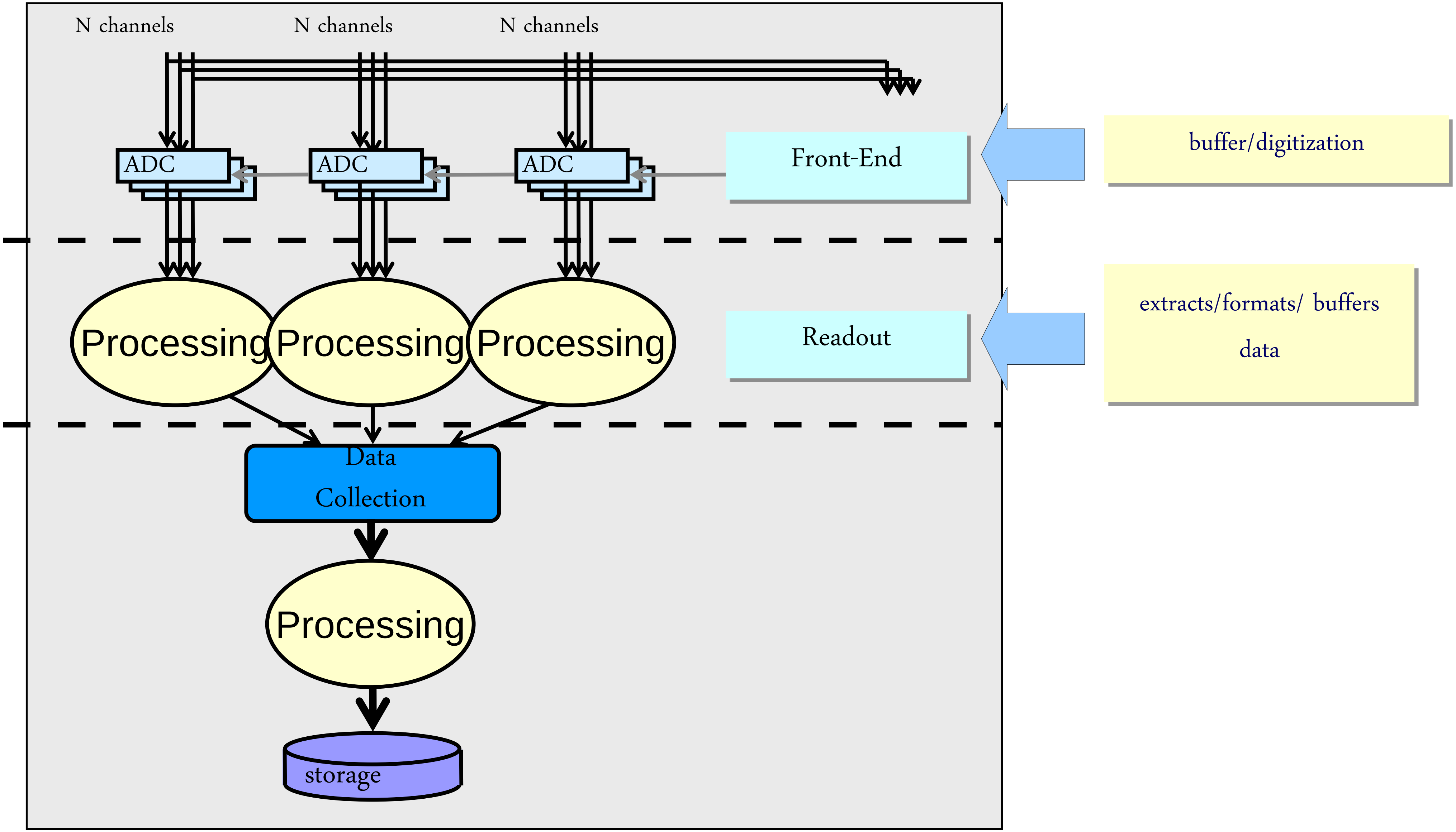


The increased number of channels requires a hierarchical structure committed to the data handling and conveyance

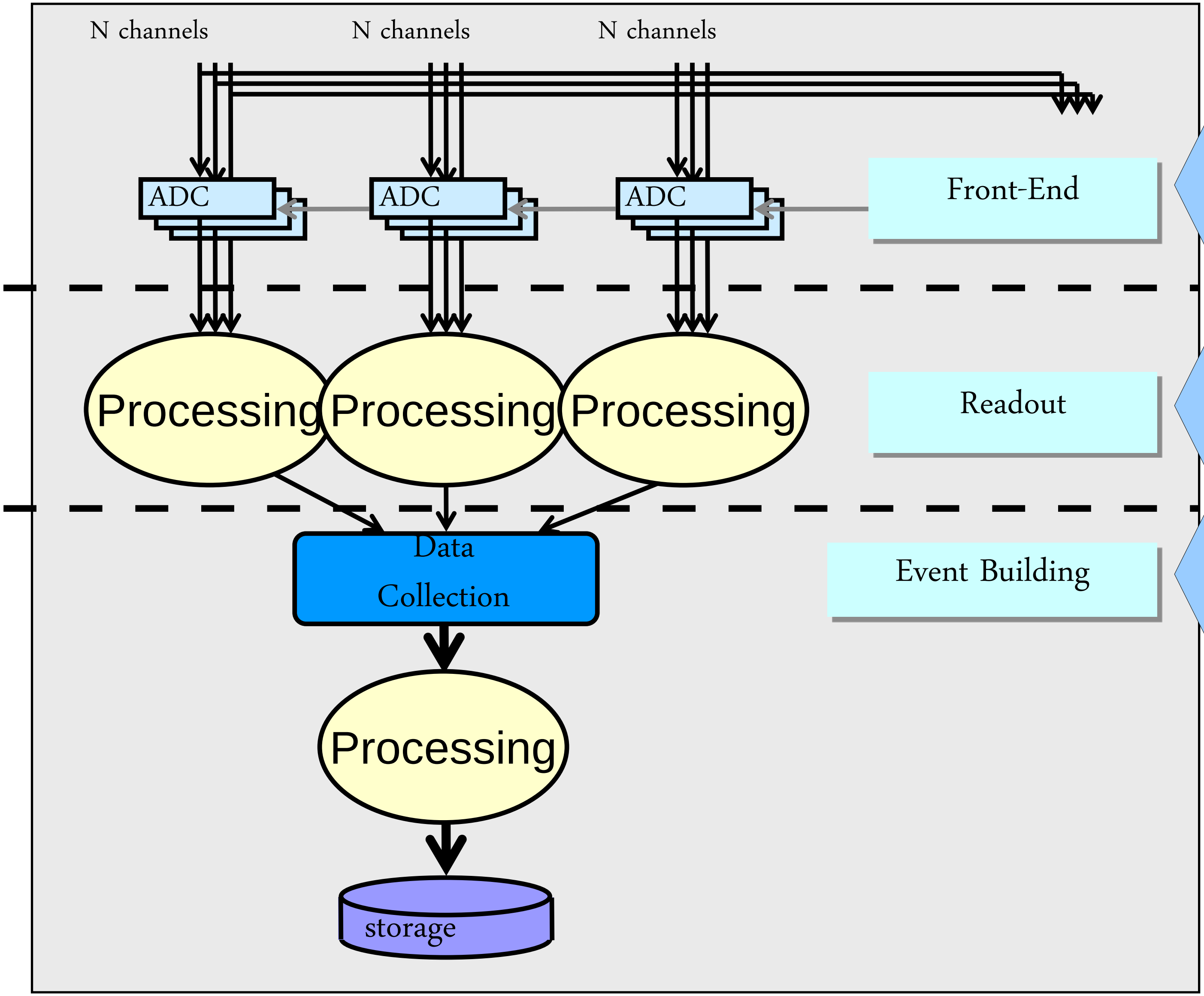
Large DAQ: Constituents



Large DAQ: Constituents



Large DAQ: Constituents



Front-End

buffer/digitization

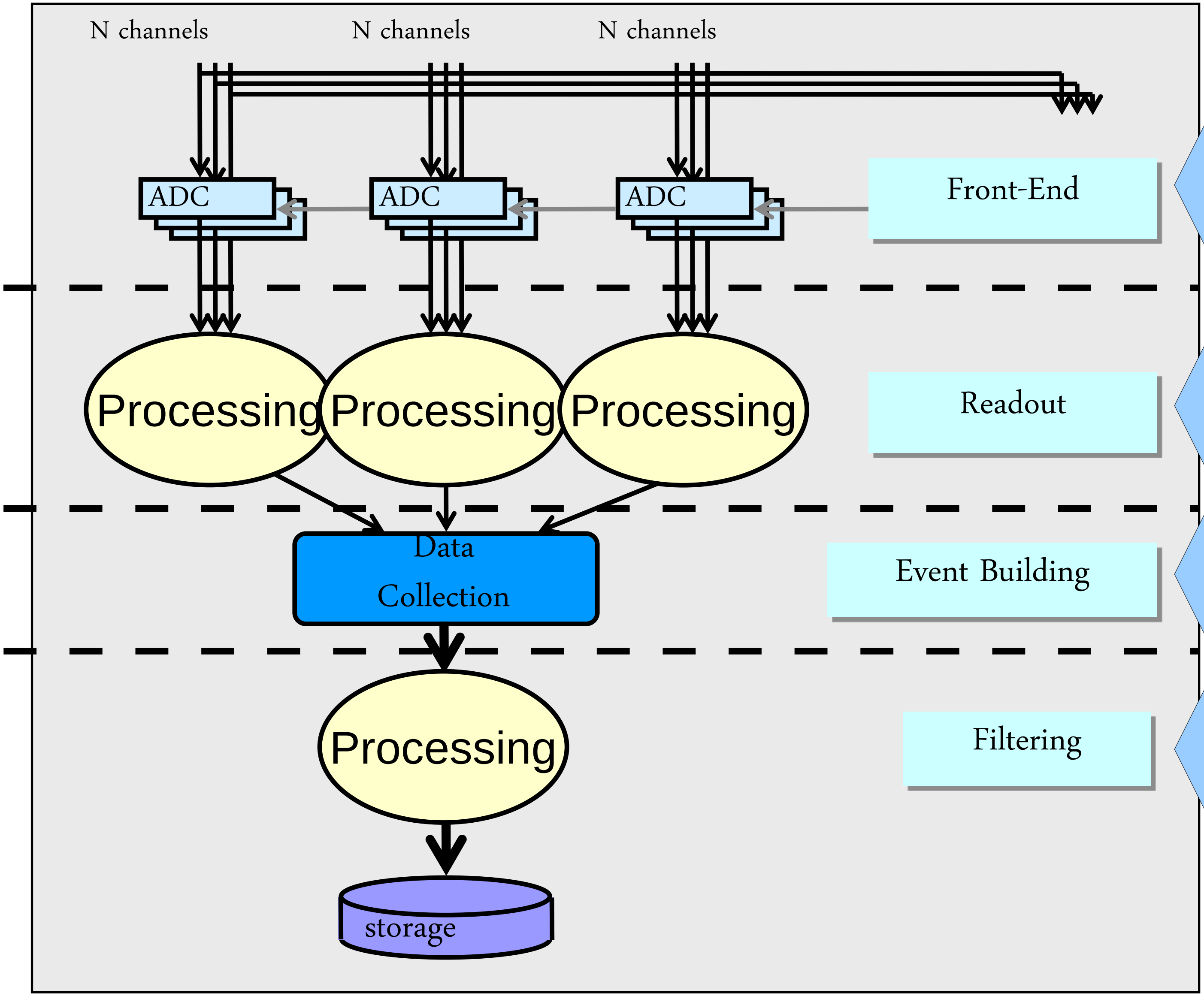
Readout

extracts/formats/ buffers data

Event Building

assembles/buffers events

Large DAQ: Constituents



Front-End

buffer/digitization

Readout

extracts/formats/ buffers data

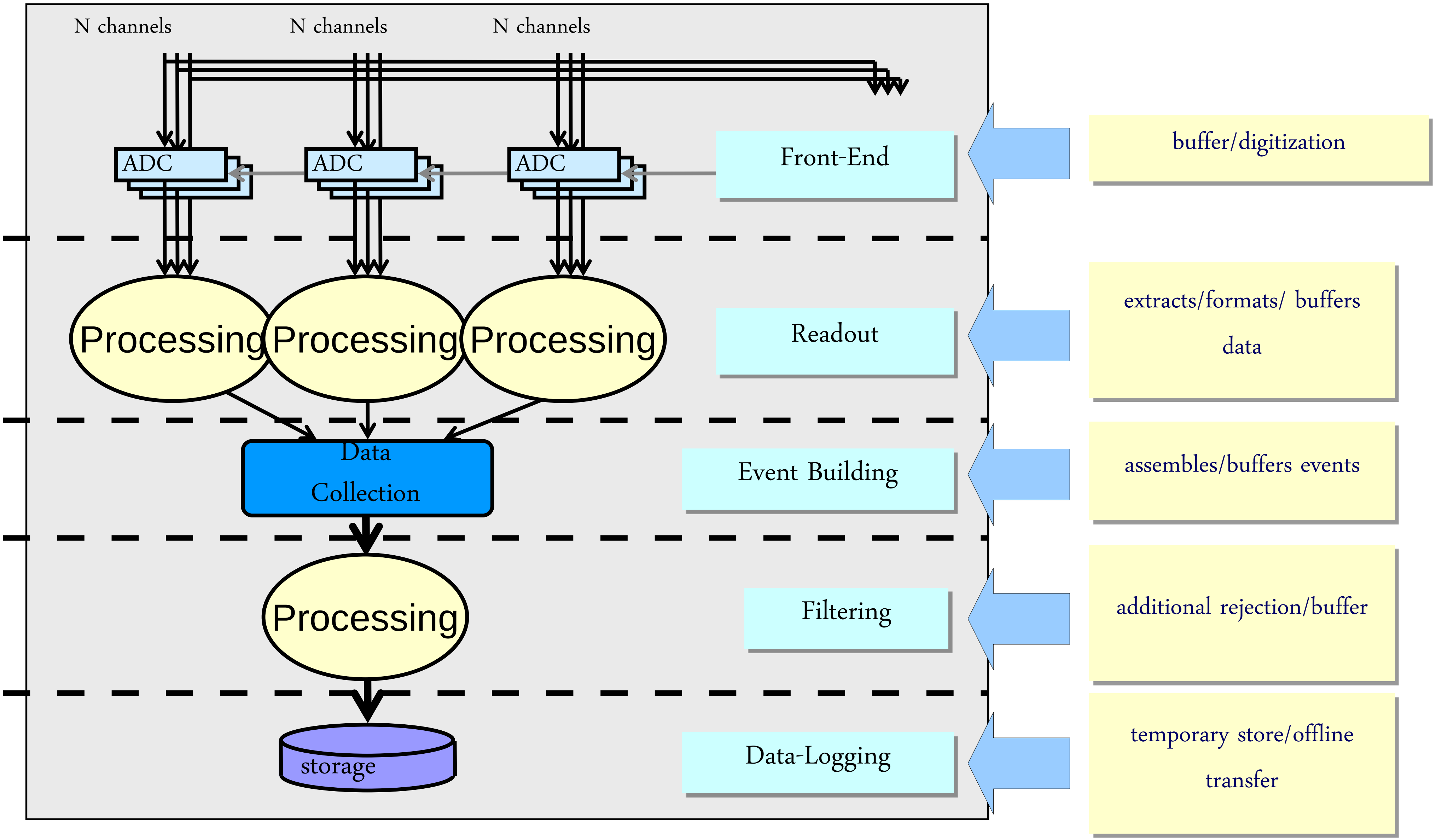
Event Building

assembles/buffers events

Filtering

additional rejection/buffer

Large DAQ: Constituents



Front-End — buffer/digitization

Readout — extracts/formats/ buffers data

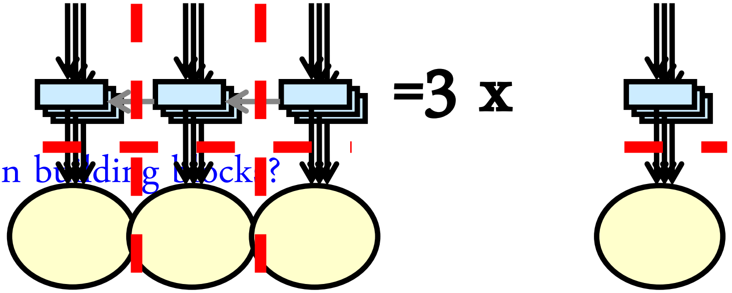
Event Building — assembles/buffers events

Filtering — additional rejection/buffer

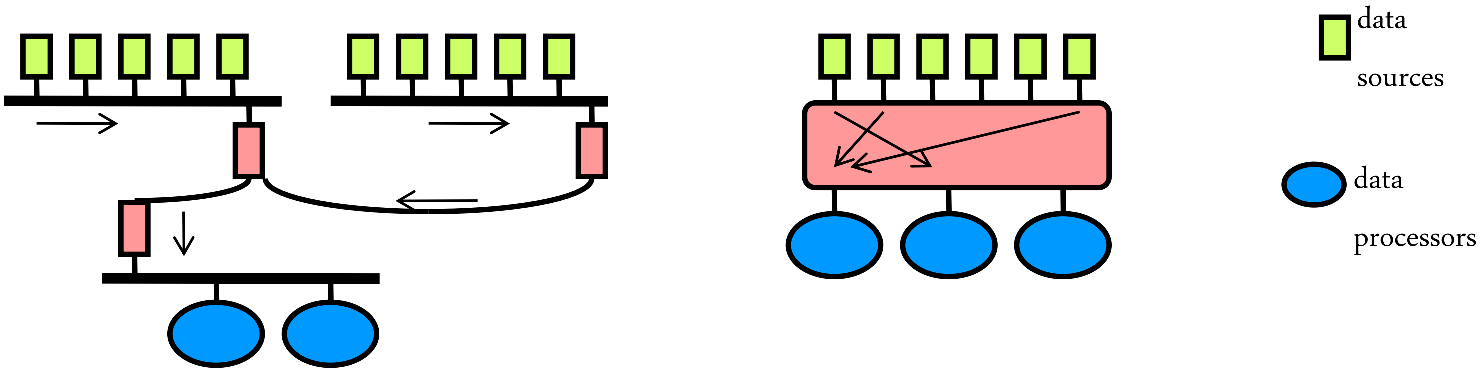
Data-Logging — temporary store/offline transfer

Readout Topology

- Reading out or building events out of many channels requires many components
- In the design of our hierarchical data-collection system, we have better define “building blocks”
  - example: readout crates, event building nodes, …

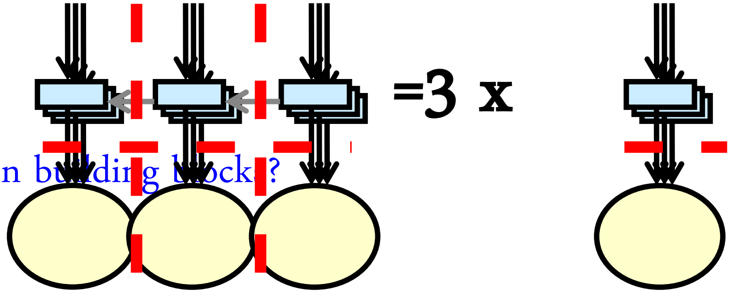


- How to organize the interconnections inside the building blocks and between building blocks?
- Two main classes: bus or network



Readout Topology

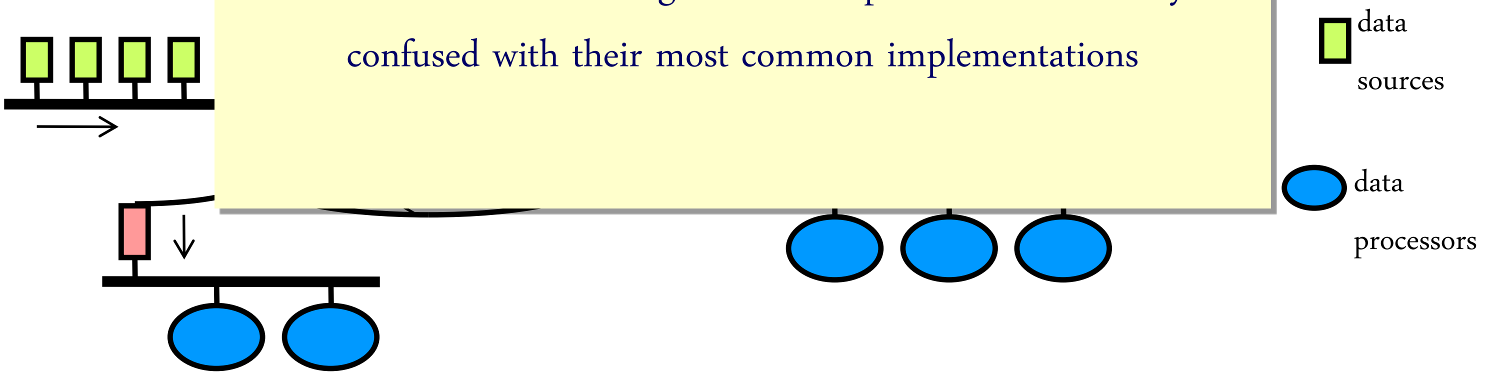
- Reading out or building events out of many channels requires many components
- In the design of our hierarchical data-collection system, we have better define “building blocks”
  - Example: readout crates, event building nodes, …



- How to organize the interconnections inside the building blocks and between building blocks?
- Two main classes: bus or network

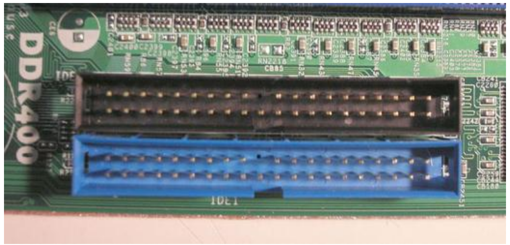
Warning

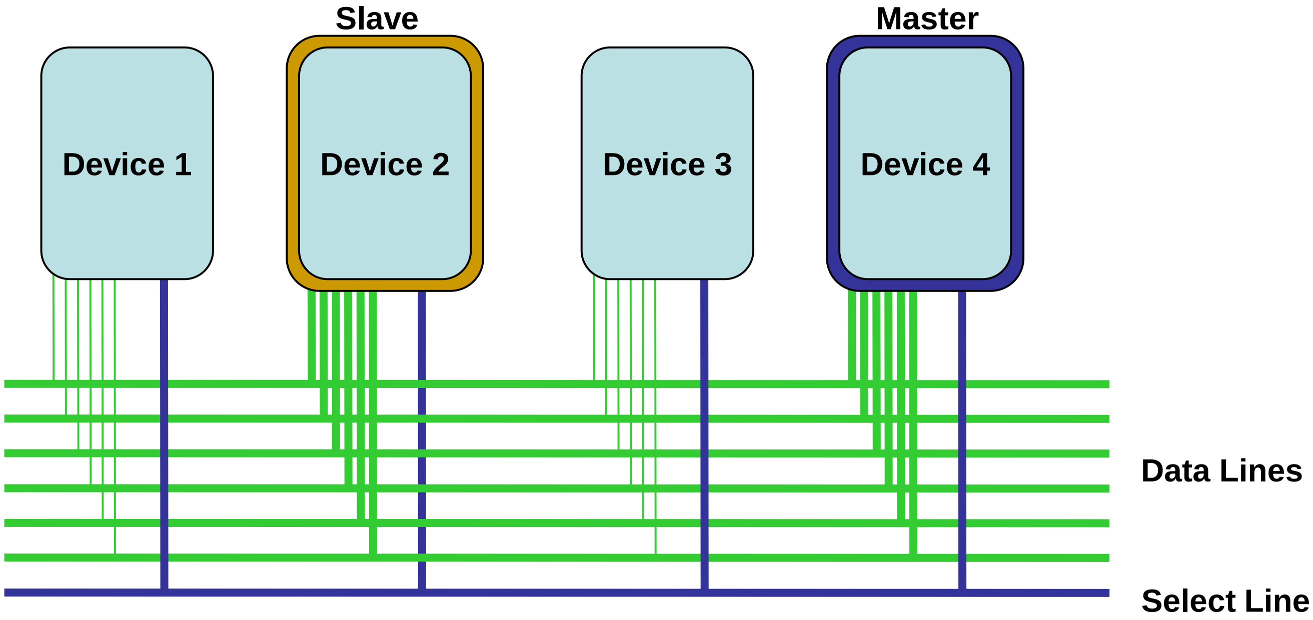
Buses and network are generic concepts that can be easily confused with their most common implementations



Buses

- Examples: VME, PCI, SCSI, Parallel ATA, …
  - local, external, crate, long distance
- Devices are connected via a shared bus
  - bus ⟶ group of electrical lines
  - sharing implies arbitration
- Devices can be master or slave
- Device can be addresses (uniquely identified) on the bus

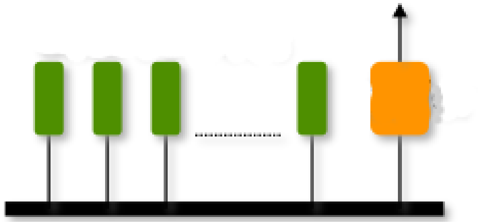




Bus facts

➔Simple □

- fixed number of lines (bus-width)
- devices have to follow well defined interfaces
  - mechanical, electrical, communication, ...



➔Scalability issues □

- bus bandwidth is shared among all the devices
- maximum bus width is limited
- maximum bus frequency is inversely proportional to the bus length
- maximum number of devices depends on the bus length

Bus facts

➔Simple □

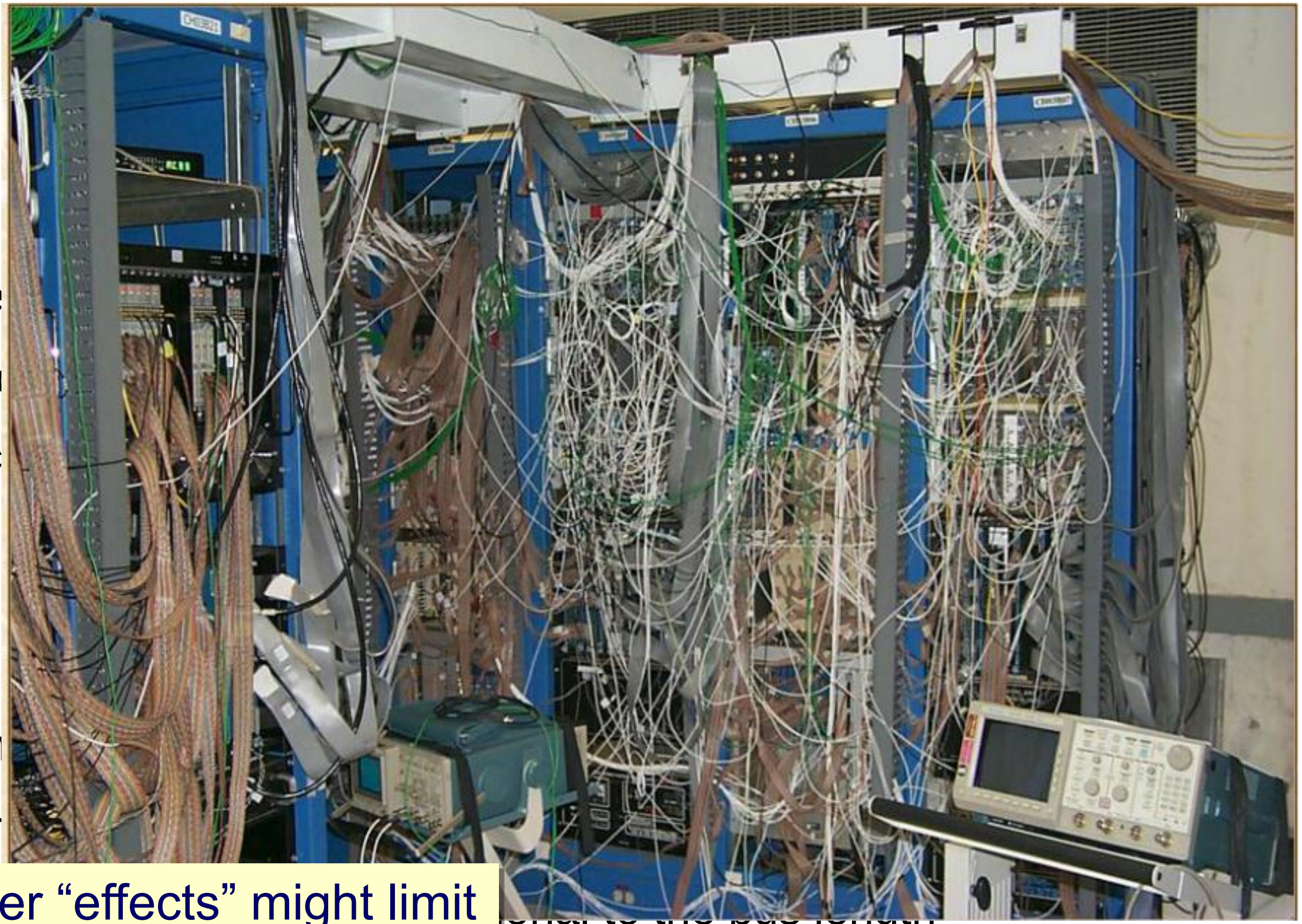
- fixed number of line
- devices have to foll
  - Mechanical, elec

➔Scalability issues □

- bus bandwidth is sh
- maximum bus width

ional to the bus length

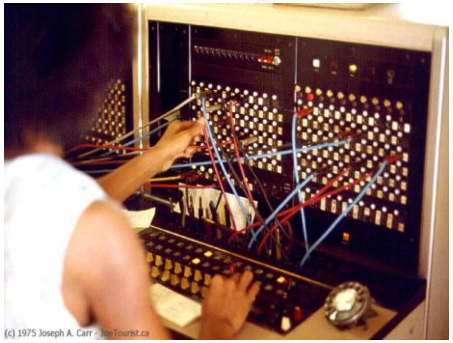
e bus length

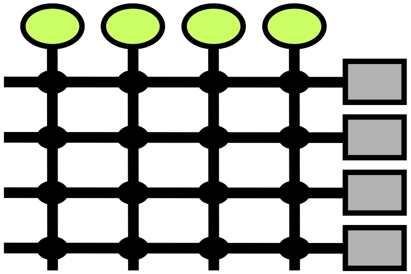


On the long term, other “effects” might limit the scalability of your system

Network

- ➔ Examples: Ethernet, Telephone, Infiniband, …
- ➔ All devices are equal
- ➔ Devices communicate directly with each other
  - – no arbitration, simultaneous communications
- ➔ Device communicate by sending messages
- ➔ In switched network, switches move messages between sources and destinations
  - – find the right path
  - – handle “congestion” (two messages with the same destination at the same time)
    - would you be surprised to learn that buffering is the key?

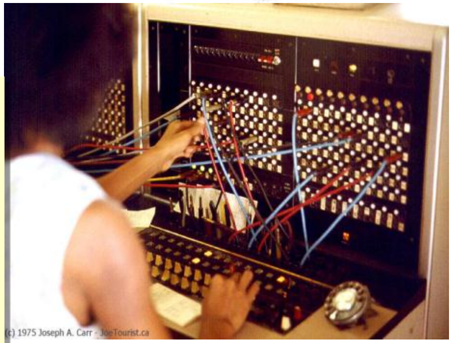

(c) 1975 Joseph A. Carr - JoeTourist.ca

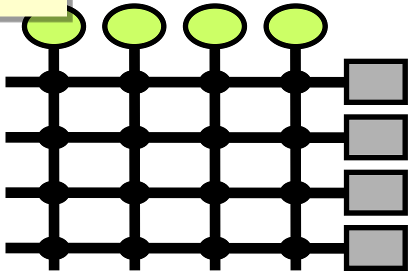


Network

- ➔ Examples: Ethernet, Telephone, Infiniband, …
- ➔ All devices are equal
- ➔ Devices communicate directly with each other
  - – no arbitration, simultaneous communications
- ➔ Device communicate by sending messages
- ➔ In switched r
  sources and
  - – find the rig
  - – handle "congestion" (two messages with the same destination at the same time)
    - • would you be surprised to learn that buffering is the key?

Thanks to these characteristics, networks do scale well. They are the backbones of LHC DAQ systems





Modular Electronics

➔ Standard electronics "functions" implemented in well-defined "containers"

- re-use of generic modules for different applications
- limit the complexity of individual modules → reliability & maintainability
- easy to upgrade to newer versions
- profit from commercially available "functions"

➔ "Containers" are normally well-defined standards defining mechanical, electrical, … , interfaces

- "easy" design and integrate your own module

➔ Historically, in HEP, modular electronics was bus-based

- currently entering a mixed phase …

Allow building your own data-acquisition system just connecting predefined functions → Fast & Efficient

NIM

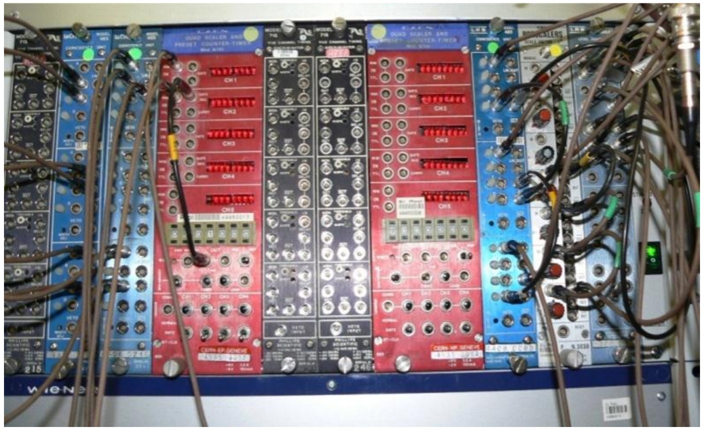
➔NIM (1964)

- “Nuclear Instrumentation Modules”? “National Instrument Module”? Just **NIM**

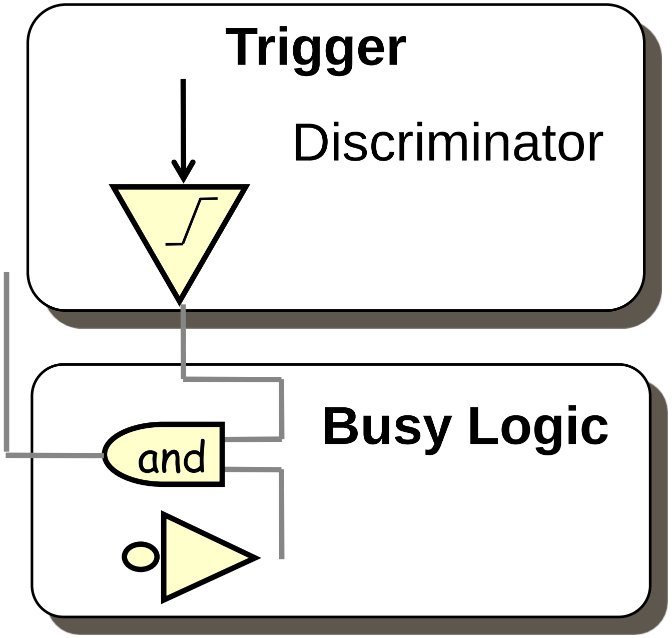
➔NIM modules usually

- do not need software, are not connected to PCs
- implement logic and signal processing functions
  - Discriminators, Coincidences, Amplifiers, Logic gates, …

➔Typically implement basic Trigger and Busy system

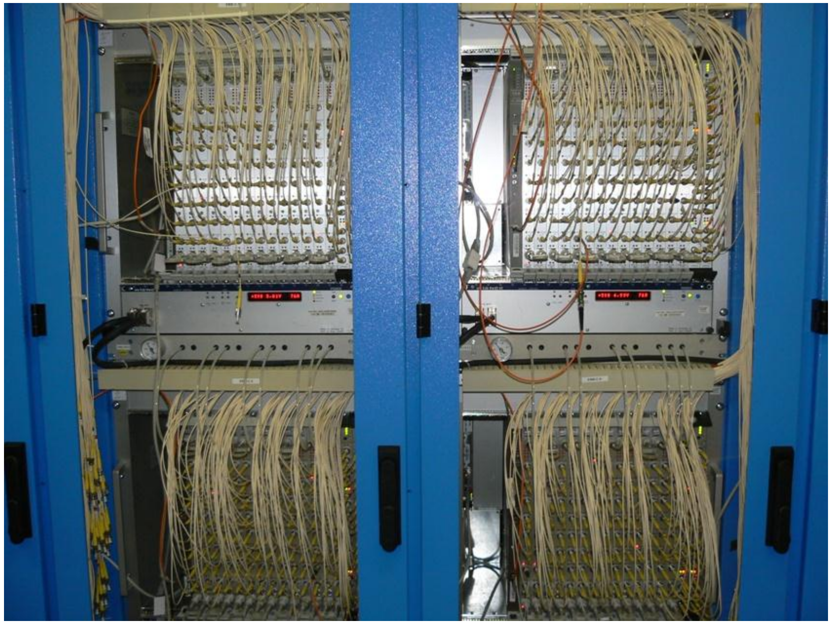


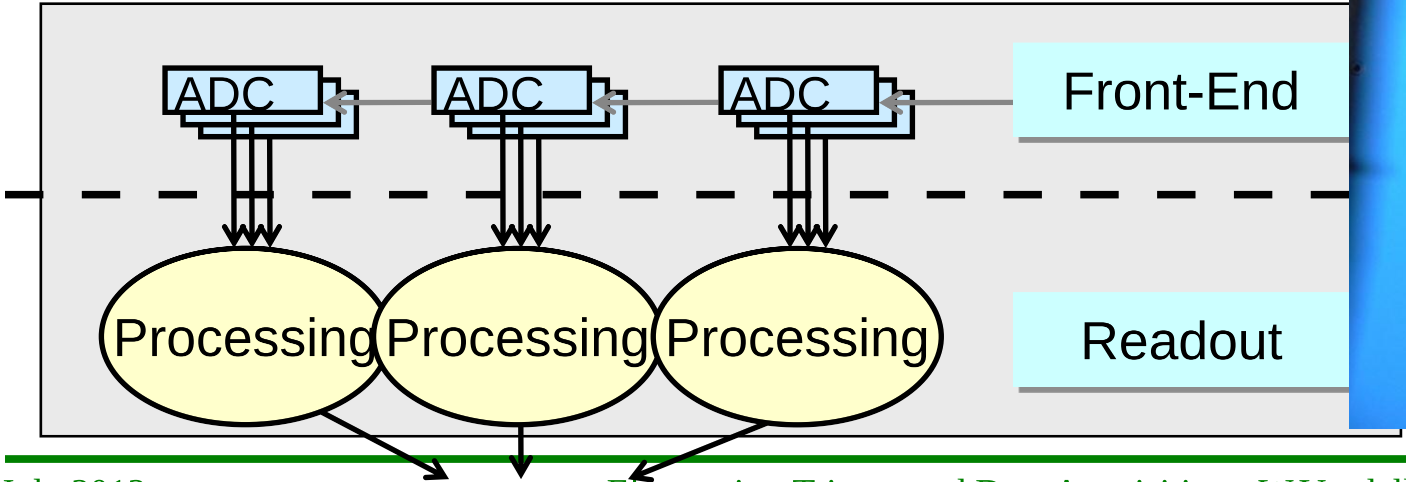
New modules still appear on the market
Very diffused in medium-sized HEP experiments
Found in the counting rooms of LHC experiments



VMEbus

- ➔ VMEbus: modules communicated via a “backplane”
  - electrical, mechanical and communication protocols
- ➔ Choice of many HEP experiments
  - relatively simple protocol
  - a lot of commercially available functions
- ➔ More than 1000 VMEbus crates at CERN





Other (arising) standards

➔PCI-based

➔We know buses have limited scalability. Can we have “network-based” modular electronics?

➔VXS → essentially VME plus switched interconnectivity

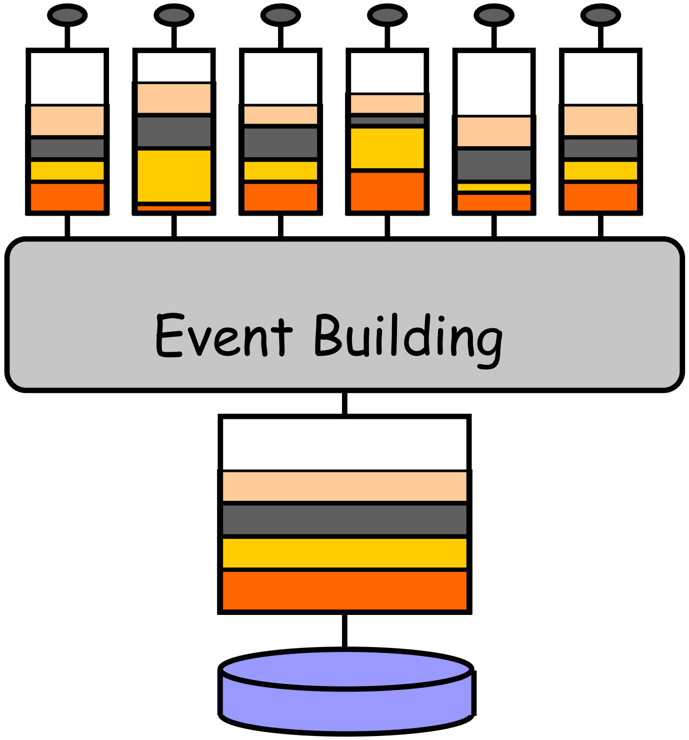
➔ATCA and derivatives

- standard designed for telecom companies
- high redundancy, data-throughput, availability
- **being considered for several LHC upgrade programs**

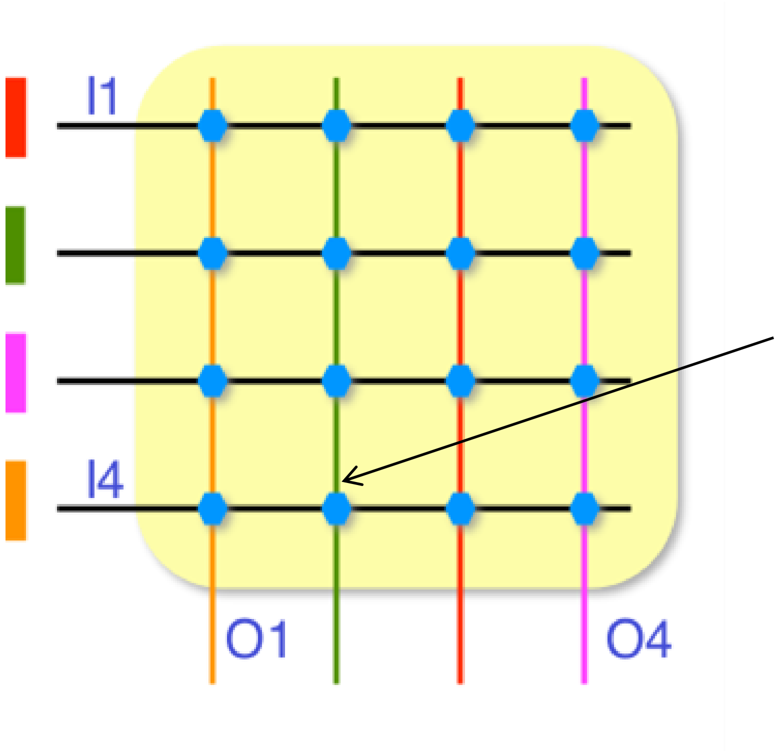
Event Building

Event Building: network perspective

- ➔ Event Building: collection and formatting of all the data element of an event into a single unit
  - normally last step before high-level trigger or storage
  - can be implemented on buses, can use custom interconnects, can be based on (Ethernet) **network**
- ➔ Network-based EB is choice of all LHC experiments and a case study for networking in DAQ

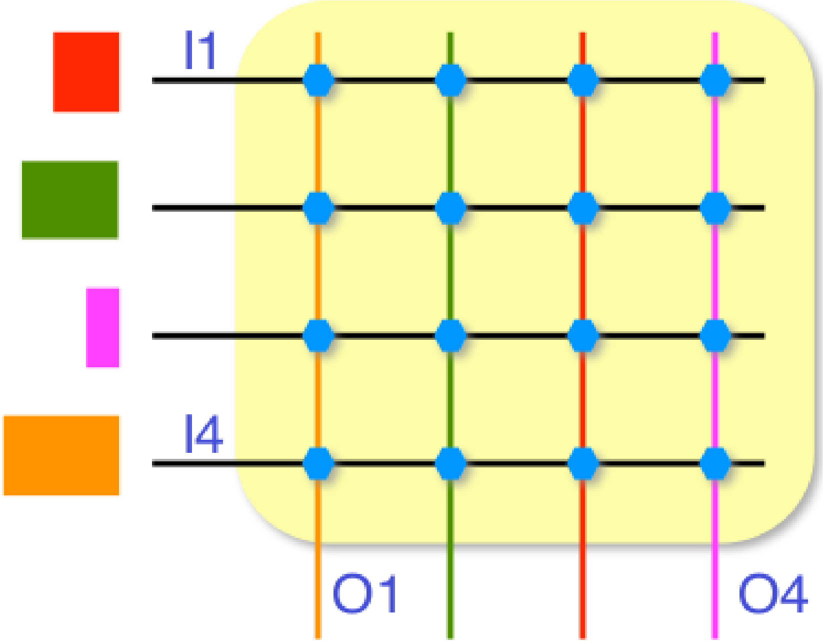


Network switch: crossbar



- ➔ Each input port can potentially be connected to each output port
- ➔ At any given time, only one input port can be connected to a given output port
- ➔ Different output ports can be reached concurrently by different input ports

Network switch: crossbar

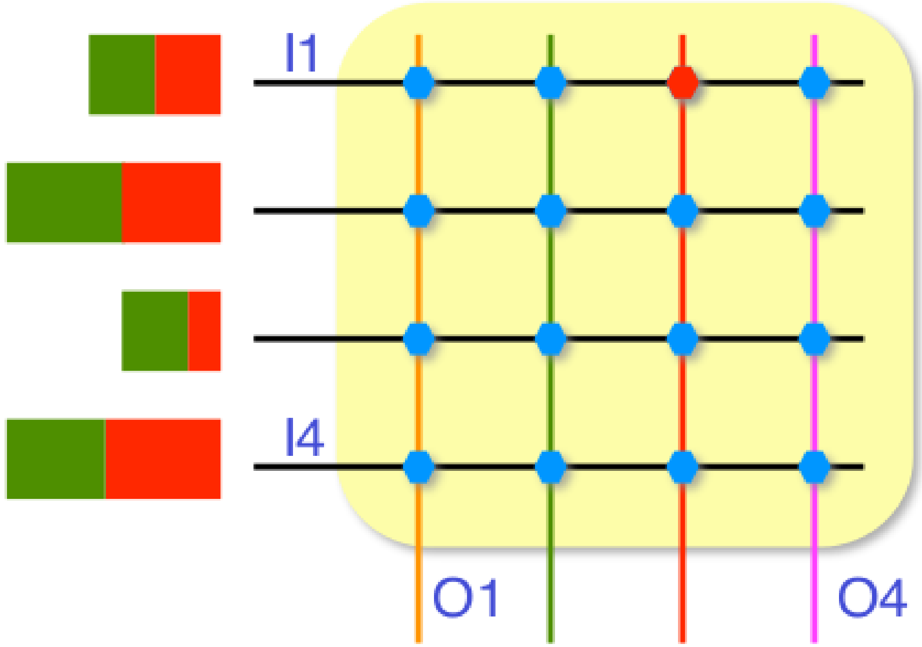


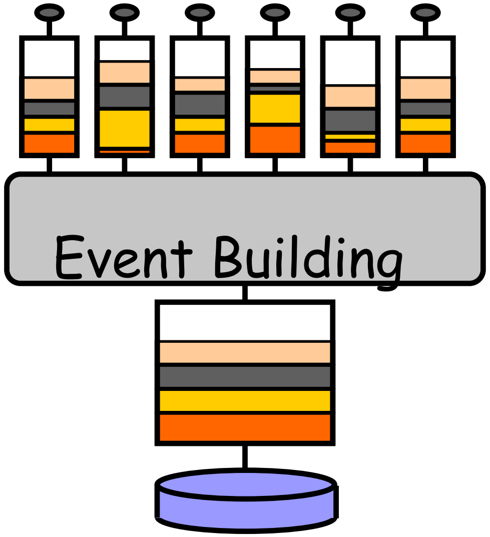
➔ Ideal situation → all inputs send data to different outputs

No interference (Congestion)

All input ports send data concurrently

Crossbar switch: event building





➔EB workload implies converging data flow

- all inputs want to send to same destination **at the same time**

➔“Head of line blocking”

- congestion

Congestion

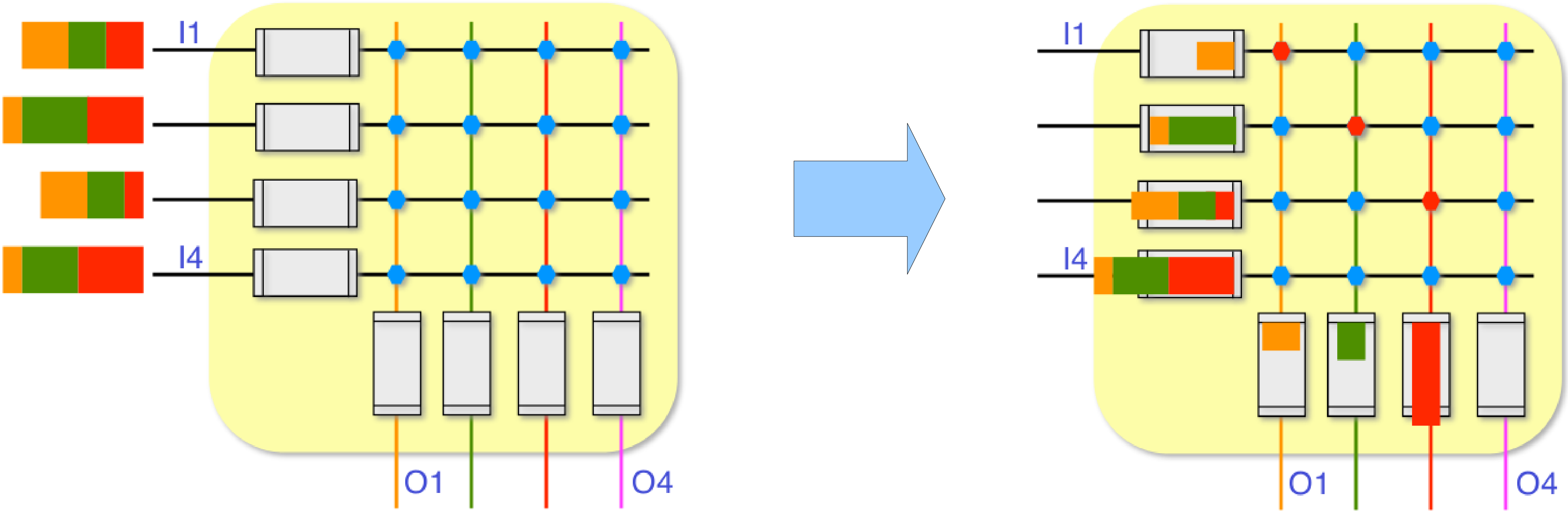
➔ Well know phenomena ..

- in Geneva and other cities

➔ Differently from road traffic, Ethernet HW is allow to "drop" packets

- Higher level protocols have to take care of re-sending
- Possibly important performance impacts

Queuing



➔Adding input and output FIFO dramatically improve the EB pattern handling

➔EB workload anyway problematic

- FIFO size is limited, variable data size
- limited internal switching speed

}

Traffic shaping
or
Network over-sizing

LHC experiments

Multi-level trigger systems

➔ Sometime impossible to take a proper decision in a single place

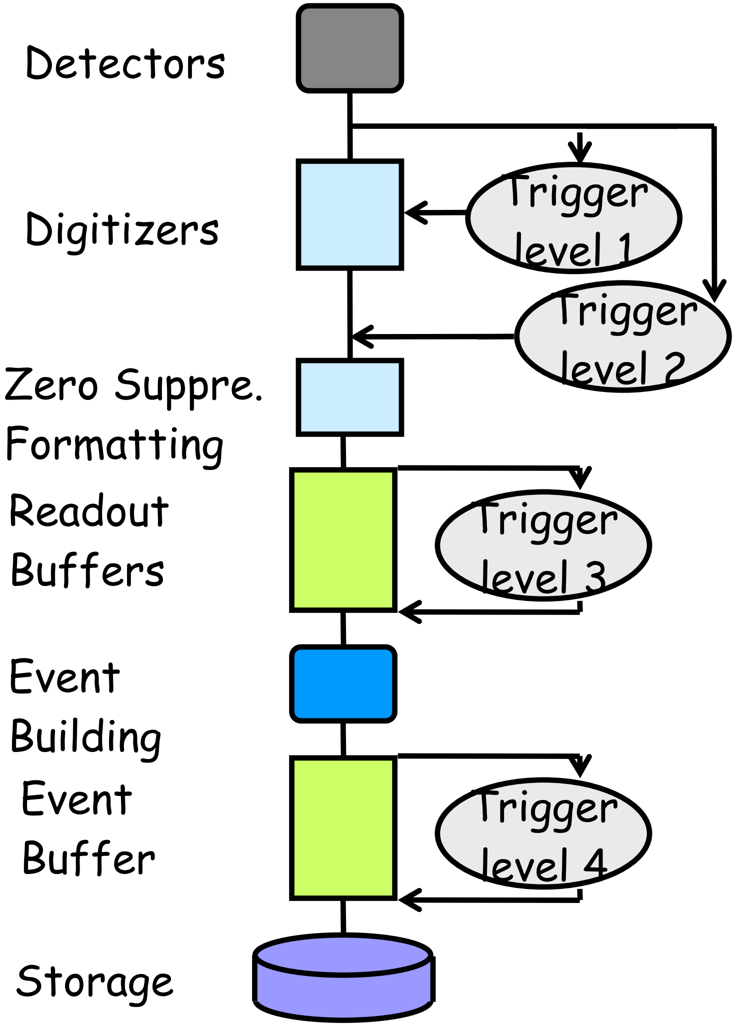
- too long decision time
- too far
- too many inputs

➔ Distribute the decision burden in a hierarchical structure

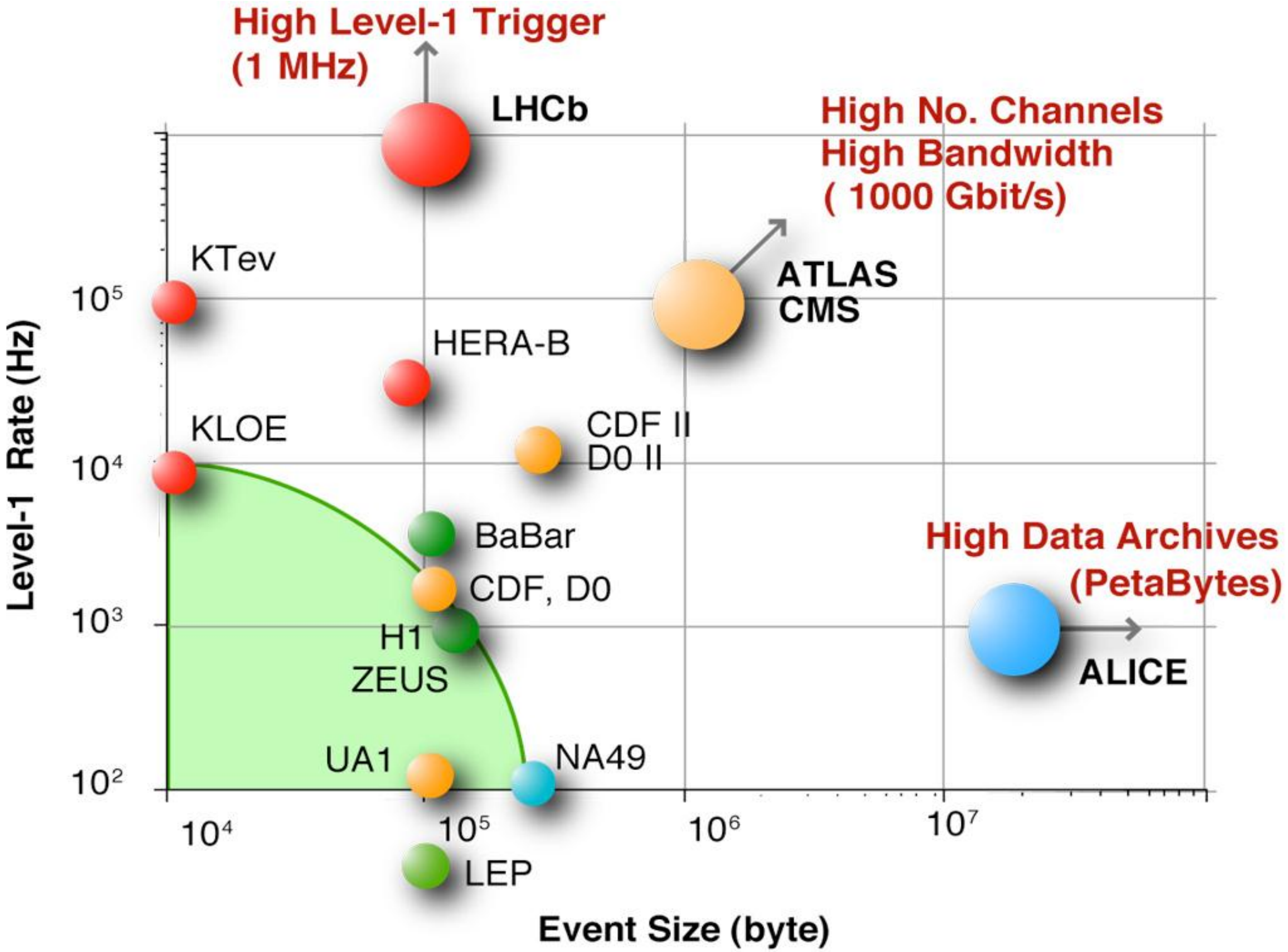
- usually $\tau_{N+1} \gg \tau_N$, $f_{N+1} \ll f_N$

➔ At the DAQ level, proper buffering must be provided for every trigger level

- absorb latency
- de-randomize



LHC DAQ phase-space



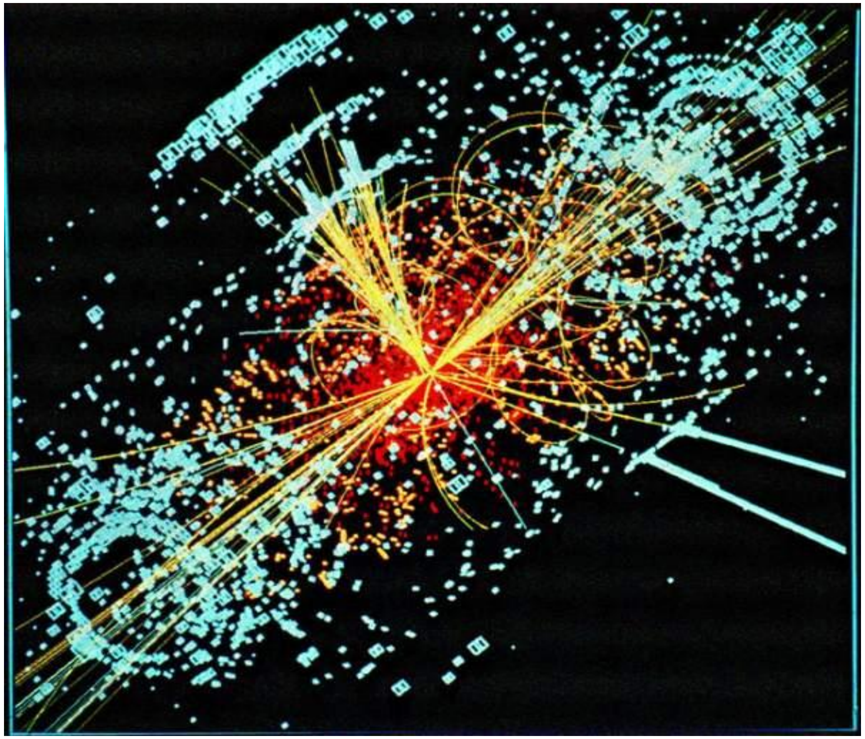
Trigger & DAQ Challenges at the LHC

- ➔LHC experiments have O(10^7) channels operating at 40MHz (25ns) → **40TB/s**
- ➔In addition, most of these are useless

$$\sigma_H / \sigma_{Tot} \sim O(10^{-13})$$

- ➔Events are extremely complicated
- ➔Experiments are huge (O(10m))

Multi-level trigger system and …

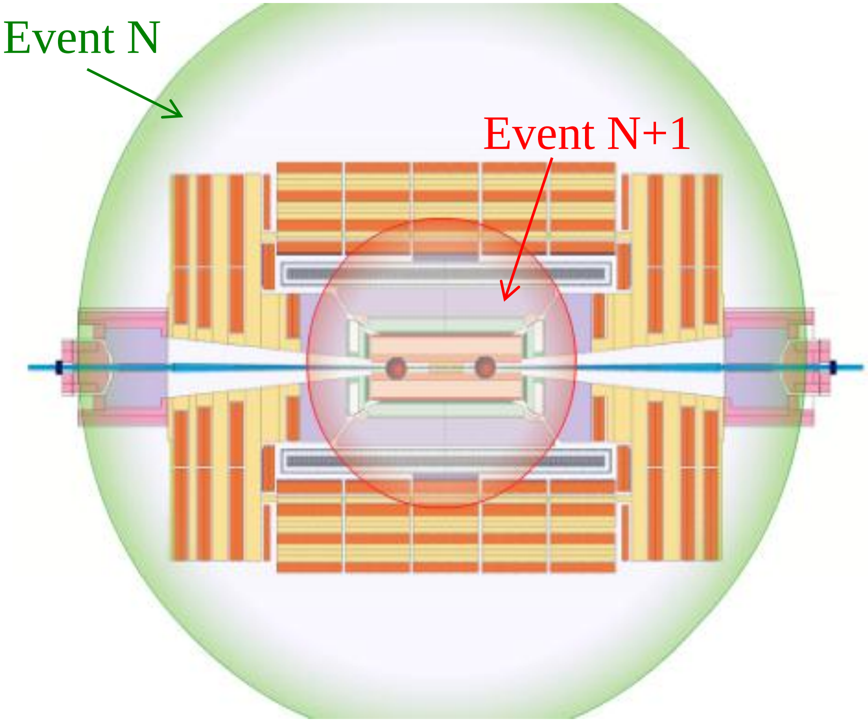


LHC L1 Trigger and FE electronics

➔ Particle time of flight >> 25ns

➔ Cable delays >> 25ns

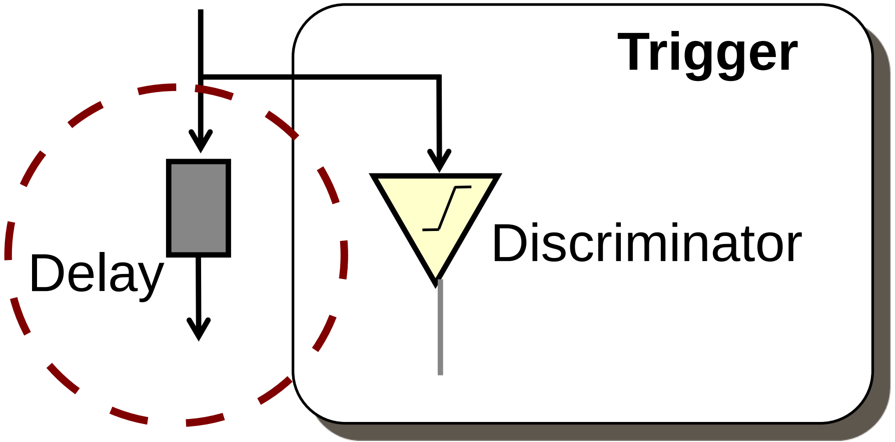
Dedicated synchronization, timing and signal distribution facilities



➔ Typical L1 decision latency is O(□s)

- dominated by signal propagation in cables

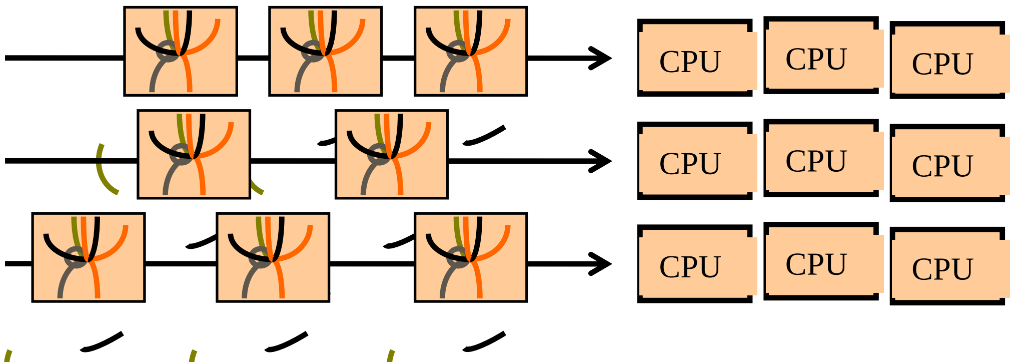
Digital/analog <u>custom</u> front-end pipelines store information during L1 trigger decision



LHC: After L1?

➔ Custom hardware L1 trigger and front-end electronics followed by network-based **High-Level Trigger** farm(s)

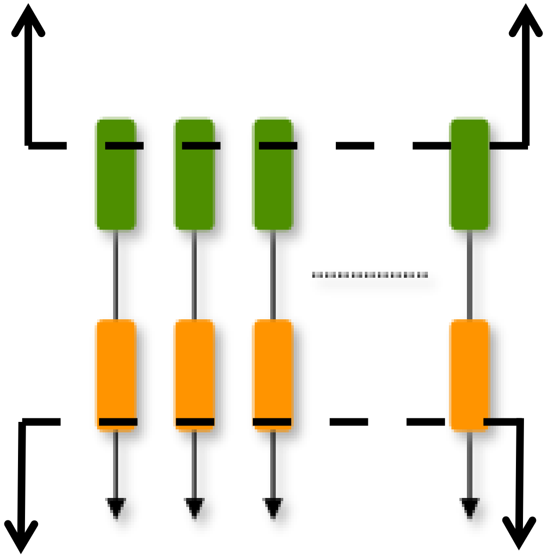
- commercially available HW organized in a farm
  - events are independent



➔ Connection between custom section and the network-based one achieved via dedicated HW and point-to-point connectivity

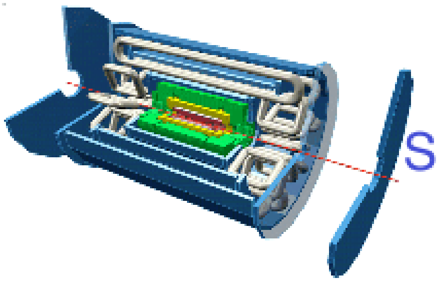
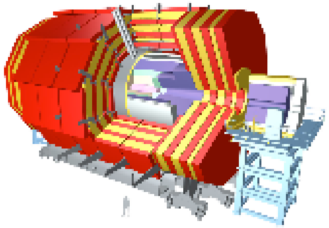
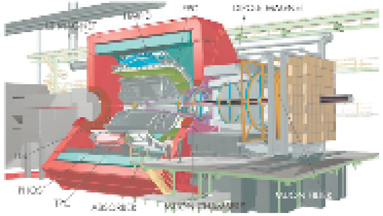
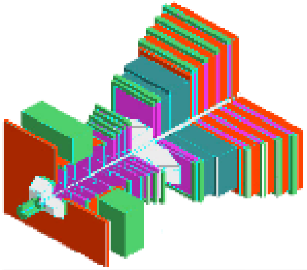
- electrical or optical, standard or custom

Custom elx

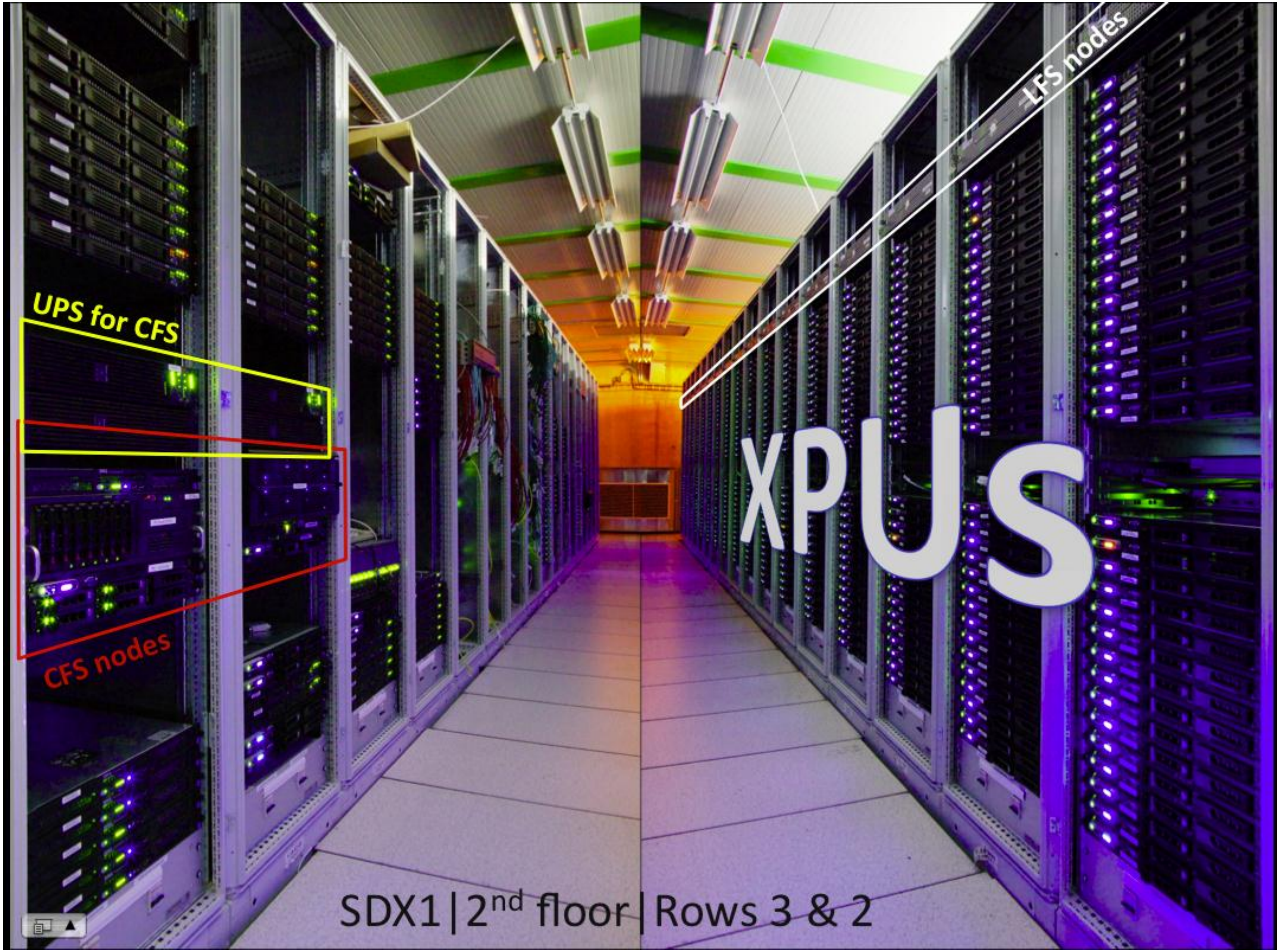


Network based

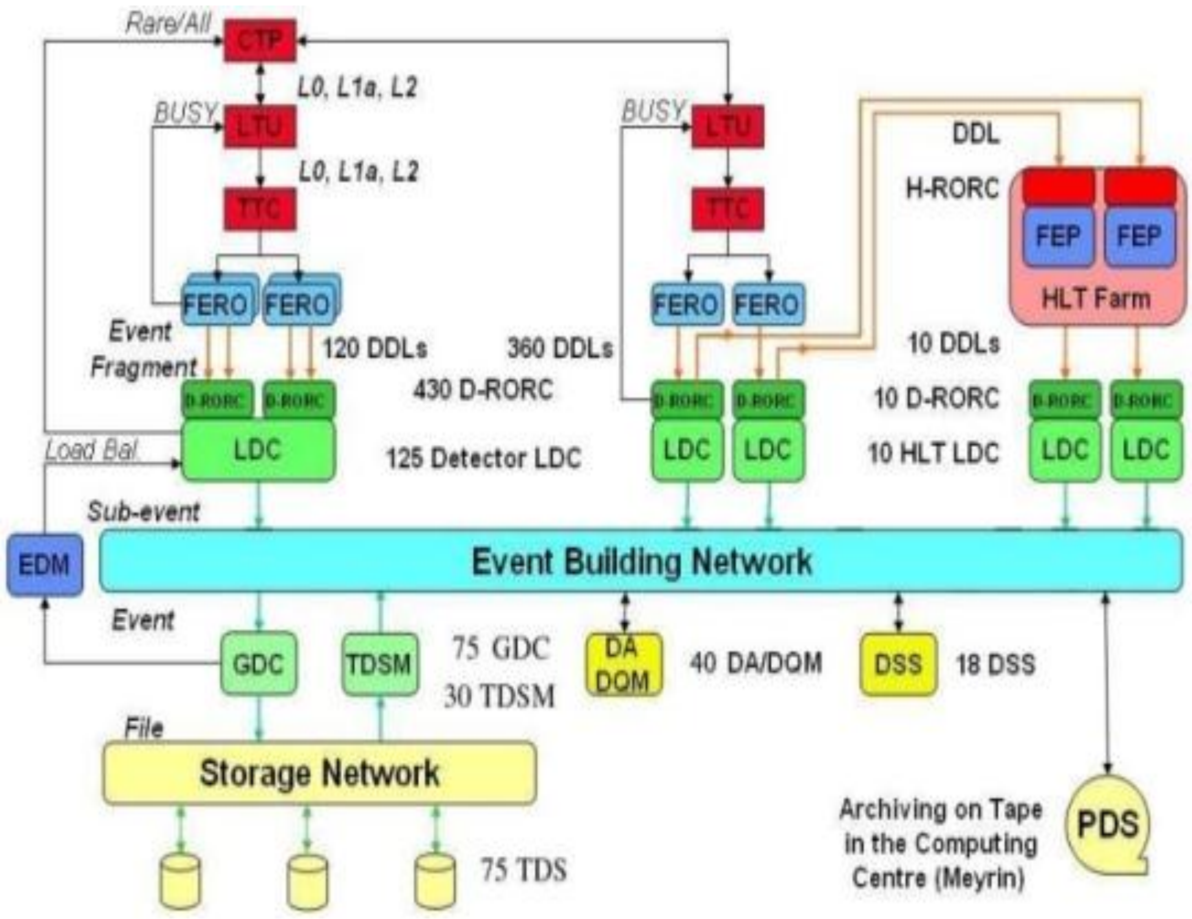
Read-out links at the LHC

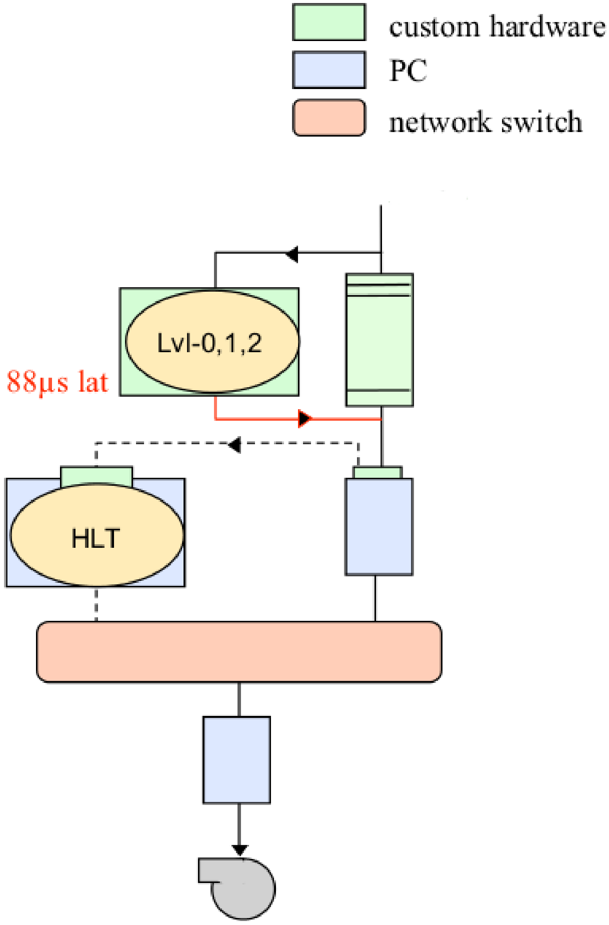
				Flow Control
	SLINK	Optical: 160 MB/s Receiver card interfaces to PC.	≈ 1600 Links	Yes
	SLINK 64	LVDS: 400 MB/s (max. 15m) (FE on average: 200 MB/s to readout buffer) Receiver card interfaces to commercial NIC (Network Interface Card)	≈ 500 links	yes
	DDL	Optical 200 MB/s Half duplex: Controls FE (commands, Pedestals,Calibration data) Receiver card interfaces to PC	≈ 500 links	yes
	TELL-1 & GbE Link	Copper quad GbE Link Protocol: IPv4 (direct connection to GbE switch) Forms “Multi Event Fragments” Implements readout buffer	≈ 400 links	no

ATLAS HLT Farm



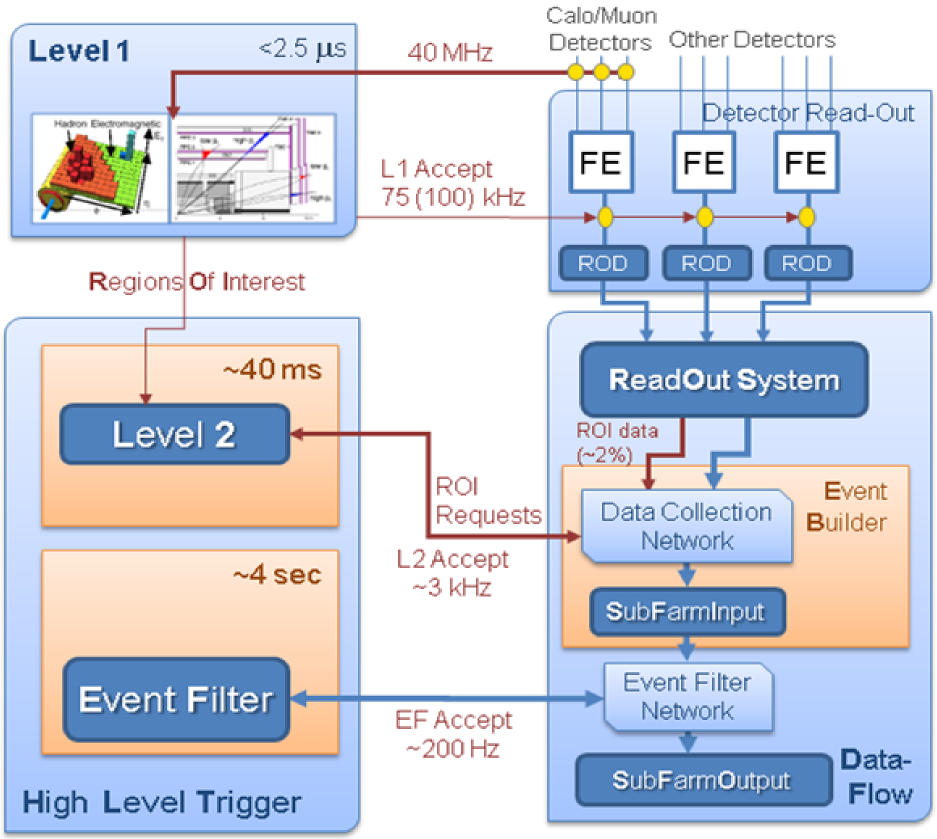
ALICE

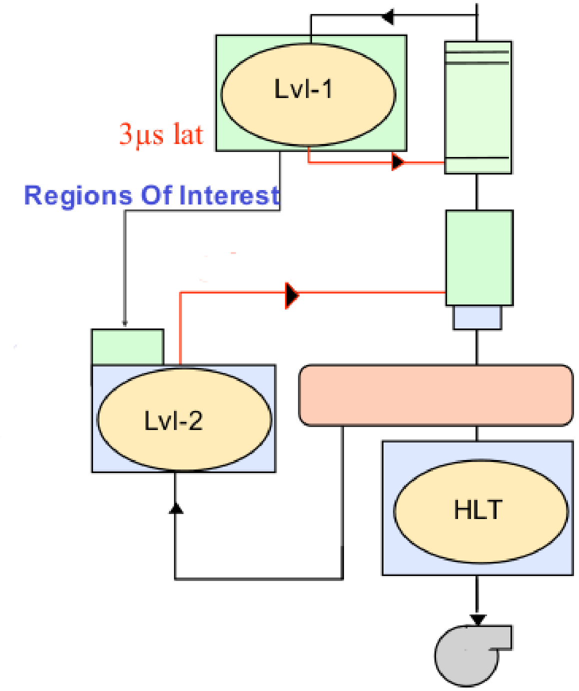

Rare/All
CTP
L0, L1a, L2
BUSY
LTU
L0, L1a, L2
TTC
BUSY
LTU
TTC
DDL
H-RORC
FEP
FEP
HLT Farm
FERO
FERO
FERO
FERO
Event
Fragment
120 DDLs
360 DDLs
10 DDLs
D-RORC
D-RORC
430 D-RORC
D-RORC
D-RORC
10 D-RORC
D-RORC
D-RORC
Load Bal.
LDC
125 Detector LDC
LDC
LDC
10 HLT LDC
LDC
LDC
Sub-event
EDM
Event Building Network
Event
GDC
TDSM
75 GDC
30 TDSM
DA
DQM
40 DA/DQM
DSS
18 DSS
File
Storage Network
75 TDS
Archiving on Tape
in the Computing
Centre (Meyrin)
PDS


custom hardware
PC
network switch
Lvl-0,1,2
88µs lat
HLT

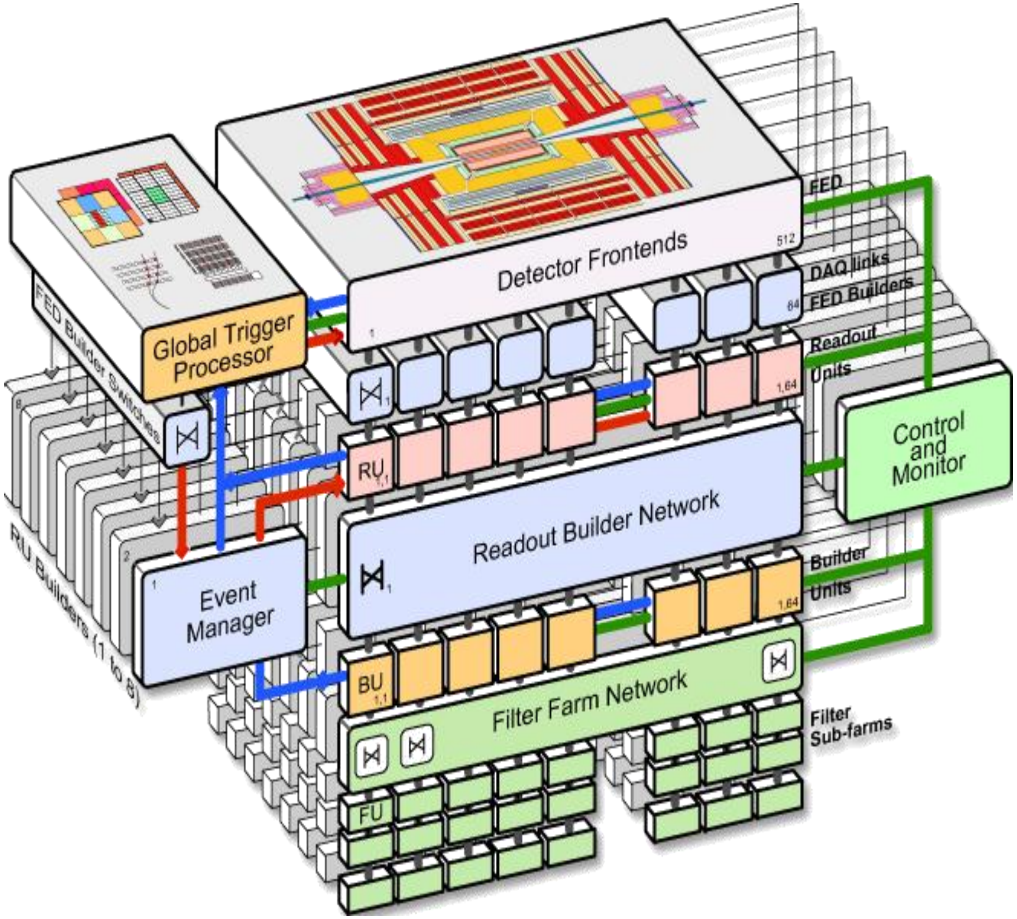
ATLAS

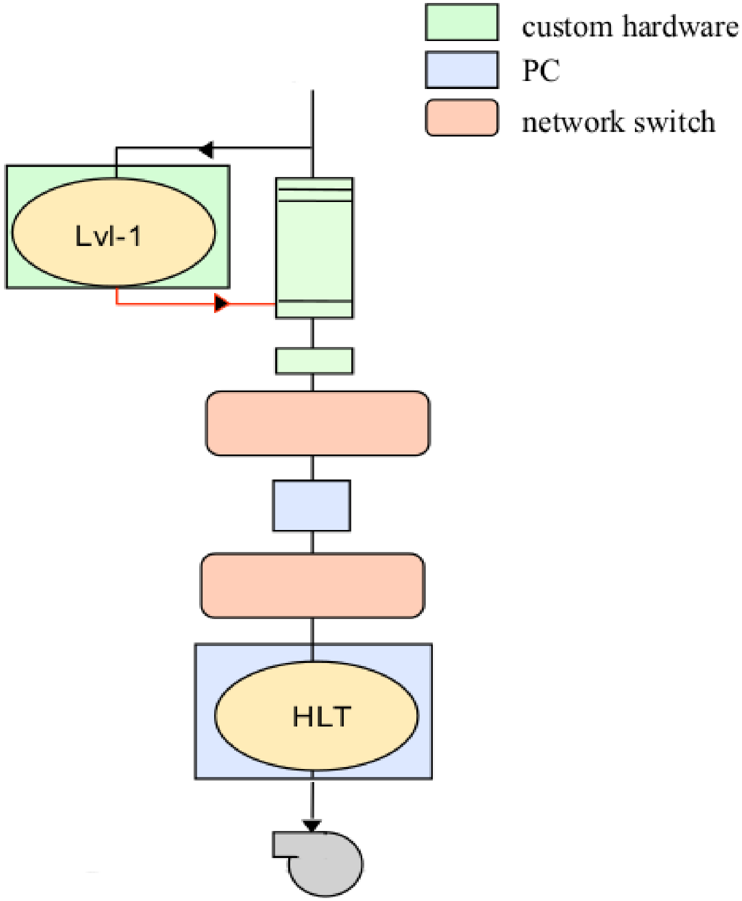
🟩 custom hardware
🟦 PC
🟧 network switch



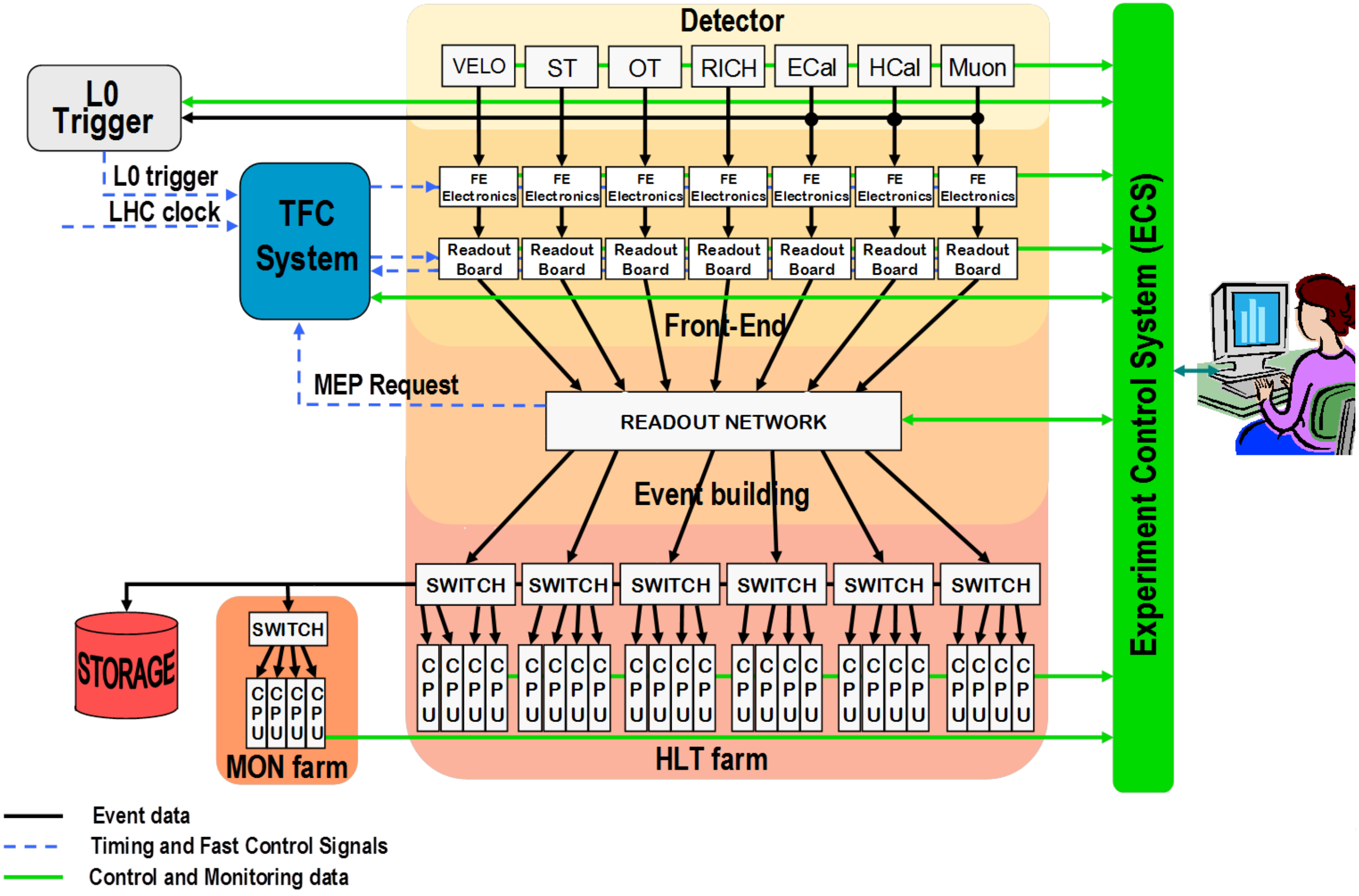


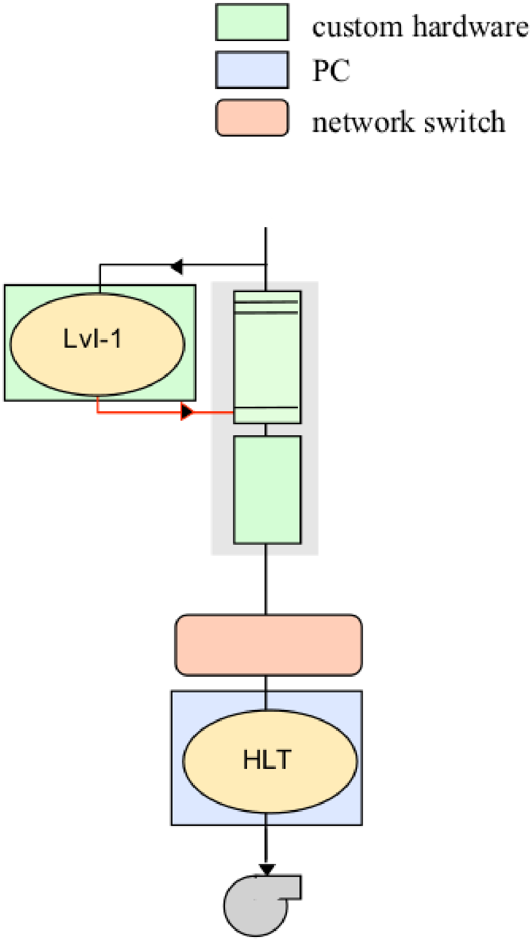
CMS


Global Trigger Processor
FED Builder Switches
RU Builders (1 to 8)
Event Manager
Detector Frontends
512
FED
DAQ links
84
FED Builders
Readout Units
1,64
RU
1,1
Readout Builder Network
Control and Monitor
Builder Units
1,64
BU
1,1
Filter Farm Network
Filter Sub-farms
FU


custom hardware
PC
network switch
Lvl-1
HLT

LHCb


Detector
VELO
ST
OT
RICH
ECal
HCal
Muon
L0 Trigger
L0 trigger
LHC clock
TFC System
FE Electronics
Readout Board
Front-End
MEP Request
READOUT NETWORK
Event building
SWITCH
CPU
STORAGE
MON farm
HLT farm
Experiment Control System (ECS)
Event data
Timing and Fast Control Signals
Control and Monitoring data


custom hardware
PC
network switch
Lvl-1
HLT

Almost The End

What I did not talk about

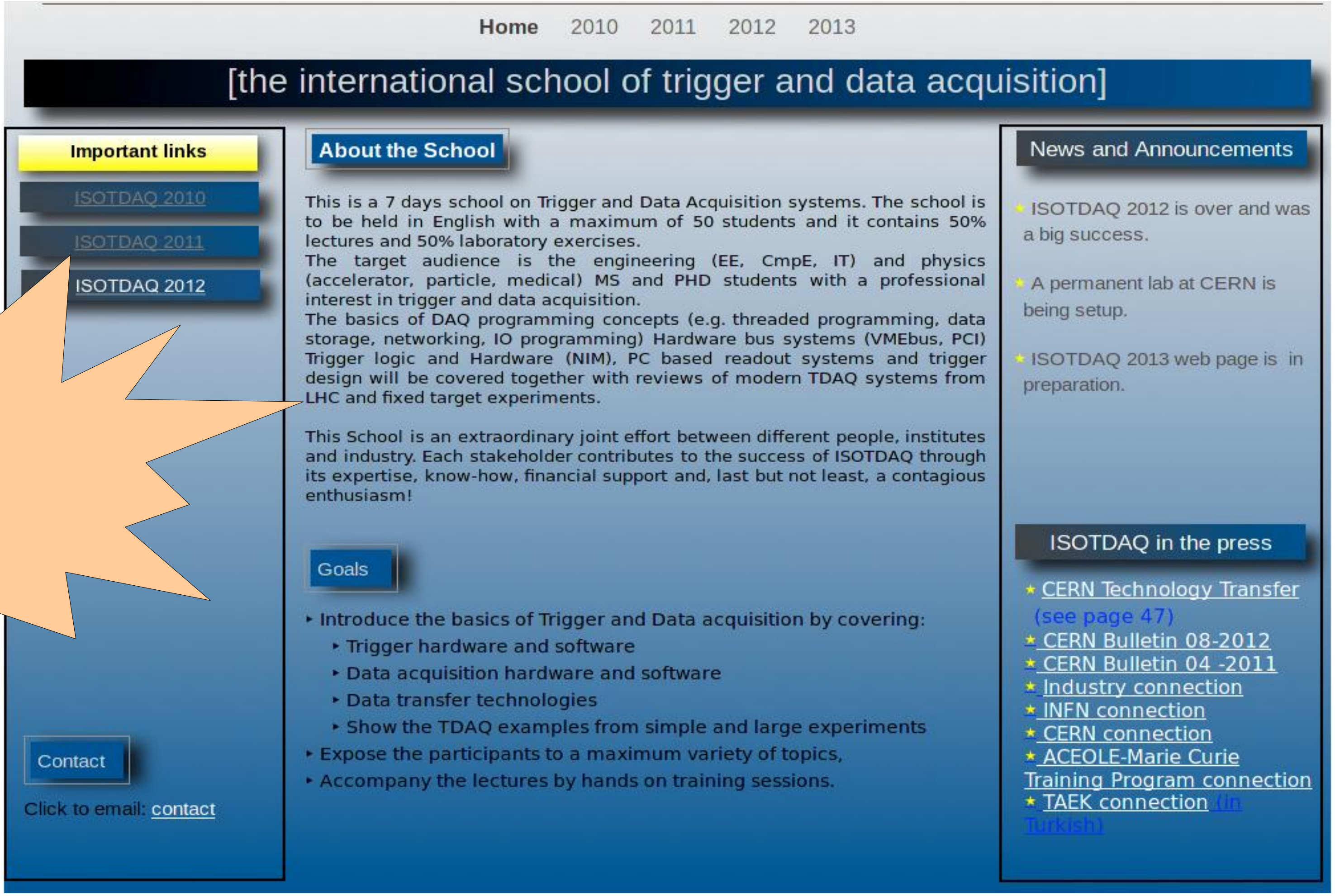
➔ Many many topics

- Run Control → Steering the DAQ, Finite State Machine
- Configuration → Storing, distributing and archiving SW, HW and trigger configuration
- Monitoring → The quality of the data, the state of the detector, the functionality of the DAQ

➔ Your chance of hearing about these and much more and learn through practice …

ISOTDAQ 2013

➔ Fourth edition of the **International School of Trigger and Data Acquisition** will be held in Thessaloniki in January 2013



[the international school of trigger and data acquisition]

About the School

This is a 7 days school on Trigger and Data Acquisition systems. The school is to be held in English with a maximum of 50 students and it contains 50% lectures and 50% laboratory exercises.
The target audience is the engineering (EE, CmpE, IT) and physics (accelerator, particle, medical) MS and PHD students with a professional interest in trigger and data acquisition.
The basics of DAQ programming concepts (e.g. threaded programming, data storage, networking, IO programming) Hardware bus systems (VMEbus, PCI) Trigger logic and Hardware (NIM), PC based readout systems and trigger design will be covered together with reviews of modern TDAQ systems from LHC and fixed target experiments.

This School is an extraordinary joint effort between different people, institutes and industry. Each stakeholder contributes to the success of ISOTDAQ through its expertise, know-how, financial support and, last but not least, a contagious enthusiasm!

Goals

- Introduce the basics of Trigger and Data acquisition by covering:
  - Trigger hardware and software
  - Data acquisition hardware and software
  - Data transfer technologies
  - Show the TDAQ examples from simple and large experiments
- Expose the participants to a maximum variety of topics,
- Accompany the lectures by hands on training sessions.

News and Announcements

- ISOTDAQ 2012 is over and was a big success.
- A permanent lab at CERN is being setup.
- ISOTDAQ 2013 web page is in preparation.

ISOTDAQ in the press

- CERN Technology Transfer (see page 47)
- CERN Bulletin 08-2012
- CERN Bulletin 04 -2011
- Industry connection
- INFN connection
- CERN connection
- ACEOLE-Marie Curie Training Program connection
- TAEK connection (in Turkish)

Contact

Click to email: contact

Watch this space

http://isotdaq.web.cern.ch/isotdaq/isotdaq/Home.html

The End

References

➔ Lectures and papers from H. Spieler

- http://www-physics.lbl.gov/~spieler/

➔ Lecture at ISOTDAQ schools

- http://isotdaq.web.cern.ch/isotdaq/isotdaq/Home.html

➔ Of course, previous Summer Student courses

- http://indico.cern.ch/scripts/SSLPdisplay.py?stdate=2011-07-04&nbweeks=7