Electronics, Trigger, DAQ

CERN Summerstudent Programme 2010

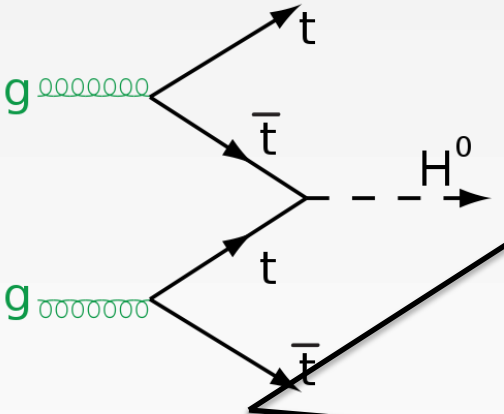
Niko Neufeld, CERN-PH

Contents

- Lecture 1: Mostly electronics, some triggering
- Lecture 2: More electronics, basics of Data Acquisition
- Lecture 3: Data Acquisition 2 and more Trigger
- Lecture 4: Data Acquisition of LHC experiments and running an experiment
- Lecture 5: High Level Trigger

- Topics are related, no 100% separation between the 3
- We want to fill in some of the blanks from W. Riegler's detector lectures to G. Dissertori's data analysis primer
- Wherever possible we will take a "system" perspective: which requirement of our experiment lets us do what? What are the consequences of a certain choice for the detector (the experiment) at large

Physics, Detectors, Electronics Trigger & DAQ



rare, need many collisions

High rate collider

Fast electronics

Big data acquisition

trigger

Disclaimer

- Electronics, Trigger and DAQ are vast subjects covering a lot of physics and engineering
- Based entirely on personal bias I have selected a few topics
- While most of it will be only an overview at a few places we will go into some technical detail
- Some things will be only touched upon or left out altogether – information on those you will find in the references at the end
  - Quantitative treatment of detector electronics & physics behind the electronics
  - Derivation of the "physics" in the trigger → field theory lectures
  - DAQ of experiments outside HEP/LHC
  - Management of large networks and farms &High-speed mass storage

Thanks

- Some material and lots of inspiration for this lecture was taken from lectures by my predecessors: P. Mato, P. Sphicas, J. Christiansen
- In the electronics part I learned a lot from H. Spieler (see refs at the end)
- Trigger material I got from H. Dijkstra and T. Christiansen
- Many thanks to S. Suman for his help with the animations!

Lecture 1/5

Mostly electronics

Electronics in a nutshell

Electronics: introduction

- Why do we care about electronics?
  - As physicists?
  - As computer scientists?
- The Readout Chain
  - Shaping, Amplifying
  - Digitizing, Transmitting, Noise and all that
- Timing and Synchronization
- Systems
  - Power, Cooling & Radiation

Physicists stop reading here

- It is well known that

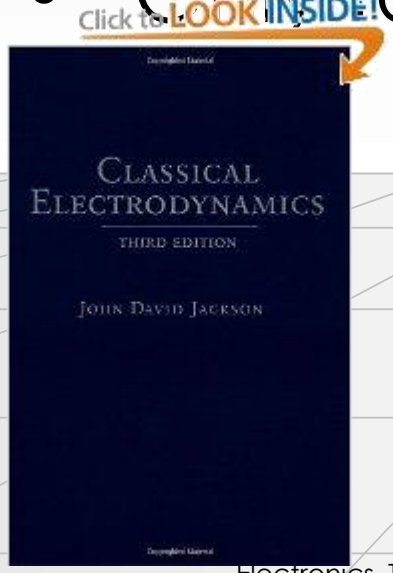
$$d\mathbf{F} = 0$$

$$d\mathbf{G} = \mathbf{J}$$

$$C : \Lambda^2 \ni \mathbf{F} \mapsto \mathbf{G} \in \Lambda^{(4-2)}$$

- "Only technical details are missing"

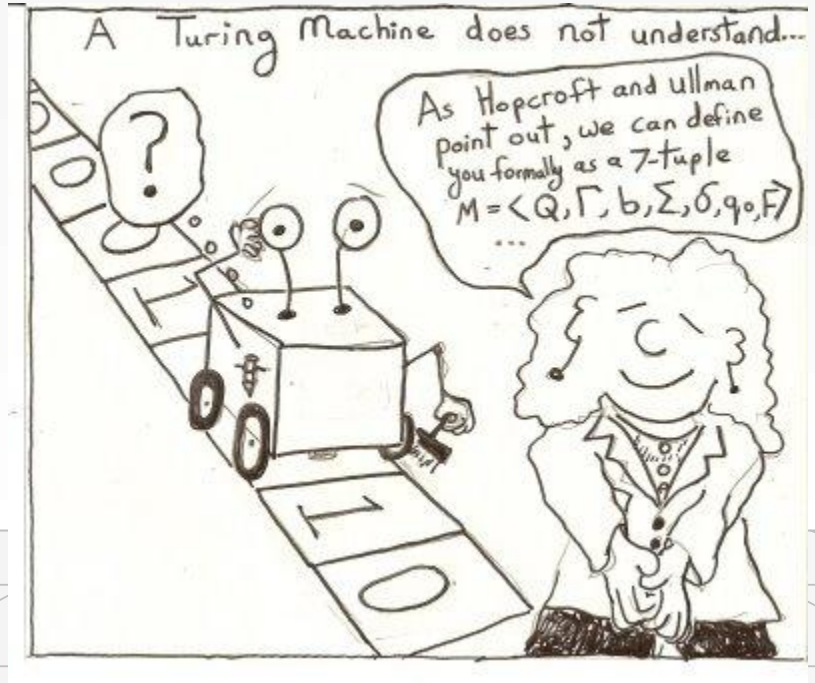
Werner Heisenberg, 1958



A physicist is someone who learned Electrodynamics from Jackson

Computer scientists are digital people

- so why bother with this gruesome analogue electronics stuff



- The problem is that Turing machines are so bad with I/O and it is important to understand the constraints of data acquisition and triggering

The bare minimum

- From Maxwell's equations derive

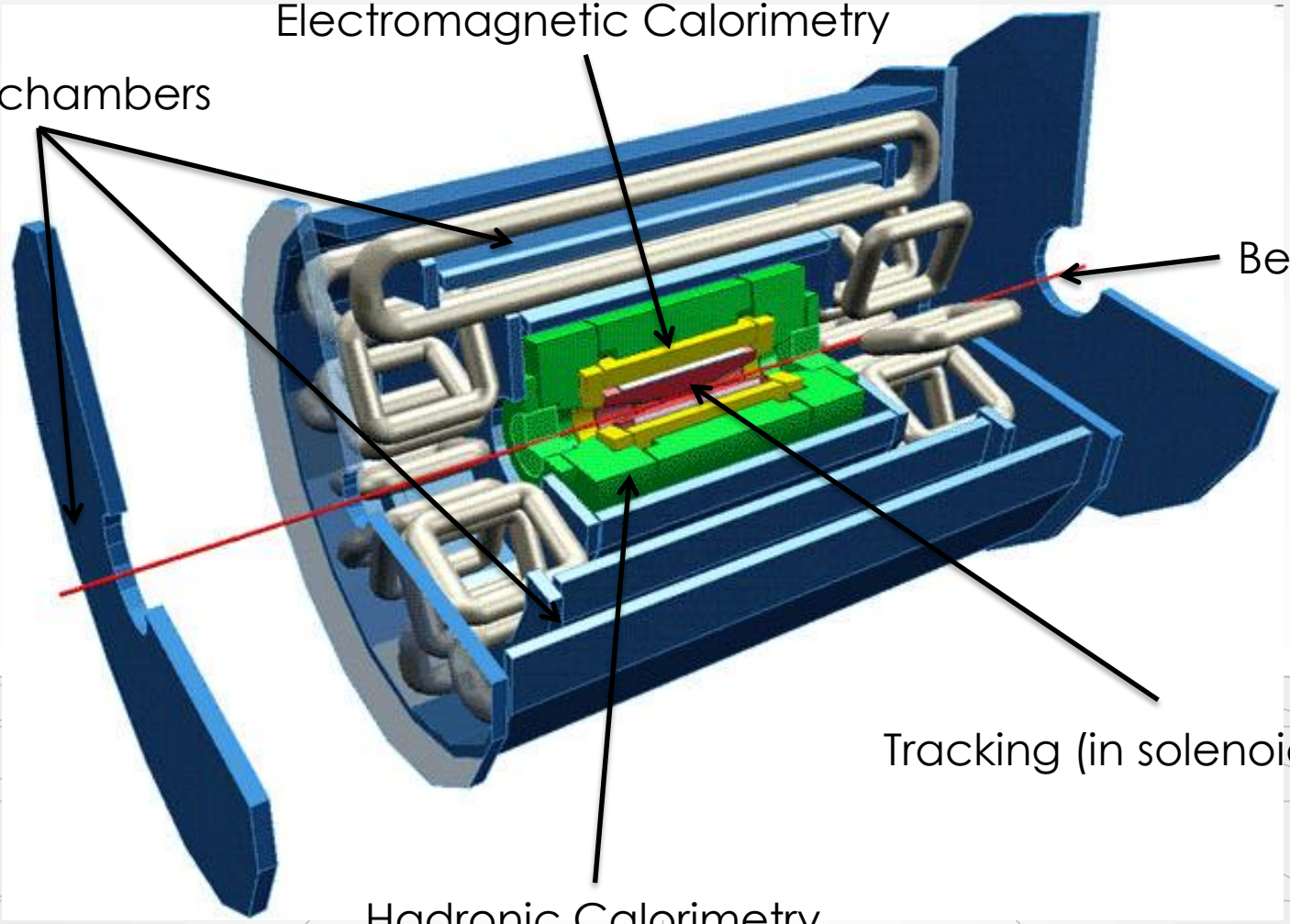
- Ohm's law and power
- The IV characteristics of a capacitance
- Kirchhoff's laws
- where: Q = charge (Coulomb), C = Capacitance (Farad), *U* = *V* = Voltage (Volt), *P* = Power (Watt), *I* = Current (Ampere)

$$I = \frac{U}{R} \qquad P = U \times I$$

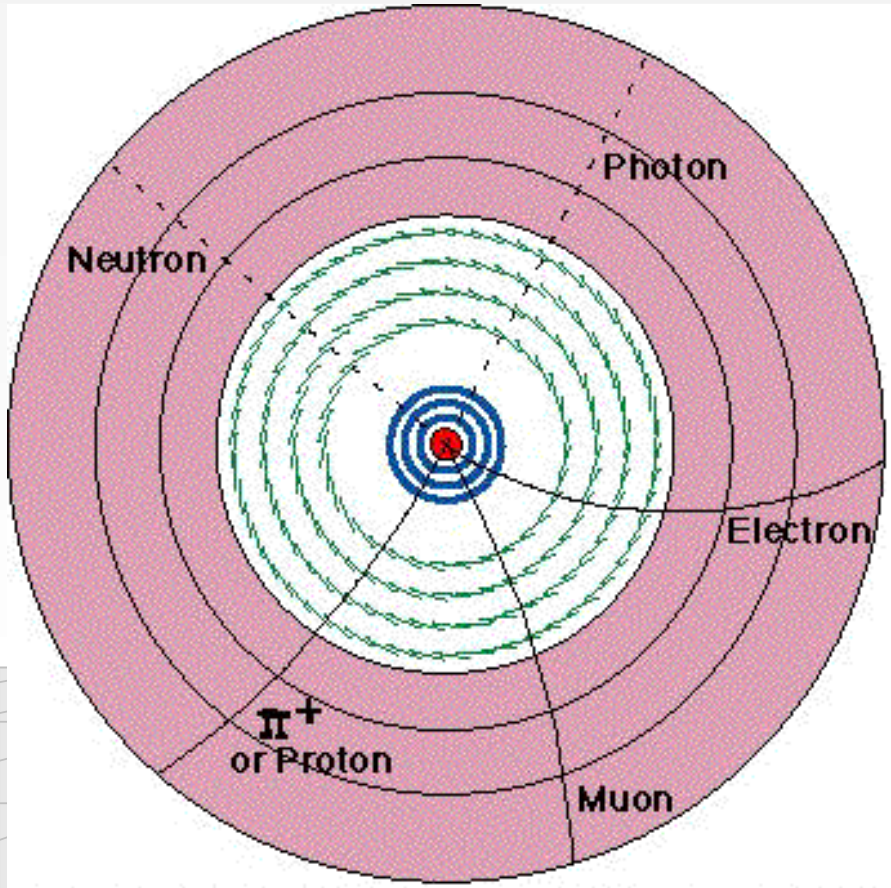
$$Q = C \times V$$

Detector Frontend Electronics (FEE)

Looking at ATLAS

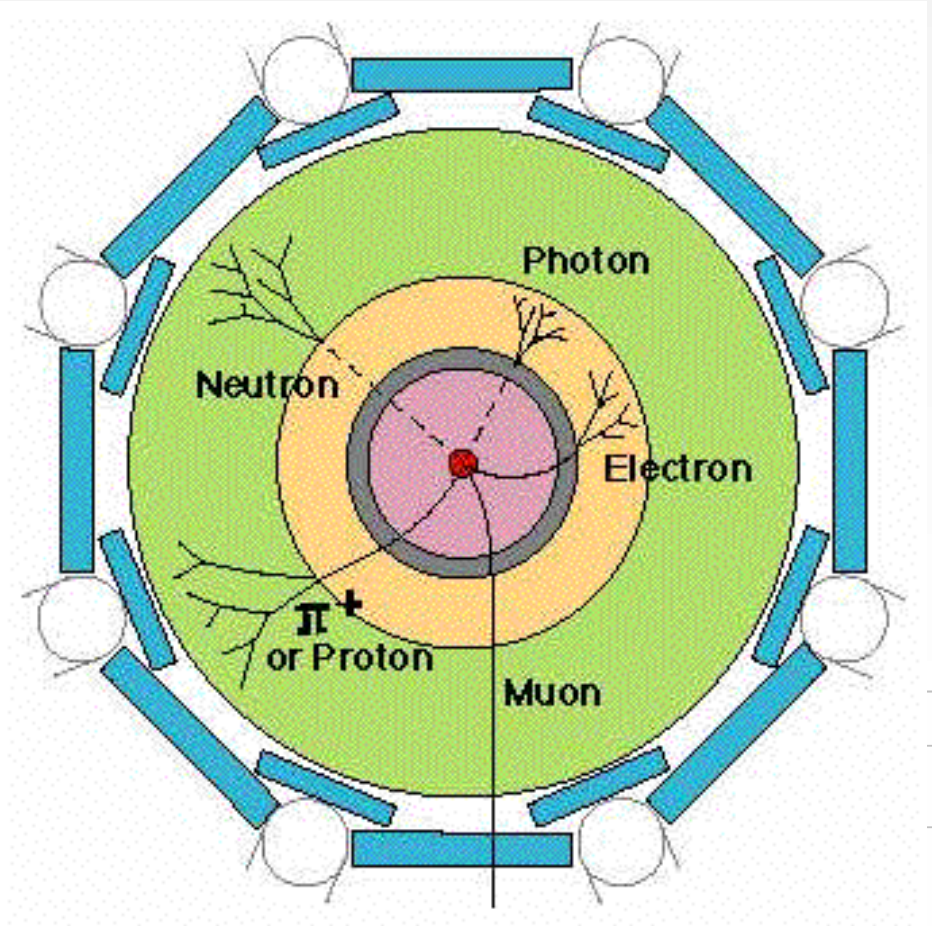


Tracking



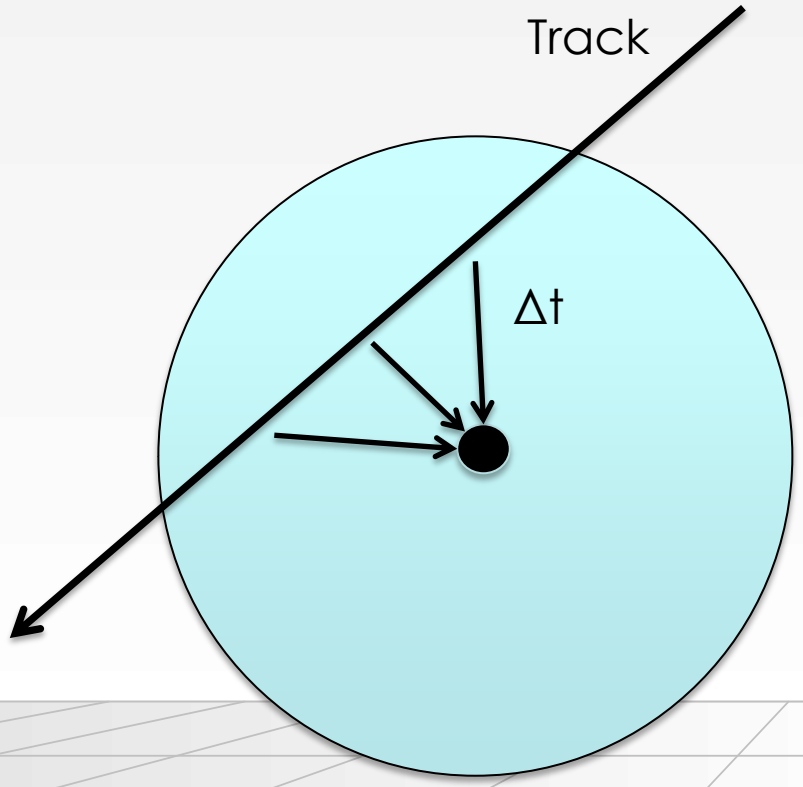
- Separate tracks by charge and momentum
- Position measurement layer by layer
  - Inner layers: silicon pixel and strips → presence of hit determines position
  - Outer layers: "straw" drift chambers → need time of hit to determine position

Calorimetry



- Particles generate showers in calorimeters
  - Electromagnetic Calorimeter (yellow): Absorbs and measures the energies of all electrons, photons
  - Hadronic Calorimeter (green): Absorbs and measures the energies of hadrons, including protons and neutrons, pions and kaons
- amplitude measurement
- position information provided by segmentation of detector

Muon System



- Electrons formed along the track drift towards the central wire.
- The first electron to reach the high-field region initiates the avalanche, which is used to derive the timing pulse.
- Since the initiation of the avalanche is delayed by the transit time of the charge from the track to the wire, the detection time of the avalanche can be used to determine the radial position(*).
- Principle also used in straw tracker – *need fast timing electronics*

ATLAS Muon drift chambers have a radius of 3 cm and are between 1 and 6 m long

(*) Clearly this needs some start of time t=0 (e.g. the beam-crossing

Summary of measurements

- Si Tracking position to ~10 μm accuracy in rφ (through segmentation) timing to 25 ns accuracy to separate bunch crossings
- Straw Tracker position to 170 μm at r > 56 cm
- EM calorimeter energy via LAr ionization chambers position through segmentation
- Hadron calorimeter energy via plastic scintillator tiles position through segmentation
- Muon System signal via ionization chambers position through timing measurement

Although these various detector systems look very different, they all follow the same principles:

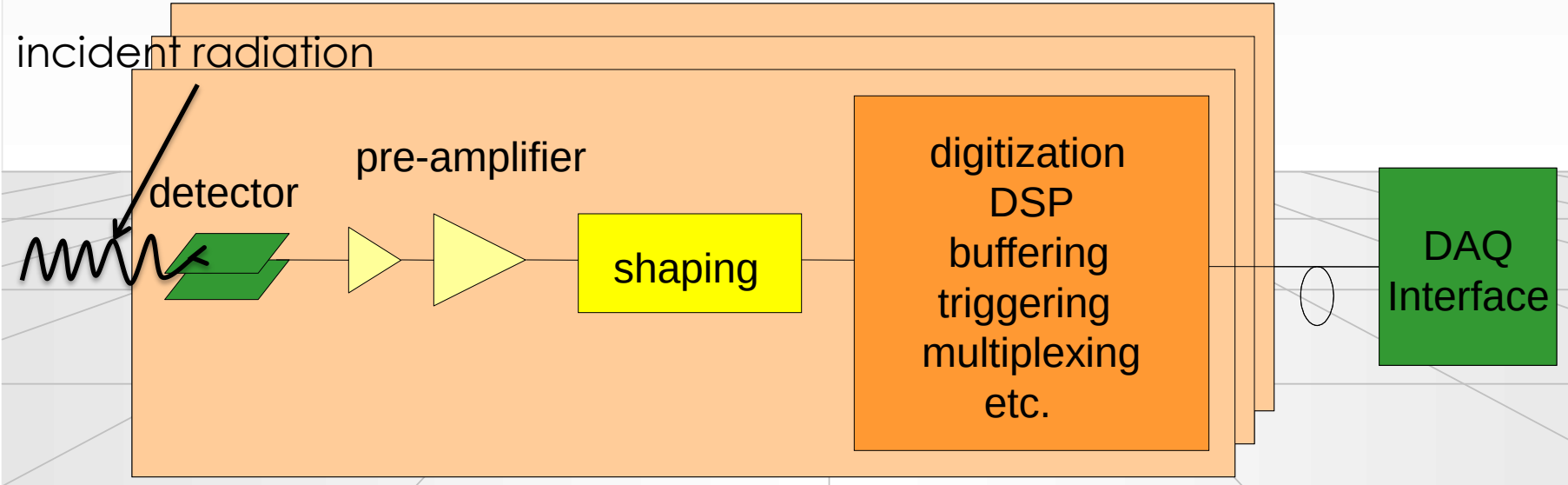
- Sensors must determine
    1. presence of a particle
    2. magnitude of signal
    3. time of arrival
- Some measurements depend on *sensitivity*, i.e. detection threshold, e.g.: silicon tracker, to detect presence of a particle in a given electrode
- Others seek to determine a *quantity very accurately*, i.e. resolution, e.g. : calorimeter – magnitude of absorbed energy; muon chambers – time measurement yields position

All have in common that they are sensitive to:

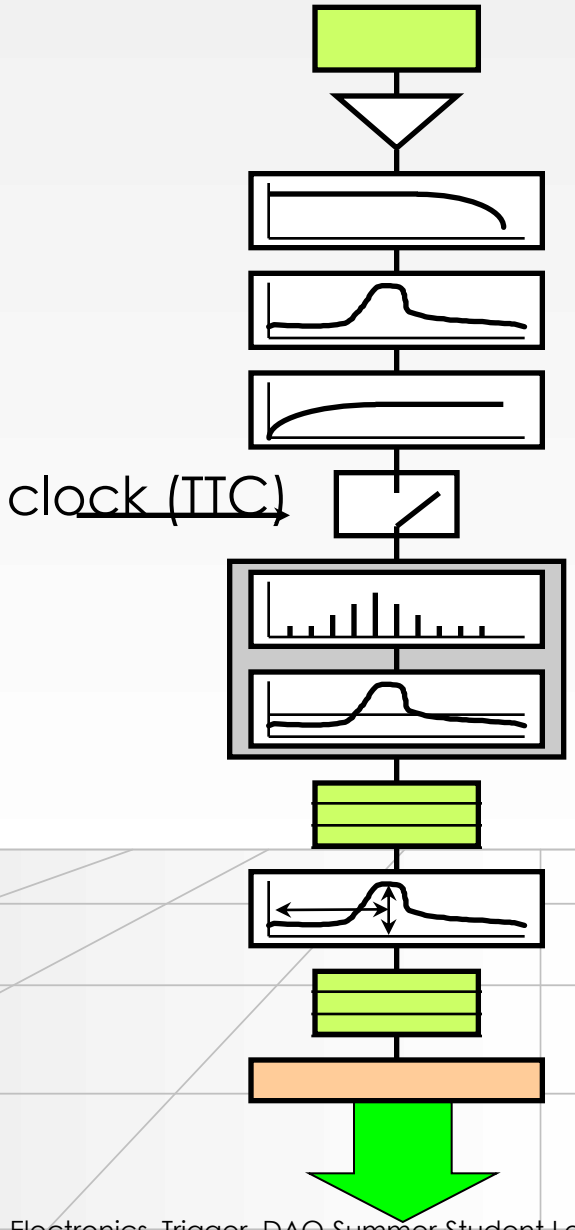
1. signal magnitude
2. fluctuations

The “front-end” electronics`

- Front-end electronics is the electronics directly connected to the detector (sensitive element)
- Its purpose is to
  - acquire an electrical signal from the detector (usually a short, small current pulse)
  - tailor the response of the system to optimize
    - the minimum detectable signal
    - energy measurement (charge deposit)
    - event rate
    - time of arrival
    - insensitivty to sensor pulse shape
  - digitize the signal and store it for further treatment



The read-out chain



Detector / Sensor

Amplifier

Filter

Shaper

Range compression

Sampling

Digital filter

Zero suppression

Buffer

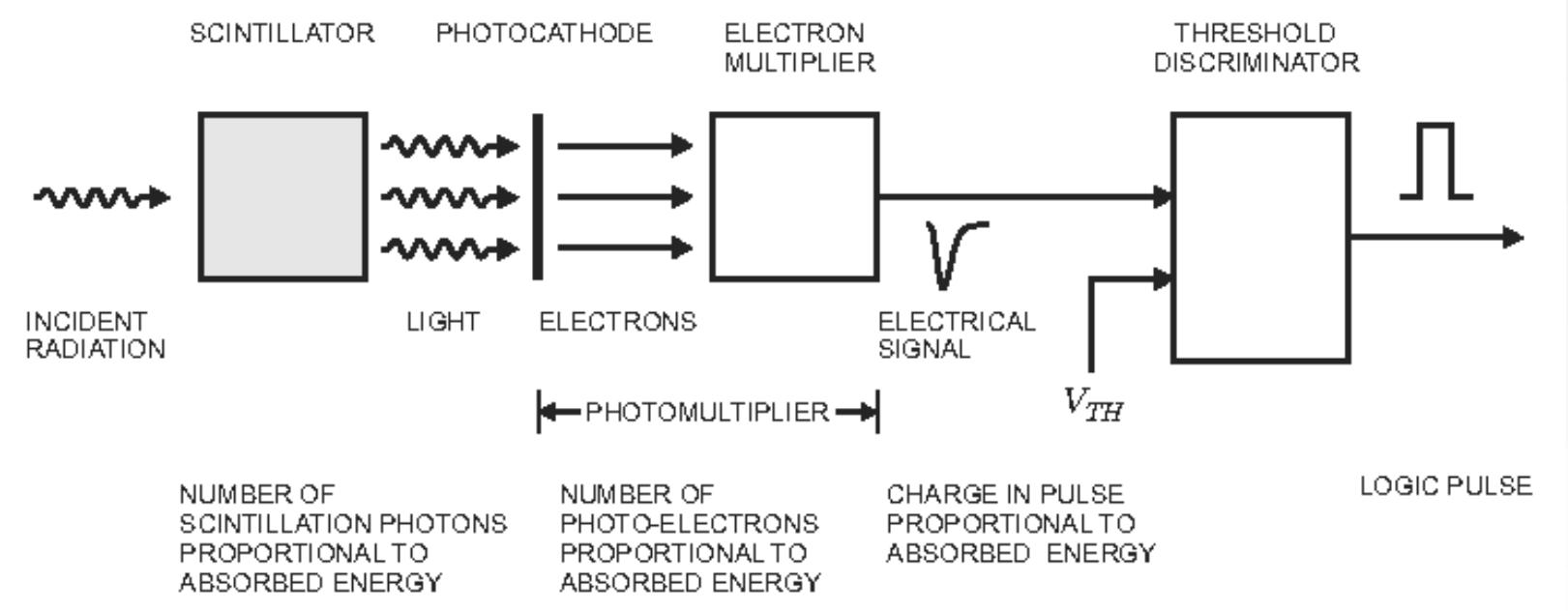
Feature extraction

Buffer

Format & Readout

to Data Acquisition System

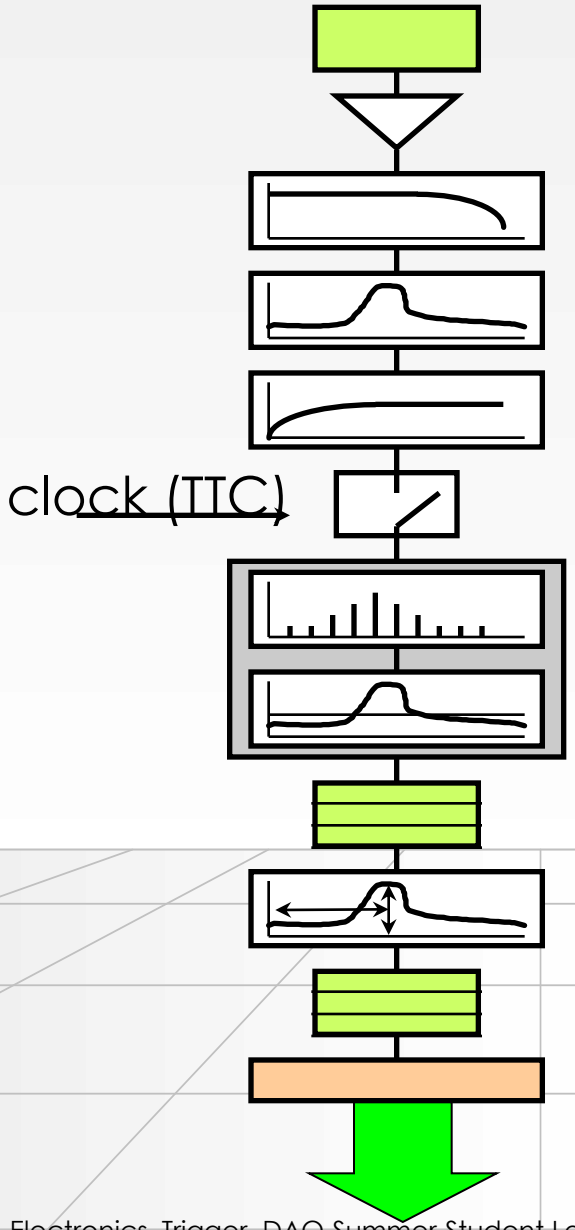
Example: Scintillator



from H. Spieler "Analog and Digital Electronics for Detectors"

- Photomultiplier has high intrinsic gain (== amplification) → no pre-amplifier required
- Pulse shape does not depend on signal charge → measurement is called *pulse height analysis*

The read-out chain



Detector / Sensor

Amplifier

Filter

Shaper

Range compression

Sampling

Digital filter

Zero suppression

Buffer

Feature extraction

Buffer

Format & Readout

to Data Acquisition System

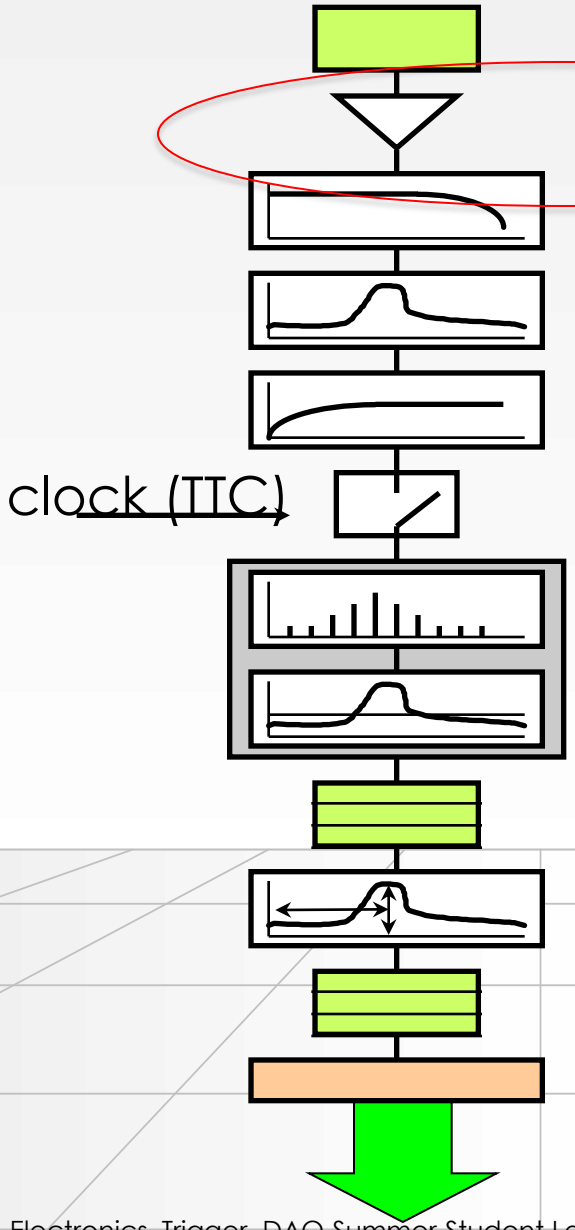
The signal

- The signal is usually a small current pulse varying in duration (from ~ 100 ps for a Si sensor to O(10) μs for inorganic scintillators)
- There are many sources of signals. Magnitude of signal depends on deposited signal (energy / charge) and excitation energy

$$S = \frac{E_{absorbed}}{E_{excitation}}$$

Signal	Physical effect	Excitation energy
Electrical pulse (direct)	Ionization	30 eV for gases 1- 10 eV for semiconductors
Scintillation light	Excitation of optical states	20 – 500 eV
Temperature	Excitation of lattice vibrations	meV

The read-out chain



Detector / Sensor

Amplifier

Filter

Shaper

Range compression

Sampling

Digital filter

Zero suppression

Buffer

Feature extraction

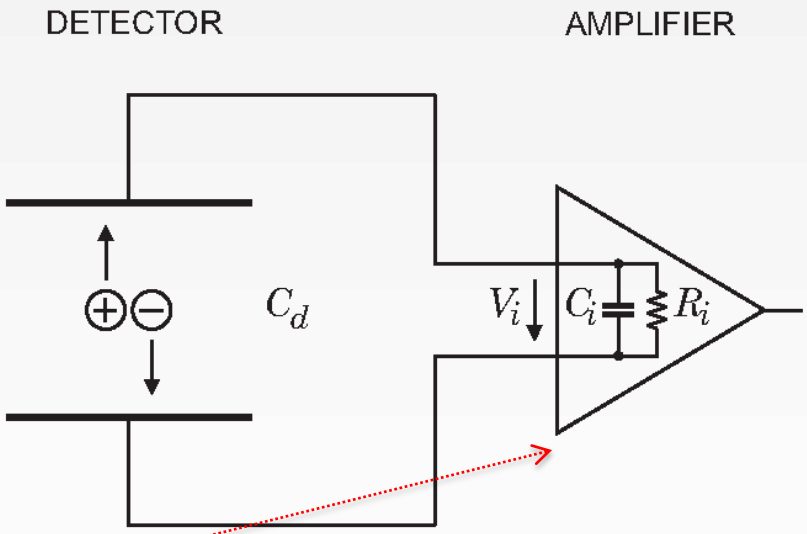
Buffer

Format & Readout

to Data Acquisition System

Acquiring a signal

- *Interesting signal is the deposited energy* ➔ need to integrate the current pulse
  - on the sensor capacitance
  - using an integrating pre-amplifier
  - using an integrating Analog Digital Converter (ADC)
- The signal is usually very small ➔ need to amplify it
  - with electronics
  - by signal multiplication (e.g. photomultiplier)

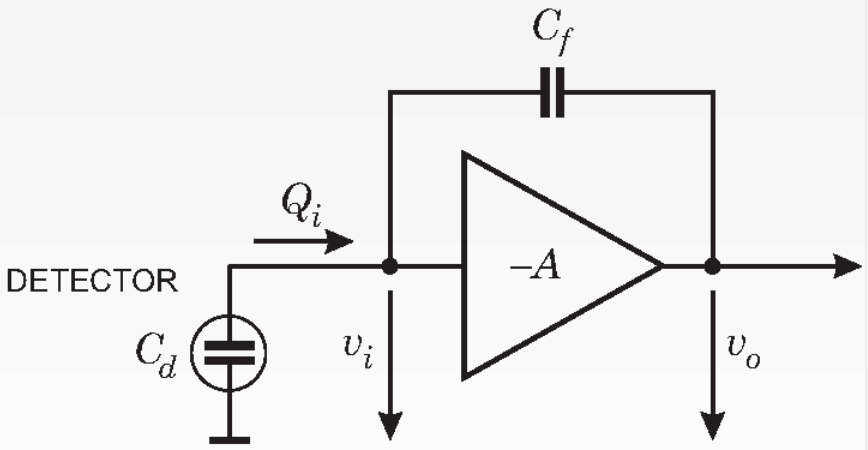


$$V_i = \frac{Q_S}{C_d + C_i}$$

Not so practical! Response depends on sensor capacitance

Charge sensitive amplification

- Feedback amplifier with gain –A
- Assume infinite input impedance (no current flows into the amplifier)
- Input signal produces v_i at the input of the amplifier generating $-Av_i$ on output
- All charge must build up on feed back capacitance



- Charge gain depends only on C_f
- C_f x A needs to be large compared to C_d

Fluctuations and Noise

- There are two limitations to the precision of signal magnitude measurements
  1. Fluctuations of the signal charge due to a an absorption event in the detector
  2. Baseline fluctuations in the electronics ("noise")
- Often one has both – they are independent from each other so their contributions add in quadrature:

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

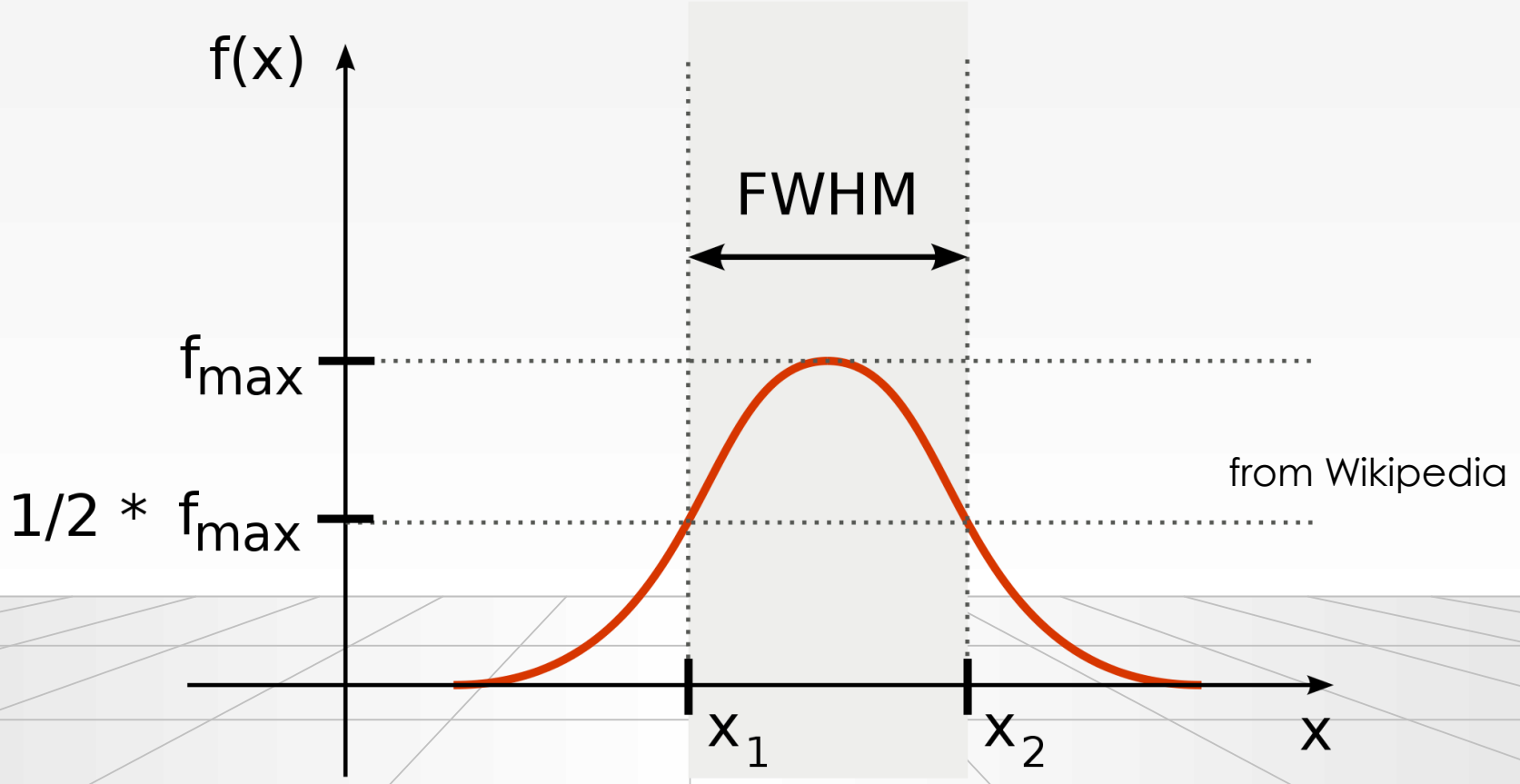
- Noise affects all measurements – must **maximize signal to noise ration S/N ratio**

Signal fluctuation

- A signal consists of multiple elementary events (e.g. a charged particle creates one electron-hole pair in a Si-strip)
- The number of elementary events fluctuates $\Delta N = \sqrt{FN}$ where F is the Fano factor (0.1 for Silicon)
- $\Delta E = E_i \Delta N = \sqrt{FEE_i}$ r.m.s.

$$\Delta E_{FWHM} = 2.35 \times \Delta E_{rms}$$

Full Width at Half Maximum (FWHM)

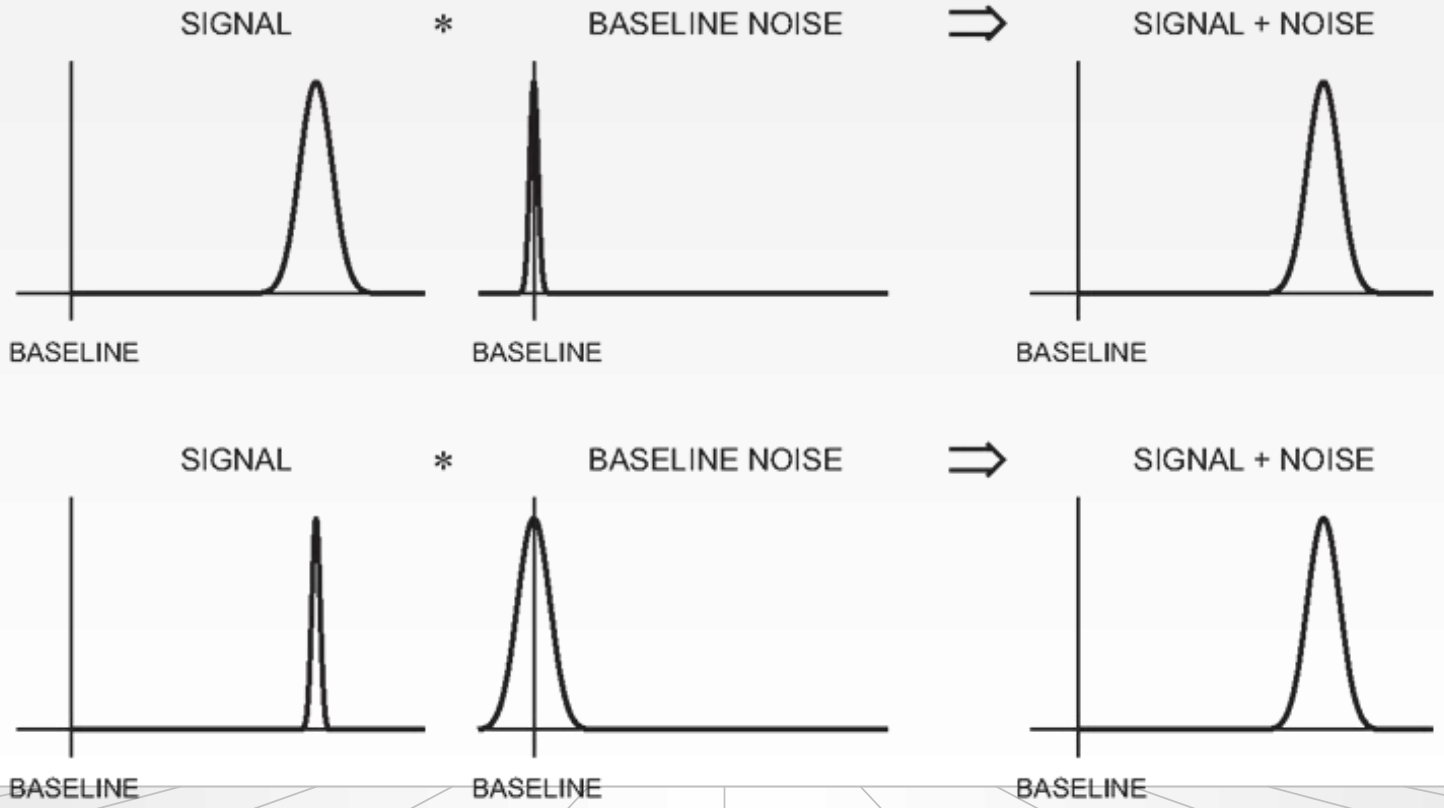


FWHM = 2.35 σ for a Gaussian distribution

Electronics Noise

- Thermal noise
  - created by velocity fluctuations of charge carriers in a conductor
  - Noise power density per unit bandwidth is constant: white noise ➔ larger bandwidth → larger noise (see also next slide)
- Shot noise
  - created by fluctuations in the number of charge carriers (e.g. tunneling events in a semi-conductor diode)
  - proportional to the total average current

SNR / Signal over Noise



Need to optimize Signal over Noise Ratio (SNR)

Two important concepts

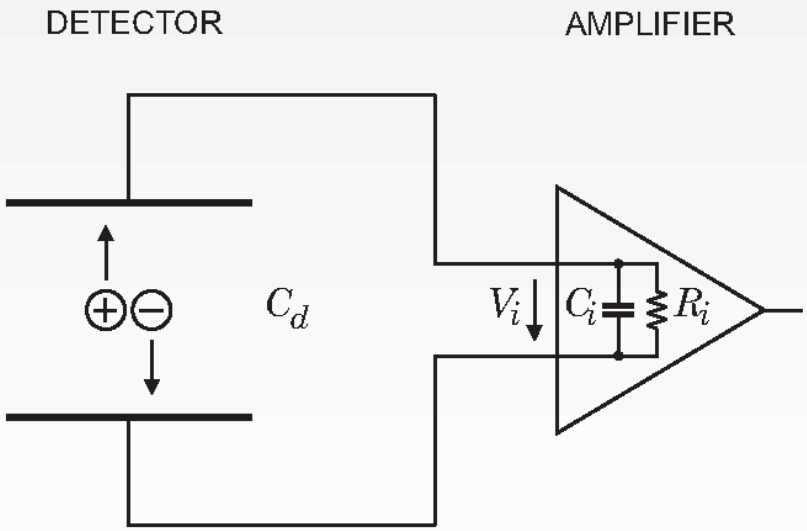
- The *bandwidth BW* of an amplifier is the frequency range for which the output is at least half of the nominal amplification
- The *rise-time* t_r of a signal is the time in which a signal goes from 10% to 90% of its peak-value
- For a linear RC element (amplifier):

$$\mathbf{BW * t_r = 0.35}$$

- For fast rising signals (t_r small) need high bandwidth, but this will increase the noise (see before) → shape the pulse to make it "flatter"

SNR and detector capacitance

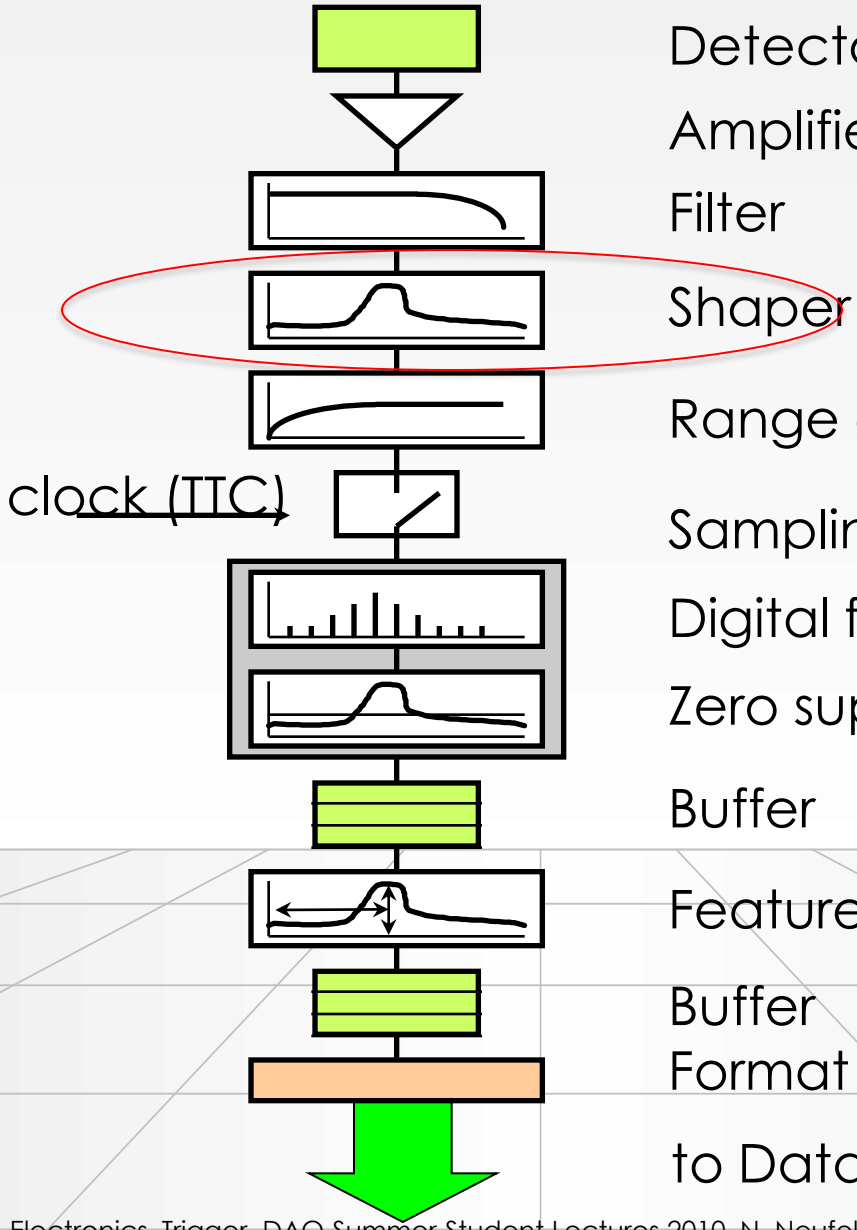
- For a given signal charge Q_s: $V_s - Q_s/(C_d + C_i)$
- Assume amplifier has an input noise voltage V_n, then



- $\mathrm{SNR} = \frac{V_s}{V_n} = \frac{Q_s}{V_n \times (C_n + C_d)}$

SNR is inversely proportional to total capacitance on the input → thicker sensor gives more signal but also more noise

The read-out chain



Detector / Sensor

Amplifier

Filter

Shaper

Range compression

Sampling

Digital filter

Zero suppression

Buffer

Feature extraction

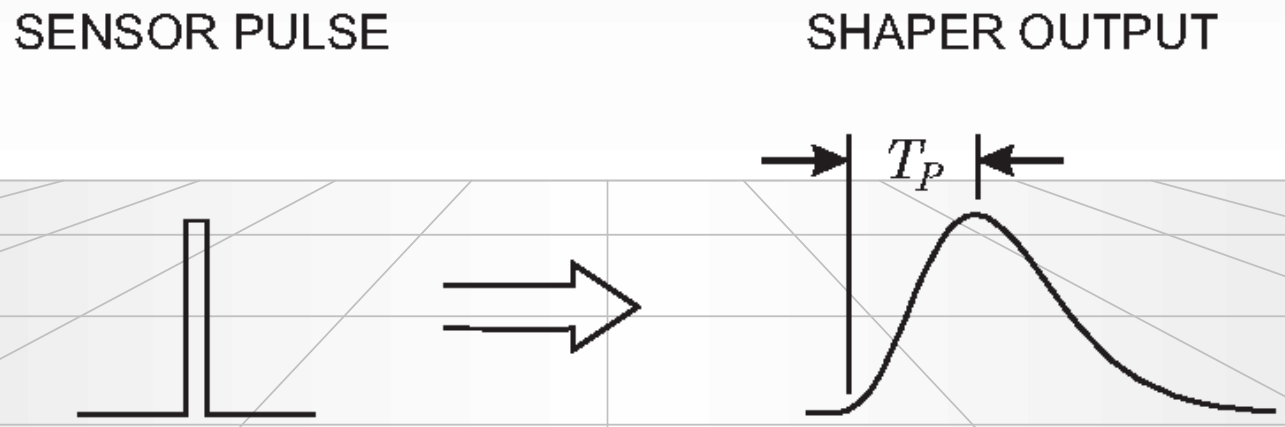
Buffer

Format & Readout

to Data Acquisition System

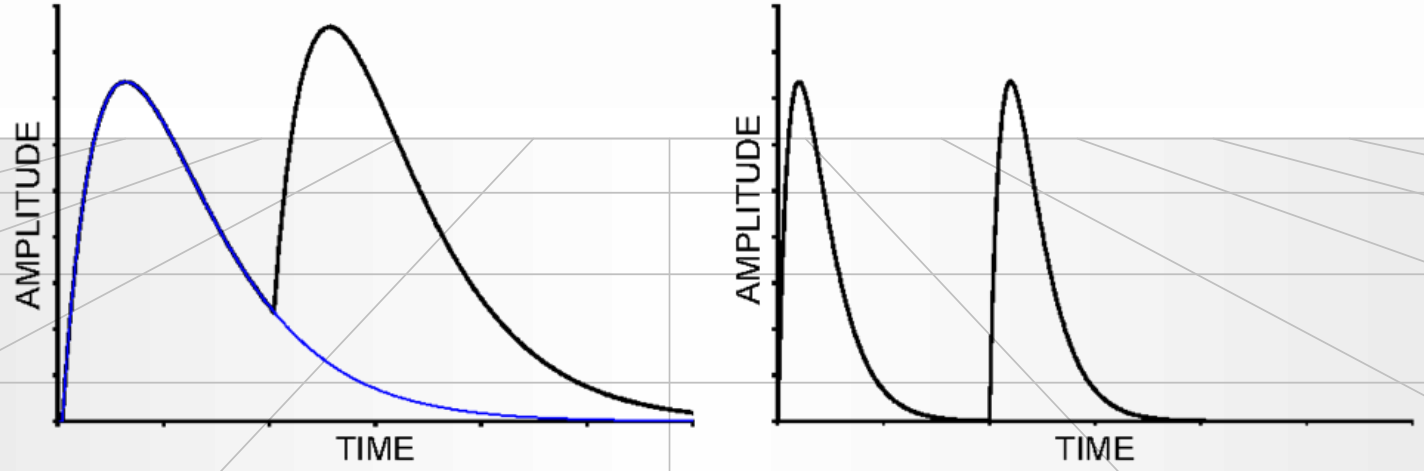
The pulse-shaper should "broaden"…

- Sharp pulse is "broadened" – rounded around the peak
- Reduces input bandwidth and hence noise



…but not too much

- Broad pulses reduce the temporal spacing between consecutive pulses
- Need to limit the effect of "pile-up" → pulses not too borad
- As usual in life: a compromise, in this case made out of RC and CR filters



Lecture 2/5

More electronics & trigger

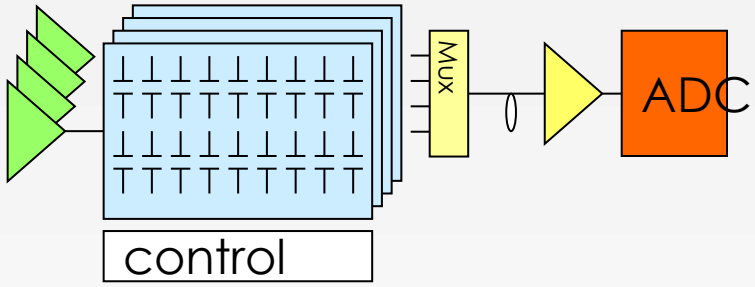
Analog/Digital/binary

After amplification and shaping the signals must at some point be digitized to allow for DAQ and further processing by computers

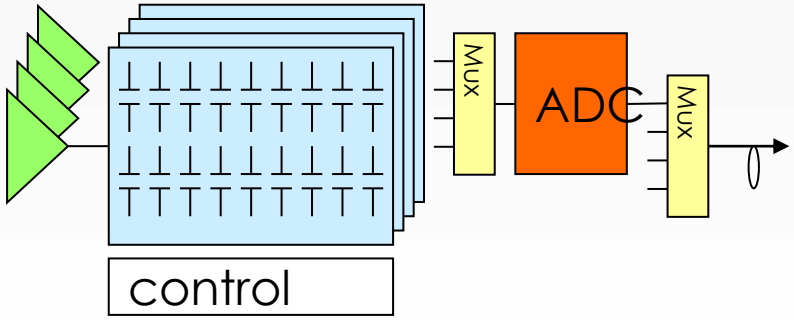
1. Analog readout: analog buffering ; digitization after transmission off detector
2. Digital readout with analog buffer
3. Digital readout with digital buffer

- *Binary*: discriminator right after shaping
  - Binary tracking
  - Drift time measurement

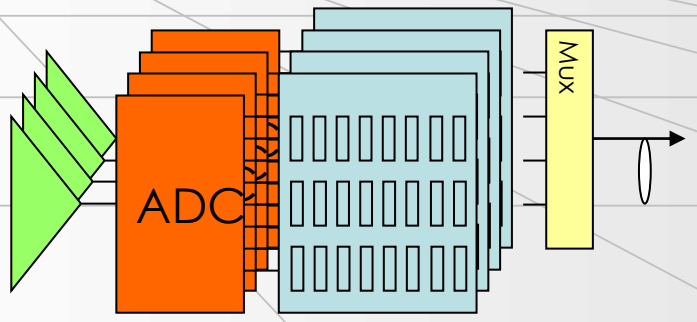
1) Analog memory



2) Analog memory

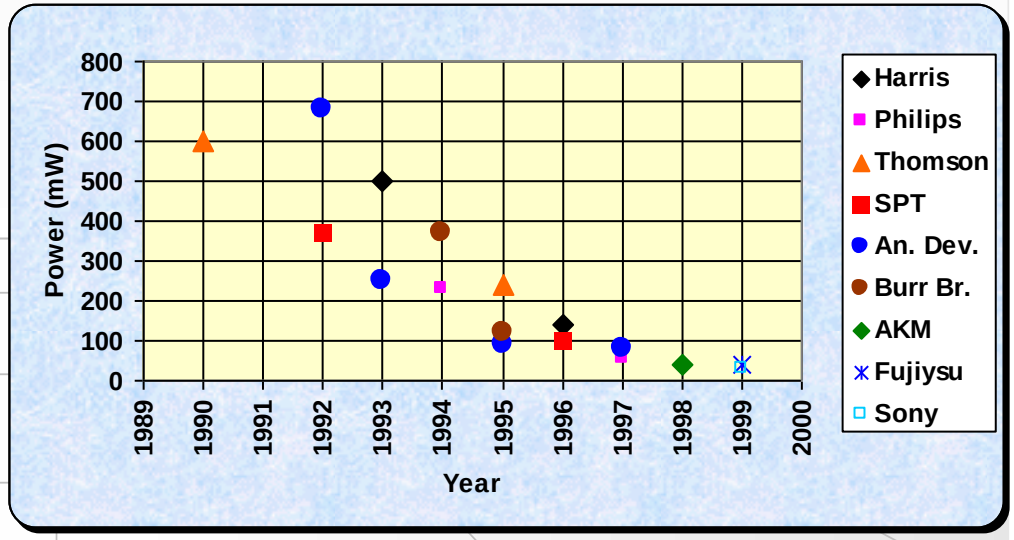


3) Digital memory



Analog to digital conversion

- There is clearly a tendency to go digital as early as possible
  - This is extensively done in consumer goods
- The "cost" of the ADC determines which architecture is chosen
  - Strongly depends on speed and resolution
- Input frequencies must be limited to half the sampling frequency.
  - Otherwise this will fold in as additional noise.
- High resolution ADC also needs low jitter clock to maintain effective resolution

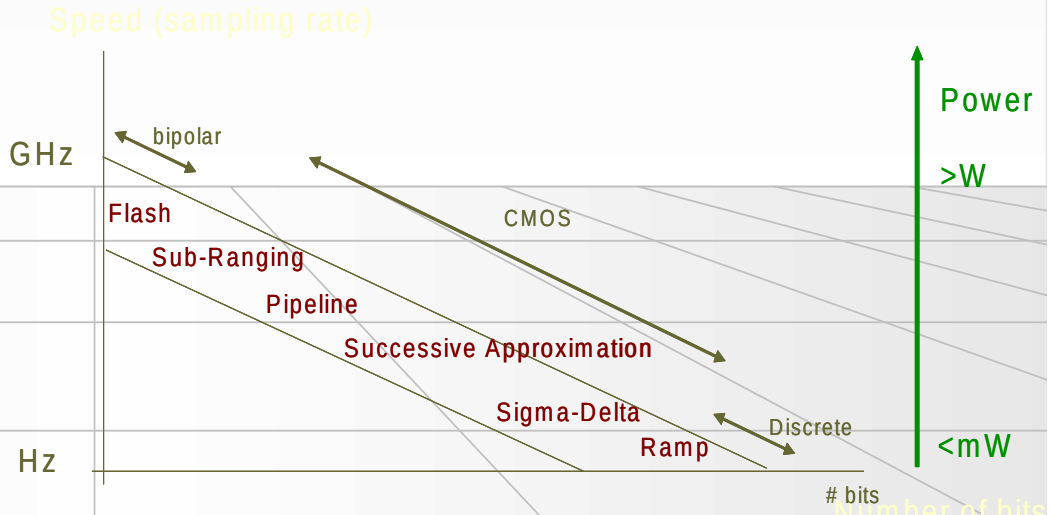


An important truth

- A solution in detector-electronics can be:
  1. fast
  2. cheap
  3. low-power
- *Choose two of the above*: you can't have three

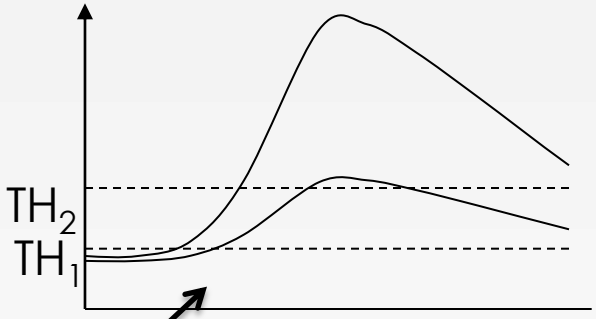
Cost means:

Power consumption
Silicon area
Availability of radiation hard ADC

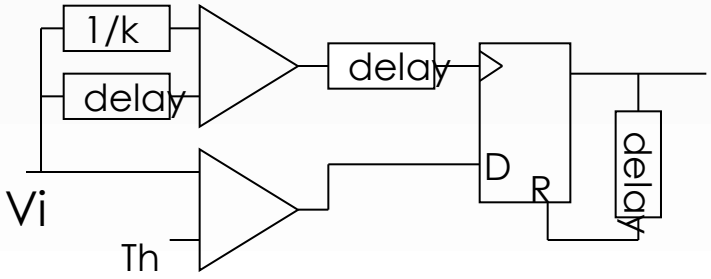


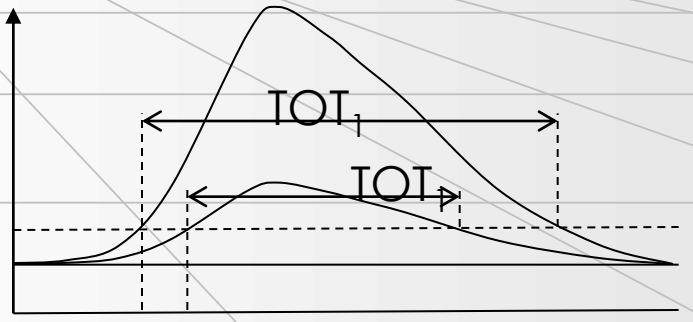
Time measurements

- Time measurements are important in many HEP applications
  - Identification of bunch crossing (LHC: 25ns)
  - Distinguishing among individual collisions (events) in continuous beam like experiments (or very short bunch interval like CLIC: ~250ps)
  - Drift time
    - Position in drift tubes (binary detectors with limited time resolution: ~1ns)
    - Time projection chamber (both good time and amplitude)
    - Time Of Flight (TOF) detectors (very high time resolution: 10-100ps)
- Time walk: Time dependency on amplitude
  - Low threshold (noise and pedestal limited)
  - Constant fraction discrimination
    - Works quite well but needs good analog delays (cable delay) which is not easy to integrate on chip.
  - Amplitude compensation (done in DAQ CPU's)
    - Separate measurement of amplitude (expensive)
    - Time measurements with two thresholds: 2 TDC channels
    - Time over threshold (TOT): 1 TDC channel measuring both leading edge and pulse width
- Time Over Threshold (TOT) can even be used as a poor mans ADC
  - E.g. ATLAS Pixel



Constant fraction discriminator

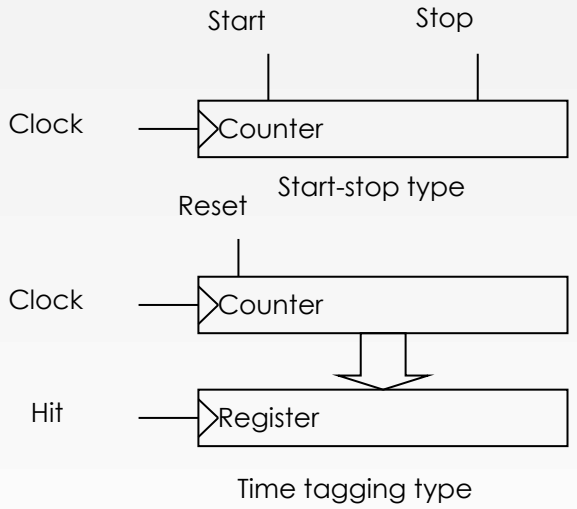


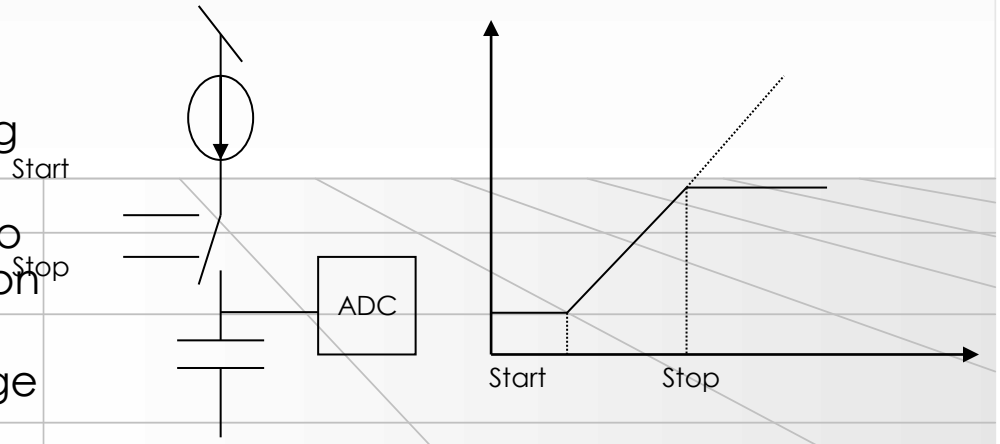


$T = T_r - f(TOT)$

Time to digital conversion

- Counter
  - Large dynamic range
  - Good and cheap time references available as crystal oscillators
  - Synchronous to system clock so good for time tagging
  - Limited resolution: ~1ns
- Charge integration (start – stop)
  - Limited dynamic range
  - High resolution: ~1-100 ps
  - Sensitive analog circuit needing ADC for final conversion.
  - Sensitive to temperature, etc. so often needs in-system calibration
  - Can be combined with time counter for large dynamic range





Readout

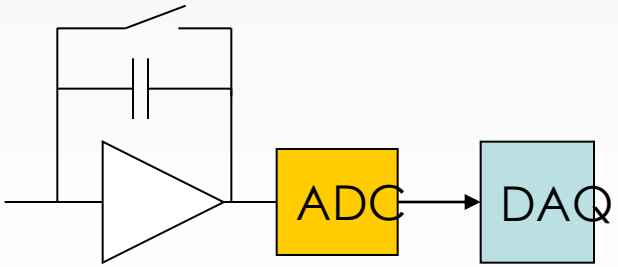
After shaping and amplifying

As usual ☺ what you do depends on many factors:

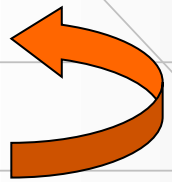
- Number of channels and channel density
- Collision rate and channel occupancies
- *Triggering:* levels, latencies, rates
- Available technology and cost
- What you can/want to do in custom made electronics and what you do in standard computers (computer farms)
- Radiation levels
- Power consumption and related cooling
- Location of digitization
- Given detector technology

Single integrator

- Simple (only one sample per channel)
- Slow rate (and high precision) experiments
- Long dead time
- Nuclear physics
- Not appropriate for HEP

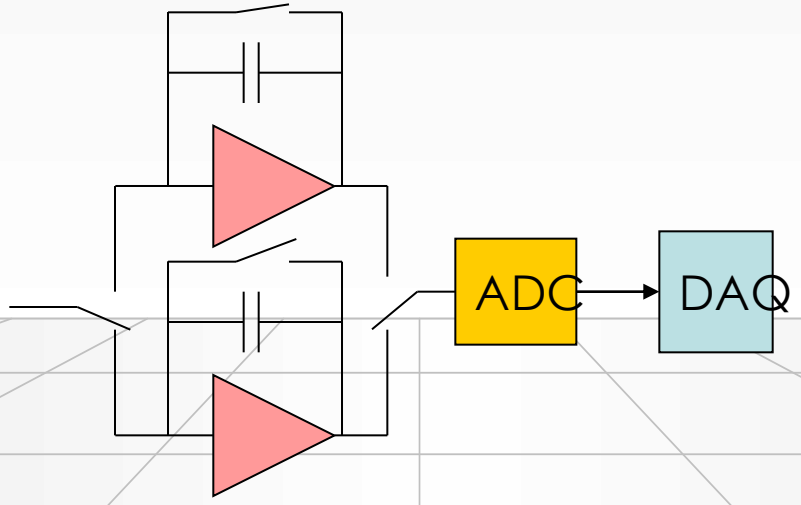


1. Collect charge from event
2. Convert with ADC
3. Send data to DAQ



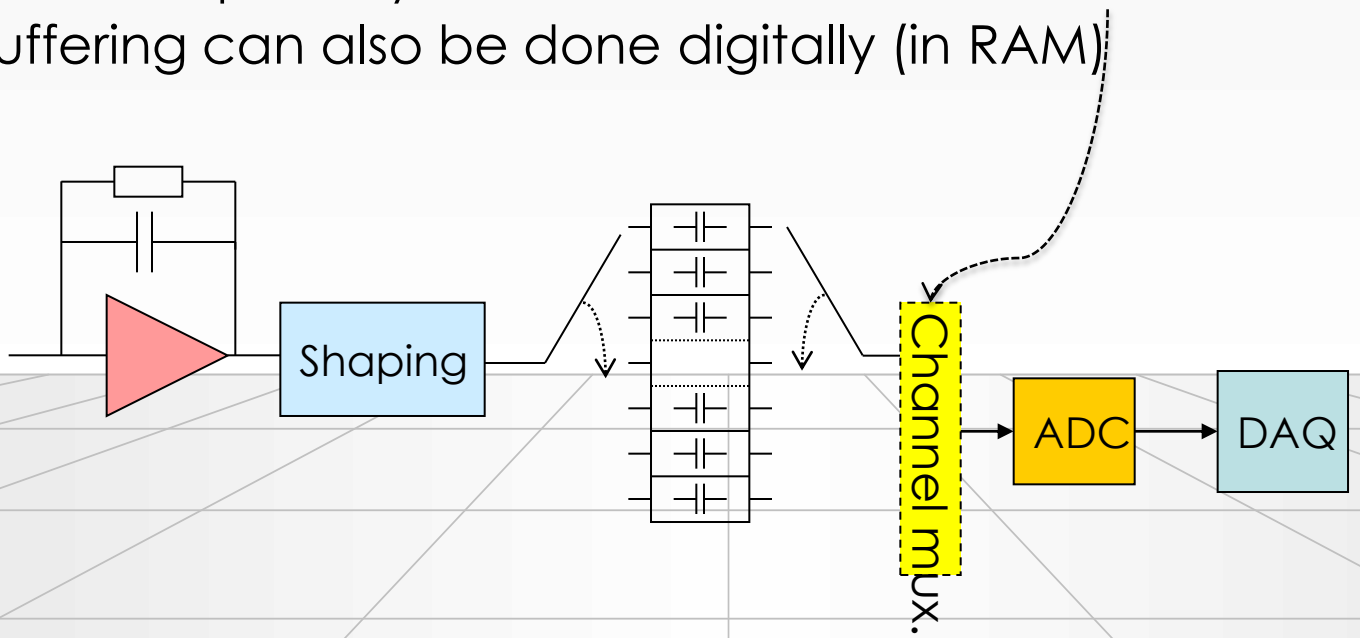
Double buffered

- Use a second integrator while the first is readout and reset
- Decreases dead time significantly
- Still for low rates



Multiple event buffers

- Good for experiments with short spills and large spacing between spills (e.g. fixed target experiment at SPS)
- Fill up event buffers during spill (high rate)
- Readout between spills (low rate)
- ADC can possibly be shared across channels
- Buffering can also be done digitally (in RAM)



Analog buffers

- Extensively used when ADC not available with sufficient speed and resolution or consuming too much power
- Large array of storage capacitors with read and write switches (controlled digitally)
- For good homogeneity of memory
  - Voltage mode
  - Charge mode with Charge integrator for reading
- Examples:
  - Sampling oscilloscopes
  - HEP: CMS tracker, ATLAS calorimeter, LHCb trackers, etc.

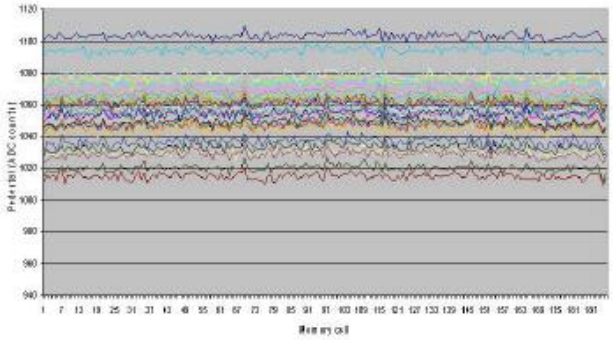


Fig. 9 Pedestals for each memory cell in the analog memory. All 32 channels plotted for each of the 192 columns. This plot is of a packaged PACE3 device. 1 ADC count = 0.435mV.

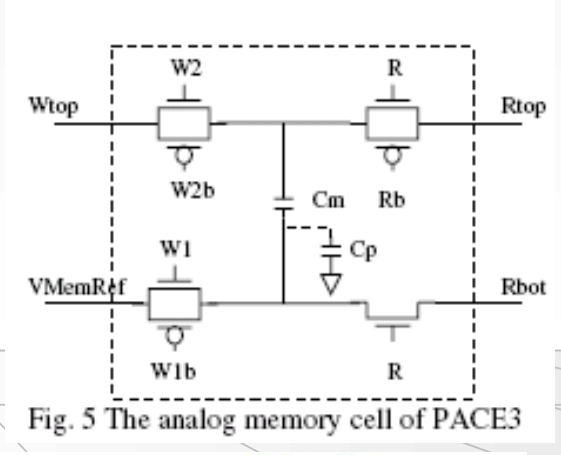
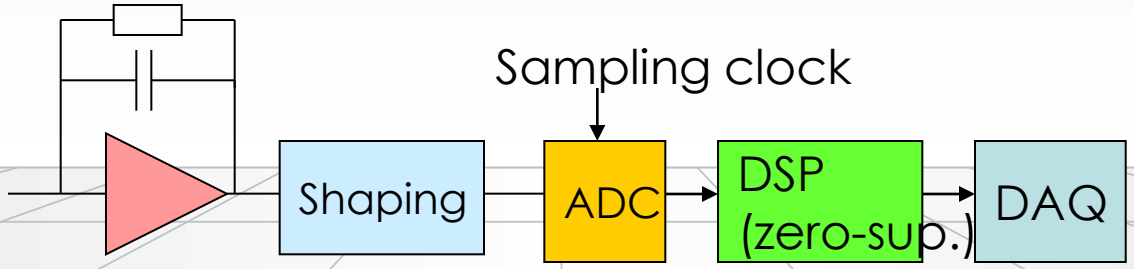


Fig. 5 The analog memory cell of PACE3

Constantly sampled

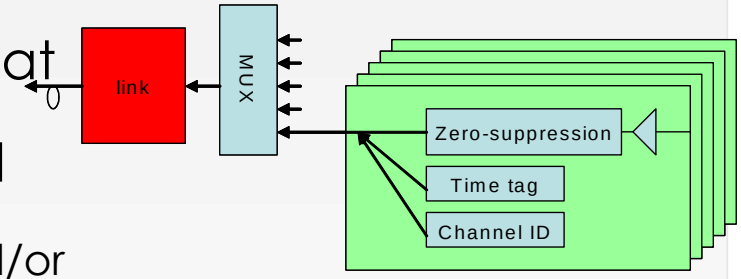
- Needed for high rate experiments with signal pileup
- Shapers and not switched integrators
- Allows digital signal processing in its traditional form (constantly sampled data stream)
- Output rate may be far to high for what following DAQ system can handle

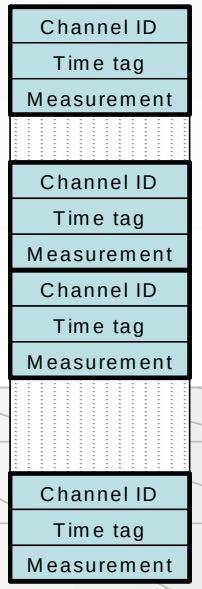


- With local zero-suppression this may be an option for future high rate experiments (SLHC, CLIC)

Excursion: zero-suppression

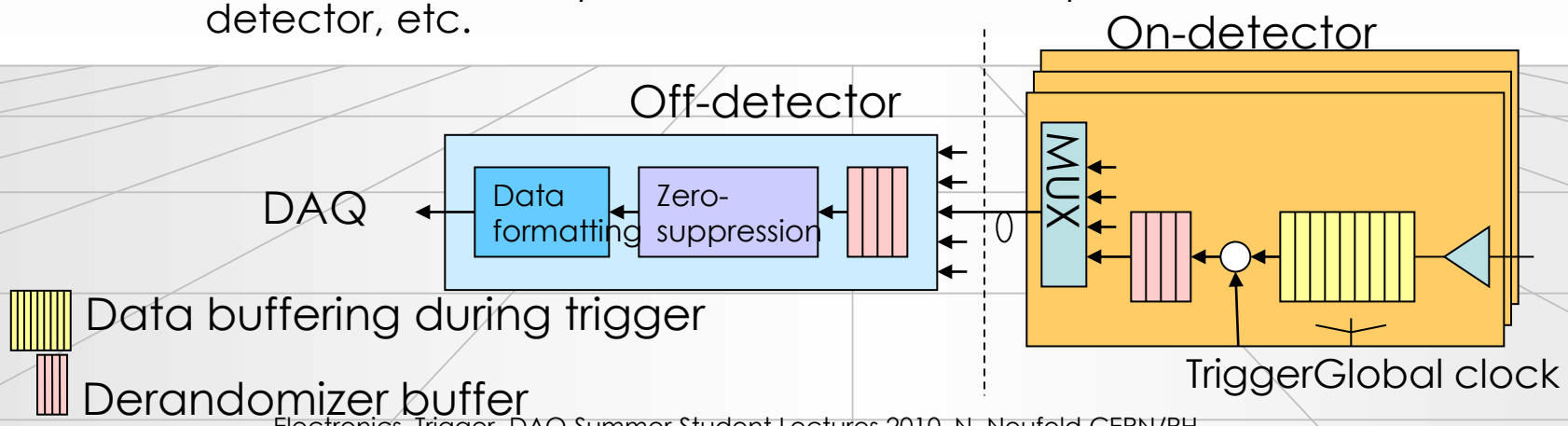
- Why spend bandwidth sending data that is zero for the majority of the time ?
- Perform zero-suppression and only send data with non-zero content
  - Identify the data with a channel number and/or a time-stamp
  - We do not want to loose information of interest so this must be done with great care taking into account pedestals, baseline variations, common mode, noise, etc.
  - Not worth it for occupancies above ~10%
- Alternative: data compression
  - Huffman encoding and alike
- TANSTAFL (There Aint No Such Thing As A Free Lunch)
  - Data rates fluctuates all the time and we have to fit this into links with a given bandwidth
  - Not any more event synchronous
  - Complicated buffer handling (overflows)
  - Before an experiment is built and running it is very difficult to give reliable estimates of data rates needed (background, new physics, etc.)





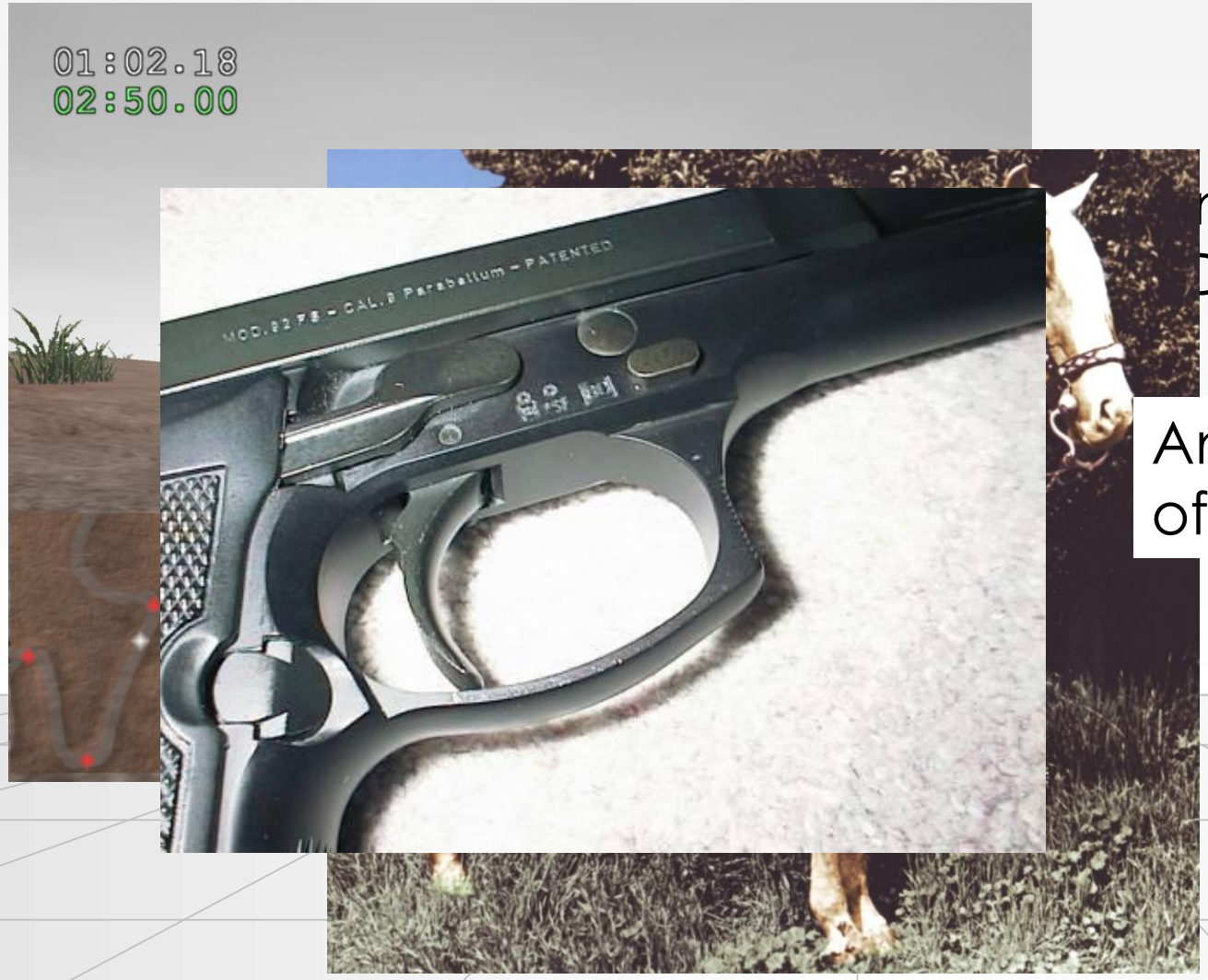
Synchronous readout

- All channels are doing the same "thing" at the same time
- Synchronous to a global clock (bunch crossing clock)
- Data-rate on each link is identical and depends only on *trigger -rate*
- On-detector buffers (*de-randomizers*) are of same size and there occupancy ("how full they are") depends only on the *trigger-rate*
- ☹ Lots of bandwidth wasted for zero's
  - Price of links determine if one can afford this
- ☺ No problems if occupancy of detectors or noise higher than expected
  - But there are other problems related to this: spill over, saturation of detector, etc.



Trigger & DAQ
(Sneak Preview)

What is a trigger?



n open-source
D rally game?

An important part of a Beretta

The most famous horse in movie history?

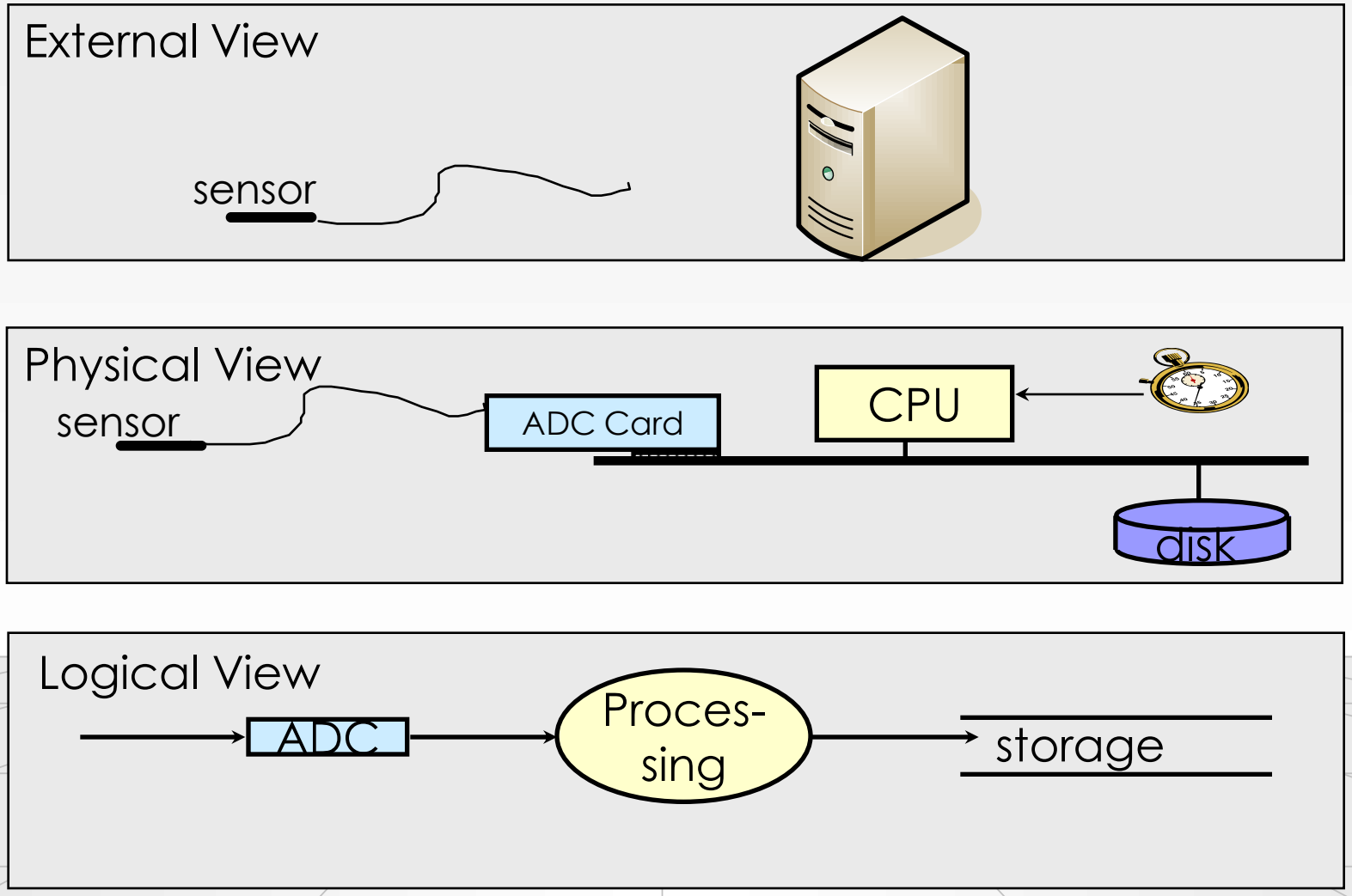
What is a trigger?

Wikipedia: "**A trigger is a system that uses simple criteria to rapidly decide which events in a particle detector to keep when only a small fraction of the total can be recorded.** "

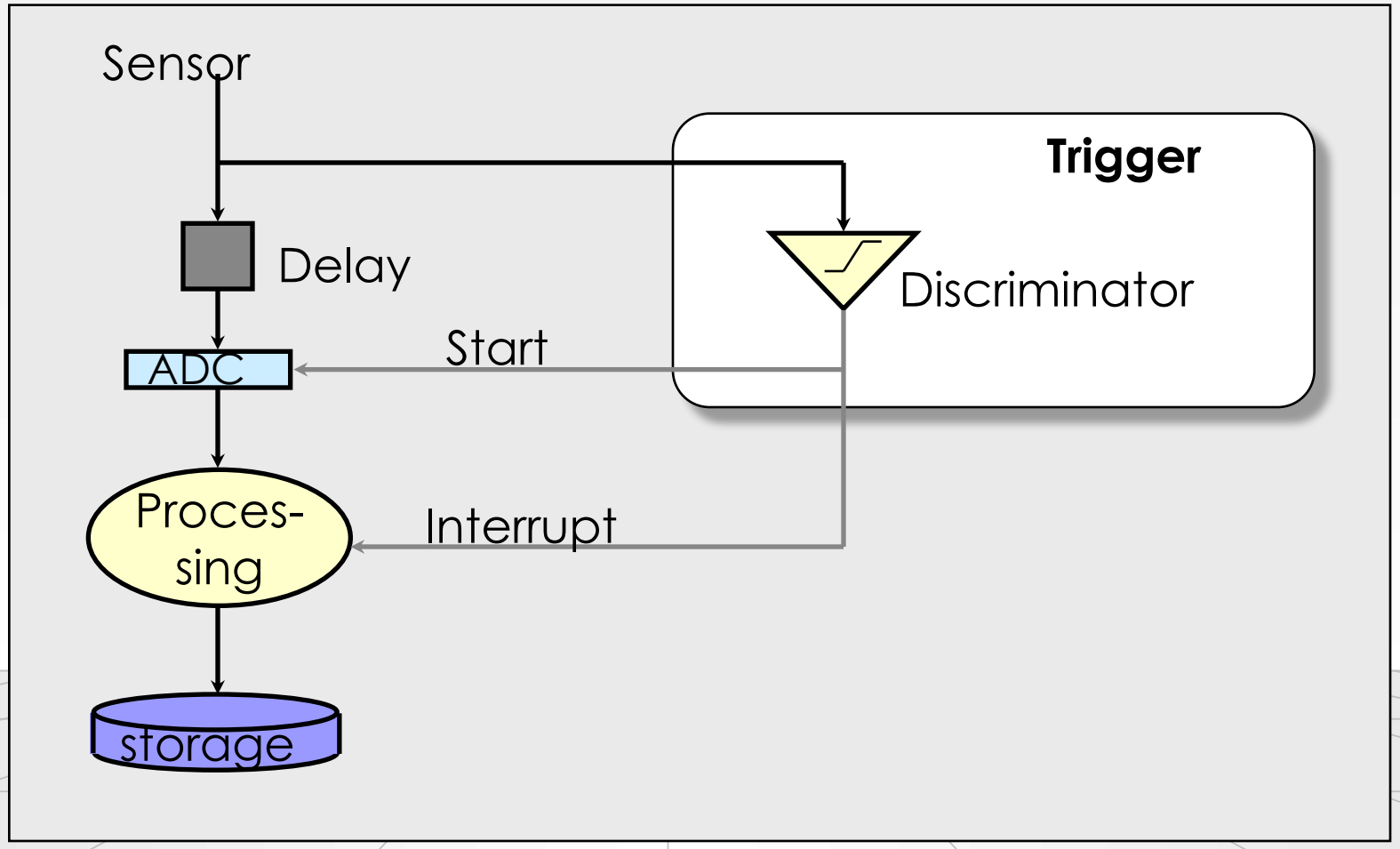
Trigger

- Simple
- Rapid
- Selective
- When only a small fraction can be recorded

Trivial DAQ

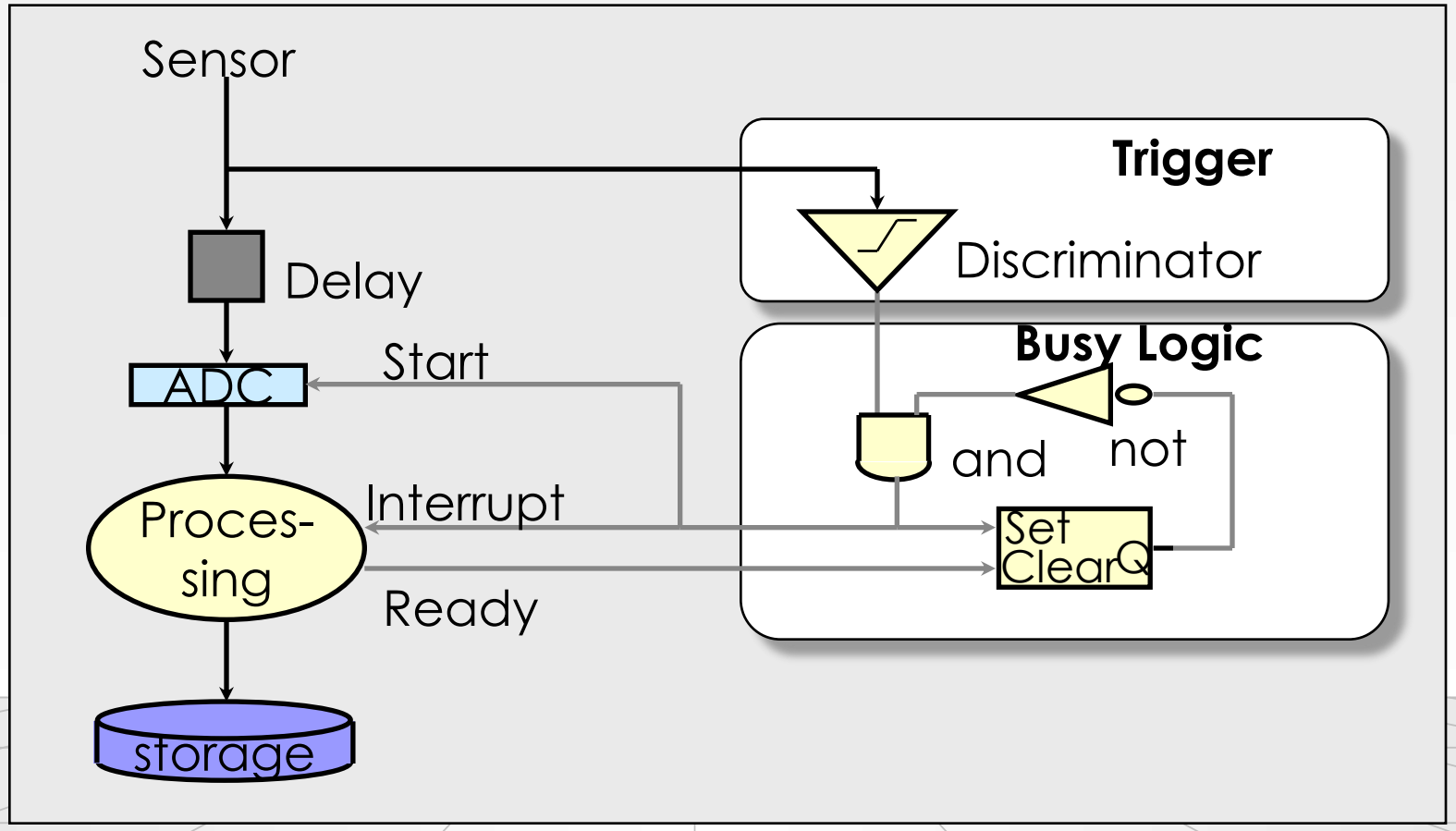

External View
sensor
Physical View
sensor
ADC Card
CPU
disk
Logical View
ADC
Proces-
sing
storage

Trivial DAQ with a real trigger



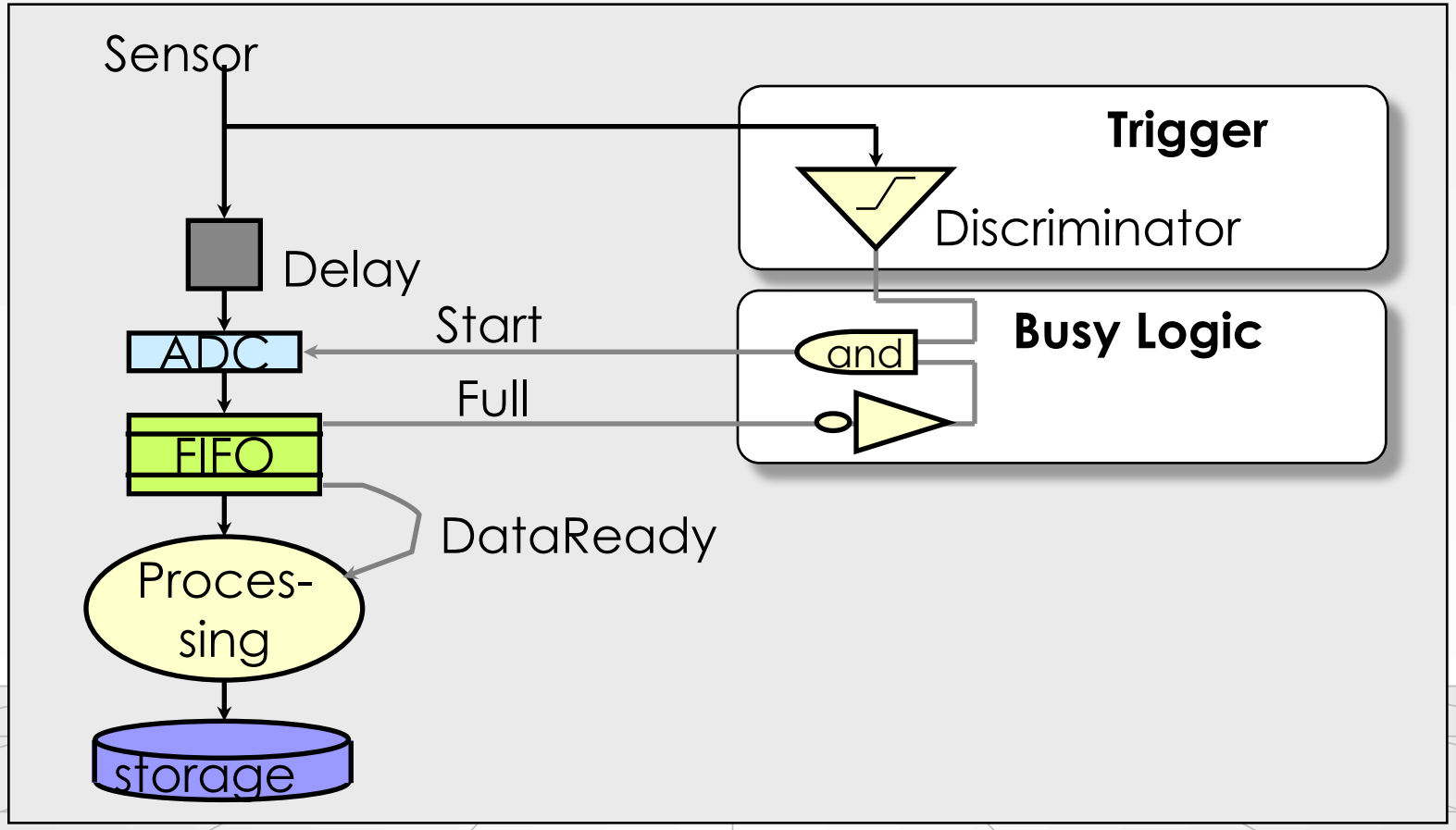
What if a trigger is produced when the *ADC* or *processing* is busy?

Trivial DAQ with a real trigger 2



Deadtime (%) is the ratio between the time the DAQ is *busy* and the total time.

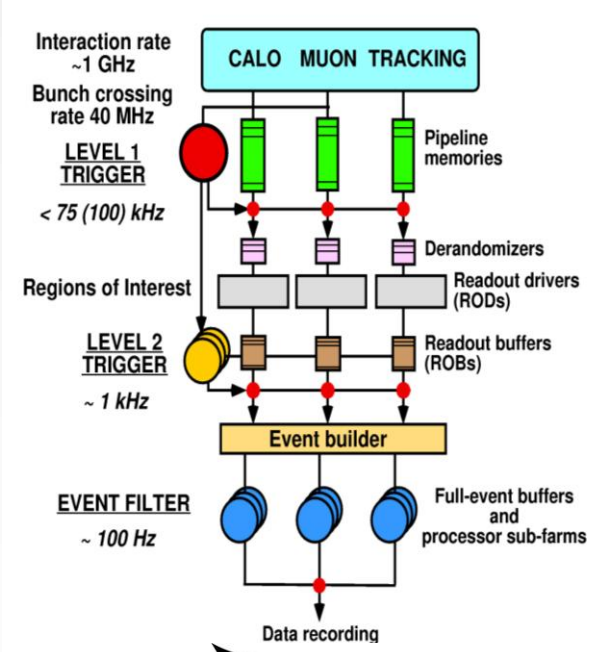
Trivial DAQ with a real trigger 3

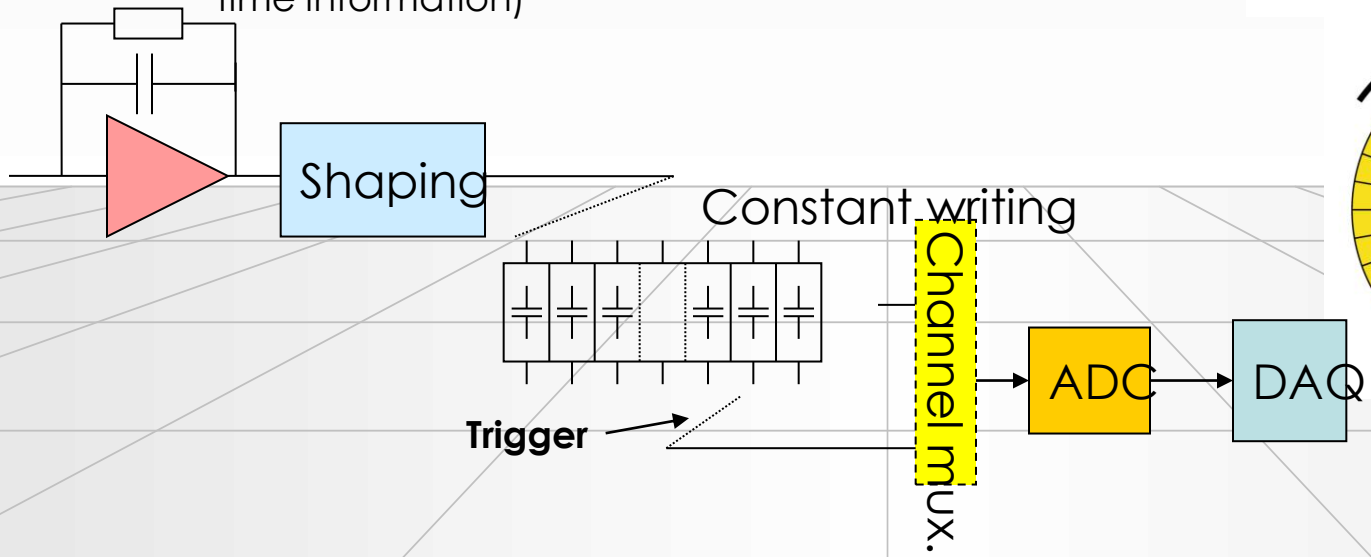


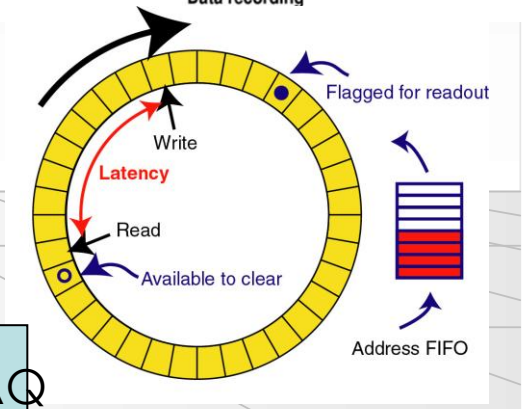
Buffers are introduced to de-randomize data, to decouple the data production from the data consumption. **Better performance**.

Triggered read-out

- Trigger processing requires some data transmission and processing time to make decision so front-ends must buffer data during this time. This is called the trigger latency
- For constant high rate experiments a "pipeline" buffer is needed in all front-end detector channels:
  analog or digital
  1. Real clocked pipeline (high power, large area, bad for analog)
  2. Circular buffer
  3. Time tagged (zero suppressed latency buffer based on time information)

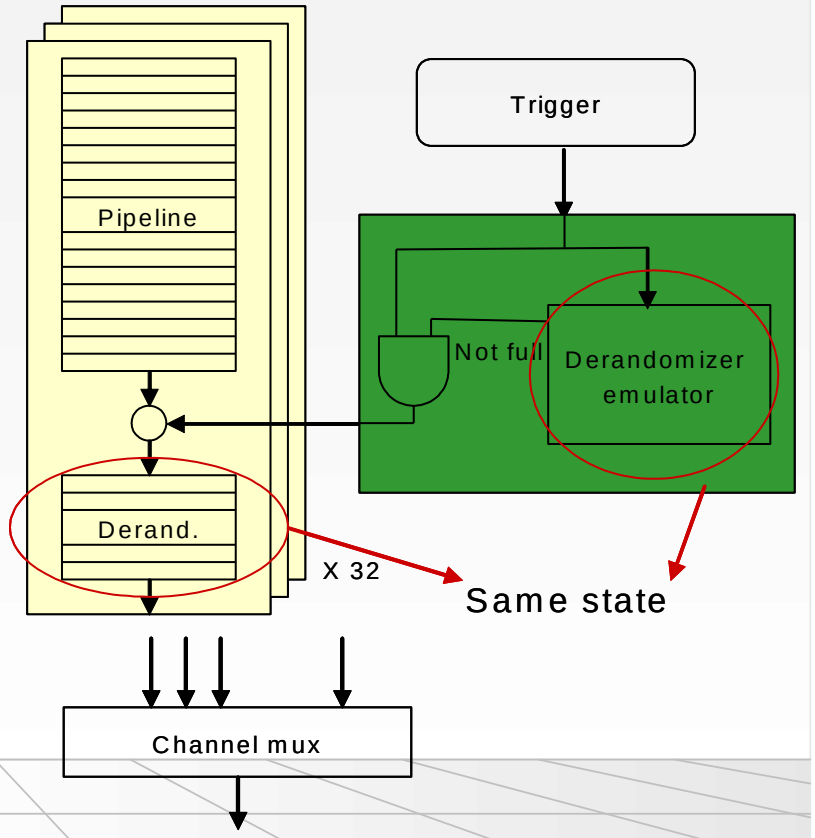




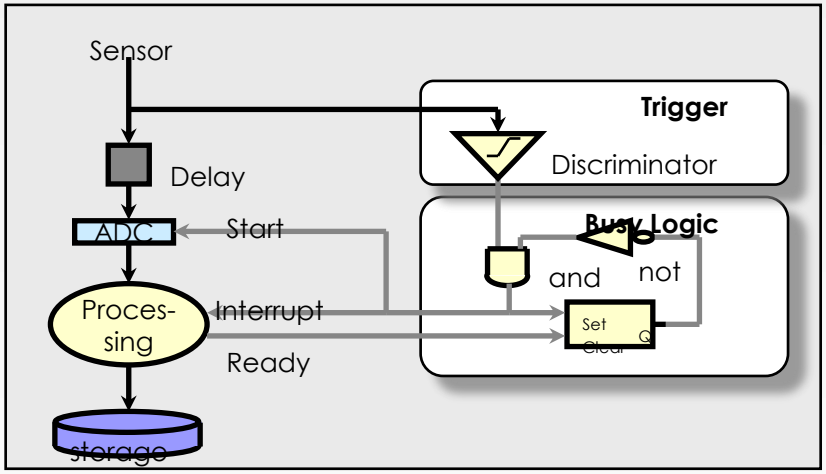


Trigger rate control

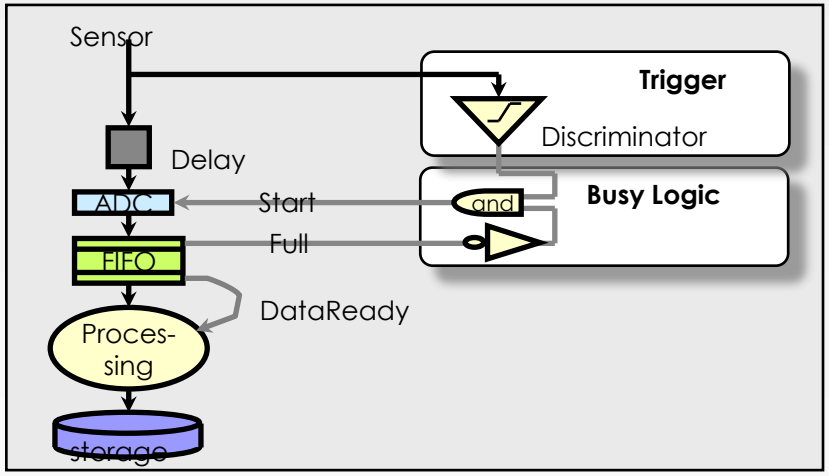
- Trigger rate determined by physics parameters used in trigger system: 1 kHz – 1MHz
  - The lower rate after the trigger allows sharing resources across channels (e.g. ADC and readout links)
- Triggers will be of random nature i.e. follow a Poisson distribution → a burst of triggers can occur within a short time window so some kind of rate control/spacing is needed
  - Minimum spacing between trigger accepts → dead-time
  - Maximum number of triggers within a given time window
- Derandomizer buffers needed in front-ends to handle this
  - Size and readout speed of this determines effective trigger rate



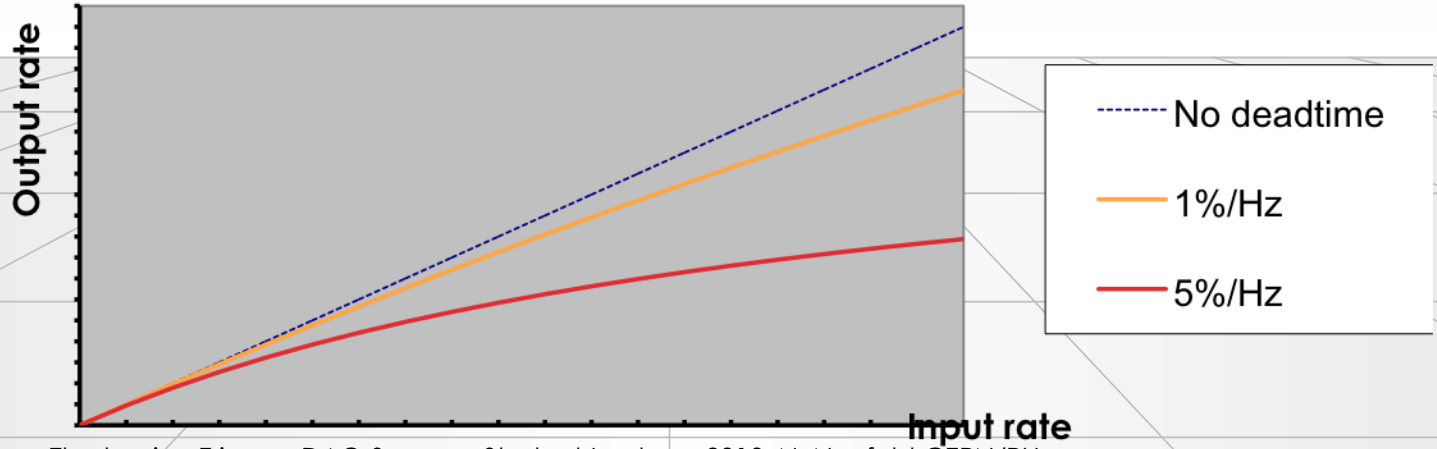
Effect of de-randomizing



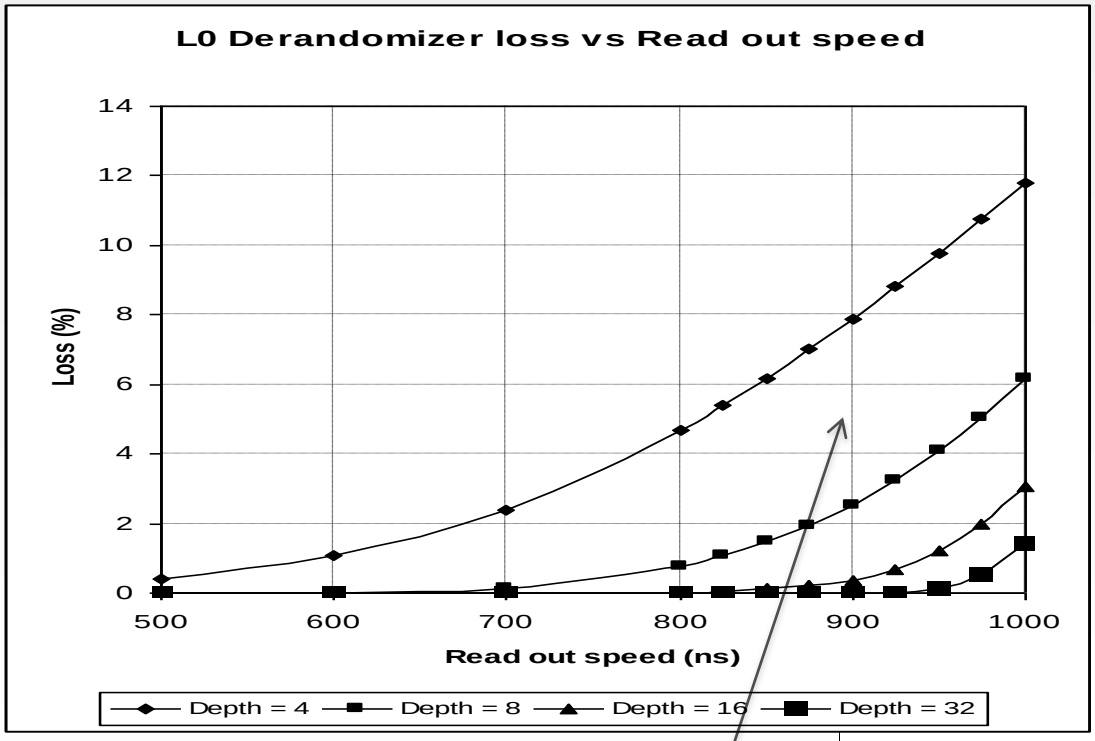
The system is *busy* during the ADC conversion time + processing time until the data is written to the storage



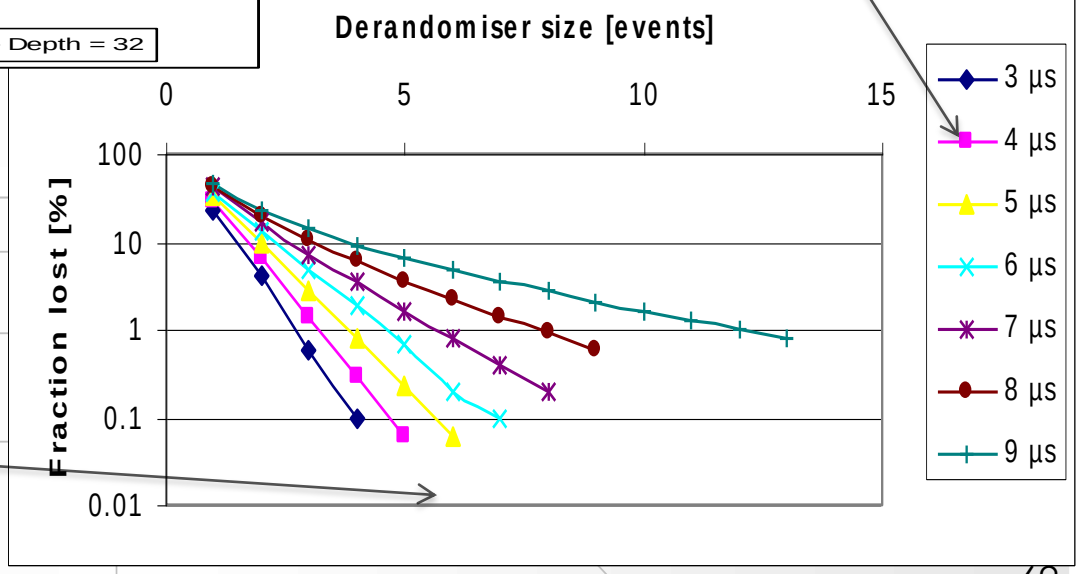
The system is *busy* during the ADC conversion time if the FIFO is not full (assuming the storage can always follow!)



System optimisation: LHCb front-end buffer



Trigger latency
Fixed to 4 μs in LHCb



Working point for LHCb

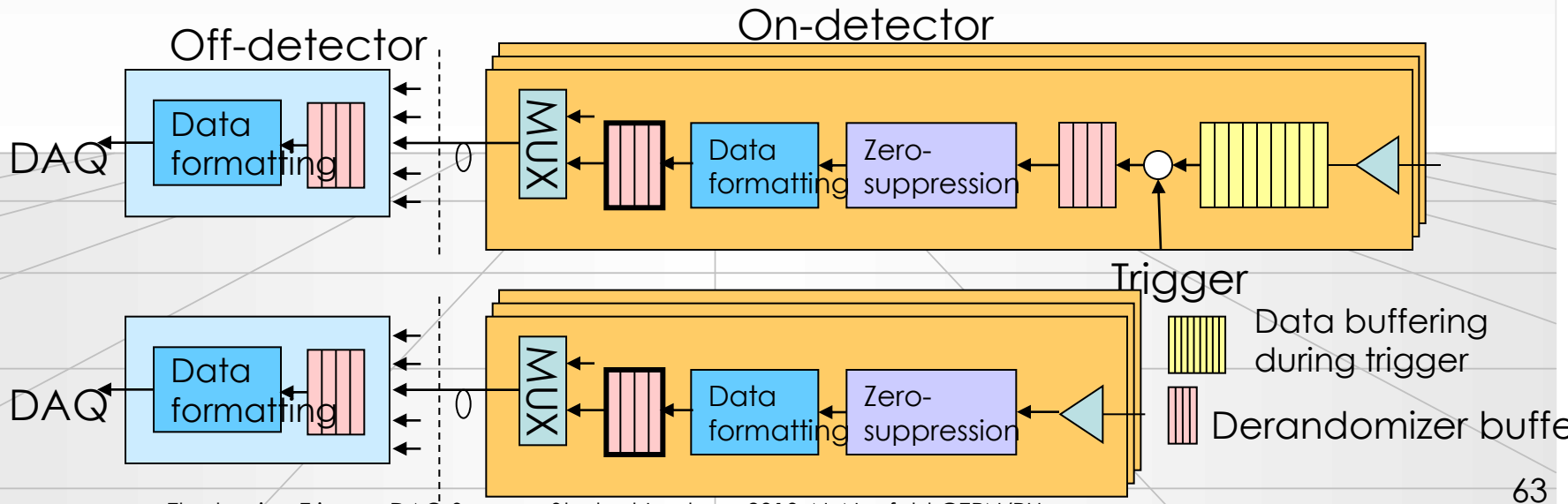
Max readout time: 900 ns

Derandomzier depth:
16 events

➔ 1 MHz maximum trigger accept rate

Asynchronous readout

- Remove zeros on the detector itself
  - Lower average bandwidth needed for readout links Especially interesting for low occupancy detectors
- Each channel "lives a life of its own" with unpredictable buffer occupancies and data are sent whenever ready (asynchronous)
- In case of buffer-overflow a truncation policy is needed → BIAS!!
  - Detectors themselves do not have 100% detection efficiency either.
  - Requires sufficiently large local buffers to assure that data is not lost too often (Channel occupancies can be quite non uniform across a detector with same front-end electronics)
- DAQ must be able to handle this (buffering!)
- Async. readout of detectors in LHC: ATLAS and CMS muon drift tube detectors, ATLAS and CMS pixel detectors, ATLAS SCT, several ALICE detectors as relatively low trigger rate (few kHz).



Summary: Readout to DAQ

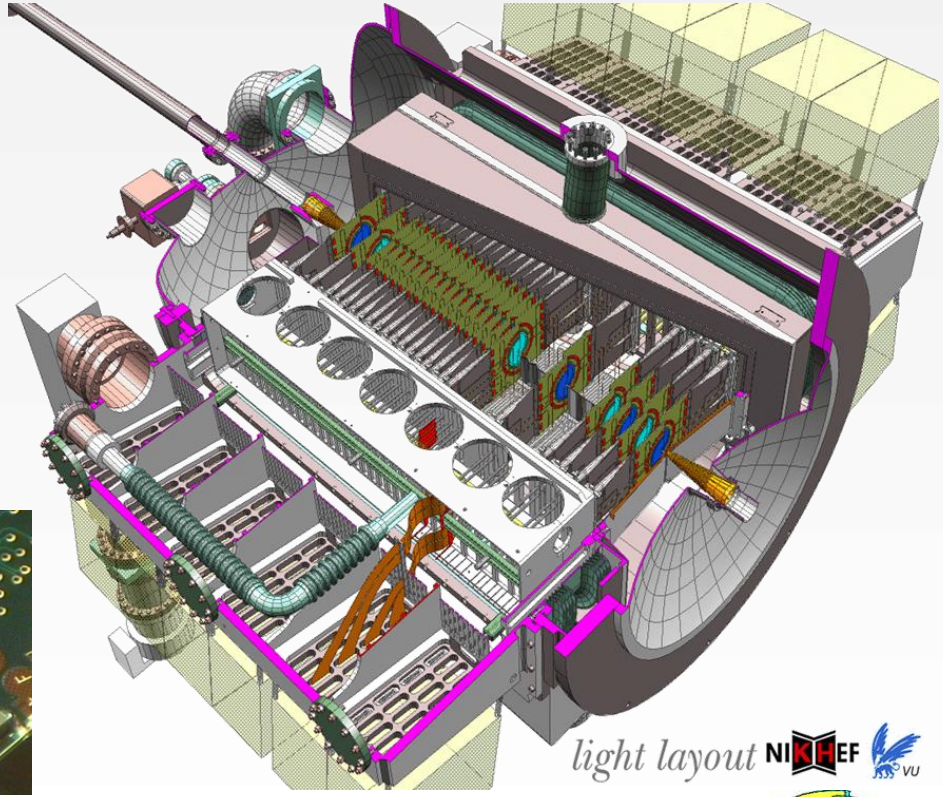
- Large amount of data to bring out of detector
  - Large quantity: ~100k in large experiment
  - High speed: Gbits/s
- Point to point unidirectional
- Transmitter side has specific constraints
  - Radiation
  - Magnetic fields
  - Power/cooling
  - Minimum size and mass
  - Must collect data from one or several front-end chips
- Receiver side can be commercially available module/components (use of standard link protocols when ever possible)

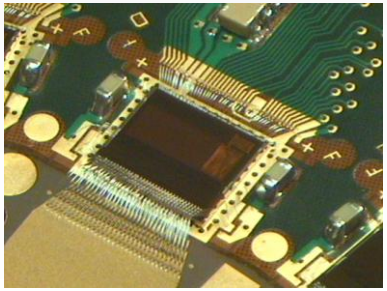
Lecture 3/5

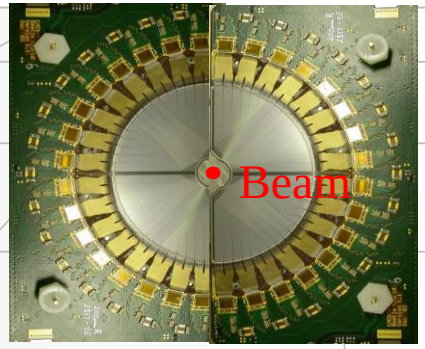
Electronics for LHC experiments

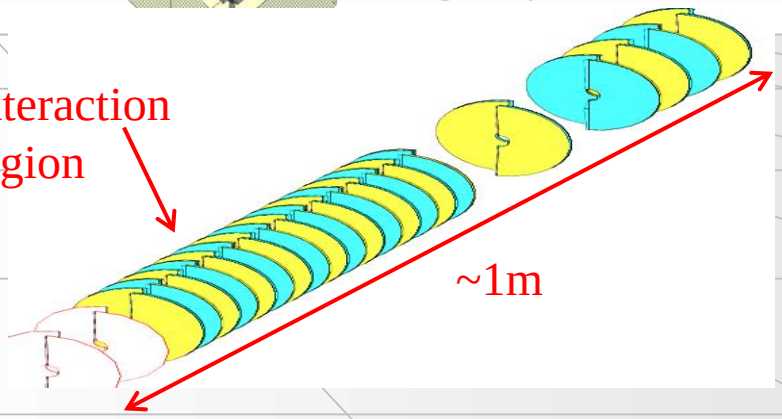
An example: the LHCb Vertex detector and its readout IC beetle

- 172k Channels
- Strips in R and φ projection (~10um vertex resolution)
- Located 1cm from beam
- Analog readout (via twisted pair cables over 60m)



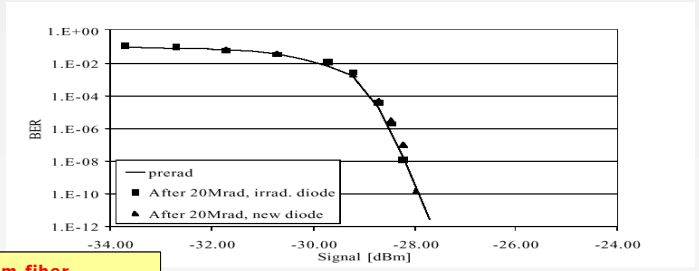


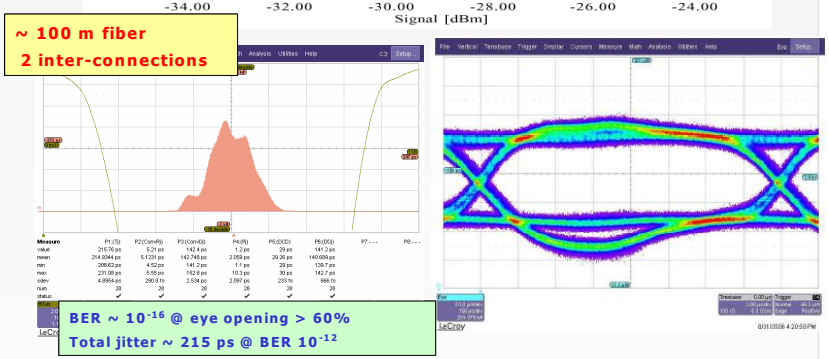


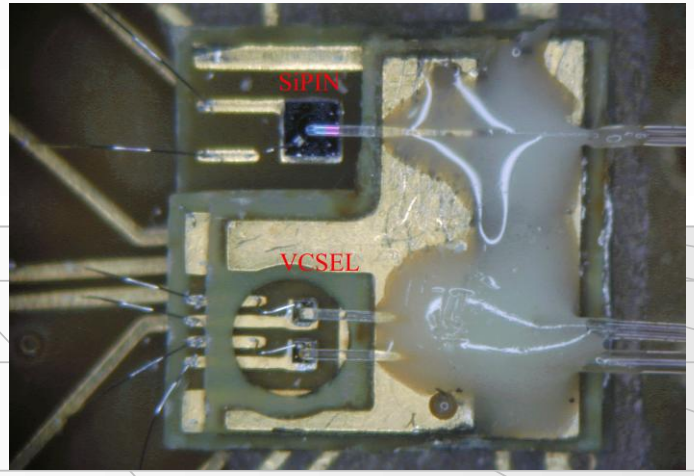


Digital optical links

- High speed: 1Ghz - 10GHz – 40GHz
- Extensively used in telecommunications (expensive) and in computing ("cheap")
- Encoding
  - Inclusion of clock for receiver PLL's
  - DC balanced
  - Special synchronization characters
  - Error detection and or correction
- Reliability and error rates strongly depending on received optical power and timing jitter
- Multiple (16) serializers and deserializers directly available in modern high end FPGA's.







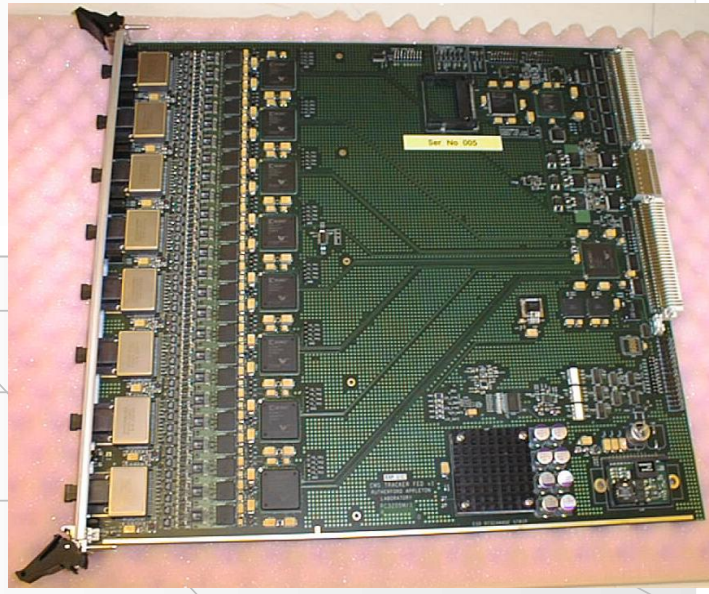
DAQ interfaces / Readout Boards

- Front-end data reception
  - Receive optical links from multiple front-ends: 24 - 96
  - Located outside radiation
- Event checking
  - Verify that data received is correct
  - Verify correct synchronization of front-ends
- Extended digital signal processing to extract information of interest and minimize data volume
- Event merging/building
  - Build consistent data structures from the individual data sources so it can be efficiently sent to DAQ CPU farm and processed efficiently without wasting time reformatting data on CPU.
  - Requires significant data buffering
- High level of programmability needed
- Send data to CPU farm at a rate that can be correctly handled by farm
  - 1 Gbits/s Ethernet (next is 10Gbits/s)
  - In house link with PCI interface: S-link

Requires a lot of fast digital processing and data buffering: **FPGA's**, DSP's, embedded CPU

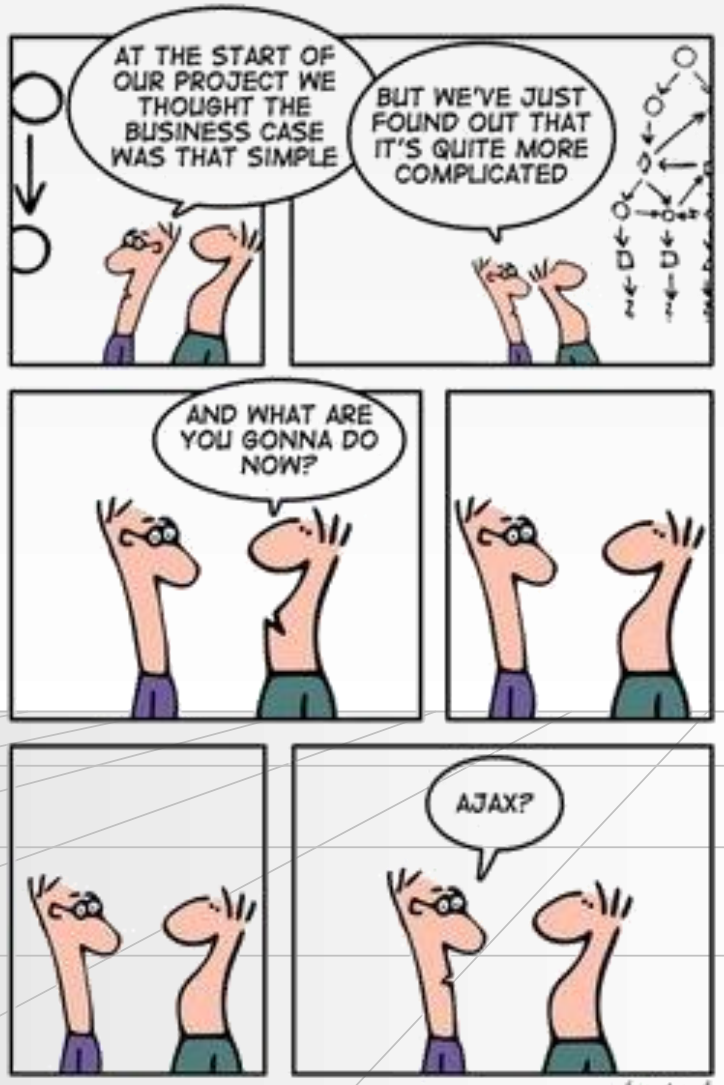
Use of ASIC's not justified

Complicated modules that are only half made when the hardware is there: FPGA firmware (from HDL), DSP code, on-board CPU software, etc.



(Large) Systems

New problems



- Going from single sensors to building detector read-out of the circuits we have seen, brings up a host of new problems:
  - Power, Cooling
  - Crosstalk
  - Radiation (LHC)
- Some can be tackled by (yet) more sophisticated technologies

Radiation effects

- In modern experiments large amounts of electronics are located inside the detector where there may be a high level of radiation
  - This is the case for 3 of the 4 LHC experiments (10 years running)
    - Pixel detectors: 10 -100 Mrad
    - Trackers: ~10Mrad
    - Calorimeters: 0.1 – 1Mrad
    - Muon detectors: ~10krad
    - Cavern: 1 – 10krad

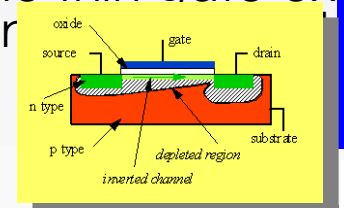
1 Rad == 10 mGy

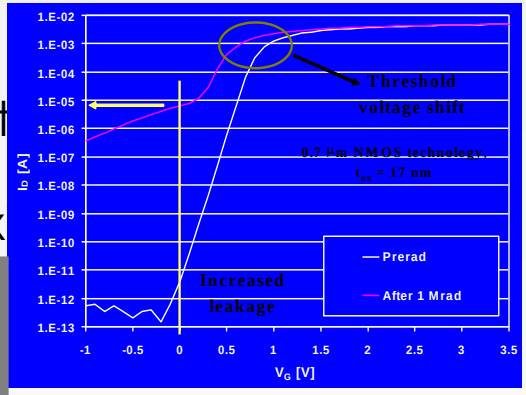
1 Gy = 100 Rad

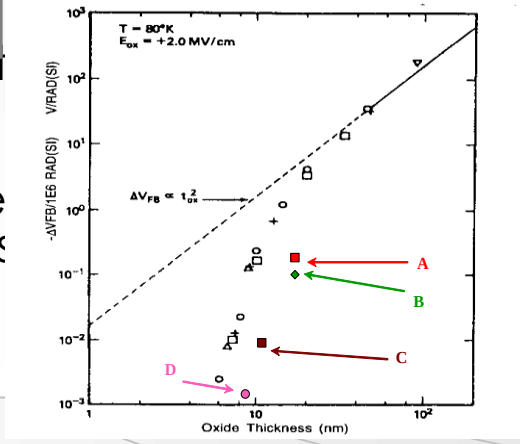
- Normal commercial electronics will not survive within this environment
  - One of the reasons why all the on-detector electronics in the LHC experiment are custom made
- Special technologies and dedicated design approaches are needed to make electronics last in this unfriendly environment
- Radiation effects on electronics can be divided into three major effects
  - Total dose
  - Displacement damage
  - Single event upsets

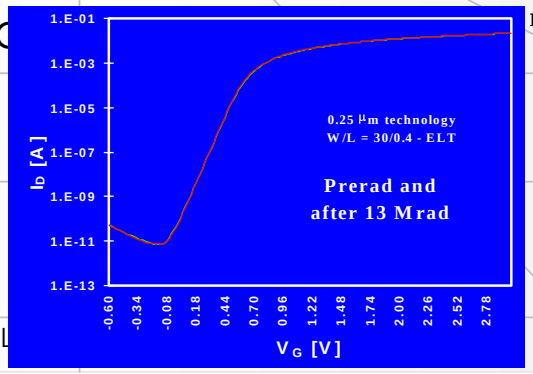
Total dose

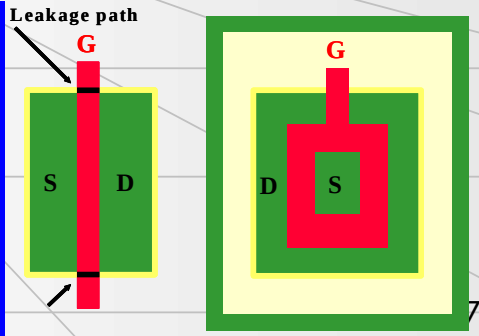
- Generated charges from traversing particles gets t
  insulators of the active devices and changes their
- For CMOS devices this happens in the thin gate ox
  have a major impact on the functior
  - Threshold shifts
  - Leakage current
- In deep submicron technologies
  (<0.25um) the trapped charges are removed by t
  through the very thin gate oxide
    - Only limited threshold shifts
- The leakage currents caused by end effects of the
  (NMOS) can be cured by using enclosed transistors
  - For CMOS technologies below the 130nm generation the use of enclosed NMOS devices does not seem necessary.
    But other effects may show up
- No major effect on high speed
  bipolar technologies





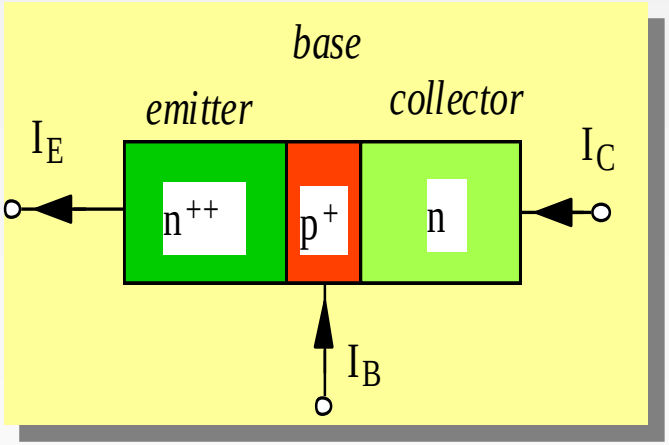


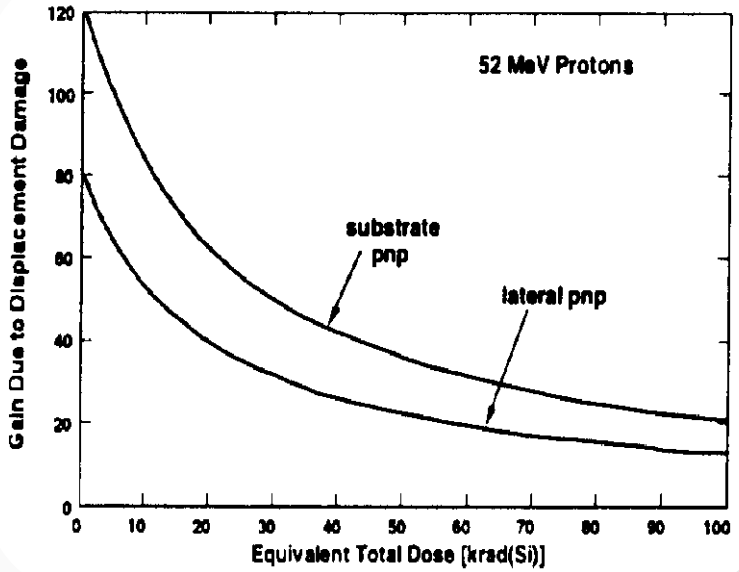




Displacement damage

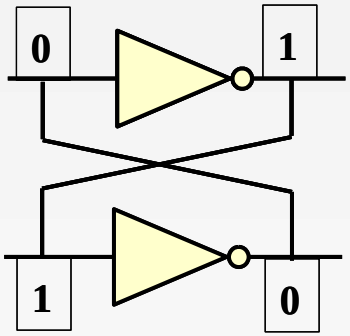
- Traversing hadrons provokes displacements of atoms in the silicon lattice.
- Bipolar devices relies extensively on effects in the silicon lattice.
  - Traps (band gap energy levels)
  - Increased carrier recombination in base
- Results in decreased gain of bipolar devices with a dependency on the dose rate.
- No significant effect on MOS devices
- Also seriously affects Lasers and PIN diodes used for optical links.

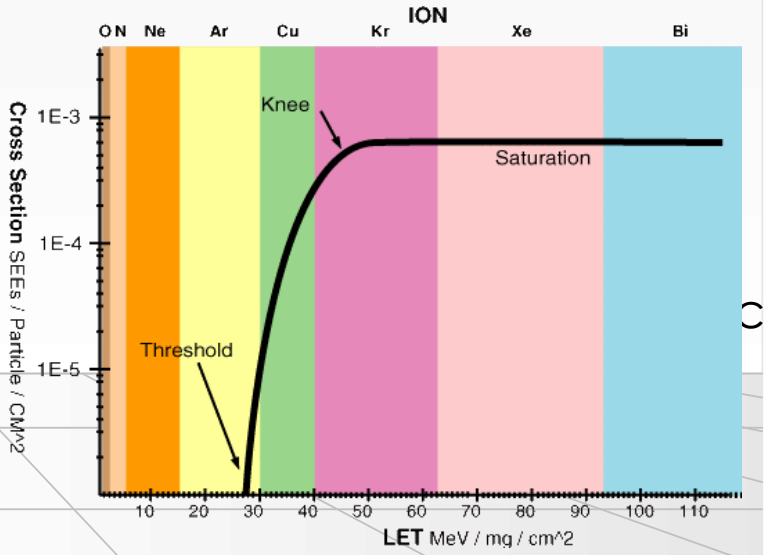




Single event upsets

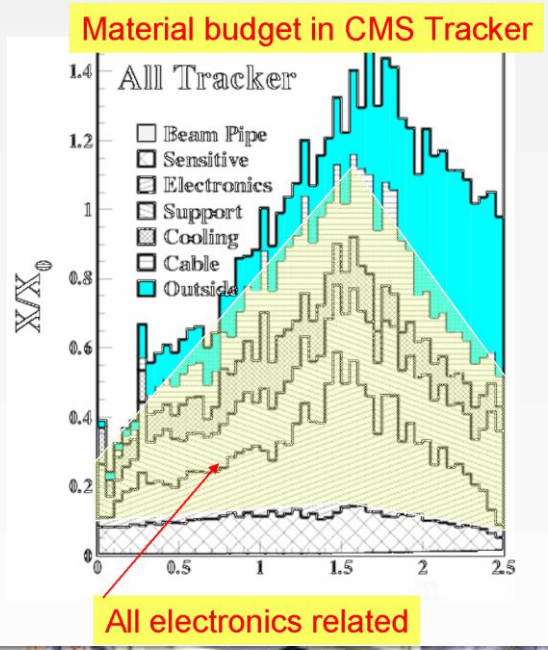
- Deposition of sufficient charge can make a memory cell or a flip-flop change value
- As for SEL, sufficient charge can only be deposited via a nuclear interaction for traversing hadrons
- The sensitivity to this is expressed as an efficient cross section for this to occur
- This problem can be resolved at the circuit level or at the logic level
- Make memory element so large and slow that deposited charge not enough to flip bit
- Triple redundant (for registers)
- Hamming coding (for memories)
  - Single error correction, Double error detection
  - Example Hamming codes: 5 bit additional for 8 bit data
    - ham[0] = d[1] $ d[2] $ d[3] $ d[4];
      ham[1] = d[1] $ d[5] $ d[6] $ d[7];
      ham[2] = d[2] $ d[3] $ d[5] $ d[6] $ d[8];
      ham[3] = d[2] $ d[4] $ d[5] $ d[7] $ d[8];
      ham[4] = d[1] $ d[3] $ d[4] $ d[6] $ d[7] $ d[8];
      $ = XOR
  - Overhead decreasing for larger words
    32bits only needs 7hamming bits

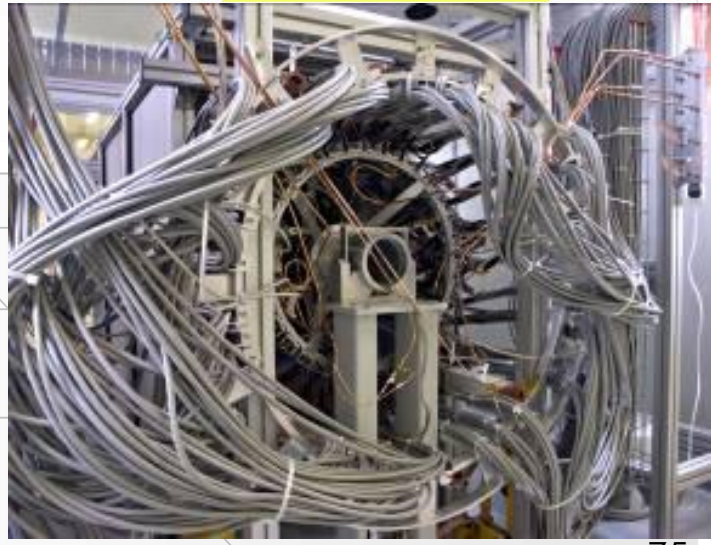




Powering

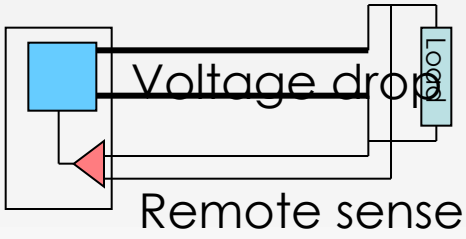
- Delivering power to the front-end electronics highly embedded in the detectors has been seen to be a major challenge (underestimated).
- The related cooling and power cabling infrastructure is a serious problem of the inner trackers as any additional material seriously degrades the physics performance of the whole experiment.
- A large majority of the material in these detectors in LHC relates to the electronics, cooling and power and not to the silicon detector them selves (which was the initial belief)
- How to improve
  1. Lower power consumption
  2. Improve power distribution

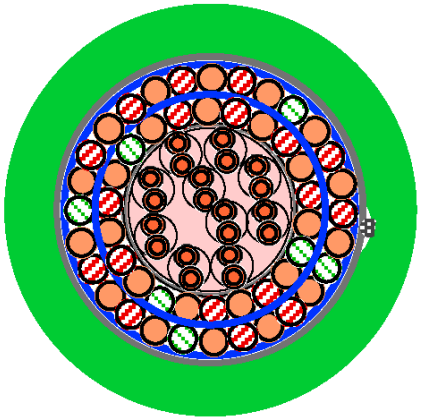


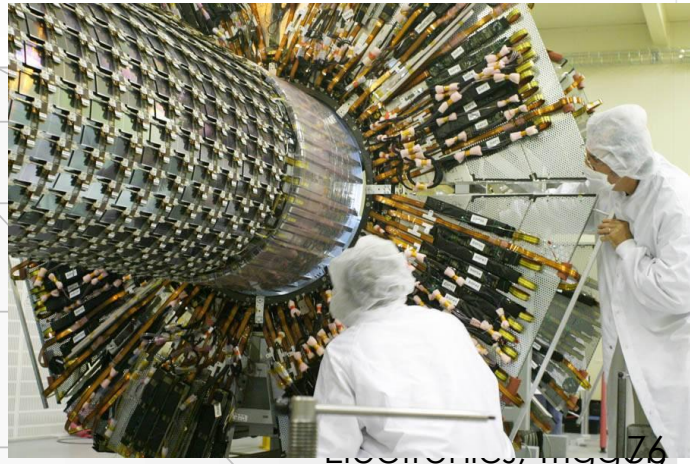


The problem as is

- Total power: ~500kw (to be supplied and cooled)
  - Trackers: ~ 60 kW
  - Calorimeters: ~ 300 kW
  - Muon: ~ 200 kW
  - Must for large scale detectors be delivered over 50m – 100m distance
- Direct supply of LV power from ~50m away
  - Big fat copper cables needed
    - Use aluminum cables for last 5-10m to reduce material budget
  - Power supply quality at end will not be good with varying power consumption (just simple resistive losses)
    - If power consumption constant then this could be OK
  - Use remote sense to compensate
    - This will have limited reaction speed
    - May even become unstable for certain load configurations
  - Power loss in cables will be significant for the voltages (2.5v) and currents needed: ~50% loss in cables (that needs to be cooled)
- Use of local linear regulators
  - Improves power quality at end load.
  - Adds additional power loss: 1 – 2 v head room needed for regulator
  - Increases power losses and total efficiency now only: ~25% (more cooling needed)







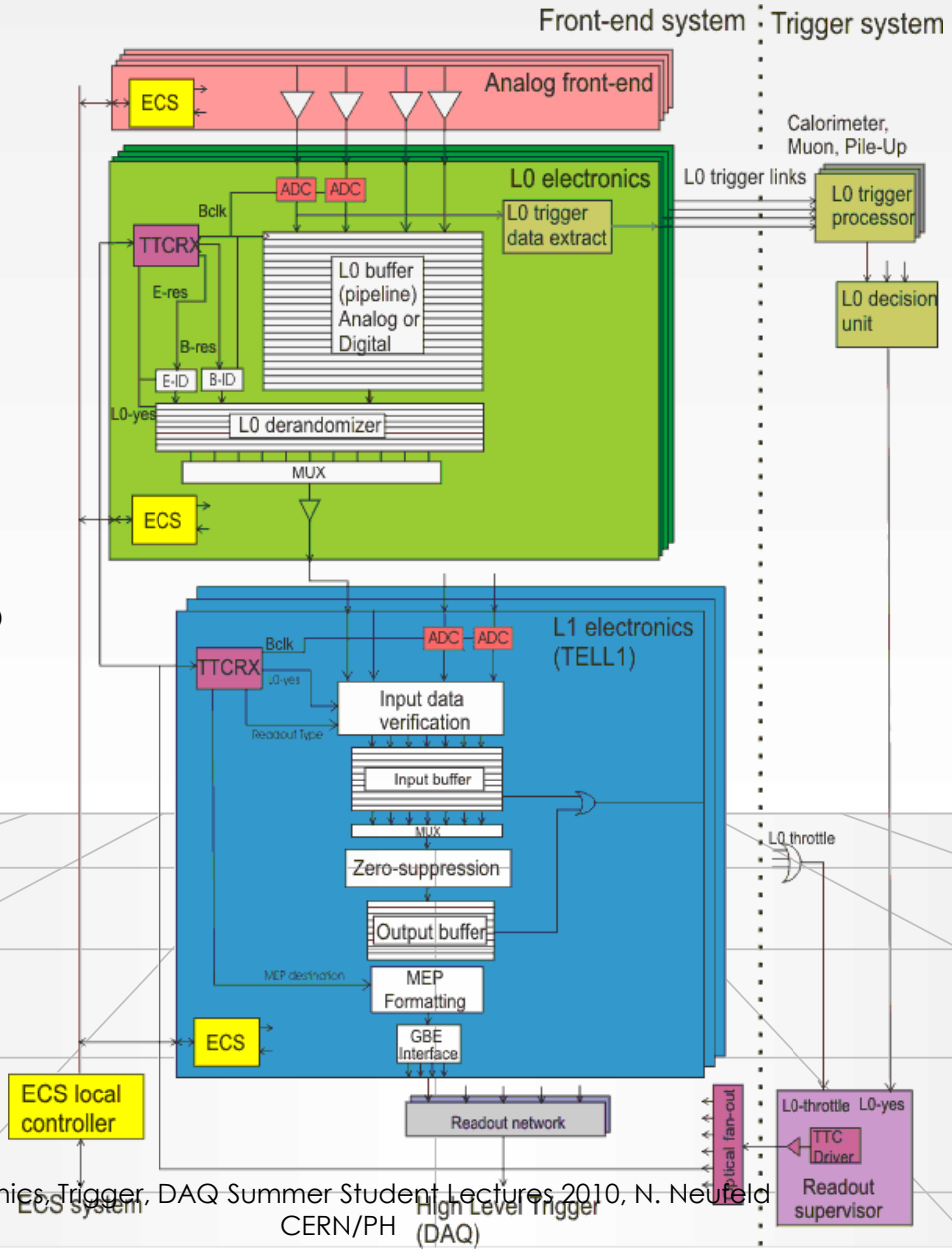
Use of DC-DC converters

- For high power consumers (e.g. calorimeter) the use of local DC-DC converters are inevitable.
- These must work in radiation and high magnetic fields
  - This is not exactly what switched mode DC-DC converters like
  - Magnetic coils and transformers saturated
  - Power devices do not at all like radiation: SEU - > single event burnout -> smoke -> disaster
- DC-DC converters for moderate radiation and moderate magnetic fields have been developed and used
  - Some worries about the actual reliability of these for long term

FEE & DAQ by electronics engineers

FEE = Front End Electronics

Example from LHCb



Lecture 4/5

Data Aquisition

Introduction: DAQ

- Data Acquisition is a specialized engineering discipline thriving mostly in the eco-system of large science experiments, particularly in HEP
- It consists mainly of electronics, computer science, networking and (we hope) a little bit of physics

Outline

- Introduction
  - Data acquisition
  - The first data acquisition campaign
- A simple DAQ system
  - One sensor
  - More and more sensors
- Read-out with buses
  - Crates & Mechanics
  - The VME Bus
- A DAQ for a large experiment
  - Sizing it up
  - Trigger
  - Front-end Electronics
  - Readout with networks
    - Event building in switched networks
    - Problems in switched networks
- A lightning tour of ALICE, ATLAS, CMS and LHCb DAQs

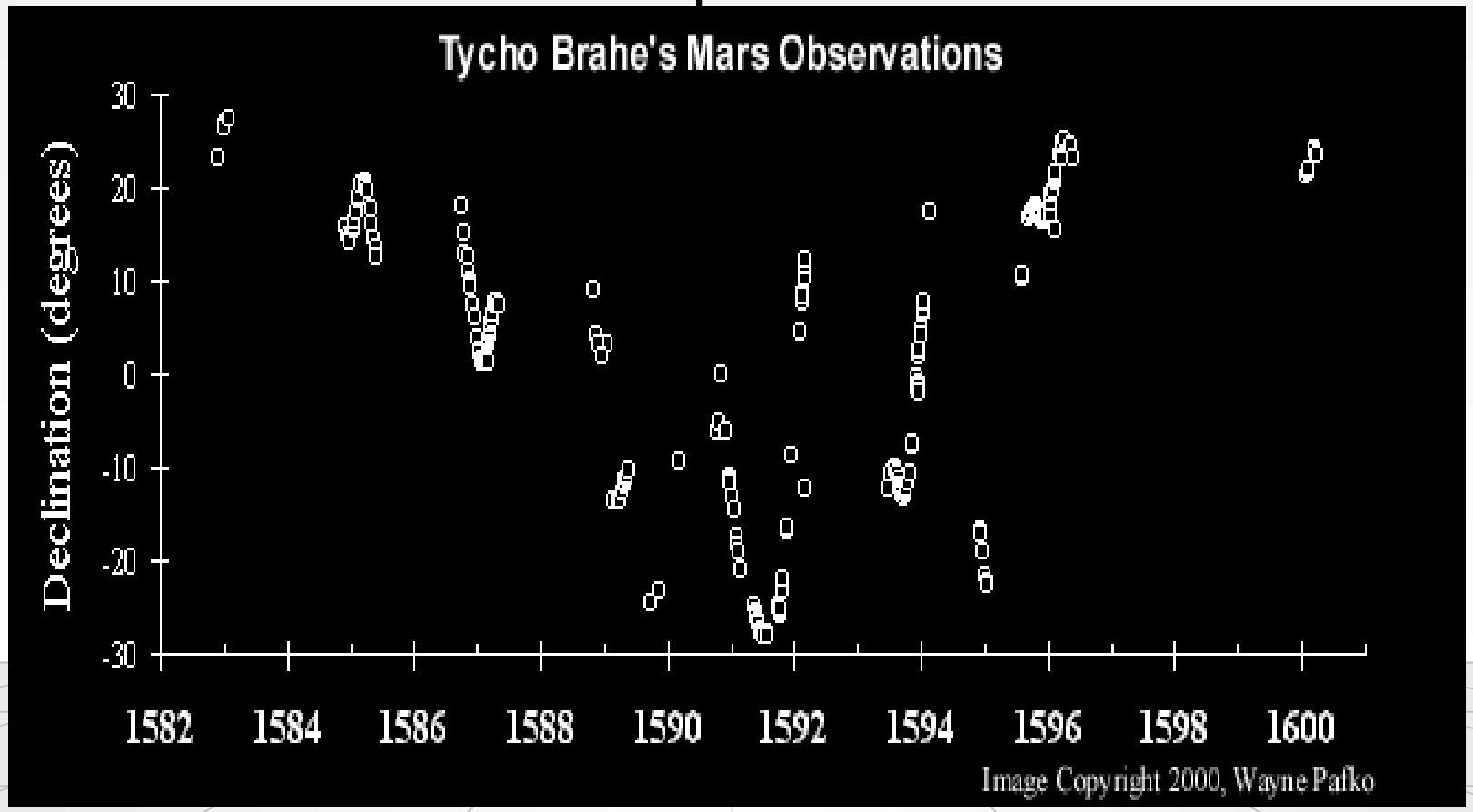
Tycho Brahe and the Orbit of Mars

I've studied all available charts of the planets and stars and none of them match the others. There are just as many measurements and methods as there are astronomers and all of them disagree. What's needed is a long term project with the aim of mapping the heavens conducted from a single location over a period of several years.

Tycho Brahe, 1563 (age 17).

- First measurement campaign
- Systematic data acquisition
  - Controlled conditions (same time of the day and month)
  - Careful observation of boundary conditions (weather, light conditions etc…) - important for data quality / systematic uncertainties

The First Systematic Data Acquisition



- Data acquired over 18 years, normally e every month
- Each measurement lasted at least 1 hr with the naked eye
- Red line (only in the animated version) shows comparison with modern theory

Tycho's DAQ in Today's Terminology

- Bandwith (bw) = Amount of data transferred / per unit of time
  - "Transferred" = written to his logbook
  - "unit of time" = duration of measurement
  - bw_{Tycho} = ~ 100 Bytes / h (compare with LHCb 40.000.000.000 Bytes / s)
- Trigger = in general something which tells you when is the "right" moment to take your data
  - In Tycho's case the position of the sun, respectively the moon was the trigger
  - the trigger rate ~ 3.85 x 10^{-6} Hz (compare with LHCb 1.0 x 10^{6} Hz)

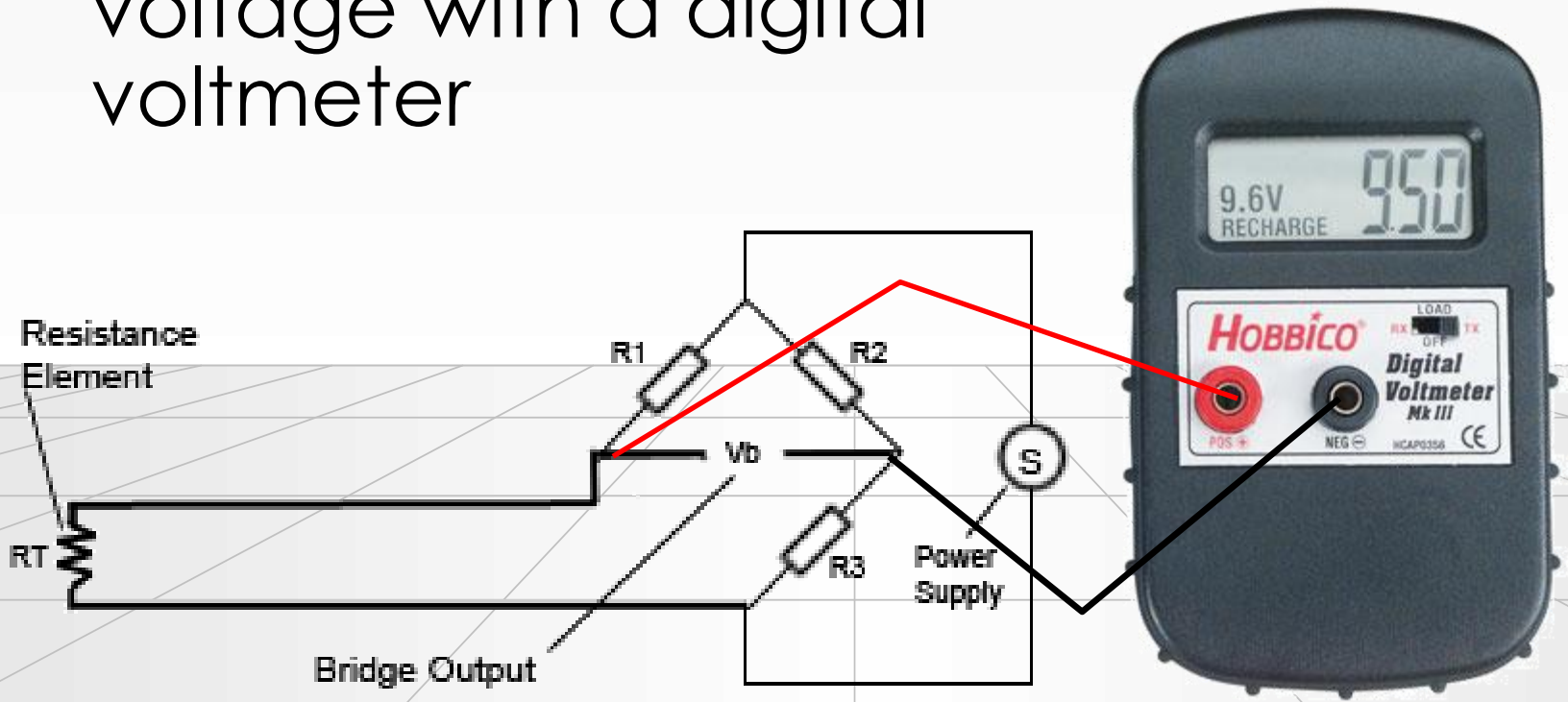
Some More Thoughts on Tycho

- Tycho did not do the correct analysis of the Mars data, this was done by Johannes Kepler (1571-1630), eventually paving the way for Newton's laws
- Morale: the size & speed of a DAQ system are not correlated with the importance of the discovery!

A Very Simple Data Acquisition System

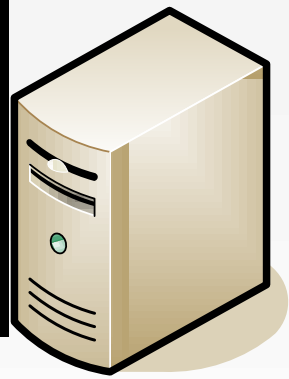
Measuring Temperature

- Suppose you are given a Pt100 thermo-resistor
- We read the temperature as a voltage with a digital voltmeter



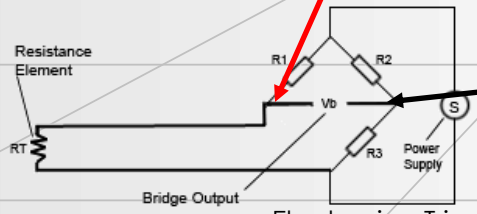
Reading Out Automatically

```
#include <libusb.h>
struct usb_bus *bus;
struct usb_device *dev;
usb_dev_handle *vmh = 0;
usb_find_busses(); usb_find_devices();
for (bus = usb_busses; bus; bus = bus->next)
          for (dev = bus->devices; dev; dev = dev-
>next)
                    if (dev->descriptor.idVendor ==
HOBBICO) vmh = usb_open(dev);
usb_bulk_read(vmh ,3,&u,sizeof(float),500);
```

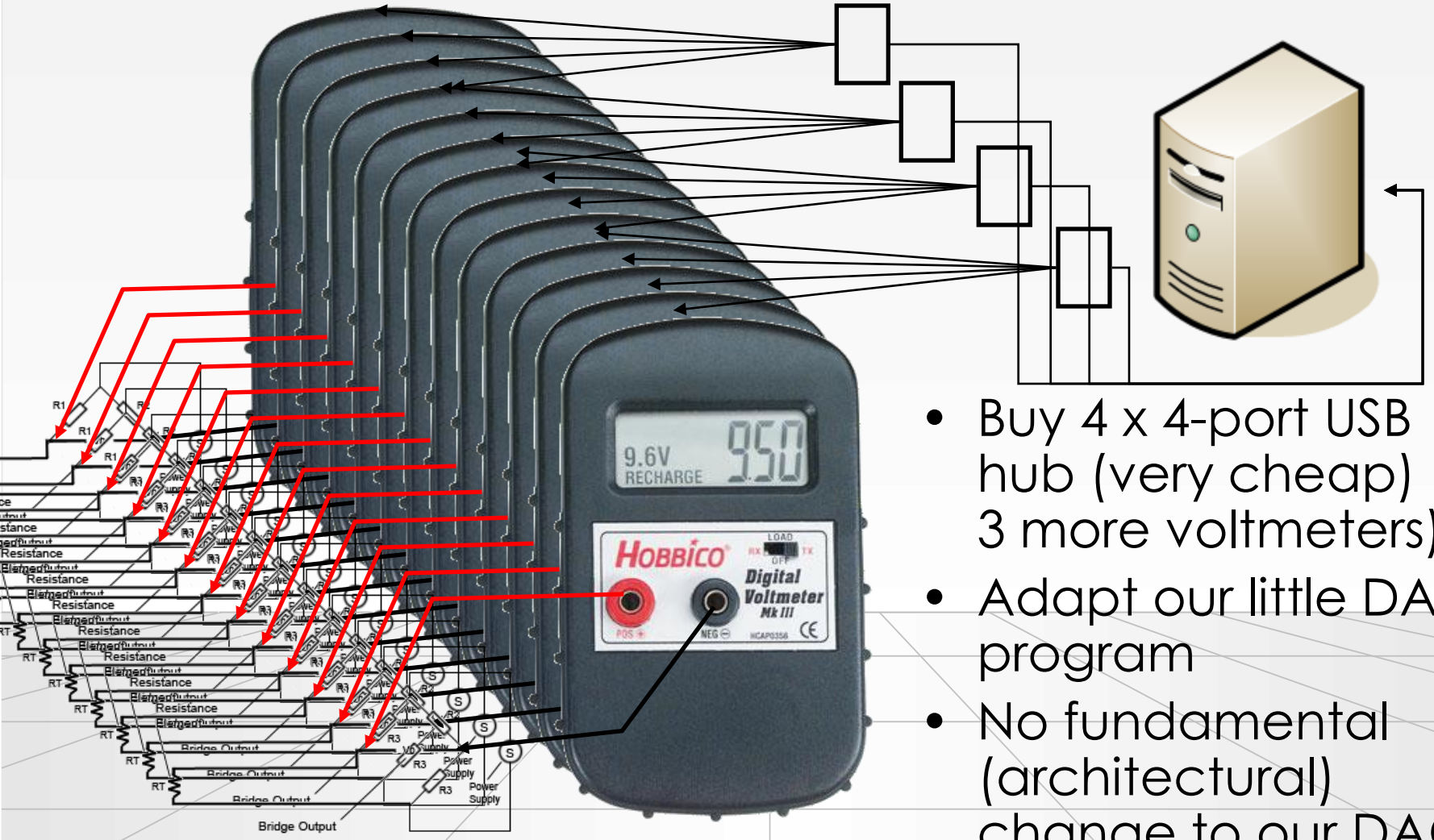


Note how small the sensor has become.
In DAQ we normally need not worry about the details of the things we readout

USB/RS232



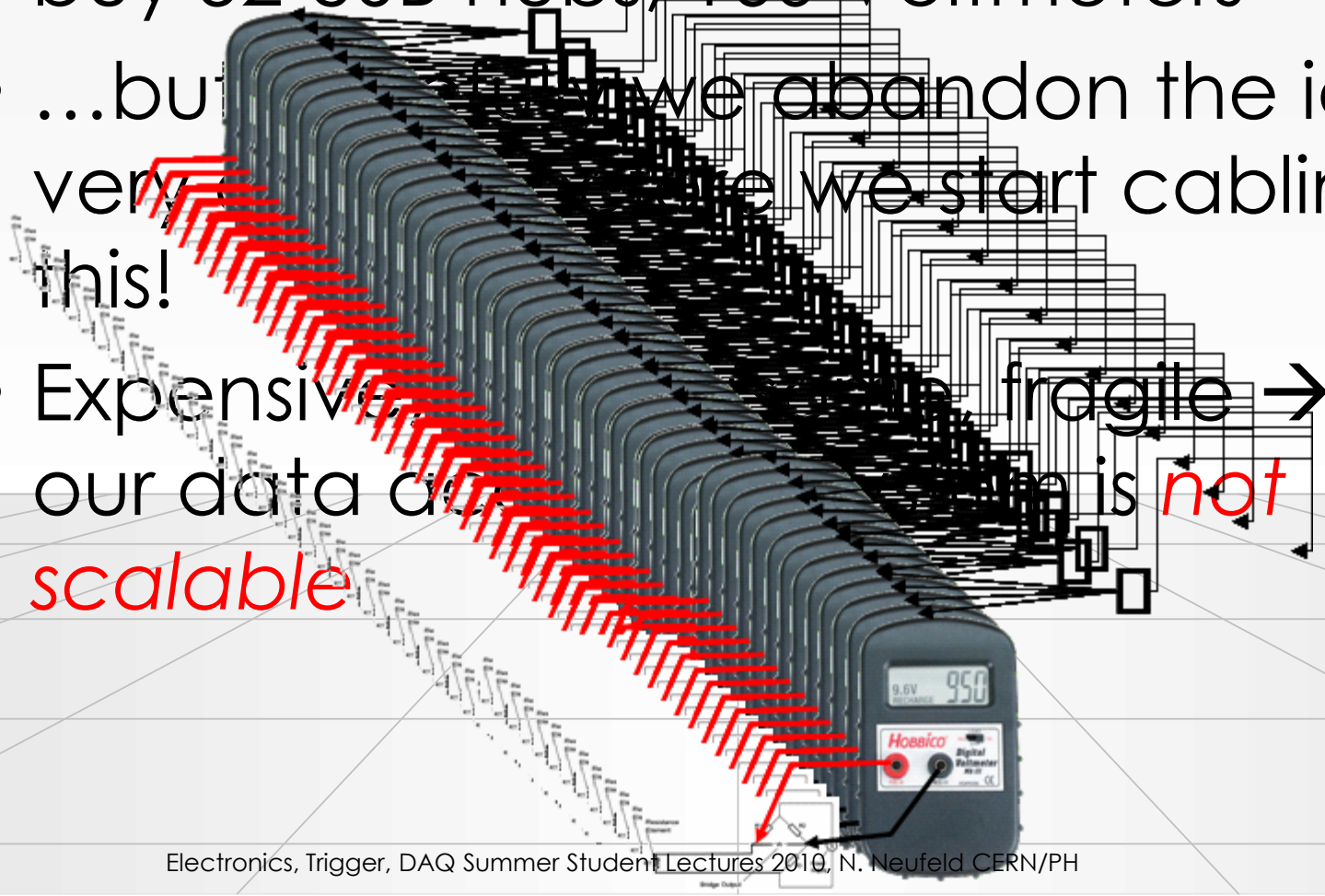
Read-out 16 Sensors



- Buy 4 x 4-port USB hub (very cheap) (+ 3 more voltmeters)
- Adapt our little DAQ program
- No fundamental (architectural) change to our DAQ

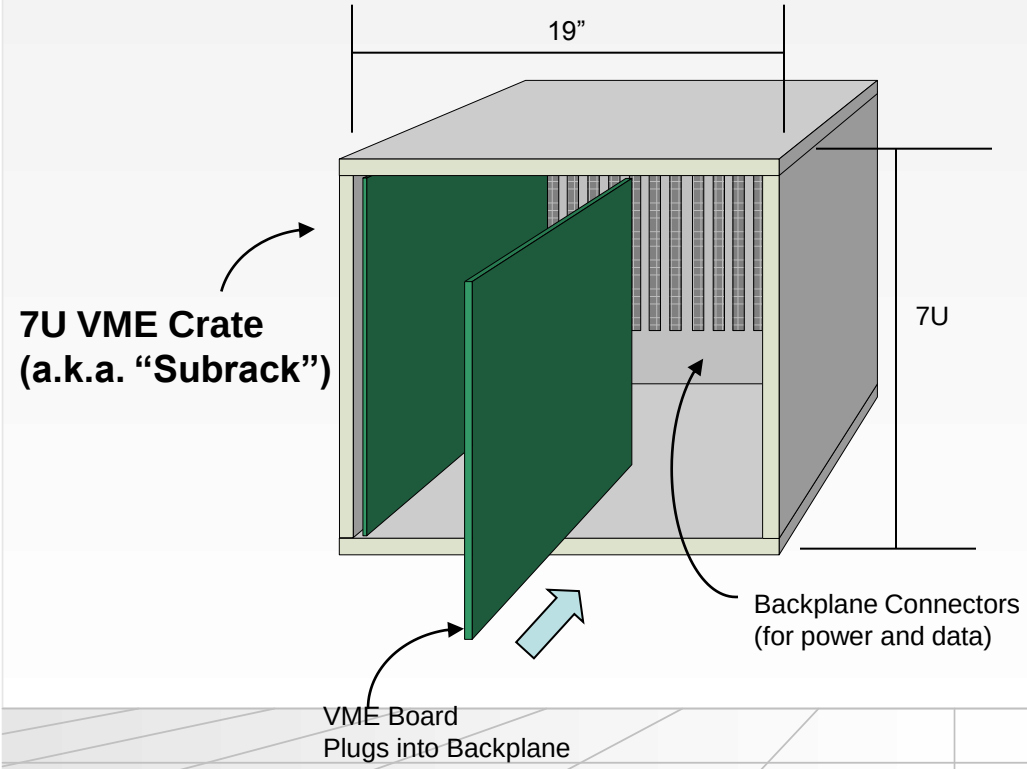
Read-out 160 Sensors

- For a moment we (might) consider to buy 52 USB hubs, 160 Voltmeters
- …but after a while we abandon the idea very soon before we start cabling this!
- Expensive, cumbersome, fragile → our data acquisition is *not scalable*



Read-out with Buses

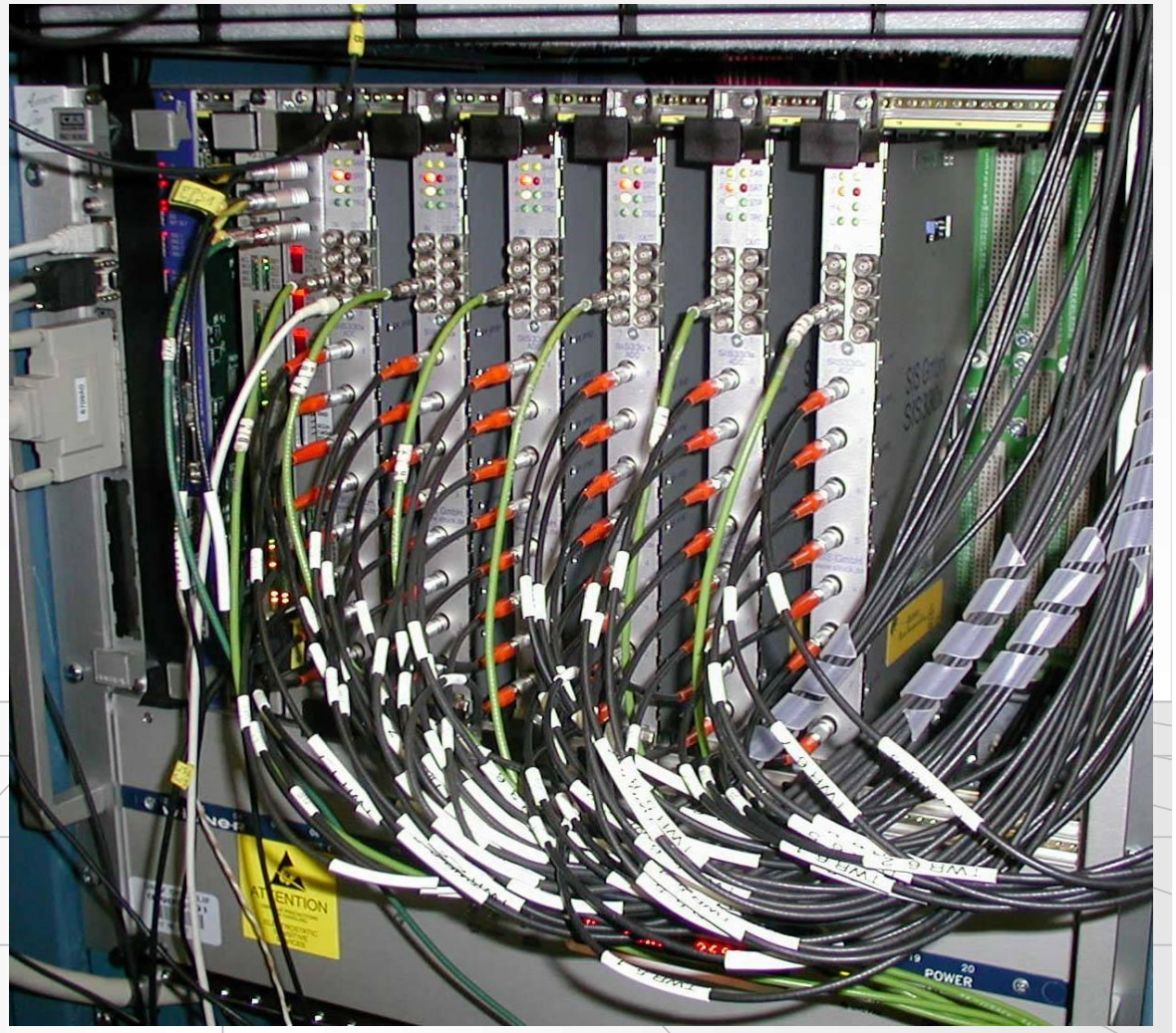
A Better DAQ for Many (temperature) Sensors



- Buy or build a compact multi-port volt-meter module, e.g. 16 inputs
- Put many of these multi-port modules together in a common chassis or crate
- The modules need
  - Mechanical support
  - Power
  - A standardized way to access their data (our measurement values)
- All this is provided by standards for (readout) electronics such as VME (IEEE 1014)

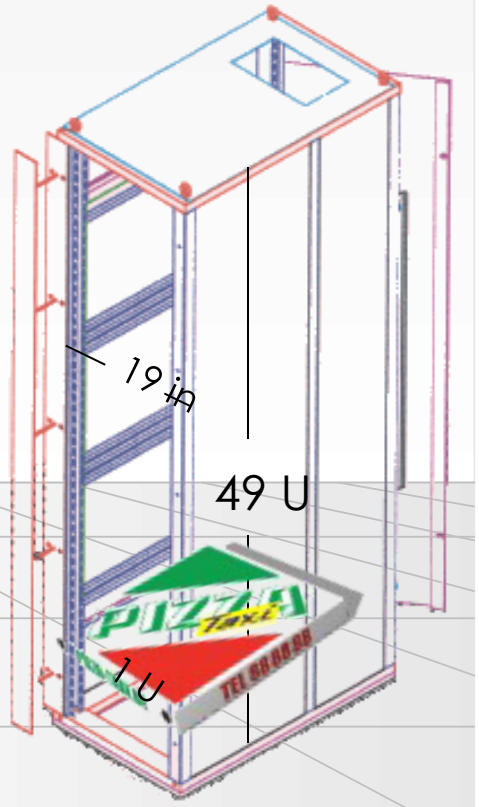
DAQ for 160 Sensors Using VME

- Readout boards in a *VME-crate*
  - mechanical standard for
  - electrical standard for power on the backplane
  - signal and protocol standard for communication on a *bus*



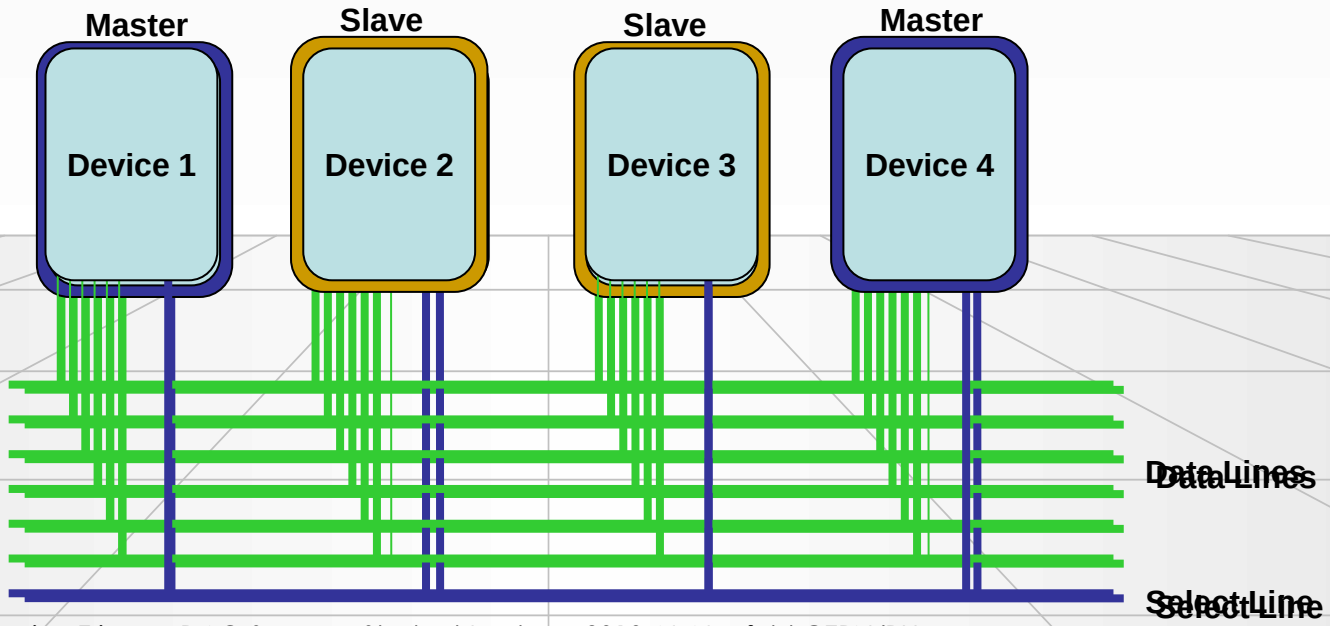
A Word on Mechanics and Pizzas

- The width and height of racks and crates are measured in US units: inches (in, '') and U
  - 1 in = 25.4 mm
  - 1 U = 1.75 in = 44.45 mm
- The width of a "standard" rack is 19 in.
- The height of a crate (also sub-rack) is measured in Us
- Rack-mountable things, in particular computers, which are 1 U high are often called *pizza-boxes*
- At least in Europe, the depth is measured in mm
- Gory details can be found in IEEE 1101.x (VME mechanics standard)



Communication in a Crate: Buses

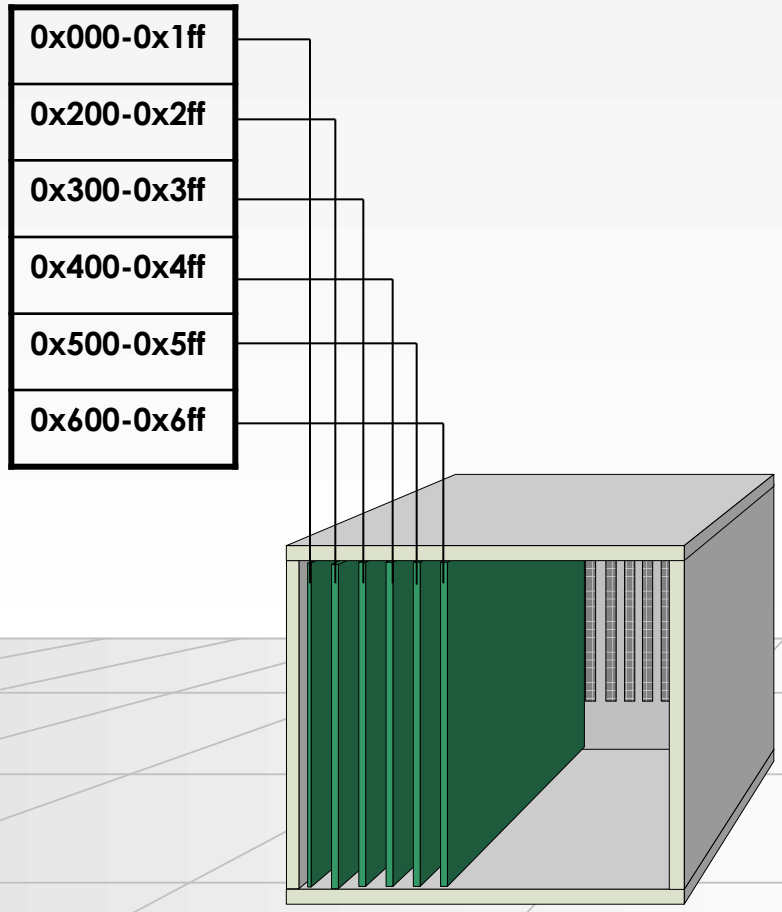
- A bus connects two or more devices and allows the to communicate
- The bus is shared between all devices on the bus → arbitration is required
- Devices can be masters or slaves (some can be both)
- Devices can be uniquely identified ("addressed") on the bus



Buses

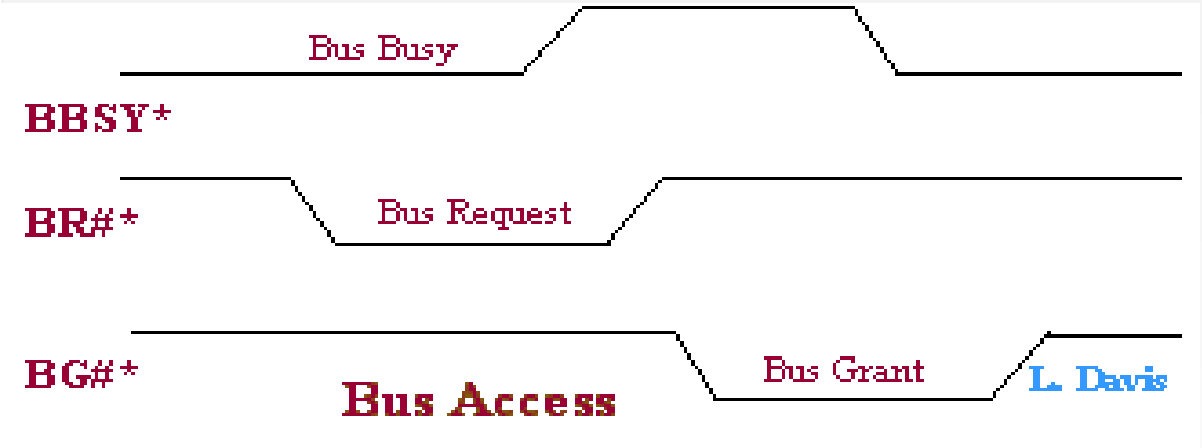
- Famous examples: PCI, USB, VME, SCSI
  - older standards: CAMAC, ISA
  - upcoming: ATCA
  - many more: FireWire, I2C, Profibus, etc…
- Buses can be
  - local: PCI
  - external peripherals: USB
  - in crates: VME, compactPCI, ATCA
  - long distance: CAN, Profibus

The VME Bus



- In a VME crate we can find three main types of modules
  - The controller which monitors and arbitrates the bus
  - Masters read data from and write data to slaves
  - Slaves send data to and receive data from masters
- Addressing of modules
  - In VME each module occupies a part of a (flat) range of addresses (24 bit to 32 bit)
  - Address range of modules is hardwired (conflicts!)

VME protocol 1) Arbitration

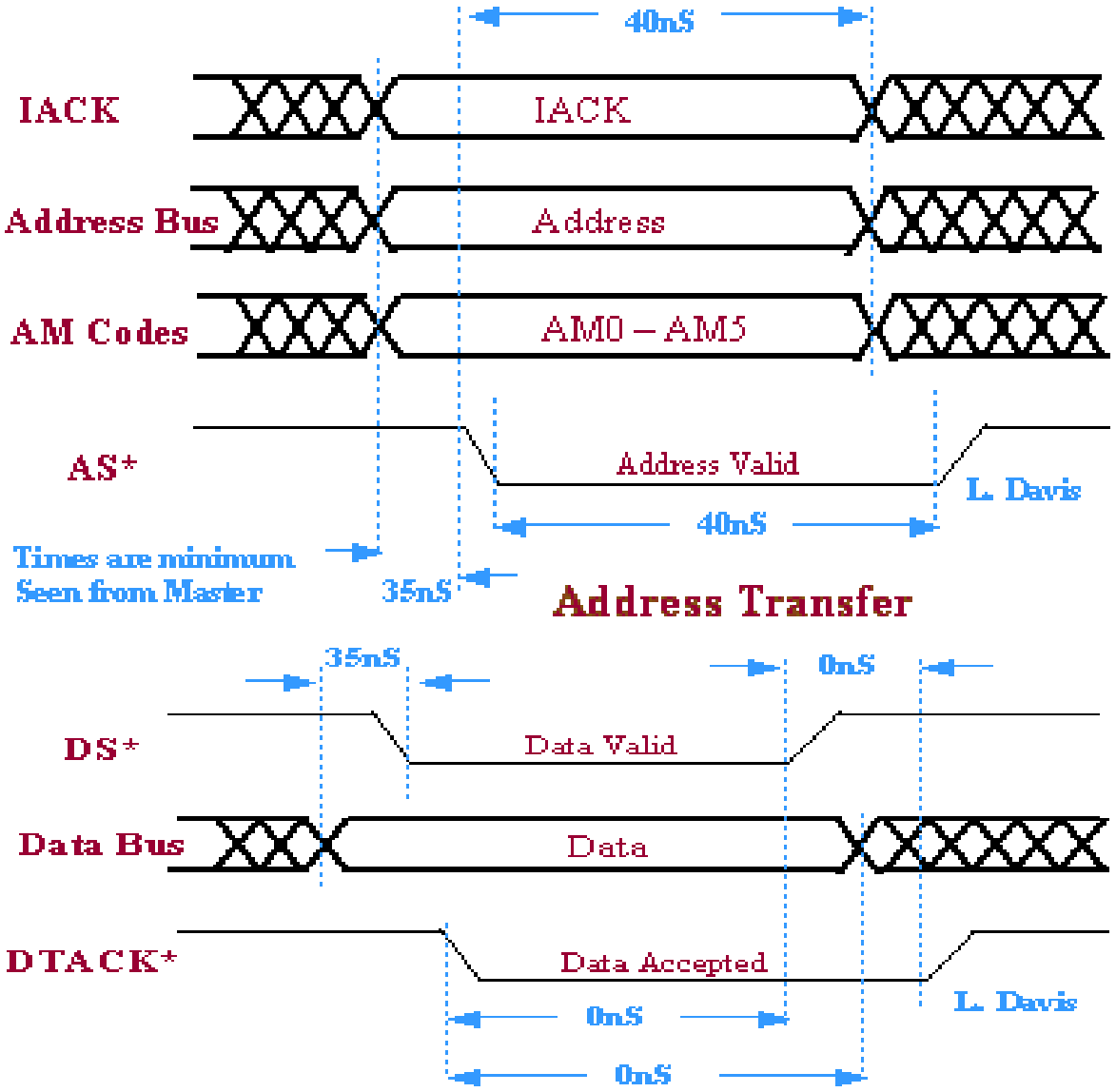


- Arbitration: Master asserts*) BR#, Controller answers by asserting BG#
- If there are several masters requesting at the same time the one physically closest to the controller wins
- The winning master drives BBSY* high to indicate that the bus is now in use

Pictures from http://www.interfacebus.com

*) assert means driving the line to logical 0 (VME control lines are inverted or active-low)

VME protocol 2) Write transfer

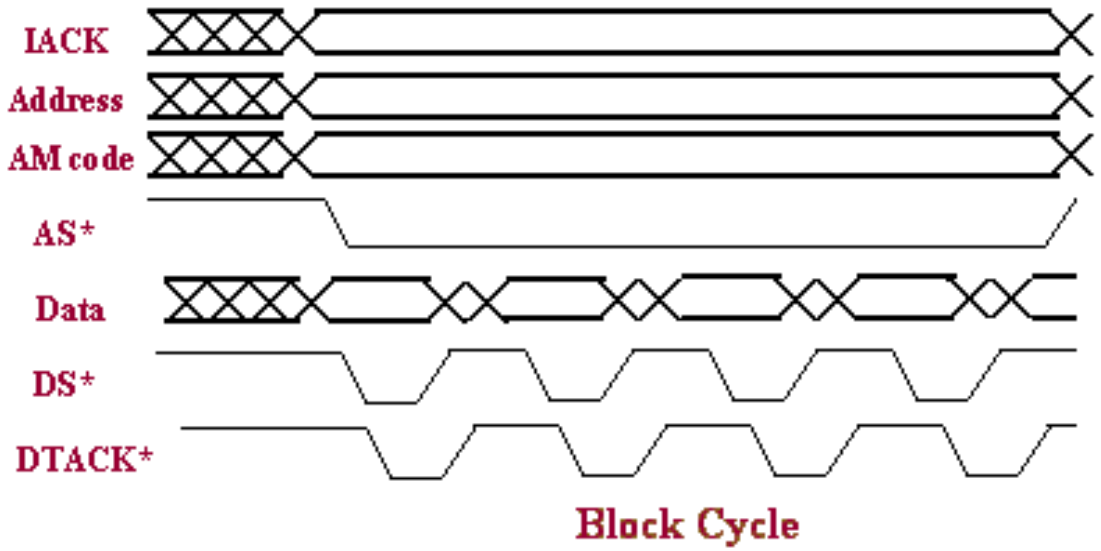


- The Master writes data and address to the data / respectively data bus
- It asserts DS* and AS* to signal that the data and address are valid
- The slave reads and acknowledges by asserting DTACK
- The master releases DS*, AS* and BSBSY*, the cycle is complete
- Note: there is no clock! The slave can respond whenever it wants. VME is an asynchronous bus

Speed Considerations

- Theoretically ~ 16 MB/s can be achieved
  - assuming the databus to be full 32-bit wide
  - the master never has to relinquish bus master ship
- Better performance by using block-transfers

VME protocol 3) Block transfer



Block Cycle

- Block transfers are essential for Direct Memory Access (DMA)
- More performance can be gained by using the address bus also for data (VME64)

- After an address cycle several (up to 256) data cycles are performed
- The slave is supposed to increment the address counter
- The additional delays for asserting and acknowledging the address are removed
- Performance goes up to 40 MB/s
- In PCI this is referred to as "burst-transfer"

Advantages of buses

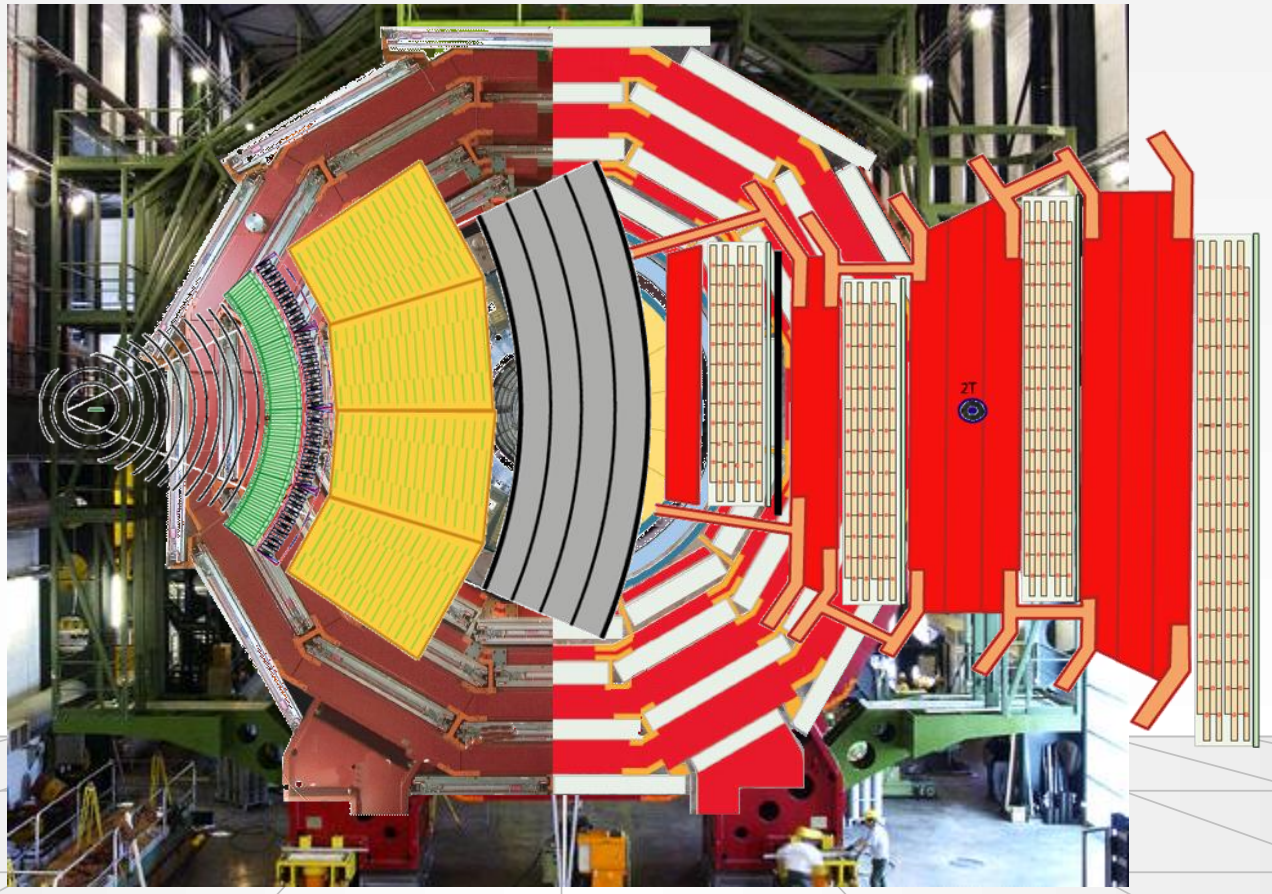
- Relatively simple to implement
  - Constant number of lines
  - Each device implements the same interface
- Easy to add new devices
  - topological information of the bus can be used for automagically choosing addresses for bus devices: this is what plug and play is all about.

Buses for DAQ at LHC?

- A bus is shared between all devices (each new active device slows everybody down)
  - Bus-width can only be increased up to a certain point (128 bit for PC-system bus)
  - Bus-frequency (number of elementary operations per second) can be increased, but decreases the physical bus-length
- Number of devices and physical bus-length is limited (scalability!)
  - For synchronous high-speed buses, physical length is correlated with the number of devices (e.g. PCI)
  - Typical buses have a lot of control, data and address lines (look at a SCSI or ATA cable)
- Buses are typically useful for systems < 1 GB/s

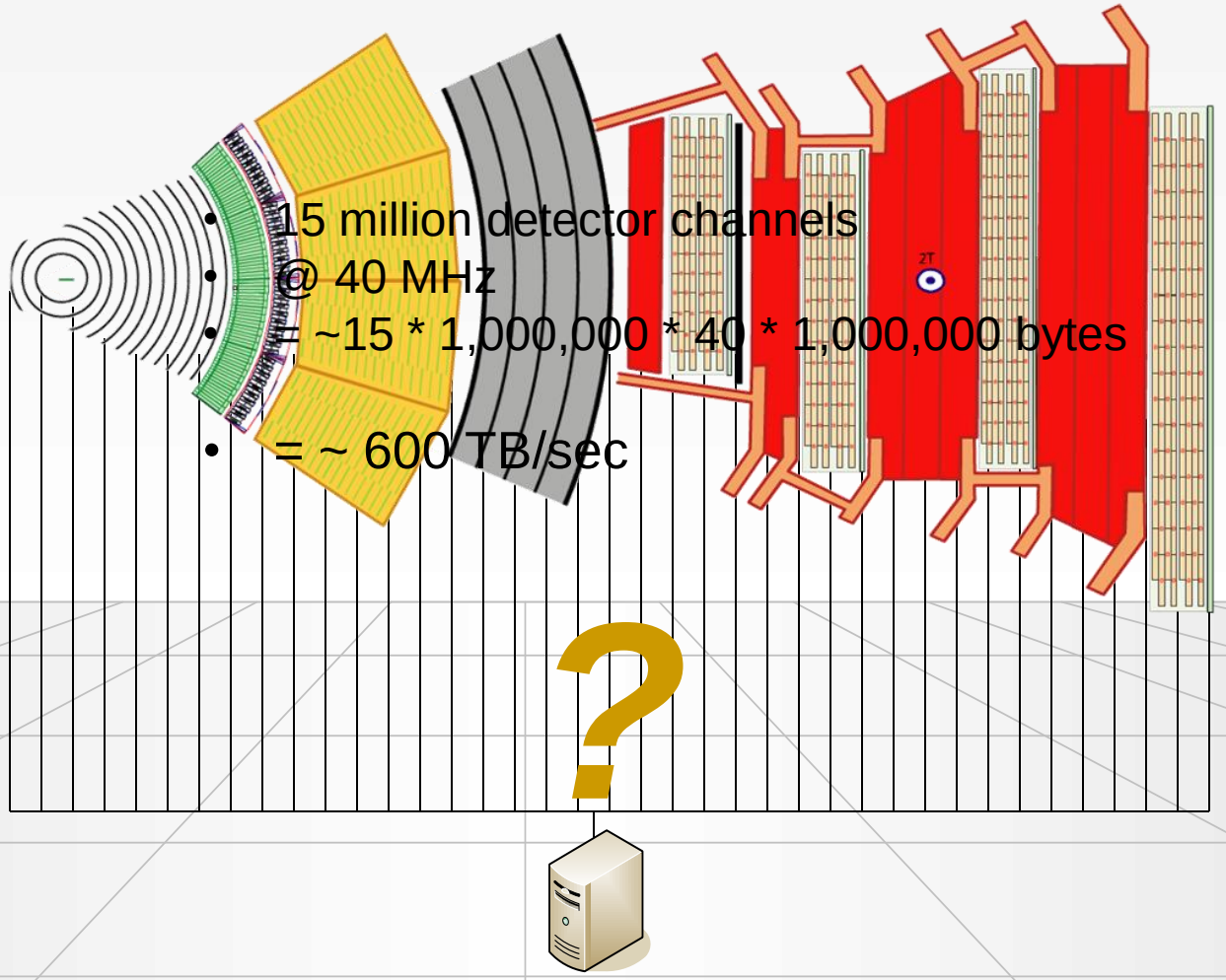
Data Acquisition for a Large Experiment

Moving on to Bigger Things…



The CMS Detector

Moving on to Bigger Things…



- 15 million detector channels
- @ 40 MHz
- = ~15 * 1,000,000 * 40 * 1,000,000 bytes
- = ~ 600 TB/sec

Designing a DAQ System for a Large HEP Experiment

- What defines "large"?
  - The number of channels: for LHC experiments O(10^7) channels
    - a (digitized) channel can be between 1 and 14 bits
  - The rate: for LHC experiments everything happens at 40.08 MHz, the LHC bunch crossing frequency (This corresponds to 24.9500998 ns or 25 ns among friends)
- HEP experiments usually consist of many different sub-detectors: tracking, calorimetry, particle-ID, muon-detectors

First Questions

- Can we or do we want to save all the data?
- How do we select the data
- Is continuous read-out needed, i.e. an experiment in a collider? Or are there idle periods mixed with periods with many events – this is typically the case for fixed-target experiments
- How do we make sure that the values from the many different channels refer to the same original event (collision)

What Do We Need to Read Out a Detector (successfully)?

- A selection mechanism ("trigger")
- Electronic readout of the sensors of the detectors ("front-end electronics")
- A system to keep all those things in sync ("clock")
- A system to collect the selected data ("DAQ")
- A Control System to configure, control and monitor the entire DAQ
- Time, money, students

Large DAQ

Data Acquisition

- Event-data are now digitized, pre-processed and tagged with a unique, monotonically increasing number
- The event data are distributed over many *read-out boards* ("sources")
- For the next stage of selection, or even simply to write it to tape we have to get the pieces together: enter the DAQ

Network based DAQ

- In large (HEP) experiments we typically have thousands of devices to read, which are sometimes very far from each other → *buses can not do that*
- Network technology solves the scalability issues of buses
  - In a network devices are equal ("peers")
  - In a network devices communicate directly with each other
    - no arbitration necessary
    - bandwidth guaranteed
  - data and control use the same path
    - much fewer lines (e.g. in traditional Ethernet only two)
  - At the signaling level buses tend to use parallel copper lines. Network technologies can be also optical, wire-less and are typically (differential) serial

Network Technologies

- Examples:
  - The telephone network
  - Ethernet (IEEE 802.3)
  - ATM (the backbone for GSM cell-phones)
  - Infiniband
  - Myrinet
  - many, many more
- Note: some of these have "bus"-features as well (Ethernet, Infiniband)
- Network technologies are sometimes functionally grouped
  - Cluster interconnect (Myrinet, Infiniband) 15 m
  - Local area network (Ethernet), 100 m to 10 km
  - Wide area network (ATM, SONET) > 50 km

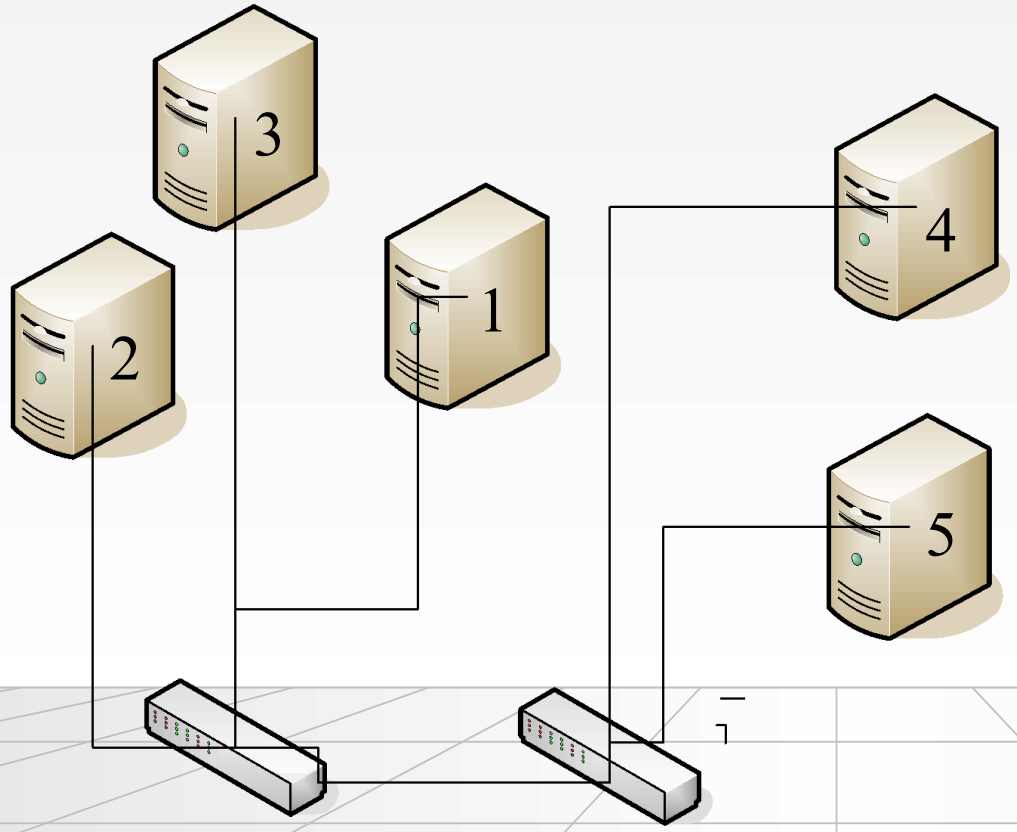
Connecting Devices in a Network

- On an network a device is identifed by a network address
  - eg: our phone-number, the MAC address of your computer
- Devices communicate by sending messages (frames, packets) to each other
- Some establish a connection lilke the telephone network, some simply send messages
- Modern networks are *switched with point-to-point links*
  - circuit switching, packet switching

Switched Networks

- In a switched network each node is connected either to another node or to a switch
- Switches can be connected to other switches
- A path from one node to another leads through 1 or more switches (this number is sometimes referred to as the number of "hops")

A Switched Network



- While 2 can send data to 1 and 4, 3 can send at full speed to 5
- 2 can distribute the share the bandwidth between 1 and 4 as needed

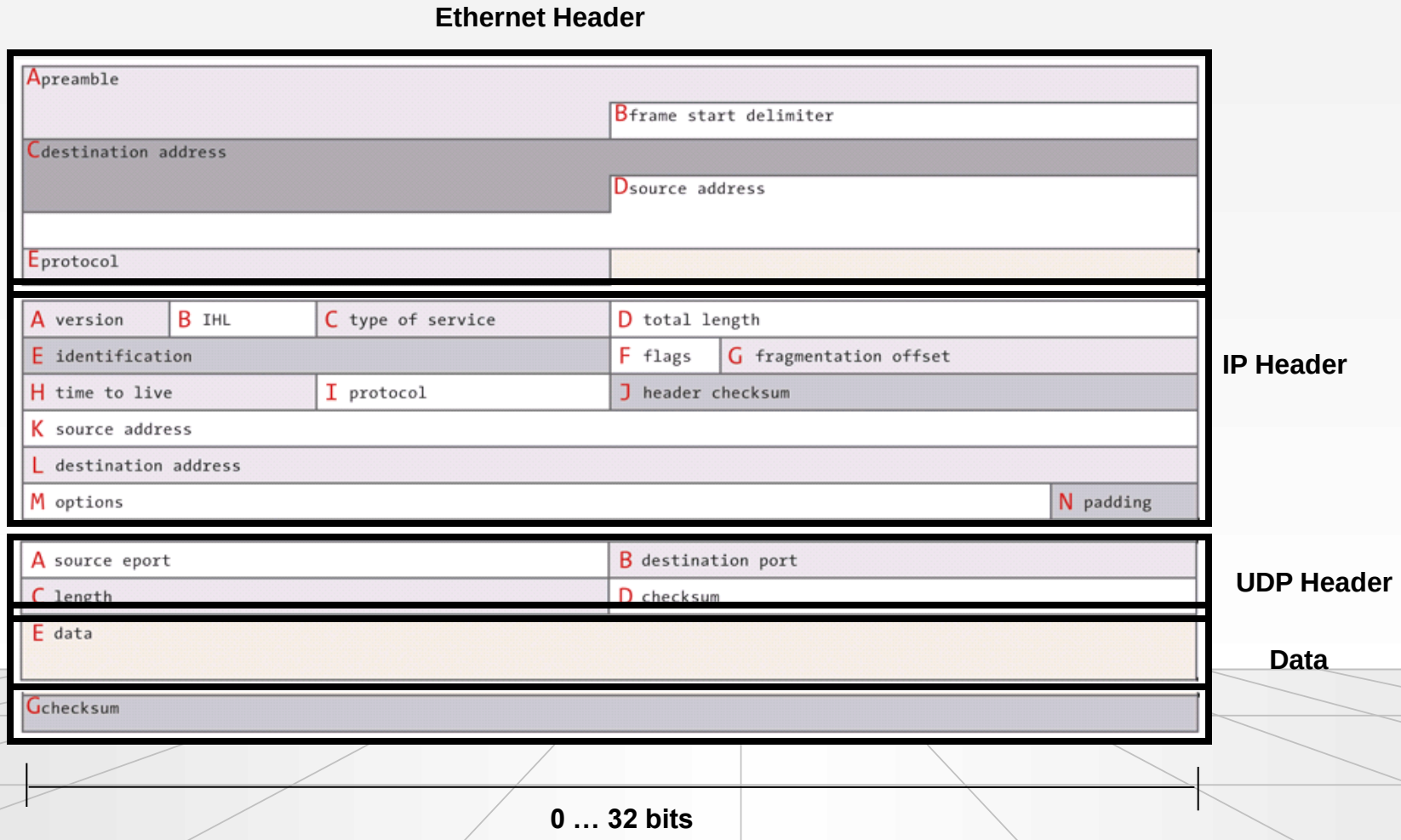
Switches

- Switches are the key to good network performance
- They must move frames reliably and as fast as possible between nodes
- They face two problems
  - Finding the right path for a frame
  - Handling congestion (two or more frames want to go to the same destination at the same time)

Ethernet

- Cheap
- Unreliable – but in practice transmission errors are very low
- Available in many different speeds and physical media
- We use IP or TCP/IP over Ethernet
- By far the most widely used local area network technology (even starting on the WAN)

IP Packets over Ethernet



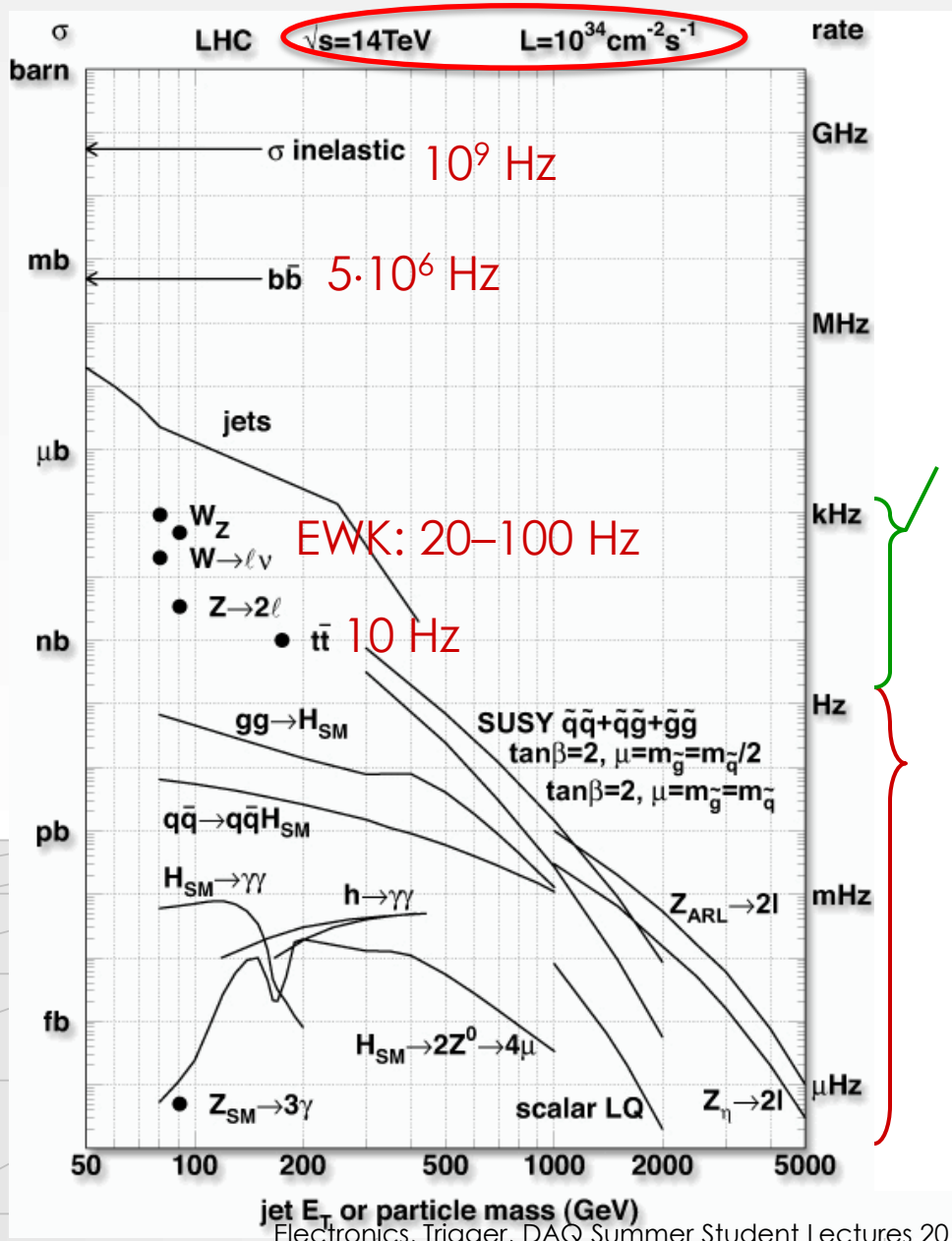
Lecture 5/5

Trigger and Data Acquisition for LHC experiments

Building a trigger (recap)

- Keep it simple! (Remember Einstein: "As simple as possible, but not simpler")
- Even though "premature optimization is the root of all evil", think about efficiency (buffering)
- Try to have few adjustable parameters: scanning for a good working point will otherwise be a night-mare

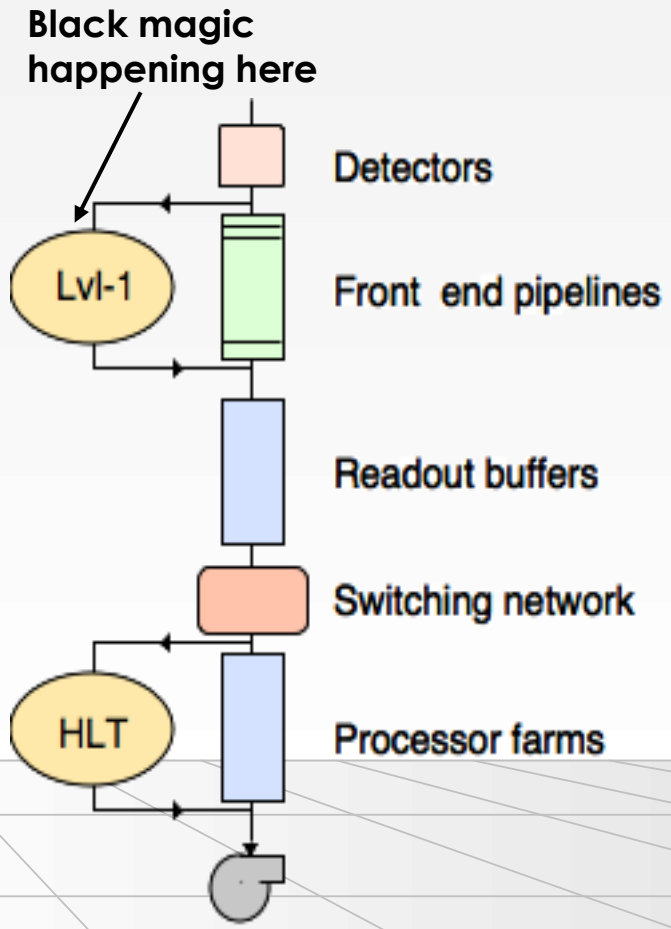
Should we read everything?



- A typical collision is "boring"
  - Although we need also some of these "boring" data as cross-check, calibration tool and also some important "low-energy" physics
- "Interesting" physics is about 6–8 orders of magnitude rarer (EWK & Top)
- "Exciting" physics involving new particles/discoveries is ≥ 9 orders of magnitude below σ_{tot}
  - 100 GeV Higgs 0.1 Hz
  - 600 GeV Higgs 0.01 Hz
- We *just* ☺ need to efficiently identify these rare processes from the overwhelming background before reading out & storing the whole event

Trigger for LHC

- No (affordable) DAQ system could read out O(10^7) channels at 40 MHz → 400 TBit/s to read out – even assuming binary channels!
- What's worse: most of these millions of events per second are totally uninteresting: one Higgs event every 0.02 seconds
- A *first level trigger (Level-1, L1)* must somehow select the more interesting events and tell us which ones to deal with any further

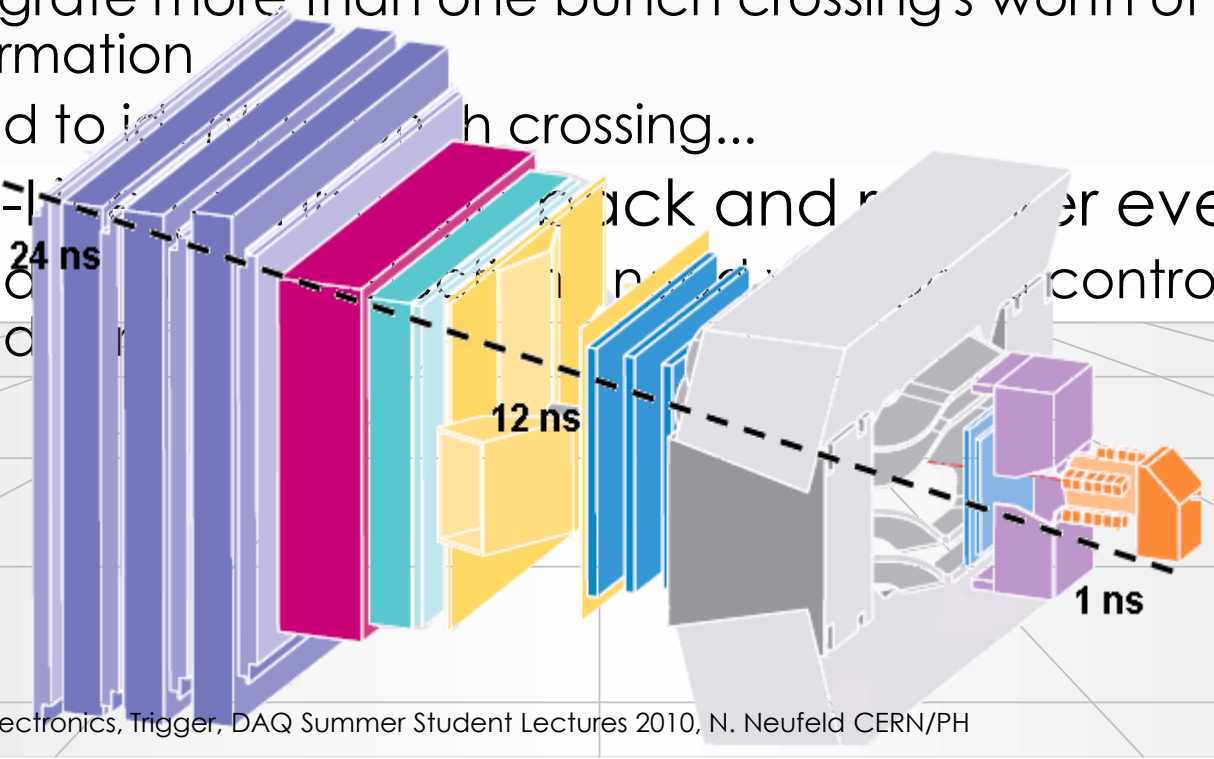


Inside the Box: How does a Level-1trigger work?

- Millions of channels →: try to work as much as possible with "local" information
  - Keeps number of interconnections low
- Must be fast: look for "simple" signatures
  - Keep the good ones, kill the bad ones
  - Robust, can be implemented in hardware (fast)
- Design principle:
  - fast: to keep buffer sizes under control
  - every 25 nanoseconds (ns) a new event: have to decide within a few microseconds (μs): trigger-latency

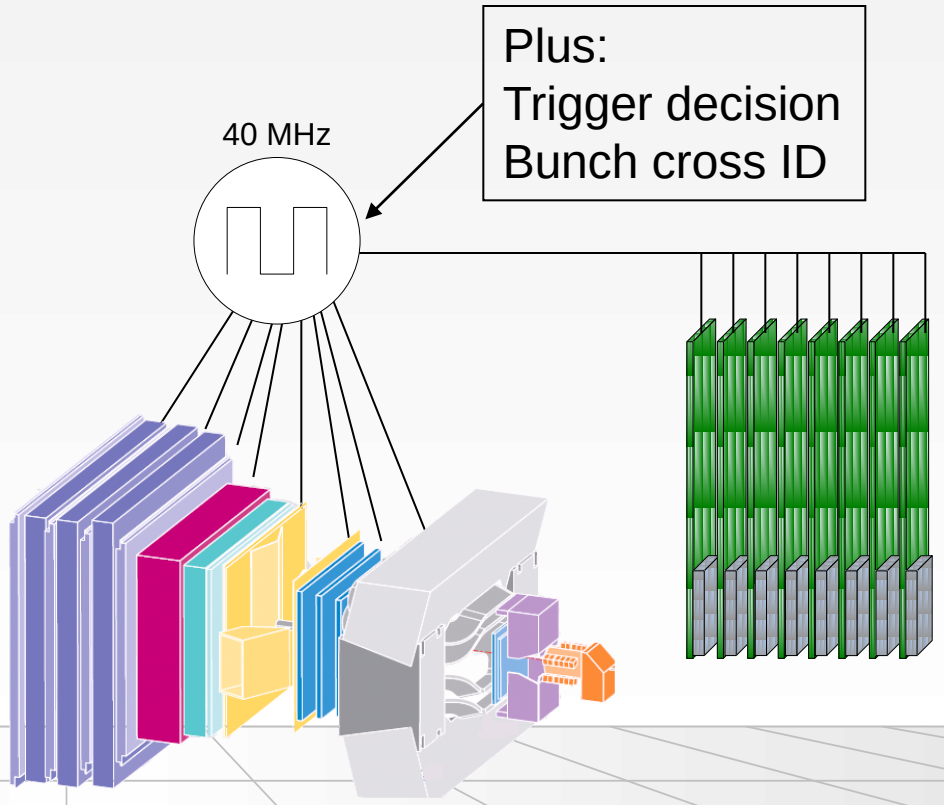
Challenges for the L1 at LHC

- N (channels) ~ O(10^7); ≈20 interactions every 25 ns
  - need huge number of connections
- Need to synchronize detector elements to (better than) 25 ns
- In some cases: detector signal/time of flight > 25 ns
  - integrate more than one bunch crossing's worth of information
  - need to i… h crossing...
- It's On-l… ack and r… er events)
  - need… control over all cond…



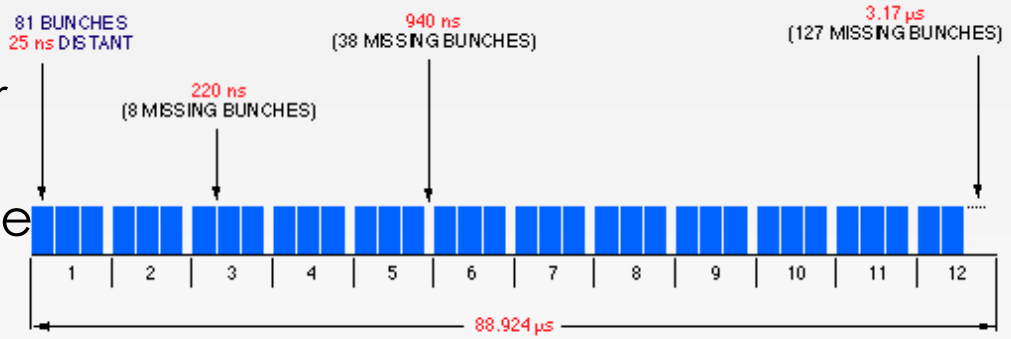
Clock Distribution and Synchronisation

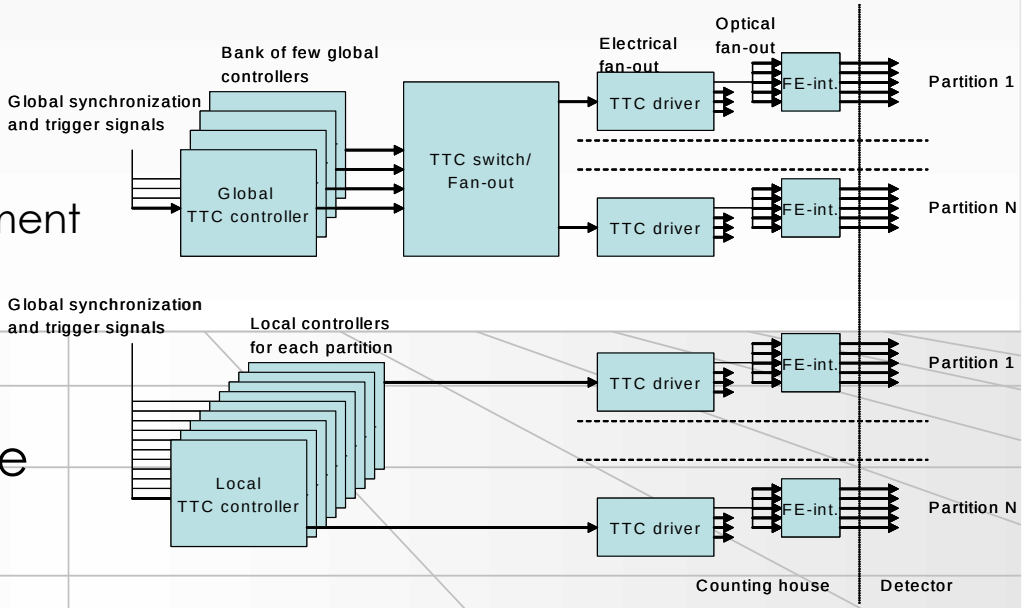
- An *event* is a snapshot of the values of all detector front-end electronics elements, which have their value caused by the same collision
- A common clock signal must be provided to all detector elements
  - Since the c is constant, the detectors are large and the electronics is fast, the detector elements must be carefully time-aligned
- Common system for all LHC experiments TTC based on radiation-hard opto-electronics



Timing & sync control

- Sampling clock with low jitter
- Synch reset
- Synchronization with machine bunch structure
- Calibration
- Trigger (with event type)
- Time align all the different sub-detectors and channels
  - Programmable delays
- Fan-out – unidirectional
  - Global fan-out to whole experiment or
  - Sub-detector fan-out
- Must be reliable as system otherwise may get de-synchronized which may take quite some time to correct



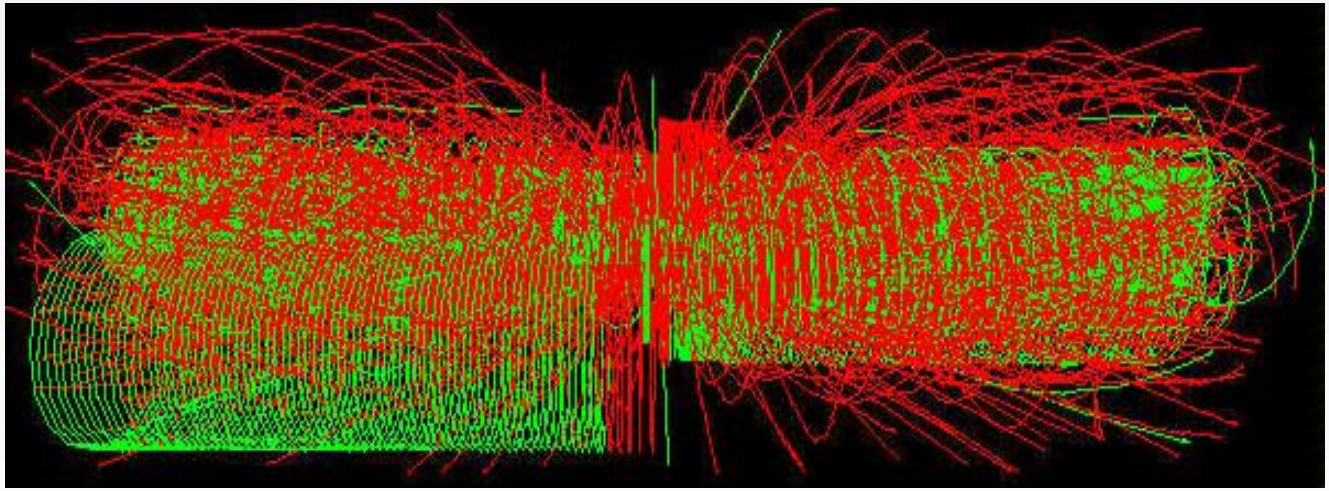


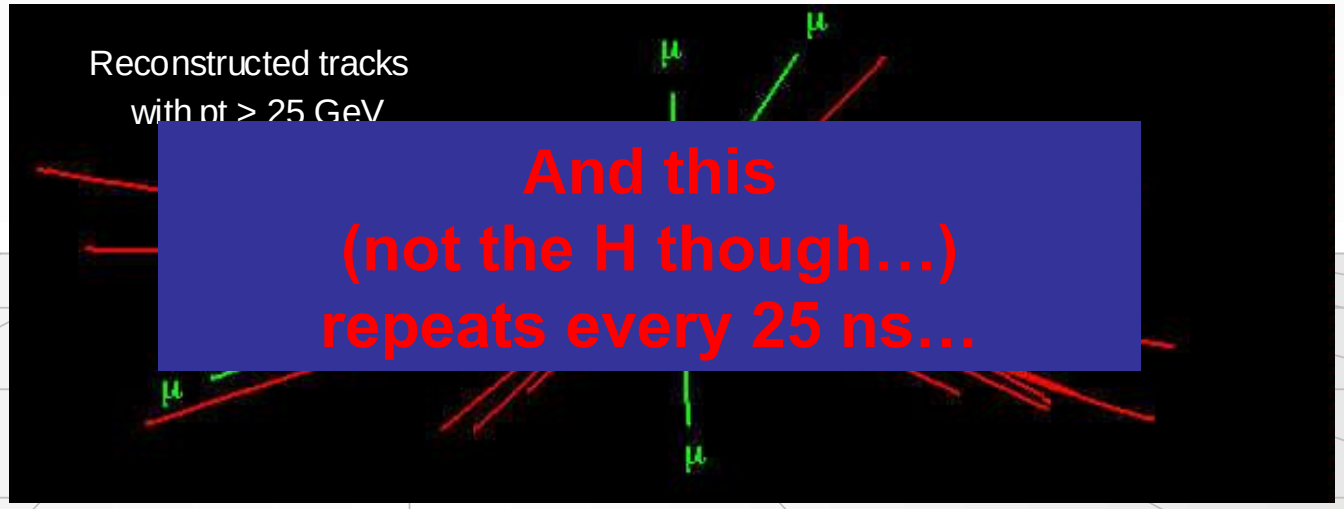
Know Your Enemy: pp Collisions at 14 TeV at 10^{34} $cm^{-2}s^{-1}$

- σ(pp) = 70 mb --> >7 x 10^8 /s (!)
- In ATLAS and CMS* 20 min bias events will overlap
- H→ZZ

Z →μμ

H→ 4 muons: the cleanest ("golden") signature



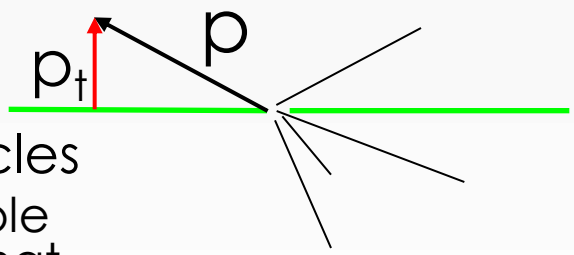


And this (not the H though…) repeats every 25 ns…

*)LHCb @2x10^{33} cm^{-2}-1 isn't much nicer and in Alice (PbPb) it will be even worse

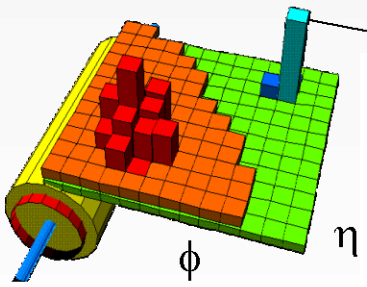
Mother Nature is a … Kind Woman After All

- pp collisions produce mainly hadrons with transverse momentum "p_t" ~1 GeV
- Interesting physics (old and new) has particles (leptons and hadrons) with large p_t:
  - $W \rightarrow e\nu$: $M(W)=80$ GeV/c^2; $p_t(e)$ ~ 30-40 GeV
  - $H(120\text{ GeV}) \rightarrow \gamma\gamma$: $p_t(\gamma)$ ~ 50-60 GeV
  - $B \rightarrow \mu D^{*+}\nu$ $p_t(\mu)$ ~ 1.4 GeV
- Impose high thresholds on the p_t of particles
  - Implies distinguishing particle types; possible for electrons, muons and "jets"; beyond that, need complex algorithms
- Conclusion: in the L1 trigger we need to watch out for high transverse momentum electrons, jets or muons

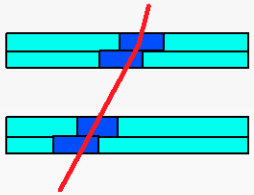


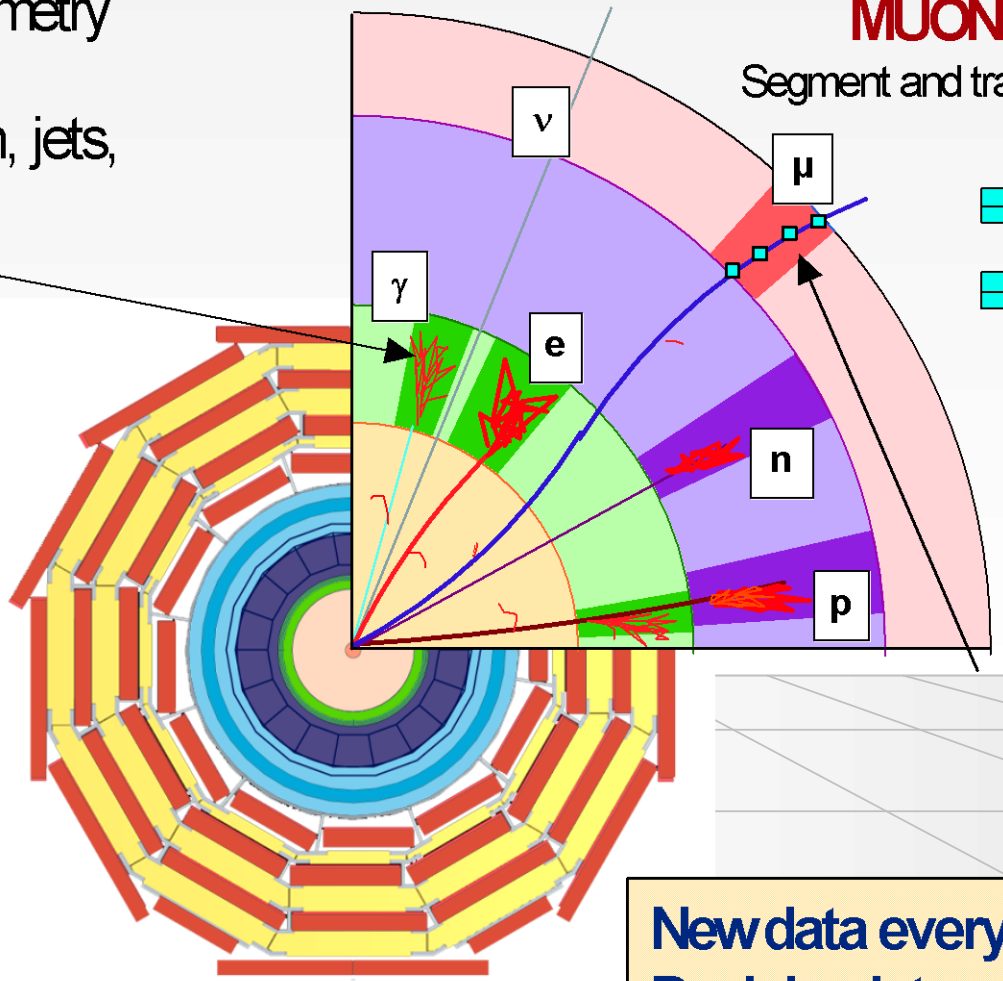
How to defeat minimum bias: transverse momentum p_t

Use prompt data (calorimetry and muons) to identify:
High p_t electron, muon, jets, missing E_T



MUON System
Segment and track finding

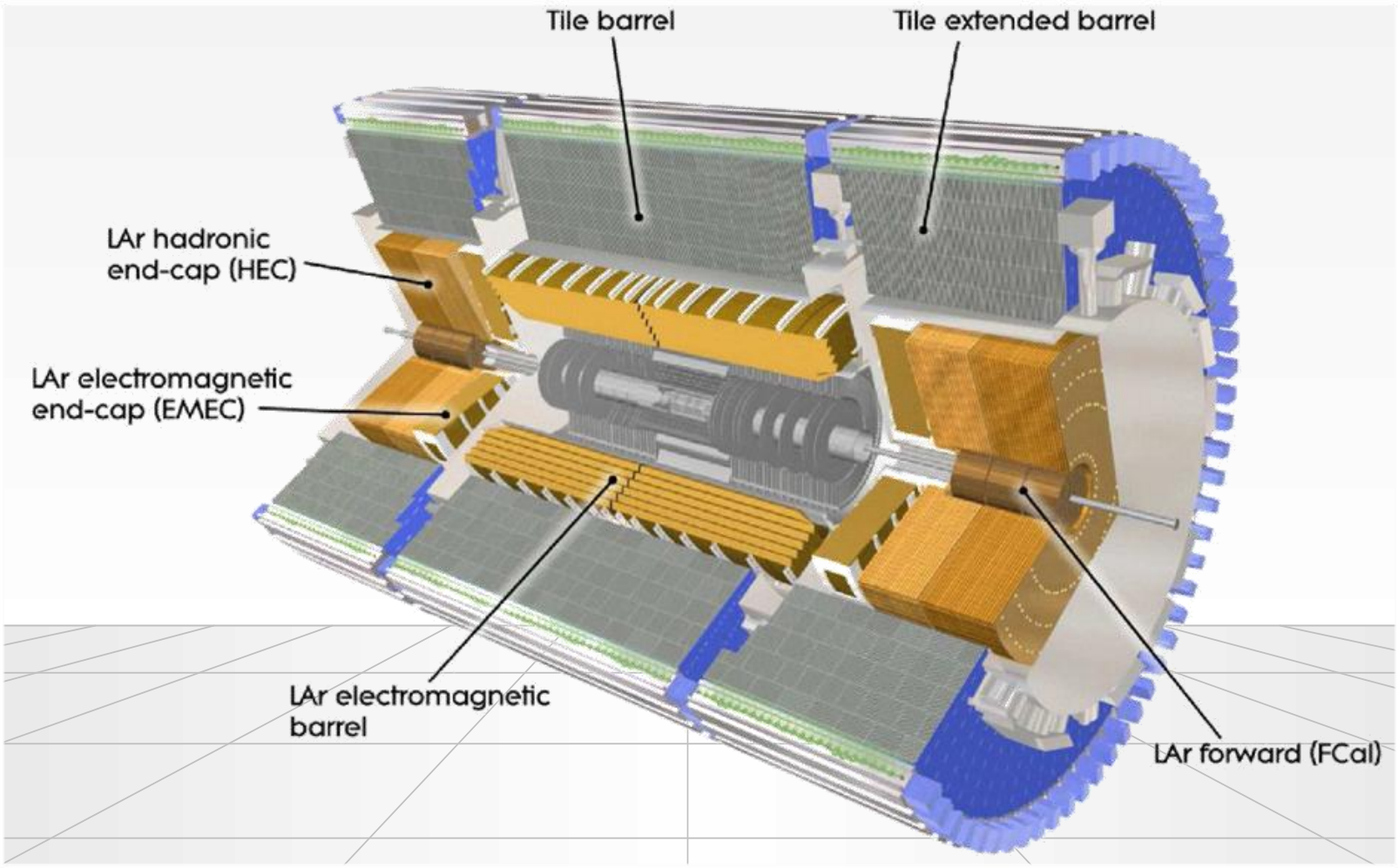




CALORIMETERs
Cluster finding and energy deposition evaluation

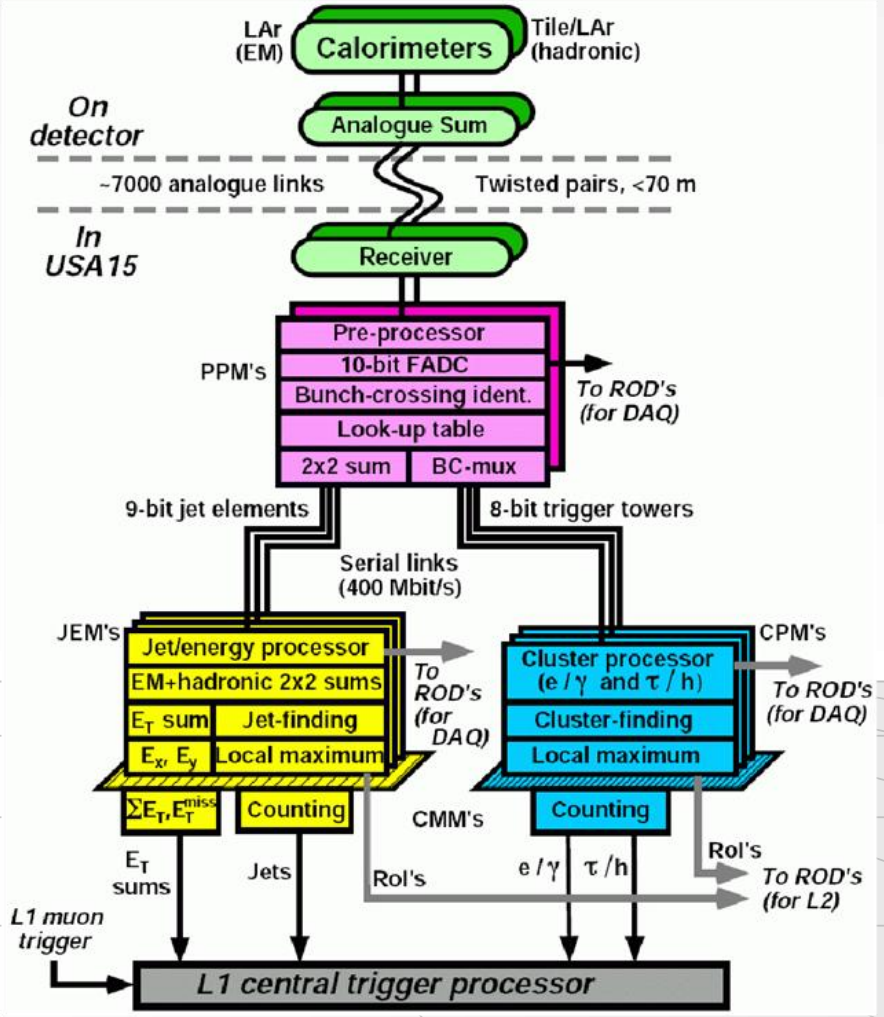
New data every 25 ns
Decision latency ~ µs

ATLAS Calorimeters



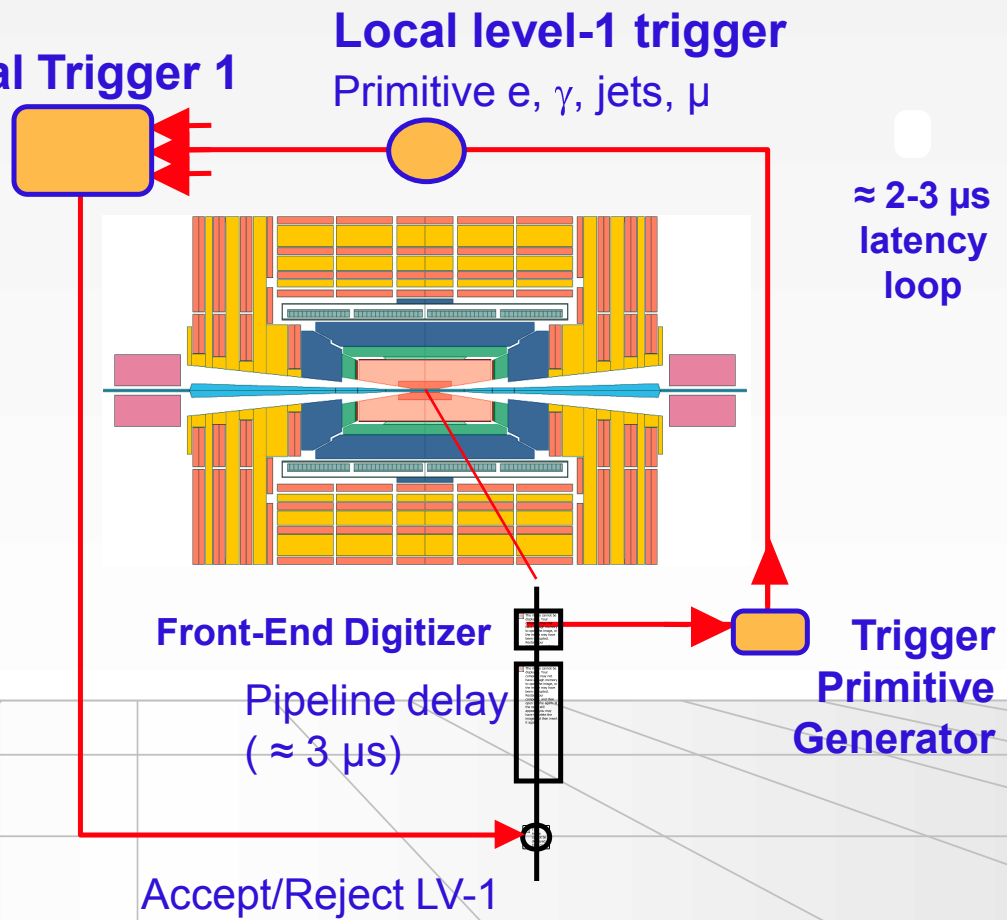
ATLAS L1 Calo Trigger

- Form analogue towers 0.1 x 0.1 ($\delta\eta$ x $\delta\varphi$)
- igitize, identify
- bunch-xing, Look-Up Table (LUT) → E_T
- Duplicate data to Jet/Energy-sum
- (JEP) and Cluster (CP) processors
- Send to CTP 1.5 μs after bunch-crossing ("x-ing").
- Store info at JEP and CP to seed next level of trigger

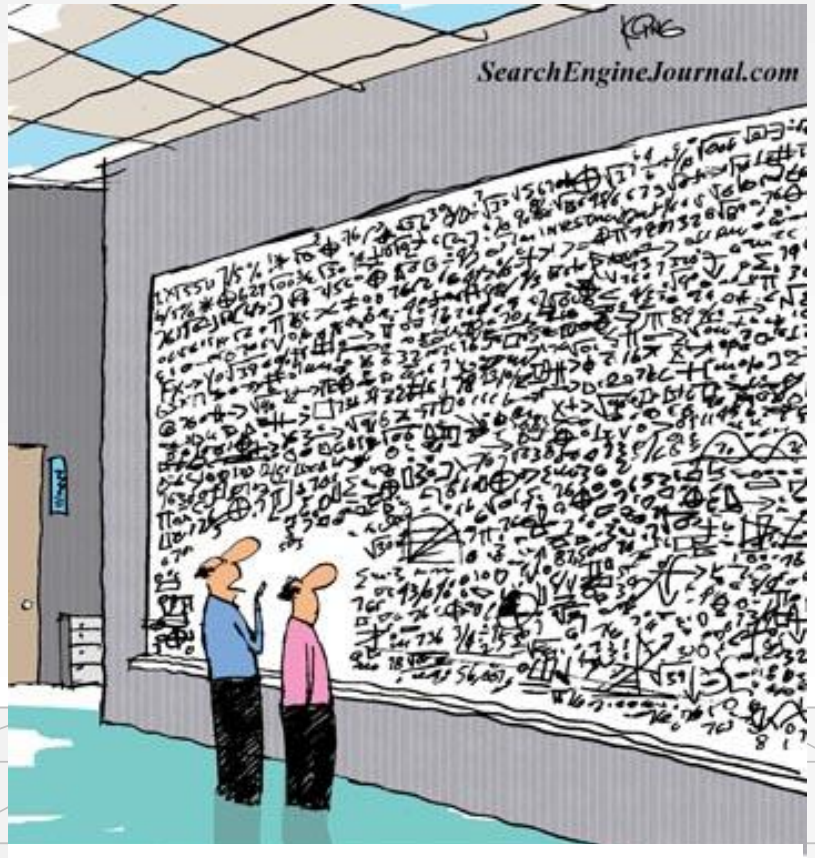


Distributing the L1 Trigger

- Assuming now that a magic box tells for each bunch crossing (clock-tick) yes or no
  - Triggering is not for philosophers – "perhaps" is not an option
- This decision has to be brought for each crossing to all the detector front-end electronics elements so that they can send of their data or discard it

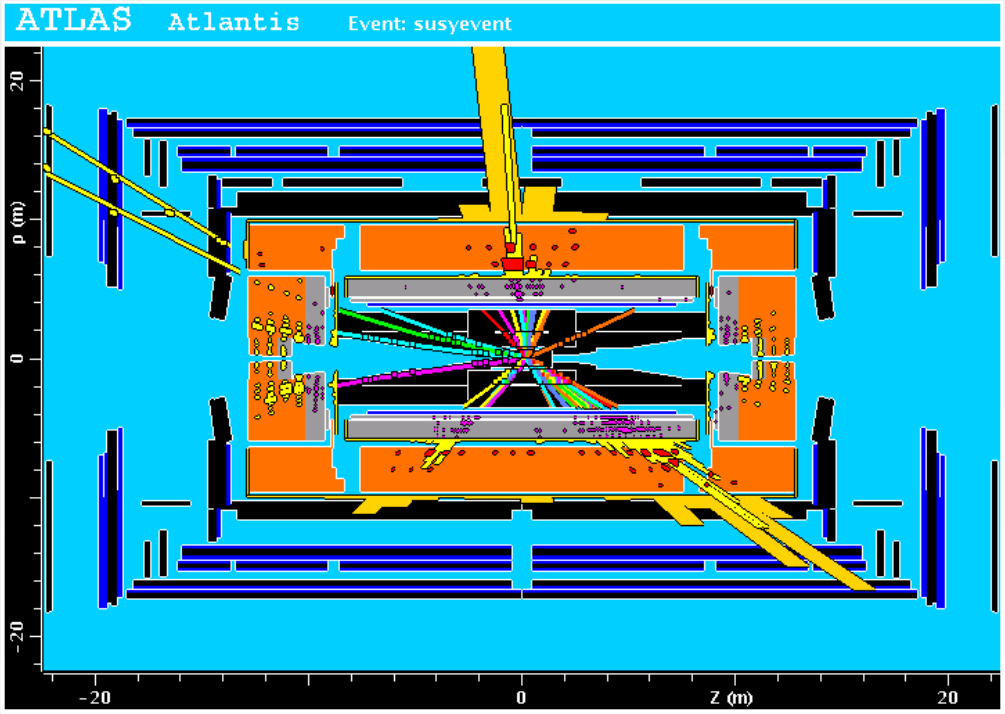


High Level Trigger

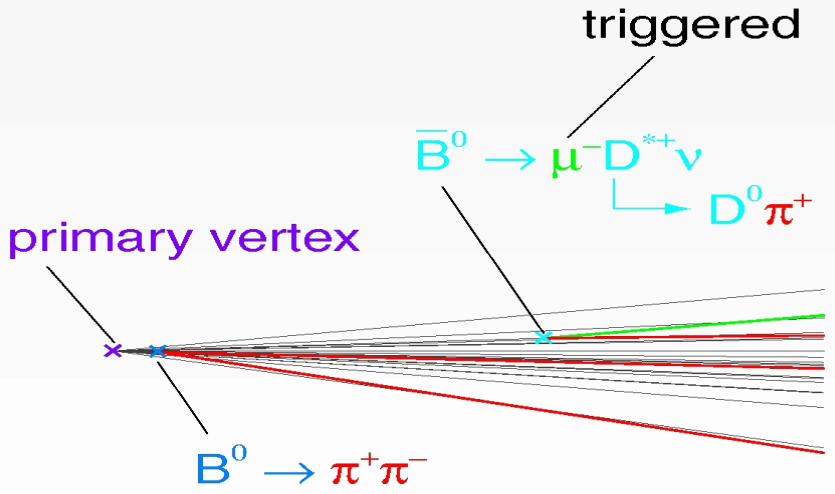


And that, in simple terms, is what we do in the High Level Trigger

High Level Trigger

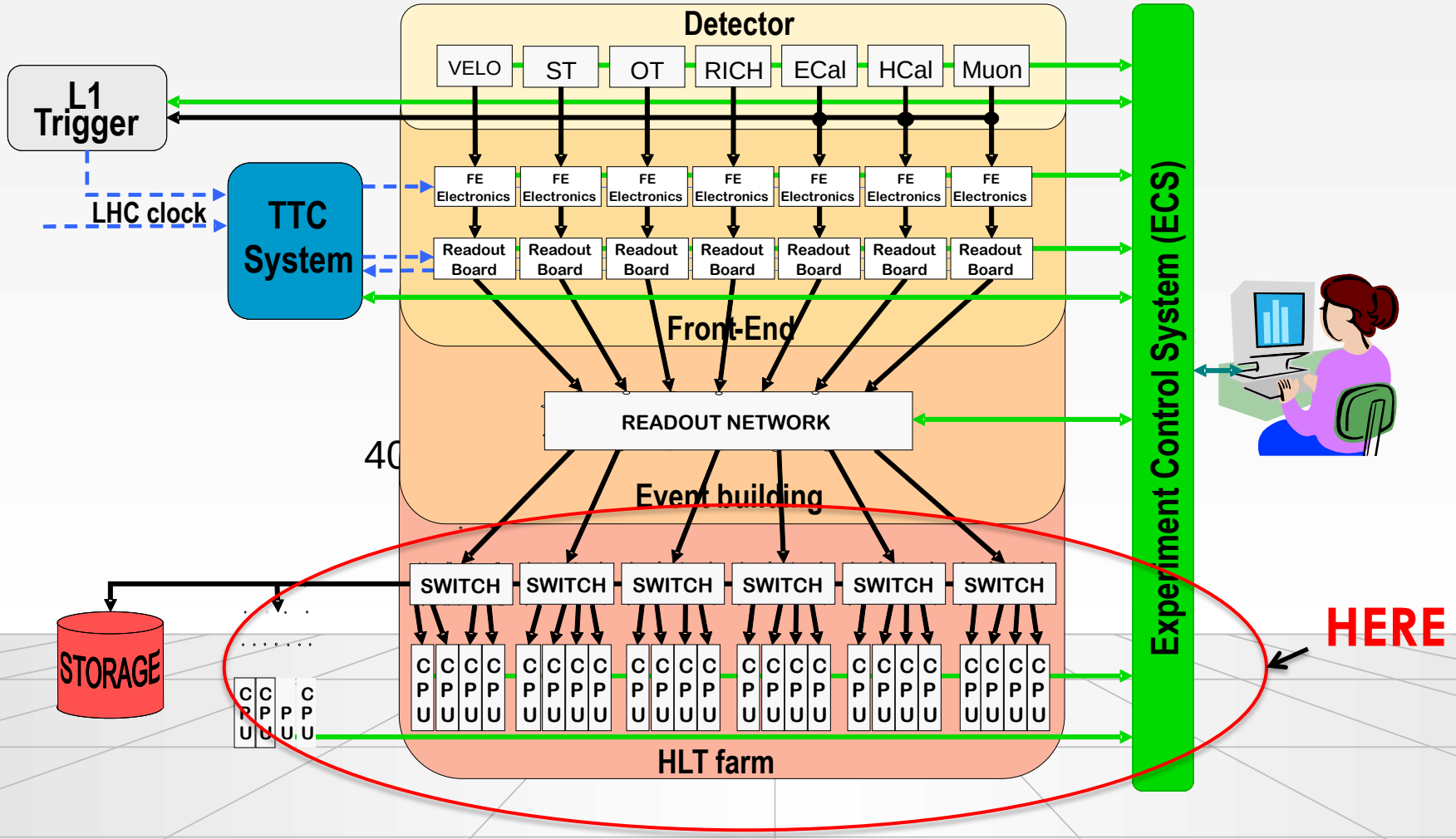


Complicated Event structure with hadronic jets (ATLAS) or secondary vertices (LHCb) require full detector information



Methods and algorithms are the same as for offline reconstruction (Lecture "From raw data to physics")

The High Level Trigger is …



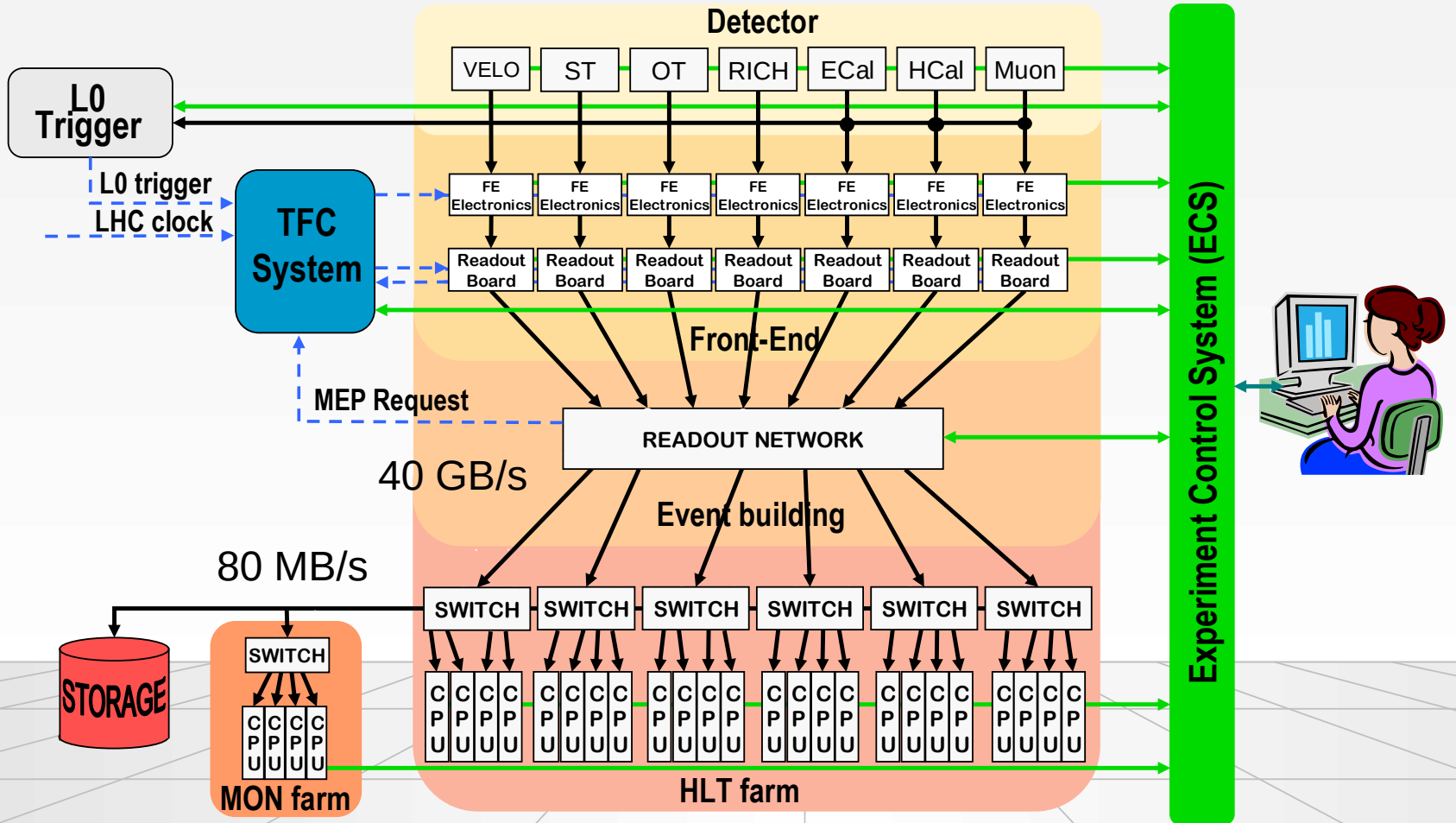
After L1: What's next?

- Where are we after L1
  - ATLAS and CMS : rate is ~75 to 100 kHz, event size ~ 1 - 2 MB
  - LHCb: rate is 1 MHz, event size 40 kB / ALICE: O(kHz) and O(GB)
- Ideally
  - Run the real full-blown physics reconstruction and selection algorithms
  - These application take O(s). Hence: even at above rates still need ***100 MCHF server farm (Intel will be happy!)***
- In Reality:
  - Start by looking at only part of the detector data **seeded by what triggered the 1st level**
  - LHCb: 1st level Trigger confirmation" algorithms: < 10 ms/event
  - Atlas: Region of Interest" (RoI): < 40 ms/event
- ➔ Reduce the rate by factor ~ 30, and then do offline analysis

Event Building

(providing the data for the High Level Trigger)

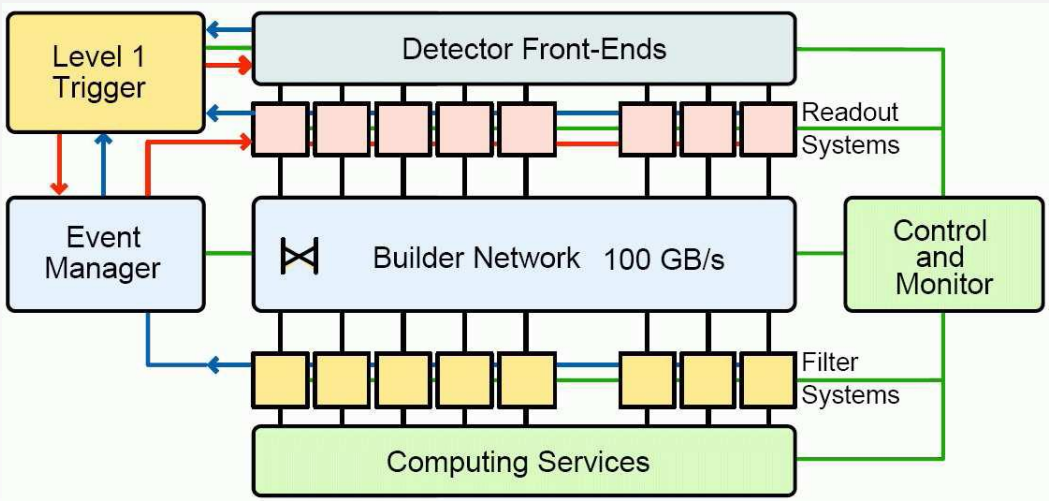
LHCb DAQ



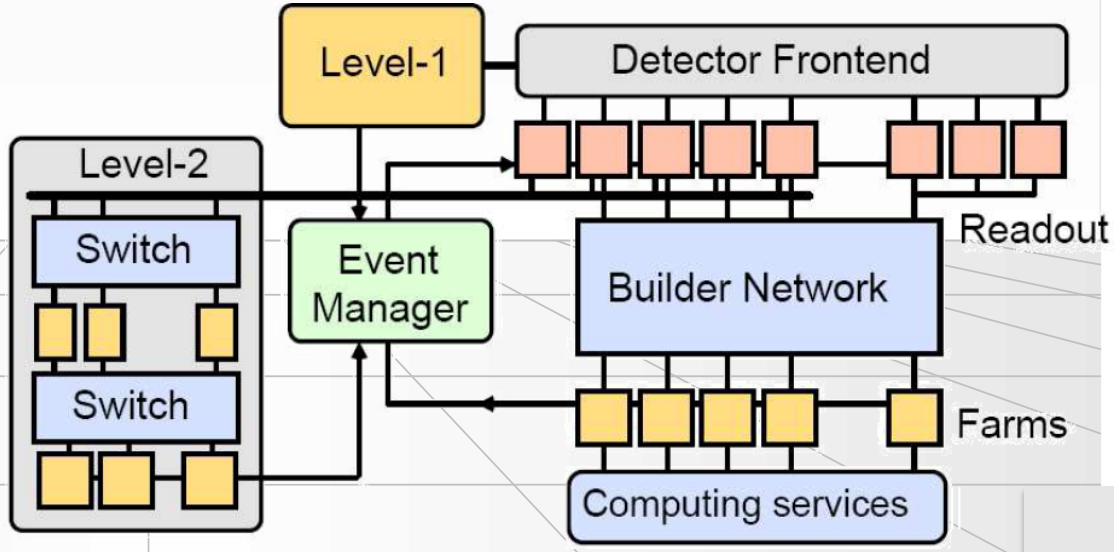
Average event size 40 kB
Average rate into farm 1 MHz
Average rate to tape 2 kHz

Two philosophies

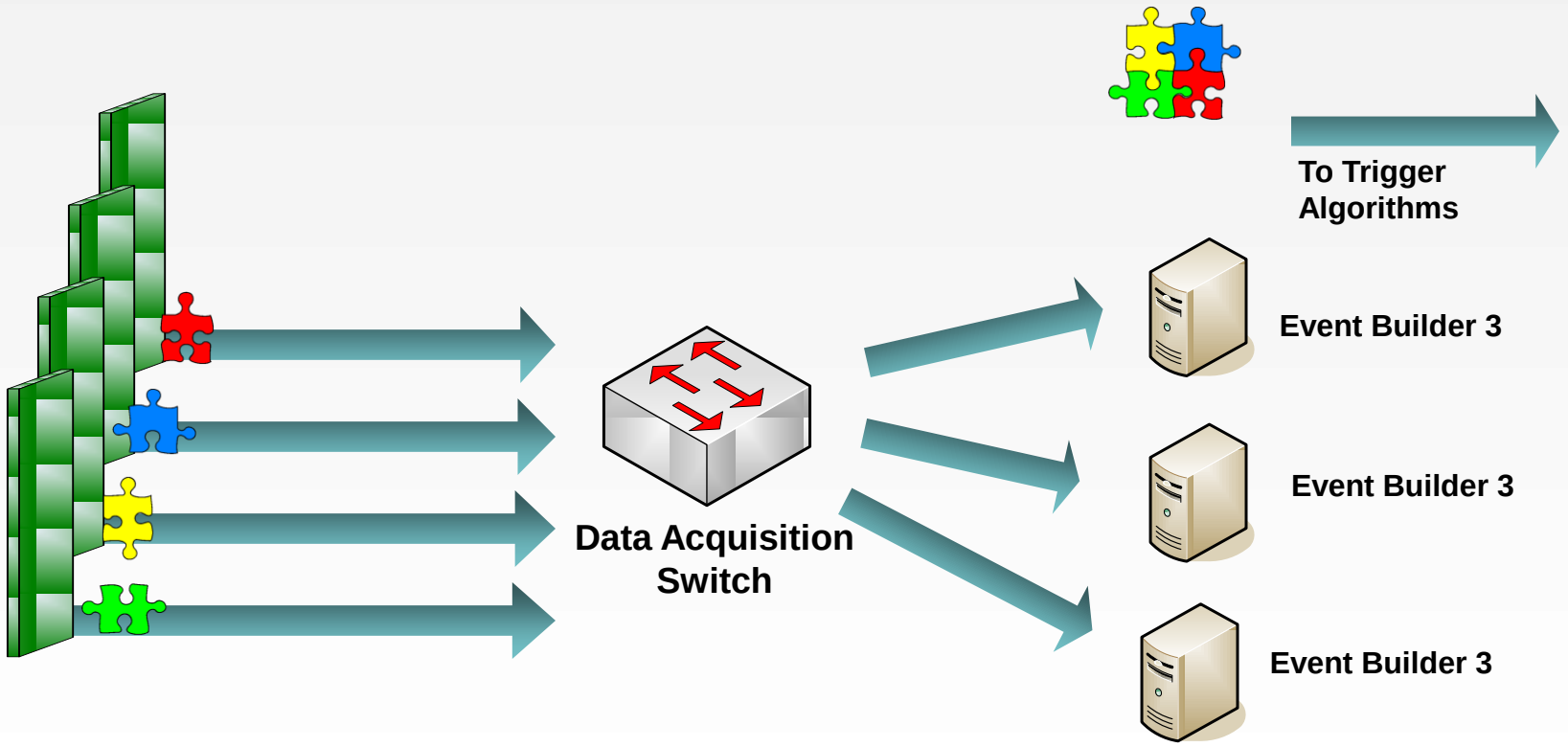
- Send everything, ask questions later (ALICE, CMS, LHCb)



- Send a part first, get better question
  Send everything only if interesting (ATLAS)

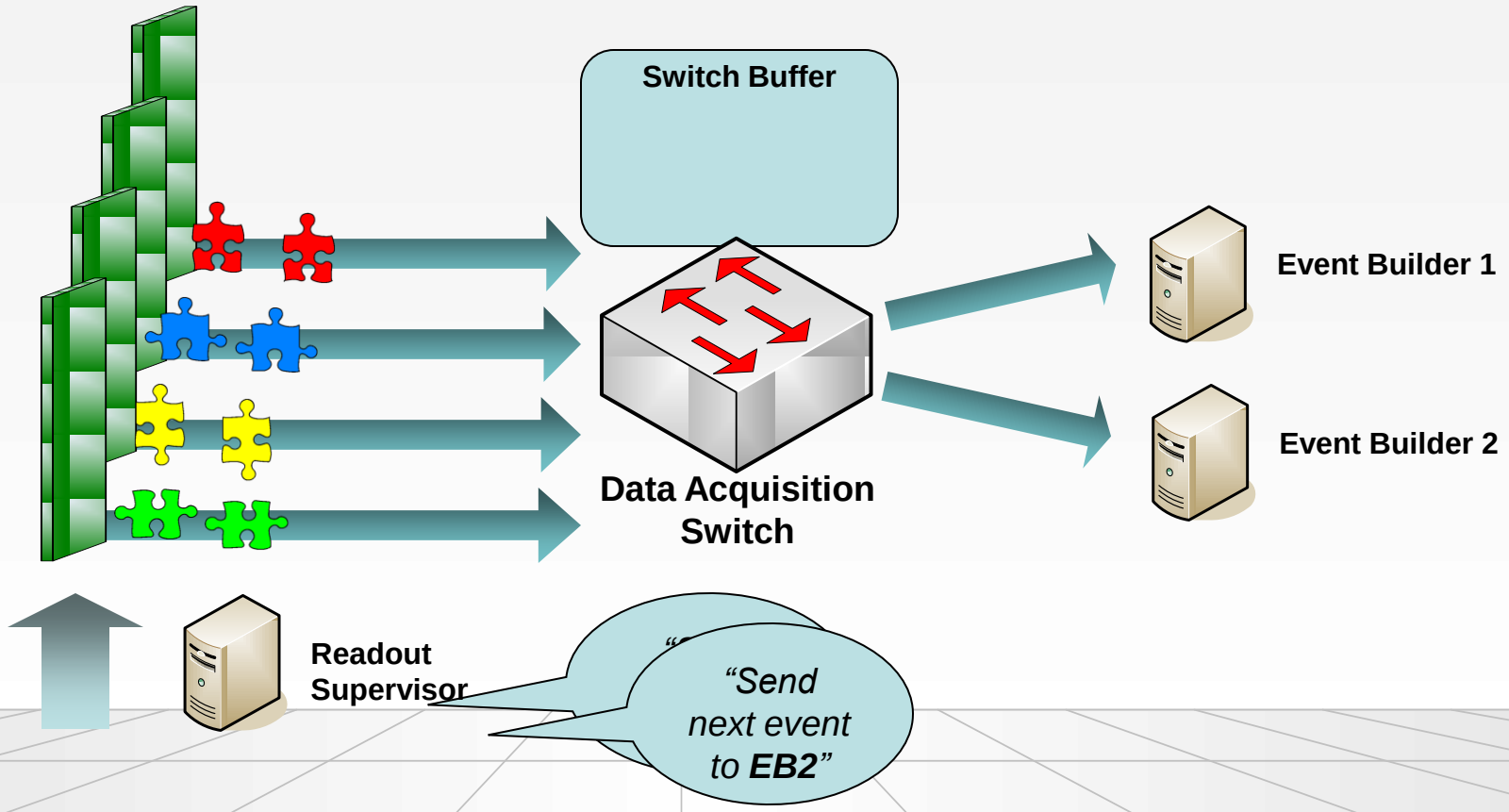


Event Building



1. Event fragments are received from detector front-end
2. Event fragments are read out over a network to an event builder
3. Event builder assembles fragments into a complete event
4. Complete events are processed by trigger algorithms

Push-Based Event Building

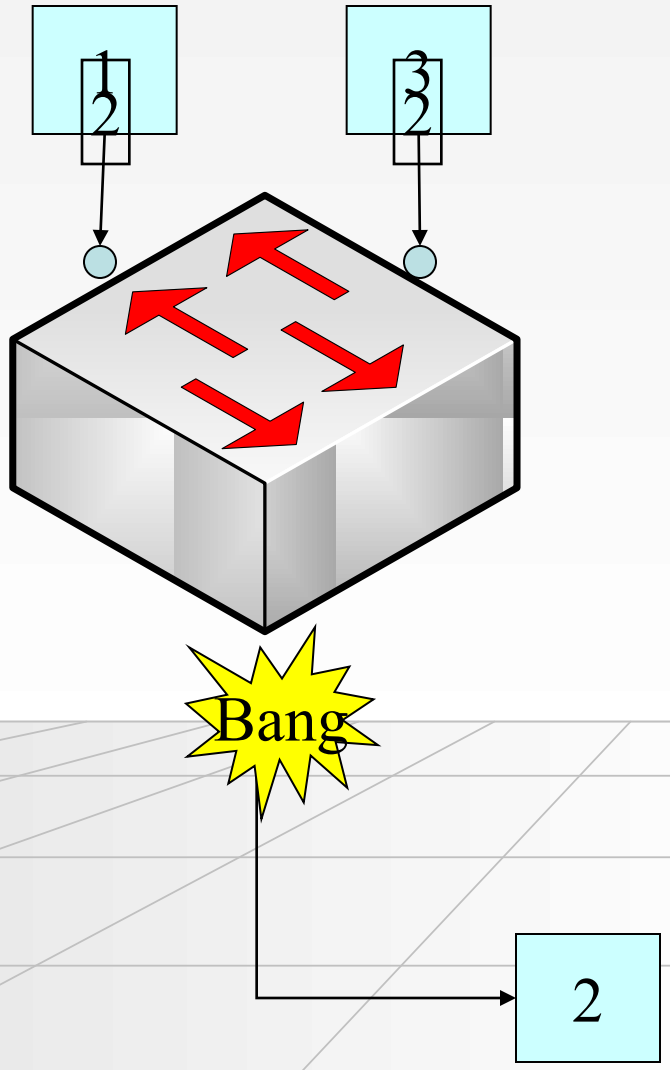


1 **Readout Supervisor tells readout boards where events must be sent (round-robin)**

2 **Readout boards do not buffer, so switch must**

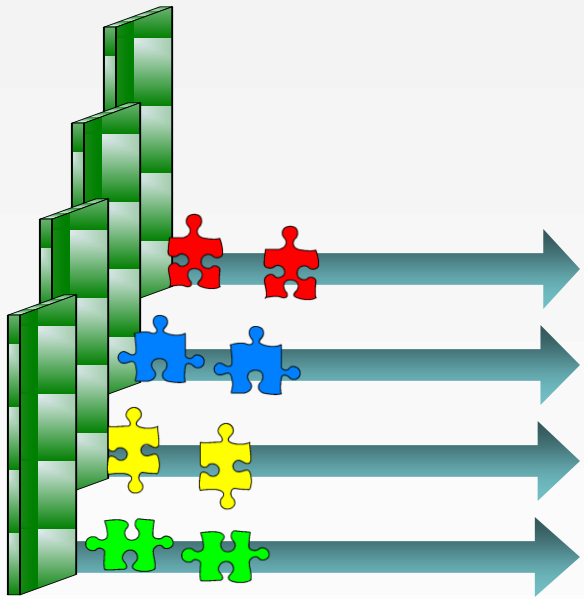
3 **No feedback from Event Builders to Readout system**

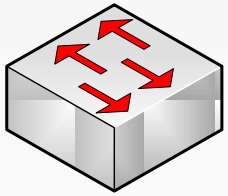
Congestion



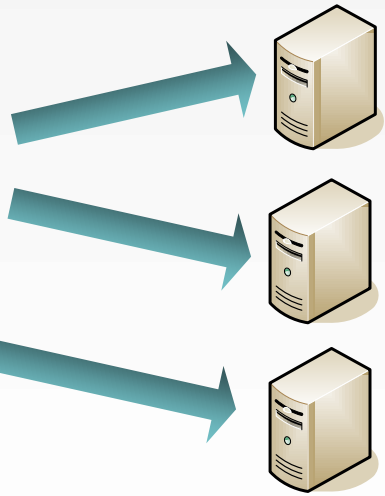
- "Bang" translates into random, uncontrolled packet-loss
- In Ethernet this is perfectly valid behavior and implemented by very cheap devices
- Higher Level protocols are supposed to handle the packet loss due to *lack of buffering*
- This problem comes from ***synchronized*** sources ***sending*** to the same destination at the ***same time***

Pull-Based Event Building





Data Acquisition Switch



Event Builder 1

Event Builder 2

Event Builder 3

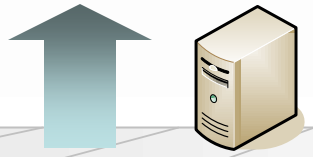
"Send me an event!"

"Send me an event!"

"Send me an event!"

Event 1

EB1: 0
EB2: 0
EB3: 0



"Send next event to **EB3**"

1 Event Builders notify Readout Supervisor of available capacity

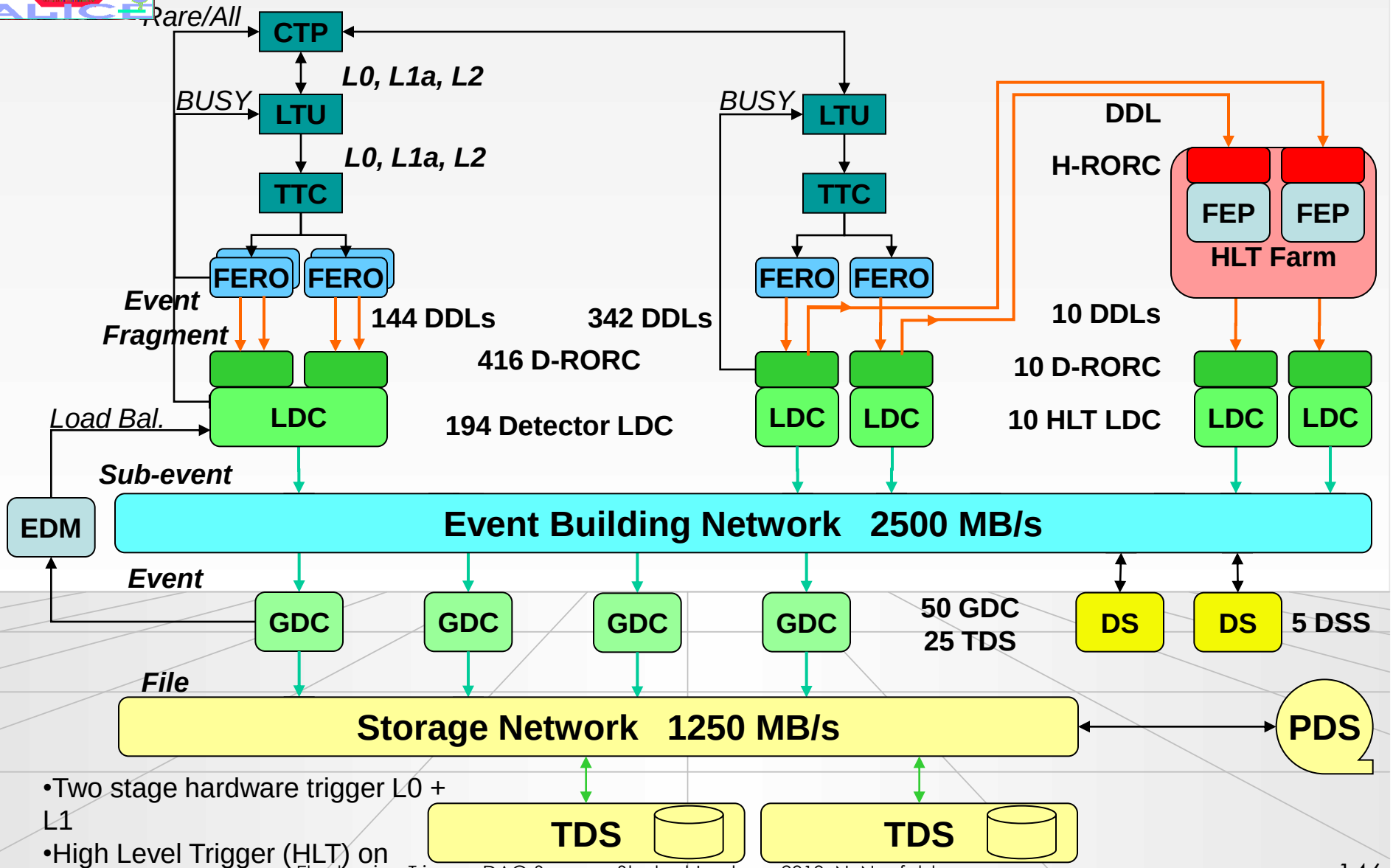
2 Readout Supervisor ensures that data are sent only to nodes with available capacity

3 Readout system relies on feedback from Event Builders

AACL

ALICE, ATLAS, CMS, LHCb
DAQs in 4 slides

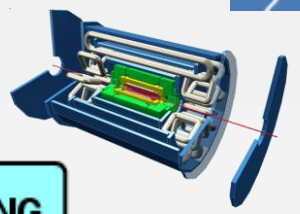
ALICE DAQ



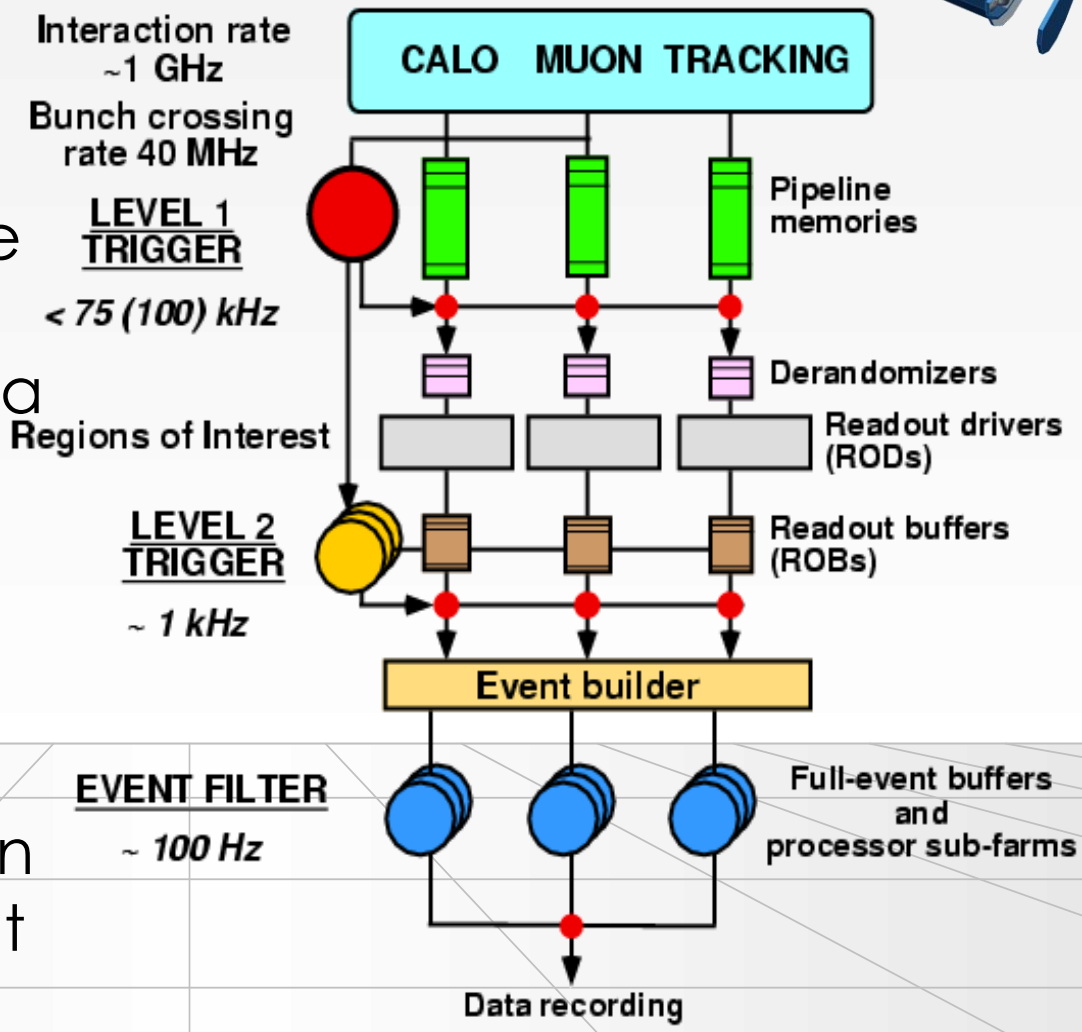
- Two stage hardware trigger L0 + L1
- High Level Trigger (HLT) on separate farm

ATLAS DAQ

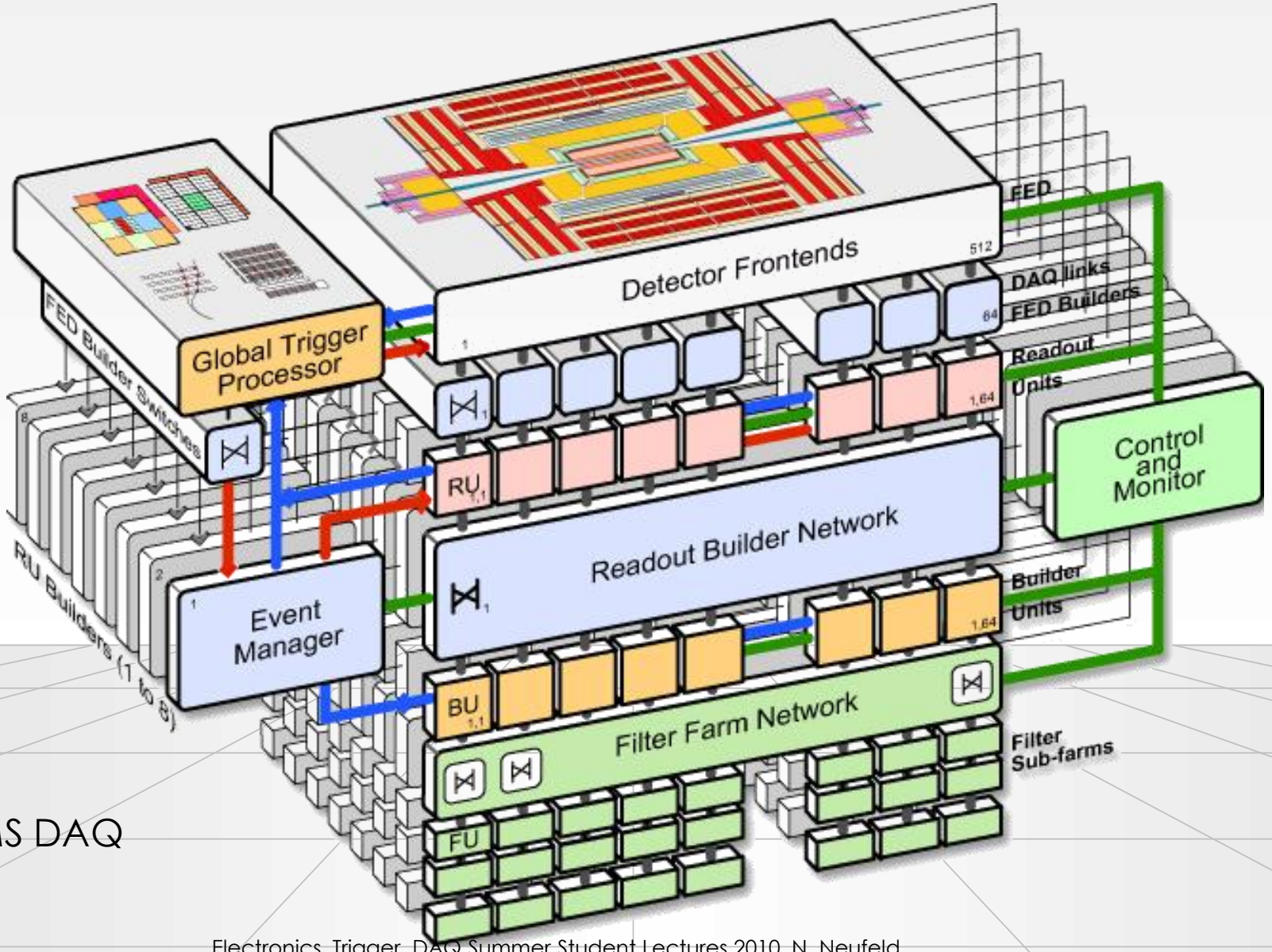
ATLAS



- L1 selects events at 100 kHz and defines *regions of interest*
- L2 pulls data from the region of interest and processes the data in a farm of processors
  L2 accepts data at ~ 1 kHz
- Event Filter reads the entire detector (pull), processes the events in a farm and accepts at 100 Hz

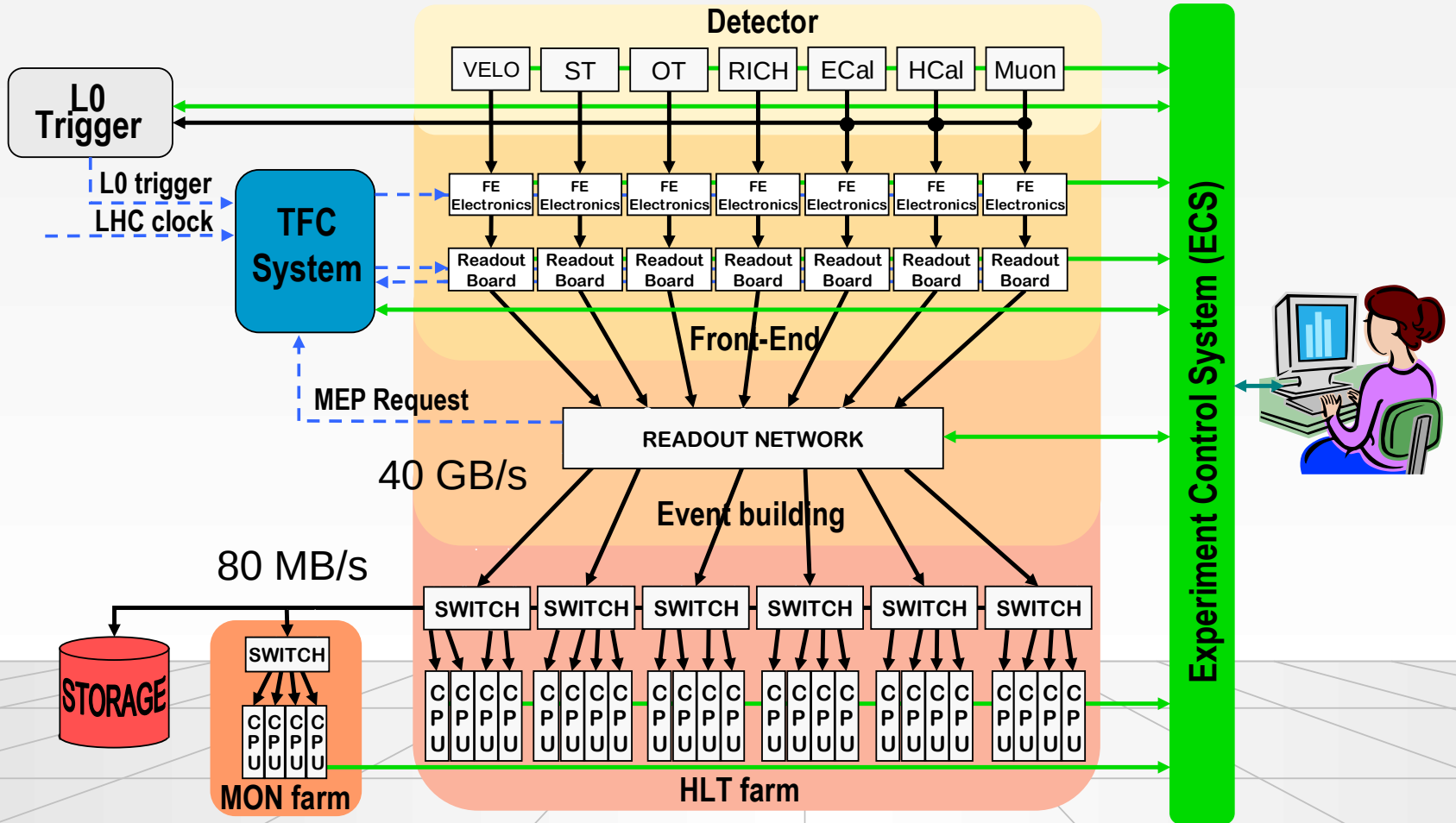


Readout Architectures



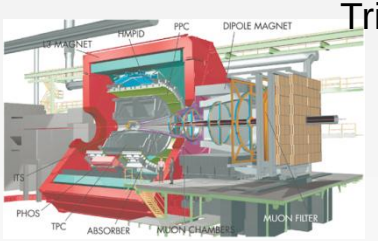
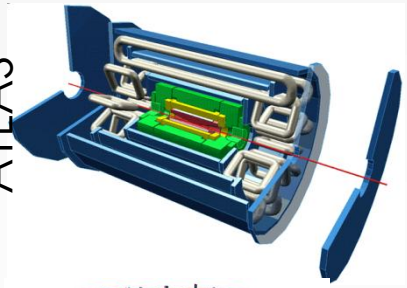
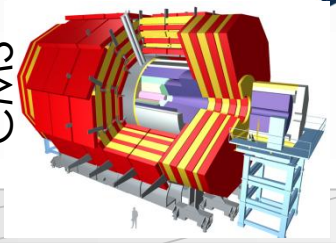
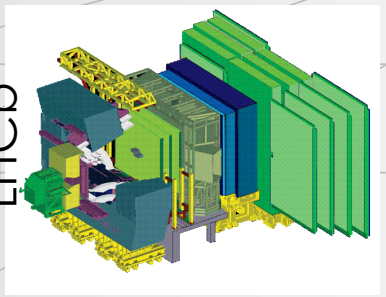
CMS DAQ

LHCb DAQ

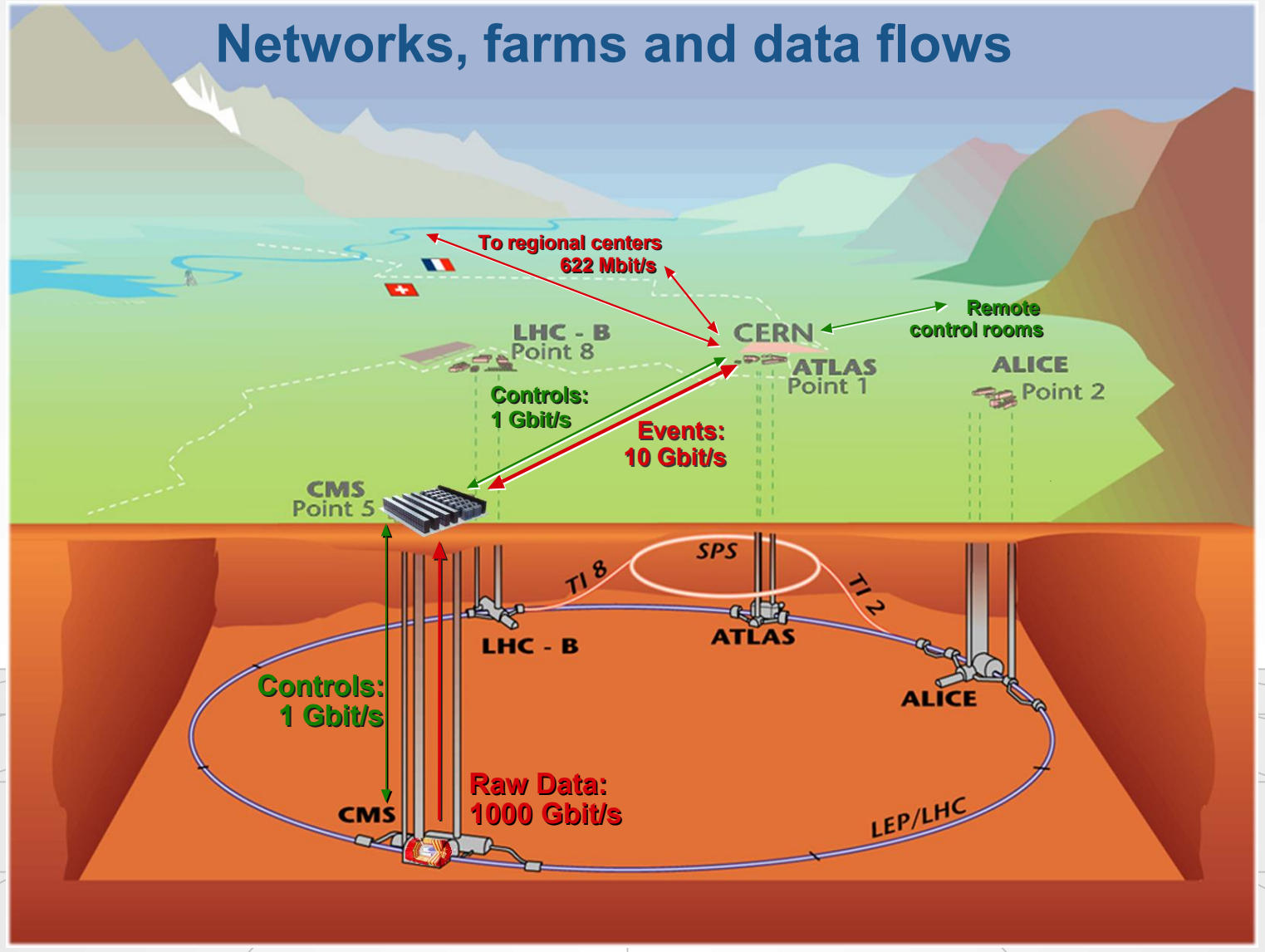


Average event size 40 kB
Average rate into farm 1 MHz
Average rate to tape 2 kHz

Trigger/DAQ parameters

	No.Levels Trigger	Level-0,1,2 Rate (Hz)	Event Size (Byte)	Readout Bandw.(GB/s)	HLT Out MB/s (Event/s)
ALICE 	4	Pb-Pb **500** p-p **10^3**	**$5x10^7$** **$2x10^6$**	**25**	**1250** (10^2) **200** (10^2)
ATLAS 	3	LV-1 **10^5** LV-2 **$3x10^3$**	**$1.5x10^6$**	**4.5**	**300** ($2x10^2$)
CMS 	2	LV-1 **10^5**	**10^6**	**100**	**~1000** (10^2)
LHCb 	2	LV-0 **10^6**	**$3.5x10^4$**	**35**	**70** ($2x10^3$)

On to tape...and the GRID



The end

The ISOTDAQ permanent lab

Publcite Advertisement

In February 2010 the first ISOTDAQ school was held in Ankara.
See: http://isotdaq.web.cern.ch/isotdaq/isotdaq/2010.html
Slides and videos can be found at:
http://indico.cern.ch/conferenceTimeTable.py?confId=68278#20100201

- Some of the exercises that were prepared for this school were (or will soon be) installed at CERN
- Interested students can register themselves to do some of these exercises. To register:
  - Add yourself to: https://twiki.cern.ch/twiki/bin/view/Sandbox/DaqSchoolLab#Requests_for_access
  - Or write an e-mail to markus.joos@cern.ch
- It is recommended to have a look at the slides / videos before doing an exercise

NOTE: We cannot guarantee that the exercises will take place. So, don't be d

Further Reading

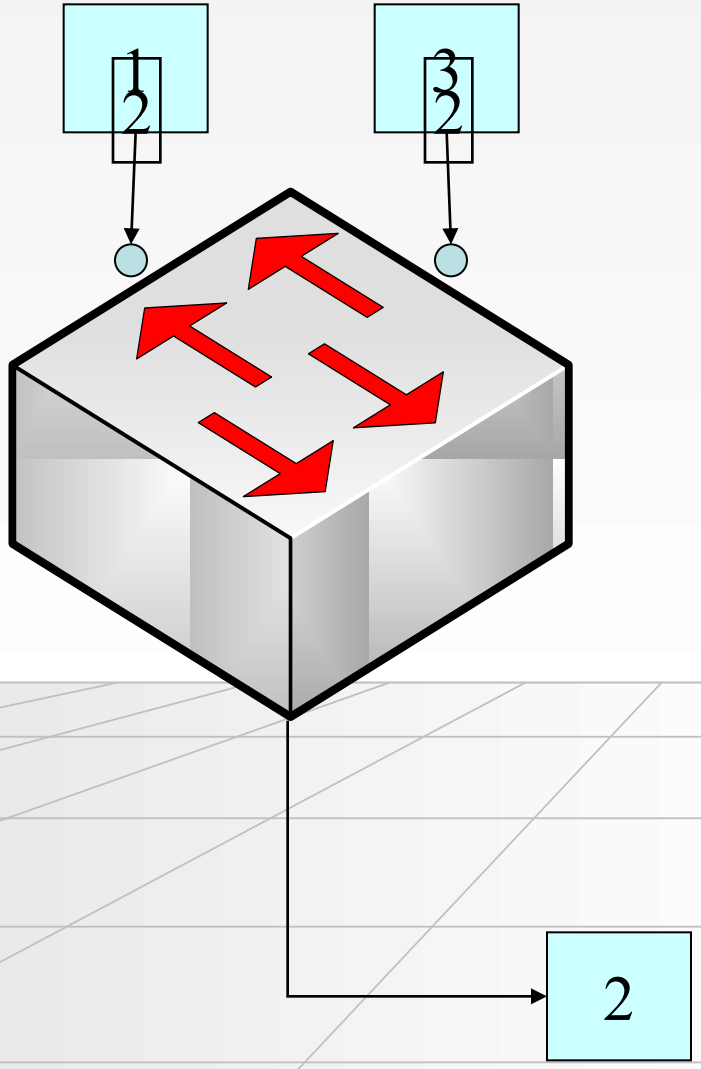
- Electronics
  - Helmut Spielers web-site: http://www-physics.lbl.gov/~spieler/
- Buses
  - VME: http://www.vita.com/
  - PCI http://www.pcisig.com/
- Network and Protocols
  - Ethernet "Ethernet: The Definitive Guide", O'Reilly, C. Spurgeon
  - TCP/IP "TCP/IP Illustrated", W. R. Stevens
  - Protocols: RFCs www.ietf.org in particular RFC1925 http://www.ietf.org/rfc/rfc1925.txt "The 12 networking truths" is required reading
- Wikipedia (!!!) and references therein – for all computing related stuff this is usually excellent

- Conferences
  - IEEE Realtime
  - ICALEPCS
  - CHEP
  - IEEE NSS-MIC
- Journals
  - IEEE Transactions on Nuclear Science, in particular the proceedings of the IEEE Realtime conferences
  - IEEE Transactions on Communications

More Stuff

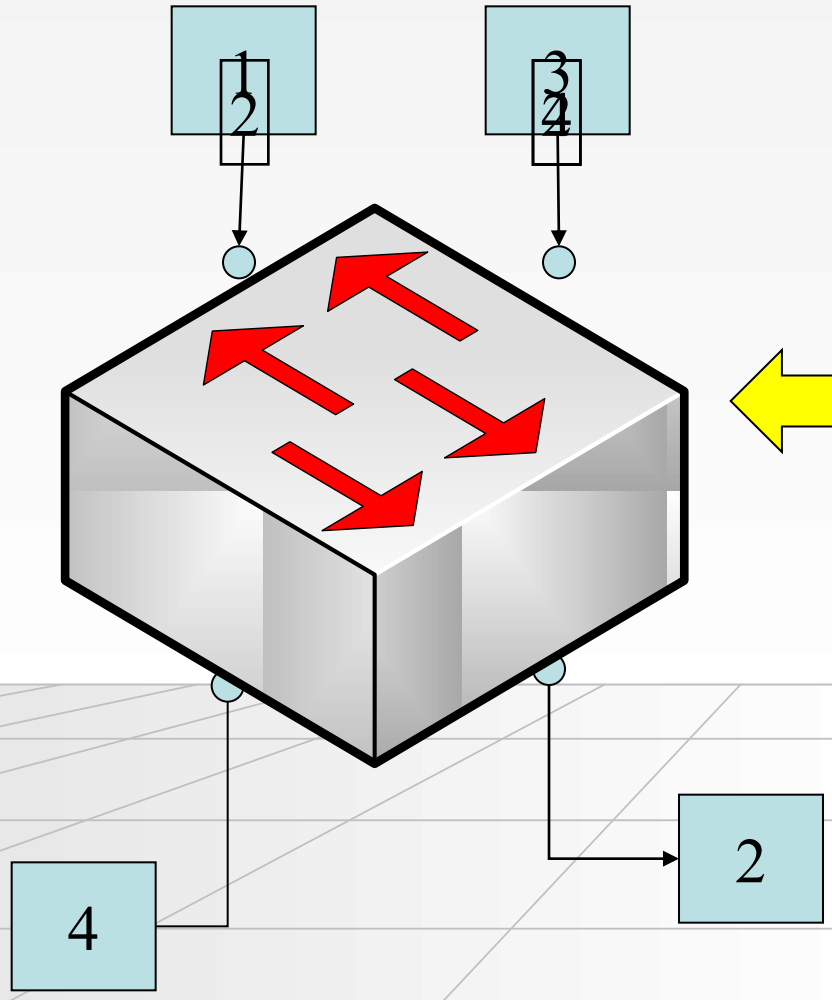
Data format, DIY DAQ, run-control

Overcoming Congestion: Queuing at the Input



- Two frames destined to the same destination arrive
- While one is switched through the other is waiting at the input port
- When the output port is free the queued packet is sent

Head of Line Blocking

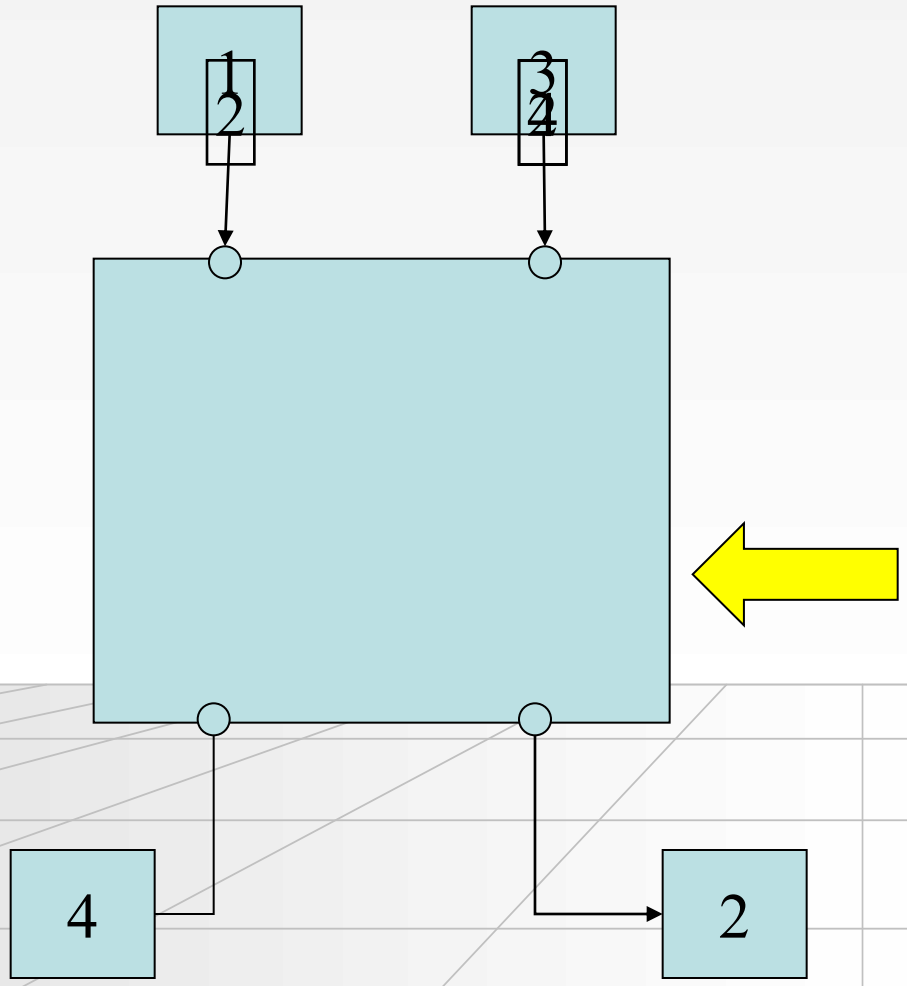


- The reason for this is the First in First Out (FIFO) structure of the input buffer
- Queuing theory tells us* that for random traffic
  - Packet to node 4 must wait even though port to node 4 is free

(and infinitely many switch ports) the throughput of the switch will go down to 58.6% → that means on 100 MBit/s network the nodes will "see" effectively only ~ 58 MBit/s

*) "Input Versus Output Queueing on a Space-Division Packet Switch"; Karol, M. et al. ; IEEE Trans. Comm., 35/12

Output Queuing



- In practice virtual output queueing is used: at each input there is a queue → for n ports $O(n^2)$ queues must be managed
- Assuming the buffers are large enough(!) such a switch will sustain random traffic at 100% nominal link load

Packet to node 2 waits at output to port 2. Way to node 4 is free

Raw data format

There are 10 kinds of people in the world

```
0000240 2828 2828 2828 2828 c4ff a201 0000 0501
0000260 0101 0101 0101 0000 0000 0000 0000 0201
0000300 0403 0605 0807 0a09 010b 0300 0101 0101
0000320 0101 0101 0001 0000 0000 0100 0302 0504
0000340 0706 0908 0b0a 0010 0102 0303 0402 0503
0000360 0405 0004 0100 017d 0302 0400 0511 2112
0000400 4131 1306 6151 2207 1471 8132 a191 2308
0000420 b142 15c1 d152 24f0 6233 8272 0a09 1716
0000440 1918 251a 2726 2928 342a 3635 3837 3a39
0000460 4443 4645 4847 4a49 5453 5655 5857 5a59
0000500 6463 6665 6867 6a69 7473 7675 7877 7a79
0000520 8483 8685 8887 8a89 9392 9594 9796 9998
0000540 a29a a4a3 a6a5 a8a7 aaa9 b3b2 b5b4 b7b6
```

```
<ADCVALUE>
<TIME>00:04:10</TIME>
<VALUE>0.2334</VALUE>
<PCISTATUS>OK</PCISTATUS>
</ADCVALUE>
<ADCVALUE>
<TIME>00:05:10</TIME>
<VALUE>0.9999</VALUE>
<PCISTATUS>ERROR</PCISTATUS>
</ADCVALUE>
<ADCVALUE>
<TIME>00:06:10</TIME>
<VALUE>0.6334</VALUE>
<PCISTATUS>OK</PCISTATUS>
</ADCVALUE>
<ADCVALUE>
<TIME>00:07:10</TIME>
<VALUE>0.8334</VALUE>
<PCISTATUS>OK</PCISTATUS>
</ADCVALUE>
```

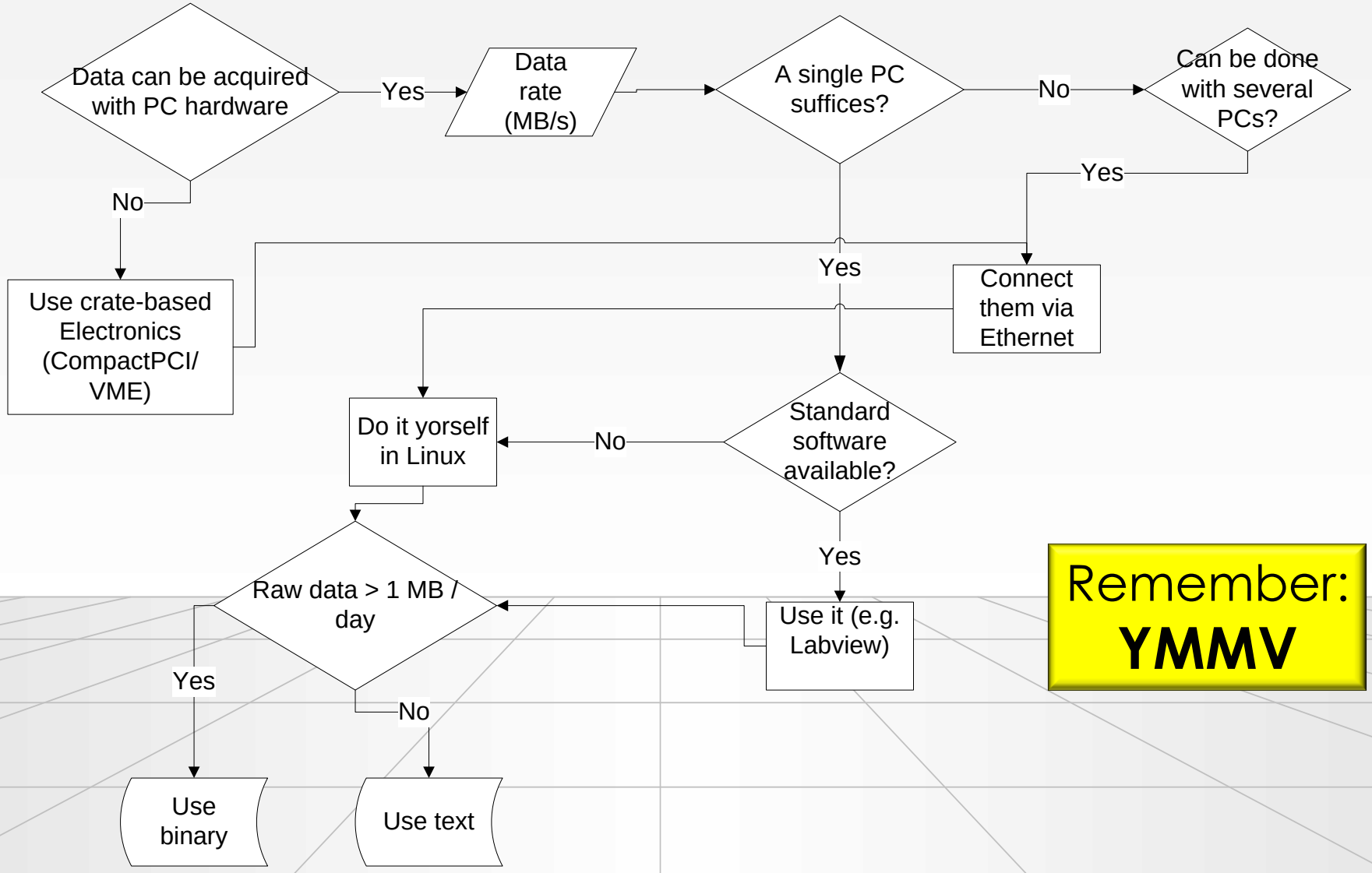
Those who can read binary and those who cannot

Binary vs Text

- 11010110 Pros:
  - compact
  - quick to write & read (no conversion)
- Cons:
  - opaque (humans need tool to read it)
  - depends on the machine architecture (endinaness, floating point format)
  - life-time bound to availability of software which can read it

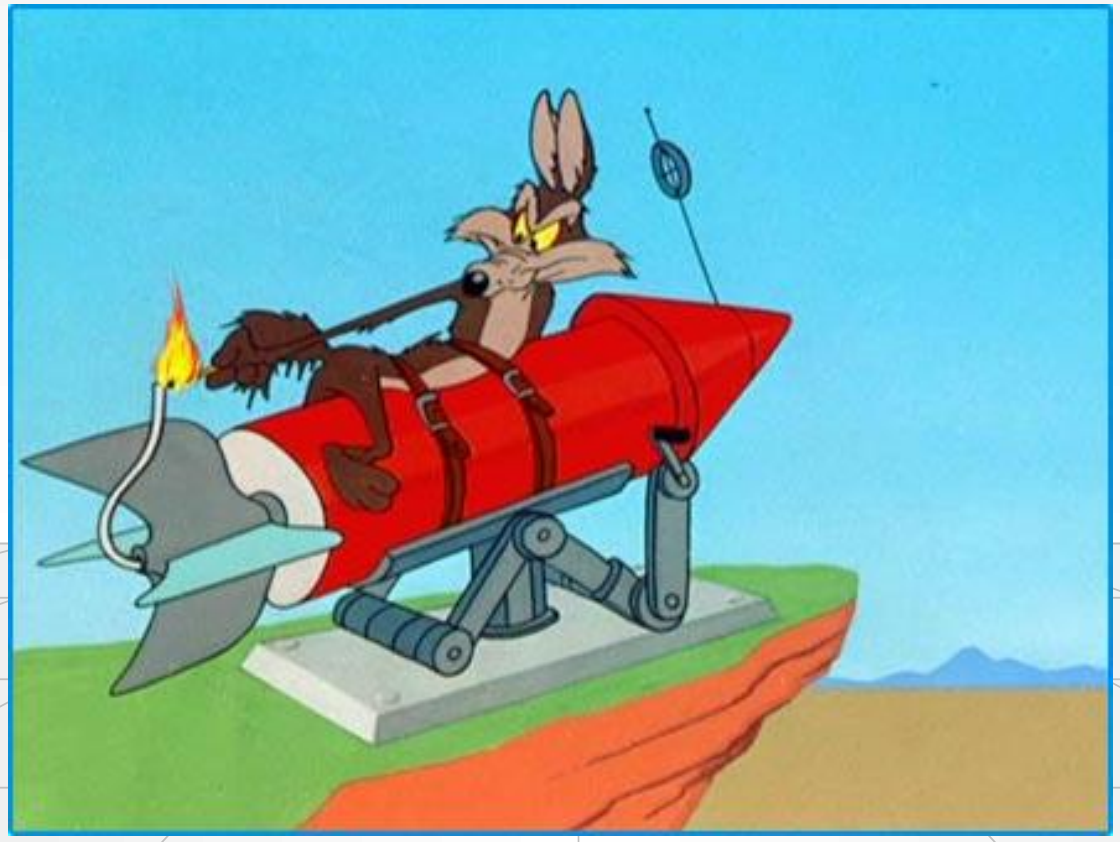
- <TEXT></TEXT> Pros:
  - universally readable
  - can be parsed and edited equally easily by humans and machines
  - long-lived (ASCII has not changed over decades)
  - machine independent
- Cons:
  - slow to read/write
  - low information density (can be improved by compression)

A little checklist for your DAQ



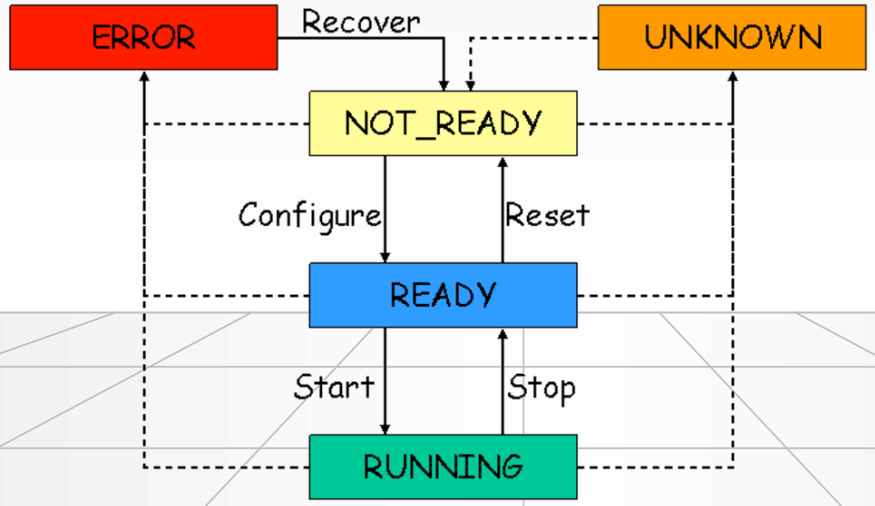
Remember:
YMMV

Runcontrol



Run Control

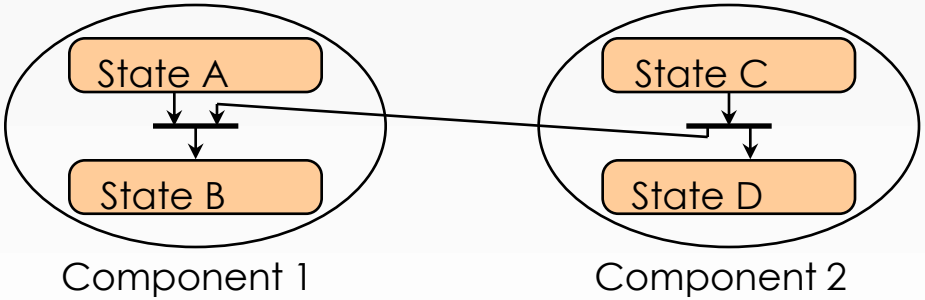
- The run controller provides the control of the trigger and data acquisition system. It is the application that interacts with the operator in charge of running the experiment.
- The operator is not always an expert on T/DAQ. The **user interface** on the Run Controller plays an important role.
- The complete system is modeled as a **finite state machine**. The commands that run controller offers to the operator are state transitions.



LHCb DAQ /Trigger Finite State Machine diagram (simplified)

Finite State Machine

- Each component, sub-component of the system is modeled as a *Finite State Machine*. This abstraction facilitates the description of each component behavior without going into detail
- The control of the system is realized by inducing transitions on remote components due to a transition on a local component



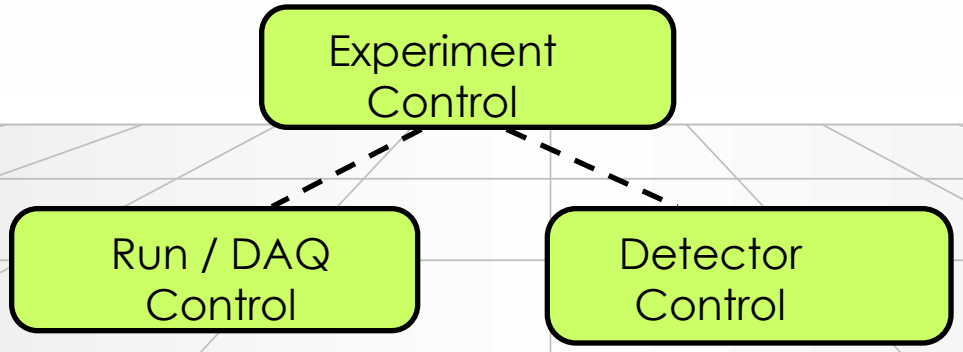
Component 1 can only complete the transition to State B if Component 2 is in state D.

- Each transition may have actions associated. The action consist of code which needs to be executed in order to bring the component to its new state
- The functionality of the FSM and state propagation is available in special software packages such as SMI

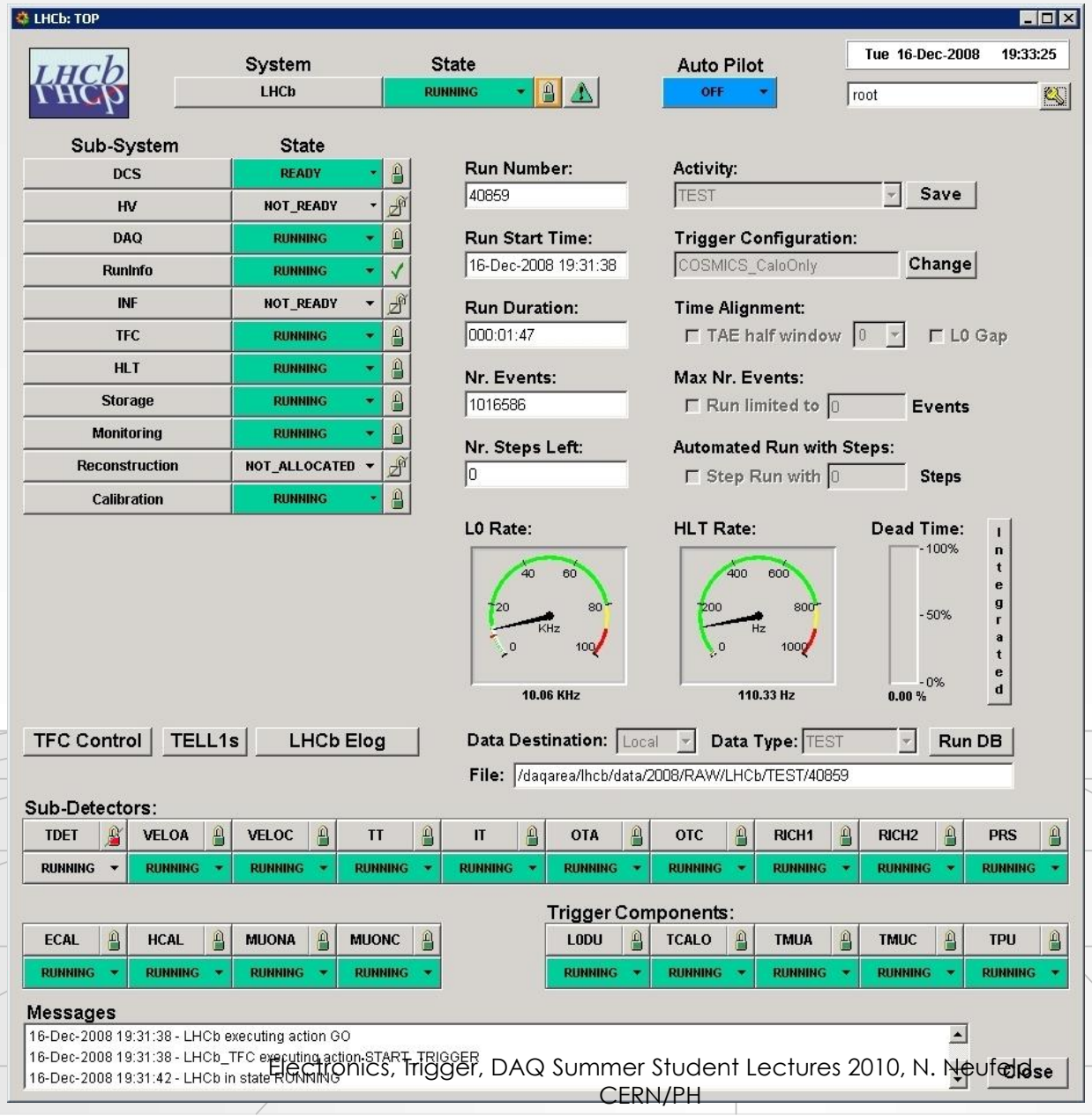
Detector Control

- The detector control system (DCS) (also Slow Control) provides the monitoring and control of the detector equipment and the experiment infrastructure.
- Due to the scale of the current and future experiments is becoming more demanding: for the LHC Experiments: ≈ 100000 parameters

Control hierarchy



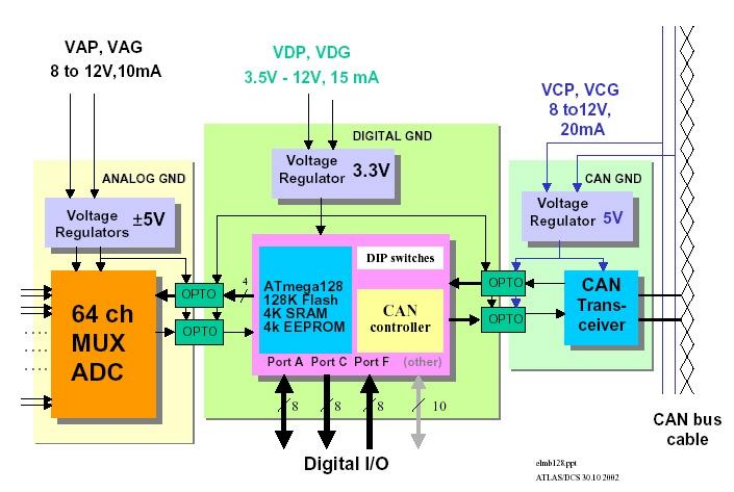
Run Control GUI

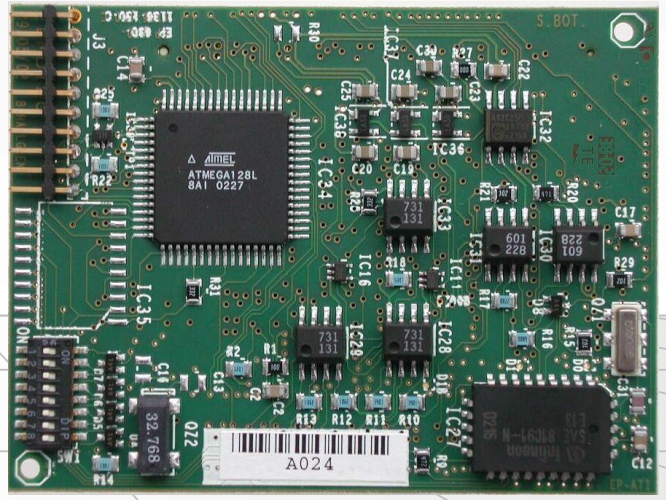


Main panel of the LHCb run-control (PVSS II)

Control and monitoring

- Access to setup registers (must have read-back)
- Access to local monitoring functions
  - Temperatures, power supply levels, errors, etc.
- Bidirectional with addressing capability (module, chip, register)
- Speed not critical and does not need to be synchronous
  - Low speed serial bus: I^2C, JTAG, SPI
- Must be reasonably reliable (read-back to check correct download and re-write when needed)





Example: ELMB

Online Trigger Farms 2009

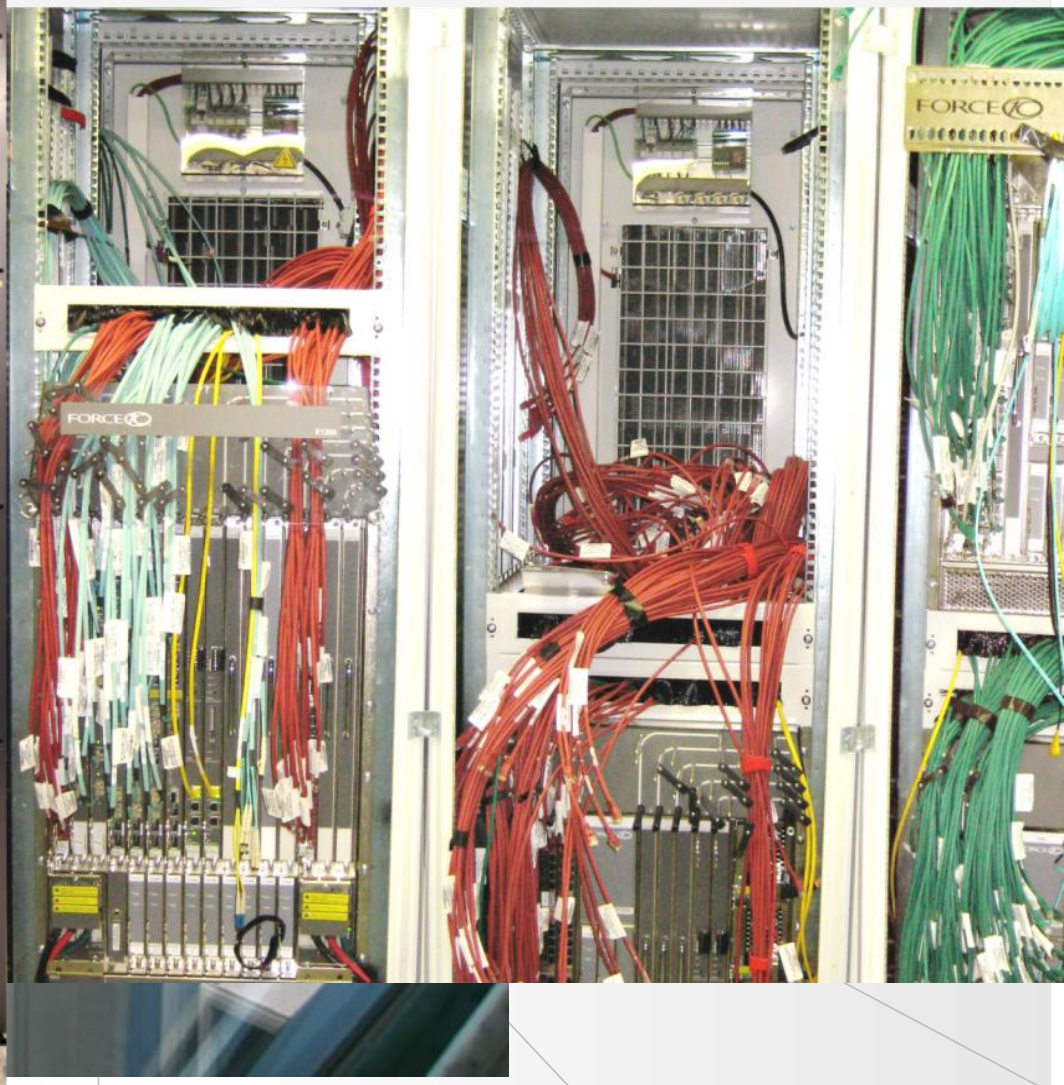
	ALICE	ATLAS	CMS	LHCb	CERN IT
# servers	81[1]	837	900	550	5700
# cores	324	~ 6400	7200	4400	~ 34600
total available power (kW)		~ 2000[2]	~ 1000	550	2.9 MW
currently used power (kW)		~ 250	450[3]	~ 145	2.0 MW
total available cooling power	~ 500	~ 820	800 (currently)	525	2.9 MW
total available rack-space (Us)	~ 2000	2449	~ 3600	2200	n/a
CPU type(s)	AMD Opteron	Intel Hapertown	Intel (mostly) Harpertown	Intel Harpertown	Mixed (Intel)

(1) 4-U servers with powerful FPGA preprocessor cards H-RORC
(2) Available from transformer (3) PSU rating

Gallery

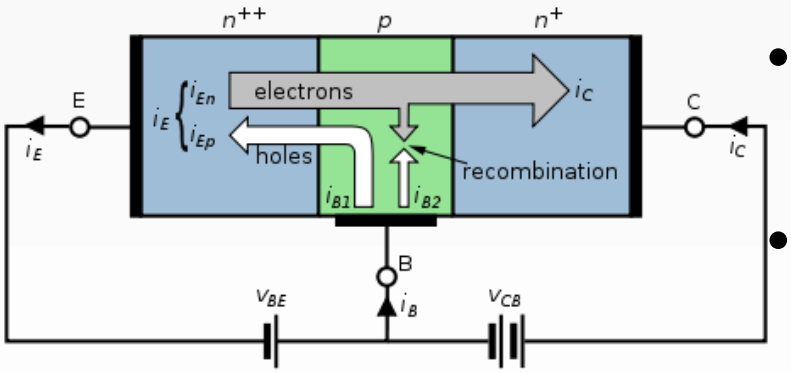
ALICE Storage System

Online Network Infrastructure



Even more stuff

Transistors

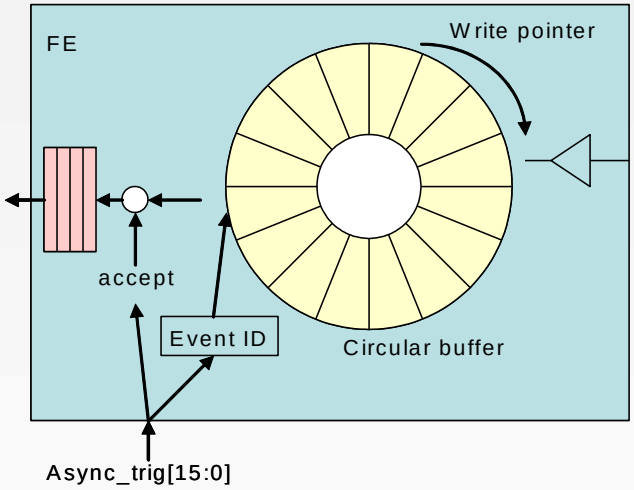


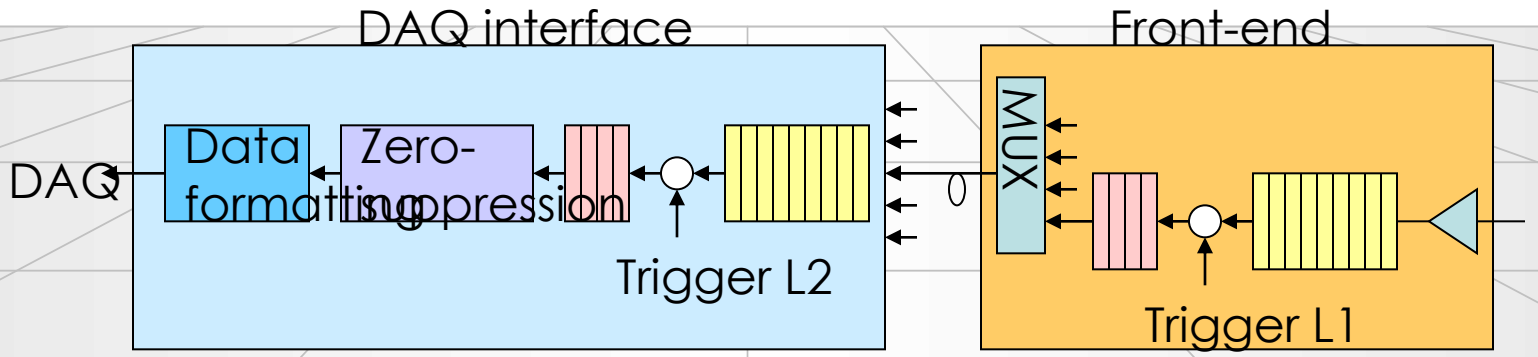
from Wikipedia

- Exampe: bi-polar transistor of the NPN type
- C collector, E emitter, B Base
- EB diode is in forward bias: holes flow towards np boundary and into n region
- BC diode is in reverse bias: electrons flow AWAY from pn boundary
- p layer must be thinner than diffusion length of electrons so that they can go through from E to N without much recombination

Multilevel triggering

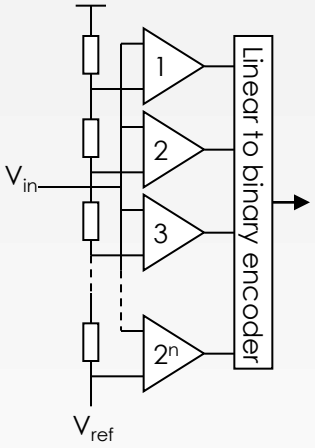
- First level triggering.
  - Hardwired trigger system to make trigger decision with short latency.
  - Constant latency buffers in the front-ends
- Second level triggering in DAQ interface
  - Processor based (standard CPU's or dedicated custom/DSP/FPGA processing)
  - FIFO buffers with each event getting accept/reject in sequential order
  - Circular buffer using event ID to extracted accepted events
    - Non accepted events stays and gets overwritten by new events
- High level triggering in the DAQ systems made with farms of CPU's: hundreds – thousands. (separate lectures on this)

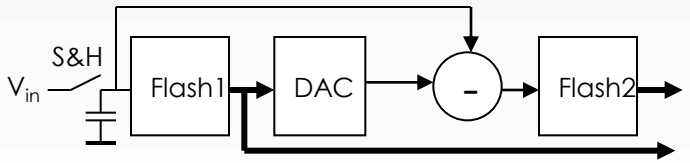




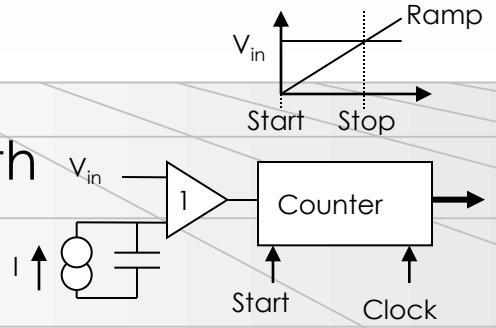
ADC architectures

- Flash
  - A discriminator for each of the 2^n codes
  - New sample every clock cycle
  - Fast, large, lots of power, limited to ~8 bits
  - Can be split into two sub-ranging Flash $2x2^{n/2}$ discriminators: e.g. 16 instead of 256 plus DAC
    - Needs sample and hold during the two stage conversion process



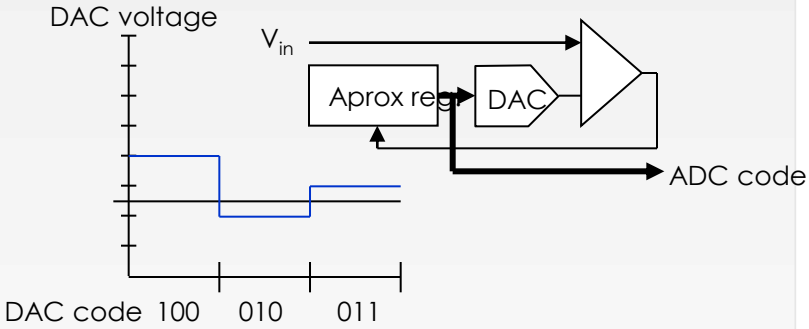


- Ramp
  - Linear analog ramp and count clock cycles
  - Takes 2^n clock cycles
  - Slow, small, low power, can be made with large resolution

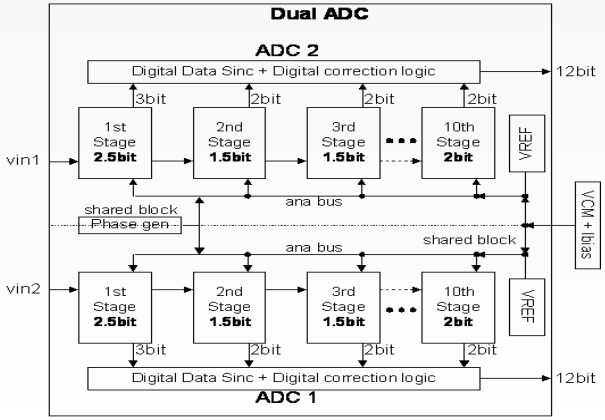


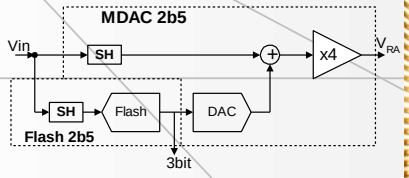
ADC architectures

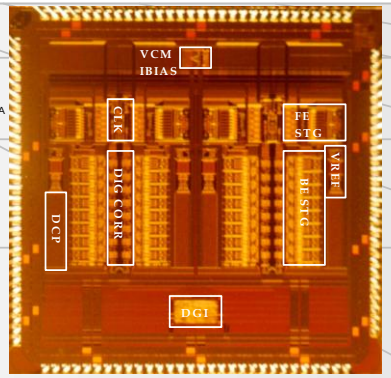
- Successive approximation
  - Binary search via a DAC and single discriminator
  - Takes n clock cycles
  - Relatively slow, small, low power, medium to large resolution



- **Pipelined**
  - Determines "one bit" per clock cycle per stage
    - Extreme type of sub ranging flask
  - n stages
  - In principle 1 bit per stage but to handle imperfections each stage normally made with ~2bits and n*2bits mapped into n bits via digital mapping function that "auto corrects" imperfections
  - Makes a conversion each clock cycle
  - Has a latency of n clock cycles
    - Not a problem in our applications except for very fast triggering
  - Now dominating ADC architecture in modern CMOS technologies and impressive improvements in the last 10 years: speed, bits, power, size







ADC imperfections

- Quantization (static)
  - Bin size: Least significant bit (LSB) $=V_{max}/2^n$
  - Quantization error: RMS error/resolution: $\mathrm{LSB}/\sqrt{12}$
- Integral non linearity (INL): Deviation from ideal conversion curve (static)
  - Max: Maximum deviation from ideal
  - RMS: Root mean square of deviations from ideal curve
- Differential non linearity (DNL): Deviation of quantization steps (static)
  - Min: Minimum value of quantization step
  - Max: Maximum value of quantization step
  - RMS: Root mean square of deviations from ideal quantization step
- Missing codes (static)
  - Some binary codes never present in digitized output
- Monotonic (static)
  - Non monotonic conversion can be quite unfortunate in some applications.
    A given output code can correspond to several input values.